# Event-Driven Design
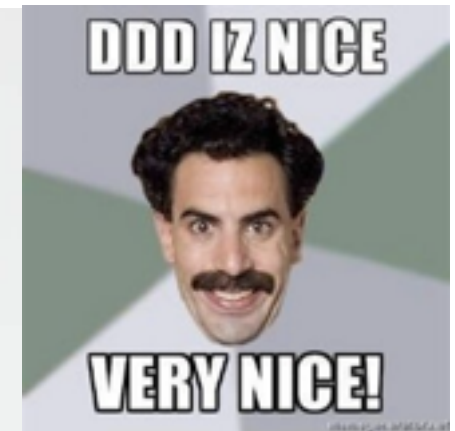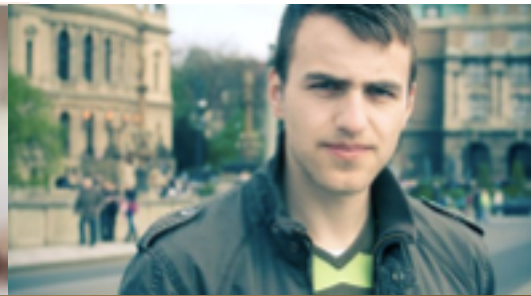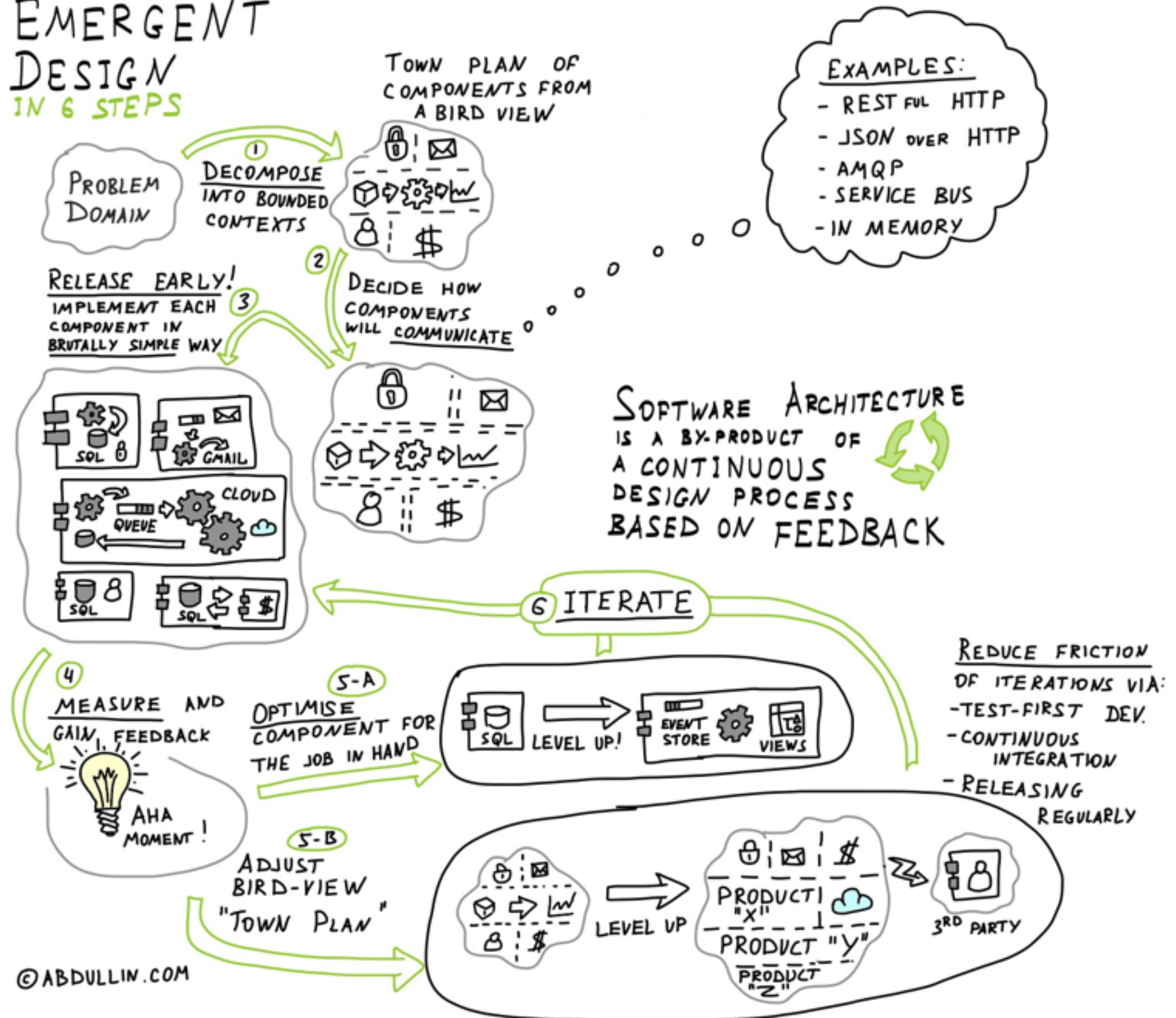
Rinat Abdullin | abdullin.com | @abdullin

