



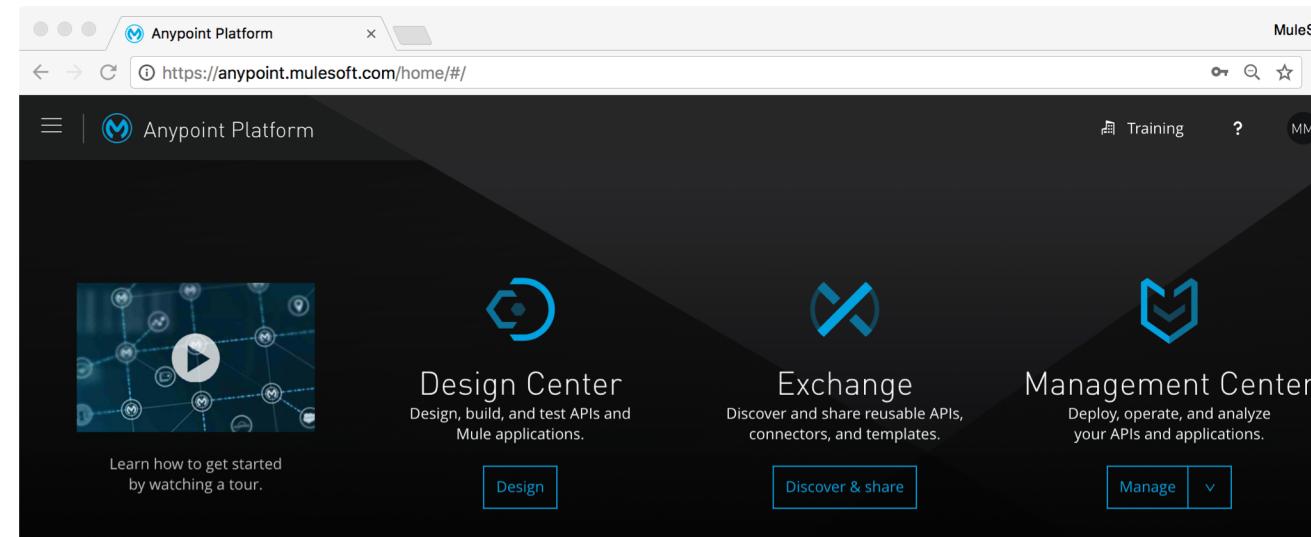
Module 2: Introducing Anypoint Platform



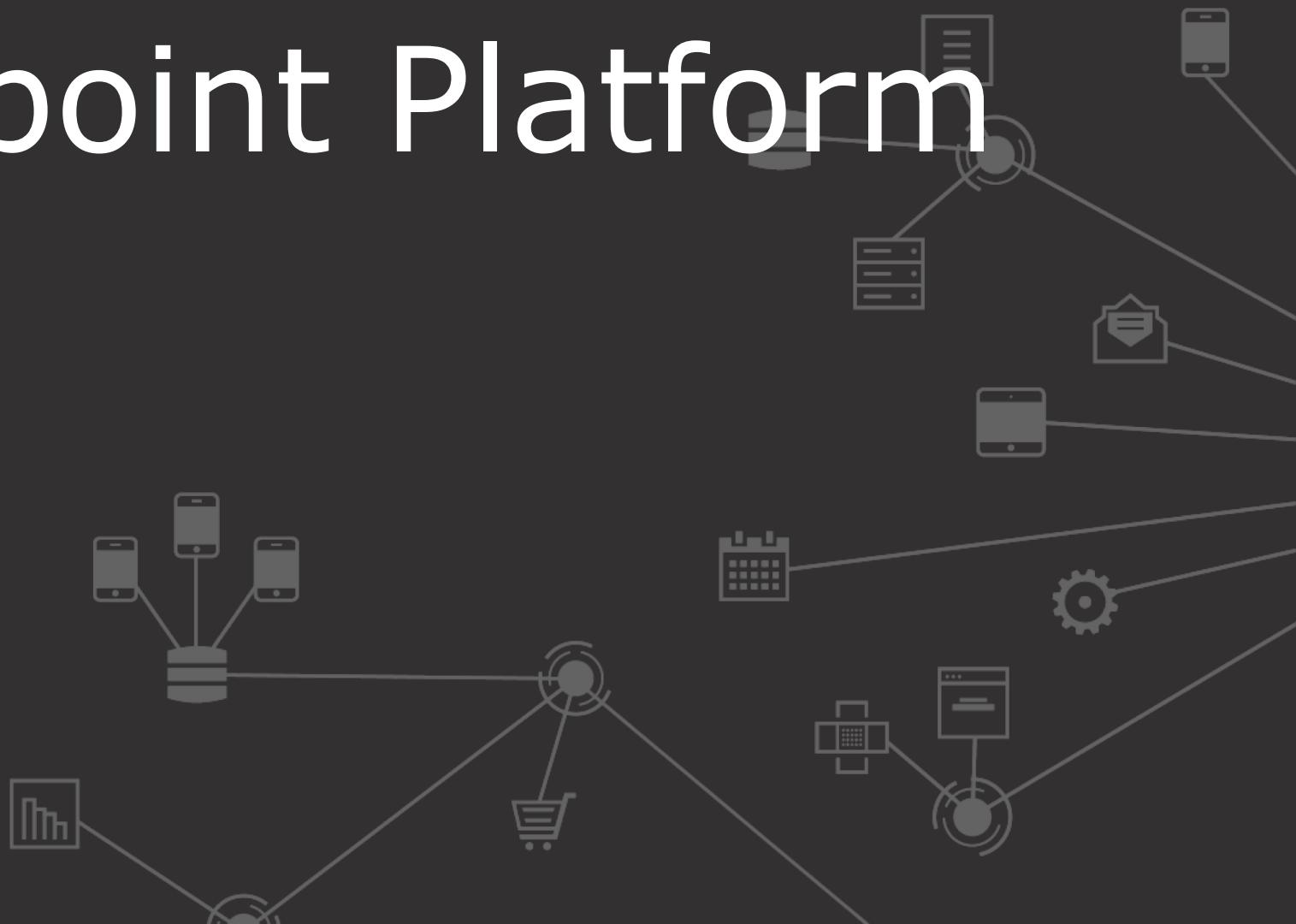
At the end of this module, you should be able to



- Describe the benefits of Anypoint Platform and MuleSoft's approach to be successful with it
- Describe the role of each component in building application networks
- Navigate Anypoint Platform
- Locate APIs and other assets needed to build integrations and APIs in Anypoint Exchange
- Build basic integrations to connect systems using flow designer



