

Modularity: A Retrospective



dean@deanwampler.com
@deanwampler

 AnyScale

Outline

- A little history
- What is “modularity”? Why is it useful?
- Modularity in Software:
 - What we’ve tried: the good and the bad
 - What we still need to do

A Little History



Interchangeable Parts

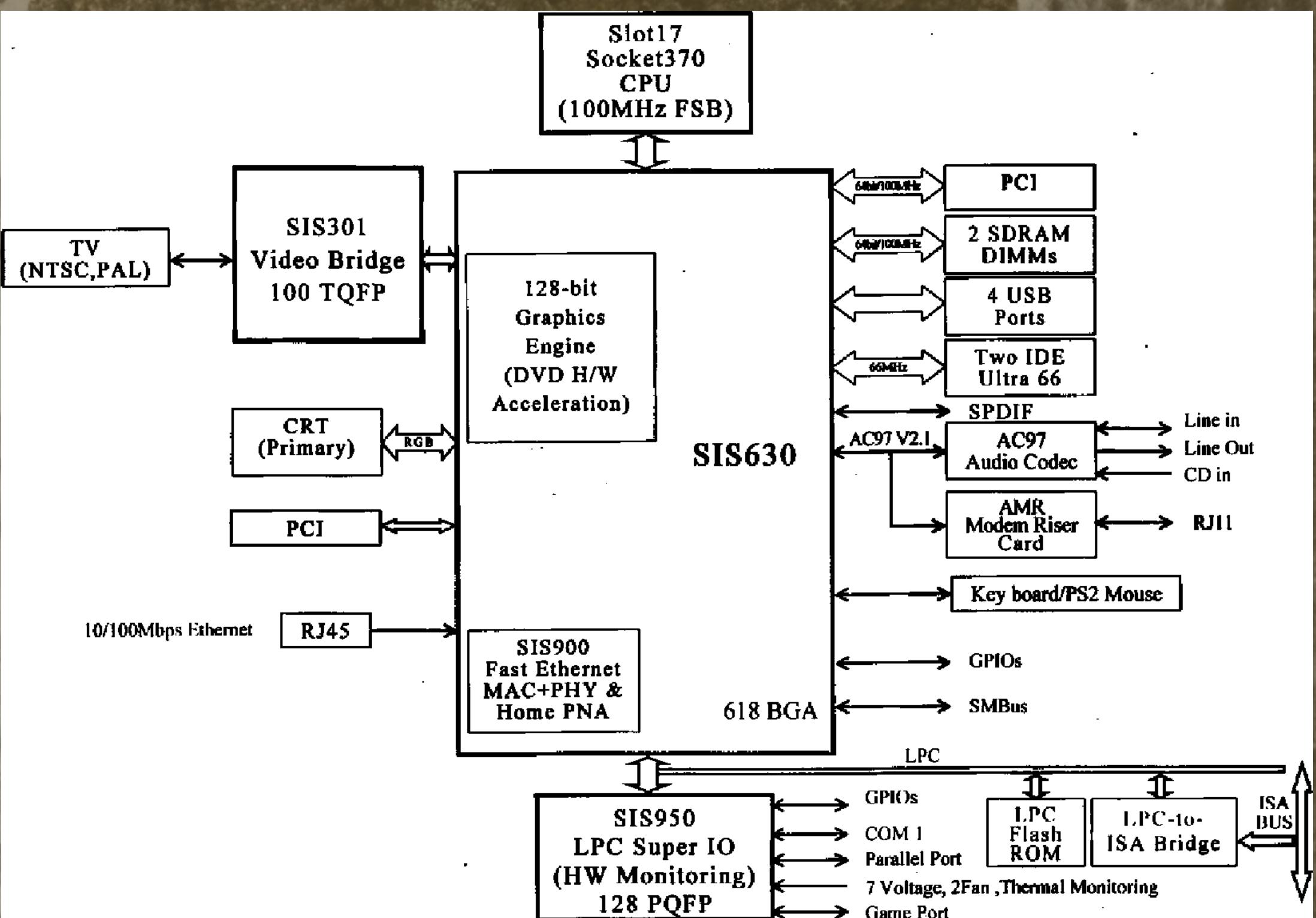
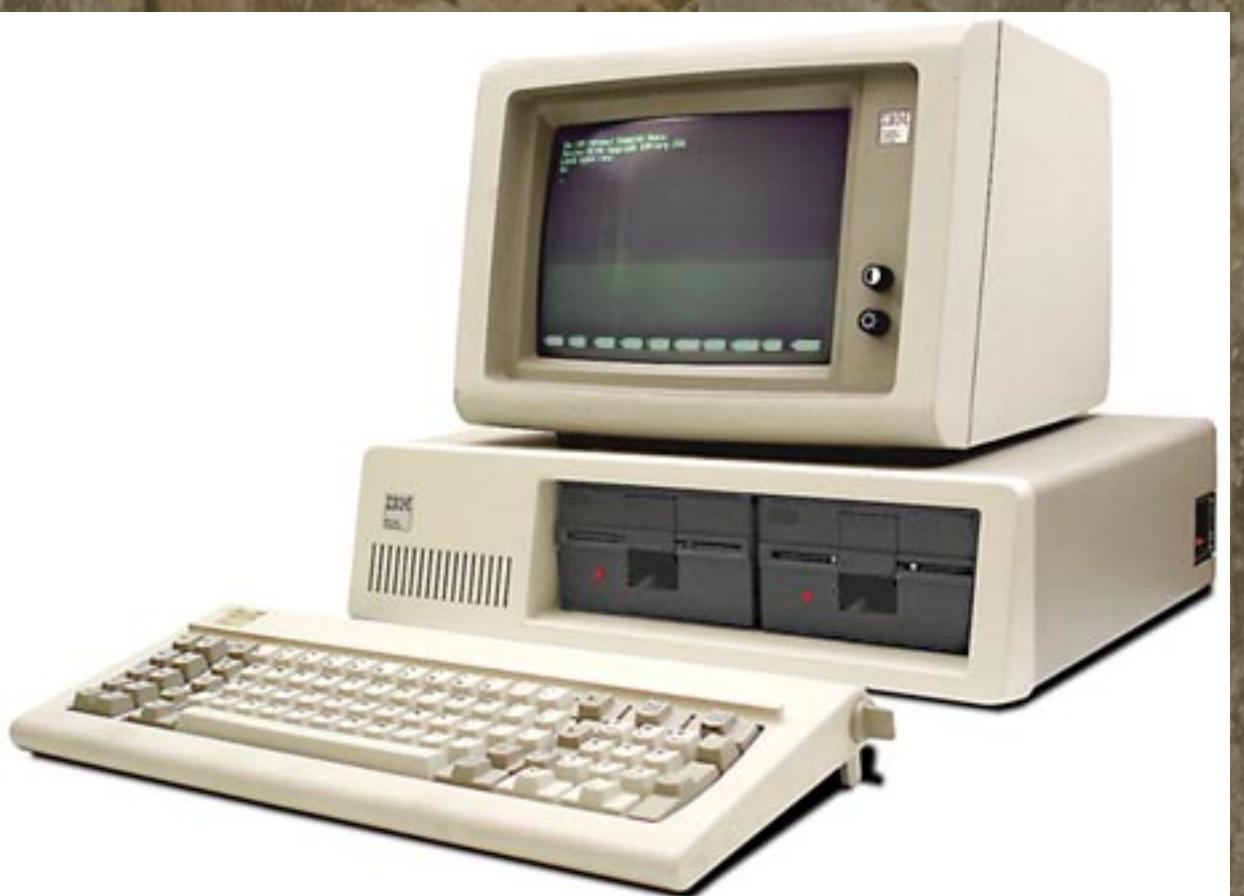
- 1760s: French General Jean-Baptiste Vaquette de Gribeauval
- Standardized cannon bores, shells
- Eli Whitney - early 1800s
- Muskets



