

What is Domain Driven Design?

Contact

lewis@banked.com

Lewis Jones— Lead Engineer

Who am I?

Lewis Jones - Lead Engineer



2012 - 2015

Gorkana/Cision



2015

Makers Code bootcamp



2016 - 2020

Simply Business



2020

Smart Pension

Banked :

2021 —

Banked

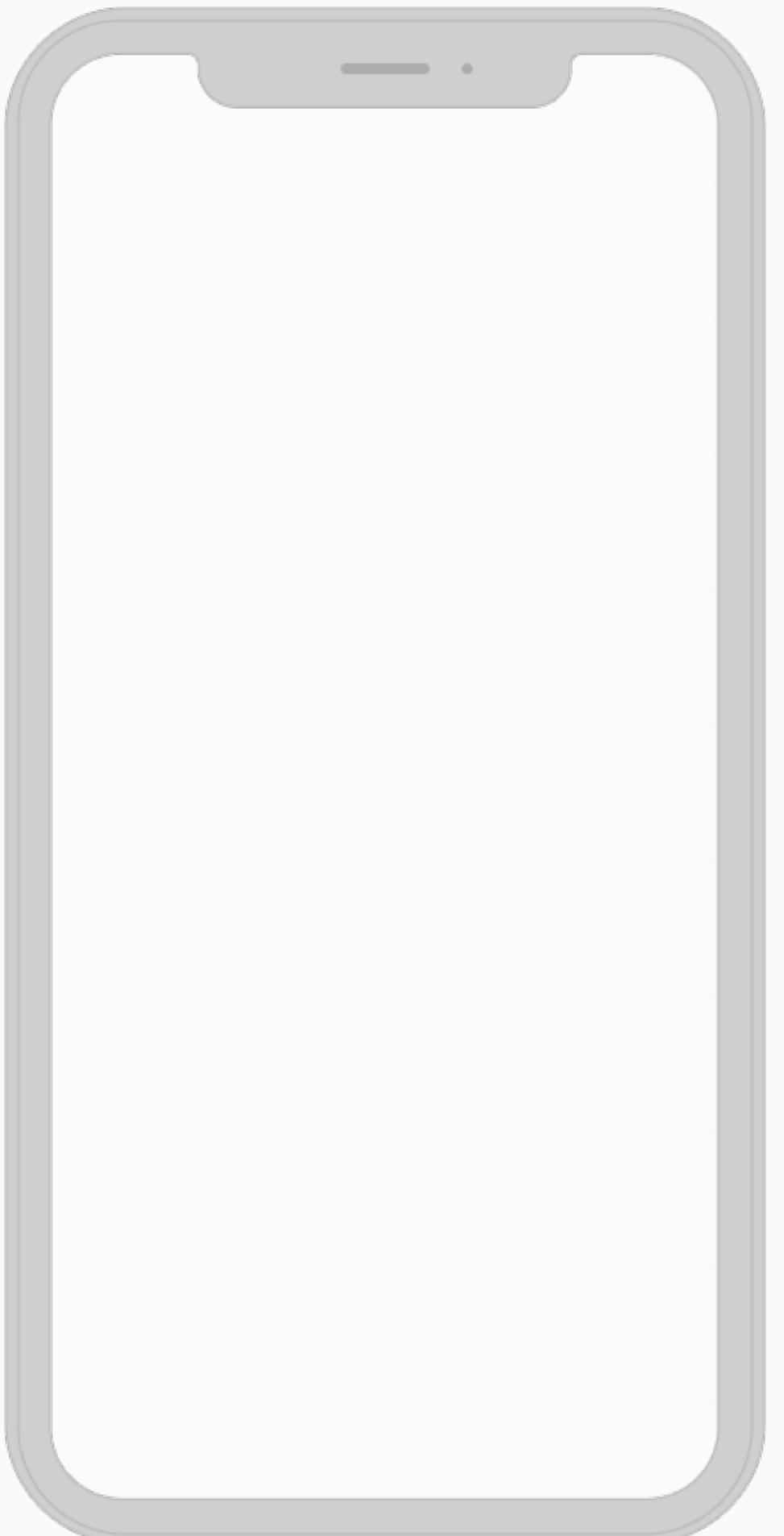
What is Banked?

A global payments network built on modern banking rails.

Banked powers real-time payments for consumers, businesses and banks, improving customer experience, payment security, business efficiency and cost effectiveness.

A better way to take and make payments.

banked.com



Domain Driven Design.

What is it?

