



Monolith to Event-Driven Microservices

Eko Kurniawan Khannedy

Eko Kurniawan



- Senior Principal R&D Engineer at Blibli.com
- 10 Years of Experience in IT Industry
- Software Architect behind:
 - Blibli.com New Backend System
 - Blipay

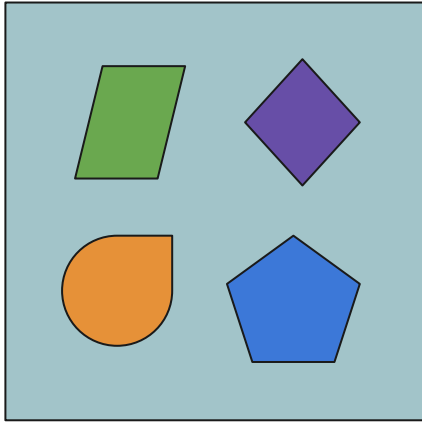


Agenda

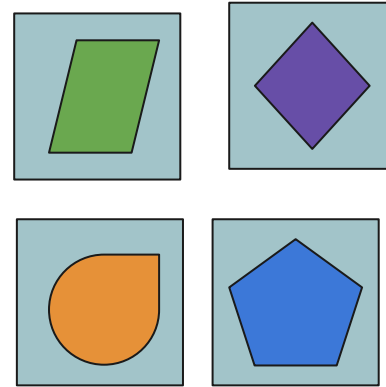
- Microservices
- Event-Driven Architecture
- Hands-On using Java

Microservices

Monolith vs Microservices Application

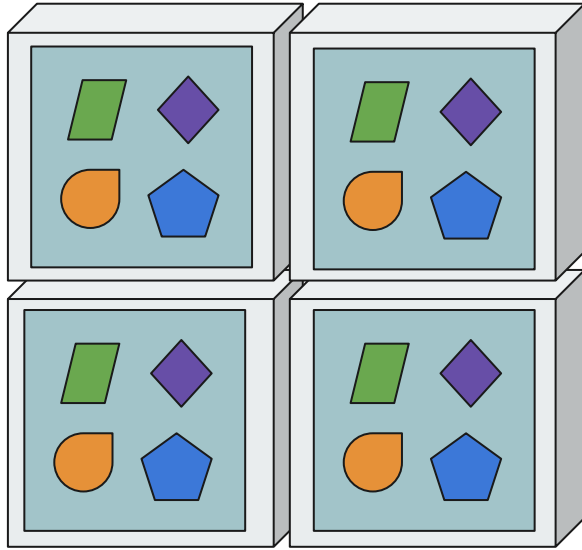


Monolith

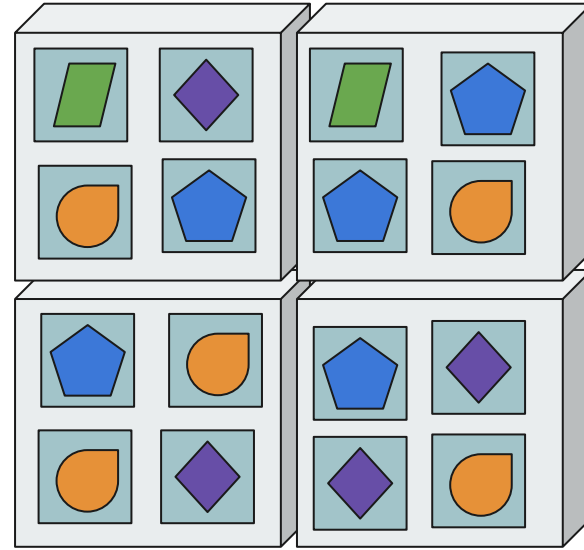


Microservices

Monolith vs Microservices Deployment



Monolith



Microservices

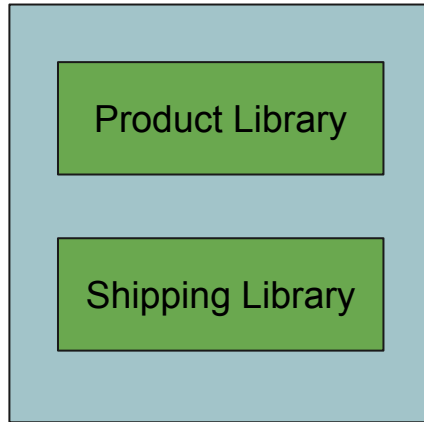


Characteristic of Microservices

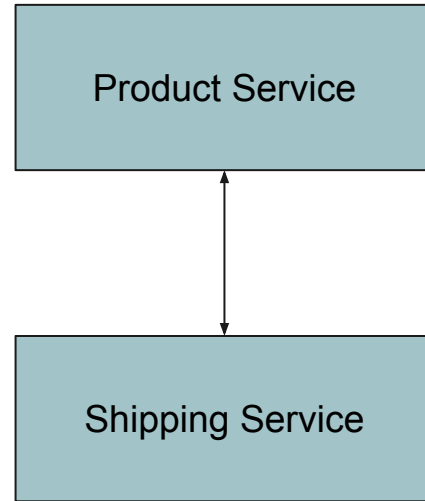
1. Componentization via Services
2. Organized around Business Capabilities
3. Products not Projects
4. Smart endpoints and dumb pipes
5. Decentralized Governance
6. Decentralized Data Management
7. Infrastructure Automation
8. Design for failure
9. Evolutionary Design

