Modularity: A Retrospective

dean@deanwampler.com
@deanwampler

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

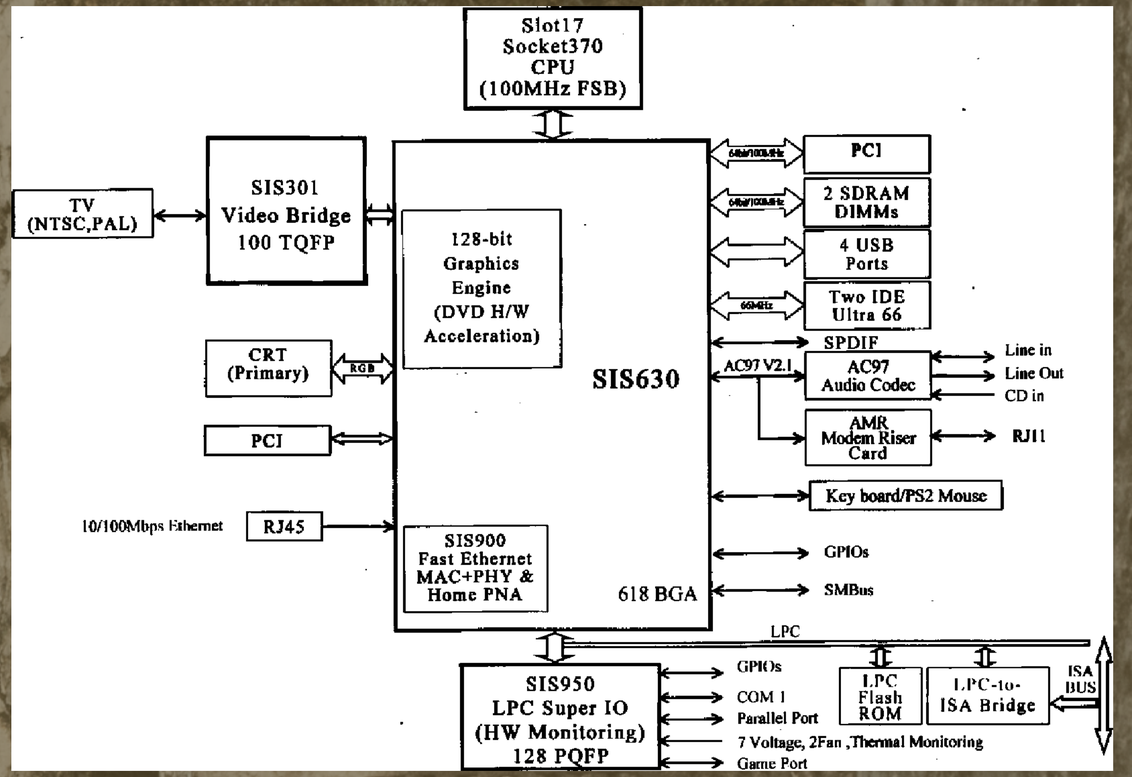
Interchangeable Parts

- 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

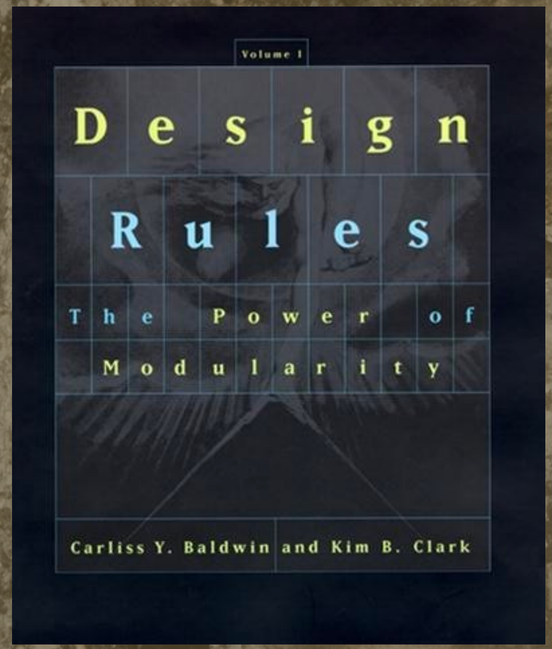
- Electronics
  - → PCs

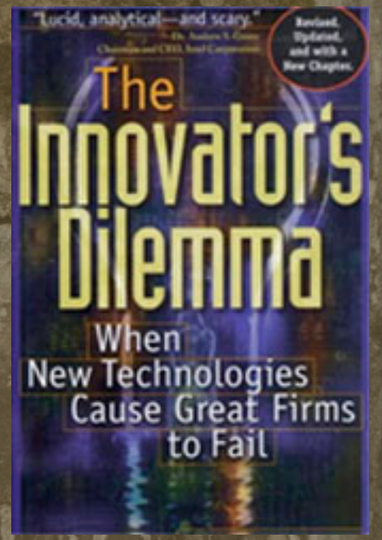


https://fccid.io/NYYTP100/Block-Diagram/BLOCK-DIAGRAM-133927
http://oldcomputers.net/ibm5150.html

Interchangeable Parts

- Electronics
  - → PCs



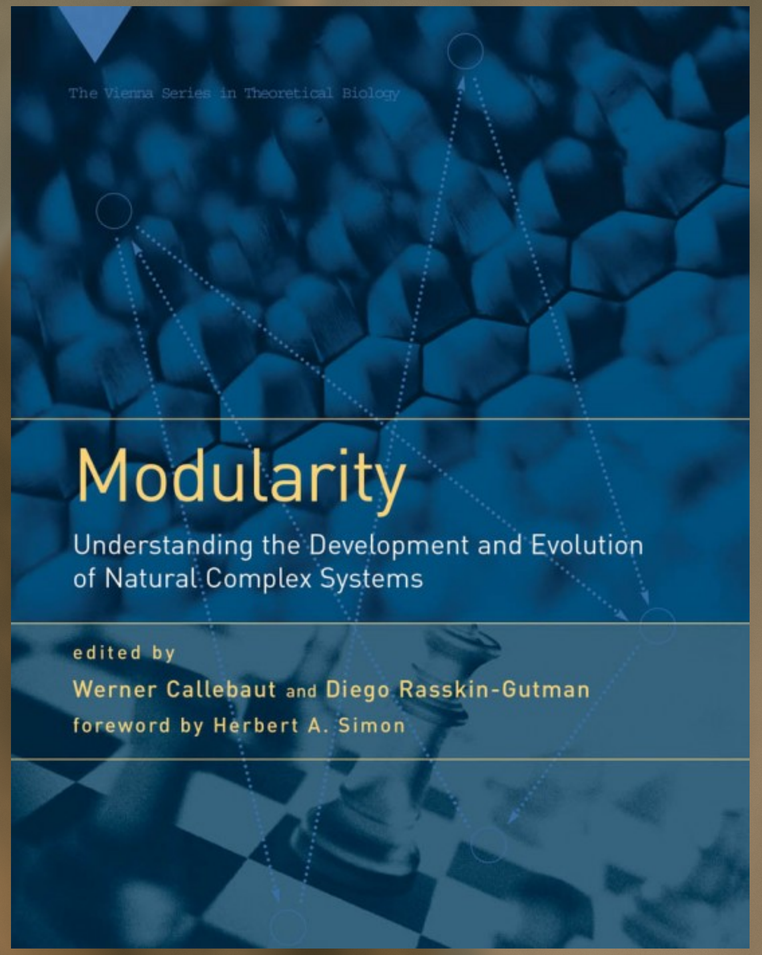


https://mitpress.mit.edu/books/design-rules-volume-1
http://claytonchristensen.com/books/the-innovators-dilemma/


