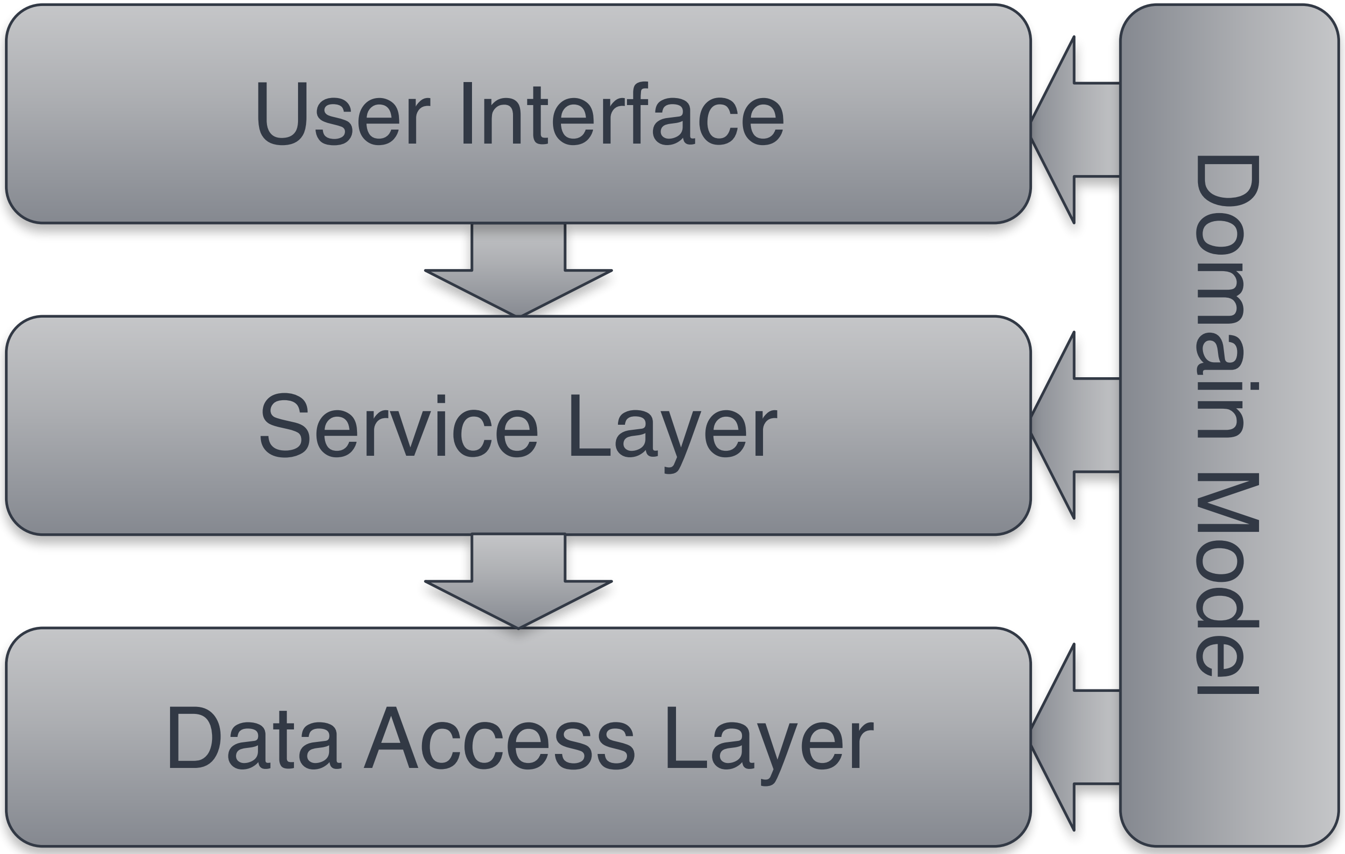
Event-Driven Microservices

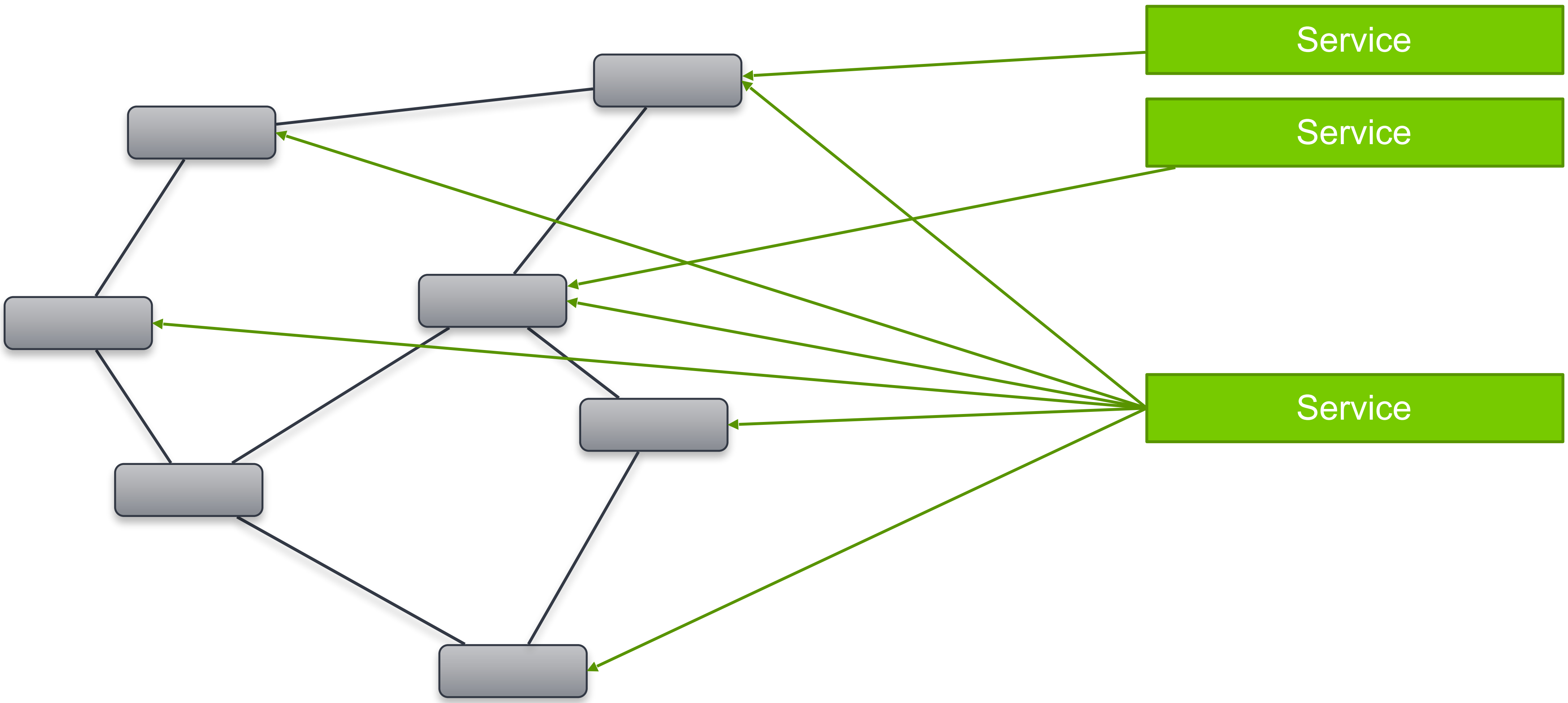
Not (just) about events

Allard Buijze
Founder & CTO, AxonIQ

✉allard@axoniq.io
@allardbz

Layered architecture




Service
Service
Service

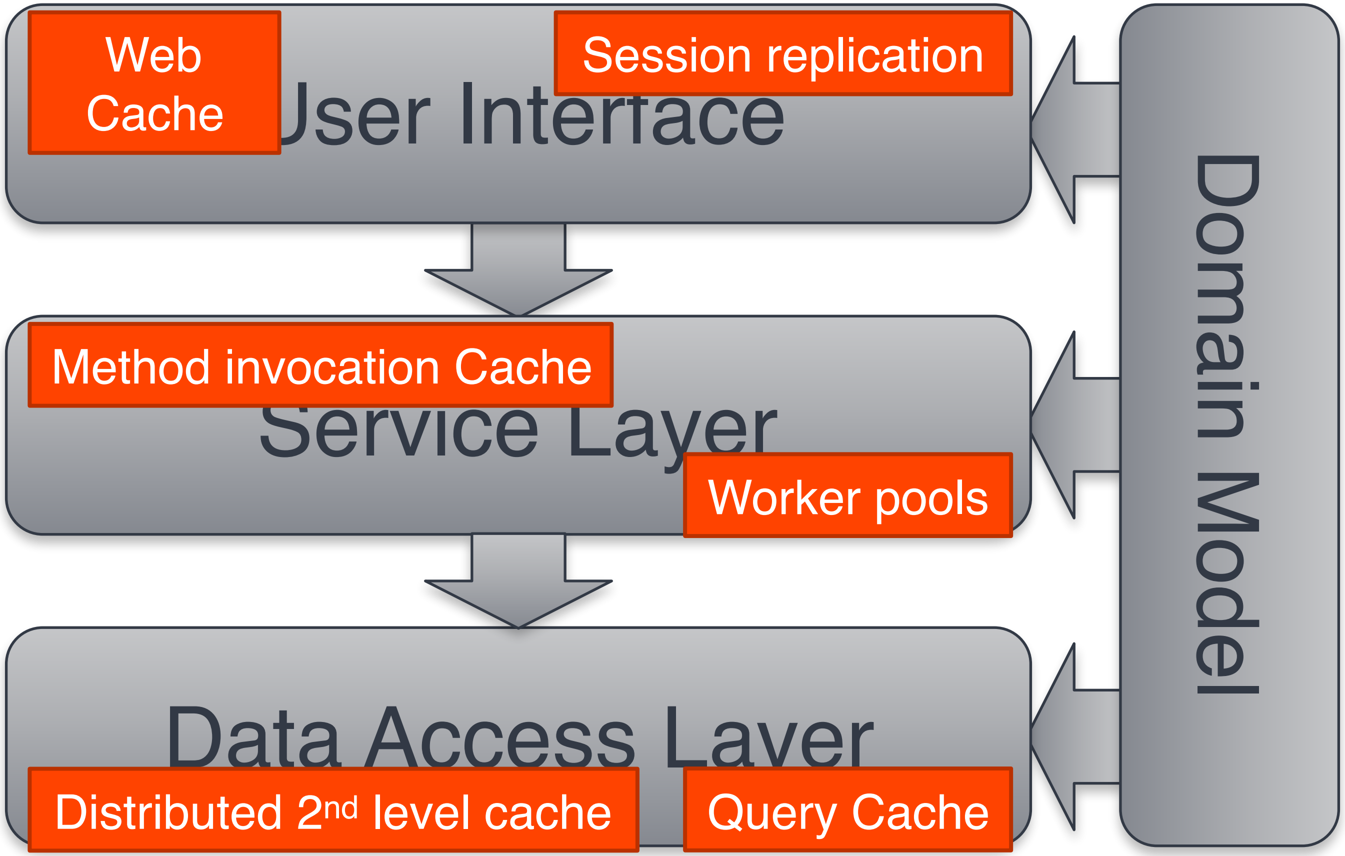
‘Normal’ SQL QUERY

```sql
CREATE ALGORITHM = UNDEFINED DEFINER = `dbat`@`%` SQL SECURITY DEFINERVIEW `BUSINESS_PROCESS_VIEW_DISABLED` AS SELECT `tpo`.`NAME` AS `TRADING_PARTNER_NAME`, `bt`.`NAME` AS `TRADING_PARTNER_BOARDING_TYPE`, `rto`.`NAME` AS `TRADING_PARTNER_ROOT_NAME`,
`rteid`.`VALUE` AS `TRADING_PARTNER_ROOT_EID`, `tpot`.`ORG_SUB_TYPE_NAME` AS `TRADING_PARTNER_CUSTOMER_TYPE`, `tpo`.`STATUS` AS `TRADING_PARTNER_STATUS`, `tpot`.`NAME` AS `TRADING_PARTNER_TYPE`, `teid`.`VALUE` AS `TRADING_PARTNER_EID`, `tduns`.`VALUE` AS
`TRADING_PARTNER_DUNS`, `tgln`.`VALUE` AS `TRADING_PARTNER_GLN`, `bp`.`BUSINESS_PROCESS` AS `BUSINESS_PROCESS`, (CASE CONCAT((`myo`.`ORGANIZATION_ID` = `rel`.`PARTY1_ID`), (`myo`.`ORGANIZATION_ID` = `rel`.`PARTY2_ID`), `bp`.`PARTY1_DIRECTION`) WHEN '10TO'
THEN 'TO' WHEN '10FROM' THEN 'FROM' WHEN '11TO' THEN 'TO' WHEN '01TO' THEN 'FROM' WHEN '01FROM' THEN 'TO' END) AS `DOCUMENT_DIRECTION`, `bp`.`STATUS` AS `STATUS`, `idt`.`DOC_TYPE` AS `INBOUND_TYPE`, `idt`.`DOC_SUB_TYPE` AS `INBOUND_SUB_TYPE`,
`idt`.`DOC_VERSION` AS `INBOUND_VERSION`, `iodt`.`DOC_DIRECTION` AS `INBOUND_DIRECTION`, `odt`.`DOC_TYPE` AS `OUTBOUND_TYPE`, `odt`.`DOC_SUB_TYPE` AS `OUTBOUND_SUB_TYPE`, `odt`.`DOC_VERSION` AS `OUTBOUND_VERSION`, `oodt`.`DOC_DIRECTION` AS
`OUTBOUND_DIRECTION`, `bp`.`GO_LIVE_DT` AS `GO_LIVE_DATE`, `bp`.`BUSINESS_PROCESS_ID` AS `ID`, `myo`.`ORGANIZATION_ID` AS `ORG_ID`, `myo`.`NAME` AS `ORG_NAME`, `meid`.`VALUE` AS `ORG_EID`, `ro`.`ORGANIZATION_ID` AS `ROOT_ORG_ID`, `ro`.`NAME` AS
`ROOT_ORG_NAME`, `reid`.`VALUE` AS `ROOT_ORG_EID`, `idt`.`DOC_TYPE_ID` AS `INBOUND_DOC_TYPE_ID`, `odt`.`DOC_TYPE_ID` AS `OUTBOUND_DOC_TYPE_ID`, `rel`.`RELATIONSHIP_ID` AS `RELATIONSHIP_ID`, `oop`.`ISA05` AS `OB_ENV_SNDR_QUAL`, `oop`.`ISA06` AS
`OB_ENV_SNDR_ID`, `oop`.`ISA07` AS `OB_ENV_RCVR_QUAL`, `oop`.`ISA08` AS `OB_ENV_RCVR_ID`, `oop`.`GS02` AS `OB_IN_SNDR_ID`, `oop`.`GS03` AS `OB_IN_RCVR_ID`, `iop`.`ISA05` AS `IB_ENV_SNDR_QUAL`, `iop`.`ISA06` AS `IB_ENV_SNDR_ID`, `iop`.`ISA07` AS
`IB_ENV_RCVR_QUAL`, `iop`.`ISA08` AS `IB_ENV_RCVR_ID`, `iop`.`GS02` AS `IB_IN_SNDR_ID`, `iop`.`GS03` AS `IB_IN_RCVR_ID`, `bpop`.`IB_OVR_ENV_RCVR_QUAL` AS `IB_OVR_ENV_RCVR_QUAL`, `bpop`.`IB_OVR_ENV_RCVR_ID` AS `IB_OVR_ENV_RCVR_ID`,
`bpop`.`IB_OVR_ENV_SNDR_QUAL` AS `IB_OVR_ENV_SNDR_QUAL`, `bpop`.`IB_OVR_ENV_SNDR_ID` AS `IB_OVR_ENV_SNDR_ID`, `bpop`.`IB_OVR_INNER_ENV_RCVR_ID` AS `IB_OVR_INNER_RCVR_ID`, `bpop`.`IB_OVR_INNER_ENV_SNDR_ID` AS `IB_OVR_INNER_SNDR_ID`,
`bpop`.`OB_OVR_ENV_RCVR_QUAL` AS `OB_OVR_ENV_RCVR_QUAL`, `bpop`.`OB_OVR_ENV_RCVR_ID` AS `OB_OVR_ENV_RCVR_ID`, `bpop`.`OB_OVR_ENV_SNDR_QUAL` AS `OB_OVR_ENV_SNDR_QUAL`, `bpop`.`OB_OVR_ENV_SNDR_ID` AS `OB_OVR_ENV_SNDR_ID`, `bpop`.`OB_OVR_INNER_ENV_RCVR_ID`
AS `OB_OVR_INNER_RCVR_ID`, `bpop`.`OB_OVR_INNER_ENV            AS `IB_TRX_ENV_RCVR_            `IB_TRX_ENV_SNDR_QUAL` AS `IB_TRX_ENV_SNDR_QUAL`,
`bpop`.`IB_TRX_ENV_SNDR_ID` AS `IB_TRX_ENV_SNDR_            `.`IB_TRX_INNER_EN            `OB_TRX_ENV_RCVR_QUAL`,
`bpop`.`OB_TRX_ENV_RCVR_ID` AS `OB_TRX_ENV_RCVR            B_TRX_ENV_SNDR_ID            _INNER_RCVR_ID`,
`bpop`.`OB_TRX_INNER_ENV_SNDR_ID` AS `OB_TRX_INN            `rel`.`XSBL_ID`            LATIONSHIP` `rel` ON ((`rel`.`RELATIONSHIP_ID`
= `bp`.`RELATIONSHIP_ID`))) LEFT JOIN `ORGANIZAT            .`ORGANIZATION_ID            _ID` = `ro`.`ORGANIZATION_ID`))) LEFT JOIN
