FUJ00093048

RMGA HNG-X Counter Application Review

A review of the integrity of the HNG-X application relating capturing of sales and financial transactions at the Counter

Circulation: Alan D'Alvarez, HNG-X Programme Manager
Graham Allen, HNG-X Applications Engineering Manager
Andy Thomas, HNG-X Counter Architect
David Johns, Lead HNG-X Architect
Maz Kostuch, Director of Service Delivery and Projects, PSD
David Leask, Managing Customer Solution Architect, PSD

Authors: Paul Roberts, Applications Architect, AS
Stuart Rye, Managing Consultant, PSD

1 Introduction

1.1 Terms of Reference

Following the occurrence of a transaction being duplicated on the HNG-X Branch database, Stuart Rye and Paul Roberts were asked to review the Counter Application architecture and design and ensure that it fully supports the need to protect the integrity of financial transactions.

The scope is focused on the integrity of the Counter Application in relation to financial transactions captured at the Counter. It does not extend to data migration or other transient issues caused by the switch from Horizon to HNG-X.

1.2 Background

The objective of 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. A key component of HNG-X is the Counter Application. In contrast to the Horizon Counter Application, the HNG-X version retains operational data (e.g. Reference Data) and business logic, but transactional information is stored directly in the Data Centre. The Counter side of the new applications is based principally on Java technology. The Counter hardware is reused from Horizon with the initial migration deploying the new application on the existing Windows NT 4.0 operating system. Where it has been feasible to reuse components from the existing Horizon Counter, these have been carried forward into HNG-X

It is currently in pilot with 12 branches. On 27th January 2010, the Data Reconciliation Service (DRS) process detected an error in a banking transaction. Subsequent investigations revealed that the Branch database had two transactions with different JSN[1]s but the same SSN[2] for a specific Counter on that day but the 3rd Party banking system only had one transaction. The clerk did not know that a duplicate transaction had been created.

An analysis of the database has revealed one other occurrence, again at Derby but on a different day and involving a different clerk. This had not been detected by the DRS as it did not contain a banking component and there is no other business reconciliation which might have spotted it. I disagree. My understanding is that the other example was also Banking and also picked up by DRS. Email from Claire Drake on 28/1 refers. First incident was a Deposit and the second was a Withdrawal.

The net effect would be that the Post Office and the Branch records would not match. Where this happens, the Post Office investigates the branch and Postmaster, with a view to retraining or even uncovering fraud. It would seriously undermine Post Office credibility and possibly historic cases if it

[1] JSN – Journal Sequence Number – Used for audit purposes and must be unique without gaps per each Counter.
[2] SSN – Session Sequence Number – Unique within the session and can be duplicated after a period of time

FUJ00093048

could be shown that a discrepancy could be caused by a system error rather than postmaster/clerk action. Most importantly, the central database as the system of record would be called into question.

The new Counter Application records all financial transaction data in the Data Centre. The PC in the Branch runs the Counter Application (with other sub system components). It is critical that the central database properly records the actions at the Counter and reliably reflects the actions of the clerk.

There was a suggestion that the running of performance monitoring software in Derby created the conditions which triggered the failure. Hasn't that now been discounted? The development team concluded the failure was caused by a bug and a resolution has been identified which includes further measures to remove the possibility of this occurring in the future.

The Counter development team were able to recreate the Derby error by heavily loading the processor in a terminal and double keying the "settlement" action.

Further assurance is required that Fujitsu has got to the root cause and there are no other potential issues with the integrity of the HNG-X as a system of record.

1.3 Approach to assurance

Paul Roberts and Stuart Rye joined the HNGX team on Thursday 4th February. We met with the following people:

- David Johns – Lead architect – 4 hours
- Andy Thomas – Counter architect – 6 hours
- Steve Porter – Counter development team – specific questions
- Alan Holmes – Auditing – 15 minutes

The following documents were provided:

- HNG-X_Solution_Architecture_Overview_2009-07-28.ppt
- ARCSOLARC0001_v4.1_Draft Overall Solution Architecture.doc
- ARCAPPARC0003 Counter Architecture.doc
- ARCAPPARC0009 Counter Business Architecture.doc
- REQCUSSTG0002 Branch Exception Handling Strategy.doc
- DESAPPIFS0012 BAL Service Interface Specification.doc

Our approach was to:

- Understand the business process – from the Counter to back end / 3rd parties
- Understand the business risks and controls
- Understand the system architecture, risks and controls
- Explore the specific failure in Derby and its resolution
- Explore any other areas of risk where the Post Office and Branch could end up with a variance due to a system error.

The review was conducted with the assumption that the HNG-X system was a technology upgrade and not a business process reengineering project. Business requirements and controls in HNG-X should be the same as the current Horizon platform. It was declared that the Counter Application has been re-written based on Business specifications provided by the Post Office since the Horizon Counter Application had been developed using a third party product.

