

N26

Building Evolutionary Architectures

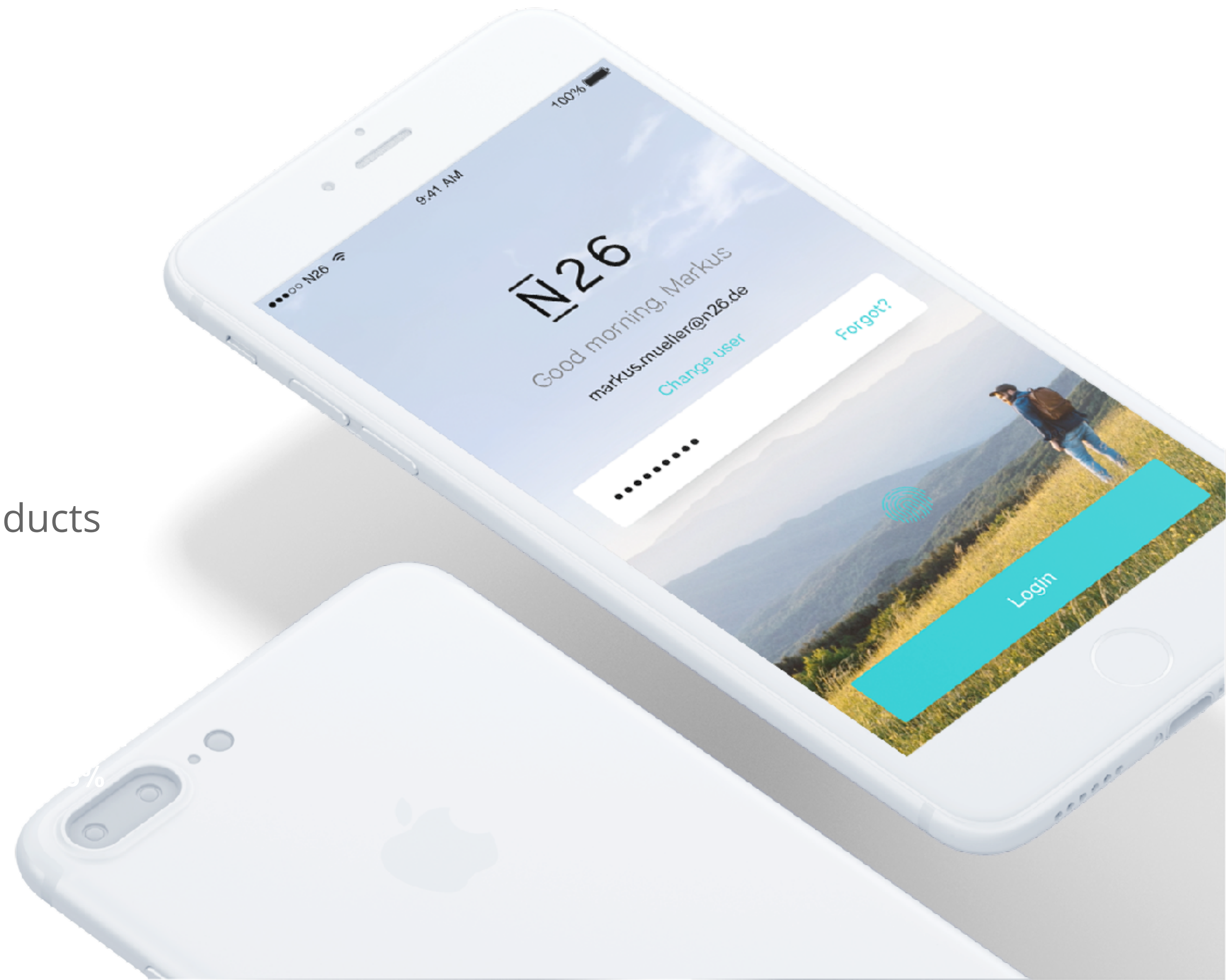
Patrick Kua (@patkua)

November 2017

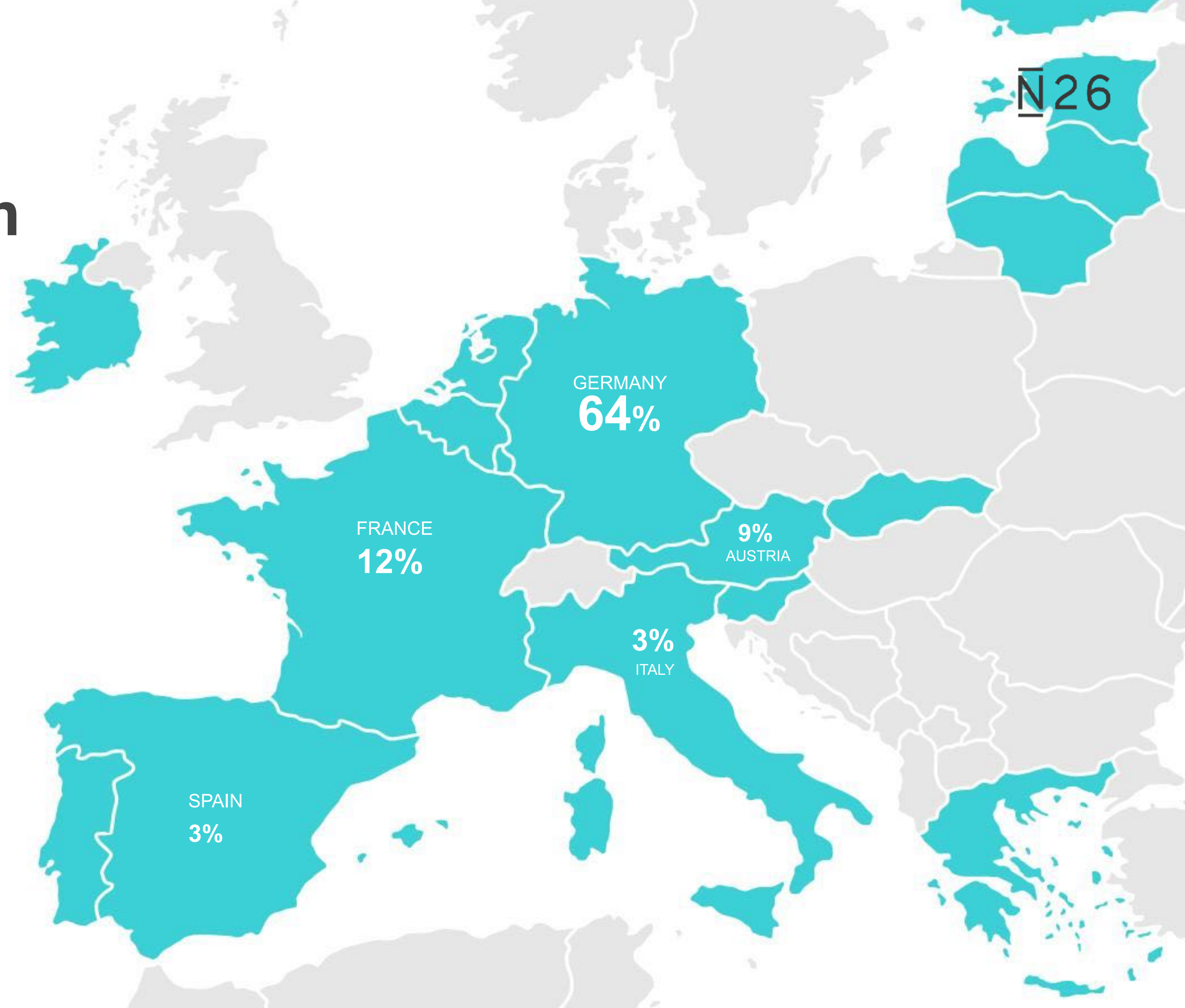


Digital Solution to Banking

- Full European banking license
- First fully mobile bank account
- One-touch access to all financial products



**500.000+ users in
17 countries**



Consistent strong recognition by Apple and Google



Featured as “Editor’s Choice”
Featured at Apple WWDC
Several promotions through Apple

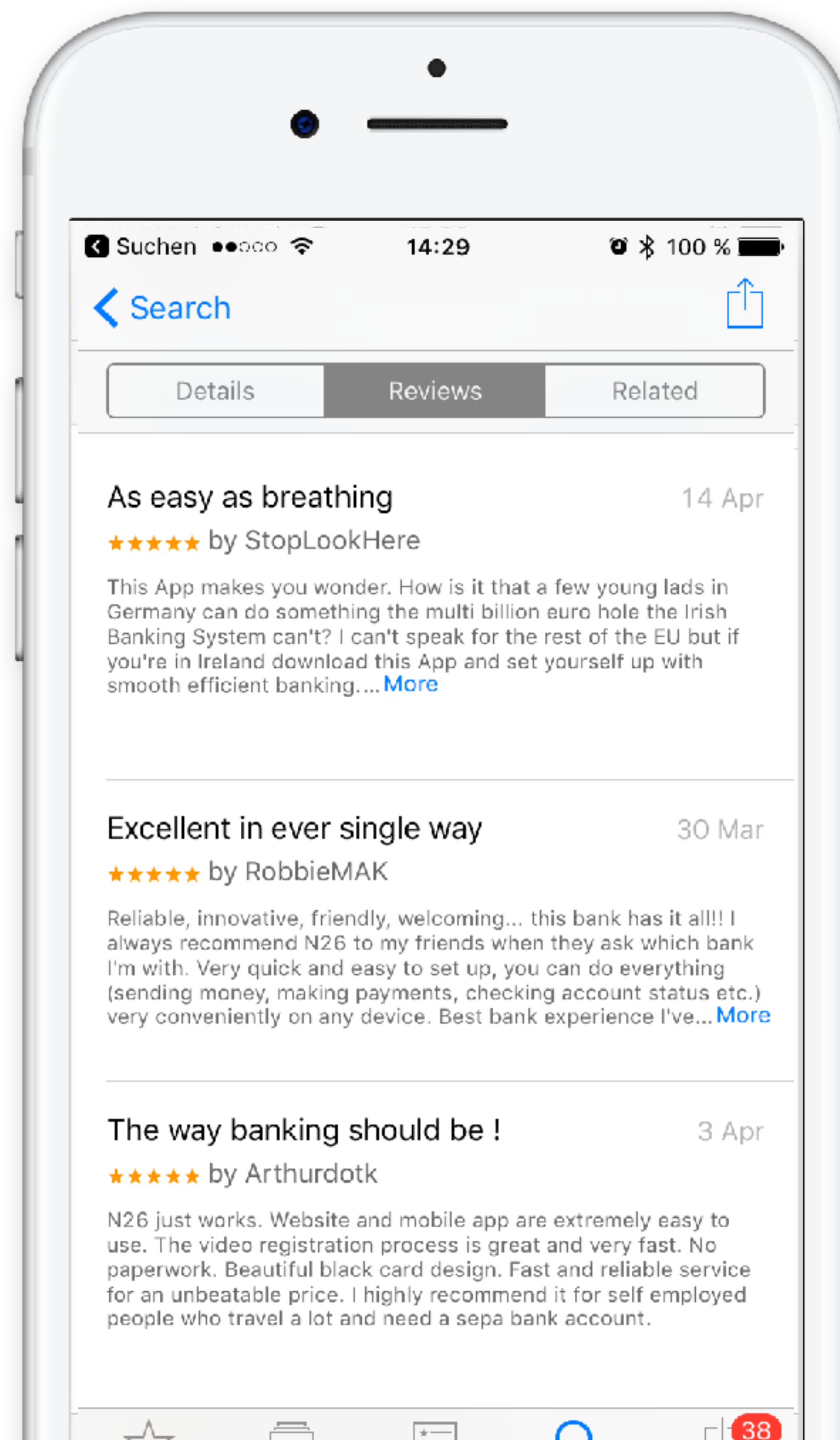


Featured in “Best Apps in 2016”
Featuring in the Play Store
with major releases

Users love the N26 experience



5,685 Reviews



CTO of N26

I am building the technology group behind the modern bank designed for the digital age.

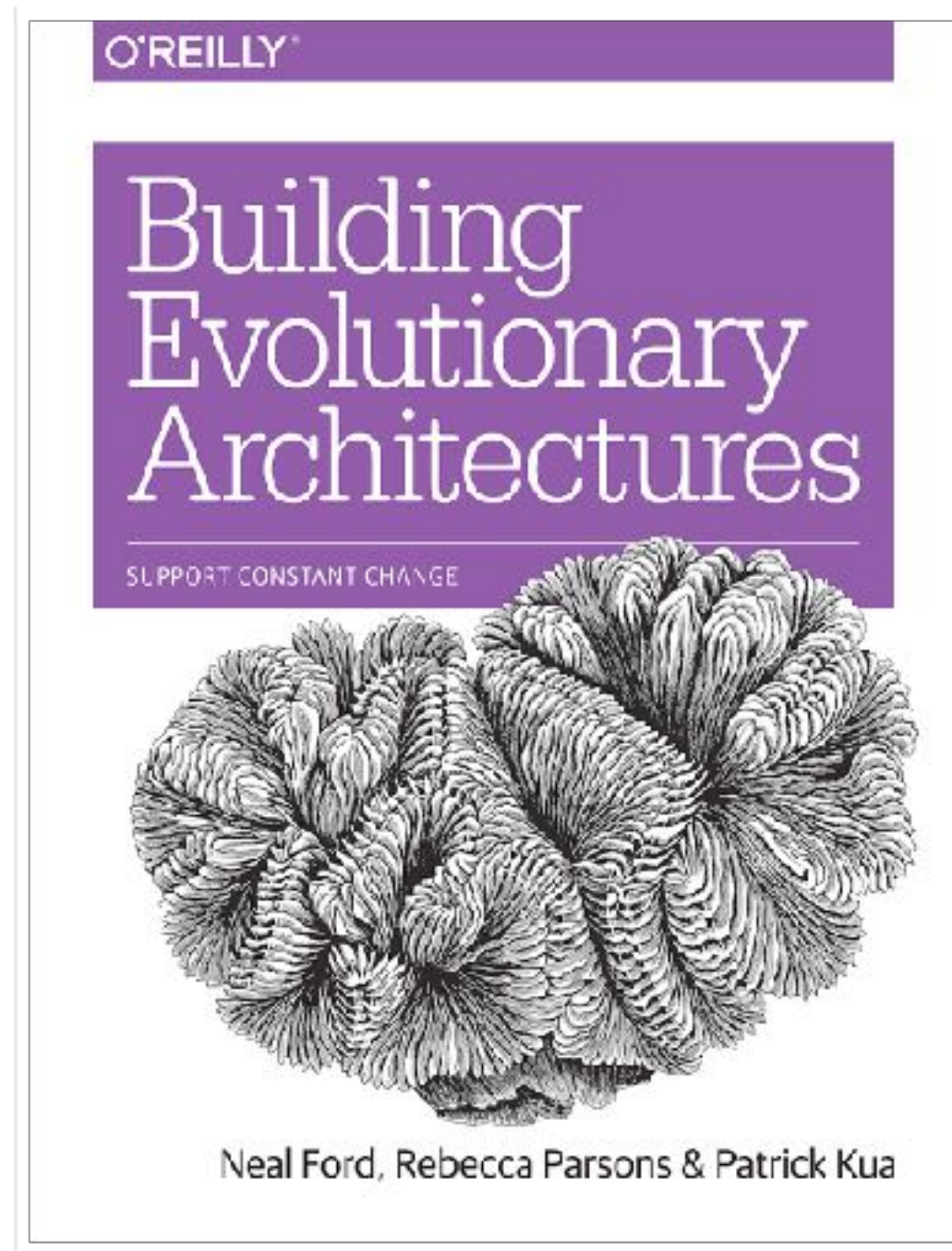
We are looking for people to join us on that journey:

<https://n26.com/jobs/> (Berlin)

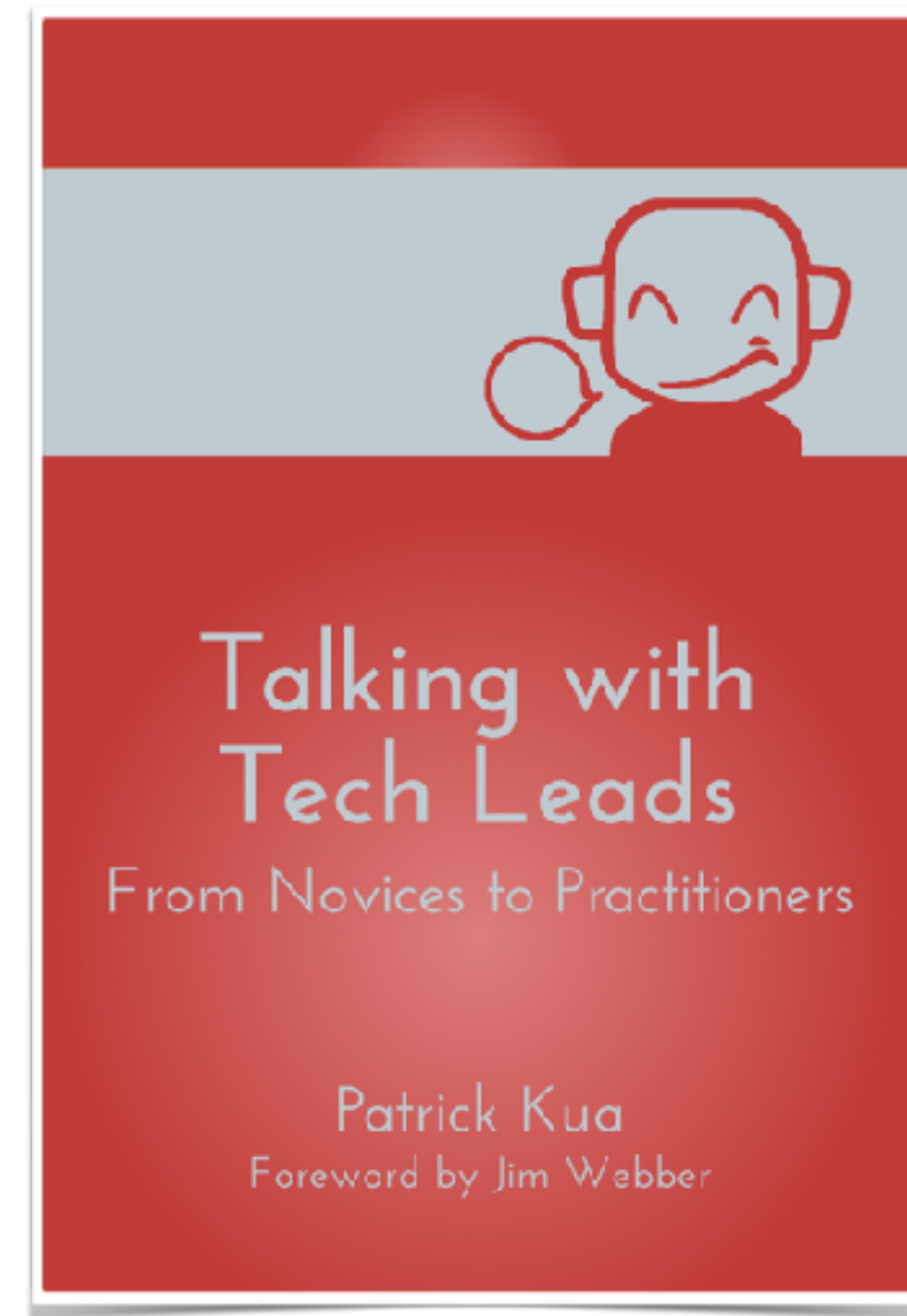
#leader #coach #architect #developer #life-long-learner
#speaker #author



