NOVEMBER 2019

FSBC Working Paper

# How Should Companies Select a Specific Blockchain Framework?

**Daniel Höfelmann, Philipp Sandner**

**This paper is intended to support decision-makers by selecting a suitable blockchain or distributed ledger technology (DLT). Ethereum, Hyperledger Fabric, R3 Corda and Quasar/Stellar were analyzed for its use in a permissioned scenario. Moreover, the frameworks were evaluated according to various points such as performance values, cost efficiency and security criteria.**

## Objective

The terms distributed ledger technology (DLT) and blockchain are often used synonymously. In this paper, four different DLTs were examined for the use in a permissioned scenario to help companies to select the right technology. The four DLTs studied are Corda, Hyperledger Fabric, Ethereum and the less popular solution Quasar. Since Quasar can be based on different infrastructures, the combination with Stellar was chosen for this consideration.

The goal of this paper is to support decision-makers and project managers by selecting a suitable DLT by illustrating the advantages and disadvantages of the various systems. These were evaluated according to the following criteria: Ease of installation, efficiency and performance, cost efficiency, release capability and „up-to-dateness", security and administration.

**Frankfurt School Blockchain Center**
www.fs-blockchain.de
contact@fs-blockchain.de

**Follow us**
www.twitter.com/fsblockchain
www.facebook.de/fsblockchain

**Frankfurt School of**
**Finance & Management gGmbH**
Adickesallee 32-34
60322 Frankfurt am MainGermany

The presented interpretations, analyses and conclusions reflect the opinions of the authors.

## Terminology and framework

In this paper, the term DLT refers to all systems that theoretically enable many or all economic actors to simultaneously and completely view transactions of all kinds. According to this definition, the blockchain is only one form of DLT, which enables the consideration of other DLTs such as Hyperledger Fabric, R3 Corda or Quasar/Stellar in addition to Ethereum.

In this paper it is assumed that the use of the reviewed DLTs takes place in a permissioned network, i.e. in an environment in which access to the system is restricted or restrictable for some or almost all actors.

If the access of a system is restricted and subject to the control of individuals or a limited number of actors, it cannot be assumed per se that the attributes of a public blockchain will be preserved (in particular the immutability of the history and the high security against many attack vectors).

These attributes, at least in the layout of the blockchain, are strongly designed with the distribution of the system by economic actors who do not trust each other and who can all join the system independently and are thus not controlled ("permissionless").

Depending on the setup of the system, permissioned systems must therefore be reviewed to see which of the attributes may be omitted or endangered, and whether an architecture such as a conventional server-client relationship makes more sense from an economic perspective.

On the other hand, the limitations of DLT systems can also be useful. For example, if subsidiaries do not trust each other, they can establish a common basis of trust by using a DLT. Such a solution can also offer advantages in supply chain management. NGOs, public administration and other institutions could use permissioned systems sensibly and to the benefit of all. It depends on the concrete design of the restrictions, control and above all steering mechanisms.

## DLTs in comparison

**Ethereum - The all-round blockchain.** Ethereum has been developed as a permissionless, public blockchain, in which every smart contract can be programmed in connection with decentralized applications (dApps). For this, a virtual machine (VM) is provided on the blockchain, for which a fee must be paid depending on the effort required to execute the programming code.

The most used programming language for Ethereum is Solidity. With Solidity smart contracts can be written and executed on the Ethereum blockchain. The programming language is based on C++, Python and JavaScript, and is implemented on the Ethereum Virtual Machine (EVM). The main advantage of Solidity lies in its comprehensibility and simple implementation.

Besides, there is also a Go implementation for writing Ethereum smart contracts. However, programs written in Go cannot be implemented directly on the EVM. Programmers need to write compilers to convert smart contracts from Go to EVM bytecode.

In the open and permissionless version of Ethereum, any developer can create an application and can interact with the blockchain. Users of dApps can also access the blockchain. Other players are miners. These create blocks in which user transactions are summarized and secured by a checksum (hash). The node operators validate these blocks and confirm them.

**Hyperledger Fabric - The framework of the Linux Foundation.** Hyperledger Fabric is a blockchain framework of the Linux Foundation. In the project "Hyperledger" several DLT infrastructures and projects of the Linux Foundation are aggregated.

