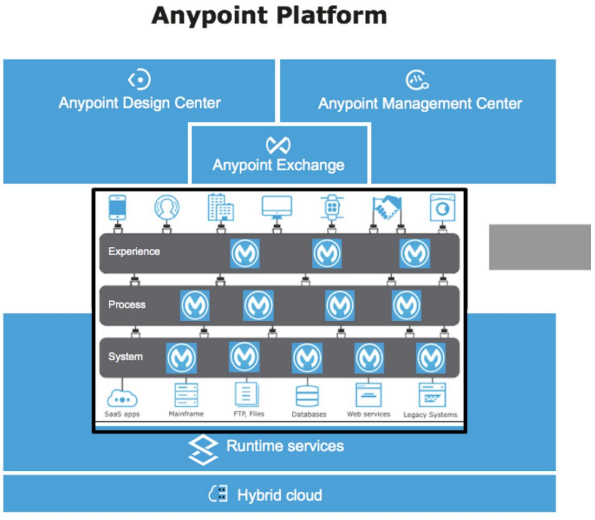
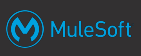
Module 2: Identifying Anypoint Platform Components and Capabilities

Goal



At the end of this module, you should be able to



- Identify overall design intentions of Anypoint Platform
- Review Anypoint Platform capabilities and high-level components
- Distinguish between Anypoint Platform service and deployment models
- Align Anypoint Platform components and capabilities with an integration use case

Putting Anypoint Platform and Mule applications into an integration architecture

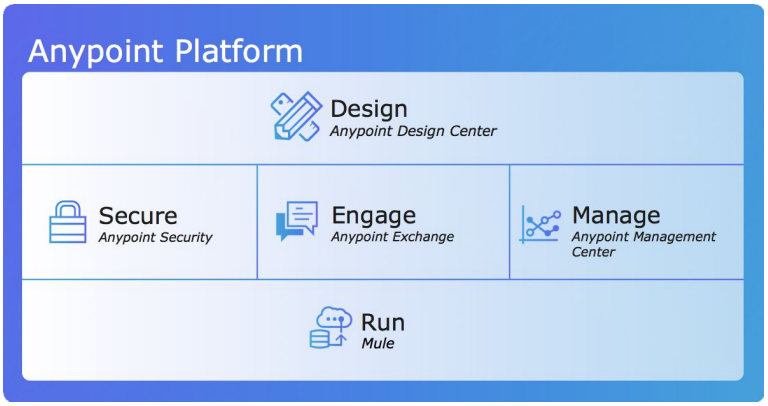
Leveraging MuleSoft to realize integration solutions

- Before creating an integration architecture for the course case study, you must understand the **platform** and **tools** provided by **MuleSoft**
- This is the goal of the next two modules, before starting to fill in the architecture documents in the rest of the course
  - Module 2: Identifying the Components and Capabilities of Anypoint Platform
  - Module 3: Designing Integration Solutions with Mule Applications
    - Identify the components and capabilities of the Mule runtime and associated development toolsets

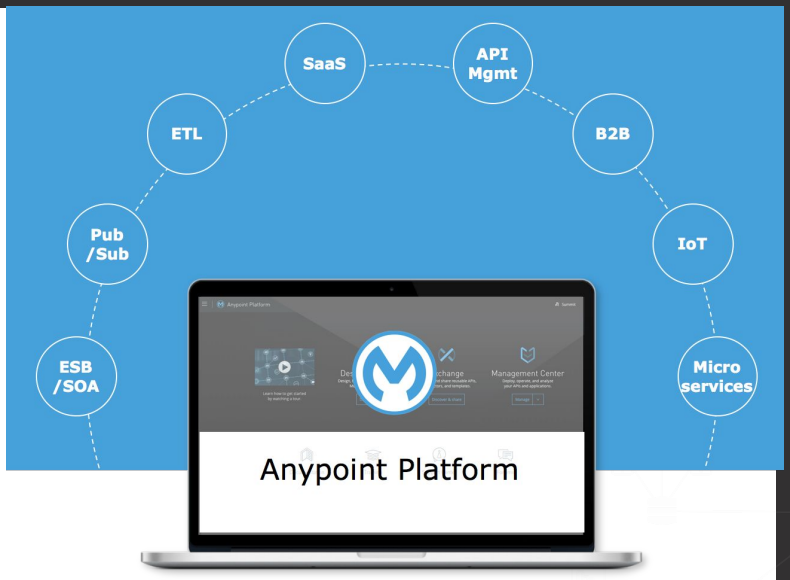
Introducing Anypoint Platform

What is Anypoint Platform?

