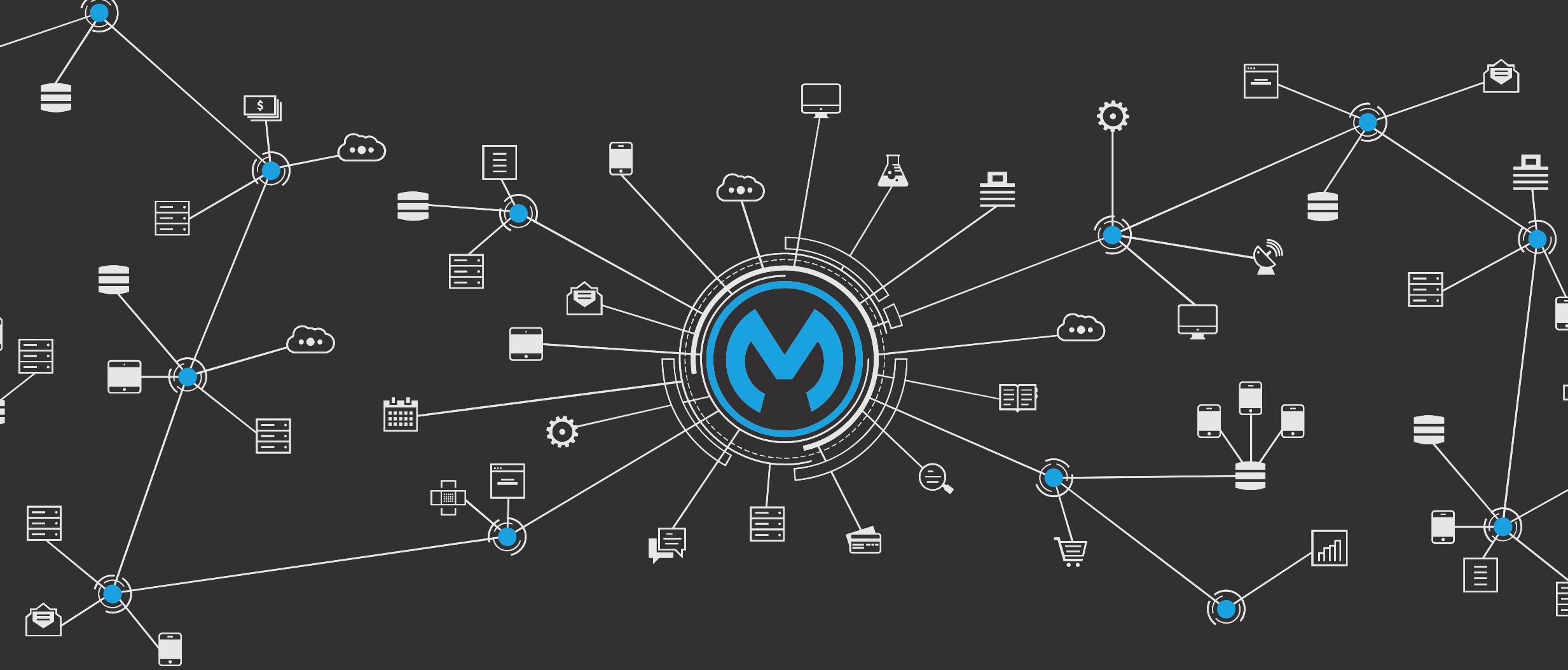
BP

Proof-of-Concept Anypoint APIs for KPI

Design Document





Contents

[**Introduction**](#_8ag47mwxgv8r) **4**

[Scope](#_4drm11fjv39w) 4

[Out Of Scope](#_wqagv3e6egn6) 4

[**Solution Context**](#_88a9kn65fkfy) **5**

[Objectives](#_fsmze13pnspt) 5

[**Technical Architecture**](#_7yfylm3qx25s) **6**

[Network](#_bmkbic1aezvc) 6

[Protocol](#_cxsvbtpk7w37) 6

[Transaction Behaviour](#_zg0es58d8nxe) 6

[**Cloudhub Deployment**](#_m2mf5iae59wo) **6**

[Deployment Target](#_hkh25tmqcjau) 6

[Business Group](#_u2eimgt56rm6) 6

[Runtime Version](#_y4nlsfwz9abr) 7

[Worker Sizing](#_vrd7fhy86jfn) 7

[Region](#_ucr2j9cjbwv6) 7

[Automatic Restart](#_61w7zthegkja) 7

[Persistent Queue](#_lspo21npx4q6) 7

[Logging](#_8ho43ziqh8hn) 7

[**Application Components Design**](#_qtbp54osq2kn) **7**

[Experience API](#_69z5fd3d9flt) 7

[Exposing the API](#_ktsdp65lejl3) 7

[Transport Layer Security](#_hitg6ydscph4) 7

[Process API](#_72uopk9xrcr5) 8

[Exposing the API](#_rjiu0tptbe1e) 8

[Transport Layer Security](#_6wd4xxn1h9sx) 8

[System API](#_s5n3nr7bz294) 8

[Exposing the API](#_q5dk0ogichwk) 8

[Transport Layer Security](#_8wiedzp2cns1) 8

[Cloudhub API](#_w2n917uxh14f) 8

[Cloudhub APIs](#_4ug3lez7obn) 8

[Pre Crowd 2 Migration](#_dy0azmn02ybq) 8

[After Crowd 2 Migration](#_vs9e6mta9e90) 8

[**Message integration**](#_t9rf0fwmnoy) **9**

[Read JMS XML, validate dates, generate GLOSS format (Brett)](#_9s1uud79b1hz) 9

[What tools were used to build the transformation? (Native or add-on)](#_m3nu99m44k5u) 9