<https://en.wikipedia.org/wiki/Artillery>
<https://en.wikipedia.org/wiki/Musket>

Interchangeable Parts

- Electronics
- → PCs

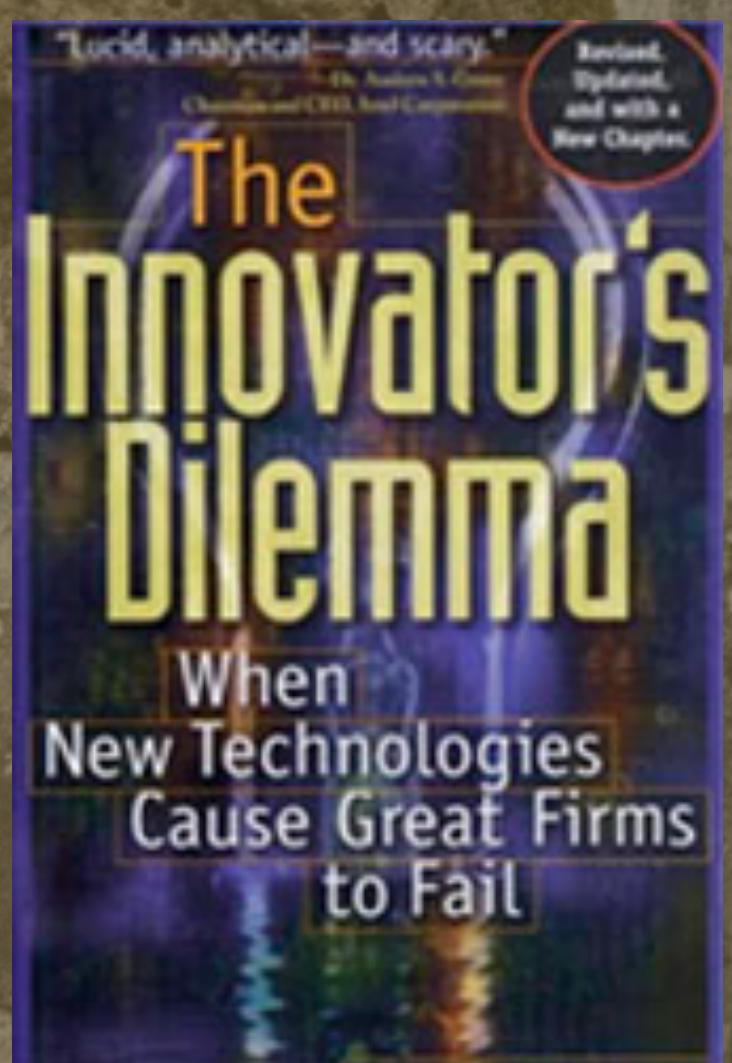
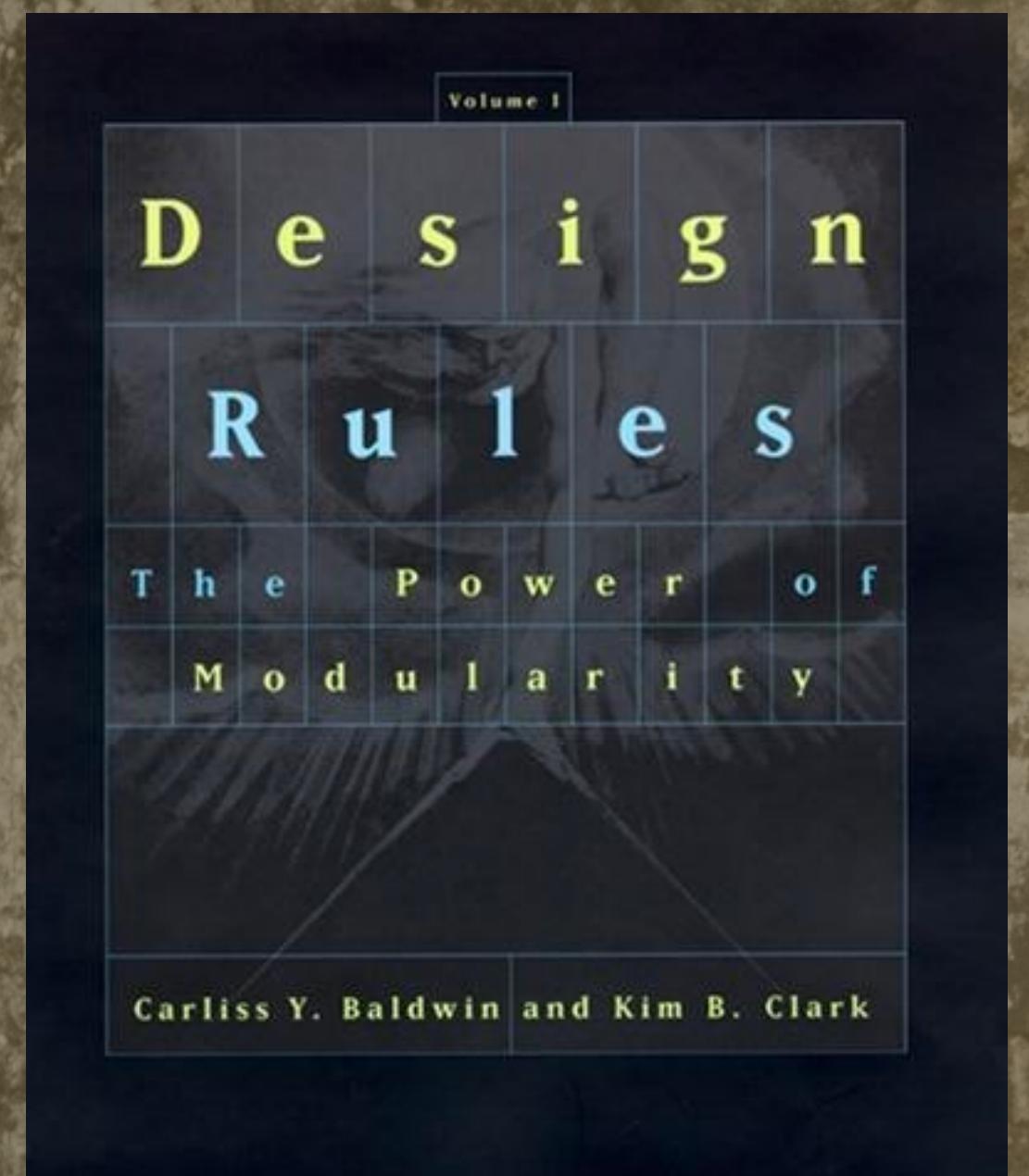


<https://fccid.io/NYYTP100/Block-Diagram/BLOCK-DIAGRAM-133927>

<http://oldcomputers.net/ibm5150.html>

Interchangeable Parts

- Electronics
- → PCs

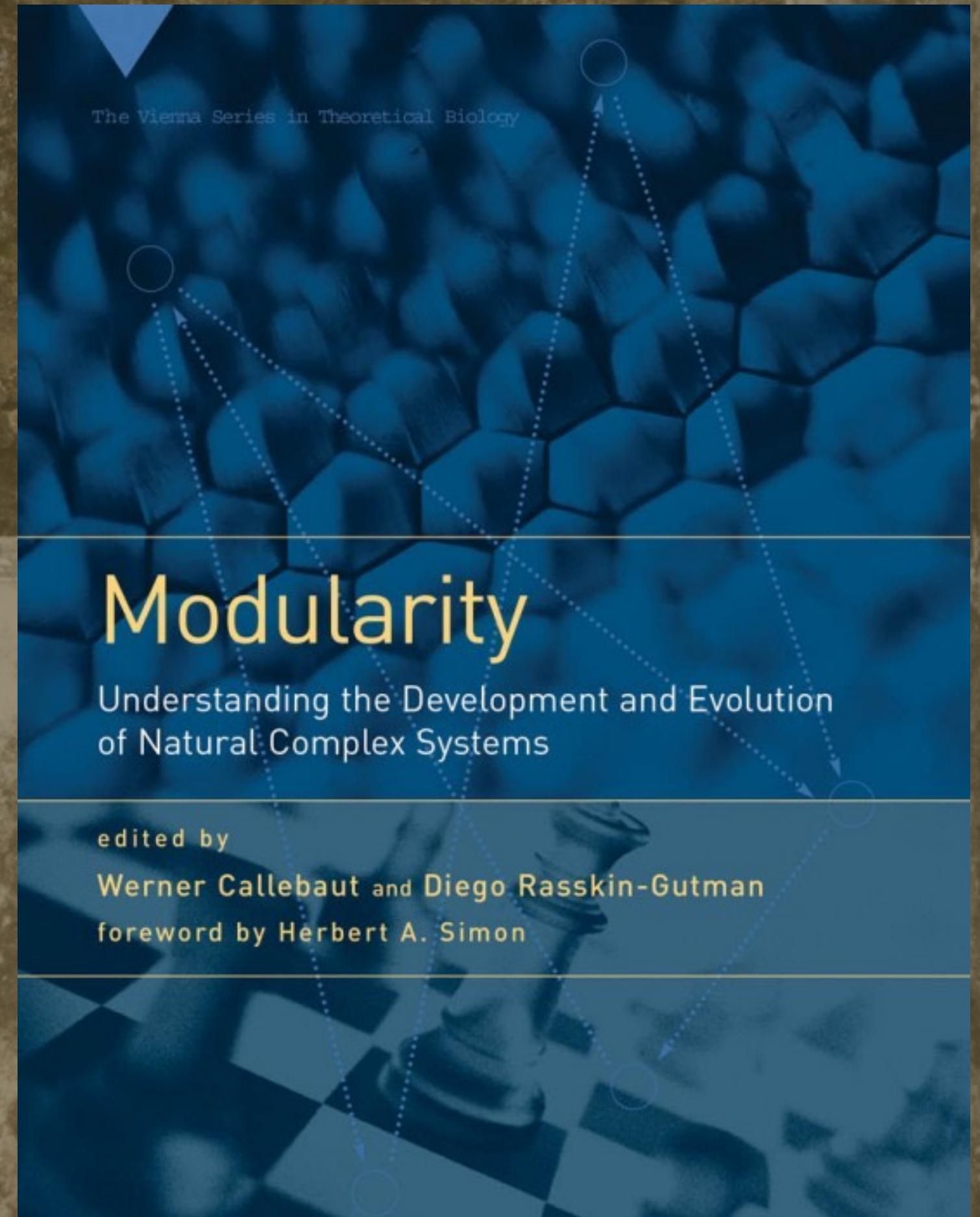


<https://mitpress.mit.edu/books/design-rules-volume-1>

<http://claytonchristensen.com/books/the-innovators-dilemma/>

Biology??

- RNA evolves energy-stable structures that resist mutation
- But we need mutation...



