

PRIORITIZING TECHNICAL DEBT AS IF TIME AND MONEY MATTERED

CodeScene™

Powered by Empear

Lehman's "Laws" of Software Evolution



Continuing Change

“a system must be continually adapted or it becomes progressively less satisfactory”

Increasing Complexity

“as a system evolves, its complexity increases unless work is done to maintain or reduce it”

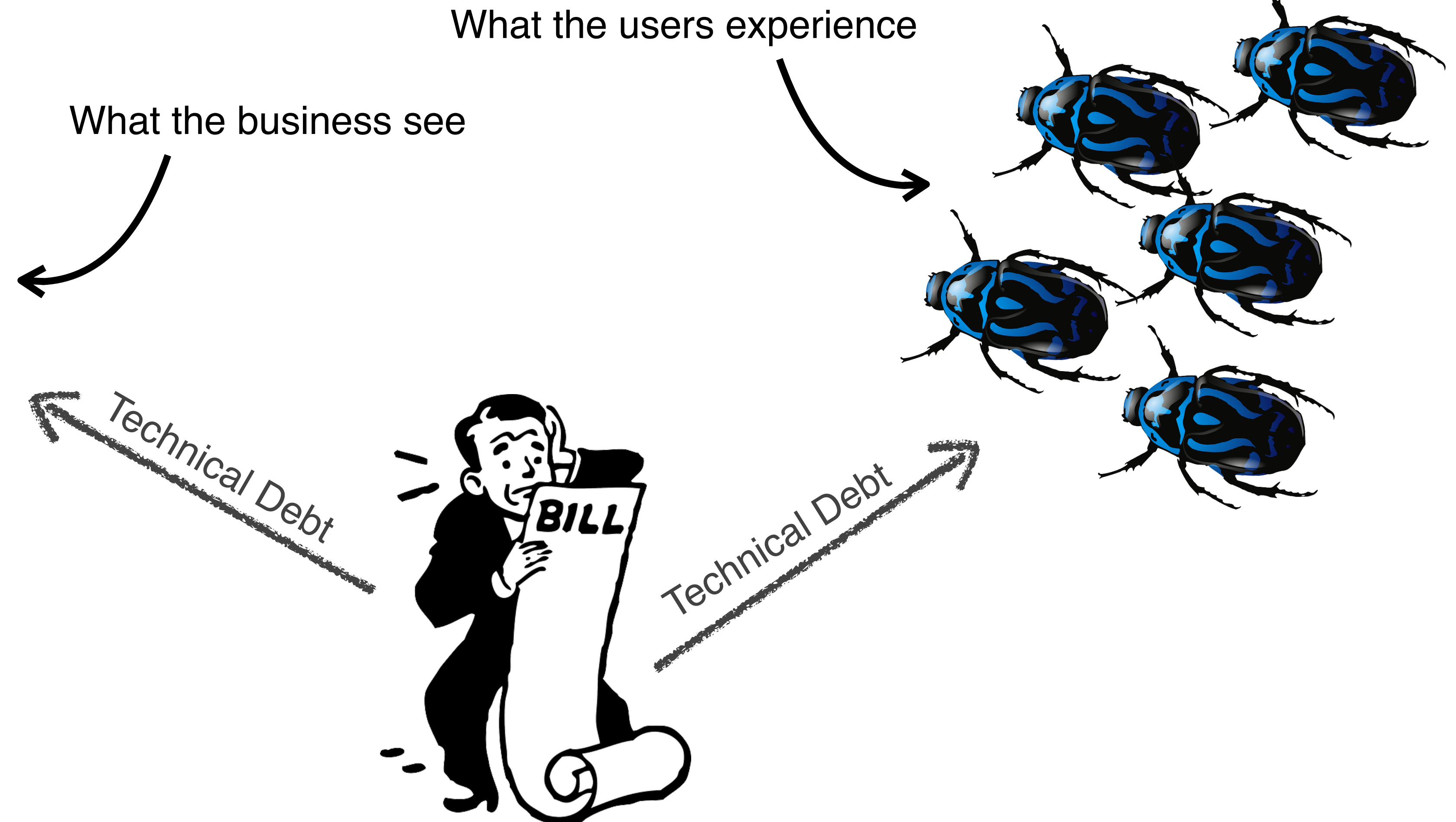
Are We Treating Symptoms Instead of the Real Issues?

Roadmap



Long Lead Times
Lack of Predictability

Product



QUANTIFYING TECHNICAL DEBT?



?

==



The background of the slide is a close-up, slightly blurred image of ancient Egyptian hieroglyphs carved into a light-colored stone or papyrus. The hieroglyphs are arranged in vertical columns and include various symbols such as birds, lotus flowers, and human figures. The lighting is warm, creating a golden-brown hue across the entire image.

4000 Years of Technical Debt?

4000 Years Ago => The Start of Recorded History

THE PERILS QUANTIFYING TECHNICAL DEBT

Q: What behaviour do we reinforce by quantifying technical debt?

important!



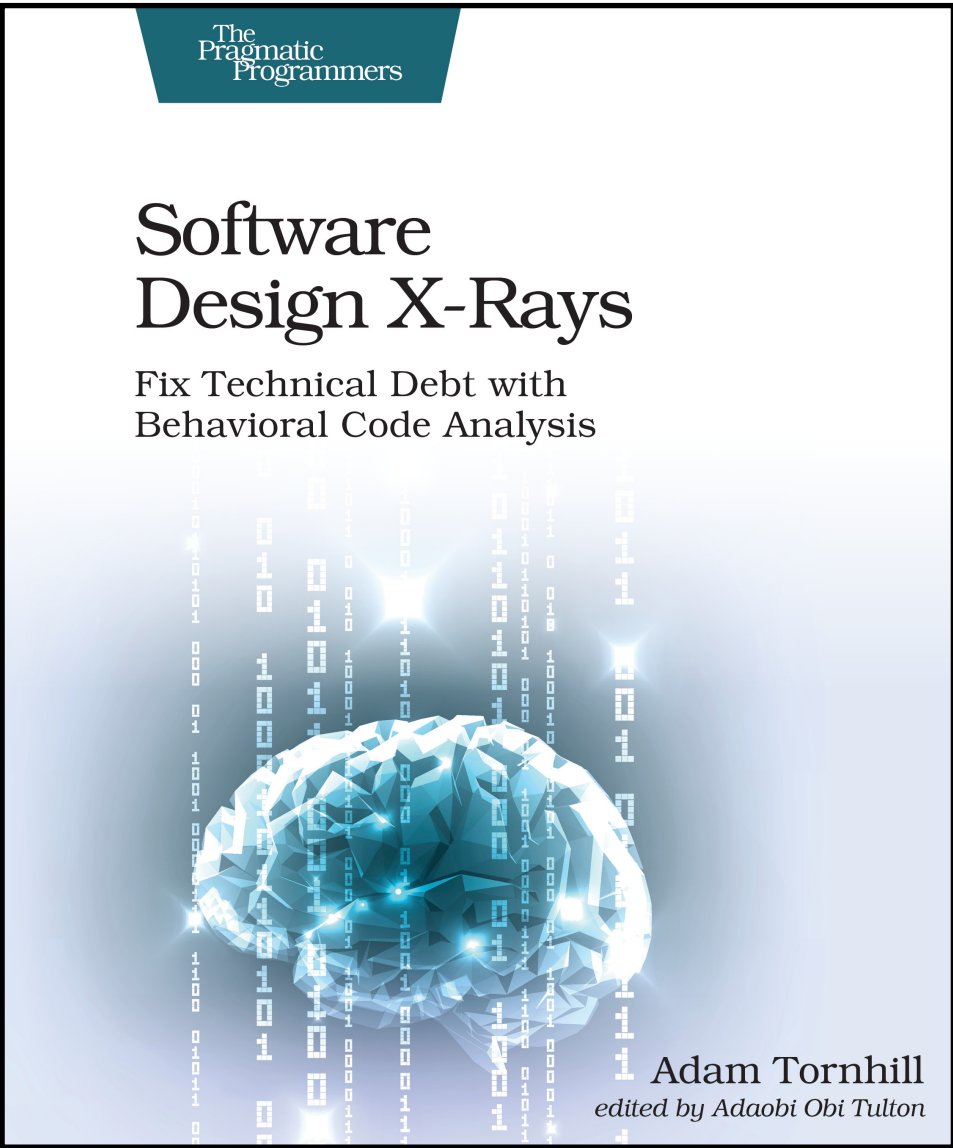
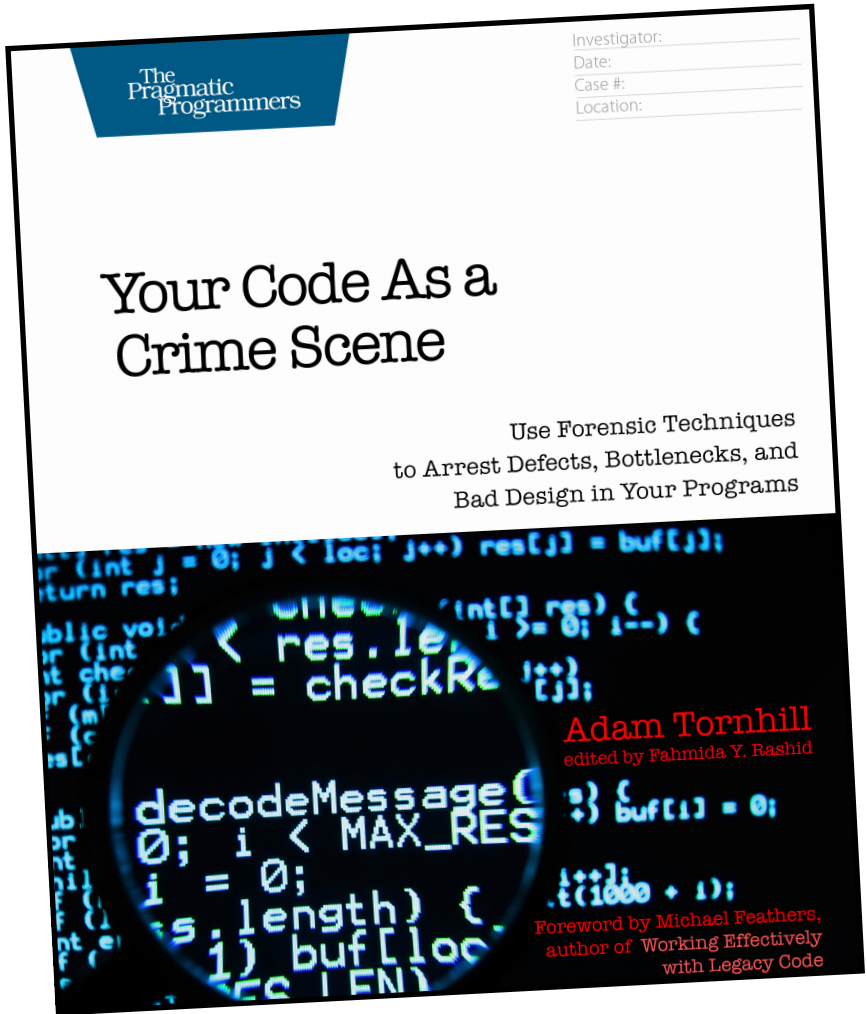


Quantifying Technical Debt Isn't Actionable.

There's always a trade off between improving existing code versus adding new features.

So how do we prioritise?

Version-Control — A Behavioral Data Source



Social Information

Commit: b557ca5
Date: 2016-02-12
Author: Kevin Flynn

Fix behavior of StartsWithPrefix

027

src/Mvc.Abstractions/ModelBinding/ModelStateDictionary.cs
src/Mvc.Core/ControllerBase.cs
src/Mvc.Core/Internal/ElementalValueProvider.cs
139 src/Mvc.Core/Internal/PrefixContainer.cs

Commit: fd6d28d
Date 2016-02-10
Author: Professor Falken

Make AddController not overwrite existing IControllerTypeProvider

814813

src/Core/Internal/ControllersAsServices.cs
test/Core.Test/Internal/ControllerAsServicesTest.cs
test/Mvc.FunctionalTests/ControllerFromServicesTests.cs

Commit: 910f013
Date :2016-02-05
Author Lisbeth Salander

Fixes #4050: Throw an exception when media types are empty.

