API Based
Self Service for
Network Participant

V 2.0

protean
Change *is* growth

Agenda

- Scenarios considered
- Stages in Subscription
- Support
- Implementation Approach and Important Dates
- FAQs

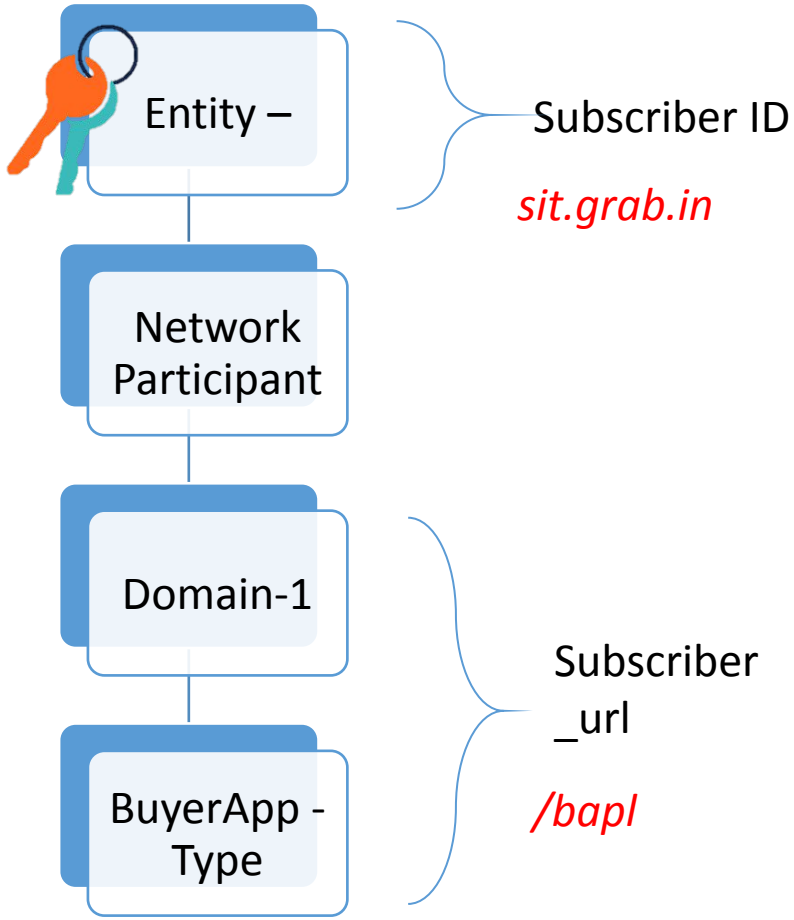
Scenarios considered

Scenarios

		Combination			Message Objects			
Operation #	Scenario List	buyer	Seller	msn	entity {Key}	network participant		Minimum Validation
						Other Fields	seller on record {Unique Key ID : Keys }	
	Entity Registration Scenarios							Schema, OCSP, Domain ownership , keys
1	Buyer New entity registration	Yes	No	No	yes	yes	no	Schema, OCSP, Domain, keys
2	Non-MSN Seller New entity registration	No	Yes	No	yes	yes	no	Schema, OCSP, Domain, keys
3	MSN Seller New entity registration	No	Yes	Yes	yes	yes	Yes	Schema, OCSP, Domain, keys
4	Buyer and Non-MSN Seller new registration	Yes	Yes	No	yes	yes	no	Schema, OCSP, Domain, keys
5	Buyer and MSN Seller new registration	Yes	Yes	Yes	yes	yes	yes	Schema, OCSP, Domain, keys
	Key Rotation Scenarios							Schema,AuthHeader, OCSP, Reg Domain vs Calling Domain, keys
6	Buyer App key rotation	yes	no	no	yes	no	no	Schema,AuthHeader, OCSP, Reg Domain vs Calling Domain, keys
7	Non-MSN Seller App Key rotation	no	Yes	no	yes	no	no	Schema,AuthHeader, OCSP, Reg Domain vs Calling Domain, keys
8	MSN Seller App Key rotation	no	Yes	Yes	yes	yes	yes	Schema,AuthHeader, OCSP, Reg Domain vs Calling Domain, keys
	Registration Amendment Scenario							Schema,AuthHeader, OCSP, Reg Domain vs Calling Domain, keys
9	Buyer adding New entity registration for Non-MSN Seller	Yes	Yes	No	yes	yes	no	Schema,AuthHeader, OCSP, Reg Domain vs Calling Domain, keys
10	Buyer adding New entity registration for MSN Seller	Yes	Yes	Yes	yes	yes	yes	Schema,AuthHeader, OCSP, Reg Domain vs Calling Domain, keys
11	MSN Seller adding seller on record	No	Yes	Yes	yes	yes	yes	Schema,AuthHeader, OCSP, Reg Domain vs Calling Domain, keys