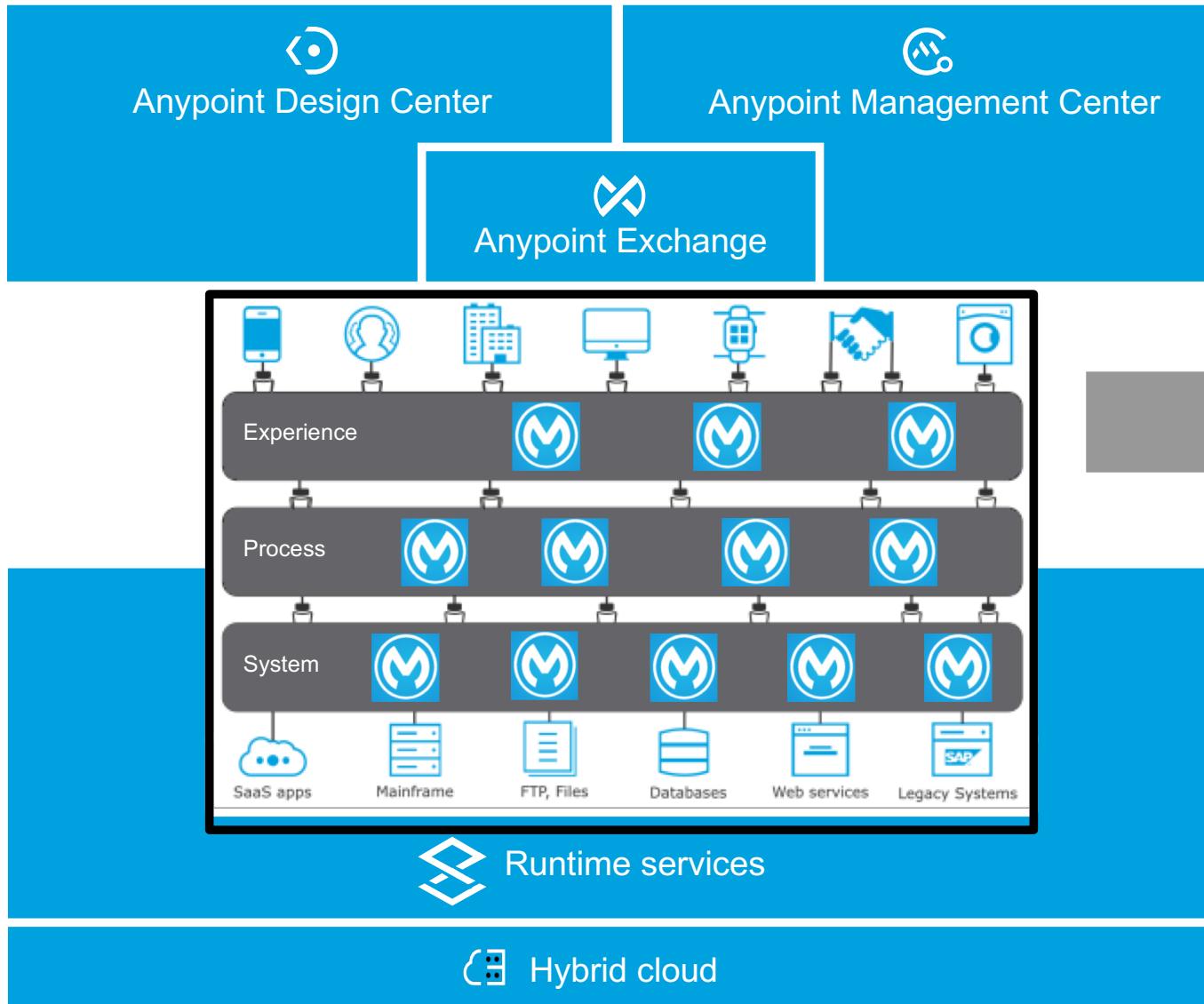
Introducing Anypoint Platform



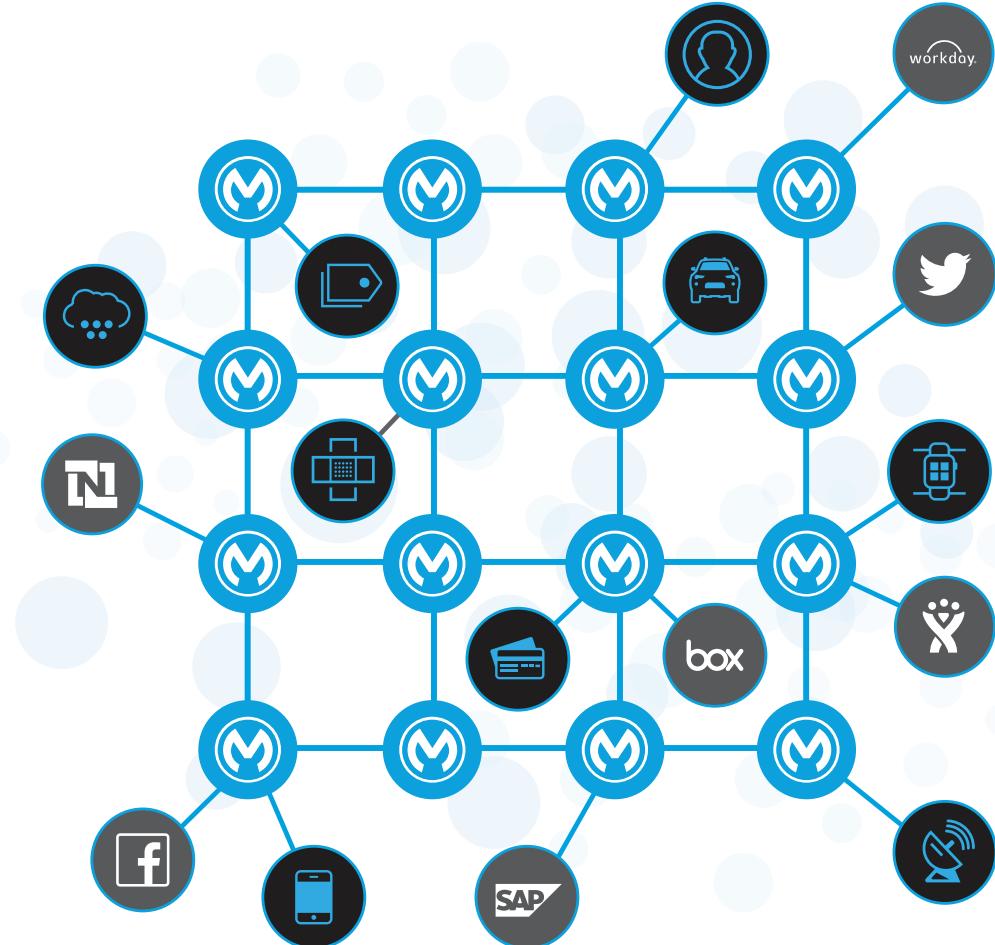
Anypoint Platform uniquely enables the building of an application network

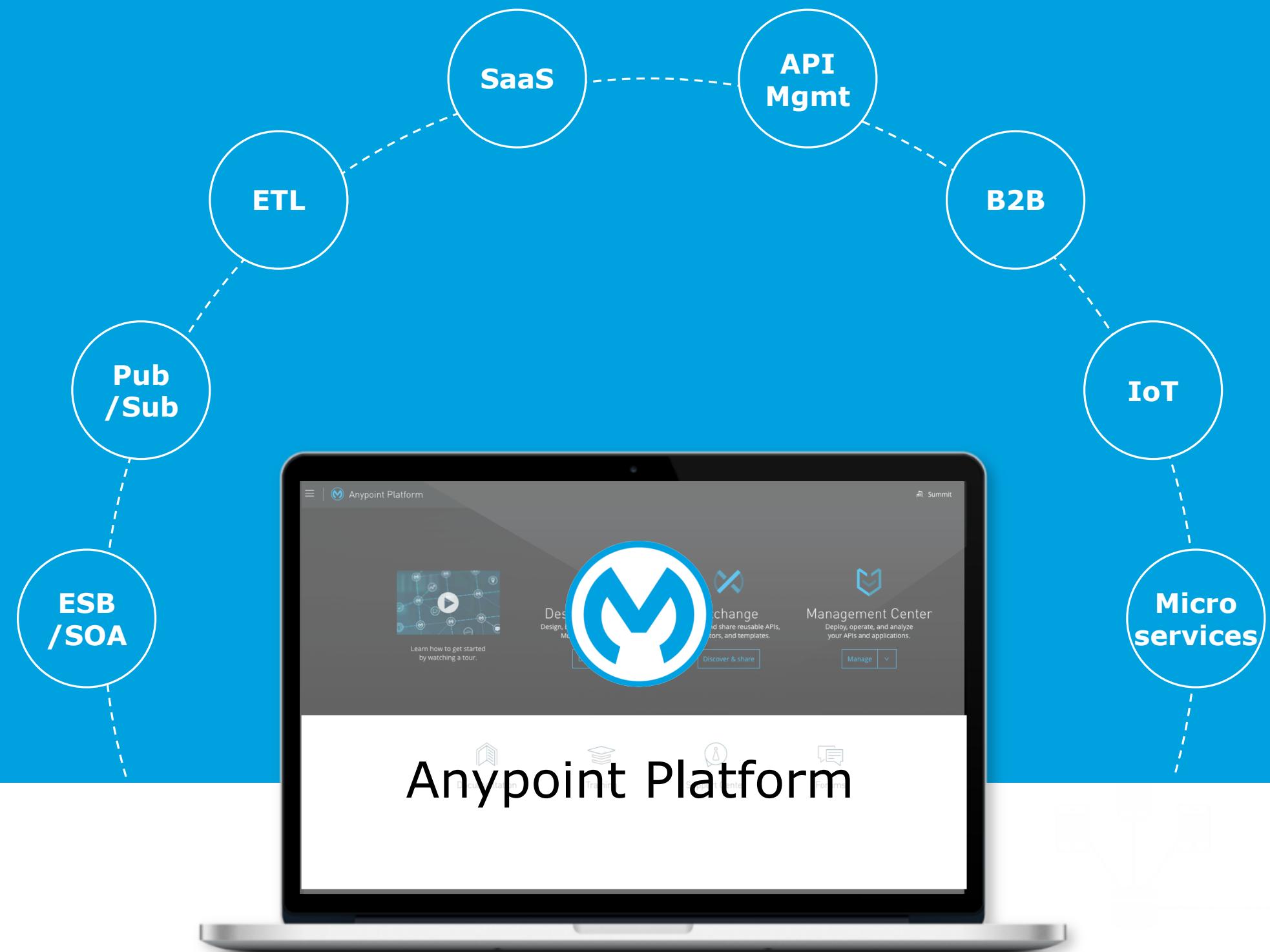


Anypoint Platform



Application network





The most advanced enterprise platform for designing, developing and managing APIs and integrations