# Plan

- Software Design

- Event Storming

- Practical Event-Driven Design

- CQRS Beers

DDD IZ NICE

VERY NICE!

# EMERGENT DESIGN
## IN 6 STEPS

PROBLEM DOMAIN

1 DECOMPOSE INTO BOUNDED CONTEXTS

TOWN PLAN OF COMPONENTS FROM A BIRD VIEW

EXAMPLES:
- RESTful HTTP
- JSON over HTTP
- AMQP
- SERVICE BUS
- IN MEMORY

2 DECIDE HOW COMPONENTS WILL COMMUNICATE

RELEASE EARLY!
3 IMPLEMENT EACH COMPONENT IN BRUTALLY SIMPLE WAY

SQL    GMAIL

QUEUE    CLOUD

SQL    SQL

SOFTWARE ARCHITECTURE IS A BY-PRODUCT OF A CONTINUOUS DESIGN PROCESS BASED ON FEEDBACK

6 ITERATE

4 MEASURE AND GAIN FEEDBACK

AHA MOMENT!

5-A OPTIMISE COMPONENT FOR THE JOB IN HAND

SQL    LEVEL UP!    EVENT STORE    VIEWS

REDUCE FRICTION OF ITERATIONS VIA:
- TEST-FIRST DEV.
- CONTINUOUS INTEGRATION
- RELEASING REGULARLY

5-B ADJUST BIRD-VIEW "TOWN PLAN"

LEVEL UP    PRODUCT "X"    PRODUCT "Y"    PRODUCT "Z"    3RD PARTY

©ABDULLIN.COM

"Software development is a learning process.

Working code is a side-effect."
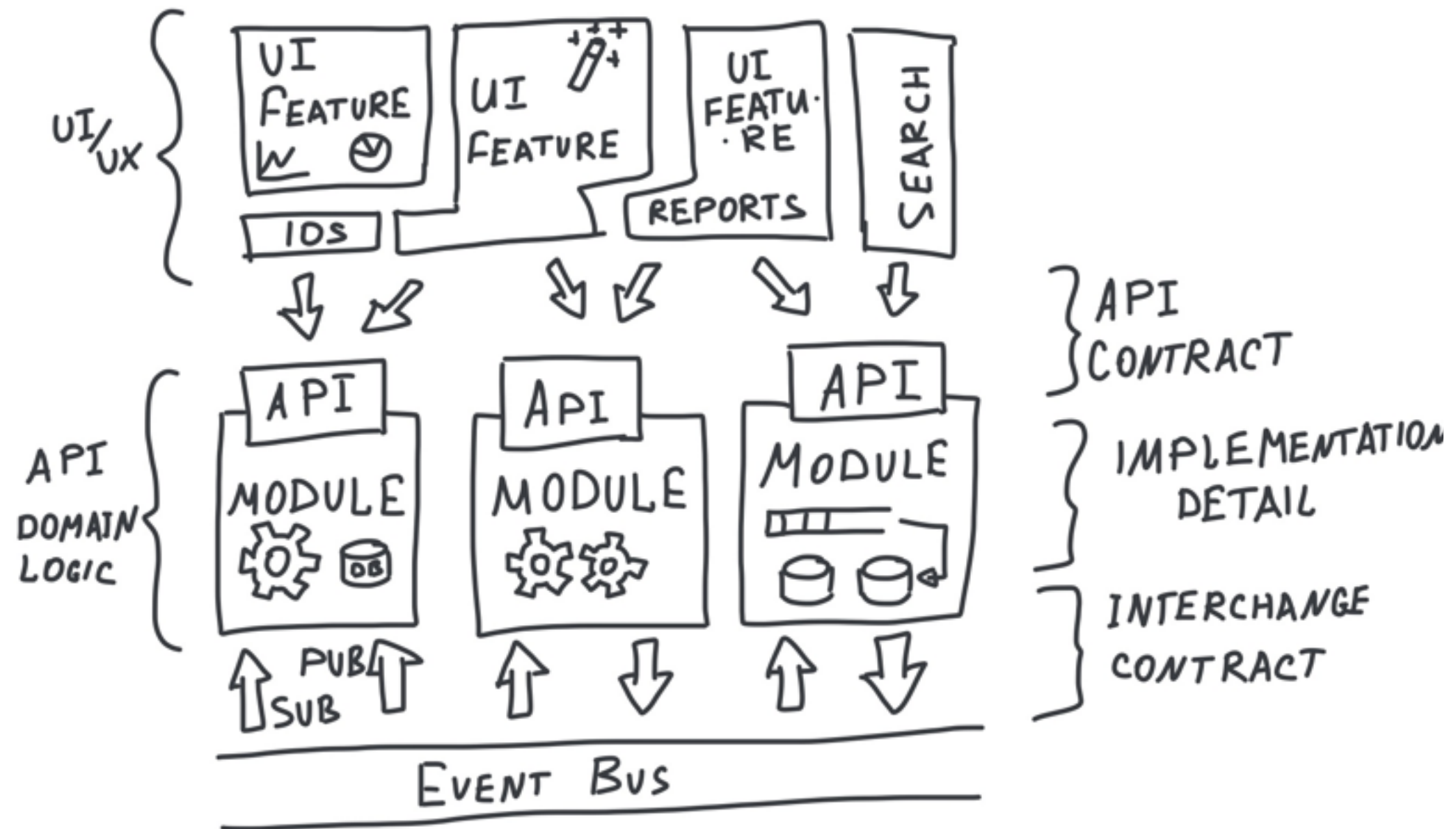
–EU DDD Community