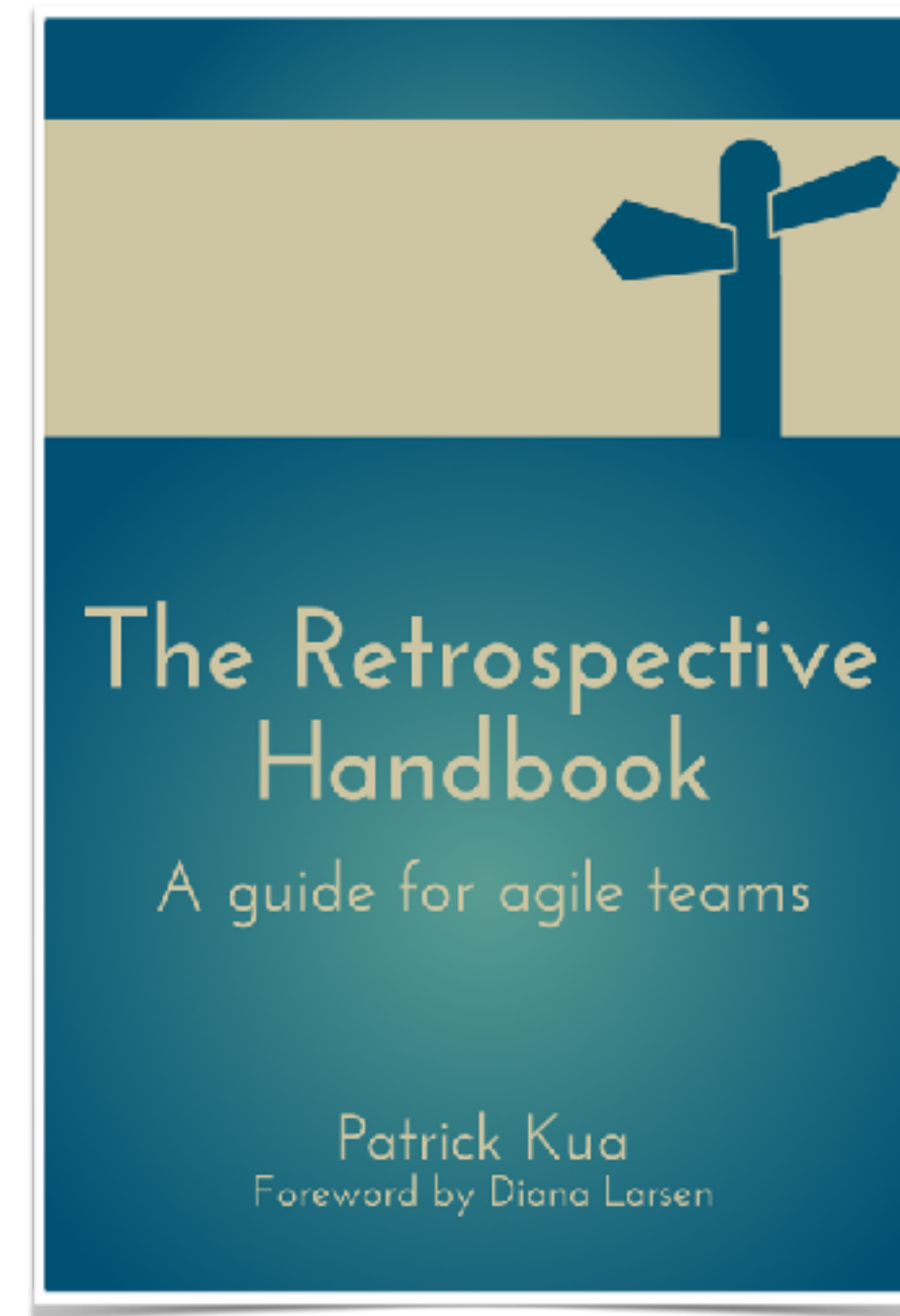
CTO of N26



thekua.io/evolarch



thekua.io/twtl



thekua.io/retrobook



EVOLUTION



CHANGE

... is inevitable

CHANGE

Technical Domain

Technical

Programming languages

Libraries

Frameworks

Tools

Operating environments

Technical constraints

CHANGE

Technical Domain

Domain

Revenue models

Base technology adoption

Competitors

Customer needs

Markets

Products

CHANGE

... is inevitable

If CHANGE

... is inevitable then

A case study

The trap many companies fall for





Idea

Investigation



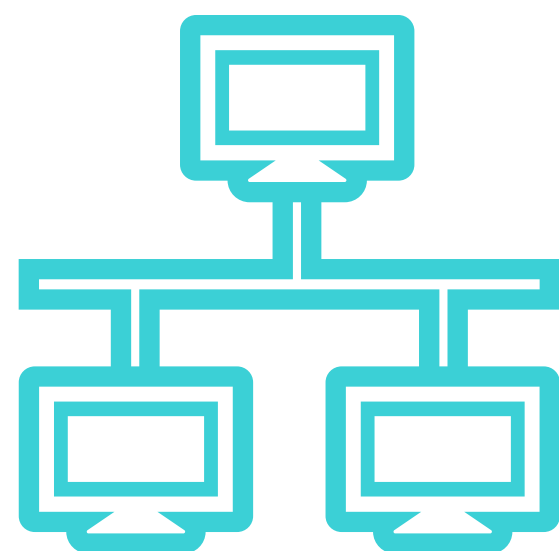
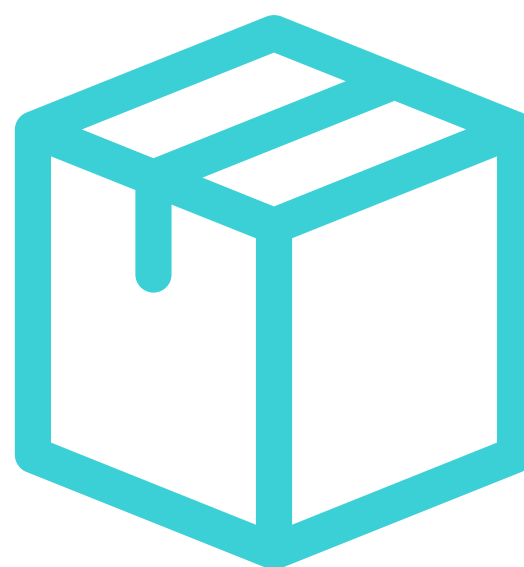
Shopping List



Plan for Change

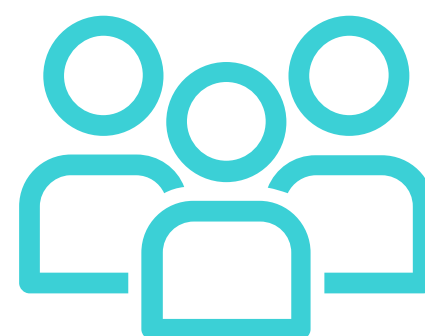


Product



Connect them all

Start



Later...



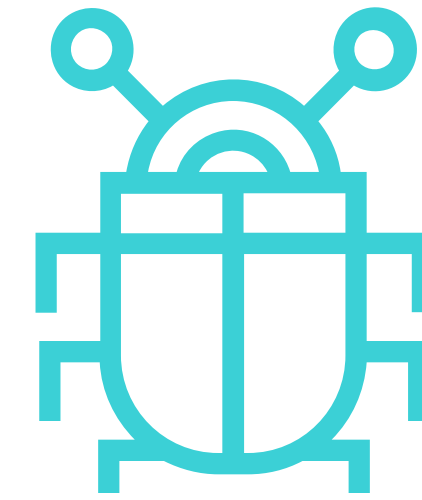
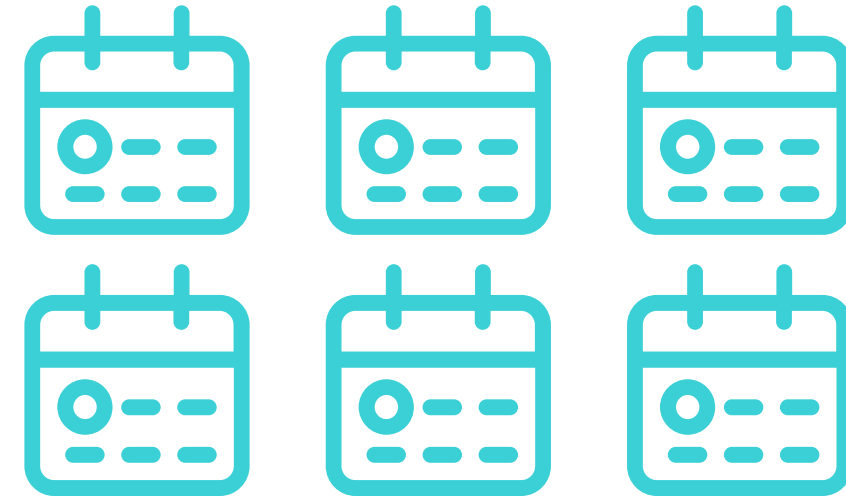
Almost "done"



Change!!!

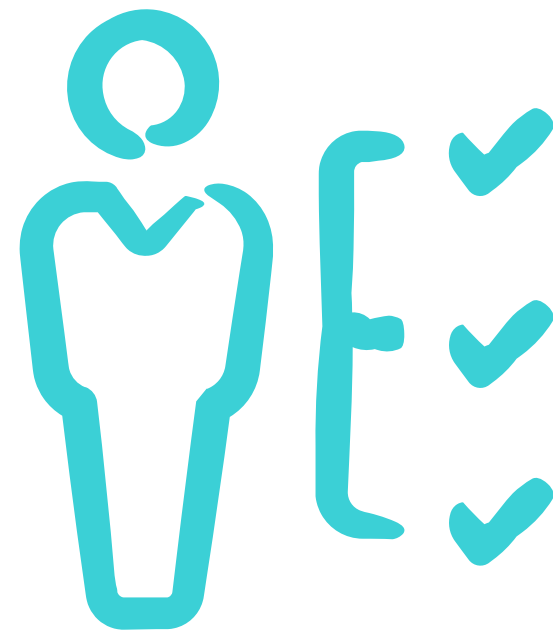


3-6 months



Still buggy

Vendor's schedule



Custom
costs



Change is hard



One precious
Environment



Stop!!!

WHAT IF...

we architected a
system for change?



DEFINITION



DEFINITION

An evolutionary architecture supports **incremental, guided** change as a first principle along **multiple dimensions**

