ONDC Integration Hackathon

Notes to the Hackathon Participants

1. This document is the primary document of reference for the ONDC Integration Hackathon.
2. Please go through the details carefully and raise your queries before **11:00 AM** on **7th January 2022**.
3. Please post your queries in the Slack channel.

1. Instructions to Participants

Participants are requested to follow these instructions that will be applicable during the Integration Hackathon.

a. **Setup entry in ONDC registry**

- **Participants should setup an instance with URL that is publicly accessible and make an entry in the ONDC registry;**
- **Please reach out to us on the Slack channel to get yourself added to the registry;**

b. **Reference application, sample code:**

- Beckn BAP reference app (https)
- Beckn BAP reference app (http)
- Generating key pairs, using authorization headers for digital signature
- Signing Beckn APIs in http

c. **Github repository**

- **Use Case Scenarios**
- Postman Collection
- Instructions for using Postman
- Sample JSON

d. **Communication channels:**

- Slack
- **Checkin calls**
- **Submission Sheet**

2. Overview of Use-Case Scenarios

Use case scenarios simulate real world scenarios of transactions envisaged on ONDC, with different buyer and seller distributed apps, connected in a decentralized mode.

These use-case scenarios will cover 2 different domains (Retail and Logistics), and will include the following:

- ***Scenario 1** - **Execute a complete retail transaction, using a buyer app and seller app, resulting in order confirmation;***
- ***Scenario 2** - **Execute a complete retail transaction, using a buyer app and seller app, resulting in order confirmation and fulfillment by any one logistics provider;***
- ***Scenario 3** - **Execute a complete retail transaction, using a buyer app and more than one seller app, resulting in order confirmation and fulfillment by more than one logistics provider;***

Note: Authorization header validations for APIs will be enabled during the hackathon, with prior notification, to ensure that participants get a chance to test their signing & verification functionality.

*The user journey and detailed transaction steps, for each scenario, is defined below. Any participant can decide to execute **one or more of these scenarios**.*

3. Detailing - Scenario 1

3.1 User Journey

Saurabh decides to order dinner for his family. Saurabh opens an app called “Open Commerce for All”.

a. Saurabh enters his location as “Malviya Nagar, Delhi”. He types in the category of “Quesadillas” and waits.

b. Saurabh sees a list of outlets that sell Quesadillas. He sees that his favourite Quesadillas are available from a Chili's outlet near his location.

c. He selects the following by clicking “Add to Cart”:

- Chipotle Quesadillas - 1
- Parmesan Crusted Cottage Cheese Quesadillas - 2
- Fresh Mex Quesadillas (with cajun rubbed cottage cheese) - 1

d. Saurabh adds these items to the cart and is about to checkout when he realizes he should also order a Salad. He searches for the category “Salad”.

e. He sees that Chili's has several options for Salad.

f. He selects “Caribbean Salad” by clicking “Add to Cart”. He then verifies his order by clicking on “View Cart”.

g. Saurabh then proceeds with the following steps to confirm the order:

- Click on “Checkout”;
- Clicks on “Add Shipping Details” and provide all details including the landmark;
- Click on “Proceed to Pay”;
- Following payment options are available - “Cash on Delivery”. He selects this option and clicks “Confirm”.

h. After confirmation of the order, Saurabh is redirected to an order screen that says, “Order confirmed”.

3.2 Transaction Steps

a. Retail Network search / on_search, resulting in following steps (note - this will be repeated once per category) :

1. Buyer App : add search intent, location and call search to ONDC Gateway
2. ONDC Gateway : broadcast search
3. Seller App(s) : add list of providers with matching items and return on_search to ONDC Gateway
4. ONDC Gateway : forward on_search from Seller App(s)
5. Buyer App : view catalog of any provider from Seller App(s) (direct search to Seller App)
6. Seller App(s) : send catalog of selected provider to Buyer App with matching items from original search (direct on_search to Buyer App)

7. Buyer App : select category ID inside Seller App(s) provider catalog and call search (direct search to Seller app)
8. Seller App(s) : add items under the selected category ID and return on_search

b. Retail Network select / on_select, resulting in following steps:

1. Buyer App: add Seller App(s) item to cart and call select
2. Seller App(s): add item, calculate quote and return on_select
3. Buyer App: remove Seller App(s) item from cart and call select
4. Seller App(s): remove item from cart, calculate quote and return on_select

c. Retail Network init / on_init, resulting in following steps:

1. Buyer App: add billing & shipping details for Seller App(s) and call init
2. Seller App(s): recalculate quote, add payment terms and return on_init

d. Retail Network confirm / on_confirm, resulting in following steps:

