

INNOQ Tech Lunch March 2, 2022

Why "API-first" might not be a good idea

Stefan Tilkov

stefan.tilkov@innoq.com

@stilkov

INNOQ

<https://www.innoq.com/>

The API-Driven Enterprise

New applications can
easily created by
orchestrating APIs

Responsibilities are
unambiguously defined

Functionality and data
are re-used

Time-to-market
is reduced

Apps can be created
by 3rd parties

Support for multiple different
(existing and new) channels

Business logic decoupled from user
interfaces and channel details

APIs – The Holy Grail



API-First Approach

**Clearly-defined
contracts before
implementation**

**Standard versioning
and documentation
strategy**

**Schema and
contract reuse**

**Powerful API
management
software**

**Growing pool of useful
and reusable APIs**

Top 10 of Strategic API-first Problems

1. Lack of a clear API strategy
2. Poor API design
3. Inconsistent API versions
4. Lack of API documentation
5. Poor API performance
6. Lack of API security
7. Poor API discoverability
8. Lack of API governance
9. Poor API interoperability
10. Lack of API ecosystem

10.

**You cannot build an API with useful functionality
based on guesses about user needs**

9.

If APIs are driven by underlying system capabilities, they will be only as good or bad as they are

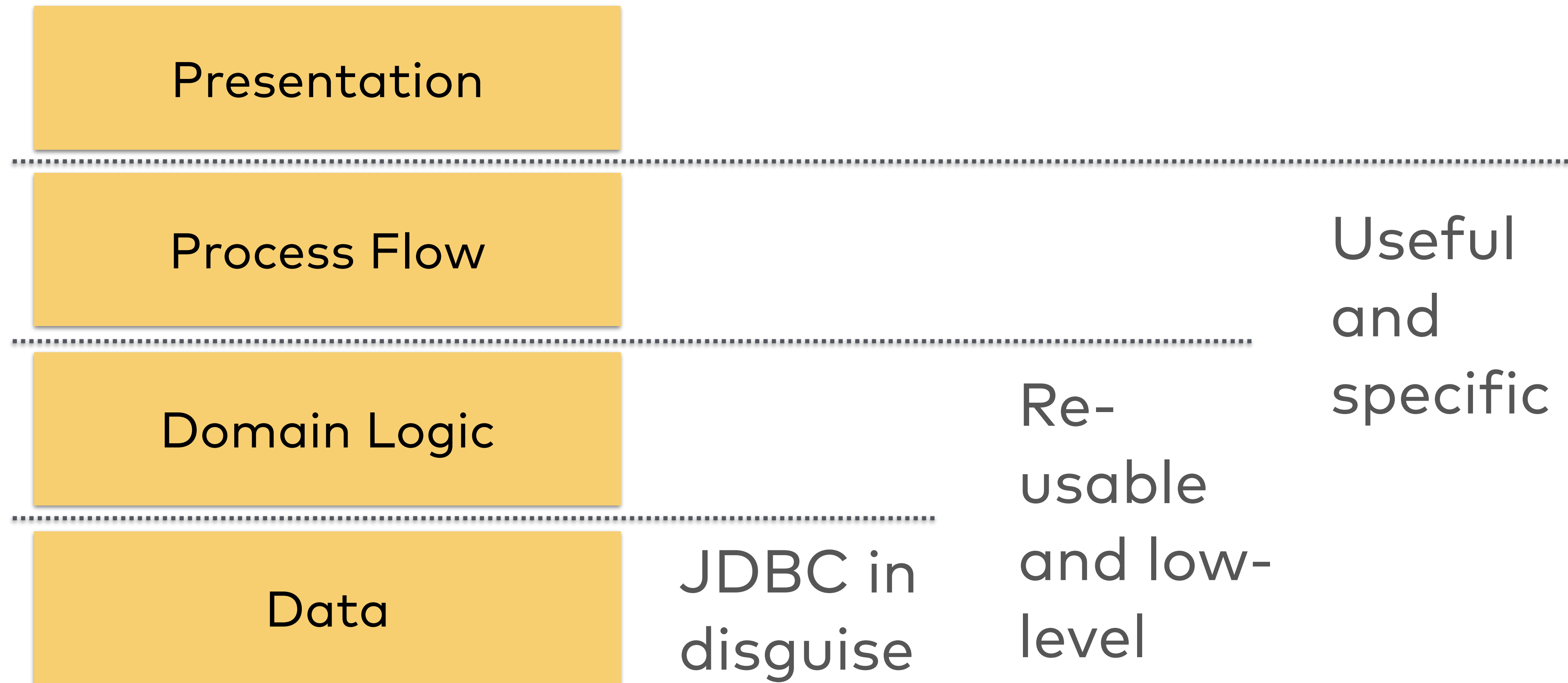
8.

