

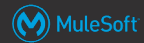


Module 8

Augmenting API-Led Connectivity With Elements From Event-Driven Architecture

1

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



- Selectively choose elements of **Event-Driven Architecture** in addition to API-led connectivity
- Make effective use of **events and message destinations**
- Impose **event exchange patterns** in accordance with API-led connectivity
- Describe **Anypoint MQ** and its features
- Apply Event-Driven Architecture with Anypoint MQ to address NFRs of the "**Customer Self-Service App**" product

2

Section 1

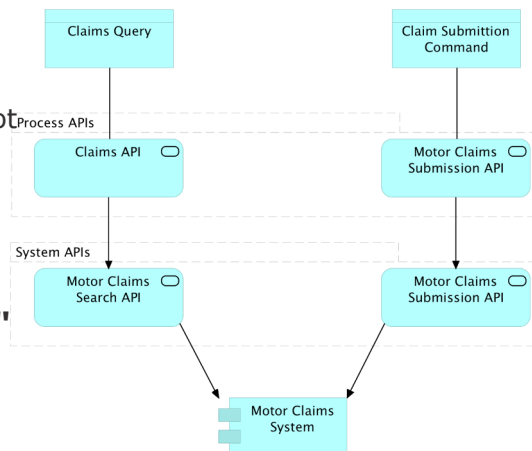
Choosing Event-Driven Architecture to meet some NFRs of the "Customer Self-Service App" product

