

Breaking the Monolith

Stefan Tilkov | @stilkov | innoQ

WARNING



**Boxes & Lines
Wild Handwaving**

Enterprisey Stuff

Today's Topic:

Systems

System boundaries

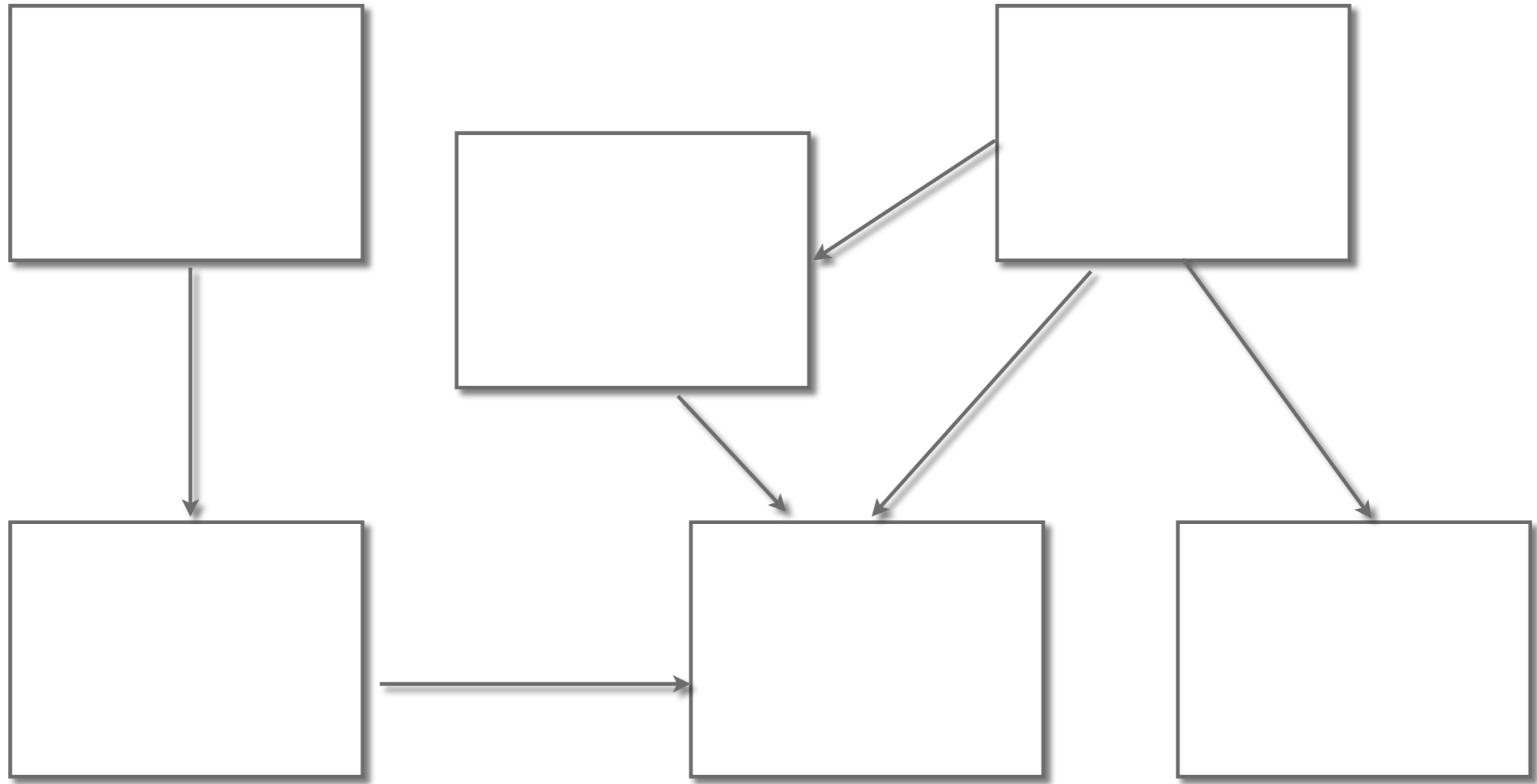
What's a *system*?

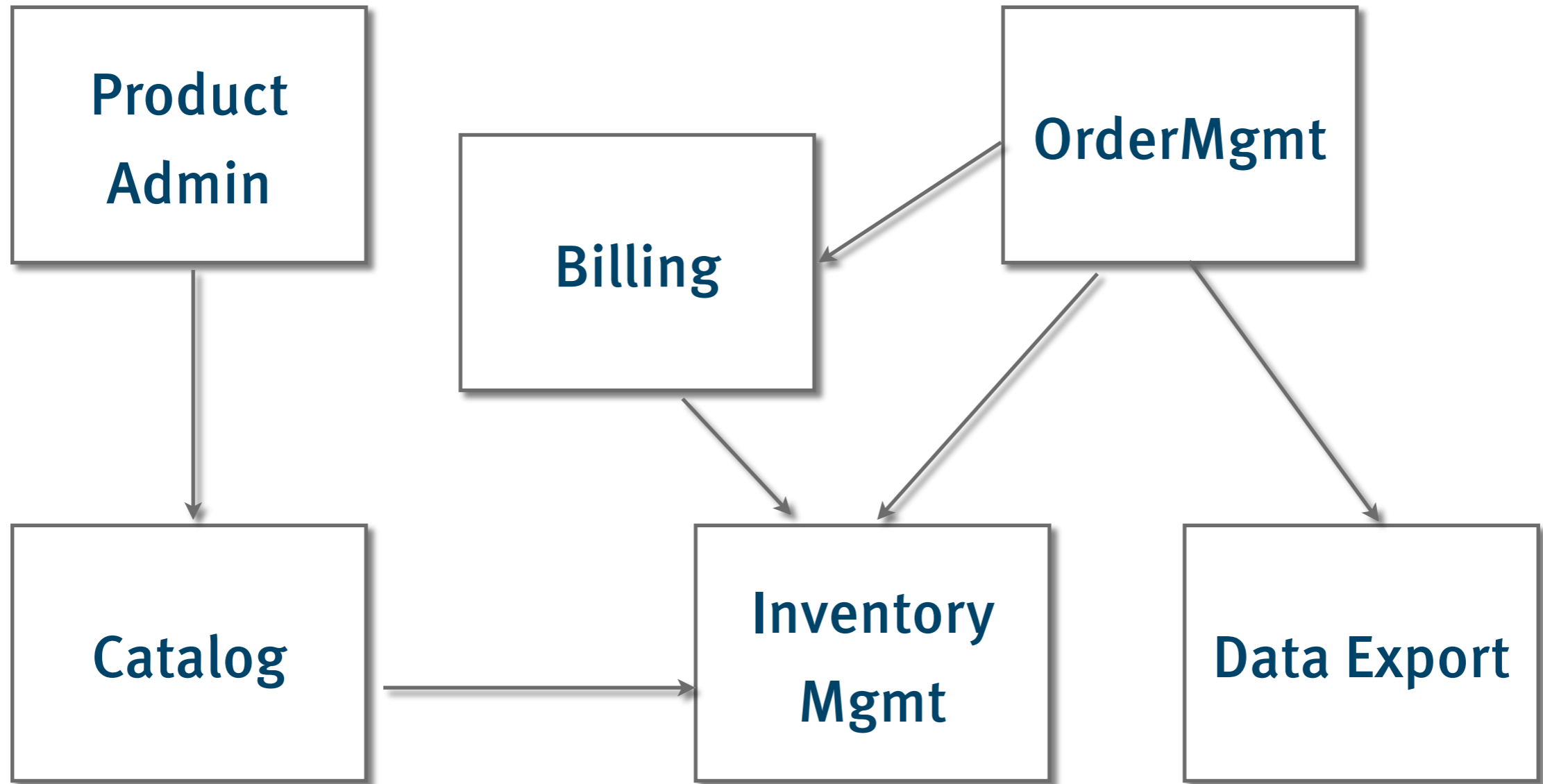
**Task: *Draw a high-level
architecture diagram!***

UI

Logic

Persistence





Lots of things ...

Modules
Components
Subsystems
Libraries
Classes
...

... all in one system

How do we come up with system boundaries?

