



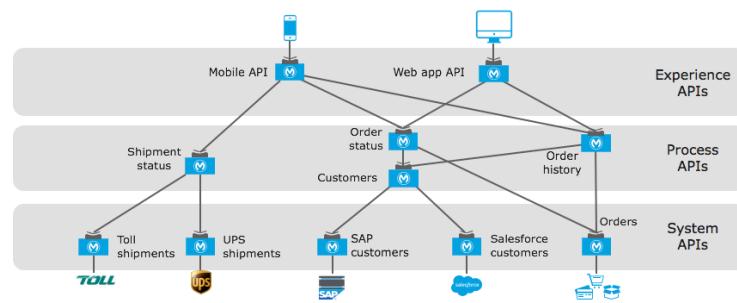
Module 1: Introducing Application Networks and API-Led Connectivity



At the end of this module, you should be able



- Explain what an application network is and its benefits
- Describe how to build an application network using API-led connectivity
- Explain what web services and APIs are
- Explore API directories and references
- Make calls to secure and unsecured APIs

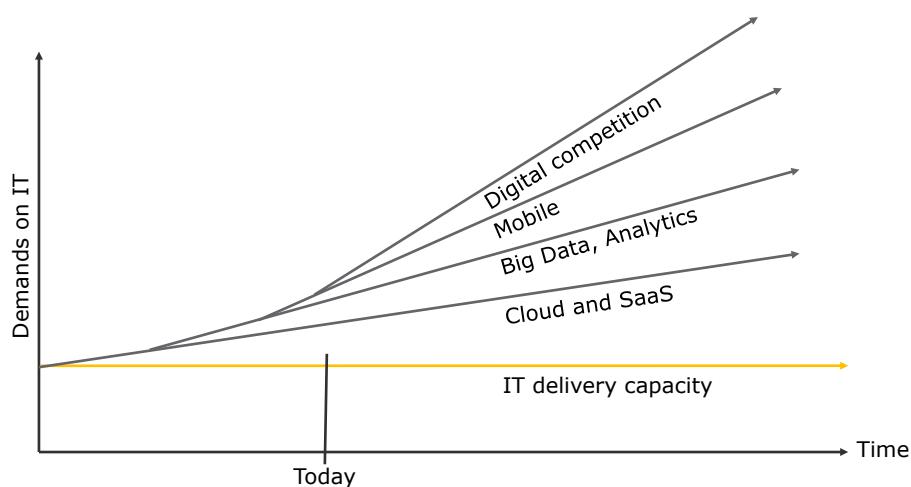


All contents © MuleSoft Inc.

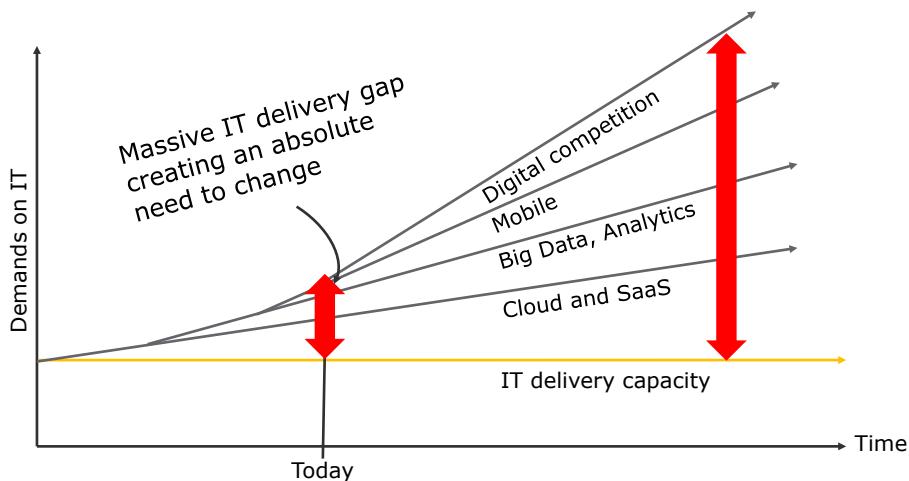
Identifying the problems faced by IT today



Biggest challenge: IT cannot go fast enough



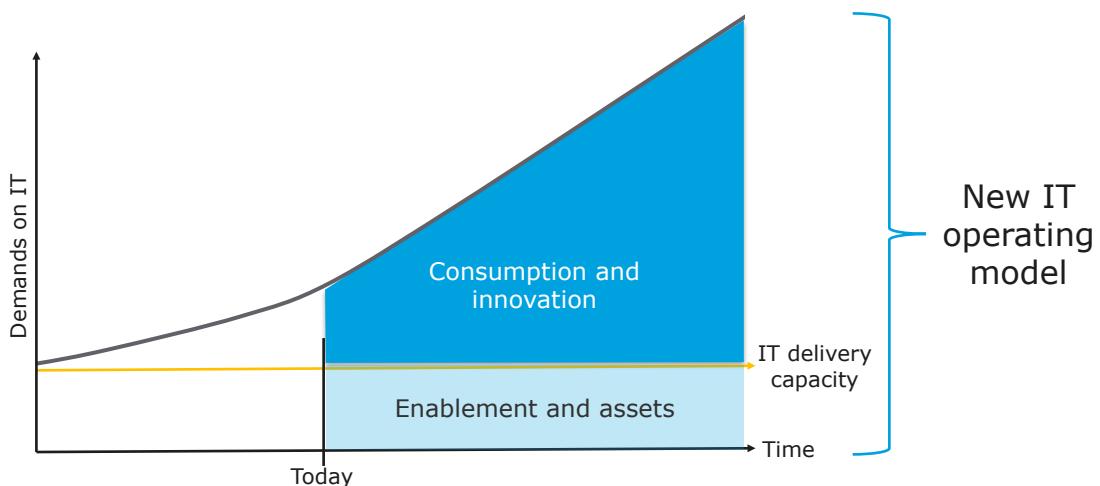
Digital pressures create a widening IT delivery gap



All contents © MuleSoft Inc.

