# Pragmatic Microservices with DDD, CQRS and Event Sourcing using Axon Framework



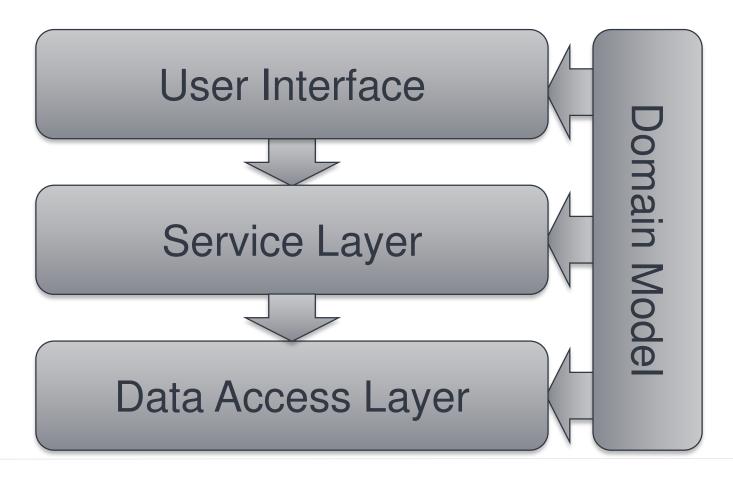
Allard Buijze
Founder & CTO, AxonIQ
Creator of AxonFramework

☑ allard@axoniq.io

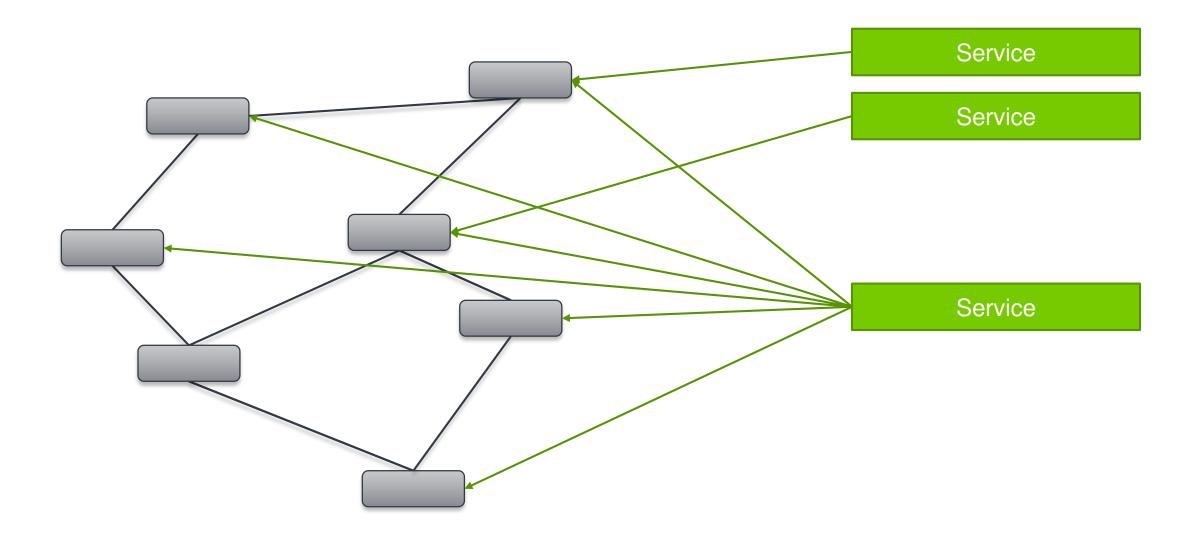
**y** @allardbz



### Layered architecture









#### 'Normal' SQL QUERY

.phob...of\_288\_284.8cos\_10. \*phob...of\_288\_284.8cos .phob...16\_288\_284.8cos\_10. \*pt. 18\_288\_284.8cos \*pt. 18\_288\_284.8cos\_10. \*phob...of\_062\_10628\_28

