

# Harness the power of Event-Driven Systems

without burning your hands (or budgets)



Allard Buijze

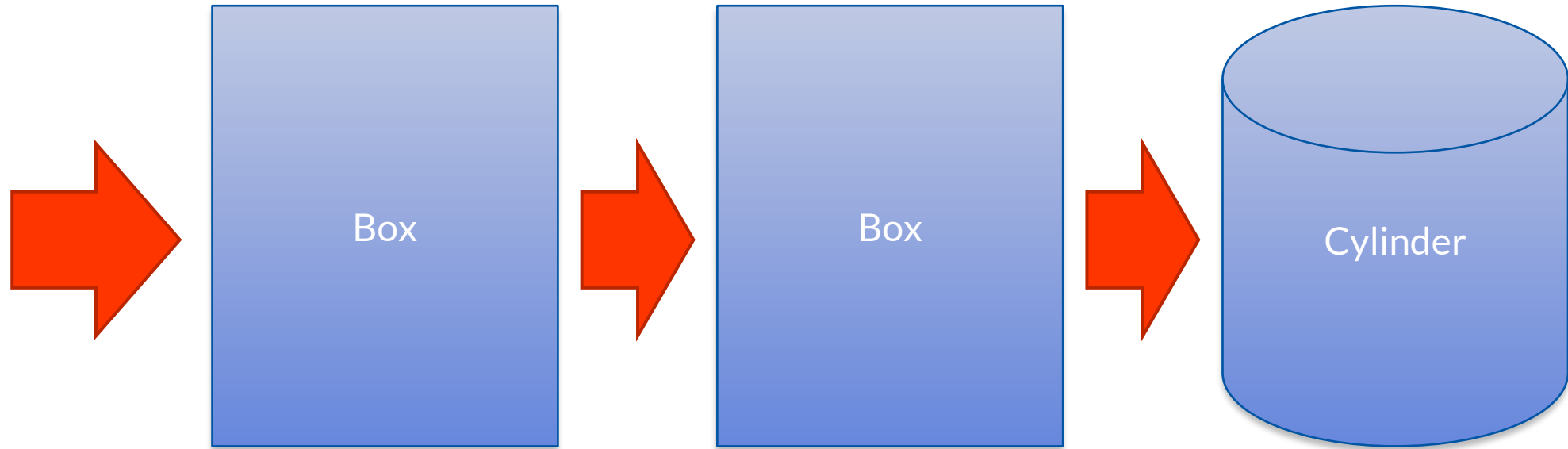
Founder & CTO, AxonIQ

✉ [allard@axoniq.io](mailto:allard@axoniq.io)

🐦 [allardbz](https://twitter.com/allardbz)

