Building Evolutionary Architectures

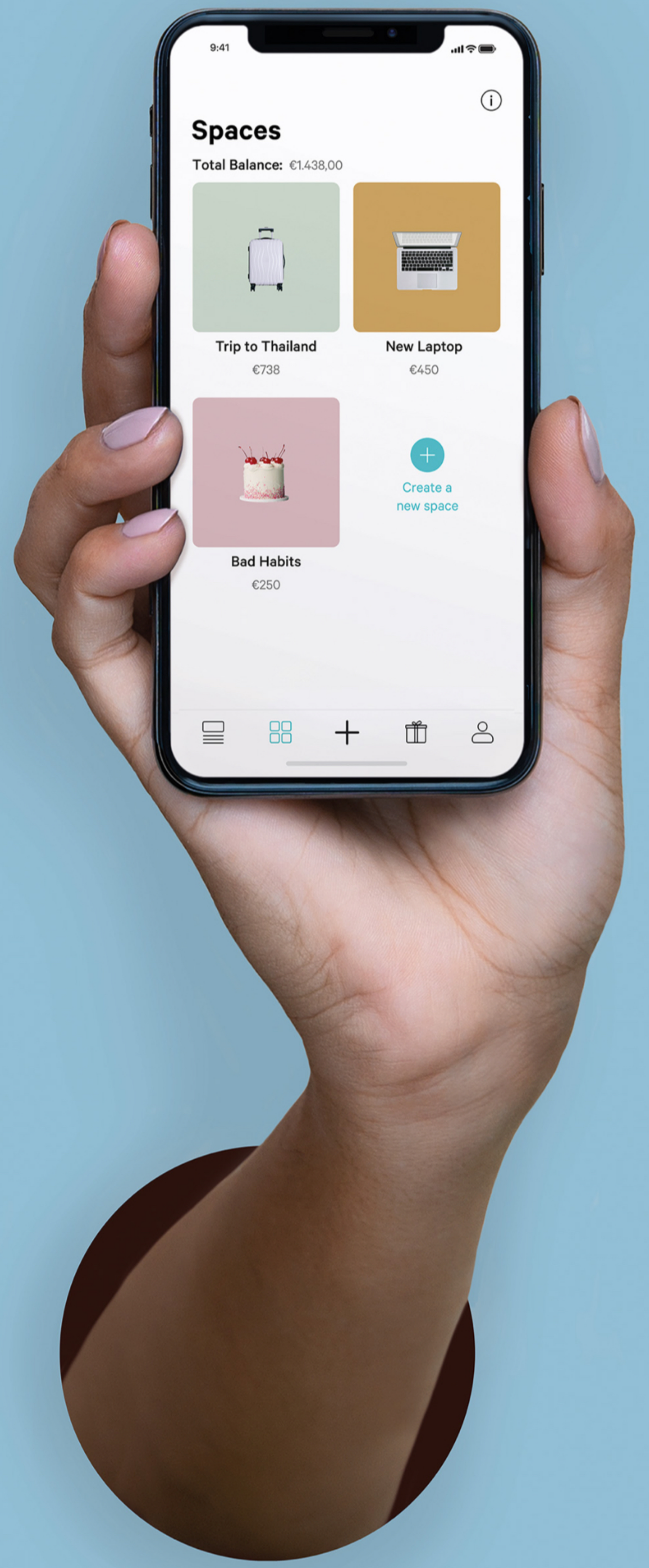
@patkua

TU Berlin - Jan 2019

The first bank you’ll love

#n26reasons

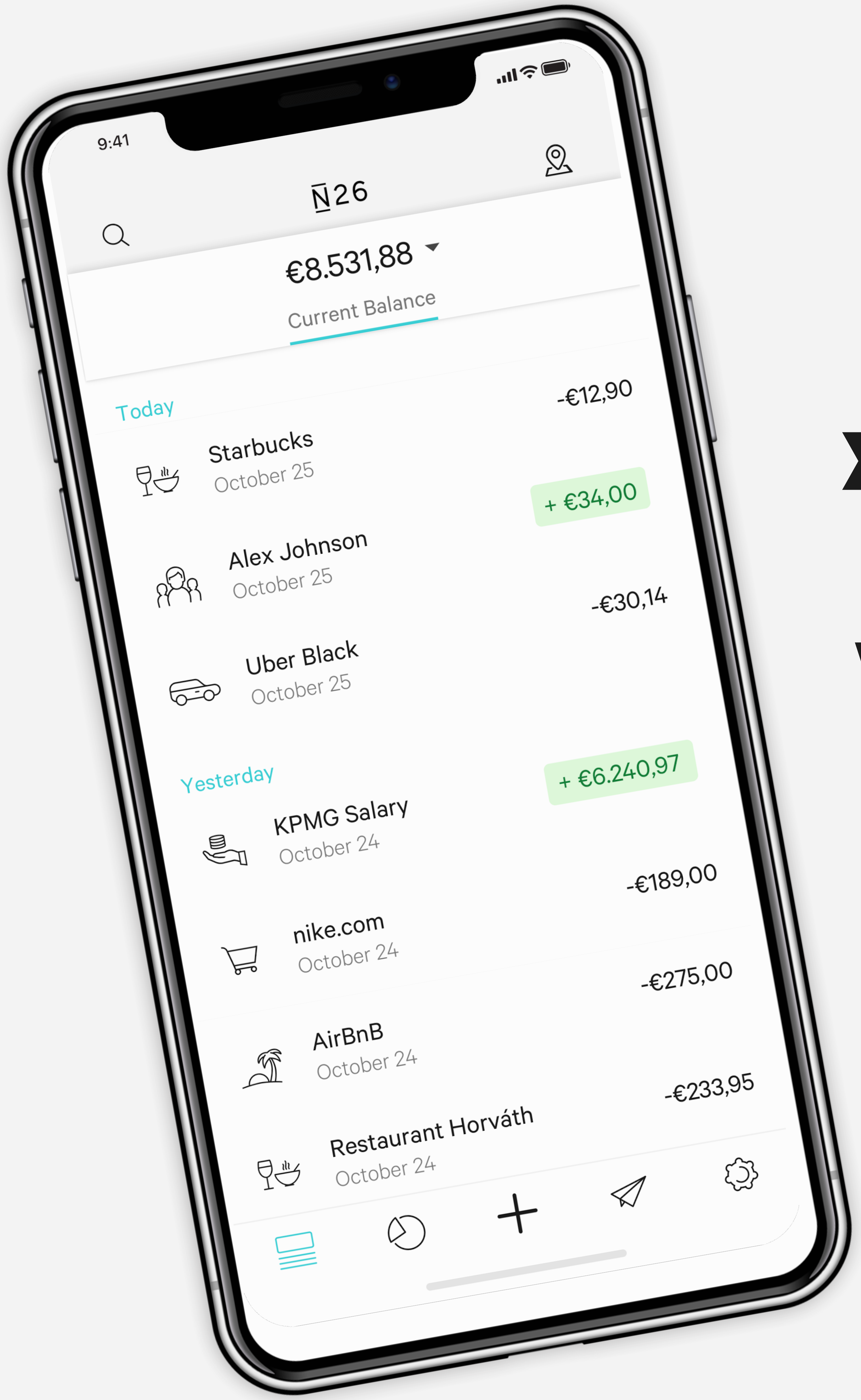
Banking,
this beautiful



#26reasons

N26 Spaces

Make your goals a reality
with Spaces.



9:41

N26

€8.531,88

Current Balance

Today

Starbucks
October 25
-€12,90

Alex Johnson
October 25
+ €34,00

Uber Black
October 25
-€30,14

Yesterday

KPMG Salary
October 24
+ €6.240,97

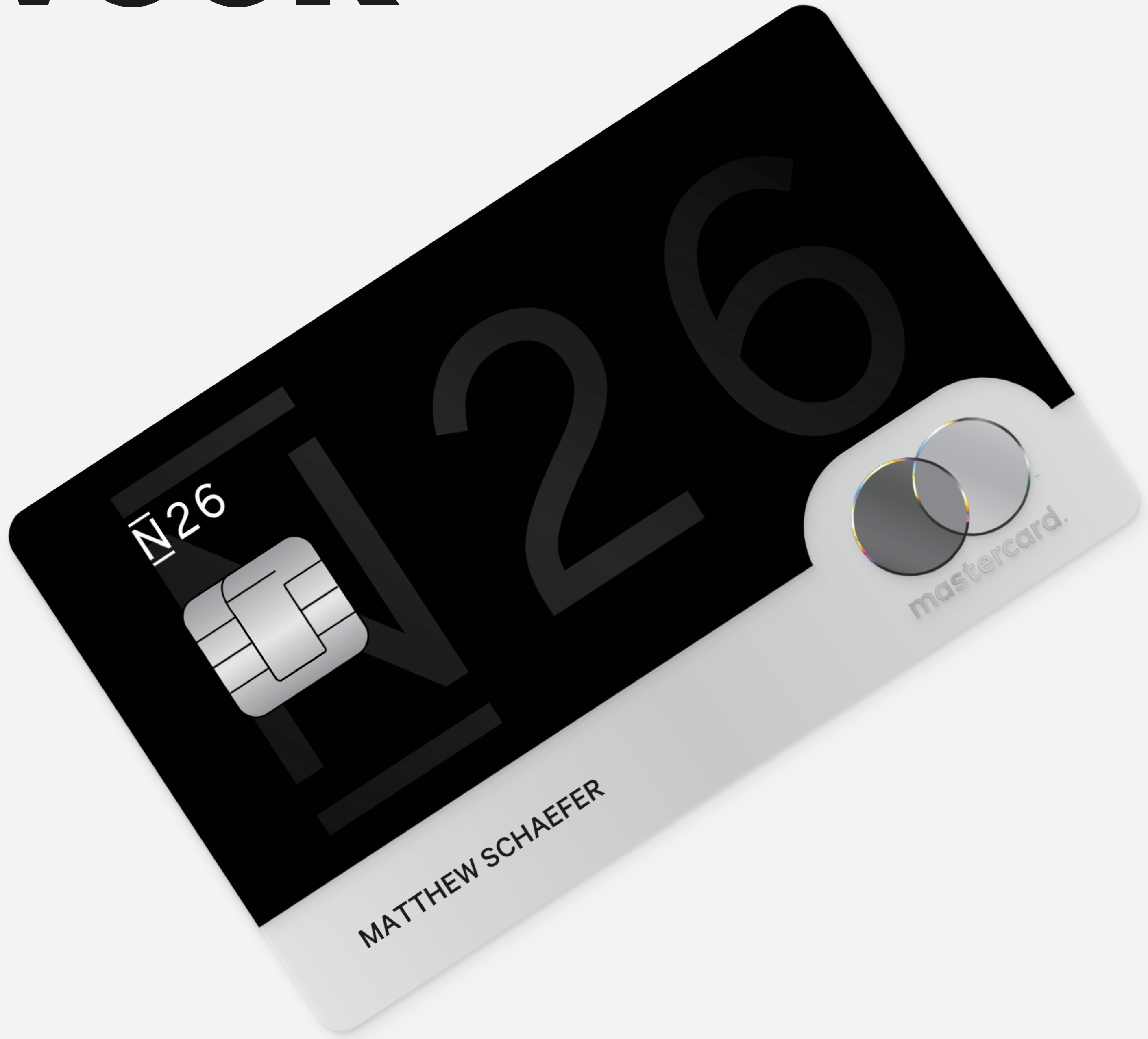
nike.com
October 24
-€189,00

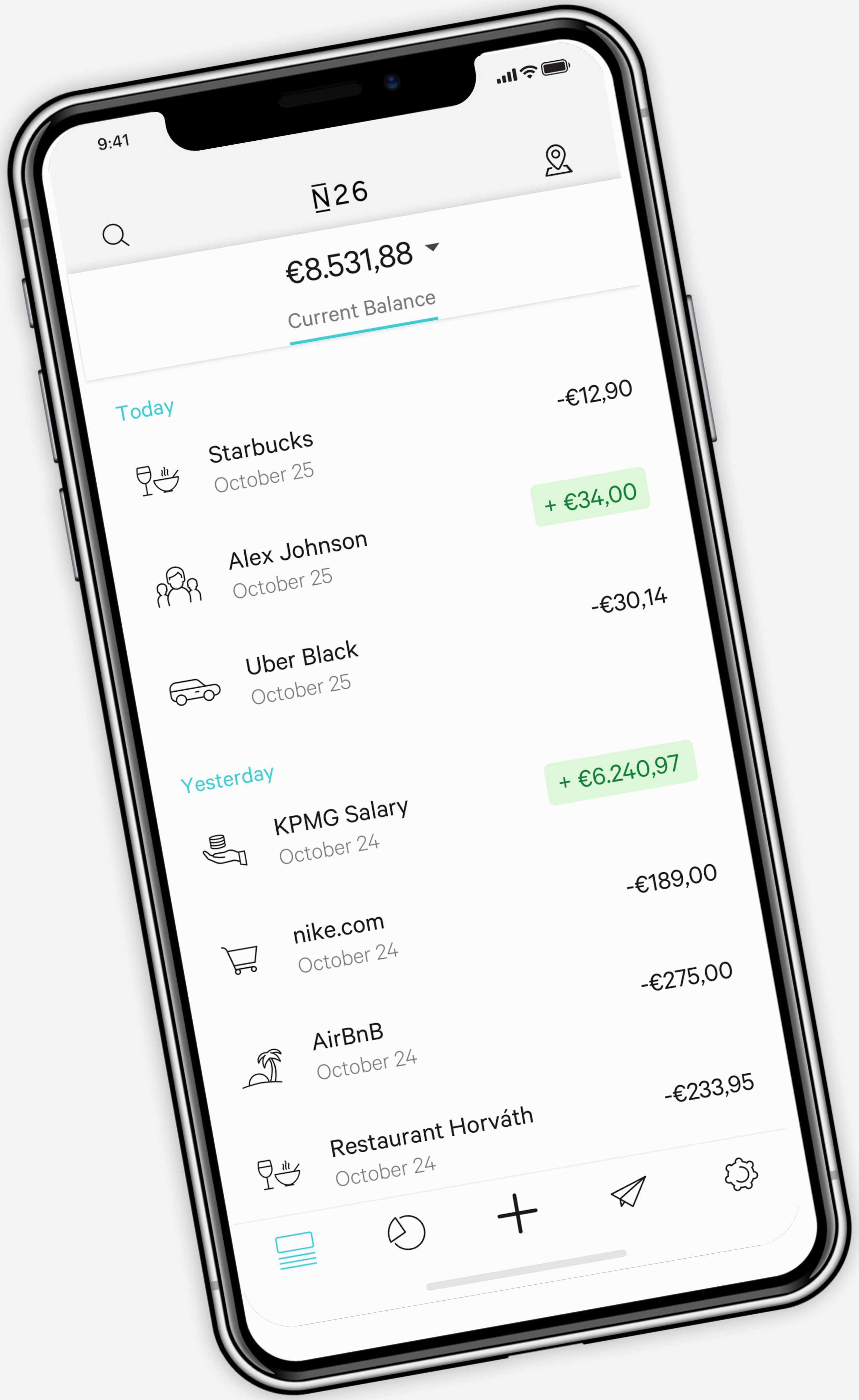
AirBnB
October 24
-€275,00

Restaurant Horváth
October 24
-€233,95

» N26 is the modern bank with 120+ deploys/week «

enabled by a platform built
with CD and Infra-as-Code





9:41

N26

€8.531,88

Current Balance

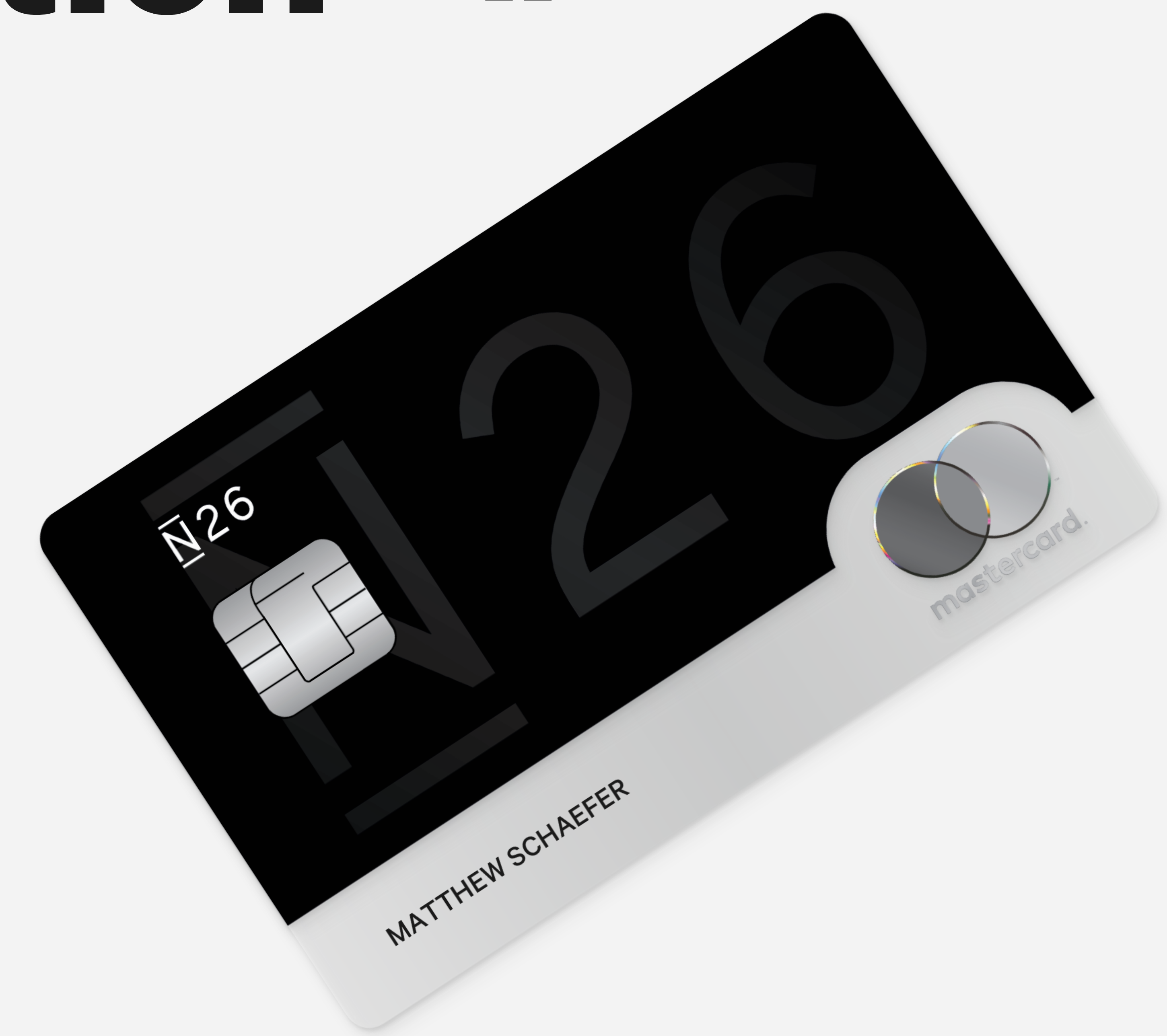
Today

Starbucks	October 25	-€12,90
Alex Johnson	October 25	+ €34,00
Uber Black	October 25	-€30,14

Yesterday

KPMG Salary	October 24	+ €6.240,97
nike.com	October 24	-€189,00
AirBnB	October 24	-€275,00
Restaurant Horváth	October 24	-€233,95

» $300M D-round with $2.7B valuation «



CTO of N26

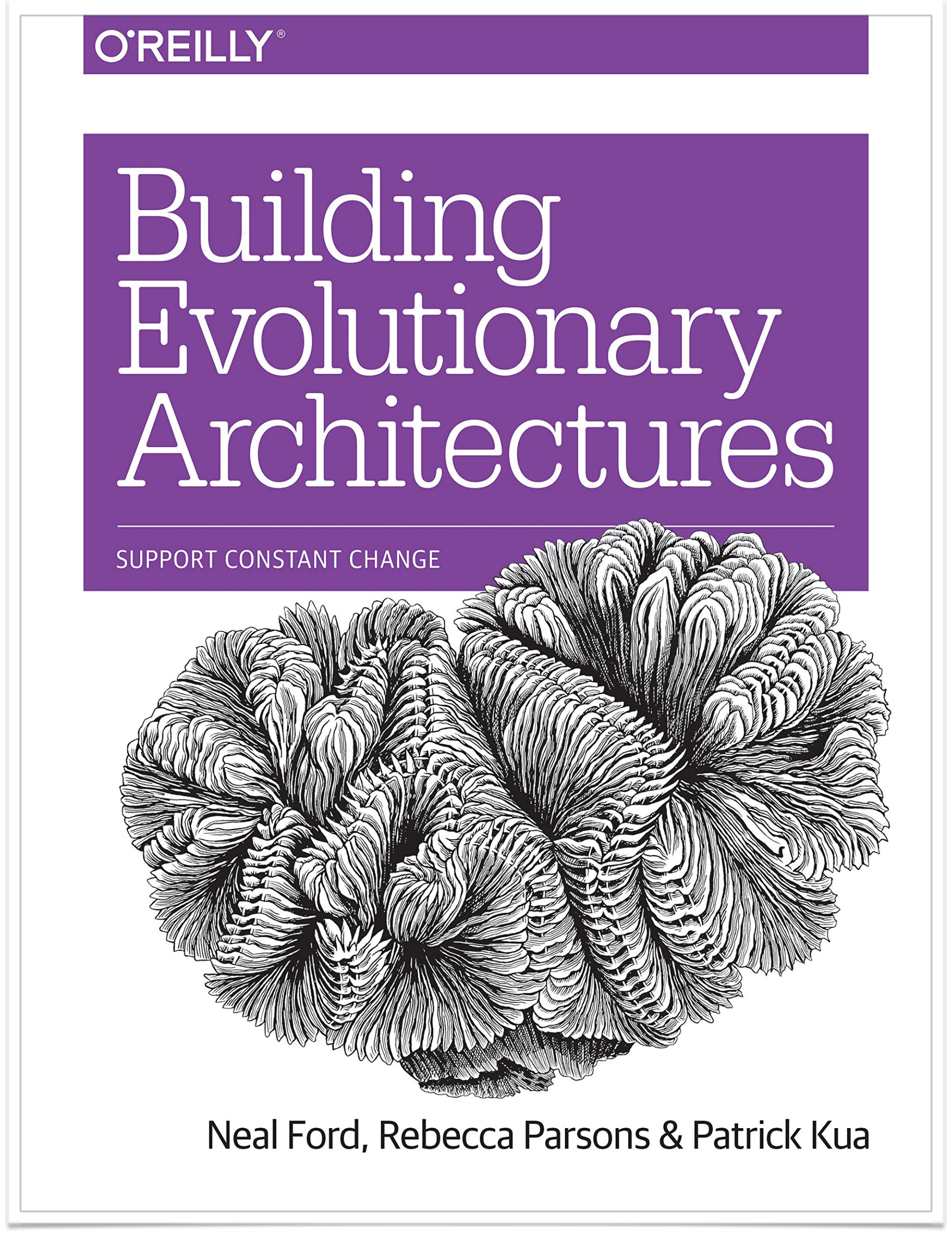
I am building the technology group behind the modern bank designed for the digital age.

