



Einfach gut testbar

Grundlagen von Codedesign und Architektur für gute Testbarkeit

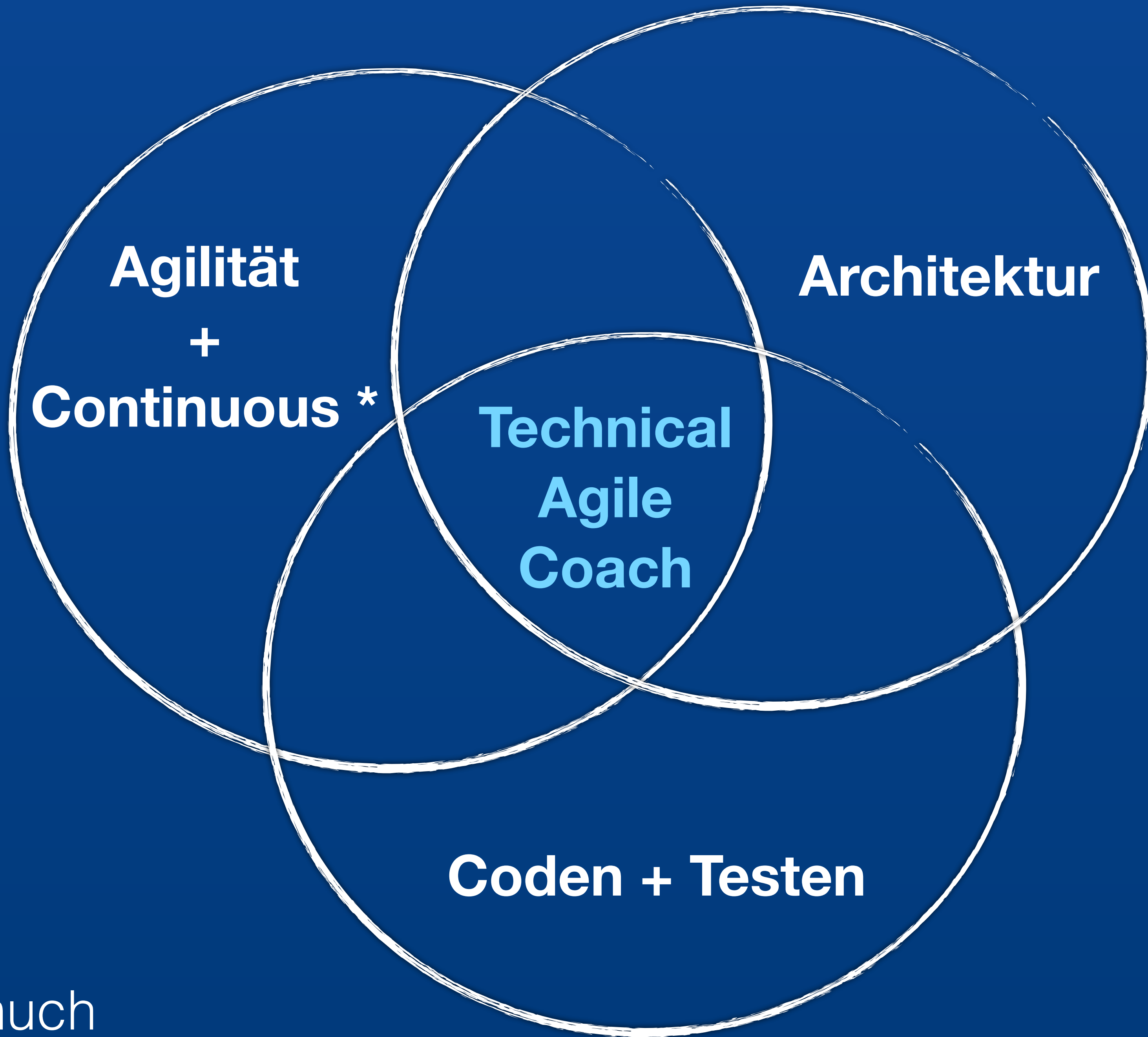
Thomas Much

 @thmuch

9. November 2022



www.tk.de/IT



 @thmuch