# Once upon a time...

# “Universal ‘BBC’ architecture”



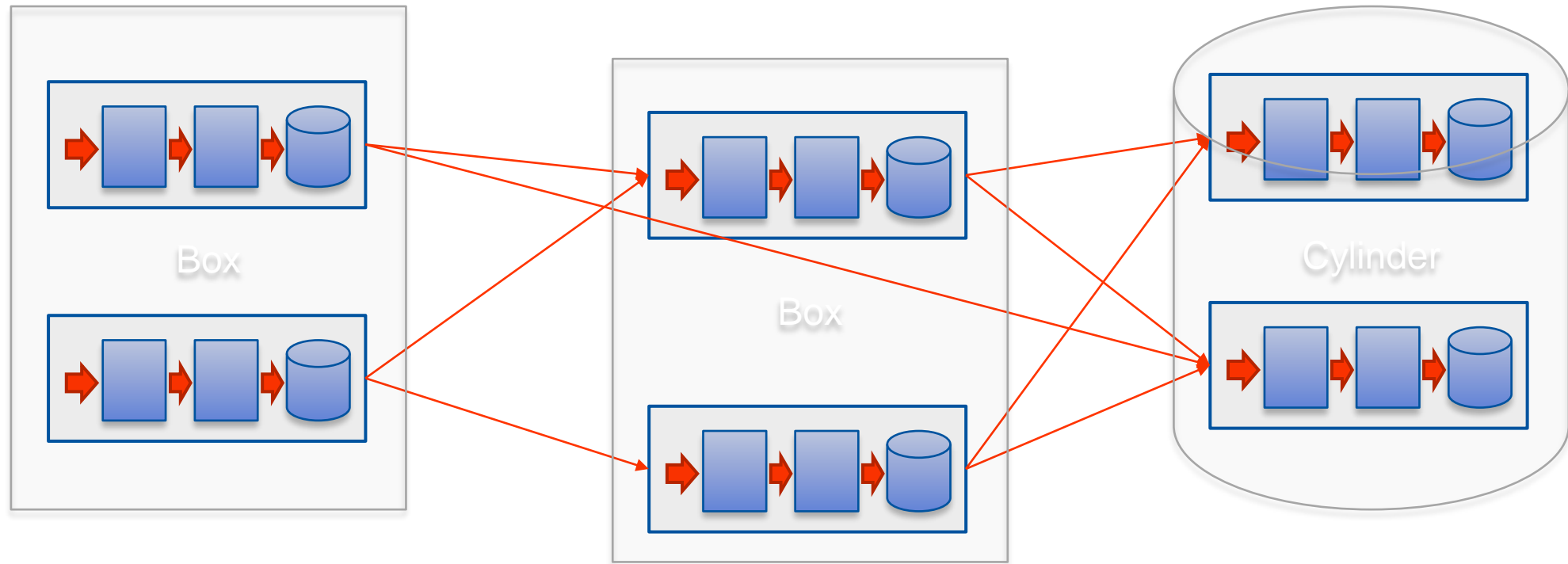
Source: Ted Neward



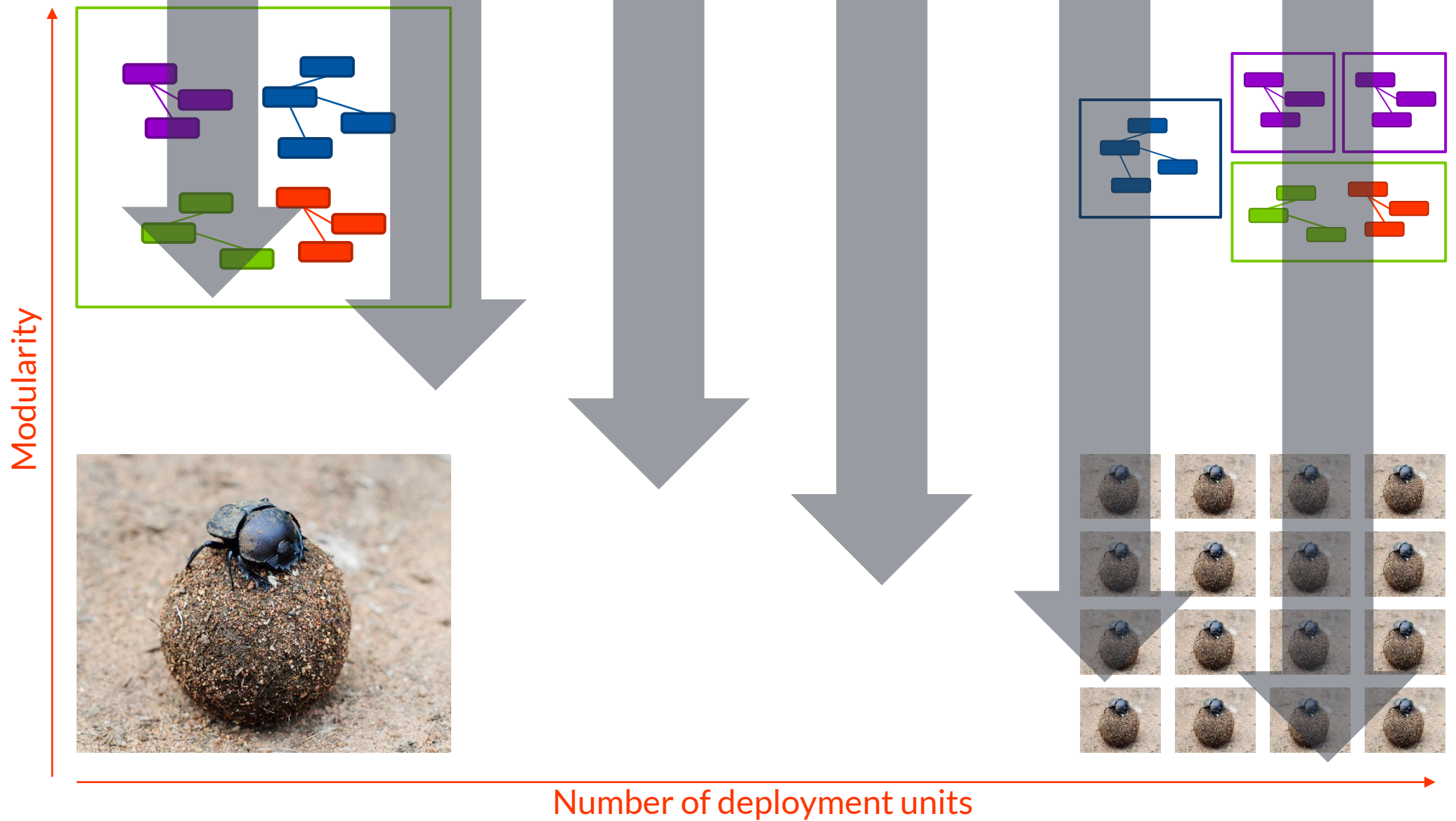
Source: <http://www.sabisabi.com/images/DungBeetle-on-dung.JPG>

