



JCON
2020

ArchUnit

Unit Testing Architecture and Design

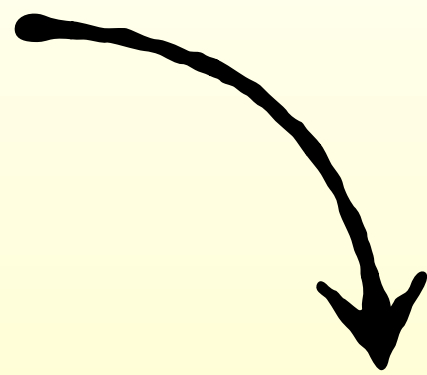
Thomas Much

 @thmuch

#JCON2020

28.10.2020

Author

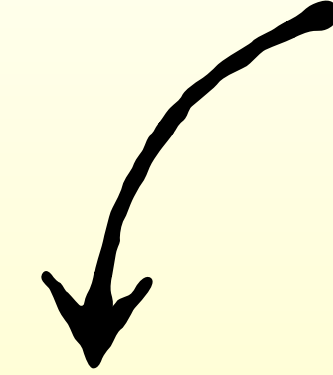


Peter Gafert
@codecholeric
@archtests



Many contributors

Happy user



Thomas Much
@thmuch



What is architecture, anyway?

Decisions, Technologies ...

everything "important", what's hard to change later on

Structure and conventions within one artefact (service)

Communication between systems / services

Shared understanding of the system and its parts

...

Today:
ArchUnit

Unit testing architecture and design –

Why?