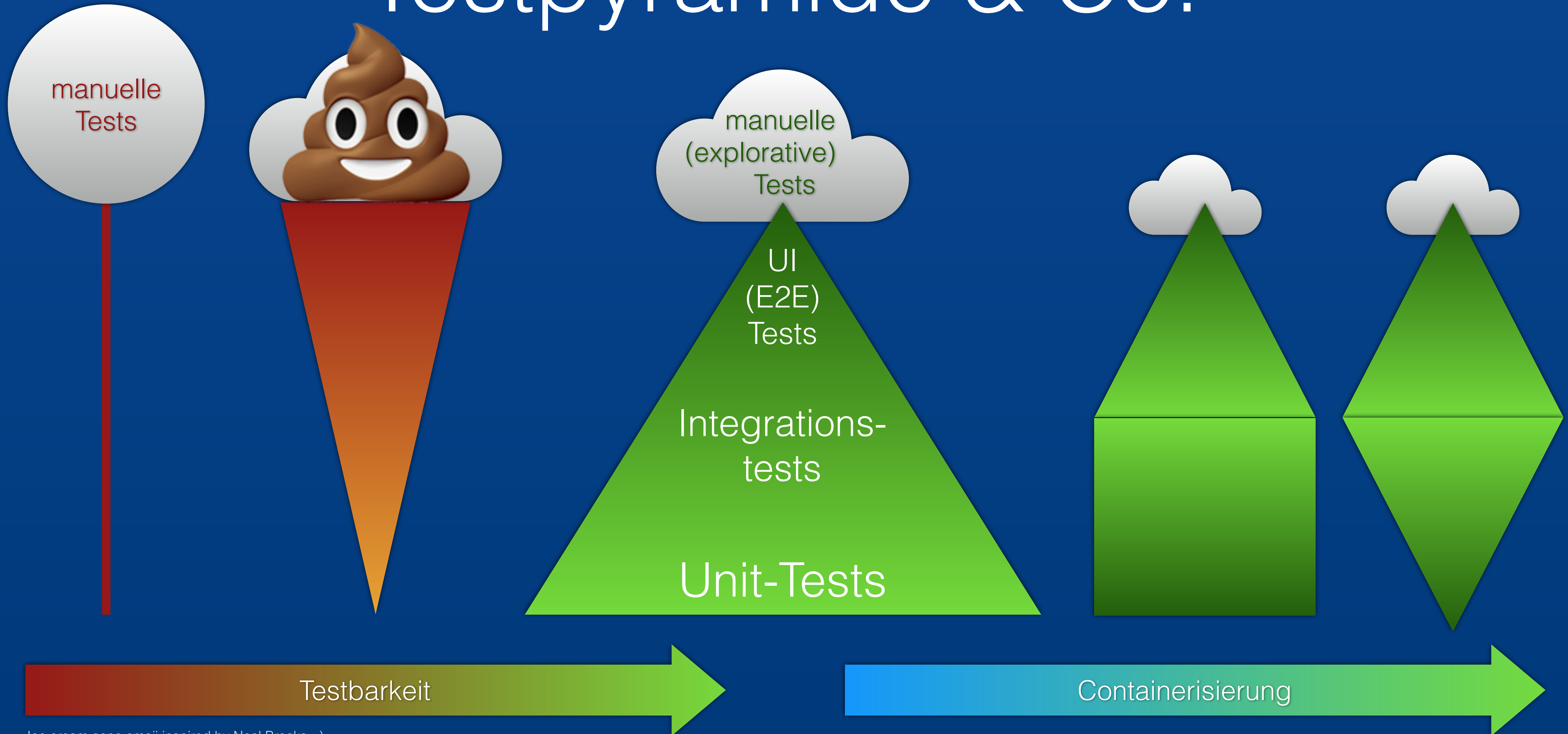
Testbarkeit

Automatisierung **praktikabel**

Testdaten **unter Kontrolle**

Schnelles Feedback

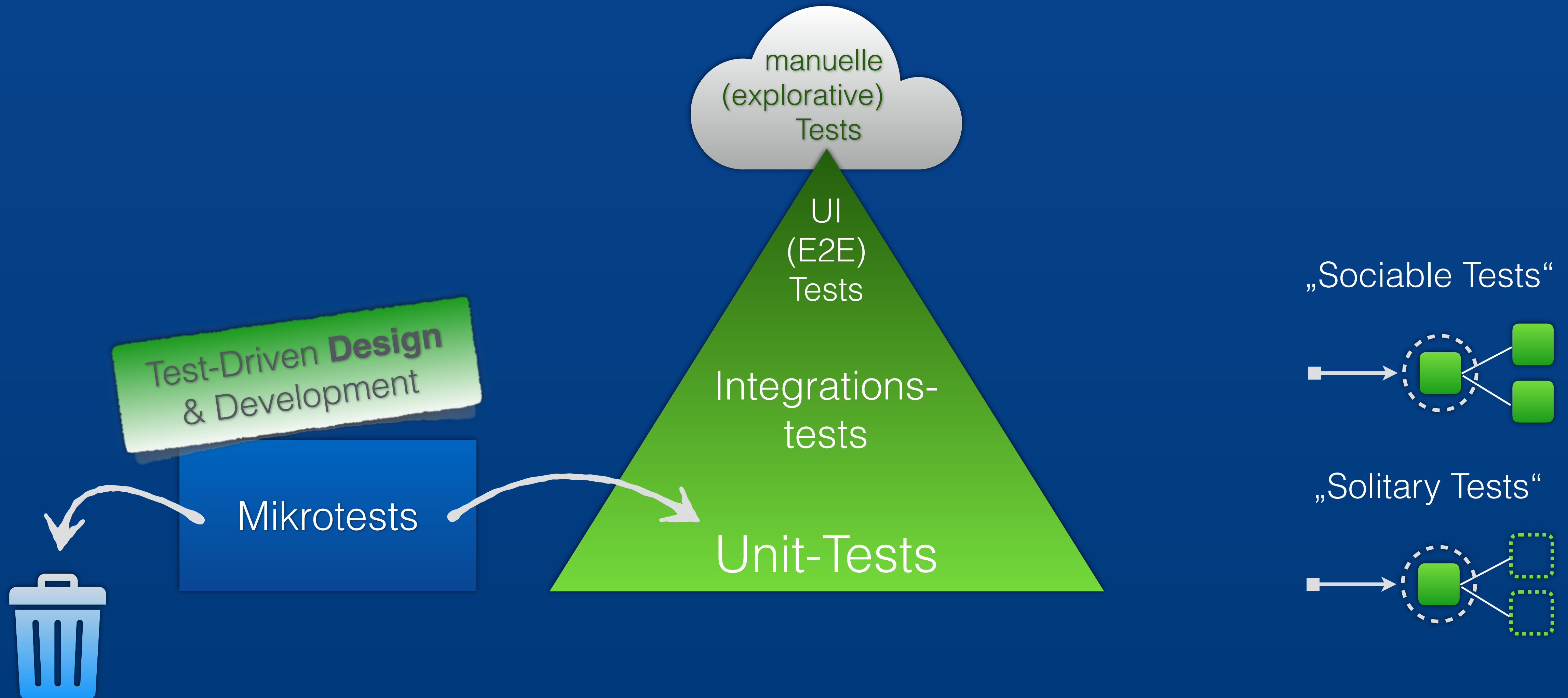
Testpyramide & Co.



