

DOMAIN-CENTRIC?

WHY HEXAGONAL AND ONION ARCHITECTURE ARE ANSWERS TO THE WRONG QUESTION — AND WHAT TO ASK INSTEAD

Oliver Drotbohm odrotbohm info@odrotbohm.de

github.com/odrotbohm

odrotbohm (Oliver Drotbohm) · GitHub

Overview | Repositories 130 | Projects | Packages | Stars 79



Oliver Drotbohm

odrotbohm

Follow

Frameworks & Architecture in the Spring engineering team, OpenSource enthusiast, all things Java, DDD, REST, software architecture, drums & music

3.9k followers · **32** following

- Spring Open Source Engineering
- Dresden, Germany
- 17:06 - same time
- www.odrotbohm.de
- @odrotbohm.de
- @odrotbohm@chaos.social
- odrotbohm
- in/odrotbohm

Pinned

spring-projects/spring-modulith Public
Modular applications with Spring Boot
Java ☆ 1.1k 182

xmolecules/jmolecules Public
Libraries to help developers express architectural abstractions in Java code
Java ☆ 1.5k 116

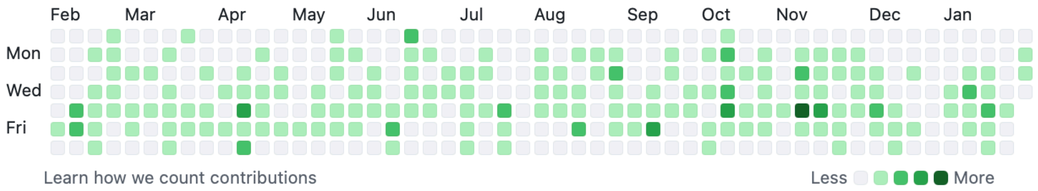
xmolecules/jmolecules-integrations Public
Technology integration for jMolecules
Java ☆ 114 27

spring-restbucks Public
Implementation of the sample from REST in Practice based on Spring projects
Java ☆ 1.3k 422

spring-playground Public
A collection of tiny helpers for building Spring applications
Java ☆ 101 11

lectures Public
Lecture scripts and slides I use during the Software Engineering course at TU Dresden
Java ☆ 75 25

1,717 contributions in the last year



Learn how we count contributions

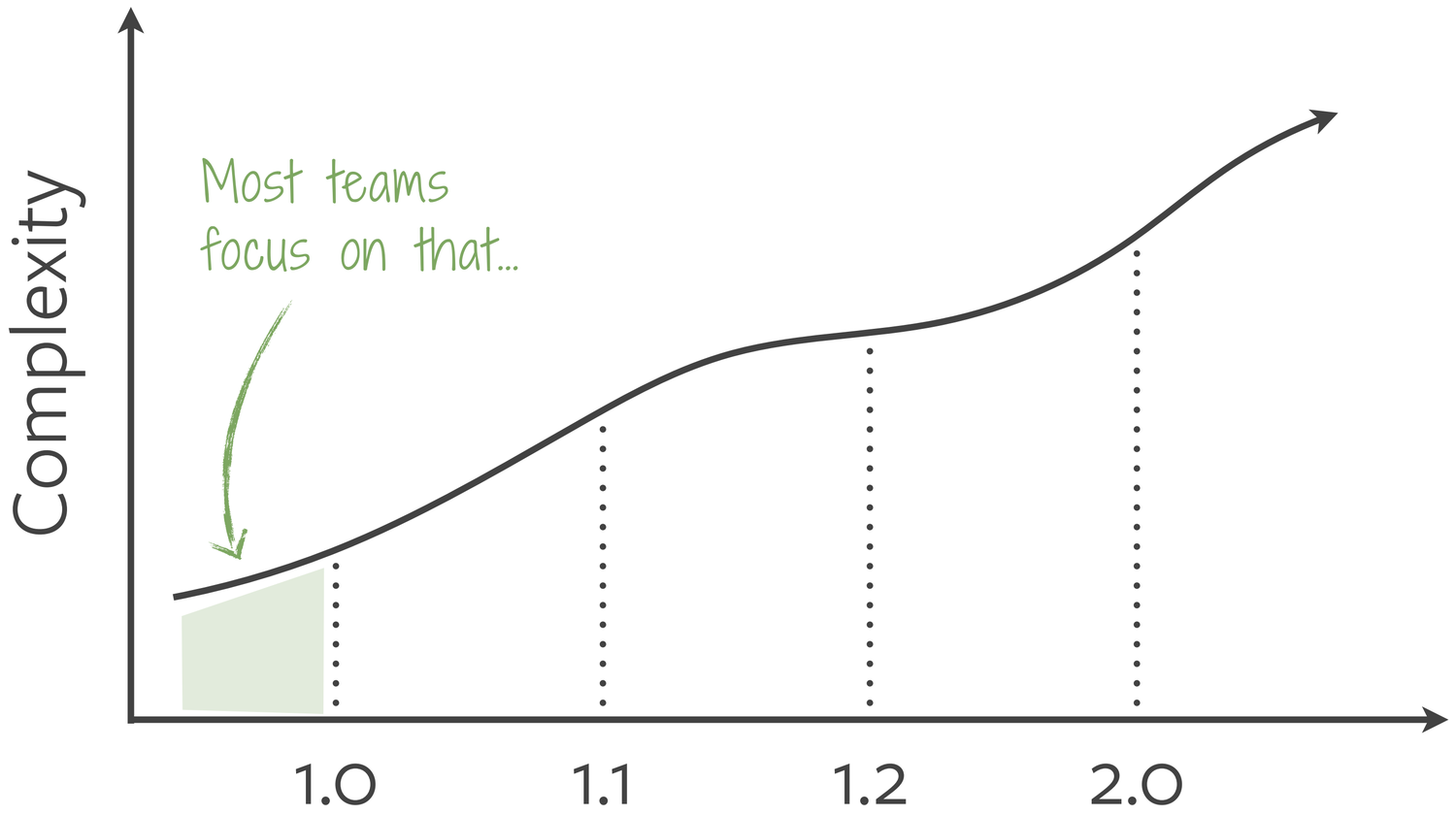
Less More

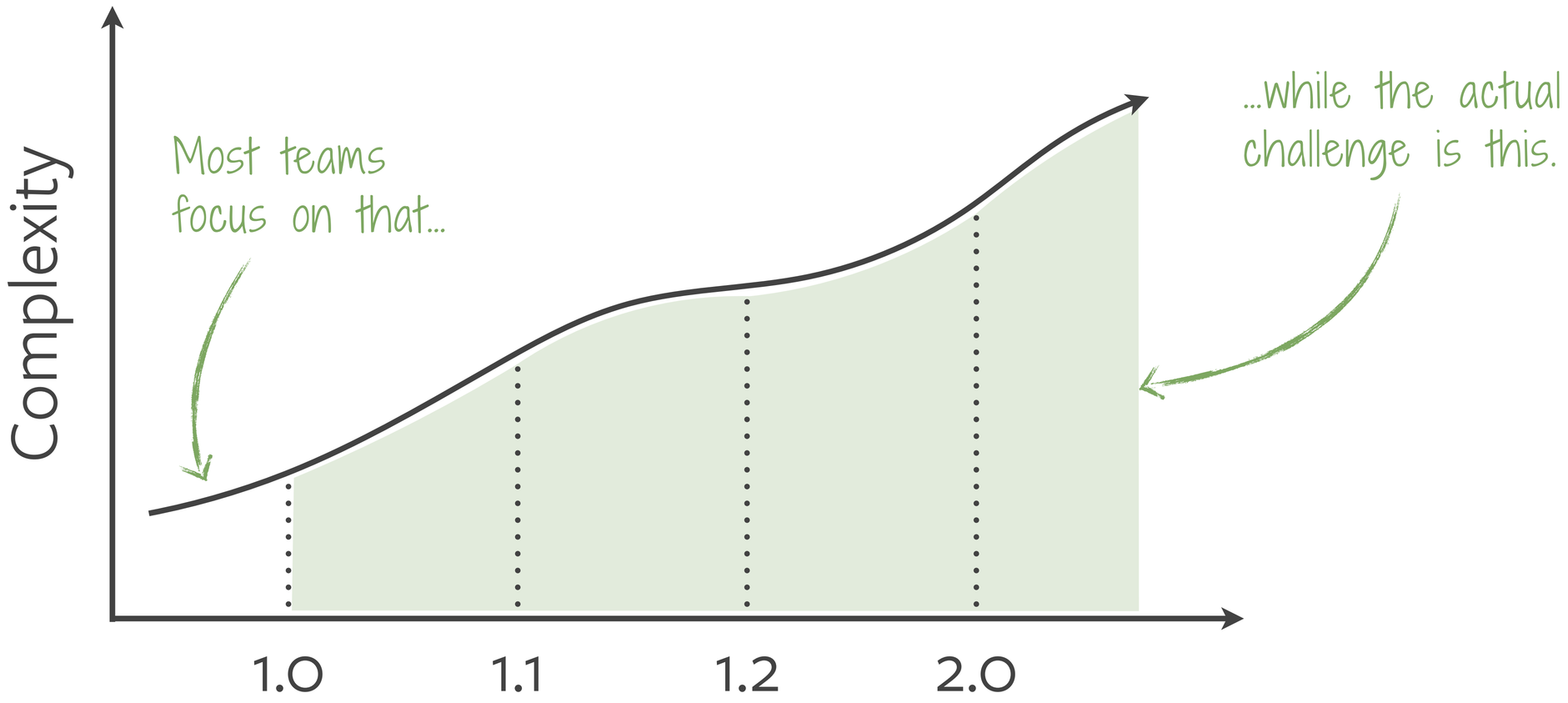
2026 2025 2024 2023 2022 2021 2020

@spring-projects @xmolecules @spring-io More

Activity overview

Code review

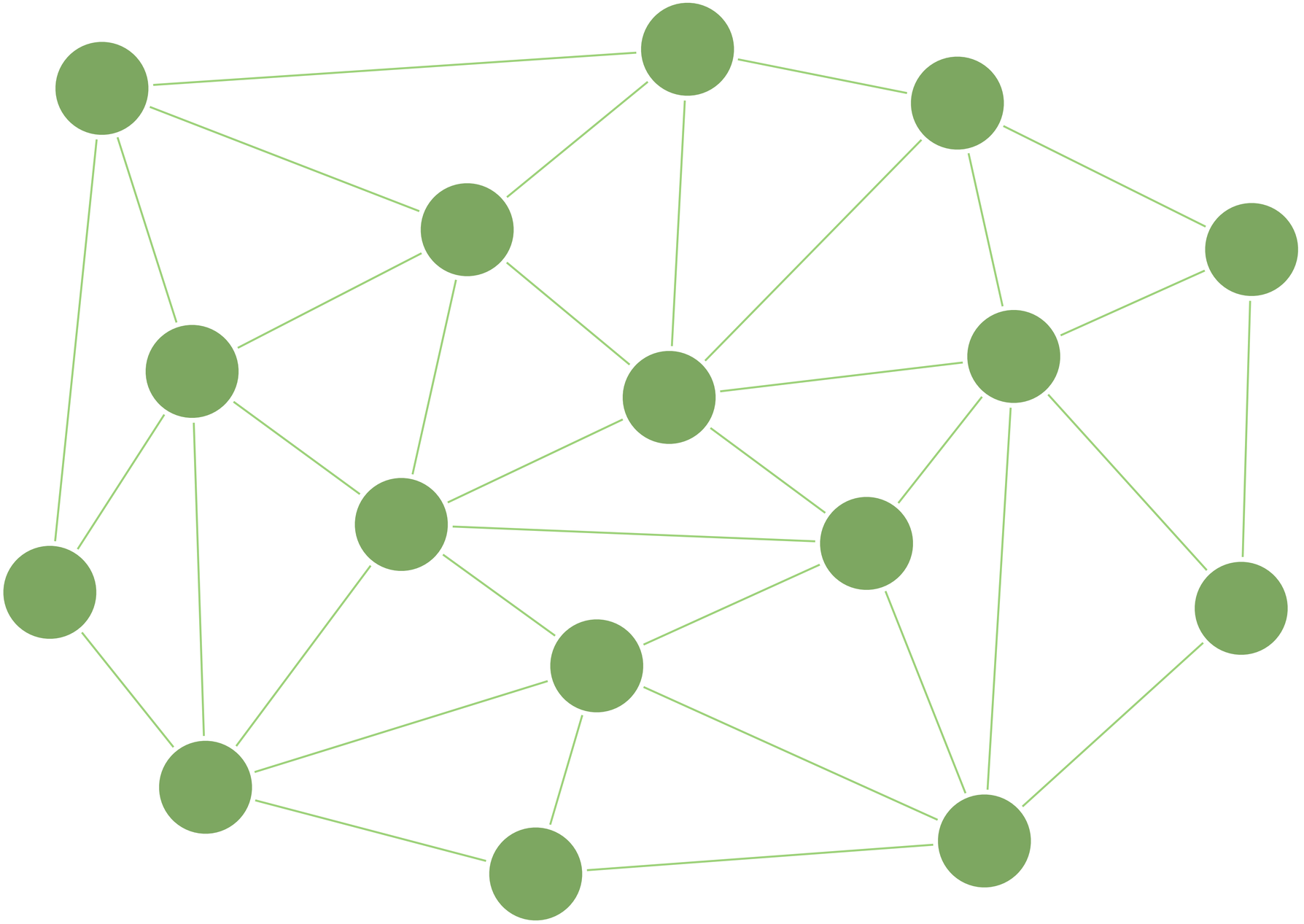

Complexity
Most teams
focus on that...
1.0
1.1
1.2
2.0

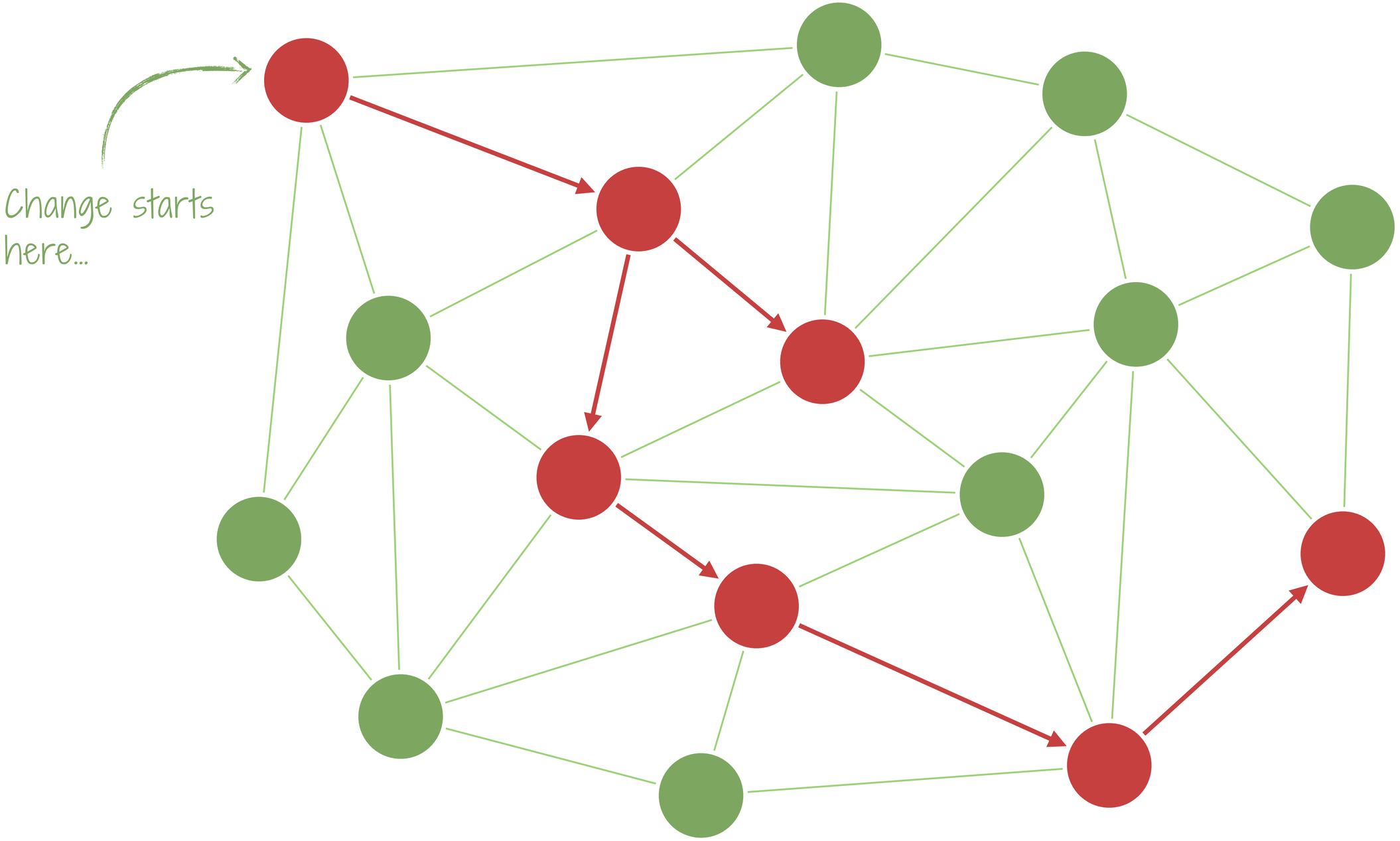

Complexity
Most teams
focus on that...
...while the actual
challenge is this.
1.0
1.1
1.2
2.0

Cost of Software
≈
Cost of Change

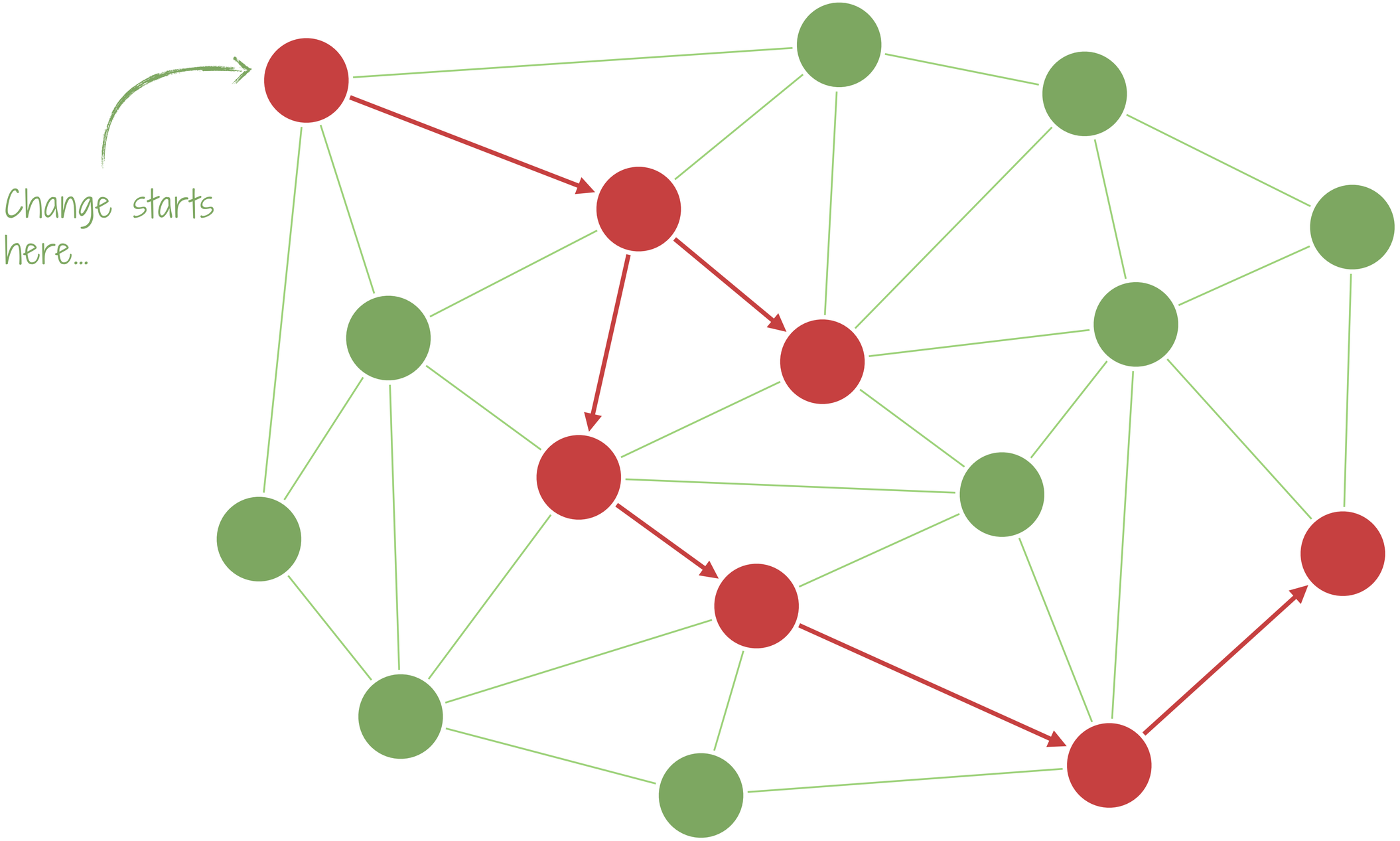
Beck - A Daily Practice of Empirical Software Design — https://www.youtube.com/watch?v=yBEcq23OgB4 (2023)
Yourdon, Constantine - Structured Design — Fundamentals of a Discipline of Computer Program and Systems Design (1979)

