DEUTSCHE BUNDESBANK
EUROSYSTEM

DEUTSCHE BÖRSE GROUP

www.deutsche-boerse.com

BLOCKBASTER

Final Report

Table of Content

1 Overview of BLOCKBASTER

In March 2016 Group Deutsche Börse and Deutsche Bundesbank jointly started a project to undertake **block**chain **ba**sed **s**ettlement **te**chnology **r**esearch (in short: BLOCKBASTER). The project, experimenting with blockchain in order to improve back-office services, was motivated by developments in the financial industry as well as by advances in new technology. Both parties, Deutsche Börse and Deutsche Bundesbank, operate financial market infrastructures and have enjoyed a long-time joint development of specific features of settlement systems. Therefore, the continued operation of financial market infrastructures provided a common ground and starting point for a joint experiment to find out more about the usefulness of blockchain technology.

The aim of the joint project by Deutsche Börse Group (GDB) and Deutsche Bundesbank (BBk) was to create a conceptual prototype for a Blockchain-based transfer and settlement of securities and cash. The Prototype was intended to function as a sandbox for testing Blockchain-based securities settlement including the full lifecycle of bond issuance, settlement, corporate actions, redemption as well as cash transfer including provisioning of digital coins, transfer between market participants and settlement of cash payments.

In detail, the aims were

- to understand the functioning of Blockchain and related technologies,
- to assess the safety and reliability of Blockchain-based transfers,
- to improve the understanding of the possible transaction costs of Blockchain-based settlement,
- to judge on the efficacy and efficiency of Blockchain-based processes,
- to assess the appropriateness of Blockchain-based transactions for financial transactions.

The project was purely for research purposes and was not intended to produce an operational system.

2 Business Case

Both parties agreed to focus on the role of a financial market infrastructure. Therefore, the business case was concentrated on back-office issues. A matched trade in cash or securities, both matched at an exchange or over-the-counter, should be settled via blockchain. Therefore, pricing or market-making were not part of the prototype. In addition, the settlement was considered to be a gross settlement of every transaction. Special features of modern RTGS-systems like liquidity-saving mechanisms or distinguished algorithms to serve particular needs of liquidity and risk managers were not included in order not to distract from the analysis whether and how blockchain or distributed ledger technology could be used to improve on settlement.

2.1 Concept

The concept of the prototype is depicted in Figure 1. As a starting point we assume a transaction bank having multiple customers (banks) who hold part of their liquidity in an account with the transaction bank. For the sake of this project this transaction bank could be a central bank or a commercial bank. The latter case implies that the banks hold commercial bank money with the transaction bank, the former case implies central bank money. In our concept we called the transaction bank the “Coin Providing Authority”. It plays a special role for the cash-cycle since digital coins, representing the money on the blockchain, can only be transferred onto the blockchain or off the blockchain via the coin providing authority. If money is to be transferred onto the blockchain the requesting entity – which we call “Coin Distributor” – has to own money at an account with the coin providing authority outside the blockchain. That means, no money is created or destroyed on the blockchain and the amount of money circulating is still determined solely outside the blockchain. Money which is available outside the blockchain can be transferred onto the blockchain and become a “Digital Coin”.

