



HYPERLEDGER

EB 20/21	Enterprise Blockchain Technologies	Number:	4
Module I - Introduction		Issue Date:	-
Background: Introduction to Hyperledger Fabric		Due Date:	-

Preliminary Notes

We now introduce the Hyperledger Fabric blockchain, an enterprise-grade permissioned distributed ledger platform that offers modularity and versatility for a broad set of industry use cases. This laboratory is based on several sources [2, 3].

1 Hyperledger Ecosystem

Hyperledger is an open-source community. Hyperledger stewards the development of frameworks, tools, and libraries for enterprise-grade blockchain deployments. It hosts Hyperledger Fabric, Sawtooth, Indy, and tools like Hyperledger Caliper and Hyperledger Explorer. There are also tools hosted at Hyperledger Labs¹, an open collaboration space made by blockchain enthusiasts, that can benefit the Hyperledger ecosystem. An example is Hyperledger Umbra², allowing to run Hyperledger blockchains under Mininet for the purposes of testing scalability and consensus mechanisms.

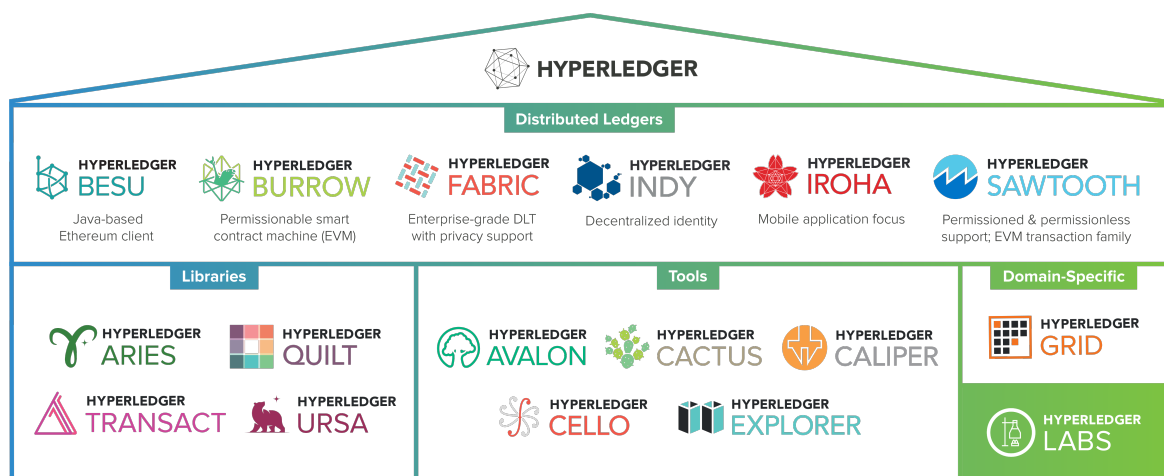


Figure 1: Hyperledger Greenhouse

“Hyperledger launched in 2016 with a technical and organizational governance structure and 30 founding corporate members. Initially, the Hyperledger Technical Steering Committee welcomed two business blockchain framework codebases into incubation: Hyperledger Fabric, a codebase combining work by Digital Asset, libconsensus from Blockstream, and OpenBlockchain from IBM; and Hyperledger Sawtooth, developed at Intel’s incubation group. Of the 70+ open source organizations the Linux Foundation has launched, Hyperledger holds the distinction as the fastest-growing”³.

¹<https://www.hyperledger.org/blog/2018/01/23/introducing-hyperledger-labs>

²<https://github.com/hyperledger-labs/umbra>

³<https://www.hyperledger.org/about>

EB 20/21	Enterprise Blockchain Technologies	Number:	4
Module I - Introduction		Issue Date:	-
Background: Introduction to Hyperledger Fabric		Due Date:	-

2 Hyperledger Fabric Blockchain

*Hyperledger Fabric (Fabric)*⁴ [2] is a private blockchain framework, backed by the Linux Foundation and IBM, which is the main study object of this course. Hyperledger Fabric is a successful open-source project founded in 2016, counting with more than 13k commits, more than 250 contributors, and 10k stars in Github.⁵

Learning all about Hyperledger Fabric can open multiple windows of career prospects. Some of the top jobs trending in the United States of America are that of a Hyperledger Developer, Blockchain Engineer, Blockchain Solutions Architect, Project Manager, and jobs associated with web development (blockchain clients). “As a qualified Hyperledger developer with good work experience, your salary package can fall anywhere between \$122–179k. But if you’re just starting out, it might vary between \$80–110k”⁶.

Hyperledger Fabric is also greatly popular in academia. A search on Google Scholar with the terms “Hyperledger Fabric” returned 6,440 results (August 2020). Only in 2020, 1,710 papers were written.

Hyperledger Fabric is then a great resource to initiate your venture into the blockchain world and learn about cryptography and distributed systems.

2.1 A Technical Viewpoint on Fabric

Along this section, we introduce Fabric and its components, how they are related, and how they work as a whole. Please be mindful of the contents learned during Lab1, Lab2, and Lab3.

2.2 Consensus

Organizations group *nodes* (or *peers*, or *peer nodes*) who form independent trust domains, i.e., peers from the same organization trust each other. Several organizations can create a *consortium* – a group of non-trusting organizations that want to achieve a business goal.