Abstract APIs are easy to build, much easier than analyzing stakeholder needs

7.

Re-usable services often end up being way too generic and data-oriented

Building services that matter



6.

APIs are often abused to cement boundaries that should not be there in the first place

Antipattern: Horizontal Conway Split

Group 1

Motto: »Java is a legacy programming language used in the last century«

Group 2

Motto: »Obviously, you can't build correct programs with JavaScript«

HTML/CSS/JavaScript Frontend

JSON API

Java Backend

**Every meaningful feature development
requires frontend and backend work**

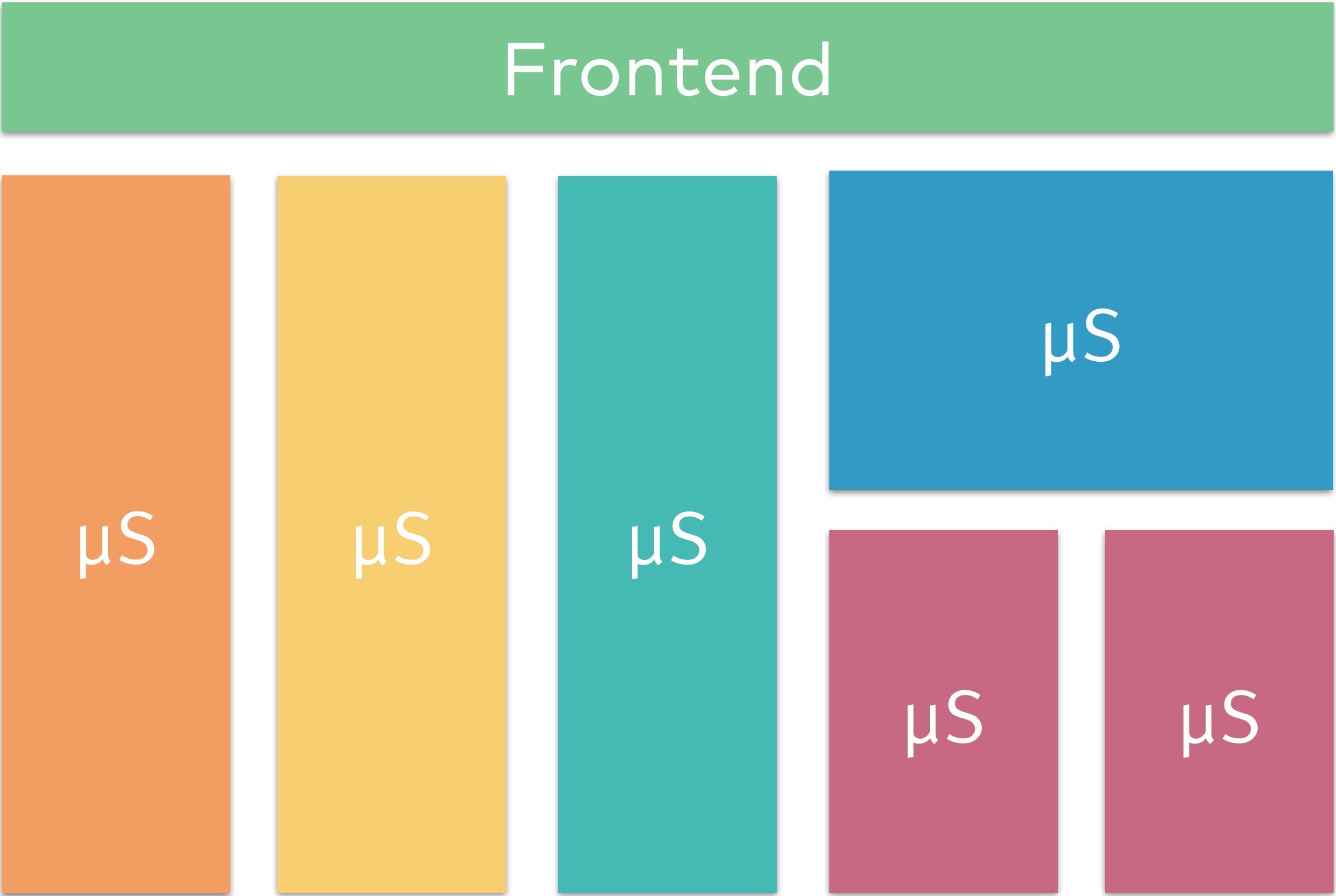
**When faced with unresponsive backend teams,
frontend teams quickly become full-stack teams**

