



Improving Reliability, Scalability, and Extensibility with Event- Driven Architecture

Scott Carter, Principal Software Engineer

8/8/2022

Agenda

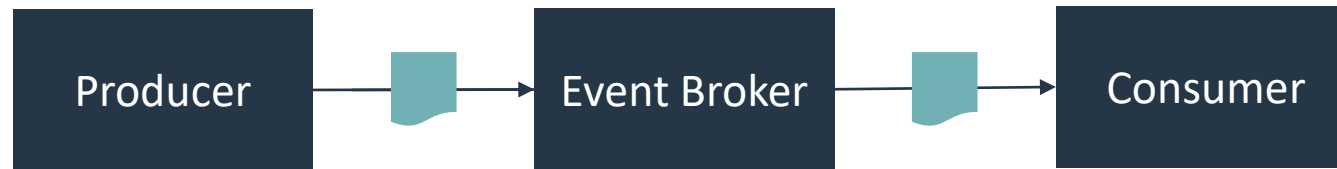
- What is Event-Driven Architecture (EDA)?
 - Events
 - Event Broker
 - Producers
 - Consumers
- Comparison with Request-Based Solutions
- EDA Challenges
- Getting started with EDA

Definition and Components

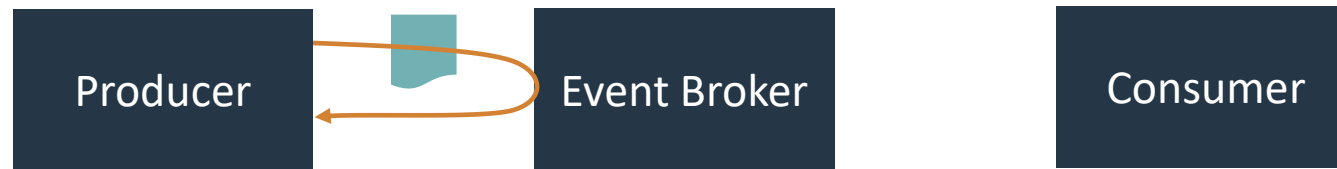
What is Event-Driven Architecture?

System of loosely coupled microservices that exchange information through the production and consumption of events