1. Buyer App: add promise / proof of payment to Seller App(s) and call confirm
2. Seller App(s): add order ID, update fulfilment state and return on_confirm

4. Detailing - Scenario 2

4.1 User Journey

Saurabh decides to order dinner for his family. Saurabh opens an app called “Open Commerce for All”.

a. Saurabh enters his location as “Malviya Nagar, Delhi”. He types in the category of “Quesadillas” and waits.

b. Saurabh sees a list of outlets that sell Quesadillas. He sees that his favourite Quesadillas are available from a Chili's outlet near his location.

c. He selects the following by clicking “Add to Cart”:

- Chipotle Quesadillas - 1
- Parmesan Crusted Cottage Cheese Quesadillas - 2
- Fresh Mex Quesadillas (with cajun rubbed cottage cheese) - 1

d. Saurabh adds these items to the cart and is about to checkout when he realizes he should also order a Salad. He searches for the category “Salad”.

e. He sees that Chili's has several options for Salad.

f. He selects “Caribbean Salad” by clicking “Add to Cart”. He then verifies his order by clicking on “View Cart”.

g. Saurabh then proceeds with the following steps to confirm the order:

- Click on “Checkout”;
- Clicks on “Add Shipping Details” and provide all details including the landmark;
- Click on “Proceed to Pay”;
- Following payment options are available - “Cash on Delivery”. He selects this option and clicks “Confirm”.

h. After confirmation of the order, Saurabh is redirected to an order screen that says, “Order confirmed”. A little while later, Saurabh receives a notification that says, “Searching for delivery agents near Chili's”.

Chili's backend was pre-configured to automatically discover nearby logistics services, when an order gets confirmed based on the store location and the delivery location.

i. Upon confirmation, the store owner clicks the order.

j. The store owner sees that there are two logistics providers that deliver to the shipping address for that order.

k. The store owner quickly selects the cheapest option.

l. The store owner confirms the delivery.

m. The logistics provider assigns a delivery agent to that order and sends the delivery order details to the store backend which forwards it to the “Open Commerce” app. The buyer sees the delivery status “Delivery agent en-route to store” on his app.

The logistics provider continuously updates the delivery status to the store backend, that automatically forwards the status to the buyer.

4.2 Transaction Steps

a. Retail Network search / on_search, resulting in following steps (note - this will be repeated once per category) :

1. Buyer App : add search intent, location and call search to ONDC Gateway
2. ONDC Gateway : broadcast search
3. Seller App(s) : add list of providers with matching items and return on_search to ONDC Gateway
4. ONDC Gateway : forward on_search from Seller App(s)
5. Buyer App : view catalog of any provider from Seller App(s) (direct search to Seller App)
6. Seller App(s) : send catalog of selected provider to Buyer App with matching items from original search (direct on_search to Buyer App)
7. Buyer App : select category ID inside Seller App(s) provider catalog and call search (direct search to Seller app)
8. Seller App(s) : add items under the selected category ID and return on_search

b. Retail Network select / on_select, resulting in following steps:

1. Buyer App: add Seller App(s) item to cart and call select
2. Seller App(s): add item, calculate quote and return on_select
3. Buyer App: remove Seller App(s) item from cart and call select
4. Seller App(s): remove item from cart, calculate quote and return on_select

c. Retail Network init / on_init, resulting in following steps:

1. Buyer App: add billing & shipping details for Seller App(s) and call init
2. Seller App(s): recalculate quote, add payment terms and return on_init

d. Logistics Network search / on_search, resulting in following steps:

1. Buyer App : **init** of retail network triggers the Logistics Buyer App - add pickup location, drop location and package details and call search to ONDC Gateway with logistics context
2. ONDC Gateway : broadcast search from Buyer App
3. Seller App(s) : add catalog and return on_search to ONDC Gateway
4. ONDC Gateway : forward on_search from Seller App(s) to Buyer App

e. Retail Network confirm / on_confirm, resulting in following steps:

1. Buyer App: add promise / proof of payment to Seller App(s) and call confirm
2. Seller App(s): add order ID, update fulfilment state and return on_confirm

f. Logistics Network init / on_init, resulting in following steps:

1. Buyer App : **confirm** of retail network triggers Logistics Seller App - add billing details, shipping details and call init to Seller App
2. Seller App : recalculate quote and return on_init to Buyer App

g. Logistics Network confirm / on_confirm, resulting in following steps:

1. Buyer App : add proof of payment and call confirm to Seller App
2. Seller App : assign delivery agent, update fulfilment status and return on_confirm to Buyer App

h. Logistics Network status / on_status, resulting in following steps:

1. Buyer App : get latest status of fulfillment by calling status to Seller App
2. Seller App : send following status to Buyer App: “Agent has been Assigned” via on_status
3. Buyer App : forward “Agent has been Assigned” status to Retail Buyer App via on_status
4. Seller App : send following status to Buyer App: “Agent at store” via on_status
5. Buyer App : forward “Agent at store” status to Retail Buyer App via on_status
6. Seller App : send following status to Buyer App: “Agent has picked up items” via on_status
7. Buyer App : forward “Agent has picked up items” status to Retail Buyer App via on_status
8. Seller App : send following status to Buyer App: “Agent is en-route to drop” via on_status
9. Buyer App : forward “Agent is en-route to drop” status to Retail Buyer App via on_status
10. Seller App : send following status to Buyer App: “Agent is at drop location” via on_status
11. Buyer App : forward “Agent is at drop location” status to Retail Buyer App via on_status
12. Seller App : send following status to Buyer App: “Order delivered” via on_status
13. Buyer App : forward “Order delivered” status to Retail Buyer App via on_status

e. Retail Network track / on_track, resulting in following steps:

1. Buyer App : get tracking info from Seller app by calling track
2. Seller App : fetch tracking info from Logistics Seller App by calling track via the Logistics Buyer App
3. Seller App : forward tracking info, if received, via on_track from the Logistics Seller App to the Retail Buyer App

5. Detailing - Scenario 3

5.1 <u>User Journey</u>

Saurabh decides to order dinner for his family. Saurabh opens an app called “Open Commerce for All”.

a. Saurabh enters his location as “Malviya Nagar, Delhi”. He types in the category of “Quesadillas” and waits.

b. Saurabh sees a list of outlets that sell Quesadillas. He sees that his favourite Quesadillas are available from a Chili's outlet near his location.

c. He selects the following by clicking “Add to Cart”:

- Chipotle Quesadillas - 1
- Parmesan Crusted Cottage Cheese Quesadillas - 2
- Fresh Mex Quesadillas (with cajun rubbed cottage cheese) - 1

d. Saurabh adds these items to the cart and is about to checkout when he realizes he should also order a Salad. He searches for the category “Salad”.

e. He sees that Chili's has several options for Salad.

f. He selects “Caribbean Salad” by clicking “Add to Cart”. He then verifies his order by clicking on “View Cart”.

g. Saurabh suddenly remembers that he needs personal care items for the next day. He types in the category of “Personal Care” and waits.

h. He sees that a nearby supermarket has the items that he needs.

i. He selects the following by clicking “Add to Cart”:

- Neutriderm Moisturising Lotion (125 ml) - 1
- Fiama Peach and Avocado Gel Bar (Medium)- 2
- MamyPoko Pants (XL) - 4

j. Saurabh then verifies his order by clicking on “View Cart”.

k. Saurabh then proceeds with the following steps to confirm the order:

- Click on “Checkout”;
- Repeat the following steps for each outlet from which items are being purchased:
    - Clicks on “Add Shipping Details” and provide all details including the landmark;
    - Click on “Proceed to Pay”;
    - Following payment options are available - “Cash on Delivery”. He selects “Cash on Delivery” and clicks “Confirm”.

l. After confirmation of the order, Saurabh is redirected to an order screen that says, “Order confirmed”.

Both the stores (Chili's / SuperMarket) backend was pre-configured to automatically discover nearby logistics services, when an order gets confirmed based on the store location and the delivery location.

m. Upon confirmation, the store owner clicks the order.

n. The store owner sees that there are two logistics providers that deliver to the shipping address for that order.

o. Each store owner selects a different logistics provider.

p. The store owner confirms the delivery.

q. The logistics provider assigns a delivery agent to that order and sends the delivery order details to the store backend which forwards it to the “Open Commerce” app. The buyer sees the delivery status “Delivery agent en-route to store” on his app.

The logistics provider continuously updates the delivery status to the store backend, that automatically forwards the status to the buyer.

