TRAILHEAD
TECHNOLOGY PARTNERS

Avoidifying Over-Complexification

Rooting Out Over-Engineering in Your Projects

Jonathan "J." Tower

TRAILHEAD
TECHNOLOGY PARTNERS

TRAILHEAD
TECHNOLOGY PARTNERS

credit: scienceworld.scholastic.com

10 Types of Over-Engineering

&

10 Rules to Help Avoid It

TRAILHEAD
TECHNOLOGY PARTNERS

Jonathan "J." Tower

Principal Consultant & Partner

TRAILHEAD
TECHNOLOGY PARTNERS

Microsoft MVP in .NET

jtower@trailheadtechnology.com

trailheadtechnology.com/blog

jtowermi

jtower

FREE

CONSULTATION

bit.ly/th-offer

github.com/trailheadtechnology/over-engineering

Why Worry About Over-Engineering

TRAILHEAD
TECHNOLOGY PARTNERS

The Issues With Over-Engineering

- Increased Costs
- Extended Dev Time
- Higher Risk Of Bugs
- Slowing New Developers
- Higher Complexity
- Loss of Focus on Features
- Not Relying on Outside Experts
- Reduced Testing Exposure
- Not Agile
- Solving the Wrong Problems
- New Tool vs Right Tool
- Often Lead to Rewrite
- Harder to Maintain

TRAILHEAD
TECHNOLOGY PARTNERS

The Issues With Over-Engineering

Increased Costs

Extended Dev Time

Higher Risk Of Bugs

Slowing New Developers

Higher Complexity

Loss of Focus on Features

Not Relying on Outside Experts

Reduced Testing Exposure

Not Agile

Solving the Wrong Problems

New Tool vs Right Tool

Often Lead to Rewrite

Harder to Maintain

TRAILHEAD TECHNOLOGY PARTNERS

The Issues With Over-Engineering

Increased Costs

Extended Dev Time

Higher Risk Of Bugs

Slowing New Developers

Higher Complexity

Loss of Focus on Features

Not Relying on Outside Experts

Reduced Testing Exposure

Not Agile

Solving the Wrong Problems

New Tool vs Right Tool

Often Lead to Rewrite

Harder to Maintain

TRAILHEAD TECHNOLOGY PARTNERS

The Issues With Over-Engineering

Increased Costs

Extended Dev Time

Higher Risk Of Bugs

Slowing New Developers

Higher Complexity

Loss of Focus on Features

Not Relying on Outside Experts

Reduced Testing Exposure

Not Agile

Solving the Wrong Problems

New Tool vs Right Tool

Often Lead to Rewrite

Harder to Maintain

TRAILHEAD TECHNOLOGY PARTNERS

The Issues With Over-Engineering

Increased Costs

Extended Dev Time

Higher Risk Of Bugs

Slowing New Developers

Higher Complexity

Loss of Focus on Features

Not Relying on Outside Experts

Reduced Testing Exposure

Not Agile

Solving the Wrong Problems

New Tool vs Right Tool

Often Lead to Rewrite

Harder to Maintain

TRAILHEAD
TECHNOLOGY PARTNERS

The Issues With Over-Engineering

Increased Costs

Extended Dev Time

Higher Risk Of Bugs

Slowing New Developers

Higher Complexity

Loss of Focus on Features

Not Relying on Outside Experts

Reduced Testing Exposure

Not Agile

Solving the Wrong Problems

New Tool vs Right Tool

Often Lead to Rewrite

Harder to Maintain

TRAILHEAD TECHNOLOGY PARTNERS

The Issues With Over-Engineering

Increased Costs

Extended Dev Time

Higher Risk Of Bugs

Slowing New Developers

Higher Complexity

Loss of Focus on Features

Not Relying on Outside Experts

Reduced Testing Exposure

Not Agile

Solving the Wrong Problems

New Tool vs Right Tool

Often Lead to Rewrite

Harder to Maintain

TRAILHEAD TECHNOLOGY PARTNERS

The Issues With Over-Engineering

Increased Costs

Extended Dev Time

Higher Risk Of Bugs

Slowing New Developers

Higher Complexity

Loss of Focus on Features

Not Relying on Outside Experts

Reduced Testing Exposure

Not Agile

Solving the Wrong Problems

New Tool vs Right Tool

Often Lead to Rewrite

Harder to Maintain

TRAILHEAD
TECHNOLOGY PARTNERS

The Issues With Over-Engineering

Increased Costs

Extended Dev Time

Higher Risk Of Bugs

Slowing New Developers

Higher Complexity

Loss of Focus on Features

Not Relying on Outside Experts

Reduced Testing Exposure

Not Agile

Solving the Wrong Problems

New Tool vs Right Tool

Often Lead to Rewrite

Harder to Maintain

TRAILHEAD TECHNOLOGY PARTNERS

Why We Over-Engineer

Curiosity or Enthusiasm

Perceived Prestige

Peer Pressure / Trends

Fear of Missing Out

Marketing and Hype

Personal Interest

Fear of Code Smells

Perceived Unique Requirements

Desire for Control

Lack of Awareness

Attempted Perf Optimization

Anticipation of Future Needs

Fear of Rework

Pressure from Stakeholders

Overconfidence

Over-Compensation

Fear of Simplicity

Educational Gap

Fear of Future/Past Perf Issues

Influence of High-Perf Domains

Preemptive Solution to Rare Issues

Avoid Vendor Lock-In

Showing Off

Future Proofing

Personal Interest

Burned Previously

TRAILHEAD
TECHNOLOGY PARTNERS

Curiosity or Enthusiasm

Perceived Prestige

Peer Pressure / Trends

Fear of Missing Out

Marketing and Hype

Personal Interest

Fear of Code Smells

Perceived Unique Requirements

Desire for Control

Lack of Awareness

Attempted Perf Optimization

Anticipation of Future Needs

Fear of Rework

Pressure from Stakeholders

Overconfidence

Over-Compensation

Fear of Simplicity

Educational Gap

Fear of Future/Past Perf Issues

Influence of High-Perf Domains

Preemptive Solution to Rare Issues

Avoid Vendor Lock-In

Showing Off

Future Proofing

Personal Interest

Burned Previously

TRAILHEAD
TECHNOLOGY PARTNERS

Curiosity or Enthusiasm

Perceived Prestige

Peer Pressure / Trends

Fear of Missing Out

Marketing and Hype

Personal Interest

Fear of Code Smells

Perceived Unique Requirements

Desire for Control

Lack of Awareness

Attempted Perf Optimization

Anticipation of Future Needs

Fear of Rework

Pressure from Stakeholders

Overconfidence

Over-Compensation

Fear of Simplicity

Educational Gap

Fear of Future/Past Perf Issues

Influence of High-Perf Domains

Preemptive Solution to Rare Issues

Avoid Vendor Lock-In

Showing Off

Future Proofing

Personal Interest

Burned Previously

TRAILHEAD
TECHNOLOGY PARTNERS

Curiosity or Enthusiasm

Perceived Prestige

Peer Pressure / Trends

Fear of Missing Out

Marketing and Hype

Personal Interest

Fear of Code Smells

Perceived Unique Requirements

Desire for Control

Lack of Awareness

Attempted Perf Optimization

Anticipation of Future Needs

Fear of Rework

Pressure from Stakeholders

Overconfidence

Over-Compensation

Fear of Simplicity

Educational Gap

Fear of Future/Past Perf Issues

Influence of High-Perf Domains

Preemptive Solution to Rare Issues

Avoid Vendor Lock-In

Showing Off

Future Proofing

Personal Interest

Burned Previously

TRAILHEAD
TECHNOLOGY PARTNERS

Curiosity or Enthusiasm

Perceived Prestige

Peer Pressure / Trends

Fear of Missing Out

Marketing and Hype

Personal Interest

Fear of Code Smells

Perceived Unique Requirements

Desire for Control

Lack of Awareness

Attempted Perf Optimization

Anticipation of Future Needs

Fear of Rework

Pressure from Stakeholders

Overconfidence

Over-Compensation

Fear of Simplicity

Educational Gap

Fear of Future/Past Perf Issues

Influence of High-Perf Domains

Preemptive Solution to Rare Issues

Avoid Vendor Lock-In

Showing Off

Future Proofing

Personal Interest

Burned Previously

TRAILHEAD TECHNOLOGY PARTNERS

Curiosity or Enthusiasm	Perceived Prestige	Peer Pressure / Trends	Fear of Missing Out	Marketing and Hype	Personal Interest	Fear of Code Smells
Perceived Unique Requirements	Desire for Control	Lack of Awareness	Attempted Perf Optimization	Anticipation of Future Needs	Fear of Rework	Pressure from Stakeholders
Overconfidence	Over-Compensation	Fear of Simplicity	Educational Gap	Fear of Future/Past Perf Issues	Influence of High-Perf Domains	Preemptive Solution to Rare Issues
Avoid Vendor Lock-In	Showing Off	Future Proofing	Personal Interest	Burned Previously		

TRAILHEAD
TECHNOLOGY PARTNERS

Curiosity or Enthusiasm

Perceived Prestige

Peer Pressure / Trends

Fear of Missing Out

Marketing and Hype

Personal Interest

Fear of Code Smells

Perceived Unique Requirements

Desire for Control

Lack of Awareness

Attempted Perf Optimization

Anticipation of Future Needs

Fear of Rework

Pressure from Stakeholders

Overconfidence

Over-Compensation

Fear of Simplicity

Educational Gap

Fear of Future/Past Perf Issues

Influence of High-Perf Domains

Preemptive Solution to Rare Issues

Avoid Vendor Lock-In

Showing Off

Future Proofing

Personal Interest

Burned Previously

TRAILHEAD TECHNOLOGY PARTNERS

Curiosity or Enthusiasm

Perceived Prestige

Peer Pressure / Trends

Fear of Missing Out

Marketing and Hype

Personal Interest

Fear of Code Smells

Perceived Unique Requirements

Desire for Control

Lack of Awareness

Attempted Perf Optimization

Anticipation of Future Needs

Fear of Rework

Pressure from Stakeholders

Overconfidence

Over-Compensation

Fear of Simplicity

Educational Gap

Fear of Future/Past Perf Issues

Influence of High-Perf Domains

Preemptive Solution to Rare Issues

Avoid Vendor Lock-In

Showing Off

Future Proofing

Personal Interest

Burned Previously

TRAILHEAD TECHNOLOGY PARTNERS

Curiosity or Enthusiasm

Perceived Prestige

Peer Pressure / Trends

Fear of Missing Out

Marketing and Hype

Personal Interest

Fear of Code Smells

Perceived Unique Requirements

Desire for Control

Lack of Awareness

Attempted Perf Optimization

Anticipation of Future Needs

Fear of Rework

Pressure from Stakeholders

Overconfidence

Over-Compensation

Fear of Simplicity

Educational Gap

Fear of Future/Past Perf Issues

Influence of High-Perf Domains

Preemptive Solution to Rare Issues

Avoid Vendor Lock-In

Showing Off

Future Proofing

Personal Interest

Burned Previously

TRAILHEAD TECHNOLOGY PARTNERS

Curiosity or Enthusiasm

Perceived Prestige

Peer Pressure / Trends

Fear of Missing Out

Marketing and Hype

Personal Interest

Fear of Code Smells

Perceived Unique Requirements

Desire for Control

Lack of Awareness

Attempted Perf Optimization

Anticipation of Future Needs

Fear of Rework

Pressure from Stakeholders

Overconfidence

Over-Compensation

Fear of Simplicity

Educational Gap

Fear of Future/Past Perf Issues

Influence of High-Perf Domains

Preemptive Solution to Rare Issues

Avoid Vendor Lock-In

Showing Off

Future Proofing

Personal Interest

Burned Previously

TRAILHEAD
TECHNOLOGY PARTNERS

Curiosity or Enthusiasm

Perceived Prestige

Peer Pressure / Trends

Fear of Missing Out

Marketing and Hype

Personal Interest

Fear of Code Smells

Perceived Unique Requirements

Desire for Control

Lack of Awareness

Attempted Perf Optimization

Anticipation of Future Needs

Fear of Rework

Pressure from Stakeholders

Overconfidence

Over-Compensation

Fear of Simplicity

Educational Gap

Fear of Future/Past Perf Issues

Influence of High-Perf Domains

Preemptive Solution to Rare Issues

Avoid Vendor Lock-In

Showing Off

Future Proofing

Personal Interest

Burned Previously

TRAILHEAD
TECHNOLOGY PARTNERS

Curiosity or Enthusiasm

Perceived Prestige

Peer Pressure / Trends

Fear of Missing Out

Marketing and Hype

Personal Interest

Fear of Code Smells

Perceived Unique Requirements

Desire for Control

Lack of Awareness

Attempted Perf Optimization

Anticipation of Future Needs

Fear of Rework

Pressure from Stakeholders

Overconfidence

Over-Compensation

Fear of Simplicity

Educational Gap

Fear of Future/Past Perf Issues

Influence of High-Perf Domains

Preemptive Solution to Rare Issues

Avoid Vendor Lock-In

Showing Off

Future Proofing

Personal Interest

Burned Previously

TRAILHEAD
TECHNOLOGY PARTNERS

Curiosity or Enthusiasm

Perceived Prestige

Peer Pressure / Trends

Fear of Missing Out

Marketing and Hype

Personal Interest

Fear of Code Smells

Perceived Unique Requirements

Desire for Control

Lack of Awareness

Attempted Perf Optimization

Anticipation of Future Needs

Fear of Rework

Pressure from Stakeholders

Overconfidence

Over-Compensation

Fear of Simplicity

Educational Gap

Fear of Future/Past Perf Issues

Influence of High-Perf Domains

Preemptive Solution to Rare Issues

Avoid Vendor Lock-In

Showing Off

Future Proofing

Personal Interest

Burned Previously

TRAILHEAD TECHNOLOGY PARTNERS

Curiosity or Enthusiasm

Perceived Prestige

Peer Pressure / Trends

Fear of Missing Out

Marketing and Hype

Personal Interest

Fear of Code Smells

Perceived Unique Requirements

Desire for Control

Lack of Awareness

Attempted Perf Optimization

Anticipation of Future Needs

Fear of Rework

Pressure from Stakeholders

Overconfidence

Over-Compensation

Fear of Simplicity

Educational Gap

Fear of Future/Past Perf Issues

Influence of High-Perf Domains

Preemptive Solution to Rare Issues

Avoid Vendor Lock-In

Showing Off

Future Proofing

Personal Interest

Burned Previously

TRAILHEAD
TECHNOLOGY PARTNERS

Curiosity or Enthusiasm

Perceived Prestige

Peer Pressure / Trends

Fear of Missing Out

Marketing and Hype

Personal Interest

Fear of Code Smells

Perceived Unique Requirements

Desire for Control

Lack of Awareness

Attempted Perf Optimization

Anticipation of Future Needs

Fear of Rework

Pressure from Stakeholders

Overconfidence

Over-Compensation

Fear of Simplicity

Educational Gap

Fear of Future/Past Perf Issues

Influence of High-Perf Domains

Preemptive Solution to Rare Issues

Avoid Vendor Lock-In

Showing Off

Future Proofing

Personal Interest

Burned Previously

TRAILHEAD TECHNOLOGY PARTNERS

Curiosity or Enthusiasm

Perceived Prestige

Peer Pressure / Trends

Fear of Missing Out

Marketing and Hype

Personal Interest

Fear of Code Smells

Perceived Unique Requirements

Desire for Control

Lack of Awareness

Attempted Perf Optimization

Anticipation of Future Needs

Fear of Rework

Pressure from Stakeholders

Overconfidence

Over-Compensation

Fear of Simplicity

Educational Gap

Fear of Future/Past Perf Issues

Influence of High-Perf Domains

Preemptive Solution to Rare Issues

Avoid Vendor Lock-In

Showing Off

Future Proofing

Personal Interest

Burned Previously

TRAILHEAD TECHNOLOGY PARTNERS

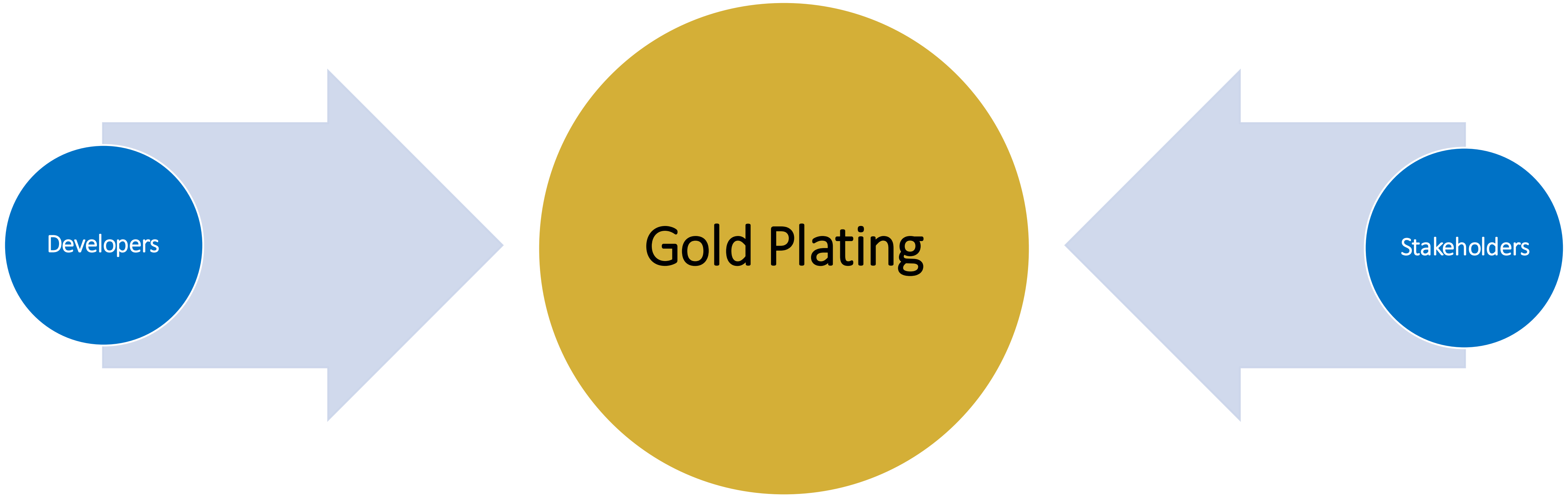
10 Common **Types** of Over-Engineering

TRAILHEAD
TECHNOLOGY PARTNERS

Gold-Plating

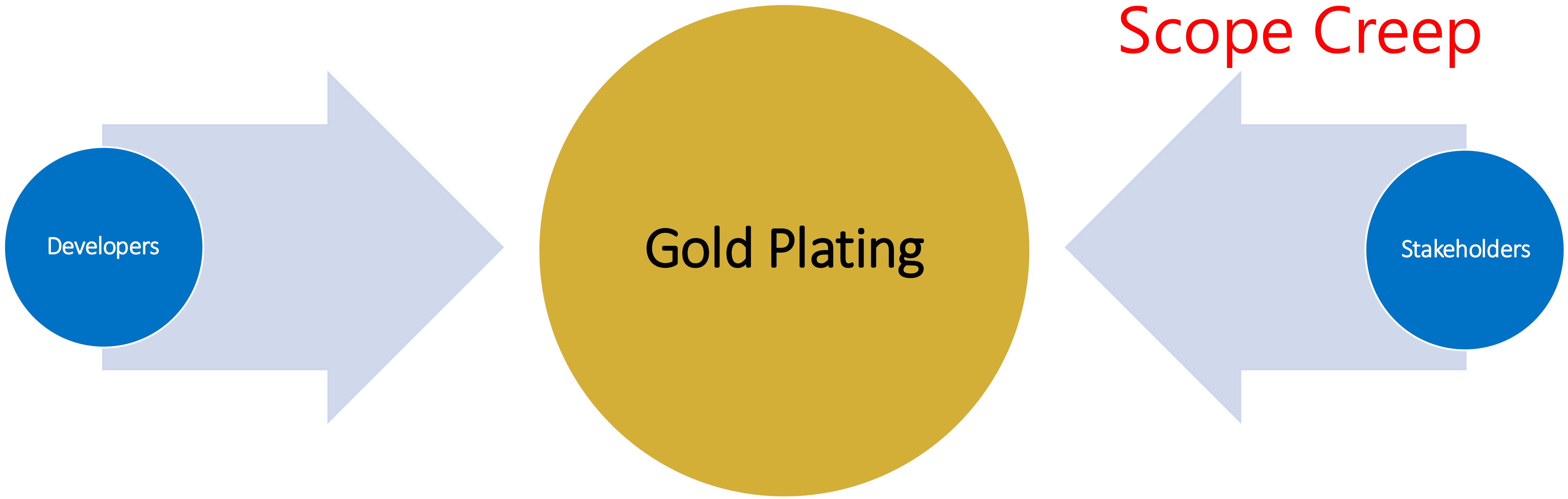
Over-Engineering Type 1

Sources of Gold-Plating

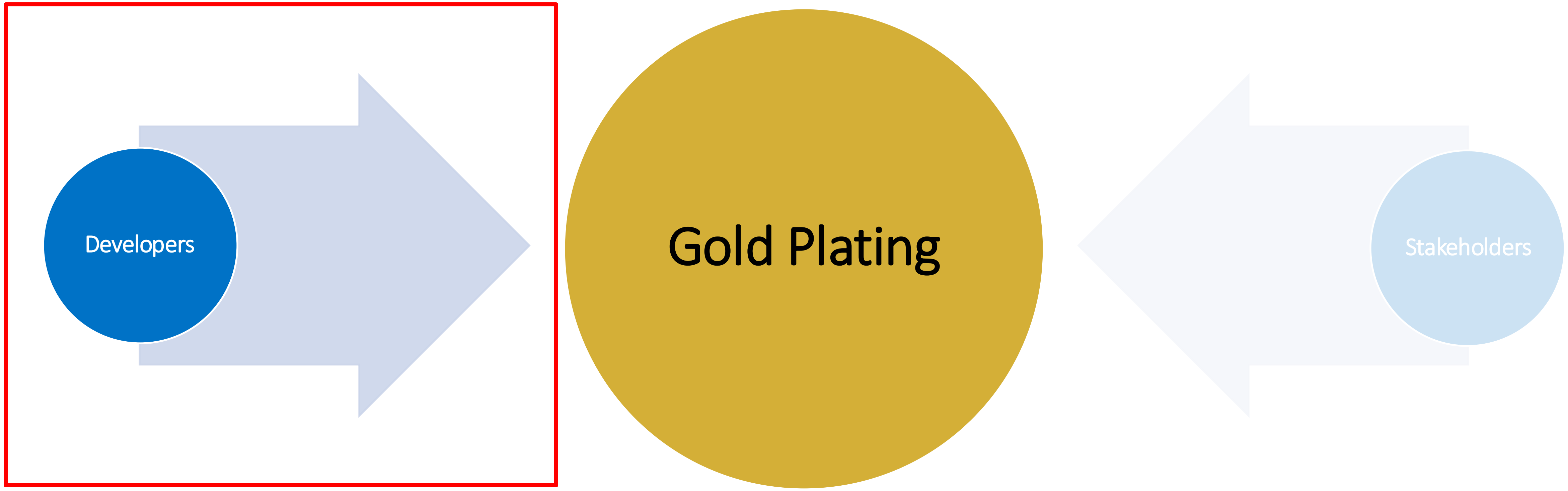


TRAILHEAD TECHNOLOGY PARTNERS

Sources of Gold-Plating

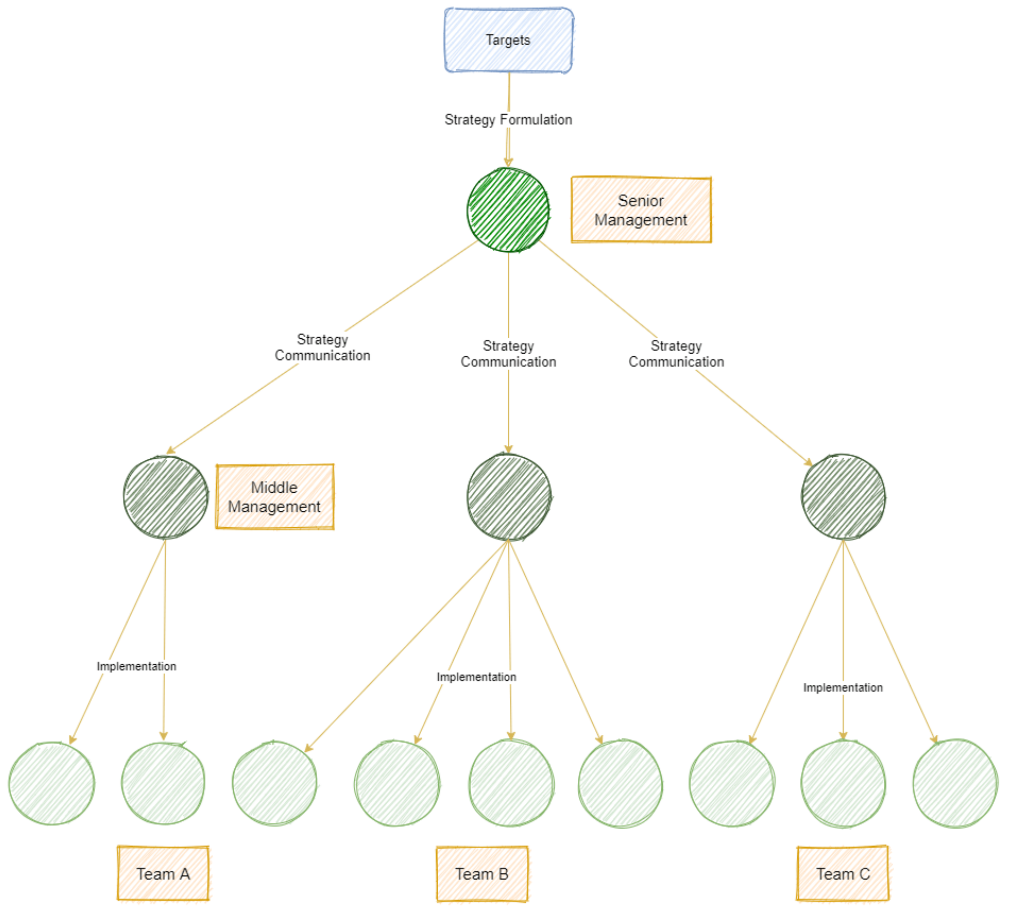


Sources of Gold-Plating



OO Gymnastics

Over-Engineering Type 2



Types of OO Gymnastics

Unnecessary Generics

Complex Inheritance Hierarchies

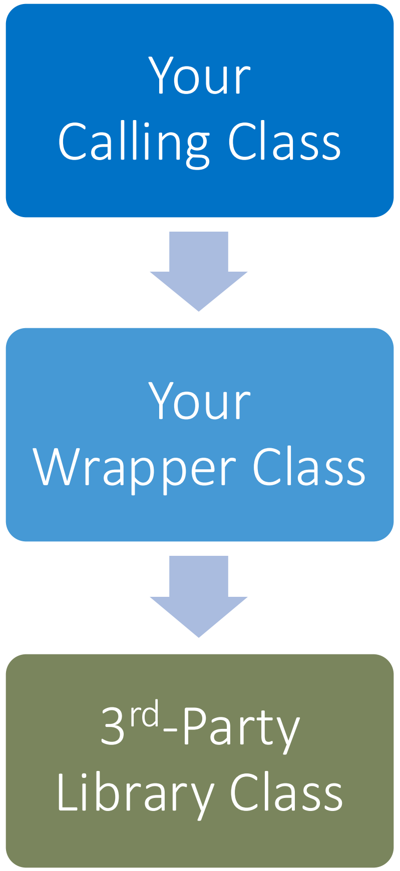
Properties Instead of Fields

Over-Abstraction

Over-Engineering Type 3

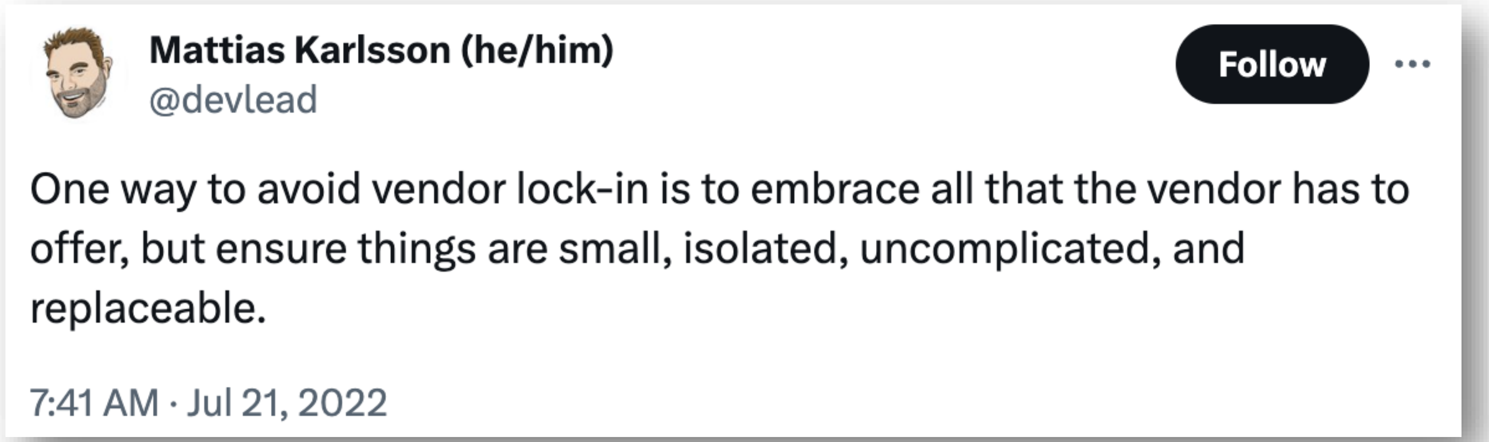
TRAILHEAD TECHNOLOGY PARTNERS

Over-Abstraction



Why You Don't Need Over-Abstraction

- Embrace all 3rd-party library has to offer
- Ensure to select libraries that:
  1. Are small
  2. Isolated
  3. Uncomplicated
  4. Replaceable



Mattias Karlsson (he/him)
@devlead

One way to avoid vendor lock-in is to embrace all that the vendor has to offer, but ensure things are small, isolated, uncomplicated, and replaceable.

