Rules to be followed for **Design and Development** of an app, to be deployed at PLATFORM:

App will be designed in 3 parts:

**Part 1: Client-side part of an app.**

**Part 2: Smart Contract deployed at CS(component of platform).**

**Part 3: Query to database through REST API maintained at Data Services.**

**Part 1: Client side**

3 situations when Client side will contact to platform.

1. REQUEST for transaction execution.
2. QUERY from State Database.
3. Notification from Proxy for executed transaction mapped with proxy\_txn\_id (will describe later)
4. **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.

**Part 2: Smart Contract deployed at CS(Consensus Services)**

* This part of code will be deployed at every node of CS.
* This code will be executed after submitting any method = “REQUEST”.
* One should make sure it is very optimized code as it’s going to run on every node for consensus. So, this part of code will determine the efficiency of your app.

**Rules to be followed while writing smart-contract 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

* We can write any desired function with 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

* Make sure you use only proxy **‘timestamp**’ if needed.
* ***p\_txn\_id*** as a primary key while creating database tables/collection record if needed.

**Part 3: Query to database through REST API maintained at Data Services.**

* There is no specific rules for creating this part of the app.
* You just need to query your records maintained in State database (If any)
* Only 1 thing you need to remember:
  + All urls for api should start with your **application\_id**.
  + Example: If app\_id is “**ckyc**” then,

Url = “**ckyc**/.../…”