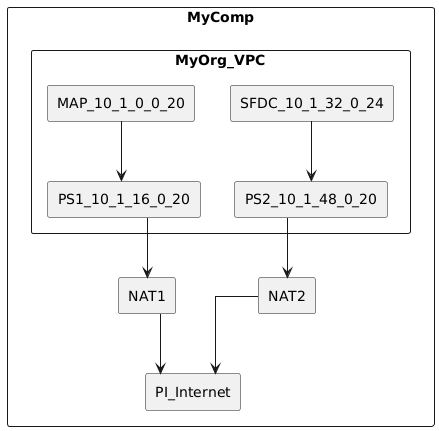
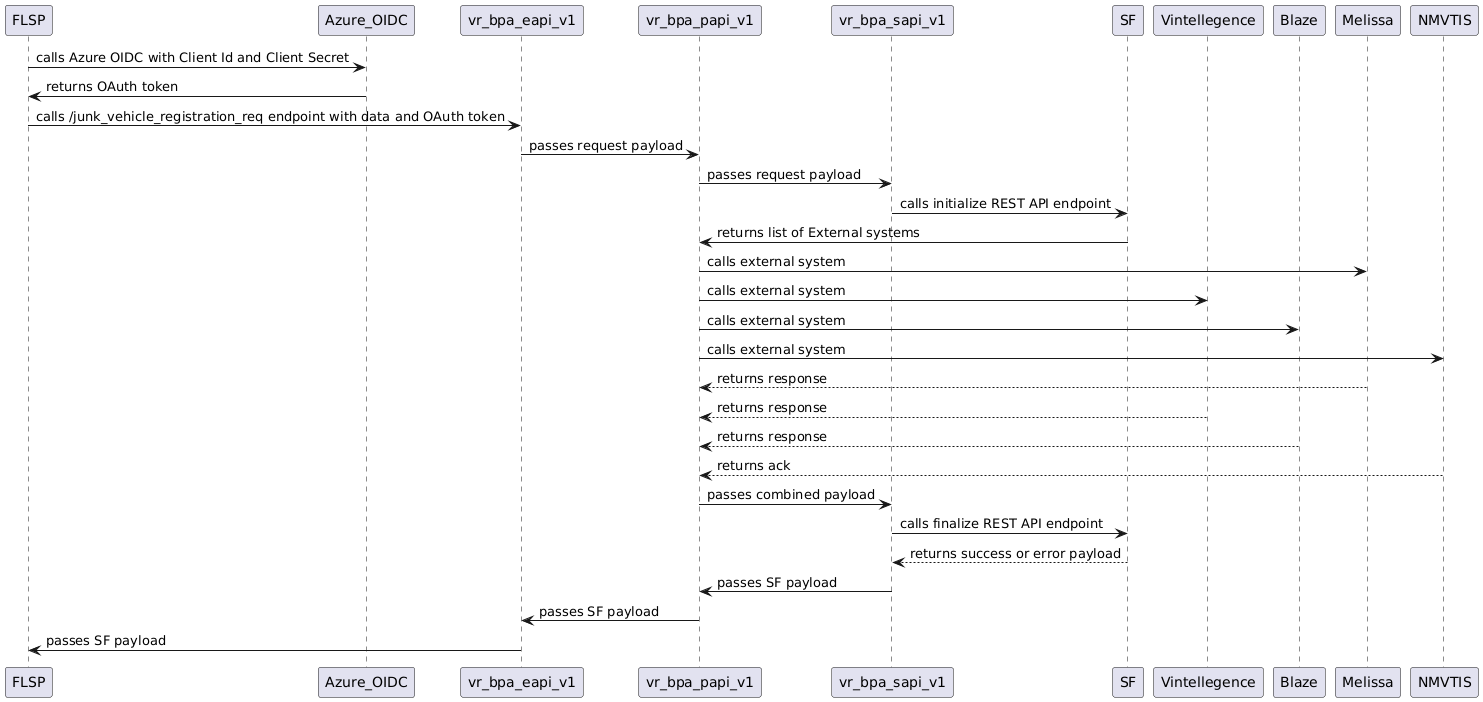
Mulesoft Technical Specification: BPA Vehicle Data Sync

# Processing Steps

1.FLSP (one of the BPA partners) calls Azure OIDC with Client Id and Client Secret  
  
2. FLSP gets an OAuth token back from Azure OIDC   
  
3. FLSP calls vr-bpa-eapi-v1’s /junk-vehicle-registration-req endpoint with vehicle’s junk-status inquiry data (JSON) (section link here) and passes the OAuth token for security pass   
  