`ORGANIZATION` `tpo` ON (((`tpo`.`ORGANIZATION_            )))) LEFT JOIN `O            NIZATION_ID`))) LEFT JOIN `BOARDING_TYPE` `bt`
ON ((`bta`.`BOARDING_TYPE_ID` = `bt`.`BOARDING_            `ORGANIZATION_ID`            = `rto`.`ORGANIZATION_ID`) AND
(`rteid`.`PRIMARY_FLAG` = 1) AND (`rteid`.`ORG_            TYPE_ID` FROM `DI            )))) LEFT JOIN `ORG_TYPE_ASSOC` `tpoa` ON
((`tpoa`.`ORGANIZATION_ID` = `tpo`.`ORGANIZATION            oa`.`ORG_TYPE_ID`            `myo`.`ORGANIZATION_ID`) AND
(`meid`.`PRIMARY_FLAG` = 1) AND (`meid`.`ORG_IDE            PE_ID` FROM `DIM_            )) LEFT JOIN `ORG_IDENTIFIER` `teid` ON
(((`teid`.`ORGANIZATION_ID` = `tpo`.`ORGANIZATIO            _ID` = (SELECT `DI            ER_TYPE` WHERE (`DIM_ORG_IDENTIFIER_TYPE`.`NAME`
= 'EID')))))) LEFT JOIN `ORG_IDENTIFIER` `tduns`            ns`.`PRIMARY_FLAG` =            ER_TYPE`.`ORG_IDENTIFIER_TYPE_ID` FROM
`DIM_ORG_IDENTIFIER_TYPE` WHERE (`DIM_ORG_IDENTIFIER_TYPE`.`NAME` = 'DUNS')))))) LEFT JOIN `ORG_IDENTIFIER` `tgln` ON (((`tgln`.`ORGANIZATION_ID` = `tpo`.`ORGANIZATION_ID`) AND (`tgln`.`PRIMARY_FLAG` = 1) AND (`tgln`.`ORG_IDENTIFIER_TYPE_ID` = (SELECT
`DIM_ORG_IDENTIFIER_TYPE`.`ORG_IDENTIFIER_TYPE_ID` FROM `DIM_ORG_IDENTIFIER_TYPE` WHERE (`DIM_ORG_IDENTIFIER_TYPE`.`NAME` = 'GLN')))))) LEFT JOIN `ORG_IDENTIFIER` `reid` ON (((`reid`.`ORGANIZATION_ID` = `ro`.`ORGANIZATION_ID`) AND (`reid`.`PRIMARY_FLAG` =
1) AND (`reid`.`ORG_IDENTIFIER_TYPE_ID` = (SELECT `DIM_ORG_IDENTIFIER_TYPE`.`ORG_IDENTIFIER_TYPE_ID` FROM `DIM_ORG_IDENTIFIER_TYPE` WHERE (`DIM_ORG_IDENTIFIER_TYPE`.`NAME` = 'EID')))))) LEFT JOIN `ORG_DOC_TYPE` `iodt` ON ((((`iodt`.`ORG_DOC_TYPE_ID` =
`bp`.`IB_DOC_TYPE_ID`) AND (((`myo`.`ORGANIZATION_ID` = `rel`.`PARTY1_ID`) AND (`bp`.`PARTY1_DIRECTION` = 'FROM')) OR ((`myo`.`ORGANIZATION_ID` = `rel`.`PARTY2_ID`) AND (`bp`.`PARTY1_DIRECTION` = 'TO')))) OR ((`iodt`.`ORG_DOC_TYPE_ID` =
`bp`.`OB_DOC_TYPE_ID`) AND (((`myo`.`ORGANIZATION_ID` = `rel`.`PARTY1_ID`) AND (`bp`.`PARTY1_DIRECTION` = 'TO')) OR ((`myo`.`ORGANIZATION_ID` = `rel`.`PARTY2_ID`) AND (`bp`.`PARTY1_DIRECTION` = 'FROM'))))))) LEFT JOIN `DIM_DOC_TYPE` `idt` ON
((`idt`.`DOC_TYPE_ID` = `iodt`.`DOC_TYPE_ID`))) LEFT JOIN `ORG_DOC_TYPE` `oodt` ON ((((`oodt`.`ORG_DOC_TYPE_ID` = `bp`.`IB_DOC_TYPE_ID`) AND (((`myo`.`ORGANIZATION_ID` = `rel`.`PARTY1_ID`) AND (`bp`.`PARTY1_DIRECTION` = 'TO')) OR ((`myo`.`ORGANIZATION_ID`
= `rel`.`PARTY2_ID`) AND (`bp`.`PARTY1_DIRECTION` = 'FROM')))) OR ((`oodt`.`ORG_DOC_TYPE_ID` = `bp`.`OB_DOC_TYPE_ID`) AND (((`myo`.`ORGANIZATION_ID` = `rel`.`PARTY1_ID`) AND (`bp`.`PARTY1_DIRECTION` = 'FROM')) OR ((`myo`.`ORGANIZATION_ID` =
`rel`.`PARTY2_ID`) AND (`bp`.`PARTY1_DIRECTION` = 'TO'))))))) LEFT JOIN `DIM_DOC_TYPE` `odt` ON ((`odt`.`DOC_TYPE_ID` = `oodt`.`DOC_TYPE_ID`))) LEFT JOIN `BUSINESS_PROCESS_OVERRIDE_PARAM` `bpop` ON ((`bpop`.`BUSINESS_PROCESS_OVERRIDE_PARAM_ID` =
`bp`.`BUSINESS_PROCESS_ID`))) LEFT JOIN `ORG_DOC_TYPE_PARAMETER_BUSINESS_PROCESS_VIEW` `iop` ON ((`iop`.`ID` = `bp`.`IB_DOC_TYPE_ID`))) LEFT JOIN `ORG_DOC_TYPE_PARAMETER_BUSINESS_PROCESS_VIEW` `oop` ON ((`oop`.`ID` = `bp`.`OB_DOC_TYPE_ID`))) WHERE
((((`myo`.`ORGANIZATION_ID` = `rel`.`PARTY1_ID`) AND (`tpo`.`ORGANIZATION_ID` = `rel`.`PARTY2_ID`)) OR ((`myo`.`ORGANIZATION_ID` = `rel`.`PARTY2_ID`) AND (`tpo`.`ORGANIZATION_ID` = `rel`.`PARTY1_ID`))) AND ISNULL(`bp`.`ORIGIN_BUSINESS_PROCESS_ID`))
```

22 JOINS

6 SUBQUERIES

Layered architecture

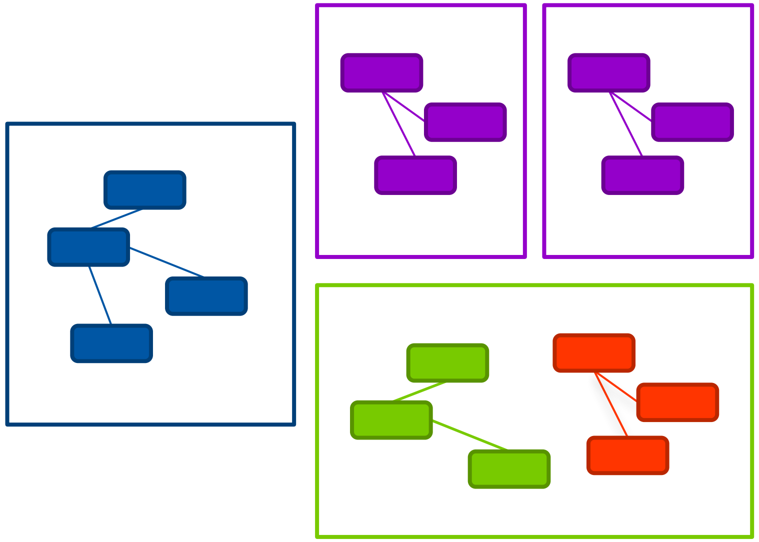


352 KTH
MODIFIEDPLANET.COM

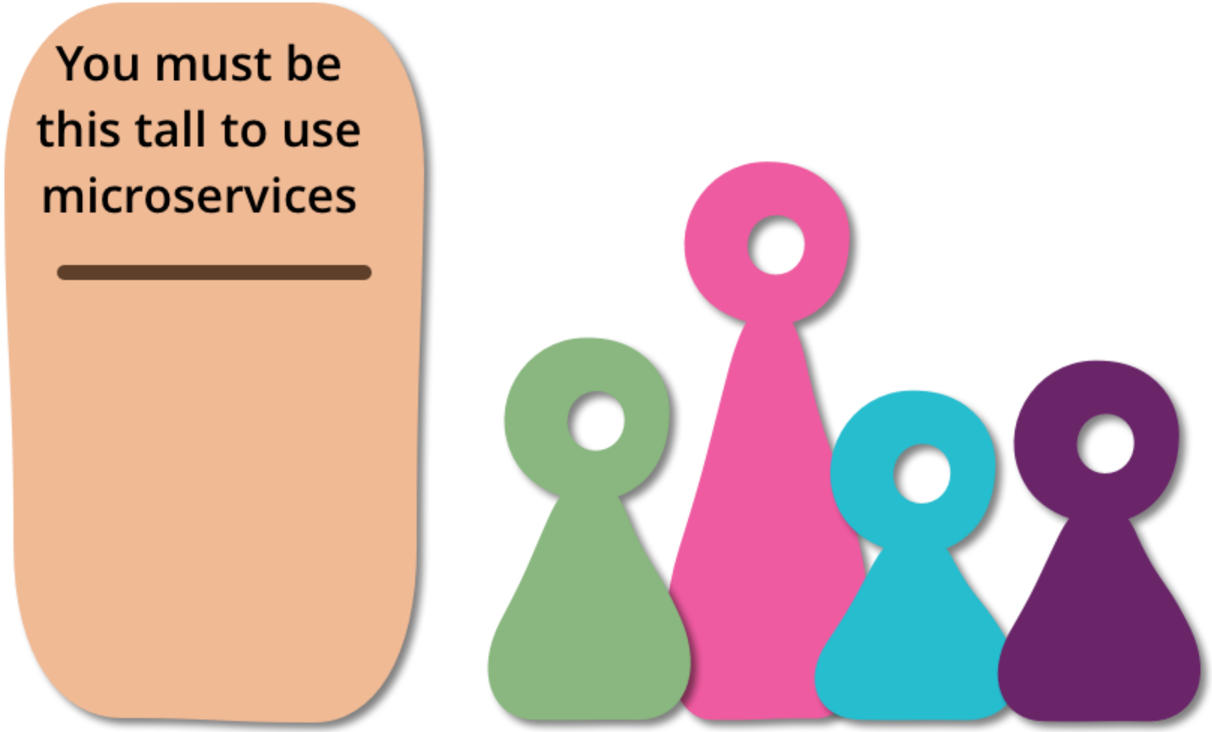
Source: http://www.sabisabi.com/images/DungBeetle-on-dung.JPG

Microservices systems

- Splitting up systems into smaller, simpler components
  - Agility
  - Scalability



Are you tall enough?