<https://mitpress.mit.edu/books/modularity>

Chapter 6, Evolutionary Lock-in and Origin of Modularity in RNA Structure

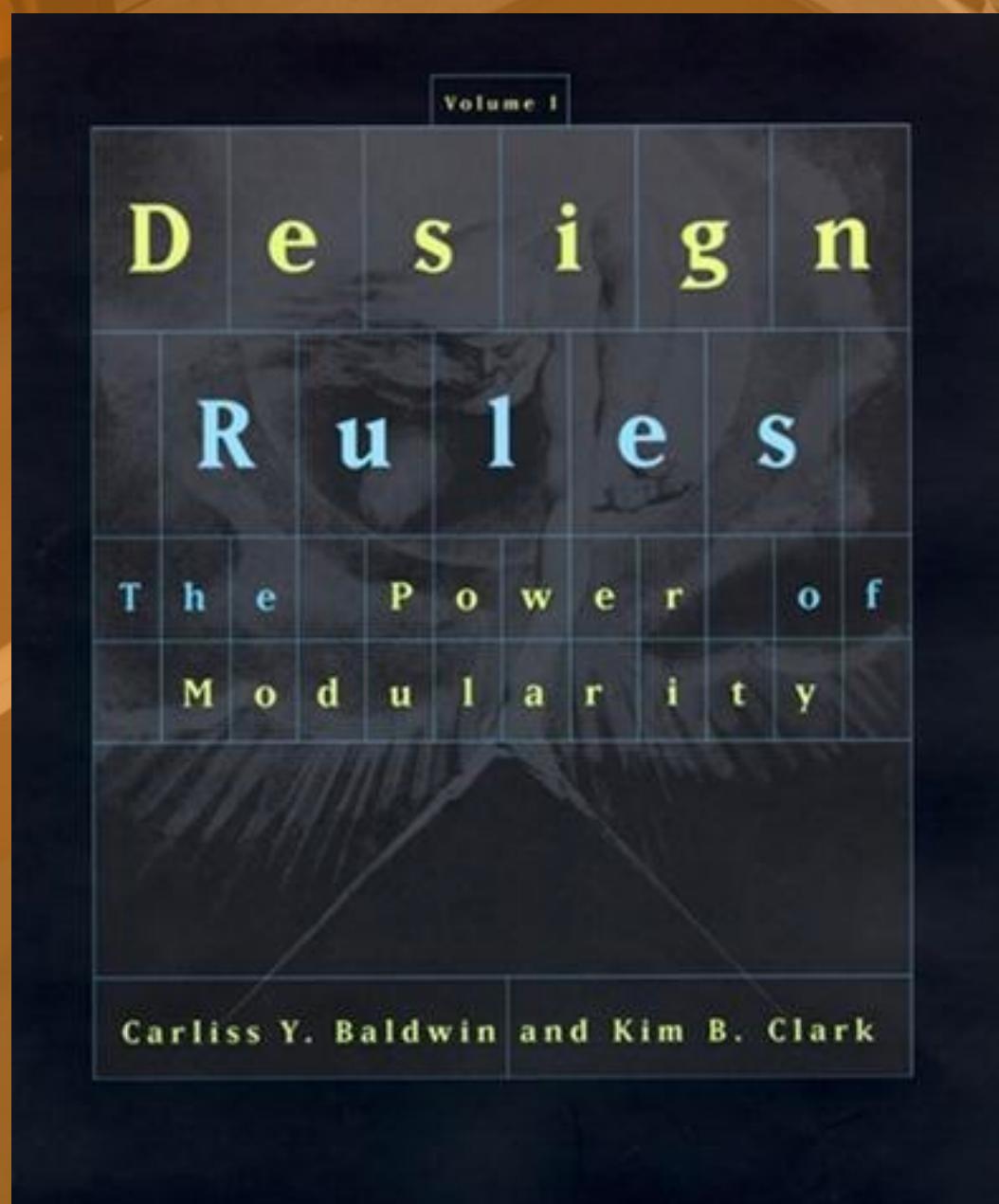
@deanwampler

Modularity Defined



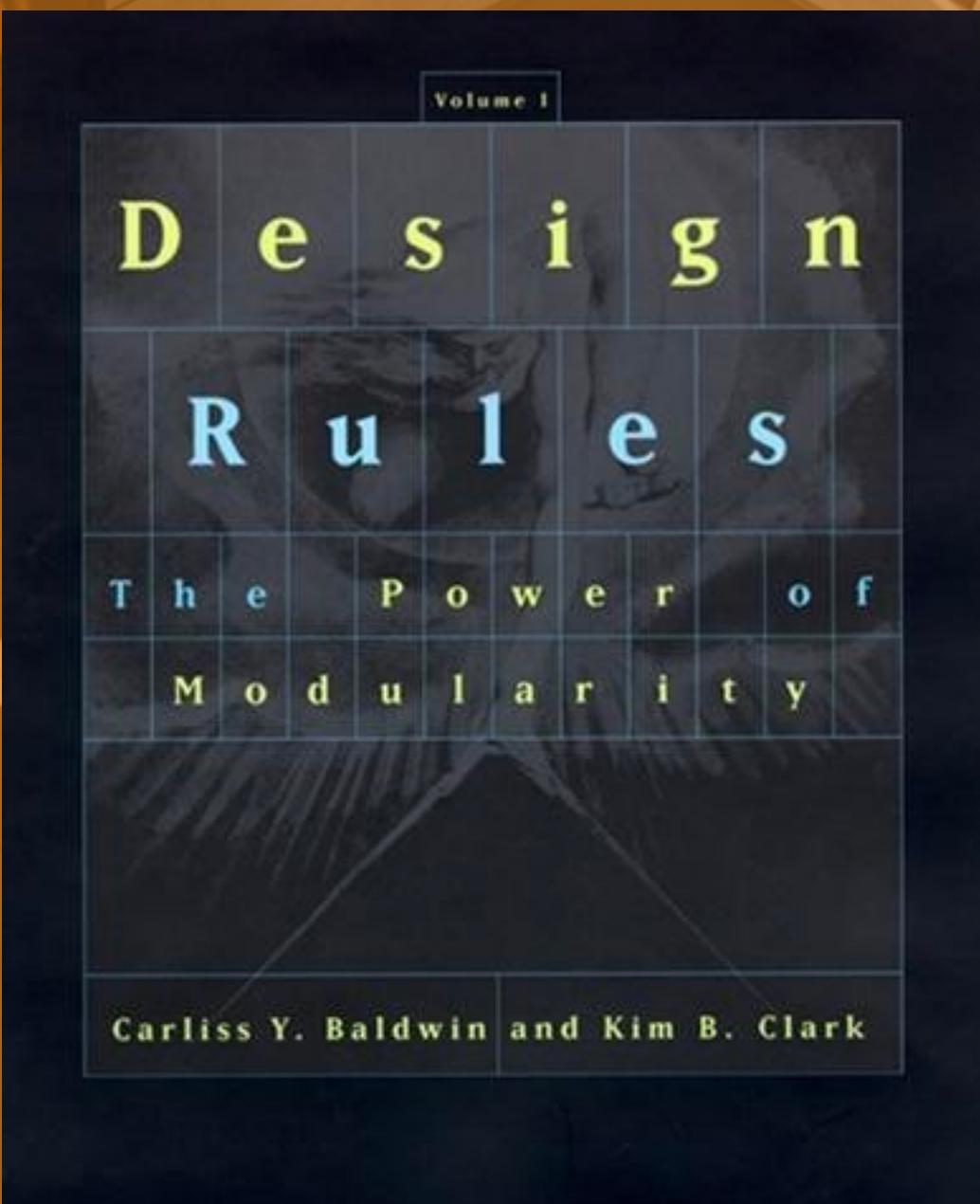
Definition

A module is a unit whose structural elements are powerfully connected among themselves and relatively weakly connected to elements in other units.