Fabric allows for different kinds of participants in the network, which facilitates the *execute-order-validate* paradigm for distributed execution of chaincode. Endorsement peers (*endorsers*) execute (*endorse*) the smart contracts (*chaincode*) and return blockchain clients the validation output of submitted transactions, which contain the endorsement peers’ signatures. This process allows parallel execution and addresses non-deterministic code.

To achieve consensus, Fabric uses a permissioned voting-based scheme, which achieves low-latency. The *endorsement policy* defines the voting-based scheme to be used by peers and, consequently, the weight of each peer regarding the validity of a

⁴<https://www.hyperledger.org/use/fabric>, <https://hyperledger-fabric.readthedocs.io/en/release-2.2/>

⁵<https://github.com/hyperledger/fabric>

⁶<https://medium.com/the-capital/how-to-become-a-hyperledger-developer-in-2020-bb19151e2a8>

EB 20/21	Enterprise Blockchain Technologies	Number:	4
Module I - Introduction		Issue Date:	-
Background: Introduction to Hyperledger Fabric		Due Date:	-

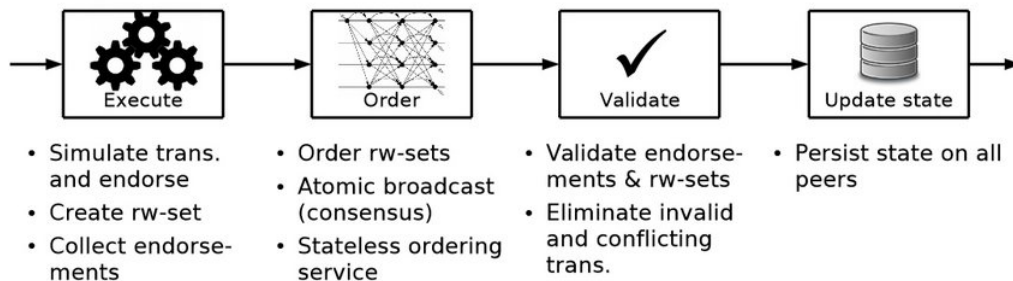


Figure 2: Execute-Order-Validate paradigm [?]

transaction. In essence, the endorsement policies gather approvals for the execution of each transaction.

Apart from the consensus, Fabric needs to order blocks of transactions proposed by different peers. The consensus on the block order is pluggable, i.e., using Kafka's Zookeeper (in previous 1.x.x Fabric versions) or RAFT (as in 2.x.x), both crash fault-tolerant consensus mechanisms. Pluggability allows the developer to adapt the consensus to the requirements of the environment. The consensus algorithms are currently implemented to achieve consensus deterministically. This way, forks as in the Bitcoin network are prevented. Fabric can utilize different consensus protocols that do not require a native cryptocurrency, which reduces attack vectors and performance issues due to expensive operations, for instance, the ones required by proof of work. Hyperledger Fabric added a novel way to deal with privacy and confidentiality through its *channel* architecture and private data feature⁷.

In conclusion, the consensus in Fabric is not limited to the agreed-upon order of a batch of transactions between peers. Rather, it is also the result of the execute-order-validate process during a transaction's flow from its proposal to its commitment to each peer's ledger.

2.3 Chaincode

Blockchain developers can write chaincode using a mainstream programming language, such as Javascript, GoLang, or Java. Chaincode allows for implementing various use cases. For instance, although Fabric does not have a built-in cryptocurrency, it is possible to create an underlying token with chaincode, representing assets or rights to perform specific actions. Such assets can be exchanged between network participants through transactions.

Chaincode is a critical element of a Fabric network, as it dictates the rules to be followed by member participants. It is run in Docker containers and is, thereby, isolated from the shared ledger.

⁷<https://hyperledger-fabric.readthedocs.io/en/release-1.4/private-data/private-data.html>

EB 20/21	Enterprise Blockchain Technologies	Number:	4
Module I - Introduction		Issue Date:	-
Background: Introduction to Hyperledger Fabric		Due Date:	-



Figure 3: Hyperledger Fabric

There are two types of chaincode: *application chaincode*, that executes the application logic and communicates with the peers using *gRPC messages*, and *system chaincode*, ran on the *configuration channel*. System chaincode assures chaincode execution correction (for example, that the endorsement policies are respected). Chaincode can be deployed dynamically, and it is usually running concurrently on the network. It runs directly on the peers' processes. The configuration channel stores the definition of MSPs (membership service providers), configuration about the consensus, ordering service parameters, and rules on how the channel configuration is tweakable. Chaincode executes transaction proposals against world state data. The world state is a concept that represents the latest value for a key stored by a blockchain. By providing direct access to these keys' latest value, there is no need to traverse the entire transaction log and calculate its values. Each peer contains a ledger component, formed by the *block store*, which stores blocks containing transactions and the *peer transaction manager (PTM)*. There is a different ledger for each channel, as channels enforce chaincode and data isolation.

2.4 Peer nodes

Fabric defines on its model several kinds of peer nodes:

- *Committing peer (or simply peer)*. Each peer maintains the current snapshot of the current state of the ledger as a key-value store. Such peers cannot invoke chaincode functions.
- *Endorser peer*. Endorser peers have chaincode installed. When they receive a transaction proposal, they simulate the transaction execution on isolated containers. Based on that simulation, such peers prepare a transaction proposal sent to the orderer peer. The existence of endorser peers avoids sequential execution of transactions by all peers.
- *Orderer peer*. Orderers receive endorsed transactions and assemble them into blocks. After grouping transactions, orderers assure consensus by propagating such blocks to committing peers. They are validated and then committed to the shared ledger. Orderer peers record valid and invalid transactions, while other peers only contain valid transactions.

