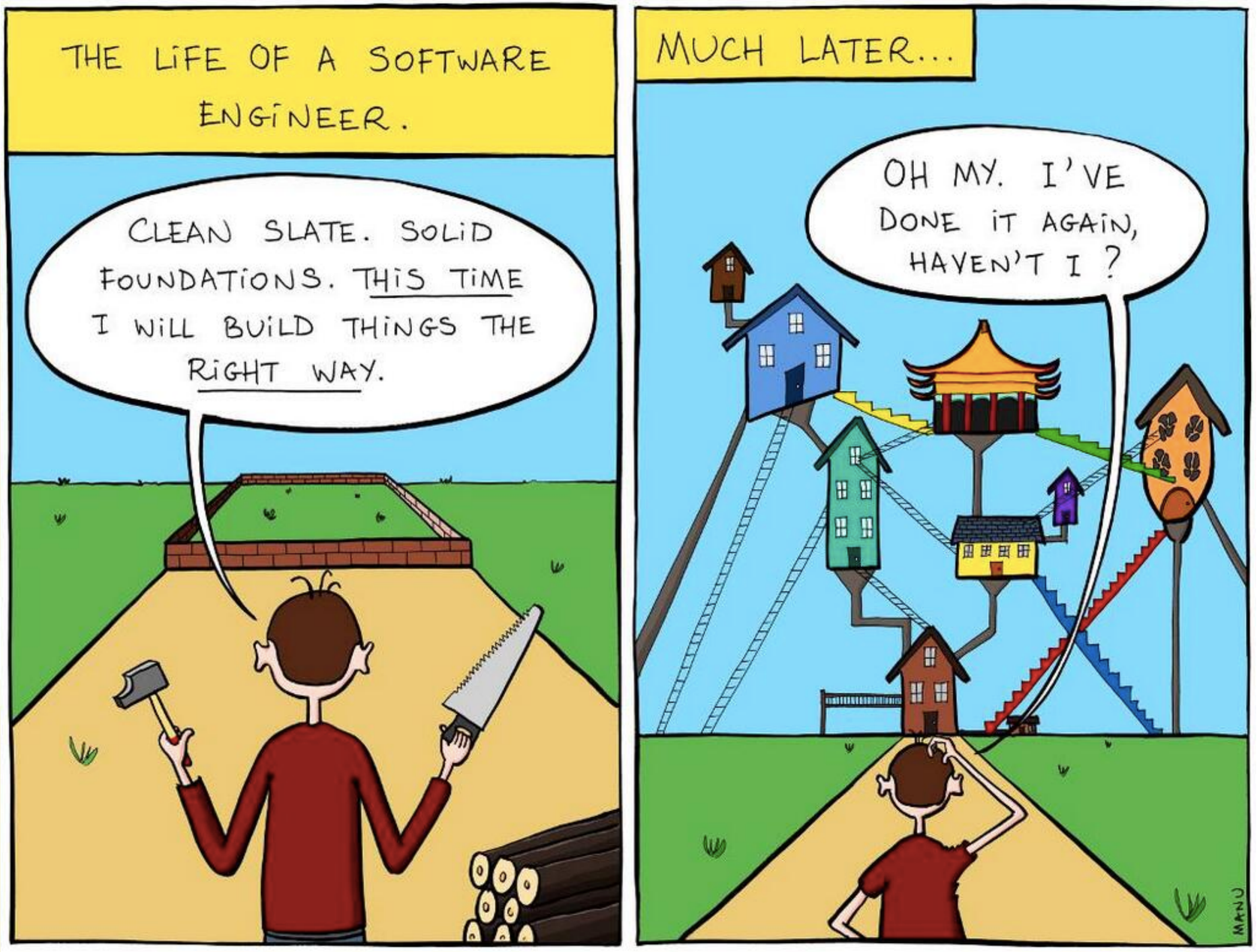
Microservices Pros and Cons

Rebecca Parsons @rebeccaparsons
Sofia Tania @developingviews

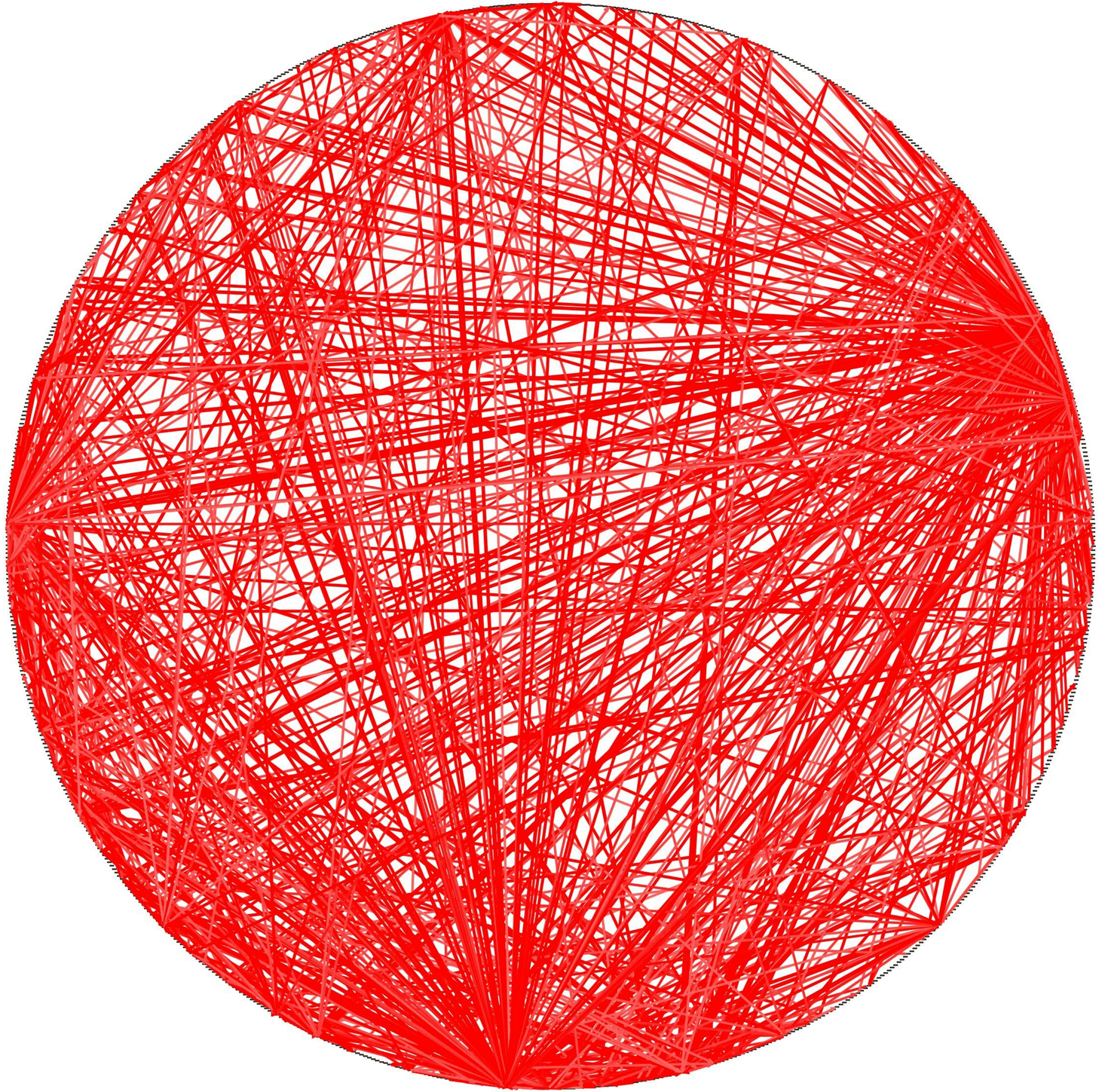
Magic and silver bullets

WTF?