4. Upon a successful request made by FLSP, vr-bpa-eapi-v1 passes the same req payload as it is to vr-bpa-papi-v1’s /junk-vehicle-registration-req endpoint. If the req is not successful (e.g. RAML validation failed) vr-bpa-eapi-v1 throws error to FLSP processed by the common-error-handling-framework   
  
5. vr-bpa-papi-v1 passes the same req payload vr-bpa-sapi-v1’s /junk-vehicle-registration-req endpoint   
  
6. vr-bpa-sapi-v1 (transforms the payload or passes it through?) calls SF initialize REST API endpoint   
  
7. SF makes necessary business level validations onto the payload and eventually returns a list of External systems to call for additional vehicle info if needed, along with the possible request payloads to make calls to those external systems. Currently these External systems are limited to Vintellegence, Blaze, Melissa and NMVTIS. Note: These possible request payloads are nothing but the req payloads to corresponding existing Mule SAPI/EAPIs to call those external systems   
  
8. The response is passed to the vr-bpa-papi-v1  
  
9. vr-bpa-papi-v1 then orchestrates parallel calls to those external systems via existing Mule APIs   
  
10. Out of the 4 possible external systems vr-bpa-papi-v1 gets responses from Melissa, Vintellegence and Blaze but gets only an ack from NMVTIS (since that is an asynchronous process). NOTE: a. NMVTIS rest is synced to SF asynchronously via vr-nmvtis-sol-aamva-sapi b. In case of partial success (e.g. only 2 external systesm responded with success but another responded with failure), Mule does not stop the process, rather retain the error and moves ahead to step 10   
  
11. vr-bpa-papi-v1 combines the response payloads (except NMVTIS) and calls vr-bpa-sapi-v1 (endpoint name here)   
  
12.vr-bpa-sapi-v1 calls SF Finalize REST API endpoint (endpoint here whenever available) and passes the combined payload   
  
13.vr-bpa-sapi-v1 retries x times to allow SF to get the NMVTIS info fed (unless SF responds with a 200 ok success payload at the first try)   
  
14. After the retries SF always responds with a 200 ok either with a success payload (print data info like barcode info etc.) or an error payload (business validation errors— ICD section link here)   
  
15. vr-bpa-sapi-v1 passes the same 200 ok SF payload back to vr-bpa-papi-v1   
  
16. vr-bpa-papi-v1 passes the same 200 ok SF payload back to vr-bpa-eapi-v1   
  
17. vr-bpa-eapi-v1 passes the same 200 ok SF payload back to FLSP 

# Mulesfot API Details

| API Name | Type | Github Link |  
  
|--------------------|-----------|------------------------------------|  
  
| Azure OIDC | SYSTEM | http://github.com/my-comp/azure-oidc |  
  
| vr-bpa-eapi-v1 | EXPERIENCE| http://github.com/my-comp/vr-bpa-eapi-v1 |  
  
| vr-bpa-papi-v1 | PROCESS | http://github.com/my-comp/vr-bpa-papi-v1 |  
  
| vr-bpa-sapi-v1 | SYSTEM | http://github.com/my-comp/vr-bpa-sapi-v1 |  
  
| SF initialize REST | SYSTEM | http://github.com/my-comp/sf-initialize-rest |  
  
| NMVTIS rest | SYSTEM | http://github.com/my-comp/nmvtis-rest |  
  
| SF Finalize REST | SYSTEM | http://github.com/my-comp/sf-finalize-rest |

# API Protocols

| API Name | Transmission Protocol | Description |  
  
|--------------------|-----------------------|------------------------------------------------------------------------------------------------------------------|  
  
| vr-bpa-eapi-v1 | OAuth | OAuth is an open standard for access delegation, commonly used as a way for Internet users to grant websites or applications access to their information on other websites without giving them the passwords.|  
  
| vr-bpa-papi-v1 | Client Authentication | Client authentication is a method of confirming a user's identity through the use of a digital certificate, username and password, or other authentication methods.|  
  
| vr-bpa-sapi-v1 | Client Authentication | Client authentication is a method of confirming a user's identity through the use of a digital certificate, username and password, or other authentication methods. |

# API Networks and Security

MyComp and PI are 2 networks here.   
  
My company named 'MyComp' is an AWS private cloud. My Organization, named 'MyOrg' is VPC which is a part of the MyComp and ranging 10.1.0.0/16.   
  
MyOrg has 4 subnets, 2 private subnets and 2 public subnets . Mulesoft Anypoint Platform or 'MAP' is one private subnet ranging 10.1.0.0/20.   
  