'hpop'.'06\_FME\_INNER\_ENV\_BNOR\_ED' AS '06\_FME\_IN'
'bp'.'RELATIONSHEP\_[D']) LEFT DOES '08EANTEAN'
'08GGANTHANTOSS' 'spp' OS ((('spp' 'OBLANTEAN'
'C'spoid'.'PRIDGIST\_FILS' = 13 AND ('spoid'.'090\_3'
('spoid'.'PRIDGIST\_FILS' = 13 AND ('spoid'.'090\_5')
('spoid'.'PRIDGIST\_FILS' = 13 AND ('spoid'.'090\_5')
('moid'.'PRIDGIST\_FILS' = 13 AND ('smoid'.'090\_5')
((('daid'.'PRIDGIST\_FILS' = 13 AND ('smoid'.'090\_5'))
((('daid'.'PRIDGIST\_ANT) 'D' = 'spoid'.'090\_5')
('moid'.'PRIDGIST\_ANT)
'(('daid'.'PRIDGIST\_ANT) 'D' = 'spoid'.'090\_5')
'(('daid'.'PRIDGIST\_ANT) 'D' = 'spoid'.''090\_5')

22 JOINS

BE ID. ABON DIN (
DAME ON THE OR OF THE OR OF

AS '15 THE DAY SCAN

"IS THE DIMEN IN

#### 6 SUBQUERIES

. .ees. . OROWRITATION ID. ) FAB.

[TO. = .ee] OR ((.ee] . WHINDING LAME . .ee.

INDIR BLAND ID.))

THE BLAND ID.)

THE BLAND ID.)

THE BLAND ID.)

OR THE ENV NOVE DEAL ..

mbe, ... coedwateration it.; yes ()))) FELA Yolk .ced\_Like yeac, .show, ce

N THE TOTAL ORG IDENTIFIES THE FROM TO THE TOTAL OR THE T

P THE EWY IMOR COAL' AS 'IS THE EWY INCH COAL'.

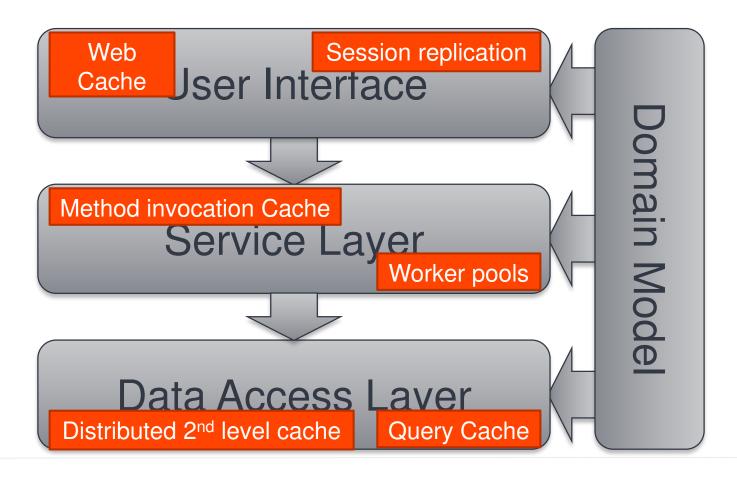
'BIM\_GOG\_IDENTIFIES\_TYPE' MERGE ('DOM\_GOG\_IDENTIFIES\_TYPE', 'MADE' = '1908'))))) LEFT JOIN 'GOG\_IDENTIFIES\_TYPE\_ID' = '1908')))) LEFT JOIN 'GOG\_IDENTIFIES\_TYPE, 'SAME ' "1908')))) LEFT JOIN 'GOG\_IDENTIFIES\_TYPE, ID' = '1908', 'GOG\_IDENTIFIES\_TYPE