“

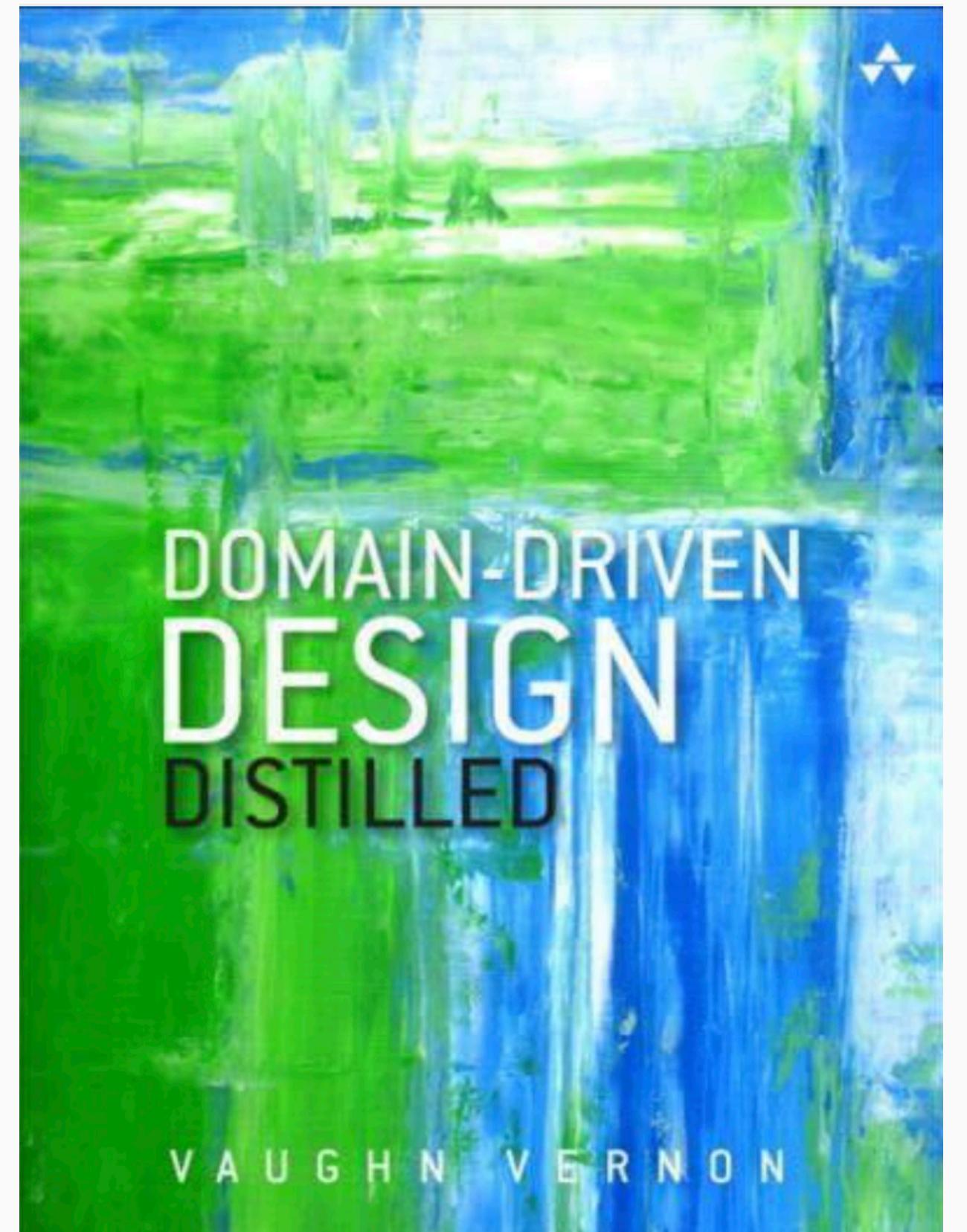
Questions about whether design is necessary or affordable are quite
beside the point: design is inevitable. The alternative to good design is
bad design, not no design at all.

DOUGLAS MARTIN