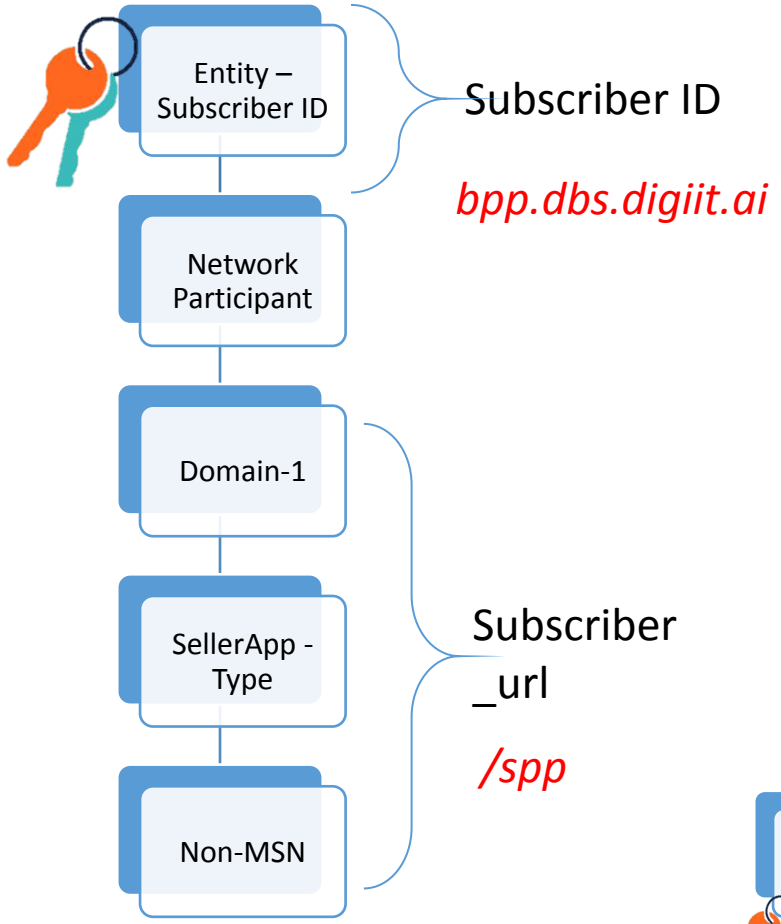
Minimum

BuyerApp



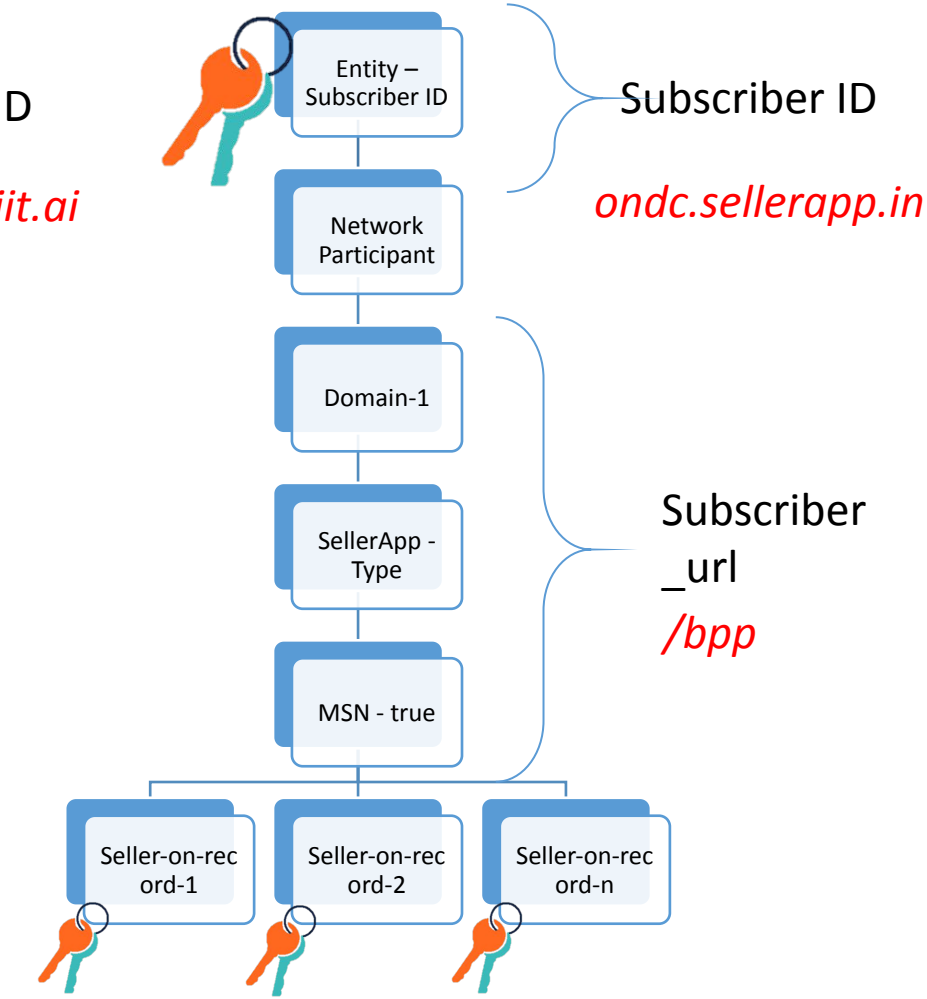
https://*sit.grab.in/bapl*

SellerApp (Non_MSN)