Additionally, Fabric defines *anchor peers* and *leader peers*. Anchor peers serve as an intermediary between peers from its organization and peers from an external one.

EB 20/21	Enterprise Blockchain Technologies	Number:	4
Module I - Introduction		Issue Date:	-
Background: Introduction to Hyperledger Fabric		Due Date:	-

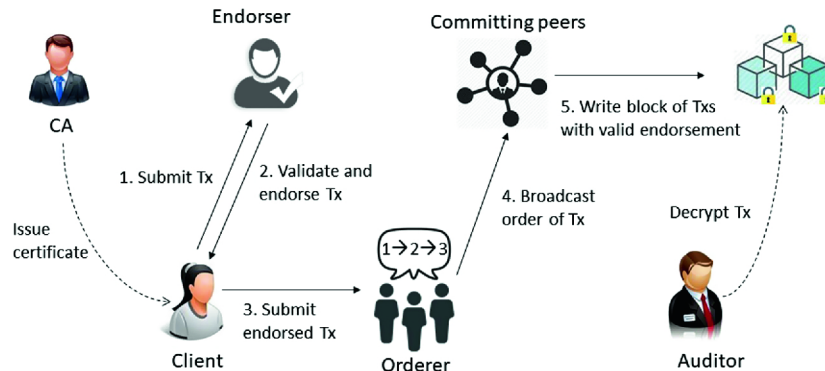


Figure 4: Example of each node role [5]

Leader peers take the responsibility of distributing the transactions from the orderer to committing peers.

2.5 Channels

Channels allow participants to establish a communication path between the subset of participants who can visualize a subset of transactions. For instance, there can be a subset of peers in the same network that have access to only a certain kind of transaction. In addition to channels, Fabric supports *private data*, which allows a defined subset of organizations on a channel to isolate their data from others. For example, when several traders engage in a blockchain ecosystem, it might be desirable for a seller to hide the price of their items from buyers⁸. In specific, organizations with permissions can endorse, commit, or query private data, which is logically separated from channel ledger data. In case of a dispute, private data can be shared and shown. For further privacy, hashes of private data go through the orderer instead of the data itself. The orderer does not access the transactional data to order it into blocks - but only its hash. It is disseminated peer-to-peer rather than via blocks. When transaction data must be kept confidential from ordering service nodes, it can use private data collections rather than channels.

⁸<https://hyperledger-fabric.readthedocs.io/en/release-2.2/private-data/private-data.html>

EB 20/21	Enterprise Blockchain Technologies	Number:	4
Module I - Introduction		Issue Date:	-
Background: Introduction to Hyperledger Fabric		Due Date:	-

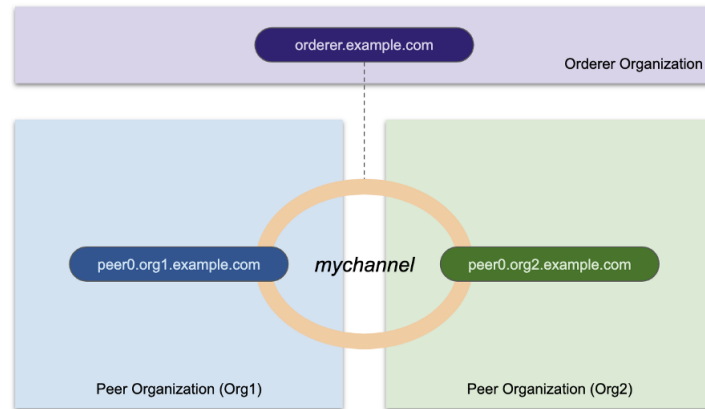


Figure 5: A simple Fabric network

For example, Figure 5⁹ depicts a simple Fabric network compressed on two orgs with one peer each, connected by one channel. One orderer orders the transactions approved by both orgs.

2.6 Transaction Flow

The transaction flow, which follows the execute-order-validate paradigm, is depicted in Figure 7, is as it follows:

- *Transaction proposal.* A blockchain client, which represents an organization, creates a transaction proposal and sends it to endorsement peers, as defined in the endorsement policy. The proposal contains information regarding the proposer's identity, the transaction payload, a nonce, and a transaction identifier.
- *Execute (endorsement):* the endorsement consists in the simulation of the transaction. The endorsers produce a write-set, containing the keys and their modified values, and a read-set. The endorsement peers execute the transactions in an isolated environment. The endorsement is sent as the proposal response. It contains the write-set, read-set, the transaction ID, endorser's ID, and the endorser's signature. When the client collects enough endorsements (which need to have the same execution result), it creates the transaction and sends it to the ordering service. The endorsement phase eliminates any eventual non-determinism.
- *Order:* after the endorsement, there is the ordering phase, performed by orderers. The ordering service checks if the blockchain client that submitted the transaction proposal has appropriate permissions (broadcast and receiving permissions) on a given channel. Ordering produces blocks containing endorsed transactions, in

⁹<https://medium.com/@kctheservant/test-network-script-walk-through-95ca973bc676>

EB 20/21	Enterprise Blockchain Technologies	Number:	4
Module I - Introduction		Issue Date:	-
Background: Introduction to Hyperledger Fabric		Due Date:	-