Source: martinfowler.com/bliki/MicroservicePrerequisites.html

Microservices vs Monoliths

Monoliths

Almost all the successful microservice stories have started with a monolith that got too big and was broken up

Microservices system

Almost all the cases where I've heard of a system that was built as a microservice system from scratch, it has ended up in serious trouble.

Martin Fowler

Source: http://martinfowler.com/bliki/MonolithFirst.html

“Noun Driven Design”

Noun? → Service!

“Noun Driven Design”

OrderService

“Noun Driven Design”

CustomerService

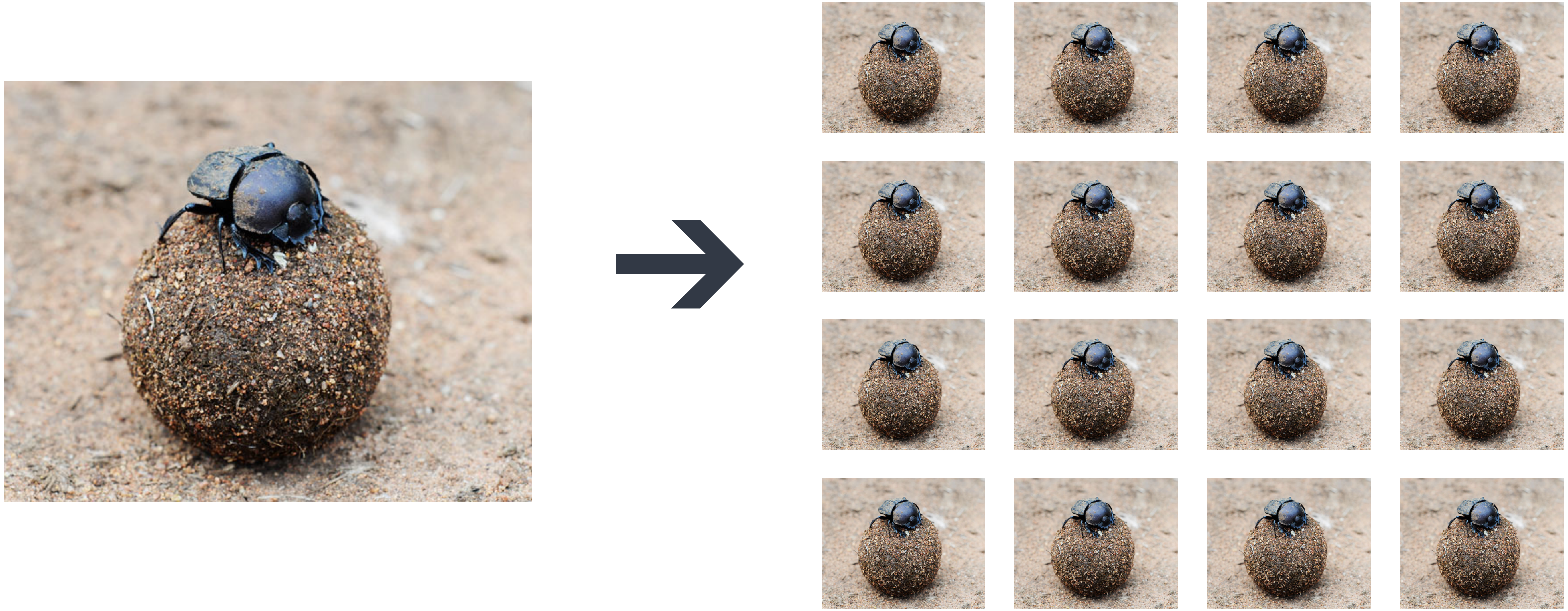
“Noun Driven Design”

ProductService

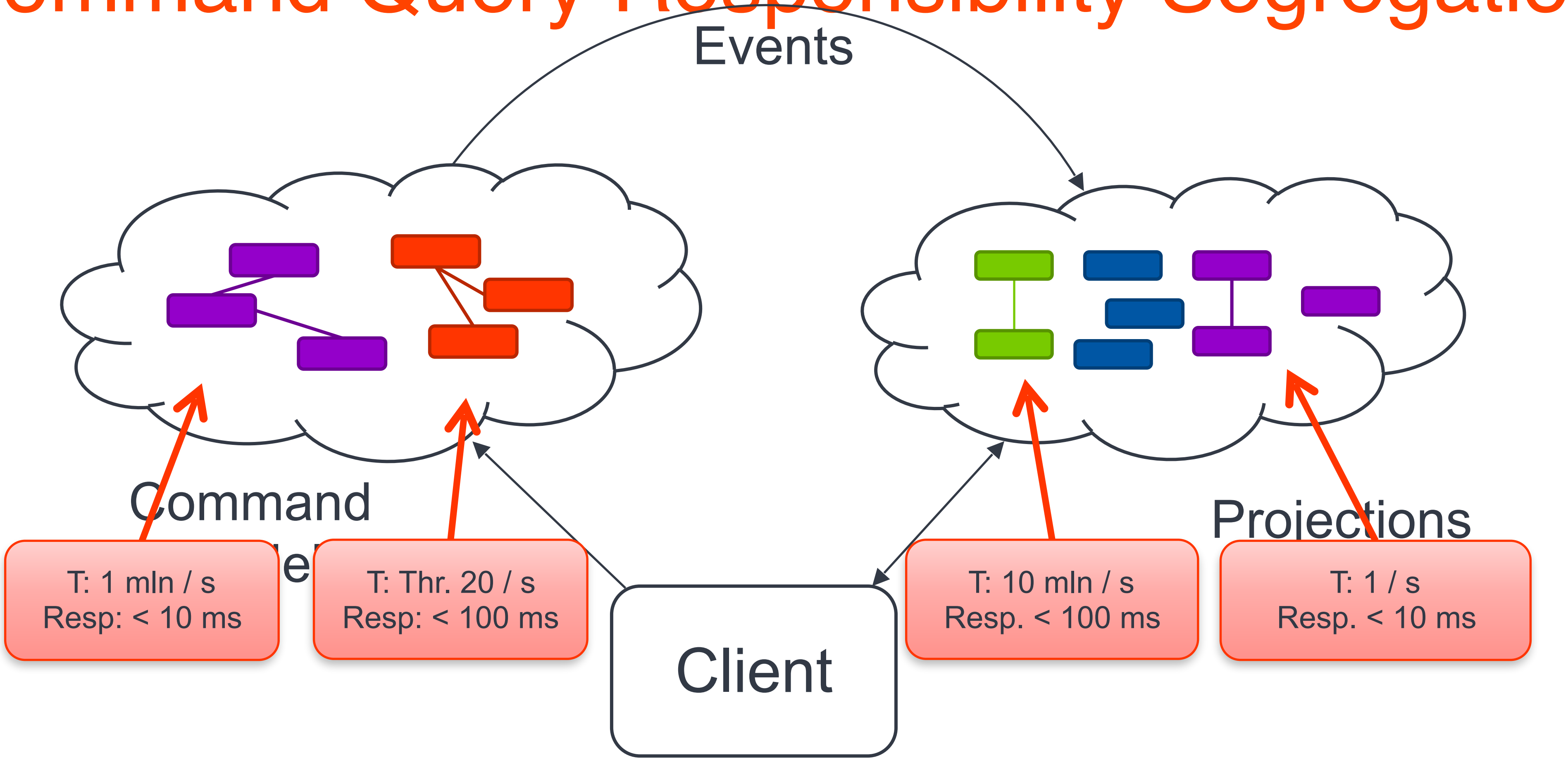
“Noun Driven Design”

InventoryService

“Entity Services”