If teams don't want to collaborate, a well-defined interface contract addresses symptoms only

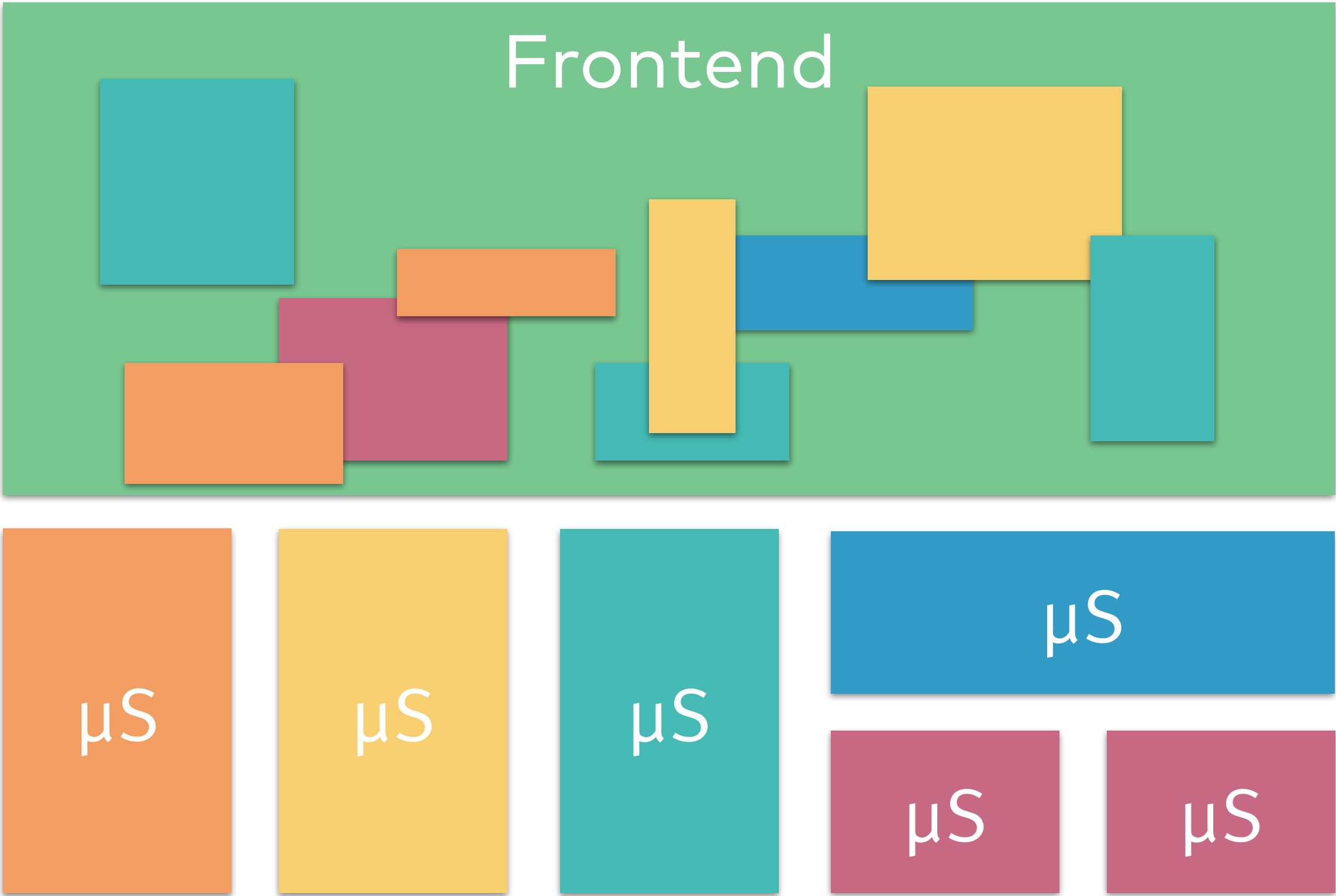
5.