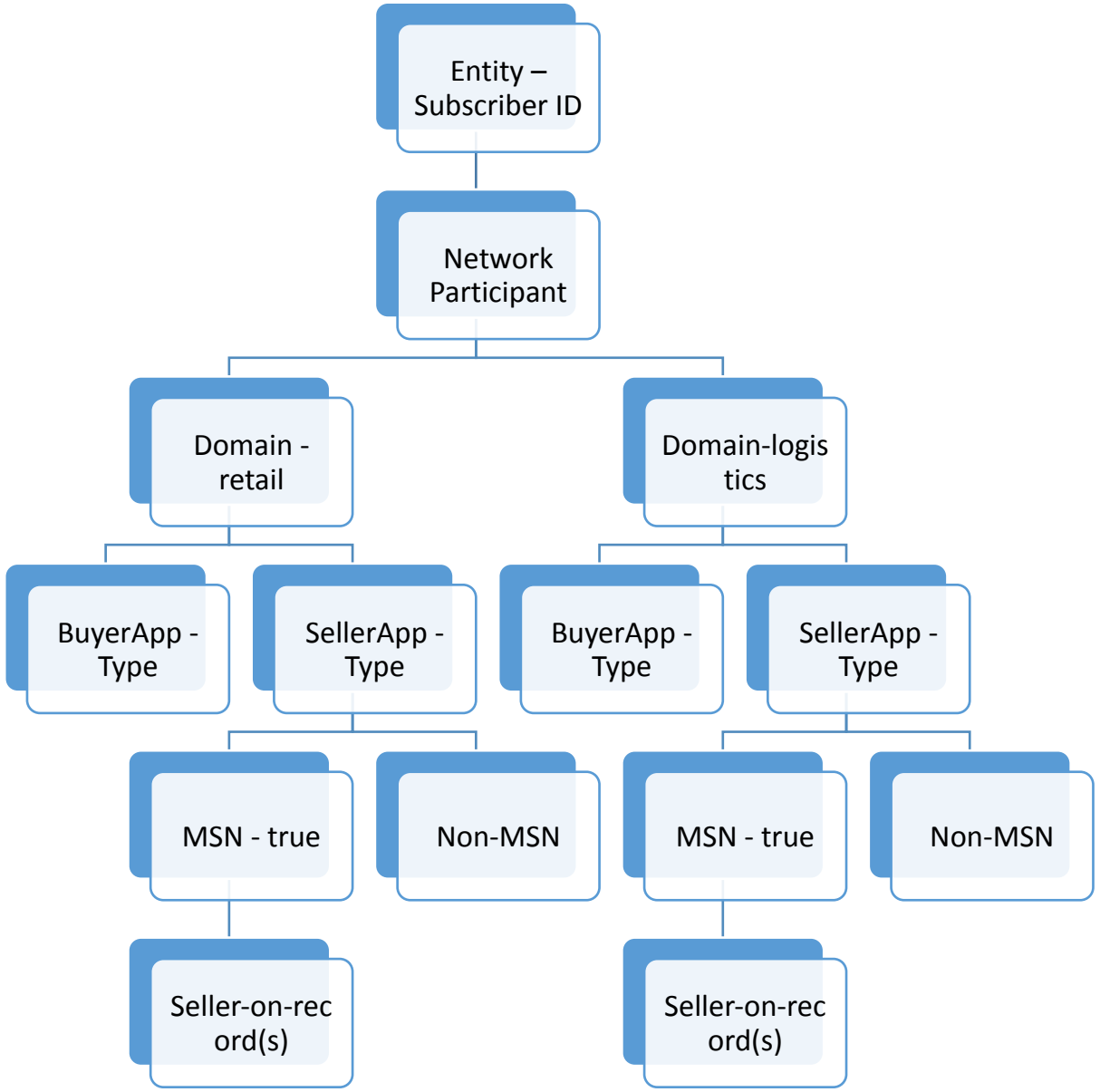
https://*bpp.dbs.digiit.ai/spp*

SellerApp (MSN)



https://*ondc.sellerapp.in/bpp*

Maximum



Assuming there are two domains active as on date in ONDC – retail and logistics

Validations/checks

1. At the time of entity registration
   - Domain Ownership
2. For all API Calls
   - Online Certificate Status Protocol
   - Key verification
   - Schema Validation
3. For all API calls after registration
   - Request initiated from Domain Vs Registered Domain Check

Stages in Subscription

Stage

A. Pre-requisite before calling /subscribe (Steps 1-5)
B. /subscribe
C. Registry calling /on_subscribe on Network participant hosted system
D. Active Network participant available in /lookup

Overall Steps involved : Stage A

1. subscriber_id: Buy or Register domain name.
2. SSL Certificate:
   a) Purchase SSL Certificate with subscriber_id (Domain Name)
   b) All communication with ONDC to happen from subscriber_id over SSL
   c) Online Certificate Status Protocol check will be done for each request on the basis of SSL configured on subscriber_id
3. /on_subscribe : Develop and host it on subscriber_id (Domain Name)
4. signing_public_key, signing_private_key and encryption_public_key, encryption_private_key : Generate key pairs
5. request_id for Domain Ownership Check :
   a) Create an request_id (ex. UUID)
   b) Sign request_id using signing_private_key generated in step 4
   c) Create “ondc-site-verification.html” and place it in root of subscriber_url by adding SIGNED_UNIQUE_REQ_ID
   d) Registry shall check existence of ondc-site-verification.html at https://subscriber_id/ondc-site-verification.html


ondc-site-verification.html

Overall Steps involved : Stage A (contd….)