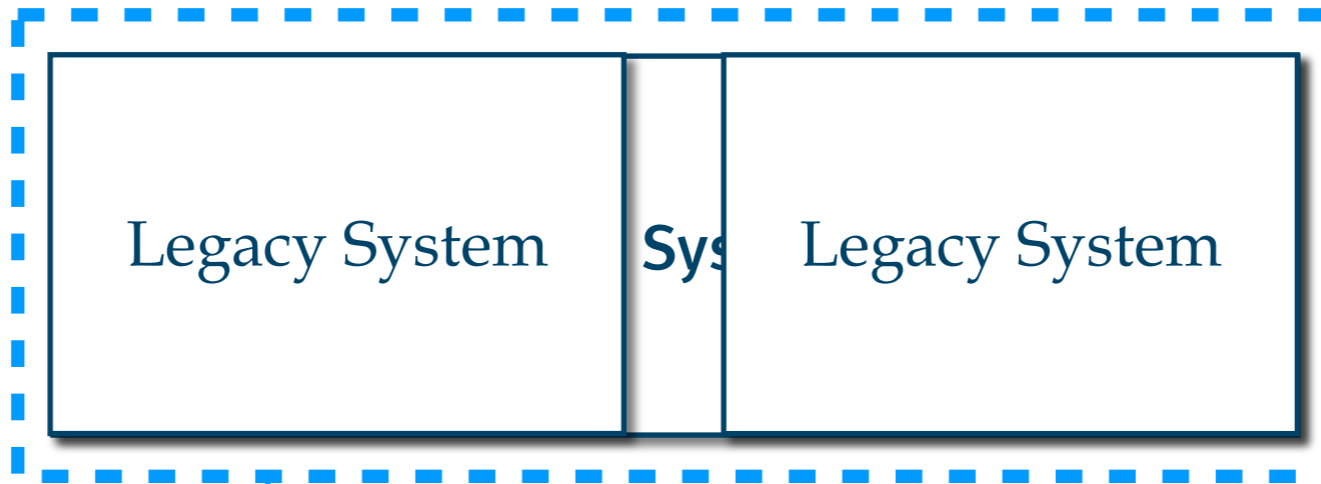
Modernization



Legacy System

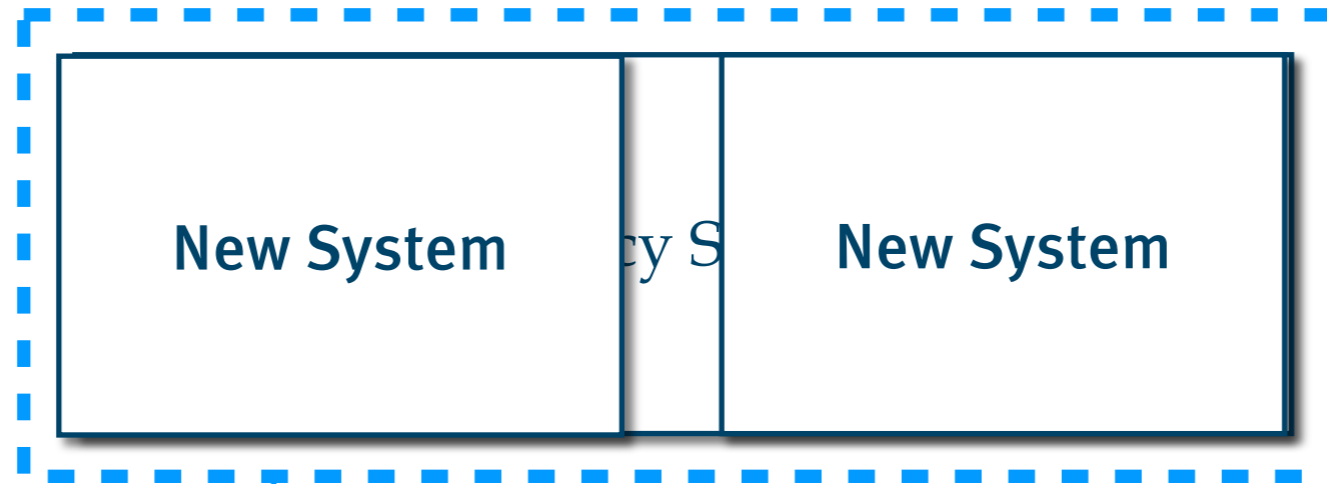
Project scope

Consolidation



Project scope

Modularization



Project scope

1 Project = 1 System?

Modularization & Size

Size	Modularization
1-50 LOC	single file
50-500 LOC	few files, few functions
500-1000 LOC	Library, class hierarchy
1000-2000 LOC	Framework + application
>2000 LOC	multiple applications