2 Findings

2.1 General

- The HNG-X system does represent a technology migration, with no material changes in business function. However, the development team involved in the Counter Application was intentionally drawn from people with no familiarity of the current platform, partly because the skills base is different and partly due to the fact the Horizon Counter Application had been developed using a

FUJ00093048

third party product which would not feature in the HNG-X solution.. Was this really intentional? There was some overlap (eg Walter and me!)

- There are no material changes to business function. There were some in specific areas, but in general this was the aim. However, the adoption of a real time transactional system has increased traffic to and from the remote database and therefore the process time. Time at the counter is critical to the Post Office so a decision was made to find efficiencies where possible. In particular, the settlement process for cash only payment was improved, with the removal of an extra button press previously required to complete settlement. POL also took the opportunity to make simple process changes, but in general things were pretty much the same.

2.2 Business Process (see Appendices)

- The business process is essentially the capture and settlement of a complex retail shopping transaction. Customers can, inter alia, purchase stamps, top-up prepayment keys (I don't think we do that any more), send Moneygrams and withdraw cash using their Bank card. They can settle their basket with a range of methods – cash, vouchers, cards.
- The clerk logs the customer's transactions in a basket (just like online shopping). These could be a mix of all types of transactions. Each basket has an identifier called an SSN assigned when the basket is opened. These are specific to the Counter and basket but not unique as the SSN is reset to 0 after 999,999. A basket cannot be opened whilst another is open.
- Once complete, payment is taken for relevant items and/or cash paid out for a withdrawal and the basket settled i.e. recorded on the central database and flushed from the Counter. Each basket at the database is assigned a unique and auditable reference called a JSN. Not quite correct. A JSN is assigned to any auditable data that is recorded at the data centre. Baskets represent 90%+ of these. SSNs only relate to baskets so SSNs may have gaps, but JSNs shouldn't.
- Card (banking) transactions are pre-authorised with the issuing bank when the card is presented. Authorisation is a pre-requisite to the basket being settled. A unique identifier is generated for the banking transaction at the point of the authorisation request. If authorisation fails, the basket remains unsettled and the customer must abandon the withdrawal or settle another way. Banking withdrawal and settlement are different things and this is confusing the two. (but it probably doesn't matter).
- Accumulated banking transactions are settled as a batch process at the end of the day, based on the baskets recorded as settled on the database. This is the Data Reconciliation Service. DRS reconciles, Banking, Plastic and ETU not just banking. Note that DRS doesn't do the settlement – that is done by POL. What DS does is reconcile the Counter and Bank views of what transactions have occurred.

2.3 Settlement

The clerk indicates the transaction is settled in two ways: "Fast Cash" and "Settle". There is a button for each. Each triggers a process resulting in the basket being logged on the database.

- Fast Cash
    - "Fast Cash" in that the clerk has only one key to press. They can do this when the transaction is settled by cash only.
    - This places an entry for the cash amount in the basket which sets the total to nil and the basket settles.
- Settle
    - The clerk uses "settle" to bring up a menu of settlement options where the customer is paying by means other than cash only. The clerk records the relevant settlement methods and amounts. Each amount is logged in the basket. Once the basket reaches "nil", the basket settles.
    - If the basket is already "nil", the "settle" button goes straight to settlement.

When the basket settles, the Counter confirms with the Branch database that the basket is set to "nil" i.e. all payment received / cash paid out. On receipt of confirmation from the database that the basket has been properly captured and committed, the process of flushing the basket is started. In

FUJ00093048

some cases a receipt is printed and cached based on the basket details before it is flushed, in others the receipt is cached (usually for a banking only transaction, where the receipt is printed at the point of authorisation and no other items are present). Once these procedures are complete, the basket is flushed.

2.4 Reversal Settlement

This occurs when a customer returns due to some error being made. For example, I bought 2nd class stamps but I needed 1st class stamps or the customer was bought and/or sold the wrong item.

There are two different reversals that can be performed:

- Customer returns with receipt within a specific time window
    - This initiates a reversal settlement process and is referred to as an "existing reversal". The process inserts a "reserve" transaction on the database before settling the reversal. The clerk is guided through the reversal process and there should be no opportunity for a double settlement to occur. By "double settlement", I assume what is meant is that the same transaction can be reversed more than once. ERs use the standard basket technology and so also have the same controls as normal settlement.
- Customer returns without receipt or with receipt outside the specific time window
    - This will follow the same process as for a normal settlement and is known as a "new reversal". The financial amounts will be a different polarity. This process is protected by the same controls as for a normal settlement.