# Context Map

Map is not the territory

# Divide and Conquer

# Contracts at Boundary

Events, API, ACL

# Many Faces of Domain Event

## Real·World ♡

Married to → Family was
Dr. Watson    Attending

⬚

Officially
Registered
Marriage          Sherlock
                  Was Late
⬚                    ⬚

Honey          Nobody Was
                    Killed
Moon Delayed

← ABSTRACTS

## Model ⌶

**User Renamed**

Mary Morsten
Changed Name to
Mary   Watson.
Recorded in 1889
Reason : Marriage

⬆ Implements

## Represents

{ "TYPE": "USER-RENAMED--V2",
  "EVENTID": "1371C6 b236 DDC7...",
  "OLDNAME": "MARY MORSTEN",
  "NEW NAME": "MARY WATSON",
  "RECORDED": "1889-06-14  15:22",
  "REASON": "MARRIAGE" }

→ INSTANCE OF

Type   User Renamed

Event ID : ID

Old Name : Full Name

New Name : Full Name

Recorded Utc : Time

Reason : Rename Reason

## INSTANCE : 1371C6...

## Contract

# Questions?

# Event-Storming

Let's explore our boundaries

# Back to the code

Practical Applications

UI/UX {

UI FEATURE W

UI FEATURE

UI FEATURE
REPORTS

SEARCH

IOS

API CONTRACT }

API DOMAIN LOGIC {

API MODULE

API MODULE

API MODULE

IMPLEMENTATION DETAIL

PUB SUB

INTERCHANGE CONTRACT

EVENT BUS

API (ACL)

Event Bus
(with Replay)

Publish
Events

API HANDLER
(BEHAVIOR)

QUERY

STATE

Event
Handler

WRITE

EVENTS
SUBSCRIBE

THEN

WHEN REQ.        RESP.