Cost of Change
≈
Coupling

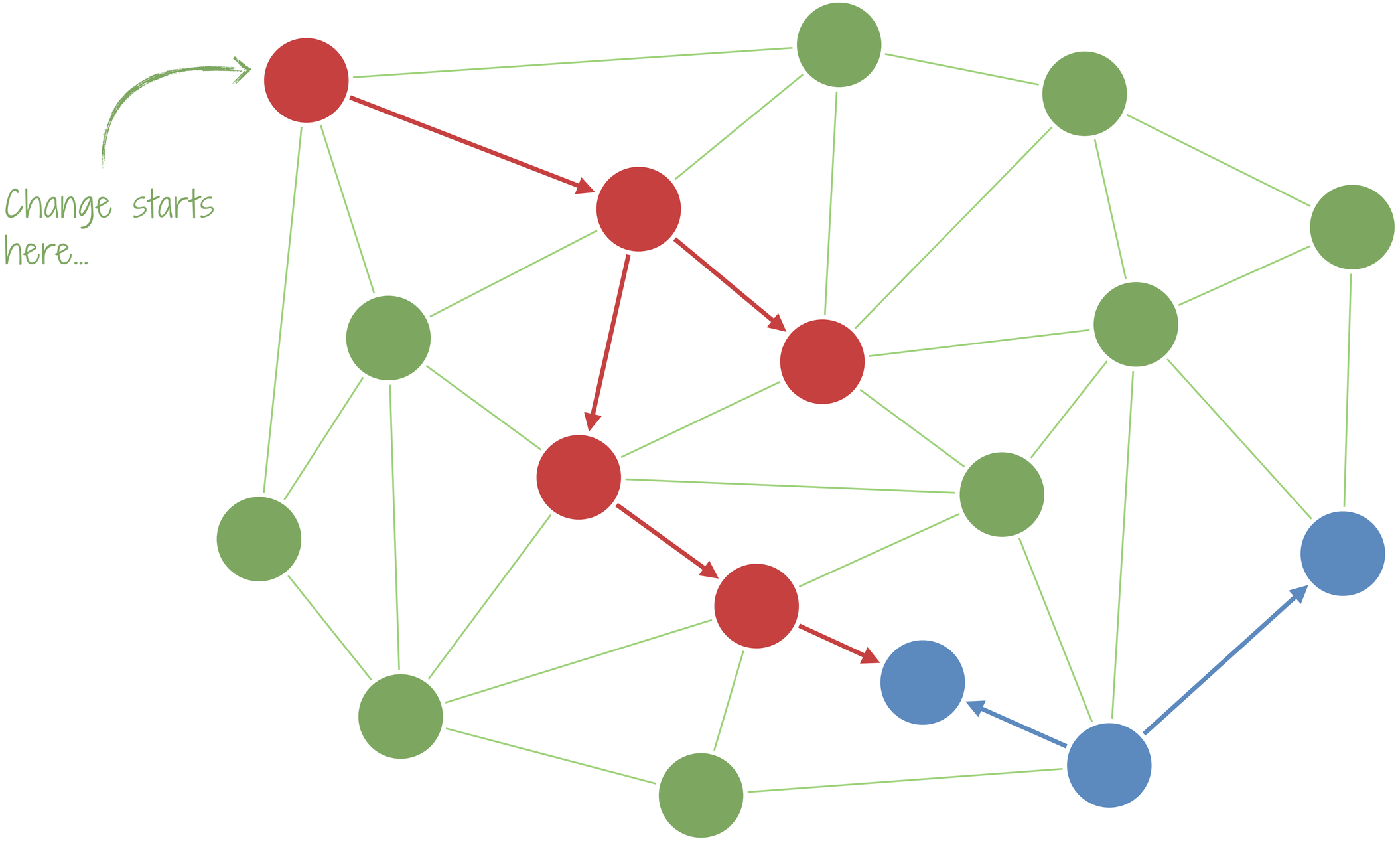


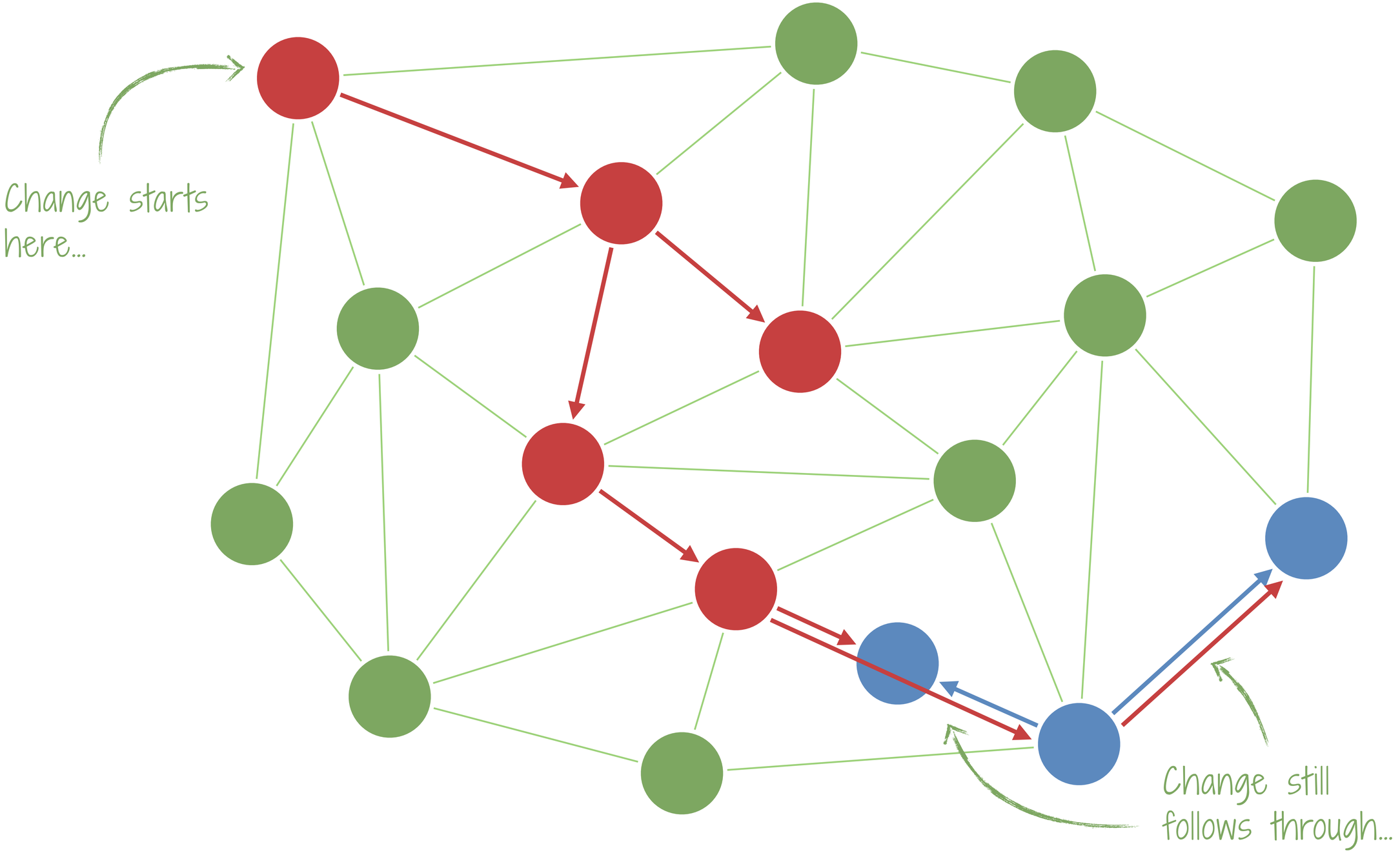

Change starts here...

Not all coupling is born equal


Change starts here...

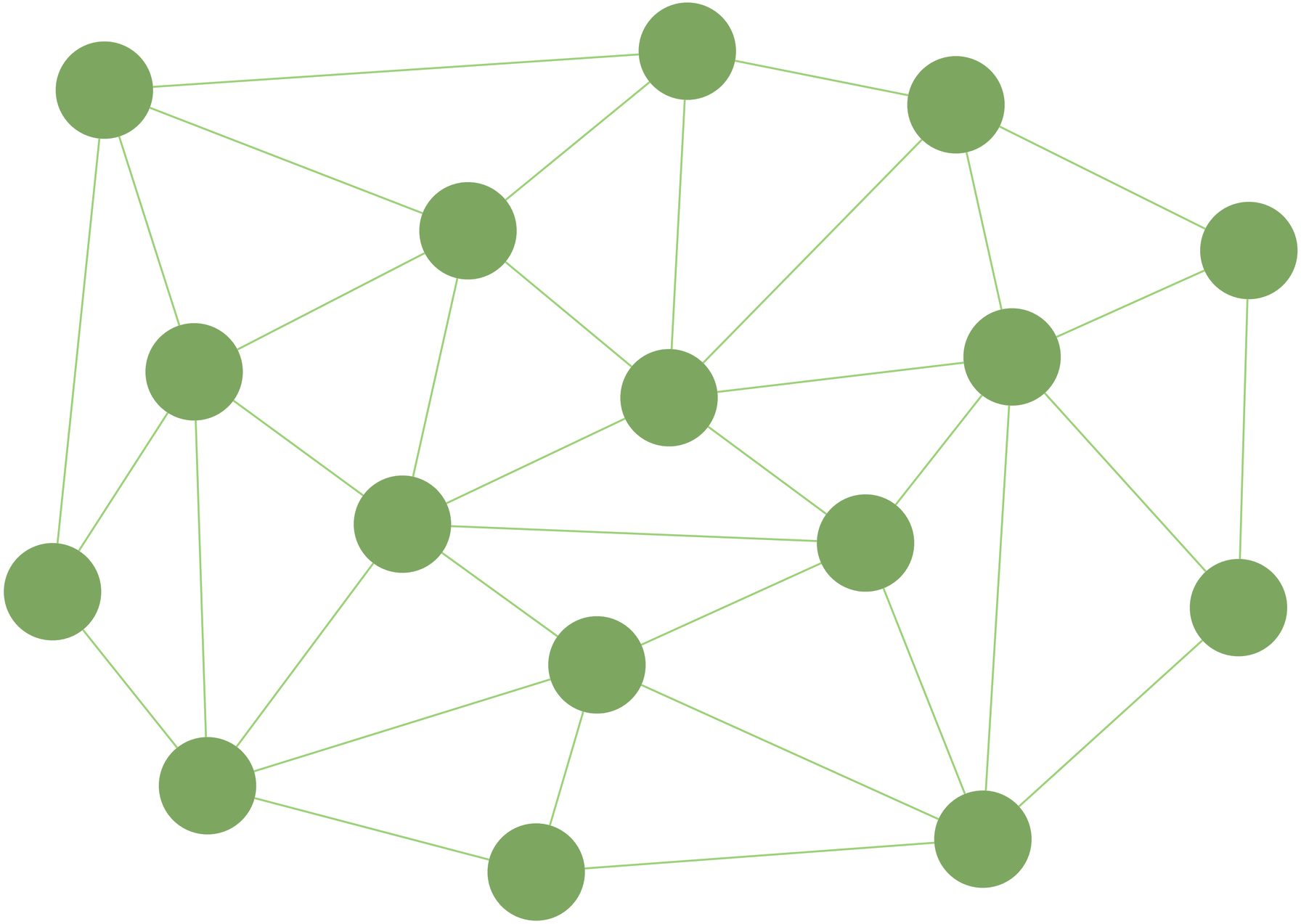
Decoupling has its cost, too

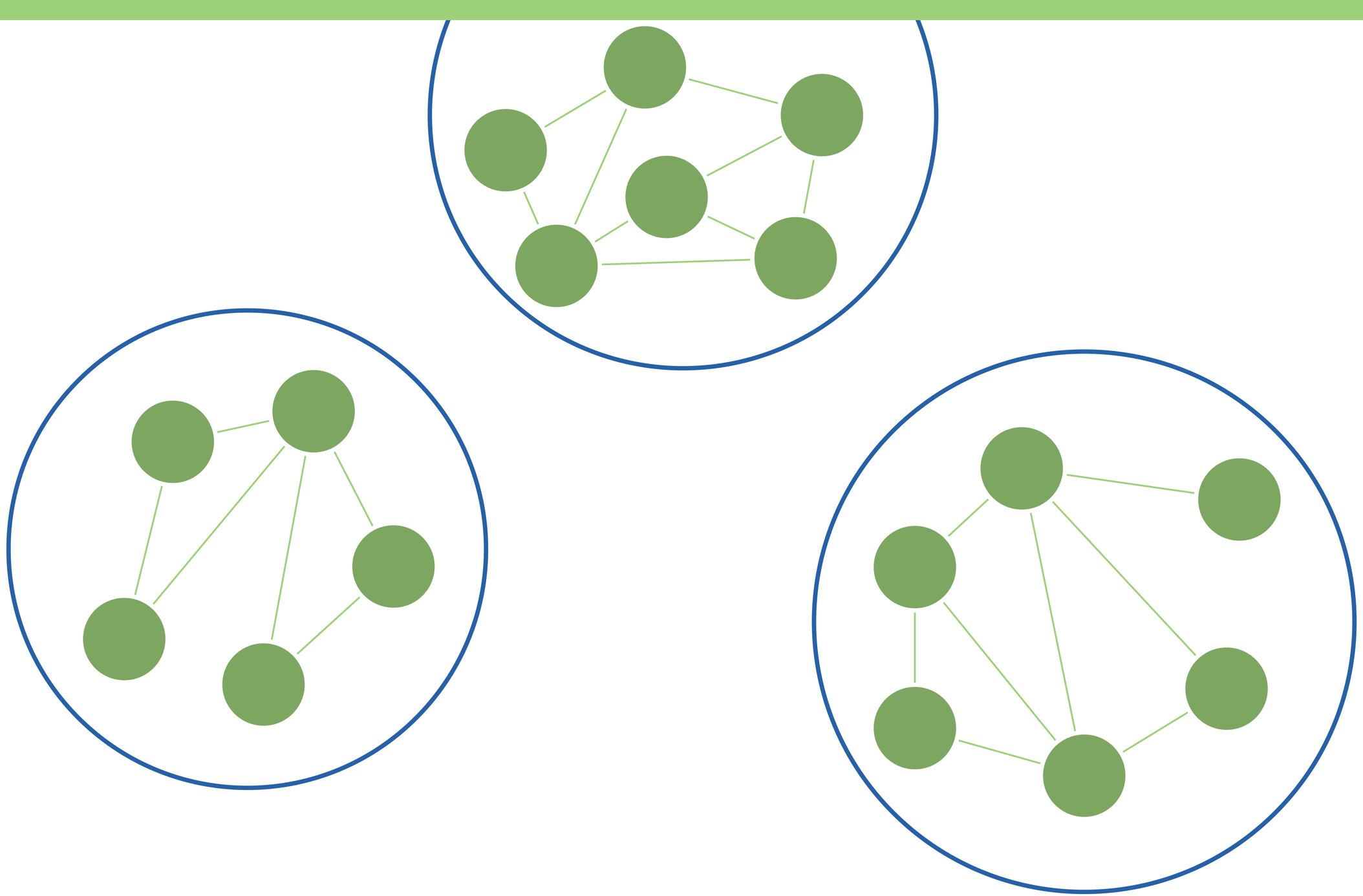

Change starts here...


Change starts here...
Change still follows through...

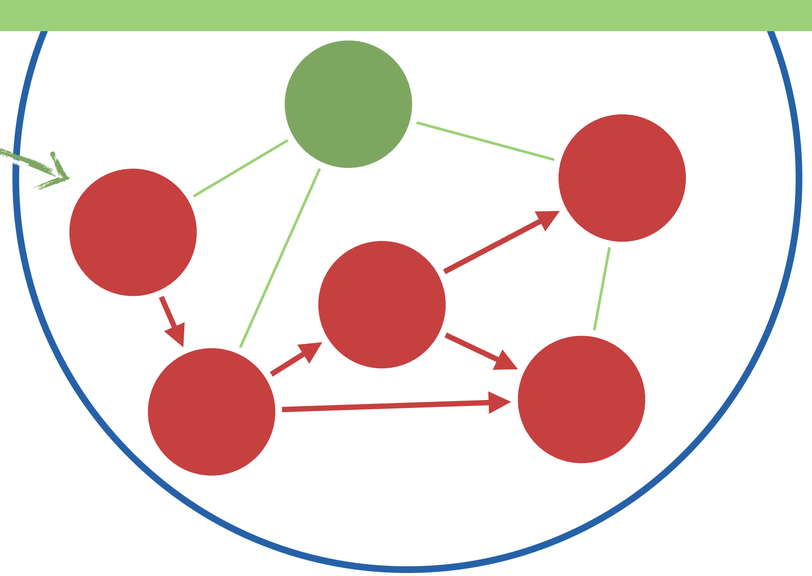
Cohesion

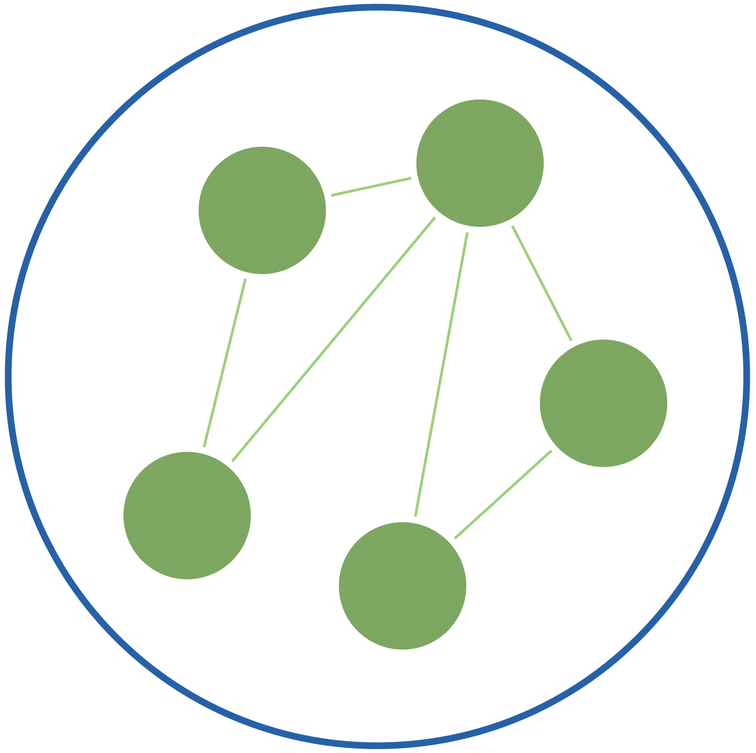
Cohesion is coupling in the right™ places

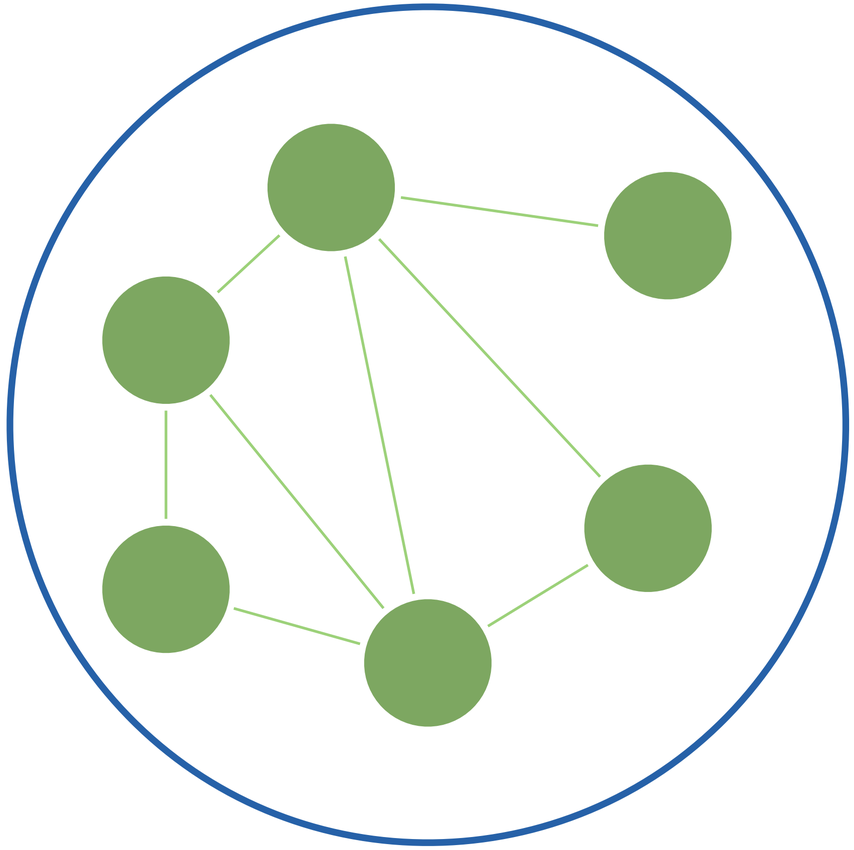




Change starts here...







