A photograph of a modern, open-plan office space. In the foreground, a man is sitting at a desk, working on a laptop. The desk is equipped with a green ergonomic chair. In the background, there are more desks, some with blue and green partitions, and other people working. The office has a high ceiling with exposed pipes and modern lighting fixtures. A large green graphic overlay covers the bottom right portion of the image, containing the text "eventsourcing" and "Axon Framework".

# eventsourcing Axon Framework

# I'm Andi Pangeran



**Currently :**



**GOPAY - Backend Developer**

**Prev :**



*Jurnal.id - Senior Software Engineer*



*J-travel - Tech Lead*

**Community :**



@apangeran



@a\_pangeran



<https://cybercoding.wordpress.com>

# Overview

## Event-Sourcing

- hidden gem from tactical **DDD**
- **Event Sourcing** concept
- **Axon Framework**
- **Demo**

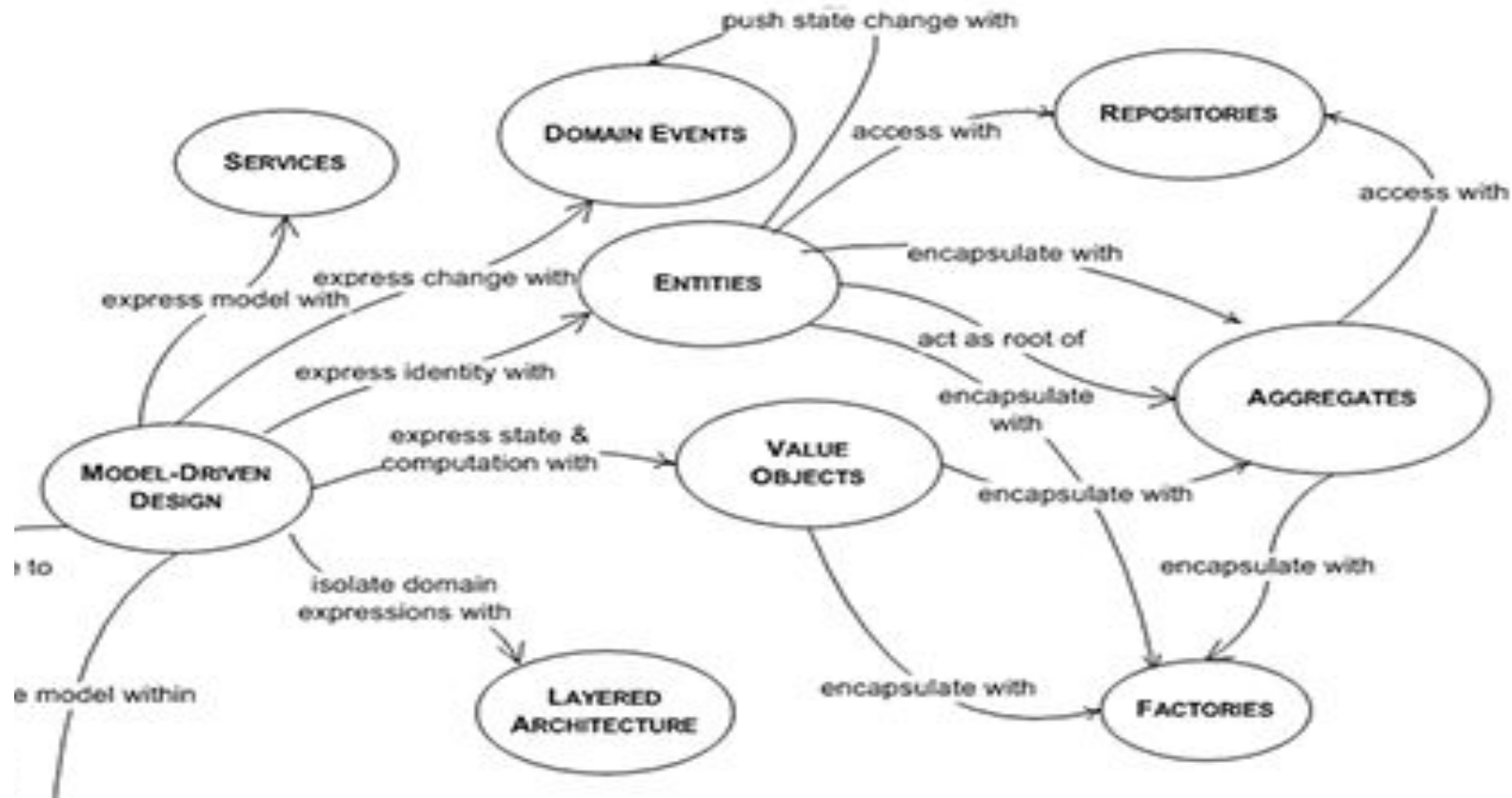
**a hidden gem of tactical DDD...**



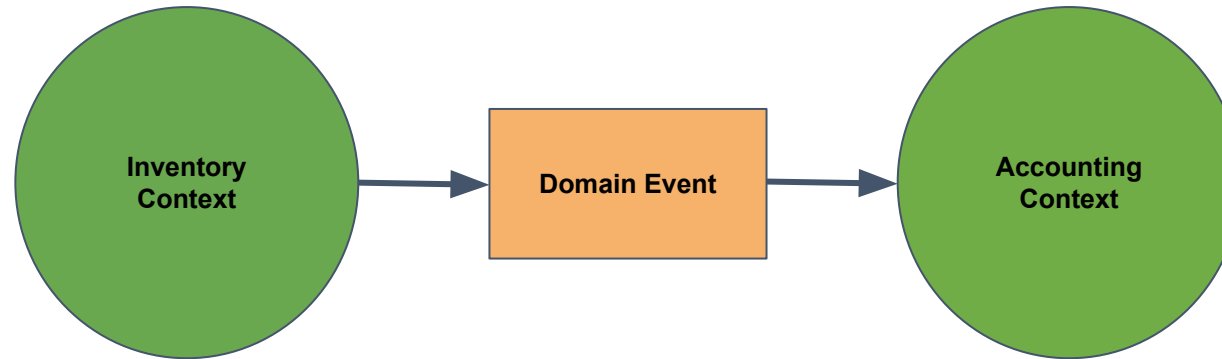
# Domain Driven Design



# Tactical Design

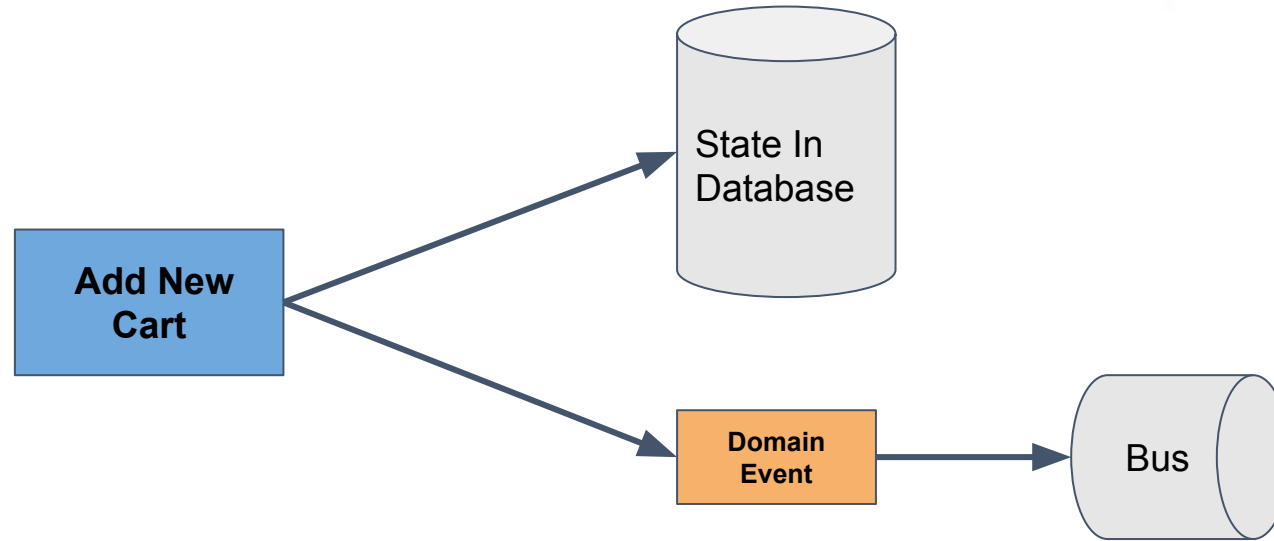


# Domain event



- something meaningful that happened in the Domain
- can be used for communicating state changes of Aggregates
- Immutable Value Object

# Consistency Issue ?



- **Dispatch** Event before **committing**: is when you dispatch an event before the database transaction is completed.
- **Commit** before **dispatching** event: is when you dispatch an event after the database transaction is completed.