[**POC Summary**](#_lmnkbm7ird6j) **12**

[**Open Queries**](#_mwo1wesxy2zd) **12**

# 

# Introduction

Purpose of this document is to provide design for proof of concept APIs for anypoint KPIs. The intention is to capture the design for KPIs to be made available for BP Enterprise BI tool.

## Scope

The scope of design is primarily targeted at creating a proof of concept.

The KPIs that are scoped for design are:

* Number of APIs / Integrations developed on Anypoint
* Number of APIs / Integrations being managed by Anypoint in production
* Number of APIs which are available to external consumers / partners

The KPIs are targeted to be made available as APIs for consumer system.

These APIs will allow for creating a integration between BP Enterprise BI tool and anypoint platform to automate ingestion of usage data.

## Out Of Scope

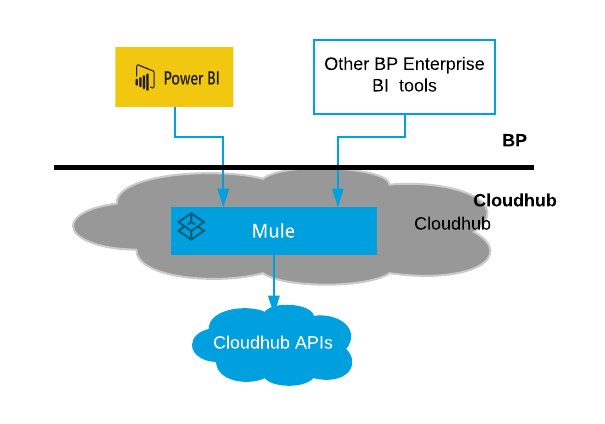
The following are considered out of scope for the Proof of concept.

* Datastore for storing data extracted from Cloudhub APIs

# Solution Context

## Objectives

The below figure captures the Solution context for the Anypoint KPIs to be available for BP Enterprise BI tools like Power BI and Other BI Enterprise BI tools.



Power BI and Other BP specific enterprise BI tools are hosted in BP managed sites.

Anypoint KPI APIs are proposed to be hosted in cloudhub and built using mule.

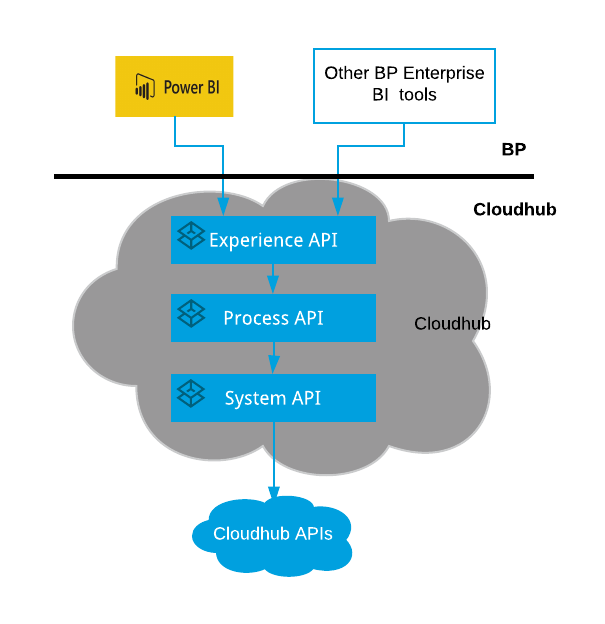
Cloudhub REST APIs will be used for extracting usage details from cloudhub.