- Uniquely built as a single product
- Deploy anywhere
- Wide range of use cases

1 Design Center, 1 Mgmt Center, 1 Runtime



Specialists



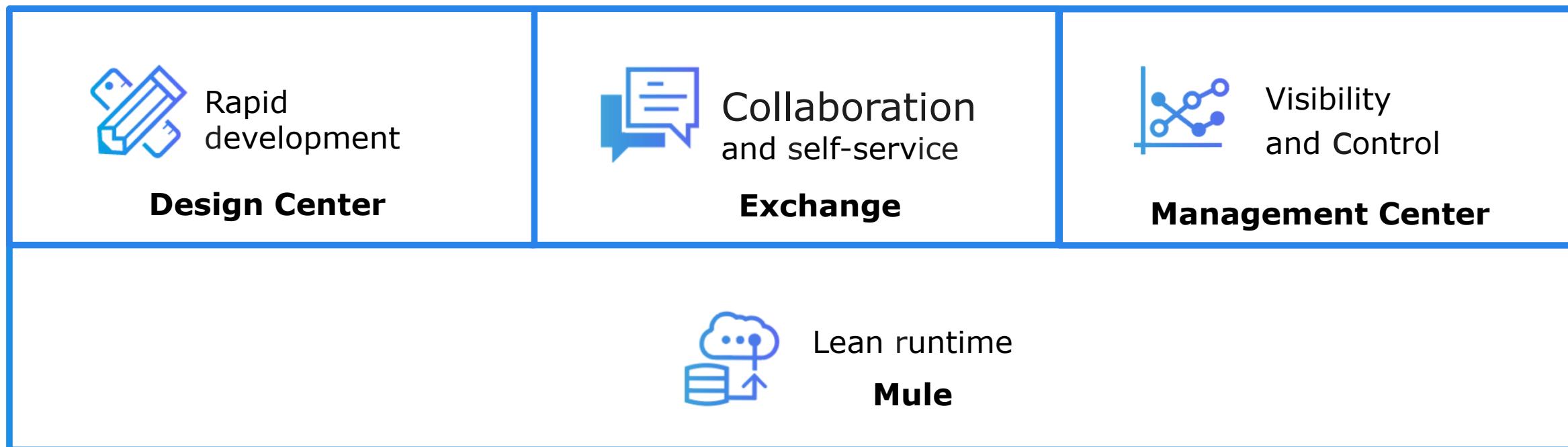
Admin, Ops,
DevOps



Ad-hoc
integrators



App devs



On-premises &
Private Cloud



Hybrid



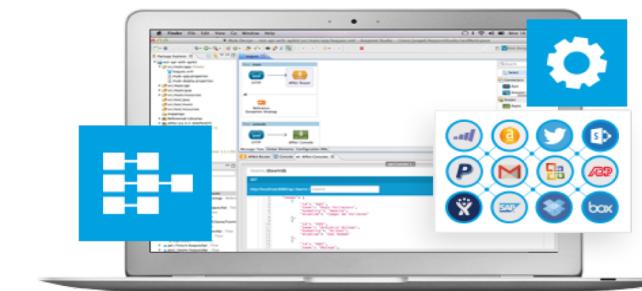
Hosted by
MuleSoft



Cloud service providers

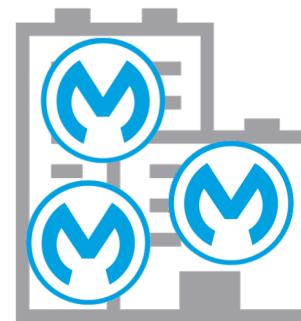


Design once, deploy anywhere



Anypoint Design Center

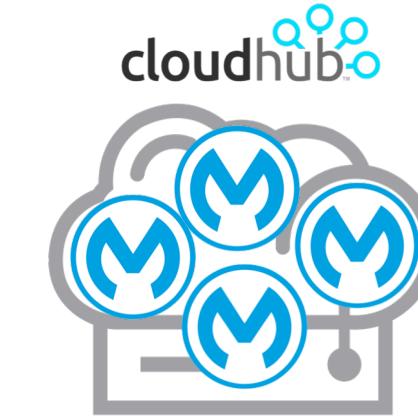
Anypoint Management Center



On-prem



Private cloud



fully managed iPaaS



Databases



FTP, Files



Web services



SaaS Apps



On-prem Apps



Social Apps

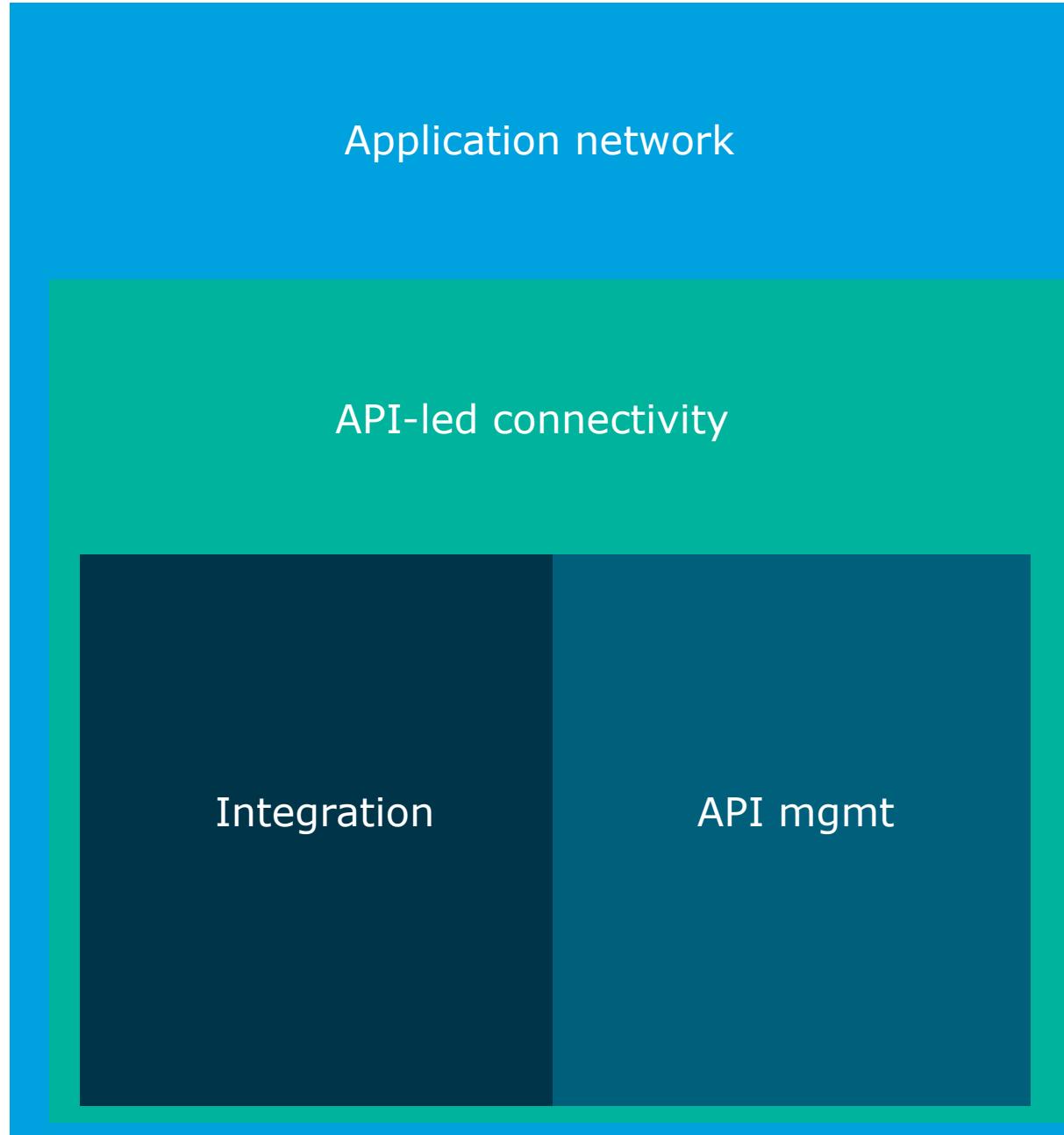


IoT



Partners

Unique benefits for each layer



Speed of delivery



Actionable visibility



Secure by design



Future-proof architecture



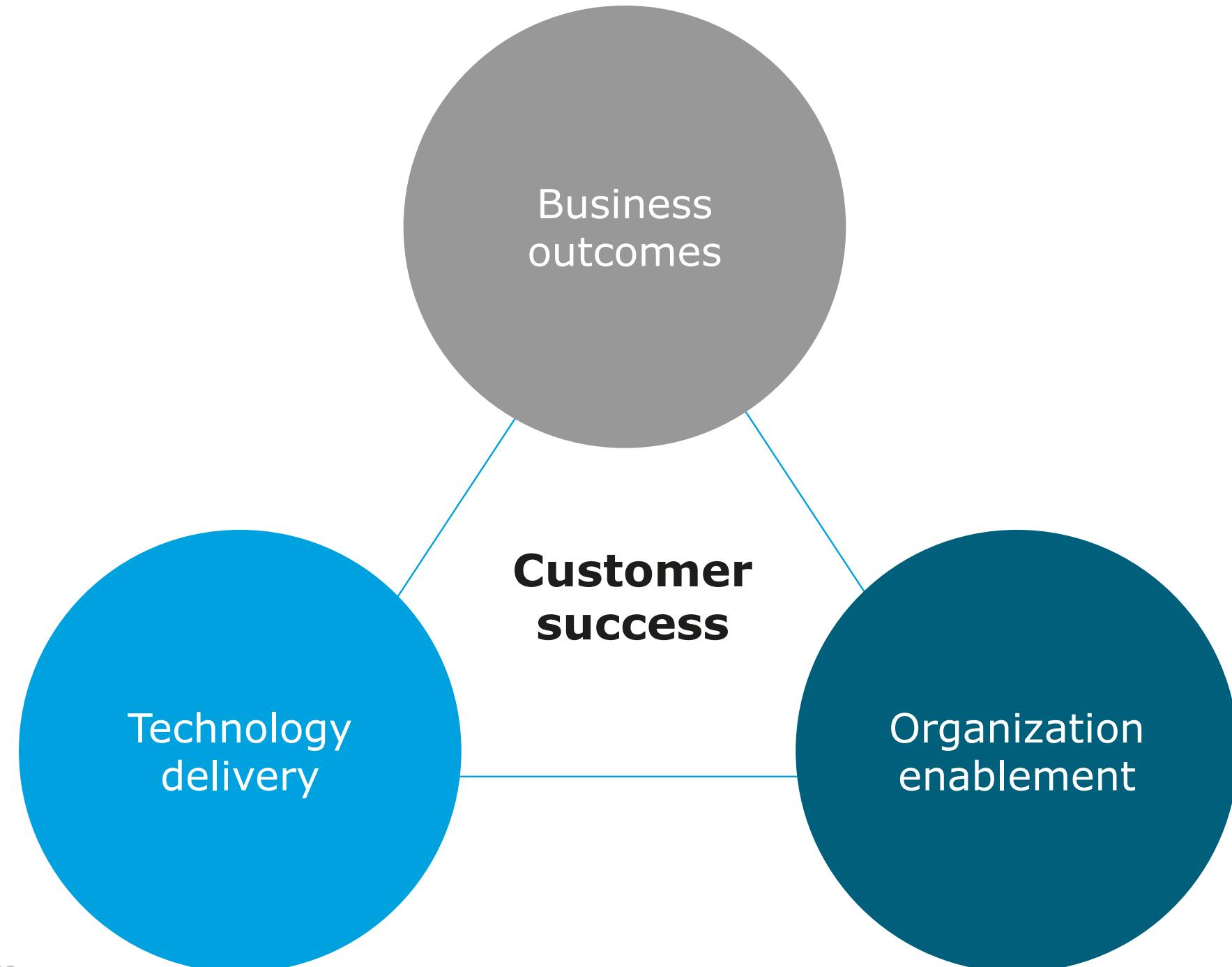
Intentional self-service

Achieving success with Anypoint Platform

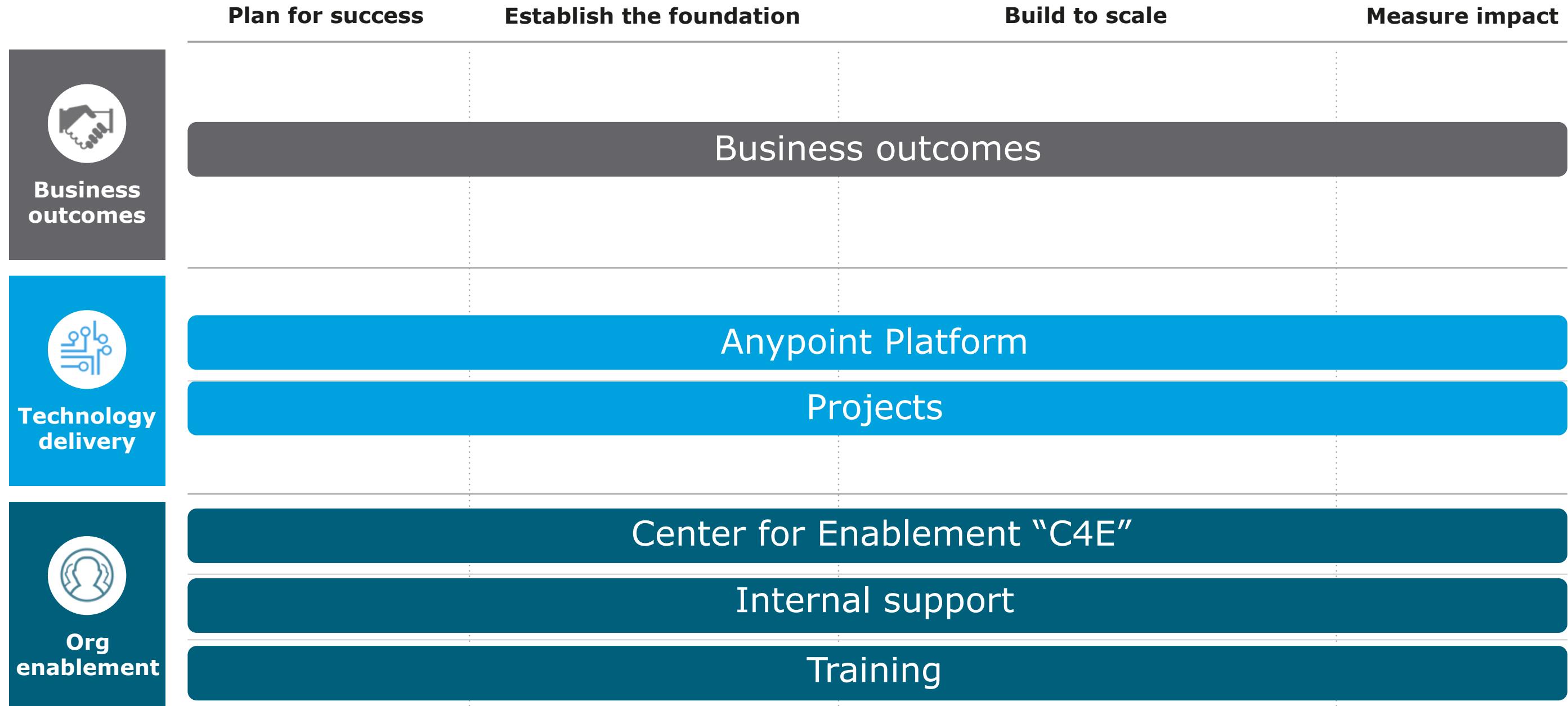


The background features a faint, light-gray network diagram composed of several interconnected nodes. These nodes include a central circular node, a rectangular node with a bar chart icon, a square node with a shopping cart icon, a gear-shaped node, a calendar node, a server rack node, a mail envelope node, a smartphone node, and a document icon node. Lines connect these nodes in a complex web, symbolizing the integration and connectivity provided by the Anypoint Platform.