Schnelle Tests

Viele
umfassende
schnelle
stabile
Tests

Units oft größer als Mikrotests!

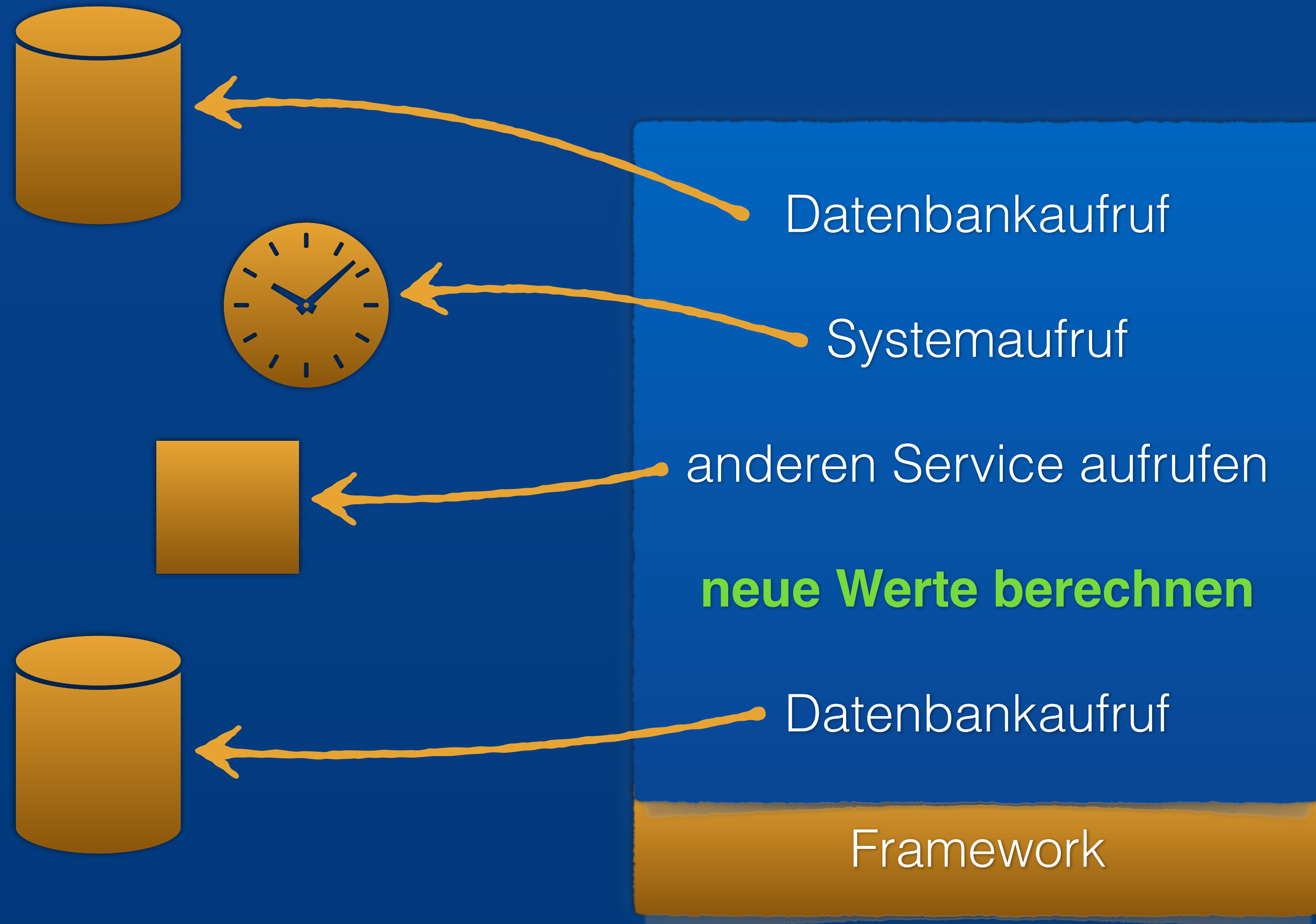


Schnelle Tests

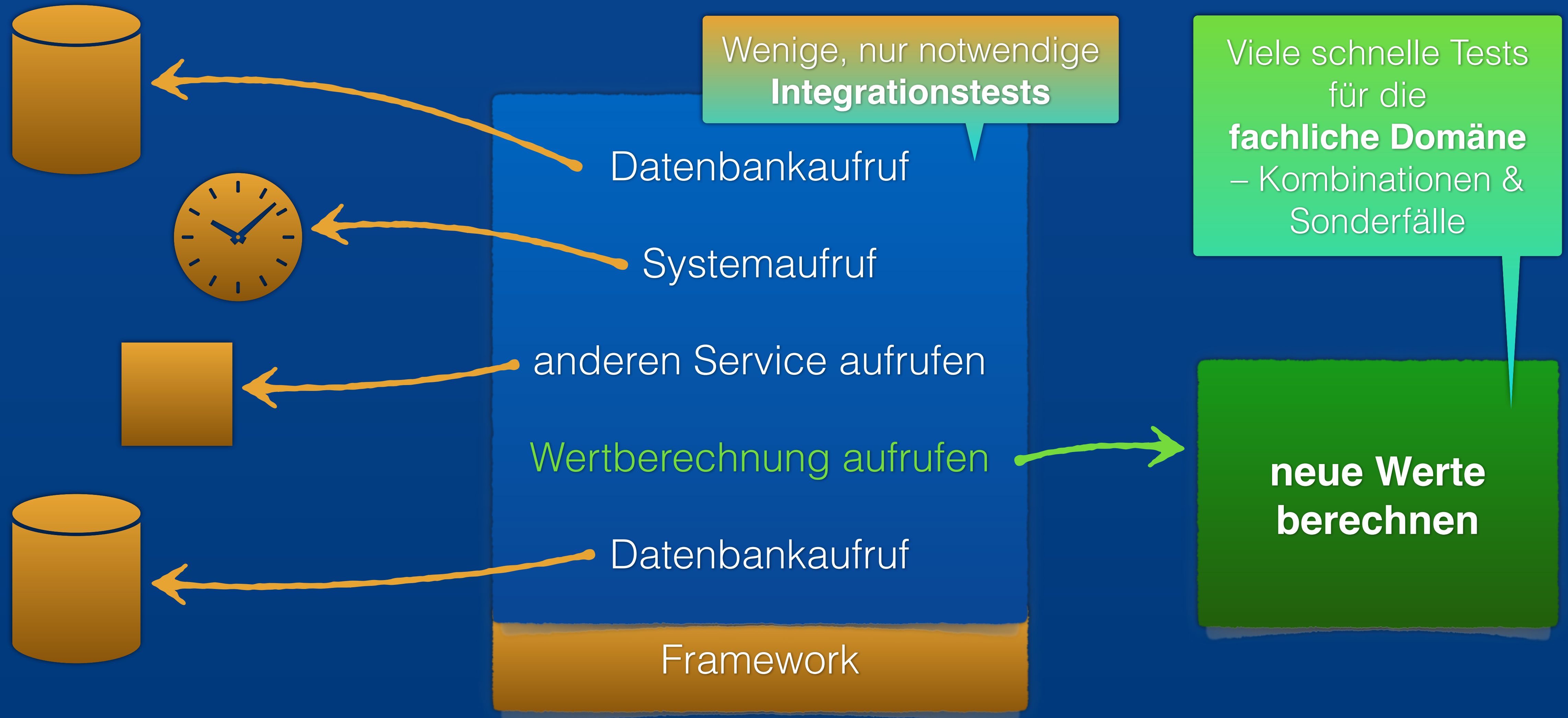
Code und **Architektur** müssen
für **Testbarkeit** designed sein

Abhängigkeiten
machen das schwierig

