



Module 4

Identifying, Reusing and Publishing APIs



1

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



- Map Acme Insurance's planned **strategic initiatives** to products and projects
- **Identify APIs** needed to implement these products
- Assign each API to one of the **three tiers** of API-led connectivity
- Reason about **composition and collaboration** of APIs
- **Reuse APIs** wherever possible
- **Publish APIs and related assets** for reuse
- Understand how **API Community Manager** helps build developer communities around APIs in an application network

2

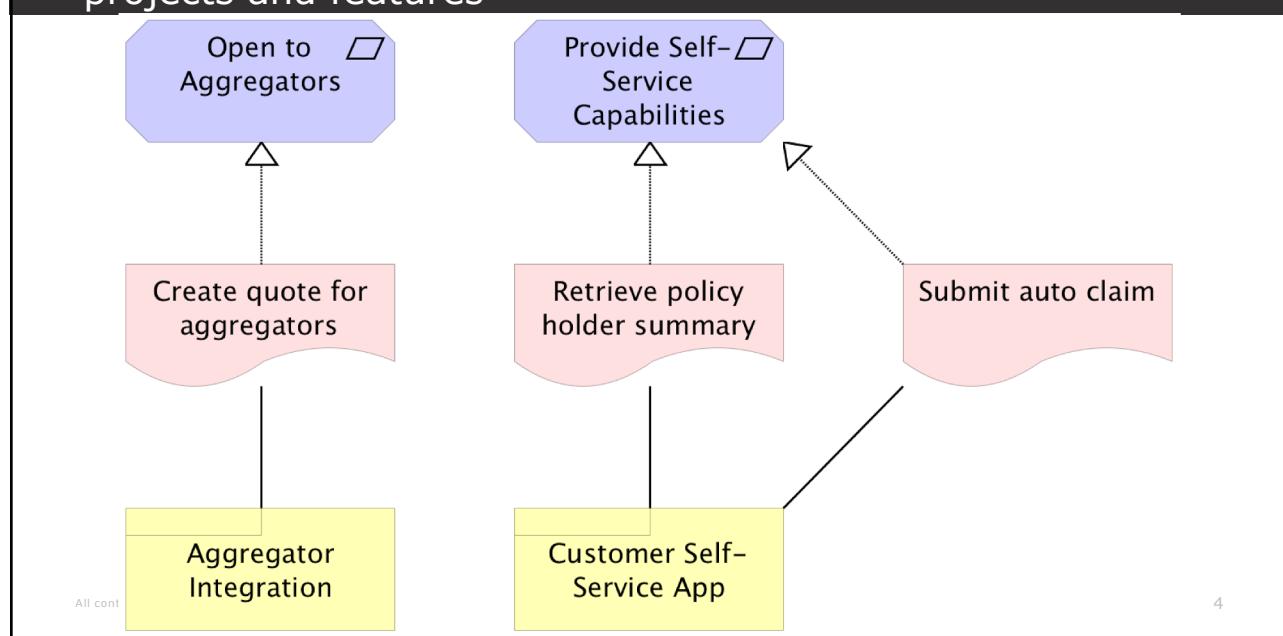
Section 1

Productizing Acme Insurance's strategic initiatives



3

Translating strategic initiatives into products, projects and features

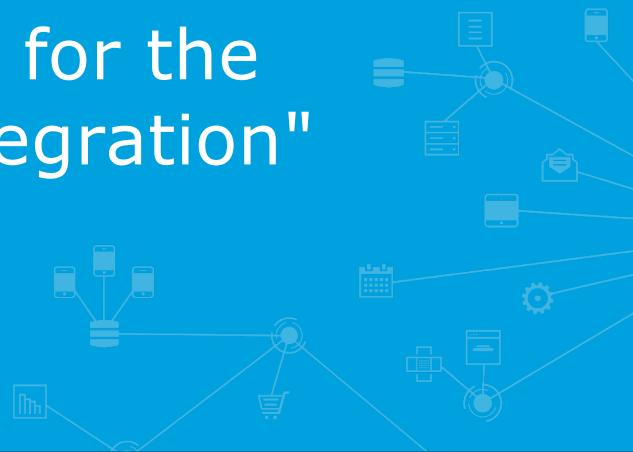


4

2

Section 2

Identifying APIs for the "Aggregator Integration" product



5

Towards an application network



- "Aggregator Integration" product has **one defining feature**:

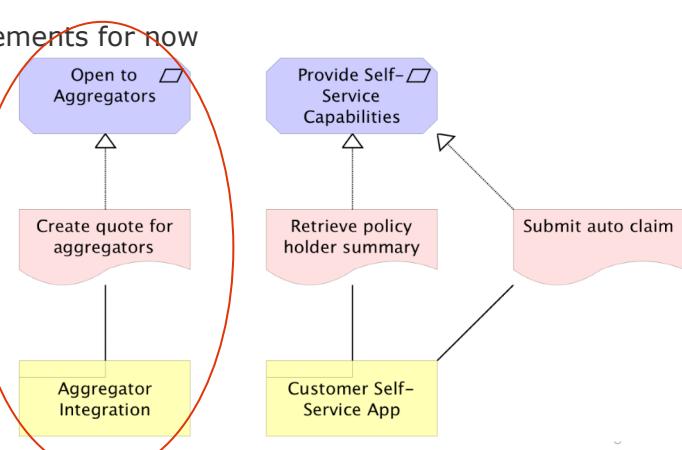
- **Create quote for aggregators**

- Only consider functional requirements for now

- **First API-led connectivity**

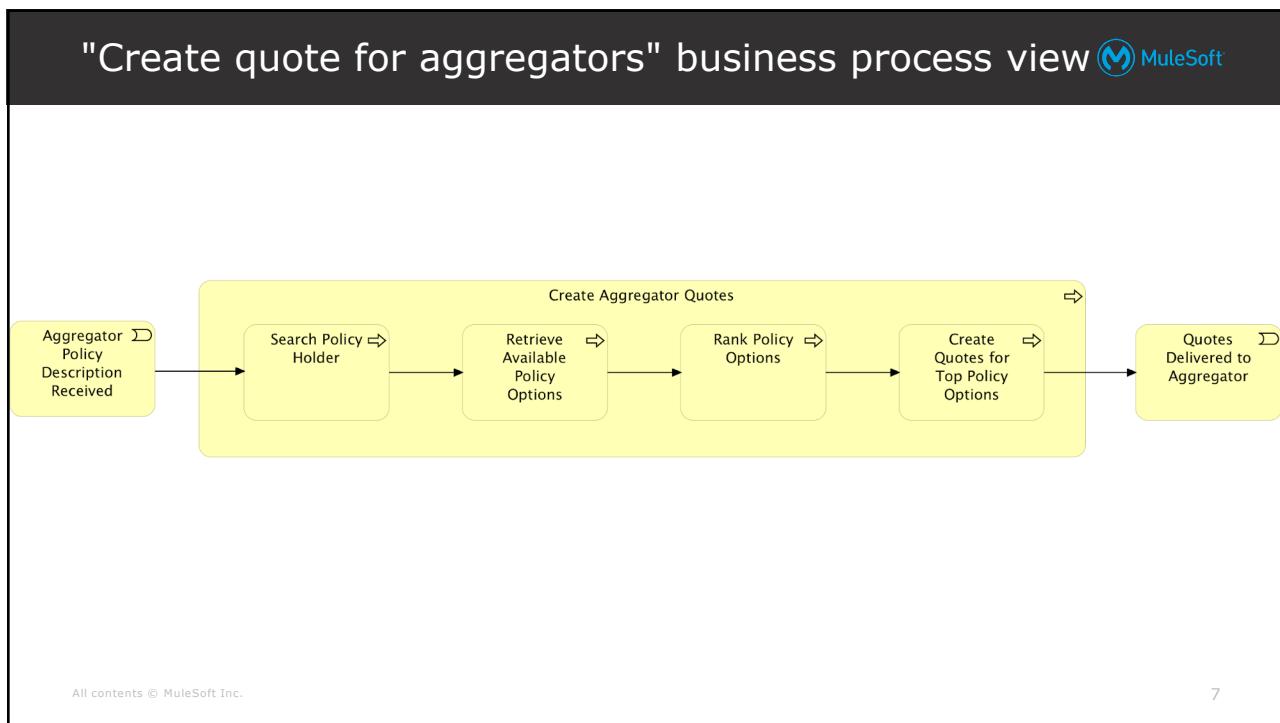
project at Acme Insurance

- **Establish** an Enterprise Architecture compatible with an application network
 - Resulting application network will at first be **minimal**