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

N26

CTO of N26

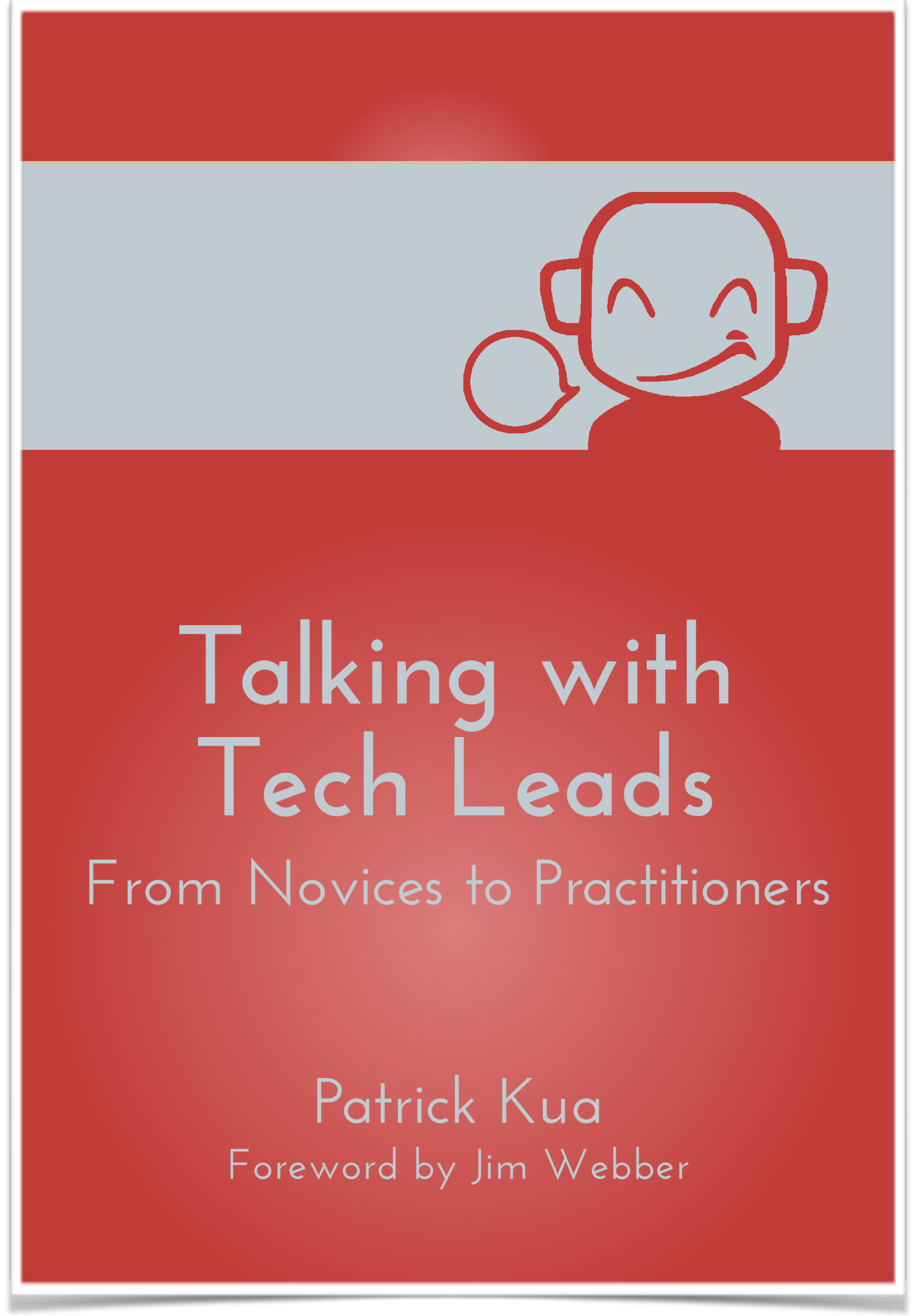

O'REILLY

Building Evolutionary Architectures

SUPPORT CONSTANT CHANGE

Neal Ford, Rebecca Parsons & Patrick Kua

thekua.io/evolarch

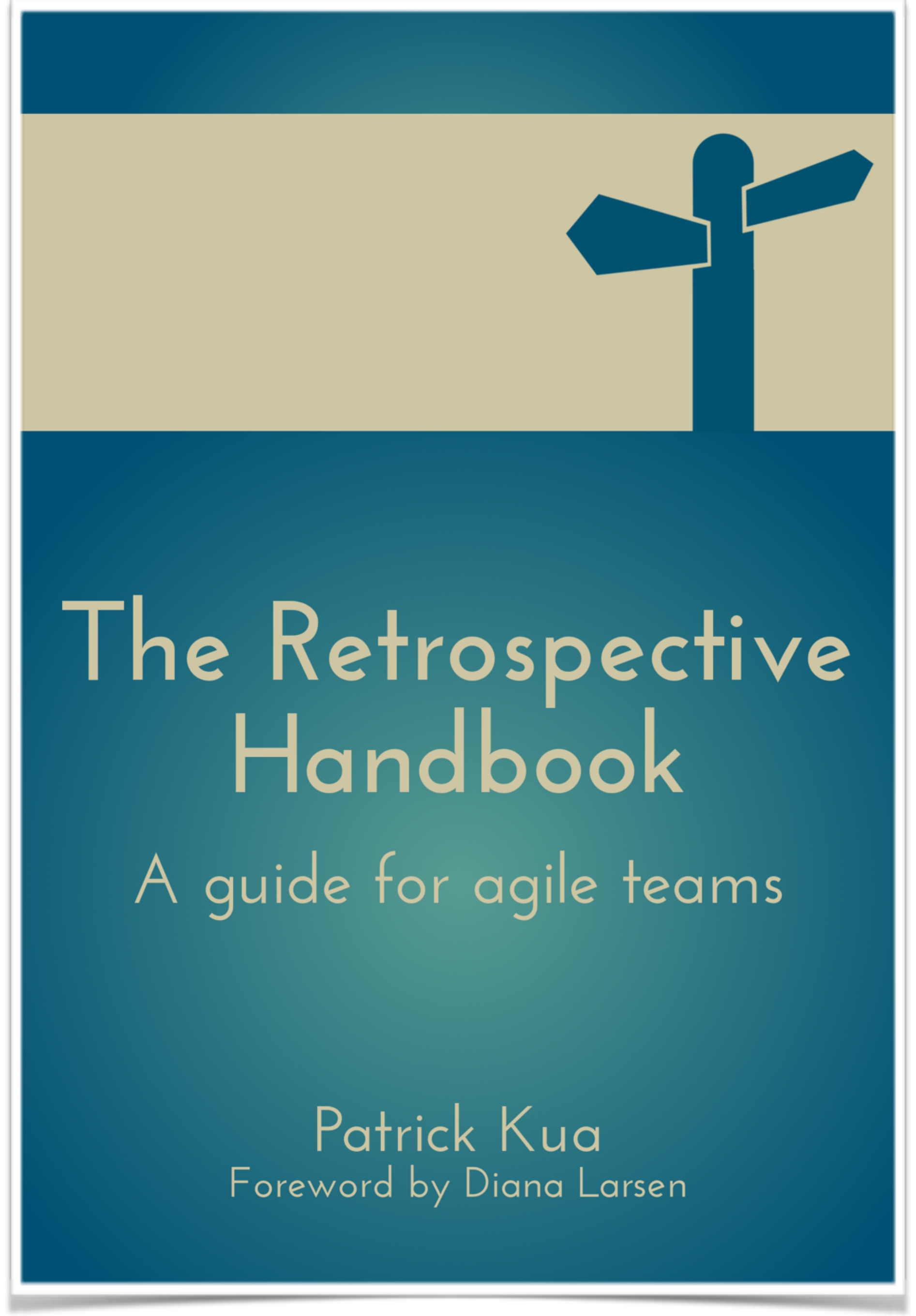

Talking with Tech Leads

From Novices to Practitioners

Patrick Kua

Foreword by Jim Webber

thekua.io/twtl


The Retrospective Handbook

A guide for agile teams

Patrick Kua

Foreword by Diana Larsen

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

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

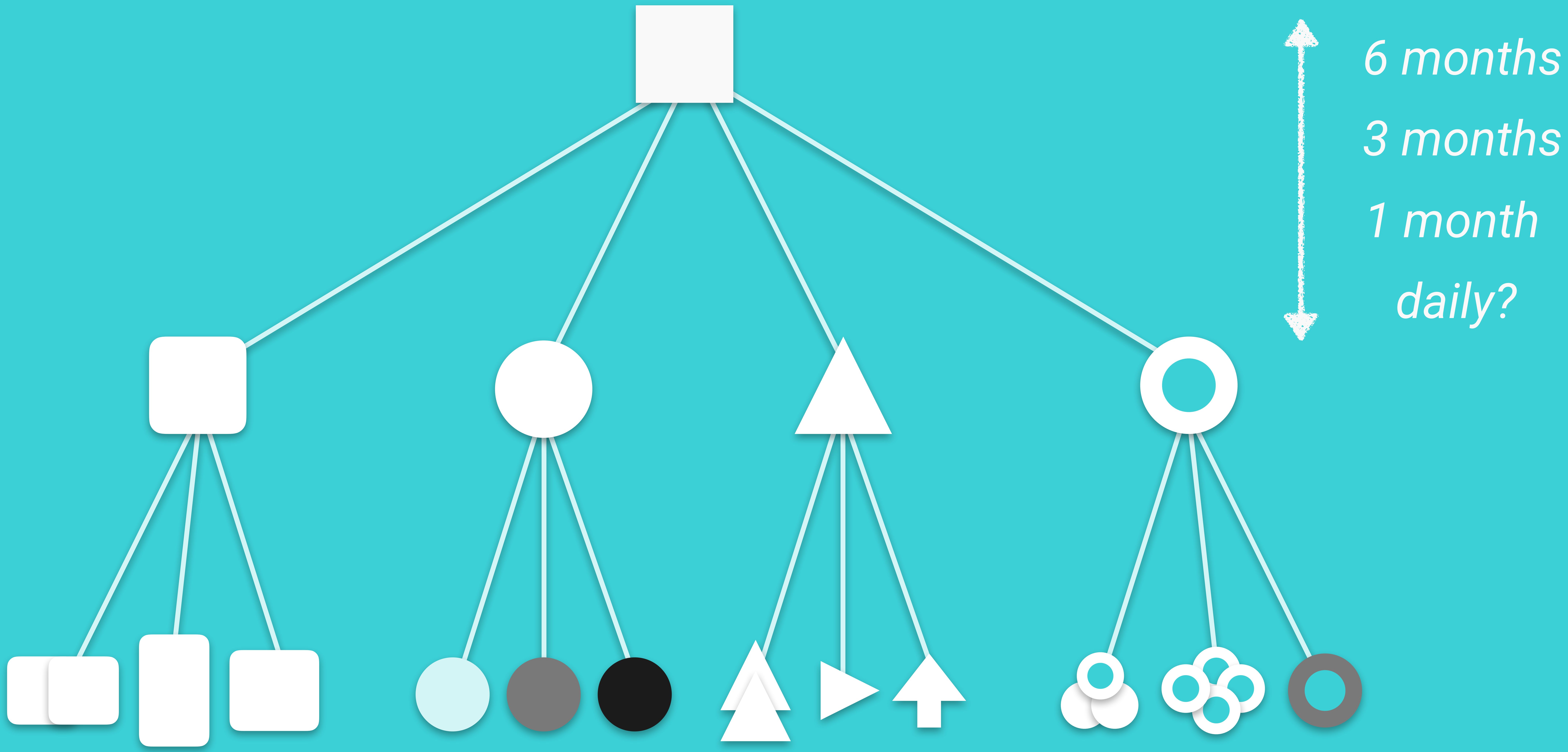
GENERATIONS

N26

Incremental releases represent

GENERATIONS

GENERATIONS



6 months

3 months

1 month

daily?