APIs almost never fulfill user needs unless they're either adapted or massively orchestrated

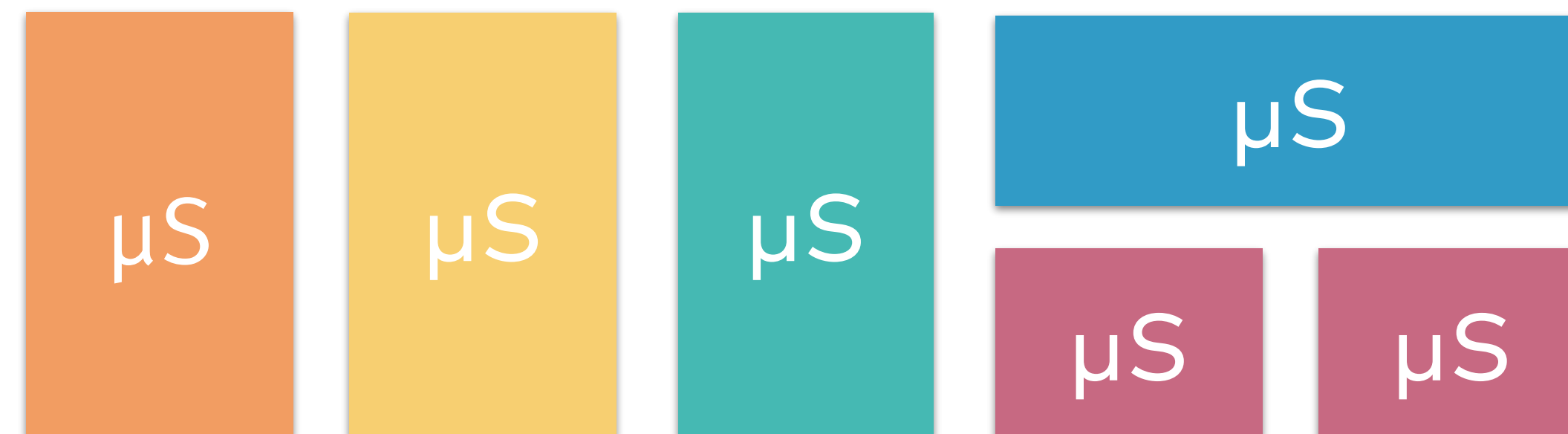
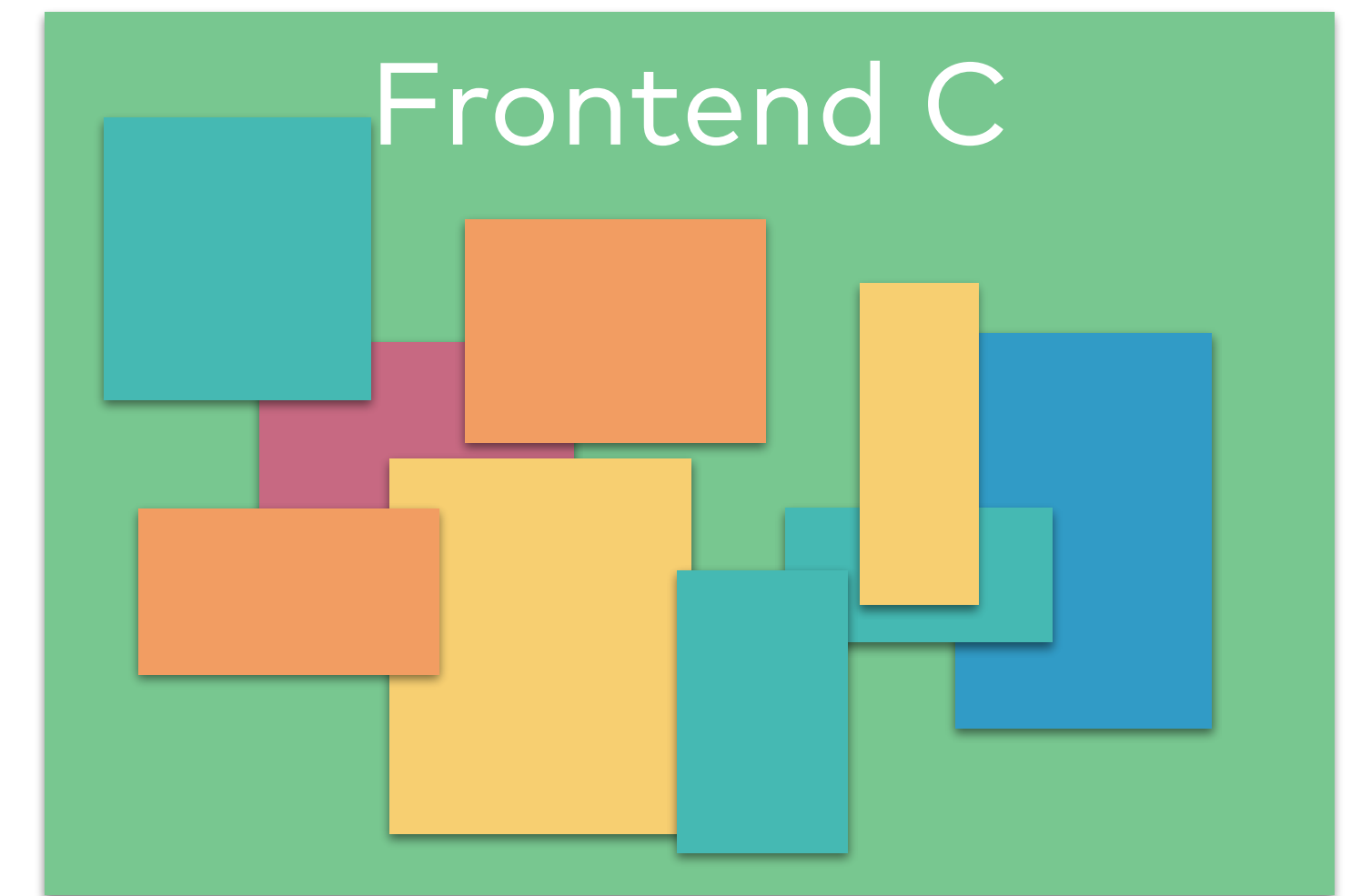
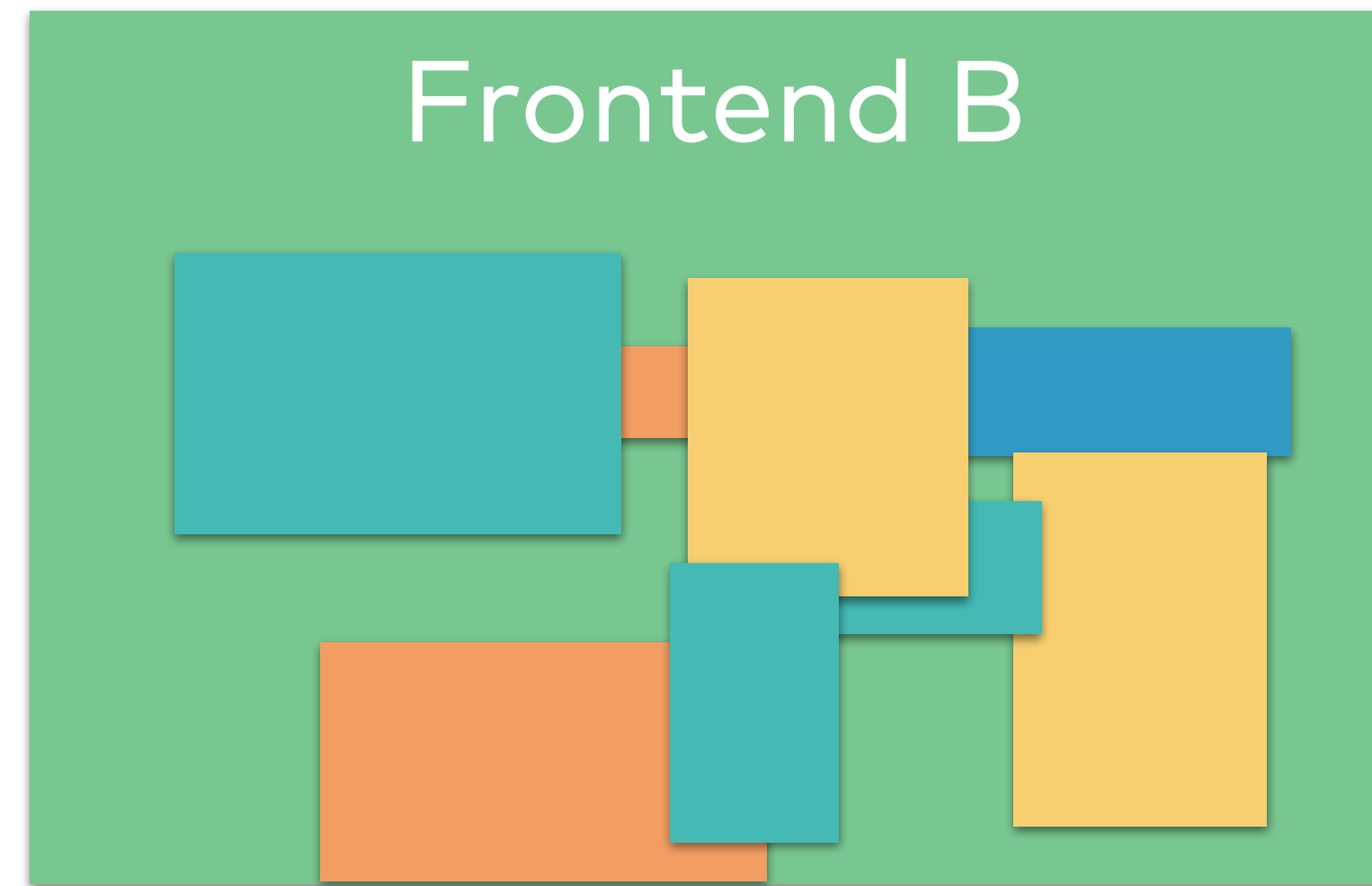
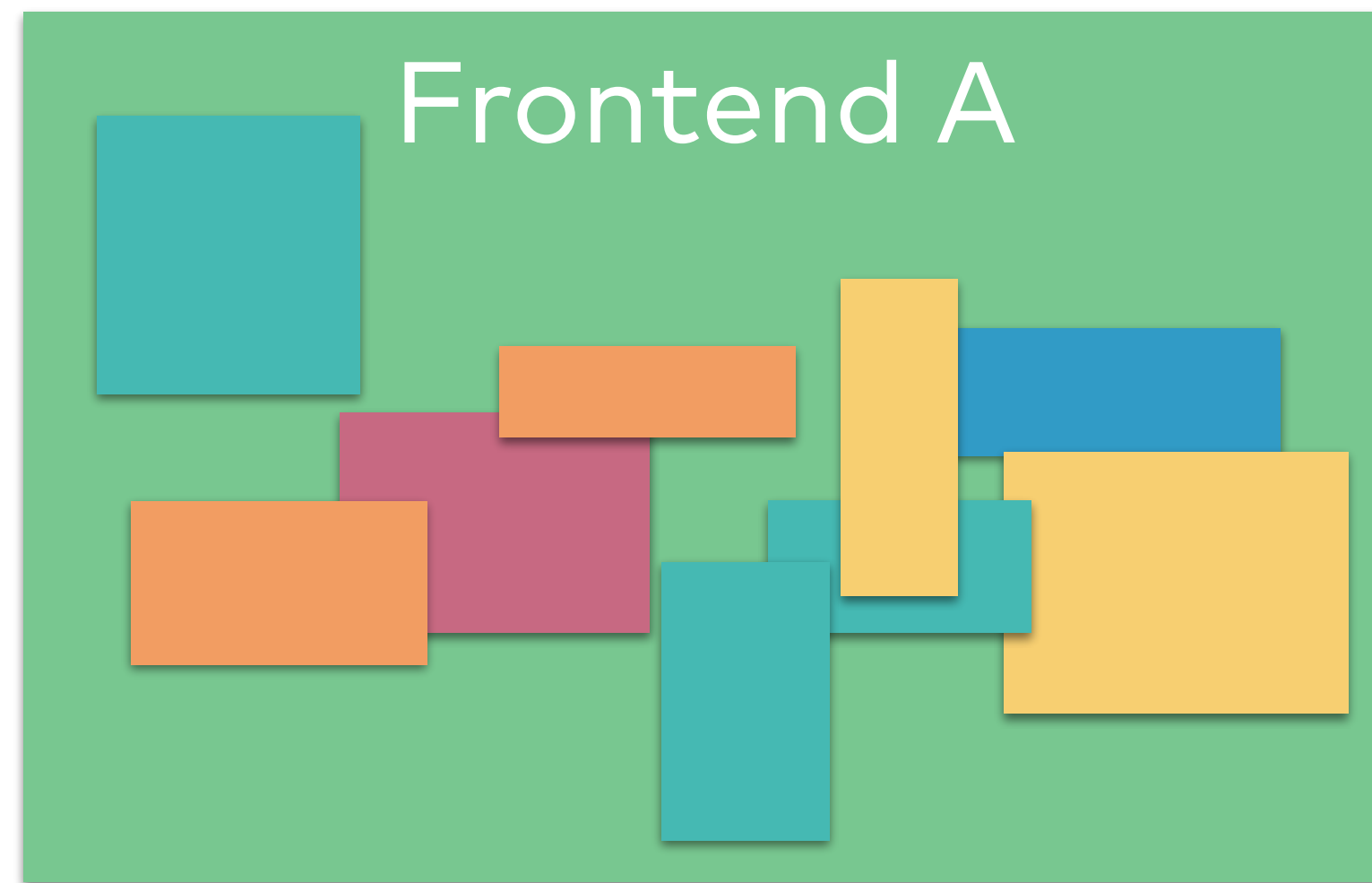
Frontend + services in a backend architect's mind



Frontend + services in the real world



Redundancy with Multiple Frontends



4.

APIs are not valuable until they provide value

3. API design is much harder than you think

2.

Visions of rich API repositories almost always remain visions only

1.

**Unless providing APIs is your business model,
the best API is no API**

Takeaways

APIs are not a silver bullet

**APIs need to be driven by
actual application use-cases**

**APIs are hard – only build
those that are worth it**

**Application before service,
use before re-use,
specific before generalized**

That's all I have. Thanks for listening!

Stefan Tilkov
@stilkov
stefan.tilkov@innoq.com
Phone: +49 170 471 2625



innoQ Deutschland GmbH

Krischerstr. 100
40789 Monheim am Rhein
Germany
Phone: +49 2173 3366-0

Ohlauer Straße 43
10999 Berlin
Germany

Phone: +49 2173 3366-0

Ludwigstr. 180E
63067 Offenbach
Germany

Phone: +49 2173 3366-0

Kreuzstraße 16
80331 München
Germany

Phone: +49 2173 3366-0

innoQ Schweiz GmbH

Gewerbestr. 11
CH-6330 Cham
Switzerland
Phone: +41 41 743 0116

@stilkov