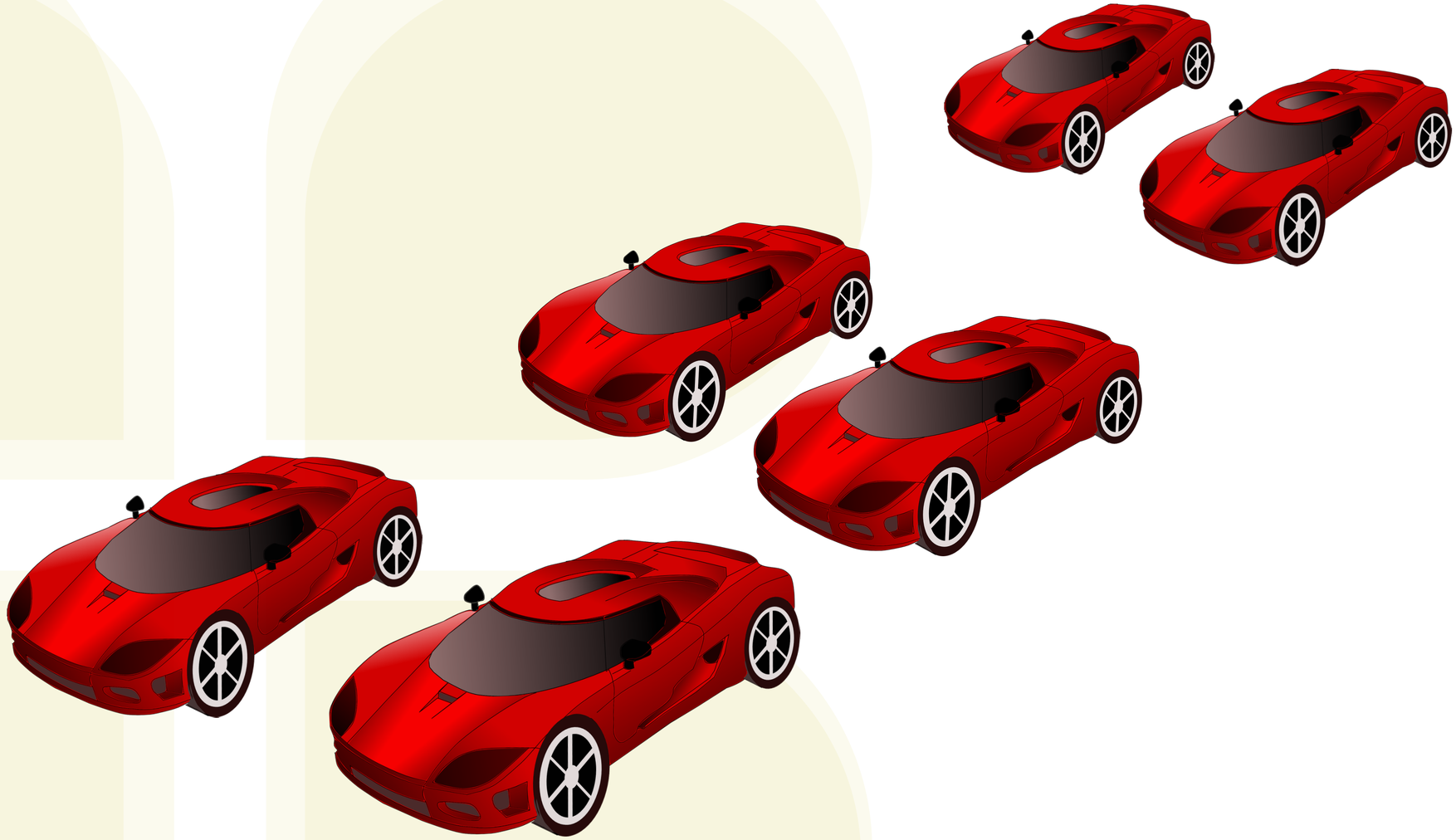
[Source: Manu Cornet – Bonker's World]

But first…

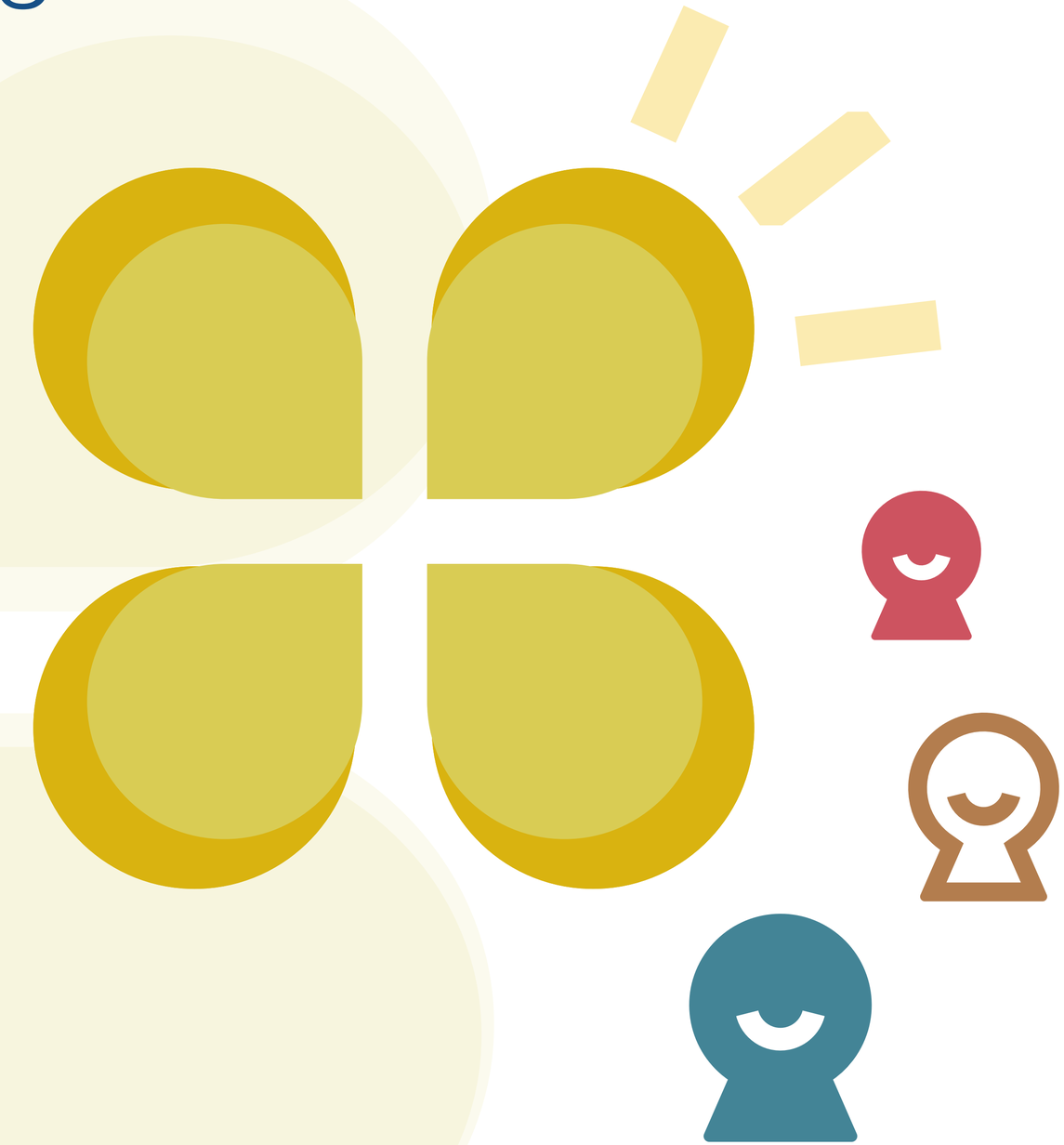


Big and slow

Small and fast



Along came microservices

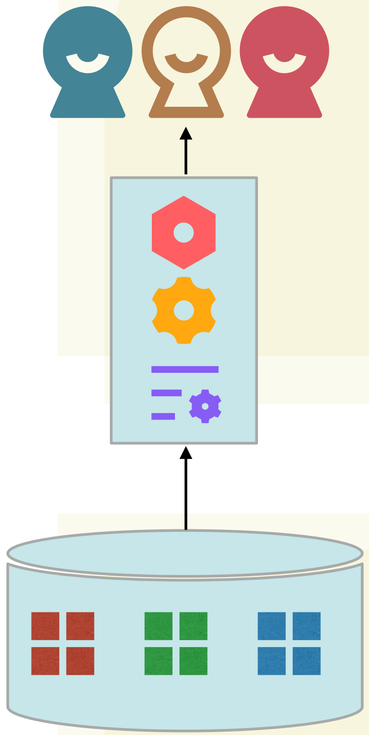


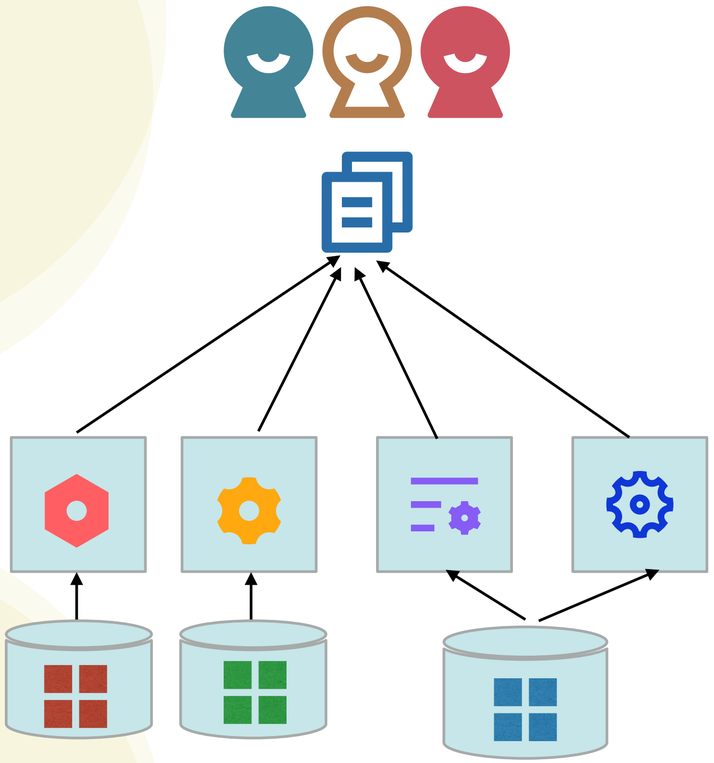
Microservices
WTF!