<https://martinfowler.com/articles/microservices.html>

Componentization via Services



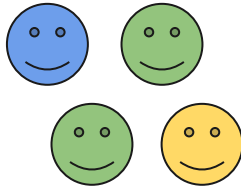
Library



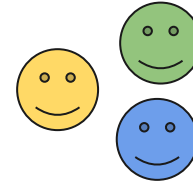
Service

Organized around Business Capabilities

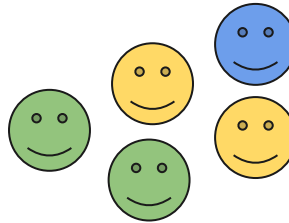
Merchant



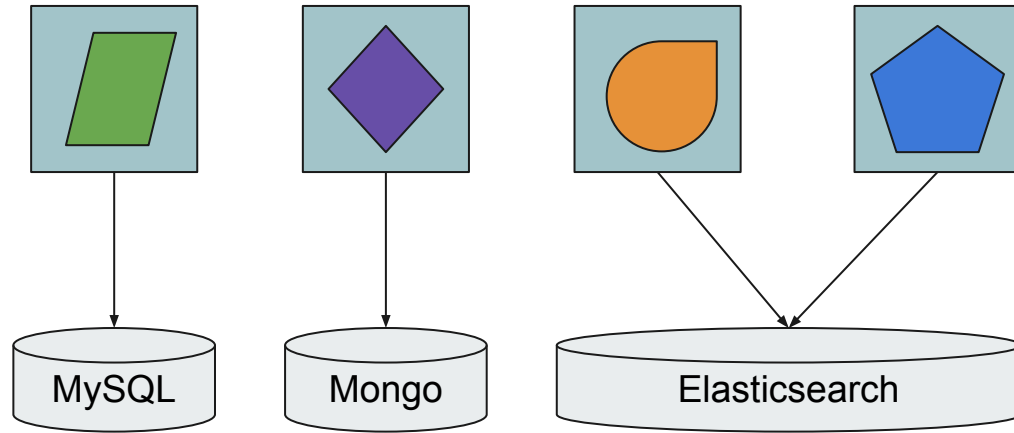
Shipping



Product

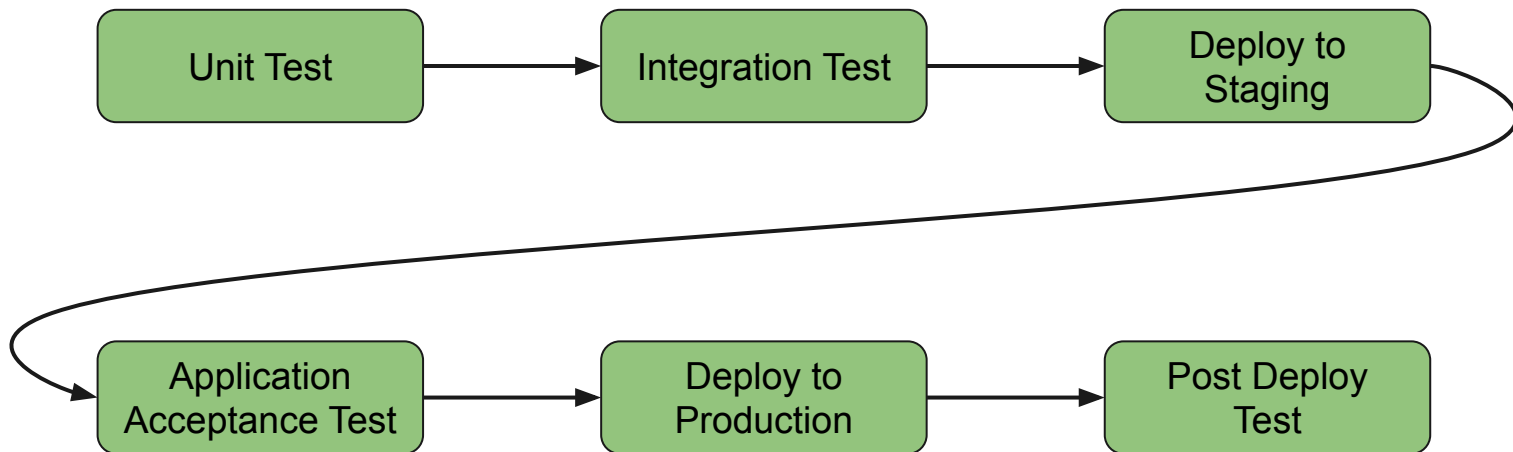


Decentralized Data Management





Infrastructure Automation



Continuous Delivery



Design for failure

- Monitoring
- Circuit Breaker
- Chaos Monkey



How Big is Microservices?

- Single Responsibility
- Small enough to fit in in my head
- X people per Service