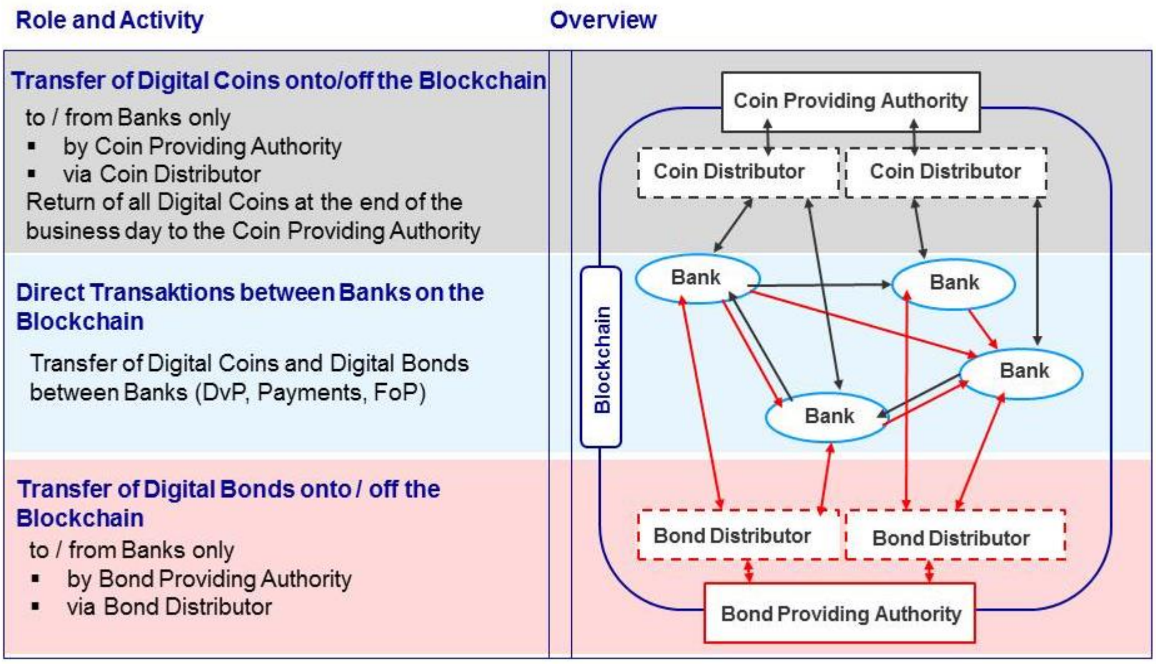

Figure 1: High Level Overview of the Functionality

In order to avoid any substitution of forms of money by digital coins, the digital coin is only available for intraday payments and settlement. At the end of the business day all digital coins are transferred automatically off the blockchain to the accounts with the coin providing authority. The coin distributors can distribute coins to banks which do not have to be direct customers of the coin providing authority. However, we did not foresee non-banks to receive, hold or transfer digital coins.

In close analogy securities which are held with a central securities depository can be transferred onto the blockchain to become a “Digital Bond”. They can only be transferred via the “Bond Providing Authority” which resembles in its functions a central security depository. They are transferred to the bond distributor who in turn can transfer them to banks, but not to non-banks. Bonds transferred onto the blockchain stay there until they are deliberately transferred back or redeemed. So banks can hold digital bonds over night, whereas they cannot hold digital coins overnight.

Banks can use digital coins and digital bonds to transfer them among each other in three forms: payments (only transfer of digital coins), security settlement free-of-payment (only transfer of digital bonds) or security settlement delivery-versus-payment (simultaneous transfer of digital coins and digital bonds). In addition, the prototype was designed to perform two corporate actions: Interest payments and redemption.

2.2 Neutrality in terms of monetary policy

Since no money is created or destroyed on the blockchain and no digital coins exist overnight we could rule out any implications for monetary policy. There will be no balance sheet effects for banks or for the coin providing authority. There will be no substitution between different forms of cash, since at the end of the business day all available money is off the blockchain. Non-banks have no access to digital coins, so the functions of credit supply and risk transformation by banks remain unaffected. This was intentionally part of the concept since the project was intended to focus on the use of blockchain for settlement purposes. Safety and efficiency of payments and settlement systems were the focus of this project.

2.3 Functional requirements

The aim of the Blockbaster project was to evaluate Distributed Ledger and in particular blockchain technology as a basis for payments and securities settlement. The first step towards this aim was achieved in November 2016 with the successful demonstration of the capability to execute the desired functionality: security settlement delivery-versus-payment (DvP), security settlement free-of-payment (FoP), payments, redemptions and interest payments. Up until then we had managed to run the prototype on a blockchain-basis by Hyperledger, Hyperledger Fabric, version 0.6, and to display that in public. All functions performed well, running on a PC using a graphical interface where different participants could log in using their credentials. Therefore, confidentiality of data was in principle provided and all regulations starting with know-your-customer rules could be fulfilled since we operated a permissioned blockchain using the coin providing authority and the bond providing authority as operators of the system. All other participants received their permission by these two authorities.

3 Performance Tests

Once functional requirements were demonstrated, the next step in the evaluation was to show the capability to handle the transaction volume and speed of a production system without excessive resource utilization. Therefore we tested the performance of two different technical stacks under assumed market conditions. These stacks were Hyperledger Fabric (version 1.0) and the Digital Asset Platform and subjected to demanding performance tests.

