FUJ00083961

CHARTERIS

Subpostmasters v Post Office Limited
Expert Report of Dr Robert Worden

Draft 68
27 November 2018

Table of Contents

FUJ00083961

CHARTERIS

FUJ00083961

CHARTERIS

FUJ00083961

1. INTRODUCTION

1.1 Background of the case

1

1.2 Experience

2

1.3 Sources of information

1.4 Document structure

3 Section 2 gives a summary of my opinions on the Horizon issues.

4 Section 3 describes the business requirements for the Horizon system, including the point of sale requirement in PO branches, the accounting requirement, and the range of services that PO offers on behalf of its clients

5 The Horizon system has undergone frequent changes, in a complex history since its inception in 1999. I will describe its architecture in two main time periods.

6 Section 4 describes the Horizon system as it was in the period 2000 - 2009. After giving a central 'snapshot' of this period, I describe the most important changes during that period

7 Section 5 describes 'Horizon New Generation' (HNG-X, and later HNG-A), introduced in 2010, in which a major element of the architecture was changed. In HNG, instead of holding persistent transaction data in each branch, all transaction data was held centrally in a single branch database. HNG involved a complete refresh of the hardware and software in the branches. Many central elements of Horizon persisted over both periods, as will be described in section 6.

8 Sections 4 and 5 both describe complex Horizon architectures, with many major components (most of which are in scope for the Horizon trial), and interactions between the components which can be partially understood from architecture diagrams, showing how data are passed between the systems. However, this still leaves many topics incompletely described, which will be important background for the Horizon trial.

9 Section 6 builds on the previous sections to address these topics across the whole sequence of Horizon architectures. The topics described in section 6, with reference to specific Horizon architectures, are:

- How the Horizon architecture supports User Error Detection - including stock checks made in the branches, branch balancing and rollover processes, reconciliation with external systems, several kinds of audit, and other management checks which are made on Horizon data.
- How the Horizon architecture supports Intrinsic Error Prevention - including double entry checks and cross-checks of different versions of the same data.
- The uses of reference data and data-driven software
- Architecture and business processes for Reconciliation, Transaction Corrections and Transaction Acknowledgements
- Measures for security and user authentication

FUJ00083961

- Hardware and software resilience of the whole Horizon system - including hardware redundancy, and resilience of underlying software components such as DBMS and communication software. This is needed to understand the possible impact on branch accounts of various kinds of interruption of the Horizon service, including communication failures.
- Processes for development and testing of Horizon
- Fujitsu and PO processes for supporting Horizon in service, including error detection and correction.

10 Sections 4, 5, and 6 together provide the basis for understanding the whole Horizon architecture, how it was intended to achieve robustness against a variety of threats, and the extent to which it did achieve robustness. This understanding is, in my opinion, the essential basis for approaching the Horizon issues.

11 The 15 Horizon issues are addressed in five groups, in the following order:

12 Section 7 addresses the Horizon issues which mainly concern the robustness of Horizon. These are issues 3, 4, and 6.

13 Section 8 addresses Horizon issue 1, the extent to which bugs in Horizon may have affected the claimants' branch accounts.

14 Section 9 addresses Horizon issues 5 and 15, which concern reconciliation and transaction corrections.

15 Section 10 addressed those Horizon issues which concern the access of subpostmasters to information. These are issues 2, 9 and 14.

16 Section 11 addressed those Horizon issues which concern the facilities available to PO centrally. These are issues 7, 8, 10, 11, 12, and 13.

17 Appendix A contains a glossary of terms used in this report.

18 Appendix B gives some background on accounting systems and the principles of double entry book-keeping which they embody.

19 Appendix C gives a detailed technical analysis of those bugs and KELs which have required detailed analysis, when addressing Horizon issue 1.

20 Appendix D is a tabular survey of the KELs which I have examined in order to estimate the impact on claimants' branch accounts of all detected bugs and errors in Horizon, when addressing Horizon issue 1.

1.5 Duty to the court

21

FUJ00083961

2. EXPERT ISSUES AND SUMMARY OF OPINIONS

22 In this summary and in the body of my report I have grouped the 15 Horizon issues into five groups of related issues, so that my opinions can be organised as clearly as possible. The groups have overlaps with each other and therefore require some cross-referencing between groups. In what follows, for each group, I will first state the Horizon issues, and then give a summary of my opinions. In section 2.6, I summarise where my opinions differ from Mr Coyne's opinions, as expressed in his report.

2.1 Robustness of Horizon

23 The first group of issues, which addresses the robustness of Horizon, includes the Horizon issues 3, 4, and 6.

24 **Issue 3**: To what extent and in what respects is the Horizon System "robust" and extremely unlikely to be the cause of shortfalls in branches?

25 **Issue 4**: To what extent has there been potential for errors in data recorded within Horizon to arise in (a) data entry, (b) transfer or (c) processing of data in Horizon?

26 **Issue 6**: To what extent did measures and/or controls that existed in Horizon prevent, detect, identify, report or reduce to an extremely low level the risk of the following:

a) data entry errors;

b) data packet or system level errors (including data processing, effecting, and recording the same);

c) a failure to detect, correct and remedy software coding errors or bugs;

d) errors in the transmission, replication and storage of transaction record data; and

e) the data stored in the central data centre not being an accurate record of transactions entered on branch terminals?

27 In my opinion the most important of these is issue 3, which encompasses a large and mature area of modern IT practice. Nearly all business IT systems need to be robust - as the business depends on them - and there is a large, mature and tested set of techniques for achieving robustness. Issues 4 and 6 then in effect address some specific subsets of issue 3.

28 The term robustness (which is closely related to resilience) receives its meaning in the phrase 'robust against X', where X is some threat. In the following paragraph, I list some of the main types of threat.

29 Horizon's main purpose is to support the financial transactions and management of PO branch accounts - through from the point of sale in the branches to PO's annual financial accounts (although aspects of this are outside the scope of the Horizon trial). To meet this requirement, Horizon and related systems must provide a very high degree of confidence to subpostmasters and to PO management that the resulting accounts are robust against:

a) Failures in Horizon hardware, software, or communications

b) Errors in business processes made outside Horizon

c) User errors made on Horizon, in the branches or in the back office

d) Deliberate errors of any kind - e.g. fraud

FUJ00083961

e) Software errors and bugs

30 Each one of these threats is a broad heading, encompassing many different specific risks, all of which must be guarded against.

31 It is important to understand that robustness against these threats does not require perfection. The threats include imperfections in the IT system or in its surrounding environment. Robustness does not consist in entirely abolishing these threats - it consists in managing their effects, so that they are acceptable.

32 Parts of the claimants' case appear to be based on misunderstanding this point. They appear to think that any evidence that Horizon was not perfect is also evidence that it was not robust. This is not the case. No business IT system is perfect or bug-free. The question of robustness is an entirely separate question - given that there were imperfections, how well were they handled, and their effects contained?

33 Because imperfections are a fact of life, and because business IT systems need to be robust, there is an extensive, mature and tested set of techniques for achieving robustness. In this report, I shall refer to these techniques as 'countermeasures'. Although the term is not widely used in the IT industry, it is a concise label which conveys the meaning. A countermeasure is a technique that is used to address one or more types of threat. One type of threat may be addressed by a variety of countermeasures, acting in concert.

34 In this report, I shall focus on eighteen categories of countermeasure, which are summarised in the table. [The master version of this table is in section 7.4 and will be copied here at a late stage.]

35 For each robustness countermeasure, I have provided a three-letter acronym to identify it. Most of these acronyms are not commonly used in the industry but are provided to enable the reader to recognise and cross-reference the different countermeasures, in the many places where they will appear in this report.

No.	Countermeasure	Explanation and examples	Described in Section
1	Reliable and redundant hardware (RHW)	Redundancy guards against many types of hardware failure. Examples: RAID discs, disaster recovery sites.	
2	Robust data communication and replication (ROC)	Communication systems and protocols are designed to recover from and protect against many kinds of communication failure. Examples: TCP/IP, Riposte.	
3	Transactional Integrity and database recovery (TIN)	Database management systems provide many facilities so that numerous kinds of failure cannot leave the data in an inconsistent, unusable state, or lose any data that have been previously stored	
4	Defensive programming (DEP)	Software is divided into small self-contained modules, which do not assume that other modules are correct, but defend themselves by checking their inputs and raising alerts early	
5	Generic, data driven software (DDS)	Different use cases for software often have much in common. Software is written generically to be able to handle the different cases, using reference data to define which use case is to be handled. Example: variations in PO client products are handled by reference data.	

FUJ00083961

6	Secure kernel hardware and software (SEK)	When a large complex IT system is subject to threats, the design may include a small, well tested and secure kernel which is proof against those threats. Examples: secure kernels of operating systems, Horizon core audit process.	
7	Redundant data storage and computing, with cross-checks (RDS)	In large IT systems and sets of systems, data are stored redundantly in several places, and routine operations check automatically that the different copies of the data remain consistent.	
8	Double entry accounting (DEA)	Accounting systems operate by the principles of double entry book keeping, so that any change to the accounts must be made in a transaction whose summed effect on all accounts is zero. Transactions which do not obey this constraint are rejected.	
9	Early detection of user errors (DUE)	At the point of user input, as many checks as possible are made of the correctness of the input - so that the system will not accept erroneous input and may warn the user of errors.	
10	Later correction of user errors (UEC)	In accounting systems, the system's version of reality is periodically checked against external versions of reality and corrected if wrong. Examples: cash balancing and rollover, reconciliation and Transaction Corrections (TCs).	
11	Manual workarounds (WOR)	Whenever any part of Horizon does not work as required, there may be potential to define and apply manual workarounds.	
12	Testing good practice (TGP)	The purpose of system testing is not to prove that the system is correct, but to prove that it is incorrect in any way possible. Examples: regression testing, user testing, testing edge cases.	
13	Manual Inspection of data (MID)	Any large business IT system is used by many people, who view its outputs and check them against each other for consistency, and against their own knowledge of the business. Subpostmaster (SPMs), monitoring their branch accounts, were a key component of this.	
14	Bug Finding and Correction (BFC)	Whenever the system shows any anomalous behaviour, that is investigated, its causes found and corrected. Interim workarounds are deployed. Extra checks may be added to ensure that other similar threats are handled correctly.	
15	Large-scale IT architecture (ARC)	In any large IT estate, principles of IT architecture are used to achieve robustness - such as using a distributed network of loosely coupled sub-systems with clearly distinguished functions. The sub-systems are built to well-defined standards with clear interfaces.	
16	Managing non-functional requirements (NFR)	Robustness is improved by paying close attention to non-functional requirements and the associated 'ilities' such as manageability, supportability, maintainability and adaptability.	

36 There is generally no simple relationship between threats and countermeasures. While some threats are mainly addressed by a few countermeasures, there are other threats which are addressed by several countermeasures, acting in concert in diverse ways.

FUJ00083961

37 Much of sections 4, 5, and 6 has been devoted to describing how these countermeasures have been built into the architecture of Horizon.

38 From this analysis I conclude that, in building and supporting Horizon, Fujitsu have successfully followed industry best practice in making Horizon a robust system. While I have found some ways in which Horizon and the processes around it are not as robust as they might be, these have been few, compared to the many ways in which robustness was achieved.

39 This conclusion is reinforced by analysis I have made for the other Horizon issues, particularly for issue 1, the extent of bugs affecting claimant accounts. When looking at specific evidence about anomalous effects in Horizon - such as specific KELs and Peaks, which may be evidence of bugs in Horizon, or simply of user errors - I have repeatedly found that these KELs and Peaks provide evidence of the robustness countermeasures built into Horizon and showed that they worked.

40 In this respect, one type of countermeasure has been particularly important. Horizon has essential checks built into it to detect and correct the effects of user errors. These include monthly stock checks, reconciliation and transaction corrections.

41 There are large classes of errors of type (a) in paragraph 26 above, arising from software bugs in Horizon, whose only effect is to mimic, or possibly to increase the chances of, some error in types (b) - (d). So, while these software bugs may produce temporary anomalies in branch accounts, the same correction processes will remove those errors before long, and there will be no permanent effect on branch accounts.

2.2 Extent of Bugs in Horizon

42 The second group of issues consists of Horizon issue 1, which concerns bugs in Horizon which might have had an impact on branch accounts:

43 **Issue 1**: To what extent was it possible or likely for bugs, errors or defects of the nature alleged at §§ 23 and 24 of the GPOC and referred to in §§ 49 to 56 of the Generic Defence to have the potential to (a) cause apparent or alleged discrepancies or shortfalls relating to Subpostmasters' branch accounts or transactions, or (b) undermine the reliability of Horizon accurately to process and to record transactions as alleged at §24.1 GPOC?

44 It is necessary to define measures for the 'extent' of issue 1. At least two measures are possible: the number of such bugs and errors, and their net expected financial impact on claimants' branch accounts, compared to the total shortfall experienced by all claimants, which is of the order of £18.5 million. I have assessed both measures. The latter measure may be more useful , because if any set of bugs has expected financial impact much less than £18M, it can do little to account for the claimants' shortfalls, either collectively or individually.

45 I have assessed these measures from several different sources of information: (a) the defects in Horizon cited by the claimants from paragraph 5.4 onwards of Mr Coyne's expert report; (b) a survey of all KELs; (c) data on their shortfalls provided by the claimants in their schedules of information; and (d) other evidence cited by Mr Coyne. Summarised conclusions from each of these analyses follow. After that, I summarise the overall conclusion which follows from all of them taken together.

FUJ00083961

2.2.1 Analysis of Three Identified Errors

46 In their report, the claimants draw attention to three bugs in Horizon which are known to have had some effect on some branches' accounts. These are known as:

- receipts/payments mismatch
- suspense account bug
- Callendar Square, Falkirk bug

47 It appears that no claimant's branch accounts were affected by any of these bugs.

48 Because two of these bugs affected some branch accounts, they were subject to extensive analysis by Fujitsu. I have used these analyses to make my own analysis of (a) how well the robustness countermeasures acted, and (b) what would be the effect on claimant's accounts if other comparable bugs had occurred.

49 Because these bugs were each complex in their effects and were examined by Fujitsu in some detail, my full analysis is complex and is contained in an appendix.

50 I conclude that while these bugs evaded the automatic countermeasures built into Horizon, nevertheless they were detected by manual countermeasures and were handled effectively.

51 If the scale of the financial impact of these bugs is taken as a measure of the likely impact of similar bugs (if they had occurred) on claimants' accounts, then their financial impact would be minimal - of the order of 0.01% of the total shortfalls experienced by the claimants.

52 This follows from the known shortfalls caused by the bugs. In the case of the receipts/payment mismatch, its full effect across all 62 affected branches was of the order of £20,000. Because the claimants managed 560 branches out of 11,000, the expected impact of a similar bug on claimants' accounts is about 560/11,000 or 1/20 of this, or £1,000. This compares to the total shortfall of £18.5M experienced by all claimants. Therefore, the impact of a bug similar to the receipts/payment mismatch on all claimants' accounts amounts to one part in 20,000 of the total shortfall, which is 0.005%. Similar figures apply to the suspense account bug. It appears that the Callendar Square bug may have had no impact on branch accounts, but I am still investigating this.

2.2.2 Analysis of the Financial Impact of All Bugs in Horizon

53 There is witness statement evidence about the typical behaviour of SPMs in reporting anomalies in their accounts, and in trying to get them resolved. This evidence implies that any bug in Horizon with a large potential impact on branch accounts is very likely to be reported and investigated on most of the occasions when it occurs; while even a bug with much smaller financial impact will be reported on some occasions by some SPMs and will be investigated.

54 This implies that there is hardly such a thing as an 'unknown bug' in Horizon which affects branch accounts. Any bug with potential impact on branch accounts is highly likely to be reported and investigated - and because of its potential impact on branch accounts, it will be investigated with high priority.

FUJ00083961

55 I have examined the processes used by Fujitsu to investigate anomalies reported by SPMs or others, to diagnose their causes, to provide support to SPMs, and to fix any bugs revealed in this way. With certain exceptions discussed in this report, I have found those processes to be efficient.

56 If any anomaly is found that is likely to have an effect on branch accounts over a period of time, then it is highly likely that it will lead to a KEL which describes the problem and the advice to be given to SPMs. I have found the process for creating and maintaining these KELs to be efficient and effective. There has been a small core team at work, with very good knowledge of Horizon, who were able to recognise the sources of problems and ensure that appropriate guidance was available to SPMs in any period when a bug was being fixed.

57 For these reasons, the KELs are a good source of evidence about all bugs in Horizon with impact on branch accounts. It is possible to estimate the total impact of all bugs in Horizon on branch accounts, by searching the KELs and their associated Peaks for all evidence of such bugs and adding up their financial impact.

58 Because there are more than 8000 KELs, I have not been able to survey all of them. However, it is possible by automatic filtering to create enriched samples of those KELs most likely to have impact on branch accounts, and to examine a large proportion of those samples. One can then calculate the maximum possible impact of these KELs on branch accounts and correct the result for limitations of the sampling method.

59 To compute the total impact on claimants' branch accounts, one must then allow for the fact that claimants only held 561 out of more than 11,000 branches.

60 I have done this. So far, I have been able to examine 50 KELs, out of an enriched sample of 259 KELs, which I believe contain at least 50% of the KELs caused by any bug with impact on branch accounts.

61 The result of this analysis is that the total impact of all bugs in Horizon on claimants' branch accounts is expected to be at most of the order of £9,000 - compared with the total £18.5 million of shortfalls experienced by the claimants. Therefore, bugs in Horizon cannot account for a significant part of the shortfalls experienced by the claimants.

62 That conclusion is currently based on a sample of 50 KELs, as I stated above. I shall continue to examine more KELs to increase the sample size, and thus to improve the reliability of the conclusion. However, it is in my opinion already a very clear-cut and reliable conclusion.

2.2.3 Analysis of Data Provided by the Claimants

63 I have made a second analysis of the total financial impact of bugs on the claimants' accounts, which is not such a precise analysis as the one above and does not give such a stringent upper limit on financial impact as that analysis. However, it is based on completely independent evidence, and so acts as a useful backup for the analysis above.

64 The claimants have provided data on the individual shortfalls experienced by each one of them. Some summary figures obtained by aggregating these data are:

- Total number of claimants: 561
- Total number of months in service, for all claimants: 52,000

FUJ00083961

- Average duration of service per claimant: 92 months
- Total shortfalls experienced by all claimants: £18.5 million
- Average rate of shortfalls across all claimants: £360 per claimant per month in service.

65 I have used these data to test the hypothesis put forward by the claimants - that some large proportion of the shortfalls they experienced was due to bugs in Horizon, detected or undetected.

66 It is reasonable to suppose, as a first assumption, that Horizon bugs occur at random, and affect all claimants in all months in equal measure, apart from random fluctuations. Later in my report, I challenge that assumption, but the result summarised here does not change. I may also assume that, for any claimant, the total shortfall is the sum of two terms: one from bugs in Horizon, the other from human errors.

67 In that case, the claimants with the smallest average shortfall per month will be those with the lowest level of human errors in their branches. One can use those claimants to estimate an upper limit on the average impact per claimant per month of Horizon bugs - the first of the two terms.

68 I have made this estimate, using the 95 claimants with the smallest average shortfall per month. These claimants account for approximately 10,000 months, of the total of 52,000 months in service. If Horizon bugs occur at random in any month for any claimant, the level of random fluctuations in Horizon bugs over 10,000 claimant months is small, and the resulting estimate of the financial impact of Horizon bugs will be reliable.

69 These 95 claimants all have a net shortfall per month of less than £50, and the average shortfall across these claimants is £25 per month. If, as we expect, this is an upper limit on the level of impact of Horizon bugs (as all branches are expected to have some human errors), then it only accounts for 8% of the claimants' total shortfalls. This is only an upper limit - as one expects every claimant to have some non-zero level of human errors.

70 I have also done a similar analysis of aggregated claimant data in a time-dependent manner, to allow for the fact that the level of Horizon bugs may have fluctuated over time, as bugs were introduced or fixed. The results of this analysis are similar to those from the time-independent analysis, for all the three-year time periods.

2.2.4 Responses to Mr Coyne's Opinions

71 [Mostly TBD, but probably to include material along these lines updated in the light of the expert report]

72 In paragraph 1.4 of their outline, the claimants identify eight KELs which, they say 'caused discrepancies affecting branch accounts'.

73 I have analysed all these KELs and the associated Peaks. Contrary to what the claimants say, these KELs provide good evidence of the robustness countermeasures in Horizon in action, and there is no reason to believe that any of them had any permanent effect on branch accounts. Later in my report, I have identified the reasons why they would have no permanent effect.

2.2.5 Conclusions on Horizon Issue 1 from the Analyses Taken Together

74 I have made two independent analyses - from KELs and from claimant data - of the total impact of all Horizon bugs on claimants' branch accounts. These analyses each imply that bugs in Horizon can at most only account

FUJ00083961

for a small proportion - less than 8% - of the shortfalls experienced by the claimants. The analysis from KELs gives a much smaller figure, of less than 0.1%; but 8% could only be exceeded if both analyses are wrong.

75 While this analysis has calculated the likely size of the summed impact of Horizon bugs on all claimants, I also need to calculate its likely impact on any individual claimant. The main analysis implies that the impact on all claimants, spread over all their 52,000 months of tenure, was of the order of £10,000. In order to be conservative, I shall multiply this figure by a factor 10 - assuming (to give the claimants the maximum benefit of the doubt) that my analysis has somehow omitted 90% of the bugs with financial impact - or underestimated their financial impact by a factor 10. It might be possible, although I think it highly unlikely, that the claimants could establish this. Assume for the moment that they do.

76 The mean financial impact of bugs in Horizon on any one claimant in any one month of his tenure, is then £100,000/52,000 = £2. A mean loss per month of £2 from Horizon bugs can occur through bugs with higher financial impact, but only if they occur with low probability. For instance, a loss of £200 could occur, but only with probability one part in 100. Or a loss of £2000 could occur, with probability one part in 1,000. So a significant impact in any one month can only occur with very low proabability.

77 The mean loss per month of £2 from Horizon bugs is to be compared with the mean loss per month suffered by the claimants, of £260. Typically, claimants suffered larger losses in individual months. If, then, a claimant were to say: "In a particular month, I suffered a loss of £2000, and I assert that it was caused by a bug in Horizon", then (as above) the probability of that account being correct is only one part in 1,000. These conclusions are not altered by the considerations in Mr. Coyne's report, because he has nowhere addressed the net impact of bugs on branch accounts.

2.3 Reconciliation and Transaction Corrections

78 The third group of issues includes the Horizon issues 5 and 15, concerning the related topics of reconciliation with external parties, and Transaction Corrections (which often arise from reconciliation). These have been treated as a group because they belong so closely together. However, as they are an important part of the way Horizon is made robust against a variety of user errors, they relate to issue 3 in the first group. The issues are:

79 **Issue 5**: How, if at all, does the Horizon system itself compare transaction data recorded by Horizon against transaction data from sources outside of Horizon?

80 **Issue 15**: How did Horizon process and/or record Transaction Corrections?

81 These are both fairly descriptive issues, being 'how' questions and requiring a detailed 'how' answer. In bare summary, my opinion on issue 5 is:

- For every client organisation which uses PO branches as agents for its financial transactions, both the PO and the client organisation keep an electronic record of every transaction.
- These records are periodically compared, or reconciled, to check that they match, down to the level of individual transactions.

FUJ00083961

- This reconciliation process is done by a range of IT systems, some of which are part of Horizon, and some of which are not.
- Whenever a mismatch is discovered, the contract between PO and the client organisation defines how it is to be investigated and handled. [Should I mention this? Is it correct?]
- If it is decided that PO are responsible for the discrepancy (which happens, in the majority of cases because PO branches do manual processes, whereas client IT systems are generally fully automated) then PO makes up the difference, and this leads to an entry in PO accounts on POLSAP.

82 Our opinion on issue 15 (transaction corrections) can be summarised:

- If, following a new accounting entry in POLSAP resulting from a reconciliation discrepancy, PO believes that the branch was responsible for an error, a Transaction Correction (TC) is created on POLSAP.
- The TC is passed to BRDB but does not at that point affect the branch accounts.
- From BRDB, information about the TC is passed overnight to the branch, so the SPM sees it the next morning.
- If, at that point, the SPM accepts that the correction arose from an error in his branch, he accepts the TC, and it then affects his branch accounts. Usually this adjustment makes the accounts correct, putting an error right.
- If the SPM does not accept this, there are processes for investigating the possible cause of the TC.

83 The two issues relate to points raised in the claim and defence in the sense that, provided Horizon and the systems it interacts with perform reconciliation and transaction corrections accurately, and in a way that SPMs can understand, there is no detrimental impact on branch accounts; but if those functions are not accurate, there may be unintended effects on branch accounts.

84 My analysis of reconciliation and transaction corrections is that they are carried out accurately, and the results communicated correctly to SPMs; so that they are not responsible for shortfalls in branch accounts.

85 [We have seen overall statistics on the number of TCs - about 1 per branch per month. We need to find out what proportion of these TCs are contested by the SPM, for what reasons, how they are investigated, and what proportion of the contested TCs are resolved in favour of the SPM].

2.4 Facilities available to Subpostmasters

86 The third group of issues includes the Horizon issues 2, 9, and 14, because these all relate to the Horizon facilities available to subpostmasters when running their branches. The issues are:

87 **Issue 2**: Did the Horizon IT system itself alert Subpostmasters of such bugs, errors or defects as described in (1) above and if so how

88 **Issue 9**: At all material times, what transaction data and reporting functions (if any) were available through Horizon to Subpostmasters for:

a) identifying apparent or alleged discrepancies and shortfalls and/or the causes of the same; and

FUJ00083961

b) accessing and identifying transactions recorded on Horizon?

89 **Issue 14**: How (if at all) does the Horizon system and its functionality:

a) enable Subpostmasters to compare the stock and cash in a branch against the stock and cash indicated on Horizon?

b) enable or require Subpostmasters to decide how to deal with, dispute, accept or make good an alleged discrepancy by (i) providing his or her own personal funds or (ii) settling centrally?

c) record and reflect the consequence of raising a dispute on an alleged discrepancy, on Horizon Branch account data and, in particular:

i. does raising a dispute with the Helpline cause a block to be placed on the value of an alleged shortfall; and

ii. is that recorded on the Horizon system as a debt due to Post Office?

d) enable Subpostmasters to produce (i) Cash Account before 2005 and (ii) Branch Trading Statement after 2005?

e) enable or require Subpostmasters to continue to trade if they did not complete a Branch Trading Statement; and, if so, on what basis and with what consequences on the Horizon system?

90 These issues are addressed as a group because they all relate to facilities available to subpostmasters, so our opinions all follow from a description of those facilities.

91 These are all descriptive issues - 'how' and 'what' questions - so the answer to each question emerges from a detailed description of the Horizon facilities and cannot be easily summarised here.

92 They are related in several ways to allegations in the pleadings, and we summarise our opinions on those relations:

93 TBD.

2.5 Facilities available to Post Office

94 The fourth group of issues includes the Horizon issues 7, 8, 10, 11, 12, and 13, which all relate to facilities available to the Post Office centrally or to Fujitsu, rather than to subpostmasters. The issues are:

95 **Issue 7**: Were Post Office and/or Fujitsu able to access transaction data recorded by Horizon remotely (i.e. not from within a branch)?

96 **Issue 8**: What transaction data and reporting functions were available through Horizon to Post Office for identifying the occurrence of alleged shortfalls and the causes of alleged shortfalls in branches, including whether they were caused by bugs, errors and/or defects in the Horizon system?

97 **Issue 10**: Whether the Defendant and/or Fujitsu have had the ability/facility to: (i) insert, inject, edit or delete transaction data or data in branch accounts; (ii) implement fixes in Horizon that had the potential to affect transaction data or data in branch accounts; or (iii) rebuild branch transaction data:

a) at all;

FUJ00083961

b) without the knowledge of the Subpostmaster in question; and

c) without the consent of the Subpostmaster in question.

98 **Issue 11**: if they did, did the Horizon system have any permission controls upon the use of the above facility, and did the system maintain a log of such actions and such permission controls?

99 **Issue 12**: If the Defendant and/or Fujitsu did have such ability, how often was that used, if at all?

100 **Issue 13**: To what extent did use of any such facility have the potential to affect the reliability of Branches' accounting positions?

101 In issue 7, if the word 'access' is interpreted as 'access to read' (a common interpretation in IT), then the answer is trivial; both PO and Fujitsu regularly needed to access the data to read it. If 'access' is interpreted as 'change or update' the answer is also yes, but under tightly controlled conditions which will not be described in this summary. In our opinion, these conditions were so tightly controlled that adverse effects on the accounts of a branch were very unlikely.

102 Issues 8, 10 and 11 are descriptive questions with detailed answers (including nine parts, in the answer to issue 10), which will not be summarised here. Issue 12 is a factual question, and I have been advised that the ability was used on only one occasion, which we describe in section 11 of this report.

103 Issue 13 draws out the impact of issues 8 and 10 on branch accounts. In our opinion, any such effect was either zero or very small indeed and could only account for at most 0.01% of the discrepancies alleged by the claimants.

2.6 Mr Coyne's opinions

104 In summarising my response to Mr Coyne, I first point out four important limitations which I have found to permeate his report:

a) **Focus on Impact of Errors**: In his report, Mr Coyne draws attention to a number of errors and imperfections in the operation of Horizon over its 18-year history. As Horizon is a very large IT system, inevitably there have been errors. The key issue in Horizon Issue 1, is the extent to which those errors had impact on branch accounts. Mr Coyne's report fails to focus on the question of financial impact. Mr Coyne's report failis to provide this focus, and by indiscriminately citing issues irrespective of their financial impact.

b) **Understanding of Robustness**: Mr Coyne's conclusions on robustness are equivocal - sometimes finding Horizon robust, and at other times not. His conclusions partly reflect the agreement reached in the experts' joint memorandum (that robustness is not the same as perfection), while in other places, in contradiction to that agreement, he appears to equate robustness with perfection. However, more important than this - Mr Coyne has failed to understand or describe the fact that robustness is not some general hygiene factor, to be assessed as more or less good. It is a specific and established set of IT practices and countermeasures. It is possible to classify the types of these countermeasures, to assess how each type was applied in the building of Horizon, and to assess how well they have worked in practice. Mr Coyne's report does not do this.

FUJ00083961

c) **Analysis and Context**: In his report, Mr Coyne cites a large number of KELs, Peaks, and other reports. His citation of a KEL or a report is very brief - typically one or two paragraphs. These brief citations do not explain the context and meaning of each KEL or report. While some KEL or report may appear at first sight to be relevant to an issue, or by selective quotation appear to have a certain implication, this is impossible to assess from the very brief treatment, and lack of any deeper assessment, given by Mr Coyne. By creating uncertainty rather than resolving it, Mr Coyne does not assist the court.

d) **Linkage from Evidence to Conclusions**: Because Mr Coyne's treatment of each piece of evidence that he cites is so brief and superficial, when he states his conclusions, the linkage from evidence to conclusions is tenuous and difficult to understand. At best it can be said that the evidence cited tends to build up an impression, which loosely points in the direction of the conclusion. However, the impression may be misleading (as the evidence is not analysed) and the linkage is not explained; the conclusion does not follow from the evidence.

105 In responding to Mr Coyne's report, the limitations (c) and (d) have caused me particular problems. In order to comment on any piece of evidence cited by Mr Coyne (such as a KEL, a Peak or a report) in a way that will assist the court, I need in each case to provide a depth of analysis not provided by Mr. Coyne. I need to provide both context and analysis (for instance, in terms of financial impact or robustness countermeasures) which Mr Coyne has not provided.

106 In round terms, for each KEL or report cited by Mr Coyne, I would need to do three times as much work and write five times as much in my report. Because many of the KELs, Peaks and reports cited by Mr Coyne were not known to me before I saw his report, this has not been possible in the time available for preparation of my own report. This is not a problem I can use inexpert resources on.

107 The approach I have taken in this report is therefore:

- First, to ensure that my own opinions on each issue are stated as clearly and concisely as possible, with linkage to the evidence I cite.
- For each Horizon issue, to contrast my own opinions with Mr. Coyne's - pointing out where my opinions agree with or differ from his opinions or go beyond them.
- For the KELs and Peaks cited by Mr Coyne, to provide a preliminary analysis in an appendix in tabular form. This analysis will be converted to a more thorough analysis in my supplemental report. This, I expect, will lead to a fuller account, and possibly to revisions on individual KELs, but not to any substantive new opinions.
- For the reports cited by Mr Coyne, to illustrate by selected examples why a deeper analysis is required to assist the court. In my supplemental report, I will provide that deeper analysis.

FUJ00083961

3. BUSINESS APPLICATIONS IN HORIZON

3.1 Overview of Horizon Requirements

108 The functionality of Horizon is more than that of an accounting system, because Horizon also supports a large and increasing, number of business applications.

109 For every kind of activity which a customer might enter a post office branch to carry out (such as buying a book of stamps, or paying a bill, or renewing road fund tax, or withdrawing cash from an account) there needs to be functionality in Horizon, both to support the counter activity of carrying out the transaction, and for the back office activity of settling with the PO's 'client' organisation, who has provided some service to the customer - such as the DVLA, or a bank. Accounting is a thread running through all of these business requirements, but it is only a part of them.

110 The number of services provided by PO branches is large and has increased steadily from 1998 to the present day. The functionality of Horizon has expanded in line with the growth in service, both on the counter and in the back office.

3.2 The Point of Sale Application, and Customer Settlement

111 For part of its activities - such as selling stamps - a Post Office branch acts like a retail outlet, and it needs hardware and software to support this activity. This is the Electronic Point Of Sale Software (EPOSS) component of Horizon. EPOSS must allow the counter staff to record that some goods have been provided to a customer, compute the price of those goods, and allow the customer to pay the money required for all their purchased goods, for instance by cash or a credit card.

112 If a customer wants to carry out two or more different activities in one visit to the counter - for instance, to settle a bill and to buy some stamps - Horizon should not oblige the customer to settle the amount in two separate pieces. So, Horizon has the concept of a customer carrying out a 'basket' of activities and settling the total amount due for the basket in any way they wish - by one credit card transaction, by a cheque, by cash, or by a mixture of these.

113 However, baskets of PO activities and non-PO activities are not supported. If a customer wishes to buy a newspaper and some stamps, the newspaper is not sold by PO - it is sold by a separate retail outlet which uses the same premises. So, the customer has to settle in two parts. In this respect, the National Lottery is an exception and spans the two businesses, as will be described later.

114 So, Horizon needs to support retail-like activities (such as buying stamps) and agency-like activities (such as paying a bill) within a single customer basket, which may be settled by a compound set of payments.

3.3 Agency Activities

115 The Post Office refers to other organisations, for which it provides customer services in its branches, as its 'clients'. They include high street banks (for offering banking services), gas and electricity companies (for paying bills), DWP (for paying benefits and pensions) and DVLA (for paying road fund tax).

FUJ00083961

116 The Post Office currently has about 140 client organisations, which shows the diversity of services available in a branch. This also implies that, for most of these clients, the service provided through the Post Office will be different in nature from the service provided for other clients, so some unique software functionality must be provided both in the branch and the back office, to support the activities for that client. This is a part of what makes Horizon such a large and complex system.

117 It is not possible or useful in this report to describe all 140 types of service provided at PO counters, or the software needed to support them. We will only touch on a few services which either illustrate the diversity of services in a representative way or are important in this dispute.

118 A high-level classification of the services now offered in branches on an agency basis includes the following[1]:

- **Paying bills** to BT, utilities, local Government
- **Prepayment services** - DVLA savings stamps, gift vouchers and entertainment tickets
- **Acquiring licences** - local Government permits, television, motor vehicle
- **Money management** - banking deposits and cash withdrawals, savings and investments
- **Insurance services** - general, travel
- **Pensions payments** - e.g. for MoD
- **Lottery** - for Camelot

3.4 Requirements: Branch and Back Office

119 As well as the counter activities described above, Horizon also needs to support the periodic process of balancing and rollover for each branch. Every branch operates in Trading Periods (TP), which are either four or five weeks –according to a timetable published periodically by PO). At the start of each TP, the branch is supposed to be 'in balance'. This means that the physical stock and cash in the branch agrees with the data on stock and cash held in Horizon. Then, during the Trading Period, Horizon records all customer transactions made at the branch, so it records the changes in cash and stock. It also records any replenishments or remittances[2] of cash or stock in the branch. Thus, Horizon records all changes in cash and stock held at the branch during the TP, and can compute, from the starting amounts and the changes, the expected amounts of cash and stock at the end of the period.

120 At the end of each Trading Period, the subpostmaster (SPM) counts the physical cash and stock in the branch and compares it with Horizon's expectations of the same values. This is called 'balancing'. If the numbers are all equal, the branch is in balance and can 'roll over' to the next period. If the two sets of numbers are not equal, this implies that some of the transactions entered into Horizon during the Trading Period were erroneous or had failed to be entered. For instance, if the counted stock of stamps is less than the expectation from Horizon, this implies that some stamps were given away or lost, without recording a transaction on Horizon. Because it is

1 This list, mainly taken from a 2003 document, will need to be updated.

2 At PO, 'remittance' is often abbreviated to 'remming'. This means sending surplus cash from a branch to the centre, or replenishing cash or stock in a branch from the centre.

FUJ00083961

assumed that this arose through some error by the SPM or his staff, the SPM is required to take responsibility for the discrepancy, in some way he or she chooses - for instance, by paying in cash to cover the discrepancy; or by putting the amount in a local suspense account, to be resolved or paid later[3]. Then the branch is again in balance, and can roll over to start the next TP.

121 To support this process, at the end of each TP, Horizon is required to provide the figures of estimated cash and stock; and if the SPM finds any discrepancy, to enable them to record how the discrepancy will be resolved; and when this has been done, to allow the branch to roll over and start the next TP.

122 Horizon must also support the activities of replenishing stock such as stamps, and of replenishing or remitting cash.

123 It must also support other administrative activities in the branch, such as enrolling new staff and enabling them to use the counter system.

124 The back-office settlement activity of Horizon may be illustrated in the case of a single client organisation, the DVLA. Across the UK in any day, the PO accepts a large amount of money from customers paying their road fund tax. All this money needs to be paid to DVLA. Therefore, PO has a back-office activity - carried out centrally - of summing all these amounts of money and paying DVLA. DVLA knows how much money it expects to receive in this way and checks the amount it expects against the amount calculated by PO. This cross-check is called reconciliation and supporting it and reflecting its outcomes are central to Horizon. Some kinds of reconciliation cannot be done as often as daily, because of variable time lags in the information available to clients.

3 The facility for local suspense accounts has not always been available to SPMRs. It was removed for a period of time. We need PO to tell us when.

FUJ00083961

4. ORIGINAL HORIZON (1998 - 2010)

125 It is important to appreciate of the level of complexity of the Horizon requirements, and of the Horizon IT systems built to meet them. In a document ' Fujitsu's Systems and Operational Services to UK Post Office and the Worldwide Trend of Post Offices' Fujitsu have described Horizon as 'Europe's largest non-military IT contract', so Horizon is at the high end of complexity amongst IT systems. It represents many thousands of man-years effort in development and testing, and its documentation alone is of the order of 80,000 documents. Inevitably, the court will only have the time within the course of the trial to understand limited and selected aspects of Horizon. The foundation sections of our report are intended to ensure that the readers understand those aspects of Horizon that most need to be understood to address the Horizon issues.

126 On the other hand, compared with the IT estates of various large organisations we have worked for (such as the NHS, Barclays Bank, UBS or RBS), the Horizon system is probably no more complex, and in some ways less complex. The banks' IT estates, like Horizon, have a complex corporate back end and an extensive branch office network. They were developed over a longer time period (30-40 years) using development team sizes similar to or larger than Horizon, often merging together or integrating the IT systems of previously independent organisations. This gave them a degree of legacy complexity, and design compromise, and corporate amnesia, not found in Horizon. There are parts of these IT estates which 'nobody dares touch'. The same is not the case with Horizon.

127 From time to time, when describing aspects of the Horizon architecture in the next three sections, I shall refer to various robustness countermeasures, which have been introduced in a table in section 2 of this report. This will help to describe the countermeasures by illustration, and shows where they are built into the Horizon architecture. In the table, the countermeasures have each been given a three letter acronym such as RDS (Redundant Data Storage). It may be worth having a printed copy of the table to hand when reading these sections, to see both the acronym and the summary description of the countermeasure.

4.1 The Four-Level Architecture

128 In what follows, the words 'level', 'layer' and 'tier' all have the same meaning.

129 Nearly all complex IT applications are designed in levels or 'layers', to isolate different kinds of complexity in different layers, and to reduce the possibility of unwanted interactions between functions in different layers. A typical 'client server' layering structure includes at least a user interface layer, a business logic layer, and a data layer. The layering in Horizon is more complex than this.

130 The main purpose of defining an architecture in layers is to separate the functionality into parts in the different layers, with well-defined and simple interfaces between the layers. This not only makes each layer easier to design, build and test; but also, if there are errors not found in testing, it makes it easier to understand and isolate the cause of the errors by inspecting the exchanges between layers. Thus a layered architecture is an important countermeasure for robustness; I have denoted it by the acronym 'ARC'.

131 The architecture of Horizon up to 2002 is described in the document TD/ARC/001 'Technical Environment Description' which is 476 pages long. This document states: *'The system architecture adopted to meet these requirements is*

FUJ00083961

not based on conventional client-server models. Nor does it conform to traditional central-system models. It adopts an entirely original and highly innovative four-tier model that effectively merges the qualities of central systems and client server systems.'

132 This architecture is explained in a diagram, which appears to have five layers rather than four. If the two boxes of 'Agents' and 'Correspondence' are counted as one 'Agents' layer, we get four layers, of:

- Counter
- Agent
- Host
- External Interface

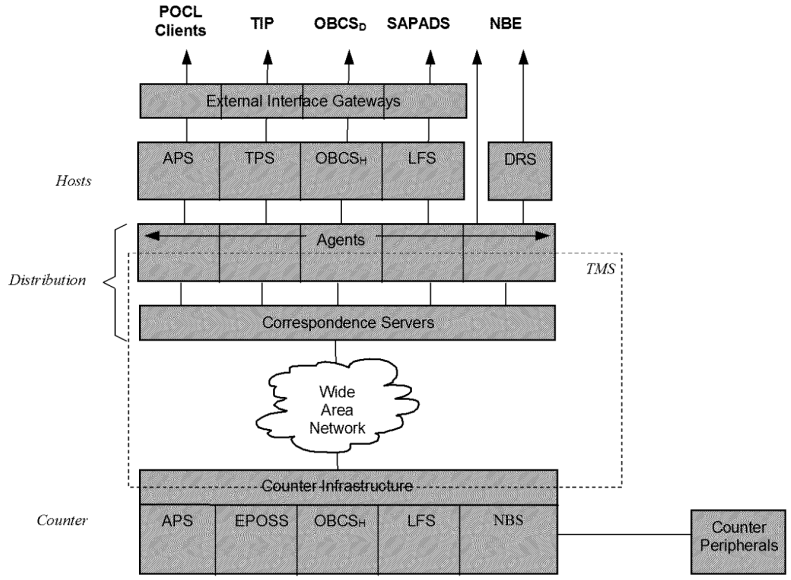


Figure 4.1 - Horizon layered architecture

133 This apparently simple diagram hides a huge amount of complexity in Horizon, and that complexity can only be described in stages.

134 The counter layer consists of all hardware and software in the branch. It includes all hardware and software required to support the counter activities required for all products and customer services offered in the branch. It will be described in the next sub-section 4.2, which will define the acronyms in the Counter layer of the diagram. In this section we will note one aspect of the counter layer: that it was largely built based on a commercial product, Riposte from Escher.

135 Riposte provided much of the Graphical User Interface (the basis of all user input and output at the counter) and provided a mechanism for secure distribution of messages between the branches and the two back-office campuses, which were located at Bootle and Wigan. This message distribution passed through the Wide Area Network in the diagram.

136 The Correspondence Servers handled communication over the network.

FUJ00083961

137 The function of the Agent layer was to provide two-way translation of data between the formats used in the counter layer and the network (these formats were described by Attribute Grammars) and the formats used in the Host layer.

138 An Attribute Grammar is a way of describing a tree-like message structure in terms of its parts and their sub-parts. In more recent IT systems, tree-like messages are usually sent in XML (Extensible Message Language), with their structure defined in a notation called XML Schema. This is used in parts of HNG. Because the first Horizon was developed before the use of XML became widespread, Attribute Grammars fulfilled this function in Horizon. I believe this is because the Escher Riposte product worked in this way at the time.

139 As well as reliable communication, Riposte provided a facility for reliable replication of data between the branches and the back-offices campuses. This means that if certain types of data were created at the branches, Riposte guaranteed that the same data would be available on the campuses - although if the underlying network was unreliable, it might take some time for Riposte to deliver this guarantee. Replication guaranteed that despite any network failures, no change to data made at a branch would be omitted at the campus or made more than once at the campus.

140 The bulk of the back-office functionality was provided in the Host layer, which will be described in section 4.3. Host applications were and are typically batch systems, processing data in large batches on a daily basis. A complex daily batch schedule was used to control the sequence and timing of these batch processes, using the Maestro scheduling product. The acronyms in the Host layer of the diagram above will be described in that section. It was the Host layer (and for most purposes, only the Host layer) which communicated with the IT systems of PO client organisations, through the External Interface Gateways.

141 There is an important simplification in the four-tier architecture. Each different business application in Horizon (typically tied to a different PO client organisation) can be regarded as a vertical 'slice' though the diagram and is largely independent of the other slices. It is intuitively obvious that different business applications (such as DWP Pensions, and Camelot Lottery) need have very little to do with one another (apart from being able to settle customer payments in the same basket - a facility provided separately from the applications). Therefore, the apparent complexity of some large Horizon architecture diagrams can be largely ignored when considering a single business application. This is another example of robustness through architecture (ARC).

4.2 Hardware and Software in the Branches

142 Although the hardware in the branches was not always reliable, and communications in particular were not highly reliable, there were strong measures built into Horizon to ensure that hardware failures and communication failures could not adversely affect branch accounts. These measures are described in section 6. They make up the robustness countermeasures of reliable hardware (RHW) and robust data communications (ROC).I shall therefore not spend much time here describing the hardware aspects of Horizon, either in the branches or the back-office campuses.

143 In the original Horizon architecture, sufficient data was held persistently in the branches, that a branch could continue to trade, and could support most business applications, even if the wide-area network was unavailable.

FUJ00083961

Whenever the network became available again, Riposte data replication would ensure that the required data became available to the back-office systems. The only applications which could not run in this way were those that required some immediate validation from a client organisation - for instance, withdrawing cash from a bank account. Therefore, a branch was able to hold all the data resulting from a day's trading and more.

144 As will be described in section 5, with HNG this was no longer the case. Persistent data was all stored remotely in the branch database - so that without a working network, a branch could no longer trade. More reliable network infrastructure by 2010 had made this a viable approach.

145 As well as supporting the business applications described in section 3, the software in the branches needs to support:

- Local user management
- Stock management
- Cash drawer management
- Balancing and reconciliation
- The production of local reports.

146 There had to be sufficient locally-stored data to support all these processes. To keep the counter clerk's view of all these applications consistent and simple, the user interface for all these local applications was provided by the Riposte desktop.

147 To describe how the branch layer of business applications was built on Riposte would involve a lot of technical complexity, most of which would not go to understanding the issues in the trial. I shall instead pick out some aspects which are relevant to the Horizon issues:

- **Zero-sum baskets for customers**: Whatever applications were invoked to serve a customer, the net impact of all the services provided for one customer was a sum of money which the customer was required to settle. It was required that the cash or other money produced by the customer should exactly match the cost of services provided; therefore, the whole basket of services and customer settlement had to be zero-sum, before the basket could be recorded in the branch (and then, through Riposte replication, later recorded in the back-office systems). This was a necessary requirement on all business applications, because the impact of every business application would need at some stage (typically overnight) to be fed into PO's accounting systems, which operated by double entry bookkeeping. The only way to put postings into the accounting system was by double entry, which in turn could only be done for zero-sum baskets. This is the robustness countermeasure of double entry accounting (DEA)
- **Other branch actions had to be zero-sum**: Any other actions performed in the branches which had an impact on PO accounts (including stock management, cash drawer management, balancing and reconciliation) could only be carried out in the branch in packages of updates which were zero-sum, when summed across different PO account codes. This had to be the case, because the only way that the results

FUJ00083961

could be posted to the accounts was to respect double entry bookkeeping - which is zero-sum across the accounts. This is another instance of the countermeasure DEA.

- **Transactional integrity**: All branch applications (including all customer business applications, balancing and reconciliation, cash management and stock management) were built so that any zero-sum package of updates from those applications would either succeed completely, or would fail completely and have no impact. This transactional integrity was enforced by the Riposte infrastructure, and I denote it by the robustness countermeasure with acronym TIN. Therefore, it was impossible in any event (such as hardware failure) for a part-completed set of updates to be recorded in the branch and then replicated to the back-office systems. This was necessary to prevent the accounting system from being subjected to non-zero sum updates, which would violate its double entry basis and cause later failures of its trial balances.
  The only exception to this principle was the so-called 'recoverable transactions' - where some irreversible interaction with a PO client system took place part way through a transaction - so it could not be undone in the case of a later failure. In these cases, the user on the counter would be guided through a short set of recovery steps, to produce a consistent zero-sum result which reflected what had happened. It was, of course, possible for the user to make some mistake in these steps, which may have been unfamiliar. In these cases, the mistake would be detected later by a reconciliation process, which would typically lead to a transaction correction. This robustness measure was a correction of user errors (UEC).
- **Applications driven by reference data**: Many of the business applications were not coded individually but were coded as generic applications which could be configured to run different specific applications by altering reference data. These were referred to in Horizon as 'soft-centred' applications. They had considerable benefits of adaptability and reliability over hard-coded applications (of which there were still a few). New applications can be built and deployed simply by providing reference data, rather than code. Errors could often be corrected rapidly, by simply correcting a piece of reference data. Reference data is much more concise and understandable than code, so it is much easier to create it or detect errors in it. Finally, any errors in the underlying generic code would affect a set of specific applications, and so be easy to detect. This was the robustness measure of data driven software (DDS)

4.3 Back-End Architecture

148 The three layers of the architecture which resided in the campuses at Wigan and Bootle were the Agent layer (which included the Correspondence layer), the Host layer, and the External Interface layer.

149 As has been described above, the role of the Agent layer was to manage communications and translate data between the representation used in the branches and the network on Riposte, and the representations used in the Host layer.

150 The main design document on the first Horizon[4] says:

'The systems at the Host Layer can provide permanent storage for information if required by the application's business rules. The Host systems can accept data from external Clients, and translate a file-based view of this information into discrete transactions or

4 TD/ARC/001

FUJ00083961

"messages". These are then passed to the Counters via the Agent and Correspondence Layers. Similarly, messages received from the Counters are translated back into a file-based view for transmission to the external Clients.'

151 Another description in the same document says:

'[Host systems] Servers run mainly large background batch processes and represent the part of the architecture that is responsible for the following functions.

- *Manipulating the information received from the External Client Systems into a form that is appropriate for the presentation mechanism and vice versa*
- *Applying business rules that are relevant to that information*
- *Storing non-transient information within the "Data Storage" component. This includes metrics needed for the computation of SLAs that may modify the payments due from PO Ltd for the achievement of key deliverables.*
- *Manipulating any such stored information'*

152 These descriptions only begin to describe the range of functions in the Host layer; to do more, I need to look at specific IT systems in that layer, aligned with different business streams. As will be described in section 5, many of these IT systems did not change with the introduction of HNG in 2010.

153 One fairly simple diagram, which shows a number of important components of the back-office systems, is the following:

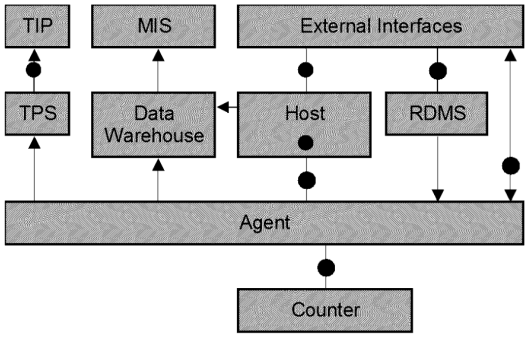


Figure 4.2 - Application components

154 It is worth briefly describing those elements of this diagram which have not already been described as part of the four layers introduced in this section:

- **RDMS** stands for Reference Data Management System. As was described above, many business applications in the branches are driven by reference data, and this approach has many advantages over hard-coding of all the different business applications. It is much more flexible, to manage changes over time and across branches; and it is more reliable. However, this approach implies that the reliability of Horizon depends on the reliability of the reference data (much of which, for instance, is maintained by PO staff rather than by Fujitsu IT staff). Therefore, a dedicated IT application is needed to manage the reference data, and to distribute it appropriately to branches. This is needed for the robustness measure of data-driven software (DDS)

FUJ00083961

- The **Data Warehouse** consists of one or more databases, whose structure is designed to support flexible and open-ended querying and reporting by PO business staff, to help them understand the whole state of PO business from day to day. Many different kinds of information which pass through the host systems are siphoned off into the data warehouse and stored there in data structures designed for querying and reporting. Functionality which depends on a data warehouse includes MIS (described next) and other applications such as 'data mining' to look for unanticipated trends and correlations in data. The data warehouse contributes to two robustness measures: redundant storage and computing (RDS), and Manual Inspection of Data (MID)
- **MIS** stands for Management Information System, the component built on the data warehouse to provide PO staff with the flexible access to information about all aspects of PO business. The data warehouse and the MIS are an important part of the checks built into Horizon. In cases of human error in business processes, operational errors in managing PO business on Horizon, or software errors in Horizon, some resulting discrepancy or aberration will be rapidly visible through the MIS. Many pairs of eyes are inspecting the outputs of the MIS, in hundreds of different reports or spreadsheets. One purpose of this is to ensure the rapid detection and correction of many types of errors. These include software errors. So the MIS also contributed to the robustness measures of RDS and MID.
- **TPS** stands for Transaction Processing System. The purpose of TPS is to 'harvest' all types of transaction taking place in the branches, and to pass them on to other IT systems in the Post Office - initially to TIP, and later to POL FS.
- **TIP** stands for Transaction Information Processing, which in 2003 (the date of the diagram above) was the gateway to all other PO data processing, including accounting. After 2004, PO accounts were held on a SAP system, POL FS; so TPS passed data to POL FS, rather than to TIP.
- The black circles denote points *'at which ownership of data conceptually changes and hence at which audit information is generated*'. Audit is addressed in section 4.4 below, and is part of the robustness measure SEK (secure kernel)

155 To quote the 2003 design document: '*One essential task that can only be carried out at the Host layer is reconciliation. The Host is the only system component that can detect discrepancies between the transactions carried out at the Counter (and hence reported back to PO Ltd via TPS), and those that were authorised or expected. It should be in a position to send reconciliation reports back to its Client. These enable the discrepancy with the TPS records to be identified and resolved.*'

156 This reconciliation, carried out in the Host layer, is an essential element within Horizon for detecting and correcting errors made at the counter (robustness measure UEC). Reconciliation and Transaction Corrections (TCs) are described for both Horizon and HNG in section 6.