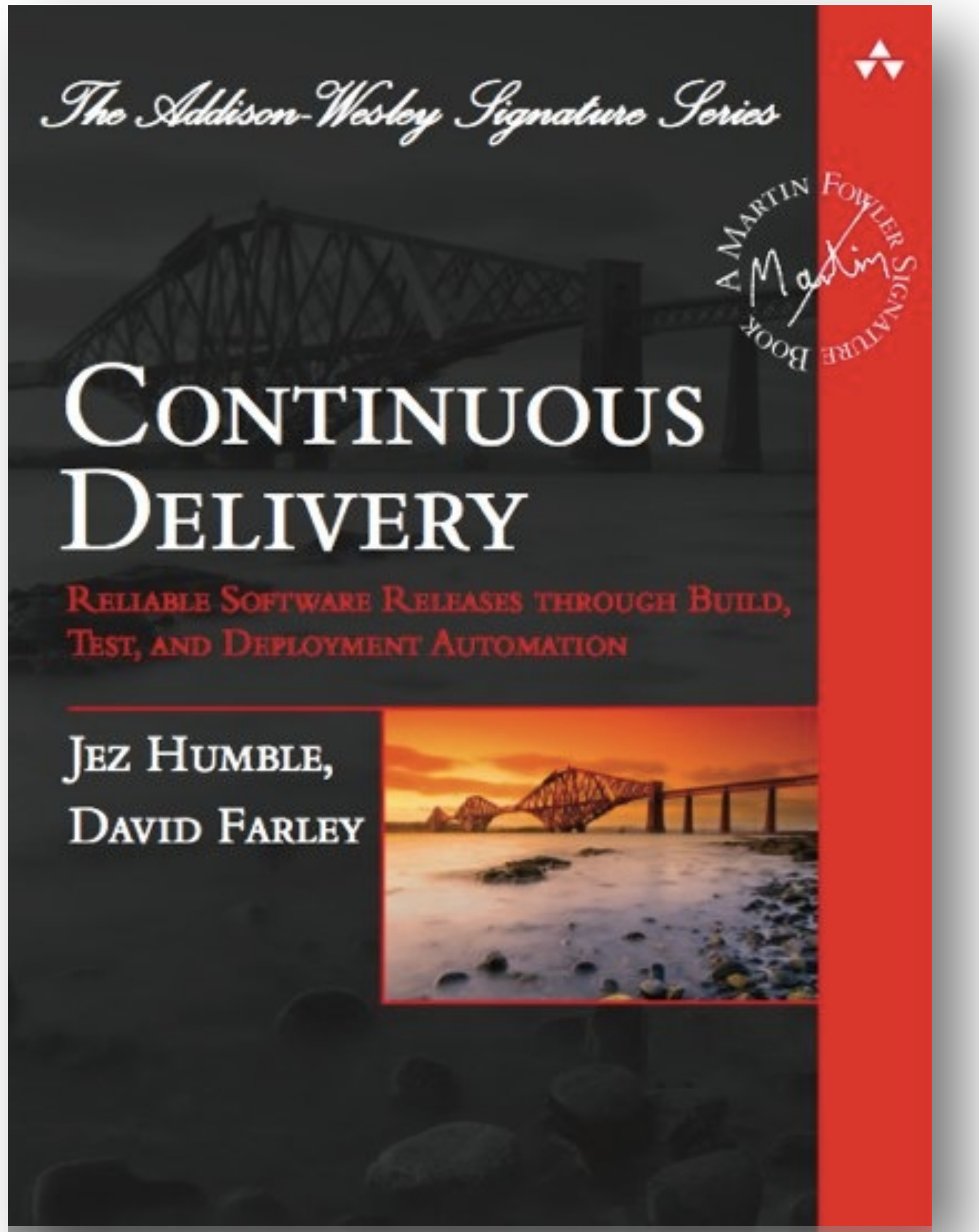
GENERATIONS

GENERATIONS = CYCLE TIME

Time taken to get a simple change into production

repeatably
reliably

GENERATIONS = CYCLE TIME



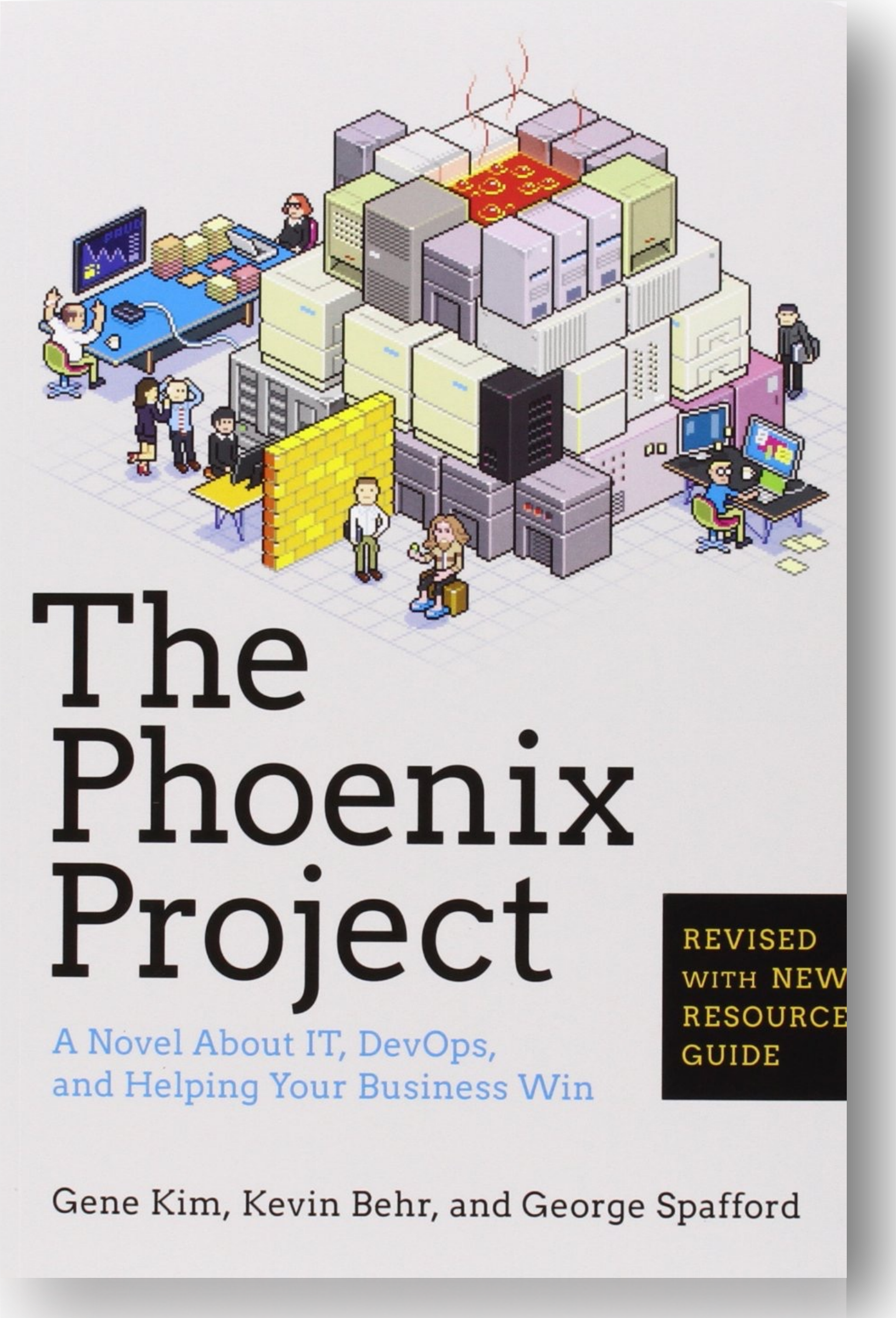
The Addison-Wesley Signature Series

A Martin Fowler Signature Book

Continuous Delivery

Reliable Software Releases through Build, Test, and Deployment Automation

Jez Humble, David Farley

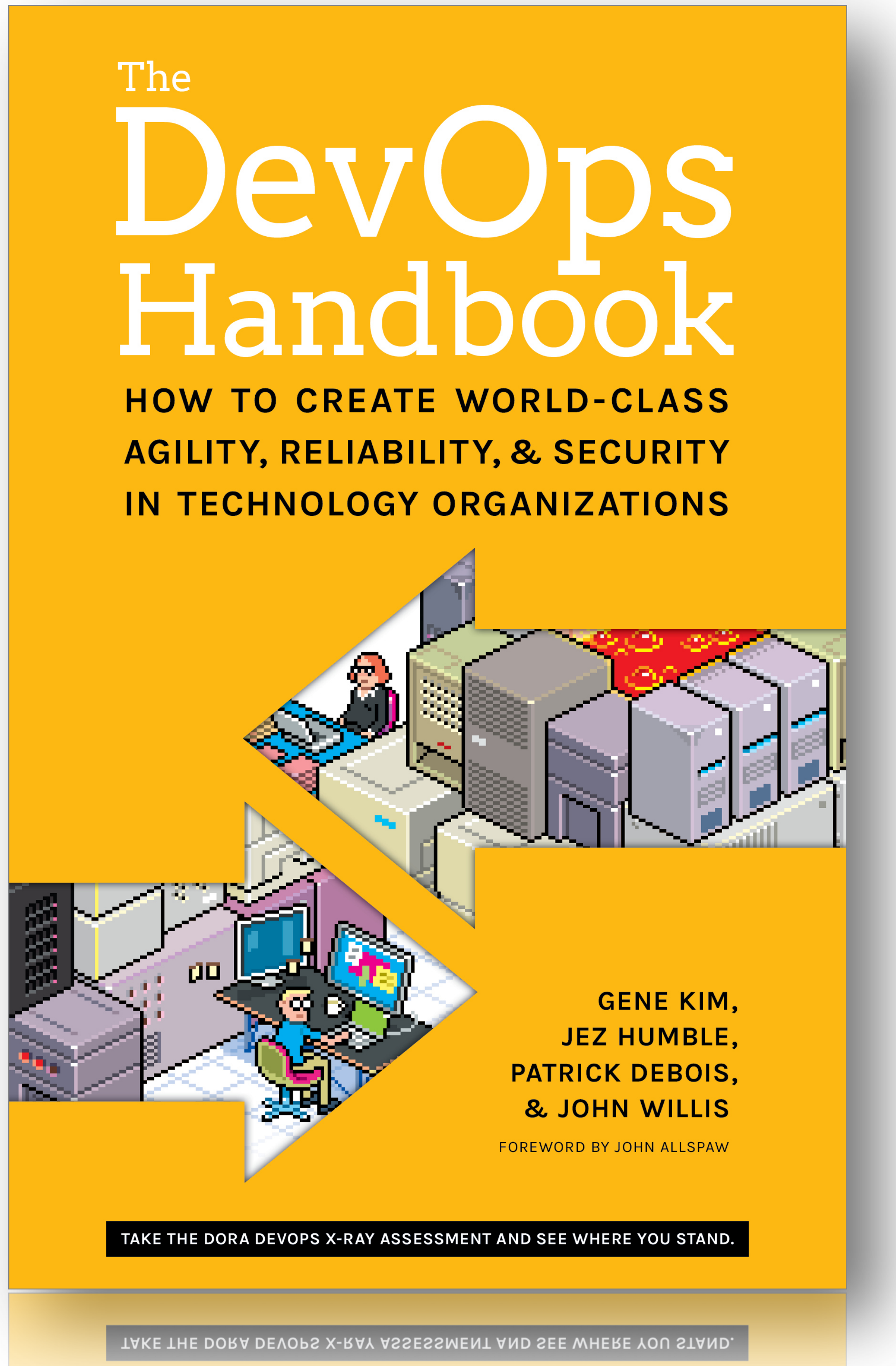


The Phoenix Project

A Novel About IT, DevOps, and Helping Your Business Win

REVISED WITH NEW RESOURCE GUIDE

Gene Kim, Kevin Behr, and George Spafford



The DevOps Handbook

HOW TO CREATE WORLD-CLASS AGILITY, RELIABILITY, & SECURITY IN TECHNOLOGY ORGANIZATIONS

GENE KIM, JEZ HUMBLE, PATRICK DEBOIS, & JOHN WILLIS

FOREWORD BY JOHN ALLSPAW

TAKE THE DORA DEVOPS X-RAY ASSESSMENT AND SEE WHERE YOU STAND.

GENERATIONS = CYCLE TIME

Continuous Integration

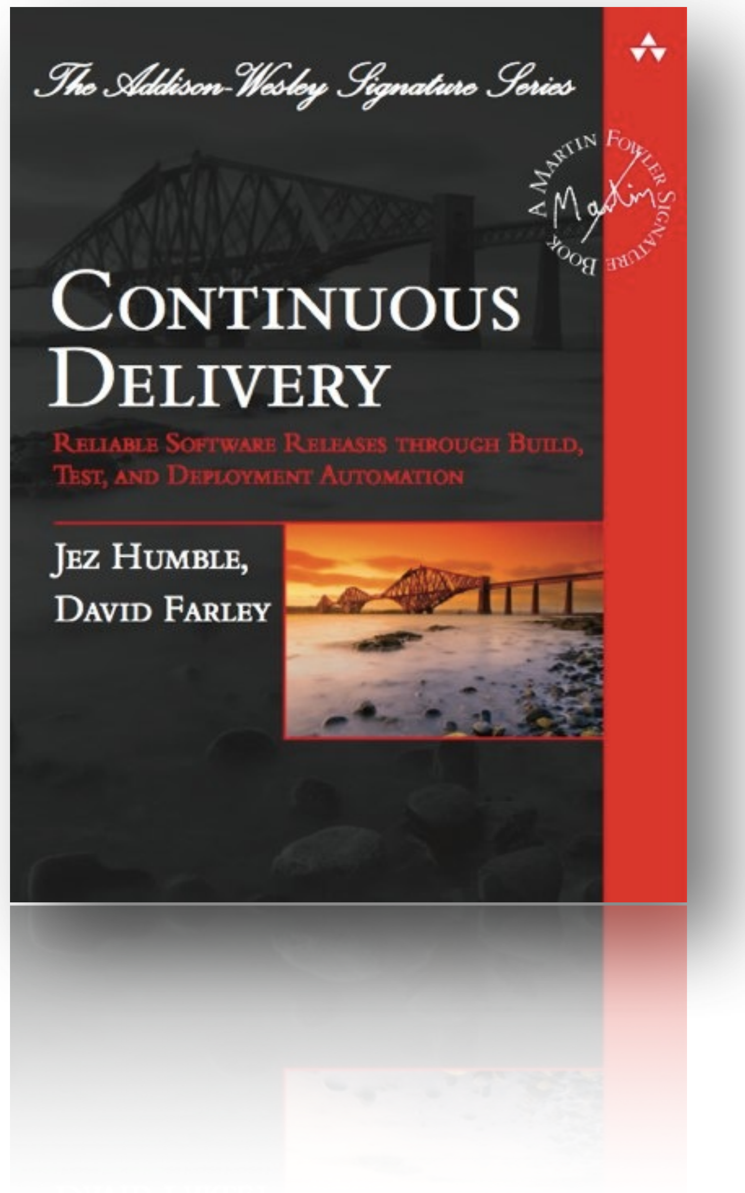
Automate everything

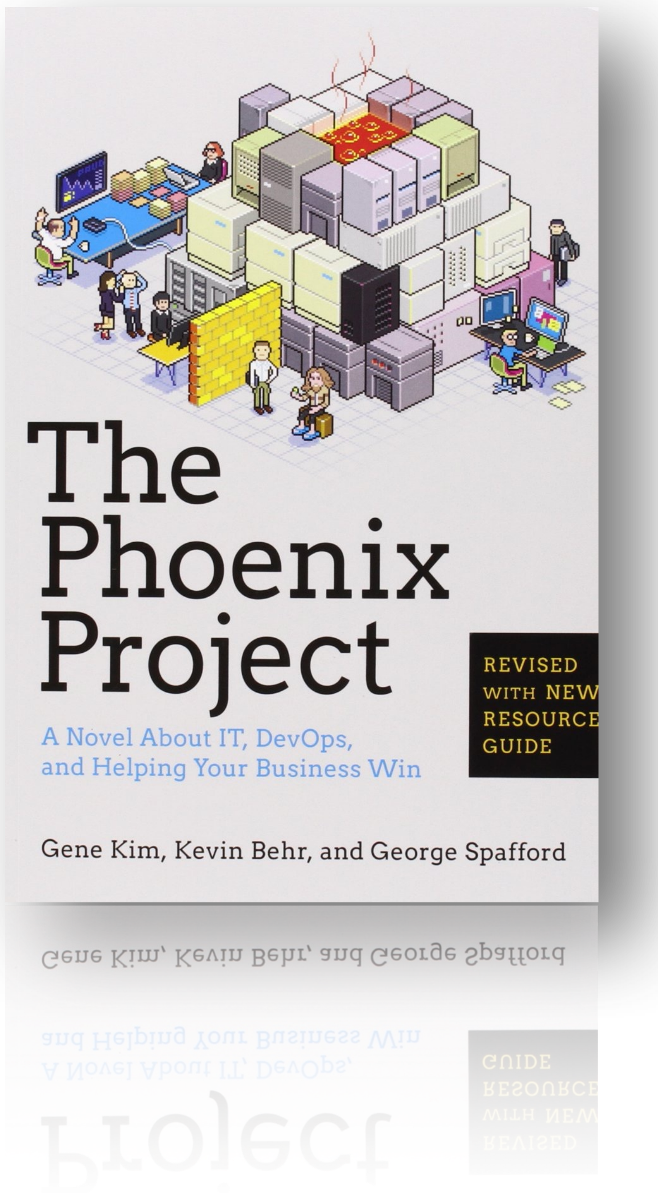
Keep everything in source control

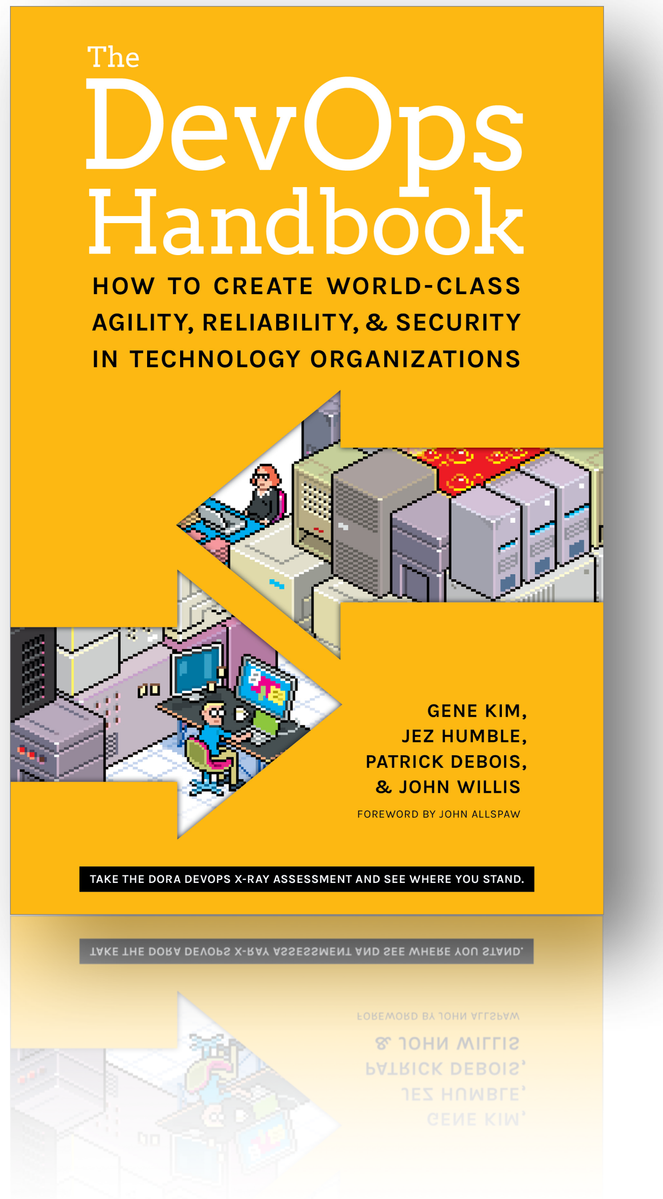
“Done” means released (into production)

Shared release responsibility

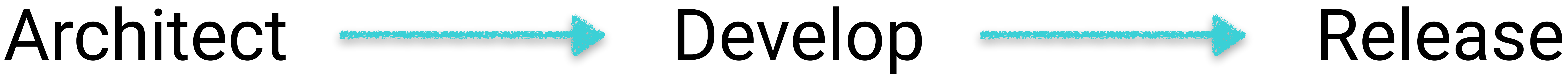
Improve continuously


The Addison-Wesley Signature Series
A Martin Fowler Signature Book
Continuous Delivery
Reliable Software Releases through Build, Test, and Deployment Automation
Jez Humble, David Farley


The Phoenix Project
Revised with New Resource Guide
A Novel About IT, DevOps, and Helping Your Business Win
Gene Kim, Kevin Behr, and George Spafford


The DevOps Handbook
How to Create World-Class Agility, Reliability, & Security in Technology Organizations
Gene Kim, Jez Humble, Patrick Debois, & John Willis
Foreword by John Allspaw
Take the DORA DevOps X-Ray assessment and see where you stand.

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

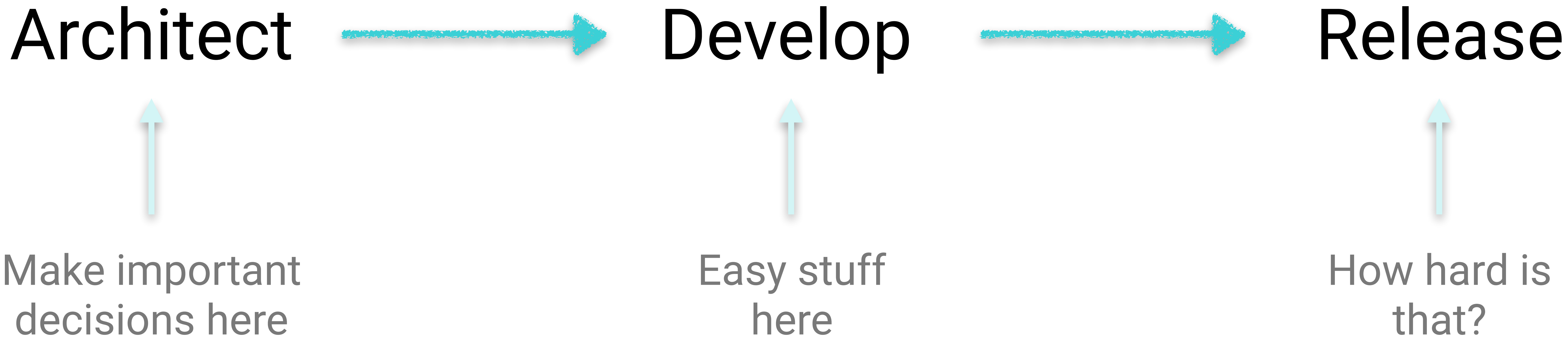
Source: Architecture for Software Developers

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)

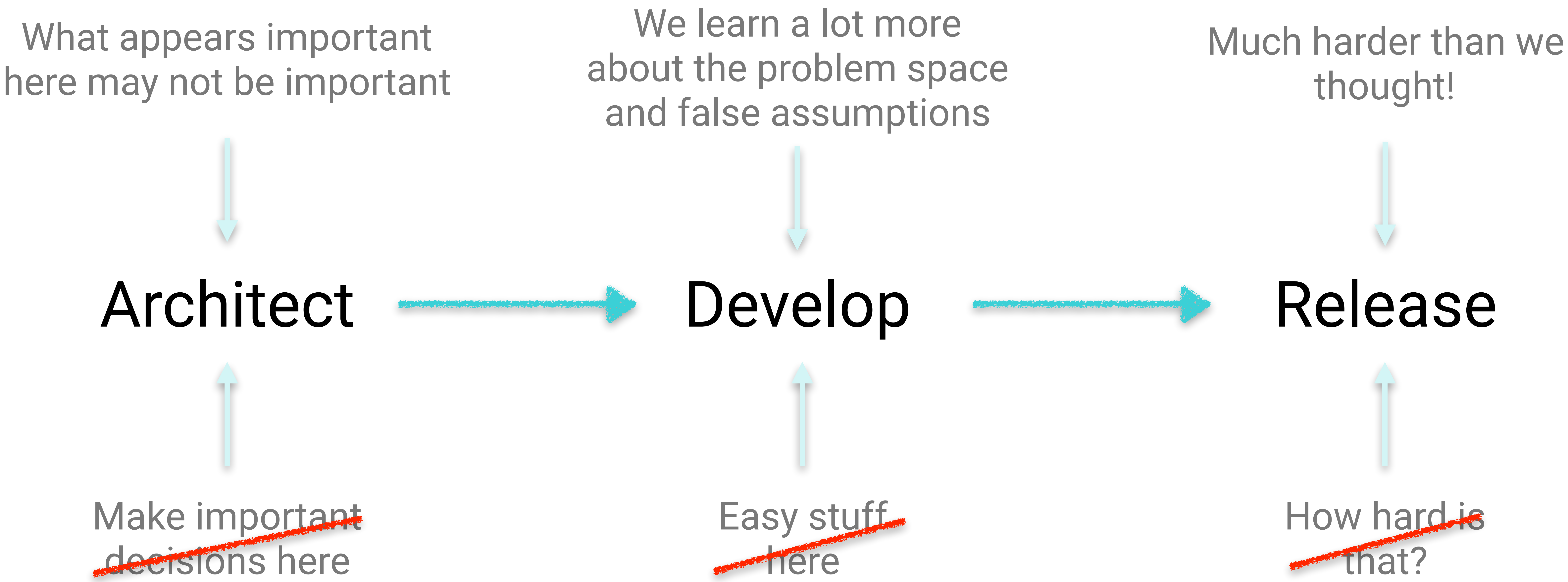


Architect → Develop → Release

Make important decisions here

Easy stuff here

How hard is that?



What appears important here may not be important

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

Much harder than we thought!