3

Revisiting the NFRs for the "Customer Self-Service App" product

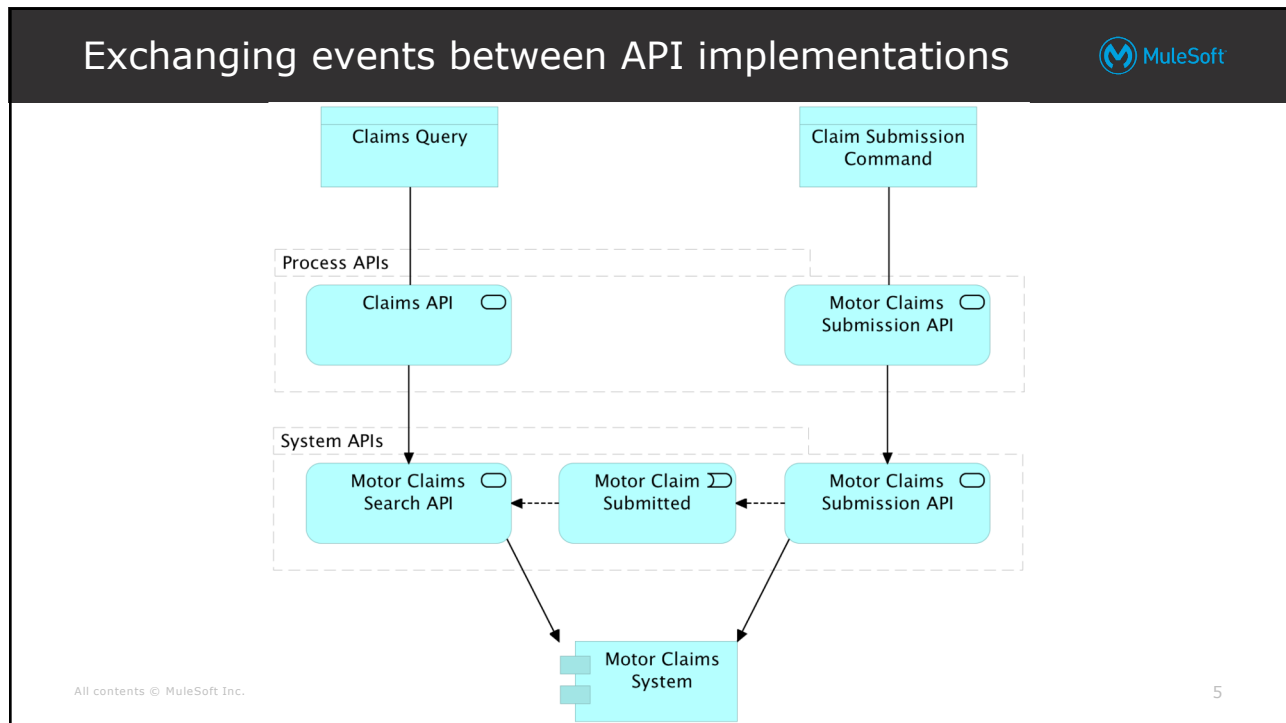


- "Customer Self-Service App" requires
 - **Claim submissions** from the Customer Self-Service Mobile App
 - **Visible immediately** through that app
 - Although **Motor Claims System** does not give access to newly submitted claims until **after lengthy verification** phase
- Link from **"Submit auto claim"** to **"Retrieve policy holder summary"** outside of Motor Claims System



All contents © MuleSoft Inc.

4



5

Section 2

Understanding the nature of Event-Driven Architecture in the context of API-led connectivity

6

Defining Event-Driven Architecture



- Architectural style
- **Asynchronous exchange of events**
- Between **application components**
- Form of **message-driven architecture**
 - Exchanged messages are/describe events
 - Typically **publish-subscribe**

All contents © MuleSoft Inc.

7

7

Exercise: Differences between API-led connectivity and Event-Driven Architecture



1. Compare events and APIs
2. Compare Event-Driven Architecture and API-led connectivity
3. Does Event-Driven Architecture lead to the emergence of an application network?
4. If so, what are the consumable assets associated with Event-Driven Architecture?

All contents © MuleSoft Inc.

8

8

Comparing events and APIs



- **Programmatic**
- **Meaning:** state change vs programmatic interface to a service
- **Dynamic nature:**
 - Event corresponds to API invocation
 - Historical fact vs action to be performed
- **Static nature:** Event type corresponds to API and API data model
- **Granularity:** API comparable to group of event types
- **Synchronicity**
- **Communication path:**
 - API client -> API implementation :: producer -> destination -> consumer(s)
- **Broker**
- **Contract:** API spec vs destination + event type

All contents © MuleSoft Inc.