157 Reconciliation and TCs, which are primarily intended to correct human errors, also have the effect of detecting and correcting the effects of many possible software errors. If there were any such software error, it would probably occur with such high frequency, and occur uniformly across all branches, giving rise to so many TCs, that the PO would soon suspect a software error (for instance, seeing the effect repeatedly in some MIS report) and require Fujitsu to correct it.

FUJ00083961

158 The likelihood of any software error in Horizon staying disguised as a human error, and thus of not being detected, is extremely small; and even if it were not detected, by the argument above, it would have no distorting effects on branch accounts.

159 The PO has approximately 140 different client organisations, and so there are up to 140 different types of reconciliation which may be carried out. In my opinion, it would be extremely unlikely for any large client organisation to appoint the PO as agents for any other kind of financial transaction such as bill paying, without requiring a reconciliation check that PO was paying them the correct amounts of money. Such a lack of due diligence by the client organisation would not protect the interests of their shareholders or other stakeholders. So, this kind of error detection and correction is used for the vast majority of money that passes through PO branches - for all of its agency business. It combines the robustness measures of Redundant data storage (RDS) and user error correction (UEC).

160 Host applications fall into one of three classes:

- Complex applications that require a large amount of persistent storage, with high volumes and/or high transaction rates. These generally have their own Oracle database and are located on one of the Host Central Servers. They incorporate the robustness measure of transactional integrity and database recovery (TIN).
- Less complex applications, with little persistent storage requirement. These may run on the Host Central Server, or on a Host Ancillary Server, an Intel Platform running under Windows NT Server. Oracle or Microsoft SQL Server can be used to provide the database functionality and storage mechanisms. They still benefit from the TIN robustness measure.
- Simple applications that have no requirement for a persistent database may be implemented on a dedicated Intel-based Host Ancillary Server running under Windows NT server. Typically, these generate or process tabular files of text and numbers.

161 I shall mainly consider the first type of host application, which are responsible for the great majority of the money passing through PO branches.

162 Other system diagrams of the Host layer are complex, showing many distinct systems, and there is no 'universal' diagram which is suitable for explaining every issue.

4.4 Audit Information

163 The Horizon system includes an audit database, which is an accurate and immutable record of any activity on any counter which can affect the branch accounts. In the event of any discrepancy arising anywhere in Horizon (for instance, due to a bug in some other Horizon application, or some operational error in running a batch process, or a dispute about what data was entered at the counter) it is possible to compare other records - for instance, records extracted from other applications, or the data warehouse - with the audit records, which are guaranteed to be an accurate record of what was entered into Horizon at the counter. In this way many kinds of error can be traced and corrected. Audit records are normally retained for seven years, as required by the PO

FUJ00083961

contract with SPMs. The audit database is a robustness measure of a secure kernel (SEK) which also involves redundant data storage (RDS).

164 It is important to understand the many measures used to ensure the integrity of the audit data. A slide set produced by Fujitsu describes this well, and I will summarise the main points here. This can be done through following the sequence of operations by which data travels from the counter to the audit database.

165 In Horizon pre-2010, all data travels from the counter through the software application at the branch, through Riposte data replication to the two campuses, then through the Audit Agent to the Audit Store. This is shown in the diagram:

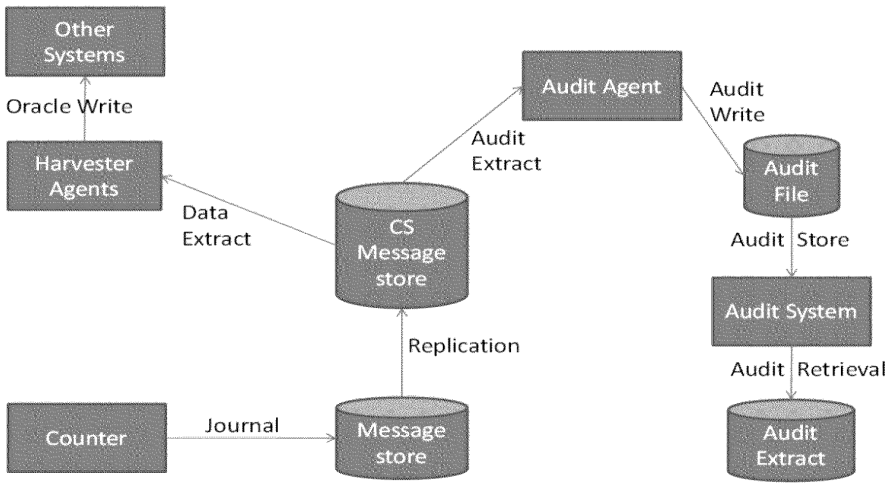


Figure 4.3 - Horizon audit data flow

166 In HNG, all data travels from the counter through the software application at the branch, through communications hardware and software to the Branch Access Layer (BAL), into the Branch Database (BRDB) and then nightly to the audit store.

FUJ00083961

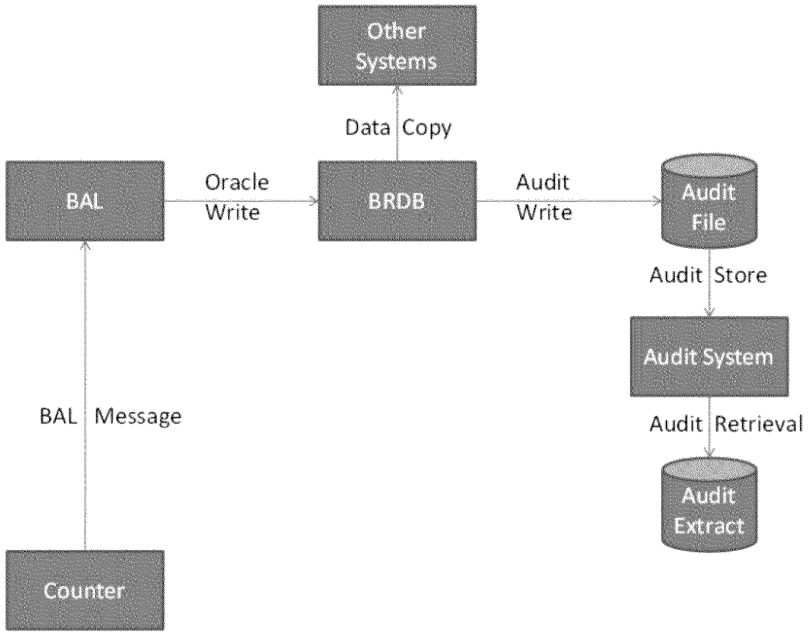


Figure 6.4 - HNG audit data flow

167 The principles ensuring the integrity of the audit data are the same in both cases of Horizon and HNG:

- When a user signs on at a counter, the password he or she provides is used to create cryptographic keys, which are used to encrypt all messages sent over the network to the BRDB, and subsequently used to create digital signatures on the audit records. Thus, any audit record is digitally signed in a way that proves it could only have originated from a certain counter, and that it has not been modified since it left that counter.
- As the counter clerk provides one or more services to a customer, these services and the money paid for them in settlement by the customer are collected in a basket whose monetary sum must be zero. At all stages on the journey of this basket to the audit record, there are checks that it has a zero sum. This is typically not just a check that two numbers are equal and of opposite sign; it is a check that several numbers add up to zero. Thus, any failure in hardware or software, which affects one or more of the numbers, is most likely to destroy the zero sum; if the zero sum survives and the record is stored in the audit database, all the numbers in it are an accurate record of what happened at the counter.
- All baskets are given a journal sequence number (JSN) which must ascend in increments of 1, with no gaps or duplicates. This ensures that no gaps or duplications are introduced in the baskets from any counter, for instance by communications failures or recovery processes. No extra baskets can be introduced without destroying the sequence. All audit entries are time-stamped.

FUJ00083961

- In communication, data replication, and in storage in any database, principles of transactional integrity are applied. This means that a basket is either stored in its entirety, or no part of it is stored. If it is not stored, appropriate information is sent to the branch, and recovery processes initiated.
- Digital signatures and seals such as MD5 hashes[5] on the audit records ensure they cannot be altered once they have been created.
- Recovery procedures are designed so that should any of these checks fail (e.g. in the event of a hardware failure at the counter), appropriate remedial steps are taken, and the integrity of the audit is preserved.

168 In my opinion, these integrity measures are well designed.

4.5 Changes During the Period 2000 - 2009

169 A series of significant changes were made in Horizon during the period 2000 – 2009. Each new application typically required changes at the branch and at the campuses. Some changes were superseded by later ones. Some important changes were:

- In 2003 DRS was introduced.
- POL FS came in around 2004.
- In 2005, Pension & Allowance Order Books were replaced by the Post Office Card Account which necessitated building banking services into Horizon. PO had always had business relationships with banks including Girobank.
- AP/ADC was introduced around 2007/2008.
- Around 2010, POL FS and SAP ADS were merged to make POL SAP.

5 An MD5 hash is a short string, computed from the entire contents of a file, which will change if the contents of the file are changed in any way.

FUJ00083961

5. HORIZON NEW GENERATION (2010 - PRESENT)

5.1 Motivation for the Move to HNG

170 Horizon moved from the previous Riposte-based architecture to Horizon New Generation (HNG) in 2010. At this time, there was no sudden change in the range of business applications supported by Horizon in the branches. This range of applications has increased continually over the lifetime of Horizon and HNG.

171 There were several motivations for the change from Horizon to HNG. The main driver was to exploit advances in the underlying communication technology, and improvements in its reliability - which meant that it had become possible to store all persistent data at the centre rather than in a branch, with the consequence that a branch could only operate when communications were available - but the risk of failed communications was by then so low as to be acceptable. This change mirrored the wider changes across the IT industry, where increased reliability of communications means that applications can now be 'cloud-based' (entirely dependent on remote data, stored by some cloud provider such as Amazon; and dependent on remote functionality in the cloud) and therefore simpler to deploy and manage.

172 The centralised storage of transaction data allowed several changes and improvements:

- A simplification and rationalisation of the architecture in many respects (just as most cloud-based applications are now simpler than their antecedents);
- Simpler management of the branches in the event of hardware failures or replacements and other events, because in those cases branch data would not be lost and did not need to be recovered;
- No dependence on Riposte data replication, which meant that Riposte could be removed entirely, and all applications could be supported by more modern software technology.

173 These, rather than any change in the business applications to be supported, were the motivations for the move from Horizon to HNG.

174 The document 'Counter Business Architecture' (ARC/APP/ARC/0009) states that:

'The objective of the HNG-X programme is to develop a system with structural and operational characteristics that substantially reduce ongoing support and maintenance costs with respect to the current Horizon system.

The overall requirement is that the business capabilities offered by the current system (Horizon) are preserved in the new system (HNG-X). However, a limited number of business capabilities will be revised based on a joint optimisation of business requirements and system properties.'

5.2 The New Division Between Branches and the Back End

175 The fundamental change was that in HNG, no transaction data was held in any persistent form in the branches. The Counter Business Architecture document explains the rationale for this:

'The analysis of the serviceability profile for Horizon has highlighted data management as one of the most significant drivers for cost. The storage of transactional data within counters causes the need for security mechanisms that impact both the structural complexity and the operational performance of the Counter Business Application. In addition, the presence of sensitive data on the counter increases the time, complexity, and ultimately the cost of maintenance procedures.'

FUJ00083961

176 In Horizon, on completion of a basket of customer services, that basket was held locally in the branch in the Riposte message store - until Riposte could replicate it to the campuses at Bootle and Wigan, which might have been hours or days later, depending on the state of communications. Meanwhile, the branch could continue to function for many types of transaction. However, the number of applications such as bank withdrawals which required immediate confirmation from a third party, and therefore could not function in the absence of communications, had steadily increased.

177 In HNG, before completion of a basket of customer services, that basket was transmitted and had to be acknowledged by the Branch Database (BRDB). The basket could not complete successfully at the counter until that had happened - so Horizon could not operate in the branches without working communications.

178 Because the branch was no longer responsible for persistent storage or replication of transaction data, the architecture within the branches was simplified.

179 The main difference at the back end was the existence of the Branch Database, which was the main persistent store of all transactions for all branches. Many business applications in the back end were unchanged (and were referred to as 'legacy'[6]), except for the need for them to interface with the BRDB rather than with the previous Agent layer. Other copies of transaction data continued to be stored in those applications.

5.3 New Architecture in the Branches

180 The previous branch architecture had been based on Riposte, which provided functionality on many levels (including for instance user interfaces, some business applications, and message storage and replication). In HNG, Riposte was completely removed; therefore, all elements of the branch software, as well as hardware, were replaced.

181 Whereas much of the Horizon branch code had been written in Visual Basic, for HNG nearly all the branch software was written in Java - a newer language with good support for modern programming paradigms such as object orientation and service-oriented architecture. This allowed a more modern and elegant software architecture in the branch, which did not have to be fitted around the architecture of Riposte.

182 The many modern object-oriented features of the Java language provided better support for several robustness features, such as defensive programming (DEP).

183 A view of this architecture is shown in the following diagram:

6 In IT, the term 'legacy' is used to refer to older technology, which may have been superseded. For Horizon, this means the original generation before Horizon Online was implemented in 2010.

FUJ00083961

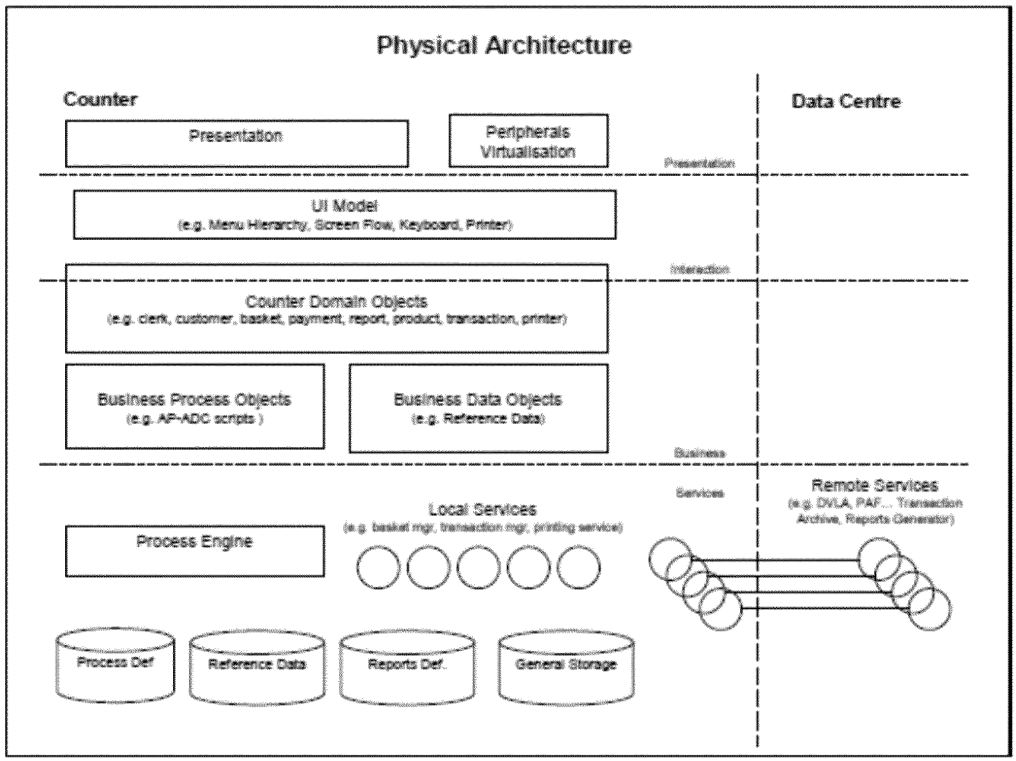


Figure 5.1 - Counter application architecture

184 The top 'Presentation' layer is responsible for displaying information to the users and for accepting their inputs. The next 'Interaction' layer provides the building blocks for this interaction, such as menus. The effect of these two layers is to provide a user interface similar in style to that which had been provided by Riposte in Horizon - to make the user experience similar to what it was in Horizon, but using Java technology, rather than Riposte, to build it. These two layers were largely responsible for the early detection of user errors (DUE).

185 As had been the case for the previous Horizon, the layered architecture of the counter software in HNG, with a clean separation between layers, was an important means of providing architectural robustness (ARC).

186 The 'Business' layer provides the functionality of the many business applications, in an object-oriented fashion. This means that there are several general-purpose software objects (i.e. modular blocks of software) with names such as customer, basket and payment, which represent the required behaviour of those entities in the real world, in a way that can be easily reused in many different business applications. The reuse of core design elements for many applications was another example of architectural robustness (ARC)

187 The Business Process objects and Business Data objects are more specialised to support the many business applications. As their names imply, the Business Process objects support the sequence of steps which make up a business process, and the Business Data objects hold the necessary data, which is presented at the counter or stored. However, this is generally not done by writing completely different software for each business application (i.e. for each type of service that can be offered to a customer). Many applications are driven by reference data, such as data which defines the sequence of steps in completing each type of service for a

FUJ00083961

customer. This reference data-driven style of software is common modern practice and is effective in making software easier to write and test. New applications can frequently be supported just by adding new reference data, rather than by writing new software. This was intended to achieve robustness through the use of generic, data driven software (DDS)

188 For instance, all automated payment (AP) applications are provided in this reference data-driven manner. This makes it very easy to build and test a new AP application, for a new client organisation.

189 The Counter Business Architecture summarises the capabilities in the business layer.

'In summary the set that are provided by the Counter Business Application are:

- *Point of Sale Capability;*
- *In / Out Payment Capability;*
- *APOP Facility;*
- *Banking Capability;*
- *DVLA Licensing Capability;*
- *Electronic Top-Up Capability;*
- *Bureau de Change Capability;*
- *Postal Services Capability;*
- *Generic Online Capability;*
- *Payment Management Capability (Cash, Cheque, Vouchers, Debit or Credit Cards);*
- *Cash and Stock Management Capability;*
- *Branch Management Capability (Stock Unit Balancing, Branch accounting, Branch Reports, Reversals and Refunds, Transaction Corrections);*
- *Branch Administration Facility (User Log On / Off, User / Password Management, Stock Unit Creation / Allocation, Provision of Secure Inactivity Time-Out Facilities, Generic User Help System),*
- *Branch Support Facility (Sales Prompts, Bulk Input of transactions, Reference Data, PAF, Message Handling, Audit and Training).'*

190 Just as the Presentation layer does, the Services layer provides a set of software objects which provide services in support of many business applications. Most of these services are not to do with the user interface but help in organising information and sending it for storage in the BRDB.

191 For instance, the facilities for double entry bookkeeping (ensuring that each basket is zero-sum before it is sent) and transactional integrity (ensuring that a basket is either sent and stored in its entirety, or none of it is stored at all) are provided generically in the services layer, and so do not need to be coded individually in the business objects. This design practice helps to ensure that the powerful checks of transactional integrity and double entry

FUJ00083961

bookkeeping (robustness measures TIN and DEA) are applied universally, and do not have to be built individually into any new business application.

192 One key component of the services layer is the Process Engine. This provides a simplified way for the counter to provide services which involve a sequence of steps. The sequences of steps need not be defined in Java code but are defined in a specialised Process Definition Language (PDL), which is executed by the Process Engine. PDL was developed for HNG by Fujitsu. The use of PDL means that complex sequences of steps are much simpler to define and test. This is another example of generic data-driven software (DDS)

193 The disc-shaped boxes in the services layer in the diagram above show that some data are stored persistently on the branch hardware; however, these data do not include customer transaction information. They include business process definitions (definitions of sequences of steps in a process), other reference data, data defining reports that can be output in a branch, and other information required to support operations. The reference data is refreshed daily from the data centre. There are services which provide these data to the other layers in forms that are convenient for them to use.

194 As can be seen from the diagram, the Services layer of the branch architecture is the only layer which communicates with the data centre, through the communications subsystem. Individual services provide reliable and robust communication for various types of information.(this is the robustness measure ROC) The purpose, as always, of this layered approach is to provide each kind of functionality (such as reliable and robust communication with the data centre) in one layer only, and not have to reinvent it for many different business applications. In effect, the services layer in HNG now provides many of the services which were formerly provided by Riposte.

195 The services layer also provides interfaces for online services, where to provide some service at the counter, it is necessary to contact some non-PO IT system. These online services include:

- Banking
- Credit / debit cards
- Mobile phone E-Top Ups
- DVLA online
- APOP services, such as postal orders
- PAF lookup
- Generic Online Services
- Some types of PINPad accesses (WSPOS[7]).

5.4 Back-End Architecture: Changed and Unchanged Elements

196 The two completely new elements of the HNG back end are the Branch Access Layer and the Branch Database.

7 Web Services POS (Point of Sale)

FUJ00083961

197 The principal function of the Branch Access Layer (BAL) is to exchange messages with the counter software in the branches. However, the BAL goes well beyond the mere exchange of information, into checking that the information is conformant (for instance, that each basket is zero-sum, applying the DEA robustness measure), logging of all exchanges, and recovery from many kinds of error conditions. Because it has to handle more than 25 million transactions per day, the BAL has many design features to ensure high performance (principally by distributing the load in parallel across many machines), as well as robustness - for instance, through reliable and redundant hardware (RHW).

198 The Branch Database (BRDB) is a large, high-performance Oracle database whose main function is to store all customer transactions which originate in any branch. It, too, has many features to ensure high performance and robustness, for instance through transactional integrity and recovery (TIN).

199 The types of data held in the branch database include:

- Customer transaction data, including both internal counter transactions and external client transactions
- Reference data to be distributed to branches
- Data that applies only to individual branches, such as users, stock units and messages
- Branch report data
- Recovery data
- Journal data
- Postal address data.

200 The logical sub-divisions of the branch database are shown below:

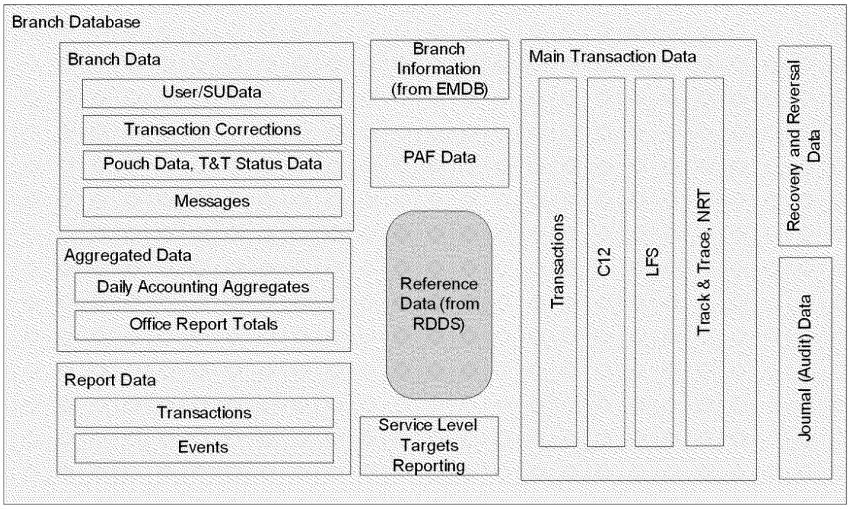


Figure 5.2 - Logical subdivisions of the Branch Database

201 The architecture of the HNG data centre is shown in the next diagram:

FUJ00083961

CHARTERIS

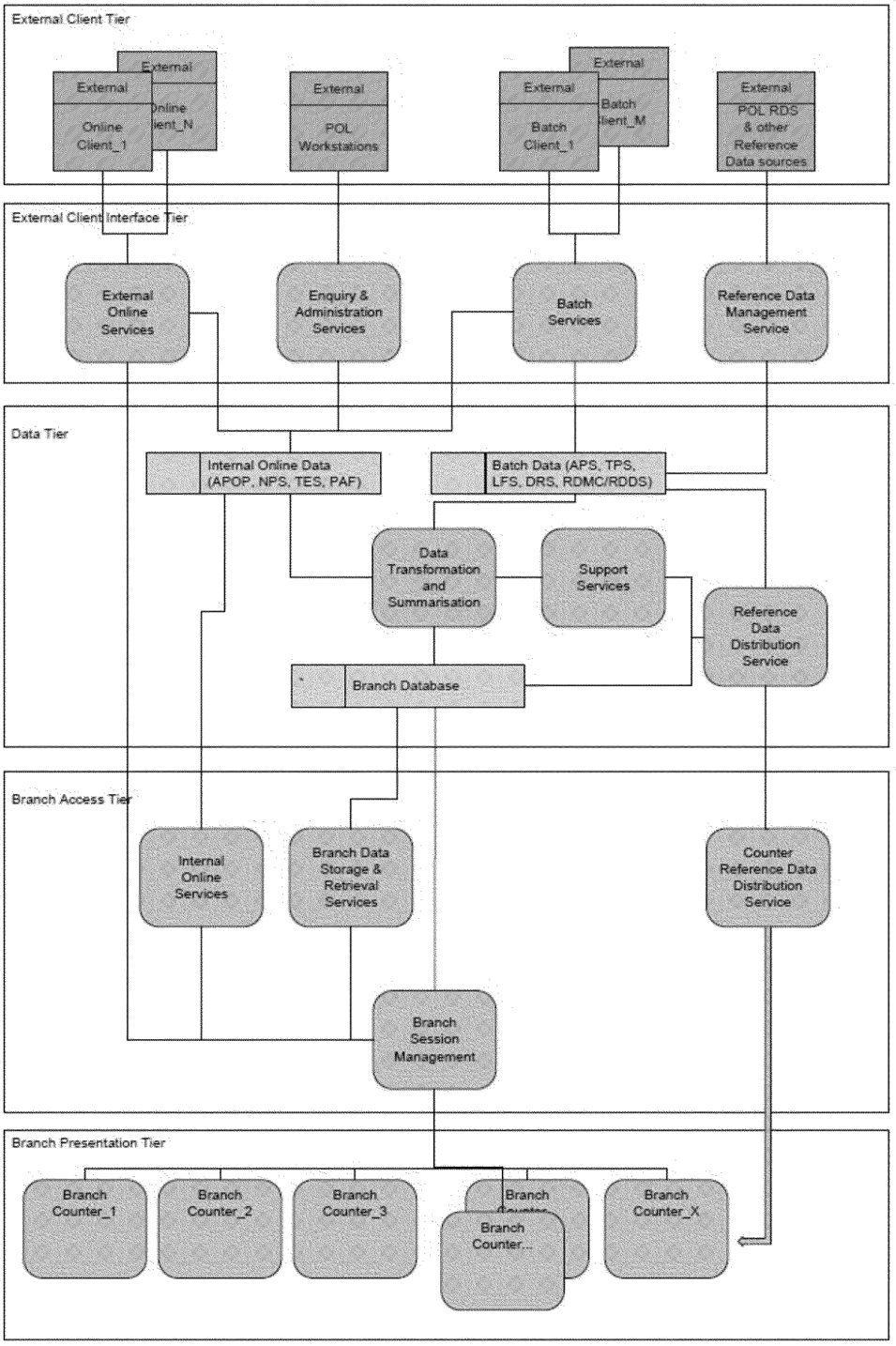


Figure 5.3 - Horizon data centre application architecture

202 From the bottom of this diagram upwards, the Branch Presentation Tier is the branch software, discussed in the previous section. The Branch Access tier (or layer) has been described above, as has the Branch Database.

203 The many layers and the defined interfaces between the applications - rather than a few monolithic applications) are all applications of architectural robustness (ARC)

FUJ00083961

204 The data tier, which includes the branch database, is a data-oriented view of the business applications, and other functionality. The External Client Interface Tier provides interfaces in both directions to external client IT systems, pictured in blue in the top layer.

205 Most of the complexity of HNG occurs in the Data Tier and the External Client Interface Tier, which together do all the back-office processing for all the different applications (more than 140 of them) supported in the branches. Because the PO has more than 140 client organisations, and each one of them may have differing requirements for back-office processing such as settlement and reconciliation (depending on their own differing IT systems), there are at least 140 kinds of back office processing to be supported. While many of these have strong similarities between them, the differences between client organisations cannot be entirely removed by the External Client Interface Tier; so, much of the complexity and diversity of these applications resides in the Data Tier. Many of these applications are batch applications, harvesting transaction data from the BRDB and running once per day in a complex batch schedule.

206 The documentation provided by Fujitsu includes many different 'wiring diagrams' of these back-office applications - each from a slightly different perspective, emphasising some aspects and abstracting out , or omitting, others. Because the full picture is so complex (with probably more than 140 boxes, and many more lines between them), it is very hard to provide the reader with simplified and useful views which may not need to be revised and amplified later for specific purposes. However, the following data-oriented diagram gives one useful view:

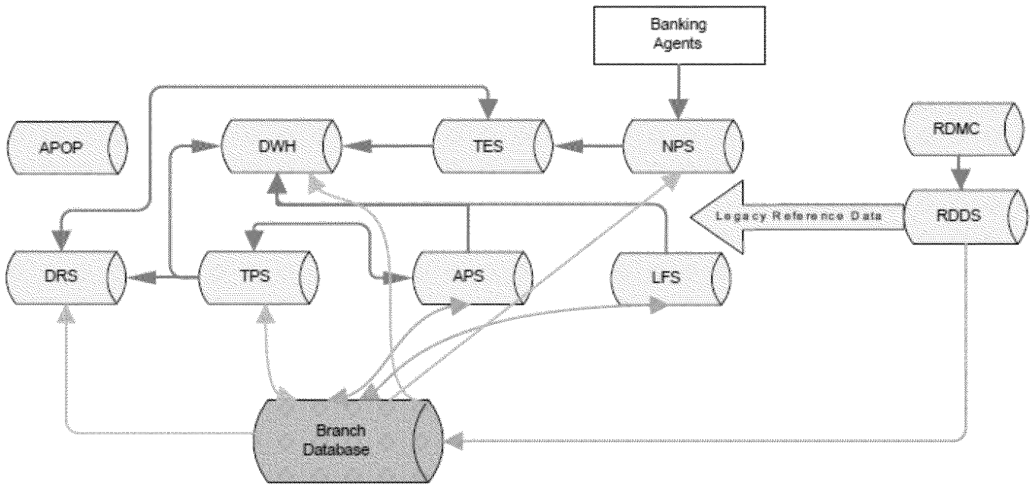


Figure 5.4 - Application Database Architecture

207 This diagram shows the central role of the Branch Database (shown in pink) and a number of so-called 'legacy databases' shown in pale pink which survived unchanged from the previous Horizon. The acronyms for the legacy databases are as follows:

- APOP: Automated Payment Out-pay Database
- APS: Automated Payment Service
- DRS: Data Reconciliation Service

FUJ00083961

- DWH: Data Warehouse
- LFS: Logistics Feeder Service
- TES: Transaction Enquiry Service
- TPS: Transaction Processing Service
- RDMC: Reference Data Management
- RDDS: Reference Data Delivery Service

208 There are of course omissions from this diagram - including, for instance, the Audit database, which continued in its previous role as described in the section 4.4 above, storing the same data in the same form as before, but now taking its information from the branch database - and providing robustness through a being a secure Kernel (SEK) with redundant storage of data (RDS).

FUJ00083961

6. ARCHITECTURAL TOPICS ACROSS HORIZON AND HNG

6.1 User Error Detection and Prevention

209 Horizon processes something of the order of 6 million counter transactions per day. Therefore, even with highly trained and careful users, it is to be expected that there will be several thousand user errors per day, spread across all the branches. It would be surprising if the rate of user errors in customer transactions was less than 1 in 10,000 - which would lead to 2,500 user errors per day.

210 It is therefore essential that Horizon should be designed to minimise the level of user errors which occur, and that it should correct them whenever possible. In addition, Horizon needs to be proof against any user error which might benefit a subpostmaster (fraud), either alone or in collusion with a customer. If a user error can be detected immediately, that implies that it can be prevented; and it generally will be. Therefore, 'error correction' often means 'error correction sometime after the event'.

211 In the design of the Horizon counter user interface, there are large numbers of measures to prevent user errors. Many of these measures have by now simply become common practice in the design of user interfaces - such as the use of menus and buttons, rather than free text input, to allow the user at any time only to choose one of the actions or inputs which are allowed at that time, or the use of facilities for inputting numerical values, which only accept numbers (not characters) in an allowed range, and confirmation buttons to ensure that the user really intended to take the action he chose. These are all cases of the robustness measure DUE.

212 Among these measures is the check that when a customer makes one or more payments for a basket of items or services, the sum of the payments entered must be the same as the summed cost of the items purchased; the basket cannot be concluded unless it is zero-sum. Also, in all possible cases (such as credit card payments) capture of the amount paid is automatic rather than manual, preventing any user error. This form of robustness is a combination of DUE and DEA.

213 Despite these measures, there remain cases where the amount of cash entered into Horizon (which gives a balancing basket) is not the amount actually put in the till; or where the amount of stock given out, such as stamps, is not the same as that entered in Horizon. There is in principle no way in which Horizon could detect or prevent these kinds of user errors. They are errors made outside Horizon, in the handling of cash or stock. So, they can only be caught by later error correction measures (user error Correction - UEC). These measures are powerful and, in the absence of later errors made in the correction process itself, will always eventually correct these sorts of user error. The delay involved in 'eventually' will be discussed below.

214 The first form of error correction is the regular checking of cash and stock made by the subpostmaster. If the wrong amount of cash is put in the till, a later check of the cash will reveal the discrepancy - which must then be corrected by putting cash in or taking cash out of the till - similarly for stock such as stamps. This means that, if there is no error in the later checking process, any errors in handling cash or stock during the day will be exactly corrected, leaving Horizon with an accurate picture of cash and stock. Multiple errors can be corrected in a single accurate check.

FUJ00083961

215 There are cases involving external clients which cannot be corrected by this check, but which can be corrected later by a reconciliation process. Suppose a customer comes to pay a £50 gas bill, but only offers £30 cash; and the counter clerk does not spot this. There are three possible cases:

a) The counter clerk enters into Horizon that a £50 gas bill has been paid, but only enters £30 as tendered by the customer. This error is prevented immediately, because the basket will not balance to zero.

b) The counter clerk enters that a £50 bill has been paid, but only puts £30 in the till. This error will be found later by the daily check of cash - and the subpostmaster will lose the £20 he has to put in to correct it. The customer saves £20.

c) The counter clerk enters that a £30 bill has been paid and enters £30 as tendered by the customer. This error is not prevented by the zero-sum check or found by the daily cash check; but it will be corrected later by reconciliation.

216 In case (c), the Post office will report to the gas company that a £30 bill has been paid and will pay the gas company a sum for that bill (and others). The gas company will know that the bill they issued to this customer was £50, not £30 - so will realise that they are £20 short. They will require an extra £20 from the Post Office, who in turn will issue a transaction correction, which in effect requires the subpostmaster to cover the loss. He will be able to see which day the bill was paid, and to see from Horizon that his desk clerk recorded an amount of £30 - while the gas company can provide evidence that the amount of the bill was £50. Again, the subpostmaster has to find £20 to correct the error, just as in case (b).

217 The extent of the delay in case (c) depends on the agreement between the Post Office and the gas company - in particular how often they carry out reconciliation - and on any further internal delays in the Transaction Correction process. However, there are cases similar to (c) where the delay is inevitably longer.

218 Suppose, as case (d), that the customer pays a £50 bill with £50 cash. In the rather unlikely event that the desk clerk puts the £50 in the till, but neglects to enter the bill payment at all into that customer's basket (so that any other items in the basket still balance), then, in his next check of cash, the subpostmaster will find that he is £50 'ahead' and may take out the cash. Nothing has been entered in Horizon and Post Office makes no payment to the gas company. The gas company will simply think that the customer has not yet paid his bill.

219 It is only much later, when the gas company starts chasing this customer, and he says: "But I paid that bill at the Post Office; and (possibly) I have evidence to prove it" that the gas company realises that they are owed £50 not by their customer, but by the Post Office. The gas company demands £50 from the Post Office; and, when the customer has identified which branch he paid the bill at, the Post Office issues a transaction correction against that branch. The net result is that the subpostmaster, who at one time thought he was £50 ahead, ends up exactly in balance, and his accounts on Horizon are correct again. However, this process may take many months, depending on how long the gas company gives its customers to pay their bills.

220 Through these measures, essentially all errors in entering amounts of cash or stock in daily trading are prevented or are corrected after some delay. It is hard to think of any such error in counter transactions which is not

FUJ00083961

caught in one of these ways. This is necessary, because, as we have seen, there are probably several thousand such errors made at the counter every day.

221 User errors made in stock taking or monthly balancing have a similar effect; they get corrected eventually. Suppose, during monthly balancing, the subpostmaster mis-counts some item of stock; so, for instance, he thinks the stock is in balance with Horizon, when it is not. Horizon thinks that the stock is in balance, with a physical stock of X units. But in fact, the physical stock is Y units. The subpostmaster should have counted it as Y units, and then made up the discrepancy (X-Y) in cash; but he did not. Then, over the next month, the changes in stock (as recorded in Horizon, and in fact) are Z units. At the end of the month, Horizon thinks that the stock should be (X+Z) units. But the physical stock is actually (Y + Z) units. So, the discrepancy, which is (X-Y) units, still has to be made up at the end of the next month. The effect of a user error in balancing in one month is just to postpone the need to balance, for another month. After balancing correctly in that month, the accounts will be accurate again.

222 In effect, users know that making an error in balancing in their own favour in one month will not get them 'off the hook' for making it good in the next month. They can postpone a problem, but they cannot make it go away. This is essential to avoid a class of deliberate user errors.

223 All of these robustness measures are part of the essential countermeasure of User Error Correction (UEC)

224 There are possible user errors in recovery situations. In practice, because these situations occur more rarely than typical counter transactions, and are often more complex and unfamiliar to them, they are more prone to user errors.

225 For instance, recovery is necessary when there is a hardware failure or communication failure in the middle of a basket of customer transactions. In HNG, because customer baskets are sent to the BRDB in one 'all or nothing' success unit, the usual effect of a hardware failure in the middle of a basket is no effect on the BRDB at all, so there is nothing to recover from; when the hardware is working again, if the customer is still there, the desk clerk merely has to start the whole basket again. The effect of hardware failure on Horizon was similar. Of course, when the failure occurs, the desk clerk must know whether any cash from the customer has been put in the till; and if the customer does not want to wait until the hardware is recovered, must give it back to him.

226 This raises two possibilities which must be taken care of:

- A hardware failure occurs at such a time that the desk clerk does not know whether a customer basket has been completed.
- A hardware failure occurs before a basket has been completed, but when some irreversible interaction with an external party (such as a credit card payment) has been made.

227 The first case is addressed by Horizon, after hardware recovery, making it clear to the desk clerk which was the last basket to complete, so he can carry on appropriately. If he has completed a basket, but Horizon has not registered it, he needs to re-enter it. Consider what happens if the clerk misunderstands this information, and thinks a transaction was completed and sent to the BRDB, when it was not; or if there is a long delay before

FUJ00083961

recovery and the transaction is not re-entered. For simplicity, consider a sale of stamps. This means a physical transaction has taken place (cash in, stamps out) whereas it has not been recorded in Horizon. Compared with physical reality, Horizon's stock of cash will be low, and its stock of stamps will be high. This means that at the monthly balancing and rollover, the subpostmaster will find these two discrepancies and will have to make them good, at no net cost to himself.

228 The second case is known as a 'recoverable transaction', because some action is needed to bring the transaction to a consistent state. A recoverable transaction occurs when some irreversible interaction with an external agency, such as an authorisation of a payment by a bank, occurs at some stage during a customer basket, and the basket later fails for some reason (such as a hardware or communication failure). Then typically some action is later required from the counter staff to 'recover' the transaction to a consistent state. User errors may occur during this recovery process - which is less familiar than the normal operation of Horizon. In these cases, typically the error is caught later in a reconciliation with the external party and is corrected by a TC.

229 User errors may also occur when making changes to stock in the presence of no customer, such as 'remming in' or 'remming out' (short for 'remitting in' or 'remitting out') cash or stock. For example, consider remming in cash that has been sent to the branch. The Post Office has a separate record of how much cash was sent. Consider when the clerk enters the wrong amount; £2000 was sent by PO, but the clerk entered £1000 (while putting £2000 in the till). This means that Horizon's record of cash in the branch is £1000 less than physical reality, which would allow the subpostmaster to take out £1000 at the next balancing and rollover. However, probably before that (because it depends only on PO's internal processes), there will be a reconciliation which detects that while some other part of PO sent £2000, the branch only remmed in £1000, This leads to a transaction correction which both alerts the subpostmaster to the mistake and puts Horizon's record of cash back in line with reality. If this happens before the next balancing and rollover, it saves any further cash movement at rollover; if it occurs after, the subpostmaster can take out £1000 which he will later have to repay.

230 So, because of reconciliation, the long-term effect on branch accounts of any user errors in stock operations are zero. This is of course necessary, to prevent any deliberate errors which would fraudulently benefit the user. However, the effect of some errors can be to require the subpostmaster, in order to roll over to the next trading period, to effectively lend money to the Post Office. This can happen if, near the end of a trading period, the user remits in £3000 but mistakenly records it twice. Then, Horizon records £6000 remitted in, whereas physically only £3000 have been added to branch cash.

231 It is necessary to understand the basis of these measures for user error correction (UEC) and to understand that they are successfully used many thousands of times in a year, across the PO branch network. They have not only been designed into Horizon; they have been tested in live use many times over. If they did not work very well, it would be impossible for PO or for SPMs to produce reliable accounts.

6.2 Intrinsic Error Prevention

232 'Intrinsic error prevention' is a broad term, because it includes any design feature of the Horizon code whose effect is to make the occurrence and persistence of serious software errors less likely. In this definition, we need to consider the levels of severity of software error, considering factors such as:

FUJ00083961

- whether the error affects only the immediate user experience, or whether it has any effects on stored data;
- whether it can affect the accuracy of branch accounts, and if so whether temporarily or permanently;
- whether its effects are so severe and obvious that it will inevitably be rapidly spotted and corrected;
- how frequently it is likely to occur.

233 Generally, we are considering only the most serious errors - those which can have a long-term effect on branch accounts, which can occur with significant frequency, and which can remain undetected for long periods of time.

234 Intrinsic error prevention includes the following techniques:

a) Double entry bookkeeping (DEA), which ensures that any numerical error affecting only one part of an accounting transaction will destroy a trial balance and be rapidly detected.

b) Transactional integrity (TIN), which ensures that in many cases, partial updates to databases, which would destroy their integrity and consistency, cannot happen.

c) Measures designed to detect or correct user errors, which, as described in the previous section, in many cases also have the effect of detecting or correcting software errors. These are so important that they have been described separately in section 6.2.3.

d) Defensive programming (DEP), where small parts of a program are written to assume that other parts of the program may be in error and are written to always check their inputs for the presence of errors.

e) Redundant storage of data (RDS), where the same information is stored repeatedly and in different forms in distinct parts of the IT estate, with consistency checks on versions of the same data. These checks include arithmetic checks of monetary sums, and many manual inspections of data (MID).

f) The audit system provides a highly secure and tamper-proof record of what is entered into Horizon at the counter, which can be used, in cases of any anomaly, to provide a 'gold standard' for comparison with data held in other parts of the Horizon estate, supporting the diagnosis of software errors. This acts as a secure kernel and redundant store of data (SEK and RDS)

g) Data-driven programming (DDS), where specific functionality is achieved by generic software modules, driven by reference data: such generic modules are simpler to code and easier to test, and the reference data is easier to manage and is less error-prone, than software code. Errors in the generic code would have such widespread effects as to be rapidly detected and corrected.

h) Software coding standards, to ensure consistency of work by different developers and to discourage coding techniques which are more error-prone. These are in effect a form of architectural robustness (ARC).

235 It is my understanding, from reading the extensive documentation of Horizon and HNG, that all these techniques have been widely and consistently applied across the whole Horizon IT estate. To catalogue the many ways in which all the techniques have been used across Horizon would be a very lengthy exercise, and in my view would not assist the court.

FUJ00083961

236 I shall discuss each of the topics (a) - (h) in the following sub-sections, illustrating where it is applied in Horizon rather than listing all its applications; then I shall later apply the topics to the analysis of specific bugs.

6.2.1 Double Entry Bookkeeping (DEA)

237 The double entry bookkeeping check has been described with examples in Appendix B.8. Wherever double entry bookkeeping is used to store any accounting information, it implies that every financial transaction is split, at an early stage of its journey through the IT systems, into separate monetary amounts (accounting postings) whose sum should be zero. In the simplest case, there are two postings of amounts +X and -X; but there may equally be three postings X, Y and Z which sum to zero. The different postings then take different routes through the system - whether they are similar parallel routes into the same tables of a database, or widely divergent routes to different databases.

238 This means that any software error affecting the finances (i.e. an error which would have an effect on branch accounts) is likely to have different effects on the different postings. Typically, a software error will affect one type of posting, but not another. This means the sum of the postings in a basket will no longer be zero, and the sum of all postings will be changed. Errors which do not affect the zero sum (e.g. doubling all postings in a zero-sum set) may occur but are in practice rare. Some errors can cause the same zero-sum set of postings to be made twice or not at all - which will not destroy the trial balance - but this type of error is usually detected by the measures for detection of user errors. We intend to address this more fully in our expert report.

239 Accounting systems such as POL FS sometimes require that account postings are put to them in zero-sum sets (this is an example of defensive programming, described below); but more important, in all cases they from time to time perform 'trial balances' to ensure that all postings that have been put to them, in whatever time order, have had a zero sum since the last trial balance. A wide class of software errors, which might affect branch accounts, would destroy the trial balance. A failure to balance the accounts is always treated as a serious condition, so any software bug which led to it would have to be diagnosed and corrected very rapidly. Most such bugs are found in testing and never make their way into live use.

240 The double entry or zero-sum constraint was applied widely in Horizon and HNG. In the HNG counter software and the Horizon counter software, any customer basket, made up of any mix of products, was required to be zero sum, and this check was made at several places in the counter software. It was also made in the HNG Branch Access Layer before entry into the BRDB; and finally, postings from the BRDB into POL FS had to be zero sum (were not allowed to destroy the trial balance) We intend to verify this for our expert report. Similar constraints applied to many types of non-customer operation, such as replenishment of stock or monthly balancing; although, as we shall see in section 9, they did not apply to all such operations.

6.2.2 Transactional Integrity and Recovery (TIN)

241 The transactional integrity constraint has been described, in the context of accounting systems, in Appendix B.5. Because transactional integrity is a fundamental facility built into all database management software, and it is necessary, for any relational database, to describe in its schema the integrity constraints which it must obey at all times, we know that transactional integrity was applied to all of the many databases of financial information in the Horizon system - including the BRDB, the POL FS database, and many others.

FUJ00083961

242 This means that any compound package of updates, applied to any of these databases, would have been applied as a single transaction or 'success unit' which would either completely succeed, or completely fail leaving no trace. It would be impossible to leave any of these databases in an inconsistent state, not satisfying its integrity constraints.

243 Transactional integrity gave protection against a wide variety of conditions:

- Hardware errors or communication failures
- Software errors which led to failures and cancellations
- Users deciding to cancel some operation when it is half-way through.

244 The use of transactional integrity at the database level makes it much easier to write software which will recover correctly from a wide range of conditions such as these.

6.2.3 Measures to Correct User Errors, which also Cancel the Effects of Software Errors (UEC)

245 In section 6.1, I have described how many classes of user error are detected and then corrected by stock counting or reconciliation processes, so that there is no permanent error introduced in branch accounts by those user errors.

246 The same design features - introduced in Horizon for correcting user errors - also cancel the effects of a large class of possible software errors.

247 Consider for instance a software error whose effect was to lose a whole basket of customer transactions, while making it appear to the counter user that the basket had been fully processed. The effect of this software error would be identical to the user error of carrying out a physical transaction with a customer - such as selling stamps - and neglecting to enter it into Horizon at all. The previous two checks - double entry accounting and transactional integrity - would not catch this software error. However, since its effects are identical with those of a user error, the measures which ultimately cancel out the effects of the user error, would also cancel out the effects of the software error.

248 In this case, the regular check of stock against physical stock would reveal two discrepancies - one of stamps, and one of cash. In order to achieve balance and roll over, the subpostmaster would have to make good both discrepancies, which he can do at no cost to himself. The final effect is that the accounts on Horizon are accurate - and the software error has not adversely affected the subpostmaster.

249 Similarly, a software error which resulted in a basket of postings being stored twice would resemble the user error of entering the same basket twice - and its effects would be later cancelled by the same mechanism.

6.2.4 Defensive Programming (DEP)

250 It is a universal modern software engineering practice to write programs defensively. This means to design a program as a set of small software modules, making the interfaces between the modules as simple as possible, with each module expecting the inputs it receives from other 'sending' modules to obey certain constraints - such as the double entry bookkeeping constraint of zero-sum baskets. The sending module is built so that its outputs should always obey those constraints, and it is tested to ensure that its outputs obey those constraints.

FUJ00083961

251 However, in a belt-and-braces approach, the receiving module should not trust the sending module - but should where possible check that its inputs obey the constraints it is expecting to be obeyed and should automatically raise an alarm if they do not. The receiving module is tested with sets of invalid inputs, to ensure that it really does raise the alarm. Then, raising the alarm is treated as a serious condition, which requires immediate diagnosis of the error and re-testing of the source module - which is usually done in testing, before any live use. If this is done, then any failures in the design, coding or testing of the sending module are detected by the receiving module.

252 This technique has become especially powerful with the use of object-oriented programming, which encourages and supports design in terms of small software modules (objects) with well-defined interfaces between them.

253 The result is that as the scale and complexity of IT applications has grown, the number of serious bugs which survive testing or persist in live use does not grow linearly with the number of lines of code - because although the number of bugs initially written in to the code may grow with the number of lines of code, the number of checks which detect and expose those bugs also grows, and may even grow faster; so the number of serious bugs which survive beyond integration testing does not grow. This good practice has been essential as the complexity of IT systems has grown in recent years.

6.2.5 Redundant Storage of Data (RDS, MID)

254 In a complex organisation such as the Post Office, there are large numbers of staff and managers with different, but overlapping, responsibilities for various parts of the business. For instance, for each of the Post Office's many client organisations, there may be a manager within the Post Office with responsibility for all the services offered in branches on behalf of that client; and there may be staff reporting to that manager. There may be supervisory managers with responsibility for groups of client organisations, and there may be staff managers with responsibility for aspects of the business across many client organisations.

255 To carry out their responsibilities, all these managers and staff require different and overlapping views of PO's business information, including that originating from Horizon. They require regular financial reports generated at different intervals, concerning different and overlapping slices of PO's business operations.

256 This means that the data generated at the Horizon counter must flow not only to the BRDB and to the central accounting system, POL FS. It must also flow to other databases and data warehouses used to generate other reports. Furthermore, as IT systems are usually, for technical reasons, more complex than a layman would expect from understanding their requirements, there is a larger number of different stores for the same data - in many different slices and representations - than one would at first expect from PO's business needs.

257 Because there are many redundant copies of the same data, it becomes possible to carry out automatic checks that these redundant copies of data are consistent with one another. In a simple example, one database may hold daily summaries of some financial or stock information, while another database holds weekly summaries. These two summaries may have reached those databases by different routes through the organisation and its IT estate. However, there can be a simple and powerful arithmetic check that the weekly summaries are consistent with the daily summaries. It is common good practice to build the IT systems to make these checks wherever

FUJ00083961

possible, and to raise an alarm if any check fails. Failure of a check may arise from some software error, or from a user error, or from an IT operational error such as failing to run some daily batch process. Whatever the cause, it needs to be diagnosed quickly - because until it is fixed, some of the reports from those data will not be useful to managers - being known to be inaccurate.

258 There is a further check on the correctness of the data, in that different managers all look at the reports they receive and have many discussions about the content of those reports. If any error in the data leads to regular inconsistencies between the different managers' views of the business, those inconsistencies are usually soon revealed and must be corrected.

6.2.6 The Audit System (SEK, RDS)

259 I have described a situation in which data about some customer transaction in a branch, having reached the BRDB, then fans out (typically through several 'harvester' programs) to many different IT applications and databases in the back office.

260 As has been described in section 4.4, the audit sub-system of Horizon holds a reliable and tamper-proof record of all accounting transactions initiated at the counter. So, in the case of any discrepancy between two or more of the many other databases and systems which comprise Horizon, the conflicting versions can each be compared with the audit system record, which can serve as a reliable record of what was entered at the counter.

261 This means that any software error occurring in any of the systems in Horizon, which has an effect on branch accounts, will lead to a discrepancy between that system and the audit system. For a software error, the same kind of discrepancy will probably occur on many occasions, and on each occasion can be investigated by a comparison with audit data. The audit data makes it easier to isolate and correct software errors in any other Horizon system - particularly those errors which might affect branch accounts.

262 Comparison with the audit system may also help in detecting data errors which may have arisen in other ways, such as:

- an error in running one of the many daily batch processes;
- a user error by some member of PO's back office staff; or
- any tampering with branch accounting data.

263 The audit system gives protection against tampering with branch account data, because audit records are signed with digital signatures dependent on a password known only to branch counter staff.

6.2.7 Data-Driven Programming (DDS)

264 Another common modern software engineering practice which has been applied in Horizon is data-driven programming. The data in question is often referred to in Horizon as reference data (although there are also other kinds of reference data).

265 For instance, as described in section 4.2, data-driven programming has been referred to in the old Horizon desktop software as 'soft-centred' applications. As a second example, section 5.3 describes the Services layer of HNG which contains a general 'process engine', driven by reference data in the form of PDL (Process

FUJ00083961

Definition Language), which provides a straightforward way of supporting different business processes at the counter.

266 The effect of this data-driven approach that, instead of having to write specific software to handle different uses cases, the developer writes generic software which is driven by different reference data for different use cases. This improves the reliability of software in three ways:

- The generic software is often simpler that the specific software which would have to be written to support different uses cases; therefore, it is less prone to errors.
- The generic software is tested by applying it to all the different use cases; therefore, the generic software is more thoroughly tested, and less likely to contain undetected errors.
- The reference data, which must be supplied for each use case, is much simpler and easier to read and understand than the code which would otherwise be written, and is often the subject of static validation checks, which are easily made. Therefore, any errors in the reference data are easily detected and corrected.

6.2.8 Software Coding Standards (ARC)

267 The use of software coding standards is mentioned in section 6.7 on development and testing of Horizon. It is also covered here because of its impact on the likely level of serious errors in Horizon.

268 If programmers are allowed to develop software, each in their own preferred style, then each one may use different programming techniques, with the result that they may have difficulty understanding the code that other people have written. It is a practical necessity for programmers to understand, extend and modify each other's work - so lack of mutual understanding of code would be harmful. Therefore, most organisations apply software coding standards to ensure consistency and mutual understanding of code.

269 These coding standards develop over time in such a way as to discourage programming styles which are more error-prone. They may include recommendations such as defensive programming, and other forms of checking - which will reduce the level of undetected errors.