One Global Pattern…
Duality of “Forces”
@deanwampler

In Biology

- RNA evolves energy-stable structures that resist mutation
  - But mutation still possible

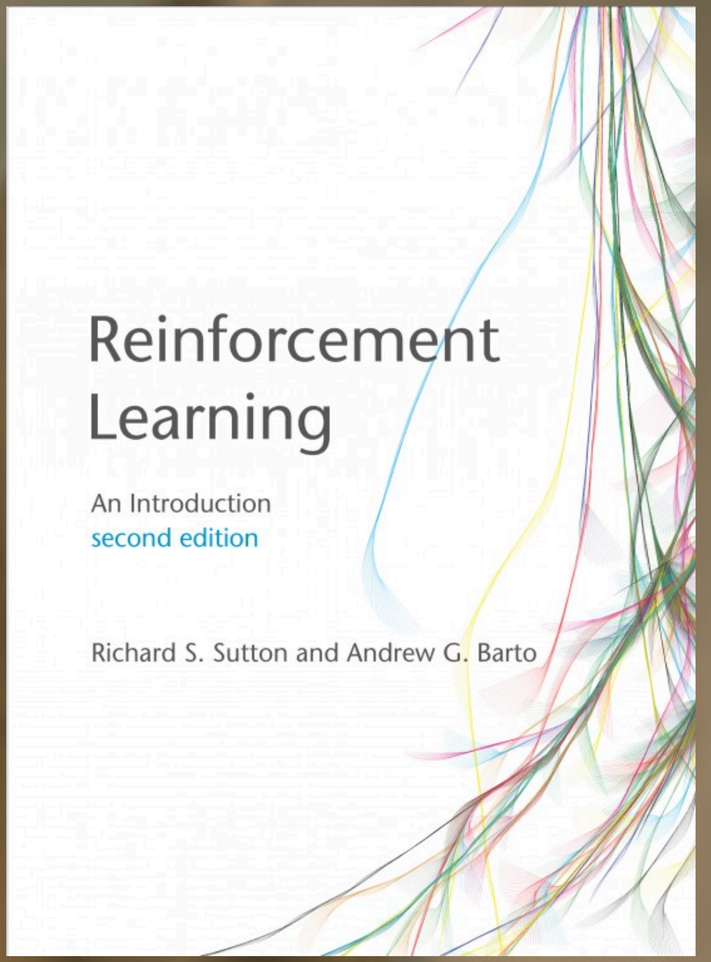


https://mitpress.mit.edu/books/modularity

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

Reinforcement Learning

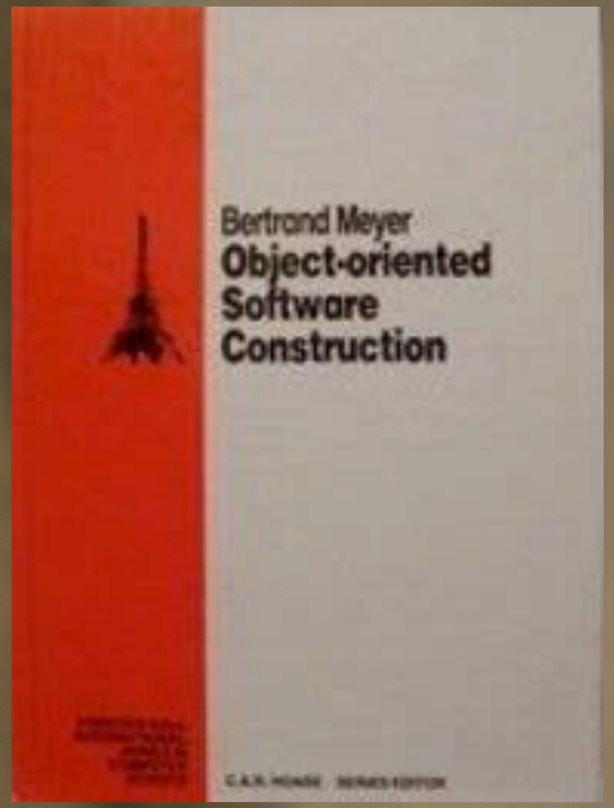
- Exploit rewarding paths
- … but explore other paths that might be more rewarding



https://mitpress.mit.edu/books/reinforcement-learning-second-edition

Open-Closed Principle

- A module should open for extension
- … but closed for modification



Wikipedia Page

<rant/>

- All systems have design patterns, including FP
  - Specific patterns will vary

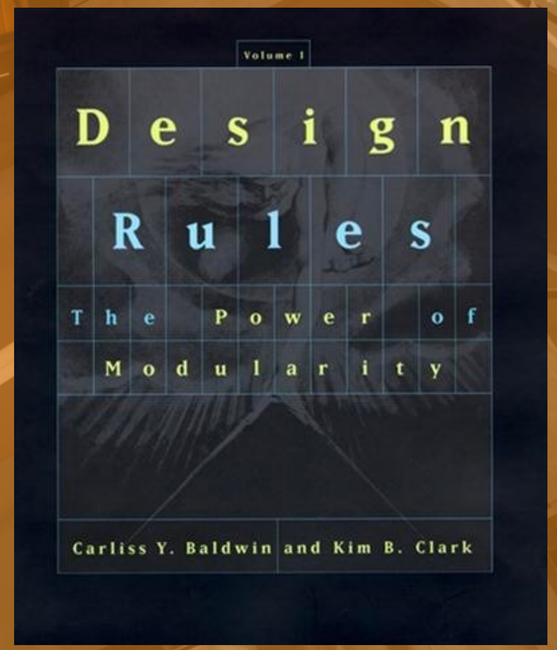
FP “patterns”? Monads, monoids, applicatives, functors, …

Modularity Defined

Definition

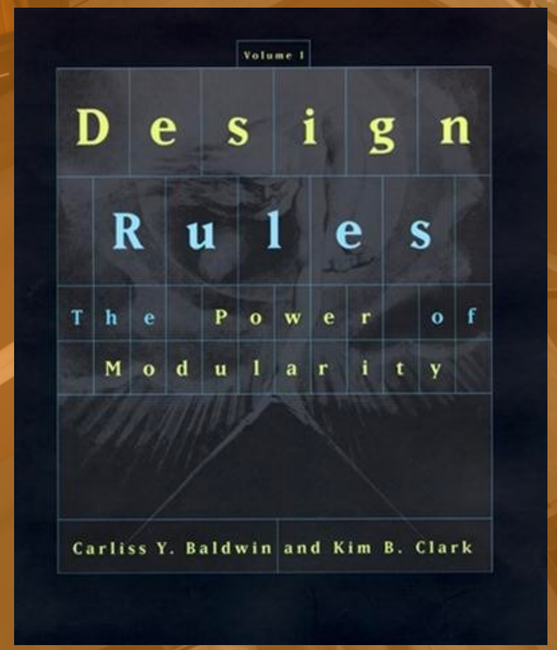
I.e., Cohesion vs. Coupling

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



Terms

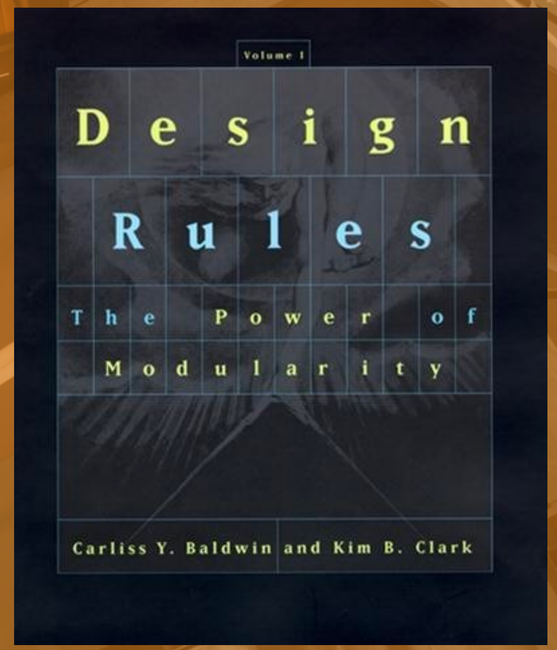
- Design Rules - fixed decisions
- Architecture - the pieces & how they are wired together
- Interfaces - How they talk to each other