7:41 AM · Jul 21, 2022

TRAILHEAD TECHNOLOGY PARTNERS

Over-Built Scalability

Over-Engineering Type 4

Scalability Rule of Thumb

"Build your software for, at most, 1-2 orders of magnitude more than you currently need."

- J. Tower

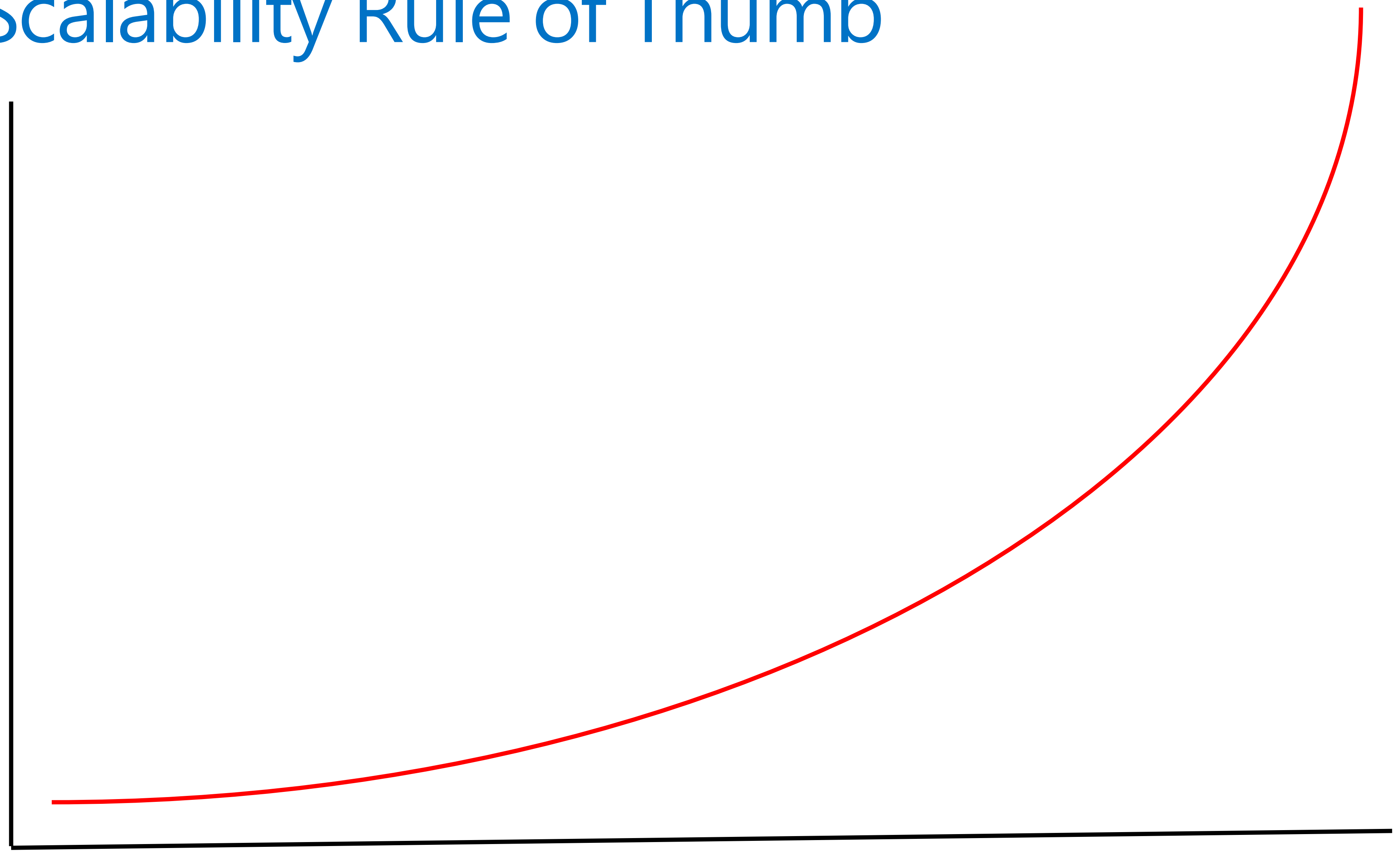
Scalability Rule of Thumb

10 → 100 - 1,000

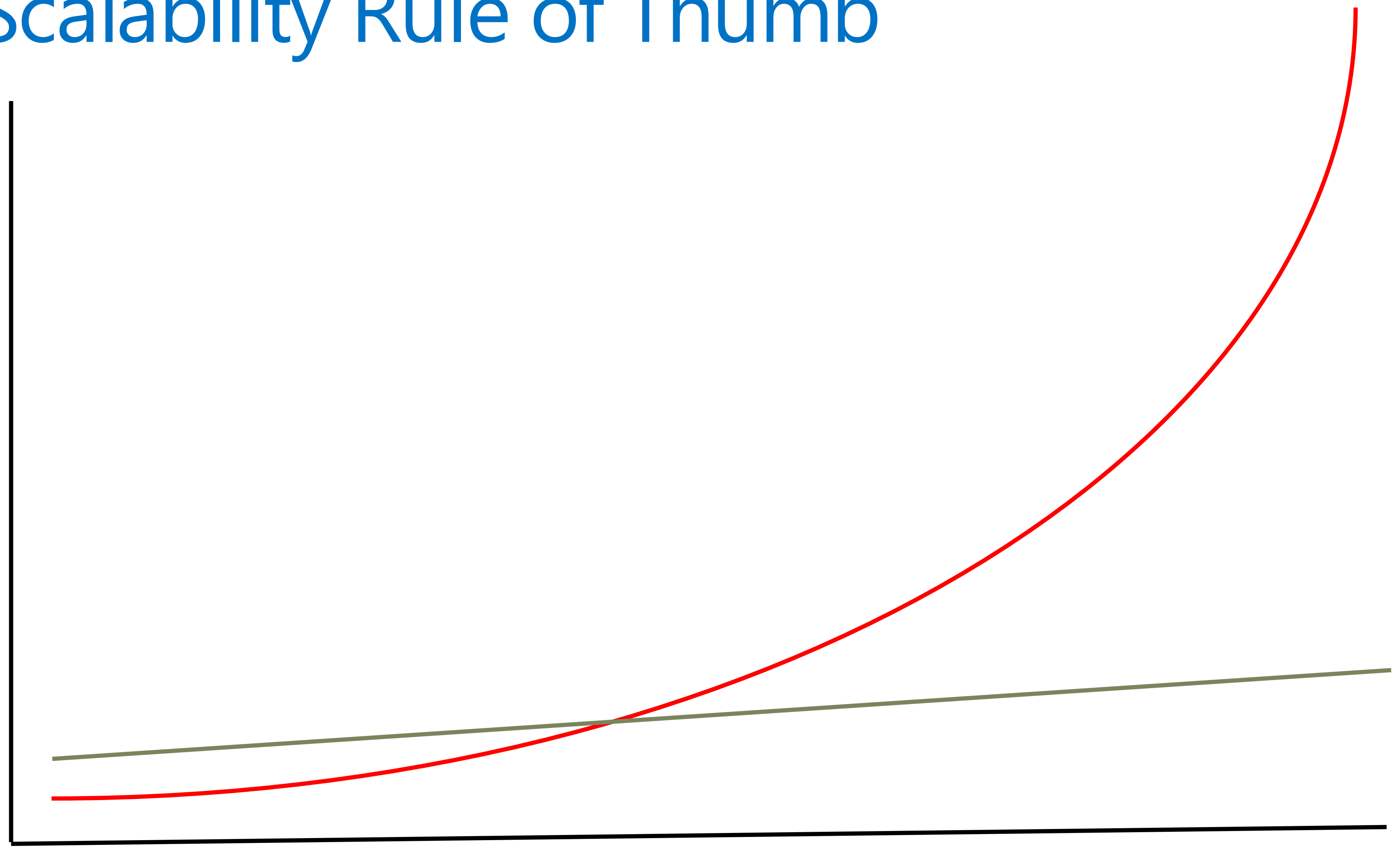
100 → 1,000 - 10,000

1,000 → 10,000 – 100,000

Scalability Rule of Thumb



Scalability Rule of Thumb



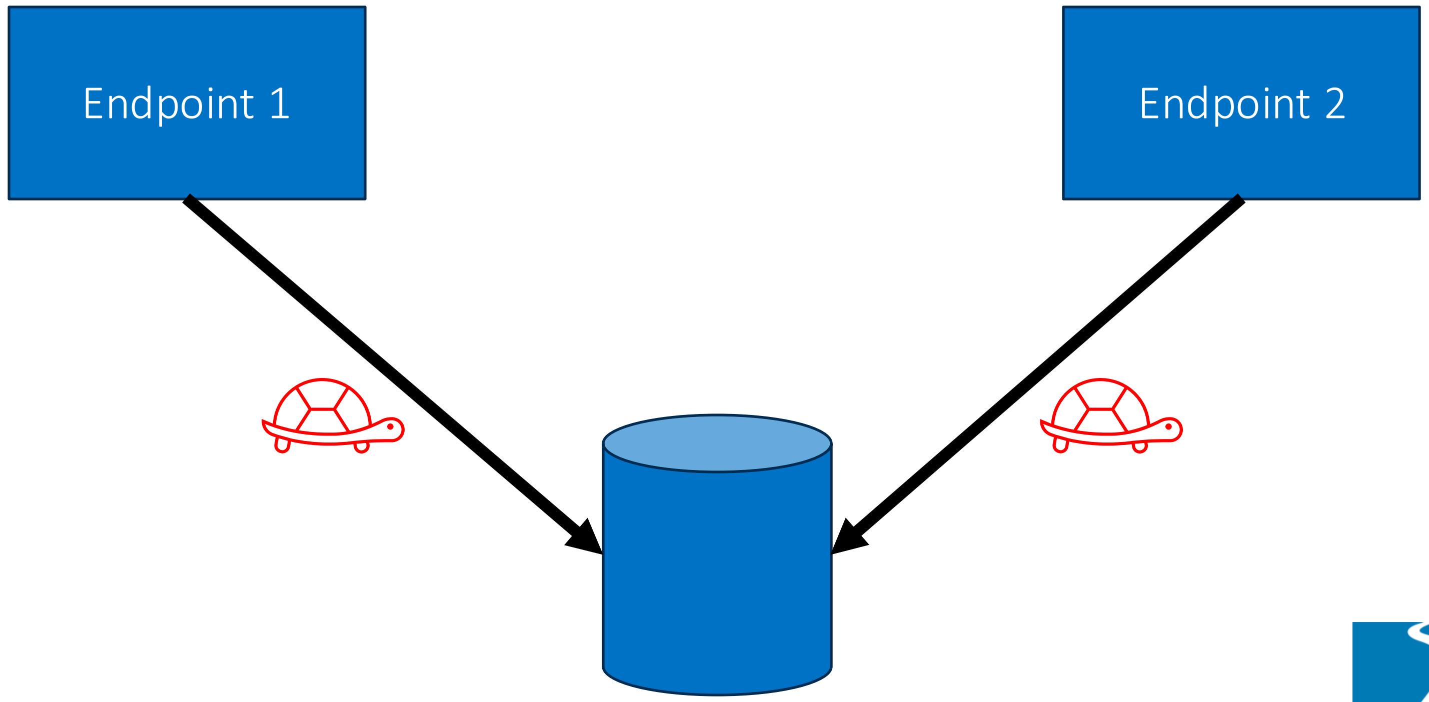
Premature Optimization

Over-Engineering Type 5

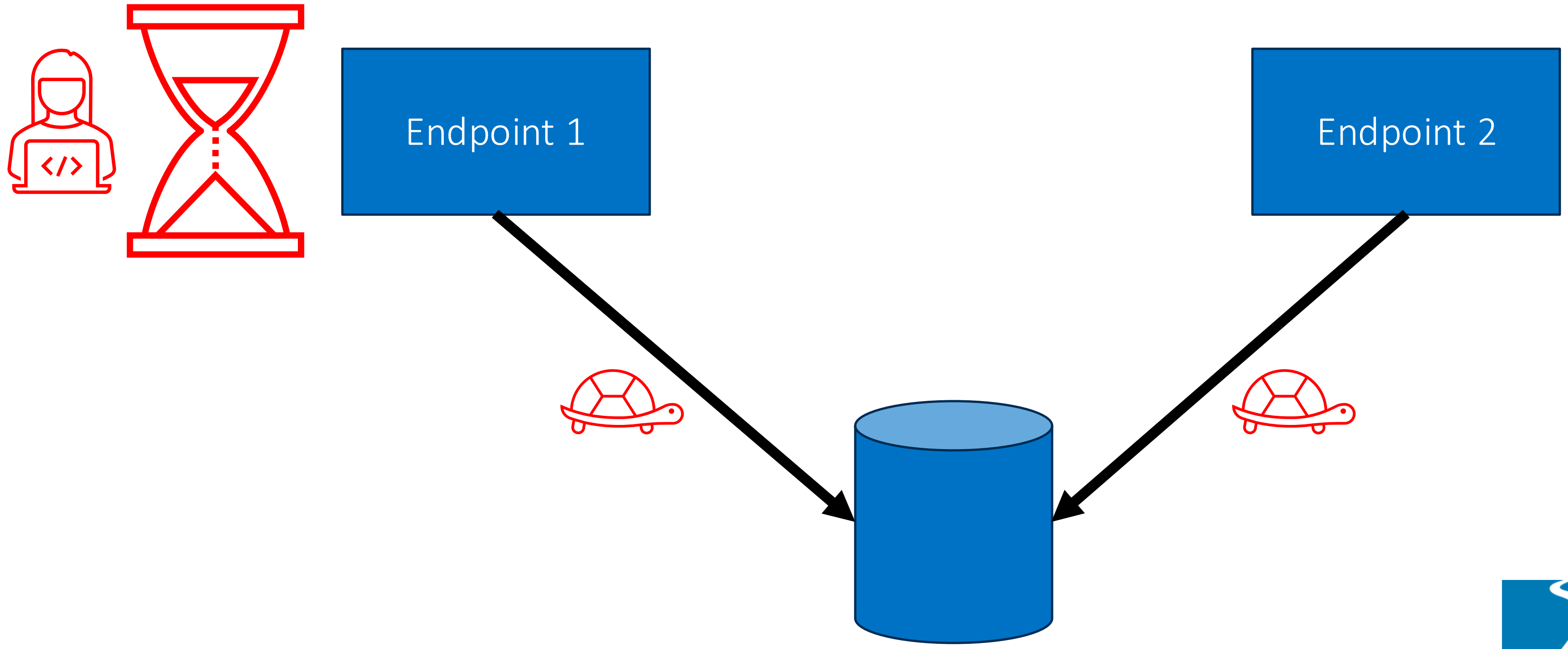
Develop Useful Features
Build For Billions of Users
JAKE-CLARK.TUMBLR
imgflip.com