201

src/Mvc.Core/Formatters/InputFormatter.cs

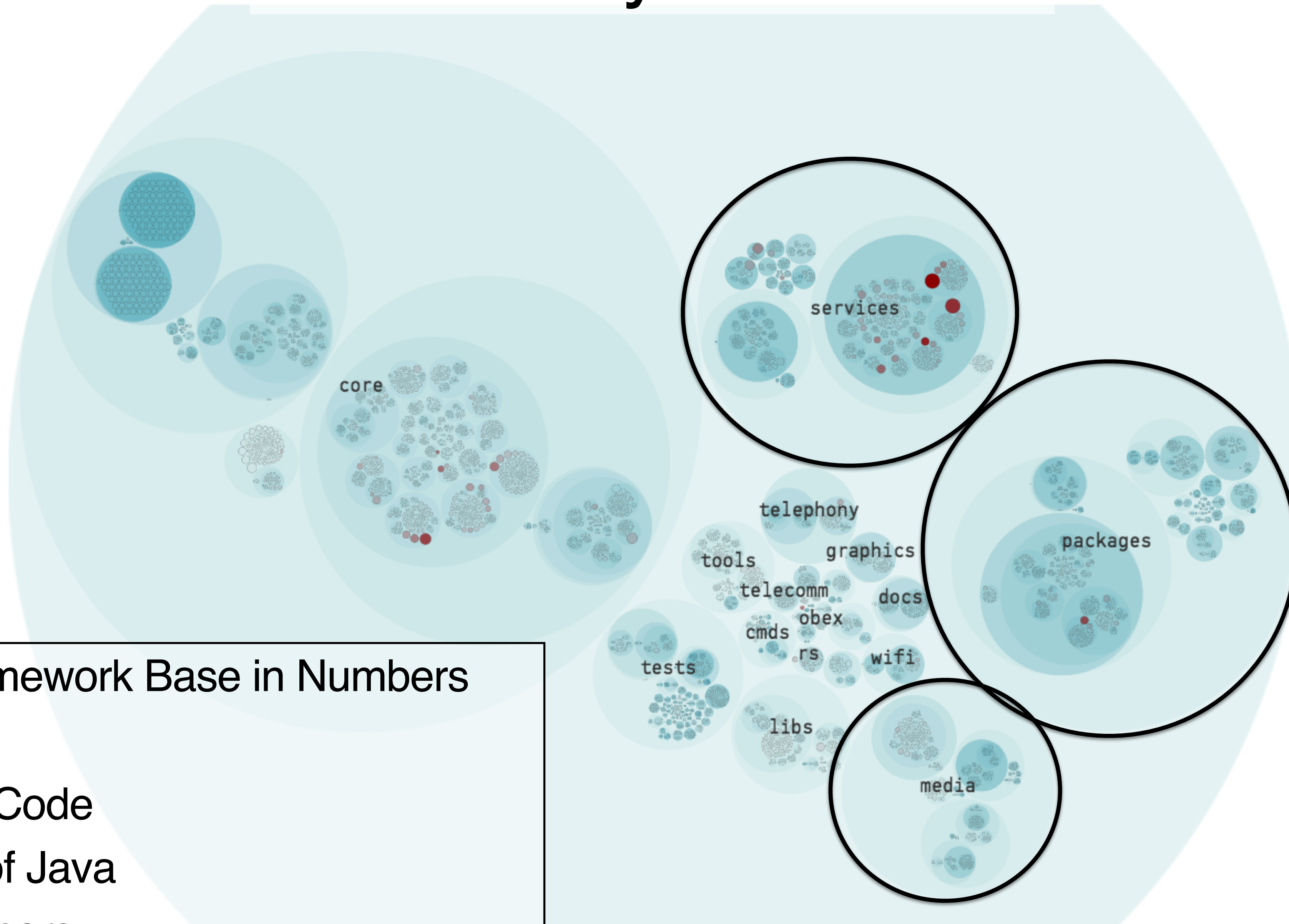
A Time Dimension

CASE STUDY: PRIORITIZING TECHNICAL DEBT

CodeScene™

Powered by Empear

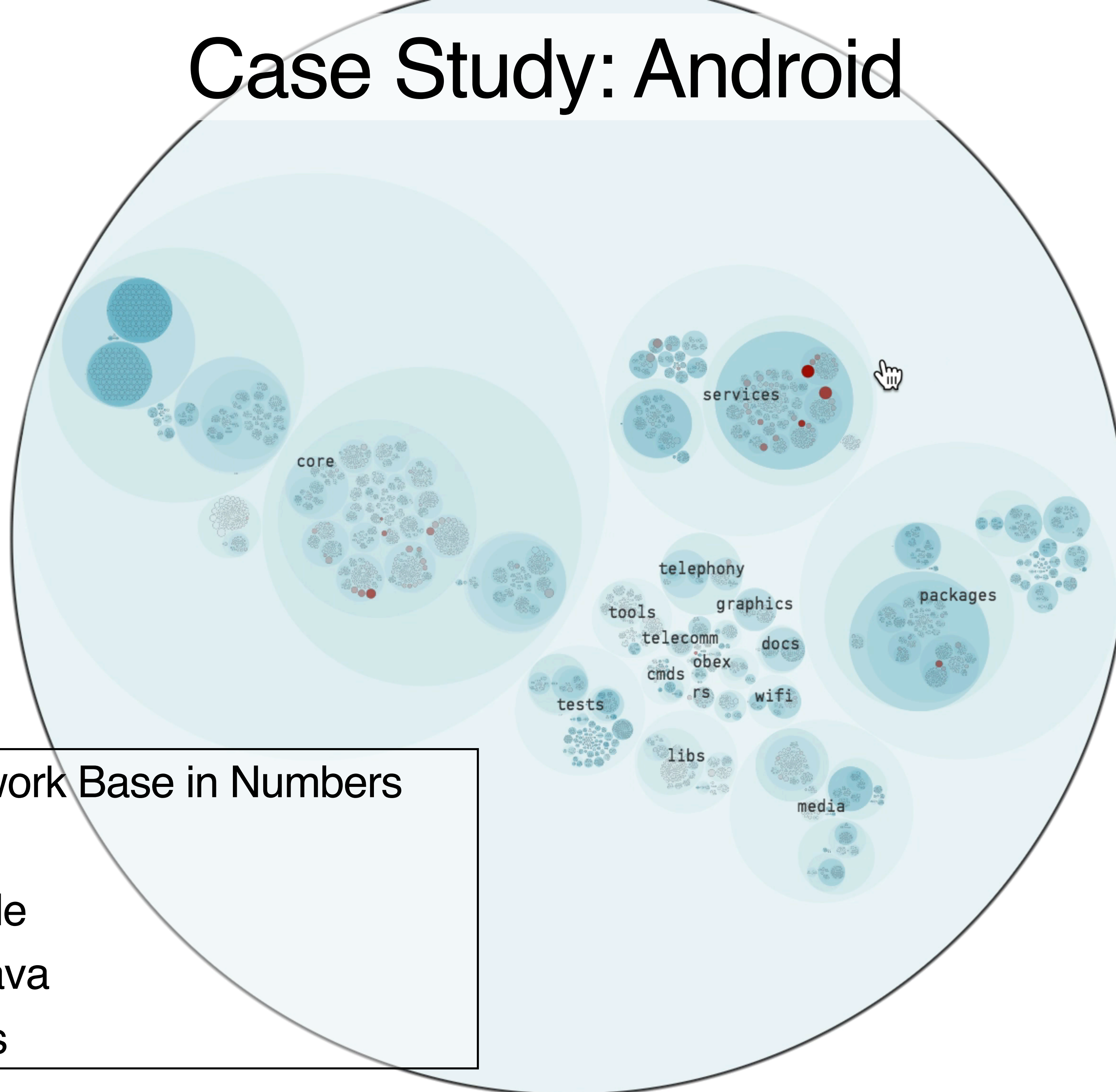
Case Study: Android



The Platform Framework Base in Numbers

3 Million Lines of Code
2,1 Million Lines of Java
2,000 Unique Authors

Case Study: Android



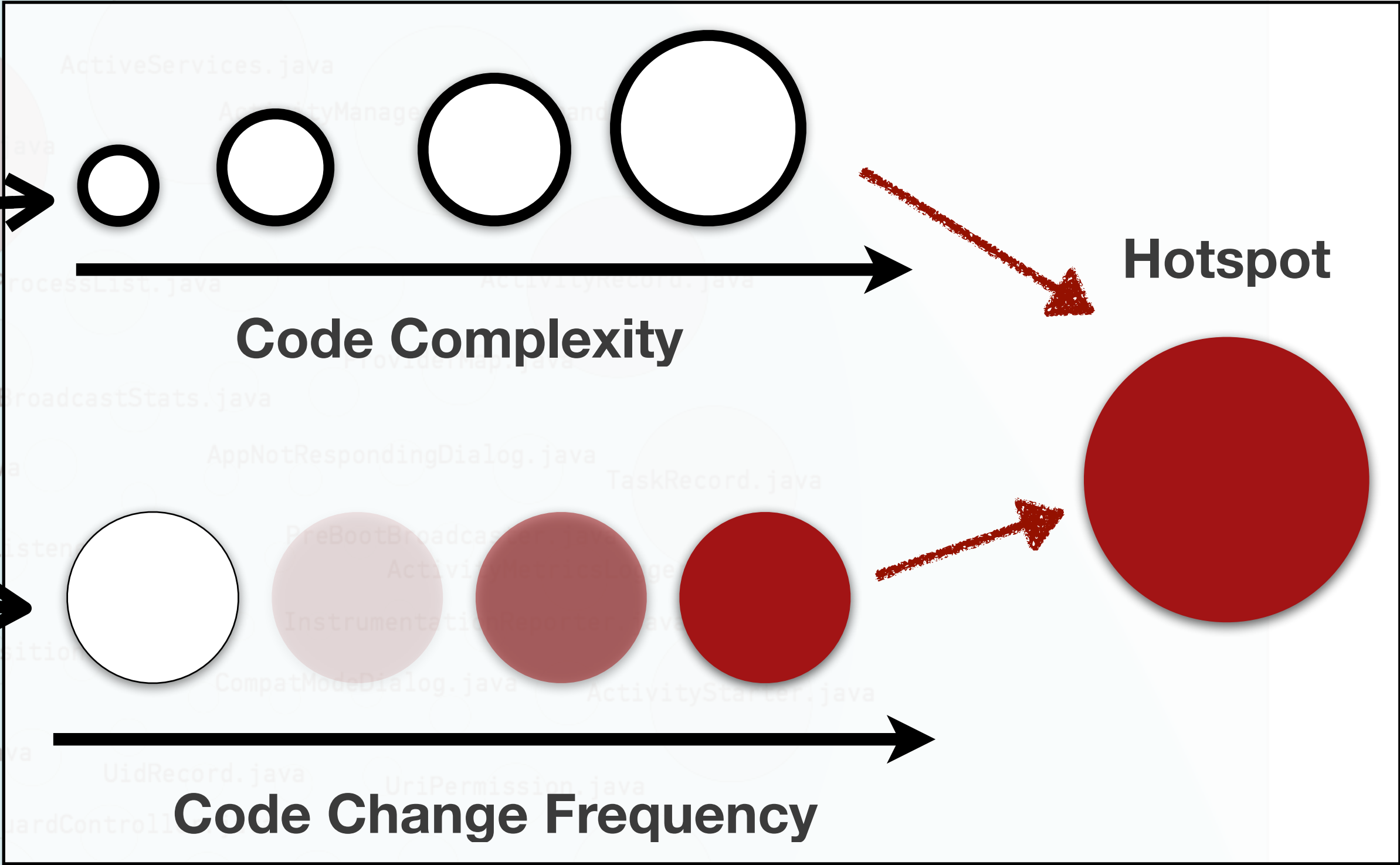
The Platform Framework Base in Numbers

3 Million Lines of Code
2,1 Million Lines of Java
2,000 Unique Authors

Principal

ActivityManagerService.java

Interest Rate



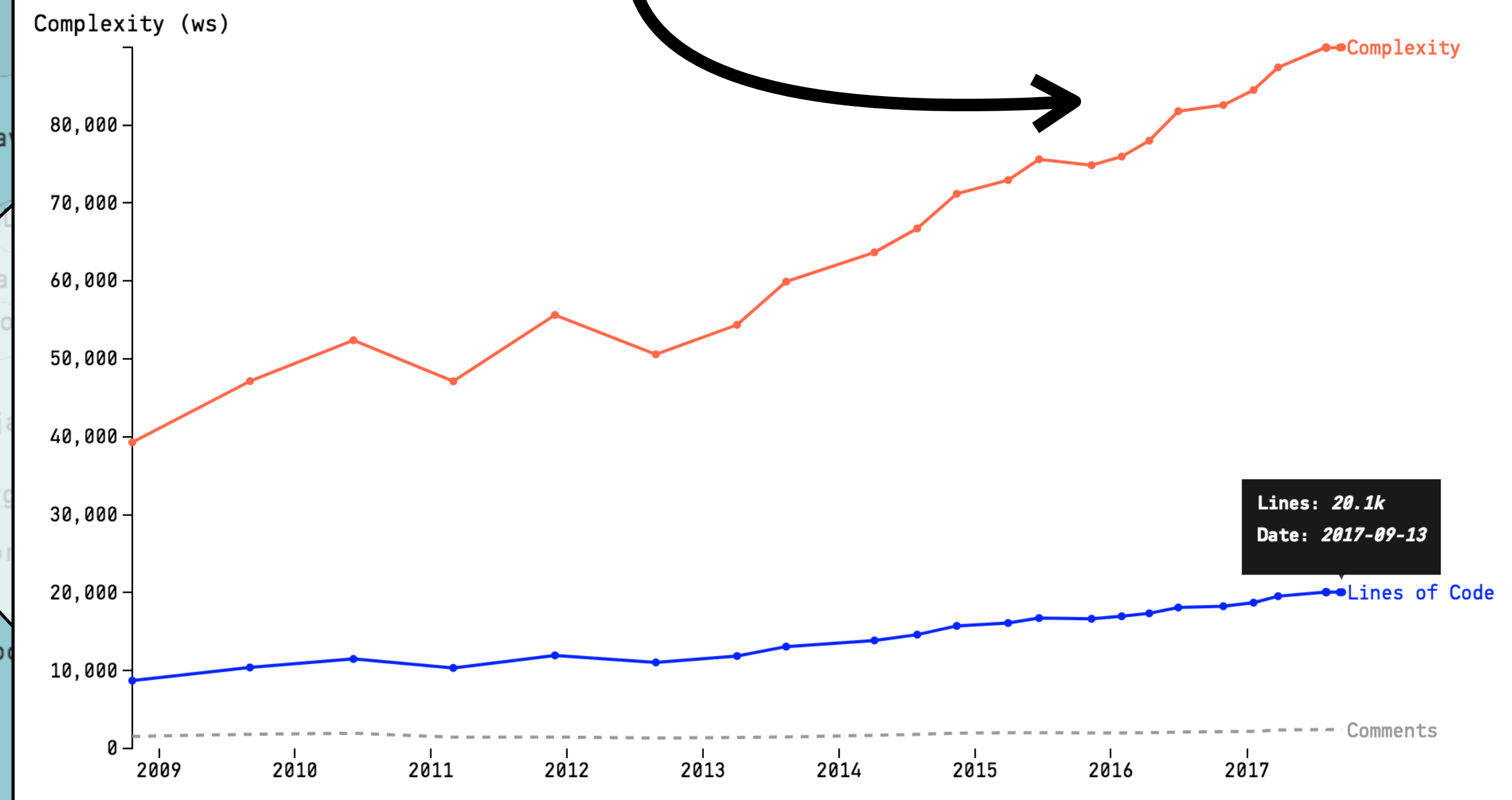
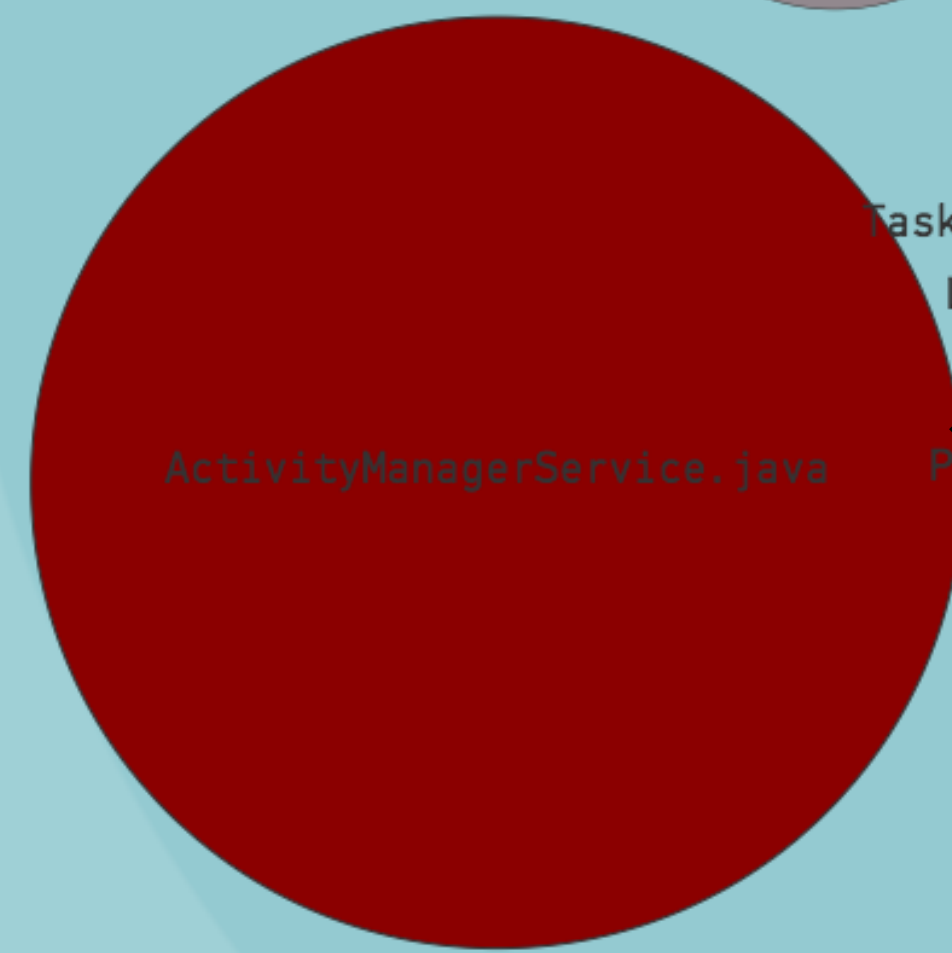
Symptoms of low Code Health



Low Cohesion, many responsibilities
Overall Complex Methods, many conditionals
Deeply Nested Logic, if-statements inside if-statements
Primitive Obsession, missing a domain language
Excess Function Arguments, missing abstractions

A Growing Problem

ActivityManagerService.java



Actionable Insights?