Benefits

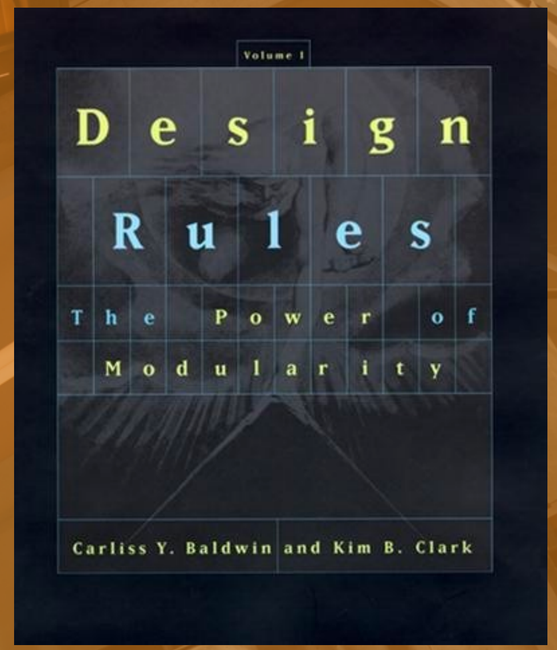
- Allows greater complexity
- Allows concurrent work
  - Embrace Conway's Law

Wikipedia Page



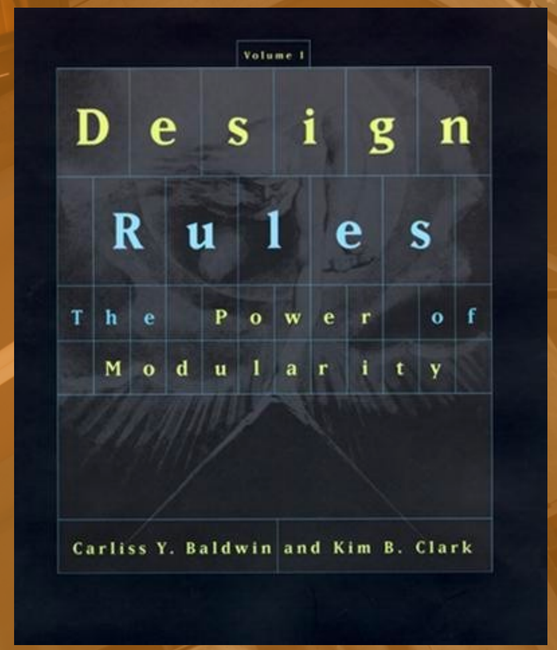
Benefits

- Accomodates uncertainty
  - the flexibility to evolve internally



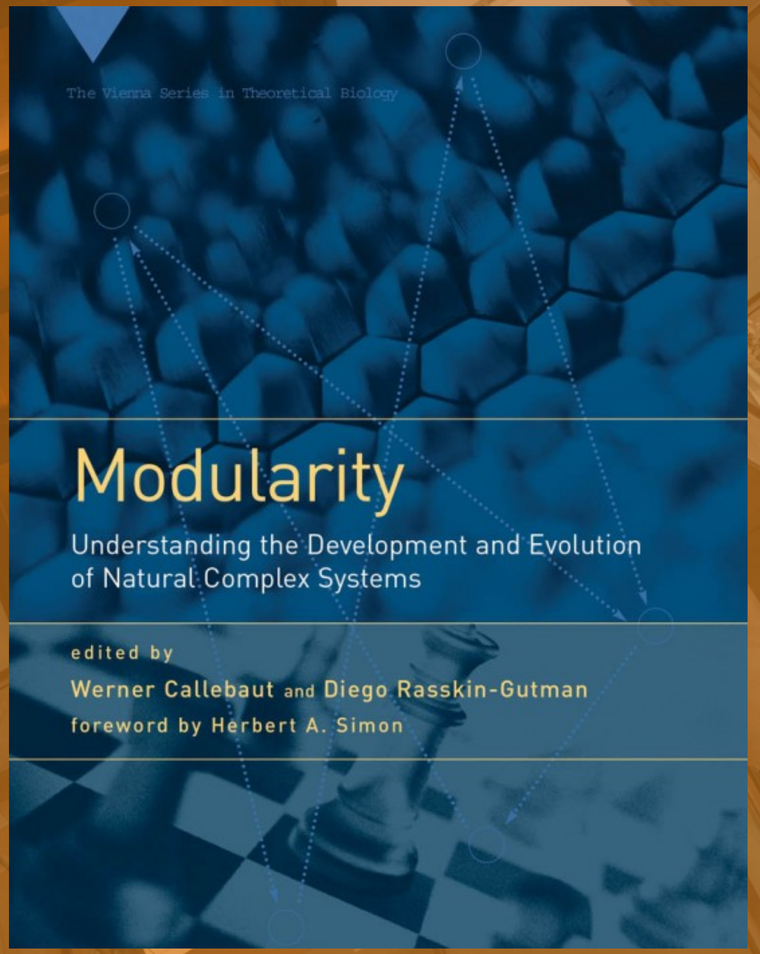
Benefits

- Not stated in the book are "DevOps" concerns, e.g.,
  - Independent scalability
  - Independent replacement



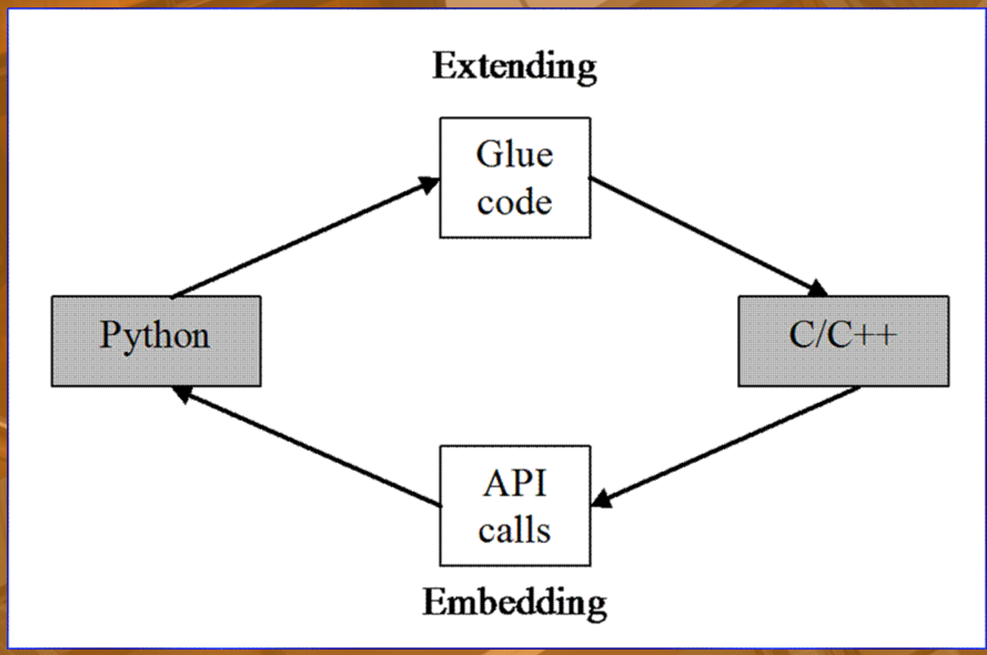
Insights

- Modules can be
  - structural
  - procedural



Insights

- Procedures = Libraries of components + Scripts as glue for procedures



What is the type of a procedure??