System characteristics

Separate (redundant) persistence

Internal, separate logic

Domain models & implementation strategies

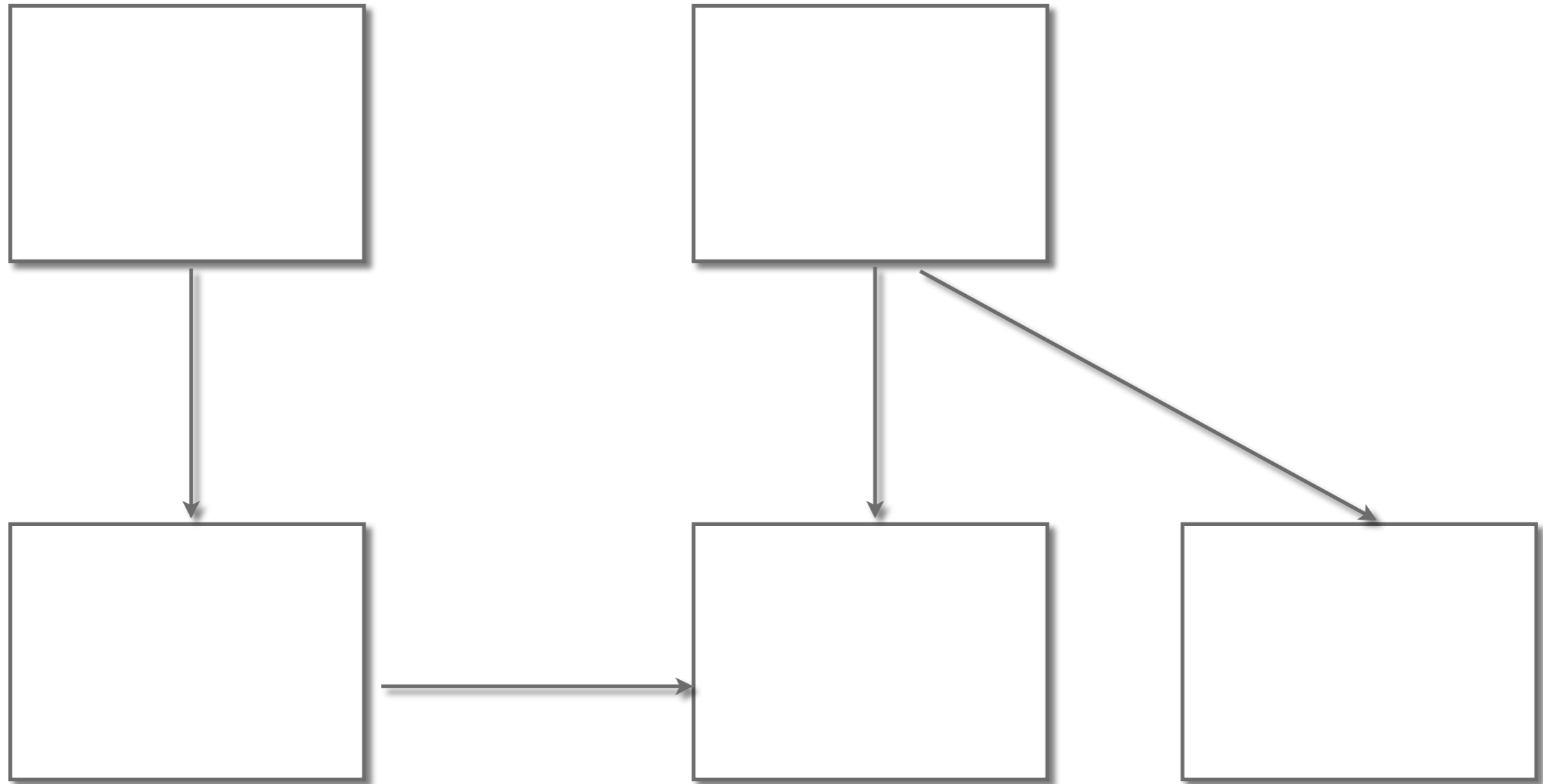
Separate UI

Separate development & evolution

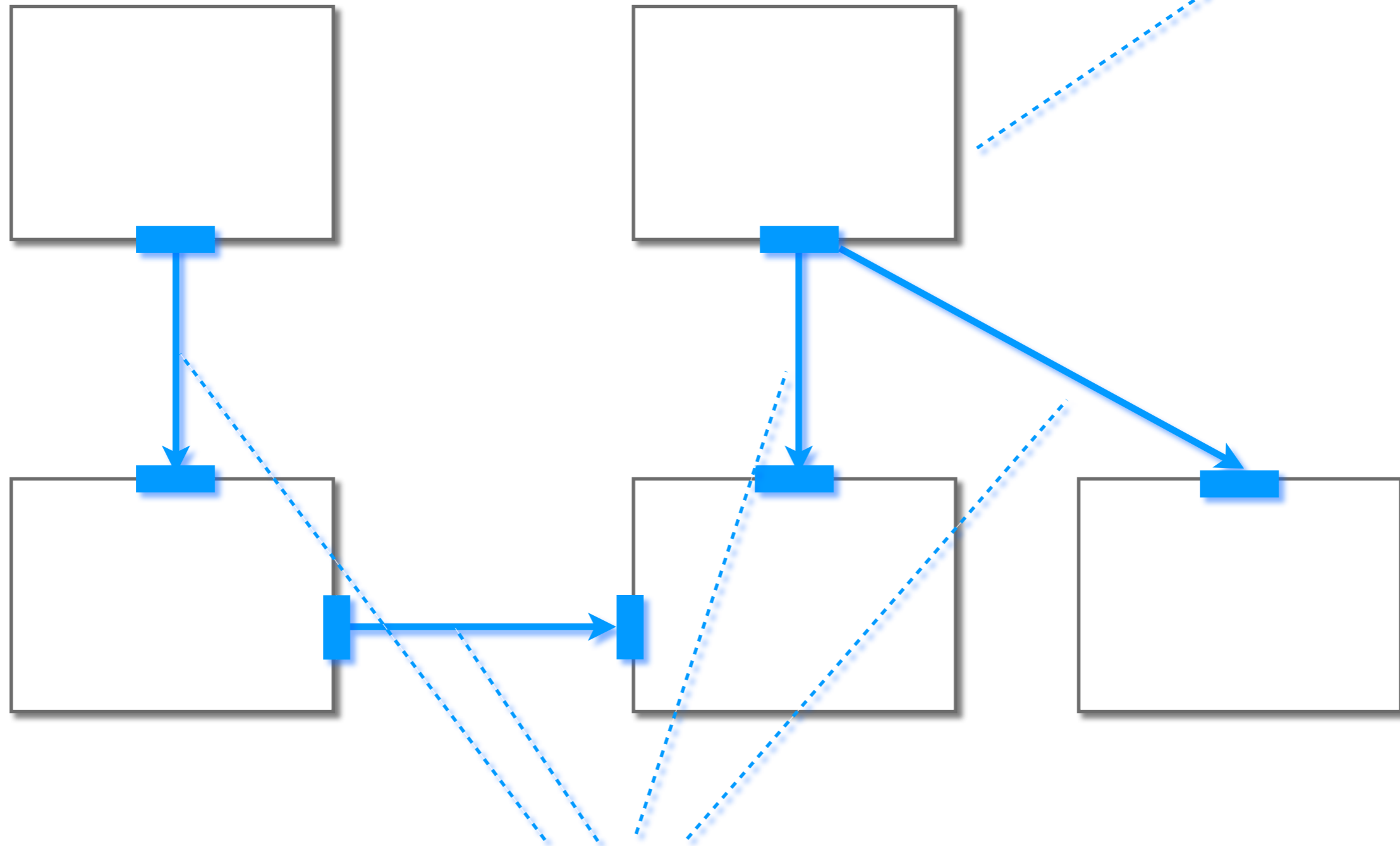
Autonomous operations

Limited interaction with other systems

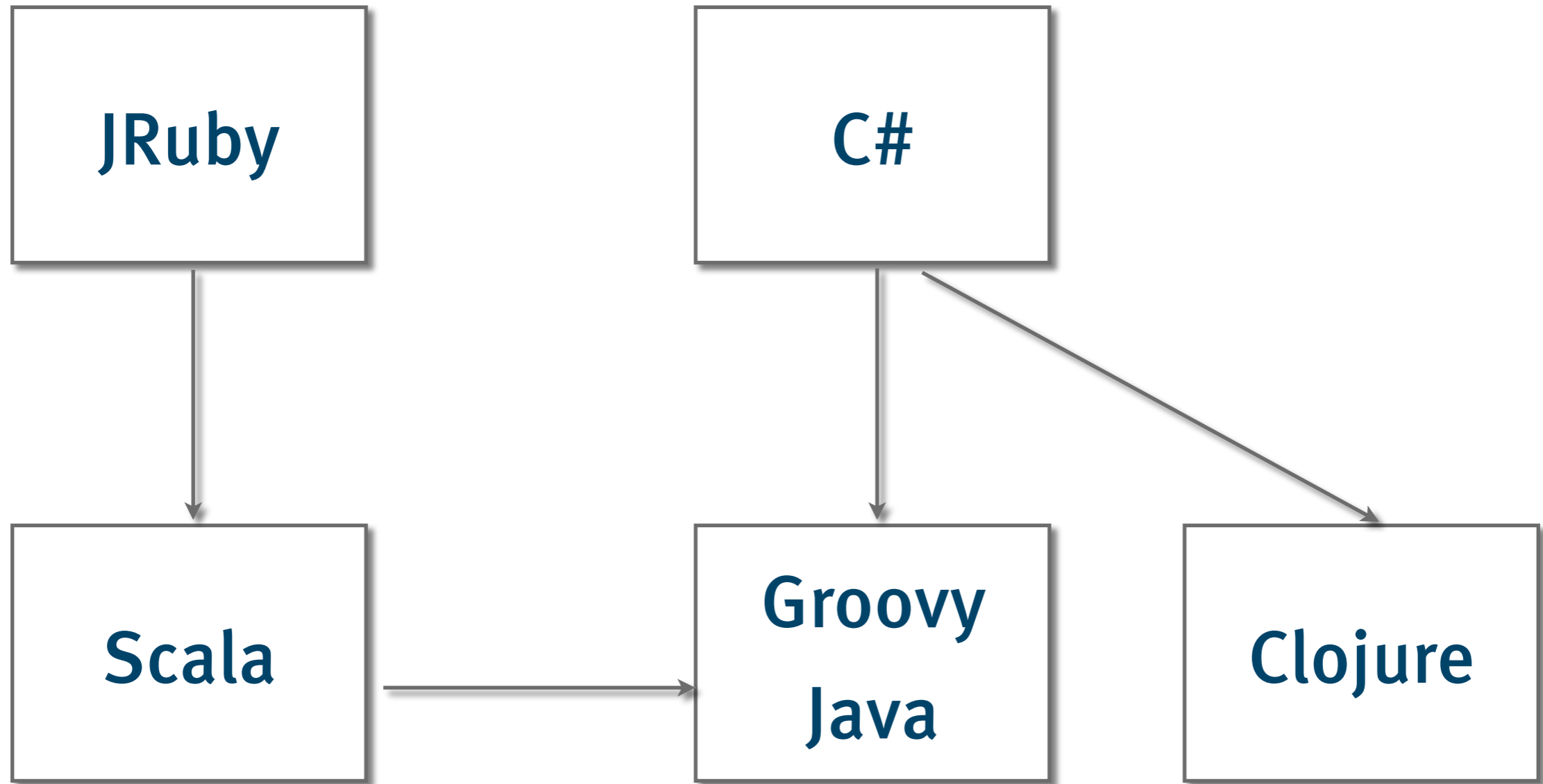
Micro vs. Macro architecture

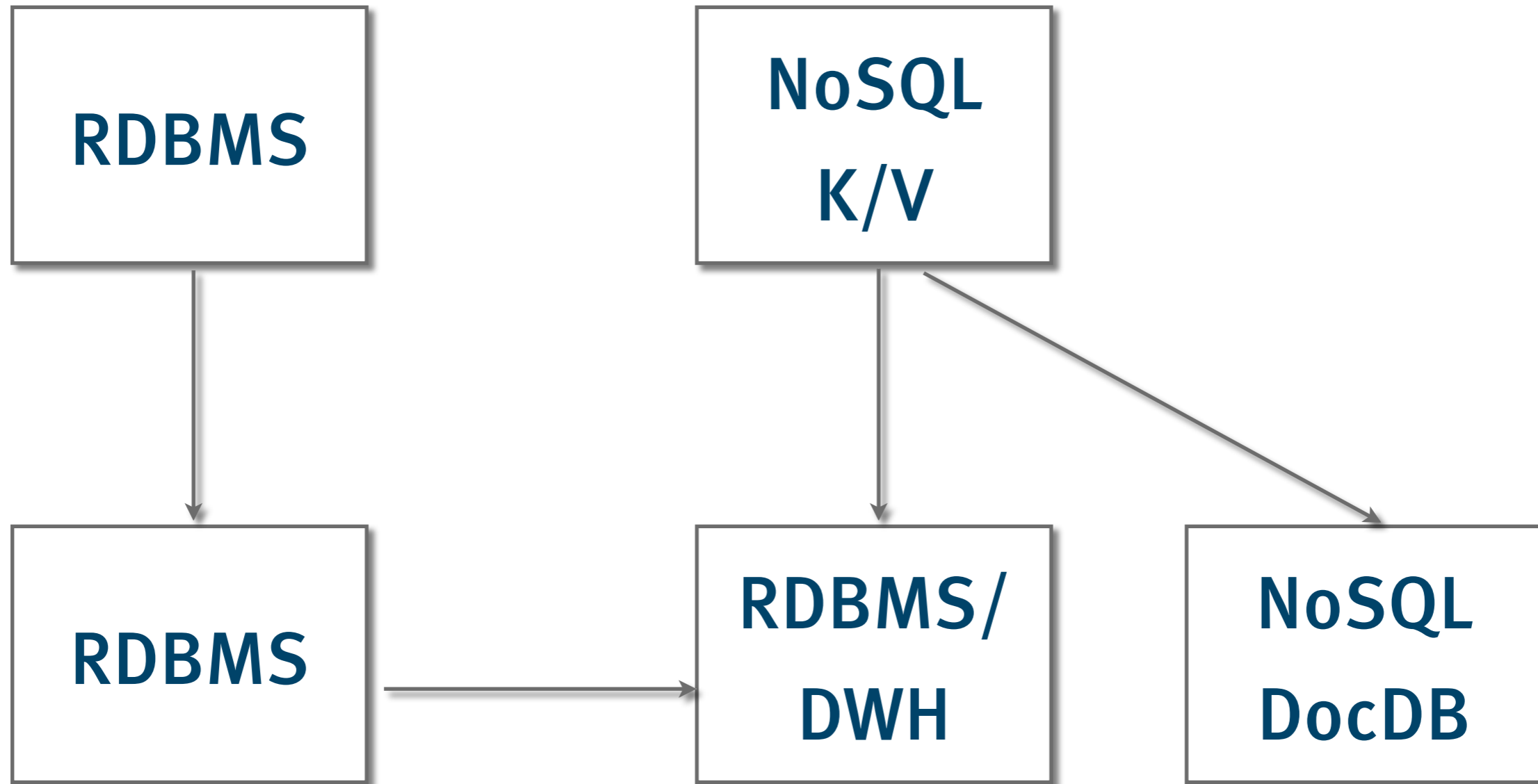


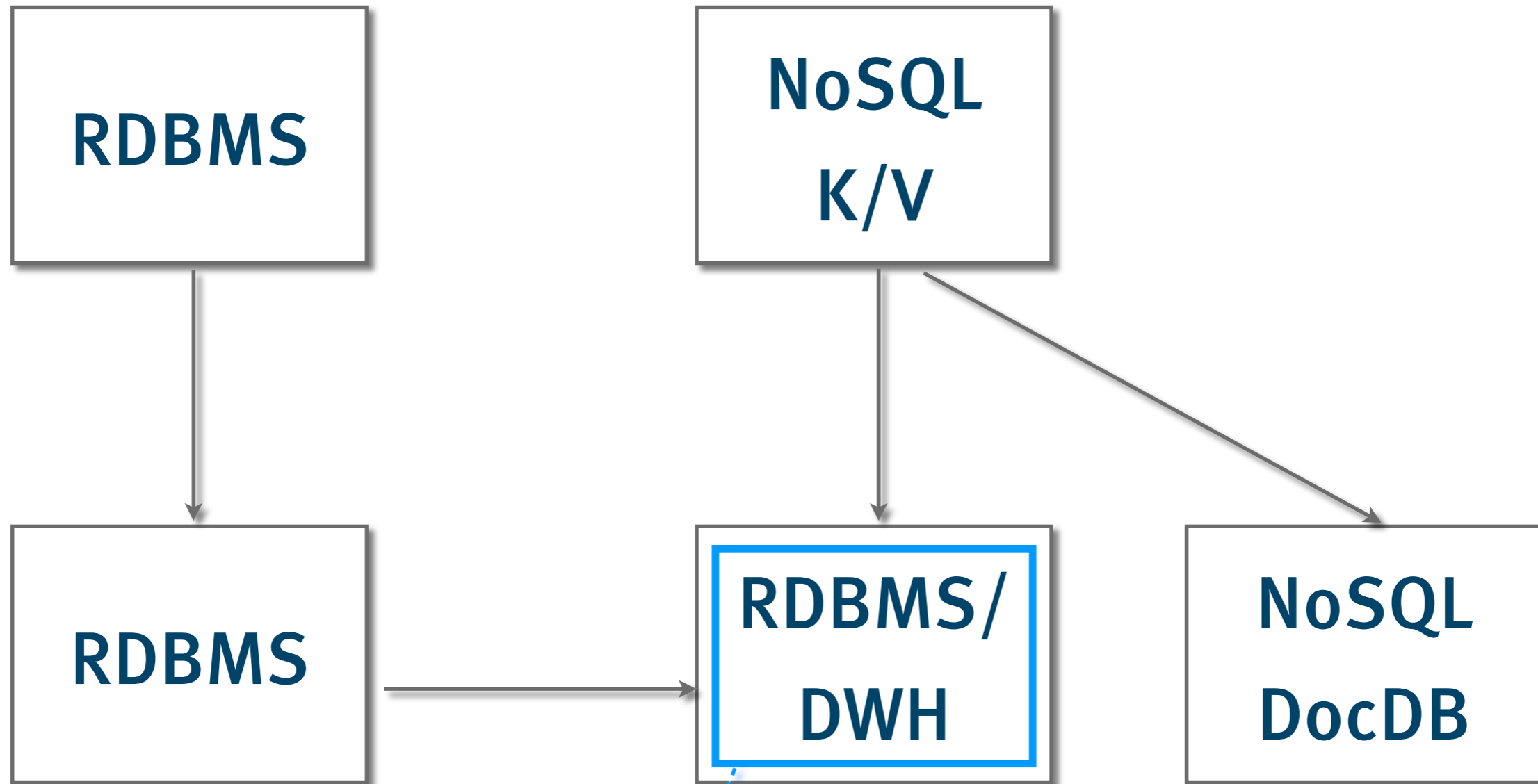
Domain architecture



Macro (technical) architecture







Micro architecture

Afraid of chaos?

