

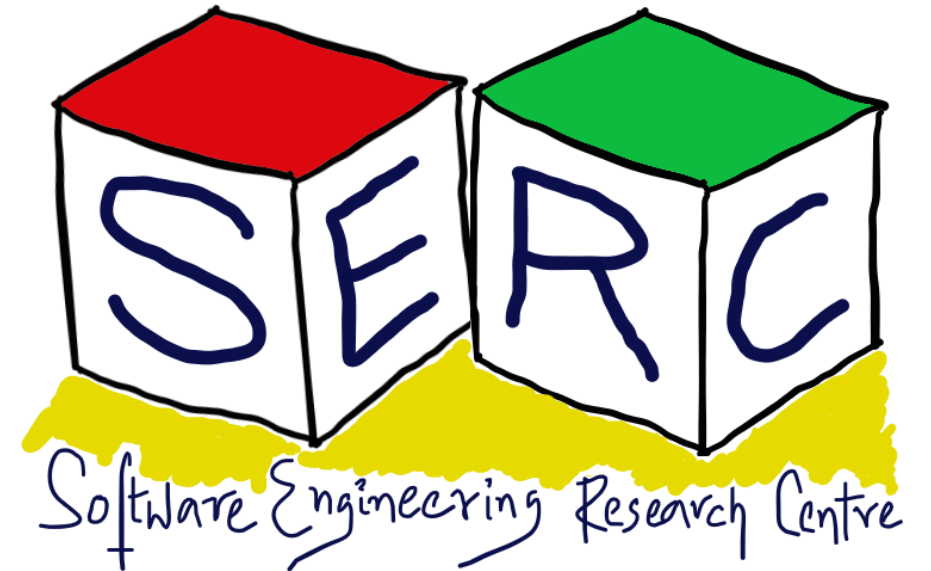
Event Driven Architectures

CS6.401 Software Engineering

Dr. Karthik Vaidhyanathan

karthik.vaidhyanathan@iiit.ac.in

<https://karthikvaidhyanathan.com>



Acknowledgements

The materials used in this presentation have been gathered/adapted/generate from various sources as well as based on my own experiences and knowledge -- Karthik Vaidhyanathan

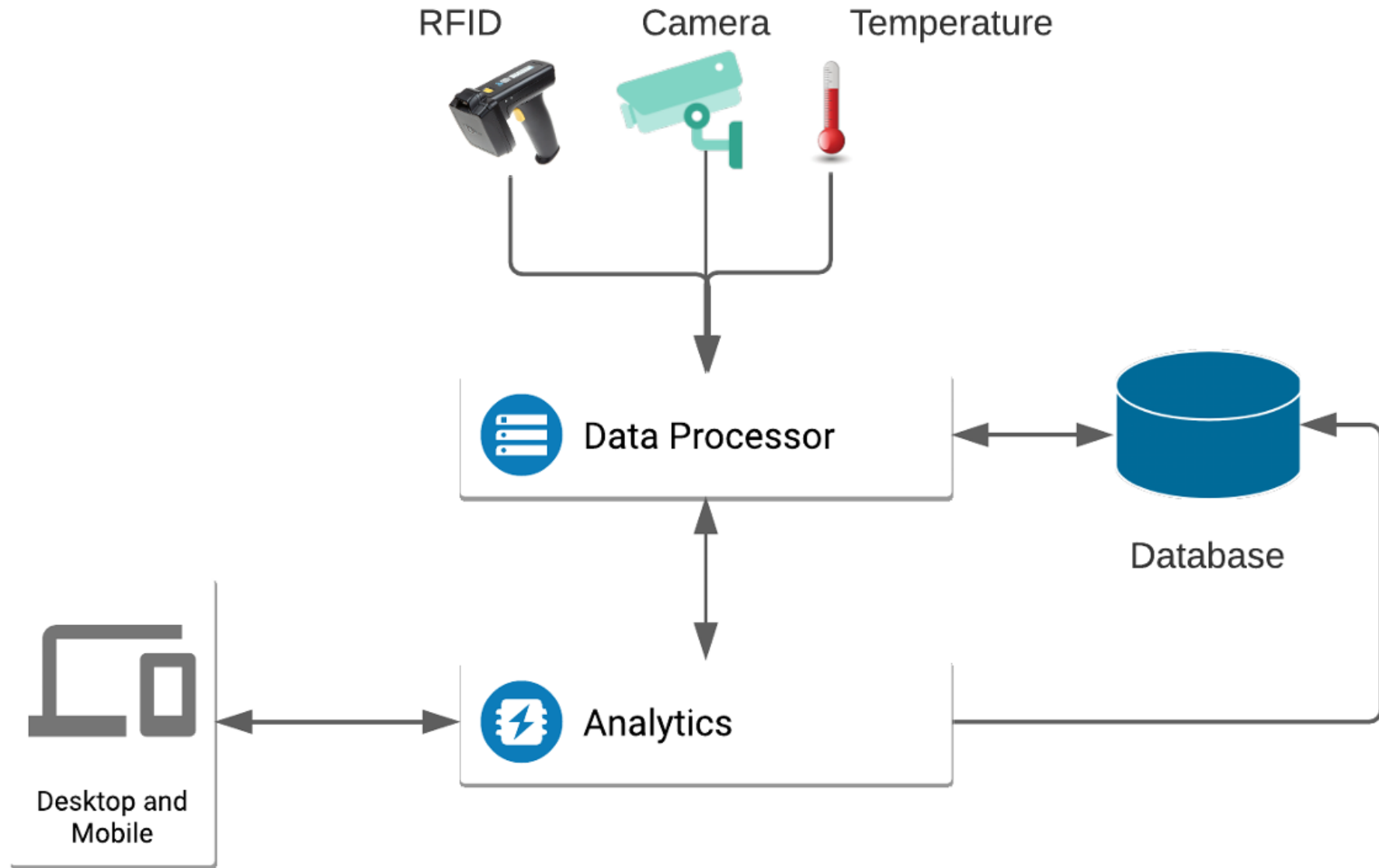
Sources:

1. Software Architecture Patterns, Oreilly
2. Various sources from the web that has been duly credited in the respective slide



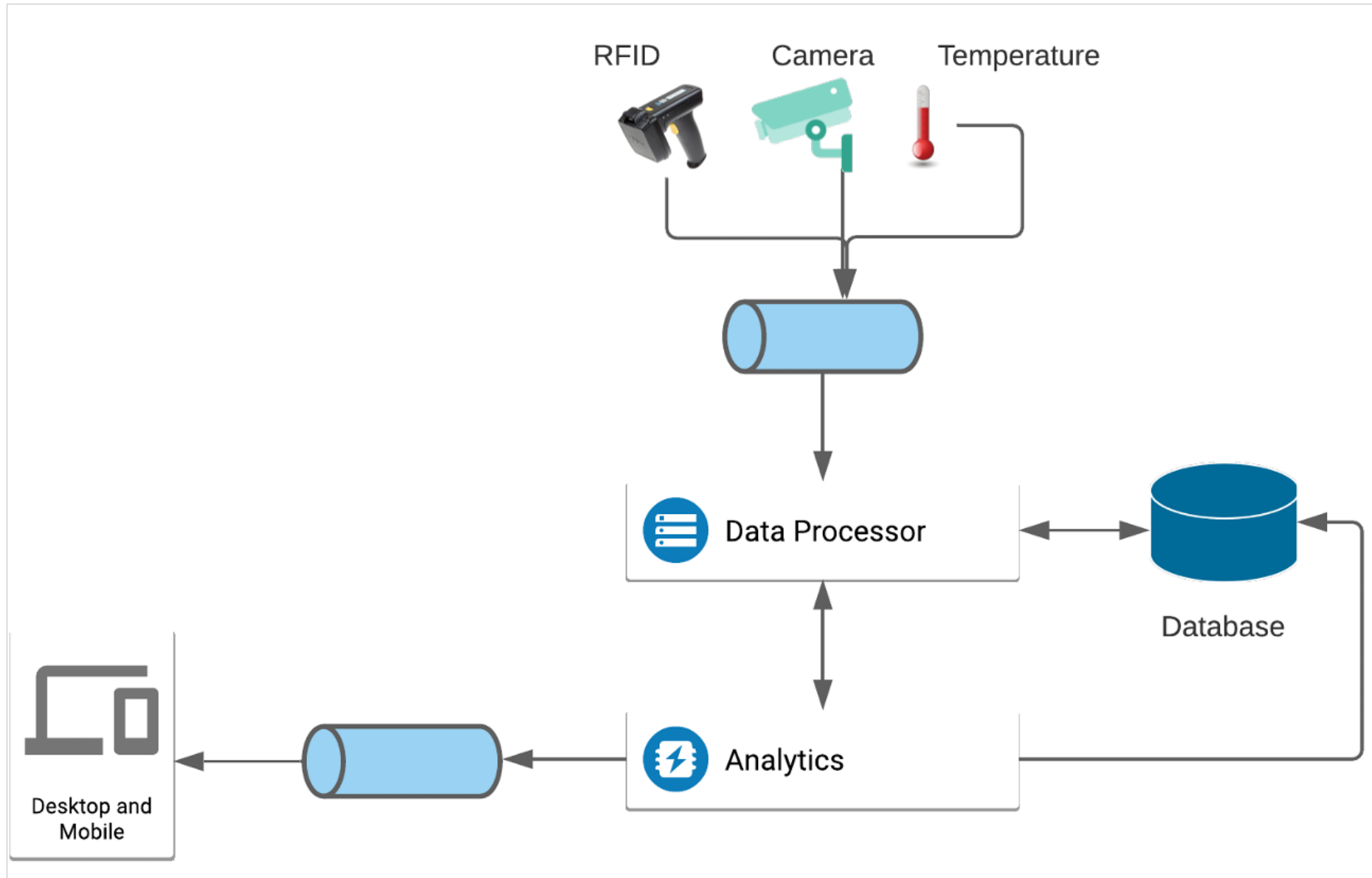
Events?