6.2.9 Summary of Intrinsic Error Prevention Methods

270 In my expert report, I intend to show from my analysis of the Horizon design, coding and of its development standards, and of Fujitsu's test records, that as Horizon has grown, the expected level of serious software errors, as defined above, has not increased but rather has decreased. We expect to find this because the number of checks implied by (a) - (h) has grown faster than the number of lines of code. If the number of checks grows faster than the number of possible software errors (which is roughly proportional to the number of lines of code), then the number of errors which escape all checks does not increase, but rather decreases to a very low level

271 The test of this conclusion is the in-service record of Horizon. The claimants' expert has, by examining the Known Error Log and other sources, found evidence for a number of specific bugs which have occurred during the in-service life of Horizon and HNG, and I have conducted similar searches. In our expert report we shall analyse those bugs according to criteria described under heading (a) - (h), to assess how many of them could

FUJ00083961

have had permanent effects on branch accounts. The result of this analysis will be the main conclusion of our expert report.

6.3 Financial Audit

272 A core requirement of any accounting system is to support the activity of annual financial auditing, which is required in order that external stakeholders in the organisation, such as shareholders, can have confidence in the published annual accounts. Confidence in the annual accounts has a number of dimensions - including checks that any discretionary element to the accounts, which might move profit from one year to another, have been fairly applied by the officers of the company. Audit also includes checks that the financial accounts are an accurate reflection of business activities, and that the accounts could not have been altered by any fraudulent activity.

273 So, the annual financial audit may be regarded as just another management cross-check on the figures in the accounting system - but for a large organisation like the PO, it needs to be a particularly careful and thorough one. Auditors need to be aware of how inaccuracies, poor controls, adverse financial events or fraud might be covered up or hidden in the accounts, and to examine those areas with particular care.

274 Constraints of time and resources may mean they can only do this on a selective basis - but even when auditors work selectively, they must be empowered to drill down to any level of detail, and the accounting system and the auditing process must give them an ability to do so. They need to be able to retrieve any figures they choose, analyse them as necessary, and interview managers about them. They need to be able to do this outside the control of the organisation itself. This places large demands on the ability of an accounting system to produce many different views and subsets of the accounting data - and any one of those views may then be subject to the consistency check of discussions between the auditors and the management.

275 During much of the life of Horizon, the PO accounting system has been POL FS (later known as POL SAP), both built on the SAP ERP system. SAP is a very mature and capable ERP system, with highly mature and capable accounting facilities and reporting facilities built into it. It is entirely capable of supporting a probing and independent financial audit process. At all times, the financial data within POL FS about branch operations has come from Horizon or HNG. Had those systems been providing erroneous or unreliable data to POL FS, it would have been the business of the financial auditors to draw attention to this fact. Our expert report may include limited factual evidence about financial audit.

276 Financial audit is a test of the robustness of the IT system and the business processes around it. Auditors check by looking for redundant storage of data (RDS) with extensive manual inspection of data (MID), checking the constraints including those of double-entry book-keeping (DEA). I am not aware that financial audit of PO has led to any serious concerns, but I need to review the evidence.

6.4 Reconciliation, Transaction Corrections and Acknowledgements (UEC)

277 The claim and the defence have drawn attention to transaction corrections. This section briefly describes how they operate and how they usually arise as from a reconciliation operation.

FUJ00083961

278 Whenever the PO acts as an agent for some external client organisation, financial transactions take place for which both the PO and the client organisation have a record. The PO's record of the transaction starts at some branch counter on Horizon. The client's record of the transaction may be available to the client immediately - as in the case of a cash withdrawal from a bank - or it may only be possible to link the client's record with the PO record after some delay. This happens, for instance, where customers pay bills at a PO branch. Initially, the client organisation issues a bill to a customer - so knows the amount of the bill. The customer then pays the bill at a PO branch - after which the client organisation may check that the amount paid was as billed.

279 Whatever the delay involved, there is eventually a process of comparing a client organisation's record of events taking place at branches, with the PO's own records. This process is called reconciliation, and it is done on a transaction by transaction basis. There are many differences of detail in how reconciliation is carried out for different client organisations, or where it is carried out; sometimes the client organisation does it from a file sent it by the PO, and sometimes the PO does it.

280 However it is done, and wherever it is done, the result of reconciliation is always in principle the same. For the vast majority of transactions carried out (approximately 6 million per day), the client's record of a transaction and the PO's record match exactly, and there is nothing more to be done. However, for a small minority of transactions (which is typically a few thousand per day) there is some mismatch, which needs to be investigated and corrected. How this is done may depend on the terms of the contract between the PO and the client organisation.

281 When reconciliation finds a transaction for which the PO record and the client record do not match, it is passed to a department in PO accounts which handles reconciliation discrepancies. Each such discrepancy is, until it has been dealt with, an error in the accounts - and so it must be dealt with. The task of this department is to determine how each discrepancy arose, which therefore how it needs to be dealt with. There is only a small number of ways in which a discrepancy may have arisen:

- It may have arisen though a mistake in the client organisation - in which case, the client organisation will need to correct its accounts, taking a loss if necessary. When the client organisation's systems are entirely automated (as, for instance, with a bank) this is extremely unlikely to occur - and the PO would need to have good grounds for believing it arose from a client mistake.
- It may have arisen from some mistake made in the back office of the PO, which would include a bug in Horizon or a failure in running some batch process. Since these processes are highly automated, it is likely that any such failure, if it occurred, would produce a series of reconciliation failures with some very distinctive pattern, which would be easily recognised as such and soon corrected. While this was being done, it would be necessary to correct the PO accounts by adjusting the figures of some central department, which would not affect the accounts of any branch.
- The most likely cause is that the discrepancy arose through some manual error in the branch. In this case, the correction of the discrepancy must be made in the branch accounts, unless the subpostmaster can show

FUJ00083961

in some way that it did not arise in the branch or was not his or her responsibility. The process for doing this is a transaction correction.

282 When the appropriate department in PO decides that responsibility for a discrepancy lies with the branch, a request for a transaction correction is issued and is passed from POL FS to the BRDB. At this point, there is no impact on branch accounts. The request is passed on from the BRDB to the branch Horizon system, so that it will show on the subpostmaster's screen when he starts the Horizon system the next morning. At this point, the subpostmaster may either accept the correction or may dispute it and ask for further investigation.

283 It is only when the subpostmaster has accepted a transaction correction that it enters his branch accounts, and therefore enters the audit sub-system in a record sealed with his own password.

284 Transaction corrections may arise for reasons other than reconciliation discrepancies. For instance, when a subpostmaster recognises that his accounts need correction (for instance, after mistakenly remming in the same amount of cash twice), he may request a transaction correction.

285 Transaction Acknowledgements (TAs) occur more frequently than Transaction Corrections, because they occur through normal branch business, in the absence of any errors.

286 For instance, when a customer pays a bill at a Paystation, the Paystation terminal transmits amounts to Ingenico, who inform the PO each day what the transaction totals were for each branch. This results in a TA being sent to the branch. When the TA is accepted by the subpostmaster, it enters the branch accounts to balance the cash from the Paystation.

287 Similar processes apply to Camelot, because lottery terminals are not directly connected to Horizon, and operate outside Post Office business hours.

288 Thus, Transaction Acknowledgements are used to balance branch accounts for cash received at a branch which is not automatically entered into Horizon at the time it is received - because the cash amount is recorded on a separate device not connected to Horizon.

289 There are cases where the TA circuit through Ingenico does not work, because of connectivity issues. In those cases, the subpostmaster can find out the cash amount from a printed summary from the Paystation and request a manual TC to balance the cash with the branch accounts. Without either a TA or a TC, the branch would have a cash surplus.

6.5 Hardware and Software Resilience (RHW)

294 The Post Office's branch business depends heavily on Horizon. One of its design principles is that the system should support non-stop trading during core business hours. Therefore, it is important that the system operate successfully even when its computer hardware, software or networks fail. The ability of an IT system to protect users from any type of disruption and to maintain acceptable service levels is known as 'resilience'.

295 Resilience is required in all the major components of Horizon in branches, data centres and networks:

- Hardware
- System software, such as the DBMS and communications products

FUJ00083961

- Horizon infrastructure and applications software
- Networks

296 'A single point of failure (SPOF) is a risk posed by a flaw in the design, implementation or configuration of a system in which one fault or malfunction causes an entire system to stop operating.' [8] One strategy for minimising the impact of a failure is to replicate major components of the system. Error processing and recovery procedures also improve resilience.

297 Several specific qualities of the system, which can be viewed as aspects of resilience, are addressed in other sections of this report:

- Recovery from failure is discussed under Intrinsic Error Prevention - section 6.2.
- Transactional Integrity – section 6.2.2.
- Security – section 6.6.

298 See Appendix C for more detail about the resilience built into Horizon hardware, software and networks.

6.5.1 Conclusions

299 Fujitsu has established a resilient framework to guard against disruption of Post Office's Horizon service. The framework rests on three Rs – replication, redundancy and recovery. It covers risks ranging from failures of peripherals attached to a PC in a branch up to destruction of an entire data centre.

300 The Horizon network has been steadily upgraded with alternative routes being provided.

301 Recovery is automated as far as practicable, but still depends on some user intervention under guidance.

6.6 Security and User Authentication

302 Post Office services supported by Horizon are used every working day by millions of people. The value of their transactions represents an important part of the UK economy. It is vital that users, PO staff, independent auditors, the government and the public at large trust Horizon. This trust is underpinned by a secure computer system, qualified in broad terms as follows: '*IT security is the protection of computer systems from theft and damage to their hardware, software or information as well as from disruption or misdirection of the services they provide.*'[9]

303 This section of the report summarises Horizon security and user authentication measures, starting with the legacy system.

304 Horizon data is secured using the standard security principle of 'separation of duties'[10]. Separation of duties ensures that an individual cannot complete a critical task by themselves. For example: someone who submits a request for reimbursement should not also be able to authorise payment. An applications programmer should not also be the server or database administrator - these roles and responsibilities must be separated from one another.

8 https://searchdatacenter.techtarget.com/definition/Single-point-of-failure-SPOF

9 Wikipedia

10 The general principle is defined in https://en.wikipedia.org/wiki/Separation_of_duties. Horizon's adoption is confirmed in SVM/SDM/PRO/0875, section 1.5.3.

FUJ00083961

305 This principle has been implemented by ensuring:

- Development units cannot have update access to any of the system data.
- Database administration functions are carried out by IS (Infrastructure Services) staff.
- Data repair is carried out by SSC staff.

306 Data protection legislation requires that access to personal data remains within the European Union and PCI (Payment Card Industry) data security standards mandate physical security restrictions must be applied where update access is allowed to user data. Only SSC and the IS Unix team fulfil these requirements for data access. The responsibility for data correction is rests with the SSC although IS sometimes act under SSC authorisation. In my opinion, this SEC countermeasure helps to reduce errors and prevent fraud and enhances the overall robustness of Horizon.

307 Access to Horizon services and system components is restricted to those who are properly authorised to use those specific services and components. Authentication seeks to verify the identity of a person (or system component) seeking to gain access to a system resource.

308 Three important pillars of security are as follows:

- Confidentiality - unauthorised observation or inference of information (e.g. about benefits paid to claimants);
- Integrity - unauthorised manipulation of information (e.g. of the data passed to and from PO and its clients);
- Availability - unauthorised denial of service.

309 Vulnerabilities could cause lasting reputational damage and financial loss to both PO and Fujitsu.

310 A fourth pillar is audit and accountability: ensuring that users are held accountable for their actions by recording information about these actions. Threats are reduced if users understand that they will be held accountable for their actions. Audit is discussed in Appendix B.12 and in sections 4.4 and 6.2.6.

311 Security and user authentication are highly technical topics. This section of the report aims to provide an introduction only. See Appendix D for the next level of detail.

312 As described above, the Horizon architecture includes measures to address confidentiality, integrity, availability and audit. Some of the data is supplied by the government and is classified as Restricted. Because of this, security measures must follow the guidance of CESG[11]. Other data is associated with financial transactions and so the regulations of the financial services industry are applied.

313 A detailed risk assessment was undertaken for Horizon when the system was being designed in the late 1990s, which resulted in the following security policies being adopted:

- Physical and logical access to the system is controlled, with access granted selectively and permitted only where there is a specific need. Access is restricted to people with appropriate authorisation.

11 The UK government's National Technical Authority for Information Assurance, now part of the National Cyber Security Centre

FUJ00083961

- The identity claimed by a user is verified before any access is granted to the system. Authentication mechanisms also ensure that trust relationships are established between components within, and external to, Horizon.
- All users are individually accountable for their actions. Owners are assigned for all information assets. The owners are responsible for defining who is authorised to access the information. Responsibilities may be delegated, but accountability remains with the designated owner of the asset.
- Audit mechanisms monitor and detect events that might threaten the security of Horizon itself or any service to which it is connected. These mechanisms also ensure that transactions and other events are reliably and securely recorded as described earlier in this report.
- Security personnel are alerted to violations that could seriously threaten the services.

314 The main security risks that could impact branch accounts may be summarised as follows:

- Unauthorised access – processing transactions in branch via a user account or remotely without appropriate permission; this risk includes direct access to or manipulation of branch accounts without permission. The risk is minimised by rigorous control of user identities and their access to resources.
- Theft or damage to Horizon equipment, after which the correct position is not reinstated. Resilience to this risk depends on robust procedures for recovering from failure or loss of system components and any other dislocations of the service.

6.6.1 Conclusions

315 Security was recognised as a fundamental requirement of Horizon from the earliest days, rather than something that had to be 'bolted on' later.

316 The principle of separation of duties improves the overall robustness and security of the system.

317 Users (and other 'agents' such as counter PCs) must authenticate their identities before they are allowed to access the system. Each user, even after they have passed this first gate, is only permitted to perform one or more well-defined roles. Each role is authorised to exercise a specific set of functions and not others. This mechanism of role-based access control is established as good practice across systems like Horizon.

318 Horizon uses encryption and digital signatures, which are also seen as industry-standard techniques, as part of its security framework.

319 We consider that the security measures employed by Horizon are at the level required for such a system.

6.7 Development and Testing of Horizon

320 Build on research notes

321 **Note**: This is not just about FJ – PO also has roles to play

322 Need to focus slightly more on development of Horizon over its lifetime. Start using 'Legacy Horizon' more often.

FUJ00083961

323 Horizon Issue 3 reads as follows: To what extent and in what respects is the Horizon System "robust" and extremely unlikely to be the cause of shortfalls in branches?

324 Robustness and the likelihood that the system is the cause of shortfalls are determined partly by the way the system operates and partly by the way it is built and maintained. I have described the facilities provided by Horizon and its associated controls earlier in this report. This sub-section introduces the development and testing lifecycle. The following section 6.8 describes how Horizon operations are supported and the system maintained.

325 From the early days of Horizon, more than twenty years ago, Fujitsu has deployed hundreds of staff on the system. Teams on that scale need comprehensive and detailed documentation, underpinned by rigorous processes, to be successful.

326 Good quality documents and processes will only be a drain on resources unless they are put into practice and followed.

327 In this section of the report (and the next, section 6.8), I provide my opinions as to whether this has occurred and whether the methodology is effective.

6.7.1 Organisation and Governance

328 The aims of this part of the report are twofold:

- to show how Fujitsu has managed and controlled the people behind Horizon to meet PO's objectives; and
- to comment on how well their organisation and governance mechanisms meet the needs.

329 In 1986, Post Office Counters Limited was formed as a subsidiary of the Post Office Corporation, with services being split between the two entities: the delivery of mail was handled by the Corporation and the Post Office retail network was transferred to the new subsidiary. In 2001, the Post Office Corporation became Royal Mail Group (RMG) and Post Office Counters Limited was renamed Post Office Limited. On 1 April 2012, the UK government privatised the Royal Mail Group, with Post Office Limited remaining in public ownership. Up to this point, although Post Office was a separate legal entity, it shared some back-office services with Royal Mail.[12] I include this corporate history in my report to explain why both PO and Fujitsu have placed some reliance on RMG documents.

330 PO contracted with Fujitsu as its Preferred Systems Integrator. Most of Horizon's development, testing and operation are led by Fujitsu but this must be conducted in close cooperation with PO.

331 ICL created its Pathway division for Horizon; it later became part of Fujitsu's Post Office Account (POA). Pathway was formally organised, in the sense that its structure and the responsibilities of the component parts were defined in documents that formed part of the way the programme was delivered. At that time, the most important part of the organisation was known as the Development Directorate. This was responsible for the

12 This paragraph (as far as this point) is an abbreviated version of paragraph 20 from the Witness Statement (for the Common Issues trial and dated 24 August 2018) of Angela van den Bogerd, PO's People Services Director.

FUJ00083961

design and development of all the hardware and software, and for its integration into the Horizon system. Fujitsu refers to the document that defines the organisation of this directorate as DE/PLA/009; it was produced in the year 2000. In addition to describing the responsibilities of particular roles, it defines how this part of the organisation was managed and controlled. Those mechanisms included various governance boards. The processes, standards and programme documentation more generally lay down how people should work to meet PO's objectives for Horizon.

332 The organisation that originally built Horizon was absorbed into the broader POA Systems Integration (SI) Directorate.

333 Fujitsu has adapted the Horizon organisation in line with the evolution of the system. One such change was the implementation of HNG. This was a major programme of work lasting several years and culminating in 2010. That programme was run alongside 'business as usual', i.e. the operation and support of the original Horizon system. Fujitsu therefore had to organise their resources over that period to balance and meet those demands.

334 Fujitsu remains PO's main partner for Horizon. Nevertheless, PO has steadily diversified its supplier community. For example:

- Atos now provides first line support via the Service Desk;
- Computacenter supplies hardware; and
- Verizon looks after networking.

This strategy may deliver a better match to PO's requirements, at the cost of increasing the number of relationships that it has to manage.

335 Fujitsu's current organisation chart[13] for their Post Office Account, which includes Horizon, runs to more than 30 pages. It is clear from the first page (see Figure 6.1 below), as well as from many of the others, what a prominent role is played by Operations.

13 Dated 27 July 2018

FUJ00083961

CHARTERIS

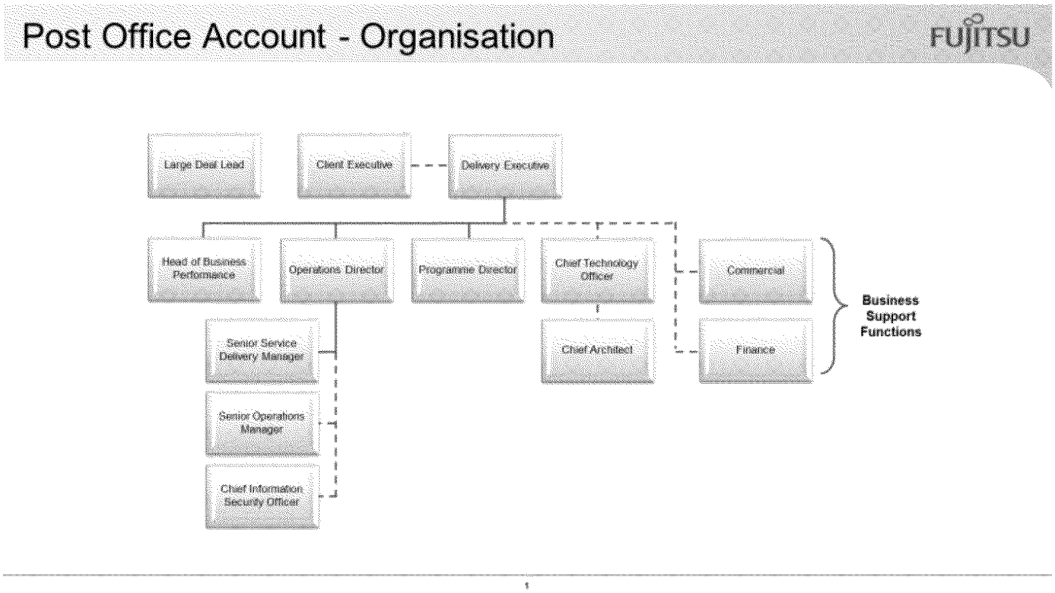


Figure 6.5 - POA Organisation

336 The Service Management chart (Figure 6.2) shows how key functions such as Incident and Problem Management and Software Support (3rd Line) are managed. The chart also highlights the importance attached to governance and risk management.

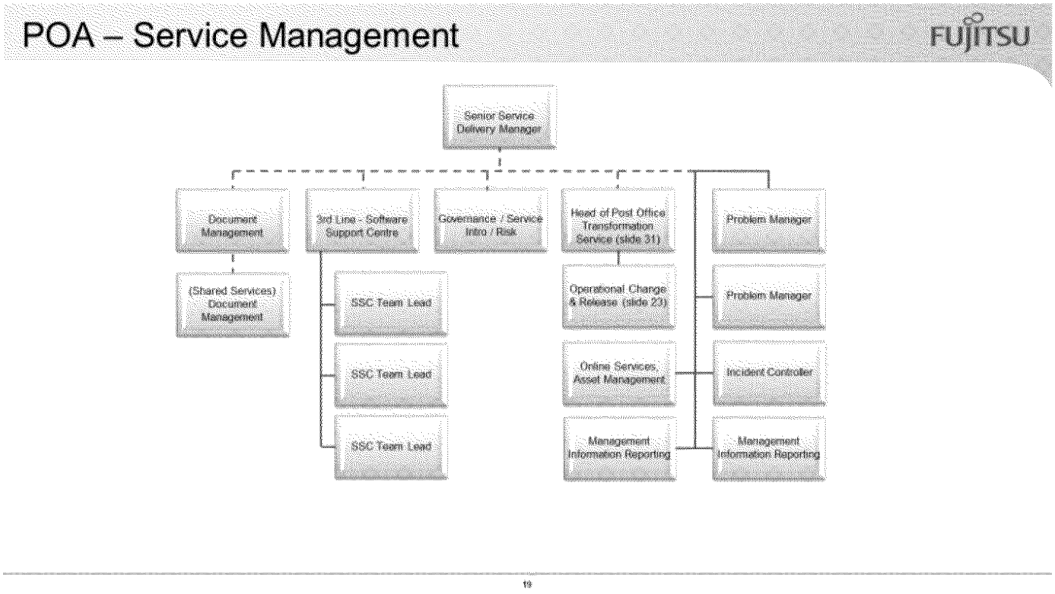


Figure 6.6 - POA Service Management

Governance

337 The contract[14] between PO and Fujitsu laid the foundation for a constructive and successful working relationship between the two parties. Schedule A2 states as its first objective, in clause 1.1:

14 I have seen the All In One Volume Contract v12 dated 03 July 2017.

FUJ00083961

'The Parties intend that the relationship and governance arrangements under this Schedule A2 will help achieve the aim of ensuring an effective working relationship between the Parties and the overall success of the Agreement.'

338 A2 defines five high-level decision-making Relationships between the parties. The document lays out the representatives, their responsibilities and frequency of meeting. The Programme/Release Relationship (one of the five) is governed by four boards.

339 As Horizon matured, the governance arrangements were refined and extended. Boards and other forums were created as mechanisms for managing and controlling the work of Fujitsu teams themselves as well as the working relationships between Fujitsu, PO and third parties.

340 In 2009, RMG defined a Programme Governance Framework[15]. This described how the delivery of programmes and projects was to be managed. It covered the following topics:

- Programme/project organisation
- Roles and responsibilities
- Structure and form of the governance processes and meetings central to management of the programme.

341 Although this framework did not apply to all Horizon programmes, its principles (such as the application of MSP and PRINCE2[16]) did apply and have been built into other governance arrangements.

342 Two examples of governance boards are the Business Impact Forum (BIF) and the Peak Targeting Forum (PTF)[17]. These particular forums were defined in 2014 to manage observations recorded in the Peak system (known informally as 'Peaks'), which are intended for maintenance releases. Those destined for major releases are dealt with by another forum, the Quality Filter Process (QFP). The processes surrounding these forums are discussed in section 6.8.3 below.

343 Peak is a software incident and defect management system. It is used by the Third- and Fourth-Line Support Units. It enables diagnostic details to be captured in a searchable format and allows the tracking of problems from detection through to resolution.[18] Peak is also discussed further in section 6.8.3 below.

344 In my opinion, the organisation and governance of Horizon are commensurate with the challenges. They have also been continually adapted in line with the demands arising from PO requirements.

6.7.2 Quality

345 Fujitsu is committed to quality across its business, including customer work. It established a set of policies and processes for the Post Office Account, which were known as the Business Management System (BMS)[19]. This was intended to ensure that the company met all the requirements agreed with Post Office. The BMS and its supporting documentation were consistent not only with Fujitsu's Corporate Policies and Processes, but also with national and international standards (BS EN ISO 9001:2000, the TickIT Guide v5.0 and the CMMI for

15 RMGA/PGM/MGT/STD/0001

16 MSP (Managing Successful Programmes) and PRINCE2 are widely accepted methods for programme and project management respectively.

17 See SVM/SDM/STD/2593 Terms of Reference for BIF and PTF

18 See CS/MAN/011 Peak User Guide

19 PA/POL/002

FUJ00083961

Software and Systems Engineering v1.1)[20].

346 In the year 2000, a plan[21] was published to manage quality assurance for the Horizon programme. The plan was updated annually.

347 Be specific about the quality and level of detail of the documents – structure, organisation and management, traceability of requirements, cross-referencing

348 Regularly maintained – most important documents have passed through several revisions: for improvement and in line with business changes

349 Samples

a) Many sample documents have been examined

b) Original Horizon documents were comprehensive and thorough

c) HNG is essentially an upgrade

d) Not all documents have been revised

e) Updates tend to focus on changes

350 Paragraph 126 introduces the three architectural layers of Counter, Agent and Host. Section 6.2.8 above discusses software coding standards. Fujitsu has explained that they followed separate standards for each layer. For example, the Host Applications Design and Development Standards are referenced as DES/GEN/STD/0001.

Quality Control

351 Check for current definitions and broader usage vs QA

352 Introduce QCC countermeasure – defined in the master table, so how to refer here?

353 'Quality and Change Control (QCC) – Systems are more robust if quality is inherent. This is achieved by organising properly the people who build, maintain and operate the system, by managing them well and by governing what they do through rigorous but effective processes. A system will only continue to be robust if changes are controlled in a way that enhances quality without unnecessary administration.'

354 Governance, management (see above), reviews and testing – are examples of QC techniques (any important omissions?)

Audits

355 Bill Membery is currently Head of Quality and Compliance for Fujitsu's POA. He has overseen audits of Horizon and provided a Witness Statement (dated 28 September 2018). The following paragraphs are derived from that evidence:

20 ISO is the International Standards Organisation, and ISO 9001 is a Quality Management System. The TickIT guide provides guidance on applying ISO 9001 in the IT industry. CMMI stands for Capability Maturity Model Integration, which is the leading measure of software engineering process maturity.

21 QU/PLA/005

FUJ00083961

'8. The Horizon system (both Horizon Online and Legacy Horizon) are and have been subject to audits to internationally recognised standards. These audits check both technical aspects of the system and the working practices of Fujitsu around Horizon. They provide assurance that Horizon and Fujitsu are working within a robust system of control measures. The audits review both the design of the controls and their implementation in practice. Completion of these audits provides assurance to Fujitsu, Post Office and other third parties (e.g. clients of Post Office) that the financial information within Horizon can be relied on.'

356 Following the introduction of HNG in 2010, PO and Fujitsu agreed that Horizon should be audited against ISAE[22] 3402. This standard applies specifically to service, as opposed to development, organisations.

357 Mr Membery's evidence continues as follows:

'11. Ernst &Young have carried out the Horizon ISAE 3402 Audit since the 2012/2013 financial year (preparations for the audit began with Post Office in 2011).

12. The Horizon ISAE 3402 Audit covers control components including: the control environment; control activities; information and communication; monitoring; risk assessment; integrity and ethical values; business lines and their functions; governance and oversight of control activities; human resources; policies and practices; and performance management.'

358 Before ISAE 3402 was adopted, Horizon was audited against ISO 20000[23], ISO 9001 and BSI 7799 which was superseded by ISO 27001[24]. As PO accepts card payments via Horizon, the system is also audited for compliance with the Payment Card Industry Data Security Standards (PCI DSS). Those audits are carried out by Information Risk Management plc, with an emphasis on cardholder data.

359 Both ISO 9001 and ISO 27001 audits were (and continue to be) carried out by different accredited companies and auditors such as the British Standards Institution (BSI) and Bureau Veritas for ISO 27001. Audits to these standards were also carried out by Fujitsu internally.

360 Thus, the quality of Horizon and its associated services have been monitored consistently throughout the life of the system.

CMMI

361 CMMI (Capability Maturity Model Integration) is the leading objective measure of software engineering process maturity internationally. It can be used to assess an organization against a scale of five process maturity levels:

- Level 1 – Initial
- Level 2 – Repeatable
- Level 3 – Defined
- Level 4 – Managed

22 International Standard on Assurance Engagements

23 ISO 20000 is the international standard that describes best practice for IT service management. It is strongly linked to ITIL (described in section 6.8.1).

24 ISO 27001 is a specification for an information security management system, which superseded the equivalent British standard BSI 7799.

FUJ00083961

- Level 5 – Optimising

362 Fujitsu undertook a corporate initiative to obtain CMMI Maturity Level 3. As part of this initiative, POA achieved that status in December 2005. CMMI measurement was continued as HNG-X moved forward with the aim that the programme would be formally appraised during 2007 to Maturity Level 3.

363 Recognition at CMMI Level 3 underpins my view that the Horizon processes have been well established and steadily improved over time. This, in turn, enhances both the quality and robustness of the system.

364 Still need to reference source documents

365 Is this enough on CMMI (JC doesn't mention)?

366 Or, do I need to draw out conclusions about independent verification of process maturity and quality

6.7.3 Lifecycle

367 Must minimise changes here. Coyne fails to cover dev/test altogether.

368 Consider specifically whether descriptions of evolution of the processes over the history of Horizon below are sufficient

369 Check for obvious improvements over time, especially to HNG-X

370 Note that some significant new documents were introduced with HNG – such as?

371 Adapt the following text

372 One of the early building blocks of Horizon was TD/ARC/001 Technical Environment Description. This document discusses the methods used to develop applications to run on the Horizon architecture.

373 A fundamental requirement of Horizon was the ability to develop new applications, and integrate them into the system, in response to new business opportunities, in a speedy and cost-effective manner.

374 Applications or their components are developed by:

- Suppliers who supply generic software such as database products;
- Third parties who supply applications or application components that meet Horizon requirements;
- Fujitsu developers who develop software to meet specific business or strategy goals;
- PO who can now introduce new clients without reference to Fujitsu by configuring standard components.

375 It is important that applications from any of these sources should be capable of integration with other applications.

376 As discussed above, Horizon applications (branch and back office) are architected as horizontal layers. The core applications delivering Horizon functionality are supported by a series of ancillary systems - such as the Transaction Processing System (TPS), Management Information System (MIS) and a system to manage reference data. A series of further applications assist staff with system management and support services.

377 Infrastructure (in the form of hardware platforms and networks) are accommodated in the lifecycle alongside the applications.

FUJ00083961

378 TD/ARC/001 proposed a reference model for application development. System development lifecycles are often depicted using a V-model and Horizon is no exception. The left side of the 'V' represents the creation of system components to meet business requirements. The right side represents testing and integration of those components culminating in the release (and maintenance) of the solution to meet the original need. Horizon's model is shown below:

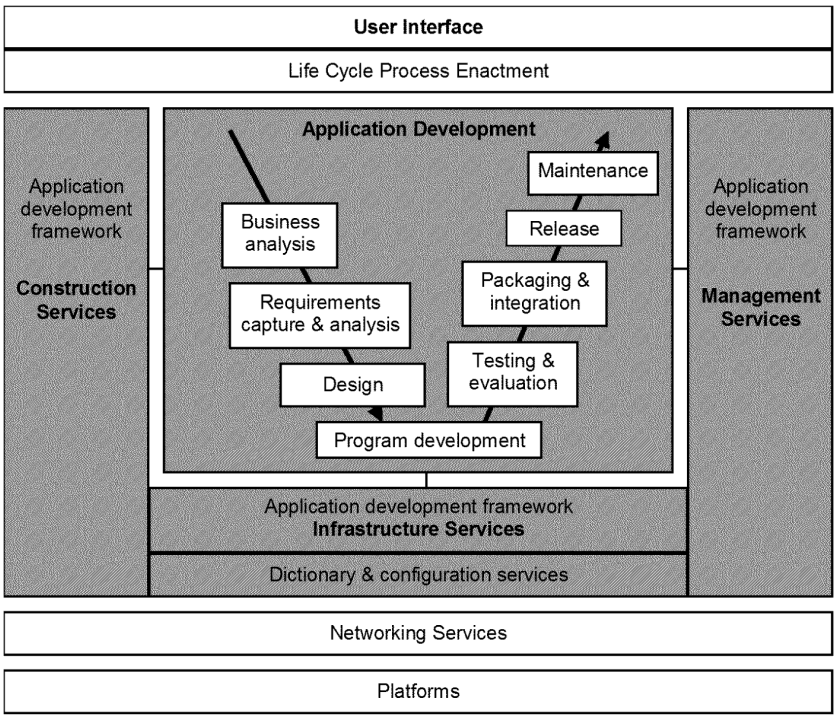


Figure 6.7 - Application development reference model

379 Most stages of the process result in documents, which are used by subsequent stages. Fujitsu's documentation is comprehensive, systematic and generally of good quality. The architecture seems complete and the system appears to be built on sound principles.

380 Must bring in Lifecycle Processes (DE/PRO/003)

- Consider using the diagram in s8.1 – but we don't want two
- Need to compare terminology and structure of the two sources

381 Consider fleshing out this table a little, especially in line with FJ feedback

382 The application development activities shown in Figure 6.3 are further described below:

Activity	Notes
Business analysis	Initial PO requirements, new business or other changes Requirements capture, analysis and specification
Design	Sophisticated structure of documentation, rigorously

FUJ00083961

- **High Level Design** - **Low Level Design** - **Platform design** - **External interface** design - **Application design**	followed through Applications are designed using object orientation, which means that modules must be self-contained, and only communicate via pre-defined and documented interfaces that are not dependent on the application's physical implementation
Construction	Visual Basic, C and C++ programming languages were used in the original Horizon system along with Oracle development tools. Third party and commodity products (such as operating systems and device drivers) are simply bought in.
Testing and evaluation	**Unit (or component) testing** takes place once the product is developed or procured and ensures that the product conforms to its requirements. **Link (or integration) testing** verifies that the product interworks with other major components. This level of testing is generally performed on a complete release. **Acceptance testing** proves to PO's client that the development meets its functional requirements and to PO that it may be brought into use without impacting on the existing solution or other applications.
Release management	A Horizon release may be either major or an incremental set of interdependent components.

Table 6.1 - System development and testing activities

383 A structure of requirements, design, development and test documentation provides a robust platform for constructing a high-quality system

384 Establish whether thorough and good quality processes were in place from the start of implementation (need to cite examples, judiciously)

385 Summarise evolution of Horizon to HNG, HNG-X and HNG-A (checking RW from Foundation report) – per HNG-X Testing Strategy – with emphasis on improvements based on experience (pragmatism, efficiency, focus)

FUJ00083961

386 HNG has built on the principles of the original lifecycle but used improved methods and technology where appropriate. For instance, the move away from Riposte and Visual Basic to Java brought in more modern development tools and enabled techniques such as object orientation as described earlier in this report. The architecture now is data driven to the extent that PO can introduce new clients without reference to Fujitsu.

6.7.4 Testing

387 Table 6.1 introduces three levels of testing:

- Component testing
- Integration testing
- Acceptance testing

388 Fujitsu carries out a risk assessment of planned changes to Horizon to determine the extent and depth of the testing required.[25]

389 The term 'regression testing' means to re-run tests to ensure that previously developed and tested software still performs correctly after a change. If not, the software would have regressed. It could therefore be argued that this type of testing is more accurately described as 'non-regression'.

390 Changes that may require regression testing include bug fixes, enhancements and configuration changes (reference data). The risk assessment is used to determine which tests to repeat.

391 Fujitsu uses automation and test management tools to improve the quality of its testing processes. Test automation enables regression to be tested more reliably and at reduced costs and elapsed time.[26]

392 Non-functional aspects (NFR), such as resilience and recovery, and security, are also tested.

393 Steve Parker (Fujitsu's SSC Manager) confirms the quality of their regression testing at paragraph 45 of his witness statement: '*I am aware of only one or two cases where a fix regressed in my time at Fujitsu.*'

394 In Legacy Horizon, a leading product called Quality Centre (QC) was used to manage test scripts. When one of the scripts failed, it was reviewed and if the failure was believed to be due to a build or software defect, the tester could use QC to create a Peak incident (described in section 6.8.3 below).

395 In section 7 below, I introduce a series of robustness measures that have been built into Horizon. Most of those rely upon functional and non-functional requirements, which are tested whenever changes are made. Therefore, the testing process also assures the quality of the countermeasure implementation.

396 Fujitsu's testing team is independent of development, support and maintenance and services – with a separate reporting line into senior management. Their job is to find bugs in Horizon and they are managed accordingly.

397 Within the current organisation[27], for instance, major change programmes are managed by a Programme Director. One of their direct reports is the Programme Test Manager.

---

25 https://en.wikipedia.org/wiki/Regression_testing

26 See HNG-X Testing Strategy TST/GEN/STG/0001. VI/STR/062 was the equivalent document for Legacy Horizon.

27 See section 6.7.1

FUJ00083961

398 Component-level testing aims to exercise every path through each component's logic. This includes checking the handling of abnormal and error conditions, as well as the 'happy path' – i.e. producing the expected outcome. Because there are many more combinations of errors and exceptions, this is where the majority of testing is directed – and this is the part of the process where user errors (UEC) are tested.

399 Unit testing was specified as part of the Low Level Designs, in line with good practice.

400 Fujitsu has created many hundreds of test reports on Horizon. I have examined a number of unit test reports, which confirm the quality of testing performed. They record, in some detail, the results of testing – including the outcomes of the individual functional and non-functional tests.

401 Study test documentation to form a view about the likelihood of 'micro-bugs' being detected (could dip into Peaks using appropriate code).

6.7.5 Adherence to processes and standards

402 Mention audits – good quality documentation - systematically created, followed and maintained

403 Refer to report outlines (including Foundation), but there may not be anything very substantial to say – so could defer to s6.8.8

404 Ideally, we need more evidence that documentation has been put into practice and used, e.g. test records.

405 Refer to s6.8.8 (also for continuous improvement 6.8.9)

6.7.6 Conclusions

406 Our summary opinion concludes that Horizon's 'robustness measures have been professionally designed, implemented and tested'

407 HNG ultimately took a practical and highly professional line

408 Opine on the rigour, absolute quality and robustness of the processes, their maturity (probably in general terms) and how they compare with what would be expected to be done by a reasonably competent supplier of this system

409 Longevity

410 Assess whether FJ used the defined processes in practice and on the results achieved

411 Relate to pleadings (not explicitly)

412 Defence

- 54(5): 'Fujitsu operates industry standard processes for developing and updating Horizon and for investigating and resolving any identified potential system errors.'
- 48(3)(a): 'Fujitsu's role included identifying and remedying coding errors and bugs in Horizon.'

413 Horizon's development lifecycle follows best practice. The Design phase is particularly stratified with separate consideration given to architecture, high and low-level design and interfaces. This approach gives confidence that important aspects of the design are described in comprehensive levels of detail. It has also resulted in a large set of documentation to be managed.

FUJ00083961

6.8 Horizon Service

414 In earlier sections of this report I have discussed the rationale for Horizon, what the system comprises and how it was built. All the claims to be considered by the court arose when the system has been in live operation supporting the business. Therefore, this section introduces how Horizon is serviced and supported.

415 More intro needed. Follow ITIL logic, may need new and fleshed out sub-sections below – but must be conservative about doing this.

6.8.1 ITIL

416 ITIL is the basis for the Horizon service model. Originating from the UK Government in the 1980s, ITIL (formerly an acronym for Information Technology Infrastructure Library) is a set of detailed practices for IT service management that focuses on aligning IT services with the needs of business. ITIL underpins ISO 20000, the International Service Management Standard for IT, although there are some differences between that standard and the ITIL framework.

417 ITIL describes processes, procedures, tasks, and checklists which are not organisation-specific or technology-specific but can be applied by any organisation to enable its IT services to deliver the value that the business requires. The following diagram illustrates the scope of ITIL, positioning most of its key processes and important functions such as the Service Desk.

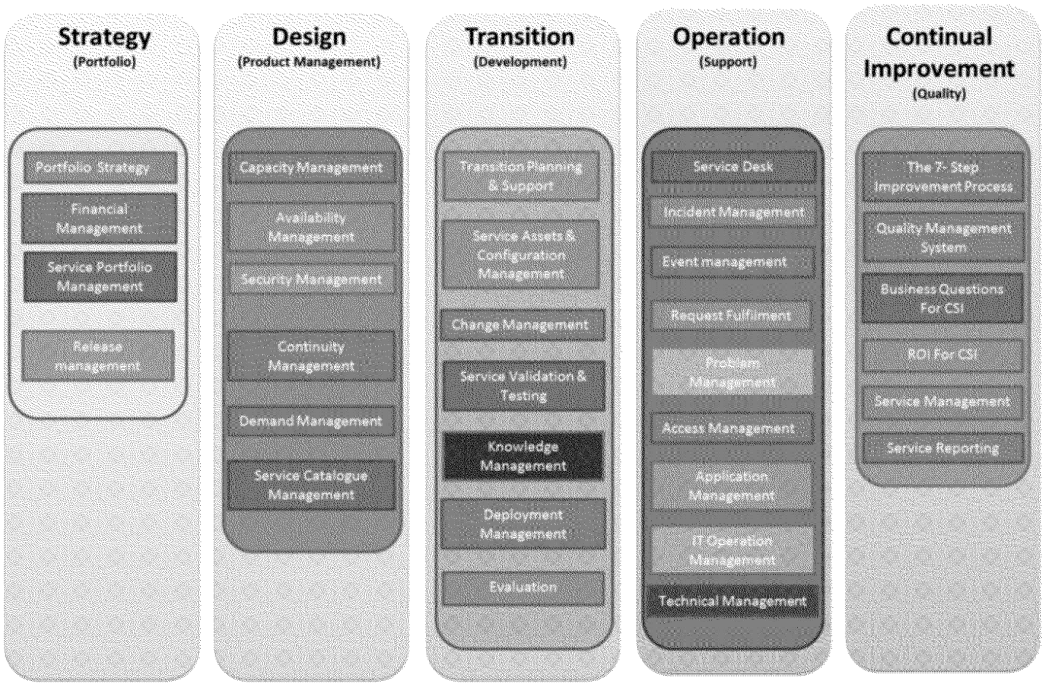


Figure 6.4 - ITIL Framework

6.8.2 Systems Management

418 The scale and complexity of the PO branch estate requires proactive and comprehensive systems management. Every branch and individual counter position is under management and is being supported in successfully performing business transactions.

FUJ00083961

419 The same applies to applications running in the data centres. Any disruption can impact large parts of the branch estate.

420 In terms of the ITIL model illustrated above, systems management falls into the Operation column under the processes of Application and IT Operation Management.

421 Systems management facilities are needed to maximise Horizon availability and to ensure that the system delivers the service levels agreed with PO. These facilities also reduce service delivery costs as follows:

- Reducing the need for human intervention, e.g. visits to branches to install software
- Automating routine management activities, thus reducing the need for human operators
- Enabling problems to be anticipated and avoided, or their impact reduced
- Managing hardware upgrades as cost effectively as possible
- Ensuring the auditing of security events, such as authentication failures and unauthorised attempts to access resources.

422 Some of the key management products used by Horizon are as follows:

- Tivoli, supplied by IBM, was used for all services on NT platforms. It provided a central event management service, software distribution and resource monitoring facilities.
- HP OpenView together with Cisco Works are used to manage network products such as routers.
- BMC Patrol is used to handle the host central servers and the Oracle applications running on them. Patrol is specifically tailored to the management of Unix systems and applications. A Patrol Tivoli Event Adapter is provided to map Patrol events onto Tivoli events. BMC can generate pager alerts if problems arise on the platforms it manages
- Maestro, also from IBM, is used to provide scheduling facilities for batch operations.

6.8.3 Horizon support services

478 From time to time most subpostmasters and their staff have questions, problems or requests related to their usage of Horizon. PO and Fujitsu have provided a set of services and tools to support users.

479 Re-visit what Coyne says and then re-visit email to RW 25-Sep (MID): cite any relevant figures I'm confident in (with evidence)

480 Horizon support was designed and is still maintained through a structured set of documentation, which defines policies, organisation, process and detailed procedures.

481 Most organisations providing technical support adopt a tiered model. PO and Fujitsu are no exceptions.[28]

> *'The support strategy expects that incidents will be raised by users and then passed through the chain of support units until a resolution can be supplied to the user. It is important that an incident starts at 1st line and then follows each stage of the chain as appropriate. This ensures:*

28 See End to End Application Support Strategy (SVM/SDM/PRO/0875)

FUJ00083961

1. *The incident is quickly defined and logged*
2. *An initial response is given*
3. *Priority is correctly evaluated*
4. *The correct skills are applied such that a resolution is supplied quickly*
5. *The call is correctly recorded, auditable and relevant metrics can be produced.*

As incidents move from left to right across the support chain they become:

- *More difficult to resolve*
- *More time consuming to resolve*
- *The training level and cost of the staff resolving the incident rises*
- *Tooling and supporting infrastructure costs rise*

Support costs and timescales for resolution increase as the incident moves to the right. Hence the effort spent "moving support to the left". Ensuring that the incident is resolved as early in the chain as possible reduces the cost and increases customer satisfaction (assuming a first time fix is achieved).'

482 As Mr Parker explains in his witness statement (paragraph 25):

'Having said that, there is often overlap of skills between adjacent lines of support and while a team may be responsible for a particular level of support, staff within that team can have skills which allow them to perform a role that is more usually performed by the next level of support.'

483 Horizon is supported by a four-level model, which is essentially a triage process:

1st line	• The Horizon Service Desk (HSD) is the branches' first point of contact for technical issues relating to the Horizon software or the hardware provided in branch. It has been operated by Atos from Manila in the Philippines since June 2014 but was previously run by Fujitsu. - deals with straightforward queries such as password issues and scheduling hardware engineers; - monitors the live estate: a System Management Centre (SMC) is run by a part of Fujitsu based in India. This team monitors Horizon system operations, taking corrective actions defined in the Horizon knowledge base whenever possible; - refers other issues to 2nd line support. • PO operates a helpdesk for operational business issues called the National Business Support Centre (NBSC). SPMs requiring assistance to determine the cause of a discrepancy contact NBSC in the first instance.
2nd line	Provided by senior members of the HSD and SMC and junior members of the SSC (Software Support Centre) - who also provide 3rd line. Note that the SSC is

FUJ00083961

	shared across Fujitsu customers. 2nd line support mainly involves searching knowledge articles based on the descriptions of issues reported by branches, gathering evidence and applying simple, well-defined work-arounds (often on the phone).
3rd line	Provided by SSC staff with a detailed knowledge of the Horizon application based on documentation and some inspection of source code. SSC use a defect management system called Peak (PinICL until 2003) to log and manage incidents passed to them which were suspected to be faults. Peak is also used to manage faults identified by testing. The SSC maintains the SSCWeb application, which includes the Known Error Log (KEL). This enables searching of system operations events and provides access to help text as well as other support and technical information. The KEL describes the symptoms of problems with some analysis of causes, potential solutions to the problems and workarounds (WOR) that might be needed before a permanent solution can be implemented.
4th line	Members of this team have specialised knowledge of specific areas of the system and are responsible for the producing permanent fixes to repair the root causes of incidents or other problems in the live application. They amend source code to fix problems. There is clearly an overlap between 4th line and the development team, which adds new features into the application.

484 Therefore, 1st and 2nd line do not identify software bugs.

485 Over the lifetime of Horizon, the organisation of support services has evolved.

Tools for managing incidents, problems and knowledge

486 Support service and its attendant systems (see my comments on JC's view of the support process)

487 Introduce key tools –TfS, Peak, KELs – summarise history (PinICL, Powerhelp)

488 a system known as TfS[29] through a gateway into Peak

489 Need to talk about history availability - limited knowledge, but some detail forwarded by JG

Incident Management process

490 ITIL defines processes for Incident Management and Problem Management, but Horizon uses its own definitions. Problem Management is addressed in section 6.8.4 below.

491 Define Incident Management here

29 TfS stands for TRIOLE for ServiceNow. This system is changing to ServiceNow.

FUJ00083961

492 Need to talk about investigation, analysis and correction of errors identified by both SPMs. Must include a description of a workaround as a countermeasure for robustness (WOR) – the overriding urgency and importance of 'keeping the show on the road' quickly, determining practical steps to avoid the problem. Give an example – search RW KEL analysis (preferably on £ selection) for 'workaround' or 'avoid'.

493 The following diagram[30] illustrates the incident and defect management process using KELs and Peaks:

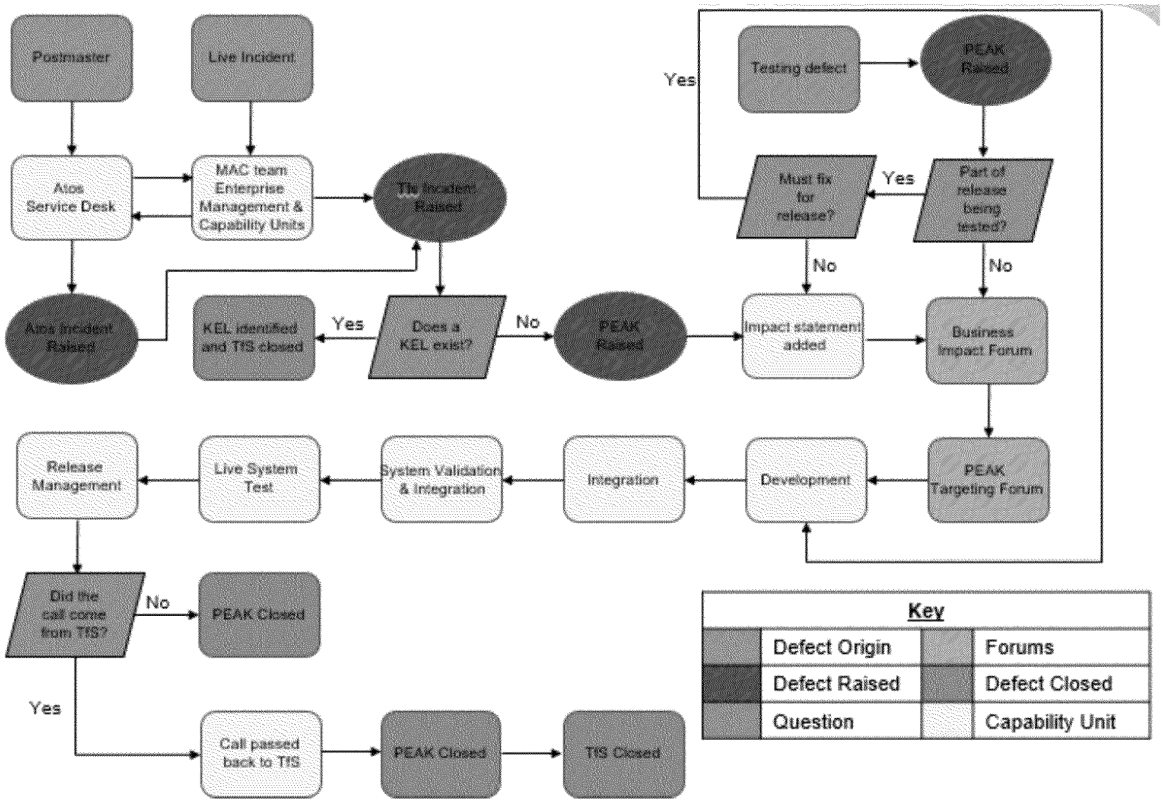


Figure 6.y – Incident and Defect Management process

494 The Fujitsu MAC (Major Account Control) team manages all changes in branches.

495 The importance of 'keeping the show on the road' is confirmed in the current End to End Application Support Strategy: '*It is incumbent on this support route to restore normal service operation as quickly as possible and minimise the adverse effect on business operations.*' [31]

Known Error Log (KEL)

496 The KEL is primarily a knowledge base, rather than a 'bug list'. The log currently comprises more than 8,000 entries.

497 KELs may be raised by testers to flag minor issues that are not resolved immediately. In exceptional circumstances, the development group has also raised KELs to inform the support teams of potential problems

30 The diagram was taken from a 2015 proposal for improving this lifecycle, which has since been implemented.

31 SVM/SDM/PRO/0875, section 1.4

FUJ00083961

coming their way. The KEL is used mainly for supporting operational users, rather than by Fujitsu's internal teams.

498 There is no defined vocabulary, which means that searching for particular terms may be unreliable. The lack of defined terms may reduce quality when new incidents and problems are recorded. This is counterbalanced using both process and procedure documentation (including standards) and categorisation wherever practical. KELs are searched and read by a wide community of support users, so their quality is actively maintained via mandatory reviews both by a manager and by a forum (KEL Approval). Check facts here.

499 Refer to appendix on sample KELs

500

501 Did some FJ forum check long-running KELs?

- See RW email Fri 21-Sep @ 17:08 (Bug date ranges)
- Look for minutes - check attendees (e.g. Anne Chambers)

502 Describe process for merging KELs and spotting common patterns (problem management?)

503 KEL deletion/archiving *could* be a weakness for FJ

Commentary

504 Perhaps defer to Conclusions (6.8.10)?

505 Commentary on thoroughness in managing calls

506 Processes are followed and are generally effective (see App D), sometimes patchy though rarely with any potential impact on branch accounts – not perfect (in the same way as the system itself), sometimes too little evidence or intermittent problem that is difficult to reproduce. Peaks and KELs may be seen as the tips of a large iceberg. They are written by experts for the benefit of other experts. Therefore, they are normally full of technical terms, acronyms and local abbreviations which sometimes make them difficult to decipher.

507 Nevertheless, in my view, KELs have been successful in helping support staff to …

6.8.4 Problem Management

508 Problem Management is an ITIL process for managing the lifecycle of *all* problems that happen in an IT service. Its primary objectives are to prevent incidents from happening and to minimise the impact of incidents that cannot be prevented.

509 The objectives of Problem Management[32] are as follows:

- To minimise disruption to the business by proactive identification and analysis of the cause of service Incidents and by managing Problems to closure
- To ensure escalation as part of a defined escalation process
- To efficiently and effectively identify the Root Cause of Incidents/Problems

32 SVM/SDM/PRO/0025

FUJ00083961

- To define Known Errors and their permanent fixes and/or temporary Workarounds, associated with all incidents
- To track and maintain relationships between Known Errors and existing incidents
- To identify Problem trends, to assess the performance of the Problem Management process and the quality of services delivered to POL.

510 'Lights out' operation at the campuses needs automated problem detection and management. Platforms generate events in response to problems. These problems are any deviations from the expected operational processing - for instance when a batch job fails, or a software error is detected.