5. Create “ondc-site-verification.html” and place it in root of subscriber_id

ondc-site-verification.html

Overall Steps involved : Stage B

Call /subscribe

```
{
  context*
                    {
                      operation
                    }
  message*          {
                      request_id          string
                                          A unique request id for this subscription.
                      timestamp*          string($date-time)
                                          Time of request generation in RFC3339 format
                      entity*             Entity > {...}
                      network_participant  > [...]
                    }
}
```

Overall Steps involved : Stage B (Contd…)

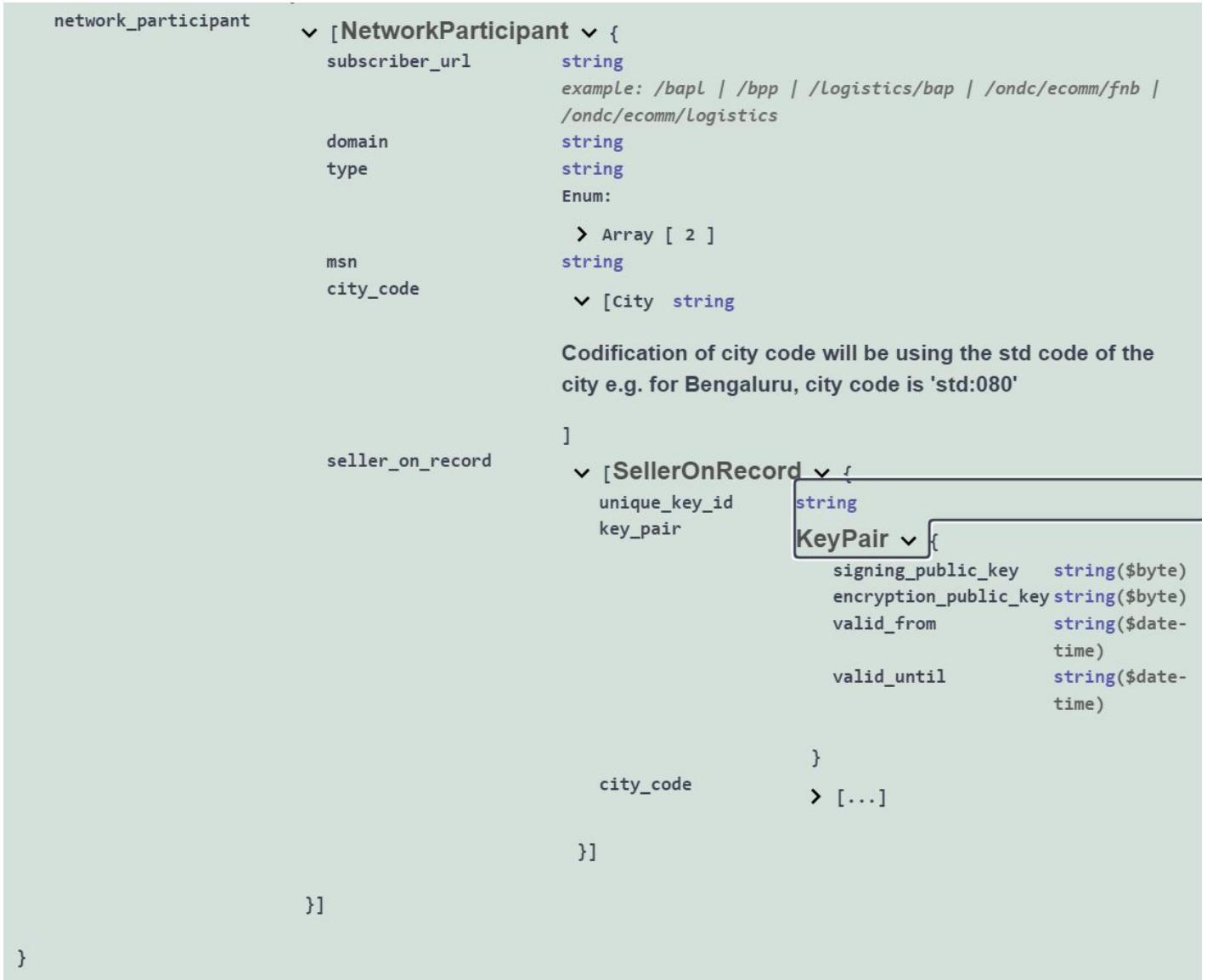
entity*

Entity {

gst	> {...}	
pan	> {...}	
name_of_authorised_signatory	string	
address_of_authorised Signatory	string	
email_id	string	
mobile_no	string	
country	string	*example: IND*
		Country code as per ISO 3166-1 and ISO 3166-2 format
subscriber_id	string	*example: sit.grab.in | bpp.dbs.digiit.ai | [illegible]*
callback_url	string	
key_pair	KeyPair {	
	signing_public_key	string($byte)
	encryption_public_key	string($byte)
	valid_from	string($date-time)
	valid_until	string($date-time)
	}	

}

Overall Steps involved : Stage B (Contd…)



Call /subscribe

Registry calling /on_subscribe on Network participant hosted system : Stage C

Registry will use encryption_public_key to encrypt challenge string

```
{
  "subscriber_id": "ondc.org",
  "challenge": "encrypted_challenge_string"
}
```

Request

Network participant need to use encryption_private_key to decrypt

```
{
  "answer": "decrypted_challange_string"
}
```

Response

Active Network participant available in /lookup : Stage D

Request

```
{
  "subscriber_id": "string",
  "type": "string",
  "city": "string",
  "domain": "string"
}
```

Response

```
{
  "subscriber_id": "string",
  "country": "string",
  "city_code": [
    "string"
  ],
  "domain": "string",
  "type": "string",