TRAILHEAD TECHNOLOGY PARTNERS

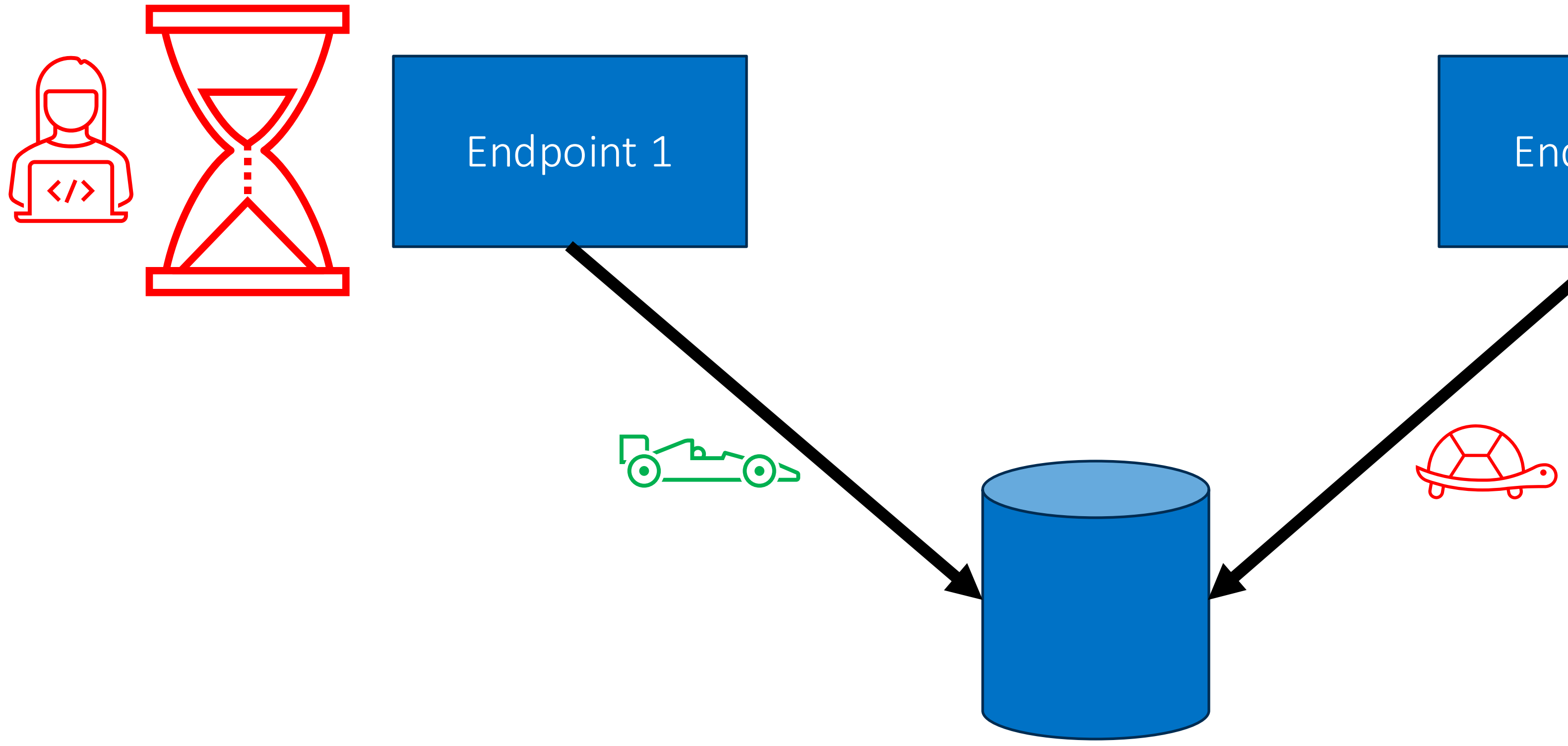
Premature Optimization



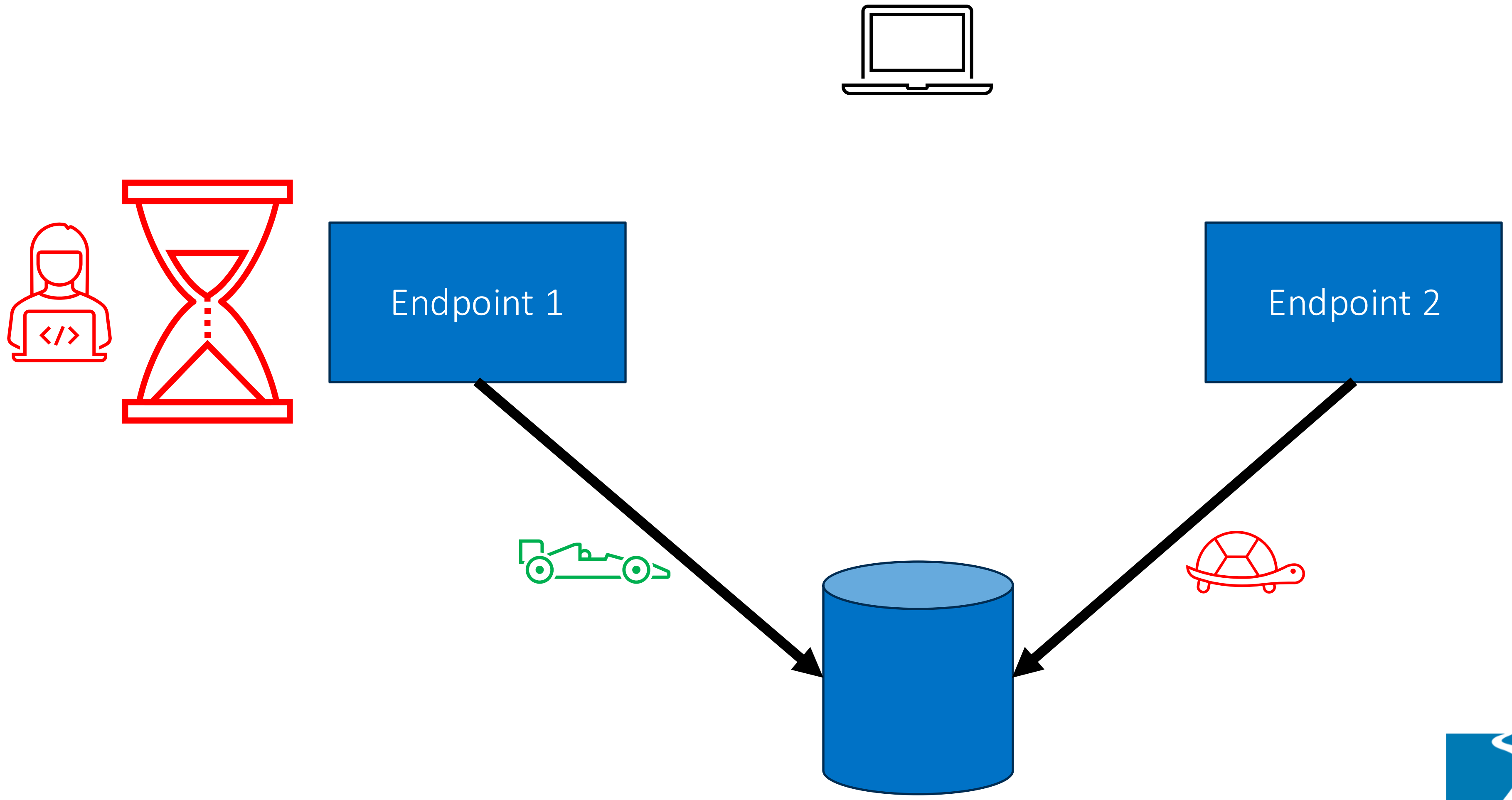
Premature Optimization



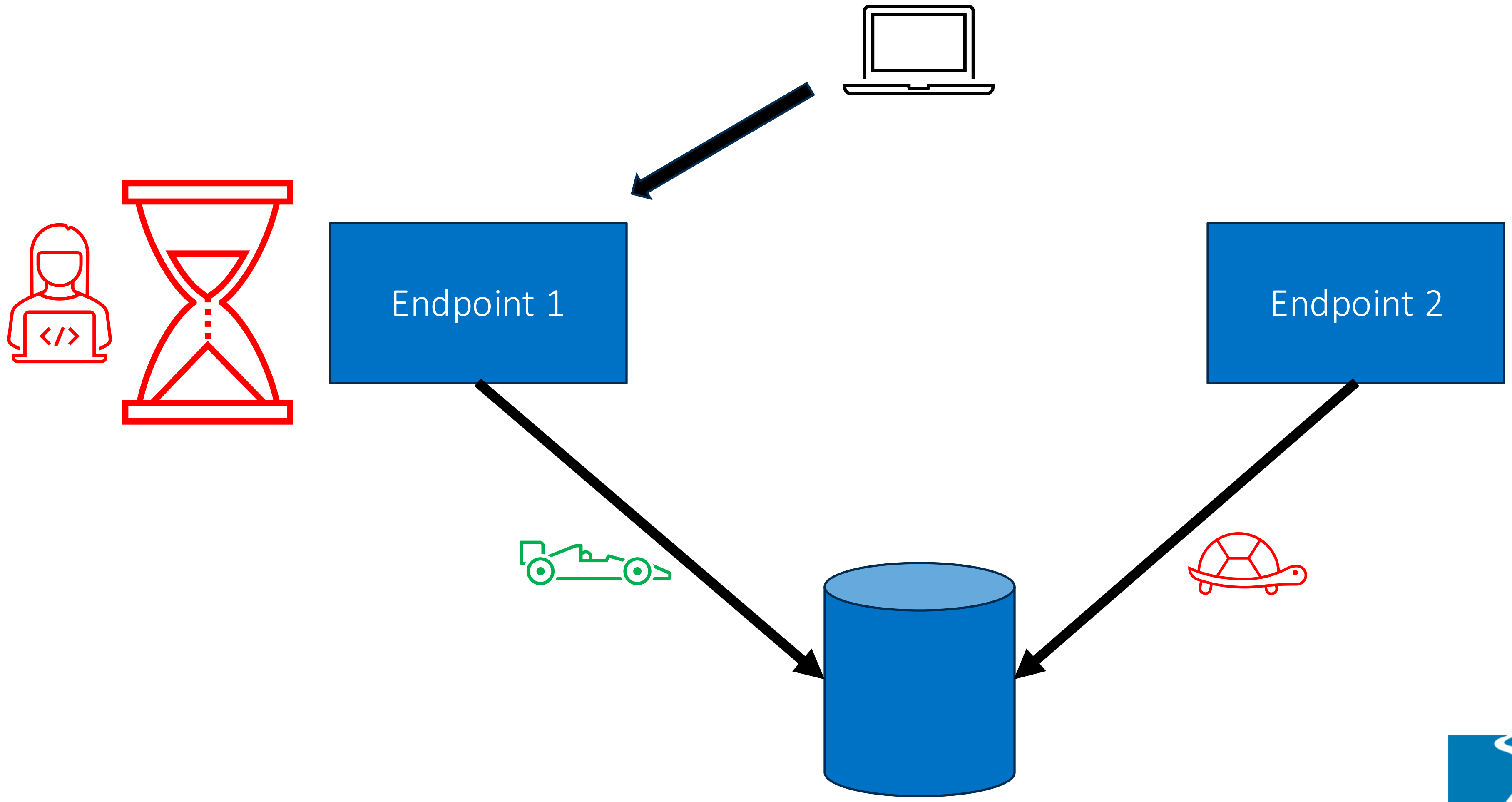
Premature Optimization



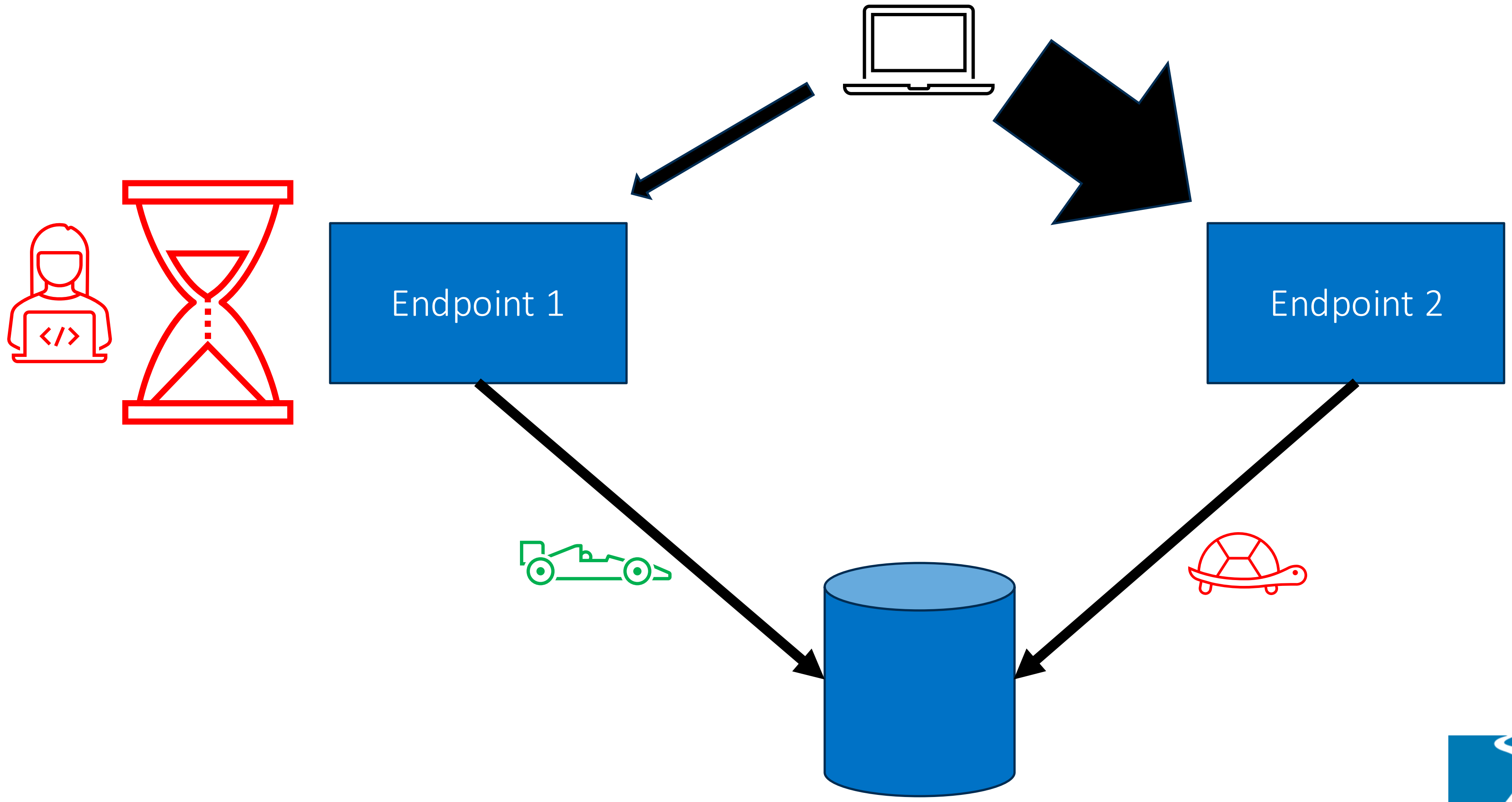
Premature Optimization



Premature Optimization



Premature Optimization

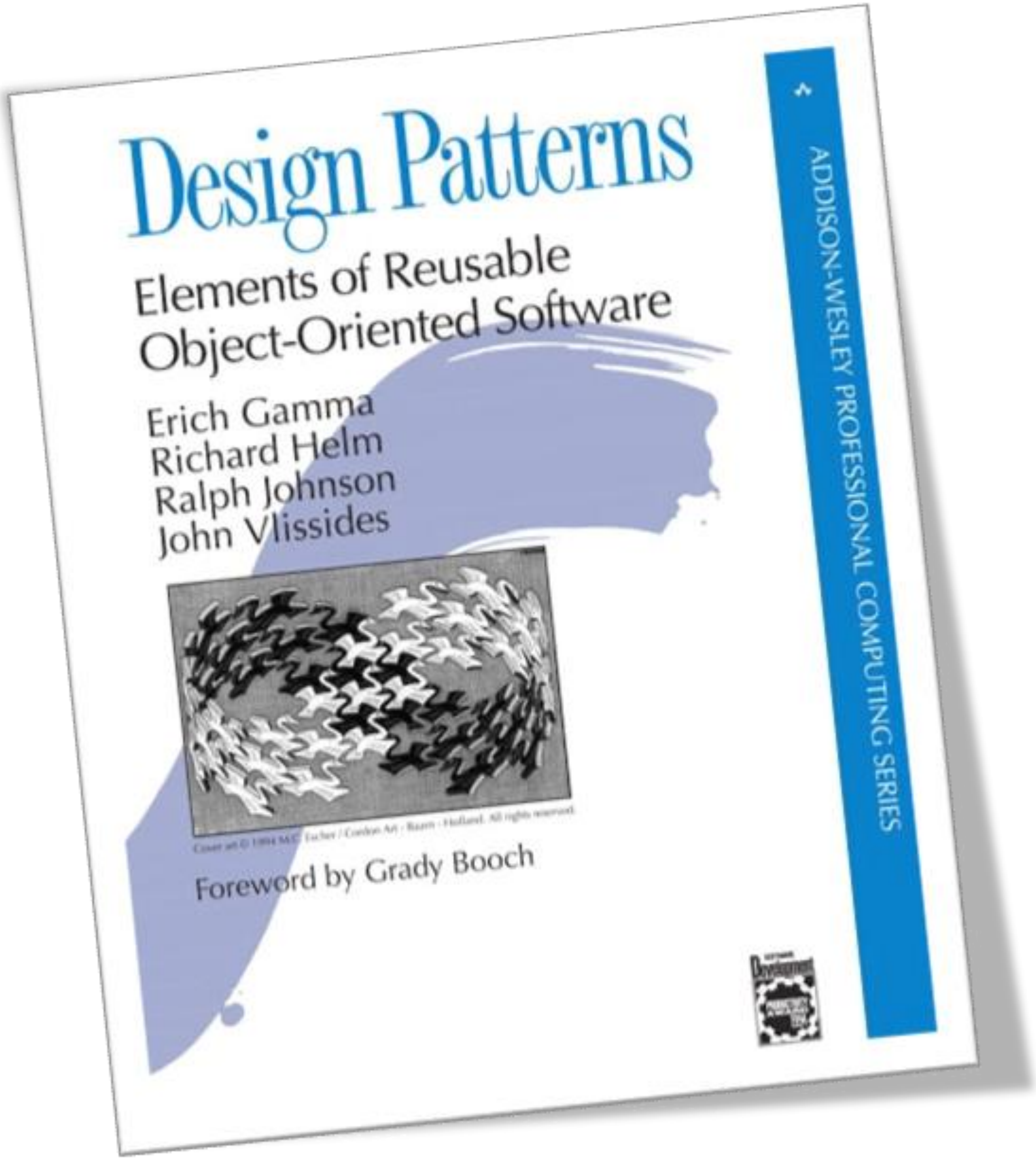


Overuse of Design Patterns

Over-Engineering Type 6

Design Patterns

- Abstract Factory
- Builder
- Factory Method
- Object Pool
- Prototype
- Singleton
- Adapter
- Bridge
- Composite
- Decorator
- Facade
- Flyweight
- Private Class Data
- Proxy
- Chain of responsibility
- Command
- Interpreter
- Iterator
- Mediator
- Memento
- Null Object
- Observer
- State
- Strategy
- Template method
- Visitor



Design Patterns

Elements of Reusable Object-Oriented Software

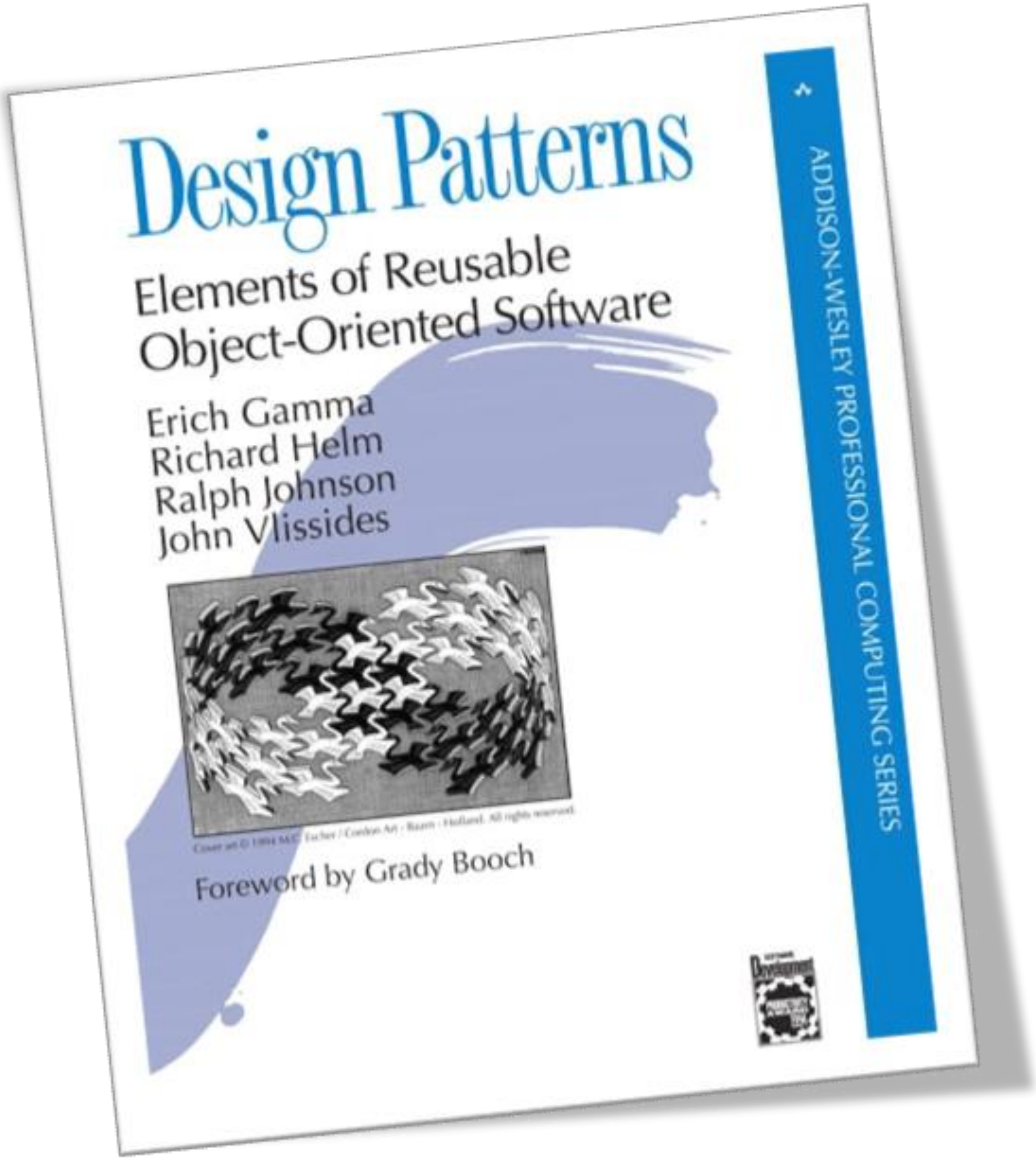
Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides

Cover art © 1994 M.C. Escher / Cordon Art - Baarn - Holland. All rights reserved.

Foreword by Grady Booch

ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

TRAILHEAD
TECHNOLOGY PARTNERS



Design Patterns

Elements of Reusable Object-Oriented Software

Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides

Cover art © 1994 M.C. Escher / Cordon Art - Baarn - Holland. All rights reserved.

Foreword by Grady Booch

ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

TRAILHEAD
TECHNOLOGY PARTNERS

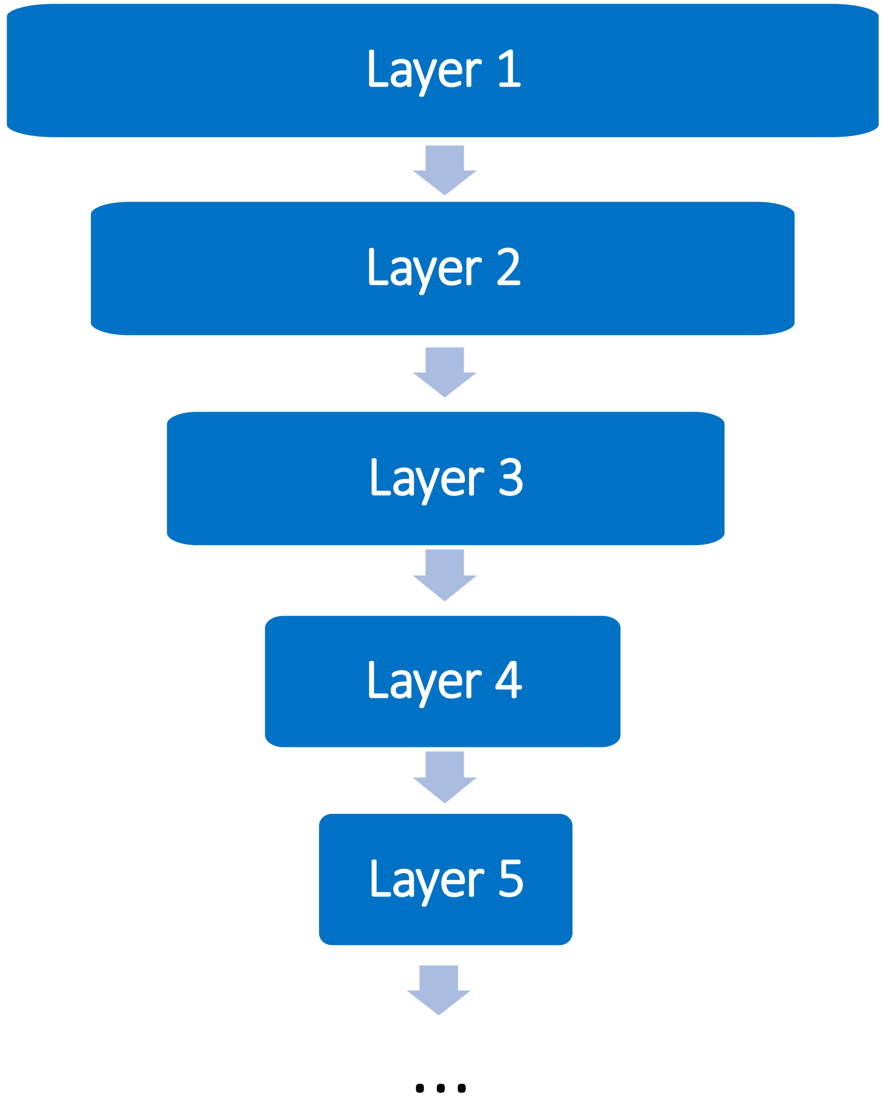
“If the only tool **you have is a hammer**, it is tempting to treat **everything** as if it were a nail.”

- Abraham Maslow

Lasagna Architecture

Over-Engineering Type 7

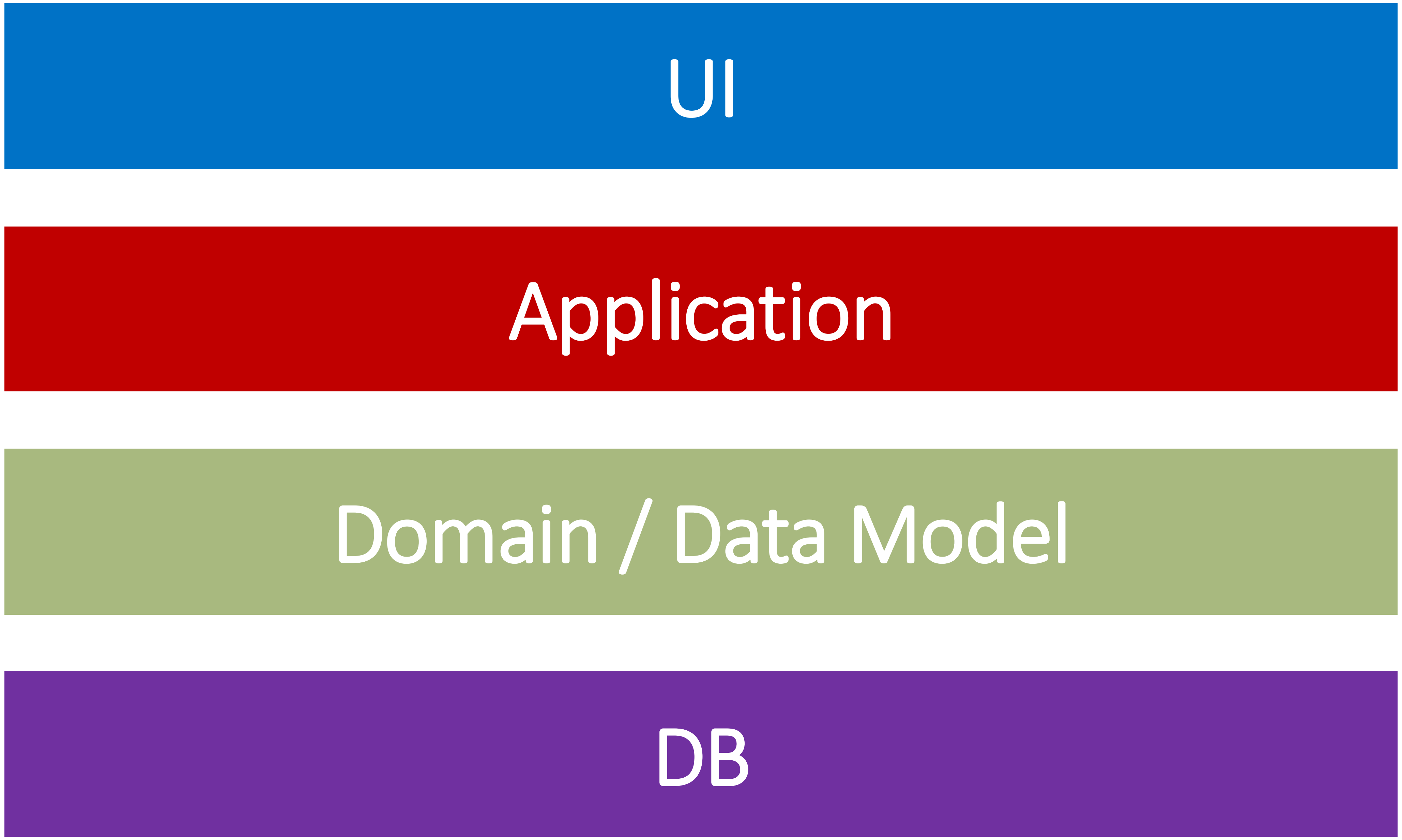
Too Many Layers

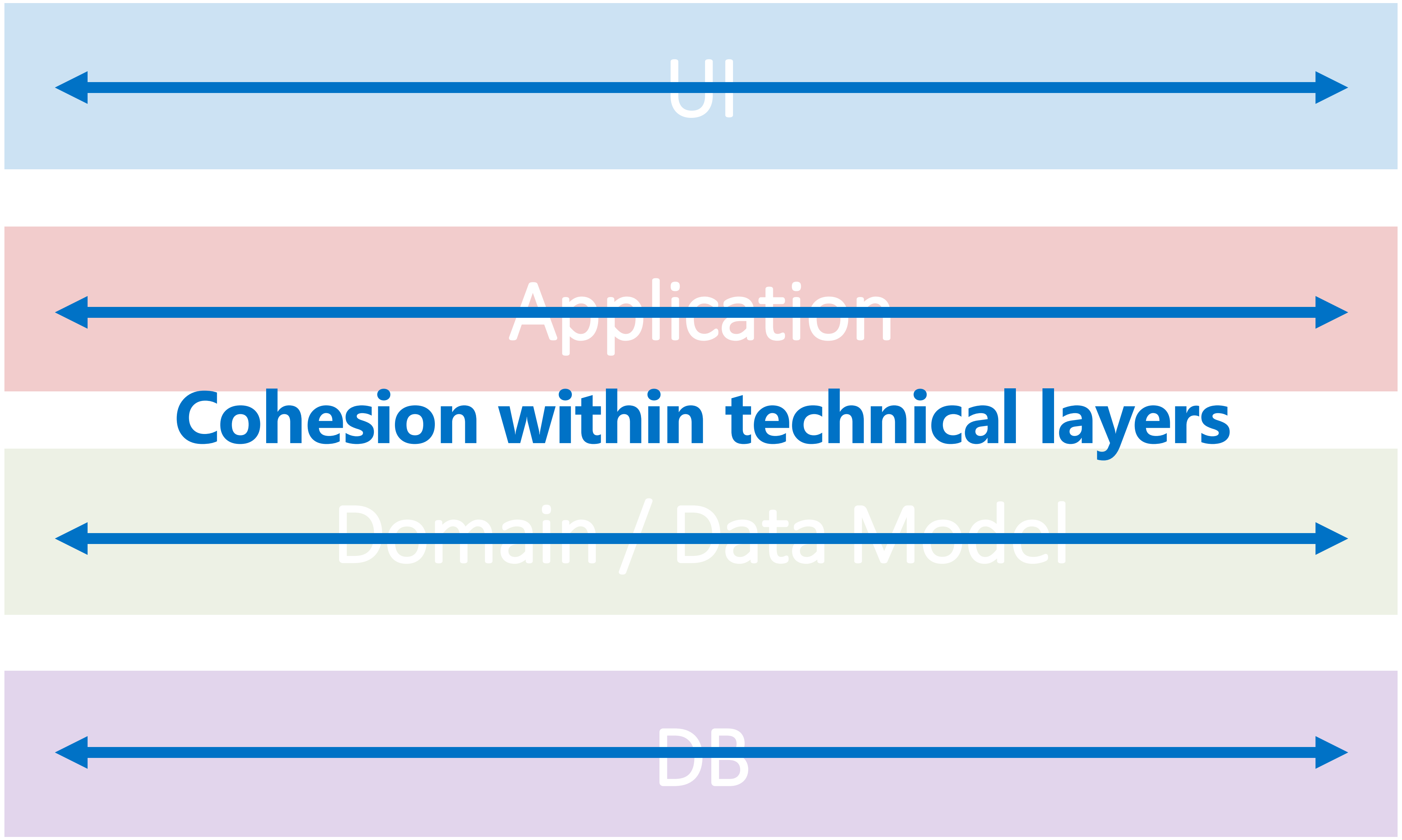


Vertical Slice Architecture

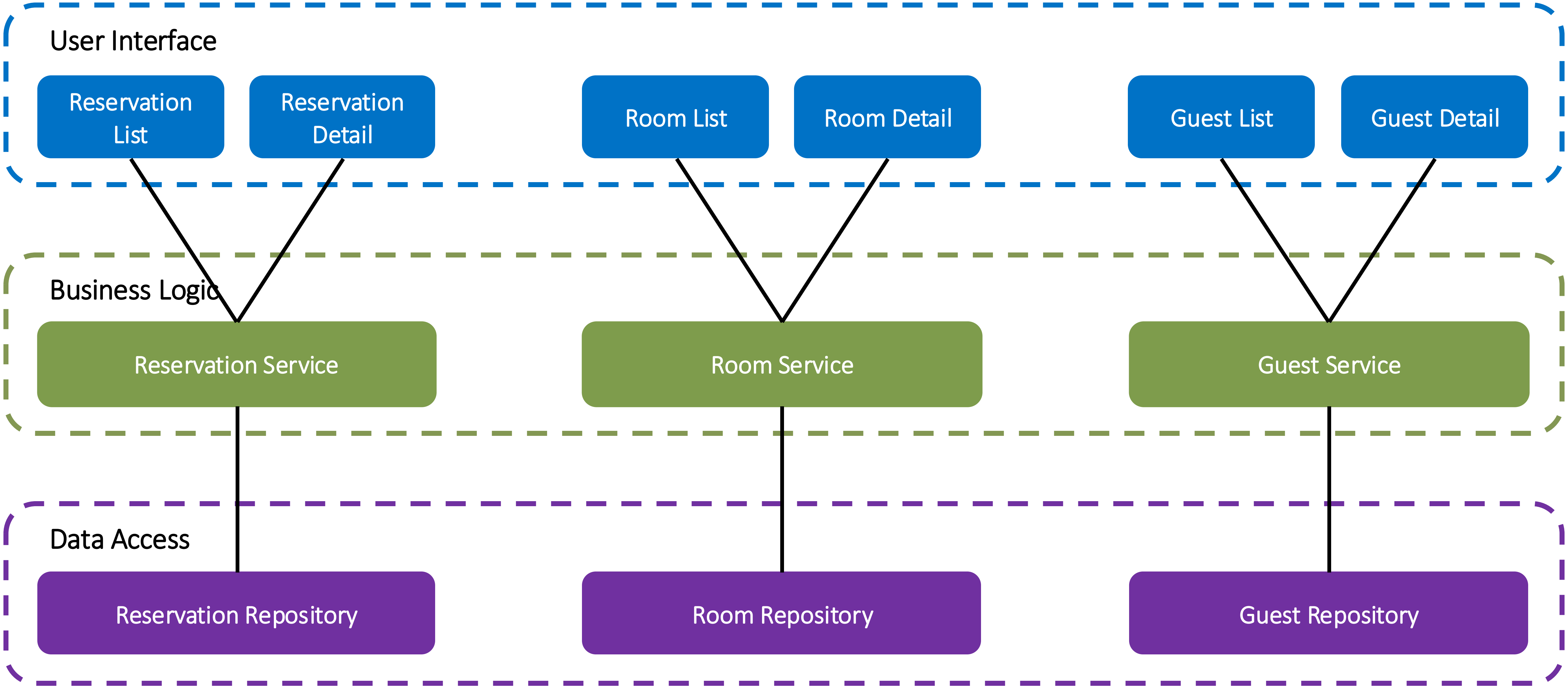
Vertical slice architecture is a **software architecture** pattern that **organizes** code **by features** or use cases **instead of technical** concerns.

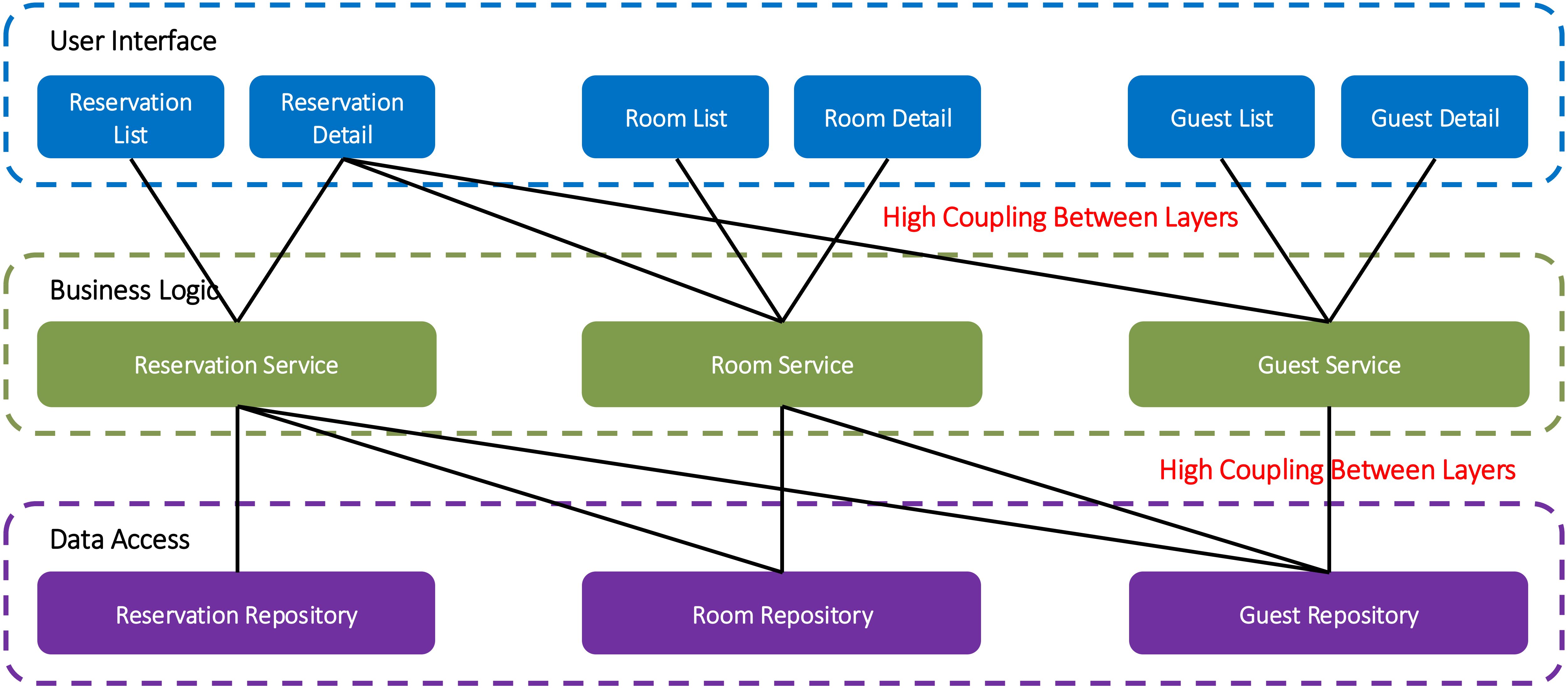
No more layers for their own sake!


