TECHONEY

Development of a **blockchain-based ecosystem** that allows an **improved positioning of small producers of honey** on local and international markets

DELIVERABLE

D4.1.3 Design and architecture of the consortiam blockchain platform (LIGM)

DUE DATE OF DELIVERABLE	30 April 2024 (M24)
START DATE OF PROJECT	30.September.2022
DURATION	36 months
LEAD PARTNER FOR DELIVERABLE	LIGM
DISSEMINATION LEVEL	PU (Public)

The TECHONEY project is part of the **PRIMA** programme (Project ID: *1711*, Call: *2021*, Topic ID: *2.3.1*) with support from the European Union's Horizon 2020 research and innovation programme and the national research and innovation funding agencies of Spain (AEI, *PCI2022-132917*), France (ANR), Algeria (DGRSDT), Luxembourg (FNR, *16735444*), Tunisia (MESRS), Morocco (MHESRI-M), Italy (MUR), and Turkiye (TÜBİTAK).

Funded by the European Union

DOCUMENT CHARTER

TITLE BLOCK

PROJECT	PROJECT FULL TITLE	DEVELOPMENT OF A BLOCKCHAIN-BASED ECOSYSTEM THAT ALLOWS AN IMPROVED POSITIONING FOR SMALL SCALE PRODUCERS OF HONEY ON LOCAL AND INTERNATIONAL MARKETS		TECHONEY
	PROJECT ACRONYM	TECHONEY		
	EC PROGRAMME	**PRIMA**	GRANT AGREEMENT NO	**+++.+++**
		PROGRAMME	Partnership on Research and Innovation in the Mediterranean Area	
		SECTION 2	Multitopic 2021	
		THEMATIC AREA 3	Agri-food chain	
		TOPIC 2.3.1	Increasing the resilience of small-scale farms to global challenges and COVID-like crisis by using adaptive technologies, smart agri-food supply chain and crisis management tools	
		FUNDING SCHEME	"Collaborative Project"	
		TYPE OF ACTION	"Research and Innovation Actions (RIA)"	
	COORDINATOR	**Tiziana de Magistris** · Principal Investigator Centro de Investigación y Tecnología Agroalimentaria de Aragón (**CITA**) · (Spain)		
	START DATE	01.May.2022	DURATION	**36 months**
	PROJECT URL	TECHONEY.org	E-MAIL	**info@techoney.org**
DELIVERABLE	DELIVERABLE	**D4.3.1 Techoney blockchain traceability system design based on hyperledger fabric**		
	WORK PACKAGE	WP4: Implementation of the blockchain distributed network		
	DATE OF DELIVERY	M24 (End April 2024)		
	NATURE	R (Report)	DISSEMINATION LEVEL	PU (Public)
	LEAD BENEFICIARY	Dr. Hakim Badis, LIGM, Gustave Eiffel University (**UGE**) · France		
	AUTHOR	Dr. **Hakim Badis** (LIGM, UGE), Dr. Muluken Eressa		
	REVIEWER	Dr. Hella Kaffel, Dr. Uwe Roth and Dr. Muluken Eressa		
	APPROVER	Dra. Tiziana de Magistris (CITA)		
	FILENAME	TECHONEY-Deliverable_4.1.3		
	Abstract of the Deliverable	The objective of this deliverable is to report on the selected blockchain framework and the network architecture that is being developed to test the traceability system for the techoney project. Among the available blockchain frameworks, we chose the Hyperledger fabric to build and test the Techoney traceability system for a variety of reasons. Besides it being a permissioned blockchain, where only a selected few can participate in business transactions, fabric blockchain network is also highly performant and scalable. Its scalability is due to its modular design. In the event where there is a high demand in customer transactions, the performance of the network can be improved by increasing the number of peer nodes and channels without causing any transactional overheads. This is a feature which cannot be easily achieved in the currently existing blockchain frameworks.		

VERSION TRACKER

VERSION	REVIEWER	PARTNER	ISSUE DATE	CHANGES
1	Hakim Badis	LIGM	04.Feb.2024	Initial draft
2	Hakim Badis	LIGM	16.March.2024	Peer review input
4	Muluken Eressa	LIGM	28.April.2024	Peer review input
5	Hakim Badis	LIGM	06.May.2024	Final version

STATEMENTS

ORIGINALITY

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

DISCLAIMER

- *All intellectual property rights are owned by the* **TECHONEY** *consortium members and are protected by the applicable laws.*
- *Except where otherwise specified, all document contents are: "© TECHONEY Consortium - All rights reserved". Reproduction is not authorized without prior written agreement.*
- **TECHONEY** *consortium members have agreed to full publication of this document. The commercial use of any information contained in this document may require a license from the owner of that information.*
- **TECHONEY** *consortium members are committed to publish accurate and up to date information. However, the* **TECHONEY** *consortium members cannot accept liability for any inaccuracies or omissions nor do they accept liability for any direct, indirect, special, consequential or other losses or damages of any kind arising out of the use of this information.*
- *This publication reflects only the* **TECHONEY** *consortium members view and the European Commission is not responsible for any use that may be made of the information it contains.*

REQUEST TO PARTNERS

This document fulfills **Project milestone #3** of **TECHONEY** PRIMA Proposal:[1]

- What: "Design and architecture of the consortium blockchain platform".
- When: M12 (month 12 – May 2023).
- How: Document uploaded in the private area of the project website

All partners are kindly asked to

- Read and understand this Deliverable.
- Use it for regular checking of actual performance vs project objectives.
- Report to project coordinator potential deviations of the project.

© 2022 **TECHONEY** Consortium – All rights reserved.

[1] Depicted at "Project Milestones". Section 3.2 ("Management structure, milestones and procedures") of the PRIMA Full Proposal - Technical Annex (Part II)

TABLE OF CONTENTS

DOCUMENT CHARTER 2
STATEMENTS 3
GLOSSARY: ACRONYMS 4
PROJECT ABSTRACT 6
EXECUTIVE SUMMARY OF THE DELIVERABLE 8
DOCUMENT STRUCTURE 8
INTRODUCTION 9
HYPERLEDGER FABRIC BLOCKCHAIN 10
HYPERLEDGER FABRIC CONSORTIUM FOR HONEY SUPPLY CHAIN 15

GLOSSARY: ACRONYMS

- **AI**: Artificial Intelligence

- **AR**: Augmented Reality
- **BDVA**: Big Data Value Association
- **C&D**: Communication and Dissemination.
- **CAP**: Common Agricultural Policy
- **CM**: Communication Manager
- **D**: Deliverable
- **DIAS**: Data and Information Access Services
- **EC**: European Commission
- **EU**: European Union
- **HILE**: Honey Innovation and Learning Living Lab
- **ICT**: Information and Communication Technologies
- **ICT**: Information and Communications Technology
- **KPI**: Key Performance Indicator
- **SC**: Steering Committee
- **SO**: Specific Objective
- **URL**: Uniform Resource Locator
- **WP**: Work Package

PROJECT ABSTRACT

TECHONEY project's main objective is to identify strategies and establish lines of resilience to the new challenges determined by the COVID-19 pandemic for beekeepers in the Mediterranean (Med) agricultural systems through the implementation, definition, enhancement and transfer of competitive, profitable, efficient and trustful honey supply-chain alternatives that address beekeepers' capacities and attractiveness to fulfil consumer needs on unexpected food market changes.

