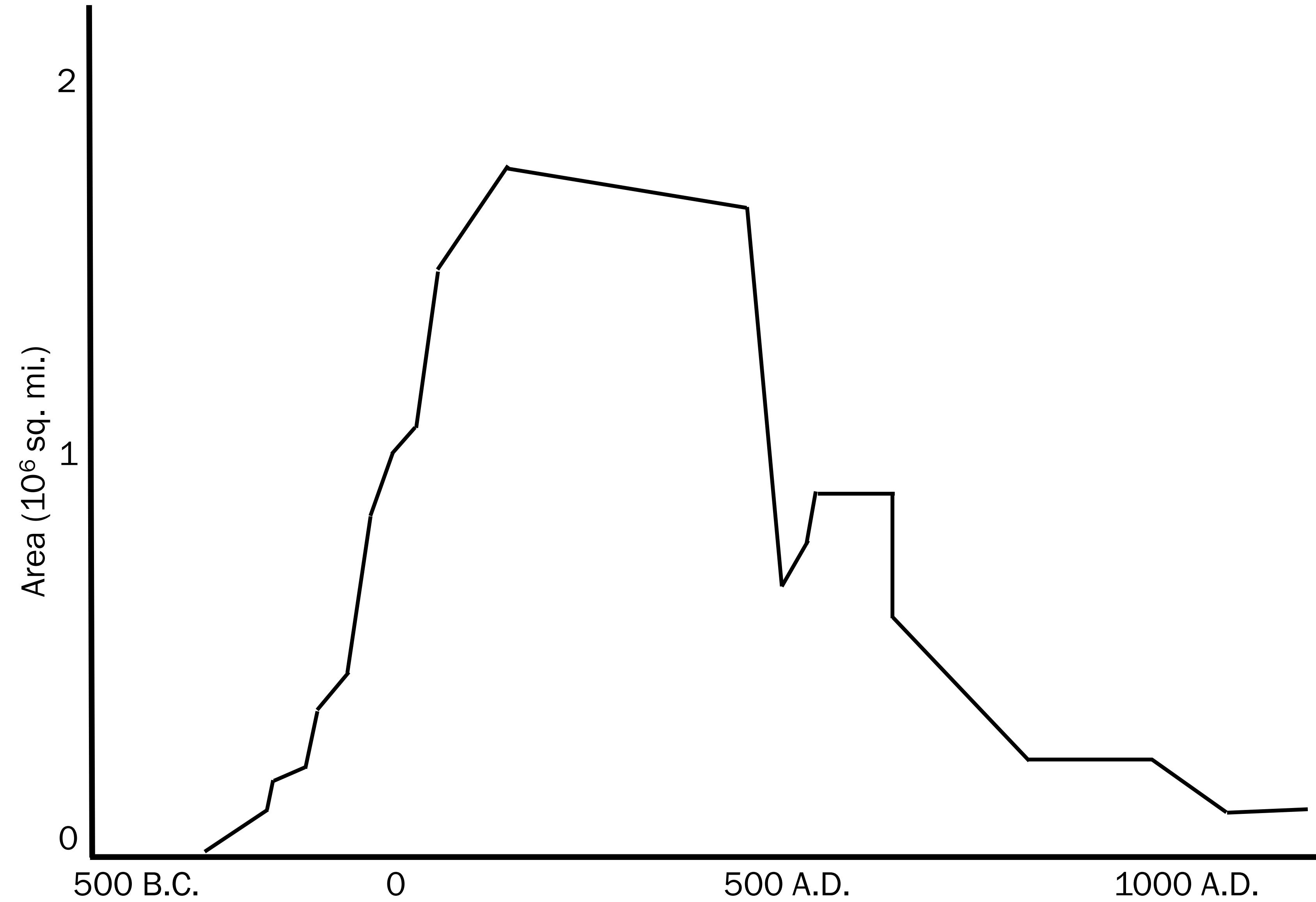


ESCAPE THE TAR PIT WITH A SERVICE MESH

Andrew Jenkins
@notthatjenkins

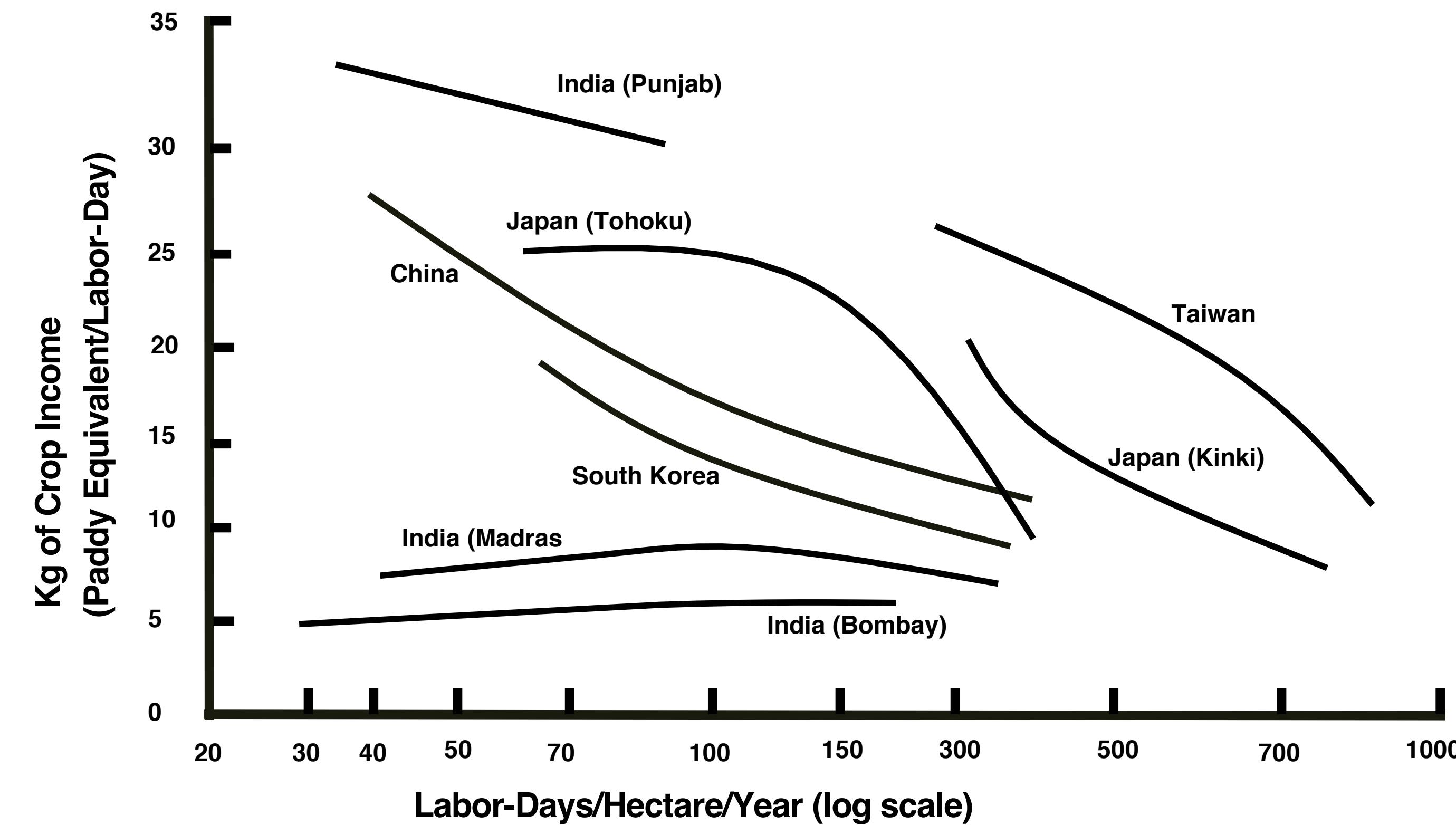
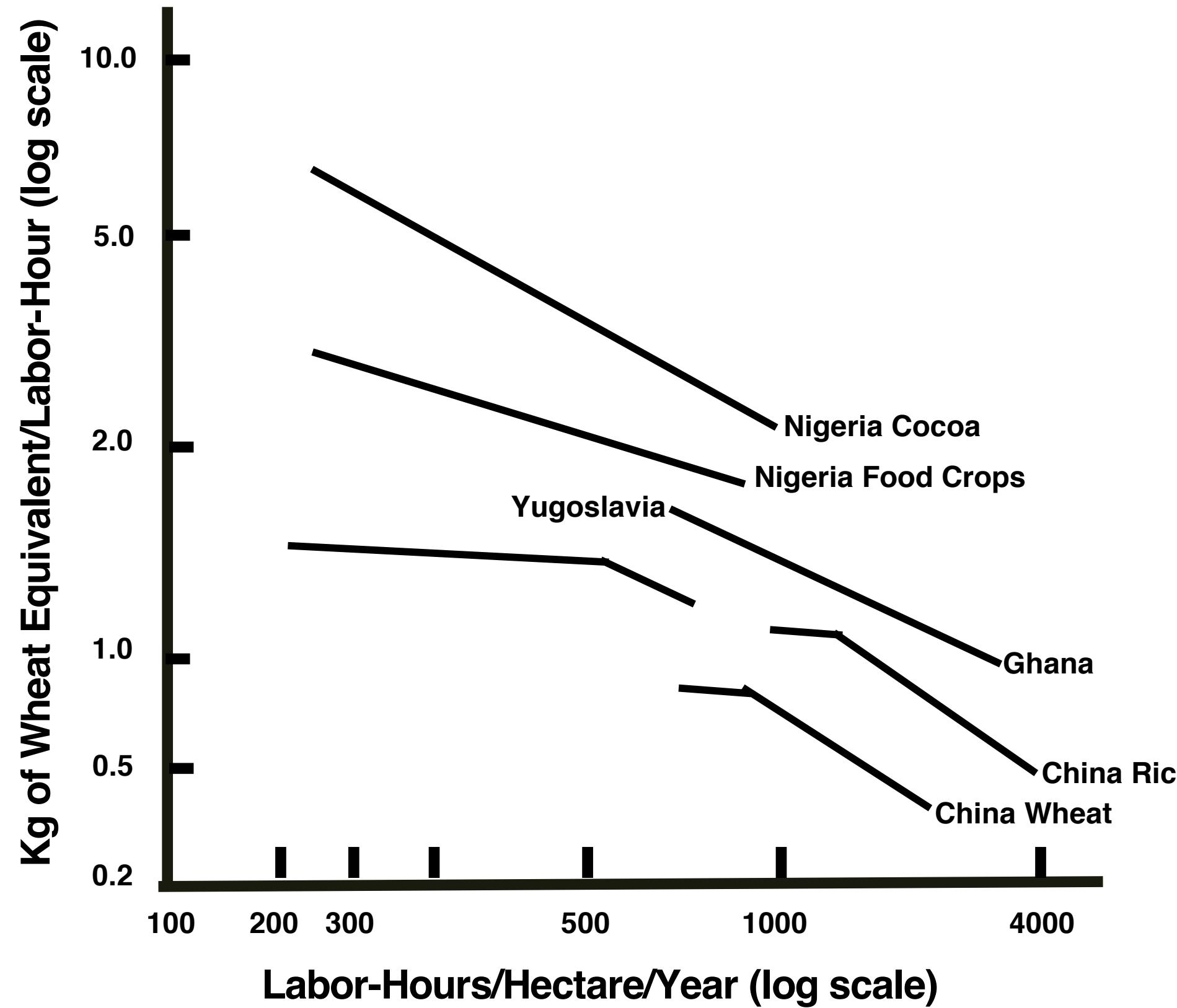


