

iBonus System Core Technology White Paper

Moving from Second Generation to Third Generation

Introduction

The emergence of smart card system used for loyalty and prepaid industry implies that the member card data is to be distributed among all smart cards instead of being kept at Server. Payment terminal can read and write values directly to and from the card. Real time queries between payment terminal and server are less relied on. Therefore the requirement of the processing power of the server and the capacity of the communication channel is reduced. The deployment cost of a smart card system become much lower. However it doesn't mean that developing a smartcard system can therefore be easy. The deployment of a large scale smart card loyalty and payment system, without the help of a stable network environment and powerful server, requires extra cares to ensure its stability, security and performance.

Being aware of the requirement addressed above, Lucky Technology Limited has spent 2 years redeveloping iBonus System. iBonus Loyalty and Prepaid System Version 3 has radically simplified its deployment process. System engineer does not need to attend the server in order to connect a new Terminal to server. iBonus Terminal works perfectly with just a virtual IP. No port forward and tricky network configuration is needed in order to connection iBonus Server. That speeds up the deployment process especially in large scale deployment.

Being the core engine of a loyalty system, iBonus System requires high system integrity and performance. Journal database, transaction queuing and record duplication prevention are to ensure that every transaction is collected properly. The chance of losing or duplicating transaction is virtually zero even the system is operated under an unstable network. Transaction record sequencing ensures all member records are updated with the newest transaction record even the records are not sent back to server in order. Fraud proof feature is added to ensure the security during card replacement and repair. Journal database engine is engaged in iBonus Terminal to protect the data from sudden power shutdown and makes a ten fold increase in data processing power. All the above components are designed and developed with complete software development life cycle in order to ensure their robustness and correctness of the system.

Software Development Methodology

iBonus development team knows what is required in a membership system. The design of the transaction type has balanced the need of security and ease of use. For example, normal transaction, like Card Purchase, can be done by simply entering the amount and present the member card; for the Undo Purchase transaction, however, it has to be done with staff card confirmation. This is to prevent staff from using that less frequently used function accidentally.

All the transaction types and core components are designed by Unified Modeling Language (UML) to ensure that the whole software development team and the designers have the clear picture about the underlying logic before coding. Also it will enhance the visibility and correctness during customization.

iBonus Terminal and iBonus Server are treated as core components of the iBonus Membership System. Especially there is strict requirement on the correctness and

robustness of generic transactions, advanced card handling and the data communication algorithms. Before its release, it has been passed through a series of functional tests and capacity tests. Each function is tested to comply with the specification. With such prudent and our experience in the field, we believe that iBonus System can let all our end users and developers to deploy their loyalty system with confidence.

Rapid System Deployment

Unattended Server Design

When deploying iBonus Terminal at each outlet, it only needs to enter the Server IP and Server port at each Terminal. iBonus Server automatically register the Terminal when the Terminal connects to the server the first time. Deregistration is also to be done at Terminal side. The setup process can be done even there is no one attending the iBonus Server. It saves up the man power during deployment.

Dynamic Virtual IP Operation

Every TCP/IP connections between iBonus Terminal and iBonus Server is initiated at Terminal side. With the simplicity of the protocol, Terminals can work perfectly with just a dynamic virtual IP. Port forwarding at the Terminal side is not necessary. It saves the effort on configuring the tricky router settings.

Generic Transaction Set

iBonus Terminal sets the standard of transaction types of prepaid and loyalty system. It provides generic transaction types for performing calculations of stored values, such as, prepaid, bonus point and accumulated spending balance. The behavior of all transactions types are consistent with the combinations of card status, system setting and system status, like, decryption keys, expiry settings, decimal mode, bonus rate, blacklist, card sector usage and card value limits. For example, transaction limit is defined in Staff Card. It limits the maximum prepaid transaction amount of all money transactions. Blacklisted card is detected and expiry date in Last Visit mode is extended in all operations. Without the need of redeveloping the complex and sophisticated algorithm behind, system integrators can make use of them and quickly deploy a loyalty system.