  "msn": "string",
  "signing_public_key": "string",
  "encr_public_key": "string",
  "valid_from": "2022-07-06T04:22:43.396Z",
  "valid_until": "2022-07-06T04:22:43.396Z",
  "created": "2022-07-06T04:22:43.396Z",
  "updated": "2022-07-06T04:22:43.396Z",
  "network_participant": [
    {
      "subscriber_url": "/bapl | /bpp | /logistics
      "domain": "string",
      "type": "buyer",
      "msn": "string",
      "city_code": [
        "string"
      ],
      "seller_on_record": [
        {
          "unique_key_id": "string",
          "key_pair": {
            "signing_public_key": "string",
            "encryption_public_key": "string",
            "valid_from": "2022-07-06T04:22:43.396Z",
            "valid_until": "2022-07-06T04:22:43.396Z"
          },
          "city_code": [
            "string"
          ]
        }
      ]
    }
  ]
}
```

- ✔ Authheader: Signed RequestBody
- ✔ OCSP Check
- ✔ Requested subscriber_id Vs Registered subscriber_id

Implementation Approach

Implementation Approach

- Call with Network Participant
  - 5th July
- Pre-Prod on-boarding
  - 8th Jul
- Pre-Prod Review
  - 11th Jul
- Production Migration Data to be presented for review to ONDC in case if on-boarding is stable by 11th July

Support

Support

- ONDC to create and share New JIRA Project for V2 Onbaording APIs
- Protean Team to provide monitor and provide resolution from 0900 HRS to 2400 HRS
- Daily dashboard publish of JIRA Issues by Protean Team to ONDC
  - Total Reported Issues
  - Severity wise
    - Open Issues
    - Closed Issues

FAQs

FAQs

1. As a network participant how do i create keys required for signing and encryption ?
   - Develop self
2. Is it ok to share signing private key and encryption private key ?
   - No, private keys needs to be kept confidential and secured by network participant. For example, make use of HSM
3. Why is ondc-site-verification.html required to be created and kept in root of subscriber_url ?
   - This is to verify ownership of domain that is used for calling APIs on ONDC.

Thank You