MAP is connected to a Public subnet names PS1 subnetted 10.1.16.0/20. Salesforce Data Cloud, names as 'SFDC' is the second private subnet deployed in MyOrg 10.1.32.0/24. SFDC is connected to the 2nd public subnet named PS2 subnetted 10.1.48.0/20. PS1 is connected to NAT1.   
  
PS2 is connected to NAT2. NAT1 and NAT2 are NAT gateways.   
  
Both NAT1 and NAT2 both are connected to the Public Internet called PI via an Internet gateway.   
  
PI resides totally outside MyComp and not a part of MyComp

# Data Mapping

Here is a sample mapping table between EAPI, PAPI, and SAPI for the given processing steps:  
  
  
  
| Processing Step | EAPI (Snake Case) | PAPI (Camel Case) | SAPI (Salesforce Naming) |  
  
|------------------------|--------------------------|--------------------------|---------------------------|  
  
| Account Number | account\_number | accountNumber | Account\_Number |  
  
| Allocated County | allocated\_county | allocatedCounty | Allocated\_County |  
  
| Cert Non Operation Date| cert\_non\_operation\_date | certNonOperationDate | Cert\_Non\_Operation\_Date |  
  
| Cert Non Operation Indc| cert\_non\_operation\_indc | certNonOperationIndc | Cert\_Non\_Operation\_Indc |  
  
| Certification Date | certification\_date | certificationDate | Certification\_Date |  
  
| Certification Indc | certification\_indc | certificationIndc | Certification\_Indc |  
  
| Clearing Indc | clearing\_indc | clearingIndc | Clearing\_Indc |  
  
| Cost Value | cost\_value | costValue | Cost\_Value |  
  
| Dealer Dismantler Number| dealer\_dismantler\_number| dealerDismantlerNumber | Dealer\_Dismantler\_Number |  
  
| Equip Num | equip\_num | equipNum | Equip\_Num |  
  
| Fee Acceptance Indc | fee\_acceptance\_indc | feeAcceptanceIndc | Fee\_Acceptance\_Indc |  
  
| File Code | file\_code | fileCode | File\_Code |  
  
| First Partner Id | first\_partner\_id | firstPartnerId | First\_Partner\_Id |  
  
| Fuel Type | fuel\_type | fuelType | Fuel\_Type |  
  
| Gross Combined Weight | gross\_combined\_weight | grossCombinedWeight | Gross\_Combined\_Weight |  
  
| Gross Vehicle Weight | gross\_vehicle\_weight | grossVehicleWeight | Gross\_Vehicle\_Weight |  
  
| Last Transfer Date | last\_transfer\_date | lastTransferDate | Last\_Transfer\_Date |  
  
| Length Inches | length\_inches | lengthInches | Length\_Inches |  
  
| Lienholder Name On Record| lienholder\_name\_on\_record| lienholderNameOnRecord | Lienholder\_Name\_On\_Record |  
  
| Make | make | make | Make |  
  
| Mussel Fee | mussel\_fee | musselFee | Mussel\_Fee |  
  
| Num Of Transfers | num\_of\_transfers | numOfTransfers | Num\_Of\_Transfers |  
  
| Owner Address | owner\_address | ownerAddress | Owner\_Address |  
  
| Owner Info | owner\_info | ownerInfo | Owner\_Info |  
  
| Owner Name On Record | owner\_name\_on\_record | ownerNameOnRecord | Owner\_Name\_On\_Record |  
  
| Ownership Cert Issue Date| ownership\_cert\_issue\_date| ownershipCertIssueDate | Ownership\_Cert\_Issue\_Date |  
  
| Plan Non Oper | plan\_non\_oper | planNonOper | Plan\_Non\_Oper |  
  
| Prior Plate With Owner Disp| prior\_plate\_with\_owner\_disp| priorPlateWithOwnerDisp| Prior\_Plate\_With\_Owner\_Disp|  
  
| Prior Use Tax | prior\_use\_tax | priorUseTax | Prior\_Use\_Tax |  
  
| RDF Code | rdf\_code | rdfCode | RDF\_Code |  
  
| RDF Indicator | rdf\_indicator | rdfIndicator | RDF\_Indicator |  
  
| Reg Plate Number | reg\_plate\_number | regPlateNumber | Reg\_Plate\_Number |  
  
| Repossession Date | repossession\_date | repossessionDate | Repossession\_Date |  
  
| Second Partner Id | second\_partner\_id | secondPartnerId | Second\_Partner\_Id |  
  
| TN Date | tn\_date | tnDate | TN\_Date |  
  
| Trans Code | trans\_code | transCode | Trans\_Code |  
  
| Type License Code | type\_license\_code | typeLicenseCode | Type\_License\_Code |  
  
| VIN Hin | vin\_hin | vinHin | VIN\_Hin |  
  
| VLF Wgt Exempt | vlf\_wgt\_exempt | vlfWgtExempt | VLF\_Wgt\_Exempt |  
  
  
  
