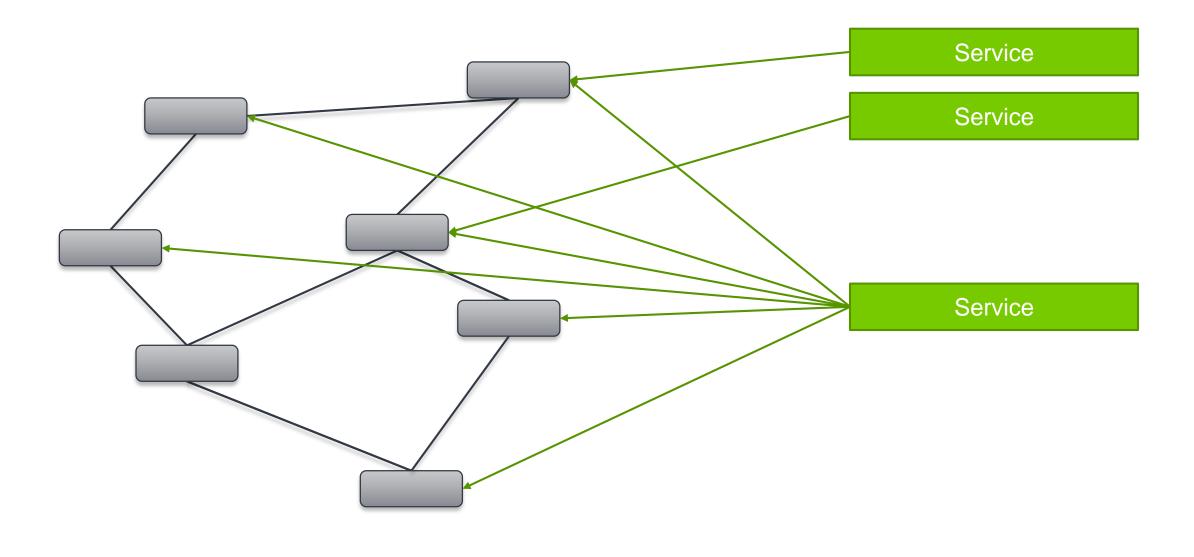
#### Pragmatic Event-Driven Microservices