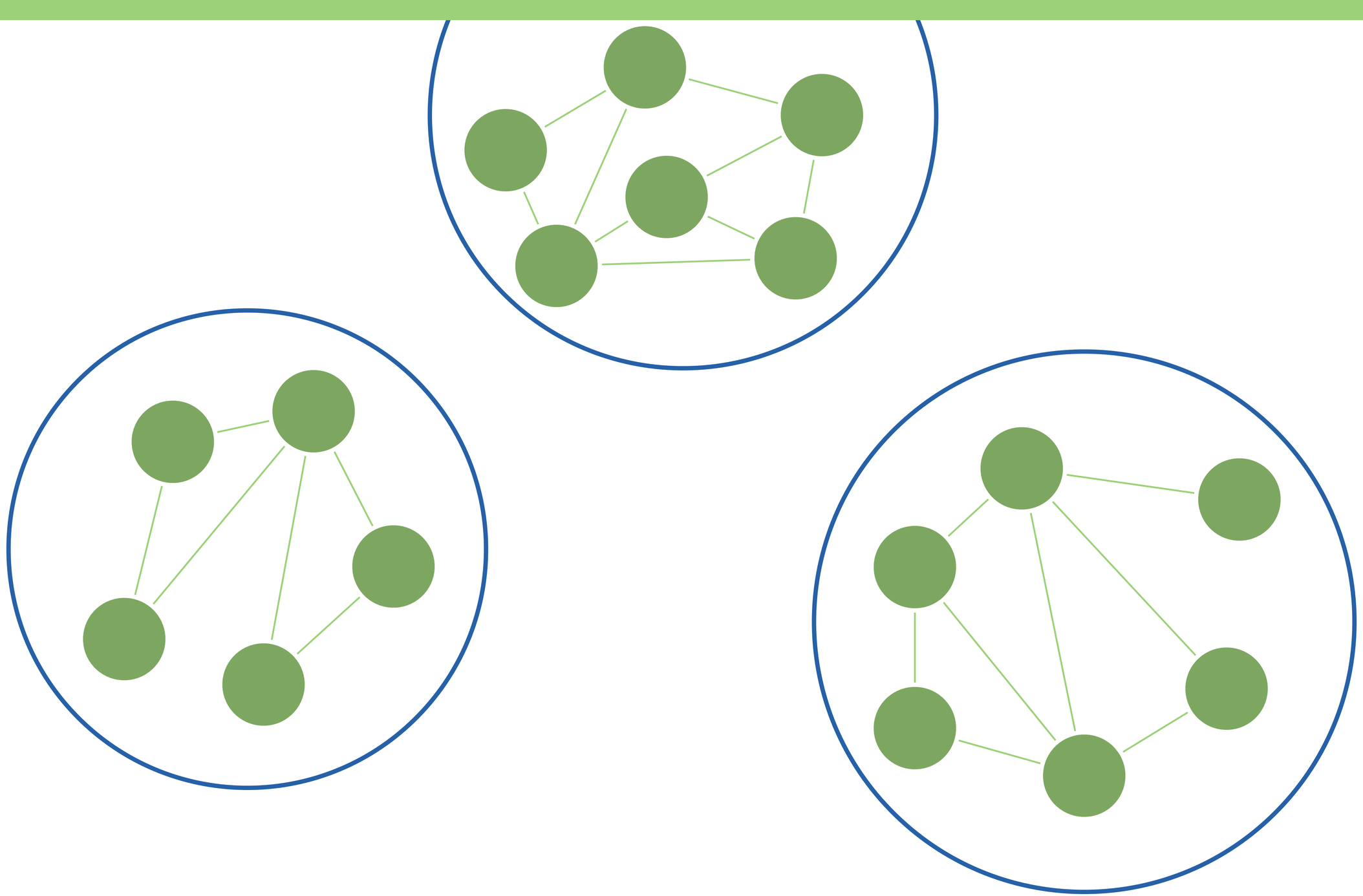


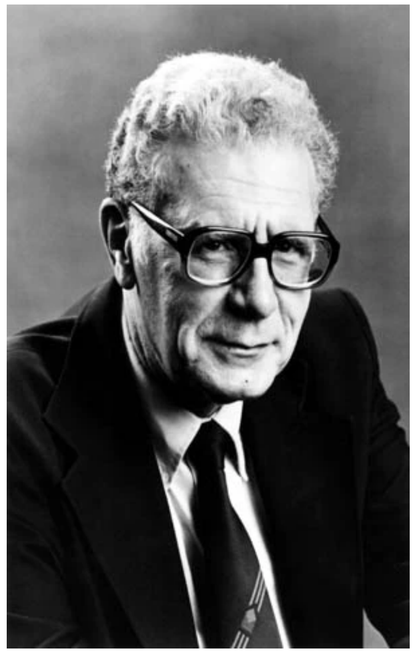
Oliver Drotbohm
@odrotbohm@chaos.social

A software system's structure is essentially a formalized bet on change patterns you anticipate having to deal with in the future.

14. Jan. 2025, 11:55 · · Ivory for Mac

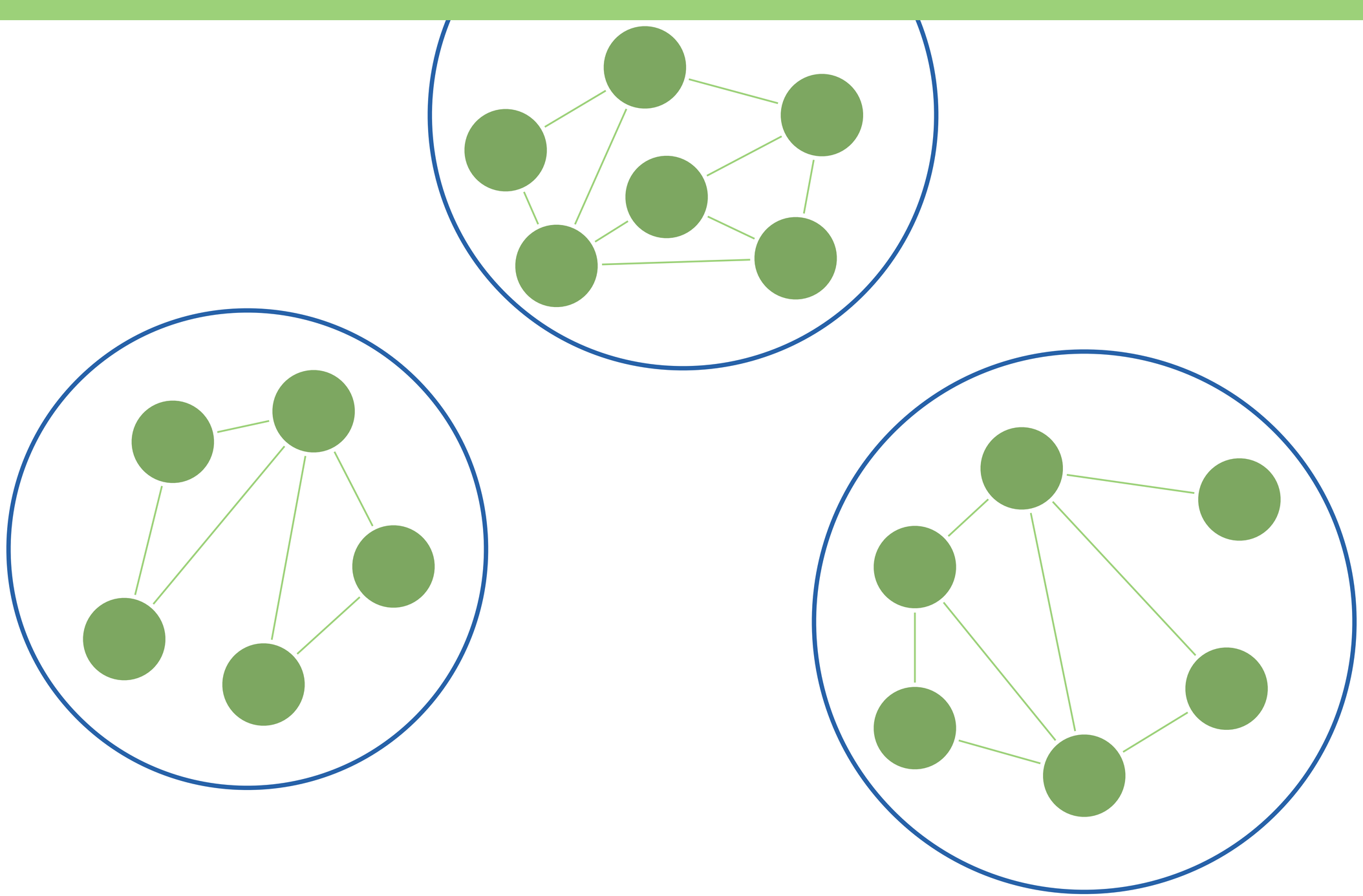
https://chaos.social/@odrotbohm/113826340530554922

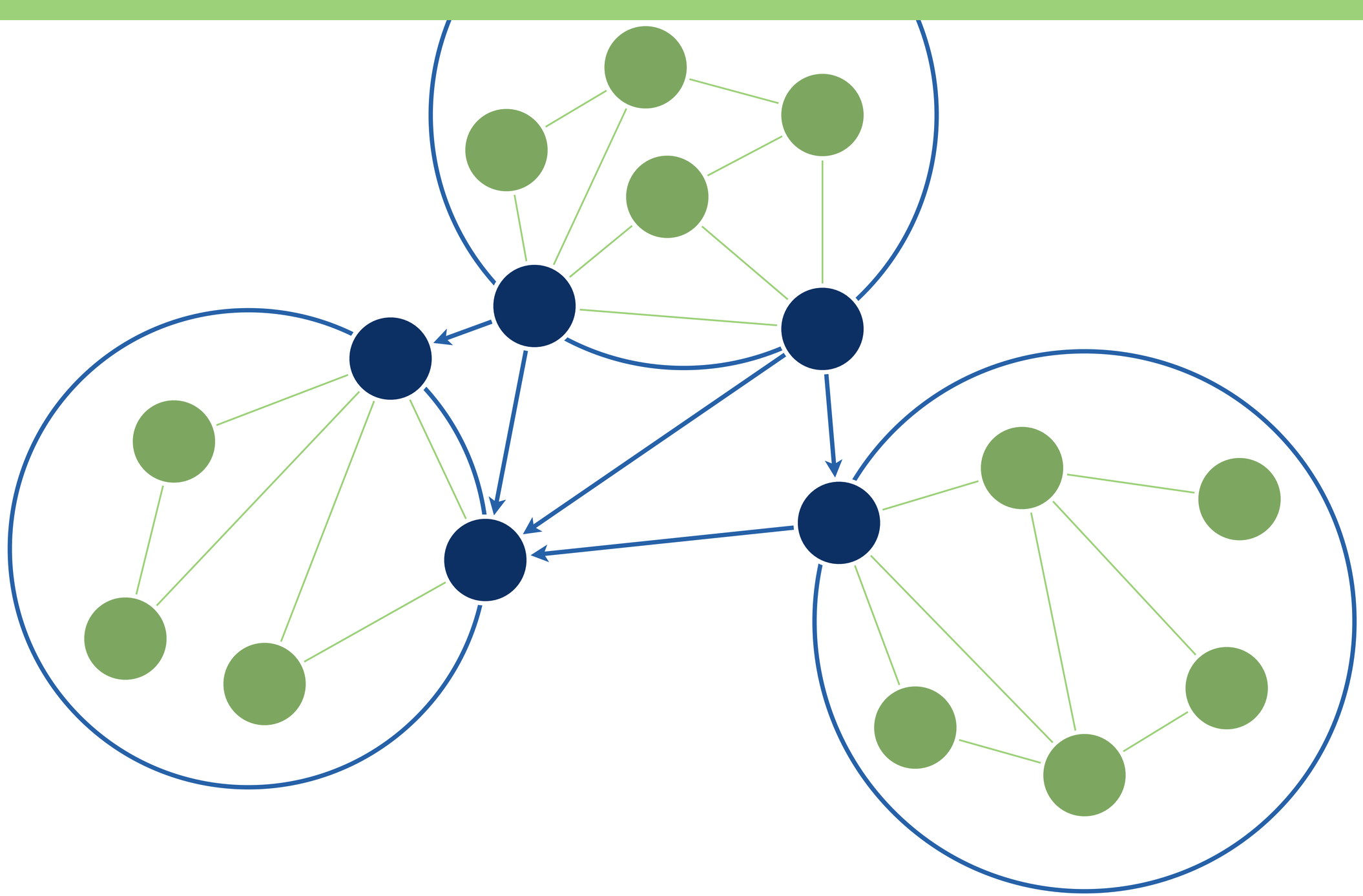




”A system is never the sum of its *parts*, it’s the product of their *interactions*.”

— Russel L. Ackoff



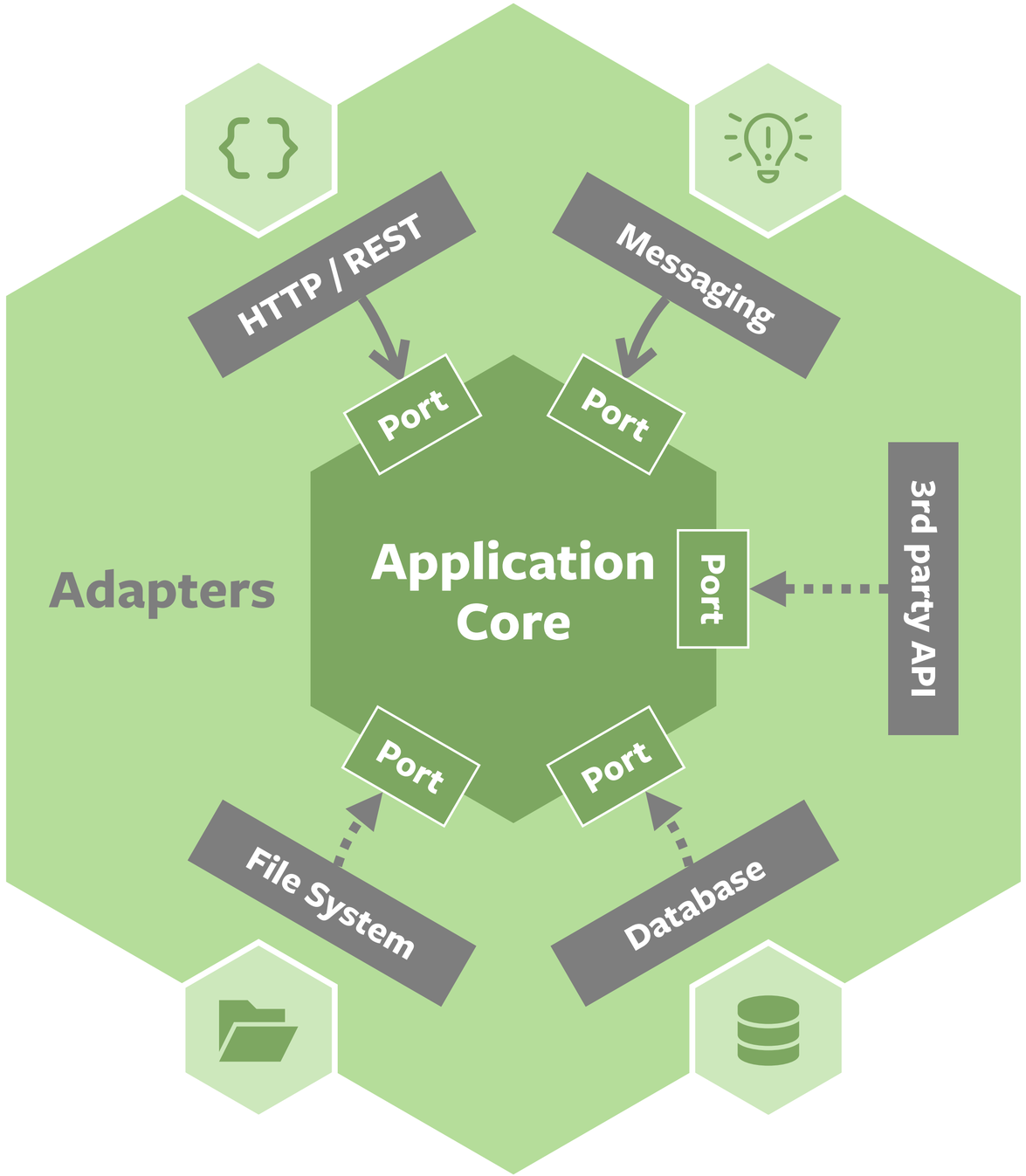


Software design is the act of modeling units of strong cohesion, loosely coupled to each other.