The right tool for the job

Microservices
are awesome!

Decentralized Governance

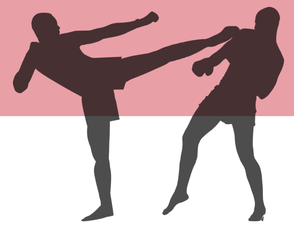




Microservices
WTF!

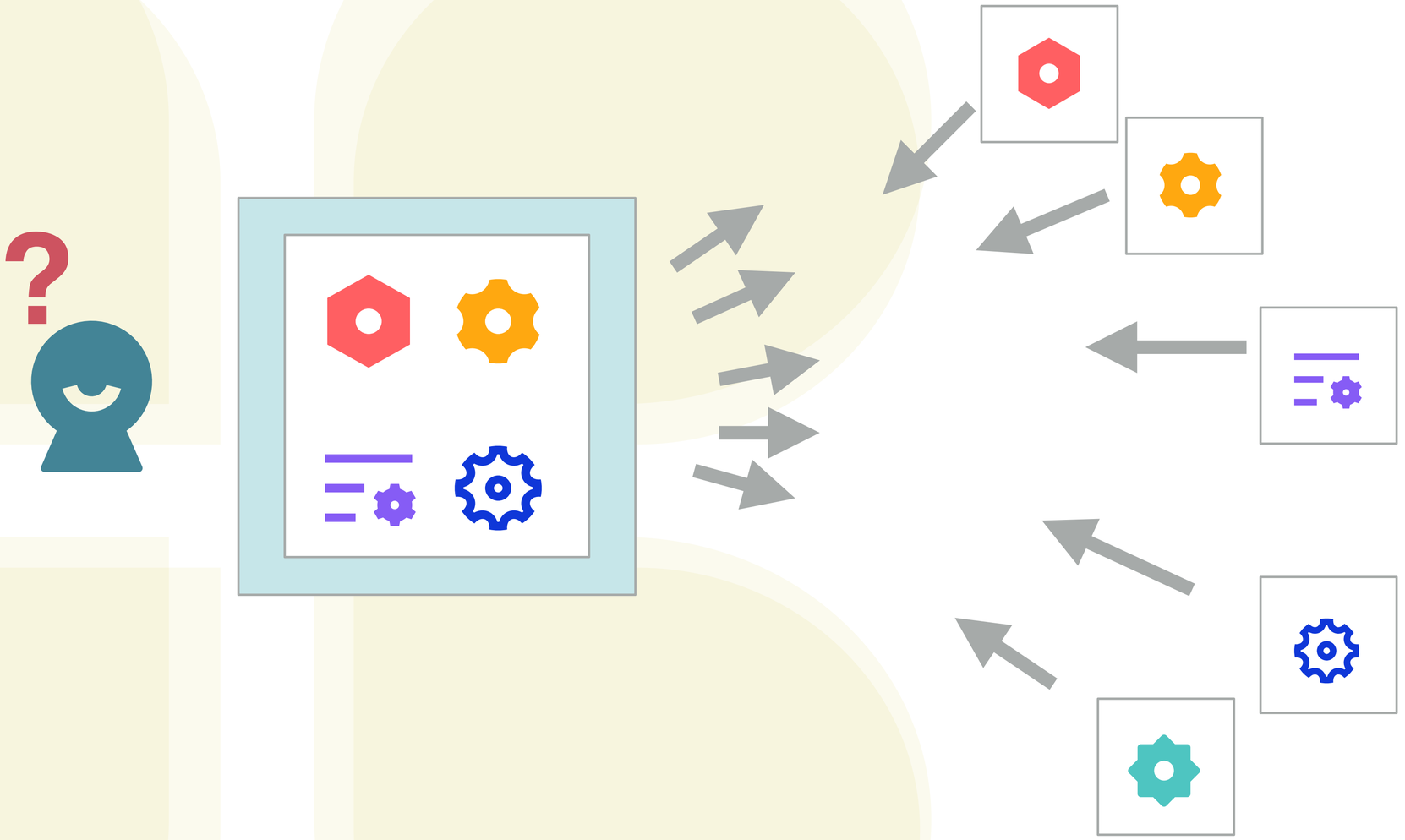
The right tool for the job

Cognitive complexity



Microservices
are awesome!

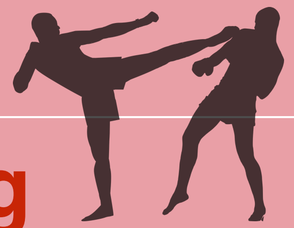
Cognitive Complexity



Microservices
WTF!