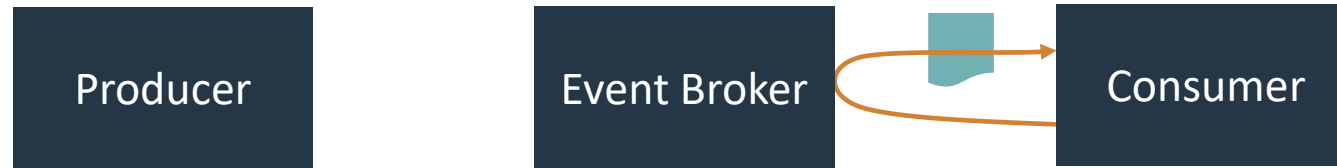
Basic Components



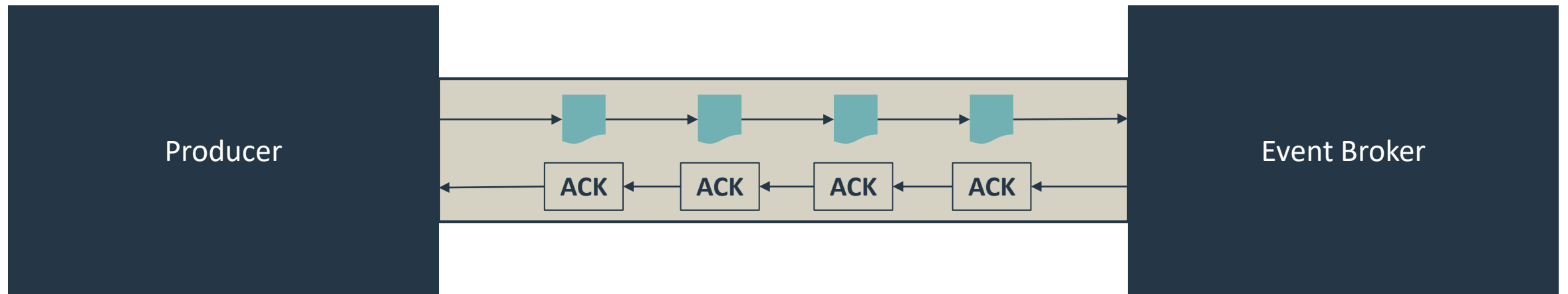
Asynchrony



Asynchrony

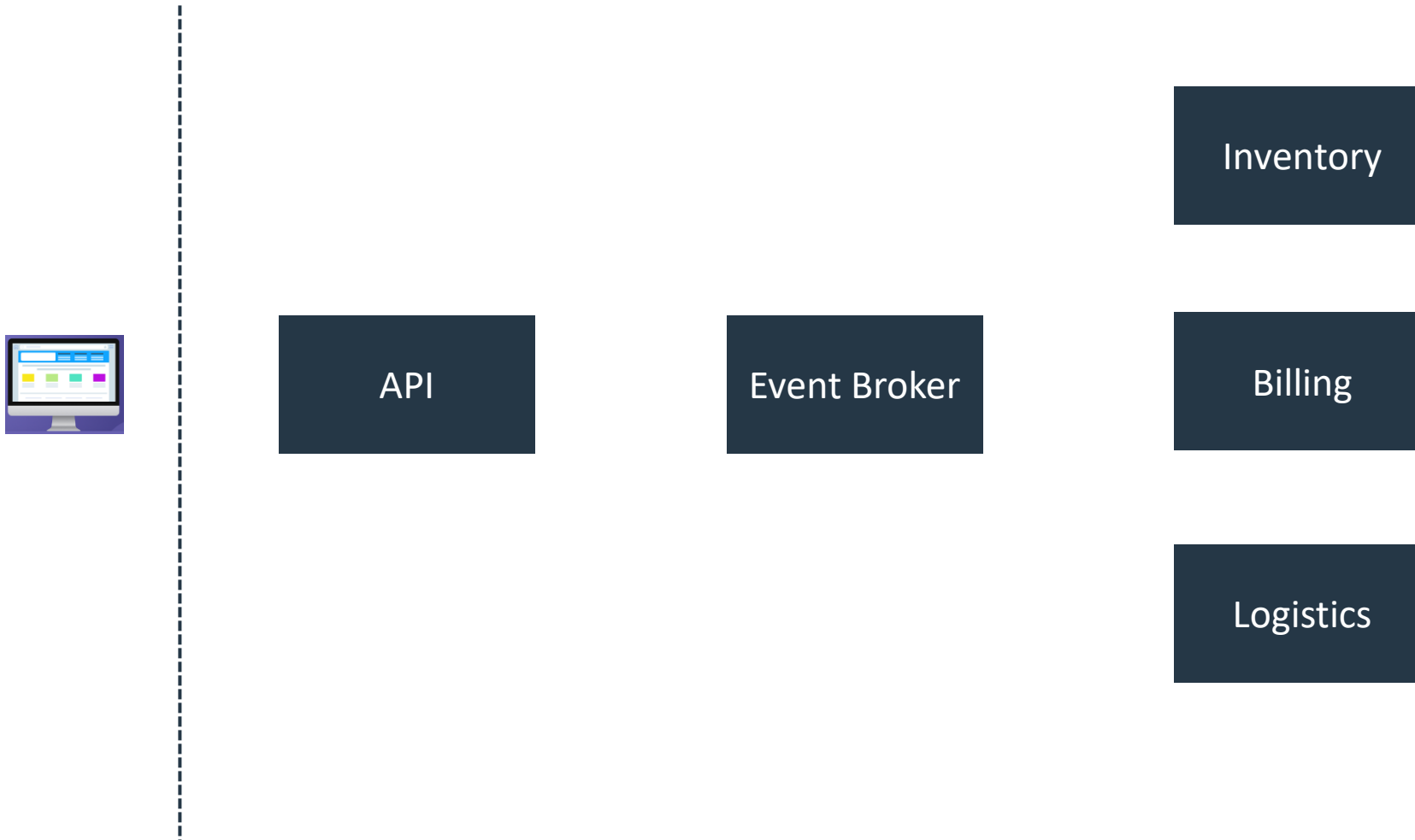


Efficient Communications

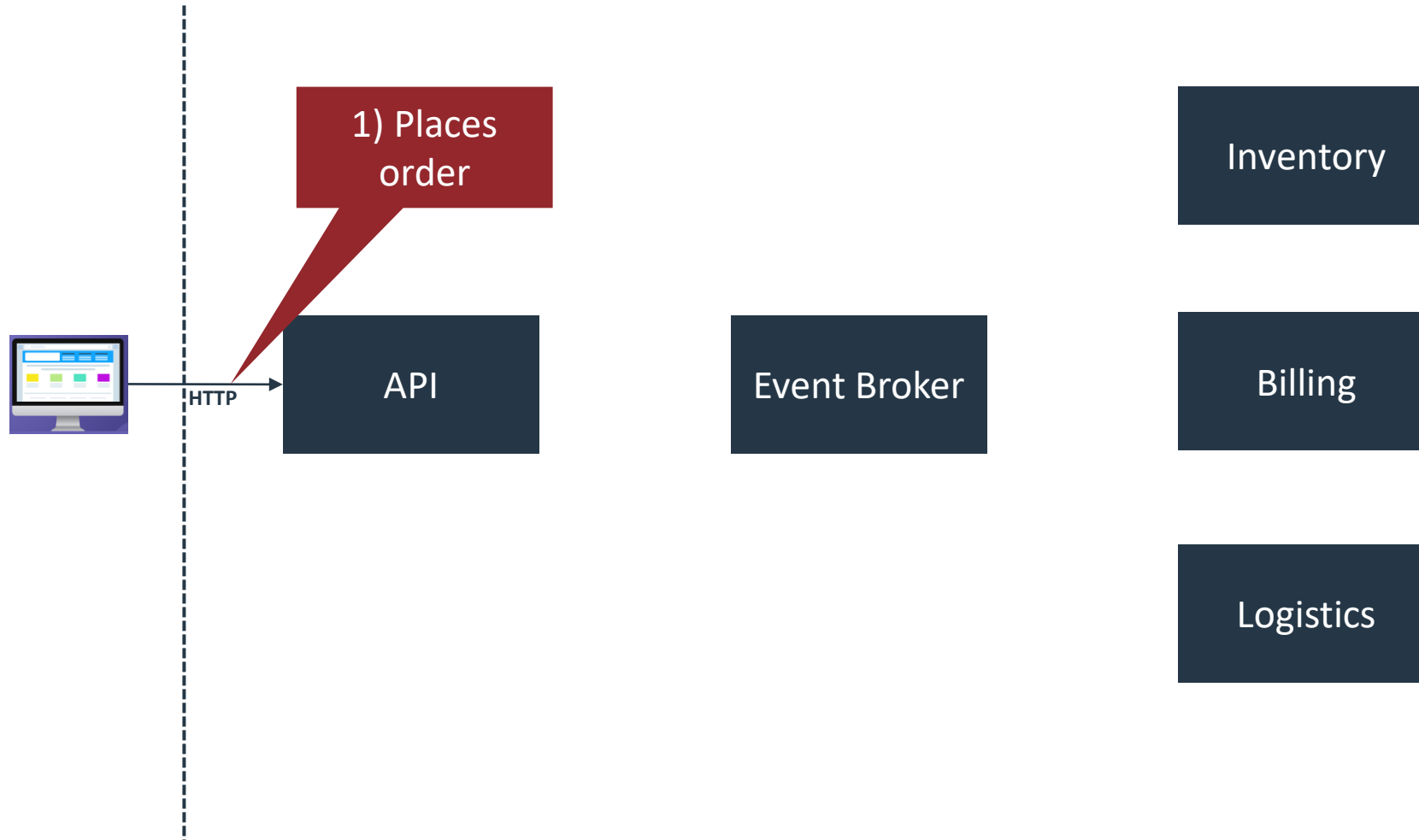


Example

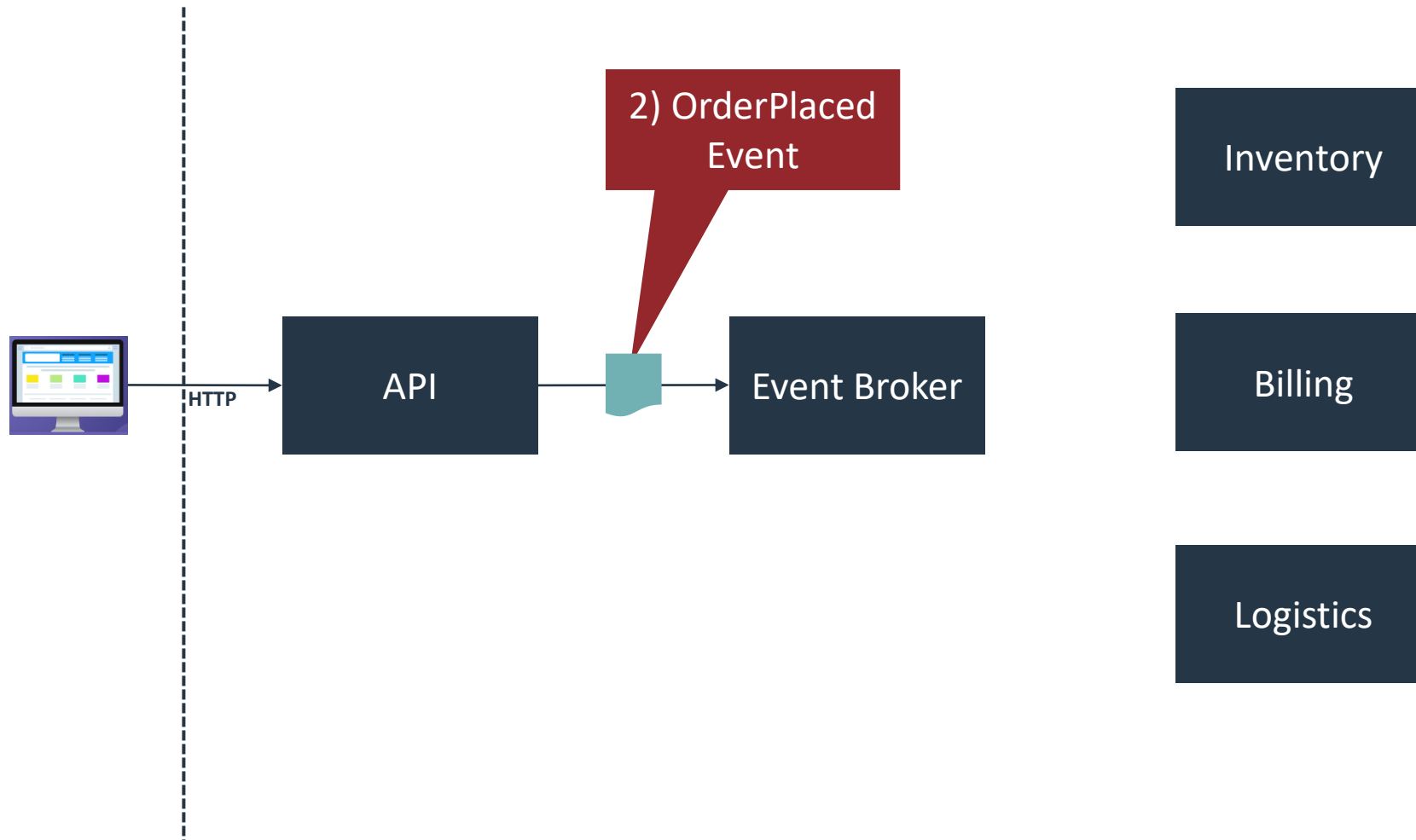
Sample Event-Based Point of Sale System



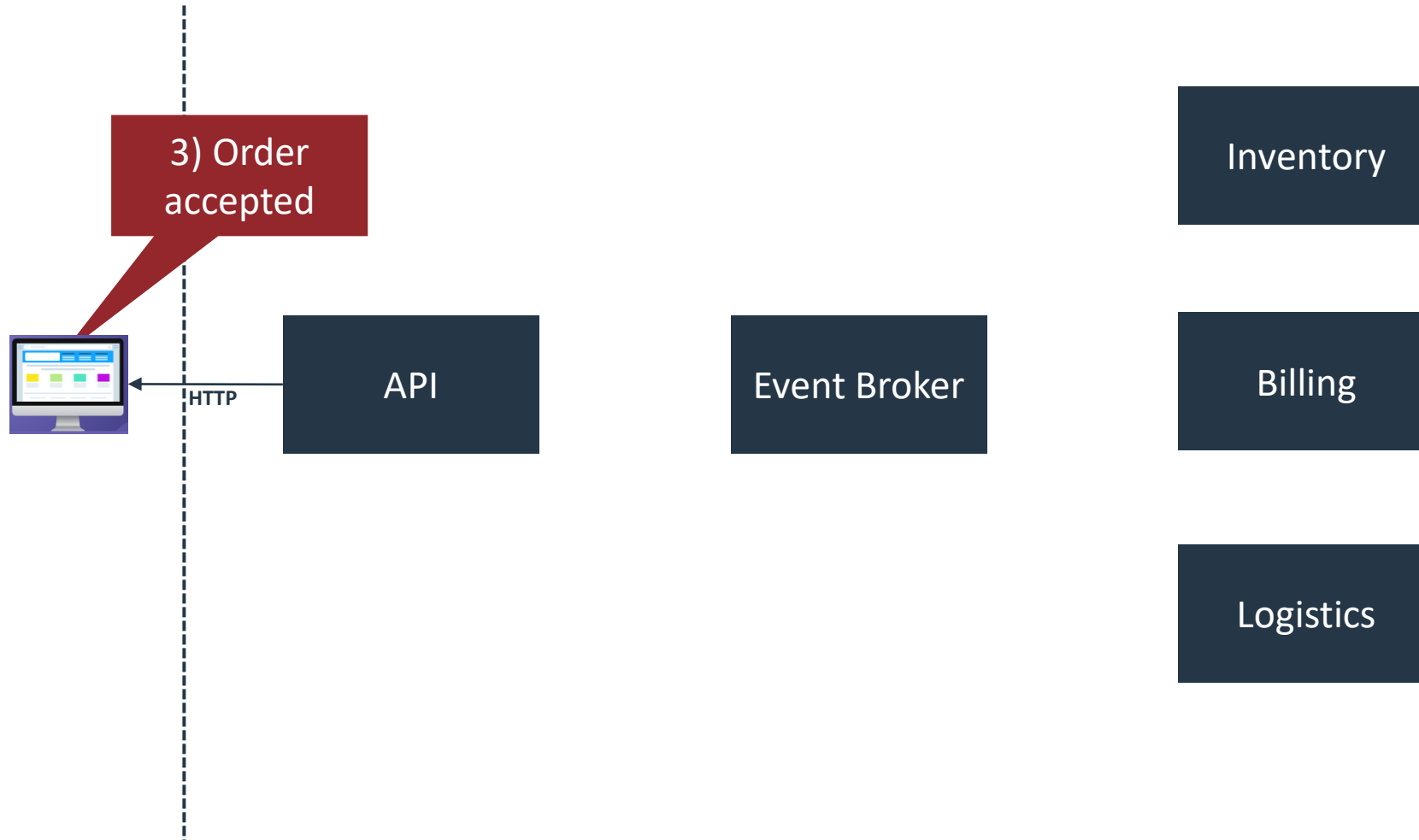
Sample Event-Based Point of Sale System



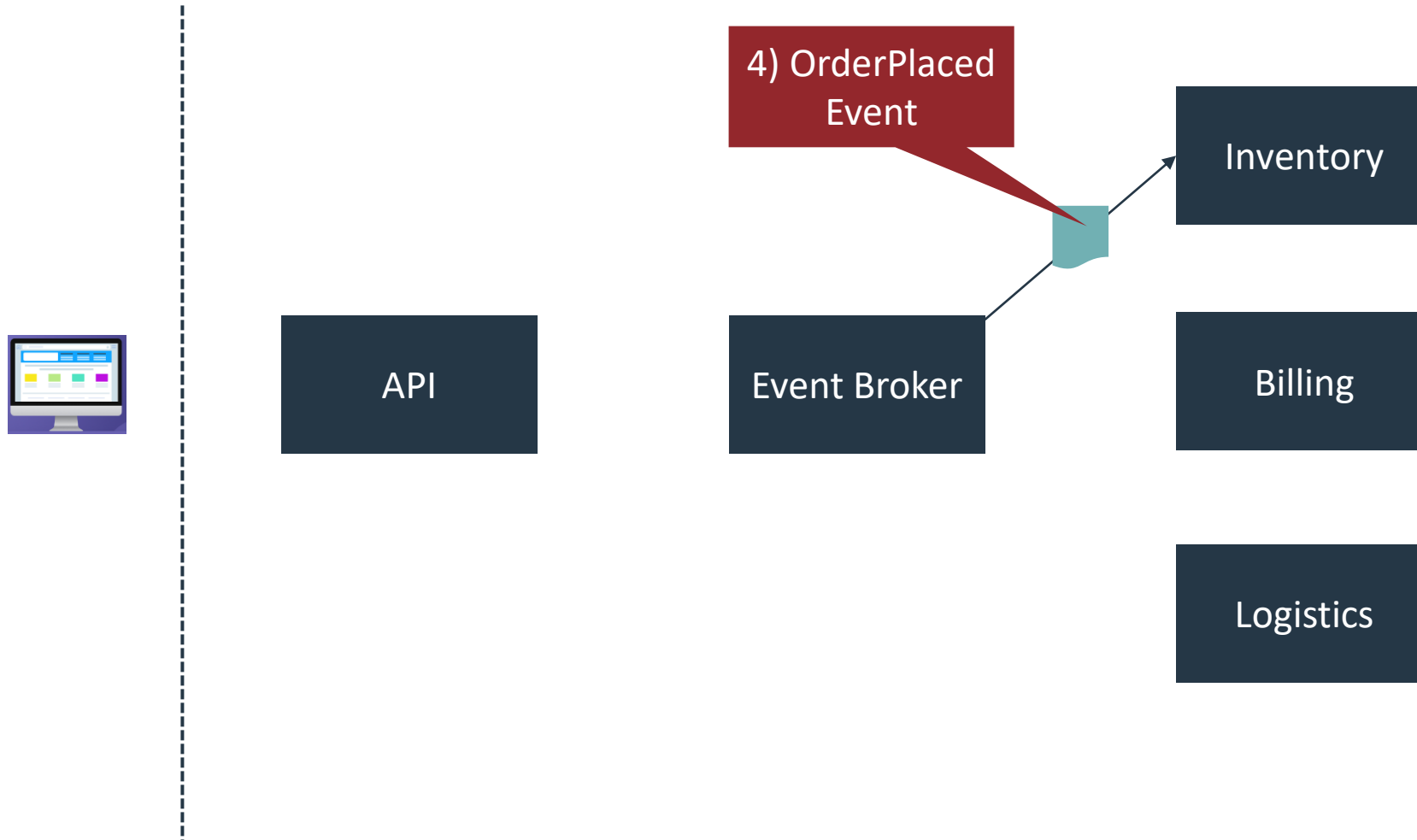
Sample Event-Based Point of Sale System



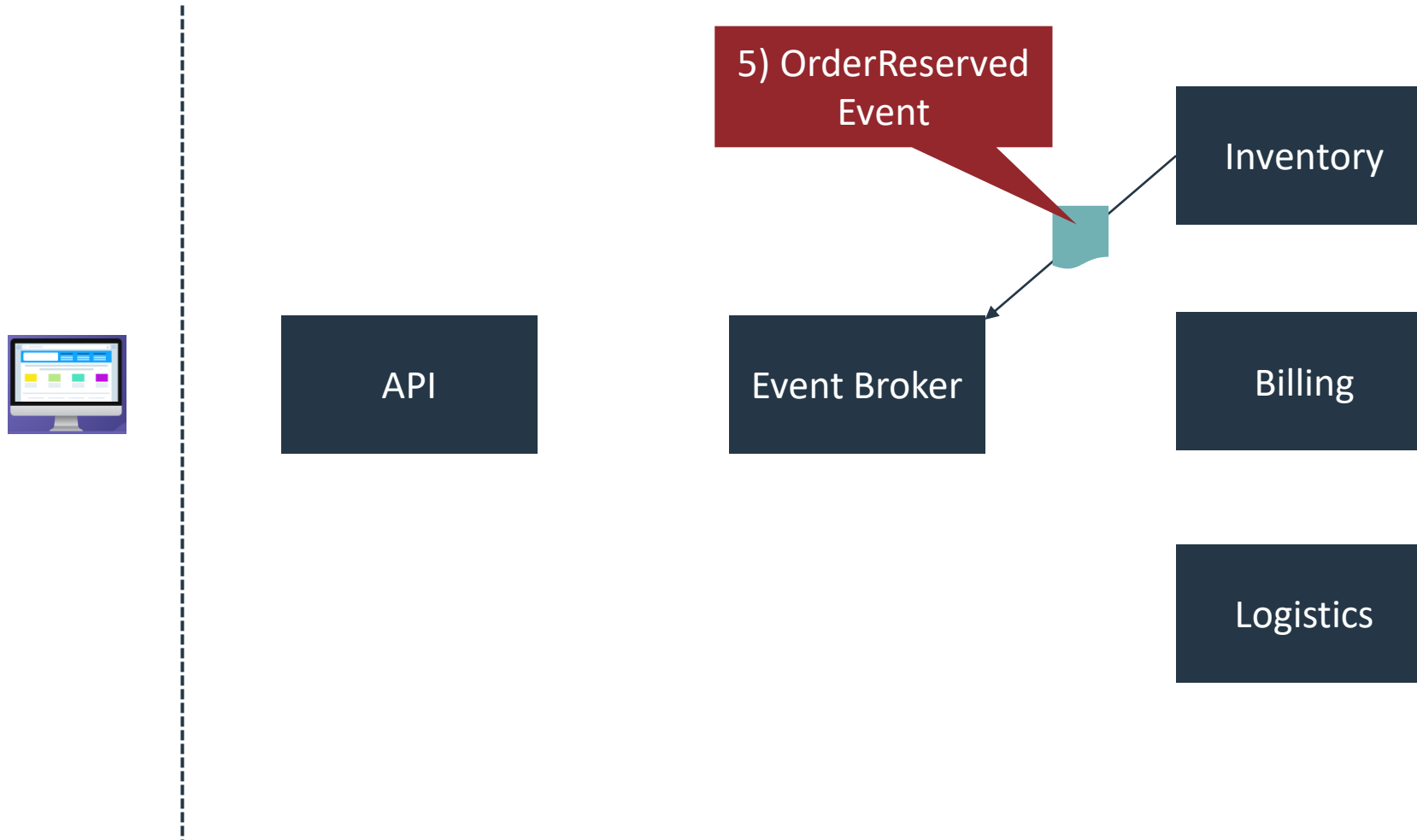
Sample Event-Based Point of Sale System



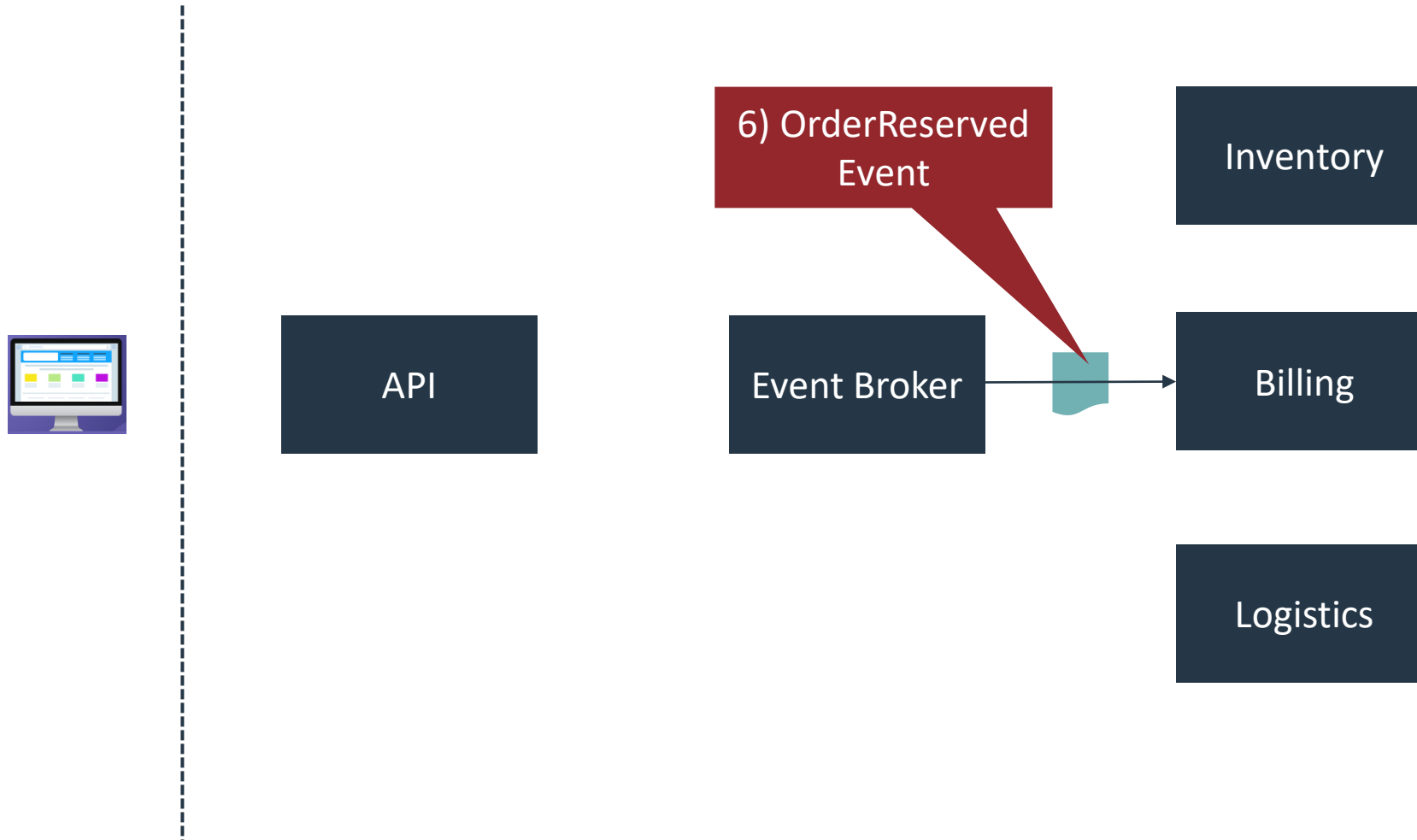
Sample Event-Based Point of Sale System



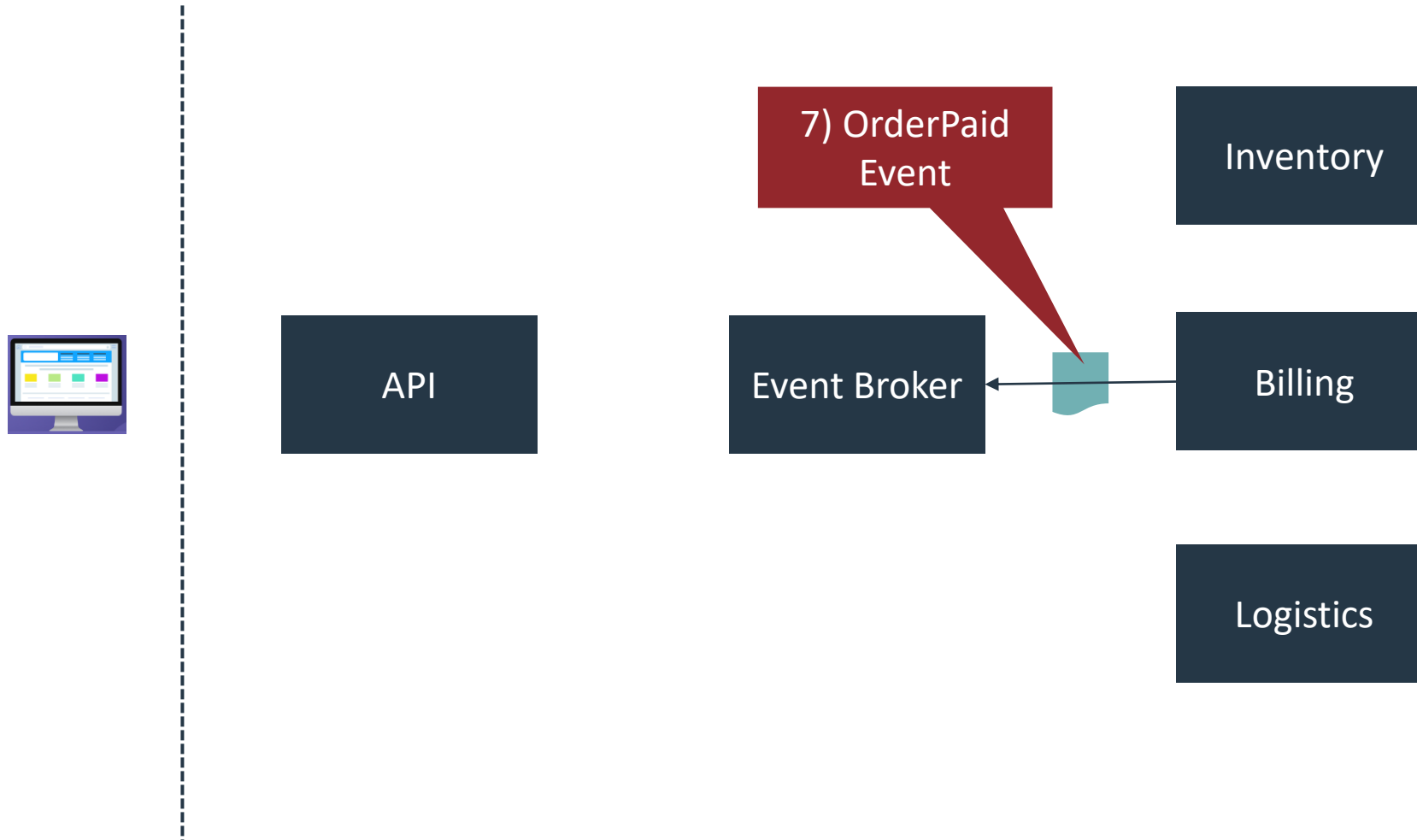
Sample Event-Based Point of Sale System



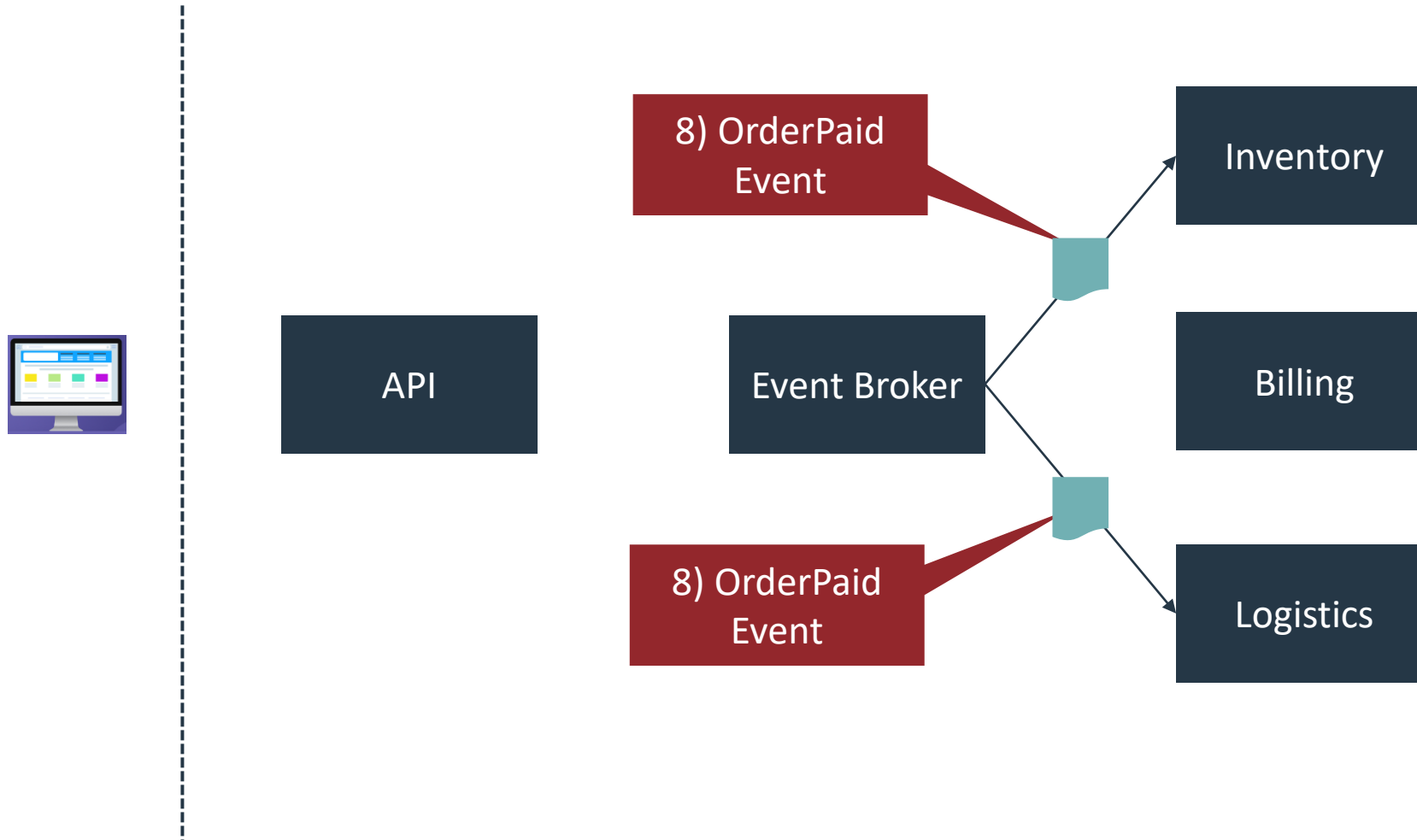
Sample Event-Based Point of Sale System



Sample Event-Based Point of Sale System



Sample Event-Based Point of Sale System

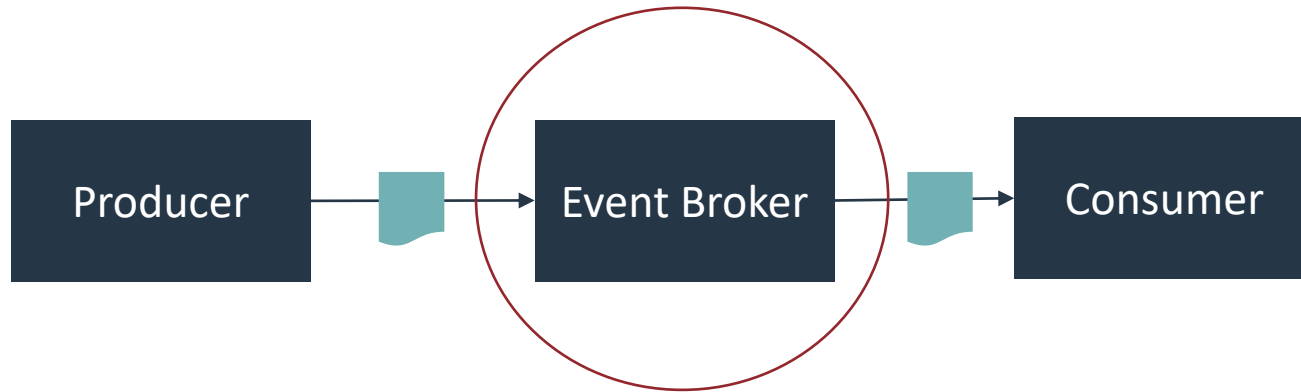


Component Closer Look

Events

```
{  
  "eventName" : "orderPlaced",  
  "orderId": "760b5301-295f-4fec-95f8-6b303a3b824a",  
  "customerId": 28623823,  
  "productId": 31334,  
  "quantity": 1,  
  "paymentMethodId": 2  
  "timestamp": "2021-02-09T11:12:17+0000"  
}
```