Separation of Concerns Architectures

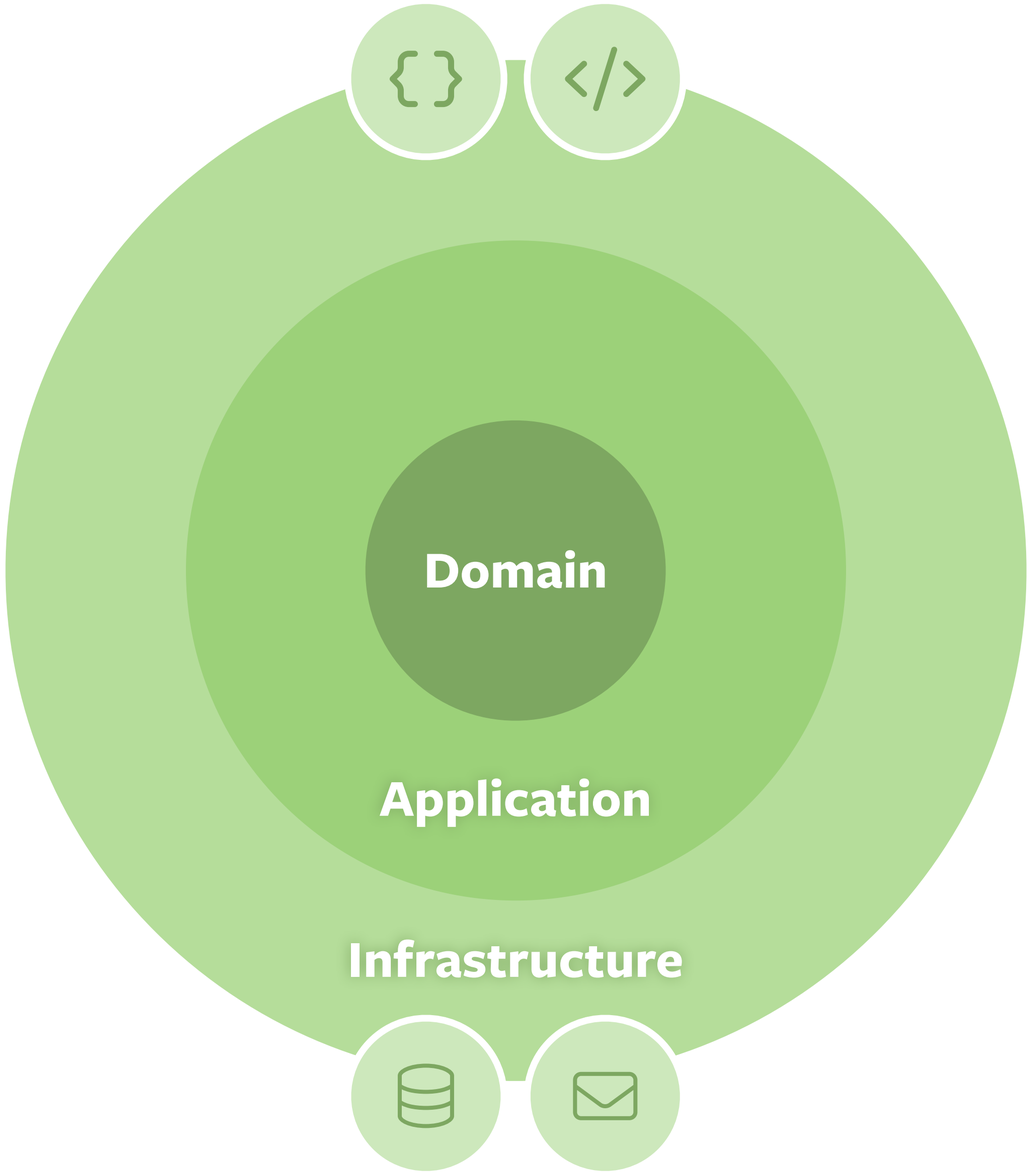
Hexagonal Architecture

2005


HTTP / REST
Messaging
Port
Port
Adapters
Application Core
Port
3rd party API
Port
Port
File System
Database

Onion Architecture

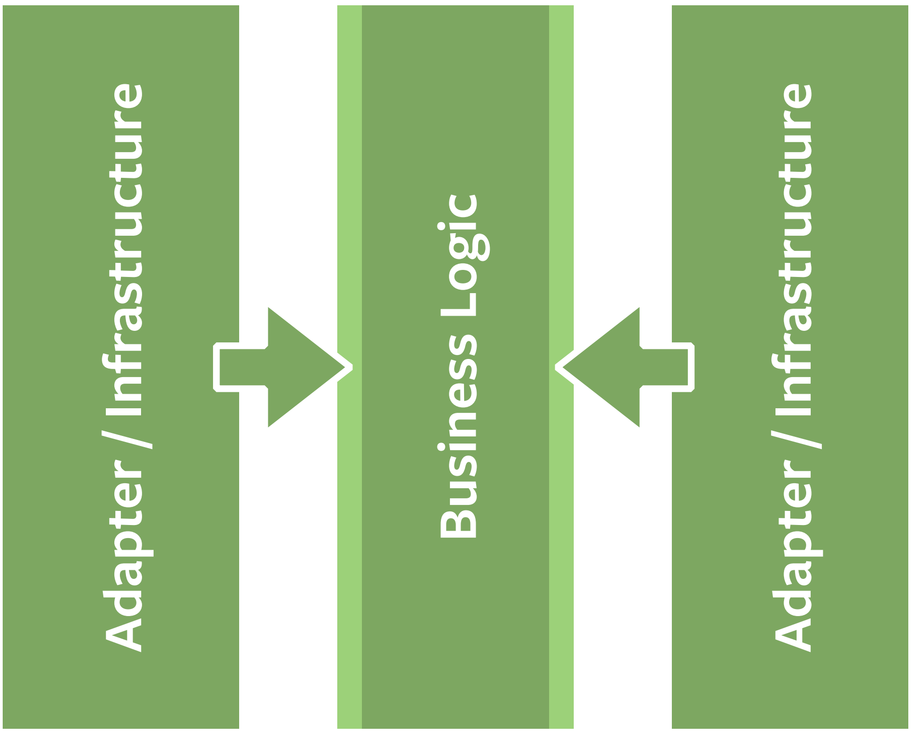
2008

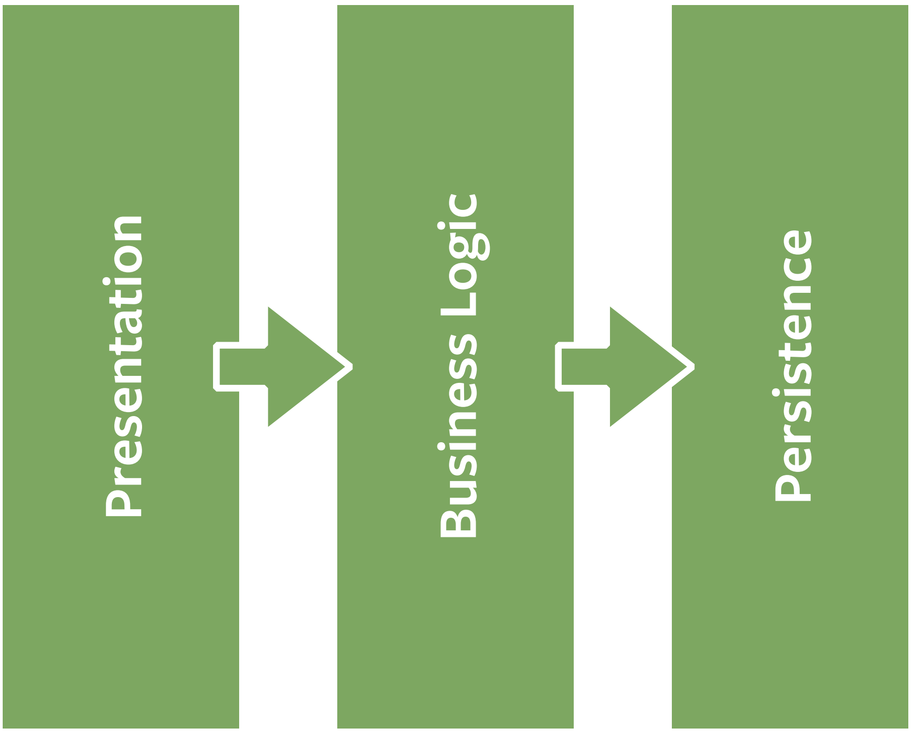

Domain
Application
Infrastructure

Layered Architecture

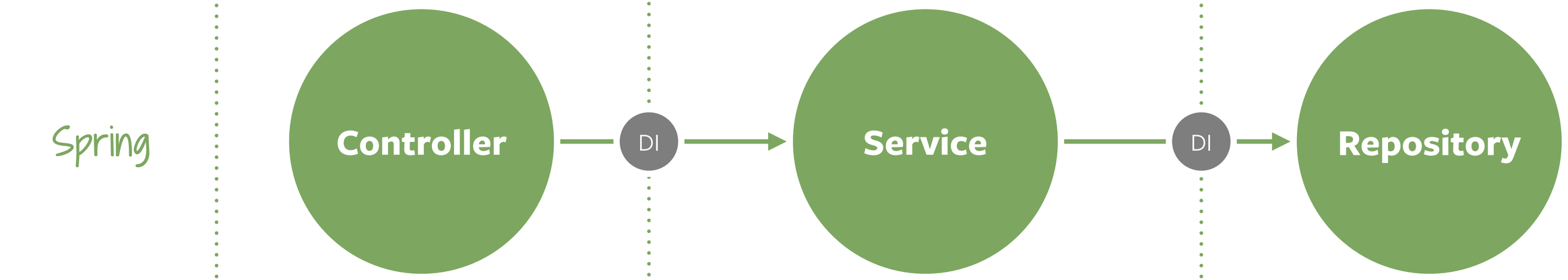
Hexagonal- / Onion-Architecture

Hexagonal- / Onion-Architecture





Layered Architecture



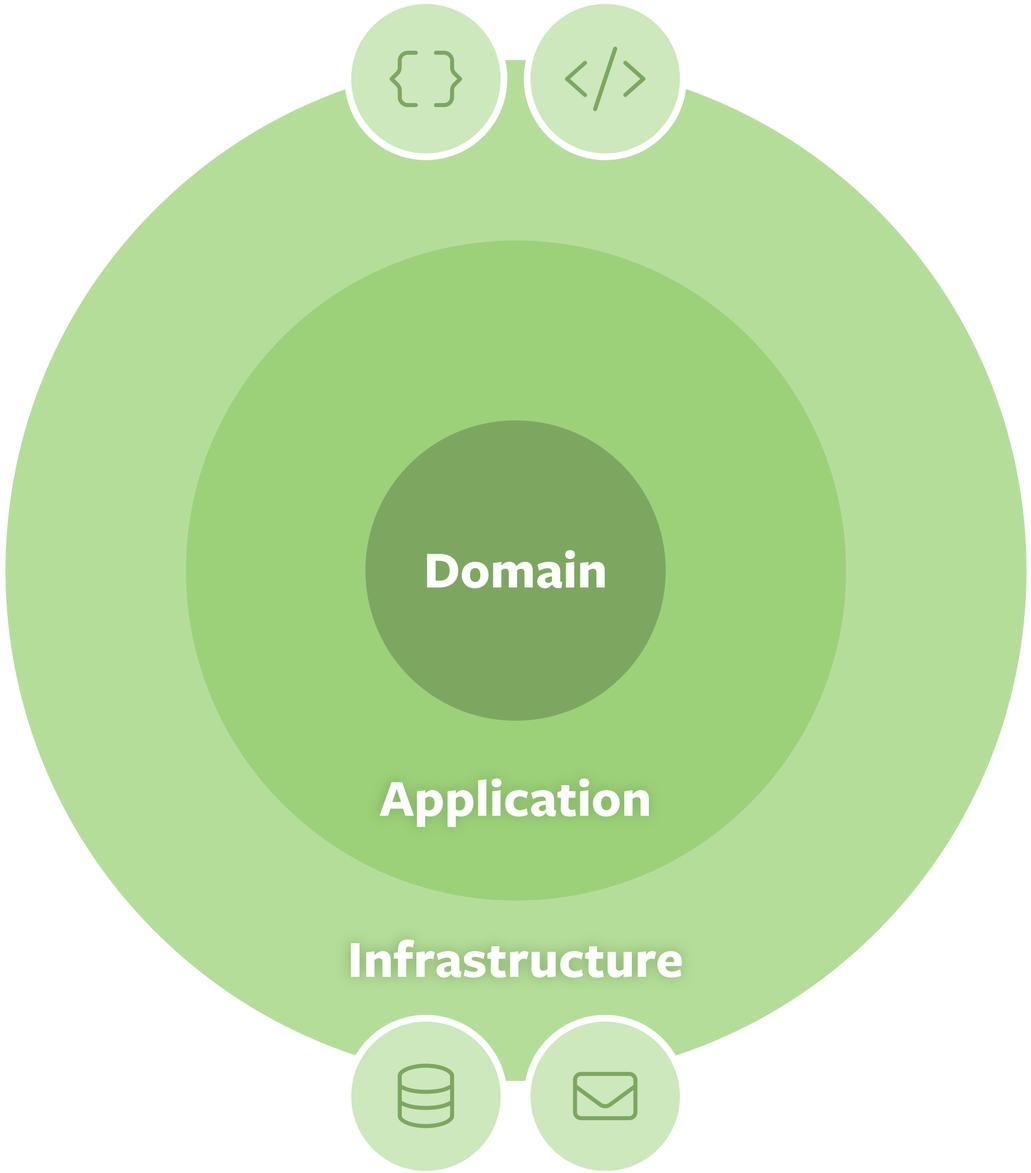
Layered	Presentation →	Business →	Persistence
Hexagonal	Driving Adapter →	Port ← Application → Port	← Driven Adapter
Onion	Infrastructure →	Application / Domain	← Infrastructure

