

ONDC Innovation Hackathon

Innovation Track

Problem Statement (Challenge 9)

The objective of this challenge is to create and maintain a distributed ledger of open data in ONDC

Team Name: NPCI Innovators

GITHUB ID : <https://github.com/harshmalik9423/ONDCHackathon>

Team Members:

- Anushka
- Ashutosh
- Harsh
- Jay
- Sanchit

Contents

Functional Overview	4
Information Management	5
Why Distributed Network Ledger Network	6
Nodes	6
Gossip protocol	6
Network Architecture	7
APIs	7
Key Assumptions	8
How its scalable to 1 million nodes	9
Security of Platform	9

Introduction

Today, e-commerce markets work in silos, with each e-commerce platform connecting only their own merchants and customers to each other.

ONDC's network-centric model integrates these closed ecosystems and allows each player to be discoverable to a wider audience. Once the platforms on the buyer side and seller side implement a set of open Application Programming Interface (API) specifications, it will automatically create a network which enables open, interoperable transactions between users no matter what platform/application they use to be digitally visible. This offers a seamless digital commerce experience for everyone. We proposed to address the challenge 9 of ONDC challenge using open source and standard hyper ledger fabric framework and existing Beckn protocol APIs. We have fulfilled both Mandatory and Optional requirement posed in the challenge:

- a. Design a distributed ledger for a network that sources events, at different levels of granularity, from different nodes of the network, aggregates the data by applying mathematical transforms and synchronizes the resultant data across the nodes of the network; examples of data that may be considered for sourcing include ratings data, aggregate data for sales and inventory etc.
- b. The design should consider a network with at least 1 million nodes.
- c. Develop an open standards based reusable software component that implements (a) above.
- d. Extend the distributed ledger in (a) above to handle exception & edge cases, such as a node joining the network at any time and synchronizing with the latest aggregates, a node becoming offline for some time, having its aggregates synchronized once it becomes online,

Based on our solution, the following platforms/ applications can be onboarded on ONDC as a network participant:

A. Retail Demand - A platform that allows individual consumers to discover and transact with retailers/Kiranas/etc.

B. Retail Supply - A platform that allows retailers/ Kiranas etc. to offer products and services to individual consumers

C. Wholesale Demand - A platform that allows retailers/ Kiranas etc. to discover and transact with wholesalers/distributors/manufacturers

D. Wholesale Supply - A platform that allows wholesalers/ distributors/ manufacturers etc. to offer products to the retailers

E. Logistic Demand - A platform that allows retailers & wholesalers to place orders for delivery services

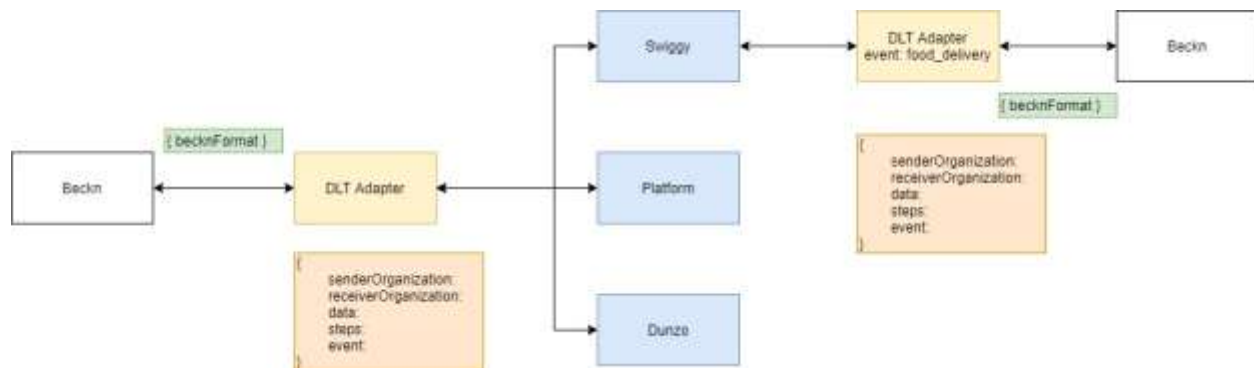
F. Logistic Supply - A platform that allows logistics providers to offer delivery services to retailers & wholesalers