Table of Generic Transaction Types

Name	Description	Member Card Calculations	Function Available at
Card Purchase	Purchasing goods by card stored value while adding bonus values	- Prepaid Balance + Bonus Points + Accumulated Purchase	Prepaid Mode
Cash Purchase	Purchasing goods by cash while adding bonus point.	+ Bonus Points + Accumulated Purchase	Bonus Mode
Reload	Adding stored value	+ Prepaid Balance	Prepaid Mode

Redeem	Exchanging bonus points for gifts	- Bonus Points	Bonus Mode
Quick Reload	Do Reload and Card Purchase at the same time.	+ Reload amount to Prepaid Balance Then - Prepaid Balance + Bonus Points + Accumulated Purchase	Prepaid Mode
Undo Card Purchase	Undoing a Card Purchase operation. (Staff Card reconfirmation is required.)	+ Prepaid Balance - Bonus Points - Accumulated Purchase	Prepaid Mode
Undo Reload	Undoing a Reload operation (Staff Card reconfirmation is required.)	- Prepaid Balance	Prepaid Mode
Undo Cash Purchase	Undoing a Cash Purchase operation. (Staff Card reconfirmation is required.)	- Bonus Points - Accumulated Purchase	Bonus Mode
Undo Redeem	Undoing a Redeem operation. (Staff Card reconfirmation is required.)	+ Bonus Points	Bonus Mode
Carry Forward	Setting a new member card to certain values	In prepaid mode, set Bonus Points and Prepaid Balance In Bonus mode, set only Bonus Points	Prepaid Mode and Bonus Mode

Full Point of Sales (POS) Functions Coverage

iBonus System unifies the smart card system with the POS system. All the transaction types in iBonus Terminal are accessible by POS through RS-232 serial port. Coming with a ready made Windows based API, all the generic transaction set and advanced card handling functions can be accessed with ease. In addition, with the help of read member card and read staff card functions, POS can command the most suitable transaction based on the POS side member profile.

High Scalability

Support for Multiple Database Servers

iBonus Server fully supports both Microsoft Access Database and Microsoft SQL Server 2005 through ADO .NET technology. Administrator can make selection to be based on the scale of deployment and budget. For small deployment with limit number of member or the system is under evaluation, Microsoft Access database, which is built in Windows XP, can be used for cost saving. For a large deployment with sophisticated reporting or clusters for database, Microsoft SQL Server 2005 can be used.

Batch Process

iBonus Server writes each record into its journal transaction queue at its local database, and sends back notify signal to Terminal. The server side transaction queue is optimized for data collection. The data stored in the queue will later on be processed by the batch. The replies for the request are stored in indexed tables. When the Terminals are querying the Server, the server can get the reply by simple table look up. This design radically reduces the demand of server processing power and therefore allowing a large amount of Terminals to connect to and large amount of member to use the system.

High Data Integrity

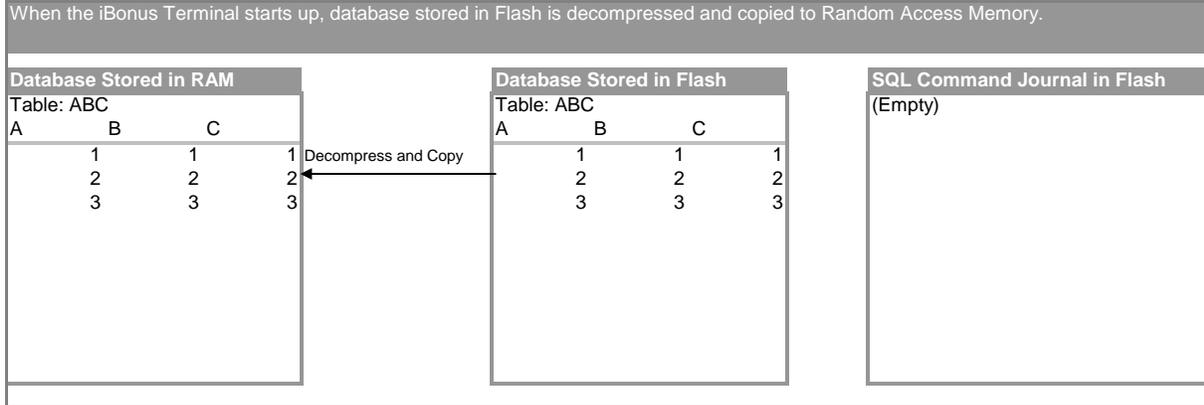
Care has been taken to ensure the data integrity of iBonus System. The following technologies have been used to ensure all data of iBonus Terminal to be sent to and stored at server correctly.