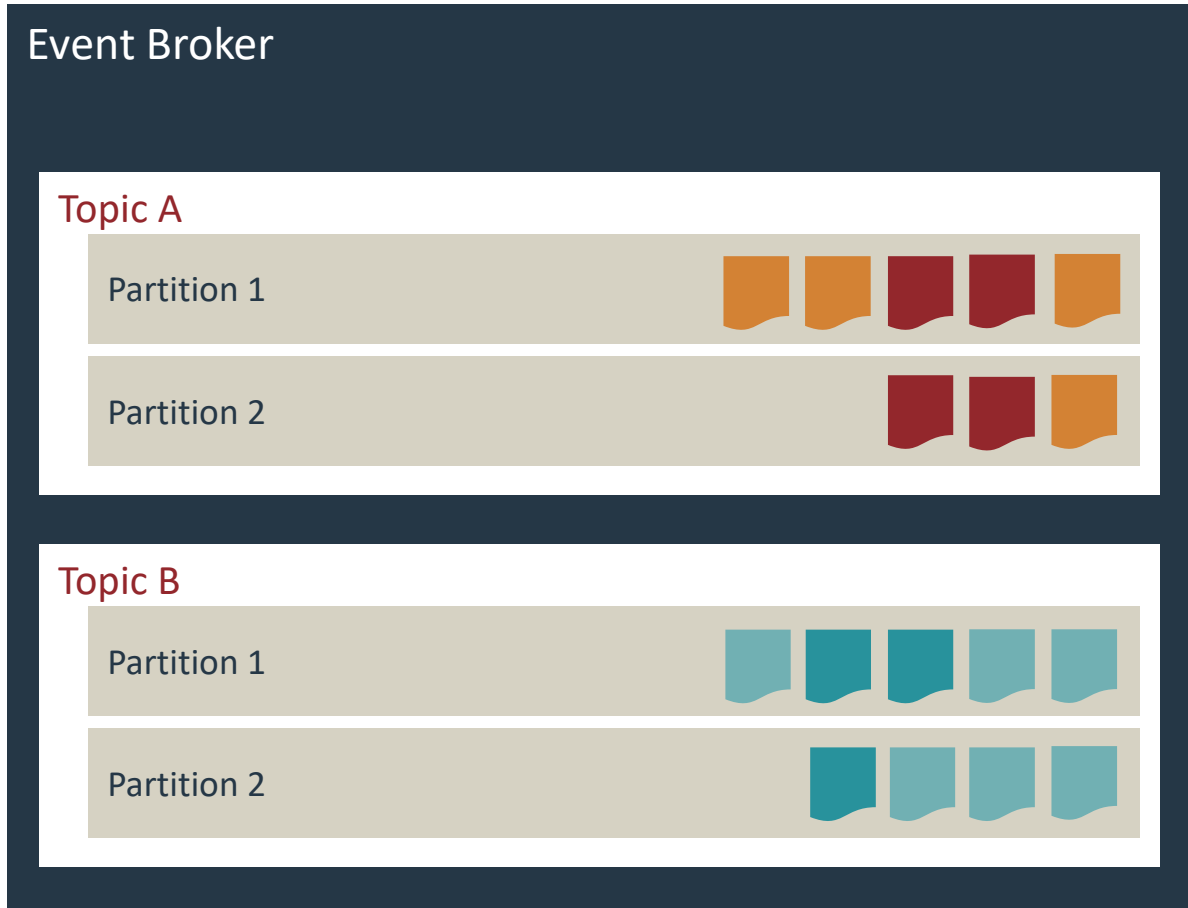
Event Broker



Event Broker Organization



Event Broker Organization



Bank Account Event Stream

Account Topic / Partition 1



Most Recent

Oldest

Bank Account Event Stream

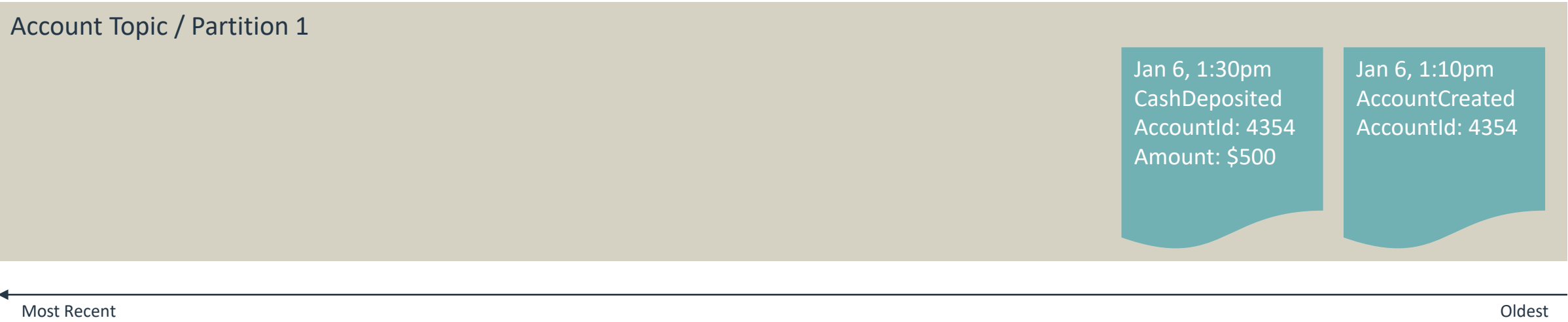
Account Topic / Partition 1

Jan 6, 1:10pm
AccountCreated
AccountId: 4354

←
Most Recent

Oldest →

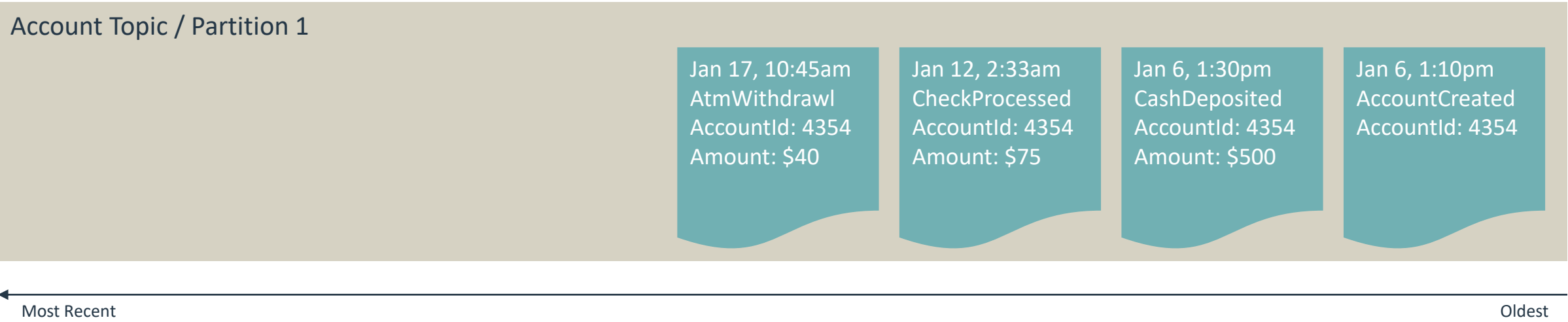
Bank Account Event Stream



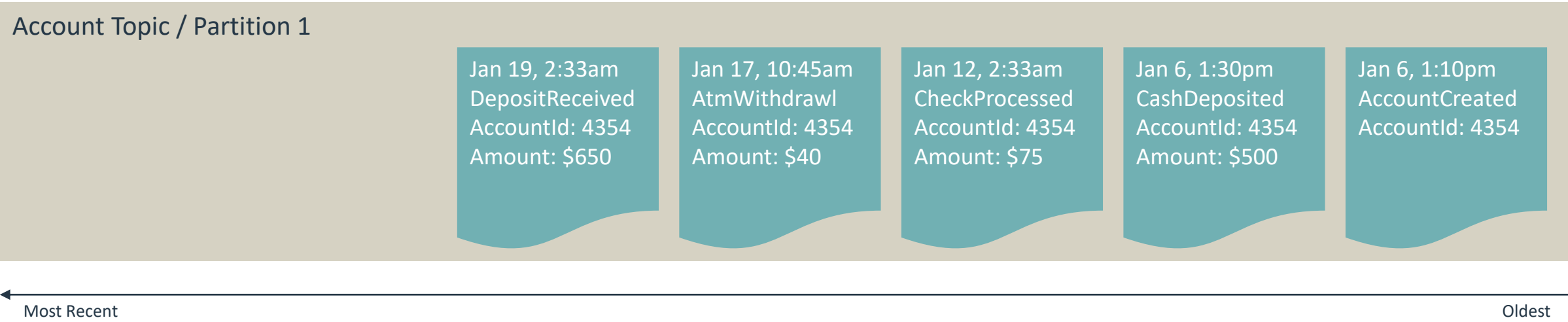
Bank Account Event Stream



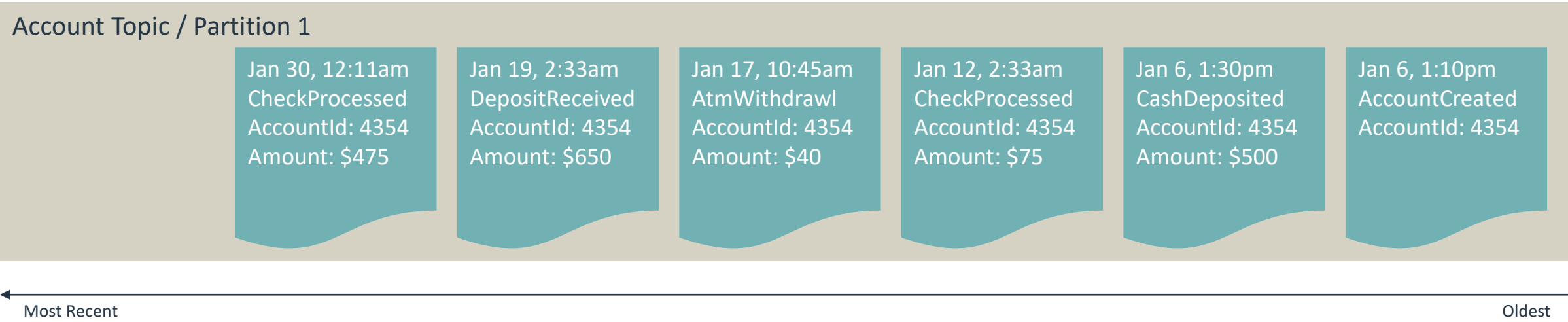
Bank Account Event Stream



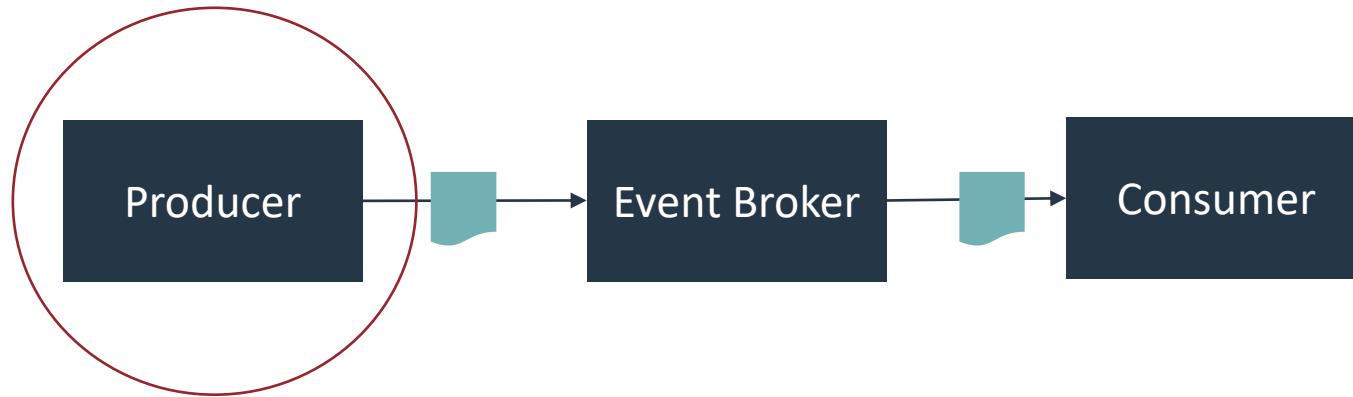
Bank Account Event Stream



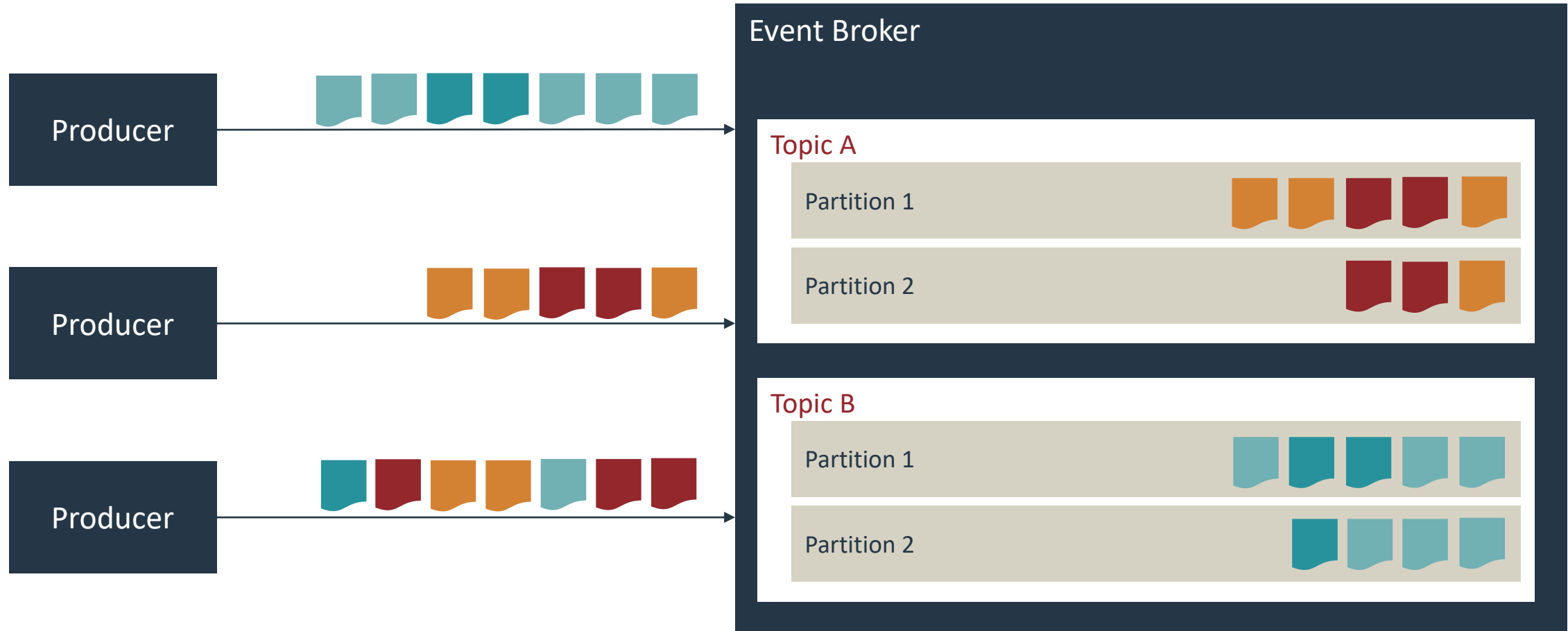
Bank Account Event Stream



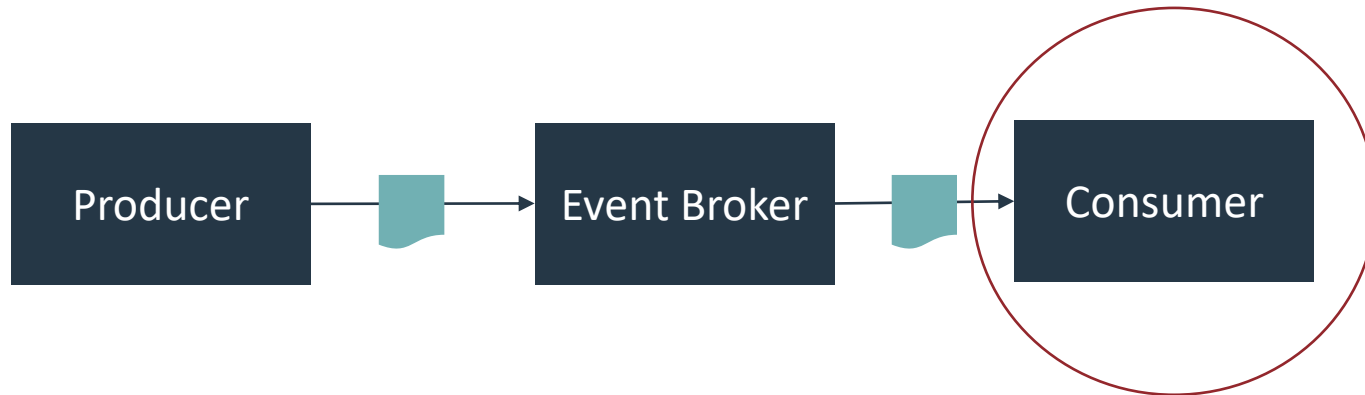
Producer Details



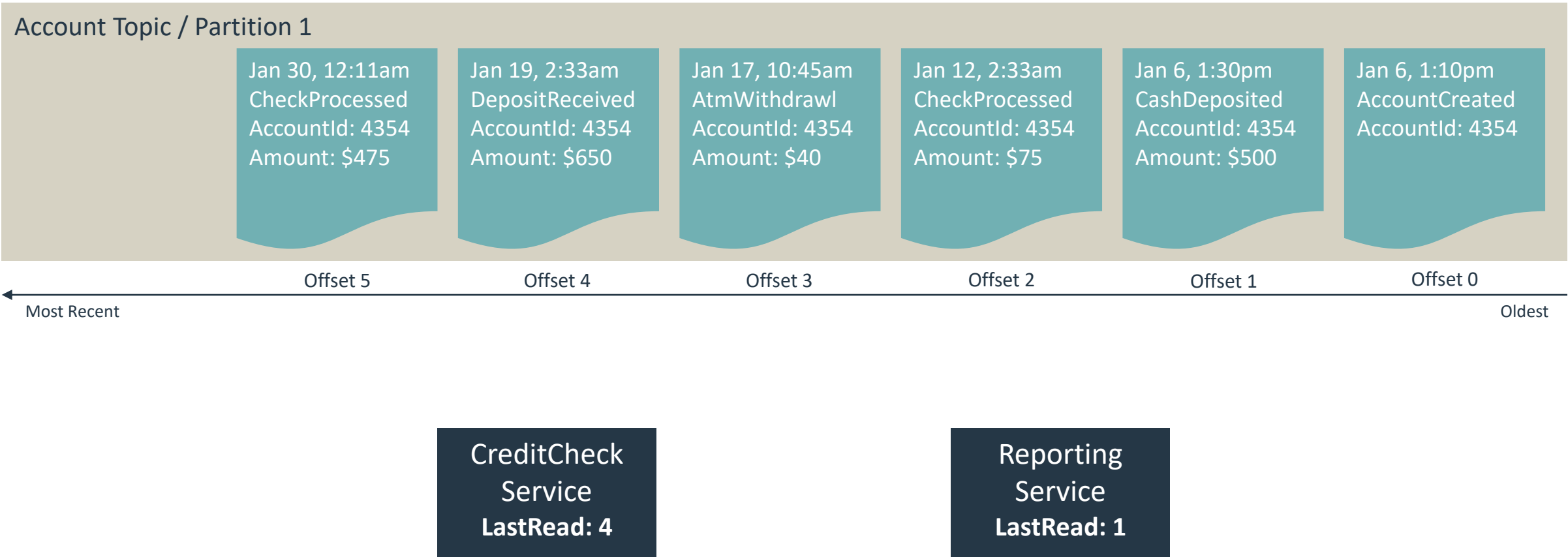
Publishing Various Event Types



Consumer Details

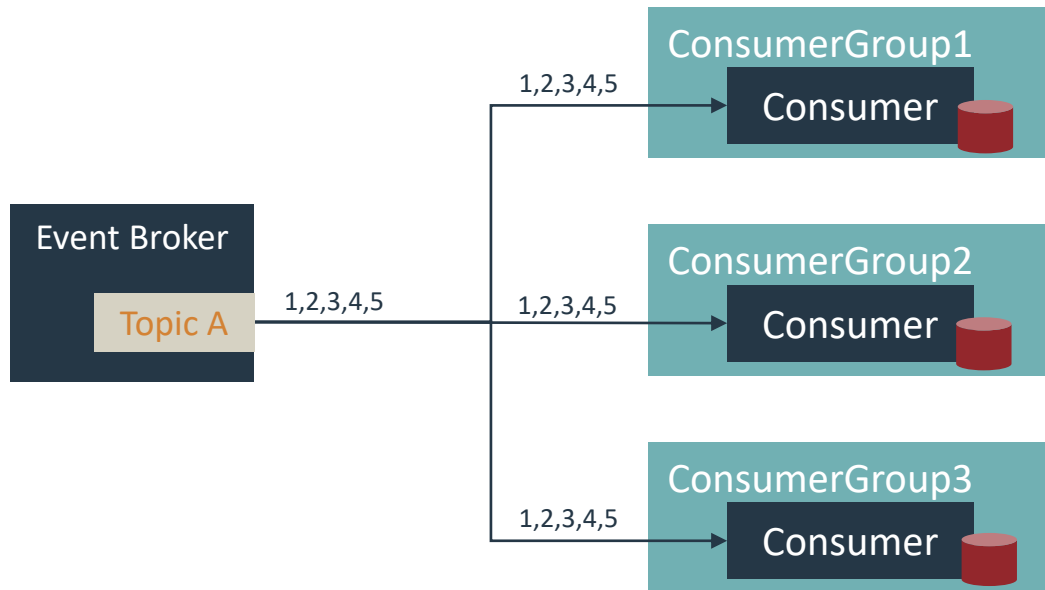


Ordered Consumption



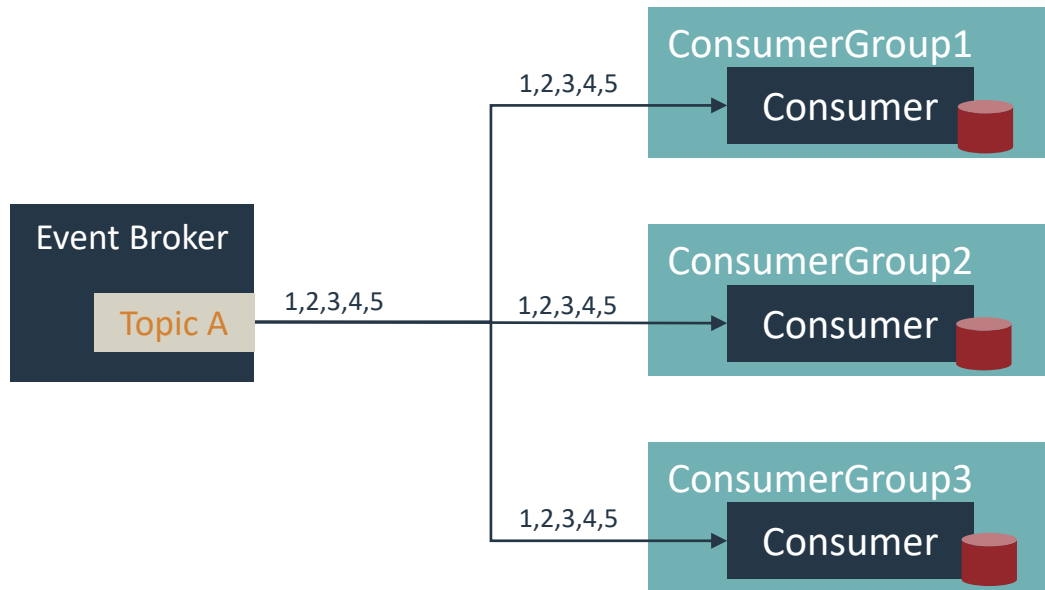
Scaling with Consumer Groups

Multiple Consumer Groups

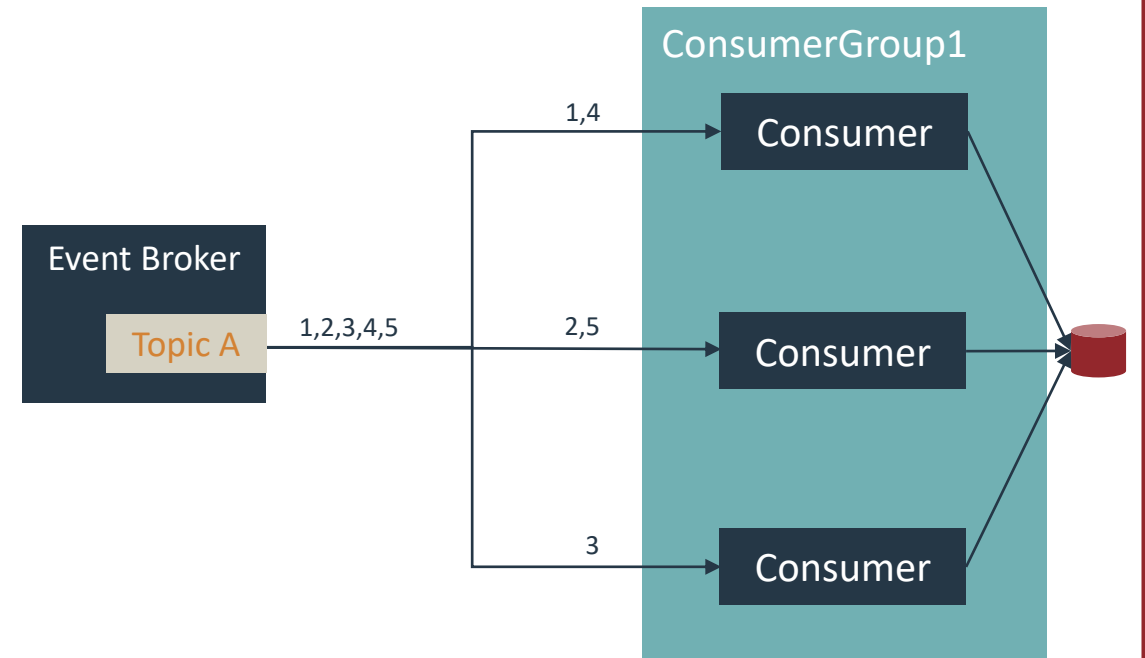


Scaling with Consumer Groups

Multiple Consumer Groups



Multiple Consumers In Single Group

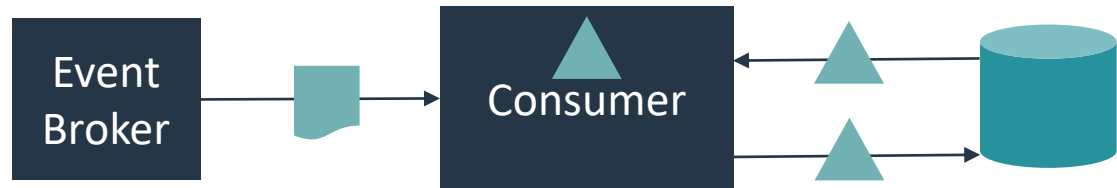


Consumer Data Management

- Consumers may not act on received events immediately
 - May collect data needed to process later events
- Helps decouple microservices