API

GIVEN

THEN

COMPONENT

EVENTS

EVENTS
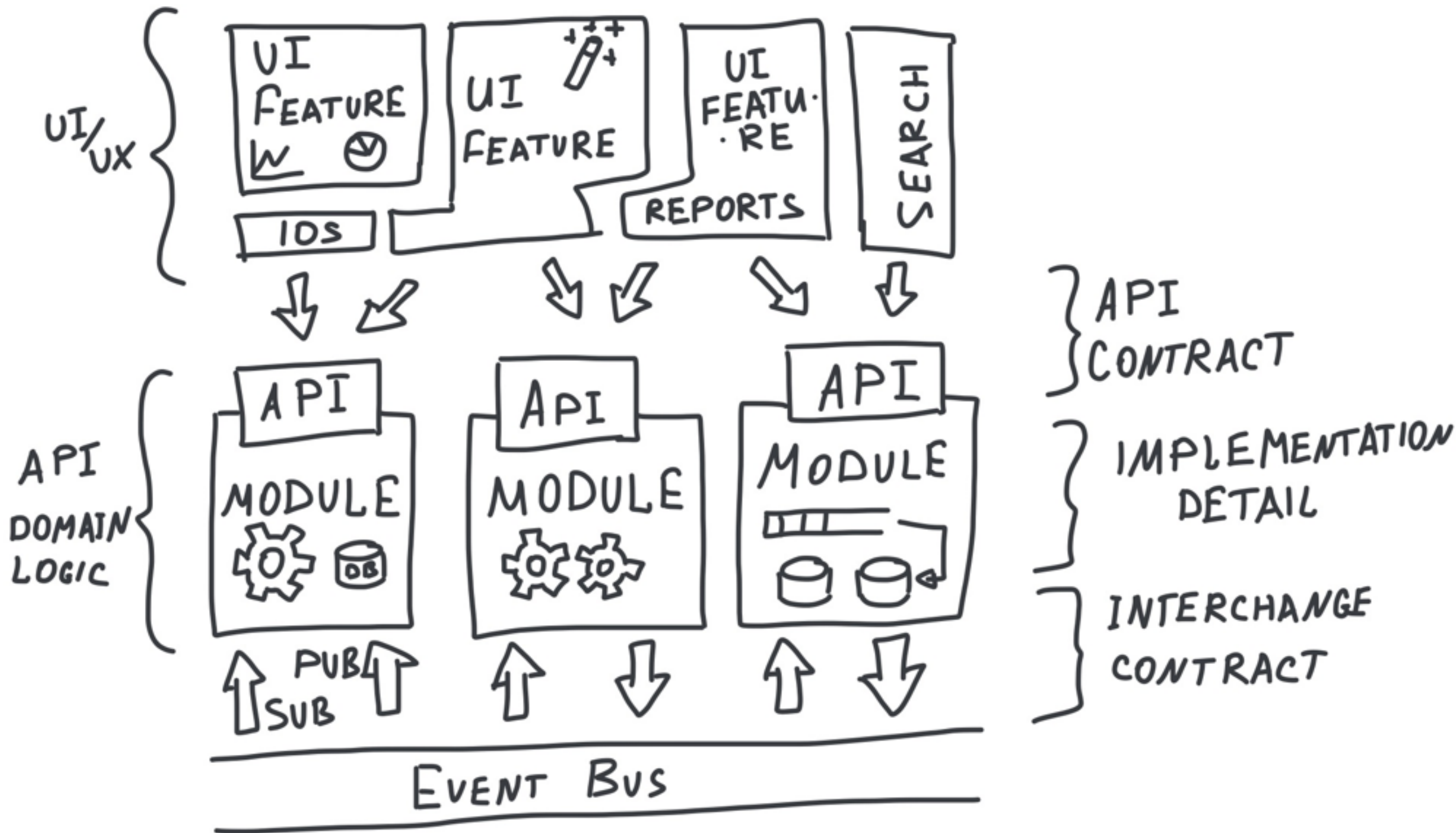