- A **unified**, highly productive, **hybrid** integration platform that creates a seamless distributed system of apps, data, and devices
- Can also manage full API lifecycles to promote API-led development



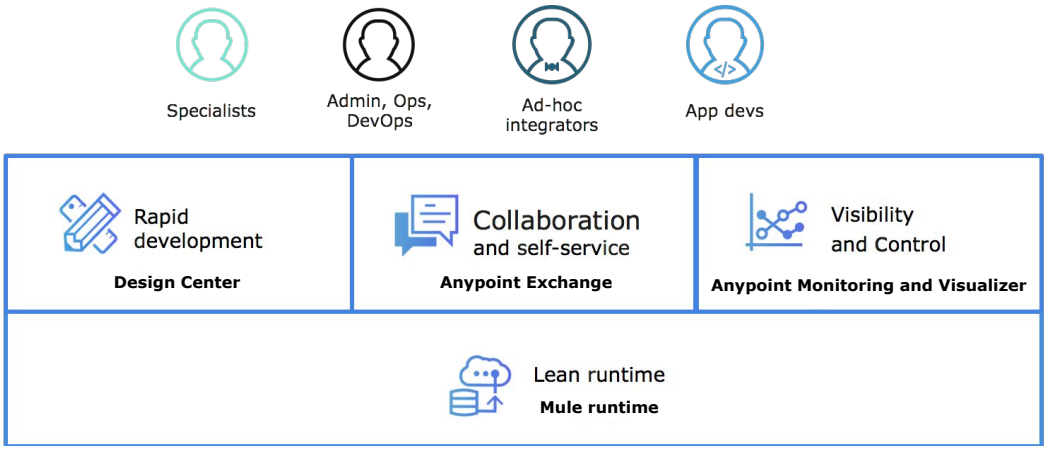
Anypoint Platform manages Mule application lifecycles



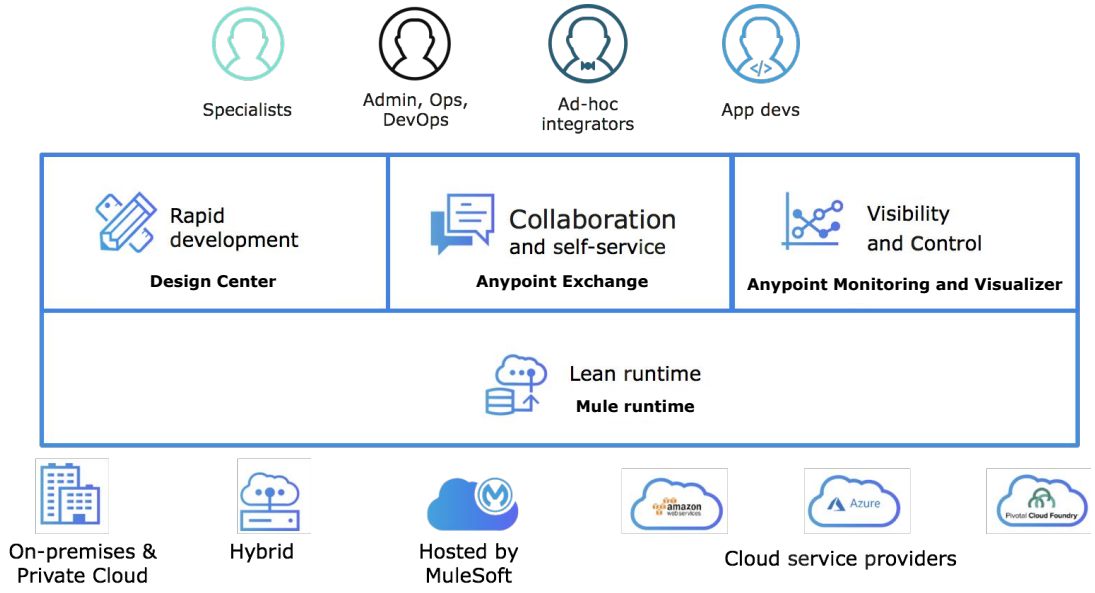
Advanced enterprise platform for designing, developing, and managing APIs and integrations

- Uniquely built as a single product
- Deploy anywhere
- Flexible and wide range of use cases

One unified platform to design and manage integration solutions, and exchange related assets



One common runtime for all types of deployments



Design once, deploy anywhere

Anypoint Design Center

Anypoint Management Center

On-prem

Private cloud

fully managed iPaaS

Databases | FTP, Files | Web services | SaaS Apps | On-prem Apps | Social Apps | IoT | Partners

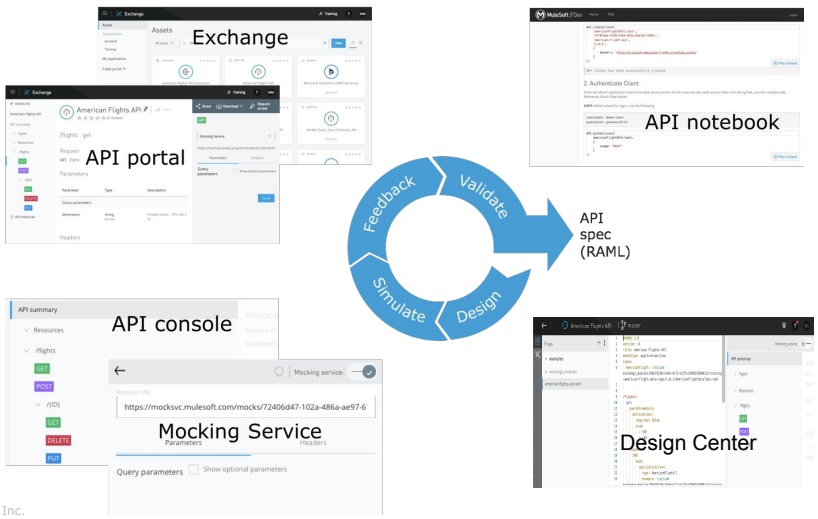
Identifying the components of Anypoint Platform

