



Blockchain Powered Procurement System for GAIL



Organisation Name :- **GAIL**
Problem Statement Code :- **BK225**
Team Name :- **CRYPTO-BUDS**
Team Leader :- **DEVAISHI TIWARI**
College :- **IIT GUWAHATI**
College Code :- **U-0053**

CONTENTS

1. PROBLEM STATEMENT
2. CURRENT SCENARIO
3. OVERVIEW
4. WHY BLOCKCHAIN
5. USE CASES
6. TECHNICAL WORKFLOW

PROBLEM STATEMENT

GAIL procures most of the raw materials, items, office supplies and services through public procurement portal based on tendering system. We need to develop a blockchain secured system to control this procurement process.

The implementation should use Open source technologies like Hyperledger Fabric or equivalent and be able to show simple and good UI + clear documentation with out of the box thinking.

CURRENT SCENARIO

‘Govt tendering, procurement systems not start-up friendly’

Swathi Moorthy | New Delhi | Updated on January 22, 2019 | Published on January 22, 2019



Over 800 start-ups are registered on Government e-Market (GeM) place, but orders are yet to come - Thulasi Kakkat

Most cannot meet eligibility norms such as turnover in previous years

Despite the government's focus on nurturing the start-up ecosystem, getting a government contract continues to be challenge for start-ups.

Current procedures are **dysfunctional** with **asymmetric** emanating of **information** which leads to corruption in allocation and monitoring of tenders.

This causes major uncertainty in the whole system and **hinders transparency, accountability and equity**, all indispensable aspects of an effective procurement system.

These problems also **increase the costs** of the goods, construction works, and services provided through procurement during a period of severe public finance constraint.

OVERVIEW

To develop a **blockchain powered procurement system** to ensure transparency and break the cycle of systematic corruption and ignorance in the whole process.

This will be done by **automatic bid evaluation** (independent from any GAIL user interference) and assigning the project to the best applied bid in terms of **various factors** like price, contractor rating , time period of project, national standards qualification of products, quality of products/services and past relations with GAIL.

Automatic secure payment equal to the one decided at the time of project allocation is also handled after GAIL accepts the services.

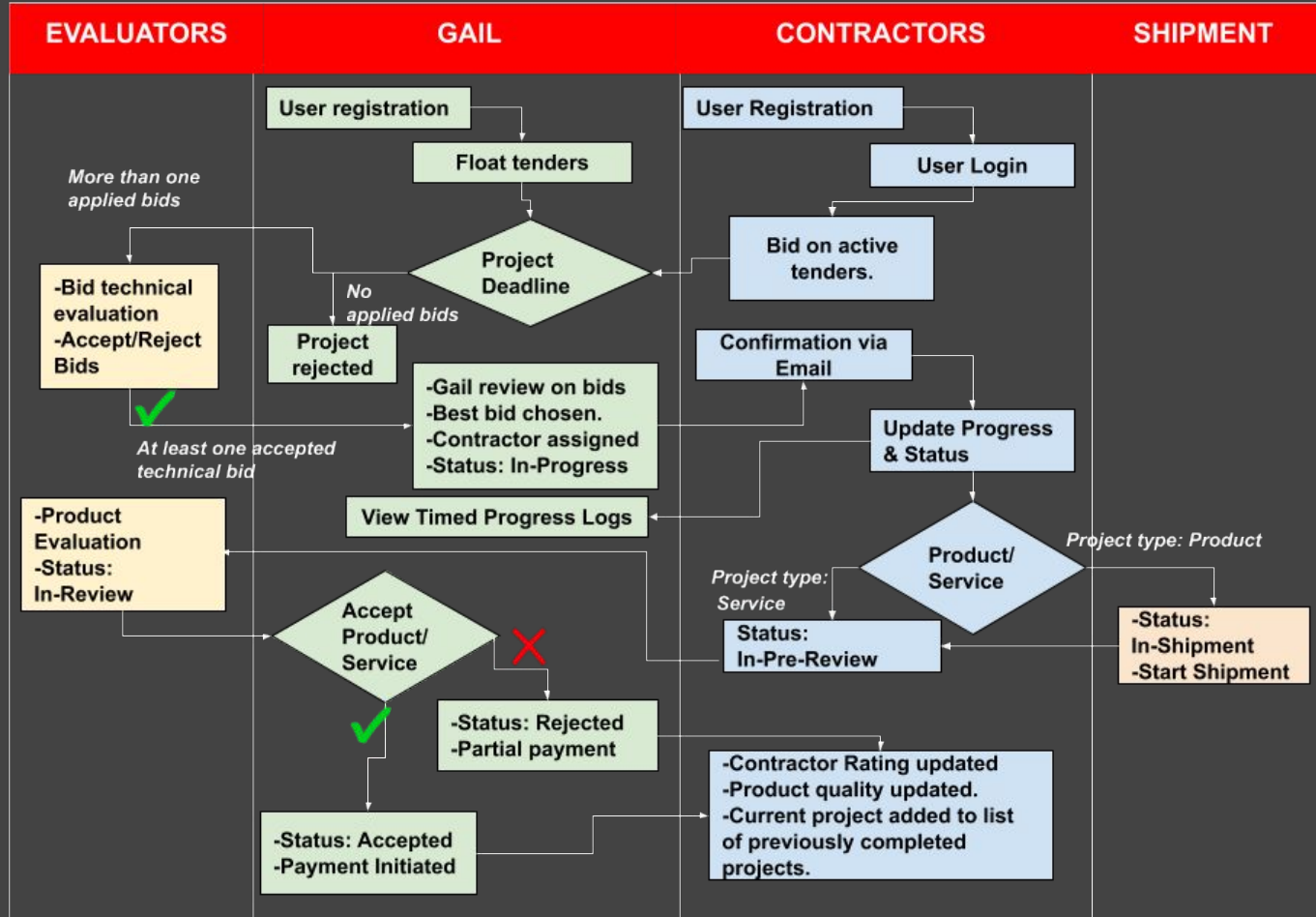
Maintain **log** of every process and automation wherever possible to ensure no delays and misuse.