Monolith

- Simplicity
- Consistency
- Easy to Refactor

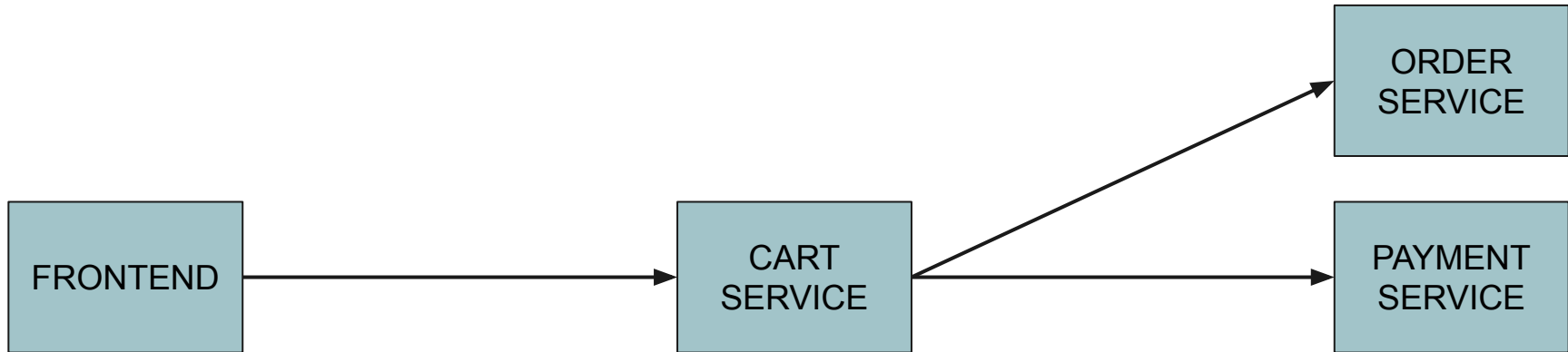
Microservices

- Partial Deployment
- Availability
- Multiple Platform
- Easy to Scale

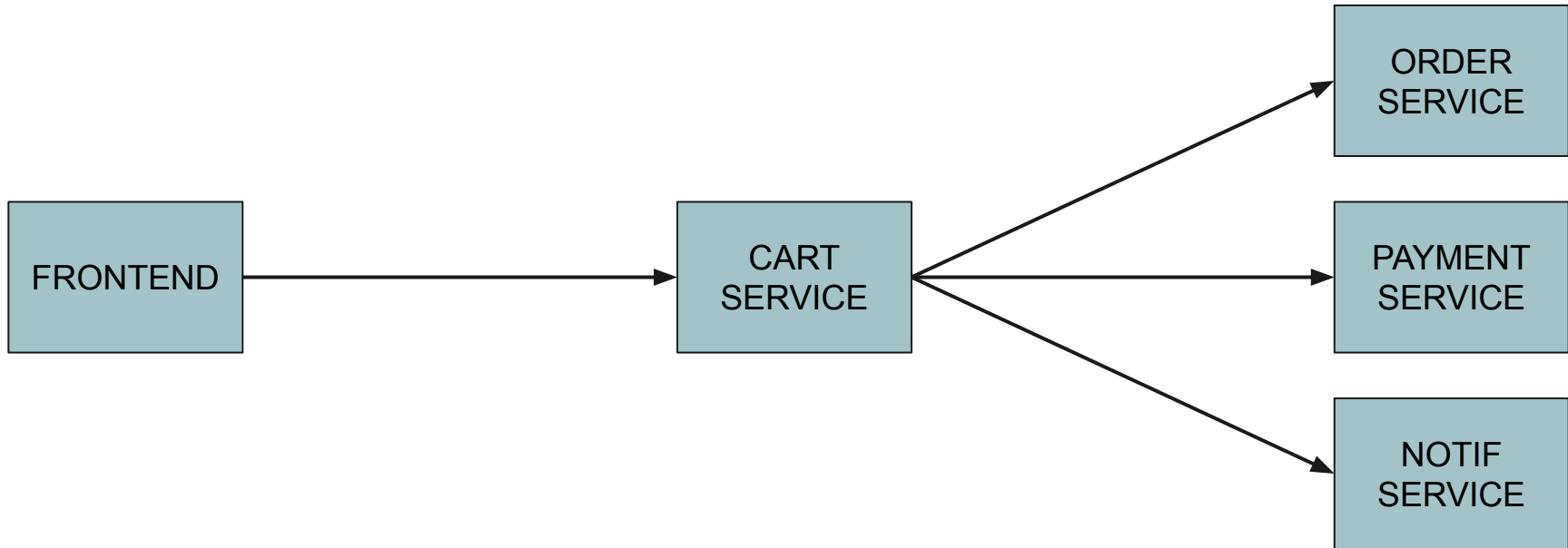
Event-Driven Architecture



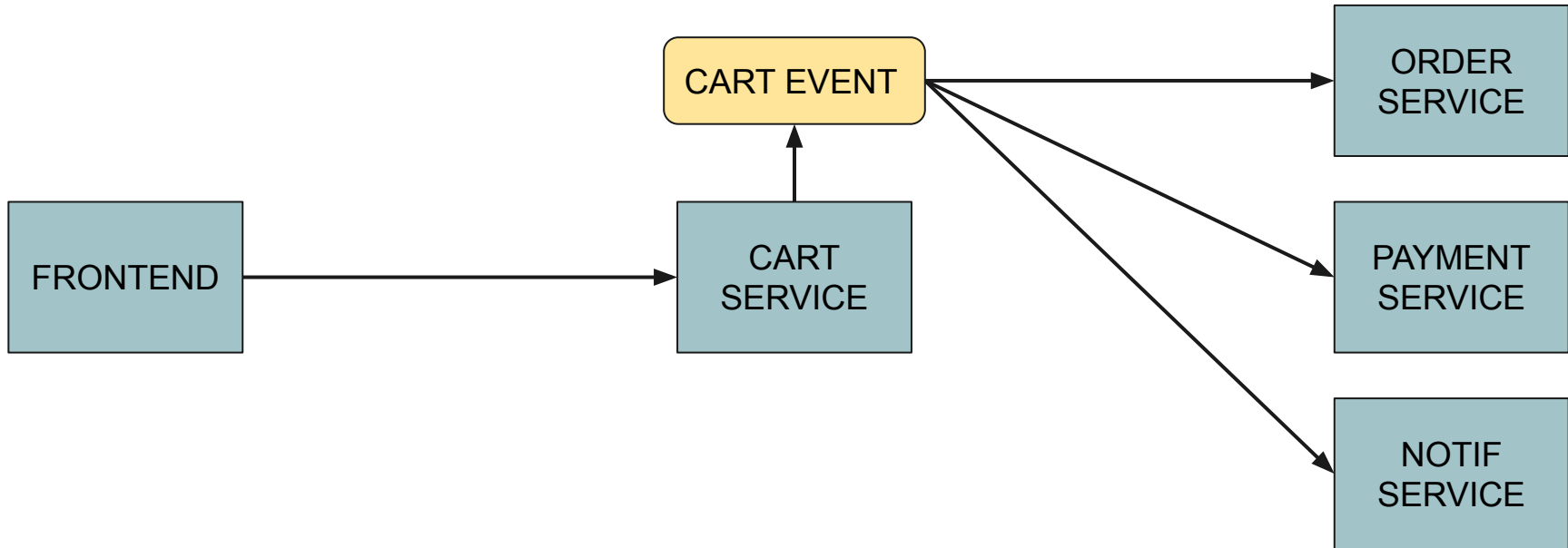
API-Driven Microservices



API-Driven Microservices Problem

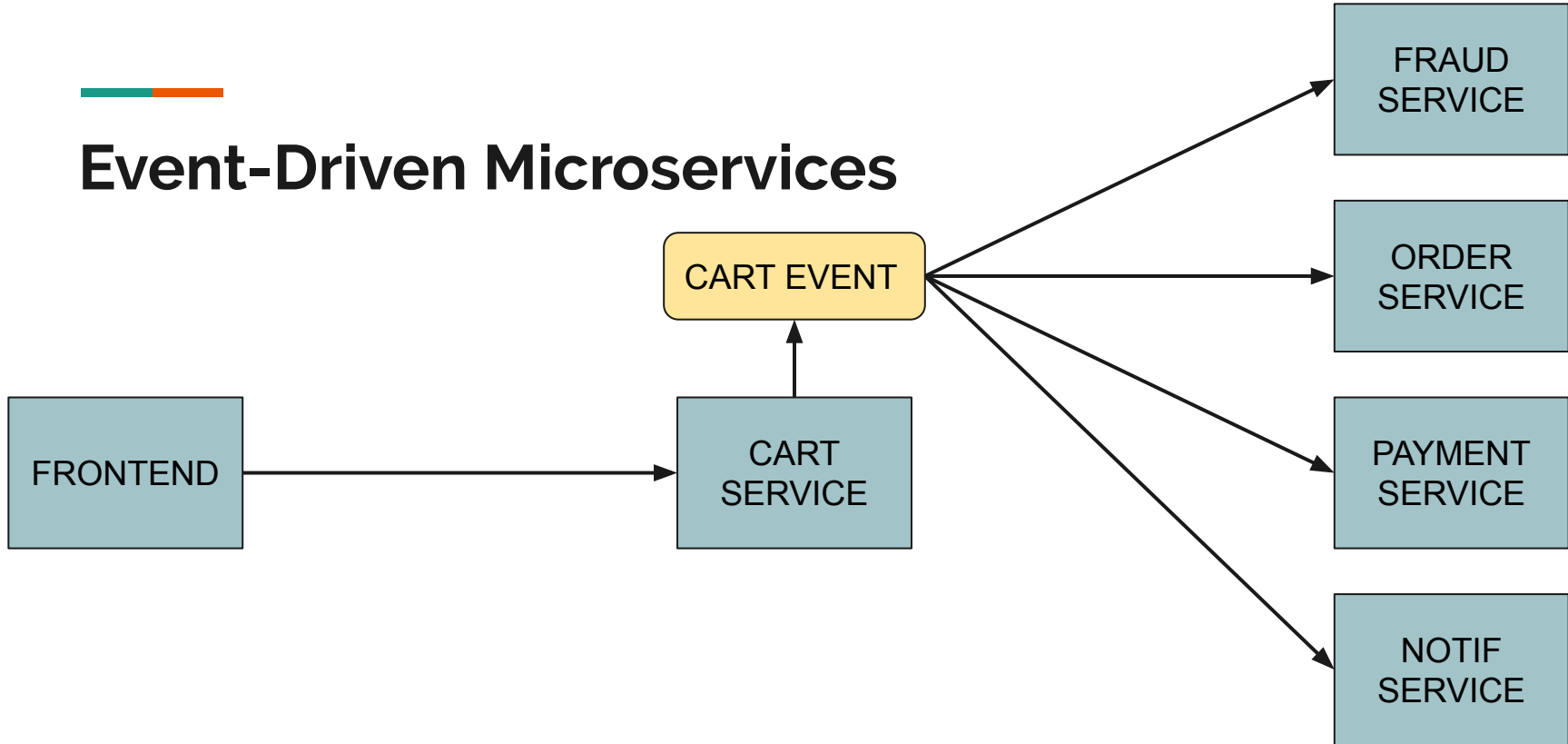


Event-Driven Microservices

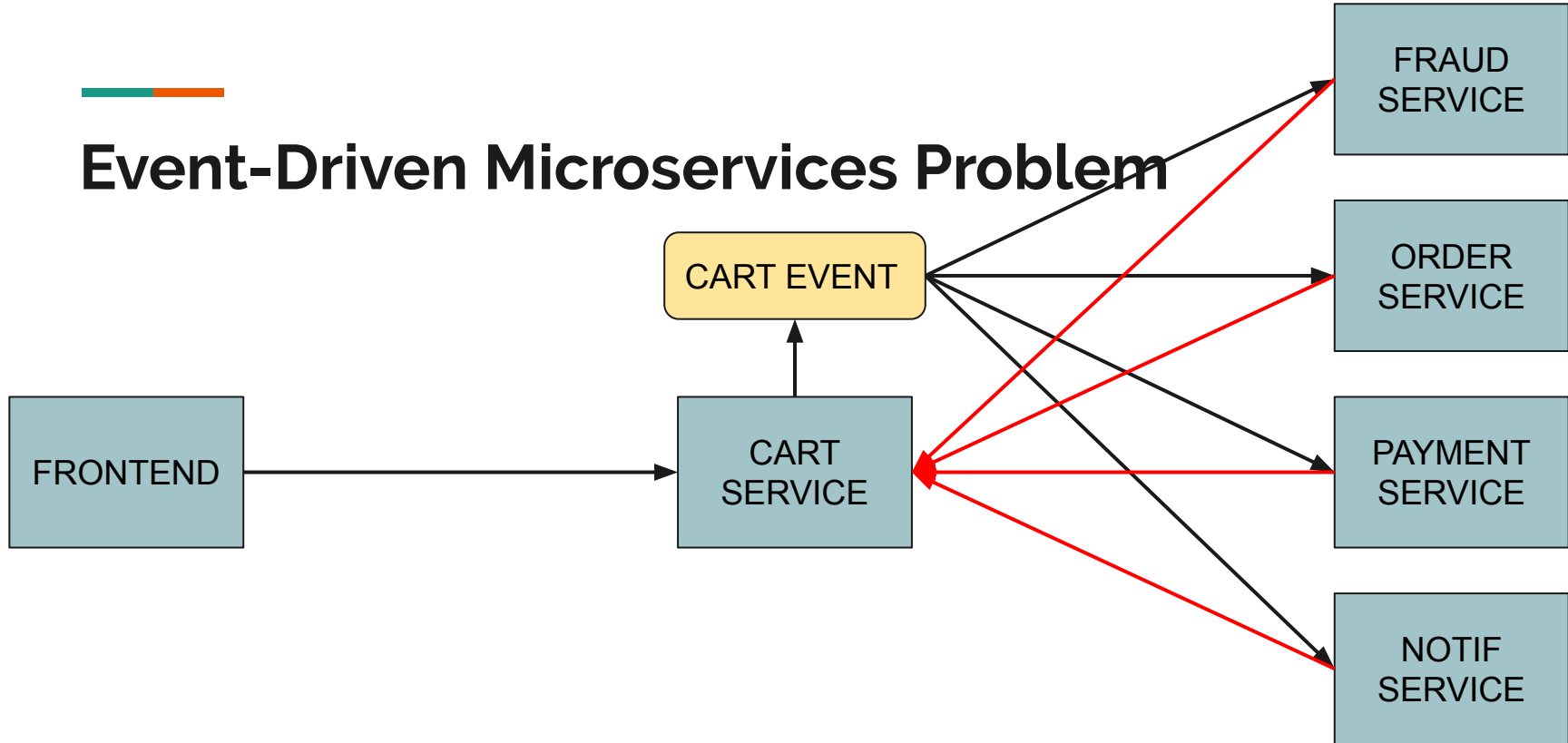




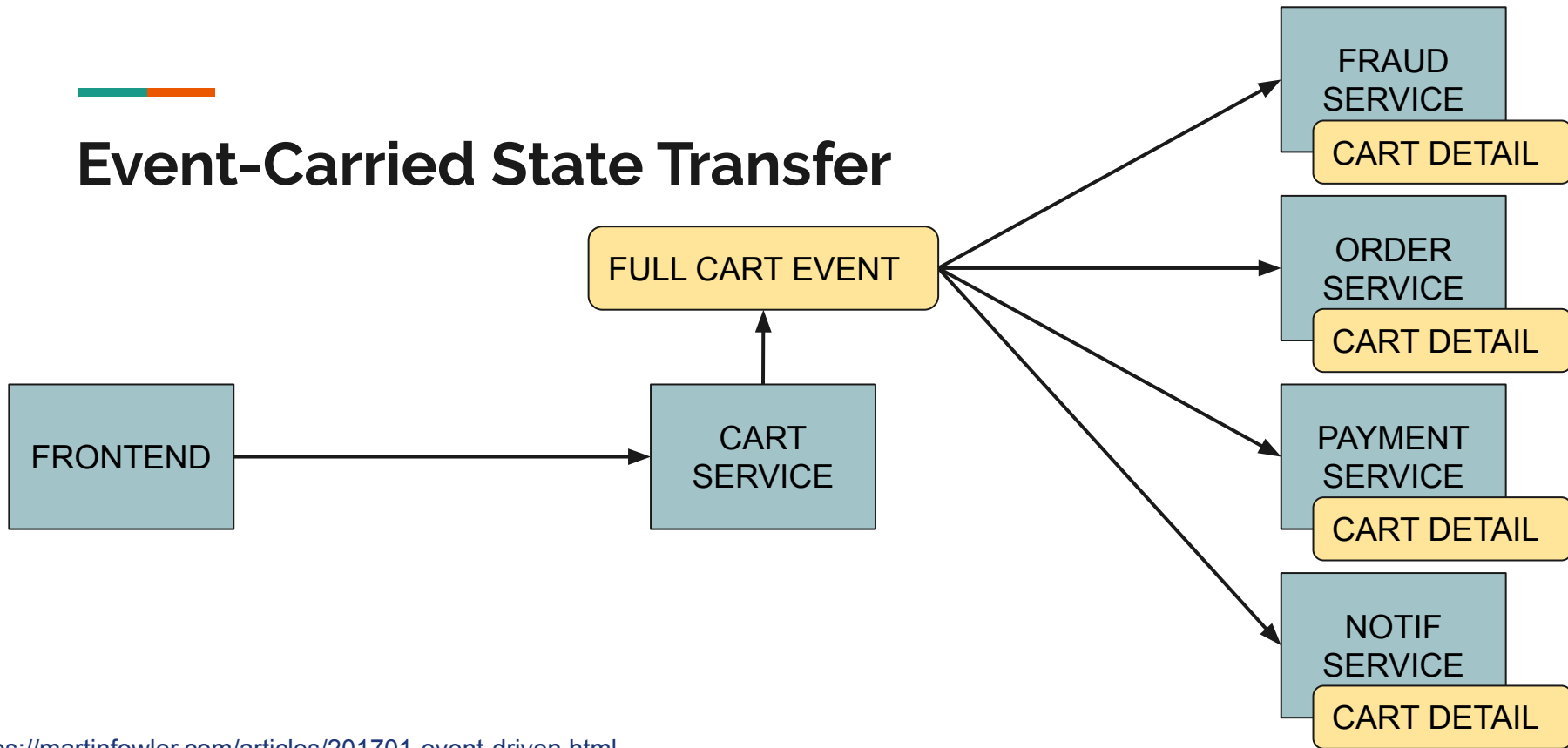