Functional Overview

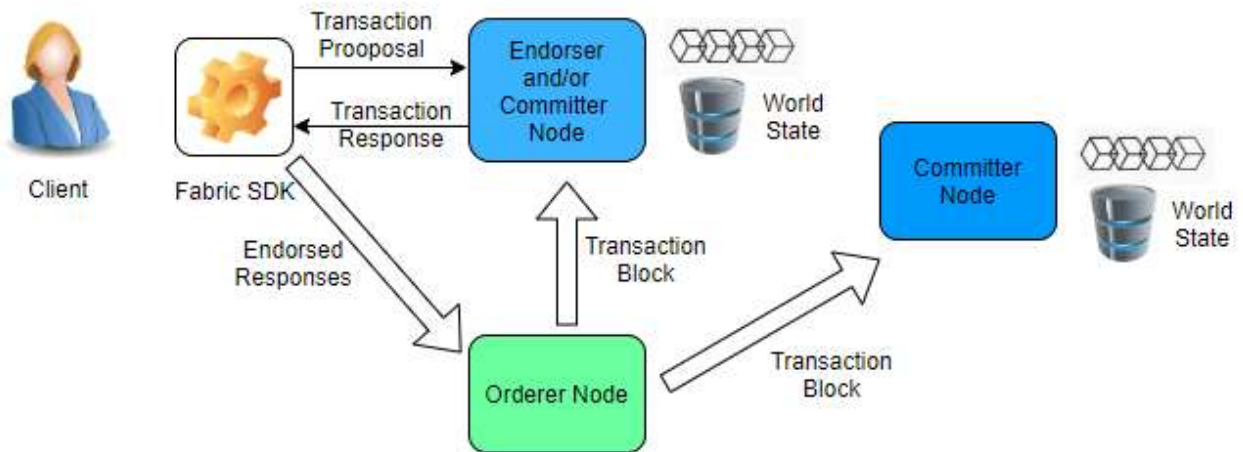
ONDC network comprises buyer nodes, seller (or fulfilment) nodes and gateways which route requests between the buyer and seller / fulfilment nodes. All network participants are registered in the ONDC registry after onboarding. Discovery takes place through the search request and all search requests are broadcasted to all relevant seller / fulfilment nodes in the registry. Responses received from the seller / fulfilment nodes are aggregated by the buyer node and displayed to the user. Hence, all seller / fulfilment nodes are discoverable for a buyer node.

There is no intermediary in this open network. The gateway and registry are used only for search/discovery. During search, the gateway will broadcast the search, allow it to reach all seller platforms and return responses relevant for the search context and list in registry. This can subsequently be accessed by the buyer platform. The role of the registry and gateway does not extend beyond search. Post search, the order lifecycle are bilateral transactions between the buyer platform (where the purchase intent originated) and the selected seller platform. There is no intermediary and these transactions are dual digital signed encrypted

