

Beyond Microservices: Streams, State and Scalability

Gwen Shapira, Engineering Manager
[@gwenshap](https://twitter.com/gwenshap)

In the beginning...



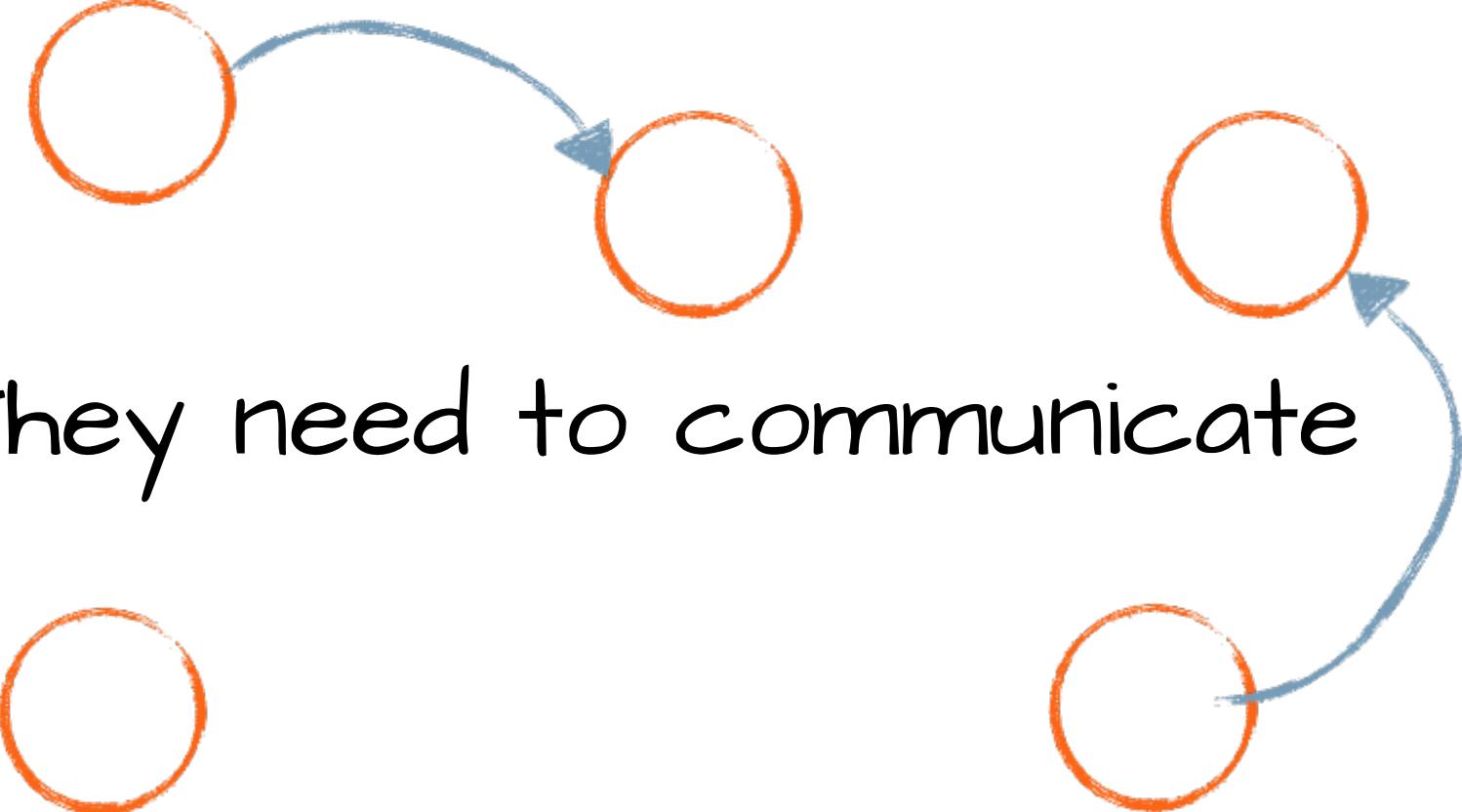
We Have Microservices





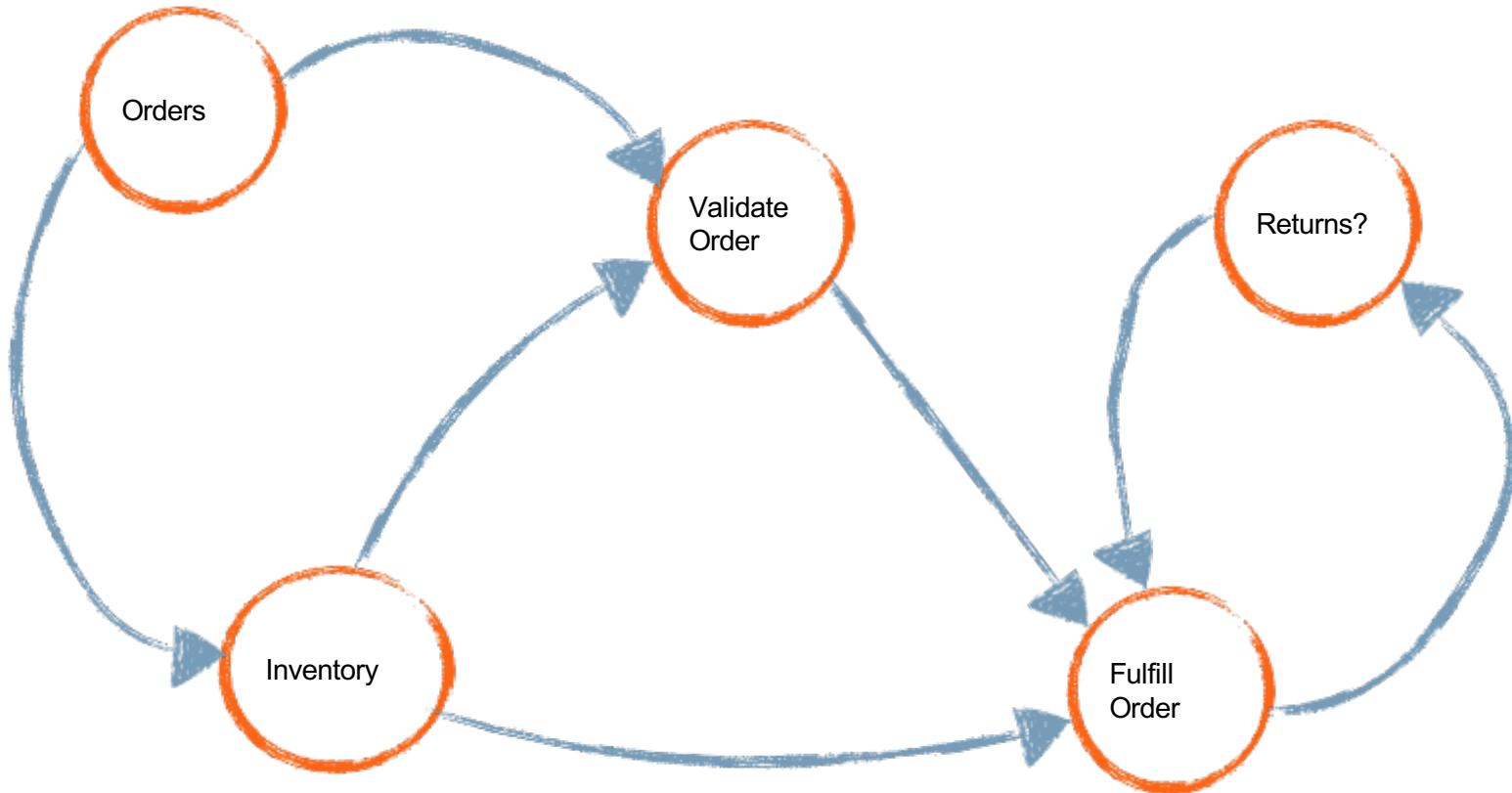
We ❤️ Microservices





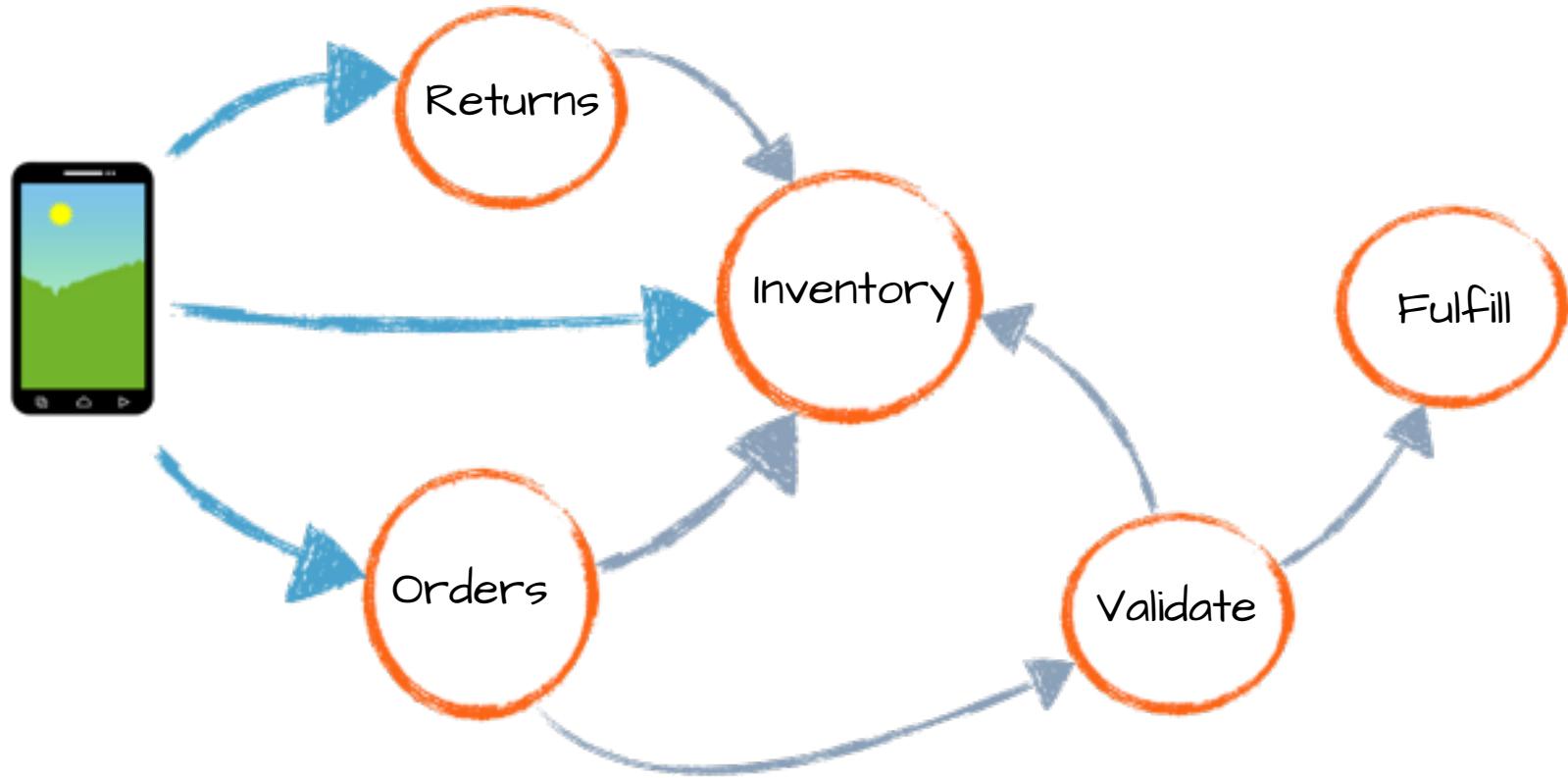
They need to communicate

I know! I'll use REST APIs

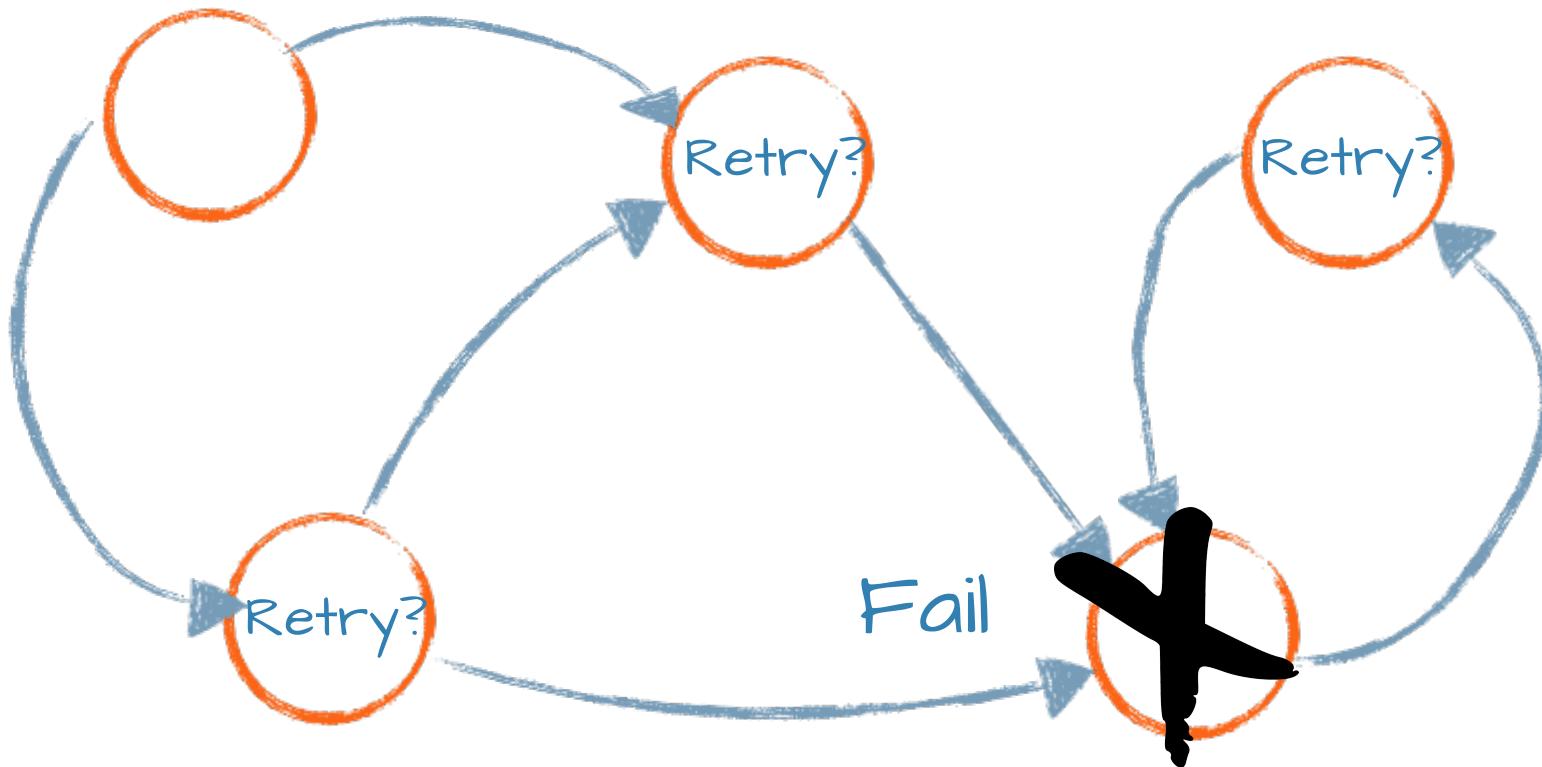


Synchronous request-response communication
Leads to
Tight point-to-point coupling