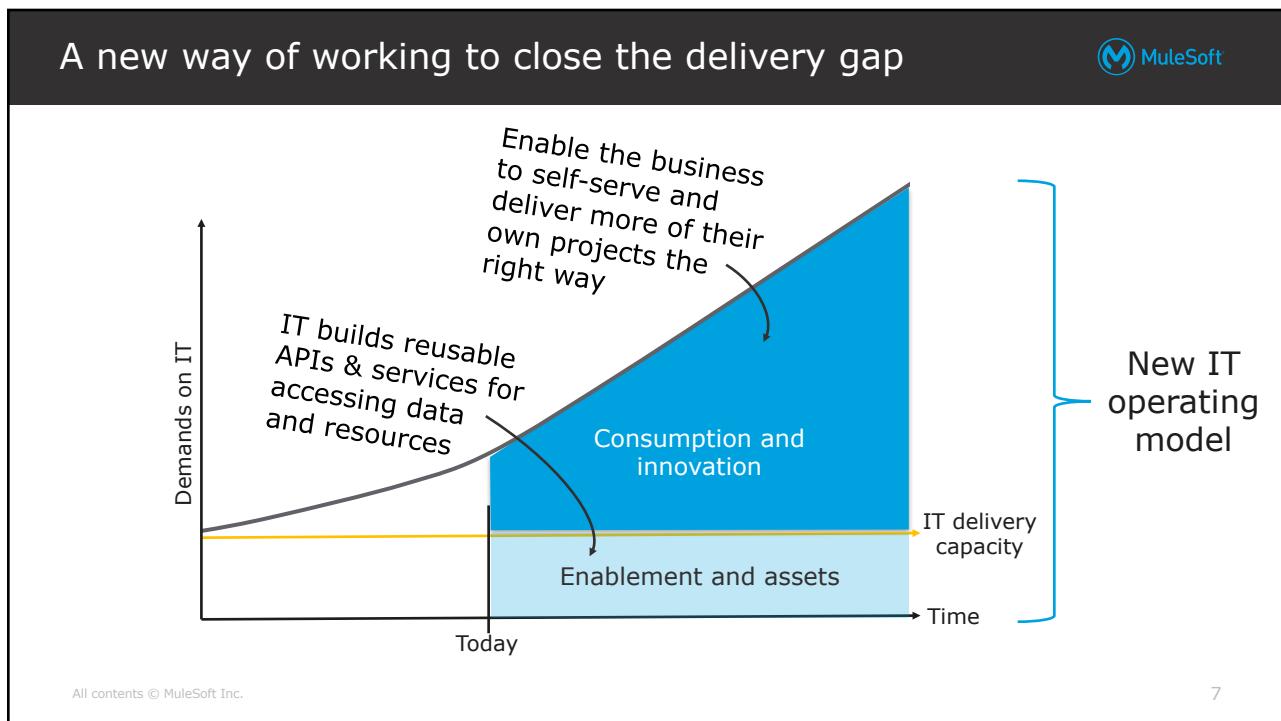
5

A new way of working to close the delivery gap



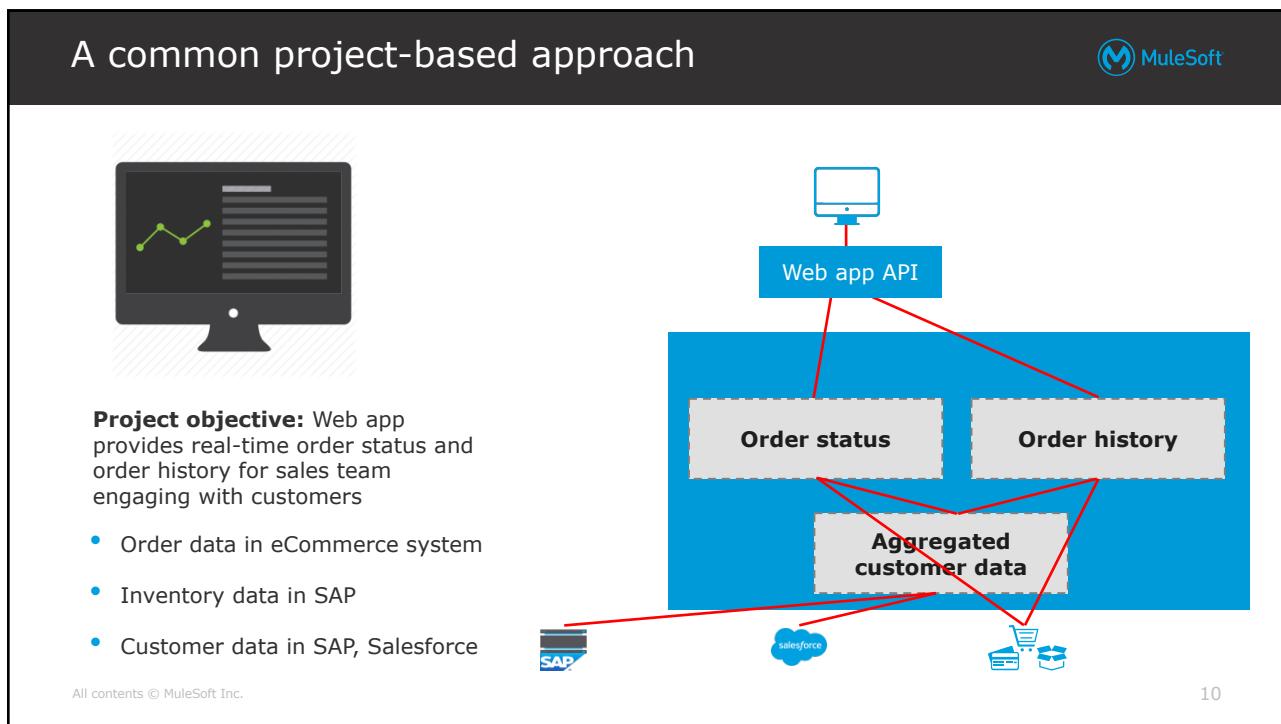
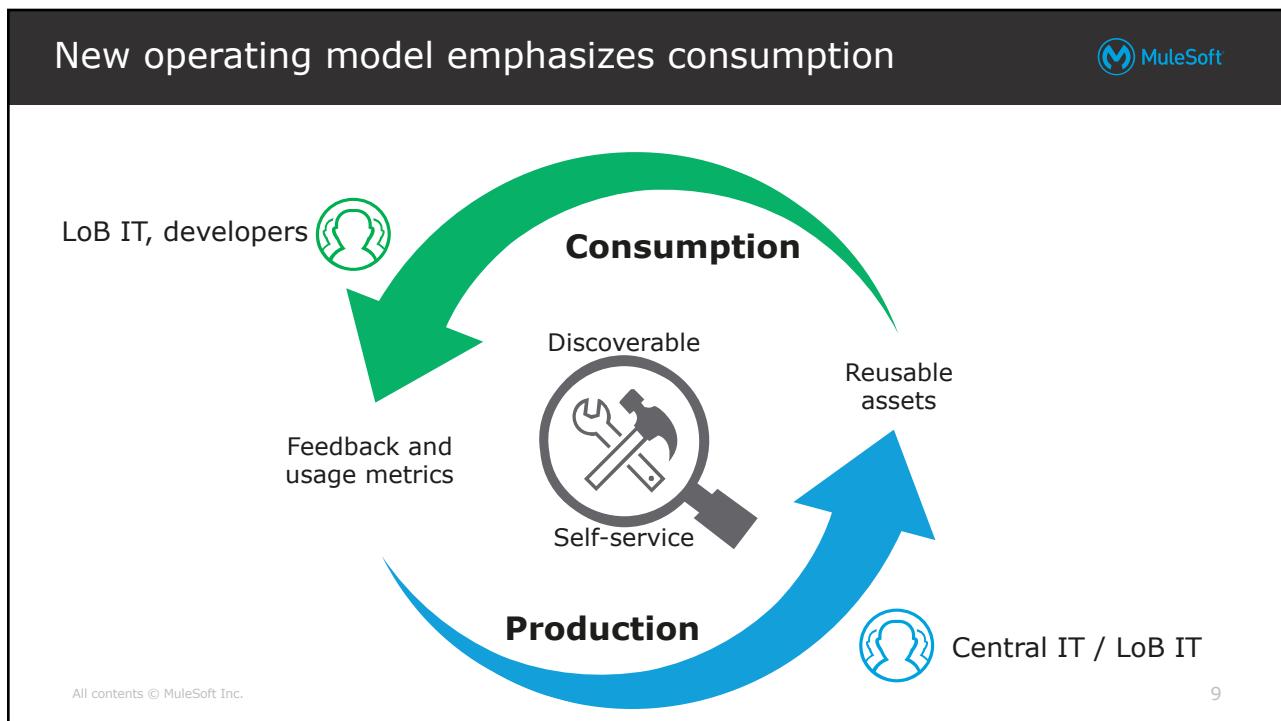
All contents © MuleSoft Inc.

6



Introducing a new IT operating model

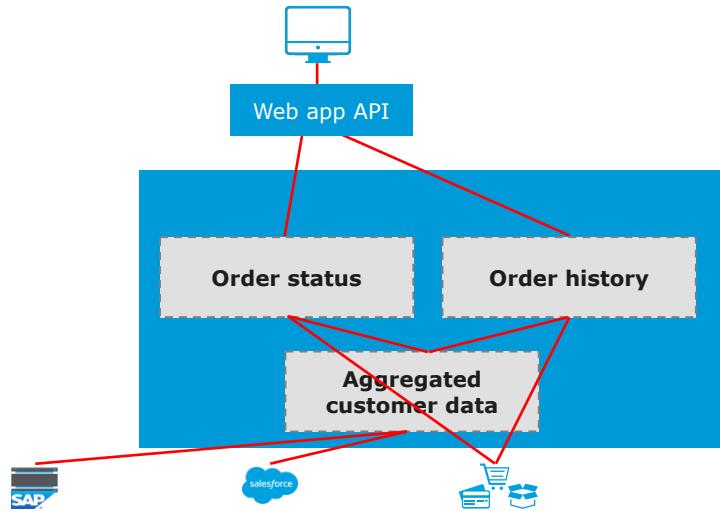




A common project-based approach



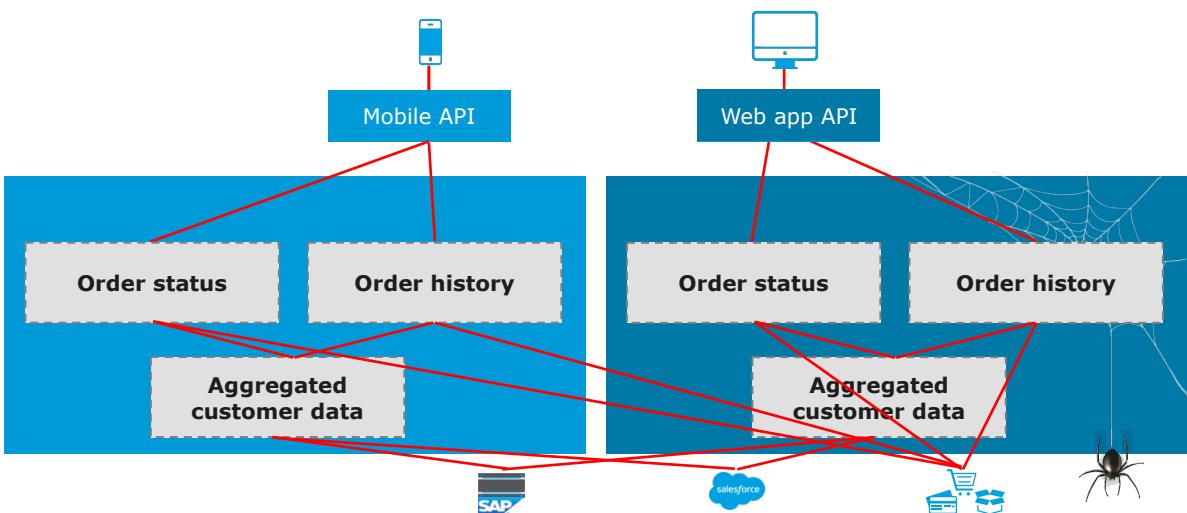
- ✓ On time and within budget
- ✗ Limited opportunity for reuse
- ✗ Tight coupling = brittleness
- ✗ Difficult to govern
- ? Meets business requirements



All contents © MuleSoft Inc.