TECHONEY project proposes the development of a **traceability system to guarantee the quality and safety of honey within the supply chain** for more effective communication to consumers and to strengthen access to different markets (e-commerce, direct sales, etc.). This approach will be unfolded by the joint creation of two levels of interaction: **(1)** physical: characterization of honey; **(2)** "living laboratory": creation of a *Honey Community Living Lab* (**HCLL**) and a *Honey Innovation and Learning Ecosystem* (**HILE**) that will be the arena to collect information from beekeepers, stakeholders, and consumers to transfer and apply the new optimized models.

TECHONEY is structured in 4 main technological pillars: **(1)** creation of a consortium IoT (*Internet of Things*) Blockchain platform that involves various actors in the honey supply chain to ensure transparency and traceability, in addition to reducing costs and ensure the traceability in the honey supply chain; **(2)** creation of a transformative learning community to ensure a smart-short-resilient shared supply chain; **(3)** characterization of the quality of honey to guarantee its traceability within the Blockchain directly by consumers; **(4)** ICT tools for honey supply chain participants and consumers.

TECHONEY will be committed not only to promote the continuation of the direct sales of honey from producers to consumers even after the COVID crisis but also **develop a common methodology and clear new optimized resilience protocol** to be used by small-scale farmers, beekeepers, smallholders as a new business model with a more efficient added-value chain, sustainable with fair profit, accepted by final consumers, which will be replicable to other food products and supply chains.

TECHONEY will **help beekeepers to**: **(1)** generate a traceability mechanism for honey produced in the Med. Area; **(2)** diversify markets and distribution channels offers business flexibility and freedom from dependence on a single market, which will reduce risk in the event of a crisis; **(3)** cooperate and pool resources among themselves (pooling of the workforce, etc.) for logistical flexibility and solidarity in the supply chain, which will also reduce the risks in the event of a crisis.

TECHONEY proposes to design and develop a **multidimensional framework** to analyse 6 full honey supply chains (farm to table) that will integrate economic, social and environmental indicators and a traceability system, with a *bottom-up* approach considering the stakeholders´ perspectives. A consortium blockchain, coupled with IoT (*Internet of Things*), system will be created to offer real-time tracking and complete traceability of honey along the supply chain from the characterization of honey in a certified laboratory, the retailer, until the end consumer.

The characterization of the quality and safety aspects of local honey through **local certified laboratories jointly with the use of e-commerce and quality labeling schemes** will increase the opportunity for beekeepers to be identified locally, and allow them to gain access to new markets (foreign markets). The implementation of e-commerce with the mobile application will enable local honey to be **better traced by consumers who attach more value to local food and local beekeepers**.

TECHONEY contributes to increasing farm profitability and increasing flexibility and risk mitigation capabilities. A shared, short and circular supply chain will allow actors in the honey supply chain to **access markets** and have **higher incomes**, **share resources and skills** and **save money** by reducing costs (economies of scale) and increase the efficiency, sustainability and flexibility of processes to strengthen resilience and flexibility to face crises and lower risks. The learning community lab and the use of the Blockchain network will secure the storing of records, will **strengthen intellectual property rights**, as well as **bring transparency throughout the supply chain**; it will **reduce frauds**, **enhance food safety** and **improve the communication** between retailers and beekeepers. The traceability system offered will also allow consumers to give direct feedback to beekeepers.

To reach the overall objective, several **specific objectives** (SO) are set out as intermediate goals:

- **SO1:** Map the current added-value chains and complexity level for honey products in six case studies (Spain, Algeria, Tunisia, Turkey, Luxembourg and Morocco).

- **SO2:** Increase the competitiveness and power of the beekeeping supply chain by understanding consumers' and retailers' opinions and acceptance through non-hypothetical methods.
- **SO3:** To promote the traceability to differentiate local honey and guarantee food safety of honey produced in the Med. area by characterizing it in a reliable, certifiable, and documentable manner.
- **SO4:** Design and explore the feasibility of a new traceability ecosystems and effective business models across different stakeholders.
- **SO5:** Development of a new "Multichannel distribution" e-commerce platform implementing new business model as well as integrating and scaling up the outcomes from WPs 1, 2, 3 and 4.
- **SO6:** Maximize outreach and beneficial influence of the project results and reach the target users (beekeepers, small-scale food manufacturers and local distributors, canteens and retailers, local public authorities) through an effectively established communication and dissemination plan, including innovative training capsules.

Moreover, a **TECHONEY** web ICT tool consumer/farmer-centred will be developed, by testing and evaluating several machine and deep learning algorithms, providing small-scale beekeepers with key information on new markets and opportunities, contributing to a better decision making and to ensure the traceability of their product. Consumers will have exhaustive knowledge of the different honeys of the Med. area, knowing their traceability from the initial producer, guaranteeing the quality and safety of each product.

TECHONEY *is a project coordinated by CITA (Spain) and funded through the PRIMA Section 2 Multitopic 2021 – Thematic Area 3-Agrofood chain – Topic 2.3.1 Increasing the resilience of small-scale farms to global challenges and COVID-like crisis by using adapted technologies, smart agri-food supply chain and crisis management tools. (RIA*[5])" under the funding scheme of "Collaborative Project" and type of Action "Research and Innovation Actions (RIA)."*

EXECUTIVE SUMMARY OF THE DELIVERABLE

This Deliverable D4.1.3 is about the selection of a blockchain framework for the Techoney Blockchain based traceability system design, the proposed PoC architecture and the functional requirement site map for user interface design. Among the available blockchain frameworks, we chose the Hyperledger fabric to build and test the Techoney traceability system for a variety of reasons. Hyperledger Fabric is an open source blockchain maintained by the Linux foundation under the umbrella of the Hyperledger project. Among the various blockchain platforms available, Fabric Blockchain offers a number of advantages in relation to performance, scalability and the needed privacy within the honey supply chain. For instance, unlike other blockchain frameworks, fabric has a modular architecture with components like consensus and services (smart contracts) in a plug-and-play fashion. This modular design along with a support for a permissioned network for privacy, ensures high transaction throughput without compromising on data privacy or security. Furthermore, the creation of channel for a more private sub-network within the permissioned network allows not only for more privacy but also can lower network congestion. Thereby further enhancing the resiliency and performance of the network during a high transactional period. The realization of a honey supply chain based on the Hyperledger fabric blockchain demands a careful consideration of the legal, regulatory, governance, configuration and practical requirements pertaining to the framework; acknowledge the needs, concern and preference of the direct participants within the supply chain and design smart contracts that encapsulate the underlying business logic and binding agreements. Consequently, this deliverable also proposes a functional requirement site map for a user interface (front-end) design for an easier interaction with the underlying blockchain and smart contract in a way that encapsulate the needs of the different stakeholders in the honey supply chain.

