

**JAVAUSEARGROUP**HANNOVER

# ArchUnit

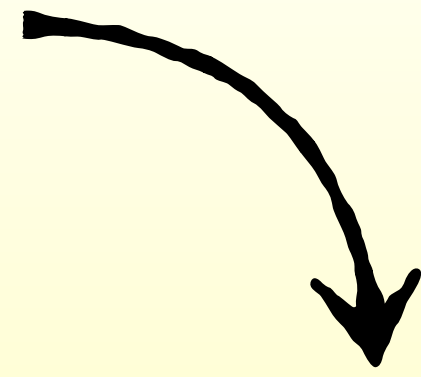
Testen von Architektur und Design

Thomas Much

 @thmuch

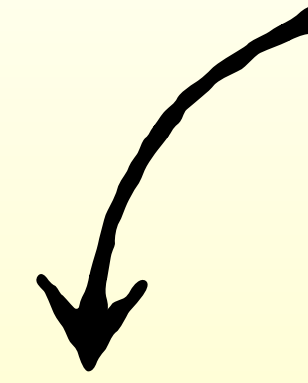
14.03.2019

**Autor**



Peter Gafert  
@codecholeric  
@archtests

**glücklicher Anwender**



Thomas  
@thmuch



Architektur und Design automatisiert prüfen –

**Warum?**

# Microservices & SCS



Customer