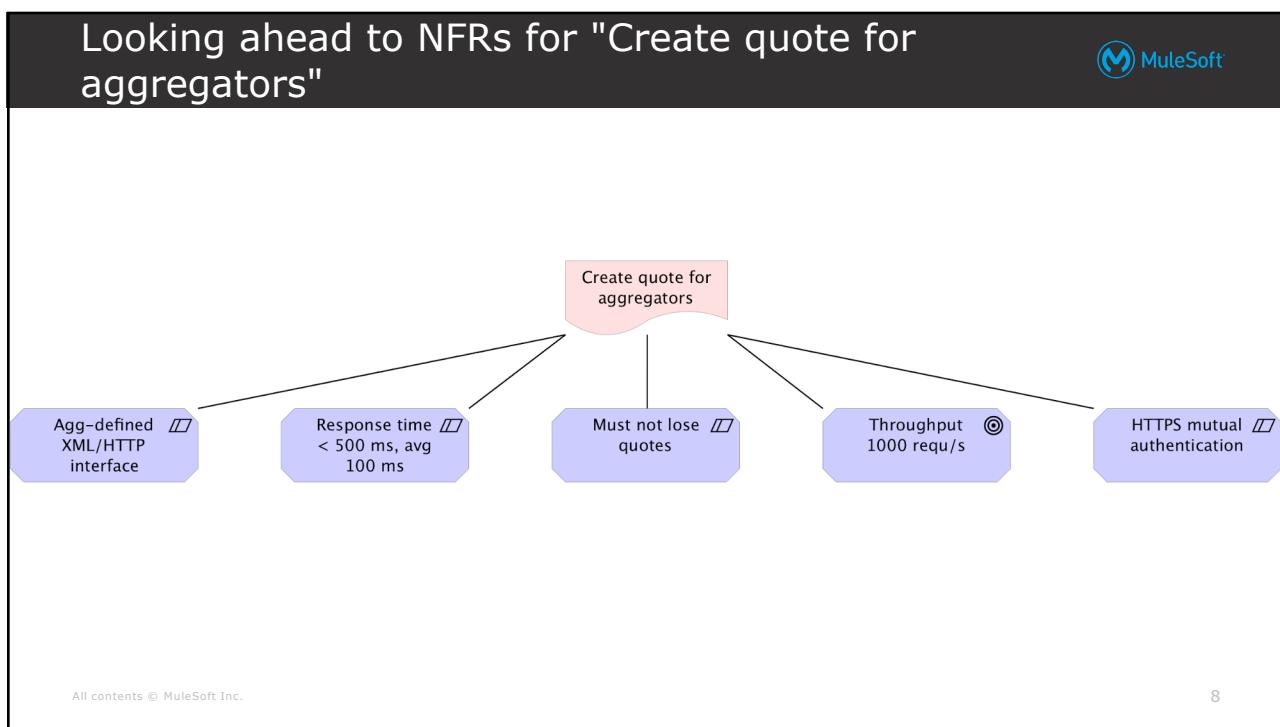


All contents © MuleSoft Inc.

6



7

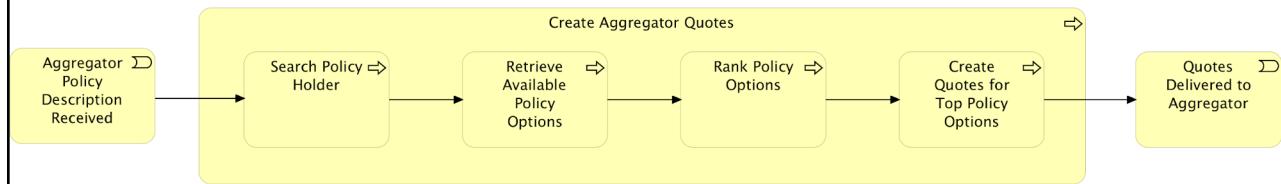


8

Exercise: Identify APIs for the "Create quote for aggregators" feature

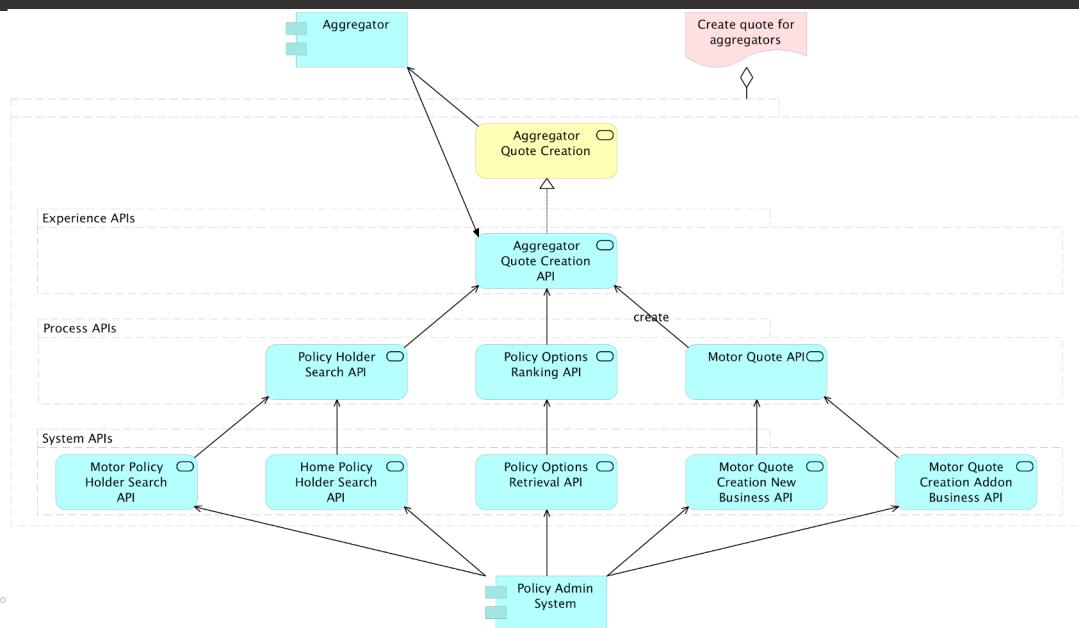


- Using "Create Aggregator Quotes" business process and making assumptions about the capabilities of the Policy Admin System:
- Break down **functionality** of the "Create quote for aggregators" feature into APIs in the 3 tiers of API-led connectivity, and define their interactions:
 - Experience APIs** with interface defined by Aggregator
 - Process APIs** implement and orchestrate business/process logic
 - System APIs** are defined by needs of Process APIs and capabilities of the Policy Admin System



9

APIs for the "Create quote for aggregators" feature



10

10

APIs for the "Create quote for aggregators" feature



- "Aggregator Quote Creation EAPI":
 - (Aggregator policy description) → (0-5 ranked motor policy quotes)
- "Policy Holder Search PAPI":
 - (personal identifiers) → (matching policy holders)
- "Policy Options Ranking PAPI":
 - (policy holder properties, policy properties) → (policy options ranked from highest to lowest)
- "Motor Quote PAPI":
 - create: (policy description, policy holder description) → (motor policy quote)

All contents © MuleSoft Inc.