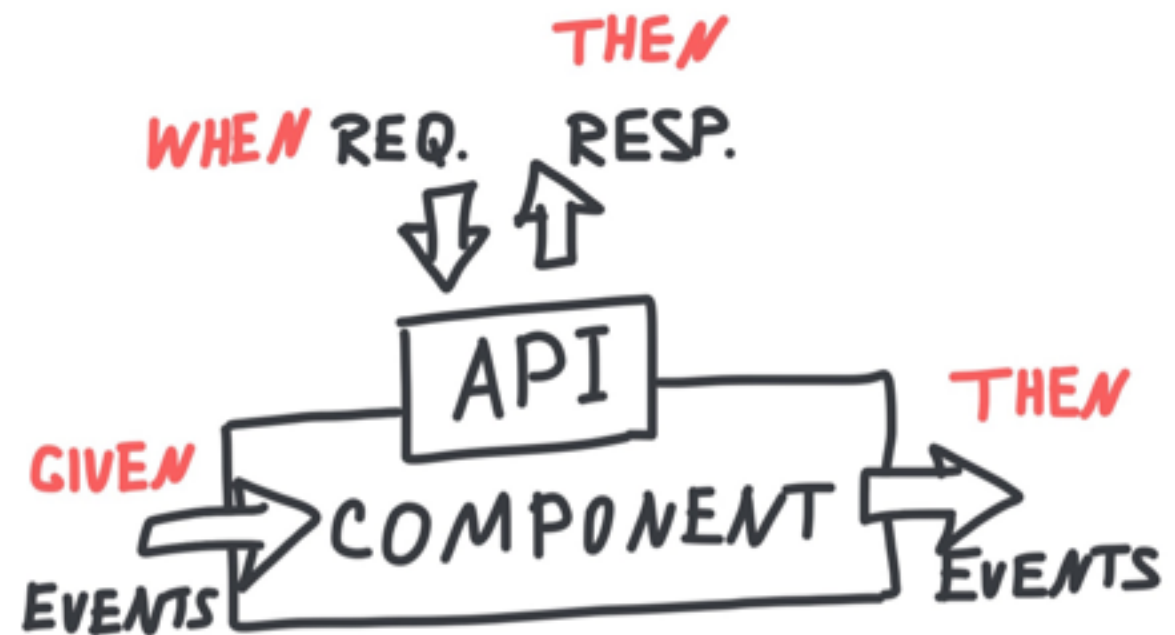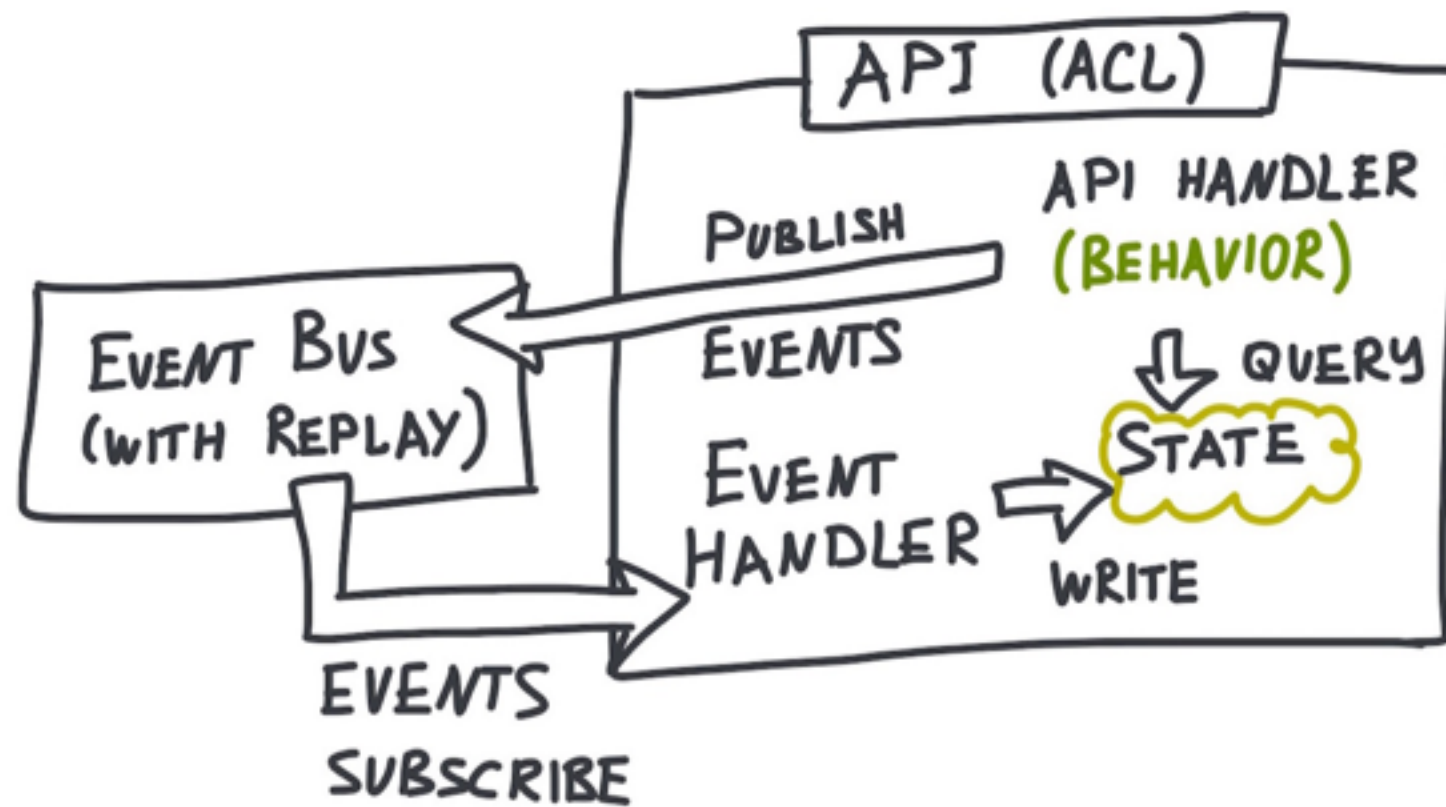
# Check out the code

