**13/01/2017:**

ZAPI plug-in + JIRA – Approved.

Juho – Working on the JIRA and ZAPI and also on reporting side..

QA Change management port,

Testing – Test Execution- Reusable Assets –Template

Queries:

1. Jenkins integration with postman.
2. Nordea laptop allocation and access card.
3. Converting JSON report to html report (Template) Handle bars
4. Is anything I can learn more on this?
5. OSB layer – service layer to interact with database from Robot Framework.
6. Jenkins – Bamboo – Gradle – Robot Framework.
7. API testing using Robot Framework.

Installation:

1. Python (<https://github.com/robotframework/RIDE/wiki/Installation-Instructions> )
2. wxPython
3. robotframework (pip install robotframework)
4. ride (pip install robotframework-ride)
5. requestslibrary (pip install robotframework-extendedrequestslibrary)
6. httplibrary (pip install --upgrade robotframework-httplibrary

<https://nguyenthethang.com/2015/07/14/acceptance-test-robot-framework-test-returned-json-by-rest-api/> )

<http://stackoverflow.com/questions/12910709/how-to-use-rest-library-in-robot-framework>

cmd: python –version, pybot –version

Request keywords: <https://bulkan.github.io/robotframework-requests/>

**16/01/2017:**

* Juho mentioned there are some limitations if we use requestlibrary with Jython. So he is creating the api testing keywords with Java library and will publish in Maven repository soon.
* Once my desktop is allocated, I need to check with him regarding this.
* Create some keywords using Java keyword library to check below functionalities.
  + Response code check
  + Verify the body attribute value.
  + Verify the Header attribute value.
  + Verify the response time.

**Important urls:** httpbin.org

**17/01/2017:**

* **url:** <https://github.com/NitorCreations/RobotFramework-EclipseIDE/wiki/Installation> Robotide installation to eclipse.
* Create keywords and write a robot framework test script using the same.
* Use gradle to run the testng test scripts.
* Java collections tutorial : <http://beginnersbook.com/2013/12/hashmap-in-java-with-example/>
* Rest api testing: <http://www.methodsandtools.com/archive/restapitesting.php>
* [**https://www.joecolantonio.com/2016/03/08/getting-started-robot-framework-java/**](https://www.joecolantonio.com/2016/03/08/getting-started-robot-framework-java/)
* [**http://stackoverflow.com/questions/31538919/how-to-implement-a-java-library-in-robot-framework**](http://stackoverflow.com/questions/31538919/how-to-implement-a-java-library-in-robot-framework)

**18/01/2017:**

* <https://seleniumbycharan.wordpress.com/2015/07/26/api-testing-using-selenium-webdriver-and-java/>
* <http://helpqaspb.com/restapi.htm>
* <https://seleniumbycharan.wordpress.com/2016/07/13/simple-rest-api-testing-framework/>
* <https://github.com/robotframework/RIDE/issues/1442>
* JSON library jar: <http://stackoverflow.com/questions/8997598/importing-json-into-an-eclipse-project>

**PSD2:**

Payment Service Directive is set of rules and guidelines to follow for payment industry to regulate payment industry and to increase pan-European compitetion and participation in payment industry also from non banks.

-PSD adopted by European union in 2007.

-To drive further innovation and competition and transperancy in payment services, PSD2 has been adopted in 2015.

- PSD2 defines traditional financial institutions as “Account Servicing Payment Service Providers”(AS PSP)

- New players as “**Payment Initiation Service Providers**” (PISP) and “**Account Information Service Providers**” (AISP).

- PISP: **This is to target payment initiation services that establishes the software bridge between the Merchant website and online banking platform of the payer’s bank in order to initiate internet payments on the basis of credit transfer.**

-Rather than payer initiating the payment directly with their AS PSP, the payer intiates payment via PISP, which in turn passes instructions to AS PSP.