Microservices & SCS

Customer

Product

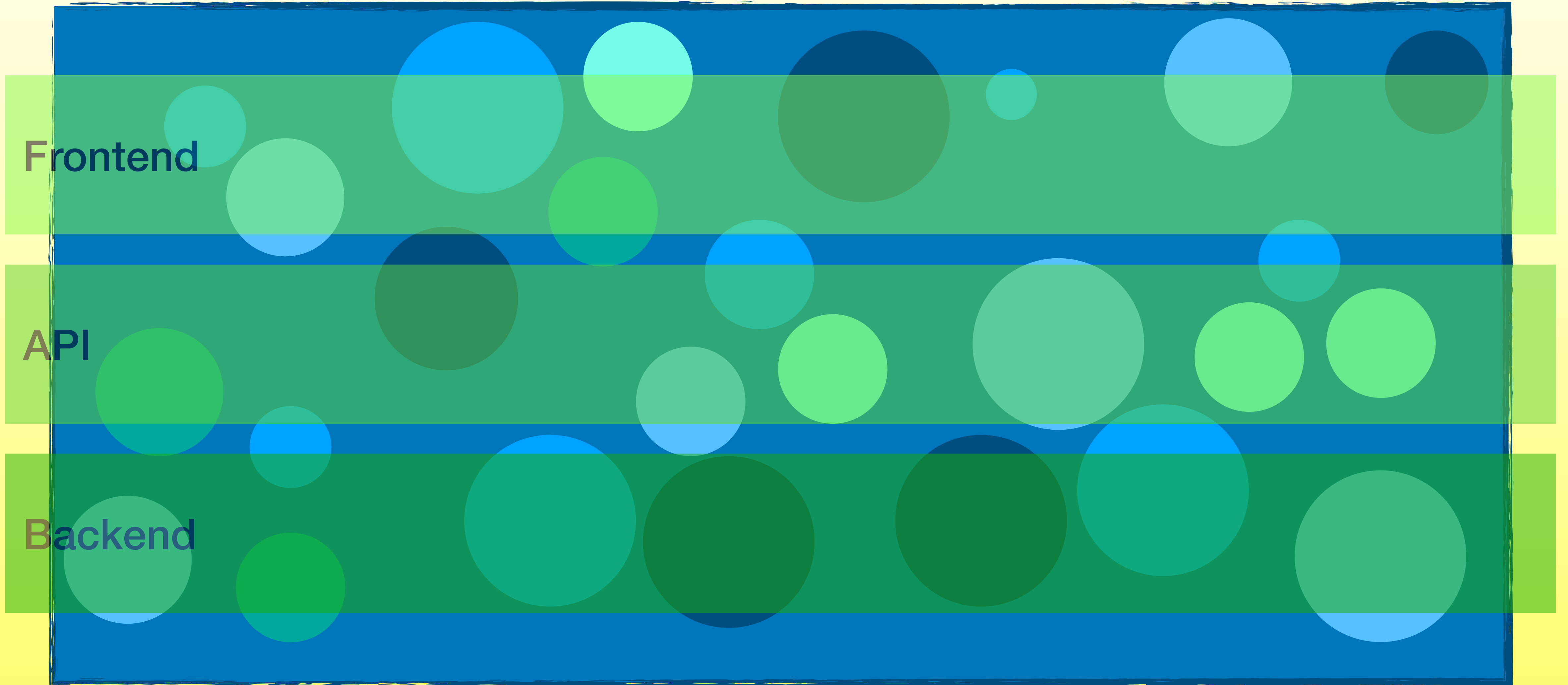
Order