# Microservices to the rescue!

# “Universal Microservices architecture”

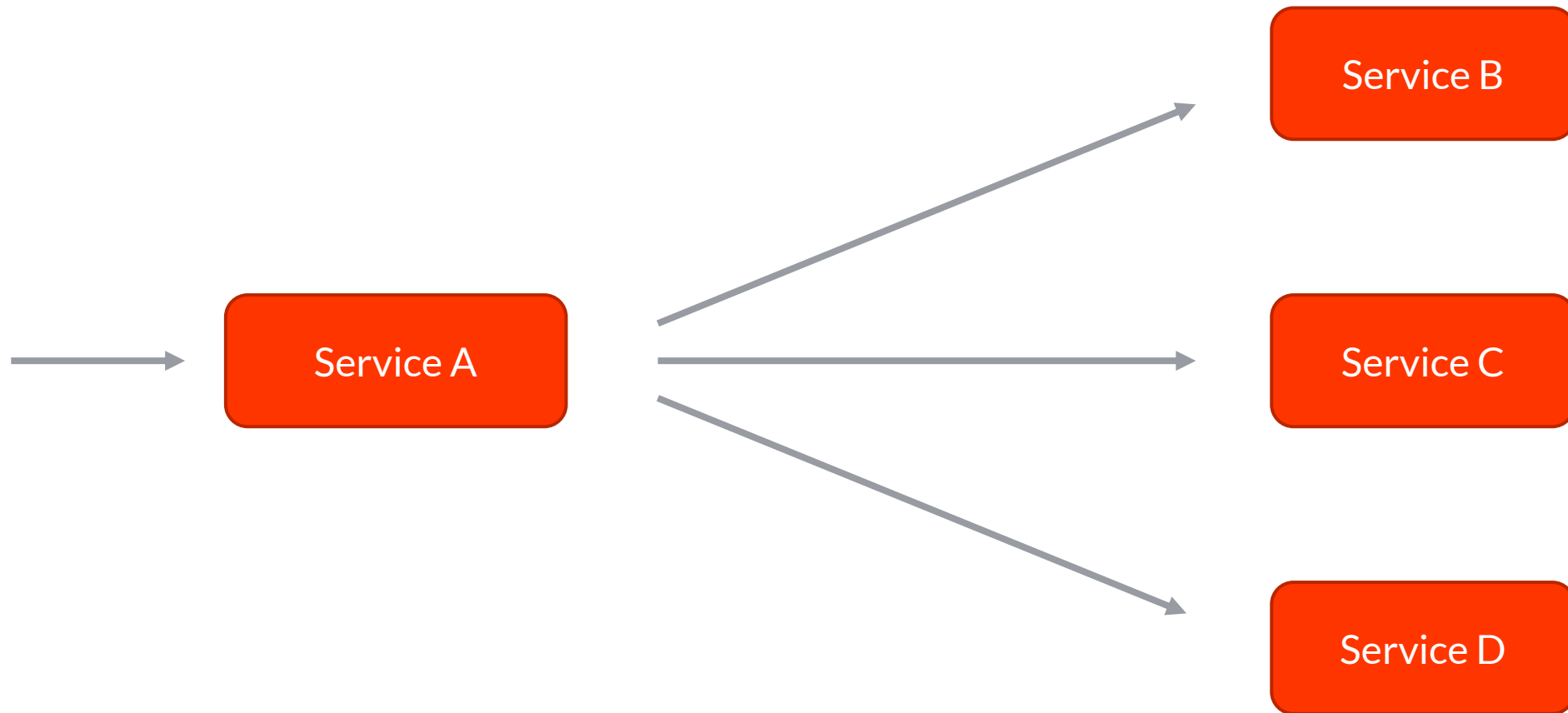


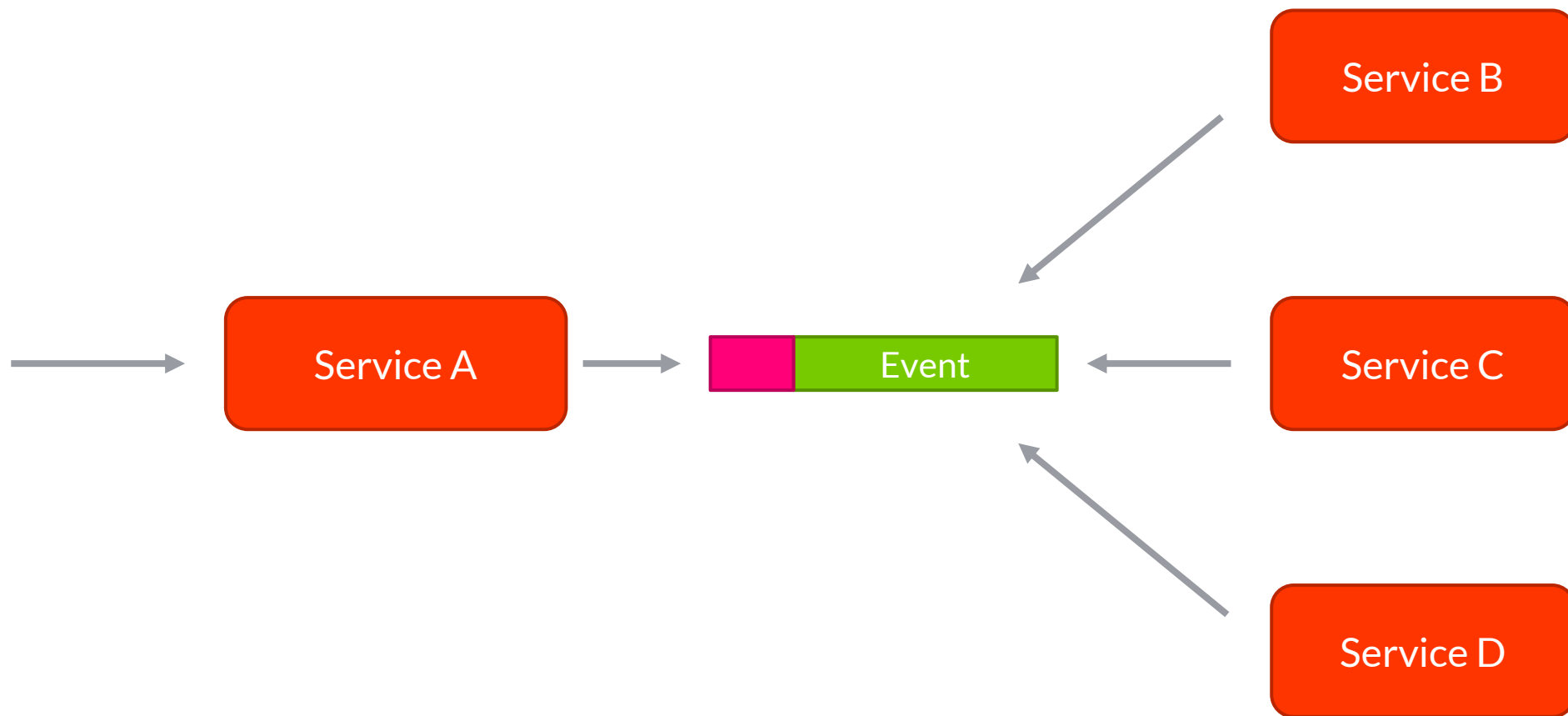
# “Anti-modularity forces”