Product

Order

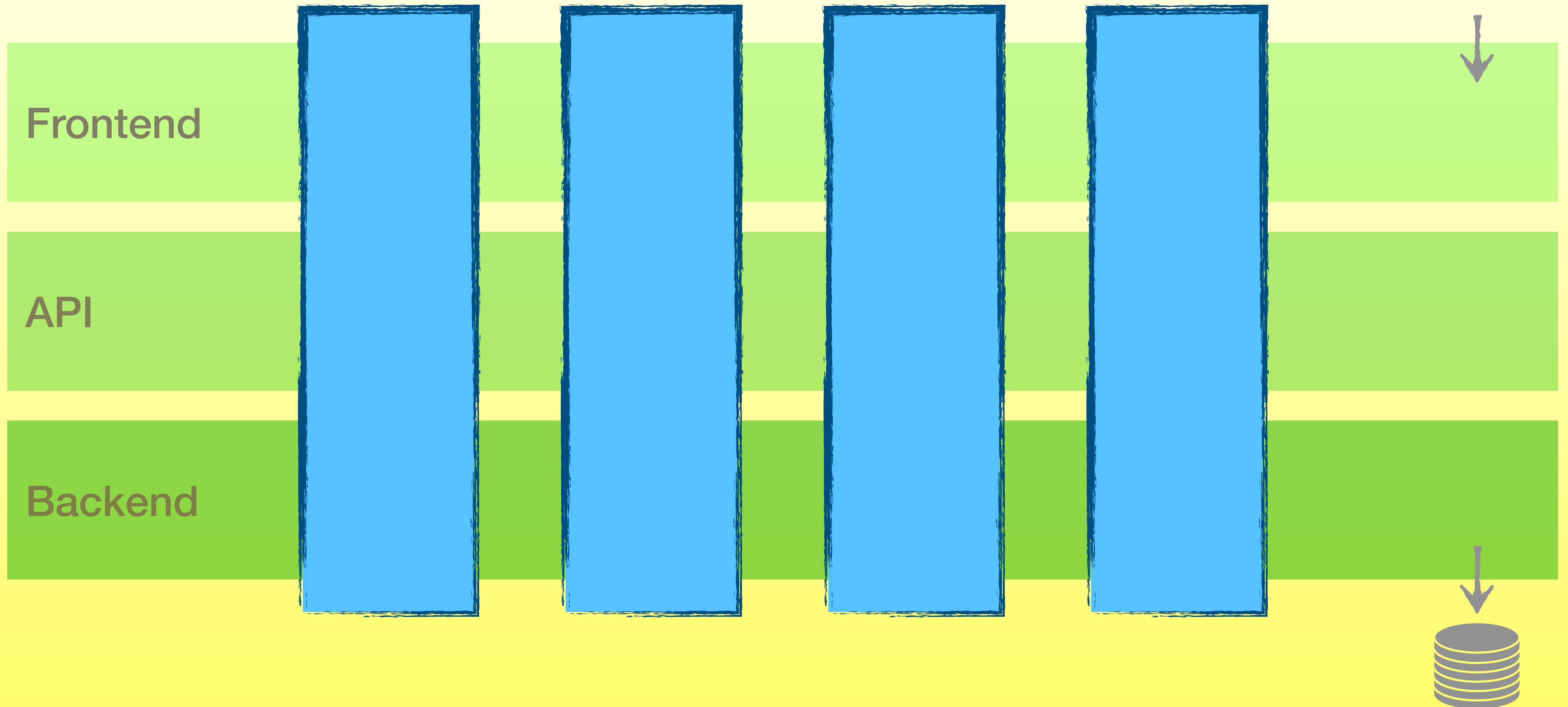
Search



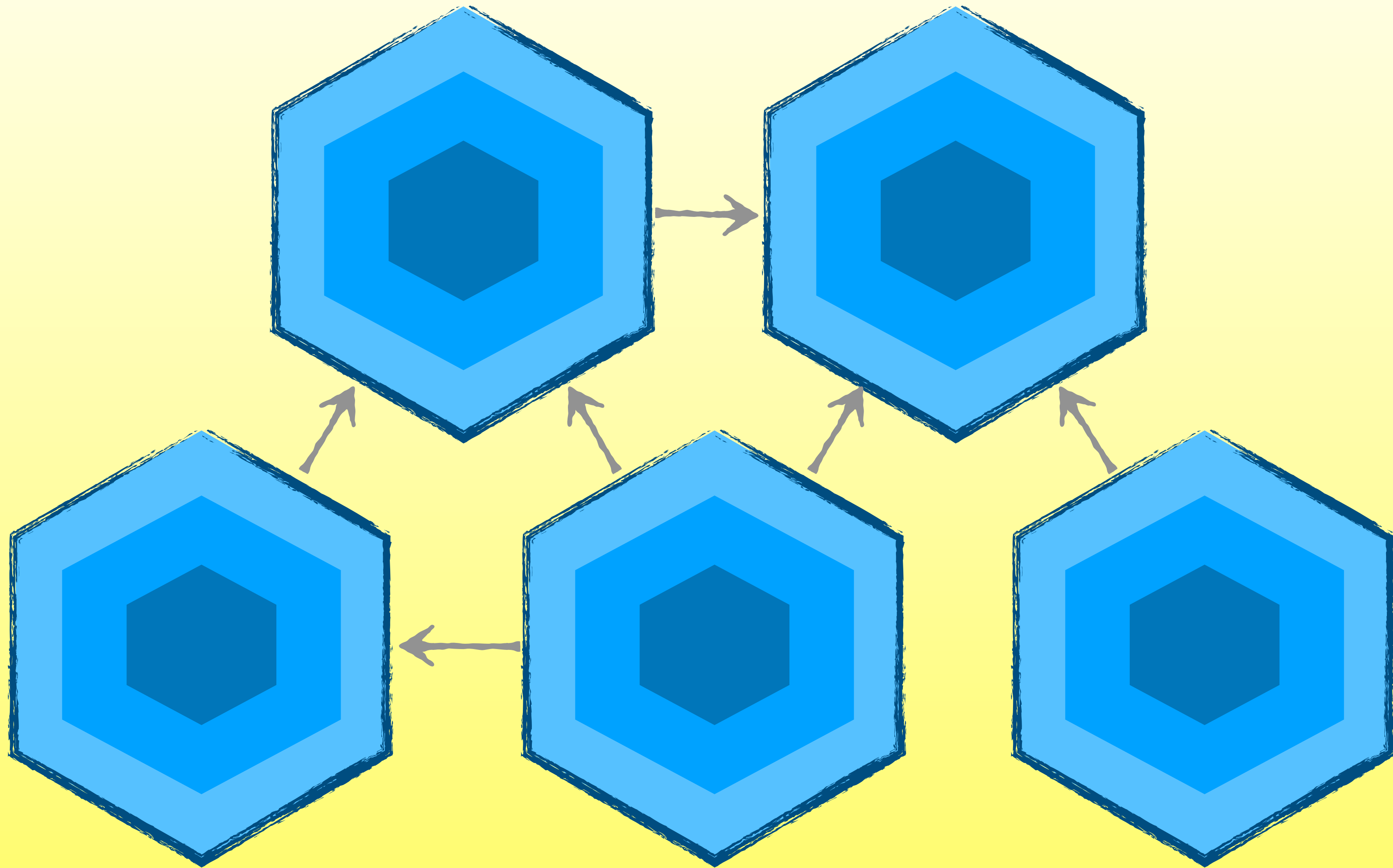
# Monolithen



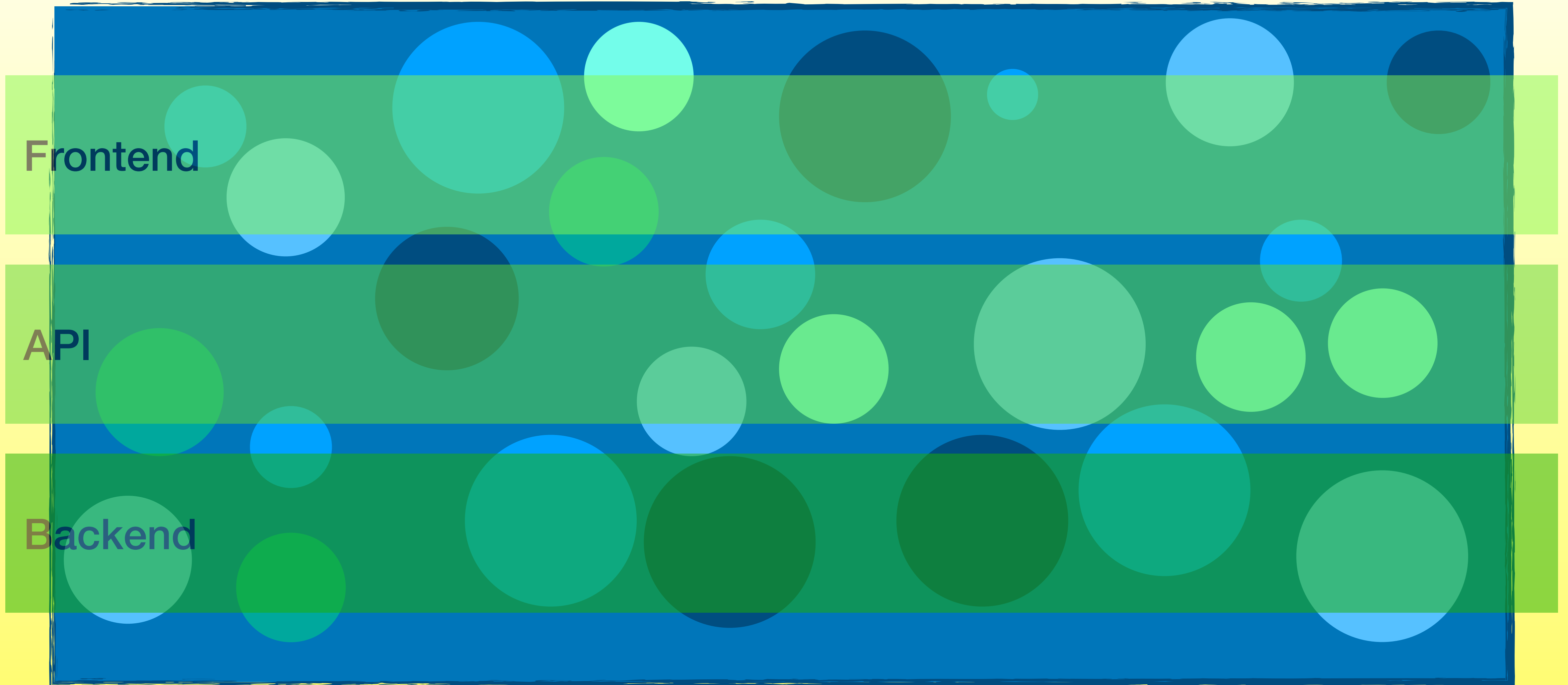
# Microservices / Schichten?