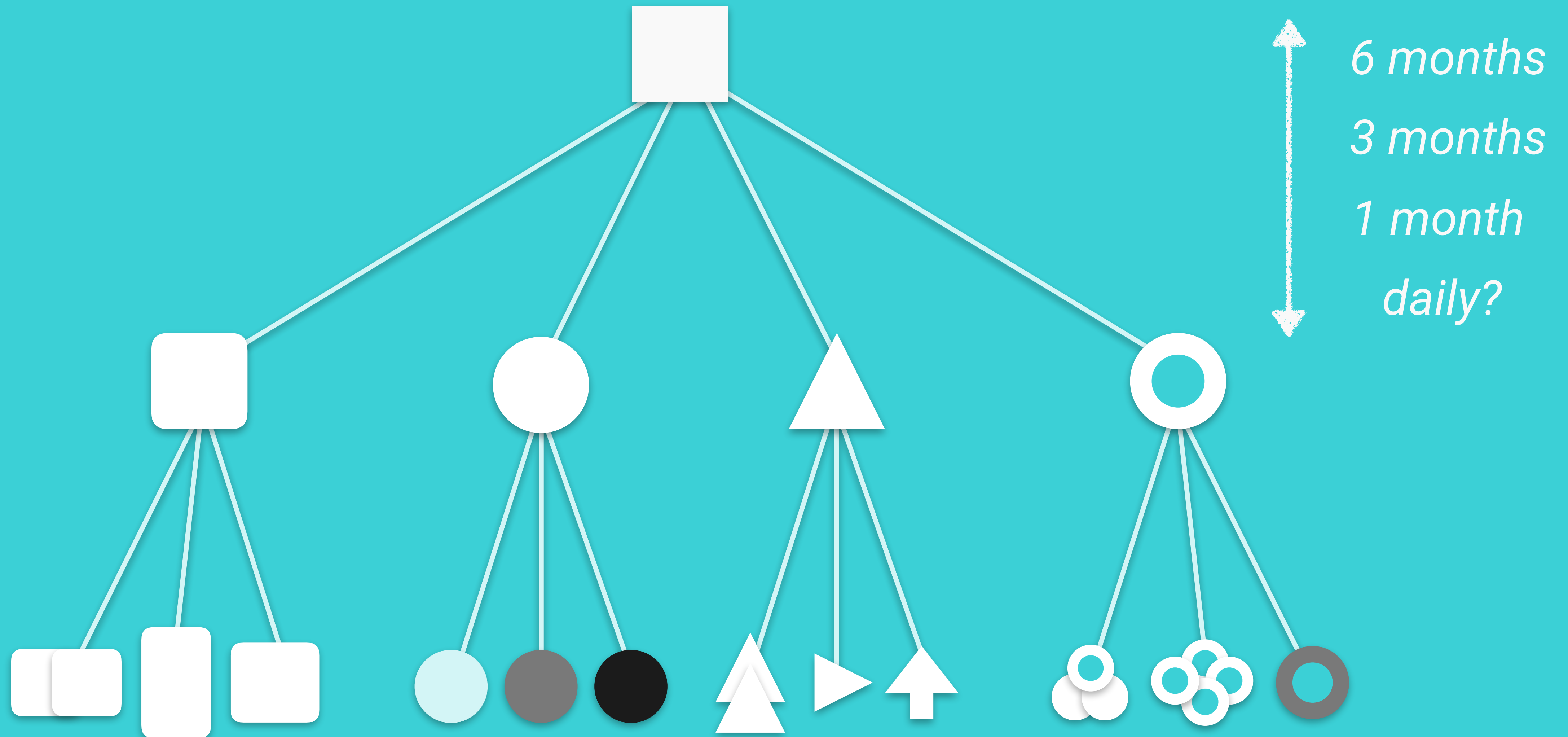


Architectures are evolved through
incremental releases

GENERATIONS

Incremental releases represent
GENERATIONS

GENERATIONS



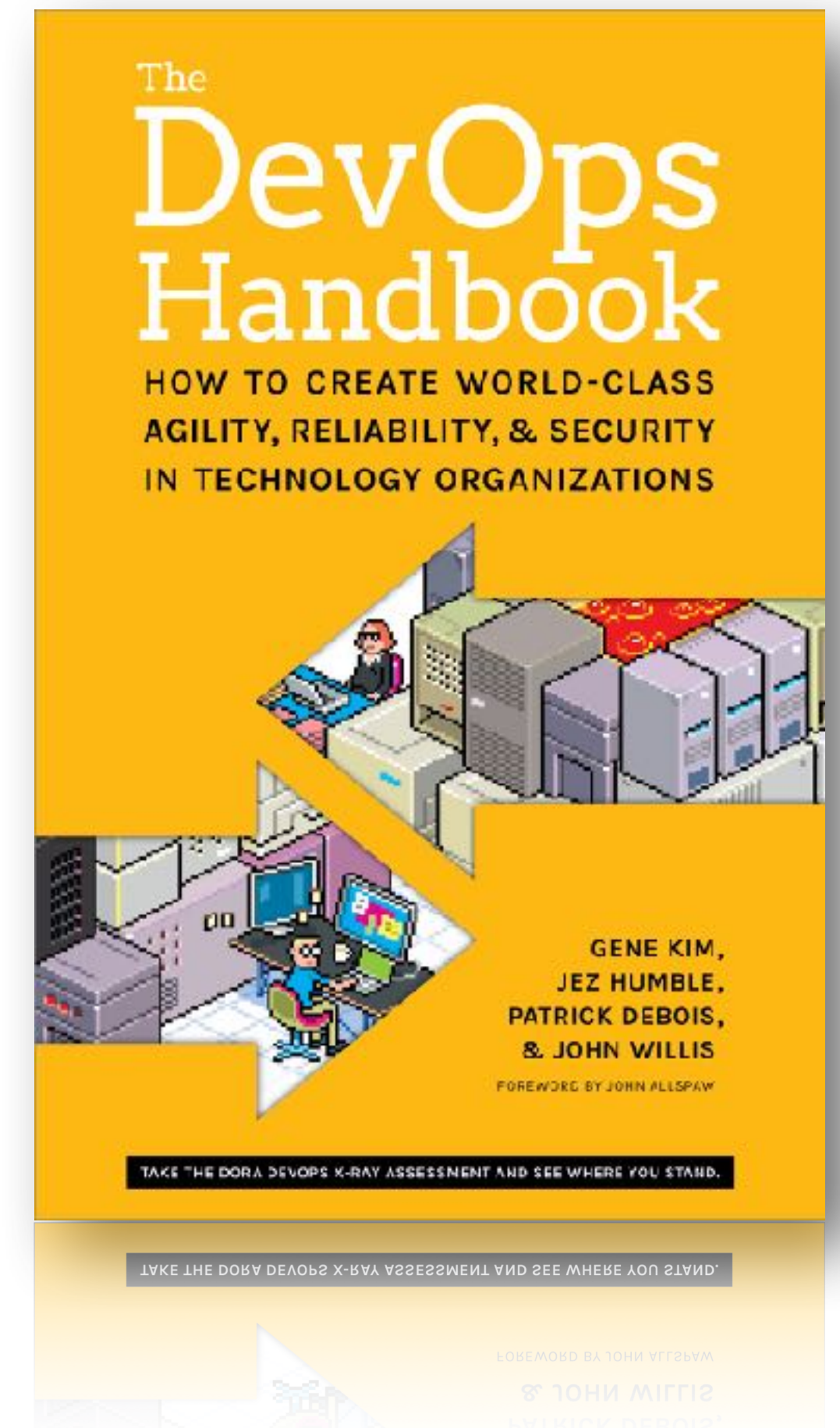
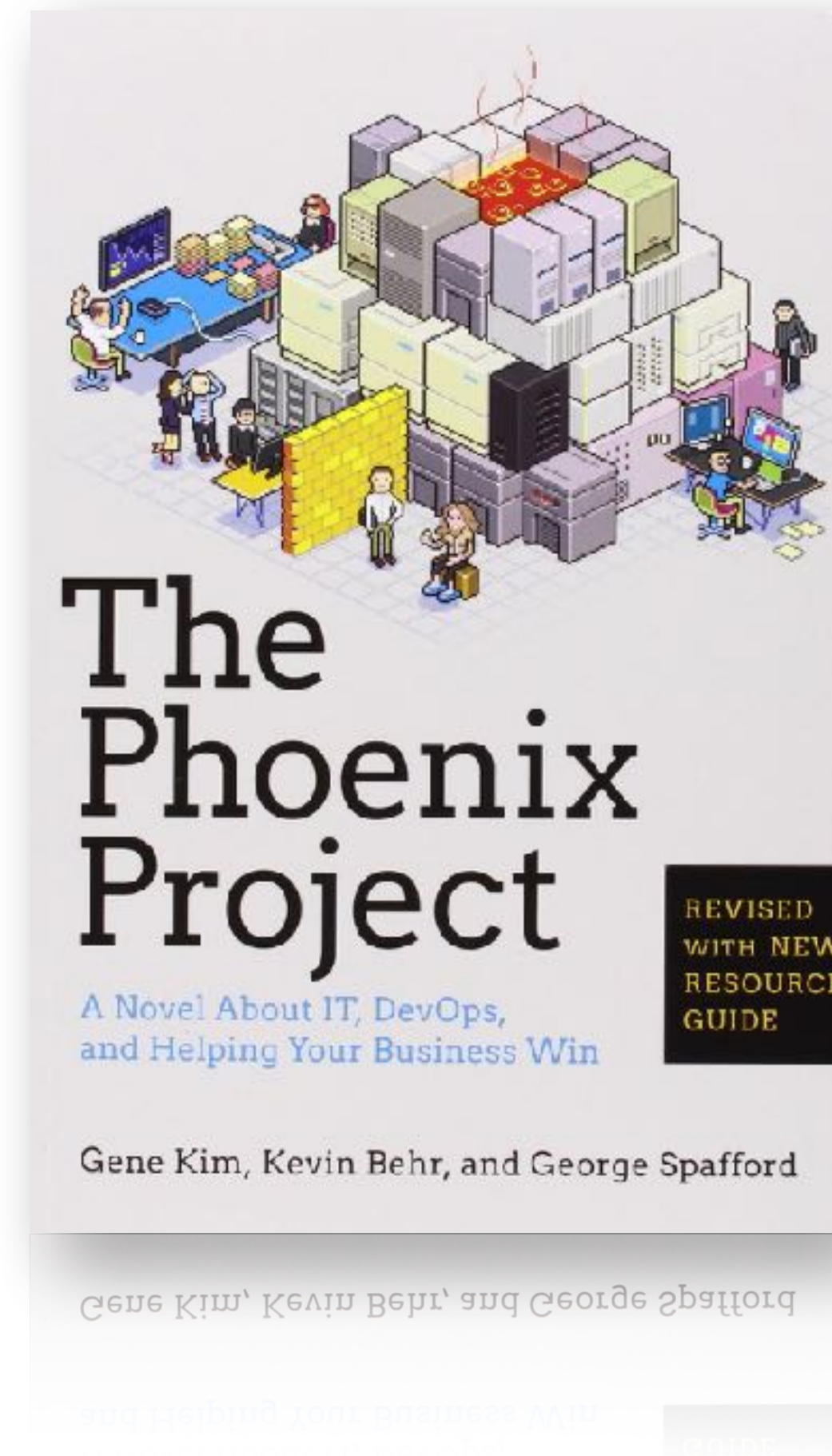
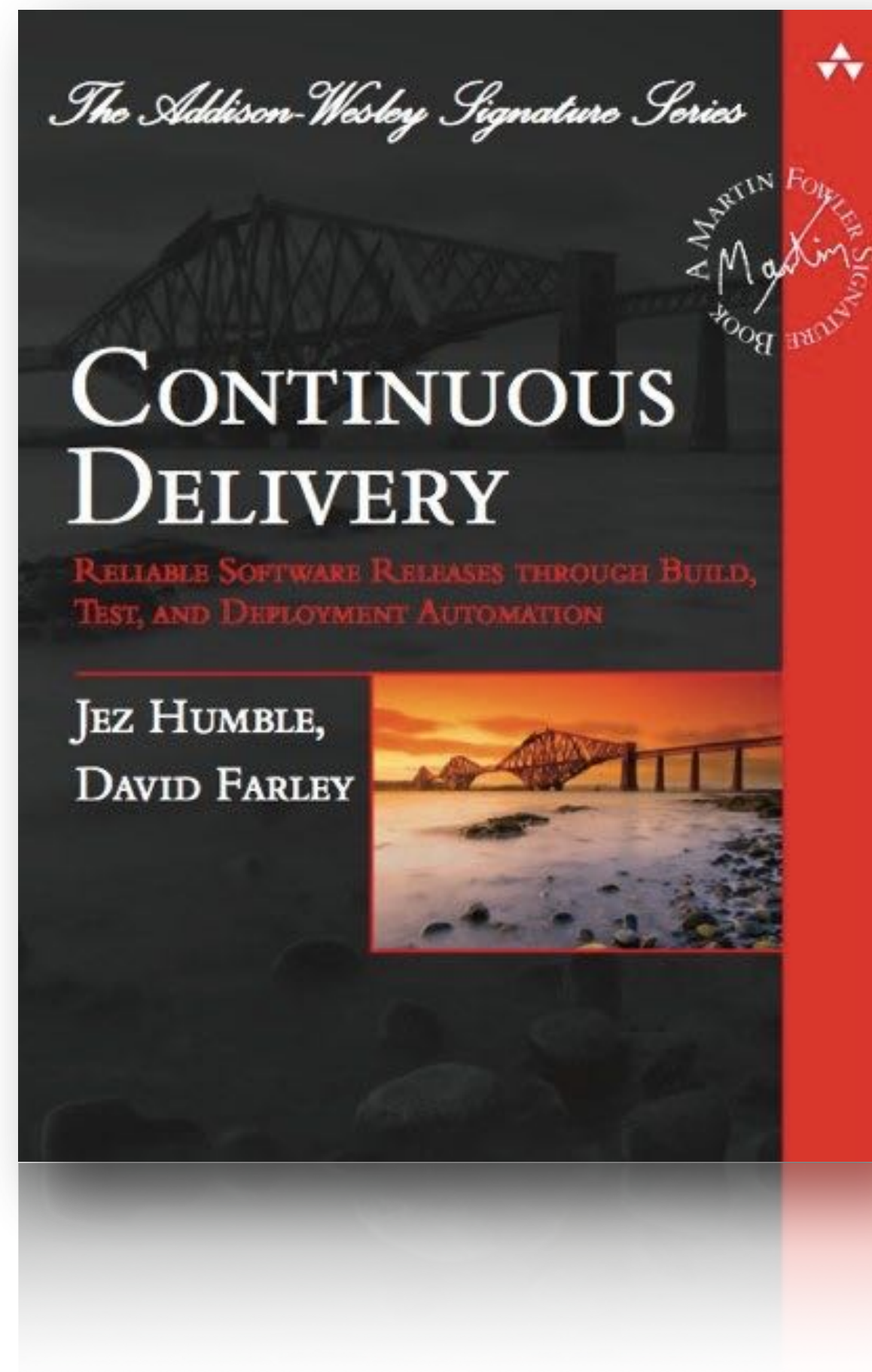
GENERATIONS

GENERATIONS = CYCLE TIME

Time taken to get a simple
change into production

repeatably
reliably

GENERATIONS = CYCLE TIME



GENERATIONS = CYCLE TIME

Continuous Integration

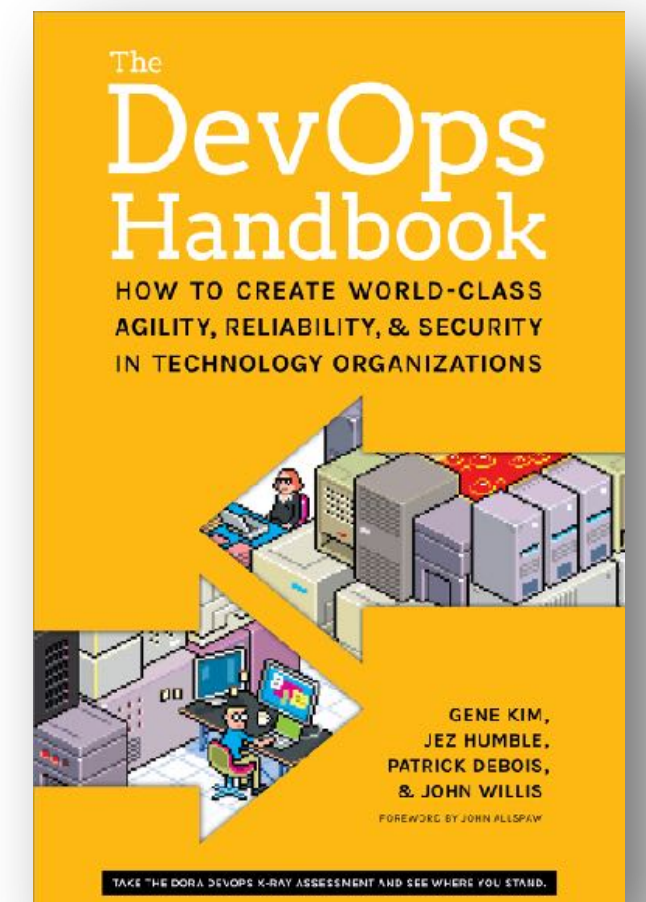
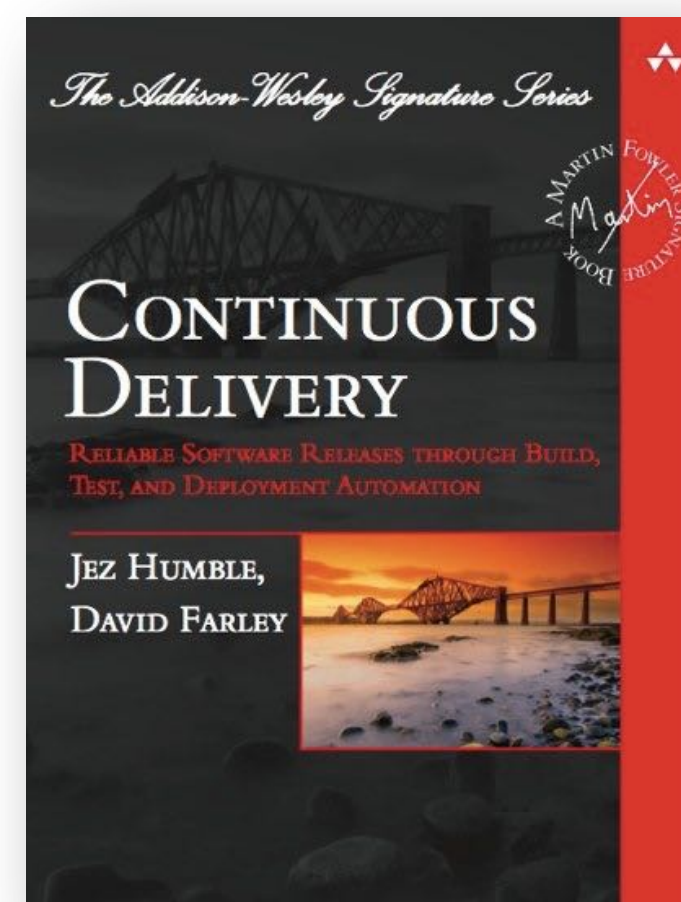
Automate everything

Keep everything in source control

“Done” means released (into production)

Shared release responsibility

Improve continuously



BUILDING ARCHITECTURE

Architect  Develop  Release

WHAT IS ARCHITECTURE?

WHAT IS ARCHITECTURE?

“Software architecture is the **decisions** which are both important and hard to change.”

- Martin Fowler

Source: <https://www.youtube.com/watch?v=DngAZyWMGR0>

WHAT IS ARCHITECTURE?

“...architecture is about **understanding** what you need to build, creating a vision for building it and making the appropriate **design decisions**”

- Simon Brown

Source: Architecture for Software Developers

WHAT IS ARCHITECTURE?

“Architecture is the **decisions**
that you **wish you could get right** early in a project”

- Ralph Johnson

Source: Who needs an Architect? (Fowler 2003)

Architect



Develop



Release



Make important
decisions here



Easy stuff
here



How hard is
that?

What appears important
here may not be important



Architect

~~Make important
decisions here~~



We learn a lot more
about the problem space
and false assumptions



Develop

~~Easy stuff
here~~



Much harder than we
thought!