11

11

APIs for the "Create quote for aggregators" feature



- "Motor Policy Holder Search SAPI":
 - (personal identifiers) → (matching motor policy holders)
- "Home Policy Holder Search SAPI":
 - (personal identifiers) → (matching home policy holders)
- "Policy Options Retrieval SAPI":
 - (policy properties) → (policy options)
- "Motor Quote Creation New Business SAPI":
 - (policy description, policy holder description) → (motor policy quote)
- "Motor Quote Creation Addon Business SAPI":
 - (policy description, policy holder identifier) → (motor policy quote)

All contents © MuleSoft Inc.

12

12

Exercise: Pros and cons of fine-grained APIs and API implementations



The previous is an example of a **pronouncedly fine-grained API architecture**.

A break-down of the same functionality into much more **coarse-grained APIs** is equally possible:

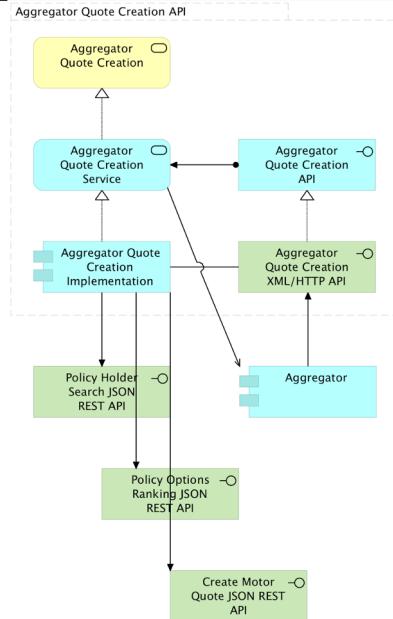
- **Compare and contrast** coarse-grained and fine-grained APIs and API implementations

Exercise: Pros and cons of fine-grained APIs and API implementations



- **Deployability**: evolution and rollout of functionality
- **Management**: monitoring, access control, QoS, ...
- **Scalability**: memory, # CPUs, # machines, ... allocated/scaled for each API implementation
- **Resources**: minimum CPUs/vCores, CloudHub workers, ... per API implementations
- **Complexity**: lower complexity of each node in the application network results in higher complexity of entire application network
- **Latency**: API invocations add latency, unless mitigated
- **Failure modes**: address potential failures of remote interactions
- **Team organization**: parallelization of implementation effort
- **Agility and innovation**: cycle times

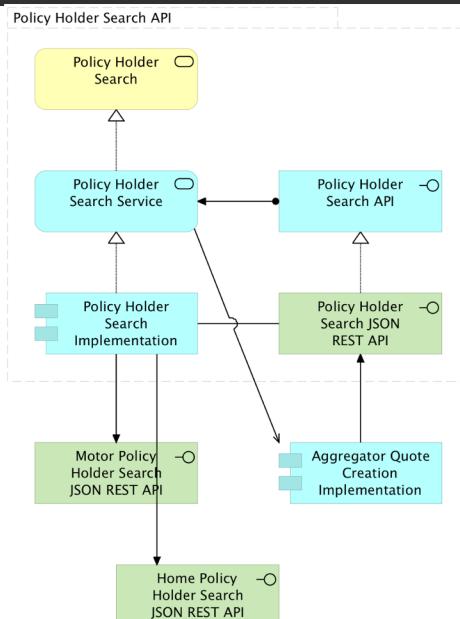
Aggregator Quote Creation EAPI



All contents © MuleSoft Inc.

15

Policy Holder Search PAPI

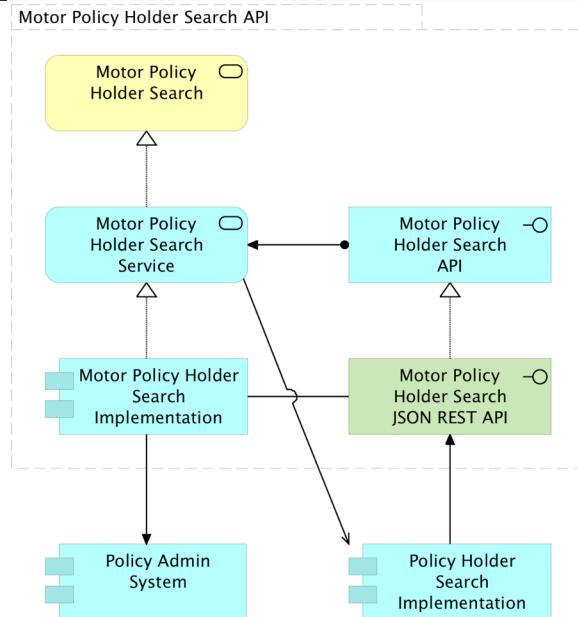


All contents © MuleSoft Inc.

16

16

Motor Policy Holder Search SAPI

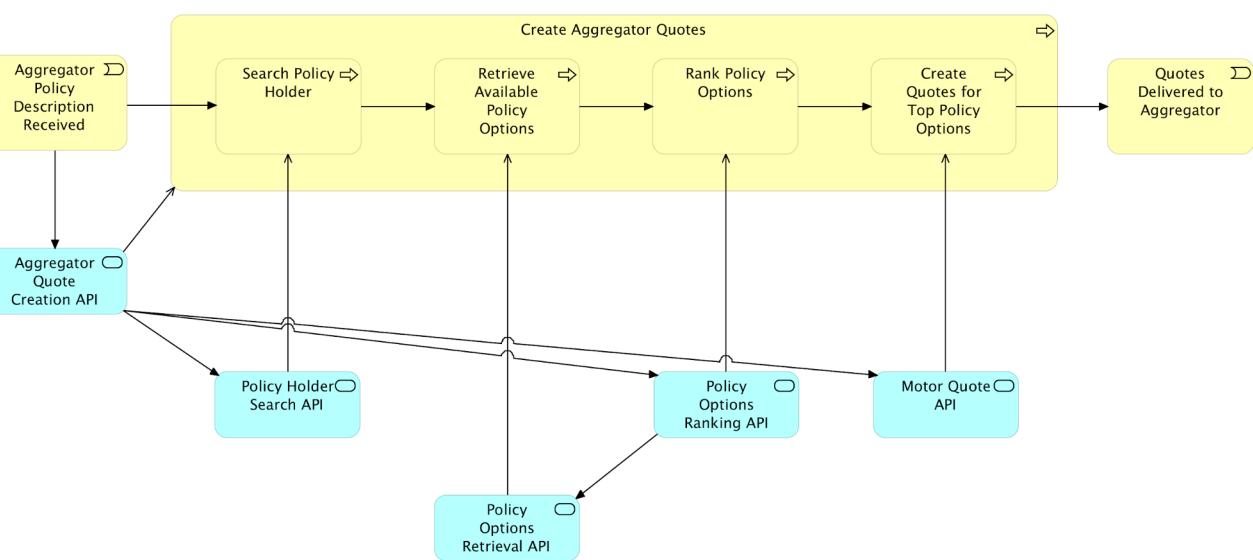


All contents © MuleSoft Inc.

17

17

API-business alignment



All contents © MuleSoft Inc.