Anypoint Platform is a collection of runtimes, frameworks, tools, and web applications

- **Tools and frameworks** for building applications
- One **Mule runtime** for running Mule applications and applying policies
  - The same Mule runtime is used in MuleSoft-hosted infrastructure (CloudHub) or in customer-hosted infrastructure (on-premises or in the cloud)
- A suite of **web applications** for
  - Discovering and learning about APIs and other assets
  - Building integration applications that consume APIs
  - Deploying, running, managing, and monitoring applications
  - Defining and managing APIs

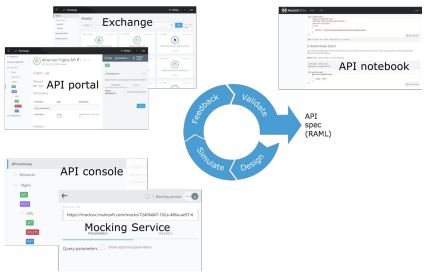
Anypoint Platform components used to develop and manage APIs during the MuleSoft design phase

- A unified platform to design, build, test, share, and manage APIs



Implementing with an API-led approach using the unified Anypoint Platform

- **Mule applications** can first be designed with API specifications that are easier for less technical stakeholders to understand
  - Write, publish, and version APIs in **Anypoint Design Center**
  - Manage APIs with **Anypoint API Manager**
  - Offload API governance and policies from the API implementation (the Mule application) to a centralized management plane used by runtime admins, not developers
  - Share, mock, test, and reuse APIs with **Anypoint Exchange**



MuleSoft recommended REST API specification options

- MuleSoft recommends using a modern, open, flexible API documentation language to model REST services
  - Should be **language agnostic** to easily model XML, JSON, and Java objects in a more readable syntax
  - Should be readable by less technical, more business focused staff
  - Should still allow auto-generation of skeleton implementations by tools
- **REST API Modeling Language** (RAML) is a MuleSoft invented open standard
  - Based on YAML (YAML Ain't Markup Language)
- **OpenAPI Specification** (OAS)
  - Formerly called Swagger
  - Another open standard to define REST interfaces in YAML or JSON format

```
#%RAML 1.0
title: Orders API

/orders:
  get:
  /{orderId}:
    post:
      body:
        application/json:
      responses:
        200:
        404:
```

MuleSoft development tools and the Anypoint Platform fully support RAML and OAS

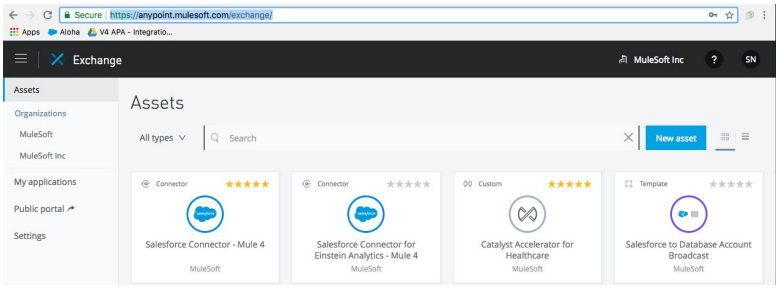
- Supports a **design-first approach**
- Interactions can first be architected and designed using REST API specifications
  - Typically with RAML or OAS
- Anypoint Design Center assists designing, sharing, versioning, and iterating on RAML and OAS specifications
- Based on a RAML or OAS specification
  - Anypoint Studio can **auto-generate skeleton implementation** flows using the APIkit component
  - Anypoint Connect can automatically create and publish an Anypoint Connector to Anypoint Exchange

Deploying, managing, and monitoring Mule applications

- Mule applications are deployed to a **Mule runtime**
  - Mule runtimes can be MuleSoft-hosted in the **cloud (CloudHub) or customer-hosted** in the cloud or on-premises
- A Mule runtime is a lightweight Java-based integration platform
  - Allows developers to connect apps together quickly and easily, enabling them to exchange data
  - Acts as a transit system for carrying data between apps (the Mule)
  - Can connect any systems using any protocols
    - Including HTTP, web services, JDBC, FTP, and JMS

Exercise 2-1: Explore Anypoint Platform and Anypoint Exchange

- Identify asset types supported by Anypoint Exchange
- Identify resources defined in a REST API
- Identify API dependencies and RAML fragments
- Identify how different versions of an API are stored in Anypoint Exchange

