UBIN PHASE 2

Technical Documentation

Overview

This report covers the general topics in Ubin Phase 2 including project background, requirements and future considerations.

Table of Contents

1 Executive Summary

Ubin Phase 2 is a collaborative project led by the Monetary Authority of Singapore (MAS) and The Association of Banks in Singapore (ABS).

It is managed and delivered by Accenture, with participation from 11 financial institutions. The 13-week project explores the use of Distributed Ledger Technology (DLT) for specific Real Time Gross Settlement (RTGS) functionalities. Particularly, it focuses on the feasibility of decentralising Liquidity Saving Mechanisms (LSM), while maintaining privacy of banking transactions.

Leveraging the capabilities of the Accenture Liquid Studio and its Liquid Delivery Methodology with Microsoft Azure as the cloud platform, three prototypes were developed by three workstreams on three different DLT platforms: Corda, Hyperledger Fabric and Quorum. The prototypes successfully demonstrate several points. Firstly, that key functions of a RTGS system such as fund transfer, queueing mechanism and gridlock resolution can be achieved through different techniques and solution designs. Secondly, decentralising the key functions of a RTGS system may not only mitigate the inherent risks of a centralised system, such as single point of failure, but may also affirm the promised benefits of DLT, for example cryptographic security and immutability.

Given that privacy is paramount in an interbank payment system, this project validates that privacy of RTGS transactions may be ensured by all workstreams with their distinct methods. Specifically, Corda with its Unspent Transaction Output (UTXO) model and confidential identities, Hyperledger Fabric leveraging its Channels design, and Quorum using Constellation and zero knowledge proofs (ZKP).

Other observations and findings from this project include the scalability and resilience of the three designs. Significantly, this project concludes that all three workstream designs have successfully demonstrated the feasibility of removing a central infrastructure operator in a DLT-based RTGS system. Therefore, with the feasibility of DLT in a RTGS system, the role of MAS as an infrastructure operator in facilitating interbank payments needs to be re-evaluated.

Ubin Phase 2 not only successfully demonstrates that RTGS functions may be decentralised without comprising privacy, but also marks the success and significance of an industry-wide collaboration in laying the foundation for future innovation.

2 Objectives and Scope

2.1 Background

2.1.1 Project Ubin

In late 2016, in line with the vision for Singapore to become a Smart Financial Centre, the Monetary Authority of Singapore (MAS) commenced a collaborative project with 11 leading financial institutions and 5 technology providers. It explores the use of Distributed Ledger Technology (DLT) for clearing and settlement of payments and securities. The goal of this endeavor, known as Project Ubin, was for MAS and the financial industry to gain a better understanding of DLT and the feasibility of developing more resilient and efficient alternatives to today's financial market operations and systems.

In Phase 1, a proof of concept was conducted on Ethereum, testing the feasibility of using a central-bank-issued digital currency (SGD equivalent) for interbank payments. Ubin Phase 2,

managed and developed by Accenture, assesses the potential implications of deploying DLT for specific RTGS functionalities by focusing on Liquidity Saving Mechanism (LSM). It is also an objective of Phase 2 to understand how real-time gross settlement privacy can be ensured using DLT.