Search

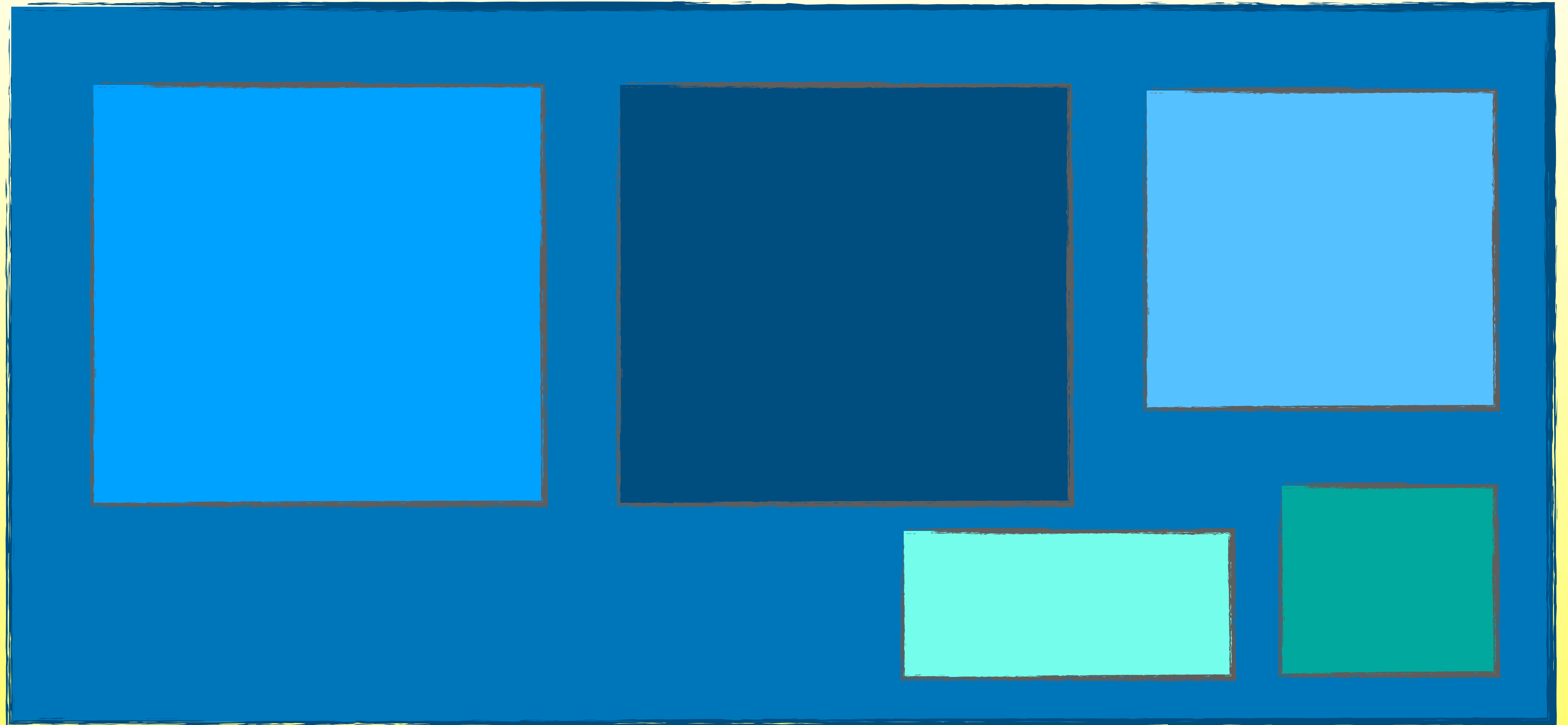
Monoliths



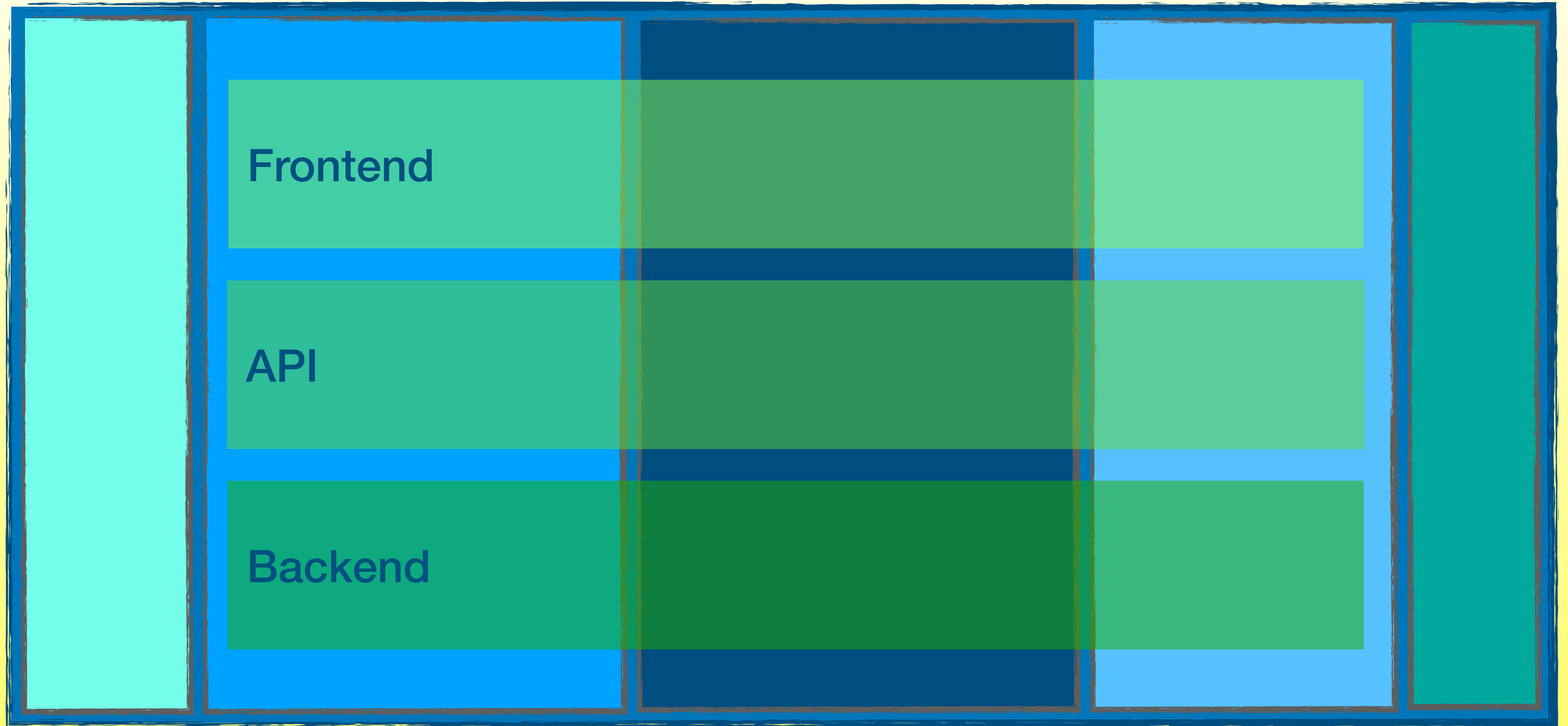