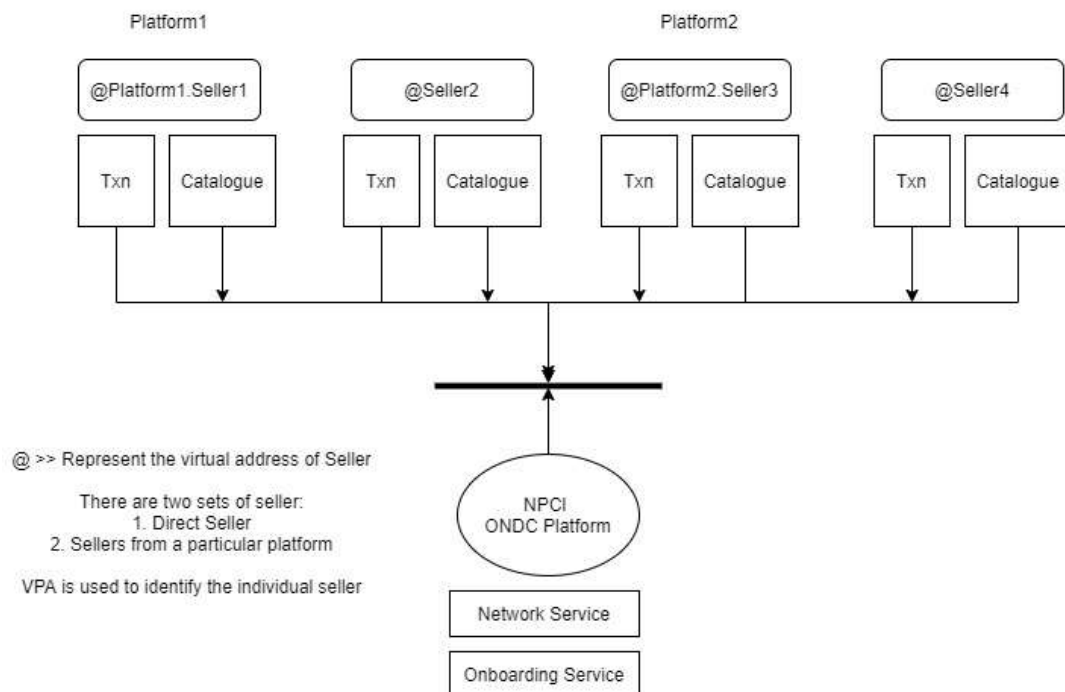
Platform Overview



Transaction Flow



Information Management



Product (Catalog): - using API's any seller can add their products to the onChain DB (CouchDB) in network which will be visible to all (/add_product)

Orders: Transaction ID and Transaction hashes are stored/Updated in onchain DB and the complete order data is stored/updated in offChainDB (/confirm)

Aggregated Data: the data is getting aggregated using API calls (which is triggered automatically using CRON jobs or on postfulfillment) and the calculated data is stored in the network onChainDB (/aggregation)

Why Distributed Network Ledger Network

The peer itself will always attempt to catch up to current block height. It can catch up in one of three ways:

- 1) If the peer is a gossip leader in the org, then it requests a stream of blocks from an orderer via deliver API, starting at the peers current block height.
- 2) If the peer is not a gossip leader and is current or just a little behind (within a small threshold), then it gets blocks via gossip from the org's leader peer (or potentially another peer in the org)
- 3) If the peer is not a gossip leader and is way behind (beyond the threshold), then it gets blocks via block transfer from another peer (can be cross-org).

Nodes

BuyerORG -> Buyer Admin -> onboard buyers on the blockchain network using API's

SellerORG -> Seller Admin -> onboard all sellers on the blockchain network using API's