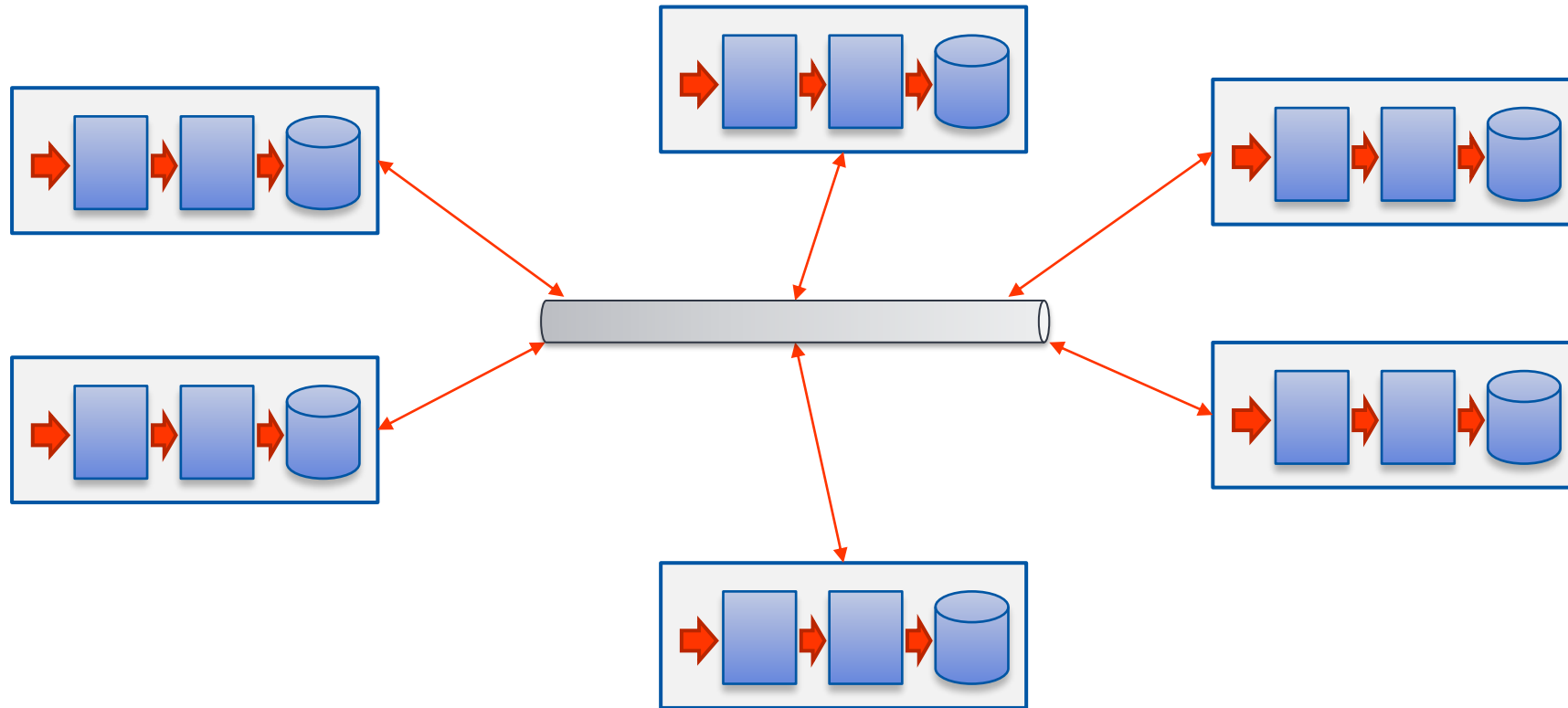
# Events to the rescue!







# “Event-Driven Microservices architecture”



# The promise

- Cloud-Native / Reactive System
- Location transparency
- Evolutionary architecture
- Reusable event streams









# Events as side-effect

- How do you guarantee (eventual) consistency across sub-systems?
- Any afterthought will increase the complexity of a system

# Events & stored state

## What happened

OrderCreated (id: 123)

## What we store

id: 123

status: awaiting confirmation



# Events & stored state

## What happened

	OrderCreated	(id: 123)
	ItemAdded	(2x Deluxe Chair, €399)

## What we store

id: 123

Items:

- 2x Deluxe Chair - € 798

status: awaiting confirmation

# Events & stored state

## What happened

	OrderCreated	(id: 123)
	ItemAdded	(2x Deluxe Chair, €399)
	ItemRemoved	(1x Deluxe Chair, €399)

## What we store

id: 123

Items:

- 1x Deluxe Chair - € 399

status: awaiting confirmation

# Events & stored state

## What happened

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

## What we store

id: 123

Items:

- 1x Deluxe Chair - € 399

status: confirmed by client

# Events & stored state

## What happened

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

## What we store

id: 123

Items:

- 1x Deluxe Chair - € 399

status: shipped

# Events & stored state

## What happened

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

## What we store

id: 123

Items:

- 1x Deluxe Chair - € 399

status: shipped & cancelled

# Events & stored state

## What happened

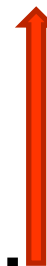
OrderCreated	(id: 123)
ItemAdded	(2x Deluxe Chair, €399)
ItemRemoved	(1x Deluxe Chair, €399)
OrderConfirmed	
OrderShipped	
OrderCancelledByUser	
ReturnShipmentReceived	



Value is here. Not here.