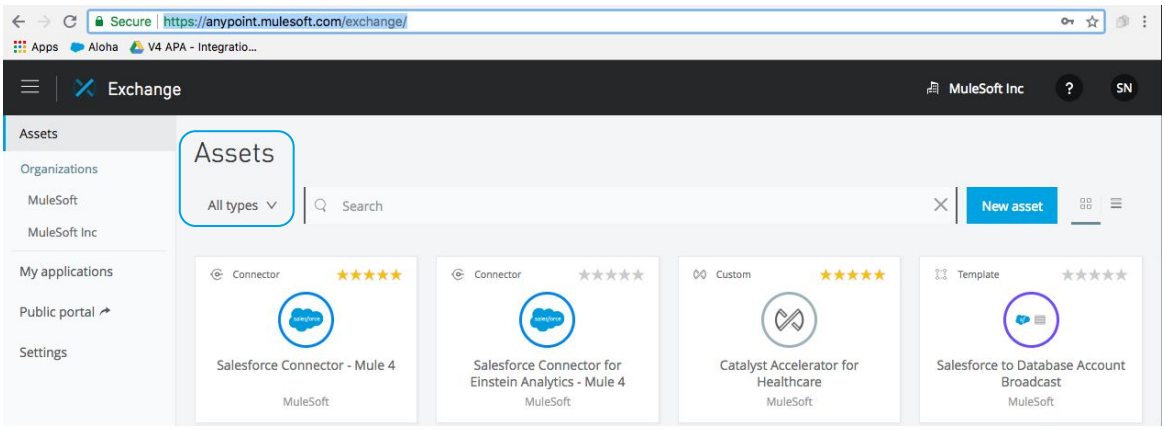


Exercise step

- Log in to Anypoint Exchange using https://anypoint.mulesoft.com/login/#/signin?apintent=exchange

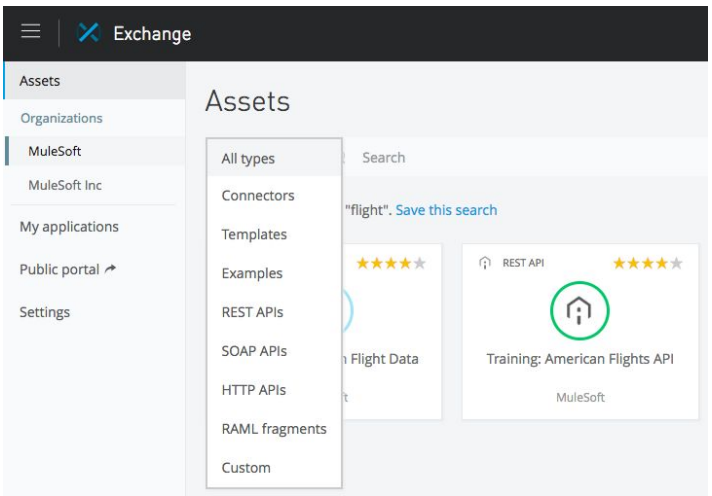
Exercise step

- Review Anypoint Exchange asset types



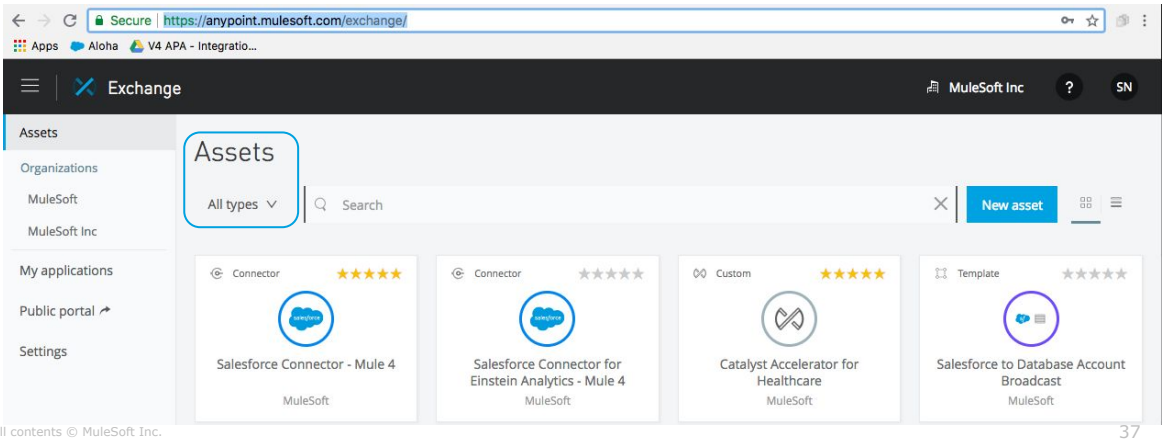
Exercise solution

- Identify Anypoint Exchange asset types



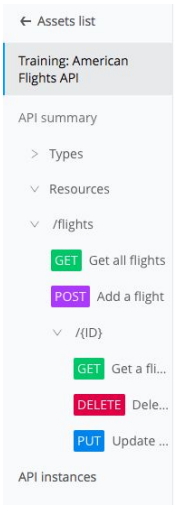
Exercise step

- Locate assets in the public Anypoint Exchange for the American Flights API used in the Anypoint Platform Development: Fundamentals course