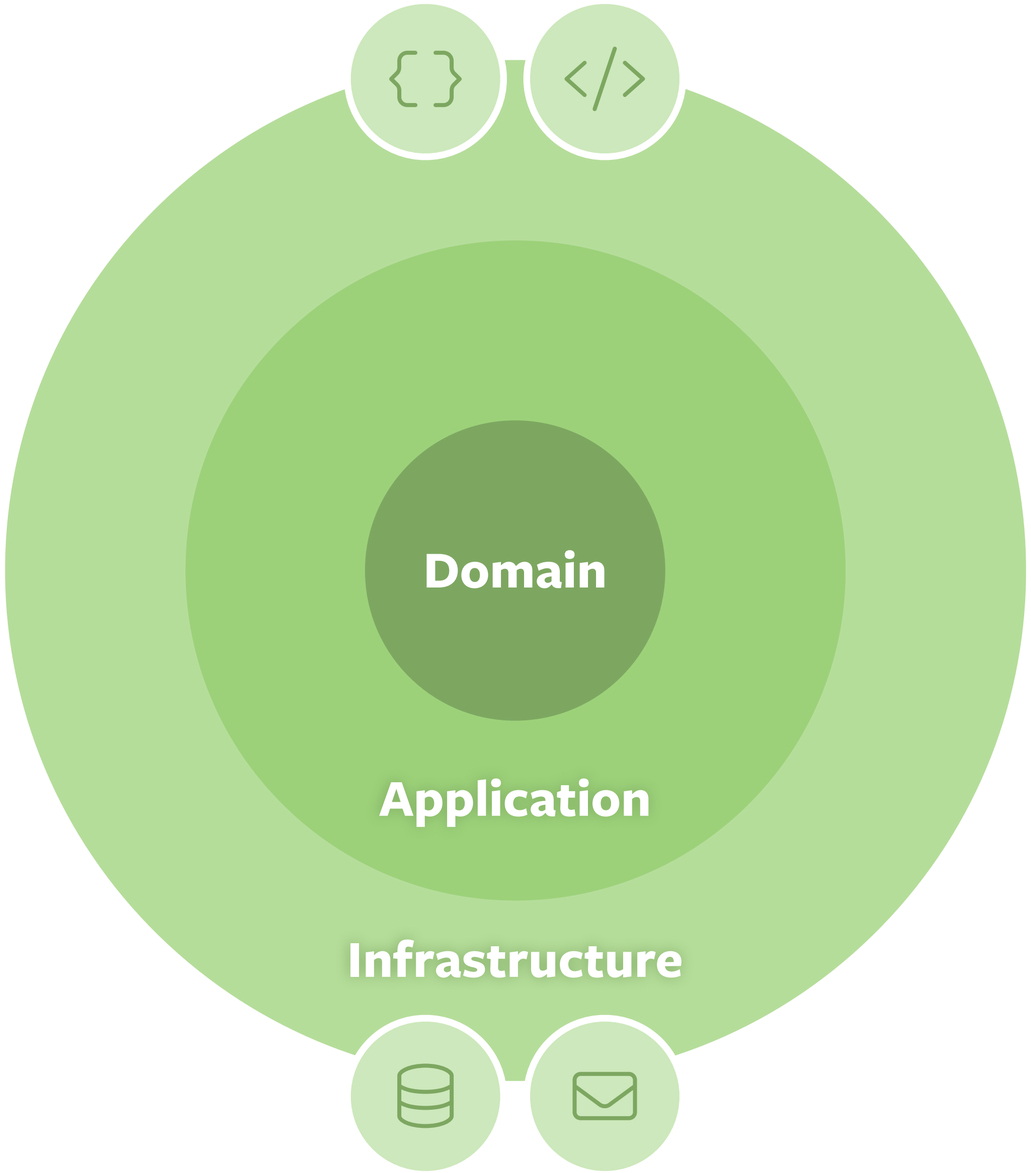
Goals & Effects

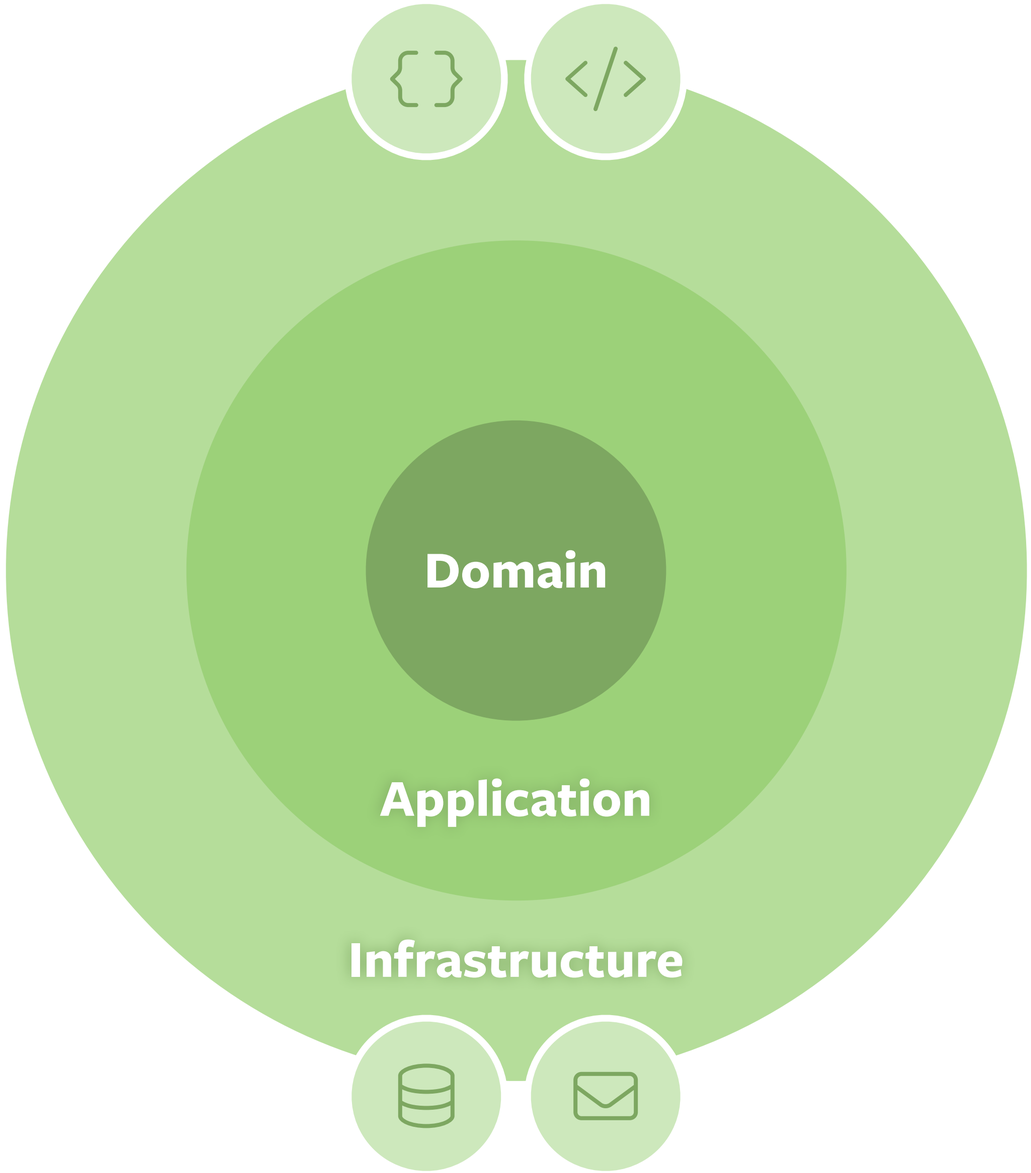
Testability

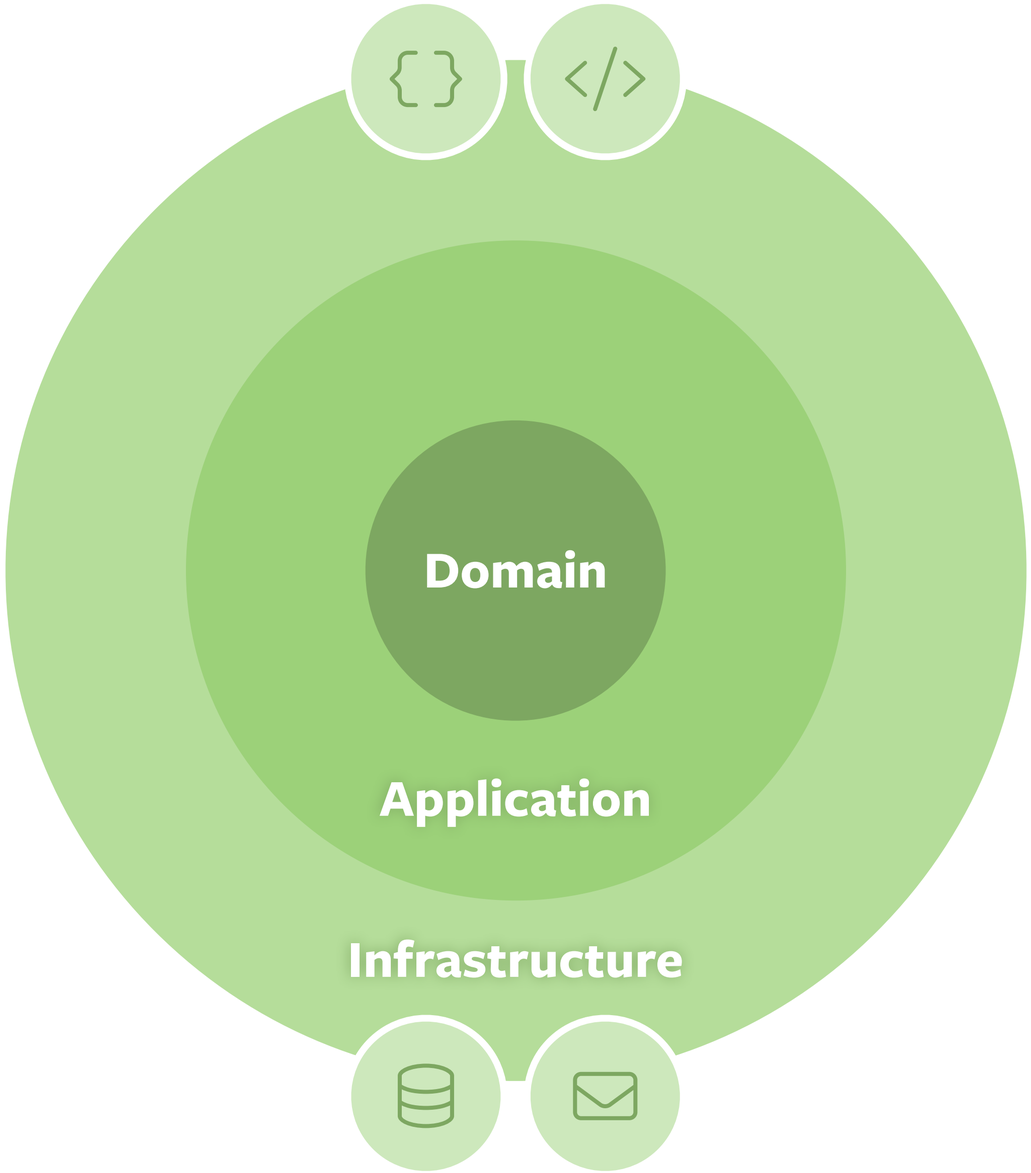
Technology-Free Domain

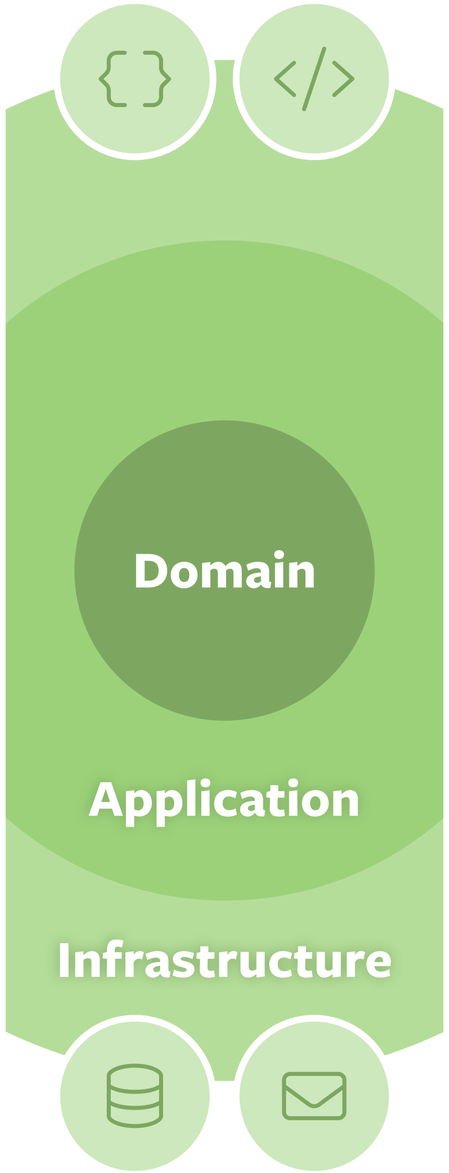
Domain-Centric?

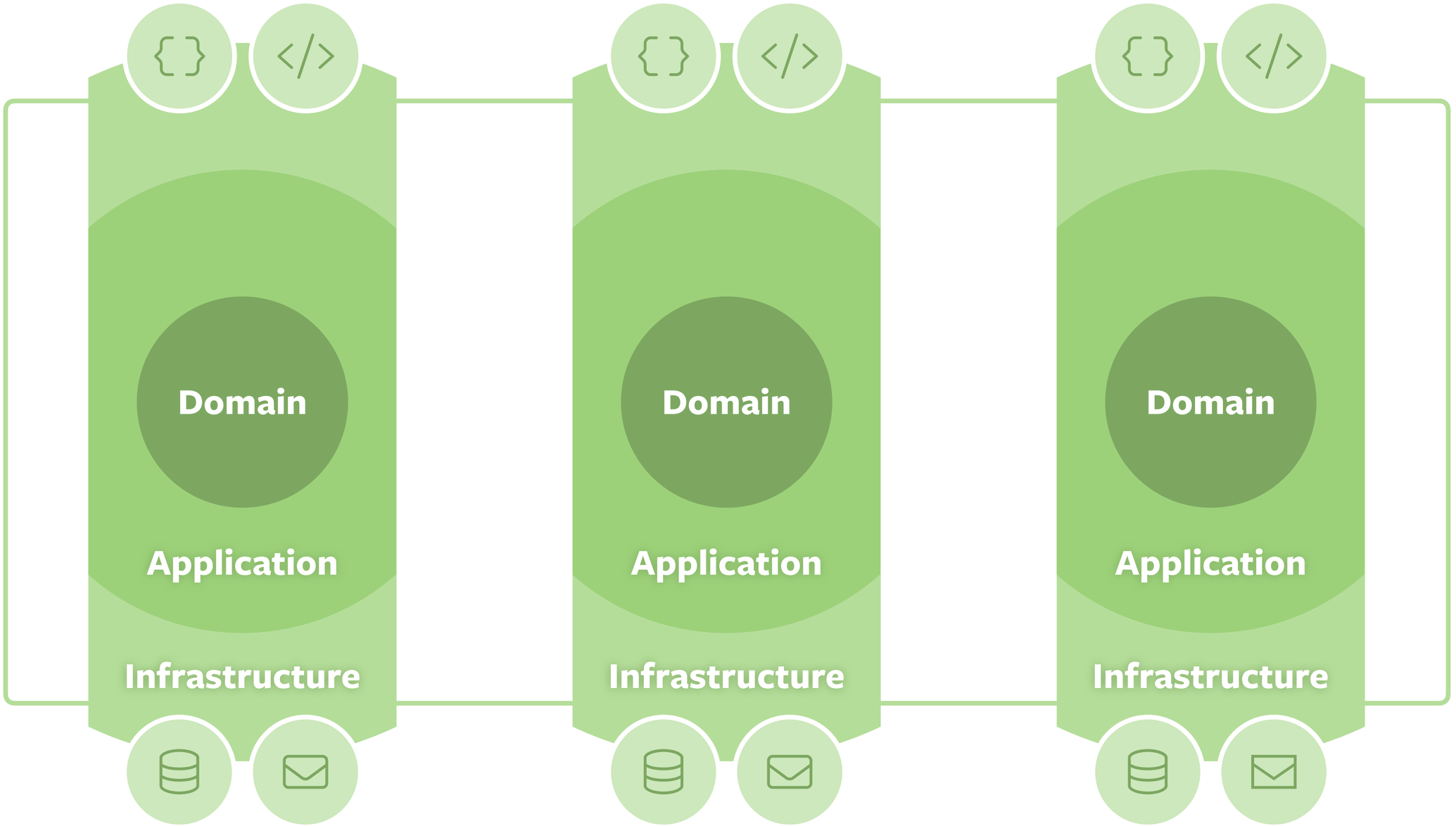

Domain
Application
Infrastructure

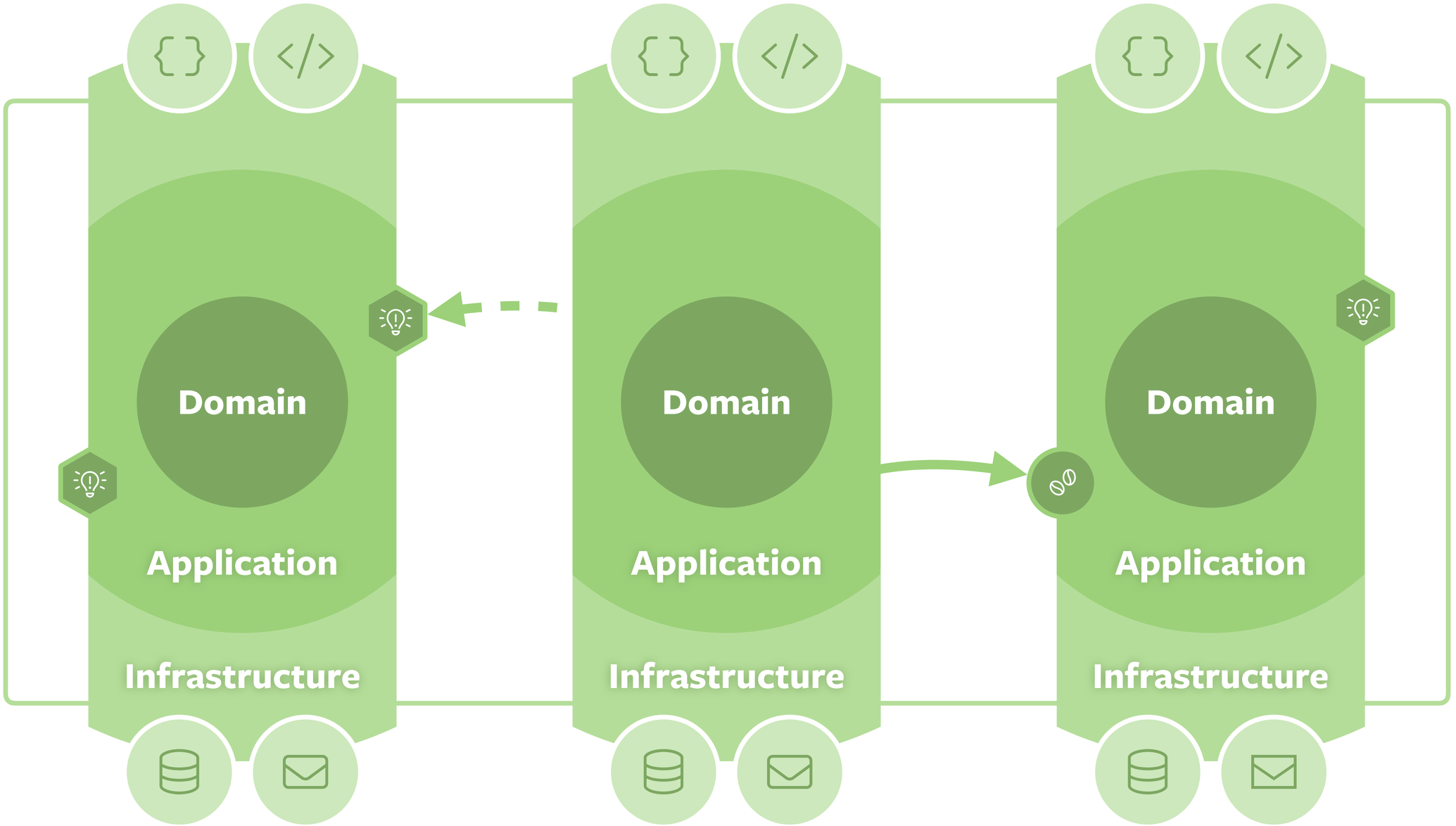

Domain
Application
Infrastructure

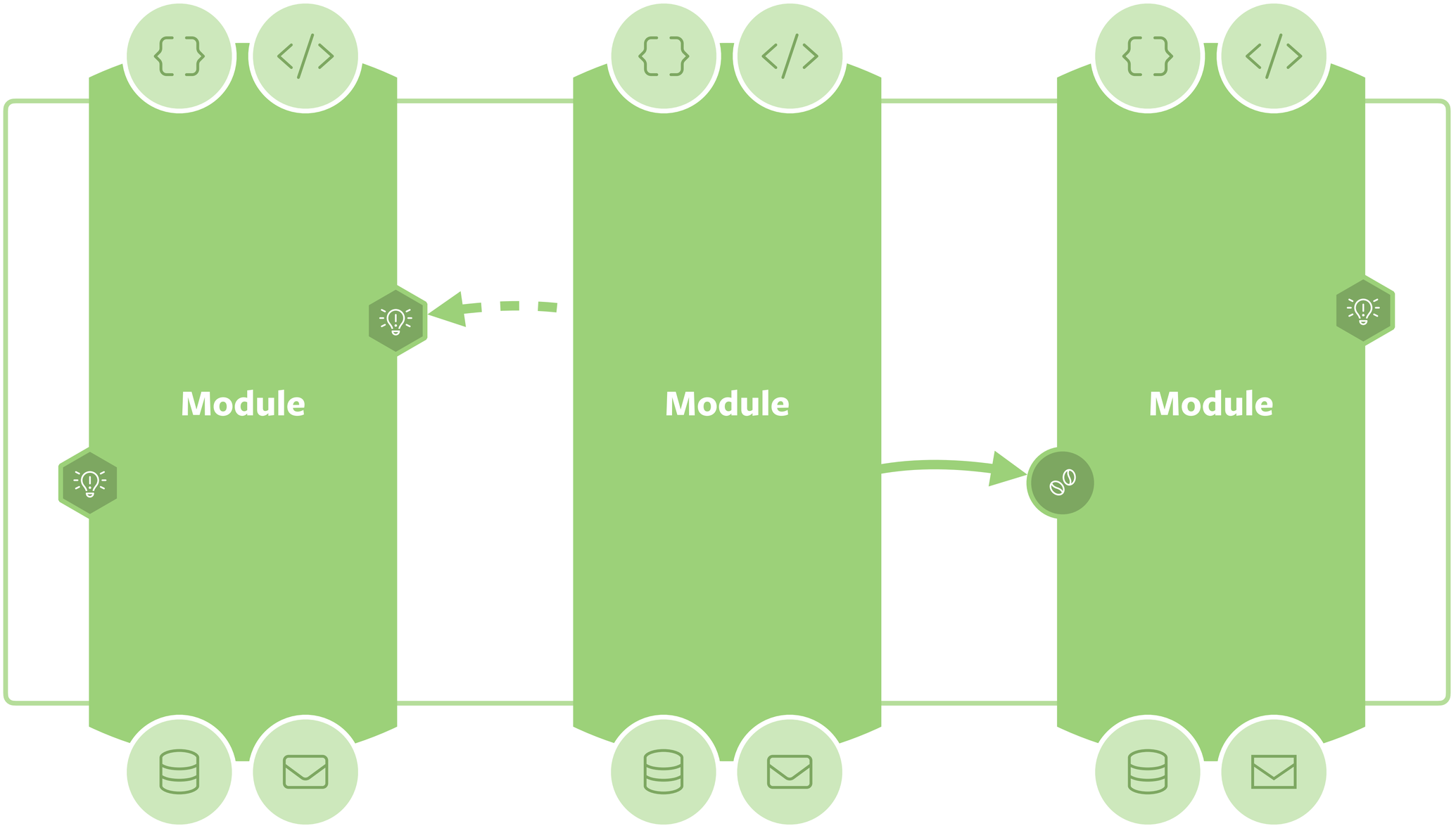

Domain
Application
Infrastructure


Domain
Application
Infrastructure


Domain
Application
Infrastructure


Domain
Application
Infrastructure
Domain
Application
Infrastructure
Domain
Application
Infrastructure


Domain
Application
Infrastructure
Domain
Application
Infrastructure
Domain
Application
Infrastructure



https://odrotbohm.de/2023/07/sliced-onion-architecture

Code structure

Severely inspired by Simon Brown's https://static.simonbrown.je/modular-monoliths.pdf

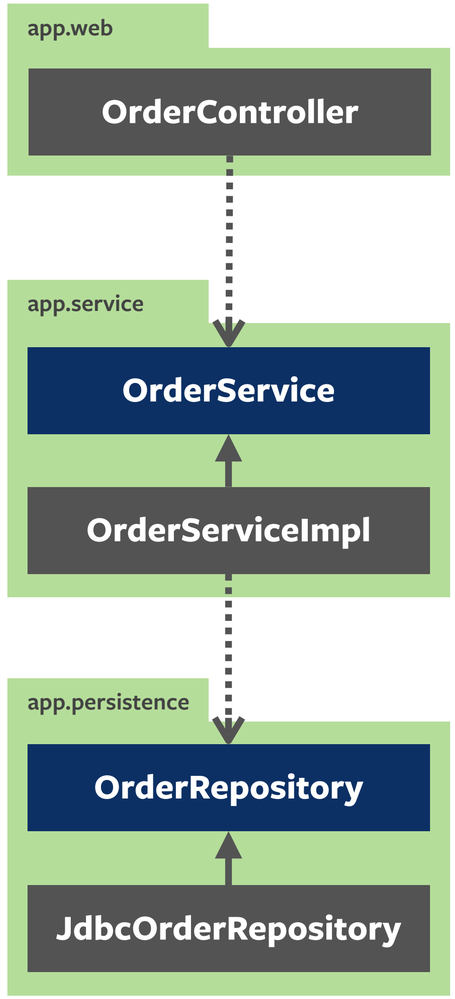


Oliver Drotbohm 🥁 & 👨‍💻
@odrotbohm

Coined Drotbohm's law yesterday: from the structure of a software system you can derive the book the architect read most recently...