Monoliths



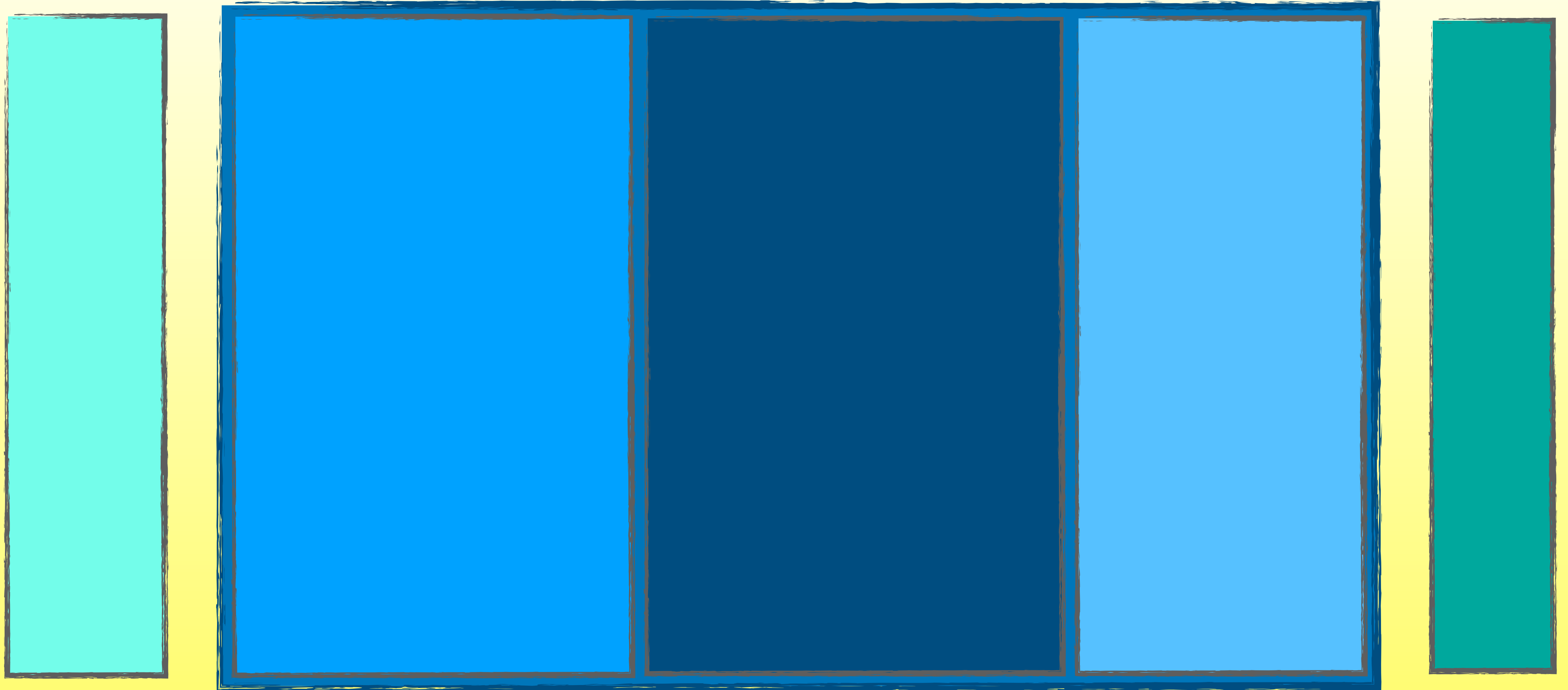
Friendly monoliths