Consumer Data Management

Projections / Material Views



Consumer Data Management

Projections / Material Views

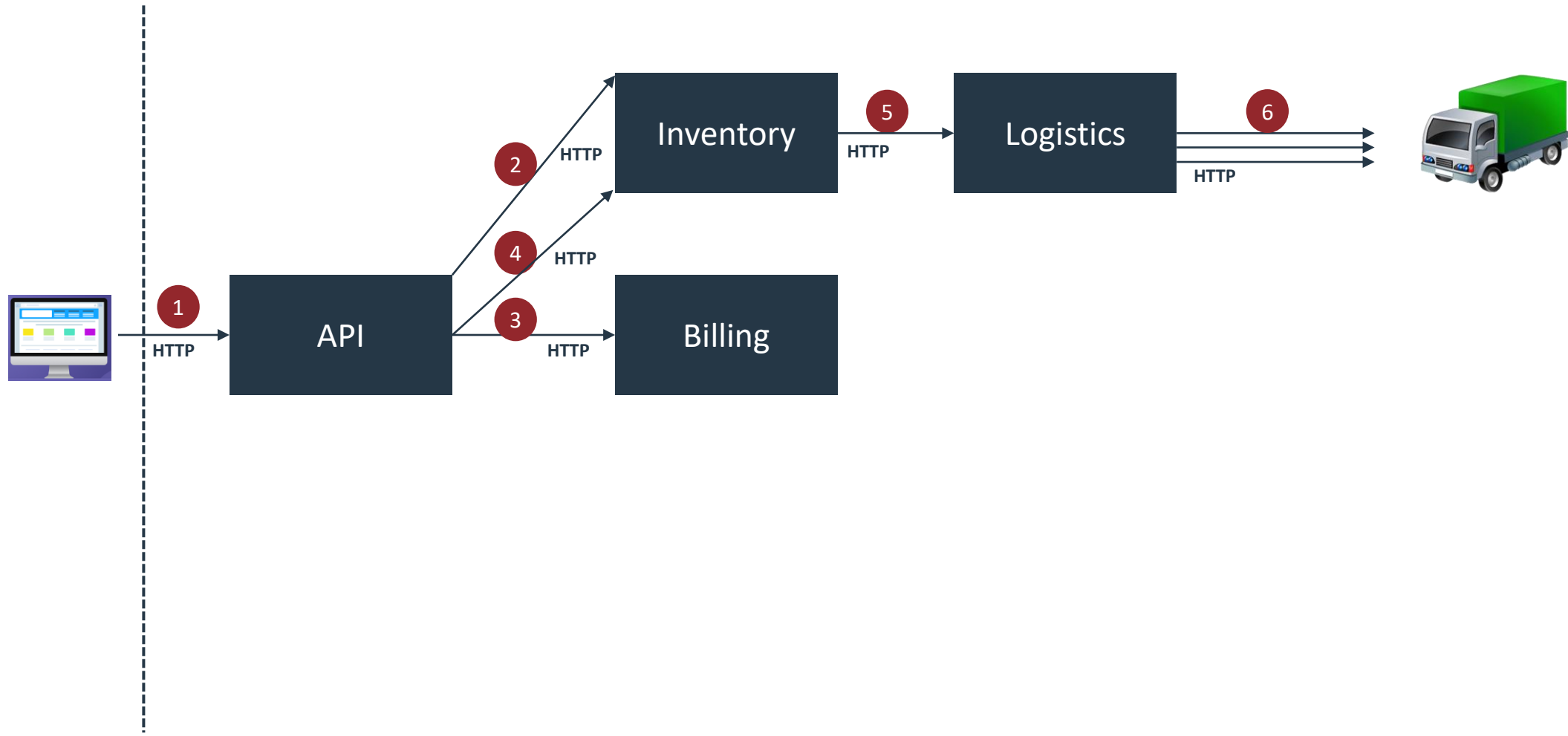


Event Sourcing

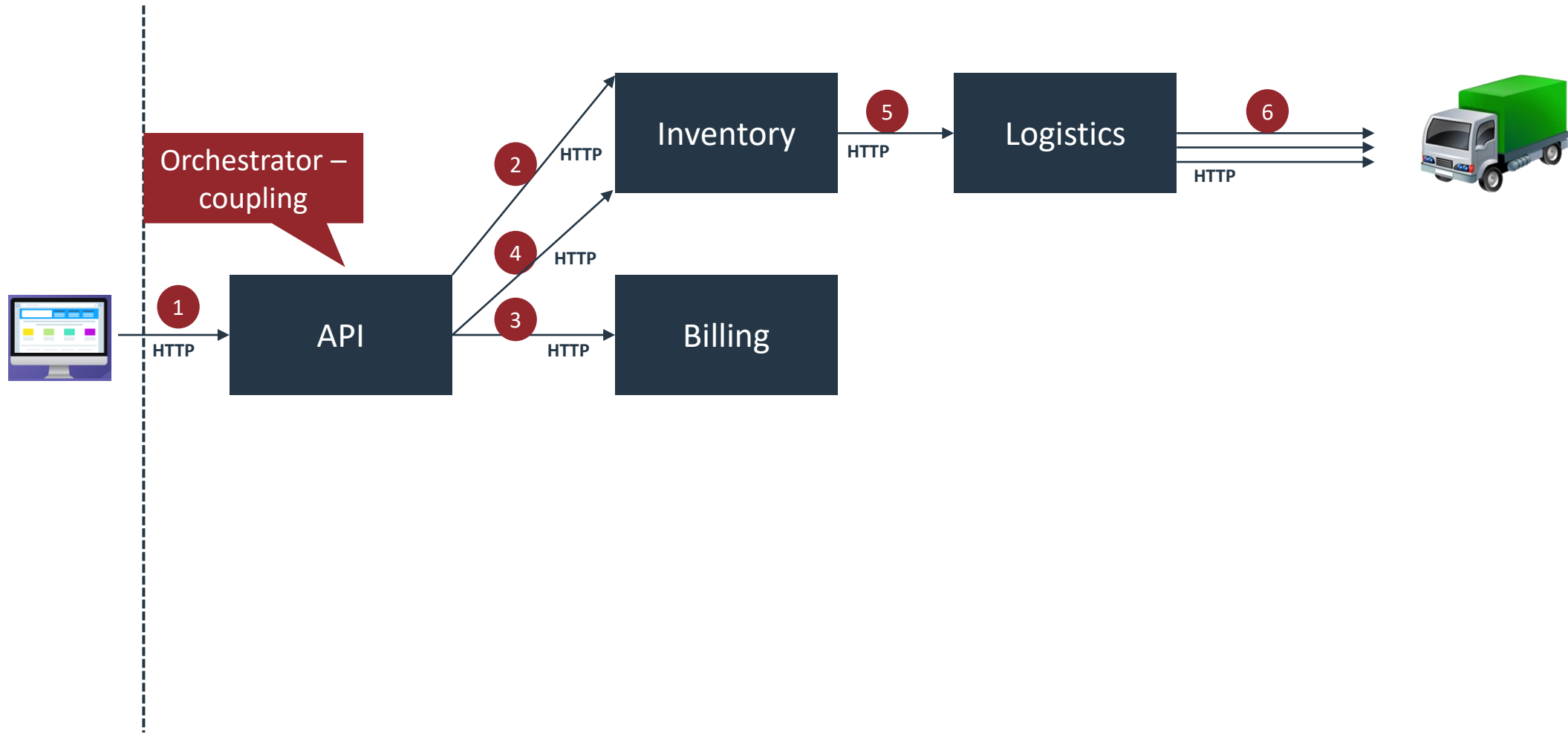


Request-Based vs Event-Based Solution Comparison

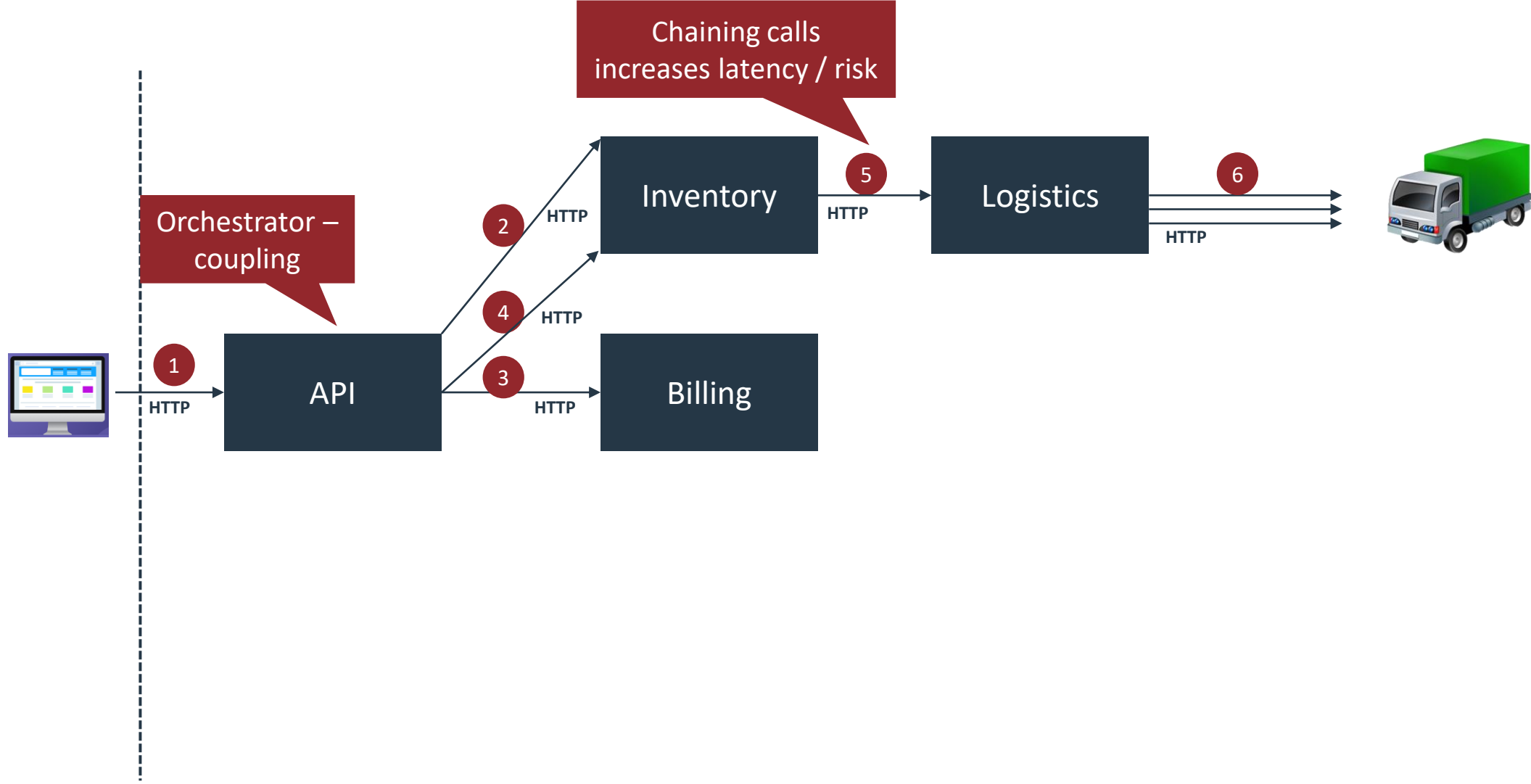
Request-Based Point of Sale



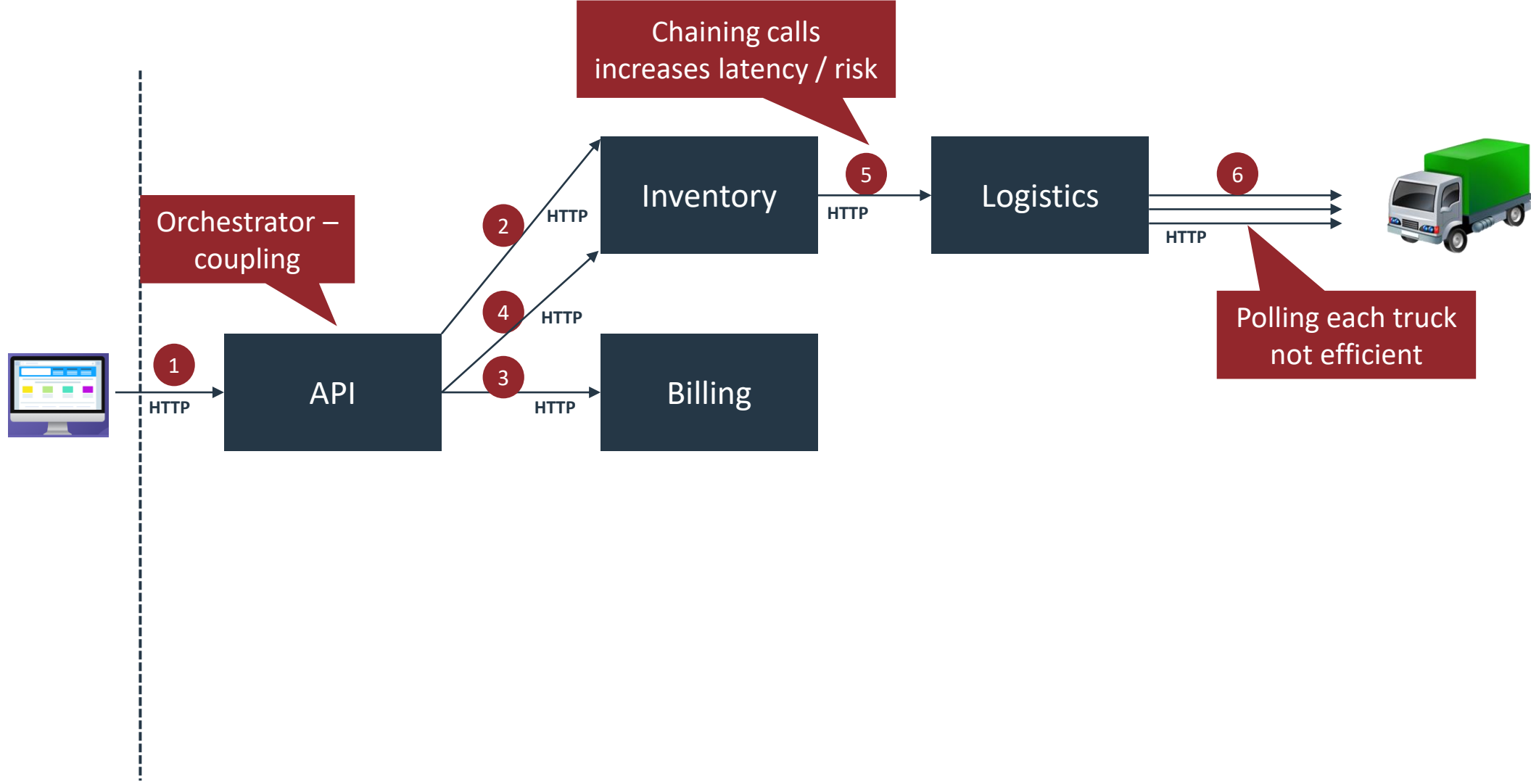
Request-Based Point of Sale



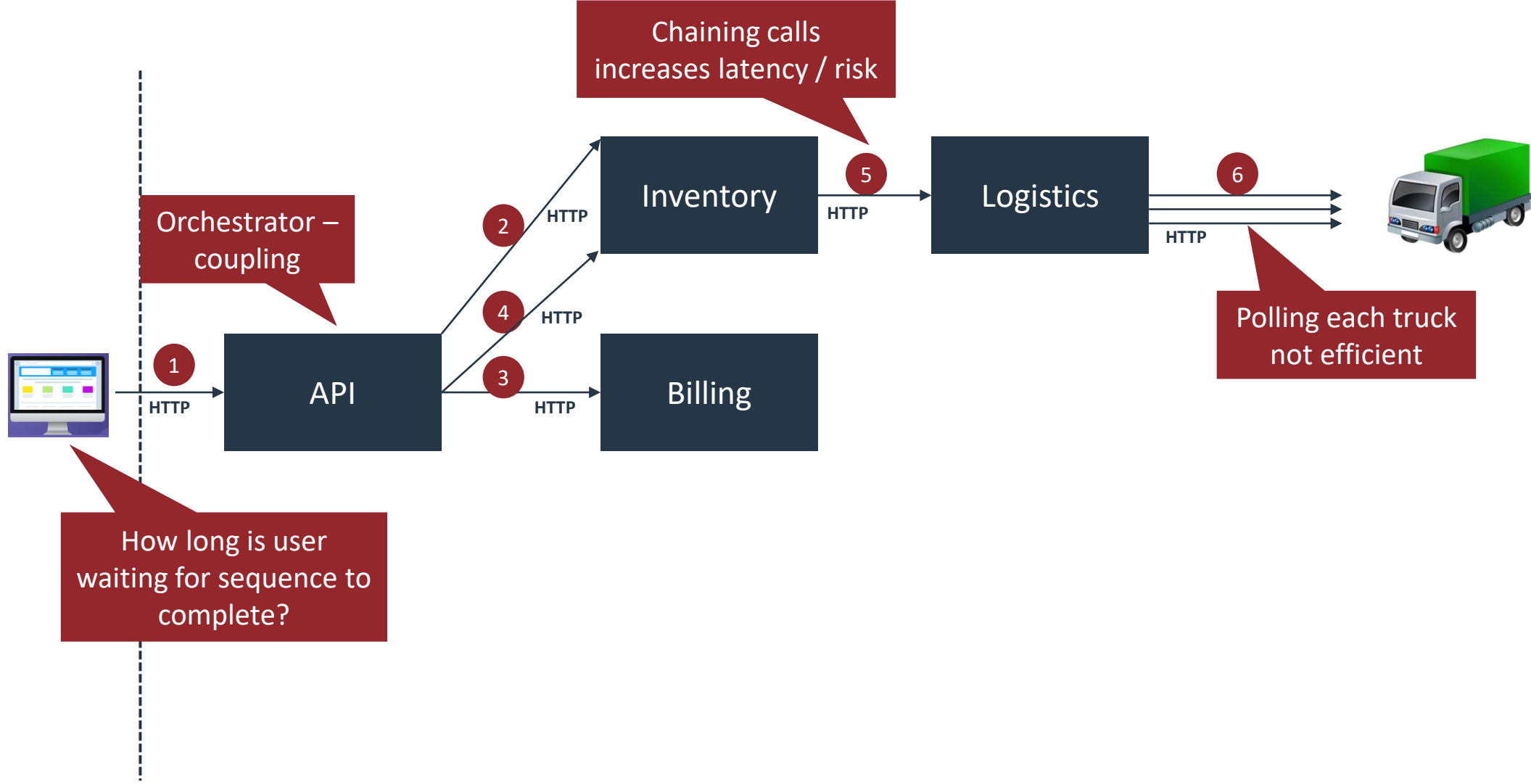
Request-Based Point of Sale



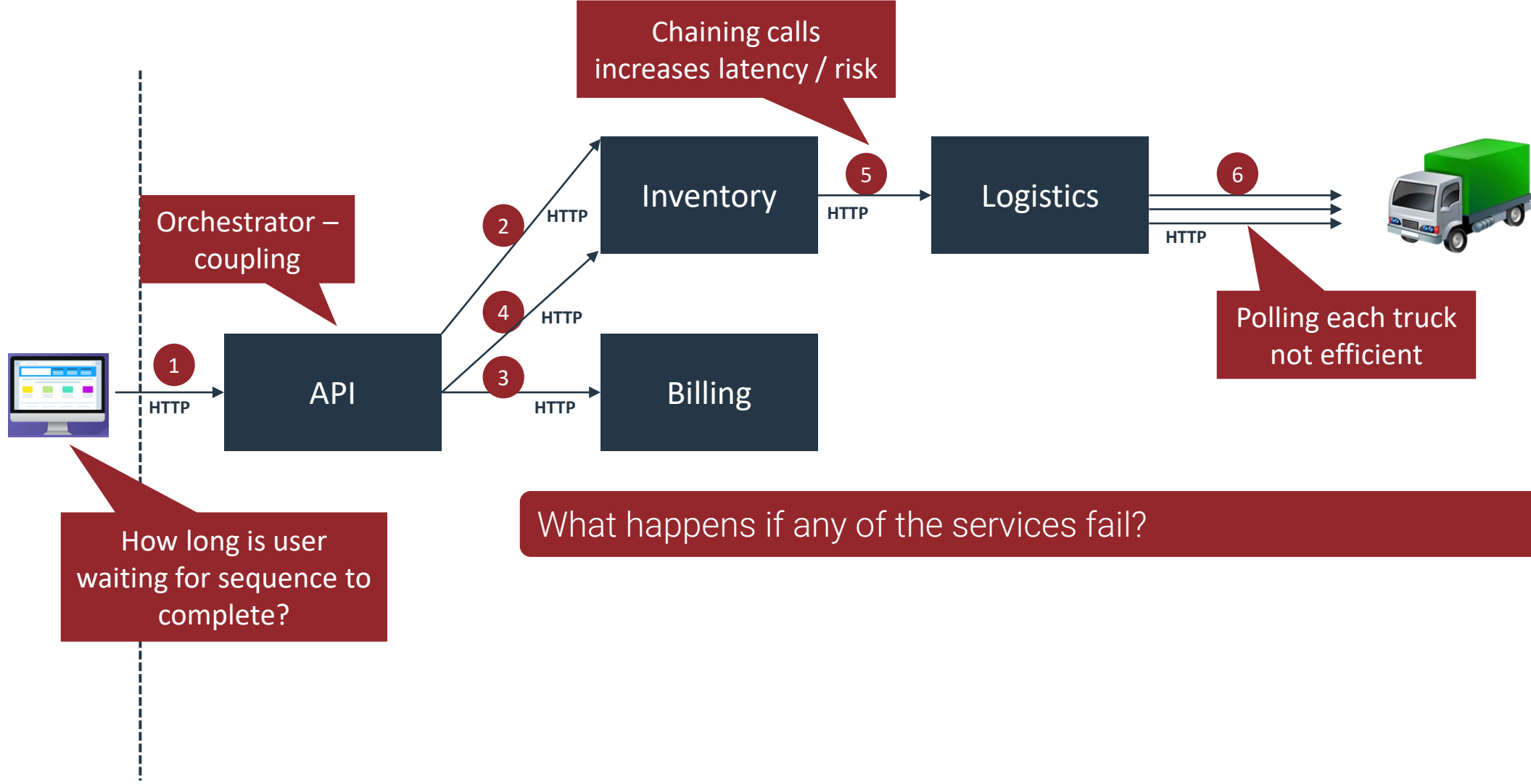
Request-Based Point of Sale



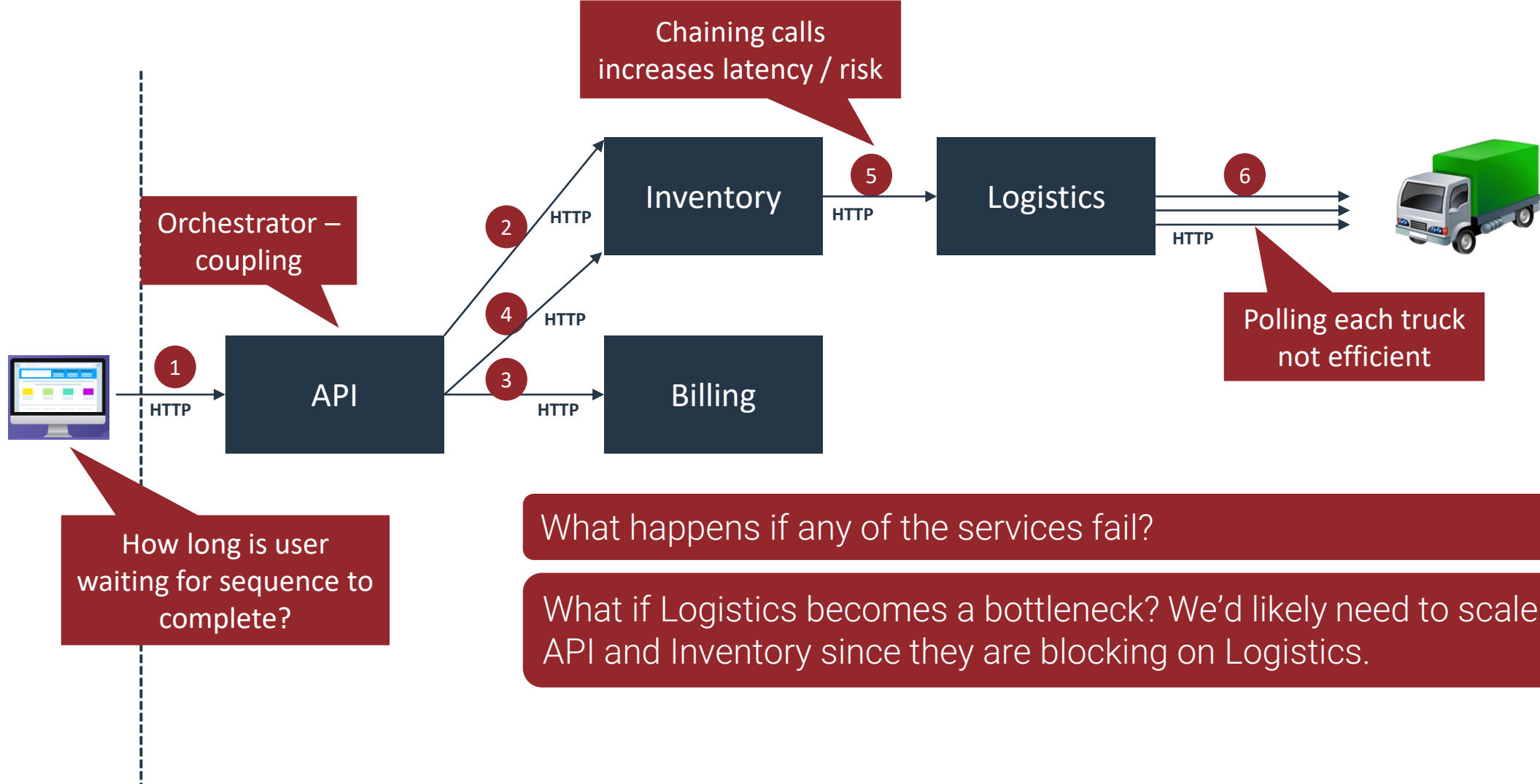
Request-Based Point of Sale



Request-Based Point of Sale



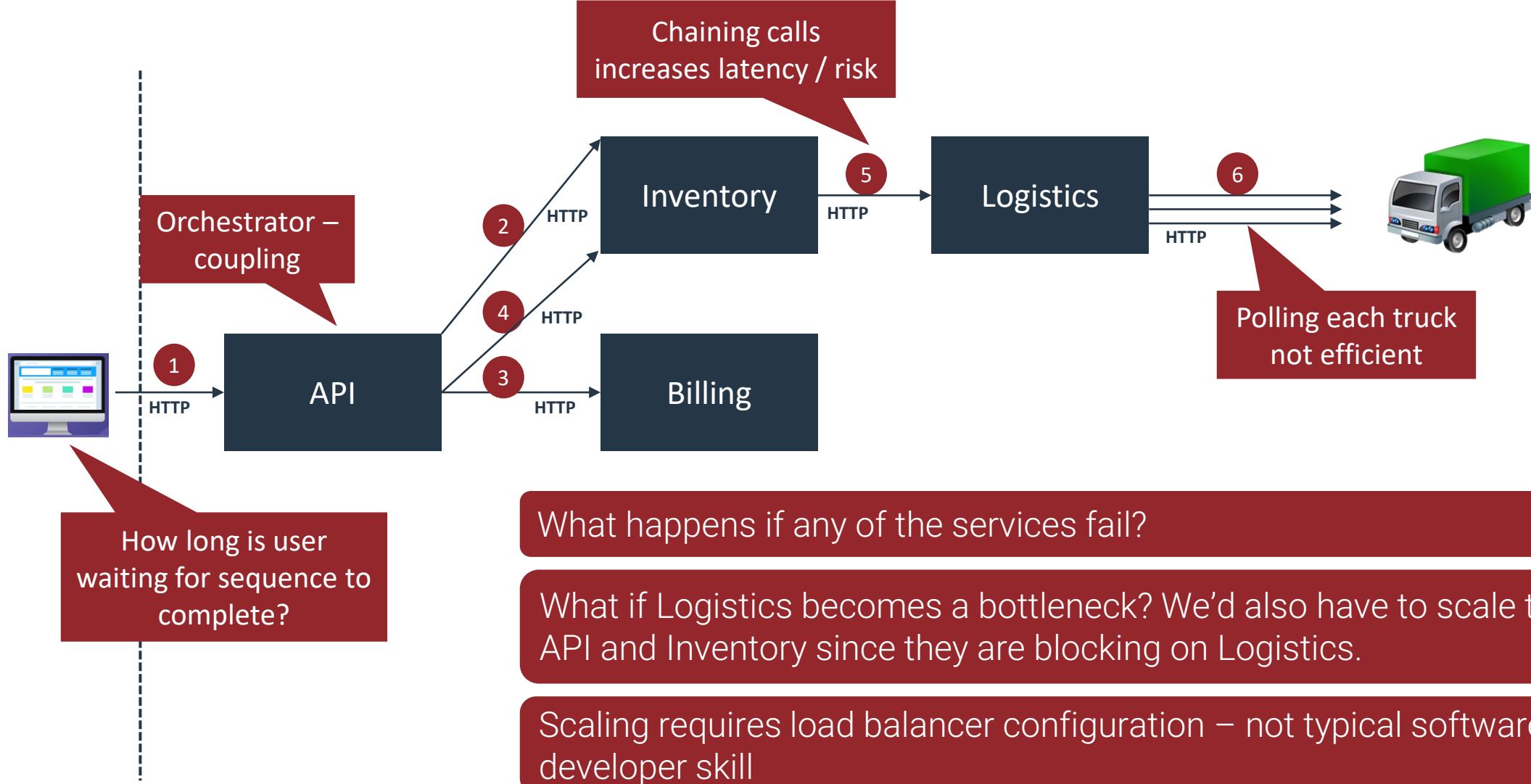
Request-Based Point of Sale



What happens if any of the services fail?