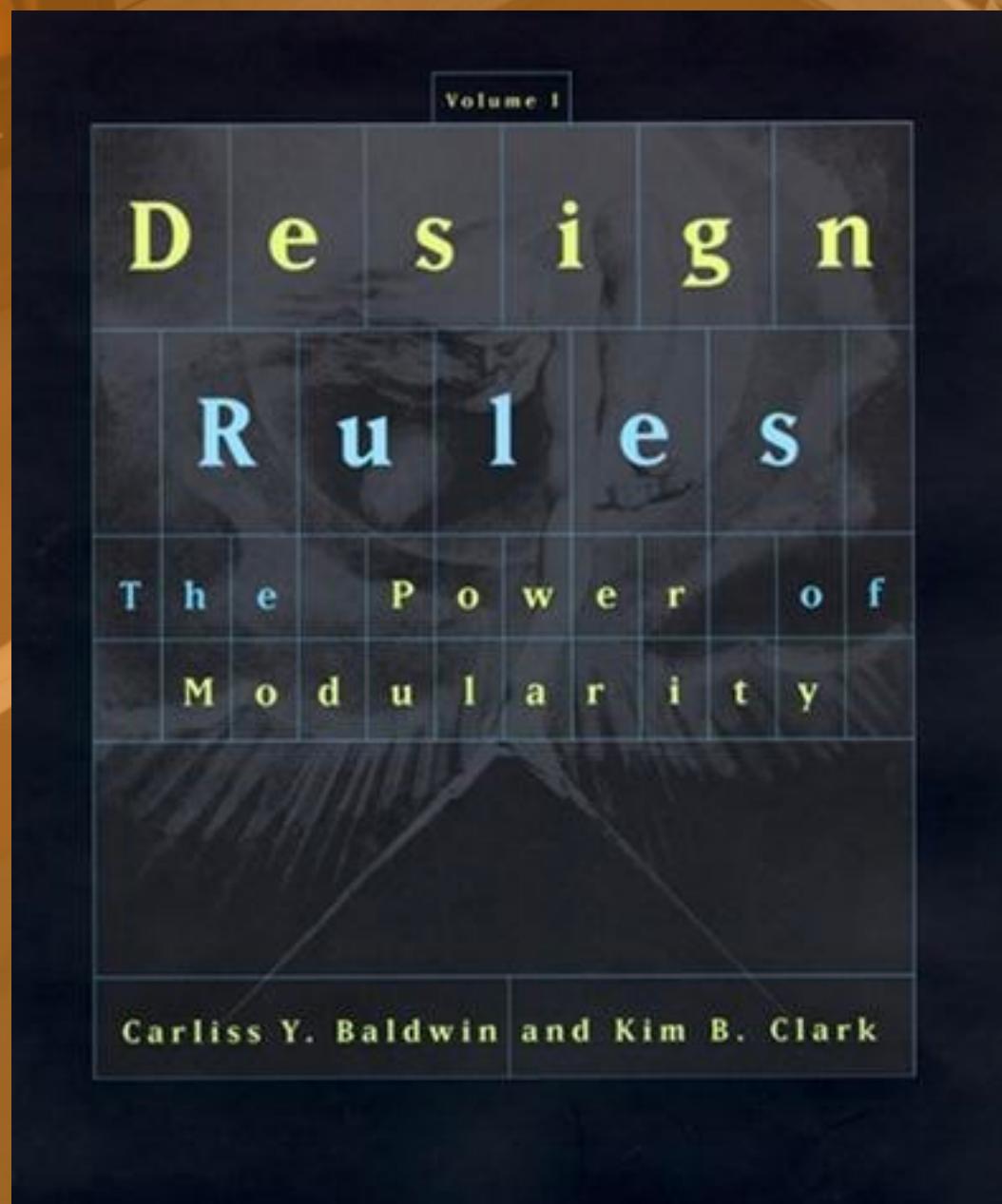
Terms

- Design Rules - fixed decisions
- Architecture - the pieces & how they are wired together
- Interfaces - How they talk to each other



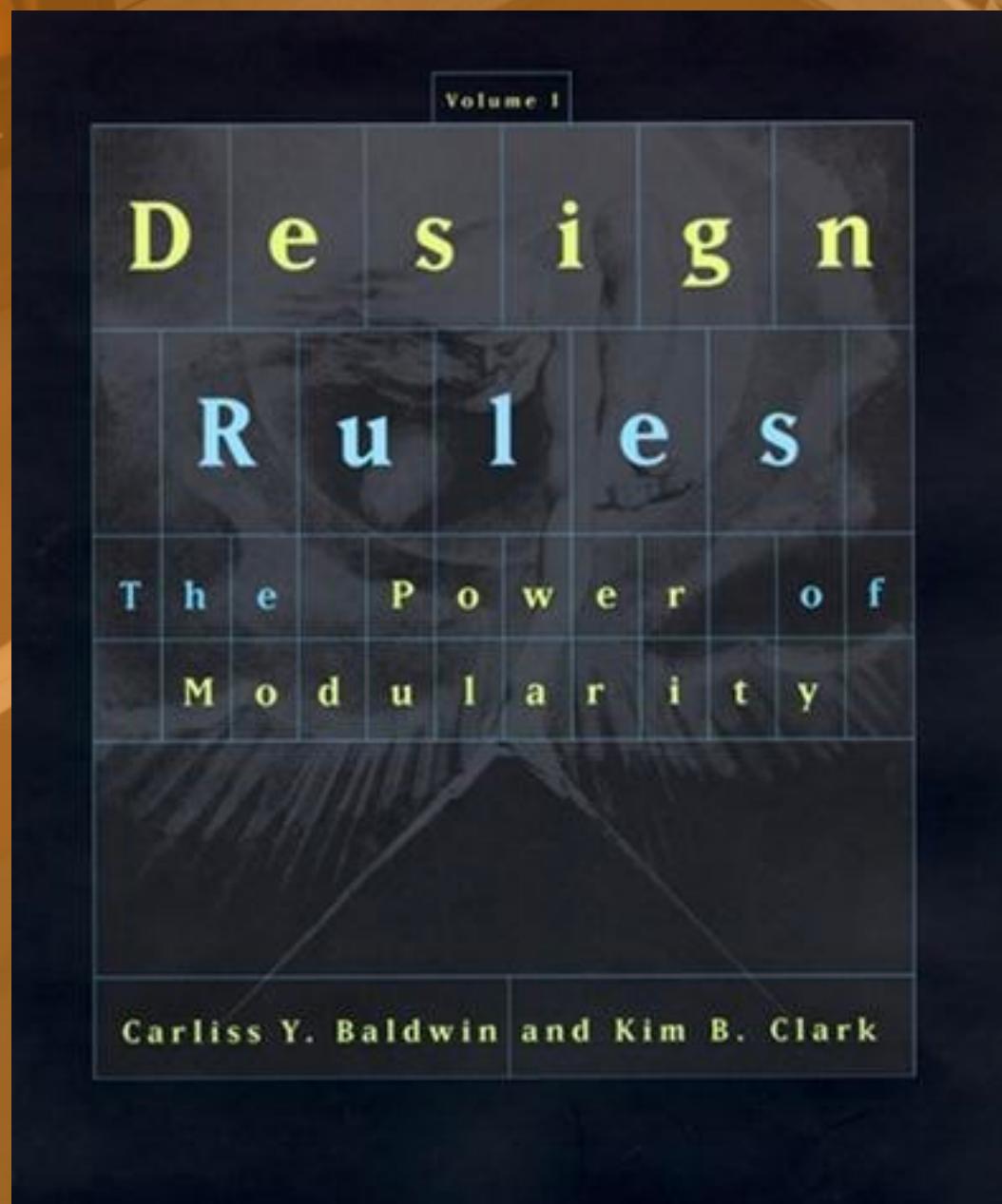
Benefits

- Allows greater complexity
- Allows concurrent work
- Accommodates uncertainty
 - the flexibility to evolve internally



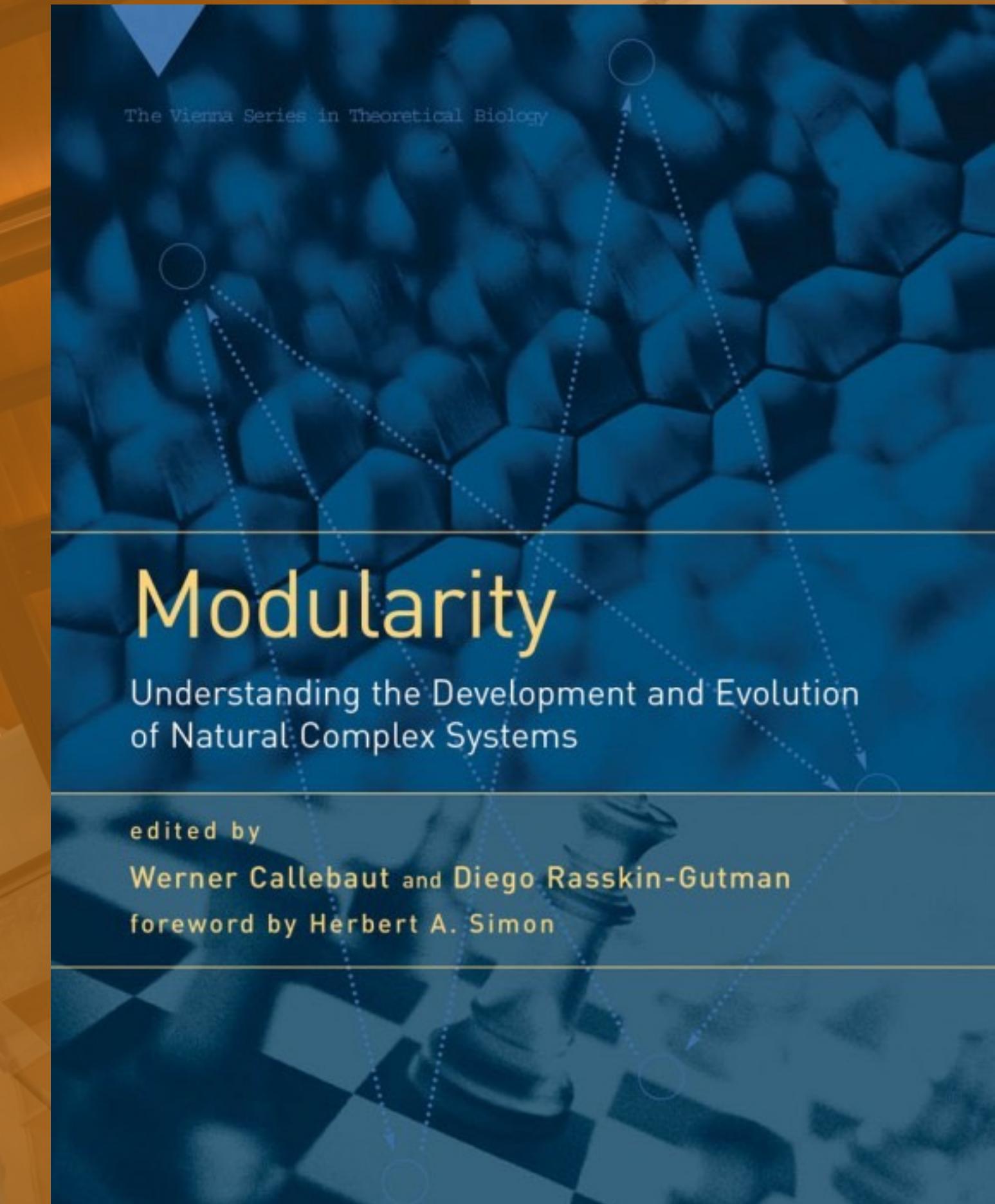
Benefits

- Not stated here is independent
- scalability
- evolution
- replacement



Insights

- Modules can be
 - structural
 - procedural
 - people?