MANY PATHS OF AN EMERGENT DESIGN. APPLY TO EACH COMPONENT SEPARATELY.

START HERE →

HELLO WORLD ⚙ SHELL CONSOLE

- ADD CLI/REPL
- SCRIPT ANY COMPONENT

GOOD ENOUGH FOR PRODUCTION

COMPLEXITY INCREASES ↓

- EASY SCALING
- NO-DOWNTIME UPGRADES (VIP SWAP)

STATELESS SERVICES ←

API-FIRST DESIGN (API WITH MOCK IMPL.)

IN-MEMORY REPOSITORY ▣ →

NOSQL STORE

- CROSS-PLATFORM
- TOOLS TO DEBUG, TEST, BENCHMARK
- REPLAY HTTP SESSIONS TO TEST

- TEST DOUBLE
- OTHER TEAMS CAN WORK

PUT LONG REQUESTS IN QUEUE ▭⚙

JSON/XML OVER HTTP

EVENT SOURCING
AGG. ⟶ EVENTS

SPECIFICATION TESTS
⟶ ▭

GIVEN WHEN THEN

- REVERSE PROXIES
- API EVOLUTION
- BETTER ENTERPRISE INTEGRATION

COMPETING CONSUMERS ← NODE1 / NODE2

WORK PARTITIONING → NODE1 → NODE2

RESTFUL CONSTRAINTS

PROJECT EVENTS TO VIEW CACHE
EVENTS ⟶ CACHE

- FAST QUERIES
- GEO-AFFINITY
- MULTIPLEXED VIEW CACHES

- HUMAN-READABLE DOCS
- BDD
- NON-FRAGILE TESTS

- ELASTIC SCALABILITY
- CLOUD DEPLOYMENTS

PERFECT DESIGN

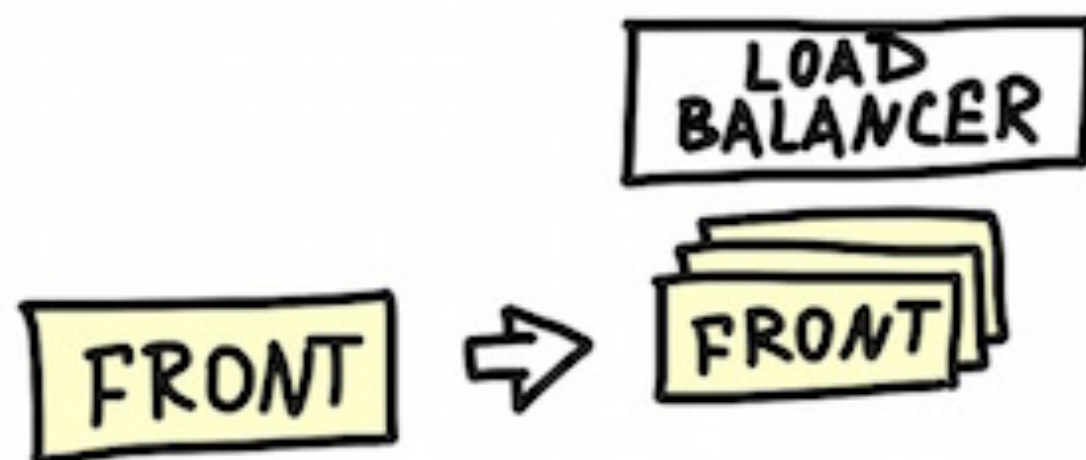THERE MUST BE A WAY TO GET HERE... SOMEHOW...

AN EXAMPLE OF EVOLUTION PATHS FOR A COMPONENT. EACH PROJECT WILL HAVE A DIFFERENT VERSION.