MuleSoft's approach is centered around 3 core pillars



Defining the path to achieve success

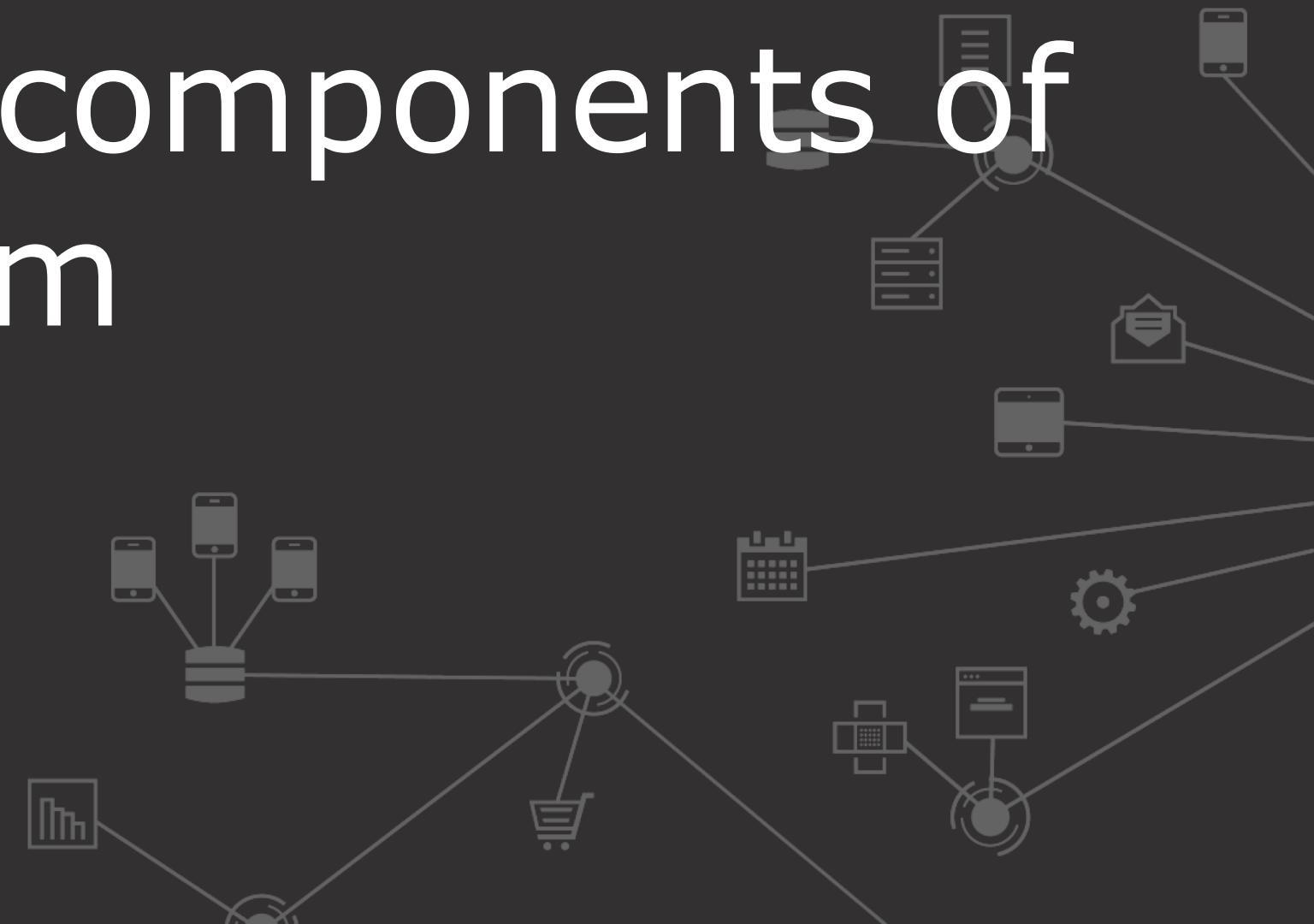


MuleSoft has a blueprint for you to follow

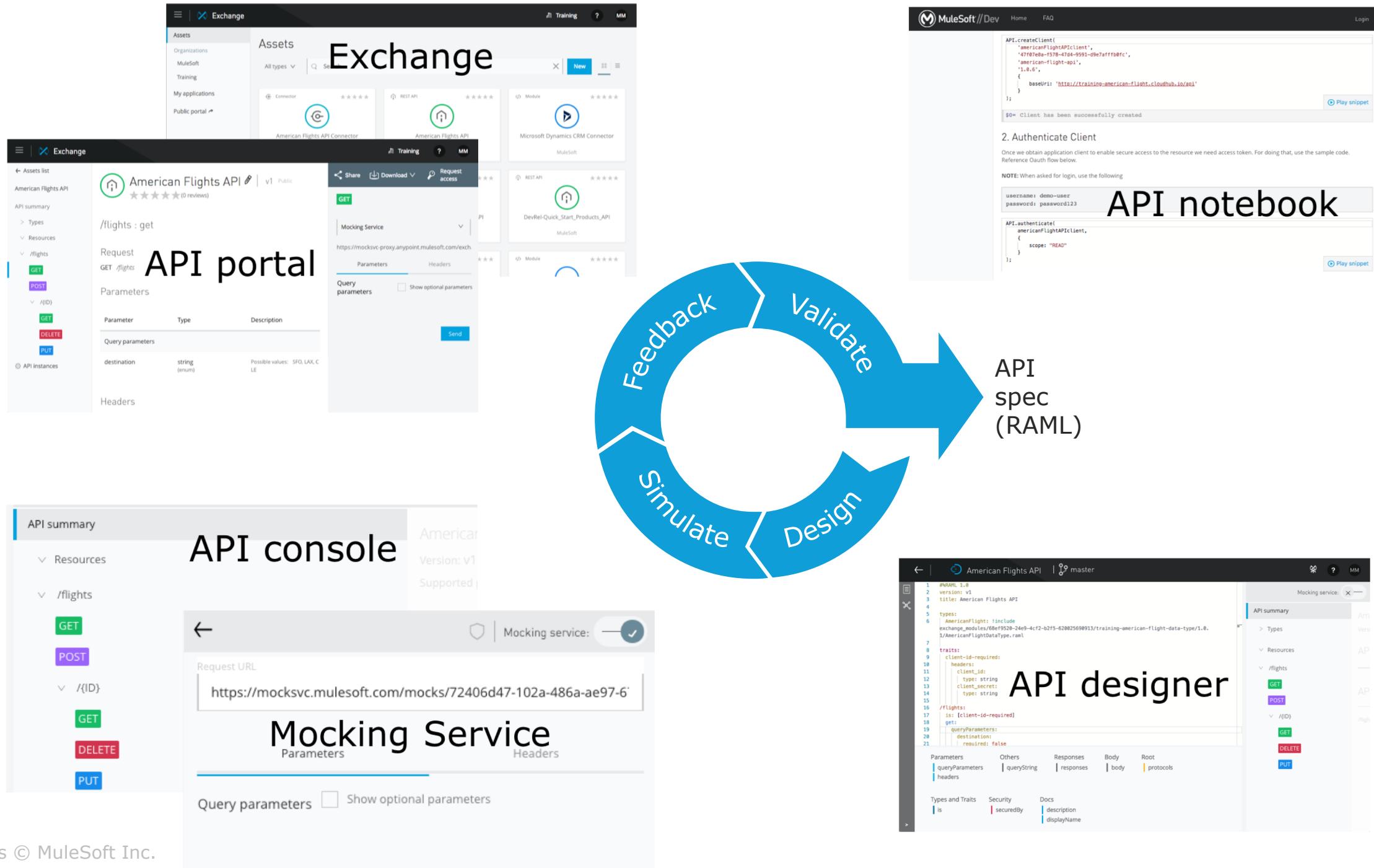


	Plan for success	Establish the foundation	Build to scale	Measure impact
Business outcomes	Agree on business outcomes and KPIs Develop the overall success plan	Monitor and manage	Refresh the success plan	Measure business outcomes
Technology delivery	Define Anypoint platform vision and roadmap Design Anypoint platform architecture and implementation plan	Deploy Anypoint Platform	Refine and scale Anypoint Platform	Measure Anypoint platform KPIs
	Prioritize IT projects and quick wins Staff and onboard the project teams	Define reference architecture Launch initial projects and quick wins	Onboard additional project teams Launch additional projects	Measure project KPIs
	Assess integration capabilities Establish the C4E operating model	Build and publish foundational assets Evangelize	Drive consumption	Measure C4E KPIs
Org enablement	Onboard MuleSoft Determine the internal support operating model	Staff, train and launch team Publish support guidance and self-serve materials	Monitor Anypoint Platform	Measure support KPIs
	Agree on initial roles Train the initial team(s)	Develop the broader training plan Launch experiential learning opportunities	Update training plan	Conduct skills assessment