Necessary Rules & Guidelines

Cross-system

Responsibilities

UI integration

Communication protocols

Data formats

Redundant data

BI interfaces

Logging, Monitoring

System-internal

Programming languages

Development tools

Frameworks

Process/Workflow control

Persistence

Design patterns

Coding guidelines

(Deployment, Operations)

Domain Architecture



Cross-system Rules



System-internal Rules



Main objectives over time

Ease of development

Homogeneity

Cohesion

Simplicity

Modularity

Decoupling

(Support for)

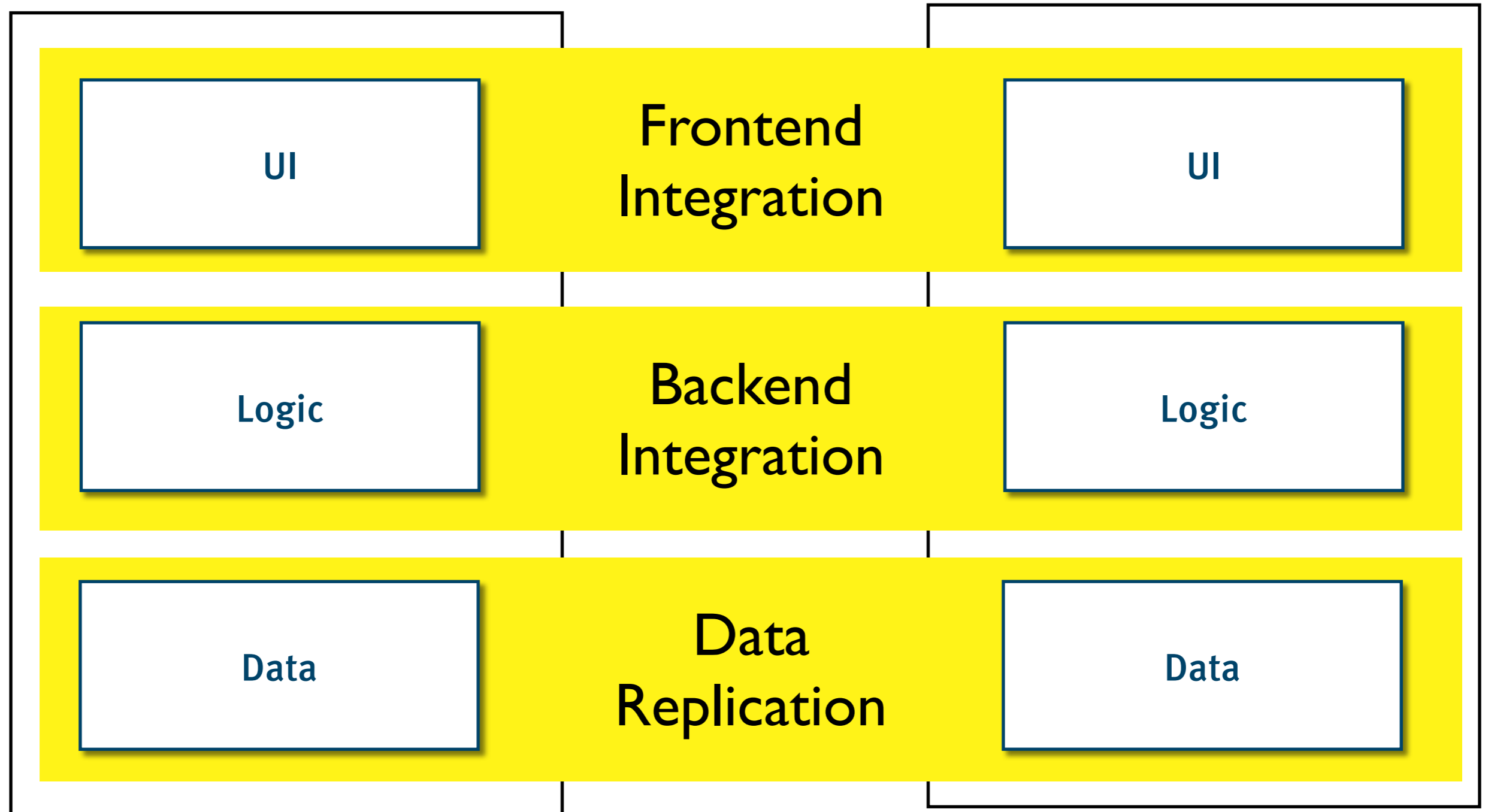
Heterogeneity

Autonomy



Separate systems, loosely coupled?