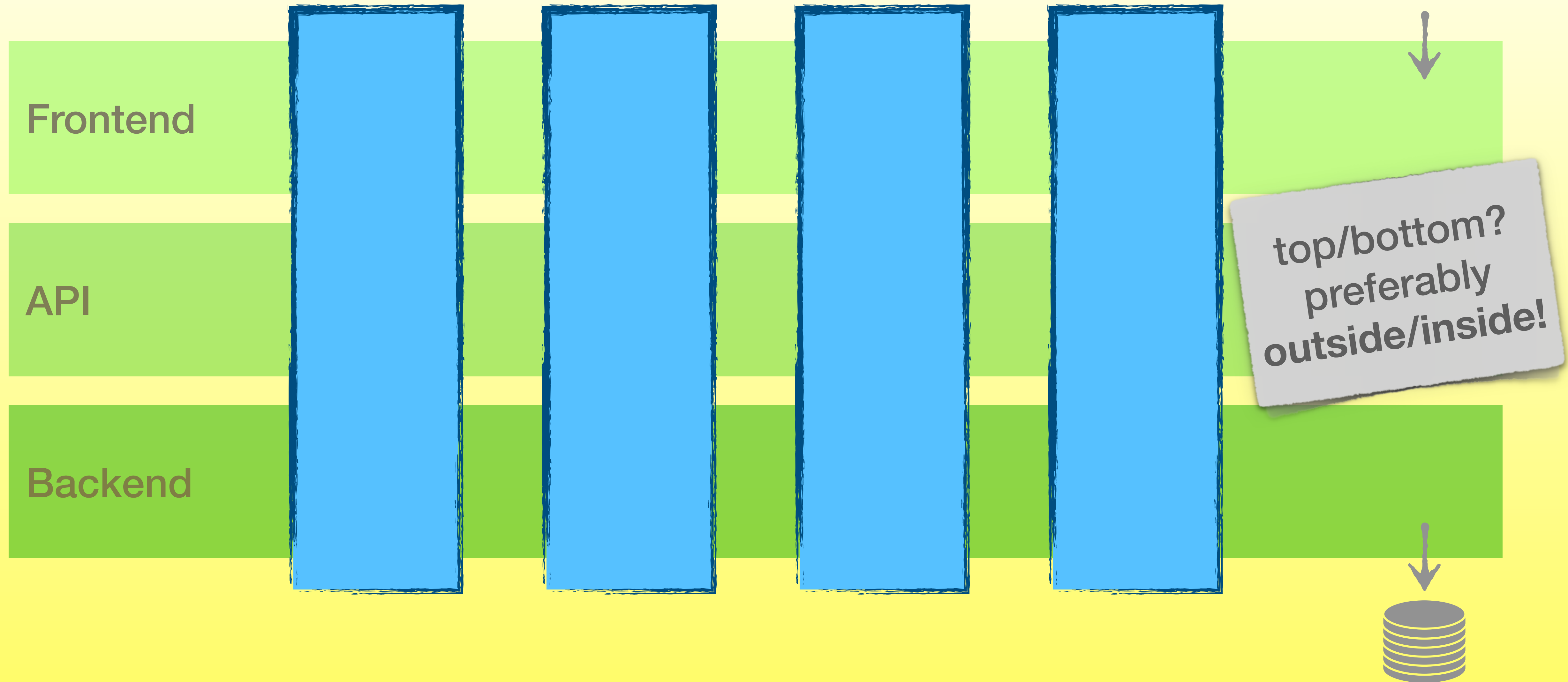
Even more friendly monoliths