Event-Driven Microservices



Event-Driven Microservices Problem



Event-Carried State Transfer





API-Driven

- Highly-coupled
- Centralized business logic
- Consistent
- Increase response time
- Sync

Event-Driven

- Decoupling
- Decentralized business logic
- Eventual consistent
- Reduce load on producer
- Replicated data
- Async



The Challenges of Event-Driven Microservices

- Idempoten
- Asynchronous
- Message Schema
- Versioning
- Tracing
- Message Order
- Consumer Complexity
- Message Broker

Hands-On using Java



Tools Required for This Workshop

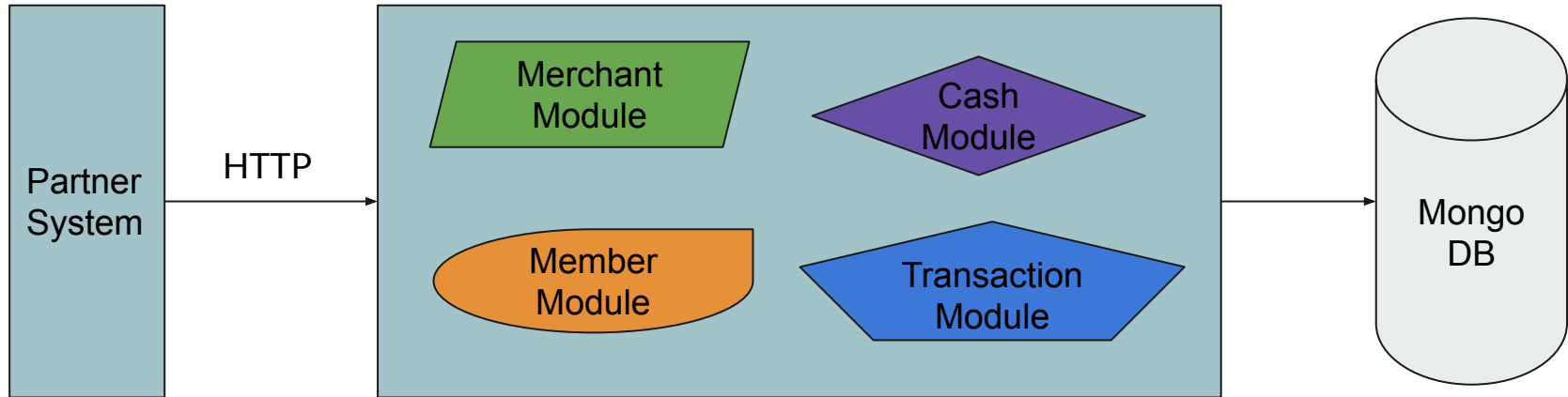
- Java Development Kit version 1.8 or Higher
- Apache Maven version 3
- Apache Kafka version 2.3
- MongoDB version 4
- Elasticsearch version 7.2



MonoPay

- MonoPay is whitelabel system for e-Wallet
- Merchant can use MonoPay to support e-Wallet feature in their system
- MonoPay support member management, topup management, purchase management and point management
- MonoPay use monolith architecture