An Intuition



How good is the above design?

Add Pub/sub components





Event Driven Architecture (EDA)

Event-driven Architectures: An Overview

- Independent components asynchronously emit and receive events communicated over event buses
- Produce, detect and consume events
- Highly decoupled components – Minimal amount of coupling (topics, queue names, etc.)

Design elements

- Components: concurrent event generators and event consumers
- Connectors: event bus (may be more than one)
- Data: events

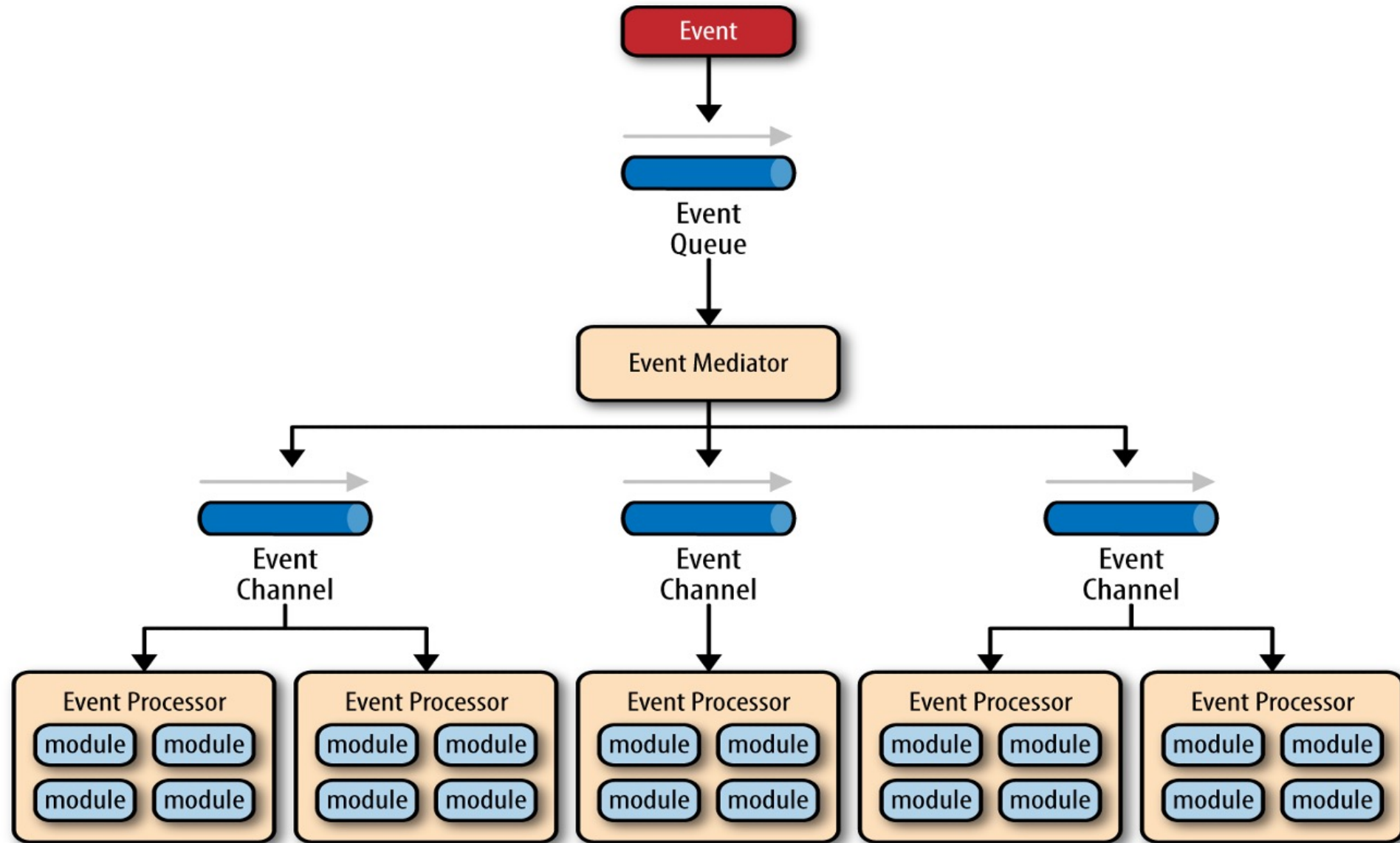
Topology

- Communication via the event bus or link only (Mediator or Broker)