Hyperledger Fabric, hereinafter referred to as Fabric, is an access-restricted ledger. Fabric's architecture is based on multiple ledgers whose operations are independent of each other. However, there is an addressing system that allows a transaction of one ledger to view and address the transactions and smart contracts of another ledger. Fabric offers an extensible and modular architecture that can be used for different areas and is therefore independent of a specific application area.

Programs in Fabric were originally referred to as chaincode, today the term smart contract is also used in this context. Fabric supports writing chaincodes in Go and Java. The chaincodes are finally executed in a docker container.

The access restriction of Fabric as well as the validating and non-validating nodes are defined by the network operator(s). Depending on requirements, the network operator can distribute different access rights to users in order to carry out the necessary transactions within the network.

Fabric's restricted character results from users' need for privacy. However, privacy does not apply to regulatory authorities, which have the possibility of identification and verification. The encryption of the identity, for example, is therefore carried out in such a way that it remains hidden from other unwanted participants, but can be viewed, for example, by regulators.

Fabric requires a cryptographic certificate prior to all transactions, which contains the confidential data of a user and is registered in the network. From each identity, the protocol can generate security keys that members can use to conduct transactions on the network. The identities of the transaction partners are hidden to ensure privacy within the network.

Within Fabric, content confidentiality is achieved by encrypting transactions so that only those involved can decrypt and execute them. Fabric offers a solution with "channels": Certain users communicate in a sub-network to which only authorized users have access. In addition, a business logic (implemented by one or more smart contracts) can also be cryptographically secured (if confidentiality is required by its stakeholders) so that it is only loaded and decrypted at a certain point in time.

**R3 Corda - The "blockchain" for the financial industry.** Corda is a global logical ledger of R3 in which all participating economic actors interact with each other. It enables the parties to securely, consistently, reliably, privately and bindingly record and manage agreements with each other. The word “global” in the global logical ledger means that each participant only sees the data that affects them. The logical part refers to the physical implementation, which can be composed differently. Unlike Fabric and Ethereum, Corda was developed exclusively for the financial industry.

The code in Corda is written with Kotlin, a JetBeans programming language based on JVM and JavaScript. Kotlin has a high level of integration, allowing the JVM to use any related programming paradigm.

The recording and processing of financial agreements involves three main points: First, the records managed by the system are made available only to those actors who have a legitimate interest in the assets and agreements they manage - similar to Fabric's channel solution. Secondly, the execution of agreements managed by the system is described in a computer code that explicitly refers to a transversal legal prose and secures the legitimacy of the agreement. And finally, parts of the system must be open (open source, open development processes and open technological industry standards) in order to achieve broad acceptance in the financial world.

The modularity and interoperability of Corda also enables companies to integrate existing systems, such as databases, into the Corda network.

**Quasar/Stellar - The consortia blockchain cash system.** Quasar is a restricted, DLT-based, electronic POS system with integrated rules. These rules serve to comply with legal and regulatory guidelines. Quasar enables instant and irreversible digital payments between businesses, individuals and devices on the Internet of Things (IoT). Quasar is based on the "multi-purpose wallet output model" developed by the company Quantoz and can be used for many applications, such as extending legacy systems for financial services.

The code in Quasar is written in C++. The use of open standards and a distributed system of nodes to which all devices can freely connect enables any developer to develop new payment applications in Quasar.

The integration of third-party wallets, tools, devices and services leads to a rapid increase in acceptance among developers and users. Third party developments are integrated through open APIs and offer system operators a fast way to develop new products.

Quasar is designed as a network of servers (nodes) at multiple locations that operate Stellar as a distributed ledger. This ledger records every activity in the system. “White-listed entities” (e.g. participating banks) can operate nodes. These nodes communicate with each other to verify transactions and synchronize the ledger. The ledger records money as credit issued by the system operator (issuing authority or bank). This system operator acts as a bridge between the traditional bank account and the Quasar network.

## The analysis

The following six categories were selected for the analysis of the four DLTs: Ease of installation, efficiency and performance, cost efficiency, release capability and up-to-dateness, security and management.