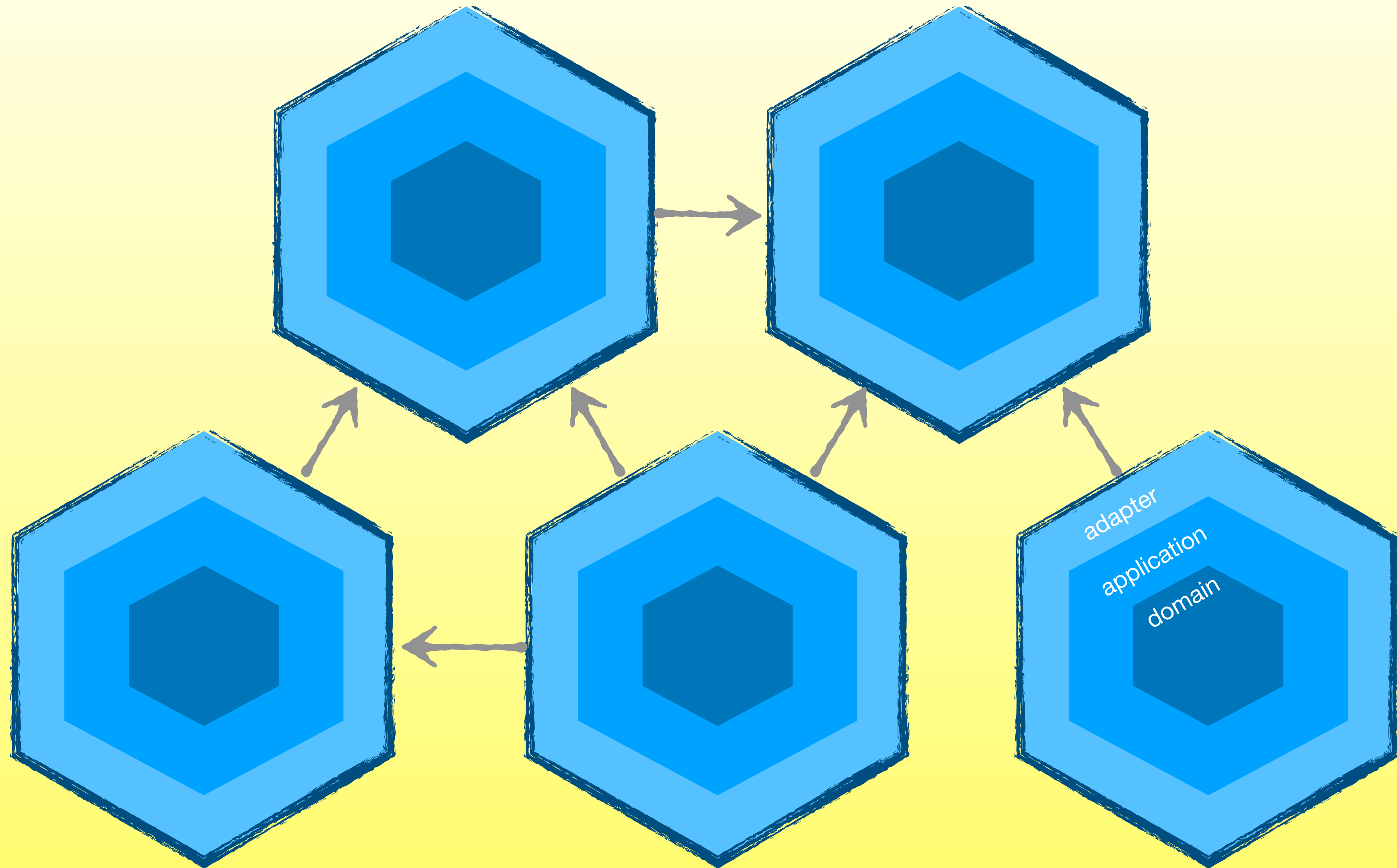
Trimmed monoliths



Microservices & layers?