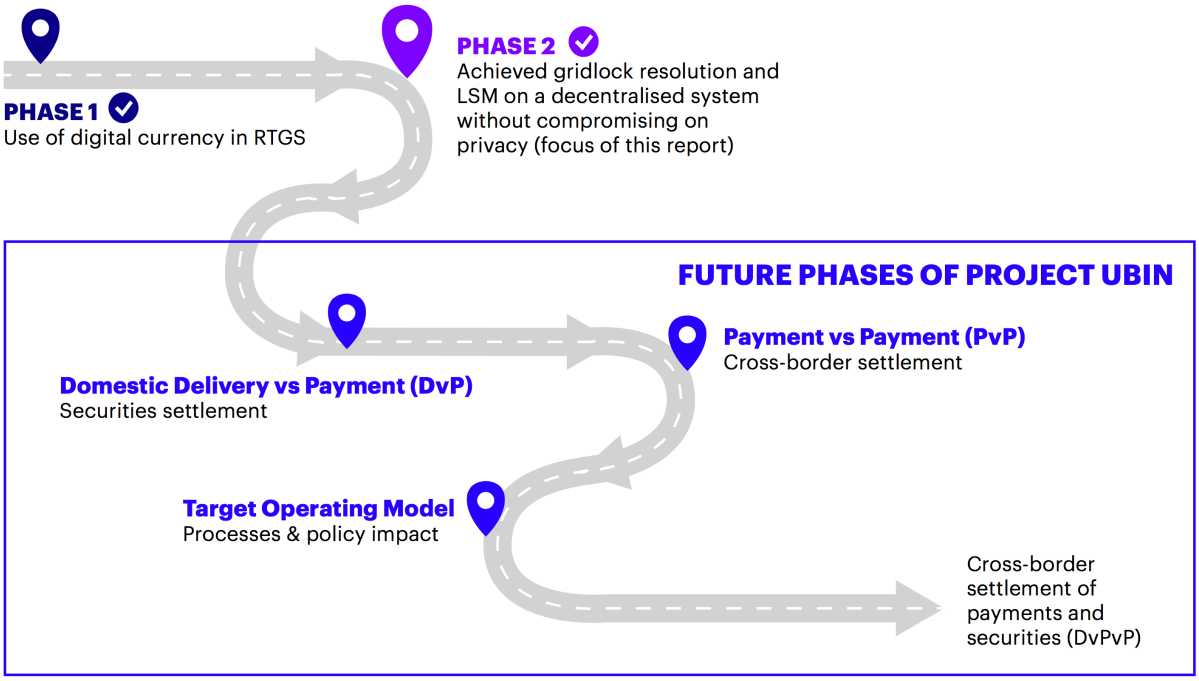


Figure 1: Overall Journey of Project Ubin

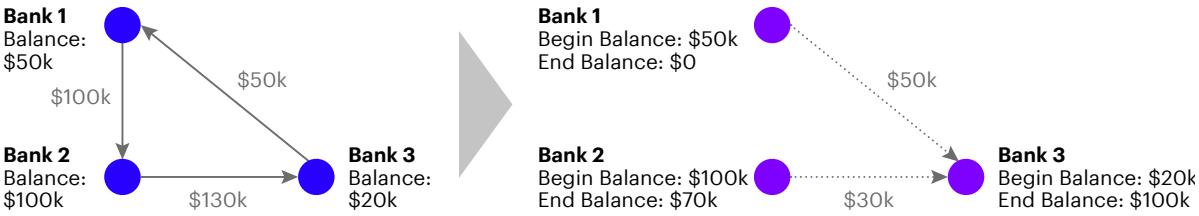


Figure 2: Illustration of a Simple Gridlock Scenario

2.1.2 Real-Time Gross Settlement System

Real Time Gross Settlement (RTGS) systems are typically used for high value transactions requiring immediate settlement. MAS operates a RTGS system called the MAS Electronic Payment System (MEPS+). MEPS+ plays an integral role in the functioning of Singapore's financial market. It processes about 25,000 transactions a day, with each transaction valued up to SGD 1 billion, and a total daily transaction value of up to SGD 70 billion.

Most RTGS systems around the world are operated on a centralised infrastructure, which is subject to risks such as a single point of failure. The high-value nature of RTGS transactions also demand the system to process transactions seamlessly to reduce intraday liquidity

gridlocks in the financial market. A key function in RTGS systems is the ability to resolve payment gridlock through a Liquidity Saving Mechanism (LSM).

Figure 2 illustrates a simple gridlock scenario where three payment instructions, $50,000, $100,000, and $130,000 are unable to settle (in a sequential manner) as each sender bank does not have sufficient funds. The purpose of a LSM is to resolve the gridlock scenarios when these transactions are executed on a net basis. LSM is traditionally performed by a centralised system as the algorithms for LSM typically requires a consolidated, single view of all payment instructions in the system.

Today, MAS plays both the roles of an infrastructure operator and an overseer of the MEPS+. The latter sections in the report refer to MAS as having both roles but mostly as the infrastructure operator.

2.1.3 Distributed Ledger Technology (DLT)

At its most basic, Distributed Ledger Technology (DLT) is a general purpose data technology that allows different actors to share access to the same data with confidence. Participating actors can trust that the data has not been tampered with and can control access to the data. DLT was arguably born out of 1920s/30s' cryptography and has been widely popularised by the introduction of the Bitcoin in 2009. Today, the technology has evolved to solve different business problems.

DLT is of particular interest to the financial sector and traditional centralised clearing and settlement systems for several reasons. Firstly, it has the potential to increase the reliability and traceability of information stored in a decentralised network. This decentralised processing and storing of information potentially mitigates the single point of failure present in the current centralised system.

Moreover, strict rules are embedded within DLT on how ledgers are governed and recorded. Multiple parties must come to a consensus on the legitimacy of a transaction before it can be recorded in the distributed ledger. This helps to reduce or eliminate the need to reconcile transactions, since the data has been agreed and attested to by all or multiple parties.

Lastly, where privacy and confidentiality are paramount in the financial sectors, additional protocols and enhancement of the technology will better enable the decentralised information to be private or restricted. Figure 3 is an illustration of how DLT may potentially disintermediate the central operator in a centralised RTGS system as the settlement processes and ledger are distributed among the different banks in the network.

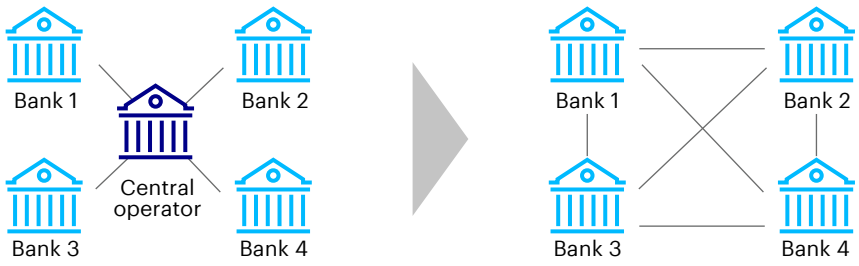
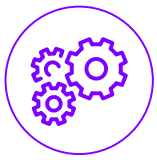


Figure 3: Transition from a RTGS with central operator to a decentralised RTGS

2.2 Scope and Approach

The objective of Ubin Phase 2 was to develop three prototypes with specific RTGS functionalities. Each prototype is developed on a different DLT platform: Corda, Hyperledger Fabric and Quorum running on a common cloud platform – Microsoft Azure. A key functionality showcased in Ubin Phase 2 is the ability to execute Liquidity Saving Mechanisms (LSM) without compromising privacy in a decentralised network. The prototypes are developed to address the following six key criteria:

Digitalisation of Payments
Central Bank Digital Currency (CBDC) with real-time gross settlement capabilities



Decentralised Processing
Distributed and resilient infrastructure with no single point of failure

Payment Queue Handling
Uniform queueing system with prioritisation, holding and cancellation facilities

Privacy of Transactions
Only relevant parties will have visibility to transaction details

Settlement Finality
Final and irrevocable settlement of payment instructions with deterministic finality

Liquidity Optimisation
Implement netting and gridlock resolution algorithms to maximise liquidity efficiency

Ubin Phase 2 started with design workshops in July 2017. There were three workstreams with each representing a platform: Corda workstream, Hyperledger Fabric workstream and Quorum workstream. The workstreams were tasked to design a solution on a common set of functionalities, with the goal of conducting an objective assessment on the three platforms. It is not the intent of this proof of concept to draw a quantitative comparison among the three workstreams.