A total of 38 criteria were assigned to these categories. The four DLTs were reviewed with respect to these criteria and rated on a 5-step scale from "negative" to "neutral" to "positive". The following tables contain only the criteria and their evaluation using an extended light scale.

Table 1

### DLT in comparison: Installation - User friendliness

| Criterion | Ethereum | Hyperledger Fabric | R3 Corda | Quantoz Quasar |
|---|---|---|---|---|
| Installation | 🟢 | 🟡🟢 | 🟢 | 🟡🟢 |
| Programming languages (platform) | GoLang, C++, Java | GoLang, Java | Java | C++ |
| Programming languages (smart contracts) | Solidarity | Go, Java | Kotlin | QuBic |
| Modularity | 🔴 | 🟢 | 🟢 | 🟡 |
| Documentation | 🟢 | 🟡 | 🟡🟢 | 🟡 |

**Legend:** 🟢 Positive 🟡🟢 Rather Positive 🟡 Neutral 🟡🔴 Rather Negative 🔴 Negative ⚫ Not Evaluable

For the installing user (see Table 1) Ethereum offers a good documentation on Github, as well as numerous well documented forks and docker packages, which make the installation very easy for an experienced developer. It should be the same with Fabric and Corda, but during the analysis there seemed to be little activity with Fabric, especially regarding commits (submission of code).

Corda is only a little documented on Github. It is stated that the code should not be changed autonomously. Instead, proposals for changes should be submitted as proposals in order to maintain the quality of the platform.

Quasar is not publicly documented or it was not possible to view and review public documentation. Stellar, on the other hand, is well described on the Stellar Foundation site and on Github, so the documentation criterion was considered neutral.

Fabric scores particularly well in terms of modularity. Fabric is only one of eleven Hyperledger projects. Other modules are solutions such as "Burrow", which is intended to enable the implementation of Ethereum-based Solidity smart contracts, or "Indy", which focuses exclusively on Hyperledger with the solution of digital identities.

In Fabric the design goal of a very modular structure was pursuit from the beginning on, so that many functional levels are separated.

When considering efficiency and performance (see Table 2), it should be noted that in the case of a limited set-up, which may even be controlled by a company, some configurations, such as the proof algorithm, could be adapted in such a way that, for example Ethereum could also work more efficiently. This was taken into consideration as far as possible, but the basic time required for a consensus procedure was also taken into account. It should be pointed out that Ethereum has thought about shortening the necessary data on the blockchain at a very early stage and also implements this idea.

Table 2

### DLTs in comparison: Efficiency/Performance

| Criterion | Ethereum | Hyperledger Fabric | R3 Corda | Quantoz Quasar |
|---|---|---|---|---|
| Data storage | 🟡 | 🟢 | 🟢 | 🟡 |
| Block shortening and StateDB | 🟢 | 🟢 | 🟢 | 🟢 |
| Data volume per second | 🔴 | 🟡 | 🟡 | 🟡 |
| Multi vs. single threading (DLT) | 🔴 | 🔴 | 🔴 | 🔴 |
| Multi vs. single threading (validation) | 🟡 | 🟡 | 🔴 | 🔴 |
| Average transaction finality | 🟡 | 🟡 | 🟡 | 🟡 |
| Hardware | 🟡 | 🔴🟡 | 🟡🟢 | ⚫ |

Quasar/Stellar, Corda and Fabric are on a par with Ethereum on this point. Ethereum is only inferior in terms of possible transaction volumes. In the standard configuration with proof-of-work, there are natural limits to the number of transactions and these can also occur in other configurations. Quasar/Stellar, Corda and Fabric name higher values here, however, Corda and Fabric have documented these only in very favourably selected tests based on research and carried out only by the respective manufacturers/consortia. An audit and/or confirmation of high transaction volumes by third parties was not available for all three systems. In this respect, Quasar/Stellar, Corda and Fabric cannot be certified as positive values.

As far as transaction fees are concerned (see Table 3), all DLTs perform positively, as the fees can be freely configured for all of them. The potential maintenance costs for applications on Ethereum and Fabric are seen as rather positive. In both cases, you can choose from a large number of external developers, which has a positive influence on pricing. For Corda and Quasar/Stellar, however, it is at least unclear how mature the market of external developers with relevant experience is.