Command Query Responsibility Segregation



Events

Command

e

Projections

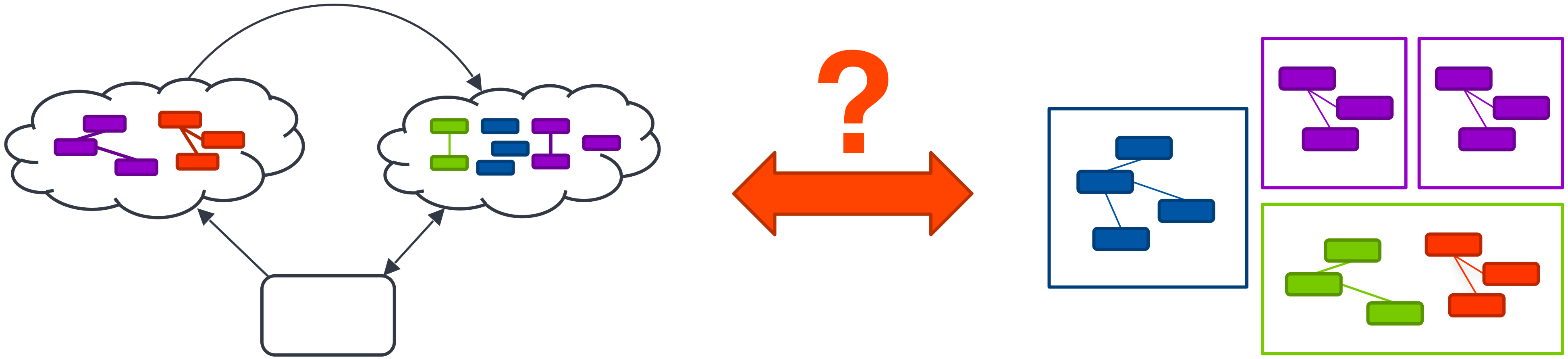
T: 1 mln / s
Resp: < 10 ms

T: Thr. 20 / s
Resp: < 100 ms

Client

T: 10 mln / s
Resp. < 100 ms

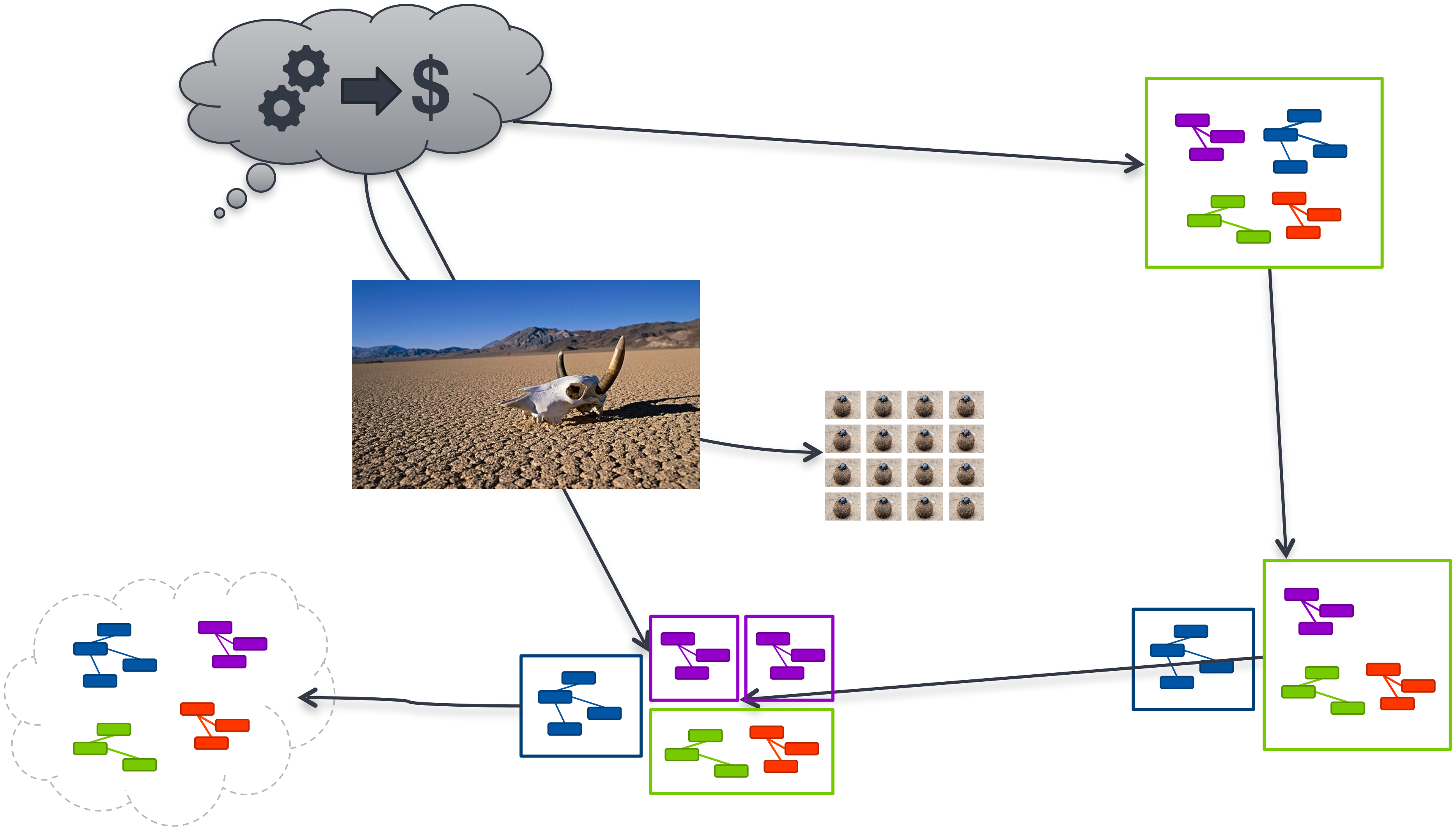
T: 1 / s
Resp. < 10 ms



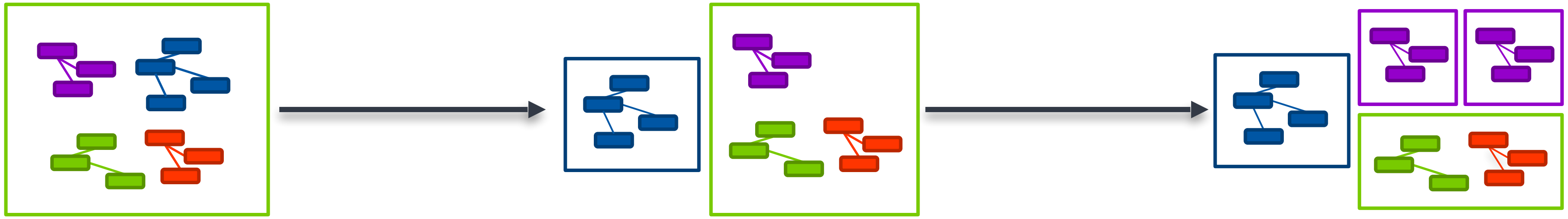
Monoliths

St Breock Downs Monolith - www.cornwalls.co.uk

≠



Location transparency



A component should neither be aware of nor make any assumptions about the location of components it interacts with.

Location transparency starts with good API design

(but doesn't end there)

‘Event-Driven’ Microservices

“Event” all the things!

‘Event-Driven’ Microservices

Maslow’s Hammer

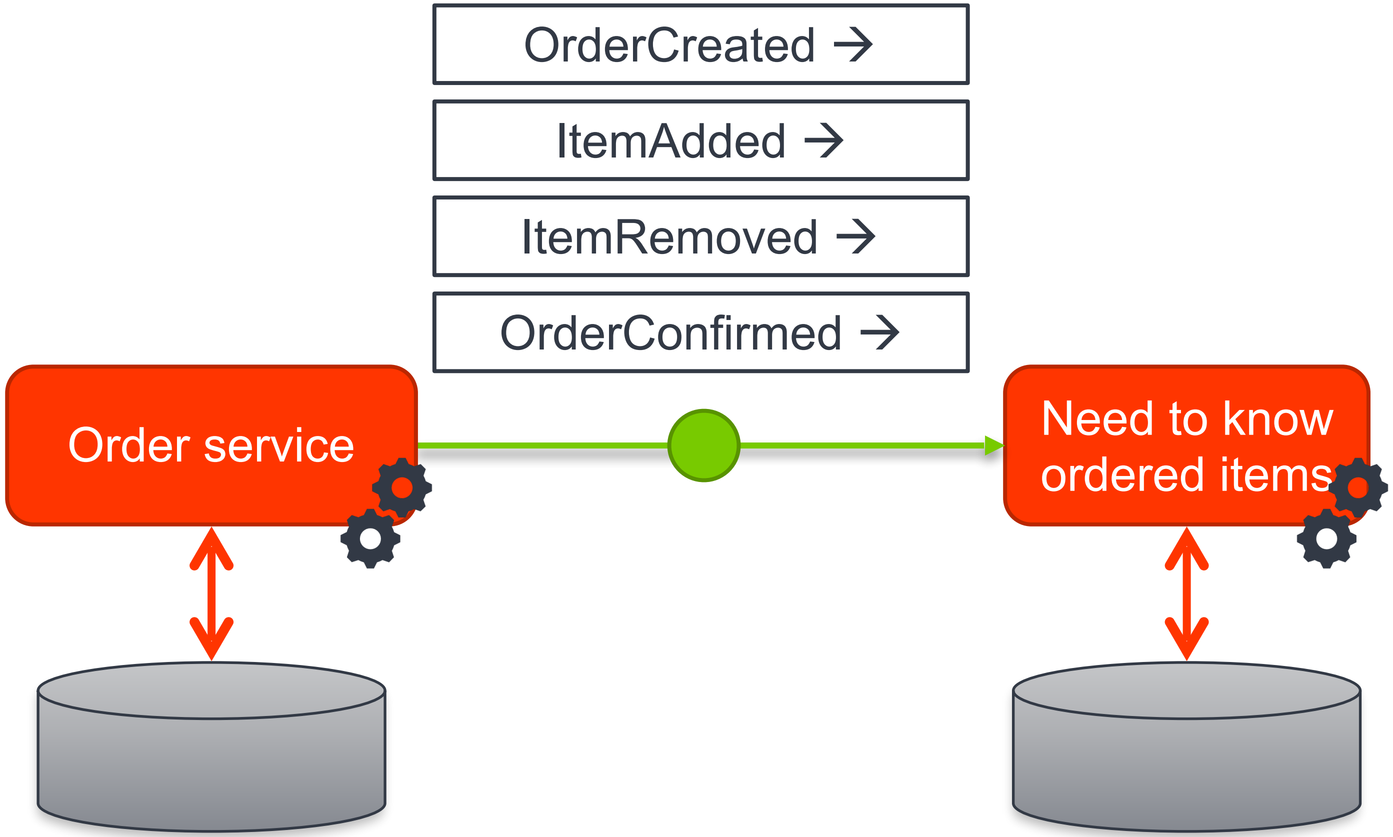
‘Event-Driven’ Microservices

Birmingham Screwdriver

‘Event-Driven’ Microservices

“Maslow Syndrome”

‘Event-Driven’ Microservices



OrderCreated →

ItemAdded →

ItemRemoved →

OrderConfirmed →

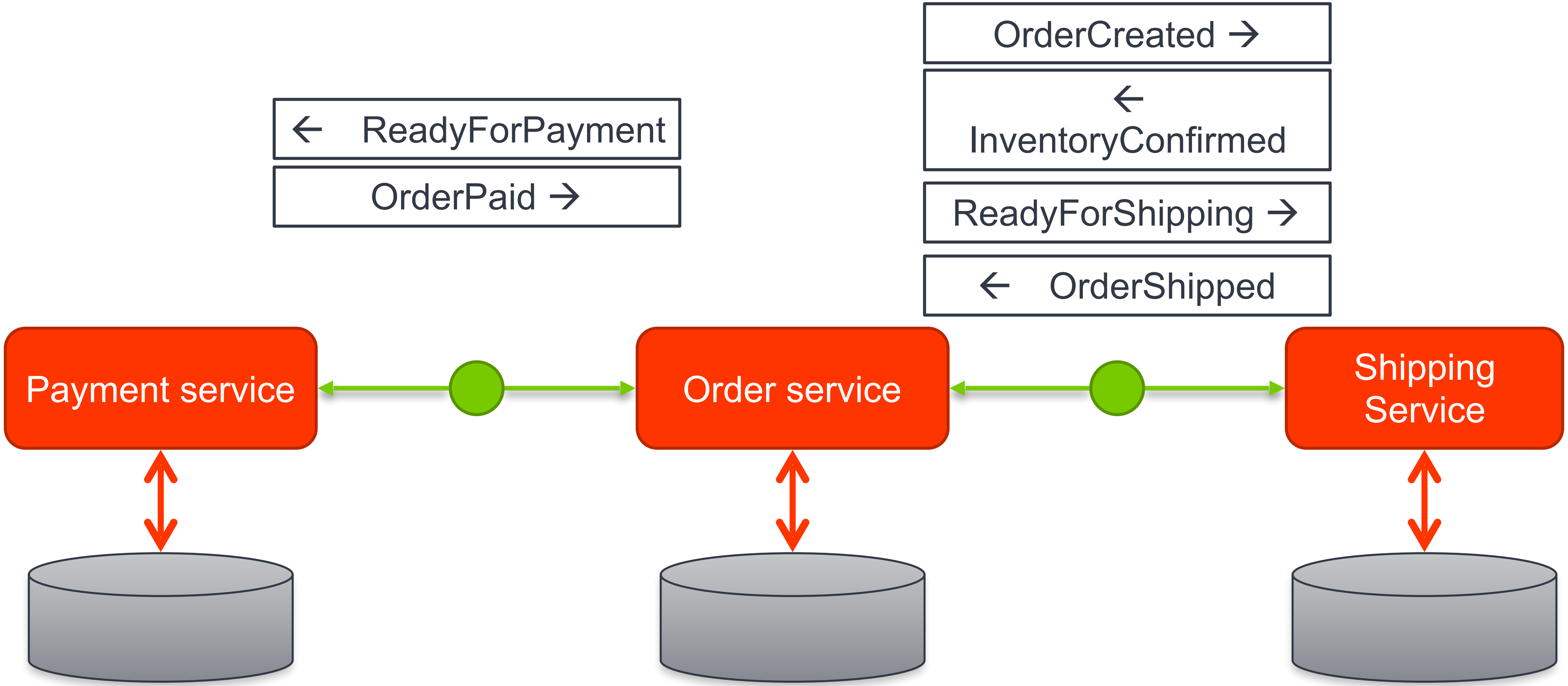
Order service

Need to know ordered items

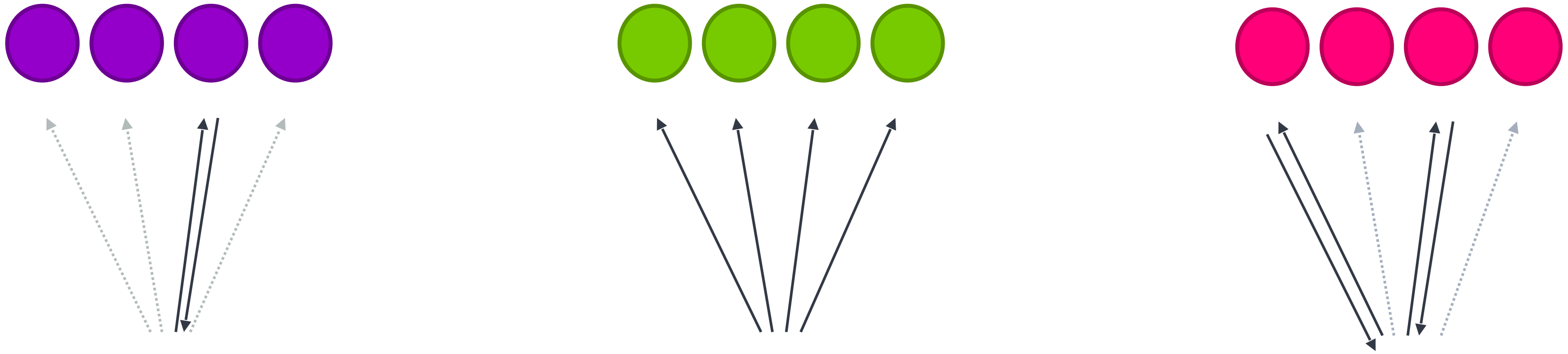
AxonIQ

@allardbz

Or worse…


← ReadyForPayment
OrderPaid →
OrderCreated →
← InventoryConfirmed
ReadyForShipping →
← OrderShipped
Payment service
Order service
Shipping Service

