Technical Architecture

Team Id	NM2023TMID04410
Project Name	Project-Drug Traceability

Introduction:

- Pharmaceutical supply chain (PSC) consists of multiple stakeholders including raw material suppliers, manufacturers, distributors, regulatory authorities, pharmacies, hospitals, and patients.
- The complexity of product and transaction flows in PSC requires an effective traceability system to determine the current and all previous product ownerships. In addition, digitizing track and trace process provides significant benefit for regulatory oversight and ensures product safety.
- Blockchain-based drug traceability offers a potential solution to create a distributed shared data platform for an immutable, trustworthy, accountable and transparent system in the PSC.
- We represent an overview of product traceability issues in the PSC and envisage how blockchain technology can provide effective provenance, track and trace solution to mitigate counterfeit medications.
- We propose two potential blockchain based decentralized architectures, Hyperledger Fabric and Besu to meet critical requirements for drug traceability such as privacy, trust, transparency, security, authorization and authentication, and scalability.
- We propose, discuss, and compare two potential blockchain architectures for drug traceability.
- We identify and discuss several open research challenges related to the application of blockchain technology for drug traceability.
- The proposed blockchain architectures provide a valuable roadmap for Health Informatics researchers to build and deploy an end-to-end solution for the pharmaceutical industry.

Blockchain based architectures for drug traceability:

- In this section, we present and discuss two blockchain-based architectures to fulfil important requirements for drug traceability.
- The proposed architectures are based on two blockchain platforms namely, Hyperledger Fabric and Hyperledger Besu as they provide higher degree of trust, decentralization, transparency, privacy, security, data integrity, deployment, modularity and scalability when compared to other blockchain platforms such as Ethereum, Quorum, BigChain, etc.
- These architectures can be key enablers for creating private permissioned blockchain ecosystems where pharmaceutical stakeholders and their end-users are registered, controlled, and regulated by a regulating authority or a group of authorities/stakeholders.
- The two proposed architectures and their respective transaction flows are described in the following subsections, followed by in-depth technical comparison.

Hyperledger Fabric architecture:

- Hyperledger Fabric is a platform providing distributed ledger solutions, underpinned by a modular architecture delivering high degrees of confidentiality, resiliency, flexibility and scalability.
- It is an enterprise grade DLT based on blockchain technology that uses smart contracts to enforce trust between multiple parties. Hyperledger Fabric eliminates the concept of mining, but still keeps the good properties of a typical cryptocurrency blockchain (such as Bitcoin, Ethereum) like: block immutability, order of events determinism, prevention of double spending, etc.
- Hyperledger Fabric has been confirmed to offer superior transaction throughput, up to several thousand transactions per second.
- These characteristics, among other that will be described below, make Hyperledger Fabric a perfectly suitable candidate for complex supply chain systems with multiple physical and logical processes and parties.
- By using general purpose programing languages (Java, Go, NodeJS) to develop smart contracts, the adoption bar for this technology is lower than for others using dedicated programing languages (e.g. Solidity in Ethereum).
- The Hyperledger Fabric drug traceability architecture proposed in this paper provides an initial design of an enterprise-level blockchain-based supply chain system, where different stakeholders in the pharmaceutical supply chain are identified, their relationships established using different channels to provide maximum privacy, confidentiality, and data security.

- A concept of channels is unique to Hyperledger Fabric. Channels offer clear separation of business logic and data privacy policies between different stakeholders operating in the same system.
- By default, Hyperledger Fabric provides a secure and transparent crash –fault tolerant transaction ordering for ensuring deterministic recording of events, secure communication and reliable exchange of medication related transactions amongst a group of untrusted stakeholders.
- This helps to create a consistent track-and-trace provenance system to ward off counterfeit medications in PSC.
- The proposed blockchain architecture introduces a new modular approach to provide high levels of flexibility, resiliency, scalability, and privacy.
- Finally, at the core of the Hyperledger Fabric architecture there are Peer nodes (peers) and the Ordering Service (OS). Peers store ledger copies, execute smart contracts (also referred to as chaincode in Hyperledger Fabric), endorse, and commit transactions.
- The OS accepts the endorsed transactions from client applications, orders them into blocks with cryptographic signatures of the endorsing peers, and finally broadcasts these blocks to the committing peers in the blockchain network for validations against the endorsement policies.