The right tool for the job

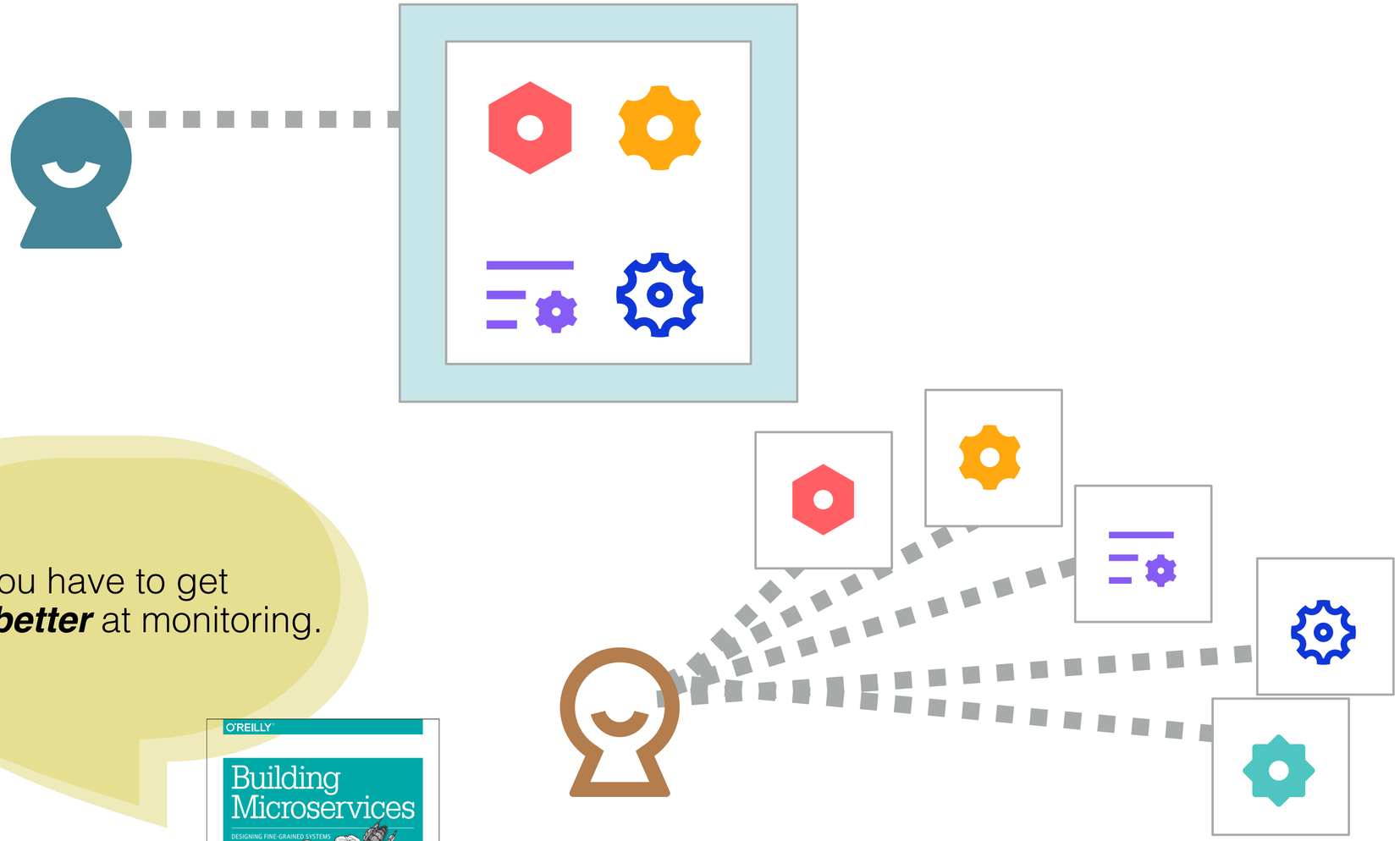
Cognitive complexity



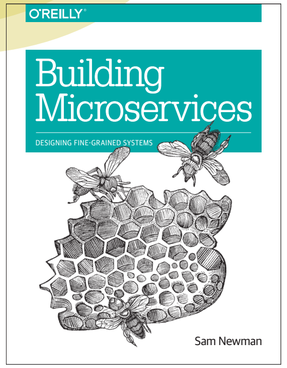
Observability & testing

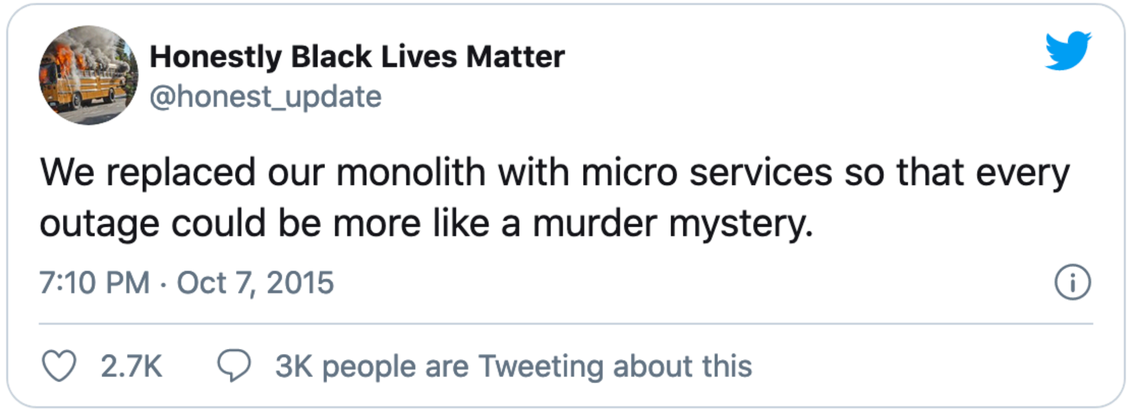
Microservices
are awesome!

Observability

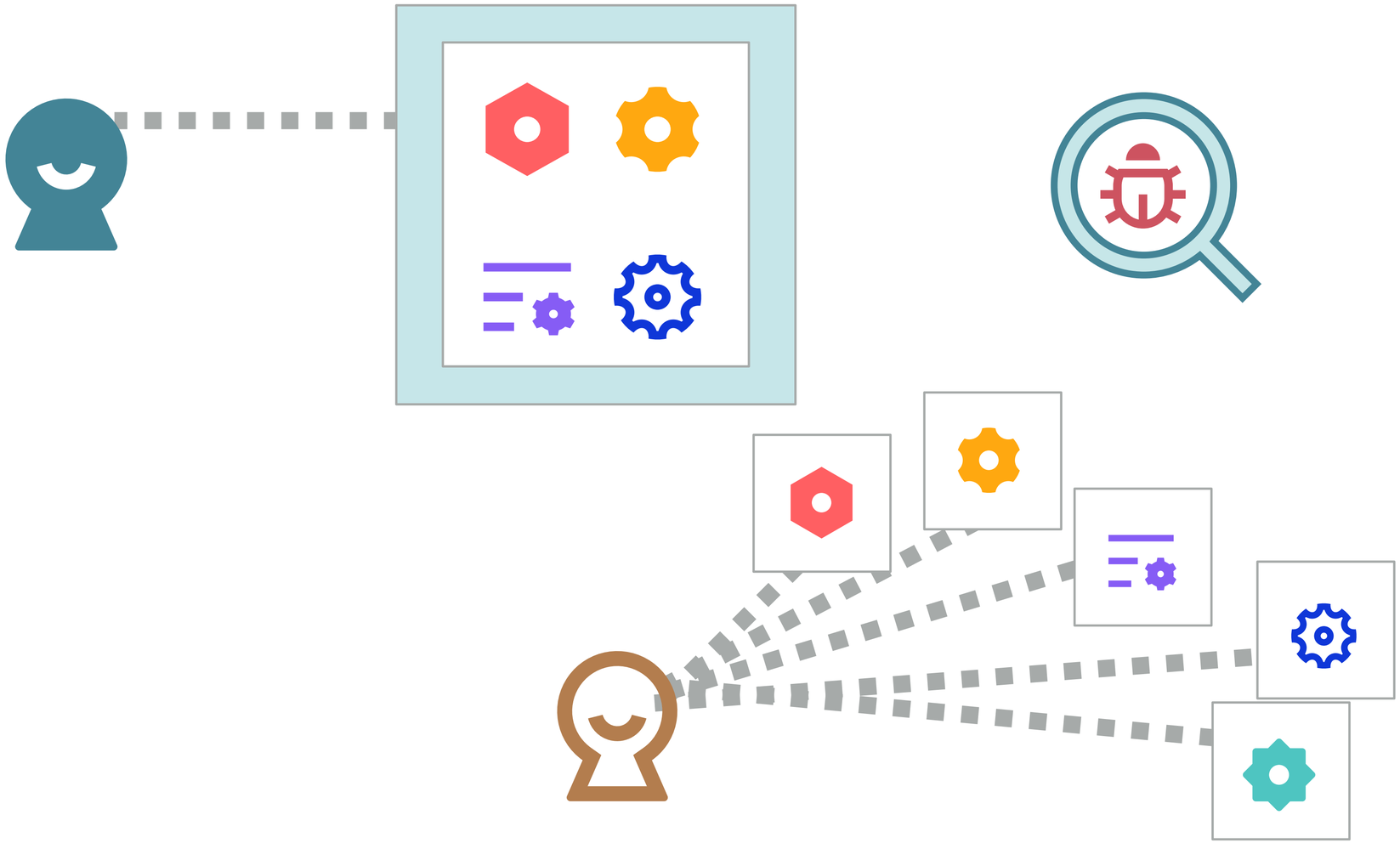


You have to get
much better at monitoring.





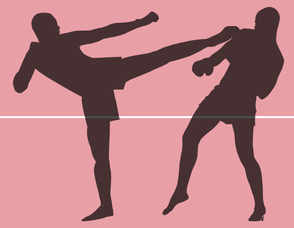
Testing



Microservices WTF!

The right tool for the job

Cognitive complexity



Monitoring & testing

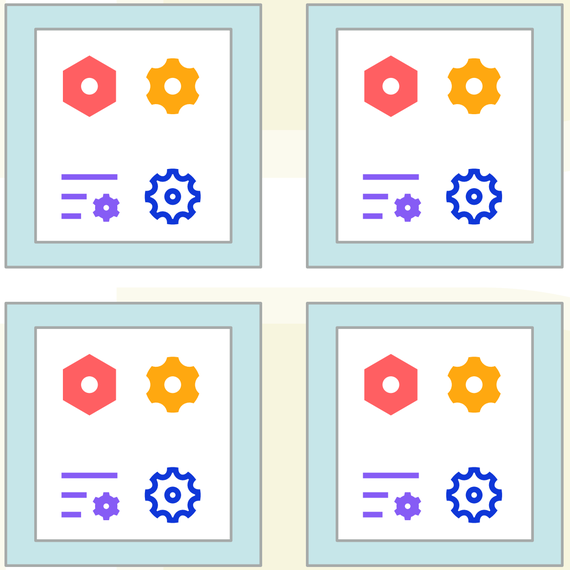
Scale & resilience

Microservices are awesome!