## What we store

id: 123
Items:
- 1x Deluxe Chair, €399
status: return shipment received



# Event Sourcing

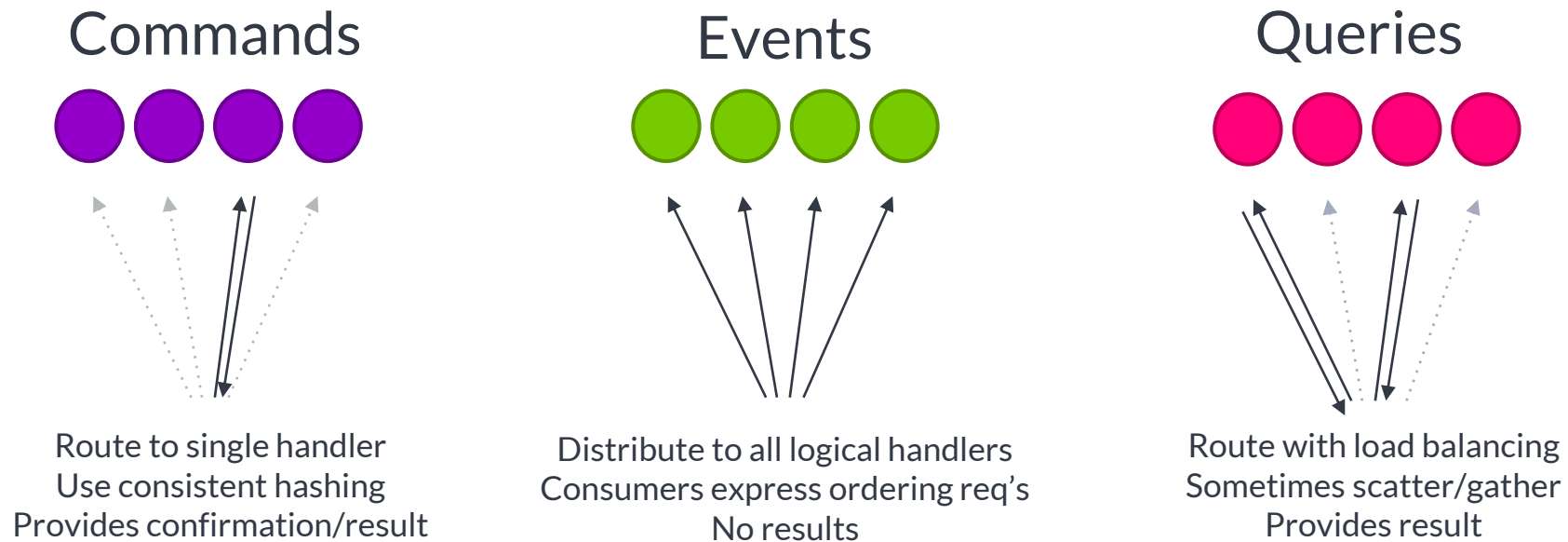
the truth,  
the whole truth,  
nothing but the truth

# Awkward coupling

- Events invert and obfuscate dependencies
- Communication can't always be in past tense