18

18

Exercise: Improve reusability of "Create quote for aggregators" feature APIs



You discover that **other Aggregators use different data formats** for communicating with insurance providers:

1. Analyze the APIs identified so far with respect to their dependency on the data format exchanged with the Aggregator
2. Identify new APIs and refine existing APIs to maximize reuse when other Aggregators will need to be supported in the future
3. Describe as clearly as possible which elements of the currently identified APIs will have to change to accommodate your proposed changes

All contents © MuleSoft Inc.
19
19

Exercise: Improve reusability of "Create quote for aggregators" feature APIs



1. Only "Aggregator Quote Creation EAPI" depends on Aggregator-defined data format
2. In future: one Experience API per Aggregator
 - a. Common functionality of these Experience APIs encapsulated in a new Process API: "One-Step Motor Quote Creation PAPI"
 - i. Accepts and returns Aggregator-neutral description of policy and quotes
3. Changes:
 - a. "Aggregator Quote Creation EAPI": Interface unchanged, implementation changed significantly to mainly delegate to "One-Step" Process API
 - b. New "One-Step" Process API with implementation similar to current "Aggregator Quote Creation EAPI"
 - c. New Experience APIs, one per Aggregator, delegating to "One-Step" API
 - d. Current Aggregator does not experience any change

All contents © MuleSoft Inc.
20
20

Section 3

Reusing and publishing API-related assets for the "Aggregator Integration" product



21

Steps to reusing API-related assets



- Attempt **reuse** by finding similar APIs
 - Search Exchange
- **Announce** the fact that the chosen APIs will be implemented
 - Basic **API specification/RAML definition** required for each API
 - Create API spec in **Design Center**, **export to Exchange**
 - **Version** states **readiness**: v0 for API spec, 0.0.1 for Exchange asset
 - **API portal** automatically rendered in Exchange
- The **C4E** provides guidance and support
 - Defines **naming conventions** for assets published to Exchange

22

Defining RAML



- **RESTful API Modeling Language**
- Language for the **machine- and human-readable** definition of APIs that embody most or all of the principles of REST
 - Uniform interface, stateless, cacheable, client-server, layered system, code on demand
 - Adheres to the HTTP specification in
 - HTTP methods (GET, HEAD, OPTIONS; PUT, DELETE; POST, PATCH; ...)
 - HTTP response status codes
 - HTTP request and response headers
- Comparable to OpenAPI Specification (**OAS**)

All contents © MuleSoft Inc.

23

23

Defining RAML definition



- An **API specification** expressed in **RAML**
- Comparable to an **OpenAPI definition**
- Comprising one main **RAML document**, optionally with
 - RAML **fragment** documents
 - XSD and JSON-**Schema** documents
 - **examples**, etc.

All contents © MuleSoft Inc.

24

24

Defining OpenAPI Specification (OAS)



- Standard, programming language-agnostic **interface description language for REST APIs**
- **JSON** or **YAML**-based
- Comparable to **RAML**
- “The spec formerly known as **Swagger**”

All contents © MuleSoft Inc.

25

Defining OpenAPI definition



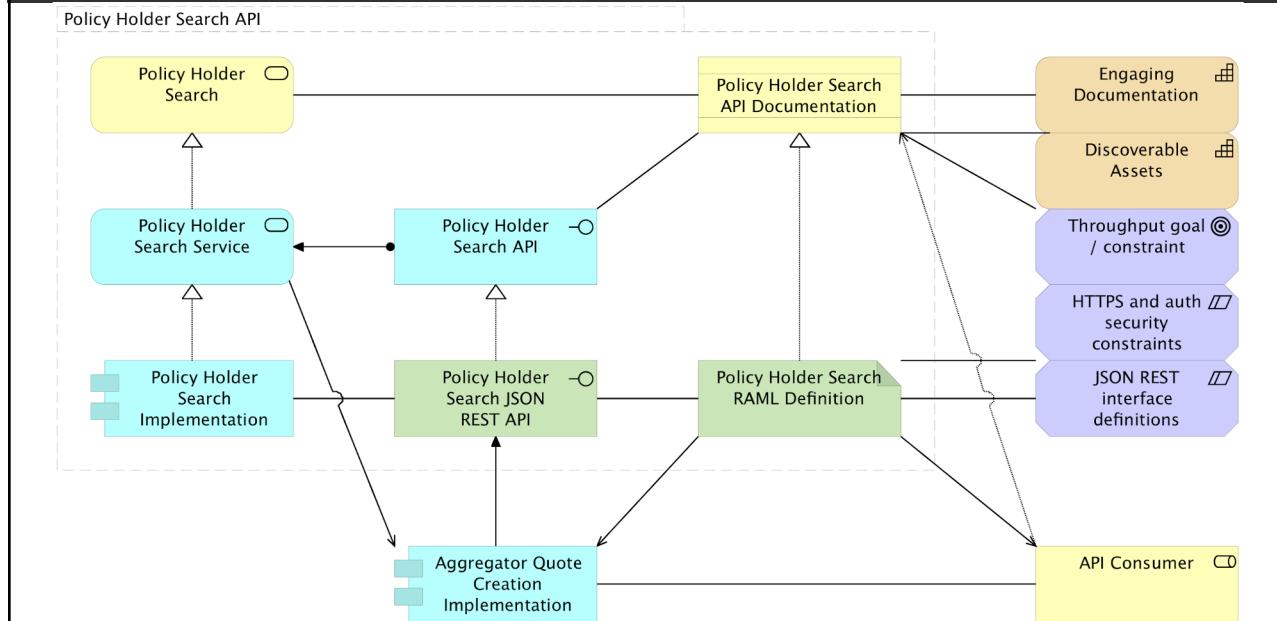
- An **API specification** that uses and conforms to the OpenAPI Specification (**OAS**)
- Expressed in **JSON** or **YAML**
- Comparable to a **RAML definition**

All contents © MuleSoft Inc.

26

26

"Policy Holder Search PAPI" documentation



27

Using Design Center to sketch and simulate a RAML definition for "Policy Holder Search PAPI"