Scaling

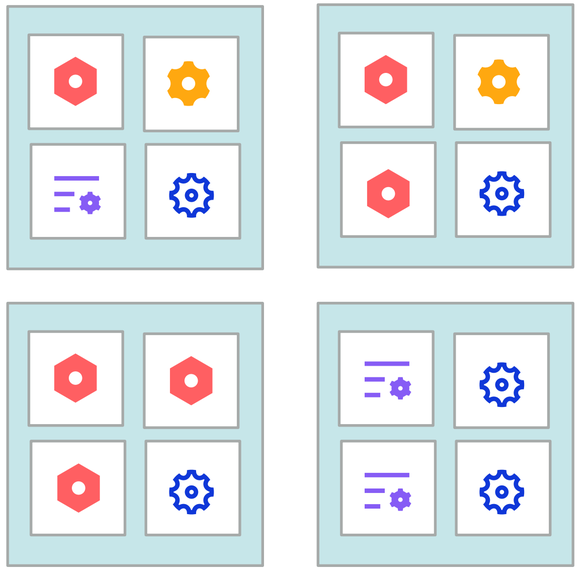
A monolithic application puts all its functionality into a single process…

… and scales by replicating the monolith on multiple servers

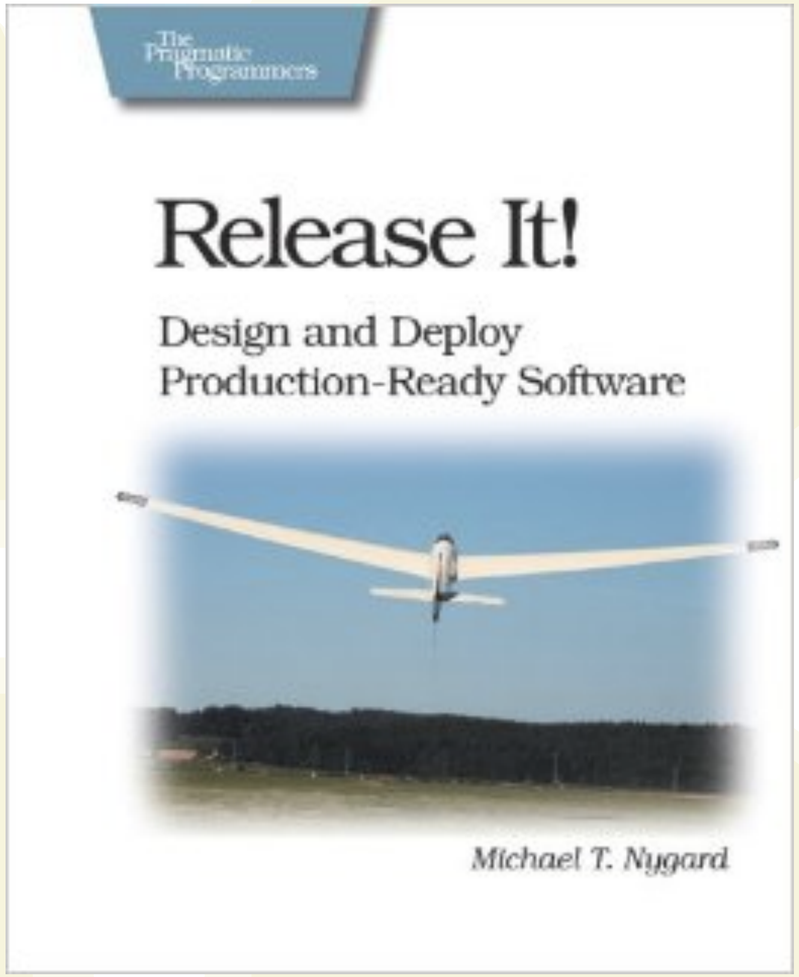


A microservices architecture puts each element of functionality into a separate service…

… and scales by replicating the across servers, replicating as needed



Design for failure



if you fail to design for production your life will be filled with "excitement"

- Michael Nygard

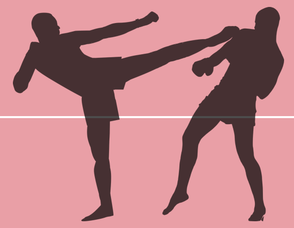
Fallacies of Distributed Computing

The network is reliable;
Latency is zero;
Bandwidth is infinite;
The network is secure;
Topology doesn't change;
There is one administrator;
Transport cost is zero;
The network is homogeneous.

Microservices WTF!

The right tool for the job

Cognitive complexity



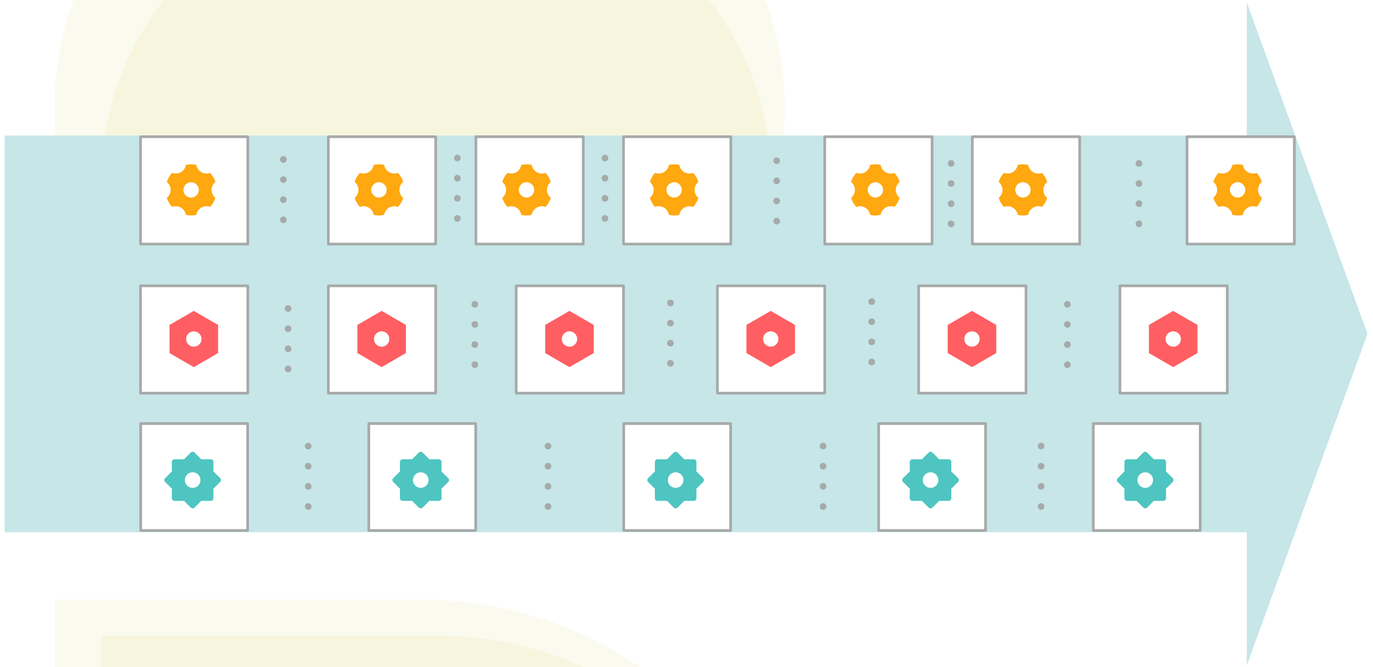
Monitoring & testing

Scale & resilience

Independent deployment

Microservices are awesome!

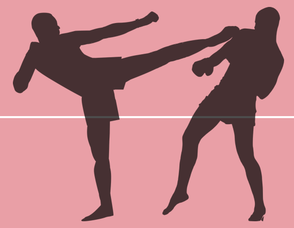
Independent Releases



Microservices WTF!

The right tool for the job

Cognitive complexity



Monitoring & testing

Scale & resilience

Independent deployment

Configuration

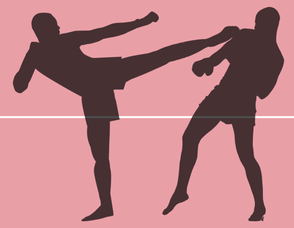
Microservices are awesome!

Release Complexity

Microservices WTF!

The right tool for the job

Cognitive complexity



Monitoring & testing

Scale & resilience

Independent deployment

Configuration

Microservices are awesome!