11

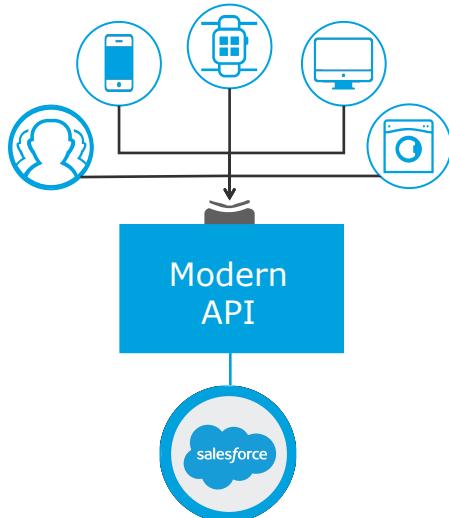
6 months later...



All contents © MuleSoft Inc.

12

Modern API: The core enabler of a new operating model

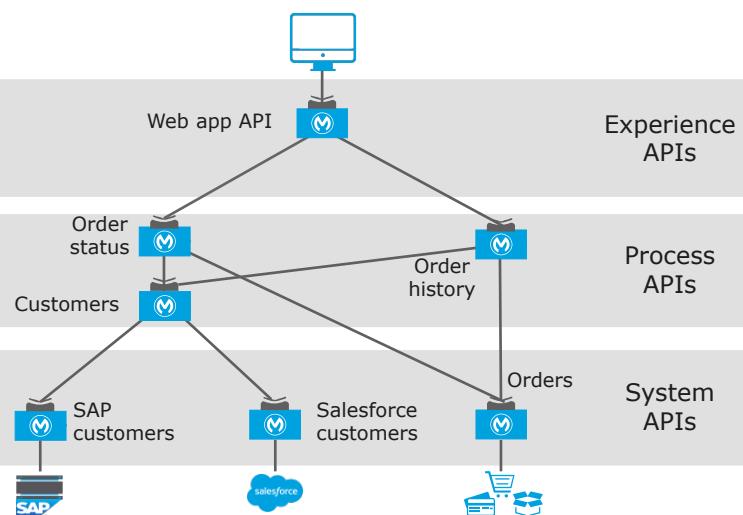


- Discoverable and accessible through self-service
- Productized and designed for ease of consumption
- Easily managed for security, scalability, and performance

All contents © MuleSoft Inc.

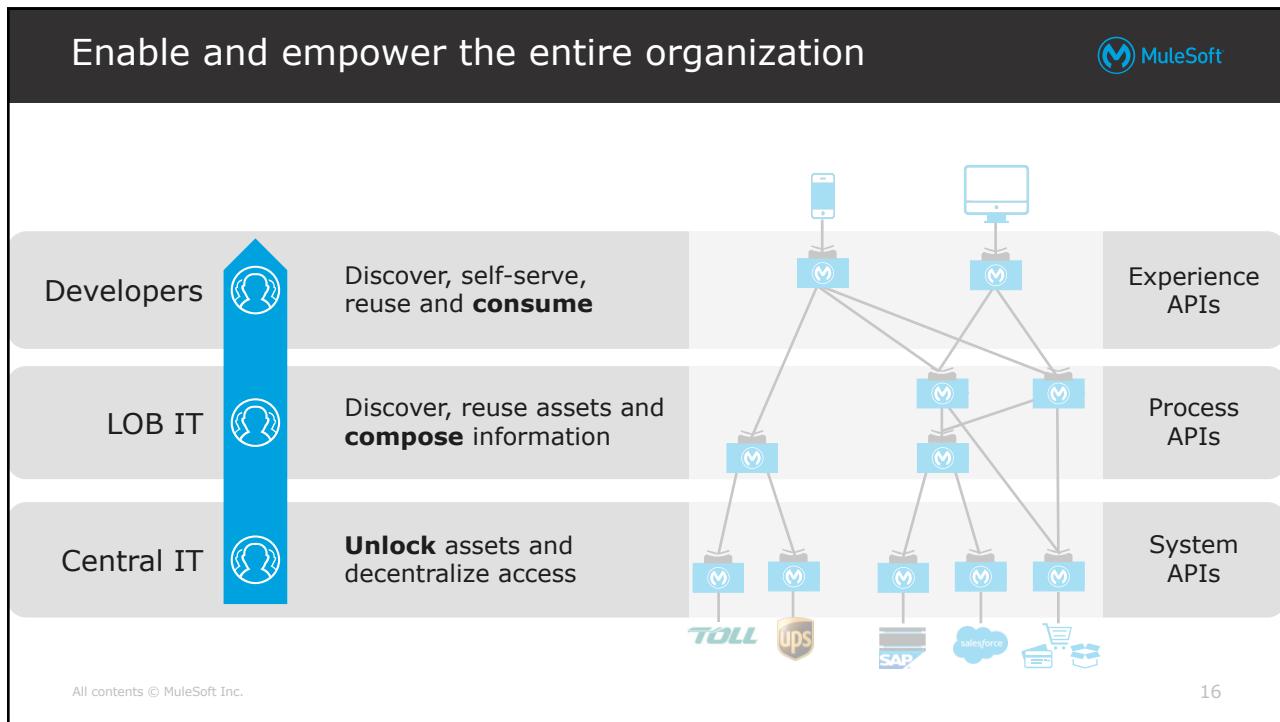
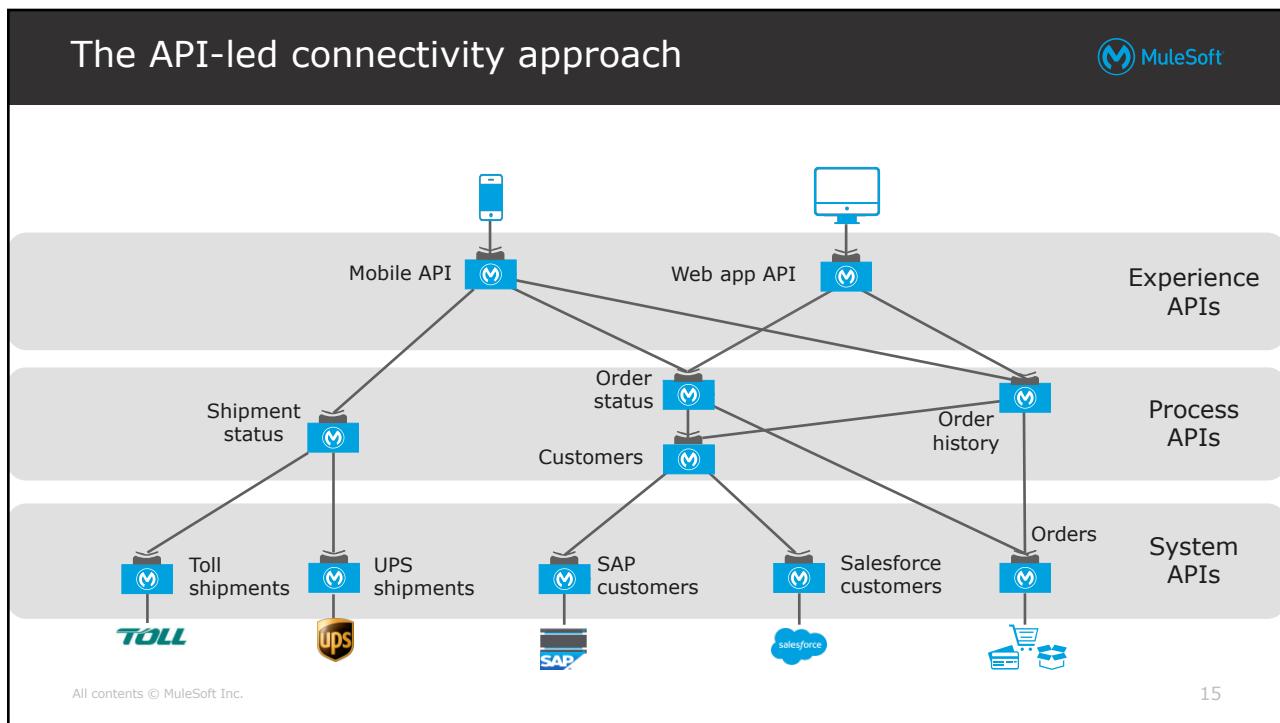
13

The API-led connectivity approach



All contents © MuleSoft Inc.