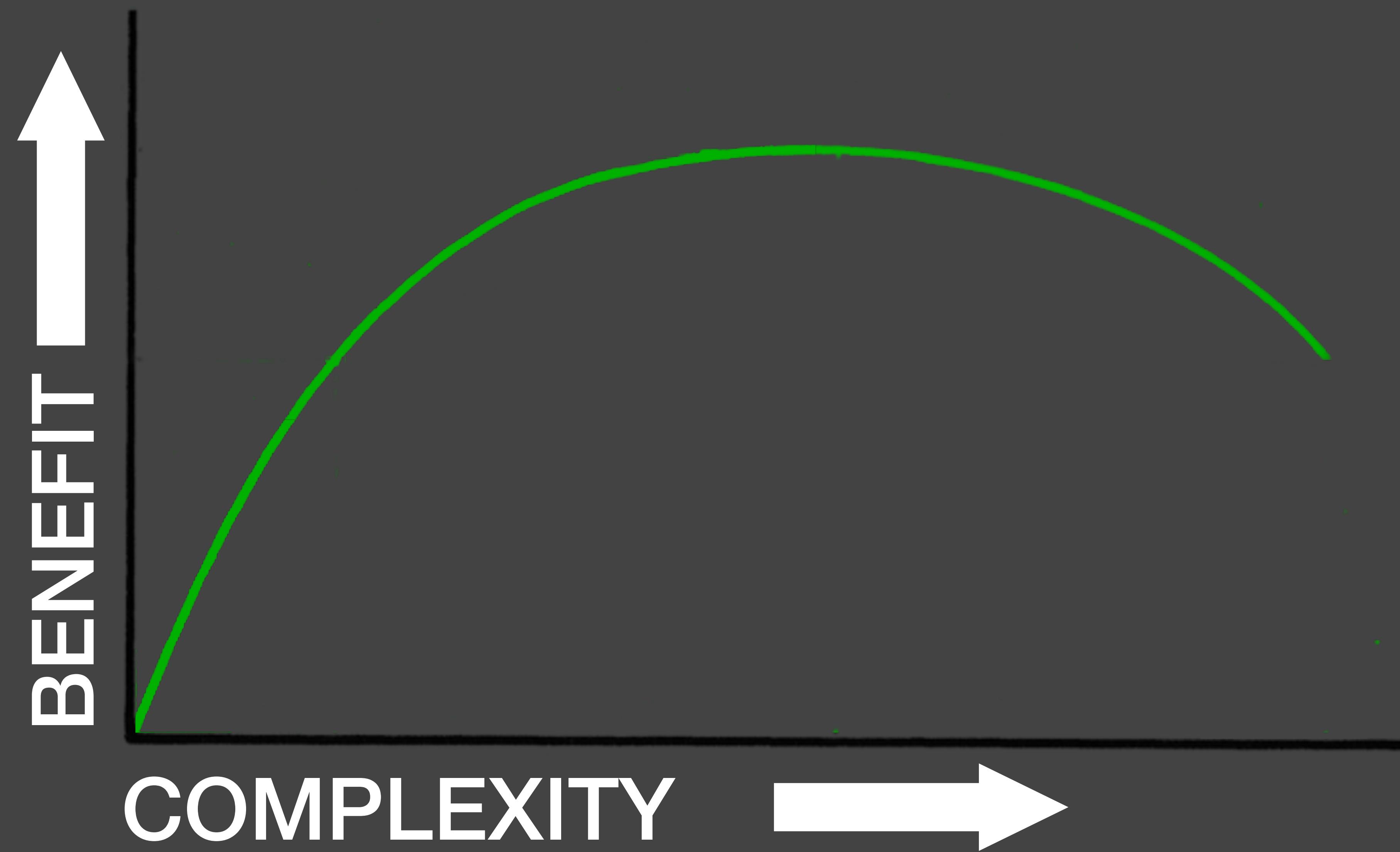
ASPEN MESH





Marginal Cost of Complexity





Technological
Breakthrough

Sailing, Telegraph

Breakaway

Moore's Law, Cloud

Bad for Emperor

Good for Microservices?