Release

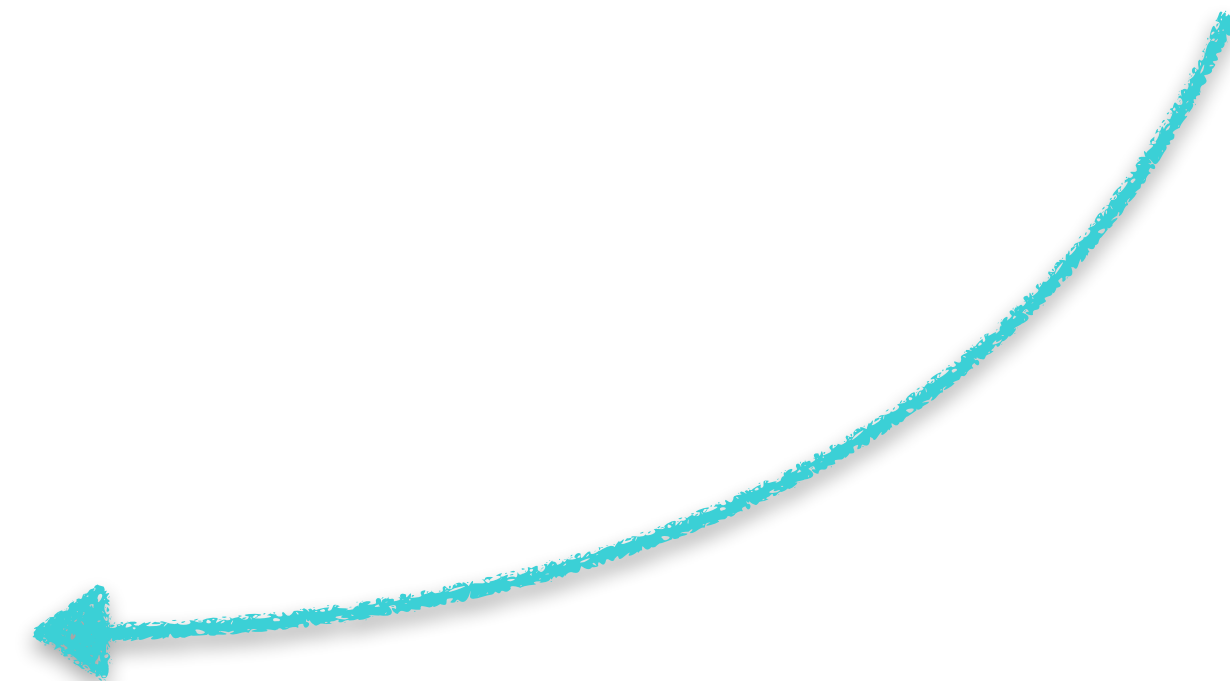
~~How hard is
that?~~

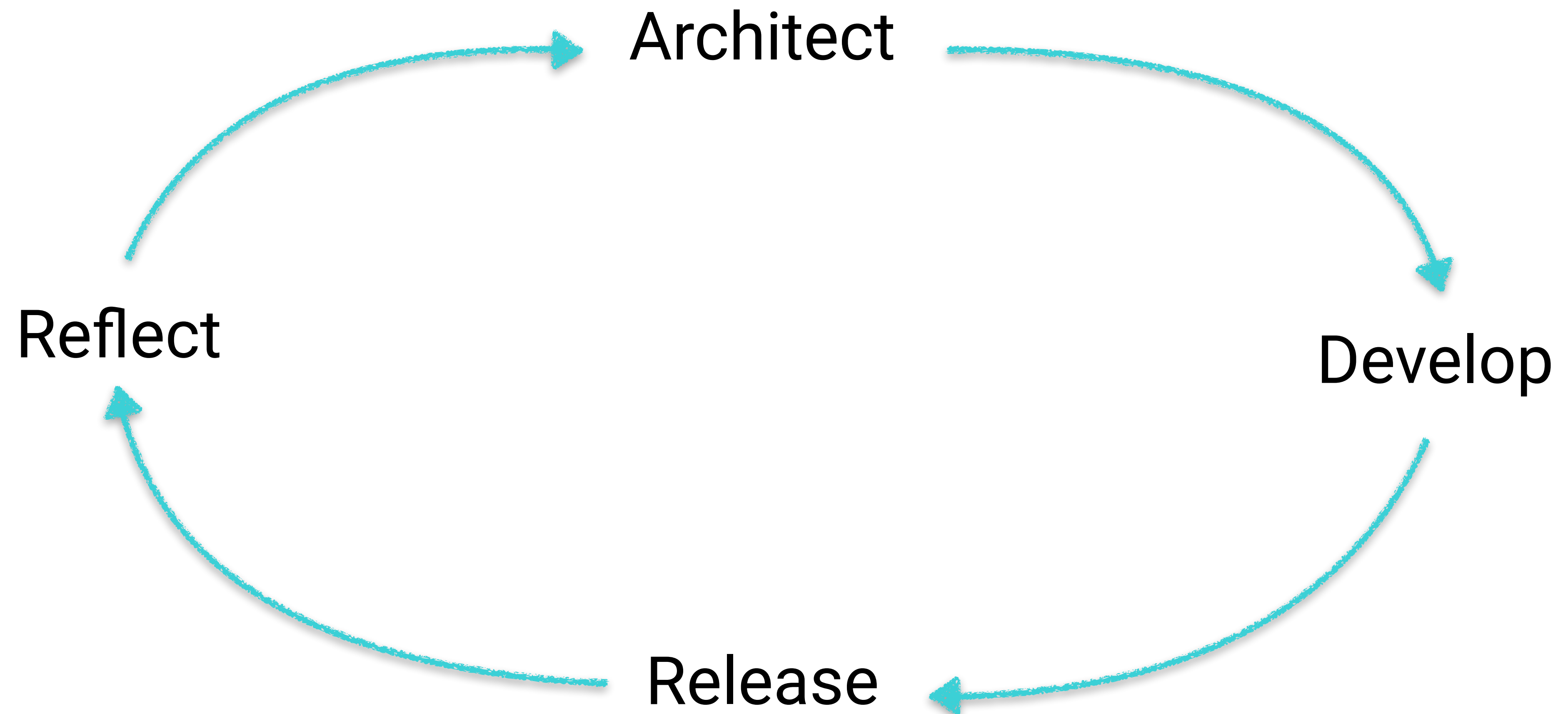


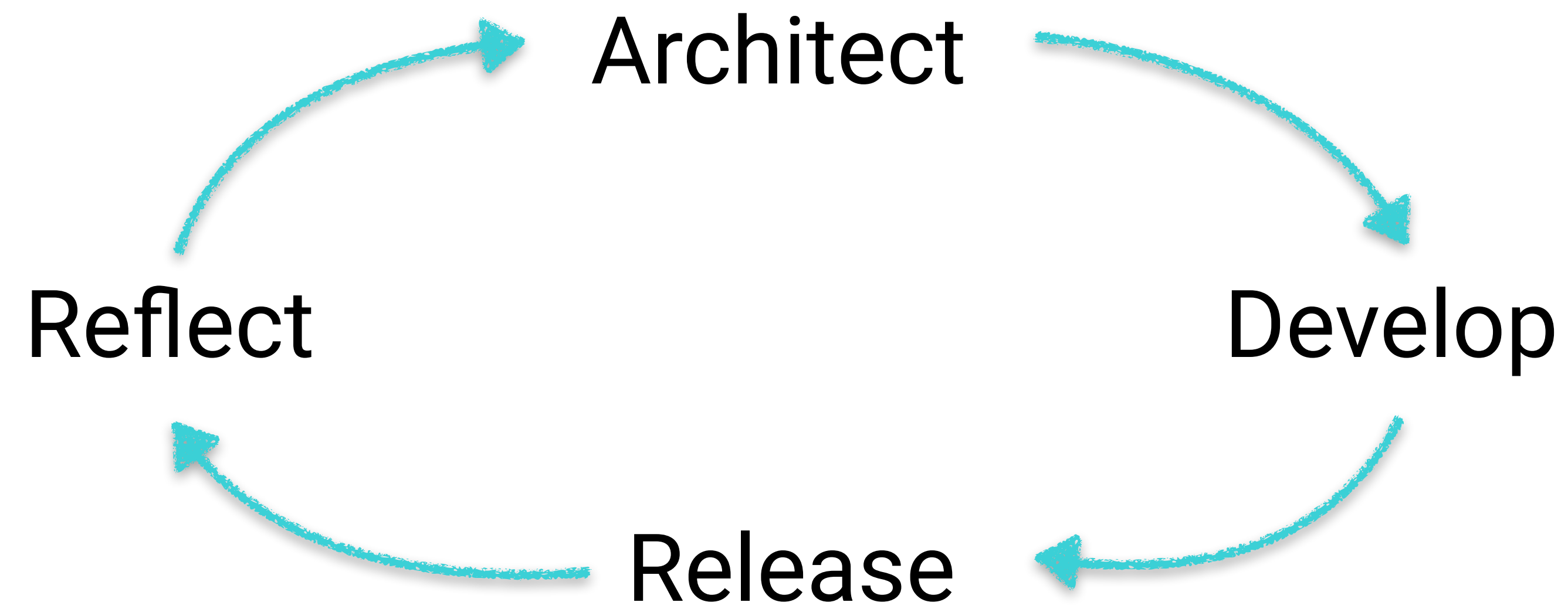
Architect

Develop

Release







Cycle time = constraint

DEFINITION

An evolutionary architecture supports **incremental, guided** change as a first principle along **multiple dimensions**



Evolutionary architectures
are **guided** with
FITNESS FUNCTIONS

FITNESS FUNCTIONS

“An objective function that measures how close a given solution fits to a particular goal”

FITNESS FUNCTIONS

NFRs
CFRs
Quality Attributes

IMPORTANT

UNIMPORTANT

Strong audit trail

Large # of users

Low response time

Heavy legal compliance

Availability

Mobile responsive

Internationalisation & Localisation

Monitoring

FITNESS FUNCTIONS

NFRs
CFRs
Quality Attributes

IMPORTANT

Strong audit trail

Large # of users

Mobile responsive

Availability

UNIMPORTANT

Low response time

Monitoring

Internationalisation & Localisation

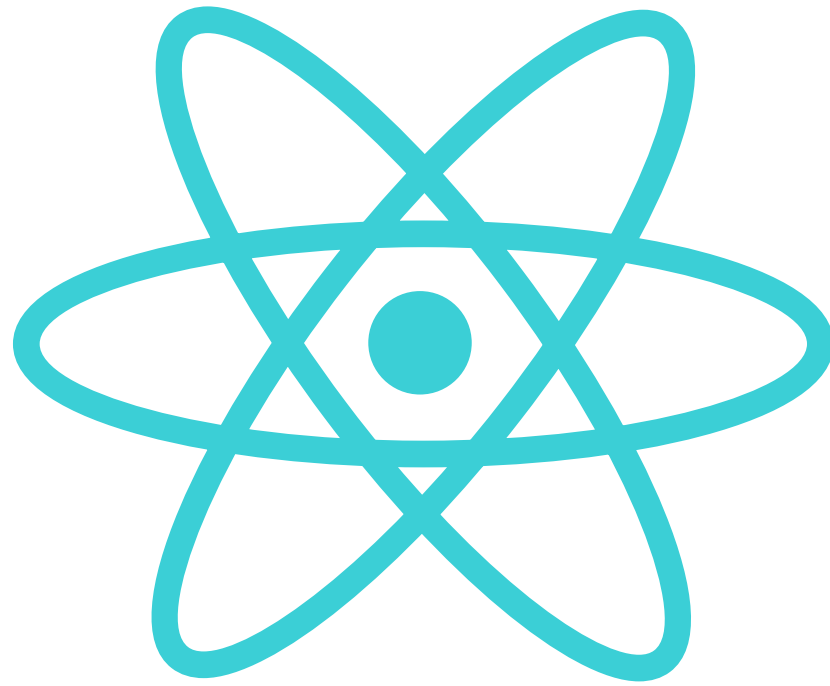
Heavy legal compliance

FITNESS FUNCTIONS

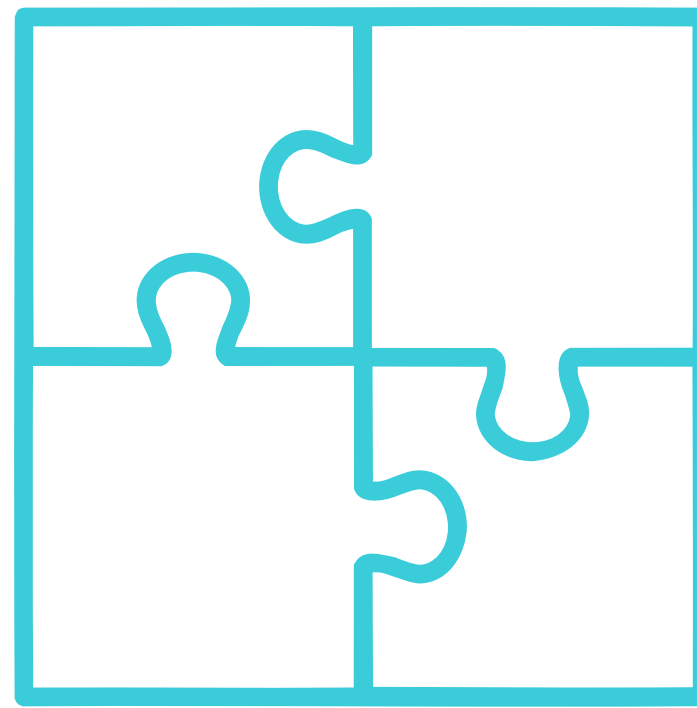
“An objective function that measures how close a given solution fits to a particular goal”

Metrics, Tests and Process

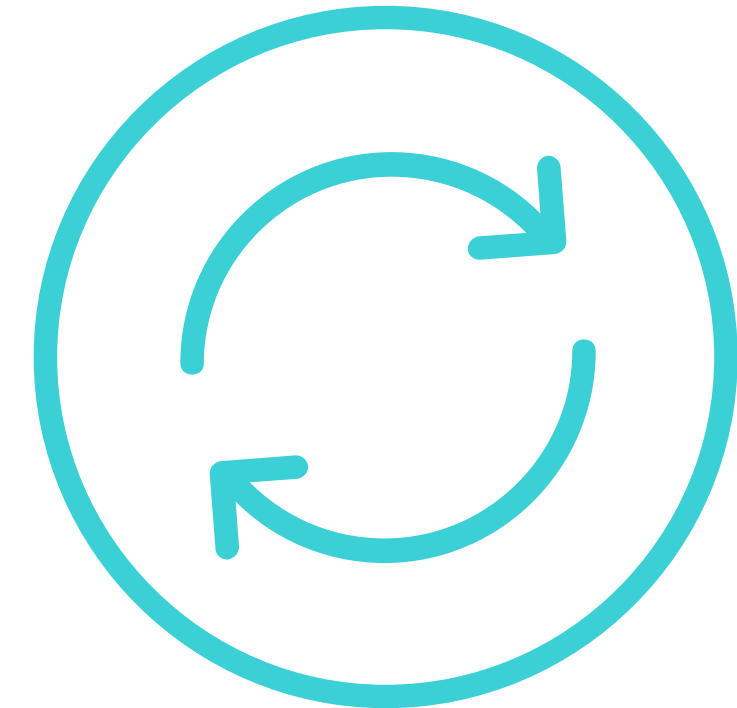
FITNESS FUNCTIONS



atomic



holistic

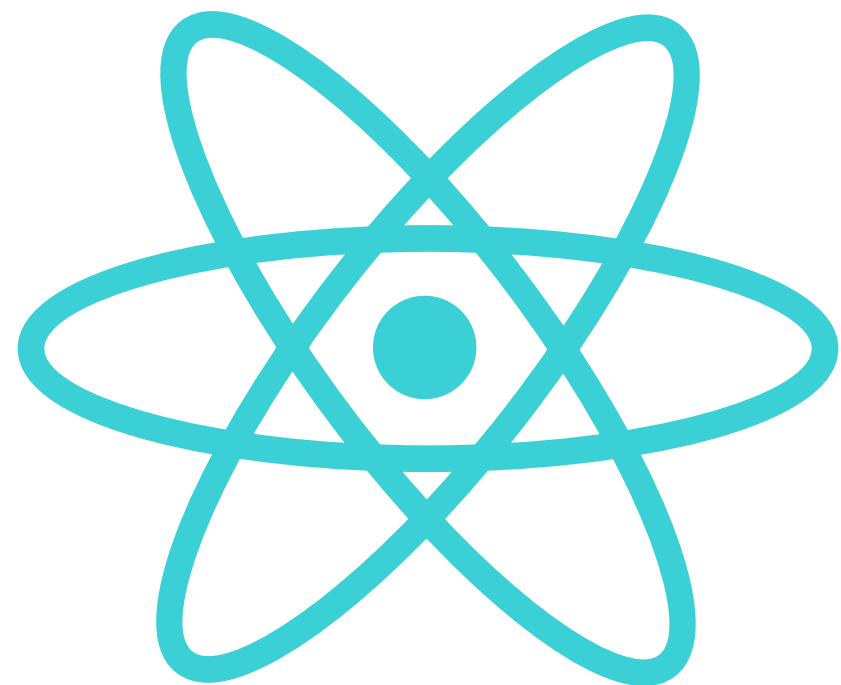


continuous

Metrics, Tests and Process

FITNESS FUNCTIONS

```
/**  
 * Ensure codebase does not contain cyclic dependencies  
 */  
public void testAllPackages() {  
    Collection packages = jDepend.analyze();  
    assertFalse("Cycles exist", jDepend.containsCycles());  
}
```

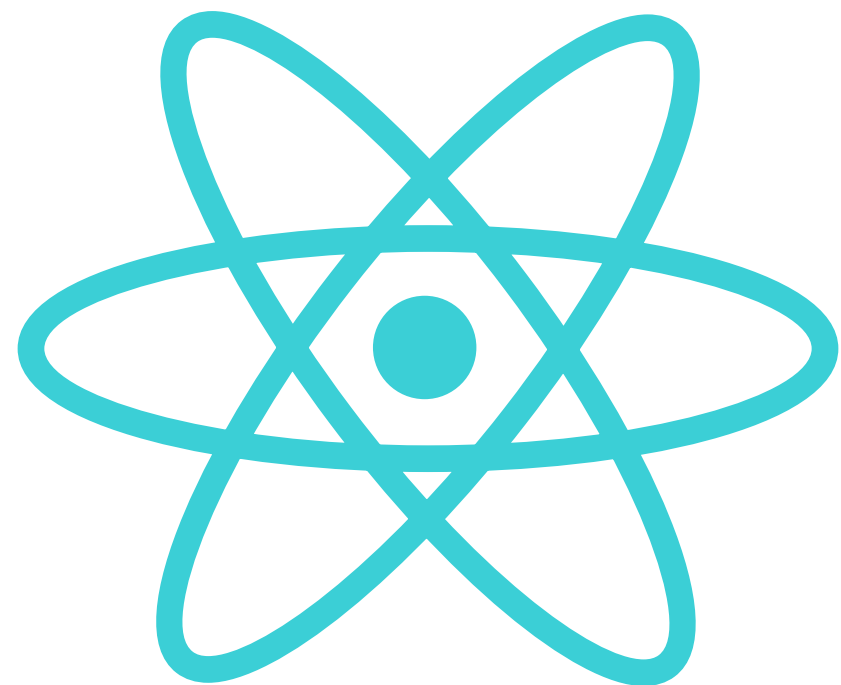


atomic

Source: <https://github.com/clarkware/jdepend>

FITNESS FUNCTIONS

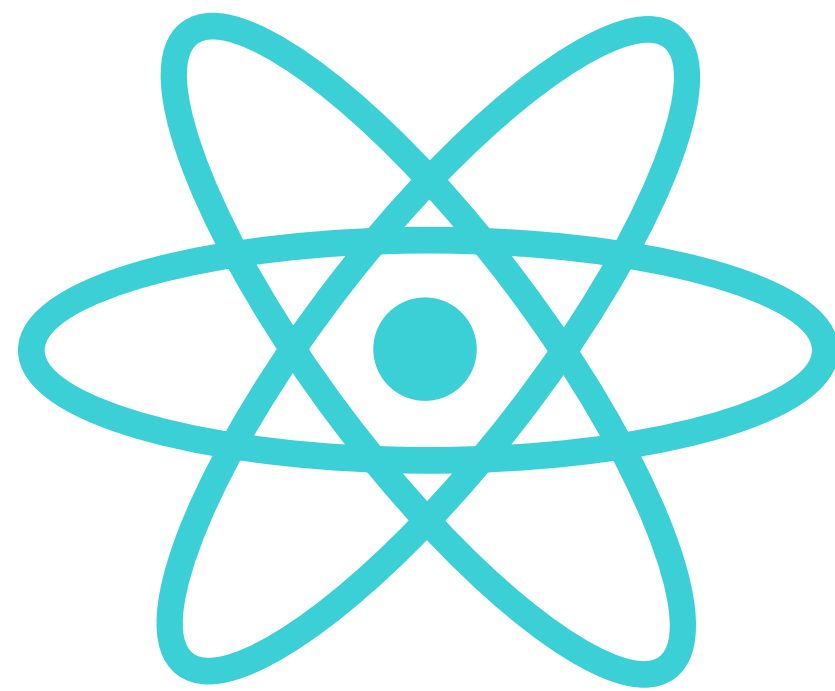
```
public void testAllPackages() throws Exception {  
    JDepend jDepend = buildNewJDepend();  
    DependencyConstraint constraint = new DependencyConstraint();  
    JavaPackage web = constraint.addPackage("com.thekua.web");  
    JavaPackage util = constraint.addPackage("com.thekua.util");  
    JavaPackage repository = constraint.addPackage("com.thekua.dao");  
  
    web.dependsOn(util);  
    repository.dependsOn(util);  
    web.dependsOn(repository);  
  
    jDepend.analyze();  
  
    assertTrue("Dependency mismatch", jDepend.dependencyMatch(constraint));  
}
```