Drug traceability flow with Hyperledger Fabric:

- In this section, we describe how transactions in the pharmaceutical supply chain are executed and communicated between different stakeholders using the execute-order-validate transaction processing methodology typical for Hyperledger Fabric.
- This is shown in Figure 2. The steps taken to complete a transaction processing cycle in this architecture are described in detail and numbered below.

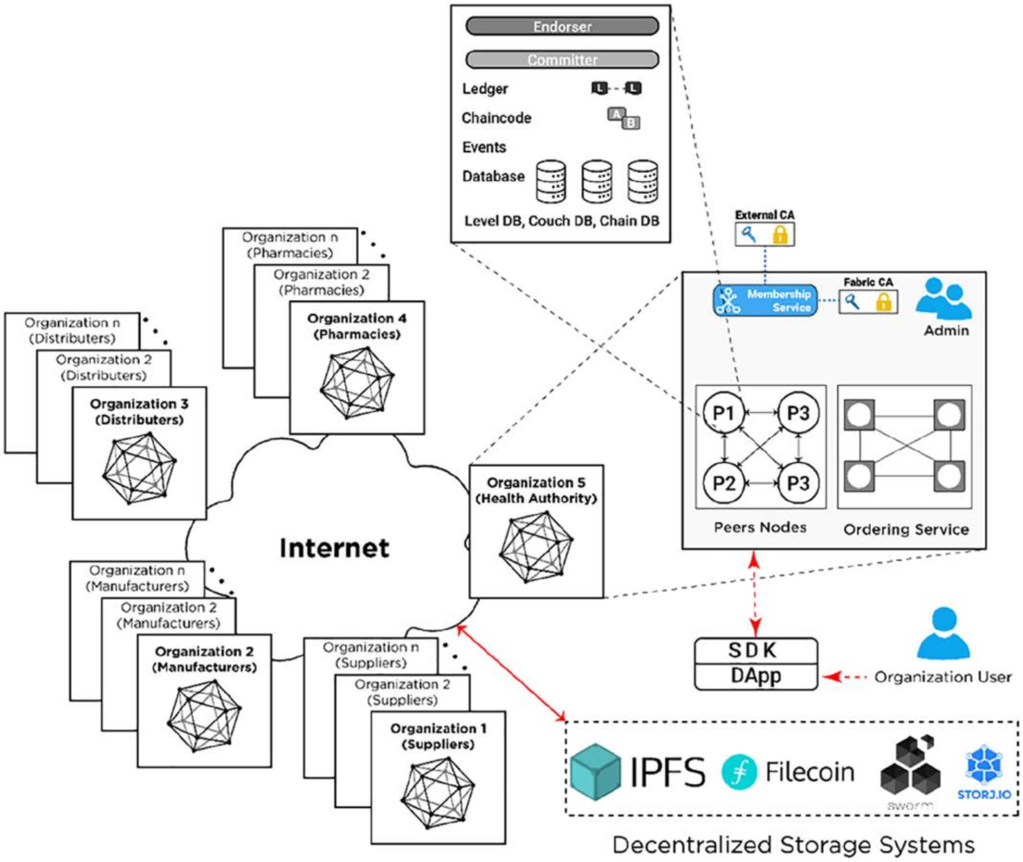
Endorser
Committer
Ledger
Chaincode
Events
Database
Level DB, Couch DB, Chain DB
External CA
Membership Service
Fabric CA
Admin
P1
P3
P2
P3
Peers Nodes
Ordering Service
SDK
DApp
Organization User
Organization n (Pharmacies)
Organization 2 (Pharmacies)
Organization 4 (Pharmacies)
Organization n (Distributers)
Organization 2 (Distributers)
Organization 3 (Distributers)
Internet
Organization 5 (Health Authority)
Organization n (Manufacturers)
Organization 2 (Manufacturers)
Organization 2 (Manufacturers)
Organization n (Suppliers)
Organization 2 (Suppliers)
Organization 1 (Suppliers)
IPFS
Filecoin
swarm
STORJ.IO
Decentralized Storage Systems

Figure 2. Hyperledger Fabric blockchain architecture.