Testing

Informal Reasoning

Abstraction that fits inside our head

Out of the Tar Pit

Ben Moseley
ben@moseley.name

Peter Marks
public@indigomail.net

February 6, 2006

Abstract

Complexity is the single major difficulty in the successful development of large-scale software systems. Following Brooks we distinguish *accidental* from *essential* difficulty, but disagree with his premise that most complexity remaining in contemporary systems is essential. We identify common causes of complexity and discuss general approaches

Make it so

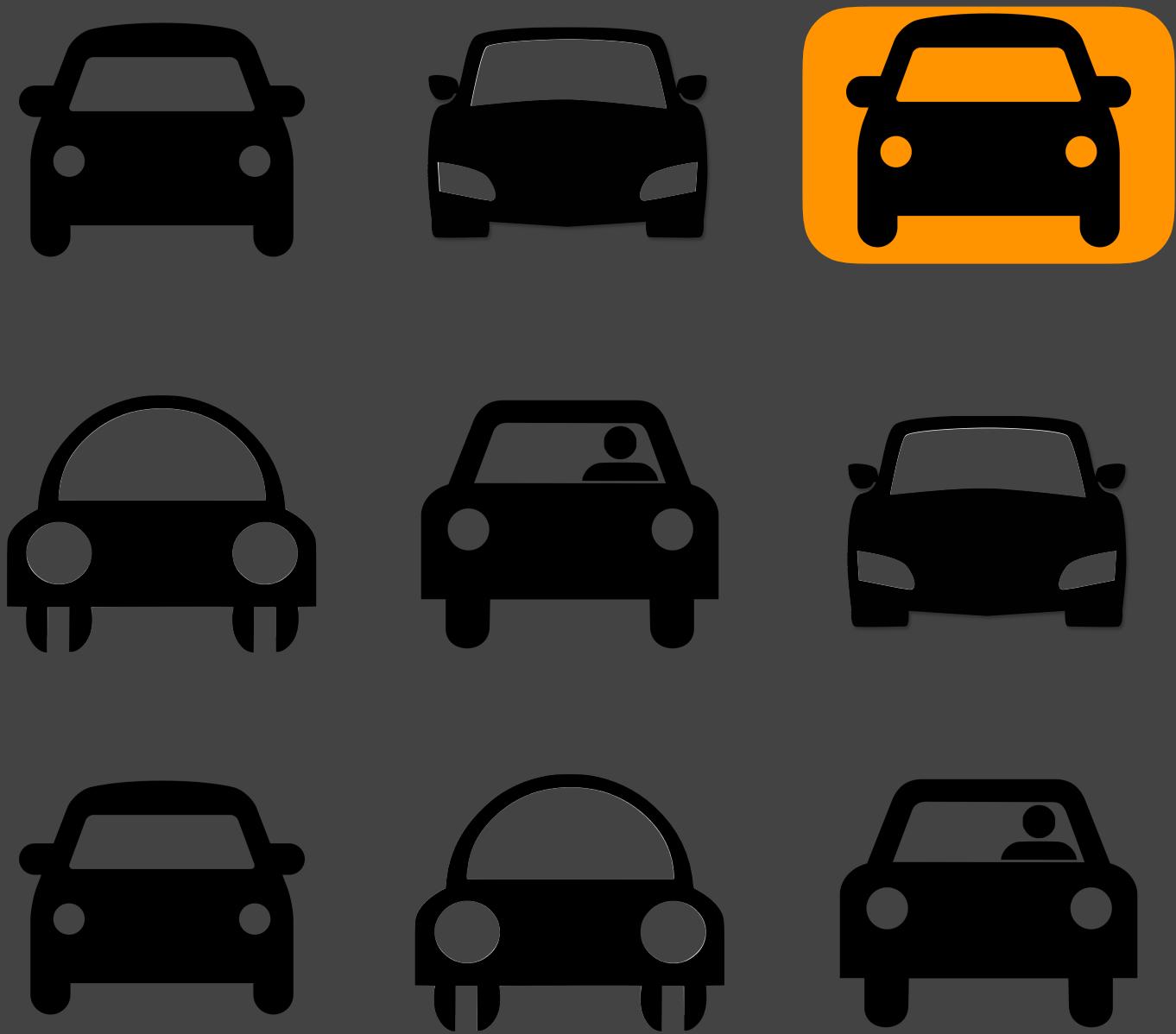
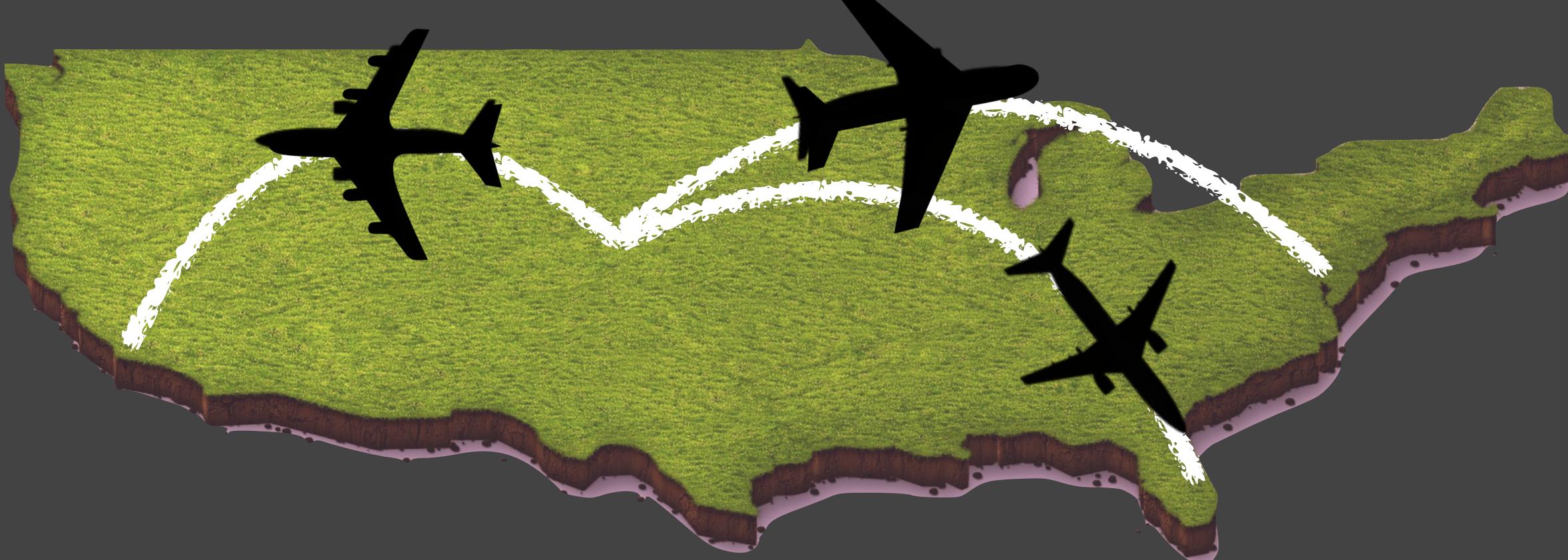
simple ... *obviously* no deficiencies