UI
Application
Domain / Data Model
DB

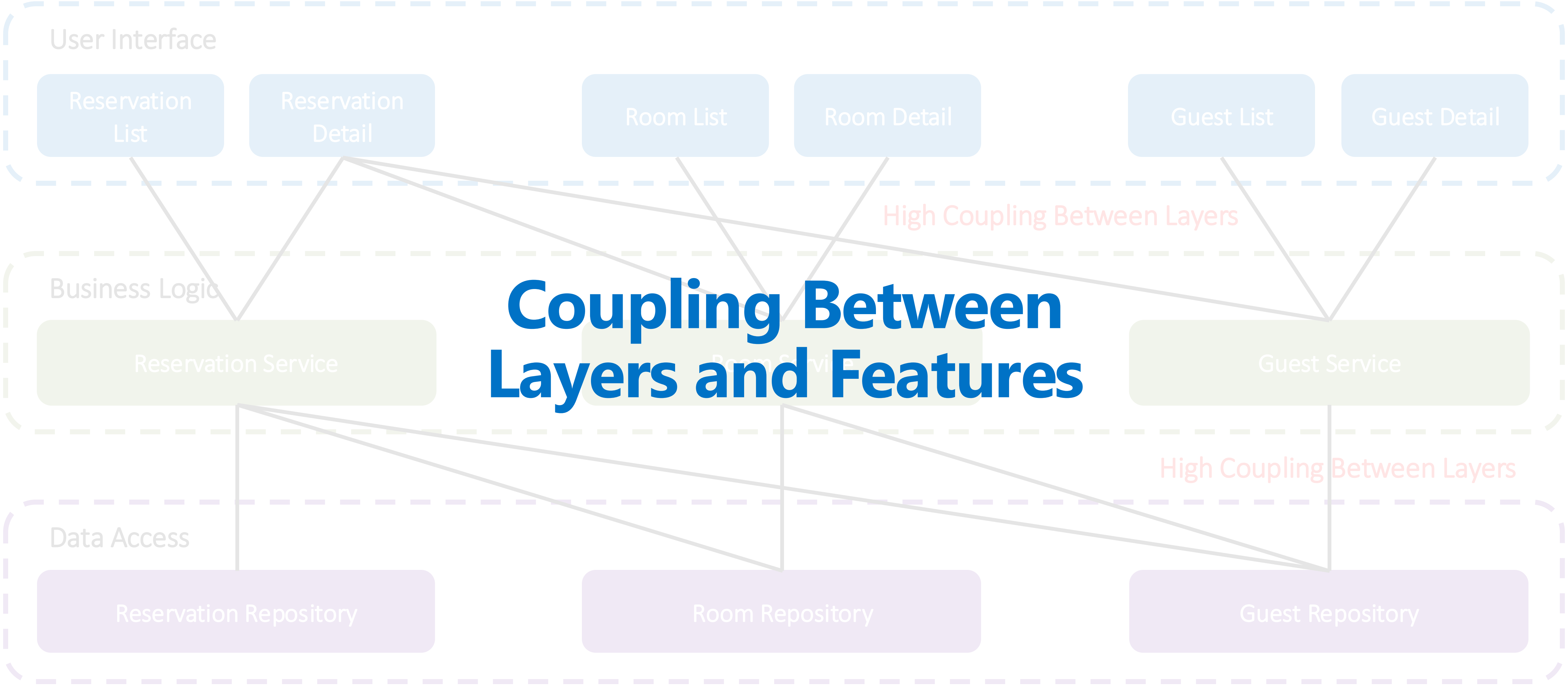


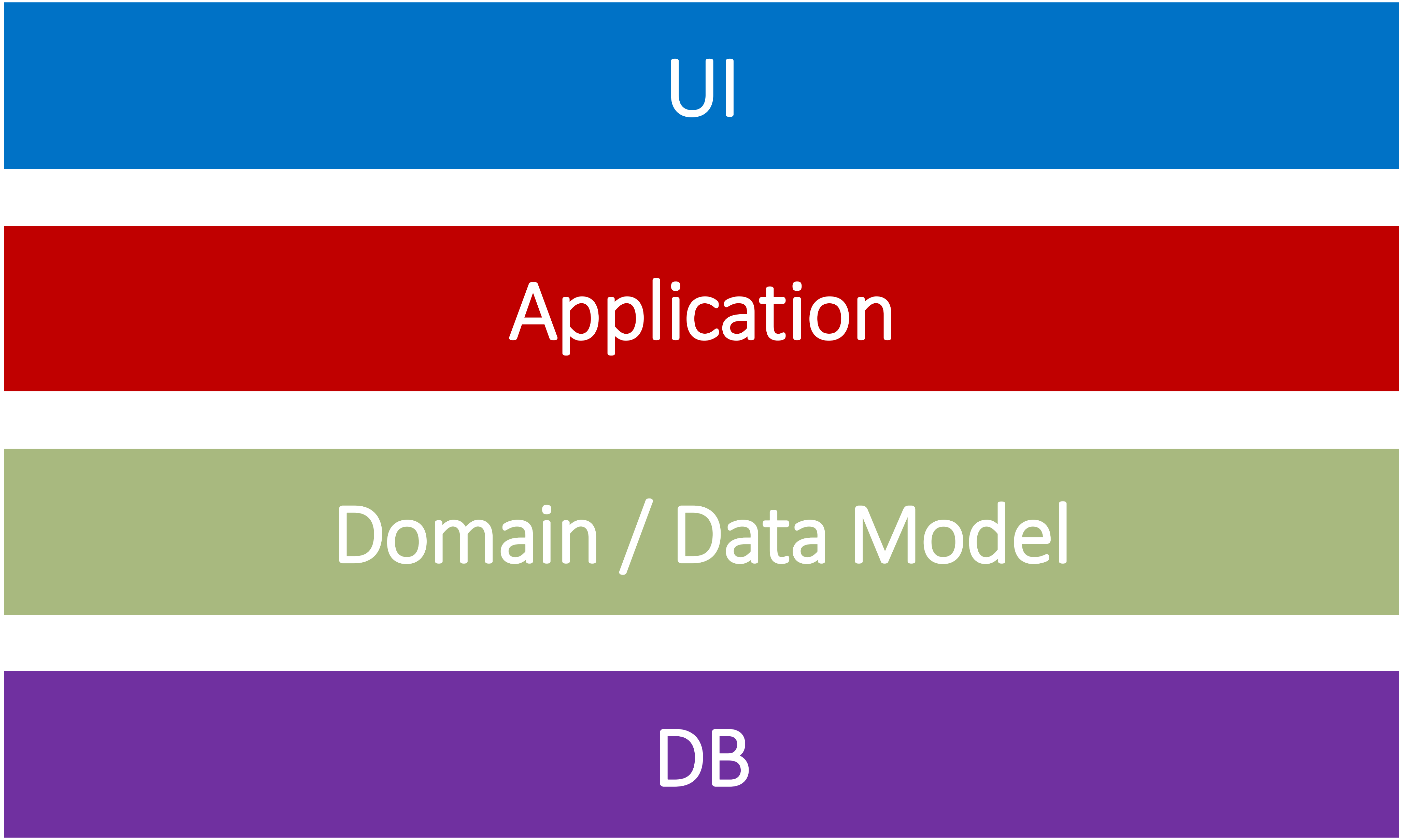
Cohesion within technical layers

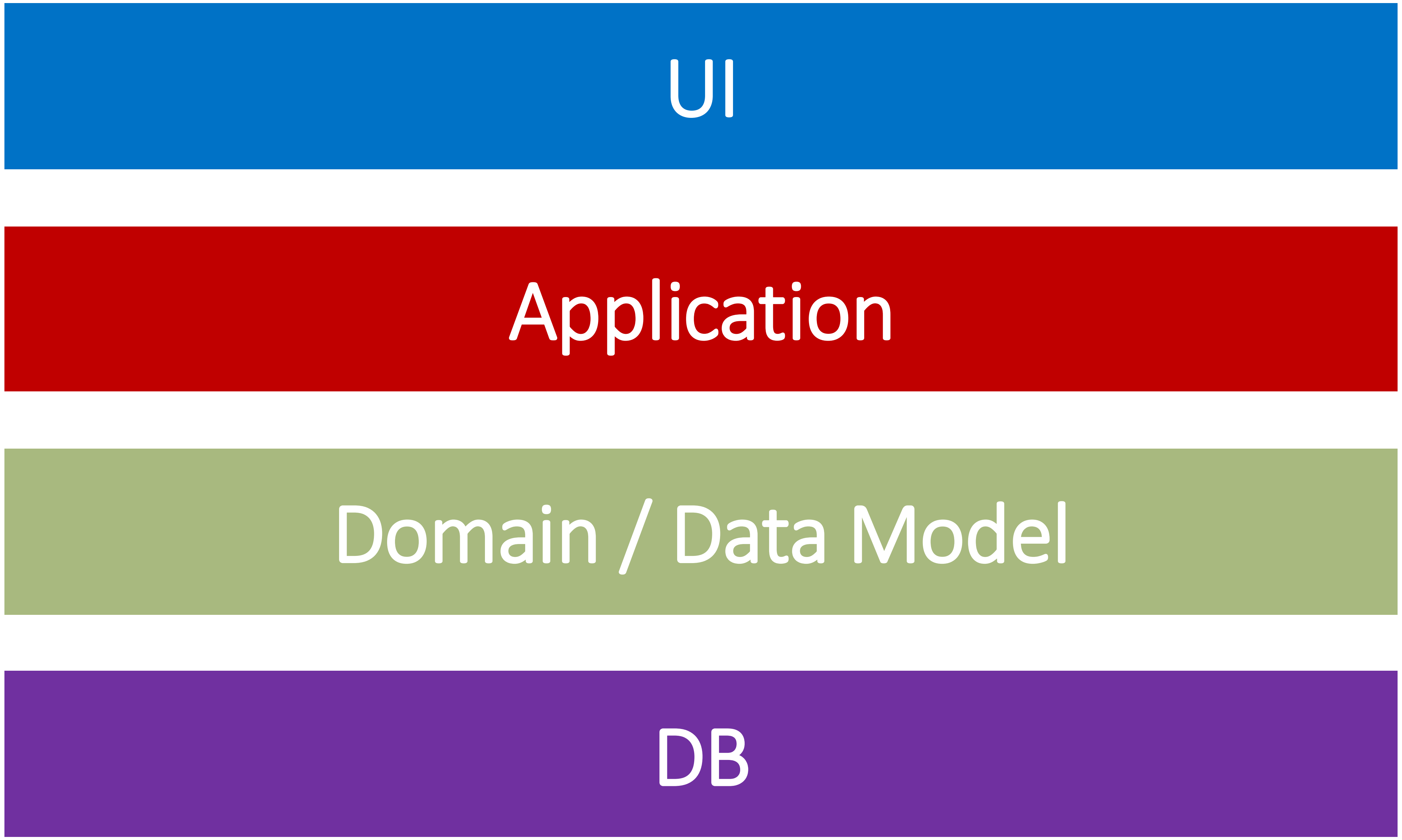

User Interface
Reservation List
Reservation Detail
Room List
Room Detail
Guest List
Guest Detail
Business Logic
Reservation Service
Room Service
Guest Service
Data Access
Reservation Repository
Room Repository
Guest Repository

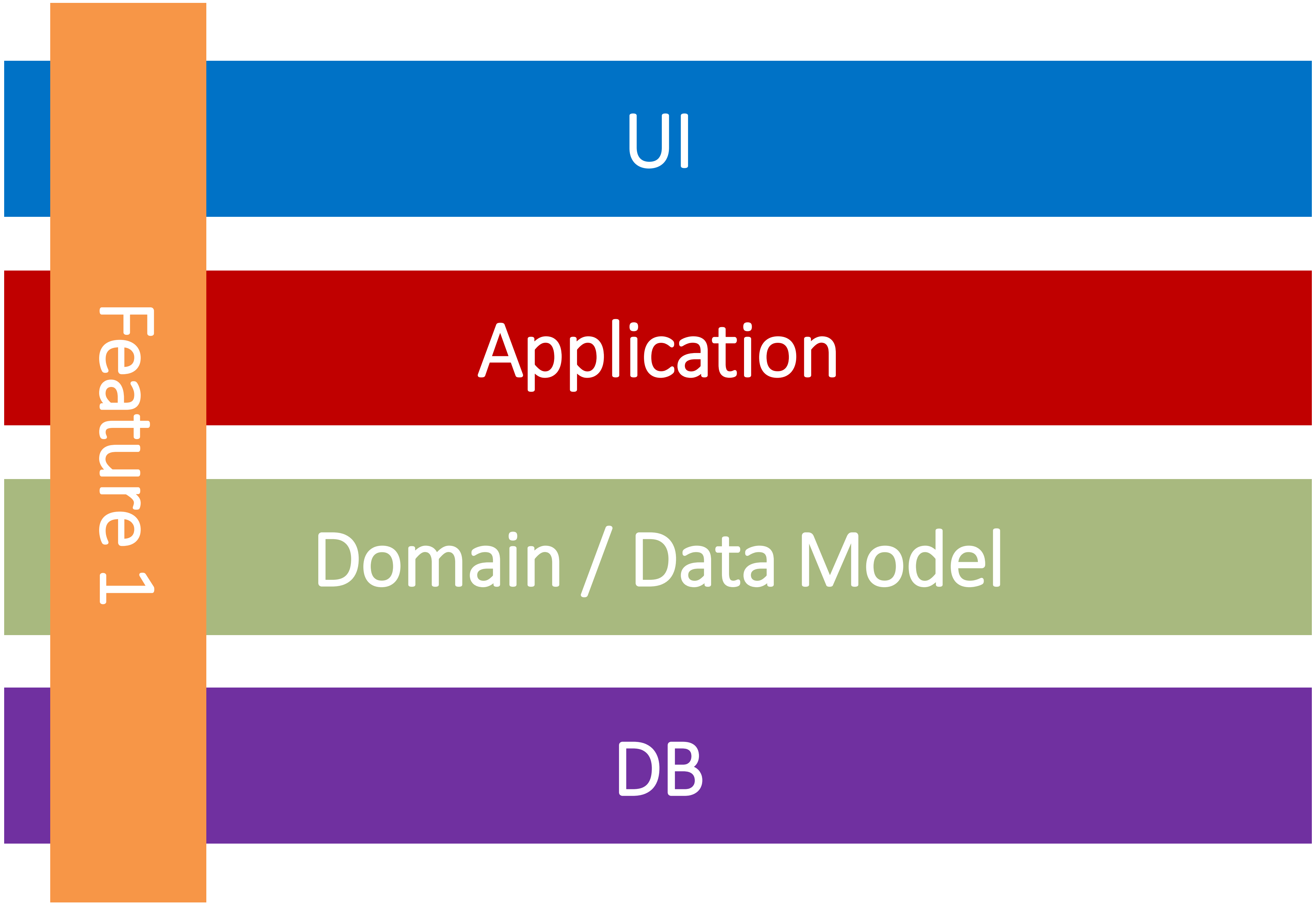

User Interface
Reservation List
Reservation Detail
Room List
Room Detail
Guest List
Guest Detail
High Coupling Between Layers
Business Logic
Reservation Service
Room Service
Guest Service
High Coupling Between Layers
Data Access
Reservation Repository
Room Repository
Guest Repository