14



Drive outcomes with API-led connectivity

MuleSoft

On time and within budget

Drives reuse vs build

Designs in readiness for change

Builds in governance, compliance, security, and scalability

Meets the needs of your business

All contents © MuleSoft Inc.

17

C4E: Organizing differently to drive API-led connectivity

MuleSoft

- C4E is a cross functional team
- C4E ensures that assets are
 - Productized and published
 - Consumable
 - Consumed broadly
 - Fully leveraged
- Success of C4E measured on asset consumption

All contents © MuleSoft Inc.

18

Achieving an application network



Application landscape

MuleSoft



Every project adds value to the application network

MuleSoft

Project 1 – API-led

All contents © MuleSoft Inc.

21

Every project adds value to the application network

MuleSoft

Project 1 – API-led

C4E

Self-serve assets on the application network

- Order status
- Shipments
- Orders
- Shipment status
- Customers

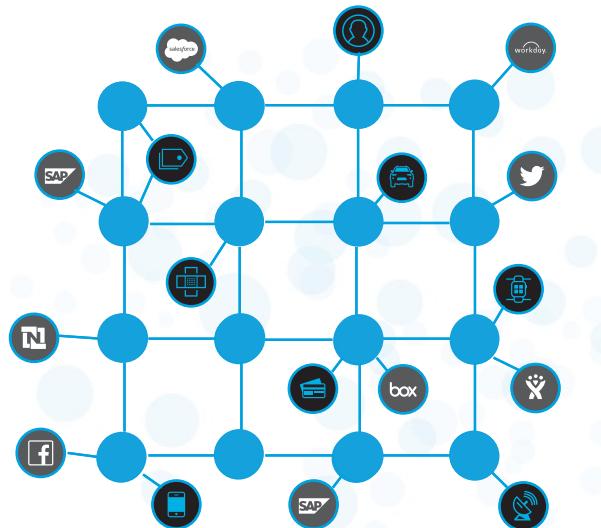
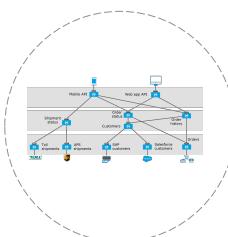
All contents © MuleSoft Inc.

22

Every project adds value to the application network

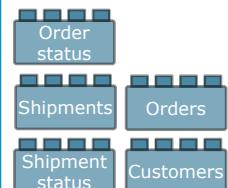


Project 1 – API-led



C4E

Self-serve assets on the application network



All contents © MuleSoft Inc.

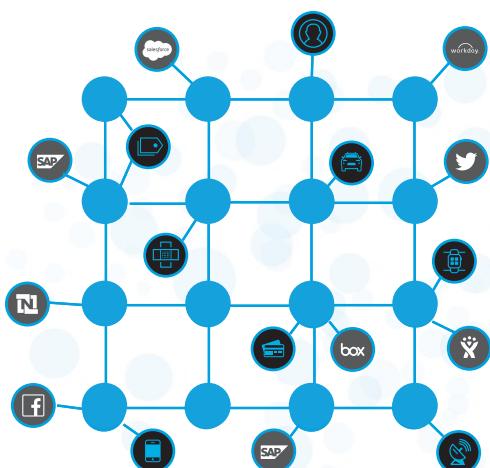
23

Speed. Agility. Innovation.



An application network

- Emerges bottoms-up via self-service
- Provides visibility, security and governability at every API node
- Is recomposable: it bends, not breaks – **built for change**



All contents © MuleSoft Inc.

24

Deconstructing APIs



What exactly is an API?



- An **API** is an **A**pplication **P**rogramming **I**nterface
- It provides the information for how to communicate with a software component, defining the
 - Operations (what to call)
 - Inputs (what to send with a call)
 - Outputs (what you get back from a call)
 - Underlying data types
- It defines functionalities independent of implementations
 - You can change what's going on behind the scenes without changing how people call it

What do people mean when they say API?



They could be referring to a number of things...

1. An API interface definition file

- Defines what you can call, what you send it, and what you get back

2. A web service

- The actual API implementation you can make calls to or the interface of that API implementation

3. An API proxy

- An application that controls access to a web service, restricting access and usage through the use of an API gateway

All contents © MuleSoft Inc.

27

Reviewing web services



What is a web service?



- Different software systems often need to exchange data with each other
 - Bridging protocols, platforms, programming languages, and hardware architectures
- A **web service** is a method of communication that allows two software systems to exchange data over the internet
- Systems interact with the web service in a manner prescribed by some defined rules of communication
 - How one system can request data from another system, what parameters are required, the structure of the return data, and more

The parts of a web service



- **The web service API**
 - Describes how you interact with the web service
 - It may or may not (though it should!) be explicitly defined in a file
 - It could be any sort of text in any type of file but ideally should implement some standard API description language (or specification)
- **The web service interface implementing the API**
 - Is the code providing the structure to the application so it implements the API
 - This may be combined with the actual implementation code
- **The web service implementation itself**
 - Is the actual code and application

Two main types of web services



- SOAP web services
 - Traditional, more complex type
 - The communication rules are defined in an XML-based WSDL (Web Services Description Language) file
- **RESTful web services**
 - Recent, simpler type based on representational state transfer (REST) based communications
 - Use the existing HTTP communication protocol

All contents © MuleSoft Inc.

31

Reviewing RESTful web services



RESTful web services



- REST stands for **R**epresentational **S**tate **T**ransfer
 - An architectural style where clients and servers exchange representations of resources using standard HTTP protocol
- Other systems interact with the web service using the HTTP protocol
 - The HTTP request method indicates which operation should be performed on the object identified by the URL
 - GET, POST, DELETE, PUT, PATCH



All contents © MuleSoft Inc.

33

RESTful web service requests



- Data and resources are represented using **URIs**
- Resources are accessed or changed using a fixed set of operations
 - **GET** retrieves the current state of a resource in some representation (usually JSON or XML)
 - **POST** creates a new resource
 - **DELETE** deletes a resource
 - **PUT** replaces a resource completely
 - If the resource doesn't exist, a new one is created
 - **PATCH** partially updates a resource
 - Just submitted data



All contents © MuleSoft Inc.

34

Example RESTful web service calls



- (GET)/companies
- (GET)/companies?country=France
- (GET)/companies/3
- (POST)/companies with JSON/XML in HTTP body
- (DELETE)/companies/3
- (PUT)/companies/3 with JSON/XML in HTTP body

All contents © MuleSoft Inc.

35

Example RESTful web service response



- JSON (JavaScript Object Notation)
 - A lightweight data-interchange format (without a lot of extra XML markup)
 - Human-readable results (usually JSON or XML)
 - Supports collections and maps

```

{
  "flights": [
    {
      "airlineName": "United",
      "price": 400,
      "departureDate": "2015/03/20",
      "planeType": "Boeing 737",
      "origin": "MUA",
      "code": "ER38sd",
      "emptySeats": 0,
      "destination": "SFO"
    },
    {
      "airlineName": "United",
      "price": 945,
      "departureDate": "2015/09/11",
      "planeType": "Boeing 757",
      "origin": "MUA",
      "code": "ER39rk",
      "emptySeats": 54,
      "destination": "SFO"
    },
    {
      "airlineName": "United",
      "price": 954,
      "departureDate": "2015/02/12",
      "planeType": "Boeing 777",
      "origin": "MUA",
      "code": "ER39rj",
      "emptySeats": 23,
      "destination": "SFO"
    }
  ]
}
  
```

All con

36

Walkthrough 1-1: Explore an API directory and an API reference



- Browse the ProgrammableWeb API directory
- Explore the API reference for the Twitter API

Developer Use cases Products Docs More Apply Sign In

Search all documentation...

Post, retrieve and engage with Tweets

Basics

Accounts and users

Tweets

[Post, retrieve and engage with Tweets](#)

Get Tweet timelines

Curate a collection of Tweets

Optimize Tweets with Cards

Search Tweets

Filter realtime Tweets

Sample realtime Tweets

Get batch historical Tweets

Overview Guides API Reference

The following API endpoints can be used to programmatically create, retrieve and delete Tweets, Retweets and Likes:

Tweets

- POST statuses/update
- POST statuses/unretweet/:id
- POST statuses/destroy/:id
- GET statuses/show/:id

Retweets

- POST statuses/retweet/:id
- POST favorites/create/:id
- POST favorites/destroy/:id
- GET favorites/list

Likes (formerly favorites)

37

Calling RESTful web services



Calling RESTful web services



- To call web services, you need to write code or have a tool to make the HTTP requests
 - Need to be able to specify the HTTP method, request headers, and request body
 - Postman, A cURL command-line utility, Advanced Rest Client (for Chrome) +

```

1+ [
2+   {
3+     "ID": 1,
4+     "code": "rree0001",
5+     "price": 541,
6+     "departureDate": "2016-01-20T00:00:00",
7+     "origin": "HNL",
8+     "destination": "LAX",
9+     "emptySeats": 0,
10+    "plane": {
11+      "type": "Boeing 787",
12+
13+    }
14+
15+  }
16+
17+ ]
  
```

All contents © MuleSoft Inc.