Why Blockchain ?

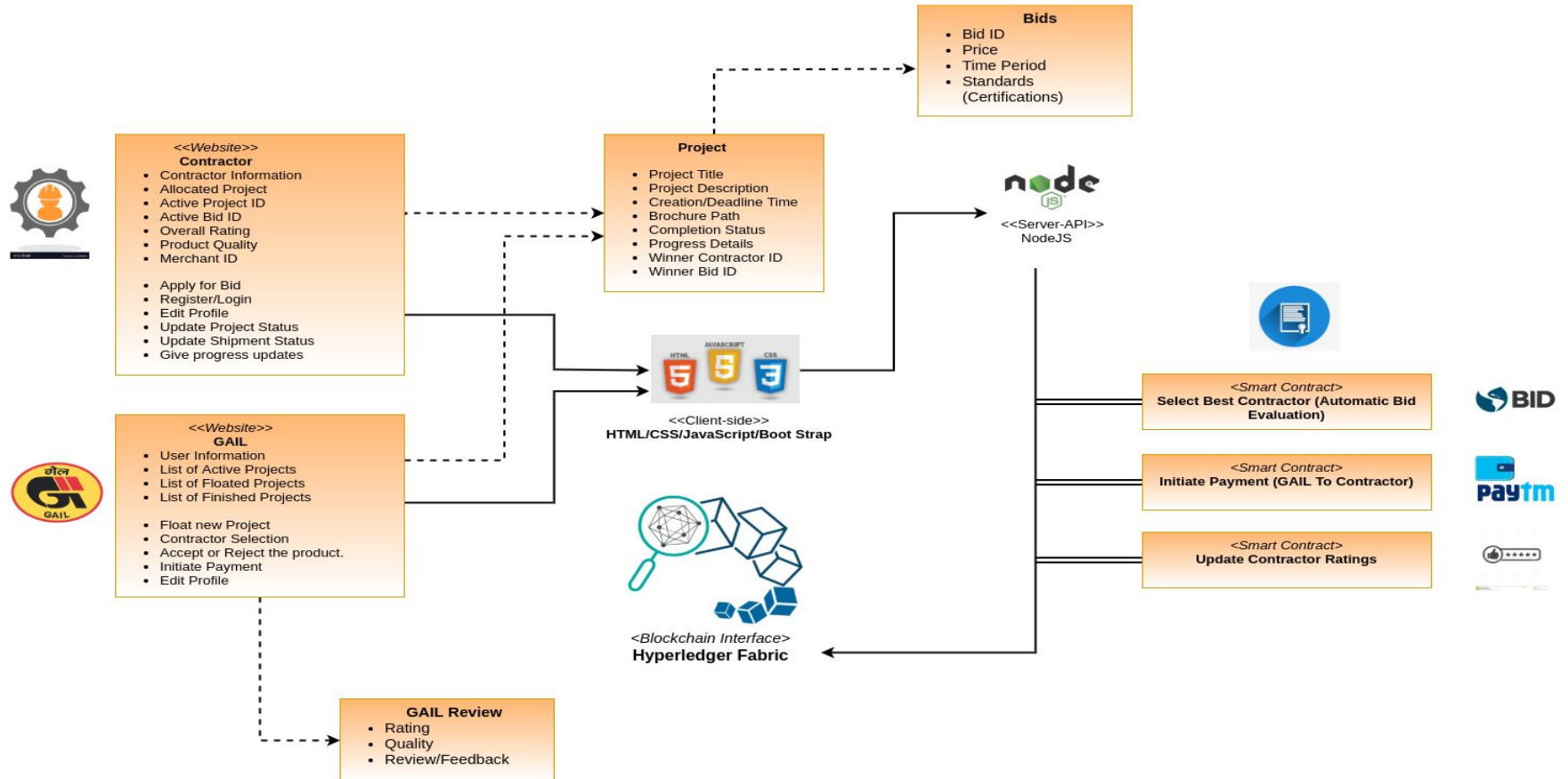
- A blockchain is a **distributed, append-only database (ledger)**, maintained by a **decentralized computing network** running software that determines the **consensus** state of the database.
- There is **no central administrator**, but every computer in the network keeps a full **copy of the database** and processes every transaction.
- Such a system offers a lot of **advantages** over a traditional database.