Microservices Messaging



Commands

Route to single handler
Use consistent hashing
Provide result

Events

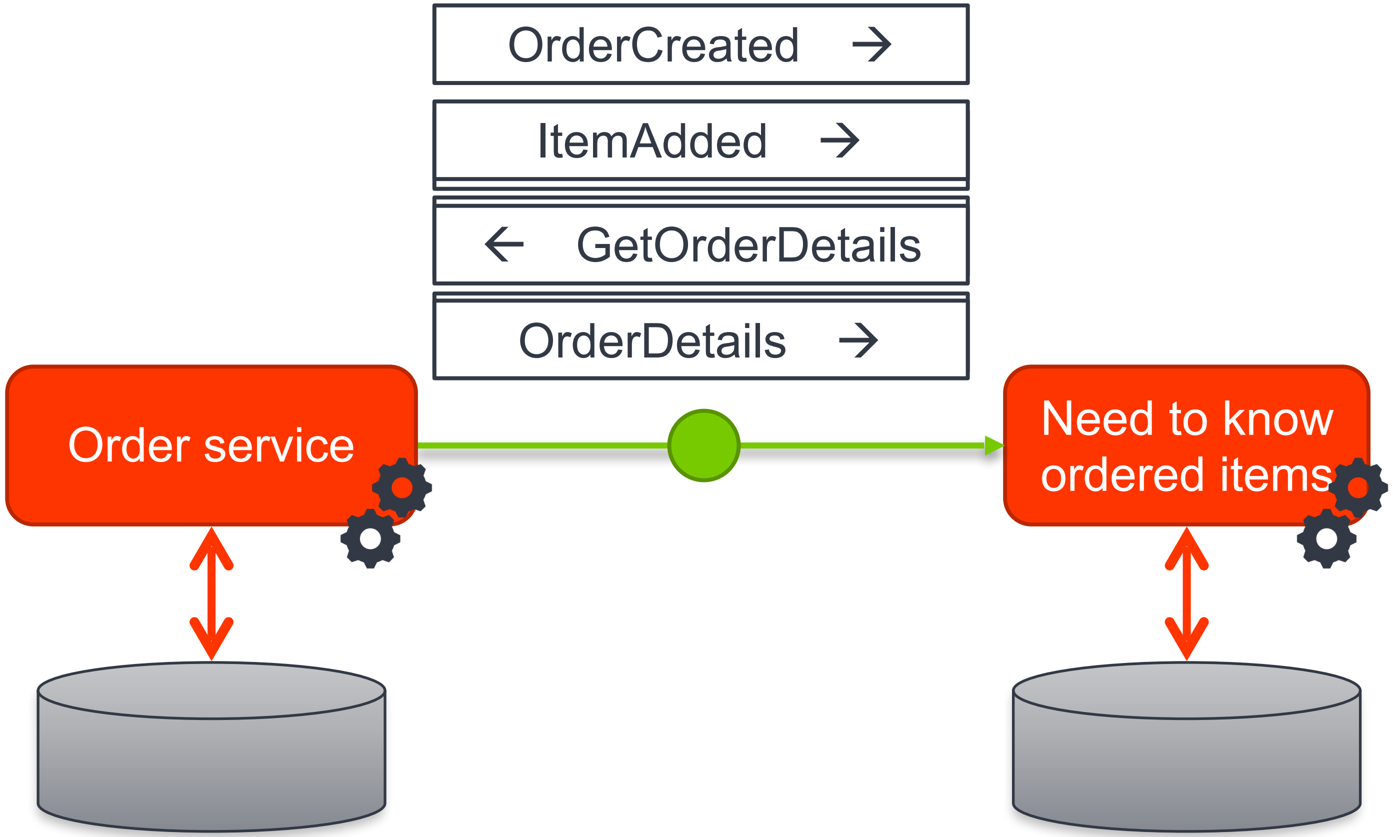
Distribute to all logical handlers
Consumers express ordering req's
No results

Queries

Route with load balancing
Sometimes scatter/gather
Provide result

"Event" and "Message" is not the same thing

‘Event-Driven’ Microservices



OrderCreated →

ItemAdded →

← GetOrderDetails

OrderDetails →

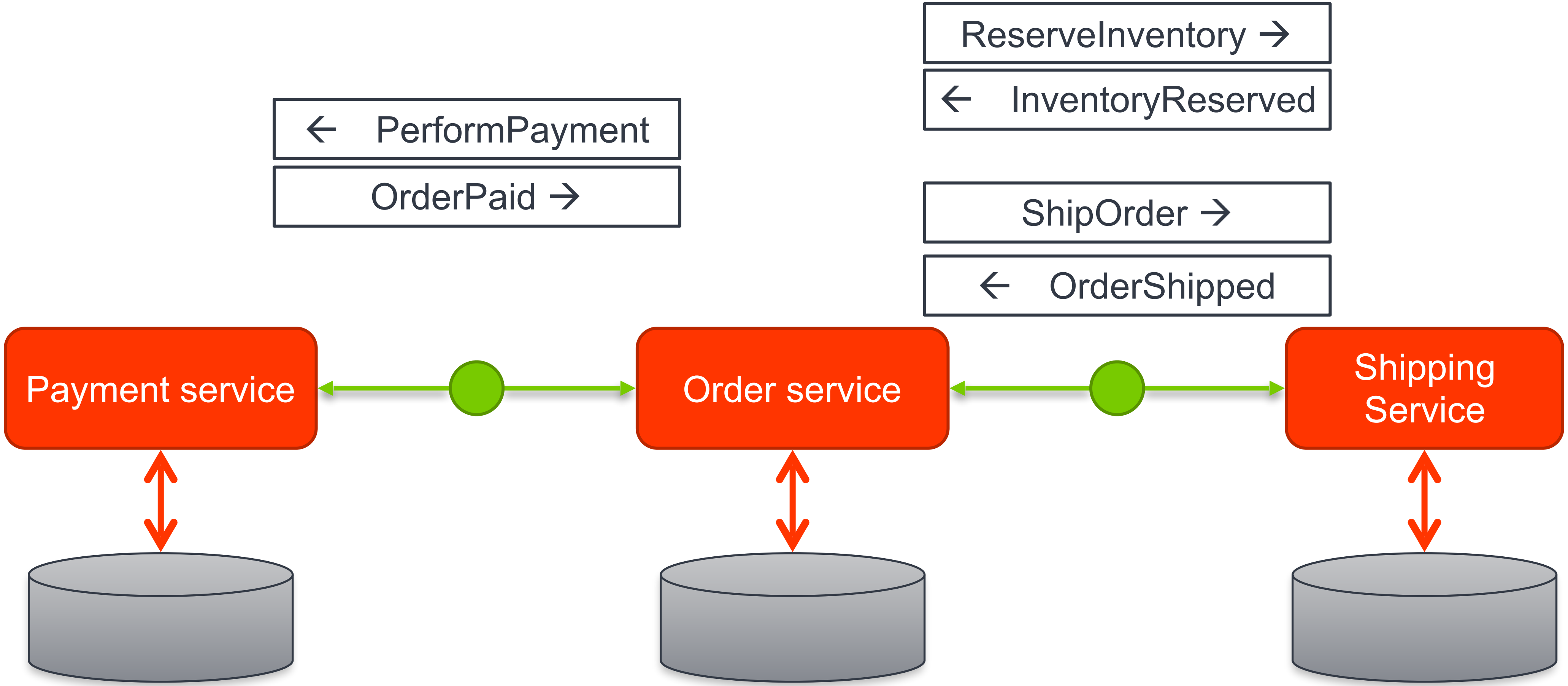
Order service

Need to know ordered items

AxonIQ

@allardbz

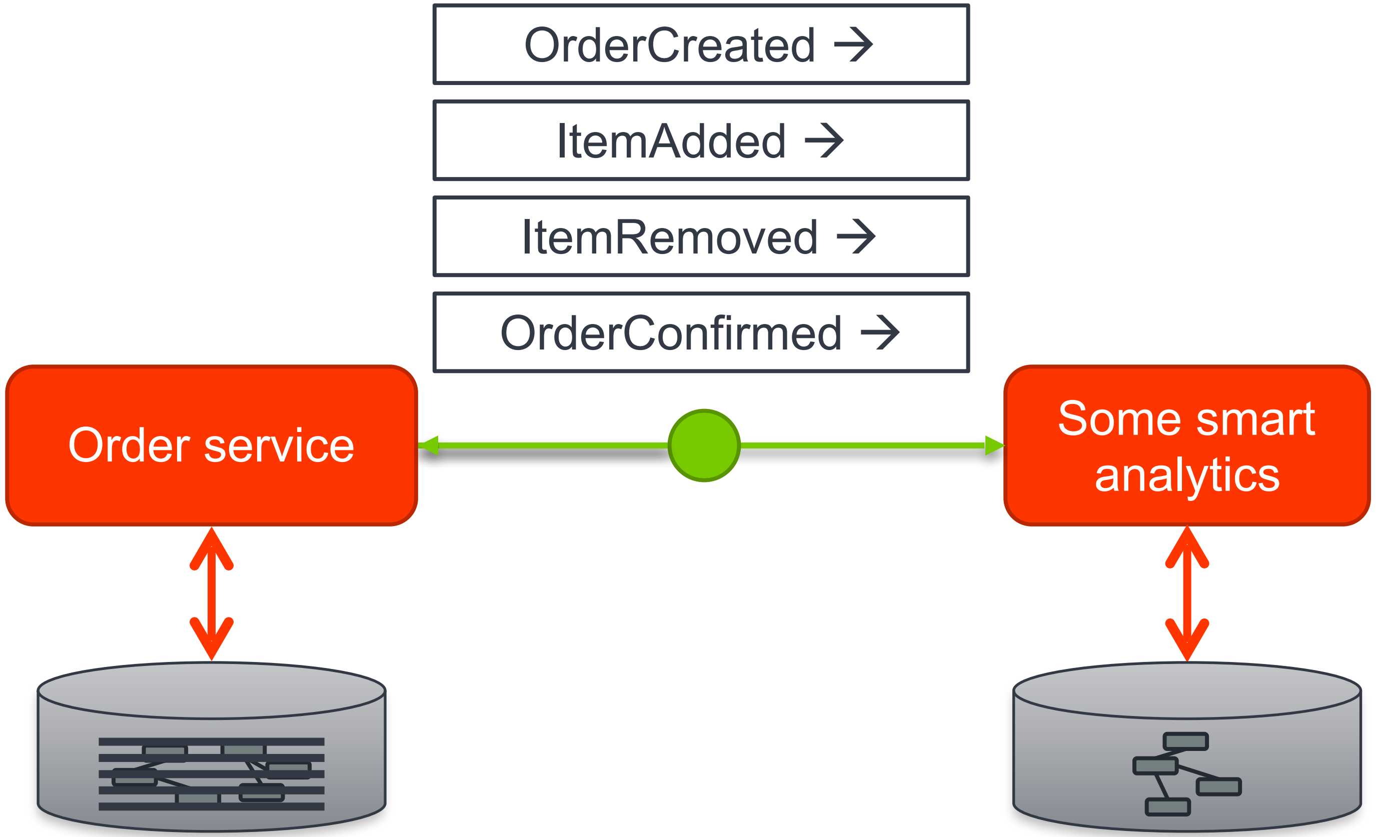
Or…


ReserveInventory →
← InventoryReserved
← PerformPayment
OrderPaid →
ShipOrder →
← OrderShipped
Payment service
Order service
Shipping Service

Events retain value

Event Sourcing is an Architectural pattern in which Events are considered the "source of truth", based on which components (re)build their internal state.

Event Sourcing



OrderCreated →

ItemAdded →

ItemRemoved →

OrderConfirmed →

Order service

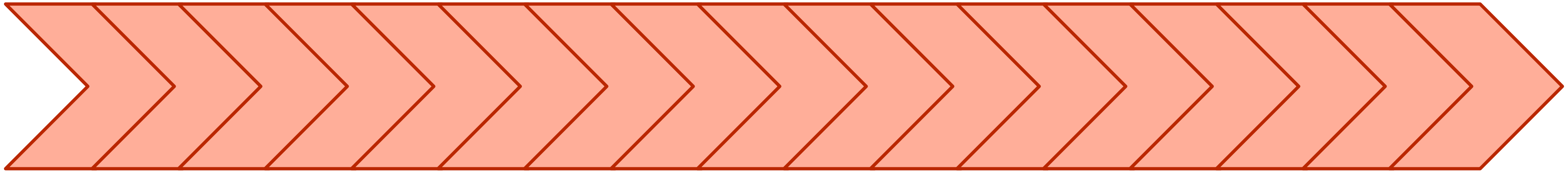
Some smart analytics

Event Store

An Event Store stores the published events to be retrieved both by consumers as well as the publishing component itself.

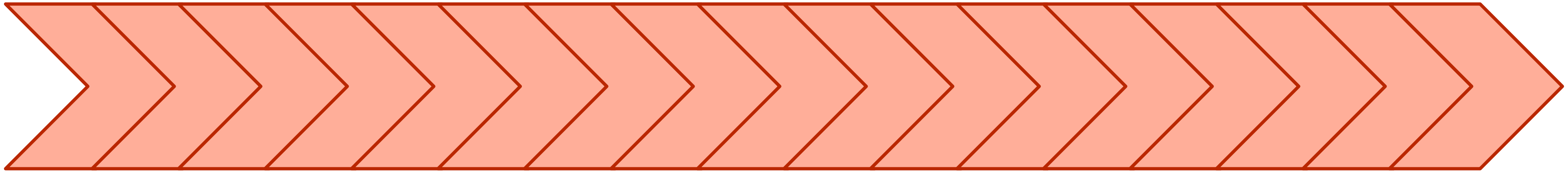
Event Store operations

- Append
- Validate ‘sequence’