Architect

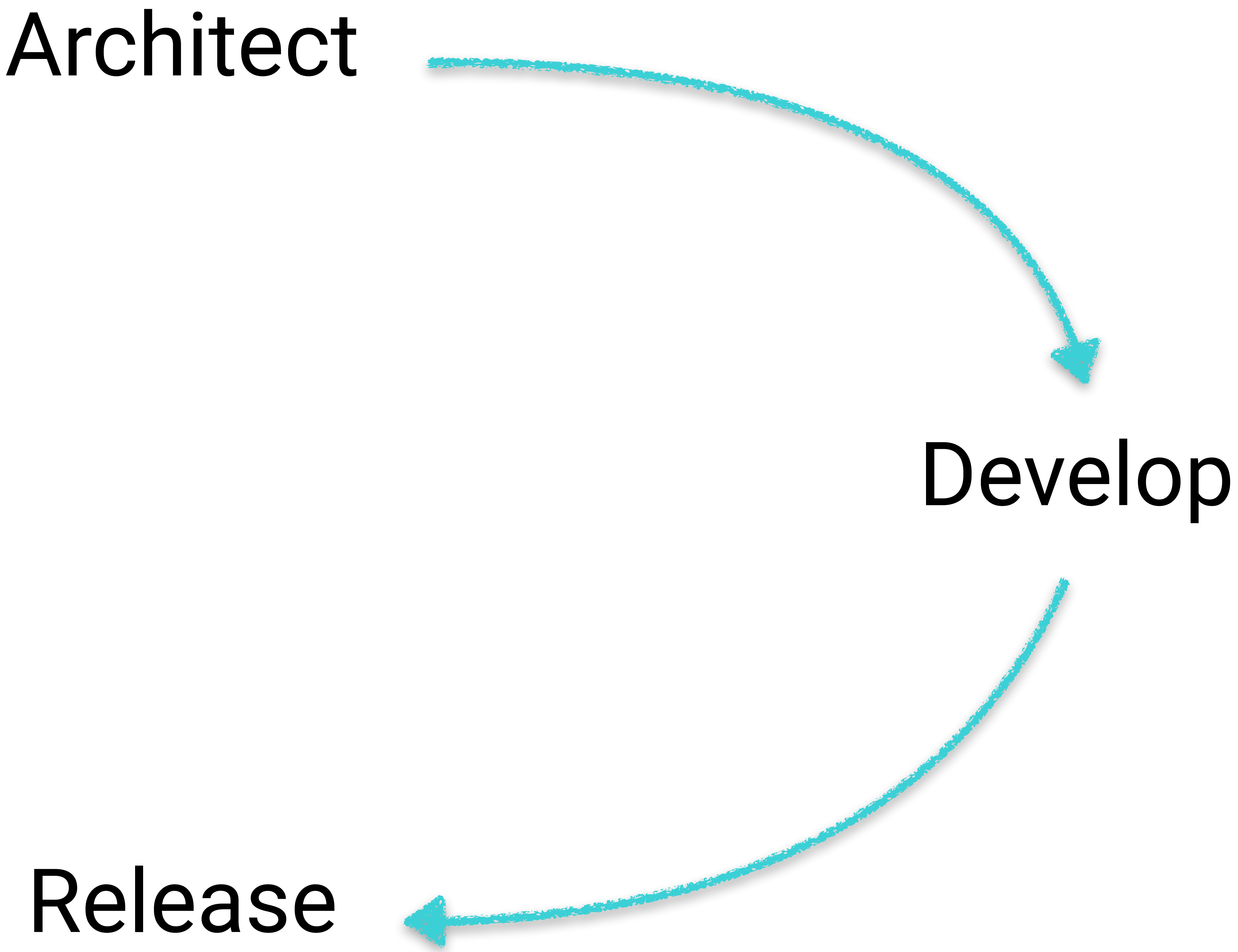
Develop

Release

~~Make important decisions here~~

~~Easy stuff here~~

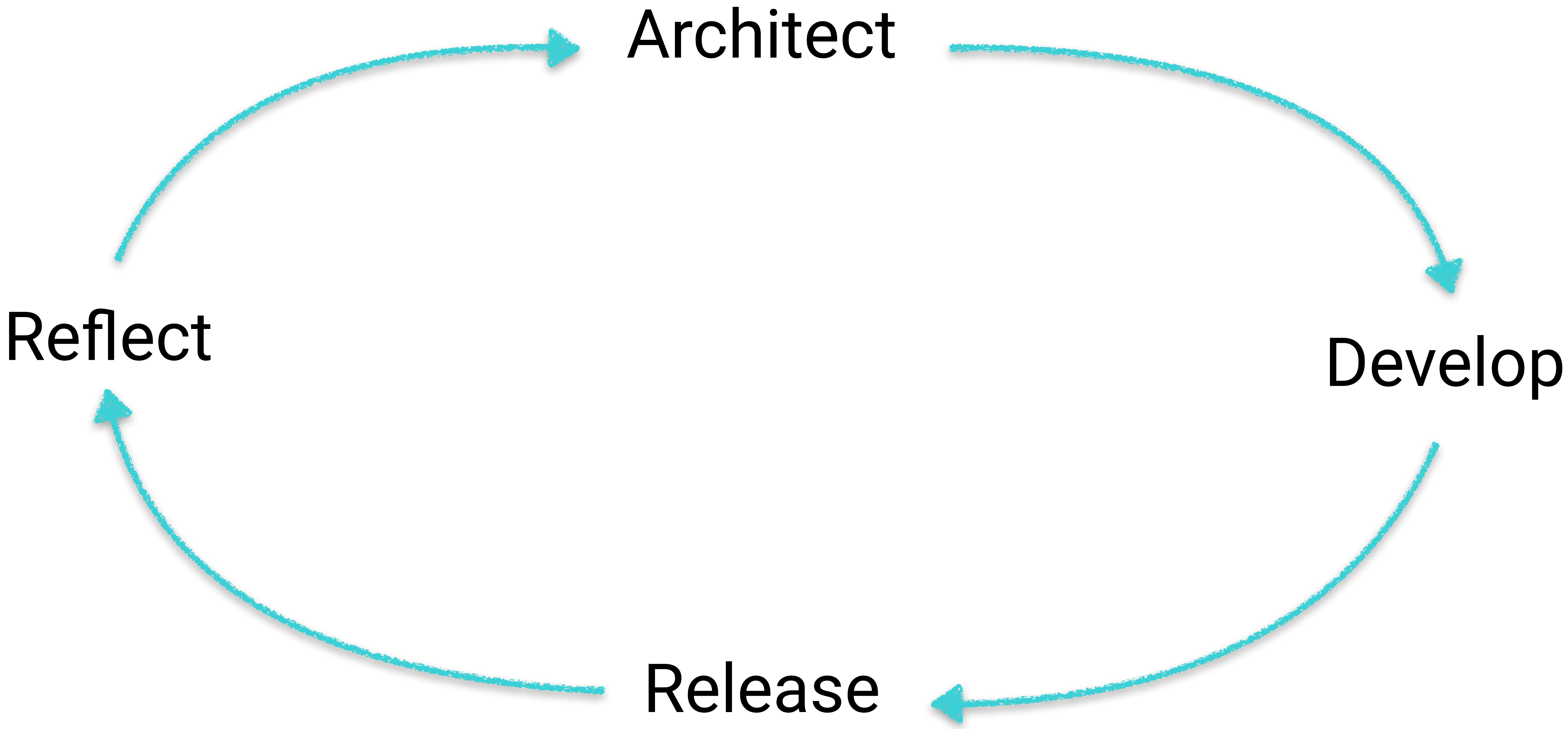
~~How hard is that?~~



Architect

Develop

Release

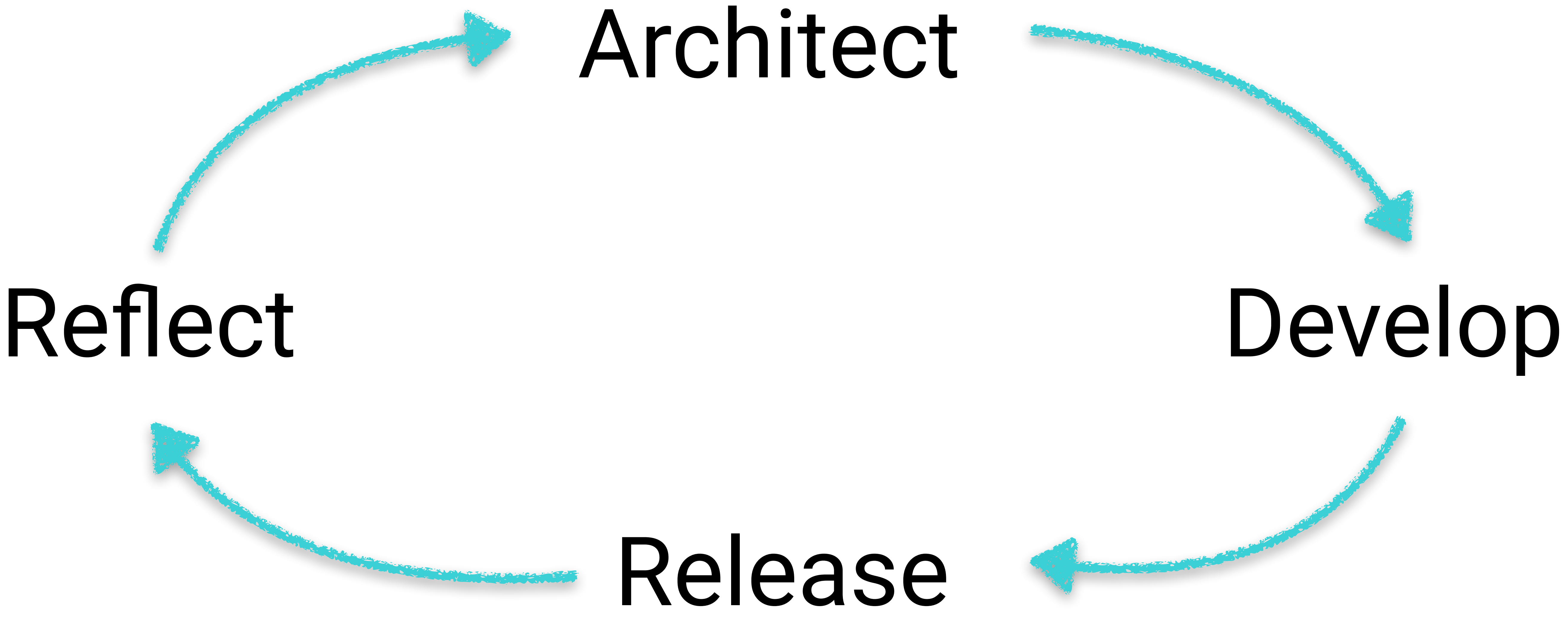


Architect

Develop

Release

Reflect



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	UNIMPORTANT
Strong audit trail	Low response time
Large # of users	Monitoring
Mobile responsive	Internationalisation & Localisation
Availability	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

FITNESS FUNCTIONS & CD

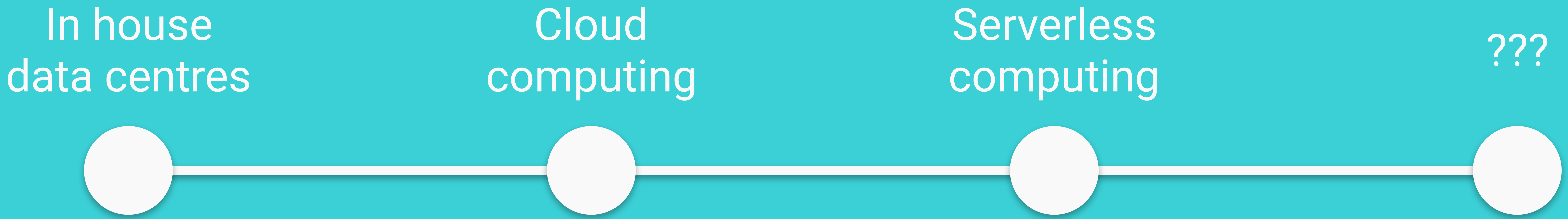
There are known knowns
There are known unknowns

BUT *“If it hurts, do it more o*

There are also unknown Continuous D
unknowns

- Donald Rumsfeld

Metrics, Tests and Process



In house data centres

Cloud computing

Serverless computing

???