5.2 Transaction Steps

a. Retail Network search / on_search, resulting in following steps (note - this will be repeated once per category):

1. Buyer App : add search intent, location and call search to ONDC Gateway
2. ONDC Gateway : broadcast search
3. Seller App(s) : add list of providers with matching items and return on_search to ONDC Gateway
4. ONDC Gateway : forward on_search from Seller App(s)
5. Buyer App : view catalog of any provider from Seller App(s) (direct search to Seller App)
6. Seller App(s) : send catalog of selected provider to Buyer App with matching items from original search (direct on_search to Buyer App)
7. Buyer App : select category ID inside Seller App(s) provider catalog and call search (direct search to Seller app)
8. Seller App(s) : add items under the selected category ID and return on_search

b. Retail Network select / on_select, resulting in following steps:

1. Buyer App: add Seller App(s) item to cart and call select
2. Seller App(s): add item, calculate quote and return on_select
3. Buyer App: remove Seller App(s) item from cart and call select
4. Seller App(s): remove item from cart, calculate quote and return on_select

For every (Retail Seller App + Logistics Provider), the following steps (c to e) will be repeated:

c. Retail Network init / on_init, resulting in following steps:

1. Buyer App: add billing & shipping details for Seller App(s) and call init
2. Seller App(s): recalculate quote, add payment terms and return on_init

d. Logistics Network search / on_search, resulting in following steps:

1. Buyer App : **init** of retail network triggers the Logistics Buyer App - add pickup location, drop location and package details and call search to ONDC Gateway with logistics context
2. ONDC Gateway : broadcast search from Buyer App
3. Seller App(s) : add catalog and return on_search to ONDC Gateway
4. ONDC Gateway : forward on_search from Seller App(s) to Buyer App

e. Retail Network confirm / on_confirm, resulting in following steps:

3. Buyer App: add promise / proof of payment to Seller App(s) and call confirm
4. Seller App(s): add order ID, update fulfilment state and return on_confirm

f. Logistics Network init / on_init, resulting in following steps:

1. Buyer App : **confirm** of retail network triggers Logistics Seller App - add billing details, shipping details and call init to Seller App
2. Seller App : recalculate quote and return on_init to Buyer App

g. Logistics Network confirm / on_confirm, resulting in following steps:

1. Buyer App : add proof of payment and call confirm to Seller App
2. Seller App : assign agent, update fulfilment status and return on_confirm to Buyer App

h. Logistics Network status / on_status, resulting in following steps:

1. Buyer App : get latest status of fulfillment by calling status to Seller App
2. Seller App : send following status to Buyer App: “Agent has been Assigned” via on_status
3. Buyer App : forward “Agent has been Assigned” status to Retail Buyer App via on_status
4. Seller App : send following status to Buyer App: “Agent at store” via on_status
5. Buyer App : forward “Agent at store” status to Retail Buyer App via on_status
6. Seller App : send following status to Buyer App: “Agent has picked up items” via on_status
7. Buyer App : forward “Agent has picked up items” status to Retail Buyer App via on_status
8. Seller App : send following status to Buyer App: “Agent is en-route to drop” via on_status
9. Buyer App : forward “Agent is en-route to drop” status to Retail Buyer App via on_status
10. Seller App : send following status to Buyer App: “Agent is at drop location” via on_status
11. Buyer App : forward “Agent is at drop location” status to Retail Buyer App via on_status
12. Seller App : send following status to Buyer App: “Order delivered” via on_status
13. Buyer App : forward “Order delivered” status to Retail Buyer App via on_status

e. Retail Network track / on_track, resulting in following steps:

1. Buyer App : get tracking info from Seller app by calling track
2. Seller App : fetch tracking info from Logistics Seller App by calling track via the Logistics Buyer App
3. Seller App : forward tracking info, if received, via on_track from the Logistics Seller App to the Retail Buyer App

6. Evaluation

The evaluation and awarding process is defined below.

Evaluation Process:

1. To qualify for an award, a participant needs to complete **all steps for a use-case scenario**.
2. A participant submits a claim along with a video recording that all steps for a use-case scenario are complete.
3. The submission will be done in a participant-specific sheet here.
4. The inspection of the claim will be based on the video submission and any subsequent inquiries by the Hackathon judging team and will accordingly accept/reject the submission.
5. Any submissions based on tool simulated API calls (e.g Postman simulated API calls) instead of actual implementation of APIs will be rejected.
6. Any reference app will not be evaluated for an award.

Awarding Process:

1. There will be 1 award for each use-case scenario.
2. **A submission, for a use-case scenario, accepted by the Hackathon judging team, and that completes the scenario first will be given the award.**
3. Total awards will be as follows:

Scenario 1 - ₹1 lakhs (to be split equally between buyer app, seller app);

Scenario 2 - ₹3 lakhs (to be split equally between buyer app, seller app, logistics app);

Scenario 3 - ₹6 lakhs (to be split equally between buyer app, all seller apps, all logistics apps);