MonoPay Backend System



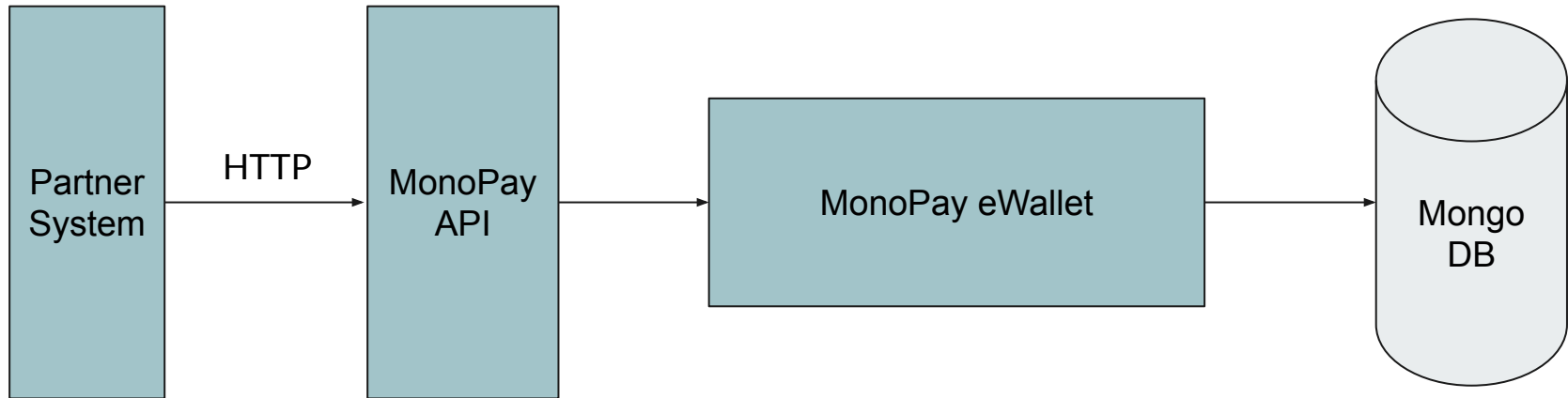


MonoPay Wallet Project

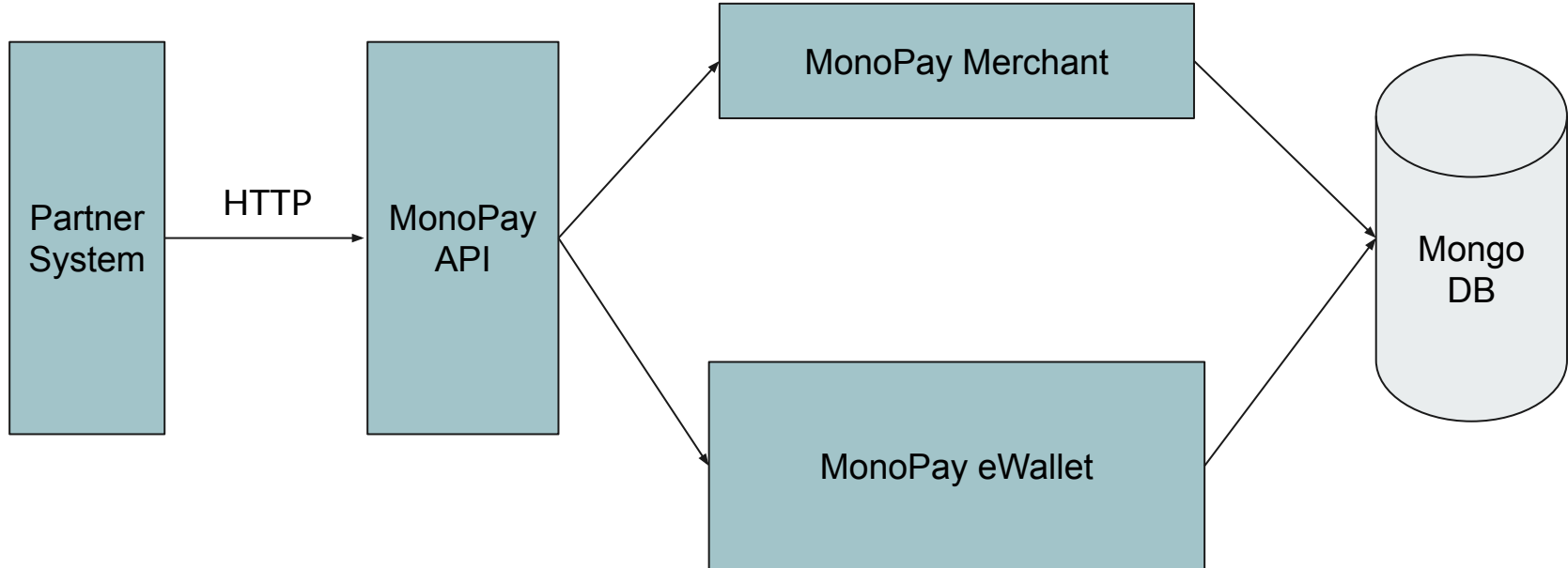
<https://github.com/khannedy/monopay-wallet>

How to migrate MonoPay to Event-Driven Microservices?