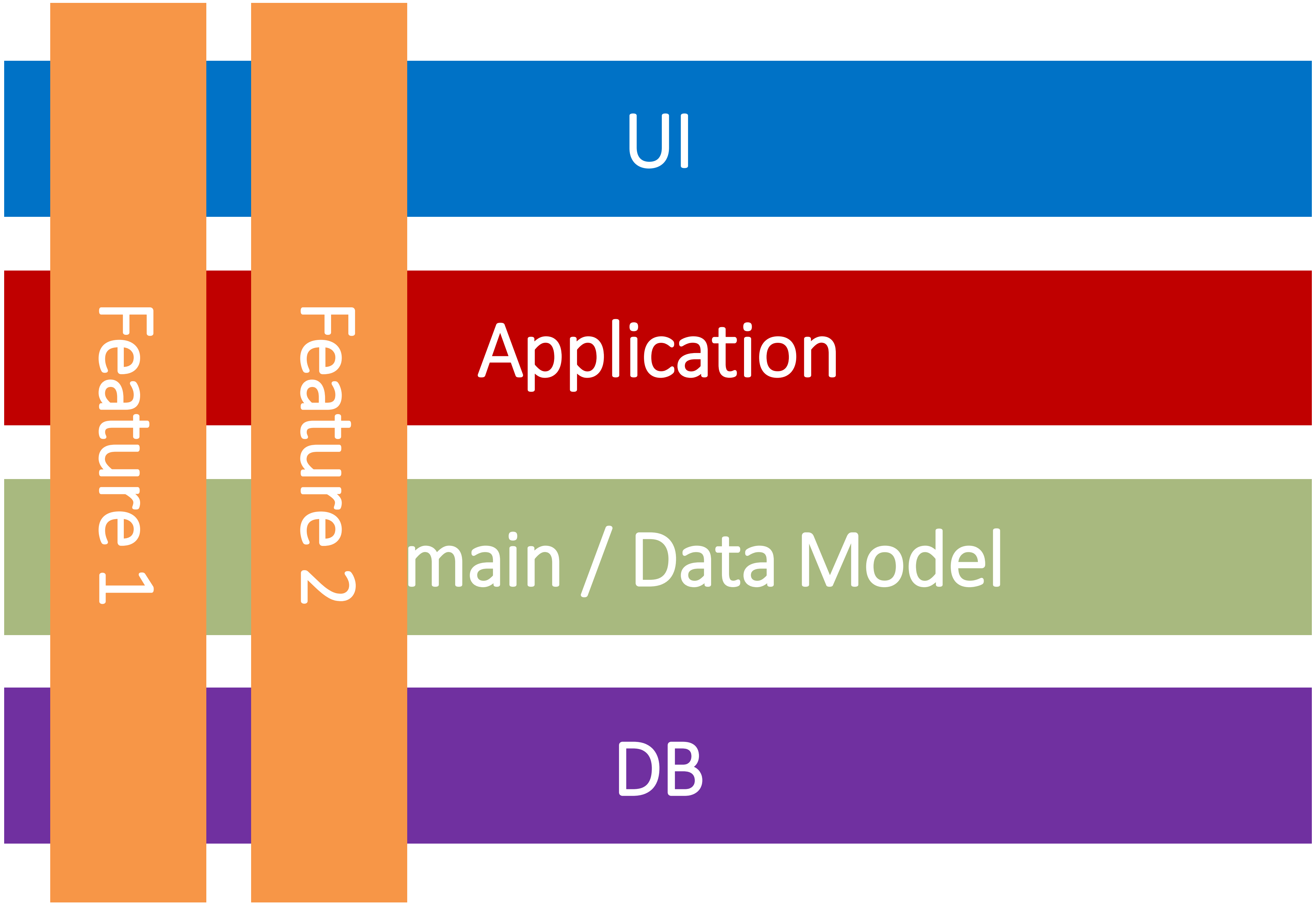
Coupling Between Layers and Features

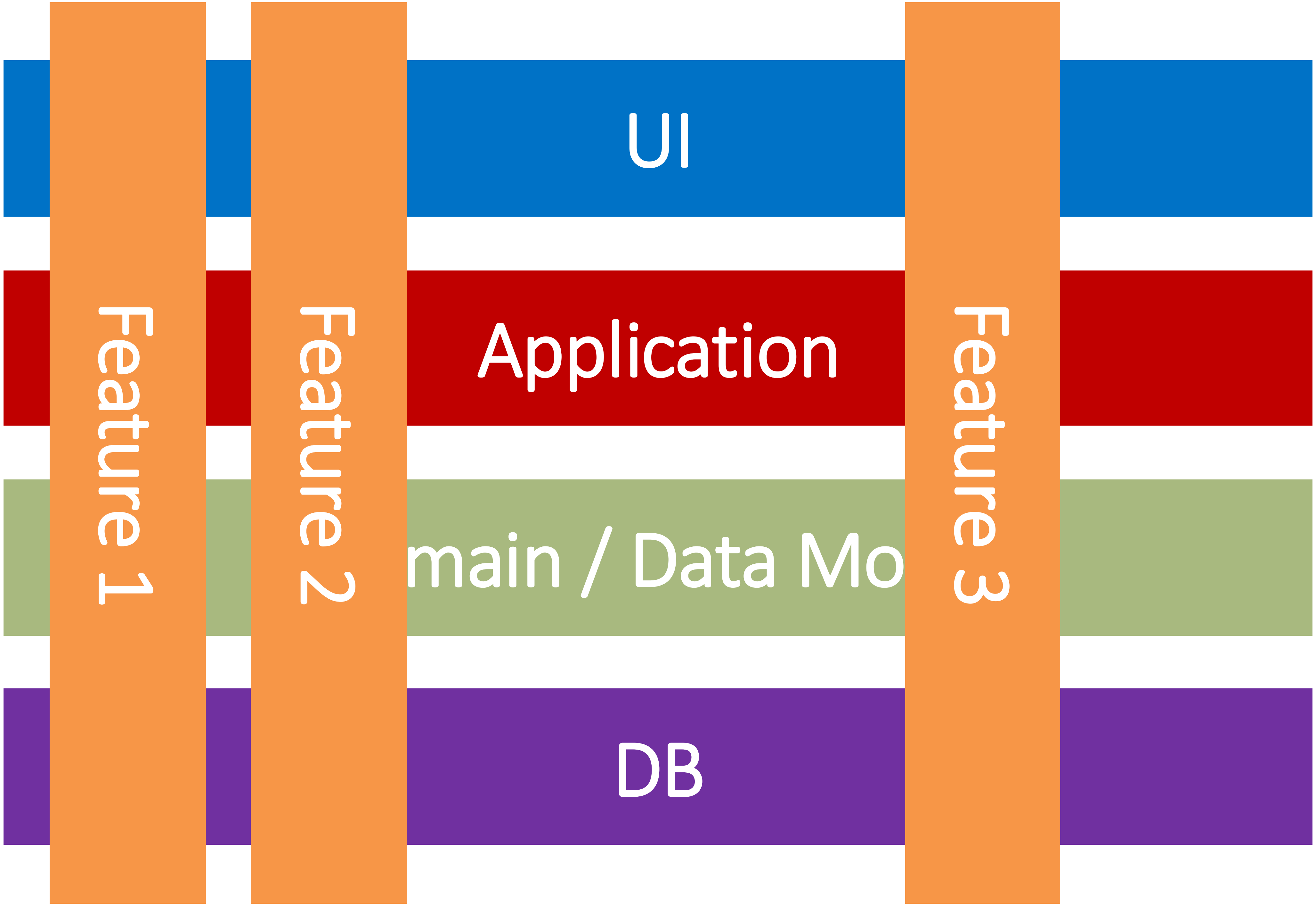


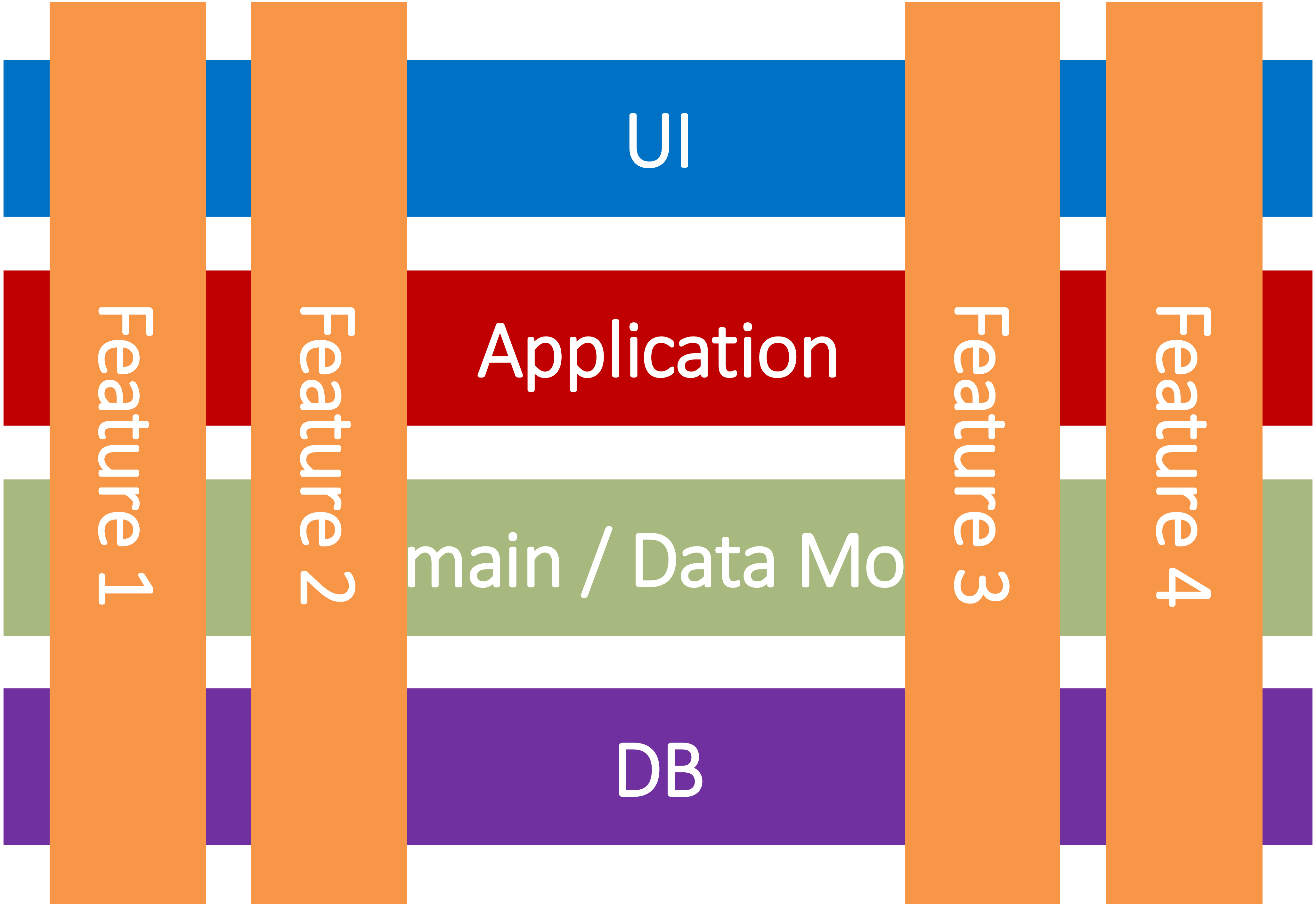

UI
Application
Domain / Data Model
DB

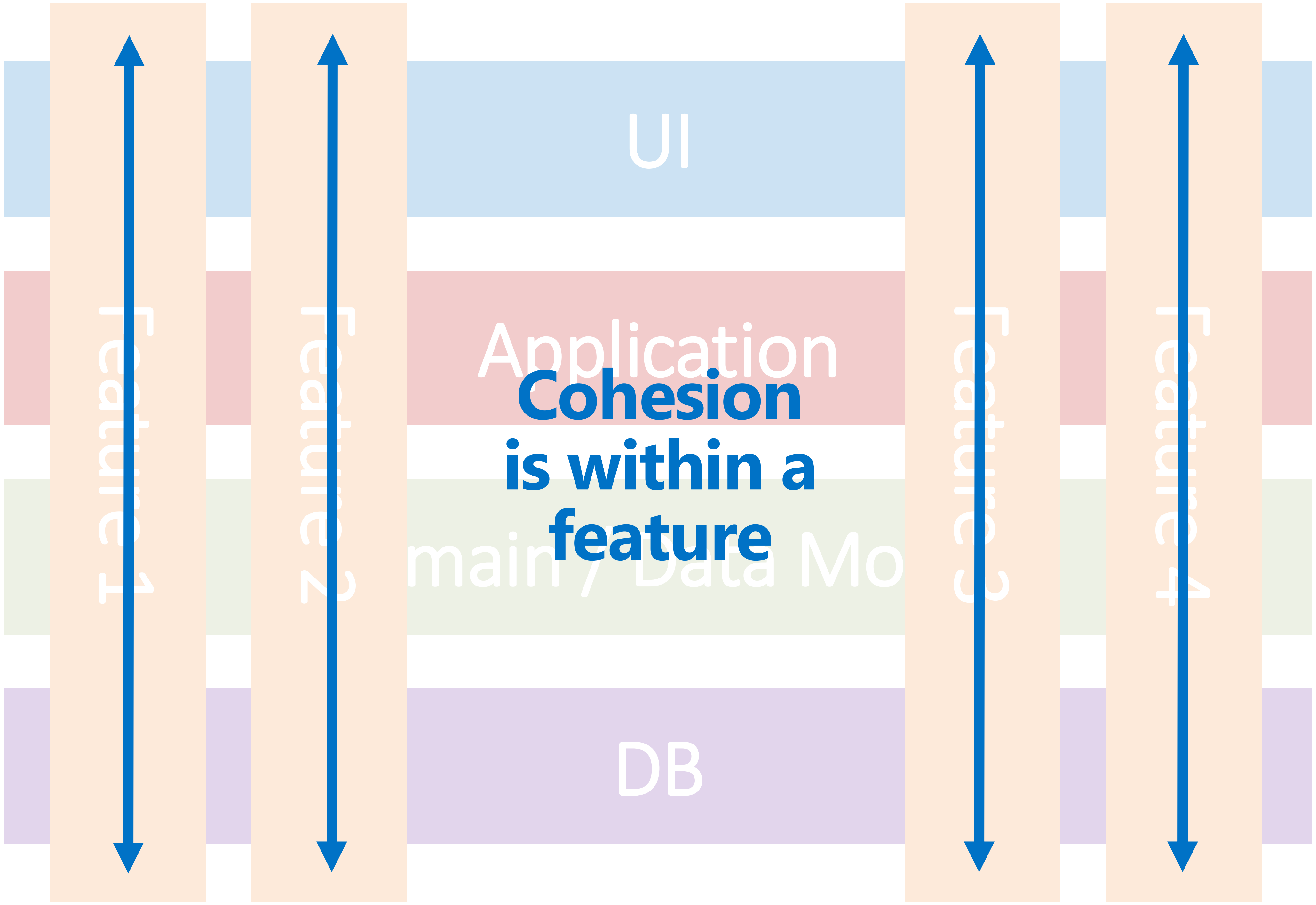

UI
Application
Domain / Data Model
DB


UI
Application
Domain / Data Model
DB
Feature 1

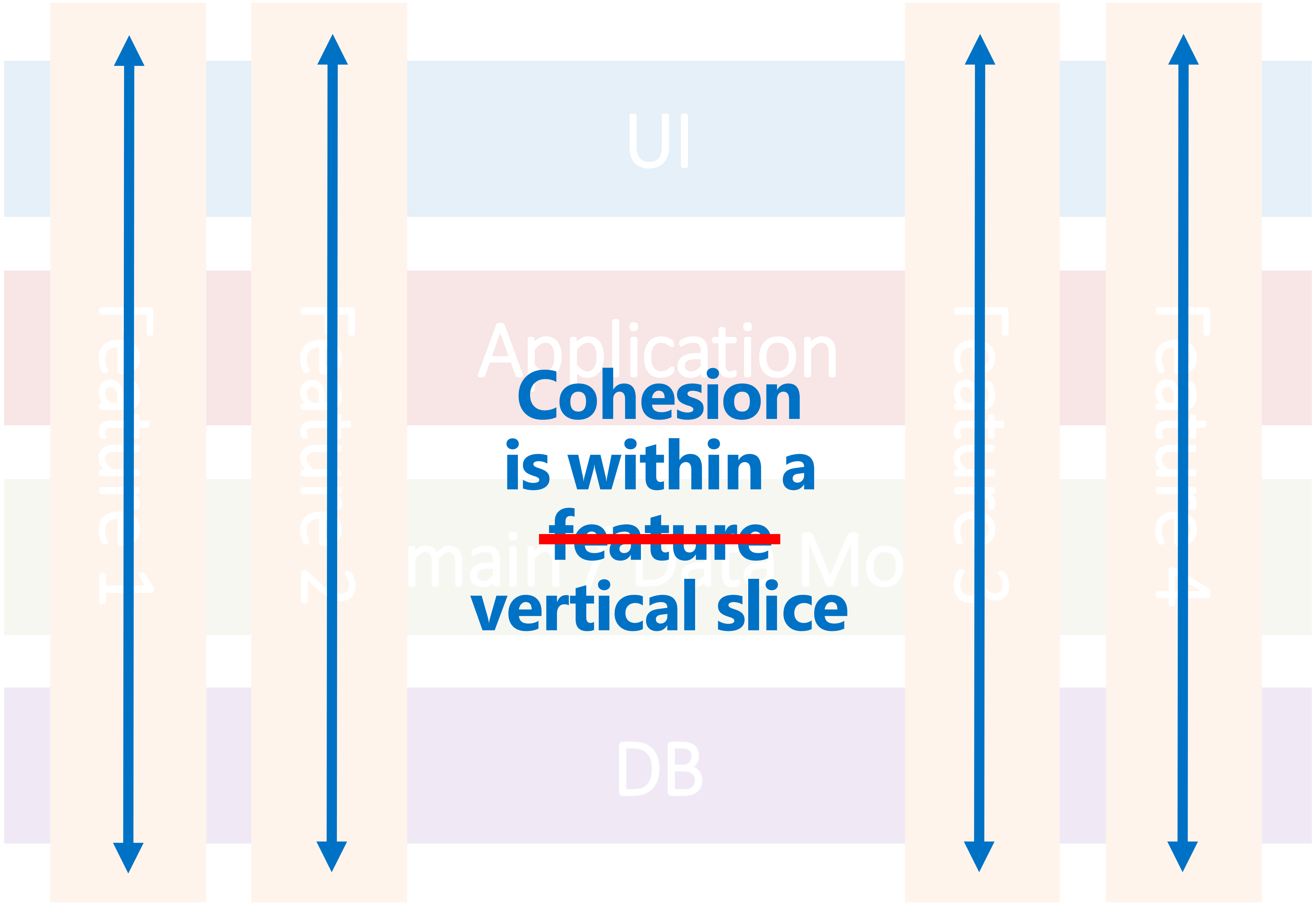

UI
Application
main / Data Model
DB
Feature 1
Feature 2


UI
Application
main / Data Mo
DB
Feature 1
Feature 2
Feature 3


UI
Application
main / Data Mo
DB
Feature 1
Feature 2
Feature 3
Feature 4



Cohesion is within a feature

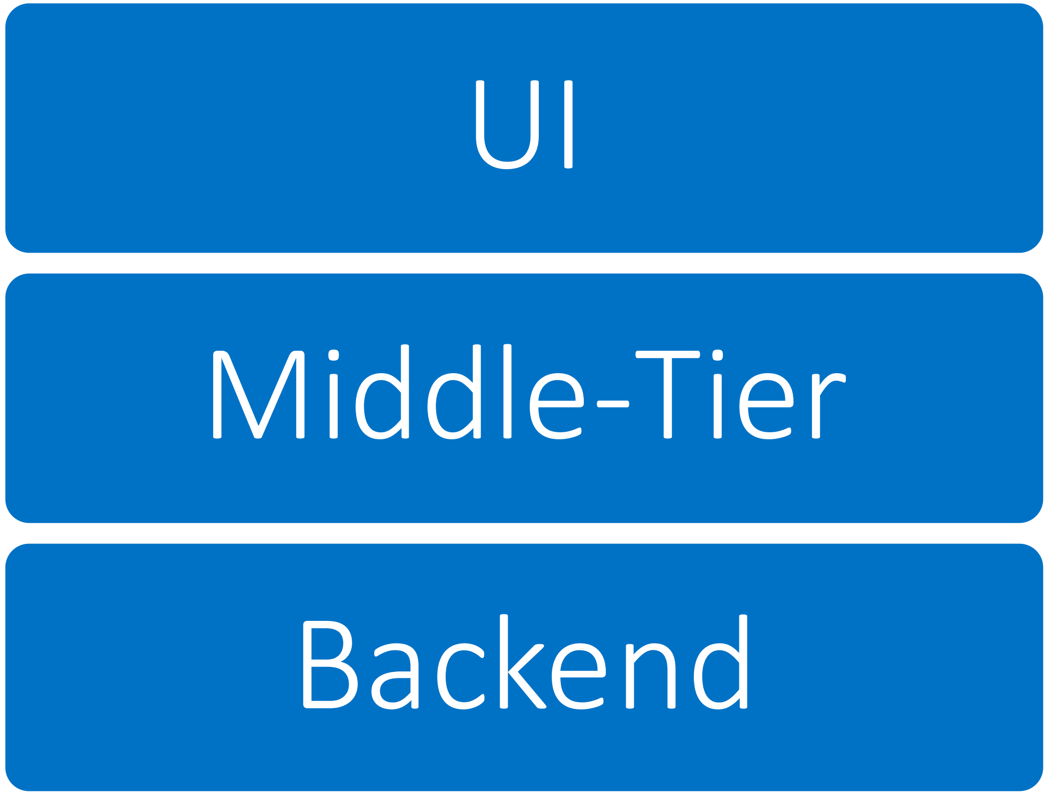


Cohesion is within a ~~feature~~ vertical slice

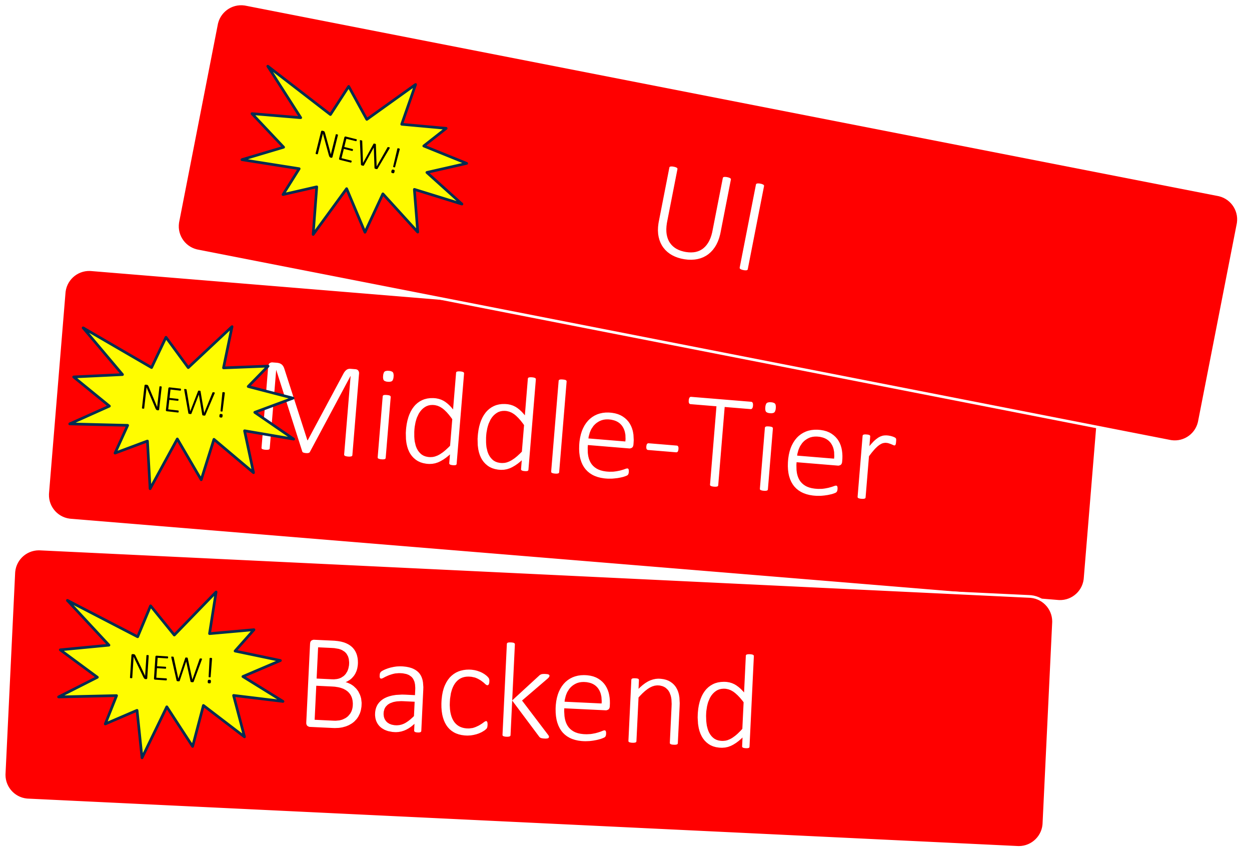
Shiny Object Syndrome

Over-Engineering Type 8

Which Stack Is More Fun?