The screenshot shows the MuleSoft Design Center interface for the Policy Holder Search PAPI. On the left, the RAML file `policy-holder-search-papi.raml` is displayed with its code:

```

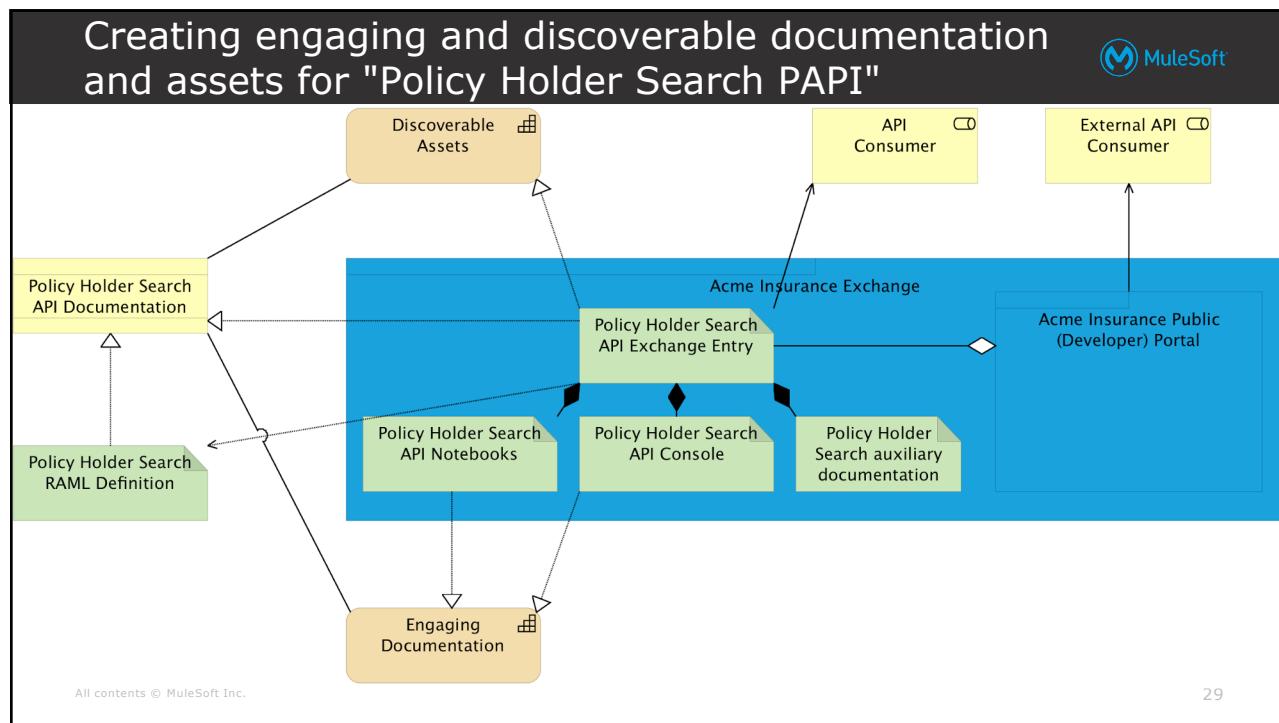
24   |     name: Max The Mule
25   |     dob: 2006-05-04
26
27   |   Policyholder:
28   |     description: A person that is a policy holder.
29   |     type: Person
30   |     properties:
31   |       customerSince?: date-only
32   |       lobs: LoB[]
33
34   |     example:
35   |       auto-minimal:
36   |         name: Max The Mule
37   |         lobs: [ PersonalMotor ]
38
39   |     full:
40   |       name: Max The Mule
41   |       dob: 2006-05-04
42   |       customerSince: 2006-05-05
43   |       lobs: [ PersonalMotor, Home ]
44
45
46   /policyHolders:
47   |   description: |
48   |     Policy holders of any policy of a personal LoB, such that the policy holders
49   |     are people.
50
51   |   get:
52   |     description: Search for policy holders matching the query parameters.
53   |     queryParameters:
54   |       name?:
55   |         type: string
56   |         description: A case-insensitive portion of the matching policy holder's
57   |           name,
58   |           example: max
59   |           dob?:
60   |             type: date-only
61   |             description: The exact date of birth of the matching policy holder.
62   |             example: 2006-05-04

```

The right pane shows the API simulation interface for the `/policyHolders : get` endpoint. It includes sections for Request, Parameters, Properties, Path parameters, Query parameters, and Response.

28

28



29

Publishing an Exchange entry for "Policy Holder Search PAPI"

Version	Instances
1.0.3	Mocking Service Staging - v1:7483787
1.0.2	⋮
1.0.1	⋮

30

Exchange auto-generates a REST Connector

31

Understanding the API Console for "Policy Holder Search PAPI" (Design Center, Exchange, Studio)

32

Including an API Notebook for "Policy Holder Search PAPI"



The screenshot shows the MuleSoft Anypoint Platform interface. On the left, there's a sidebar with options like 'Assets list', 'Policy Holder Search PAPI' (which is selected), 'On Call', 'Onboarding', 'API notebook' (which is highlighted in blue), 'API implementations', 'Runbook', '+ Add new page', 'API summary', '+ Add terms and conditions', and 'API instances'. The main area is titled 'Policy Holder Search PAPI' with a version 'v1'. It contains an 'API notebook' section with a toolbar for bold, italic, code, etc., and buttons for 'Delete Page', 'Markdown', and 'Visual'. Below this, there's a code editor with the following Java code:

```
API.createClient('client', '42d8fc5a-b6f4-4f3a-a583-498debc7dba3', 'policy-holder-search')
```

With a 'Play' button. Further down, there's another code editor with:

```
var body = client.policyHolders.get(null, null).body;
```

With a 'Play' button. At the bottom, there are buttons for 'Discard changes' and 'Save as draft'.

33

Publishing "Policy Holder Search PAPI" to Acme Insurance's Public (Developer) Portal



The screenshot shows the Acme Insurance public developer portal. On the left, there's a sidebar with 'Home' and 'My applications'. The main area has a dark header with the text 'Acme Insurance public developer portal'. Below it, there's a message: 'Build your application network faster! Get started with powerful tools, intuitive interfaces, and the best in class documentation experience.' In the center, there's a search bar and a list of assets. One asset is highlighted: 'Policy Holder Search PAPI' by 'AnyInsurance Owner' (with a 5-star rating). To the right, a modal window is open titled 'Share Policy Holder Search PAPI'. It shows sharing details: 'Add to the public portal', a search bar for users ('Search for a user'), and dropdowns for 'Viewer' and 'Add'. It also shows the user 'AO AnyInsurance Owner (anyinsurance-owner)' and roles 'X Admin' and 'Admin'. At the bottom of the modal are 'Cancel' and 'Share' buttons. The overall interface is clean and modern.

34

Publishing "Policy Holder Search PAPI" to Acme Insurance's Public (Developer) Portal

The screenshot shows the MuleSoft Anypoint Platform interface for managing APIs. On the left, a sidebar lists navigation options like 'Assets list', 'Policy Holder Search PAPI', 'On Call', 'Onboarding', 'API notebook', 'API implementations', 'Runbook', 'API summary', 'Types', 'Resources', '/policyHolders', and a 'GET' button. The main content area displays the 'Policy Holder Search PAPI' details, version v1, with a 4-star rating and 1 review. It includes sections for 'API summary', 'API base URI' (http://ans-policyholdersearch-papi.cloudhub.io/{version}), and 'API resources' (listing '/policyHolders' with a description: 'Policy holders of any policy of a personal LoB, such that the policy holders are people.'). A 'Reviews' section is also present. The top right corner features the MuleSoft logo.

35

Repeat for all APIs for the "Create quote for aggregators" feature

The screenshot shows the MuleSoft Anypoint Platform Exchange interface. The left sidebar shows categories: All, MuleSoft, Acme Insurance (which is selected), My applications, and Public Portal. The main area is titled 'Acme Insurance' and shows a search bar with 'REST APIs' selected. Below it, a message says 'Showing results for REST APIs.' A grid of eight API cards is displayed, each representing a different REST API: 'Policy Options Ranking PAPI', 'Policy Options Retrieval SAPI', 'Aggregator Quote Creation EAPI', 'Policy Holder Search PAPI', 'Home Policy Holder Search SAPI', 'Motor Policy Holder Search SAPI', 'Mobile Policy Holder Summary EAPI', and 'Motor Quote Creation Addon Bus SAPI'. Each card includes a small icon, a star rating (all are 5 stars), and the owner information 'AnySurance Owner'.

36

Section 4

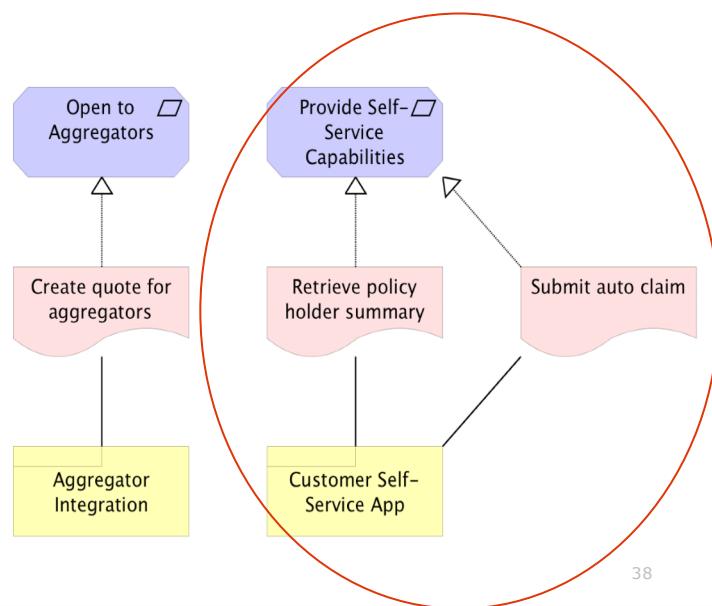
Identifying, reusing and publishing APIs and API-related assets for the "Customer Self-Service App" product