Software

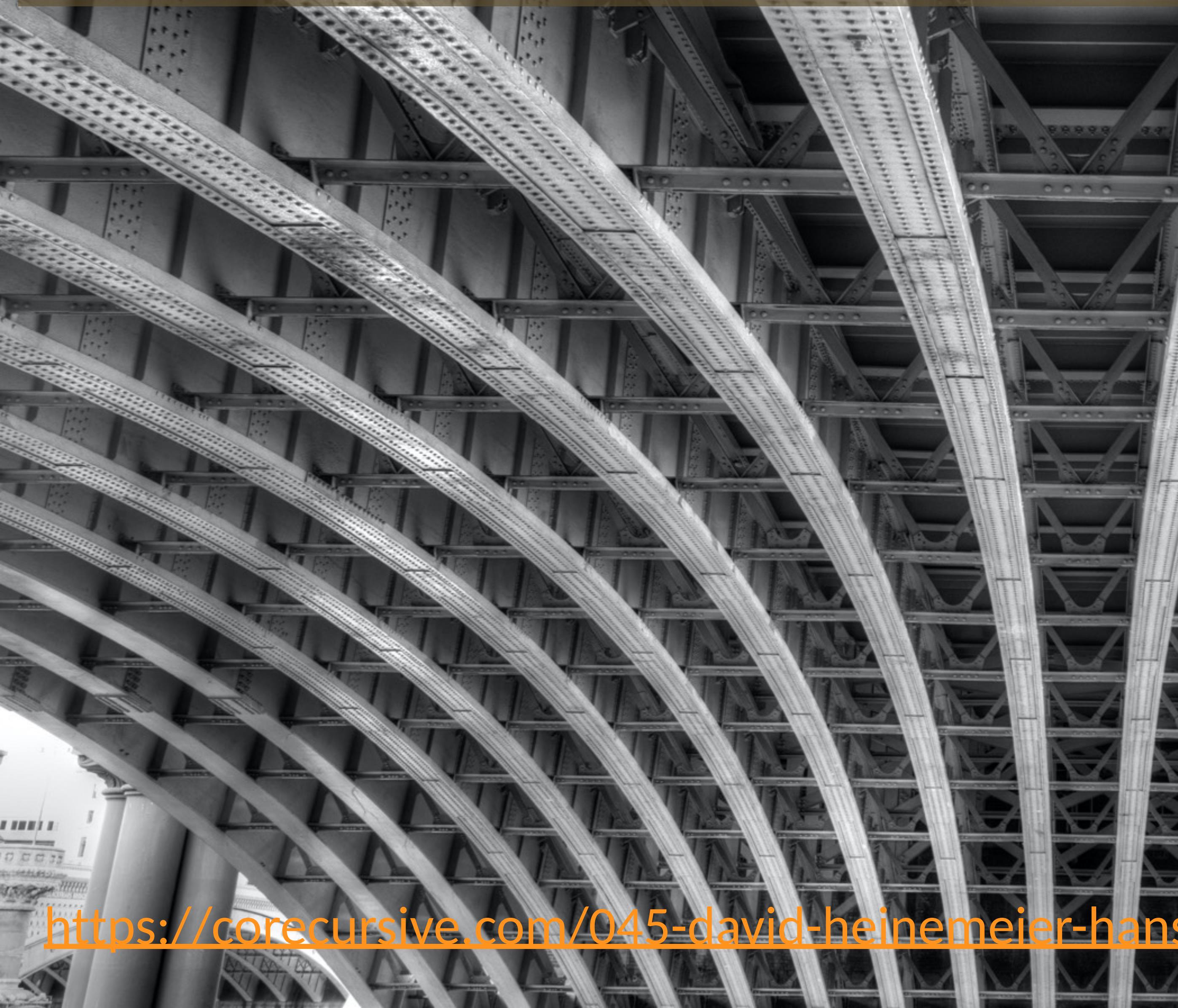


Topics

- Microservices?
- DLL Hell...
- OOP vs. FP vs. ... what?
- New Modularities

Microservices

Microservices



Yawar Amin @yawaramin · 21h

'I think microservices and the hype around them is one of the most damaging trends that has hit web development in the last ten years.'

'Microservices is what you do when you have teams so large that they essentially need ... [full control]'

- @dhh

[corecursive.com/045-david-hein...](https://corecursive.com/045-david-heinemeier-hansson-software-contrarian/)

Troof!



David Heinemeier Hansson, Software Contrarian - CoRecursive Podcast

David Heinemeier Hansson talks to Adam about being avoiding a software monoculture. He explains why we should find a programming language th...

corecursive.com

3

1

20



Show this thread

<https://corecursive.com/045-david-heinemeier-hansson-software-contrarian/>

@deanwampler

Microservices

- Why Microservices?
- Conway's Law
- Independent scaling, etc.

Istio



Istio 1.4

Docs

Istio 1.4.4 is now available! Click here to learn more.

- Abandoned Microservices?
- Because most deployments have one administrator.
- They release all components together



Istio

Connect, secure, control, and monitor your services.

<https://istio.io/>



DLL hell

DLL hell

holden karau
@holdenkarau

If I never have to debug a Guava diamond dependency once more in my life it will be too many.

9:43 PM · Feb 17, 2020 · Twitter Web App

19 Likes

Comment Retweet Like Share

- Is semantic versioning really enough?

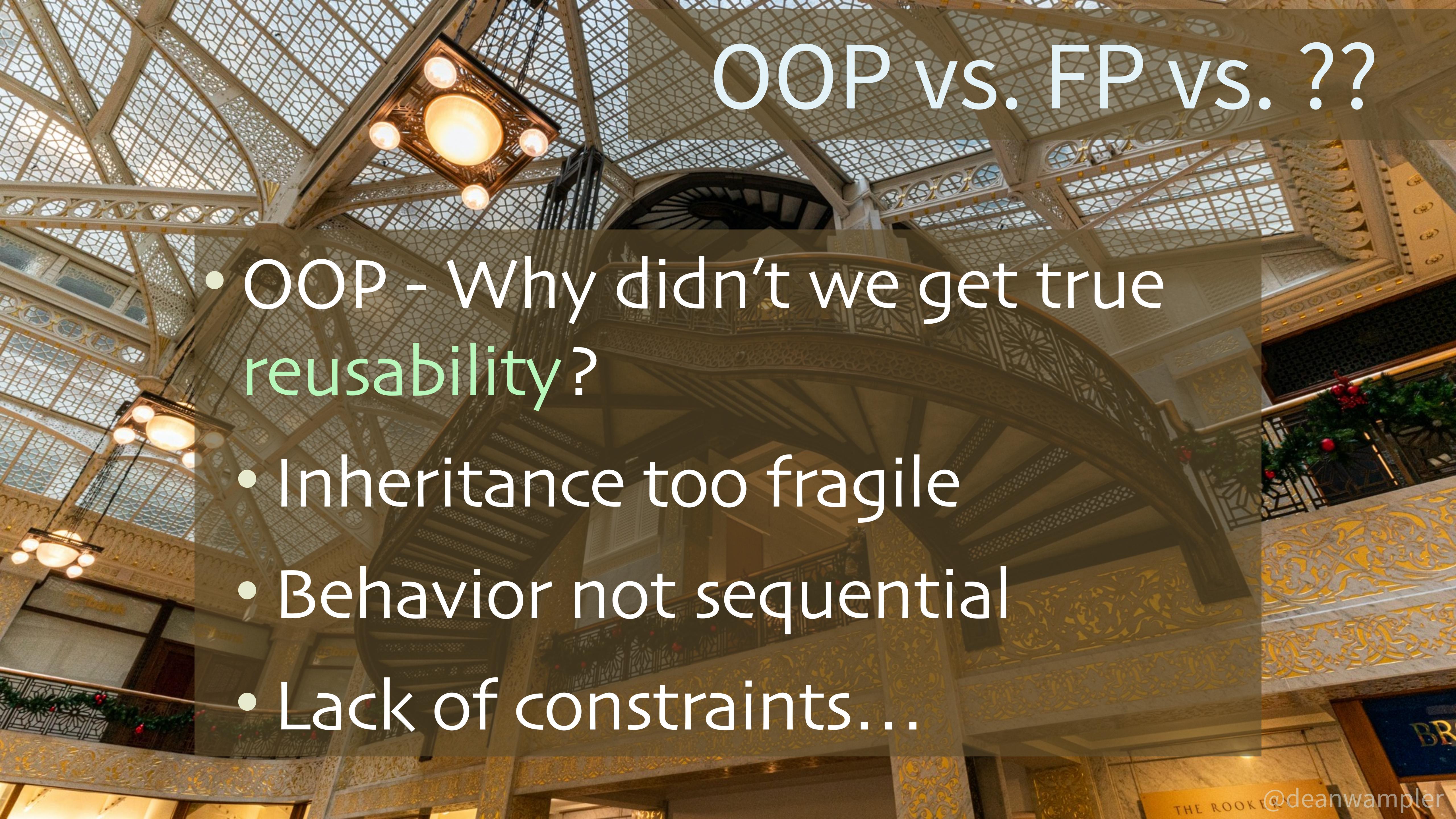
DLL hell

- Clojure - one you release a function, type, etc., never change it.
- Can we define more rigorous specifications of interfaces?