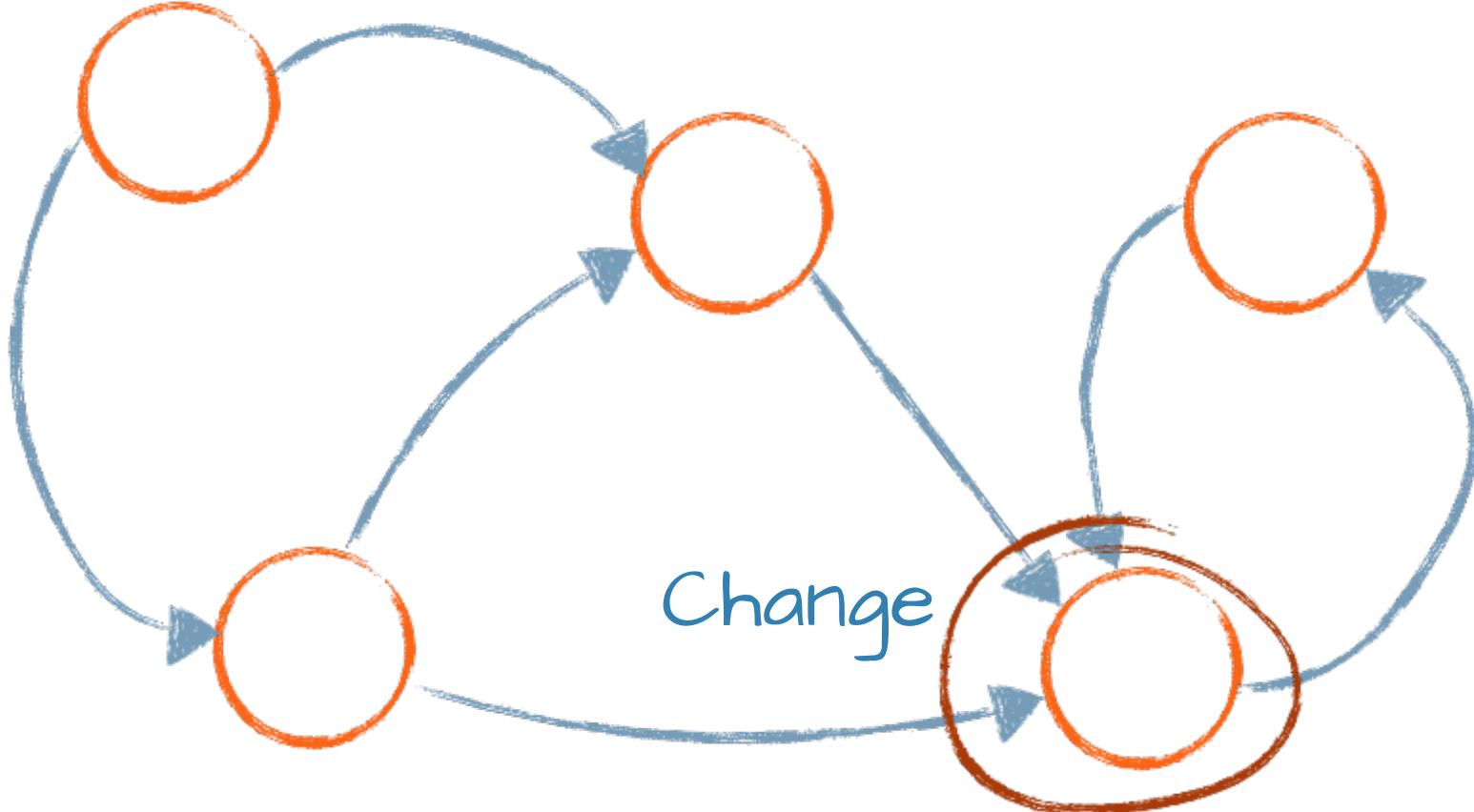
Clients know too much



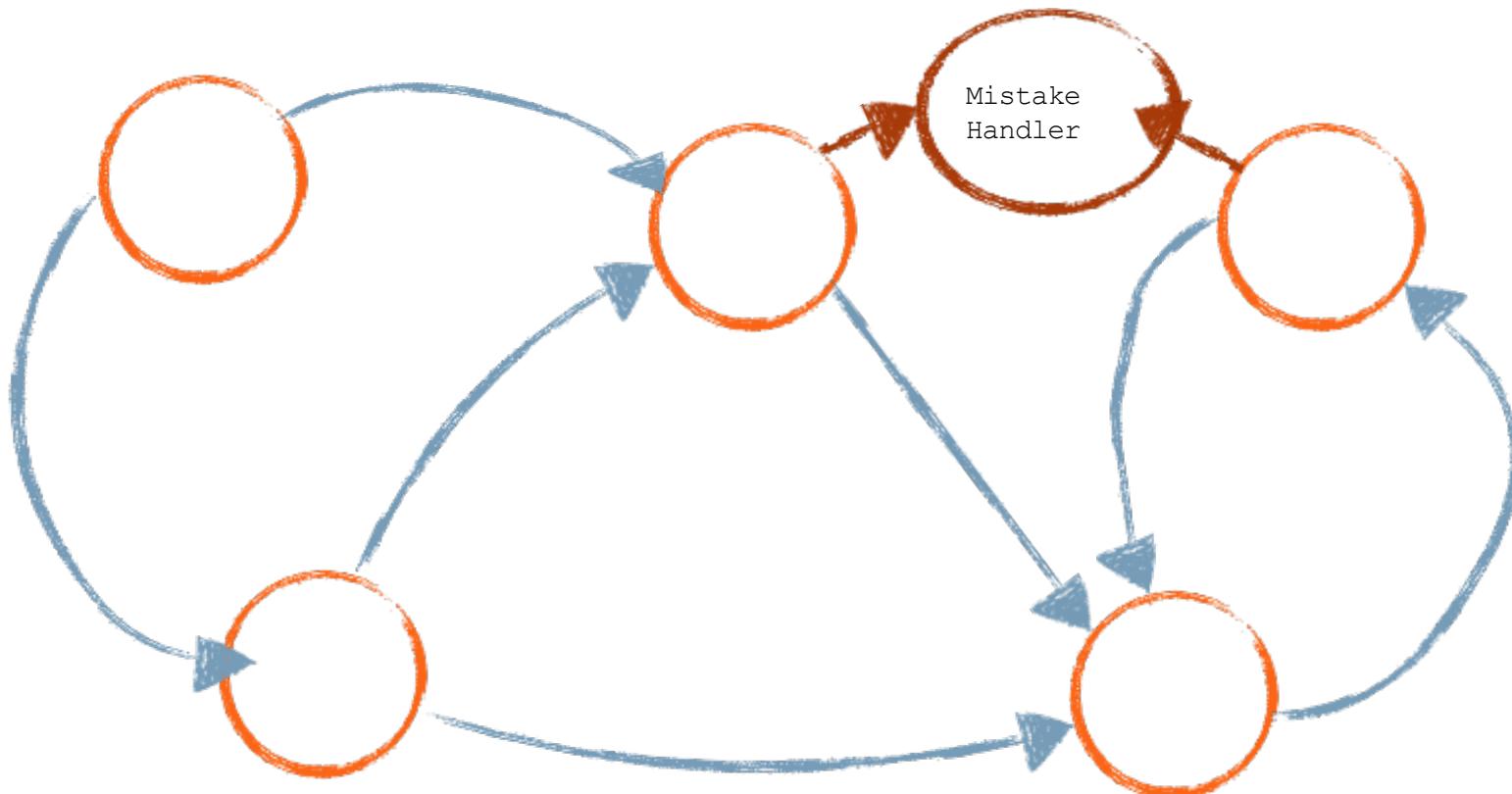
shifts in responsibility, redundancy



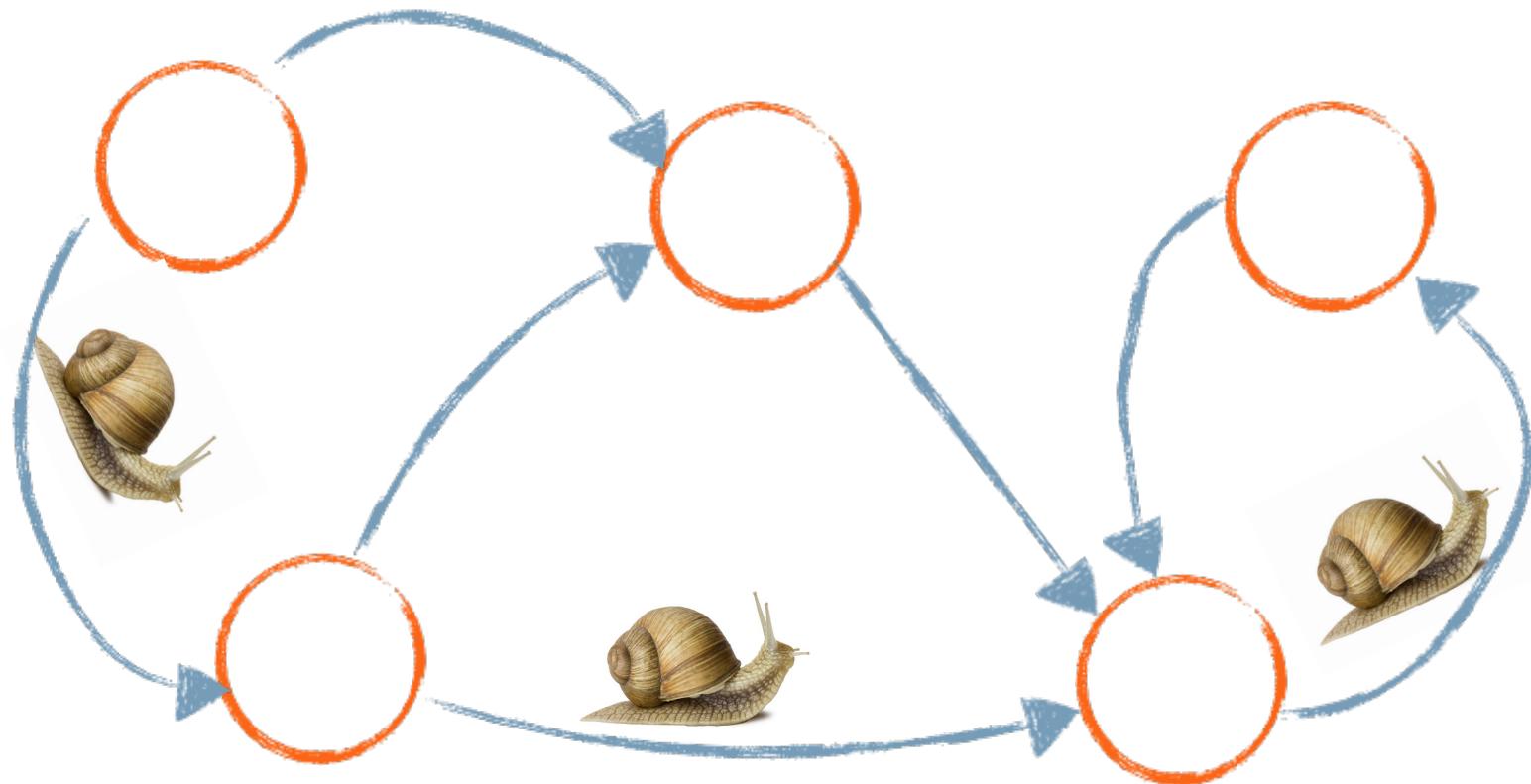
Making Changes is Risky



Adding Services Requires Explicit Calls



REST = HTTP + JSON = SLOW



We can do better.

Nice to meet you!

- Moving data around for 20 years
- Engineering Manager at Confluent
- Apache Kafka Committer
- Wrote a book or two
- Tweets a lot - @gwenshap