Integration options

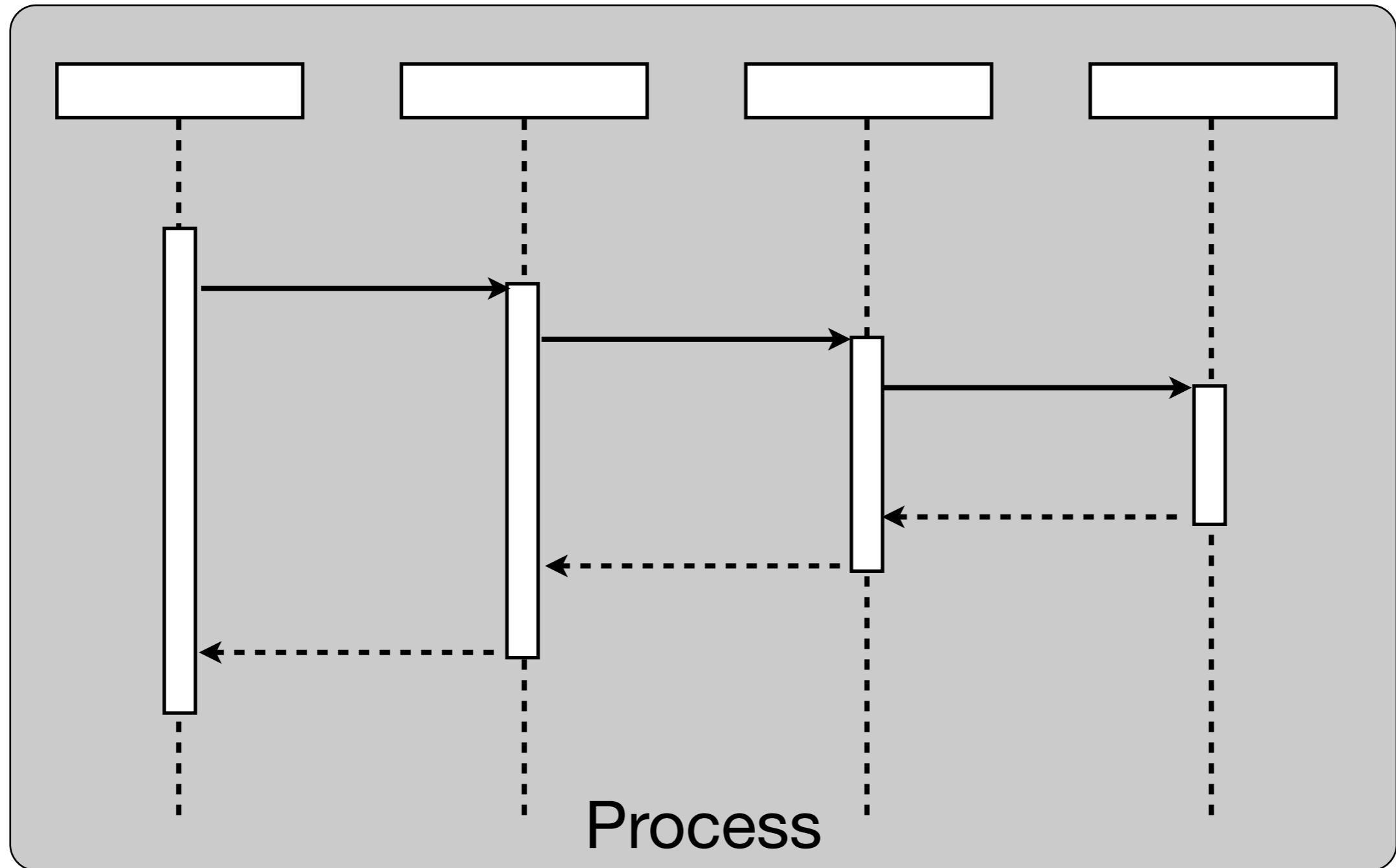


Data integration

Data *replication*

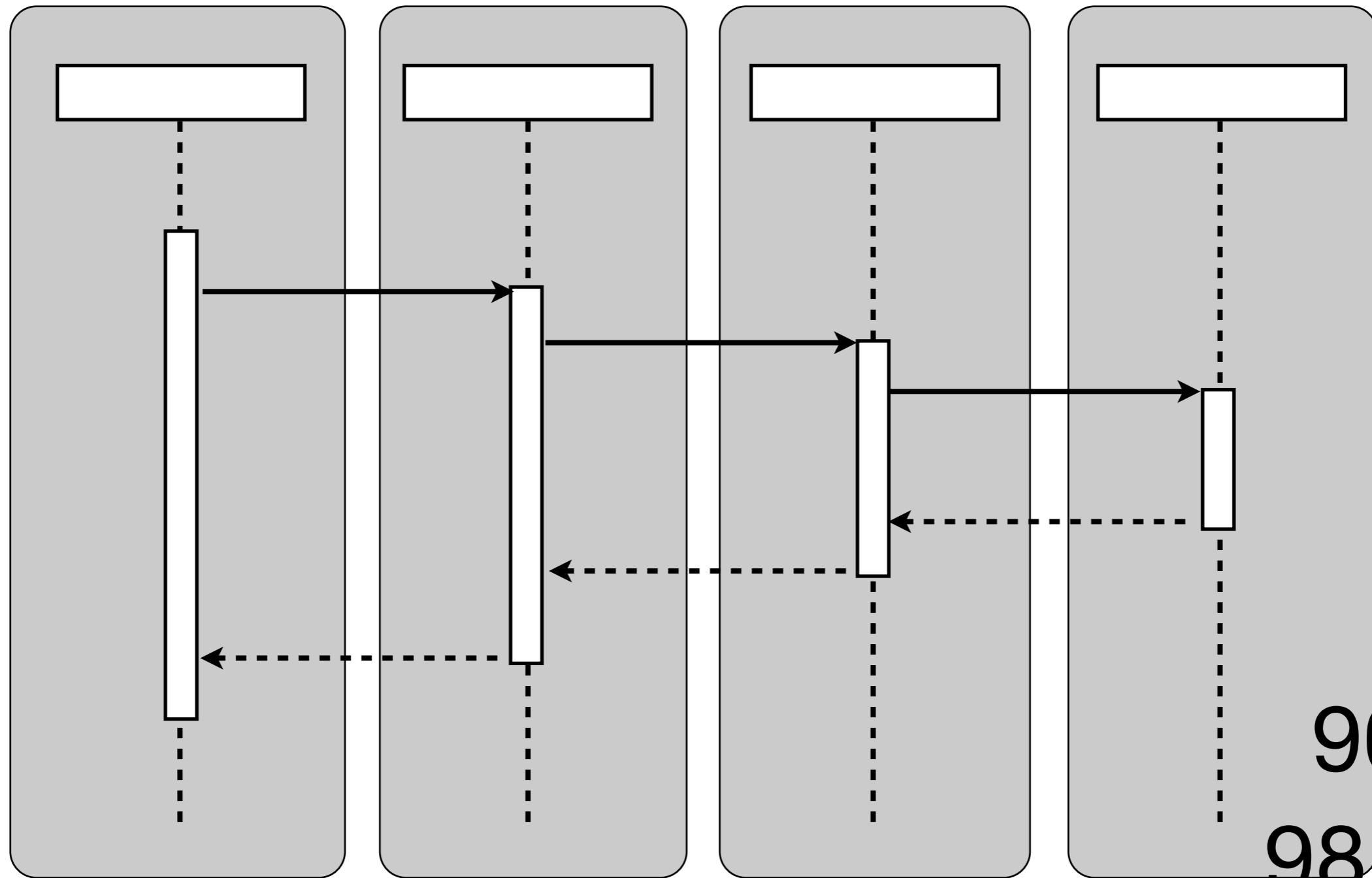
Or: Why data redundancy is good for you

Call Stack

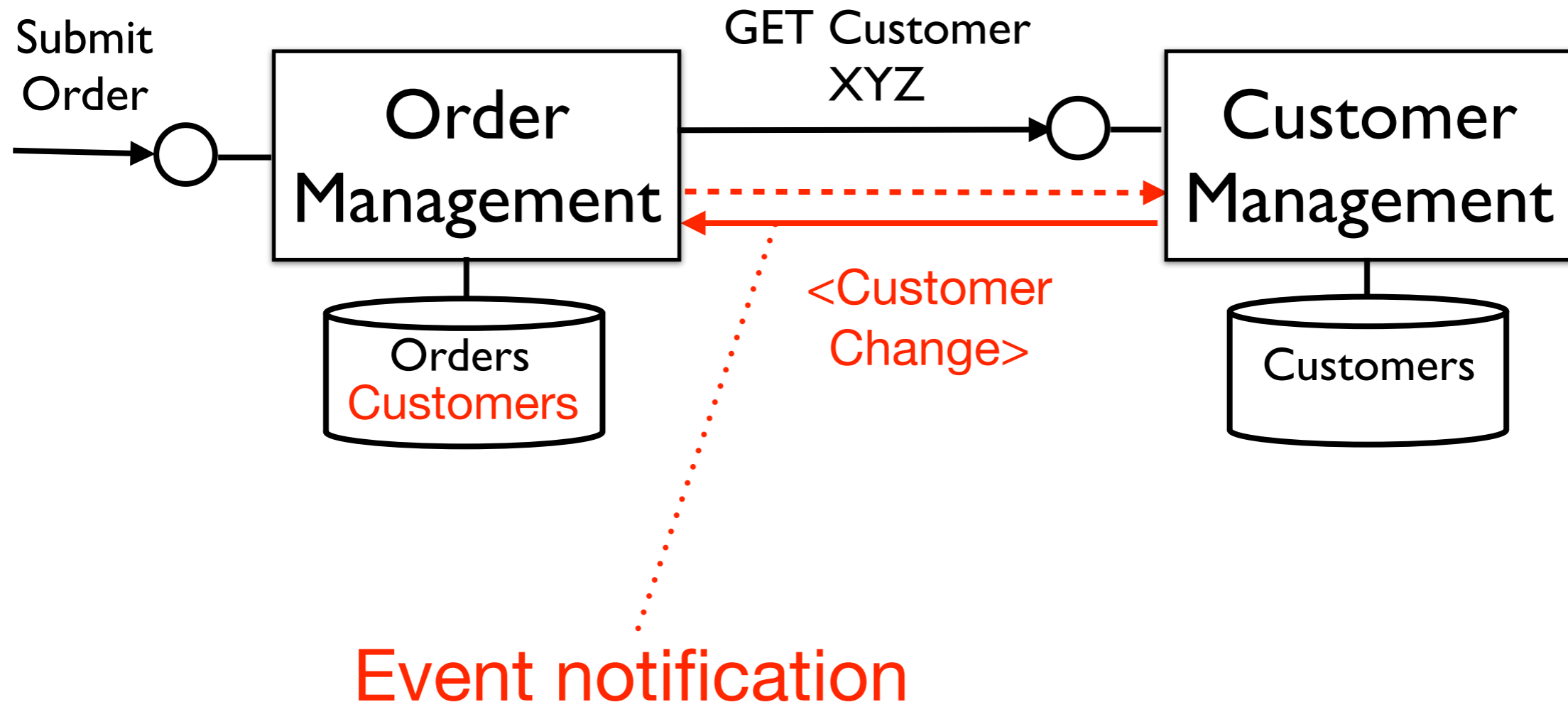


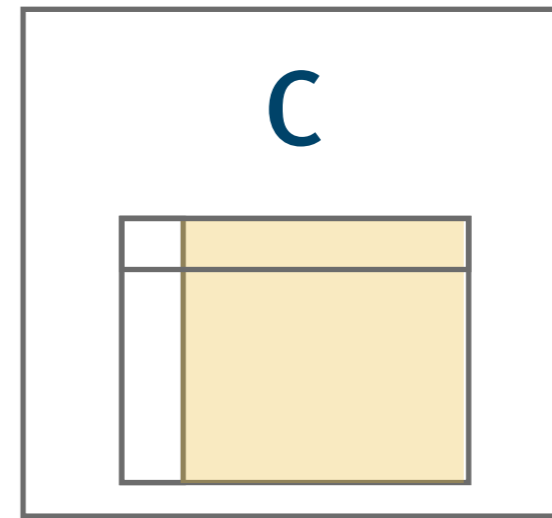
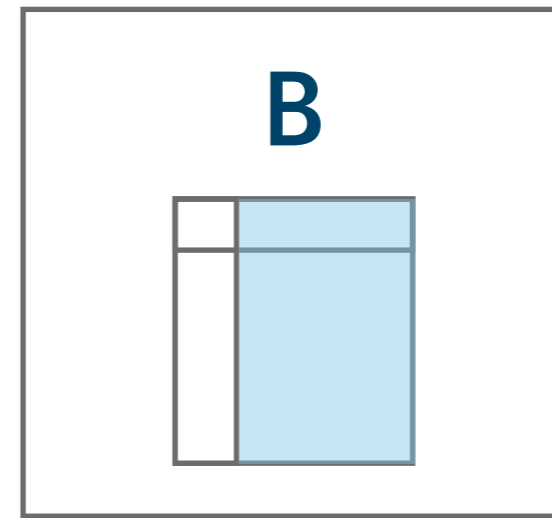
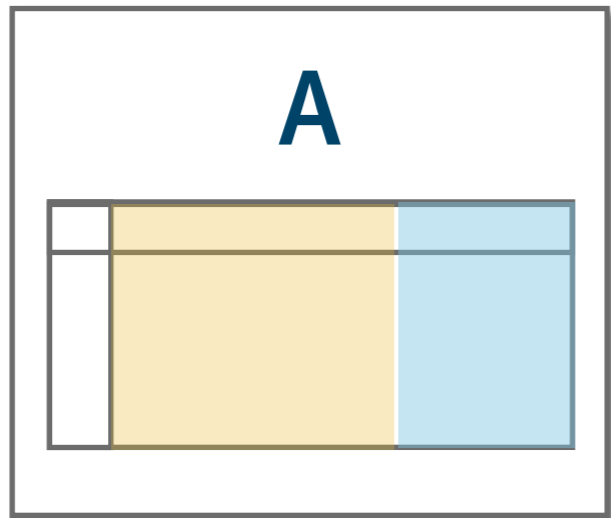
Assumed Success Probability: 99.9%