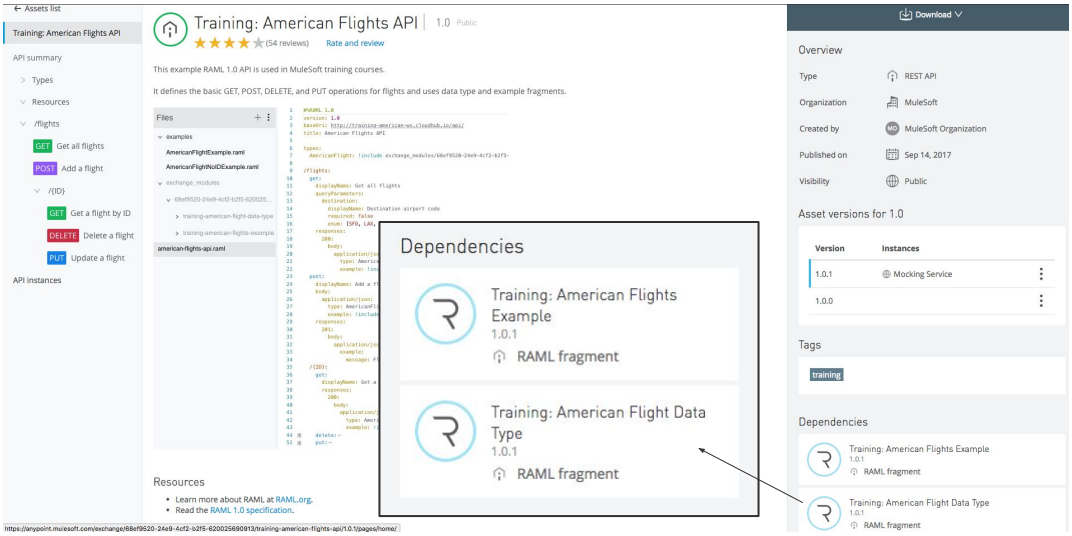
Exercise solution

- Identify REST resources defined in a REST API



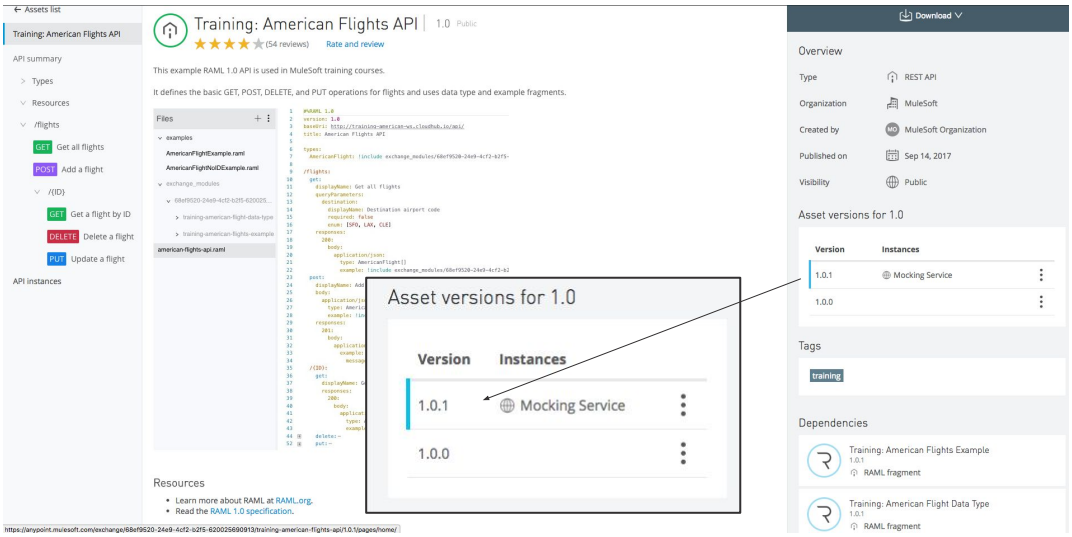
Exercise solution

- Identify API dependencies



Exercise solution

- Select an asset version

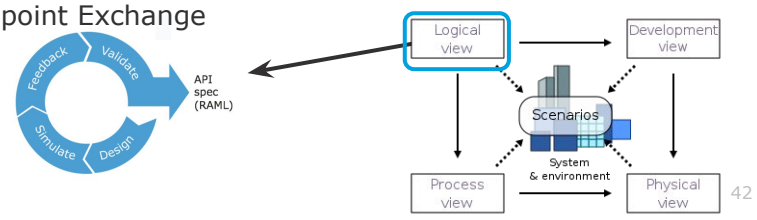


Exercise reflection questions

- If you ever designed or implemented a REST API
  - How did you document the REST API?
  - How RESTful was the API?
  - How easily could someone come in and start refactoring the REST API several years from now?