511 Tivoli provides facilities to take events from one or more sources and use defined rules to establish whether local actions should be taken and/or whether they should be forwarded to central event servers. Event servers include functions such as the ability to correlate events, to evaluate the actions to be taken, and to schedule these actions.

512 When the automated systems are unable to handle a problem, they bring it promptly to the attention of a human operator. This person, who may be on a remote site, can manage the impact of the problem, assign it to an appropriate support team, and track processing using automated Help Desk facilities.

513 Sources of problems include applications and the platform software as well as hardware failures.

514 All access by operations staff (including 2nd and 3rd line support) to manage IT systems is audited.

515 E&Y report to 2011 identified a low-priority recommendation arising from Incident and Problem Management. Corrective actions were agreed. [see reading notes for further details]. Comment on the associated risks. NB: This is just one example

516 Discuss the likelihood of this process detecting bugs, assessing the impact, implementing correction, etc.

517 The triage process seems effective – problems find their way to a small core of knowledgeable and capable individuals such as Anne Chambers – and Mark Wright?

518 Check JC ref (POL-0152874)

519 Re-visit Major Incident Process and review sample reports

6.8.5 Reference Data Management

520 An outline of the process and controls with opinion. This can be brief, because the main issue is addressed in section 11.

521 Recap Foundation treatment [ok]

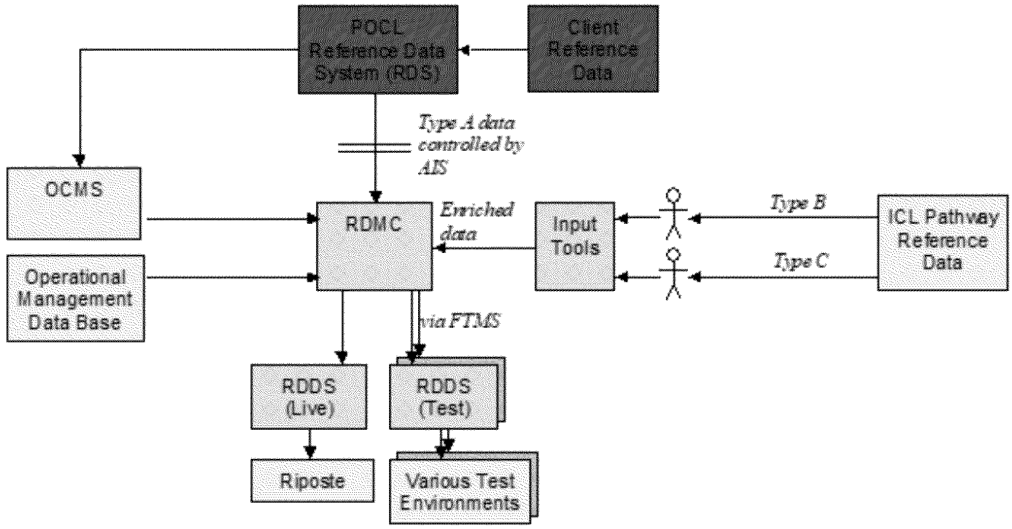
522 RDMS stands for Reference Data Management System; RDMC: Reference Data Management Centre

523 RDDS: Reference Data Delivery Service. All this is part of Horizon (per TD/ARC/001 colour scheme).

524 Diagram below[33] – is all about distribution and management. The ref data itself is controlled by PO outside of Horizon.

33 From TD/ARC/001, section 5.3.3.2

FUJ00083961



525

526 Read Coyne (with footnoted refs) – see following paras, own research (including in audit reports on management controls

527 The management and operations with regard to Reference Data has been outsourced from within Post Office control to a sub-contractor (ATOS) since June 2014. Post Office's Reference Data Management Centre (RDMC) supports the loading, storage and release of Reference Data within the Horizon system. The Reference Data Distribution Service (RDDS) distributes Reference Data to Post Office branches and other data centre systems. The POL Reference Data Team is a team dedicated to delivery of Reference Data and verification of operational business change through Reference Data.

528 Despite the criticality of the integrity of Reference Data, a document from July 2017 suggests that changes to Reference Data were not subject to any appropriate change control process. The document - Operations Board 21 July 2017.pdf, [POL-0221328] - reports; "… we have now aligned that all Reference Data changes go through the appropriate change process". This is one isolated mention.

529 This is consistent with the position that prior to July 2017 Reference Data could be changed without any formal consideration as to what the impact might be. What is the evidence?

530 Ref data is diverse – e.g. determining contents of a report, product prices and exchange rates (types summarised in TD/ARC/001)

531 The effects of changing it are also diverse – ranging from the cosmetic (layout of a screen) to more significant (with an impact on branch accounts)

532 Maintenance may require remote access [find ref]

533 Quality control sounds like a FJ weakness, scan related KELs – probably be critical here

534 QCC robustness countermeasure against errors in ref data

535 Commentary, under Robustness in section 7.6 Documents Cited by the Claimants, on audit findings.

FUJ00083961

6.8.6 Change Management

536 Normally applied to business change – communication, training, implementation strategies. So be clear what is being discussed here:

- documents
- hardware and networks
- software - applications and platform (check terminology used in Foundation)
- data – system data, user transactions and other data, reference data used to configure the system
- service management and operational processes.

537 Change management was designed to comply with the ISO/IEC 20000-1:2005 objectives, e.g. "To ensure all changes are assessed, approved, implemented and reviewed in a controlled manner"

538 MSC/OCP/OCR (mentioned in s11)

539 RFI: corrections would have been done with the authorisation of Post Office via the method in use at

540 the time, e.g. OCPs

541

542 Give any stats showing beyond doubt that this process is used and managed

543 E&Y report to 2011 (7 years ago) identified that certain unauthorised users were allowed to deploy changes without authorisation (user admin problem) and insufficient evidence that POL was involved in testing and approving the deployment of changes. Corrective actions were agreed. [see research notes for further details]. Comment on the associated risks.

6.8.7 Software Distribution

544 Horizon software runs on more than 10,000 platforms – including servers in data centres as well as all Post Office branches. Systems on this scale require some degree of automated software distribution.

545 Horizon software to be distributed to target systems is delivered (with its release notes) through a formal release management mechanism. The software is pre-packaged so that it can be delivered and optionally installed in a fully automated manner. Where such automation is not possible, the necessary manual interventions are clearly documented.

546 Reference data updates are deployed in a fully automated manner using the targeting information provided along with the data itself.

6.8.8 Adherence to processes [and standards]

547 We need to explain how the tools described above have been used to provide detailed evidence (give numbers) of processes being followed

548 ITIL …

549 Audits – recommendations and responses (can only aggregate and sample, not study in depth)

550 In-service history

FUJ00083961

551 How to analyse and summarise that?

552 What is relevant and important?

6.8.9 Continuous improvement

553 The company seeks to improve its capabilities continually[34].

554 Use Post Office Account Business Management Policy (PA/POL/002) – see Research Notes

555 Could be useful in other places too, e.g. on preventative

556 Also covers 7.7 (dev/test): HNG-X more sophisticated than HNG

557 Arises from internal and external reviews and audits, lessons learned (S90 example, but search for others)

558 Provide specific evidence, e.g. processes of review and improvement (S60 Lessons Learnt for one)

559 Also, evident in day-to-day operational process refinements

560 Bring in maturity (cf. CMMI)?

6.8.10 Conclusions

561 These now need a different emphasis. Relate to pleadings (not explicitly). Look for specific claims against these processes (in pleadings).

562 Good quality documentation, systematically ... – see above. Regular revision and improvement enhance quality.

563 Good processes, rigorously used, will deal with anomalies from start to finish – give more detail

564 How does Horizon compare with similar systems? Think of examples (consider my own experience).

565 Acknowledge specific challenges – but they must be clearly defined, e.g. the numbers of calls, user errors, training, large system to maintain, acknowledge that people do not always follow processes and may not succeed in fully dealing with any bugs discovered or in testing properly – but …

566 A robust framework of management control and QA – appropriate for such a contract, best practice, etc

567 Pick up points from WBD letter to Freeths, 28-Jul-2016, Schedule 6, para 1.3:

> *'The important issue is not, therefore, whether these defects exist, as they likely do, but whether there are in place adequate controls to identify defects and take any necessary remedial action in order to avoid harm to branches. We have explained the importance of postmasters monitoring their own accounts and raising any issues. Furthermore, changes to Horizon are controlled through a robust change management and operational governance process including a joint Fujitsu and Post Office test team. Horizon is also regularly subjected to comprehensive and independent reviews, testing and audit procedures [Schedule 4, clause 4.2]. Beyond these formal reviews, there is also regular interaction between technical teams at Post Office and Fujitsu.'*

568 Bring forward and summarise from earlier sub-sections (of 6.8), where relevant

569 I am confident that the Horizon triage process ensures that only a 'tiny' proportion of software errors affecting branch accounts can persist - cf. analysis in earlier sections

570 Problem management (use ITIL definition, but check what was said in FR)

---

34 The term 'continuous improvement' is more commonly used. In the late 1990's, ISO changed their usage from 'continuous' to 'continual' (see https://en.wikipedia.org/wiki/Continual_improvement_process#ISO_change_from_"continuous"_to_"continual"). This report uses both terms.

FUJ00083961

571 KELs, Peaks and TfS entries include detailed records of status and history

572 Effectiveness of capturing and processing of all support calls. How thorough are the support processes? What is the likelihood that the KELs cover a large proportion, of all bugs?

573 KELs are used effectively in Horizon – most of the most important artefacts (use this?) in the support process. Knowledge management (check definition) is often talked about, but not always done.

574 ITIL is widely used across the IT industry, because it provides a framework based on decades of experience in managing systems in service. Horizon system management, based on the ITIL framework, is supported by a series of market-leading tools, which enable Fujitsu to manage the full range of resources.

575 Based on what I have learned so far, Horizon support services seem fragmented (this is not uncommon, but …). They are provided by some combination of the NBSC in Manila, the SMC in India and the SSC in the UK. Those teams are supported by Peak and KEL and possibly other systems too.

576 I am aware of indications that help services have not been consistently well received. This area may emerge as a weak link in an otherwise well managed IT service.

FUJ00083961

7. EXPERT ISSUES – ROBUSTNESS OF HORIZON

7.1 Issues Addressed in this Section

577 This section of my report addresses the Horizon issues in our Group 1. These are Horizon Issues 3, 4, and 6, which concern robustness of Horizon. These issues are:

578 **Issue 3**: To what extent and in what respects is the Horizon System "robust" and extremely unlikely to be the cause of shortfalls in branches?

579 **Issue 4**: To what extent has there been potential for errors in data recorded within Horizon to arise in (a) data entry, (b) transfer or (c) processing of data in Horizon?

580 **Issue 6**: To what extent did measures and/or controls that existed in Horizon prevent, detect, identify, report or reduce to an extremely low level the risk of the following:

a) data entry errors;

a) data packet or system level errors (including data processing, effecting, and recording the same);

b) a failure to detect, correct and remedy software coding errors or bugs;

c) errors in the transmission, replication and storage of transaction record data; and

d) the data stored in the central data centre not being an accurate record of transactions entered on branch terminals?

7.2 Robustness of Horizon: My Opinion

581 I here summarise my opinion on Horizon issue 3.

a) My opinion from the evidence is that at all times for which there are KELs - which is nearly all the lifetime of the system - Horizon has been a very robust system, compared to other major systems I have worked on in sectors such as banking, retail, telecomms, government, and healthcare.

b) I have described 17 types of robustness countermeasure, which I have applied routinely on projects over many years. In my opinion, Fujitsu have applied these countermeasures effectively in building and supporting Horizon.

c) As an accounting system, Horizon particularly needs countermeasures to ensure the accuracy of the accounts, in the face of many types of adverse event. I have focused particularly on these countermeasures and their effectiveness. In my opinion, these countermeasures are well designed, and have been effective in preventing errors in accounts. Very few adverse events - including user errors and software bugs - have evaded all the countermeasures to the extent of causing significant inaccuracies in branch accounts. Horizon is very unlikely to cause significant shortfalls in branches.

582 This summary of my opinions applies also to the Horizon issues 4 and 6, which in my opinion address some special cases of issue 3.

583 The experts have agreed that robustness is not a matter of perfection, or the complete absence of bugs. Horizon, in common with all large commercial IT systems, was not completely free of bugs.

FUJ00083961

584 Robustness involves the use of a set of techniques, which I call countermeasures, to ensure that many kinds of potentially harmful events (including hardware failures, communications failures, user errors and software bugs) do not have harmful consequences - or if they do, the harmful consequences are minimised.

585 Robustness is a core requirement for any major commercial IT system, and has been so for many years. Large parts of IT project budgets are spent ensuring that systems are robust.

586 The techniques for achieving robustness are so important that they have become a well established and central part of commercial IT practice. I have listed 17 major techniques, or types of countermeasure, which I have routinely applied on major projects over thirty years. These techniques act in concert to minimise harmful effects.

587 To make Horizon robust, Fujitsu had to effectively apply these established techniques. I have examined the evidence of how Fujitsu designed and built Horizon, how they tested it (in section 6.7), and how they supported it (in section 6.8). The 8390 KELs, in particular, are a rich source of evidence about Horizon in service - about events which threatened to have harmful consequences, and how well or badly the robustness countermeasures acted in those cases. My analysis of many KELs persuades me that the countermeasures in Horizon worked well in the live use of Horizon..

588 In some market sectors where I have worked - such as banking, telecomms, and healthcare - robustness is often compromised by the presence of very old legacy software (sometimes coming from merged organisations) which is hard to maintain or adapt - and has resulted in an over-complex frozen 'spaghetti' architectures. Horizon does not suffer from these problems. Horizon was a 'green fields' development started in 1996 - essentially unencumbered by any IT legacy. Therefore it was much easier to build a robust architecture from the start.

589 In his report, Mr. Coyne has not described any robustness countermeasures, or assessed how well they were applied in Horizon. In my opinion, robustness of Horizon is the central expert issue of this case, because it is robustness - rather than the number of bugs - which determines the financial impact of bugs on claimants' accounts. It would be helpful for the experts to share ideas about robustness countermeasures. I invite Mr. Coyne to meet soon on a without prejudice basis, with the intention of drawing up an agreed list of countermeasures, and seeking agreement on how well or badly they were applied.

7.3 Horizon Issue 3

590 I first address Horizon issue 3: To what extent and in what respects is the Horizon System "robust" and extremely unlikely to be the cause of shortfalls in branches?

591 Issue 3 overlaps with issue 1, because 'unlikely to be the cause of shortfalls in branches' overlaps with the issue of bugs, errors or defects which 'cause apparent or alleged discrepancies or shortfalls' (Issue 1).

592 However, for clarity of presentation and description, I shall postpone the discussion of the 'unlikely to be the cause of shortfalls' aspect of issue 3 until the next section of the report, which is about Horizon issue 1. This is for two reasons:

FUJ00083961

- As I shall describe in section 8, the extent of shortfalls caused by Horizon depends on the robustness of Horizon, and on a range of well-known robustness countermeasures as implemented in Horizon. So, describing those countermeasures, and how successfully they have been implemented, as will be done in this section, is an essential pre-requisite for addressing Horizon Issue 1 in section 8. Mixing the two issues together would lead to a confusing presentation.
- Robustness involves preventing or minimising many harmful effects, as well as shortfalls.

593 In their outline of August 17 2018, the claimants say that they do not understand any 'objective meaning' of the term 'robust', and imply that it may be an IT marketing term, used just for public relations.

594 In my opinion, this is not so.

595 In large IT projects, robustness (and the almost synonymous term, resilience) are very important design objectives. Large parts of project budgets are devoted to achieving them.

596 The term 'robust' receives its meaning from the phrase 'robust against...[some risk or threat]' and there are many threats which business IT systems need to be robust against. Horizon needs to be robust against hardware failures, communications failures, power cuts, user errors, disasters, fraud, and hacker attacks - to name just some of the high-level threats, which can then be further subdivided. For instance, communication failures can be subdivided into a simple failure to communicate, and the communication of erroneous information. Robustness of IT systems is a large and mature topic.

597 In software engineering there is a well-established practice and terminology for discussing robustness under the heading of 'risk management', within which the extent of robustness, as in Horizon issue 3, can be addressed. The extent of the robustness of Horizon consists of:

- The list of risks and threats that Horizon needs to be robust against
- For each risk, some measure of the probability of it occurring (which may for instance be measured on a scale of red/amber/green, in typical risk management methodologies)
- For each risk, the seriousness of the consequences both before it is not handled, and after risk management measures have been put in place (also possibly on a scale or red/amber/green)

598 These risk management methodologies are a part of the Software Engineering Process Maturity Model which Fujitsu used to audit and assess their development processes - as was described in section 6.7.

599 My emphasis in this section will not be to discuss the dimensions of threat or risk that robustness is designed to counter, but to discuss the many different countermeasures which have been developed in the IT industry to counter these risks, and how well or otherwise Fujitsu have applied these countermeasures.

600 In practice there is no simple relationship between risks and countermeasures. Typically the effects of any type of risk are to be mitigated by several different types of countermeasure, acting together in different ways.

601 I introduced the various types of robustness countermeasure in a table in the summary of my opinions at section 2 of this report and defined a three-letter acronym for each class of countermeasure. Typical of these acronyms are RDS (Redundant Data Storage) and UEC (User Error Correction).

FUJ00083961

602 In sections 4-6 of this report, I gave further details about how the different countermeasures have been built into the architecture of Horizon, and how they have been tested.

603 It is important to understand the robustness does not mean 'be perfect' - as that is impossible. It means: 'manage the risks of imperfection so they are acceptable'.

604 Countermeasures typically work by there being many different countermeasures acting together - so that even if one countermeasure is not fully implemented, and does not catch and counter some risk, other countermeasures will act as extra lines of defence - so that very few threats get past all the lines of defence.

605 Therefore, as was agreed in the joint expert statement of 04 September 2018, robustness does not mean perfection. It means recognising that imperfections exist, and adopting countermeasures (typically, many of them) to minimise the likely harmful impact of those imperfections. Here, imperfections can include external factors such as user errors or power failures, or internal factors such as errors in software. They may include errors in the countermeasures themselves.

606 A familiar example may illustrate this. Many people create and edit documents using Microsoft Word. This product is not perfect. It may freeze up for no good reason, or it may allow its users to work for hours without saving their work - and so be vulnerable to hardware failures or power cuts. But over the years, it has become more robust, so that the costs of these failures are now usually acceptable. Whatever goes wrong, there is usually a recent 'auto-saved' version of your document, so you do not lose too much of your work. That is practical robustness, not perfection.

607 The robustness of an IT system can be understood through a biological analogy, to the immune system. Just as there are many risks and threats to an IT system, so there are many risks and threats to the human body. The body cannot be designed as if there were no such things as bacteria, viruses, injuries, poisons, genetic defects and so on; so the immune system is part of a multi-layered defence system against these threats. The robustness of the immune system depends on its being multi-layered; if a first layer of defence, such as white blood cells, does not deal with the threat, then an increasingly complex set of defences, including T cells, B cells and so on, is mobilised until the threat is dealt with. Running a fever is not a sign of perfection; but it is a sign that the body is defending itself robustly.

608 Similarly, it is easy to point to evidence such as KELs which imply that Horizon was not perfect; but perfection in an IT system is no more possible than a lifetime of perfect health. Often the same KELs describe how the threat was addressed in Horizon, through a multi-layered defence system which includes robustness of the platform software, robustness of the IT architecture, error correction measures such as reconciliation, and investigation of remaining anomalies through the four levels of IT support. Robustness depends on these many layers. A failure of Horizon's robustness would mean the failure of many layers of defence.

7.4 Countermeasures to Achieve Robustness

609 This section contains a brief survey of all the different types of countermeasure which are commonly used to achieve robustness. Because these countermeasures have been described earlier in the report, the description takes the form of a table, with references to the sections of the report which describe the countermeasure. [this

FUJ00083961

will be the master version of this table, to be reordered, updated and eventually copied to the summary of opinions].

610 For each countermeasure, I define a three-letter acronym for easy reference through the report. [These are not yet in the best order.]

No.	Countermeasure	Explanation and examples	Described in Section
1	Reliable and redundant hardware (RHW)	Redundancy guards against many types of hardware failure. Examples: RAID discs, disaster recovery sites. Software is designed in many ways to be robust against hardware failures	4.2, 5.4, 6.5
2	Robust data communication and replication (ROC)	Communication systems and protocols are designed to recover from and protect against many kinds of communication failure. Examples: TCP/IP, Riposte	4.2, 5.3, 6.5
3	Transactional Integrity and database recovery (TIN)	Database management systems provide many facilities so that numerous kinds of failure cannot leave the data in an inconsistent, unusable state, or lose any data that have been previously stored	4.2, 4.3, 5.4, 6.2, 6.5
4	Defensive programming (DEP)	Software is divided into small self-contained modules, which do not assume that other modules are correct, but defend themselves by checking their inputs and raising alerts early	5.3, 6.2, 6.5
5	Generic, data driven software (DDS)	Different use cases for software often have much in common. Software is written generically to be able to handle the different cases, using reference data to define which use case is to be handled. Example: variations in PO client products handled by reference data.	4.2, 4.3, 5.3, 6.2
6	Secure kernel hardware and software (SEK)	When a large complex IT system is subject to threats, the design may include a small, well tested and secure kernel which is proof against those threats. Examples: secure kernels of operating systems, Horizon core audit process	4.3, 4.4, 5.4, 6.2
7	Redundant data storage and computing, with cross-checks (RDS)	In large IT systems and sets of systems, data are stored redundantly in several places, and routine operations check automatically that the different copies of the data remain consistent.	4.3, 4.4, 5.4, 6.2, 6.3, 6.5
8	Double entry accounting (DEA)	Accounting systems operate by the principles of double entry book keeping, so that any change to the accounts must be made in a transaction whose summed effect on all accounts is zero. Transactions which do not obey this constraint are rejected.	4.2, 5.3, 5.4, 6.1, 6.2, 6.3
9	Early detection of user errors (DUE)	At the point of user input, as many checks as possible are made of the correctness of the input - so that the system will not accept erroneous input and may warn the user of errors.	5.3, 6.1
10	Later correction of user errors (UEC)	In accounting systems, the system's version of reality is periodically checked against external versions of reality and corrected if wrong. Examples: cash balancing and rollover, reconciliation and TCs.	4.2, 4.3, 6.1, 6.2, 6.4
11	Manual workarounds (WOR)	Whenever any part of Horizon does not work as required, there may be potential to define and apply manual workarounds.	-
12	Testing good practice (TGP)	The purpose of system testing is not to prove that the system is correct, but to prove that it is incorrect in any way possible. Examples: regression testing, user testing, testing edge cases.	6.7

FUJ00083961

13	Manual Inspection of data (MID)	Any large business IT system is used by many people, who view its outputs and check them against each other for consistency, and against their own knowledge of the business. SPMs , watching their branch accounts, were a key component of this.	4.3, 6.2, 6.3
14	Bug Finding and Correction (BFC)	Whenever the system shows any anomalous behaviour, that is investigated, its causes found and corrected. Interim workarounds are deployed. Extra checks may be added to ensure that other similar threats are handled correctly.	-
15	Large scale IT architecture (ARC)	In any large IT estate, principles of IT architecture are used to achieve robustness - such as using a distributed network of loosely coupled sub-systems with clearly distinguished functions. The sub-systems are built to well-defined standards with clear interfaces.	4.1, 5.3, 5.4, 6.2, 6.5
16	Quality and change Control (QCC)	CE to fill in	-
17	Managing non-functional requirements (NFR)	Robustness is improved by paying close attention to non-functional requirements and the associated 'ilities' such as manageability, supportability, maintainability and adaptability	-
18	Security (SEC)	CE to add	

611 These different types of robustness countermeasure have been described and introduced in sections 4-6 of this report, and I will not repeat the descriptions here. References to the relevant sections for each type of countermeasure will be given in the table above.

7.5 My Experience of Robustness Countermeasures

612 As I have described above, the robustness countermeasures are so essential to large commercial IT systems, that they have become a routine part of large IT projects. I have been applying them on such projects since around 1979. I note here some of the countermeasures which are particularly familiar to me.

613 TIN: I designed and developed one of the first relational database management systems (DBMS), and managed its commercial exploitation by my company Logica for several years. During that time, transactional integrity and recovery (TIN) became an essential requirement for our clients. I personally designed the transactional integrity facilities of our DBMS, and was involved in configuring the product for many commercial clients. Since then, TIN has been a routine foundation of most of the software I have been involved with.

614 DEA, RDS, MID, and UEC: I have been involved with computerised accounting systems in several roles both as an architect (for instance, in deployments of SAP) and as a user, when I held line management positions in Logica. In those roles I spent much of my time scrutinising management accounting data (MID), checking it against other data sources (RDS), and ensuring it was an accurate reflection of reality (UEC). All of these were underpinned by the principles of double entry accounting (DEA)

615 RHW, ROC: Particularly when working in the finance sector (investment banking and retail banking), the projects I have worked on have depended on redundant and reliable hardware (RHW), and have had to be robust against all kinds of communication failure (ROC).

FUJ00083961

616 ARC, NFR, QCC: I have acted many times in major projects where large scale IT architectures (ARC) and non functional requirements (NFR) have figured centrally in the customer's requirements. Often these projects have involved multiple releases of software, requiring mature techniques for change control (QCC)

617 WOR: In all IT projects, there has to be some consideration of manual 'last lines of defence' (WOR). IT disputes often involve cases where these defences were tested, and failed. I have been involved in investigating a major retail banking failure of this nature, and another such failure in government computing.

618 DEP, DDS, DUE, TGP. BFC: I continue to be involved in implementation projects in healthcare IT, developing and testing software in Java and other languages. In these projects, defensive programming techniques (DEP), data-driven software (DDS) and robust, error-proof user interfaces (DUE) are essential - as are thorough testing (TGP) and bug-fixing (BFC) to ensure patient safety.

619 I have not had much personal involvement in building secure kernel software (SEK), or computer security techniques (SEC) - although I am familiar with the underlying mathematical specification methods.

7.6 Effect of Countermeasures on Bugs Which Might Affect Branch Accounts

620 In this section I discuss how various countermeasures acted specifically on any bugs in Horizon which might have had some effect on branch accounts. The discussion of this section is partly based on the KELs which I have sampled. One purpose of this discussion is to state and justify some assumptions I shall make when addressing Horizon Issue 1, on the extent of the impact of bugs.

621 In surveys I have made of KELs, I have observed:

i. Many KELs are not about bugs in Horizon. In those cases, Horizon is acting as intended, and the KEL exists to give appropriate advice to an SPM in specific circumstances.

ii. Of the KELs which are related to faults in Horizon, many of those faults self-evidently have no effect on branch accounts; for instance, they may just be about some inconvenience for an SPM, or about some back-end reporting issue.

iii. Of the remaining KELs, which describe bugs with some potential for impact on branch accounts, in many cases it can be easily inferred that some countermeasure would prevent any actual impact on branch accounts. The main countermeasures involved, and how they prevent financial impact, are described below. When some countermeasure would avoid financial impact, sometimes this is stated explicitly in the KEL, but often it is not. I assume this is because KELs were written by people deeply familiar with Horizon, who expected their readers to understand that there would be no impact (as noted at paragraph 66 of Mr. Parker's witness statement).

622 The countermeasures which most frequently prevent any long-term financial impact of bugs in class (iii) are:

- TIN: transactional integrity means that some customer transaction either succeeds in its entirety, or in the case of some error, has no effect on the branch database - as if the transaction had never been started.
- UEC: of any software bug has the same effect as a user error (for instance, as an error in recovering a recoverable transaction, or in cash management) then the normal measures for correction of user errors

FUJ00083961

(mainly, Transaction corrections, and monthly balancing) will correct the error , just as they correct the many user errors which occur.

- RDS/MID: Many errors produce anomalies which are visible to the SPM (by his inspection of system behaviour, MID) - either immediately, or later in monthly balancing. Once he has reported it, Fujitsu have many ways to look at system logs and other redundantly stored data (RDS), to establish what happened and ensure there is no adverse impact on branch accounts.
- WOR: Once they were alerted to any condition which caused difficulty for the SPM, the various levels of support appear to have been good at finding workarounds for either the SPM or the back office to apply, to ensure there were no harmful effects.

623 When all these KELs are removed from consideration, very few remain - where there is a possible bug in Horizon, and it is not obvious (for one of the reasons above) that it will have no impact on branch accounts . I have only found a very small number of these KELs. Mr Coyne has examined over 5000 KELs, and for none of them has he presented the analysis which would be needed to show that a KEL is in this group - for instance, to show that it is not in groups (ii) or (iii), for instance because of countermeasures.

624 One can classify bugs in Horizon, with possible impact on branch accounts, into the following groups:

a) Bugs whose effect is immediately evident to the clerk in the branch - for instance, because it prevents him from doing something he needs to do to serve a customer.

e) Bugs whose effect is only visible to the SPM when he does monthly balancing and rollover - the bug causing a discrepancy which he can notice and investigate.

f) Bugs which are never visible, either to the SPM or centrally, but which nevertheless affect his branch accounts.

625 In my opinion, there are no bugs of class (c). I shall explain why this is so. Obviously one cannot deduce it from examining KELs, because if a bug were totally invisible, it would never cause calls to a help desk, so might never trigger creation of a KEL.

626 It is a central principle of Horizon that the Core Audit Database acts as a secure 'gold standard' for branch accounts (countermeasure SEK) and that the audit record can only contain events which originated at the counter - either in customer transactions or monthly balancing. Therefore it is impossible for any bug to influence the branch accounts without showing itself at the branch, either in customer transactions (class(a)) or in monthly balancing (class(c)), so it is put into the audit database.

627 The same conclusion is also implied by different considerations.

628 Suppose there was some Horizon bug, not visible to the SPM at any time, which caused money to leak out of his branch accounts. Because PO accounts are kept in POLSAP, which uses double entry accounting, any money which leaks out of branch accounts must show up in some other account. Setting aside the possibility that losses in one branch show up as gains for another, this other account must be some PO central account, which aggregates the financial impact across all or many branches.

FUJ00083961

629 Because many branches are aggregated in this central account, even small amounts in single branches will add up to large amounts in this central account - amounts up to 11,000 times larger. There are PO managers whose role it is to monitor these accounts and understand them. Their actions and decisions must be annually inspected and reviewed by external auditors. It seems to me unlikely that the aggregate financial impact of any bug , which was somehow silent in the branches, would not be spotted by PO central managers or auditors, or both (unless its impact in every branch was very small indeed). This combination of the countermeasures DEA and MID guarantees that there can be no silent bugs with significant financial impact.

630 There are therefore only classes (a) and (b) of bugs with possible financial impact.

631 In class (a) - bugs whose effects are immediately evident to the clerk - there are many KELs which record events which puzzled the clerk, even if their financial impact was small or zero. Therefore I assume that any such bug would be reported in some non-zero proportion of its occurrences, however small its financial impact. Even if the SPM does not report it immediately, he or his assistant notices it at the time; so if there is a later anomaly in his monthly balancing, he may relate the two and then call the help desk.

632 For class (b) - bugs whose impact is only evident at monthly balancing - there is unfortunately a paucity of evidence in KELs. The purpose of KELs is not to record the branches affected by a bug, or the amounts involved; nor is that the purpose of any related PEAK. Such information is expected to be recorded elsewhere.

633 There is relevant evidence in one case where we have a fuller analysis - the local suspense account bug. Here, an analysis by Gareth Jenkins states that 16 branches were affected, with discrepancies of the following amounts:

Branch	Name	Amount
002647	Aberystwyth	-£6.71
002840	Inverness	£140.61
010007	East Dulwich	-£0.01
011458	Willen Village	-£9,799.88
012004	Lower Edmonton	£16.12
054011	Lower Regent Street	£3.34
101832	Dundas	£5.84
104937	Grange	£0.03
104937	Grange	-£49.65
155025	Hounslow	-£113.14
156715	Gilford	£11.55
211844	Rosyth Terminus	£36.20
211844	Rosyth Terminus	-£77.97
243242	Wardles Lane	-£0.51
266418	Bowness Road	£3,186.70
297611	Merthyr Dyfan	£160.92

634 Only the two branches with the largest discrepancy reported it. This allows me to conclude tentatively that:

- if a branch sees a discrepancy in monthly balancing of £3000 or more, the SPM is very likely to report it
- If the discrepancy is between £100 and £200 (three cases, none reported) , the likelihood of the SPM reporting it is probably less about than 20% (because $0.8^3 = 0.5$; if the probability of reporting is 20%, the probability of not reporting on any one occasion is 0.8)

FUJ00083961

- If the discrepancy is less than £50 (10 cases, none reported) the likelihood of the SPM reporting it is less than about 5%

635 I can go a little further than this by making weak inferences about how a manager of a small business, such as an SPM, needs to prioritise his time in monthly balancing, and other evidence. The witness statement of Angela Van Den Bogerd at paragraph 187 says 'Generally, when discrepancies are of a value of several hundreds of pounds, I would expect Subpostmasters to contact NBSC.' I assume the following, as best assumptions of SPM behaviour in reporting anomalies in their monthly balancing:

a) If a discrepancy is £1000 or more , the SPM probably needs to investigate it. If he cannot find the cause in his branch, the likelihood of his reporting it through a help line is 80%

b) If a discrepancy is of the order of £300, 30% of SPMs will report it

g) If a discrepancy is of the order of £100, 10% of SPMs will report it

h) For a discrepancy of £10 or less, it is usually not worth the SPM's time to investigate it (because errors in counting cash or stock are often larger than this); so these are reported on less than 1% of occasions.

636 In this respect , I also note that a bug which causes some mean discrepancy (for illustration, £300) causes a range of different discrepancies on different occasions. Therefore it may sometimes cause discrepancies much larger than £300, which are more likely to be reported.

637 There is an important exception to these expectations. If an SPM knows that there is already some large discrepancy in his accounts (for instance, if he had a larger discrepancy in his previous month's accounts, and has no reason to expect it to have gone away) then he is less likely to notice or report smaller discrepancies. The last category (d), which do not reveal themselves immediately to the clerk, and whose financial impact at month-end is of the order of £10 or less, I shall refer to as 'micro bugs'. As was described above, even these micro-bugs, through Double Entry Accounting, are likely to produce large aggregated effects, which would be noticed in some central PO account, or in an audit of the accounts. I shall discuss micro-bugs further in section 8.

638 Apart from the micro bugs, the analysis of this sub-section shows that the SPMs themselves acted as an important countermeasure against bugs impacting their accounts, through their own inspections of Horizon and its data (MID).

639 I emphasise that these are tentative assumptions, based on the evidence above , and on my understanding of how the manager of a small business might prioritise his time. They will enter in my analysis of the financial impact of bugs, in section 8; but before they are used in any calculation, I shall move them in a conservative direction, to make the result more reliable and more favourable to the claimants.

640 Once any anomaly is reported to a help desk, PO and Fujitsu had processes to triage it, and to create a KEL if there was any need to advise branches how to handle a problem, or to take other action in the back office.

641 From my analysis of Fujitsu's support structures and processes in section 6.7, I have found these processes to be fairly efficient. Specifically, Mr. Parker's witness statement at paragraph 62.8 states that if any anomaly was thought to have the potential to influence branch accounts, it was treated as high priority. My assumption is

FUJ00083961

therefore that these processes resulted in a KEL on more than 90% of the occasions where it was reported and there might be some effect on branch accounts. Once again, I shall move this assumption in a conservative direction, to favour the claimants, before using it in any calculation. If there is any reason to alter it, I shall do so and redo the calculations.

642 Mr Parker's witness statement states at paragraph 61.9 that about 15% of KELs have been archived, and are not in the set of 8390 KELs supplied to myself and Mr. Coyne. They can be recovered with some effort. I have asked to examine some of these archived KELs, to ascertain if there is anything special about them. Meanwhile, I shall account for them statistically in any numerical calculations I make, as KELs I am unable to examine, just like the other KELs I have not examined for lack of time.

643 Taken together, these points mean that the KELs form a reliable source of evidence about Fujitsu's performance on bug-finding and correction. The factors I have noted above (SPM reporting behaviour, and Fujitsu's creation of KELs) mean that I can sample the KELs and their Peaks to see how well Fujitsu diagnosed and fixed bugs (this includes all bugs, including those that affected branch accounts).

644 My sampling of KELs implies:

- Across all bugs, Fujitsu were generally able rapidly to identify the cause and any fix required
- A small proportion of bugs could not be reproduced in testing, and remained perplexing. In my experience, this is to be expected in any complex system.
- Once they had identified the cause, Fujitsu were generally able to identify all occurrences of the bug in system logs - and often to suggest workarounds while it was being fixed
- The speed with which the fix was made and put into live use depended on its priority (except for reference data fixes, which were generally made very fast)
- On a small proportion of occasions, fixing one problem caused another, which was observed later (the suspense account bug was one of these - it was a side-effect of a previous fix). Mr Parker's witness statement says that he is aware of only one or two such cases.

7.7 Assessing How Well Countermeasures Were Applied

7.7.1 Methods of Assessment

645 I have assessed how well each type of countermeasure was applied, in several different ways:

- I have assessed in section 4-6 to what extent Fujitsu intended to apply the countermeasure, as evidenced by the design documents for Horizon, and documents describing the processes they intended to apply.
- I have assessed in section 6.7 how well some of the countermeasures were tested in Fujitsu's testing processes.
- By examining KELs and Peaks, I have assessed to what extent countermeasures acted in live use of Horizon

FUJ00083961

646 The first topic was addressed in section 4-6 of this report, where I described the Horizon architecture and how various components of the architecture implemented the robustness countermeasures.

647 The second topic was addressed in section 6.7 of the report, where I described the testing process, including how the robustness countermeasures were tested.

648 The Knowledge Error Log (KELs) are over 8,390 records of issues where help desk support was required, and where centrally compiled knowledge would be of assistance. In section 6.7, I have described the processes by which KELs were created, maintained, and used in support of the branches.

649 KELs were used to handle a variety of issues, including user errors, software errors, and reference data errors. In my opinion, the KELs are the best remaining record of what happened when one of these issues occurred - including the robustness countermeasures which were triggered and helped to mitigate its effects. Therefore inspecting the KELs is the best way to assess how effectively, or otherwise, the robustness countermeasures did their job in live use. However they are not a perfect record, as, for instance, they assume a lot of knowledge of Horizon.

650 As I have described above, robustness countermeasures are designed to ensure that several kinds of harmful effect do not ensue - including error in branch accounts. In assessing the effectiveness of countermeasures, I have focused particularly on this aspect of their use - how well they prevented any harmful effects on branch accounts.

651 I have now examined of the order of 200 out of the 8390 KELs, in various different samples. From this analysis, a picture of the robustness countermeasures in action has emerged. However, I need to recognise a number of limitations of the analysis:

- The KELs are written by people who were deeply familiar with the architecture and details of Horizon, for other people who were equally familiar. Therefore they often assume knowledge of Horizon which I do not have. Allusions and terms which were obvious to the reader are no longer obvious to the experts, and it would take a disproportionate effort to disinter them.
- Horizon is a highly complex system, resulting from many thousands of man years of work. KELs refer to many complex parts of Horizon, and it has not been possible in the time I have to follow through all the threads from KELs, understanding in depth all the components they refer to. I have had to rely on the understanding described in sections 4-6 of this report.
- For some KELs - particularly those mentioned only recently in Mr. Coyne's report - there has not been time to analyses them in the depth I would like. Either Mr Coyne or I may be able to provide deeper analysis in our supplemental reports.

652 So there are some KELs for which I cannot be certain exactly which countermeasures were effective, whether or not they had any impact on branch accounts, or how much impact they had. This does not show any limitation in Fujitsu's analysis at the time; it only show the limitations of the analysis I am able to do now.

FUJ00083961

653 Appendix E contains four tables of my analyses of different sets of KELs. The first two of these tables contain analyses of how those KELs show that countermeasures were applied:

- Table A analyses 30 KELs selected at random from the 8390 KELs (actually chosen as every 100th KEL in an alphabetically sorted list), noting some of the applications of countermeasures evidenced by each KEL
- Table B analyses 62 KELs cited in Mr Coyne's report, noting some of the applications of countermeasures evidenced by each KEL

290 Each table also comments on the potential impact of the KEL on branch accounts, but that is not my purpose in referring to them in this section.

291 I note two further limitations of these analyses of countermeasures in KELs:

- They are not intended to be exhaustive. I believe that by further examination of each KEL and its Peaks I could find evidence of other countermeasures in action
- KELs are a biased source of information about countermeasures - giving more information about manual countermeasures, and generally not mentioning automated countermeasures, whose effect is often to forestall the existence of any KEL in the first place. For instance, TIN frequently prevents erroneous or not completed transactions from having any effect - so that no KEL is ever required.

292 Nevertheless I shall briefly summarise the results from the first table of 30 KELs. In these KELs , I have found 54 instances of countermeasures being applied - which in itself shows how widely they were built into Horizon. The countermeasures which are in evidence 3 or more times are DEP, DDS, RDS, TGP, WOR, MID, and BFC.

7.8 Opinions on Robustness Countermeasures

7.8.1 Reliable and Redundant Hardware (RHW)

654 Sections 4-6 survey the evidence that in the data centres, robust and redundant hardware was used extensively, because the consequences of failures in the back office would be so serious.

655 The hardware in the branches was not as reliable or redundant as the back end hardware, but in my opinion this was an acceptable economic tradeoff, because the costs of hardware failure in a branch were so much less serious. Hardware failures frequently occurred in the branches - for instance, failures of keypads. The evidence suggests that branch software was designed with robustness countermeasures such as TIN to ensure that hardware failures would rarely, if ever, have adverse effects on branch accounts. Had this not been the case, the number of errors in branch accounts induced by hardware failures would have been unacceptably high.

656 I conclude that, partly because techniques for dealing with hardware failure are so mature, and have been so for the full lifetime of Horizon, Fujitsu only had to apply these established techniques, and they did so effectively. I have seen no evidence that they systematically failed to do so.

7.8.2 Robust Communication and Replication (ROC)

657 In the early days of Horizon, the underlying data communications infrastructure to the branches was so unreliable that it was essential to provide reliable replication and communication of data in spite of it, and

FUJ00083961

Riposte effectively provided this. I have only seen one KEL where Riposte data replication was suspected not to have worked correctly, in particular circumstances.

658 Even in the later HNG era, failures of the underlying communication infrastructure are still so frequent that it is essential for the layers above it to be robust against such failures. Part of this robustness is provided by standard communication protocols, which Fujitsu used; part was built into the Horizon software, using measures such as DEP and TIN.

7.8.3 Transactional Integrity and Recovery (TIN)

659 Transactional integrity and recovery has been built into database management systems (DBMS) since 20 years before the start of Horizon, and DBMS are used for essentially all Horizon storage of persistent data. Therefore transactional integrity has been a core feature of the design of all components of Horizon from the start. This is evident in many of the Horizon design documents. It is clear from those documents that TIN is a core component of Horizon's robustness against many kinds of failure, such as hardware failures, communication failures, or the user abandoning or cancelling some task in the middle.

660 While it is not mentioned explicitly because it was often so obvious for the intended readers, KELs show many examples where TIN was relied upon to avoid harmful effects.

7.8.4 Defensive Programming (DEP)

661 The layered architecture used in Horizon, and described in sections 4-6, provides evidence of defensive programming - where each layer would defensively protect itself against errors in the layers it depended on.

662 The KELs provide many other examples of defensive programming - where the effects of some error (in hardware, software, or user input) was rapidly trapped by some defensive programming measure - by some module testing its inputs - before it had penetrated far into the system. Defensive programming is essential to make it easy to find the origin of problems, by trapping them before they have gone far. The KELs generally show that problems were easily diagnosed - implying that they were defensively detected near their source.

7.8.5 Data Driven Software (DDS)

663 As was described in sections 3-6, many parts of the Horizon architecture were implemented in a data-driven way - using reference data to define how some generic piece of software would run in many specific applications. This was widely applied, and had advantages for robustness:

- The generic software was more economical to write and test, than specific software for all the applications
- It was repeatedly tested in all the different applications, so was generally highly reliable
- Faults produced by faulty reference data were easy to correct, by fixing the reference data.

664 It also had a potential drawback for robustness, if the reference data was not carefully managed and had errors.

665 KELs show a significant number of faults arising from faulty reference data. These would only rarely affect branch accounts - as they often prevented the use of some transaction at the counter, rather than allow it to be done wrongly - but they may have had the potential to affect branch accounts in some cases. Reference data

FUJ00083961

faults were generally easily diagnosed, and were rapidly fixed by changing the data - so did not have any impact for long.

7.8.6 Secure Kernel Hardware and Software (SEK)

666 The core audit system was a important secure kernel in the Horizon system. It served as the 'gold standard' record of what transactions had been entered in the branch. All the evidence I have seen implies that it was carefully implemented and served its purpose correctly through the life of Horizon.

667 Because the core audit system was a backstop countermeasure, which was only used when other ways of investigating any anomaly had not given an unambiguous result, it was only rarely used, and the KELs provide little evidence of its use. This comparative lack of KELs using the audit system provides confirmatory evidence that the other countermeasures were effective.

7.8.7 Redundant Data storage (RDS)

668 Horizon was very large and complex system, in which the same data were stored redundantly in many different sub-systems. There are very many examples, in KELs and otherwise, where comparisons of these redundant copies of data, automated or manual, were used to detect anomalies. this makes RDS one of the most important robustness countermeasures used in Horizon.

669 Because it often required human inspection to compare the redundant copies of the same data, the countermeasure of RDS often goes hand in hand with manual inspection of data (MID). Some of the many examples of where RDS was used, with or without MID, are:

- Reconciliation is a automated comparison of two copies of the same transaction data - which is responsible for trapping a large number of errors of various kinds.
- Accounting systems store and display financial data from many different sources, in a variety of different ways and different 'slices' - sliced by time, department, product and so on. Managers spend large amounts of their time scrutinising and comparing these figures, and there are automated comparisons.
- System logs and event logs are a redundant storage of information about what happened in transactions, independent of the transaction data which results in the BRDB and other places. When diagnosing any anomaly, Fujitsu staff placed great reliance on these logs, and comparing them with transaction data (e.g. in Credence) to understand what had happened. They were usually successful in doing so.

670 The KELs provide many examples of where RDS was used, with or without MID, to understand the origins of problems.

671 Where RDS was used, and MID was also necessary, this raises a question. If data were stored redundantly somewhere, why was the comparison not made automatically? There may or may not be a good answer to this question. This could have been an automatic countermeasure, and would have trapped an error faster and more automatically than the RDS/MID combination.

FUJ00083961

672 There are examples in KELs where in my opinion RDS could have been used in a fully automated data comparison , but was not, and required MID. This is a potential criticism of Fujitsu's robustness measures. But the tradeoffs to be made in each individual case are complex, and go beyond my knowledge of Horizon.

7.8.8 Manual Inspection of Data (MID)

673 Manual Inspection of data often goes hand in hand with RDS (where the inspection consists of a comparison) , but does not in all cases. It is possible to look at some data and to know 'this is wrong' before you compare it with some other data. Managers often do this with financial data.

674 So while many uses of MID involve RDS, as in the previous sub-section, there are many which do not.

675 One important case, shown in many KELs, is where someone in a support team scans some system logs or event logs, and sees something suspicious. A comparison may be made against some SPM's account of what he did at the time.

676 For this reason and others, in my opinion MID was one of the most important countermeasures in Horizon. MID was , of course, not an automated countermeasure, and was often needed only when automated countermeasures had not prevented a problem. In my opinion, any commercial IT wll incorprate some level of MID countermeasures. It is inevitable in the nature of any commercial accounting system that managers at various levels will scrutinise the financial data held in the system, and that this continual cross-checking will be an important check of the quality of the data.

7.8.9 Double Entry Accounting (DEA)

677 Since Horizon was an accounting system, and interfaced closely with another accounting system (POLSAP), principles of double entry accounting were widely applied - but not universally. Therefore many potential errors were trapped by DEA.

678 There are examples where the use of DEA can be inferred from a KEL, although it is seldom explicitly stated that DEA applied. Wherever a KEL mentions POLSAP, there is usually DEA. Similarly, most changes to the BRDB were made in double entry baskets, which must balance to zero.

679 However, not all changes to the BRDB were of this nature. For instance, in the receipts/payments mismatch bug, the change made to BRDB was not a double entry change, but the related change made to POLSAP was. This led to a discrepancy between BRDB and POLSAP, which could be detected later (RDS/MID). This prompts the question: should the monthly balancing have been designed to make the change to BRDB a double entry change? Would this have been more robust? Or are there changes to BRDB which cannot be double entry? It is possible that Horizon should have been more robust in this respect.

7.8.10 Early Detection of User Errors (DUE)

680 Validation of user inputs has been standard practice in building user interfaces since before the start of Horizon. This involves making all checks possible at the time of input - such as data format checks - or constraining the possible user inputs by a variety or means, such as menus of allowed values..

681 Fujitsu applied these standard techniques to the Horizon desktop, both in the development of Horizon and later of HNG.

FUJ00083961

682 There are many cases of user error which cannot be detected by any IT system. Typical of these is the user simply entering the wrong amount of money for some transaction.

683 In between, there are cases where one could reduce the frequency of user errors by demanding more of the user. Typical of these would be requiring the user to enter the same data twice. The design of these features clearly involves a tradeoff. If some user error is comparatively infrequent, and if its effects can later be corrected (as they usually can be in Horizon - see the next sub-section), then it may not be a good choice to encumber all users with the extra work of a more error-proof interface.

684 Taking these factors into account, in my opinion Horizon was well designed in respect of detecting user errors, and there is not a strong basis for criticising it.

7.8.11 Later Correction of User Errors (UEC)

685 As I have described in sections 4-6, part of Horizon is an accounting system - whose function is to hold a version of the financial state of an organisation, which is always to be kept as accurate as possible. In order to do this, it is essential to have robust processes to correct for the effects of user errors - to check the system's version of the financial state against external reality, and correct it if necessary. Horizon had these , in two main forms:

- Reconciliation and Transaction Corrections: These were applied to essentially all transactions which PO carried out as agents for its clients, such as banks. This had the effect of retrospectively correcting many kinds of errors, including hardware issues (e.g. involving pinpads) or user failures to manually recover recoverable transactions.
- Daily cash balancing: this was an important measure to catch errors in handling cash as soon as possible, so they could be remedied while their possible causes were still fresh in memory.
- Monthly balancing and rollover: This had the effect of correcting many forms of error, including user errors in entering cash or stock transactions into Horizon.

686 In my opinion these formed an essential and highly effective countermeasure.

687 They had an additional important effect. Not only would they correct for user errors, as above; they would also correct for a wide class of other errors, including software bugs, whose effects were the same as user errors.

688 Some of the KELs acknowledge explicitly that these countermeasures would correct the effects of user errors on branch accounts after a delay. For many other KELs, one can infer that the error would have been corrected by the UEC countermeasure, even though it is not explicitly stated. In my opinion this was part of the support team's knowledge of Horizon, which was so evident that it did not need to be stated. Correction of user errors was a routine and obvious part of Horizon's functionality. It was the main purpose of monthly balancing , and of TCs; this did not need to be stated in a KEL.

7.8.12 Manual Workarounds (WOR)

689 KELs provide many examples of manual workarounds being suggested for particular problems. These could be simple (e.g. close down the system and restart it) or could be more complex sequences of actions. In many cases,

FUJ00083961

they were not for the SPM to do, but were corrective actions to be taken on some back end system at the data centres. The descriptions of these corrective actions could be complex sequences, and the descriptions almost universally assumed familiarity with the system in question; the terminology may be unfamiliar to outsiders.

690 The KEL typically does not describe how many times the workaround was used , but may give some expectation of how many times it is expected to be necessary (as in 'if this occurs again...'), or for how long it will be needed (until some fix goes live)

691 The impression gained from these KELs is that Fujitsu were in most cases able to understand the origin of a problem , and suggest a manual workaround, fairly rapidly. They give the impression of a support team that knew what it was doing, not thrashing around for possible solutions; there is not much 'try this and see if it works'.

692 I have considered whether or not this impression is a retrospective bias arising from KELs being 'tidied up' over time, and I consider in the main that it is not. Even short-lived KELs (e.g. those fixed rapidly in reference data) have a decisive air about them.

693 The other impression gained from these KELs is that as soon as Fujitsu understood the origin of a problem (which they usually did) they were then, on many occasions, able to detect its past and future occurrences from examining system and event logs (MID), even if branches did not report it. This also indicates a support team that knew what it was doing.

7.8.13 Testing Good Practice (TGP)

694 An important part of the robustness of Horizon, although not often directly evidenced in KELs, is good practice in testing, to ensure that not many bugs, and only bugs with infrequent occurrence, get though into live use.

695 I have examined Fujitsu's testing practices in section 6.7, and found them to be effective and professional. Particular points relevant to the robustness of Horizon are:

- The use of independent testing teams, whose incentive is to find bugs (rather than to show that the system is correct)
- The use of many test scripts to systematically test 'unhappy paths' where the user does something unexpected or incorrect
- The use of automated regression testing, to ensure that any new release passes all the tests which the previous release had passed.
- Testing of robustness and recovery situations, testing the countermeasures described in this section

696 The other evidence which, in my opinion, supports the effectiveness of Fujitsu's testing is the rarity , in the record of KELs and Peaks, of serious bugs which affected branch accounts in live use.

697 In this respect I note that Mr. Coyne has looked for such bugs, examining over 5000 KELs, and not found them. For none of the KELs he cites does he give the analysis which would be needed to show an effect on branch accounts. I have done this analysis, and most of the KELs he cites have no effect on branch accounts. It can also be inferred with some confidence (although not complete confidence) that none of the KELs examined

FUJ00083961

by Mr Coyne said explicitly 'this will affect branch accounts' - or if it had, he would have quoted it. KELs are typically fairly short documents, easy to scan.

7.8.14 Bug Finding and Correction (BFC)

698 By inspecting the KELs and their related Peaks, one can usually reconstruct the history of any software bug noted in a KEL and find how long it took to fix it and put the fix in live use.

699 I have not yet done this in a systematic or numeric way, and can report only impressions at this stage. My impressions from the KELs I have examined are [a bit of a repeat here]:

- For only a small minority of the KELs were Fujitsu perplexed by the nature of a problem - and those were usually the problems which they could not reproduce in test. This happens from time to time for diverse reasons (particularly when the circumstances triggering a bug are rare, i.e. it is not an important bug), and does not reflect badly on the support team. For most problems, Fujitsu were able fairly rapidly to identify the cause - typically within days or weeks - and know what was needed to fix it.
- If a problem could be corrected in reference data, it was usually fixed very rapidly
- If a problem required code changes, the speed with which these were done depended on the perceived importance of correcting it. The very few examples suggest that if a bug had the potential to affect branch accounts, it was treated with high priority. [a WS to this effect would be very useful. It would also support the converse inference - that low priority fixes were so because they had no potential to affect branch accounts]

7.8.15 IT Architecture (ARC)

700 Much of the robustness of Horizon derives from its architecture. As described in sections 4-6, the architecture of the Horizon back end consists of a large number of discrete systems with well-defined functionality and interfaces, rather than some smaller number of monolithic systems.