**events first with event sourcing...**


**before that, let's see how non evensourced system  
looks like...**

# Store Current State

ID	IDENTITY	VENDOR	MODEL
1	CP001	LOREM IPSUM	WIDGET A
2	CP002	LOREM IPSUM	WIDGET B

INSERT INTO PRODUCTS VALUES (1, 'CP001', 'LOREM IPSUM', 'WIDGET A') 

INSERT INTO PRODUCTS VALUES (2, 'CP002', 'LOREM IPSUM', 'WIDGET B')

UPDATE PRODUCTS SET MODEL = 'WIDGET C' WHERE ID = 2 

# Store Current State

## ○ Keeping only current state.

lot of valuable historic data is lost, e.g. the information about the time a state change happened, who invoked the change, and so on. This historic data may remind you of something called an **Audit Log**.

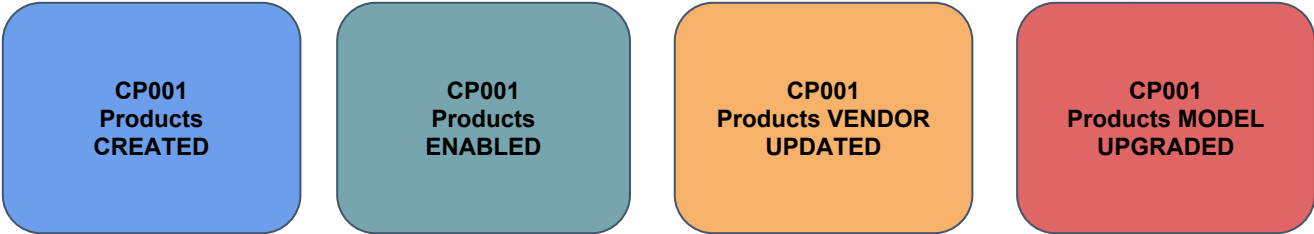
## ○ Conceptual Model.

there is no single model that could be used universally for every possible case that the system needs to address. Moreover, the structure of the application data view tends to change overtime.