. "FRIMARY FLAG





#### Layered architecture





















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





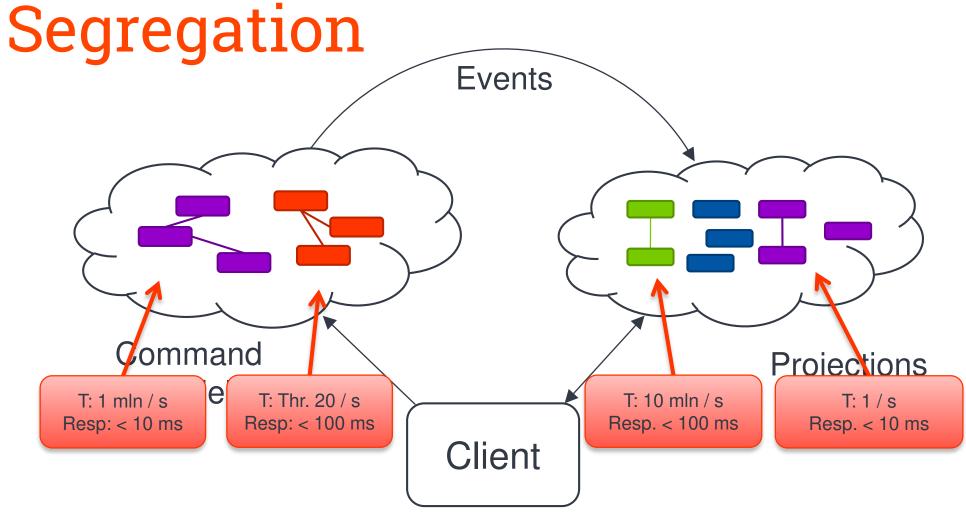








Command Query Responsibility





#### 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**

#### **State storage**

- id: 123
- items
  - 1x Deluxe Chair € 399
- status: return shipment rcvd

#### **Event Sourcing**

- OrderCreated (id: 123)
- ItemAdded (2x Deluxe Chair, €399)
- ItemRemoved (1x Deluxe Chair, €399)
- OrderConfirmed
- OrderShipped
- OrderCancelledByUser
- ReturnShipmentReceived





#### The value of Event Sourcing

- Audit trail
- Analytics
- Improve modeling
- Given-when-testing

•





#### The power of...

## Not now





Live coding...

## **Event Sourcing**

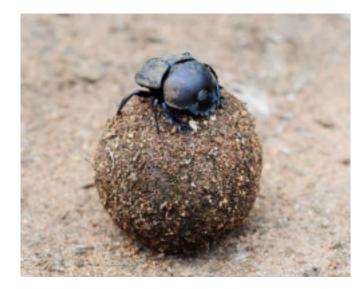




#### Monoliths







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.