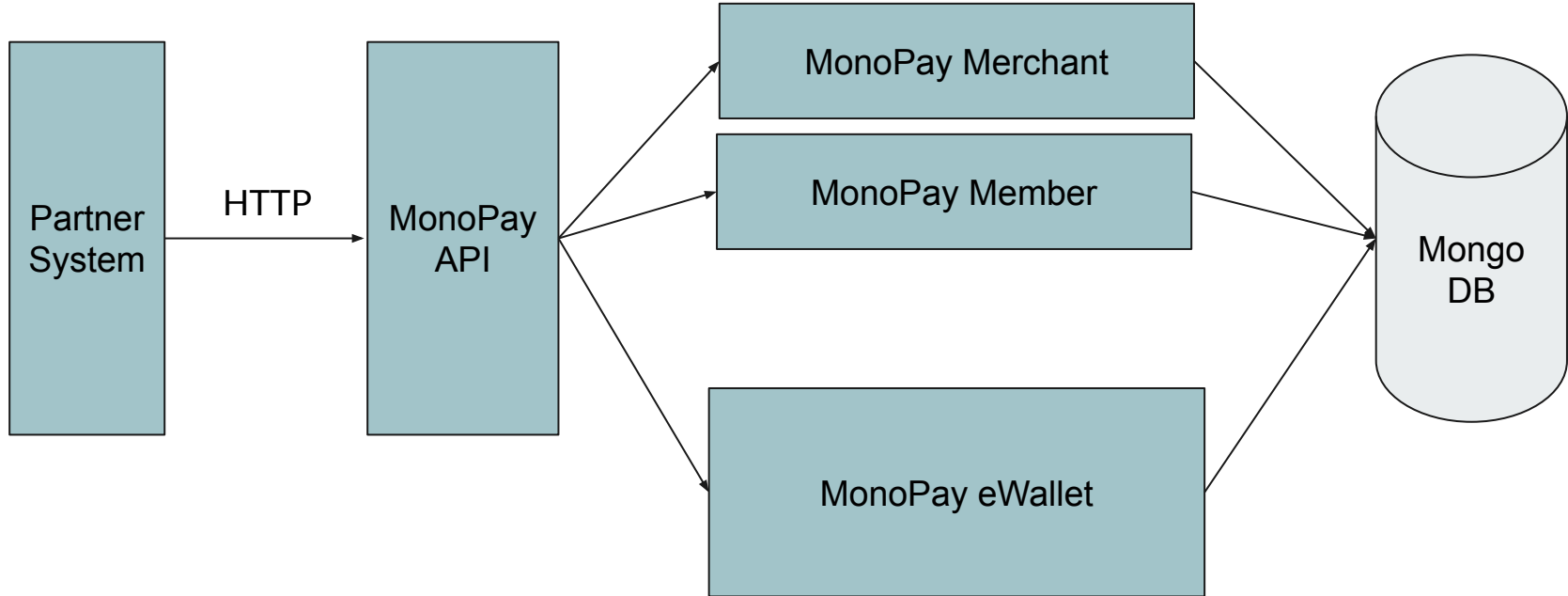
Migrating MonoPay to Microservices



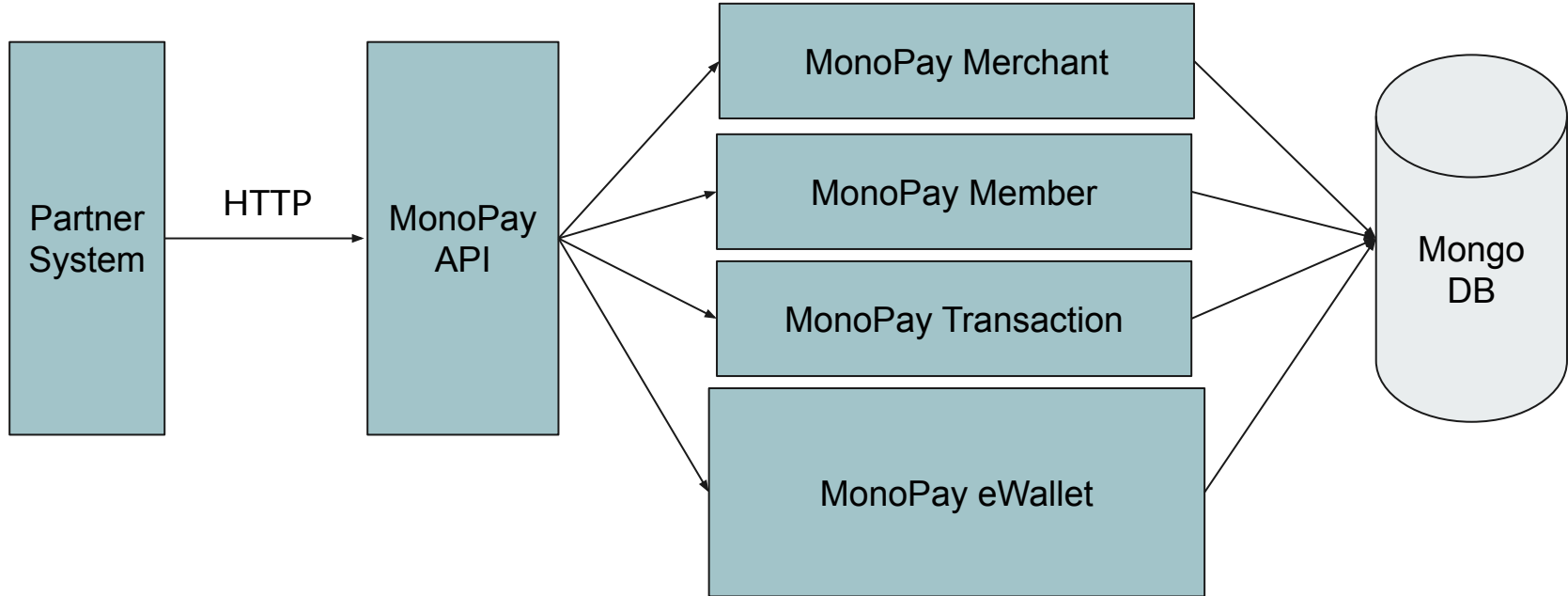
Migrating MonoPay to Microservices



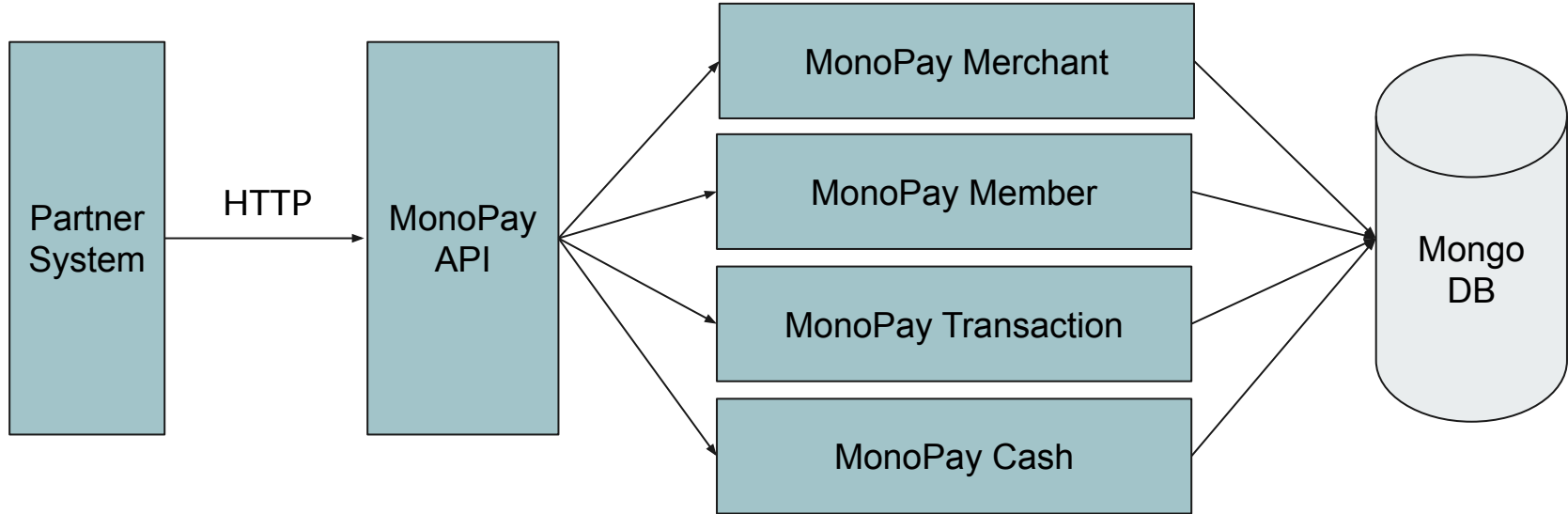
Migrating MonoPay to Microservices



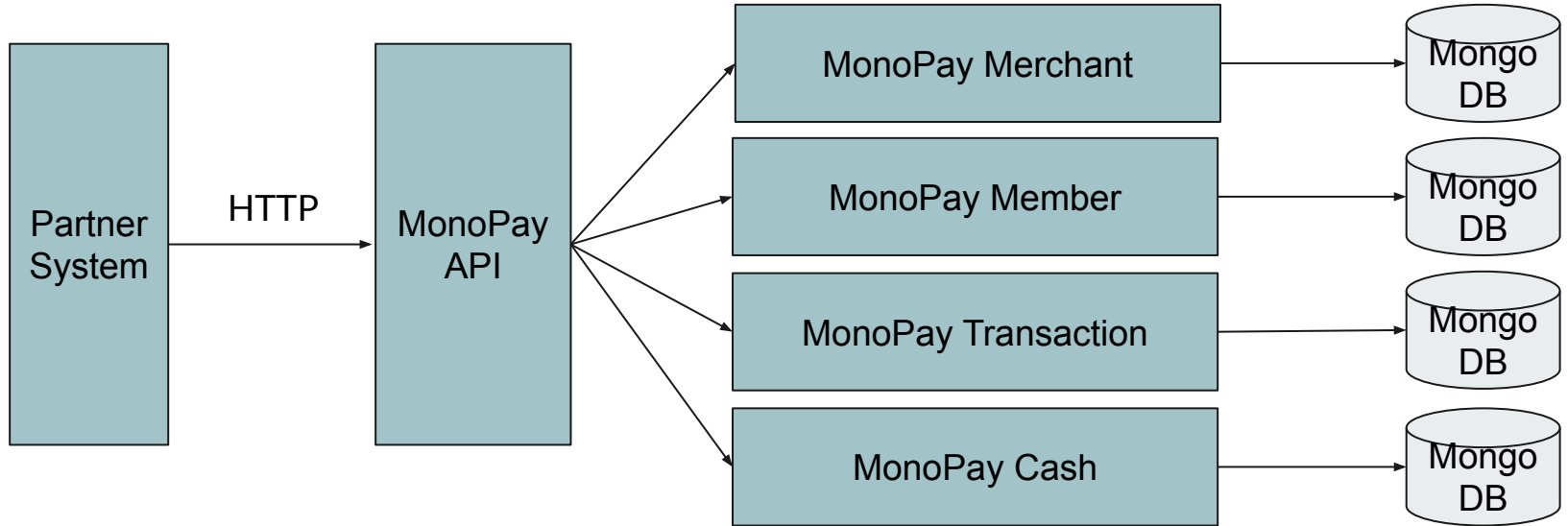
Migrating MonoPay to Microservices



Migrating MonoPay to Microservices



Migrating MonoPay to Microservices



How to Integrate between Services?