# 

# 

# Technical Architecture

API led approach to be used for this proof of concept for building Mule.



Mule is layered into Experience Layer, Process Layer and System Layer.

## Network

VPC is expected to be setup between network where BP hosted service are located and cloudhub.

## Protocol

APIs will be https protocol based for proof of concept.

## Transaction Behaviour

The APIs will be synchronous in natures for this proof of concept. And will retrieve the usage details from Cloudhub. For every API call Cloudhub will be invoked to retrieve information.

# Cloudhub Deployment

## Deployment Target

The Mule API are targeted to be deployed to cloudhub.

## Business Group

The business group where the Mule APIs will be deployed is currently not know. This need to be identified closer to deployment planning.

## Runtime Version

Mule APIs runtime version planned for deployed is 3.9.0

## Worker Sizing

As the KPIs are proof of concept, 0.1 vCore is proposed to be allocated to each of the Mule API build. This would be 500 MB of memory.

The number of workers to be allocated is 1 worker. As this is proof of concept, there is no HA feature required. Application logging is expected to be low.

## Region

Default region associated with the organisation is expected to be used.

Region can be adjusted closer to where BP enterprise BI tools are located.

## Automatic Restart

Automatic restart application when not responding is to be enabled to allow application to restart automatically when not responding.

## Persistent Queue

Persistent queue is not to be used for this Proof of concept. As the message loss for the use cases are not expected to be recovered.

## Logging

Cloudhub logging is to be used for logging for this proof of concept.

# Application Components Design

## Experience API

### Exposing the API

The API will be available for access only within the BP VPC. And will not be accessible from public IP.

### Transport Layer Security

The service will be exposed over HTTPS. The port settings to be used for deployment is ${https.private.port}.

## Process API

### Exposing the API

The API will be available for access only within the BP VPC. And will not be accessible from public IP.

### Transport Layer Security

The service will be exposed over HTTPS. The port settings to be used for deployment is ${https.private.port}.

#### Process API

## System API

### Exposing the API

The API will be available for access only within the BP VPC. And will not be accessible from public IP.

### Transport Layer Security

The service will be exposed over HTTPS. The port settings to be used for deployment is ${https.private.port}.

### Cloudhub API

Pre Crowd release 2 APIs are planned to be used for the proof of concept. Currently there is no visible plan for migrating BP cloudhub to Crowd Release 2.

#### System API

Pre-Crowd 2 release Anypoint APIs that need to be wrapped.

**Login API <login>**

|  |  |
| --- | --- |
| URL | https://anypoint.mulesoft.com/accounts/login |
| Method | POST |
| Body | {"username":"{{ap\_username}}", "password": "{{ap\_password}}"} |
| Content-Type | application/json |

**User Details (get list of all orgs)**

|  |  |
| --- | --- |
| URL | https://anypoint.mulesoft.com/accounts/api/me |
| Method | GET |
| Header | Authorization: {{login.token\_type}} {{login.access\_token}} |

**For each Org <org>**

|  |  |
| --- | --- |
| URL | https://anypoint.mulesoft.com/apiplatform/repository/v2/organizations/{{org\_id}}/apis |
| method | GET |
| Header | Authorization: {{login.token\_type}} {{login.access\_token}} |

**For each API version <versionObj>**

|  |  |
| --- | --- |
| URL | https://anypoint.mulesoft.com/apiplatform/repository/v2/organizations/{{org\_id}}/apis/{{apiId}}/versions/{{versionId}} |
| method | GET |
| Header | Authorization: {{login.token\_type}} {{login.access\_token}} |

**API Version Contract <versionContracts>**

|  |  |
| --- | --- |
| URL | https://anypoint.mulesoft.com/apiplatform/repository/v2/organizations/{{org\_id}}/apis/{{apiId}}/versions/{{versionId}}/contracts |
| method | GET |
| Header | Authorization: {{login.token\_type}} {{login.access\_token}} |

**API Version Policies <policies>**

|  |  |
| --- | --- |
| URL | https://anypoint.mulesoft.com/apiplatform/repository/v2/organizations/{{org\_id}}/apis/{{apiId}}/versions/{{versionId}}/policies |
| method | GET |
| Header | Authorization: {{login.token\_type}} {{login.access\_token}} |

## Cloudhub APIs

### Pre Crowd 2 Migration

Currently BP cloudhub platform is on pre-crowd 2 release. The design has been built assuming the Pre Crowd 2 release cloudhub APIs.