complex ... no *obvious* deficiencies

C. A. R. Hoare

The first method is far
more difficult

Many DIFFERENT THINGS



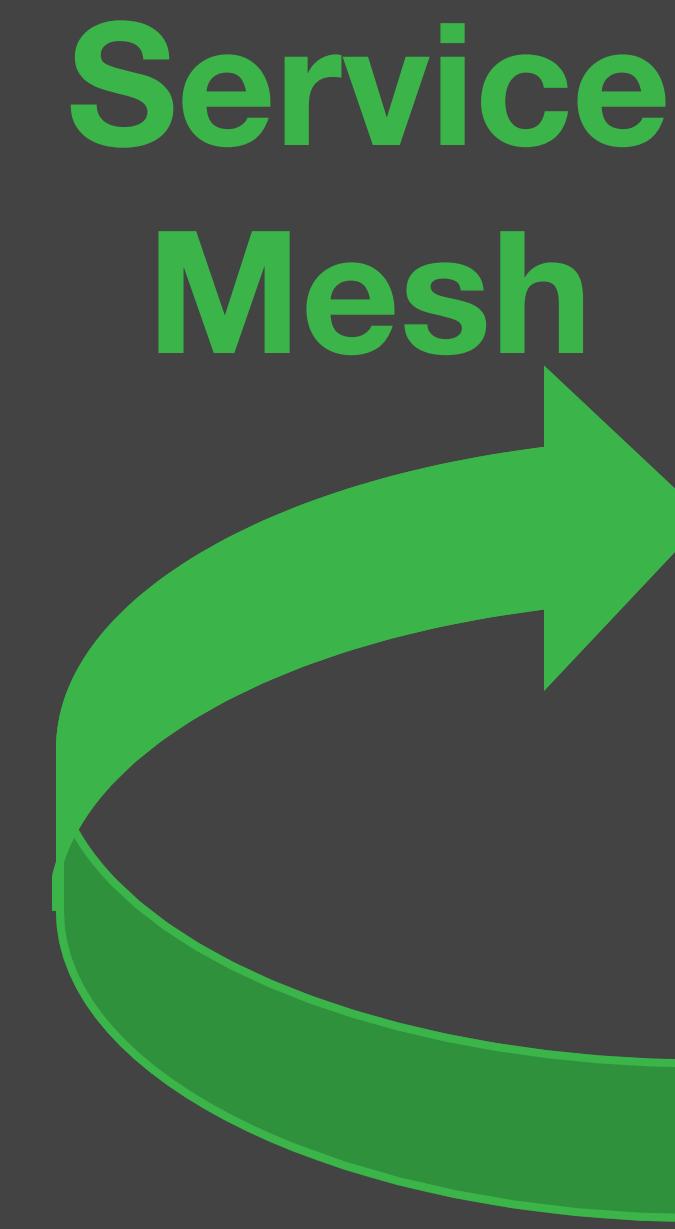
...one
LARGER SYSTEM

Essential Complexity *Essence of Problem*

Accidental Complexity

Stuff you own

Essential Complexity *Essence of Problem*



Shared Accidental
Complexity

*Stuff your project
runs/makes/owns*

Accidental Complexity

Stuff you own

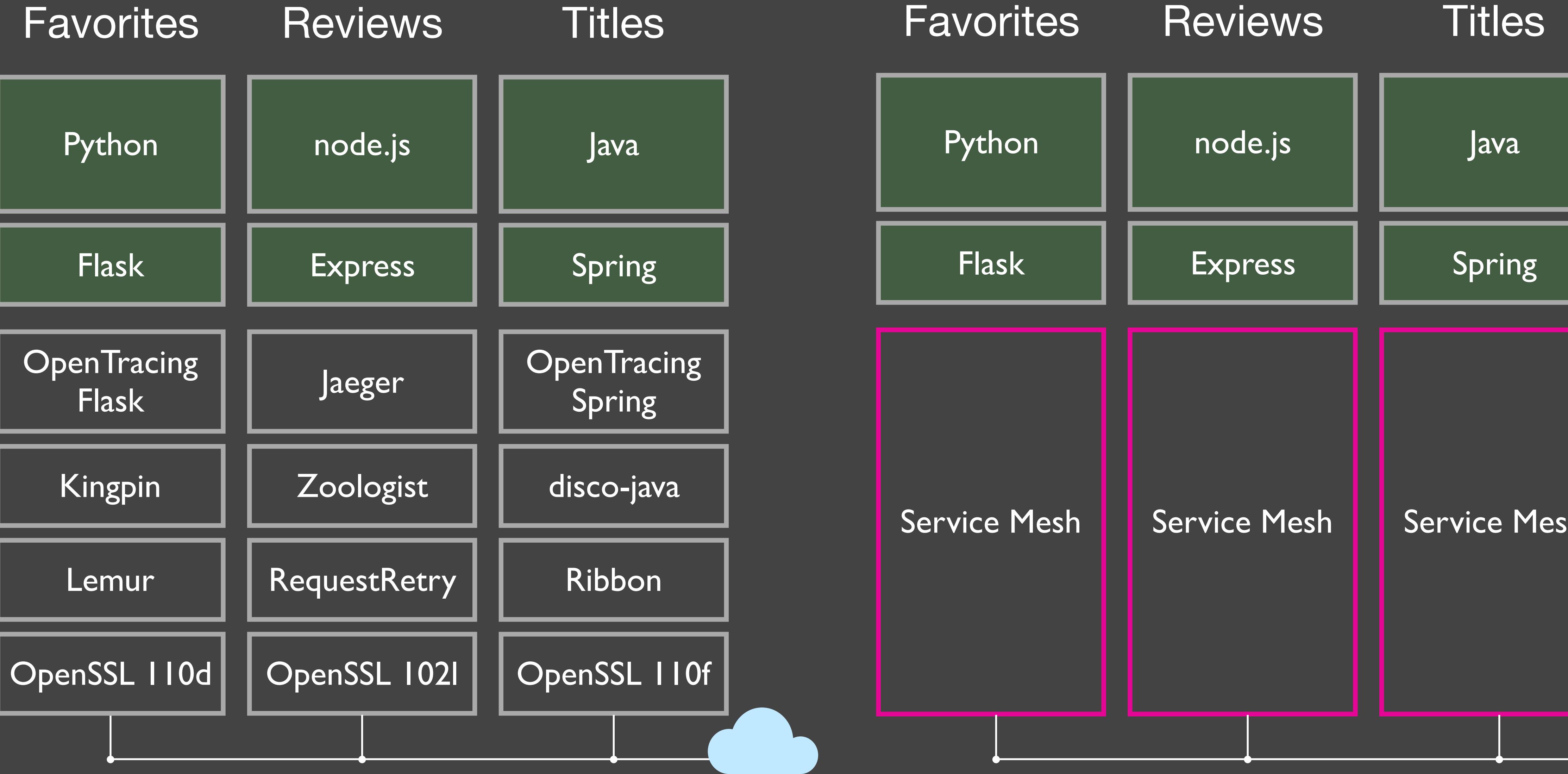