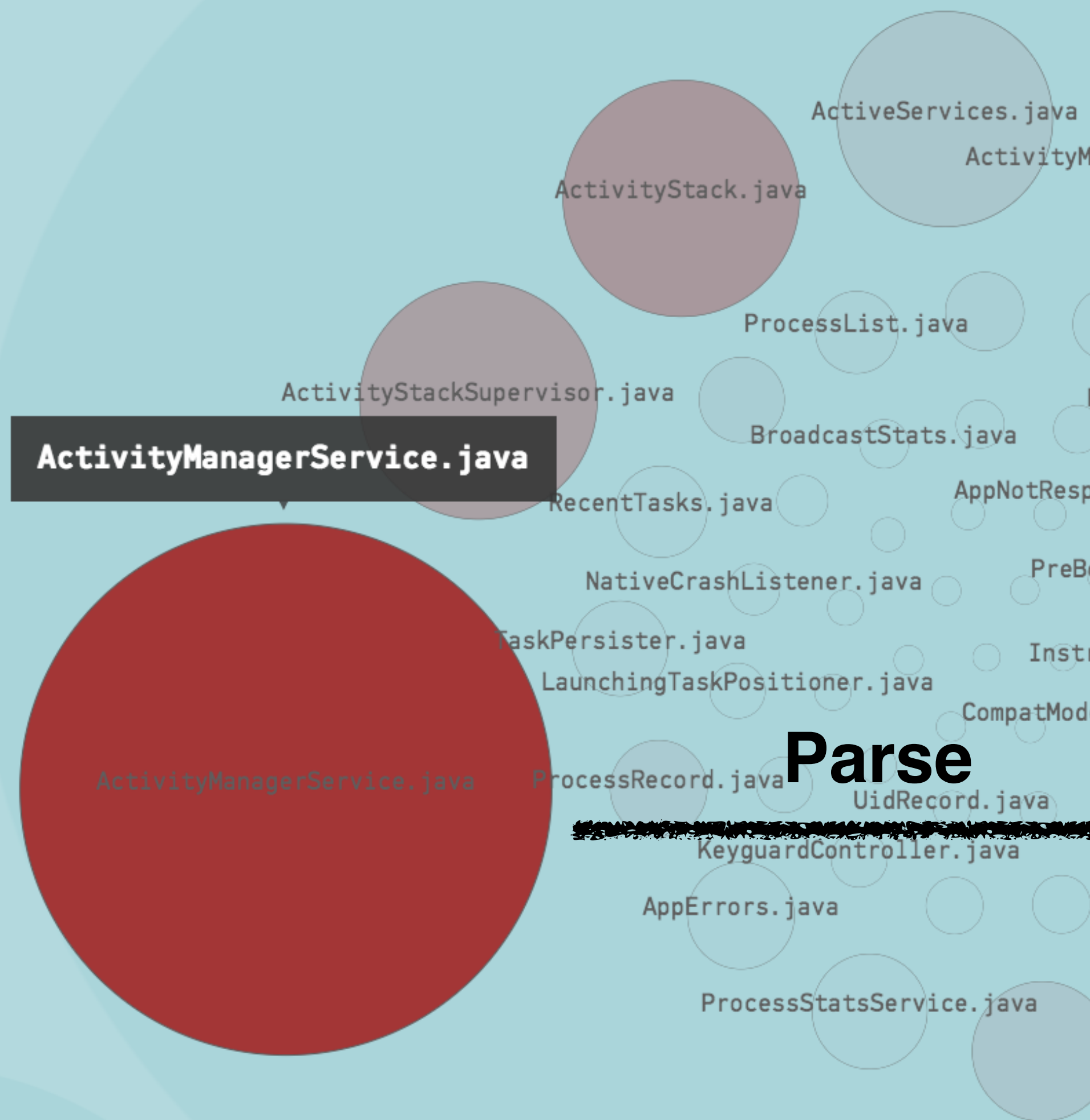
ActivityManagerService.java

20,000 Lines of Code!

74 Developers over the past 3 Months

Where Do We Start?

Hotspots: X-Ray `ActivityManagerService.java`

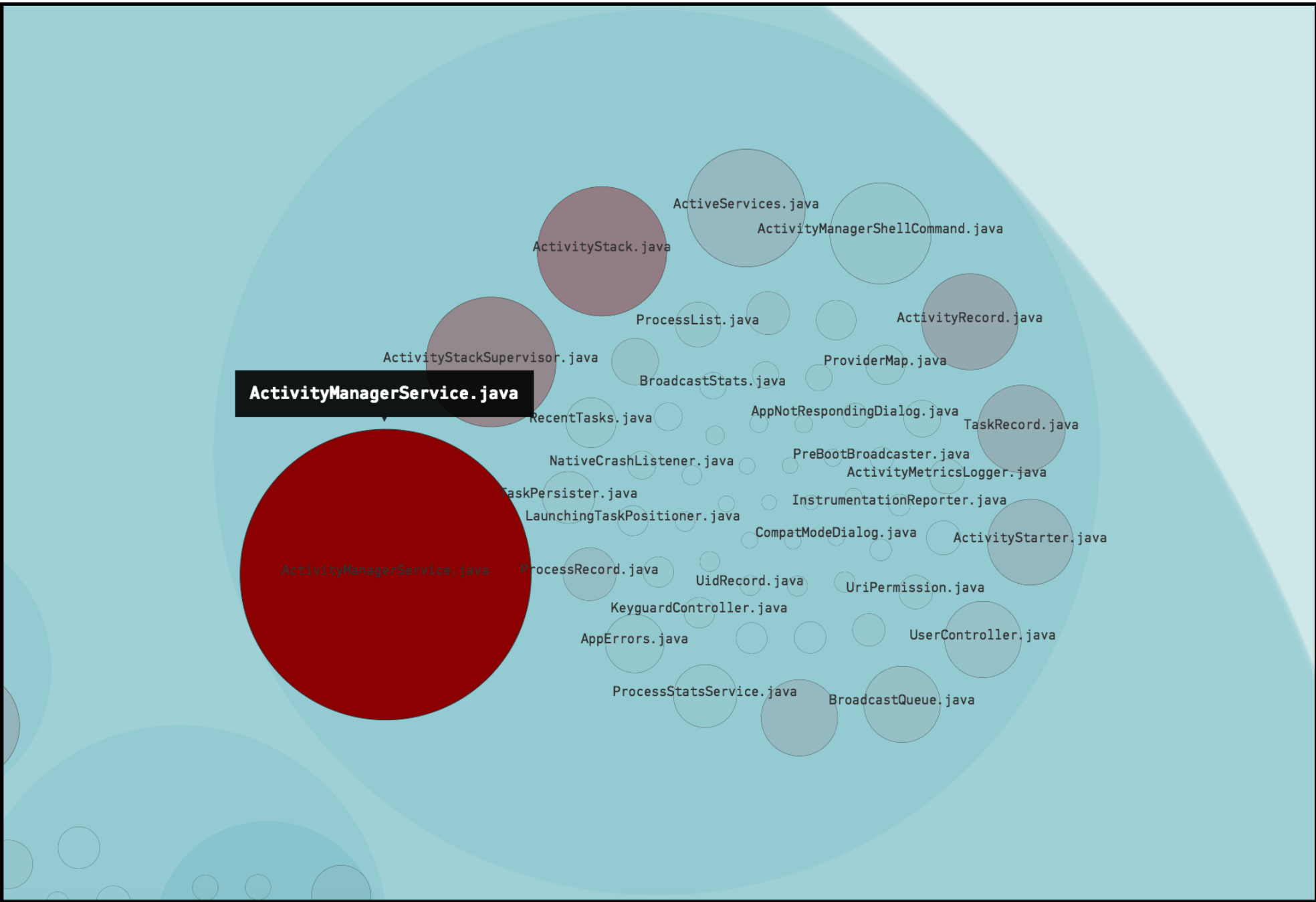


Function Level Hotspots



Recommended functions to improve.

X-Ray of ActivityManagerService.java



◆ Function	◆ Change ◆ Frequency	◆ Lines of Code	◆ Cyclomatic Complexity
ActivityManagerService.MainHandler.handleMessage	98	500	106
ActivityManagerService	75	160	12
applyOomAdjLocked	73	256	72
dumpStackTraces	73	188	40
dumpProcessesLocked	69	430	120
broadcastIntentLocked	60	647	171
enterPictureInPictureMode	60	71	7

Why Hotspots Work

CodeScene™

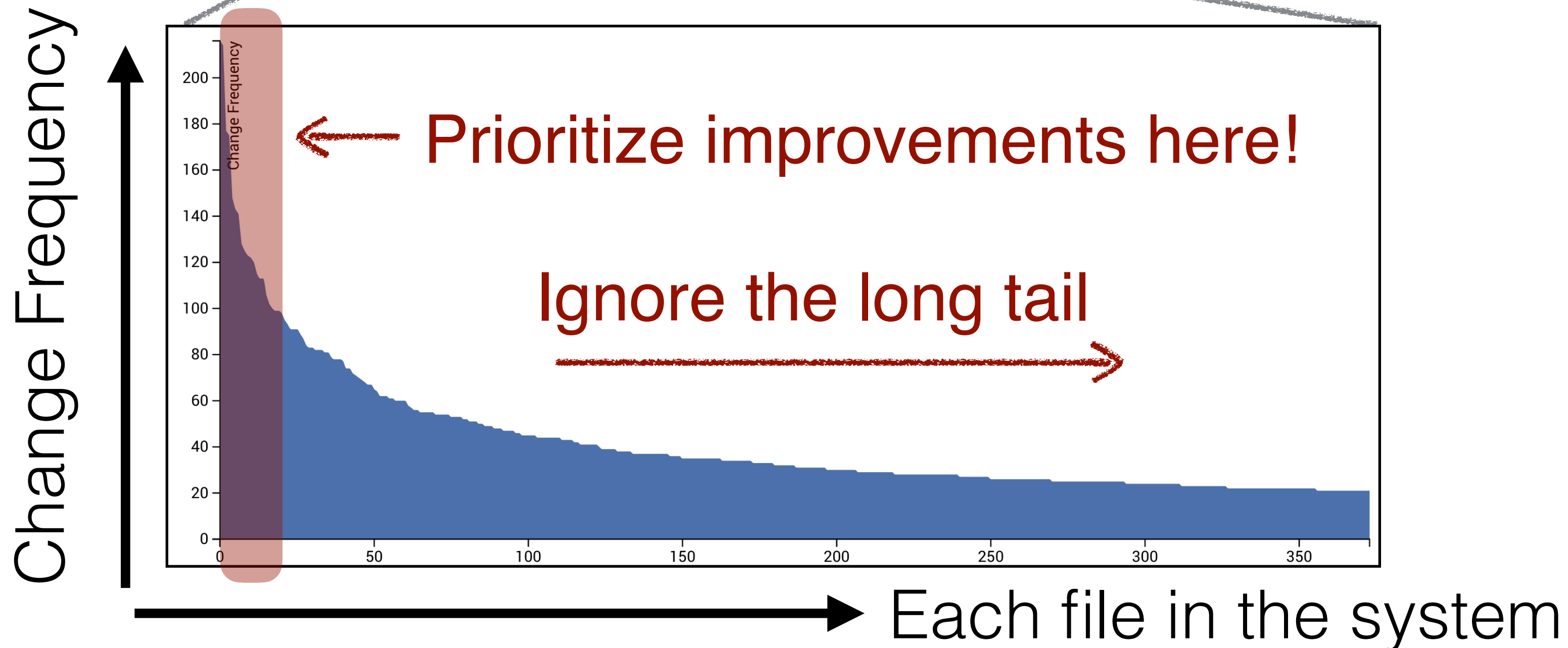
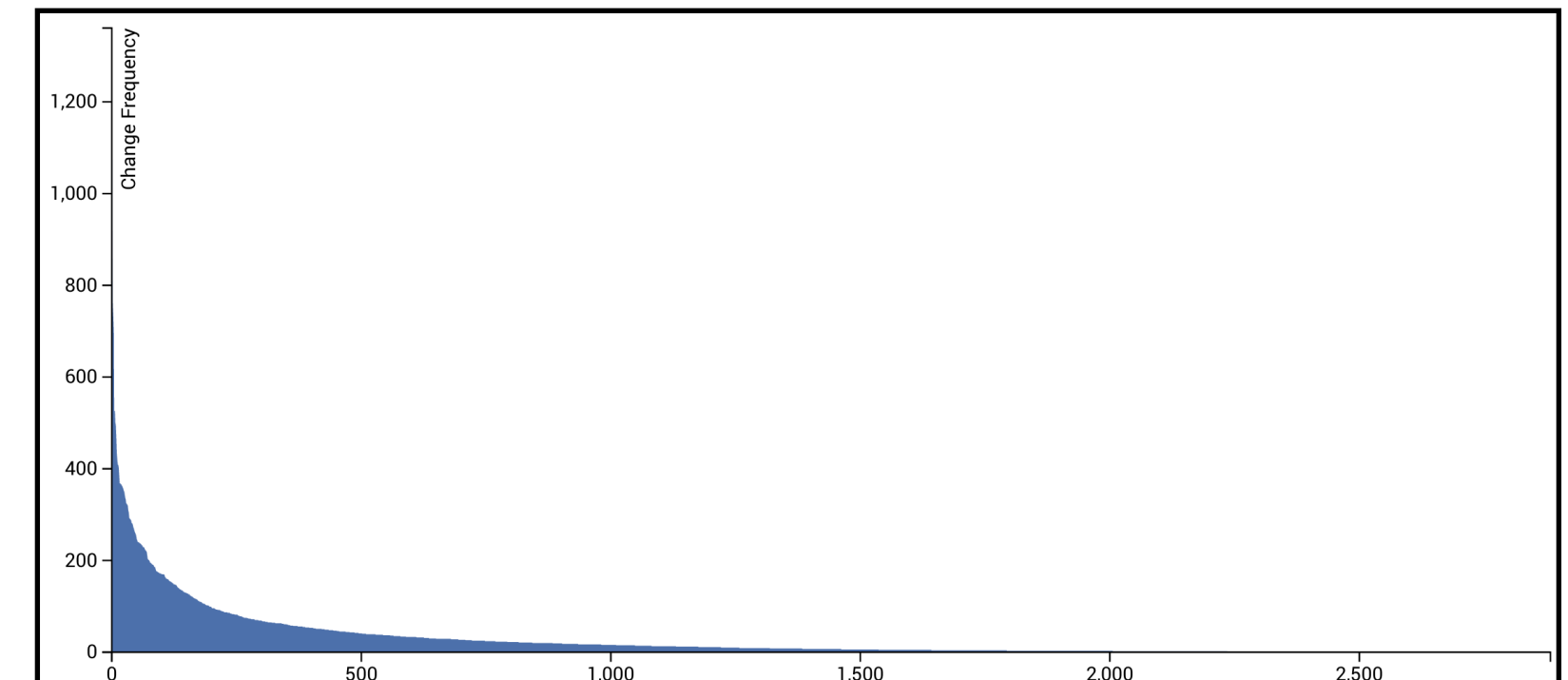
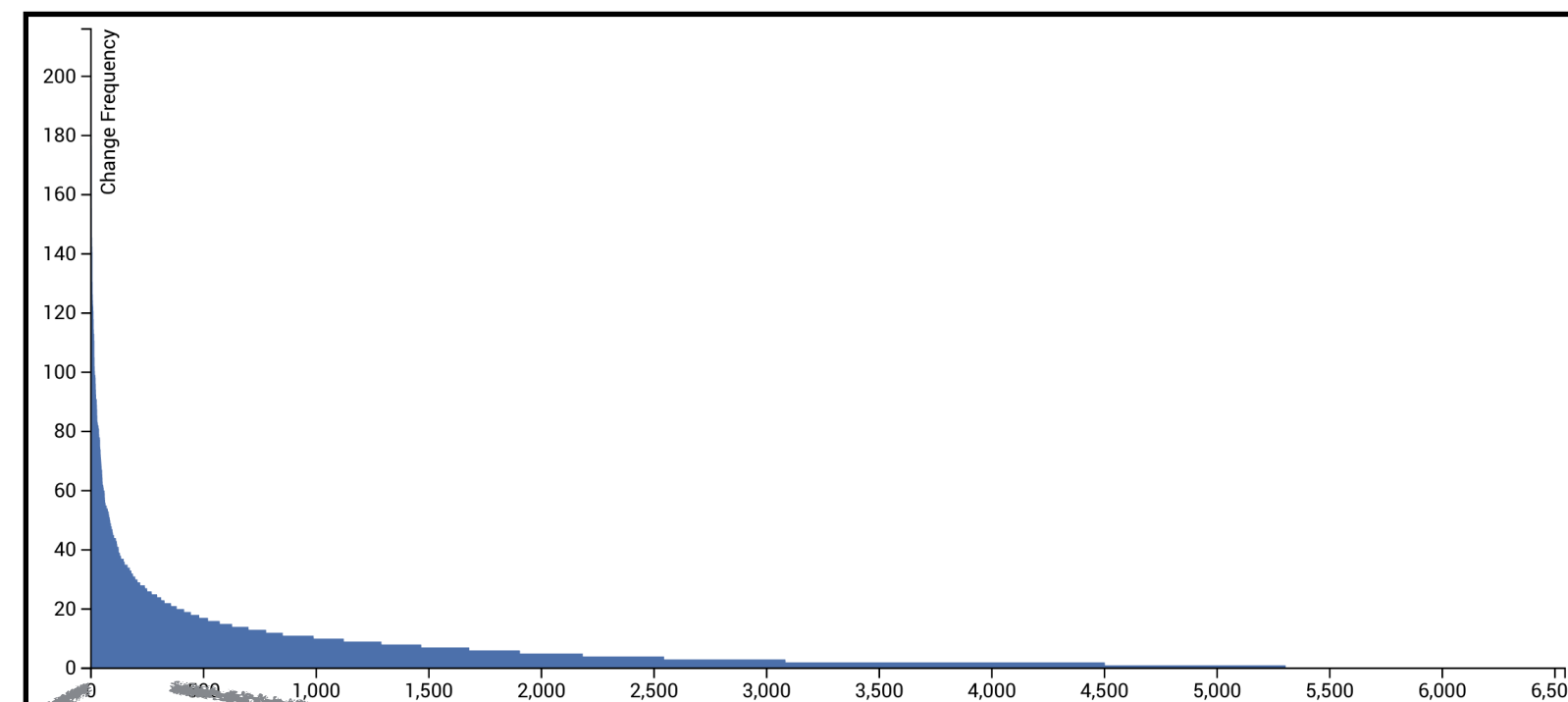
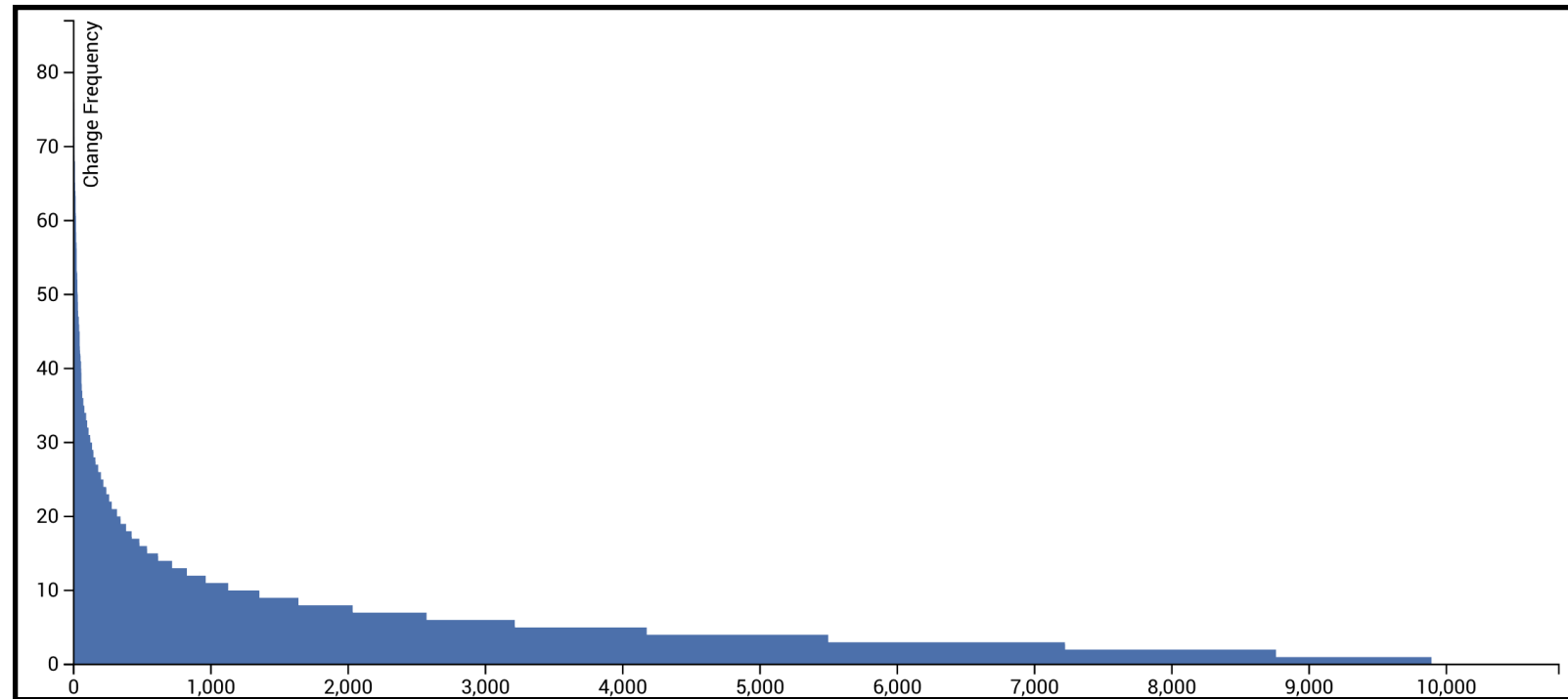
Powered by Empear