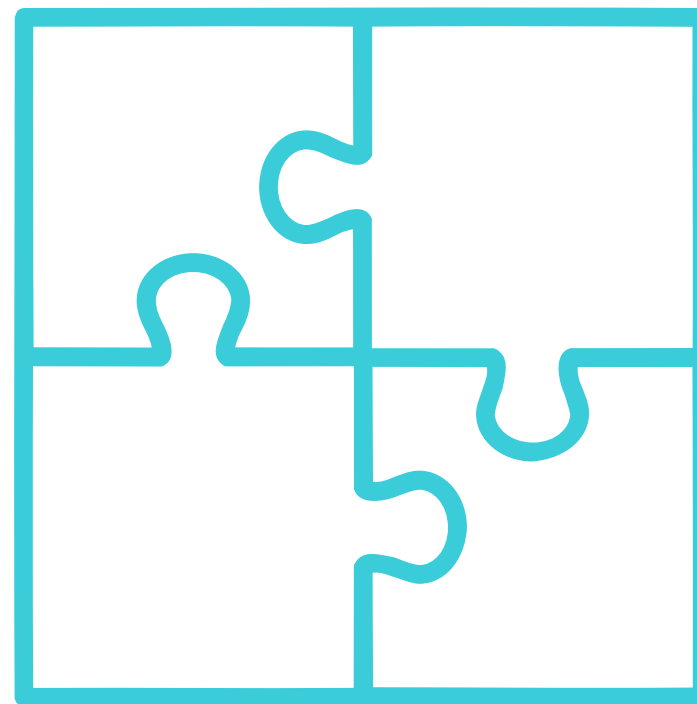
atomic

Source: <https://github.com/clarkware/jdepend>

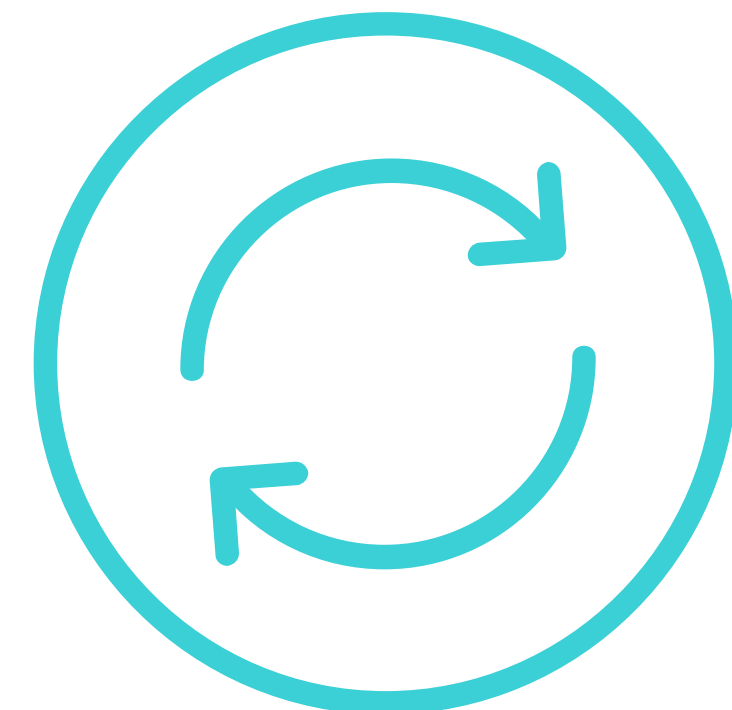
FITNESS FUNCTIONS



atomic



holistic



continuous

FITNESS FUNCTIONS

Performance was particularly critical

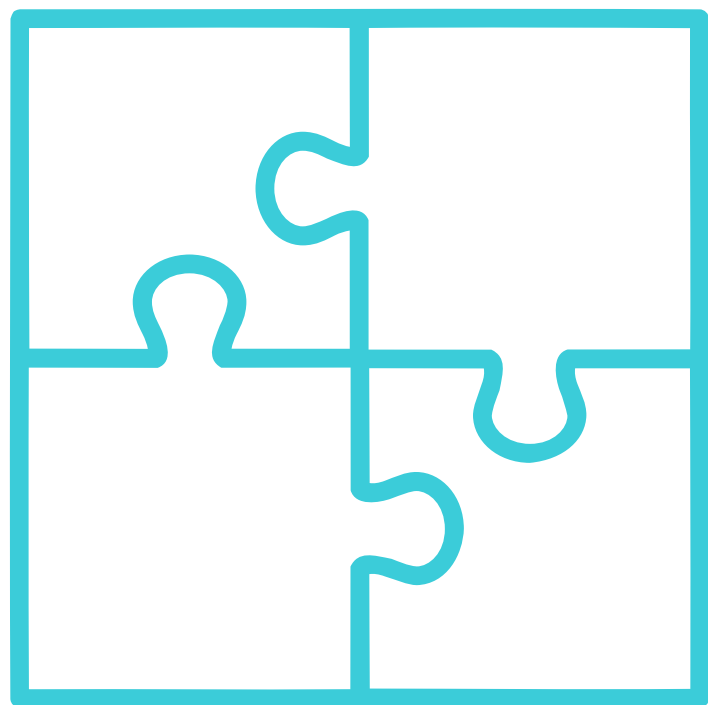
Build a CD pipeline of performance tests

Sanity performance test

Memory soak test

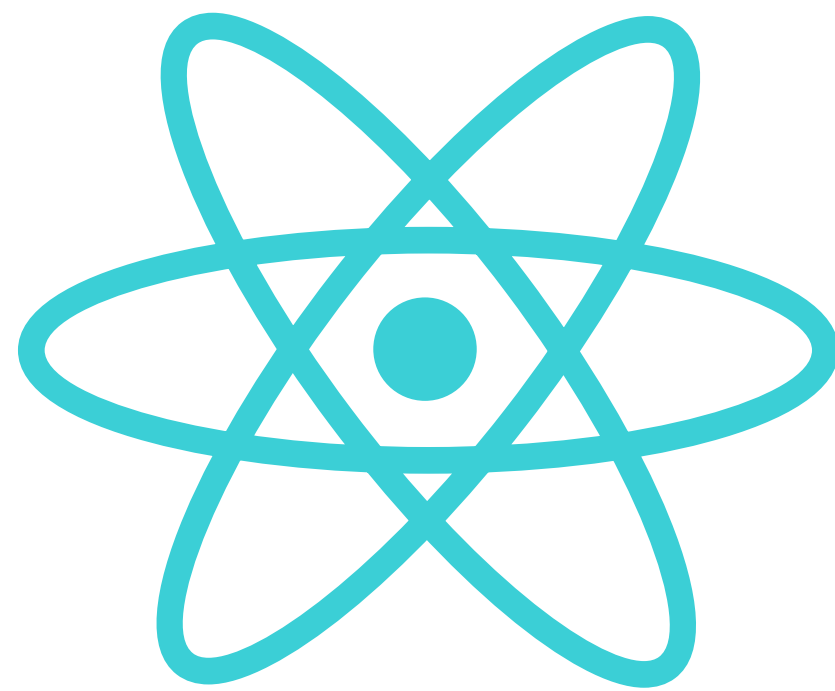
Detected unusual behaviour in a test run

Found that disk controller had been misconfigured

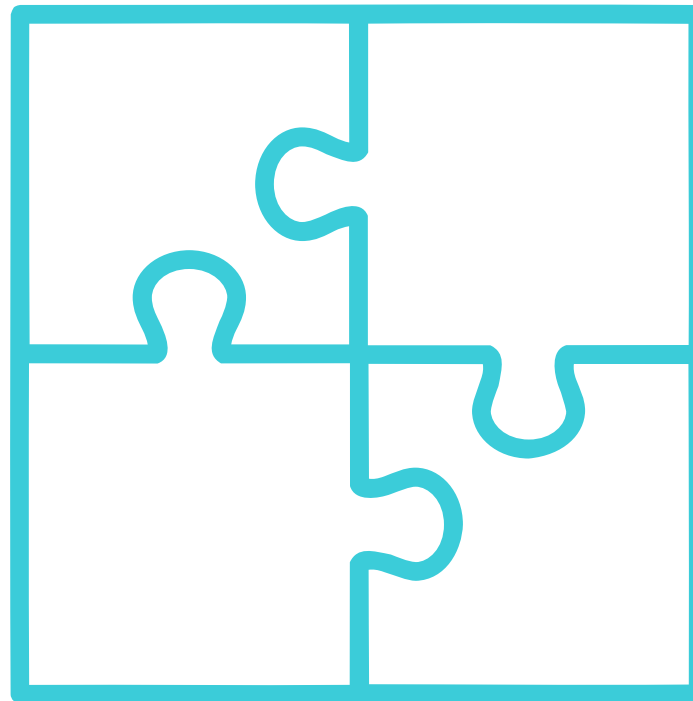


holistic

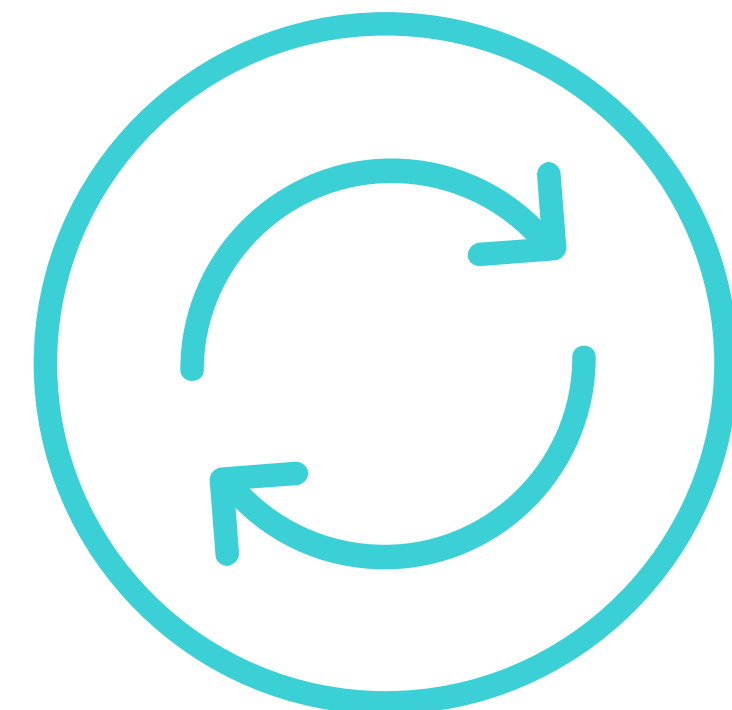
FITNESS FUNCTIONS



atomic

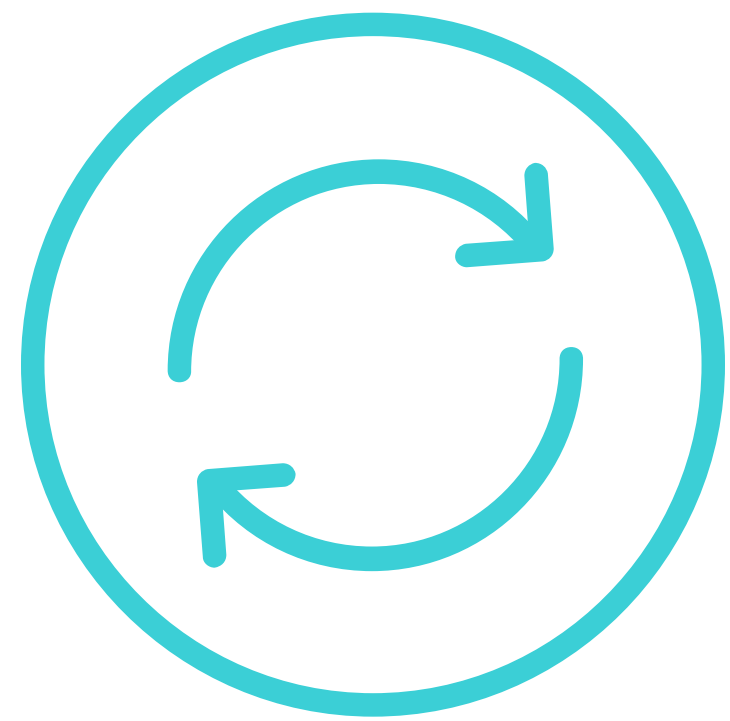


holistic



continuous

FITNESS FUNCTIONS



continuous



FITNESS FUNCTIONS & CD

“If it hurts, do it more often”

- Continuous Delivery

Metrics, Tests and Process

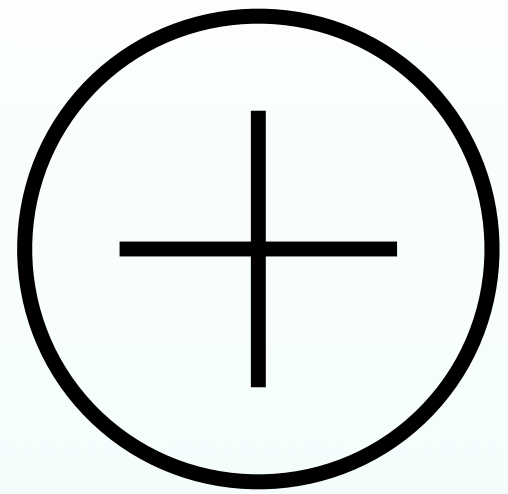
DEFINITION

An evolutionary architecture supports **incremental, guided** change as a first principle along **multiple dimensions**

