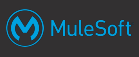
Anypoint Platform Architecture: Application Networks

Module 8
Augmenting API-Led Connectivity With Elements From Event-Driven Architecture

Objectives



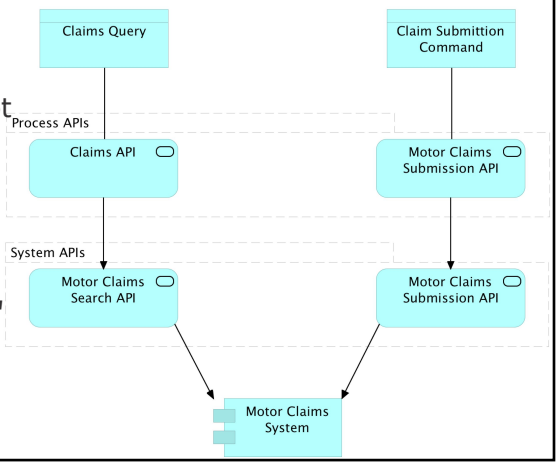
- Selectively choose elements of **Event-Driven Architecture** in addition to API-led connectivity
- Make effective use of **events and message destinations**
- Impose **event exchange patterns** in accordance with API-led connectivity
- Describe **Anypoint MQ** and its features
- Apply Event-Driven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

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

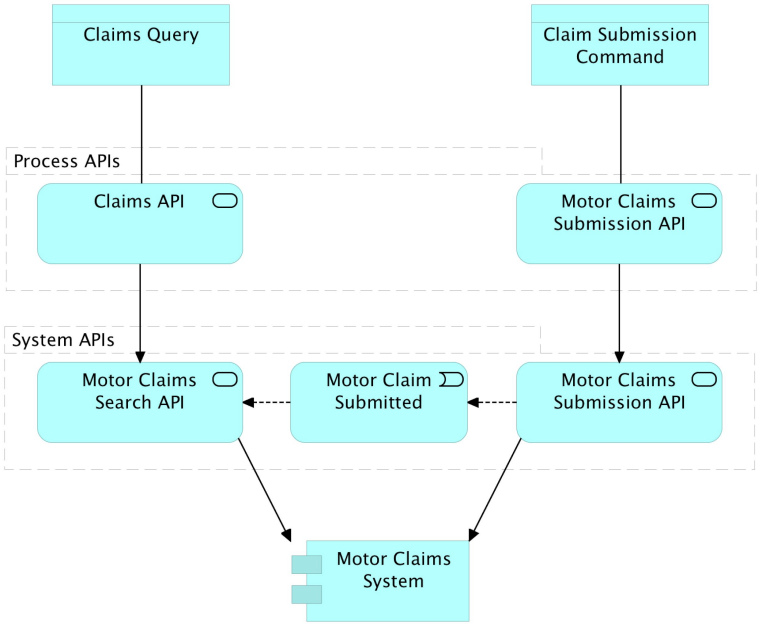
Revisiting the NFRs for the "Customer Self-Service App" product

MuleSoft

- "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



Exchanging events between API implementations



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

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**

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?

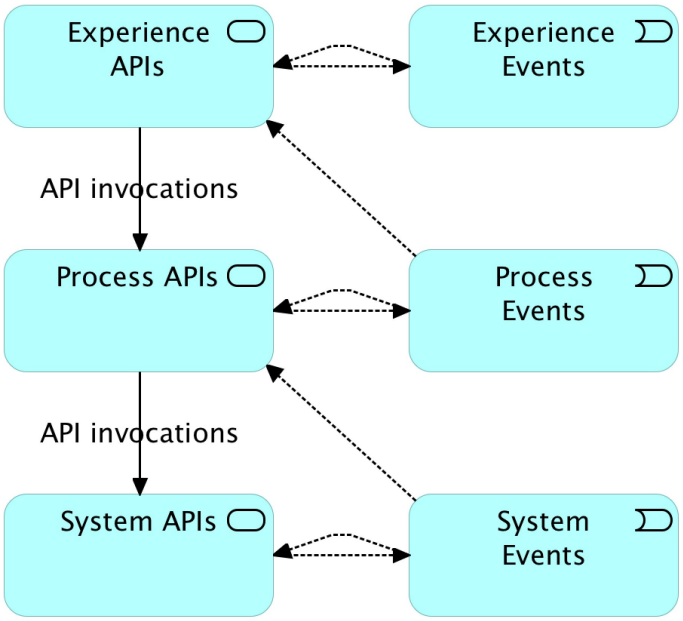
Comparing events and APIs

- **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**: RAML definition vs destination + event type