DOCUMENT STRUCTURE

This D4.1.3 document is organized according to the following structure:

- **INTRODUCTION:** Short introduction into blockchain frameworks, confidentiality and data privacy
- **HYPERLEDGER FABRIC BLOCKCHAIN:** Introduction to the core components of the fabric network
- **HYPERLEDGER FABRIC CONSORTIUM FOR HONEY SUPPLYCHAIN:** Depicts a sample consortium for the proposed PoC architecture

INTRODUCTION

In the deliverable D4.2.1, we conducted research on possible blockchain frameworks for the realization of the Techoney supply chain, along with the preparation of questionnaires for a mobile and web interface that takes into account the needs of all participating stakeholders. Based on our investigation, in this deliverable 4.3.1 and all the deliverable that follows, we have decided to implement the Techoney blockchain using the Hyperledger fabric networks. Hyperledger Fabric is an open source blockchain maintained by the Linux foundation under the umbrella of the Hyperledger project. Among the various blockchain platforms available, Fabric Blockchain offers a number of advantages in relation to performance, scalability and the needed privacy within the honey supply chain. For instance, unlike other blockchain frameworks, fabric has a modular architecture with components like consensus and services (smart contracts) in a plug-and-play fashion. This modular design along with a support for a permissioned network for privacy, ensures high transaction throughput without compromising on data privacy or security. Furthermore, the creation of channel for a more private sub-network within the permissioned network allows not only for more privacy but also can lower network congestion. Thereby further enhancing the resiliency and performance of the network during a high transactional period.

Confidentiality is something that our beekeepers have been really curious about which is also important in general supply chain management, particularly in businesses like honey production where sensitive data must be protected. Fabric Blockchain provides robust privacy features, allowing participants to transact securely while maintaining secrecy. Its permissioned network model enables granular control over access rights, ensuring that only authorized parties can view and verify specific transactions. This privacy-enhancing capability is crucial for safeguarding proprietary information and maintaining trust among stakeholders. In Hyperledger fabric, peer nodes constituent the core elements of the blockchain network. Their responsibility lies on the management of smart contract, the immutable ledger, the execution and endorsement of transaction proposals. The number of peer nodes is not only dependent on the number of the participating organizations, but also on the anticipated volume of transactions, the required level fault tolerance, load balancing, availability etc. The scalability of the fabric framework allows for the addition/deletion and easier reconfiguration of these elements. This feature is crucial for accommodating the diverse stakeholders and large volumes of transactions that we anticipate in the honey supply chain.

HYPERLEDGER FABRIC BLOCKCHAIN

Hyperledger Fabric has a different design compared to other blockchains. First of all, Hyperledger Fabric compartmentalize transactions into separates functions. Such as ordering transactions in blocks based on their chronological appearance, then validating transactions to check their validity and authenticity and finally committing transactions into the immutable ledger. In doing so, Fabric has an Ordering service, which are nodes that are tasked with ordering the transactions into blocks, called Orderers. On the other hand, nodes that are tasked with validating and committing transactions are called Committers. A third type of node is called Endorsers. When a client proposes a transaction, an endorser or group of endorsers forwards these transactions to the ordering service, which forwards ordered blocks to committers to be validated and committed to the ledger via consensus. We will dive more into this in the transaction life cycle part. For simplification in this part, we refer to these three types of nodes as Peers. A key concept that Fabric endorses is the Membership Service Provider (MSP). Fabric is not a public blockchain, it is permissioned. Nodes wishing to join the network are required to be identified and authenticated with an Enrollment Certificate (E-Cert) created by a Certificate Authority (CA) which can be provided by Fabric itself or an external agency, depending on the needs of the system. The MSP is an internal PKI component to the Fabric network and allows for different types of certifications. Furthermore, Hyperledger Fabric supports distributed computing via Chaincode.

Chaincodes are docker based environments where we deploy smart contracts containing the underlying business logic for the application. These smart contracts are used to submit and update transactions on the ledger, and query the transactions stored on the world state databases maintained by peers. The World state database contains all committed transactions. Hyperledger fabric provides a pluggable interface for its world state. A variety of databases can be used to store the current states of the transactions. However, the default world state is stored as a key-value pair using Level DB. For a more robust query operation, a JSON object-based database such as CouchDB can also be employed depending on the system design preferences. A smart contract can also be used to produce events which external systems can subscribe too, to listen for events on blockchain. Due to the different approach used in transactions and the different transaction life cycle, Fabric offers a deterministic environment for code execution allowing for more general coding languages like Go, JavaScript and Java to be used for implementing the business logic in smart contracts. Moreover, Fabric offers multiple APIs and components that can be used to facilitate the development of interfaces of an easier interaction with the blockchain.

The following figure illustrates the above details.

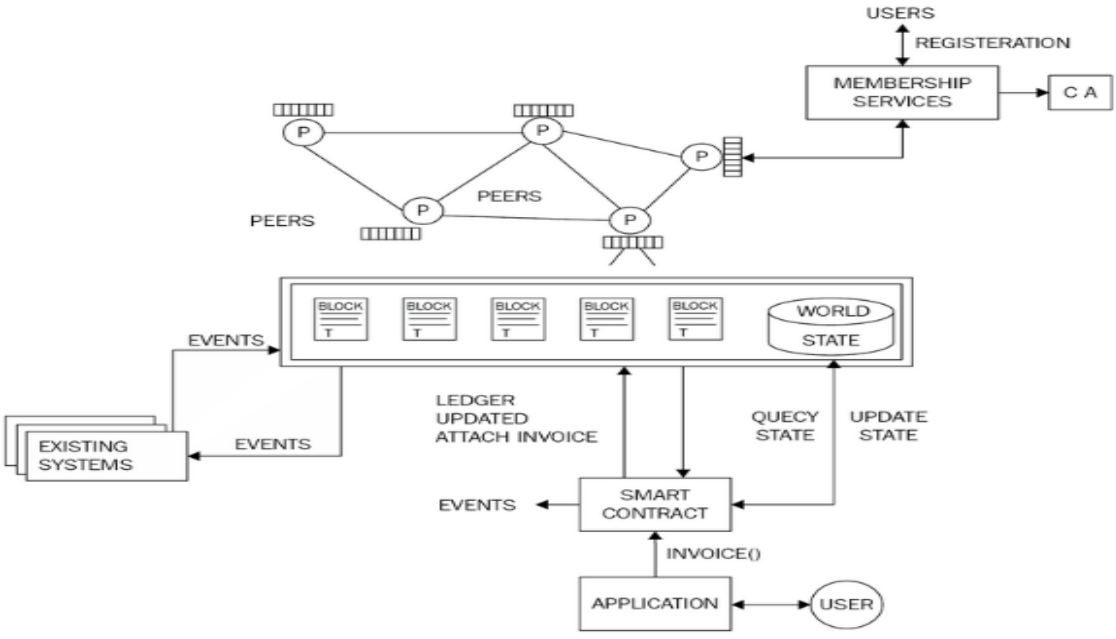

Figure 1. *Hyperledger fabric network overview.*

The roles of each components forming the Hyperledger fabric network shown in Figure.1. can be summarized as follows:

1. MSP attributed certificates for users wishing to be part of the network, they can be E-certs allowing full-time enrollment or T-certs used for a one-time transaction;
2. Enrolled users are now part of the peers, they can either be part of the ordering service or part of the other peers (endorsers and committers);
3. Peers communicate via a Peer-to-Peer protocol based on google RPC (gRPC), they have four (4) types of messages: discovery, synchronization, consensus and transaction (deploy, query and invoke);
4. The ordering service broadcasts ordered transactions in blocks for committers to validate and commit.
5. Peers maintain a ledger and a world state database;
6. Applications interact with the ledger and the world state database via smart contracts;
7. Smart contracts are deployed in isolated environments using docker containers, called chaincode;
8. Smart contracts can query the world state database, create transactions and update the ledger, they are invoked by applications/clients and provide events;
9. External systems listen to events on the blockchain using event adapters.

As mentioned in the above section, Fabric is a deterministic environment for code execution and allows for more general coding languages to be used in smart contracts, in the next section, we explore this feature by detailing the transaction life cycle on Fabric, which allows to understand why it is deterministic, and we compare it to transaction life cycle on Ethereum to show the contrast between the two blockchains.

Transactional life cycle

A transaction in Fabric has three major steps: execute, order and validate. After a proposal transaction initiated by an application/client, is sent to an endorser. The endorser executes a chaincode and generates a Read-Write (RW) set simulating the transaction. In this phase, the ledger is not updated yet, only a RW depicting reads and updates is created. This is called an endorsed transaction; the endorser sends it back to the proposing party (application/client) which submits it to the ordering service. Upon receiving the endorsed transactions from different parties, the ordering service (which is a cluster of ordering nodes) sorts endorsed transactions into blocks and broadcasts them to committers. Committers validate the transactions and write them to the ledger and the world state database so they become final permanent transactions. The following figures illustrate the transaction life cycle in fabric.

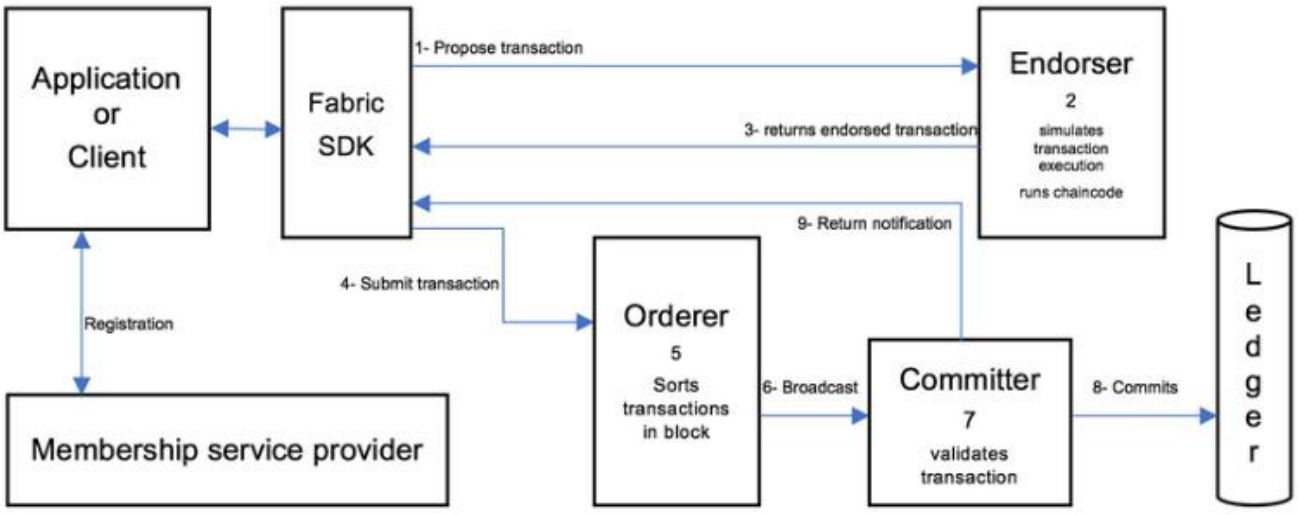

Figure 2. *Transaction Life Cycle in Fabric.*

As depicted in Figure 2 and Figure 3, the client contacts multiple endorsers to endorse the proposed transaction. The first one to do it, returns an endorsed transaction to the client, which in turn broadcasts it to the network to be ordered by the ordering service which in turn submits it to a committing peer. The committing peer validates the endorsed transaction by verifying the endorsement and commits it to the ledger.