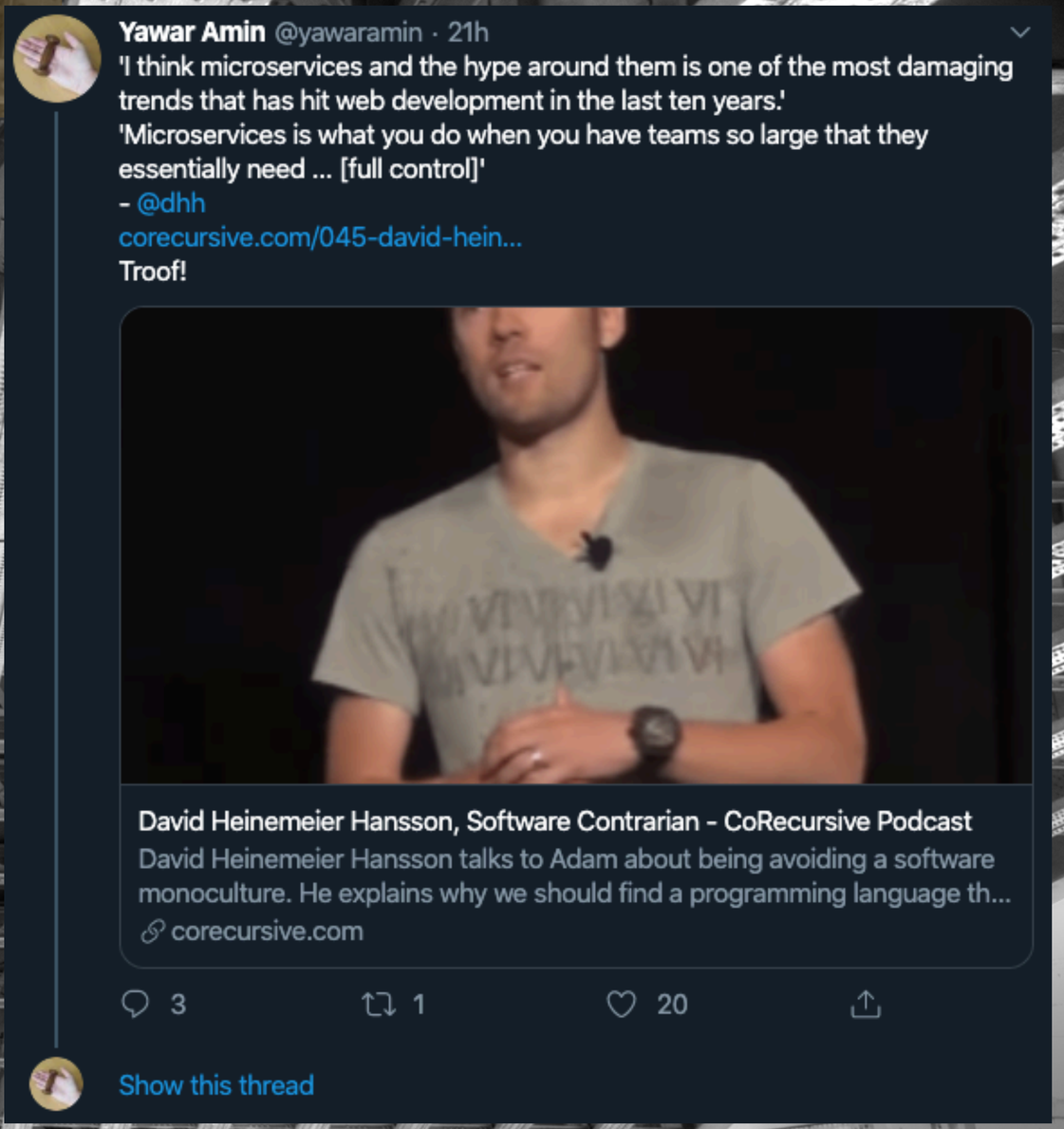
Software

Examples

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

Microservices

Microservices



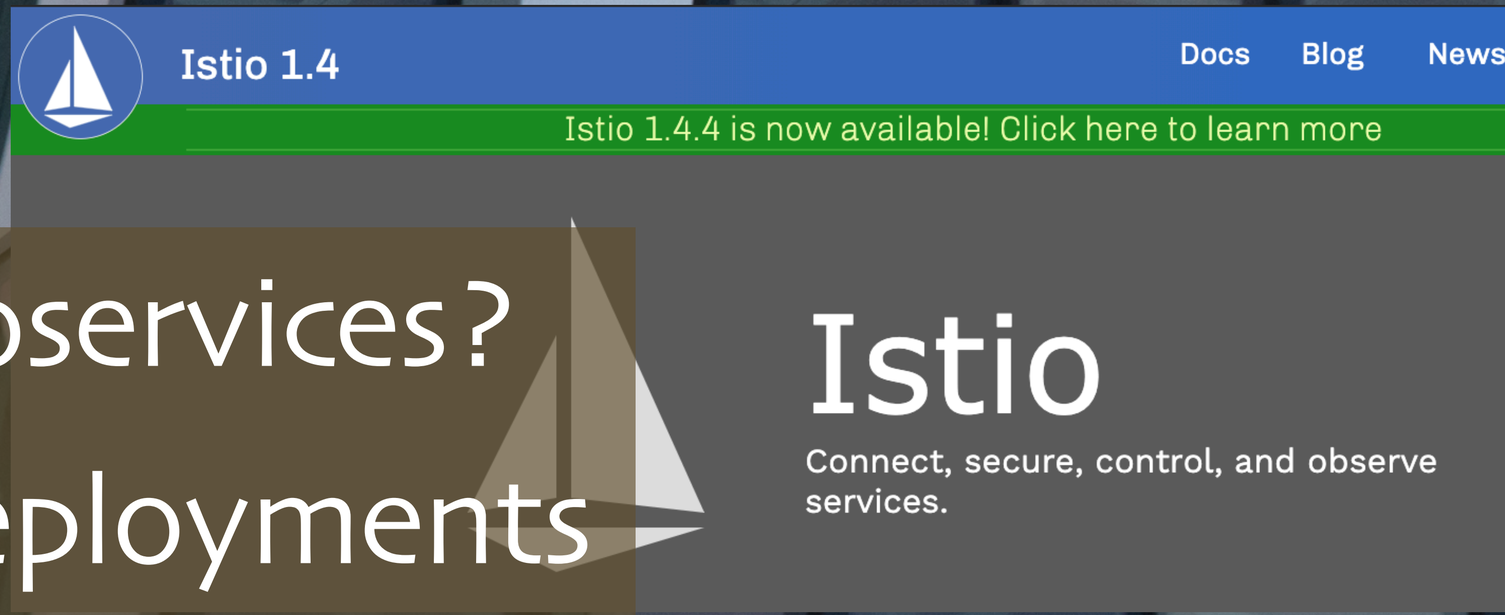
https://corecursive.com/045-david-heinemeier-hansson-software-contrarian/