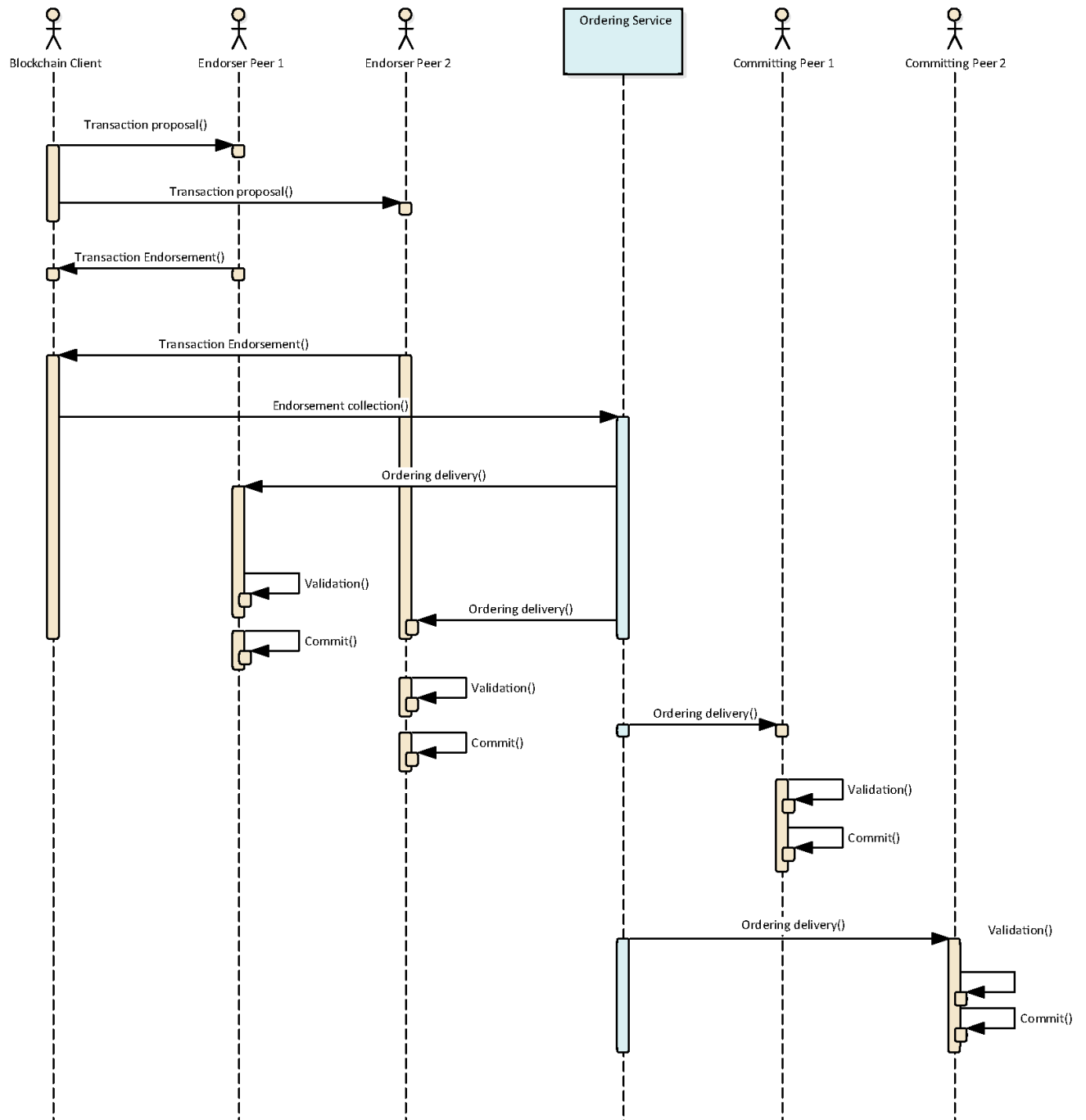


Figure 6: Transaction Flow [2]

EB 20/21	Enterprise Blockchain Technologies	Number:	4
Module I - Introduction		Issue Date:	-
Background: Introduction to Hyperledger Fabric		Due Date:	-