Abhängigkeiten



Abhängigkeiten



Trennung von
Integrationscode
und
Domänencode

Teil 1

Live Demo

Muster (Patterns) & Stile

Code-Design-Patterns

"Integration Operation Segregation Principle" (IOSP)

"Single Layer of Abstraction" (SLA)

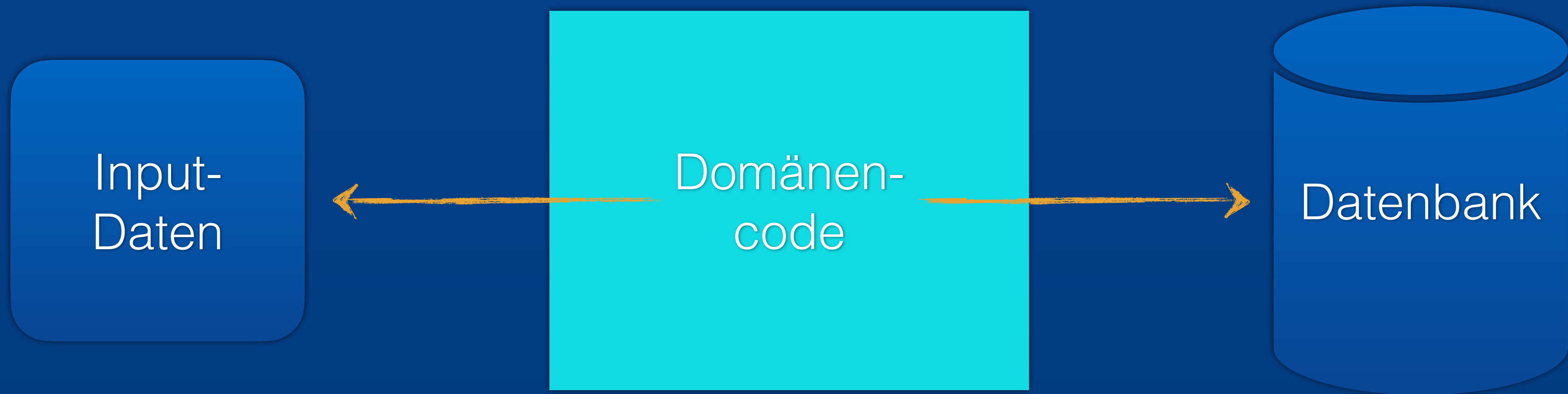
etc.