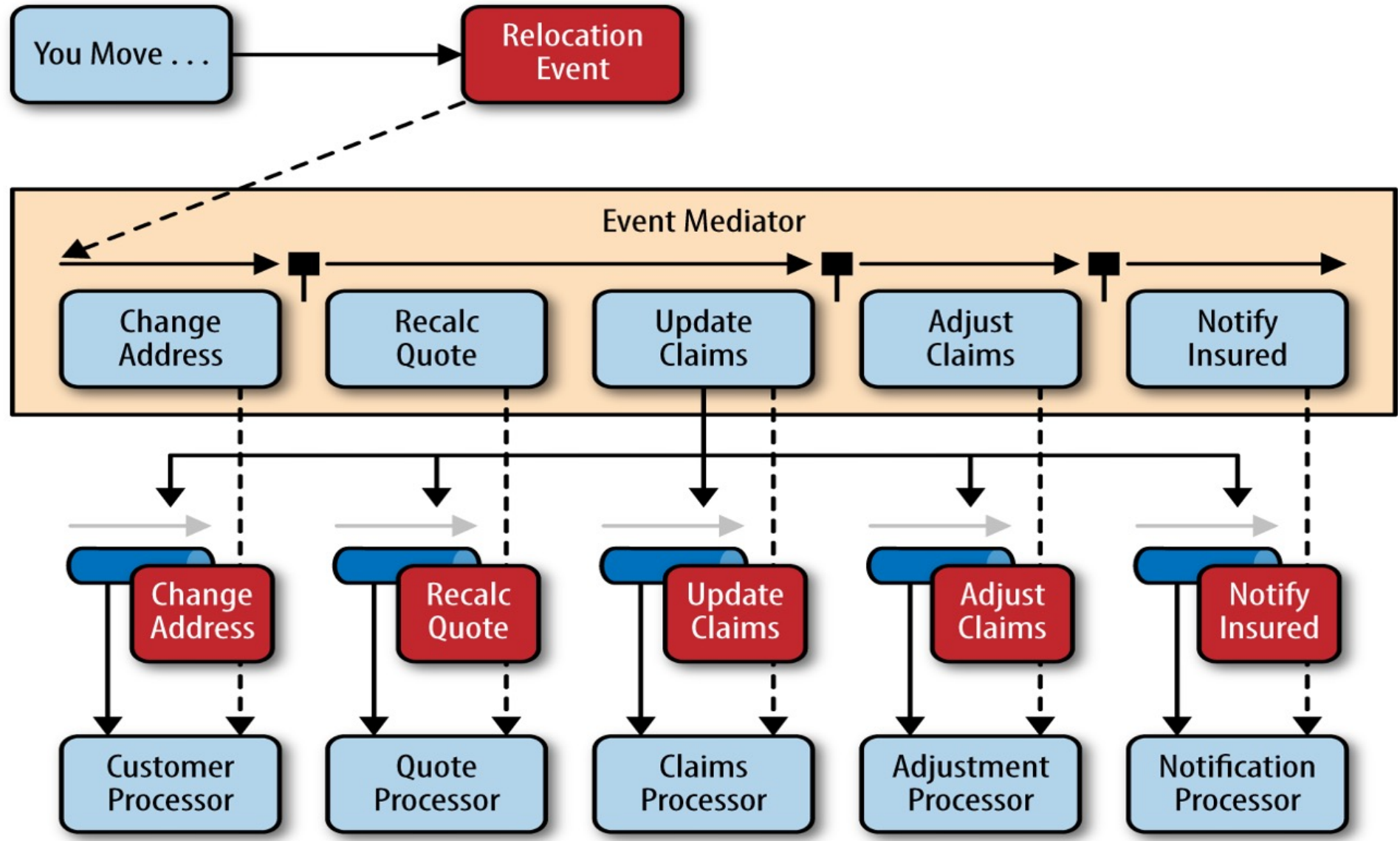


EDA: Mediator Topology

Event-Driven Architectures: Mediator



Event-Driven Architectures: Mediator