ABDULLIN.COM
@ABDULLIN

# SCALING

## SCALE FRONT·END
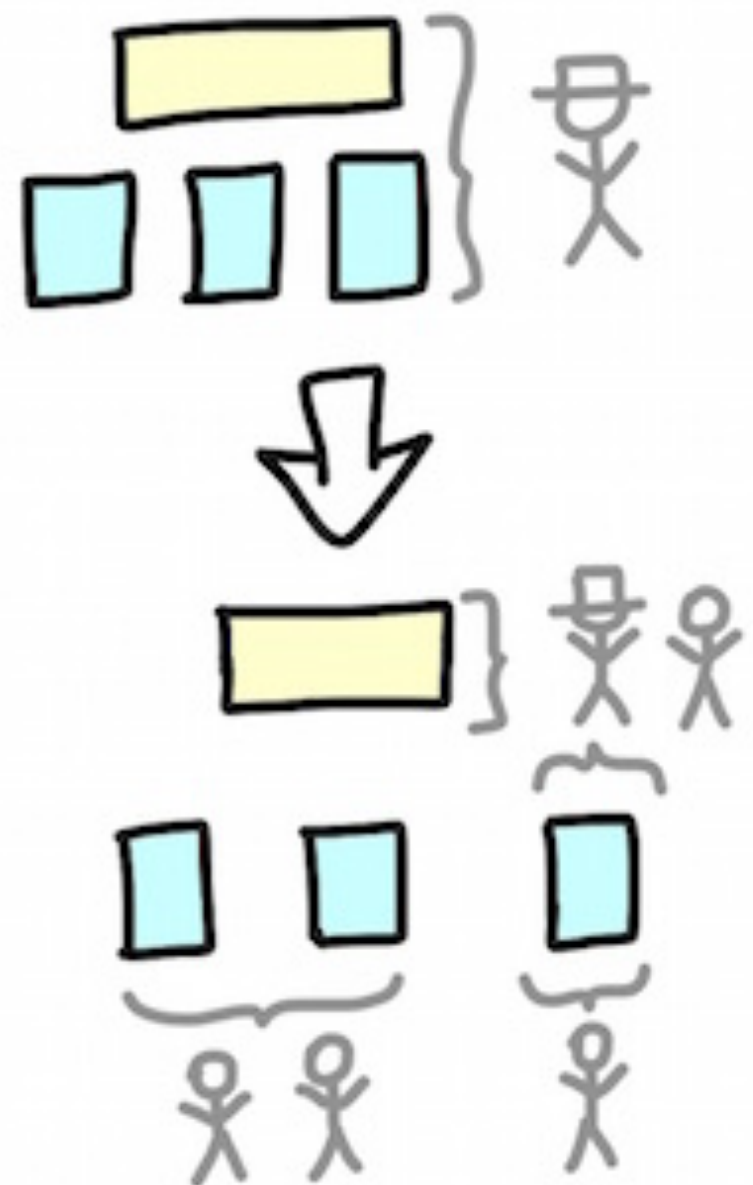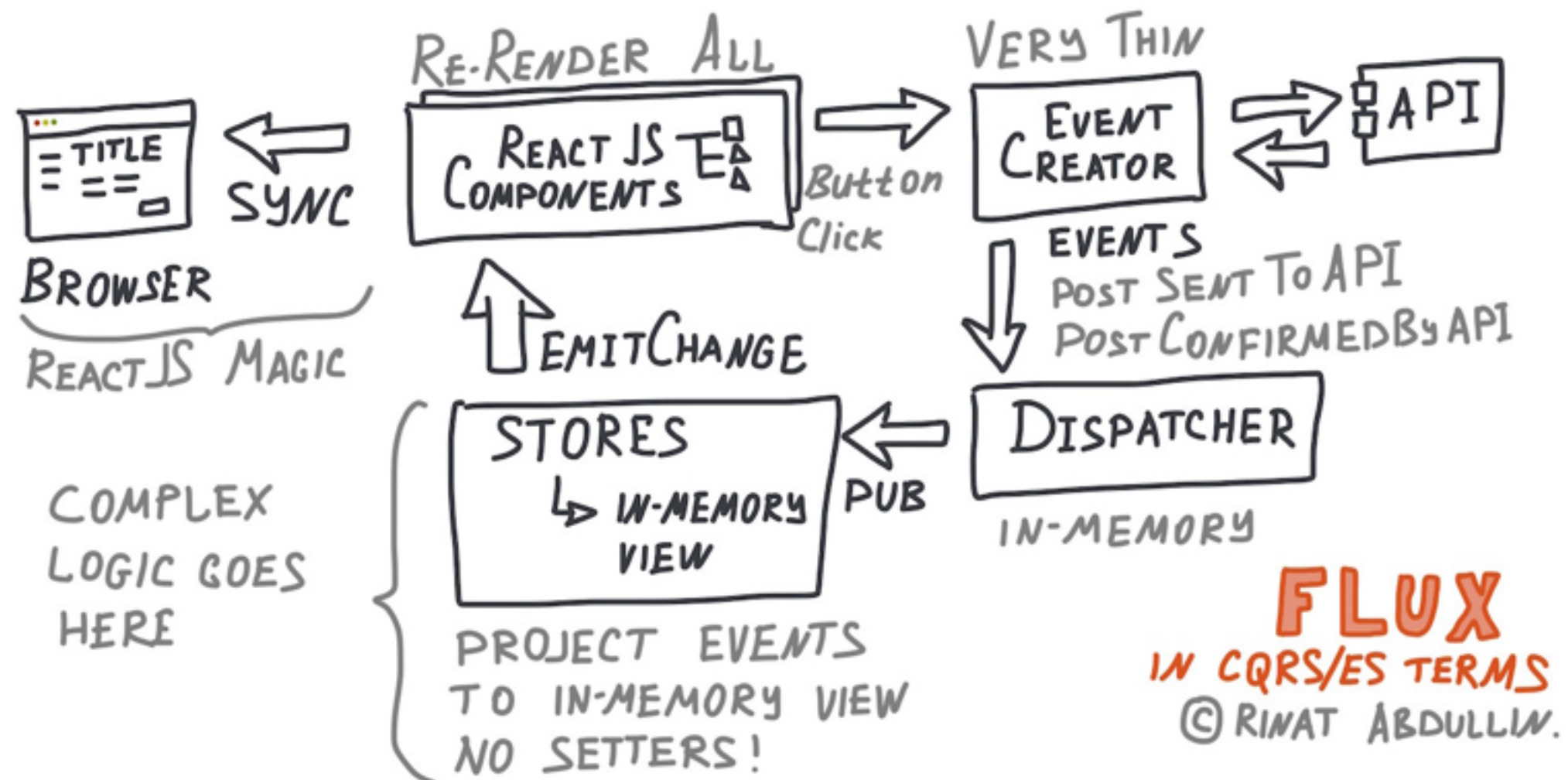
LOAD BALANCER

FRONT → FRONT