OOP vs. FP vs. ??

@deanwampler



OOP vs. FP vs. ??

- OOP - Why didn't we get true reusability?
- Inheritance too fragile
- Behavior not sequential
- Lack of constraints...

Constraints

Constraints



Constraints



<https://www.youtube.com/watch?v=GqmsQeSzMdw>

@deanwampler

Constraints



Constraints



The more kinds of things something could *potentially* be,
the less we can reason about what it *actually* is.

Constraints

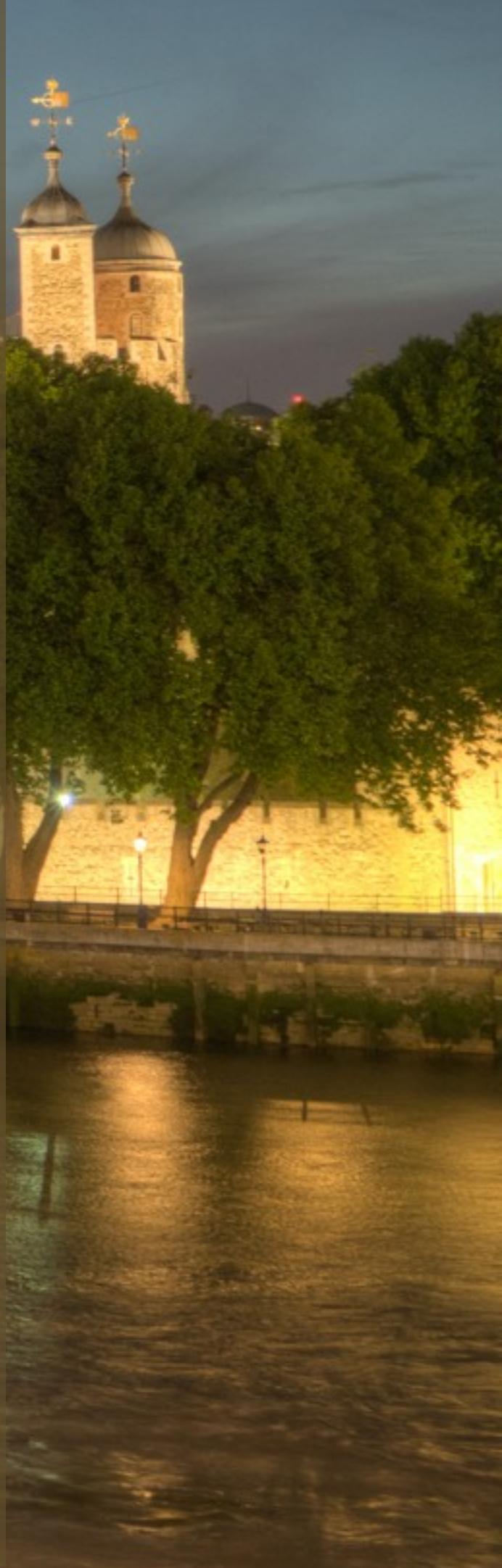
```
def foo(array: Array[Int]): Int
```

vs.

```
def foo[A](array: Array[A]): Int
```

Constraints

- Side effects? - they undermine
- reuse
- compositionality
- concurrency
- “algebraic reasoning”