33

Making calls to RESTful APIs



- **Unsecured APIs**
 - The API may be public and require no authentication

- **Secured APIs**
 - The API may be secured and require authentication
 - You may need to provide credentials and/or a token
 - Often a proxy is created to govern access to an API
 - We will call and then later create an API secured by credentials
 - You can also secure an API with other authentication protocols
 - OAuth, SAML, JWT, and more

All contents © MuleSoft Inc.

40

Getting responses from web service calls



- RESTful web services return an HTTP status code with the response
- The status code provides client feedback for the outcome of the operation (succeeded, failed, updated)
 - A good API should return status codes that align with the HTTP spec

Response

200	Type application/atom+xml	
401	Schema	Examples
	<input checked="" type="checkbox"/> View <pre><?xml version='1.0' encoding='UTF-8'?> <feed xmlns='http://www.w3.org/2005/Atom' xmlns:openSearch='http://a9.com/-/spec/opensearch/1.1/' xmlns:gContact='http://schemas.google.com/contact/2008' xmlns:batch='http://schemas.google.com/gdata/batch' xmlns:gd='http://schemas.google.com/g/2005' gd:etag='feedEtag'></pre>	

All contents © MuleSoft Inc.

41

Common HTTP status codes



Code	Definition	Returned by
200	OK – The request succeeded.	GET, DELETE, PATCH, PUT
201	Created – A new resource or object in a collection.	POST
304	Not modified – Nothing was modified by the request.	PATCH, PUT
400	Bad request – The request could not be performed by the server due to bad syntax or other reason in request.	All
401	Unauthorized – Authorization credentials are required or user does not have access to the resource/method they are requesting.	All
404	Resource not found – The URI is not recognized by the server.	All
500	Server error – Generic something went wrong on the server side.	All

All contents © MuleSoft Inc.

42

Walkthrough 1-2: Make calls to an API



- Use Postman to make calls to an unsecured API (an implementation)
- Make GET, DELETE, POST, and PUT calls
- Use Postman to make calls to a secured API (an API proxy)

The screenshot shows the Postman interface with the following details:

- Method:** POST
- URL:** http://training-american-ws.cloudhub.io/api/flights/
- Body (JSON):**

```

1 - [
2   "code": "Q0574",
3   "price": 399,
4   "departureDate": "2016/12/20",
5   "origin": "ORD",
6   "destination": "SFO",
7   "emptySeats": 200,
8   "plane": {"type": "Boeing 747", "totalSeats": 400}
9 ]

```

- Response Status:** 201 Created
- Response Time:** 132 ms
- Response Size:** 221 B
- Response Body (Pretty JSON):**

```

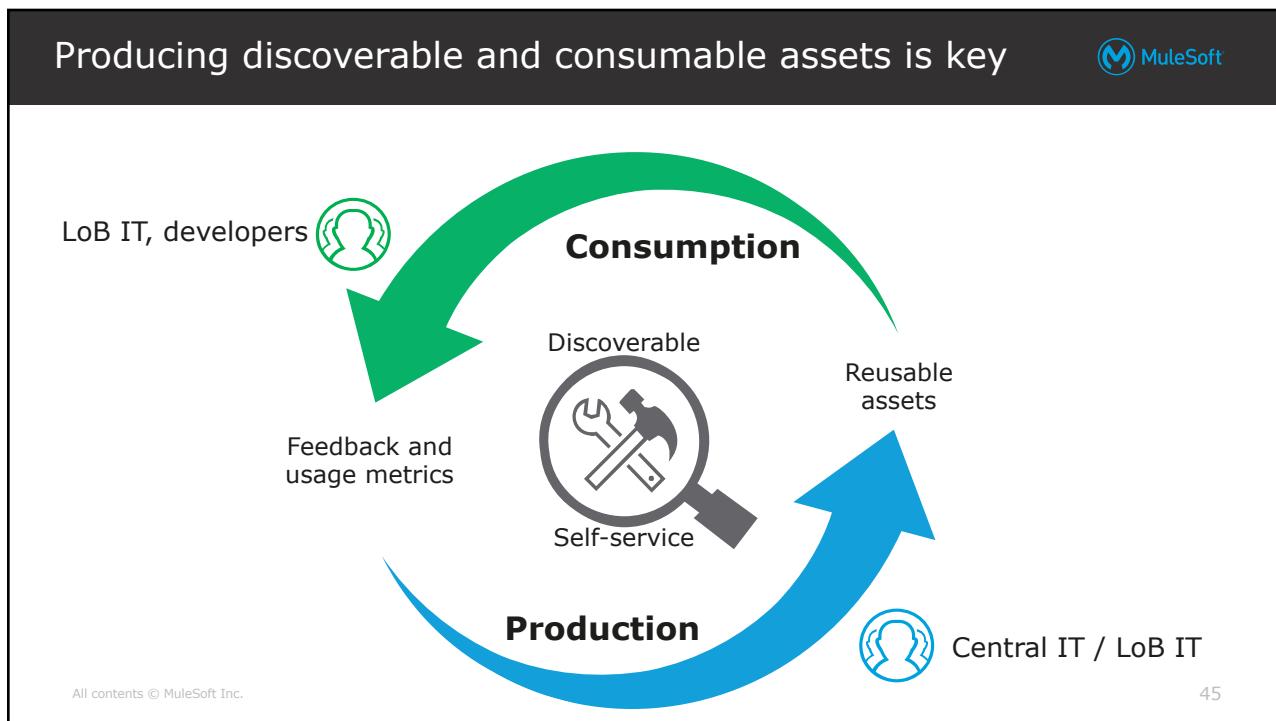
1 - {
2   "message": "Flight added (but not really)"
3 }

```

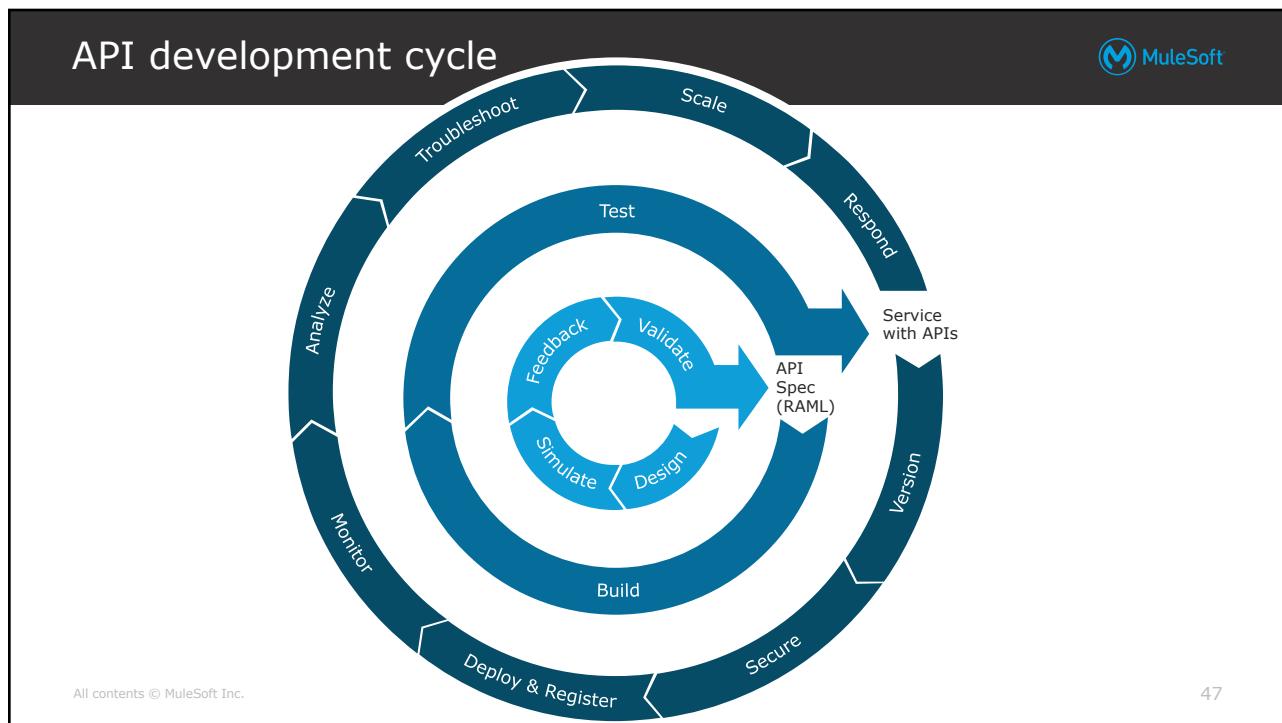
43

Successfully creating application networks using API-led connectivity





- Designing for API success
- Create APIs that developers **can find** and **want to use** and share with others
 - Design the API for the business use cases it will fulfill, not to model the backend services or applications they expose
 - Focus on performance of client applications and user experience
 - Take an **API design-first approach!**
 - **Get API design right** before investing in building it
 - Define it iteratively getting feedback from developers on its usability and functionality along the way
 - Building the implementation of an API is time consuming and expensive to undo
- All contents © MuleSoft Inc.
- 46



Summary



Summary



- Companies today need to **rapidly adopt and develop** new technologies in order to stay relevant to customers & keep competitive
- IT needs to be able to rapidly integrate resources and make them **available for consumption**
 - An **API-led connectivity** approach can help achieve this
- To drive API-led connectivity, create a **C4E** (Center for Enablement)
 - A cross-functional team to ensure assets across the organization are productized, published, and widely consumed
- **An application network** is a network of applications, data, and devices connected with APIs to make them pluggable and to create reusable services

All contents © MuleSoft Inc.