Why You Don't Have To Fix All Technical Debt

1 Year in Roslyn (C#, VB)

6 Years of Erlang

12 Years of Ruby on Rails



Code Quality In Context: Why you shouldn't fix all code issues

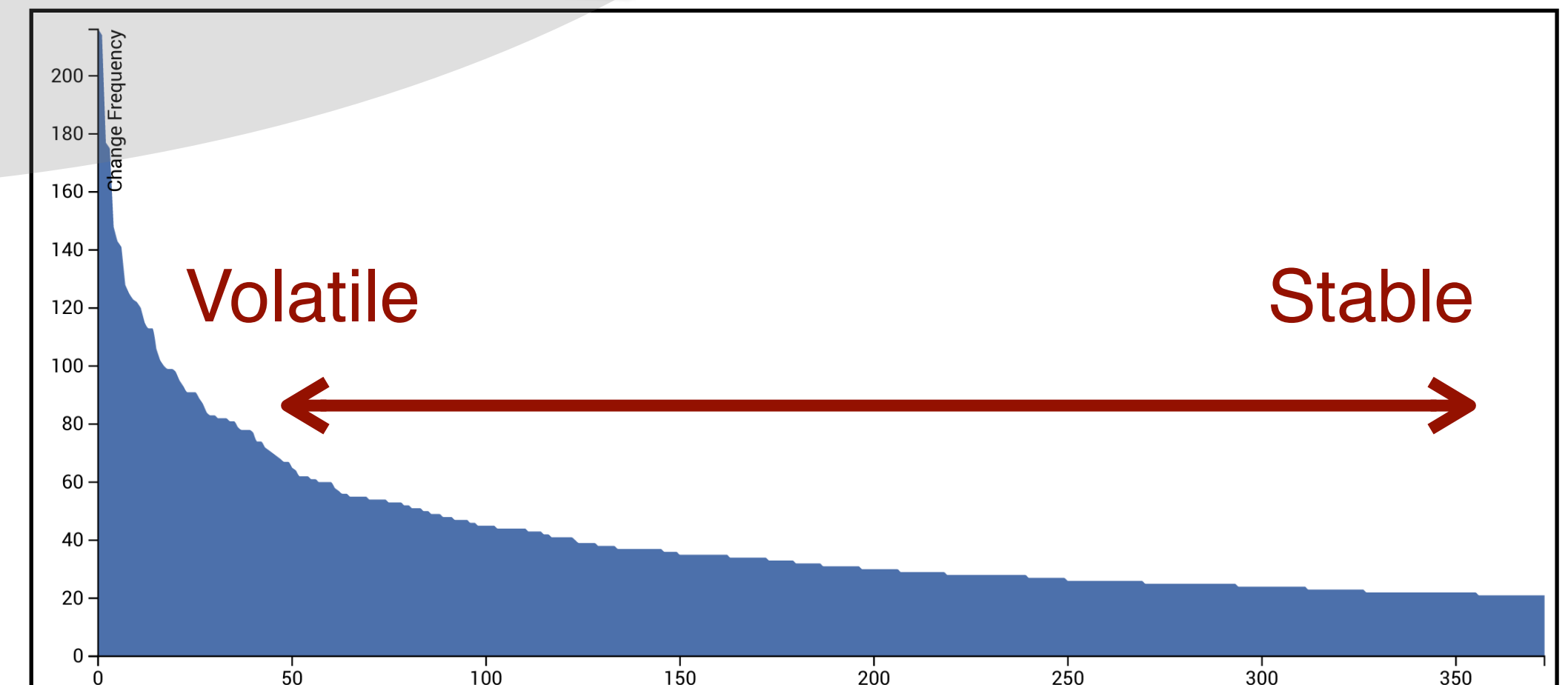
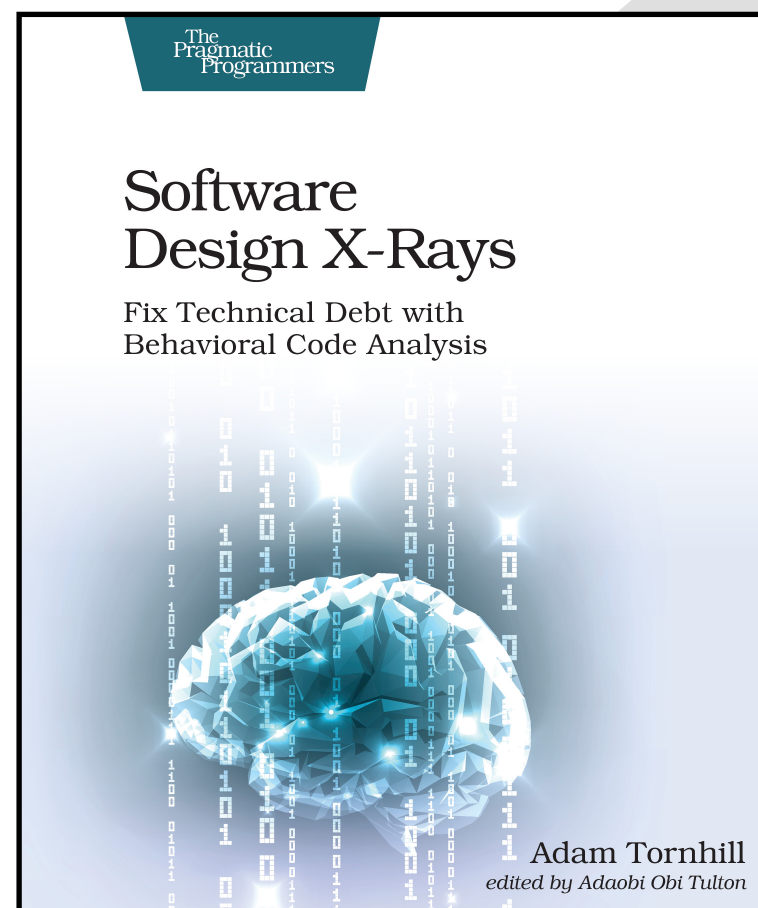
Your Best Bug Fix Is Time

The Three Generations of Code • 79

code. A team of researchers noted that a module that is a year older than a similar module has roughly one-third fewer faults. (See [Predicting fault incidence using software change history \[GKMS00\]](#).) The passage of time is like a quality verdict, as it exposes modules to an increasing number of use cases and variations. Defective modules have to be corrected. And since bug fixes themselves, ironically, pose a major risk of introducing new defects, the code has to be patched again and again. Thus, bugs breed bugs and it all gets reflected as code that refuses to stabilize and age.

Test Cases Don't Age Well

While old code is likely to be good code in the sense that it has low maintenance costs and low defect risk, the same reasoning doesn't



Read more: <https://adamtornhill.com/articles/code-quality-in-context/why-i-write-dirty-code.html>

The Legacy Code Link: Why Much Technical Debt Isn't Really Technical Debt

CodeScene™

Powered by Empear

What Is Legacy Code?

“Legacy Code” is typically used to describe code that:

- lacks in quality, and that
- we didn't write ourselves.

The Technical Debt That Wasn't

Product #1



Product #2



Product #3

?

Case Study:

How quick can you turn your current codebase into legacy code?

CodeScene™

Powered by Empear

@AdamTornhill

Case Study: Off-Boarding

Identify the main developers
behind each module

Commit: b557ca5
Date: 2016-02-12
Author: Kevin Flynn

Fix behavior of StartsWithPrefix

27 src/Mvc.Abstractions/ModelBinding/ModelStateDictionary.cs
10 src/Mvc.Core/ControllerBase.cs
1 src/Mvc.Core/Internal/ElementalValueProvider.cs
1 39 src/Mvc.Core/Internal/PrefixContainer.cs

Commit: fd6d28d
Date 2016-02-10
Author: Professor Falken

Make AddController not overwrite existing IControllerTypeProvider

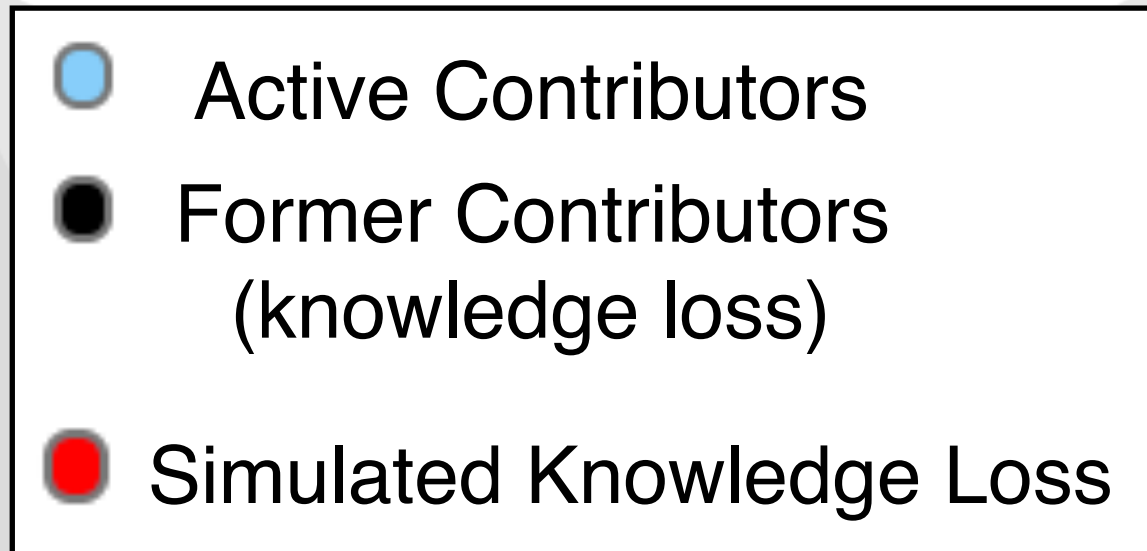
8 1 src/Core/Internal/ControllersAsServices.cs
48 0 test/Core.Test/Internal/ControllerAsServicesTest.cs
13 0 test/Mvc.FunctionalTests/ControllerFromServicesTests.cs

Commit: 910f013
Date :2016-02-05
Author Lisbeth Salander

Fixes #4050: Throw an exception when media types are empty.