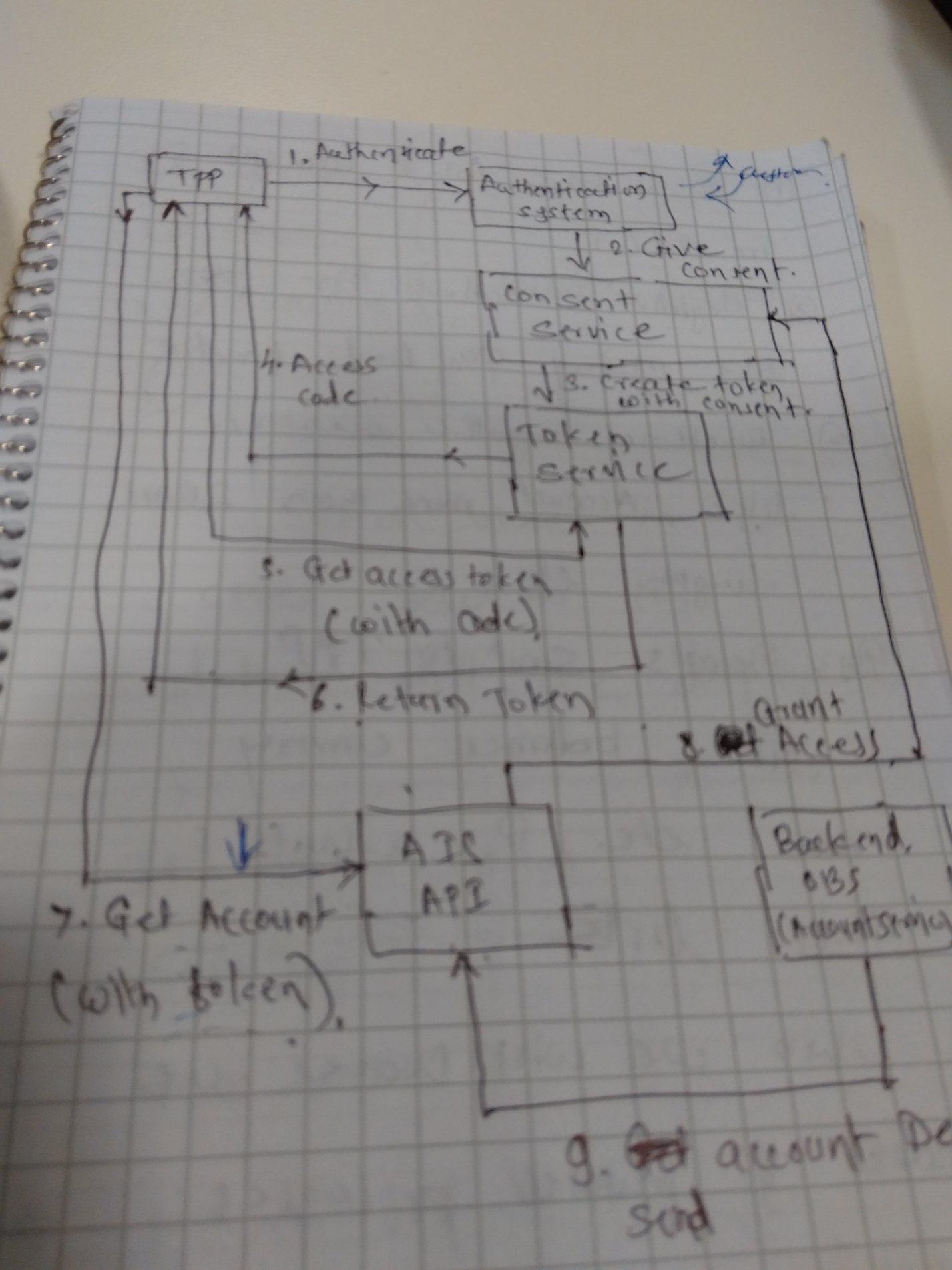
-PISP: PISP initiate only online banking e-Payment and doesn’t cover any kind of card payment.( Can be either Bank or TPP or Merchant)