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

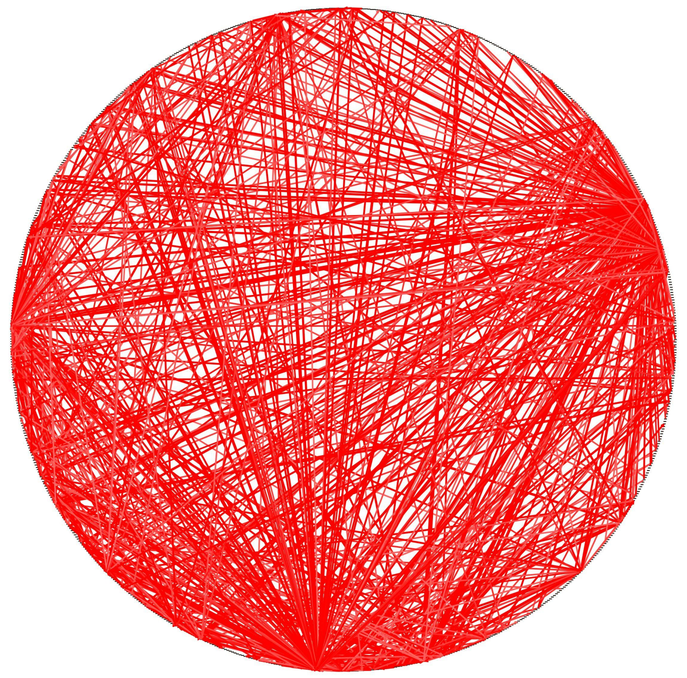
Interface changes

New domain

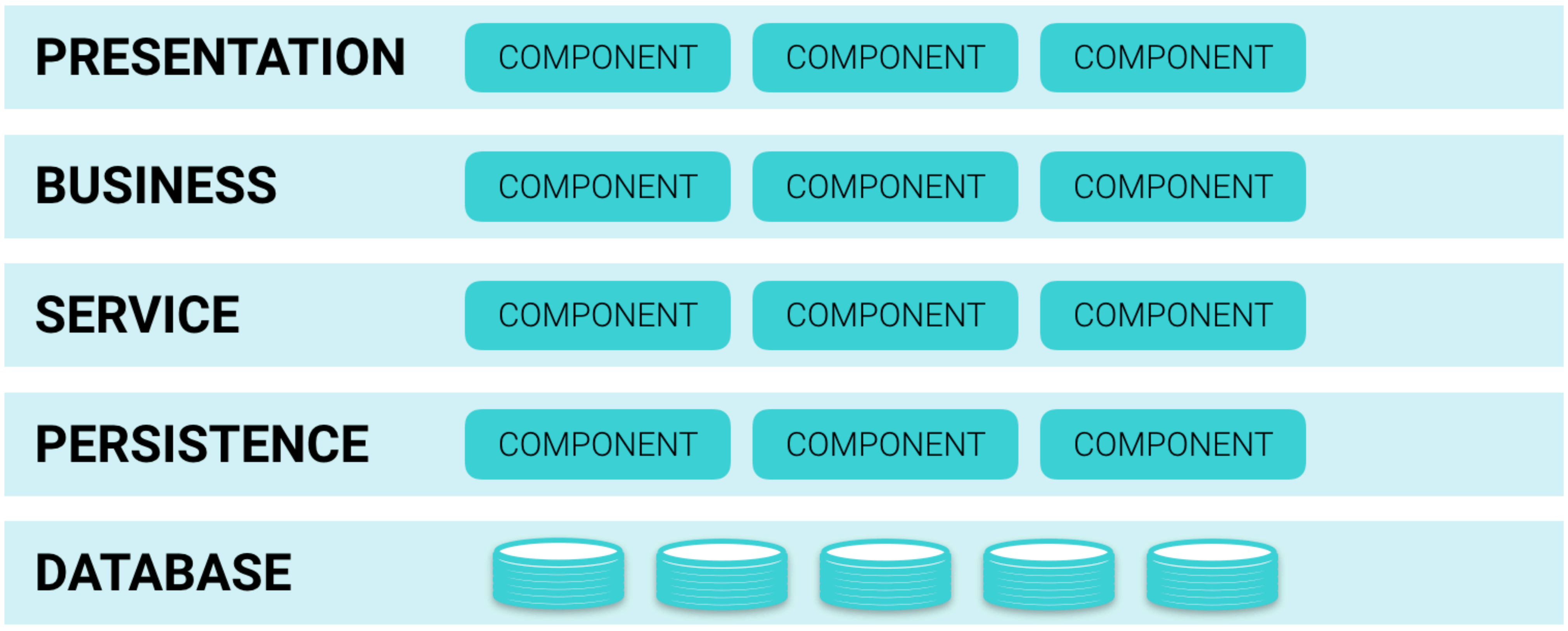
New interfaces

New applications

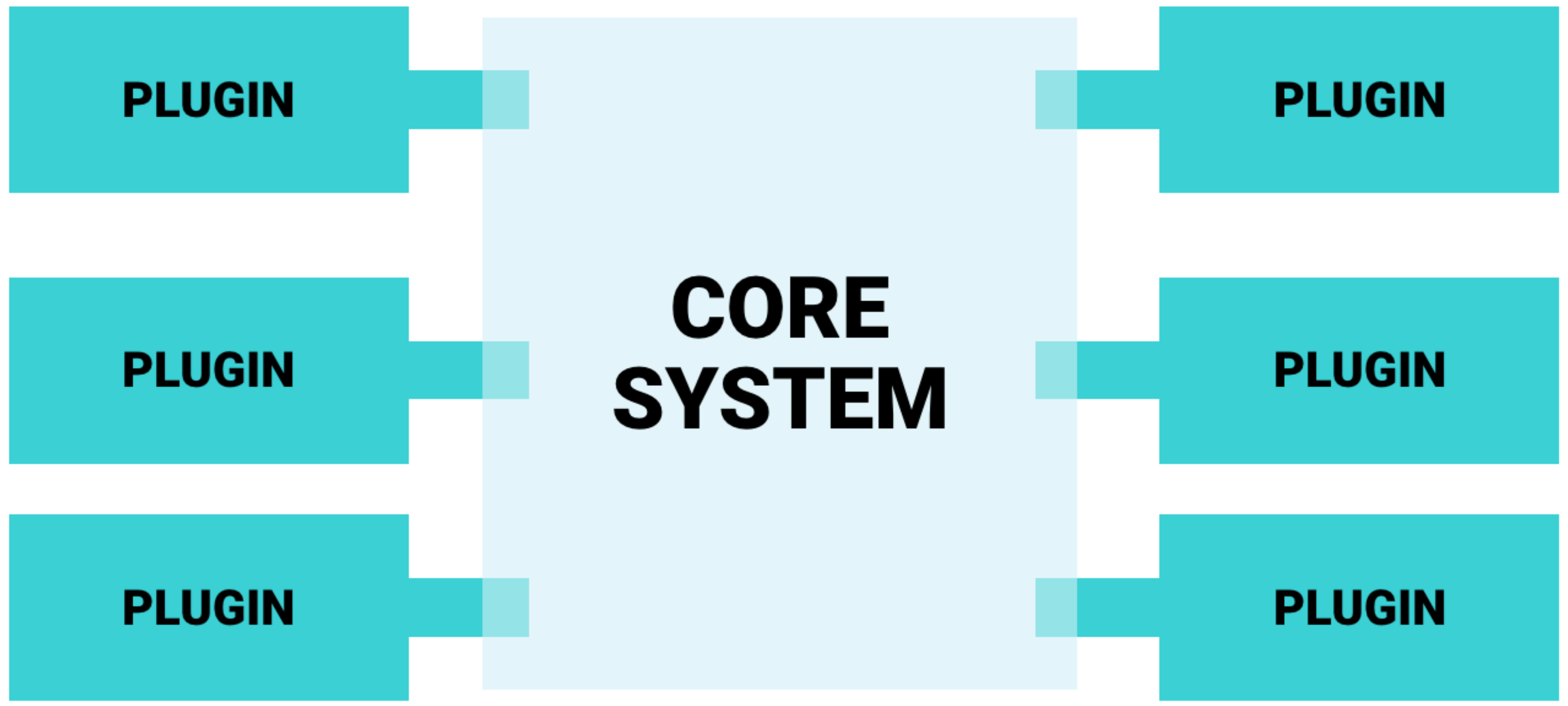
ARCHITECTURAL APPROACHES



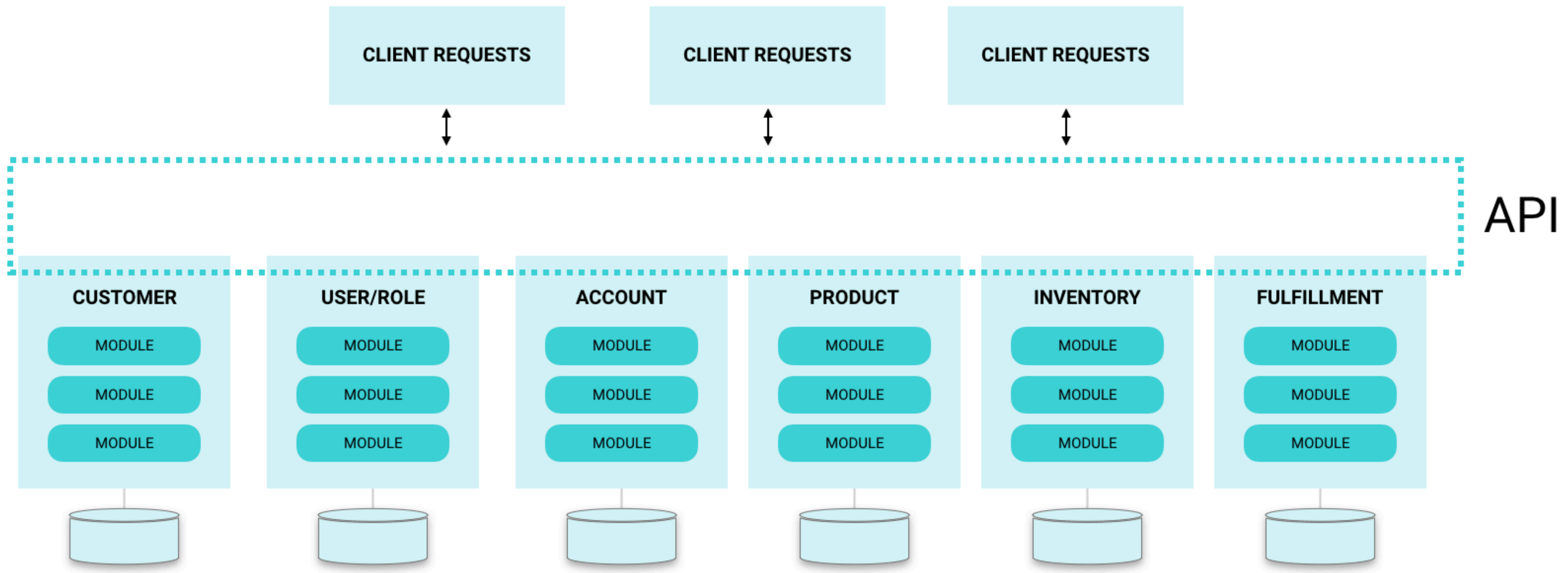
BIG BALL OF MUD



LAYERED ARCHITECTURE

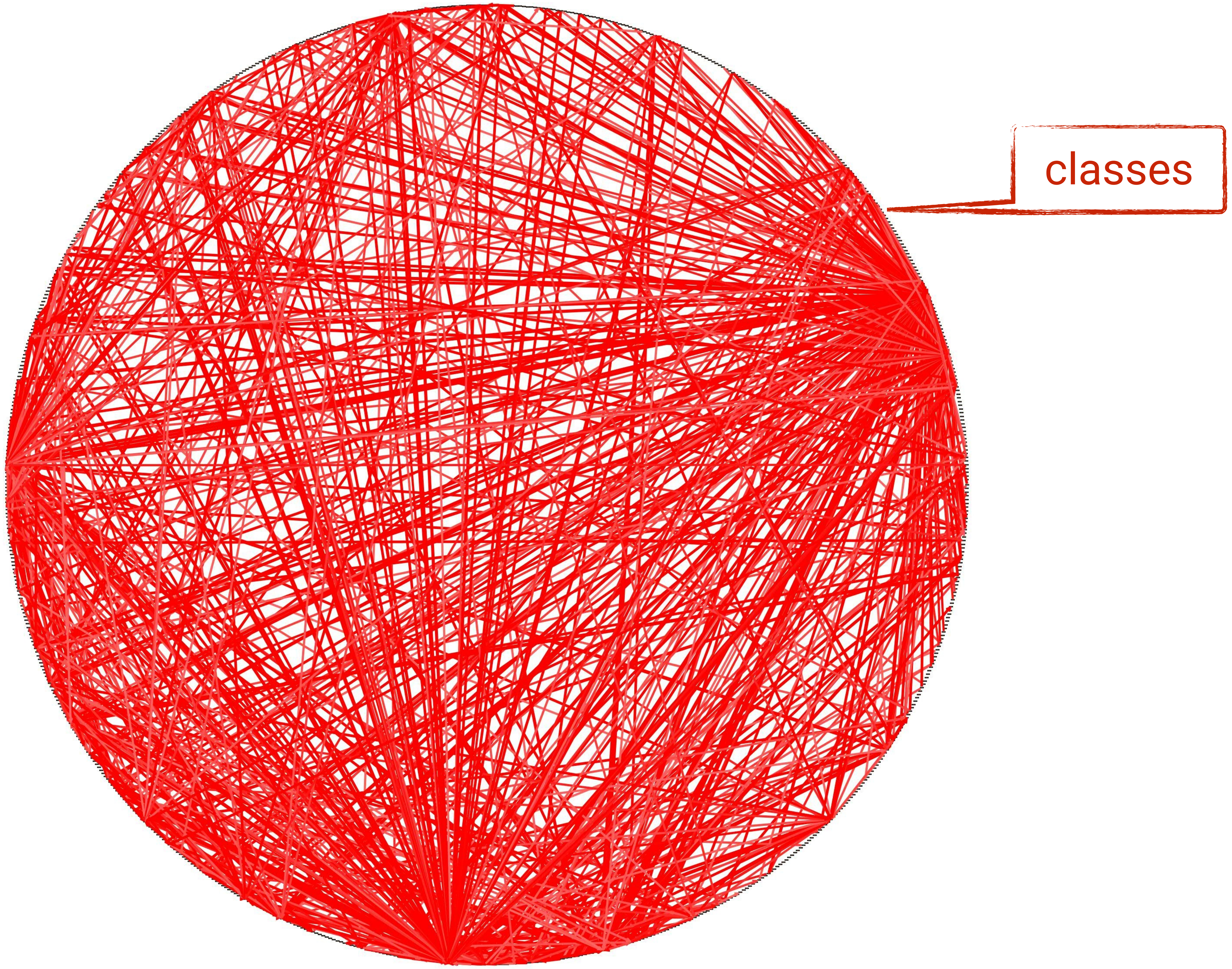


MICRO KERNEL



MICROSERVICES

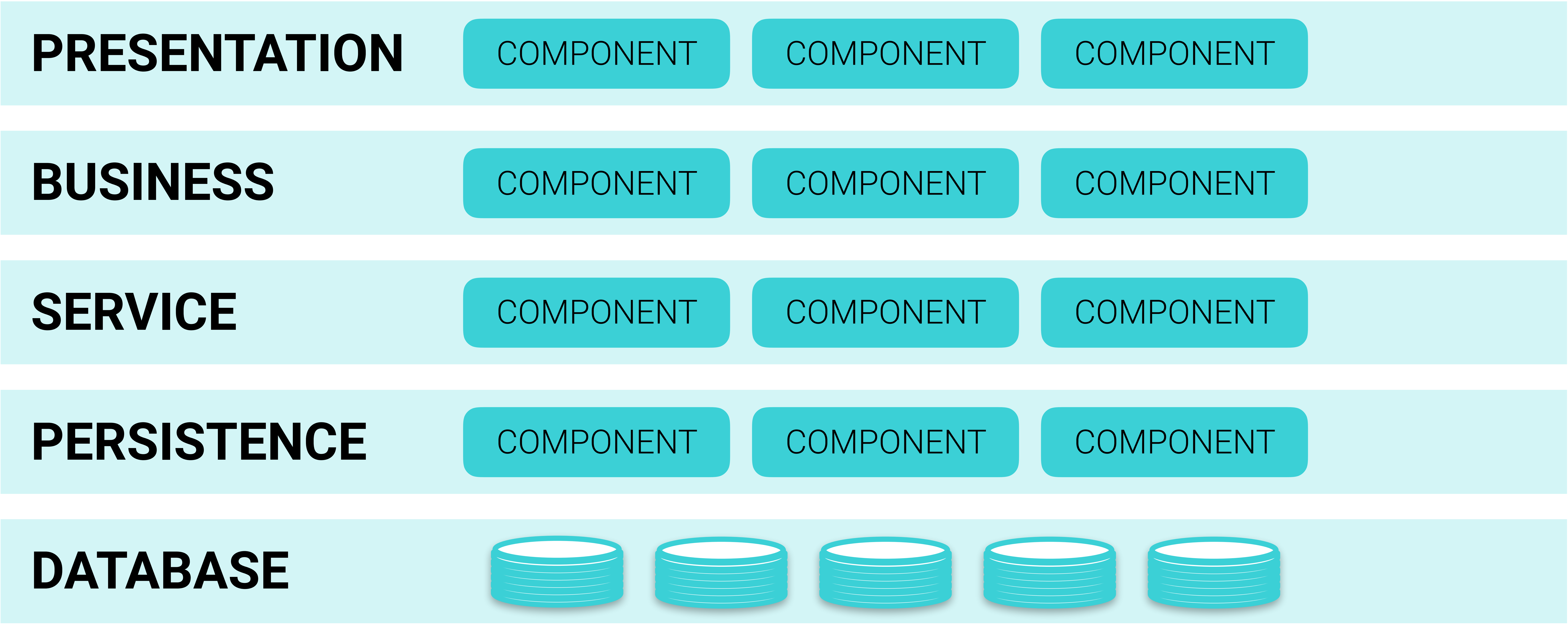
BIG BALL OF MUD



classes

coupling connections

LAYERED ARCHITECTURES



PRESENTATION COMPONENT COMPONENT COMPONENT

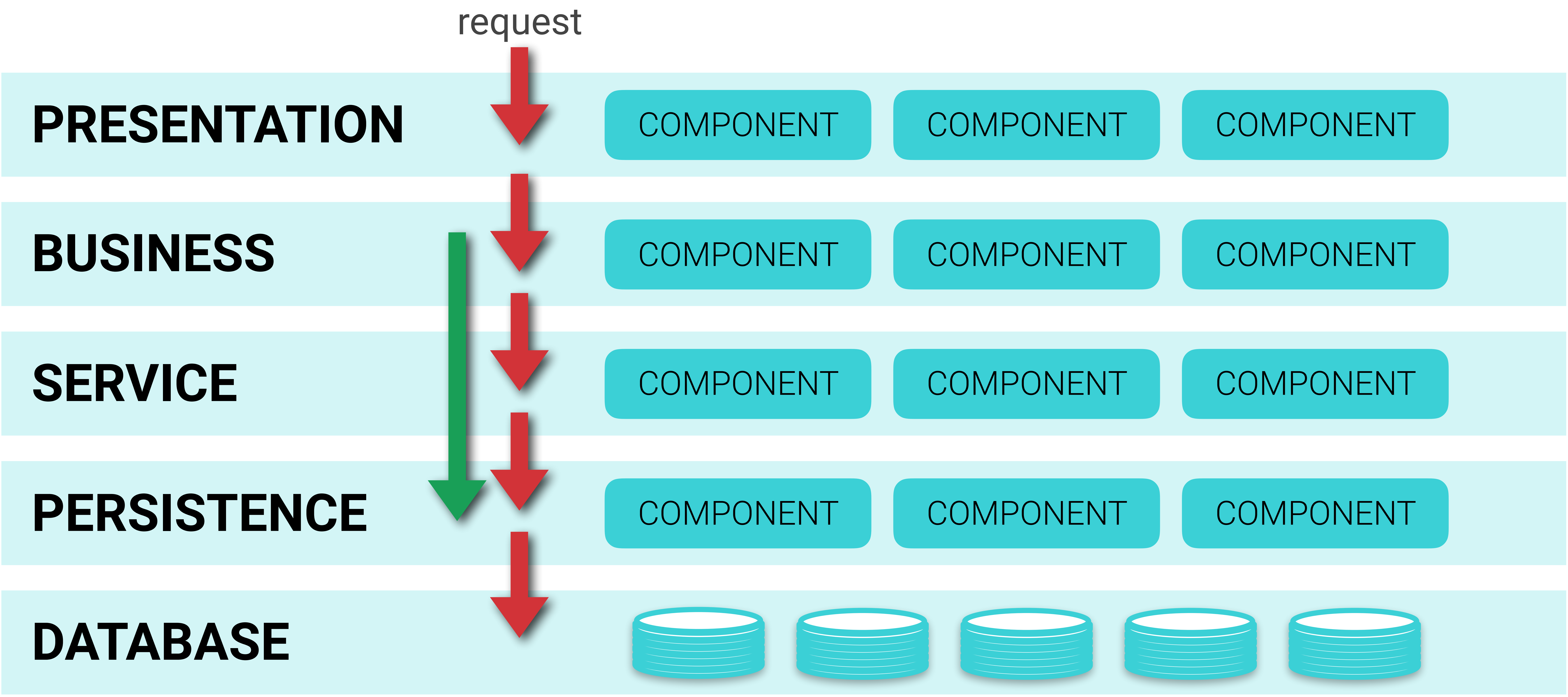
BUSINESS COMPONENT COMPONENT COMPONENT

SERVICE COMPONENT COMPONENT COMPONENT

PERSISTENCE COMPONENT COMPONENT COMPONENT

DATABASE

LAYERED ARCHITECTURES



request

PRESENTATION

COMPONENT COMPONENT COMPONENT

BUSINESS