2.5 Controls

- Control over the data capture process (for example, disabling input from peripherals such as the touch screen, bar code, scales etc.) is supposed to be maintained and integrity provided by ensuring the clerk or peripherals cannot initiate out of sequence events. (This control failed in Derby.)
- Banking transactions have an end of day control run by the DRS which reconciles the intra-day authorisations with the end of day settlement. This will alert duplicate or missing entries. Banking transactions have a unique identifier. These cannot be duplicated (except by bugs!).
- All other financially relevant 3rd party interfaces (in particular Moneygram) have business logic built into the process to prevent duplication or a loss of a transaction. With Moneygram, the clerk is forced to acknowledge the point of financial settlement in the process and there is a specific input button presented after this point to commit the funds. There is no opportunity for this to be repeated or initiated by any other means.
- The JSN has a uniqueness constraint as it is used for auditing purposes. It is also "dense" meaning that there should be no gaps for a given Counter. It is valid for a Counter to send transactions with a duplicate JSN, for example during a retry process. In this scenario, if the transaction is already stored in the Branch database, it will fail on the uniqueness constraint which will force a comparison of the transaction to be stored and the transaction already stored. If they are the same, the to-be stored transaction is discarded. A discrepancy in the transactions will force an error to be raised and the Counter will be logged off.
- The SSN is not unique and is repeated after 1 million Baskets per Counter.
- If a basket is not successfully stored in the Branch database, the Counter performs a recovery procedure which will involve retrying the basket. If this fails, then the clerk must perform a rollback which clears the basket of non-recoverable trasnactions.

2.6 Spotting the duplicate Derby basket

- It was the end of day reconciliation of the banking transaction in the DRS that alerted Fujitsu to a problem. Transaction settlement requests were presented for two transactions with the same unique transaction identifier and the second one was rejected. This revealed the presence on the HNG-X database of two duplicate, absolutely identical baskets with the same SSN and contents but different JSNs. Essentially, the Counter had submitted the same basket twice to the Branch database which had logged each as unique, with different assigned JSNs. Pre-authorisation only happened once as this took place before the basket was settled.

FUJ00093048

2.7 Cause of the duplicate basket

- In the old Horizon system, the clerk was required to press “Fast Cash” and then “Settle” to settle. In the new system, they only need to press “Fast Cash”.
- In Derby, the clerks concerned adopted the old procedure and pressed “fast cash” (which recorded the cash entry, set the basket total to “nil” and initiated the settlement process) and then “settle”. The settle button should not have been enabled but it was. As the basket was now at “nil”, the basket was settled again immediately.
- There is nothing particular about Derby. The performance monitoring was not running.

2.8 The Resolution

- The development team have implemented an immediate resolution within the Counter Application and are considering possible solutions involving the Branch database (see Appendices for process flows).
- The Counter Application:
    - It was discovered that the peripheral input lock was released before the basket was flushed and the “Settle” button enabled. This allowed the clerk to activate the “Settle” button and because the basket was already set to nil, it was sent to the Branch database and caused the duplicate. The ability to press the “Settle” button was due to timing differences. The clerk was very efficient and following the old Horizon process. Windows was servicing another process. The Counter Application code has now been modified to flush the basket before the peripheral input lock has been released. Isn’t this the same as the next bullet?
    - The development team also found another scenario when this could occur and as well as correcting this they have implemented additional code that highlights any further scenarios that have not been discovered. The additional code ensures that the Counter Application is in “Busy Wait” state at the point the transaction is sent to the Branch database. The code does this irrespective of whether the state is already correct. If the state is not correct it produces an alert message on the Counter Application. This has only been adopted for the Development and Test environments.
    - A “flag” has been added to the basket that prevents duplicate settlement initiation. This is a belts and braces approach since the previous code changes should not require this functionality.
- The database solutions:

  The solutions being considered are:

    - Check for duplicate SSN on receipt of a basket. If this is done real time, it will place a demanding overhead on the database.
    - Check for duplicate SSNs as part of an audit process during the overnight period.

2.9 Stock check

- Stock checks will highlight differences between the Post Office records and Branch records. Some post offices undertake daily checks, others weekly, others rarely. They are all supposed to check their Cash Daily (though not necessarily to compare it against the system figures). They must check Cash and stamps whenever they balance which is at least once per month. I agree that other stock may be rarely checked.

3 Conclusions

Overall, the actions taken to redress the Derby issue are appropriate. We believe the Counter Application fully supports the need to protect the integrity of financial transactions.

Specific conclusions are detailed below.

FUJ00093048

3.1 Business controls for banking transactions will alert duplicate baskets containing banking transactions because of reconciliation checks in the DRS but there has not been a control for baskets with no banking transaction – this is now being addressed.

3.2 Duplicate banking transactions with different identifiers cannot be created.

3.3 The duplicate could have happened anywhere, not just Derby; (performance monitoring not enabled).

3.4 Baskets cannot be "lost". The JSN and associated checks provide confirmation that all transactions sent from the Counter are stored in the Branch database. Recovery procedures built into the Counter Application and Business Processes ensure transactions are not lost.

3.5 Duplicate baskets cannot be created in the Counter Application and cannot be submitted to the database more than once following the resolution taken by the development team. This was less of a risk in the old technology but an increased risk with the adoption of the centralised financial transaction approach and the real-time transmission to the database from the Counter Application.