701 This form of architecture has been good practice in large-scale IT for many year, and Fujitsu simply had to follow it. This they did, which gave many benefits in design, testing, and support. It generally makes it easy for a support team to isolate the cause of a problem into one of the systems.

702 In my opinion, the quality of the Horizon architecture is generally good. This follows not only from my examination of the architecture in sections 4-6, but also from the effectiveness of the support processes which depend on it.

703 In designing the architecture of Horizon, Fujitsu had a major advantage - a fresh start in 1996, unencumbered by any legacy architecture. In many of the large systems I have worked on - particularly in the finance and healthcare sectors - new developments need to be fitted onto some unwieldy legacy architecture - making it yet more unwieldy. While the Horizon architecture is complex, it does not have the arcane over-complexity that I have seen in many commercial organisations.

7.8.16 Quality and Change Control (QCC)

704 CE -suggestions?

FUJ00083961

7.8.17 Managing Non-Functional Requirements (NFR)

705 It is bit hard to make the link from NFR to robustness in live se. There are so many of them.

706 CE - suggestions?

7.8.18 Other Aspects of Robustness

707 Further evidence on the general robustness of Horizon comes from the lack of interruptions of the Horizon service. In recent years, several major banks, and air traffic control systems, have suffered well-publicised IT 'meltdowns' which led to serious interruptions of service; these systems were not robust. As far as I am aware, in more than 18 years' service, Horizon has experienced very few such interruptions - one on 9th May 2016, cited in the claimants' WS. I am seeking further evidence on this point.

708 This incident, and others that I have found in KELs, imply that the robustness of Horizon and the business processes around it was not always as good as it might have been - but in the great majority of cases, was good enough. In a typical incident recorded in a KEL and its Peaks, there were several countermeasures (different lines of defence); some were breached, and others were not. On some of these occasions, in my opinion some of the lines that were breached could have been built more strongly, so they would not have been breached. For instance, in some cases there might have been more automatic cross-checking of data, where there was none.

7.8.19 The Effect of Multiple Countermeasures

709 This is my most important conclusion on the robustness of Horizon.

710 I have described 17 different classes of countermeasure - some overlapping - and have given examples of how they prevented errors from affecting branch accounts.

711 My most important conclusion is that these countermeasures did not need to be individually perfect, in order to be highly effective in combination.

712 To see this, suppose (for illustration) that each type of countermeasure is implemented with only 90% effectiveness - so that one bug in 10 gets past it..

713 This would mean that two countermeasures, acting independently, would catch some bug with 99% effectiveness. Only one bug in a hundred would get past both countermeasures. Three countermeasures would catch it with 99.9% effectiveness, and so on.

714 Of course it is not always possible to have several countermeasures at once to detect any bug - as bugs have something of the nature of 'unknown unknowns' - but it is an engineering principle to try to bring as many countermeasures as possible to bear on any possible threat. This principle can result in highly robust systems, in spite of typical rates of errors per line of code (such as one defect in 1000 lines of code) often achieved in the IT industry.

715 In my opinion, Fujitsu successfully applied this strategy of many diverse countermeasures, as is demonstrated in many of the KELs. As a result, Horizon is a highly robust system.

716 In the foregoing discussion, many of the countermeasures are fully automatic and act with no human intervention. Other countermeasures such as MID require human intervention. In my opinion, the fully

FUJ00083961

automatic countermeasures in Horizon were well designed and worked well. However, it is more difficult to find evidence of how well they worked in live use of Horizon, because my main source of information for this - the KELs - is biased in the following sense: the KELs record mainly occasions where some defect evaded the automatic countermeasures, and necessitated some manual countermeasure. So KELs tend not to record the operation of fully automatic countermeasures, only the manual ones. The best evidence for automatic countermeasures in service is the comparative rarity of KELs over 19 years.

7.9 Variations in the Robustness of Horizon Over Time

717 In his report, Mr. Coyne has expressed an opinion that the robustness of Horizon may have varied over time - implying that the number of bugs which could have had impact on claimants' branch accounts may also varied over time.

718 Mr Coyne has described the variability of robustness of Horizon as a possibility, and has not cited evidence that it actually varied.

719 In this sub-section I shall discuss the two related questions raised by Mr. Coyne:

- Variability in the robustness of Horizon over time
- Variability over time in the number of bugs which may have affected branch accounts.

720 The first of these questions is problematic. Neither Mr Coyne or I have suggested any numerical measure of robustness. In my opinion, the topic is so complex (consisting of many different countermeasures, acting together in many different ways) as to admit of no single numeric measure. If you cannot measure something, there is an important sense in which you cannot ask how it varies over time. You can only ask how some consequence of robustness - such as the impact of bugs on branch accounts - varied over time. That is the second question.

721 As I have described earlier in this section , robustness (as opposed to freedom from bugs) consists of effectively applying the many countermeasures described in this section, so that the effects of imperfections, including software bugs, are limited to an acceptable level.

722 Of the 17 robustness countermeasures described in this section , in my experience all of them have been available, and have been a common part of mainstream IT practice, since before the inception of Horizon. Therefore it cannot be said that any of them were not available to Fujitsu from the start. I have not found any evidence that any of the countermeasures was applied more or less effectively in any period of Horizon's lifetime.

723 It seems to me rather unlikely that , if any countermeasure is applied in some release of Horizon, that it should not be applied in the next release - i.e. that it should be deliberately dropped. In my opinion there was no incentive for PO or Fujitsu to do this. It might be said that some countermeasure could be applied less effectively in a later release, for one of two reasons:

a) If it is an automated countermeasure, because the countermeasure software had some new bug

FUJ00083961

b) If it is a manual countermeasure, because the team doing it became less effective (e.g. due to a cut in manpower, or a mis-managed organisational change).

724 These are both possible, but I have not seen evidence for either of them. (a) is made more unlikely by the practice of regression testing - ensuring that a new release passes all tests applied to the previous release.

725 Therefore I have seen no evidence that the robustness of Horizon *per se* has changed over time - although , as I have said above, the question itself is rather ill-posed..

726 I turn next to a possible consequence of changes in robustness, which is a change in the rate at which bugs affect branch accounts.

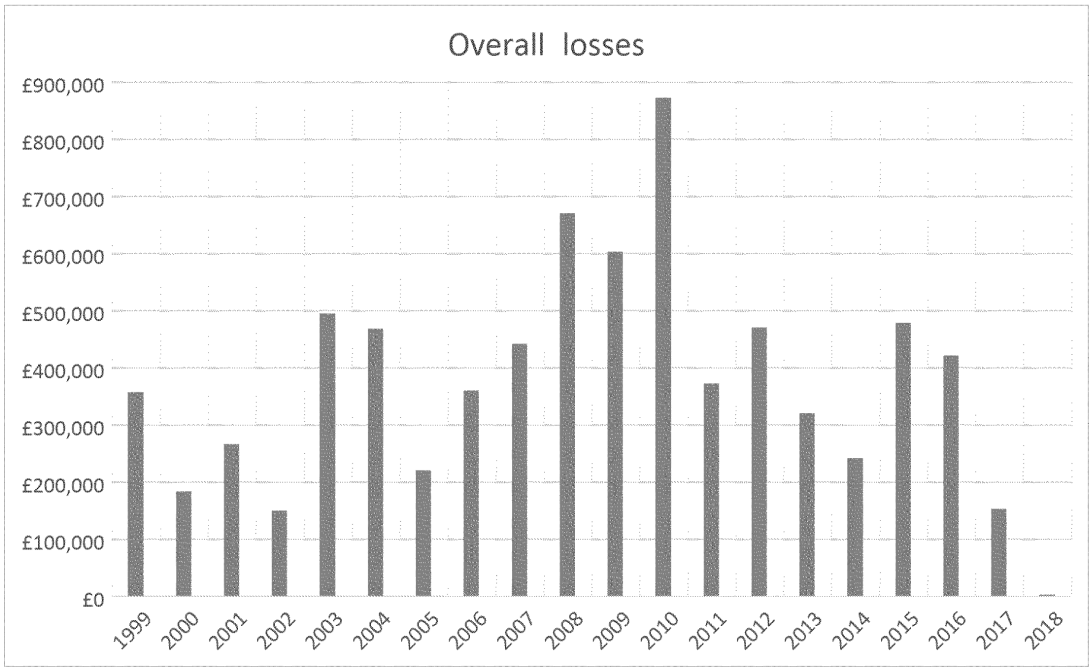
727 The claimants are now putting forward two hypotheses:

a) That some large part of their £18.7M shortfalls arises from bugs in Horizon

a) that the rate at which this happened has varied over time

728 If both hypotheses are correct, one should be able to see the result, in a variation over time in the rate at which claimants experienced shortfalls.

729 Data on the occurrence of claimant's shortfalls by year, taken from Section 3 of the individual claimants' claims, is shown below:



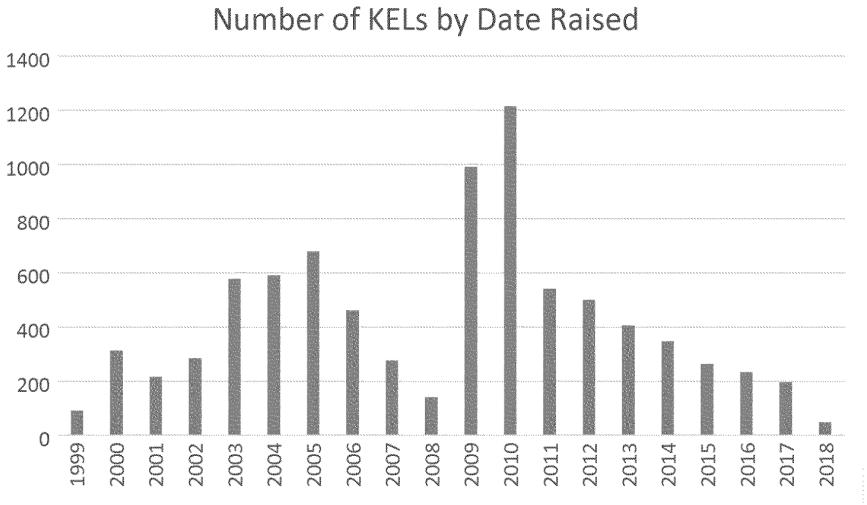
730 It can be seen that there is some variability over the years, but that variation is in my opinion mainly consistent with random fluctuations, with no systematic trend and no huge variability.

731 The main exception to this is a noticeable peak in 2010. In my opinion , this peak may have been caused by the fact that all branches were audited as they migrated from Horizon to HNG [cite evidence]. This audit would naturally expose latent problems in branch accounts, leading to a surge in shortfalls.

FUJ00083961

732 However, I note that in my opinion, as expressed in the next section, the claimants' shortfalls are not caused by bugs in Horizon, or any lack of robustness in Horizon. So in my opinion, the graph above says nothing about robustness of Horizon, or about its consequences. So I have looked for other evidence about occurrence of errors.

733 KELs are written to help address a range of issues raised by SPMs. Many of those issues arise from human errors or other causes, but some of them arise from faults in Horizon. Therefore a chart of the number of KELs raised in each year may shed some light on the occurrence of faults in Horizon. That chart (which I have made by an automated analysis of the KELs) is shown below:



734 It can be seen that there is a noticeable surge in the rate of creation of KELs in 2009, 2010 and the years immediately following them. Some of this surge may have arisen because the support team were having to deal with new issues under HNG, because it was different from old Horizon, and were therefore creating new KELs to help them do so.

735 However, this surge post-2009 also raises the possibility that there was a higher level of bugs in HNG - which were teething problems of the new HNG release - and I do not exclude that possibility. How significant is the surge?

736 To answer this question, I need to anticipate a conclusion from section 8 of this report, about Horizon issue 1.

737 In that section I show - using several separate lines of analysis - that the total impact of all bugs in Horizon, on claimants' accounts over all years, is very small , compared to their £18.7 million total shortfalls. It is much less than 1% of that amount.

738 I have therefore calculated the sum of the impact of bugs over all years, and have shown that it is very small. If one knows the sum of a quantity over all years, then the fluctuations in that quantity over the years are of little importance. The 'ups' are cancelled by the 'downs'. All that matters is the sum, which we can place an upper limit on. So any fluctuations over time, in the rate at which bugs in Horizon caused shortfalls, are of little or no importance.

FUJ00083961

739 I conclude that while there may possibly have been fluctuations over the years in the rate at which bugs in Horizon could have caused shortfalls, any such fluctuations are of no importance, because their sum over all the years of Horizon's life is very small.

7.10 Horizon Issue 4

740 **Issue 4**: To what extent has there been potential for errors in data recorded within Horizon to arise in (a) data entry, (b) transfer or (c) processing of data in Horizon?

741 Parts of this Horizon issue are difficult to interpret in a way that will assist the court. My preferred interpretation is that they are to be seen as selected subsets of Horizon Issue 3.

742 The reference in (c) to 'processing of data in Horizon' is difficult to understand, because essentially all parts of Horizon are involved in 'processing of data' . So my opinion on issue 4(c) is the same as my opinion on issue (3).

743 The reference in (b) to 'transfer of data recorded in Horizon' is difficult to understand, because many parts of Horizon are involved in 'transfer of data' - which is not necessarily restricted to 'transfer of data over communication channels'. If it is so restricted, I refer back to section 6 of this report, and to the ROC countermeasure described earlier in this section. Horizon successfully incorporated the usual measures to protect data in communication. If it is not so restricted, most of the countermeasures described in this section are applicable.

744 The reference in (a) to data entry is more specific. In my opinion, the Horizon user interface incorporated industry-standard measures to detect user errors in data entry wherever possible. Since many errors of data entry cannot be detected automatically, this was as robust as it could be. I refer to my discussion of the DUE countermeasure in this section.

745 So in my opinion all aspects of issue 4 are subsets of issue 3 on robustness, and my opinion on issue 3 applies to issue 4.

746 I note that issue 4 asks about the 'extent' of 'the potential for errors in data'. However, unlike Horizon issue 1, it does not restrict these to errors which impact branch accounts. Therefore it is not possible for me to quantify this extent in any useful way, since its scope is so broad; and 'extent of potential' is intrinsically difficult to quantify. - being even broader than an extent of actuality.

7.11 Horizon Issue 6

747 **Issue 6**: To what extent did measures and/or controls that existed in Horizon prevent, detect, identify, report or reduce to an extremely low level the risk of the following:

a) data entry errors;

b) data packet or system level errors (including data processing, effecting, and recording the same);

c) a failure to detect, correct and remedy software coding errors or bugs;

d) errors in the transmission, replication and storage of transaction record data; and

e) the data stored in the central data centre not being an accurate record of transactions entered on branch terminals?

FUJ00083961

748 Like Issue 4, issue 6 raises similar issues of interpretation, and appears to consist mainly of selected subsets of Issue 3.

749 In that Issue 6 addresses 'measures...[to] prevent, detect, identify, report or reduce...' , Issue 6 seems to be largely about robustness and countermeasures - and to that extent, is subsumed under issue 3.

750 Issue 6(a) seems to repeat issue 4(a), and my opinion is repeated.

751 In issue 6(b), I cannot usefully interpret the reference to 'system level errors' , which might refer to any aspect of the Horizon system; so all robustness countermeasures apply. 'data packet' errors might refer to many different types of data packet. If it is used a widely recognised sense, it refers to data communication, and my opinions on the ROC countermeasure apply. As for issue 4(c), the reference to 'data processing' makes the scope of issue 6(b) as wide as that of issue (3), and my opinion is repeated.

752 Issue 6(c) refers to software coding errors. This was one of the threats mentioned in my analysis of robustness under issue 3, and is addressed under several robustness countermeasures, notably BFC. My opinion is repeated.

753 Issue 6(d) appears to repeat issue 4(a), and my opinion is repeated.

754 Issue 6(e) draws out a specific point mentioned above under issue 3. Fujitsu took care to ensure that data stored in the core audit database was an accurate and secure record of transactions entered on branch terminals, and could be called upon as a 'gold standard' when investigating any anomaly. This is addressed under the countermeasure SEK. In this respect, Horizon was robust.

755 So all aspects of issue 6 are subsets of issue 3 on robustness, and my opinion on issue 3 applies to issue 6.

756 As for issue 4, the 'extent ' aspect of issue 6 is difficult to address in any quantitative manner.

7.12 Mr Coyne's Opinions on Robustness

757 In the expert's joint statement, Mr Coyne has agreed that robustness does not equate to perfection or the lack of bugs.

758 His opinions on the robustness of Horizon are stated in paragraphs 5.82 - 5.200 of his report.

759 In paragraph 5.83, he confirms the view in the joint statement that robustness does not equate to lack of bugs.

760 However, in paragraph 5.88 he says he is unable to estimate the level of robustness in Horizon, citing ' the sheer enormity of the task' and implying that he would need to understand all the code to do so. This seems to confuse robustness with being error-free. In my opinion, errors in one part of Horizon may well be successfully mitigated by a countermeasure in another part. It is important to understand the countermeasures, not the whole of the code. Mr Coyne has not done this,

761 This confusion seems to be repeated at his para 5.110, where he states that Horizon is 'neither infallible or totally robust'. In my opinion, robustness has nothing to do with infallibility, and there is no such thing as total robustness. Robustness is a matter of dealing with a variety of threats, including software errors, so as to make their consequences acceptable in a business context. This involves the use of a range of countermeasures, none of which Mr. Coyne has discussed in his report.

FUJ00083961

762 In para 5.11, Mr Coyne states (contrary to his agnostic para 5.88) that 'the electronic processes in Horizon are relatively robust'. However, in several place he implies that the robustness of Horizon may have varied over time, implying that at some times it may have allowed many errors to affect branch accounts.

763 I have addressed this question in section 7.8. There I showed that , however much the robustness of Horizon may have varied over the years, the sum total over all years of bugs which have affected claimants' accounts is very small compared to their losses. If the sum of a quantity over all years is very small, then fluctuations in that quantity over the years do not matter.

764 In the time available to me since receiving Mr Coyne's report, I have not been able to address all the detailed points he makes in paras 5.82- 5.200. Those that I have been able to address are in Appendix Z . I shall address them in more detail in my supplementary report.

7.13 Documents Cited in the Claimants' Outline of August 2018

765 [To go to Appendix J]

766 In paragraph 3.2 of their outline claim submitted on August 17th, the claimants cite a number of documents which, they say, imply that Horizon is not robust, on 'a sensible construction of the term "robust"' - which, as their paragraph 3.1 implies, they did not have at the time.

767 These documents are also cited by Mr Coyne un his report.

768 Nevertheless, on the definition of 'robust' which I have given above, I need to assess whether these cited documents provide any evidence that Horizon was not robust - or whether they merely establish that it was not perfect, which is a completely different matter, and in any case is agreed between the parties.

769 In para 3.2(a) the claimants say that Horizon contained the bugs identified in their paragraph 1. As has been described above, the mere existence of bugs in a system does not imply any lack of robustness. Robustness is a matter of how resilient the system is to the effects of the bugs in it. If there were a very large number of bugs which affect branch accounts, then that fact in itself might imply that the system is not robust. However, the number of bugs cited by the claimants in their paragraph 1 is very small , for a system resulting from thousands of man years of work.

770 The claimants have presented no evidence that the bugs they identified were not handled robustly, or that the many countermeasures failed to operate. My detailed analysis of those bugs, which follows in section 8.X, implies that they were handled robustly.

771 In their paragraph 3.2(b) that claimants cite two KELs, as 'failures of internal mechanisms intended to secure the integrity of data'.

772 The first KEL, dsed4733R, resulted from a mis-named recovery script. It is to be expected that in a system of the size of Horizon, recovery scripts will occasionally be mis-named. So the existence of one such script implies no lack of robustness. In the text of the KEL it is stated: *'until resolved the SSC will need to clear the failed recoveries on a daily basis*'. This implies that there was a resilience mechanism (WOR), to deal with the effects of this fault until it was remedied. The KEL implies that Horizon was robust.

FUJ00083961

773 The second KEL, obengc5933K, refers to a rare form of communication failure (which could not be reproduced in test), which led to a failure in a transaction recovery. This was clearly a rare and complex circumstance, which tested Fujitsu's processes for error detection and correction more than many others would.

774 However, transaction recovery often involves some user action which is not routine or familiar to the user, and therefore has a relatively high incidence of user error. Horizon has robust mechanisms to detect and correct these errors in transaction recovery. The most important mechanism is reconciliation, leading to transaction corrections for many kinds of errors, including errors in transaction recovery (UEC). There is a reference in the KEL to DRS (Data Reconciliation System): ' *On DRS, all parts [C12, C4, C112] are all present with 'AmountConfirmed' showing the amount of the txn in question* ' That reference shows that in this case, the resilience mechanisms were in place and working. Therefore, like the previous KEL, this KEL confirms the robustness of Horizon.[this may need more deep analysis, if it could lead the clerk to pay out more than he should to the customer].

775 In 3.2(c) the claimants say: *'the system did not enable such discrepancies to be detected, accurately identified and/or recorded either reliably, consistently or at all'* . This is a vague and general assertion, not backed up by any specific evidence or argument. If 'such discrepancies' refers to the two KELs cited above, it contradicts the facts recorded in those KELs.

776 In 3.2(d) the claimants say that the system did not reliably identify 'Mis-keying'. The Horizon user interface had all the usual measures built into it to identify mis-keying, making all the checks possible on user input and rejecting inputs which failed those checks (countermeasure DUE). Certain types of mis-keying (such as typing a '2' instead of a '3') in principle cannot be identified by the system at the time of input. In these respects, the claim of 3.2(d) appears to be either wrong or meaningless; I await clarification from Mr Coyne's report.

777 In 3.2(e) the claimants say that Horizon *'required numerous processes and workarounds to be in place to allow Fujitsu to modify and correct data already recorded by Horizon, which would not be required in a "robust" system*'. They have missed the point that the existence of numerous checks and workarounds (countermeasures such as RDS, MID, WOR) is precisely what is needed to make Horizon robust. Only if the workarounds had to be invoked too many times, so that they were too costly or too error-prone in themselves, could it be argued that Horizon is not robust. The claimants have cited no evidence that this was the case. The available evidence implies that the processes and workarounds were used very infrequently. That is robustness in action.

778 In 3.2(f) the claimants cite four documents which, they say, imply that Horizon was not robust. I shall assess these documents in turn.

779 The first document is a description of internal control matters arising from the 2011 audit by Ernst & Young. This is one of a series of annual audit reports by E&Y, and I need to examine more of the full series of reports to make a fuller assessment of its significance. For the moment, I shall assume the claimants have selected this report because it makes the most important points.

780 E&Y's main recommendations in their executive summary are to:

f) Improve outsourcing application management

FUJ00083961

g) Strengthen the change management process (this includes a previous recommendation in the summary)

h) Strengthen the review of privileged access

781 Section 4 of the report contains the IT-related recommendations for the current year.

782 The main recommendation under (a) is that *'POL should take ownership of the effectiveness of the control environment with Fujitsu, requiring Fujitsu to implement a control framework devised by POL'*. This can be summarised as 'POL should not leave so much of the governance to Fujitsu'. It does not say the governance is failing in any way; it just says that POL should take more ownership of it.

783 The recommendations under (b) and (c) included recommendations on both POLSAP and HNG. Those under HNG centred around the authorisation of changes to HNG - concerning too many developers having access to make changes to the live system, and insufficient authorisation and approval by POL of the changes.

784 In summary, the observations and recommendations concerned management approval for changes - whether by PO or by Fujitsu - and were to the effect that the processes were not being fully followed, or that a higher level of management control would be appropriate. The auditors did not connect these observations with any harmful consequences, such as a high level of bugs or interruptions of service, which had flowed from them. They were saying in essence that 'processes can be improved or followed more closely' - not that 'poor processes have had harmful consequences'.

785 Any auditor has an obligation to make some observations and recommendations for improvement - both to provide value to his client, and to protect his position should things go wrong at some later stage. No organisation follows all its written processes to the letter - so observations about processes not being followed are easy ones for an auditor to make, and may still prove valuable to the client. In my opinion, this is the kind of result that one expects from an audit, and is not an indicator of poor robustness in Horizon. I shall examine audit reports for other years to check this. Repeated patterns may reveal more.

786 The claimants then cite POL-217341, which is the result of review of POLSAP in 2012. This is not a review of Horizon; therefore most of it is out of scope for the Horizon trial. As the claimants describe, it observes a potential for errors and discrepancies in client balances. However, client balances concern the PO's relationship with its client organisations, on whose behalf it carries out business. Client accounts have no direct connection with branch accounts - so the connection with the Horizon issues is at best tenuous. The report recommendation contain only two mentions of HNG, both concerning the strengthening of access controls - echoing the E&Y audit report. Therefore this document has little if any relevance to the robustness of Horizon.

787 The claimants next cite POL-217378 – the minutes of a meeting on 18 September 2013 of the Defendant's Risk and Compliance Committee. The minutes record a decision not to implement a recommendation of a previous E&Y audit report (for 2013 - not the report for 2011 cited above). I am waiting for further evidence, including the 2013 audit report, to assess this issue. However, I note that it is not the duty of management to follow every recommendation in any audit report. Managers know more about their business than auditors do. Some audit recommendation may not be the most cost-effective way to achieve some objective, or may be inappropriate for many possible reasons. At a cost of £1 million, this was clearly an expensive recommendation, and the

FUJ00083961

committee decision that the benefits were not proportionate to the costs may have been a correct one. The decision not to follow it is in no sense evidence of poor management of Horizon, or of a lack of robustness following from poor management.

788 The claimants then cite POL-0216106, which is a report by Detica in 2014 into fraud and detection of non-compliance in the Post Office. While this report is interesting for its insights and data about the causes of discrepancies in branches, it is scarcely relevant to the Horizon issues. This is because Detica's brief was to examine the workings of PO's central fraud detection teams, and most of its recommendations concern those teams. This has nothing to do with bugs in Horizon, or a lack of robustness in Horizon, which might affect branch accounts.

789 The report also shows that Detica went beyond their brief, to recommend a wide-ranging programme of business process changes and changes to Horizon functionality, with a duration of four years. This recommendation was not costed, and if it was not taken up, it would not be surprising. I have not yet seen evidence on this.

790 In summary, there is nothing in the four documents cited by the claimants in para 3.2 of their outline to imply that Horizon is not a robust system.

791 To address paragraph 3.3 of the claimants' outline, I was awaiting clarification of 'the result alleged by the claimants' and 'the level of discrepancies arising from bugs and errors in issue in these proceedings'. Without some quantitative measure of these levels, it is not possible to assess the argument about the 'sheer number of transactions' which is hinted at in this paragraph. I now understand that any such clarification is not to be expected in time for my first report.

FUJ00083961

8. THE EFFECT OF HORIZON BUGS ON BRANCH ACCOUNTS

8.1 Horizon Issue 1: My Opinions

792 **Issue 1**: To what extent was it possible or likely for bugs, errors or defects of the nature alleged at §§23 and 24 of the GPOC and referred to in §§ 49 to 56 of the Generic Defence to have the potential to (a) cause apparent or alleged discrepancies or shortfalls relating to Subpostmasters' branch accounts or transactions, or (b) undermine the reliability of Horizon accurately to process and to record transactions as alleged at §24.1 GPOC?

793 The question raised in issue 1 follows on directly from the question raised in issue 3. Given that there were bugs, error or defects in Horizon (as there are in all commercial IT systems), how effective were the robustness countermeasures of Horizon in preventing or limiting any errors in claimant's branch accounts? Issue 1 asks about some of the consequences of Issue 3.

794 In section 7 I examined the robustness countermeasures. I found that in building and supporting Horizon, Fujitsu applied a set of well established countermeasures, which have been familiar to me throughout my career. I found that they applied them effectively. In particular I looked at the KELs, which record how countermeasures were applied to events that threatened to create discrepancies in branch accounts. I found that in those cases, the countermeasures worked effectively. Only in a small minority of cases was there doubt (from lack of information recorded in the KEL) or any potential to affect branch accounts erroneously.

795 Therefore in my opinion on part (a) of issue 1:

i. **Significant detected defects:** if in some month there was a significant shortfall in any claimant's branch accounts (for definiteness, a shortfall of £300 or more), the chances of that having arisen from a bug or defect in Horizon which has been detected are very small indeed. I shall quantify this below.

ii. **Undetected defects**: the claimants have raised the possibility that shortfalls might be caused by defects in Horizon which were never detected, and may not be known about to this day. As I shall describe below, because of the many countermeasures built into Horizon, the potential for any such 'unknown bugs' is very small indeed. Any bug with significant impact on branch accounts would be known about. The net impact of unknown bugs on branch accounts is very small, compared with the impact of defects which are known about and were recorded in KELs.

iii. **Financial Impact of defects**: Because of (ii), the KELs are a good source of information about bugs and the effect they might have had on branch accounts. One can examine the KELs, determine in which of them there might have been an impact on branch accounts, and place a conservative upper limit on the amount of this impact. Doing this sum, correcting for factors such as any inefficiency of the KEL creation process, lack of detail in KELs, and limitations in the sample of KELs I have been able to examine, I have calculated an upper limit on the financial impact of bugs in Horizon on the claimant's accounts. As I describe next, this upper limit is very small.

796 Even using very conservative assumptions as described below, designed in all cases to favour the claimants, the total net impact of all bugs in Horizon on the claimant's branch accounts must be less than a fraction of one

FUJ00083961

percent of the shortfalls experienced by the claimants. Bugs in Horizon cannot account for even a small part of their shortfalls - either for all claimants taken together, or for any individual claimant.

797 In my opinion this result follows inescapably from the evidence.

798 In my opinion of part (b) of issue 1: the Horizon Core Audit Process was designed to create a secure, accurate and immutable record of what was entered into Horizon at the branch, and to record verifiably who had entered it. In my opinion, regardless of any other processing done in other parts of Horizon, the core audit database was an accurate record of transactions entered in the branch. It was carefully designed and tested, and I have seen no evidence that it ever failed in service. Therefore in any case of doubt about processing done in other parts of Horizon, this record was available to establish the true state of any branch's accounts, based on transactions entered in the branch.

799 These opinions apply both to old Horizon (pre 2010) and Horizon New Generation (HNG).

800 In section 7, I addressed Horizon Issue 3: To what extent and in what respects is the Horizon System "robust" and extremely unlikely to be the cause of shortfalls in branches? I said there that I would postpone addressing the second part of that issue, 'extremely unlikely to be the cause of shortfalls in branches' to this section. As my opinion on part (a) of Issue 1 makes clear, in my opinion on Issue 3, the robustness of Horizon made it extremely unlikely to be the cause of shortfalls in branches.

8.2 Unknown Bugs in Horizon

801 The claimants have raised the possibility that shortfalls might be caused by defects in Horizon which were never detected, and may not be known about to this day.

802 They may wish to imply that the financial impact of these unknown bugs is essentially unknowable, and so may have been very large. In my opinion this is not so, because of the robustness countermeasures built into Horizon.

803 Part of the purpose of robustness in any financial system is to ensure that far-reaching errors in accounts do not occur. An important part of this is to ensure that if they should occur, they are rapidly detected - and do not persist, unknown, for long periods. Horizon was a typical financial system in this respect. In my opinion its robustness countermeasures worked well.

804 A particularly important countermeasure was the manual inspection of data (MID), by the SPMs themselves, at various times - in customer transactions, in daily cash balancing, and in their monthly balancing and rollover. At these times, one can expect at least some of the SPMs to have been highly vigilant.

805 I assessed this aspect of robustness in some detail in section 7.4, using the evidence available about known defects in Horizon with known financial impact. There I stated that:

806 I can go a little further than this [evidence from the suspense account bug] by making weak inferences about how a manager of a small business, such as an SPM, needs to prioritise his time in monthly balancing. There may also be relevant WS evidence. I assume the following, as best assumptions of SPM behaviour in reporting anomalies in their monthly balancing:

FUJ00083961

a) If a discrepancy is £1000 or more , the SPM probably needs to investigate it. If he cannot find the cause in his branch, the likelihood of his reporting it through a help line is 80%

i) If a discrepancy is of the order of £300, 30% of SPMs will report it

j) If a discrepancy is of the order of £100, 10% of SPMs will report it

k) For a discrepancy of £10 or less, it is usually not worth the SPM's time to investigate it (because errors in counting cash or stock are often larger than this); so these are reported on less than 1% of occasions.

807 The reader is referred to section 7.4 for more detailed analysis leading to this opinion.

808 Even of this conclusion is made more conservative - assuming, in the claimants' favour, that SPMs do not report bugs as frequently as described above - it means that any bug in Horizon, with significant impact on branch accounts, will, after only one or a few occurrences, be reported by some SPM. Therefore it does not remain an unknown bug.

809 Note from the estimates above that, as the financial impact of one occurrence of a bug decreases, the expected number of occurrences needed for some SPM to report it will increase - but its financial impact on each occurrence is less, so more occurrences are needed for it to have a given financial impact.

810 This means that, whatever may be the average financial impact of a bug on one occurrence, before that bug has had a large total impact (for illustration, say £3000), it is very likely to be reported, and so will be known about. The probability of a bug with financial impact remaining unreported, and remaining unknown, is small.

811 It then follows arithmetically that the total financial impact of unknown bugs - those that remain unreported - is small, compared to the total financial impact of known burgs.

812 There is one possible exception this analysis. The possible exception is a bug whose financial impact on any one occasion is so small that no SPM will ever report it (e.g. they always believe it arises from error in their branch; or it is not worth their time to report it)- but which occurs on so many occasions that its net financial impact is significant. I refer to these as 'micro-bugs' and shall analyse them on more detail in section 8.X. They do not alter the conclusion above.

813 Note that by the phrase 'unknown bug' I mean 'a bug which was not detected near the time it occurred'. I do not mean 'a bug we do not know about today'. Normally, when a bug with possible financial impact occurred, this resulted in the creation of a KEL, and the KEL is now in evidence. But I do not assume that this process was 100% efficient; so some known bugs are not recorded in KELs, and some 15% of KELs have now been archived, so the experts have not yet seen them. I shall account for this possible inefficiency in the creation and availability of KELs, just as I shall account for KELs I have not yet been able to analyse in detail, by scaling up the financial impact of bugs recorded in KELs that I have analysed. This is the standard engineering sampling approach to measurement - used whenever measurements cannot be complete.

814 This scaling up of the effects of known bugs is the proper way to account for bugs which were known at the time they occurred, but for which the evidence no longer exists - rather than treating them as some complete unknown which might be arbitrarily large. I have shown here that they cannot be arbitrarily large. Their effects must be much smaller than the effects of known bugs.

FUJ00083961

8.3 Impact of Bugs on Claimants' Branch Accounts - Qualitative Opinion

815 Issue 1 asks about the financial impact of bugs in Horizon on claimants' branch accounts.

816 Following my analysis of robustness in section 7, it will now be clear that the answer to this question depends on the robustness of Horizon - not on how many bugs there were, but on how well the effects of these bugs were countered and mitigated by the robustness countermeasures, to prevent them from creating discrepancies or shortfalls in branch accounts.

817 Because of this, any simple counting or cataloguing of bugs - for instance, derived from KELs and Peaks, as Mr Coyne has done - does little to answer the question of Issue 1. For any bug or anomaly that might have affected branch accounts, it is necessary to consider the robustness countermeasures, and how they operated in that case.

818 In my opinion, the KELs are the main source of evidence available about how the countermeasures operated in live use of Horizon. They are the most direct evidence I have, closer to the coal face of Horizon in action than other summary reports or records of meetings. I have now examined more than 150 KELs from this viewpoint - asking what can be inferred from the KEL about what countermeasures were in operation, and how effectively they operated, in the light of my own experience of the same robustness countermeasures. Before the trial, I intend to examine more KELs in this way.

819 While the KELs are by no means a perfect source of information about the operation of countermeasures - as they are often written tersely, assuming a deep knowledge of Horizon and the processes around it, and they often focus on how to help the branch, rather than on spelling out any underlying problem - nevertheless a clear qualitative picture emerges from this examination:

a) Countermeasures play a key role in countering effects such as hardware failures, communication failures and human errors, which occur with much greater frequency than software errors (because software has been extensively rested before it goes live). In this role, the countermeasures are very effective. Many KELs show the effectiveness in this respect of countermeasures TIN, DEP, RDS, DEA, MID, and UEC , as well as others. It can be inferred that for any software bug which has the same effects as a hardware error, communication error, or user error, the same countermeasures will prevent any adverse effect on branch accounts. This is the case for many of the KELs which Mr Coyne and I have examined - although he has not pointed this out.

l) Using system logs and event logs, Fujitsu were able to trace the full sequence of events in the branch, to determine with high reliability the events leading to any anomaly noted in the branch, and to assess whether it arose from human error, or from some other cause which might be a software error.

m) If a software error was suspected - and especially if it might have any effect on branch accounts -Fujitsu were in most cases able to rapidly identify its cause ; to use system logs and event logs to identify any branches affected, before or after the effect was reported; in most cases to reproduce the effect in test, and diagnose the cause. They could usually suggest workarounds for any branch during the period while the bug was being fixed.

FUJ00083961

n) The KELs were written by a fairly small group of people, who I infer were mainly in the third and fourth lines of support. They demonstrate a comprehensive grasp of Horizon and the processes around it. Problems were usually rapidly diagnosed, and as the Peaks demonstrate, the processes for correcting the software were tightly managed.

o) My impression from the KELs is of a support team that knew what they were doing, on an IT system that they understood well - trying, and succeeding, to rapidly diagnose and close down any problem, to remove it from their future workload.

820 Therefore in my opinion the robustness countermeasures in Horizon worked well in preventing bugs and other effects from introducing inaccuracies in branch accounts. This applies both to the automated countermeasures within Horizon itself, and to the manual countermeasures applied to any effect which was not countered automatically. In this respect I stress that any system which relied only on automated countermeasures within it, and had no manual countermeasures to back them up, would in my opinion not qualify as a robust system. I have never seen a major commercial system without manual countermeasures.

821 Because the robustness countermeasures worked well, as the KELs record, the vast majority of anomalies recorded there were not bugs with adverse effect on branch accounts. If a bug had had such adverse effects, it would with high probability be recorded in a KEL.

822 Therefore in my opinion, because the robustness countermeasures worked very well, there were very few bugs which introduced inaccuracies in branch accounts, and their financial impact across the PO branch network was very small.

823 Mr Coyne's report appears to imply otherwise. But he has not analysed the KELs or Peaks to sufficient depth to consider the effects of robustness countermeasures. Therefore his report contains little or no analysis to contradict my opinion. I have examined the KELs he relies upon, and they confirm my opinion.

824 I have stated this opinion in qualitative terms. However, Horizon issue 1 asks about the extent of impact of bugs, and I go on to assess the extent in quantitative terms.

8.4 Measures of Extent

825 Expert Issue 1 begins with the phrase 'to what extent'. This implies that the experts are not being asked to decide on a yes/no question, but is being asked to assess the extent of some effect. This raises the question: against what scale of measurement against should the 'extent' be assessed? I need to define the scale of measurement, to make my opinions clear.

826 There are two ways in which this extent might be measured:

- It might be measured by the expected net financial impact of all bugs, over various 'ranges' or scopes - (a) across all branches over the lifetime of Horizon, or (b) across all claimants' branches, or (c) on one claimant's branch in any one month
- It might be measured by the number of bugs over the same scopes - (a) on all branches over the lifetime of Horizon, or (b) on claimants' branches, or (c) on one claimant's branch in any one month

FUJ00083961

827 I shall express opinions about both of these measures. In my opinion the first of the two measures is more useful, for two reasons.

a) There may be a number of bugs which affect branch accounts, but whose likely financial impact is trivial. If the court has to spend time considering these bugs with non-zero but trivial financial impact, it might divert attention from considering the smaller number of bugs with significant financial impact, which could have made a more important difference to claimants' branch accounts. Focus on the financial impact of bugs will help in narrowing the scope of enquiries.

p) The first sense of 'extent' gives a way to assess not only the relative importance of different Horizon bugs, but also to assess the absolute importance of each one, asking the question: does any bug on its own (or a set of bugs considered together) provide a possible account for some significant part of the shortfalls asserted by the claimants?

828 For both measures, what I am able to infer from the evidence will be an upper limit on the extent, rather than an estimate of the extent. This is because there are KELs which record incidents which might be caused by bugs, and which might have financial impact. Here, the word 'might' applies with force; there is often not sufficient information in the KEL to conclude that there definitely was an impact, only to conclude that there might have been. This allows me to estimate only an upper limit on the sums over KELs.

829 There are difficulties in measuring the second extent:

a) The robustness countermeasures were generally designed to detect, and to minimise, the first extent (financial impact) , with less regard to the second extent - because it is less important for the business. Work was prioritised according to financial impact. So what evidence remains relates more closely to the first extent than to the second.

q) There are difficulties of definition of the second extent, which make it difficult to sum in any quantitative sense. Does the definition refer to occurrences of bugs in any branch at any time, or to distinct types of bug, however many times they may occur? If it is the former, it is almost impossible to estimate. If it is the latter, what is it that defines two distinct types of bug, as opposed to variant occurrences of the same type? As one 'type' of bug may be caused by the effects of several apparently unrelated pieces of source code or reference data, this question of distinct types of bug has no simple answer. I have found no succinct and satisfactory answers to these questions.

r) because of these difficulties, the second extent is not simply additive as the first one is. It is not easy to proceed from scope (a) (all PO branches) to scope (b) (all claimants' branches) to scope (c) (any branch in one month)

830 Therefore my opinions about the second sense of extent are a great deal more uncertain, and in my view harder to interpret, than my opinions on the first.

831 For the first sense of extent, I can say more about the absolute scale of measurement. There are 561 claimants. Each one has provided in section 8.1 of their claim summary the amount they have repaid to PO; and in section 3.1, a list of the separate shortfalls they experienced, with dates or date ranges for each shortfall. For a few

FUJ00083961

claimants, the amount repaid exceeds the sum of individual shortfalls, so I need to assume that the list of individual shortfalls is incomplete. If we make the sum of shortfalls equal or exceed the repaid amount for each claimant, the sum of all shortfalls experienced by all claimants is approximately £18.7 million. Without making that correction, it is about £17.1 million.

832 This provides an absolute scale of measurement for any bug or set of bugs in Horizon. Does that bug, or set of bugs, offer a possible account for shortfalls which would make up a significant part of the £18.7M?. When assessing the absolute significance of any bug or set of bugs, I shall apply that criterion.

833 The same criterion can be expressed in a different way. The period of tenure of each claimant in months is known. The sum of these tenure periods over all claimants is just over 52,000 months . Therefore the shortfalls experienced by the claimants amount to an average shortfall of just under £360 for each month of their tenure. For each Horizon bug, therefore, we can ask: (a) over what proportion of the 20 year lifetime of Horizon was that bug active? and (b) during that time, what might have been its average impact on branch accounts, compared to £360 per month?

834 In this way , for any bug or set of bugs, my report is intended to assist the court in measuring those bugs against the claim that bugs in Horizon accounted for a significant part of the shortfalls the claimants experienced.

835 To understand the phrase 'bugs, errors or defects', I need to include various classes of defect, including:

- Software errors in Horizon source code, either in the branch or in the back end, either developed by Fujitsu or in some underlying software product they used (such as Riposte or Oracle)
- Errors in reference data, most of which was maintained by PO staff and which determined how the Horizon software operated
- Errors in operational use of software in the back end - such as errors in scheduling of batch jobs in the back end

836 I shall consider all of these to be included in the definition of Horizon issue 1.

837 There are a number of different ways of assessing the financial impact of bugs, which I shall address in the following sub-sections:

a) The impact of the three Horizon bugs addressed by Mr Coyne in paras 5.4 - 5.14 of his report.

s) Assessments of the net impact of all bugs referred to in KELs

t) Data on claimants' shortfalls as provided by the claimants, to assess the impact of all Horizon bugs (known and not identified) on claimant's branches

u) Evidence cited by Mr Coyne

838 The most important of these analyses , and the ones which gives the most clear-cut result, are the analyses under (b) (which in any case include the analysis under (a)). The first of these results in an upper limit on the impact of all Horizon bugs on claimants' branch accounts. The limit is very small - less than 0.1% of the total shortfalls experienced by the claimants. This implies that bugs in Horizon cannot have accounted for the claimants' shortfalls.

FUJ00083961

839 The analysis by method (c) is presented only as a backup of the analyses in (b). The upper limit which it leads to is larger than the limit from method (b) (being approximately 8% of the total shortfalls experienced by the claimants). But it has the merit of being based on completely independent evidence (that provided by the claimants themselves), and so provides independent confirmation of the result from (b).

840 ~~I note that leg (b) of Horizon issue 1 (' undermine the reliability of Horizon accurately to process...') overlaps strongly with leg (a) ('cause apparent or alleged discrepancies or shortfalls....'), since apparent shortfalls arise as a result of inaccurate processing. Therefore I include consideration of leg (b) with my consideration of leg (a).~~ [I now need to stress the core audit process more for (b) - needs a sub-section of its own]

8.5 Scaling of Financial Impacts of Bugs

841 In the previous sub-section, I stated my intention to assess the financial impact of bugs in Horizon over three different scopes:

a) Across all PO branches, during the lifetime of Horizon

b) Across all claimant branches, while they held them

c) On a single claimant branch in a single month.

842 It is possible to relate the financial impacts on these scopes, by numerical scaling factors. I calculate those scaling factors in this sub-section.

843 Over the period 2000 - 2018, the PO network has consisted of more than 11,000 branches. The mean number of branches in all years over the period has been about 13,560. This figure is derived from the spreadsheet referred to at paragraph 178 of Angela Van Den Bogerd's witness statement. Therefore, the number of 'branch months' (a single branch, trading for a single month) has been 13,560 * 12 * 19 = 3,091,680.

844 This means that for a typical PO branch, the scaling factor between scope(a) and scope (c) is a factor of 3 million.

845 For claimants' branches, rather than typical branches, the scaling factor of 3 million may need to be adjusted for two possible effects:

a) It might be asserted that claimants' branches are more likely than other branches to be hit by bugs in Horizon, because of some special property of claimants.

b) Claimants' branches may, on average, be smaller or larger than typical branches across the PO network. If they are smaller, they handle fewer transactions in a month, and so are less prone to Horizon bugs in those transactions.

846 It seems inherently implausible to me that there is some special factor about claimants' branches, which makes them much more prone to bugs in Horizon - bugs which one would expect to strike any branch at random. Nevertheless, I have considered the possibility carefully in section 8.12. I have shown there that there is no significant difference between claimants' branches and other branches, in proneness to bugs in Horizon.

FUJ00083961

847 It appears, from the spreadsheet attached at paragraph 179 of Angela Van Den Bogerd's witness statement, that the claimants' branches are, in terms of customer transactions carried out per day, considerably smaller than the average across the whole PO branch network. The basis of this finding is as follows:

- From summing rows of the spreadsheet, the 561 claimants' branches carried out 558,000 customer transactions per week in 2007.
- This is 558,000/6 = 93,000 transactions per day.
- Across 561 claimant branches, this is an average of 93,000/561 = 165 customer transactions per branch per day.
- Across the whole PO network of 11,000 branches, there are approximately 6 million customer transactions per day.
- This is 6,000,000/11,000 = 545 customer transactions per branch per day
- So, in terms of customer transactions per day, the typical claimant branch was smaller than the average PO branch by a factor 165/545 = 0.30

848 So claimants' branches, being generally smaller than the PO average, have fewer transactions per month, and so are less likely to experience a Horizon bug in a given month. This increases the scaling factor above, between scopes (a) and (c) from about 3 million to about 9 million.

849 I can illustrate what 9 million means using a hypothetical example of a bug which has occurred 16 times over the lifetime of Horizon, with mean financial impact on these occasions of £1000. Call this Bug A. The financial impact of Bug A is similar to that of the Suspense Account bug.

850 If we selected a branch and a month at random, then the chances of Bug A occurring at that branch in that month are only 16 in 9 million - an extremely small probability.

851 Different types of bug occur independently, so their probabilities are additive. If there were a second bug similar to the hypothetical bug above - call it Bug B - then the chances of either Bug A or Bug B happening to one branch on one month are twice the previous figure - 32 parts in 9 million.

852 If there were 100 similar bugs - called Bug A, Bug B, Bug C, ... Bug Zz - the chances of any one of them happening to one branch on one month are still only 100 * 16 parts in 9 million, or one part in 6,000. This is still a very small probability.

853 It then follows that in order for one occurrence of a bug, of similar financial impact to the Suspense Account bug, to have even a one-in-ten chance of occurring to one branch on one month, there would need to be 64,000 such distinct bugs - because 64,000 * 16 / 9,000,000 = 900,000/9,000,000 = 1/10. There would have to be a Bug A, Bug B, Bug C, and so on, in a list with 64,000 distinct bugs.

854 I have made this calculation in an Excel spreadsheet, which is attached to my report. For convenience it is shown here:

FUJ00083961

Item	Label	Central Estimate		Source
Mean number of branches in PO network	A	13560		Spreadsheet with AVDB WS
Years lifetime of Horizon	B	19		2000 to 2018
Total branch months	C	3091680		C=A*B*12
Claimants branch size/typical branch size	D	0.3		Spreadsheet with AVDB WS
Scaling factor from all PO to one claimant branch month	E	10305600		E=C/D
Number of occurrences of Suspense Account bug	F	16		Evidence on Suspense Account bug
Mean financial impact per occurrence of Suspense Account bug (pounds)	G	1000		Evidence on Suspense Account bug
Chance of Suspense account bug occurring to a claimant's branch in one month (chance is 1 in N , where N is shown)	H	644100		H=E/F
Number of different bugs, similar to the Suspense account bug, needed to give 1 chance in 10 of causing a shortfall of £1000 in a claimant's branch account in any given month	J	64410		J = H/10

855 The claimants have never asserted that there are as many as 64,000 distinct bugs in Horizon, each one of them on the same scale of financial impact as the Suspense Account bug. Their case rests on two or three known bugs of this scale, and on the unproven assertion that there may be others.

856 Therefore, if any claimant were to assert that, for instance, a deficit of £1000 had occurred in his branch in a particular month, caused by a bug similar to the Suspense Account bug, the chances of that assertion being correct are extremely small, because Horizon bugs strike so rarely - unless Horizon contained of the order of 64,000 distinct bugs of that kind.

857 Mr Coyne has examined more than 5,000 KELs, and not found definite evidence for even one bug with impact similar to the Suspense Account bug - let alone 64,000 of them.

858 The converse of this result - the very small probability of any error in one months' accounts from a bug in Horizon - is that the accounts for any branch on any month are overwhelmingly likely to be correct (apart from effects such as delayed TCs, which are corrected later).

859 In my experience, no commercial IT system could ever go live with as many as 64,000 serious bugs - and certainly could not have the good in-service record over 18 years that Horizon has had.

860 The claimants' assertion - that some significant part of their losses was caused by bugs in Horizon - is actually even more implausible than I have described. It would require not just 64,000 distinct bugs, each with large potential impact on branch accounts - but 64,000 bugs, each of which evaded the many countermeasures built into Horizon, in order to affect branch accounts (and for branches not to have been compensated, as they were in the case of the Suspense account bug).

861 Having calculated the scaling factor of 9 million between scope (a) of the extent of bugs (impact across all branches in all the lifetime of Horizon) and scope (c) (impact on one claimant's branch in one month), it is fairly easy to relate these two scopes to scope (b) (impact across all claimant's branches).

FUJ00083961

862 Evidence submitted by the claimants implies that claimants ran branches for a total of just over 52,000 months. Therefore, the scaling factor between scopes (b) and (c) is a factor of 52,000. Alternatively, the scaling factor between scope (a) and scope (b) is 9,000,000/52,000 = 170.

863 I shall call this ratio - the amount by which the impact of bugs in Horizon (a) on all PO branches is expected to be larger than their impact (b) on all claimant's branches - the **claimant scaling factor**. I shall take it to be 170.

864 There is one immediate consequence of this scaling factor. The total of all shortfalls experienced by the claimants is £18.7 million. If, as the claimants assert, some large part of this (for illustration, say 50%) was caused by bugs in Horizon, the total impact of these bugs across all PO branches would be £18.7M * 0.5 * 170 = £1,618 million. This figure, of almost £2 billion, is the sum which PO would have gained from its SPMs, through bugs in Horizon. In my opinion, as will be explained below, it is very unlikely that there could have been be such a gain.

8.6 Analyses of the Three Errors Cited By the Claimants

865 I have analysed these three bugs using evidence available to me, and then on November 18th I received the second witness statement of Torstein Godeseth, which addresses those three bugs, and reaches conclusions similar to my previous conclusions. Where relevant I comment on Mr Godeseth's conclusions.

8.6.1 The Receipts/Payments Mismatch Issue

866 This issue is cited in paragraph 5.6 of Mr Coyne's report. It involved a bug in Horizon which was triggered by a rare circumstance (which one would not expect to be exercised in testing) and which had an effect on branch accounts.

867 At paragraph 35 of Mr Godeseth's witness statement, he says this bug was detected by routine monitoring of system events by the Fujitsu System Support Centre (countermeasures RDS, MID). In the same paragraph, he says that the bug would also show as a discrepancy in POLSAP (another example of RDS)

868 This incident involved a complex sequence of events during branch balancing, which was not detected by some of Horizon's resilience countermeasures, but which was detected by others. It was later the subject of thorough investigation by Fujitsu.

869 The bug only occurred when the user followed a rare sequence of actions during branch balancing. This sequence was to cancel one part of the balancing process (balancing one stock unit), but to proceed after that with balancing other stock units. While not actually forbidden, this sequence of actions would usually have occurred only when the user had misunderstood what he was doing (thinking he was balancing for a 4-5 week Trading Period, when he was actually balancing for a 1-week Balancing Period). Hence the circumstance was rare and had not been exercised in testing.

870 The effect of the bug was to record erroneously in the BRDB that the stock unit whose balancing had been cancelled was in balance; while also recording the imbalance on other systems (including POLSAP). The analysis of the bug in terms of Horizon's robustness countermeasures is as follows:

- Because the bug occurred in one of the countermeasures (the correction of user errors by branch balancing, UEC), that countermeasure created the error, rather than correcting it.

FUJ00083961

- Because the operation involved was not a double-entry operation on the BRDB, the countermeasure of checking the double-entry constraint DEA did not catch it.
- Horizon kept redundant copies of the information involved and checked them. The checks from this countermeasure produced error messages in logs (RDS).
- Once Fujitsu were alerted to the error, they were able to look at the error messages to find which branches were involved and the amounts involved (MID), and to find out the causes of the error.
- The error was fixed within about 2 months of its first occurrence (BFC).

871 This is a fairly complex incident, which illustrates how many resilience countermeasures there are in Horizon, and how even when some of them do not catch an error, others will do so. Most of the other KELs which I have examined tell a simpler story of how one or other countermeasure trapped some error (an error in software, or a user error).

872 The net quantitative impact of the receipts/payments mismatch was approximately £20,000 across 62 of the 11,000 branches. Paragraphs 42 and 43 of Mr Godeseth's witness statement describe how the branches were compensated. So in the event, no SPM suffered any loss.