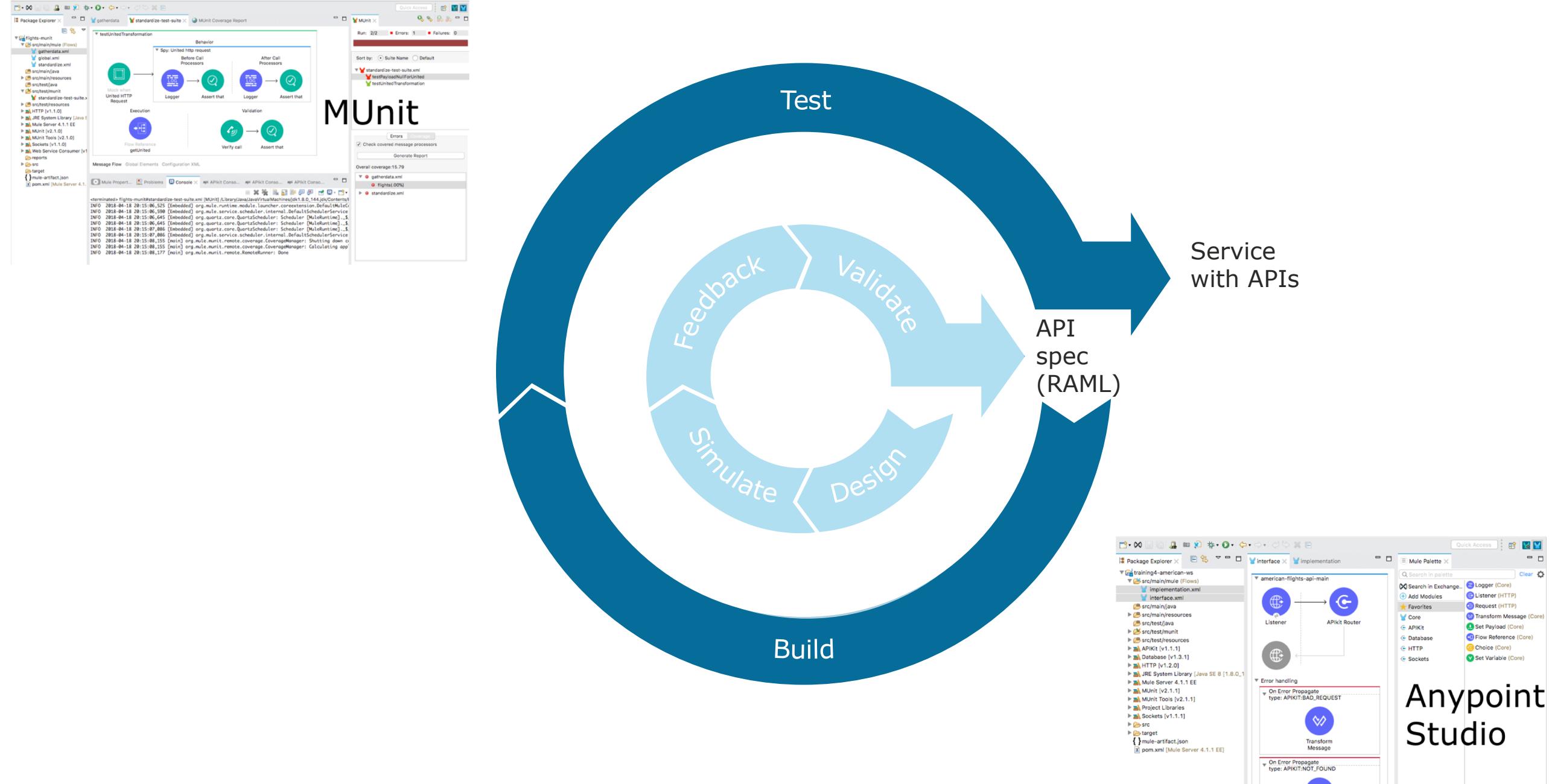
Introducing the components of Anypoint Platform



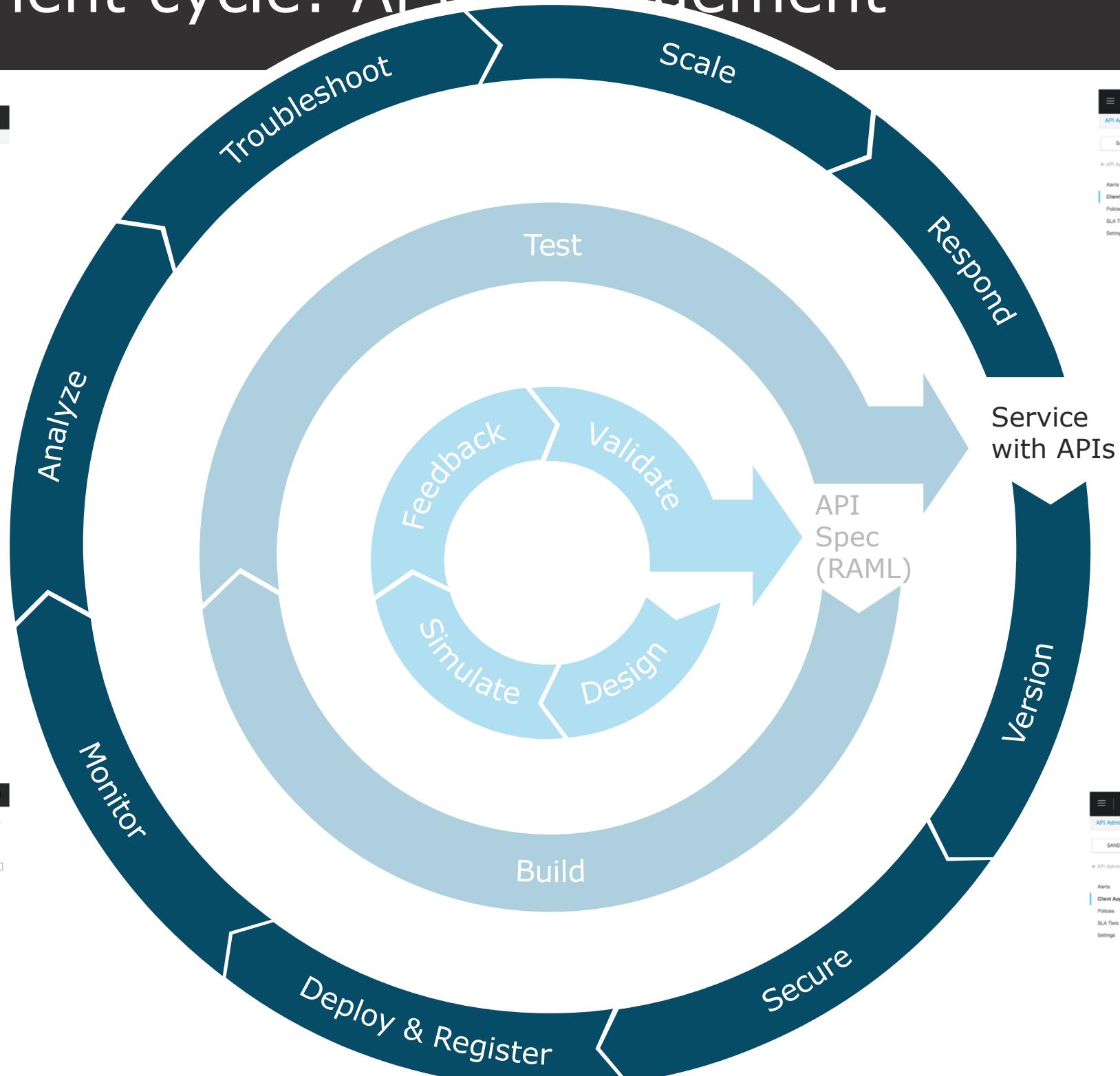
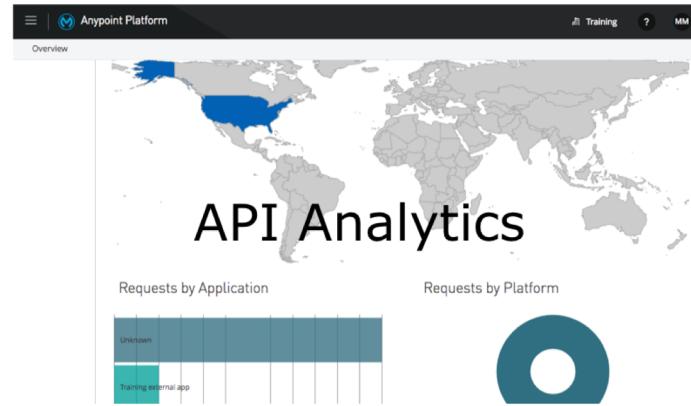
API development cycle: API specification



API development cycle: API implementation

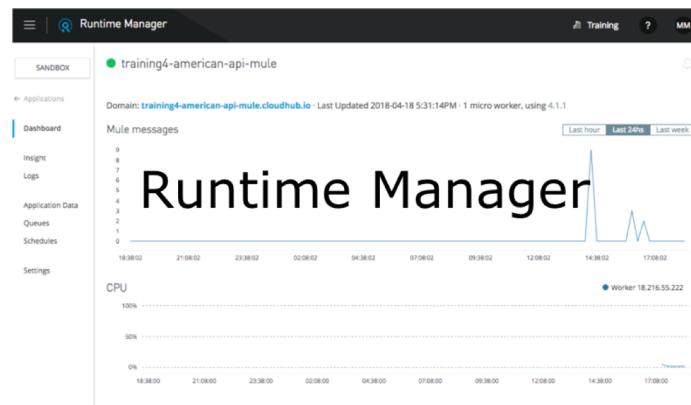


API development cycle: API management

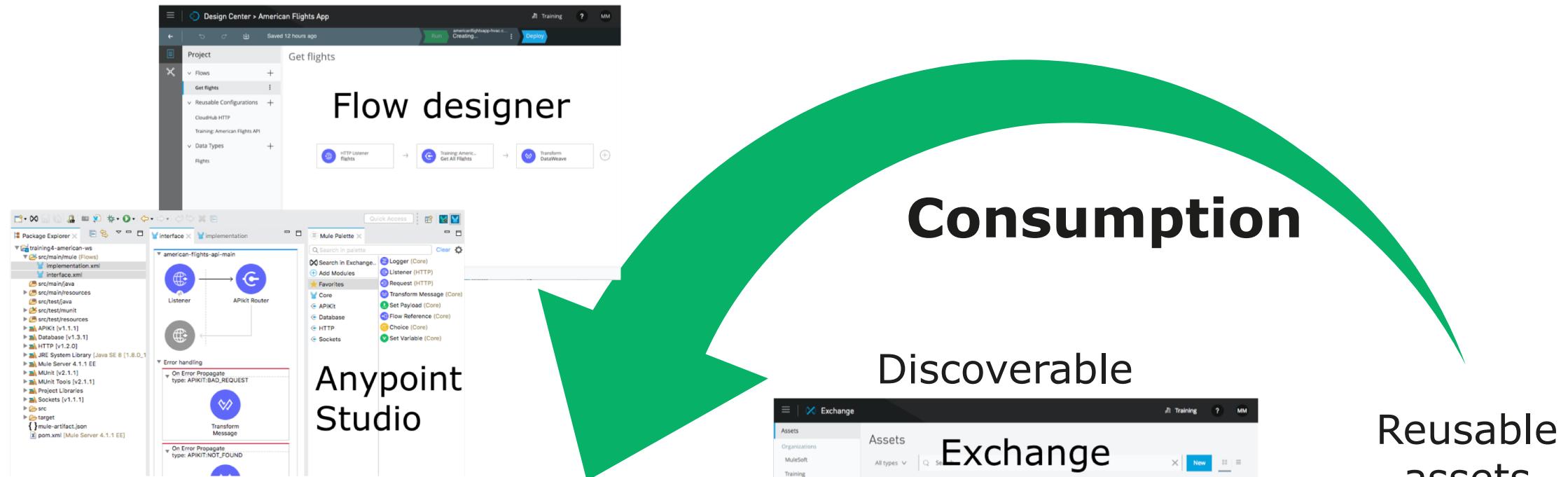


API Manager

API Manager



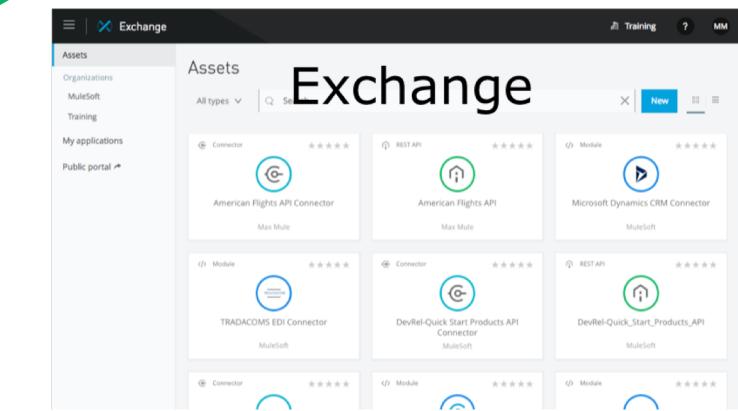
API lifecycle: Discovery and consumption



Anypoint
Studio

Flow designer

Feedback and
usage metrics



Production

Discoverable
Exchange

Reusable
assets

Consumption

Anypoint Exchange



- A library of assets
- The central repository that is critical to the success of building an application network
- Ensures assets are published somewhere they can be discovered and reused

A screenshot of the Anypoint Exchange interface. The top navigation bar includes a menu icon, the 'Exchange' logo, 'Training', a help icon, and a user profile icon. The left sidebar has tabs for 'All', 'MuleSoft', 'Training', 'My applications', and 'Public Portal'. The main area is titled 'All assets' with a 'Search' bar and a 'New' button. It displays four asset cards: 'American Flights API Connector' (Connector, 5 stars, Max Mule), 'American Flights API' (REST API, 5 stars, Max Mule), 'LDAP Connector' (Module, 5 stars, MuleSoft), and 'Amazon RDS Connector' (Module, 5 stars, MuleSoft).