COMPONENT COMPONENT COMPONENT

SERVICE

COMPONENT COMPONENT COMPONENT

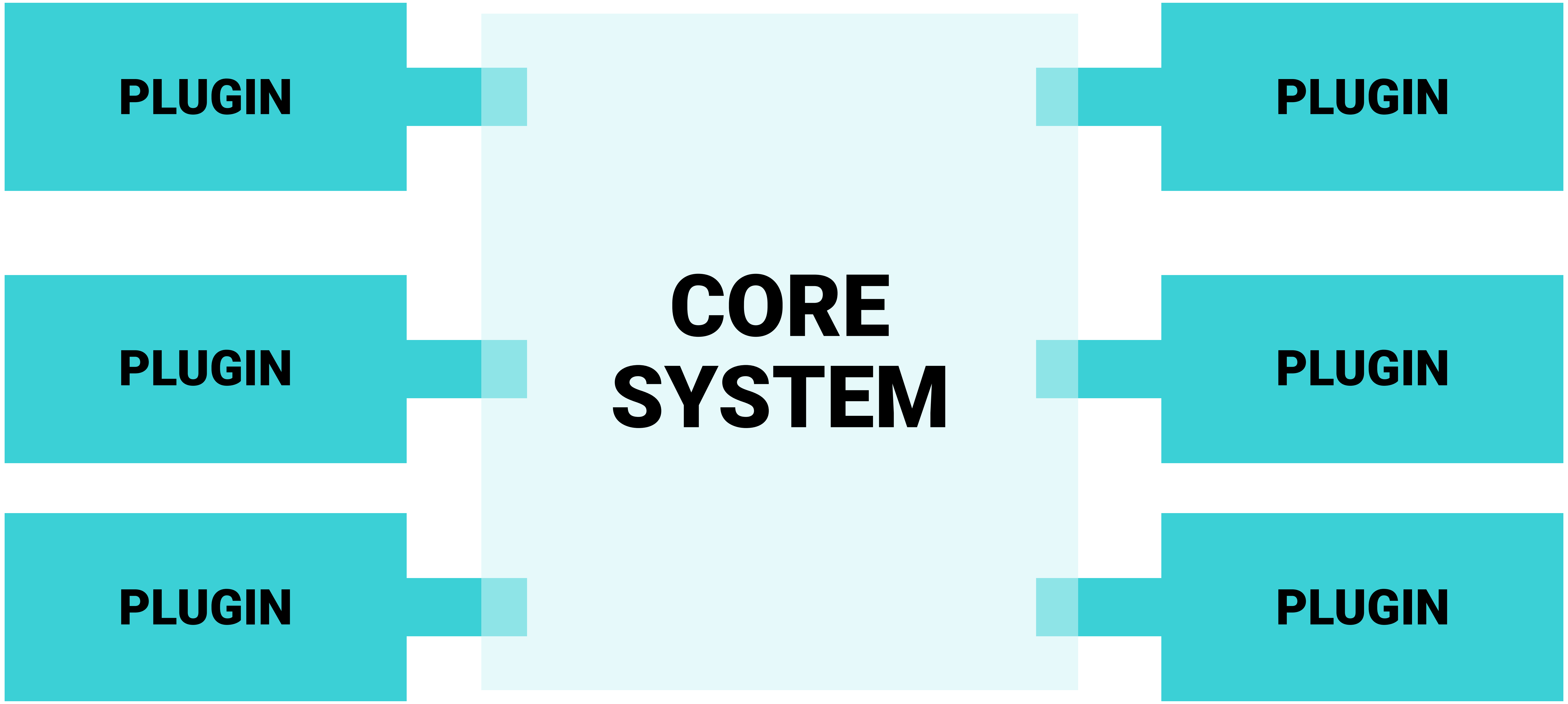
PERSISTENCE

COMPONENT COMPONENT COMPONENT

DATABASE

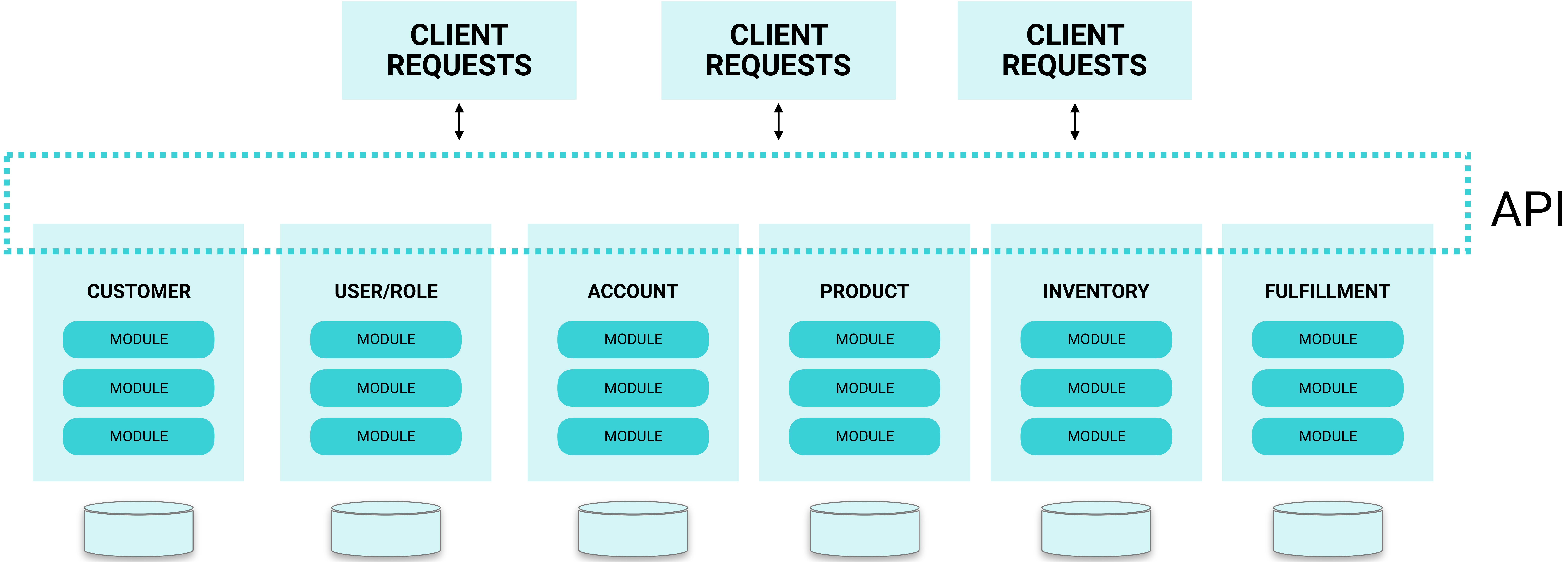
LAYERED ARCHITECTURES

MICROKERNEL



MICROKERNEL

MICROSERVICES

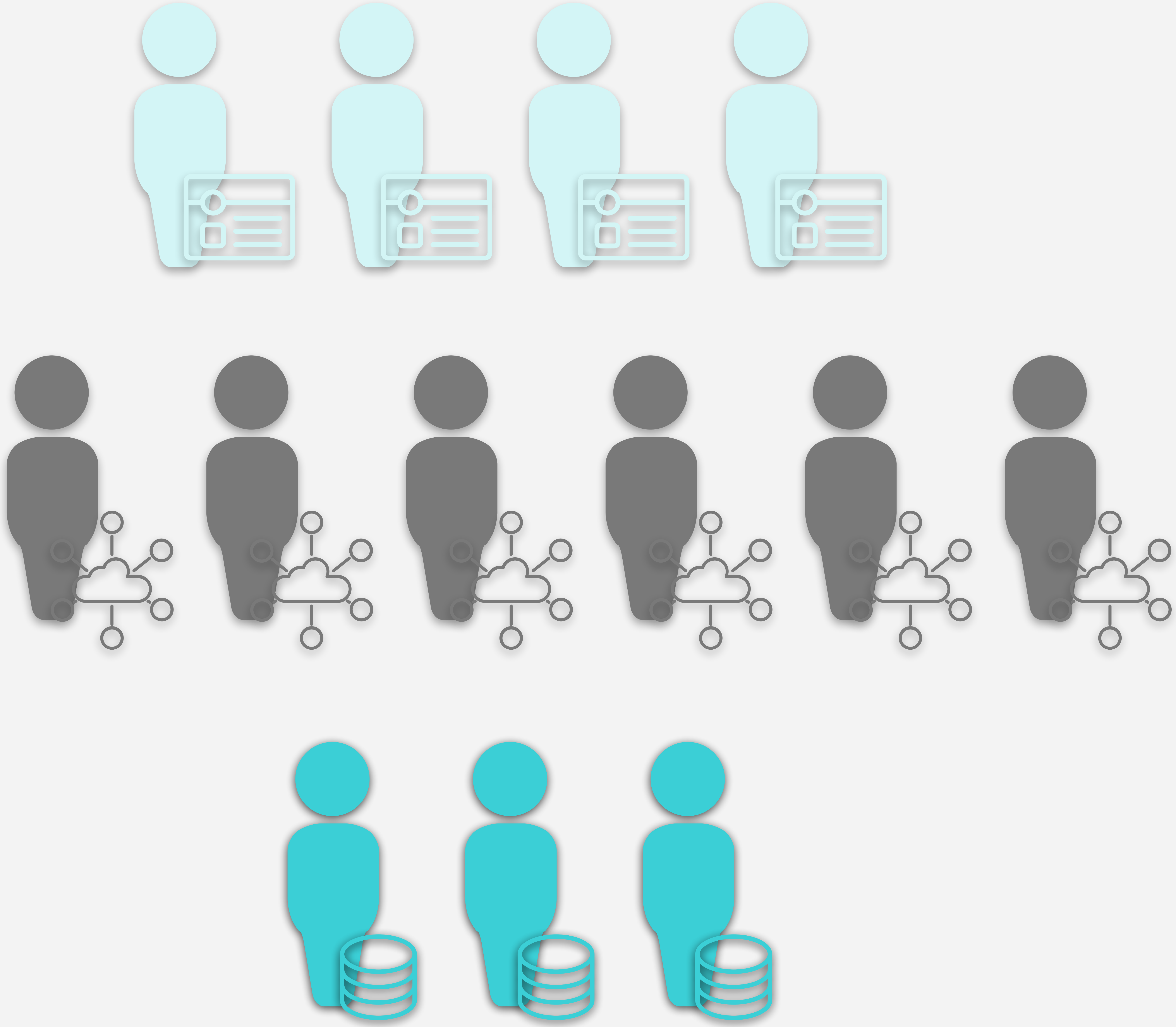


DEFINITION

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

PRACTICES

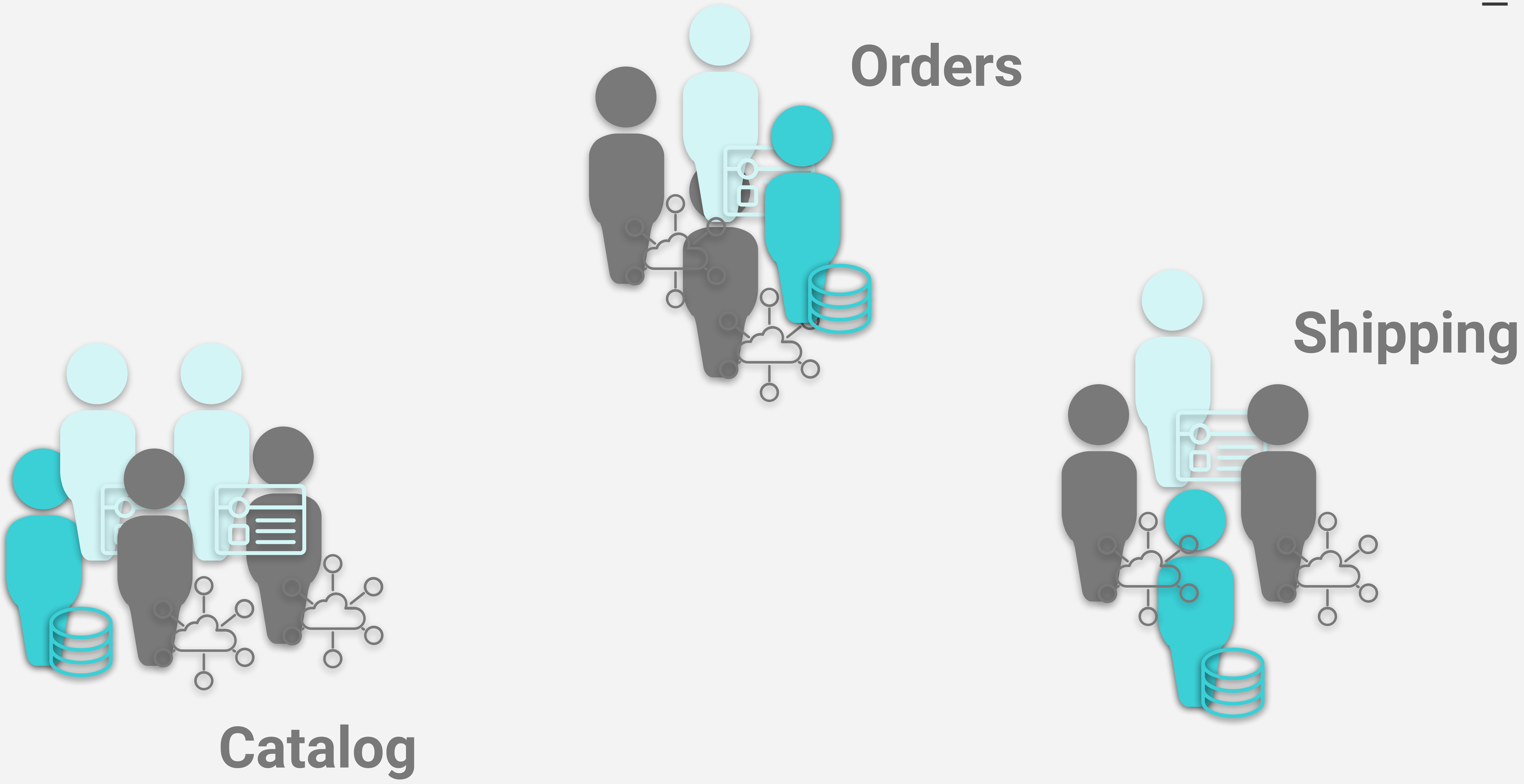
Conway’s Law:
Org Design-Architecture



User Interface

Server-side

DBAs



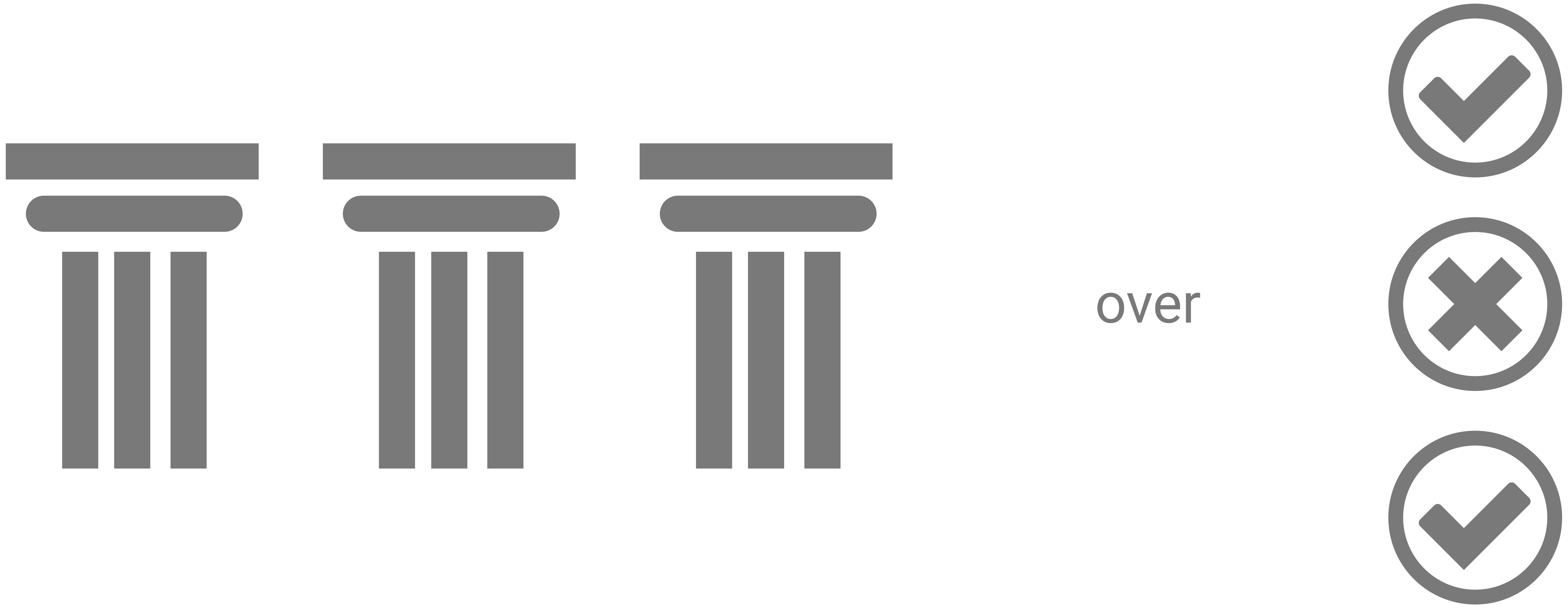
Orders

Shipping

Catalog

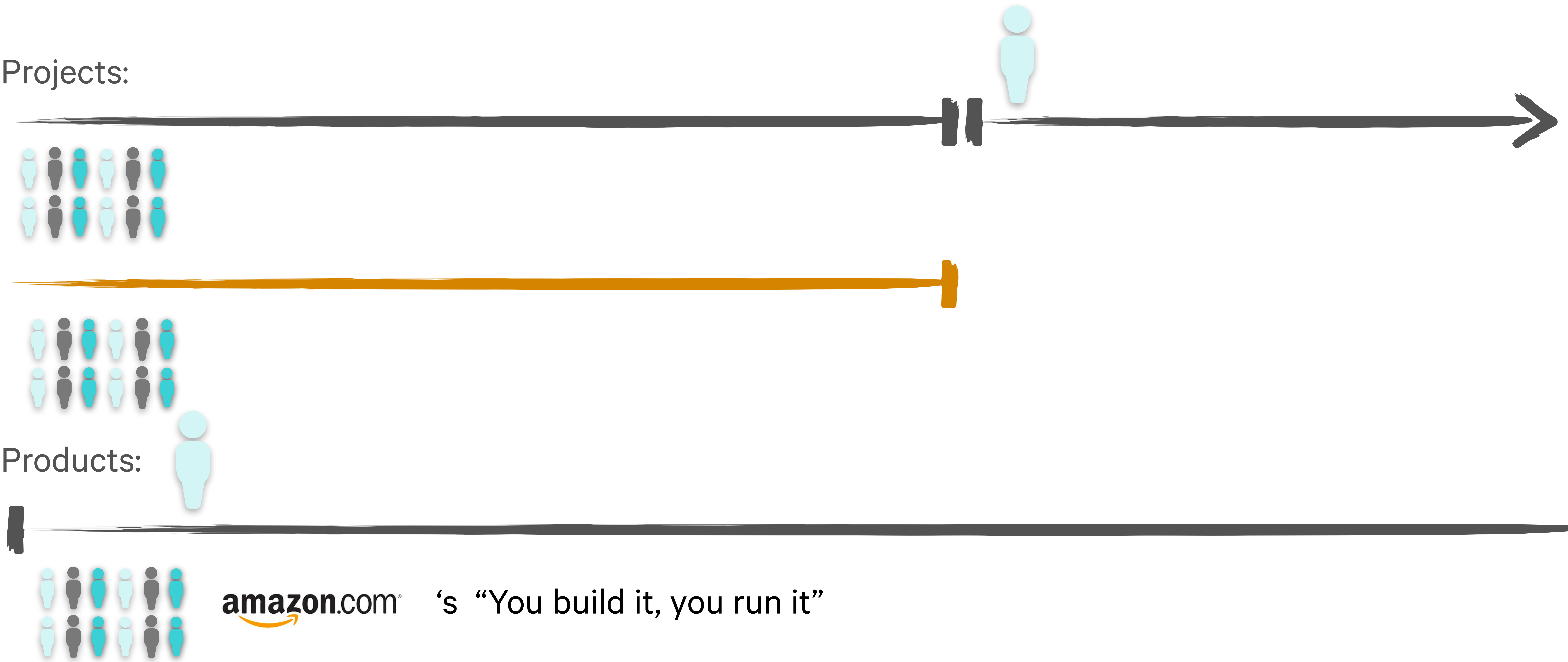
Inverse Conway Manoeuvre

PRINCIPLE-DRIVEN



over

PRODUCTS over PROJECTS

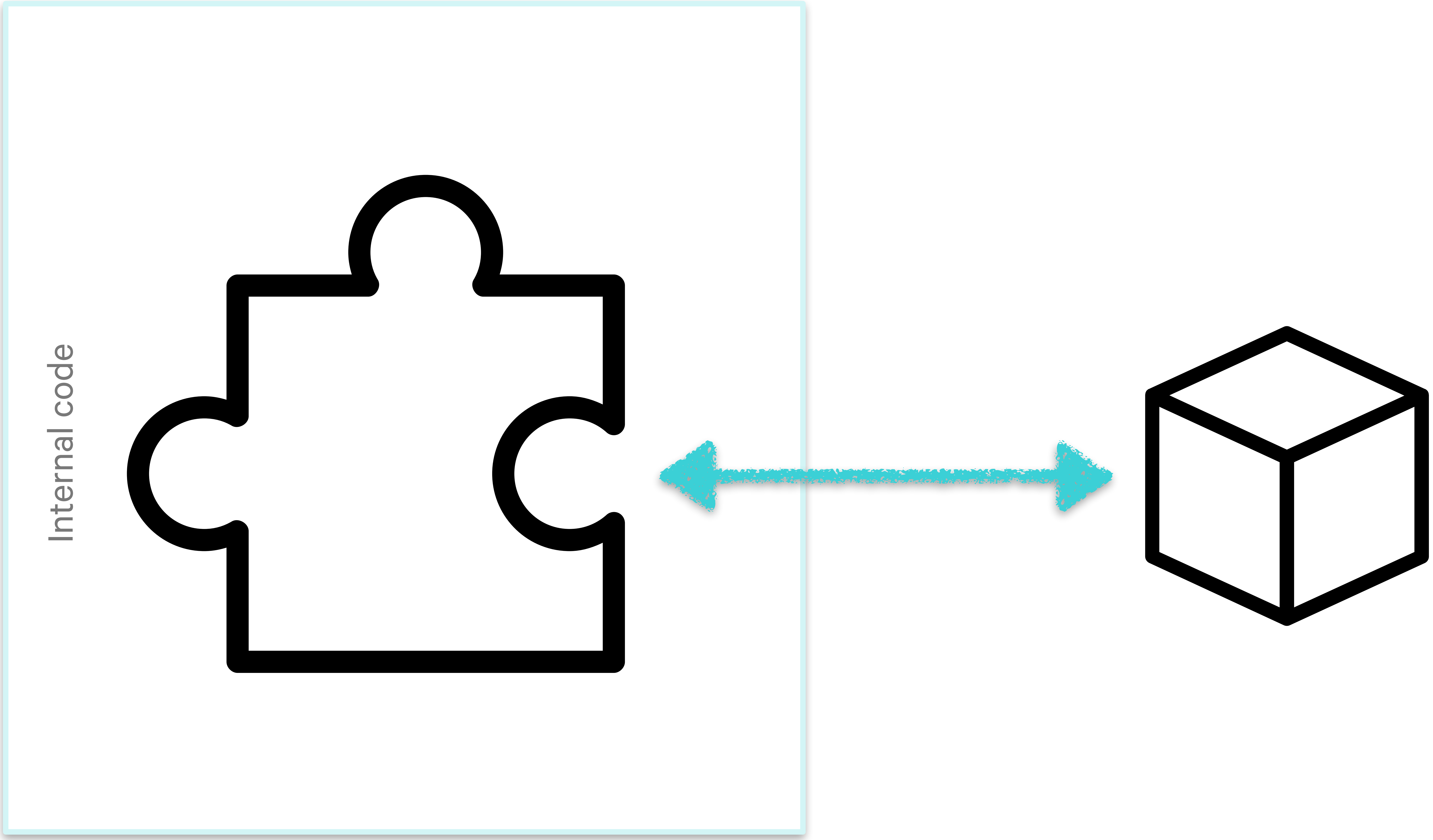


Projects:

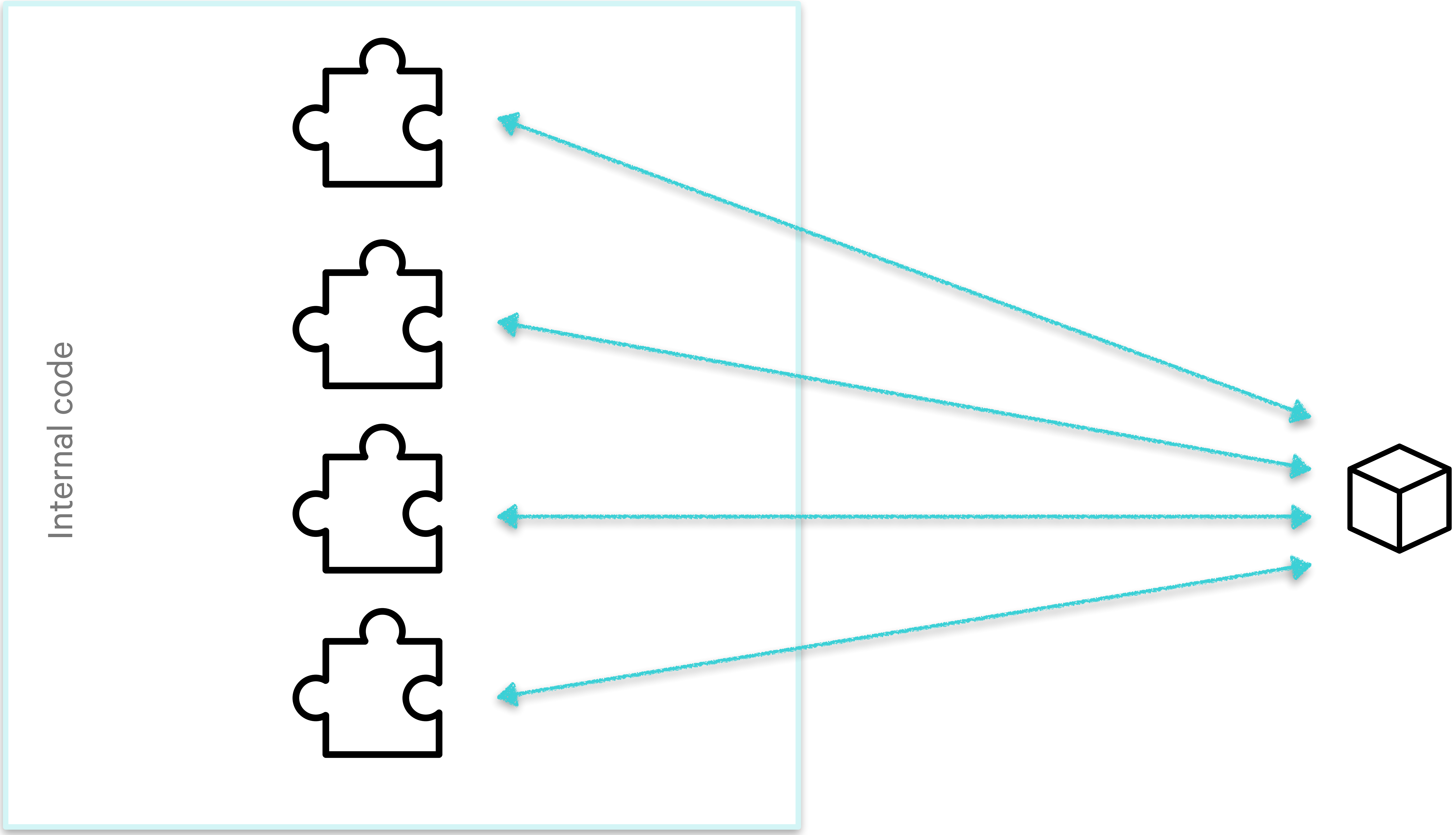
Products:

amazon.com 's "You build it, you run it"