Journal Database

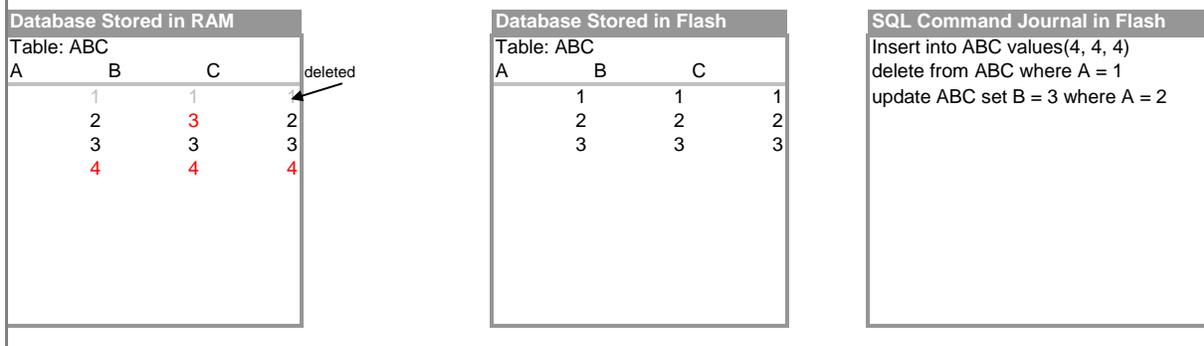
iBonus Terminal temporary stores transactions. In case of network congestion, iBonus Terminal has transaction buffer which stores up to 5000 records. As the database in iBonus is used as buffer, the data integrity of the database is taken into great care.

Journal database is implemented at iBonus Terminal. During Terminal start-up the flash memory copy of the database is decompressed to RAM. During the transaction, each updating SQL command is recorded at the journal file after successful execution on the RAM copy of the database. The database is committed either periodically or when the journal file reaches a certain size. During write back, the RAM copy is written back to the Flash memory. In case of unexpected power shutdown, all the commands in the journal file are re-executed against the flash memory copy of database.

Here is the illustration on how journal database works.