Microservices

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

Istio

- Abandoned Microservices?
  - Because most deployments have one administrator
  - They now release as one deployable service

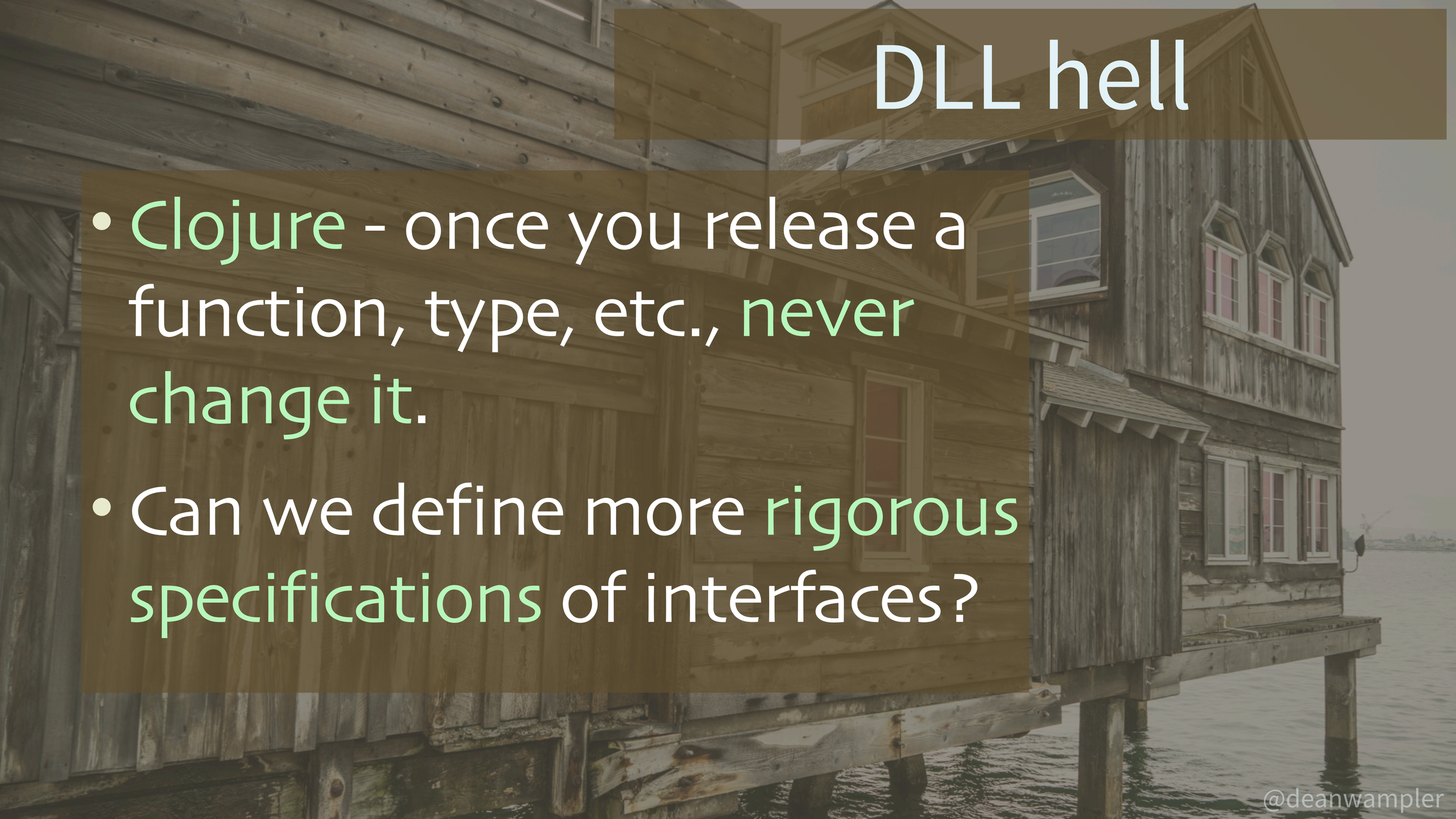


https://istio.io/

DLL hell

DLL hell

- Is semantic versioning really enough?



DLL hell

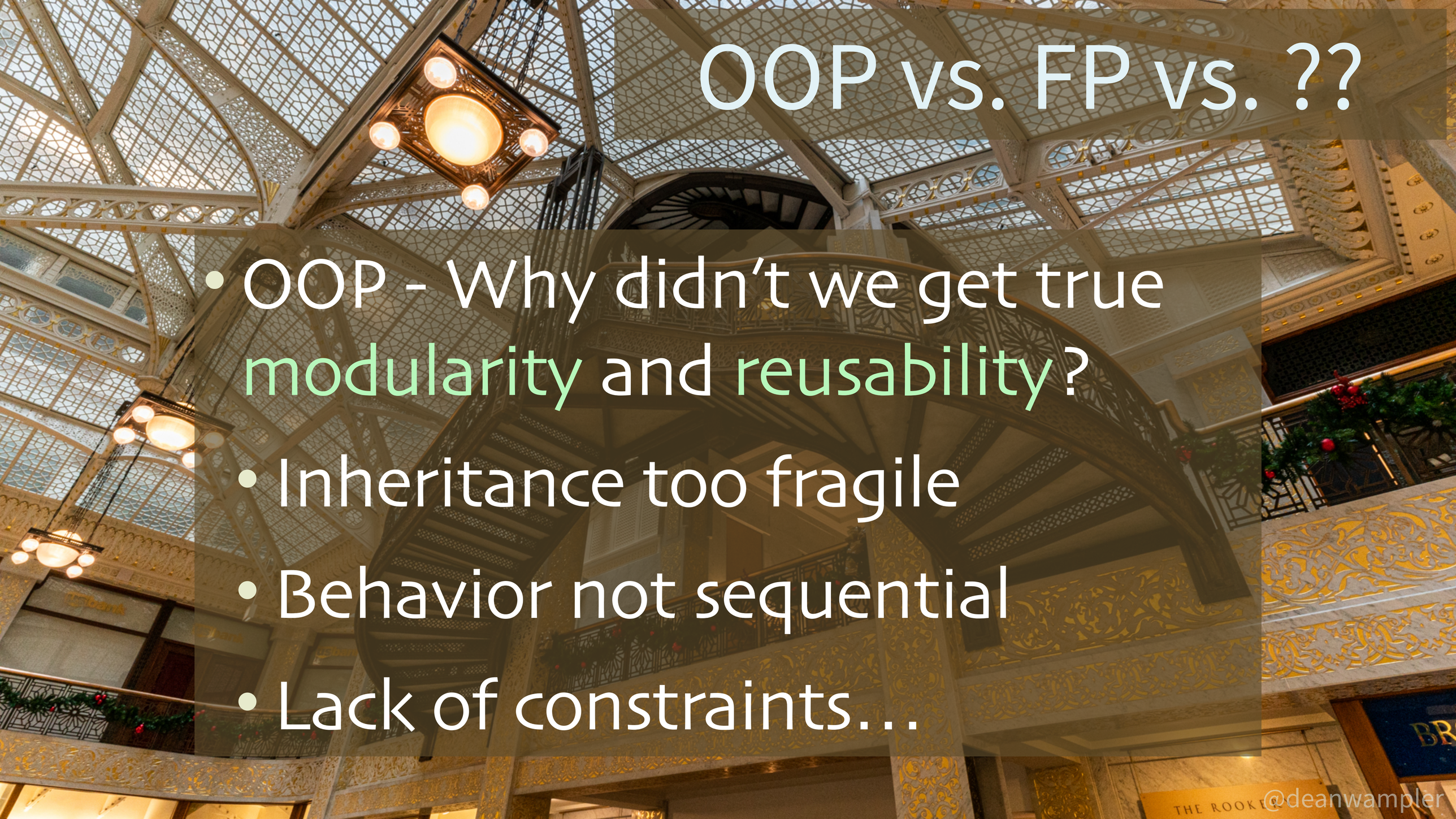
- Clojure - once you release a function, type, etc., never change it.
- Can we define more rigorous specifications of interfaces?