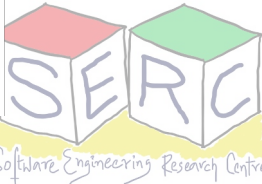
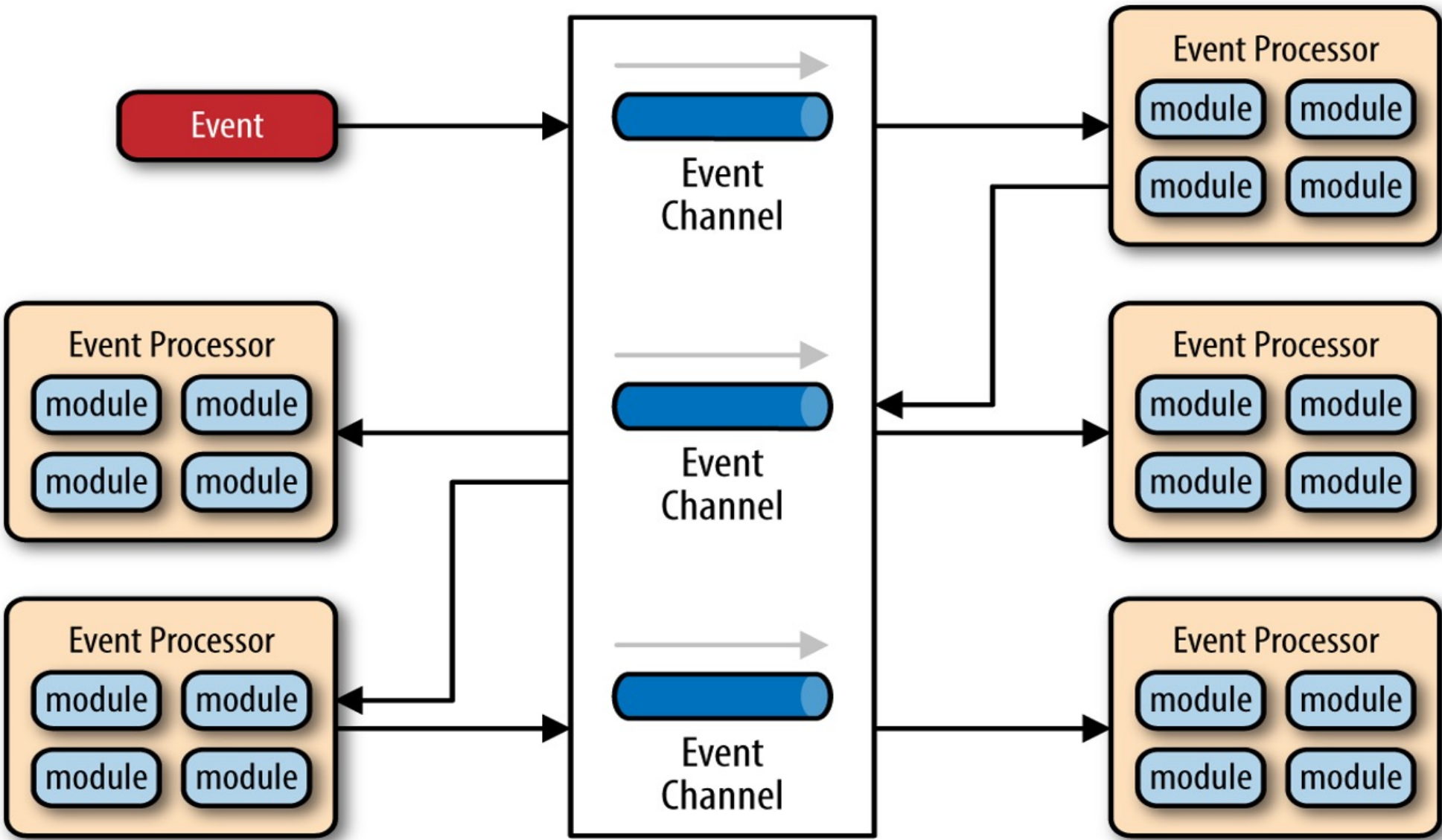
Mediator Topology: An Overview