Based on the different workstream designs, Accenture's rapid prototyping team from Singapore Liquid Studio developed and delivered three prototypes over the course of three Agile sprints. A common user interface for these prototypes was also developed in line with the goal of an objective evaluation across the three platforms.

The functional scope of this proof of concept is categorised into 11 epics, addressing the specific key criteria as described in Section 2.2. This section describes the key functions of applying DLT to the specific RTGS functionalities.

A key functional requirement in Ubin Phase 2 is to execute fund transfer in a DLT network that enables decentralisation and digitalisation of payments. This includes a queueing mechanism that allows payment instructions to be prioritised for processing, as well as determining whether payment instructions can be immediately settled, deferred for future processing or cancelled. In a straightforward scenario, eligible payment instructions are settled as debits from the sender and credits to the receiver. However, complex situations may arise when payment instructions are ineligible due to insufficient funds and these are transferred to the waiting queue, where they can be reprioritised. These complex situations may result in a gridlock situation, whereby outgoing payments cannot be fulfilled unless an incoming payment is received or a gridlock resolution is triggered.

All the transactions performed within the DLT require strict privacy, and are validated to ensure the transaction is final and legitimate.

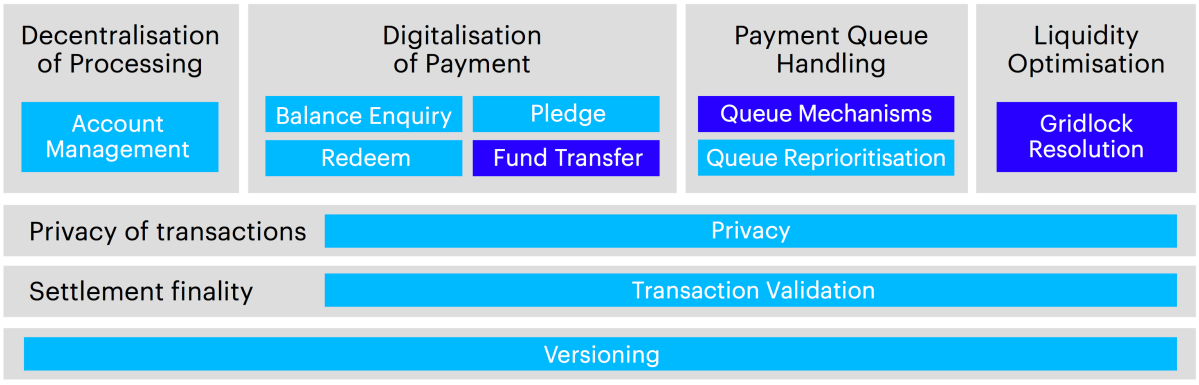


Figure 4: Functional scope of Project Ubin Phase 2

2.3 Requirements

The requirements and objectives of Ubin Phase 2 can be divided into 11 high-level Epics and further detailed into 43 user stories for each platform. These are listed in 2.4.1 Epics and User Stories.

2.3.1 High-level Functional Requirements

Project Scope (Functional)	Epic
Final and irrevocable settlement of payment instructions on Real-Time Gross Settlement (RTGS) basis	Ø Transaction Validity Ø Transfer Fund
Queuing mechanism for payment instructions that cannot be settled due to participant limits or insufficient funds	Ø Queue Mechanism
Queue re-prioritization, holding and cancellation facilities for payment instructions	Ø Queue Reprioritization

RTGS gridlock resolution algorithms that can be automatically and manually activated according to pre-defined parameters, depending on the options specified by the operator	Ø Gridlock Resolution
Validation of payment instructions sent by ABS and participants	Ø Transaction Validity Ø Transfer Fund
Net settlement mechanism to provide final and irrevocable settlement of clearing positions from the ACH, such as cheque clearing and e-GIRO figures	Ø Gridlock Resolution
Account set-up and maintenance (e.g. suspended, closed accounts)	Ø Manage Accounts

2.3.2 High-Level Technical Requirements

Project Scope (Non-Functional)	Epic
Privacy of transactions (Transaction details available only to selected parties)	Ø Privacy
Consensus mechanism which allows for deterministic finality rather than probabilistic finality	Ø Transaction Validity
Performance and scalability testing (As per current MEPS+ HA and stress testing specifications)	* Parked under future consideration

2.3.3 Epics and User Stories

Epic	Summary
Transfer Funds	• Initiate funds transfer to counterparty • Receive funds from counterparty • Move funds across channels (Hyperledger Fabric only) • Provide suggestion for fund movement across channels
Queue Mechanism	• View incoming unsettled payment instruction • View outgoing unsettled payment instruction • Queue payment instruction upon insufficient liquidity • Handle new payment instructions when there is an existing outgoing queue • Settle next payment instruction in queue automatically via FIFO by priority when sufficient funds are available
Queue Reprioritisation	• Update priority level of unsettled outgoing payment instruction • Cancel unsettled outgoing payment instruction • Queue payment instruction despite sufficient liquidity • Put unsettled outgoing payment instruction on hold • Reactivate unsettled payment instruction that is on hold
Gridlock Resolution	• Initiate gridlock resolution mechanism that performs multilateral netting