DLL hell

- Unison language
  - Why do we rely on names?

https://www.unisonweb.org/2020/04/10/reducing-churn/


OOP vs. FP vs. ??
DEAN OPTICAL
@deanwampler



OOP vs. FP vs. ??

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

Constraints

@deanwampler

Constraints

Constraints Liberate,
Liberties Constrain

Rúnar Bjarnason @runarorama
Scala World 2015

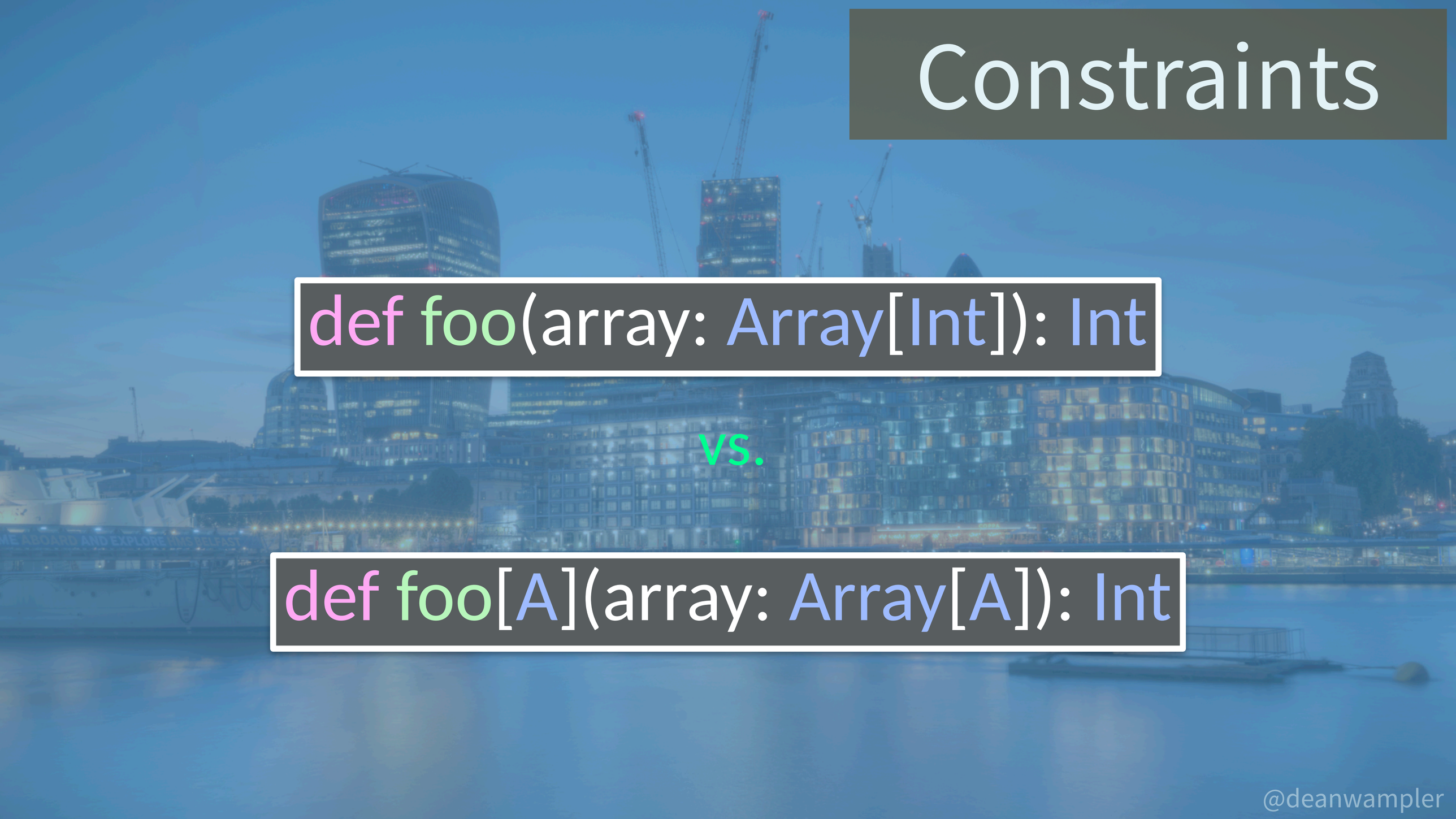
https://www.youtube.com/watch?v=GqmsQeSzMdw

Constraints

Constraints

The more kinds of things
something could *potentially* be,

the less we can reason about
what it *actually* is.