20 1 src/Mvc.Core/Formatters/InputFormatter.cs

Case Study: ASP.NET MVC Core



Application Code

Test Code

src
benchmarkapps
benchmarks

test

~180 Contributors
350,000 Lines of Code
C#

Microsoft.AspNetCore.Mvc.IntegrationTests

test

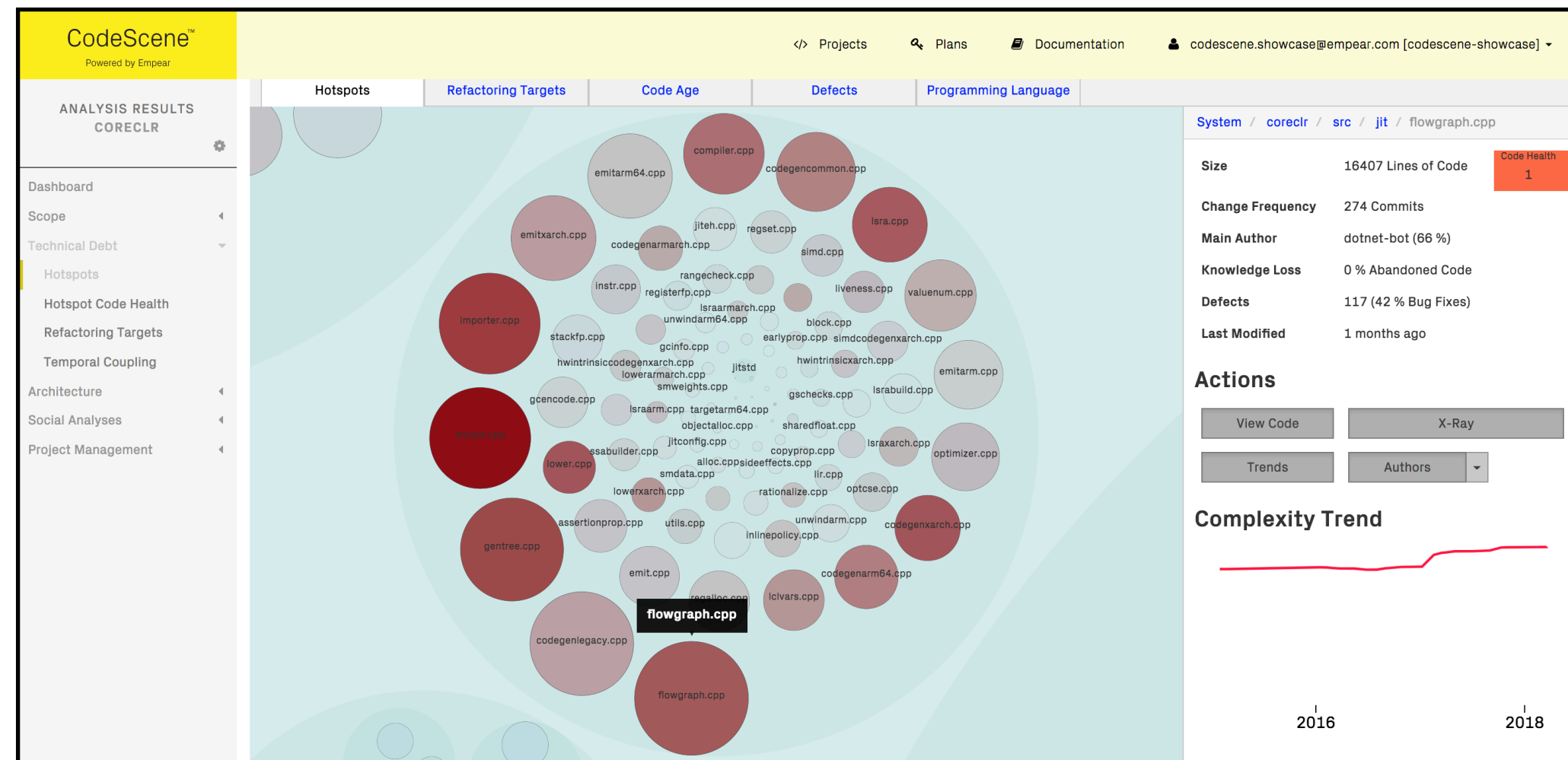
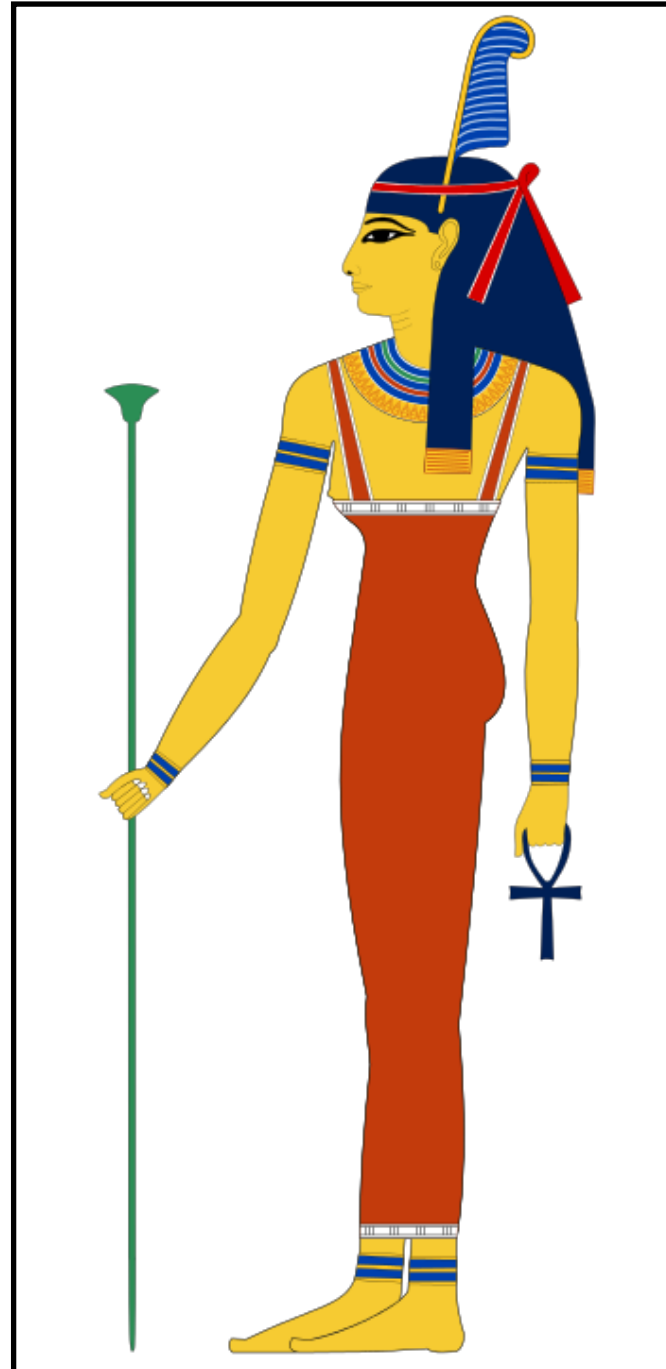
Code from <https://github.com/aspnet/Mvc>

A photograph of several large icebergs floating in a calm, blue ocean under a clear sky. The icebergs are white and jagged, with their reflections visible in the water. The text is overlaid on the top part of the image.

There's More to Code Complexity than Code

Social Factors Influence how we Perceive a Codebase

Tooling: Try it on your own Code



<https://codescene.io/>



Track functions with
`git log -L :<funcname>:<file>`

Source Code:

<https://github.com/adamtornhill/code-maat>

Microservices

Analysing Technical Debt and Non-Code Properties

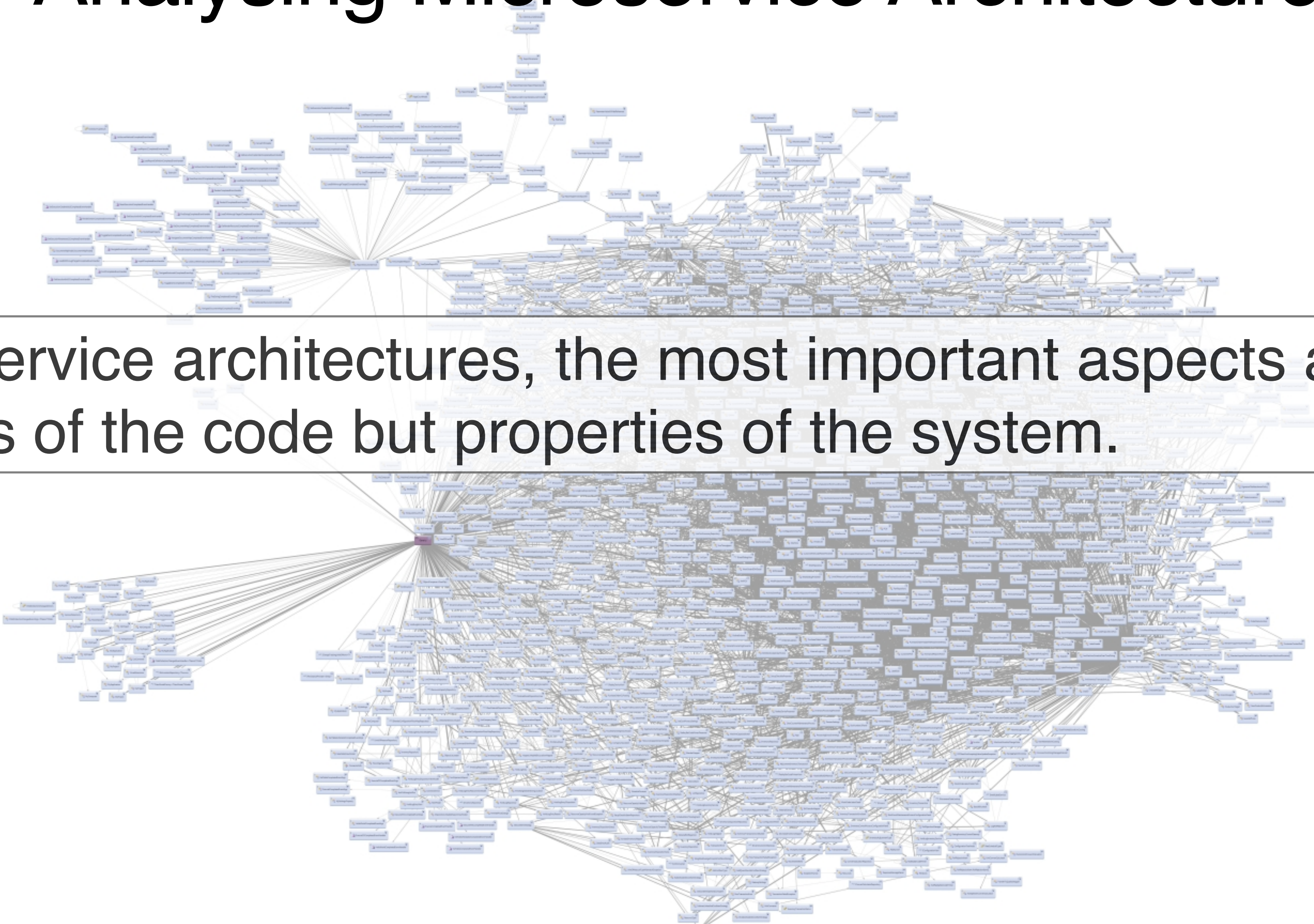
CodeScene™

Powered by Empear



Tomorrow's Legacy Code: Microservices

Analysing Microservice Architectures



In microservice architectures, the most important aspects are not properties of the code but properties of the system.

Hotspots in Spinnaker: +30 Git repos, 7 Languages

The **orca** git repository
(Orchestration Engine)