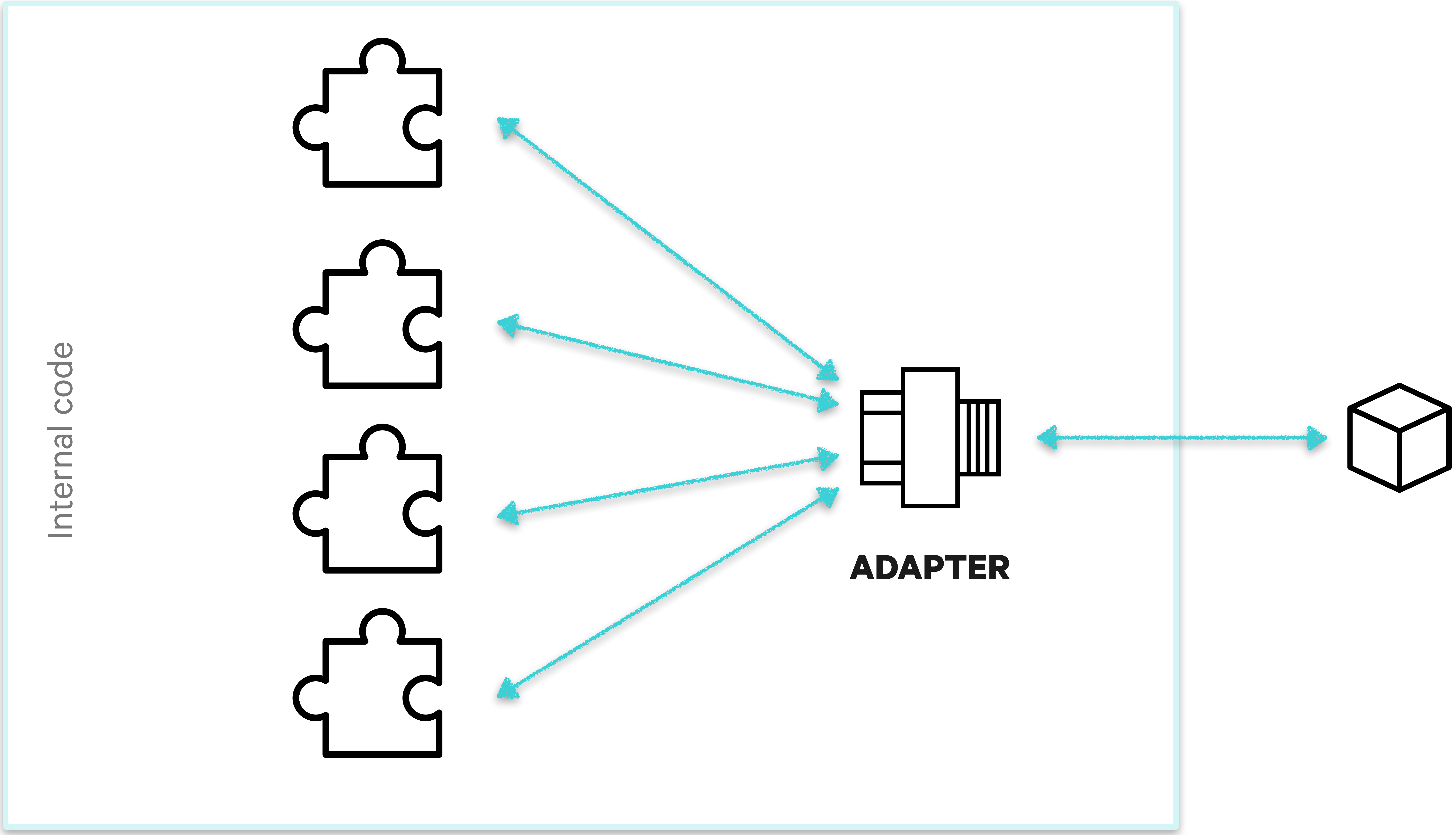
LAST RESPONSIBLE MOMENT



LAST RESPONSIBLE MOMENT

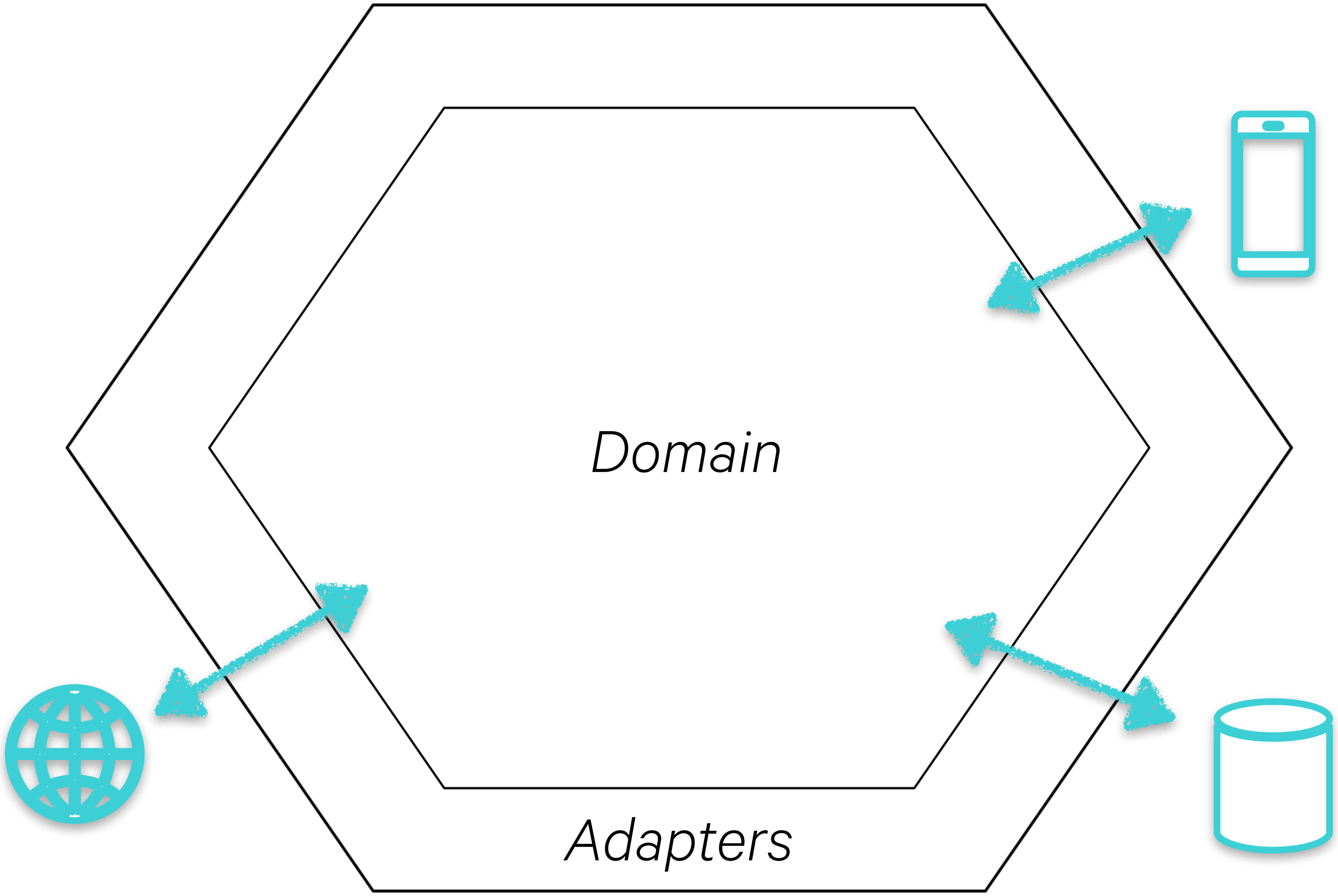


LAST RESPONSIBLE MOMENT



LAST RESPONSIBLE MOMENT

Ports and Adapters



SENSE AND PROBE

over

BUT

Not an excuse to abstract all the things

Think like a town planner

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

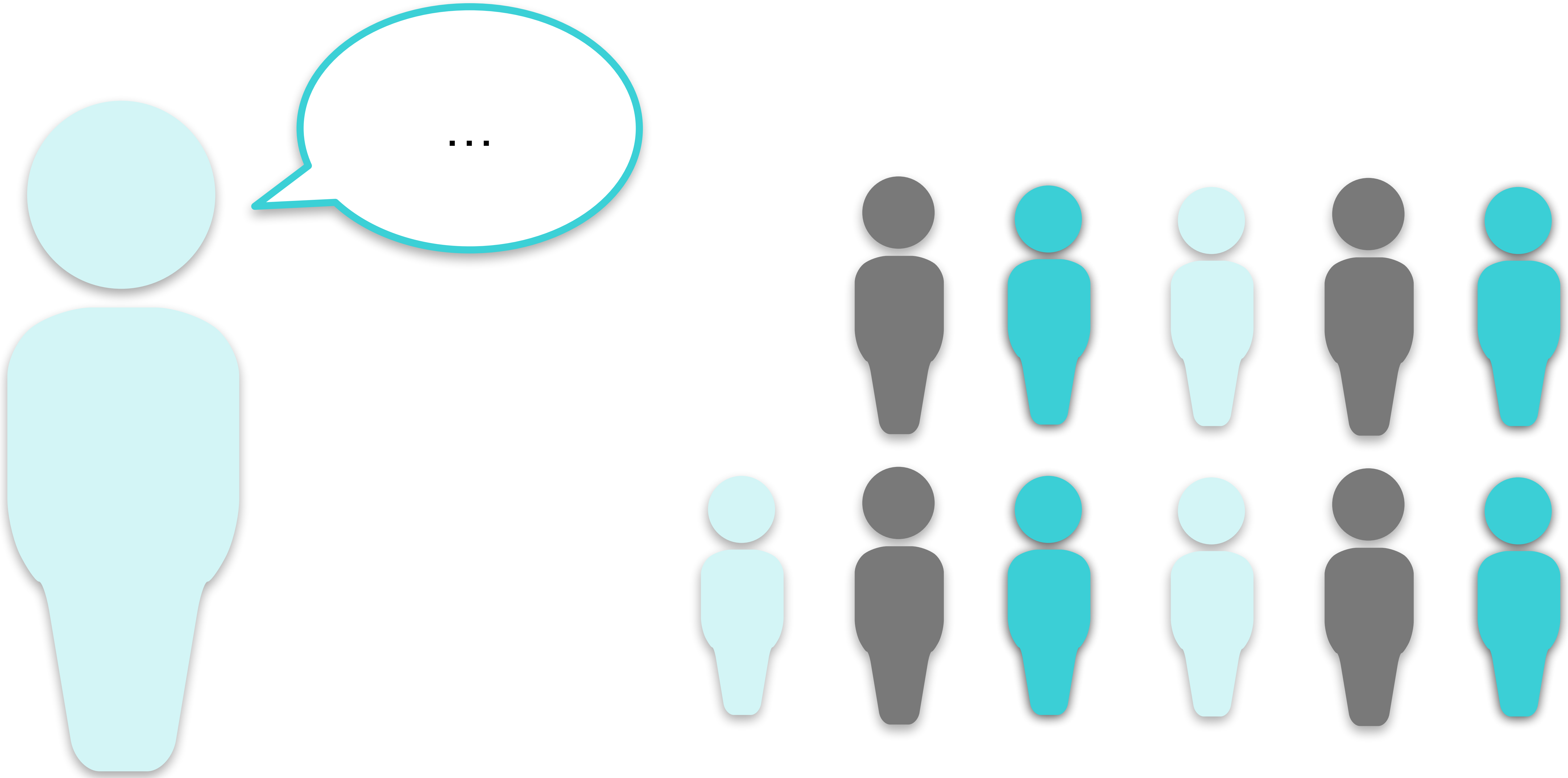
Avoid making the decisions for them.