3.1 General test approach

In order to allow for realistic performance evaluations of the respective platforms, a base scenario was agreed to regarding the number of transactions (i.e. Payments, FoP as well as DvP) and actors (i.e. transacting banks as well as bond and coin distributors). Performance was measured and evaluated across three indicators:

- The transaction volume was measured based on the throughput - the number of successfully executed transactions per minute,
- The speed was measured based on the latency - the time to execute a single transaction in the load scenario,
- The resource utilization was measured based on the storage usage, the CPU utilization and the network traffic.

In addition, the number and types of errors that occurred were recorded.

3.2 Hyperledger

Hyperledger is an open source collaborative effort created to advance cross-industry blockchain technologies. It is a global collaboration, hosted by The Linux Foundation, including leaders in finance, banking, Internet of Things, supply chains, manufacturing and Technology. One of the founding members was the Deutsche Börse Group.

3.2.1 Product description

The following information is mainly taken from the master documentation of Hyperledger Fabric version 1.0 which can be found here[1].

[1] For a more complete architectural description of Hyperledger Fabric, please read “Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains” https://arxiv.org/abs/1801.10228

One of the resulting blockchain architectures under Hyperledger platforms was called Hyperledger Fabric. Intended as a foundation for developing applications or solutions with a modular architecture, Hyperledger Fabric allows components, such as consensus and membership services, to be plug-and-play. With its open-source approach and IBM as main propagator, Hyperledger Fabric is used in many different proofs of concepts and prototypes in many different industries and use cases.

Logical components for the blockchain workflows are realized within Hyperledger Fabric via so called nodes (technical components, see Figure 2):

- With client nodes the interaction of the user with the system is realized, client nodes send transactions into the system and forward validated transactions.
- Endorser nodes are responsible for the validation of transactions. So called endorsement rules allow for a rule-based implementation of different validation requirements.
- Committer nodes write blocks of transactions to the ledger and maintain the state.
- Ordering service (one functional service realized through several interacting nodes) ensures the logical/temporal sequence of transactions.

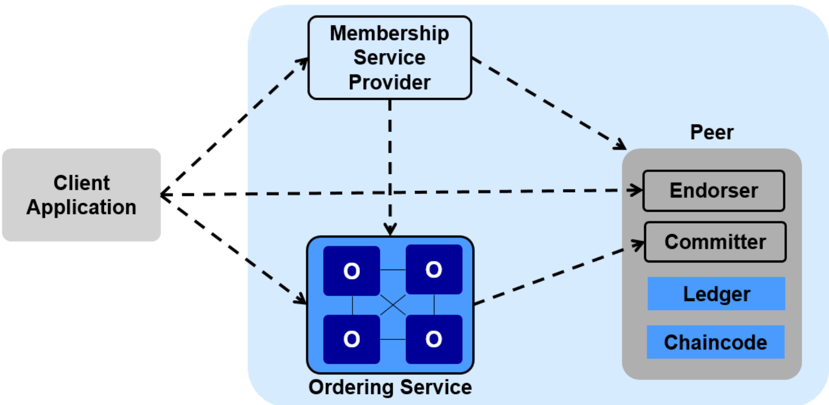

Figure 2: Simplified network architecture of Hyperledger Fabric version 1.0

Before any transaction is initiated, users must be registered and enrolled with the organization’s membership service to authenticate to the network. The chaincode (code representing the business logic, “smart contracts”) is installed on the peers and instantiated on the blockchain. The chaincode contains logic defining a set of transaction instructions. An endorsement policy has also been set for this chaincode, stating that all peers must endorse any transaction in our case. The transaction flow is illustrated in Figure 3.