What if Logistics becomes a bottleneck? We'd likely need to scale the API and Inventory since they are blocking on Logistics.

Request-Based Point of Sale

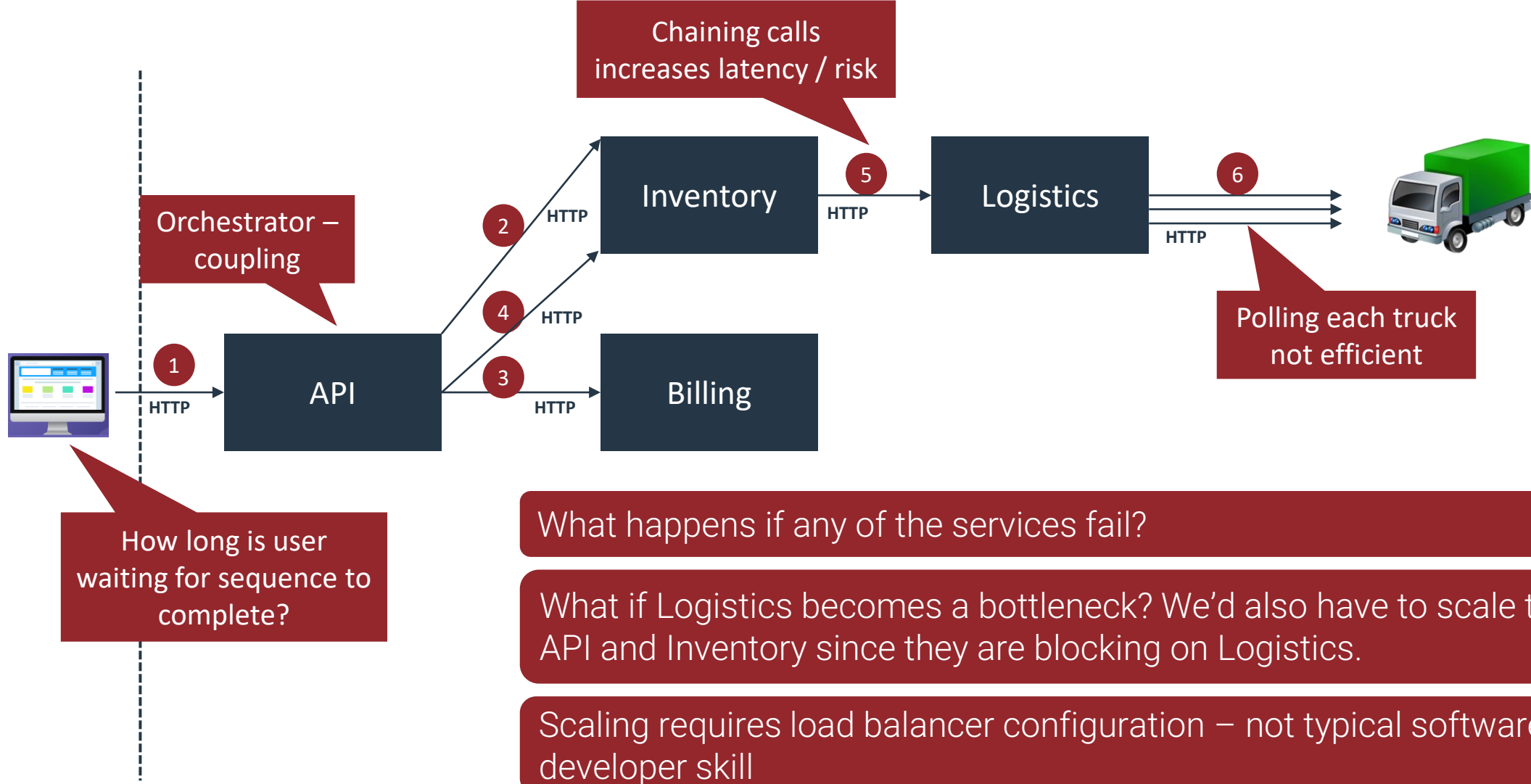


What happens if any of the services fail?

What if Logistics becomes a bottleneck? We'd also have to scale the API and Inventory since they are blocking on Logistics.

Scaling requires load balancer configuration – not typical software developer skill

Request-Based Point of Sale



What happens if any of the services fail?

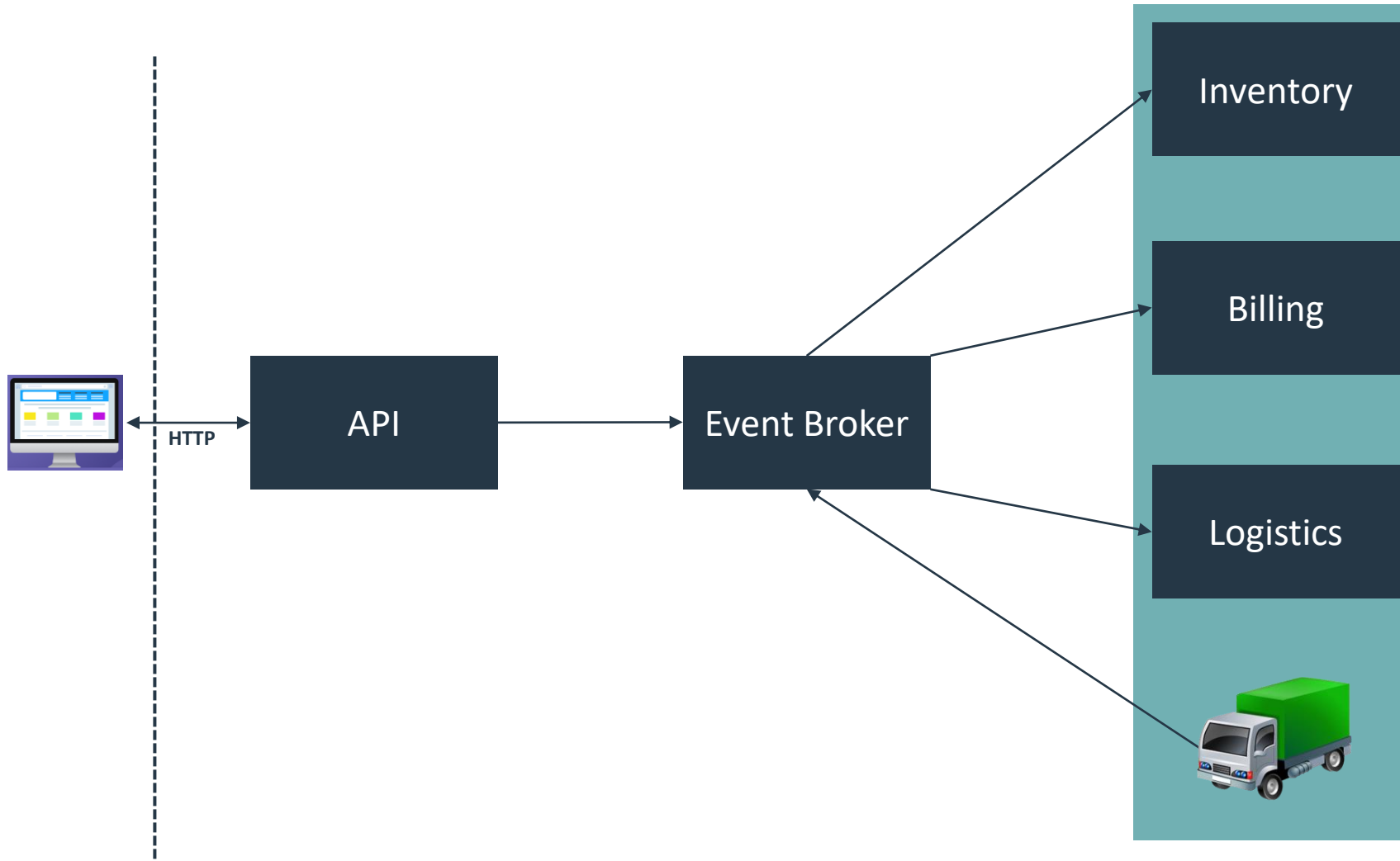
What if Logistics becomes a bottleneck? We'd also have to scale the API and Inventory since they are blocking on Logistics.

Scaling requires load balancer configuration – not typical software developer skill

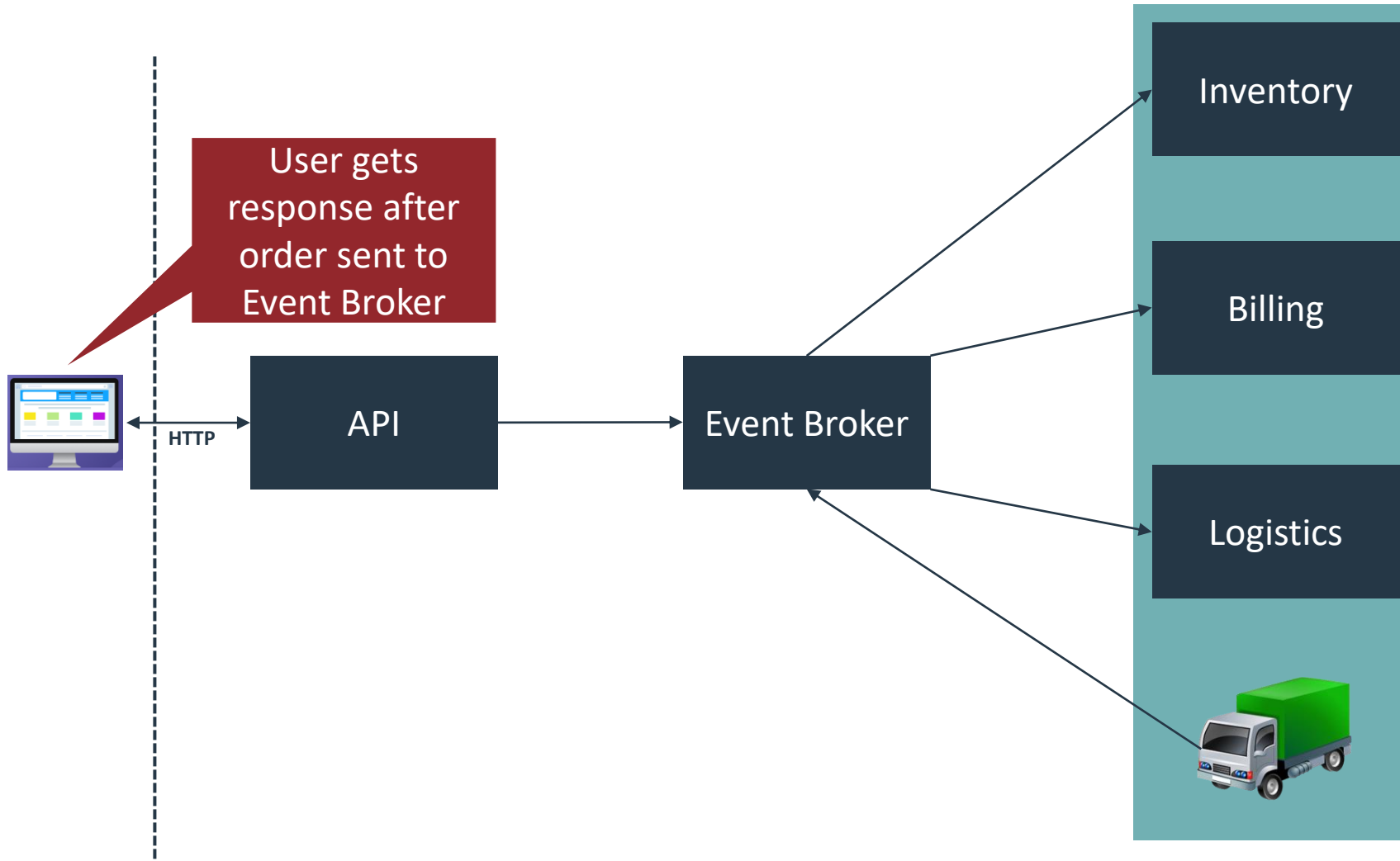
What if we want to add a Fraud or Recommendation service?