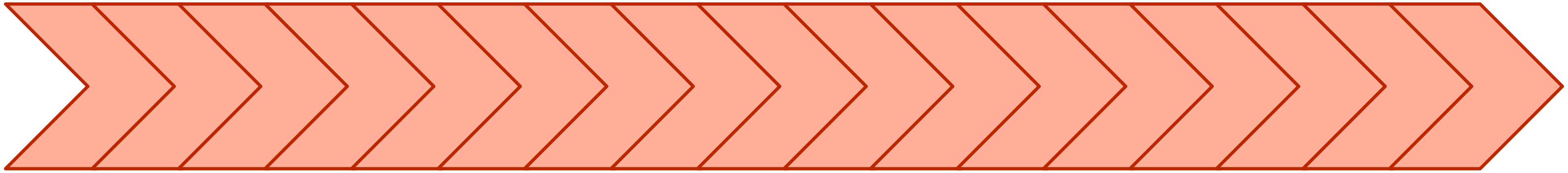
Event Store operations

- Full sequential read



Event Store operations

- Read aggregate's events

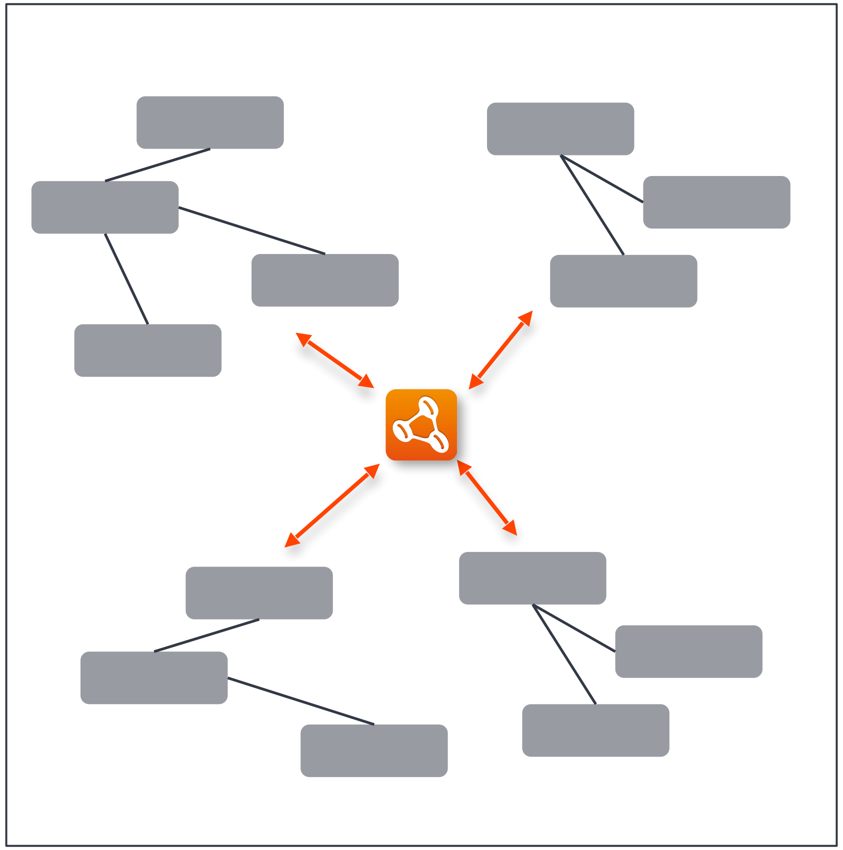


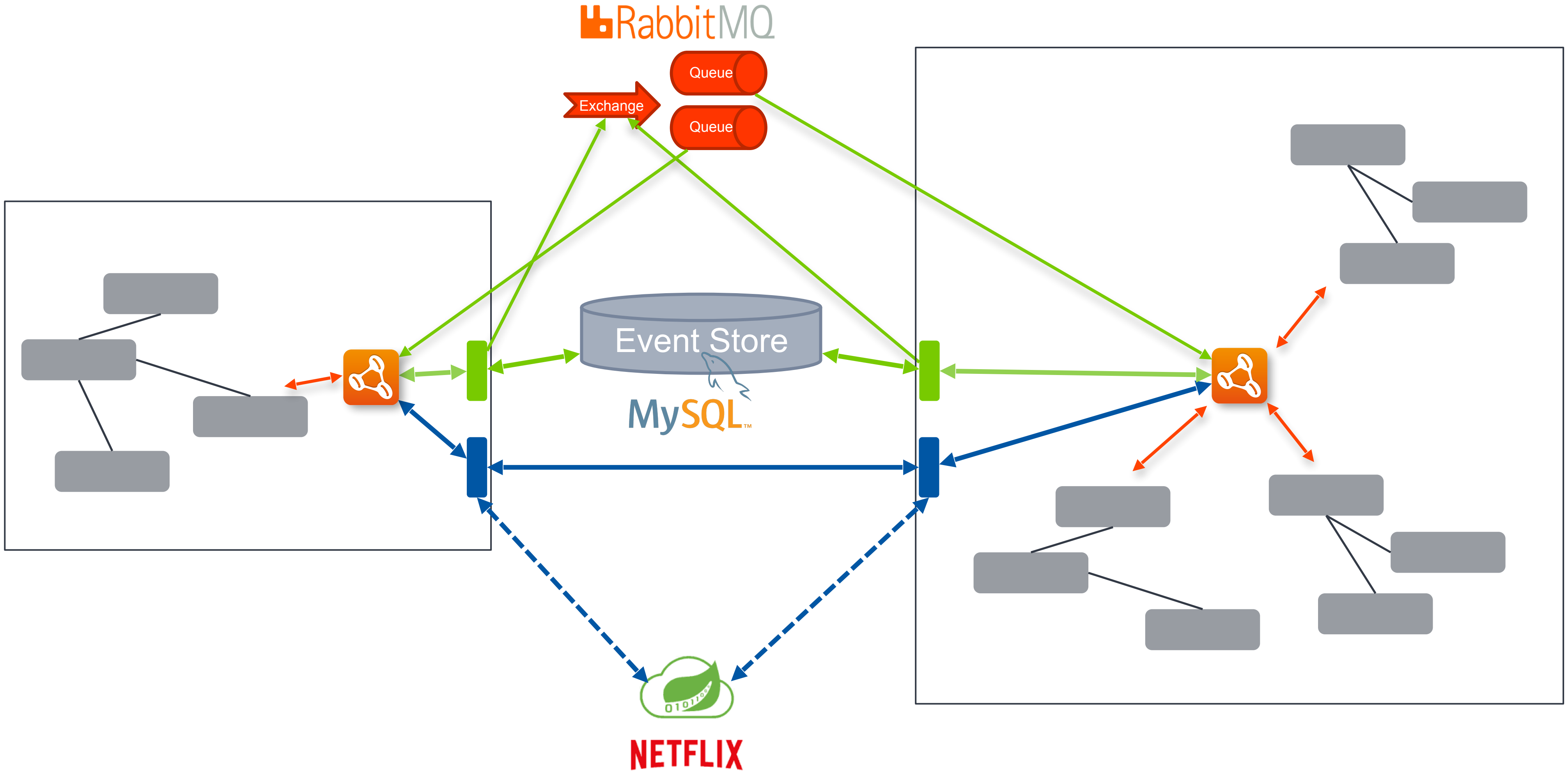
AxonIQ

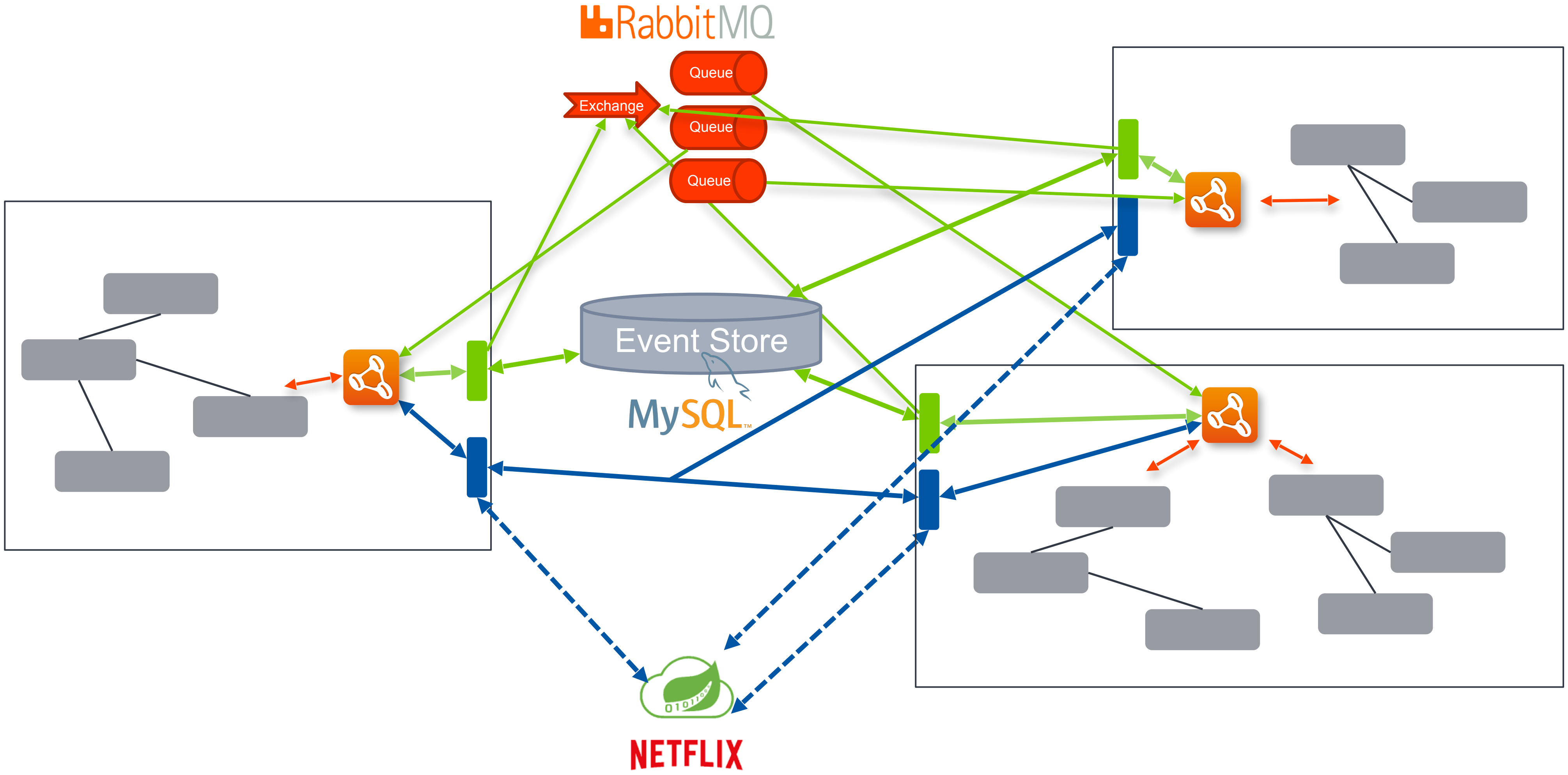
1. Consider explicit messages
2. Define which routing patterns to apply
3. Choose technology/protocol accordingly

AxonFramework AxonServer HTTP GRPC

MySQL RabbitMQ Apache kafka HTTP/2




RabbitMQ
Queue
Exchange
Queue
Event Store
MySQL
NETFLIX


RabbitMQ
Queue
Exchange
Queue
Queue
Event Store
MySQL
NETFLIX

“Just enough” intelligence

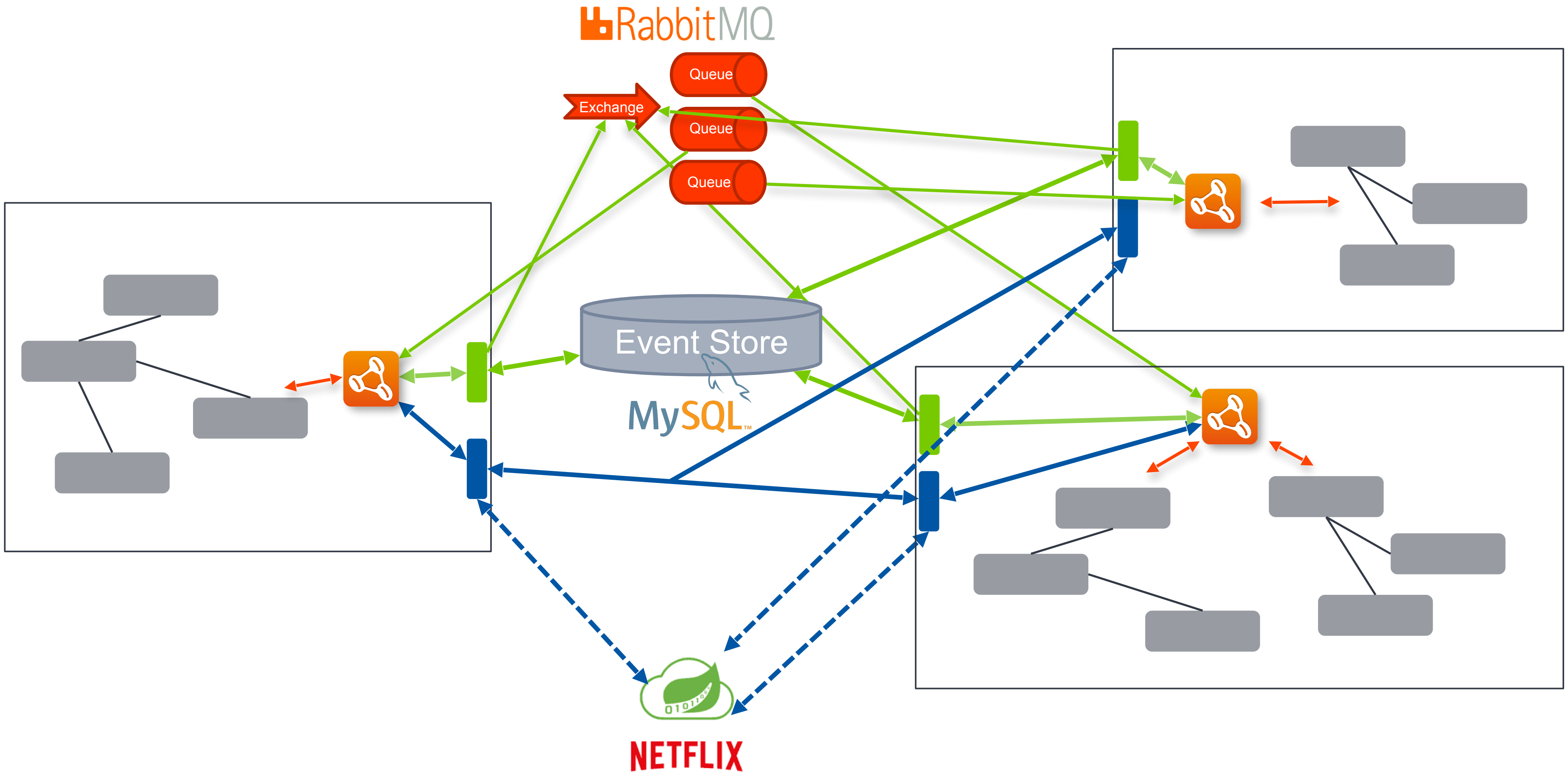
Message Broker
Sends messages. Main value add is reliability.