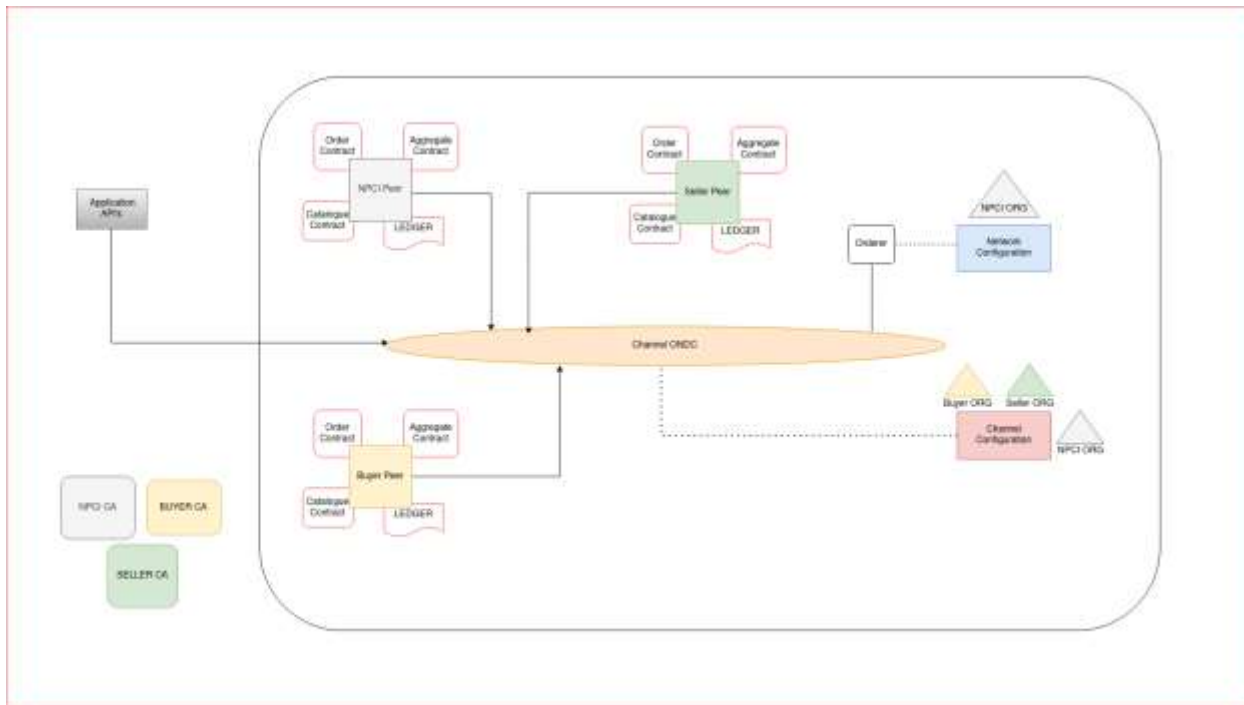
NPCIORG -> NPCI Admin -> add the Orgs in the network for platforms (Amazon, flipkart, myntra)

(/user): The user can send its registration details to get enrolled in the network. We can keep checks during enrolling the users to ensure the validity of the users. According to the organization user will get the roles and access in the network.

Gossip protocol

Peers are always catching up to current orderer's block height, regardless of whether the peer is milliseconds, seconds, or minutes behind current orderer's block height, it is considered 'healthy' in that it can accept chaincode requests. However, the further behind the peer is, the more likely it is that its endorsements will ultimately get invalidated due to staleness, therefore it is good practice to send endorsement proposals to peers with maximum block height. Service Discovery returns height of known peers for this purpose, also you can query the peers height directly from an SDK, or look at the ledger_blockchain_height metric

Network Architecture



APIs

Beckn

/Search

The BAP will send the search intent to the BPP for services

/Select

The BAP will send items to the BPP to build the order

/Init

The BAP will send the billing and/or shipping details the BPP to initialize the order

/Confirm

The BAP will send the confirm request to the BPP after agreeing to the payment and fulfilment terms sent by the BPP.

CALLBACKS

[/On_Search](#)

The BPP will send the catalog of items/services matching the search intent to the BAP.

[/On_Select](#)

The BPP will send the draft order with the quoted price to the BAP.

[/On_Init](#)

The BPP will send the draft order with the payment and fulfilment terms to the BAP.

[/On_Confirm](#)

The BPP will send the active confirmed order object to the BAP.