Stuff You Already Know



All New Stuff

Rolling Your Own

Over-Engineering Type 9

Issues with Rolling Your Own

Increased Costs

Extended Development Time

New Dev Training

Reduced Testing Exposure

Issues with Rolling Your Own

Increased Costs

Extended Development Time

New Dev Training

Reduced Testing Exposure

Misapplied Libraries/Frameworks

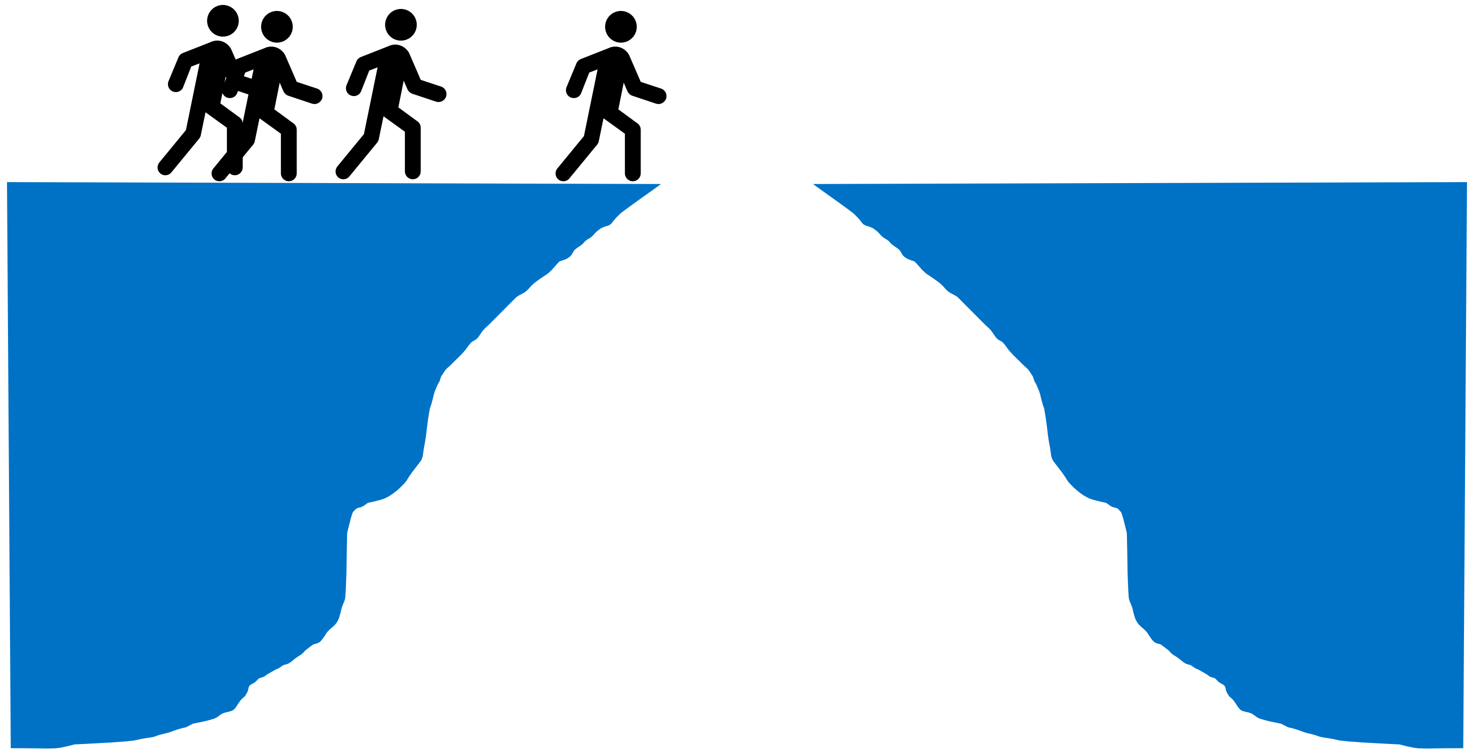
Over-Engineering Type 10

10 **Rules** To Avoid Over-Engineering

Document Your Engineering Decisions

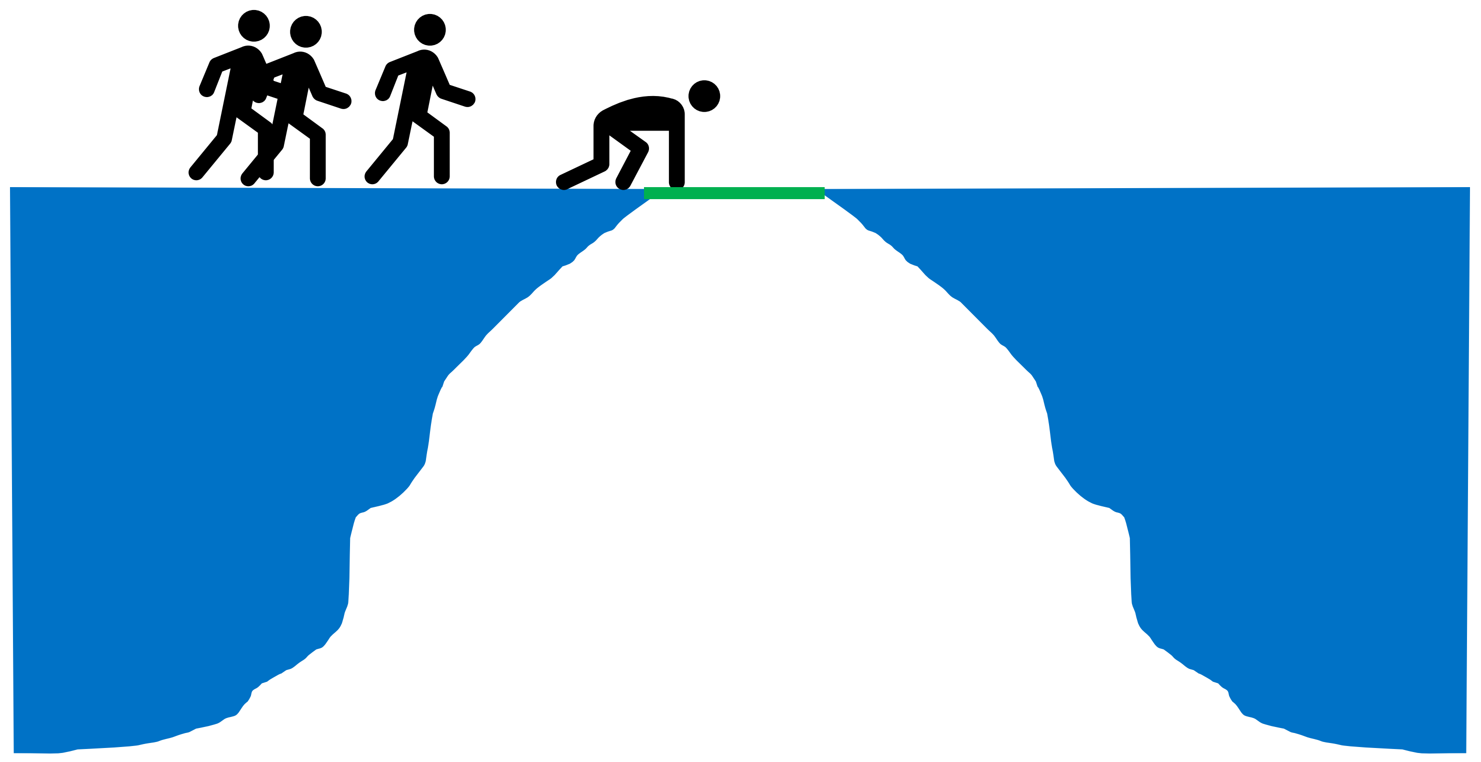
Rule 1

Avoid Mistakes In The Future

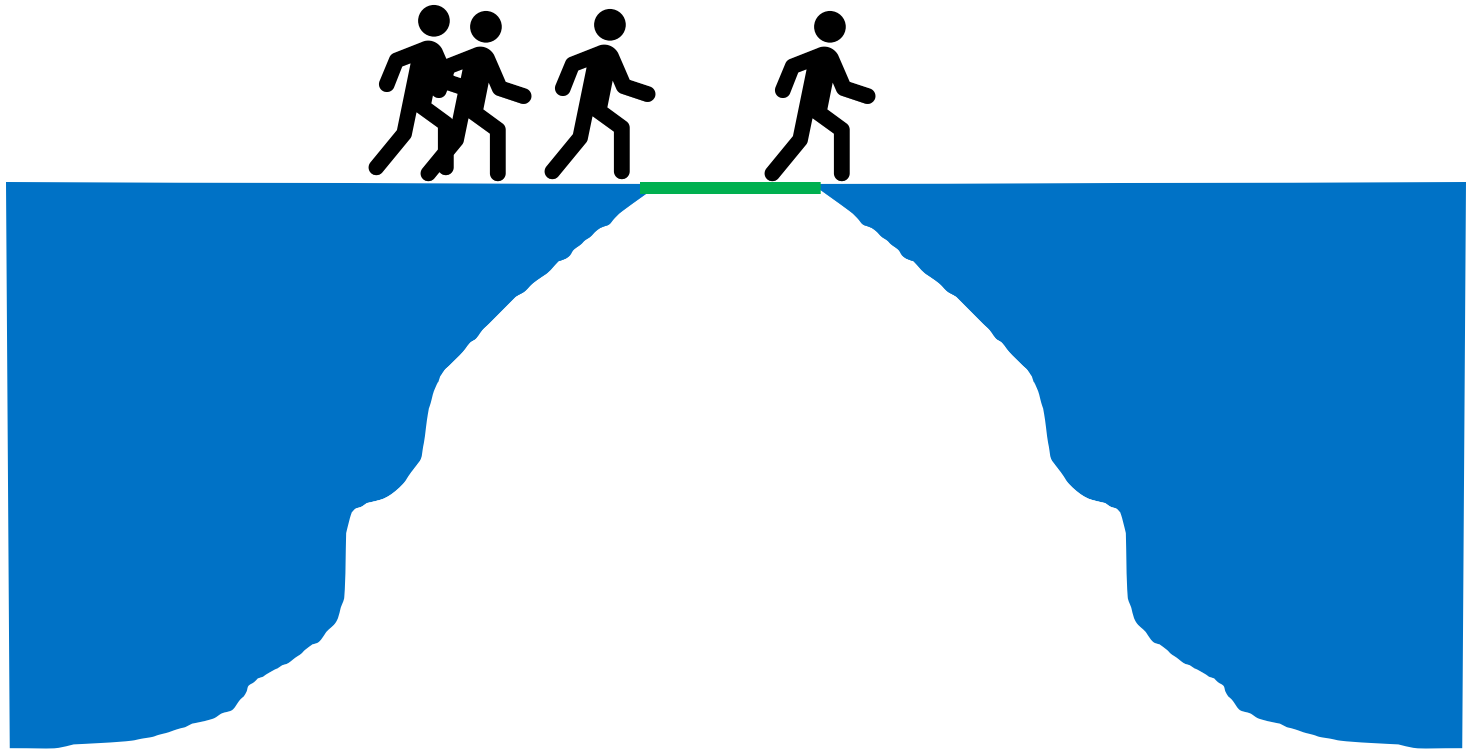


TRAILHEAD
TECHNOLOGY PARTNERS

Avoid Mistakes In The Future

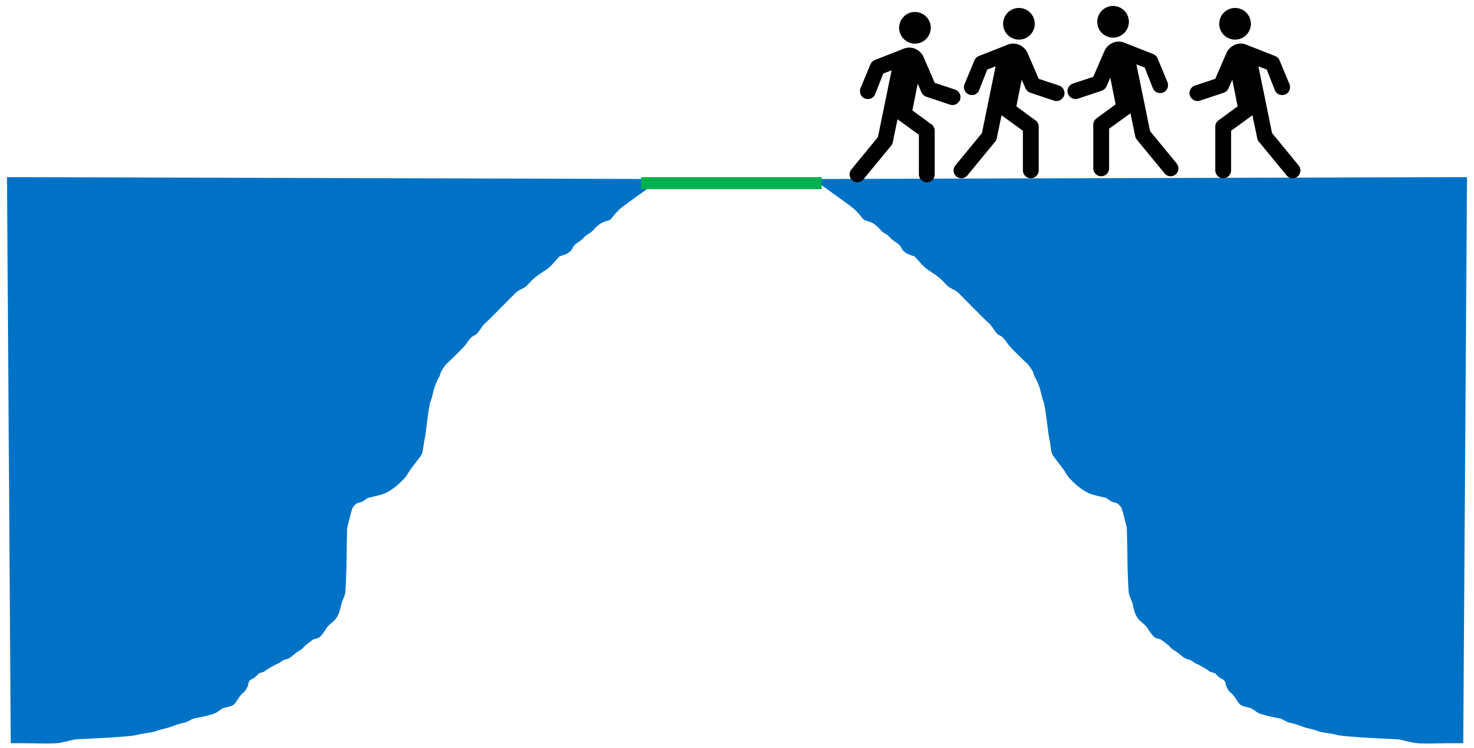


Avoid Mistakes In The Future

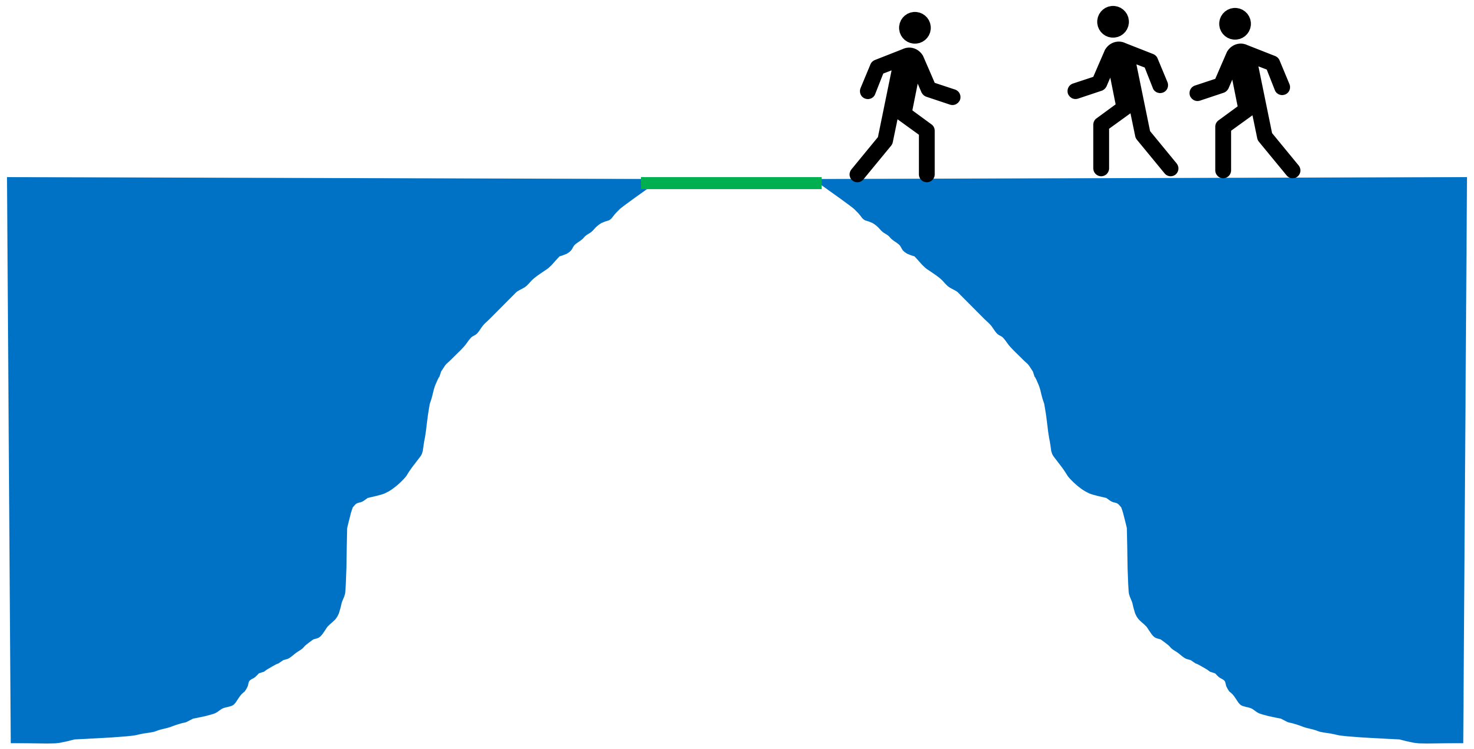


TRAILHEAD TECHNOLOGY PARTNERS

Avoid Mistakes In The Future

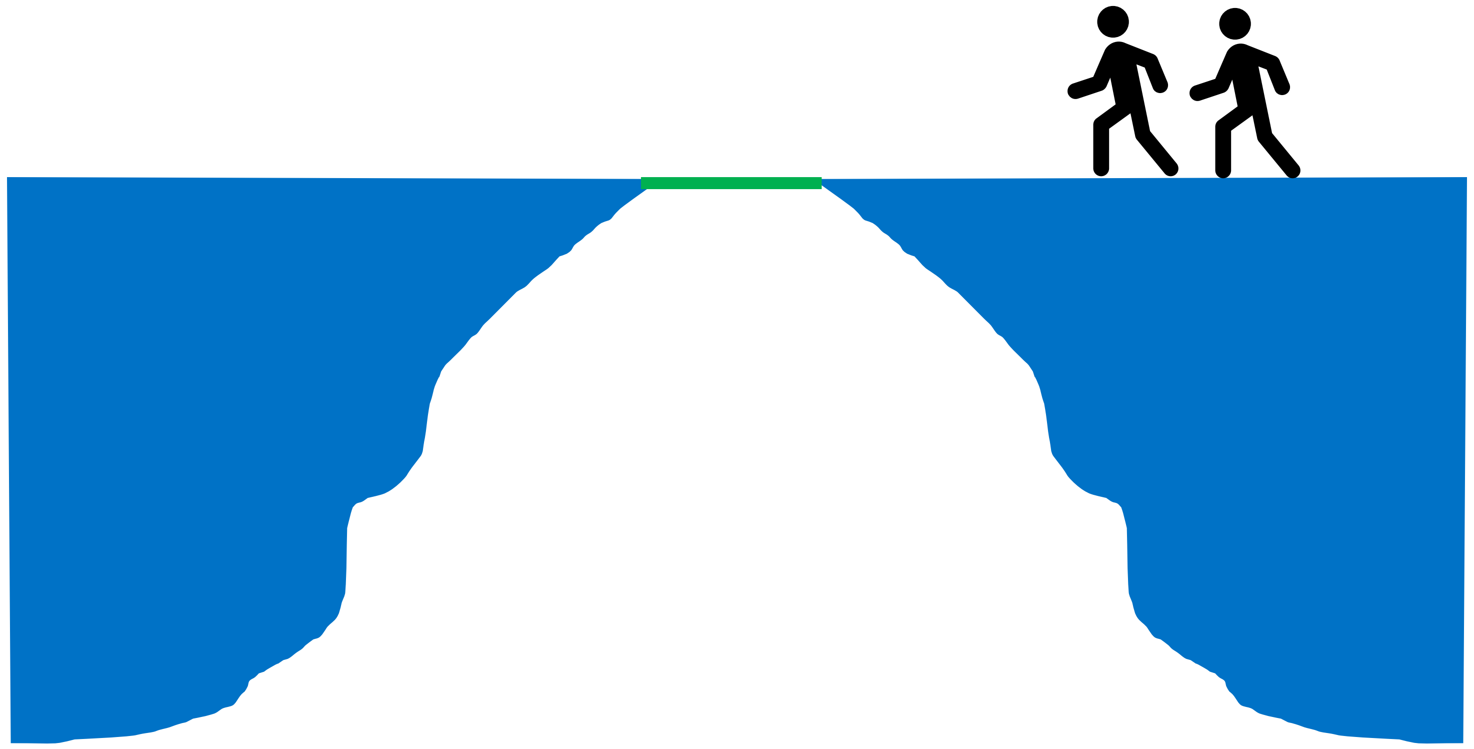


Avoid Mistakes In The Future



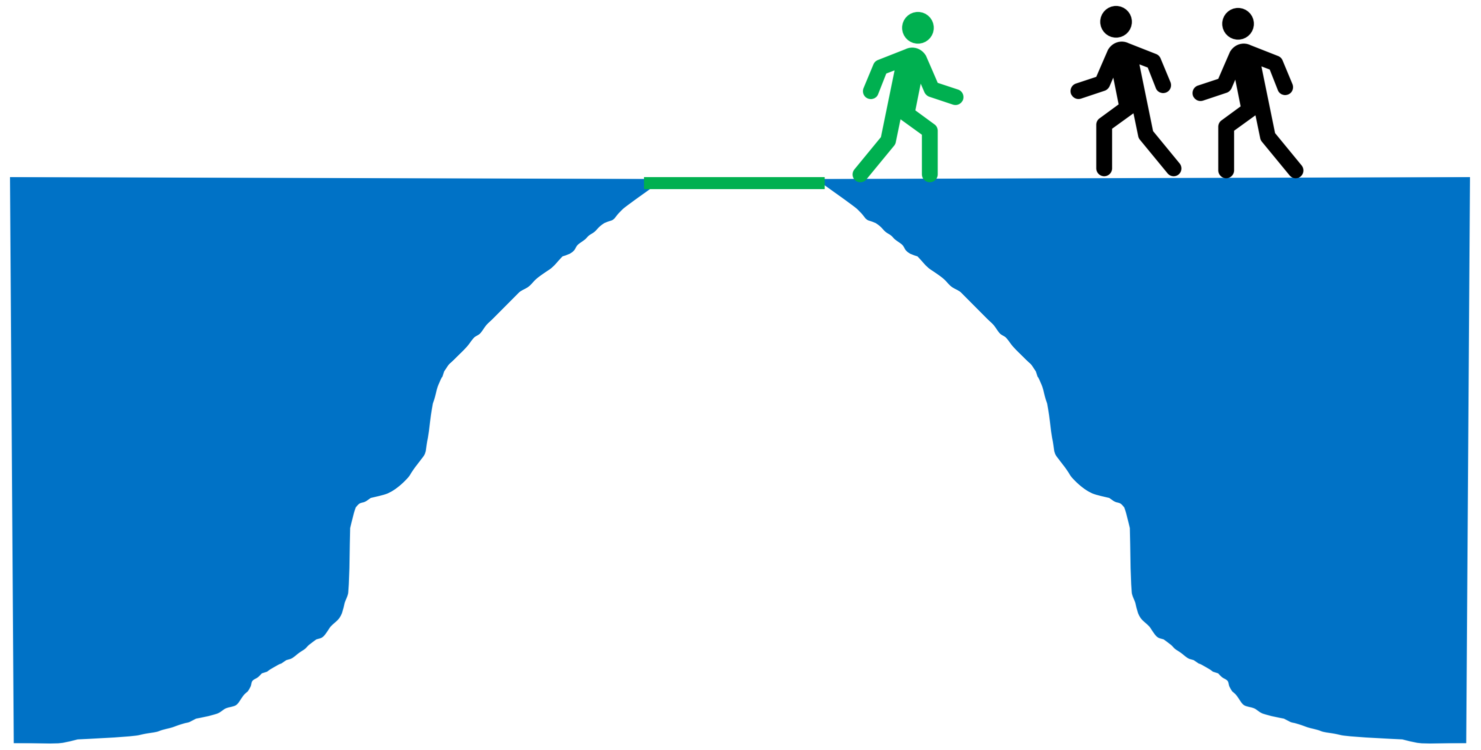
TRAILHEAD TECHNOLOGY PARTNERS

Avoid Mistakes In The Future



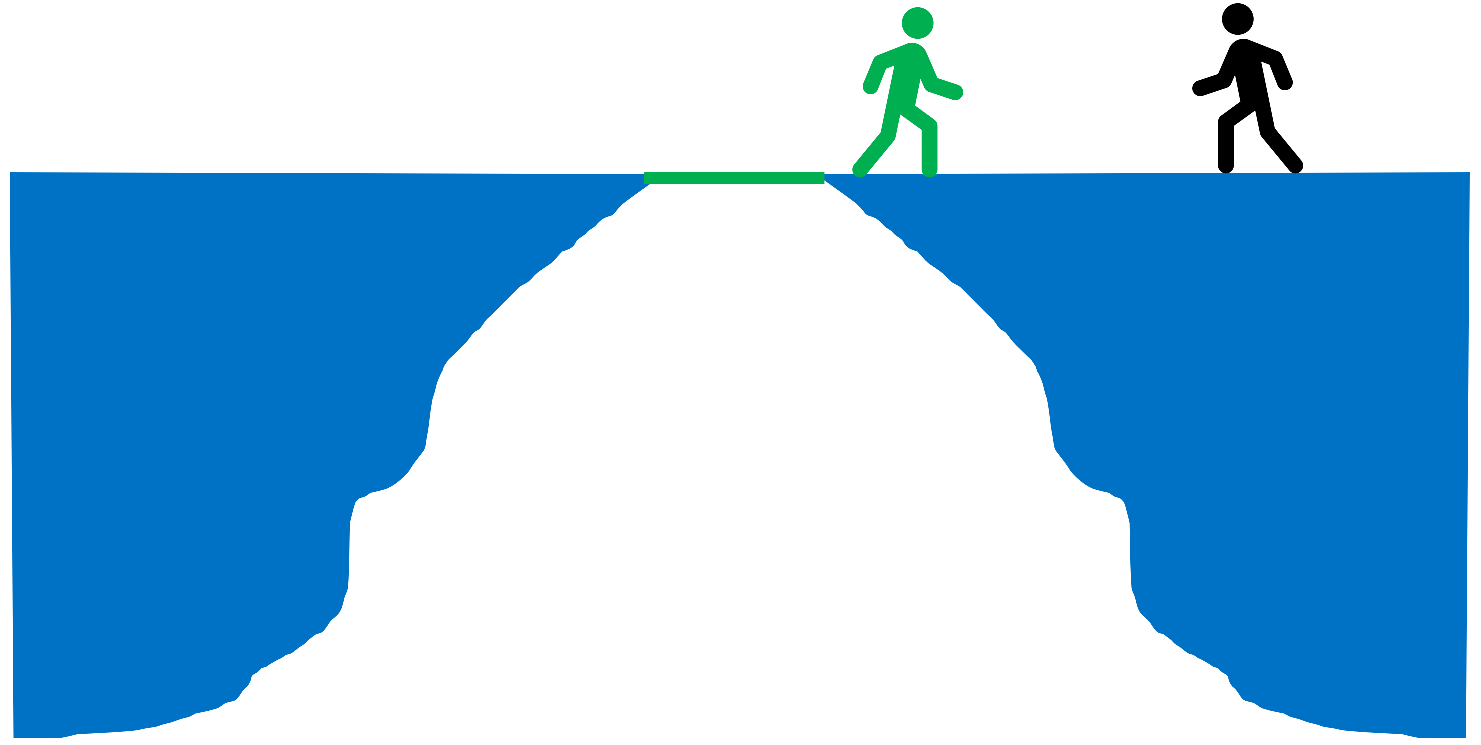
TRAILHEAD
TECHNOLOGY PARTNERS

Avoid Mistakes In The Future



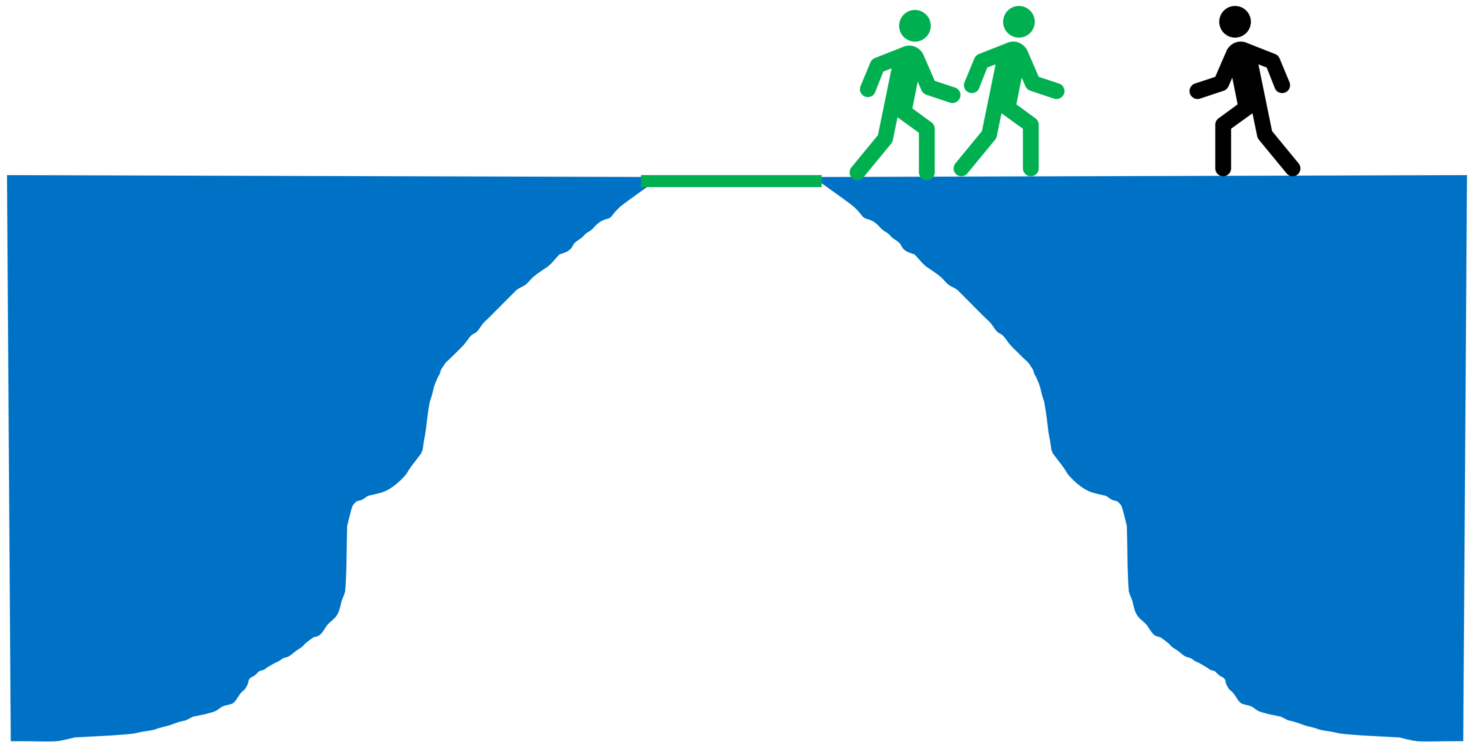
TRAILHEAD
TECHNOLOGY PARTNERS

Avoid Mistakes In The Future



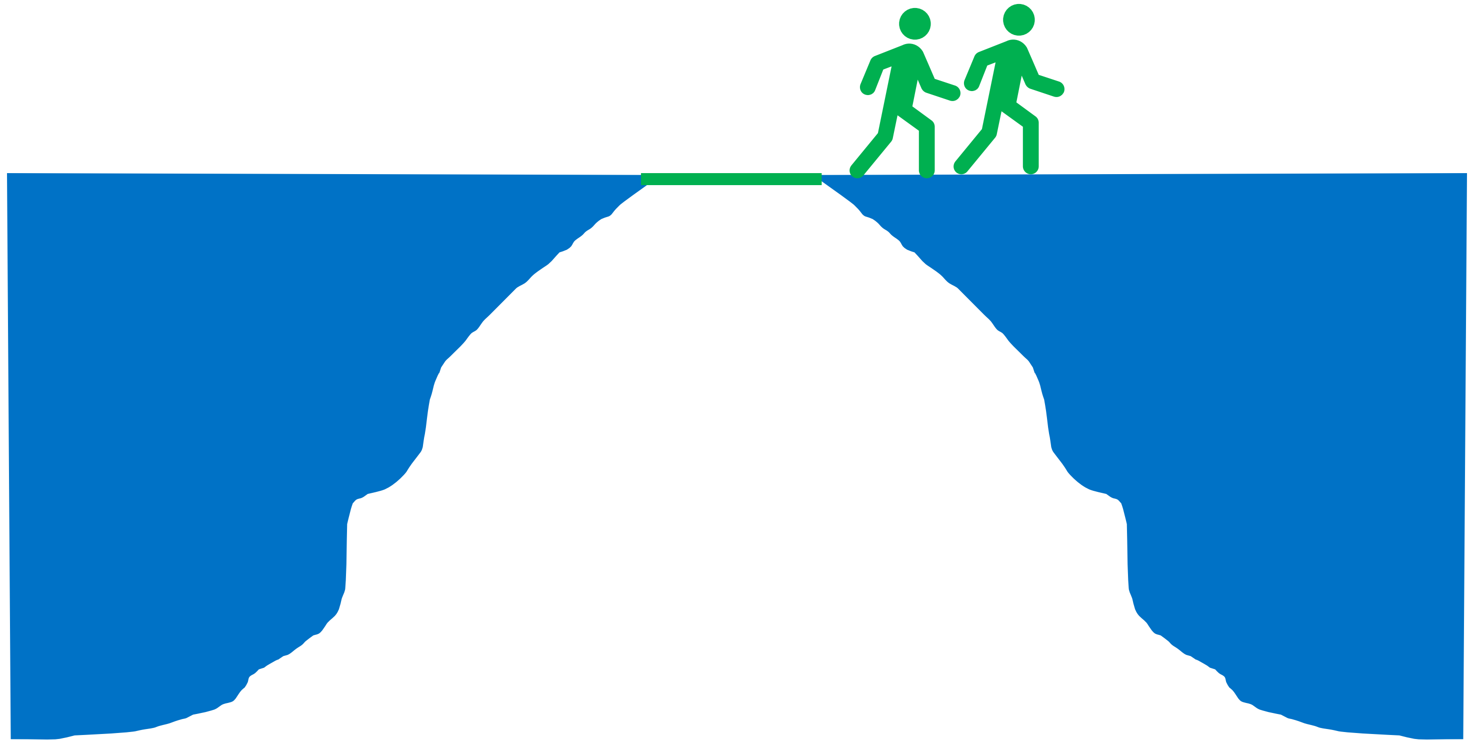
TRAILHEAD TECHNOLOGY PARTNERS

Avoid Mistakes In The Future



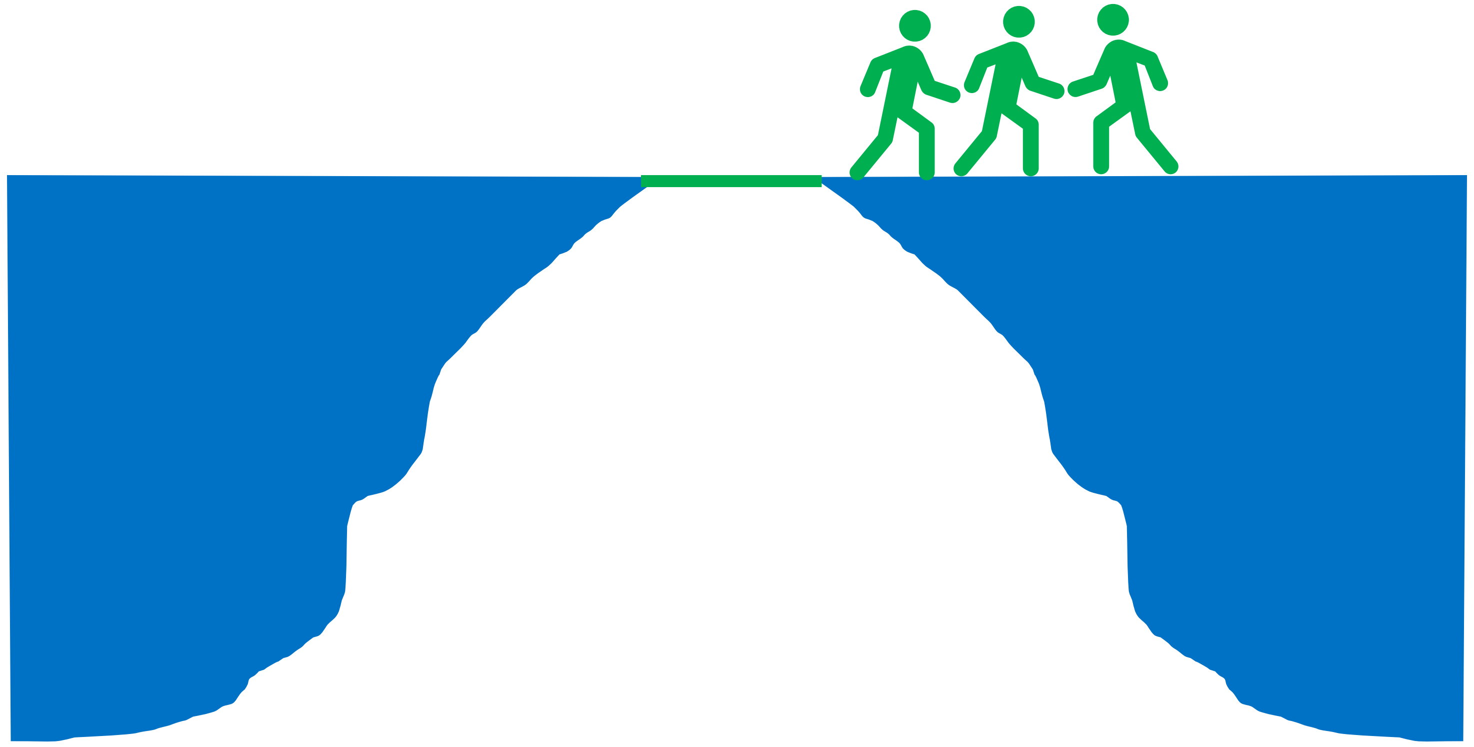
TRAILHEAD TECHNOLOGY PARTNERS

Avoid Mistakes In The Future



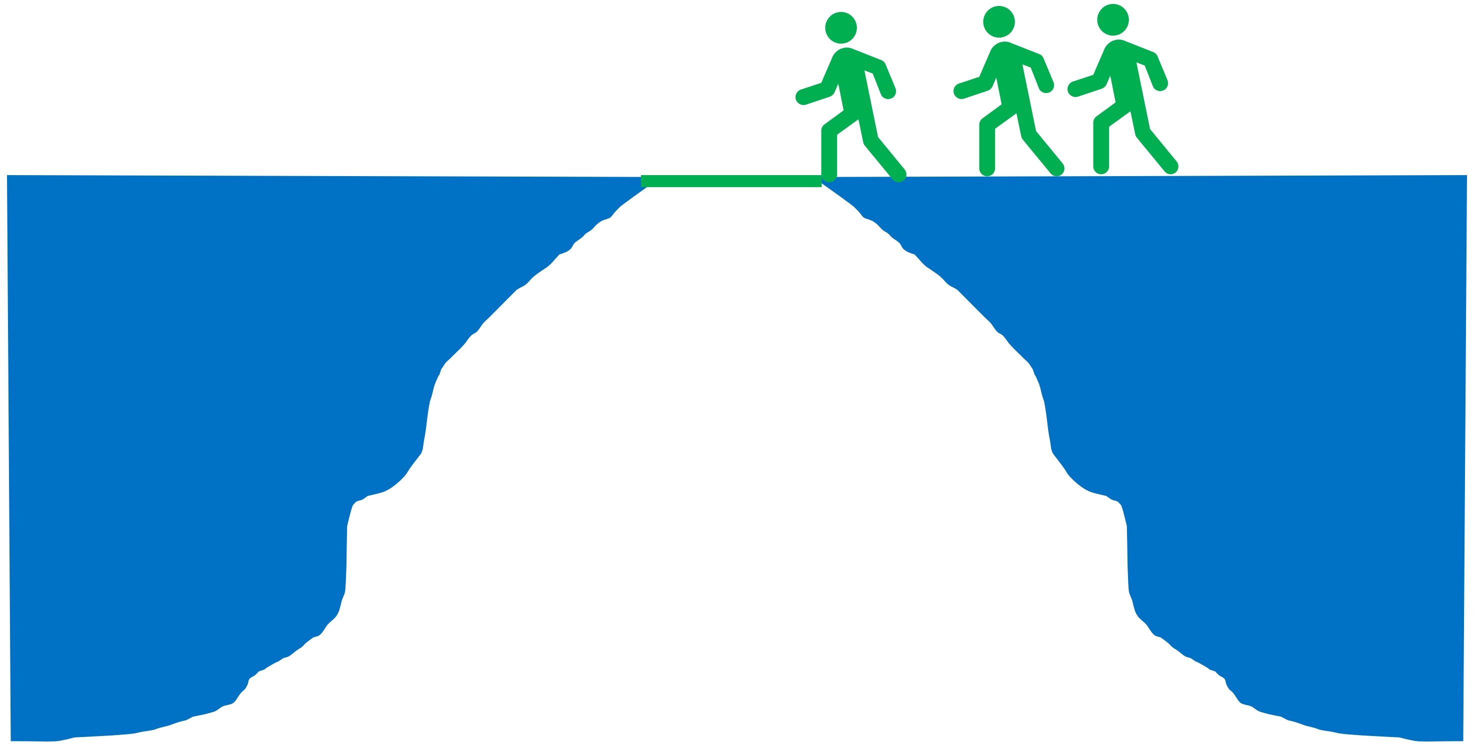
TRAILHEAD TECHNOLOGY PARTNERS

Avoid Mistakes In The Future



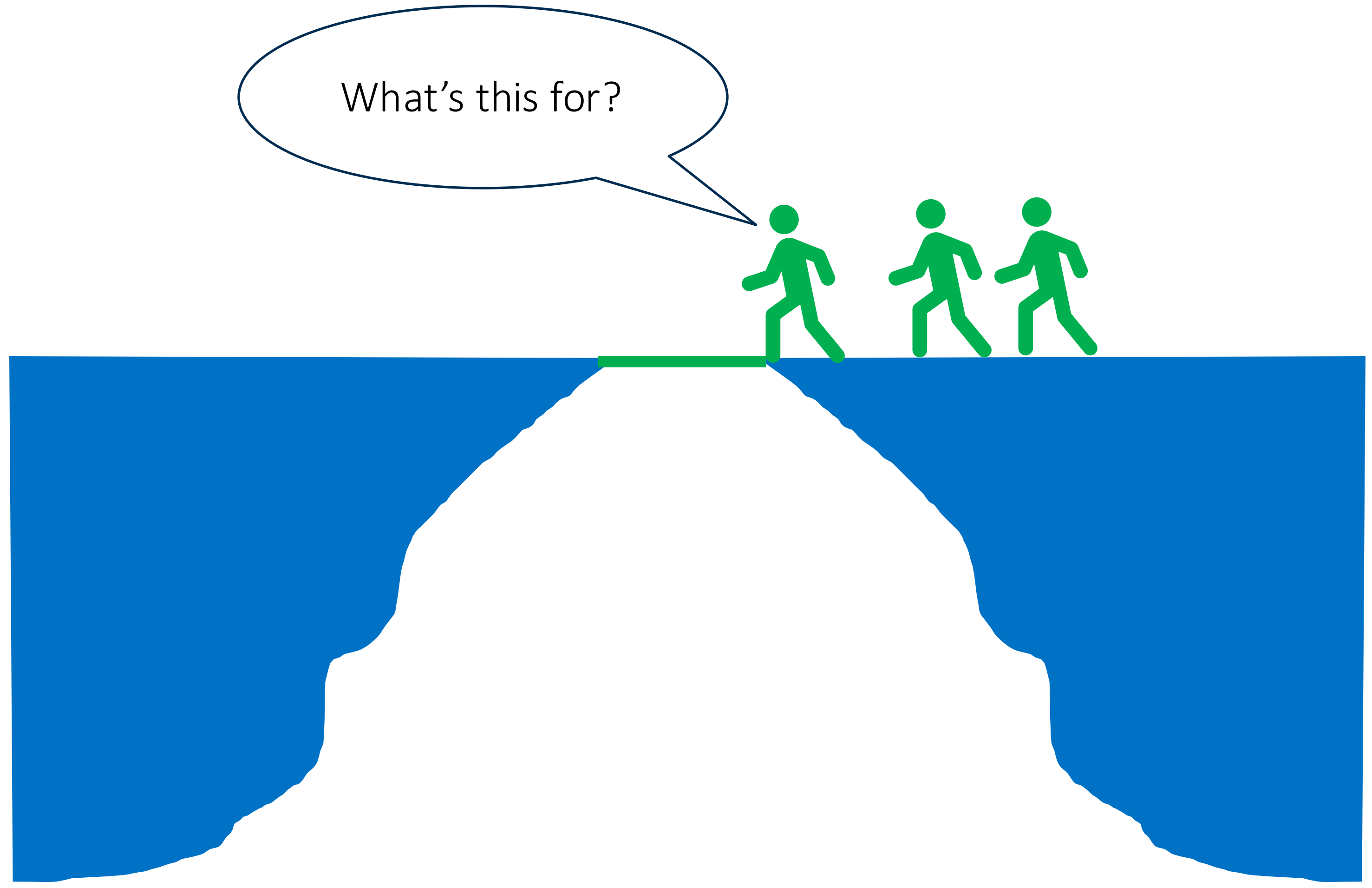
TRAILHEAD
TECHNOLOGY PARTNERS

Avoid Mistakes In The Future

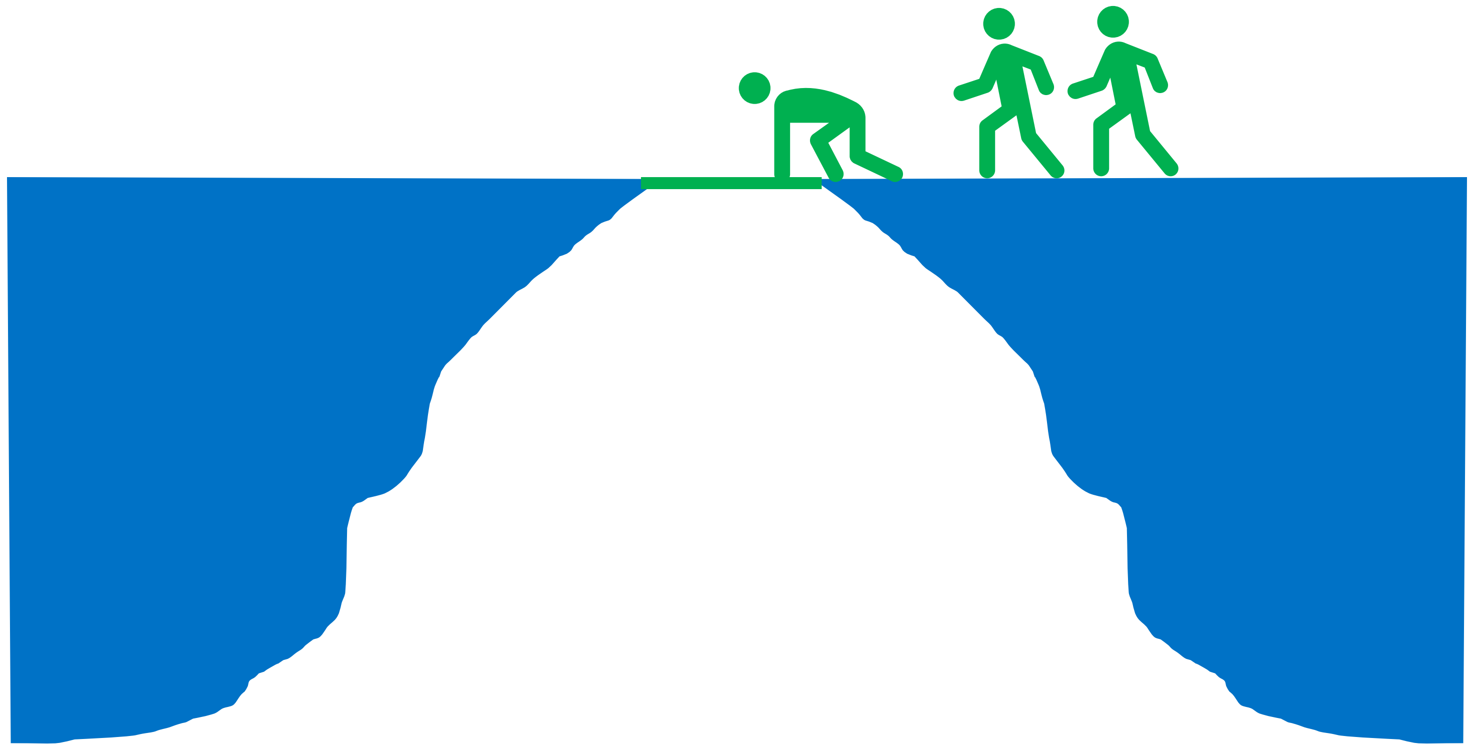


TRAILHEAD TECHNOLOGY PARTNERS

Avoid Mistakes In The Future

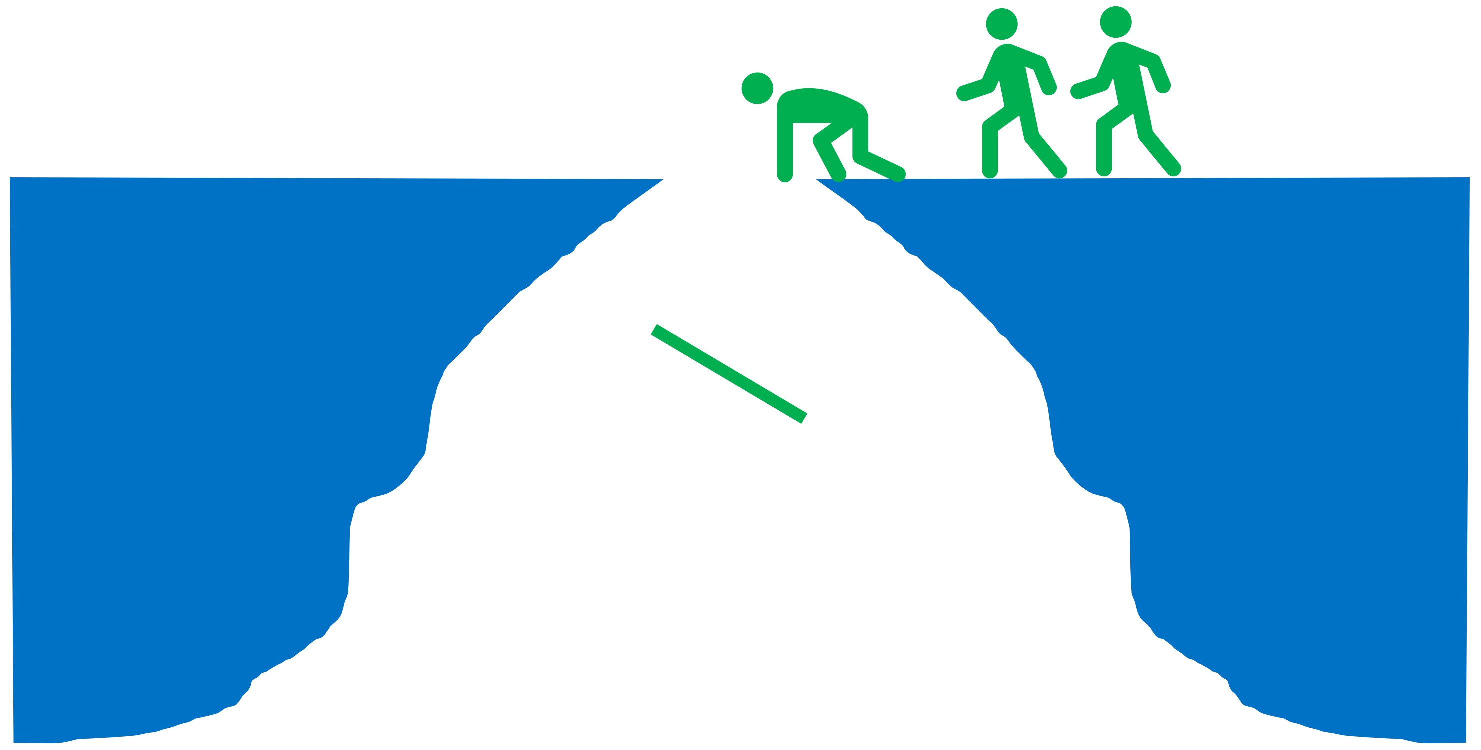


Avoid Mistakes In The Future



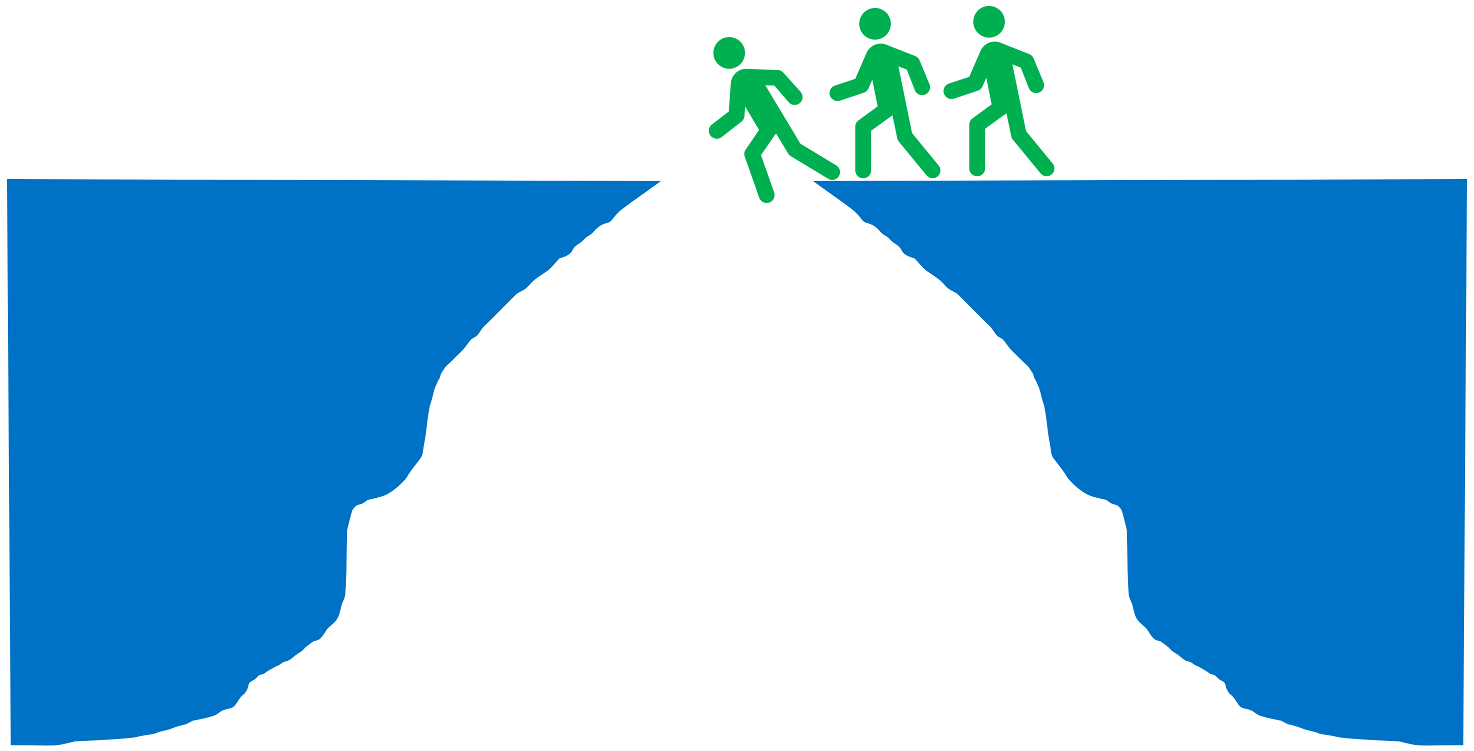
TRAILHEAD TECHNOLOGY PARTNERS

Avoid Mistakes In The Future



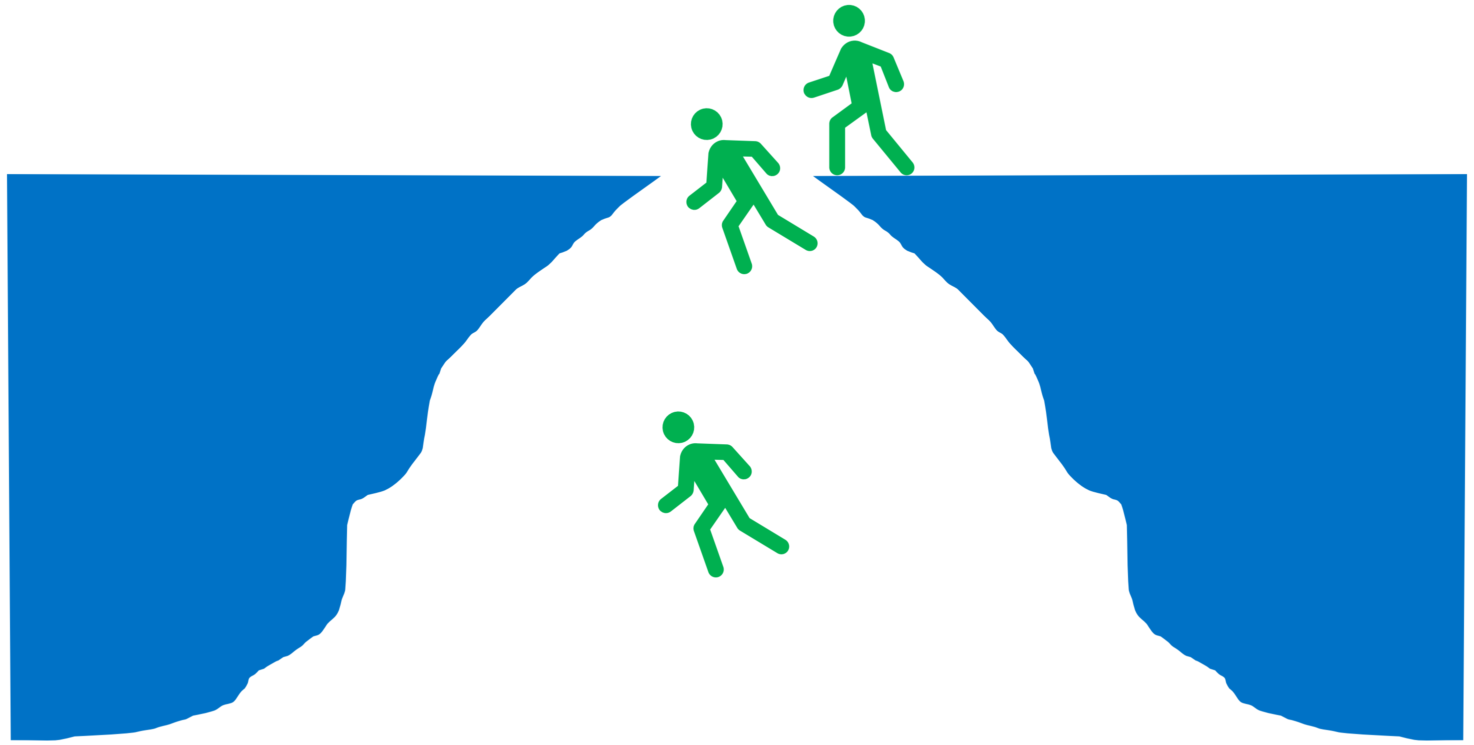
TRAILHEAD
TECHNOLOGY PARTNERS

Avoid Mistakes In The Future

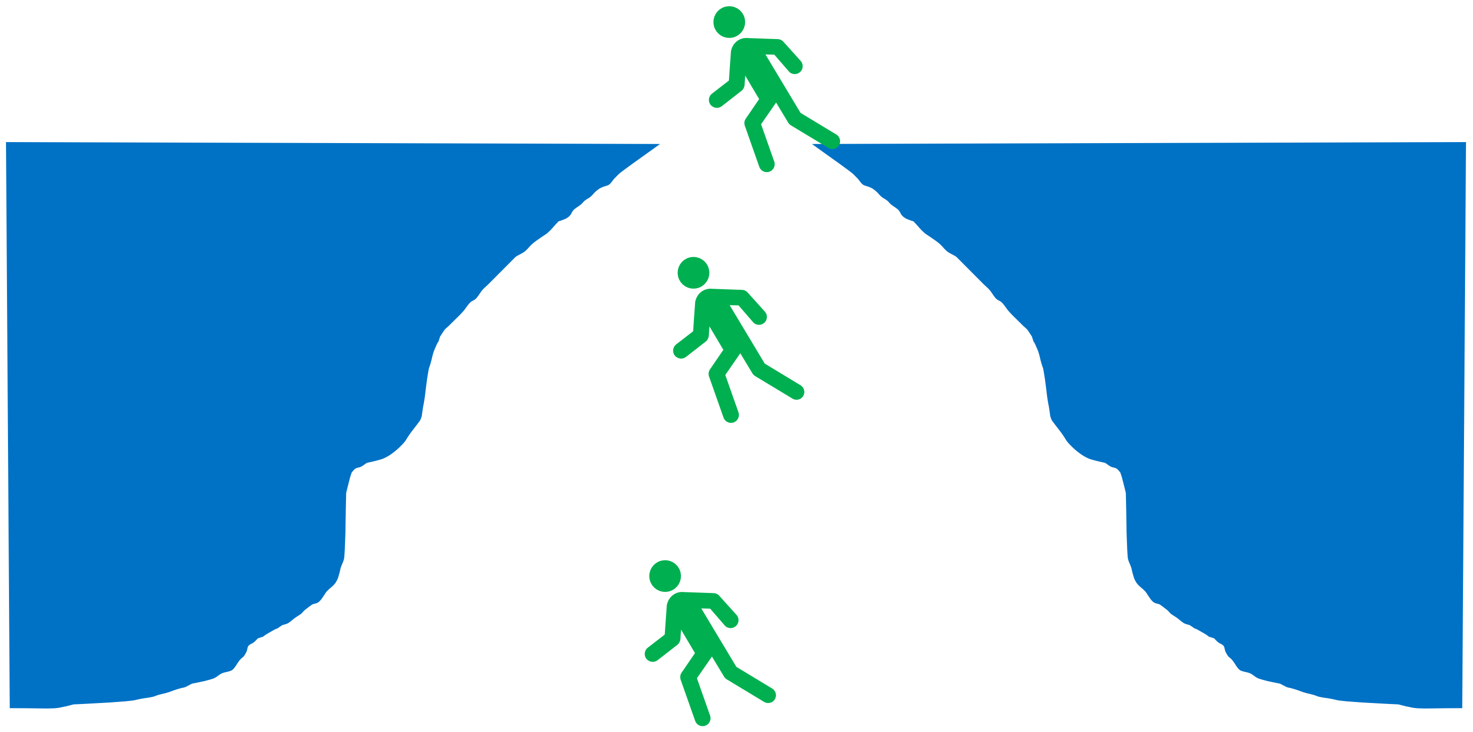


TRAILHEAD TECHNOLOGY PARTNERS

Avoid Mistakes In The Future



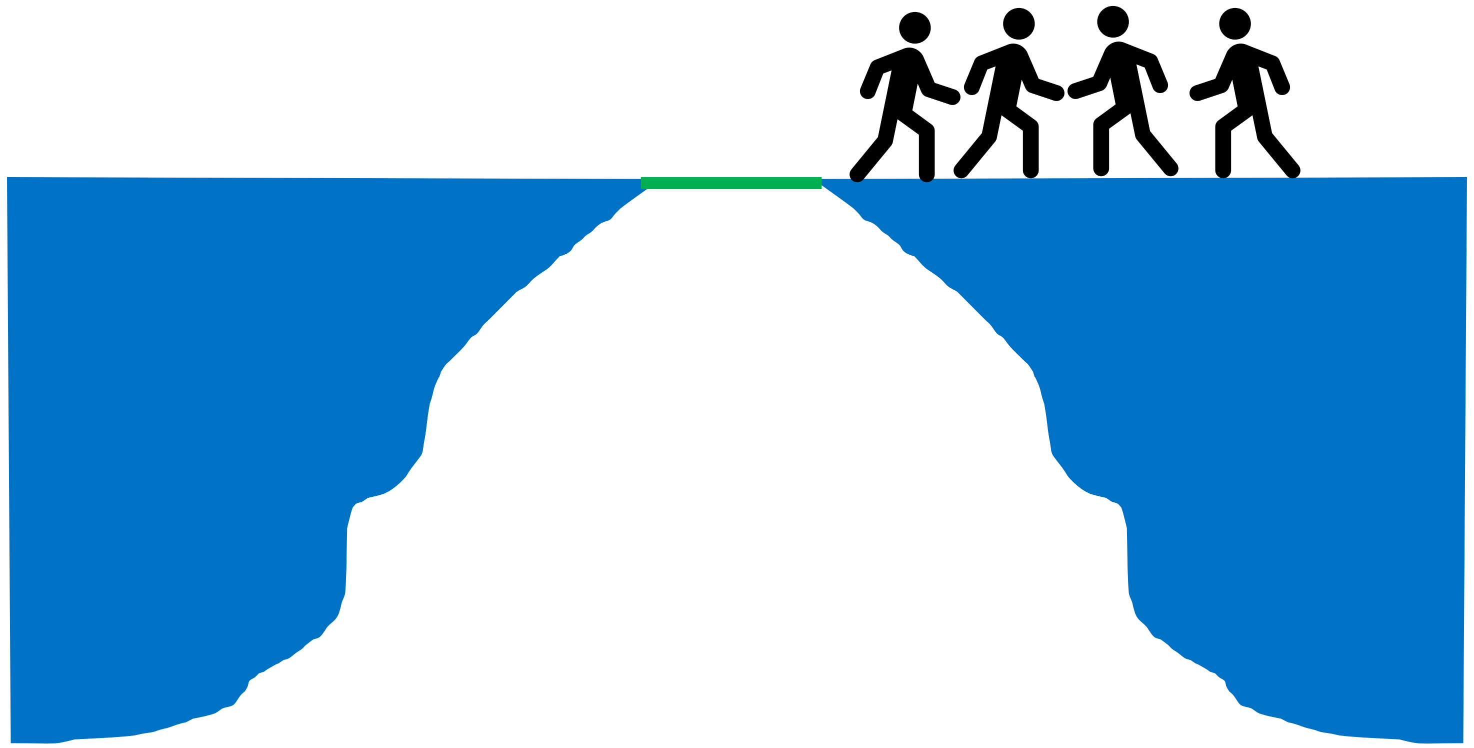
Avoid Mistakes In The Future



TRAILHEAD TECHNOLOGY PARTNERS

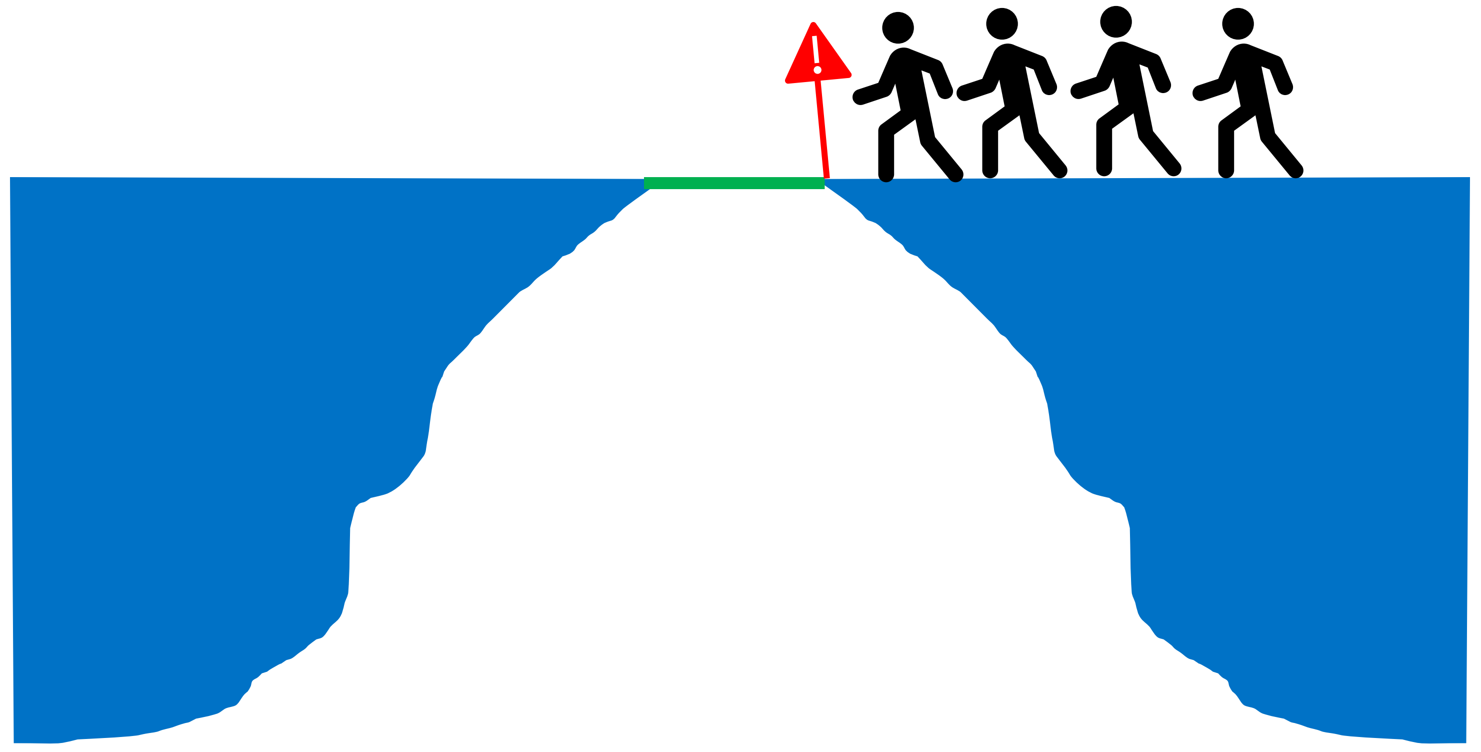
Avoid Mistakes In The Future