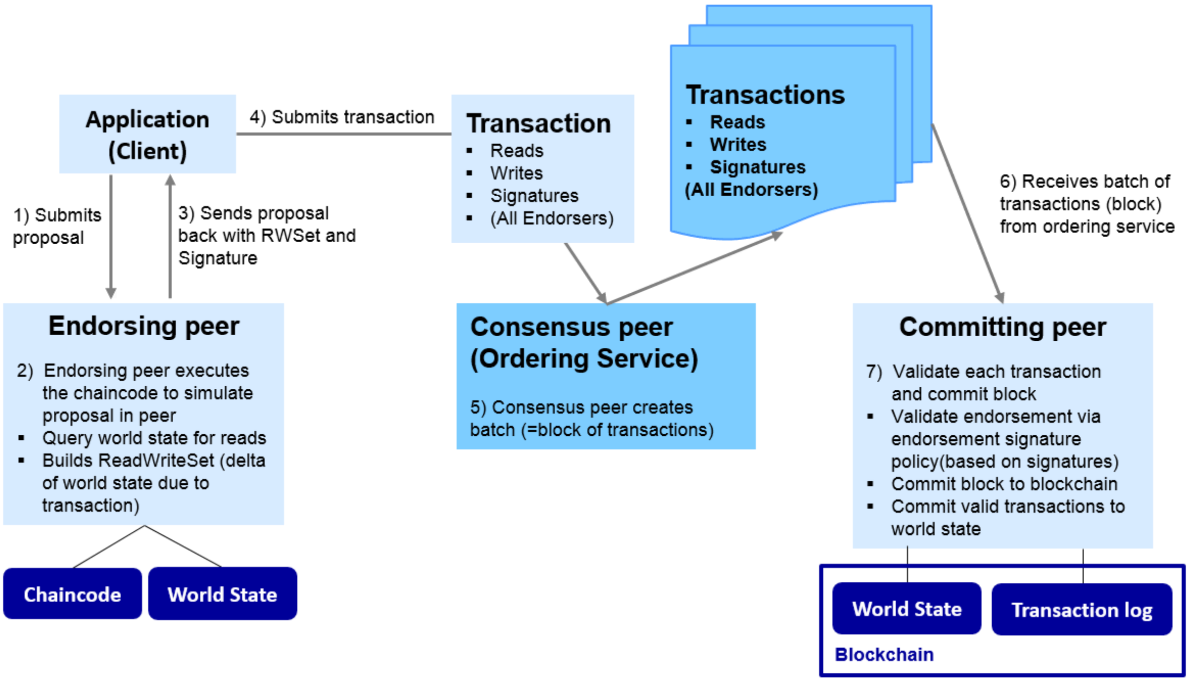

Figure 3: Transaction flow in Hyperledger Fabric version 1.0

A user participates in the network through a client application and initiates a transaction via his application UI. Next, a transaction proposal is constructed. A transaction proposal is a request to invoke a chaincode function so that data can be read and/or written to the ledger (i.e. write new key value pairs for the assets). The transaction proposal is then submitted to the designated endorsing peers according to the endorsement policy. Each of the designated endorsing peer executes the chaincode to simulate the proposal independently. The results of the execution/simulation are passed back to the user as a “proposal response” along with the endorsing peer’s signature. The user’s client application verifies the signature and compares the proposal responses to determine if the proposal responses are the same. The user’s application (client node) then submits the transaction and signatures of endorsing peers to the ordering service, which orders the transactions chronologically by channel and creates blocks of transaction per channel. The blocks are sent to the committing peers. The transactions within the block are validated to ensure the endorsement policy is fulfilled and to ensure the changes are applied to the correct state of the ledger. Valid blocks are then committed to the Blockchain. Additionally, an event is emitted to notify the client application that the transaction (invocation) has been immutably appended to the chain.

3.2.2 Test system

The first Blockbaster prototype which was presented in November 2016 was built based on the then recent version of Hyperledger Fabric, version 0.6. This first prototype used two different roles: The first role is the validating peers which are tasked with ensuring the validity of the transactions as well as detecting and preventing malicious behaviour, for example double-spending. Among the validating peers there was a consensus mechanism, which could not only validate the transactions but also determine an order in which the transactions were written into the blocks. The other role was the non-validating peer which stored and propagated the blockchain and the new blocks. The design was suitable for a PBFT-consensus mechanism which was in our case substituted by a so called proof of authority where the consensus was decided upon by the two central institutions: coin providing authority and bond providing authority.

In version 1.0 performance issues were addressed. One of the most important architectural change addressing performance improvements was the fact that the transaction validation and the transaction commitment (writing of transactions to the blockchain) was split into three functional steps: endorsement, time ordering and commitment and could thus be parallelized.

With version 1.0 private channels have been introduced in Hyperledger Fabric. These allow a group of participants to create a separate ledger of transactions. Loosely spoken private channels are blockchains between a specifically defined set of users. This ensures that only users “subscribed” to the respective channel have access to the data in that channel.