Proposed Platform

[/product](#)

To register product by seller

[/getAgg](#)

To get aggregated data

[/user](#)

To Enroll new buyer and seller

Key Assumptions

1. Transactions are already captured in blockchain between buyers and sellers
2. Transaction information captures 80% details of Beckn protocol specs across its lifecycle ie. Search,Order,Fulfillment and Post fulfillment
3. The transaction happens between buyer platform and Seller platform or directly with Seller.
4. Seller can directly enroll into the blockchain network and start participating in the ecosystem.
5. Seller once on boarded as peer in the blockchain network can be discoverable at Buyer platform
6. We assumed a registry feature in blockchain which provide user friendly Virtual addressed to fetch Seller information at aggregated level.

7. For time being, we allowed seller to add inventory details in the catalogue itself, in subsequent stages this will be separated out

How its scalable to 1 million nodes

We have leveraged two ways to achieve scalability in the solution:

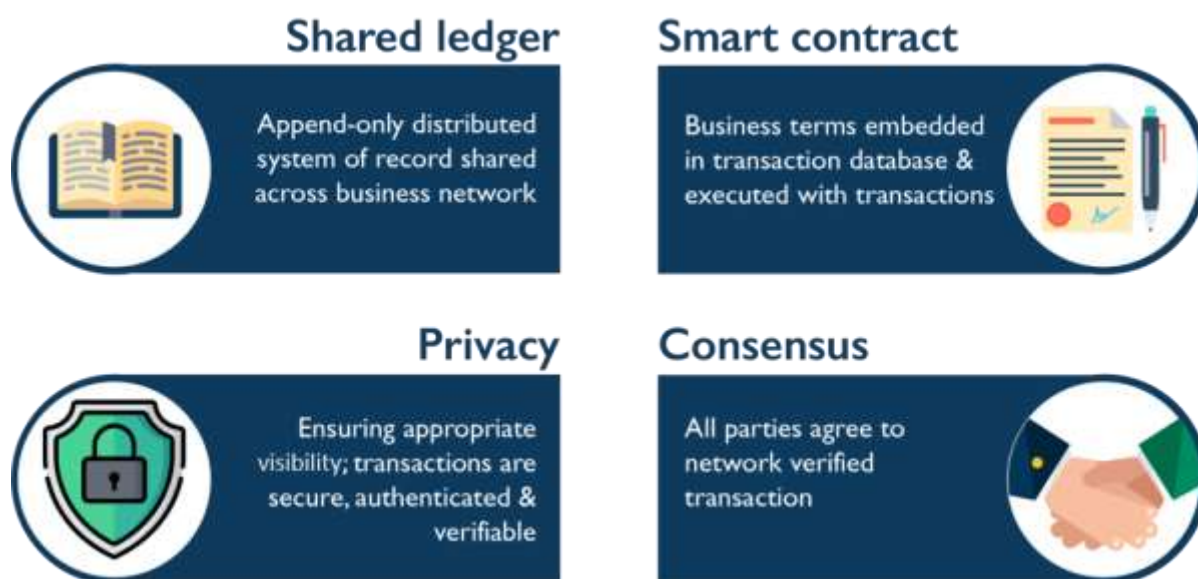
1. Gossip protocol : Peers are always catching up to current orderer's block height, regardless of whether the peer is milliseconds, seconds, or minutes behind current orderer's block height, it is considered 'healthy' in that it can accept chaincode requests. However, the further behind the peer is, the more likely it is that its endorsements will ultimately get invalidated due to staleness, therefore it is good practice to send endorsement proposals to peers with maximum block height. Service Discovery returns height of known peers for this purpose, also you can query the peers height directly from an SDK, or look at the ledger_blockchain_height metric

2. Leveraging the principle of On-chain and off chain data in the blockchain and roll ups:

- On-chain -- Transaction hash (SHA 256), Transaction ID, Buyer of Seller core information, Rolled up- Aggregated information at seller level
- Off-chain - Transaction details
- Platform node - is aggregating the information at each nodes and making it available to the network

Security of Platform

Hyperledger Fabric is a permissioned blockchain where each component and actor has an identity, and policies define access control and governance.



More details of model - https://hyperledger-fabric.readthedocs.io/en/release-2.2/security_model.html