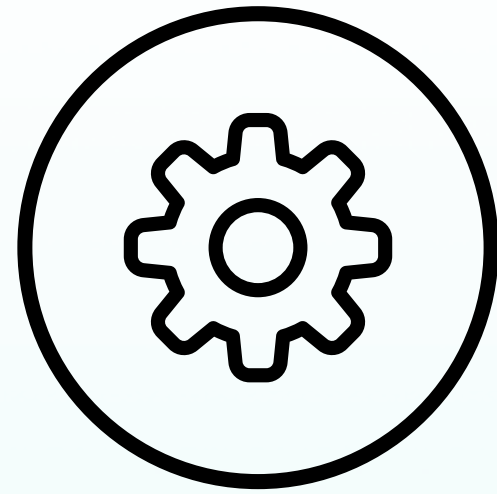


DIMENSIONS OF CHANGE

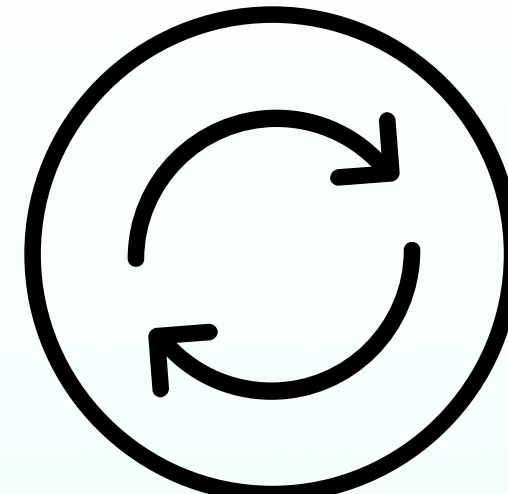
TECHNICAL CHANGES



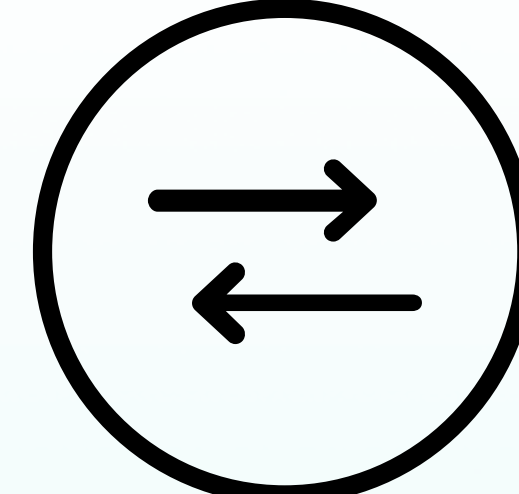
Introduce



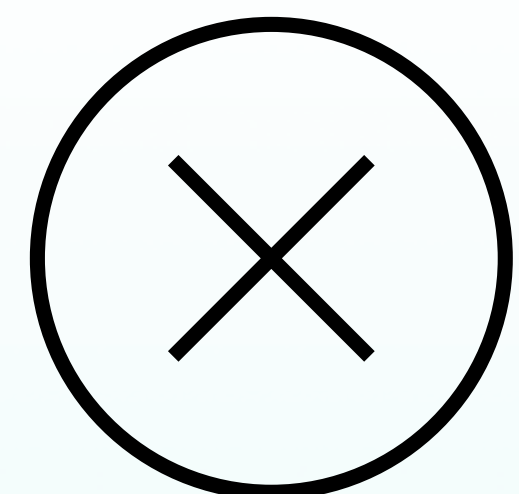
Configure



Upgrade



Replace



Remove

Library, Tool, Framework, Platform

DOMAIN CHANGES

Business
flow

Features

Conditions

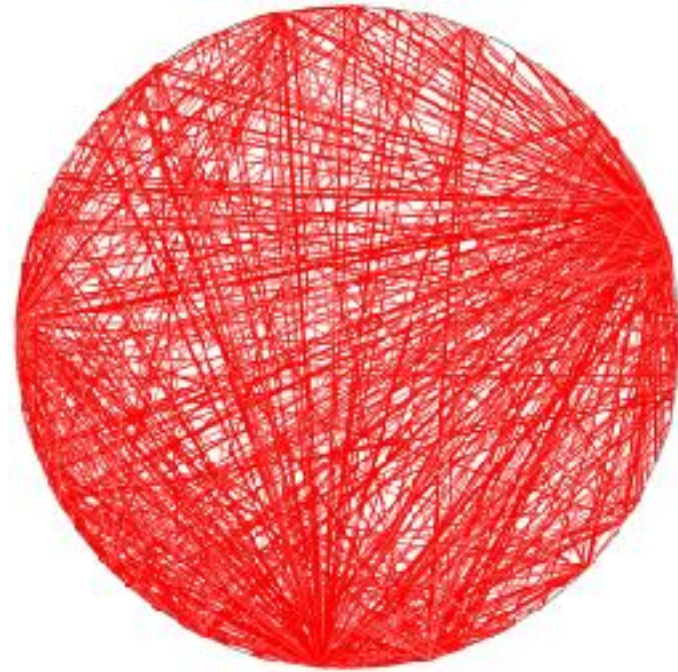
Interface
changes

New
domain

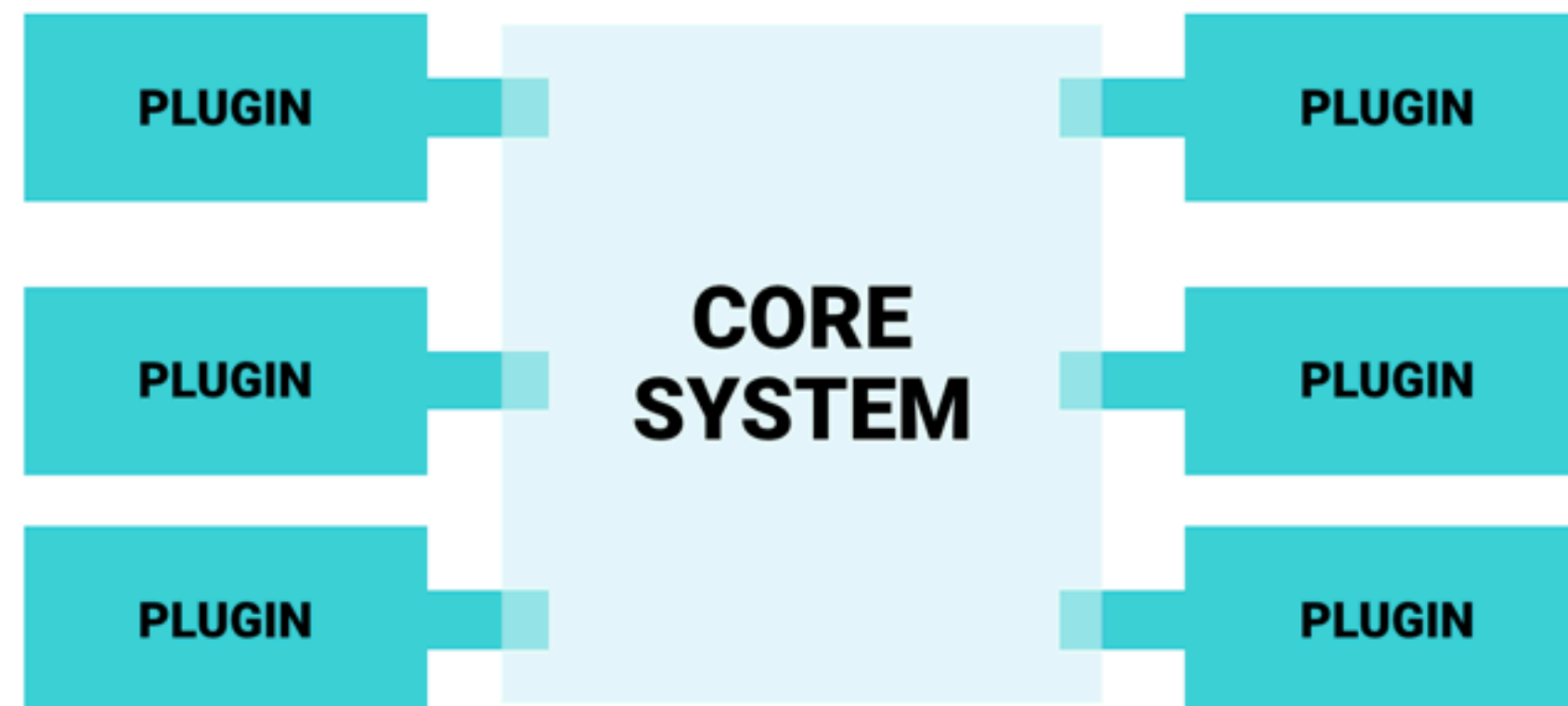
New
interfaces

New
applications

ARCHITECTURAL APPROACHES



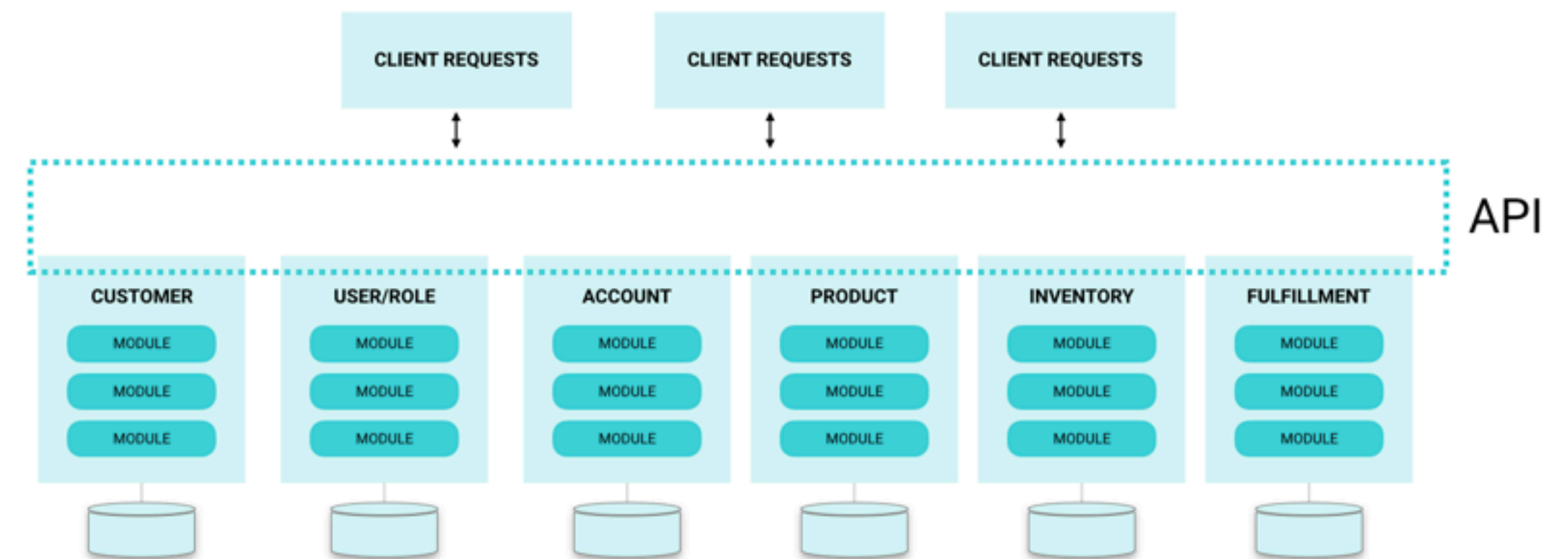
BIG BALL OF MUD