3.2.3 Test setup

Creating test data as well as defining the test scenarios for Hyperledger Fabric were mainly driven by two architectural features of version 1.0:

- The platform contains an integral messaging service that allows to receive blockchain events, e. g. confirmation that a transaction has been written, from the blockchain. This service is elementary for building applications based on the Hyperledger Fabric.
- The architectural split of endorsing (verification of transactions) and committing (writing transactions to the Blockchain) leads to the requirement that the Blockchain state – as far as the transaction under consideration is concerned – has to be kept unchanged for the time interval between endorsing and committing.

Given the features mentioned above, the tests rely on an adjusted set of test data of 1,000 different users/banks, 500 bonds and 200,000 transactions. The 200,000 transactions were split into 100,000 DvPs (Delivery versus Payment), 50,000 FoPs (Free of Payment) and 50,000 Cash Payments. All nodes and components were set up in AWS within one Availability Zone and each of them was a separate AWS EC2 instance.

Two metrics were used to measure the performance tests:

- Latency: the difference between the transaction submission time and the time when the transaction is written to a block
- Throughput: volume of transactions per minute & second

3.2.4 Results

The tests were run on average for 35 minutes to complete the intended transactions and sufficiently high throughput rates and an average latency of two digit millisecond could be identified. We could observe a certain amount of transactions conflicts caused by the architectural features of Hyperledger Fabric version 1.0 mentioned above. These transaction conflicts, however, could be accepted within the scope of the performance tests due to improvements on the architecture made in the latest version of Hyperledger Fabric[2].

The transaction conflicts for FoP was noticeably lower than those for DvP or Cash Payment transactions. One explanation for this fact could be that the transactions contain a relatively high number of bonds, while the currency leg in Payment as well as in FoP transactions consists always only of EUR. Thus, transaction conflict is less likely to occur for FoP transactions than for DvP and Payment transactions, as the “currency” state of the blockchain is more likely to change between endorsement and commitment of the transactions than the bond state.

Both transaction throughput and latency depend on the chaincode implemented to the performance test. More complicated chaincode tends to slow down the transaction processing speed and worsen the overall performance.

Other finding is related to the batch timeout in the ordering node. Batch timeout is the waiting time after the first transaction arrives for additional transactions coming into ordering nodes before cutting a block. By increasing the batch timeout, the transaction latency can be improved, while decreasing batch timeout improves the total transaction throughput.

Further analysis shows that differences in latency are negligible given the statistics of the test setup.

However, the sources of transaction conflicts have been identified and solutions seem to be already available in later Hyperledger Fabric versions or through custom implementations by the IBM Hyperledger Fabric team (see footnote 2). Therefore, we conclude that Hyperledger Fabric seems to be sufficiently performant in order to consider it for a productive implementation for the use case under consideration, conditional on the fact that IBM’s (at this point in time) proprietary solution for the architectural issue works sufficiently well. The latency and throughput of the same transaction dataset could be potentially improved in the latest Hyperledger Fabric version. However, this has to be proved by conducting dedicated performance tests based on the latest version for the use case selected.

3.3 Digital Asset

Digital Asset (DA) is a New York headquartered global fintech company established in 2014. It provides a distributed ledger platform, smart contract modeling language, DAML, and distributed ledger technology (DLT) solutions, both to financial market infrastructures and other industries.

[2] IBM’s proposal for an enhanced concurrency control of Hyperledger Fabric can be found here: https://docs.google.com/document/d/1Z37O9nbpqBmukZQ88r2MmCr_DzTez--mCV5m3awhP-U/edit; https://docs.google.com/presentation/d/1bxUTkBJGBBD535AbSCkgoR4yqRdoZ2aZcfa5FUgL9B4/edit

3.3.1 Product description

The following information is taken from Digital Asset's non-technical whitepaper which can be found here.

The DA Platform consists of three layers: the Application layer, the Business Logic layer, and the Distributed Ledger layer. The Distributed Ledger layer consisting of two sub-layers: Private Contract Stores (PCS) and the Global Synchronisation Log (GSL) as shown in Figure 4 below.