- In the proposed Hyperledger Fabric architecture, initially, an organizational user (client app) from a registered organization such as supplier or manufacturer, submits a transaction proposal (Step 1).
- The transaction proposal is a request to invoke a chaincode function with certain parameters, with the intent of reading and/or updating the ledger (Step 2). This proposal is submitted to all endorsing peers, as determined by the chaincode endorsement policy (Step 3).
- To clarify, for every chaincode there is an endorsement policy stating which organizations, and by extent which peers, must sign/check every transaction for that chaincode. The transaction proposal consists of different parameters such as client's cryptographic credentials (obtained from an MSP), the transaction payload including the name of the chaincode function to be executed with input arguments, and the channel and chaincode identifiers. The client app sends this proposal to a set of endorsing peers to get a consensus that the transaction is valid. This phase is called the *proposal phase*.
- The transaction proposal is executed by a specific number of endorsing peers determined by the chaincode's endorsement policy (Step 4).

- These results (also called endorsements), will be encrypted, and recorded along with endorsing peers' cryptographic signatures and RW sets (*readset* and *writeset)*, and sent back to the client app, as a response to the transaction proposal submitted (Step 5).
- It is important to highlight that the client app continues collect endorsements until it satisfies the chaincode's endorsement policy. No updates are made to the ledger at this point. This phase is called the *endorsement phase*.
- When the client app received enough endorsement responses, it inspects them to determine if RW sets are the same, making sure the chaincode ledger was not updated in-between proposal and endorsement phases (Step 6).
- Next, the client app assembles and broadcasts the transaction proposal and responses within a transaction message to the Ordering Service (Step 7).
- This message contains a transaction with RW sets, endorsing peer signatures and channel identifier. The decentralized Ordering Service uses a pluggable consensus protocol to calculate and establish the execution order of all the submitted transactions per channel. The Ordering service chronologically orders multiple drug transactions into blocks, chaining the blocks' hashes to previous blocks (Step 8).
- This phase is called the *ordering phase*.
- The final phase is the *execution phase*. The OS broadcasts the newly-formed blocks to the leading peers in the Hyperledger Fabric network (Step 9). The leading peers are then in charge of disseminating the blocks to other committing peers within the organization using gossip protocol (Step 10).
- Leading peers are elected per organization and they are known to the Ordering Service. Peers check if the endorsements are valid according to the chaincodes' endorsement policies' and verify that the RW sets have not been violated since last checked (Step 11).
- If any endorsement is invalid or the RW sets do not match the current world state, the transaction is marked as invalid. Alternately, the ledger is updated and all peers append the transactions to the channels' ledgers in the predefined order, ensuring determinism (Step 12).
- Valid transactions will update the world state. Invalid ones are retained on the ledger but do not update the world state. Finally, the client app that submitted the transaction proposal will be notified by each peer on the network of transaction success (Step 13).

Hyperledger Besu architecture:

- The proposed Hyperledger Besu drug traceability architecture provides a fully compatible open-source distributed ledger solution for enterprises looking for Ethereum-compatible blockchain architectures.
- Hyperledger Besu is gaining popularity among enterprises as it supports building networks supporting both private transaction processing and integration with public blockchains (Ethereum), while maintaining architectural flexibility and high transaction throughput.

- The proposed Hyperledger Besu architecture bridges the gap between private and public blockchains and helps pharmaceutical supply chain organizations to build scalable, high-performance applications on peer-to-peer private networks that fully support data privacy and complex permissioning management.
- Hyperledger Besu supports business logic through Solidity smart contracts, and can take advantage of using ERC20 tokens and Ether cryptocurrency.
- Hyperledger Besu is an open-source Ethereum client. It provides a simple JSON-RPC API for running and managing Hyperledger Besu nodes and executing transactions.
- The proposed Hyperledger Besu architecture supports storing both private and public drug transaction execution information, which is required to implement an efficient drug traceability across the pharmaceutical supply chain between different stakeholders.
- The core components of Hyperledger Besu architecture, as shown in Figure 3, are Ethereum Virtual Machines (EVMs), EtherSign, and Orion nodes. Although it is revolving around a public blockchain, privacy, and permissioning are the two key features of Hyperledger Besu architecture.
- To create a permissioned private blockchain for the pharmaceutical supply chain,Hyperledger Besu allows creating specific organizations (stakeholders) and their users (nodes) with their associated network accounts (wallets/addresses).