Domain Driven Design (DDD)

“ We are on a quest for deep learning about how the business works, and then to model software based on the extent of our learning”

Vaughn Vernon

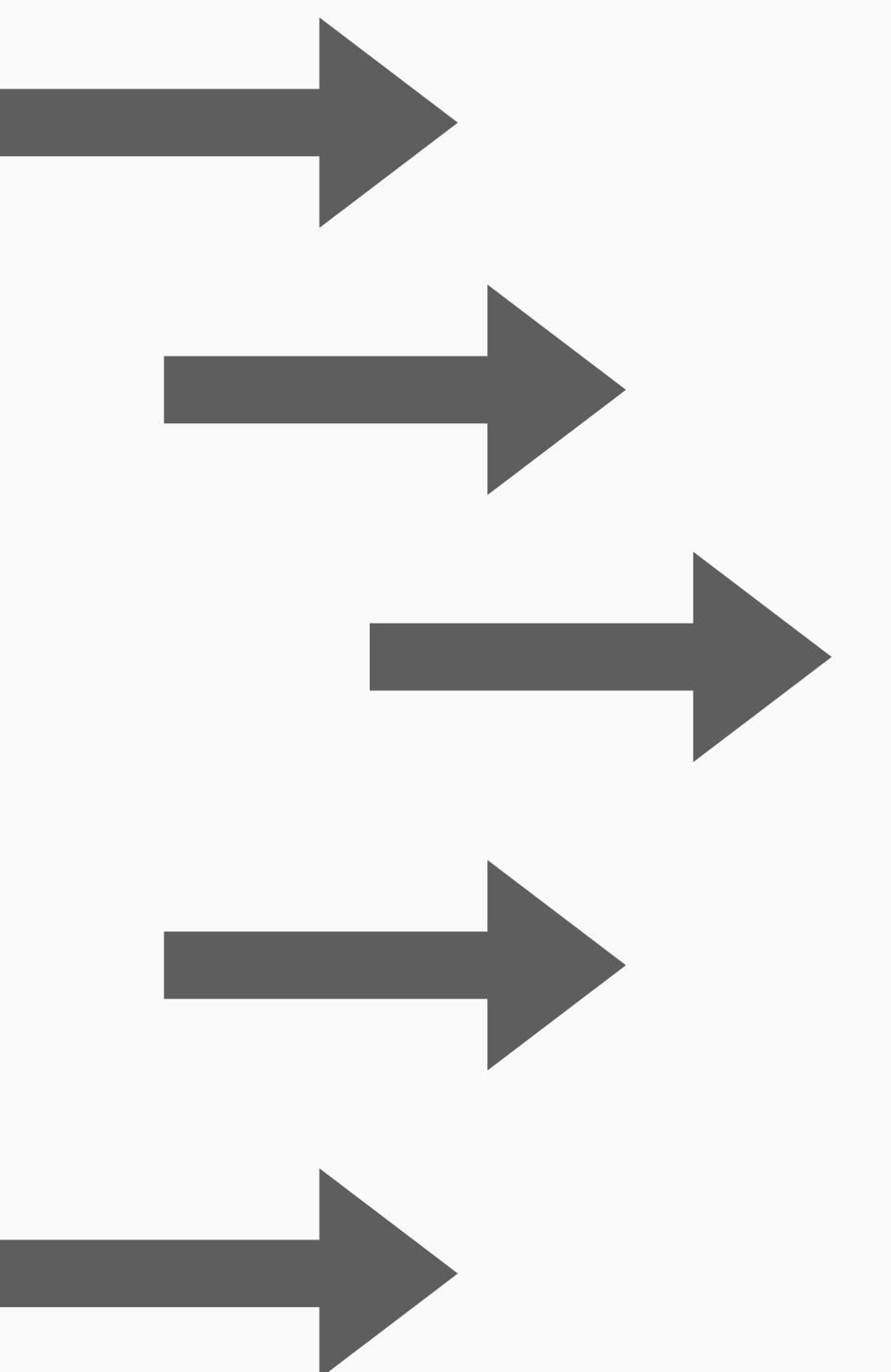
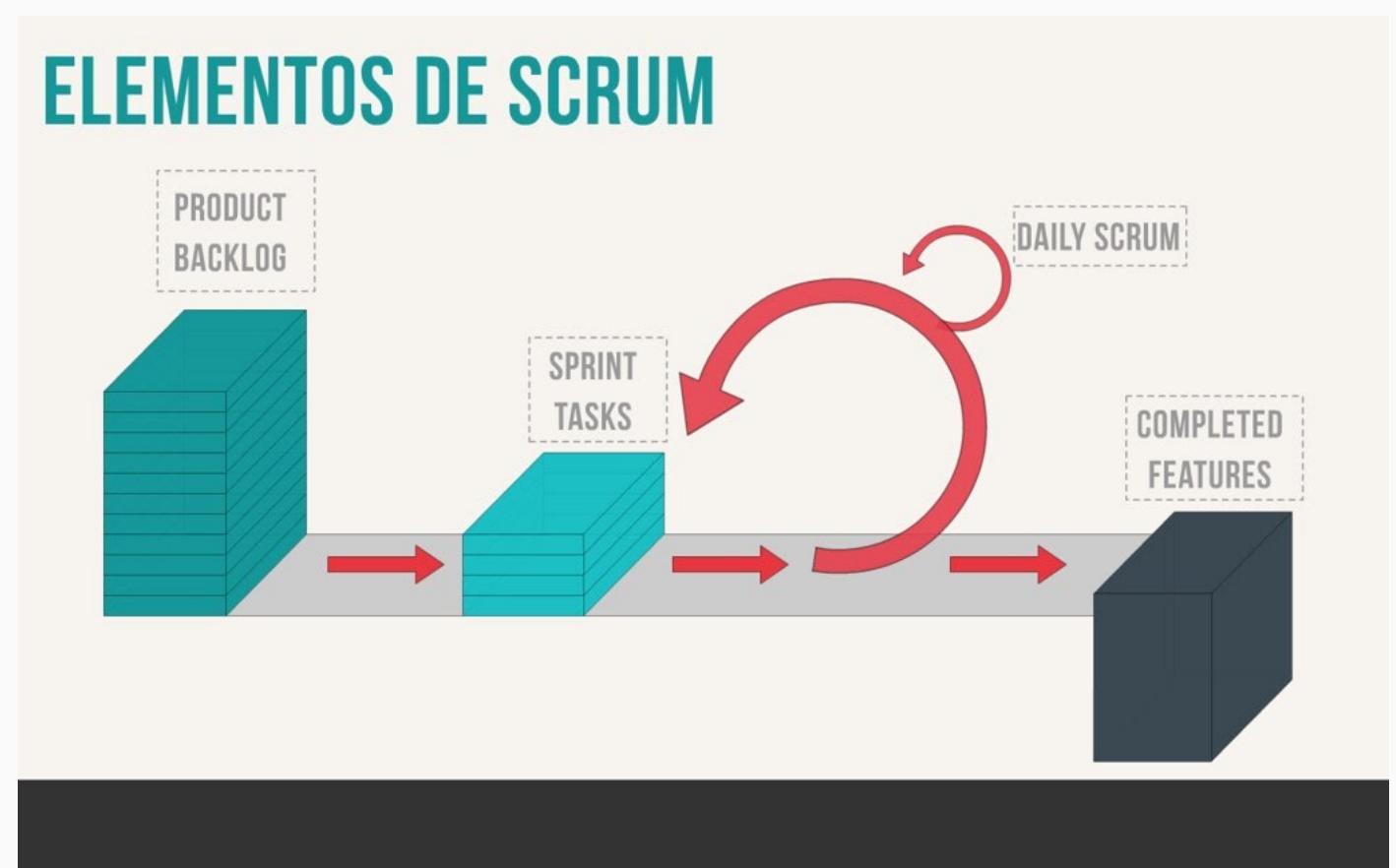