# Microservices / DDD

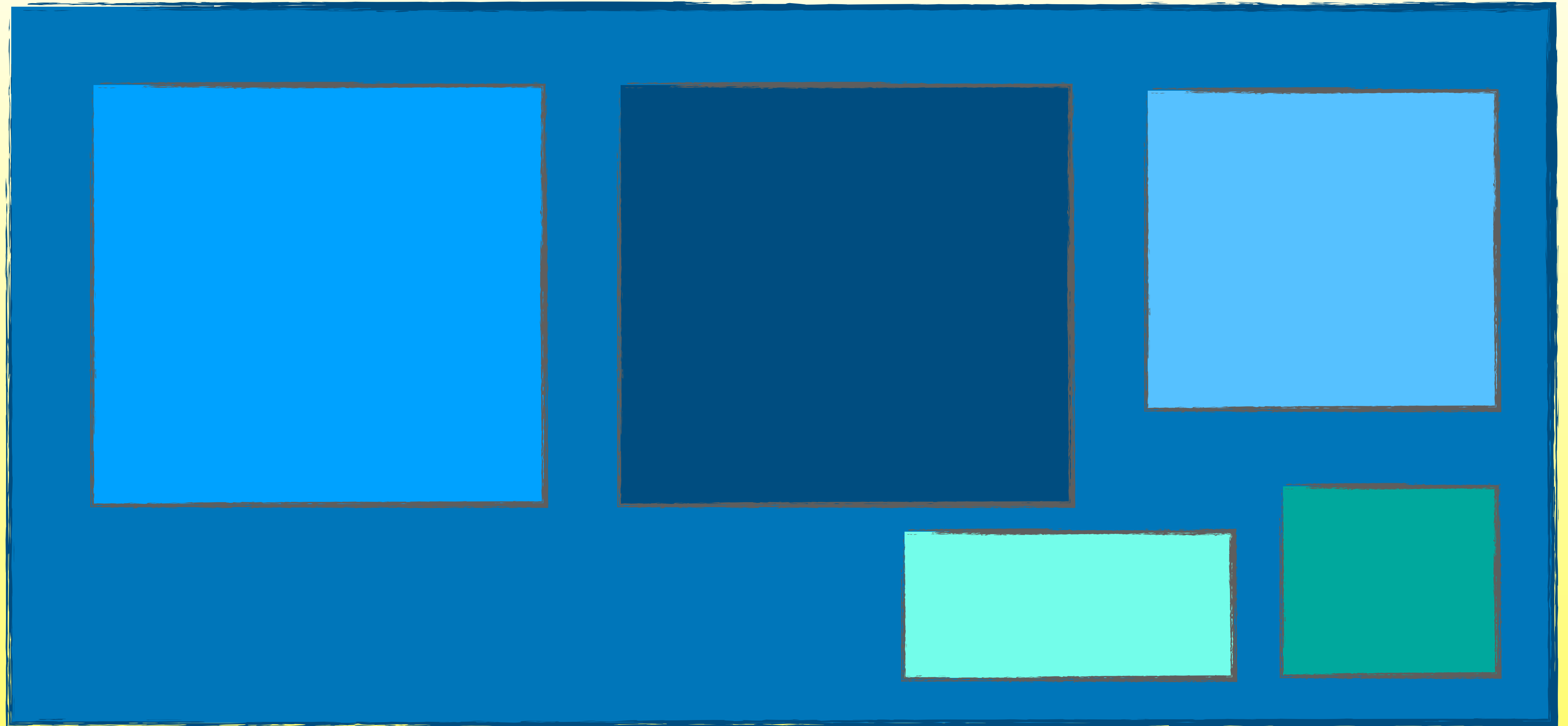


# Monolithen

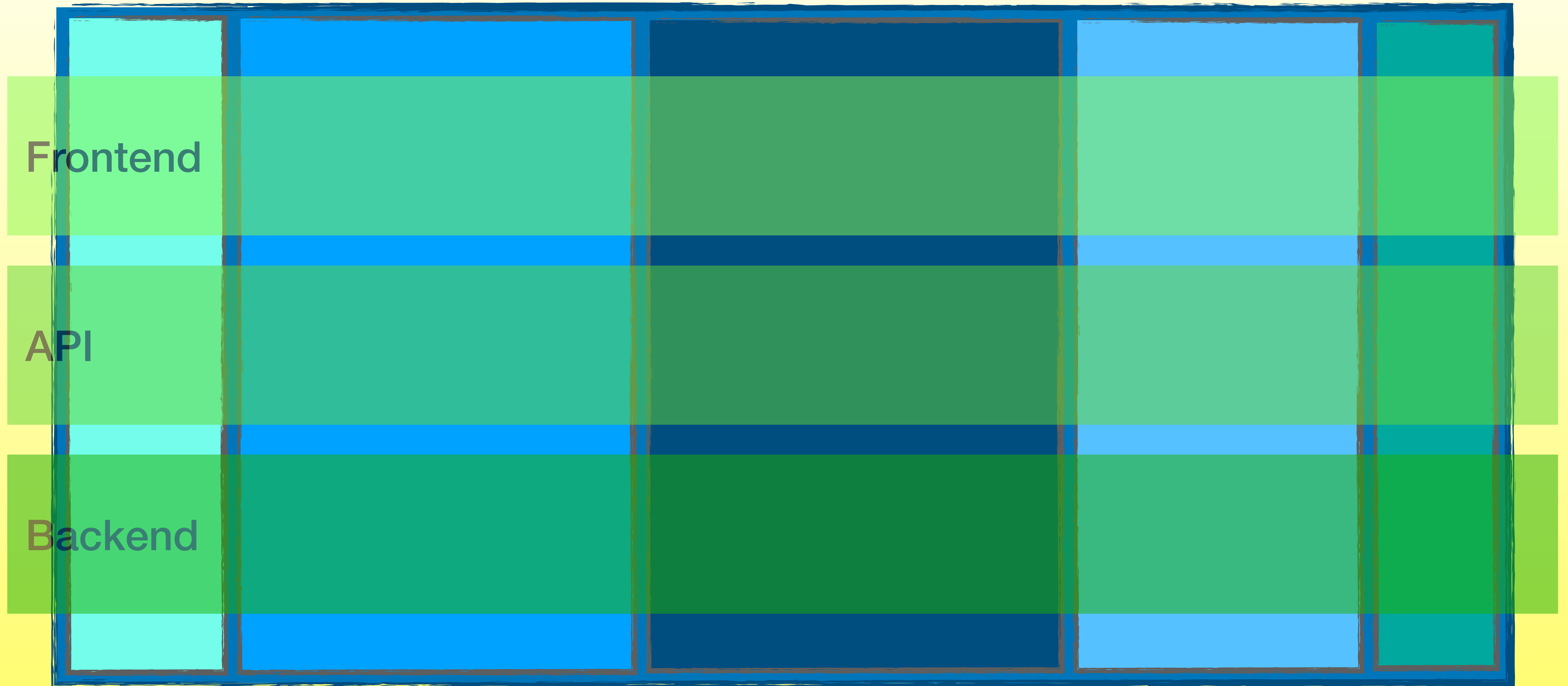




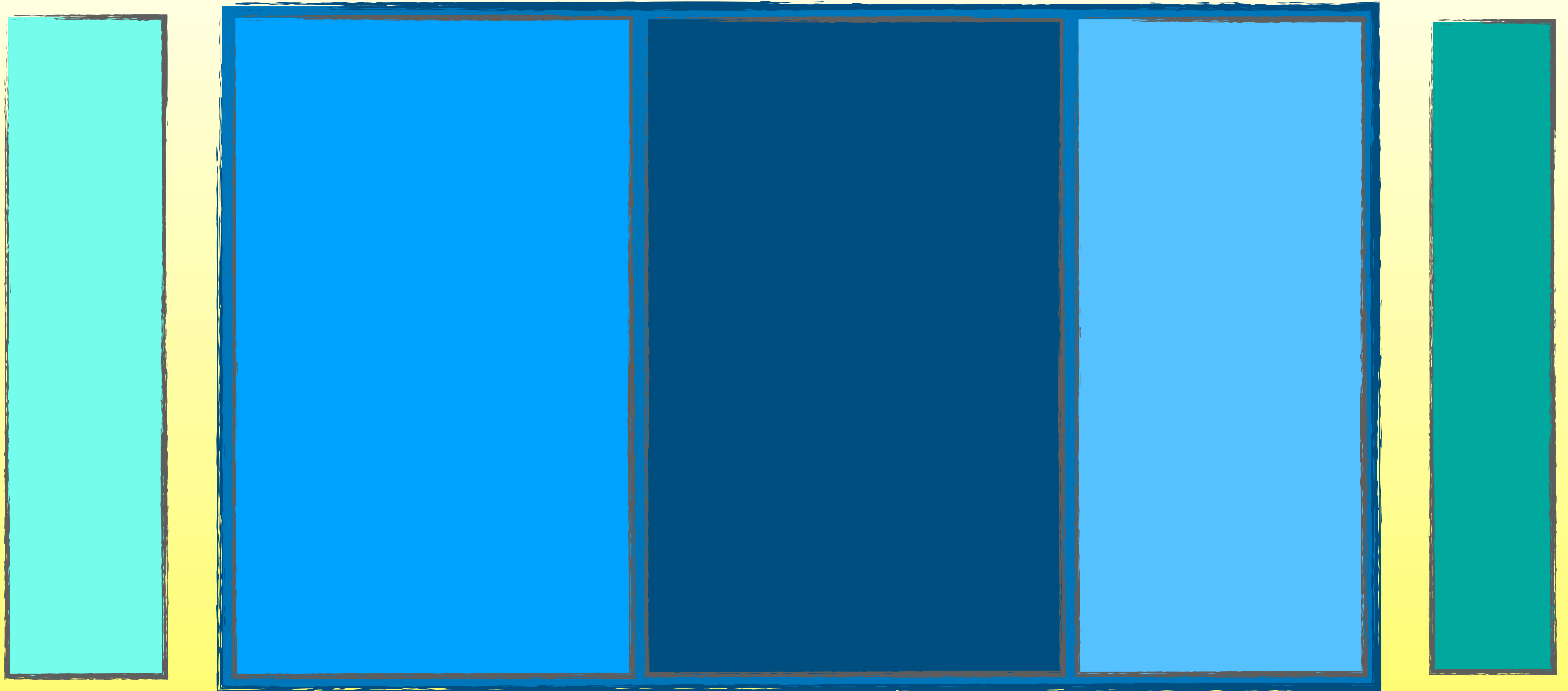
# Freundliche Monolithen



# Noch freundlichere Monolithen



# Abgespeckte Monolithen



# Architektur testen – warum?

Wartbarkeit

Ersetzbarkeit

Code-Conventions

Verständlichkeit

Dokumentation

Kommunikation

Wir  
sind  
ein  
Team!

# Architektur testen

Abhängigkeiten, Kohäsion & Kopplung

Konventionen & Patterns



# Unser Weg zu ArchUnit

JDepend

Degrath

jQAssistant

**Classycle**  
Checkstyle

~~kommerziell~~

veraltet  
*oder*  
nicht flexibel genug  
*oder*  
eigene (Konfigurations-)Sprache



**Ende 2017**  
**„ArchUnit 0.4“**



Architektur-Prüfungen als Unit-Tests

Normaler Java-Code!