-AISP: It gets the bank account details of a customer from different bank and show it to customer.( Can be either Bank or TPP)

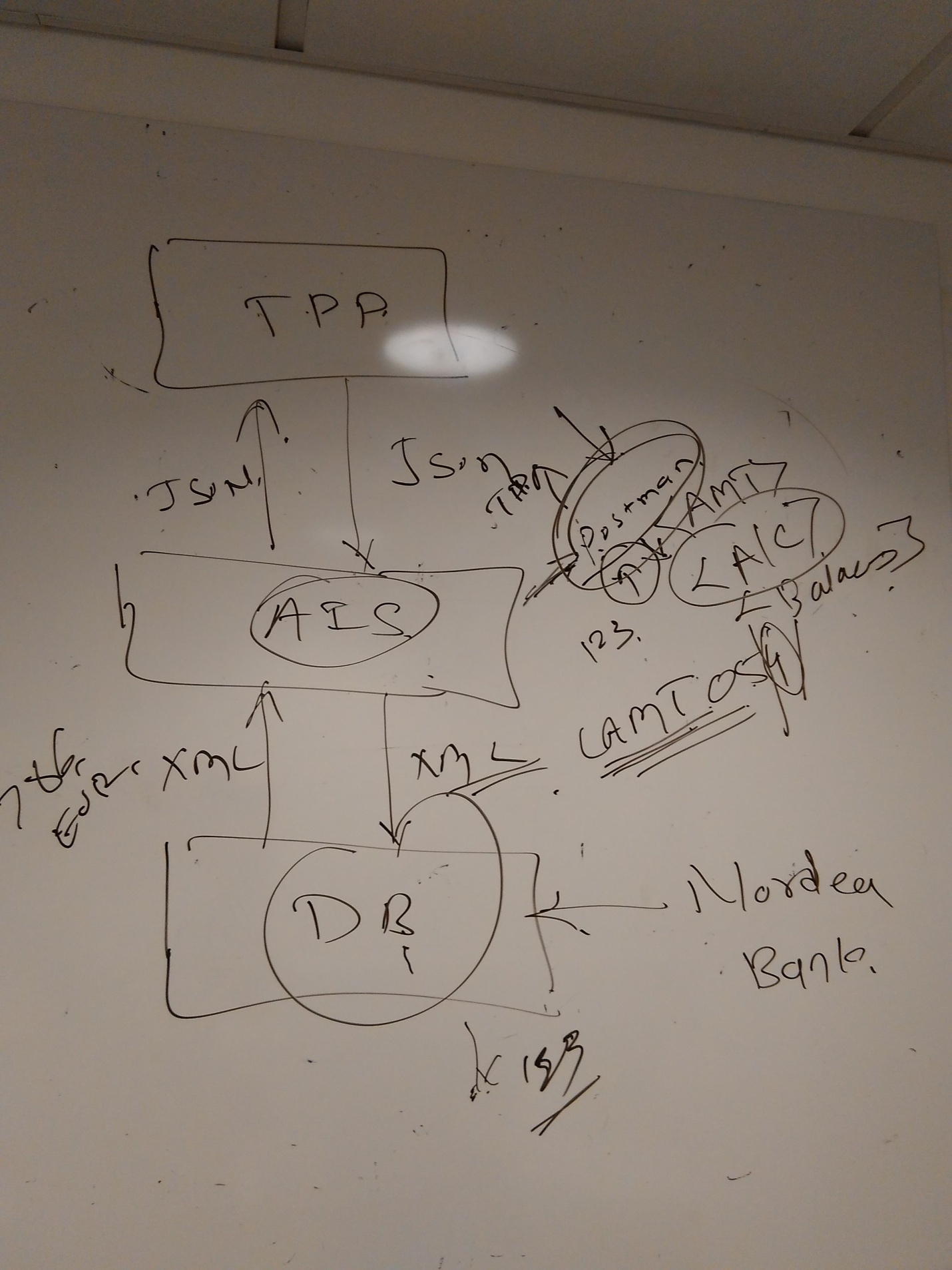
- AS PSP – It is an entity which maintains customer account. ( Mainly performed by Banks)

-TPP – It will be performing the role of PISP or AISP. ( Payment gateways, Wallets, Merchants)

- CSM (Clearing and Settlement mechanism): Mainly involved in clearing and settlement of funds for the transaction initiated using PISP.