49

Summary



- A **web service** is a method of communication that allows two software systems to exchange data over the internet
- An **API** is an application programming interface that provides info for how to communicate with a software component
- The **term API** is often used to refer to any part of RESTful web service
 - The web service API (definition file)
 - The web service interface implementing the API
 - The web service implementation itself
 - A proxy for the web service to control access to it
- **RESTful** web services use standard HTTP protocol and are easy to use
 - The HTTP request method indicates which operation should be performed on the object identified by the URL

All content

50



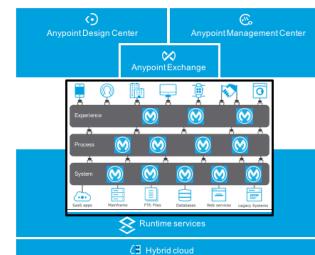
Module 2: Introducing Anypoint Platform



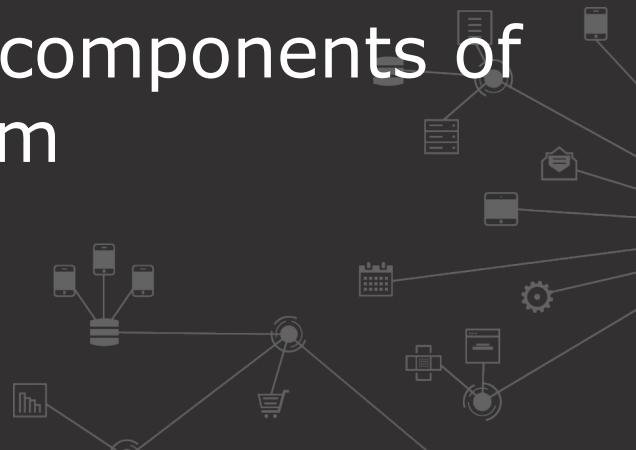
At the end of this module, you should be able to



- Identify all the components of Anypoint Platform
- 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



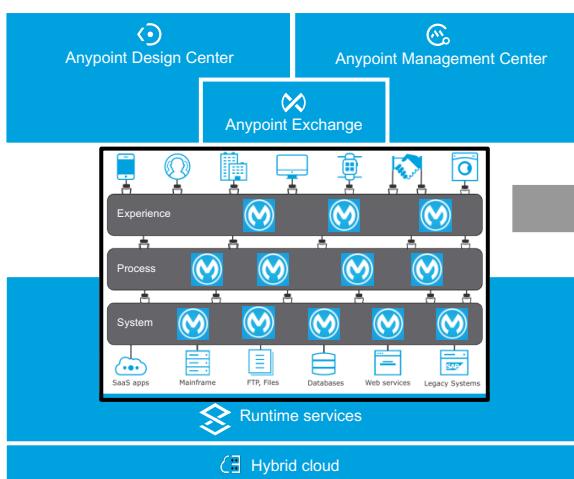
Introducing the components of Anypoint Platform