API-Driven vs Event-Driven



MonoPay Microservices

<https://github.com/khannedy/monopay-api>

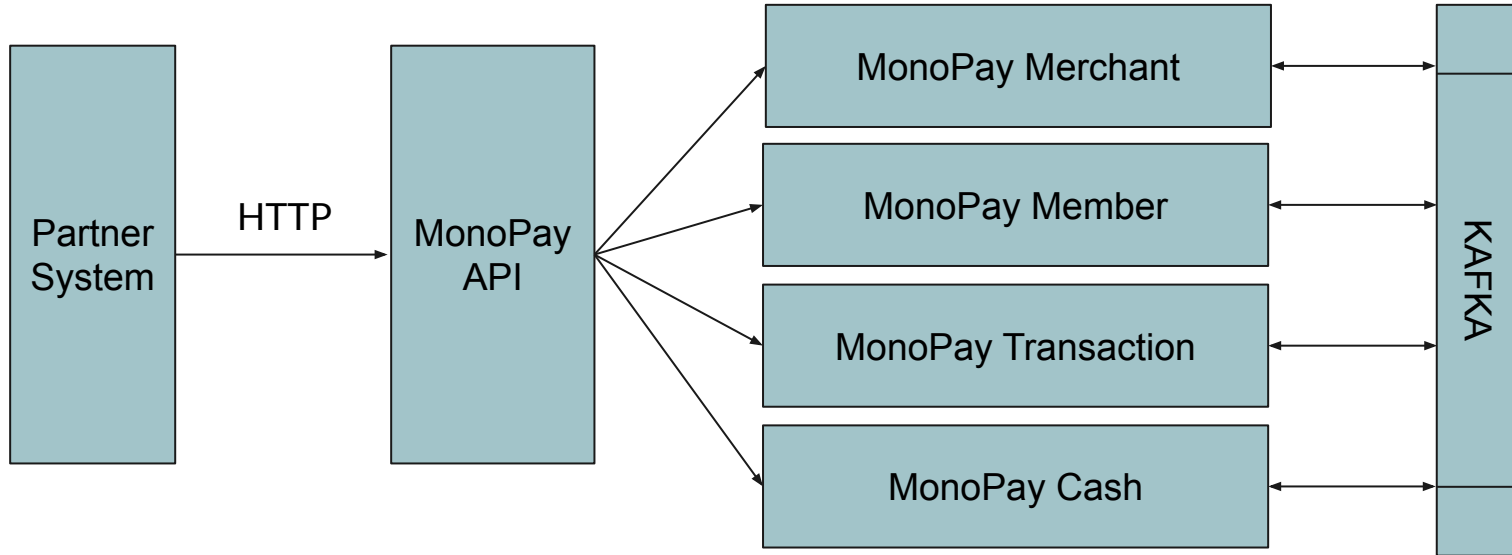
<https://github.com/khannedy/monopay-merchant>

<https://github.com/khannedy/monopay-member>

<https://github.com/khannedy/monopay-transaction>

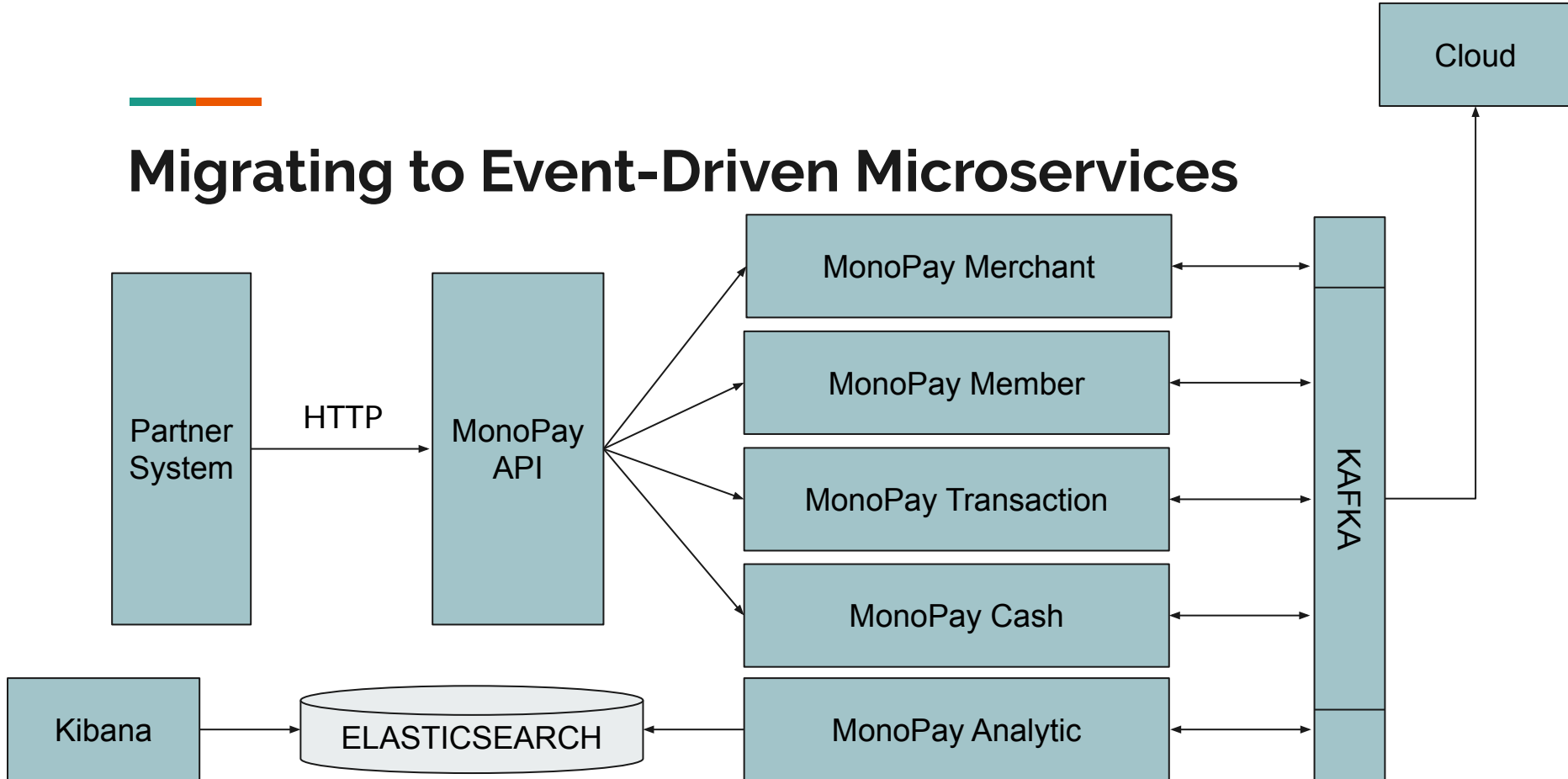
<https://github.com/khannedy/monopay-cash>

Migrating to Event-Driven Microservices



One more thing

Migrating to Event-Driven Microservices





MonoPay Microservices

<https://github.com/khannedy/monopay-analytic>



Thank You!