Distributed Call Stack



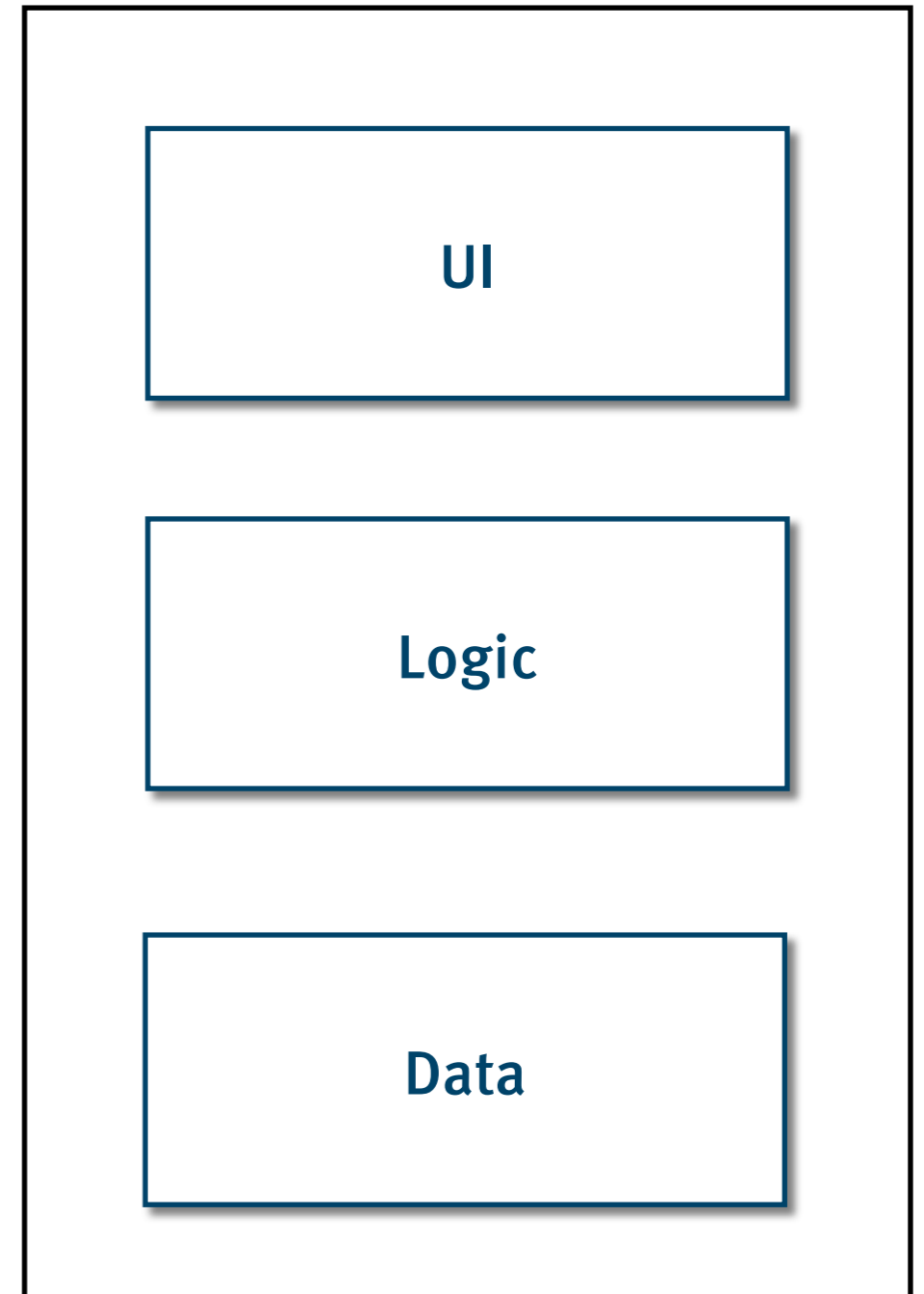
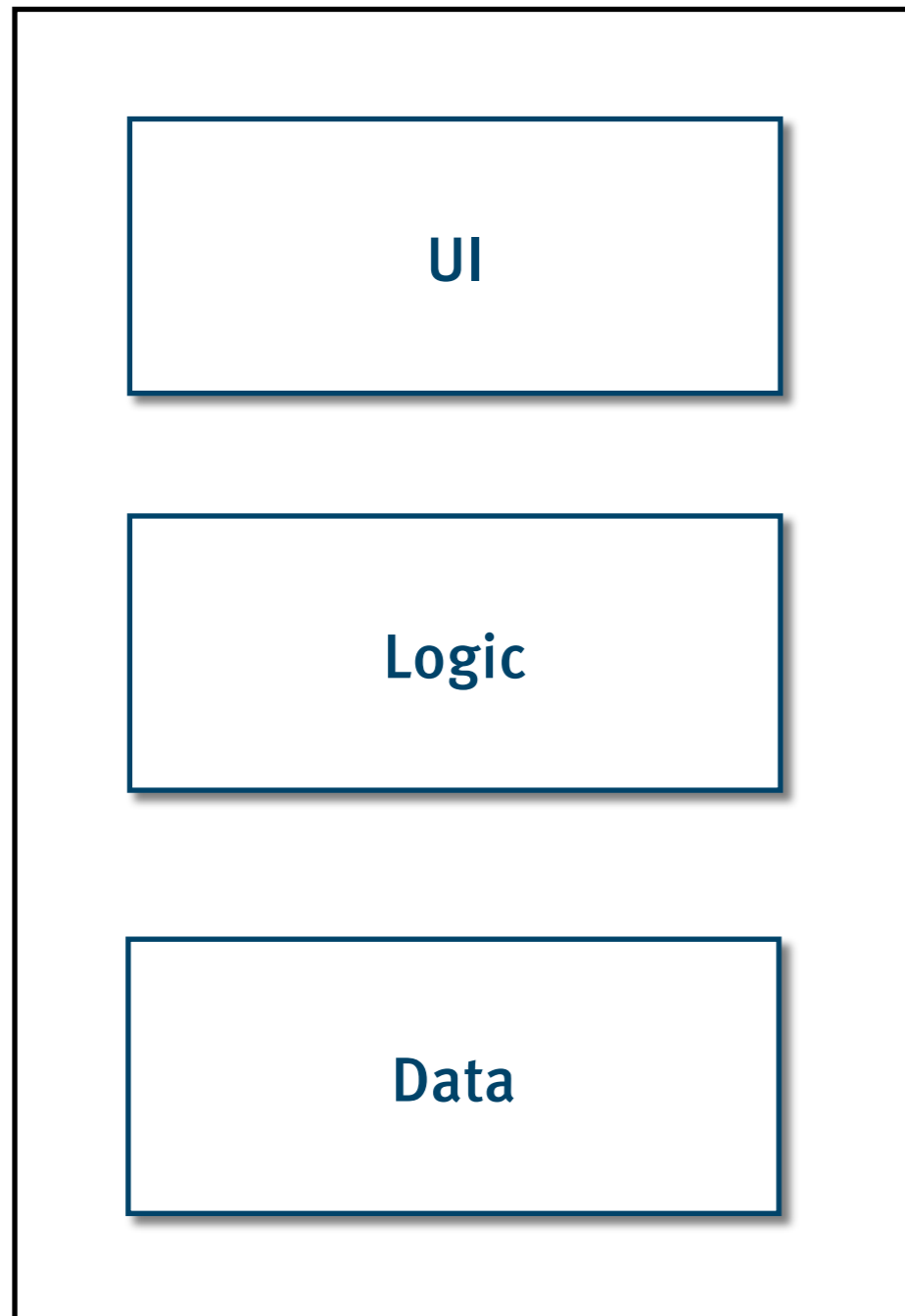
Assumed Success Probability: ~~99.6%~~