Allard Buijze – @allardbz – allard@axoniq.io







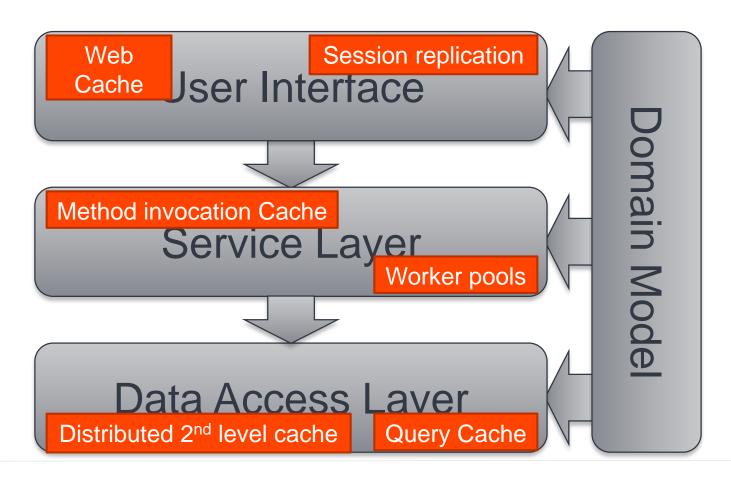
#### 'Normal' SQL QUERY

CREATE ALGORITHM = UNDEFINED DEFINER = 'dbat'@'%' SQL SECURITY DEFINERVIEW 'BUSINESS PROCESS VIEW DISABLED' AS SELECT 'tpo'. NAME' AS 'TRADING PARTNER NAME', 'tbt'. NAME', 'tbt'. NAME' AS 'TRADING PARTNER NAME', 'tbt'. NAME' AS 'TRADING PARTNER NAME', 'tbt'. N 'rtedid'. 'VALUE' AS 'TRADING PARTNER ROOT EID', 'tpot'. 'ORG SUB TYPE NAME' AS 'TRADING PARTNER CUSTOMER TYPE', 'tpot'. 'NAME' AS 'TRADING PARTNER TYPE', 'Tpot'. 'NAME' AS 'TRADING TYPE', 'Tpot'. 'NAME' AS 'TRADING TYPE', 'Tpot'. 'NAME' AS 'TRADING\_PARTNER\_DUNS', 'bglm'.'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', 'ide'.'DOC\_TYPE' AS 'INBOUND\_TYPE', 'ide'.'DOC\_SUB\_TYPE' AS 'INBOUND\_SUB\_TYPE' AS 'INBOUND\_SUB\_TYPE', 'ide'.'DOC\_VERSION' AS 'INBOUND\_VERSION', 'iode'.'DOC\_DIRECTION' AS 'INBOUND\_DIRECTION', 'ode'.'DOC\_DIRECTION', 'ode'.'DOC\_DIRECTION' AS 'OUTBOUND\_TYPE' AS 'OUTBOUND\_SUB\_TYPE', 'ode'.'DOC\_DIRECTION' AS 'OUTBOUND\_TOTAL OF THE CONTROL OF THE CONTRO '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 'ORG\_NAME', 'myo'.'ORGANIZATION\_ID' AS 'NOT\_ORG\_NAME', 'moid'.'VALUE' AS 'ORG\_EID', 'ro'.'ORGANIZATION\_ID' AS 'ROOT\_ORG\_ID', 'ro'.'ORGANIZATION\_ID' AS 'NOT\_ORG\_ID', 'ro'.'ORGANIZATION\_ID' AS 'NOT\_ORGANIZATION\_ID' AS 'NOT\_ORGANIZATION'ID' 'ROOT\_ORG\_NAME', 'xeid'.'VALUE' AS 'ROOT\_ORG\_EID', 'idt'.'DOC\_TYPE\_ID' AS 'INBOUND\_DOC\_TYPE\_ID', 'odt'.'DOC\_TYPE\_ID', 'xel'.'RELATIONSHIP\_ID' AS 'RELATIONSHIP\_ID', 'oop'.'ISAOS' AS 'OB\_ENV\_SNDR\_QUAL', 'oop'.'ISAOS' AS 'OB\_ENV\_SNDR\_ID', 'oop'.'ISAO7' AS 'OB\_ENV\_RCVR\_QUAL', 'oop'.'ISAO8' AS 'OB\_ENV\_RCVR\_ID', 'oop'.'ISAO8' AS 'OB\_ENV\_SNDR\_ID', 'oop'.'ISAO8' AS 'OB\_ENV\_SNDR\_ID', 'oop'.'ISAO7' AS 'IB\_ENV\_RCVR\_QUAL', 'iop'.'ISAO8' AS 'IB\_ENV\_RCVR\_ID', 'iop'.'GSO2' AS 'IB\_IN\_SNDR\_ID', 'iop'.'GSO2' AS 'IB\_IN\_SNDR\_ID', 'iop'.'GSO2' 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', 'iop'.'GSO2' AS 'IB\_IN\_SNDR\_ID', 'IOP'.'GSO2' AS 'IDP'.'GSO2' AS 'bpop'.'IB\_OVR\_ENV\_SNDR\_QUAL' AS 'IB\_OVR\_ENV\_SNDR\_QUAL', 'bpop'.'IB\_OVR\_ENV\_SNDR\_ID' AS 'IB\_OVR\_ENV\_SNDR\_ID' AS 'IB\_OVR\_ENV\_SN 'bpop'.'OB\_OVR\_ENV\_RCVR\_QUAL' AS 'OB\_OVR\_ENV\_RCVR\_QUAL', 'bpop'.'OB\_OVR\_ENV\_RCVR\_ID', 'bpop'.'OB\_OVR\_ENV\_SNDR\_QUAL', 'bpop'.'OB\_OVR\_ENV\_SNDR\_ID' AS 'OB\_OVR\_ENV\_SNDR\_ID' AS 'O AS 'OB\_OVR\_INNER\_RCVR\_ID', 'bpop'.'OB\_OVR\_INNER\_E B TRX ENV SNDR QUAL' AS 'IB TRX ENV SNDR QUAL', 'bpop'.'IB\_TRX\_ENV\_SNDR\_ID' AS 'IB\_TRX\_ENV\_SNDR\_ 'IB TRX INNER E OB\_TRX\_ENV\_RCVR\_QUAL`, 'bpop'.'OB\_TRX\_ENV\_RCVR\_ID' AS 'OB\_TRX\_ENV\_RCVR TRX ENV SNDR I INNER RCVR ID', 'bpop'.'OB\_TRX\_INNER\_ENV\_SNDR\_ID' AS 'OB\_TRX\_IN LATIONSHIP' 'rel' ON (('rel'.'RELATIONSHIP\_ID' = 'bp', 'RELATIONSHIP ID'))) LEFT JOIN 'ORGANIZA' ORGANIZATION I ID' = 'ro'.'ORGANIZATION ID'))) LEFT JOIN NIZATION ID'))) LEFT JOIN 'BOARDING TYPE' 'bt' 'ORGANIZATION' 'tpo' ON ((('tpo'.'ORGANIZATION 22 JOINS 6 SUBQUERIES ORGANIZATION\_ID 'rto'.'ORGANIZATION\_ID') AND ON (('bta'.'BOARDING\_TYPE\_ID' = 'bt'.'BOARDING\_ ('rteid'.'PRIMARY FLAG' = 1) AND ('rteid'.'ORG TYPE ID' FROM 'D )))) LEFT JOIN 'ORG TYPE ASSOC' 'tpoa' ON (('tpoa'.'ORGANIZATION\_ID' = 'tpo'.'ORGANIZATIO oa'.'ORG\_TYPE\_ID myo'.'ORGANIZATION\_ID') AND ('meid'.'PRIMARY\_FLAG' = 1) AND ('meid'.'ORG\_ID DE ID, EROM ,DIW )) LEFT JOIN 'ORG\_IDENTIFIER' 'teid' ON ((('teid'.'ORGANIZATION ID' = 'tpo'.'ORGANIZATION R\_TYPE' WHERE ('DIM\_ORG\_IDENTIFIER\_TYPE'.'NAME' = 'EID'))))) LEFT JOIN 'ORG\_IDENTIFIER' 'tduns '. 'PRIMARY FLAG' R TYPE', 'ORG IDENTIFIER TYPE ID' FROM 'DIM\_ORG\_IDENTIFIER\_TYPE' WHERE ('DIM\_ORG\_IDENTIFIER\_TYPE'. NAME' = 'DUNS'))))) LEFT JOIN 'ORG\_IDENTIFIER\_TYPE\_ID' = (SELECT 'DIM\_ORG\_IDENTIFIER\_TYPE'.'ORG\_IDENTIFIER\_TYPE'D' FROM 'DIM\_ORG\_IDENTIFIER\_TYPE' WHERE ('DIM\_ORG\_IDENTIFIER\_TYPE'.'NAME' = 'GLN'))))) LEFT JOIN 'ORG\_IDENTIFIER\_TYPE'.'ORG\_IDENTIFIER\_TYPE'D' AND ('reid'.'ORGANIZATION\_ID' = 'ro'.'ORGANIZATION\_ID') AND ('reid'.'ORGANIZATION\_ID') AN 1) AND ('reid'.'ORG\_IDENTIFIER\_TYPE\_ID' = (SELECT 'DIM\_ORG\_IDENTIFIER\_TYPE\_'.'ORG\_IDENTIFIER\_TYPE\_'.'NAME' = 'EID'))))) LEFT JOIN 'ORG\_DOC\_TYPE' 'iodt' ON (((('iodt'.'ORG\_DOC\_TYPE\_ID' = 'EID')))))) LEFT JOIN 'ORG\_DOC\_TYPE' 'iodt' ON (((('iodt'.'ORG\_DOC\_TYPE\_ID' = 'EID')))))) LEFT JOIN 'ORG\_DOC\_TYPE' 'iodt' ON (((('iodt'.'ORG\_DOC\_TYPE\_ID' = 'EID'))))))) LEFT JOIN 'ORG\_DOC\_TYPE' 'iodt' ON (((('iodt'.'ORG\_DOC\_TYPE\_ID' = 'EID')))))))))) LEFT JOIN 'ORG\_DOC\_TYPE' 'iodt' ON (((('iodt'.'ORG\_DOC\_TYPE\_ID' = 'EID')))))))) LEFT JOIN 'ORG\_DOC\_TYPE\_ID' 'IODT' ORG\_DOC\_TYPE\_ID' 'IODT' ORG\_DOC\_TYPE\_I 'bp'.'IB\_DOC\_TYPE\_ID') AND ((('myo'.'ORGANIZATION\_ID' = 'rel'.'PARTY1\_ID') AND (('bp'.'PARTY1\_ID') AND ('bp'.'PARTY1\_DIRECTION' = 'FROM')) OR (('myo'.'ORGANIZATION\_ID' = 'rel'.'PARTY1\_DIRECTION' = 'TO')))) OR (('iodt'.'ORG\_DOC\_TYPE\_ID' = 'rel'.'PARTY1\_DIRECTION' = 'bp'.'OB\_DOC\_TYPE\_ID') AND ((('myo'.'ORGANIZATION\_ID' = 'rel'.'PARTY1\_ID') AND ('bp'.'PARTY1\_ID') AND ('bp'.'PARTY (('idt'.'DOC\_TYPE\_ID' = 'iodt'.'DOC\_TYPE\_ID') AND (('myo'.'ORGANIZATION\_ID' = 'rel'.'PARTY1\_DIRECTION' = 'TO')) OR (('myo'.'ORGANIZATION\_ID' = 'rel'.'PARTY1\_DIRECTION' = 'TO')) OR (('myo'.'ORGANIZATION\_ID' = 'rel'.'PARTY1\_DIRECTION' = 'TO')) = 'xel'.'FARTY2\_ID') AND ('bp'.'FARTY1\_DIRECTION' = 'FROM'))) OR (('oodt'.'ORG\_DOC\_TYPE\_ID' = 'bp'.'ORGANIZATION\_ID' = 'xel'.'FARTY1\_ID') AND ('bp'.'FARTY1\_DIRECTION' = 'FROM')) OR (('myo'.'ORGANIZATION\_ID' = 'xel'.'FARTY1\_ID') AND ('bp'.'FARTY1\_DIRECTION' = 'FROM')) 'rel'.'PARTYZ\_ID') AND ('bp'.'PARTYI\_DIRECTION' = 'TO')))))) LEFT JOIN 'DIM\_DOC\_TYPE' 'odt' ON (('odt'.'DOC\_TYPE\_ID' = 'oodt'.'DOC\_TYPE\_ID'))) LEFT JOIN 'BUSINESS\_FROCESS\_OVERRIDE\_FARAM' 'bpop' ON (('bpop'.'BUSINESS\_FROCESS\_OVERRIDE\_FARAM\_ID' = 'TO')))) 'bp'.'BUSINESS PROCESS ID'))) LEFT JOIN 'ORG DOC TYPE PARAMETER BUSINESS PROCESS VIEW' 'iop' ON (('iop'.'ID' = 'bp'.'B DOC TYPE JOIN 'ORG DOC TYPE DATAMETER 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'.'PARTY1\_ID'))) AND ISNULL('bp'.'ORGANIZATION\_ID' = 'rel'.'PARTY1\_ID')))