Avoid Mistakes In The Future



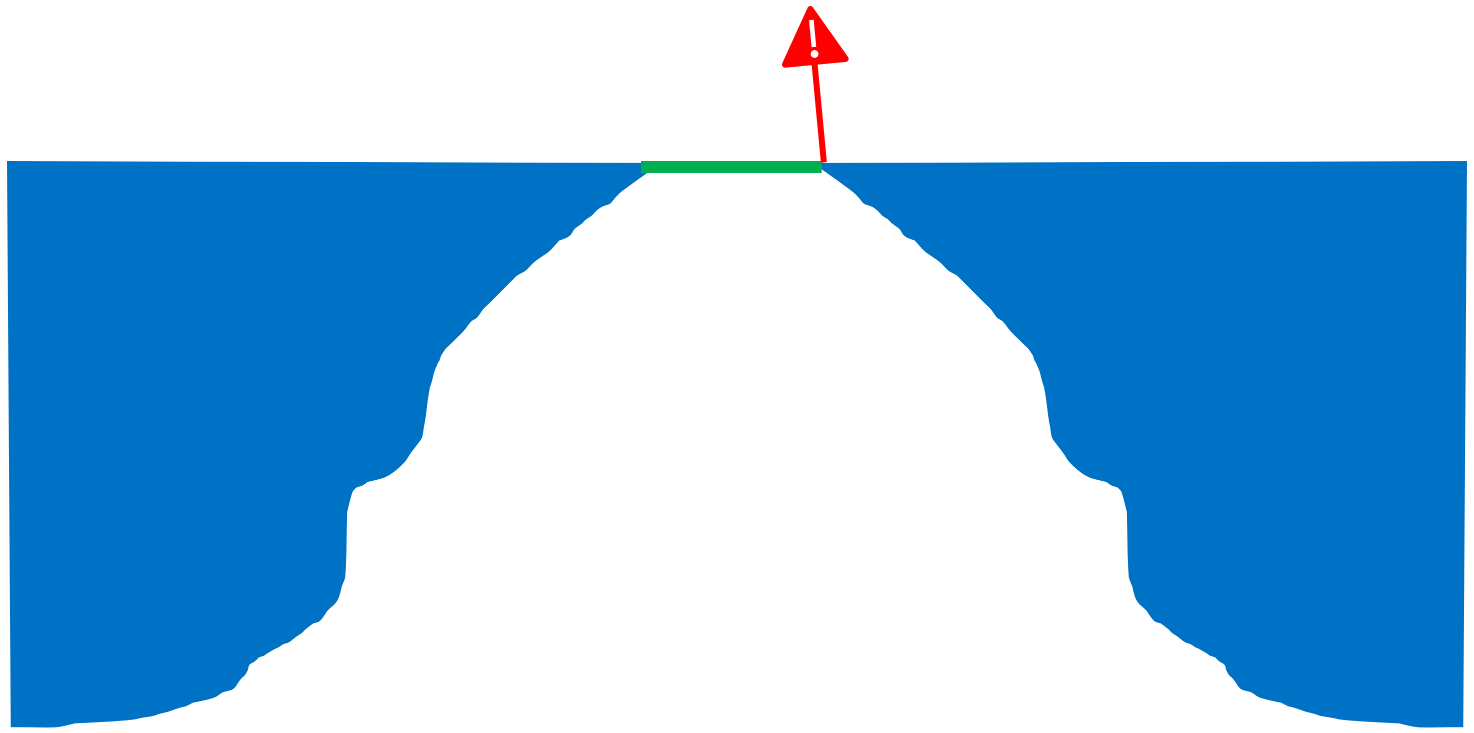
TRAILHEAD
TECHNOLOGY PARTNERS

Avoid Mistakes In The Future



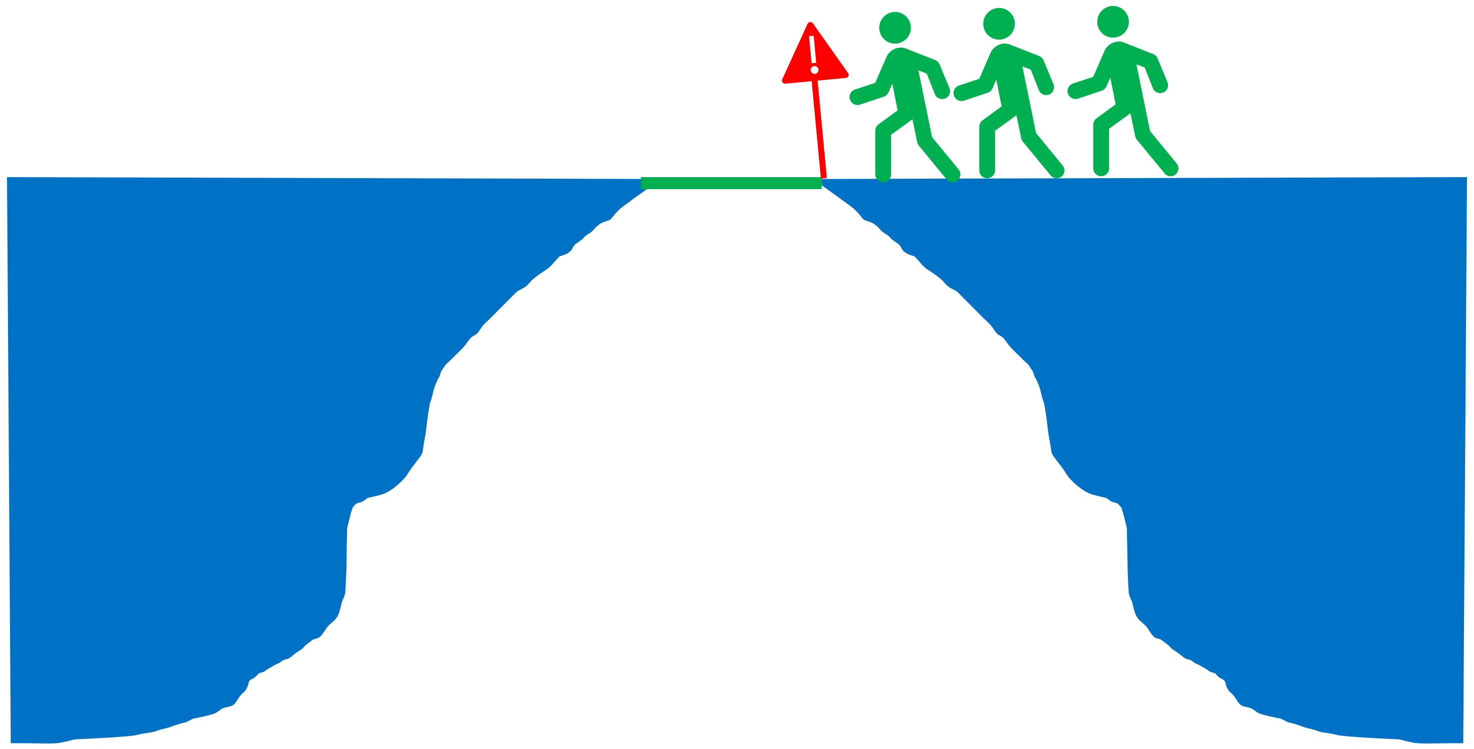
TRAILHEAD TECHNOLOGY PARTNERS

Avoid Mistakes In The Future



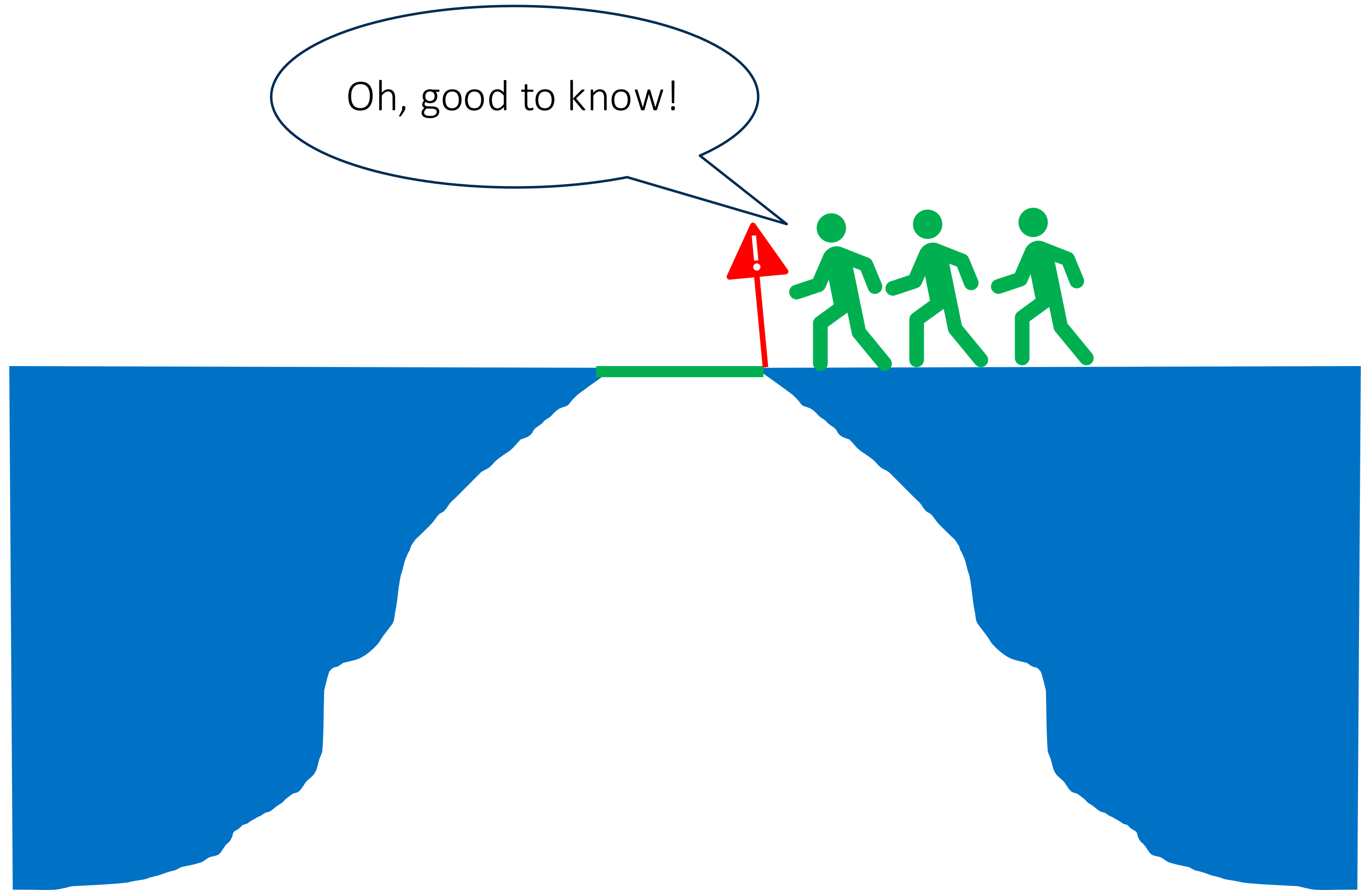
TRAILHEAD
TECHNOLOGY PARTNERS

Avoid Mistakes In The Future

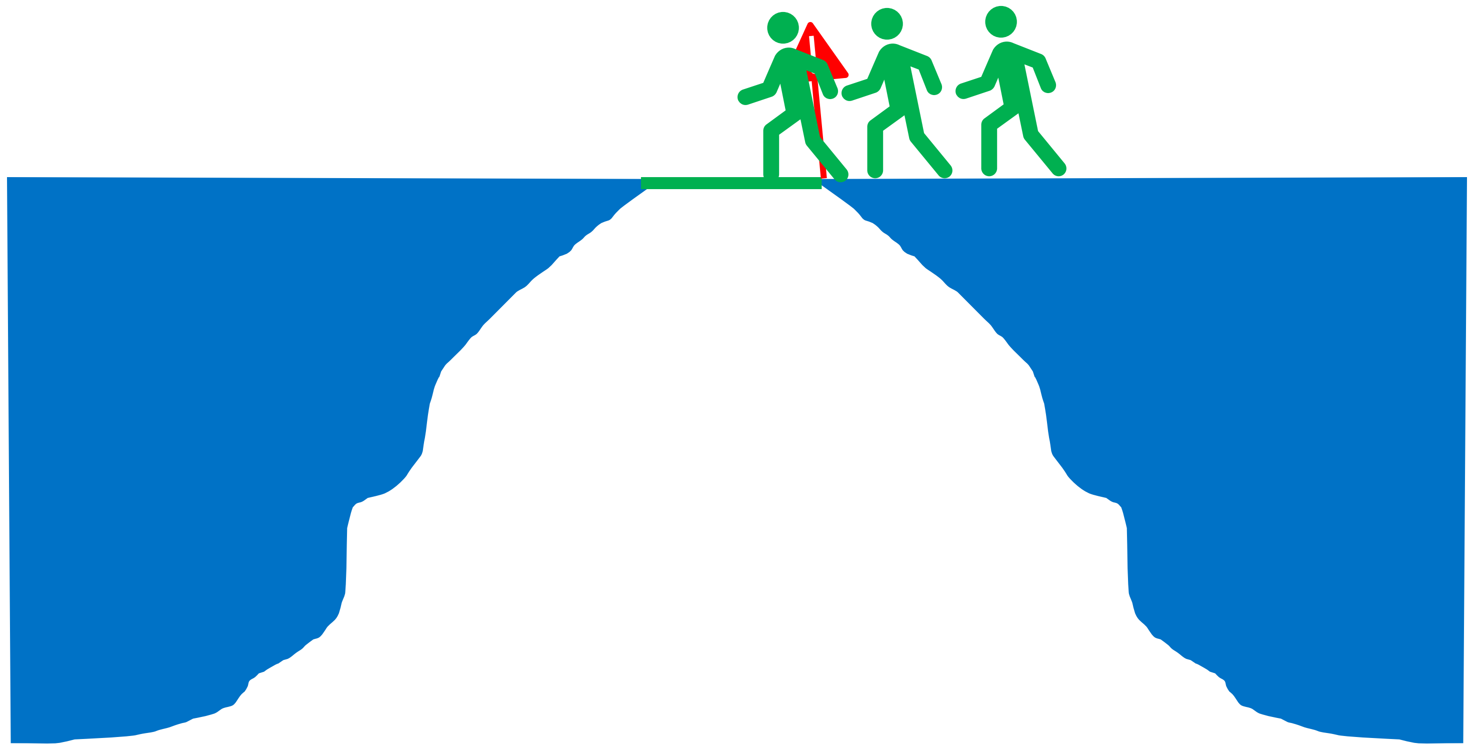


TRAILHEAD
TECHNOLOGY PARTNERS

Avoid Mistakes In The Future

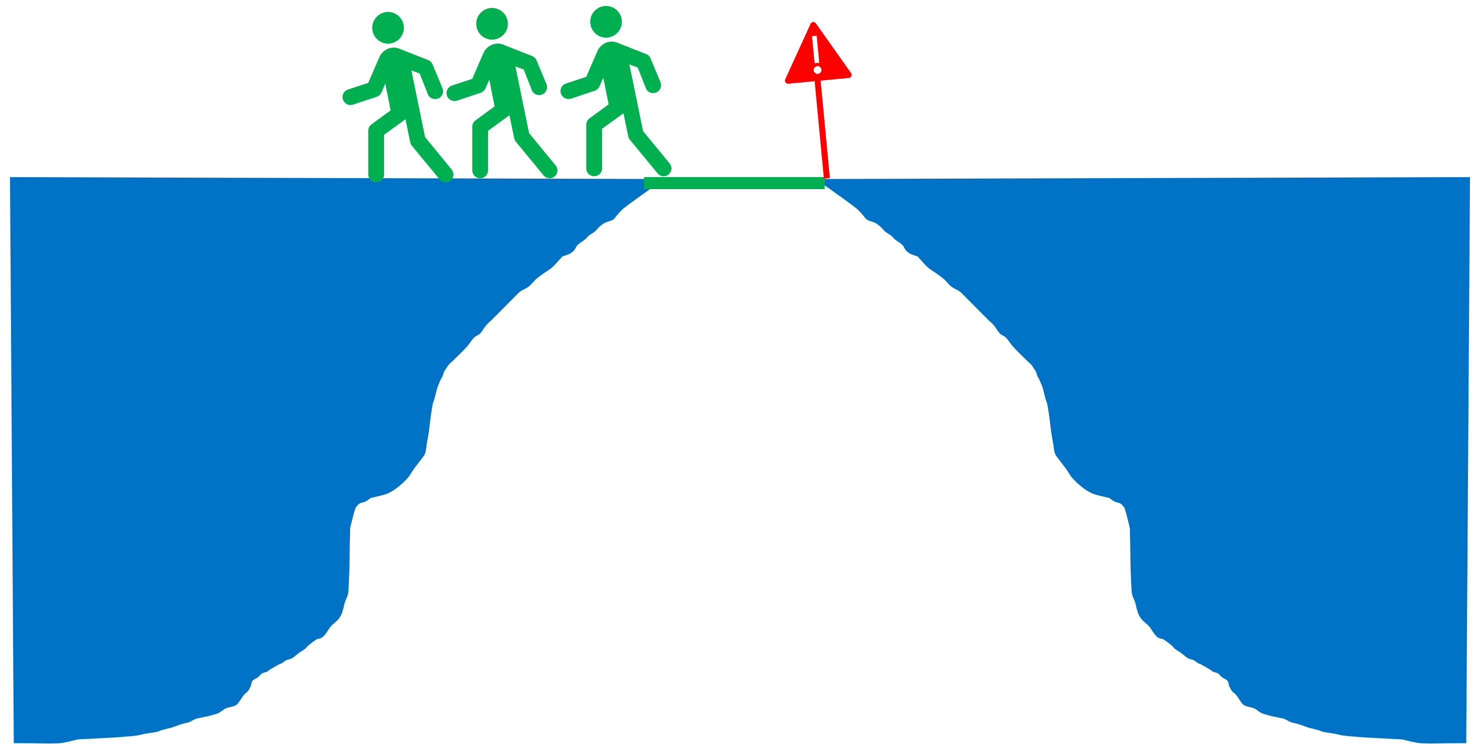


Avoid Mistakes In The Future



TRAILHEAD
TECHNOLOGY PARTNERS

Avoid Mistakes In The Future



TRAILHEAD TECHNOLOGY PARTNERS

Architecture Decision Records (ADR)

```
ADR.md
```

```
Title: Avoid Implementing Feature ‘Awaken Balrog’

Status: Accepted

Context:
In deciding whether to implement the Balrog Awakening feature, we draw inspiration from
Gandalf's advice in Moria.

Decision:
We will not implement the Balrog Awakening feature to avoid potential catastrophic
issues, such as losing our only wizard.

Consequences:
Every time we’ve awaken a Balrog in the past, we’ve lost a perfectly good wizard.
```

TRAILHEAD TECHNOLOGY PARTNERS

Architecture Decision Records (ADR)

Git

Wiki

Agile / Iterative Development

Rule 2

Agile / Iterative Development

- Implement **as little as possible** to solve the problem (**MVP**)
- **Iterate** quickly with **feedback**

YAGNI
(You Ain't Gonna Need It)

Rule 3

YAGNI

Implement Only
Current Requirements

Don’t Build For
Future Needs

KISS (Keep It Simple, Software-developer)

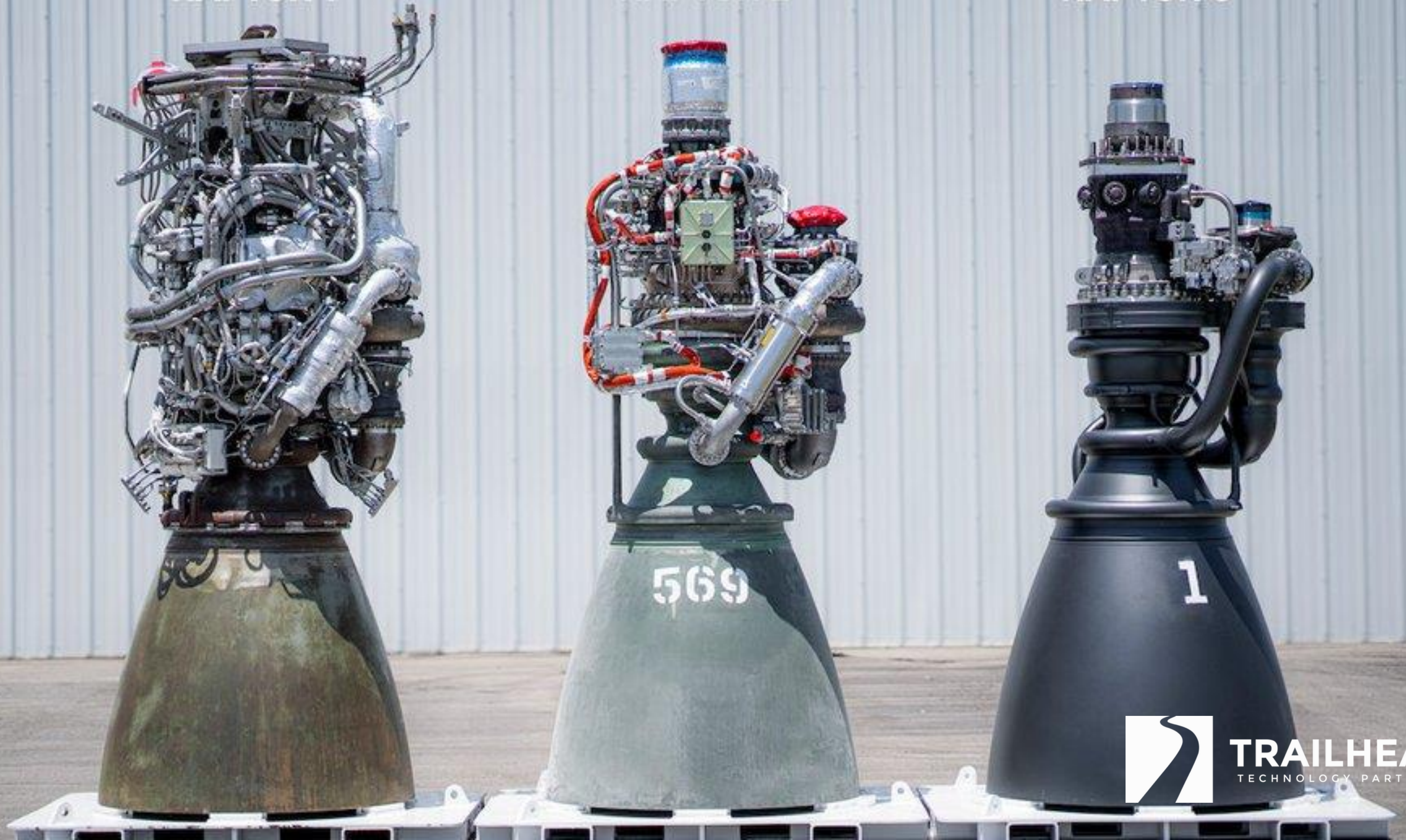
Rule 4

KISS

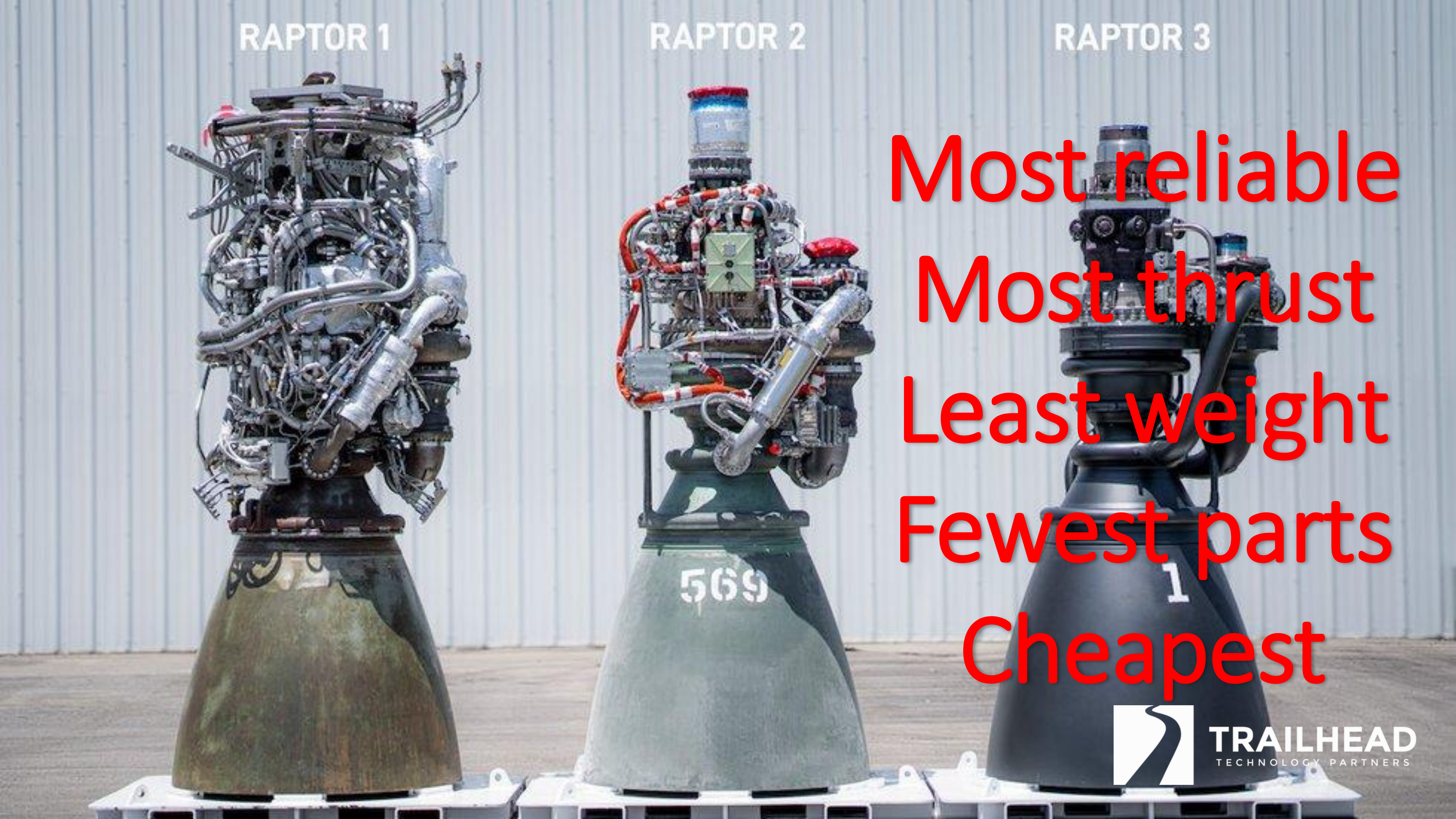
Less Surface Area
For Failures

Easier to
Maintain

TRAILHEAD
TECHNOLOGY PARTNERS



TRAILHEAD
TECHNOLOGY PARTNERS



Most reliable
Most thrust
Least weight
Fewest parts
Cheapest

TRAILHEAD
TECHNOLOGY PARTNERS

KISS

“I would have written a shorter letter, but did not have the time.”

- Blaise Pascal

DRY but in a WET Way

Rule 5

DRY but in a WET Way

DRY - Don’t Repeat Yourself

WET - Write Everything Twice

Principle of Least Astonishment (POLA)

Rule 6

Principle of Least Astonishment

"A **component** of a system **should behave** in a way that **most users will expect** it to behave, and therefore **not astonish** or surprise users."

TRAILHEAD
TECHNOLOGY PARTNERS