## SCALE BACK·END

API MODULE → API MODULE
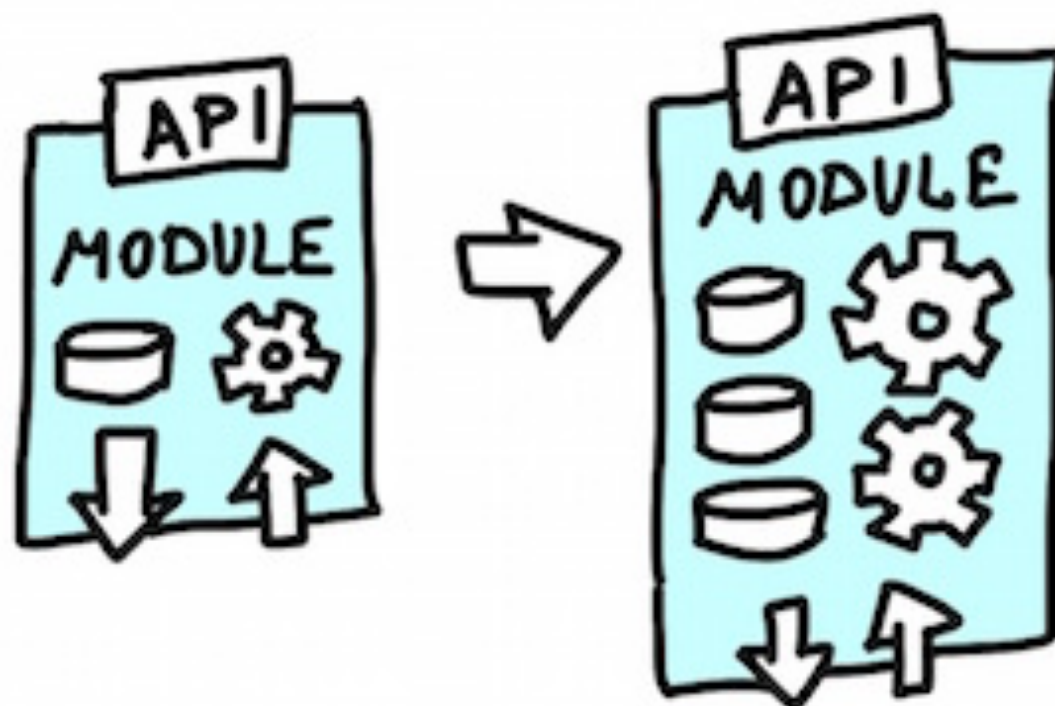
## SCALE DEVELOPMENT

# Flux/ReactJS

One of many options for UI/UX Design

# Questions?

Time for CQRS Beers!