Event-Based Point of Sale

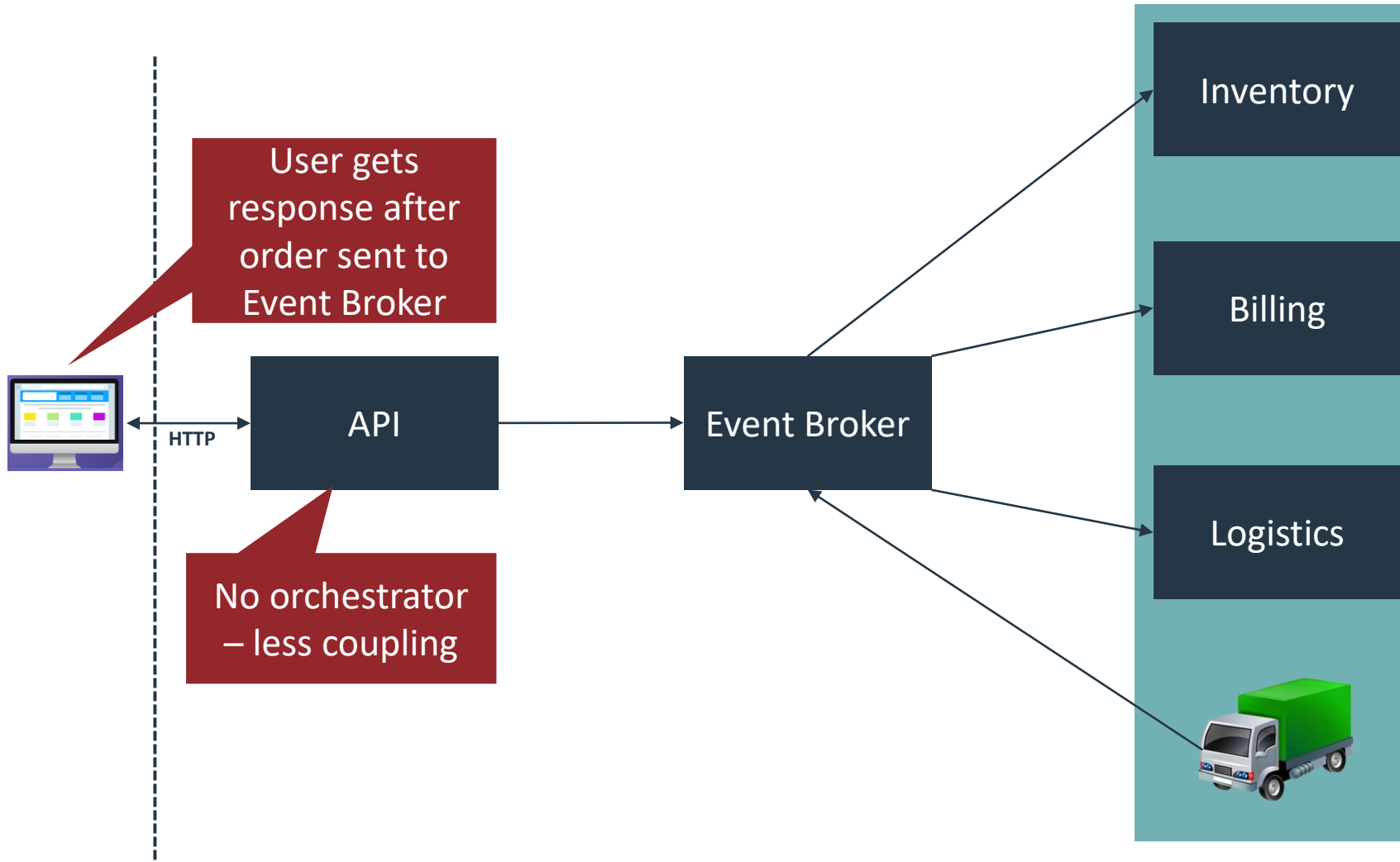
Event-Based Point of Sale



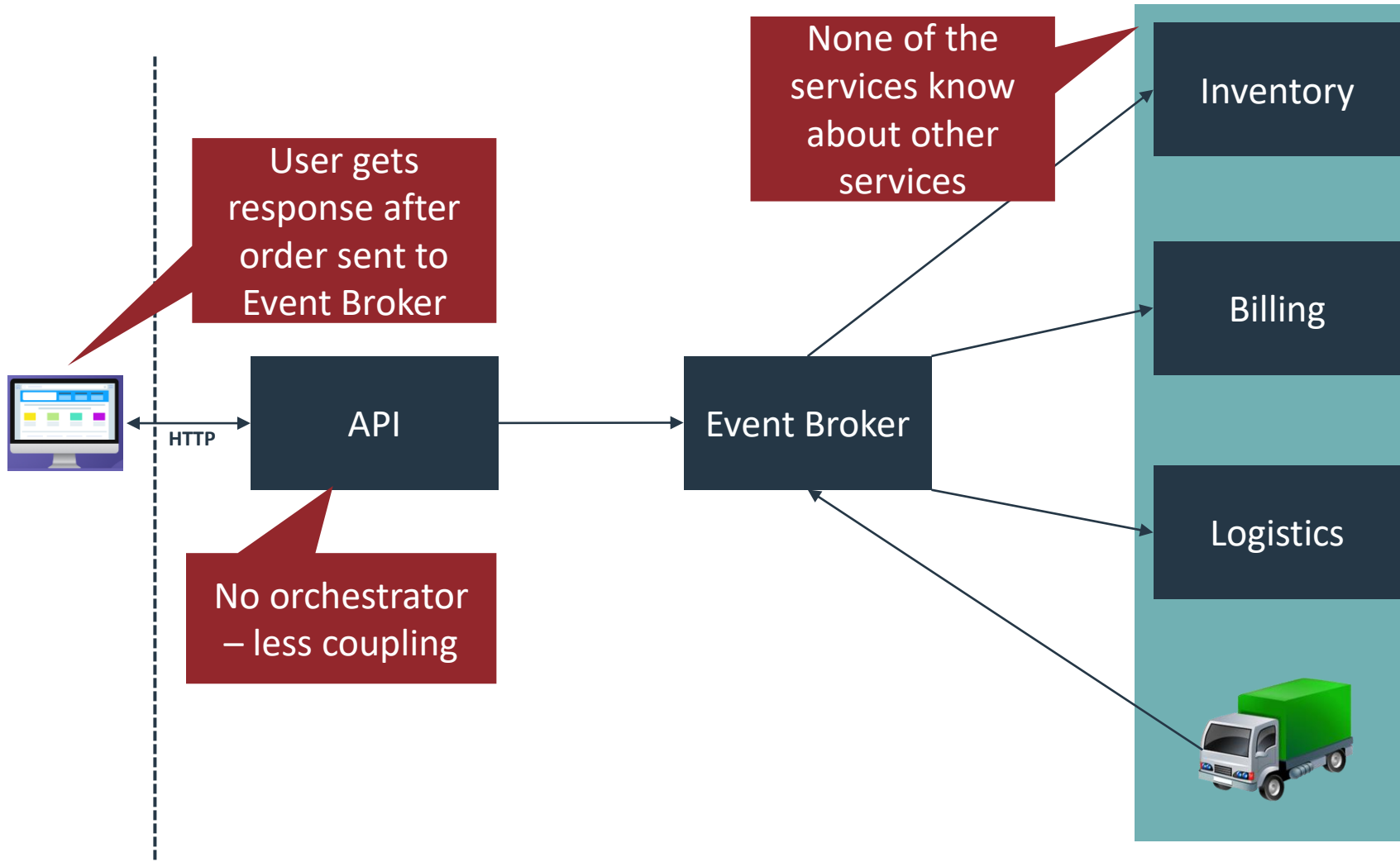
Event-Based Point of Sale



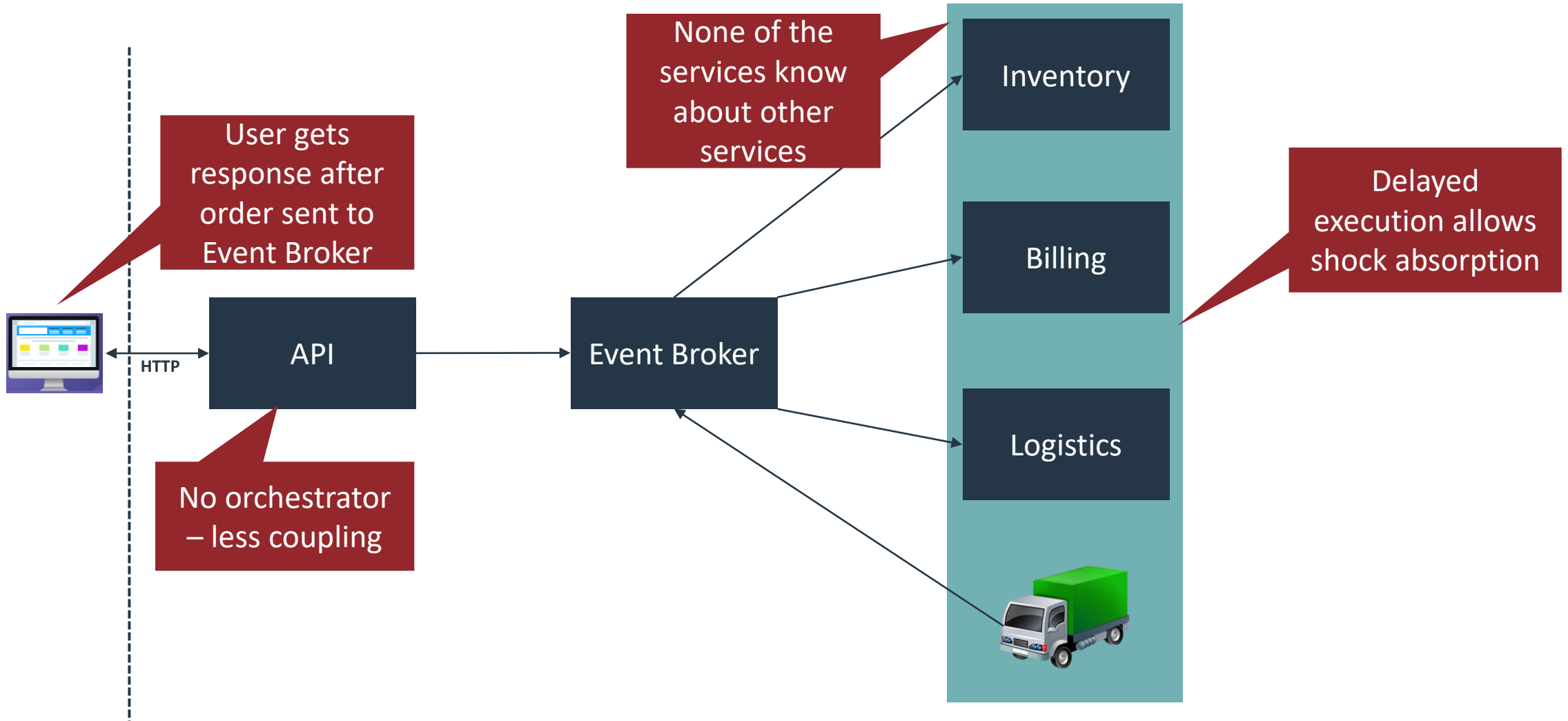
Event-Based Point of Sale



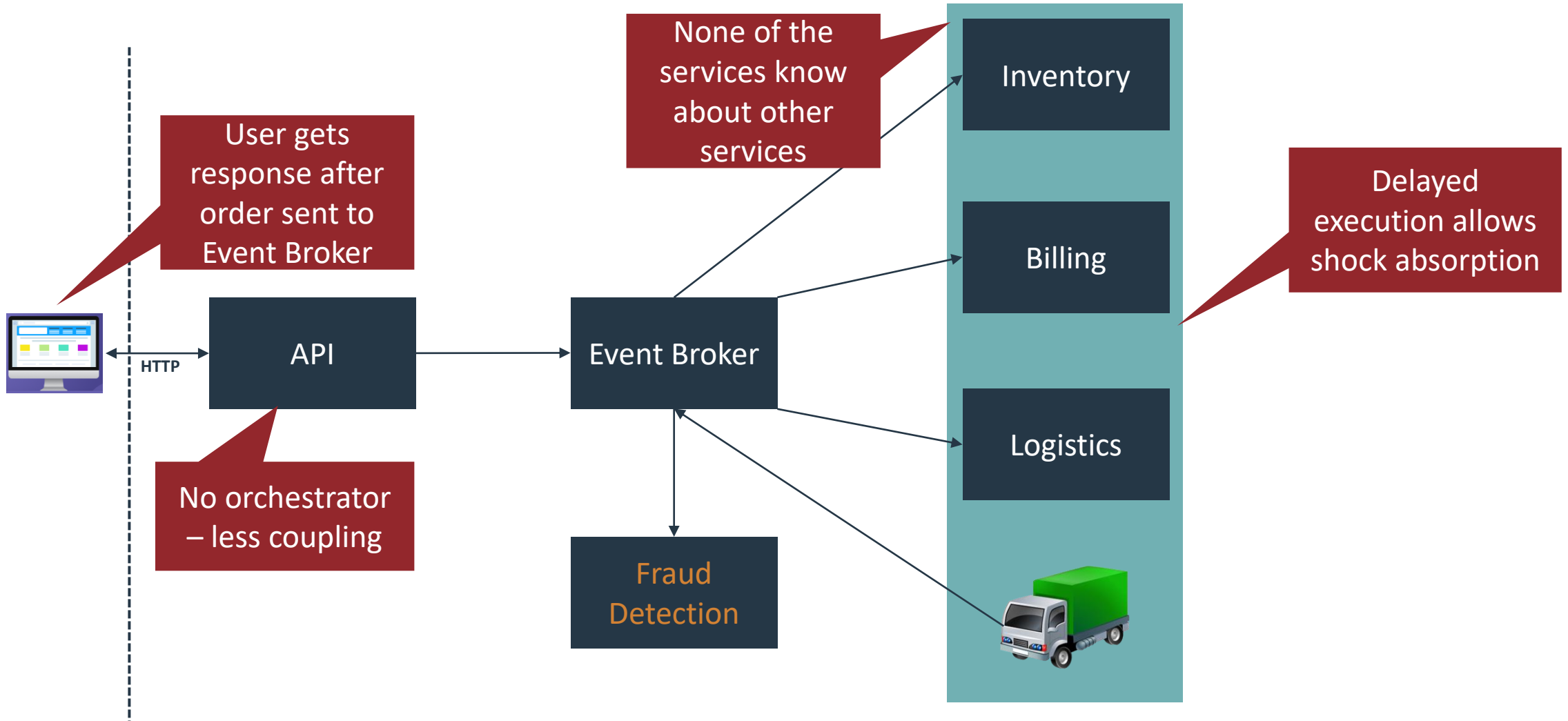
Event-Based Point of Sale



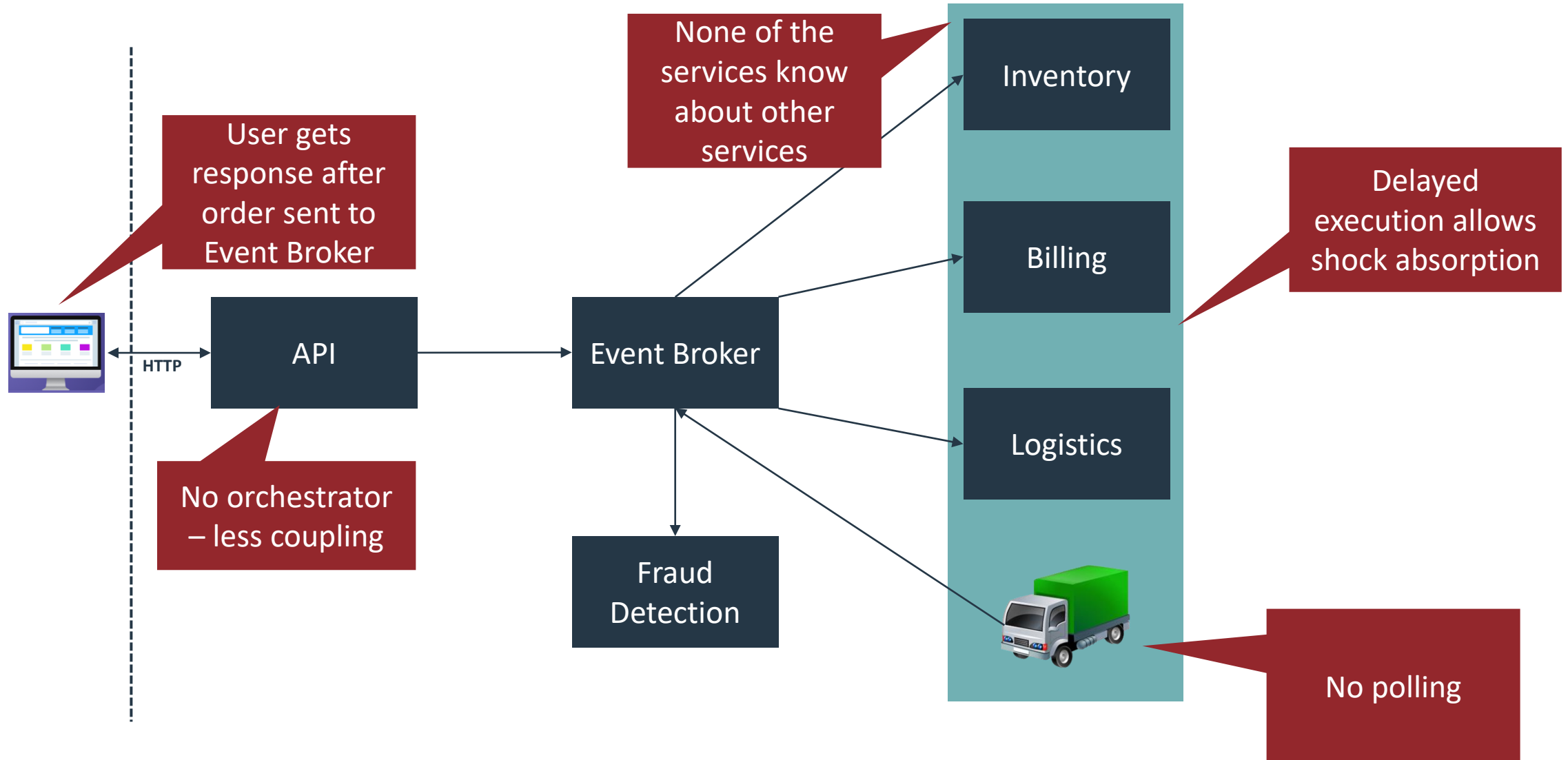
Event-Based Point of Sale



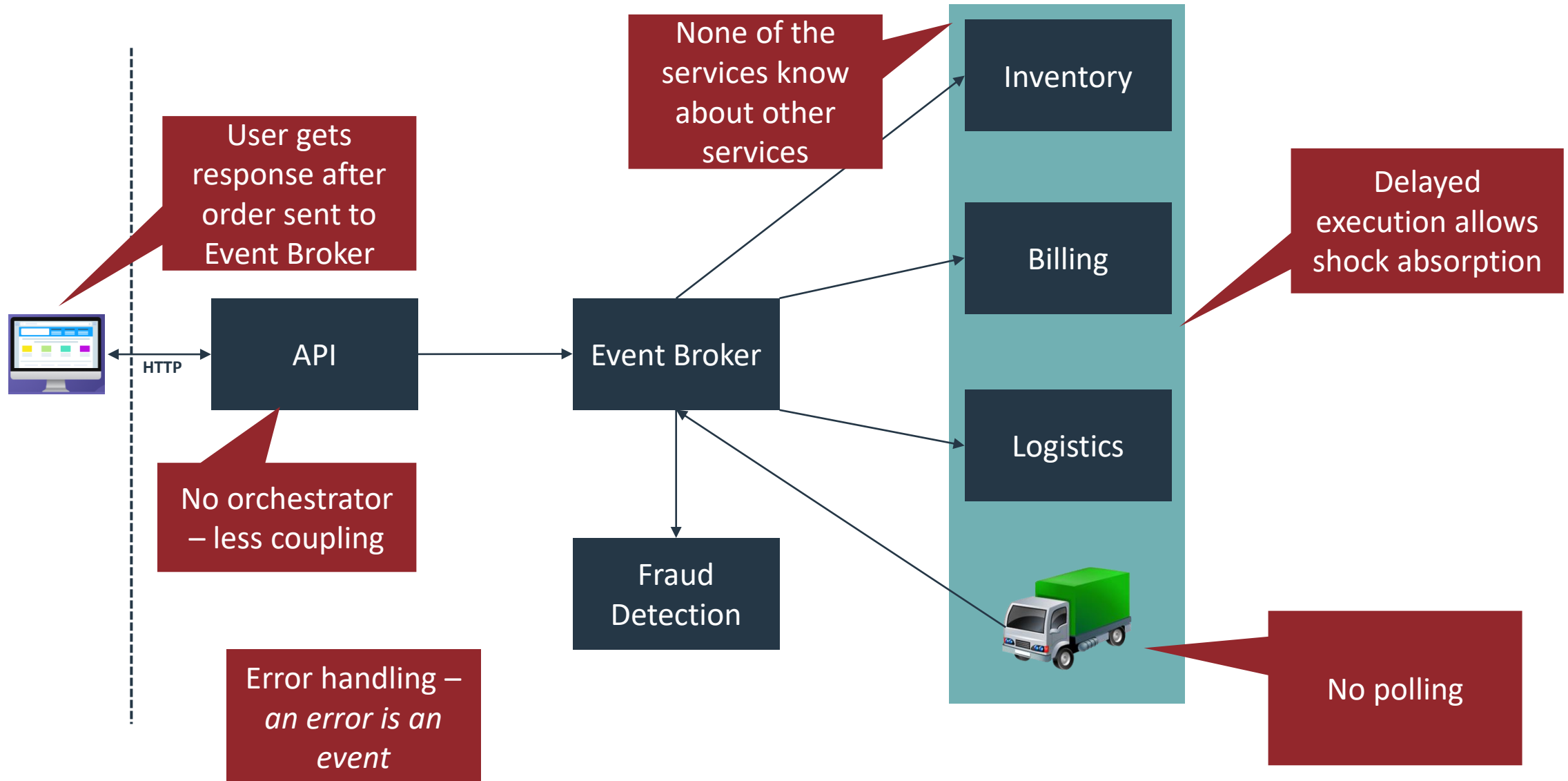
Event-Based Point of Sale



Event-Based Point of Sale

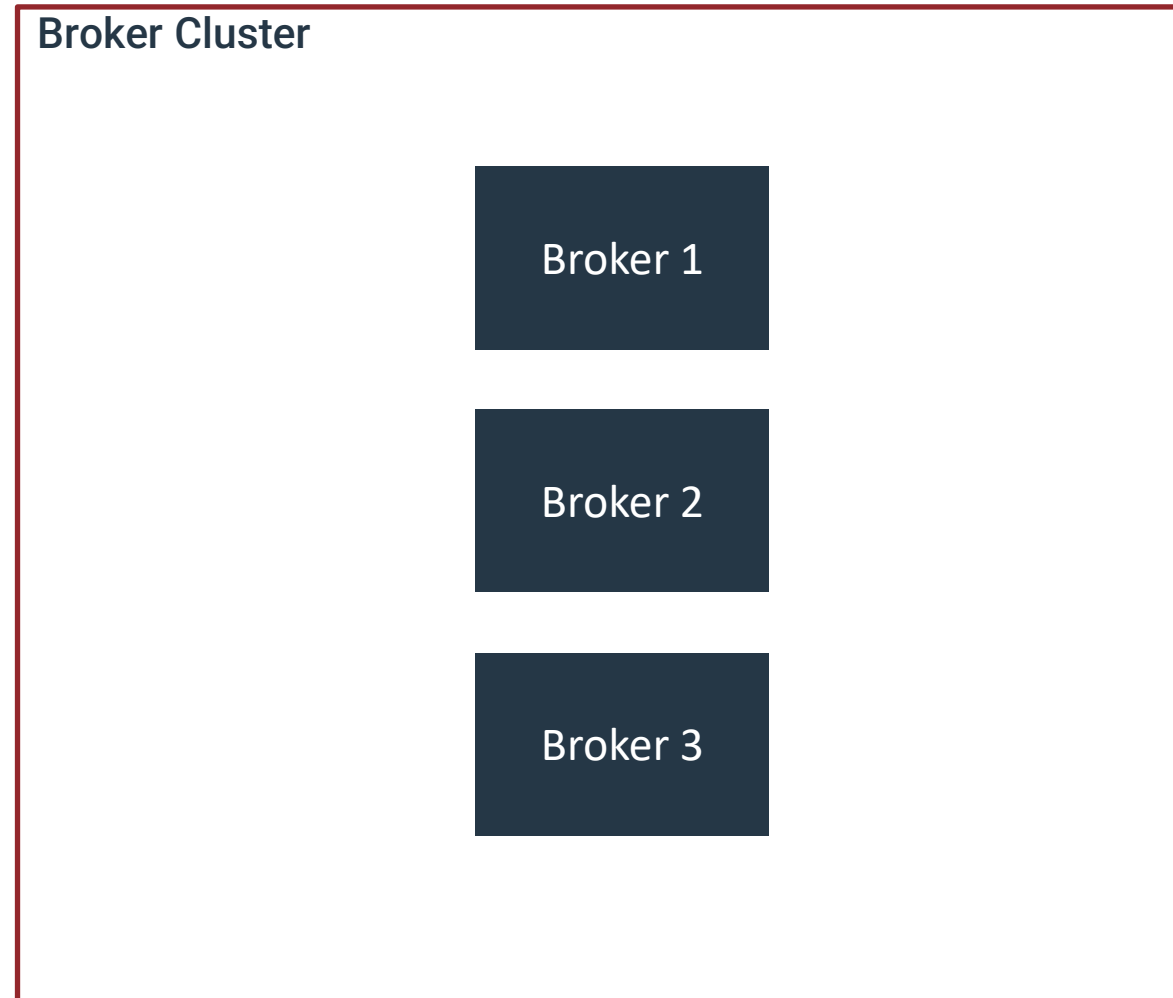


Event-Based Point of Sale

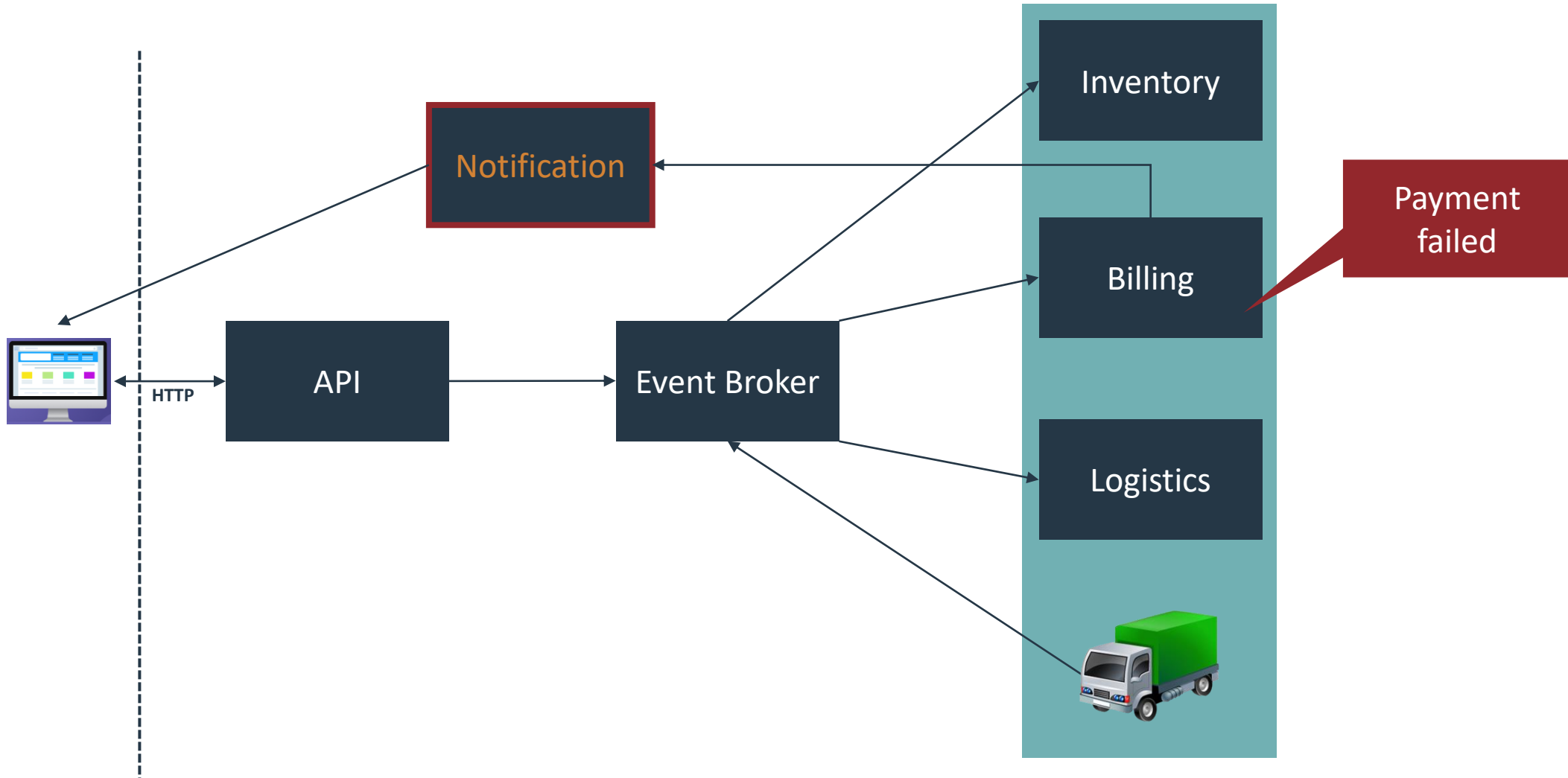


EDA Challenges

Event Broker Cluster

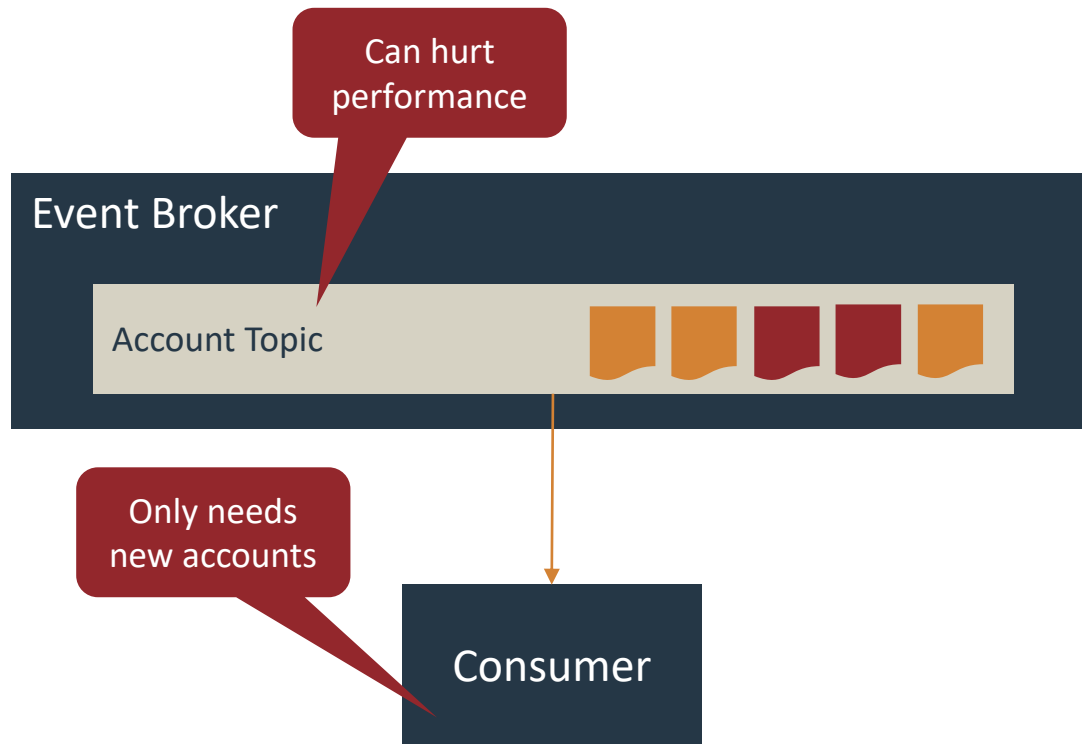


Notification System



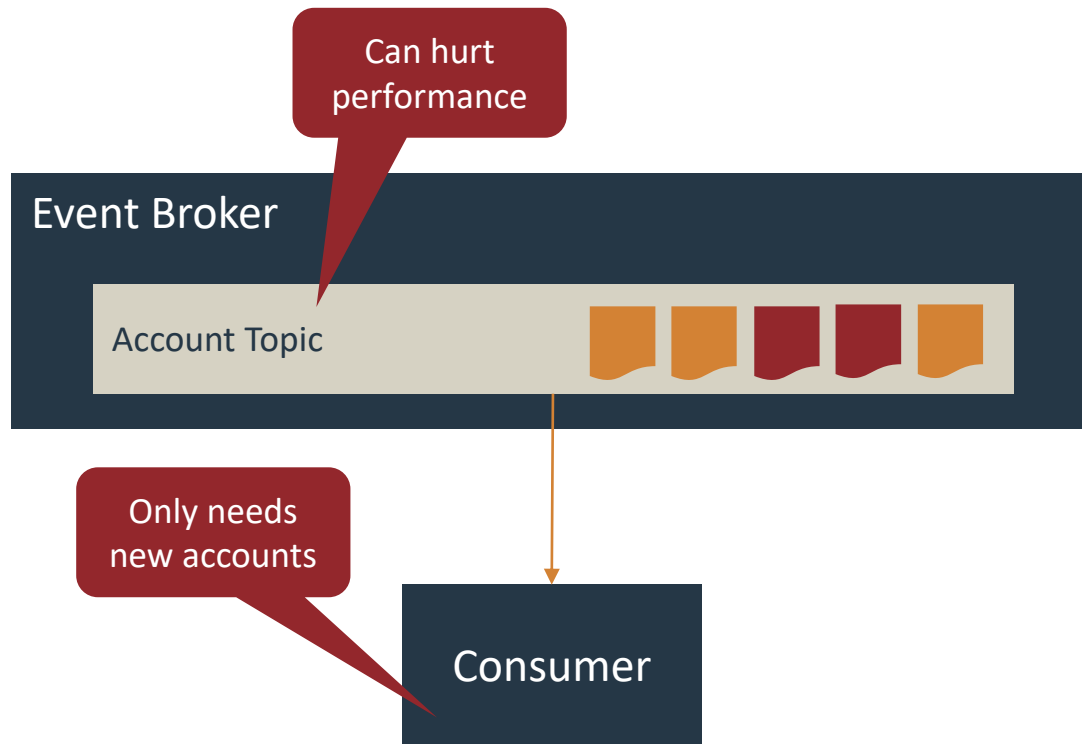
Topic / Event Design is Hard

Topic Per Aggregate

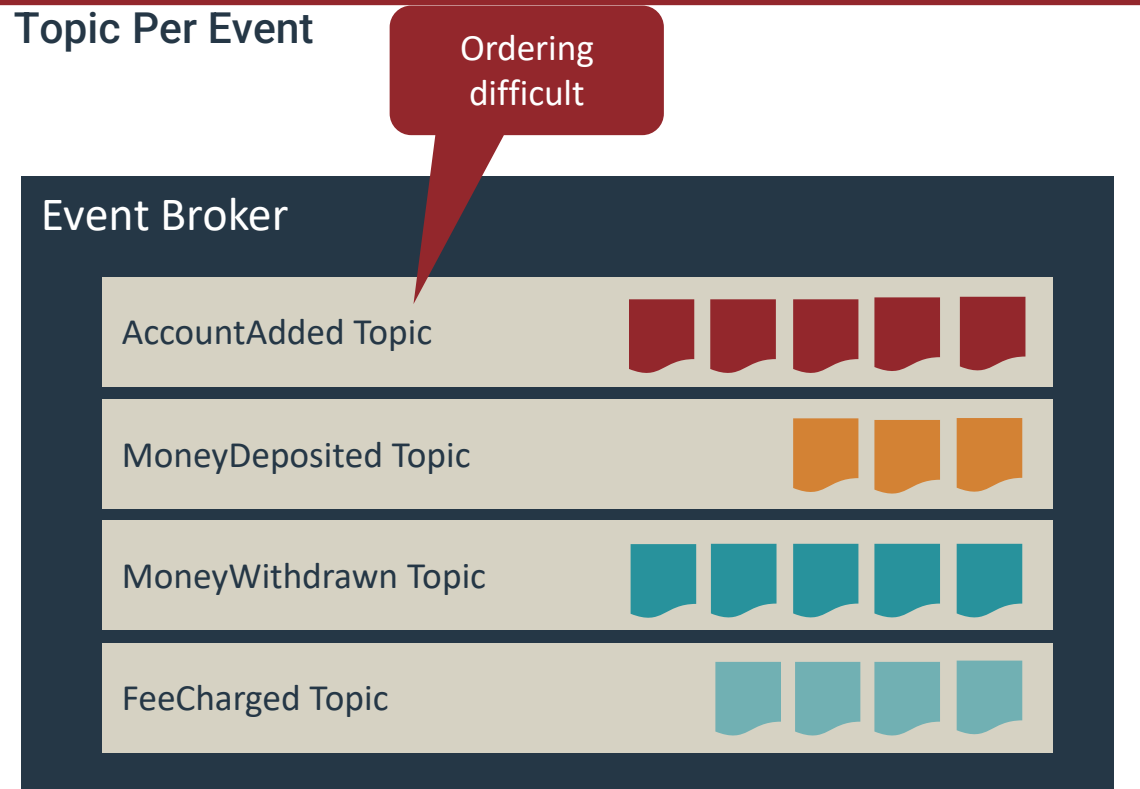


Topic / Event Design

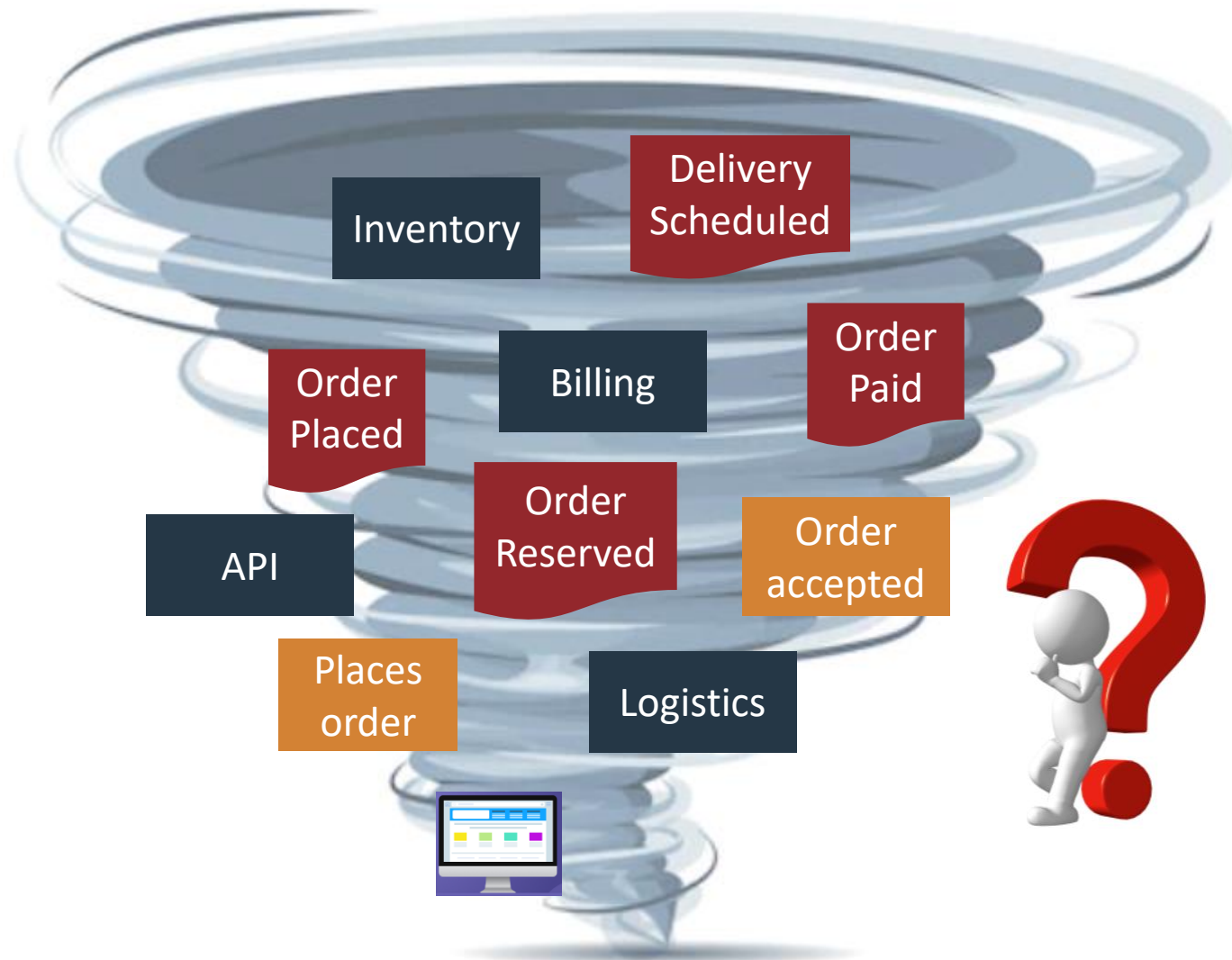
Topic Per Aggregate



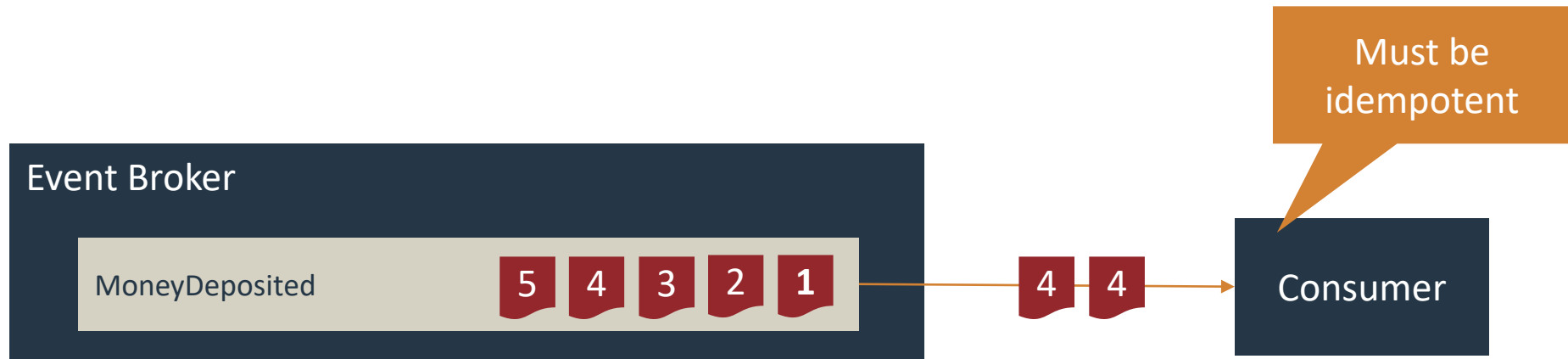
Topic Per Event