#### Layered architecture















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

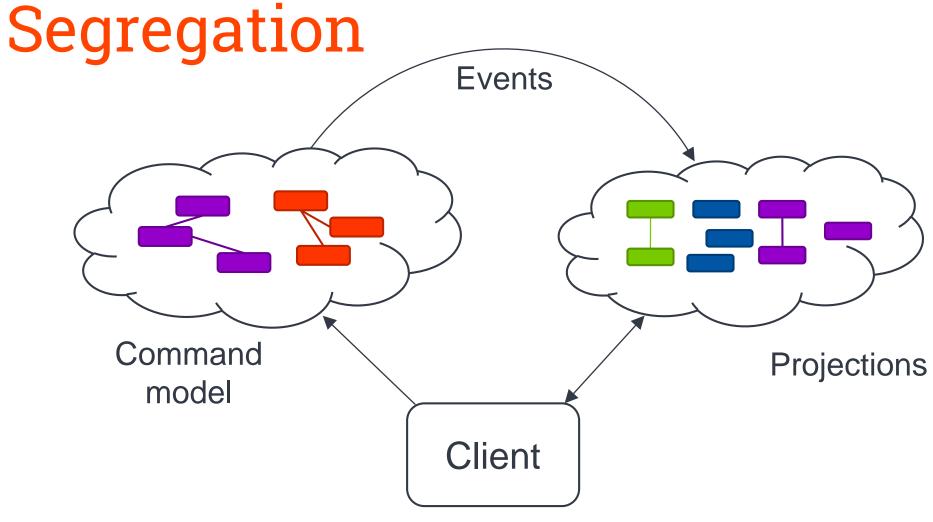








Command Query Responsibility





#### Monoliths







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



#### Microservices vs Monoliths

#### **Monoliths**

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

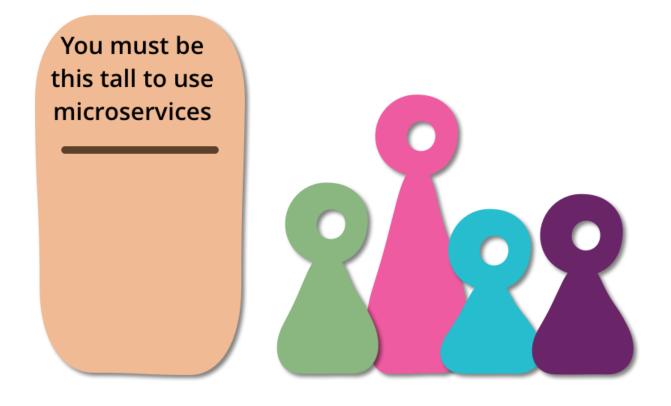
#### Microservices system

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



#### Are you tall enough?





Noun? -> Service!