Kotlin, Java, Groovy

Client library

Go lang

Read/write operations for
Google cloud

Java

Read/write operations for aws

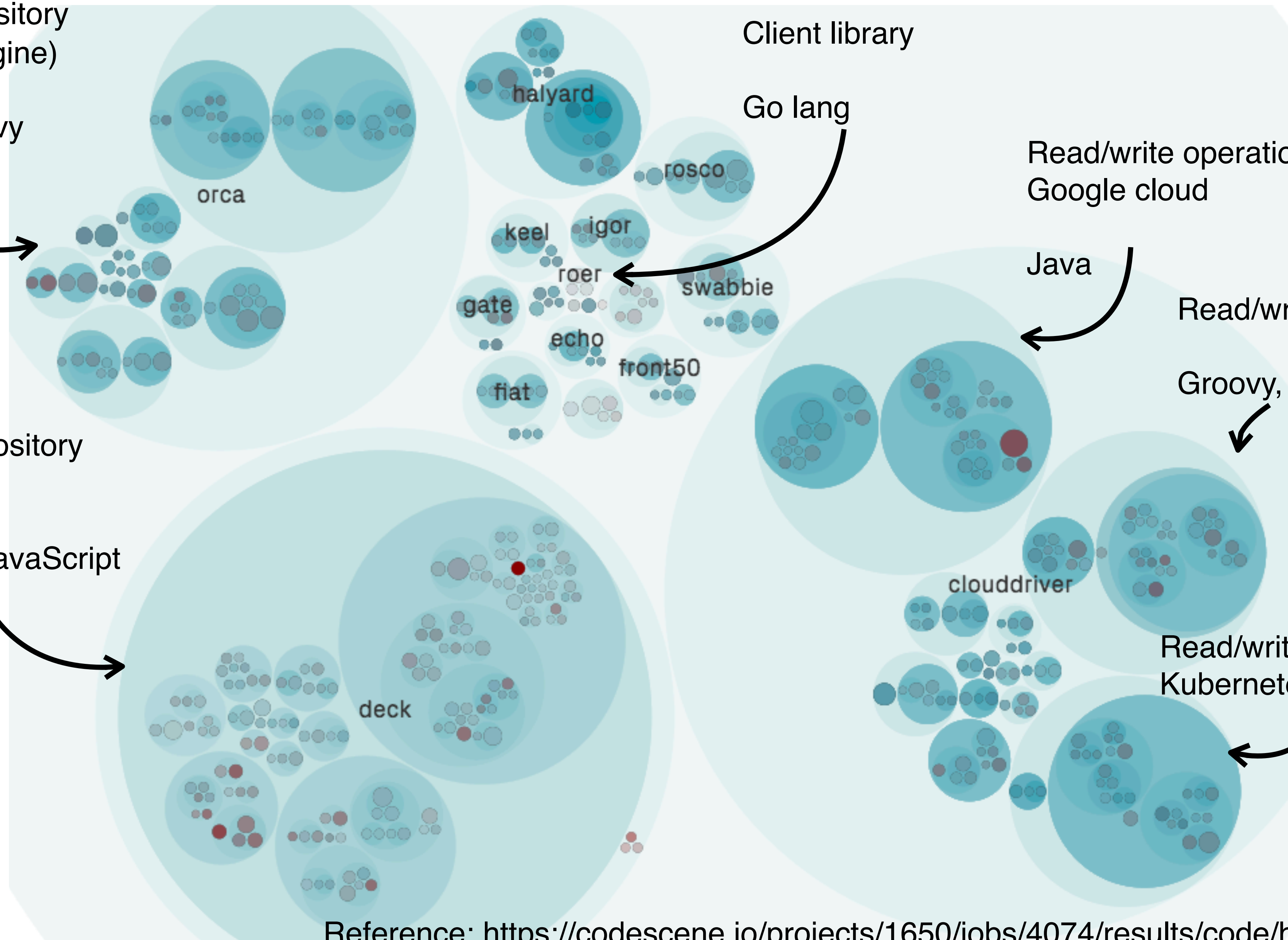
Groovy, Java

Read/write operations for
Kubernetes

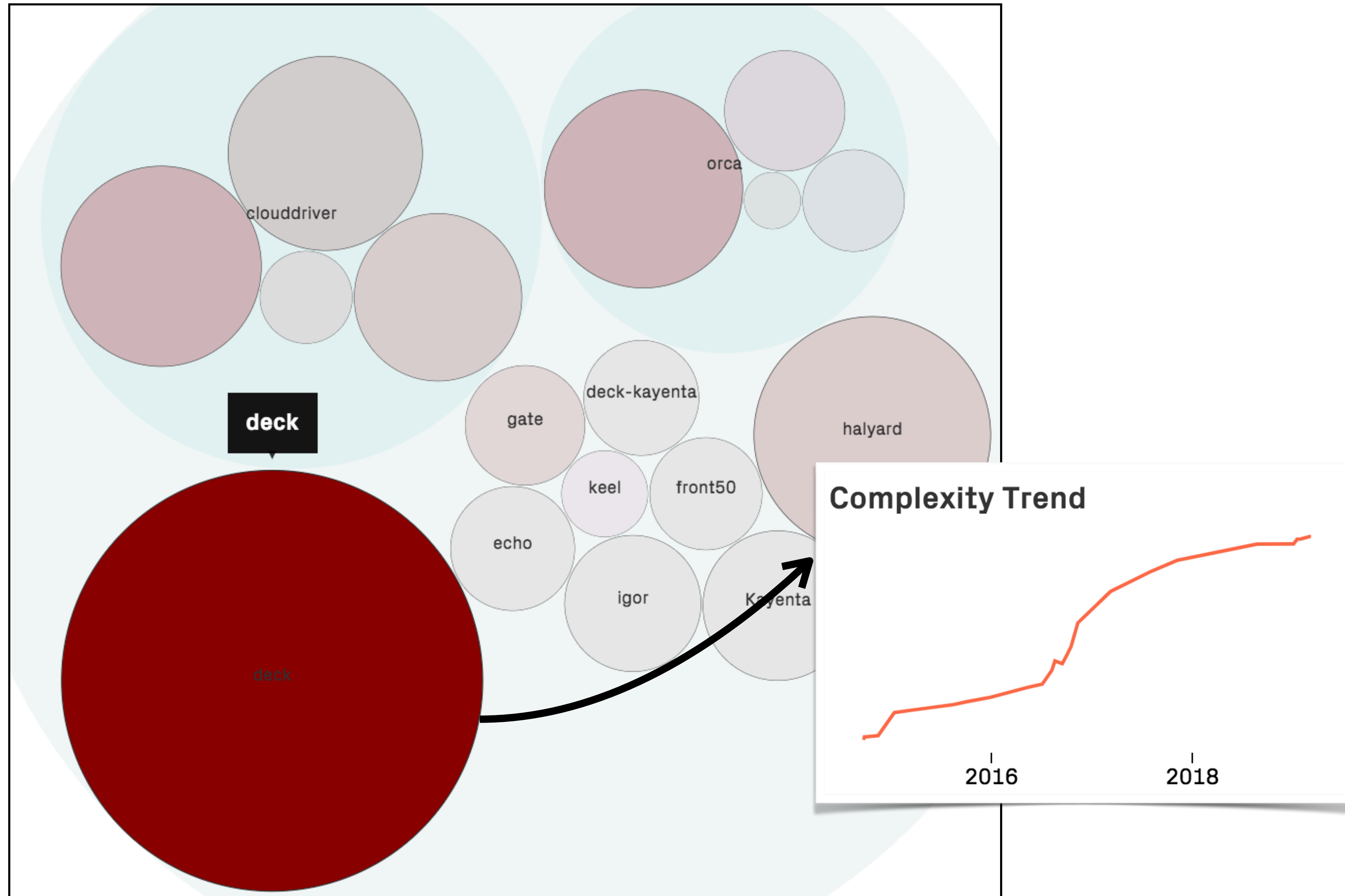
The **deck** git repository
(Management UI)

TypeScript and JavaScript

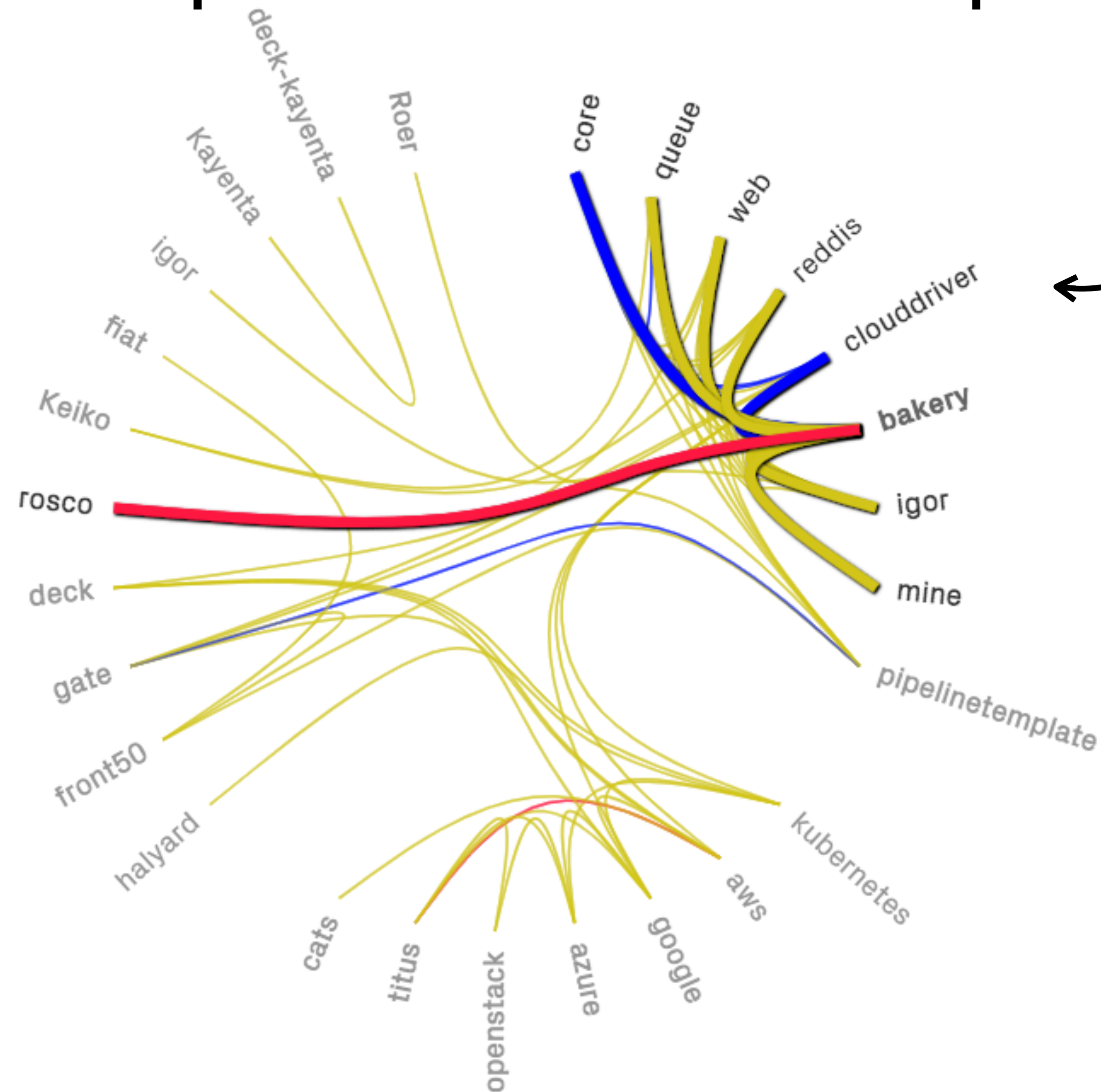
Reference: <https://codescene.io/projects/1650/jobs/4074/results/code/hotspots/system-map>



Aggregation: Architectural Hotspots in Spinnaker

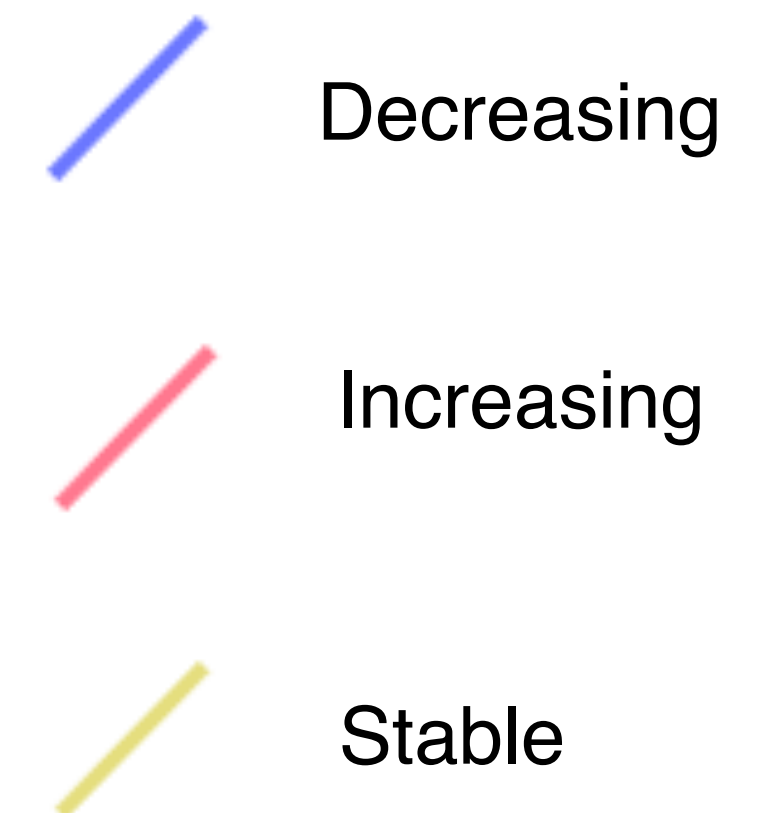


Microservice Dependencies: The Impact of Change

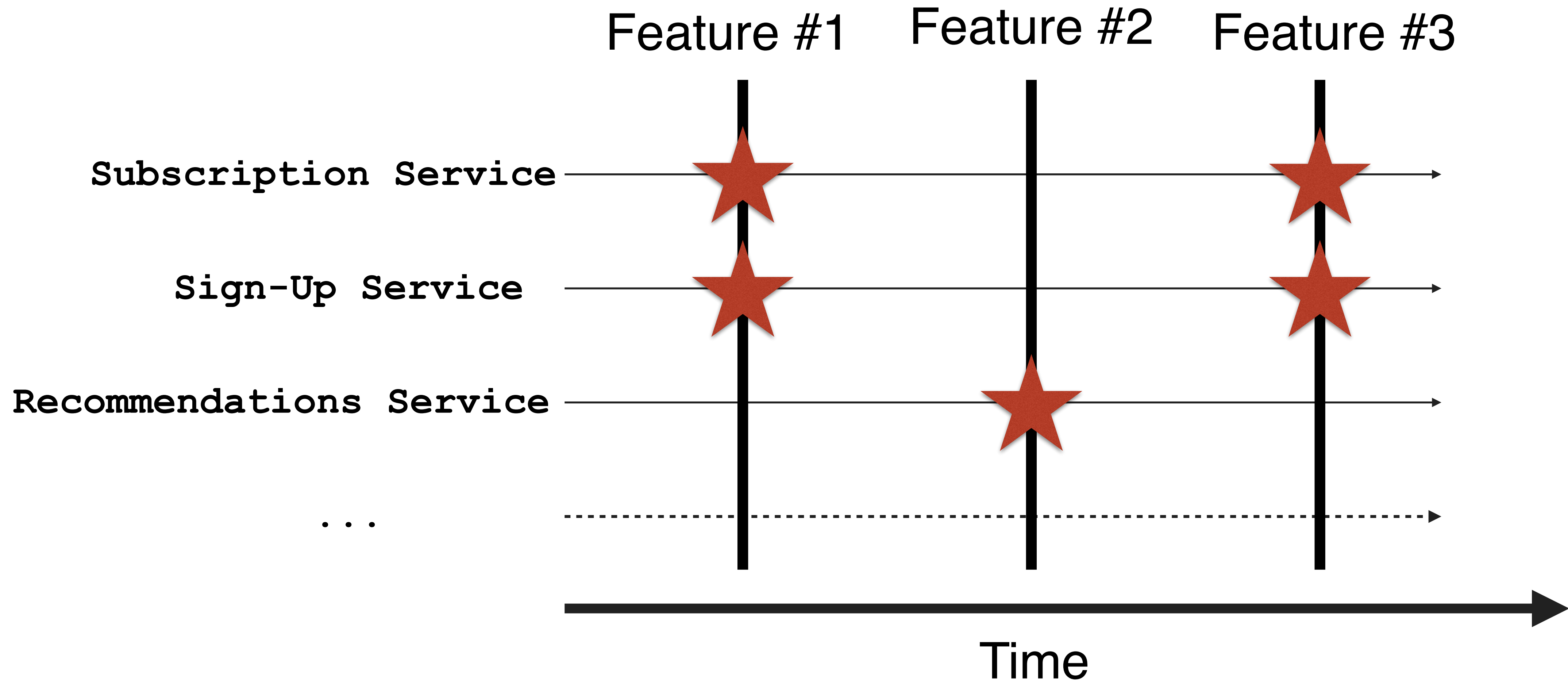


A cluster of micro services that are frequently modified when implementing new features or fixing bugs

Coupling Trends



Change Coupling

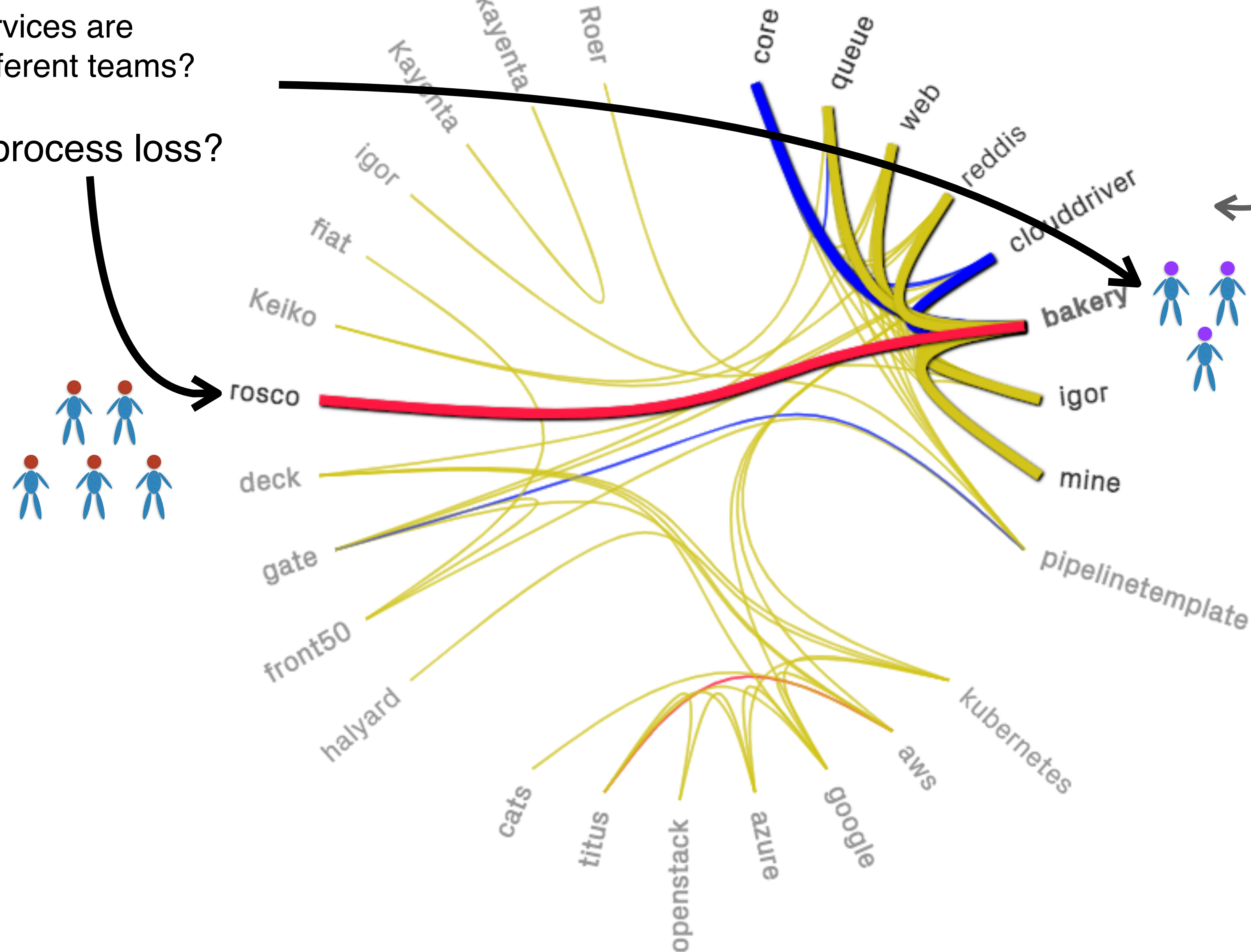


Microservice Dependencies: The Impact of Change

What if these services are developed by different teams?