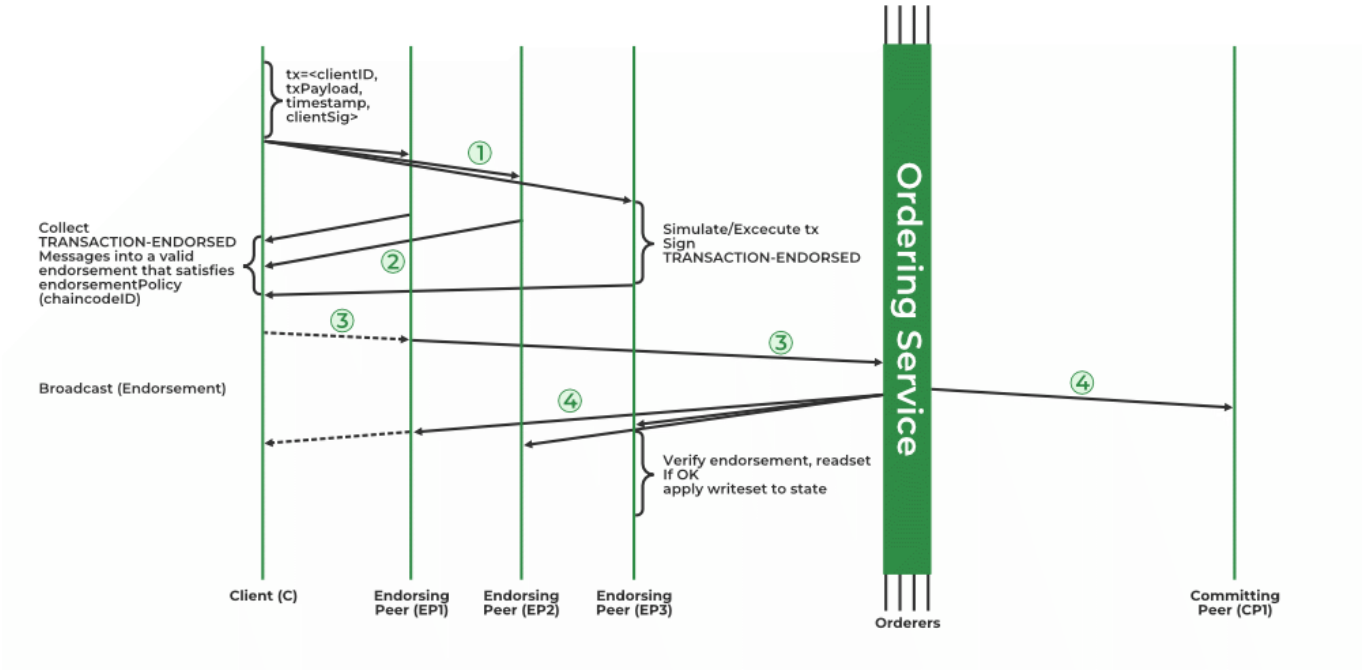

Figure 3. *Transaction sequence diagram in Fabric.*

This life cycle has major consequences. Since transaction execution precedes ordering, Fabric has a higher throughput than other blockchains that order transactions before executing them. Through such transaction ordering, the blocks that reach the committing peers for validation are final blocks and guaranteed to be correct. Consequently, Fabric has no chain forks and has a deterministic consensus. With a deterministic consensus, there is no constraints when it comes to the programming languages used for smart contract design. In Fabric, we can go beyond the classical smart contracts written in solidity like in case of Ethereum. Fabric offers so much flexibility and smart contracts can be written in any general-purpose programming language, although the project currently only leverages Go, JavaScript and Java as main programming languages. Once we discussed the transaction life cycle and the design philosophy behind Fabric blockchain networks, now let us see the features of this blockchain.

Features of Fabric Blockchain

Hyperledger Fabric is a permissioned blockchain, the presence of an MSP component allows for identifying and authenticating network nodes using certificates. Since nodes can be identified and authenticated, we can implement access control on the network which in return results in an important feature of Hyperledger Fabric: private channels. Private channels are actually sub-ledgers maintained by a certain selection of nodes. These nodes are chosen and authorized using the MSP certificates. A channel is a private space within the permissioned network containing a ledger with only a few selected peer nodes are capable of reading and writing on it. This allows for parallel transaction execution, where different channels can execute different channel-specific transactions in parallel. Channels allow a confidential flow between different parties, data isolation and a higher throughput. Another important feature of Fabric is its Ordering Service, which is responsible for ordering transactions that is separated from executioners and committers. This allows for avoiding bottlenecks and network congestion that usually happens in public blockchains where the same node is used to performs transaction ordering and execution. This separation between ordering and execution improves the scalability and performance of the network by mitigating bottlenecks and single points of failure.

Hyperledger Fabric also provides a highly modular architecture with separable components where even the consensus mechanism can be operated in a plug-and-play fashion. Meaning that the Fabric network can be run in a customized consensus and different algorithms, unlike other blockchains that come with a specific consensus implementation in mind. Although, we can introduce assets/tokens into the Fabric blockchain, it doesn't come

with a native cryptocurrency. As a result, there is no Mining which eliminate the risks and attacks related to mining like the Gas attacks in Ethereum.

We can summarize the Hyperledger fabric framework features as:

1. Fabric is highly modular.
2. Fabric allows for node identification, authentication and authorization using a Membership Service Provider which is a PKI creating different types of certificates for network members.
3. Fabric private channels are sub-ledgers with few selected peers authorized to participate in reading and writing via the MSP. These channels allow for data isolation, confidential flow between peers and parallel execution of transactions. Sub-networks within a private network, where peers have different roles depending on the channels (orderer nodes, endorsers and committers) they are assigned to.
4. The separation of nodes as orderer, endorsers and committers eliminate bottlenecks and results in better performance and network scalability.
5. Channels offer data privacy.
6. The fabric API provides a constraints free chaincode development environment. Thus, a wider range of business logic can be implemented through multiple programming languages such as Go, JavaScript and Java.
7. Fabric comes with a key-value pairs storage (LevelDB) as its world state database. However, due to its modular and plug-play component design, a variety of other databases such as couchDB, mongoDB can be used for a rich and robust query operation.

Fabric Based Application Models

Fabric comes with APIs and command line interfaces that enable applications and clients to invoke & submit transactions and query the blockchain data. Furthermore, these API also provide user identity management services, storage, and event broadcasting for connected systems.

Typically, application development in Fabric follows the MVC-architecture, where the classic MVC (Model - View - Controller) architecture is used with an added Blockchain logic.

- **View:** this is the user interface, usually a desktop, web application or mobile front-end used by the end user to interact with the system.
- **Controller:** the orchestrator between the front-end and the back-end, the back-end includes the Data Model (data) and the APIs used to interact with the blockchain.
- **Data Model:** this is the off-chain and on-chain data storage. Off-chain data represent data stored in databases which are important for the application and yet are not part of the transaction. On the other hand, the on-chain data is the query-able transactional data stored in the blockchain ledger and world state database on peers.
- **Blockchain-logic:** represent the smart contract used to manage the blockchain via the controller and the data model via transactions.

The following figure depicts a classical simple Fabric based application following the MVC-B model.

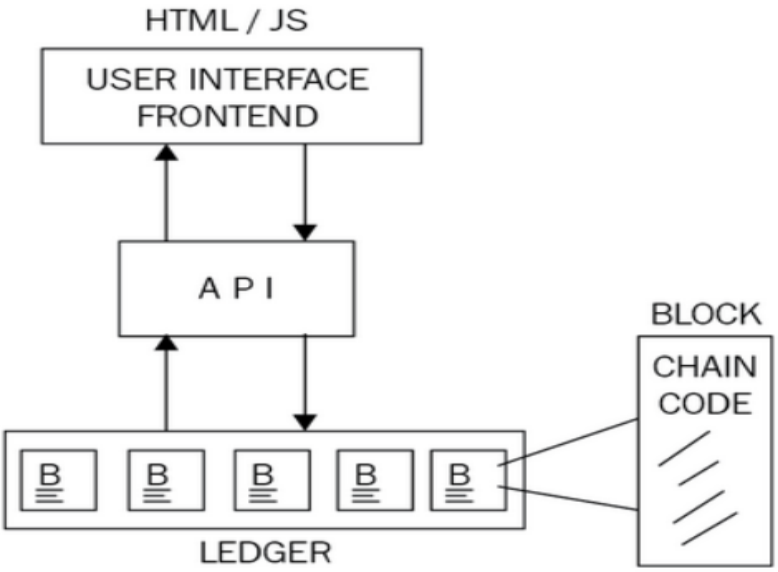

Figure 4. *The MVC-B application model.*

HYPERLEDGER FABRIC CONSORTIUM FOR HONEY SUPPLY CHAIN

In general, when a group of organization agree to use the fabric network for a business transaction, they form a consortium. As members, they discuss and agree on various issues. Such as the business logic to be implemented or the chaincode definition, the number of network nodes they will dedicate to the network and network governance. The final part is exchanging some public identifiers to distinguish one another over the internet. Apart from these, as individual organization, they need to protect their privacy and they don't disclose any more information. Meaning that, it is the responsibility of each organization, to design the interface (web/mobile) needed to interact with the blockchain.