Table 3

## DLTs in comparison: Cost efficiency

| Criterion | Ethereum | Hyperledger Fabric | R3 Corda | Quantoz Quasar |
|---|---|---|---|---|
| Transaction fees | 🟢 | 🟢 | 🟢 | 🟢 |
| Maintenance costs | 🟡🟢 | 🟡🟢 | 🟡 | 🟡 |

Table 4

## DLTs in comparison: Release capability/Up-to-dateness

| Criterion | Ethereum | Hyperledger Fabric | R3 Corda | Quantoz Quasar |
|---|---|---|---|---|
| Manufacturer | 🟡🟢 | 🟡 | 🟡 | 🔴🟡 |
| Maturity level of the DLT | 🟢 | 🟡 | 🟡 | 🟡 |
| Updates | 🟢 | 🟡🟢 | 🟢 | 🟡 |
| Upgrade capability | 🟡 | 🟡🟢 | 🟢 | 🟡🟢 |
| Community | 🟢 | 🟡🟢 | 🟡 | 🔴🟡 |

The release capability and up-to-dateness (see Table 4) clearly shows that Ethereum has the most developers, the highest level of third-party review maturity, and the largest community. Quasar/Stellar, Corda and Fabric can however score with respect to upgradeability, as installing updates is easier with them than with Ethereum.

Table 5

## DLTs in comparison: Safety

| Criterion | Ethereum | Hyperledger Fabric | R3 Corda | Quantoz Quasar |
|---|---|---|---|---|
| Operating capacity | 🟢 | 🟡🟢 | 🟡🟢 | 🟡🟢 |
| Data protection for transactions | 🟢 | 🟡 | 🟡 | 🔴🟡 |
| Regulations | ⚫ | ⚫ | ⚫ | ⚫ |
| GDPR | ⚫ | ⚫ | ⚫ | ⚫ |
| Information confidentiality | 🟢 | 🟡🟢 | 🟡🟢 | 🟡 |
| Information availability | ⚫ | ⚫ | ⚫ | 🟡🟢 |
| Authenticity of Identities | 🟡 | 🟡🟢 | 🟡🟢 | 🟡🟢 |
| Authenticity of information | 🟡🟢 | 🟡🟢 | 🟡🟢 | 🟡🟢 |
| System integrity | 🔴🟡 | 🔴 | 🔴 | 🔴 |
| Data integrity | 🟡 | 🟡 | 🟡 | 🟡 |
| Partitioning attacks | 🟡🟢 | 🟡 | 🟡 | 🔴🟡 |
| Denial of Service / Distributed Denial of Service Attacks | 🟡🟢 | 🔴🟡 | 🔴🟡 | 🔴🟡 |

With respect to security, Ethereum outperforms all others in resilience due to its existence as a public system (see Table 5). Due to the open source code, the public blockchain is subject to constant investigation by attacks from third parties, especially since its total value is in the range of billions of US dollars. Consequently, non-publicly accessible systems such as Quasar/Stellar, Corda and Fabric do not have the same resilience of security aspects. Aspects of data protection - in particular those of the General Data Protection Regulation (GDPR) - remain unclear for all DLTs investigated. It has not yet been conclusively clarified whether the keys of asymmetric encryption (private/public key pairs) and hashes represent personal data. Should this be the case in the future, the use of asymmetric encryption is fundamentally challenged. It is primarily the responsibility of the developers/architects of each application to ensure compliance with the applicable privacy policies with respect to that application.

In terms of administration (see Table 6), Quasar/Stellar, Fabric and Corda are superior to Ethereum. This is mainly due to the fact that the three systems are designed per se for business infrastructures and therefore have a higher degree of interoperability and features, such as testability and logging, integrated in their architecture. Nevertheless, these features can also be integrated into Ethereum through the adaptability of a permissioned blockchain. Only the initial effort is higher or can be much higher.