Note: Currently we have test scenarios interacting with AISP with TPP and Backend OBS layer. PFB the diagram for the same. 

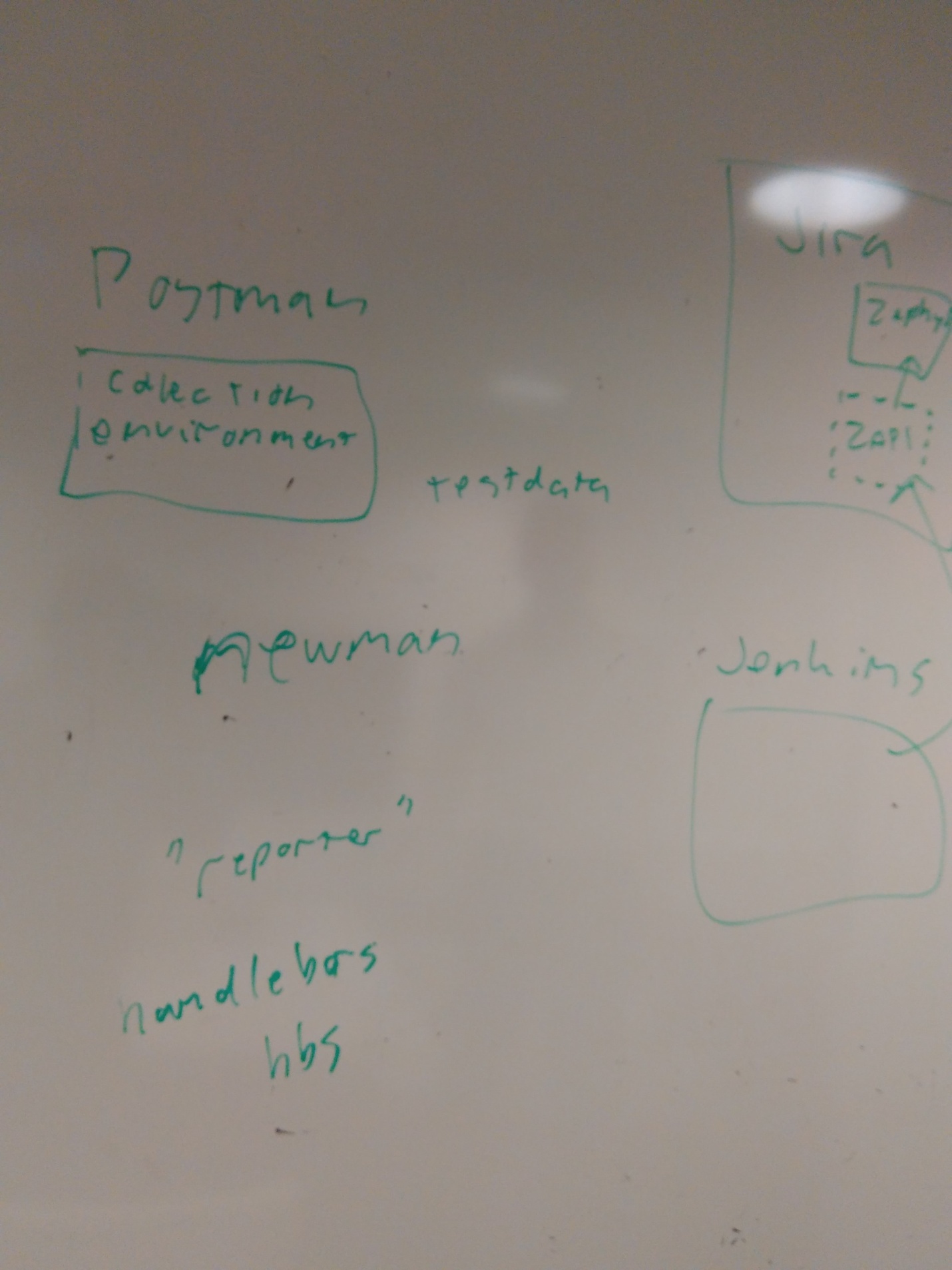
* getAccountInfoResponse
* showAccountInfo
* getAccountInfoRequest