Flexibel erweiterbar – auch Design-Prüfungen realisierbar

Prüfung auf Bytecode-Ebene

# ArchUnit einbinden

Group ID	Artifact ID	Latest Version		Updated
<a href="#">com.tngtech.archunit</a>	<a href="#">archunit</a>	<a href="#">0.9.3</a>	<a href="#">(10)</a>	20-Nov-2018
<a href="#">com.tngtech.archunit</a>	<a href="#">archunit-junit5-engine-api</a>	<a href="#">0.9.3</a>	<a href="#">(4)</a>	20-Nov-2018
<a href="#">com.tngtech.archunit</a>	<a href="#">archunit-junit5-engine</a>	<a href="#">0.9.3</a>	<a href="#">(4)</a>	20-Nov-2018
<a href="#">com.tngtech.archunit</a>	<a href="#">archunit-junit5-api</a>	<a href="#">0.9.3</a>	<a href="#">(4)</a>	20-Nov-2018
<a href="#">com.tngtech.archunit</a>	<a href="#">archunit-junit4</a>	<a href="#">0.9.3</a>	<a href="#">(4)</a>	20-Nov-2018
<a href="#">com.tngtech.archunit</a>	<a href="#">archunit-junit</a>	<a href="#">0.8.3</a>	<a href="#">(6)</a>	20-Jul-2018


```
<dependency>
  <groupId>com.tngtech.archunit</groupId>
  <artifactId>archunit</artifactId>
  <version>0.9.3</version>
  <scope>test</scope>
</dependency>
```