- Similar to the Orchestration in traditional SOA
- Two key events – Initial and Processing event
- Four main types of components:
 - Event queue – Responsible to transfer events to event mediator
 - Event Mediator – Orchestrates the processing of events to accomplish the overall functionality
 - Event Channel - Topics or queues to which events are ingested by mediator (eg: Kafka)
 - Event Processor - Implements the business logic
 - Can be fine grained or Coarse grained)
 - Advice: keep it to one functionality

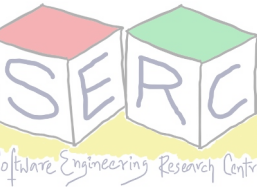
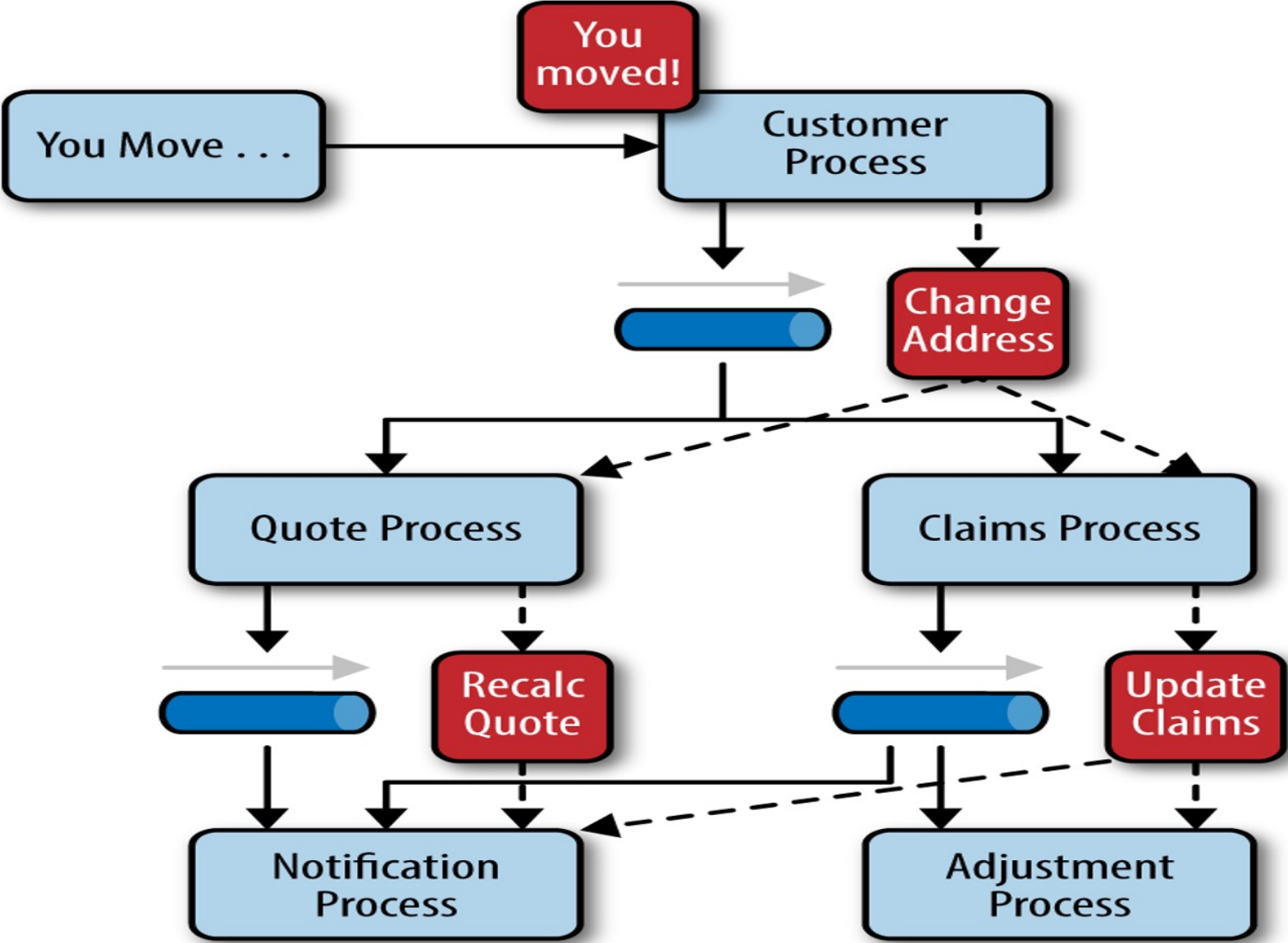