USE CASE: DOMESTIC



TECHNICAL WORKFLOW



CONTENTS

1. MAJOR COMPONENTS
2. WHY HYPERLEDGER FABRIC?
3. STRUCTURE OF BLOCKCHAIN NETWORK
4. CHANNEL CONFIGURATION
5. TECHNICAL WORKFLOW OF BLOCKCHAIN NETWORK
6. BID EVALUATION ALGORITHM
7. PROJECT STATUS LIFECYCLE (SUPPLY CHAIN)
8. CONCLUSION

MAJOR COMPONENTS

1. **Hyperledger Fabric** contains blockchain network configuration files, Docker images.
2. **Chaincode** implements the core business logic using smart contracts.
3. **Node.js Server** contains the server side backend code abstracted by a REST API service. This abstraction provides us with functionality to migrate/extend to some other client service (e.g Android).
4. **GAIL Client** contains the GAIL client website.
5. **Contractor Client** contains the Contractor client website.

WHY HYPERLEDGER FABRIC?

Hyperledger Fabric is an **open source enterprise-grade permissioned distributed ledger technology (DLT) platform**, designed for use in enterprise contexts, that delivers some key differentiating capabilities over other popular distributed ledger or **blockchain** platforms. Let's explore these differentiating features in more detail.

- **Permissioned vs Permissionless Blockchains**
- **Smart Contracts**
- **Privacy and Confidentiality**
- **Pluggable Consensus**
- **Hyperledger Fabric vs Ethereum**

WHY HYPERLEDGER FABRIC ?

Permissioned vs Permissionless Blockchains

In a **permissionless blockchain**, virtually anyone can participate, and every participant is **anonymous**. In such a context, there can be **no trust** other than that the state of the blockchain.

Permissioned blockchains, on the other hand, operate a blockchain amongst a set of known, **identified** and often vetted participants operating under a governance model that yields a certain **degree of trust**.

Smart Contracts

A **smart contract**, or what Fabric calls "**chaincode**", functions as a **trusted distributed application** that gains its security/trust from the blockchain and the underlying **consensus among the peers**. It is the **business logic** of a blockchain application.

WHY HYPERLEDGER FABRIC ?

Pluggable Consensus

Consensus is defined as the **full-circle verification** of the correctness of a set of transactions comprising a block.

Since consensus is **modular**, its implementation can be **tailored** to the trust assumption of a particular deployment or solution.

Privacy & Confidentiality

Channels and private data collections enable **private** and **confidential multilateral transactions** that are usually required by competing businesses and regulated industries that **exchange assets** on a **common network**.

STRUCTURE OF BLOCKCHAIN NETWORK

- Membership Service Provider (MSP)
- Ledger
- Smart Contracts and Chaincode
- Channels
- Peers
- The Ordering Service

STRUCTURE OF BLOCKCHAIN NETWORK

Membership Service Provider (MSP)

Certificate Authorities issue identities by generating **public** and **private key**.

The **MSP** on the ordering service contains the **peer's public key** which is then used to verify that the **signature** attached to the transaction is valid. The **private key** is used to produce a signature on a transaction that only the corresponding public key can match.

Thus, the MSP is the mechanism that allows that identity to be **trusted** and **recognized** by the rest of the network without ever revealing the member's private key.

Ledger

A **ledger** consists of two distinct, though related, parts – a **world state** and a **blockchain**.

Firstly, there's a **world state** – a **database** that holds current values of a set of **ledger states**.

Ledger states are expressed as **key-value pairs**, and can be changed frequently, as states can be **created, updated** and **deleted**.

Secondly, there's a **blockchain** – a **transaction log** that **records** all the changes that have resulted in the current the world state. The blockchain data structure once written, **cannot be modified**; it is **immutable**.

STRUCTURE OF BLOCKCHAIN NETWORK

Smart Contracts and Chaincode

From an **application** developer's perspective, a smart contract, together with the ledger, form the **heart** of a Hyperledger Fabric blockchain system.

A **smart contract** defines the **executable logic** that generates new facts that are added to the ledger. A **chaincode** is typically used by **administrators** to group related **smart contracts** for **deployment**, but can also be used for low level system programming of Fabric.