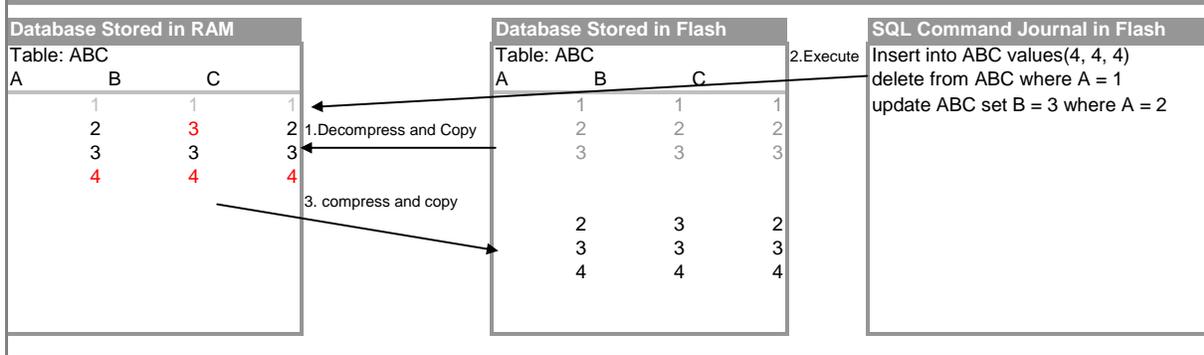


Database updating and searching is 10 times faster in RAM than in Flash memory. All update commands are executed on the RAM copy. The command is also written in the Journal. In this case, for example, the following commands are executed:
Insert into ABC values(4, 4, 4)
delete from ABC where A = 1
update ABC set B = 3 where A = 2



In case of power shutdown the following steps are taken during the next start up.

1. Decompress the Flash copy of data base from the Flash memory
2. Execute all SQL commands in Flash Journal
3. Vacuum clean, compress and write back the database into Flash