What happens without design.

What happens without design

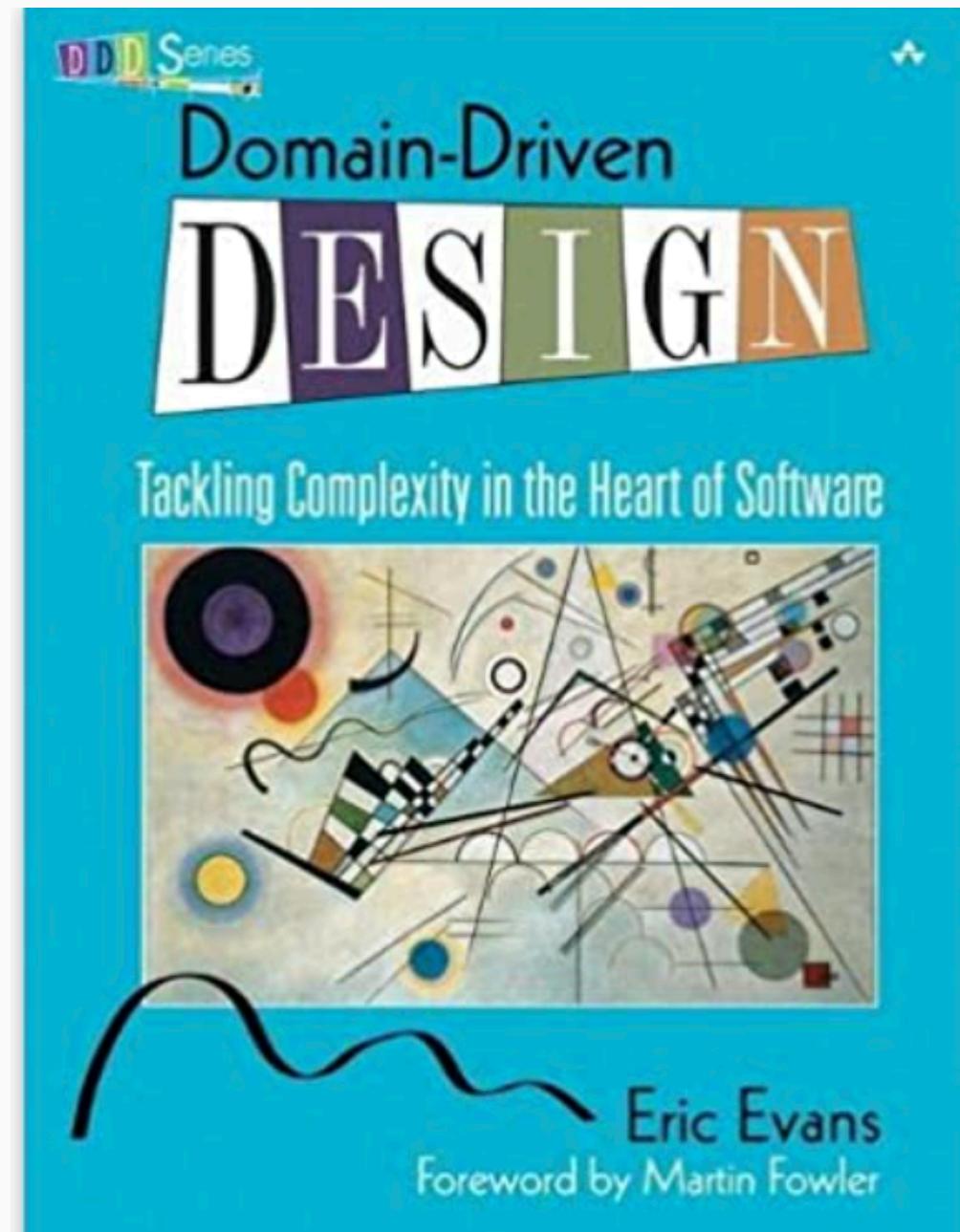
- Software development is considered a cost centre
- Developers wrapped up in technology (shiny objects)
- Database given too much priority, solutions around DB not model
- Developers don't emphasise naming objects
- Poor collaboration
- Project estimates in high demand
- Business logic is in user interface and persistence components
- Broken, slow locking DB queries
- Wrong abstractions
- Strongly coupled services