Microservices / DDD



Testing architecture – what?

Dependencies, cohesion & coupling

Conventions & patterns

Our way to ArchUnit

JDepend

Degrath

jQAssistant

Classycle
Checkstyle

paid ~~license~~

outdated

or

not customisable enough

or

proprietary (configuration) language



End of 2017
"ArchUnit 0.4"



Architecture checks as unit tests

Plain Java code!

Easily customisable – even for design checks

Checking at byte code level

ArchUnit artefacts

Group ID	Artifact ID	Latest Version		Updated
com.tngtech.archunit	archunit-junit	0.8.3	(6)	20-Jul-2018
com.tngtech.archunit	archunit-junit5-engine-api	0.14.1	(13)	24-May-2020
com.tngtech.archunit	archunit-junit5-engine	0.14.1	(13)	24-May-2020
com.tngtech.archunit	archunit-junit5-api	0.14.1	(13)	24-May-2020
com.tngtech.archunit	archunit-junit5	0.14.1	(4)	24-May-2020
com.tngtech.archunit	archunit-junit4	0.14.1	(13)	24-May-2020
com.tngtech.archunit	archunit	0.14.1	(19)	24-May-2020

```
<dependency>  
  <groupId>com.tngtech.archunit</groupId>  
  <artifactId>archunit</artifactId>  
  <version>0.14.1</version>  
  <scope>test</scope>  
</dependency>
```