Channels

Hyperledger Fabric also offers the ability to create **channels**, allowing a **group of participants** to create a **separate ledger** of transactions. This is an especially important option for networks where some participants might be **competitors** and not want every transaction they make — a **special price** they're offering to some participants and not others, for example — known to every participant. If two participants form a channel, then those participants — and no others — have **copies of the ledger** for that channel.

STRUCTURE OF BLOCKCHAIN NETWORK

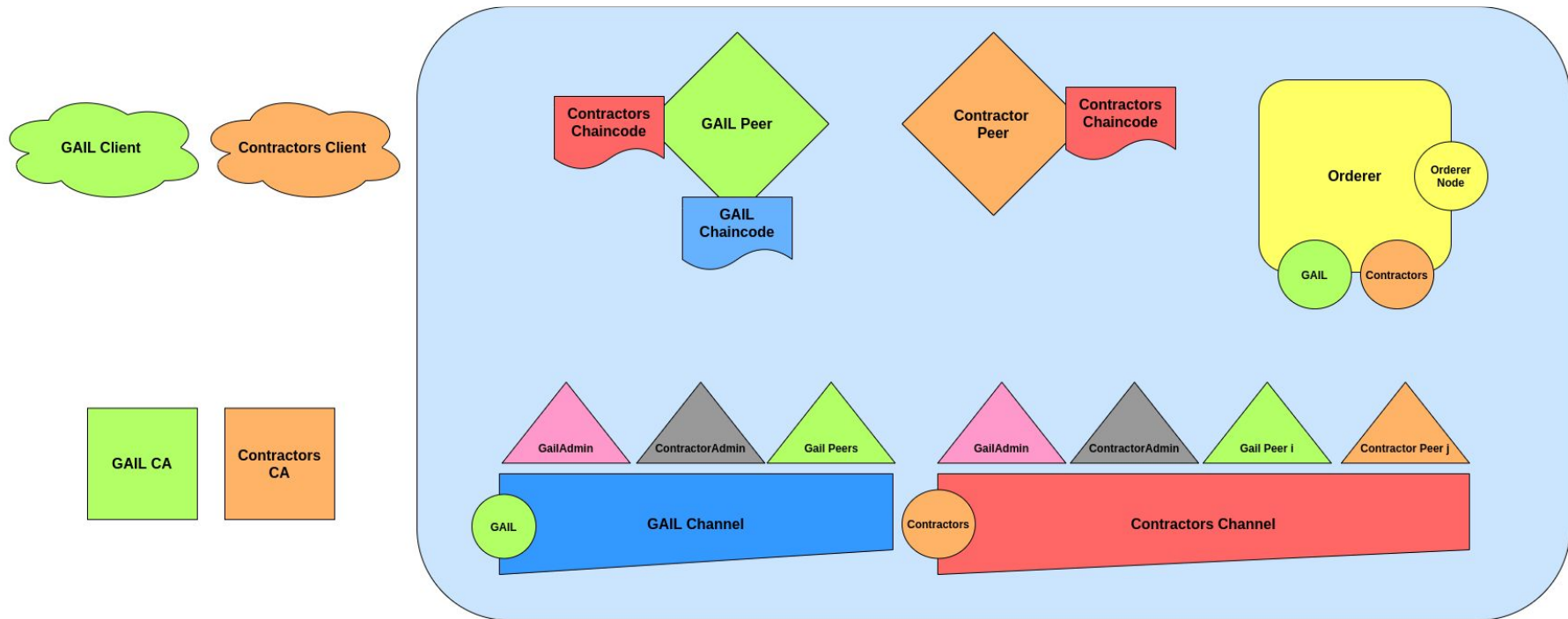
Peers

A blockchain network is comprised primarily of a set of **peer nodes** (or, simply, peers). Peers are a **fundamental** element of the network because they **host ledgers** and **smart contracts**. Smart contracts and ledgers are used to **encapsulate** the shared processes and shared information in a network, respectively. These aspects of a peer make them a good starting point to understand a Fabric network.