EDA: Broker Topology

Event-Driven Architectures: Broker



Event-Driven Architectures: Broker



Broker Topology: An Overview

- Similar to the Choreography in traditional SOA
- Two main types of components:
 - Broker – Consists of all the event channels for event processing. Can be topics or queues
 - Event Processor – Responsible for processing the event and sending a notification to the event channels



Summarizing

Event-Driven Architectures

Advantages

1. High performance
2. High Scalability
3. Ease of Deployment
4. Ease of modifications/Evolved easily

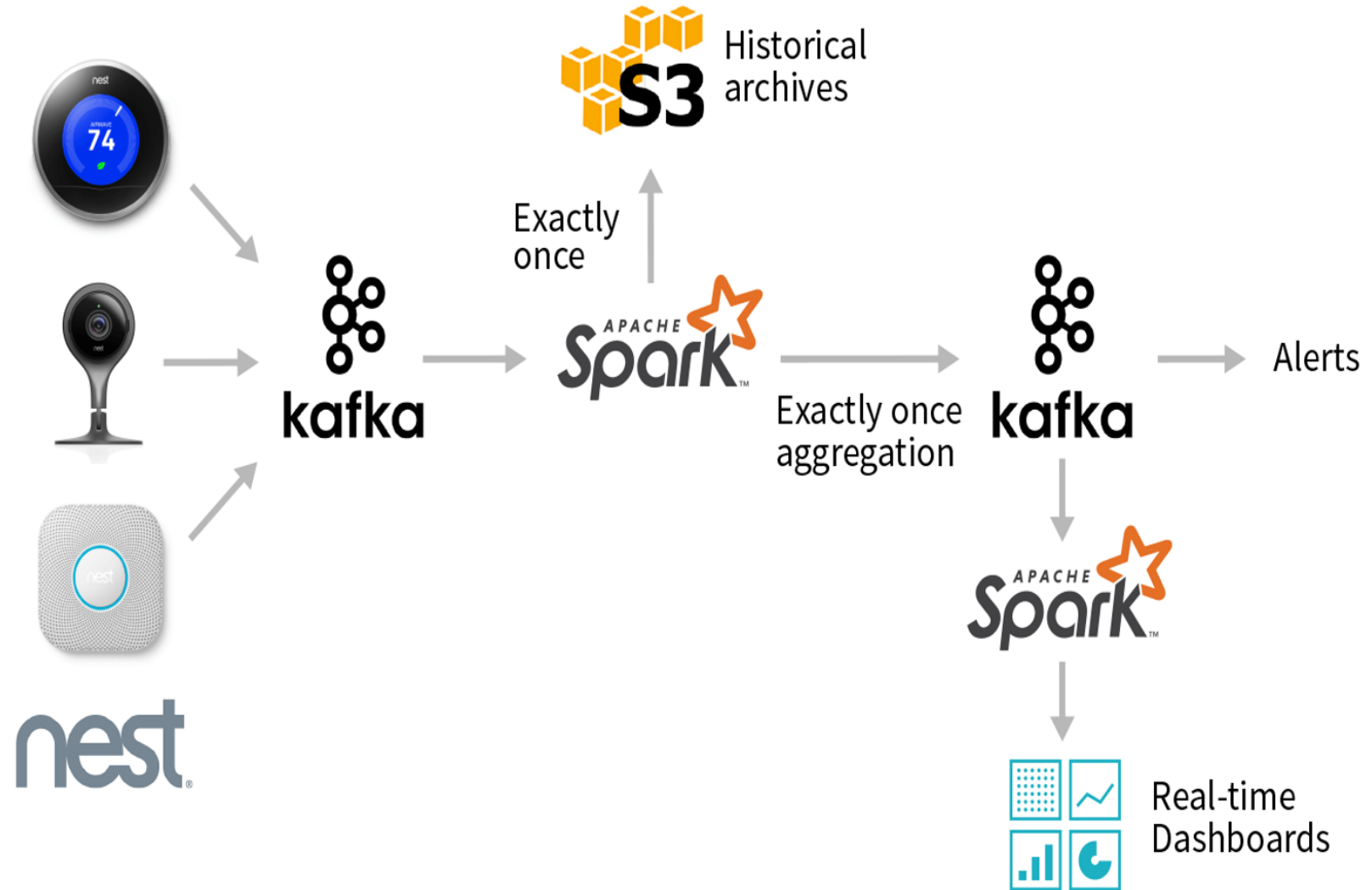
Disadvantages

1. Remote process availability – Liveliness of a consumer
2. Lack of responsiveness
3. Broker or mediator failures
4. Testing can be tedious
5. Development can be complex