Using ArchUnit

Possible with every unit test framework

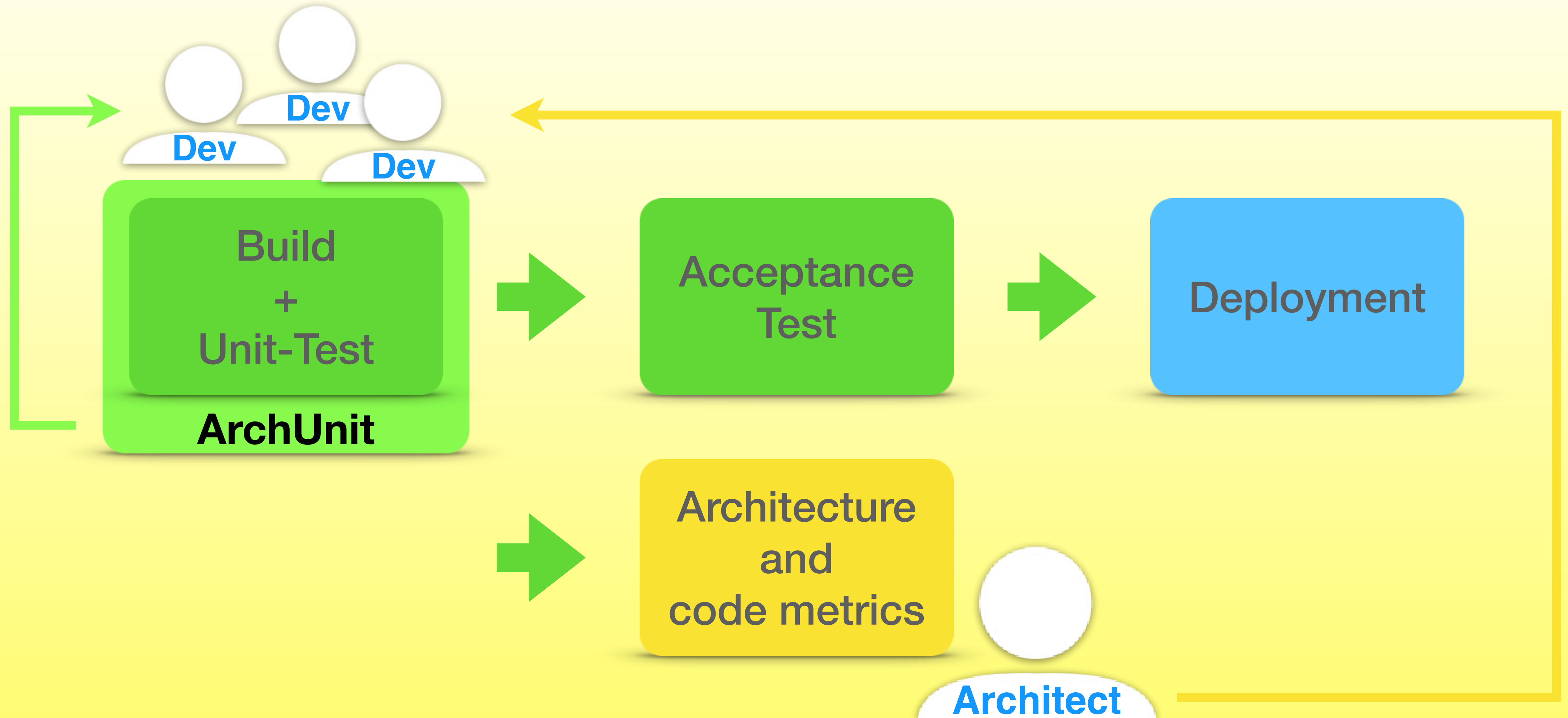
Built-in support for JUnit 4 / 5

Java & Kotlin

Maven & Gradle plugins

Live demo

Why ArchUnit?



Alternatives & complements

The logo for jQAAssistant features the text "jQAAssistant" in a dark grey font. A green checkmark is superimposed over the "j" and "Q" characters.

The logo for structure101 consists of the text "structure101" in a bold, dark blue, sans-serif font.

The logo for sonarqube features the text "sonarqube" in a dark grey font. To the right of the text are three curved blue lines of varying lengths, resembling a sonar wave.

The logo for Deptective features a yellow detective emoji wearing a brown hat and holding a magnifying glass, followed by the text "Deptective" in a dark grey font.

The logo for SONARGRAPH features the text "SONARGRAPH" in a dark grey font. The "A" is replaced by a red triangle pointing upwards. The text is underlined with a thin black line.

The logo for embold features a blue icon consisting of several horizontal lines of varying lengths, followed by the text "embold" in a bold, dark blue, sans-serif font.

... and more ...

Moduliths w/ Spring Boot

Moduliths

A playground to build technology supporting the development of

build passing

tl;dr

Moduliths is a Spring Boot extension based on ArchUnit to achieve

- *Verify modular structure between individual logical modules*

Prevents cyclic dependencies as well as explicitly defined all to public components in API packages (convention based, c)

- *Bootstrap a subset of the modules of a monolithic Spring P*

<https://github.com/odrotbohm/moduliths>

https://www.archunit.org

[Getting Started](#)[Motivation](#)[News](#)[User Guide](#)[API](#)[About](#)

Persistence

Unit test your Java architecture

Start enforcing your architecture within 30 minutes using the test setup you already have.

[Start Now](#)

ArchUnit is a free, simple and extensible library for checking the architecture of your Java code using any plain Java unit test framework. That is, ArchUnit can check dependencies between packages and classes, layers and slices, check for cyclic dependencies and more. It does so by analyzing given Java bytecode, importing all classes into a Java code structure. You can find examples for the current release at [ArchUnit Examples](#) and the sources on [GitHub](#).



<https://github.com/TNG/ArchUnit-Examples>

<https://github.com/thmuch/archunit-demos>

<https://github.com/TNG/ArchUnit>

<https://github.com/TNG/ArchUnitNET>



Layers

Architecture

Slices

Design

Verticals

Conventions

Questions?

Monoliths

Modules

SCS

Dependencies

Microservices

Coupling

Cohesion



JCON
2020

Thank you!



Thomas Much

 @thmuch

#JCON2020