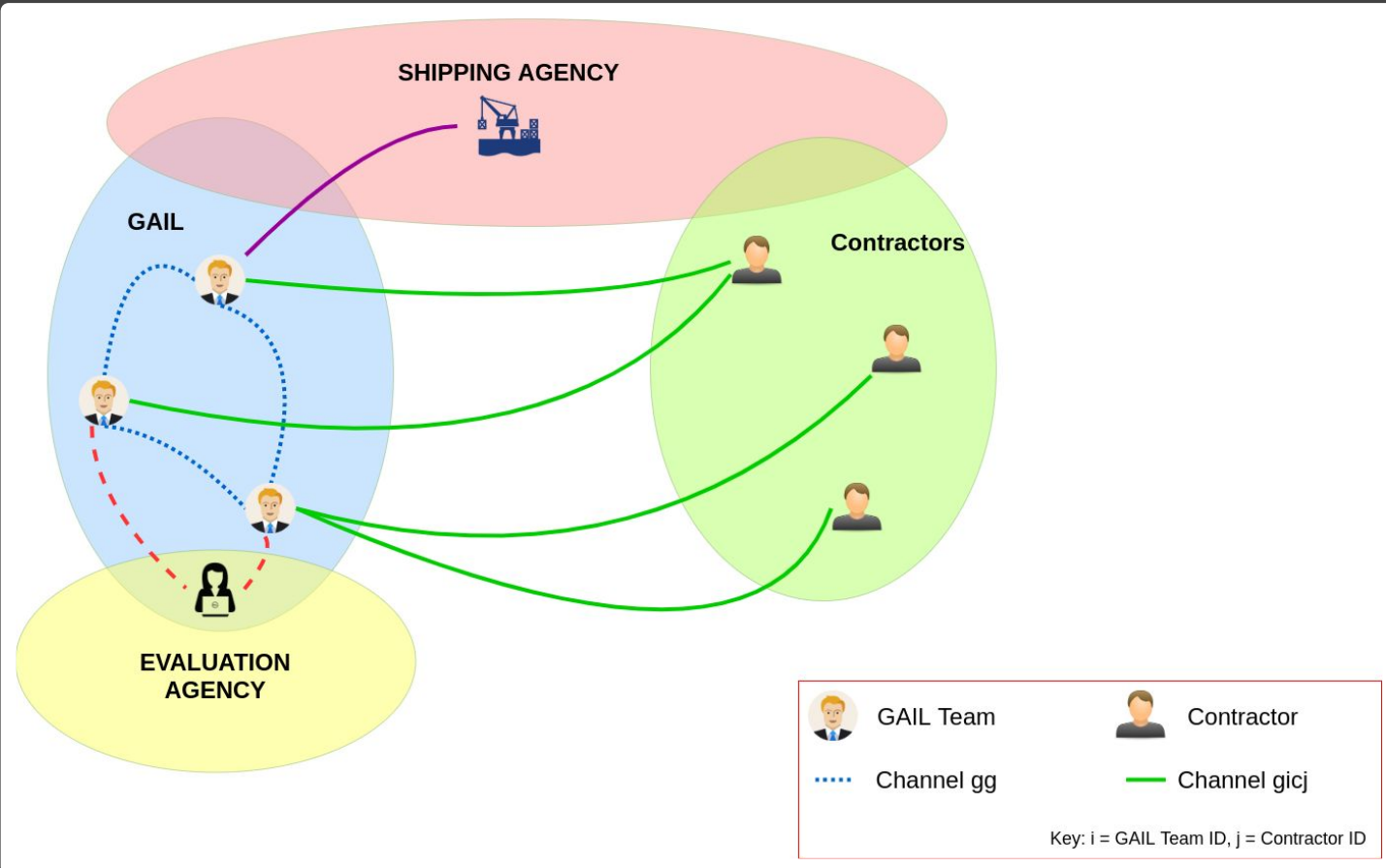
The Ordering Service

A node called an **orderer** (it's also known as an "**ordering node**") does this **transaction ordering**, which along with other orderer nodes forms an ordering service. Because Fabric's design relies on **deterministic consensus algorithms**, any **block validated** by the peer is guaranteed to be final and correct. Ledgers cannot fork the way they do in many other distributed and **permissionless blockchain networks** like **Ethereum** and **Bitcoin**.