873 Because this shortfall was carefully investigated, we know that none of the claimants' branches was affected. If we did not have this knowledge, we would expect on statistical grounds that the net effect on all claimants' branches would be £20,000 divided by the claimant scaling factor, described in section 8.5. The claimant scaling factor is 170.

874 This amount would be approximately £120 spread across all claimants, less than 0.001% of the full shortfall experienced by all claimants. Thus even a prominent and thoroughly investigated bug would have made no significant contribution to the claimants' shortfalls.

875 This accords with the previous result, noted in section 8.5, that a very large number of similar bugs (of the order of 64,000 or more) would be needed to account for the claimants' shortfalls.

8.6.2 The Callendar Square/Falkirk Bug

876 The Callendar Square bug is described in two KELs, JBallantyne5245K and JSimpkins338Q, and in several Peaks PC0075892, PC0083101, PC0086212, PC0103864, PC0126042, PC0126376, and PC0193012. It first arose in 2000, and was not fixed until release S90 in 2006. I shall describe the nature and effects of the bug, based on the KELs and an analysis by Gareth Jenkins, and then summarise my opinion on the significance of the bug for Horizon Issues 3 and 1.

877 The cause of the bug was a failure of data replication in the underlying Riposte software in the branch, which sometimes occurred when transferring stock between stock units. This was caused by a timeout or locking problem somewhere inside the Riposte product, which it was not possible for Fujitsu to fully understand at the time, or for the experts to understand now. Internal design details of system software products like Riposte are generally not made available to developers who use that software.

FUJ00083961

878 The result was typically that the stock would disappear from the sending stock unit, and not reappear in the receiving stock unit - a failure of double entry accounting (DEA) which was not evident to the SPM at the time. At paragraph 13.6 of his witness statement, Mr Godeseth comments on this failure of double entry accounting.

879 In my opinion , under the later HNG software this failure of DEA might have been immediately manifest as a failure to send a zero-sum basket to the BRDB. But in Horizon, apparently it was not immediately detected, so in this respect Horizon was possibly less robust than HNG.

880 While the failure was not immediately visible to the SPM at the time of the stock transfer, it would always be visible later when balancing. stock units. It was also, as Mr Godeseth says at paragraph 13.7 of his second witness statement, soon visible to Fujitsu in two different ways (a flag from overnight processing, and a system event). This was robustness through the countermeasure RDS.

881 So in the normal course of events, the SPM would see a discrepancy of some large and easily identifiable sum (because stock unit transfers generally involve larger sums than customer transactions) and would know, since he had not made any mistake, to call the help desk. This was countermeasure MID. As is shown be the Peaks, the presence of the Riposte error was easily identifiable from system logs, so the help desk would know it was not a user error and could correct any discrepancy in the branch accounts.

882 Thus for any SPM who was in good control of his branch numbers, and alert to discrepancies, there was little chance of this bug leading ultimately to an error in his branch accounts; he would require it to be corrected. This was ensured by a combination of the countermeasures RDS (e.g. in event logs) and MID (by both SPMs and Fujitsu)

883 At paragraph 15 of Mr Godeseth's witness statement, he says that this bug had impact on branch accounts in 20 cases. Because Fujitsu could always spot any occurrence of the bug in event logs, and because neither PO or SPM wanted SPMs to suffer shortfalls from bugs in Horizon, I would expect the SPM to be left with a shortfall (i.e. not compensated) in only a small minority of cases, if any cases. So I would expect that the net shortfall caused by all its occurrences would be possibly zero, and in any event at most a few thousand pounds.

884 Because Fujitsu had designed the counter software assuming that Riposte replication worked correctly, and could not anticipate in what ways it might not work, in my opinion it would have been very difficult for Fujitsu to fix the problem or correct it. Fujitsu were reliant on Escher to fix the problem; and apparently Escher did not do this for some years.

885 To summarise my opinions on the significance of the Callendar Square bug for Horizon issues 3 and 1:

- It was not detected immediately by the countermeasure DEA, when in my opinion it might have been detected (although possibly the Horizon architecture, dependent on Riposte replication, made this very difficult)
- However, it was later detected in branch balancing, and corrected if necessary, by the countermeasures RDS and MID. Overall, Horizon's robustness worked well.

FUJ00083961

- Therefore, like the other two known bugs addressed in this sub-section, its possible financial impact on claimants' branch accounts was very small indeed
- It took several years to fix, but in my opinion this does not reflect badly on Fujitsu. They were reliant on the Riposte product supplier, Escher, to fix the problem.

8.6.3 The Suspense Account Bug

886 This is referred to in paragraph5.12 of Mr Coyne's report. Like the receipts/payments mismatch, it concerns the process of balancing and rollover of stock units, using suspense accounts.

887 It was analysed in depth by Gareth Jenkins of Fujitsu. His summary of the effect was: *'The root cause of the problem was that under some specific, rare circumstances some temporary data used in calculating the Local Suspense was not deleted when it should have been, and so was erroneously re-used a year later'.*

888 To understand the 'specific, rare circumstances', the effect of robustness countermeasures, and the financial impact, it is necessary first to understand what was supposed to happen, and then to understand how it went wrong.

889 When certain types of data in the branch database are no longer needed for trading, they may nevertheless need to be retained for some time for various purposes, or be kept for longer in an archive. Ultimately, the space taken up by those records in the BRDB needs to be recovered, to stop the database growing without limit. There are therefore policies for initially making records inactive by a process of 'logical deletion' (which means, not actually deleting the record, but marking it as inactive and due for later deletion), for archiving of data, and for ultimate physical deletion of records.

890 These policies are different for different tables of the BRDB. They may need to change from time to time. When they do, there is a risk of transient problems - when the time window of some archiving and deletion policy changes, and records in some table fall between windows. This is what happened in the suspense account bug.

891 A branch will from time to time want to stop using a stock unit. When it does so, the records in the BRDB for the stock unit are first 'logically deleted' - to prevent the stock unit being reused before it can be reused - and later archived and physically deleted. After that, the stock unit can be reused (i.e. another stock unit with the same identity can be created and used).

892 There was a change in the archiving policy in late 2010, which meant that for a short period, the balancing discrepancies for some stock units which were to be deleted became 'orphaned', and escaped the process of archiving and physical deletion. This meant that if, after archiving, the stock unit was recreated and reused, the balancing discrepancy associated with the old stock unit (which had actually been cleared, before the old stock unit was deleted) became wrongly associated with the new stock unit a year later (when the same trading period number between 1 and 12 came up in the next year) - and the SPM was asked, wrongly, to clear it again.

893 Usually, this error would have a small financial impact and would be hard to detect. This is because the balancing discrepancy in a stock unit is expected to be small - arising from mis-counting of stock or cash; so that if the same discrepancy was then wrongly added to another discrepancy a year later, the difference might not be

FUJ00083961

noticed. For a small discrepancy, an SPM may choose to just accept it, rather than make the effort to try to understand it.

894 Of the 14 branches affected by this bug, for 12 of them the amounts were less than £161 and the SPM did not raise any query. However, for two of the branches the amounts were larger, of several thousand pounds. These large discrepancies arose not because those branches were very bad at counting stock, but for a different reason.

895 Normally, a stock unit should accurately reflect both the cash and the stock in it. When preparing to delete a stock unit, the SPM is expected to move all stock and cash from it into other stock units in the branch, using facilities in the counter software to do this. However, there is a short cut. Because the stock in a branch is not necessarily physically segregated into different stock units, the SPM might make no transfer of stock - but instead, when the stock is sold, simply credit the cash to another stock unit. Then the other stock unit will have a surplus in its cash and stock, and the stock unit to be deleted will have a deficit. These two discrepancies will cancel in the monthly balancing process, so there is no cost to the SPM.

896 This short cut was not recommended by PO in the branch trading manual [check], but was available to anyone who understood how balancing worked across several stock units - that is, to most SPMs. It saved the trouble of moving stock between units. One of the branches affected had used this short cut, and had a large deficit in the stock unit which was to be removed. Because of the archiving problem, that large deficit wrongly reappeared a year later.

897 The description above agrees with the account at paragraph 48 of Mr. Godeseth's second witness statement.

898 The circumstances for this bug to have a large financial impact on a branch were rare in four respects:

- A branch wishes to stop using a stock unit - which only happens occasionally
- The stock unit is terminated during a short overlap period, caused by a change in archiving policy
- The branch then wishes to re-create the stock unit (to reuse its identity) - another occasional event
- when 'running down' the stock unit before terminating it, the SPM used a short cut leading to a large balancing discrepancy

899 In combination, these circumstances were so rare they had large effects (greater than £200) on the accounts of only two branches.

900 We can consider what this bug implies about the resilience measures in Horizon:

a) The archiving of BRDB data is not done in double-entry transactions, and the table used to compute the initial discrepancy in a stock unit (which was a table used to prepare the BTS) was not subject to double-entry constraints. So the DEA countermeasure did not catch the error.

c) However, the status of each stock unit was also redundantly held on POLSAP, which did apply double entry constraints and did not have the same archiving policies; so once the error was discovered, the true position for affected branches was easily discovered from POLSAP and corrections were manually applied to the BRDB. This was an example of redundant data storage (RDS), using manual inspection of data (MID) followed by a successful manual workaround (WOR).

FUJ00083961

d) Because there was no automatic cross-check of stock unit positions between POLSAP and the BRDB, the error was not detected automatically by RDS.

e) There was a delay of a year in manually finding the bug and correcting it - because when the branch most affected reported the problem to PO, PO simply corrected the data centrally and did not inform Fujitsu. It was only a year later, at the next recurrence, that Fujitsu became aware of the problem and fixed it. This business process problem between PO and Fujitsu was a lapse in bug fixing and correction (BFC)

f) In spite of the bug, the core audit process maintained an accurate record of what had happened in the branches (another redundant copy, stored in a secure kernel, SEK), which was useful in diagnosing and fixing the problem.

901 Therefore none of the automatic robustness countermeasures detected the bug. It was detected by an SPM raising a query (as one would expect to happen, for any financially significant effect), and in diagnosing, correcting, and fixing it, redundantly-held copies of the data played an important role.

902 Because the affected branches were easily identified, and the impact was manually reversed for those branches (i.e. settled centrally), no SPM was adversely affected by this bug. Without this correction process, the financial impact of the bug would have been about £10,000 across the 16 branches. As before, no claimant branches were involved [check]; but if there had been other comparable bugs, their expected impact on the accounts of claimants' branches would be £10,000 divided by the claimant scaling factor, which is 170 (as described in section 8.2), or approximately £60 across all claimants Compared to the total shortfall of £18.7 million suffered by all claimants, this figure is approximately 0.0003%.

8.6.4 Opinion on the Three Identified Bugs

903 Because these three bugs all led to effects on branch accounts, they were all investigated carefully, by staff in Fujitsu with a deep knowledge of Horizon. These investigations have made it clear to me that, as I would have expected, Horizon is a highly complex system, and to fully understand some errors in it (and the robust handling of those errors) requires a knowledge of many of its component systems to some depth.

904 The experts have not had the time to do this deep analysis for more than a few errors, including these, and it would be unrealistic to expect the reader to understand these to the same depth.

905 The conclusions I draw from analysing these three bugs are:

- There are extensive robustness countermeasures in Horizon, of many types - so that even in the rare case of bugs like these which are not handled by the fully automatic countermeasures, manual countermeasures enable the bugs to be rapidly diagnosed and corrected, as soon as they are known about.
- Any error in Horizon whose financial impact is greater than about £1000 in one month is likely to be reported by a significant proportion of the SPMs who experience it, and so to be the subject of manual investigation. In all the three cases considered here, this manual investigation was successful - which further confirms the robustness of Horizon.

FUJ00083961

- The expected financial impact of these bugs taken together on claimants' branch accounts was only a fraction of a percent of the total shortfall experienced by all claimants. So bugs like these, even if a very large number of them existed (which I have seen no evidence for), cannot account for the claimants' shortfalls.
- In the two cases where we know there were some shortfalls in branch accounts (Receipts/Payments Mismatch, and Suspense Account), the branches were compensated for the losses.

906 The last point seems to me to have a general bearing on the claimants' case. There are two kinds of bugs- known bugs, which were noted and detected at the time, and 'unknown bugs' which were not detected at the time. In section 8.2, I showed that the financial impact of unknown bugs is very small. If, as these examples imply, in the case of known bugs the SPMs were compensated, then known bugs would lead to little or no personal loss to SPMs .

8.7 Financial Impact of All Bugs - Main Analysis

8.7.1 Method of Analysis and Conclusions

907 In section 7 of this report, when discussing robustness countermeasures in Horizon, I reached two conclusions:

b) If any bug had impact on branch accounts, (with the exception of micro-bugs, discussed later in this section), on a significant proportion of the occasions when it occurred, it would be reported to the help desk by the SPM

g) Any anomaly reported by an SPM which had the potential to affect branch accounts would, with fairly high probability, result in a KEL and an investigation by Fujitsu

908 It is therefore possible to use the KELs as a measure of the extent and financial impact of bugs in Horizon - by counting all the KELs which might be bugs with financial impact, summing their maximum possible financial impact, and making allowances for any inefficiencies in the processes (a) and(b).

909 As will be clear from this sub-section, the analysis relies on evidence from KELs rather than other documents. I have examined a large number of KELs which are not cited directly here, but my analysis is contained in tables in an Appendix to this report.

910 The numerical analysis is in principle quite simple. It is to count the bugs in KELs; sum their financial impact; and make allowances for inefficiency in recording bugs in KELs.

911 In practice, the analysis is made more complex by two factors:

- There are more than 8300 KELs, and it has not been possible in the time available to analyse all KELs to the depth required; so I have had to analyse the KELs on a sampling basis, and to correct the final result for the sampling.
- KELs, and the Peaks they refer to, are not a complete record of the nature of a bug, or its investigation, or its financial impact, or the branches affected; that is not their purpose. KELs and Peaks assume a deep familiarity with Horizon, and are often written in shorthand which assumes that knowledge (as is described at paragraph 66 of Mr. Parker's witness statement) . To infer financial impact from the KELs and Peaks, I

FUJ00083961

have had to make inferences, sometimes with a degree of uncertainty. I have allowed for this uncertainty by expressing the results as upper limits on the financial impact of bug, and attempting to make these upper limits conservative.

912 Having done this summation of the maximum possible impact of bugs in KELs, to find the maximum likely financial impact on all claimants' branches, it is necessary to scale the summed financial impact, to account for the following factors: (a) inefficiencies in the KEL creation process, where a Horizon bug existed but did not give rise to a KEL; (b) limitations of the sampling of KELs that I have been able to do in the time available; (c) the claimants' branches being a small proportion of the total PO branch estate. In calculating each of these scaling factors, I have attempted to be conservative, to reach a result which is most favourable to the claimants, and least subject to changes in assumptions.

913 In my opinion, this analysis gives the simplest and most direct route to estimate the total impact on the claimant's branch accounts of all bugs in Horizon, known and unknown..

914 I believe there is no simpler way to estimate the impact on claimants' accounts of all Horizon bugs, than to sum the maximum financial impact of all those I can find, and then to correct the resulting number for those I have not been able to find.

915 As I shall describe below, this analysis is still at an interim stage, as I have only received certain important inputs to it on November 17th with the defendant's witness statements, while finalising my report. As a consequence, as I shall describe below, the results are derived not just from one sampling of the KELs, but from three separate samples. Combining the results from these separate samples is not straightforward, and I have used only a simple and conservative way to combine them (conservative in the sense that it tends to favour the claimants) to derive the main result.

916 The conclusion of this analysis is that the total impact of all Horizon bugs on all claimant's accounts is probably not more than £20,000, with a high degree of confidence, compared to the total shortfall of approximately £18.7 million reported by the claimants.

917 Therefore, in my opinion, bugs in Horizon cannot account for more than 0.1% of the claimants' shortfalls. This conclusion is robust against the limitations of my analysis to date, but I shall continue to improve its accuracy. [the number may change with further analysis, but I do not expect it to change much].

8.7.2 Reporting of Anomalies and the Creation of KELs

918 In section 7.5 of this report, when discussing the effect of robustness countermeasures on Horizon bugs which might affect branch accounts, I estimated the probabilities that SPMs would report the occurrences of bugs, based on evidence from the suspense account bug, and other considerations (part of the MID countermeasure - MID by SPMs). I will use those estimates here, but before doing so, will make them more conservative - adjusted to be more in favour of the claimants - so that the court may place more reliance on the results.

919 The estimates I made there - together with the more conservative estimates I shall use here for calculation - are as follows:

FUJ00083961

920 I assumed, as 'middle of the road' best assumptions of SPM behaviour in reporting anomalies in their monthly balancing:

c) If a discrepancy is £1000 or more , the SPM probably needs to investigate it. If he cannot find the cause in his branch, the likelihood of his reporting it through a help line is 80% (for this computation, assume only 40% of SPMs report)

h) If a discrepancy is of the order of £300, 30% of SPMs will report it (for this computation, assume only 15%)

i) If a discrepancy is of the order of £100, 10% of SPMs will report it (for this computation, assume only 5%)

j) For a discrepancy of £10 or less, it is usually not worth the SPM's time to investigate it (because errors in counting cash or stock are often larger than this); so these are reported on less than 1% of occasions.

921 Thus the estimates used for calculation are twice as conservative as my central estimates, to favour the claimants.

922 Next consider a bug in Horizon, whose total net financial impact on SPMs, from all the occasions when it occurred, was £10,000 or more. For instance, this might have happened in 10 occurrences of £1000 (case(a)), or 33 occurrences of £300 (case (b)), or 100 occurrences of £100 (case(c)), or 1000 occurrences of £10 (case (d)). For the moment I exclude the last 'micro bug' possibility. I will address it later.

923 Had the bug been immediately evident to the SPM, then it would have been reported on many occasions, regardless of the size of its financial impact.

924 If the bug was not immediately evident, but its effects were only evident to the SPM in monthly balancing, I shall use the conservative estimates above of the probability of the SPM reporting each occurrence. Using those estimates, this bug would have been reported by SPMs on 10*0.4 = 4 occasions in case (a), on approximately 33*0.15 = 4 occasions in case (b), or 100*0.05 = 5 occasions in case (c).

925 It is then clear that in a mixture of these cases - where the bug occurred with variable financial impact on each occasion - it would still have been reported about 4 times. Therefore the probability of it not being reported at all is small - say less than 10%. It would have been reported at least once, with probability 90%.

926 This is my best estimate 0.9 of the probability that some SPM will report any bug, if it has impact on their branch accounts. For the purposes of calculation, I shall use a more conservative estimate of 0.7. In my opinion, it would strongly favour the claimants to assume a 30% chance that no SPM will report such a bug

927 Also in section 7.5, I estimated the probability that , when being informed of an anomaly which might lead to an error in branch accounts, Fujitsu would create a KEL. I estimated that these processes resulted in a KEL on more than 90% of the occasions where it was reported and there might be some effect on branch accounts.

928 Again, I shall use a more conservative estimate for the purposes of calculation. I shall assume only that Fujitsu created a KEL on 50% of the occasions where a reported anomaly might affect branch accounts. From evidence I have seen about Fujitsu's processes, this seems a very conservative assumption.

FUJ00083961

929 This means that for any bug whose impact on all branch accounts was £10,000 or more, the probability of it leading to a KEL was 0.7*0.5 = 0.35. At least 35% of such bugs appeared in KELs. Any bug with smaller financial impact - right down to zero - had some probability of appearing in a KEL - especially if it was immediately visible to the SPM. At least 50% of those bugs would appear in KELs.

930 This means that if I search for KELs describing bugs with financial impact, amongst the 8300 KELs that have been provided to me and those that have been archived, I will find evidence of at least 35% of all bugs with financial impact £10,000 or more, and for 50% of any bugs which were immediately evident to the SPM, regardless of the size of their impact. If I search only the 8300 KELs which have not been archived, the figures should be reduced by a factor 0.85 to account for the archived KELs.

8.7.3 Analysis of KELs with Possible Financial Impact

931 As I mentioned above, because of limited time to prepare my report, my analysis of KELs is at present based on three separate samples, which I shall describe in turn. These samples are:

- A sample of 80 KELs selected from the 8390 KELs at random
- A sample of 50 KELs all containing the symbol '£' and therefore more likely to concern financial impact - which I have analysed and subsequently Fujitsu have analysed
- The sample of 5111 KELs examined by Mr Coyne

932 Because of the very different types of analysis applied to these three samples, it has not been possible to combine them in any straightforward numerical manner, and I am only able to combine the results in a conservative manner which favours the claimants.

Randomly Chosen KELs

933 For my first sample, I selected 80 KELs in a pseudo-random manner from the 8390 KELs (in practice I chose every 100th KEL from an alphabetically sorted list of KELs, so as to avoid any possible bias in my choice). My analysis of these KELs is given in Appendix D [?]

934 For the great majority of these KELs, it was immediately obvious either that they were not bugs in Horizon, or that they would have no effect on branch accounts.

935 For the remaining KELs, I used my knowledge of the countermeasures to assess which countermeasures were applied, and whether or not they would prevent any impact on branch accounts. As a result, I found no KELs with possible impact on branch accounts.

936 From such a limited sample (one KEL in every hundred) I cannot conclude that there were no KELs with any impact on branch accounts in the whole set of KELs.

937 However, if there were 200 such KELs in the whole set (i.e. one KEL in every 40), it is statistically very likely (a chance of about 90%) that at least one of them would have got into my set of 80.

938 From, this analysis, I conclude that there are probably not more than 200 KELs which relate to bugs with impact on branch accounts.

FUJ00083961

KELs including the symbol '£'

939 For the next sample, I have used all those KELs whose text includes the symbol '£'. I chose this set because in my opinion, a KEL with some possible financial impact is more likely to contain the symbol '£', so looking at these KELs may be a faster way to find some KELs with impact of branch accounts. These KELs are easily found by a Windows search, and there are 259 of them. In the time available to me, I have been able to examine 50 of these 259 KELs. The results of this examination are:

- For 42 of the 50 KELs, either the KEL does not arise from a bug in Horizon; or if it does, there is no possible impact on branch accounts
- For 8 of the 50 KELs, there is a possible bug with possible impact on branch accounts. I found this by going some way in granting the benefit of the doubt in this respect to the claimants, and admitting the possibility of financial impact even if in my view it was remote.

940 So the sample has borne out my opinion that KELs which mention the symbol '£' are more likely to have financial impact, than KELs chosen at random.

941 Since I made this analysis, the same set of 50 KELs was passed to Fujitsu, and they analysed them. The results are appended to the witness statement of Steve Parker, which I received on 18th November.

942 Because Fujitsu have a deep knowledge of Horizon, of the usage of KELs, and the terms used in KELs, their analysis is likely to be more accurate than my own.

943 For the 42 KELs which in my initial opinion had no impact on branch accounts, Fujitsu's analysis agreed with my own.

944 Of the 8 bugs which in my opinion might have had impact on branch accounts, Fujitsu found that only 4 of them had that potential. This also agrees with my analysis, because I was giving the claimants the benefit of the doubt, and Fujitsu have more information to remove doubt.

945 Unfortunately, because I do not yet know by how much the inclusion of a '£' symbol would increase the chances of a KEL signifying a financial impact, I have no way of scaling up Fujitsu's result of 4 KELs with possibly financial impact to the whole set of 8390 KELs. However, it does give 4 more KELs with possible financial impact, to help to assess the extent of that impact.

KELs examined by Mr. Coyne

946 In paragraph 5.114 of his report, Mr. Coyne says: '*Regarding the extent of potential errors within Horizon I have analysed 5114 Horizon Known Error Logs (KELs) to determine the scope of potential bugs or 'PEAKs' (as they are referred to by Post Office and Fujitsu). Of these 5114, I have found that 163 contain PEAKs that could be of significant interest and of these 76 are referred to in the report*'

947 Mr Coyne does not define what he means by '*significant interest*', but it appears to relate to '*the scope of potential bugs* '.

948 Since Mr Coyne's report does not mention any robustness countermeasures, he has evidently not examined any of these KELs from the viewpoint of those countermeasures , to assess whether they could or could not have had any impact on branch accounts.

FUJ00083961

949 I have examined 62 KELs which I found cited in his report, and I found that fewer than 8 of them might have had financial impact. Fujitsu have examined approximately the same set of KELs, identified by PO's lawyers in Mr Coyne's report, and the result is annexed to Steve Parker's witness statement.

950 With their better knowledge of the significance of the KELs, Fujitsu found results similar to mine - apart from the known receipts/payments mismatch, there were very few uncorrected financial impacts. I have not completed comparing Fujitsu's analysis with my own.

951 Furthermore, I believe I can infer that had any of the 5114 KELs clearly and explicitly indicated a bug with significant financial impact on branches, Mr Coyne would have cited that KEL in his report.

952 On any interpretation, since about half of Mr Coyne's 163 KELs of 'significant interest' have been shown by Fujitsu and by me to have no financial impact, it can be inferred that his search of 5114 KELs has revealed no more than 100 KELs with potential financial impact.

Combining the results of the Three samples

953 My survey of 80 randomly selected KELs revealed none with financial impact. From this I inferred that in the 8390 KELs disclosed to the experts, probably no more than 200 had potential financial impact.

954 From the sample of 50 KELs mentioning the symbol '£', Fujitsu found no more than 4 with potential financial impact. Unfortunately it is not yet possible to scale up this result to the full set of KELs.

955 From Mr Coyne's sampling of 5114 KELs I inferred that no more than 100 have possible financial impact.

956 Taking these results together, I infer that in the set of 8390 KELs, no more than 200 have potential financial impact. This is my conservative estimate, to be used for calculation; my more central estimate is 100.

8.7.4 Mean Financial Impact of One Bug

957 From the previous analysis, I have found only 7 bugs with possible financial impact on branch accounts - the three known bugs cited by Mr. Coyne and analysed in my section 8.6, and the four which Fujitsu found to have potential financial impact. I summarise these in a table, to estimate the mean financial impact of any bug:

Bug or KEL	Commentary	Approximate Financial impact across all PO branches
Receipts/payments mismatch		£20,000
Suspense account		£14,000
Callendar Square		£3000
AChambers2252R		£3000
AChambers4134R		£200
ballantj020J		£300
AChambers253L		£500
TOTAL Impact		£41000

958 My estimates of the financial impact of the KELs which in Fujitsu's view might have financial impact are at present very approximate. For the resulting sum at the foot of the table, all that matters is that none of them are large compared to the impact of the three known bugs.

FUJ00083961

959 From this table, the mean financial impact of any single bug - across all branches in the PO Network - is approximately £41000/7 = £6000.

960 This is a conservative estimate of the actual financial impact of a bug on an SPM, since it is dominated by the two first rows (the known bugs), and in both those cases, all the branches affected were compensated by PO. If I were to allow for the fact that PO compensate branches whenever they are aware of a shortfall caused by a bug, the figure would be much less than £6000. My central estimate is therefore £2000.

8.7.5 Calculation of Financial Impact of All Bugs

961 The result of the analysis so far is:

- There are not more than 200 bugs with financial impact in all the KELs
- Of those, the mean financial impact per bug is not more than £6000

962 This makes a maximum financial impact of all bugs in KELs on all PO branches of 200*£6000 = £1.2 million

963 To find the impact of Horizon bugs in all KELs on all PO branch accounts, these results need to be scaled up by the following factors:

- A factor 1/0.35 as above to allow for the fact that not all anomalies may lead to a KEL - either because they are not reported by the SPM, or because Fujitsu do not create a KEL. This builds in the conservative estimates above
- A factor 1/0.85, to allow for archived KELs, which the experts have not seen
- A factor 1/170, as calculated in section 8.5 (the claimant scaling factor) , to allow for the fact that claimant branches were only a small proportion of the whole PO network

964 The result is an impact of £20,000 across all claimants.

965 In applying these scaling factors, I have assumed that the probability of a Horizon bug striking a claimant's branch, in any given month, is the same as the probability of that bug striking any other branch. I shall examine that assumption later in this section.

966 The total impact of all Horizon bugs on claimants' branch accounts of £20000 is to be compared with the £18.7 million shortfalls that they have claimed. This figure is 0.1% of their shortfalls.

967 This conclusion rests on a number of assumptions, which I have stated and justified when deriving it. In all cases, I have tried to make these assumptions conservative, erring in favour of the claimants, to make the result as reliable as possible.

8.7.6 Summary of the calculation

968 I have made the calculations above in an Excel spreadsheet, which is attached to my report. For convenience, the spreadsheet is summarised below.

FUJ00083961

Item	Label	Central Estimate	Conservative Estimate		Source
Mean number of branches in PO network, 1999 - 2018	A	13560	13560		Spreadsheet with AVDB WS
Years lifetime of Horizon	B	19	19		2000 to 2018
Total branch months	C	3091680	3091680		C=A*B*12
Claimants branch size/typical branch size	D	0.3	0.3		Spreadsheet with AVDB WS
Scaling factor from all PO to one claimant branch month	E	10305600	10305600		E=C/D
total claimant shortfall (pounds)	F	18700000	18700000		claims
total claimant branch months	G	52000	52000		claims
scaling factor from all PO branches to all claimants	H	198	198		H=E/G
KELs with potential impact on branch accounts	L	100	200		RW finding from inspection of KELs
Mean financial impact of KEL with potential impact (pounds)	M	2000	6000		RW finding from inspection of KELs
Summed financial impact of KELs wth potential impact (pounds)	N	200000	1200000		N=L*M, or direct sum of KELs impact
Probability(SPMs report bug with impact|bug occurs)	T	0.9	0.7		Evidence on SPM reporting of anomalies
Probability(FJ create KEL|SPMs report bug)	U	0.9	0.5		Evidence on FJ processes for KEL creation
Probability(FJ create KEL|Bug occurs)	V	0.81	0.35		V=T*U
Probability(KEL is not archived)	W	0.85	0.85		Evidence on KELs archiving
Probablity(KEL is created and not archived|bug occurs)	X	0.6885	0.2975		X=V*W
Summed financial impact of bugs on all PO branches, corrected for KEL sampling,creation and retention (pounds)	Y	290487	4033613		Y = N/X
Summed financial impact of bugs on claimant branches, corrected for KEL sampling,creation and retention (pounds)	Z	1466	20353		Z=Y/H
Financial impact of bugs on claimants, as a percentage of their losses	E1	0.008	0.109		E1 = 100*Z/F

969 The spreadsheet shows two alternative calculations, one from 'central' assumptions (which are my best estimate from the evidence) and one from conservative assumptions, intended to favour the claimants, and used for my main calculation (the right-hand of the two calculation columns, entitles 'Conservative Estimate').

970 It is evident that the conservative result (0.1%) is about ten times larger than the central result (0.008%). Yet the conservative result is still less than one percent of the claimants' shortfalls. Even with highly conservative assumptions, bugs in Horizon can account for much less than 1% of the claimed shortfalls.

971 Throughout the analysis of this sub-section, it has been assumed that bugs in Horizon predominantly cause shortfalls in branch accounts, rather than gains to branches. I note here that if a significant proportion of bugs in Horizon were to cause gains to branches, then the total number of bugs required to produce a given level of shortfall in claimant's branches would be even larger. Of the upper limit which I have calculated on the number of bugs, some bugs would cause gains, and so would reduce yet further the part of the shortfalls experienced by the claimants, which could be accounted for by bugs.

972 It would be straightforward for Mr Coyne to add an extra column to this spreadsheet, to repeat the same calculation with the assumptions that he believes to be correct, and to calculate his version of the estimate E1.

FUJ00083961

8.8 Alternative Approaches to Estimate The Financial Impact of Bugs

8.8.1 Number of Bugs in Horizon Required to Substantiate the Claimants' Case

973 In section 8.5, 'Scaling of Financial Impact of Bugs' , I gave what I now think is the simplest analysis of why, in quantitative terms, bugs in Horizon cannot have accounted for a large part of the claimant's shortfalls. I summarise it here.

974 Because PO have had an average of 13,560 branches over the lifetime of Horizon, the total number of monthly branch accounts has been about 3 million.

975 Therefore, if a bug like the Suspense Account bug has occurred 16 times in the lifetime of Horizon, the chance of it having occurred in any given branch in any given month is about 16 in 3 million. Because the claimants tended to have smaller branches than the average, doing fewer monthly transactions (by a factor 0.3), the chances of the bug occurring in a claimant's branch would be about 2 in 10 million.

976 I have considered a bug similar to the suspense account bug, which occurred about 10 times , and had a mean financial impact of about £1000 per occurrence. How many similar bugs would be needed, to give a one in ten chance of one such bug occurring, with an impact of £1000, on a particular claimant's branch in a particular month?

977 The answer, given by elementary arithmetic which I describe in section 8.5, is that there would need to be 64,000 of these distinct bugs. If there were fewer than 64,000 similar bugs, if any claimant were to assert that in a given month a shortfall of £1000 in his accounts was caused by bugs in Horizon, then the chances of his assertion being correct are less than one in ten.

978 So the claimants cannot credibly assert that their shortfalls were caused by bugs in Horizon, unless there were something of the order of 64,000 such bugs.

979 Only three such bugs have been found. My own search of KELs has found only 8 other possible bugs. Mr Coyne's examination of over 5000 KELs has found no other bugs which definitely caused shortfalls.

980 Thus the claimants' case requires 64,000 bugs in Horizon - but only a handful have been found by the experts. Neither expert can quantitatively support the claimants' case.

8.8.2 Considerations of Call Centre Workload

981 This section contains an alternative analysis of the financial impact of Horizon bugs on the claimants, which rests on some of the same evidence as that relied upon in the previous section, but uses a different approach to calculation.

982 I start from the fact that the sum of all shortfalls experienced by the claimants was approximately £18.7 million, over the period 2000-2018. I next suppose, following the hypothesis put forward by the claimants, that some large part of this amount (say 50%) was caused by bugs in Horizon. This gives a total claimants' shortfall, caused by bugs in Horizon, of approximately £10M during the period. (I assume this as the start of an argument of *reductio ad absurdum*, which follows.)

FUJ00083961

983 I next assume that bugs in Horizon would affect all branches - claimants and others - in approximately equal measure - as any bug in Horizon would affect all transactions across the PO network at random. (I shall examine this assumption in a later section).

984 If that assumption is correct, then the losses from bugs in Horizon suffered by all branches over the lifetime of Horizon, is the loss suffered by the claimants, multiplied by the claimant scaling factor. In section 8.5, I calculated this factor to be 170.

985 In that case, following the hypothesis put forward by the claimants, the impact of Horizon bugs on all PO branches over the period 2000-2018 would be expected to be £10M times 170 = £1,800 million.

986 I next consider the consequences this would have had, across the PO network.

987 In section 7.5, I estimated the probability of SPMs reporting anomalies in their accounts, depending on the size of the anomaly. In this section, I shall again use the same conservative form of these estimates as follows:

988 I estimated, as 'middle of the road' best assumptions of SPM behaviour in reporting anomalies in their monthly balancing:

a) If a discrepancy is £1000 or more , the SPM probably needs to investigate it. If he cannot find the cause in his branch, the likelihood of his reporting it through a help line is 80% (for this computation, assume only 40% of SPMs report)

k) If a discrepancy is of the order of £300, 30% of SPMs will report it (for this computation, assume only 15%)

l) If a discrepancy is of the order of £100, 10% of SPMs will report it (for this computation, assume only 5%)

m) For a discrepancy of £10 or less, it is usually not worth the SPM's time to investigate it (because errors in counting cash or stock are often larger than this); so these are reported on less than 1% of occasions.

989 Anomalies which were immediately apparent to the SPM, in customer transactions, would be reported with higher frequency.

990 Suppose that the impact of one occurrence of a Horizon bug on a branch's accounts was approximately £1000. To cause a shortfall of £1,800M over an 18-year period, there would need to be 1,800,000 of these occurrences. Using the conservative estimate above, at least 40% of these, or 720,000, would be reported by the SPM.

991 If, on the other hand , the impact of one occurrence was £300, to cause a shortfall of £600M would require 2 million occurrences - of which at least 15%, or 300,000, would be reported (more, if they occurred in customer transactions). Similarly, if each occurrence was of £100, there would be 6 million occurrences, of which at least 2,400,000 would be reported.

992 So, if the hypothesis put forward by the claimants is correct, regardless of the size of impact of occurrences of bugs, the Horizon help desk would have been subjected to about 720,000 or more calls from SPMs, over 18 years. This is a rate of more than 200 calls per day - all raising urgent issues with large financial impact that needed to be resolved.

FUJ00083961

993 At paragraph 184, at Angel Van Den Bogert's witness statement she says that call volumes into NBSC have been of the order of 1000 calls per day. It seems to me unlikely that such a high proportion as 20% of these should hav been about high-value discrepancies in branch accounts.

994 Those issues would have all have given rise to KELs - or to notes on existing KELs and their Peaks saying: "this issue has arisen yet again". The number of KELs, or recurrences of the same issue noted in a KEL or Peak, would have been of the order of7,000 (rather than the actual 8,000 KELs, with a small number of Peaks per KEL).

995 Also (following the hypothesis put forward by the claimants), most of these issues would need to have been resolved incorrectly - by attributing them to human error in the branch, rather than to a bug in Horizon - in order to lead to a loss in the branch. Investigation of the bug would not have led to any correction in favour of the branch.

996 There are several pieces of evidence that this account, as put forward by the claimants, cannot be correct:

- As above, he Horizon help desk was probably not bombarded by 200 or more urgent calls, reporting large anomalies, each day
- There is no evidence in KELs or Peaks or reports of 300,000 anomalies in branch accounts of £1000 or more, or of larger numbers of smaller anomalies. The number of KELs relating to any anomaly at all in branch accounts appears to be less than about 100 KELs.
- There is no evidence that most anomalies were resolved incorrectly. Rather, because of the many checks and countermeasures in Horizon, the KELs and Peaks record that most anomalies were diagnosed fairly quickly and corrected.
- There is no evidence that the number of anomalies diagnosed as error in the branch, and therefore attributed to the branch, was anything like the figure of 300,000 or more which is implied by the claimant's hypothesis.

997 I conclude that the hypothesis put forward by the claimants, that some large part of their £18.7M shortfalls was caused by bugs in Horizon, is not consistent with the evidence, by a large margin. No conceivable margin of uncertainty in the evidence could make it consistent with the claimants' hypothesis.

8.8.3 Considerations of Central Accounts and Auditing

998 This formulation of the analysis of the impact of bugs in Horizon starts from the same premises as the second analysis. Following that analysis, it assumes that:

- Of the total £18.7 shortfalls experienced by the claimants, as the claimants say, some large part (for definiteness, I take it to be 50%) was caused by bugs in Horizon - giving £10M caused by Horizon bugs.
- Because the impact of Horizon bugs is expected on average to be the same for claimants as for other SPMs, the impact of bugs on all SPMs is expected to be £10M times a claimant scaling factor, which I have calculated in section 8.2 to be 170.

FUJ00083961

999 This implies that over the lifetime of Horizon, a sum of the order of £1.7 billion has leaked out of branch accounts, caused by bugs in Horizon.

1000 PO accounts are held on POLSAP, which adheres to the principles of double entry accounting. This means that any amount of money which leaks out of branch accounts must appear in some other account. Setting aside the possibility that money leaks from one branch to another, this must be some central PO account or accounts, which aggregates the amounts from many branches or all branches.

1001 Therefore, in some central PO account or accounts, a figure of £1,7million over 18 years (or £8 million per month) must be appearing - apparently for no good reason, because it is caused only by bugs in Horizon, which PO do not know about.

1002 In spite of the large amount of money which passes through PO accounts - which is about £100 billion per annum - the great majority of this money is pass-through of agency business, which PO does for its clients. Those clients check the amounts carefully by reconciliation, and it does not seem possible that £8M per month could be hidden in those figures. That amount is greater than the total amount of transaction corrections, which have amounted to something of the order of £2M per month.

1003 Therefore the £8M per month which, on the claimant's hypothesis, arises from Horizon bugs, must be hidden in some smaller central accounting lines, possibly connected with cash and stock in the branches.

1004 It does not seem possible to me that either:

- These smaller accounting lines, (which are where PO makes a large part of its profit) are managed so loosely that an unexplained figure of £8M per month can hide in them
- In the PO's annual financial audit, done by an independent third party, an unexplained figure of £93M could pass without comment.

1005 In my opinion, this combination of the DEA and MID countermeasures would obviously and easily have detected the sums arising from Horizon bugs, if they had been anything like as large as the claimants assert.

1006 I have requested the disclosure of further evidence about this point, but I have done so too late for it to be included in this report.

8.9 Impact of Bugs in Horizon on Individual Claimants

1007 While the previous analysis has calculated the likely size of the summed impact of Horizon bugs on all claimants, I also need to calculate its likely impact on any individual claimant. The previous analysis implies that the impact on all claimants, spread over all their 52,000 months of tenure, was at most of the order of £20,000. In order to be very conservative when considering individual claimants, I shall multiply this figure by a factor 5 - assuming (to give the claimants the maximum benefit of the doubt) that my analysis has somehow omitted 8% of the bugs with financial impact - or underestimated their financial impact by a factor 5. This is an extremely conservative assumption.

1008 The mean financial impact of bugs in Horizon on any one claimant in any one month of his tenure, is then £100,000/52,000 = £2. A mean loss per month of £2 from Horizon bugs can occur though bugs with higher

FUJ00083961

financial impact, but only if they occur with low probability. For instance, a loss of £200 could occur, but only with probability one part in 100. Or a loss of £2000 could occur, with probability one part in 1,000.

1009 The mean loss from bugs of £2 per month is to be compared with the mean loss per month suffered by the claimants, of £360. Typically claimants suffered larger losses in individual months. If, then, a claimant were to say: "In a particular month, I suffered a loss of £2000, and I assert that it was caused by a bug in Horizon", then (as above) the probability of that account being correct is only one part in 1,000.

1010 If the claimant made the same claim about £2000 losses in two different months, the chances of that account being correct are one part in 1,000,000; and so on - the more months are involved, the more improbable the account becomes.

1011 Similarly, a claim that Horizon caused two losses in one month, each of £200, has only a probability $(1/100)^2$, or one part in 10,000 , of being correct.

1012 This is standard probability theory, applied to calculate the balance of probabilities of bugs impacting individual claimants.

1013 This result concurs with a previous analysis, where I showed that , in order for a claimant to credibly assert that a shortfall in his accounts in some month were caused by bugs in Horizon, there would need to be 64,000 such bugs. No such number of bugs has been found. Only a handful have been found.

1014 Because so few bugs have been found, the overwhelming probability is that any set of branch accounts in any month is accurate - with no significant shortfalls caused by bugs in Horizon. There may, however, be temporary inaccuracies in branch accounts caused by delayed TCs, which lead to later corrections.

8.10 Financial Impact of All Bugs, Using Data Provided by the Claimants

8.10.1 Analysis In This Sub-Section

1015 In this sub-section I present two types of opinion:

a) I first explain in qualitative terms why the data submitted by the claimants, as part of their claim, is not consistent with the assertion that their shortfalls arose from bugs in Horizon

n) I then describe the same point in more quantitative terms, and derive an upper limit on the part of the shortfalls that can have arisen from bugs in Horizon.

8.10.2 Qualitative Analysis

1016 In their claim, the claimants have submitted evidence about what shortfalls were experienced by each claimant, and when they occurred.

1017 Claimants held branches for a total of just over 52,000 months. If their shortfalls had been caused largely by bugs in Horizon, then one would expect those bugs to have occurred randomly, and approximately uniformly across those 52,000 months.

1018 I illustrate this by an analogy. The total claim is like a field, divided into 52,000 'plots' of approximately equal area. Bugs in Horizon are like raindrops, falling randomly and uniformly across the field. One would expect approximately the same number of raindrops to fall on each plot, apart from random fluctuations.

FUJ00083961

1019 The picture revealed by the claim is very different from this, and is not at all uniform. There is a large amount of 'clumping' of the shortfalls, in two respects.

- There is clumping in claimants. Some claimants have very small monthly average shortfalls - of £50 or less; while other claimants had very large average monthly shortfalls, of £1000 or more.
- There is clumping in time. At all time periods, many claimants experienced periods of 36 months or more with no shortfalls. Shortfalls arrived in lumps.

1020 In both of these respects , the evidence submitted by the claimants is completely inconsistent with their claim, that bugs were the origin of their shortfalls. It is not uniform rainfall; it is localised deluges. That is consistent with the causation of shortfalls by human error. This is illustrated with data and calculations in the rest of this sub-section.

8.10.3 Quantitative Analysis

1021 In the rest of this sub-section I derive results about the losses suffered by the claimants, and about what proportion of those losses could have arisen from bugs in Horizon, from a different set of evidence from that used in the previous sections.

1022 These results confirm the results derived in the previous sub-sections - that at most only a small proportion of the shortfalls experienced by the claimants could have arisen from bugs in Horizon. The upper limit I derive for this proportion - which is about 8% - is a much weaker limit than the previous upper limit, which is well below 1%. However, the two upper limits are consistent with one another, and this limit acts as a backstop and independent confirmation the other limit.

1023 As a backup to the previous analysis, this analysis has the merit of resting on completely different and independent evidence. This evidence has been provided by the claimants themselves, and so has been available to them for some time. The claimants could have made this analysis themselves at any time in the last three years.

1024 I describe here both a time-independent analysis, which places a limit on the contribution of bugs in Horizon to the total shortfall suffered by the claimants over the whole period up to 2018, and a time-dependent analysis, which places a similar limit on the contribution of bugs in Horizon, during a set of three-year time periods spanning the whole interval.

1025 Both analyses are straightforward at a verbal intuitive level, which is described in this sub-section. However, for each analysis, the detailed numerical result rests on some statistical and mathematical arguments, which are described in an Appendix.

8.10.4 Evidence used for Analysis

1026 There are 561 claimants. Each one has provided a claim summary, which includes:

- Section 8.1: the total amount of shortfalls they have repaid to PO
- Section 3.1: a listing of all the individual shortfalls they experienced, with a date or date range for each shortfall

FUJ00083961

1027 For a few claimants, the amount repaid in section 8.1 exceeds the sum of the individual shortfalls in section 3.1. For those cases, I have assumed that some shortfalls are missing from the section 3.1 data. To correct for these as far as possible, for each of those claimants I have added a single 'balancing shortfall' to the section 3.1 data, whose amount is chosen to make the sum of section 3.1 figures equal to the section 8.1 figure, and whose date range is the whole period of tenure of the claimant. This wide date range makes no assumption about when the missing shortfalls arose. From then on, I have used exclusively the section 3.1 figures to assess the shortfalls and the periods during which they occurred.

1028 This leads to the following overall figures:

- 561 claimants
- Sum of the periods of tenure of all claimants: 52,077 months
- Mean period of tenure per claimant: 92 months
- Sum of all shortfalls: £18.7 million
- Mean shortfall per claimant per month in tenure: £359

1029 A spreadsheet, giving for each claimant the dates of their tenure and the dates and amounts of all the shortfalls they experienced as in section 3.2 of their claims, has been provided to me by the defendant's solicitors, and is the basis of the analyses in this section. It is attached as an Annex to this report.

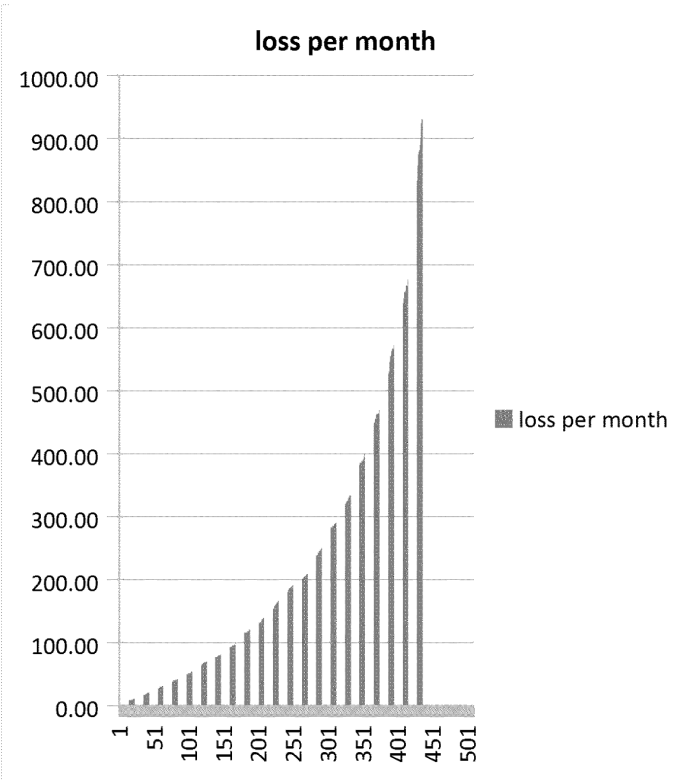
8.10.5 Some Graphical Summaries of Claimant Evidence on Their Shortfalls

1030 There are some interesting results which can be derived and shown graphically by simple processing of the spreadsheet provided to me, using the facilities of Microsoft Excel. I present these results here for their interest, and generally to increase understanding of the claim.

1031 There is a wide range of variation in the average loss per month experienced by each claimant. Some claimants experienced an average loss per month of several thousand pounds, whereas 72 of the 560 claimants each experienced an average monthly loss of £50 or less.

1032 This can be seen in the cumulative histogram below. The horizontal axis is monthly average loss of a claimant in pounds. The vertical axis is the number of claimants (out of the 561) whose average monthly loss is less than that figure. Claimants have been sorted in order of ascending average monthly loss - those with the smallest average monthly loss on the left. The diagram does not include the claimants with the highest monthly losses (who would have been to the right-hand end), because those losses (which were several thousand pounds per month) would have compressed the vertical scale so much that one could not see the monthly losses for the claimants with smallest losses. (Note - this histogram comes from an earlier analysis, when I only had available the section 8.1 shortfalls repaid, rather than the full shortfalls experienced as in the section 3.1. The qualitative result is unaltered by this difference).

FUJ00083961

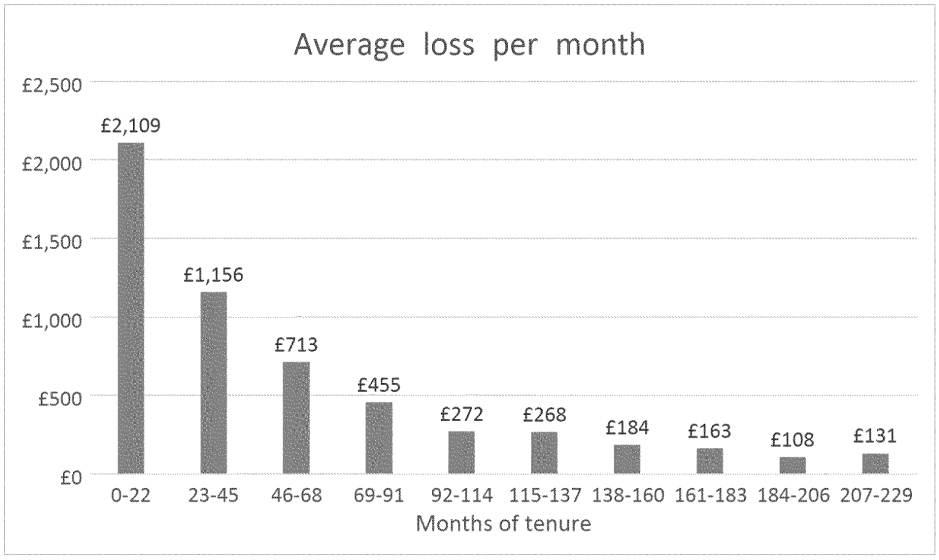


1033 This graph shows that claimants experienced a very wide range of average monthly losses - from a few pounds per month for some claimants, up to £1000 per month for others (and more claimants missed out at the right-hand end of the graph, which if included would continue to shoot upwards, as explained above).

1034 As described above, this graph on its own calls into question the idea that most of the claimant's losses were caused by bugs in Horizon - because one would expect bugs in Horizon to have affected all claimants equally, apart from random fluctuations. This would have led to all claimants suffering approximately equal losses per month - not to a 'low tail' of claimants with very small losses per month, or a 'high tail' of claimants with very high losses per month. Since the graph shows both a low tail and a high tail, it contradicts the hypothesis of random Horizon bugs impacting all claimants. It is, however, consistent with the idea of losses being mainly caused by human error - with a wide range in the rates of human error in different branches.

1035 It appears that the claimants with shortest tenures experienced the highest average monthly rate of loss:

FUJ00083961



1036 To describe two results from this chart:

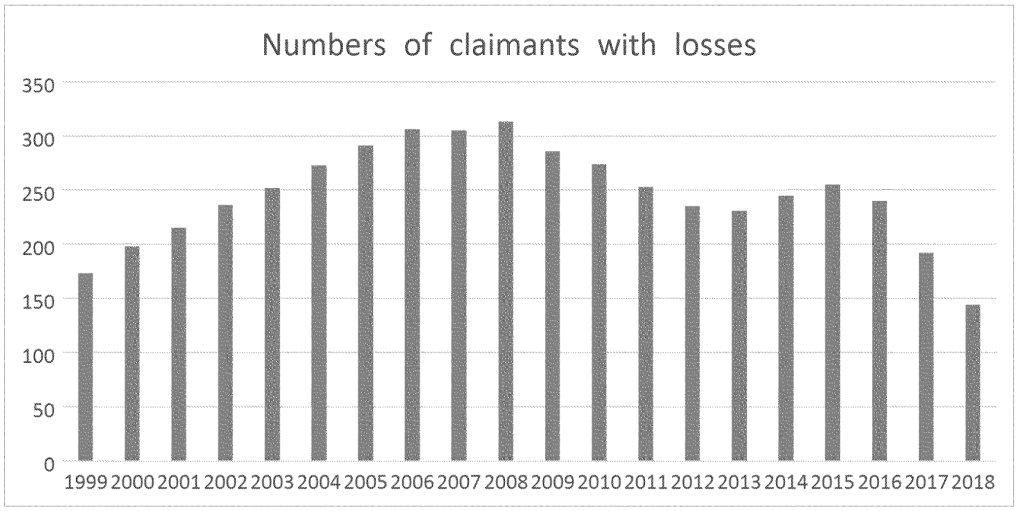
- Claimants with tenure between 0 and 22 months experienced an average shortfall of £2,109 per month
- Claimants with tenure between 138 and 160 months experienced an average shortfall of £184 per month

1037 However, within each range of tenure, there was wide variation in the rate of loss - with many claimants experiencing losses much less than £100 per month.