What about the process loss?

A cluster of micro services that are frequently modified when implementing new features or fixing bugs



Coupling Trends

Decreasing

Increasing

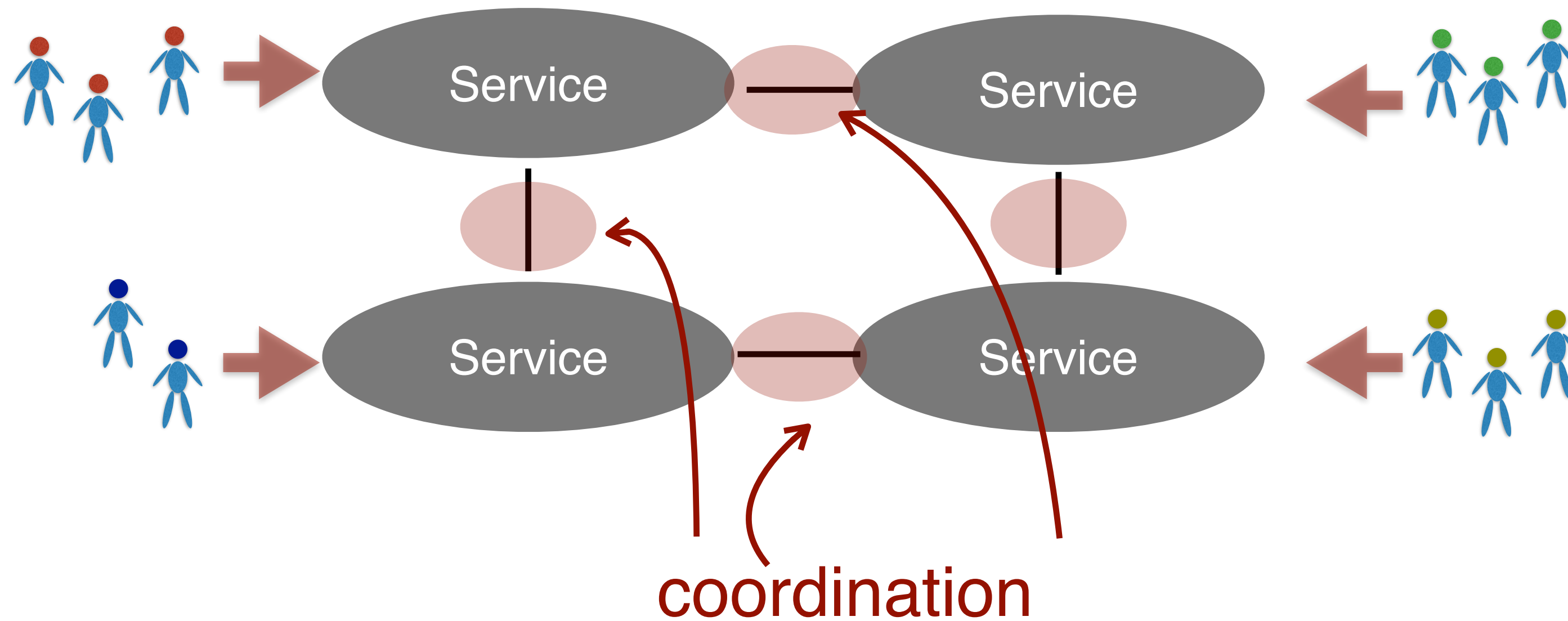
Stable

Change Coupling: Component or Feature Teams?

— Pick Your Poison —

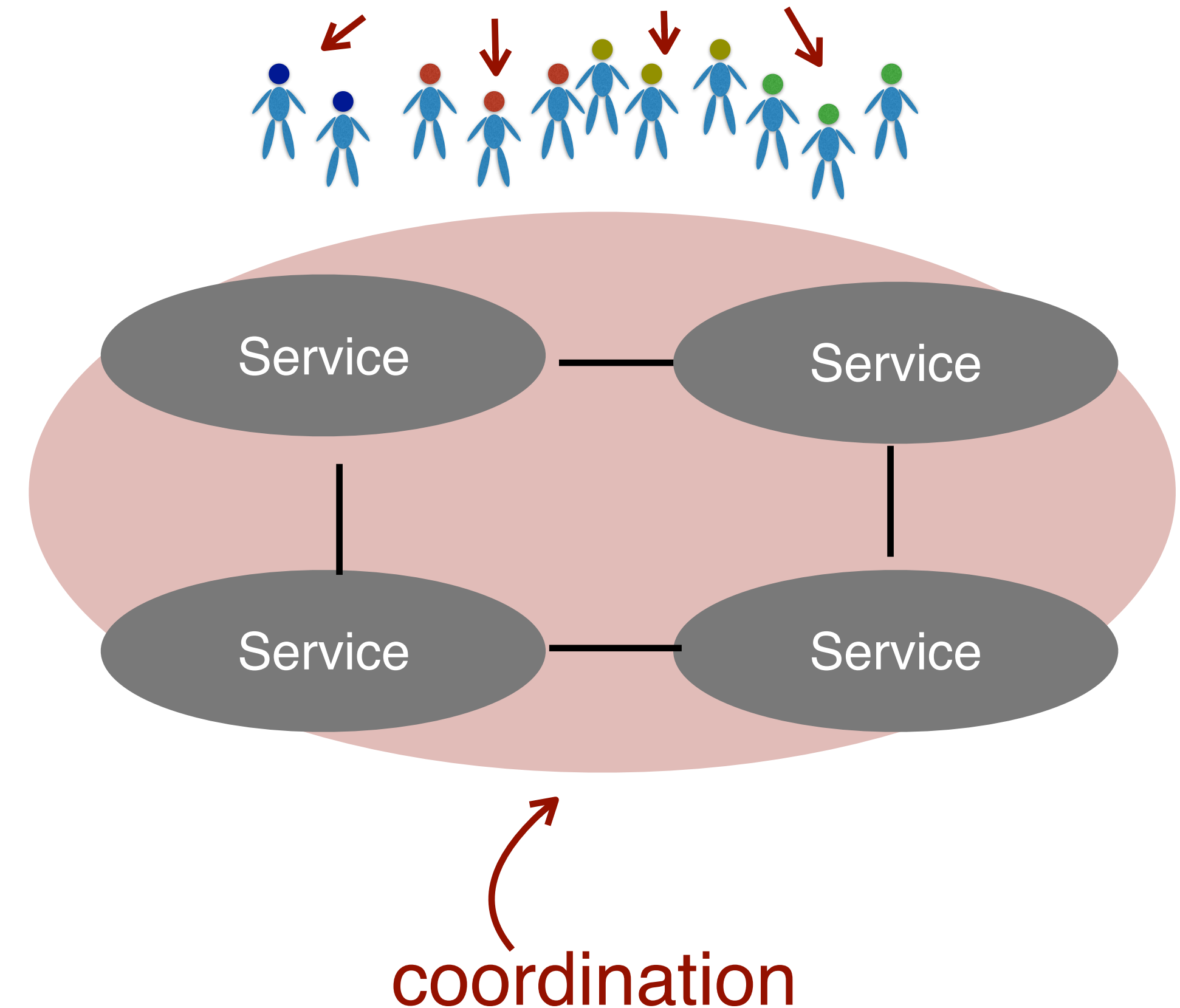
Component/Domain Teams

One team per service makes each boundary a coordination point => long lead times

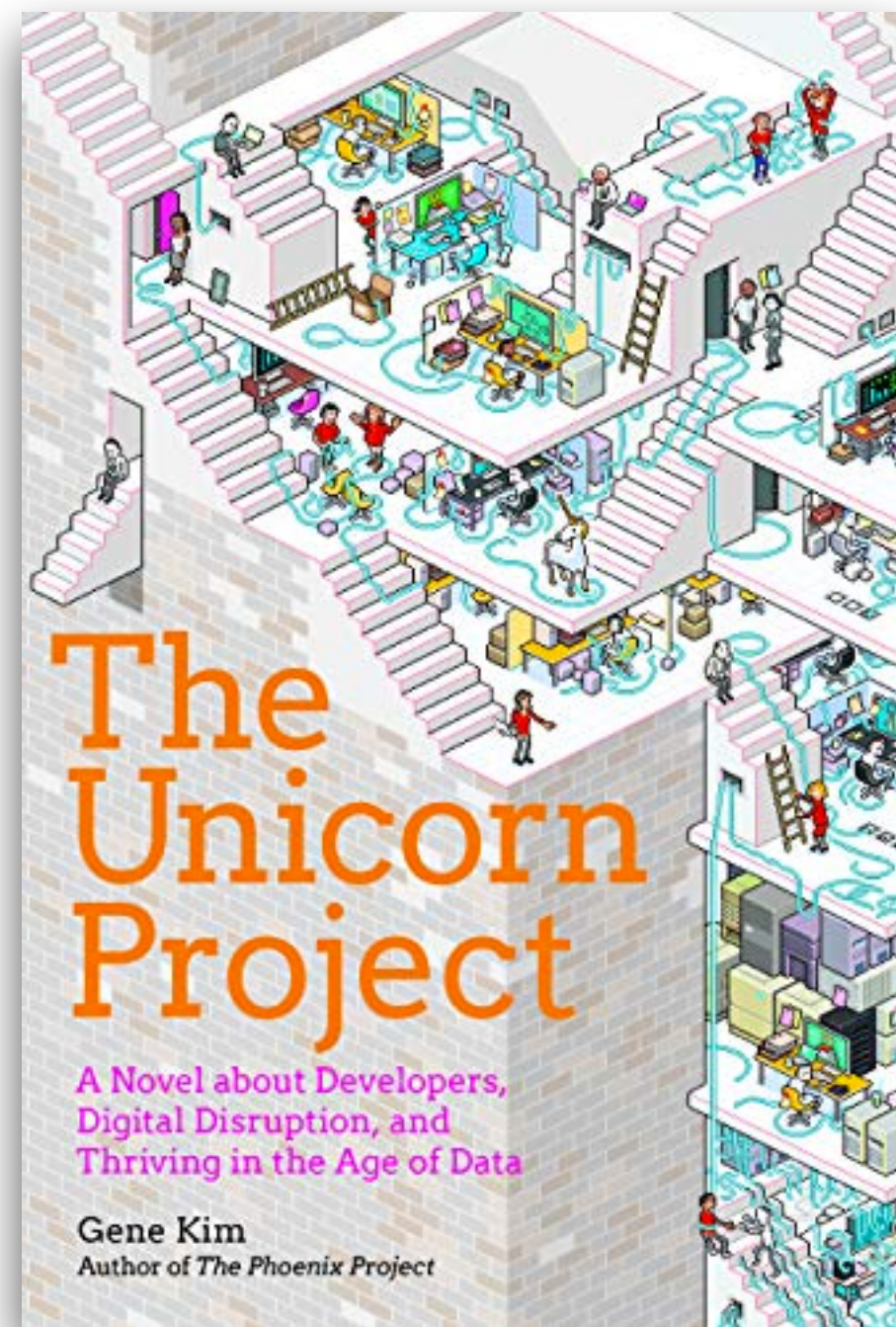


Feature Teams

Work on different features => everything becomes a coordination bottleneck



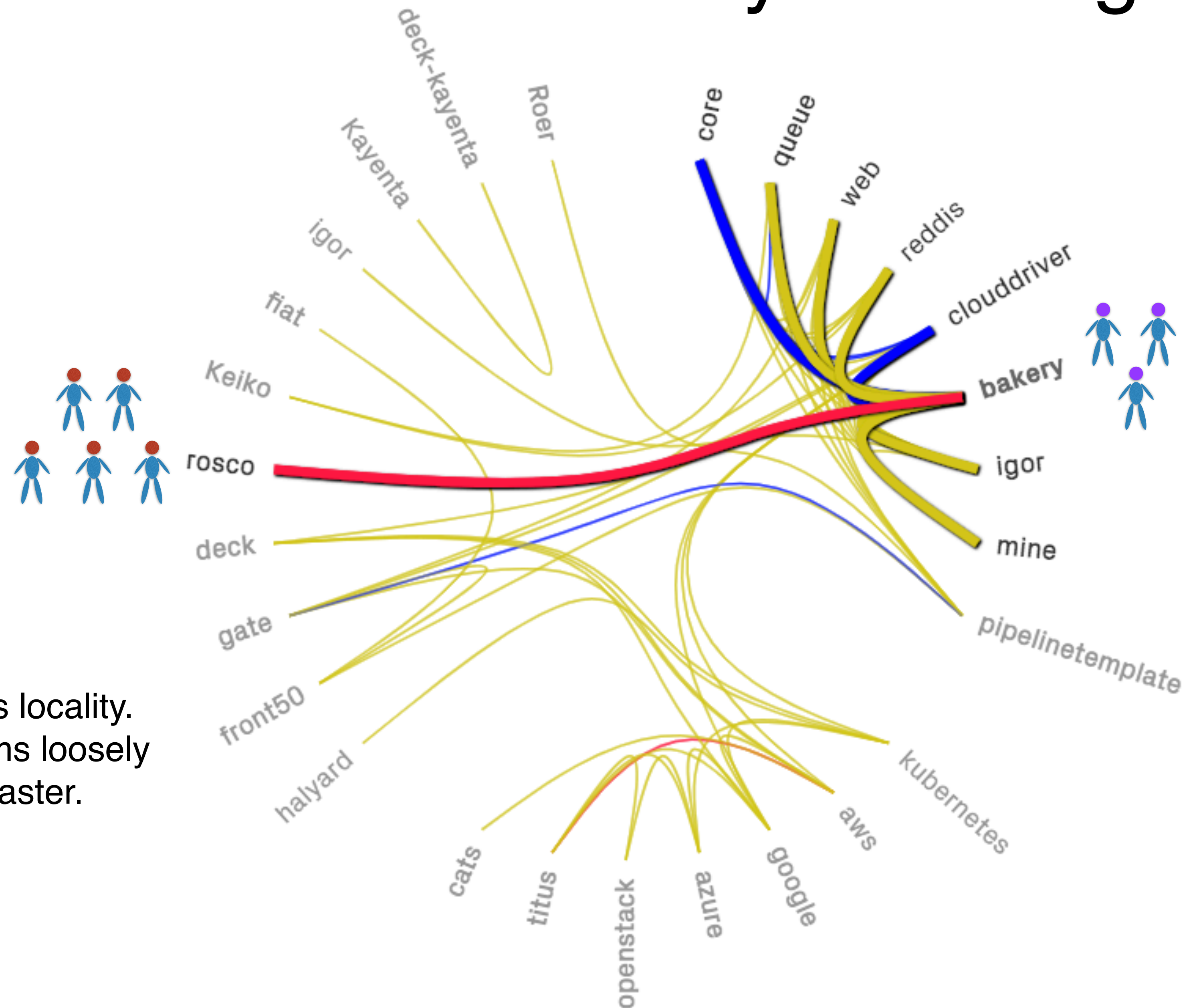
Dependencies and Teams: Locality of Change



“

Simplicity is important because it enables locality. Locality in our code is what keeps systems loosely coupled, enabling us to deliver features faster.