MICRO KERNEL

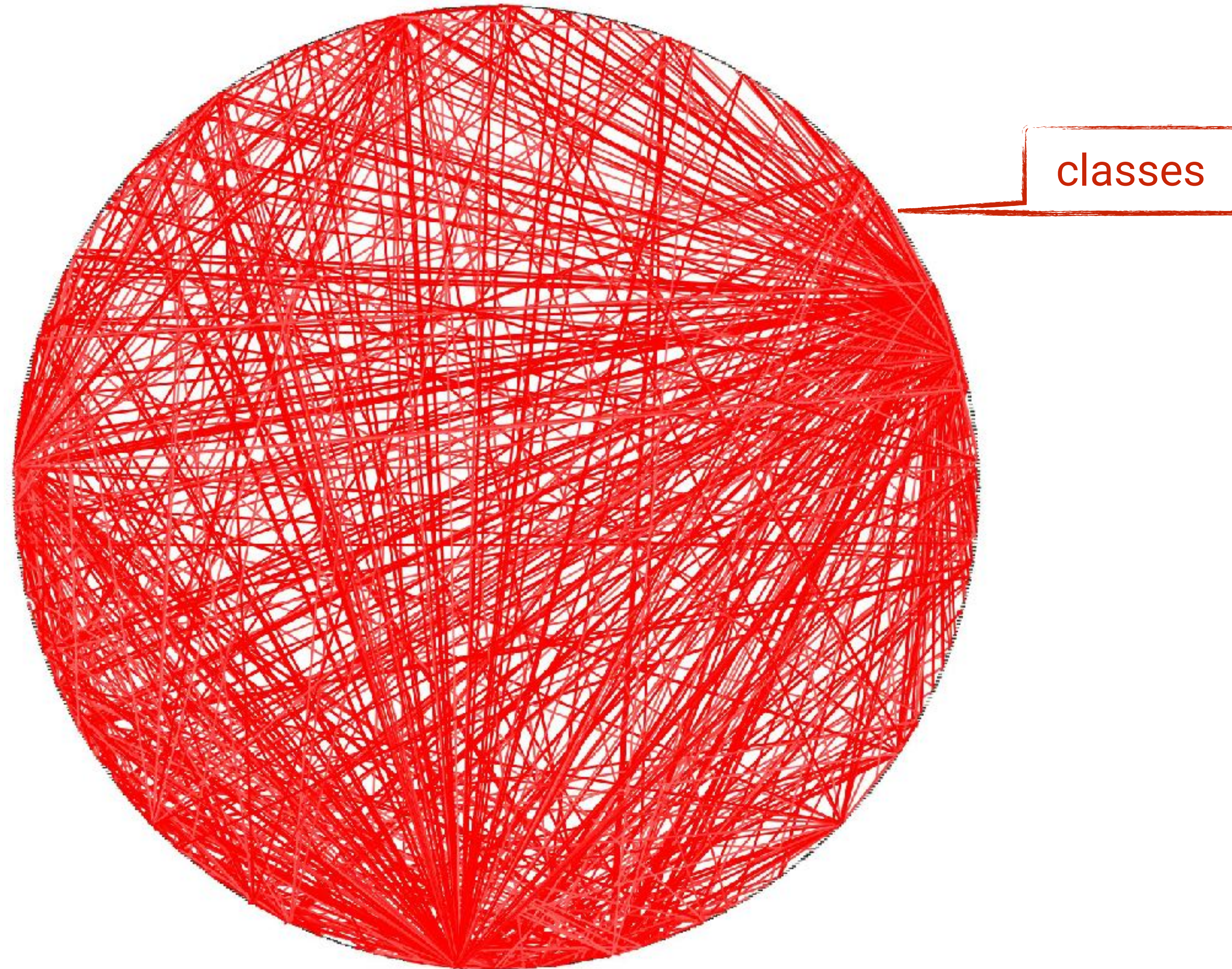


LAYERED ARCHITECTURE



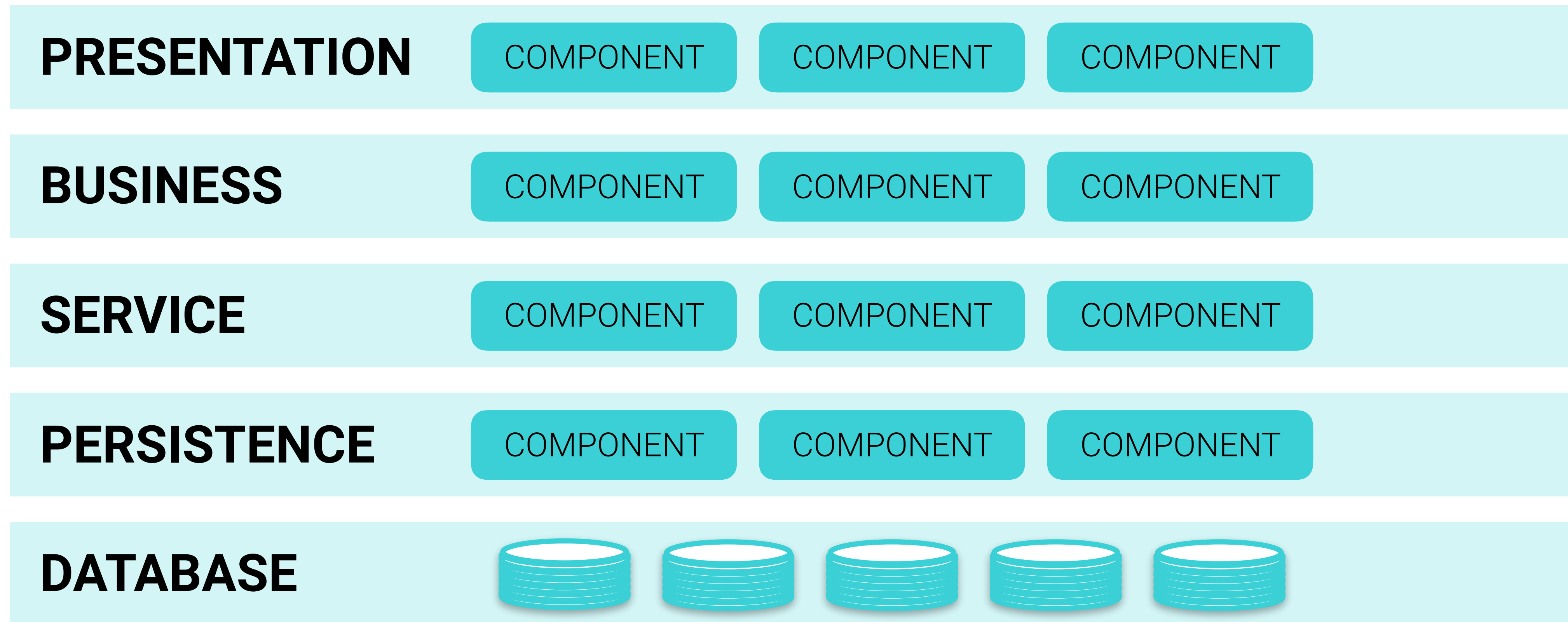
MICROSERVICES

BIG BALL OF MUD

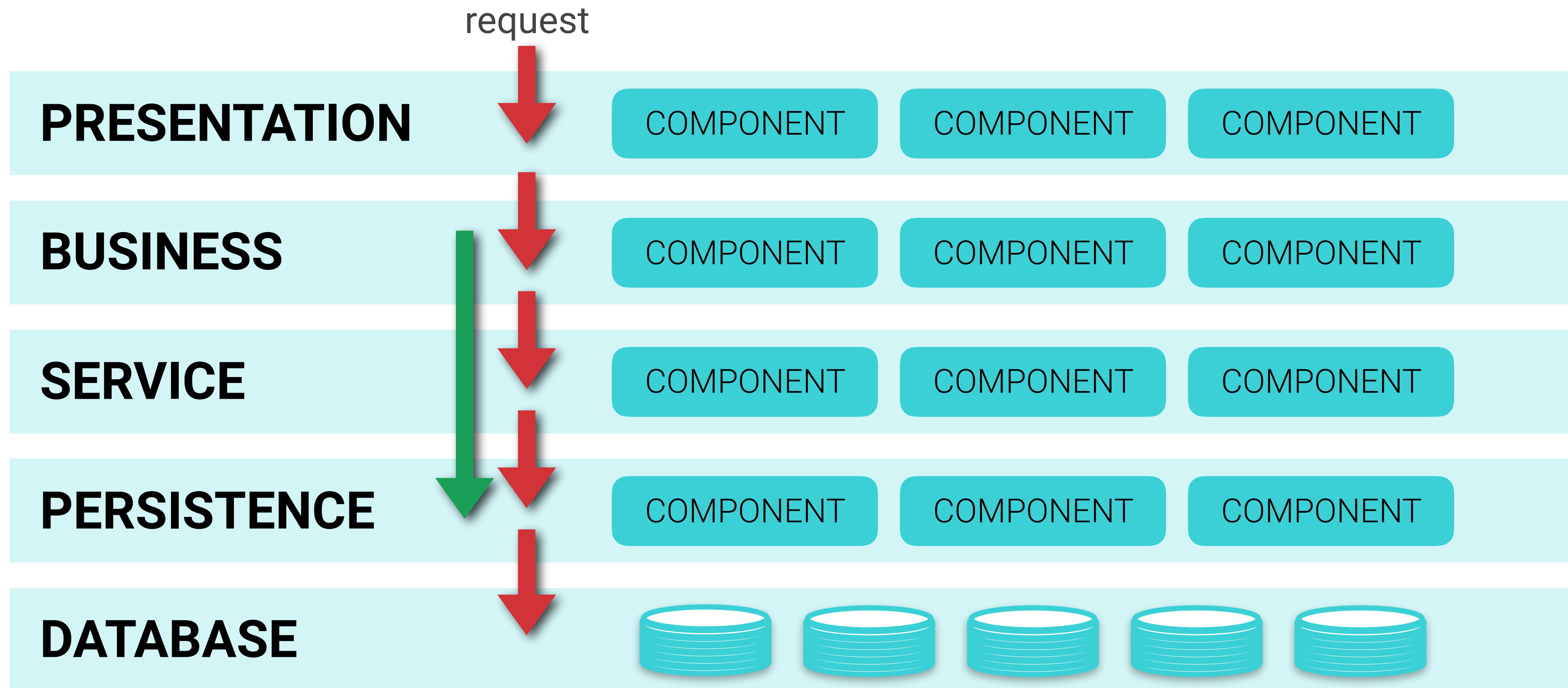


coupling connections

LAYERED ARCHITECTURES



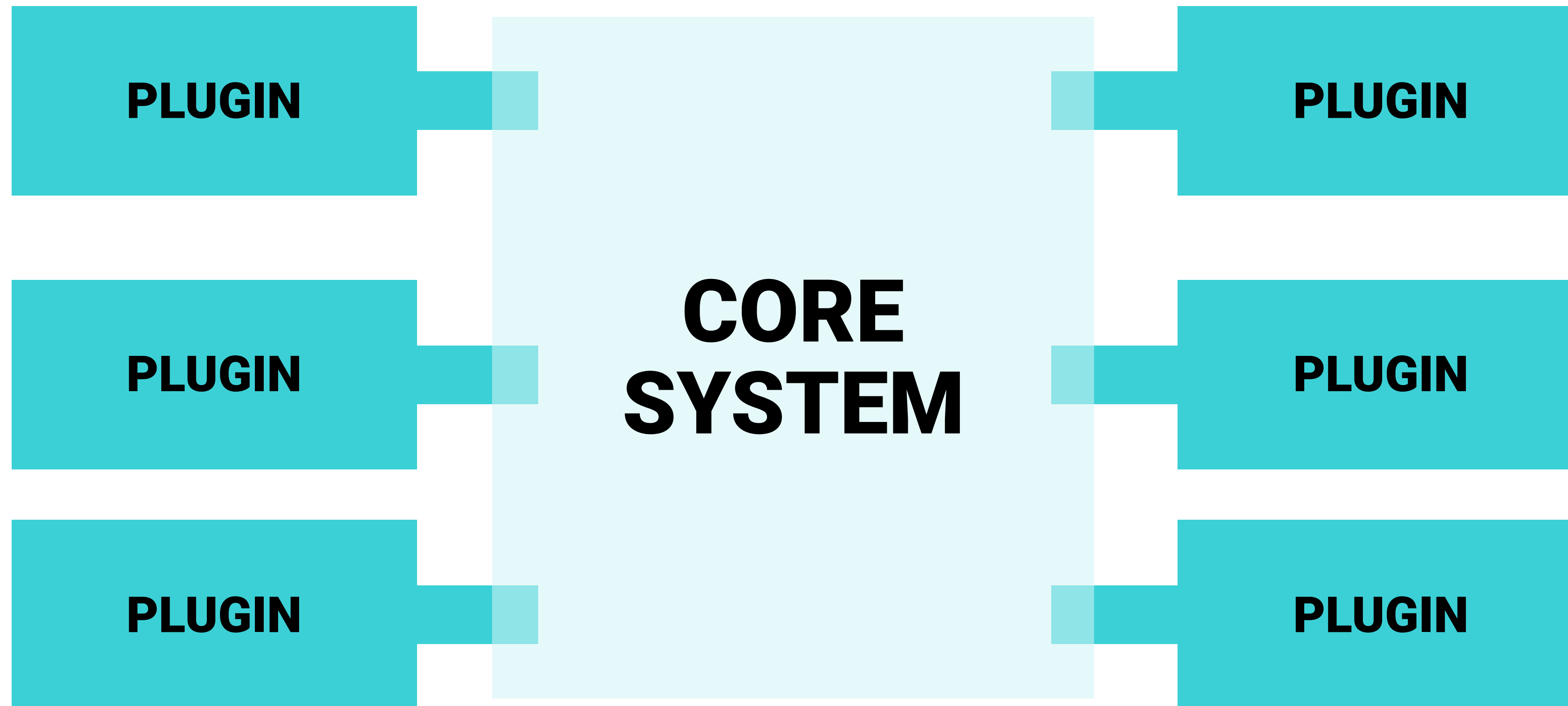
LAYERED ARCHITECTURES



LAYERED ARCHITECTURES



MICROKERNEL



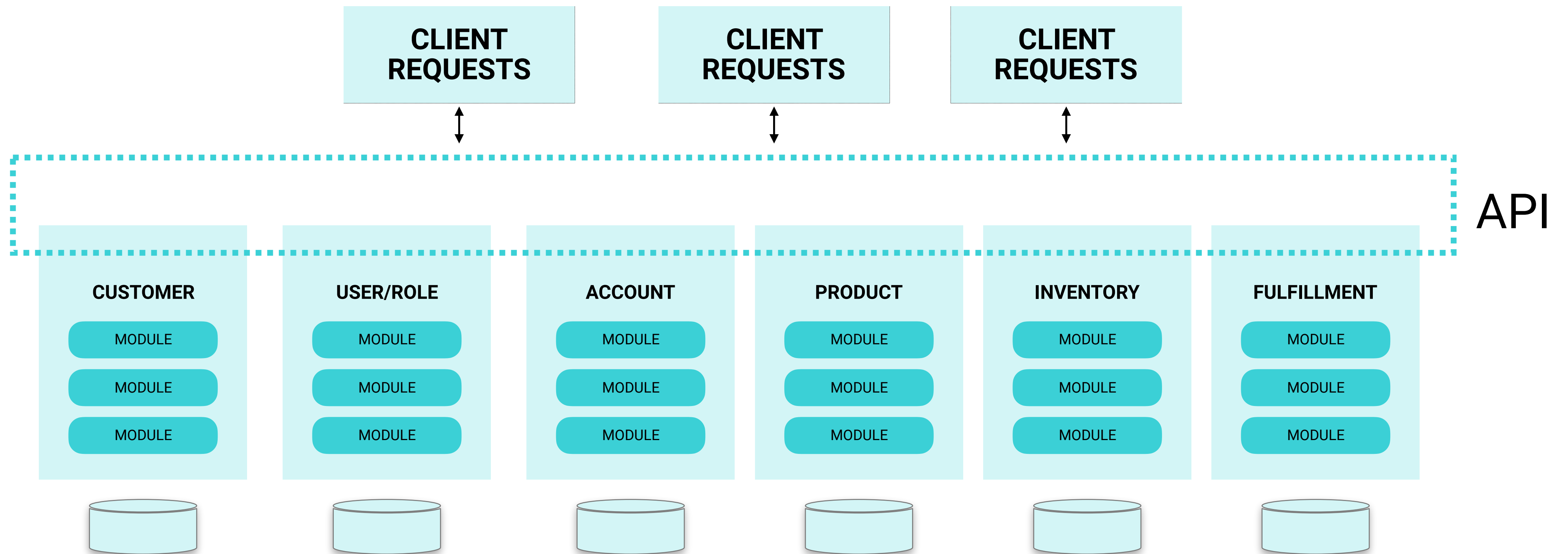
MICROKERNEL

MavenTM

 **eclipse**



MICROSERVICES



DEFINITION

An evolutionary architecture supports **incremental, guided** change as a first principle along **multiple dimensions**