3.6 The business control at the Counter recognises that dual settlement risk is inherent by having two buttons that can initiate settlement. The business requirement is that the buttons operate on an exclusive basis – i.e. the use of one disables the other. The implementation in the HNG-X Counter Application failed and enabled both buttons to be pressed. They are not exclusive. Once you're on the Settlement menu you should be able to invoke Fast cash again. Also once in settlement you can add an item and so need to be able to Settle again, so the buttons is not an issue. It is the enablement of them during settlement that is the fault.

3.7 The Counter Application design did not have sufficient controls to prevent duplicate basket transmission (regardless of the control intended to limit initiation of settlement by different buttons).

3.8 The peripheral input lock release occurred too soon in the process, and allowed "Settle" button to be pressed.

3.9 The Counter Application fix adopts a belt and braces approach and is very strong.

3.10 The design of the basket flag control is a good one

3.11 The basket submission flag avoids the timing trap that the "Settle" button lock fell into and is enabled at the correct point in the process, just before the transmission to the Branch database.

3.12 A basket cannot be flushed until confirmation is received that it has been properly committed and logged at the database.

3.13 The design for the basket flag allows for the flag to be correctly removed if a card transaction fails, allowing an alternative settlement method to be adopted. I didn't think the flag was set when a card authorisation was carried out – only when actually settling a basket.

3.14 An SSN is associated with a basket. Per Counter, per day it can be used as check for duplication (within a population of 1 million)

3.15 Given the strength of the implemented Counter Application controls a real-time check at the database for duplicate SSN receipt should not be necessary and would potentially decrease performance of the system

3.16 An end of day procedure to detect duplicate SSN is acceptable. We are not planning to do this permanently. Should we?

FUJ00093048

3.17 The “new reversal” process follows the settlement process but the transaction amounts have a different polarity. The “existing reversal” process is a guided process and the clerk does not have the opportunity to create duplicate transactions. Again not sure I follow this.

3.18 Other interactions do not create risk, e.g. post code look up, or have appropriate business and technical controls in place.

3.19 It is unsure what testing has been performed to prove peripherals are disabled and enabled at the correct point in the Business Process.

3.20 Stock check is not a reliable method for catching issues due to the erratic nature of the stock check occurring. Cash checking might be, but cash is usually slightly wrong due to human error.

3.21 The Counter Application is quirky when ordering quantities of stamps. For example, it is possible for five stamps to be recorded as one stamp if the clerk does not interact with the Counter Application correctly.

3.22 There is no practical control to stop under and overcharging for baskets caused by clerk error. For example, the clerk my give the wrong number or denomination of stamps.

3.23 There seems to be no other risk of baskets undercharging.

3.24 There is a solution to detect incorrect busy-wait states in the development and test environment which would be useful, with enhancement, in the live environment as an indicator of failing code.

4 Recommendations

4.1 The “busy wait” check should be implemented in the Production environment but with an Error message being written the Windows Event log in preference to the Alert message. This Error message would be transmitted to the central database following the business as usual processes for detecting errors on the Counter PC and appropriate procedures put in place

4.2 Do not implement the real time duplicate SSN check. Create an audit report for SSN duplicates with the same JSN on same day. This could be run on the Disaster Recovery database if performance is an issue. I think they mean BRSS rather than Standby database. However we need to get that working first!

4.3 Review peripheral disablement and enablement testing and where appropriate undertake more testing for specific use cases to check peripherals are in the correct state.

4.4 Check for duplicate SSN before raising discrepancies with a Postmaster.

4.5 Consider advising the Post Office of the benefit of more effective stock control as an indicator of clerk errors or Fraud.

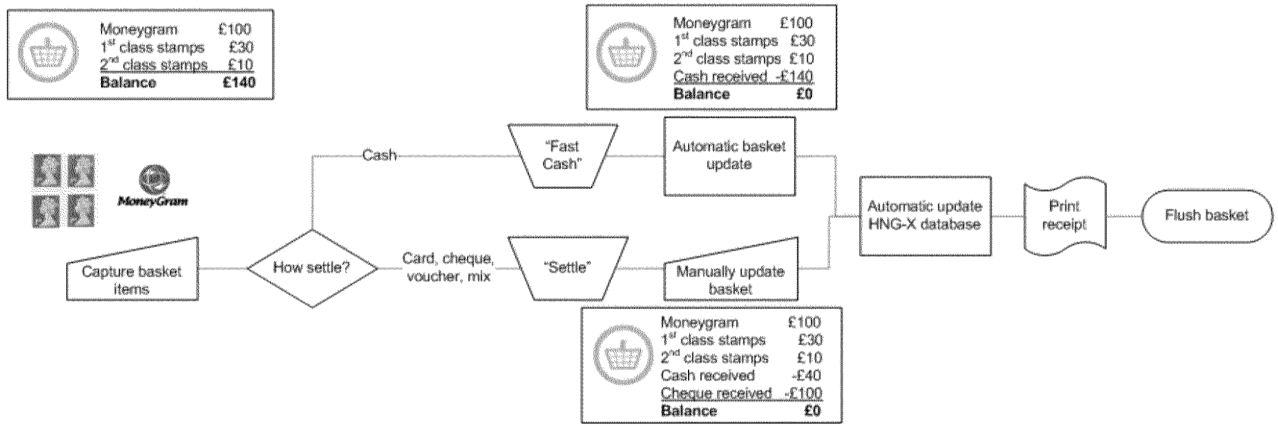
4.6 Review and strengthen negative testing, if appropriate. The recent problems reflect the asynchronous nature of the new application and traditional or historic test cases may not reflect this.