Type	Name	Rating	Published By
Connector	American Flights API Connector	★★★★★	Max Mule
REST API	American Flights API	★★★★★	Max Mule
Module	LDAP Connector	★★★★★	MuleSoft
Module	Amazon RDS Connector	★★★★★	MuleSoft

What does (and should) Exchange contain?



- MuleSoft-provided **public** assets available in all accounts to all users
 - You can work with MuleSoft to get APIs and connectors certified and added
- **Private** content only available to people in your org
 - Assets added by anyone in your org are added to your private Exchange
- Your organization should populate it to contain everything you need to build your integration projects
 - Including APIs, connectors, diagrams, videos, links, and more

All types
Connectors
Templates
Examples
REST APIs
SOAP APIs
HTTP APIs
RAML fragments
Custom

- When a REST API is added to Exchange, an **API portal** is automatically created for it
- An API portal has
 - Auto-generated **API documentation**
 - An **API console** for consuming and testing APIs
 - An **automatically generated API endpoint** that uses a **mocking service** to allow the API to be tested without having to implement it
- API portals can be shared with both internal and external users
- In the last module, you used a public API portal created from Anypoint Exchange for a private organization (Muletraining)

REST connectors in Anypoint Exchange

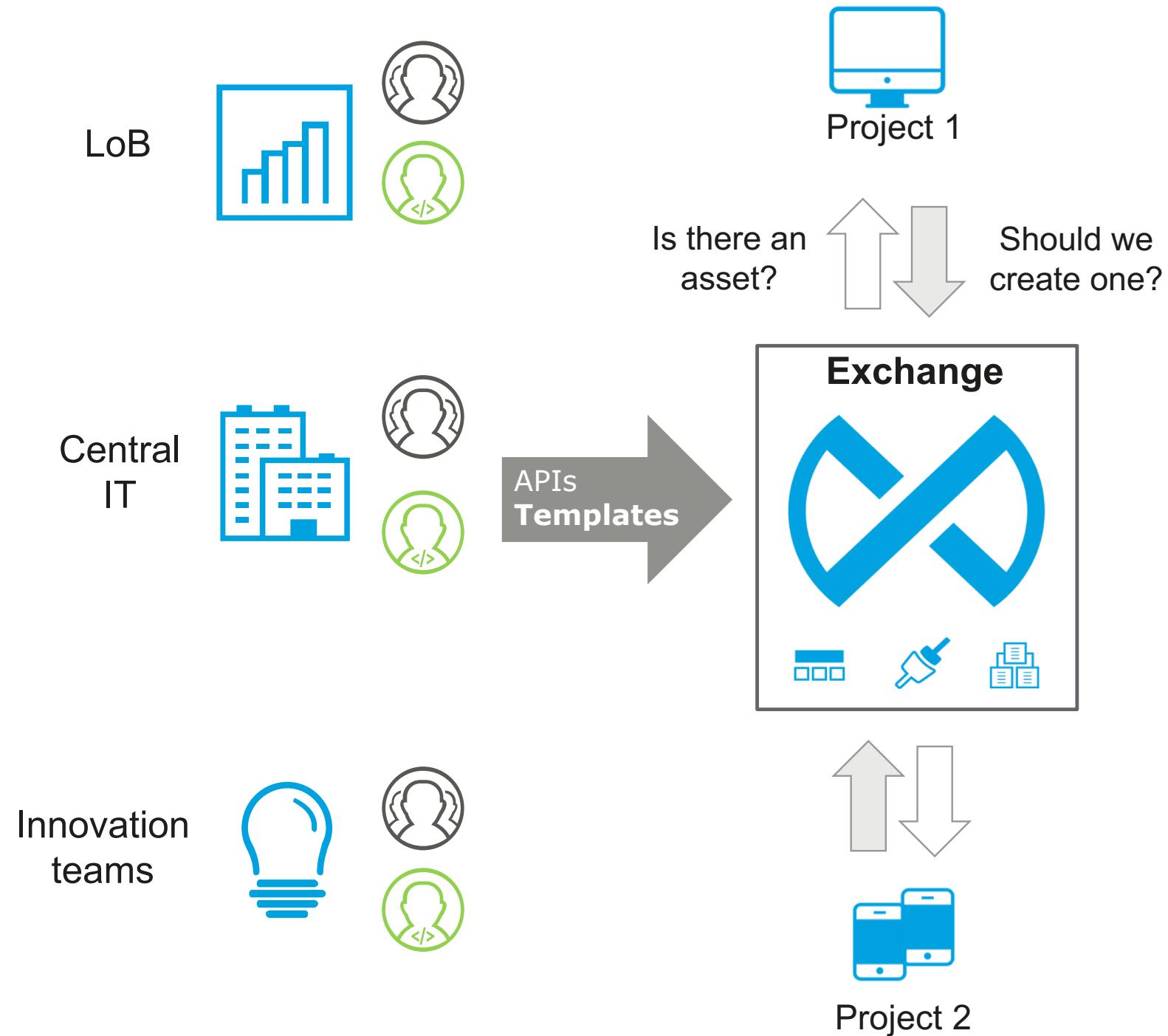


- When a RAML 1.0 API specification is added to Exchange, a **connector** is automatically created for it
 - The connector can be used in Mule applications to make calls to that API
 - REST Connect is the name of the technology that performs this conversion

A screenshot of the Anypoint Exchange web interface. The top navigation bar includes 'Training' and a user icon. The left sidebar has sections for 'All', 'MuleSoft', 'Training', 'My applications', and 'Public Portal'. The main area is titled 'All assets' and shows a search bar. Below the search bar are four cards representing connectors:

- American Flights API Connector** (Max Mule) - A connector with a teal circular icon containing a 'C'.
- American Flights API** (Max Mule) - A REST API with a green circular icon containing a house symbol.
- LDAP Connector** (MuleSoft) - A module with a blue circular icon containing the letters 'LDAP'.
- Amazon RDS Connector** (MuleSoft) - A module with a blue circular icon containing the letters 'RDS'.

Using Exchange: Success of C4E in action



Walkthrough 2-1: Explore Anypoint Platform and Anypoint Exchange

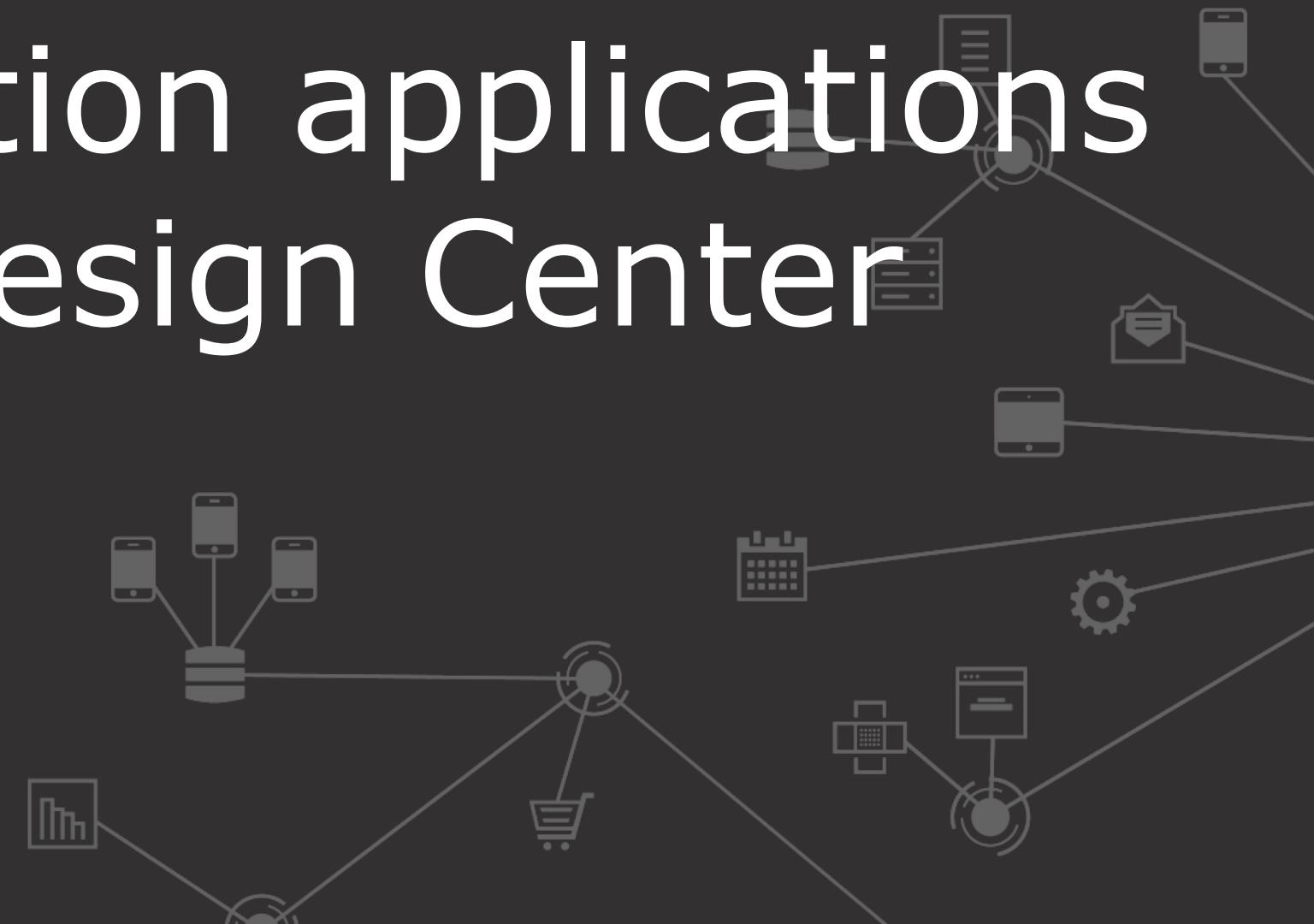


- Explore Anypoint Platform
- Browse Anypoint Exchange
- Review an API portal for a REST API in Exchange
- Discover and make calls to the Training: American Flights API in the public Exchange

A screenshot of the Anypoint Exchange web interface. The top navigation bar includes a menu icon, the 'Exchange' logo, a search icon, 'Training' link, a help icon, and a user icon. The left sidebar has a 'Assets' tab selected, along with 'Organizations', 'MuleSoft', 'Training', 'My applications', and 'Public portal' options. A search bar at the top right contains the text 'american'. Below the search bar, a message says 'Showing results for "american". Save this search'. Three API items are listed:

- REST API: Training: American Flights API (MuleSoft) - 5 stars
- RAML Fragment: Training: American Flight Data Type (MuleSoft) - 5 stars
- RAML Fragment: Training: American Flights Example (MuleSoft) - 5 stars

Building integration applications and APIs with Design Center



Design Center anatomy



Design Center

Training ? MM

Projects

Search...