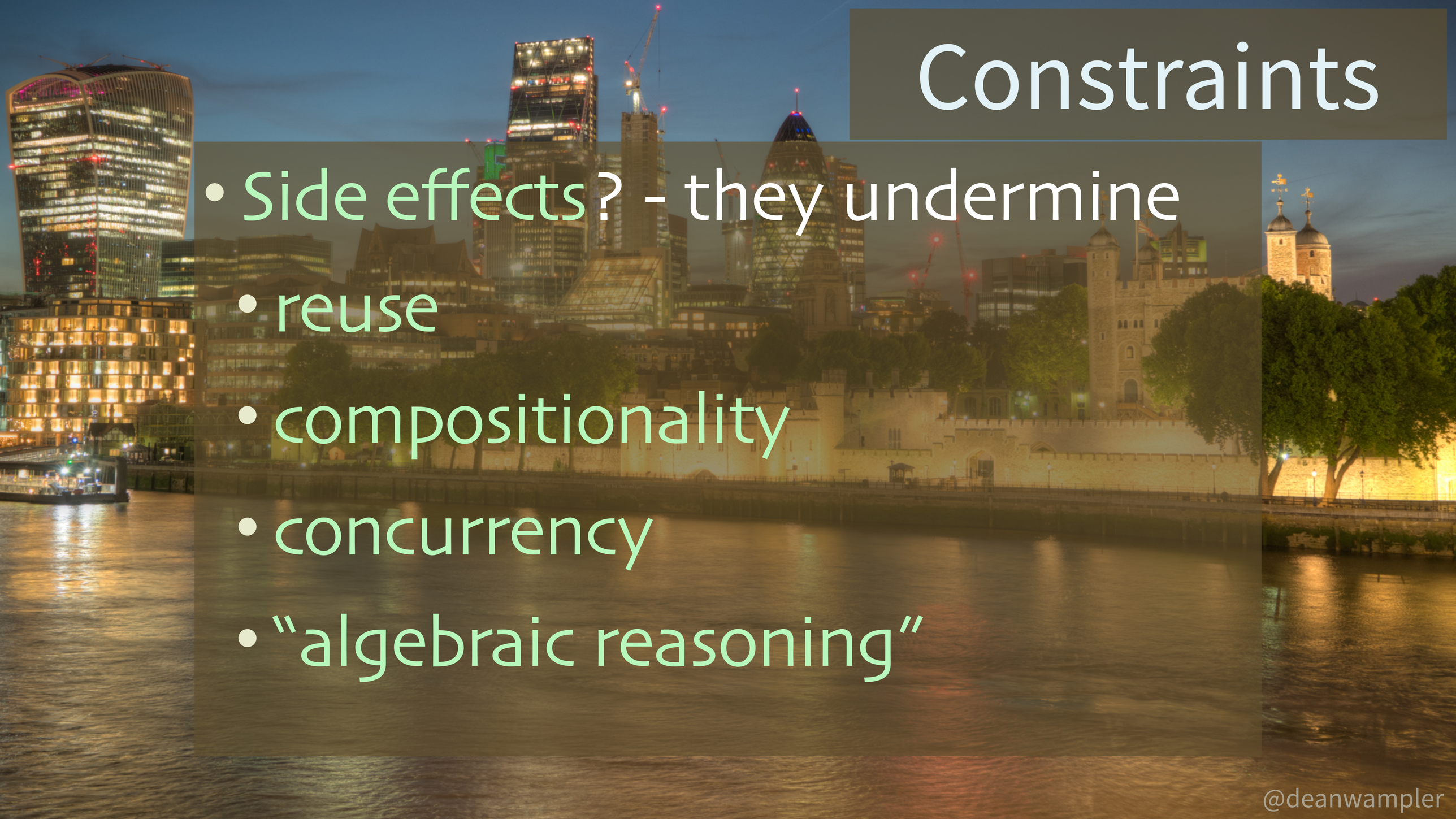


Constraints

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

vs.

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



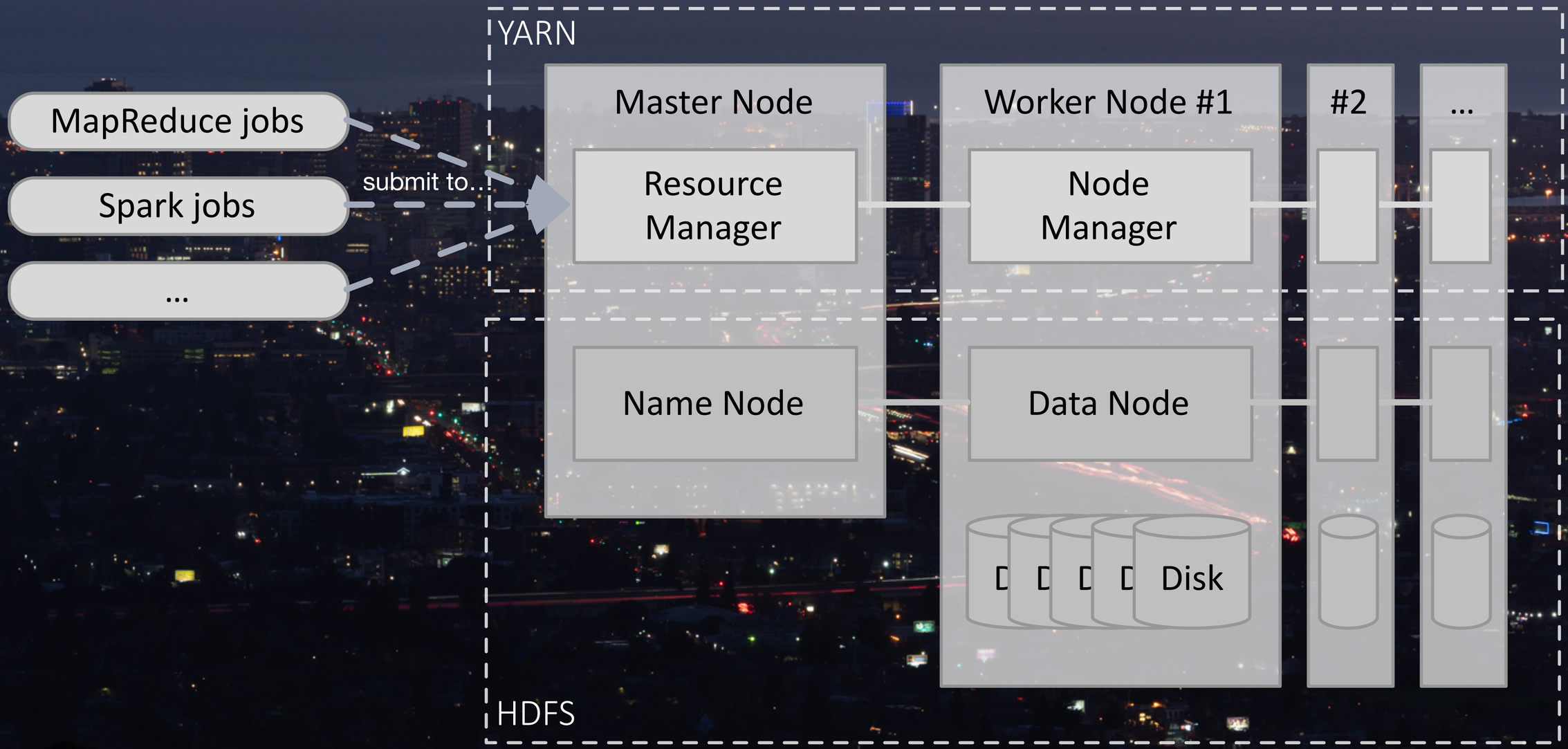
Constraints

- Side effects? - they undermine
  - reuse
  - compositionality
  - concurrency
  - "algebraic reasoning"