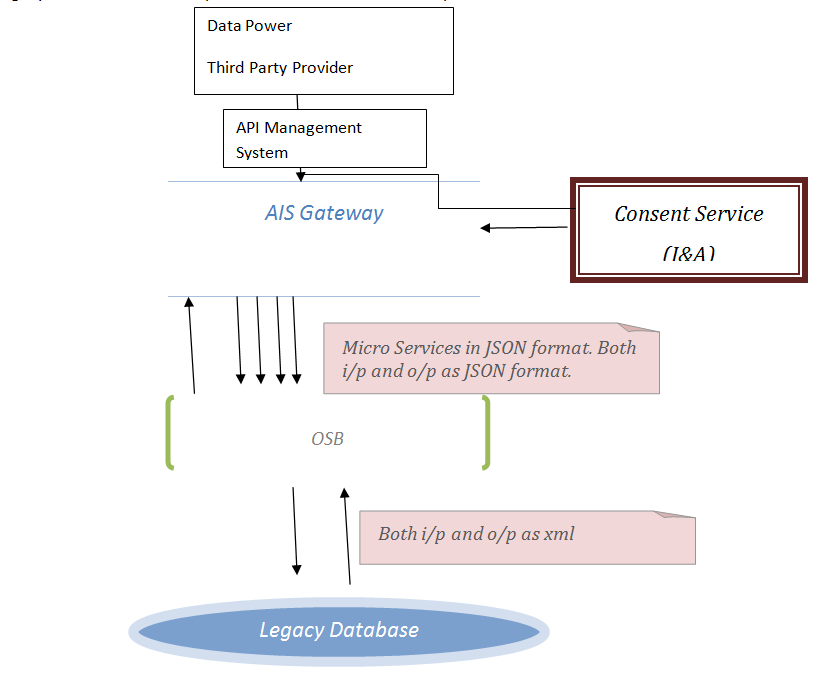
As per the current scenario Merchant website, we give the customer bank account details Visa will ask for further authentication like password or OTP, Once it authenticates Visa verifies whether it has enough credit balance and if there is enough balance it allows you to credit the payment to Merchant Bank.

Now using AISP, Merchant website gets all the available customer bank account information like balances and shows to customer.





**19/01/2017:**

* Comparing JSON and xml using java.
* Data Power takes the response from AIS Gate way and converts it to services/apis.
* We give the input url along with token to I&A (Identity and Authentication) and then it authenticates and provides consent to AIS Gate way.
* AIS gets the JSON information and sends the request for multiple micro services (Accounts, Transaction etc.) according to the the input JSON data.
* For each account it needs to send separate JSON.
* OSB converts xml response from Legacy database to JSON format and gives as input to AIS and viceversa.
* Legacy database takes the input as xml format and sends response in xml format. 

**23/01/2017:**

* <https://thecattlecrew.net/2014/07/09/osb-java-callout-with-xml-input-and-output-parameter/>
* <http://www.soawork.com/2013/03/osb-java-callout-simple-example.html>
* <http://www.oracle.com/technetwork/articles/marchioni-osb-093632.html>
* <http://www.albinsblog.com/2012/10/disablingenabling-oracle-service-bus.html#.WIXyt1N96po>
* <https://java.net/projects/oraclesoasuite11g/pages/OSB>

14/02/2017:

* <https://marketplace.atlassian.com/plugins/com.thed.zephyr.zapi/cloud/overview>