”



Architecture

Investigating Non-Locality of Change

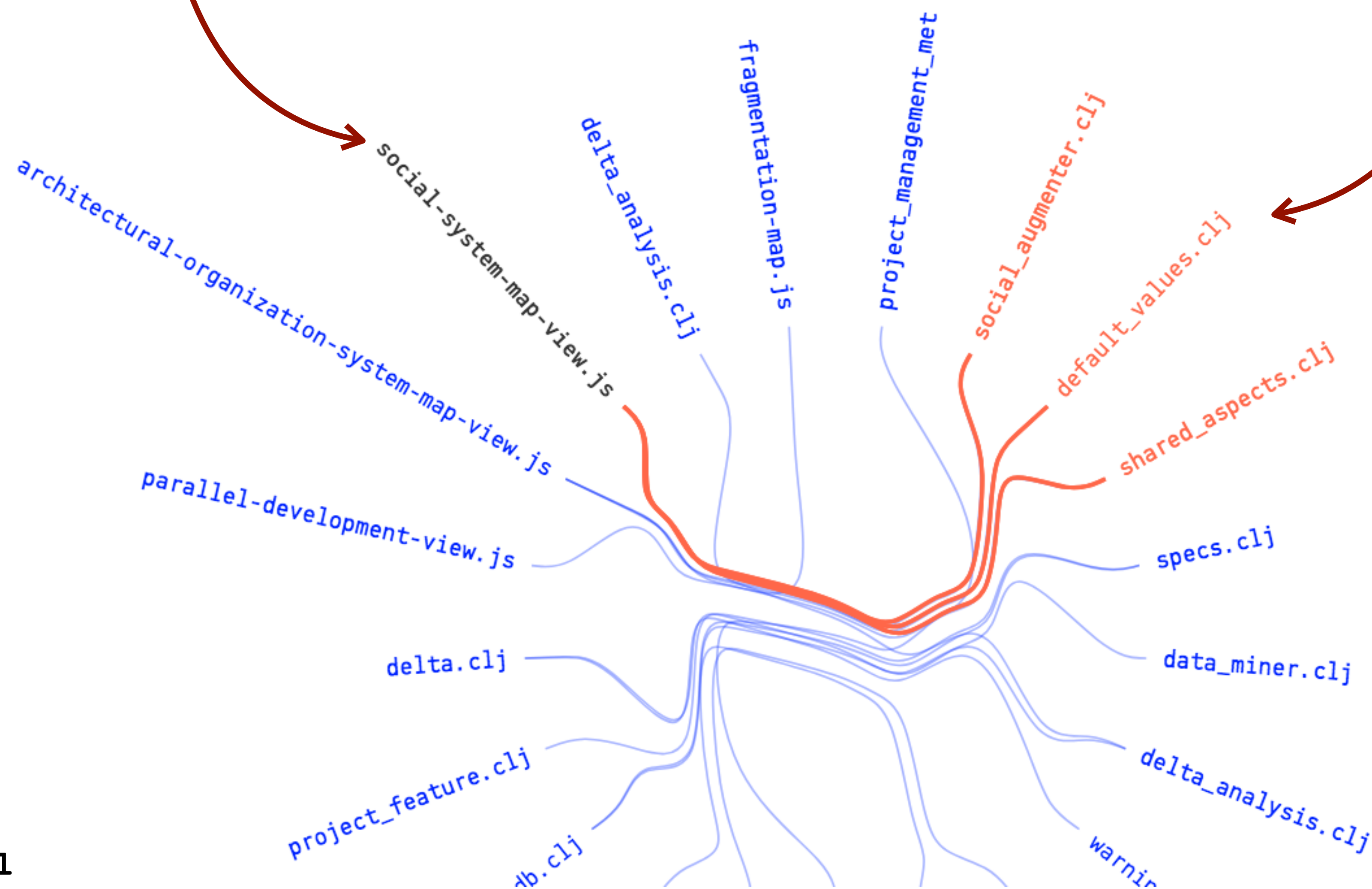
CodeScene™

Powered by Empear

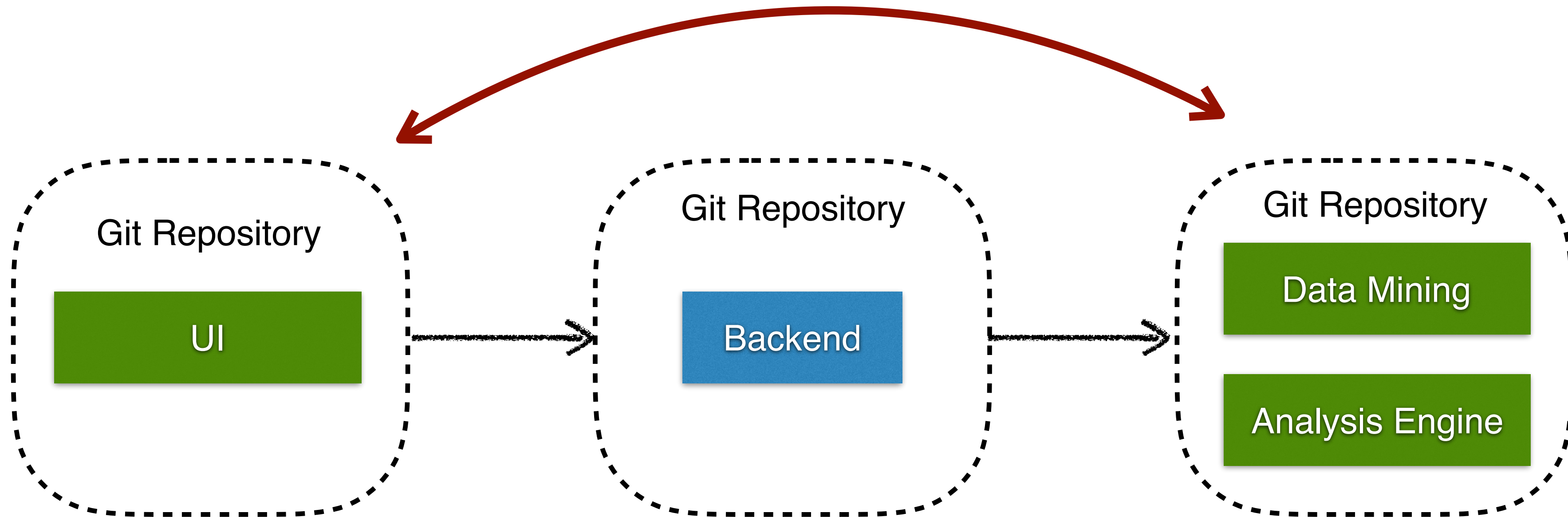
Dig Deeper: Change Coupling between Files in Separate Git Repos

UI code implemented in JavaScript...

...changed coupled to backend code implemented in Clojure (different repository)

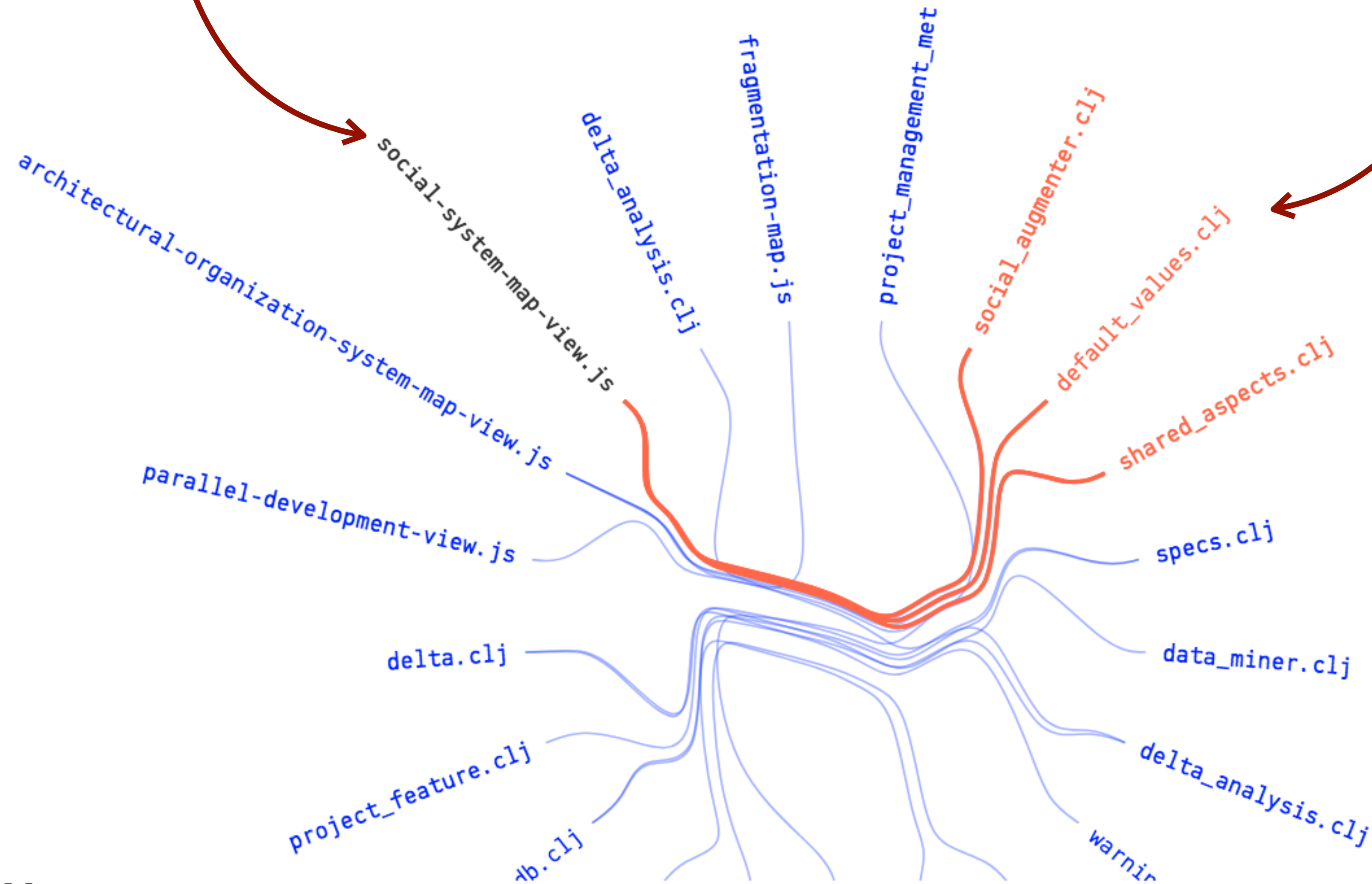


Change Coupling!



UI code implemented in JavaScript...

...changed coupled to backend code implemented in Clojure (different repository)



X-Ray For Change Coupling

JavaScript function for the visualization...

social-system-map--view.js
fileColorByAspect

...that co-evolves with business logic.

social_augmenter.clj
add-analysis-results

test.clj
suits-to

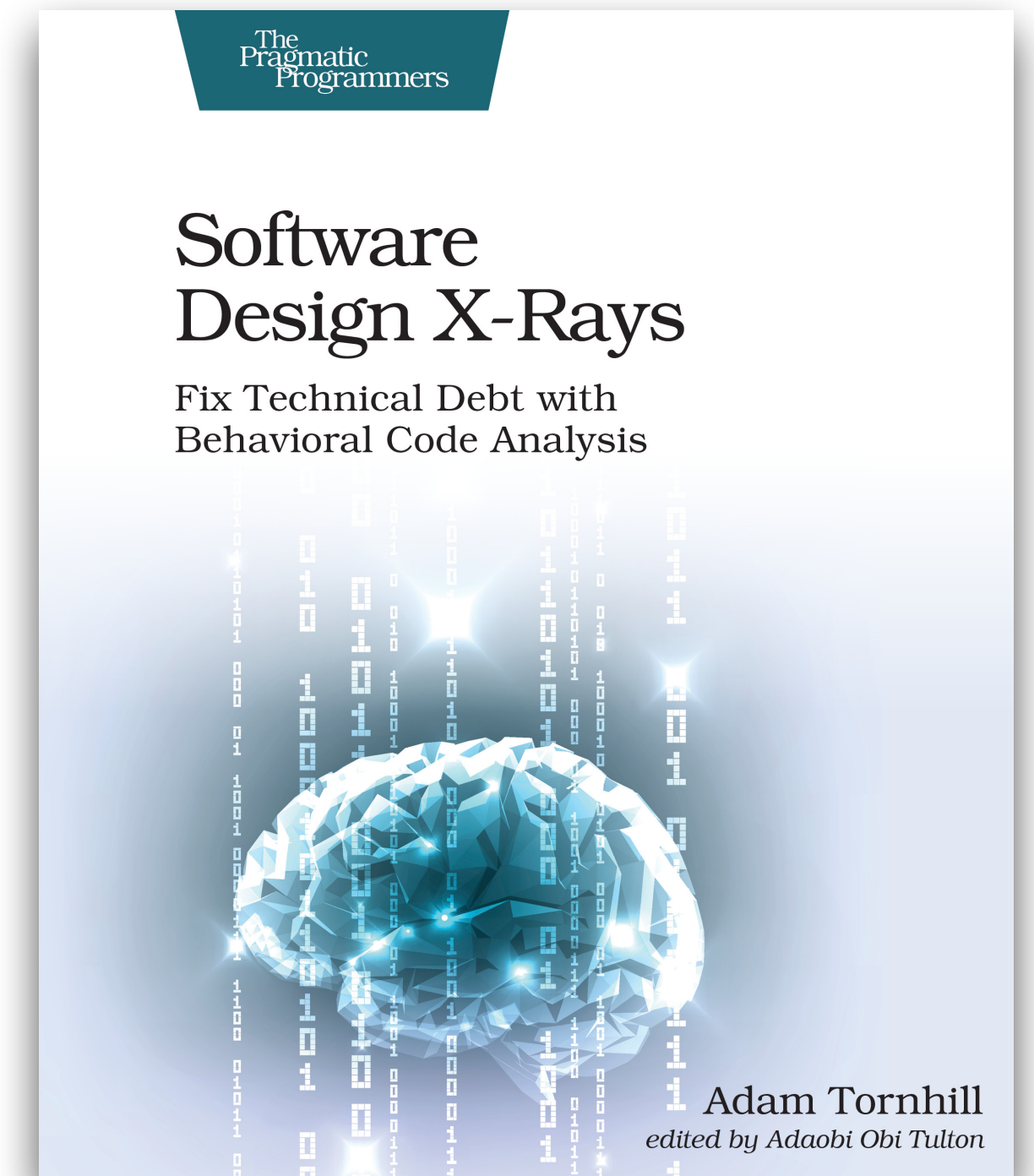
paths.clj
t-matching

system-map-file-level-view.js
combineWithBiomarkersWhenE ...

Re-Think Software Architectures: From Accidental to Essential

Let Features Drive Architectural Building Blocks, not Technology.

<https://codescene.io/>



<https://www.empear.com/blog/>

@AdamTornhill

adam.tornhill@empear.com