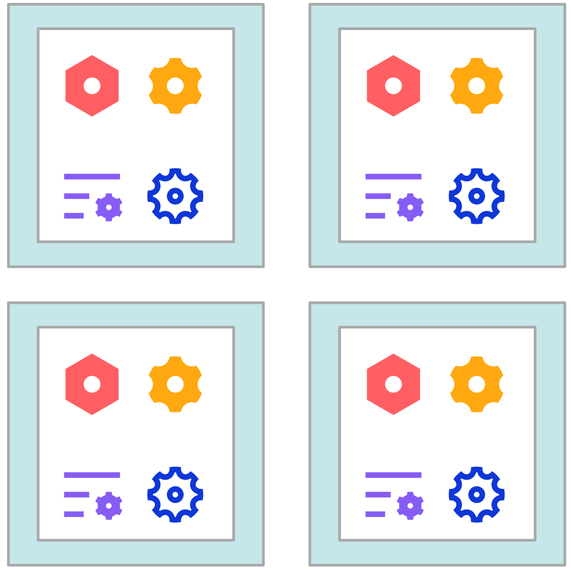
Replaceable

Replaceable Services

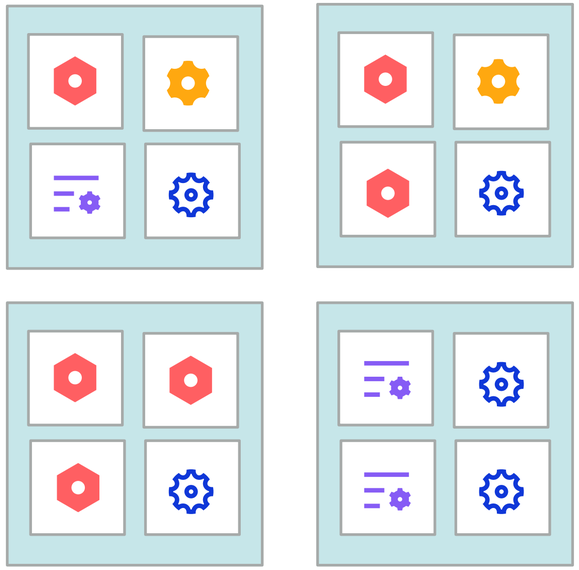
A monolithic application puts all its functionality into a single process…

A microservices architecture puts each element of functionality into a separate service…

… and scales by replicating the monolith on multiple servers



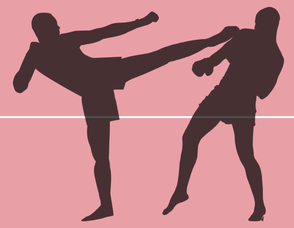
… and scales by replicating the across servers, replicating as needed



Microservices WTF!