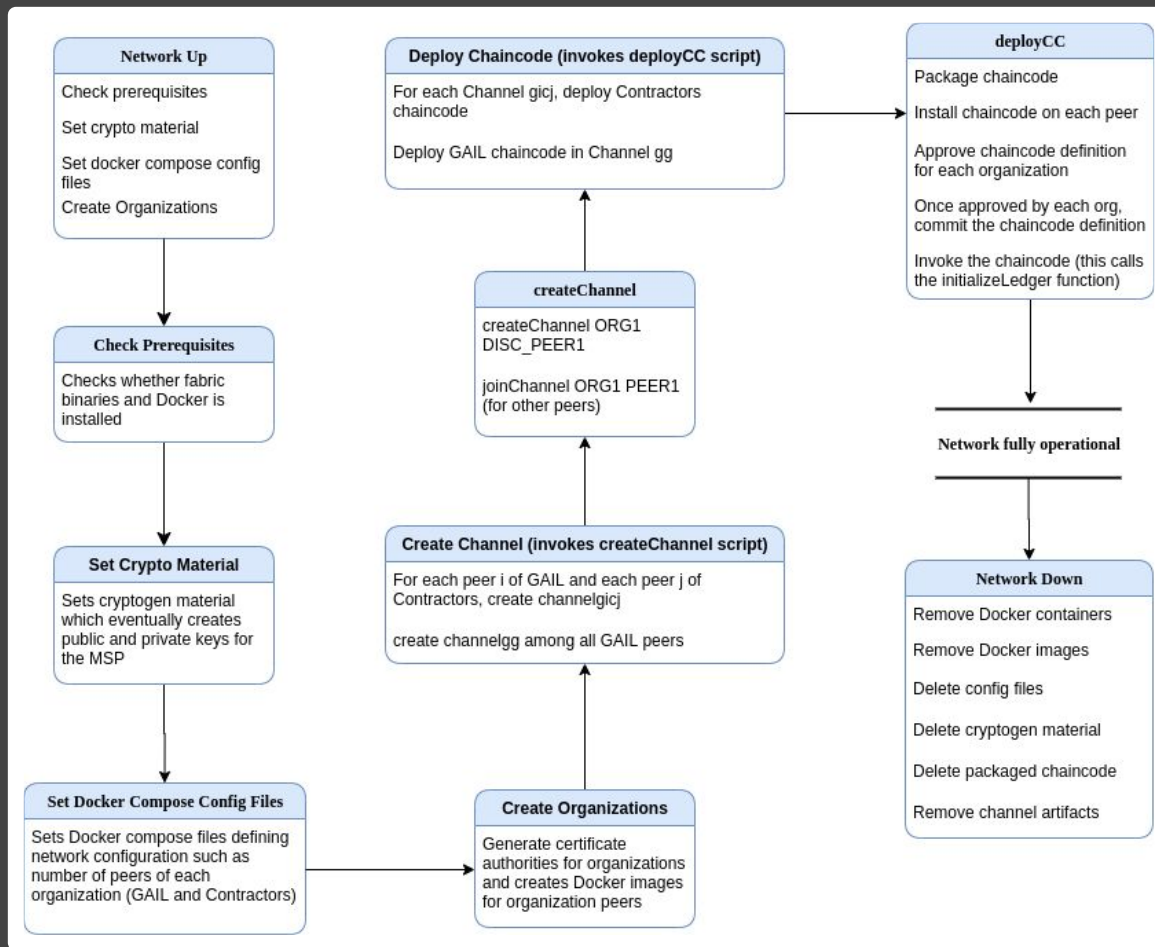
STRUCTURE OF BLOCKCHAIN NETWORK



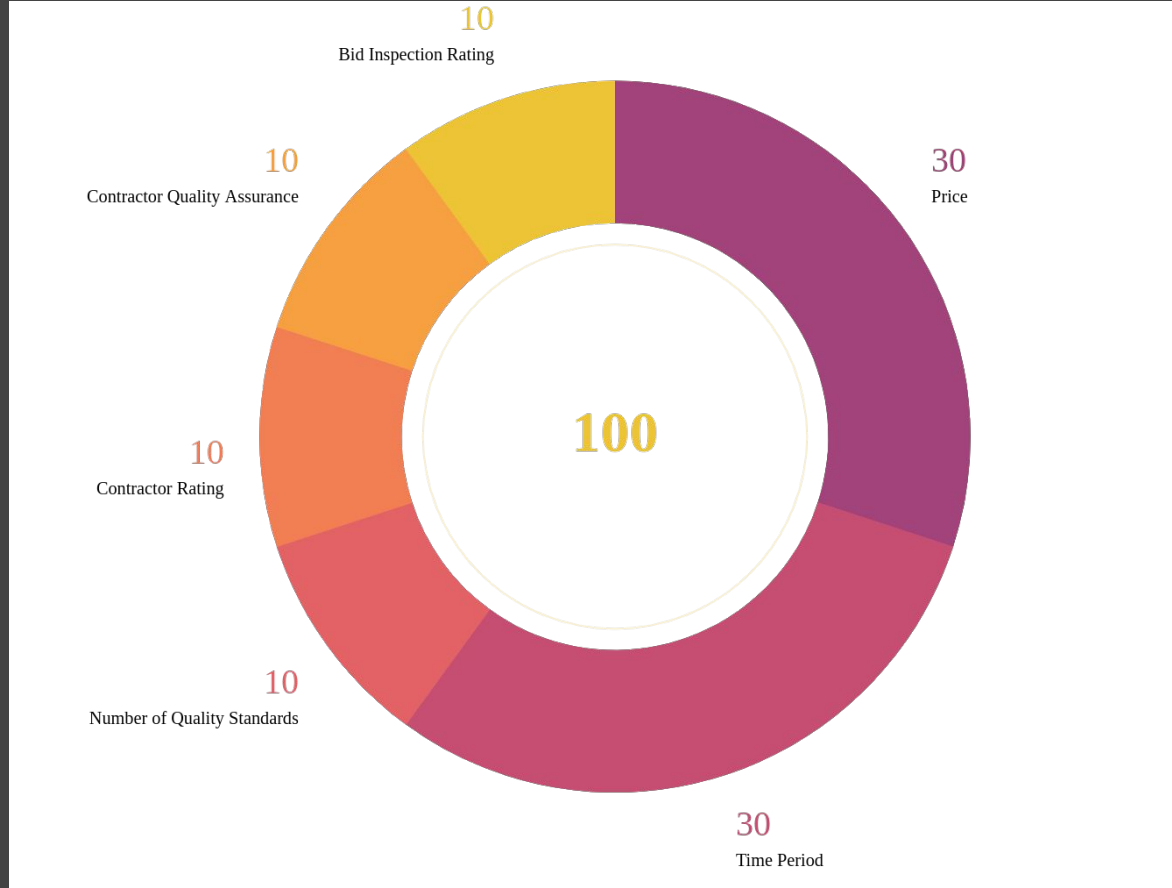
CHANNEL CONFIGURATION



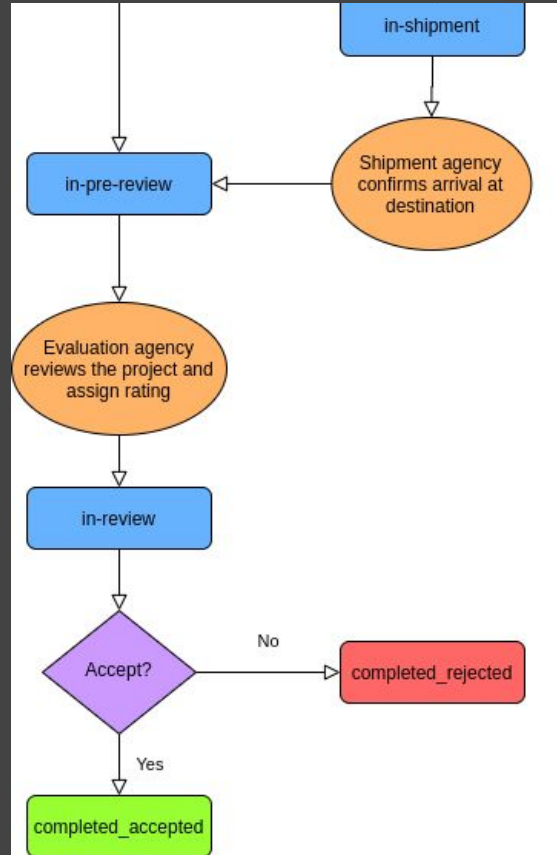
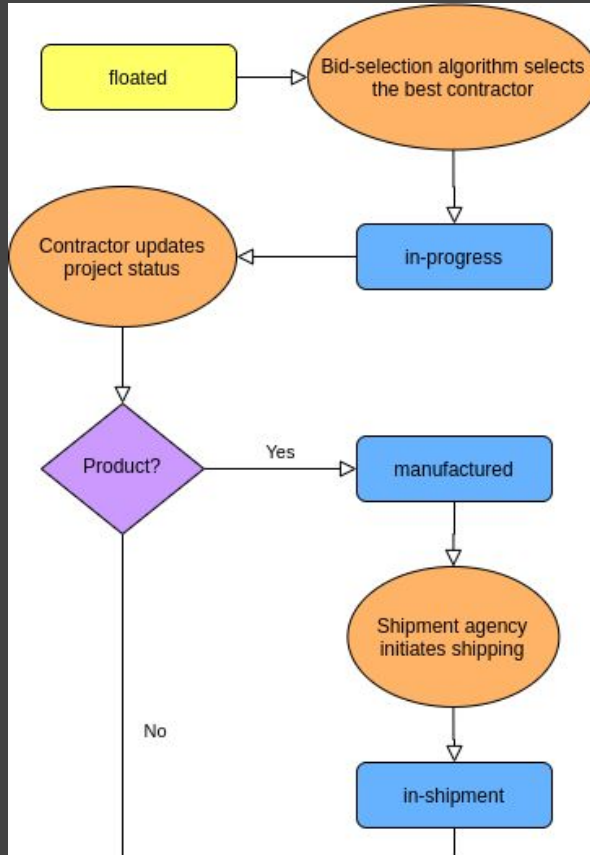
TECHNICAL WORKFLOW OF BLOCKCHAIN NETWORK



Bid Evaluation Algorithm (Weights)



Project Status Lifecycle



CONCLUSION

1. UNIQUE PROPOSITION
2. WHY HYPERLEDGER FABRIC?
3. HYPERLEDGER FABRIC VS ETHEREUM
4. PLUGGABLE CLIENTS
5. DOCUMENTATION LINK

UNIQUE PROPOSITION

- **HYPERLEDGER FABRIC:** Private and permitted blockchain system with pluggable consensus and smart contracts specially designed for developing enterprise solutions.
- **PLUGGABLE CLIENT:** Node.js Server contains the server side backend code abstracted by a REST API service. This abstraction provides us with functionality to migrate/extend to some other client service (e.g Android).

WHY HYPERLEDGER FABRIC ?

Permissioned vs Permissionless Blockchains

In a **permissionless blockchain** like **Ethereum**, virtually anyone can participate, and every participant is **anonymous**. In such a context, there can be **no trust** other than that the state of the blockchain.