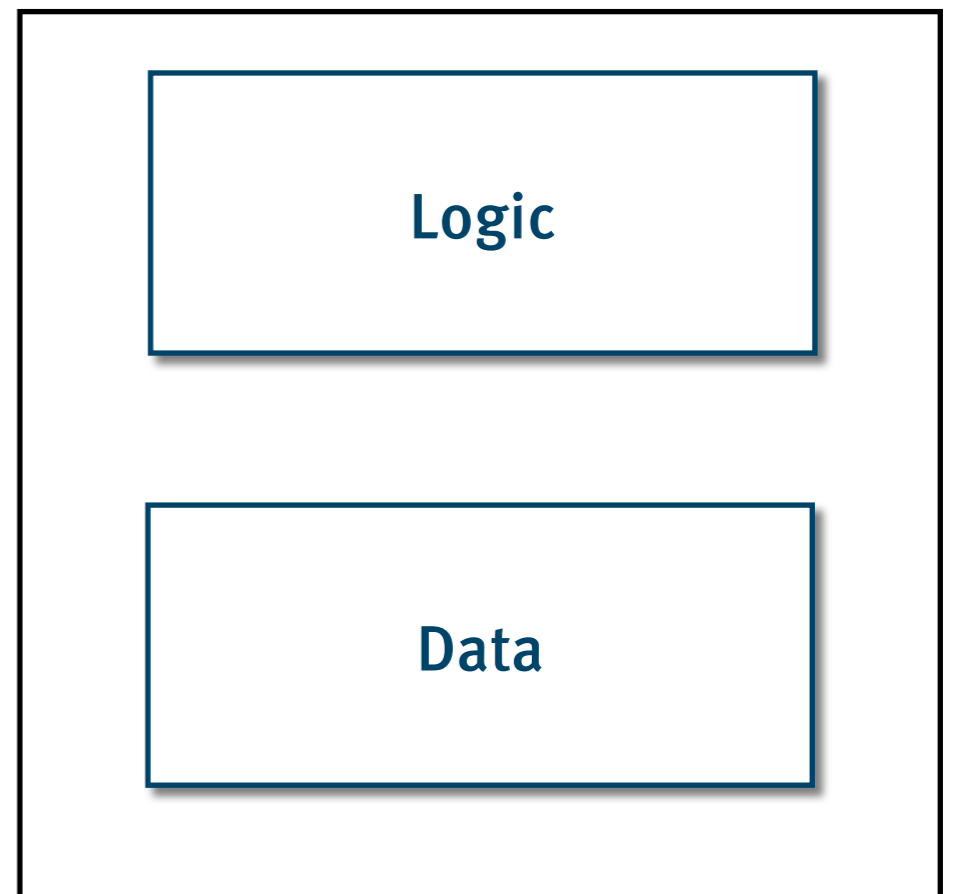
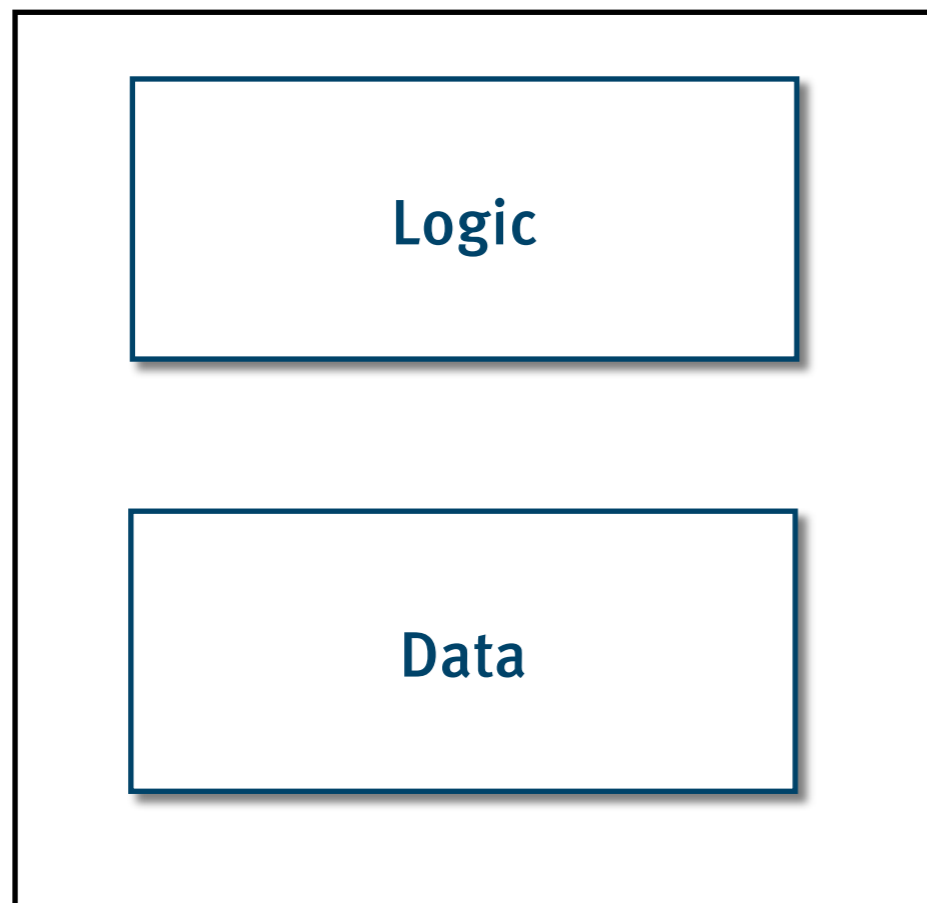
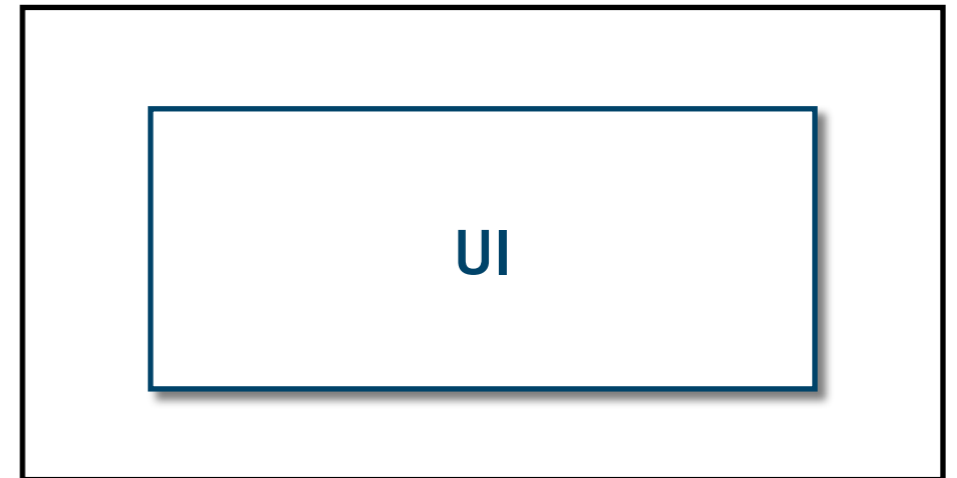
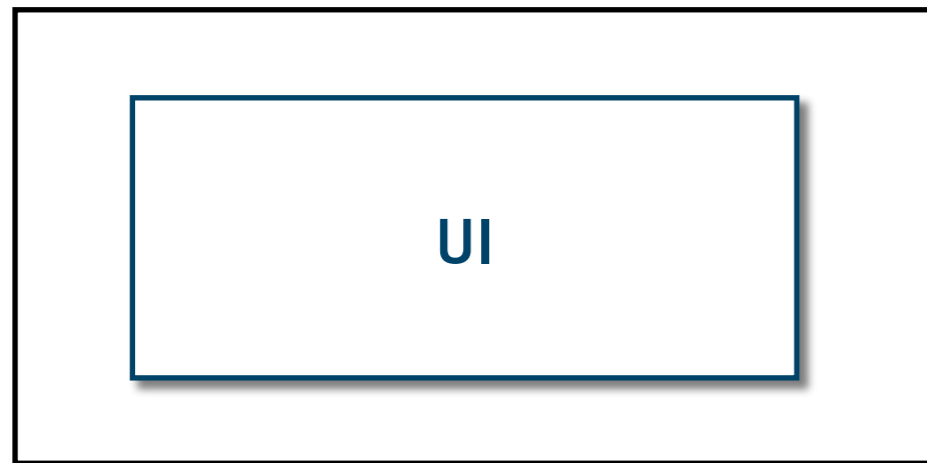