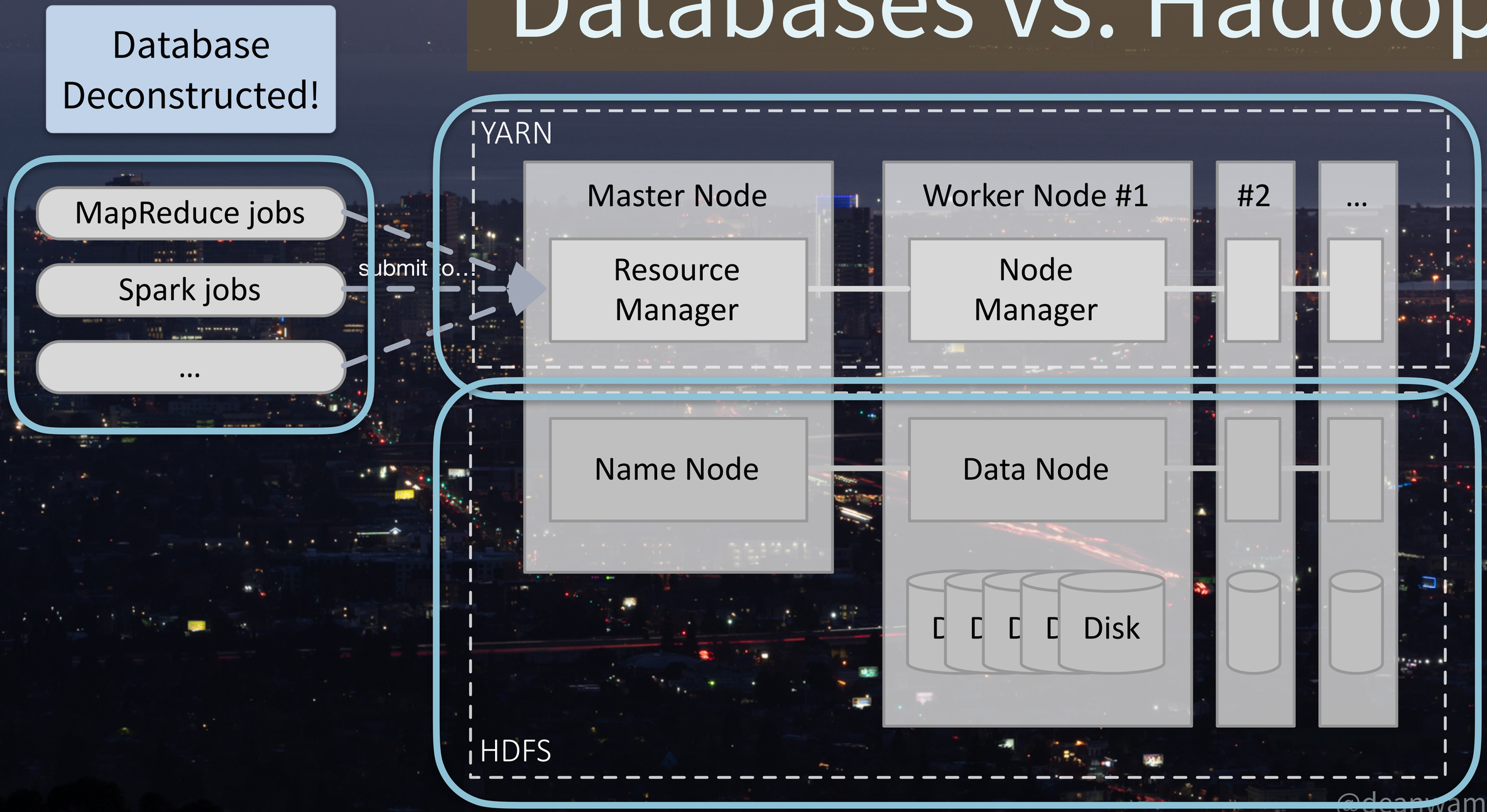
Success Stories

Databases vs. Hadoop

Databases vs. Hadoop



Databases vs. Hadoop



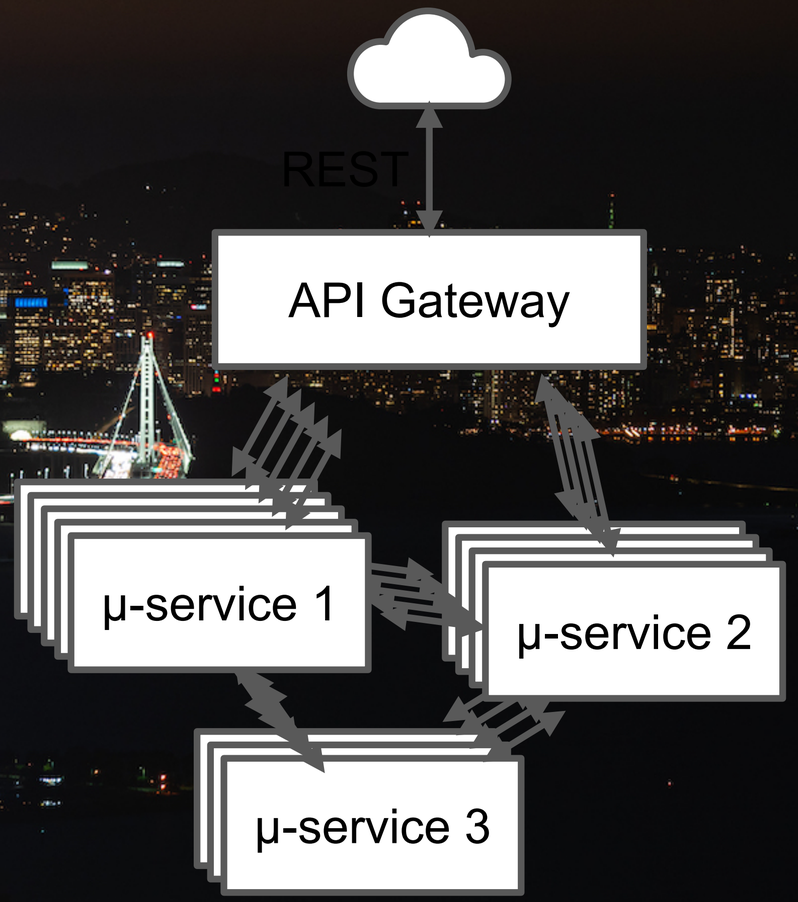
Ray (and systems like it…)

Microservices (again)

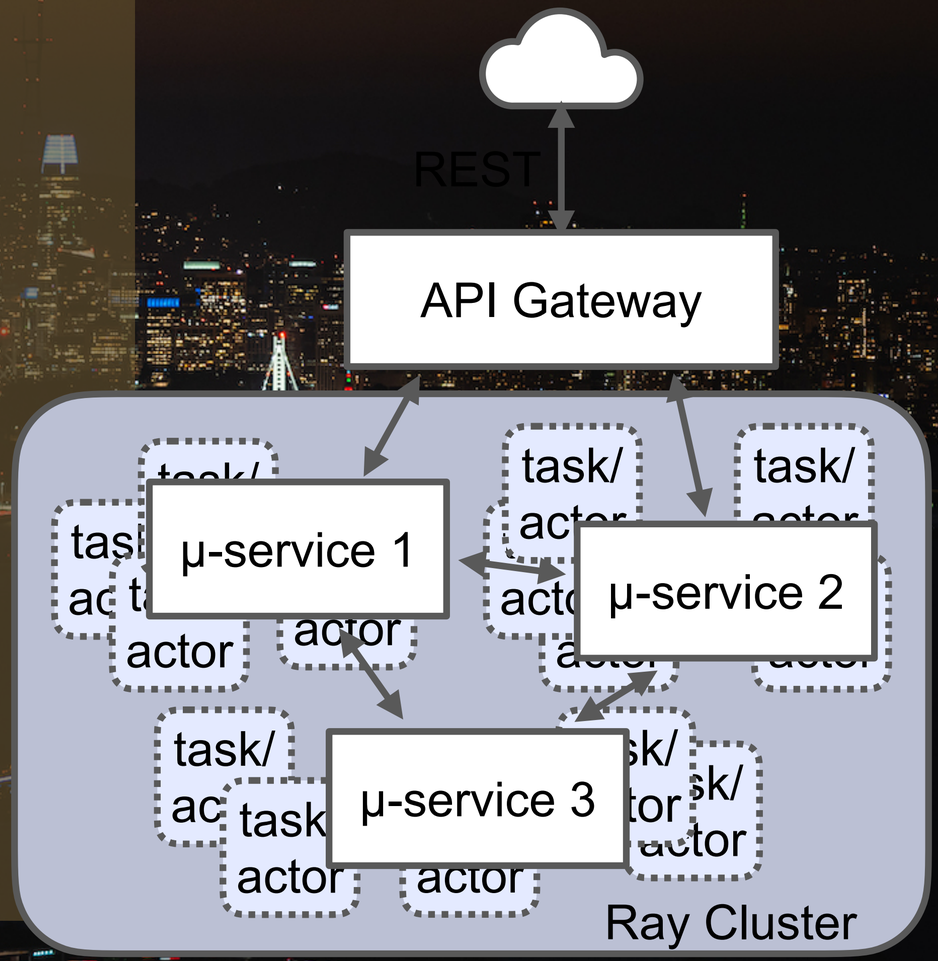
Microservices (again)

Need to Scale??

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



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



Apache Arrow

- Standardize in-memory data
  - Storage, schema, ...
  - Reuse across apps & libs

arrow.apache.org


polyglotprogramming.com/talks
dean@deanwampler.com
@deanwampler