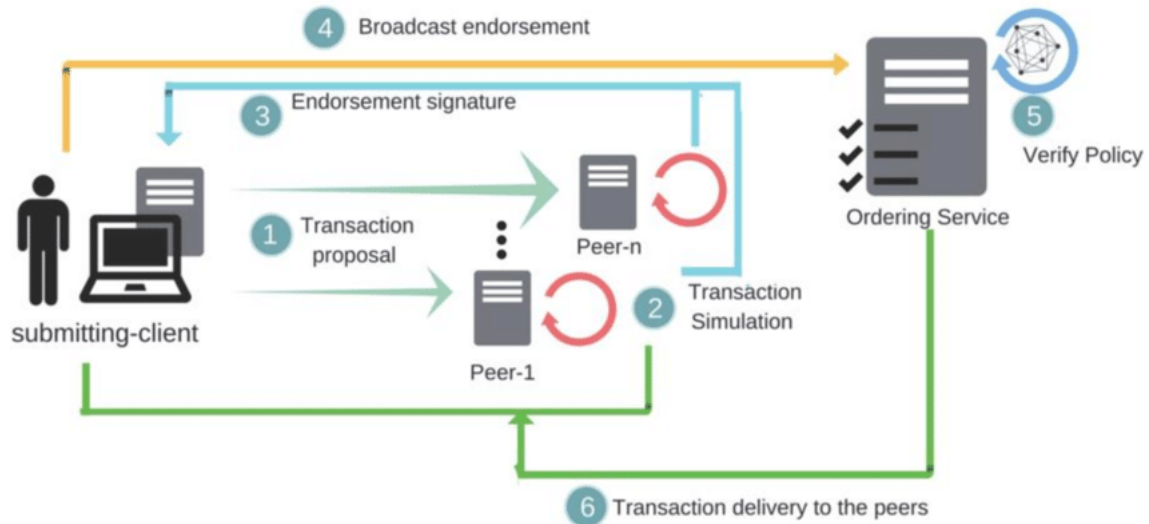


Figure 7: Transaction Flow [?]

an ordered sequence, per channel. The ordering allows the network to achieve consensus. The orderer broadcasts the transaction's outputs to all the peers. For correct ordering, there are some properties that the system must comply with, defined as:

Definition 1. Hash chain integrity: For any two blocks, B delivered with sequence number s , and B' delivered with s' at correct peers such that $s = s'$, it holds $B = B'$.

Definition 2. Hash chain integrity: If some correct peer delivers a block B with number s and another correct peer delivers block $B' = ([tx_1, \dots, tx_k], h')$ with number $s+1$, then it holds $h' = H(B)$, where $H(\cdot)$ denotes the cryptographic hash function.

Definition 3. No skipping: if a correct peer p delivers a block with number $s > 0$ then peer p has already delivered all blocks with number $i \leq s$.

Definition 4. No creation: When a correct peer delivers block B with number s , then for every $tx \in B$, some client has already broadcast tx .

The ordering step assures all the above properties for each channel. The ordering service broadcasts blocks to the peers that maintain the ledgers' state via the ordering service or gossip protocol.

Definition 5. Validity: If a correct client invokes *broadcast(tx)*, then every correct peer eventually delivers a block B that includes tx , with some sequence number.

There is the need to note that the ordering service aims to achieve consensus and prevents the network from forks. Consensus in Fabric encompasses the whole

EB 20/21	Enterprise Blockchain Technologies	Number:	4
Module I - Introduction		Issue Date:	-
Background: Introduction to Hyperledger Fabric		Due Date:	-

transaction flow, from the transaction proposal to the committing. It happens at the transaction level, where not all nodes need to engage in the consensus mechanism. Channels ensure that messages are delivered in the same logical order to committing peers. The ordering service achieves consensus in a deterministic way. As the ordering is deterministic and not probabilistic, as in Bitcoin, forks do not occur. Permissioned blockchains that rely on the BFT replication protocols to achieve consensus can only support f faulty nodes out of $3f+1$ nodes. This assumption may not match the trust model idealized for a specific application. The endorsement policy establishes the trust model, decoupled from the consensus mechanism, allowing developers to reason about the trust model independently of the consensus algorithm.

- *Validate.* Firstly, each peer validates the received transactions by checking if a transaction follows the corresponding endorsement policy. After that, a read-write conflict check is run against all transactions in the block sequentially. It compares the versions of the keys in the read-set with those currently on the ledger for each transaction. In case they do not match, the peers discard the transaction. Finally, the ledger is updated, in which the ledger appends the created block to its head. The ledger appends the results of the validity checks, including the invalid transactions.

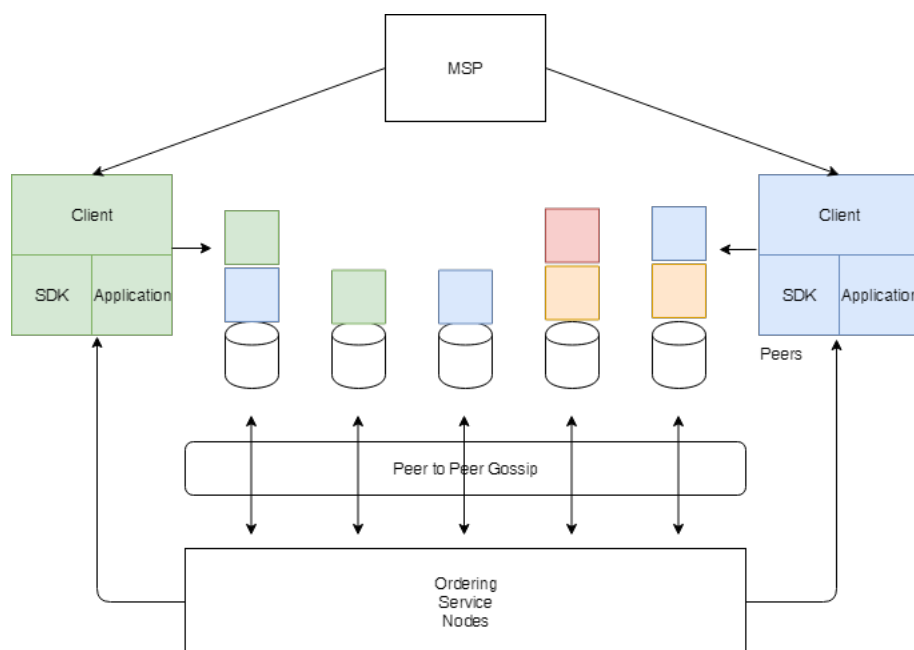


Figure 8: Example of a Fabric's network

Fabric has the following architectural components that support the transaction lifecycle:

EB 20/21	Enterprise Blockchain Technologies	Number:	4
Module I - Introduction		Issue Date:	-
Background: Introduction to Hyperledger Fabric		Due Date:	-

- *Membership Service Provider*: This service aims to identify participants (authentication) in the network uniquely. Public key infrastructure (*PKI*) is used to generate certificates that are tied to members and organizations. Each organization issues identities to its members, and every peer recognizes the members of the organization. The identification is made by associating a given peer to a cryptographic entity (i.e., digital certificate). Different identity management protocols can manage the identity of participants, such as LDAP and OpenID.
- *Ordering Service*: it manages multiple channels and comprises orderers and their processes. The batching process occurs in this phase, in which validated transactions are received and grouped into blocks. After the transaction validation by the orderer, it broadcasts state updates to the ledger, in an atomic way, following a consensus algorithm for each channel. The ordering service can reconfigure channels and restrict the broadcasting of transactions if needed.
- *Peer Gossip*: the peer gossip is responsible for broadcasting the results of the ordering phase, as well as *transfer state* for unsynched peers (recently joined, after a downtime, or a peer is slower at validating the blocks before committing). Gossip data dissemination helps to achieve consistency and data integrity across nodes. Since blocks are signed and numbered, it can reconstruct the blockchain and verify its integrity after a peer receives them. Thus, the gossip protocol manages peer discovery and channel membership and disseminates ledger data across peers on a channel.

2.7 Replication Model

Fabric introduces a hybrid replication model, combining active and passive replication (primary-backup-replication, ported to the untrusted environment). Concerning active replication or state machine replication, the ledger state only reflects the transactions after they are validated. The consensus is reached concerning their ordering. Passive replication happens as endorsers send the result of the transaction processing to commit nodes. Fabric comprises three main elements around data: *world state*, a versioned key-value store which corresponds to the distributed ledger; the *transaction log*, stores the history of all transactions (PTM); and a NoSQL database, such as CouchDB, stores the world state. It is possible to restrict users' access to view and edit specific fields and only authorizing the read-only permissions. CouchDB supports complex data queries, comparatively to LevelDB, against the whole blockchain data, making it a suitable solution for data analysis and auditing. *LevelDB* is the other built-in option for storing the world state. It is a simple, fast key-value storage library that provides an ordered mapping from string keys to string values.

EB 20/21	Enterprise Blockchain Technologies	Number:	4
Module I - Introduction		Issue Date:	-
Background: Introduction to Hyperledger Fabric		Due Date:	-

3 Blockchain4Students QUC - Hyperledger Fabric Version

Education is a sector that may benefit greatly from the blockchain. Today, most blockchain-based applications for education are directed towards supporting academic degree management and evaluation for learning outcomes, explored in projects such as QualiChain¹⁰. QualiChain is a project funded by the EC under the Horizon 2020 program. The project aims to propose a blockchain-based approach for disrupting the archiving, management, and verification of educational and employment qualifications. In particular, QualiChain will support the storage, sharing, and verification of academic and other qualifications and several additional services provided by the platform.

Besides credential management, blockchain can streamline formative and qualitative evaluation processes by tracking teaching and learning details in a transparent way. In particular, formal education produces a lot of data, coming from several parties: student records, in-class data (attendance, grades), course data, payments, diplomas, and so on.

We now introduce a use case for education, *Transparent Curricular Unit Quality System*, that serves as the basis for this course. This system is designed to improve universities' teaching conditions by enforcing accountability and transparency relatively to students' feedback on courses and its teaching staff.

3.1 Transparent Curricular Unit Quality System

Técnico Lisboa is a Portuguese public school of engineering and technology, part of the University of Lisbon. Unrelativef Lisbon promotes continuous improvement via questionnaires and the Course Unit Quality System (QUC) [1].

Regarding questionnaires, Figures 11 depicts a questionnaire answered by 36 Ph.D. students on their experience at the same university. After processing, the questionnaires yield a list of action items aiming to improve the processes' current status: propose a process that evaluates the quality of the theses' advising. This could be done via the QUCs.

QUCs aim to follow up each course unit's functioning by promoting responsible involvement of students and teachers in the teaching, learning, and assessment process. Students classify courses (and professors) on a scale from 1 to 9. The overall classification for both courses and professors is yielded. If professors or courses have low classifications, an audit can occur.

At the end of each semester, and for each Curricular Unit under an audit process, the analysis of the process will be presented with the main conclusions, the suggested correction, and implemented methods.