1038 This chart is equally not consistent with an account that losses arose from bugs in Horizon.

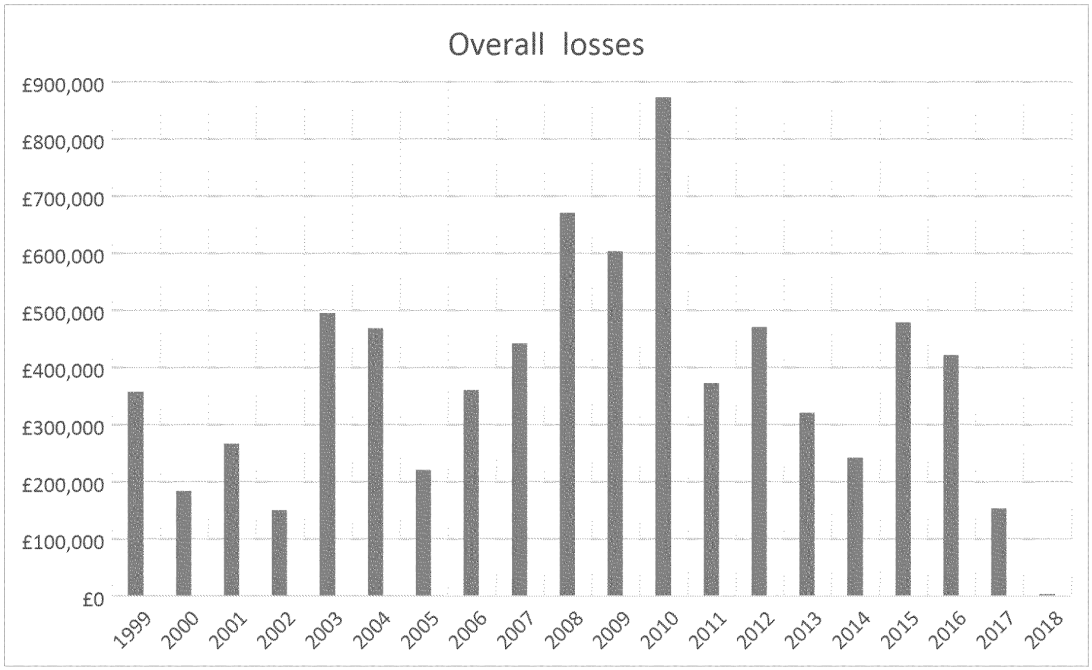
1039 One possible interpretation of the chart above is that claimants with shorter tenures were less experienced, and so were more prone to make human errors which caused losses.

1040 We can also analyse the number of claimants who were experiencing losses in each year:



1041 This shows that with minor variations, claimants have experienced losses over the whole period 1999 - 2018. We can plot the amount of losses they have experienced:

FUJ00083961



1042 I do not yet know the causes of variation in particular years, but it is clear that shortfalls have been experienced from both Horizon and HNG, in all years of their operation. Much of the variation may just arise from random fluctuations.

1043 However, the broadly flat nature of this graph, with random-looking fluctuations for year to year, qualitatively contradicts the notion, as put forward by Mr Coyne, that Horizon sometimes had 'bad periods' in which robustness countermeasures did not work well, and claimants suffered large losses as a consequence. In my opinion, any such 'bad period' would extend over two or three years, while Fujitsu grappled with widespread problems. The graph does not show this pattern.

1044 The graph clearly contradicts the notion that old Horizon (pre-2010) was notably worse than HNG (post-2010).

1045 There is an obvious spike in claimants' reported losses in 2010, which one might interpret as arising from the introduction of HNG, and teething problems in the new system. An alternate account is that all branches were audited as part of the migration to HNG [cite evidence]. Auditing branches would obviously flush out problems in branch accounts which have lain hidden, leading to a spike in reported shortfalls. Since many claimants show a pattern of not reporting losses for extended periods, followed by large 'lumps' of loss, this second account appears more likely. The 2010 audit revealed existing problems in branch accounts.

8.10.6 Analysis from Claimants with Small Monthly Average Losses

1046 For the time-independent analysis, I assume that, for any claimant, the monthly loss is a sum of two terms:

a) An amount arising from human error in the branch

o) An amount arising at from bugs in Horizon

FUJ00083961

1047 I also assume that Horizon bugs occur at random - affecting each claimant, in each month of their tenure, equally, with only random fluctuations. The rate of incidence of Horizon bugs may vary over time (as software is introduced and bugs are fixed), but I assume there is nothing about the behaviour or circumstances of any claimant which makes them more or less likely than any other claimant to suffer in any month from a Horizon bug which affects their accounts.

1048 I am aware that this assumption may be questioned. It might be argued that some aspect of a branch's business - such as its location, or the type of business it transacts, or its number of stock units - makes some claimants more prone to Horizon bugs than others. However, we know which claimants had lower levels of average loss per month, and which claimants had higher levels. Therefore, if it were to be claimed that some factor or criterion (called X) led to a high incidence of Horizon bugs, it would then be necessary to show that claimants with high monthly average losses were subject to factor X, and that claimants with low monthly average losses were not subject to factor X. So any such claim is immediately subject to test from claimant evidence. I do not yet know of any criterion that would pass this test.

1049 In a later sub-section of this section, I have looked for reasons why the assumption of equal average losses from bugs might not hold, but have not found any.

1050 Human errors, unlike bugs in Horizon, are not expected to affect all branches equally on average. Two obvious reasons for this are that different branches may be managed with different levels of skill, and customer transactions may be carried out with different levels of care in different branches.

1051 I therefore assume that bugs in Horizon affect all branches equally, apart from statistical fluctuations. Then, apart from these fluctuations, the amount (b) will be the same per month for all branches. Since the average claimant has a tenure of 92 months, we expect the amount of statistical fluctuation in amount (b) to be small - because for any claimant, assuming Horizon bugs occur at random, good months would balance out with bad months.

1052 It then follows that the claimants with the smallest average monthly loss are likely to be those with the lowest level of human error in their branches - assuming that Horizon bugs affect all branches equally.

1053 I can then use a sample of the claimants with low monthly average losses - who were subject to Horizon bugs, like all other claimants - to estimate the level of monthly loss caused by Horizon bugs; and then scale up that level of loss across all claimants, to see what proportion of their total claimed loss can have been caused by Horizon bugs. I assume that losses from Horizon bugs were never, or very rarely, cancelled out by gains from human error.

1054 In the table below, this scaling calculation has been done for various different numbers of claimants - starting with those who have the least monthly average loss, and working upwards.

Claimants	Claimant Months	Cumulative Loss	Mean Loss per Claimant per Month	Scaled Loss	Percent of total loss
1-1	197	£10		£27	
2	3	574	£5	910	
				1	1

FUJ00083961

1-26	4059	£30296	£7	£38701	2
1-41	6207	£65440	£10	£54904	2
1-55	8195	£11272	£13	£71634	3
1-72	10241	£17269	£16	£87819	4

1055 In this table:

1. The first column 'claimants' shows the number of claimants included, starting at those with the lowest average monthly losses, and working upwards.
2. The second column, 'claimant months' shows the number of months of tenure summed over all these claimants.
3. The third column, 'cumulative loss', shows the sum of all losses claimed by those claimants
4. The fourth column, 'mean loss per claimant per month' expresses the cumulative loss as an average loss per claimant per month. Column 4= Column 3/Column 2. (Column 4 is to be compared with £360 per claimant per month average over all claimants)
5. The fifth column, 'scaled loss' takes the cumulative loss (column 3) and scales it up by a factor (total claimant months/claimant months = 52,000/Column 2), where total claimant months is 52,000 months, to allow for the total number of claimant months for all claimants. This is an upper limit on the total claimant loss from Horizon bugs, assuming Horizon bugs have an equal average effect on any claimant month. It is an upper limit because, whatever set of claimants we take, there are some human errors in addition to Horizon bugs. (more simply, column 5 = 52,000 times column 4)
6. The sixth column 'scaled %' expresses the fifth column as a percent of the total claimed loss, £18.7M

1056 Each row of the table presents a different analysis, using a different subset of the claimants. For each subset (each row of the table), the claimants are ordered in order of ascending average monthly loss, and claimants 1-N are selected (where N = 12, 26, 41, and so on).

1057 I shall describe again the basis of this analysis in each row. By selecting those claimants 1-N with the smallest average monthly losses (column 1), I have selected those claimants who have the lowest level of human errors occurring their branches. Those claimants, having the lowest level of human errors, give the most accurate measure of the other term in their total losses, losses from bugs in Horizon (with the lowest level of noise from human errors). These claimants, therefore, give the best measure of the level of shortfall in their accounts, per month, arising from Horizon bugs. The sum of those claimants losses is in column 3. Their loss per month is in column 4. I have taken this measure of the level of bugs in Horizon (which is actually an upper limit, because all branches have some level of human error), and scaled it up to account for all claimants branches, in column 5.

FUJ00083961

1058 Column 6 has the figures from column 5, expressed as a percentage of £18.7 million. It is the maximum percentage of the total claimants shortfalls which could be accounted for by Horizon bugs, given the data for the claimants in that row (which defines an upper limit on the level of Horizon bugs)

1059 It is possible to take any row of the table as an estimate of this number. I have chosen the bottom row, as the most conservative and reliable estimate. It is most conservative, in that it has the highest level of human error built into it; and it is statistically the most reliable, because it rests on the largest sample of claimants.

1060 This analysis shows that, assuming Horizon bugs occur at random across all claimants, they cannot account for more than 4% of all claimed losses.

1061 However, this analysis needs to be corrected for a statistical effect - that by taking those claimants with the lowest monthly average loss, I have selected those claimants who were luckiest in not suffering from bugs in Horizon, which occur at random. The expected random fluctuation is not a large effect. Because the average tenure of a claimant is 92 months, claimants suffer Horizon bugs over many months, and good months compensate for bad months; so the amount of fluctuation between claimants is small. I have made detailed calculations to assess the extent of the effect, and it is never more than a factor 2. This statistical correction is described in the appendix.

1062 Therefore, allowing for this factor 2, the maximum level of average shortfall per month arising from Horizon bugs is £32 per month, or 8% of the average shortfalls actually suffered by the claimants.

1063 There is another possible account a part of this effect. This is that the claimants who suffered the lowest average monthly rates of loss did so because they were running the smallest branches, and so suffered from the smallest levels of bugs in Horizon. I only received evidence about the size of different claimants' branches with Steve Parker's witness statement on November 18th, so I have not yet had time to account for this factor in my analysis. I shall be able to do so in time for my supplemental report. Until I do so, it adds further level of uncertainty to the result -although I believe it is not a large uncertainty.

1064 This upper limit, of 8% of the claimants shortfalls , is a weaker upper limit than the limits derived in the previous two sections. This is to be expected, because it is based on different evidence, and there is a limited amount of this evidence from the claimants. However, the two results support one another, and each one shows independently that bugs in Horizon cannot account for more than a small part of the claimants' shortfalls.

8.10.7 Time-Dependent Analysis of Claimants' Losses

1065 Following the previous sub-section, I assume that, during any time period, bugs in Horizon affect all claimants equally, with only random fluctuations. Then I can analyse all the shortfalls experienced by any claimant during that period, to see what part of them might have been caused by bugs in Horizon.

1066 For this analysis, using the same spreadsheet of claimants' losses provided to me by the defendant's lawyers, I have used time intervals of three years, starting at the first use of Horizon. This interval has been chosen to be wide enough to get a reasonable amount of data in each interval, and to detect Horizon bugs, even if the expected time interval between their occurrences is large; yet narrow enough to allow for the fact that the level of Horizon bugs might vary over time.

FUJ00083961

1067 Examining data in these three-year intervals, I have found that for each interval:

- A large proportion of the 561 claimants were active
- A fairly large proportion of these claimants had tenure spanning the whole 36-month period
- Of these, more than 25% experienced no shortfalls at all over the 36 months.

1068 These findings are shown in the table below:

Start Month	End Month	Claimants	Full span	With no shortfalls	% with no shortfalls
1	36	244	151	39	26
37	72	315	184	96	52
73	108	342	223	97	43
109	144	331	185	76	41
145	180	276	185	75	41
181	216	227	124	41	33

1069 In this table:

- 'Start Month' is the start of the time period, measured in months from October 1999
- 'End Month' is the end of the time period, measured in months from October 1999
- 'Claimants' is the number of claimants whose period of tenure has any months in the period
- 'Full Span' is the number of claimants whose period of tenure spanned the whole period
- 'With no shortfalls' is the number of these who experienced no shortfalls during the period
- '% with no shortfalls' is the percentage of those claimants whose period of tenure completely spanned the 36-month period (Column 4) and who reported no shortfalls during the period (column 5)

FUJ00083961

1070 The main point to note is that the percentages in the last column are large- typically 40-50%. In every one of these three year periods, a large proportion of the claimants who were in post for the whole of the period, also reported no shortfalls in the same period.

1071 Is this consistent with the claimant's hypothesis, that a large proportion of these shortfalls were caused by bugs in Horizon?

1072 If bugs in Horizon caused a large part of the claimants' shortfalls - which averaged at £360 per month, or more than £12,000 in a three year period) - then one would expect those bugs to have been occurring with high frequency. It would require many bugs to occur to each branch, on average, in any 3-year period (for instance, 12 occurrences in three years, unless the impact of each occurrence was on average more than £1000).

1073 It is intuitively obvious that if some Horizon bug is occurring at random on average 12 times in a three year period, then the chances of it not occurring at all in any particular three- year period - as the claimants' own data show was often the case - are very small indeed. (a claimant may be lucky enough to dodge one bullet; but the chances of dodging 12 bullets are very small indeed) Repeated gaps of three years are not consistent with the claimants' hypothesis, of randomly occurring Horizon bugs causing their shortfalls.

1074 The appendix contains a statistical analysis of these data, showing again that bugs in Horizon occurring at random could not have caused more than about 8% of the claimants' shortfalls, in any three year period.

1075 The most important weakness in this analysis is the factor of claimant behaviour, in reporting their shortfalls. Even if shortfalls caused by Horizon bugs were occurring frequently to claimants, it is evident that they frequently delayed for long periods before reporting them - and this could be the cause of some of the three year gaps in reported shortfalls.

1076 Therefore the quantitative conclusions from this time-dependent analysis cannot be accepted until I have been able to investigate the effects of late reporting of losses more fully. Nevertheless, this analysis is useful because it shows that for any three year period in the lifetime of Horizon, the claimants' assertion - that some large part of the their shortfalls arose from bugs in Horizon - is at odds with their own evidence. They would need to account for long delays in reporting their shortfalls - delays of up to three years - in order to reconcile this evidence with their account of their losses.

8.11 Extent of Bugs - the Number of Different Bugs

1077 In section 8.4 I described what measures of 'extent' I would use in d Horizon issue 1 quantitatively. I described a first sense of 'extent' - financial impact - and a second sense - number of distinct bugs.

1078 I expressed the view that the first sense would be more useful, in that it relates more directly to the claim of financial losses.

1079 I also noted some limitations of the second sense of extent: that it is more difficult to infer from the available evidence (because robustness is directed to control the first extent, not the second); that it suffers from ambiguity of definition; and that it is difficult to scale it between the three different scopes

1080 In spite of these difficulties, I state here my opinions on the second sense of the word 'extent' in Horizon Issue 1.

FUJ00083961

1081 Using the same conservative assumptions as I used for the financial impact of bugs, I shall estimate the number of distinct bugs, with impact on branch account in Horizon.

1082 In the table in section 8.7, where I summarised these calculations, I estimated at the row marked 'L' that there were no more than 200 bugs in the KELs which might affect branch accounts.

1083 To find the total number of such bugs in Horizon, this needs to be corrected for the following factors:

- The probability that no SPM ever reported the bug - estimated conservatively at 0.8
- The probability that Fujitsu never made a KEL - estimated conservatively at 0.5
- The probability that the KEL has not been archived - estimated at 0.85

1084 The result of applying these correction factors is that the upper limit of 200 bugs in the KELs becomes an upper limit of 672 bugs in Horizon. This calculation is also included in the spreadsheet of calculations attached to this report.

1085 This count excludes micro-bugs, whose impact on each occurrence, or in each month for one branch, is so small that they may never be reported. They are discussed in sub-section 8.12.

1086 Apart from micro-bugs, the estimate is very much an upper limit - where I have counted a KEL as potentially indicating a bug, even when the evidence in the KEL is far from conclusive.

1087 In case 672 possible bugs might seem to be a large upper limit, I note again a conclusion I reached in section 8.5. In order to substantiate an assertion, by any claimant, that some shortfall of £1000 in his accounts in some month was caused by bugs in Horizon, there would need to be not just 672 bugs in Horizon - but more than 64000 bugs, each one with impact similar to the Suspense Account bug. So my estimate of a maximum of 672 bugs does little to support the claimants' case.

8.12 Analyses needed in Support of My Opinions

1088 In this section I present two analyses of topics which I need in support of opinions stated earlier in this section - but where the analysis is a bit lengthy and would have interrupted the description.

1089 [The first is a bit technical - could be appendix? Or both?]

8.12.1 Impact of Bugs in Horizon on Claimants and Non-Claimants

1090 In the analyses I have made of the financial impact of bugs, I have assumed that Horizon bugs occur at random, with equal frequency per month (or more precisely, per customer transaction) in the branches of claimants and non-claimants.

1091 This is a reasonable assumption, given the nature of bugs in Horizon - that each bug is triggered by some combination of circumstances - say, a particular action, in a particular type of customer transaction - and that, in the absence of further information, that combination is equally likely to happen in claimants' branches, as in non-claimants' branches.

1092 The opposite assumption - that claimants' branches are somehow different from non-claimants' branches, in a way that triggers more bugs - is not a part of the claimants' case. If it were to become a part, it would be a

FUJ00083961

testable part. In other words, if it were claimed that bugs in Horizon were triggered by some circumstance - which occurred more often in claimants' branches than non-claimants' branches, then that assertion could be tested. It would be possible to compare claimants' branches with non-claimant's branches in this respect, and see if there was any difference.

1093 I have tried to think of possible differences of this sort, and I have only been able to find one candidate difference.

1094 It might be said that claimants tend to make more errors than non-claimants, and that these human errors particularly trigger bugs in Horizon. Or it might be said that certain bugs in Horizon are successfully handled by non-claimants, but tend to cause claimants to make errors, which cause losses.

1095 I have examined both these possibilities, and found that neither of them can account quantitatively for a large excess of bug impact, for claimants over non-claimants. In engineering terms, they are both second-order effects (combinations of two unlikely things), and so cannot be large effects.

1096 The rate at which bugs occur, not prompted by human error, is:

rate of bugs = (rate of normal transactions) * P(bug| normal transaction)

1097 Here P(bug| normal transaction) is a conditional probability. The vertical '|' should be read as 'given'; P is 'the probability of a bug, given a normal transaction'. Here 'given X' means 'assuming X has happened'. We know, from the 6 millions Horizon transactions that happen every day, that P(bug| normal transaction) is very small.

1098 The rate at which human errors happen is:

rate of human errors = (rate of normal transactions)*P(human error|normal transaction)

1099 Here, we may assume that P(human error|normal transaction) is larger for claimants than for non-claimants. However, for both claimants and non-claimants, P(human error|normal transaction) is small - not as small as P(bug| normal transaction), but still quite small.

1100 The rate at which bugs triggered by human errors occur is:

Rate of bugs triggered by errors = rate of human errors *P(bug|human error in transaction)

= rate of normal transactions * P(human error|normal transaction)* P(bug|human error)

1101 (The expression after the '=' sign on the second line means that that quantity on the second line, like the expression after '=' on the first line, is also equal to the Rate of bugs triggered by errors. It is a mathematical shorthand)

1102 To proceed further, we need to know how large is P(bug|human error) compared to P(bug| normal transaction). We express this ratio as

P(bug|human error in transaction) = K* P(bug|normal transaction)

1103 and we ask: how big is K ? Arguably, K is greater than 1 - because transactions with human errors are rarer than normal transactions, and systems are not tested so thoroughly, to handle all those cases.

FUJ00083961

1104 However, for Horizon, a large part of its design was concerned with coping with human errors, and there is evidence that a large part of the testing effort was concerned with error scenarios. Therefore it is reasonable to suppose that K is not very much larger than 1; Because in my view the testing of error conditions in Horizon was well carried out, the most common user error conditions were tested, and K would not be larger than 4.

1105 We can then show, by combining the equations above, that

Rate of bugs triggered by errors

= rate of normal transactions * P(human error| normal transaction)* K * P(bug|normal transaction)

= K* rate of bugs* P(human error| normal transaction)

1106 This shows that the rate of bugs triggered by human errors is doubly rare - it is K times the rate of bugs, times the rate of human errors. This is much less than the rate of bugs itself, even assuming that K = 4. It is a second-order effect.

1107 To show why such a second-order effect is small, consider this example , in which the numbers are illustrative. If you wish to illustrate the point using different assumptions, you can use those assumptions in the same arithmetic.

1108 Suppose that the probability of a software error in any transaction is 1 in 1,000,000 - that is

P(bug|normal transaction) = 1/1,000,000

1109 Suppose further that the probability of a bug in a transaction with a human error is somewhat larger - say 4 times larger (K = 4):

P(bug|human error in transaction) = 4/1,000,000

1110 Suppose that the probability of human errors, by most SPMs, is 1 in 1,000:

P(human error|normal transaction) = 1/1,000

1111 Suppose also that claimants are more likely to make human errors than other SPMs - for illustration, four times more likely:

P(human error by claimant| normal transaction) = 4/1,000

1112 It then follows that the rate of bugs triggered by claimant errors is given by

Rate of bugs triggered by claimant errors = rate of bugs *4*4/1000

1113 That is, the number of bugs experienced by claimants, as a result of their errors, is just 1.6% of the rate of bugs they experience from other causes. Note that this figure is independent of the first figure P (bug|normal transaction) = 1/1,000,000, because that figure cancels itself out in the result.

1114 Even if claimants make more errors than other SPMs, their rate of errors is still low - so the bugs triggered by those errors are also rare, compared to bugs which occur without errors.

1115 This shows that even if claimants make more errors, it does not significantly increase the rate of bugs they experience.

FUJ00083961

1116 A similar analysis - which I will not present in detail here - shows that even if bugs in Horizon create situations in which claimants make more errors than other SPMs, that does not significantly increase the rate at which claimants experience shortfalls from those errors, compared to the claimants' losses from their other errors which were not triggered by Horizon bugs.

1117 I have therefore found no ways in which claimants differ from other SPMs, which would make them significantly more prone to bugs in Horizon.

8.12.2 Micro-Bugs

1118 The analyses of financial impact of possible bugs in Horizon, which I have given in previous sub-sections, address bugs with all sizes of financial impact, except for a category which I have called 'micro bugs'. These are bugs in Horizon which:

- Have the potential to introduce errors in branch accounts
- Are not immediately visible to the clerk in a customer transaction (if they were, some of them would be reported, however small their financial impact)
- When they introduce discrepancies in the process of monthly balancing, introduce only discrepancies less than £10 per occurrence, which the SPM may put down to human error, or otherwise ignore.

1119 This is my definition of a micro-bug.

1120 Because of these properties, micro-bugs are less likely to lead to KELs or to investigation by Fujitsu. The question therefore arises: could they have significant effect on branch accounts? In this sub-section, I address that question.

1121 The first thing to say is that micro bugs are intrinsically a small class of bugs - in that they do not reveal themselves to the SPM in customer transactions, and that their financial impact is reliably less than £10 in a month (otherwise they would sometimes be reported). For most bugs, the financial impact depends on the transaction they affect - and since transactions have variable size, the impact of a bug is also variable, and so will sometimes be more than £10 in a single occurrence.

1122 Micro-bugs cannot account for a large part of the claimants' shortfalls. Because these shortfalls have an average value of £360 per month, and micro bugs have impact less than £10 per occurrence, in order to provide a large part of the claimed shortfall, micro-bugs would have to occur to one branch so many times in a month that their net effect was much more than £10, and so would be noticeable to SPMs and sometimes reported.

1123 If micro-bugs were to occur with such high frequency as to account for some large part of the claimants' shortfalls (say , ten times per branch per month, to account for an average shortfall of £100 per branch per month), they would have to have occurred more than 24 million times in the lifetime of Horizon (10 time 2.4 million branch months, across the branch network over 18 years). In my opinion, it is extremely unlikely that any bug with such a high frequency of occurrence would have got through testing undetected, or would not have been noticed in live use.

FUJ00083961

1124 Because POLSAP uses double entry accounting, any effect of micro-bugs on branch accounts must also show, aggregated over all branches, in some PO central account. If that effect were significant, then in my opinion it would inevitably have been noticed by some PO manager or external auditor.

1125 The claimants have not provided any evidence that micro bug occur, or that they are a significant contributor to their shortfalls.

1126 For these reasons, in my opinion, micro bugs cannot account for any significant part of the claimants' shortfalls.

8.13 Mr Coyne's Opinions on Horizon Issue 1

1127 Mr Coyne addresses Horizon issue 1 in his summary of opinions, in paragraphs 3.1 - 3.3 of his report, and in section 5, up to paragraph 5.81.

1128 Paragraph 3.1 quotes from the expert joint statement. Paragraph 3.2 makes the uncontroversial statement that bugs in Horizon could have existed for variable periods of time. Paragraph 3.3 talks about the 'sheer volume' of KELs - implying it was large, but not supporting this with the results of any analysis.

1129 These summary paragraphs all address bugs in Horizon in general, and do not focus down on bugs which cause discrepancies in branch accounts, as Horizon Issue 1 requires. It is agreed between the parties that bugs in Horizon exist, as Mr Coyne's paras 3.1- 3.3 state.

1130 These paragraphs do not address the extent of such bugs, as required by Horizon issue 1, except in the phrase 'sheer volume' of KELs.

1131 It seems to me that Mr Coyne's opinions on Horizon Issues 1 cannot yet be contrasted with my opinions in two respects - because in those respects, he has not yet expressed an opinion:

a) He has not expressed an opinion on the many robustness countermeasures built into Horizon, and how effectively or otherwise they have acted to prevent discrepancies or shortfalls in claimants' branch accounts

p) He has not expressed an opinion on the quantitative financial impact of bugs in Horizon on claimants' accounts.

1132 Regarding point (a), in section 7, I have suggested that Mr Coyne and I might hold some without prejudice meetings to discuss robustness countermeasures, and possibly agree a list of them.

1133 Regarding point (b), the arithmetic I have used in the calculation of the maximum possible financial impact of bugs is straightforward. It is contained in a small Excel spreadsheet attached to my report.

1134 Mr Coyne may wsh to re-calculate this result, based on his own sampling of KELs, and other evidence. Although I have every confidence in my result, it is based on the limited sampling and analysis of KELs which I have been able to make so far. It is an interim result. With time and further effort, it can be improved, and the remaining uncertainties in it can be reduced.

1135 I suggest that the refinement of the numerical upper limit of the impact of bugs on claimants branch accounts can be a joint effort by the experts. I invite Mr Coyne to a series of without prejudice meetings - in which we can discuss our sampling of the KELs, the effects of countermeasures, any other relevant evidence, and the

FUJ00083961

arithmetic basis of the upper limit I have derived. In this way we can both present a better refined numerical upper limit on the impact of bugs in our supplemental reports, noting our areas of agreement and disagreement - down to the level of individual KELs. We could continue the meetings after our supplemental reports, to reach a final position which we document in the experts' final joint memorandum.

1136 The end result of this would be that each expert derives and justifies his own upper limit on the possible financial impact of bugs in Horizon on the claimants' branch accounts. We may be able to agree on some of the assumptions and elements of the calculation. Whichever is the larger of the two experts' upper limits, the experts can agree that this figure is a true upper limit on the impact of bugs on the claimants' accounts.

1137 I next comment further on Mr Coyne's detailed opinions on Horizon Issue 1, expressed in his paras 5.1 to 5.81.

1138 Paragraph 5.1 sets out some areas where bugs may occur - without addressing whether or not those bugs can cause discrepancies in branch accounts. Similarly paragraph 5.3 discusses common failure points in Horizon. It does not address what kinds of failures they might be, or whether they might affect branch accounts, in the light of the many robustness countermeasures in Horizon.

1139 The linkage between bugs and branch accounts is addressed in paragraph 5.2 (which recites a point agreed in the expert joint statement) and paragraphs 5.4 to 5.14 (which address the three known bugs admitted by PO).

1140 Paragraphs 5.15 to 5.30 discuss a number of issues, without , in my opinion, providing the depth of analysis to enable the court to determine whether or not these issues affected branch accounts - or if so, the extent to which they did so.

1141 For instance, paras 5.20 - 5.26 discuss various issues relating to cash management and pouch delivery. They do not acknowledge that problems managing cash are quite distinct from inaccuracies in branch accounts, and generally do not lead to inaccuracies. This is because of the countermeasure UEC. If a discrepancy arises during a TP between cash as recorded on Horizon and physical cash (for instance, by mis-recording the amount of cash remmed in or out), then at the end of the TP, physical cash is counted, and any error in Horizon cash is corrected. This means that the issues cited by Mr Coyne probably lead to no error in branch accounts - and he does not provide any analysis to show that they might do so.

1142 Similarly, Mr Coyne's examples of reference data errors (paras 5.30 - 5.34) show that errors occurred - not that they led to inaccuracies in branch accounts. There were many countermeasures such as TIN and UEC to ensure that they would not. Mr Coyne does not discuss these - so his examples do not illustrate bugs which affected branch accounts.

1143 The same lack of any analysis showing any true impact on branch accounts is shown Mr Coyne's other examples up to para 5.81.

1144 In the time available since receiving Mr Coyne's report, it has not been possible for me to analyse all the examples he cites to the depth I should like, to assess whether or not they have any potential to affect branch accounts. This would involve me providing a depth of analysis that he does not provide in each case. I have made a preliminary analysis of the KELs he cites; out of 62 KELs I have examined, there appears to be potential to affect branch accounts in only 8 cases. This is only potential, and does not establish that branch

FUJ00083961

accounts were affected. As described in Mr Parker's witness statement, Fujitsu have analysed the same KELs, reaching conclusions similar to my own. Fujitsu find fewer KELs which might affect branch accounts - apart from temporary impacts

1145 I have not had time to make any similar analysis of the other documents Mr.Coyne cites, to assess their context and significance. I will do so in my supplemental report, where I shall also describe any further analysis I have made of the KELs he cites.

1146 My finding of 8 KELs with possible impact on accounts, amongst 62 cited by Mr Coyne, (or Fujitsu's evidence that there were fewer than 8 such KELs) does not alter my opinions about the aggregate financial impact of bugs, as described earlier in this section. This is because I do not know the sampling criteria used by Mr Coyne in selecting those KELs - so I cannot adjust the KELs found by Fujitsu or myself for those criteria.

1147 However, I note that Mr Coyne's survey of 5114 KELs, as described in his paragraph 5.114, reinforces my conclusion that the financial impact of bugs was very small. This is because if any of those 5114 KELs had stated in obvious terms that there was a bug with impact on branch accounts, I assume that Mr. Coyne would have quoted it in his report. There are no such direct quotations.

1148 Thus Mr Coyne's opinions of Horizon issue 1 lack any focus on the impact of bugs on branch accounts. They add little to the expert joint statement and the three bugs acknowledged by PO. Mr Coyne has said nothing quantitative about the extent of such bugs (as asked in Issue 1), which might be compared with the claimant's shortfalls.

8.14 Analysis of KELs selected by the Claimants

1149 [This whole sub-section, and the table in the appendix which it refers to, are probably redundant in the light of the FJ analysis of Coyne's KELs. Views?]

1150 The claimants indicated in their outline that they are investigating issues associated with eight KELs.

1151 Because I learnt of these KELs in August 2018, I have had time to analyse them in some detail.

1152 My analysis of the first 8 KELs is in Table C of Appendix E. I state whether or not, in the light of that analysis, each KEL has any possible effect on branch accounts.

1153 As is evident from the table, in my opinion none of these eight KELs was caused by a bug in Horizon which could have had any significant or long-term impact on branch accounts.

1154 The reasons for this are diverse and cannot be summarised across the 8 KELs; but they do illustrate the diversity and the effectiveness of the robustness countermeasures built into Horizon.

1155 Because of the limited amount of information available in KELs and Peaks, and the complexity of Horizon, it remains possible that deeper analysis of these KELs would reveal some possibility of effects on branch accounts (even a remote one) in spite of the analysis above. However, even if this were the case for one or more of these KELs, there are good reasons to believe that the financial impact on branch accounts would be very small:

- Nearly all of the problems depended on rare circumstances, or could have only happened in a short time window before the error was fixed, or both.

FUJ00083961

- If the impact of any occurrence of the error was more than a small figure (say £200) it would be noticed by SPMs, reported, and investigated in depth.
- If the impact of a single occurrence were much larger (say as large as £10,000) it would be noticed immediately and not be allowed to repeat more than a very few times. So, as for the three known errors which did have impact on branch accounts, the net impact on claimants' branches would be very small compared to their £18.5 million total shortfalls - a small fraction of one percent.

1156 I conclude that the 8 KELs identified by the claimants probably had no impact on branch accounts; or even if they did, that impact could only be a fraction of a percent of the claimants' total shortfalls.

FUJ00083961

9. EXPERT ISSUES – RECONCILIATION AND TRANSACTION CORRECTIONS

9.1 The Issues

1157 This section addresses Horizon Issues 5 and 15, which concern reconciliation and transaction corrections.

1158 **Issue 5**: How, if at all, does the Horizon system itself compare transaction data recorded by Horizon against transaction data from sources outside of Horizon?

1159 **Issue 15**: How did Horizon process and/or record Transaction Corrections?

9.2 Interpretation of the Issues

1160 Issues 5 and 15 are, on the face of it, mainly factual issues, which can be addressed by factual evidence.

1161 However, the thrust of Mr Coyne's opinions on these issues - for instance in his summary paragraphs 3.13 and 3.28 - is to emphasise that reconciliation, and the creation of transaction corrections, are error-prone processes.

1162 The significance of this for the claimants' case is that any such errors might have introduced shortfalls in the claimant's branch accounts.

1163 My first task in this section is to assess that possibility - and in particular to estimate the possible extent of those shortfalls, as I did in section 8 for bugs and defects in Horizon. I do this in the next sub-section, before addressing the other factual issues. This will set those factual issues in their proper context for the claimant's case.

9.3 Financial Impact of Errors in TCs on Claimants' Branch Accounts

1164 The defendants have disclosed the following information about TCs:

	CREDIT		DEBIT			
Year	Volume of TCs Issued	Value of TCs Issued	Volume of TCs Issued	Value of TCs Issued	Total Volume	Total Value
2005	1151	-£ 316,059.35	11191	£ 8,412,703.76	12342	£ 8,096,644.41
2006	20799	-£ 5,348,456.00	87692	£ 25,215,930.31	108491	£ 19,867,474.31
2007	31288	-£ 9,190,474.09	100774	£ 32,031,684.88	132062	£ 22,841,210.79
2008	41967	-£ 8,417,508.40	98542	£ 20,971,413.52	140509	£ 12,553,905.12
2009	42999	-£ 7,939,353.32	98376	£ 19,993,591.51	141375	£ 12,054,238.19
2010	46460	-£ 8,118,634.08	103984	£ 19,454,770.24	150444	£ 11,336,136.16
2011	54006	-£ 14,580,500.19	79252	£ 19,086,336.06	133258	£ 4,505,835.87
2012	51246	-£ 11,064,648.41	73128	£ 10,089,399.59	124374	-£ 975,248.82
2013	46544	-£ 10,422,881.17	59332	£ 8,964,914.99	105876	-£ 1,457,966.18
2014	62731	-£ 11,431,411.43	51309	£ 18,989,665.02	114040	£ 7,558,253.59
2015	58814	-£ 53,667,783.90	50338	£ 11,435,707.19	109152	-£42,232,076.71
2016	54837	-£ 9,943,787.13	55114	£ 18,349,729.99	109951	£ 8,405,942.86
2017	48922	-£ 8,353,469.31	68960	£ 15,708,356.78	117882	£ 7,354,887.47
2018	9762	-£ 2,240,040.20	20834	£ 4,102,186.97	30596	£ 1,862,146.77
Grand Total	571526	-£ 161,035,006.98	958826	£ 232,806,390.81	1530352	£ 71,771,383.83

1165 This table shows that there are TCs which credit the branches (left-hand columns) and which debit the branches (right-hand columns) - with a fairly high level of cancellation between the two - £161M credit and £232M debit.

FUJ00083961

1166 Summing the magnitudes of these two gives £400M of TCs flowing through branches, over a 14 year period during which the average number of branches was about 13,600. So the mean amount of TCs (either credit or debit) was about £240 per branch per month - which was also about one TC per branch per month.

1167 How many of these TCs might have been in error? Paul Smith's witness statement describes some approximate numbers of disputed TCs. Since one may assume that any erroneous TC is likely to be disputed (along with many TCs that are correct), the level of disputed TCs is in my opinion an upper limit on the level of erroneous TCs.

1168 Where there is evidence on the proportion of disputed TCs upheld, that may give further information on the level of TCs which were erroneous in the first place. When a disputed TC is upheld, I may infer that the TC was in error; whereas, is it is not upheld, that may indicate that after further investigation, PO concluded that it was not issued in error. Although that does not indicate with certainty that the TC was correct, because there was further investigation, I infer that in most cases the TC was correct.

1169 The levels of disputed TCs and upheld disputed TCs in Mr Smith's witness statement are as follows:

Type of TC	Paragraph of WS	Approximate percentage disputed	Approximate percentage disputed and upheld, if known
Cash, Bureau and Personal Banking	17, 18	2%	0.2%
Personal Banking	19	small	small
Camelot, Debit Card & ATM	20	10%	2%
Santander	23	15%	10%
DVLA	24	small	rare
Drop & Go	25	Small	rare
Postal Orders	26	rare	rare
MoneyGram	27	Rare	rare
cheques	28	1%	rre

1170 For Camelot, Debit Card and ATM, I assume that the number of compensating TCs equates to the number of disputes upheld - that if a dispute was upheld, a compensating TC was issued..

1171 From the table above, the only category of TCs with an error rate as high as 10% is Santander. As will be evident from the table below, Santander do not account for a large proportion of TCs. Therefore the Santander row has little influence on the overall average, which I take to be 2% (as Camelot accounts for a large proportion of TCs - see below)

1172 In order to understand the level of errors in TCs a little better, it is useful to know something about the numbers of TCs of different types. The table below is a TC summary from a month in 2013:

FUJ00083961

Products (BRANCH ERRORS)	VOLUME OF TCs			NET VALUE OF TCs (K)			VOL TOTALS (%)	
	2011/12 Outturn	Current Period	YTD	2011/12 Outturn	Current Period	YTD	Current Period	YTD
AON	2			(0 K)				
ATM	1,223	84	601	1,052 K	85 K	573 K	0.65%	0.81%
Automated Payments	2,129	158	1,165	(1,658 K)	97 K	(1,548 K)	1.22%	1.56%
Bureau	3,811	407	2,245	456 K	59 K	270 K	3.13%	3.01%
Camelot	39,039	2,072	10,883	522 K	(44 K)	(221 K)	15.94%	14.58%
Cash Rems From Branch	21,660	2,765	15,466	10,685 K	363 K	4,108 K	21.27%	20.73%
Cheques To IPSL	6,431	545	2,752	6,841 K	341 K	1,358 K	4.19%	3.69%
Debitcards	148	21	92	608 K	187 K	516 K	0.16%	0.12%
DVLA	3,338	466	2,146	(169 K)	(36 K)	(128 K)	3.59%	2.88%
First Rate	185	18	84	50 K	14 K	34 K	0.14%	0.11%
Government Services	553	43	904	(16 K)	(2 K)	(117 K)	0.33%	1.21%
NS&I	1,374	61	730	(612 K)	(103 K)	(343 K)	0.47%	0.98%
Online Banking	752	60	438	(2,160 K)	(171 K)	(948 K)	0.46%	0.59%
Other	309	78	125	6 K	5 K	9 K	0.60%	0.17%
Paystation	4,342	5	8	(1,626 K)	1 K	(0 K)	0.04%	0.01%
Personal Banking	549	41	199	(244 K)	25 K	2 K	0.32%	0.27%
Postal Orders	1,535	59	578	14 K	0 K	4 K	0.45%	0.77%
Pre-order	77	30	68	(8 K)	1 K	0 K	0.23%	0.09%
Santander - Co-Op Business Encashments	296	92	175	66 K	30 K	43 K	0.71%	0.23%
Santander - Green Giro	1,864	74	623	(8 K)	(2 K)	(6 K)	0.57%	0.83%
Santander - Manual Deposit	8,771	928	5,112	(1,282 K)	(82 K)	(474 K)	7.14%	6.85%
Santander - Manual Withdrawal	168	45	122	23 K	10 K	23 K	0.35%	0.16%
Santander - Online Banking	1,689	53	824	(13,625 K)	(481 K)	(6,470 K)	0.41%	1.10%
Saving Stamps	1,058			(4 K)				
Stock - Non Rem	137	37	217	10 K	0 K	2 K	0.28%	0.29%
Santander -manual deposit								

1173 [the bottom of this table is missing- needs to be fixed]

1174 The main point to note from this table is that the two biggest categories of TC by volume are Camelot and Cash Rems from Branch. Between them, they account for more than 50% of the volume of TCs, and no other category accounts for more than about 8%. (Mr Coyne confirms this in his para 6.67 and his appendix C)

1175 For both Camelot and cash remming, there are well understood sources of error in the branches. These are described in another Excel worksheet attached to the table above:

Camelot	Correct accounting procedures followed but incorrect figures entered from Lotto summary to Horizon. (cheque prize payment included for example). Correct figures entered on Horizon but transaction details not accounted for on same or next day.
Cash Rems from Branch	Pouch remmed in at Cash Centre, contents differ to amount stated on advice note. Resulting discrepancy should be held in Rem Suspense and redeemed when Transaction Correction accepted. Pouch despatched but not remmed out or remmed out twice.

1176 The same document has a league table of the branches with the highest rate of TCs. For the leading branches, the main source of TCs was remming of cash.

1177 If there were 2% of TCs issued in error, which were resolved incorrectly against the branch, the net effect on branch accounts would be £5 per branch per month.

1178 However, because, as described in section 8.5, claimants' branches were on average three times smaller than typical PO branches (as measured by number of customer transactions per day), one would expect the number of TCs issued to claimant's branches, and the number issued in error, to be three times smaller than the average for all branches - and therefore to be approximately £2 per month.

FUJ00083961

1179 This figure is to be compared with the mean shortfall per month experienced by the claimants - which , as I described in section 8, was £360 per branch per month. £2 per month from erroneous TCs is less than 1% of this amount.

1180 Therefore, errors in reconciliation and TCs cannot account for a significant part of the claimants' shortfalls.

1181 At the level of an individual claimant - if, for example, a claimant were to say that he lost £200 in one month, due to errors in processing TCs - then in the absence of further evidence, the probability of that claim being correct is about 2/200 = 1%. As before, any claim of several erroneous TCs, on one month or in several months, would have a much smaller probability of being correct - because errors in TCs are statistically independent events, so their small probabilities multiply.

1182 I note that even this small level of shortfalls in claimant's branches from erroneous TCs is likely to have arisen from human errors in the back office, rather than from bugs in Horizon. I am not certain how that relates to the scope of the Horizon trial.

9.4 Reconciliation, Transaction Corrections and Transaction Adjustments

1183 The processes for reconciliation, transaction corrections and transaction adjustments were described in section 6.4 of this report. There is little I need add here to that description.

1184 I agree with Mr Coyne's description of Horizon architecture up to his para 6.10.

1185 In his paragraphs 6.50 - 6.59, Mr Coyne describes how TCs proceed from POLSAP to TRS to the branch, citing POL-0032855, and the options available to the SPM for handling them. These descriptions are consistent with my own knowledge.

1186 [probably more needed here]

9.5 My Opinions on Horizon Issues 5 and 15

1187 Issue 5 asks how the Horizon system 'itself' compares transaction data against transaction data from outside Horizon.

1188 The word 'itself' is problematic - because while reconciliation between transaction recorded on Horizon and the same transactions recorded by PO's clients was extensive and automated, it was done in a variety of different ways and by different IT systems, some of them outside Horizon itself. The scope of Horizon 'itself' has not been defined precisely, and I do not attempt to do so.

1189 Regardless of this, for essentially all of the clients for whom PO acted as agents, there was an automated process of comparison of the transactions as recorded by Horizon and as recorded by the client.

1190 This process compared millions of transactions per day, and was part of an important robustness countermeasure, relying on redundant data storage (RDS), with automated comparisons of the Horizon version against the client version. If discrepancies were detected, this allowed errors from a variety of sources, notably human errors in carrying out transactions or recovering recoverable transactions, to be corrected (UEC).

1191 Issue 15 asks about transaction corrections.

FUJ00083961

1192 If a discrepancy was detected in reconciliation, a correction would need to be made, consistent with the principles of double entry accounting (DEA) - in particular, keeping POLSAP and BRDB in step (another example of RDS). PO's way to make this correction involved manual inspection of data (MID) by PO central staff. If this review found that the cause was an error in the branch, PO would issue a transaction correction (TC), which was followed by review and acceptance or contesting of the TC by the SPM (again MID). This whole process was a constrained double entry process, and kept POLSAP and BRDB in step.

1193 So the joint process of reconciliation and TCs is a very important part of Horizon's robustness, and in my opinion has worked well for over 13 years.

1194 The claimants imply that errors in the MID component of reconciliation and TCs may have led to shortfalls in their branches. I have estimated the likely scale of such shortfalls, which is a second-order effect (an initial error, followed by another error in its correction). As a second-order effect, the effect is expected to be small. Based on the volume of TCs, and the proportion of contested TCs, I estimated this amount to be, on average, not more than £25 per branch per month - compared with the claimants' losses which averaged £360 per branch per month. So errors in reconciliation and TCs did not contribute significantly to the claimants' losses.

9.6 Mr Coyne's opinions

1195 Mr Coyne addresses reconciliation and transaction corrections in section 6 of his report.

1196 When describing reconciliation at para 6.13 onwards, he places an emphasis on manual processes, which in my opinion may be misleading.

1197 Because the PO handles millions of agency transactions per day on behalf of its clients, it is necessary that the process of reconciliation - detecting any discrepancies between the PO version of those transactions, and the client's version of the same transactions - has to be automated. Manual comparison would be impossible. It is only when transactions with discrepancies are revealed by this automated process, that any manual processes are used - to find the cause of the discrepancy, and to ensure it is allocated to the correct account. This does not emerge clearly from Mr Coyne's paras 6.13 - 6.21.

1198 In paragraph 6.38 Mr Coyne says that '*Post Office reported in response to my Request for Information that 10,000+ transactions per week suffer from problems and are not automatically reconciled. Such transactions require manual intervention for the reconciliation to take place*' . The PO RFI response actually said: '*Fujitsu currently "F99" 10,000+ transactions per week across all NB102 associated reports (DCP and NBS)*'.

1199 The figure of 10,000 transactions per week implies 500,000 transactions per year, which is four times higher than the total rate of transaction corrections (which, as in the table above, is usually about 120,000 per annum).

1200 This shows that the F99 event, referred to by PO, usually did not lead to a TC, and so led to no effect on branch accounts. This is confirmed by document POL-0032990, which defines F99 as: '*A transaction state that indicates that a reconciliation error has been reported but POL has advised that the issue has subsequently been resolved. This state is set using the DRS Workstation application that is used by Fujitsu Security Operations team.*'

FUJ00083961

1201 This is further supported in the document CS/SPE/011: '*3.3.1 NB102 Rule 6: 'Where an exception is set to F99 by Fujitsu Services, clearance of this exception within the appropriate NB102 section … will always refer to the previously reported state regardless of any change of state which may have occurred within the DRS.*'

1202 Therefore the 10,000 events per week referred to by Mr Coyne are events that had already been dealt with successfully by PO, in most or in all cases without a TC - and so all or most of them had no effect on branch accounts. It is misleading to portray them all as error-prone human interventions which might influence branch accounts.

1203 The general thrust of Mr Coyne's opinions on issues 5 and 15 is to emphasise that reconciliation and TCs were possibly an error-prone process - and thus to imply that these errors might have contributed to the claimants' losses. This emphasis on possible errors in reconciliation and TCs, appears, for instance, in paras 6.45 and 6.77 of his report.

1204 Mr Coyne has not attempted to quantify the number of these errors in TCs, or their impact on branch accounts. I have done so in section 9.3 of this report. I found that the possible financial impact of errors in TCs is probably less than 10% of the shortfalls experienced by the claimants.

1205 The individual incidents of possibly incorrect TCs described by Mr Coyne at his paras 6.64 - 6.69 do not alter this opinion. Clearly with several thousand TCs in any month, it is possible to cite small numbers of them that were in error; but as described above, their financial impact is small compared to the claimants' shortfalls.

1206 In para 6.3 Mr. Coyne quotes a document from PO about reconciliation which states: "…*not all system faults will lead to corrective action and this is generally done on a contractual and/or cost benefit basis*". The previous paragraph makes it clear that 'system faults' include events such as '*a telephone line being dug up*'.

1207 There are six references in Mr Coyne's report to PO making decisions 'on a cost benefit basis' - three of them in the context of reconciliation. These references might be taken to imply that a cost benefit basis is a selfish or short-sighted commercial thing to do, rather than (for instance) putting the interests of SPMs first.

1208 If that is the intended implication, then in my opinion it is not necessarily correct, as this example illustrates.

1209 Thousands of business decisions are taken every day on a cost benefit basis, in businesses of all sizes - ranging from the PO central functions to individual PO branches. If reconciliation reveals some small discrepancy in some transaction - say a few pounds and pence - then there is a valid business question of whether to spend the administrative effort required to fully investigate it and take corrective action, or more simply to absorb any loss centrally. Likely administrative costs may well dominate in many cases, so it is important for the PO to have guidelines - on a cost-benefit basis - as to what discrepancies should be handled in what way.

1210 In just the same way, each SPM will take decisions - on a cost-benefit basis - designed to make best use of his own time. For instance, in monthly balancing, the SPM must decide which discrepancies to investigate, and which to accept without investigation.

1211 The claimants may wish to imply that in some cases, the line of least resistance for some central reconciliation function would be to 'blame it on the branch'. In my opinion, it is not that simple. If PO centrally were to blame a TC on the branch, in cases where it was not in all likelihood the responsibility of the branch, this would lead

FUJ00083961

inevitably to branches disputing more TCs, and I would expect the administrative costs of investigating any disputed TC to often exceed the amount involved.

1212 So purely on a cost-benefit basis, it may be in PO's interest to keep their SPMs well supported, and not to blame them unnecessarily for discrepancies - in order to minimise PO central support costs, but also to motivate SPMs, not to distract them with unnecessary disputes and investigations, to enable the SPMs to run successful businesses for the PO, to satisfy PO's customers better, and so on.

1213 So, this area involves complex business trade-offs, which the experts have not been asked to investigate. In my opinion, it is not appropriate to portray, or to imply, any over-simplified 'cost benefit' motivation for PO to treat its SPMs badly.

FUJ00083961

10. EXPERT ISSUES – FACILITIES AVAILABLE TO SUBPOSTMASTERS

10.1 The Issues

1214 In this section I address Horizon Issues 2, 9, 14, which have been grouped together because they all concern facilities and information available to Subpostmasters.

1215 **Issue 2**: Did the Horizon IT system itself alert Subpostmasters of such bugs, errors or defects as described in [Issue] (1) above and if so how?

1216 **Issue 9**: At all material times, what transaction data and reporting functions (if any) were available through Horizon to Subpostmasters for:

a. identifying apparent or alleged discrepancies and shortfalls and/or the causes of the same; and

b. accessing and identifying transactions recorded on Horizon?

1217 **Issue 14**: How (if at all) does the Horizon system and its functionality:

a. enable Subpostmasters to compare the stock and cash in a branch against the stock and cash indicated on Horizon?

b. enable or require Subpostmasters to decide how to deal with, dispute, accept or make good an alleged discrepancy by (i) providing his or her own personal funds or (ii) settling centrally?

c. record and reflect the consequence of raising a dispute on an alleged discrepancy, on Horizon Branch account data and, in particular:

i. does raising a dispute with the Helpline cause a block to be placed on the value of an alleged shortfall; and

ii. is that recorded on the Horizon system as a debt due to Post Office?

d. enable Subpostmasters to produce (i) Cash Account before 2005 and (ii) Branch Trading Statement after 2005?

e. enable or require Subpostmasters to continue to trade if they did not complete a Branch Trading Statement; and, if so, on what basis and with what consequences on the Horizon system?

10.2 Approach to the Issues: Pre-conceptions

1218 The issues 2, 9 and 14 are on the face of them mainly factual issues, which can be largely resolved by factual evidence, and might not in themselves lead to much expert disagreement.

1219 However, there are in my opinion certain pre-conceptions behind the issues; and having read Mr Coyne's report, his report appears to reinforce some of those pre-conceptions. I address those pre-conceptions before addressing the detail of the issues.

1220 A pre-conception underlying all three issues appears to be that providing more information to SPMs would have been a good thing - enabling them to understand bugs and defects in Horizon (issue 2) identify discrepancies (issue 9) and dispute discrepancies (parts of issue 14).

FUJ00083961

1221 This seems to me to make assumptions about the role and knowledge of SPMs, and about their relationship with PO, which should not be accepted without question - and some of which, once examined, turn out to be unrealistic.

1222 In my experience of many types of IT project, expecting too much knowledge of the users - more than they need in order to use the system - is a common mistake in the design of systems, and can often make systems harder to use, and make users more likely to make errors and dislike the system. If SPMs were expected to understand bugs and defects in Horizon (as in issues 2 and 14), that would require them to understand a large amount of detail about the Horizon back-end systems - their names, roles, interactions and so on - all of which has nothing at all to do with the SPM's daily work.

1223 Suppose, as asked in Horizon Issue 2, Horizon had automatically produced some error message of the form 'Transaction X has resulted in a discrepancy in data between TPS and DRS', the only possible reaction from an SPM would be: "What on earth am I supposed to make of that?". To make any such message meaningful to the SPM would require a large amount of extra documentation and training material, which would be of no use to him in his day-to-day work. He would never spend the time to understand what is happening behind the scenes.

1224 In my experience, for any IT system to subject its users to its internal details is usually a mistake. Users typically want to know as little as possible about the internal details of the system. Good design always involves 'information hiding' and keeping things as simple as possible for users. The approach of providing a help desk, where a person could try to understand what problem the SPM was experiencing, and try to help him, was in my opinion the only viable one.

1225 In a context unrelated to the Post Office, we have all experienced the frustration we feel when some human help service has been replaced by a machine - for cost-saving reasons. Issue 2 is essentially asking - should PO have offered its SPMs less human support to save costs (even if this had been technically possible)?

1226 Similarly, there seems to be an assumption behind issues 9 and 14 that, given enough automated information, SPMs could somehow identify the causes of shortfalls (deep inside Horizon), and might have the knowledge and persistence to 'dispute' them with Fujitsu support staff, who spend all their time looking at such issues, and who have a deep knowledge of Horizon internals.

1227 This whole assumption seems to me to be a misconception. It may arise in part because lay people (such as SPMs and lawyers) do not understand that, for a variety of reasons (such as the need for robustness countermeasures, the needs of many different classes of users, the obduracy of technology, and the evolution of systems over many years), the internal details of any large IT system are always much more complex than you would expect. There is just much more code needed that you would imagine. Many failed IT projects are a testament to this.

1228 Horizon has taken more than 3,000 man-years of effort to build. To imagine that any SPM can, in his spare time when he is not managing his branch, understand enough about Horizon internals, and how they might or might not go wrong, to debate and dispute the causes with PO and Fujitsu seems to me a fantasy. It is an unrealistic

FUJ00083961

view of the knowledge and predispositions of Horizon's main users. IT developers are prone to have too high expectations of their users' knowledge; but this level of misconception would seem to be very unlikely.