Tracing

Fault Injection

Zero-Trust Security

A/B Routing

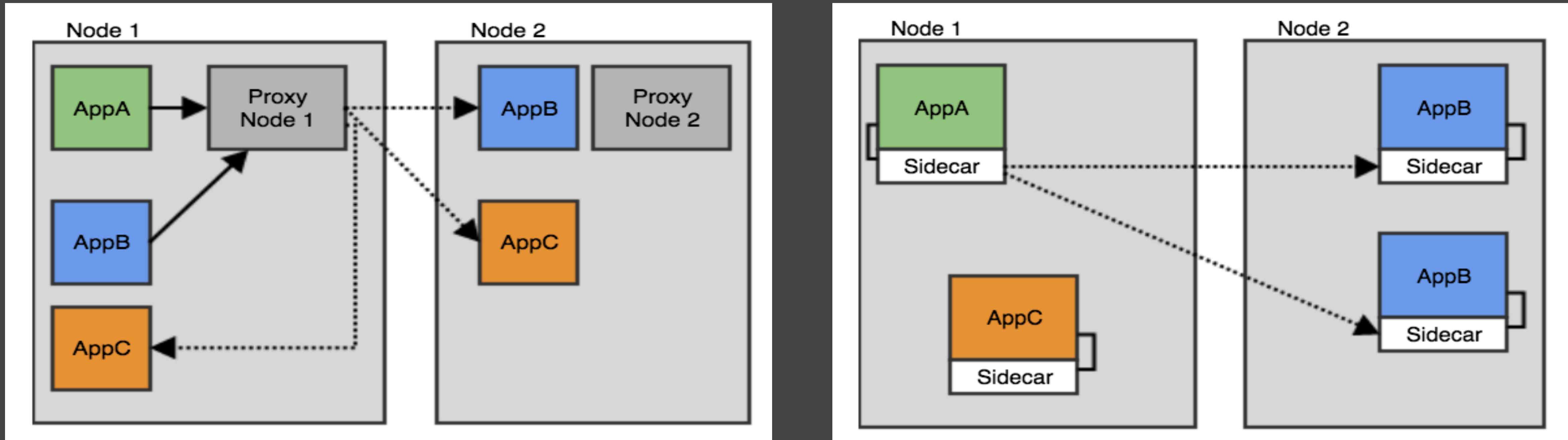
Before



Some complexity leaves

Other complexity into
Shared Accidental Complexity

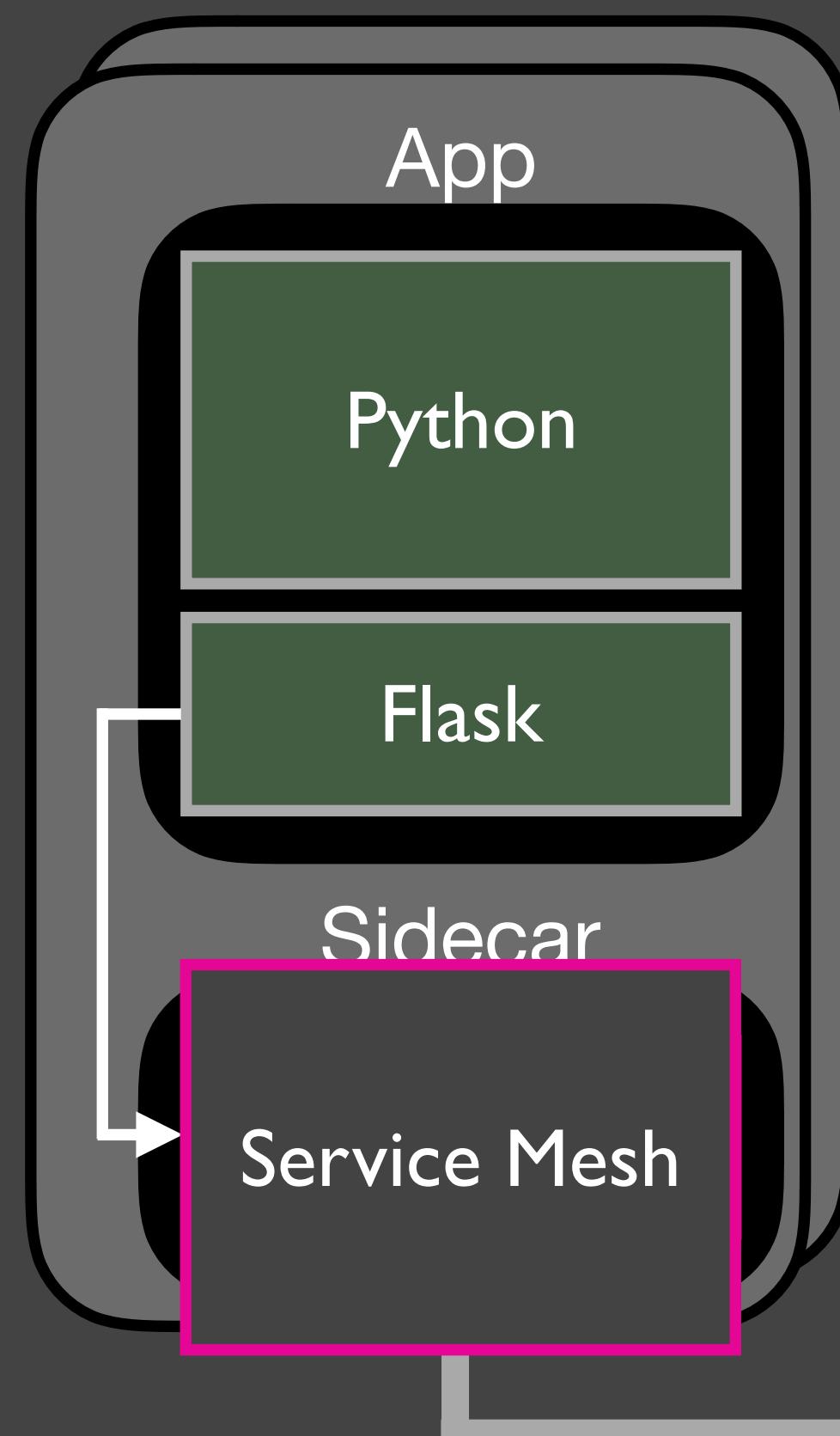
Service Mesh Architecture



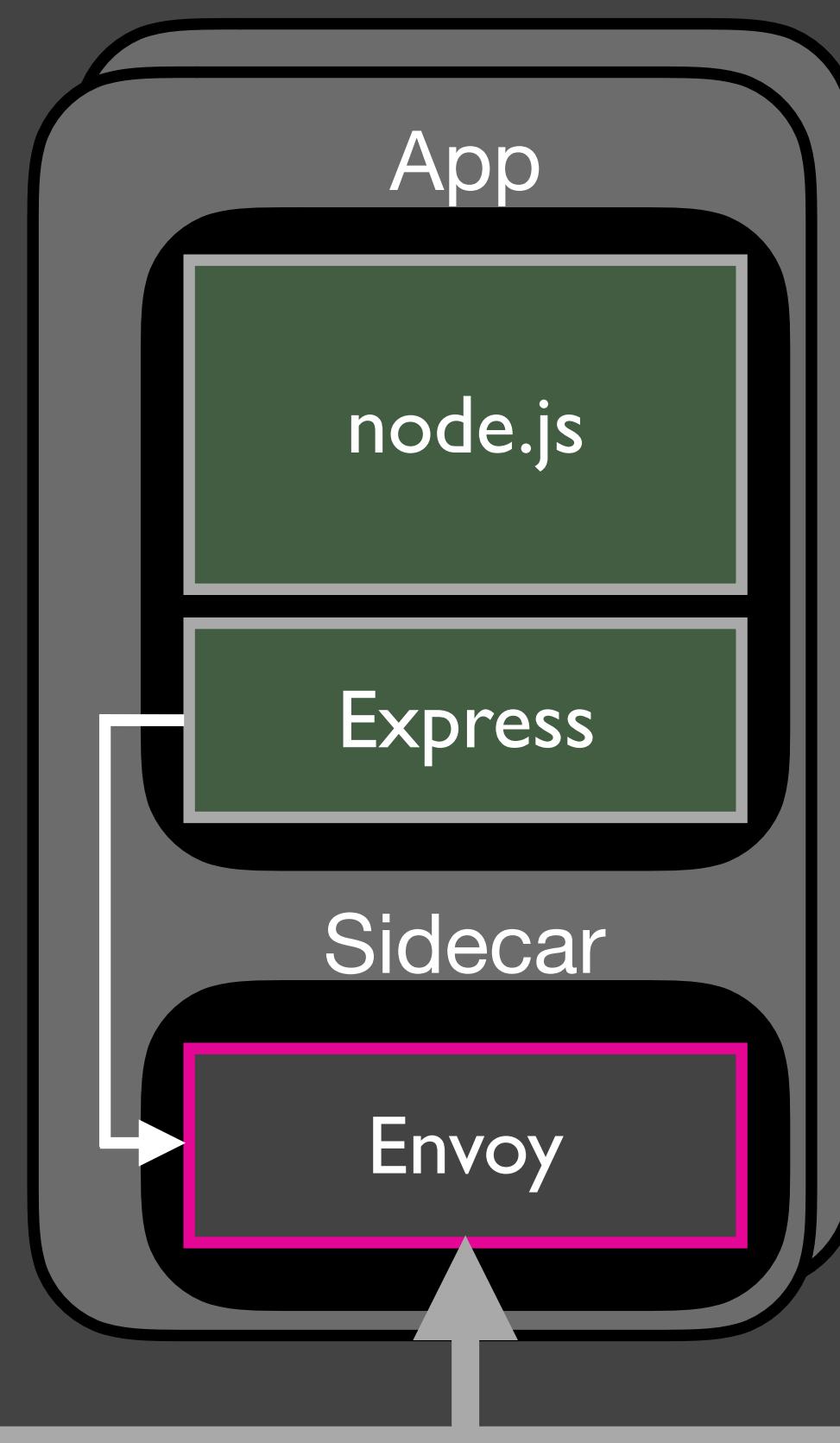
DaemonSet or
Node Agent

Sidecar

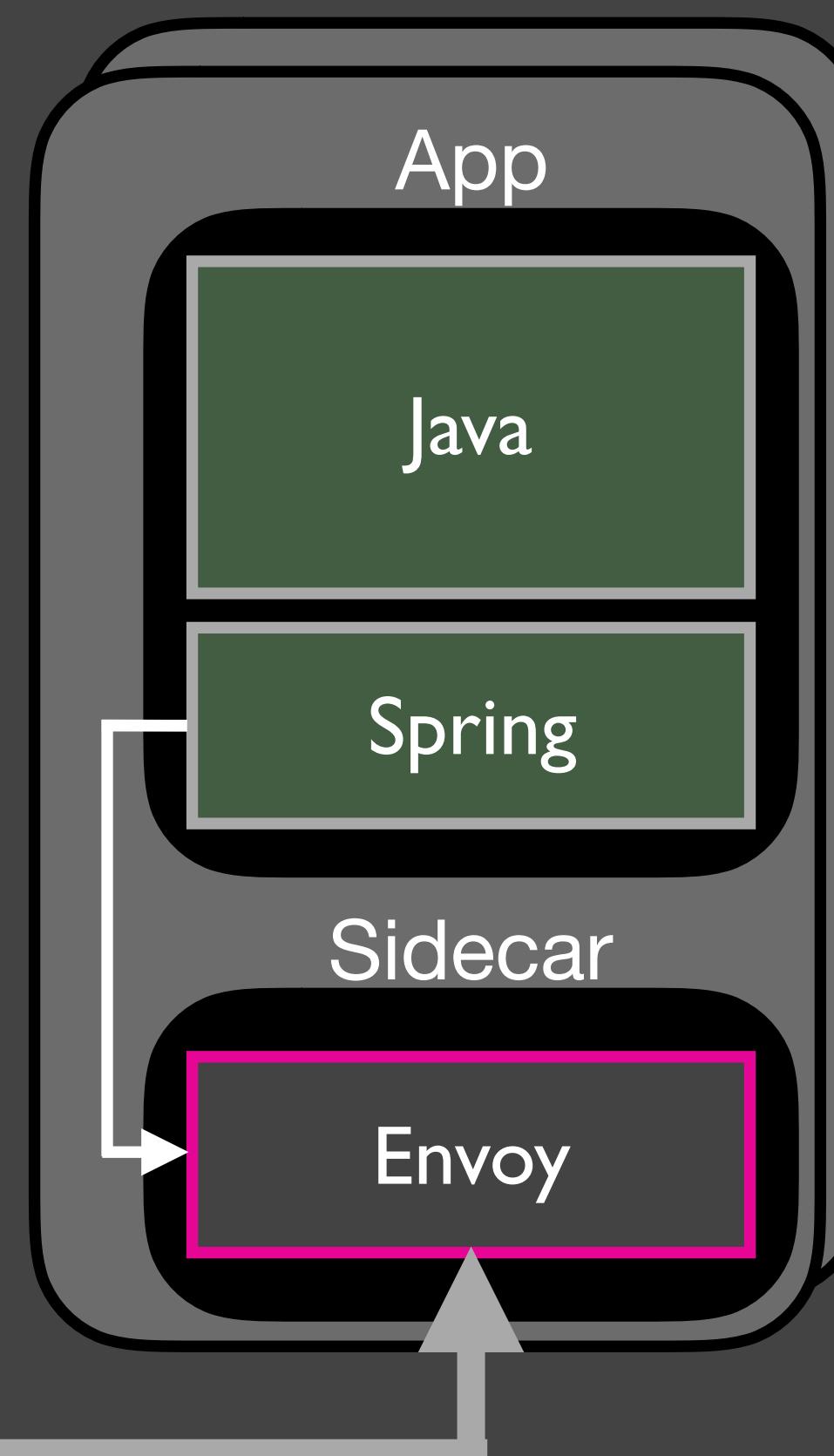
Favorites



Reviews