A big ball of mud



How can Domain Driven Design help?

Two Parts of DDD Strategic and Tactical



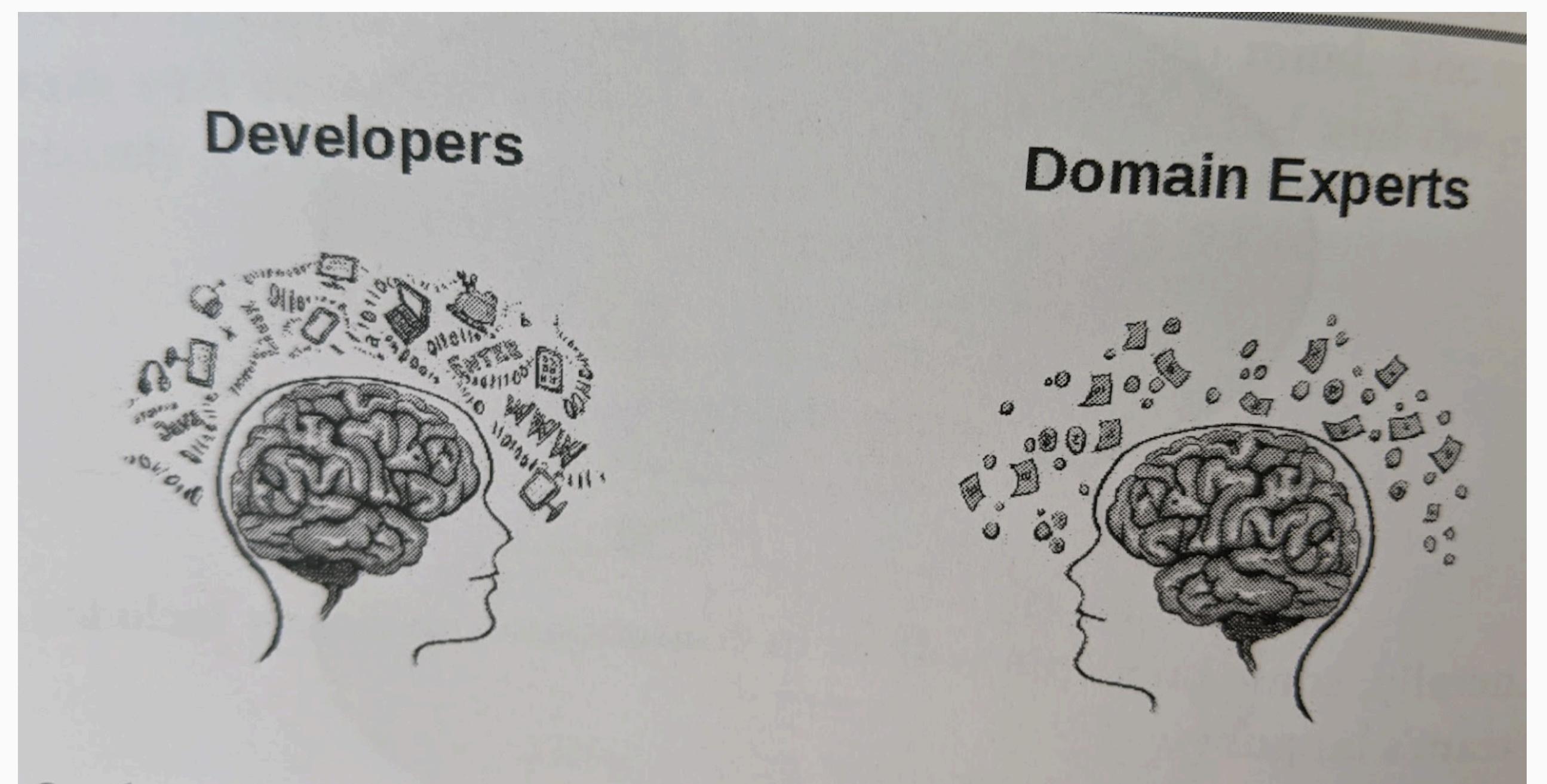
The original Book

Strategic: How team and business work together

Tactical: Implementation, Cool stuff engineers like

Model the software according to business

Engineers and domain experts
work together to design software



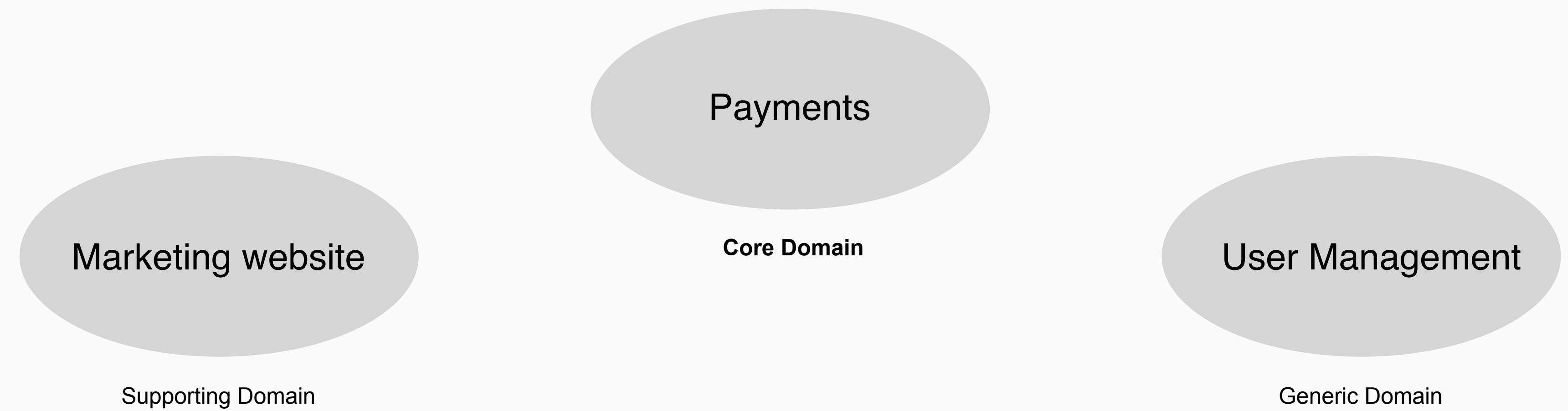
Main Strategic Principles of DDD

- Core Domain
- Bounded Contexts
- Ubiquitous Language

Core Domain

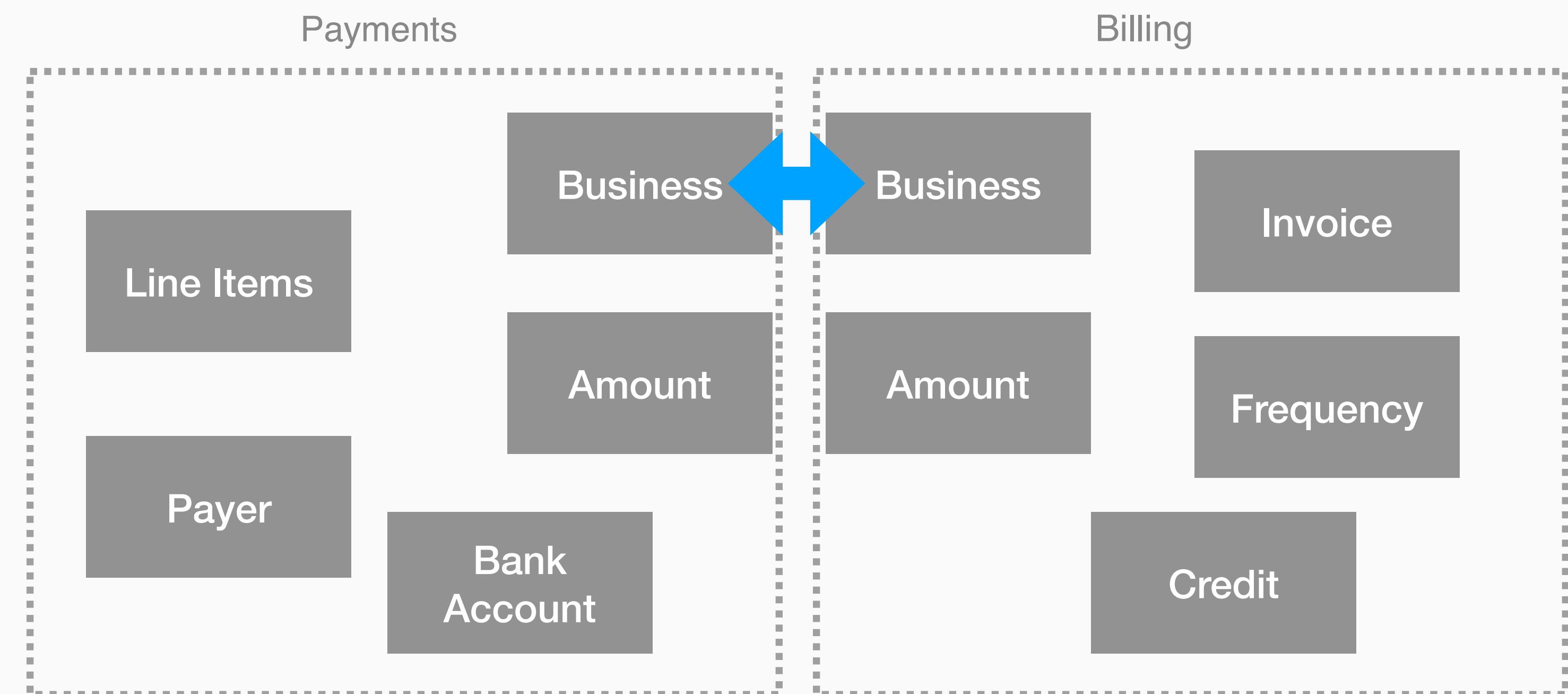
- Software model that is most important because it means to achieve greatness
- A Core Domain is developed to distinguish your organisation competitively from others
- Makes it clear what you should focus on

Domains



Core domain is where we should be focusing most energy.

Bounded Contexts



Domain built up of bounded contexts. Ubiquitous within contexts

Event Storming

Event Storming

- Rapid design technique meant to engage both domain experts and engineers
- Focuses on events and business process rather than classes and database
- Visual approach puts everyone on level footing
- Very fast and cheap to perform
- Team will have breakthroughs in understanding
-