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

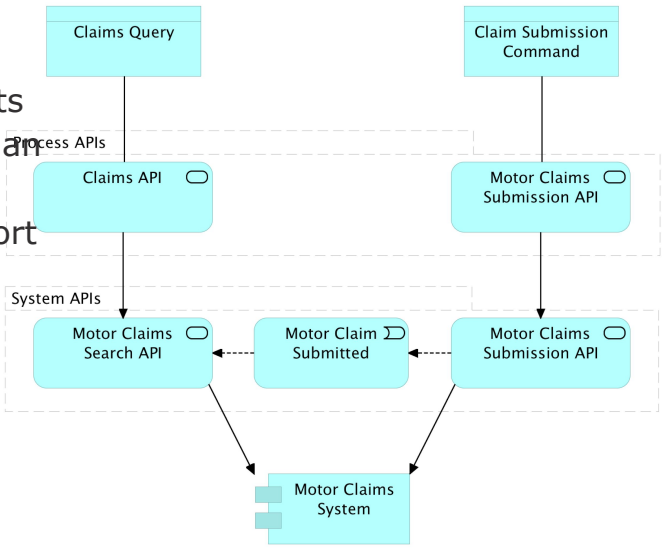
Event exchange patterns in API-led connectivity



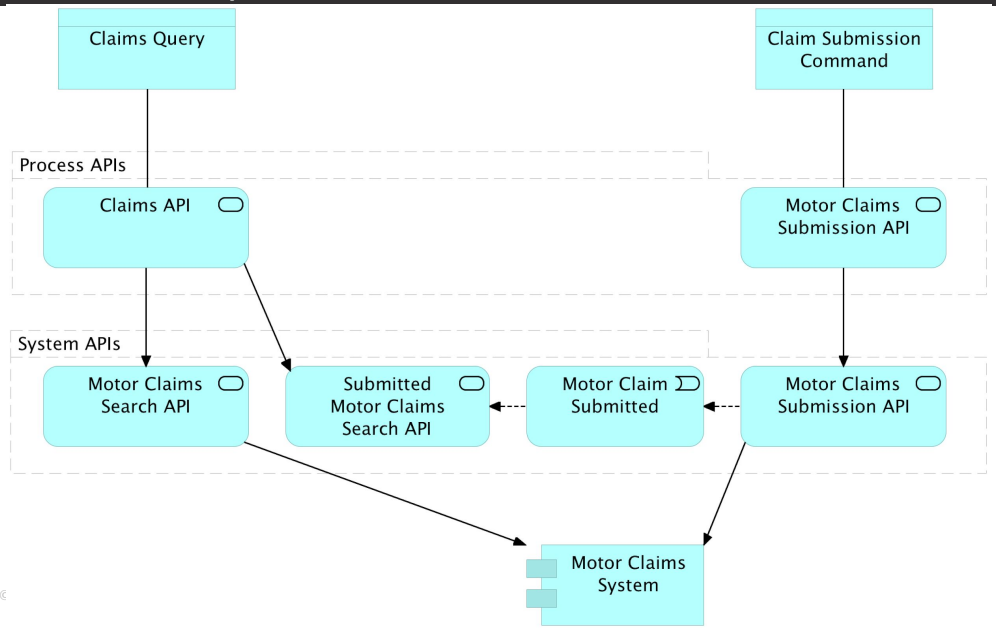
A slow-changing component must never depend on a fast-changing component

Exercise: EDA and API-led connectivity for "Customer Self-Service App" product

1. Decide precise meaning of "Motor Claim Submitted" events
2. Which API should best publish an event with this meaning?
3. Assess the implementation effort of publishing and consuming events in this scenario
4. Redesign solution to honor Single Responsibility Principle and event exchange patterns compatible with API-led connectivity



