# Process to deploy application:

The proposed platform contains multiple components interacting with each other and executing the request. An application on the proposed platform requires should be designed based on the components. The platform demands the application to be developed in 3 parts where each part can be deployed at respective components.

**1. Client Application:**

Client application forms the face of the interactions with the platform. As the platform provides Proxy components to interact with other components , client application needs to just invoke the proxy api.

There are four apis being exposed by the proxy (request, query, register and getpublickey). The client should invoke these apis with necessary parameters explained as below.

1. **Request**

* Prepare **data** having fields:
* **Method** : Type of method i.e. request.
* **Client\_id** : Your app\_id received while registration from MS
* **Receiver** : List of receiver who will receive notification after execution of request.
* **Application\_id** : Your application id/name received after getting your smart-contract onboarded/deployed on CS.
* **Payload** : A json field having 2 parameters:
  1. **Message** : Field of type json containing parameters required to call Remote function.
  2. **Operation** : Function name you need to call in already deployed smart-contract with Application\_id.

**Example:**

**data** = *{"****method****":"request","****client\_id****":BankName,"****receiver****":[BankName],"****application\_id****":"ckyc","****payload****":{"message":message,"operation":"ckyc\_operation"}}*

* Call HTTP method of type : **POST** having ***url*** = ***PROXY\_URL*** , and message with json = **data**.

Example : *r = requests.post(url=url\_proxy,json=data)*

* In response for every method = “request”, you will receive proxy\_transaction id which can be further tracked to get current status of the request.

**`**

1. **Query**

* Prepare **data** having fields:
* **Method** : Type of method i.e. query.
* **Client\_id** : Your app\_id received while registration from MS
* **Application\_id** : Your application id/name received after getting your smart-contract onboarded/deployed on CS.
* **Payload** : A json field having 1 parameter:
  1. **Url:** Url of an api you want to hit for query, deployed at DS.

**Example:**

***data =****{"****method****":"query","****client\_id****":BankName,"****application\_id****":"ckyc","****payload****":payload}*

*Where, payload = {"url" : "query\_url”}*

* Call HTTP method of type : **POST** having ***url*** = ***PROXY\_URL*** , and message with json = **data**.

Example : *r = requests.post(url=url\_proxy,json=data)*

* In response will get desired output with status codes.

1. **NOTIFICATION**

* We have to start an infinite loop which always read notification from kafka servers, sent by PROXY for request with ***proxy\_txn\_id.***
* Consumer will read kafka topic messages with topic name = client\_id , where client\_id is received while registering with MS.

**2. Business functions at CS:**

The business functions needs to be executed on all replicas of the consensus service. This code will be executed only after obtaining consensus on the state of the application for requests with “method=REQUEST”. It is the application developer’s responsibility to develop a code which will be executing on multiple heterogeneous systems at same instance and this code will determine the efficiency of your app.

**Rules to be followed while writing business functions:**

* Make a function name **get\_data** to extract your payload sent by Client\_side.
* This payload have 2 extra symbol / parameter added automatically by proxy (component of platform).
  + Timestamp: timestamp when proxy received this request
  + P-txn-id : Proxy transaction Id which is used to track the status of requests.

**Example of function get\_data:**

def **get\_data**(self,payload):

x = payload

payload = x[‘payload’][‘message’]

timestamp = x[‘timestamp’]

p\_txn\_id = x[‘p-txn-id’]

receiver = x[‘receiver’]

return payload,timestamp,p\_txn\_id,receiver

* Any function should contain only single parameter payload and to extract that payload call *get\_data(payload)* function and use return values as needed.

**Example:**

*def ckyc\_operation(self,payload\_data):*

*payload,timestamp ,p\_txn\_id,receiver = self.getData(payload\_data)*

*….*

*…*

*response = “ Successfully Executed”*

*return response*

* The field **‘timestamp**’ provided by the proxy should be used to connected the reponses.
* ***p\_txn\_id*** should be the primary key while creating database tables/collections.

**3. Application Specific Query:**

* There are no specific rules for creating query functions for an application.
* Query function should be access only the state database of the data service.
* All urls for api should start with your **application\_id**.

**Example:**

If app\_id is “**ckyc**” then,

Url = “**ckyc**/.../…”