Some 4+1 views are created during the MuleSoft project design phase

- The 4+1 views drive the initial MuleSoft project design phase, including defining new APIs
- The **Anypoint Platform and Mule applications are an integral part of the design phase**
  - Can be used to **build proof of concepts** that can quickly mock the required user stories, in line with 4+1 views
    - You will use Anypoint Platform and development tools to mock some user stories
  - Often **without writing any code**
  - Custom components might be created or can be simulated with sample data and schema from Anypoint Exchange

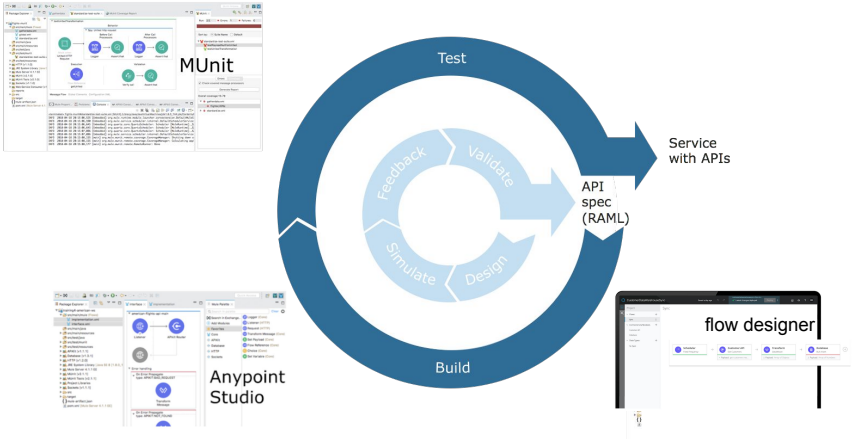


Designing for later lifecycle phases

Components used to **implement** an API or general Mule applications

MuleSoft

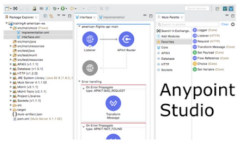
- Anypoint Platform provides tools to implement and test Mule applications
- These tools can be used with or without API specifications



Both flow designer and Anypoint Studio create Mule applications

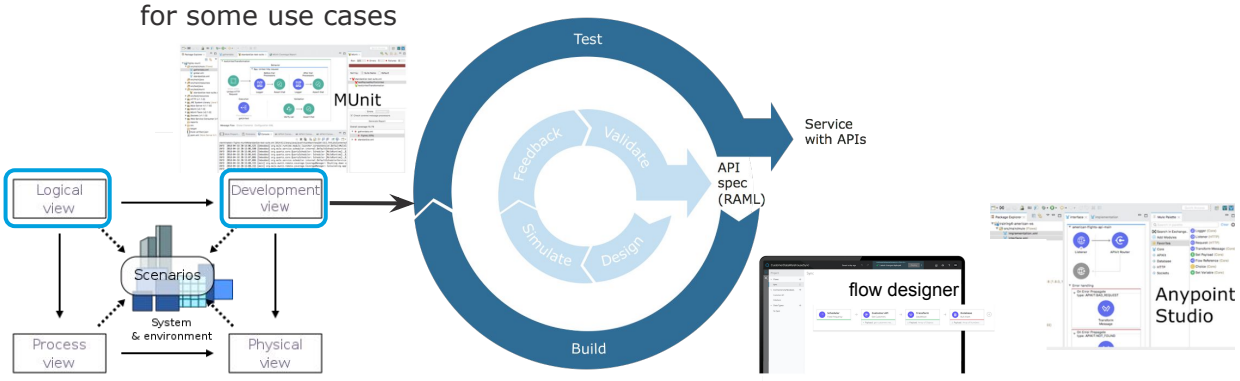
- **Mule applications** can be created in several ways
  - Visually using the online **flow designer or locally using Anypoint Studio**
    - Anypoint Studio provides some more advanced capabilities compared with flow designer
  - In a text editor by writing code (primarily XML) using Anypoint Studio (or other tools)
- Under the hood, Mule applications are **Java applications** (based on Java Spring) that are configured by Mule application XML files

```
<flow name="findFlights" >
    <http:listener doc:name="Listener" config-ref="HTTP_Listener_config" path="flights" />
    <flow-ref doc:name="processQueryParams" name="processQueryParams"/>
</flow>
```

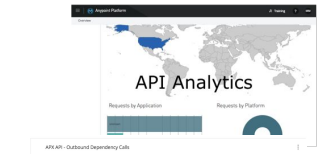


4+1 views are elaborated during the MuleSoft project implementation and testing phases

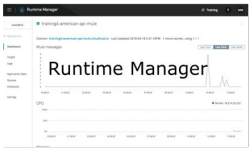
- Feedback from the design phase is used to refine the 4+1 views
- The completed architecture then drives **final implementation and testing**
  - You will use Anypoint Platform and development tools to fill in these views for some use cases