Here is a sample DataWeave script to transform from EAPI to PAPI:  
  
```  
  
%dw 2.0  
  
output application/json  
  
---  
  
{  
  
 junkVehicleRequest: {  
  
 accountNumber: payload.account\_number,  
  
 allocatedCounty: payload.allocated\_county,  
  
 certNonOperationDate: payload.cert\_non\_operation\_date,  
  
 // Add more fields here based on the mapping  
  
 }  
  
}  
  
```  
  
  
  
And here is a sample DataWeave script to transform from PAPI to SAPI:  
  
```  
  
%dw 2.0  
  
output application/json  
  
---  
  
{  
  
 JunkVehicleRequest: {  
  
 Account\_Number: payload.accountNumber,  
  
 Allocated\_County: payload.allocatedCounty,  
  
 Cert\_Non\_Operation\_Date: payload.certNonOperationDate as String {format: "yyyyMMdd"},  
  
 // Add more fields here based on the mapping  
  
 }  
  
}  
  
```

# API Logging

| API Step | Logging Level |  
  
|----------|--------------|  
  
| FLSP calls Azure OIDC with Client Id and Client Secret | INFO |  
  
| FLSP gets an OAuth token back from Azure OIDC | DEBUG |  
  
| FLSP calls vr-bpa-eapi-v1’s /junk-vehicle-registration-req endpoint with vehicle’s junk-status inquiry data | INFO |  
  
| vr-bpa-eapi-v1 passes the same req payload as it is to vr-bpa-papi-v1’s /junk-vehicle-registration-req endpoint | INFO |  
  
| vr-bpa-eapi-v1 throws error to FLSP processed by the common-error-handling-framework if the req is not successful | ERROR |  
  
| vr-bpa-papi-v1 passes the same req payload vr-bpa-sapi-v1’s /junk-vehicle-registration-req endpoint | INFO |  
  
| vr-bpa-sapi-v1 calls SF initialize REST API endpoint | DEBUG |  
  
| SF makes necessary business level validations onto the payload and returns a list of External systems to call for additional vehicle info | INFO |  
  
| vr-bpa-papi-v1 orchestrates parallel calls to external systems via existing Mule APIs | INFO |  
  
| vr-bpa-papi-v1 combines the response payloads (except NMVTIS) and calls vr-bpa-sapi-v1 | INFO |  
  
| vr-bpa-sapi-v1 calls SF Finalize REST API endpoint and passes the combined payload | DEBUG |  
  
| vr-bpa-sapi-v1 retries x times to allow SF to get the NMVTIS info fed | WARN |  
  
| SF responds with a 200 ok either with a success payload or an error payload | INFO |  
  
  
  
Sample JSON Logger:  
  
```json  
  
{  
  
 "timestamp": "2022-01-01T12:00:00",  
  
 "level": "INFO",  
  
 "message": "FLSP calls vr-bpa-papi-v1’s /junk-vehicle-registration-req endpoint",  
  
 "api\_step": "4"  
  
}  
  
```

# API Error Handling

| API | Error Verbiage | Error Handling Mechanism |  
  
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|  
  
| Azure OIDC | Error retrieving OAuth token from Azure OIDC. Please check Client Id and Client Secret. | FLSP should handle the error and retry the request if necessary. |  
  
| vr-bpa-eapi-v1 | Error processing request. Please check the request payload and try again. | FLSP should handle the error and retry the request if necessary. |  
  
| vr-bpa-papi-v1 | Error processing request. Please check the request payload and try again. | FLSP should handle the error and retry the request if necessary. |  
  
| vr-bpa-sapi-v1 | Error calling SF initialize REST API endpoint. Please try again later. | FLSP should handle the error and retry the request if necessary. |  
  
| SF Finalize REST API | Error processing request. Please check the request payload and try again. | FLSP should handle the error and retry the request if necessary. |  
  
  
  
Common Notes: Mulesoft will use a common error handler which is documented at https://sharepoint.mycomp.com/integration-docs/c4eassets/error-handling. This error handler will capture and log all HTTP errors and provide a standardized way to handle them across all APIs in the integration flow.