Get Started
+ Create

American Flights App

Created with API designer

Created with flow designer

Details

Name	Project Type	Last Update
American Flights Example	API Fragment	July 27th, 2017
MUA Flights API	API Specification	July 27th, 2017
MUA Flight Data Type	API Fragment	July 27th, 2017
American Flight Data Type	API Fragment	July 27th, 2017
American Flight Example	API Fragment	July 27th, 2017
American Flights App	Mule Application	July 27th, 2017
Training American Flights API	API Specification	July 27th, 2017

Get Anypoint Studio

Modified July 27th, 2017

Created July 27th, 2017

Created by stallons

Environment 0d6bd95d-e1d2-461d-8b2e-ad0ab31edd64

Status Ready to deploy

Deployment url americanflightsapp-jlkb.cloudhub.io

Open

Design Center applications



Application	Purpose	In this course	Additional courses
flow designer	Web app for building integration apps that connect systems and consume APIs	2 WTs	<ul style="list-style-type: none">• Anypoint Platform: Flow Design
API designer	Web app for designing, documenting, and mocking APIs	Module 3	<ul style="list-style-type: none">• Anypoint Platform: API Design
Anypoint Studio	Desktop IDE for implementing APIs and building integration applications	Module 4 In Fundamentals: Modules 6-13	

Both flow designer and Anypoint Studio create Mule applications



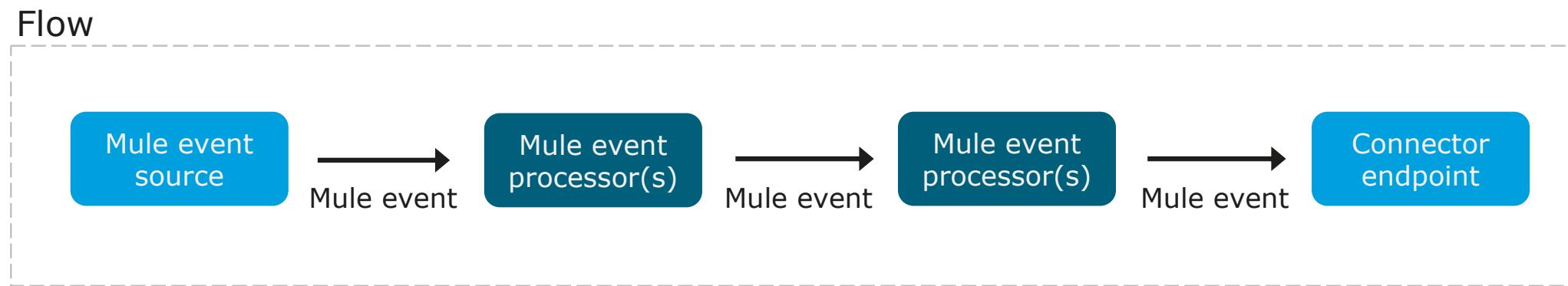
- **Mule applications** can be created
 - Visually using flow designer or Anypoint Studio
 - By writing code (primarily XML) using Anypoint Studio (or other tools)
- Under the hood, Mule applications are Java applications using Spring
- Mule applications are deployed to a **Mule runtime**
 - Mule runtimes can be MuleSoft-hosted in the cloud (CloudHub) or customer-hosted in the cloud or on-prem

- **A lightweight Java-based enterprise service bus (ESB) and integration platform** that 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 all systems including web services, JMS, JDBC, HTTP, & more
- **Decouples point-to-point integrations** by having all (non-Mule) apps talk to the bus (to a Mule runtime) instead of directly to each other
- **Can be deployed anywhere**, can integrate and orchestrate events in real time or in batch, and has universal connectivity
- **Enforces policies for API governance**

Mule 4 applications and flows



- Mule applications receive events, process them, and route them to other endpoints
- **Mule applications** accept and process a **Mule event** through a series of **Mule event processors** plugged together in a **flow**

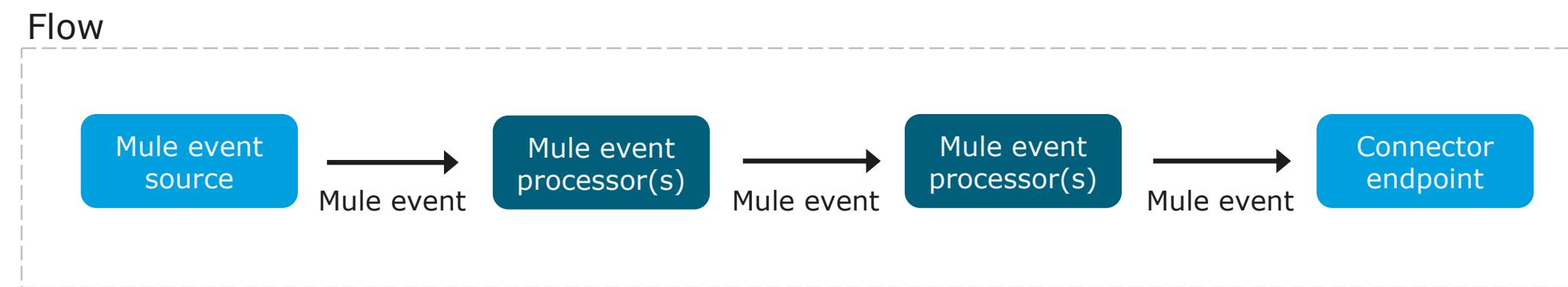


- An application can consist of
 - A single flow
 - Multiple flows
 - Multiple flows connected together

What's in a typical Mule 4 flow?



- A **Mule event source** that initiates the execution of the flow
 - Can be triggered by an event like
 - A consumer request from a mobile device
 - A change to data in a database
 - The creation of a new customer ID in a SaaS application
- **Mule event processors** that transform, filter, enrich, and process the event and its message



Creating integration applications with flow designer



Flow designer anatomy



Design Center > American Flights App

Saved 8 minutes ago Latest changes applied Running Deploy

Project explorer

Get flights

Application status

Cards

Logs

HTTP Listener flights

Training: American Flights API

Transform DataWeave

Flights

Logs

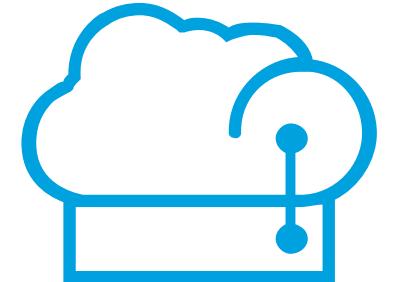
System logs:

```
* Application: americanflightsapp-hvac
* OS encoding: UTF-8, Mule encoding: UTF-8
*
*****
SYSTEM 06:22:35 Worker(18.220.102.247): Your application has started successfully.
SYSTEM 06:22:35 Your application is started.
INFO 06:22:48 ++++++
```

Running flow designer applications



- When you create a Mule application project in Design Center
 - A new application is created and opened in flow designer
 - **The application is deployed to a MuleSoft-hosted Mule runtime (called a CloudHub worker) in the cloud and started**
- When you make changes to the application in flow designer and are ready to test it
 - You need to run the application again – which updates the application deployed to the worker



MuleSoft-hosted
runtime

- **A worker is a dedicated instance of Mule that runs an app**
- Each worker
 - Runs in a separate container from every other application
 - Is deployed and monitored independently
 - Runs in a specific worker cloud in a region of the world
- Workers can have a different memory capacity and processing power
 - Apps can be scaled vertically by changing the worker size
 - Apps can be scaled horizontally by adding multiple workers
- There are workers in different environments
 - Design (for flow designer apps only), Sandbox, Production..
 - Apps can be promoted from one environment to another

Worker size
0.1 vCores
0.1 vCores 500 MB memory
0.2 vCores 1 GB memory
1 vCore 1.5 GB memory
2 vCores 3.5 GB memory

Walkthrough 2-2: Create a Mule application with flow designer



- Create a new Mule application project in Design Center
- Create an HTTP trigger for a flow in the application
- Add a Logger component
- Run and test the application
- View application information in Runtime Manager