Anypoint Platform uniquely enables the building of an application network



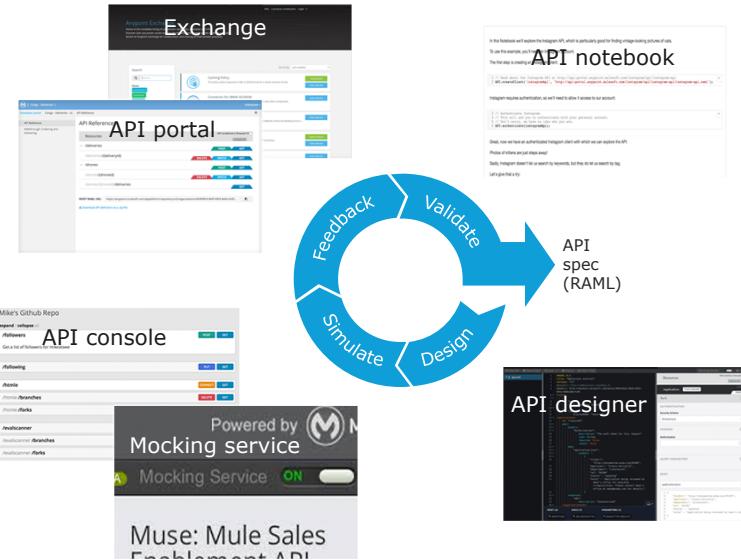
Anypoint Platform



Application network

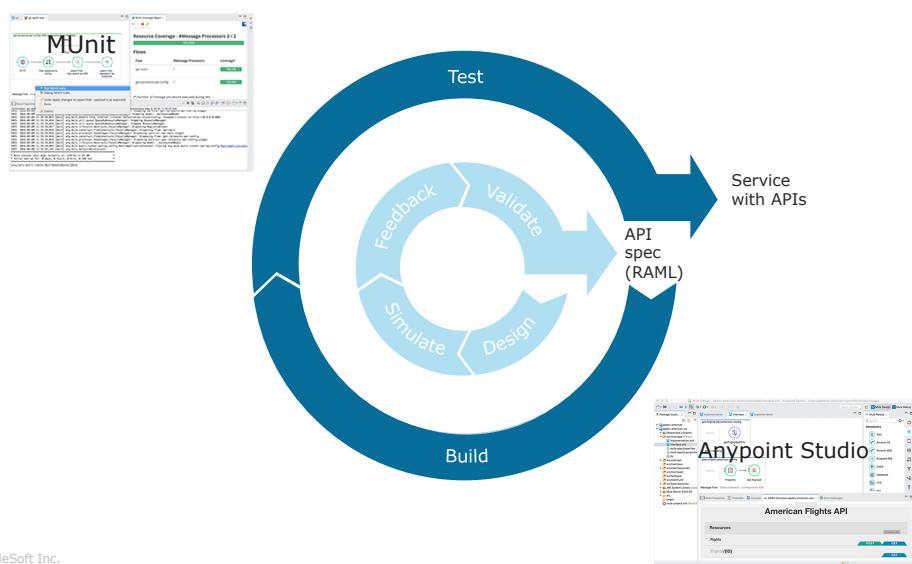


API development cycle: API specification

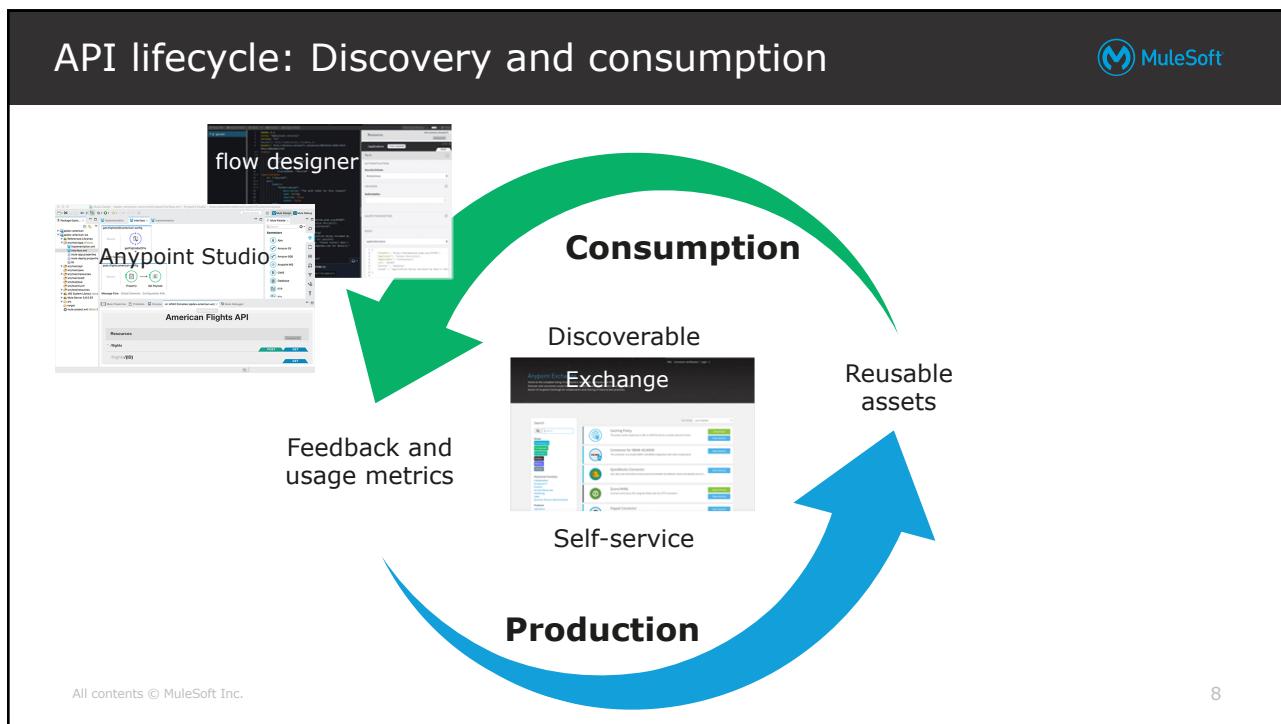
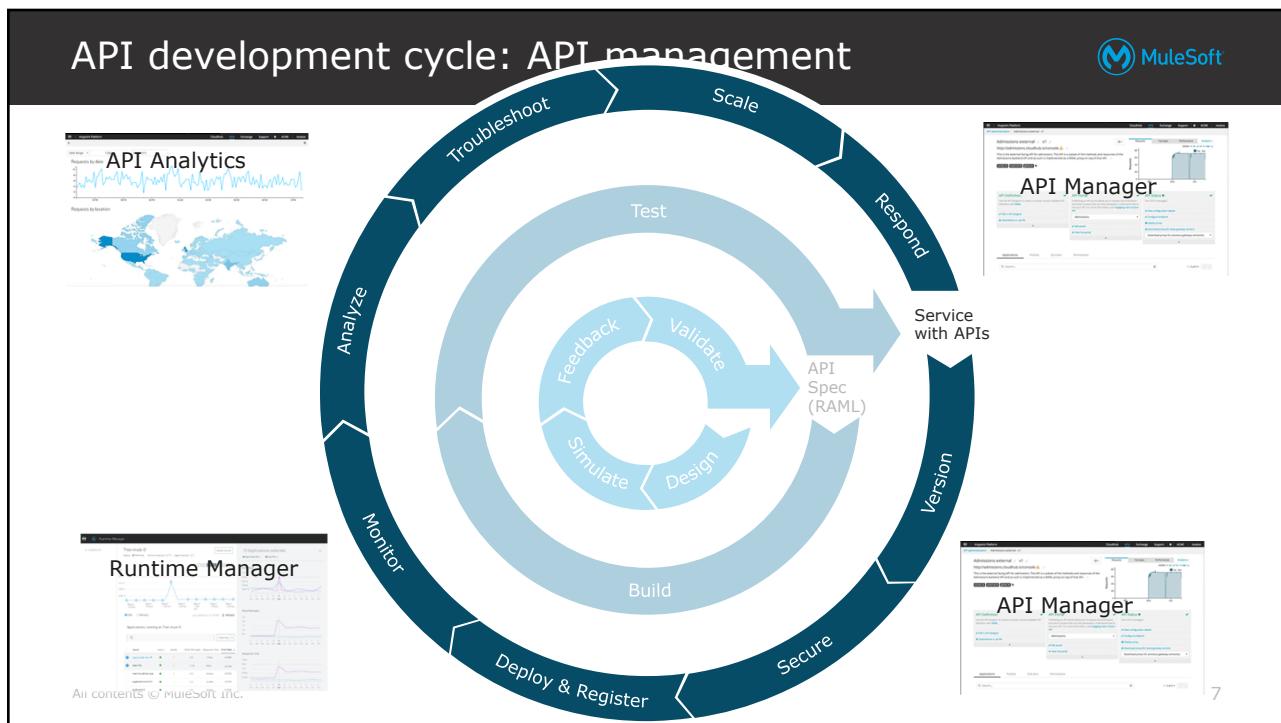


5

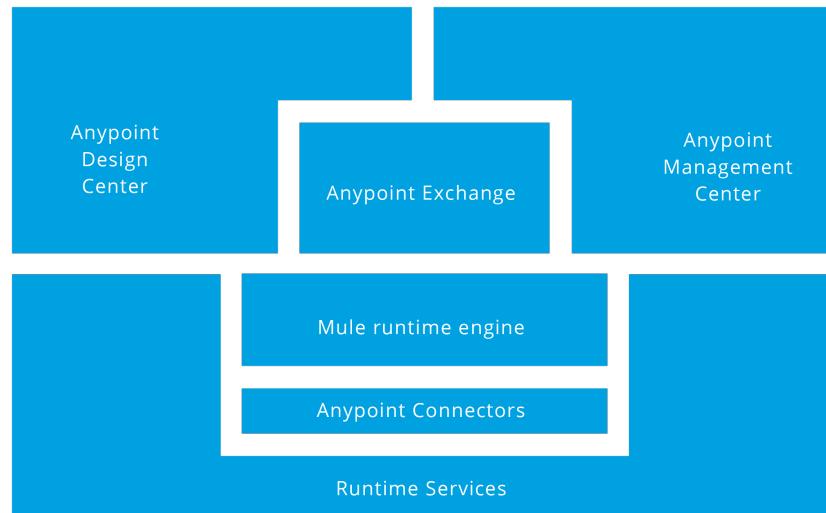
API development cycle: API implementation



6



Anypoint Platform: The components



All contents © MuleSoft Inc.

9

Anypoint Platform



- **A unified, highly productive, hybrid integration platform that creates a seamless application network of apps, data, and devices with API-led connectivity**

- A collection of runtimes, frameworks, tools, and web applications
 - **Tools and frameworks** for building applications
 - **Mule runtime** for running applications and applying policies
 - MuleSoft-hosted in the cloud or customer-hosted (on-prem or in the cloud)
 - **Web application** 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

All contents © MuleSoft Inc.

10

Anypoint Platform: The web application



- MuleSoft-hosted in the cloud at anypoint.mulesoft.com
- Customer-hosted as part of Anypoint Platform Private Cloud Edition

11

Core functionality of the web application



API designer	Designing APIs
flow designer	Building integration apps that consume APIs
Exchange	Sharing, discovering, and exploring all the resources needed for your integration projects including creating public portals
API console, mocking service, API notebook	Testing and simulating APIs
Access Management	Managing users
Runtime Manager	Deploying apps to the cloud or on-prem Managing and monitoring applications
API Manager	Managing and monitoring APIs

All contents © MuleSoft Inc.

12

Introducing Anypoint Exchange



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



The screenshot shows the Anypoint Exchange web interface. The left sidebar has a navigation menu with 'All' selected, followed by 'MuleSoft', 'Training', 'My applications', and 'Public Portal'. The main content area is titled 'All assets' and shows a grid of four asset cards:

- American Flights API Connector** (Max Mule): A connector asset with a 5-star rating.
- REST API** (Max Mule): A connector asset with a 5-star rating.
- LDAP Connector** (MuleSoft): A module asset with a 5-star rating.
- Amazon RDS Connector** (MuleSoft): A module asset with a 5-star rating.

At the bottom right of the interface, there is a page number '14'.

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

All contents © MuleSoft Inc.