37

Growing the application network for the "Customer Self-Service App" product



- Only consider functional requirements for now
- Second API-led connectivity project at Acme Insurance**
 - Grow nascent application network
 - C4E essential for **cross-LoB** activities

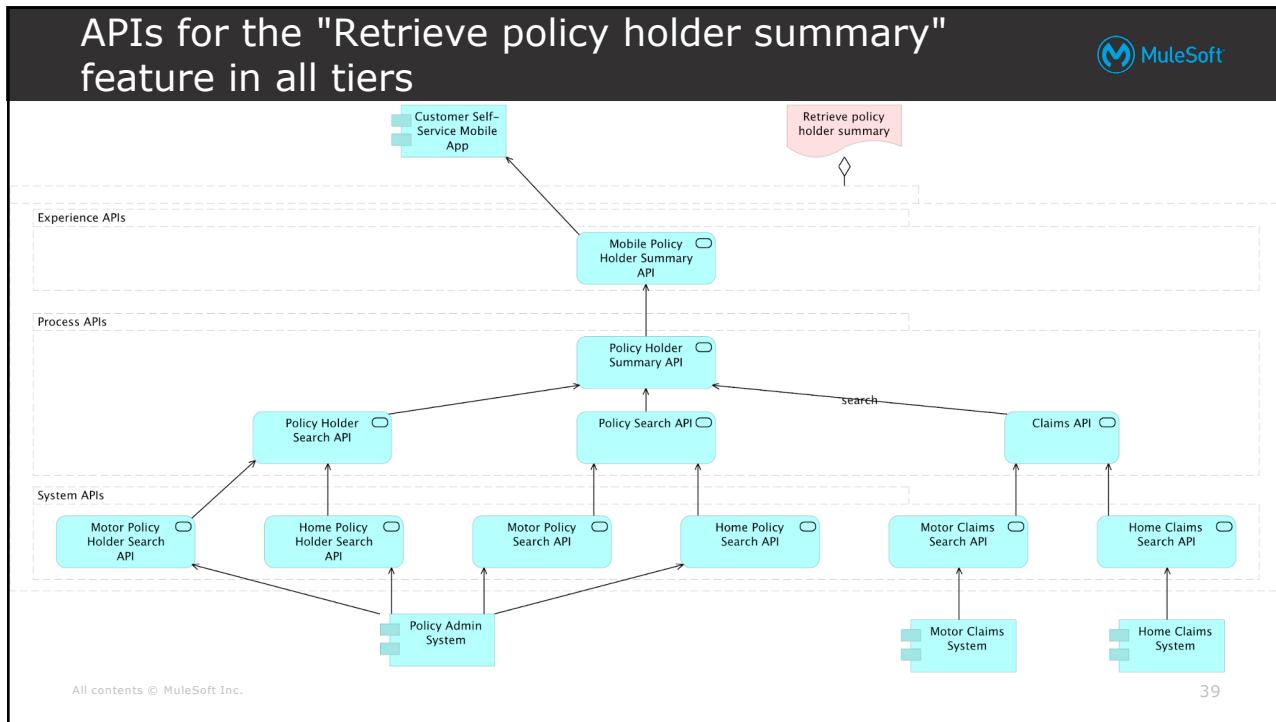


All contents © MuleSoft Inc.

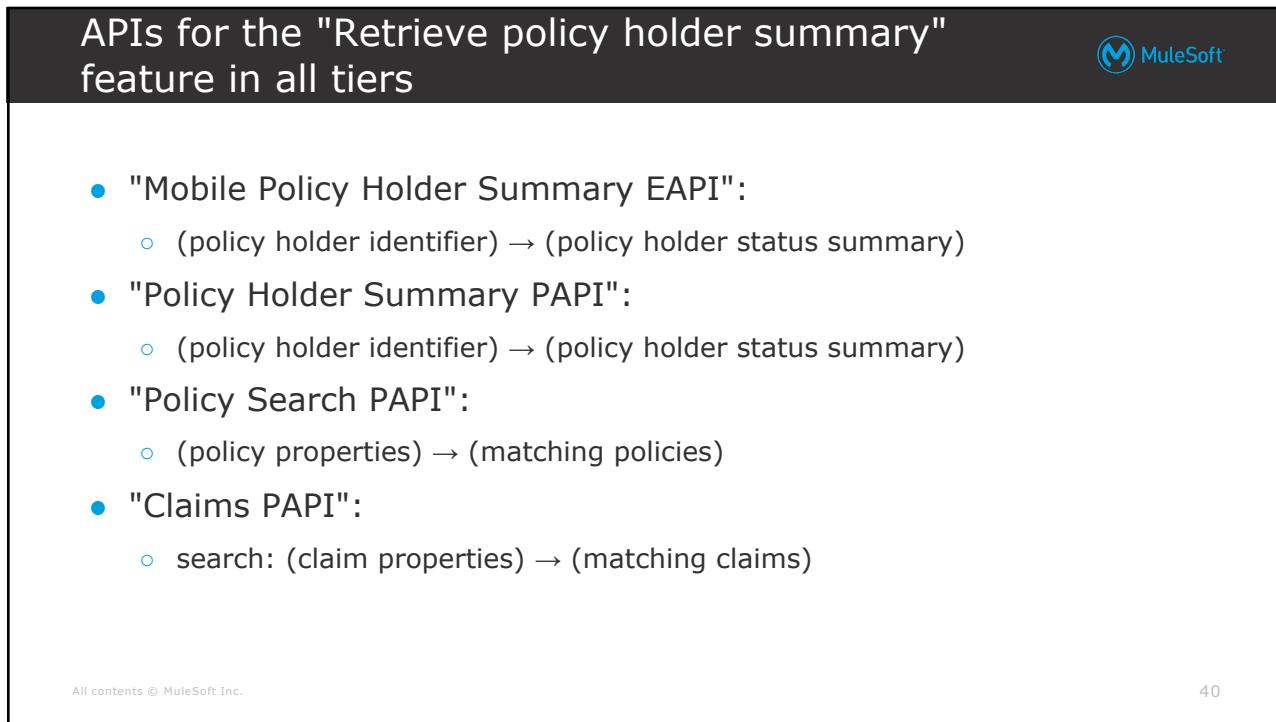
38

38

19



39



40

APIs for the "Retrieve policy holder summary" feature in all tiers



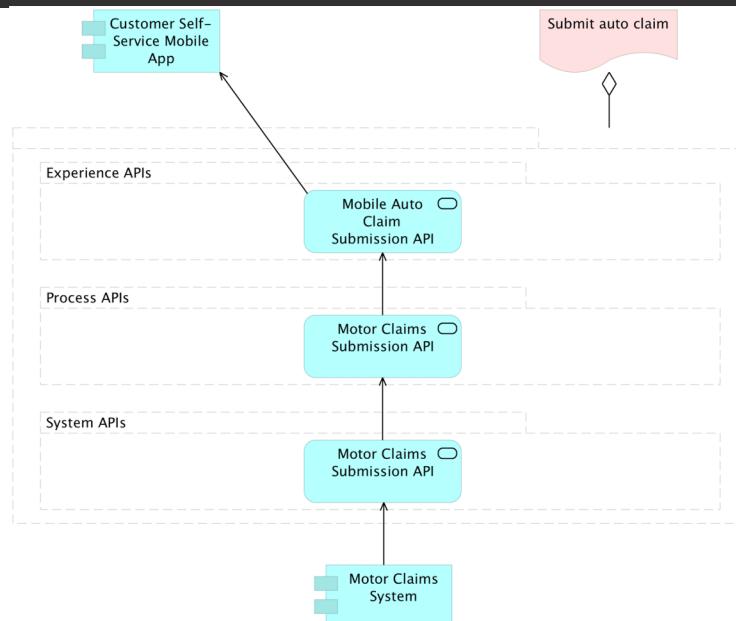
- "Motor Policy Search SAPI":
 - (motor policy properties) → (matching motor policies)
- "Home Policy Search SAPI":
 - (home policy properties) → (matching home policies)
- "Motor Claims Search SAPI":
 - (claim properties) → (matching motor claims)
- "Home Claims Search SAPI":
 - (claim properties) → (matching home claims)

All contents © MuleSoft Inc.