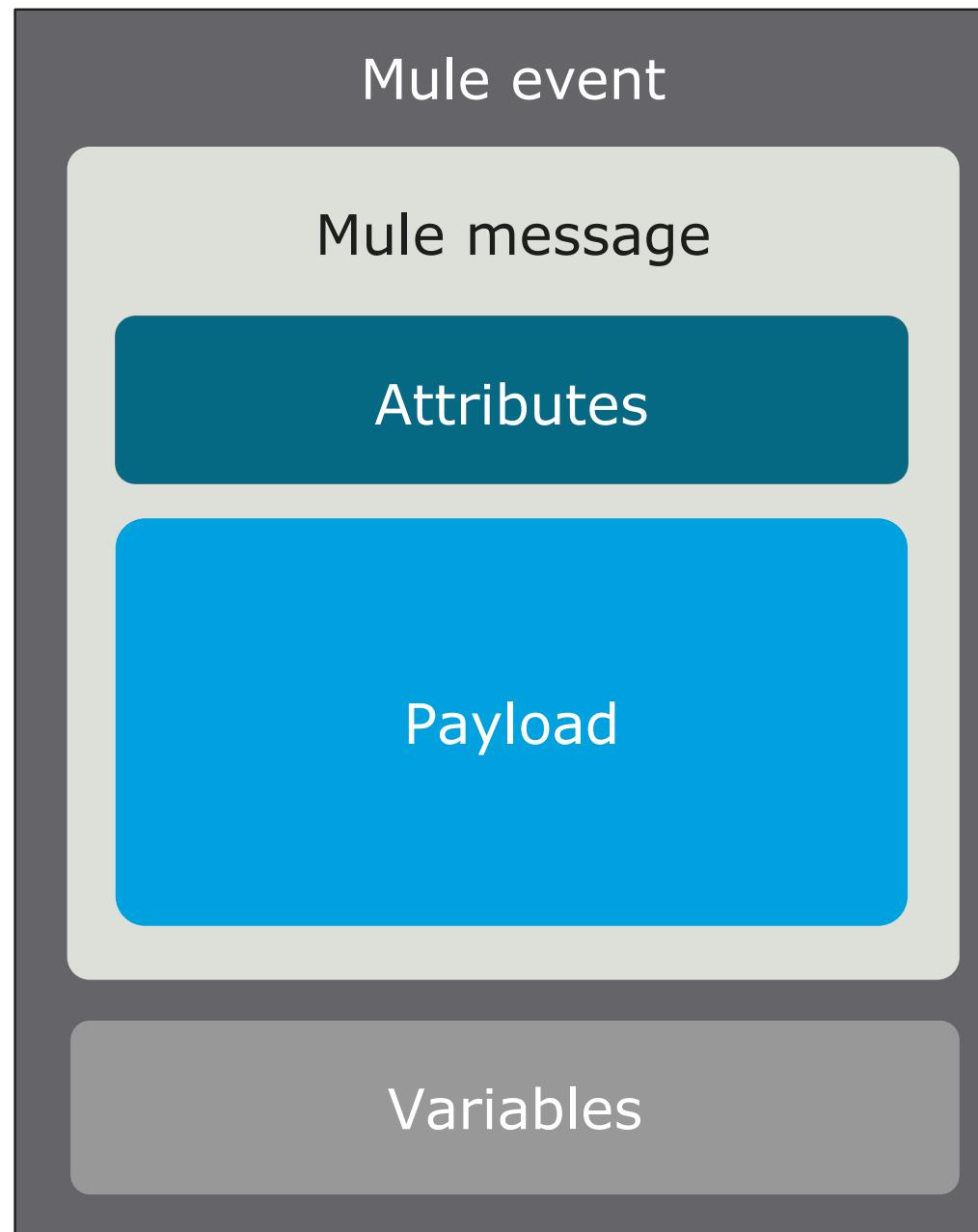
The screenshot shows the Mule Design Center interface. The top navigation bar indicates the project is 'Running' and has been 'Saved 11 minutes ago'. The left sidebar shows the 'Project' structure with 'Flows' expanded, showing a single flow named 'Get flights'. The main workspace displays a flow diagram with two components: 'HTTP Listener flights' and 'Logger INFO'. An arrow points from the 'HTTP Listener' component to the 'Logger' component. Below the workspace is a 'Logs' panel showing the following log entries:

```
INFO 11:10:16 Skipping the initialization of the mule.agent.tracking.handler.splunk Internal Handler because it's disabled.  
INFO 11:10:16 Skipping the initialization of the tracking.notification.internal.message.handler Internal Handler because it's disabled.  
INFO 11:10:16 Skipping the initialization of the mule.agent.tracking.handler.log Internal Handler because it's disabled.  
INFO 11:10:16 Initializing the mule.agent.tracking.handler.cloudhub.event ...  
INFO 11:10:16 mule.agent.tracking.handler.cloudhub.event initialized successfully.  
INFO 11:10:16 test  
INFO 11:12:13 test
```

Accessing, querying, and transforming data



Accessing and modifying Mule 4 event data



- The data that passes through flows in the app
- Metadata contained in the message header
- The core info of the message - the data the app processes
- Metadata for the Mule event - can be defined and referenced in the app processing the event

Transforming data with DataWeave



- DataWeave 2.0 is the expression language for Mule to access, query, and transform Mule 4 event data
- A JSON-like language that's built just for data query and transformation use cases
 - Full-featured and fully native framework
- Fully integrated with flow designer (and Anypoint Studio)
 - Graphical interface with payload-aware development



The Transform component

- Has input, output, and preview sections with both drag-and-drop and script editors

Select a component

All ▾

-  Salesforce
-  ServiceNow
-  Transform
-  Try
-  Validation

Transform trash refresh

Configuration Input Output X

Input	Output payload	Preview
<input checked="" type="checkbox"/> Payload <code>Array<Object></code> <ul style="list-style-type: none">> <code>plane Object?</code><ul style="list-style-type: none"><code>code String?</code><code>price Number?</code><code>origin String?</code><code>destination String?</code><code>ID Number?</code><code>departureDate String?</code><code>emptySeats Number?</code><code>Attributes <code>Void</code></code><code>Variables <code>Object</code></code>	<input checked="" type="checkbox"/> <code>Payload <code>Array<Object></code> (Flights)</code> <ul style="list-style-type: none"><code>f(x) ○ airline String?</code><code>flightCode String?</code><code>fromAirportCode String?</code><code>toAirportCode String?</code><code>departureDate String?</code><code>emptySeats Number?</code><code>totalSeats Number?</code><code>price Number?</code><code>planeType String?</code>	<pre>1 { 2 "flightCode": "ER38sd", 3 "fromAirportCode": "MUA", 4 "toAirportCode": "SFO", 5 "departureDate": "2016/03/20", 6 "emptySeats": 0, 7 "totalSeats": 150, 8 "price": 400, 9 "planeType": "Boeing 737", 10 "airline": "American" 11 } 12 { 13 "flightCode": "ER45if", 14 "fromAirportCode": "MUA", 15 "toAirportCode": "LAX", 16 "departureDate": "2016/02/11", 17 "emptySeats": 52, 18 "totalSeats": 300, 19 "price": 345.99, 20 "planeType": "Boeing 777", 21 "airline": "American" 22 } 23 } 24 }</pre>

Actions for: `(root)/#/payload`

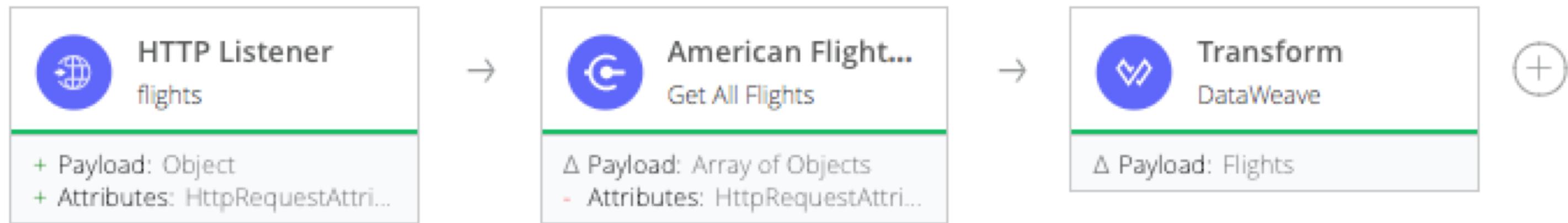
checkmark trash refresh copy undo redo search

Sample data Script Mappings

Walkthrough 2-3: Create an integration application with flow designer that consumes an API



- Examine Mule event data for calls to an application
- Use the American Flights API in Anypoint Exchange to get all flights
- Transform data returned from an API to another format



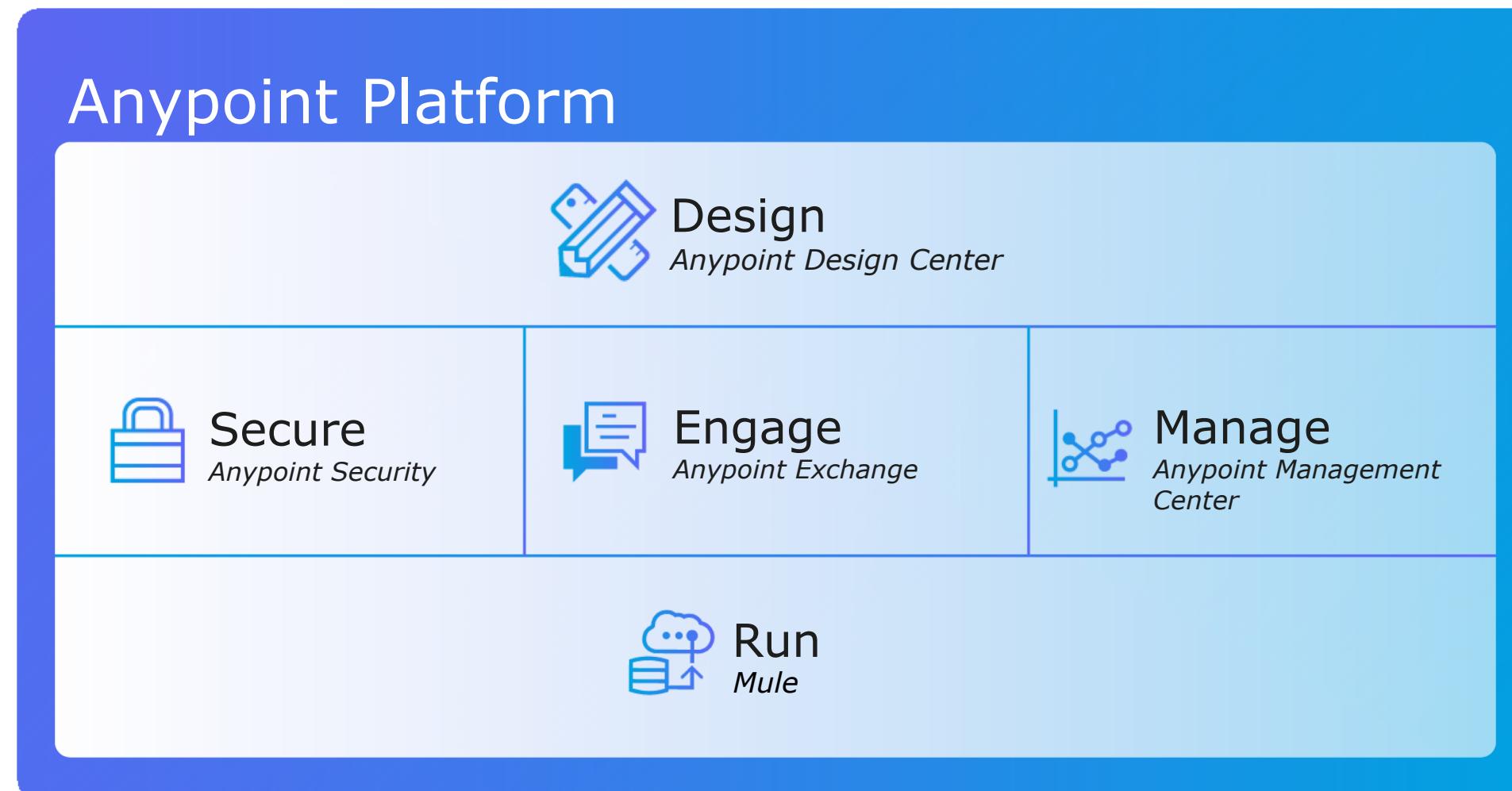
Summary



Summary: Anypoint Platform



- **Anypoint Platform** is a unified, hybrid integration platform that creates a seamless **application network** of apps, data, and devices with **API-led connectivity**



- 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** to build integration applications
 - These are Mule 4 applications that are deployed to a Mule runtime
 - To learn more, take the 1-day *Anypoint Platform: Flow Design* course
- **Mule runtimes** can be MuleSoft-hosted in the cloud (CloudHub) or customer-hosted in the cloud or on-prem
- **DataWeave 2.0** is the expression language for Mule to access, query, and transform Mule 4 event data