Post übersetzen

8:11 vorm. · 7. Juli 2016 aus Stuttgart, Deutschland



Classic Layers

Legend

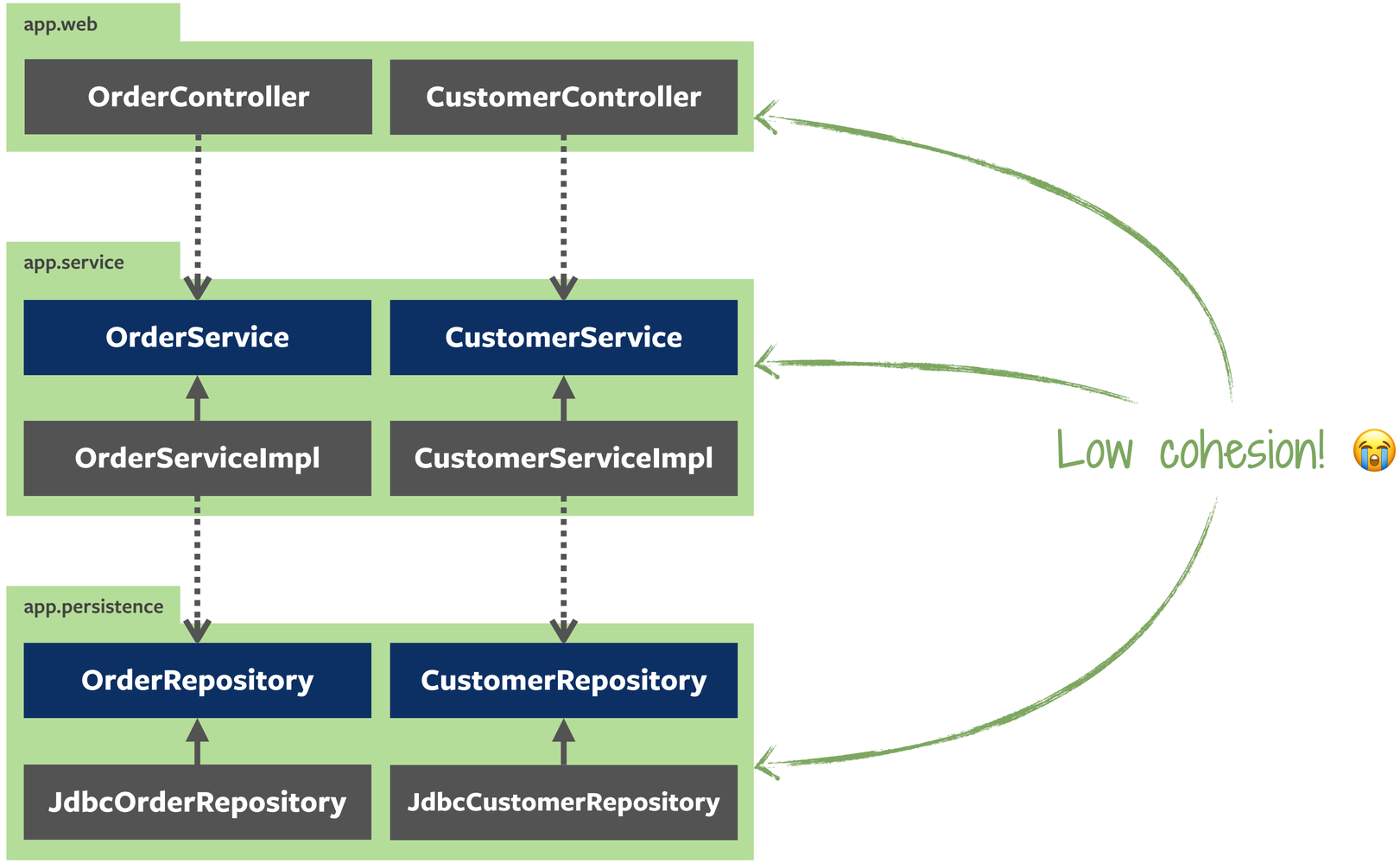
Package / Build Module

Interface

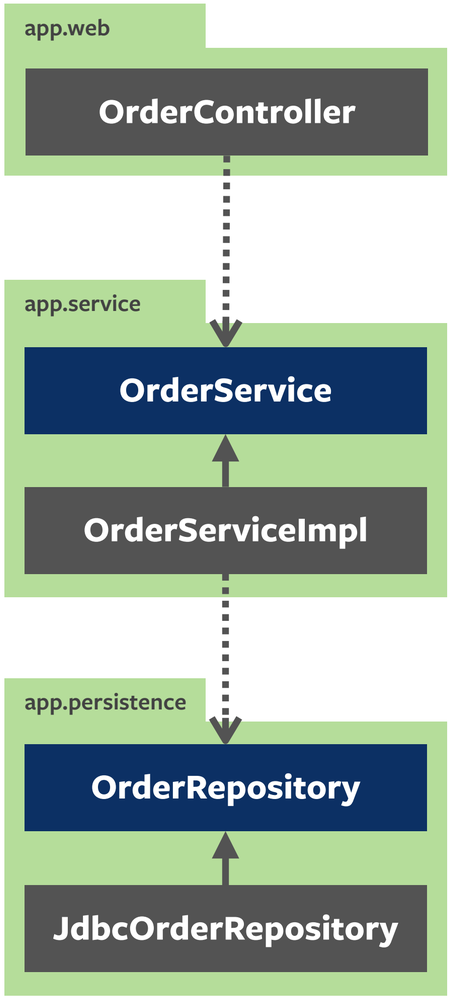
Class

Implements

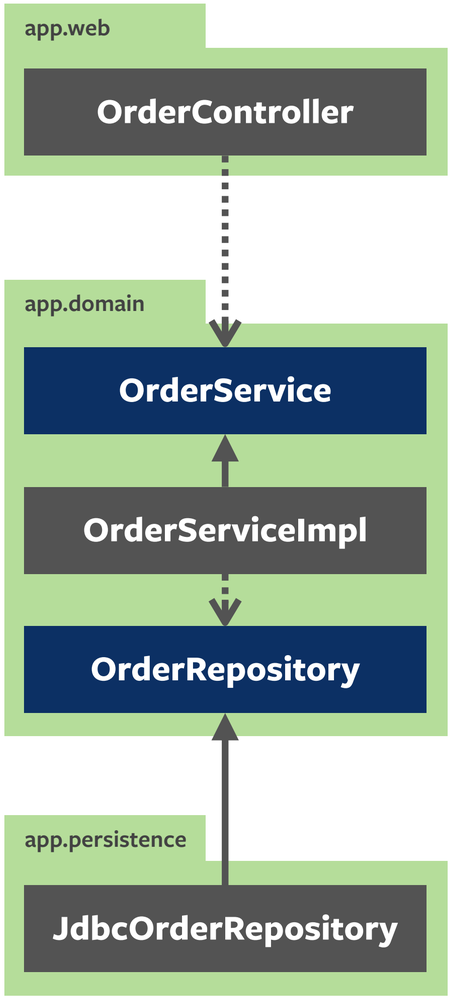
Refers to



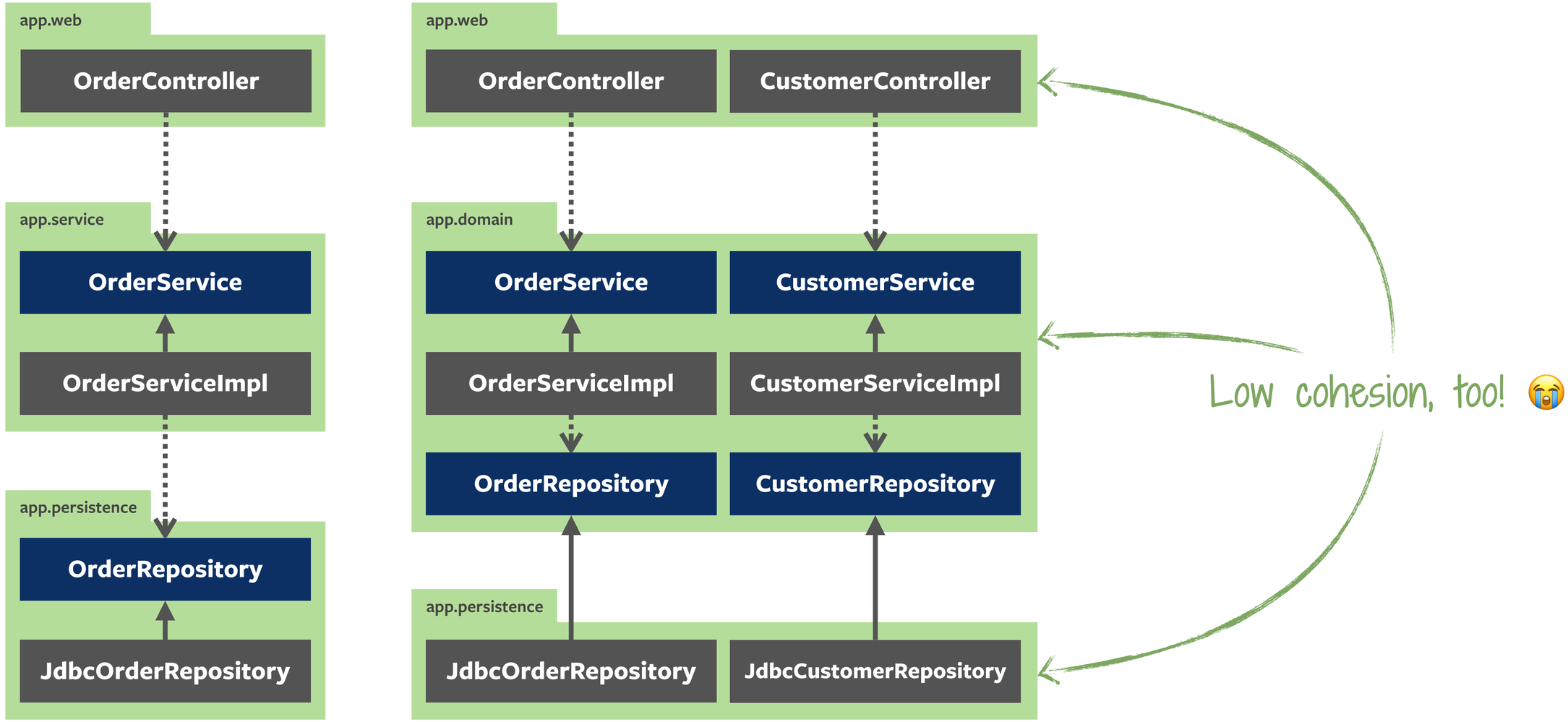
Classic Layers

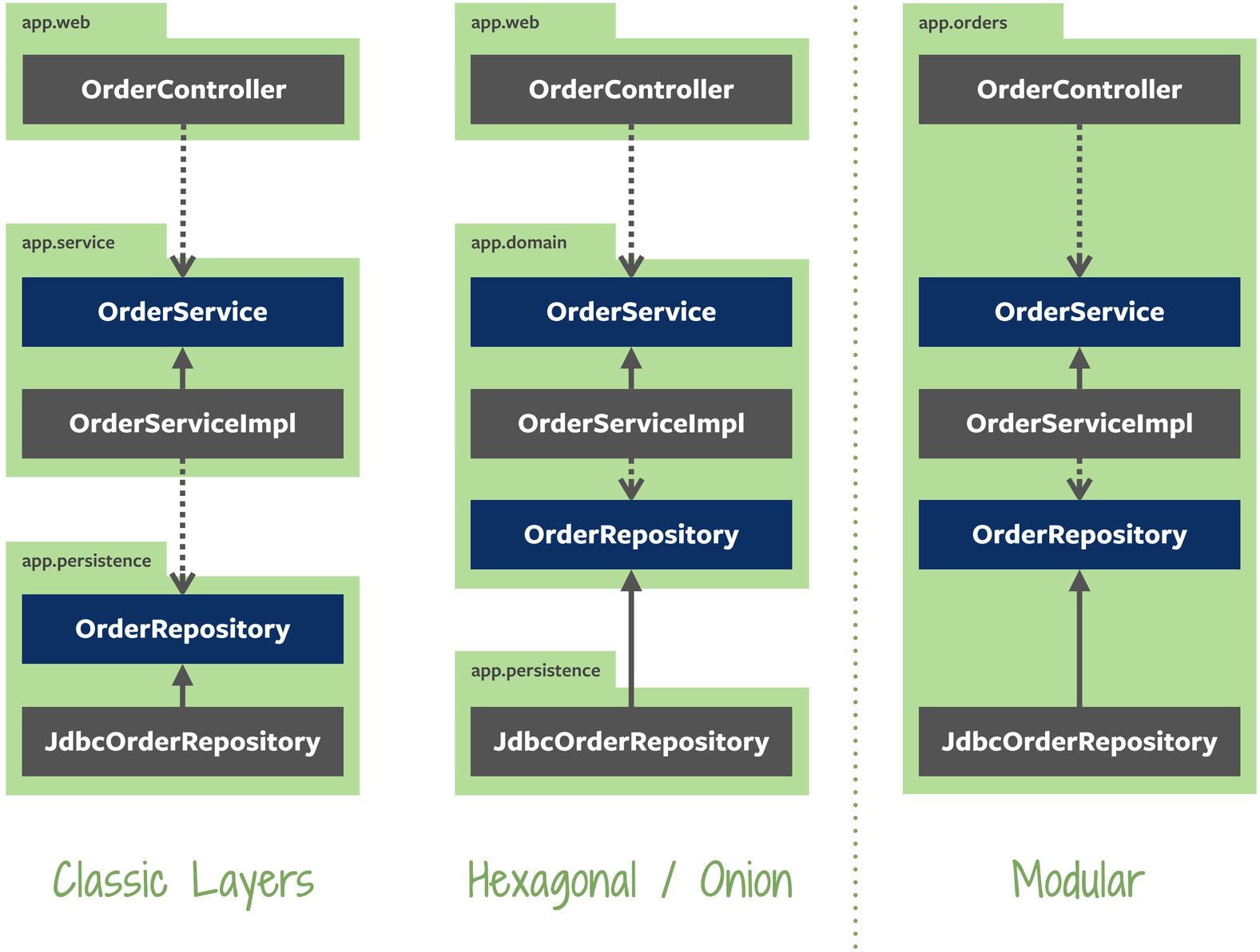


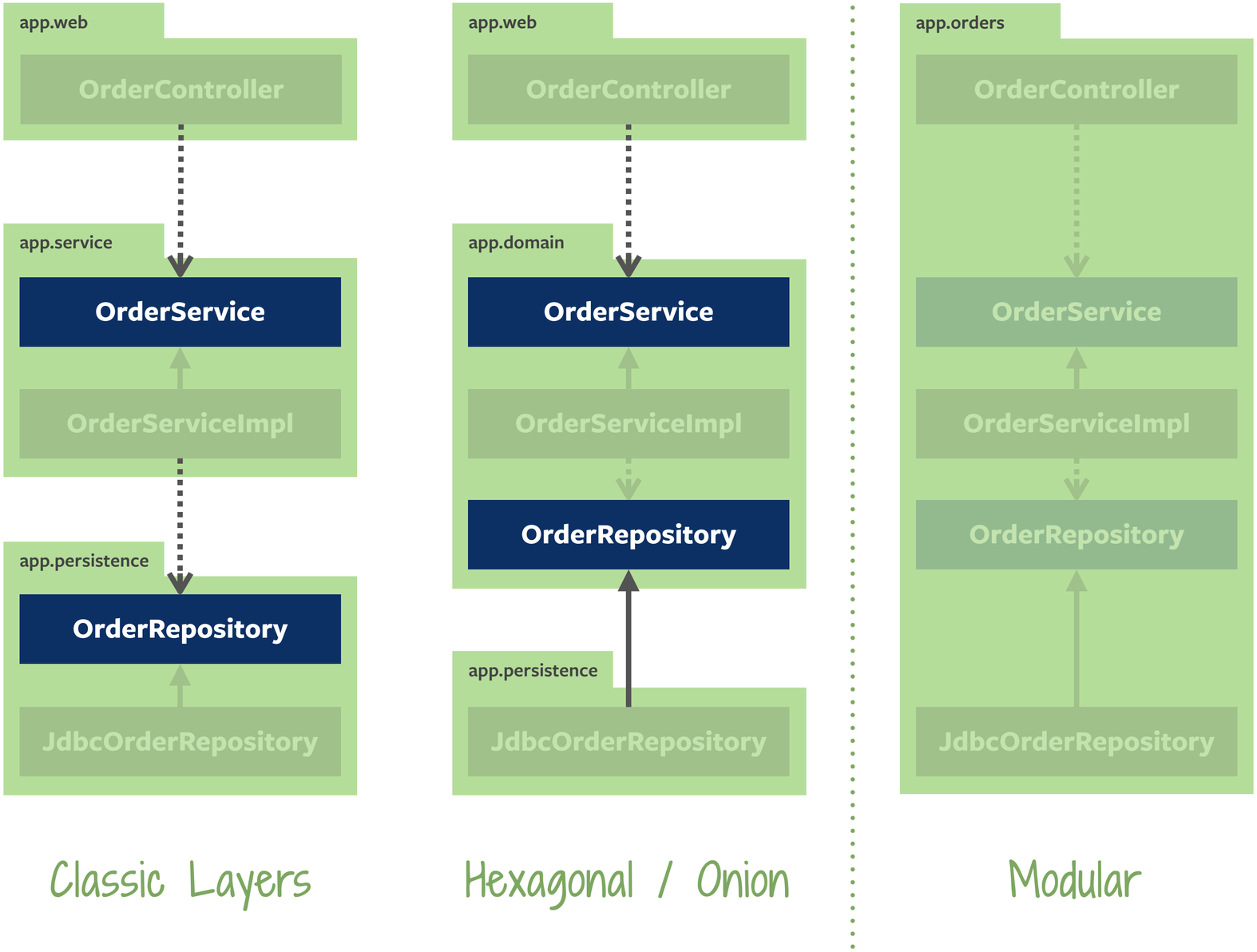
Classic Layers



Hexagonal / Onion


app.web
OrderController
app.service
OrderService
OrderServiceImpl
app.persistence
OrderRepository
JdbcOrderRepository
app.web
OrderController
CustomerController
app.domain
OrderService
CustomerService
OrderServiceImpl
CustomerServiceImpl
OrderRepository
CustomerRepository
app.persistence
JdbcOrderRepository
JdbcCustomerRepository
Low cohesion, too! 😭
Classic Layers
Hexagonal / Onion

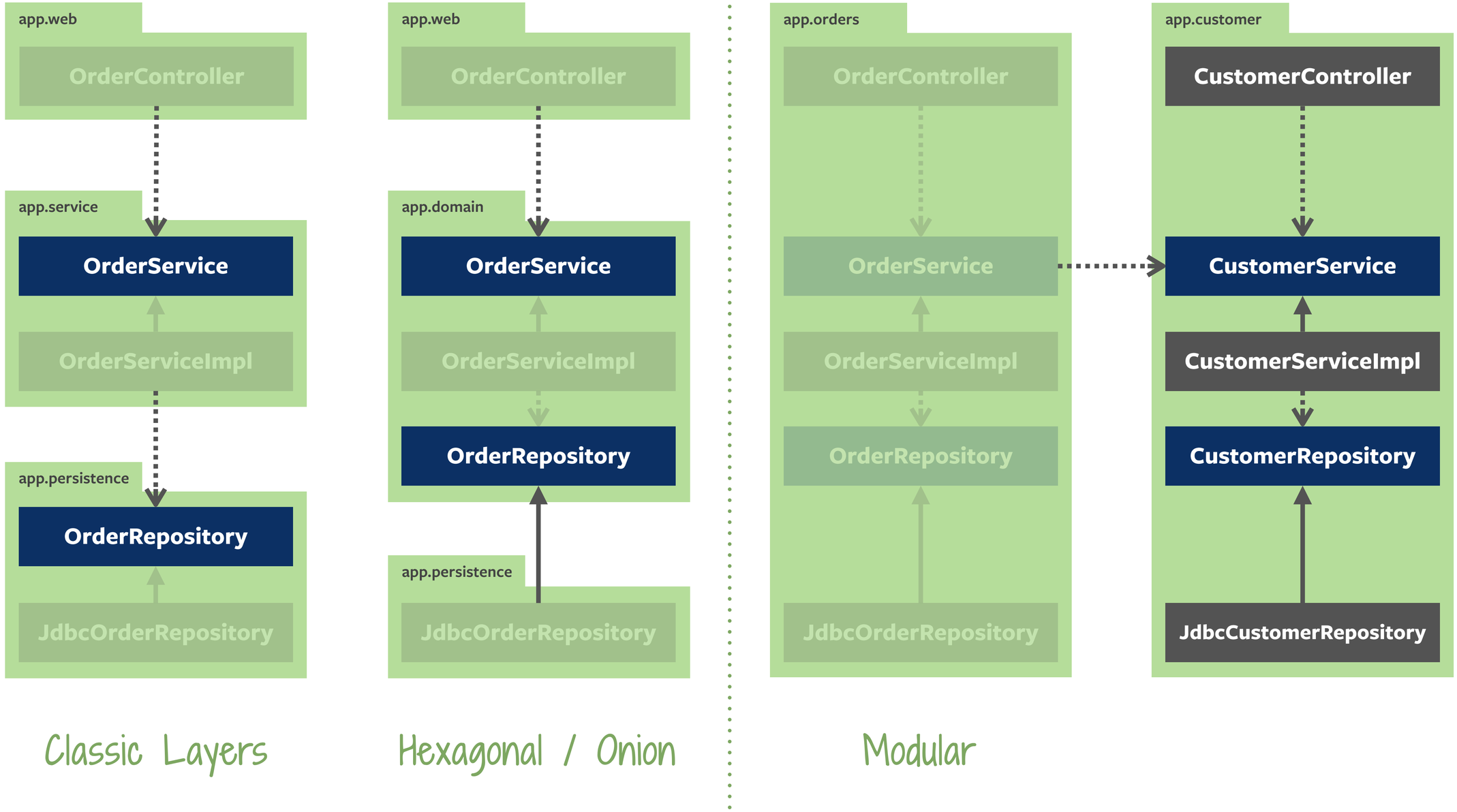

app.web
OrderController
app.service
OrderService
OrderServiceImpl
app.persistence
OrderRepository
JdbcOrderRepository
app.web
OrderController
app.domain
OrderService
OrderServiceImpl
OrderRepository
app.persistence
JdbcOrderRepository
app.orders
OrderController
OrderService
OrderServiceImpl
OrderRepository
JdbcOrderRepository
Classic Layers
Hexagonal / Onion
Modular