# Eventsourced System ?...

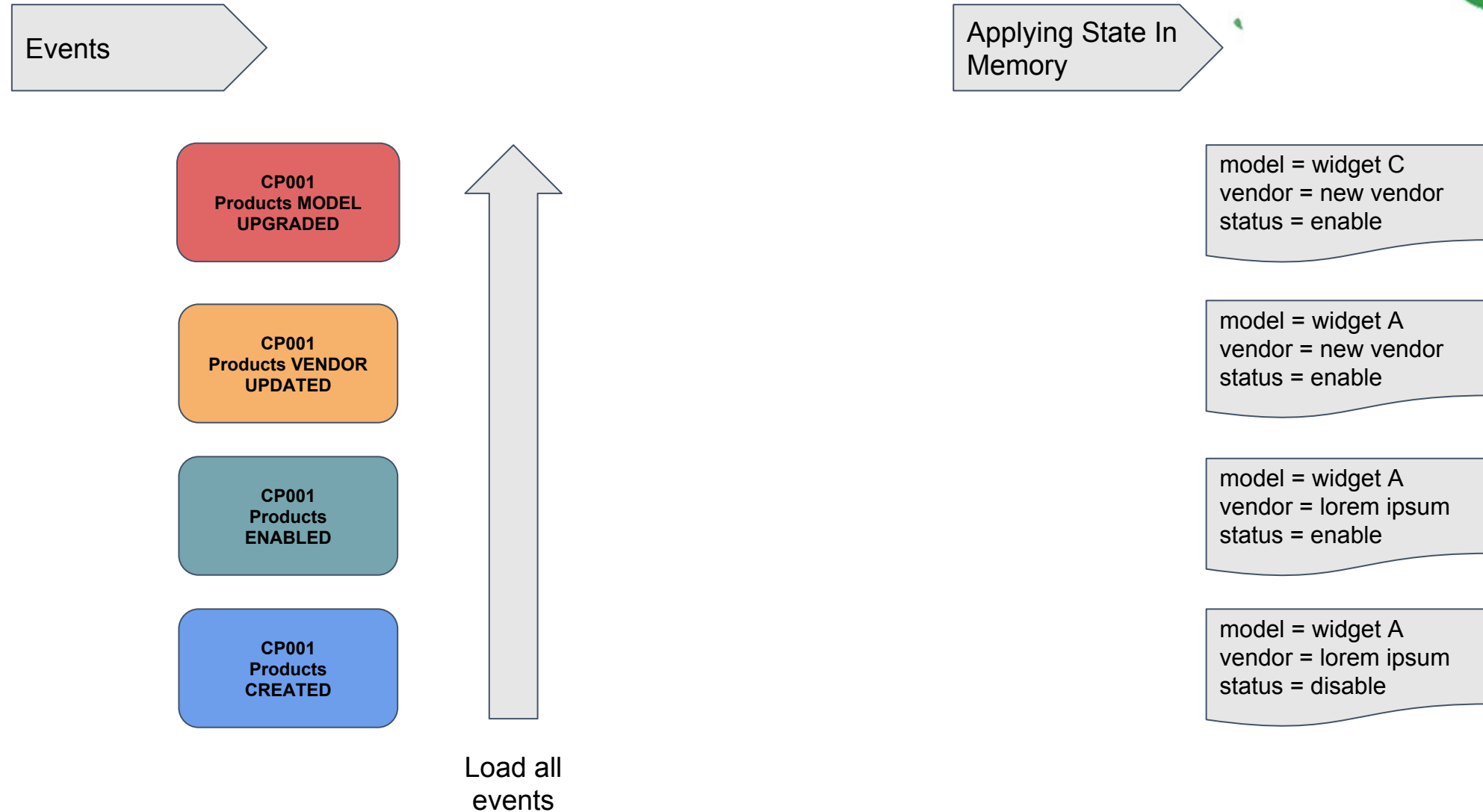


# EventLog

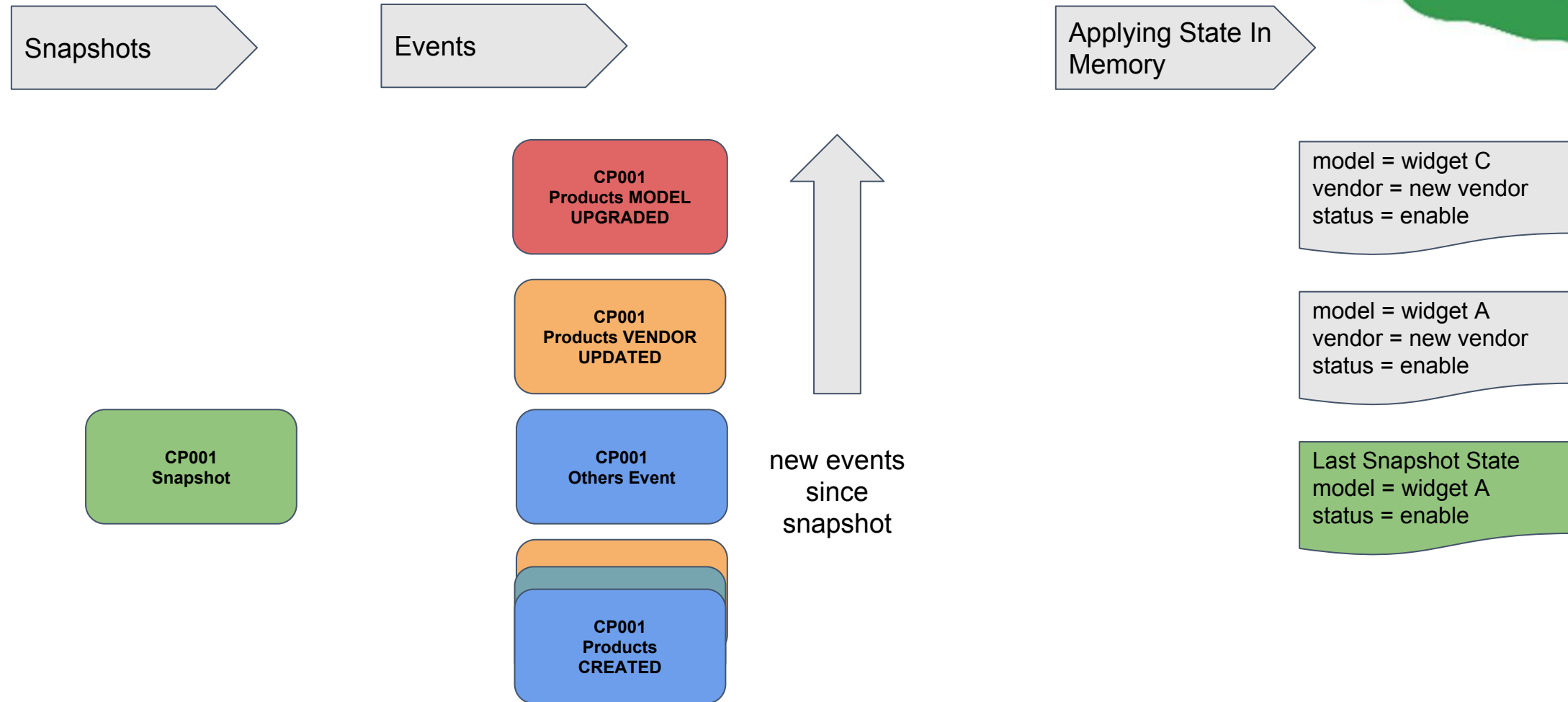


aggregateId	aggregateType	eventName	eventSequence	Payload
CP001	items	ProductsCreated	0	{}
CP001	items	ProductsEnabled	1	{}
CP001	items	ProductsVendorUpdated	2	{}
CP001	items	ProductsModelUpgraded	3	{}

# Load state via projections



# Load state with snapshot



**pros and cons event sourcing ?...**  
**let's discuss in Q&A :D**

# Event sourcing in JVM ?...



# Library / Framework



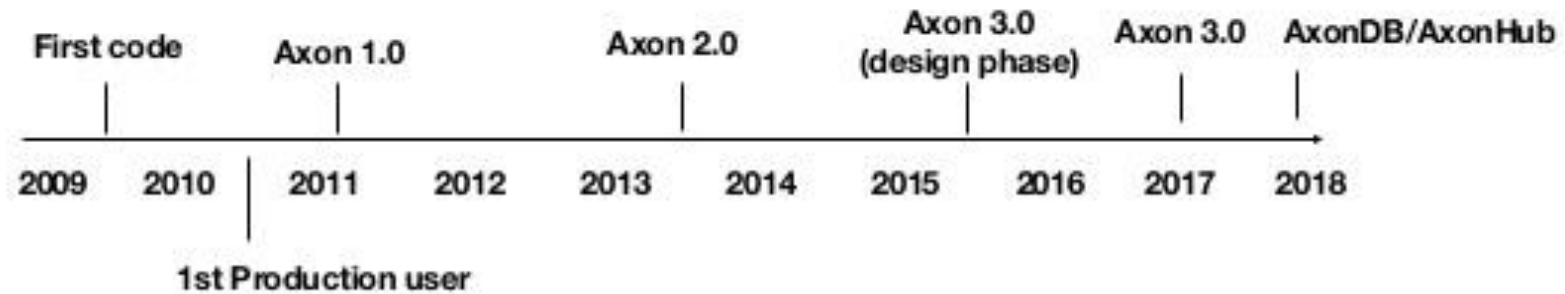
lagom

AxonFramework

# AxonFramework

<https://github.com/AxonFramework/AxonFramework>

# History



# Another Why

## Why Axon Framework?

1

Proven technology for  
DDD / event-sourcing /  
CQRS systems

2

Supports evolutionary  
approach to  
Microservices

3

Focus on business logic

4

Integrates with Spring  
Boot and other  
frameworks

5

Open source



# Axon Component

- |                                           |                                   |                                       |
|-------------------------------------------|-----------------------------------|---------------------------------------|
| <input type="radio"/> Command             | <input type="radio"/> Aggregate   | <input type="radio"/> Event Processor |