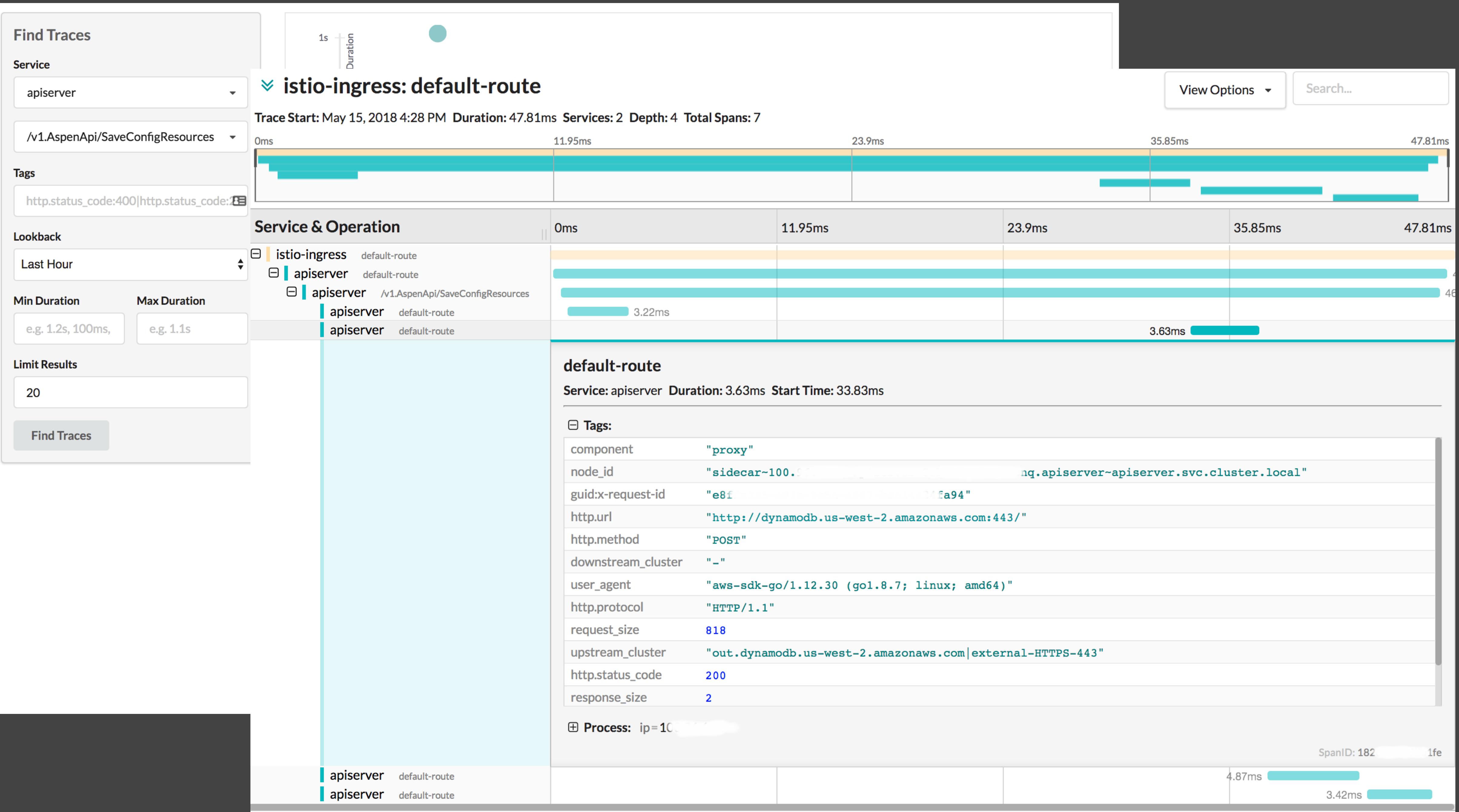


Titles

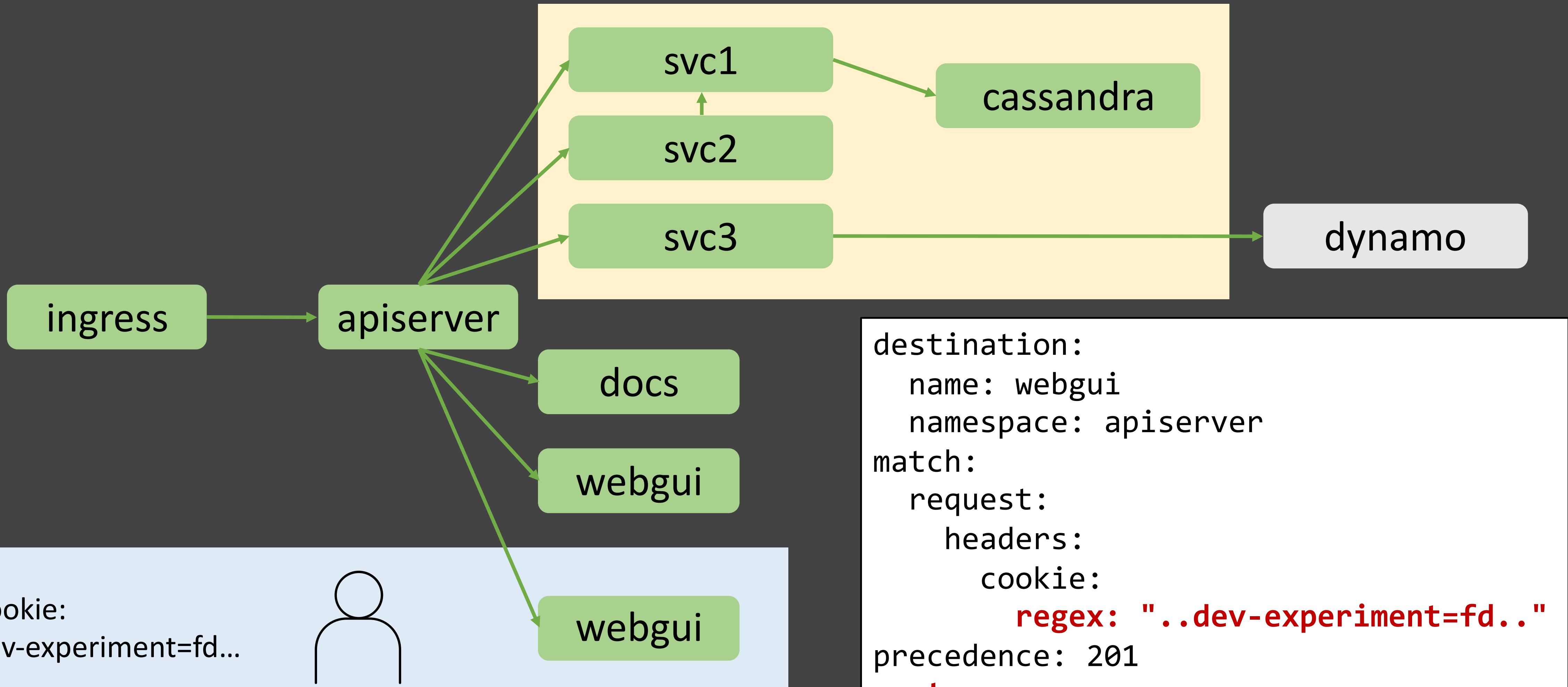


Istio Control Plane

Consistent Tracing and Debugging

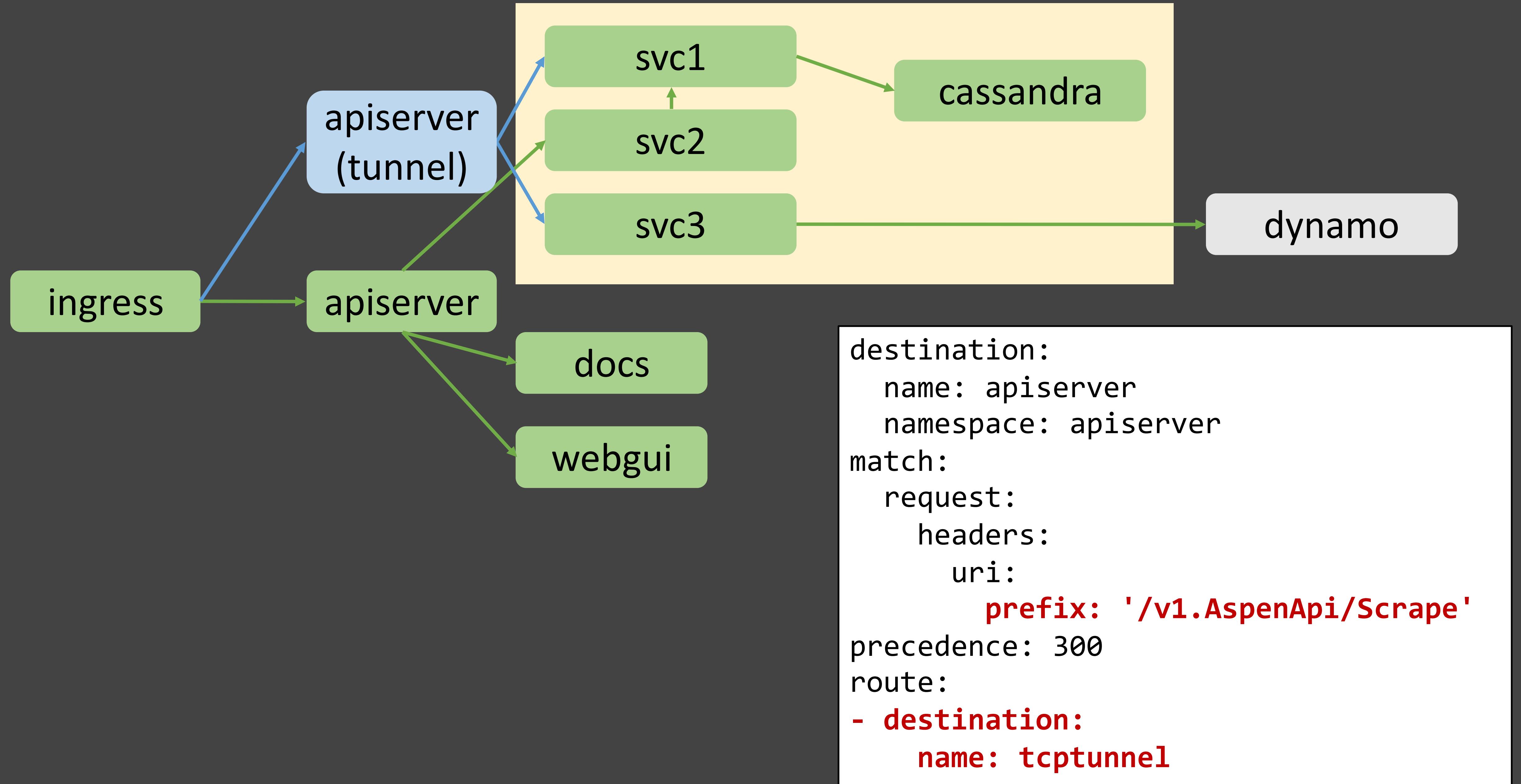


Service Experiments

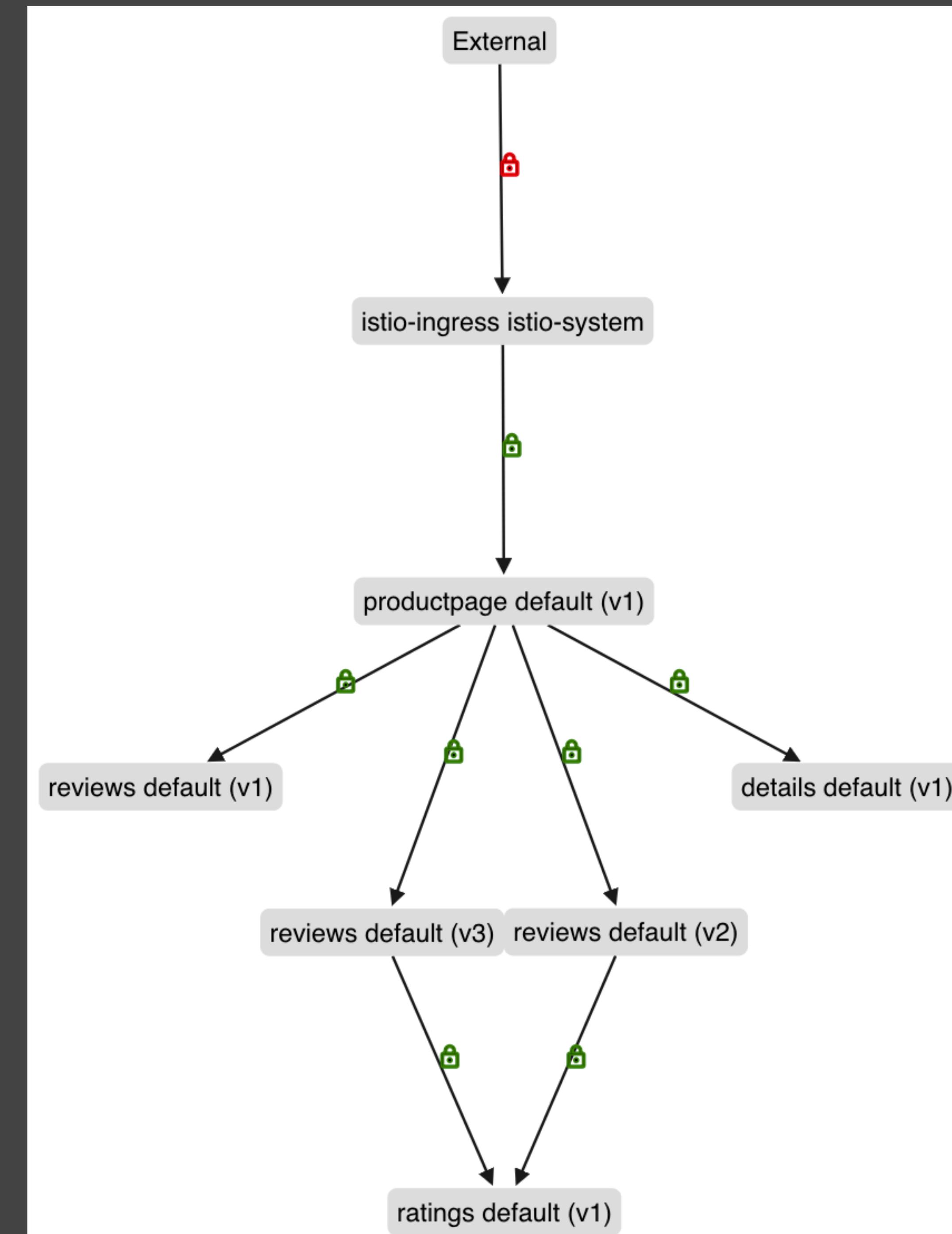


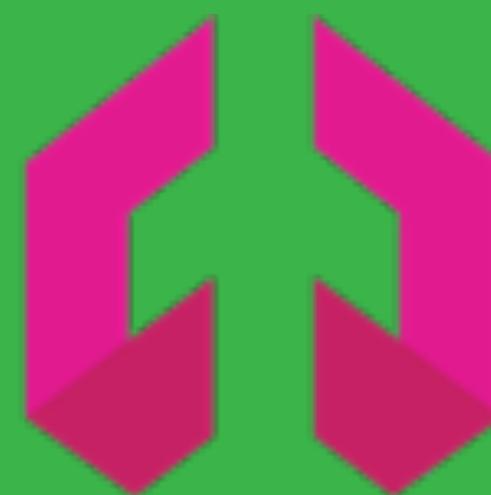
Cookie:
dev-experiment=fd...

Service Decomposition



Service Graph





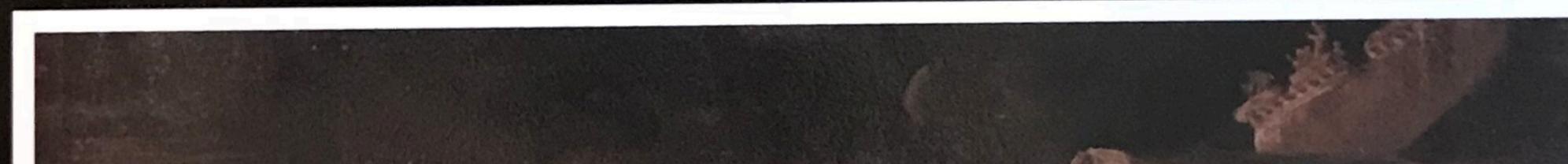
← (We're Hiring)

ASPEN MESH

New Studies in Archaeology

The Collapse of Complex Societies

JOSEPH A. TAINTER



Out of the Tar Pit

Ben Moseley
ben@moseley.name

Peter Marks
public@indigomail.net

February 6, 2006

Abstract

Complexity is the single major difficulty in the successful development of large-scale software systems. Following Brooks we distinguish *accidental* from *essential* difficulty, but disagree with his premise that most complexity remaining in contemporary systems is essential. We identify common causes of complexity and discuss general approaches which can be taken to eliminate them where they are accidental in nature. To make things more concrete we then give an outline for a potential complexity-minimizing approach based on *functional programming* and *Codd's relational model of data*.

1 Introduction

The “software crisis” was first identified in 1968 [NB69, p70] and in the