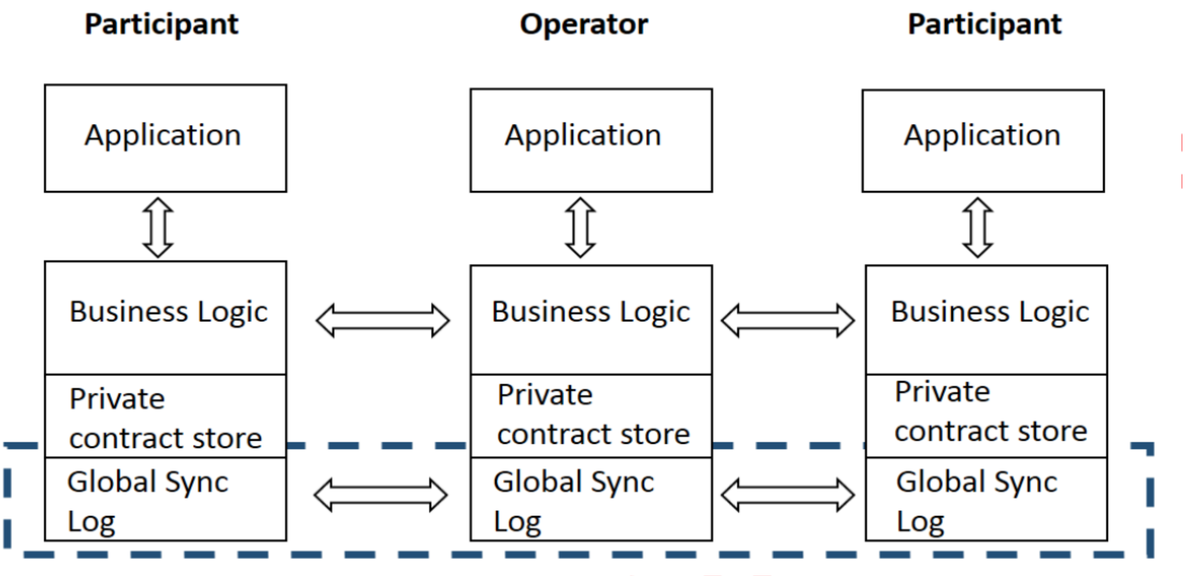

Figure 4: Overview of Digital Asset platform

The Business Logic Layer and Distributed Ledger Layer are processed by the DA Platform, and the **Application Layer** consists of custom software interfacing with the DA Platform and other systems.

In the Blockbaster setup the Application Layer consists of the User Interfaces (UI) for entering DvP, FoP and cash payment information as well information for defining and executing corporate actions.

The respective logic for these business processes resides in the Business Logic Layer. While the Business Logic Layer contains the functionality attributed in Blockchain speak to "smart contracts" the **Distributed Ledger Layer** corresponds to the data store.

Each Participant has its own **Private Contract Store** (PCS), which contains all validated contracts to which the participant is a party. The PCS is stored locally and only contains those contractual agreements that the participant is entitled to store and view.

The PCS contains a historical record of all executable contracts (both active and inactive) pertaining to a participant. To recreate this segment of the Distributed Ledger these contracts within the PCS must be paired with corresponding active evidences in the **Global Synchronization Log** (GSL).

The GSL is a log of commitments and notifications that guarantees the integrity and auditability of the distributed data stores to the contract stakeholders. The GSL establishes a common and complete set of valid transactions that, when combined with the corresponding private contract data in the PCS, comprises the Distributed Ledger. The GSL is a communication layer designed to deliver network-wide integer guarantees of transaction commitments and notifications.

In order to fully understand the mechanics of the use case investigated it is worth mentioning the “process roles” the different actors in the business process can take. In general all actors are Network Participants, i.e. Network Participants are entities that have chosen to manage their own Digital Asset Platform Instance. **Network Participant** deployments are directly connected to a shared GSL. Network participants can have two roles: Operators and Participants.

Operators codify and distribute the rules of the market. Only Operators can participate in writing and there may be one or many Operators in the same network. Each Operator can run multiple Write Instances for added resiliency. The most obvious example of an Operator is a centralized market infrastructure provider that is responsible for processing transactions.

Participants can read their data and sign transactions they participate in. Participants can see and validate their transactions evidenced on the GSL by verifying both the transaction outputs and their own signatures. In this manner, Participants act as more than just passive consumers of data from Operators, because they play the role of independently auditing and validating the integrity of the ledger.