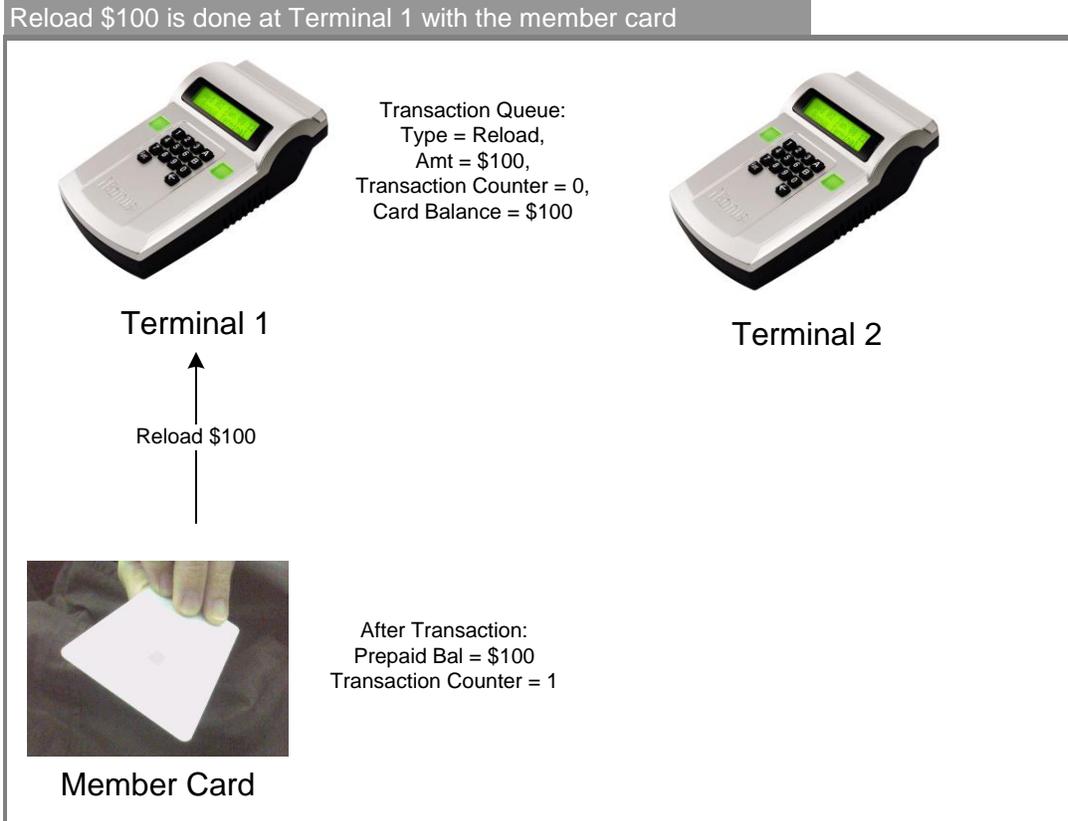


Transaction Sequencing

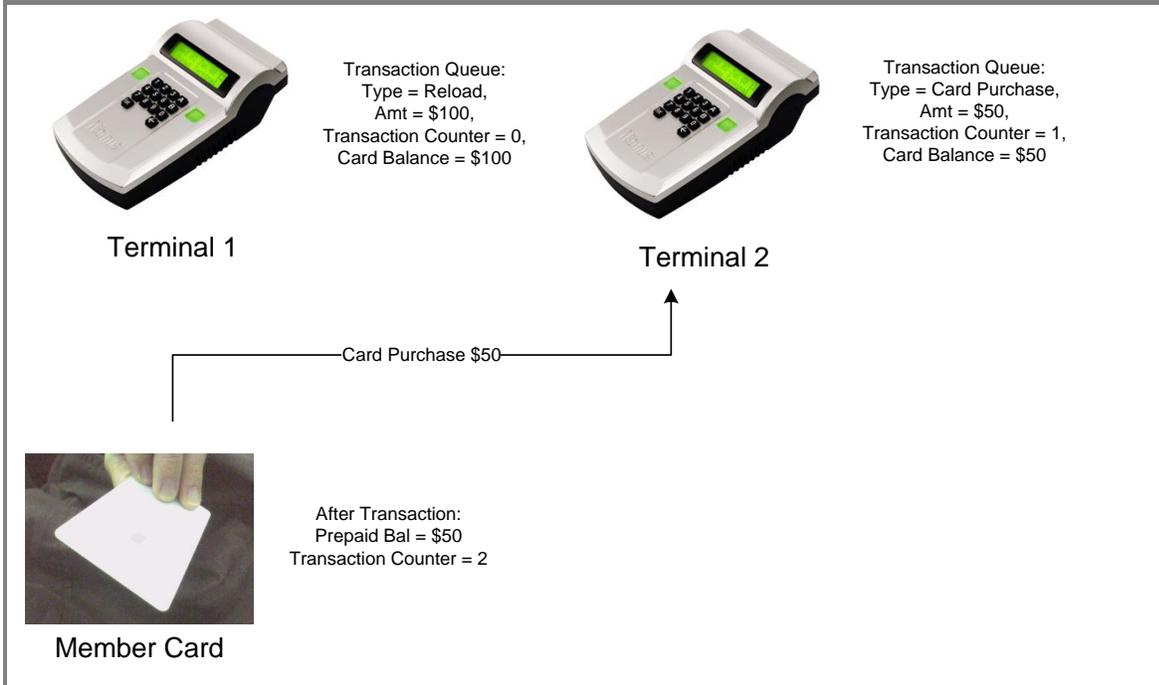
It is possible that transactions that are done in different terminals for the same card are not reaching the server chronologically due to temporary network instability. In the need of updating member's latest card value correctly, each member card has a transaction counter that is incremented after each transaction. By attaching the value onto each transaction, server can serialize the transaction in order to update the member record correctly.

There is a side benefit for using transaction counter. In case of record loss, it is still possible to tell what transaction has been done based on the sequence of the transaction record of each card.