Kafka The Definitive Guide

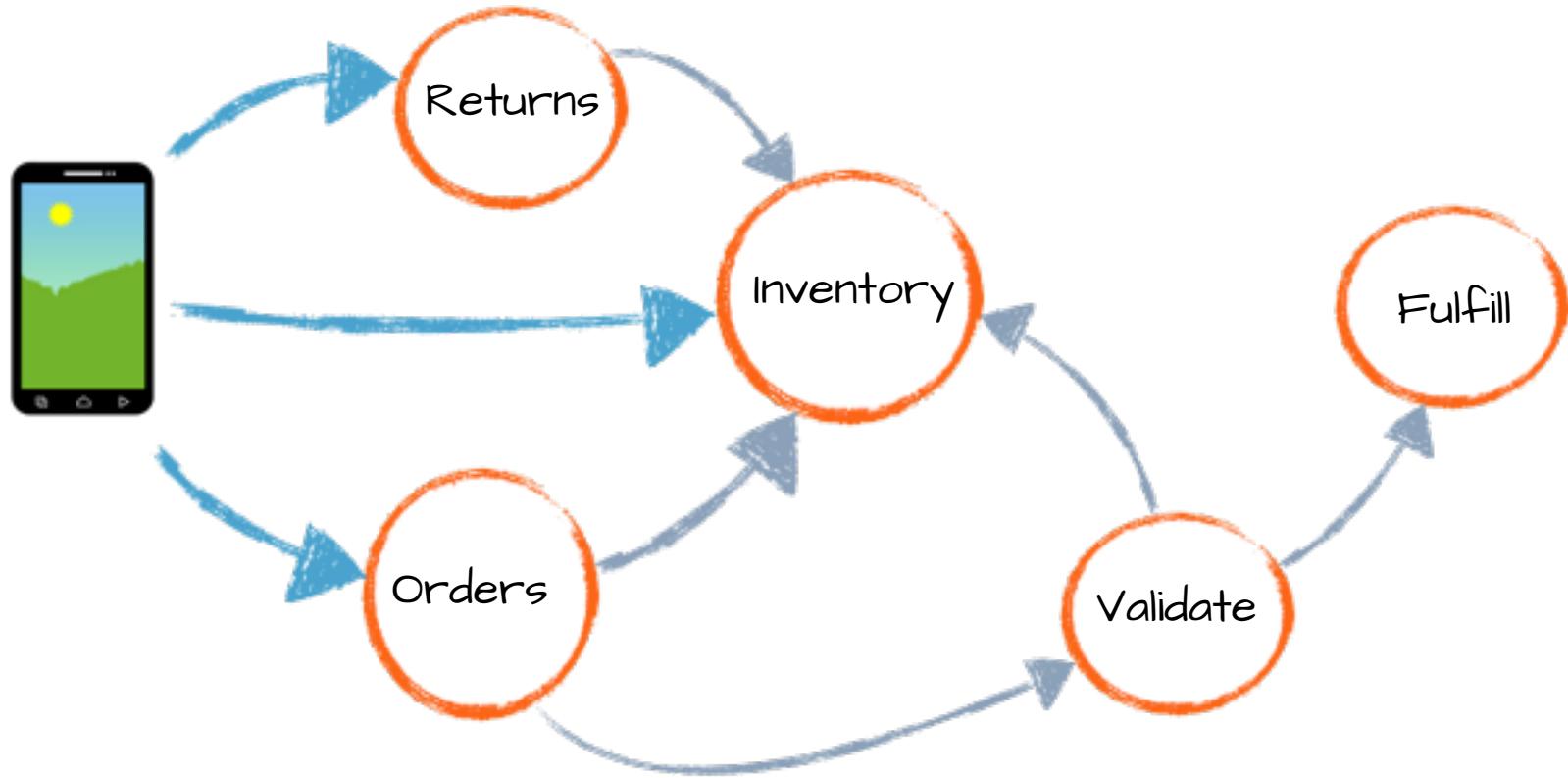
REAL-TIME DATA AND STREAM PROCESSING AT SCALE

Neha Narkhede,

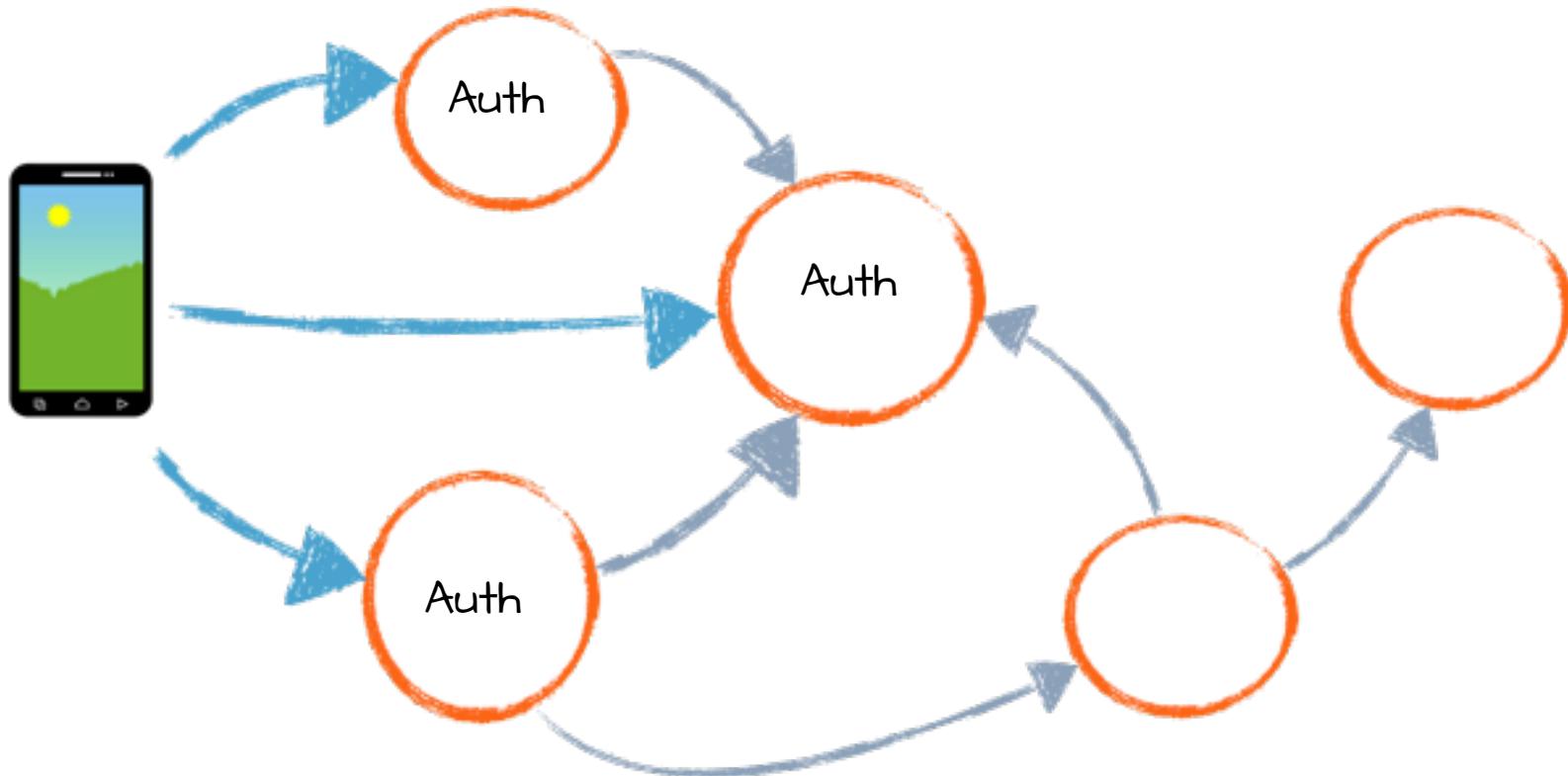
Gwen Shapira & Todd Palino

API Gateway

Clients know too much

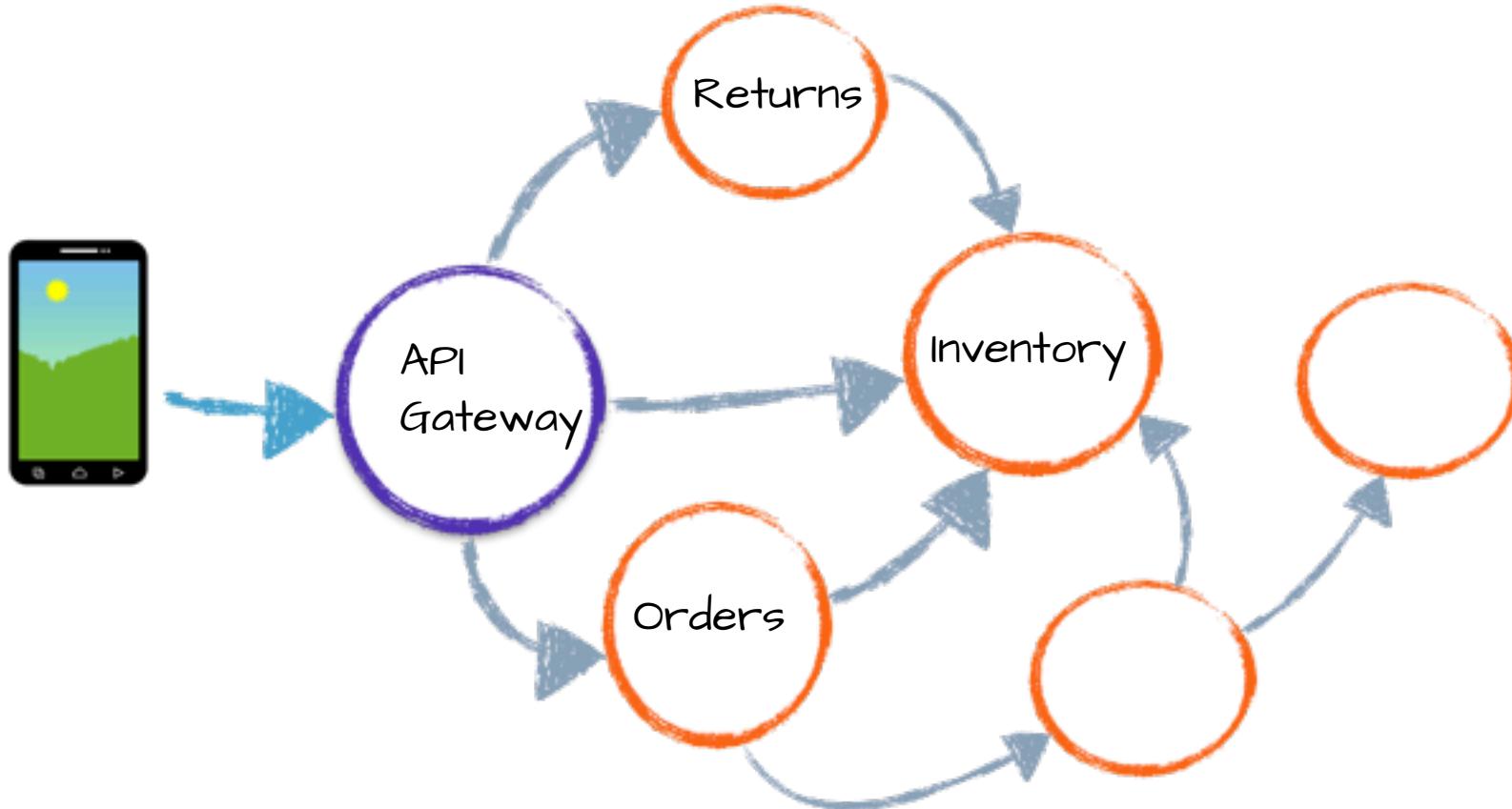


Shift in responsibility, redundancy



API Gateway

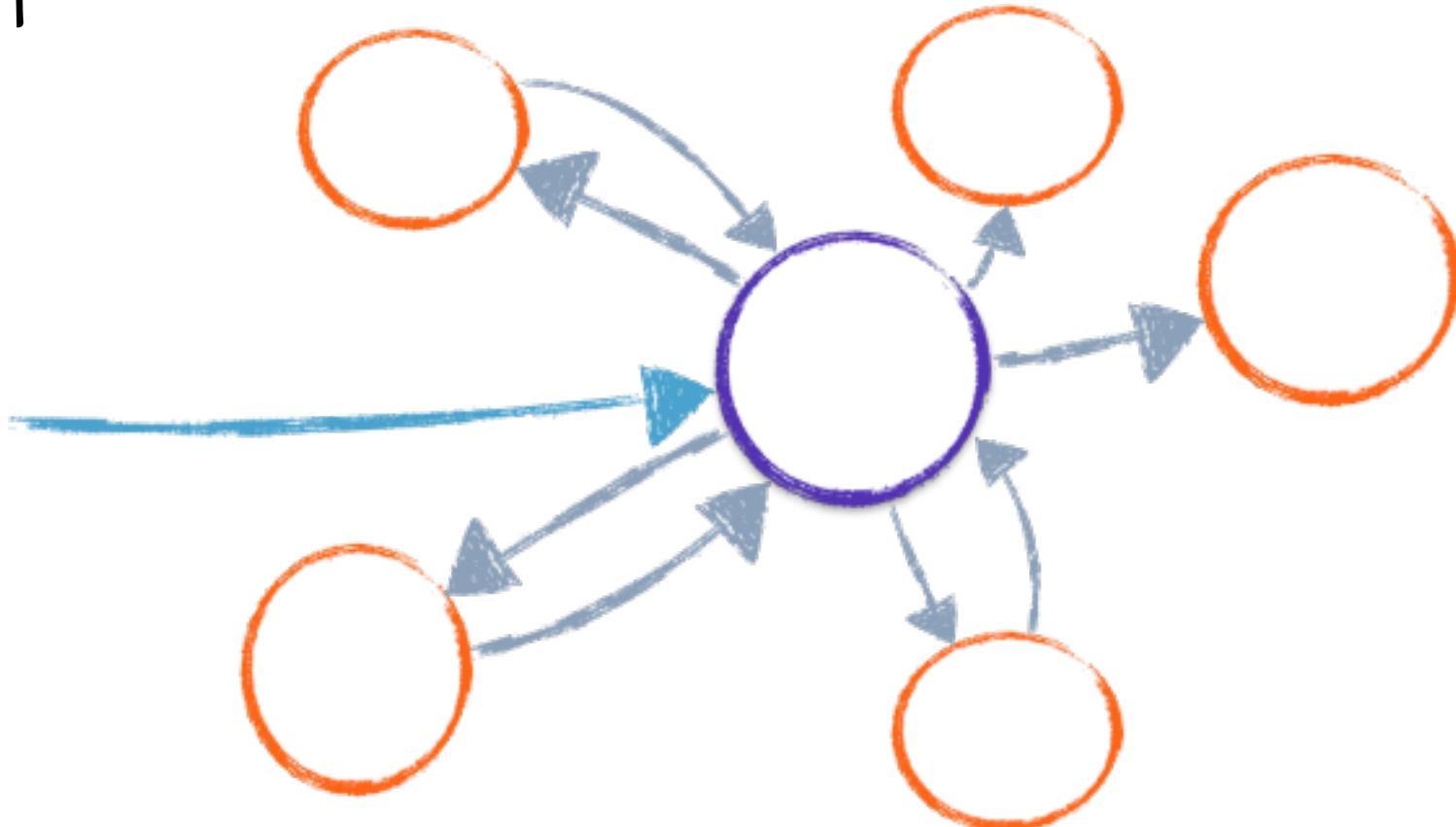
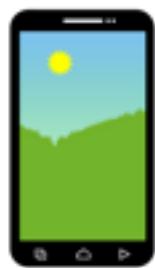
X



API Gateway Responsibility

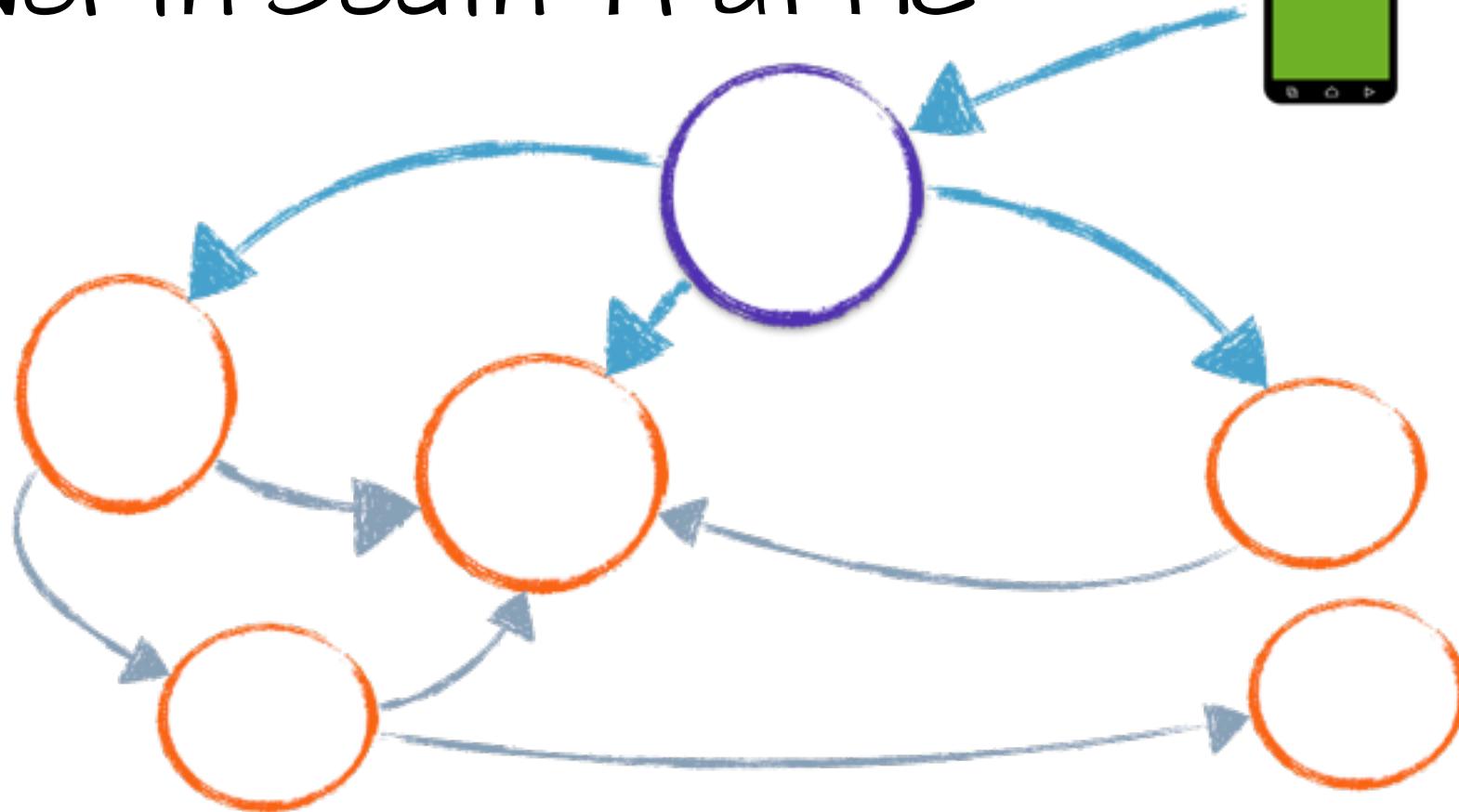
- ● Authentication
- ● Routing
- ● Rate Limiting
- ● Logging and analytics