Permissioned blockchains, on the other hand, operate a blockchain amongst a set of known, **identified** and often vetted participants operating under a governance model that yields a certain **degree of trust**.

Pluggable Consensus

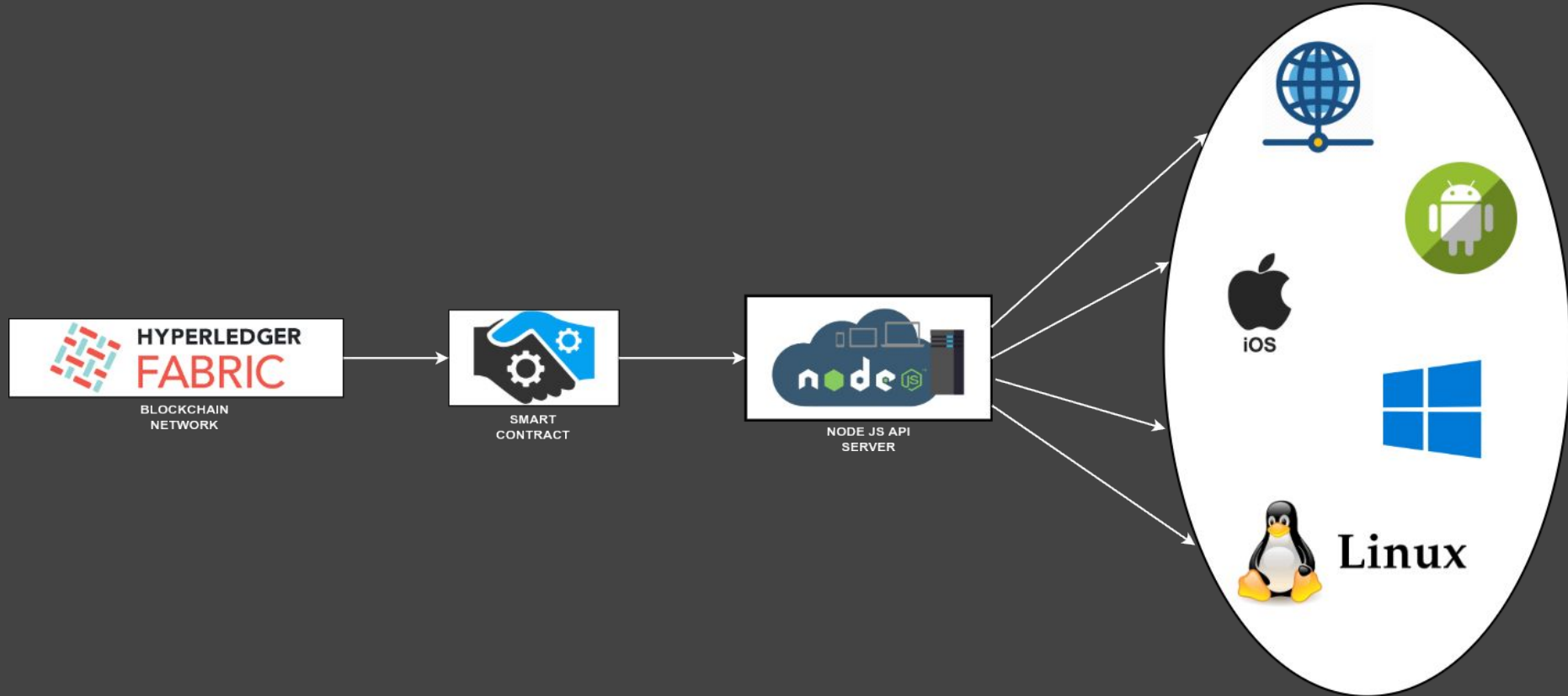
Consensus is defined as the **full-circle verification** of the correctness of a set of transactions comprising a block.

Since consensus is **modular**, its implementation can be **tailored** to the use case of the business application requiring **some orgs to endorse one component and other orgs to endorse other components**, trust assumption of a particular deployment or solution.

HYPERLEDGER FABRIC VS ETHERIUM

- Ethereum is an open source public blockchain network, whereas Hyperledger Fabric is a **private and permitted blockchain system** specially designed for developing enterprise solutions.
- As has been extremely well documented, the blocks in bitcoin and Ethereum have a storage issue.
- Then we have throughput problems since big enterprises need to deal with millions of transactions per day with near zero latency.
- Public blockchains, especially the ones that follow the proof-of-work protocol require an immense amount of computational power to solve hard puzzles.
- Ethereum is extremely transparent and every transaction is visible to everyone in the network. Hyperledger allows confidential transactions (**bids not visible to other contractors**). Consequently, it gives businesses the flexibility and security to make transactions visible to select parties having correct encryption keys.

PLUGGABLE CLIENTS



DOCUMENTATION LINK

- **Chaincode Documentation:** [Click here](#)
- **Node.js Server API Documentation:** [Click here](#)

Thank You