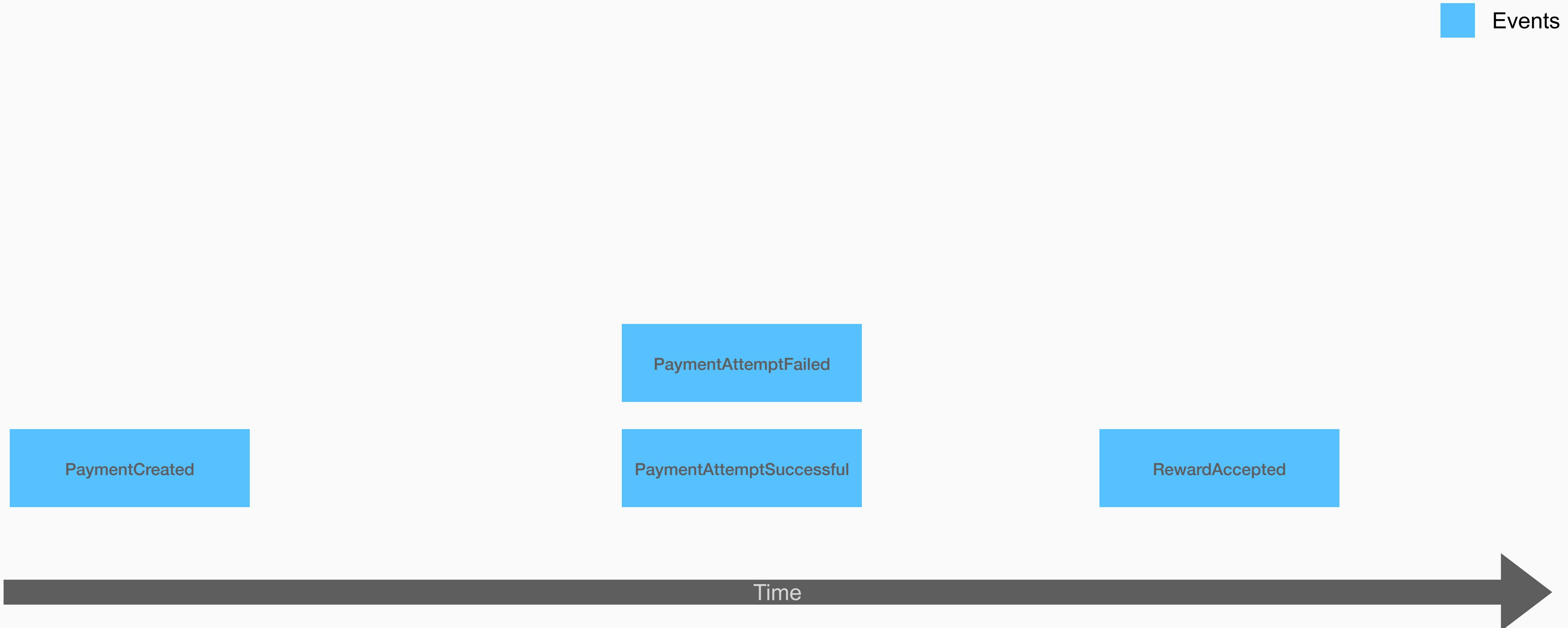
Event Storming.

Use a wall or an equivalent digital tool



Event Storming.

Storm domain events of business process

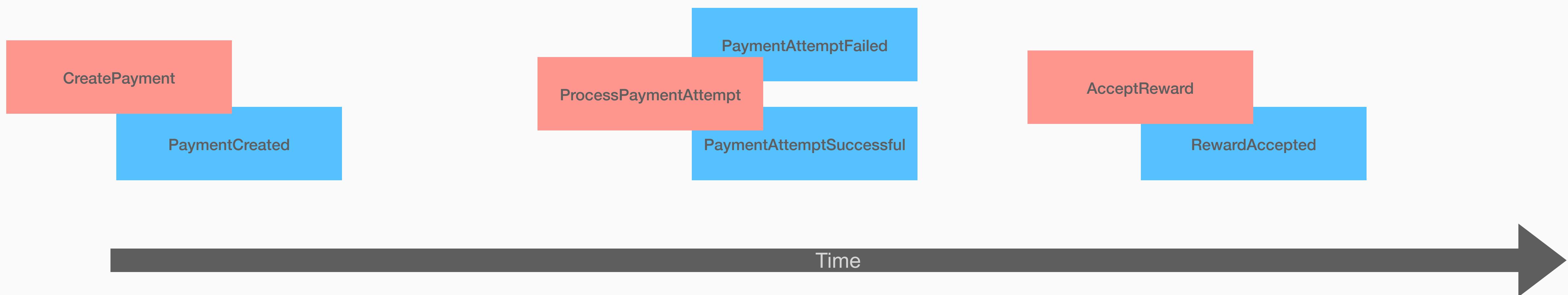


- Different colour per element
- Collaborate and question each stage!

Event Storming.