# OrderService



## CustomerService



## ProductService



# InventoryService

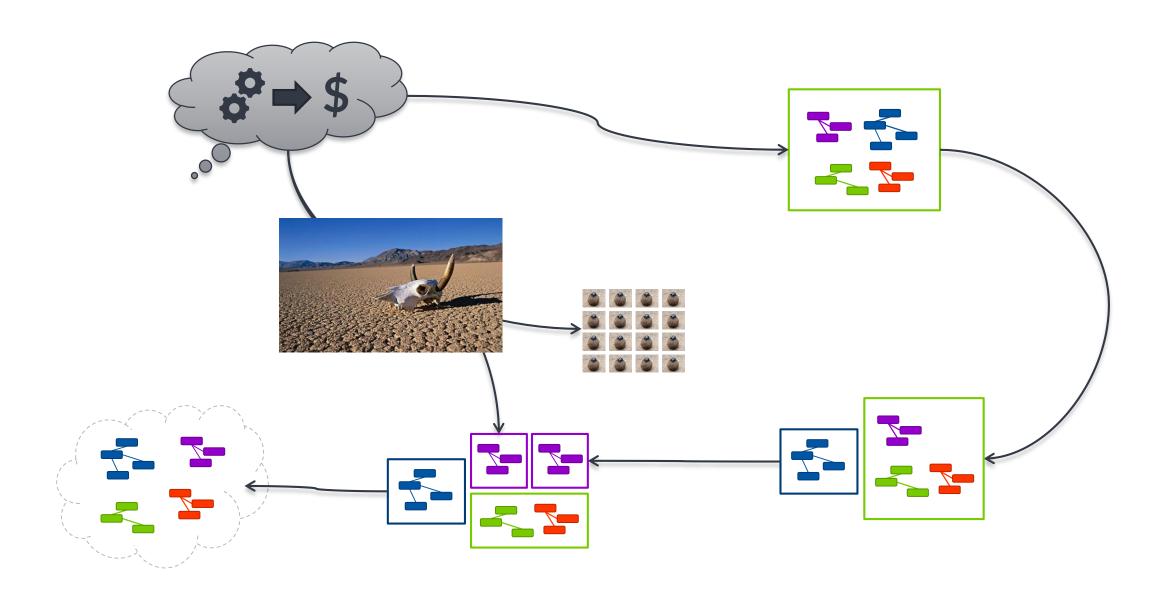






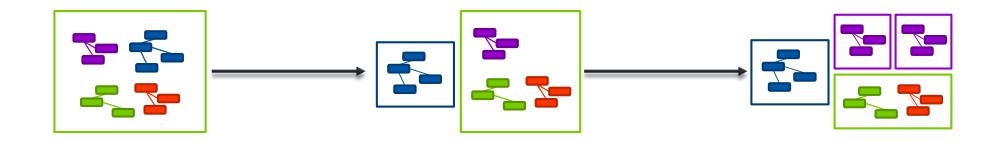








#### Location transparency



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)

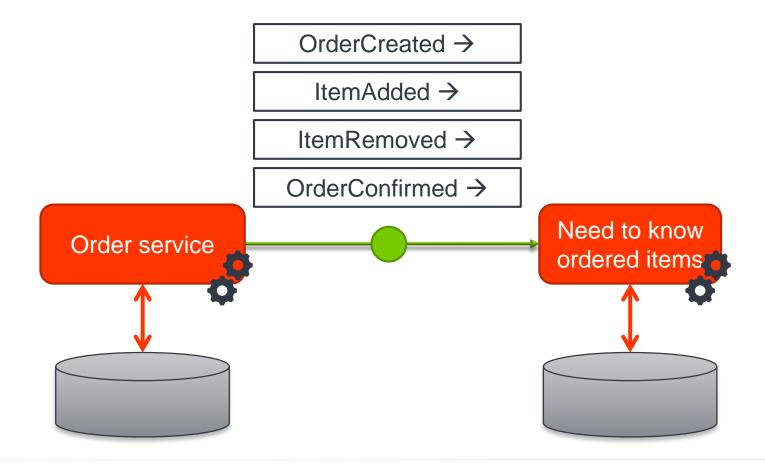


#### Reasons to send a message

Event Something has happened



#### 'Event-Driven' Microservices





#### Reasons to send a message

Event Something has happened

Command I want the system to do something

Query I want to know something

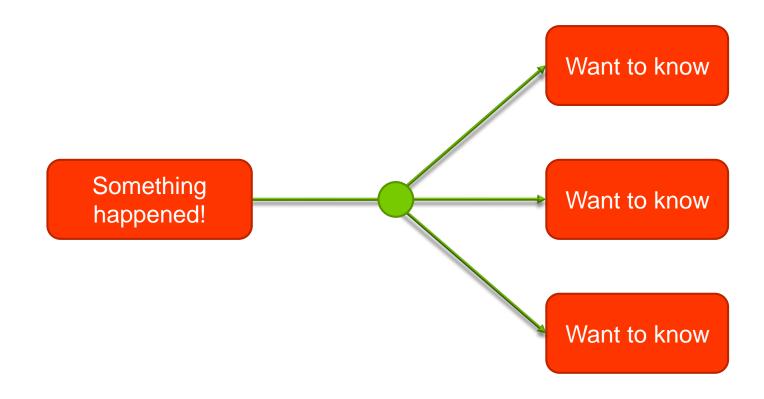


#### Something has happened - Event

- Data change
- Deadline passing
- Or anything else that's relevant in the domain