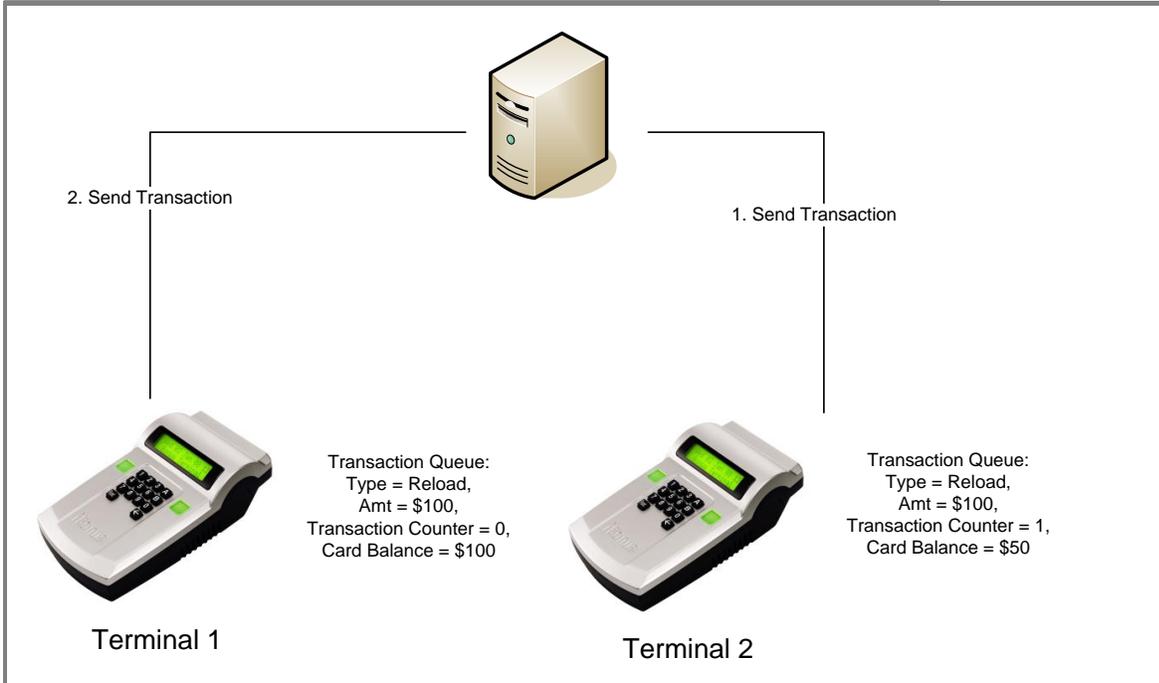
Here is an illustration on how transaction sequencing works at iBonus System:



After a short while Card Purchase \$100 is done at Terminal 2 with the same member card.



However Terminal 2 send transaction back to the Server earlier than did Terminal 1



Due to some network delay, Terminal 2 has sent back the transaction to the iBonus Server before Terminal 1 has. The transaction at the server is not collected in order. In that case, iBonus Server updates the Member table only when the transaction counter of the transaction being processed is newer than the one existing in the Member table.

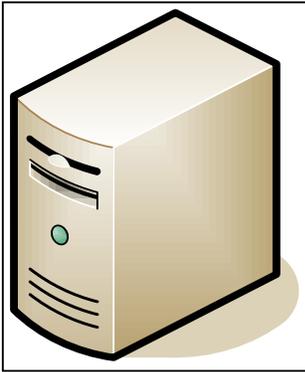


Table: Member		
Card SN	Prepaid Balance	Transaction Counter
123	\$50	1

Transaction Queue					
Card SN	Transaction Counter	Prepaid Balance	Type	Amount	
123	1	\$50	Card Purch	\$50	
123	0	\$100	Reload	\$100	

The information of this row is not updated to member table as the transaction counter is smaller than the one in member table.

Record Duplication Detection

iBonus Server ensures that no record is recorded twice in the database. This is done by recording the last record of that terminal in the Server. If the acknowledgement of the transfer of the last record is missed, the Terminal may send back the same record again. By comparing the record being transferred with the last record, Server can tell whether there is duplication and reject the record accordingly.

Fraud Proof Design for Advanced Card Control Functions

iBonus System realizes the feature could have been challenge for designers who develop a smart card based prepaid system with branch level advanced card control functions which ensures the safety of use. It has advanced algorithms for ensuring security during card expiry date renewal, replacement and repair.