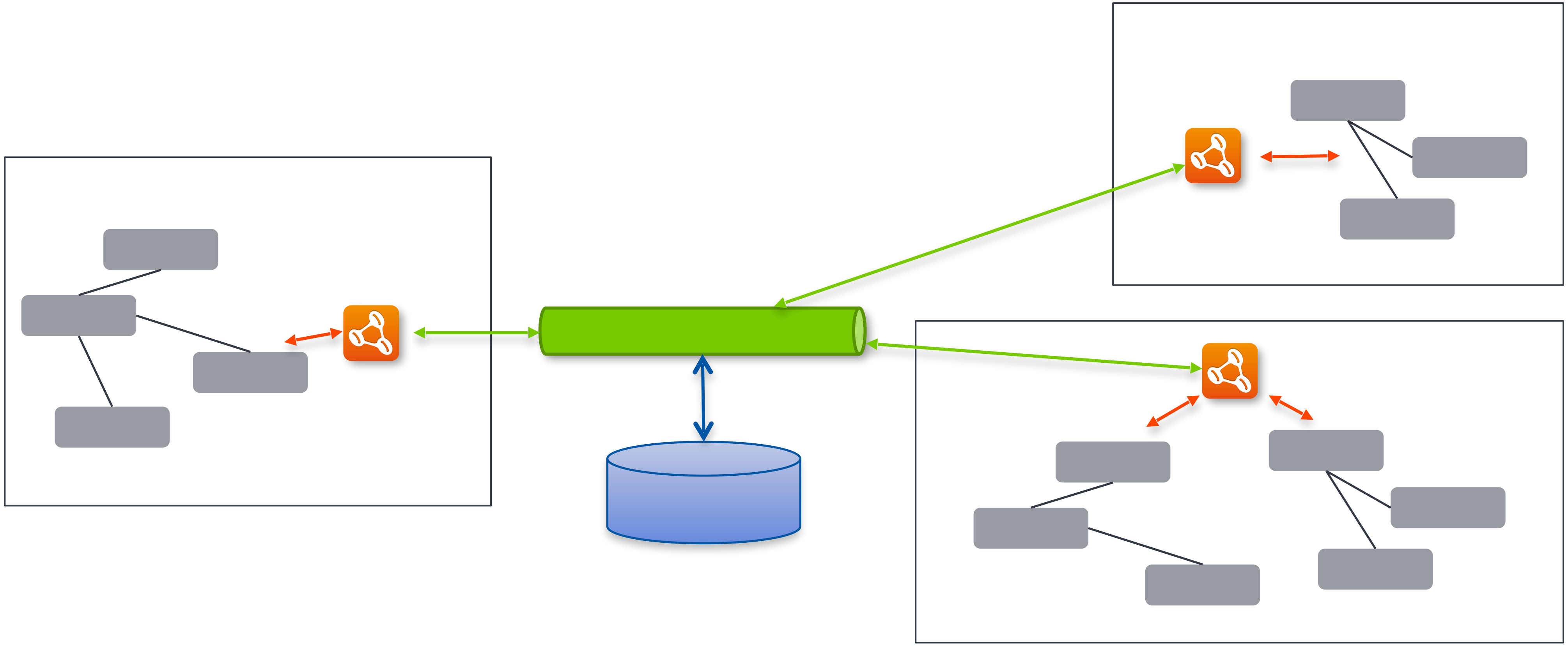
Ideal middle ground ?
Understands difference between Commands, Events, Queries and their routing patterns. Does not care about the content of these messages.

Enterprise Service Bus
Understands message content. Hard to configure and maintain.

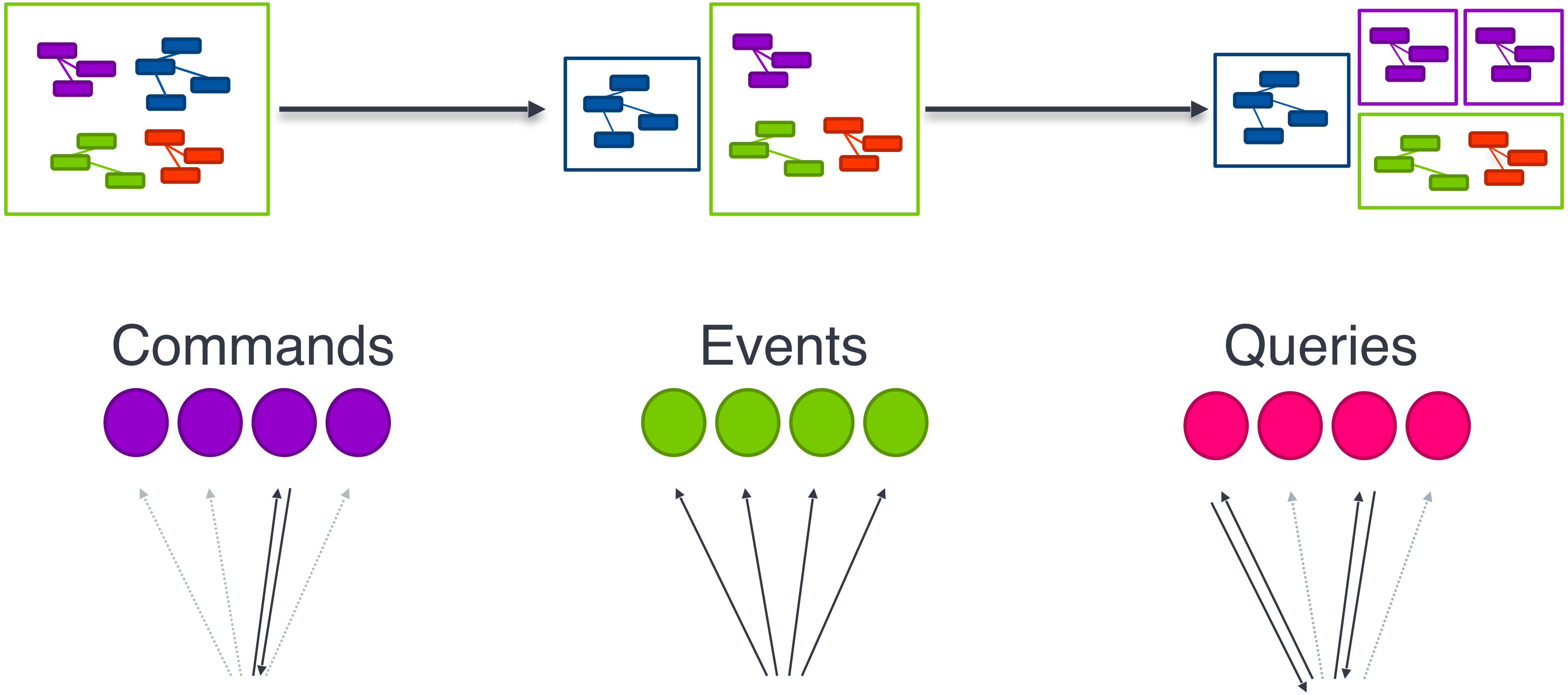
dumb

smart


RabbitMQ
Queue
Exchange
Queue
Queue
Event Store
MySQL
NETFLIX



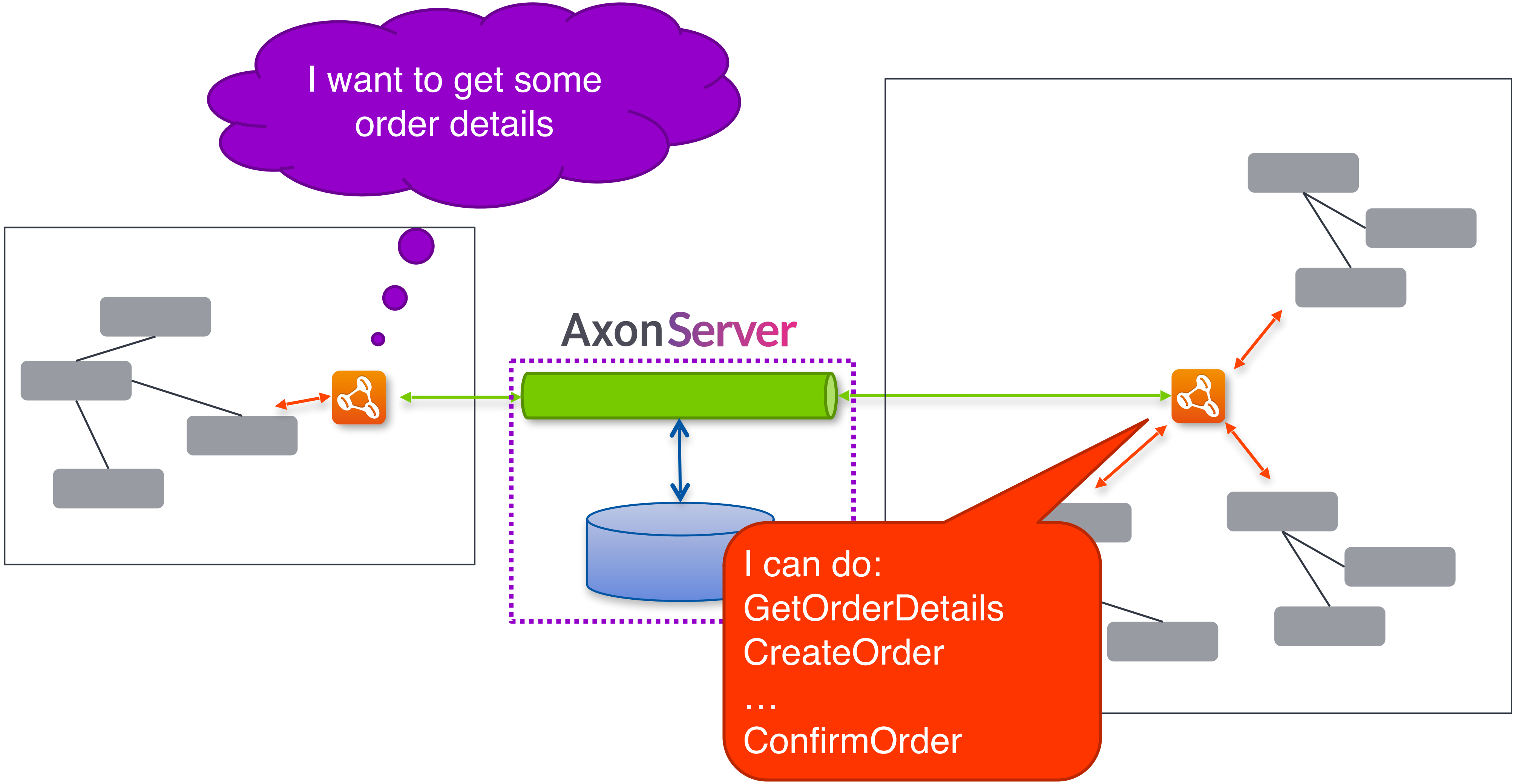
Evolutionary microservices

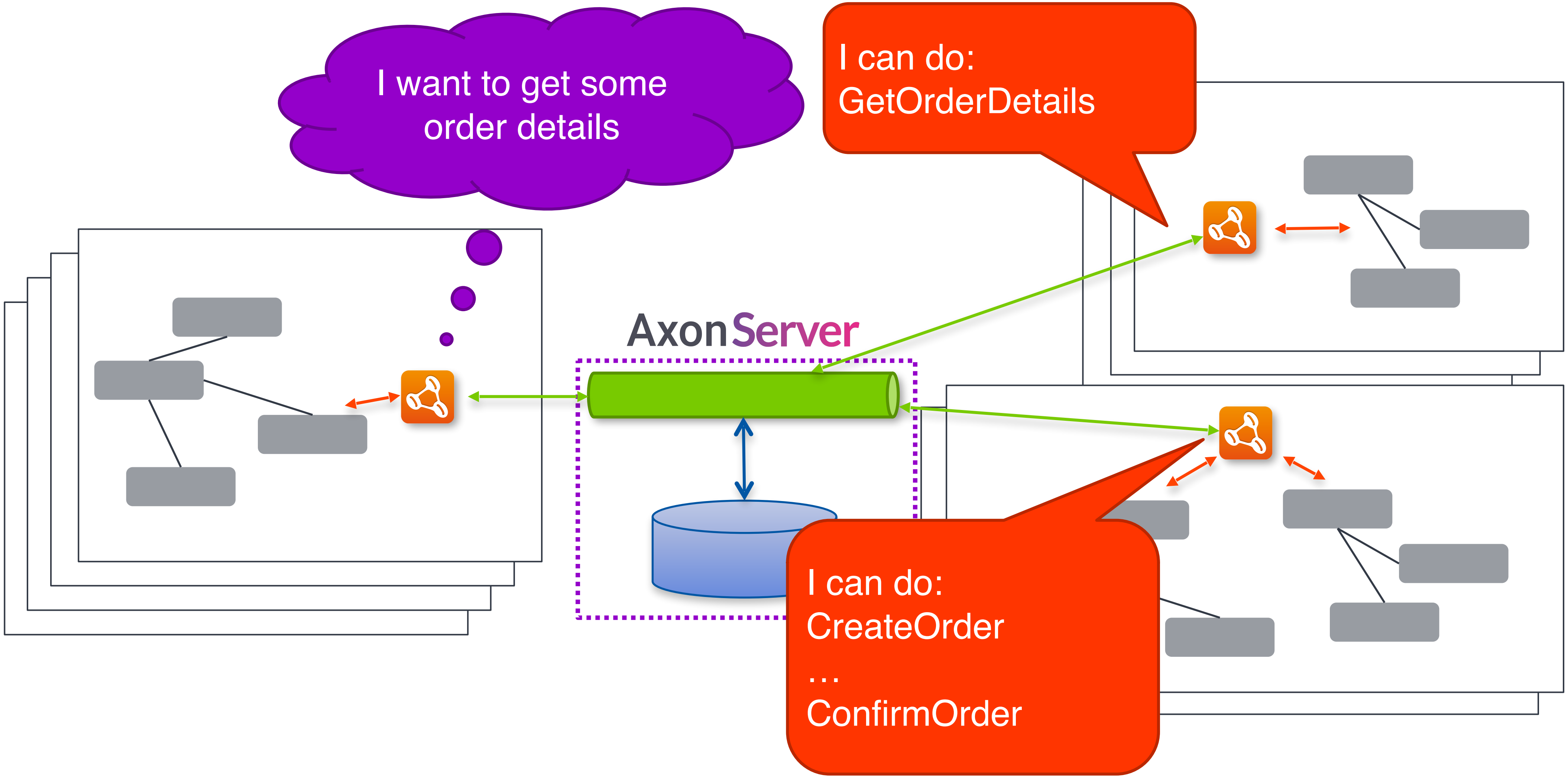


Commands

Events

Queries


I want to get some order details
AxonServer
I can do:
GetOrderDetails
CreateOrder
…
ConfirmOrder