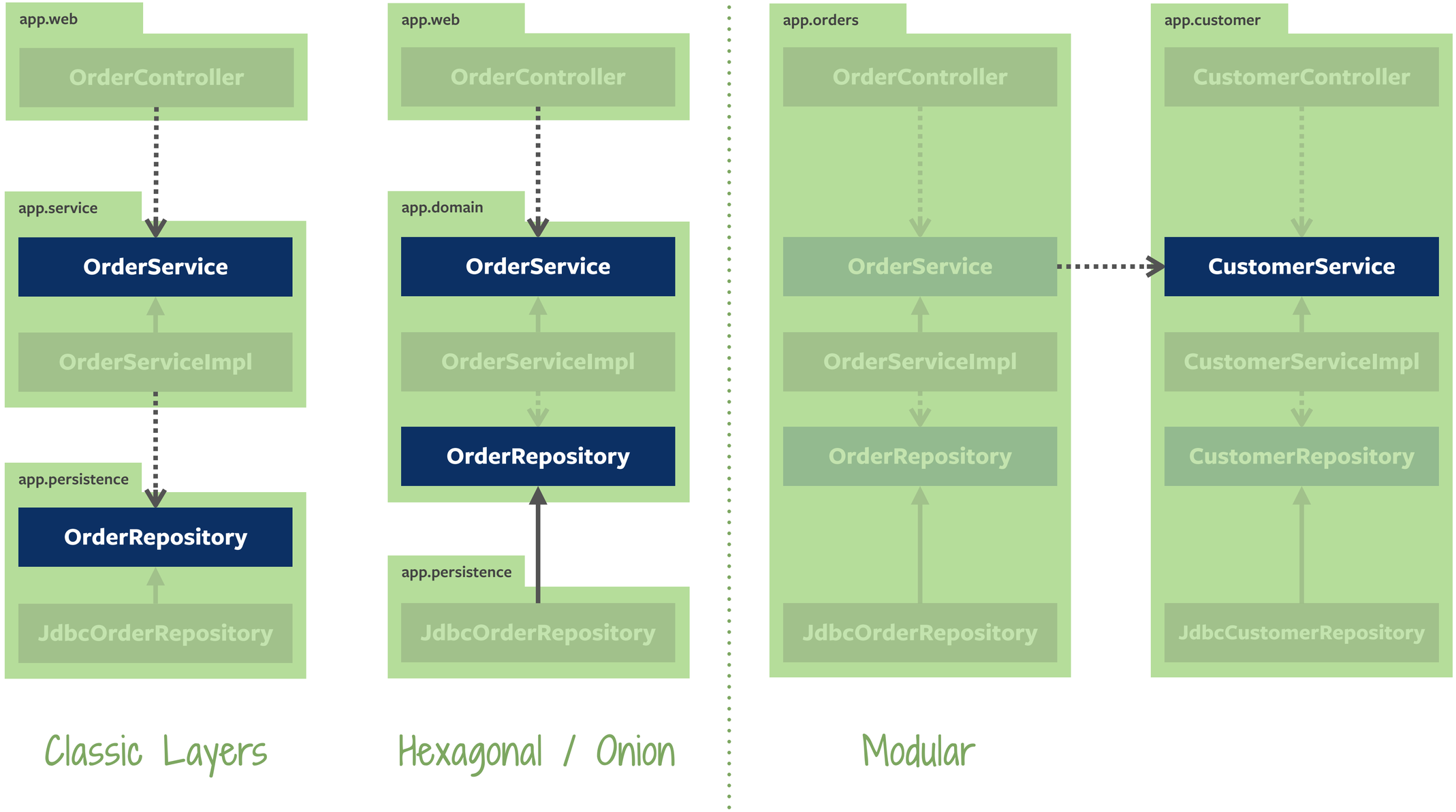


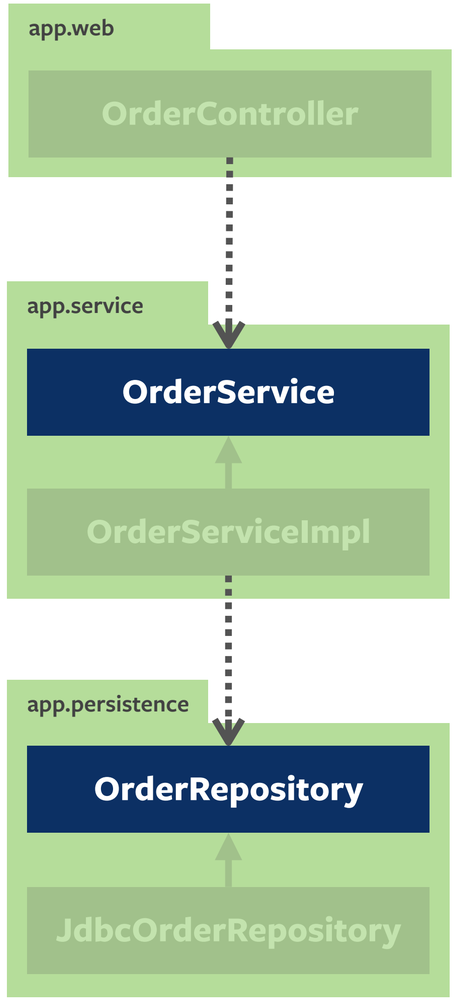
Classic Layers

Hexagonal / Onion

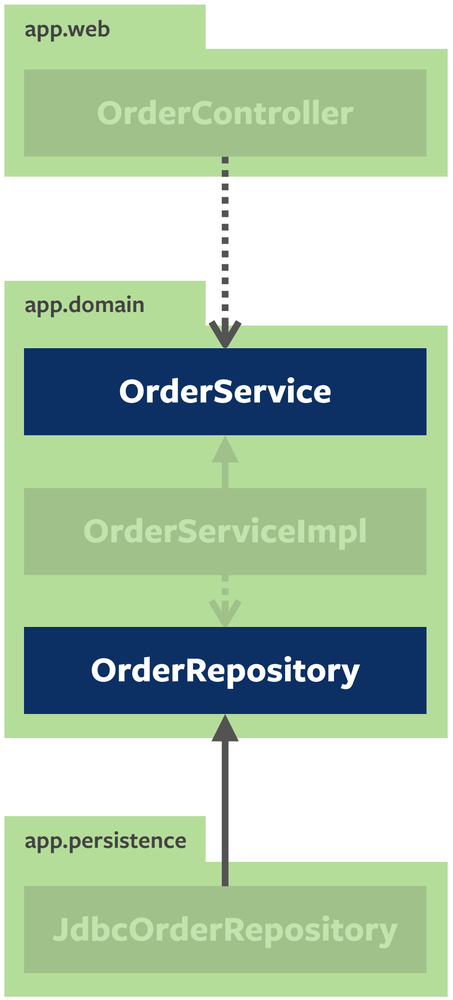
Modular


app.web
OrderController
app.service
OrderService
OrderServiceImpl
app.persistence
OrderRepository
JdbcOrderRepository
app.web
OrderController
app.domain
OrderService
OrderServiceImpl
OrderRepository
app.persistence
JdbcOrderRepository
app.orders
OrderController
OrderService
OrderServiceImpl
OrderRepository
JdbcOrderRepository
app.customer
CustomerController
CustomerService
CustomerServiceImpl
CustomerRepository
JdbcCustomerRepository
Classic Layers
Hexagonal / Onion
Modular

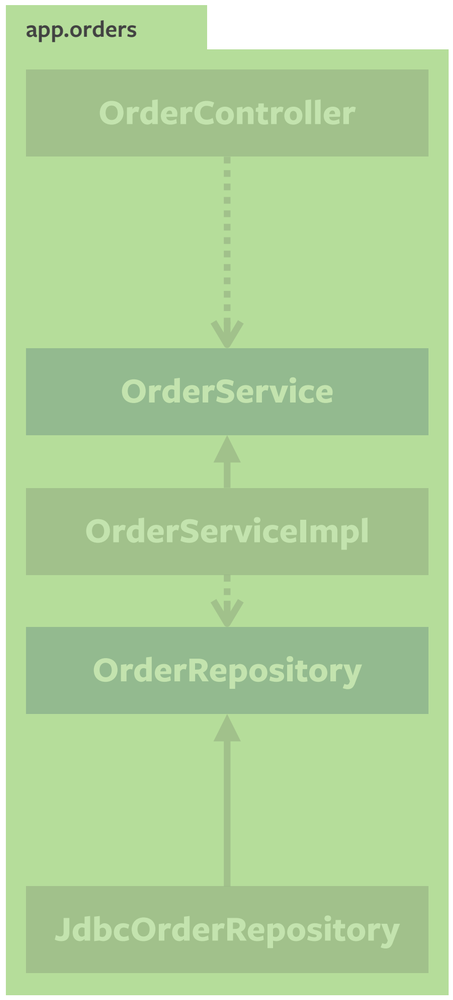

app.web
OrderController
app.service
OrderService
OrderServiceImpl
app.persistence
OrderRepository
JdbcOrderRepository
app.web
OrderController
app.domain
OrderService
OrderServiceImpl
OrderRepository
app.persistence
JdbcOrderRepository
app.orders
OrderController
OrderService
OrderServiceImpl
OrderRepository
JdbcOrderRepository
app.customer
CustomerController
CustomerService
CustomerServiceImpl
CustomerRepository
JdbcCustomerRepository
Classic Layers
Hexagonal / Onion
Modular