Separating concerns when exchanging events between API implementations

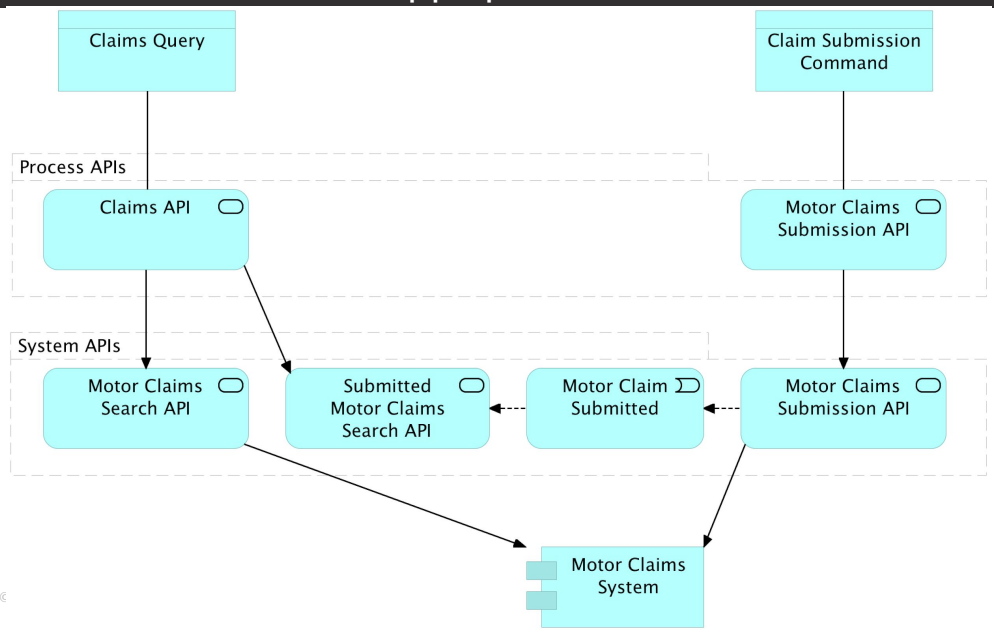


Adding support for asynchronous messaging to the Technology Architecture

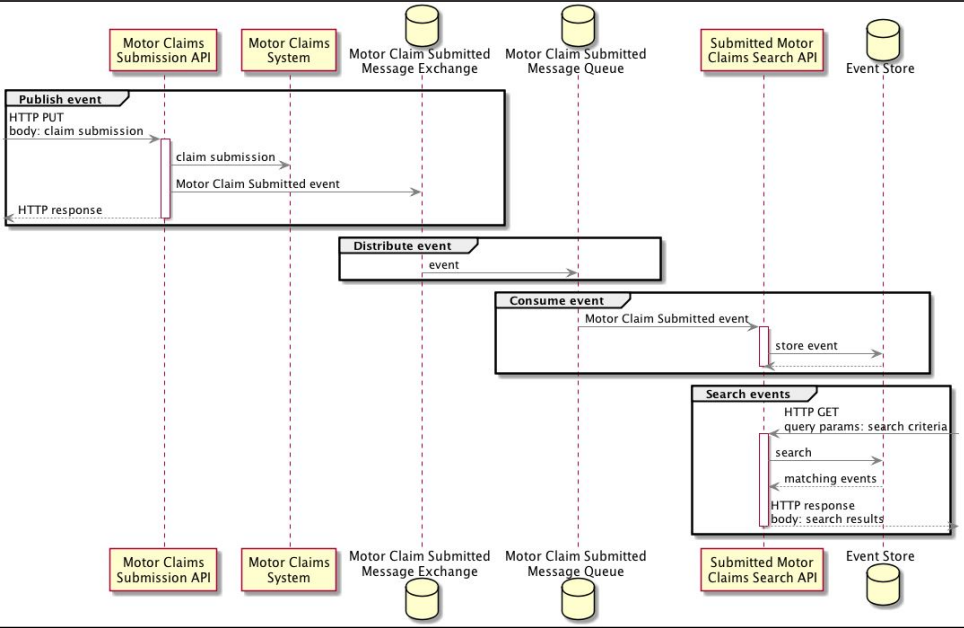
Introducing Anypoint MQ

- 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**

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



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



Summary

Summary

MuleSoft

- 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