Table 6

### DLTs in comparison: Administration

| Criterion | Ethereum | Hyperledger Fabric | R3 Corda | Quantoz Quasar |
|---|---|---|---|---|
| Interoperability | 🔴🟡 | 🟡🟢 | 🟡🟢 | 🟡🟢 |
| Access control | 🟡 | 🟡 | 🟡 | 🟡 |
| Customizability | 🟡🟢 | 🟡🟢 | 🟡🟢 | 🟡 |
| Third-party integration | 🟢 | 🟢 | 🟢 | 🟢 |
| User registration process | 🟡 | 🟡 | 🟡 | 🟢 |
| Testability | 🔴🟡 | 🟢 | 🟢 | 🟢 |
| Logging | 🟡 | 🟢 | 🟢 | 🟢 |

## Conclusion

Currently, Ethereum is the most widespread blockchain with a virtual machine and the DLT with the most developers, smart contracts and applications worldwide. Thus, Ethereum offers by far the most documentation and criticism by third parties.

On Ethereum, some smart contracts and their outsourced libraries have had significant security loopholes in the past (known are the hacks of the smart contract "The DAO", the "Parity Bug" and Parity's "I accidently killed it" security loophole in the library of a multisig wallet). Although these hacks do not affect the blockchain itself, they show that its complexity can overwhelm many Ethereum developers.

Overall, however, Ethereum appears to offer the highest investment security in the short and medium term: Ethereum is open source, has a large community and at the same time a high global distribution. Thus, the investment in significant business processes on a self-controlled or jointly controlled Ethereum blockchain has the highest probability to be used in a long-term perspective. Depending on the application, however, it should be payed attention to the performance factor (data volume per second).

As a further choice Fabric, Corda and Quasar/Stellar are to be seen as almost equal at the moment. An outstanding feature of Fabric is the Linux Foundation's experience in developing and maintaining large and complex open source projects. The fact that today almost all relevant Internet Service Providers operate servers with Linux distributions, and not Microsoft or servers from other providers, is a strong proof of this competence.

The evaluation of Stellar/Quasar proves to be difficult. On the one hand, the evaluation differs only marginally from the evaluation of Corda and Fabric. On the other hand, some points cannot be presented objectively and conclusively in the context of this report. This has to do both with the data situation to the system combination, and with a so far missing documentation of the performance outside a test environment.

This shows the biggest disadvantage of choosing a proprietary vendor. Only this vendor can validate issues and problems and maintain and evolve the technology as a whole. Since the Quasar underlying technology, Stellar, only works permissioned even in its public form, it cannot be seen at Fabric's and Corda's level.

The big challenge of all purely permissioned DLT solutions is the possible loss of all significant attributes, which are especially attributed to the public blockchains: Unchangeability of history, high security against many attack vectors such as Sybil and Denial of Services (DoS) attacks, and network inherent challenges such as Practical Byzantine fault tolerance. However, the actual loss or preservation of these attributes in a system is only possible in concrete and highly detailed individual cases. In fact, these attributes are not automatically given for public blockchains, but have to be checked in a case-by-case analysis.

If essential attributes are lost, the question arises why not solutions should be used that are well designed and technologically fully tested. This could be a centrally controlled platform with well thought-out role/right principles, with probably lower costs and higher availability of developers and system architects, and higher overall understanding across the organization.

The consideration of a concrete and detailed described target system is therefore very individual and only possible if all relevant aspects are known. Furthermore, it is recommended to observe future developments of the Ethereum Enterprise Alliance and the R3 Corda Consortium as well as concrete use cases of Hyperledger Fabric and Quasar/Stellar and to consider them in the decision-making process.

**Prof. Dr. Philipp Sandner** is head of the Frankfurt School Blockchain Center. You can contact him via mail (email@philipp-sandner.de), via LinkedIn (https://www.linkedin.com/in/philippsandner/) or follow him on Twitter (@philippsandner).

**Daniel Höfelmann**, alumnus of the Frankfurt School of Finance & Management, is Director Innovation Management at Aareal Bank in Wiesbaden. You can contact him via mail (daniel.hoefelmann@aareal-bank.com), via LinkedIn (https://www.linkedin.com/in/danielhoefelmann/) or follow him on Twitter (@danielhoefelman).