FUJ00093048

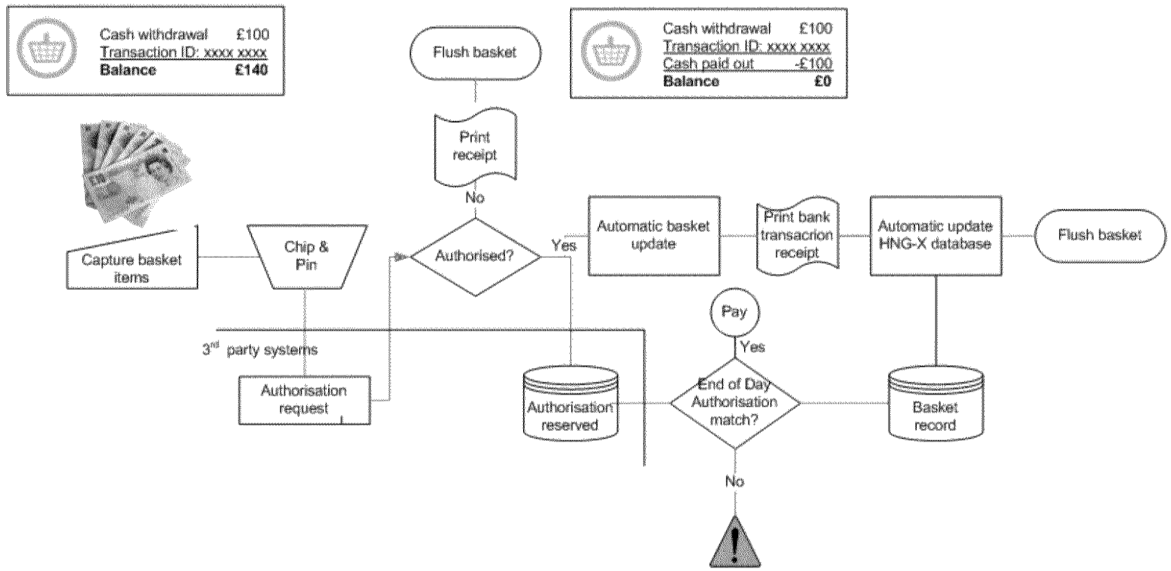
Appendices High Level Business Flows

2nd diagram is missing the need to invoke Fast cash / Settle. Also the basket Balance in the 2md diagram should be £100.. This is the area where the report authors confuse Cash Withdrawal with Card Payment. It is important to correct this or remove the diagrams.