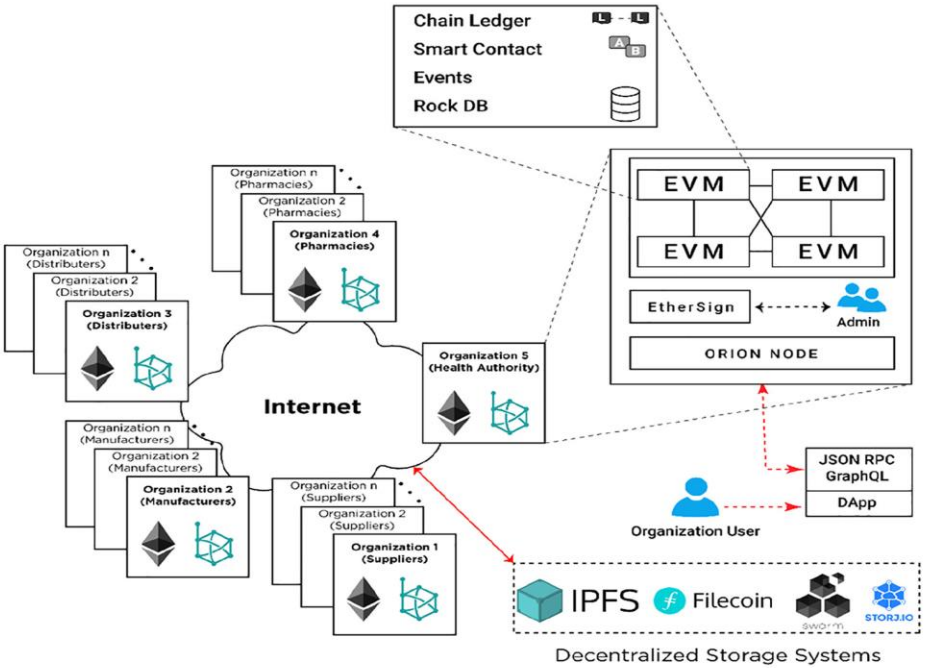
Chain Ledger
Smart Contact
Events
Rock DB
EVM
EVM
EVM
EVM
EtherSign
Admin
ORION NODE
Organization n (Pharmacies)
Organization 2 (Pharmacies)
Organization 4 (Pharmacies)
Organization n (Distributers)
Organization 2 (Distributers)
Organization 3 (Distributers)
Internet
Organization 5 (Health Authority)
Organization n (Manufacturers)
Organization 2 (Manufacturers)
Organization 2 (Manufacturers)
Organization n (Suppliers)
Organization 2 (Suppliers)
Organization 1 (Suppliers)
JSON RPC GraphQL
DApp
Organization User
IPFS
Filecoin
swarm
STORJ.IO
Decentralized Storage Systems

Figure 3. Hyperledger Besu blockchain architecture.

- To keep transactions private between involved stakeholders, Hyperledger Besu uses a Private Transaction Manager (PTM) such as Orion. PTMs that conform to the Enterprise Ethereum Alliance (EEA) Client Specification allows shared business logic in smart contracts to be made private to a limited number of participants, thus making all transactions and state associated with those smart contracts private as well. Orion, that is, native to Hyperledger Besu is such a PTM.
- Configuring a network that supports private transactions requires starting an Orion node for each Hyperledger Besu node. Lastly, to give access permissions to different organizational users and their accounts, Hyperledger Besu offers both on-chain (via smart contracts) and off-chain permissioning (via configuration files).
- A permissioned network enables node and account permissioning, making access to the network restricted to only specified nodes and accounts.
- Alongside, permissioning features of Hyperledger Besu allow real-time account suspension, denying access to broken smart contracts, restricting actions based on organization/account details, etc.
- This further enables secure and transparent communication on the network by easing the management of access control.