I want to get some order details
I can do: GetOrderDetails
AxonServer
I can do: CreateOrder … ConfirmOrder

At scale, different rules apply

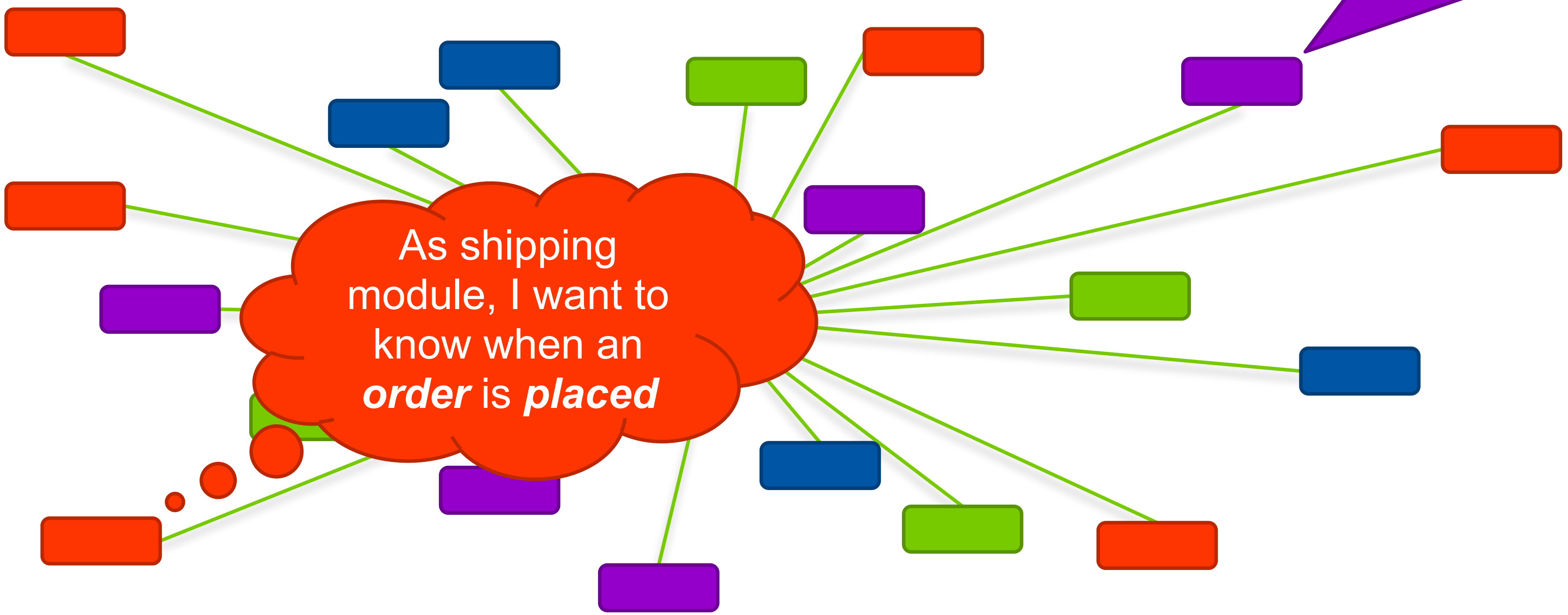
How do you route all these events to all components?

How will this scale?

You Don't!

It Won't!

Unmanageable mess



As shipping module, I want to know when an ***order*** is ***placed***

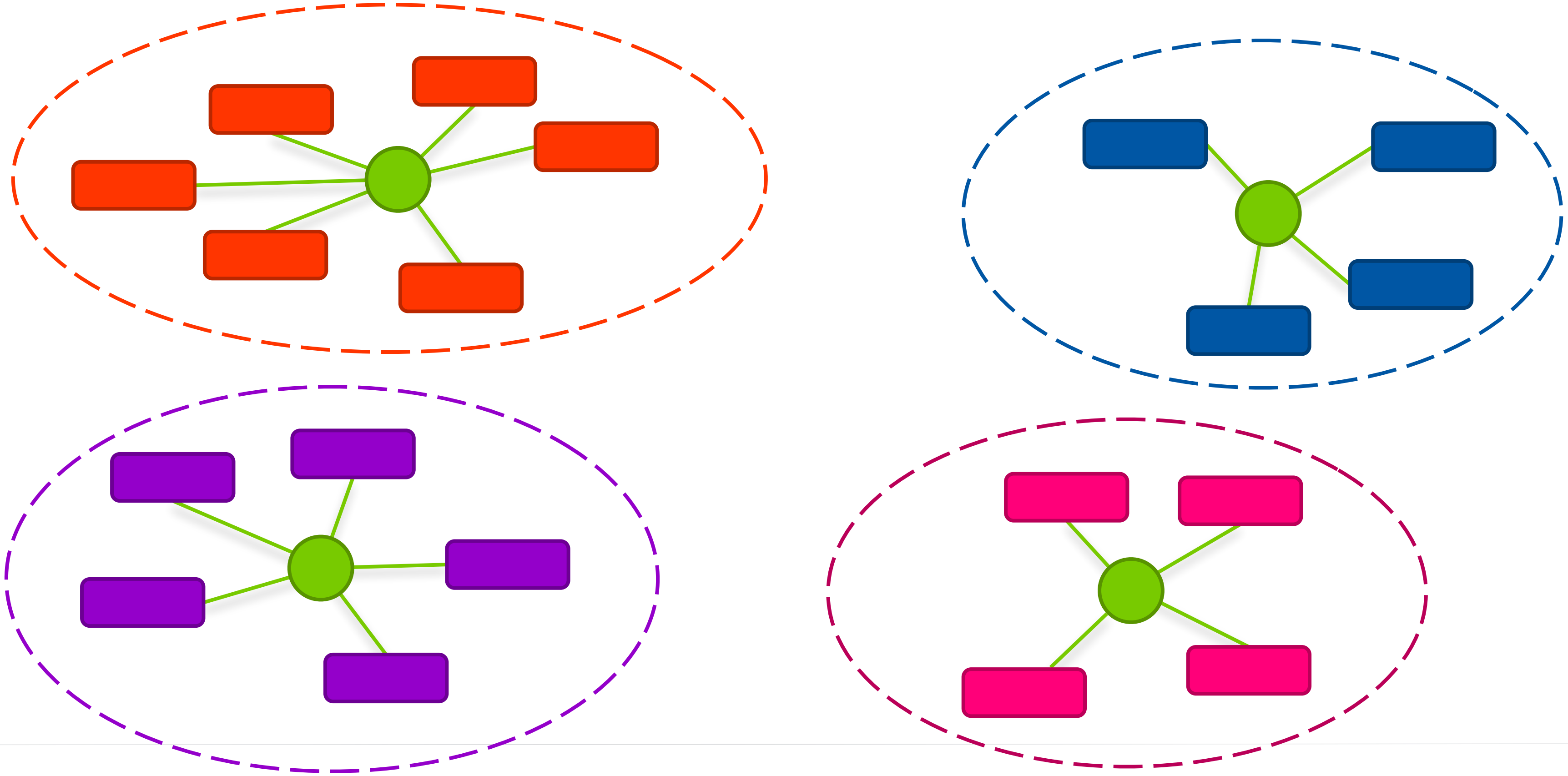
Order Created
Item Added to Order
Shipping Address Added
Billing Address Added
Order Confirmed

Communication = Contract

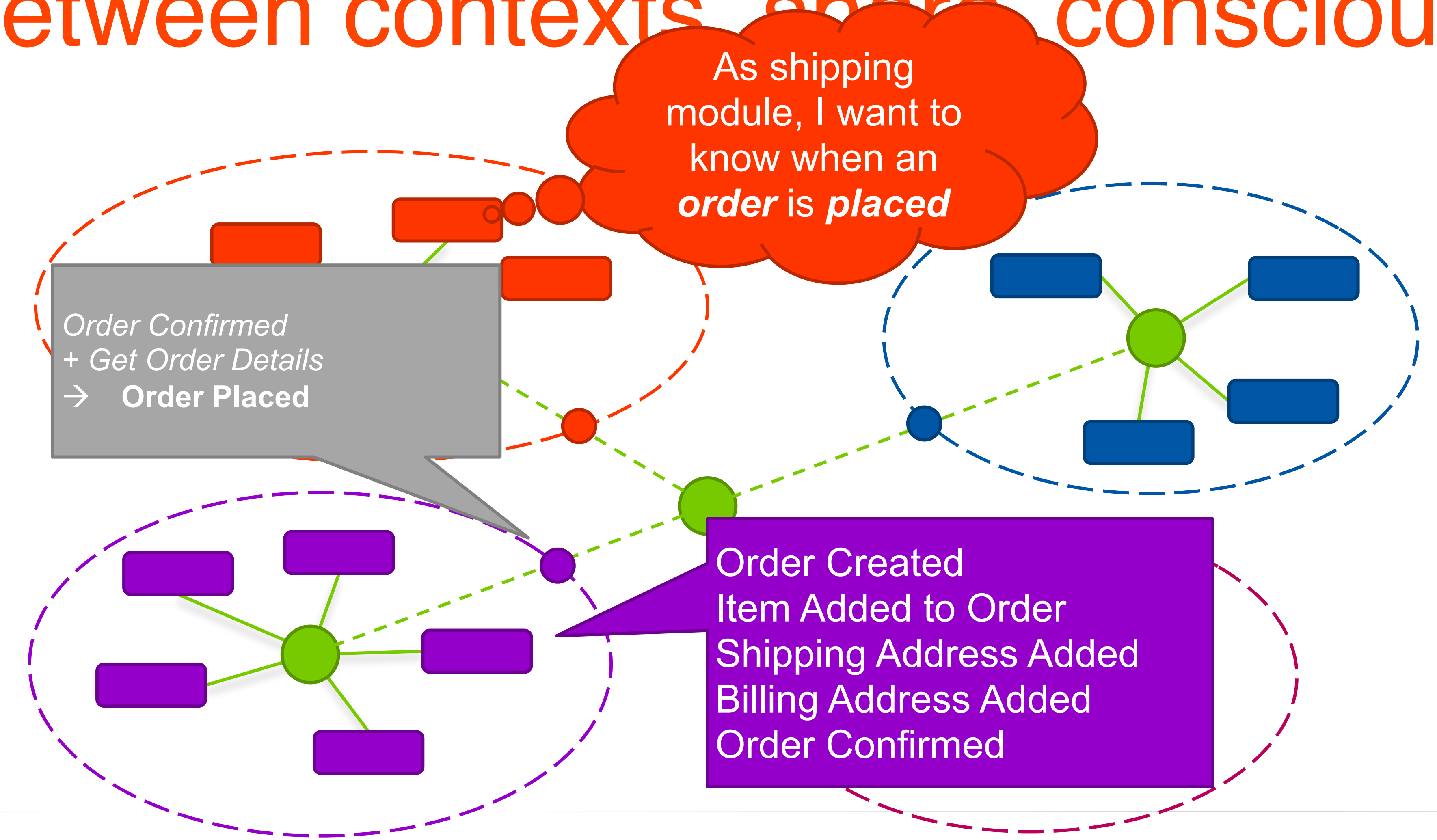
Bounded context

Explicitly define the context within which a model applies. Explicitly set boundaries in terms of team organization, usage within specific parts of the application, and physical manifestations such as code bases and database schemas. Keep the model strictly consistent within these bounds, but don't be distracted or confused by issues outside.

Within a context, share ‘everything’



Between contexts, share ‘consciously’



As shipping module, I want to know when an ***order*** is ***placed***

Order Confirmed
+ Get Order Details
→ **Order Placed**

Order Created
Item Added to Order
Shipping Address Added
Billing Address Added
Order Confirmed

Recap

- Events are useful
    - Don’t forget about Commands and Queries
    - Events retain value
- Modelling messages explicitly stimulates location transparency
    - Beware coupling
    - Consider Bonded Contexts
- “Evolve” your way into microservices

References

- Axon
  - axoniq.io
  - github.com/axonframework
  - github.com/axoniq
  - 🐦 @axonframework
  - 🐦 @axoniq

- QuickStart: axoniq.io/download