41

41

APIs for the "Submit auto claim" feature in all tiers



All contents © MuleSoft Inc.

42

42

APIs for the "Submit auto claim" feature in all tiers



- "Mobile Auto Claim Submission EAPI":
 - (claim description) → (acknowledgement)
- "Motor Claims Submission PAPI":
 - (claim description) → (acknowledgement)
- "Motor Claims Submission SAPI":
 - (claim description) → (acknowledgement)

All contents © MuleSoft Inc.

43

43

Publishing APIs and API-related assets



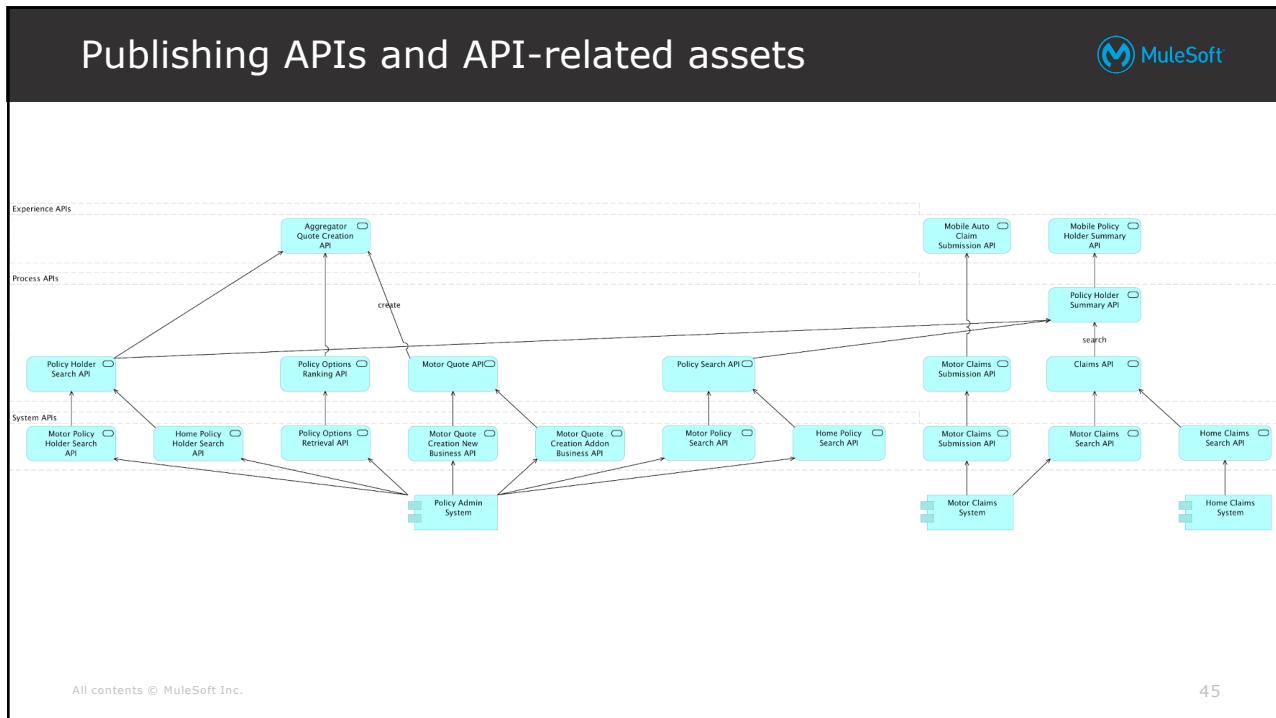
Acme Insurance's application network now has assets for all known APIs:

- **API specs** in the form of **RAML definitions**
- Entries in Acme Insurance's **Exchange**
 - Include **API Console**, **API Notebook**
 - Pointing to the API's **instances/endpoints**: so far only **mocking service**
- Acme Insurance **Public (Developer) Portal** for public APIs
- **No NFRs** have been addressed
- **No API implementations** and **no API clients** have been developed

All contents © MuleSoft Inc.

44

44



45

Publishing APIs and API-related assets

46

Section 5

Anypoint DataGraph



47

Data is fragmented across multiple APIs today



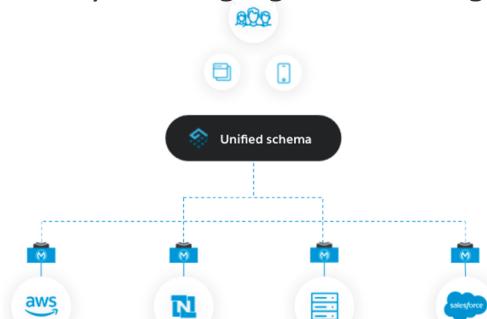
- REST APIs - the most commonly used API standard - are typically consumed one at a time.
- The developer has to write requests to every API they need
- At times, REST API responses are complex, and return every possible field.

48

Anypoint DataGraph



- Anypoint DataGraph supports the GraphQL use case of consolidating many APIs together into a single endpoint
- Anypoint DataGraph unifies all the data within application network in a unified schema by leveraging the existing capabilities.



All contents © MuleSoft Inc.

49

49

Anypoint DataGraph



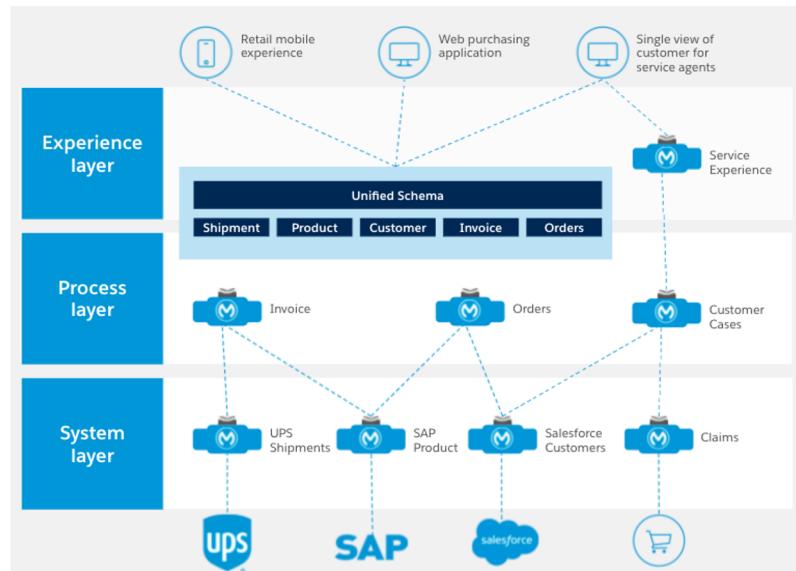
- To create a more clean and connected unified schema, Anypoint DataGraph provides three additional functionalities that allows to:
 - Enable collaboration between object types
 - Link object types to create connections between them
 - Merge object types to combine similar types into a single type to extend their fields and datasets for a more enriched query result

All contents © MuleSoft Inc.