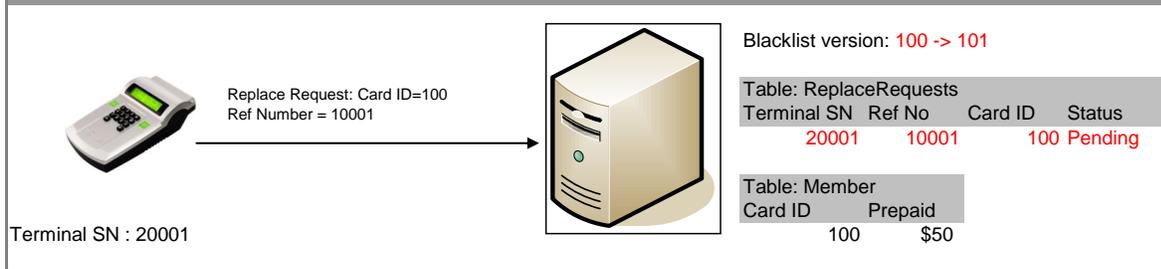
The anti-fraud design ensures:

- the newly replaced card will reflect the pre-blacklist salvage value of the lost card even it is used at any other Terminals after reported loss,
- only one replaced card is issued even the user intentionally report card loss in more than one terminal within a short time,
- the repaired card must reflect the latest card information even the card has done a transaction at another branch a short time before needing to repair,
- the blacklist wont be grown to an unlimited size by allowing only the card having expiry date to be blacklisted, only the valid card are blacklisted, and
- the card being renewed must not be a blacklisted card.

Here is an illustration on how card Replace procedure is carried out.

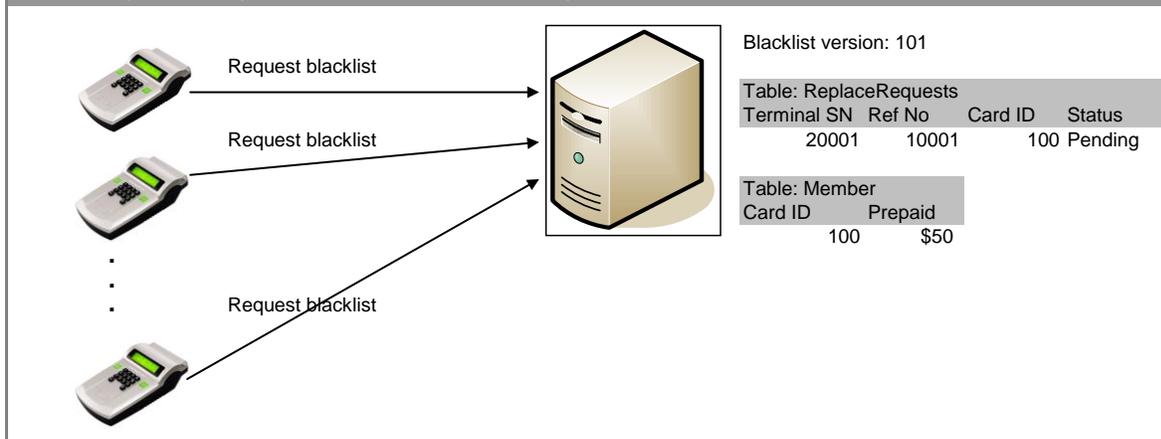
Step 1

An iBonus Terminal sends a request to the server to report card lost for Card with ID 100. Reference number is for future communication use with server. After having succeed in looking up the member table, server creates a pending record in the table ReplaceRequests