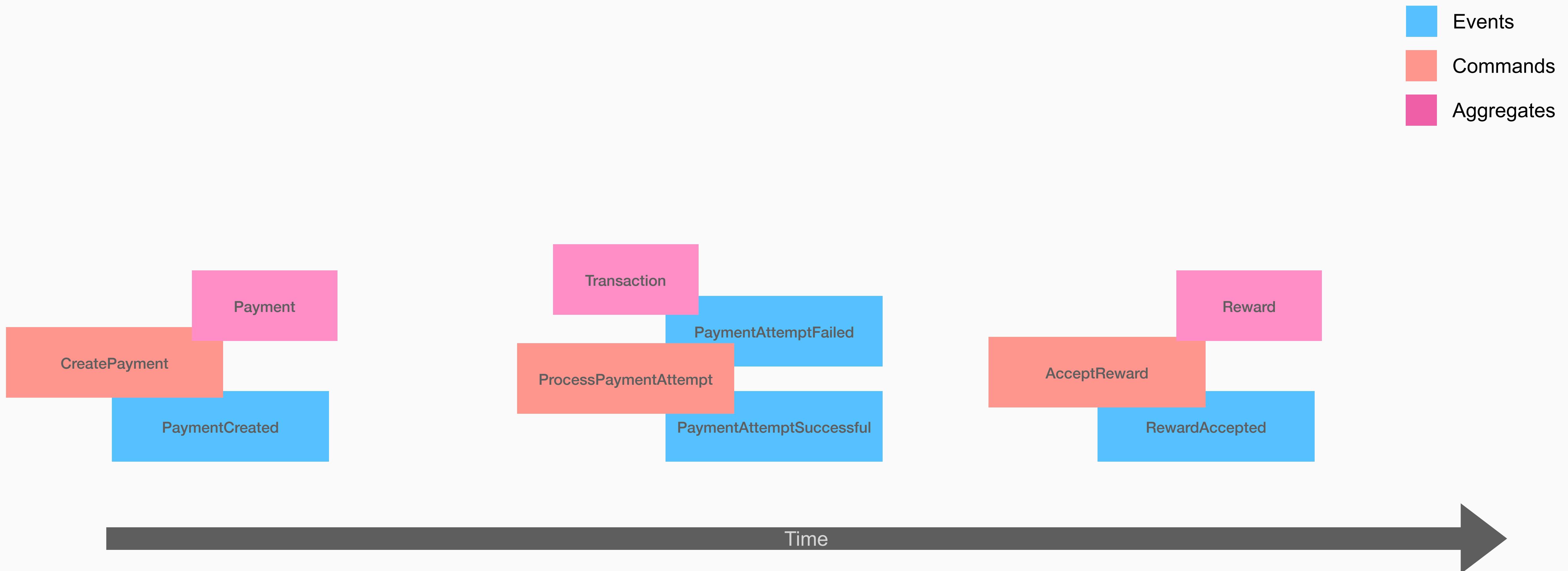
Add Commands that cause each event

Events
Commands



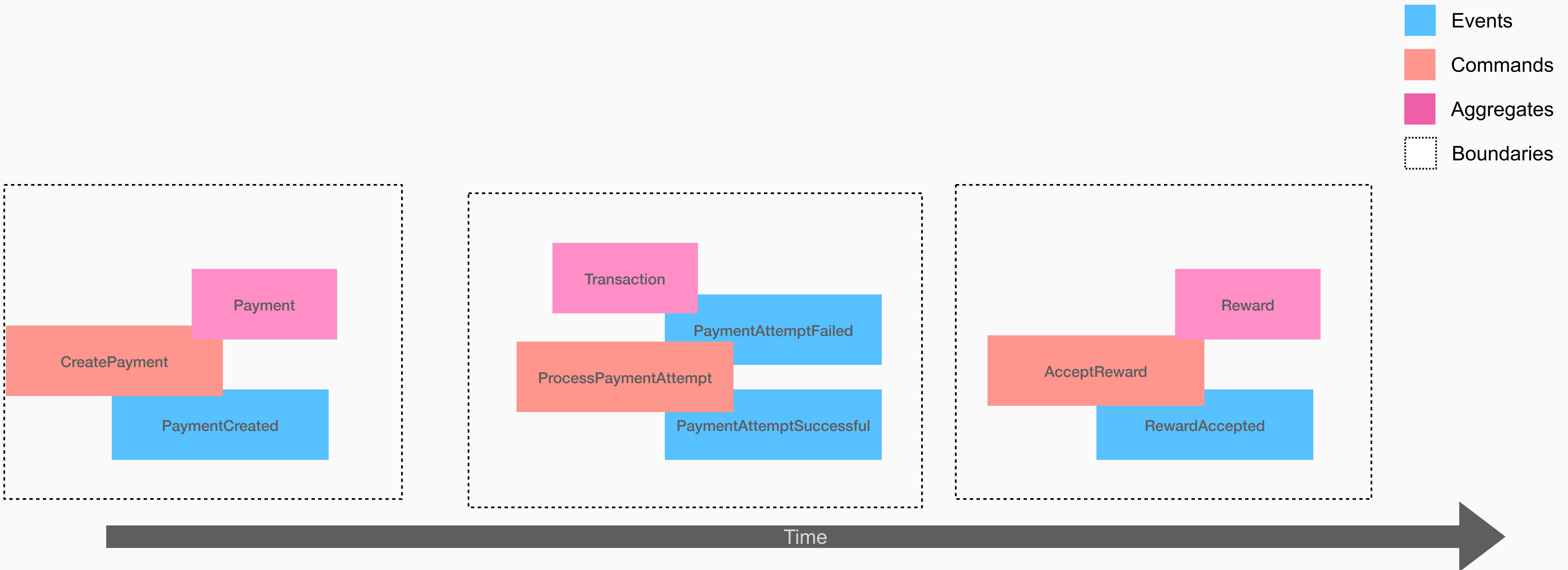
- Different colour per element
- Collaborate and question each stage!

Event Storming. Associate to aggregate



- Different colour per element
- Collaborate and question each stage!

Event Storming. Draw Boundaries



- Different colour per element
- Collaborate and question each stage!

Event Storming

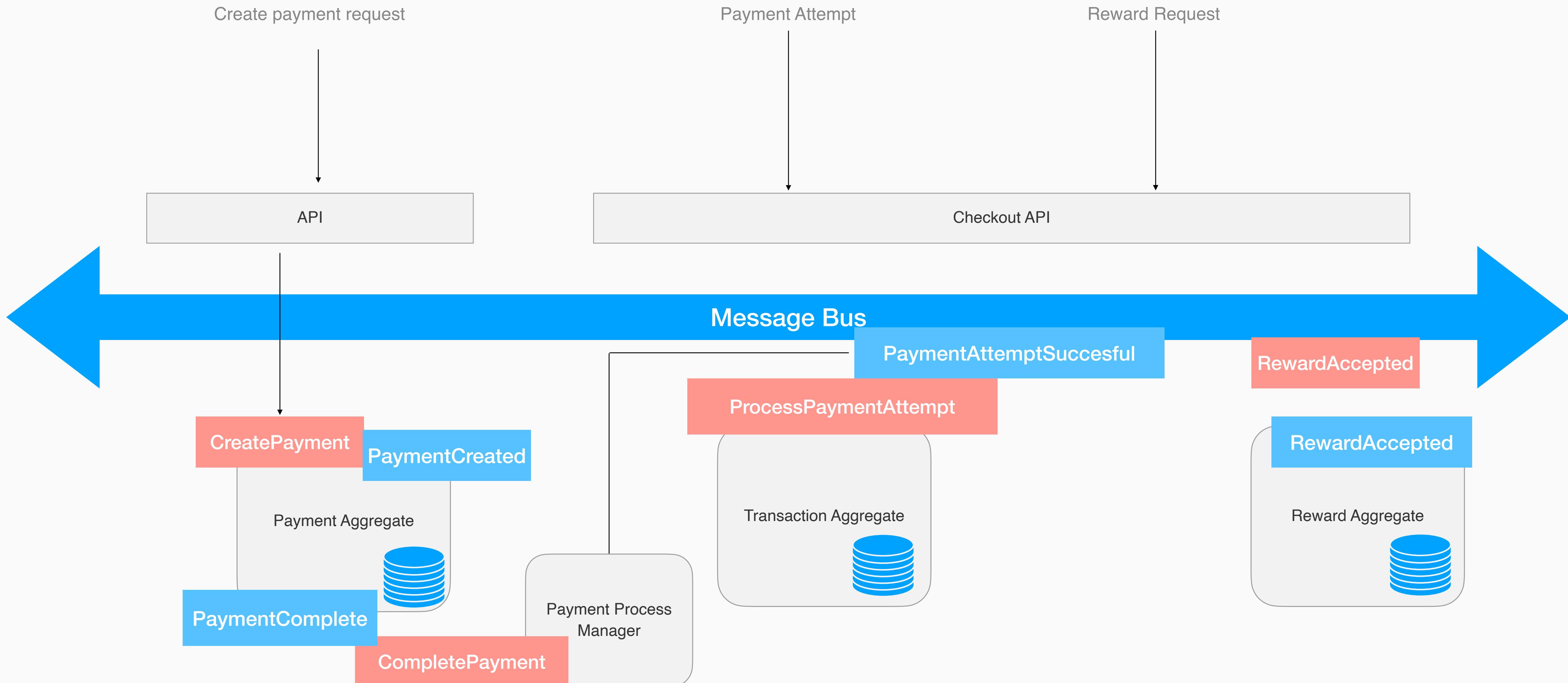
- Used to perform sessions with team to accelerate modelling efforts
- Start with short sessions (2 hours max)
- Don't get bogged down with edge cases (record and answer later)
- Revisit during project lifecycle to reinforce/modify domain as you learn