The right tool for the job

Cognitive complexity



Monitoring & testing

Scale & resilience

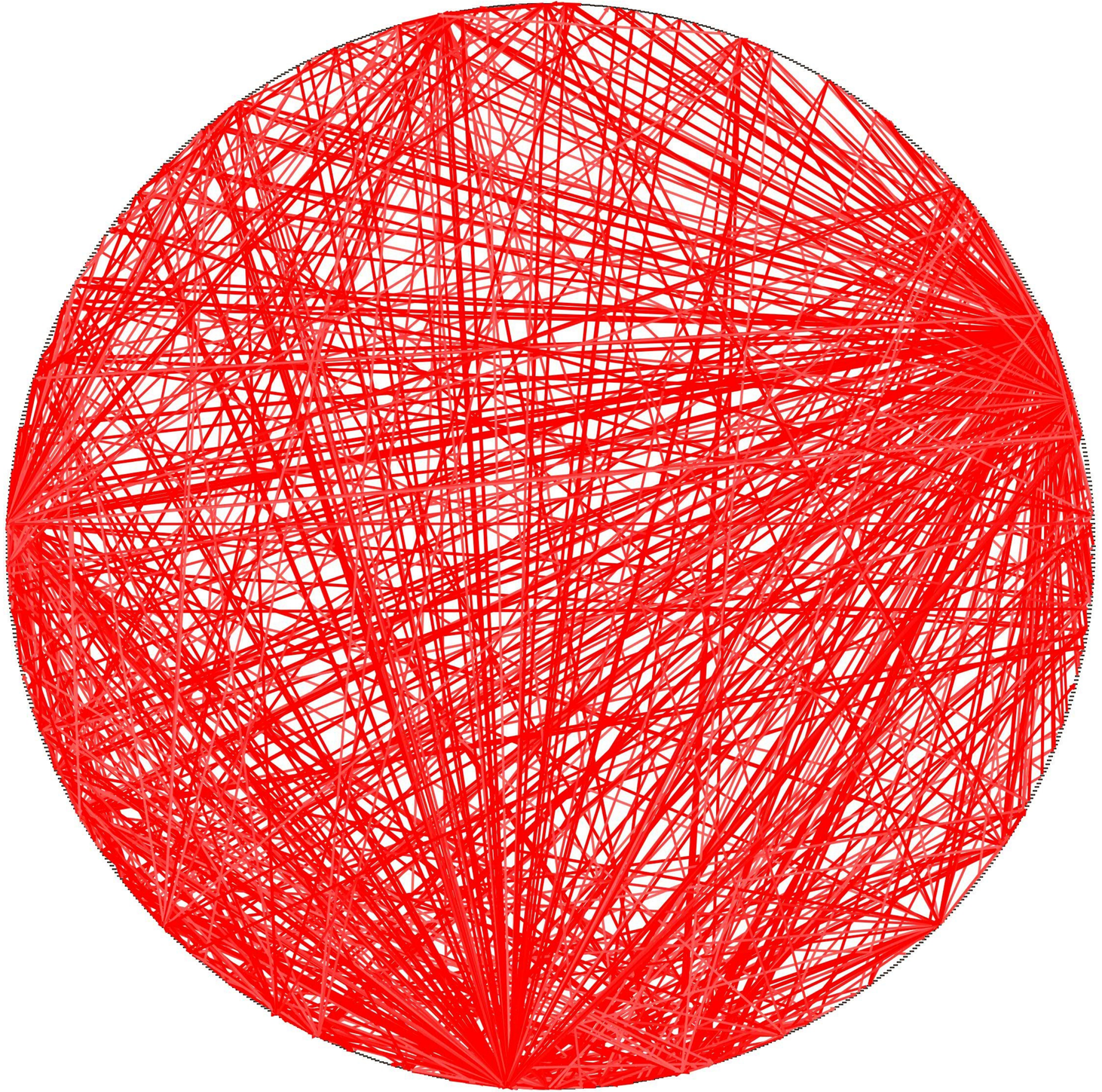
Independent deployment

Configuration

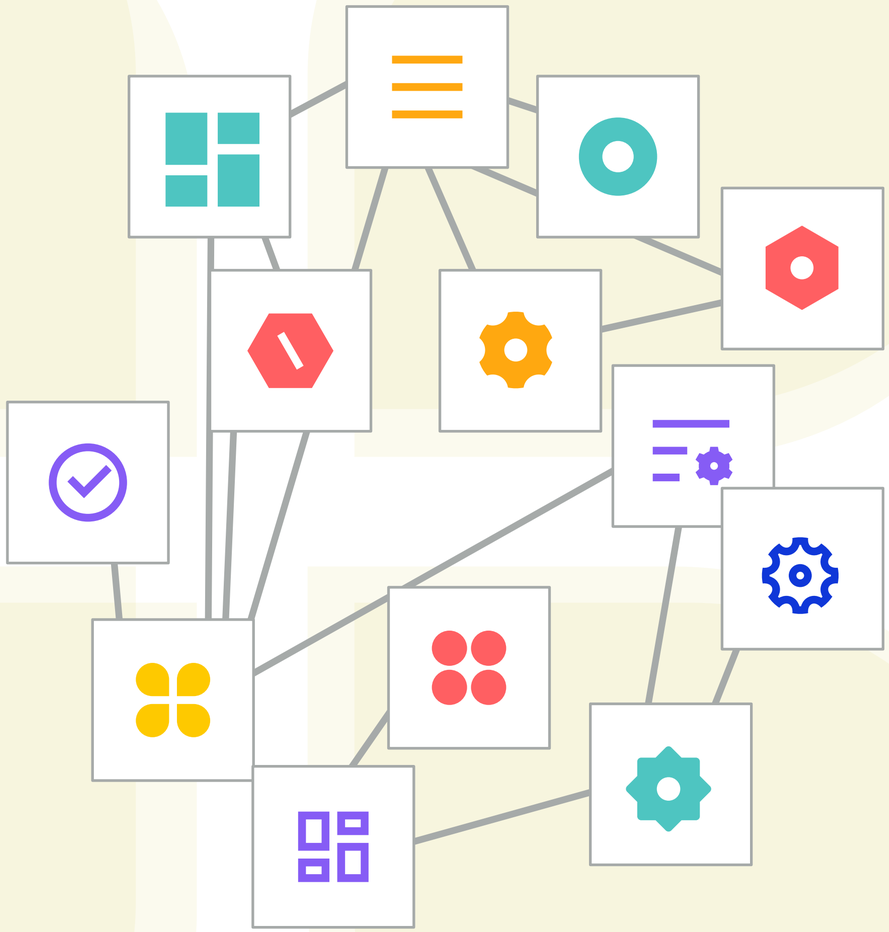
Integration

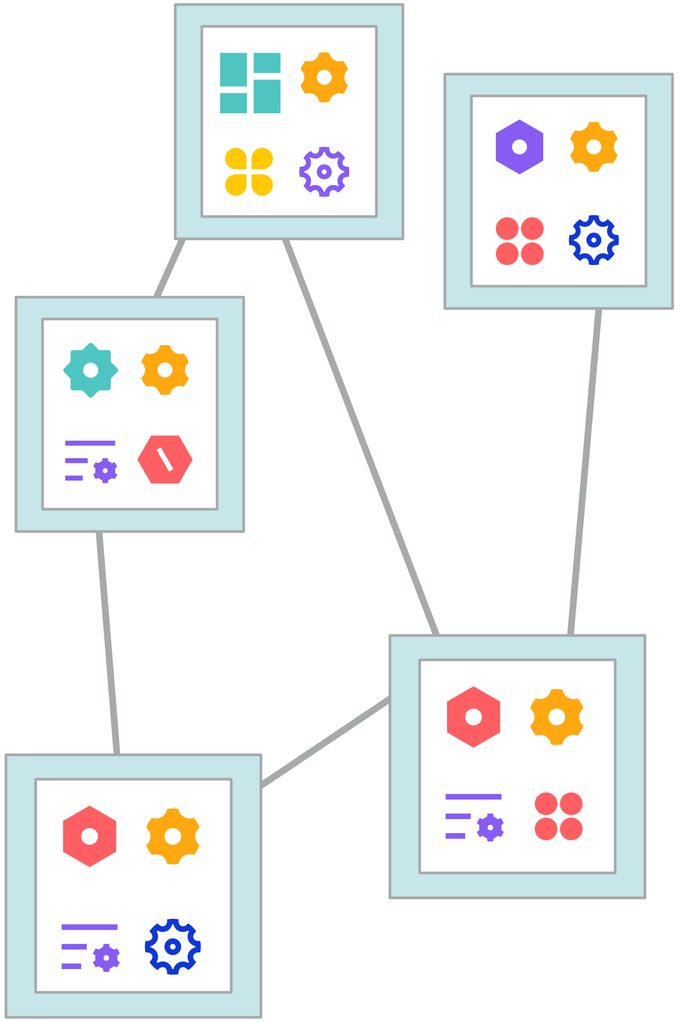
Microservices are awesome!

Replaceable

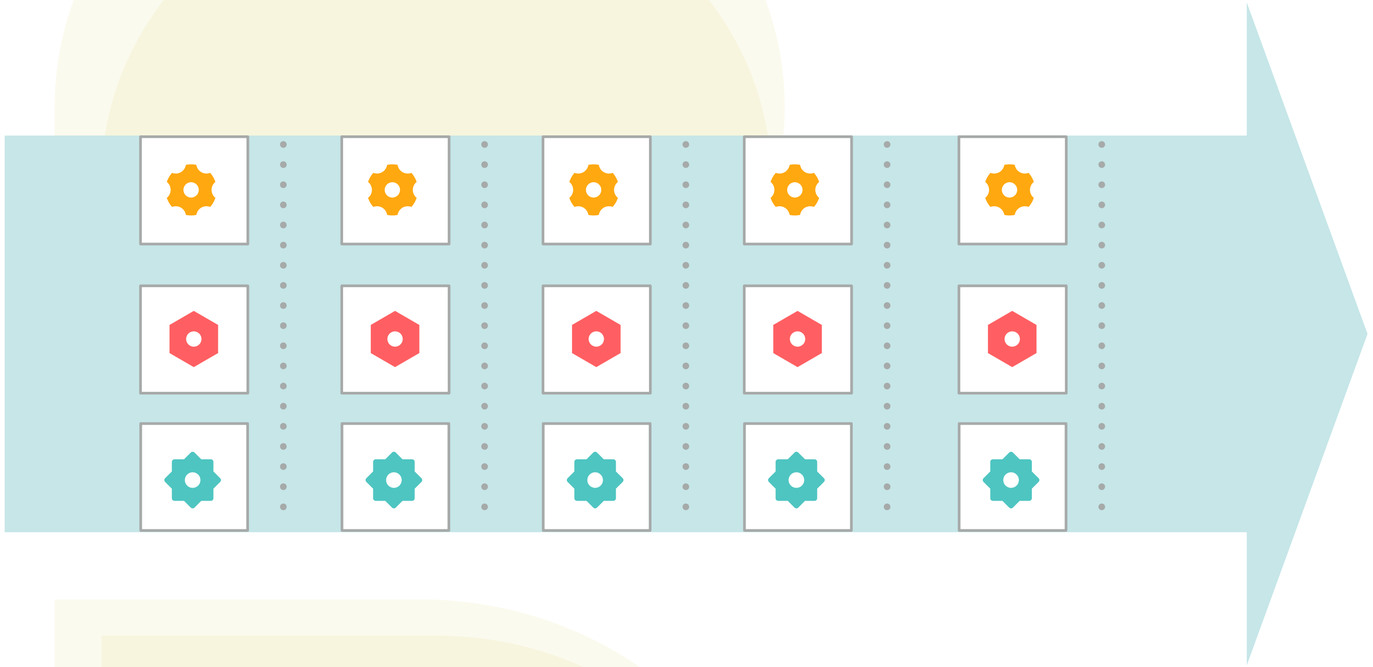


From *application* to *integration* architecture





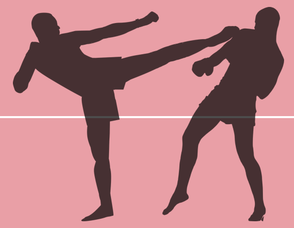
Independent Releases?



Microservices WTF!

The right tool for the job

Cognitive complexity



Monitoring & testing

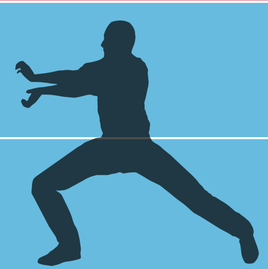
Scale & resilience

Independent deployment

Configuration

Integration

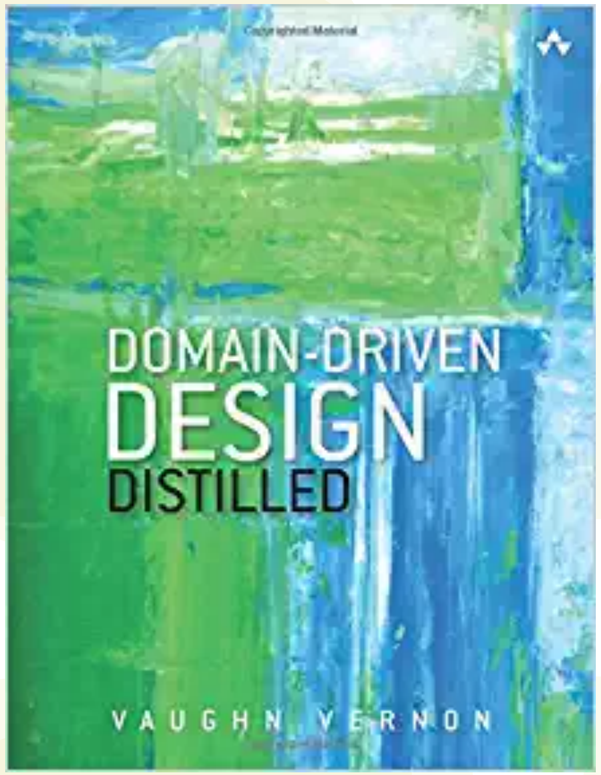
Microservices are awesome!