15

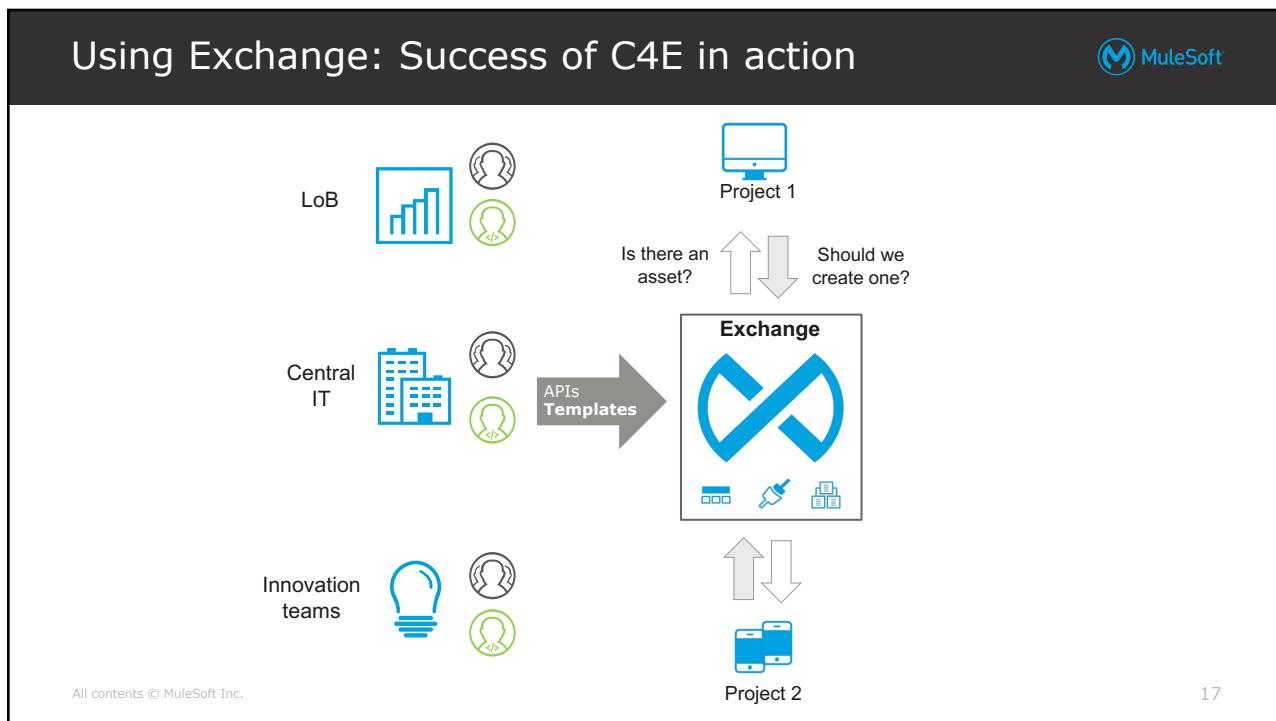
REST APIs and API portals in Anypoint Exchange



- **API portals** are automatically created for REST APIs added to Exchange
- An API portal has
 - Auto-generated **API documentation**
 - An **API console** that provides a Postman-like experience 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

All contents © MuleSoft Inc.

16



Walkthrough 2-1: Explore Anypoint Platform and Anypoint Exchange

The screenshot shows the Anypoint Platform interface, specifically the Exchange section. It displays the "Training: American Flights API" (version 1.0). The API summary shows a 5-star rating and 7 reviews. The "flights : Get all flights" endpoint is highlighted. The "Request" section shows a GET request to `http://training-american-ws.cloudhub.io/api/flights`. The "Parameters" section includes a "Query parameters" table with "destination" set to "LAX". The "Response" section shows a 200 OK status with a response time of 257.07 ms. The response body is a JSON array of flight records, with the first record partially visible.

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

All contents © MuleSoft Inc.

Building integration applications and APIs with Design Center



Design Center anatomy



Design Center

Get Started + Create American Flights App

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

Created with API designer

Created with flow designer

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	<ul style="list-style-type: none"> Anypoint Platform Development: Fundamentals Anypoint Platform Development: Advanced Anypoint Platform Development: DataWeave

All contents © MuleSoft Inc.

21

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

All contents © MuleSoft Inc.

22

Mule is the runtime engine of Anypoint Platform



- **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**

All contents © MuleSoft Inc.

23

Mule runtime editions and versions



- There are different **editions** of the Mule runtime
 - Community edition (CE): Open-source
 - Enterprise edition (EE): Hardened code line with support and additional capabilities
 - Support, additional connectors, batch, caching, security, templates, and more
- There are different **versions** of each Mule runtime
 - 3.7.X, 3.8.X, and more
- **Flow designer** uses Mule 4.0.X EE (an early access version of Mule 4)
- By default, the latest **Anypoint Studio** uses Mule 3.9 EE
 - You can install other versions and select which one to use

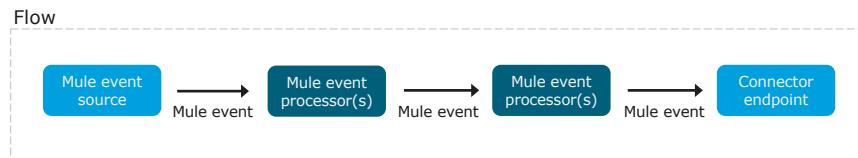
All contents © MuleSoft Inc.

24

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

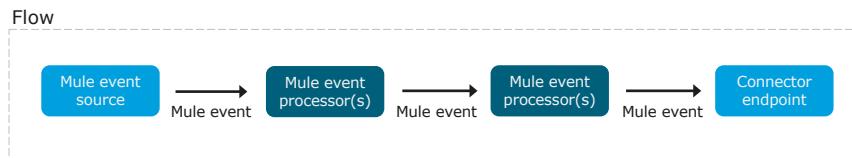
All contents © MuleSoft Inc.

25

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



All contents © MuleSoft Inc.

26

Creating integration applications with flow designer



Flow designer anatomy

MuleSoft

The screenshot shows the MuleSoft Anypoint Studio interface for creating integration applications:

- Project explorer:** Shows the project structure with a "Get flights" flow selected.
- Cards:** Displays the flow components and their connections:
 - An **HTTP Listener** component named "flights" receives requests.
 - An **American Flight...** component named "Get All Flights" processes the requests.
 - A **Transform** component named "DataWeave" transforms the payload.
- Logs:** Shows application logs with entries like:
 - SYSTEM 13:21:55 Worker(34.207.83.22): Your application has started successfully.
 - SYSTEM 13:21:55 Your application is started.
 - INFO 13:41:01 ++++++-----+-----+-----+-----+-----+-----+

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 redeploy and restart the application



All contents © MuleSoft Inc.

29

CloudHub workers



- **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

All contents © MuleSoft Inc.

Flow designer applications



- Are automatically deployed to the **Design** environment
 - Worker runtime version set to **Mule 4.0.X**
 - Flow designer apps can only be built using Mule 4, currently an early access version
 - Worker number is set to **1**
 - Worker size is set to **0.2**

The screenshot shows the MuleSoft Runtime Manager interface. On the left, there's a sidebar with options like DESIGN, Applications, Dashboard, Insight, Logs, Object Store, Queues, Schedules, All content, and Settings. The main area displays the application 'americanflightsapp-vqrf'. It shows the application file 'americanflightsapp-vqrf.jar', last updated on 2017-11-19 at 12:20:46PM, and the app URL 'americanflightsapp-vqrf.cloudhub.io'. Below this, there are tabs for Runtime, Properties, Insight, Logging, and Static IPs. Under the Runtime tab, the 'Runtime version' is set to 4.0.0, 'Worker size' is set to 0.2 vCores, and 'Workers' is set to 1. There are also checkboxes for 'Automatically restart application when not responding', 'Persistent queues', and 'Encrypt persistent queues'.

31

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
- Deploy, run, and test the application
- View application info in Runtime Manager

The screenshot shows the MuleSoft Design Center interface. On the left, there's a sidebar with Project, Flows, Reusable Configurations, and Data Types. The main area shows a flow named 'Get flights' with two components: 'HTTP Listener flights' and 'Logger'. The 'Logs' section at the bottom shows several INFO-level log entries related to the application's startup and a test message. At the top right, there's a message saying 'Latest changes deployed'.

All contents © MuleSoft Inc.

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

All contents © MuleSoft Inc.

35

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

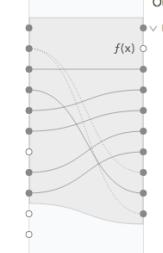
Transform
Configuration
Input
Output
X

Input

```
< Payload Array<Object>
  > plane Object?
    code String?
    price Number?
    origin String?
    destination String?
    ID Number?
    departureDate String?
    emptySeats Number?
```

Attributes Void

Variables Object

f(x)


X

All contents © MuleSoft Inc.
Actions for: (root)#[payload]
Mappings

36

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



All contents © MuleSoft Inc.

37

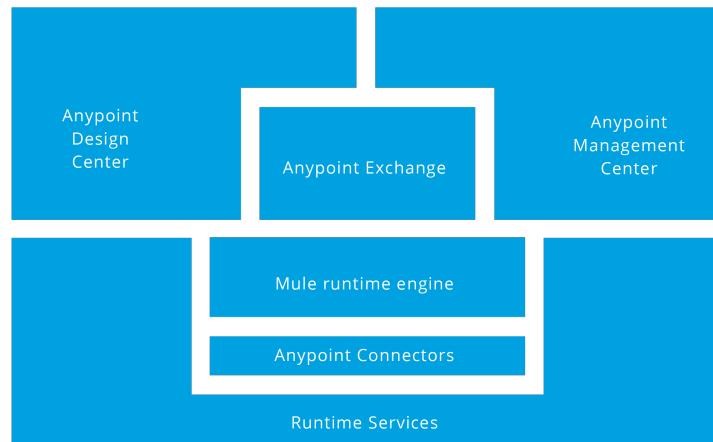
Summary



Summary: Anypoint Platform



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



All contents © MuleSoft Inc.

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** 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

All contents © MuleSoft Inc.