	• Perform netting as part of multilateral gridlock resolution • Participate in the gridlock resolution mechanism that performs multilateral netting • Initiate bilateral netting
Balance Enquiry	• Perform balance enquiry on itself • Perform balance enquiry on all commercial banks participating in the DLT • View history of transactions performed
Transaction Validity	• Ensure outgoing transaction is valid and authentic • Ensure incoming transaction is valid and authentic • Prevent double spending • Ensure completed transaction are final and irrevocable
Manage Accounts	• Identify participants • Store public key • Update public key • Update bank name • Suspend bank • Onboard new bank
Privacy	• Privacy of total account balance • Privacy during gridlock resolution • Privacy of counterparties during fund transfer • Privacy of transaction contents during fund transfer • Privacy of transaction in queue
Pledge	• Initiate the pledging of funds to the DLT account • Create pledged funds on DLT • Perform funds transfer to the pledging bank
Redeem	• Initiate fund redemption from DLT • Remove funds from DLT
Versioning	• DLT platform update strategy • Update code version • Check current code version

3 Platform and System Designs

This section is covered in the more detail in the Technical Report for each prototype.

3.1 APIs and User Interface

As part of Ubin Phase 2, the API definitions are standardised across the workstreams to allow for ease of integration in the future. In line with that, a common user interface (UI) was developed to demonstrate the functionalities in Ubin Phase 2.

There are two main dashboards designed for the UI, one for MAS as a Regulator and Central Bank and another for the banks.

The available functionalities in the MAS dashboard are:

- Fund transfer
- View balances of all banks in the network

- View transaction history (only MAS transactions - pledge, redeem and fund transfer involving MAS)

The available functionalities in the bank's dashboard are:

- Fund transfer
- Queueing mechanism - updating priority, putting on hold or activating, cancelling a queued payment instruction
- Pledge
- Redeem
- Cross-channel fund movement (Hyperledger Fabric only)
- View its own balance (For Hyperledger Fabric workstream, the balances for each of a bank's bilateral channels are also displayed)
- View transaction history

4 Functional Requirements

The functionalities in scope of this phase are grouped into 11 high-level Epics and further detailed into 43 user stories for each platform, cutting across the various activities in DLT.

To join the DLT, the participating banks need to be on-boarded. A series of activities during the on-boarding are part of the **Manage Account** process. Once the banks are on-boarded, they need to have the collateral to **Pledge** funds from MAS, in order to credit their account in DLT with Digital Currency.

To view the **Balance** in DLT account, each participating bank has a UI in each platform.

A bank can trigger **Fund Transfer** to make payment to another bank. If the sending bank has sufficient liquidity to complete the payment, the transaction will complete and settle. Settlement involves the fund being debited from the sending bank and credited to the receiving bank. On the other hand, if the sending bank doesn't have sufficient liquidity, the payment instruction will be placed in the queue. Following the **Queue Mechanism** design, payment instructions in the queue can be processed automatically upon sufficient liquidity. The issuing bank is also allowed to perform actions as designed in **Queue Re-prioritization** to change the settlement priority of payments in the queue.

As and when a gridlock is detected, **Gridlock Resolution** can be triggered to perform the LSM resolution. The design of how the gridlock can be triggered and resolved differs from platform to platform. Based on the design, once the resolution is found for participated transactions, settlement will take place. The unresolved transactions will remain in the queue.

The participating banks are also given options to **Redeem** fund with MAS, by requesting MAS to convert the digital currency and credit back to their RTGS account.

All the transaction performed within the DLT is to meet the objective of MAS whereby **Privacy** and **Transaction Validity** could not be compromised.

Due to the technology for these 3 platforms are still evolving and noted that this is a Decentralised DLT, **Versioning** of each platform had taken into consideration and will be discuss in detail in Future Consideration section.

4.1 Fund Transfer

In Project Ubin Phase 2, fund transfer refers to a payment instruction to move funds from a sender (a bank) to a receiver (another bank). The payments are settled immediately on the basis that the sender has sufficient liquidity and has no pending payment instructions in its outgoing queue (refer to Section 4.2).

4.2 Queue Mechanism

When a bank creates a payment instruction for a fund transfer but has insufficient liquidity, the payment instruction is put into a queue. A bank has visibility of all its outgoing and incoming payment instructions in queue.

When the bank obtains sufficient liquidity, the transactions in queue are settled automatically based on the following sequence:

- Priority - There are two levels of priority (*Normal* and *High*) for each payment instruction. Payment instructions with *High* priority will be settled ahead of any payment instruction with *Normal* priority, regardless of creation time.
- First in First Out (FIFO) - For payment instructions within the same priority level, the oldest payment instruction based on creation time will be settled first.

The following are options for the sending bank to perform on the payment instructions in its outgoing queue.