1. Business Process -
No banking transaction



2. Business Process -
Banking transaction



FUJ00093048

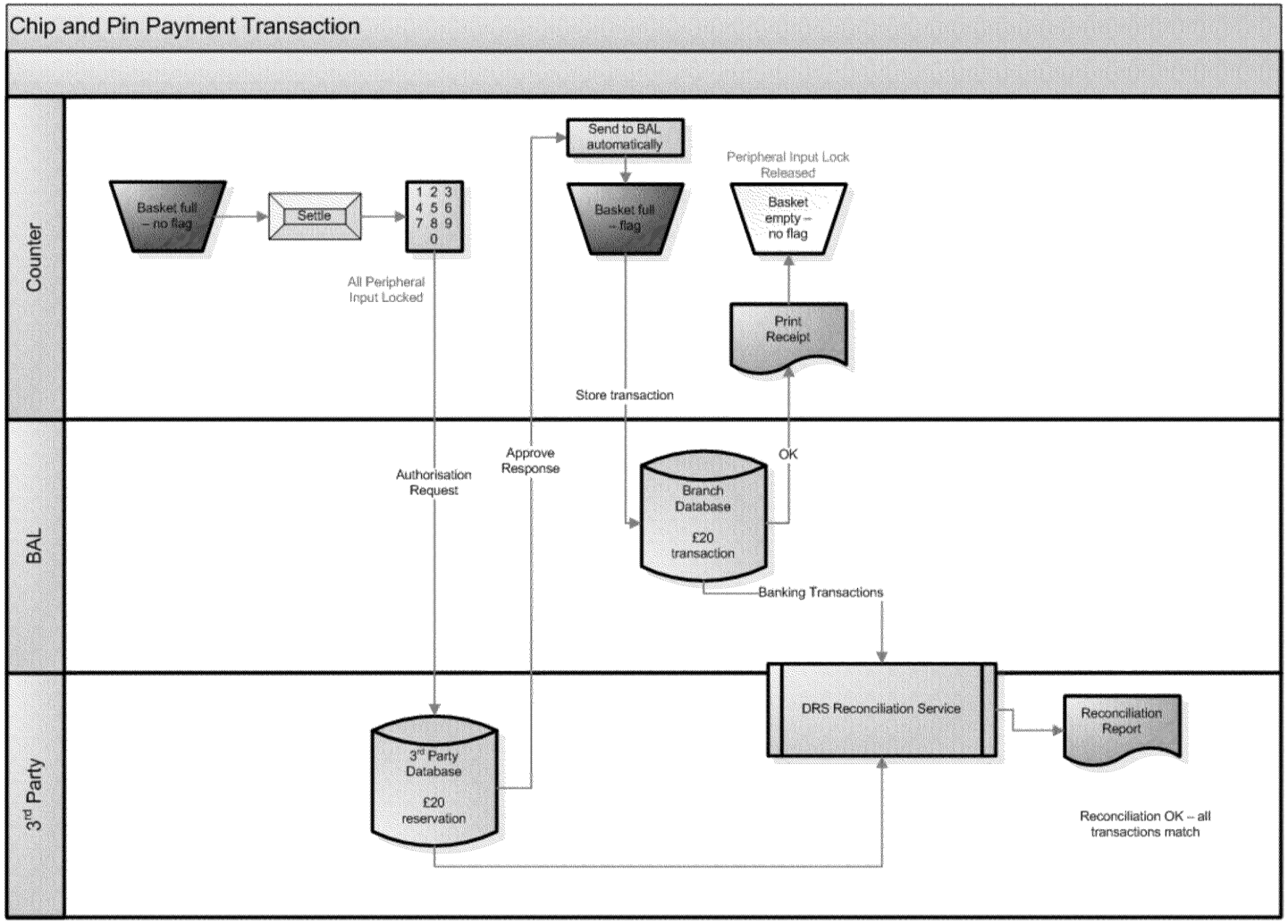
–System Flows - Banking Transaction - OK

I don't think this is correct (as with previous diagram) It looks more like settlement of a basket via cad payment than a Banking transaction.

Also in the case of card payments, the settlement model is different and is based on the settlement transaction. The payment file for Streamline is created from the settlement transaction loaded into DRS.

I am not sure whether the DRS check for a duplicate transaction would have prevented the system from putting 2 transactions in the payment file. Hence in this case the duplicate transaction could potentially have been passed to Streamline – but if so I assume that it would have been rejected there due to 2 payment requests for the same authorisation?