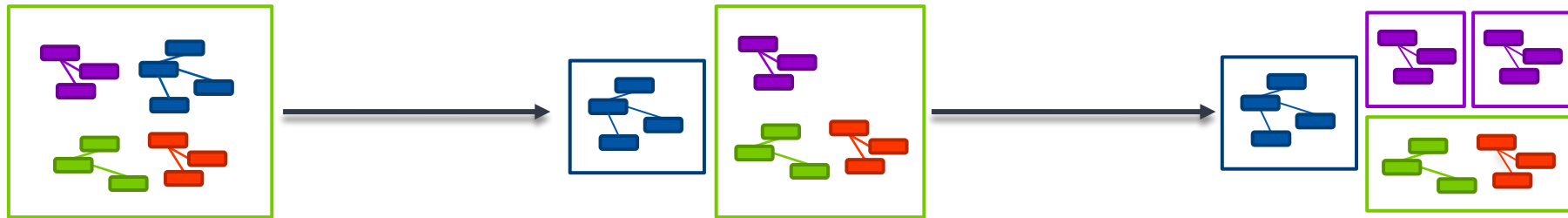


# Microservices Messaging

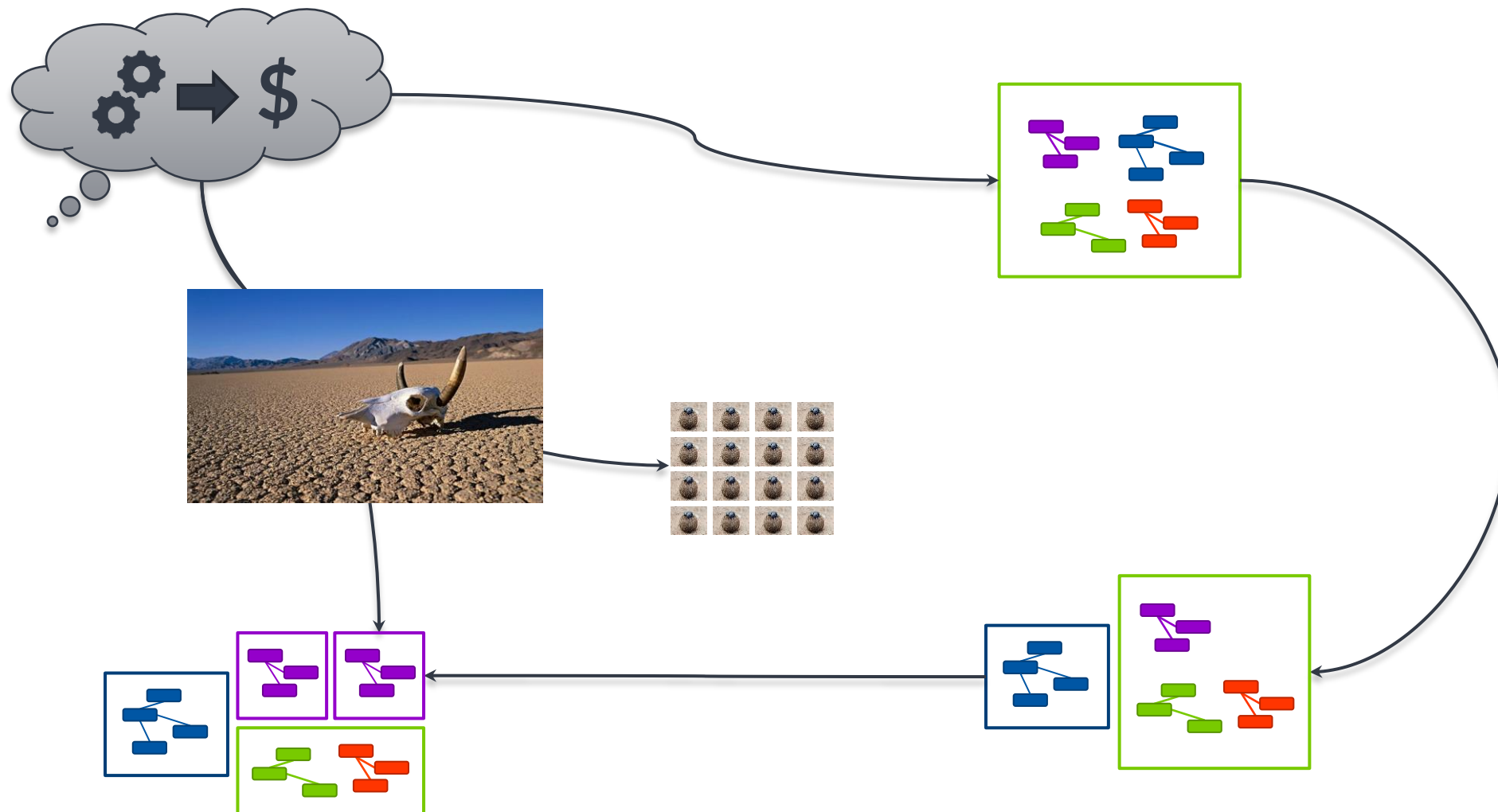


"Event" and "Message" is not the same thing

# Location transparency



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



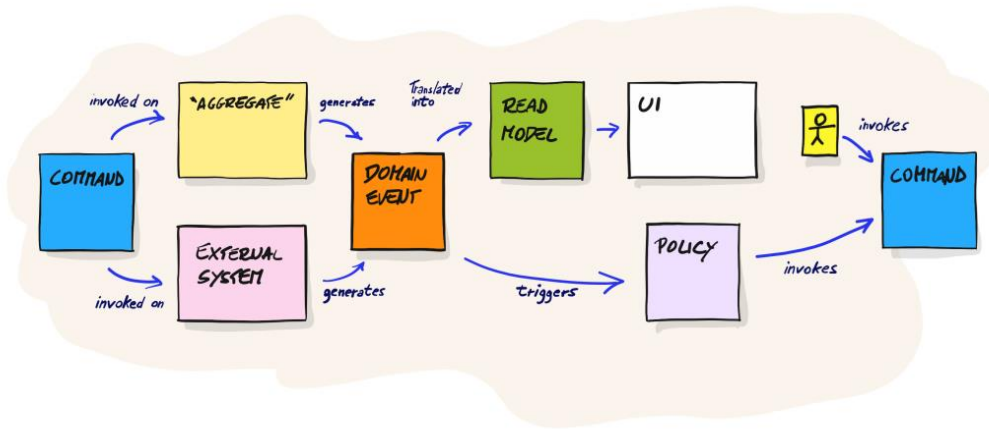
# Technical complexity

- Delivery guarantees
- Timeliness
- Error handling / exceptional cases
- Evolution of message structure
- Trigger replays of historic events

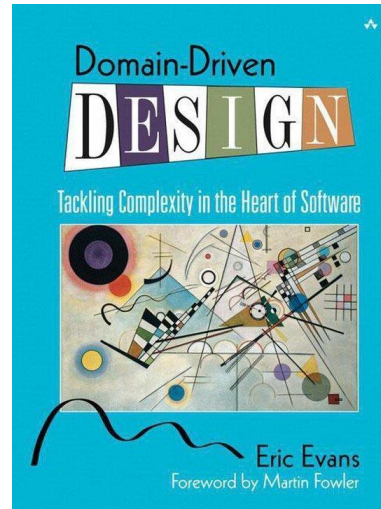




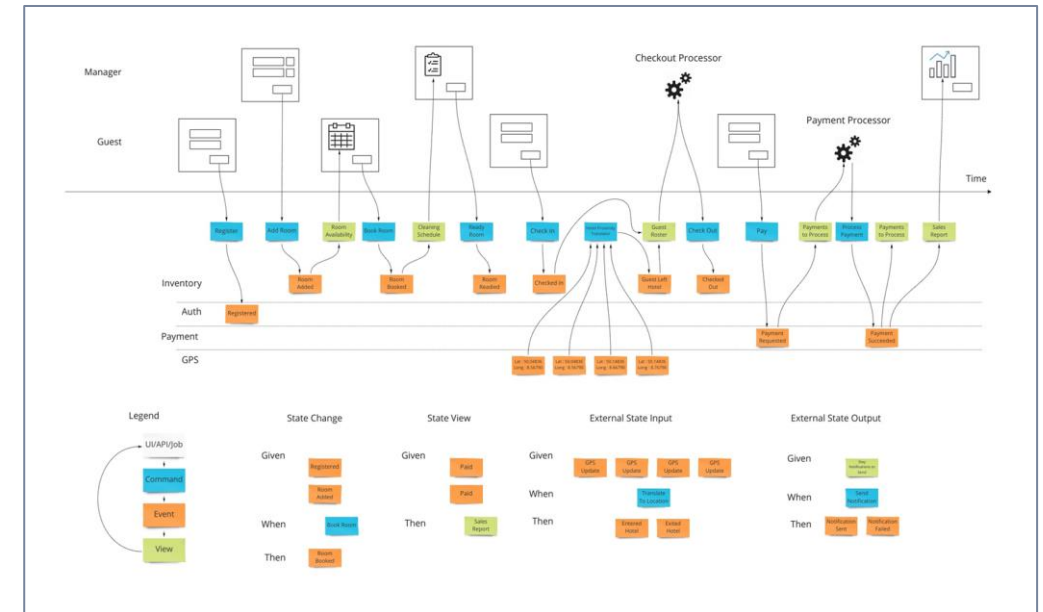
# Event Storming



# Domain-Driven Design



# Event Modelling



# In closing....



# Beware coupling

Consider *commands*  
and *queries*  
as much as *events*

# “Microservice Journey”

# Monolith first

Wax on, wax-off!

# More information...



[axoniq.io](https://axoniq.io)

[github.com/AxonFramework](https://github.com/AxonFramework)

[github.com/AxonIQ](https://github.com/AxonIQ)



[eventmodeling.org](https://eventmodeling.org)

[eventstorming.com](https://eventstorming.com)



[linkedin.com/in/abuijze](https://linkedin.com/in/abuijze)

[allard@axoniq.io](mailto:allard@axoniq.io)