- **Change the priority** of an unsettled payment from *Normal* to *High*, and vice-versa
- **Cancel** an unsettled payment - Removes payment instruction from the outgoing queue
- Put an unsettled payment to be **on-hold** - Payment instruction will remain in the outgoing queue and will not be settled or included in gridlock resolution
- **Activate** an on-hold payment - Payment instruction will be settled upon sufficient liquidity or included in gridlock resolution

Table 1: Putting an unsettled payment on-hold where position = current balance + incoming - outgoing

Queued Transfer	Balance	Position	Status
$150	$100	**-$50**	Active
$25	$100	**-$75**	Active
$10	$100	**-$85**	Active

Table 2: After putting a transfer on hold

Queued Transfer	Balance	Position	Status
$150	$100	**-$50**	Active
$25	$100	**-$50**	**Hold**
$10	$100	**-$60**	Active

If the outgoing queue contains *Active* payment instructions of any priority, a new payment instruction of *Normal* priority will automatically be appended to the outgoing queue. This is regardless of the bank's liquidity at the point of creation of this new payment instruction. However, if a payment instruction of *High* priority is created and there is no other *High* priority payment instruction in the outgoing queue, it will be settled immediately if the bank has sufficient liquidity. If there are other *High* priority payment instructions in the outgoing queue, this new *High* priority payment instruction will be added between the newest *High* priority and the oldest *Normal* payment instructions (if any).

Table 3: A normal priority funds transfer of $50 is added to the end of the queue by date.

Queued Transfer	**Priority**	**Date (d/m/y)**
$150	High	01/06/2017
$25	Normal	20/05/2017
$10	Normal	21/05/2017
$50	**Normal**	**02/06/2017**

Table 4: When the $50 funds transfer is changed to High priority it moves above the Normal priority list and into the High priority list sorted by date.

Queued Transfer	**Priority**	**Date (d/m/y)**
$150	High	01/06/2017
$50	**High**	**02/06/2017**
$25	Normal	20/05/2017
$10	Normal	21/05/2017

4.3 Gridlock Resolution

A gridlock occurs when a group of senders and receivers with queued payment instructions are unable to settle unilaterally in a sequential mode due to insufficient funds, but the net liquidity across the gridlock participants is sufficient to settle the transactions simultaneously. The diagram below illustrates a simple gridlock scenario where three payment instructions i.e. $50,000, $100,000, and $130,000 are unable to settle (in a sequential manner) as each sender bank does not have sufficient liquidity. The purpose of a Liquidity Savings Mechanism (LSM) is to resolve the gridlock scenarios when these transactions are executed on a net (parallel) basis. LSM is traditionally performed by a centralised system as the algorithms for LSM typically require a consolidated, single view of all payment instructions in the system.

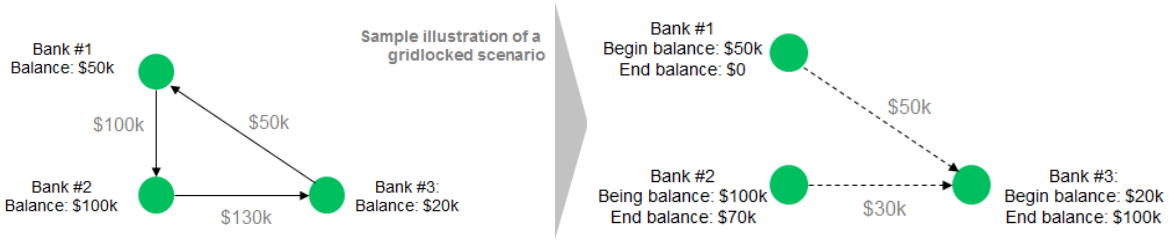


Figure 5:Sample illustration of a gridlock scenario

On the other hand, a deadlock arises when the gridlock across the participants results in a negative net liquidity where it is not possible to resolve unless new liquidity is injected to the system. An example of a deadlock scenario is illustrated below:

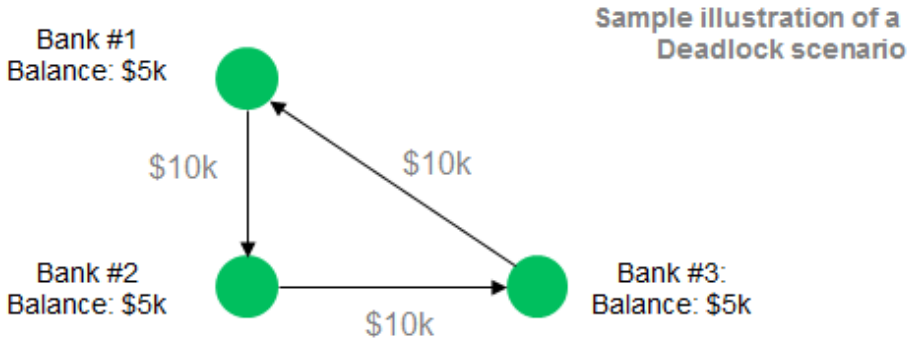


Figure 6:Sample illustration of a deadlock scenario

Similar to fund transfer and queueing mechanism, privacy is an important consideration during gridlock resolution. This section in the sub-reports describes how each workstream handles privacy in their designs.

4.4 Pledge and Redeem

A bank **pledges** funds by requesting MAS to debit the requested amount of fiat funds as collateral in its RTGS account and injecting the equivalent amount into the bank's DLT account. A bank sends a request to MAS to **redeem** funds when it would like to withdraw its funds from its DLT account and have MAS credit its RTGS account accordingly. In the design across all workstreams, the general principle followed is that the overall money supply should not have a net increase at any one point of time. For example, a pledge request should involve the debiting of the RTGS account prior to crediting the bank's DLT account.