3.3.2 Test setup

In preparation for performance testing scenarios, the Blockbaster Phase 1 functional prototype was recreated using DA’s sandbox - resembling a development environment for the DA Platform. In line with DA’s architecture, the business logic was implanted in the Business Logic Layer. The customer specific applications correspond to the role-specific UI of the prototype. In a second step this prototype was migrated to DA’s in-house production environment hosted on AWS. Performance tests were conducted for 30 test scenarios. The respective test data were generated in order to reflect both current market volumes and highest possible transaction loads. Transaction numbers were scaled up to determine whether the DA Solution can perform successfully even under the highest possible market load. The number of users and roles were varied for the different test scenarios.

As all business roles - Users, Coin Distributors (CD), Bond Distributors (BD), the Coin Providing Authority (CPA), the Bond Providing Authority (BPA) as well as the Corporate Action Executor (CAE) - generate and confirm transactions, they correspond to DA’s technical role “participant”. Technically the functionality of a logical participant is split into the application node (UI and business logic) and the participant node (synchronisation of the network and verification of adherence to market rules). Besides these participants the test setup contained one central operator verifying and writing all transactions (Committer Node).

3.3.3 Test system

The test system included the DA Platform and additional components designed to support the performance testing scenarios. A node in the test system refers to a single physical node in the system. For testing purposes, one node always means one separate, single Participant Node. Applications and the Committer are deployed as separate nodes in the testing environment to replicate the real-life behaviour of separating critical infrastructure elements from the application layer, and also because applications are in general run separated on each client's premises. The Committer Node is separated from Participant Nodes as it has a purely operational role, and separation allows us to keep functional and operational responsibilities apart.

Application Node
By choice an application node connects one-to-one with a participant node. Each application node has its own Injector, a component added to the system for purposes of performance testing, responsible for injecting transactions into the system and tracking elapsed time of the transaction lifecycle.

Participant Node
A participant node connects one-to-one to with an application node. Each participant node hosts an allocated number of parties (users). Participant nodes can host any number of parties as defined by the scenarios. All participant nodes are connected to the same Committer Node, which is the node writing transactions to the GSL.

Committer Node
There is one Committer Node with a shared Global Synchronization Log (GSL) under the application/participant nodes.

3.3.4 Findings

The performance tests were designed to measure the DA Platform's capacity to process settlement for a realistic high-volume trading system. Four transaction types were included: Payment, Free of Payment (FoP), Delivery versus Payment (DvP), and Coupon Payment. The transaction numbers were then scaled up to determine whether the DA Solution could consistently perform under higher than normal transaction loads. Thirty scenarios were designed to measure the performance and scalability of the DA Platform. Tests measured various dimensions such as node scaling, participant scaling and transaction scaling. Metrics were collected and reviewed from each test.

The base scenario used in the testing was the following:

Network and cloud setup: 3 Digital Asset deployed instances (nodes) deployed in a single AWS region in a single availability zone.

Participants: 1 Bond Providing Authority (BPA), 1 Coin Providing Authority (CPA), 1 Corporate Action Executor (CAE), 3 Bond Distributors (BD), 3 Coin Distributors (CD), and 150 Users (Banks).

Transactions (workflows) all T-Instant: 250,000 Payments; 1,000,000 FoPs; 2,500,000 DvPs; 10,000 Coupon Payment Corporate Actions under the assumption that every User (Bank) holds a portion of the relevant digital bond.

All tests were run for 30 minutes. To prove that the observed behaviour is consistent over a longer term, the base case scenario was also executed for 20 hours. These results were in-line with the 30-minute run of the base scenario.

3.3.5 Metrics definition

- Latency:
  – Measurement of time from the first command to trigger a workflow, submission to the DLT API, to the last event relative to that workflow emitted from the DLT API.
  – Intermediate steps are measured for business workflow analysis and for debugging process.
  – Average time to process one transaction is calculated for each transaction type.
- Throughput: Volume of business-relevant transactions per second.
- CPU Utilization: Refers to a computer's usage of processing resources or the amount of work handled by a CPU.
- Memory Usage per Node: Refers to usage of memory per node during processing.
- Storage Usage per Node: Refers to total usage of disk space per node during processing.
- Network Input/Output: Refers to network traffic of data sent in and out of system during processing.

3.3.6 Results

The aim of these performance tests was to evaluate the performance and scalability of the DA Solution in normal and stressed market conditions as well as when the number of parties to a transaction, participant nodes, and/or transaction volumes varied.