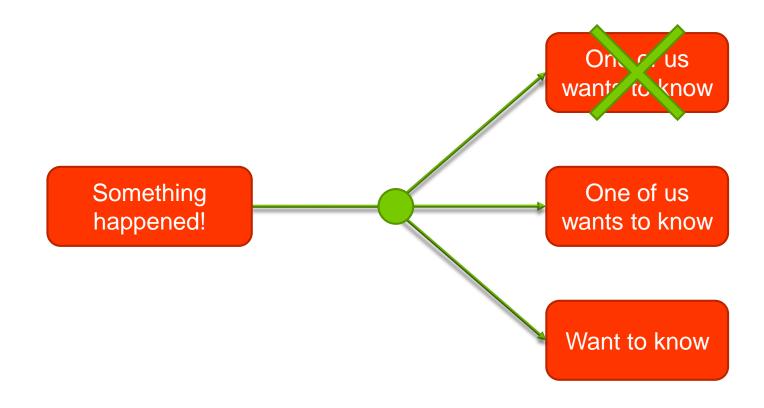


#### Publish-subscribe



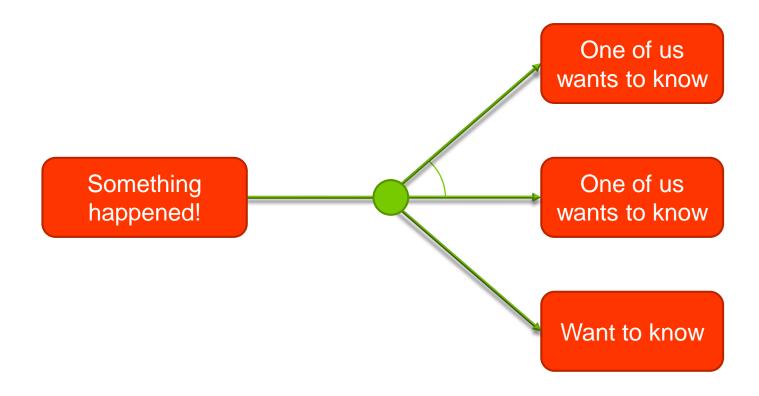


#### Exclusive consumers



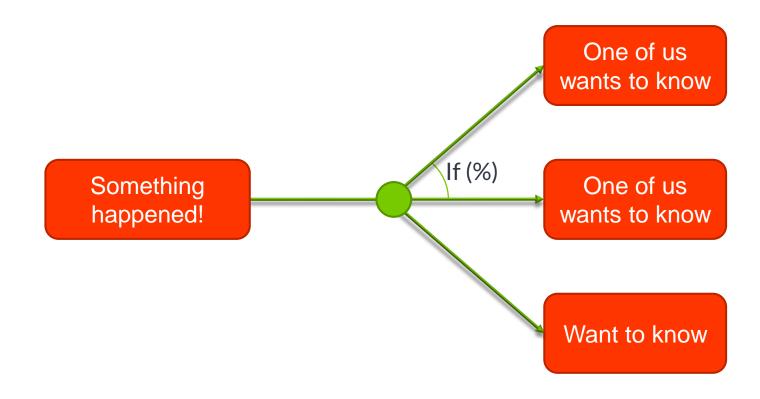


#### Competing consumers





#### Balanced consumers



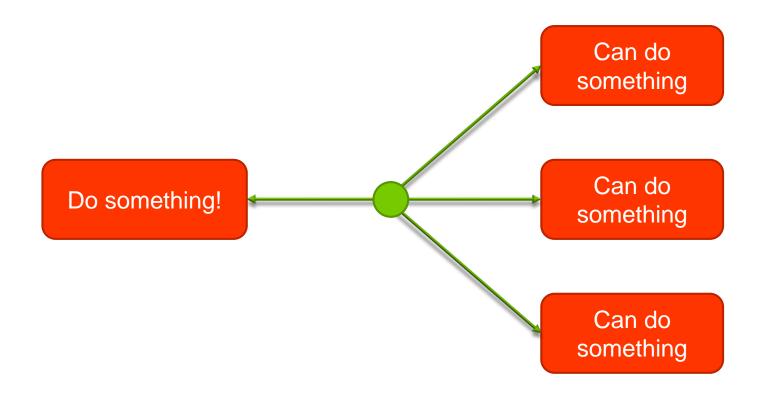


# I want something done - Command

- Request-a-side-effect
  - Change data / application state
  - Send email
- Exactly 1 destination
- OK / NOK reply
  - Maybe some data

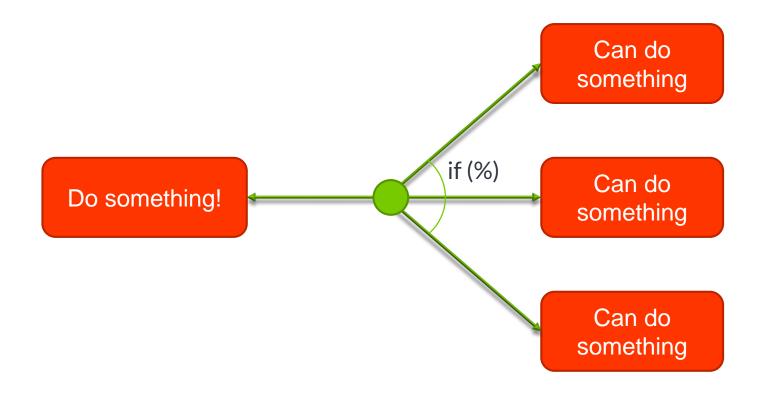


#### Command Routing





#### Command Routing



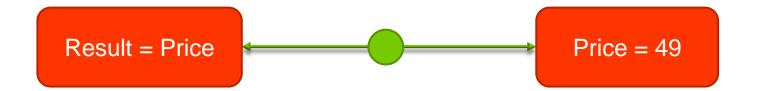


#### I want to know something -Query

- Desire for information
  - The response has more value than the question
  - (Usually) side-effect free
- Different messaging patterns
  - Single destination
  - Scatter gather query
  - Subscription