Vergleichbare **Muster & Stile** für die **Architektur!**

Architekturmuster



Architekturmuster



Architekturmuster



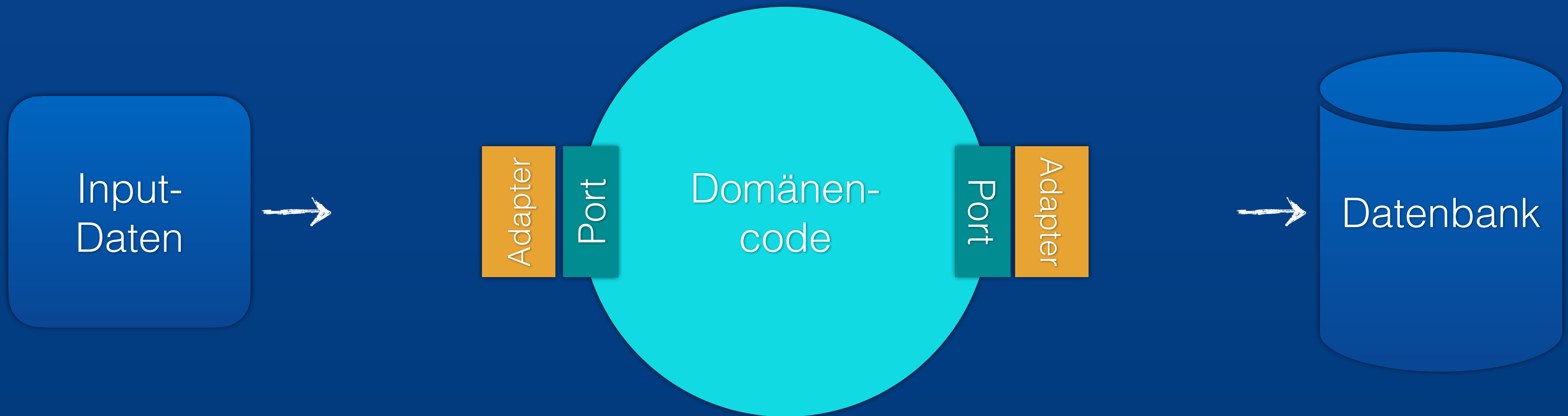
Schnelle (Unit-)Tests für ganze Anwendungsfälle