PRACTICES

Technical Domain

Continuous Delivery

Support fast feedback

Appropriate coupling

Ports and Adapters

Automated fitness functions

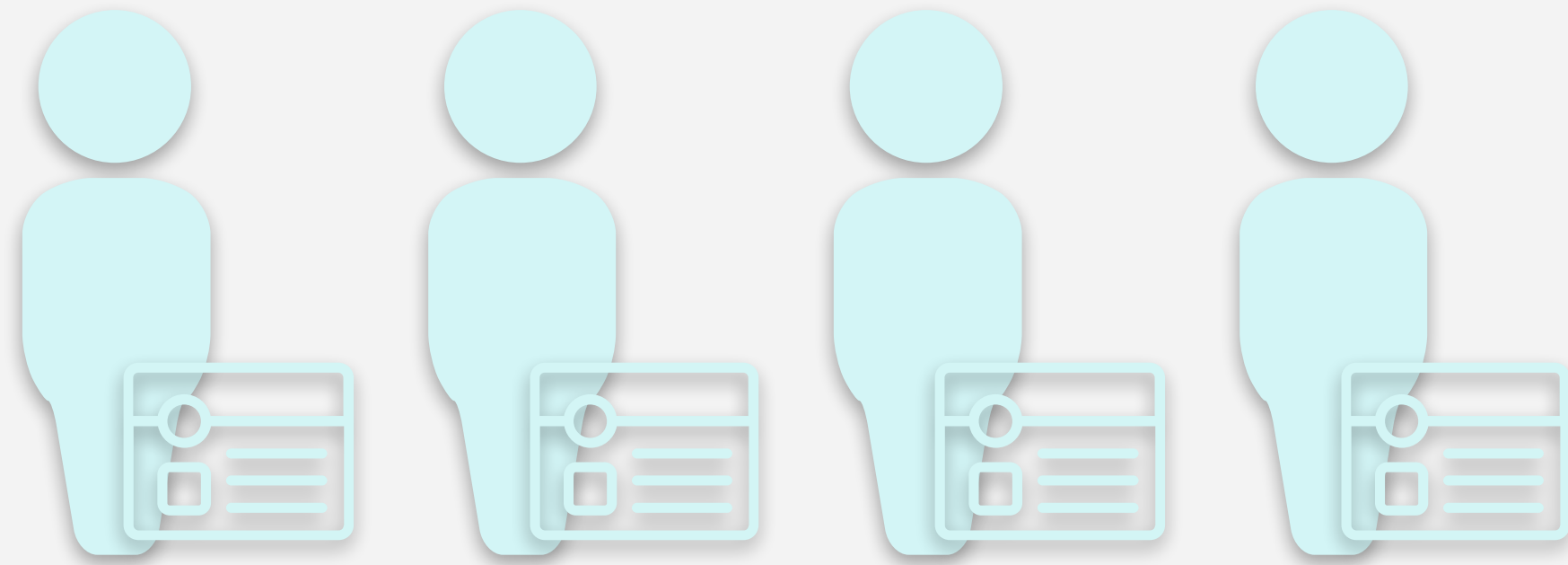
Matches business capabilities

Products over projects

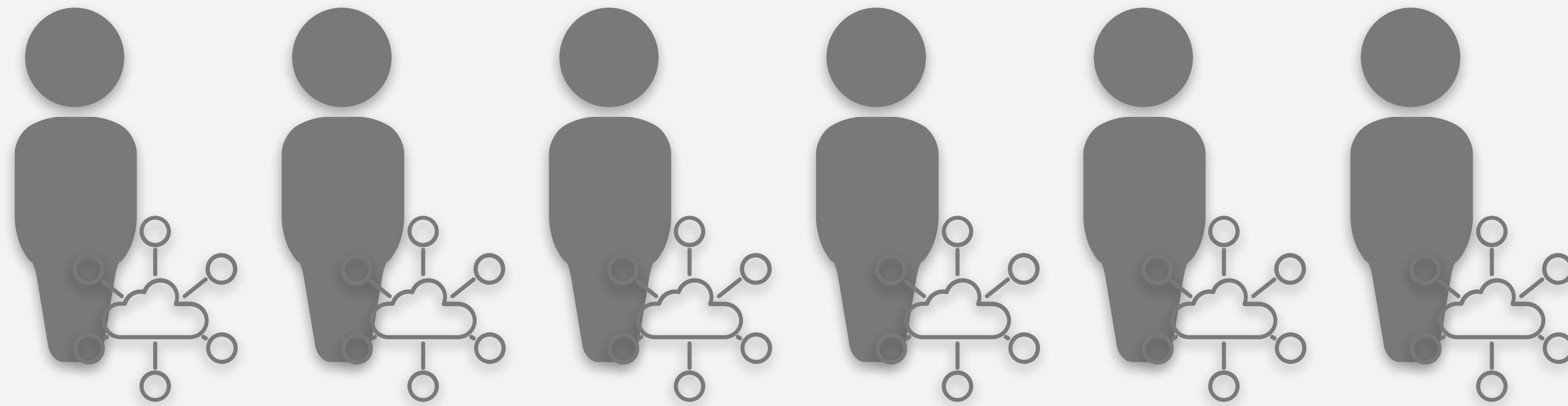
Cross functional team

Enables experimentation

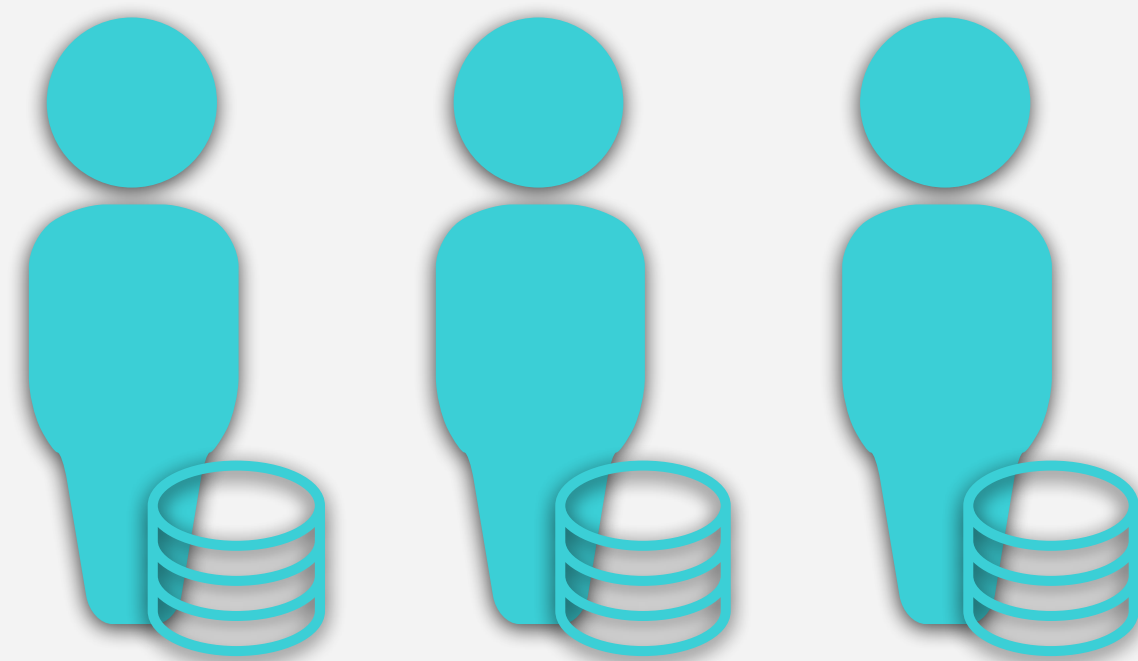
Decentralised governance



User Interface



Server-side



DBAs

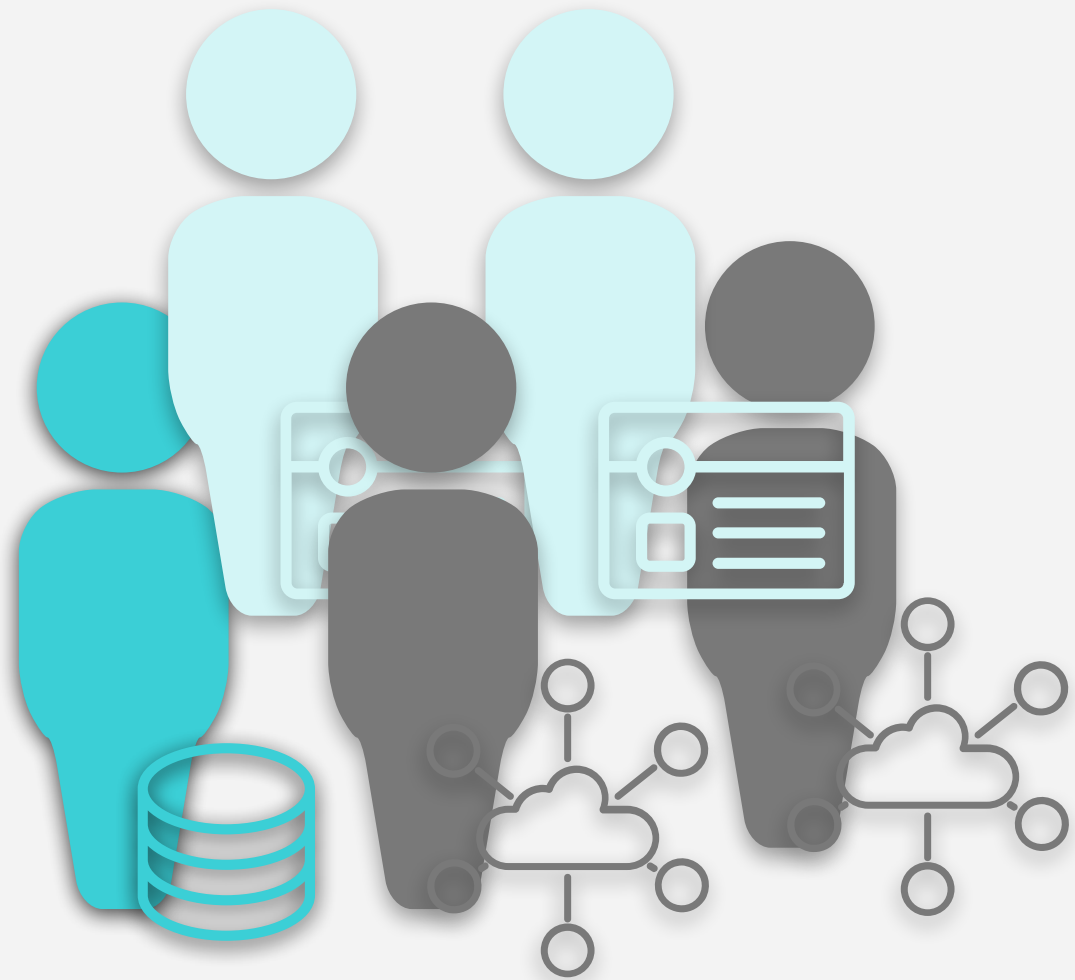
Orders



Shipping



Catalog



Inverse Conway Manoeuvre



PRINCIPLE-DRIVEN



over



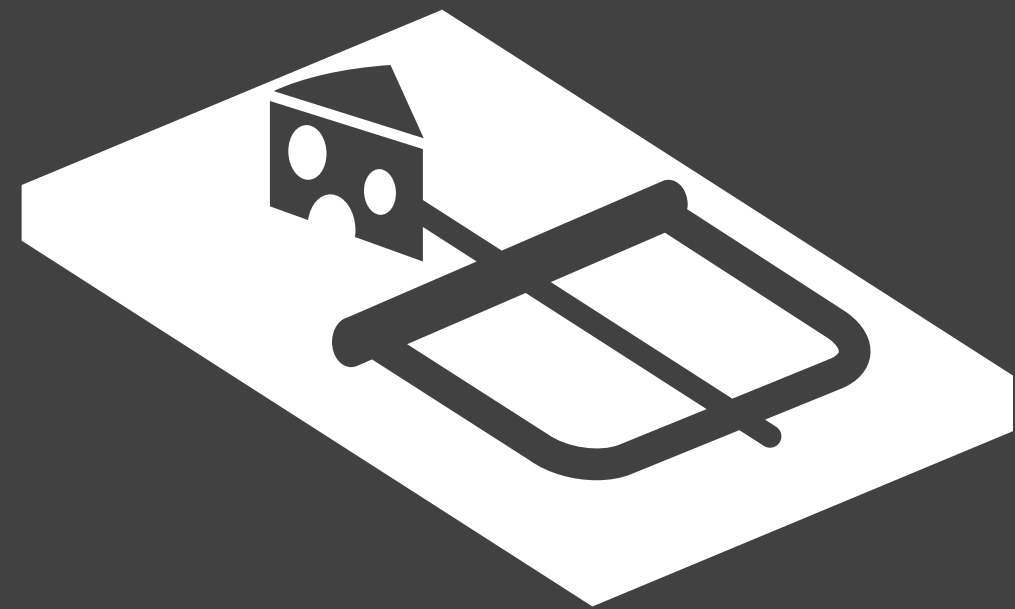
An aerial, slightly desaturated photograph of a city. A river flows through the middle of the city, curving to the right. To the right of the river is a large, open, light-brown field, possibly a park or sports field. The city is densely packed with buildings of various sizes and colors, mostly in shades of beige, brown, and grey. There are green spaces and trees interspersed among the buildings. The overall tone is muted and professional.

Think like a town planner

If a lot about architecture is about
making decisions, help people
make better decisions.

Avoid making the decisions for them.

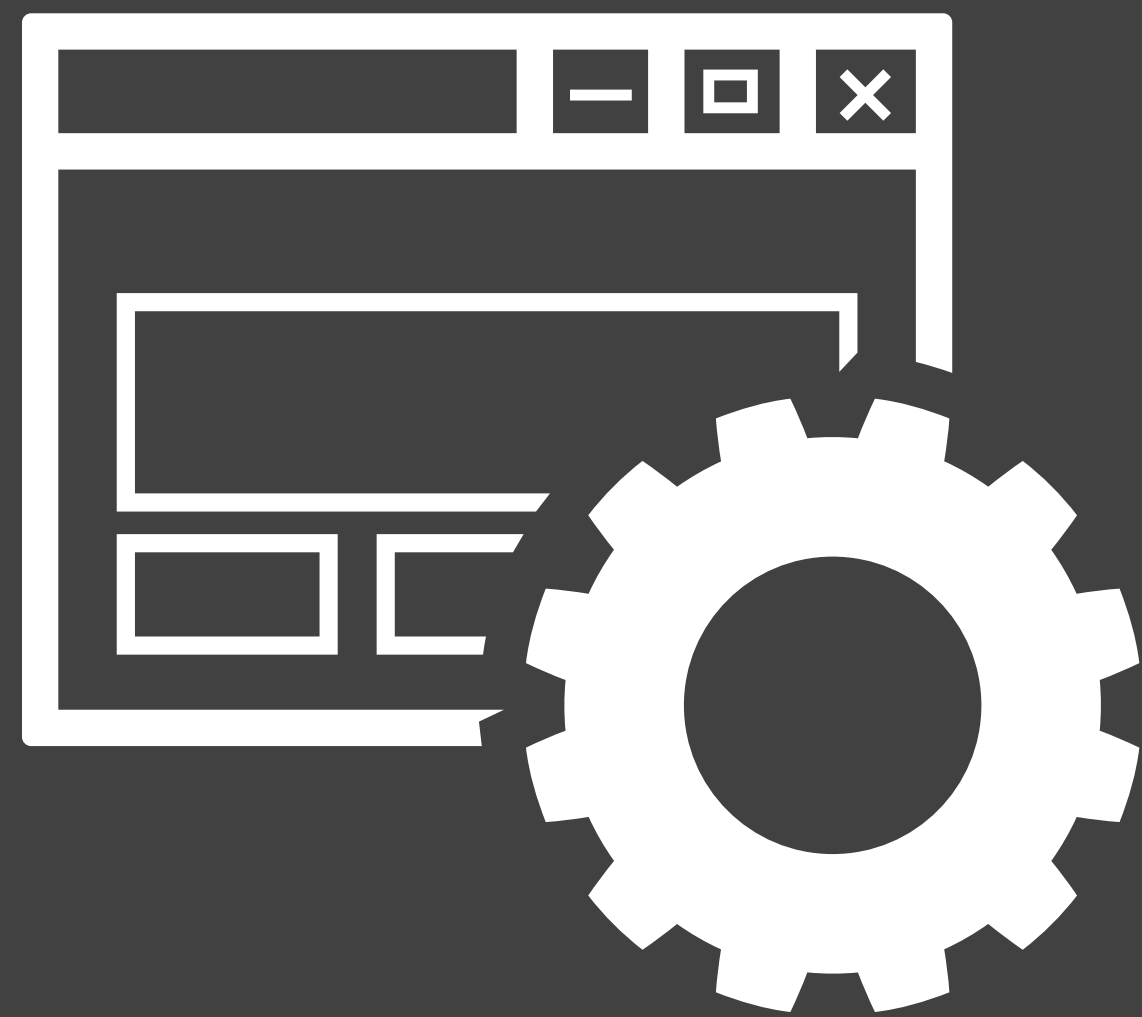
ANTI-PATTERNS



The last 10% trap



ANTI-PATTERN



Coding via Configuration



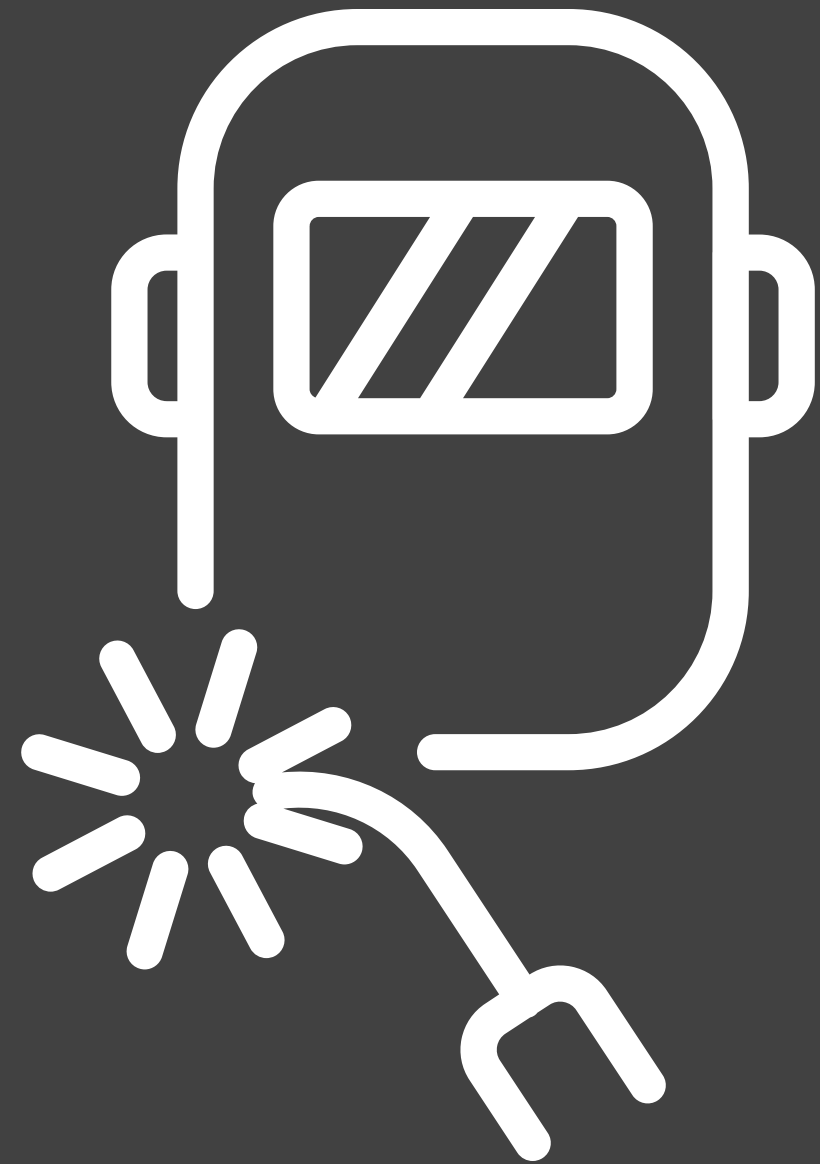
ANTI-PATTERN



Product Customisation



ANTI-PATTERN



Exuberant Coupling



ANTI-PATTERN

CHOOSING STYLES



Build

Buy

$\bar{N}26$



Build

Buy

Build and/or Buy

N 26

Custom
code

Libraries

Frameworks

COTS or
Software Products



Functionality

Build and/or Buy

N 26

Functionality

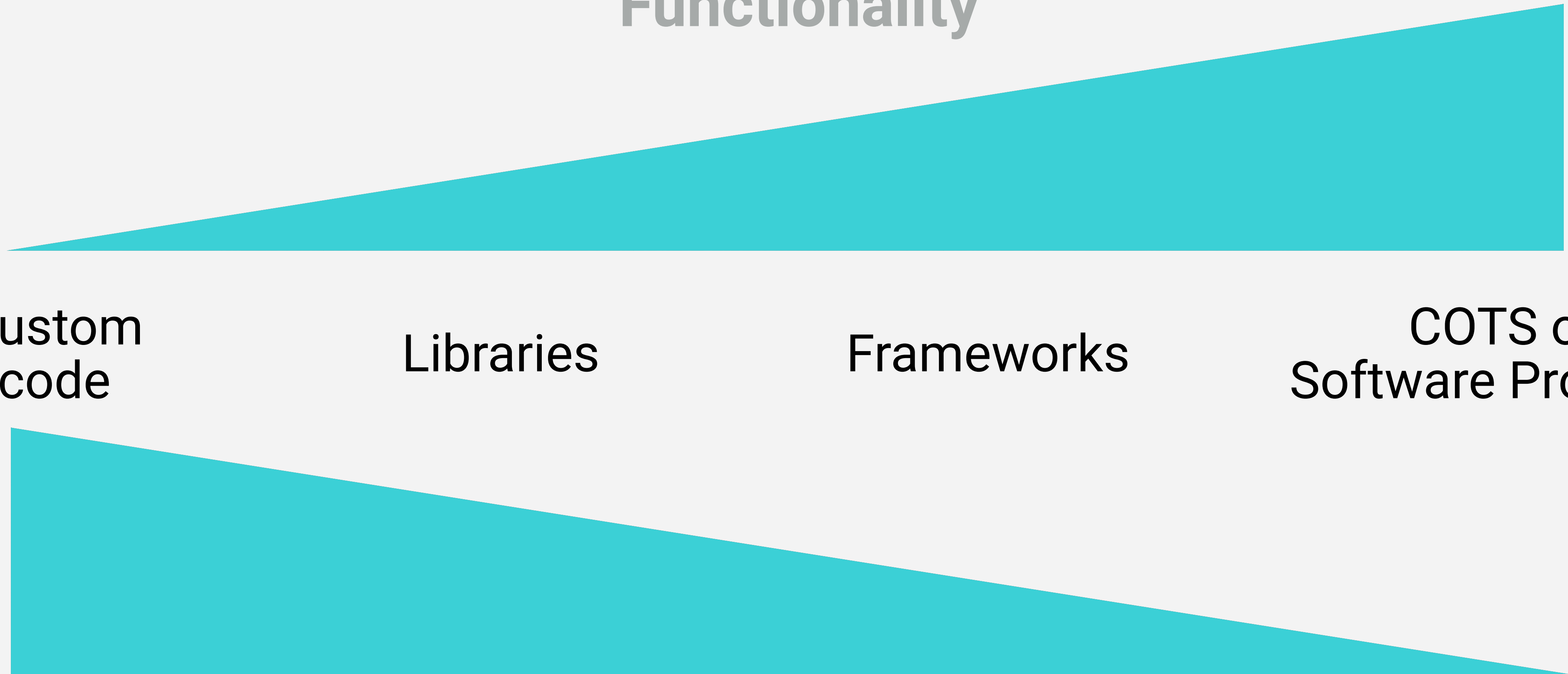
Custom
code

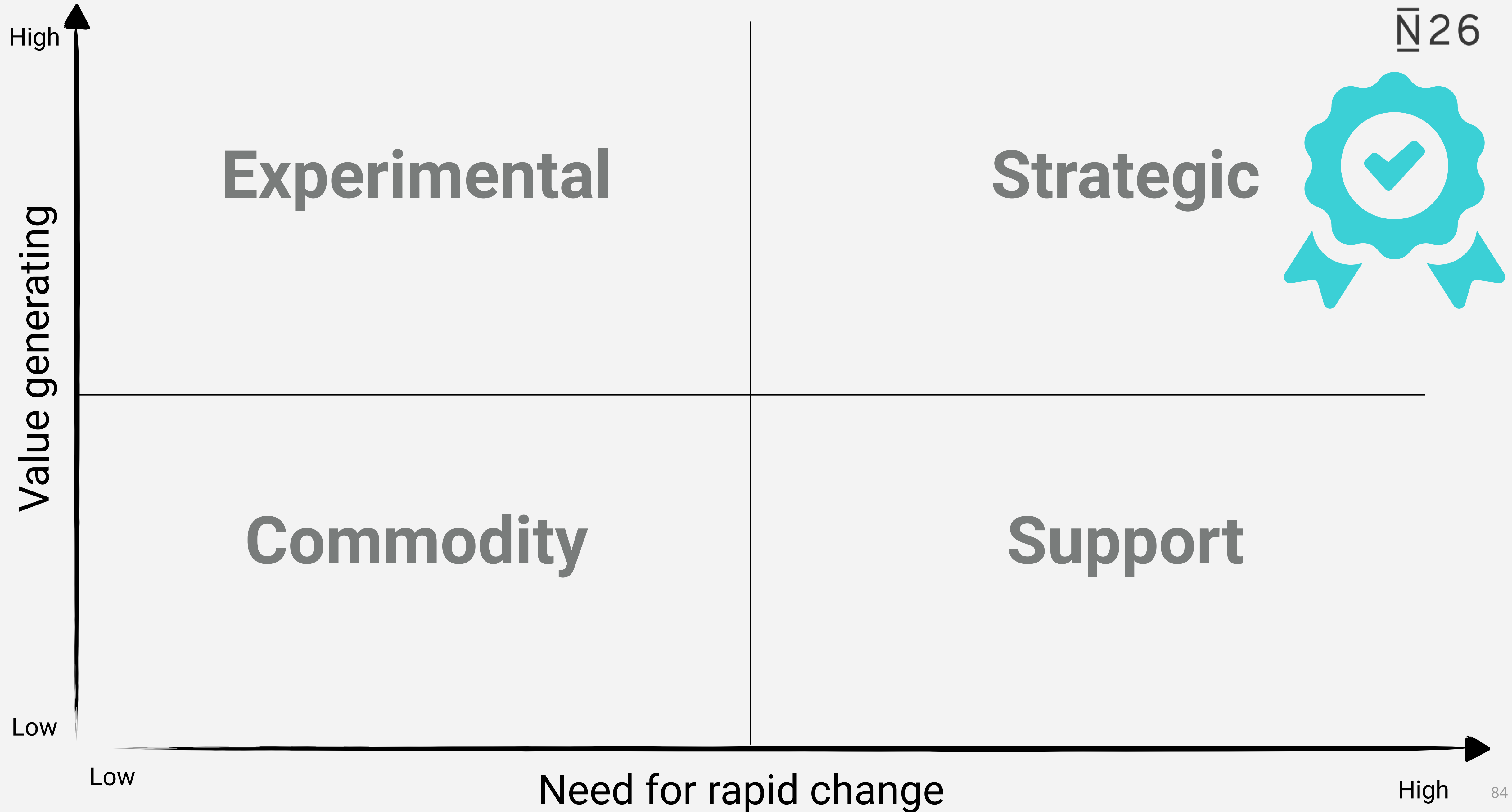
Libraries

Frameworks

COTS or
Software Products

Ability to Change





CONCLUSION

DEFINITION

An evolutionary architecture supports **incremental, guided** change as a first principle along **multiple dimensions**



TO CONSIDER

Architectural choices

Decision making process + thinking

Organisational and Cultural aspects



Thank you

Psst. We're hiring in Berlin
<https://n26.com/jobs>

PATRICK KUA, CTO
@n26, @patkua