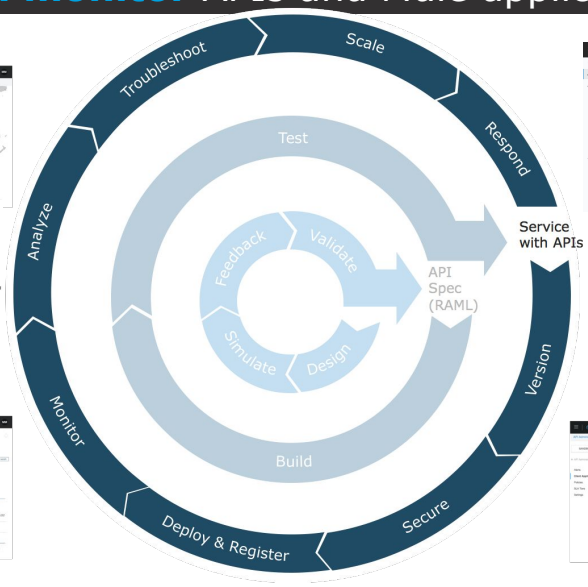


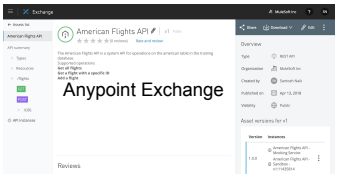
Anypoint Platform components used to **deploy, manage, and monitor** APIs and Mule applications

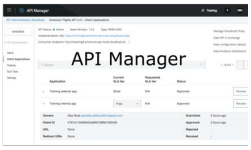


Anypoint Visualizer

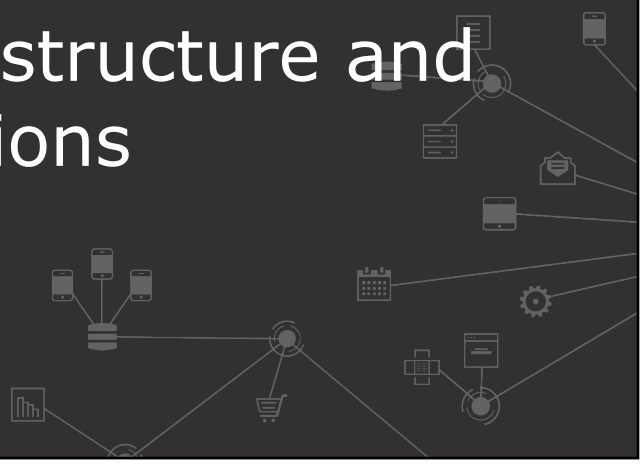






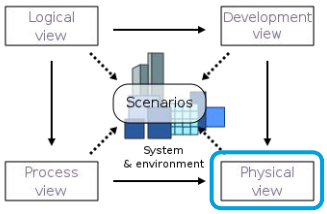


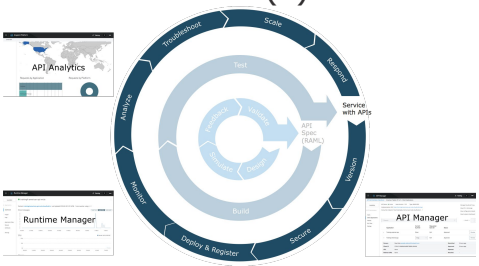
Identifying infrastructure and deployment options



Anypoint Platform deployment options

- The same Mule application can be deployed to either **MuleSoft-hosted or customer-hosted** infrastructure
  - This infrastructure and related services are called the **runtime plane**
- In all cases, the Mule application is deployed to a Mule runtime
- The difference is in how the infrastructure is provisioned and managed to host the Mule runtime(s)



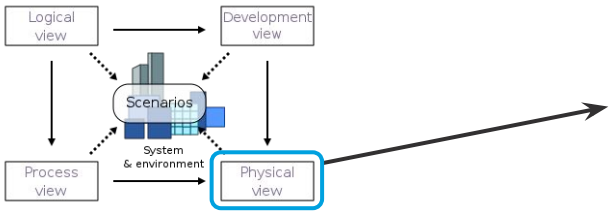


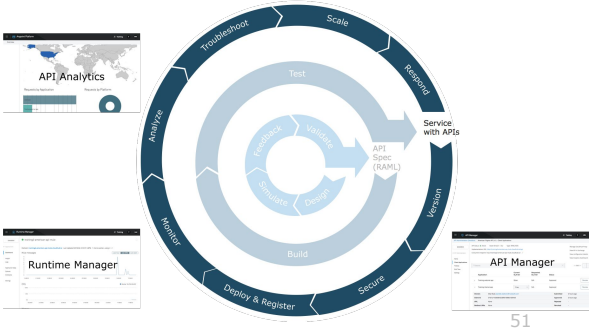
Features supported by all runtime plane types

- Deploying, stopping, starting, and restarting a Mule application through Runtime Manager
- Setting properties from Runtime Manager

Which 4+1 views are created to direct the deployment and maintenance phases

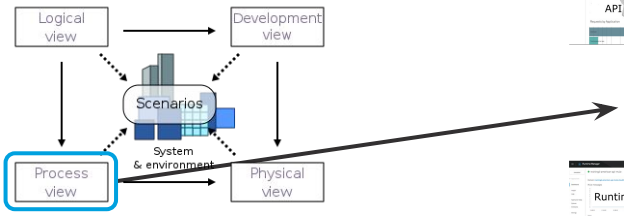
- The physical views are mainly used to design deployment and project maintenance of Mule applications and API implementations
  - The architecture shows **the process and the nodes** that run the process
  - In addition to the 4+1 view, the architecture can also document larger **CI/CD** processes or other automation
  - In this class, you will create all these architecture documents for the class use cases