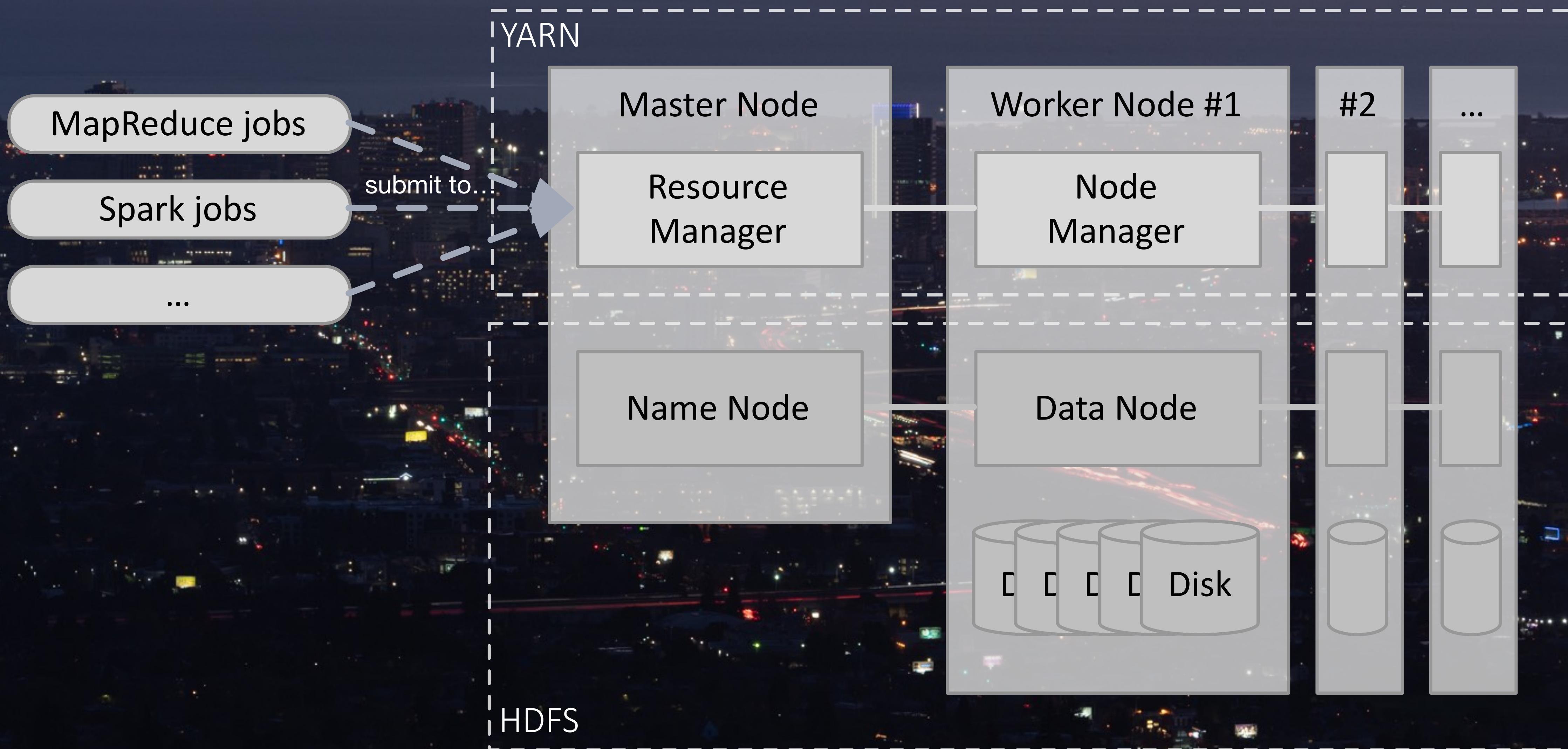
A Current Trend



Databases vs. Hadoop

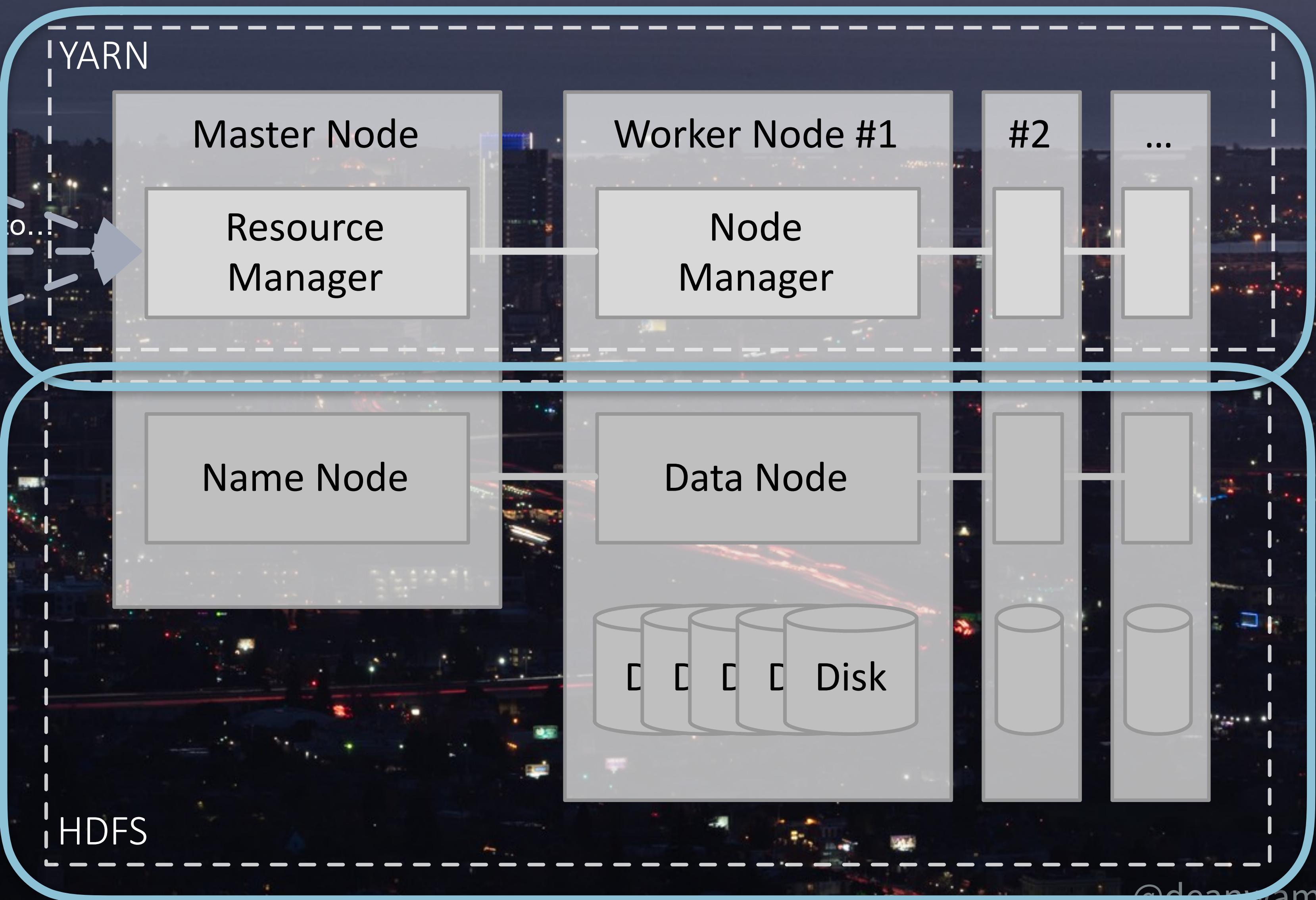
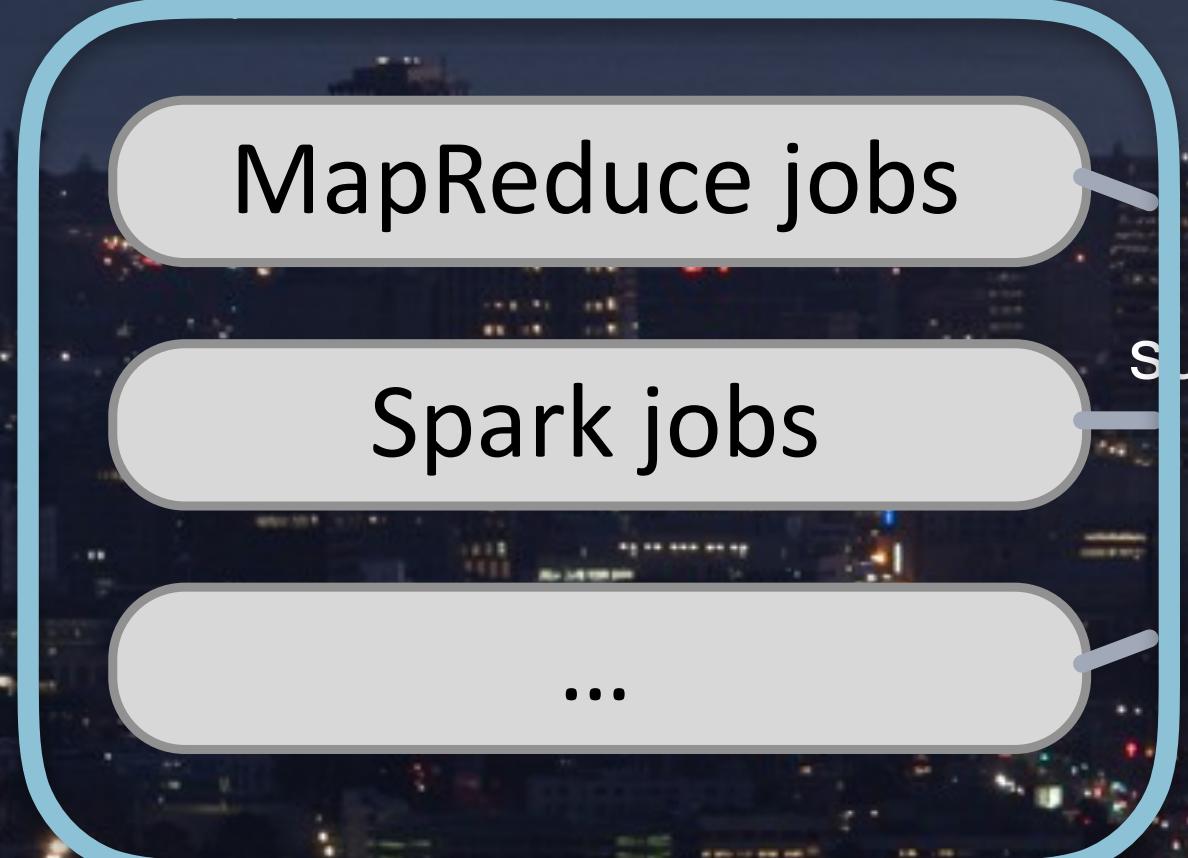


Databases vs. Hadoop



Databases vs. Hadoop

Database
Deconstructed!