50

50

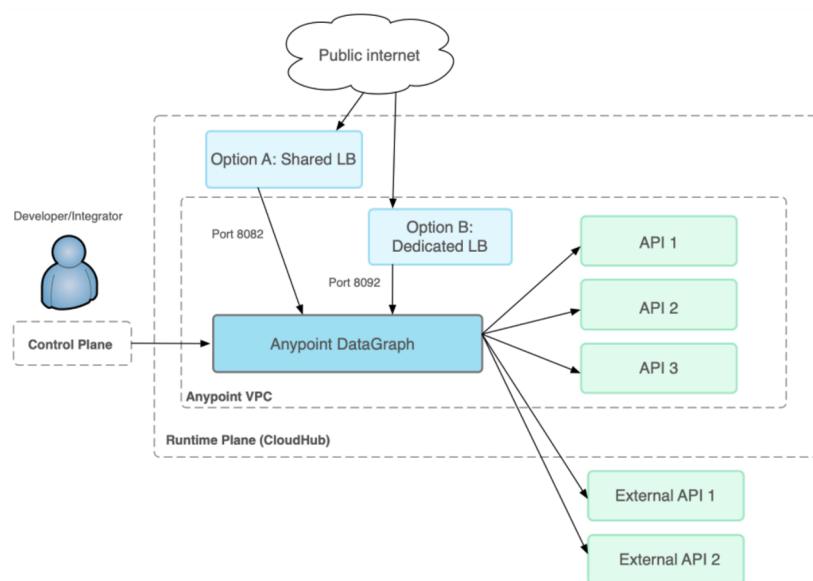
Anypoint DataGraph in API-Led Connectivity



51

51

Anypoint DataGraph architecture and networking



52

52

Section 6

Building developer communities around APIs with Anypoint API Community Manager



53

API Community Manager



- **Exchange** and Public Developer Portals have limitations
 - Not general purpose **CMSs**
 - Limited visual **customization**
 - No discussion forums or other advanced interaction channels with API consumers
- **API Community Manager**
 - Reuses information available in Anypoint Platform and **Exchange**
 - Provides advanced **presentation and interaction** capabilities

54

API Community Manager



API Consumer



API Developer



C4E Architect



Technical Writer



Marketing manager



Brand/Graphic designer



API Product Manager

Exchange API portals

- Auto-generated
- Searchable catalog
- Lightweight branding

API Community Manager

- Branded, personalized experiences
- Forums, blogs and events
- Support case management

All contents © MuleSoft Inc.

55

API Community Manager



- Build and operate **API consumer communities**
 - Internal and external developers (partners)
- **Customization**, branding, marketing, and engagement
- **Surface** selected APIs cataloged in Exchange
- Manage **API clients**, access credentials, and consumption metrics
- Implemented on **Salesforce Experience Cloud**
 - Invokes Anypoint **Platform APIs** to retrieve API-related information
 - Limitless customization through **Experience Builder**
 - Extensive **dashboards and reports** about how API consumers interact with assets in API Community Manager

All contents © MuleSoft Inc.

56

API Community Manager branding and customization



All contents ©

57

57

API Community Manager content sources



1 - 6 of 6

58

58

API Community Manager API console

 MuleSoft

Mortgage Eligibility API

Analyzes key financial assets held or registered within the customer's account, including credit history, and provides mortgage pre-approval for a specific amount.

- Home
- Introduction
- Use cases
- How-to: high-volume
- How-to: async call
- Securing customer information
- Performance tuning
- Troubleshooting
- Summary
- Endpoints

Mortgage Eligibility API

Version: 1

Analyzes key financial assets held or registered within the customer's account, including credit history, and provides mortgage pre-approval for a specific amount.

<https://mortgage-eligibility.api.hsbc.com>

Supported protocols

HTTPS HTTP

API endpoints

/customer
/customer/{cid}
/customer/{cid}/application

Try this API

Select API Instance ▾
Mocking Service
Generate SDK

Select API Instance ▾

All contents 59

59

API Community Manager API client management

 HSBC Home News Get Started API Catalog Community Support

Ready to get started?

Our API platform lets you easily access financial solutions and approved customer data, and implement it seamlessly in your products and services. All documentation and sample code is available for each API.

It's really so straightforward that the only question will be, 'what will you build'?

1 2 3 4

Get Sandbox API key

Sign up and browse our API catalogue to assemble a suite of APIs for your digital experiences.

Client ID	d31d32af-cfe6-4562-9c66-44cc98c837fb
Client Secret	*****

Download SDK and build

Discover our range of innovative resources that push boundaries and redefine what's possible.

Programming Language	JavaScript
Download	

Promote API key to production

Get straight to work in a production environment, with live testing within your app or website.

Launch

Move straight into production and connect your customers with unique digital experiences.

All contents 60

60

API Community Manager admin

The screenshot shows the API Community Manager admin interface. At the top, there's a header with the HSBC logo, a search bar, and navigation links like Home, Developer Journey, Community Member Profiles, and Company Accounts. Below the header is a blue navigation bar with tabs: Manage APIs, Manage Members, Manage CMS Content, Manage Forums, Dashboards, Community Administration, Community Builder, and Open Community. A search bar labeled "Filter APIs" is positioned below the navigation bar. The main content area displays a table titled "APIs in this Community". The columns are: Version, Visibility, Description, Business Group, and Actions (Remove, Edit). There are six entries listed:

Version	Visibility	Description	Business Group	Action
1	Everyone	Accept PII data and process applications for HSBC Credit Cards. Can be used in-line with other services such as saving or checking account opening.	HSBC CMB	<button>Remove</button> <button>Edit</button>
1	Everyone	Alert users when overdraft occurs. API response includes account name, type, and overdraft amount.	HSBC CMB	<button>Remove</button> <button>Edit</button>
1	Everyone	Sends alerts when new or existing HSBC customers open an investment product account.	HSBC CMB	<button>Remove</button> <button>Edit</button>
1	Everyone	Analyzes key financial assets held or registered within the customer's account, including credit history, and provides mortgage pre-approval for a specific amount.	HSBC CMB	<button>Remove</button> <button>Edit</button>
1	Everyone	Analyzes key financial assets held or registered within the customer's account, including credit history, and provides mortgage pre-approval for a specific amount.	HSBC CMB	<button>Remove</button> <button>Edit</button>

At the bottom right of the table, it says "61".

61

Salesforce Community Builder for API communities

The screenshot shows the Salesforce Community Builder for API communities interface. At the top, there's a header with the MuleSoft logo and navigation links like Publish and Preview. Below the header is a sidebar with icons for Pages, API Catalog, Create Record, Error, Get started, News Detail, Search, support, testpage, and various object pages. The main content area is titled "API Catalog" and shows a grid of API cards. The cards include:

- Mortgage Eligibility API: Analyzes key financial assets held or registered within the customer's account, including credit history, and provides mortgage pre-approval for a specific amount.
- Current Account Overdraft API: Alert users when overdraft occurs. API response includes account name, type, and overdraft amount.
- Investment Product Opening API: Sends alerts when new or existing HSBC customers open an investment product account.
- Saving Account Opening API: Analyzes key financial assets held or registered within the customer's account, including credit history, and provides mortgage pre-approval for a specific amount.

At the bottom right of the grid, it says "62".

62

Summary



63

Summary



- Strategic initiatives map to "**Aggregator Integration**" product and "**Customer Self-Service App**"
 - **3 Experience APIs, 7 Process APIs and 10 System APIs**
- 1 Process API and 2 System APIs **reused**
- Used **Design Center** to sketch and simulate **API specs** in the form of **RAML definitions** for each API
- Created and published **Exchange** entries with **API Console** and **API Notebook**
- APIs from Exchange can be exposed in **API Community Manager**
 - Visually more **compelling** and more **interactive** experience than through Exchange and Exchange Portals alone

All contents © MuleSoft Inc.

64

64