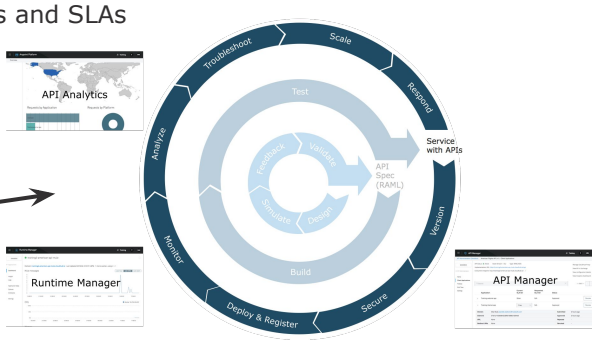




Process views are also used to document distributed data exchange

- The **Process views** also supplement the physical views to decide on deployment infrastructure options
  - Activity diagrams document data flow across systems
  - The Process view communicates NFRs for distributed data exchanges, which then informs decisions in the Process views
    - Performance requirements and SLAs
    - Reliability and HA requirements and SLAs

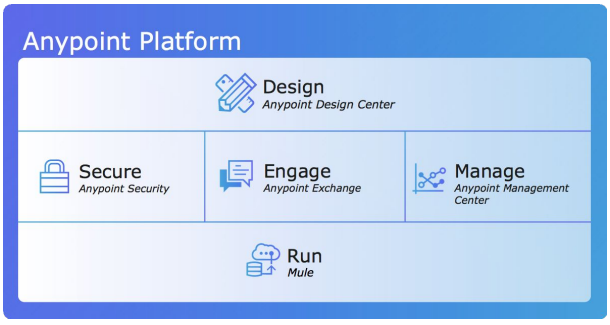




Applying Anypoint Platform components to the course case study

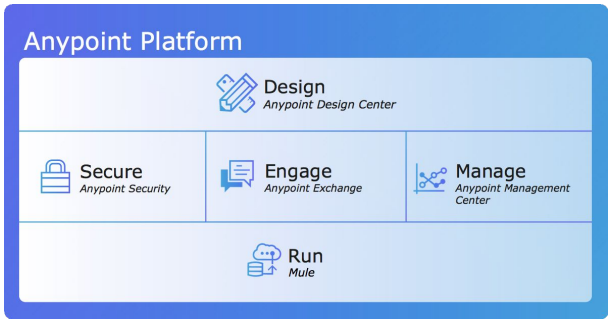
Applying Anypoint Platform to the course case study

- Now you can identify parts of the course case study that can benefit from Anypoint Platform components
  - Anypoint Studio or flow designer
  - API Manager
  - Anypoint Exchange
  - Runtime Manager
  - Anypoint Visualizer
  - Anypoint Monitoring



Exercise 2-2: Align Anypoint Platform components and capabilities with a use case

- Decide which Anypoint Platform components can be applied to meet the functional and non-functional requirements



Exercise step: Identify Anypoint Platform components to meet requirements

Requirement	Anypoint Platform Components	Comments

Exercise steps

- Open the course case study
- Identify which Anypoint Platform components can be applied to meet the functional and non-functional requirements
- Add preliminary sketches of Anypoint Platform components and capabilities to the architecture document

Exercise solution

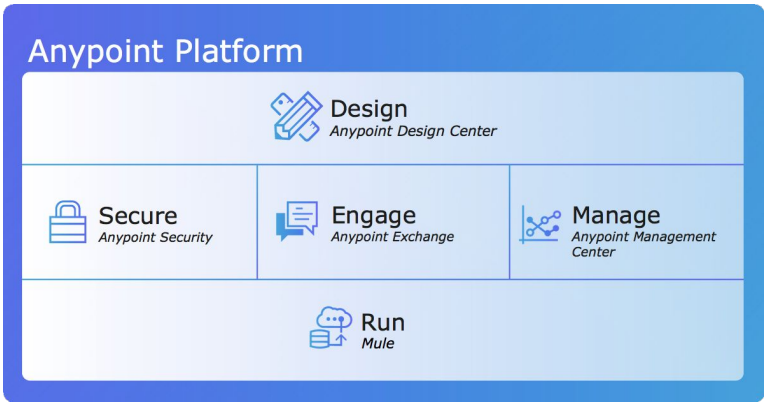
- Open the solution architecture document from your student files
- Compare your architecture document with the provided solution architecture document

Summary

Summary

MuleSoft

- A **unified**, highly productive, **hybrid** integration platform that creates a seamless distributed system of apps, data, and devices
- Can also manage full API lifecycles to promote API-led development



Summary

- Use **Anypoint Exchange** as a central repository for assets so they can be discovered and reused
  - Populate it with everything you need to build your integration projects.
- Use **flow designer or Anypoint Studio** to build integration applications
- **Mule runtimes** can be MuleSoft-hosted in the cloud (CloudHub) or customer-hosted in the cloud or on-prem