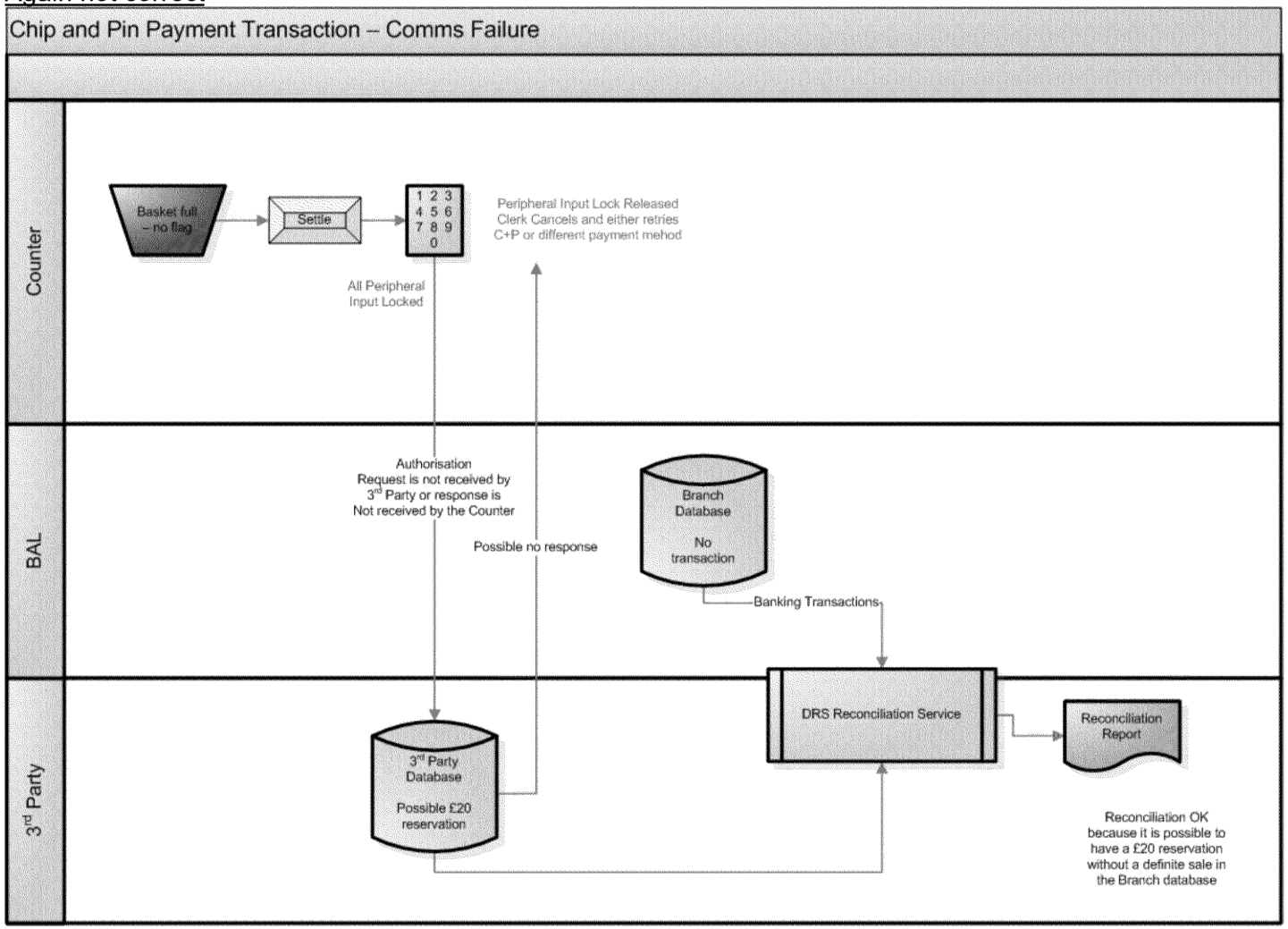
FUJ00093048



FUJ00093048

System Flows - Banking Transaction – Comms Failure

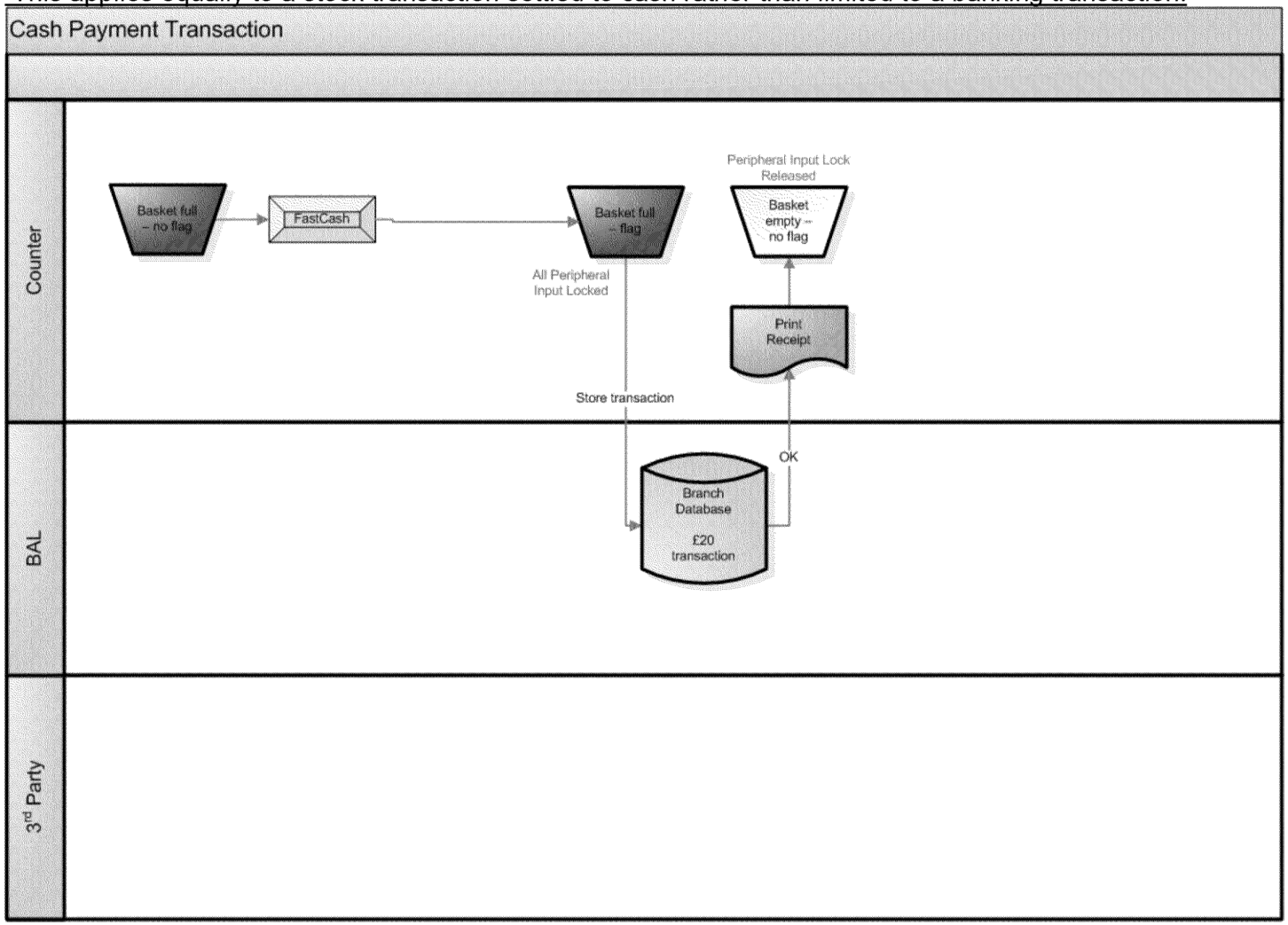
Again not correct



FUJ00093048

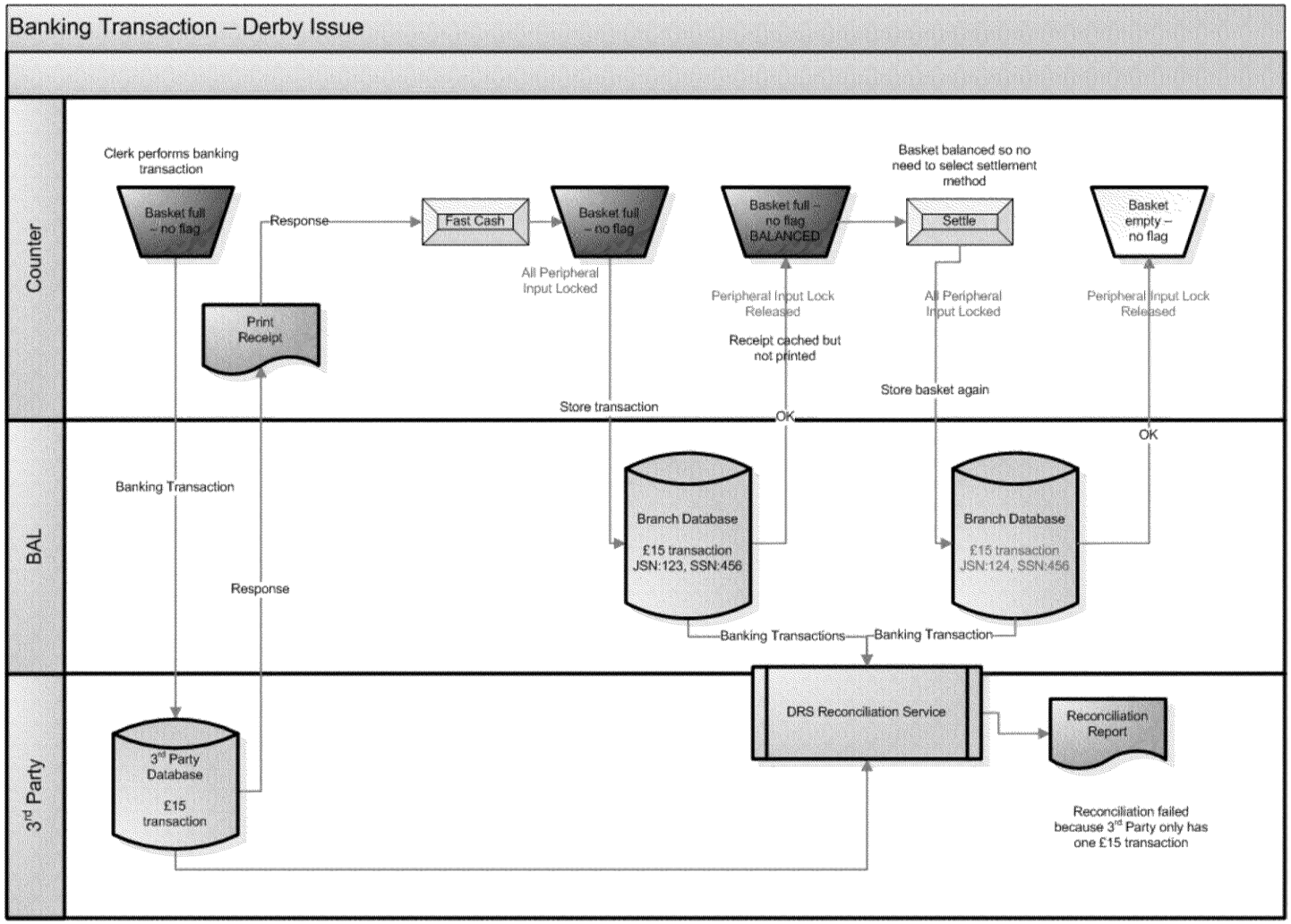
System Flows - Banking Transaction – Cash Transaction

This applies equally to a stock transaction settled to cash rather than limited to a banking transaction.



FUJ00093048

System Flows – Derby Issue



FUJ00093048

System Flows – Derby Fix