¹⁰<https://qualichain-project.eu/>

EB 20/21	Enterprise Blockchain Technologies	Number:	4
Module I - Introduction		Issue Date:	-
Background: Introduction to Hyperledger Fabric		Due Date:	-



INQUÉRITO ESTUDANTES DE DOUTORAMENTO

2019

Conselho Pedagógico do Técnico Lisboa (IST)

Inquérito aplicado entre 16.10.2019 e 15.03.2020

Caraterização dos estudantes que terminaram o Doutoramento em 2019

	Nº	Nº respostas ⁽²⁾	
		Nº previsto	"Sim" "Não"
Diplomas ⁽¹⁾	146		
Participantes ⁽³⁾		36	34 2

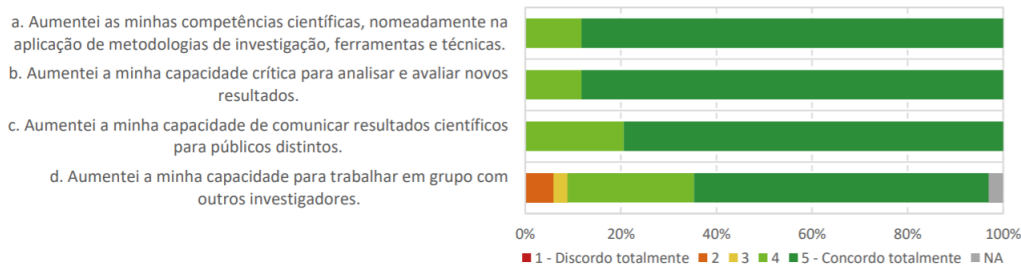
(1) Alunos que concluíram o Doutoramento no ano 2019.

(2) Os inquiridos podem optar por responder ou não ao inquérito completo, justificando-se.

(3) Por tratar-se da primeira edição, o inquérito não ficou disponível a todos os finalistas de 2018/19.

Figure 9: Questionnaire to Técnico's Lisboa PhD students (in Portuguese): number of participants

6) Desenvolvimento de competências científicas



7) Gestão do trabalho e bem-estar

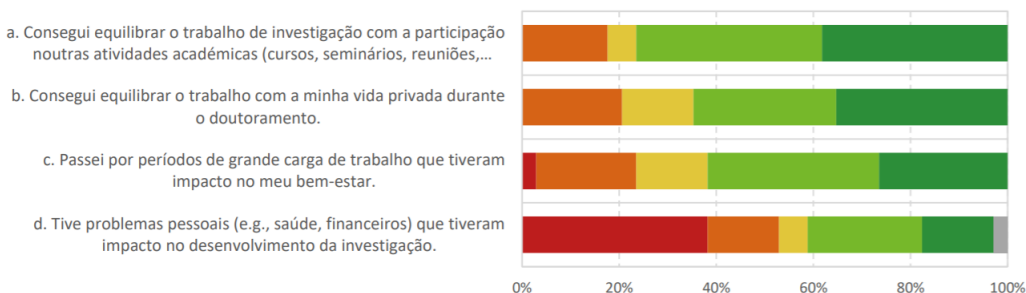


Figure 10: Questionnaire to Técnico's Lisboa PhD students (in Portuguese): development of scientific skills, and work and well-being management

EB 20/21	Enterprise Blockchain Technologies	Number:	4
Module I - Introduction		Issue Date:	-
Background: Introduction to Hyperledger Fabric		Due Date:	-

QUC - UC Auditadas / Observadas

Departamento de Engenharia Informática

Inteligência Artificial





































Curso	2007/08 2º S	2008/09 1º S 2º S	2009/10 1º S 2º S	2010/11 1º S 2º S	2011/12 1º S 2º S	2012/13 1º S 2º S	2013/14 1º S 2º S	2014/15 1º S 2º S	2015/16 1º S 2º S	2016/17 1º S 2º S	2017/18 1º S 2º S	2018/19 1º S 2º S
LEIC-T												
MEIC-A												
LEIC-A												

Figure 11: Audit timeline to the Artificial Intelligence course in the masters (MEIC) and bachelor (LEIC). The yellow square indicates that the course had a bad performance on QUC scores. The red box indicates that the course was audited.

A professor or course should improve if their QUC median results are lower than 5. A professor is considered excellent if the median of their assessment factors is 9. Figure 12 depicts some questions appearing on the QUC questionnaires.

This system aggregates feedback from both students and professors on the functioning of course units, providing a holistic view of universities, professors, and students' performance. Students and professors are required to fill QUCs at the end of each semester. When a course or professor has a low classification, derived from students' feedback, the university can intervene (audits/corrective measures). Such a system allows for continuous improvement, leading to a better educational system.

However, some problems arise: first of all, not all universities employ such a system. As such, bad teaching consequences are often left unresolved, leading to a loss of human potential. Secondly, universities can tamper/hide QUC results, as they are the centralized party processing them. Having a common, global, QUC system that can make universities comparable in terms of course unit quality needs to be decentralized. If students directly vote on such a platform, maintained by a consortium of high education institutions (such as Qualichain), the resiliency to the data dramatically increases. Finally, the criteria and scoring result may not be transparent for universities to manipulate the perception about their quality. Therefore, there is a need for a standardized system that is clear, transparent, and simple.

4 Blockchain4Students QUC Implementation

To address those needs, we leverage the B4S-QUC system, based on Hyperledger Fabric. This system contains three stakeholders: universities, professors, and students.

EB 20/21	Enterprise Blockchain Technologies	Number:	4
Module I - Introduction		Issue Date:	-
Background: Introduction to Hyperledger Fabric		Due Date:	-

1.3 Caracterização do nível de importância que atribui aos meios de estudos, quando utilizados, nesta UC:	Não se aplica	Nada importante 1	2	3	4	5	6	7	8	Muito importante 9
1.3.1 Assistir às aulas teóricas/seminário *	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1.3.2 Assistir às aulas de problemas *	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1.3.3 Assistir às aulas de laboratório *	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1.3.4 Bibliografia sugerida *	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1.3.5 Apontamentos e outros documentos do professor *	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1.3.6 Apontamentos e outros documentos do aluno *	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1.3.7 Outra informação acessível publicamente *	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figure 12: Example of a section in a QUC questionnaire (in Portuguese). There are questions regarding the importance of different study materials in a specific course.