9

9

Comparing Event-Driven Architecture and API-led connectivity

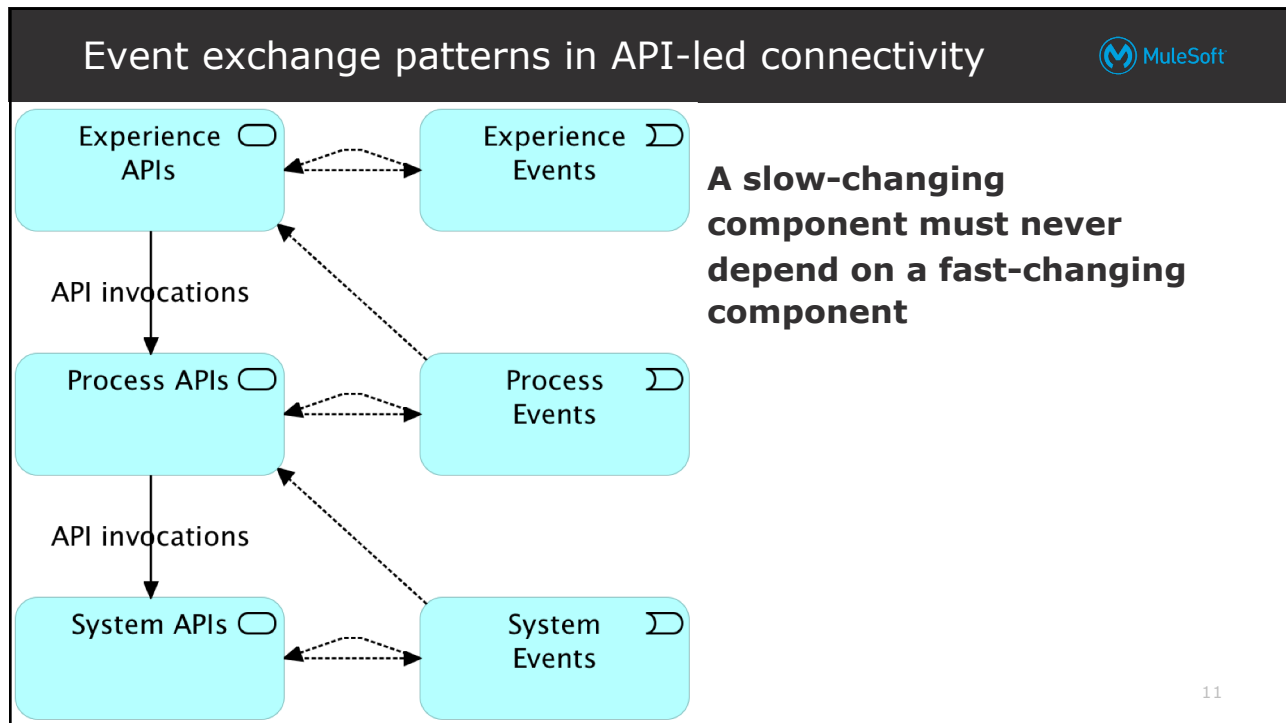


- **Three tiers**
- **Communication patterns** according to three tiers
- **Static and dynamics dependencies:**
 - On other **APIs** and/or backend systems
 - On **event types, destinations** and message **broker**
 - Event **consumers** may change dynamically
- **Shared message broker**
- API-centric **assets** published for self-service consumption
 - Versus destinations and event types
- Enforcing **NFRs via API policies** on top of existing API implementations
 - No equivalent in Event-Driven Architecture on Anypoint Platform

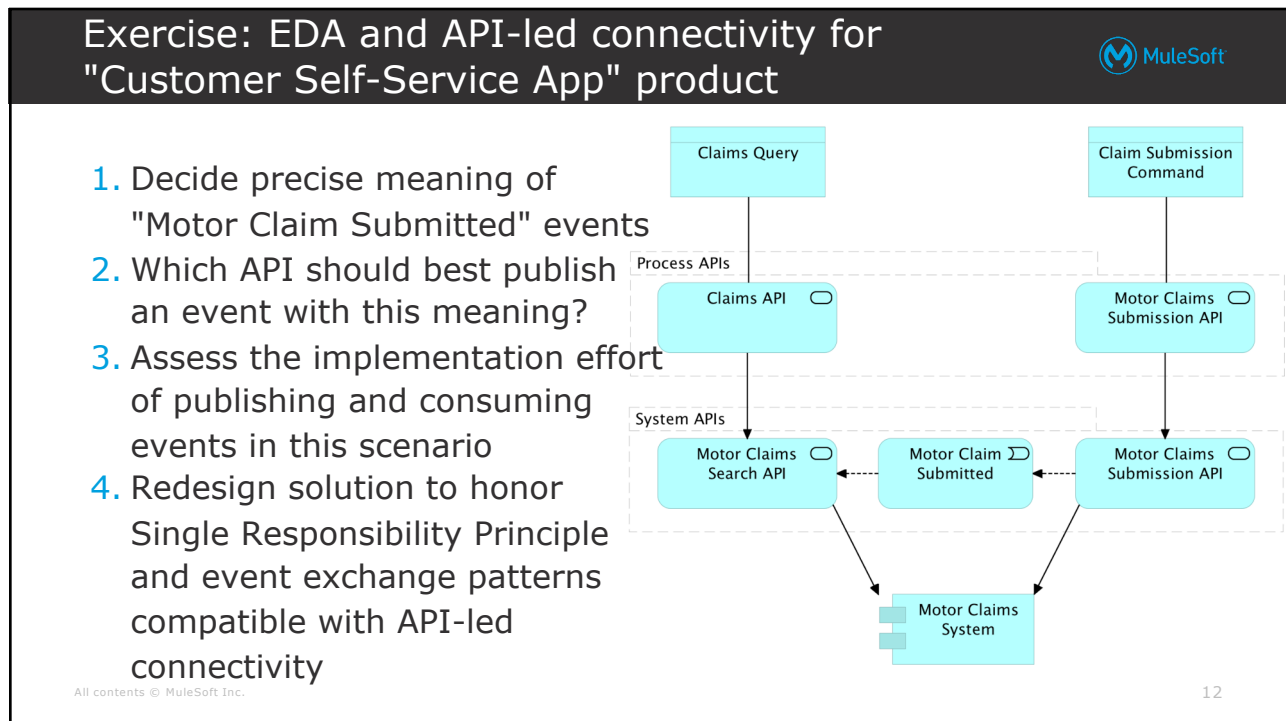
All contents © MuleSoft Inc.