In Ubin Phase 2, the interface between the DLT and the RTGS system, MEPS+ is simulated using a mock service. The assumption is that the bank has sufficient funds in MEPS+ and MAS automatically approves the pledge request. Likewise, it is assumed that all redeem requests are automatically approved by MAS, provided the redeem amount is not greater than the DLT account balance.

4.5 Balance Enquiry

A specific bank can view all outgoing and incoming payment instructions that it is involved in as well as its current balance. No other banks should be able to view its total balance at any time.

The central bank or regulator should have the visibility of the balances of all banks in the network at any one point of time. This allows the central bank or regulator to monitor the overall liquidity in the network and allows it to detect if there are any fraudulent activities e.g. sudden

reduction in balance when there is no redeem request. Given the central bank is responsible for pledge and redeem requests, it is also critical that it can monitor both the DLT accounts as well as the MEPS+ accounts to ensure the accounts can reconcile.

4.6 Versioning

Versioning, as a requirement, is to have a point of view on how the platform and application versions can be upgraded. Being distributed in nature, deployment and upgrade methods may need to evolve from the traditional approach.

5 Technical Requirements

Aside from the functional requirements that are reflective of existing RTGS functions, the DLT platform and design should ensure that transactions are valid, systematically and from business logic perspective. Additionally, the privacy of the bank and transactions should not be compromised.

5.1 Transaction Validity

A key characteristic of all DLT platforms is the consensus mechanism. As part of that, it is key to ensure transactions are valid and agreed upon by all necessary participants.

5.2 Privacy

Privacy is a key requirement for Ubin Phase 2, to ensure that the banks are only privy to the transactions involving itself. A bank should not be able to view the details of transactions of which they are not a participant. The privacy should also be maintained during gridlock resolution.

5.3 Technical Matrix

Corda	Fabric	Quorum
Corda 1.0.1 (Ubin specific)	Fabric 1.0.1 (with CouchDB) Docker 17.09.0-ce Go 1.7.6 Node.js 6.9.5 NPM 3.10.10	Quorum 1.5 Go 1.7.3 Node.js 6.11.3 NPM 3.10.10

6 Interface Specifications

The API definitions across the three workstreams are designed to be common across all. There are some additional fields that are either returned as part of the platform or are added due to design considerations (e.g. channels for Hyperledger Fabric). There are also workstream-specific APIs that are indicated where applicable.

Ubin Phase 2 defines a common set of RESTful API endpoints that are implemented across three workstreams. JSON payloads are used to make requests and receive response data.

Each of the three implementations contain differences that are specific to the respective DLT platform. There are additional fields that are added or removed between the platforms due to design considerations (e.g. channels for Hyperledger Fabric).

7 Testing

The aim of the functional tests for Ubin Phase 2 is to validate the **gridlock resolution mechanism,** and contribute to the final assessment across the three workstreams.

The types of testing executed in Ubin Phase 2 are as follows:

Unit Testing

- To verify the quality of development by engineers
- Associated with JIRA user stories
- Covers the basic functionality validations

Functional Testing

- To verify the resolution of various gridlock scenarios
- Varying number of participants/banks
- Varying data sets to cover different combinations of number of transactions, values etc.

Note:

- The purpose of these functional test scenarios is not to test the LSM algorithm but rather to verify the capabilities across the three workstreams in detecting and resolving gridlock scenarios
- The same set of test data is used across the three workstreams in order to simulate the exact-same conditions for assessment
- The priority (normal, urgent) of a payment instruction is assumed to be the same in the scenarios tested
- Gridlock timeout and/or retry test is not covered in this scenario testing

7.1 Testing Objective & Approach

7.1.1 Unit Testing

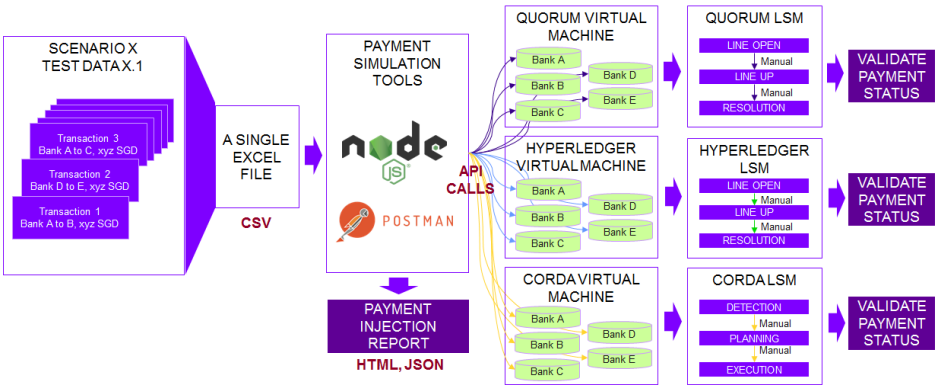
Unit test is conducted by developers during the sprint to verify that a user story has been developed as per requirements. It serves as the quality gate to transit any user story to DONE status. The user stories are also tested and showcased via Sprint Demo.

7.1.2 Functional Testing

One of the key focuses of Ubin Phase 2 is to implement a LSM or gridlock resolution mechanism in a DLT. In order to verify LSM across the different workstreams, 24 functional test cases were defined across scenarios having two, three and five bank nodes as well varying starting balances and payment instruction amounts. An identical set of test data is executed across all three workstreams and the results are verified against the expected results against the LSM algorithm for each workstream.

7.1.3 Overall Test Approach

Unit testing and functional testing was conducted through the RESTful APIs. The diagram below illustrates the test harness set up and test approach.