- Students vote on the parameters of teaching quality and course organization via their local student's group or association.
- Professors give feedback on the course they taught via the professors' department.
- Universities analyze the results and apply corrective measures to courses.

Feedback is collected from different questions, called *feedback items*. The result of a set of questions towards a specific course is called a *feedback entry*. Several feedback entries on a specific course compose the *course feedback*, which is later analyzed to produce the QUC score (1-9). The score applies to the course and to the teaching staff of that course.

Let us think about possible implementation. B4S QUC should implement a Transparent Curricular Unit Quality System. Students give their feedback on courses/professors of their university through a student group. Schools and universities are rewarded reputation tokens based on the quality of their students' feedback. This allows making professors directly accountable for their students, yielding incentives for the universities to support their improvement. However, some issues must be dealt with.

4.1 Exercises: B4S QUC

Consider the Fabric version of Blockchain4Students - a blockchain dedicated to improving continuous improvement in education.

EB 20/21	Enterprise Blockchain Technologies	Number:	4
Module I - Introduction		Issue Date:	-
Background: Introduction to Hyperledger Fabric		Due Date:	-

4.1.1 Threats to Privacy

Exercise 1: Can a centralized system implement the use case?

Exercise 2: Which privacy concerns arise regarding this network?

Exercise 3: How can one assure privacy and confidentiality for students and professors, without sacrificing traceability?

Exercise 4: What would a suitable endorsement policy be, for the studied ecosystem?

4.1.2 Data Integrity Threats

To mitigate data integrity threats, it is important to perform an analysis of current challenges and risks. We provide a general threat model concerning data integrity in information systems and data integrity threats on the blockchain. Threats related to the blockchain are based on our previous work [4].

This section presents the threats to which data is exposed (in this case, the 'feedback entries'). The threat model will focus on three factors: i) internal access attacks ii) remote attacks, and iii) Hyperledger Fabric blockchain attacks.

We assume that there are two types of adversaries. A third-party adversary, *Adversary A_t* , can access the trusted computing base and remotely penetrate the system by exploiting security flaws or by hijacking root user credentials. Stemming from this, A_t might tamper data to compromise auditing and forensic procedures or tamper system data to damage the organization. On the other hand, an adversary from within the organization, *Adversary A_o* , can be an employee with root privileges. Such a person can access the databases and can be motivated to tamper with data for personal gains. Such adversaries can tamper with data to hide their traces or tamper data to aid third-parties illegally.

There can be two main types of attacks to data integrity: the *physical access attack* and the *remote vulnerability attack*. In the physical access attack, the adversary *Adversary A_t* or *Adversary A_o* has access to the critical system components. The attacker generates transactions to change the current values of the objects held in a database. As objects are being updated, the database generates an audit log, tracking the attacker's changes. The attacker then focuses on deleting the evidence by deleting the generated audit logs or changing its content. The attacker can manipulate the auditing process by modifying the history maintained by the audit log.

As a consequence, the obfuscation of illegal activities and impersonating someone's actions can occur. In remote vulnerability attacks, the attacker exploits default vulnerabilities on systems, such as software malfunctions and security vulnerabilities (e.g., SQL-injection). Although such attacks are common, corporate organizations have their systems and databases secure against conventional attacks. The research is focused on physical access attacks. Five threats compose the threat model [4], which are distributed through the various exercises:

EB 20/21	Enterprise Blockchain Technologies	Number:	4
Module I - Introduction		Issue Date:	-
Background: Introduction to Hyperledger Fabric		Due Date:	-

Exercise 5: Elaborate on the following threats:

Threat 1. Feedback item tampering from an external element: An attacker violates the integrity of the 'feedback entries by editing them.

Threat 2. Feedback item tampering from an internal adversary: An attacker from one of the stakeholders violates the integrity of the 'feedback entries by editing them directly.

Threat 3. Feedback item tampering by the system administrator: The administrator of the system, with the highest permissions, violates the integrity of the 'feedback entries by editing them. There is the possibility of obfuscating activity traces by deleting evidence on other systems

Threat 4. A participant edits feedback entries that are protected by the blockchain.

Threat 5. The majority of participants conspire and modify the 'feedback entries.

Exercise 6: How to alleviate the mentioned threats?

References

- [1] QUC System — Qualidade das Unidades Curriculares • QUC.
- [2] E. Androulaki, A. Barger, V. Bortnikov, S. Muralidharan, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Murthy, C. Ferris, G. Laventman, Y. Manevich, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, S. W. Cocco, and J. Yellick. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. In *Proceedings of the 13th EuroSys Conference, EuroSys 2018*, pages 1–15, New York, New York, USA, apr 2018. Association for Computing Machinery, Inc.
- [3] R. Belchior. *JusticeChain: Using Blockchain To Protect Justice Data*. PhD thesis, Instituto Superior Técnico, 2019.
- [4] R. Belchior, A. Vasconcelos, and M. Correia. Towards Secure, Decentralized, and Automatic Audits with Blockchain. In *European Conference on Information Systems*, 2020.
- [5] T. H. Yuen. PACHain: Private, Authenticated and Auditable Consortium Blockchain. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11829 LNCS, pages 214–234. Springer, oct 2019.