Anti-pattern

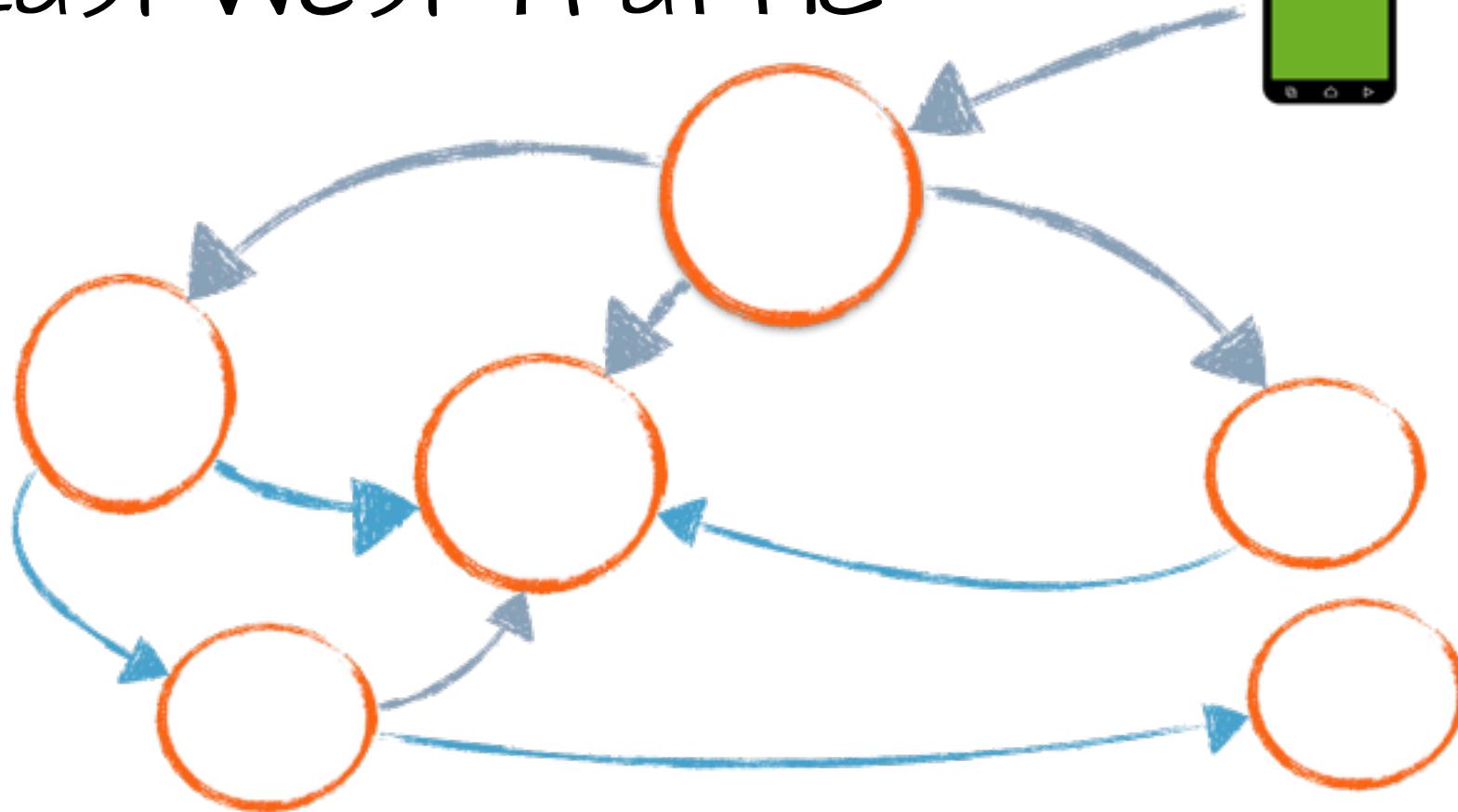


Service Mesh

North-South Traffic



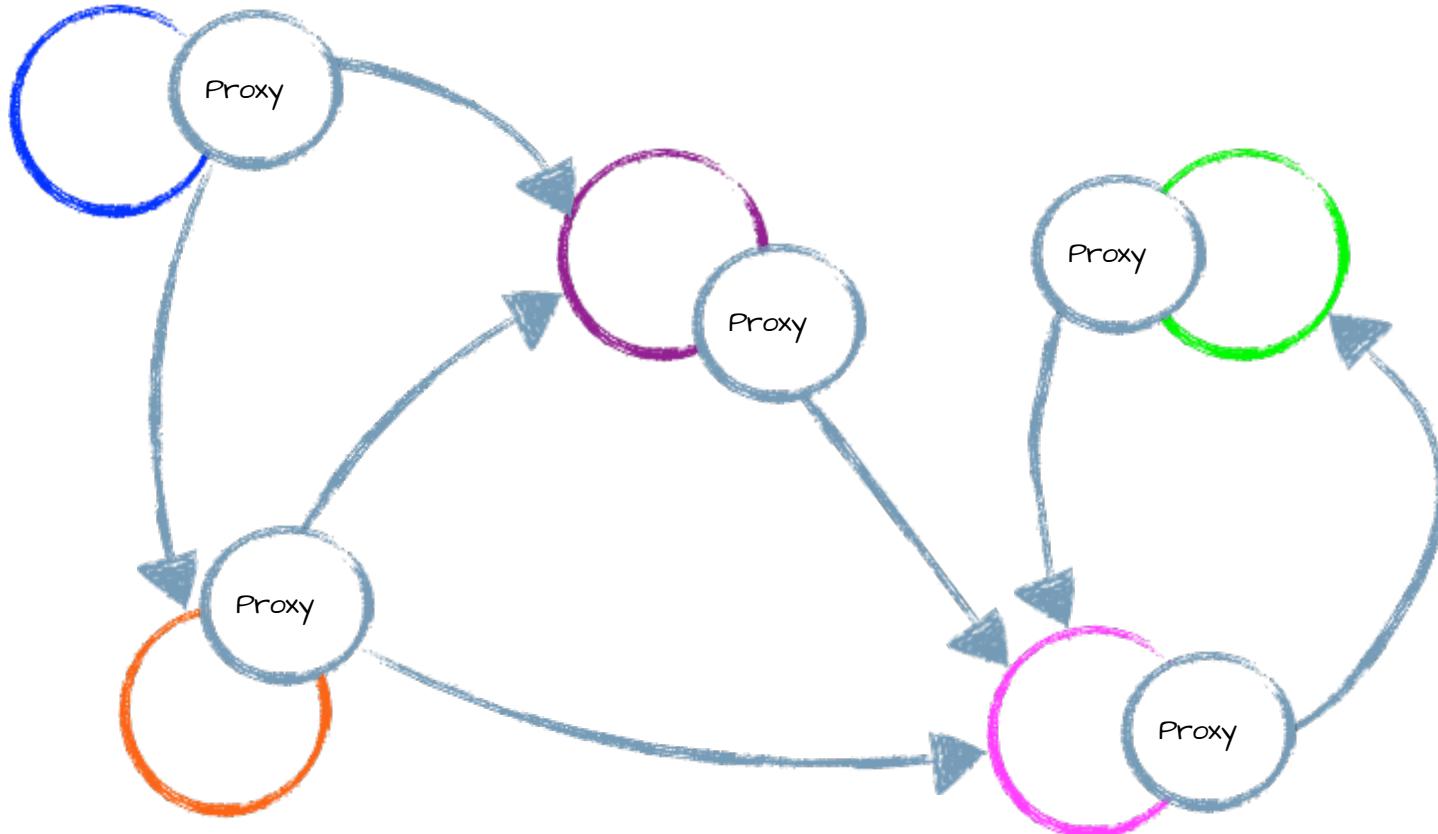
East-West Traffic

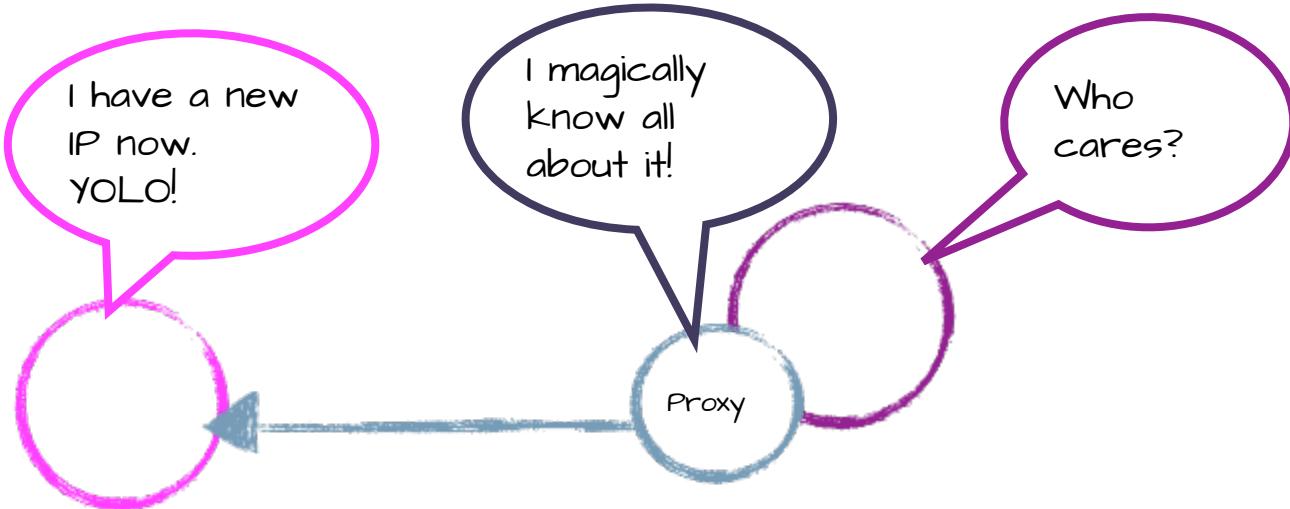


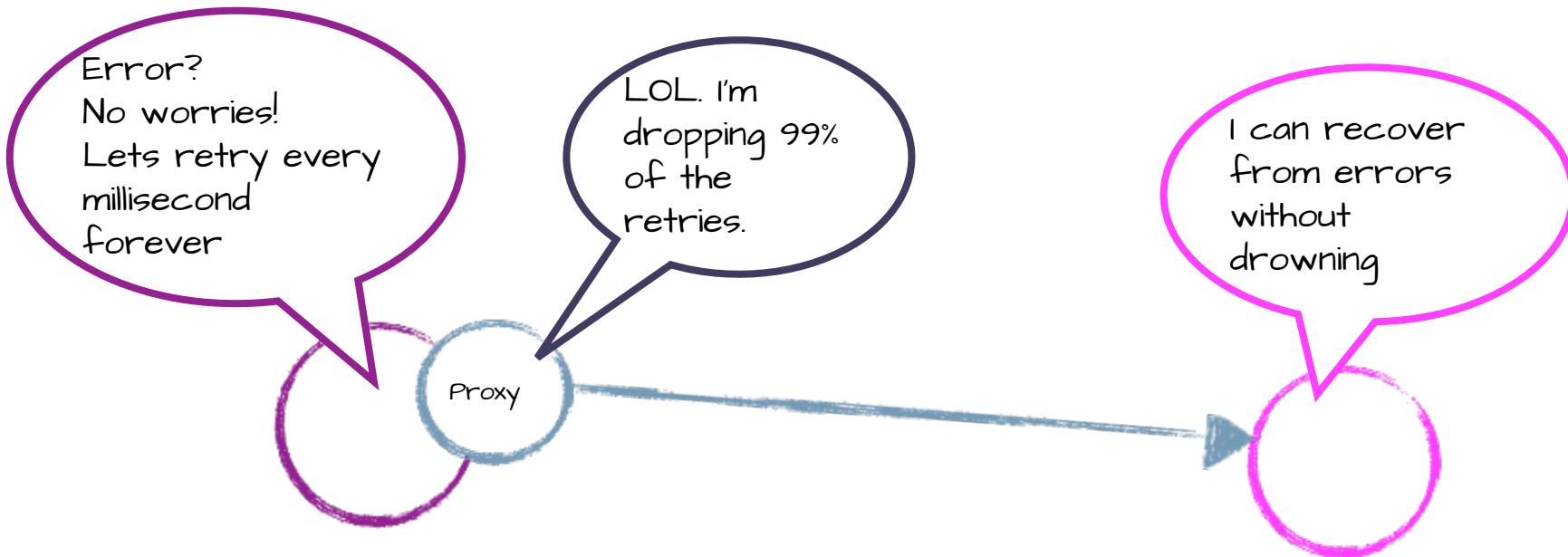
Side-car



Proxy as sidecar:

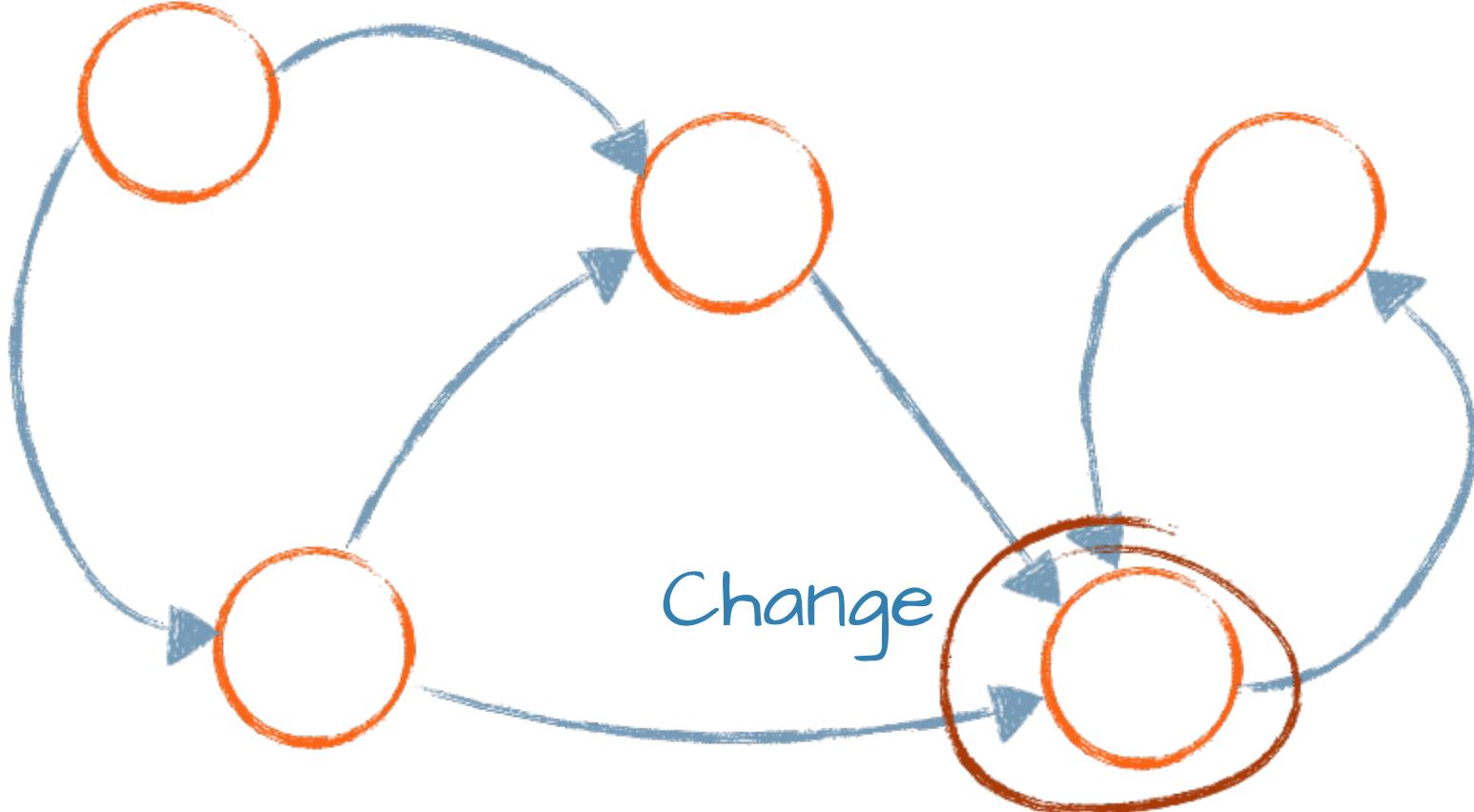




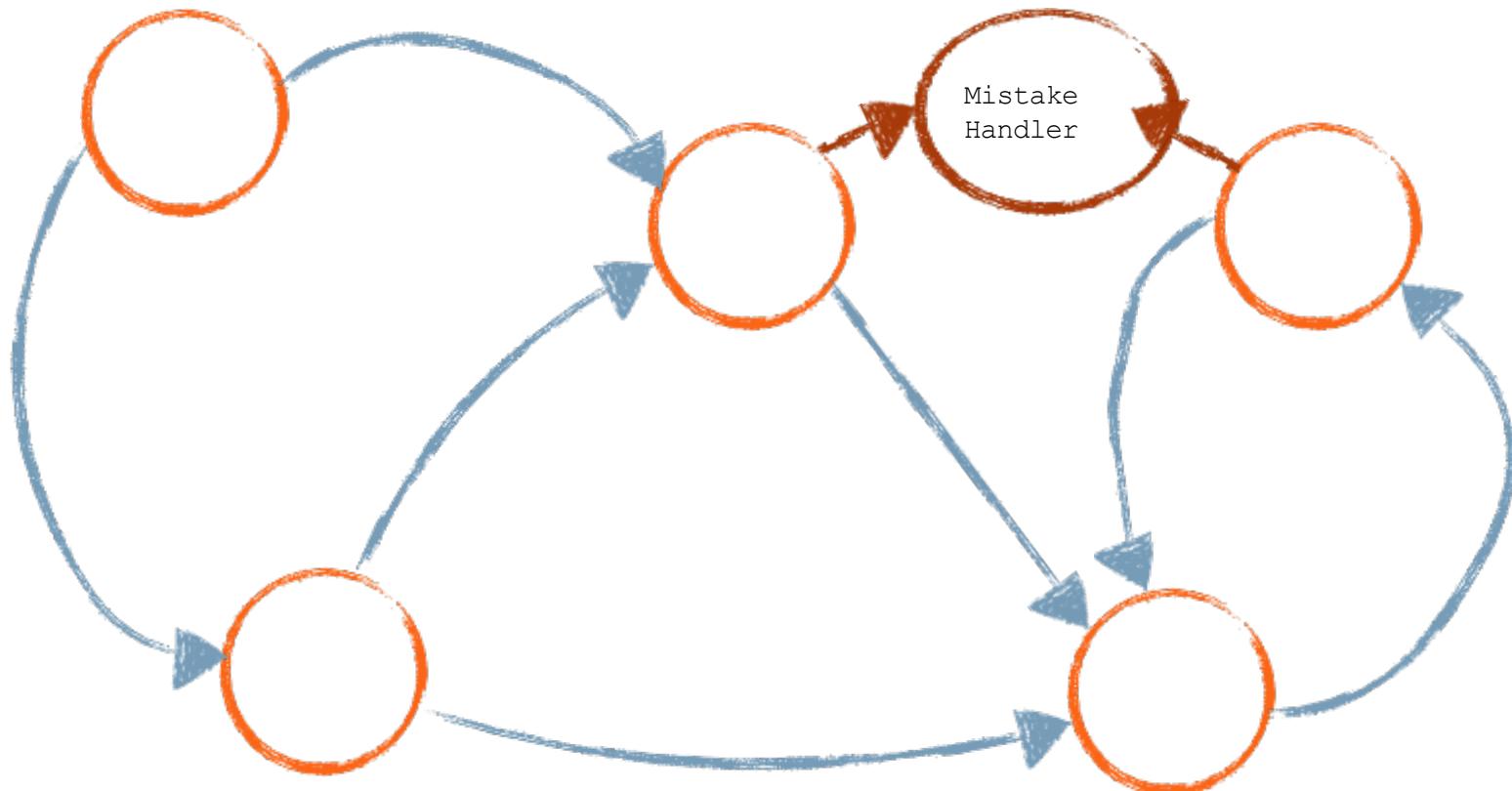


Event Driven

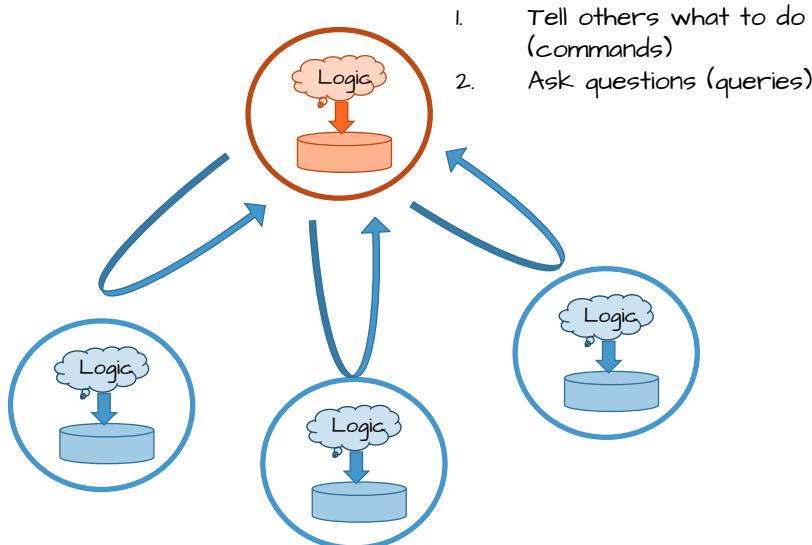
Making Changes is Risky



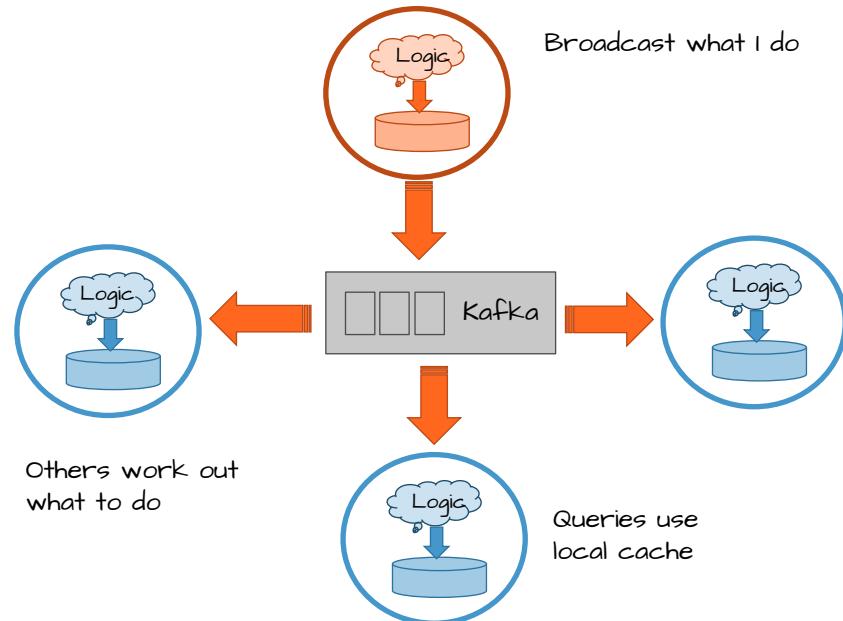
Adding Services Requires Explicit Calls



Request Driven



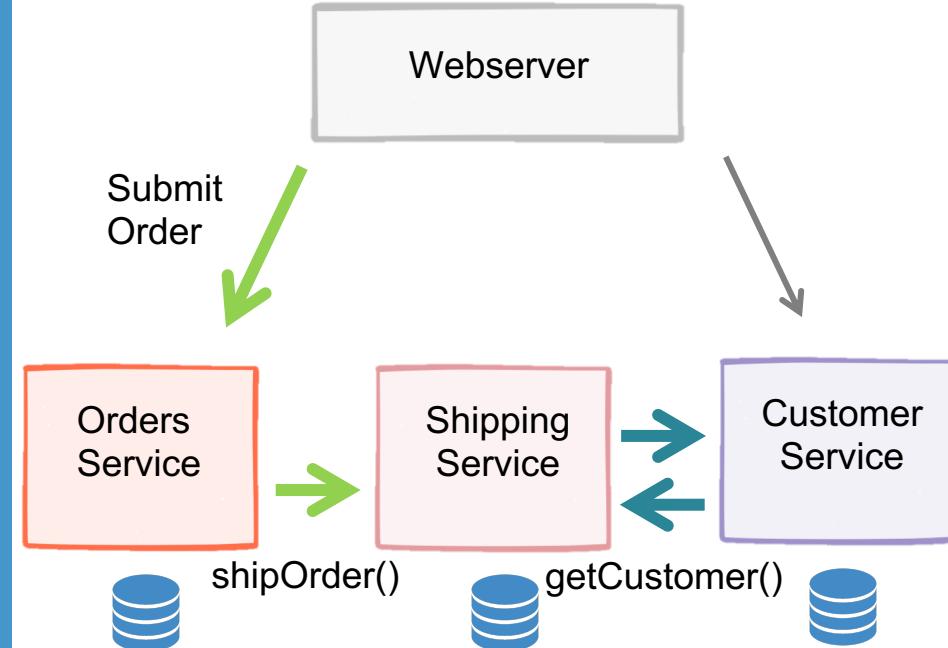
Event Driven



Events are both facts and triggers

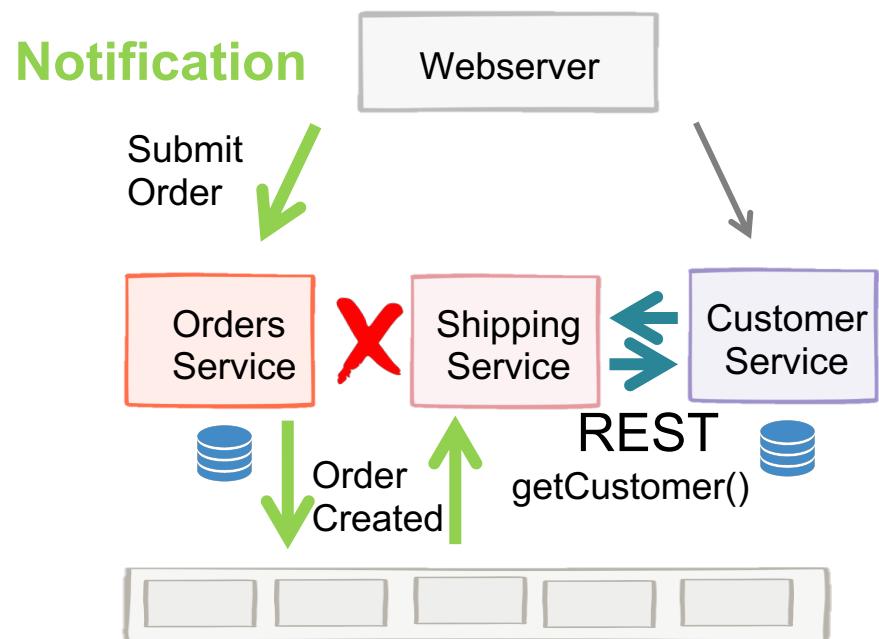
Buying an iPad (with REST)

- Orders Service calls Shipping Service to tell it to ship item.
- Shipping service looks up address to ship to (from Customer Service)



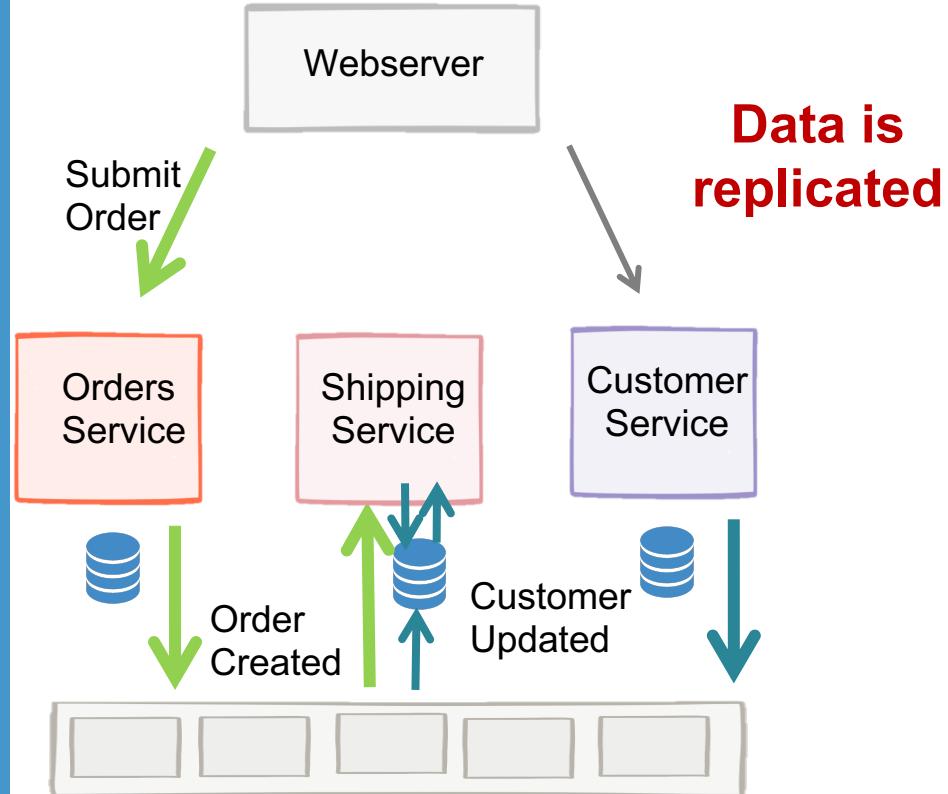
Using events for Notification

- Orders Service no longer knows about the Shipping service (or any other service). Events are fire and forget.

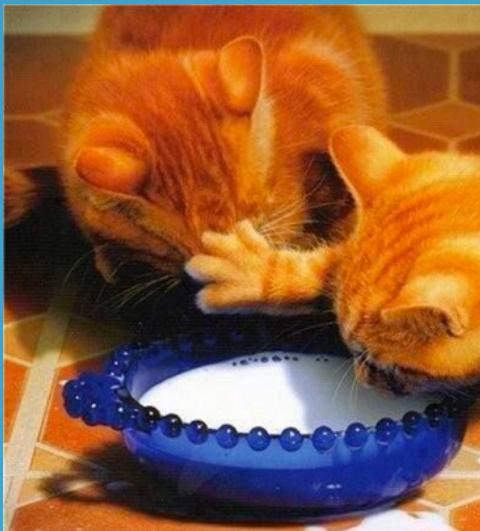


Using events to share facts

- Call to Customer service is gone.
- Instead data is replicated, as events, into the shipping service, where it is queried locally. .

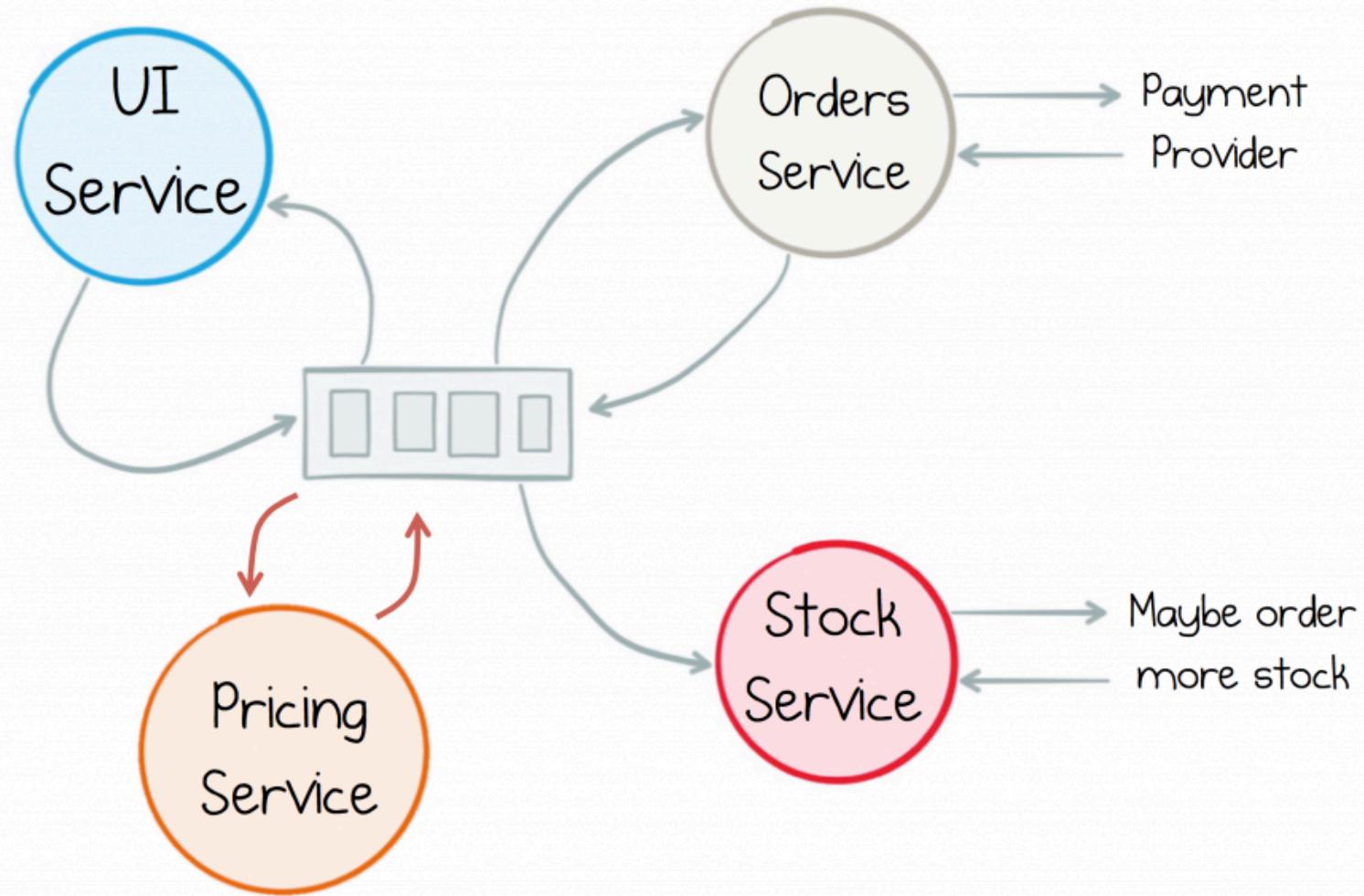


DB for Each Microservice?