| <input type="radio"/> Command Gateway     | <input type="radio"/> Repository  | <input type="radio"/> Event Handler   |
| <input type="radio"/> Command Bus         | <input type="radio"/> Event Store | <input type="radio"/> Event Scheduler |
| <input type="radio"/> Command Interceptor | <input type="radio"/> Snapshotter | <input type="radio"/> Saga            |
| <input type="radio"/> Command Handler     | <input type="radio"/> Event       |                                       |



**since version 3,  
Axon has Spring AutoConfiguration...**



**demo...**

**<https://github.com/meong1234/simplewallet>**

a good software developer is like a  
werewolf: afraid of silver bullets.  
jochen mader

# Reference

<https://enterprisecraftsmanship.com/2017/10/03/domain-events-simple-and-reliable-solution/>

<https://martinfowler.com/eaDev/DomainEvent.html>

<https://www.slideshare.net/RafaelWinterhalter/eventsourcing-microservices-on-the-jvm-109271476>

<https://blog.arkency.com/2015/03/why-use-event-sourcing/>

<http://codebetter.com/gregyoung/2010/02/20/why-use-event-sourcing/>

<https://ookami86.github.io/event-sourcing-in-practice>

<https://axoniq.io/>



# HIDUP TANPA BATAS

GO  JEK - in aja