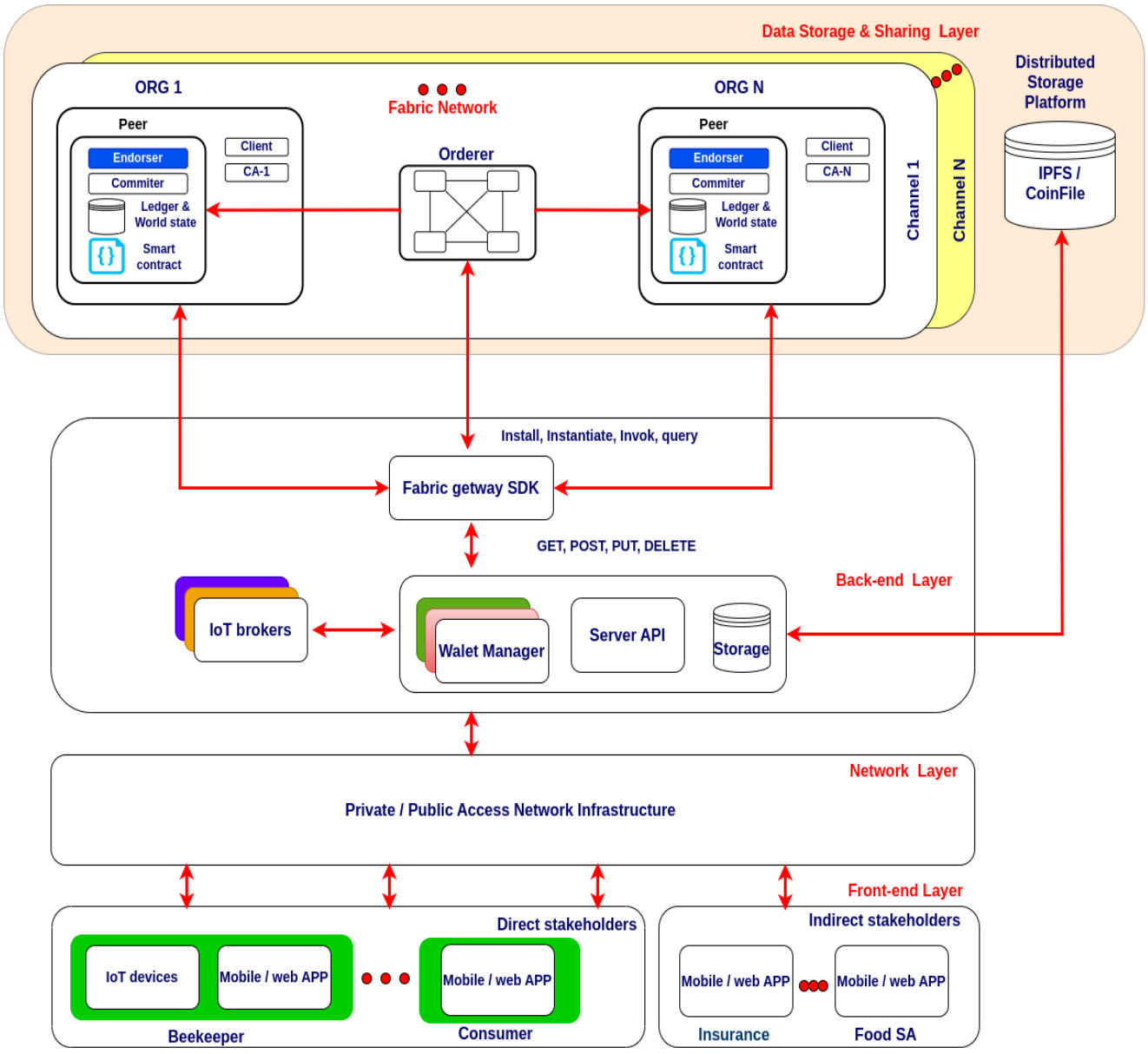

Figure 5. *Three-tier Techoney PoC architecture*.

As a result, these organization usually have network administrators and software developers to monitor the network nodes at their end and design a user interface to integrate the chaincode with the company's inner workings and access the blockchain. However, the majority of PRIMA's consortium is made from small scale beekeepers, wholesalers and retailer, who might not have either the finance to hire a trained personal or the expertise to manage the network nodes at their end. As a result, it is our objective to design an interface (web/mobile) that serves as a universal gateway for the participants to utilize the fabric blockchain for a business transaction. To that end, we have followed a 3-tier architecture (i.e., Front-end, Back-end and Blockchain layer) for the Techoney traceability system realization as shown in Figure.5. As mentioned in the previous section,

Fabric provides API for easier application development and blockchain interaction. As a result, the client/front-end layer is not required to directly interact with the blockchain. Their interaction is facilitated through the back-end layer. The back-end layer encapsulates most of the blockchain interaction, so that the front-end layer can be focused more on interface design for a friendly user interaction.

Front-end layer

The involvement of all stakeholders in the realization of a traceable blockchain-based honey supply chain is paramount. The ultimate goal of this blockchain based e-commerce is to give beekeepers the needed exposure so that they can trade their products. It should also guarantee the safety and traceability of the product. And yet, not all beekeepers or any of the stakeholders involved in production, processing, transportation and trading of honey have similar concerns when it comes their business representation on the blockchain. For instance, the size & reach of their trade horizon makes a difference in terms of the services and product type they provide. The type of stories they would like to tell to their potential clientele, how their information should be structured, how they describe the type and quality of honey they produce: raw, bottled or processed etc., all goes toward designing not only a client interface but also dictate the very transactional data that will be used for recording into the ledger and product tracking during traceability. As such, understanding and identifying stakeholders in the honey supply chain, what their needs are and their individual preferences, in addition to aligning their interest, it can also simplify designing the interface. While we have prepared the questionnaire in the previous deliverable, on this deliverable we have prepared a site map that will used as a draft for the initial user interface design as shown in the Figure.6.