Mapping Architecture to CQRS

DDD maps to reactive/CQRS architecture nicely

- Command, Query Response Segregation
- Aggregates are called aggregates as they aggregate state
- Robustness is built into these architectures
 - Ordered eventual consistency negates race conditions
 - Retries built in
 - Horizontal scaling via message consumers
 - Query efficiently from read models

CQRS Implementation



The difference between commands and events

Commands:

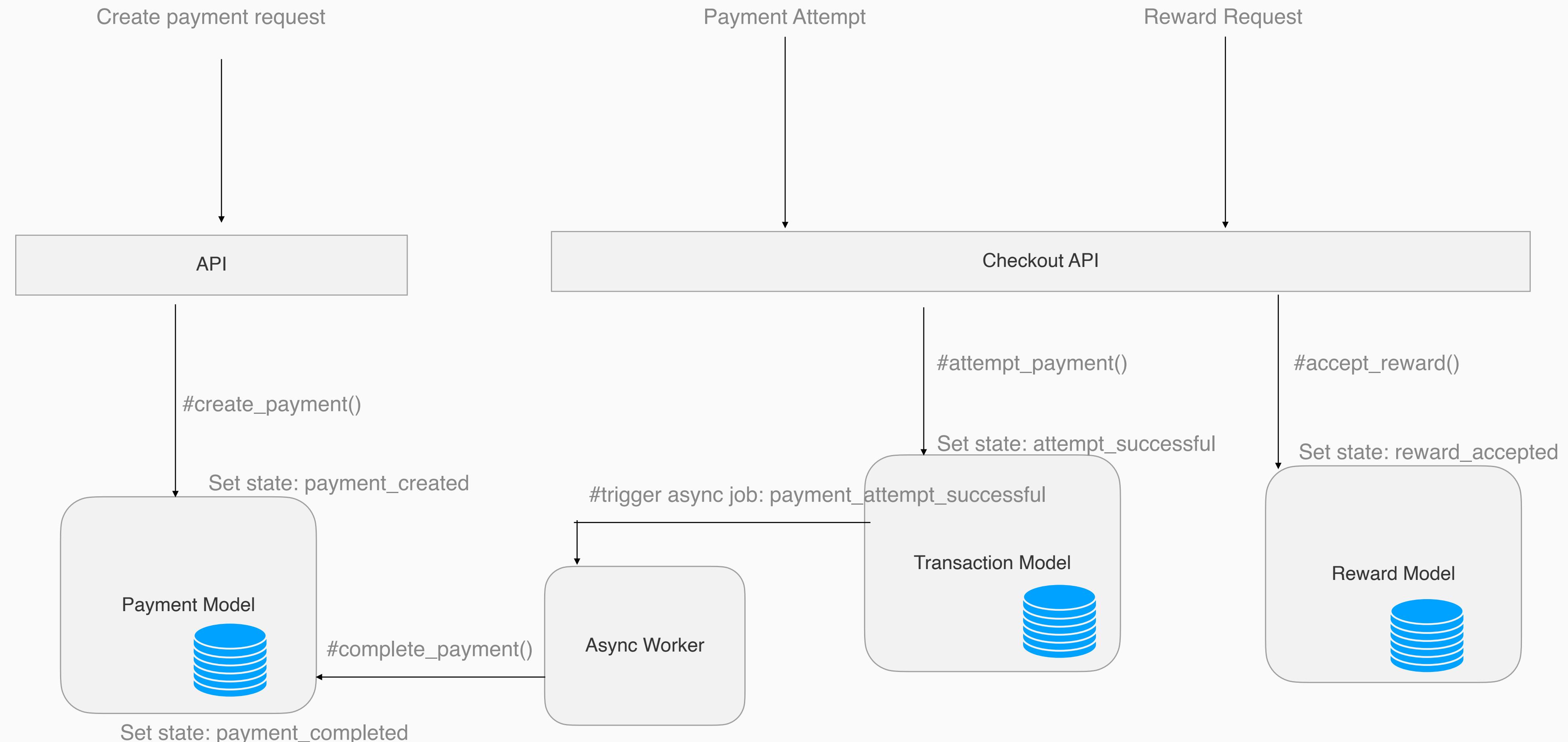
- Statement of intent, (Do Something)
- Handled by one aggregate (or process manager) only
- The Result of command published as an event

Events:

- Past statement of something that has happened
- Can be handled by 1 or many handlers

Don't let CQRS put you off

Non event-driven architecture



Summary

- Without design we will end up with a big ball of mud
- DDD is a technique for aligning software with business
- Collaboration and strategic alignment are fundamental to success
- Ubiquitous language exists within a domain
- Event storming is a technique for learning about domain and architecture
- DDD does map onto Reactive architecture well
- There is value in DDD and event storming even if you can't use or don't want event driven architecture

Thank you :)

Contact

lewis@banked.com

Lewis Jones— Lead Engineer