Complicated Troubleshooting

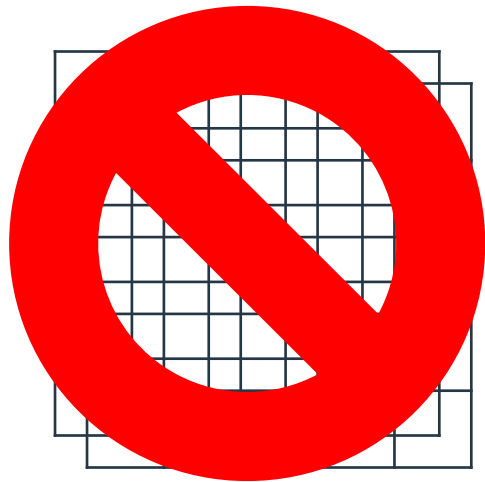


New Error Handling Concerns



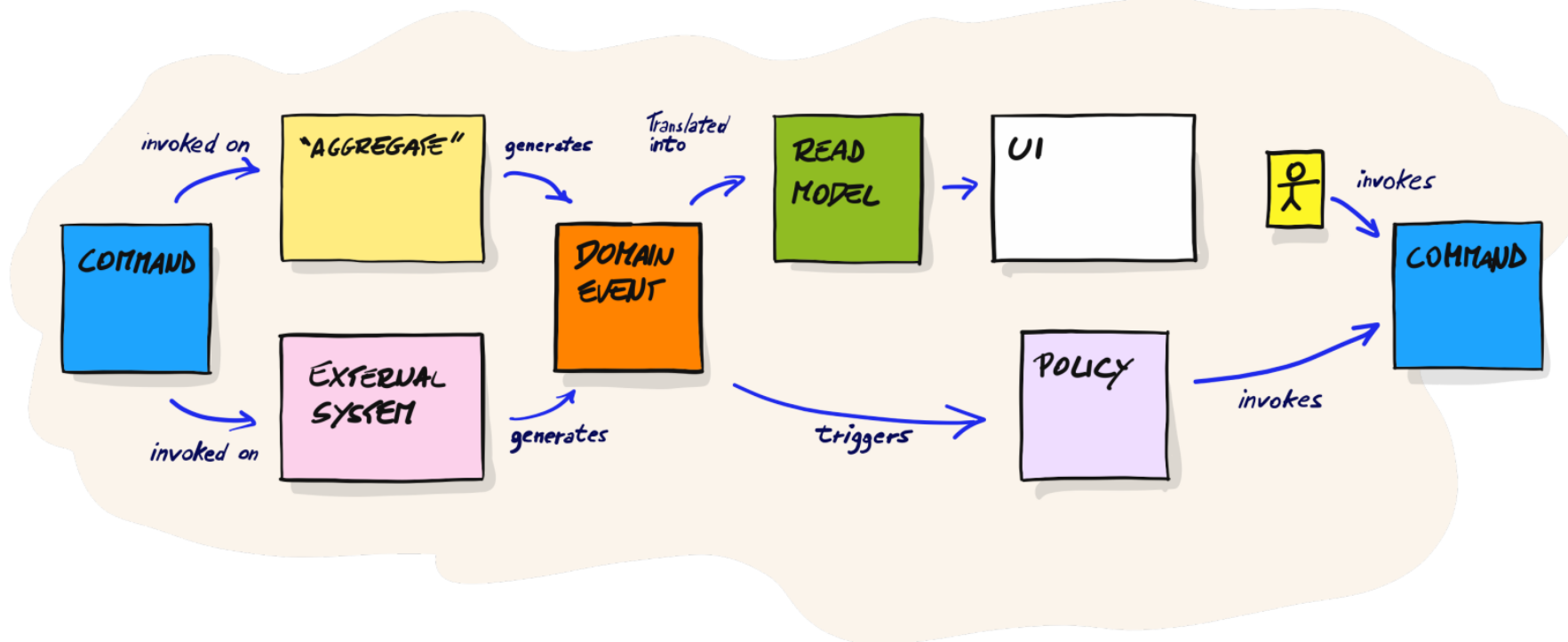
Getting Started with EDA

Create an Event Mindset

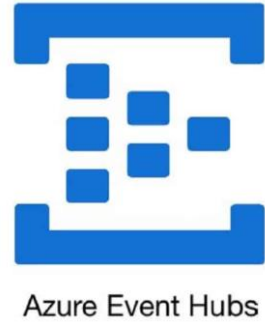


Practice Event Storming

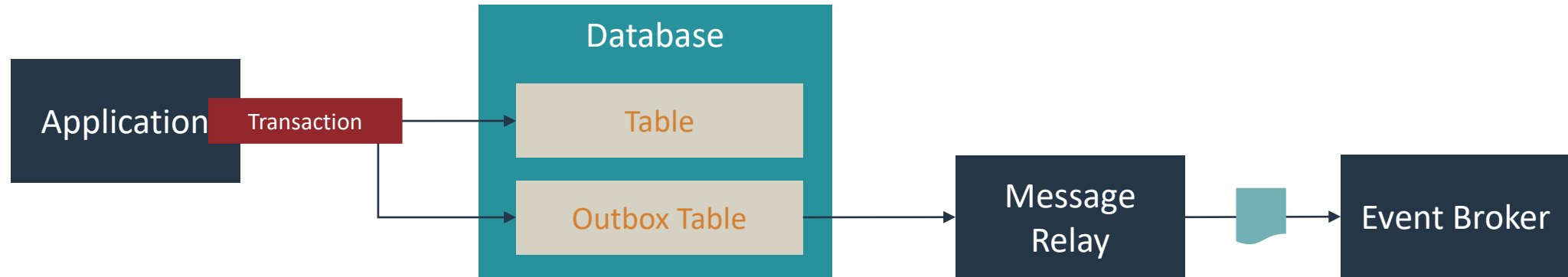
EVENT STORMING



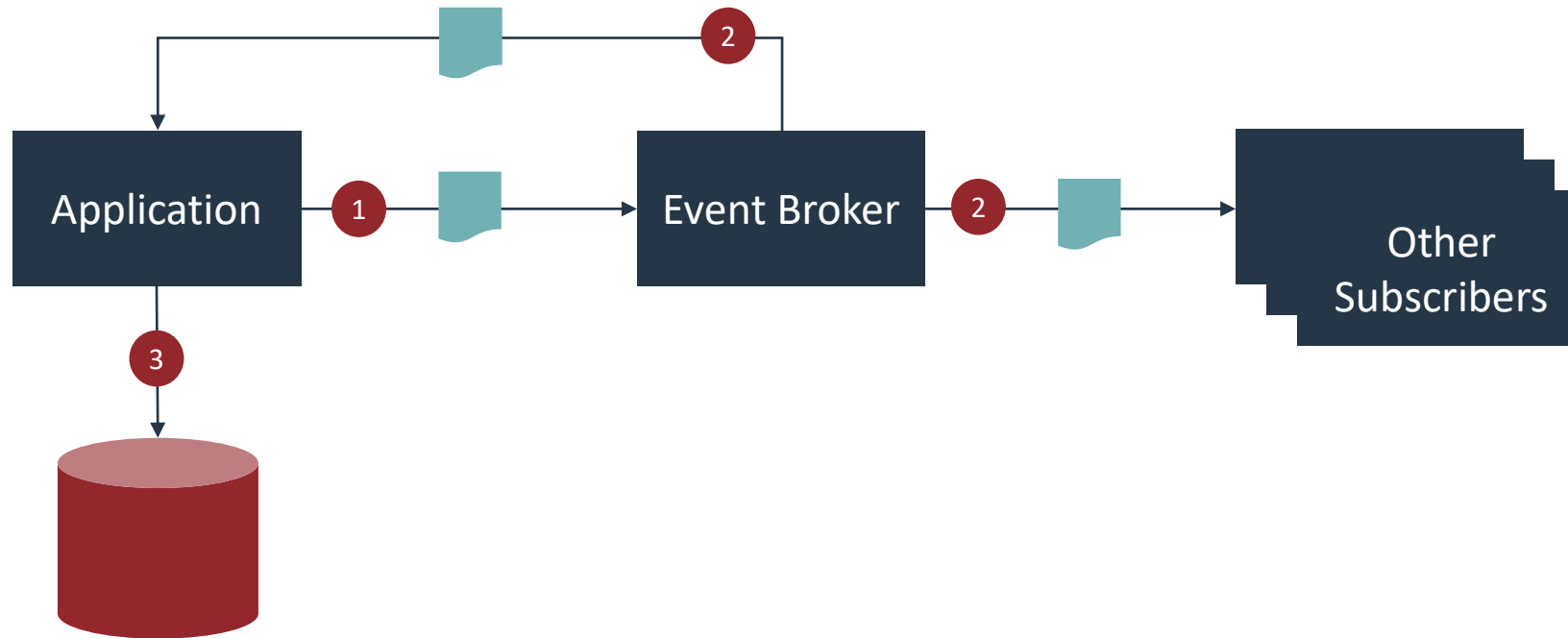
Learn About Messaging Options



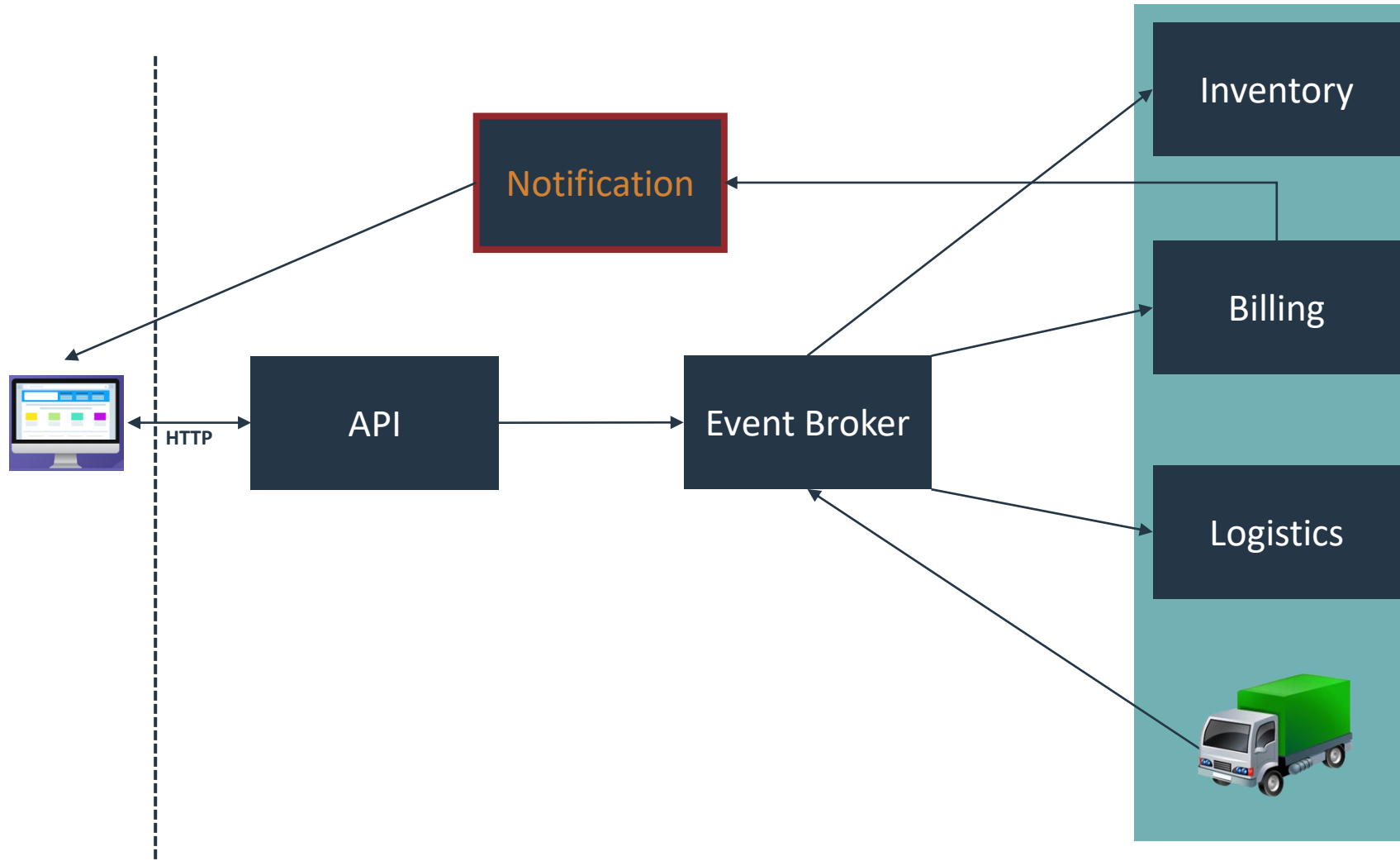
Consider Outbox Pattern



Prefer Listen to Yourself Pattern



Learn About Push Notification Options



Understand Eventual Consistency



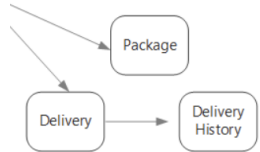
Each service can have different entity state



Use Event Storming to identify potential inconsistencies



Each potential inconsistency concerning?



Adjust service boundaries to resolve inconsistency?



Replace Read After Write with push notification?

Wrap Up

Summary

- Improve Reliability
 - **Asynchronous communications** minimize coupling