Architekturmuster

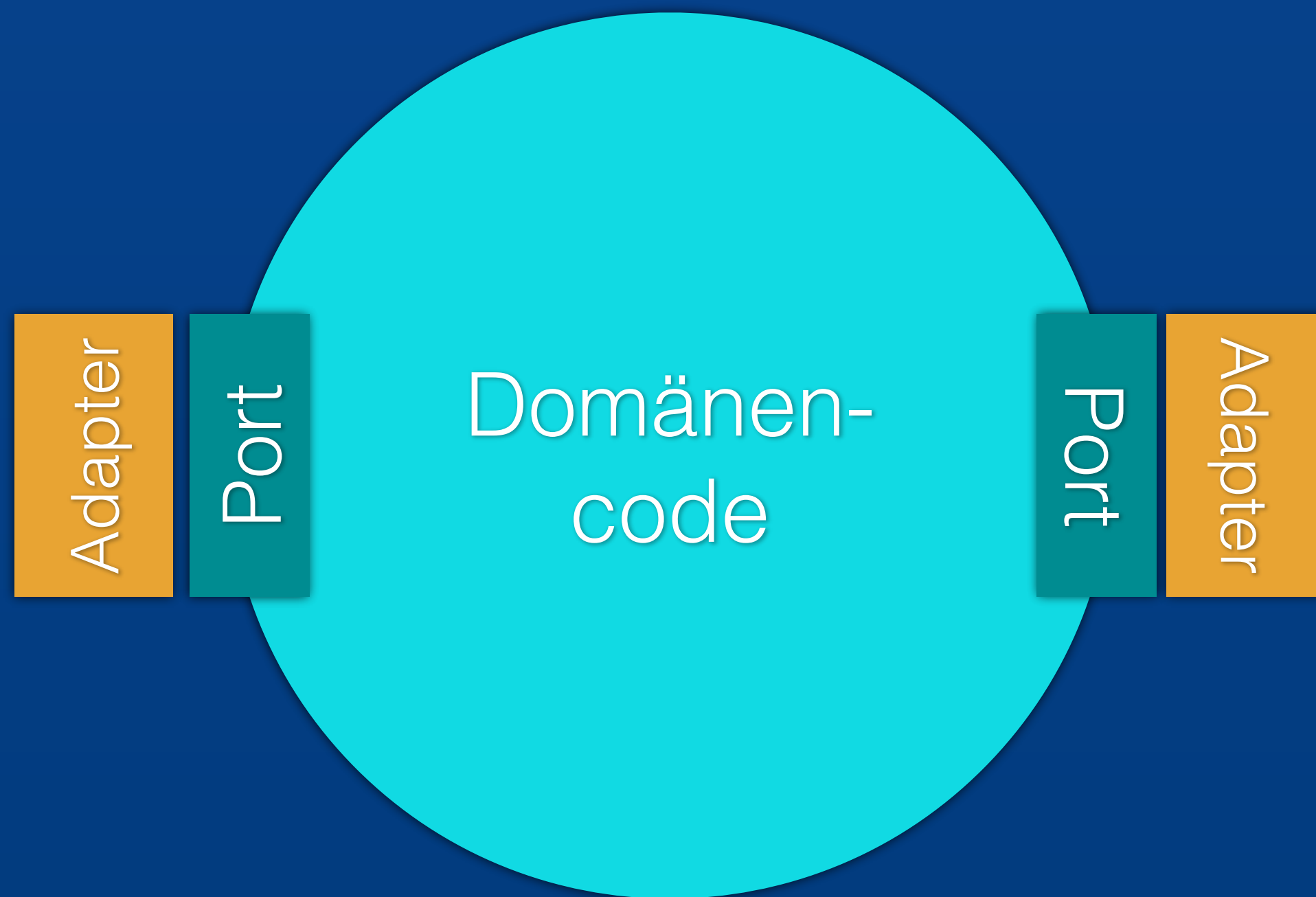


Architekturstile



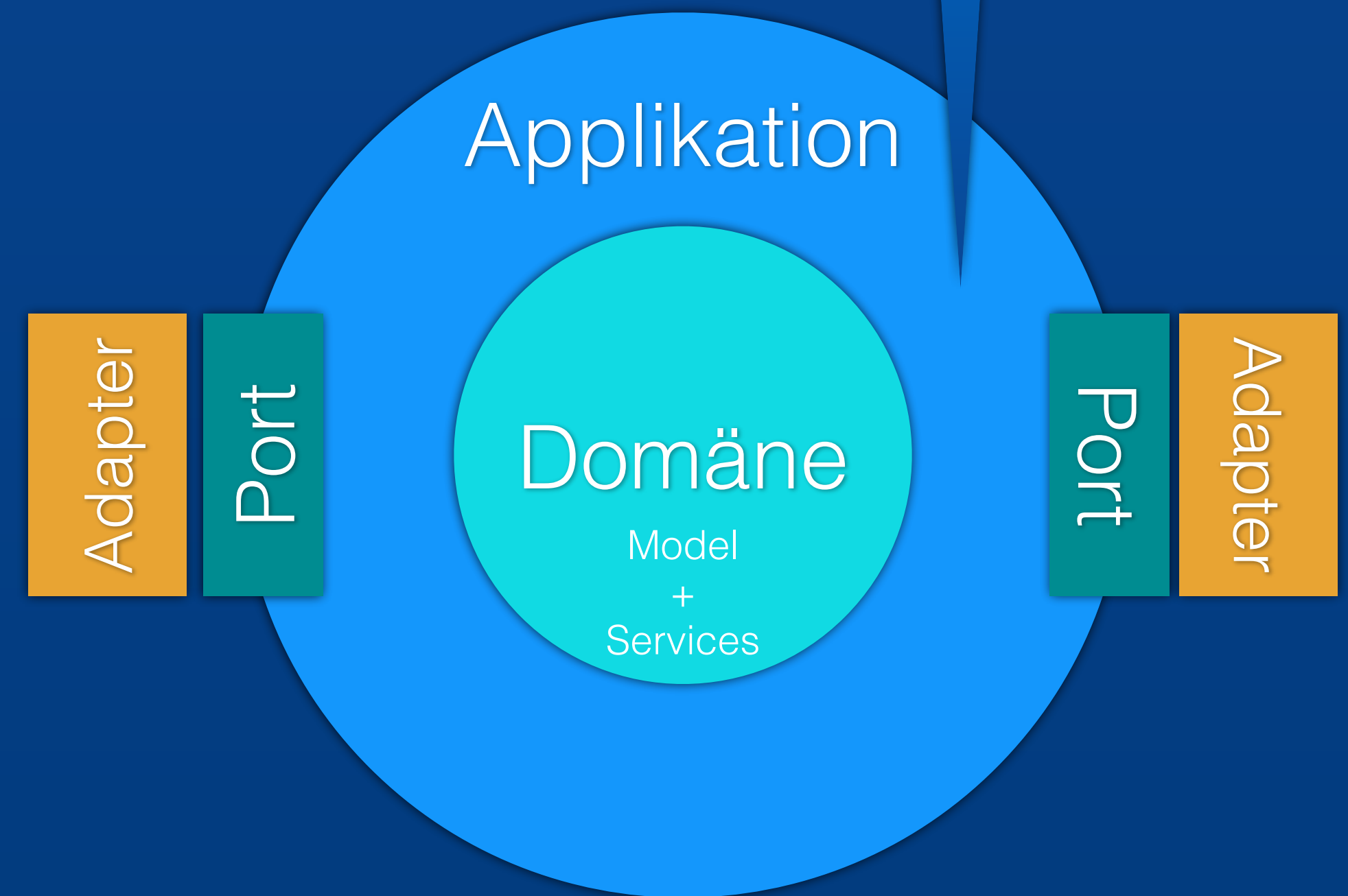
Architekturstile

Ports & Adapters ("Hexagonal")