10

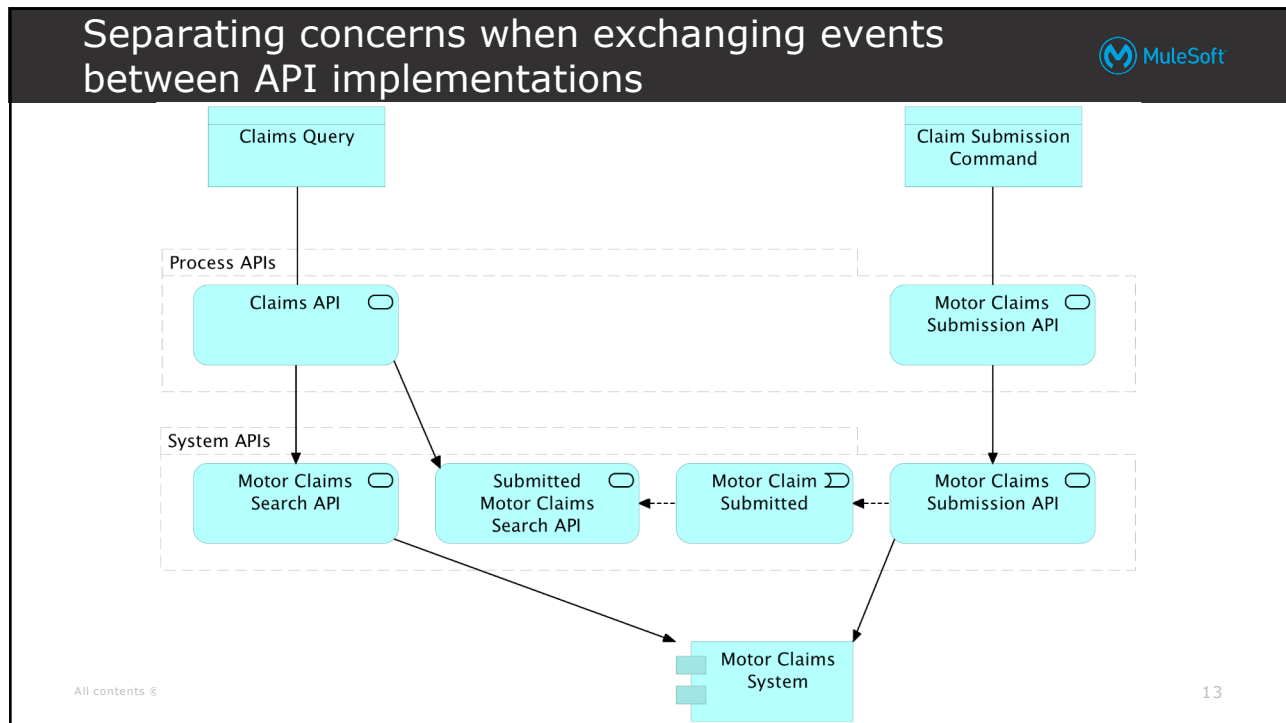
10



11



12



13

Section 3

Adding support for asynchronous messaging to the Technology Architecture

14

Introducing Anypoint MQ



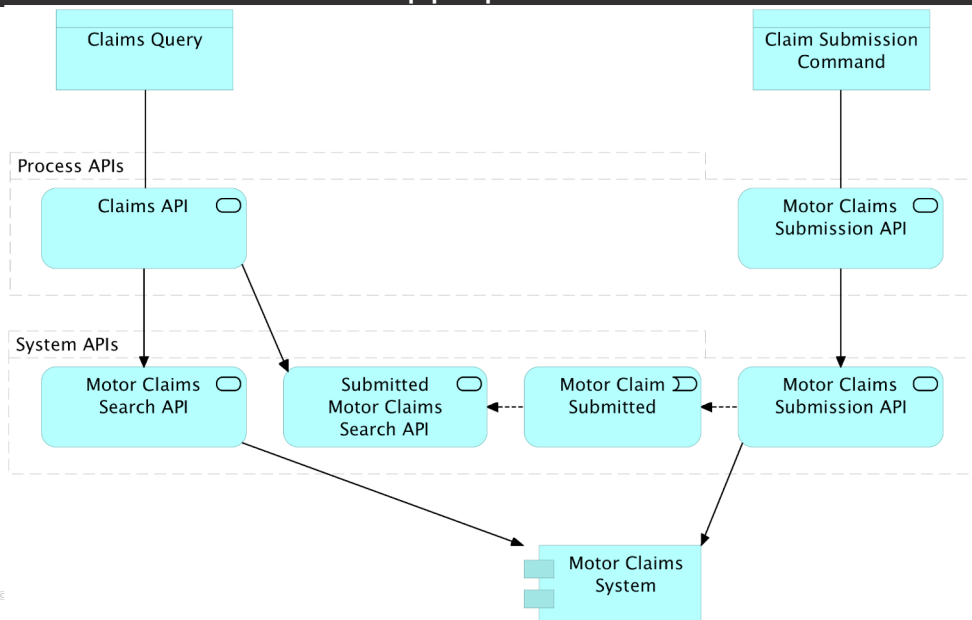
- Multi-tenant cloud-based (hosted) **messaging service**
 - Only in the **MuleSoft-hosted** Anypoint Platform, in runtime plane region
- **Role-based access-control**
- **Token-based** client access control
- **Queues and message exchanges** and bindings between them