1229 The true picture, it seems to me, is simpler. SPMs know what happens in their branch, and they should know how to use Horizon. Fujitsu support staff know all about Horizon, and what may go wrong for a variety of reasons. When some anomalous incident occurs, the only possible way to understand it is by a cooperation between these two parties, sharing their knowledge. Without that cooperation, problems cannot be understood, and will keep on repeating themselves. The Fujitsu support role requires deep and broad knowledge of Horizon internals - used to filter the large amounts of information available, to find the pieces relevant to some problem. To pass that responsibility over to the SPM would in my opinion be wholly inappropriate - they have neither the knowledge or the time to do so.

1230 There also seems to be a pre-conception behind the claimant's case that it was somehow in the interests of PO and Fujitsu not to understand problems properly, but sometimes to 'fob them off' as a mistake by the branch - because it was easier to do that, and PO might make a little more profit out of the branch in that way.

1231 I question whether this approach would have been in PO's interests. If some problem is fobbed off - as being the fault of the SPM (when it is really a software bug) - then it will keep cropping up again and again, adding to support costs - while the resulting discrepancies in accounts may or may not benefit PO, unpredictably; and will create uncertainty. The support costs are real, and both PO and Fujitsu want to minimise them - by staying on top of problems.

1232 Any competent IT support operation is grateful to its users, when they draw its attention to any problem which can be fixed, to reduce the future costs of support. It will use these opportunities to improve the system for all users - not to fob some users off. Repeated evidence in KELs shows that Fujitsu ran such a competent support operation. The great majority of KELs show problems solved.

1233 A final pre-conception to be addressed here (or maybe elsewhere?) is that the support function would always start from the assumption that any problem had arisen from an error in the branch and would not give sufficient credence to the possibility that it might have arisen from a software error.

1234 This issue deserves careful consideration, because the evidence shows that human errors in the branch did occur much more frequently than errors induced by software - as one would expect. One measure of errors in the branch is the level of TCs - of which a large proportion arose from human errors, for instance in remming cash in or out, or in manually recovering recoverable transactions. These occurred at a rate of approximately one per branch per month; whereas software errors, as shown by the KELs, were much less common.

1235 Furthermore, a software error, once diagnosed, could be permanently fixed across the whole PO network; but human errors would keep on recurring.

1236 So, when the support desk was contacted about some problem, the overwhelming likelihood was that the cause really was a human error. The starting assumption, that it was probably an error in the branch, was correct in most cases. A software error was like a needle in a haystack.

FUJ00083961

1237 In these circumstances, some other strong evidence would be required, to show if the cause was a software error, rather than human error. That strong evidence could not be a single SPM saying: "I swear I never did that.". It would have to be something stronger than that, such as a recurring pattern across several incidents, or evidence from system logs.

1238 Evidence from the KELs persuades me that the Fujitsu support service was effective at spotting recurring patterns, and at delving into logs and other evidence to find the true causes of problems. If there was a software error, in my opinion the possibility of human error could usually be eliminated. Of course, the support service might not get it right every time; and even on the occasions when they correctly attributed a problem to human error (i.e. most occasions), sometimes the SPM might cling to a different account that he had never done anything wrong. This is a natural human reaction.

10.3 Horizon Issue 2

1239 **Issue 2**: Did the Horizon IT system itself alert Subpostmasters of such bugs, errors or defects as described in [Issue] (1) above and if so how?

1240 In common with most IT systems, Horizon generates messages to report the occurrence of certain errors to users and operators. Error messages displayed on the counter screen or presented on reports alert SPMs to conditions that may indicate the presence of bugs or other defects as described in Issue 1.

1241 I agreed the following with Mr Coyne in our Joint Statement: '*The extent to which any IT system can automatically alert its users to bugs within the system itself is necessarily limited. While Horizon has automated checks, which would detect certain bugs, there are types of bugs which would not be detected by such checks.*'

1242 Further, as I discussed in section 10.2 above, it would be counter-productive for Horizon to alert its users with precise details of abnormal conditions beyond their day-to-day experience of the system – for example, in back-end and other systems remote from their counters. To do so, it would need to assume terminology and knowledge well beyond that of a typical SPM.

1243 The system does, however, record significant or unexpected events in logs. Horizon is operated by specialist staff, who are alerted by the system if certain events occur. Such alerts trigger investigations that may detect bugs, which could potentially affect SPMs and their accounts before any branch users become aware that anything is wrong. The logs may also be checked proactively by the support team in response to a report from an SPM. These measures are amply substantiated in KELs.

1244 Horizon and its ecosystem are underpinned by a complex set of software, hardware networks and business processes. In my opinion, it is more rational that any bugs or other defects are investigated and analysed by experienced people following mature processes – rather than expecting that SPMs themselves could diagnose problems if they were given more detailed information.

1245 To summarise my opinion on this issue, Horizon did not in general alert SPMs to any significant bugs or other defects in the system itself. Nor should it have done.

FUJ00083961

10.4 Horizon Issue 9

1246 **Issue 9**: At all material times, what transaction data and reporting functions (if any) were available through Horizon to Subpostmasters for:

a. identifying apparent or alleged discrepancies and shortfalls and/or the causes of the same; and

b. accessing and identifying transactions recorded on Horizon?

1247 This issue focusses on the functions available to SPMs for identifying and investigating discrepancies. The discrepancies in question are differences between the amounts of cash and stock held by the branch and the amounts calculated by Horizon. Shortfalls are discrepancies where the figures held in Horizon are higher than those declared by the SPM. Discrepancies may also be surpluses in favour of the SPM.

1248 In Horizon, balancing means counting all cash and stock holdings and checking that the position matches the figures held within the system. This is the process that identifies any discrepancies.

1249 If a discrepancy is detected, the question is: why is it there? Of course, there are many reasons why such discrepancies occur. These include the following:

- Transactions may not correctly record the changes that occurred in cash or stock levels, e.g. Horizon was told that £30 was paid out to a customer whereas £50 was actually paid.
- Changes in cash or stock were not recorded at all, e.g. a book of stamps was mislaid.

1250 Horizon has always provided SPMs with a comprehensive suite of reports, which can be previewed on screen as well as printed. More than one hundred reports are available[35]. These include:

- reports by stock unit (SU) on a daily or weekly basis, or by user;
- balance reports; and
- journals such as Transaction and Event Logs.

Not all of these are relevant for dealing with discrepancies.

10.4.1 Accounting procedures

1251 What follows in this sub-section describes primarily the procedures in place since the introduction of Branch Trading in 2005. Prior to that, a Cash Account was in use. The procedures followed in both periods of time are similar. Some of the differences are discussed in section 10.5.4 below.

1252 PO requires that cash is declared for every SU every day.[36] This entails counting the cash held within that SU and entering this information into Horizon. The system will display any discrepancy between the total figure declared and the system derived figure.

1253 Branches must perform a balance at the end of each Trading Period (TP). To help maintain control over their accounts and cash holdings, branches are advised to balance on a weekly basis. Balancing can be performed at any time. Larger or busier branches may choose to balance every working day.[37]

35 SD/DES/005 for the original system and DES/GEN/SPE/0008 for HNG

36 Post Office Onboarding - Counter Guide v6.0

37 Several paragraphs in this part of the report are based on factual evidence provided in the Witness Statement of David Johnson, PO's Training &

FUJ00083961

1254 The more frequently branches complete cash declarations and balance reports, the sooner they will be able to identify any discrepancies. Staff are more likely to recall an interaction with the customer which may have caused the discrepancy.

1255 There are two categories of report available from Horizon: Counter and Office reports. Counter reports provide details of transactions carried out by a specific SU, whereas Office reports cover all SUs – in other words, the entire branch. In branches with only one SU, both categories of report show the same information.

1256 When discrepancies are identified, the main tool used to find their causes is the Transaction Log. This allows any user with access to Horizon to obtain a chronological list of the transactions completed in the branch. The log can be used to browse through a list of transactions, or the output can be filtered by selection criteria such as TP, date, time, SU, user, product type or value. Regardless of TP dates and the intervals between balancing branch accounts (Balance Periods), the log can be used to investigate up to 60 days back in time. Prior to 2010 with the original Horizon system, this period was 42 days.

1257 A user may spot errors they have made in entering data into Horizon or when handling cash or stock by examining the transaction log. For example, a user may recall giving a customer a cash withdrawal of £100 at a particular time of the day, but by checking the log they may spot that they incorrectly processed the transaction as a deposit. This would create a shortfall of £200 in the branch accounts (Horizon will think that the user has taken a £100 deposit whereas in fact the user has given the customer £100).

1258 The following specification shows the detailed layout of the Transaction Log (using illustrative data):

Audit Advisor.

FUJ00083961


```
          1         2         3         4
01234567890123456789012345678901234567890123
01 Chelsea PO                      FAD: 123456X
02 12:42 17/01/2008      TP:10  BP:01  SU:SH1
03 Transaction Log - Office Copy
04
05 USER          TRANSACTION REF  SU  TP  BP
06 DATE       TIME
07 MODE PRODUCT             VOLUME      VALUE
08
09 EPR001       1-34414-1        SH1 10  02
10 17/01/2008 12:10
11 RIAD Colombia Peso                 1000.00
12                      18225000
13 ----------------------------------------
14
15 EPR001       1-34418-1        SH1 10  02
16 17/01/2008 12:10
17   BC NS&I Cash Dep           1-     55.00-
18 ----------------------------------------
19
20 EPR001       1-34418-2        SH1 10  02
21 17/01/2008 12:10
22   BC Cash                    1       75.00
23 ----------------------------------------
24
25 EPR001       1-34423-3        SH1 10  02
26 17/01/2008 12:10
27 RIAD 1st class stmp       1000       0.00
28 ----------------------------------------
29
30 EPR001       1-34423-4        SH1 10  02
31 17/01/2008 12:10
32 RIAD 2nd class stmp          1-      0.00
33 ----------------------------------------
34
35 EPR001       1-34423-5        SH1 10  02
36 17/01/2008 12:10
37 RIAD Roll 2nd x 500   9999999- 9999999.99-
38 ----------------------------------------
39
40 EPR001       1-34423-6        SH1 10  02
41 17/01/2008 12:10
42 RIAD PO phonecard £10          9999999.99-
43  999999999-
44 ----------------------------------------
45
          *** END OF REPORT ***
          1         2         3         4
01234567890123456789012345678901234567890123
```

Figure 10.1 - Transaction Log layout

1259 The printable area for a counter printer is only 8cm wide. Each transaction printed on the log uses a minimum of four lines. Therefore, the length of a printed Transaction Log invariably becomes unwieldy – meaning that branch staff may find it difficult to use and store such logs.

1260 The **Event Log** reports, in chronological order, events that have taken place. Outputs can be filtered using selection criteria such as date, SU, TP, and user. This log enables the user to see all the cash declarations that have been made, by which user and whether there were any discrepancies. If there was a discrepancy in a given SU followed by another cash declaration with no discrepancy, the SPM may wish to balance that particular SU to check whether the user has made an accurate cash declaration and has not concealed a discrepancy.

1261 The **Balance Snapshot** shows details of all receipts and payments since the last time an SU was balanced. It clearly identifies discrepancies (if any) and shows the stock on hand as described in the following paragraph. It can be used to check receipt and payment transaction totals.

FUJ00083961

1262 The **Stock On Hand report** shows the derived positions of cash, cheques (if applicable), stock, foreign currency, stamps and other stock on hand. This means that the user can check the physical stock on hand in the branch against the system derived figures at any time and see if there is any discrepancy.

1263 Other reports are also used for checking the details of specific transactions identified using the tools described above.

10.4.2 Opinions

1264 To address the two parts of this issue explicitly, (b) '*accessing and identifying transactions recorded on Horizon'* is the prime method of (a) '*identifying apparent or alleged discrepancies and shortfalls and/or the causes of the same*'.

1265 I have agreed the following with Mr Coyne in our Joint Statement: '*The causes of some types of apparent or alleged discrepancies and shortfalls may be identified from reports or transaction data available to Subpostmasters. Other causes of apparent or alleged discrepancies and shortfalls may be more difficult or impossible to identify from reports or transaction data available to Subpostmasters, because of their limited knowledge of the complex back-end systems. Identification requires cooperation of PO staff and subpostmasters.*'

1266 In my opinion, most discrepancies are caused by human error. The functions available from Horizon, when used in accordance with PO guidance and procedures, enable SPMs to identify the causes of such discrepancies. Indeed, SPMs and their staff are uniquely placed to investigate discrepancies, because they are the only people who have first-hand knowledge of what happens in their branches. The PO and Fujitsu support teams can only use their knowledge of systems and the data stored within them; whereas the SPM can use their knowledge of what happens in the real world.

1267 The main concern of an SPM is the successful running of their branch. This means that they may have limited time and patience to investigate discrepancies in Horizon. The reports available to them focus on activities carried out within the branch, their key area of expertise. If they are, nevertheless, unable to identify the problem, their best course of action is to ask for help.

10.5 Horizon Issue 14

1268 **Issue 14**: How (if at all) does the Horizon system and its functionality:

a. enable Subpostmasters to compare the stock and cash in a branch against the stock and cash indicated on Horizon?

b. enable or require Subpostmasters to decide how to deal with, dispute, accept or make good an alleged discrepancy by (i) providing his or her own personal funds or (ii) settling centrally?

c. record and reflect the consequence of raising a dispute on an alleged discrepancy, on Horizon Branch account data and, in particular:

i. does raising a dispute with the Helpline cause a block to be placed on the value of an alleged shortfall; and

ii. is that recorded on the Horizon system as a debt due to Post Office?

FUJ00083961

d. enable Subpostmasters to produce (i) Cash Account before 2005 and (ii) Branch Trading Statement after 2005?

e. enable or require Subpostmasters to continue to trade if they did not complete a Branch Trading Statement; and, if so, on what basis and with what consequences on the Horizon system?

1269 I respond to each question in the following sub-sections.

10.5.1 Stock and cash

a. *How (if at all) does the Horizon system and its functionality enable Subpostmasters to compare the stock and cash in a branch against the stock and cash indicated on Horizon?*

1270 As described in section 10.4 above, Horizon enables the SPM to make declarations of stock and cash. The values held by Horizon can be seen using reports such as the Stock On Hand report.

10.5.2 Resolve discrepancy

b. How (if at all) does the Horizon system and its functionality enable or require Subpostmasters to decide how to deal with, dispute, accept or make good an alleged discrepancy by (i) providing his or her own personal funds or (ii) settling centrally?

1271 I assume that this part of the issue refers to part (a). In that case, the discrepancy in question is between the stock and/or cash declared by the SPM and Horizon's view of those amounts. It is unclear to me who has 'alleged' this discrepancy.

1272 The remainder of this section 10.5 deals with the current generation of Horizon (HNG). Earlier changes are discussed in section 10.5.4 below.

1273 Horizon requires the SPM to deal with any discrepancies before rolling over their accounts to the next TP. The SPM is free to ignore discrepancies until that point. They are also free at any time to make further declarations to correct a discrepancy, so long as they make a corresponding adjustment to the physical cash on hand.

1274 At the end of the TP, the user is sent a message that indicates the amount of any discrepancy. The system invites the user to transfer this amount into the local suspense account and continue to roll over – or to discontinue this operation.[38]

1275 If, at the end of a TP, the Branch Trading Statement (BTS) shows there is a discrepancy (i.e. either a surplus or a shortfall) of less than £150, the SPM must 'make good' the discrepancy – either by removing money from the till (in the event of a surplus) or by adding money to the till (in the event of a shortfall). This would mean that the discrepancy was accepted, and the next TP can begin with a balanced account.

1276 If, at the end of a TP, a branch has a discrepancy of more than £150, they have the option to either make good or settle the discrepancy centrally. If the SPM chooses the latter option, they do not have to physically place cash in the till (in the case of a shortfall) at that point. Instead, a message is sent to PO's Finance Services Centre and the discrepancy is moved to a central account held in the SPM's name.[39]

38 Several paragraphs in this part of the report are based on factual evidence provided in the Witness Statement of David Johnson.

39 Further details of this business process are provided in the Witness Statement of Dawn Phillips, the leader of the PO Agent Accounting team who oversees the process of resolving discrepancies that SPMs have chosen to settle centrally.

FUJ00083961

1277 An SPM may wish to dispute a discrepancy. If so, the SPM can either call the NBSC Helpline or the Agent Accounting team (where the discrepancy is over £150) for assistance. The SPM cannot dispute a discrepancy on Horizon or record that they have raised a dispute.

1278 The Agent Accounting (AA) team investigates disputes, with the aim of resolving them promptly.

10.5.3 Recording disputes

c. How (if at all) does the Horizon system and its functionality record and reflect the consequence of raising a dispute on an alleged discrepancy, on Horizon Branch account data and, in particular:

i. does raising a dispute with the Helpline cause a block to be placed on the value of an alleged shortfall; and

ii. is that recorded on the Horizon system as a debt due to Post Office?

1279 Horizon does not record disputes. In response to question (i) of this issue, raising a dispute about a discrepancy (either via the NBSC Helpline or directly with the AA team) causes a block to be placed on the SPM's central PO account until the dispute is resolved. The effect is that the discrepancy is not pursued with the SPM. There is no impact on Horizon. Therefore, in response to question (ii), the discrepancy is not recorded on Horizon as either a debt due to PO or a credit due to the SPM. It is simply recorded that the discrepancy is being settled centrally. As a result, the branch accounts are restored to balance.

10.5.4 Accounting statements

d. How (if at all) does the Horizon system and its functionality enable Subpostmasters to produce (i) Cash Account before 2005 and (ii) Branch Trading Statement after 2005?

(ii) Branch Trading Statement after 2005

1280 Each branch is required to perform a full balance of every SU in the branch at the end of each TP. Before the final balance report is produced, the SPM must make declarations of stock on hand, foreign currency, stamps, travellers cheques and cash. After the balance report, a Postage Label report must be produced. The next step is to complete the Suspense Account report for the branch. Once all of the stock units in a branch have been balanced and rolled over to the next TP, the BTS can be produced.

1281 The BTS shows an overall summary by stock unit and it also has a list of stock on hand. The user must check the statement and sign it off as accurate.

(i) Cash Account before 2005

1282 Branch Trading was introduced in 2005. Before that, in the original Horizon system, branches had to produce a Cash Account rather than a BTS. The Cash Account fulfilled the same role, but it had to be produced weekly rather than monthly. It could only be done after a series of steps similar to those required today in the build-up to the BTS.

1283 Prior to the introduction of TCs along with Branch Trading, Error Notices were used to correct accounting errors made in branches.

FUJ00083961

10.5.5 Continuing to trade

e. How (if at all) does the Horizon system and its functionality enable or require Subpostmasters to continue to trade if they did not complete a Branch Trading Statement; and, if so, on what basis and with what consequences on the Horizon system?

1284 The PO guide to Branch Trading (balancing and despatch of documents)[40] advises SPMs, in section 29, that '*You must produce a Branch Trading Statement on the last working day on or before the Branch Trading Period end dates shown on your Branch Trading calendar.*' David Johnson asserts in his Witness Statement[41], however, that it is possible for a branch to continue to trade without completing a BTS.

1285 The guide also states in section 19:

'*If you have a stock unit in your branch that has not been rolled over to the next Balance/Trading Period within the last 38 days, the Horizon system will display a screen prompt to remind you to do this.*

Please remember: If you do not follow the screen prompt instructions promptly, (by the end of the working day at the very latest) the Horizon system could reach its capacity limit which may result in the loss of transaction information.

If you ignore the warning messages informing you to roll your stock units over and your transactions archive, your branch could be closed for up to four weeks.'

1286 Thus, completing the BTS is an administrative requirement, rather than a technical constraint on continuing to trade. The critical condition, which enables SPMs to continue to trade is the rolling over of all SUs to the next TP within 38 days of the previous branch rollover. Note that the longest TP is five weeks (35 days), so there is some leeway to allow for weekends and public holidays.

10.5.6 Opinions

1287 Issue 14 is almost entirely factual, asking *how* Horizon supports SPMs in dealing with discrepancies in their branch accounts. I have given my answers above.

1288 Regarding part (c) of the issue, it seems incongruous to me that Horizon does not reflect the status of a discrepancy that is settled centrally. The consequence is that SPMs work with PO entirely outside the system with little visibility of where they stand.

10.6 Mr Coyne's opinions

1289 Mr Coyne and I agree[42] that SPMs are not able to investigate every discrepancy. Indeed, it would not be reasonable to expect that they could. Nevertheless, Mr Coyne points out in paragraph 8.20 of his expert report that the information available from Horizon '*would not allow a Subpostmaster to determine whether a transaction has reconciled at APS Host or at any other level (harvester, client, etc.)*'. That expectation seems to me unrealistic.

1290 Issue 14 relates to the process of dealing with discrepancies in branch accounts. In responding to parts (b) and (c) though, between paragraphs 7.27 and 7.33 Mr Coyne focusses more on TCs, which are a mechanism for correcting discrepancies.

---

40 Version 7 December 2006

41 Paragraph 48.5

42 See paragraph 1261 above

FUJ00083961

1291 Issue 14 asks about Horizon. In part (c) (i), the experts are asked whether '*raising a dispute with the Helpline cause[s] a block to be placed on the value of an alleged shortfall*'. I take this to mean a block in Horizon. As Mr Coyne says, Dawn Phillips confirms in paragraph 10 of her Witness Statement that her team '*will place a block in their account on the system until the dispute is resolved*'. However, in paragraph 51 of his Witness Statement, David Johnson explains that when a discrepancy is to be settled centrally, '*the discrepancy is moved to a central account held in the Postmaster's name*'. This account is managed on one of PO's central systems rather than Horizon.

1292 In question (c) (ii), the experts are asked whether the shortfall is recorded on the Horizon system as a debt due to Post Office. Mr Coyne states in his paragraph 7.37, '*A loss is recorded as a debt to the Post Office in the event the discrepancy is upheld by the Post Office following any dispute.*' As David Johnson explained though, the discrepancy is moved to a central account, which is not held on Horizon.

1293 In relation to question (e), Mr Coyne states in his paragraph 7.39: '*Subpostmasters are not able to continue trading until Branch Trading Statement process is complete.*'. This is contradicted by David Johnson in his paragraph 48.5: '*It is possible for a branch to continue to trade if they do not complete a BTS'.*

1294 In para 8.18, he cites PO's Business Case for a Cash Management Improvement Programme [replicate reference]. The purpose of this programme is to '*establish more effective ways of managing our cash inventory and cash flow across the POL business*'. The business case identifies nine benefits of the programme. Mr Coyne explains that the programme seeks to improve the facilities available to SPMs for detecting, investigating and correcting any cash discrepancies. He goes on to adduce that this proposal '*confirmed that the Subpostmaster has little control beyond counter level when trying to resolve any discrepancies*'.

1295 While the definition of the proposed programme acknowledges scope for improvement, alongside eight other benefits, it does not confirm any broader lack of control for the SPM.

FUJ00083961

11. EXPERT ISSUES – FACILITIES AVAILABLE TO POST OFFICE

11.1 The Issues

1296 In this section, I address Horizon Issues 7, 8, and 10-13, which concern facilities available centrally to Post Office centrally or to Fujitsu, rather than to Subpostmasters.

1297 **Issue 7**: Were Post Office and/or Fujitsu able to access transaction data recorded by Horizon remotely (i.e. not from within a branch)?

1298 **Issue 8**: What transaction data and reporting functions were available through Horizon to Post Office for identifying the occurrence of alleged shortfalls and the causes of alleged shortfalls in branches, including whether they were caused by bugs, errors and/or defects in the Horizon system?

1299 **Issue 10**: Whether the Defendant and/or Fujitsu have had the ability/facility to: (i) insert, inject, edit or delete transaction data or data in branch accounts; (ii) implement fixes in Horizon that had the potential to affect transaction data or data in branch accounts; or (iii) rebuild branch transaction data:

b. at all;

c. without the knowledge of the Subpostmaster in question; and

d. without the consent of the Subpostmaster in question.

1300 **Issue 11**: If they did, did the Horizon system have any permission controls upon the use of the above facility, and did the system maintain a log of such actions and such permission controls?

1301 **Issue 12**: If the Defendant and/or Fujitsu did have such ability, how often was that used, if at all?

1302 **Issue 13**: To what extent did use of any such facility have the potential to affect the reliability of branches' accounting positions?

11.2 Interpretation of the Issues

1303 In approaching issue 7, I need to choose whether the word 'access' applies to read-only access, or to the ability to access and change records. The term can be used in either sense.

1304 As in his para 3.16, Mr. Coyne uses the term 'access' in the sense of access and change. However, it seems to me that Horizon issue 10 addresses the aspect of changing records, and to make issue 7 distinct from issue 10, it needs to refer to 'access' in the read-only sense.

1305 The only problem is that this makes issue 7 rather trivial - and then perhaps a subset of issue 8. Fujitsu were able to access data in this sense; it was essential, for them to be able to support the system.

1306 Issue 8 asks what data and reports were available to Fujitsu to diagnose problems. It largely concerns the robustness countermeasures RDS and MID, and how they operated. So, there is some overlap with section 7 of this report, which discusses those countermeasures.

1307 Issues 11 and 12 all refer to issue 10, so are linked to it. They are largely factual.

1308 Issue 10 (i) refers to the ability to change transaction data and is in my view the main part of issue 10 which needs an extended answer. Issues 11 and 12 then refer to the controls on that ability, and to evidence about its use.

FUJ00083961

1309 Issue 10 (ii) asks whether Fujitsu could implement fixes in Horizon (which they could) and whether some of those fixes could affect transaction data. Naturally, many types of fix can affect future transaction data, by changing the behaviour of the software. In the overwhelming majority of cases, these changes are beneficial, in making the transaction date more likely to be accurate.

1310 Issue 10 (iii) asks whether Fujitsu could rebuild transaction data. As I understand it, this would only be done in old Horizon as a part of branch hardware changes, with the SPM involved. The word 'rebuild' implies rebuilding from other data stored redundantly elsewhere. Thus, issue 10 (iii) refers to a technical robustness measure, using RDS to deal with an identified problem - rather than some discretionary change to transaction data. For the latter, as far as I know, there is no evidence that it actually happened, apart from the Witness Statement of Richard Roll.

1311 Horizon Issue 13 then asks about the extent of the impact on branch accounts. As for Horizon issue 1, 'extent' may be measured in two ways - either as the number of distinct incidents, or as their net financial impact. In my opinion, for Issue 13, the second sense is more useful. As I shall describe below, I have attempted to assess the maximum possible financial impact of all the changes listed under Issue 10, on claimant's branch accounts. [do not anticipate conclusions]

11.3 Horizon Issue 7

1312 **Issue 7**: Were Post Office and/or Fujitsu able to access transaction data recorded by Horizon remotely (i.e. not from within a branch)?

1313 In the previous sub-section, I have explained that I am interpreting 'access' as 'access to read'.

1314 'Transaction data' in Horizon derive from the following sources only[43]:

a) Counters, which record each individual exchange of cash and products with branch customers

b) TCs and TAs as discussed in section 6.4.

c) Balancing Transactions as discussed in section 11.5 below.

1315 All transaction data from every branch is transferred over the Horizon network to central servers managed by Fujitsu, from where it is distributed to other systems used by Post Office. In HNG, Post Office can remotely examine the data held in the Branch Database (BRDB) in read-only mode for business reasons, such as monitoring the levels of cash held in branches.[44] Post Office also access data derived from BRDB in systems such as Credence and Horise, In this way, both Fujitsu and PO have been able to read the data remotely.

1316 Fujitsu needs remote access for support purposes.

11.4 Horizon Issue 8

1317 **Issue 8**: What transaction data and reporting functions were available through Horizon to Post Office for identifying the occurrence of alleged shortfalls and the causes of alleged shortfalls in branches, including whether they were caused by bugs, errors and/or defects in the Horizon system?

43 Witness Statement dated 27 September 2018 - Torstein Godeseth (Fujitsu's Chief Architect), paragraph 17.2

44 Witness Statement – Steve Parker (Fujitsu's SSC Manager), paragraph 14

FUJ00083961

1318 This issue relates to 'alleged' shortfalls in branches and their causes, if they were caused by Horizon bugs. The used of the word 'alleged' implies that the experts should consider shortfalls reported by SPMs.

1319 As I explained under the previous issue, PO has access to all branch transaction data. It uses this information for several distinct purposes, which include the following:

- to reconcile business transacted at the counters with PO's clients[45];
- to manage their own business by analysing the details of what happens in branches; and
- to assist them in investigating anomalies reported by SPMs.

1320 In sections 4.3 and 5.4 above, I describe Horizon's back-end architecture. The purpose of the Transaction Processing System (TPS) is to gather the transactions taking place in the branches, and to pass them on both to Horizon's back-end systems such as APS and DRS, and to other IT systems in Post Office. PO's systems include Credence and a succession of systems based on SAP, which have culminated in POLSAP.

1321 PO's access to branch transaction data, via a suite of Horizon reports and via their own management information systems, serves to improve the robustness of the system. Storing the data in PO's systems and cross-checks between those and Horizon contribute to RDS. Because PO uses the data for its own purposes, MID also comes into play.

1322 In parallel with TPS, Horizon ensures that an accurate record of all transactions is secured in the audit store[46]. When PO is investigating anomalies reported by SPMs, they use Credence and their other management information systems in the first instance[47] – but, when they need to confirm the transactions handled in a branch, they can also ask Fujitsu to retrieve the corresponding data from audit.

1323 Thus, the information required to investigate alleged shortfalls is available to PO from several sources. Their perspective is to look into branch accounts from the outside, with no first-hand knowledge of what has occurred from day to day. On the other hand, they look out to their external clients on whose behalf they are brokering business based on those clients' services and products. By virtue of their role in the end-to-end business, PO has access to information not available to SPMs and vice versa.

11.5 Horizon Issue 10

1324 **Issue 10**: Whether the Defendant and/or Fujitsu have had the ability/facility to: (i) insert, inject, edit or delete transaction data or data in branch accounts; (ii) implement fixes in Horizon that had the potential to affect transaction data or data in branch accounts; or (iii) rebuild branch transaction data:

a. at all;

b. without the knowledge of the Subpostmaster in question; and

c. without the consent of the Subpostmaster in question.

45 See section 6.4 of this report

46 See section 4.4 of this report

47 See the Witness Statement of Tracy Mather dated 16 November 2018

FUJ00083961

1325 This issue relates to both PO and Fujitsu. It comprises three parts, numbered (i) – (iii). The experts are being asked to examine each part in three different respects identified as (a) – (c). Therefore, in principle, the issue calls for 18 opinions although they are not all distinct.[48]

1326 In the text that follows, I provide my opinions on each part of the issue. These opinions are summarised near the end of this sub-section.

(i) Insert, inject, edit or delete transaction data or data in branch accounts

1327 In my opinion, 'inject' means the same as 'insert'. Insert/inject and edit/delete are treated as separate cases. '*Transaction data or data in branch accounts*' excludes reference data. I am taking this Issue 10 to include local as well as remote access.

Post Office

1328 In paragraph 1310 above, I describe three classes of transaction. One was TCs and TAs.

1329 PO effectively uses TCs as the means to introduce new transactions into a branch's accounts (to correct errors) - but not directly. Therefore, in my opinion, this is not a way for PO to inject transactions. SPMs may dispute a TC and ask for further investigation – whereas they do not have this opportunity with transactions that have simply been inserted into their accounts.

1330 Before TCs were introduced in 2005, Error Notices fulfilled a similar function. These were sent to the branch on paper and manually entered into Horizon by a user in the branch.

1331 TAs are used to allow SPMs to accept transactions involving certain PO clients such as Camelot and Paystation into their accounts but, once again, I do not see this as PO inserting transactions.

1332 Within Horizon, each branch was set up to include a number of user accounts that could be accessed in all branches.[49] PO uses these accounts for certain branch operations such as opening /closing branches and training. So-called 'Global Users' correspond to specific roles, such as:

- Engineer – with test capabilities branch diagnostics and maintenance
- Migrate - used to open new branches
- Setup - used by mobile or relief managers
- Auditor - may view users and stock units but not carry out transactions
- Auditor Emergency Manager - used to run branches.

1333 Because Global Users are given the capability to run branches, in the same way as SPMs, they are also able to inject transactions into the accounts. Any transaction entered by a Global User is included in the transaction log against that user. The usernames start with an asterisk to differentiate them from other users.

1334 SPMs can only add new transactions to their accounts and not change or remove any existing ones. Therefore, Global Users cannot edit or delete transactions either.

48 Several of the opinions expressed here use evidence provided by Torstein Godeseth in his Witness Statements.

49 See ARC/SOL/ARC/0006

FUJ00083961

1335 These roles are performed within the branch concerned, rather than remotely. Therefore, it is likely the SPM will know that the PO user is on Horizon. They can also check their logs to find out the details of any transactions carried out.

1336 Once again, because Global Users are on Horizon within the branch, they will normally have co-ordinated their work with the SPM. However, they may not require consent as such.

Fujitsu

1337 Fujitsu users from the SSC (Software Support Centre) have the ability to inject additional transactions into a branch's accounts in HNG, using a Balancing Transaction (BT).[50] Those users are not permitted to amend or delete any transactions.

1338 BTs could be used to rectify any erroneous accounting data that may have been recorded as a result of a bug in the Horizon Counter or BAL. They are inserted using the Host BRDB Branch Transactional Correction Tool.[51] This powerful tool could cause serious problems to the Branch Database in the event of certain bugs. Therefore, its usage is limited to a small pool of SSC users who could be made aware of these consequences.

1339 BTs are clearly visible in the transaction reports that are available to SPMs via Horizon as they are stated to have been carried out on counter number 99[52]. BTs do not require acceptance by SPMs, unlike TCs and TAs.

1340 In the original Horizon system, the SSC could also inject transactions. These were also clearly identified as such. SSC users were able to update branch accounts without the consent of the SPM.

1341 However, Fujitsu could not edit or delete transaction data. As Torstein Godeseth[53] confirms: '*All accounting at the counter was carried out based on the data held in the message store. The Riposte product managed the message store and it did not allow any message to be updated or deleted*'.

1342 Under HNG, certain Fujitsu staff (Privileged Users) have access privileges that could be used to edit or delete transaction data in the BRDB. This level of access is needed for system maintenance purposes, such as updating database records to help implement planned system changes. However, Fujitsu has no process that requires transaction data to be amended or deleted. Standard Horizon functionality, such as TCs and BTs, can be used to resolve most errors that may affect branch accounts. There is therefore little need to use privileged access to manipulate transaction data to resolve an error.

1343 Any change to a transaction performed by a Privileged User would be visible to branch staff. The amended transaction would appear in reports and logs that can be viewed in branch, although it would not be flagged as a change by a Privileged User. Theoretically this is a problem, but Privileged Users cannot change the audit record and so the changed record in the BRDB would no longer match an audit extract. This means that an SPM could always find out about changes made by SSC, via a request to the helpdesk.

---

50 See DES/APP/HLD/0020 section 5.6.2

51 See DEV/APP/LLD/0142 for the Low Level Design

52 Counter 99 is readily identifiable, because it would indicate that there were 99 serving positions in a branch, which no branch has.

53 In his witness statement dated 27 September 2018

FUJ00083961

1344 In my experience, Privileged Users on Horizon have the same role as one would expect to see on any IT system. Such rights are necessary to ensure unforeseen events can be addressed if necessary. Consent is not required from the SPM for any changes in transaction data.

(ii) Implement fixes in Horizon that had the potential to affect transaction data or data in branch accounts

1345 PO cannot make changes in the Horizon software, but they do maintain its reference data, which is a vital component of the system.

1346 I have agreed with Mr Coyne '*that the very nature of rolling out fixes within any IT system, including those implemented by Fujitsu has the potential to affect transaction data or data in branch accounts.*'

1347 Fujitsu implements fixes in software and reference data. They could certainly implement fixes without the knowledge of SPMs. Indeed, such changes are not generally communicated to SPMs – unless exceptional circumstances apply. Where reference data apply to specific branches, it is more likely that the affected SPMs would be informed. However, many changes in that data apply across the system and the overwhelming majority would not be communicated explicitly to SPMs.

1348 In my opinion, any capability to communicate changes to SPMs needs to be used sparingly, to avoid flooding SPMs with information which is of little concern to them in their daily work.

1349 Fixes could also be applied without the consent of SPMs although, once again, changes in reference data that affect specific branches may be co-ordinated with the SPMs in question.

(iii) Rebuild branch transaction data

1350 In paragraph 1306 above, I explain my interpretation of this part of the issue. The word 'rebuild' implies re-creating from other data stored redundantly elsewhere. Thus, this part of the issue refers to a technical robustness measure, rather than some discretionary change to transaction data.

1351 In Legacy Horizon, the transaction database was stored locally within the branch and replicated with central servers using the Riposte product. It was possible for the data in a particular counter at the branch to become inconsistent with replicated copies or 'corrupt'. The latter term means that the data has been damaged so that it can no longer be used. In that situation, Fujitsu could intervene remotely to correct the problem - but branch transaction data was not changed in any way.[54] The workaround (WOR) involved replicating the correct data from another counter in the affected branch or from the data centre copy. The same technique was used to rebuild the counter database if branch hardware was changed.

1352 In HNG, BRDB is maintained centrally and so rebuilding is not needed for hardware changes.

1353 PO has not had the ability to rebuild branch transaction databases.

1354 The SPM could not use Horizon until the branch database is fully operational. In principle, the data could be rebuilt without the knowledge of the SPM in question, but they would be informed or become aware that they could use Horizon normally again and so they would know that something had happened.

54 Witness Statement – Steve Parker, paragraph 18 and paragraph 55

FUJ00083961

1355 Consent is not formally required but, if the data needed to be re-built before the SPM could use Horizon it is unlikely that they would object to this action.

Summary of opinions

1356 The following table summarises my opinions on each part of Issue 10:

	i. Data amendment	**ii.** Fixes	**iii.** Rebuilds
Whether PO have had the ability: a. At all	• Insert/inject: yes, Global Users have had that ability. • Edit/delete: no.	PO can change reference data, which I consider to be part of Horizon.	No
b. Without the knowledge of the SPM	No	Yes	Not applicable
c. Without the consent of the SPM	Yes	Yes	Not applicable
Whether Fujitsu have had the ability: a. At all	• Insert/inject: yes. - HNG: Only via Balancing Transactions (BTs). - Legacy: By SSC. • Edit/delete - HNG: Privileged Users.	Yes. Applies all fixes in Horizon (software or reference data), which could affect transaction data.	Yes. - HNG: The database is stored centrally where it could also be rebuilt. - Legacy: On branch hardware changes. Also, via Riposte.
b. Without the knowledge of the SPM	No. Any changes performed by Privileged Users become visible at the branch.	Yes	Yes, unless the hardware was being changed, which would have involved the SPM.
c. Without the consent of the SPM	Yes. Neither BTs nor Privileged Users require consent.	Yes	Yes, although hardware may have been changed at the SPM's request.

Table 11.1 - Issue 10 summary of opinions

11.6 Horizon Issue 11

1357 **Issue 11**: If they did, did the Horizon system have any permission controls upon the use of the above facility, and did the system maintain a log of such actions and such permission controls?

1358 This issue refers back to and therefore is linked to Issue 10, which is in three parts that I treat separately.

Issue 10 (i) Insert, inject, edit or delete transaction data or data in branch accounts

1359 PO Global Users are each assigned to a specific Role. Each role is limited to carrying out specific actions - such as logging on, viewing or entering data.[55] In the Legacy system, Global Users' access rights were strictly

55 For more information about Horizon's role-based access control, see section 6.6.2.

FUJ00083961

controlled by "one shot passwords"[56]. In HNG, the method of administration was improved.

1360 Similarly, only a small group of SSC users (30 according to Mr Godeseth) is permitted to create a Balancing Transaction (in HNG). By definition, the number of Privileged Users, who can edit or delete transaction data in BRDB, is also limited (to about 45).

1361 The High Level Design (HLD) document for the BRDB states[57] that '*Support teams will be restricted to accessing the Branch Database only under an MSC.*' I introduce the MSC process in section 6.8.6 above.

1362 The same HLD goes on to confirm:

'*There is a requirement that the SSC will have ability to insert balancing transactions into the persistent objects of the Branch Database. There are reasons for SSC having to do so e.g. to rectify erroneous accounting data that may have been logged as a result of a bug in the Counter / BAL.*

SSC will have privileges of only inserting balancing / correcting transactions to relevant tables in the database. SSC will not have any privileges to update or delete records in the database.

Any writes by the SSC to BRDB must be audited.'

1363 In Legacy Horizon, the ability to edit/delete transactions was also limited to SSC users.

1364 SSC's access to the counters has been strictly controlled. As Steve Parker explains:[58]

'*Some members of the SSC were (and some remain) able to insert transaction data. SSC access privilege gave the ability to inject transactions, but appropriate change controls were in place and no such insertion would have happened without complying with those controls.*'

1365 He continues in paragraphs 21.2 and 21.2:

'*Any transaction that was inserted would immediately cause a discrepancy to arise in the branch's accounts. For example, if a transaction were to be inserted which stated that £1,000 of stamps had been bought by a customer who paid cash, that would immediately cause a reduction in stock levels of stamps in that branch and the branch would have £1,000 less in cash than Horizon expected it to have.*

In other words, although a transaction could be inserted, it would immediately become apparent that this had been done and ultimately it would not benefit any member of staff to behave in this way.'

1366 In other words, the DEA countermeasure comes into play. There is also another DEA safeguard in place. Branch transaction data have been captured in POLSAP and its predecessors[59], which have been controlled by PO rather than Fujitsu. A subsequent change in the branch database would have led to a discrepancy between that database and the POL MIS, which could be detected later (RDS/MID).

56 See ARC/SOL/ARC/0006, section 2.1

57 DES/APP/HLD/0020, section 5.6

58 In paragraph 20.2 of his witness statement

59 I refer to these collectively as the 'POL MIS'.

FUJ00083961

1367 As Mr Parker makes clear, there is no incentive for anyone to inject transactions unless they are required to correct a branch's accounts.

1368 As the number of users with any given access rights increases, so the risk of unauthorised or inappropriate usage increases. In my opinion, the number of Privileged Users and SSC users who can create a BT seems high. This may indicate an opportunity to improve security by reducing the numbers of permissions granted.

1369 I reached the following agreement with Mr Coyne: '*Usage of the above tools and facilities* [referring to Issue 10] *should be auditable. However, the maintenance of logs would be dependent upon retention periods and size.*'

1370 Each transaction is associated with a particular user, so it is clear in the records who was responsible for its creation. All transactions are recorded in the audit store, so SPMs could find out if any had been performed without their consent or knowledge.

1371 Privileged usage has been logged since July 2015. Prior to then, only log-ons and log-offs were recorded.

1372 In summary, permission to use the facilities described under Issue 10 was controlled. Usage of those permissions and the resulting actions was also recorded. However, the controls in place have not been perfect. External audits[60] have identified room for improvement.

Issue 10 (ii) Implement fixes in Horizon that had the potential to affect transaction data or data in branch accounts

1373 PO controls changes in reference data themselves, outside the scope of Horizon. The data is delivered into Horizon for distribution to branches (see section 6.8.5 above). I have seen evidence from July 2017 in a document relating to the PO Operations Board that '*After a number of service impacting incidents over the last 3 months, we have now aligned that all Reference Data changes go through the appropriate change process.*'

1374 It could prove misleading to draw broad conclusions from this single piece of evidence, but I deduce from the statement that:

- Horizon service was disrupted in the first half of 2017 by problems arising from changes in reference data.
- Not all such changes were being managed by an effective change process.
- Action has been taken to rectify this weakness.

1375 The corrective action would have rendered Horizon more robust by means of the QCC countermeasure.

1376 I have seen no other evidence about permission controls or records relating to PO's management of changes in reference data.

1377 Fujitsu implements fixes in software by installing new versions of specific components in the live system. Reference data is updated by distributing new datasets as described in section 6.8.5.

60 See Appendix J.x

61 Described in SVM/SDM/PRO/1184

FUJ00083961

1378 The Managed Service Change (MSC) process[61] is used to control changes in the live system, new features as well as fixes. MSC forms part of the overall Manage Change process. It is administered via the TfS and Peak tools available on the SSC website and described in section 6.8.3 above. The tools create detailed records of the actions performed, i.e. the changes in Horizon.

1379 The MSC process document defines roles and accountability for managing changes. The right to use the tools on the SSC site is limited to authorised users.

1380 MSC succeeded the Operational Change Process (OCP)[62].

Issue 10 (iii) Rebuild branch transaction data

1381 This operation will be restricted to database administrators or, previously, to engineers for when they were upgrading branch hardware and rebuilding the local database from the central servers.

11.7 Horizon Issue 12

1382 **Issue 12**: If the Defendant and/or Fujitsu did have such ability, how often was that used, if at all?

1383 BTs have only been used once, on 11 March 2010 – and this was not in a branch operated by a Claimant. Torstein Godeseth provides full details in his Witness Statement dated 27 September 2018. He also states that, so far as he is aware, Fujitsu has never used its privileged access to edit or delete transaction data.

1384 PO changes reference data on a daily basis.

1385 A combined total of 36,000 MSC and OCP records have been created, which amounts to 5-10 per business day on average. There have also been some 20,000 release notes, which equates to approximately 5 releases per working day – including reference data.

1386 I have seen no evidence to confirm how often the following capabilities were used:

- Transaction changes
  - PO Global Users
  - SSC
    - Legacy
    - Privileged Users
- Database rebuilds

11.8 Horizon Issue 13

1387 **Issue 13**: To what extent did use of any such facility have the potential to affect the reliability of branches' accounting positions?

1388 I shall interpret 'extent' for Issues 13 as I have interpreted extent for Horizon Issue 1.

61 Described in SVM/SDM/PRO/1184

62 See CS/PRD/067

FUJ00083961

1389 I shall ask the question with reference to the accounts for a specific claimant in a specific month. If a claimant were to assert that the use of any such facility had introduced a discrepancy into his accounts in any specific month, what is the probability of that account being correct?

1390 The answer to this question does not depend strongly on the size of the discrepancy, but I shall assume for definiteness that the question refers to a significant discrepancy, greater than £300. As described in section 7.6, this discrepancy is large enough that at least some SPMs, faced with such a discrepancy, would notice it and query its cause. I am not considering 'micro discrepancies' which any SPM might not notice or attribute to human error.

1391 I shall address Horizon issue 13 with respect to issues 10(i), (ii) and (iii)

1392 To answer the question about 'any such facility' for issue 10(i) (insert, inject, edit or delete transaction data or data in branch accounts), it is simplest to answer it for each facility I have identified above, and then to combine the answers. I shall start with Balancing Transactions (BTs).

1393 I have cited evidence that only one BT has been made in the lifetime of Horizon. In section 8.5, I calculated that the number of monthly branch accounts in the history of Horizon, across the whole PO network, is in the region of 3 million.

1394 Therefore, in the absence of further evidence about why BTs might occur, the probability of a BT affecting a claimant's branch accounts in a given month is one part in 3 million. The chances of it being incorrect - introducing a discrepancy in the branch accounts - are even smaller. For that to have happened, the BT would need to have been made in error - and furthermore, the error would need to be not detected or corrected. In my opinion, the probability for all these to happening to one branch in one month may be of the order of one in a hundred million.

1395 (It might be argued that since claimants' branches were smaller than the average PO branch by about a factor 3, the probability of a BT affecting a claimant's branch would also be smaller by a further factor 3. But this would depend on the cause of the BT, and whether that cause was size-dependent, so I shall not assume any dependence on branch size).

1396 I next consider changes to reference data. As described above, these were made frequently by PO. Some of these changes were in error, and they may in principle have introduced discrepancies in branch accounts. However, any such issues were recorded in KELs, and I have already analysed these in section 8 - including analysing several KELs specifically about errors in reference data, which had no effect on branch accounts. My conclusions in section 8 about reference data errors were included in my conclusions about all software errors. The probability of their having any effect on a claimant's branch accounts in any given month was extremely small. For details see that section.

1397 I next consider changes to transaction data made by global users, or by SSC. As above, I have seen no evidence about the number of such changes which have been made. However, by the same argument as above, for any one such change, the probability of it affecting one claimant's branch accounts in one month are approximately

FUJ00083961

one in three million. The chances of it doing so erroneously, and not being subsequently corrected, are even smaller - perhaps one in a hundred million.

1398 It follows that there would need to be a very large number of changes to transaction data made by SSC or global users, with a large proportion of these being in error, to give even a 10% chance of introducing a significant discrepancy in a claimant's branch accounts for one month. The number required is more than a million such changes.

1399 In conclusion on Horizon Issue 13 applied to changes under issue 10(i) (insert, inject, edit or delete transaction data or data in branch accounts): for these changes to have any significant chance of affecting a claimant's branch accounts in a given month, there would need to be a huge number of them - probably of the order of 1 million. In my opinion, this is not possible,

1400 I next consider the facilities under issue 10(ii) (implement fixes in Horizon that had the potential to affect transaction data or data in branch accounts). Fixes in Horizon only had the potential to introduce discrepancies in branch accounts if those fixes introduced bugs. I have already addressed bugs in section 8. There I found that in order to give a significant chance of introducing a discrepancy in one claimant's branch accounts in one month, there would need to be of the order of 64,000 separate bugs which affected branch accounts on the same scale as the Suspense Account bug. I do not believe it possible for there to be that number of bugs which affect branch accounts, whether introduced by fixes or otherwise. Therefore, the facilities under issue 10(ii) are not capable of introducing discrepancies in claimant's branch accounts, with any significant probability.

1401 I next consider the facilities under issue 10(iii) (rebuilding branch transaction data). I have not seen evidence as to how many times this has been done. Consider one occasion of rebuilding branch transaction data. As before, the probability of this happening to one claimant's branch in a specific month is one part in 3 million. The probability of it happening erroneously, and not being corrected, are something smaller than that.

1402 Therefore, the probability of the rebuilding of transaction data introducing a discrepancy in a claimant's branch accounts in a specific month is extremely small - unless the rebuilding of transaction data has been done on a very large number of occasions. As before, more than 10,000 occasions would be required. I have seen no evidence that transaction data were rebuilt on this number of occasions and consider it extremely unlikely. Therefore, the rebuilding of transaction data would not introduce discrepancies into a claimant's branch accounts in any given month, with any significant probability.

11.9 My opinions

1403 PO has access to information derived from their back-end systems, relating particularly to their clients, which is not available to SPMs. PO and Fujitsu also have knowledge of their systems and processes as well as skilled and experienced resources, which enable them to investigate anomalies reported by SPMs.

1404 As I explained in paragraph 1262 above, SPMs and their staff are uniquely placed to investigate discrepancies, because they are the only people who have first-hand knowledge of what happens in their branches.

1405 Thus, some anomalies can only be resolved by SPMs, some only by PO and/or Fujitsu, and a third category can only be resolved by cooperation between the parties.

FUJ00083961

1406 To summarise my opinions expressed earlier in this section of the document about PO's ability to intervene on behalf of the SPM:

a) Both PO and Fujitsu have been able to affect branch accounts remotely as well as locally.

b) There is no reason why they would do so unnecessarily.

c) Controls are in place to limit the use of such facilities.

d) All such actions are recorded so that they can be traced.

1407 As described in the previous sub-section, in my opinion the probability of any of these facilities introducing a discrepancy in any claimant's branch accounts, in any month, is extremely small.

11.10 Mr Coyne's opinions

1408 Mr Coyne's report enables me to identify new areas of agreement between us, since our joint statement of 04 September 2018, as follows:

1) The reports available to PO should have enabled them '*to identify the occurrence of alleged shortfalls in the Horizon system (of those that could be identified), and they were underpinned by formal processes which would provide further information in relation to the underlying cause of a given issue, and the best way to resolve the same. In addition, Post Office should have been able to obtain any additional information it required via Fujitsu or the Subpostmasters themselves.*' [in paragraph 8.10 of his report]

2) '*Subpostmasters had access to a much smaller pool of information. This is in line with what I would expect to see given that Subpostmasters are the users of the Horizon system, and therefore would not typically be given access to anything beyond what was necessary for them to carry out their 'business as usual' activities.*' [paragraph 8.11]

3) Fujitsu has been able to access counters within a branch so that they can provide support and maintenance. [paragraph 9.4]

4) Fujitsu has been able to access transaction data recorded by Horizon both within a branch and stored centrally within BRDB. [paragraph 9.7]

1409 Alongside point (4) agreed above, Mr Coyne also notes (in paragraph 9.6) that '*it has not yet been identified that transaction data was altered at the counter.*' In other words, despite Fujitsu being able to access the counters, Mr Coyne has seen no evidence of any impact on branch accounts.

1410 Mr Coyne criticises PO's controls on reference data in paragraph 4.21 of his report:

'*Despite the criticality of the integrity of Reference Data, a document from July 2017 suggests that changes to Reference Data were not subject to any appropriate change control process. The document [17] reports; "… we have now aligned that all Reference Data changes go through the appropriate change process". This is consistent with the position that prior to July 2017 Reference Data could be changed without any formal consideration as to what the impact might be.*'

1411 The full quoted sentence reads: '*After a number of service impacting incidents over the last 3 months, we have now aligned that all Reference Data changes go through the appropriate change process.*'

1412 In my opinion, Mr Coyne's concerns are overstated:

FUJ00083961

- The sentence does not say that changes were not subject to any appropriate change control process. Nor does it say that, before July 2017, reference data were changed with no fomal impact assessment. Instead it only implies that, over the previous 3 months, changes in reference data may have impacted service.
- The sentence also confirms that corrective had been taken.

1413 I discuss the issue fully in paragraphs 1368 and 1369 above.

1414 In paragraph 9.10 of his report, Mr Coyne cites the Witness Statement of Richard Roll to confirm that Fujitsu employees could and did remotely access branch accounts to perform modifications. SSC could only insert transactions to modify accounts and not edit or delete the transactions. Insertions were strictly controlled, as explained in section 11.6 above.

1415 From paragraph 9.15 onwards, Mr Coyne addresses the concept of Global Branches. Mr Godeseth explains[63] that '*Global branches (which relate to Horizon Online only) are physical branches with Horizon terminals which are used solely for support purposes.*'

1416 Mr Godeseth explains[64] that '*Mr Coyne's allegation at paragraph 9.18 of his expert report that "An instance of a global branch would allow Fujitsu to create global users and to input transactions within core Horizon systems as though they had been entered from a physical branch" is not correct. To enter a transaction for a physical branch would mean that Fujitsu would have to be physically present at that branch.*'

1417 Mr Coyne argues (in his paragraph 9.18) that '*An instance of a global branch would allow Fujitsu to create global users and to input transactions within core Horizon systems as though they had been entered from a physical branch.*' this is not possible.

1418 Are the allegations based on Richard Roll borne out by a higher proportion of branches losses during his tenure 2001-04? RW to address, potentially in s8.

63 See Witness Statement – Torstein Godeseth – dated 27 September 2018, paragraph 51

64 See Witness Statement – Torstein Godeseth – dated 16 November 2018, paragraph 32