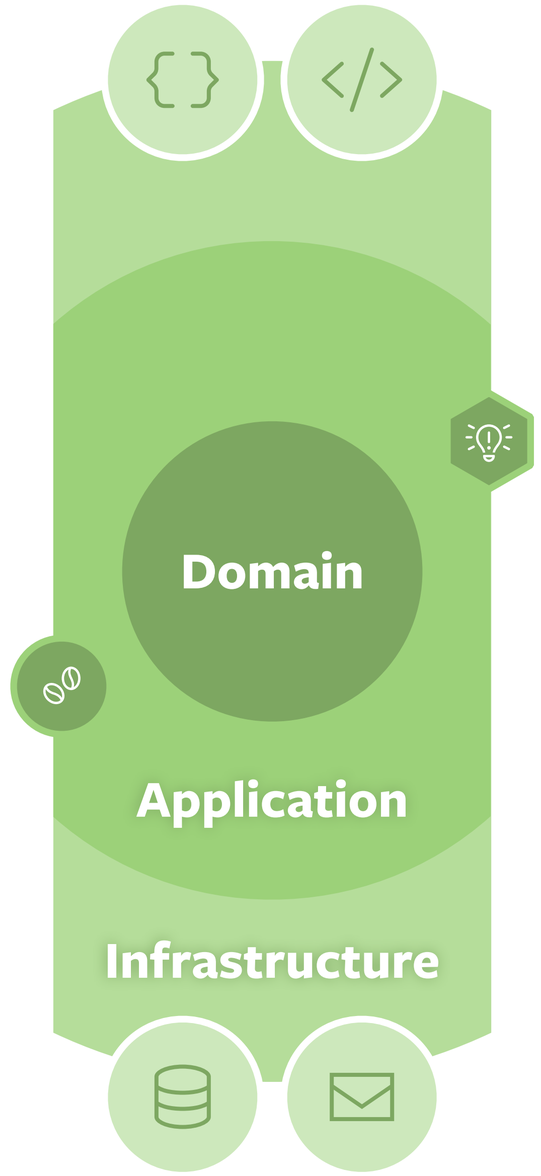
Classic Layers

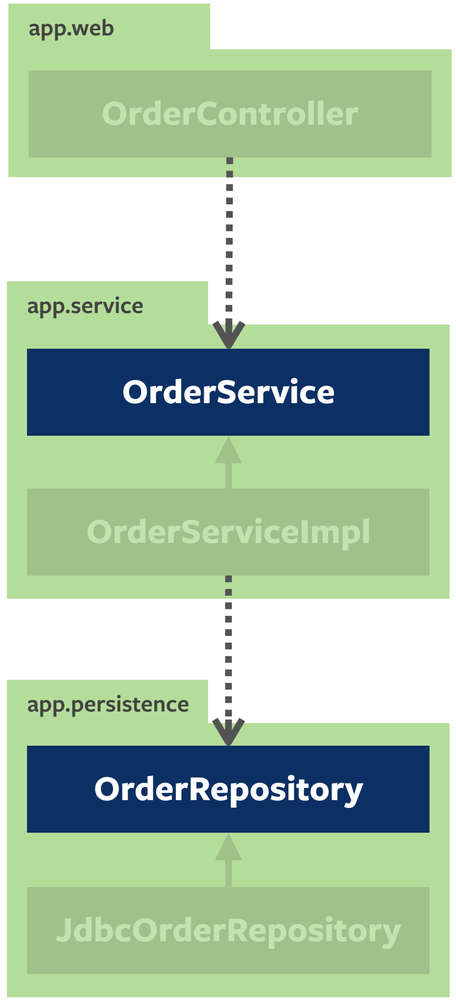


Hexagonal / Onion

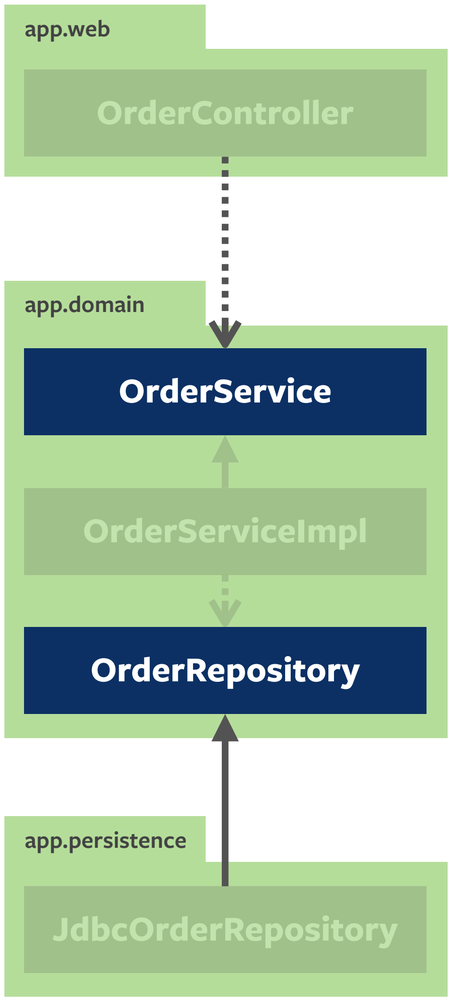


Modular

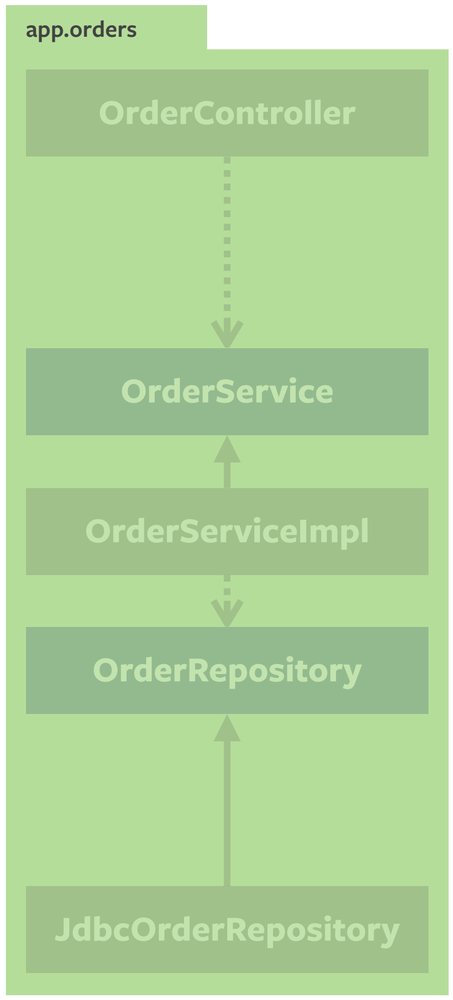




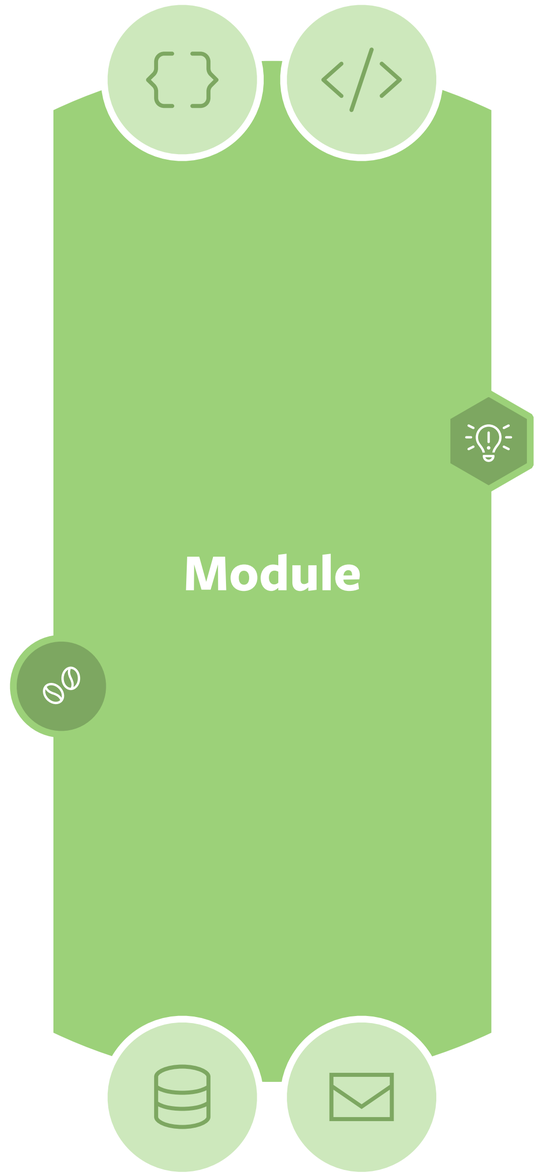
Classic Layers

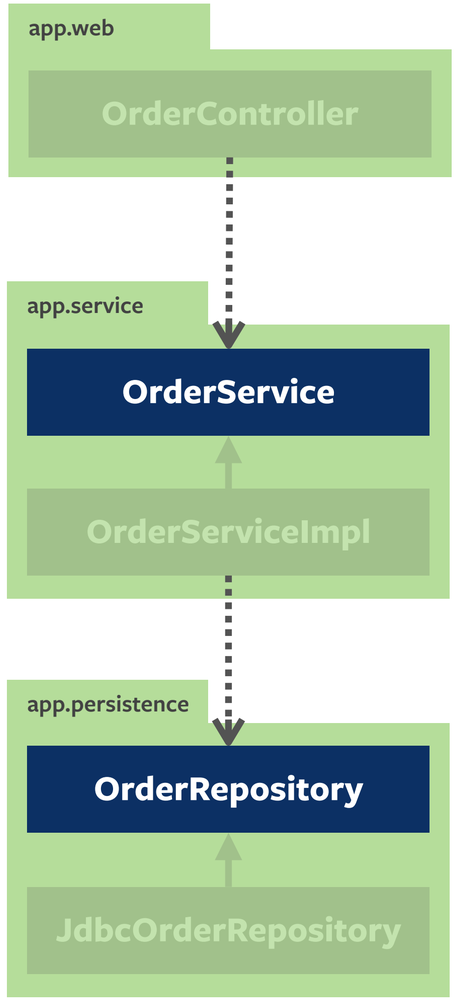


Hexagonal / Onion

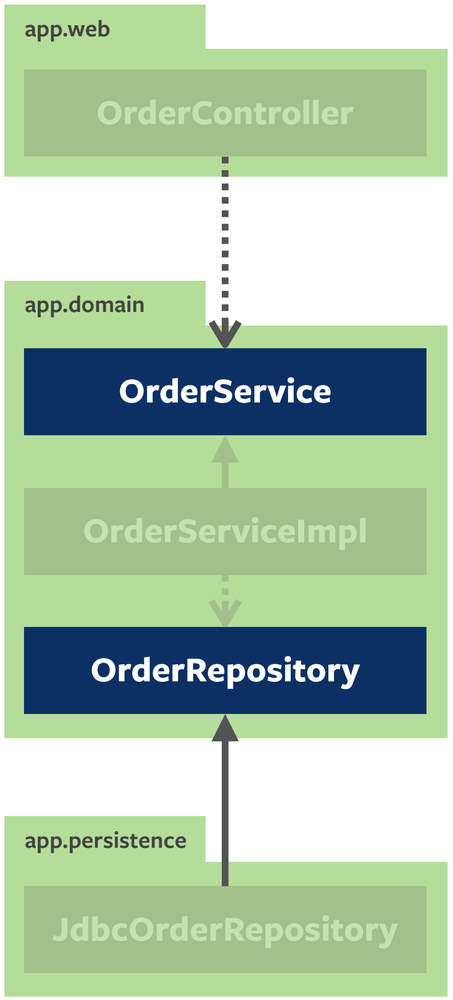


Modular

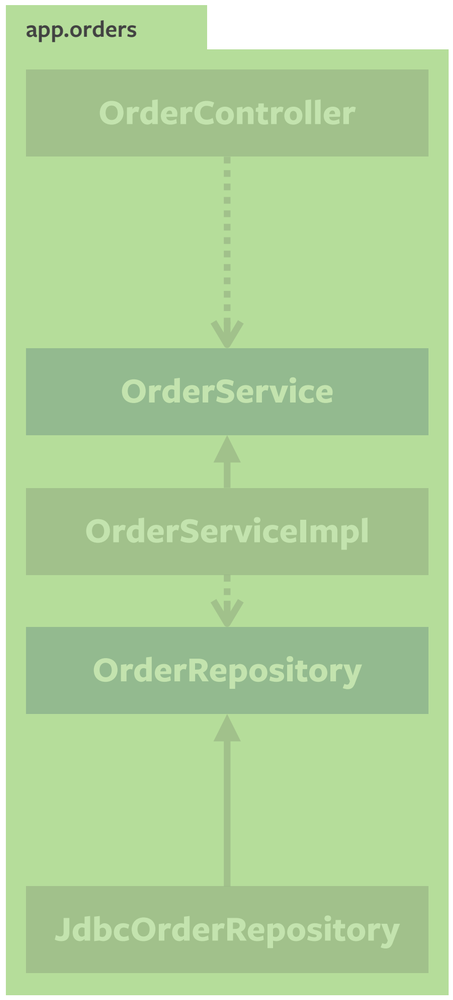




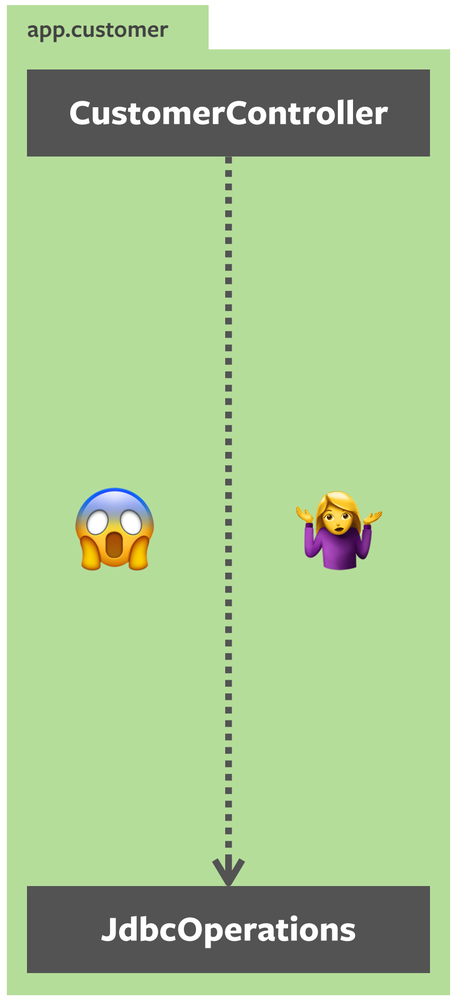
Classic Layers



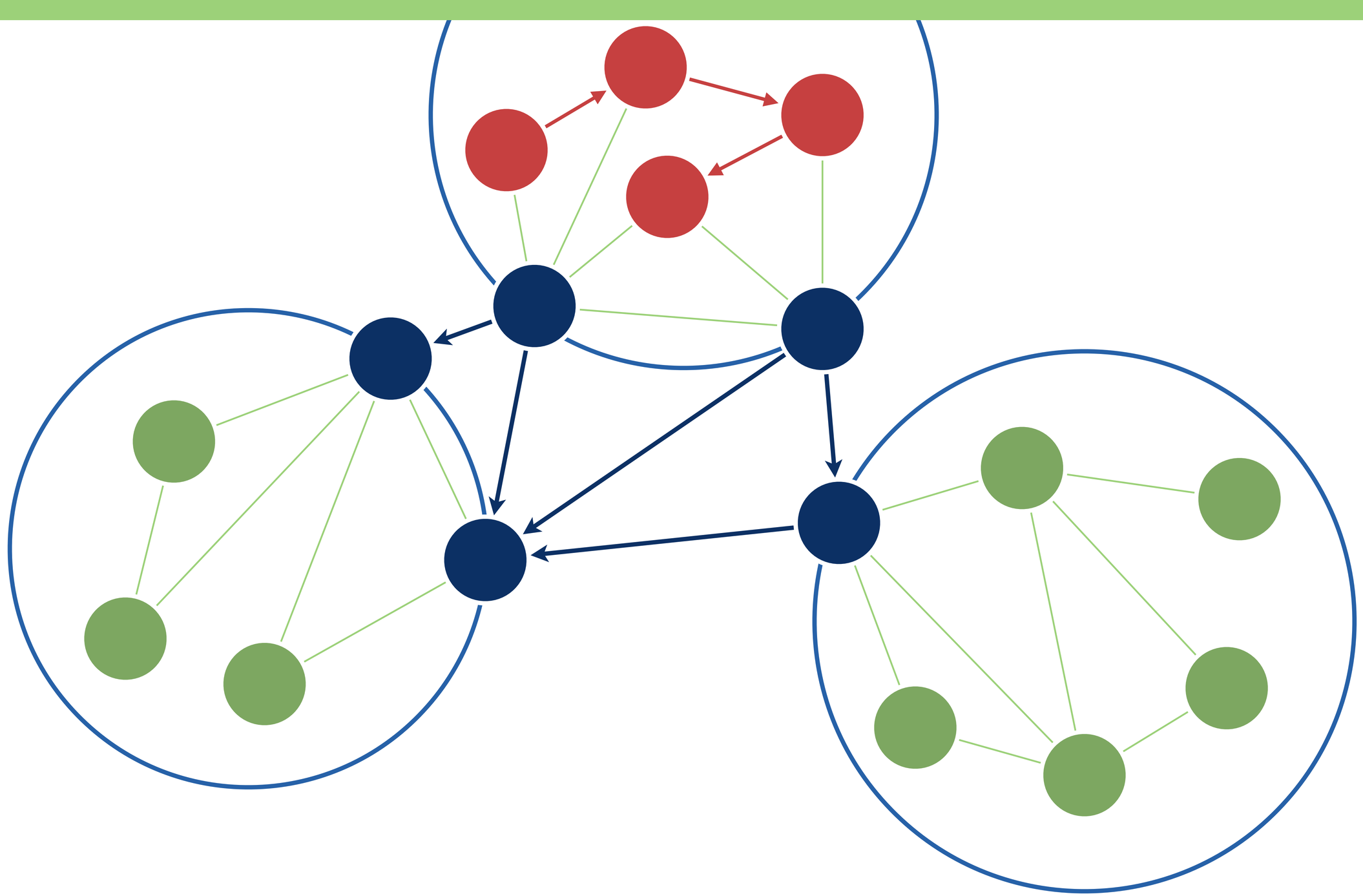
Hexagonal / Onion



Modular



Avoid internal complexity
if domain permits





Oliver Drotbohm
@odrotbohm@chaos.social

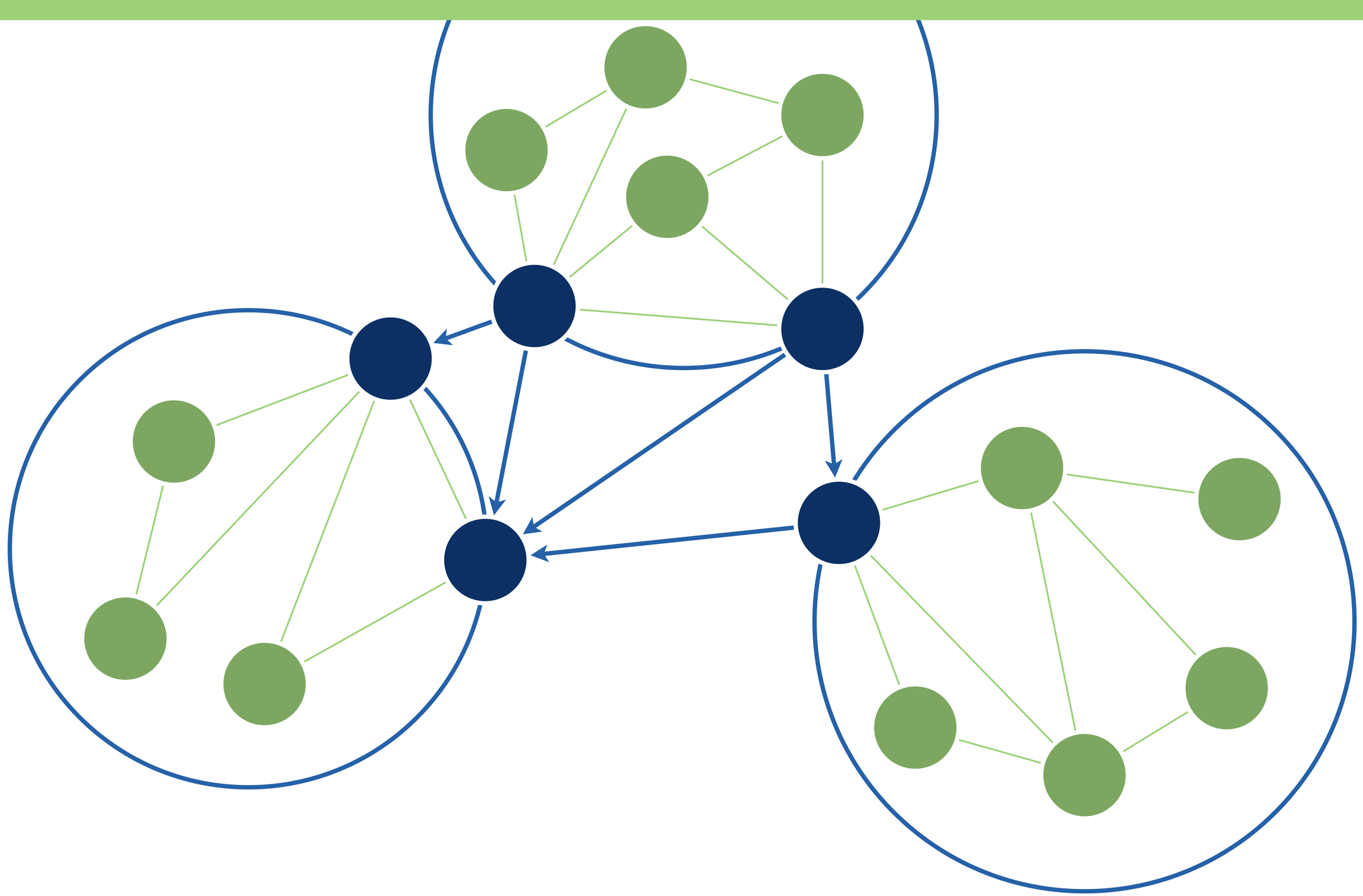
The primary objective of system decomposition is to support change. To contain the effects of a change to as few as possible elements. If your decomposition strategy does not align with that, it's distracting at best and harmful at worst.

24. Sept. 2024, 19:16 · · Ivory for iOS

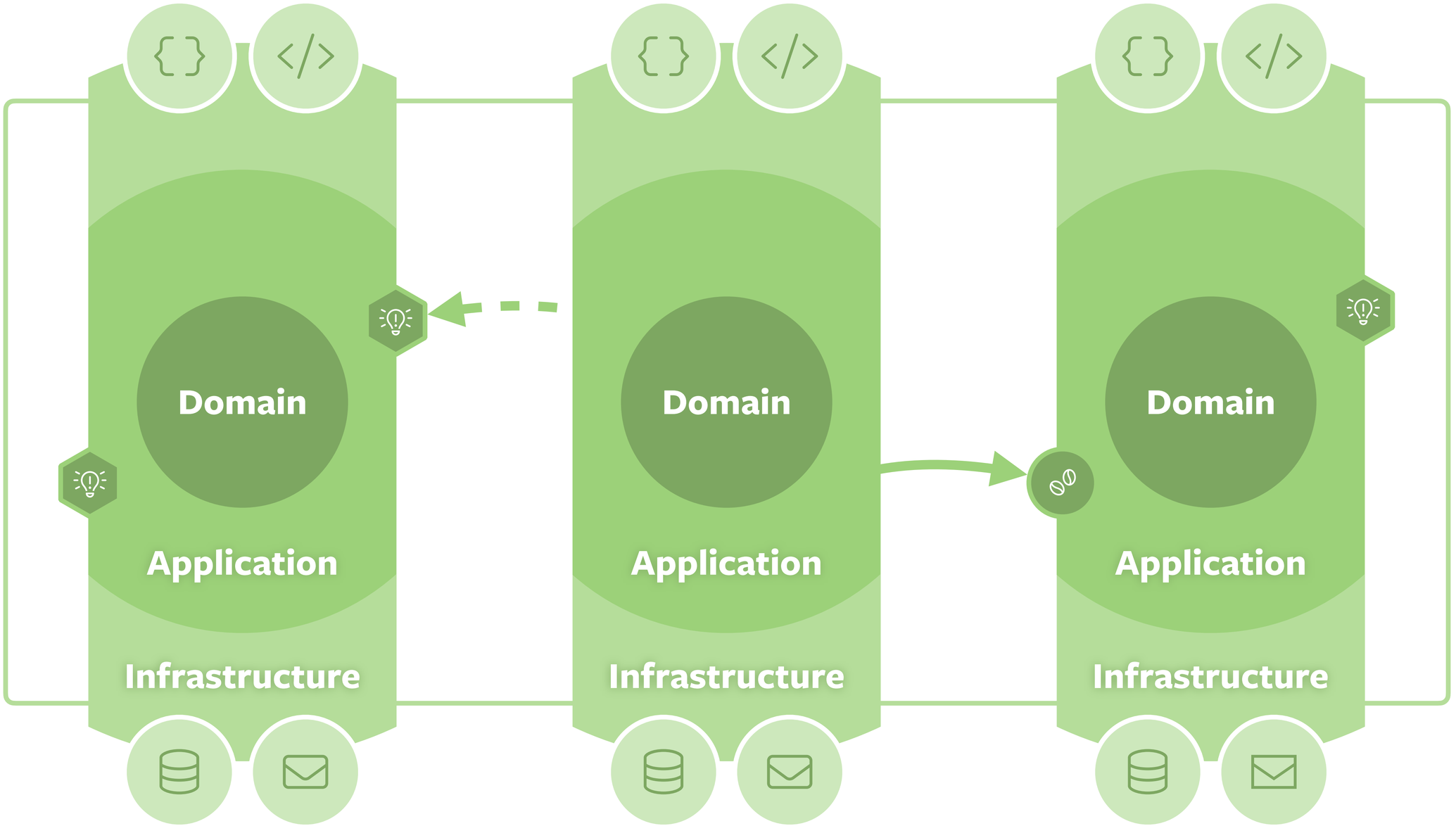
https://chaos.social/@odrotbohm/113193662068876374

Strategic considerations

Designing for cohesion leads to decoupling, but not vice versa.



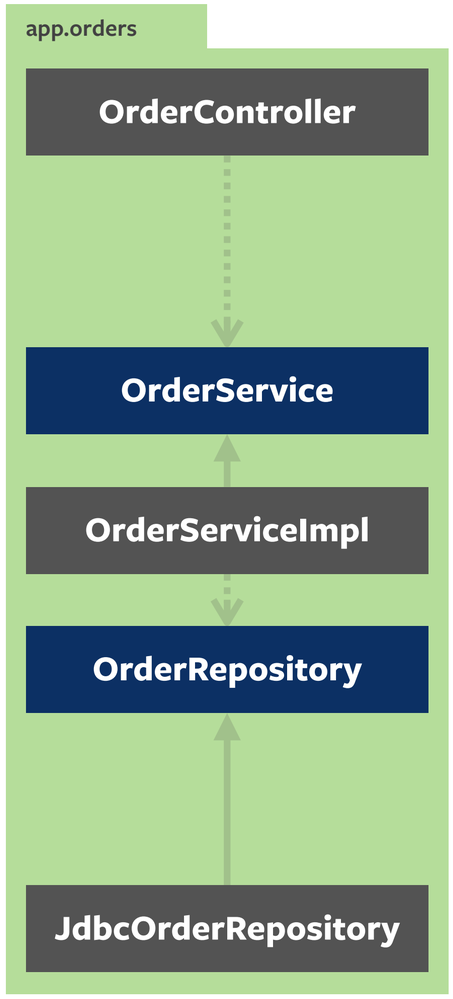
Functional Decomposition over Technical Decomposition


Domain
Application
Infrastructure
Domain
Application
Infrastructure
Domain
Application
Infrastructure

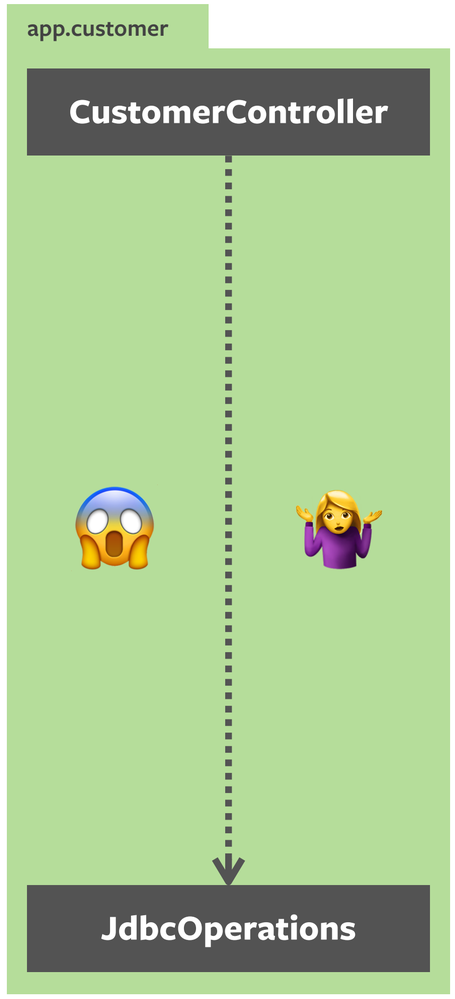
Intrinsic Complexity determines Accidental Complexity

Classic Layers

Hexagonal / Onion



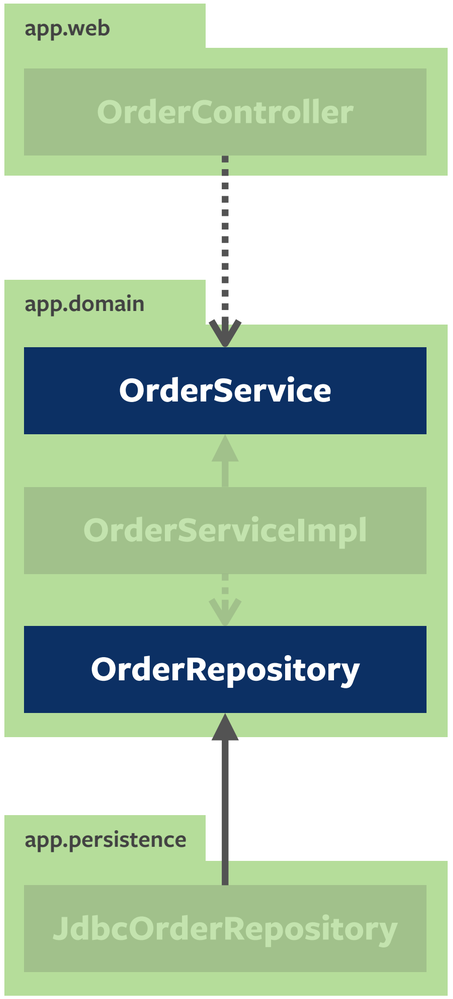
Modular



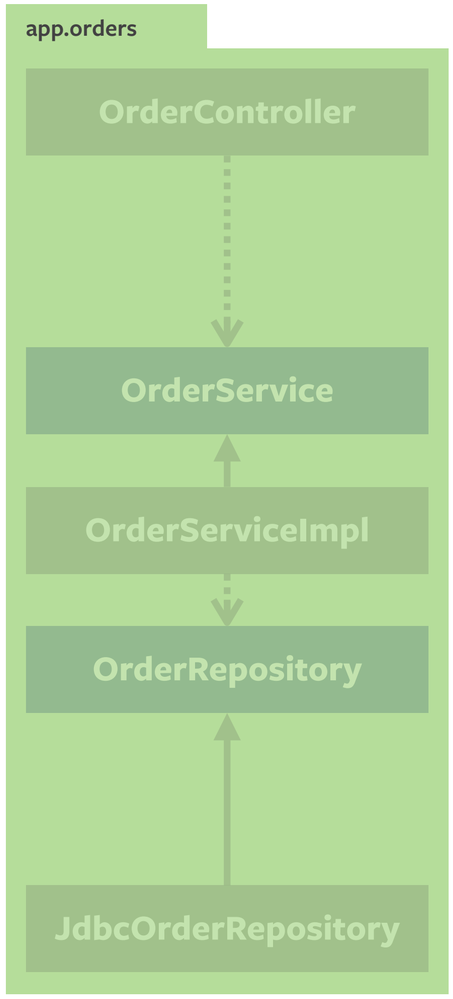
Avoid internal complexity
if domain permits

Encapsulation over Organisation





Hexagonal / Onion



Modular

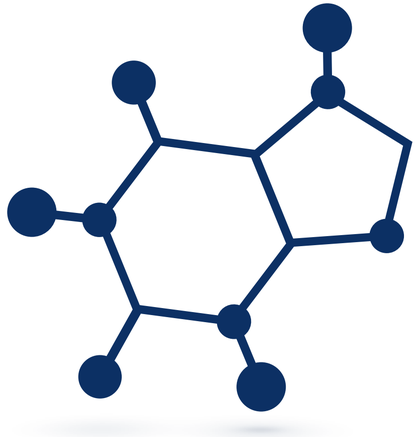


One more thing…



Spring Tools





jMolecules

Thank you!



Oliver Drotbohm odrotbohm info@odrotbohm.de