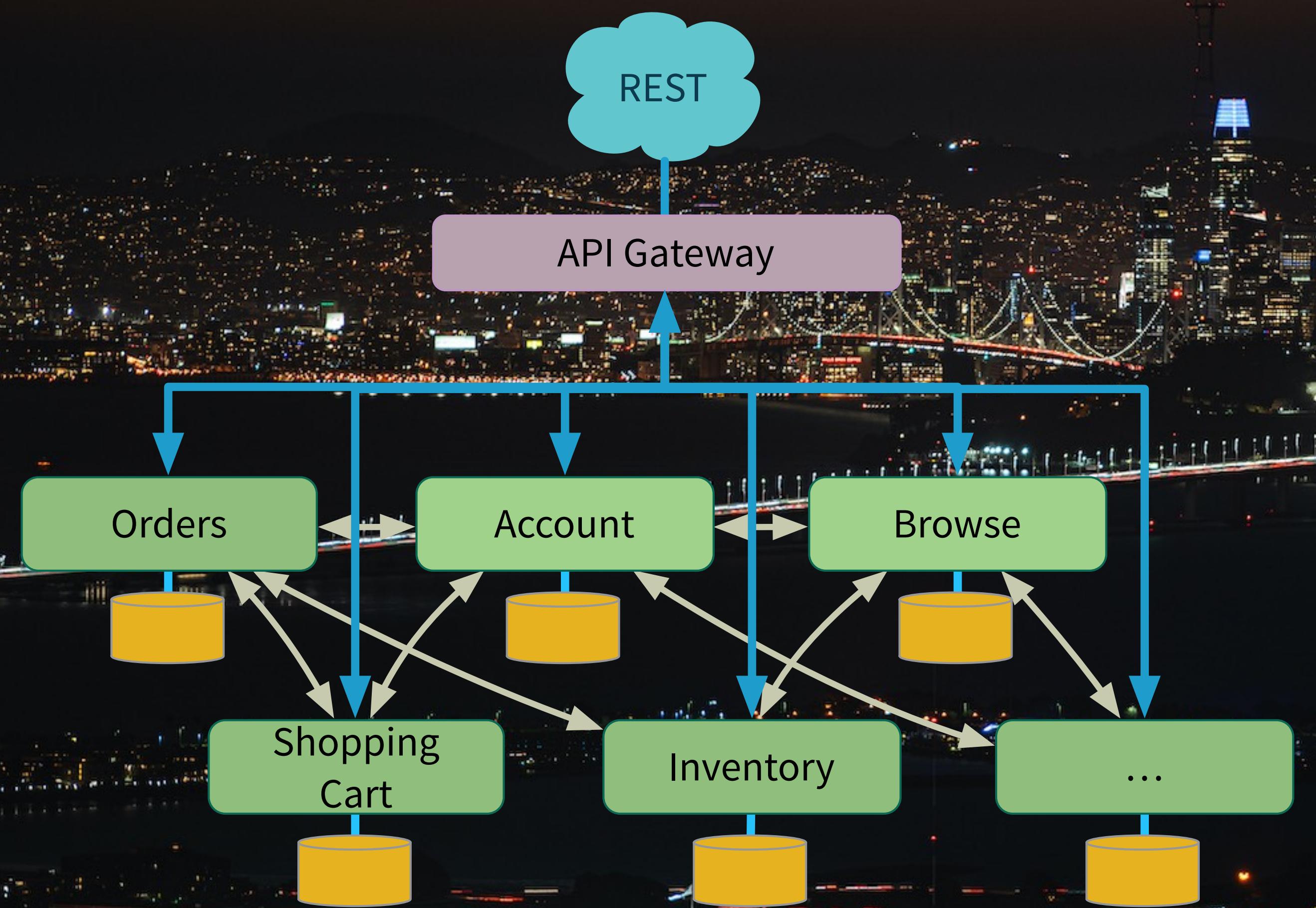


Ray

ray.io

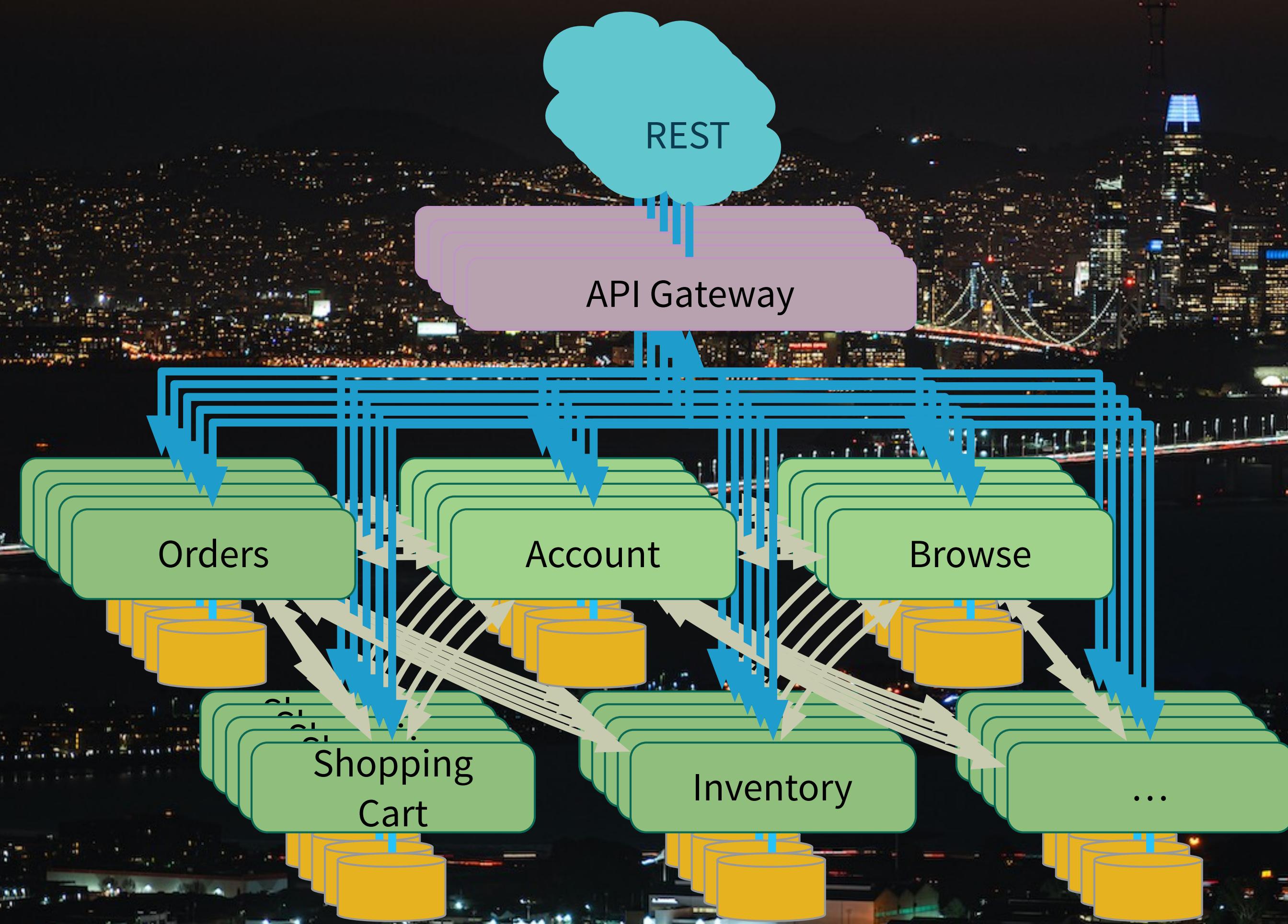
@deanwampler

Microservices (again)



Microservices (again)

Need to Scale??

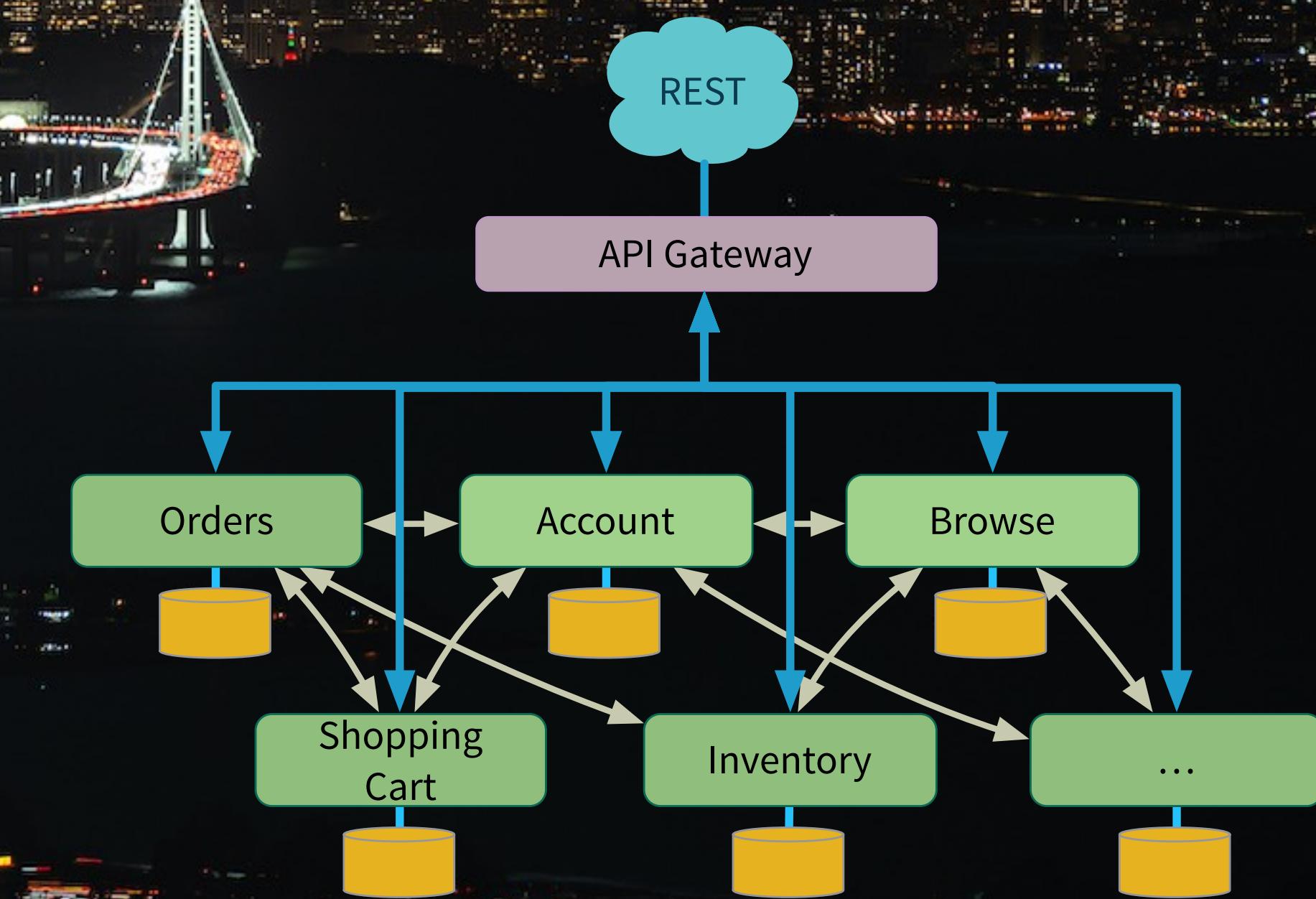


Why do YOU have to
manage all these
instances yourself??

Ray

- Pretend you have ONE instance of the microservice.
- Let systems like Ray manage horizontal scaling.

ray.io



Apache Arrow

- Does something similar for in-memory data formats
- Standardize how this is done
- Reuse across apps & libs

arrow.apache.org

The background image shows a panoramic view of the San Francisco Bay area during sunset. The sky is filled with warm orange and yellow hues. In the foreground, the city of Berkeley is visible, with its iconic Campanile tower on the left. Across the bay, the San Francisco skyline and the Golden Gate Bridge are prominent. The water of the bay reflects the setting sun.

The Future?



RAY SUMMIT

Presented by Anyscale

Scalable machine learning, scalable Python, for everyone

May 27 – 28, 2020

FEATURED SPEAKERS



Michael Jordan
Distinguished Professor, University of California, Berkeley



Azalia Mirhoseini
Senior Research Scientist, Google Brain



Wes McKinney
Founder, Ursa Labs



Manuela Veloso
Head of J.P. Morgan AI Research



Gaël Varoquaux
Tenured Research Director, Inria



Ion Stoica
Professor, University of California, Berkeley



Questions?

polyglotprogramming.com/talks
anyscale.io
ray.io

dean@deanwampler.com
@deanwampler