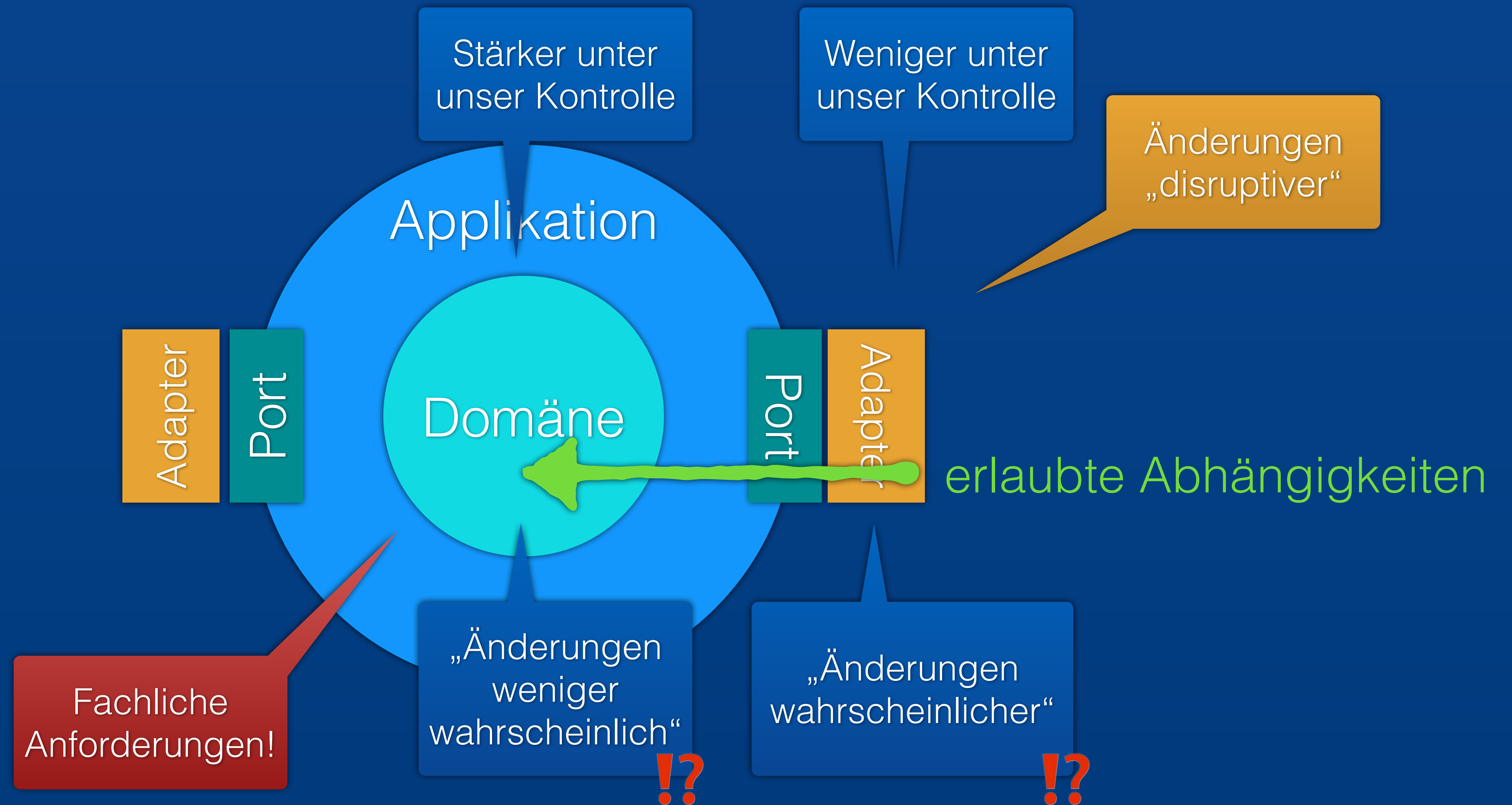


Onion

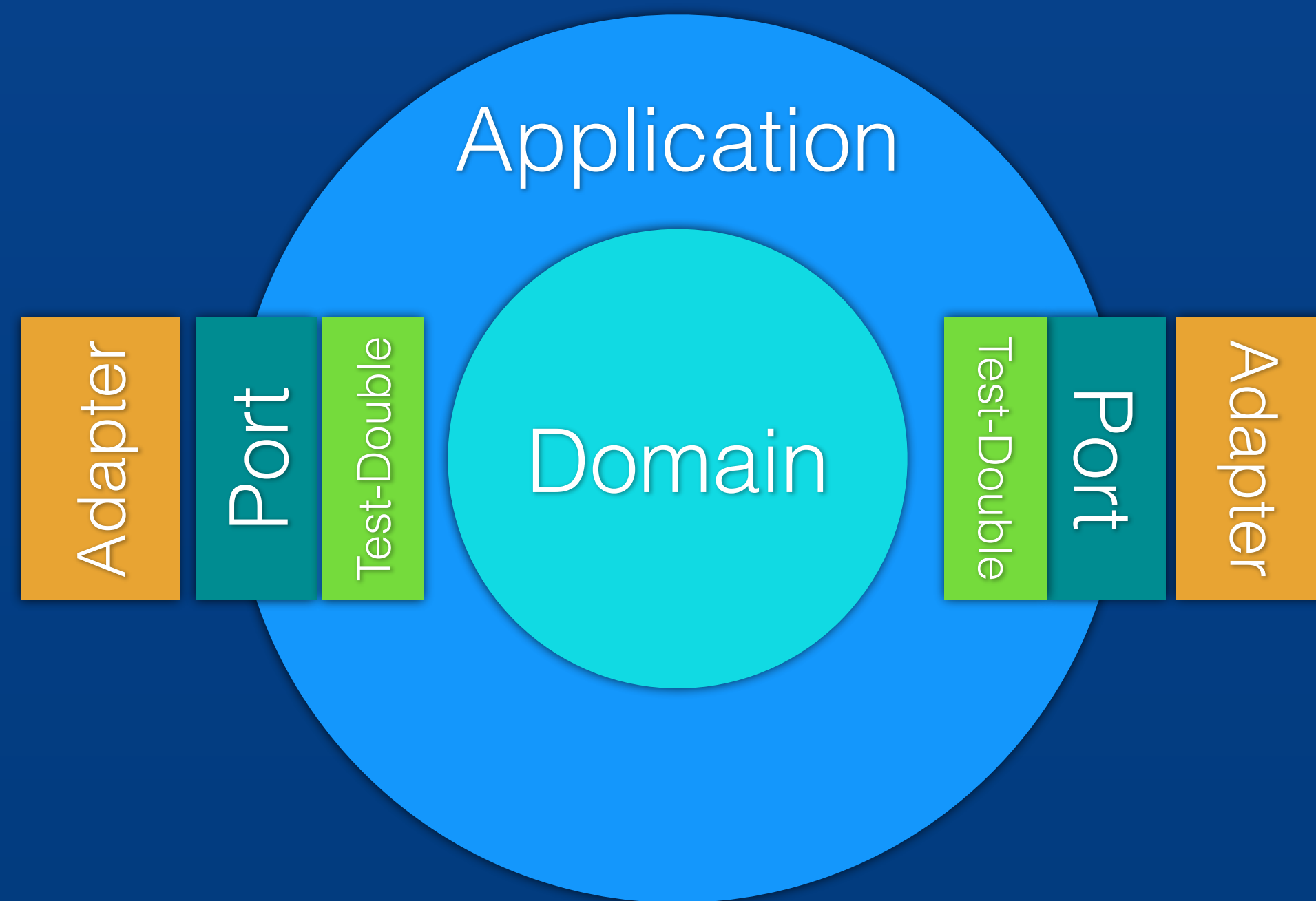
Use-Cases



Statt Schichten: Innen & außen!



Schnelle Tests für ganze Use-Cases



Schnelle (Unit-)Tests
– nicht nur Mikrotests



Teil 2

Live Demo

<https://github.com/thmuch/einfach-gut-testbar>

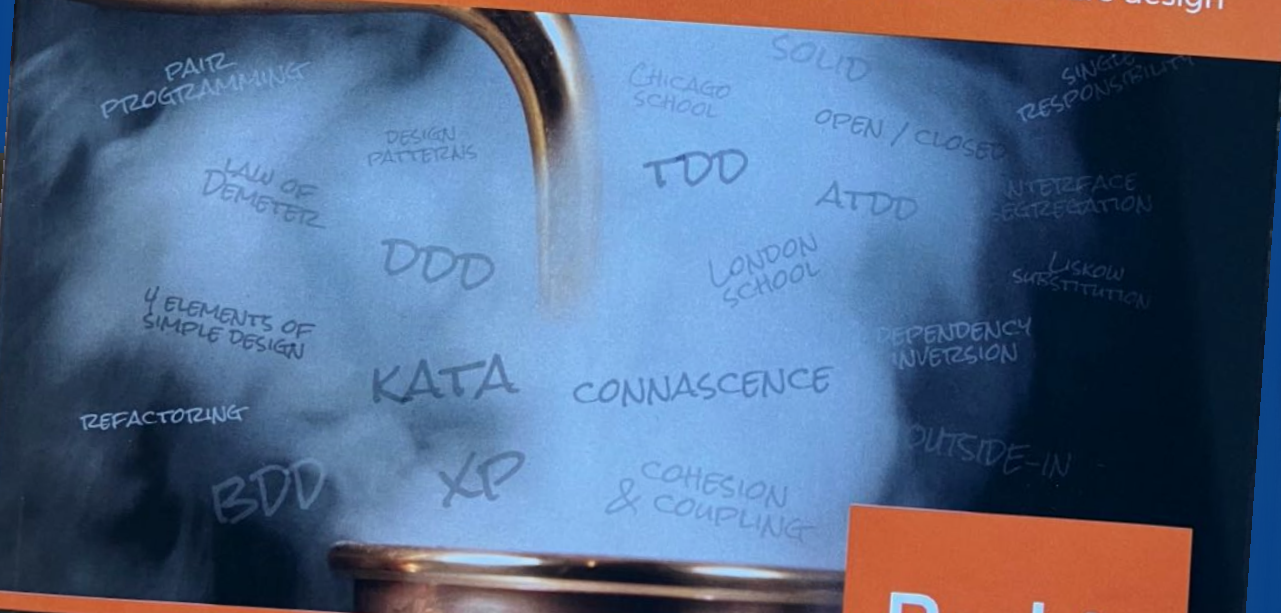
Fazit

Einfach gut testbar

- ... wenn **Code** und **Architektur** für **Testbarkeit** entworfen sind
- ... wenn **Infrastrukturcode** und **Domänencode** klar **getrennt** sind
- ... wenn **schnelle Tests** möglich sind
- ... auch für **größere Units** (> 1 Methode, > 1 Klasse)

Agile Technical Practices Distilled

A learning journey in technical practices and principles of software design



Pedro M. Santos, Marco Consolaro
and Alessandro Di Cioia

Packt
www.packt.com

WORKING EFFECTIVELY WITH LEGACY CODE

Michael C. Feathers

The Addison-Wesley Signature Series

GROWING OBJECT-ORIENTED SOFTWARE, GUIDED BY TESTS

STEVE FREEMAN
NAT PRYCE



A KENT BECK SIGNATURE
BOOK

DAVID FARLEY

MODERN SOFTWARE ENGINEERING

Doing What Works to
Build Better Software Faster

Foreword by TRISHA GEE



MANNING

Effective Software Testing

A developer's guide

Maurício Aniche

Forewords by Arie van Deursen and Steve Freeman



Clean Architecture

A Craftsman's Guide to
Software Structure and Design

Robert C. Martin Series

PRENTICE
HALL

Robert C. Martin

Forewords by James Grenning and Simon Brown

Foreword by Kevlin Henney
Afterword by Jason Gorman



Testbarkeit

Unit-Tests

Schnelle Tests

Abhängigkeiten

Architektur

Fragen?

Mikrotests

Code-Design

Größe einer Unit

Fast Feedback Loops

 @thmuch



Vielen Dank 😊



www.tk.de/IT

 @thmuch