# ArchUnit ausführen

Mit jedem Unit-Test-Framework nutzbar

Spezielle Unterstützung für JUnit 4 / 5

Java & Kotlin

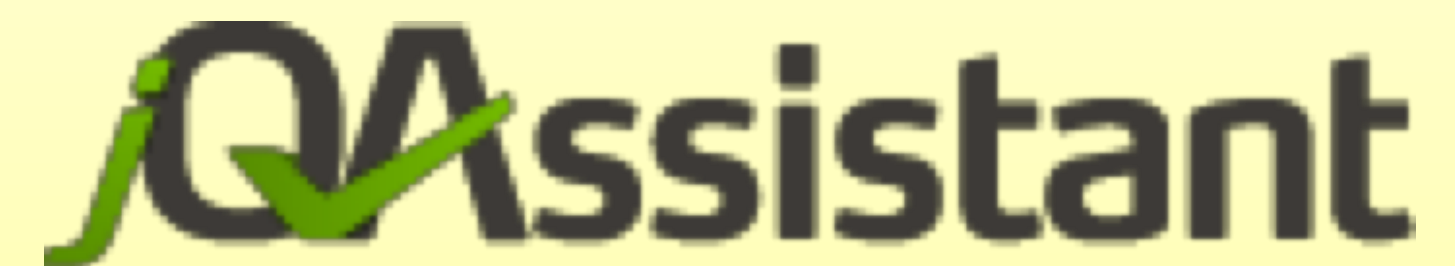
Maven-Plugin 

Live-Demo

Live-Demo



# Alternativen

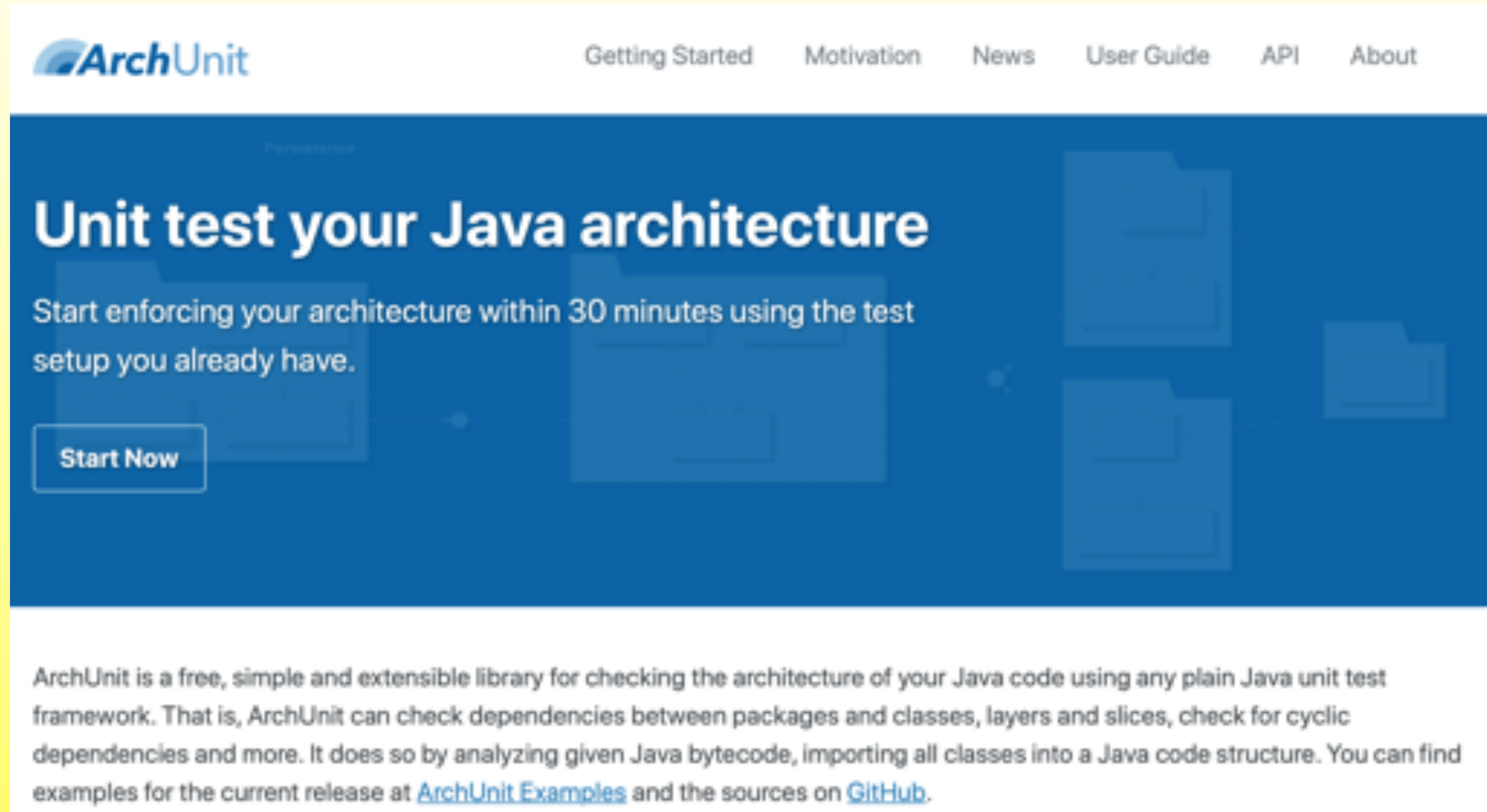
The logo for jQAAssistant features a stylized 'j' in green with a checkmark, followed by 'QAAssistant' in a dark grey sans-serif font.

The logo for structure101 is written in a bold, blue, sans-serif font.

The logo for Deptective features a detective emoji (a person in a grey suit and hat with a magnifying glass) to the left of the word 'Deptective' in a black sans-serif font. To the right of the word is a blue rounded square badge with the word 'NEW' in white. Below the logo is the URL <https://github.com/moditect/deptective> in a smaller black font.

u.a.

# <https://www.archunit.org>



The screenshot shows the ArchUnit website homepage. At the top left is the ArchUnit logo. To its right is a navigation menu with links for 'Getting Started', 'Motivation', 'News', 'User Guide', 'API', and 'About'. The main content area has a dark blue background with the heading 'Unit test your Java architecture' and a sub-heading 'Start enforcing your architecture within 30 minutes using the test setup you already have.' Below this is a 'Start Now' button. At the bottom, there is a paragraph of text describing ArchUnit as a free, simple, and extensible library for checking Java code architecture, with links to 'ArchUnit Examples' and 'GitHub'.

**ArchUnit**

Getting Started Motivation News User Guide API About

## 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](#).

# ArchUnit

Schichten

Architektur

Slices

Design

Vertikalen

Fragen?

Implementierung

Module

Monolithen  
SCS

Abhängigkeiten

Microservices

Kopplung

Kohäsion

 ArchUnit

**Danke!**



Thomas Much

 @thmuch