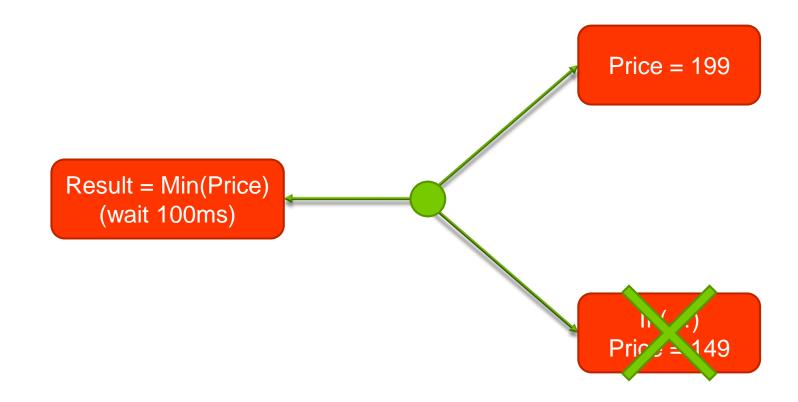


#### Query – point to point



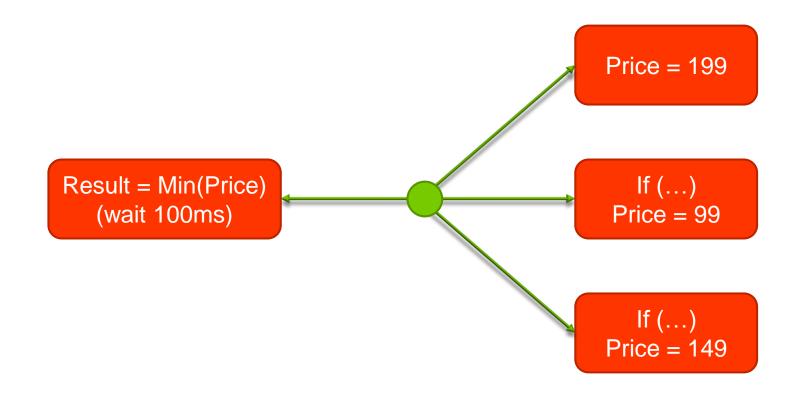


#### Query – scatter-gather



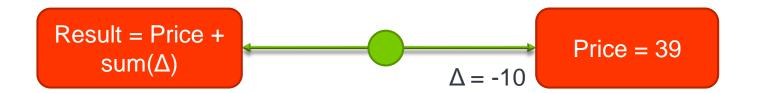


#### Query – scatter-gather



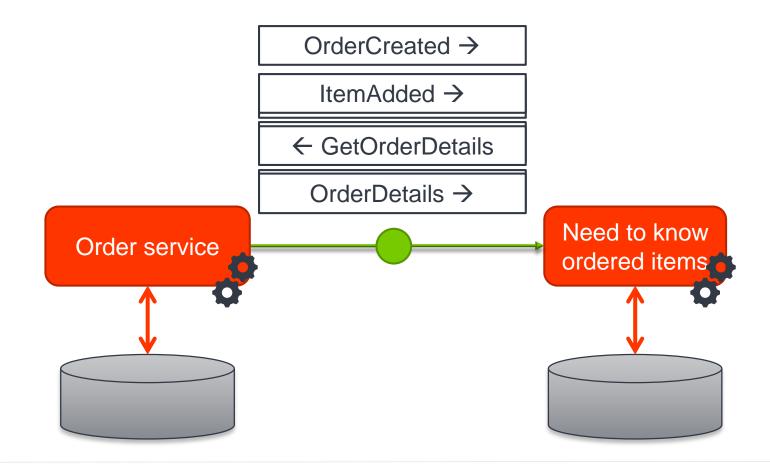


### Query – subscription





#### 'Event-Driven' Microservices





#### 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.

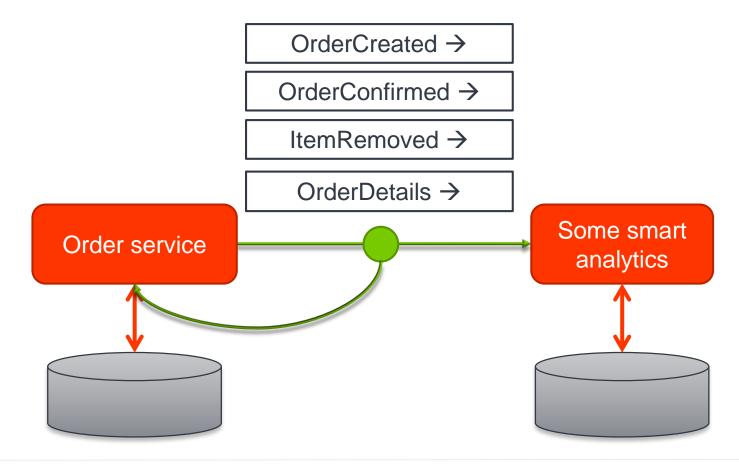


#### **Event Store**

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



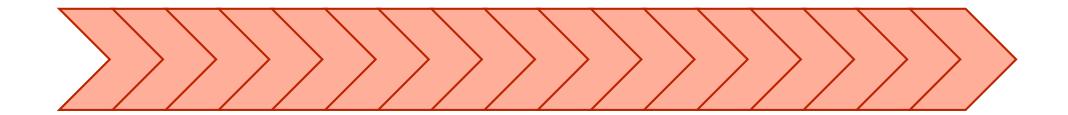
# **Event Sourcing**





#### **Event Store operations**

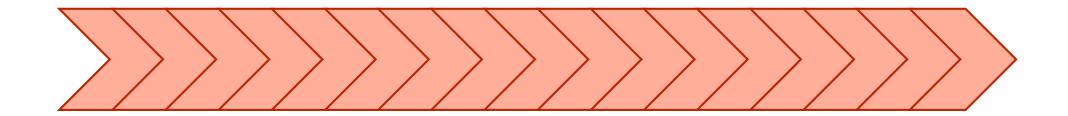
- Append
- Validate 'sequence'