 - **Capturing event data** can reduce coupling
 - **Replayability** makes it easier to recover from some issues
- Improve Scalability
 - **Consumer groups** simplify scaling
 - **Parallel processing** across microservices is easier
- Improve Flexibility
 - Event ecosystem makes it **easier to add functionality** without the risks inherent to changing existing functionality
 - Event sourcing makes **time travel simple**



Thank you

Contact:

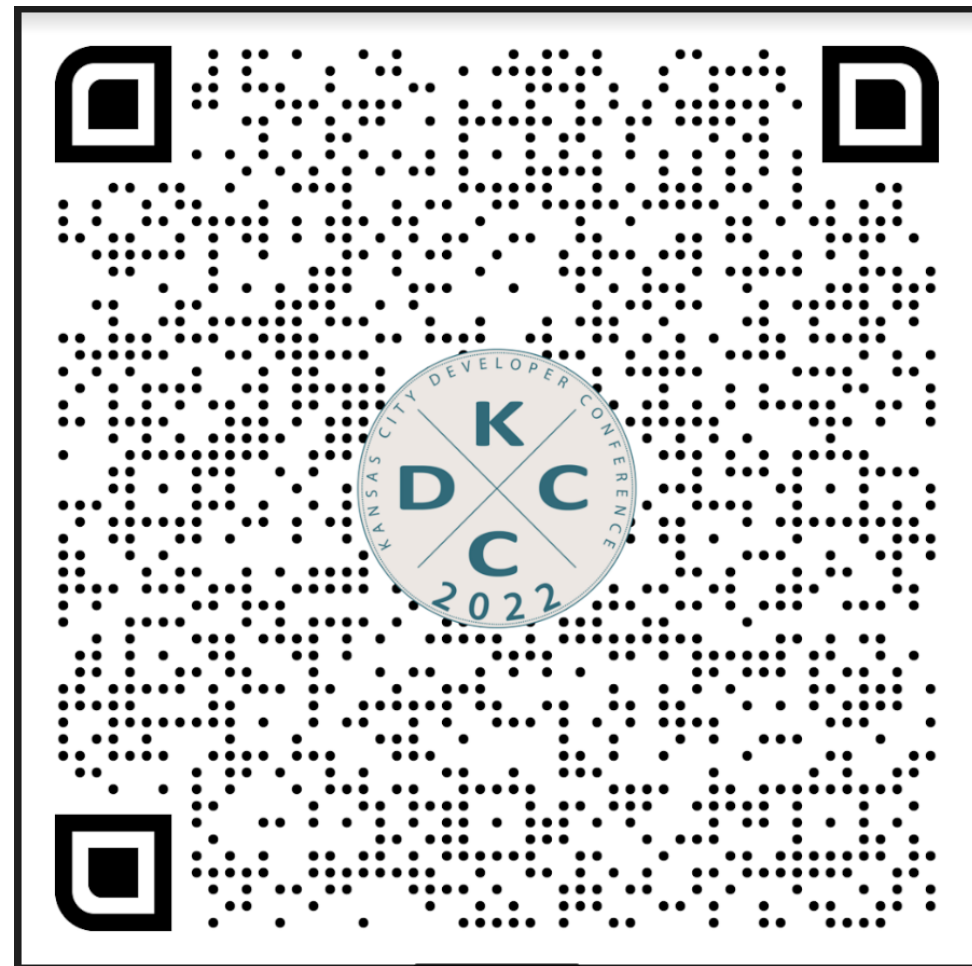
Scott Carter

Principal Software Engineer

scott.carter@wellsky.com

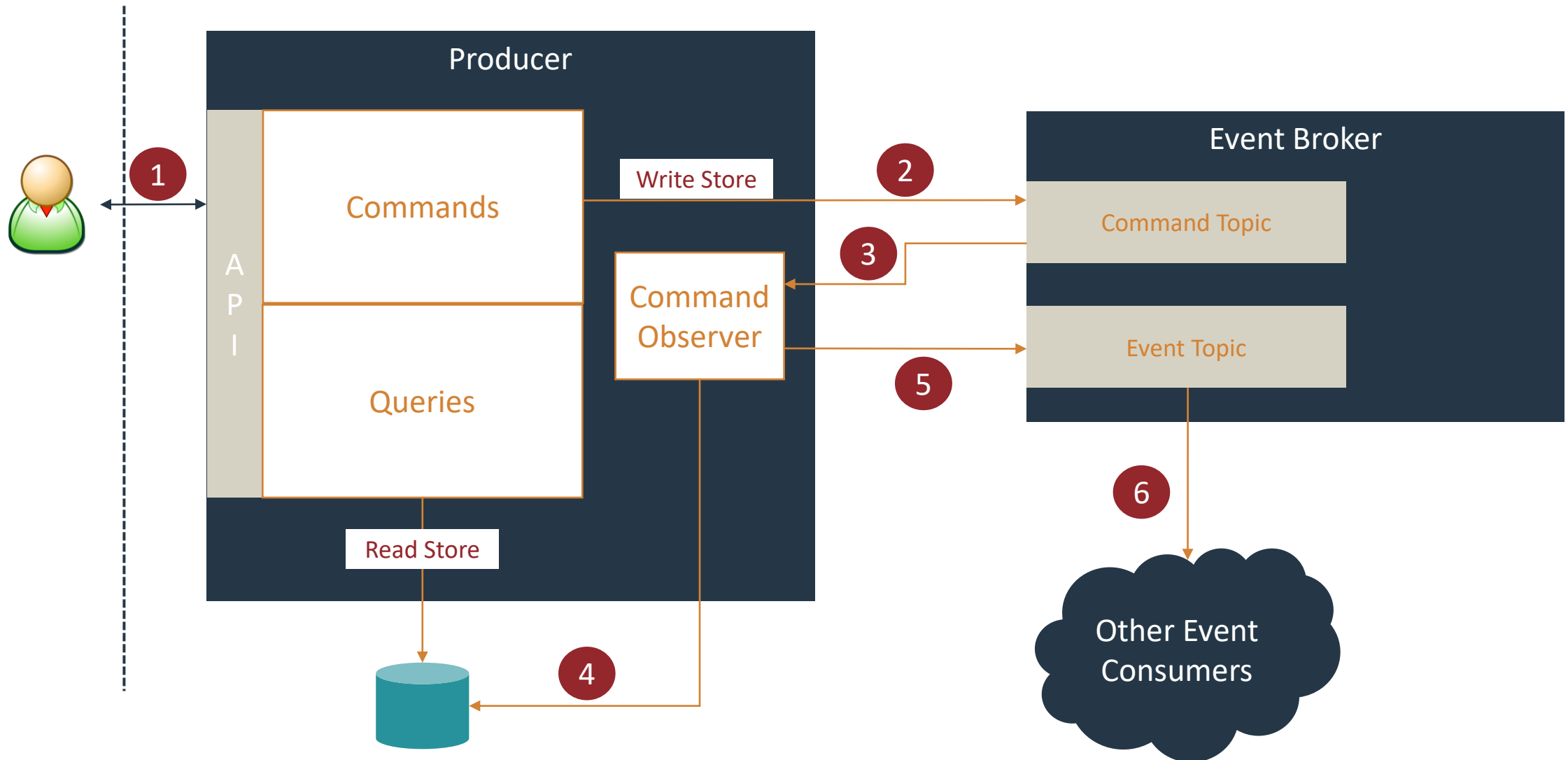
LinkedIn: <https://www.linkedin.com/in/scott-carter/>

Slides: <https://github.com/scottctr/EdaKcdc2022/blob/main/EventDrivenArchitectureBranded.pdf>



Appendix

Producers and Command / Query Responsibility Separation (CQRS)



Warning

“Synchronous communication is the crystal meth of distributed software”

Mark Thomson

Understand Eventual Consistency