Principle of Least Astonishment (POLA)

Reduces Cognitive Load

Reduces Mistakes

Easier to Learn

TRAILHEAD
TECHNOLOGY PARTNERS

Pull Reqs / Code Reviews

Rule 7

TRAILHEAD TECHNOLOGY PARTNERS

Pull Reqs / Code Reviews

Detecting Patterns of Over-Engineering

Catching Unnecessary Features Early

Ensuring Adherence to Best Practices

Encouraging Incremental Development

"Shift Left"

The **earlier** in the development lifecycle you **catch a defect**, the **less expensive** it is to fix.

Typical Code Review Checklist

- Bug Check
- Acceptance Criteria Check
- Code Standards Check
- Clarity Check
- Performance Check
- Documentation Check

Typical Code Review Checklist

- Bug Check
- Acceptance Criteria Check
- Code Standards Check
- Clarity Check
- Performance Check
- Documentation Check

✓ Simplicity Check
✓ YAGNI Check
✓ KISS Check
✓ Premature Optimization Check

Feedback Loops

Rule 8

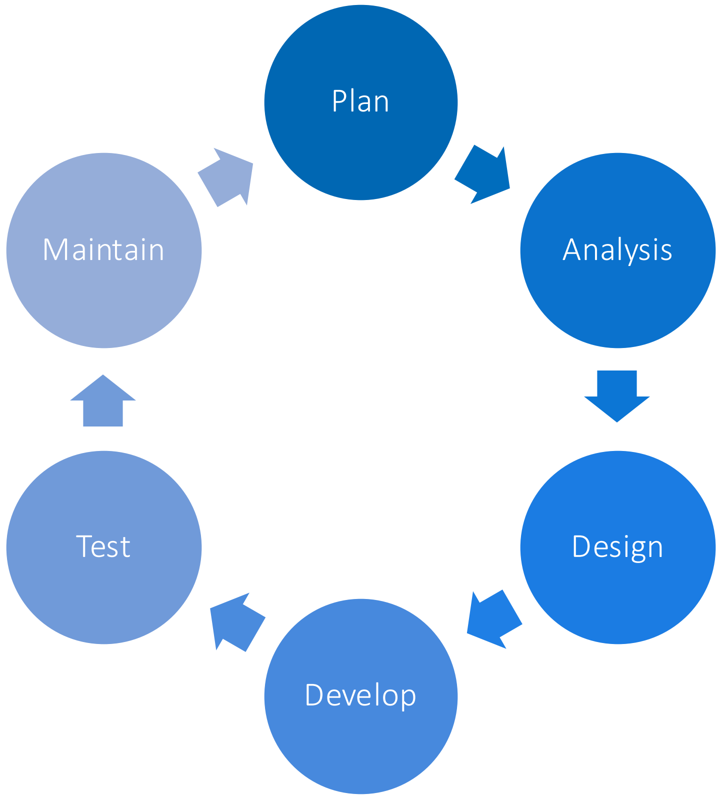
Feedback Loops

Aligning Development with User Needs

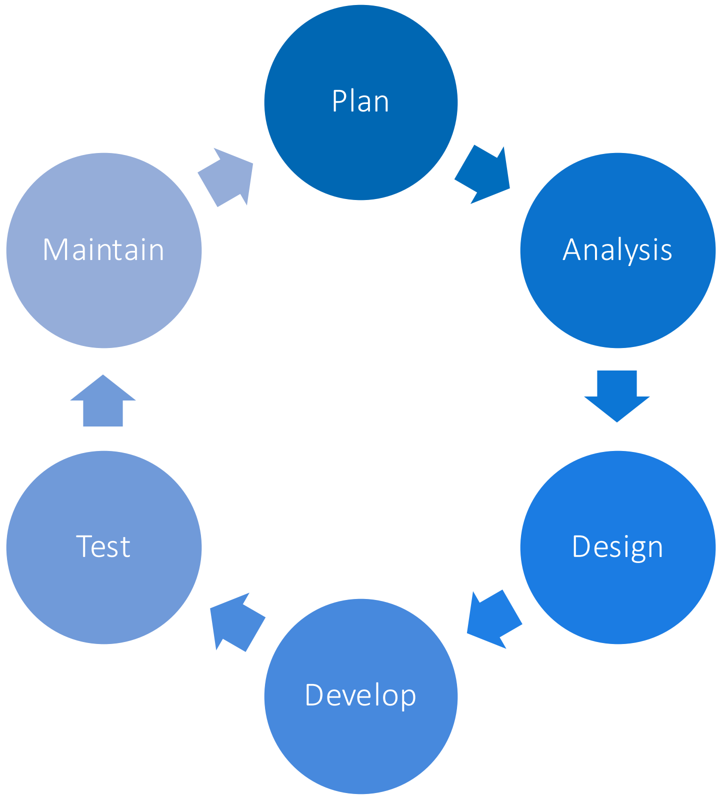
Encouraging Iterative Development

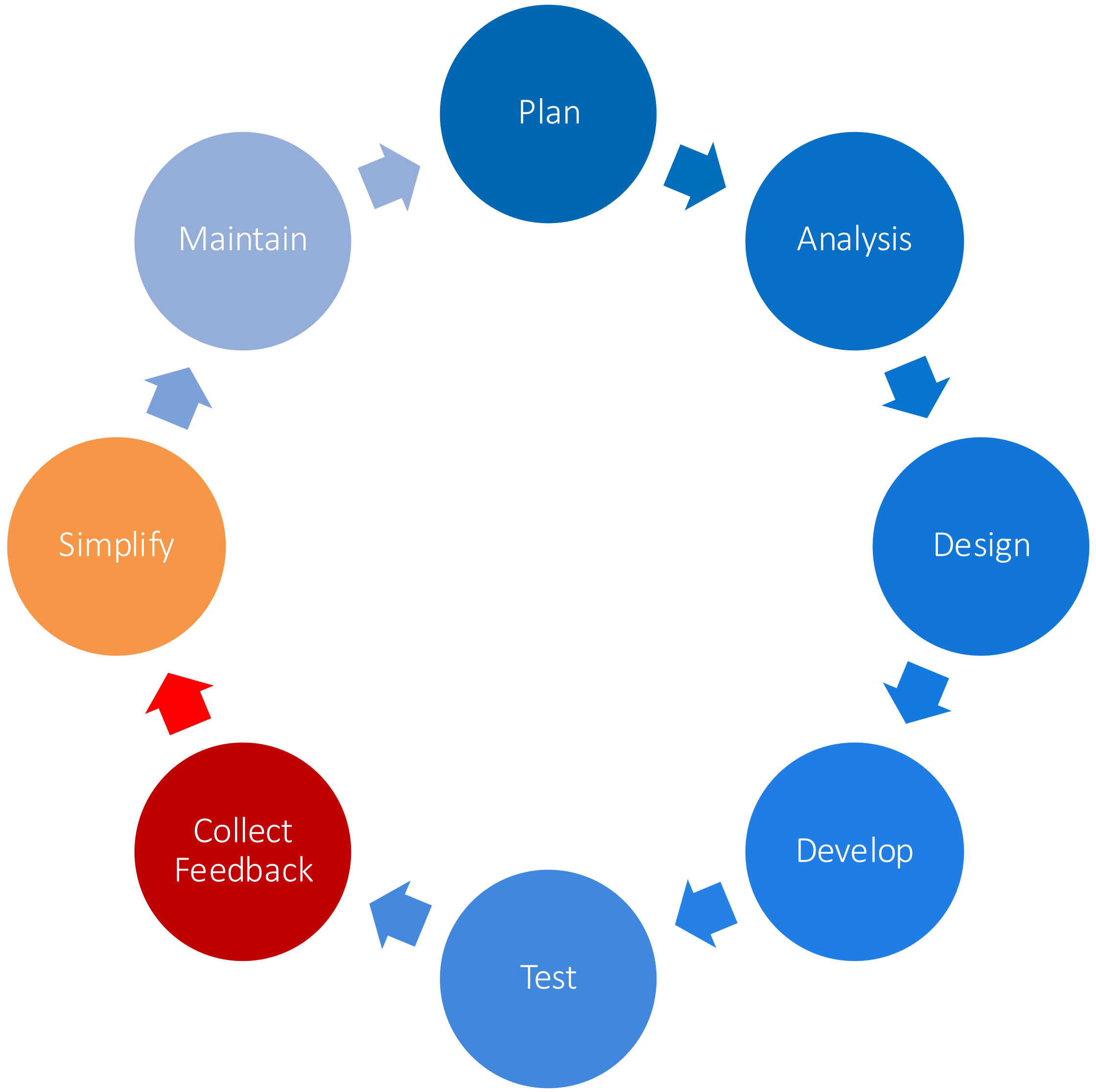
Detecting and correcting over-engineering early

Feedback Loops



Feedback Loops





TRAILHEAD
TECHNOLOGY PARTNERS

Adopt Proven Solutions

Rule 9

TRAILHEAD
TECHNOLOGY PARTNERS

Adopt Proven Solutions

Adapting Proven Solutions

More predictable outcomes

Reduces unknowns

Manages risk

TRAILHEAD
TECHNOLOGY PARTNERS

Adapting Proven Solutions

Use Frameworks
You Know

New Architectures
Only as POC

Adopt New Tech Behind
Buzzword Curve

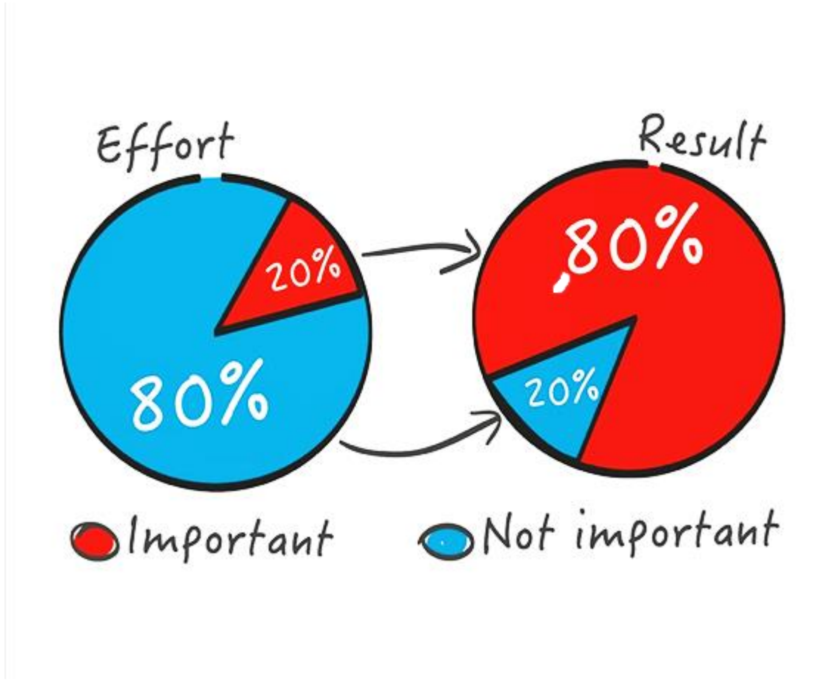
TRAILHEAD
TECHNOLOGY PARTNERS

80/20 Rule (Pareto Principle)

Rule 10

80/20 Rule

"Roughly 80% of consequences come from 20% of causes."



SOURCE https://www.sreedeep.com/the-pareto-principle-explained/

80/20 Rule Allows

Focusing on
high-impact features

Efficient
use of resources

Avoid
perfectionism

Summing Up

1. Over-engineering is **common** but **costly**
2. **Know the signs** of over-engineering
3. Implement **rules to avoid** over-engineering

TRAILHEAD TECHNOLOGY PARTNERS

Thanks! Questions?

Jonathan "J." Tower

Microsoft MVP in .NET

jtower@trailheadtechnology.com

trailheadtechnology.com/blog

jtowermi

jtower

FREE

CONSULTATION

bit.ly/th-offer

github.com/trailheadtechnology/over-engineering