## **Event Store operations**

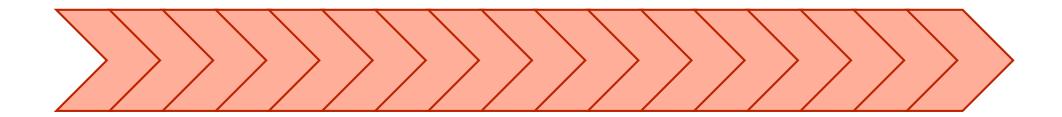
Full sequential read





# **Event Store operations**

Read aggregate's events





1. Define which routing patterns to apply

2. Choose technology/protocol accordingly

















HTTP/2



## 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

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



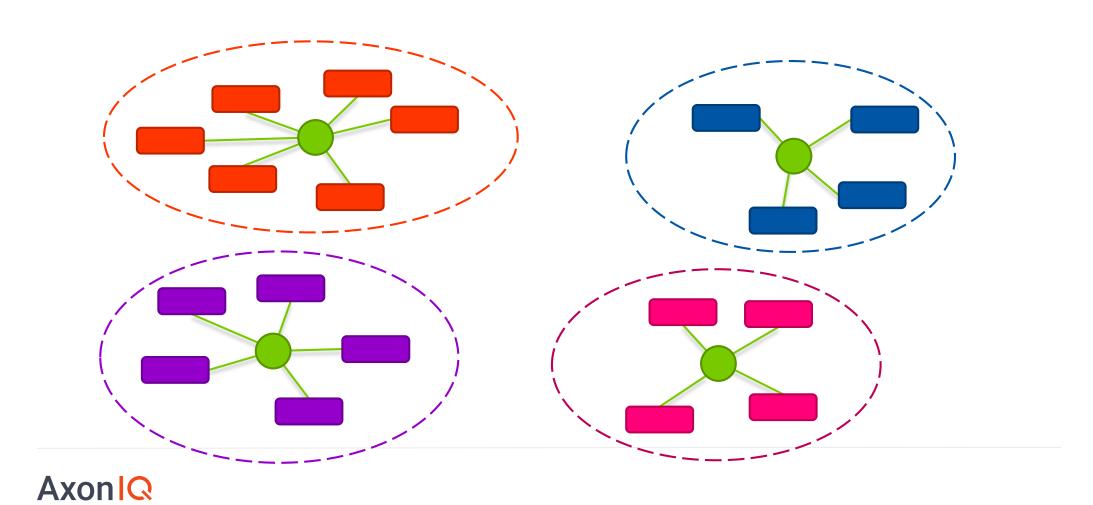


#### Bounded context

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



# Within a context, share 'everything'



Between contexts, share 'consciously' As shipping module, I want to know when an order is placed Order Created + Item Added + Order Confirmed → Order Placed **Order Created** Item Added to Order Shipping Address Added Billing Address Added Order Confirmed



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

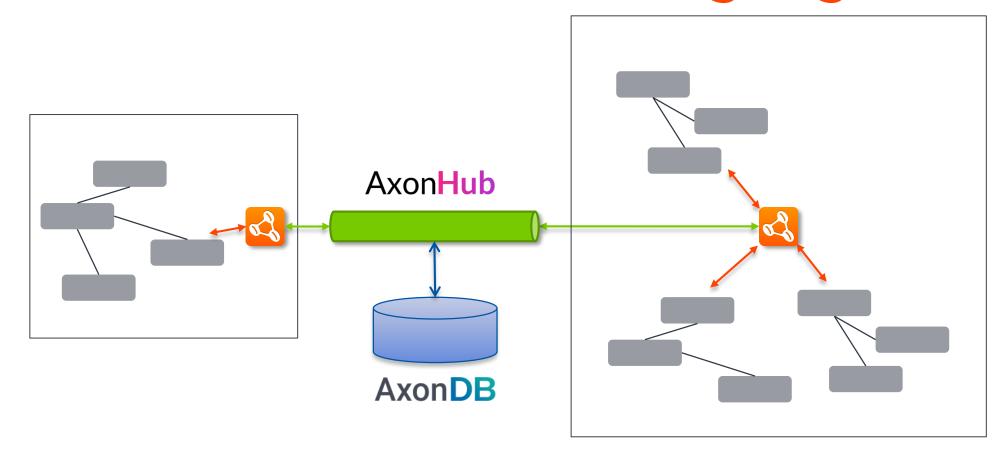


#### Where does AxonFramework fit?

- Inside each component in bounded context
- Axon provides the Java APIs towards platform
  - EventBus, CommandBus, QueryBus
- Separation of business logic and infrastructure logic



## Microservices Messaging





## "Just enough" intelligence

#### **AxonHub**

#### **Message Broker**

Sends messages. Main value add is reliability.

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



#### Our mission

Provide the APIs and implementations necessary for eventdriven microservices to cooperate harmoniously, allowing each of them to focus on the business logic.