Drug traceability flow with Hyperledger Besu:

- In this section, we describe how medication-related traceability transactions are executed and communicated between different stakeholders on a Hyperledger Besu network.
- In the proposed Hyperledger Besu architecture, when an organizational user (client) wants to perform a transaction (execute a specific smart contract function or transfer assets), it initially submits a signed private transaction request through a Distributed App (DApp) to a Hyperledger Besu EVM node (Step 1).
- The signed transaction includes the list of recipients' addresses or privacy group ID, sender address, type of transaction (e.g. restricted), etc.
- To clarify, a privacy group is a group of nodes identified by a unique privacy group ID by Orion. Orion stores each private transaction with the privacy group ID.
- The Hyperledger Besu nodes maintain the public world state for the blockchain and a private state for each privacy group.
- The private states contain data that is not shared in the globally replicated world state. Privacy groups enable access to certain data only to a group of accounts/nodes.
- The DApp user interface uses JSON-RPC to send transactions to Orion (Step 2) through the Private Transaction Handler (PTH).
- Orion distributes the transactions to other Orion nodes specified by the privacy group ID edger Besu node will process PMT, and on nodes that contain the corresponding private

precompiled smart contract, the transactions are passed to the contract for execution (Step 6).

- This contract queries the Orion for the private transaction using the transaction hash value, and passes the transaction to the Private Transaction Processor, which executes the transaction and commits the read-write operations to the private world state to update all the participating nodes (Step 7).
- Nodes without the precompiled contract will ignore the marker transaction.

Technical comparison: Ethereum, Hyperledger Fabric and Hyperledger Besu:

- We present an in-depth technical comparison of the three blockchain platforms, highlighting their advantages and disadvantages. Although Ethereum can be setup as a private network, in this comparison we will focus on it as a public network.
- Features for comparison were collected empirically, based on evidence from past research and development, as well as actual documentation and several ongoing projects. Table 2 aims to offer deeper insights into outcome of the comparison.

Table 2. Comparison of Blockchain Platforms.

Category	Sub-category	Hyperledger Fabric	Hyperledger Besu	Ethereum
General	Governing bodies	Originally contributed by IBM and Digital. Now governed by Linux Foundation	Originally contributed by ConsenSys (PegaSys). Now governed by Linux Foundation	Ethereum Enterprise Alliance
	Maturity	Announced 2017, ~60 version improvements, high-level maturity	Announced 2019, ~30 version improvements, middle-level maturity	Announced 2015, ~100 Geth client releases, nine hard forks, high-level maturity
	Intended manner of usage	Private, permissioned	Semi-private	Public
	Community of contributors	~200	~60	~450
	Cryptography	Pluggable (ECDSA with secp256r1 and secp384r1 built-in)	secp256k1	secp256k1
Network management	Network configuration complexity	Cryptographic materials dynamically grow with number of organizations, peers, and usersUser-friendly command line libraries: cryptogen, configtxgen, configtxlator. High-complexity	Requires configuring a genesis file and Orion configuration file per node (EthSigner optional). Middle-complexity	Network is ready to be used immediately

Category	Sub-category	Hyperledger Fabric	Hyperledger Besu	Ethereum
	Network deployment	Using Docker and Docker Compose	Requires starting every Besu and Orion node through a simple command line command.Can be configured via Docker and Docker Compose	Network is publicly availableNode setup is optional
	Multiple ledgers	Yes, via channels	No concept of a chain (shared ledger)	No concept of a chain (shared ledger)
	Running costs breakdown	Dynamic ledger size (storage inside Docker container).	Dynamic ledger size.4GB for private network usage, 8GB for usage with public Ethereum networks (e.g. Ropsten)	Archive – >4TB, 4GB RAMFull node – 350GB SSD, 4GB RAM,Light node – 10GB, 4GB RAM
		1GB RAM per Peer		
		2GB RAM per Certificate Authority		
		256MB RAM per CouchDB		
		512MB RAM per chaincode container		
		256MB RAM per Ordering Service node (RAFT)		
	Storage options	LevelDBCouchDB	File storage	File storage
Transaction Execution	Transaction consensus	Unique Execute-Order-Validate methodology. Tolerating, instead of eliminating non-determinism	Order-Execute methodology. Every transaction execution must be deterministic	Order-Execute methodology. Every transaction execution must be deterministic
	Applied consensus protocols	Kafka/Raft (Crash Fault Tolerance with trusted leader)	Clique and IBFT 2.0 Proof of Authority (Byzantine Fault Tolerance)	Ethash Proof of Work (Byzantine Fault Tolerance)
	Cost/Fee	Only network running costs exist	Fee is paid in Ether for smart contract deployment or transaction execution. Cost different for different networks Besu nodes are integrated with (public Ethereum, Ropsten, etc.)	Fee is paid in Ether. Higher fee indicates faster mining/confirmation time. $2.71 per transaction for September 24th, 2020
Secure communication	TLS support	Yes	Yes	Yes
Identity and privacy	Data privacy	Private data collections	Privacy groups that can access private transactions	No support
	User and Node Permissioning	Organization level: channels	On-chain through smart contract	No support
		Chaincode level: function caller certificate/MSP attributes	Of-chain through configuration files	
	Identity generation and management	Based on PKI. Organizational identity rather than individual	Public keys—distributed, and interoperable between	Public keys—distributed, and interoperable between