 - **Send** to queues or message exchanges, **consume** from queues
 - Queues statically bound to message exchanges
- **Point-to-point, pub/sub, FIFO queues, payload encryption, persistent/durable messages, DLQs, message TTL**
- **REST API and Connector**
 - API invocations to MuleSoft-hosted broker
- **Web-based management console**

All contents © MuleSoft Inc.

15

15

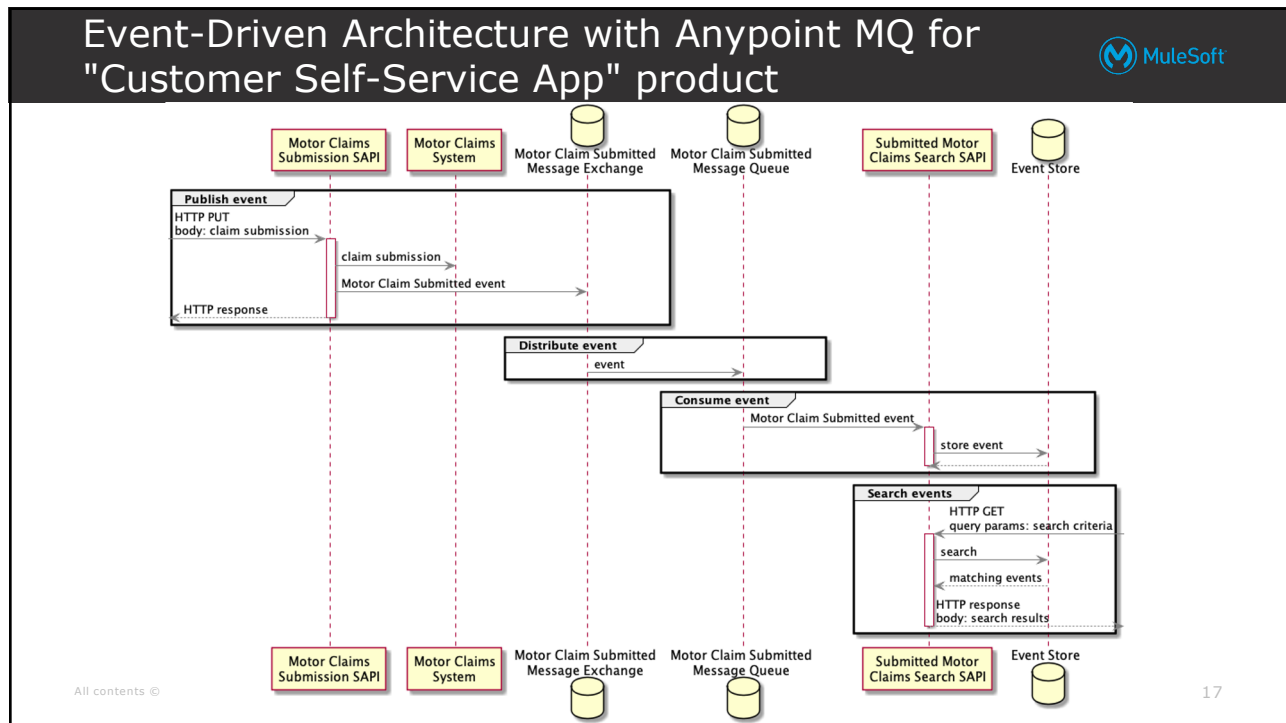
Event-Driven Architecture with Anypoint MQ for "Customer Self-Service App" product