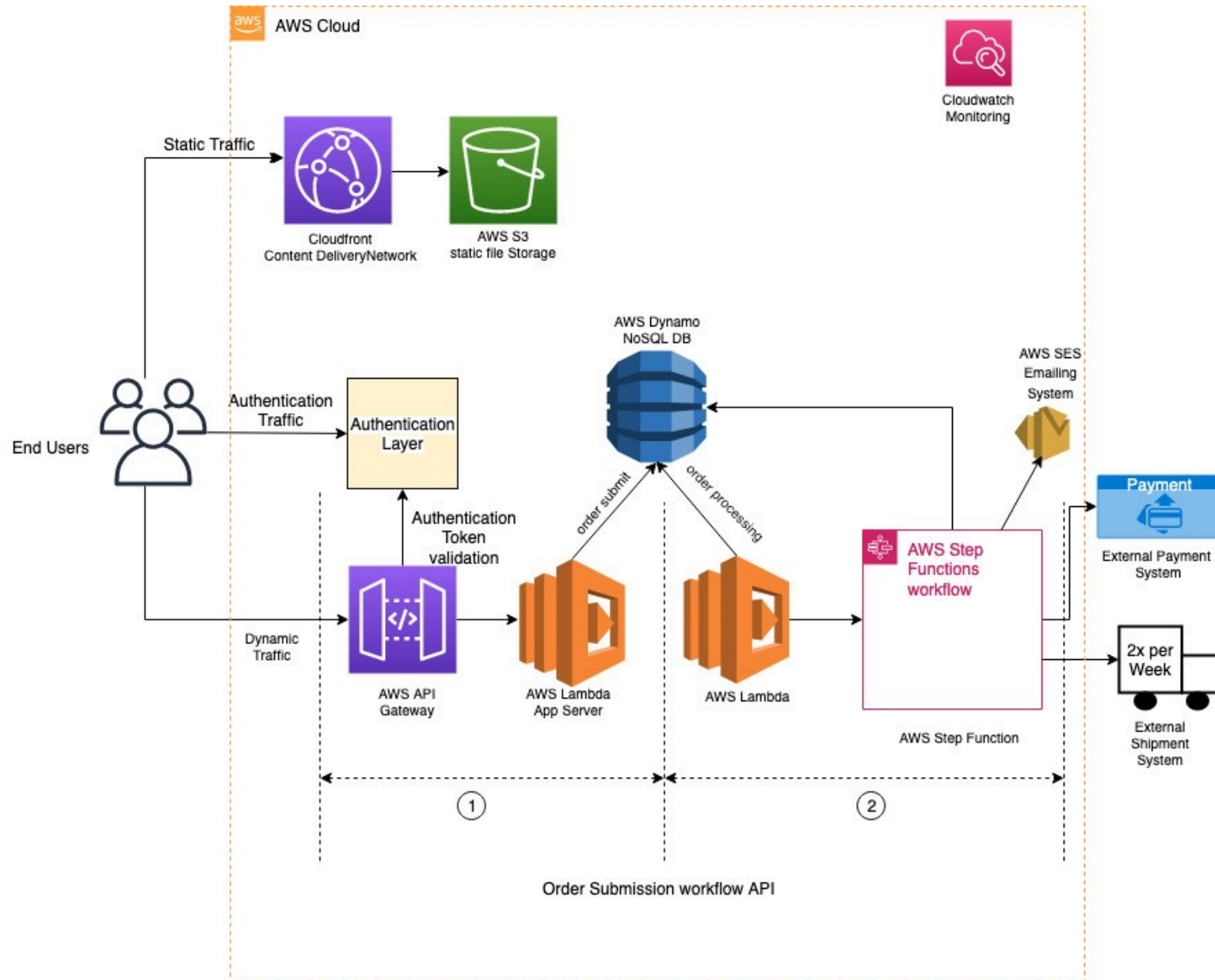


Some Examples

Event-Driven Architectures



Event-Driven Architectures



Thank You



Course website: karthikv1392.github.io/cs6401_se

Email: karthik.vaidhyanathan@iiit.ac.in

Web: <https://karthikvaidhyanathan.com>

Twitter: @karthi_ishere