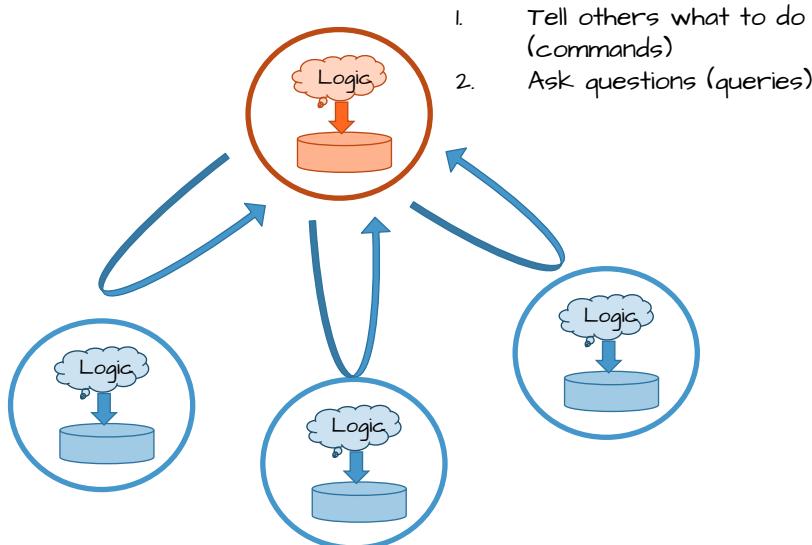
- - **It is safe:**
They are all derived from same stream of events
- - Reduced dependencies
- - **Custom projection** just the data each service needs.
- - Low latency

**Event Driven Microservices are
Stateful**

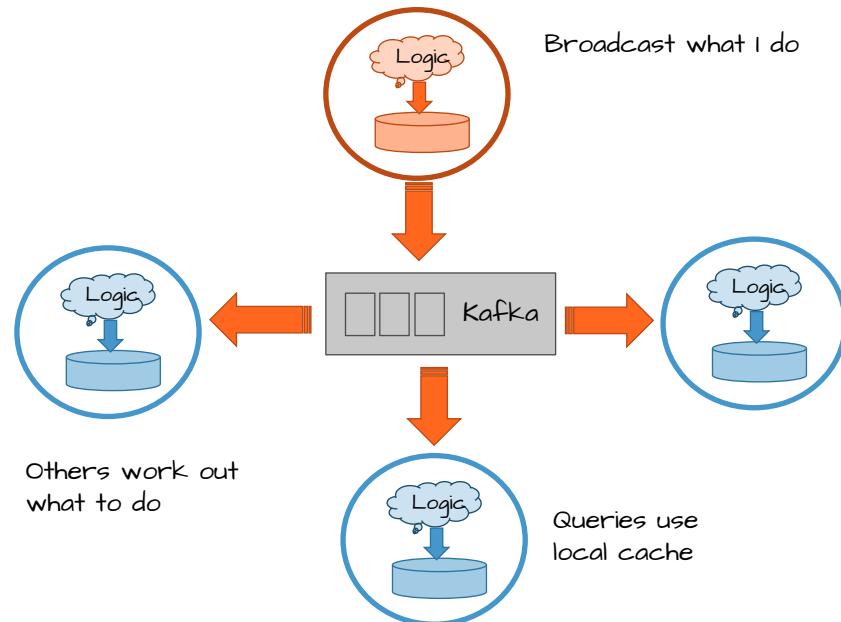


Schema

Request Driven



Event Driven

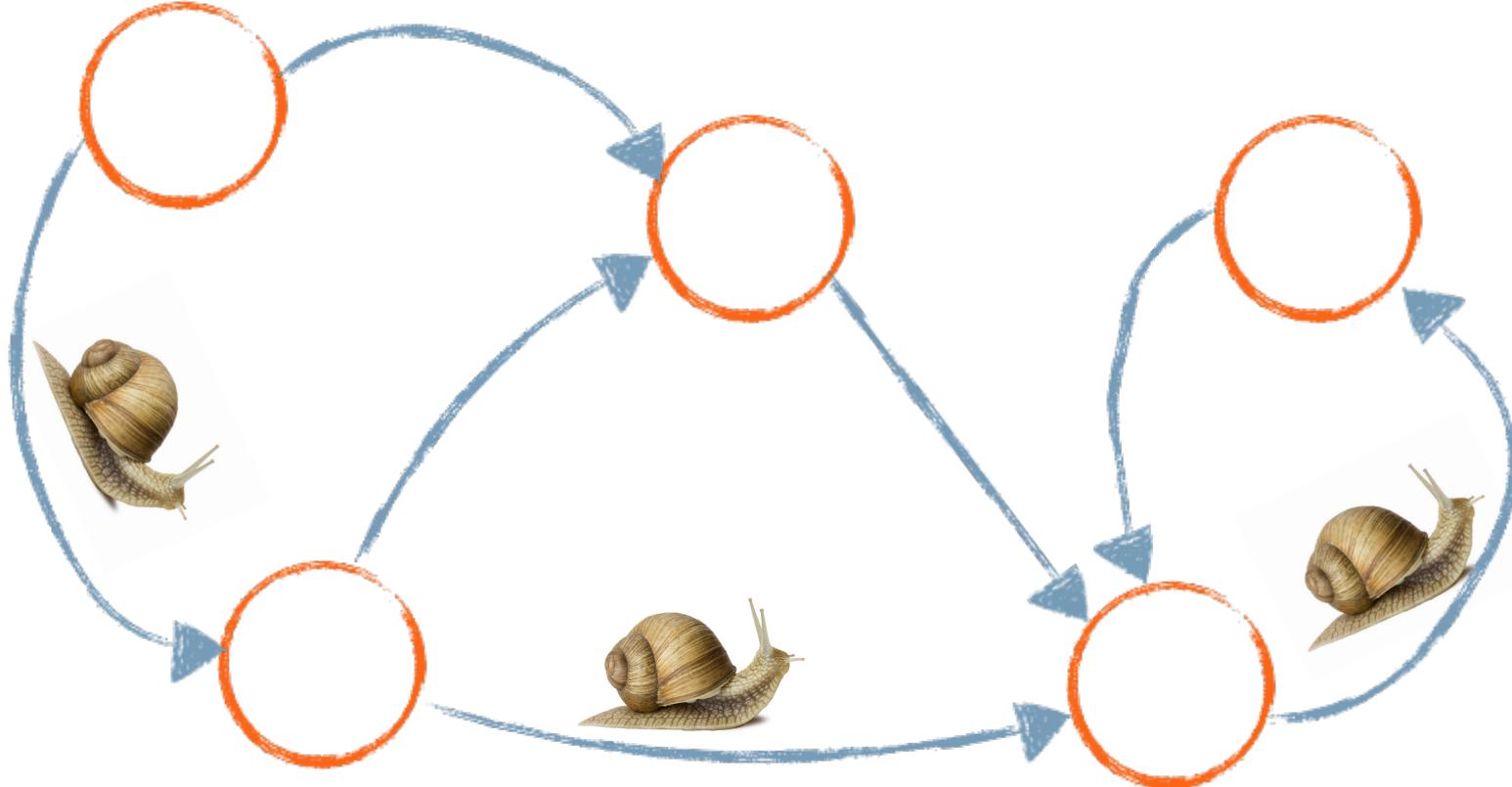


The medium is not the message.

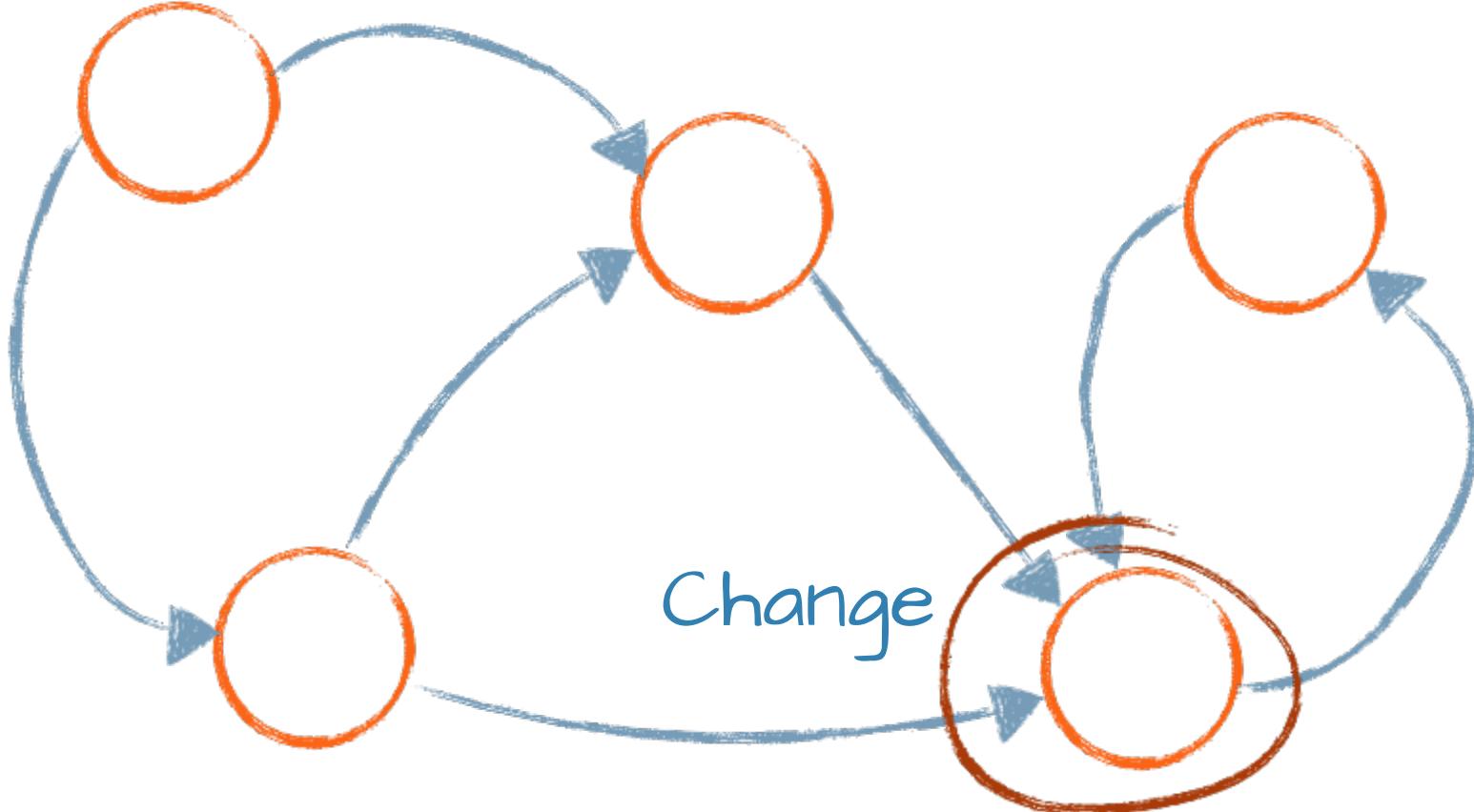
This is a message

```
{  
    sessionId: 676fc8983gu563,  
    timestamp: 1413215458,  
    viewType: "propertyView",  
    propertyId: 7879,  
    loyaltyId: 6764532  
    origin: "promotion",  
    ..... lots of metadata....  
}
```