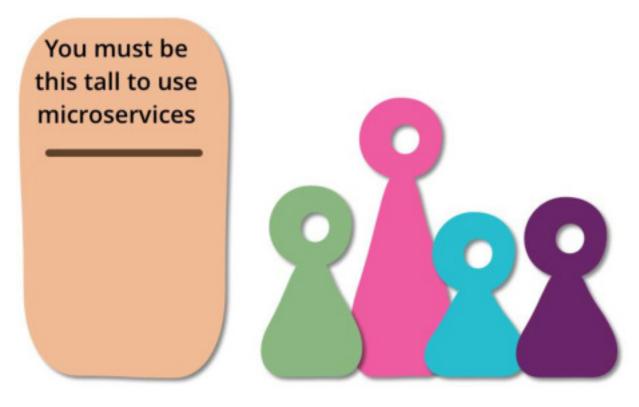
Martin Fowler

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





## Are you tall enough?



Source: martinfowler.com/bliki/MicroservicePrerequisites.html





## "Noun Driven Design"

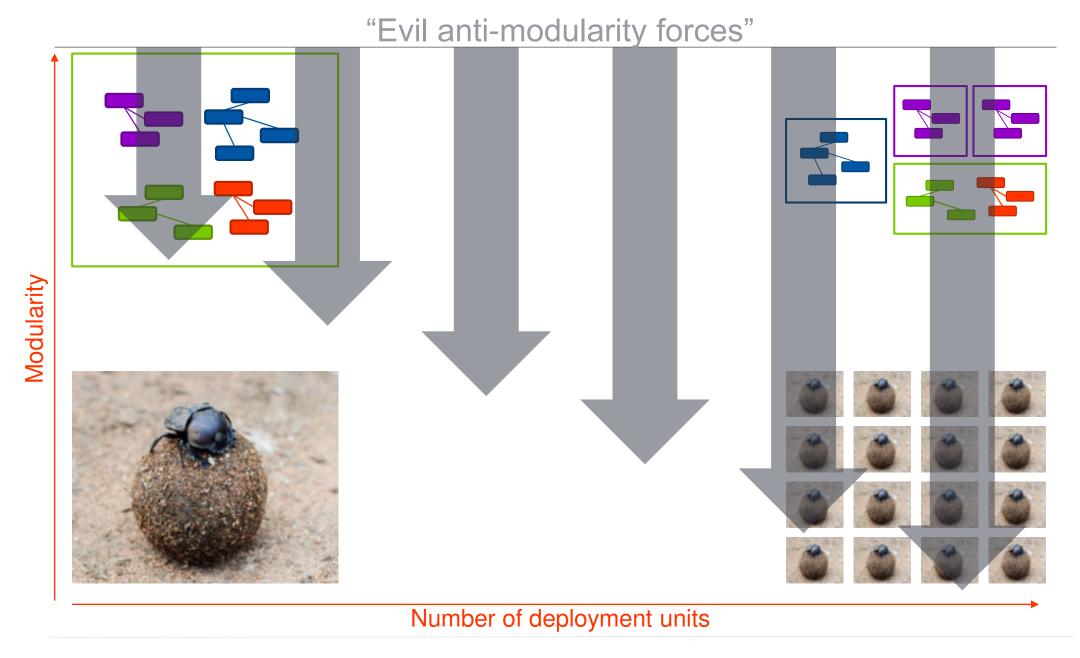




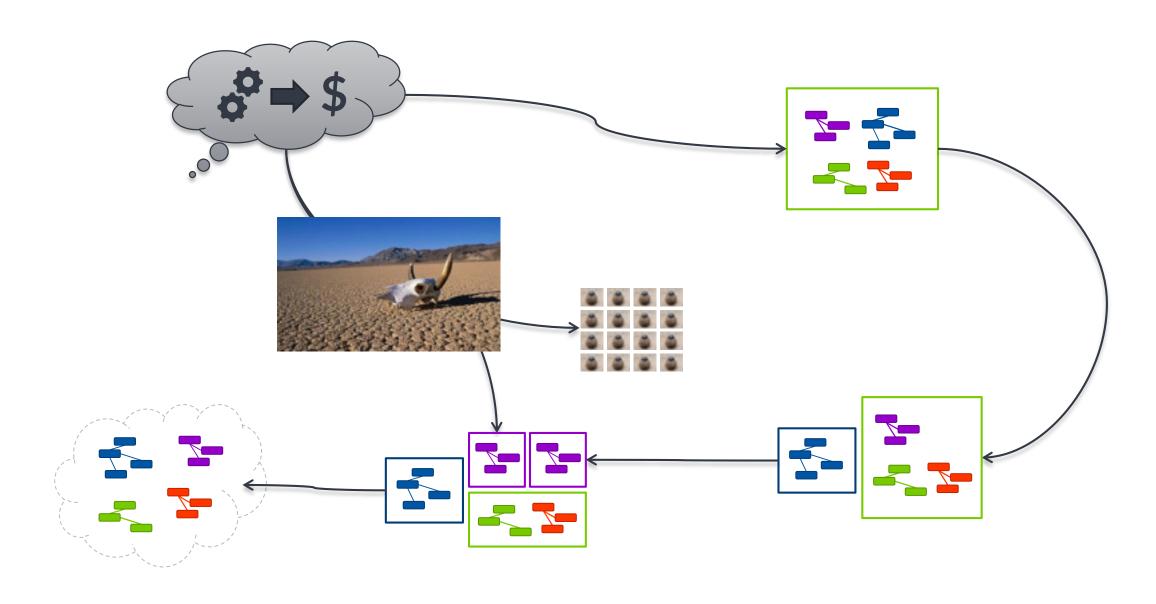
## "Entity Services"





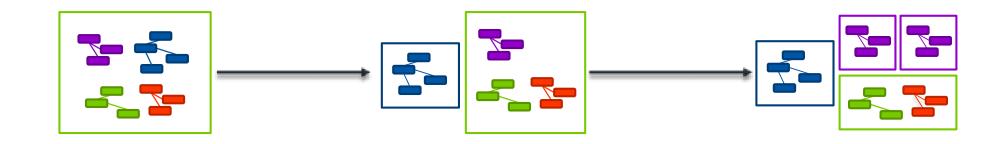








#### Location transparency



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

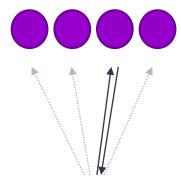
Location transparency starts with good API design (but doesn't end there)





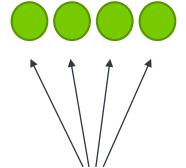
### Microservices Messaging

#### Commands



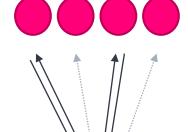
Route to single handler
Use consistent hashing
Provides confirmation/result

**Events** 



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

Queries



Route with load balancing Sometimes scatter/gather Provides result

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

Live coding...

## Distributing components





#### Rule #1 of distributed computing:

## Don't





### Try it yourself

- Sources:
  - https://github.com/abuijze/bike-rental-demo ('devdays' branch)
  - https://github.com/AxonFramework/AxonFramework
- Download Axon
  - axoniq.io/download (includes quick-start-guide)