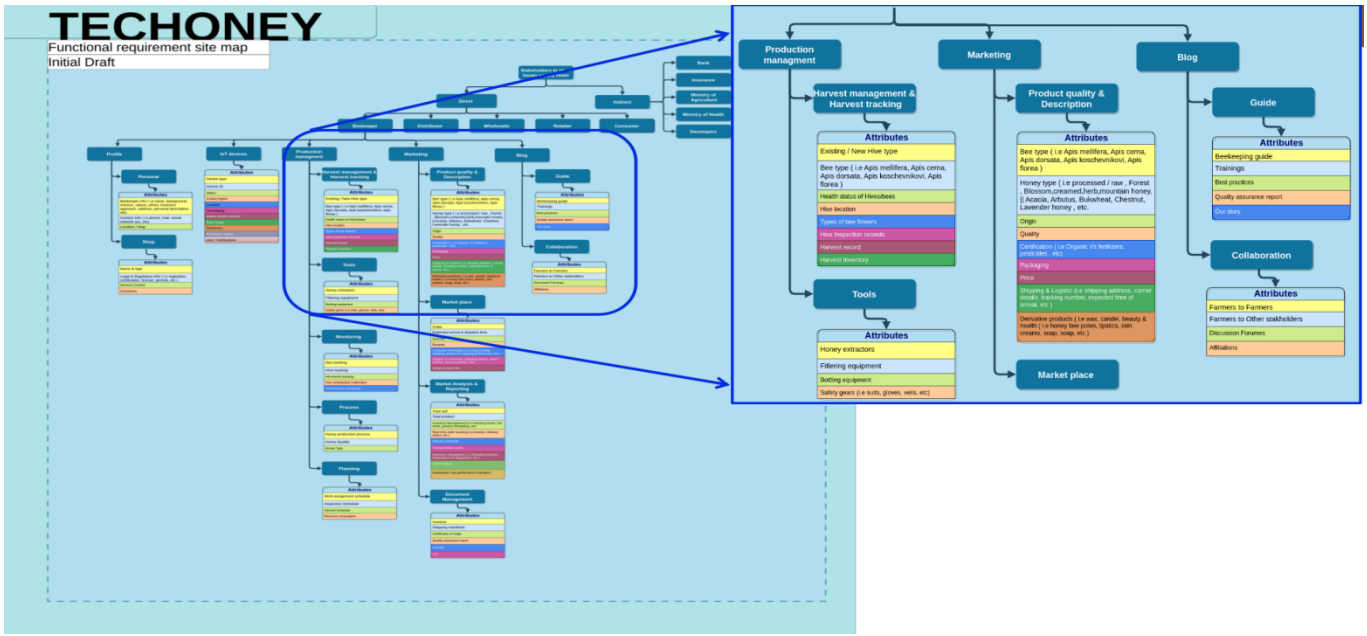

Figure 6. *User interface site-map and functional requirement for the PoC.*

In the site map shown in Figure 6, we have tried to anticipate and incorporate all the features that will be useful to each participant within the honey supply chain based on their roles. Meaning that although there will be some similarities, the functionality extended to beekeepers is a bit different from those of the wholesalers and retailers alike. For example, the beekeepers' interface will have an interface for hive monitoring and management which is unique based on their role in the supply chain.

Blockchain layer

For Techoney PoC implementation in Hyperledger fabric, we created a network configuration file for three Org Techoney consortium with a single channel for business transaction as shown in Figure 7 and Figure 8.

```
Profiles:

    TecHoneyOrdererGenesis:
        <<: *ChannelDefaults
        Orderer:
            <<: *OrdererDefaults
            Organizations:
                - *OrdererOrg
            Capabilities:
                <<: *OrdererCapabilities
        Consortiums:
            TecHoneyConsortium:
                Organizations:
                    - *Org1 /* Beekeepers organization */
                    - *Org2 /* Wholesalers orgnaization */
                    - *Org3 /* Retailer organization */
    TecHoneyChannel:
        Consortium: TecHoneyConsortium
        <<: *ChannelDefaults
        Application:
            <<: *ApplicationDefaults
            Organizations:
                - *Org1
                - *Org2
                - *Org3
            Capabilities:
                <<: *ApplicationCapabilities
```

Figure 7. *A three Org Techoney Consortium for PoC implementation.*

With Org1, Org2 and Org3 representing the beekeepers, wholesalers and retailers' organization in the PRIMA blockchain network.

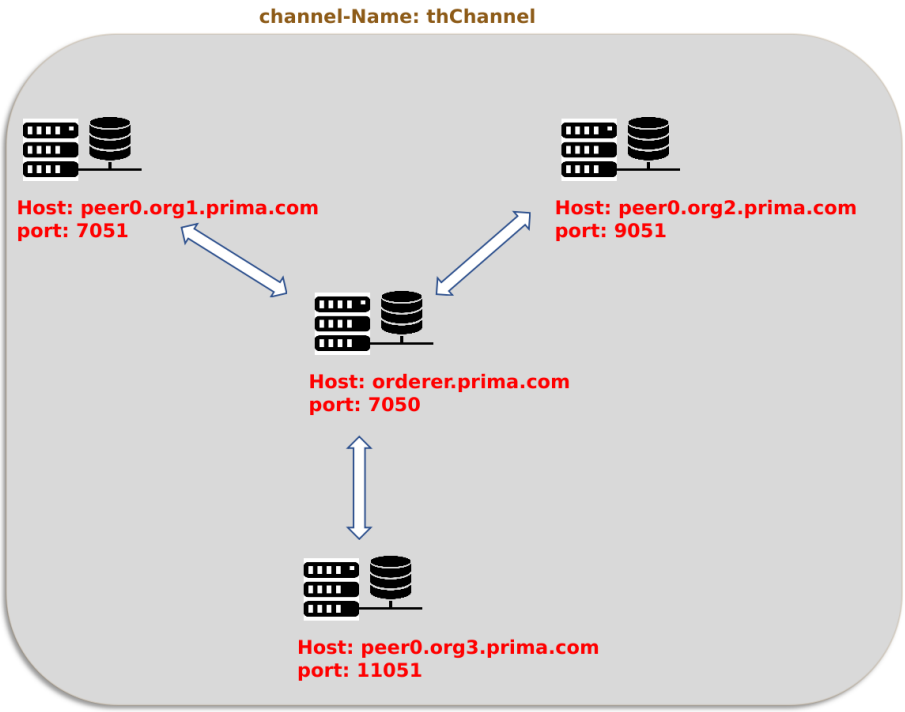

Figure 8. *PRIMA PoC blockchain network.*

Instantiating a blockchain network in Fabric requires numerous steps. From generating certificates for identity management, creating channels, specifying anchor peers for each participating organization, joining peers to the channels up-to the chaincode deployment and its installation on the given channel goes through a number of steps depending on the agreed policies. We have created a bash script to automate these processes and going through each detail in itself takes even longer. As a result, here we show the important details regarding certificate generation for each organization, PRIMA blockchain creation and a sample command line chaincode initialization as shown in Figure.9, Figure 10 and Figure 11 respectively.

```
############################################################
##### Generate certificates using cryptogen tool #########
############################################################

############################################################
############ Create Org1-Beekeeper Identities ############
############################################################
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-org1.yaml --output=organizations
org1.prima.com
+ res=0
+ set +x
############################################################
############ Create Org2-Wholesaler Identities ############
############################################################
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-org2.yaml --output=organizations
org2.prima.com
+ res=0
+ set +x
############################################################
############ Create Org3-Retailer Identities ##############
############################################################
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-org3.yaml --output=organizations
org3.prima.com
+ res=0
+ set +x
############################################################
############ Create Orderer Org Identities ################
############################################################
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-orderer.yaml --output=organizations
+ res=0
+ set +x

Generate CCP files for Org1, Org2 and Org3
/home/badhak/Desktop/PRIMA_demo/PrimaTechoney/PRIMAnetwork/../bin/configtxgen
##########  Generating Orderer Genesis block ##############
+ configtxgen -profile TecHoneyOrdererGenesis -channelID system-channel -outputBlock ./system-genesis-block/genesis.block
```