Category	Sub-category	Hyperledger Fabric	Hyperledger Besu	Ethereum
		identities used in consensus and permissioning. Management through Certificate Authority Or third party certificate provider	Ethereum based chains. Coupled to PKI via proofs	Ethereum based chains. Coupled to PKI via proofs
Business logic implementation complexity	Client application responsibility	Coordinating with other participants to obtain endorsement, managing optimistic concurrency locking on state, signature, and submission	Sending signed transactions to a single node in the network	Sending signed transactions to a single node in the network
	Smart contract execution engine	Isolated inside Docker container	EVM (sandbox)	EVM (sandbox)
	Smart contract languages	General purpose: Java, Go, NodeJS. Non-determinism is tolerated	Domain specific (Solidity), guaranteed deterministic. Frameworks: Truffle, Embark, OpenZeppelin, etc.	Domain specific (Solidity, Viper), guaranteed deterministic. Frameworks: Truffle, Embark, OpenZeppelin, etc.
	Smart contract lifecycle	Requires installation on peers, instantiation in coordination with Ordering Service. Stored off-chain	Immutable, easy to deploy, and stored on-chain	Immutable, easy to deploy, and stored on-chain
	Smart contract upgrade	Replacing code via an upgrade transaction (versioning), similar to initial contract deployment	Programing schemes to extend/migrate code and data	Programing schemes to extend/migrate code and data
	Tokenization of digital assets	No native support. FabToken token management system in Hyperledger Fabric 2.0	Native feature with several token standards (ERC20, ERC721, ERC777 etc.)	Native feature with several token standards (ERC20, ERC721, ERC777 etc.)
Integration with Third party services	Services interaction	Service interacts with an SDK (NodeJS, Java, Python, Golang)	Distributed Apps (DApps) interact with nodes. JSON-RPC over HTTP or WebSockets	Distributed Apps (DApps) interact with nodes. Web3.js Ethereum JavaScript API. HTTP or IPC connection

Hyperledger Fabric and Besu support faster state reconciliation and offer superior transaction execution speed. Smart contracts in Hyperledger Fabric might be easier to develop since they use general-purpose programing languages, as opposed to Besu and Ethereum that use a domain-specific language.

However, Hyperledger Fabric lacks a proper smart contract development framework that is available in both Besu and Ethereum (e.g. Truffle). Network configuration, setup, and deployment complexities are higher for Hyperledger Fabric, however it is easy to manage/update/upgrade since all components are Dockerized.

Using Hyperledger Fabric also comes with increased client application responsibility, but increases the amount of control on the client side. Hyperledger Fabric is superior to both Besu and Ethereum for identity management and access control by having both physical (channels) and logical (chaincodes, certificate attributes) enablers to manage them. For the pharmaceutical traceability application both Hyperledger Fabric and Besu provide the best alternatives and features for effective trace and trace solution.

Conclusions:

- We discuss how blockchain technology can be leveraged for drug traceability application in the pharmaceutical supply chain.
- We proposed two blockchain architectures based on Hyperledger Fabric and Hyperledger Besu.
- Such architectures provide a shared, trusted, permissioned and decentralized platform for storage and communications among different pharmaceutical supply chain stakeholders, and in a manner that can fulfill key requirements and features that include security, privacy, accessibility, transparency, and scalability.
- We present a comparison of the two platforms, and outlined a number of implementation challenges that hinder the wide spread adoption of blockchain technology for effective drug traceability.
- As future work, we plan to develop smart contracts, deploy the overall system components, and build user interface DApps of the proposed architectures.