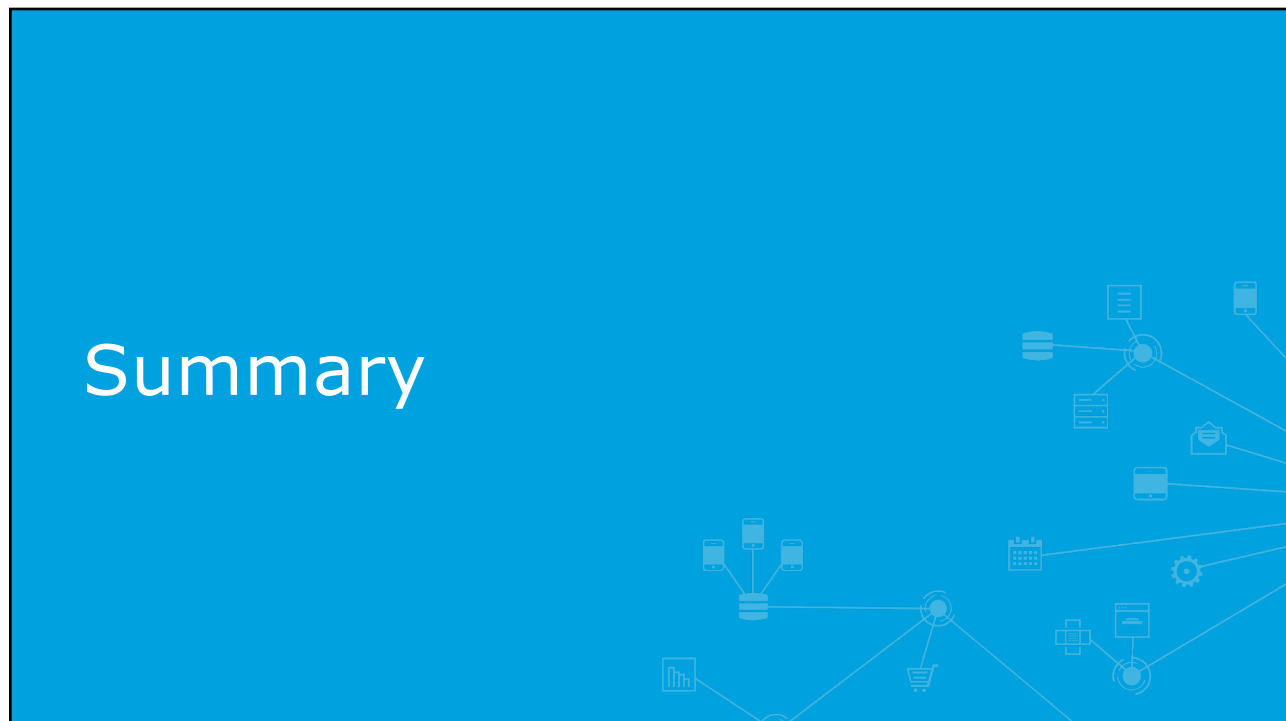
All contents ©

16

16



17



18

Summary



- Some NFRs best realized by **adding EDA** to API-led connectivity
- **Events** describe historical facts, are exchanged asynchronously between application components via destinations
- **Event exchange patterns** in an application network should follow rules of API-led connectivity
- **Anypoint MQ** is MuleSoft-hosted multi-tenant cloud-native messaging service
- **Consistency requirement of "Customer Self-Service App"** realized by introducing new System API that consumes events published by "Motor Claims Submission SAPI" without changing existing APIs

All contents © MuleSoft Inc.

19