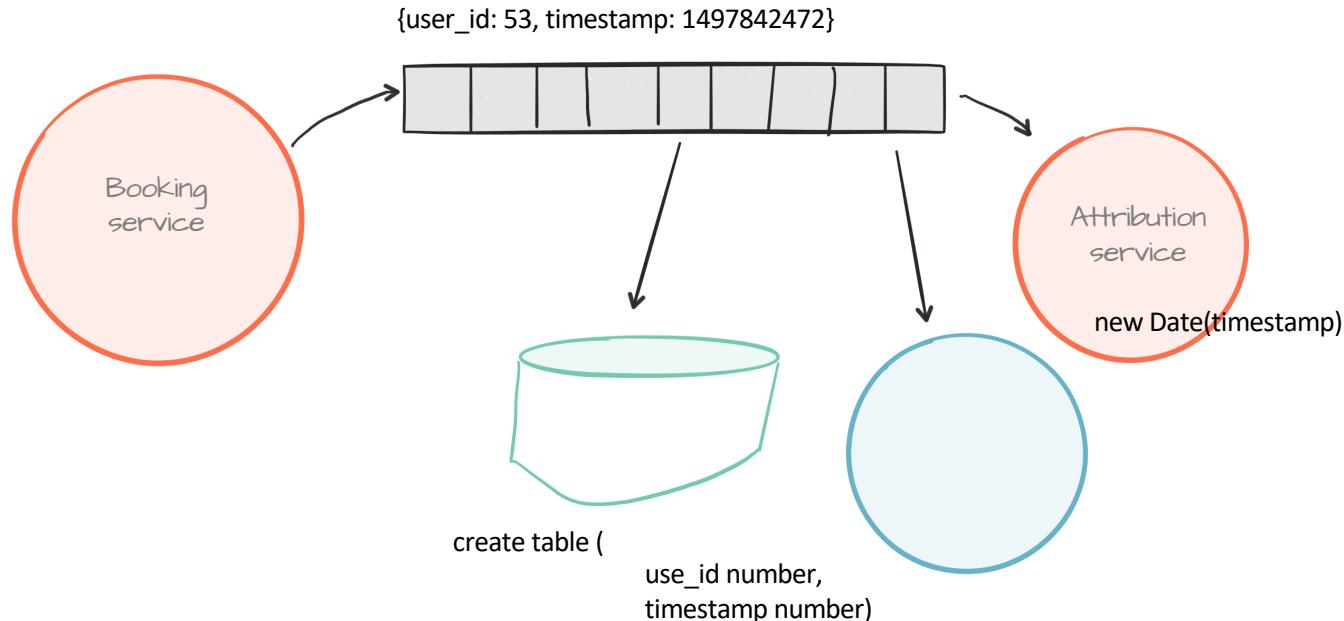
REST = HTTP + JSON = SLOW

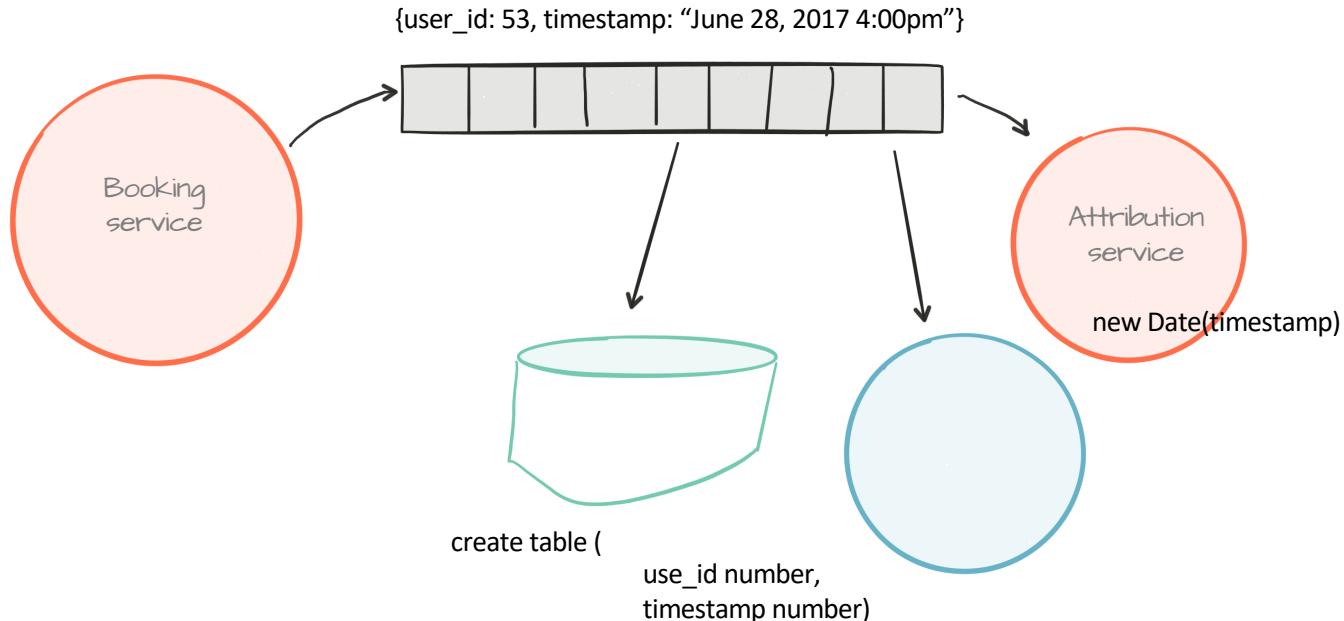


Making Changes is Risky

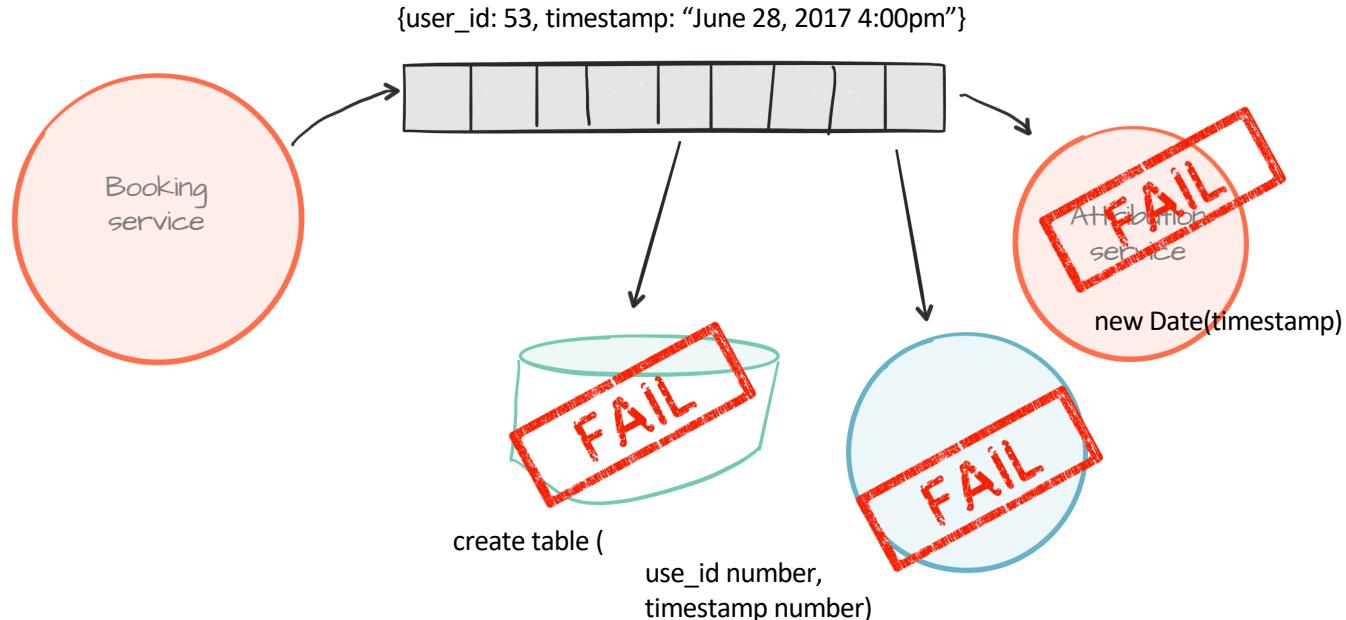


There are lots of dependencies

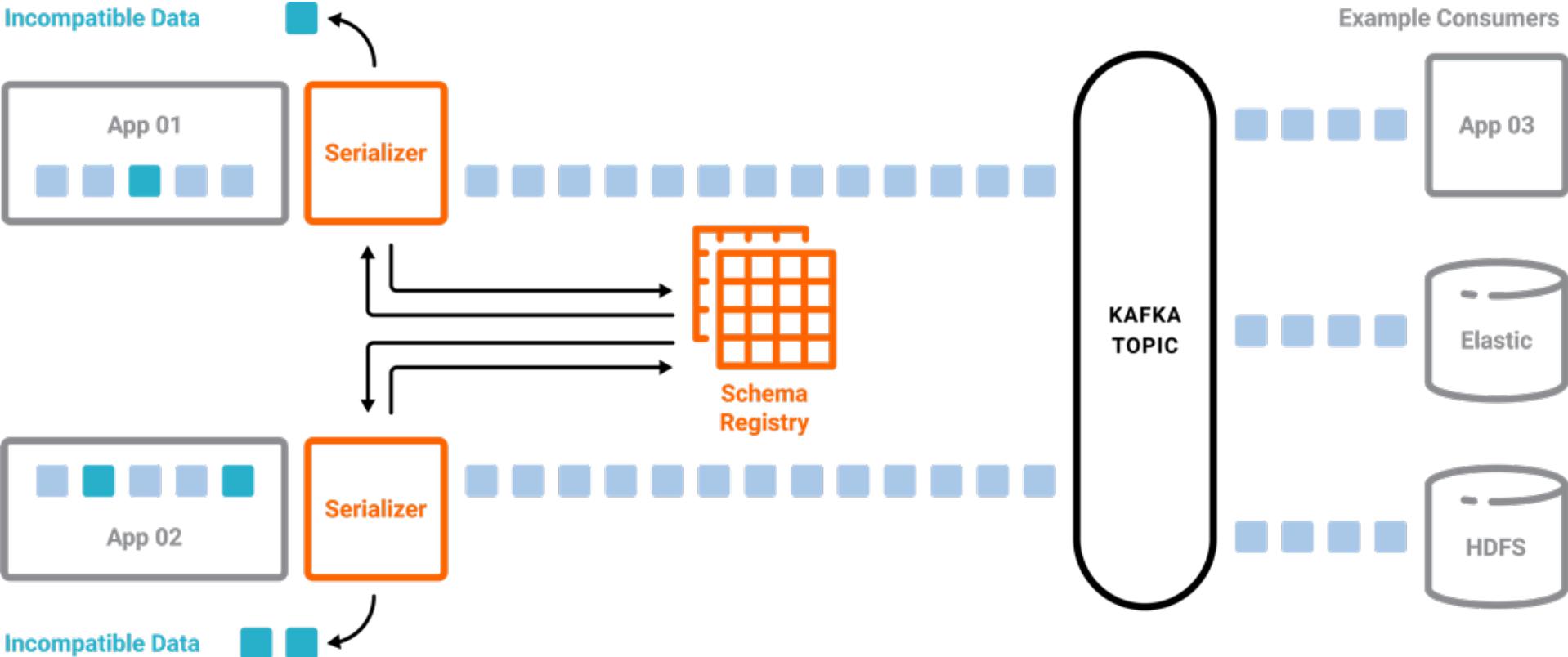




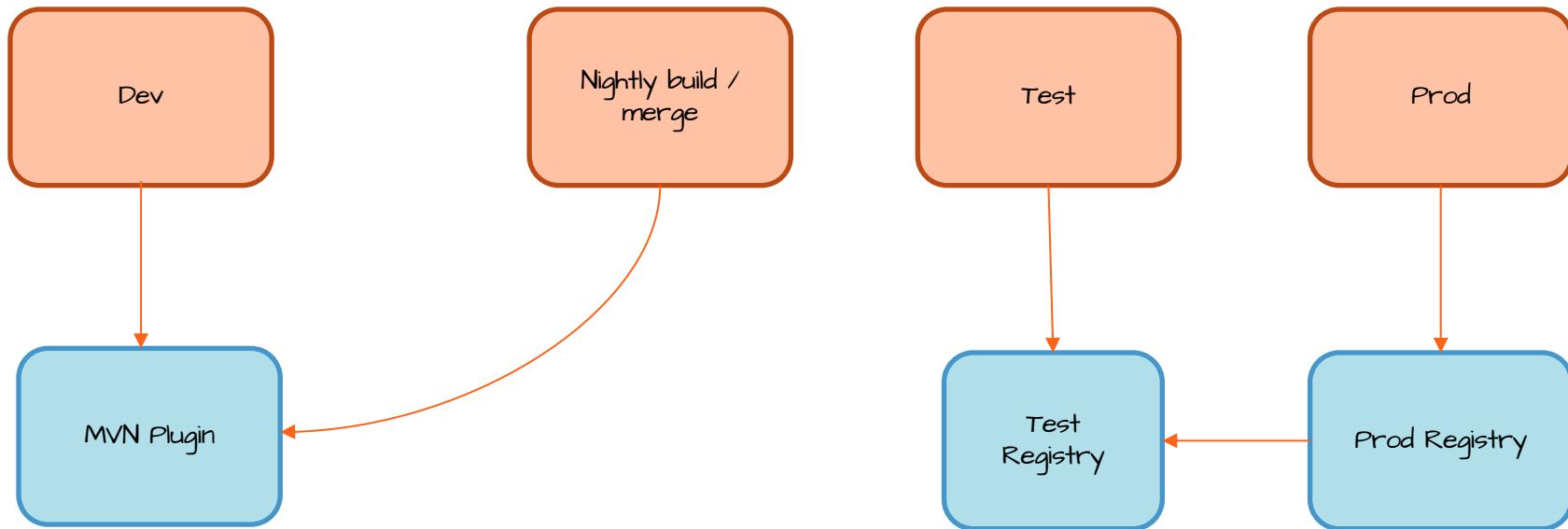
Moving fast and breaking things



**APIs between services are Contracts
In Event Driven World – Event Schemas ARE the API**

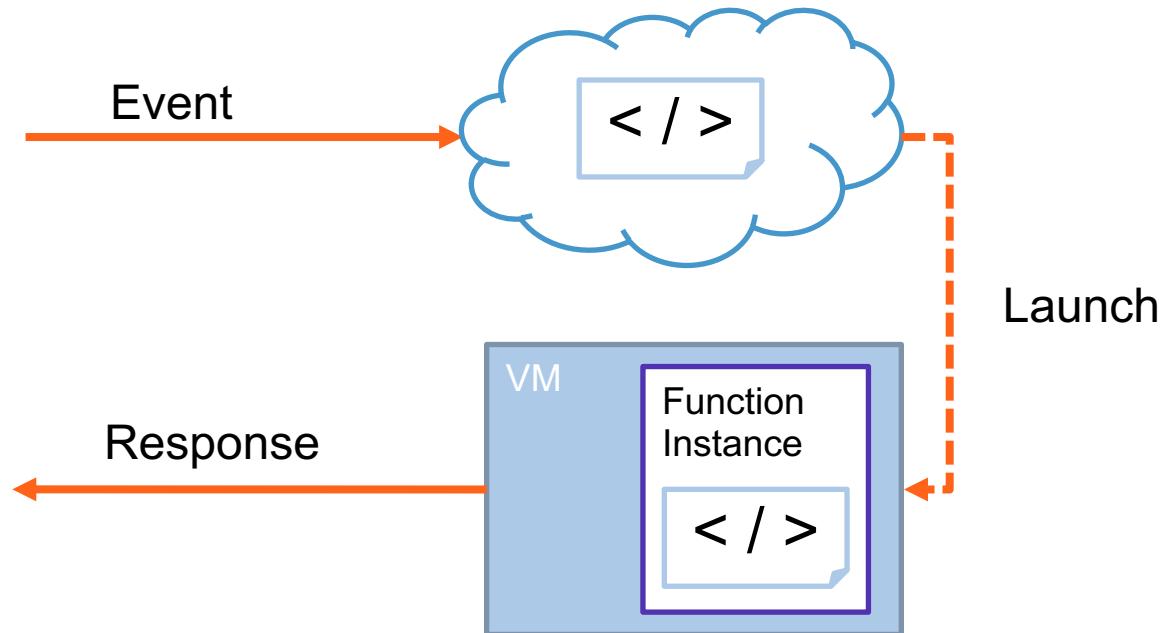


So the flow is...