Figure 9. *Identity and genesis block creation for the PRIMA Techoney PoC.*

```
Creating network "prima_network" with the default driver
Creating volume "prima_orderer.prima.com" with default driver
Creating volume "prima_peer0.org1.prima.com" with default driver
Creating volume "prima_peer0.org2.prima.com" with default driver
Creating volume "prima_peer0.org3.prima.com" with default driver
Creating peer0.org2.prima.com ... done
Creating orderer.prima.com    ... done
Creating peer0.org3.prima.com ... done
Creating peer0.org1.prima.com ... done
CONTAINER ID   IMAGE                               COMMAND             CREATED         STATUS                  PORTS                                       NAMES
f4ab61c4651a   hyperledger/fabric-peer:latest      "peer node start"   1 second ago    Up Less than a second   7051/tcp, 0.0.0.0:11051->11051/tcp          peer0.org3.prima.com
4f238cc6acab   hyperledger/fabric-peer:latest      "peer node start"   1 second ago    Up Less than a second   0.0.0.0:7051->7051/tcp                      peer0.org1.prima.com
b3494b8d226d   hyperledger/fabric-orderer:latest   "orderer"           1 second ago    Up Less than a second   0.0.0.0:7050->7050/tcp                      orderer.prima.com
c44808dfe6cf   hyperledger/fabric-peer:latest      "peer node start"   1 second ago    Up Less than a second   7051/tcp, 0.0.0.0:9051->9051/tcp            peer0.org2.prima.com

 ____    _____      _      ____    _____
/ ___|  |_   _|    / \    |  _ \  |_   _|
\___ \    | |     / _ \   | |_) |   | |
 ___) |   | |    / ___ \  |  _ <    | |
|____/    |_|   /_/   \_\ |_| \_\   |_|

Creating PRIMA Network Channel thchannel

### Generating channel create transaction 'thchannel.tx' ###
***** [Step: 1]: start call createChannelTxn on peer: peer0.org, channelID: thchannel, smartcontract: , version , sequence  *****
====== generate channel configuration transaction =====
+ configtxgen -profile TecHoneyChannel -outputCreateChannelTx ./channel-artifacts/thchannel.tx -channelID thchannel
2024-05-11 15:22:06.719 CEST [common.tools.configtxgen] main -> INFO 001 Loading configuration
2024-05-11 15:22:06.733 CEST [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded configuration: /home/badhak/Desktop/PRIMA_demo/PrimaTechoney/PRIMAnetwork/configtx/configtx.yaml
2024-05-11 15:22:06.733 CEST [common.tools.configtxgen] doOutputChannelCreateTx -> INFO 003 Generating new channel configtx
2024-05-11 15:22:06.740 CEST [common.tools.configtxgen] doOutputChannelCreateTx -> INFO 004 Writing new channel tx
```

Figure 10. *PRIMA blockchain network channel creation.*

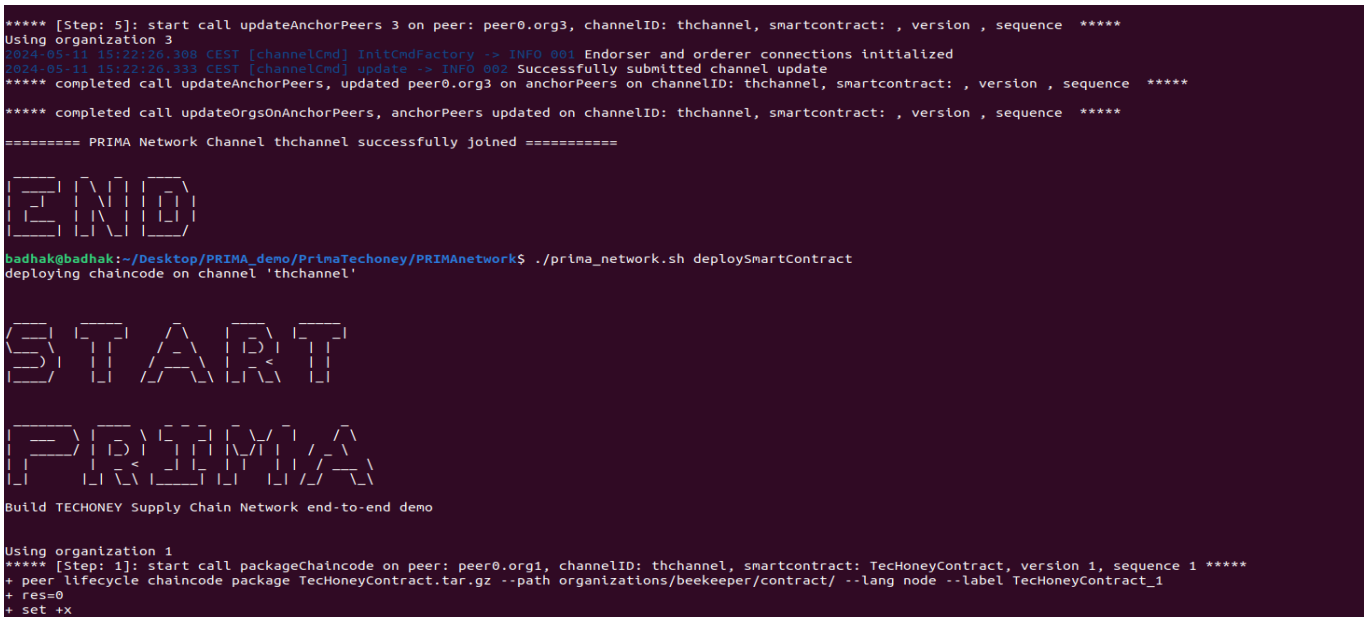

Figure 11. *PRIMA blockchain instantiation and chaincode deployment.*

In general, Hyperledger Fabric allows for a more deterministic code execution environment. This leverages the use of general-purpose programming languages like Go, Java and JavaScript and covers a wider area of business logic that classical smart contracts on Ethereum do not cover. Moreover, Fabric allows for data privacy, data isolation and higher throughput via the concept of private channels which is a network within network architecture. Features like high modularity, plug-and-play components, scalability and performance make Fabric a suitable blockchain for a wide range of applications like digital identity, supply-chain and asset (real world assets, digital assets, data) tracing. That is precisely the very reason we chose the Fabric platform our choice for developing a Blockchain based honey supply chain for the Techoney project.