ARCHITECTURAL BRIEFINGS

Design decision

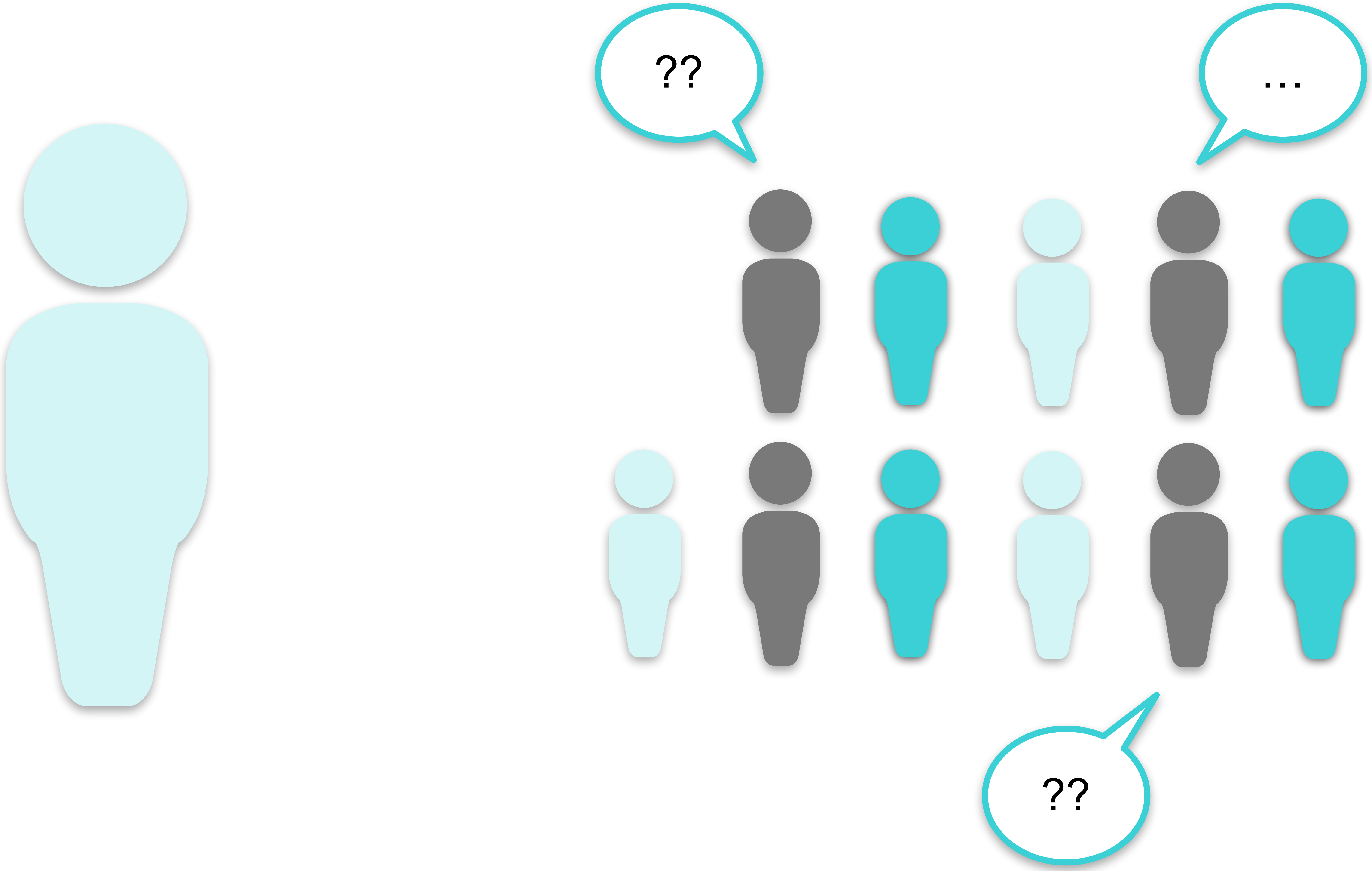
Tool

Implementation



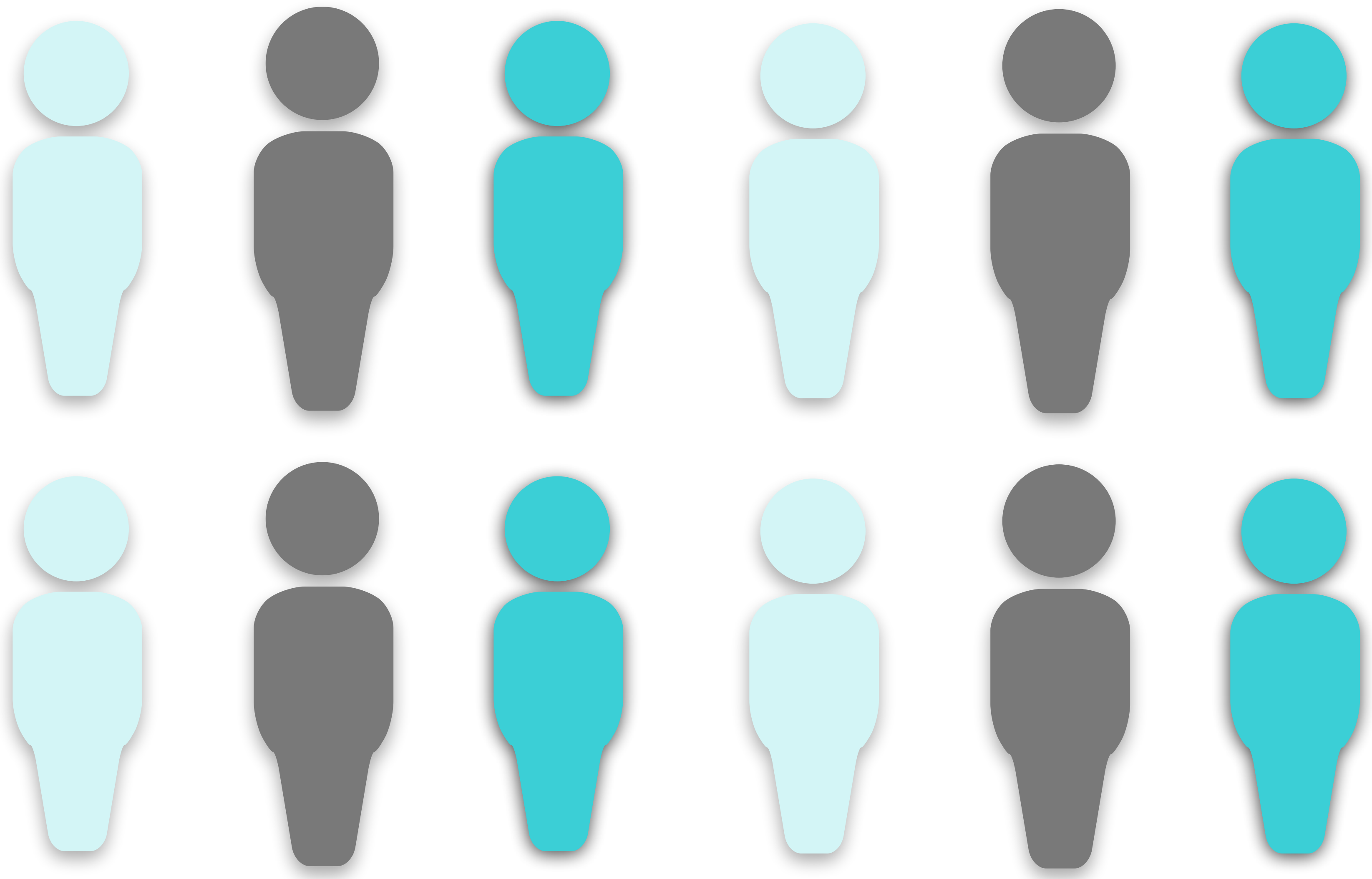
ARCHITECTURAL BRIEFINGS

Design decision
Tool
Implementation



ARCHITECTURAL BRIEFINGS

Design decision
Tool
Implementation

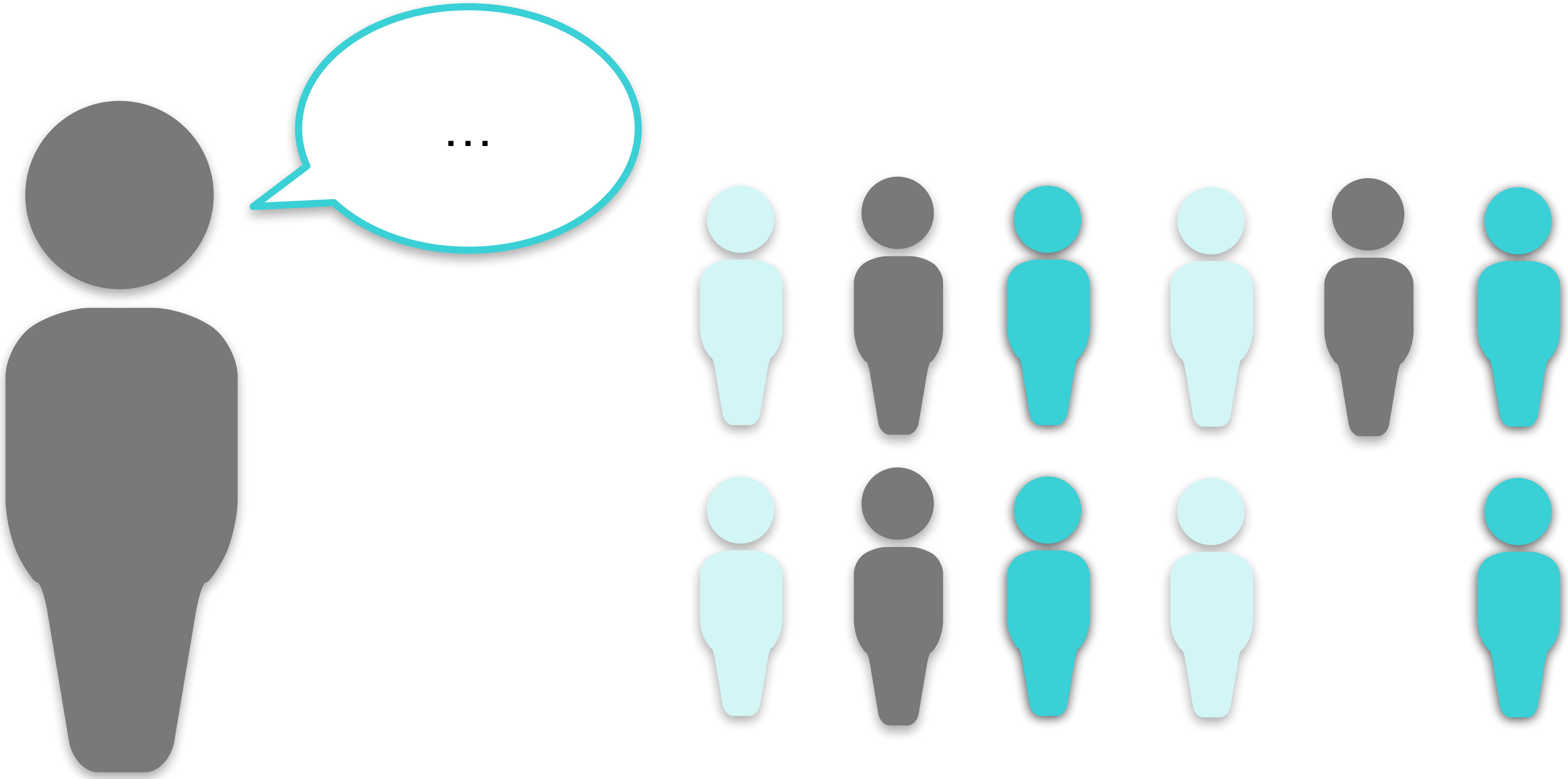


ARCHITECTURAL BRIEFINGS

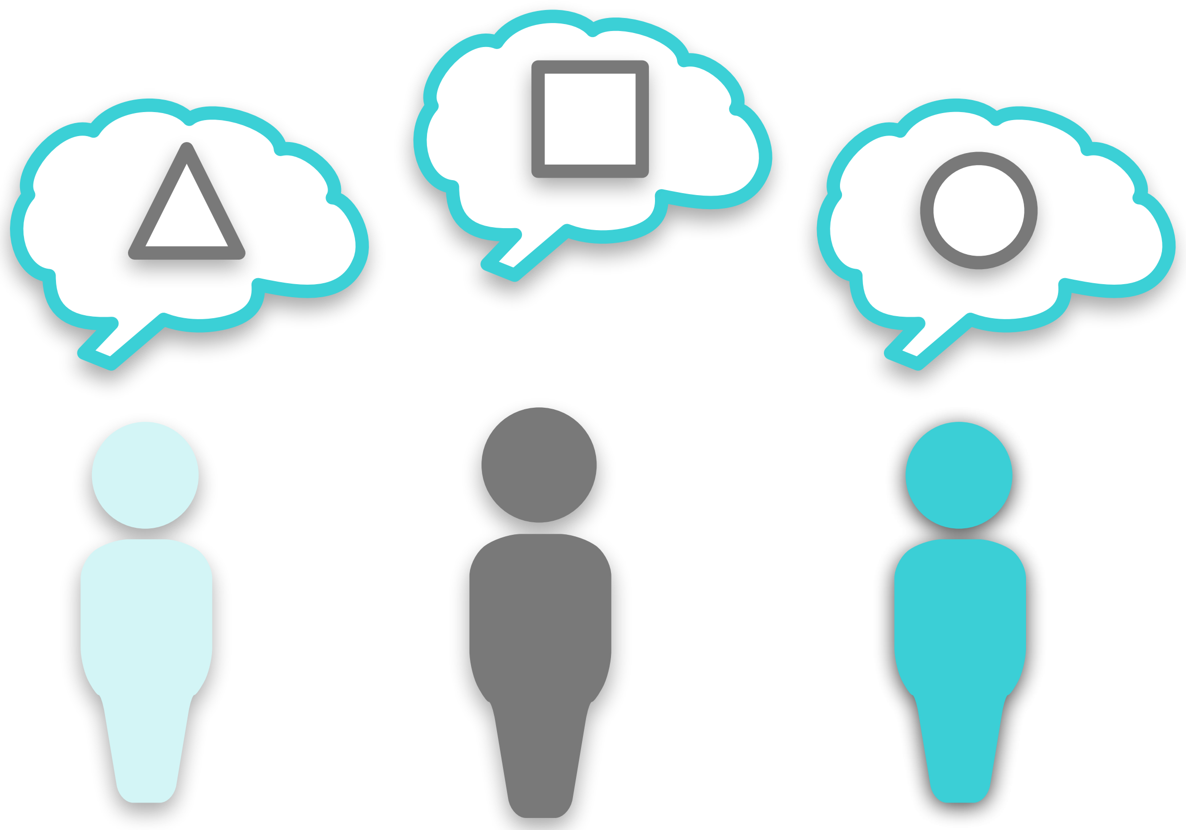
Design decision

Tool

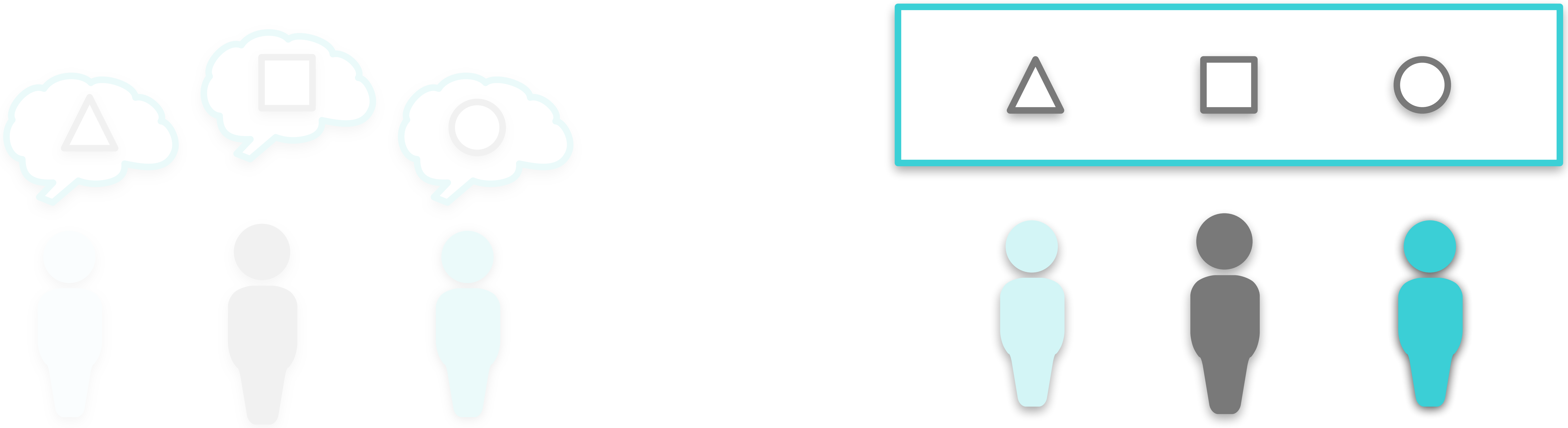
Implementation



ARCHITECTURAL BRIEFINGS

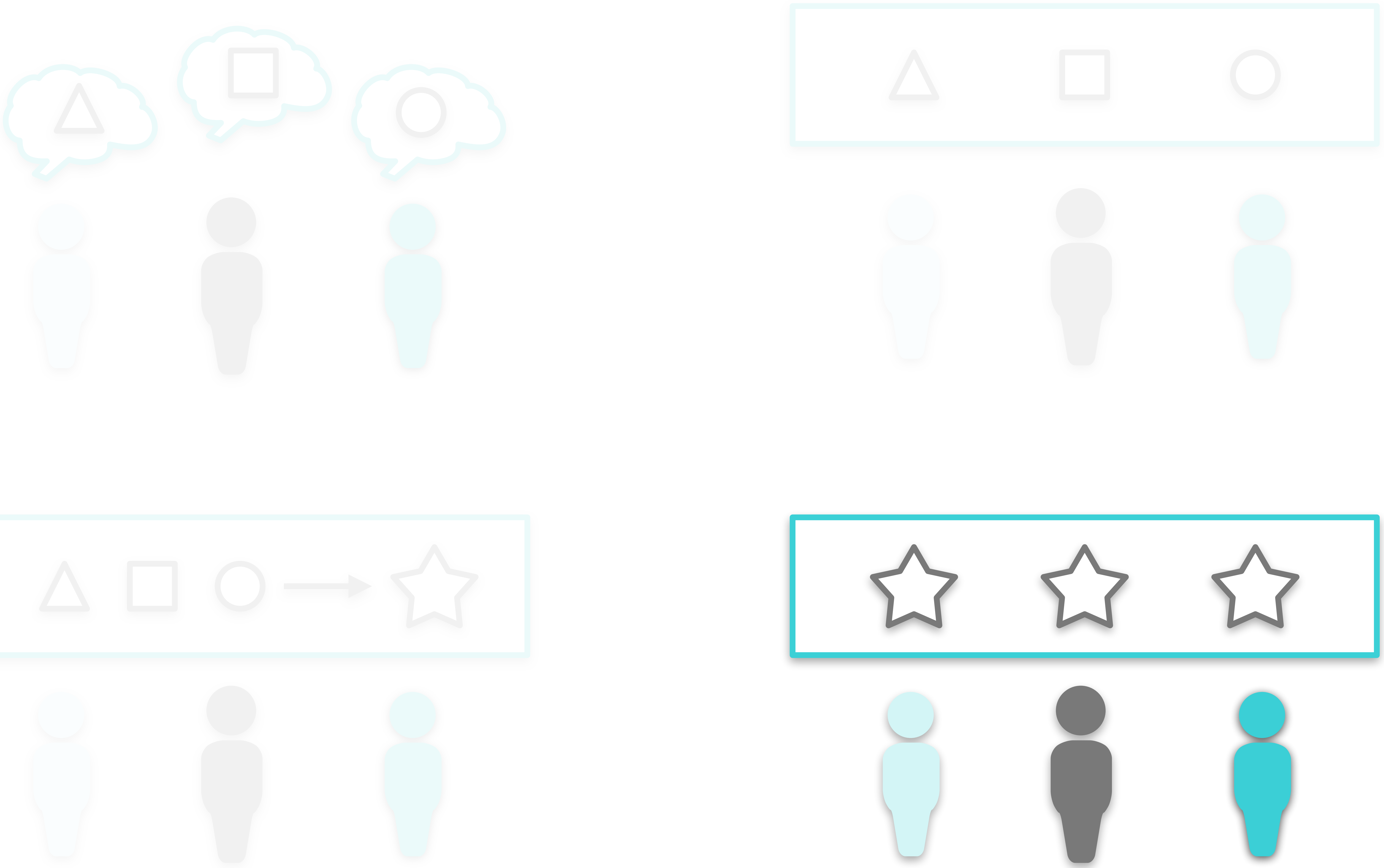


ARCHITECTURAL BRIEFINGS



ARCHITECTURAL BRIEFINGS

ARCHITECTURAL BRIEFINGS



ARCHITECTURAL BRIEFINGS

Everyone becomes an architect

Technical

Continuous Delivery ★

Support fast feedback

Appropriate coupling

Ports and Adapters ★

Automated fitness functions

Domain

Matches business capabilities

Products over projects ★

Cross functional team

Enables experimentation

Decentralised governance ★

ANTI-PATTERNS

The last 10% trap

ANTI-PATTERN

Coding via Configuration

ANTI-PATTERN

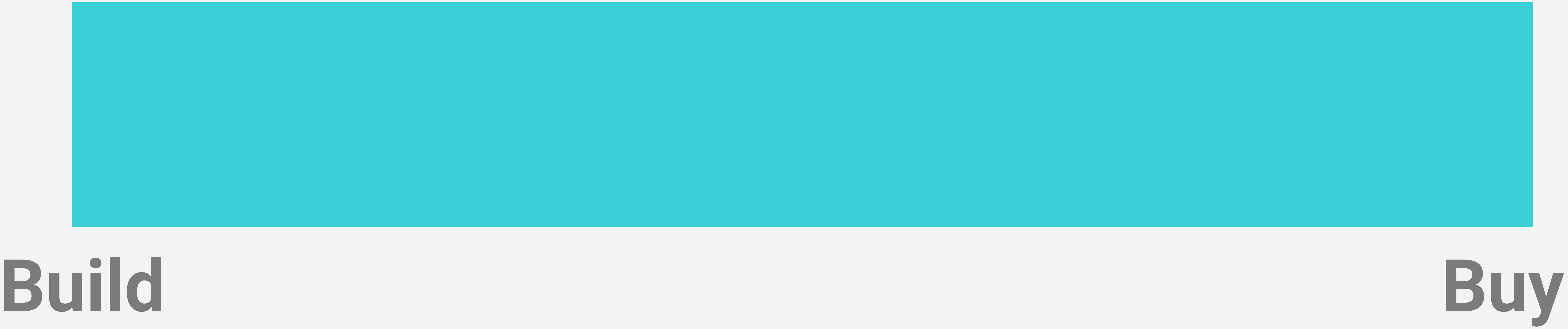
Product Customisation

ANTI-PATTERN

Exuberant Coupling

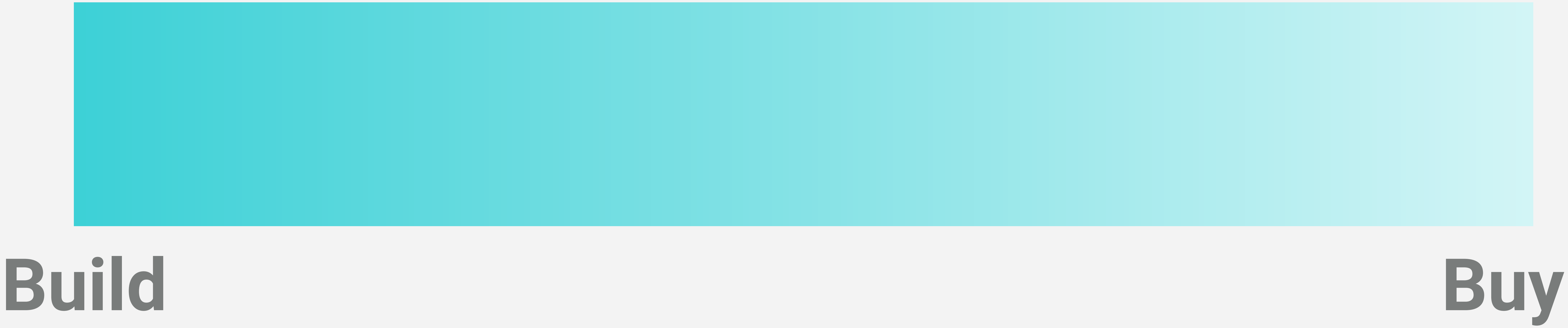
ANTI-PATTERN

CHOOSING STYLES



Build

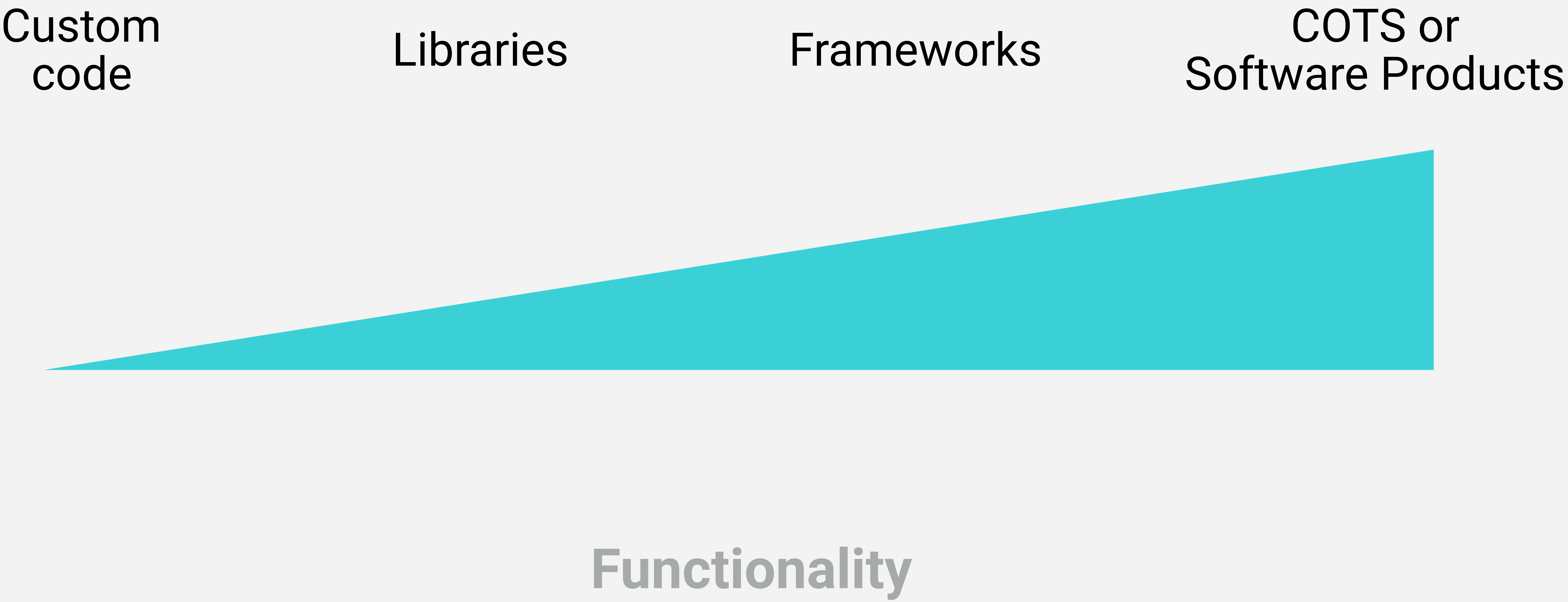
Buy



Build

Buy

Build and/or Buy



Custom code

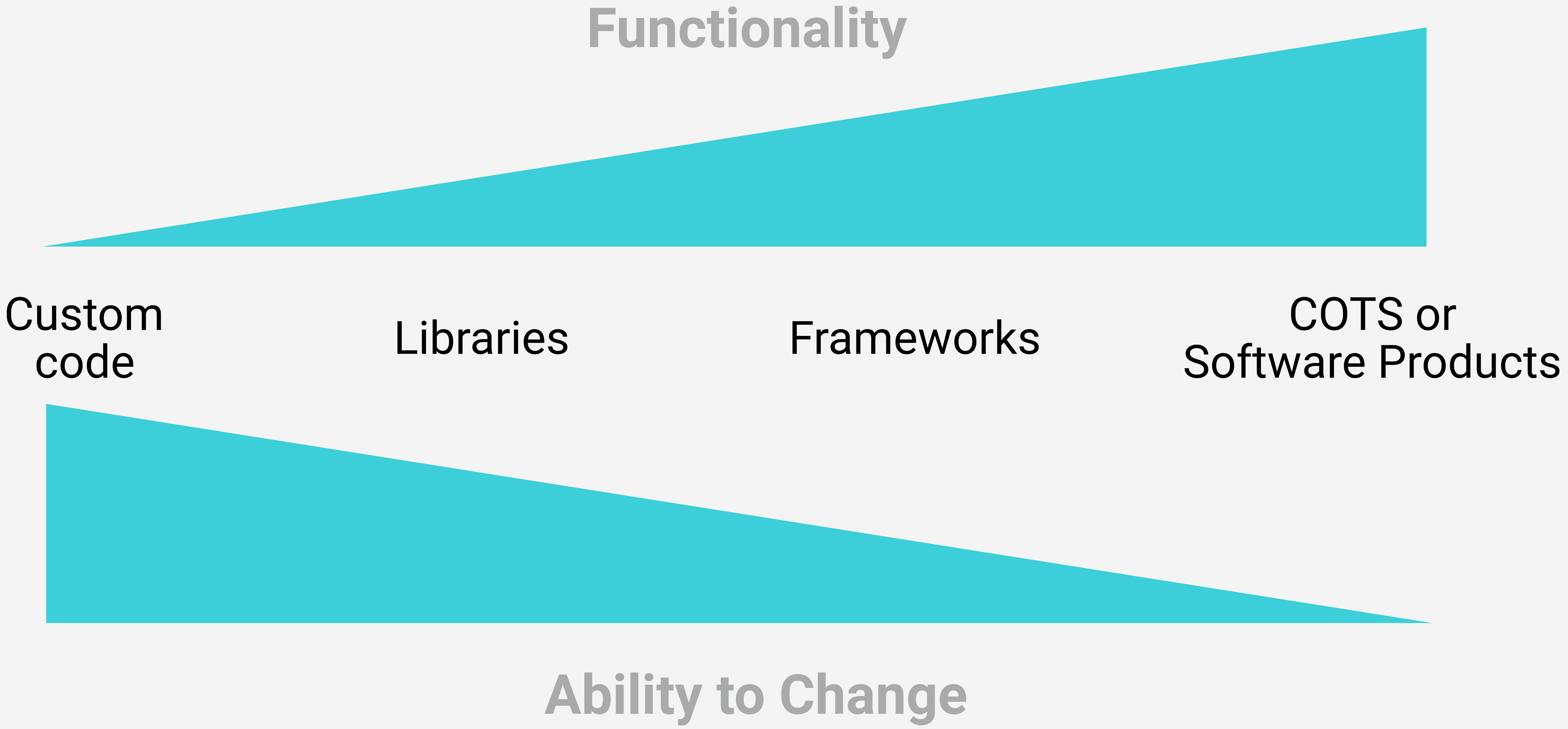
Libraries

Frameworks

COTS or Software Products

Functionality

Build and/or Buy



Functionality

Custom code

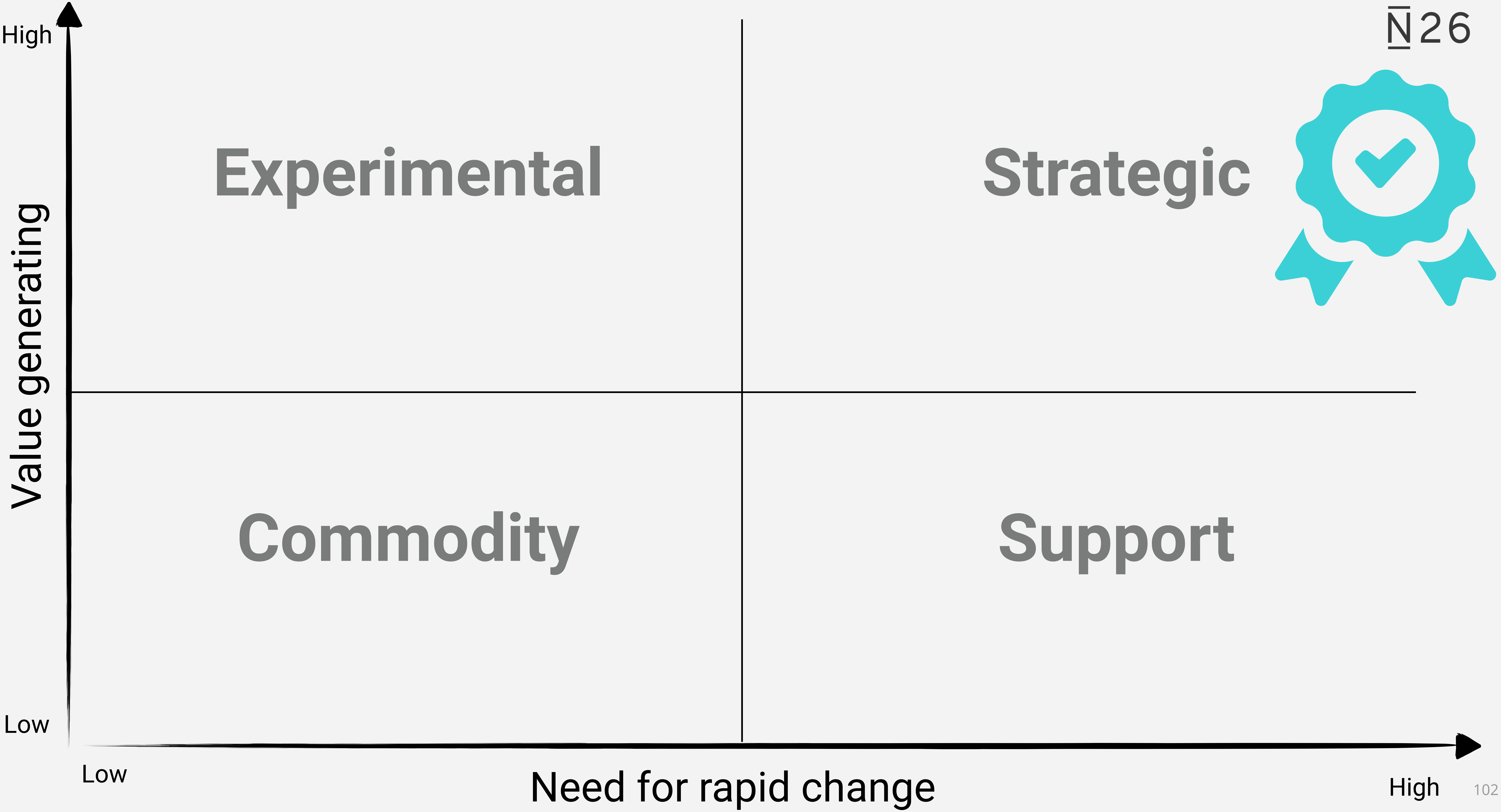
Libraries

Frameworks

COTS or Software Products

Ability to Change

N26



High

Value generating

Low

Experimental

Strategic

Commodity

Support

Low

Need for rapid change

High

CONCLUSION

DEFINITION

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

TO CONSIDER

Architectural choices

Decision making process + thinking

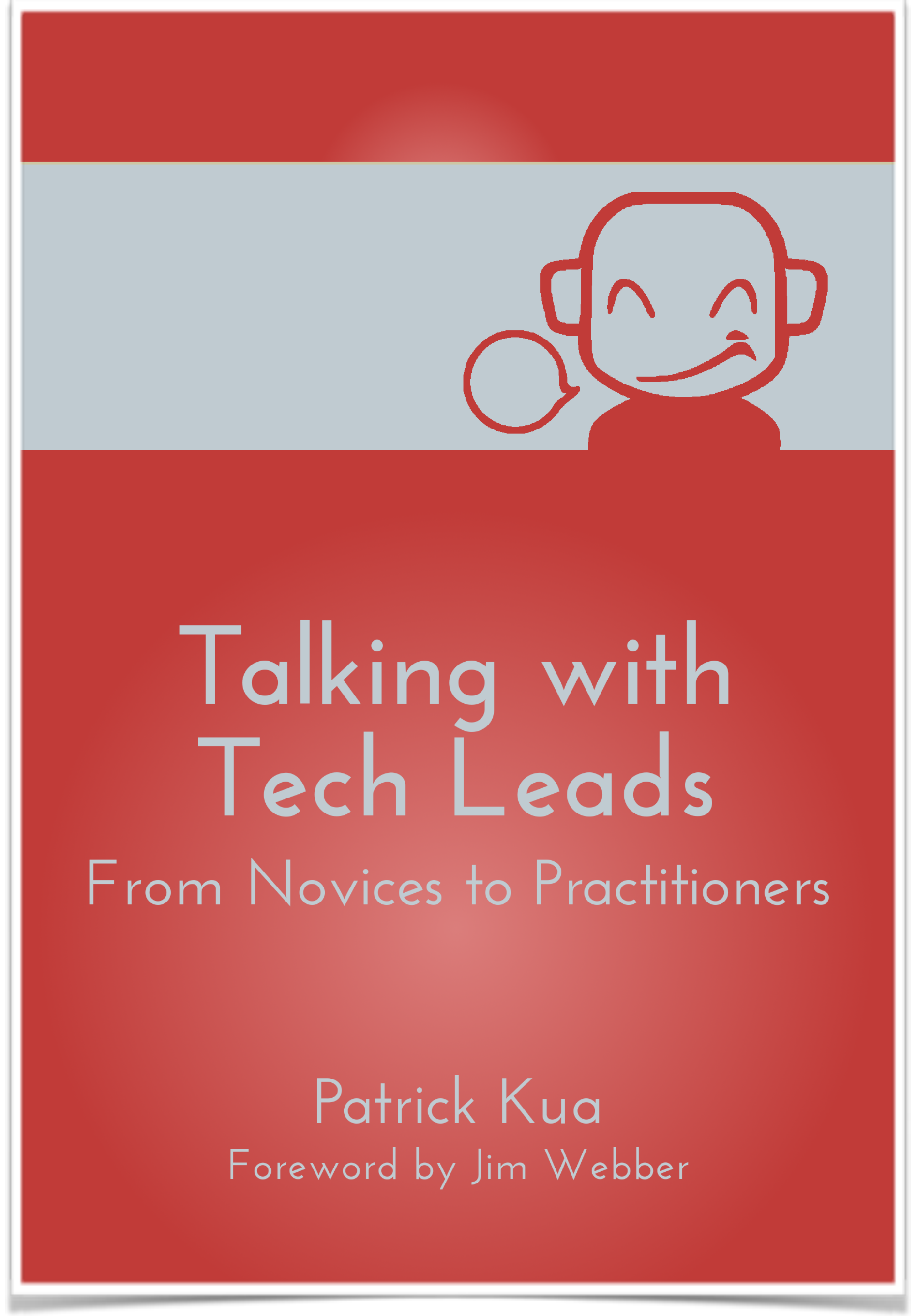
Organisational and Cultural aspects

@patkua

We’re hiring in

Berlin, Barcelona & NYC!

https://n26.com/jobs/



Talking with Tech Leads

From Novices to Practitioners

Patrick Kua

Foreword by Jim Webber

What questions do you have?