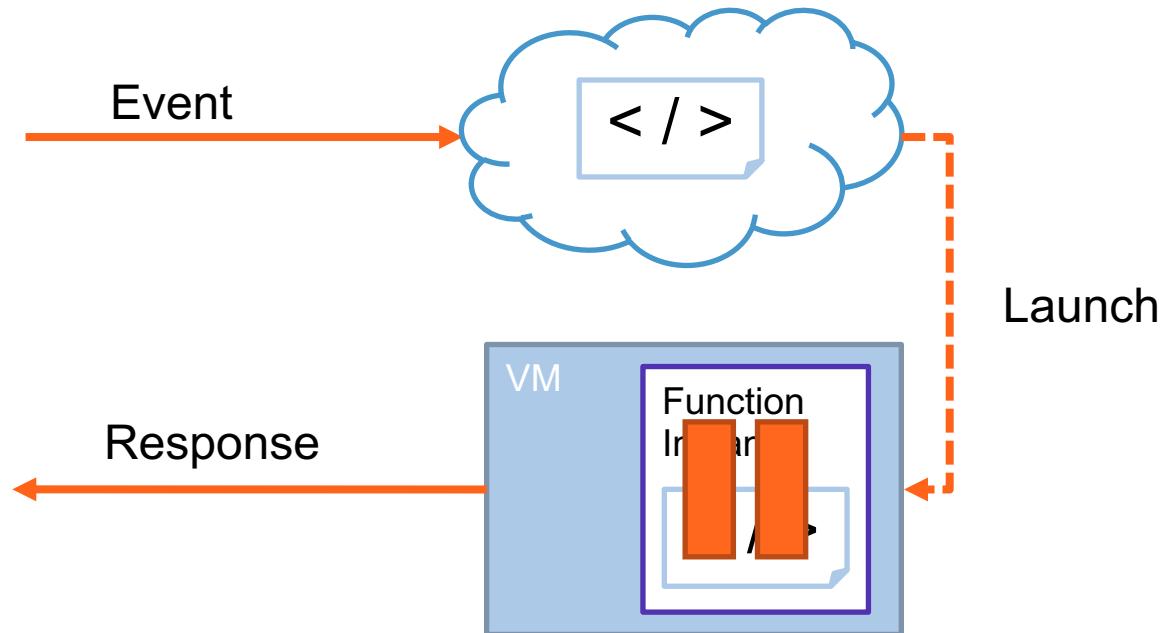


Serverless

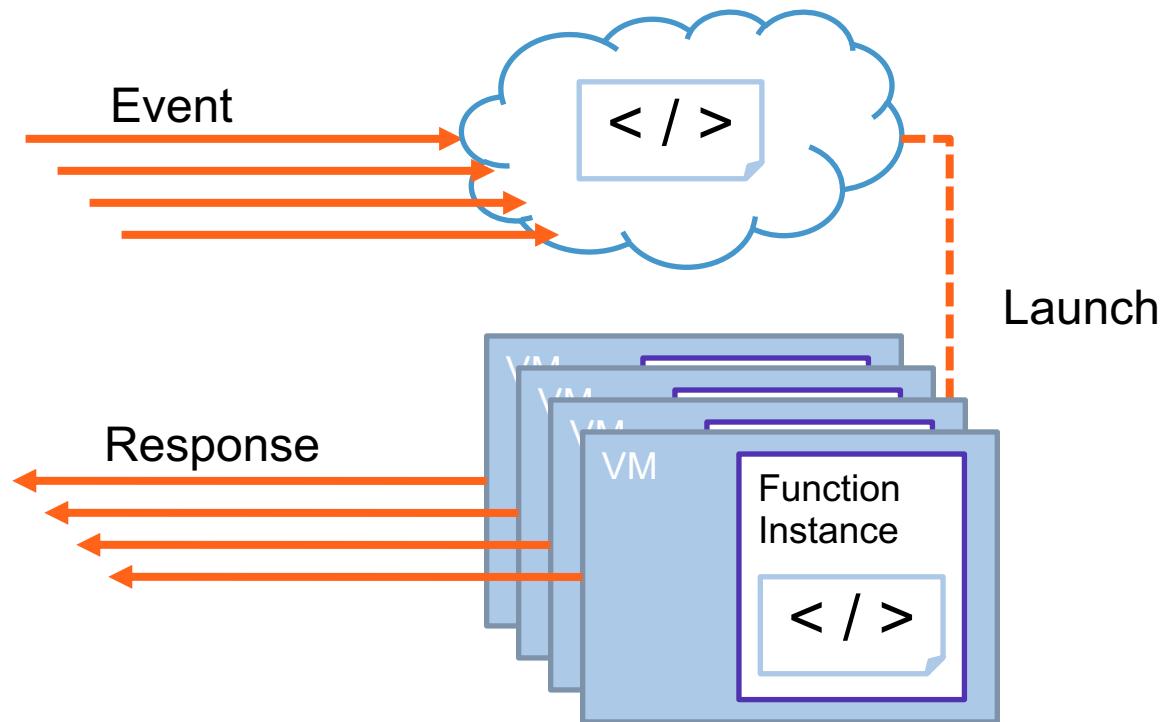
Function as a Service



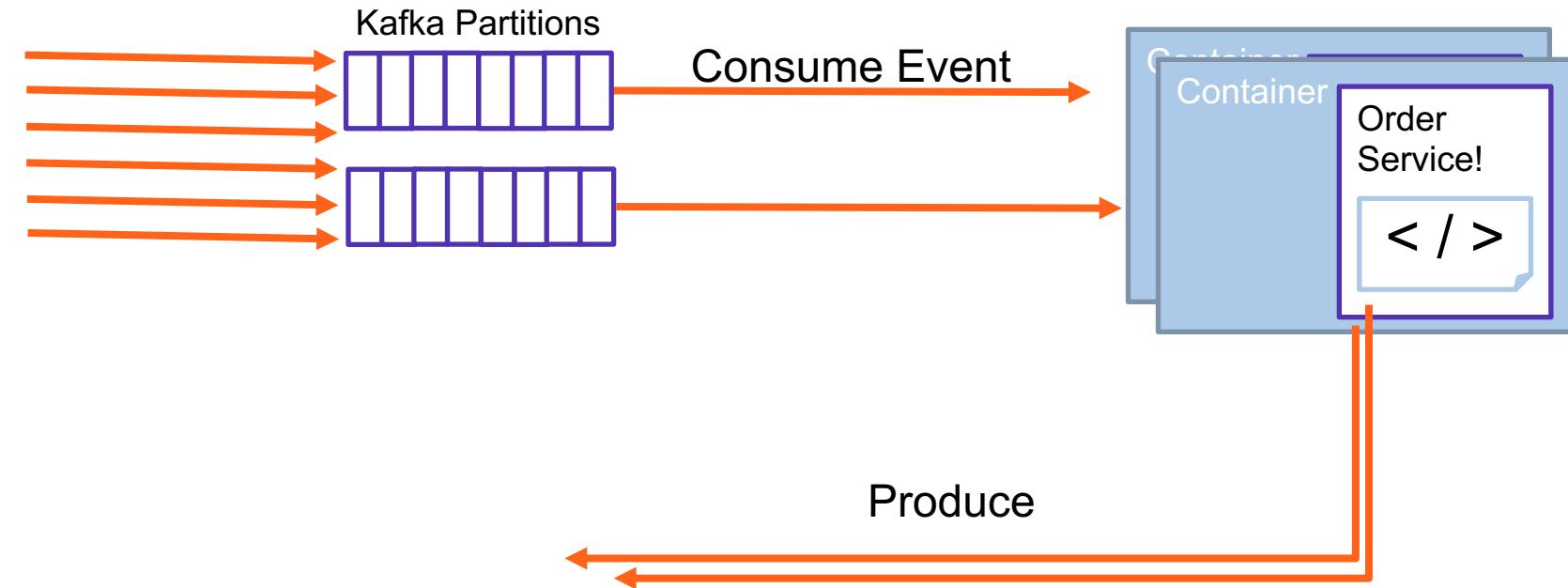
When nothing happens



At scale



Wait, this is super familiar



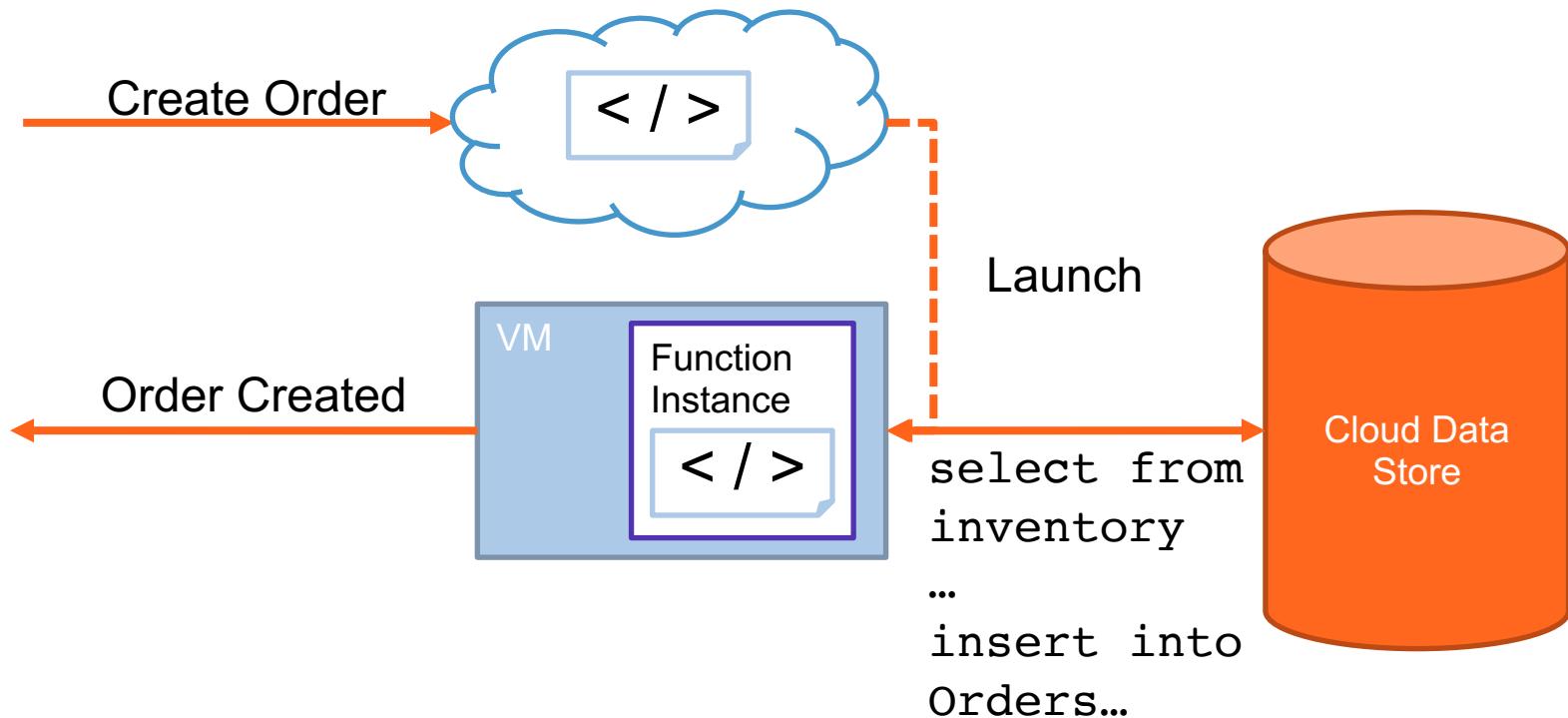
Up Next: Stateful Serverless

State is required

- Dynamic Rules
- Event enrichment
- Joining multiple events
- Aggregation

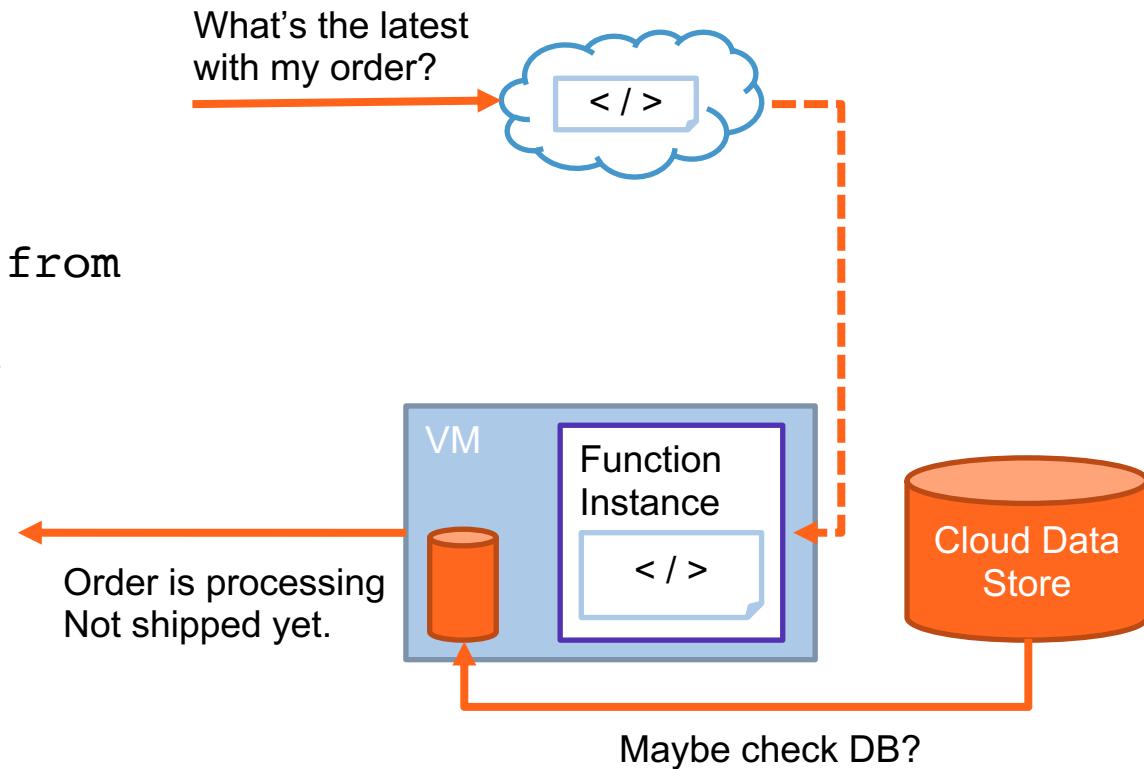


How You Probably Do State

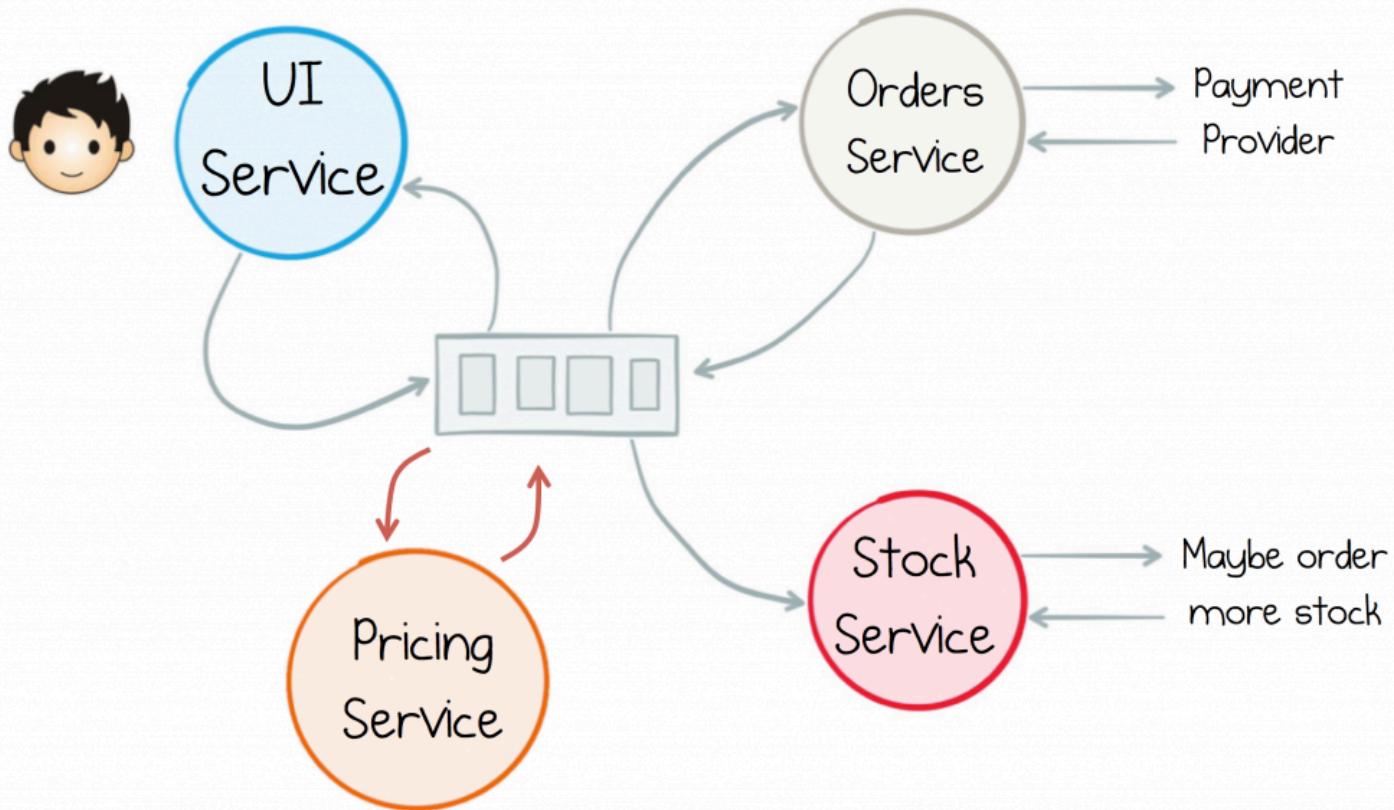


We can do a bit better

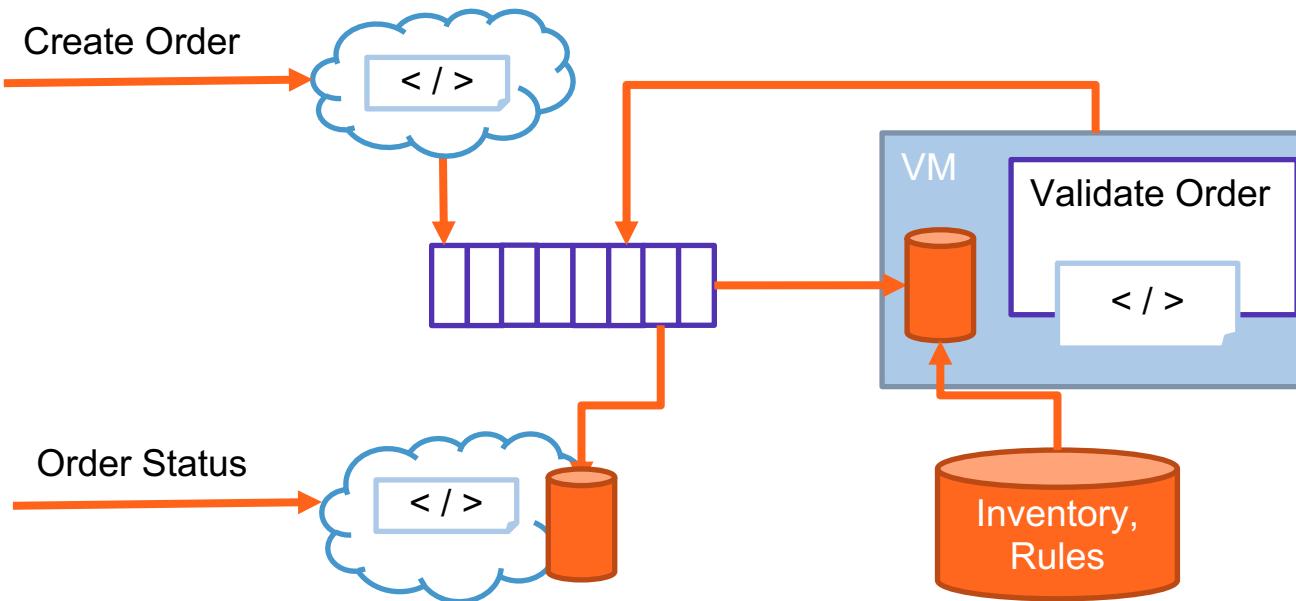
```
Select order_id,  
customer_name, product,  
quantity, price, state from  
orders  
where state != "CLOSED"
```



But I really want this back:



Stateful Serverless



What's Still missing?

- - Durable functions everywhere
- - Triggers and data from data stores to functions
- - Unified view of current state



Resources

- <http://christophermeiklejohn.com/serverless/2019/05/25/stateful-serverless-bank-example.html>
- <https://www.infoq.com/articles/service-mesh-promise-peril/>
- <https://blog.getambassador.io/api-gateway-vs-service-mesh-104c01fa4784>
- <https://www.nginx.com/blog/building-microservices-using-an-api-gateway/>
- <https://wecode.wepay.com/posts/migrating-apis-from-rest-to-grpc-at-wepay>
- <https://content.pivotal.io/slides/microservices-events-and-breaking-the-data-monolith-with-kafka>
- <https://www.slideshare.net/ConfluentInc/event-sourcing-stream-processing-and-serverless-ben-stopford-confluent-kafka-summit-sf-2019>