### After Crowd 2 Migration

The Cloudhub APIs have changed from Pre-Crowd 2 release to Crowd 2 release.

When BP migrates to Crowd 2 release the existing Cloudhub APIs would not provide accurate data.

As currently BP cloudhub platform has not yet migrated to Crowd 2 Release, the pre-crowd 2 release cloudhub APIs are being used for this design. After BP migrates to crowd 2 release, the crowd 2 release cloudhub APIs need to be used.

# Message integration

## Read JMS XML, validate dates, generate GLOSS format (Brett)

Build a flow to read JMS XML (Instrument) message and validate dates and generate the Gloss internal instrument format and write it to another MQ. During transformation, invoke a JAVA api to perform the transformation.



Note: Can we re-use the XLS or XML file which today contains the mapping between source and target format.

**What Broadridge is looking:**

* *What tools were used to build the transformation? (Native or add-on)*
* *What the mapping capabilities are? (XML, Raw, Other)*
* *How is the service tested and validated?*
* *How much custom code (if any) was needed to complete the task?*
* *How are errors and exceptions handled in this module?*
* *How components go into assembling the service? (How many moving parts)*
* *What visibility is there into the process beyond the tester?*
* *What would need to be changed based upon deploying to production?*
* *CI & CD*
* *Retry capability ( Message vs system failures )*
* *Is there any documentation that can be generated on the service?*
* *What metrics is produced from the transformation (Num Transformations/sec)*
* *Message level, Flow level monitoring*

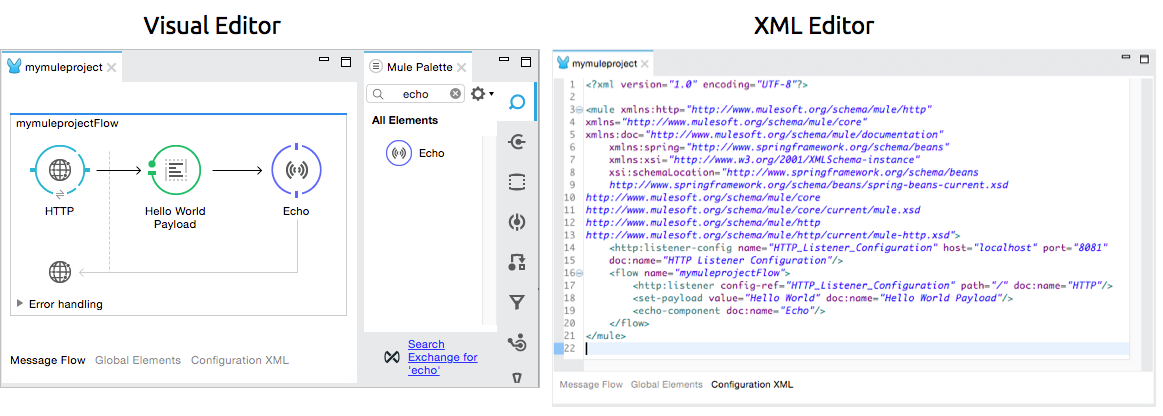
### What tools were used to build the transformation? (Native or add-on)

[Anypoint Studio](https://docs.mulesoft.com/anypoint-studio/v/6/) is MuleSoft’s Eclipse-based integration development environment for designing and testing Mule applications. You can deploy the application and run it on your Mule server.

The same editor also allows you to edit API definition files (in [RAML](https://raml.org/) and WSDL), create domains that define [shared resources](https://docs.mulesoft.com/mule-user-guide/v/3.8/shared-resources).

Anypoint Studio offers two parallel tabs that you can utilize to design and craft your applications:

* **Visual Editor**
* **XML Editor**

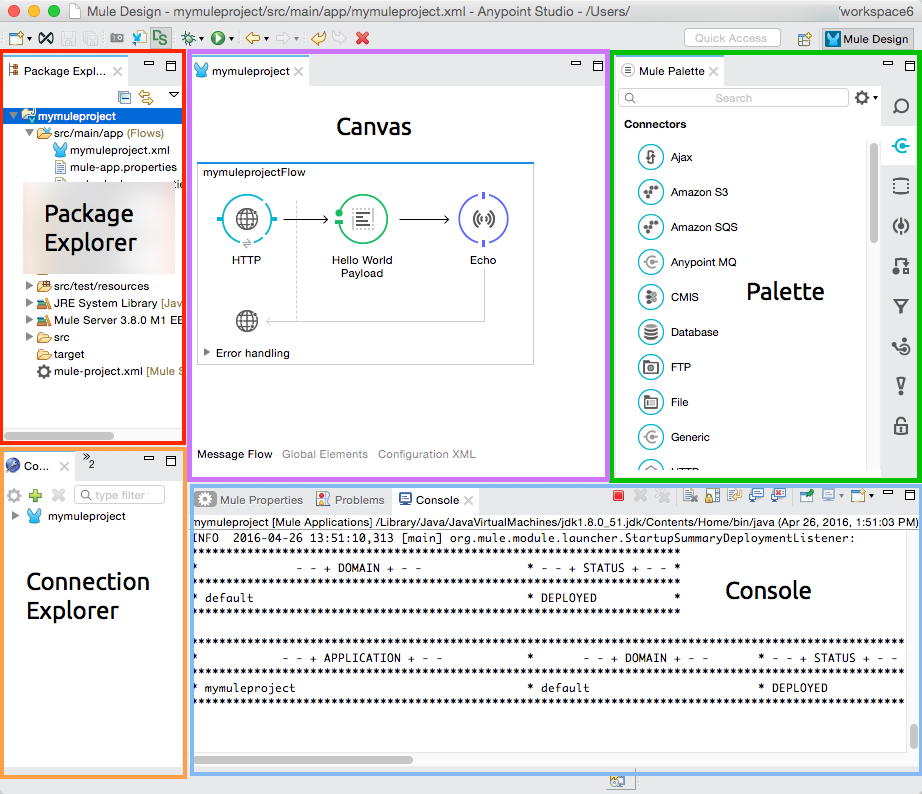
****

What you do in one editor reflects in the other. For example, if you drop a connector onto the Visual Editor canvas, the addition of the connector element to your application is recorded in the XML Editor. This real-time reflection of changes is referred to as **two-way editing**. *Same application, two different editors.*

The Visual Editor

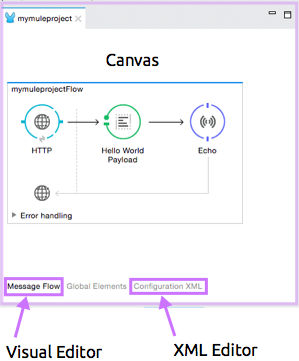
Studio’s visual editor is divided into several parts:

1. **Package Explorer**
2. **Canvas**
3. **Palette**
4. **Connection Explorer**
5. **Console**
6. **[Properties Editor](https://docs.mulesoft.com/anypoint-studio/v/6/#properties-editor)**
7. **DataSense Explorer**

****

The **Package Explorer** displays project folders and files in a tree format. Click to expand or contract Mule project folders and examine the files that make up a Mule project.

The **Canvas** provides a space for arranging pre-packaged building blocks into Mule applications. The **Palette**, on the right edge of the canvas, displays a list of building blocks that you can drag and drop onto the canvas to build your application. At the base of the canvas, Studio offers three tabs which offer different ways to interface with your application:



* **Message Flow**: Drag-and-drop interface in which you arrange building blocks to form an application
* **Global Elements**: Interface for creating or editing reusable configurations for Mule building blocks
* **Configuration XML**: XML editor interface displays the XML elements that correspond to the building blocks arranged on the Message flow canvas

The **Connection Explorer** displays a list of any global connector configurations defined in your application for quick access to your configuration details. Behind this tab, a second tab contains an **outline**, which displays a miniaturized overview of the whole canvas, along with a box cursor that scrolls with the display in the canvas pane. This provides a high-level, bird’s-eye view of your complete Mule application as it extends beyond the borders of the canvas.

# POC Summary

See the [PoC Summary document](https://docs.google.com/document/d/1G1A4uJN8xaEXPH2sIxyDj7Yt0yFMYgFECDMcSV4g0Fs/edit#) for final thoughts on the PoC.

# Open Queries

1. Which business group in BP cloudhub account will the Mule deployment be performed?
2. Does the experience API need to be exposed from Public IP or does it need to be exposed internally within the VPC only?
   1. Not sure where the PowerBI is currently cloud or on-premise?
   2. Not sure if Power BI will connect direct to cloudhub or it will connect via VPC?
3. When is BP cloudhub planned to be migrated to Crowd 2 Release?