Using the same setting as in the base scenario, the transaction volumes were scaled upwards, which resulted in a concordant rise in throughput and a much less than proportional rise in latency and memory usage.

In further scenarios, the system was tested using the parameters of the base scenario but using different numbers of nodes. Latency and memory use per node decreased accordingly with rising number of nodes approaching a minimum asymptotically.

Further the numbers of participants were changed. With increasing number of participants latency, and memory usage per node increased much less than proportionally.

Using the DA Solution, the prototype was extensively tested in 30 scenarios and achieved the defined criteria for success. The defined benchmarks based on realistic market volumes did not put a significant amount of load on the DA Platform. The DA Solution also proved able to support predictable scaling behavior, with the potential for further optimization of performance in a production environment

4 Summary and outlook

The blockchain technology is considered to improve payment and settlement systems. The Deutsche Börse Group and the Deutsche Bundesbank both are operators of financial market infrastructures and have a particular interest to improve safety and efficiency of their services to the financial industry. Therefore, the two institutions undertook a joint experiment to analyse the potential use of blockchain technology for the settlement of securities. This project called BLOCKBASTER was solely intended to yield insights and analytical results.

An own prototype was built and tested. It was based on Hyperledger Fabric version 0.6 and later rebuilt using Hyperledger Fabric version 1.0.5. The prototype was functioning well for all designed features which were limited to the following:

- transferring money onto and off the blockchain in the form of digital coins,
- transferring securities onto and off the blockchain in the form of digital bonds,
- settlement of payments, free-of-payment security transfers and delivery-versus-payments securities transfers,
- settlement of interest payments to the holders of digital bonds,
- settlement of redemptions of digital bonds.

The prototype in principle fulfilled all basic regulatory features for financial transactions: no anonymity (KYC), confidentiality of transfers (need-to-know principle), participation of admitted peers only (permissioned network).

For the tests of performance indicators to check on scalability of the blockchain solutions in-house performance tests of the own prototype were run as well as external tests by Digital Asset Holdings on an analogously developed prototype with identical features based on Digital Asset Holding's DLT platform were undertaken.

As overall results of these tests we would like to stress the following:

- Judging by the developments in the market of providers of blockchain or DLT-based settlement engines for financial services it seems to have become mainstream to rely on private/permissioned blockchains and allow the users to strictly adhere to the basic principles of traditional financial markets:
  – Proponents design dominantly permissioned blockchain with a restricted network of participants.
  – A transparent governance which explicitly states the various responsibilities especially in admitting or deleting participants.
  – In addition, transactions are not to be undertaken anonymously,
  – while the confidentiality of transactions (strict need-to-know principle) has to be respected,
  – finality of transactions must be unambigously guaranteed, and quick, ideally instantaneously.

In short: The blockchain is to be adapted to the procedures prevalent in financial markets and not vice versa.

- During the time of the project we have encountered a further development of the basic technologies which always renders test results as outdated since the technology has already improved. Providers of blockchain or DLT-based solutions improve their offers and increasingly develop specialized software to serve particular use cases. There seems to be a trend towards products highly customized for individual use cases.
- The requirements of financial market infrastructures are different from e.g. the manufacturing industry. Even within the realm of the financial industry the use cases in payment systems and security settlement systems stand out in terms of the necessary volumes and values to be processed with absolute and ideally immediate finality. Other use cases, like export financing may need much less scalability in the operational systems. At the same time the necessary

requirements in terms of data protection and safety, especially robustness are also high in financial market industries.

- We have realized that the blockchain or DLT-based solutions offered to the financial industry have over the years tremendously improved in terms of scalability. Both technology stacks used in our project seem in principle to fulfill the performance requirements of our use-case and could therefore be considered as possible candidates for building productive systems.
- To answer the question whether DLT-based solutions are superior in comparison to centralized solutions one would have to undertake a full-scale cost-benefit analysis covering the whole settlement process including follow-up processes. Potential disadvantages for DLT-based solutions may be among others high latency (for some technology stacks) and high CPU-usage. Potential advantages may be among others higher resiliency and lower cost of reconciliation due to the joint data base. The importance of these aspects will differ according to the specific use-case under consideration. Further studies have to reveal for each use-case whether a DLT-based solution may be the beneficial technology.
- The new technology demands the financial industry to closely cooperate in order to reap the potential benefits in the network industry.

PLEASE NOTE