LaneAxis Blockchain Architecture Overview

To utilize the transparent and immutable nature of the Blockchain while also requiring permissioned visibility of certain data, the **LaneAxis solution uses Hyperledger technology**. Each transaction that occurs between the shippers and the carriers is denominated in ERC20 Axis tokens.

Individual channels per shipping contract are created with the shipper and the carrier acting as nodes. **The channels ensure that trip related specifics such as the agreed upon cost for service is kept private between the contracted parties and restricts viewing by other participants of the rest of the Hyperledger.** This ensures that the shipment related data can
be kept private between the shipper, the carrier, and the drivers/agents designated by the carrier, while maintaining transparency and immutability of the contract.

Rationale for using the Hyperledger

1. Service Oriented Architecture (SOA)
2. Real-time data tracking and traceability of carriers
3. Proof of Participants' identities with open identity protection Audit Trail of activity and transactions
4. AI Powered carrier and shipper recommendation engine
5. Fraud detection
6. High throughput rate, high capacity and low latency
7. Test driven development
8. Maintaining privacy of information between two parties such as amount paid for shipping not visible for all the participants.

Key Participants

1. **Bot**

   A bot is a special type of peer with all the permissions. This user will be part of all the channels in the Hyperledger.
2. **Shippers**

Shippers are the organisations who will create a shipment request thus creating an opportunity to generate value on the Hyperledger. Every shipper runs a single node on the Hyperledger.

3. **Carriers**

Carriers are the organizations that will be accepting the orders and will be allocating the shipment resources towards a completion of a specific shipment. Each carrier will run a hyperledger node which shall be used for interaction with the REST interface for drivers and associated parties.

**Components of a Hyperledger**

**Channels**

Channels are the key to Hyperledger Fabric privacy and access control. Each channel on the LaneAxis Hyperledger represents a shipping contract entered into by the shipper and the carrier. The terms of the contract agreed upon are encoded as shared chaincode between the nodes of the channel. A LaneAxis channel has the following stages:

**Creation**

When the order is accepted by the carrier and confirmed from the shipper, the configuration transaction (configtx) is stored in a special block (aka genesis block for that channel) that contains not only incremental configuration updates but the entire configuration so that when a peer needs to know the channel’s configuration, it only needs to get the latest configtx block.

**Invitation**

More actors can be invited to a channel upon creation. These will be the drivers, insurance company, and such other ancillary parties.
Access Control
The drivers, either directly (as optional nodes), or indirectly (through REST APIs) transmit all IoT related information to the channel. The channel prevents non-participants of the channel from accessing/modifying any information within the channel. One common user to all the channels is the bot, which enables smart recommendations and triggers smart alerts during a trip as and when required.
Process Flow

The LaneAxis implementation of the Hyperledger is a decentralized ledger technology that brings together the shippers and the carriers as the nodes. When a shipper creates a request which includes the shipping details such as pickup point, the destination, and carriage load etc., the request is updated on all the nodes. The LaneAxis bot is also a node, and is distinct by the fact that it possesses all the permissions to the information privy to the channels of the Hyperledger.

The LaneAxis bot, creates a list of the carriers best suited to undertake the contract and forwards it to the shipper. This list is created from the total number of carrier auto-acceptances and the manual acceptances. This list of recommendations is created by the LaneAxis bot, based on a patented algorithm that uses multiple factors such as distance from the pick-up point, rating of the carrier, rating of the shipper, by applying specific weights to these factors. The algorithm ensures that the list of recommendation of carriers is not uniform, but includes a mix of experienced as well as new carriers. This is purposefully encoded into the algorithm to protect the long-tail.
The shipper receives this list and makes the most suitable choice. In case the shipment order is accepted by only one carrier, the LaneAxis bot automatically assigns the shipment to that carrier.

Once a shipment contract between the shipper and the carrier is finalised, the **Shipper is responsible for creating a new channel by creating a genesis block**. The shipper must also add the respective carrier to the channel. **This channel contains a shared chaincode with the shipper and the carrier acting as the nodes which will endorse the trip initiation and completion related transactions on the Hyperledger.** The LaneAxis bot is also a part of this channel by default and is responsible for calculating the internal reputation scores based on all activities in the Hyperledger and risk management.
Since shippers might need multiple shipments to be transported and the carriers might be transporting shipments of more than one shipper, the **Hyperledger allows each shipper and carrier to be a part of multiple channels**. The information contained within one channel cannot be accessed by another channel, even if the shipper and the carrier are the same, as long as the shipment is different. **Thus, each channel is a distinct unit and contains information pertinent to one shipment only.** The drivers and other such ancillaries can be assigned as optional nodes to a particular channel by the carrier. However, this step is optional and not mandatory. The driver can be added to the channel only after the shipment contract has been finalized between the shipper and the carrier. If **driver has been added to the channel, he/she can accept the details from the shipper on behalf of the carrier.** If the carrier does not add the driver to the channel, the driver receives the trip related information from the carrier’s node and transmits information back to the carrier’s node via REST APIs.
The driver picks up the shipment from the pick-up location and transmits this information to the shipper, either through the carrier's node, or directly, as the optional node, as the case may be. This displays the best suitable route to the driver who must pass through the route dependent checkpoints in the calculated intervals, within the margin of error. Short detours will not be flagged by the IoT device attached to the driver's truck but a longer detour will be flagged and attract a penalty. **Every such penalty will affect the score (out of 1000) of the carrier and lower his/her chances of getting picked by the LaneAxis bot for another shipment.**

All of the below-mentioned activities of the carrier will attract a penalty and lower their reputation score:

1. Trip cancellation after shipper’s confirmation
2. Delay in pickup
3. Delay in delivery confirmation
4. Incorrect information listed in the profile

The Shipper is also responsible for certain activities and not performing them timely will negatively affect their score as well. The activities that lower a shipper's reputation score are:

1. Failing to accepted carrier’s requests in time
2. Cancelled Shipment after confirmation
3. Payment delays or failure

Once the shipment is carried to the pre-decided destination by the driver, the shipper becomes liable to pay the carrier. Axis tokens of the amount that was mentioned in the contract is transferred by the shipper to the carrier through ERC20 compliant cold wallets. The money is transferred offline as far as the Hyperledger is concerned. All payments are denominated in LaneAxis tokens which are ERC20 tokens. The transactions are processed and recorded on the LaneAxis fork of the Ethereum Blockchain. The transaction details are communicated to the Blockchain by the Hyperledger Fabric through APIs securely and without human-intervention.
For the purpose of the Hyperledger, the carrier node broadcasts on the private channel to the shipper node that the shipment has been delivered. It can also be verified by the data transmitted by the driver to the channel, whether directly, or indirectly. Once verified, the system provides the ability for the shipper to transfer the funds to the carrier's cold wallet. The use of cold wallets is preferred to ensure maximum safety of the funds received/stored in the wallet.
Edge Cases

**What happens if the shipment is delivered after expiry?**
Since the configuration transaction (configtx) has a valid_until value, it cannot be used to confirm the shipment after the expiry.

**What happens if the shipper fails to choose a carrier within the stipulated time?**
If the shipper fails to take any relevant action, no contract is created and the shipper must create a new shipping request and choose from the new list of carriers fetched by the LaneAxis bot.