7.2 Testing Scenarios and Data

Refer to: https://github.com/project-ubin/ubin-docs/blob/master/UbinPhase2-Testing.pdf

8 Future Considerations

Ubin Phase 2 has successfully demonstrated that with three different designs on three different platforms, a traditionally centralised RTGS process could be executed in a decentralised manner without compromising privacy.

Apart from the key findings and observations, there were additional discussions and learnings brought up during the demonstration sessions led by Accenture with the consortium of bank representatives. Although these discussions were not intended topics for

Ubin Phase 2, they are worth addressing in view of operationalising a fully-functional DLT-based RTGS system. Each of these considerations are likely to require an in-depth assessment and design which could impact the current operating model, policies and procedures as well as the technicalities of the system. Although not an exhaustive list, the discussions are categorised into the following six topics.

8.1 Resiliency and Cloud-Readiness

A key benefit of DLT is the distribution of processing and data storage across all nodes in the network, mitigating the risk of single point of failure. In theory, it also allows the system to recover quickly and continue operating in the event of system failure or other disruptions, as data and processing are distributed across nodes. Although Section 5 showed that resiliency across the network can be realised with the various design considerations, more work should be done to understand the resiliency of a distributed system, particularly, the recovery point and high-availability backup. In addition, in a conventional centralised system, Disaster Recovery and High Availability strategies focus on a single organisation. In a decentralised system, the approach should consider the entire network and cross-organisation.

Similarly, all three prototypes of Ubin Phase 2 were successfully developed and deployed to the Microsoft cloud infrastructure. This underscores the feasibility of operating DLT on a cloud infrastructure.

In order to operationalise a DLT-based RTGS system on cloud, the current prototype has to be enhanced to integrate various cloud infrastructure functionalities such as environment administration,

In order to operationalise a DLT-based RTGS system on cloud, the current prototype has to be enhanced to integrate various cloud infrastructure functionalities such as environment administration, monitoring and recovery. Furthermore, interoperability of a DLT network on different cloud solutions can be further explored to cater to participants who may have different cloud providers. This also includes considering the possibility of deploying such DLT-based RTGS systems across multiple cloud services providers for a more robust and resilient solution.

8.2 24x7

Deploying a DLT-based RTGS system opens up the possibility of operating 24x7, providing new opportunities for Singapore to be a global financial centre. This is especially beneficial for cross-border transactions across countries with no overlapping office hours, e.g. Singapore and Canada. The DLT-based RTGS system also allows interbank fund transfer to be completed without requiring all participants to be active. However, there are several operational considerations that need to be addressed before a truly 24x7 decentralised interbank payment system can be deployed.

These include:

- Transacting value-date handling for transactions executed after operating hours and on public holidays Transaction/Handling fees and incentives during and after business hours

- Differing cut-off time of different banks
- Varying SLA requirements for bank operations and customer services
- Determination of foreign currency exchange rates, particularly for transactions involving non-SGD after trading hours, or dependent on parties outside the DLT network (e.g. securities or foreign markets)

8.3 Initiating and Triggering Liquidity Savings Mechanism (LSM)

With the current design, the RTGS system triggers the Liquidity Saving Mechanism (LSM) algorithms centrally through a predefined interval. Ubin Phase 2 prototypes demonstrate that there are different methods to initiate gridlock resolution in a decentralised system (and also detect and avoid simultaneous gridlock resolution). In other words, there is flexibility to select how and which node initiates gridlock resolution, be it scheduled, user-triggered or based on a predefined state or event.

However, the mechanism of initiating gridlock resolution may lead to unintended consequences and inequality among participants in the network. For example, a participating bank may instigate that gridlock resolution be triggered to its preference while other participants may not be in the right liquidity position at that time. Therefore, to ensure a degree of fairness to all participating banks, a more detailed analysis of the transaction patterns across the entire processing day is required in order to arrive at an optimal design for the mechanism to initiate gridlock resolution. Such mechanisms should also take operational factors into account, such as operating hours and liquidity threshold in the system.

In addition, LSM processing may take a longer time in a large network which may result in delay to the settlement. Hence, the desired design is to decouple LSM from fund transfer. One consideration is to improve this prototype to segregate bank balances by reserving a portion of liquidity for fund transfer during LSM processing.

8.4 Degree of Decentralisation

Another observation that came out of Ubin Phase 2 is that while decentralisation is technically feasible, there are multiple facets to consider to achieve a fully decentralised model where every node is equivalent. This is because, in practice, not all banks (nodes) are equal. Participating banks may differ in the transaction volume and the incentive to participate in an equally distributed network may vary. This may lead "smaller" participants to be unable and/or unwilling to bear the cost of ownership for infrastructure equally. Drawing inspirations from other existing RTGS systems, it is observed that there might be a potential to implement a hierarchical system whereby participation of the network can be split between direct and indirect participation. Such a model may work well especially when there is a potential of including participants outside of the financial services industry. For such a ‘semi’ decentralised network to be feasible, strong governance and policies are still needed to govern the relationship and service level agreement between participants, such that the model does not create an economic advantage to larger banks.

In a broader context, extending beyond Ubin Phase 2 which focuses on domestic interbank payment, multiple ledgers across different DLT platforms maybe required to facilitate an end-to-end Delivery vs Payment (DvP) process. Such a process will still need to ensure atomicity across the transaction lifecycle. Future development is needed to explore the feasibility of crosschain feasibility which may leverage on the work already done on the Interledger Protocol (ILP). The business and operations considerations should also be evaluated to enable a fully decentralised DvP process.