Frontend integration

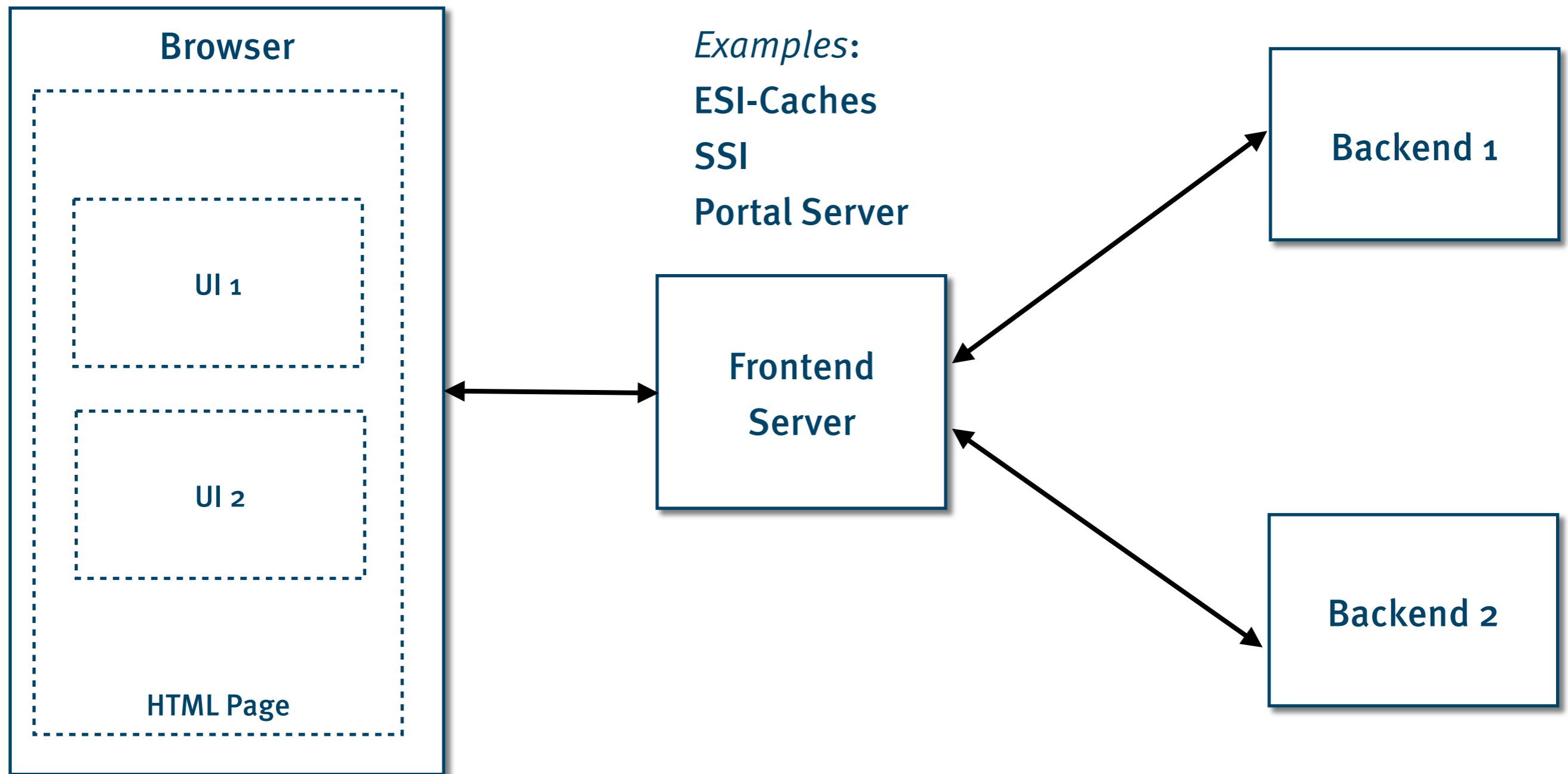
Frontend integration



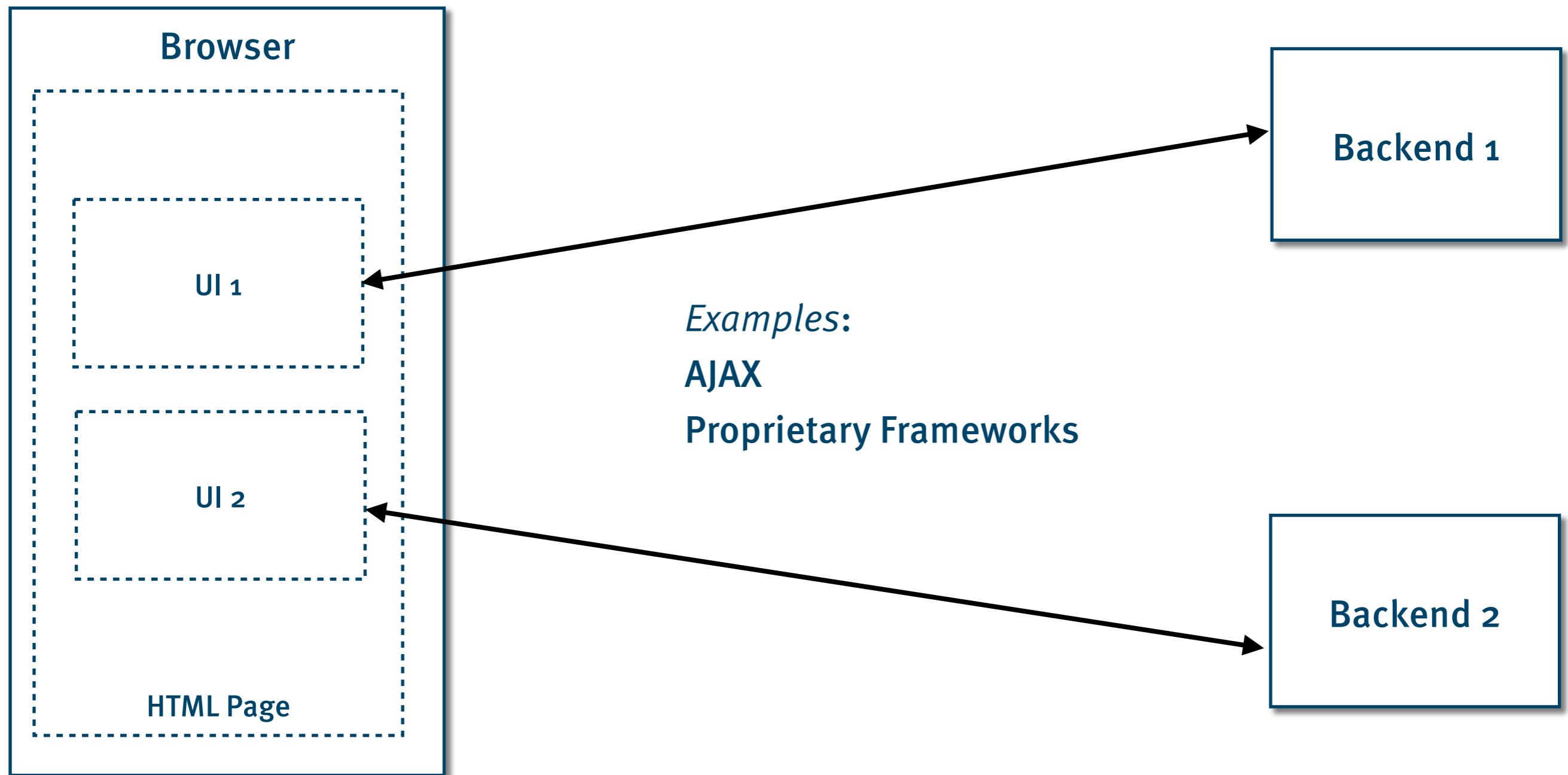
Frontend integration



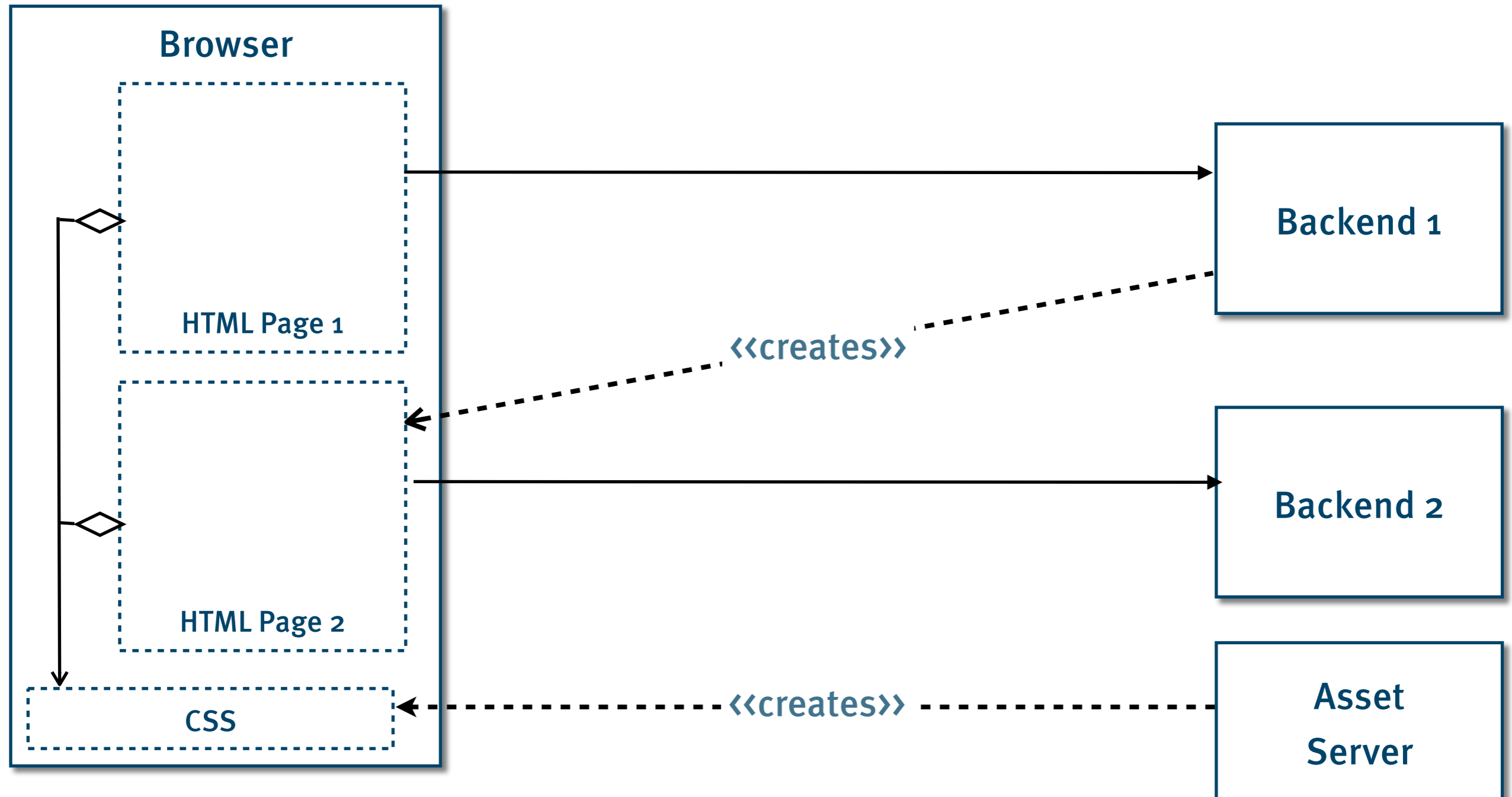
Server-side integration



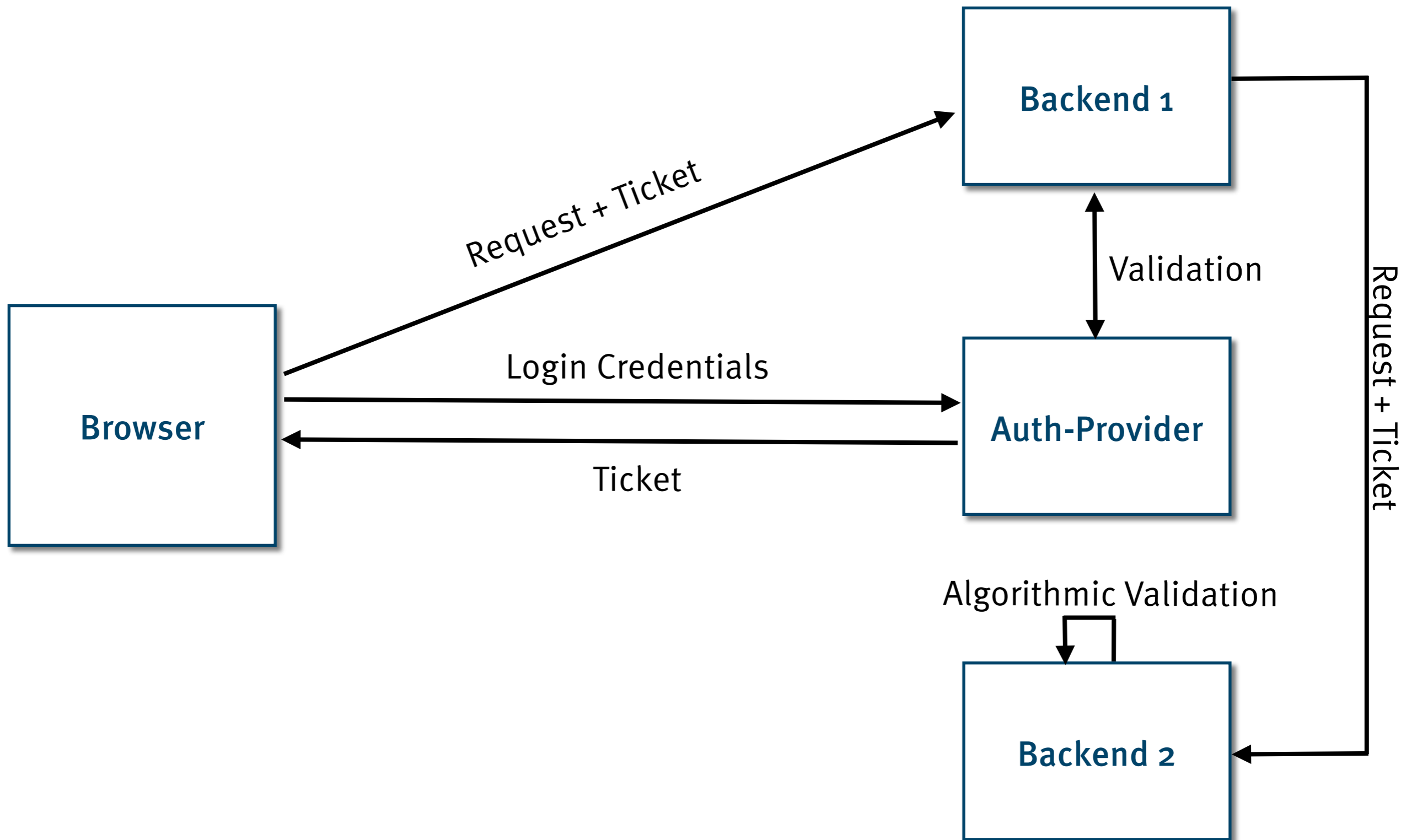
Client-side integration



Links



Single Sign-On



Summary

Think about the systems that make up your system

Separate micro and macro architectures

Address UI integration and SSO

Thanks!

Q&A

Contact

Stefan Tilkov

+49 2137 3366-0

stefan.tilkov@innoq.com

@stilkov

<http://www.innoq.com/blog/st/>



innoQ Deutschland GmbH

Krischerstr. 100
40789 Monheim am Rhein
Germany

Phone: +49 2173 3366-0

<http://www.innoq.com>

innoQ Schweiz GmbH

Gewerbestr. 11
CH-6330 Cham
Switzerland

Phone: +41 41 743 0116

info@innoq.com