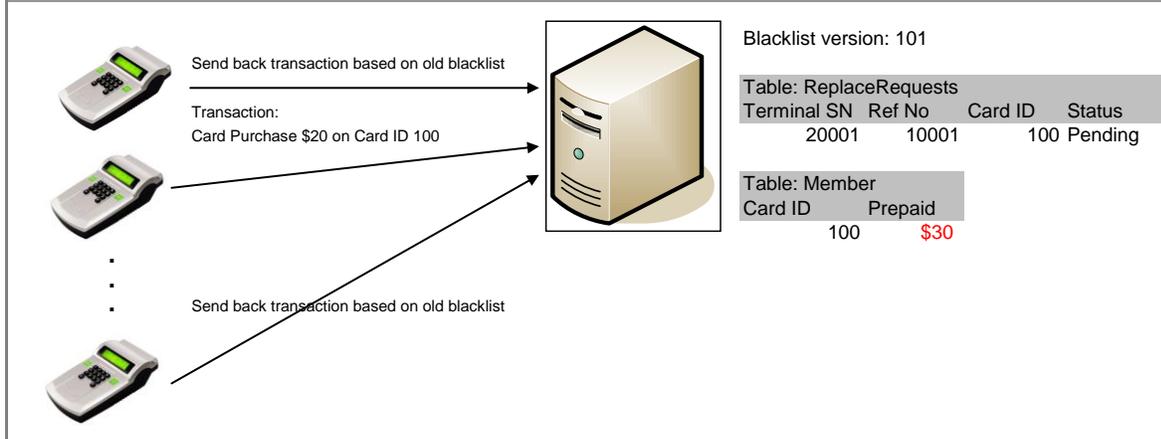
Step 2

All iBonus Terminal periodically request blacklist from iBonus Server. By default the period is 6 hours. That means the exploitation of the card is expected to be prevented with in 6 hours since the report of loss.



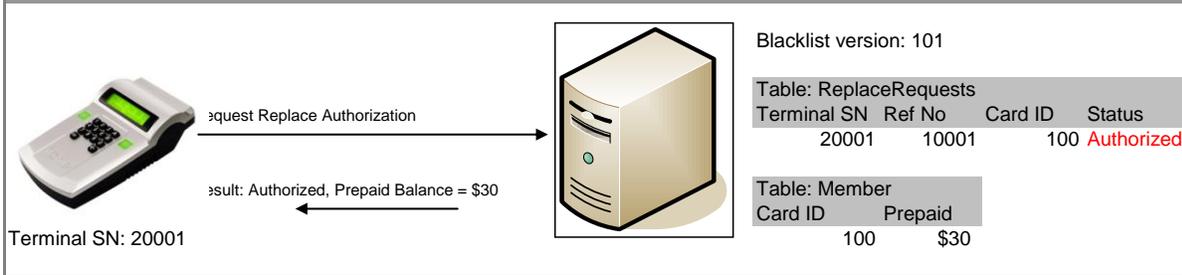
Step 3

All iBonus Terminal will then send back transaction based on the old blacklist. In this case the card is used at some Terminals using the old blacklist. Server has taken that into account and recorded it.



Step 4

After collecting and processing the transaction from all Terminals, Server will authorize the ReplaceRequests. The request originating Terminal periodically requests for replace authorization. (Default period 6 hours) The Terminal will be able to receive the most update information of the lost card. After that, staff can load a brand new card with the lost card information for replacement.



Reporting

With all the components and process ensuring the correctness of data, valuable information is hidden in it. It needs some mechanisms to mine the gold from the heaps. iBonus complete the equation by adding the reporting which caters the different roles of the system users.

By using Microsoft ADO .NET technology, iBonus Server works seamlessly with iBonus Report. With a set of program written in ASP .NET, administrator can define the role of each user. The users select the available report based on their role. Through web browser, employees and members can login their account and see the report that they are authorized to view.

In a scenario that iBonus Membership System is deployed as a multiple company loyalty system. The iBonus Report can be configured as follows:

- Marketing managers can have access to the reports on the sales aspect. They can check out the sales figure filtered by time, member info, branches, terminals and transaction type, and grouped by different period of time, card ID, staff ID and branch code.
- Accounting managers beside can have access to the sales information; they can also have access to the clearing report for cash clearance of multi-shop loyalty.
- System Administrator can see the report for Terminal Status, pending card renew, repair and replace status, blacklist status.
- Branch Manager can check out the cash collected by their own branch or staff for each day, they can also look up the repair and replace status of their own branch.
- Member can check out their latest card values, transaction history, card repair, renew and replace status.