8.5 Managing Multiple Nodes

In a decentralised RTGS system, each participant is required to manage and operate its own node. This includes maintenance and support of both hardware and compatible software. For

a DLT-based RTGS system to work effectively, it is important to ensure that all banks (nodes) across the entire network are consistent. This includes both functional and nonfunctional aspects such as:

- Functional: Consistency of algorithm and system parameters to ensure the operability of all nodes in a decentralised network. It is necessary to ensure that all participating banks (nodes) are effectively able to execute a distributed queue prioritisation, triggering of gridlock resolution and handle exceptions for each node such as time-out and retry. Also, when the network is opened up to other banks from different jurisdictions with different liquidity and infrastructure, operational level agreements and service level agreements are needed to manage the serviceability of the node.

- Non-functional considerations include operating systems and software patches, as well as infrastructure configurations as this has a bearing on security and vulnerability of the entire DLT network.

Another key aspect of managing multiple nodes is to consider the process and governance to add and remove participant nodes. A central system operator may no longer be required in a decentralised RTGS system, but a central governance body is still required to govern the consistency of the entire network. The governance parameters and process should be defined in detail to ensure the consistency and governance of a multinode decentralised RTGS system.

Moreover, network latency can be a challenge in a very large multi nodes network i.e. with more than a thousand nodes. In the case of RTGS, it may impact the efficiency of the LSM algorithms. One option is to decentralise the LSM execution by distributing the LSM processing to smaller clusters of nodes, perhaps in a pyramid or tiered system where the leftovers are aggregated and netted at the next level. Distributed netting may alleviate efficiency and scalability concerns.

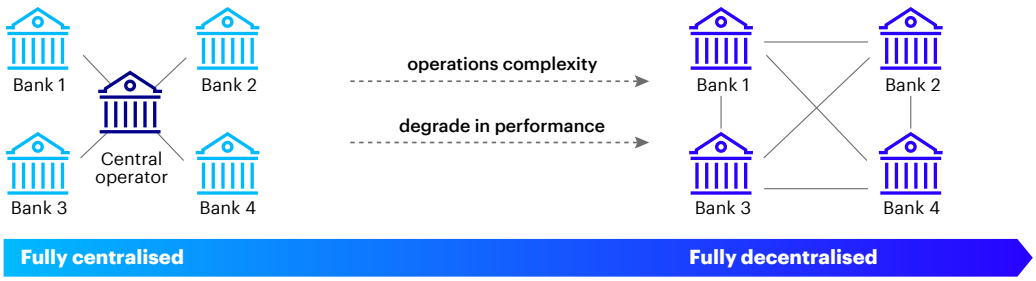


Figure 7: Illustration of transitioning from a centralised RTGS system to a fully decentralised RTGS system

8.6 Role of Central Bank

With the potential of operating a DLT-based RTGS system, the conventional role of a central bank or payment system operator as the centralised infrastructure operator in the ecosystem will be obsolete. This would also mean that a central financial market infrastructure operator would not be necessary, as the processes and data are distributed across the participants in the DLT network. A DLT-based RTGS system reduces the costs and resources for the dayto-day operations and eliminates the risk of the central bank being the single-pointof- failure of the entire financial ecosystem.

However, Ubin Phase 2's findings suggest that the mandate to oversee the safety and efficiency of the payment and settlement system remains unchanged in a new decentralised model. Some of the roles that are observed include:

- Overall liquidity manager: Monitoring overall network liquidity (i.e. identify gaps and volatility) as well as intervening where necessary
- System governance: Ensuring compliance, consistency of the system operations such as software patches and parameters, and hardware configurations, as well as creating rules and guidelines for all players to operate and further monitoring, adding or removing participant nodes
- Service Level Agreement (SLA) governor: Defining the SLA in the decentralised system and, where possible, eliminating the need of multiple bilateral SLAs between banks

9 Conclusion

With the collaborative effort of MAS, ABS, 11 financial institutions, four technology providers, and Accenture, Ubin Phase 2 successfully concluded with its intended goals achieved.

The findings from Ubin Phase 2 demonstrate that all three workstreams can perform fund transfers, queue reprioritisation and gridlock resolution in a decentralised manner, without compromising the privacy of the transactions. Each workstream has its own merits and design considerations to meet the requirements. Findings and observations from the three workstreams in Ubin Phase 2 have also contributed to topics such as scalability, performance and resiliency of a DLT-based RTGS system. The project also identified areas where the prototypes can be further improved before becoming fully operationalised.

Ubin Phase 2 has also extended the conversation beyond the technology. The project highlighted six key future considerations which include a point of view on the role of a central bank and regulator in a decentralised payment system. To achieve round-the-clock operations and managing multiple nodes in a cloud environment, rigorous governance, policies and operating models need to be put in place. Although a centralised operator is no longer required, a central bank or regulator still plays a vital role in this critical payment network infrastructure which needs to be redefined.

Building on the success of Phase 1 and this Phase 2, MAS and its partners will continue to journey towards the goal of making Singapore a Smart Financial Centre. Future phases of Project Ubin could focus on a decentralised bonds payments system, which could be supported by MAS and the participant banks with execution driven by Singapore Exchange. This could deliver a more efficient fixed income securities trading and settlement cycle through DLT. MAS will also focus on new methods to conduct cross-border payments, leveraging on the findings from Phase 1 and Phase 2. These are in tandem with the eventual aims of contributing to the community and to develop more efficient alternatives to current financial systems based on DLT.