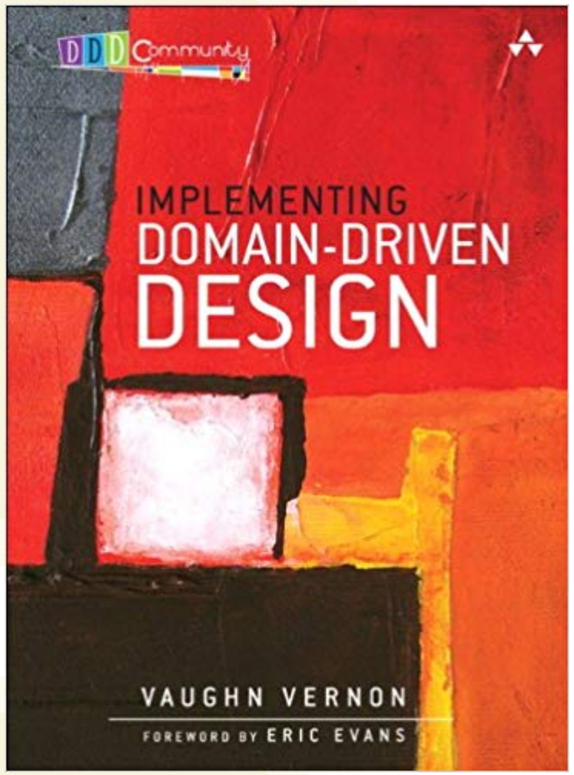


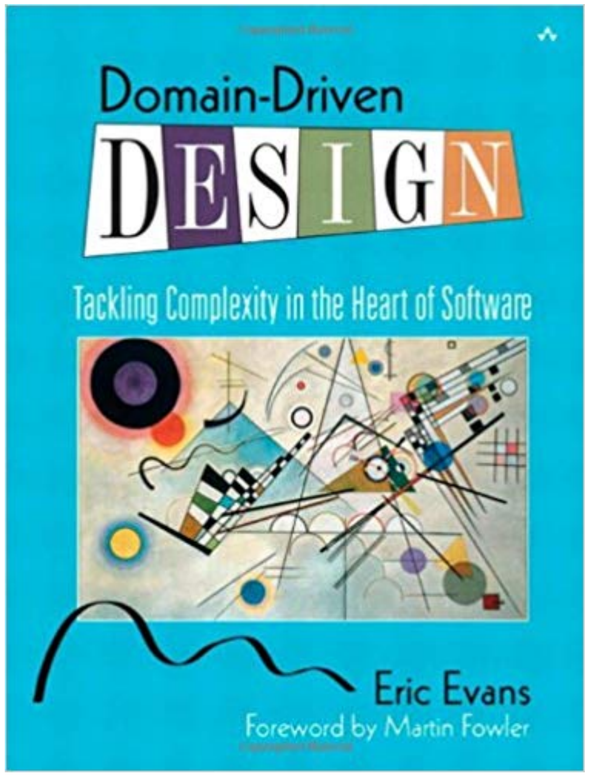
Replaceable

Domain-Driven Design

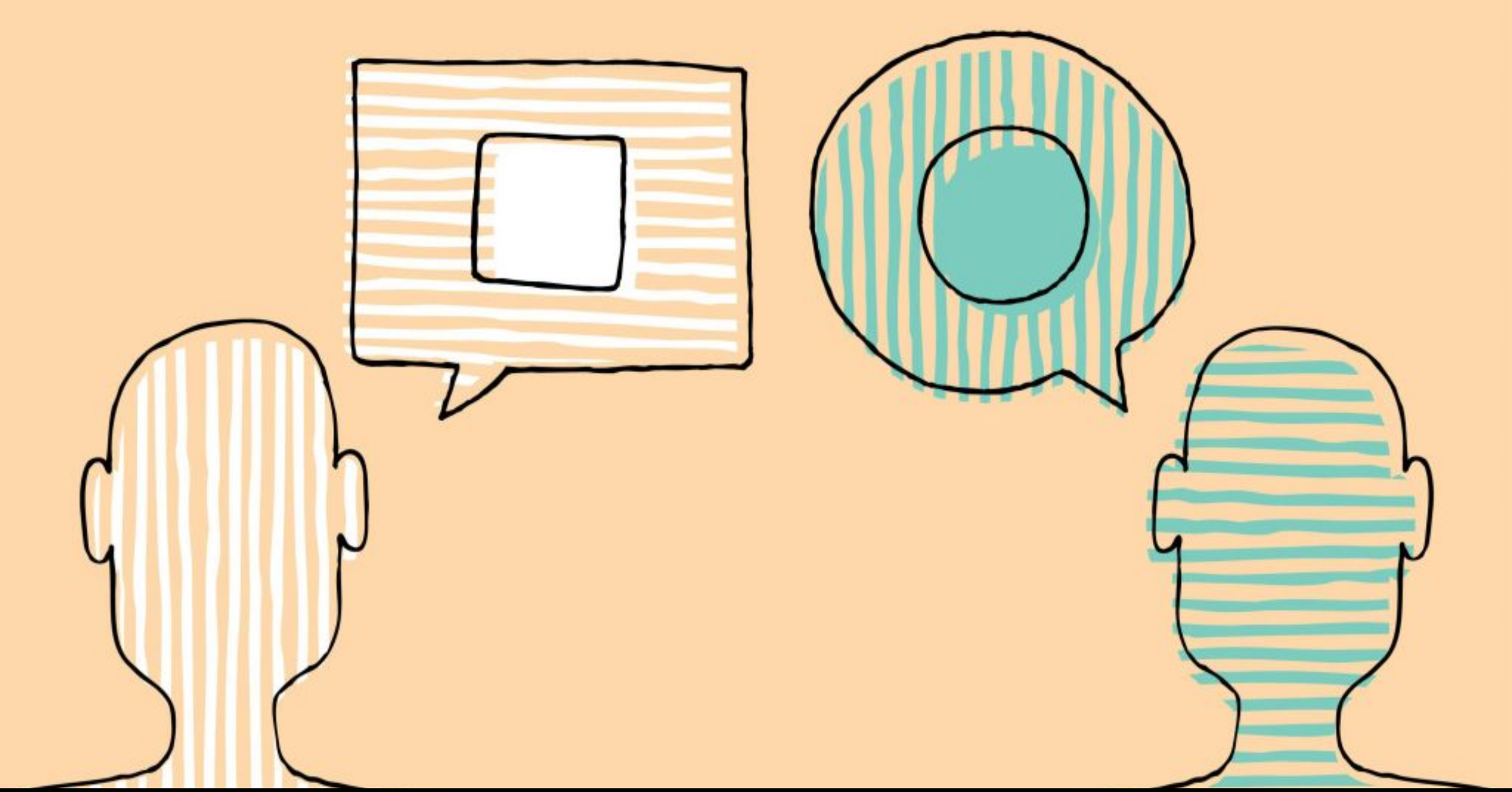
Additional Resources







Design Pattern to represent the world



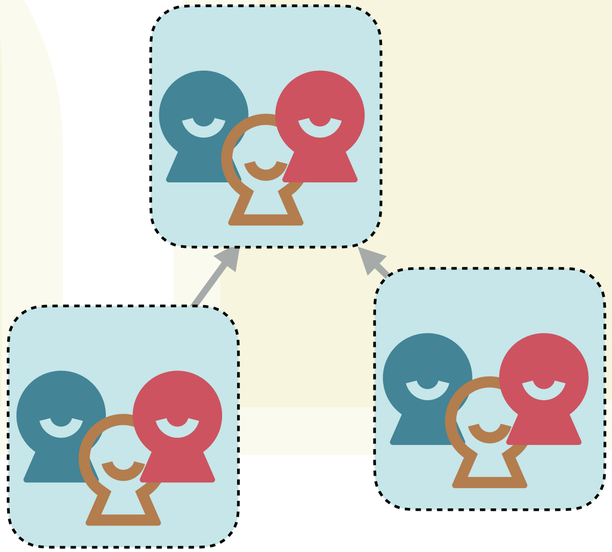
Collaboration between business and tech

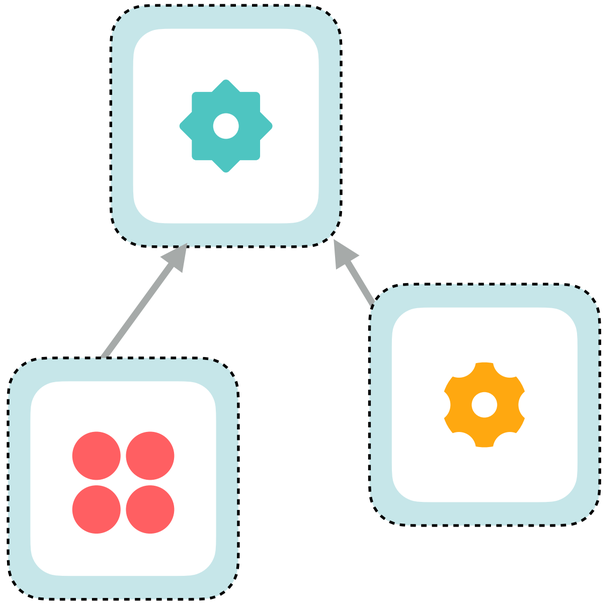
Domain-Driven Design (DDD)

1. Bounded Contexts
2. Ubiquitous Language
3. Domains (and Subdomains)

Boundaries are explicit

Domain-Driven Teams

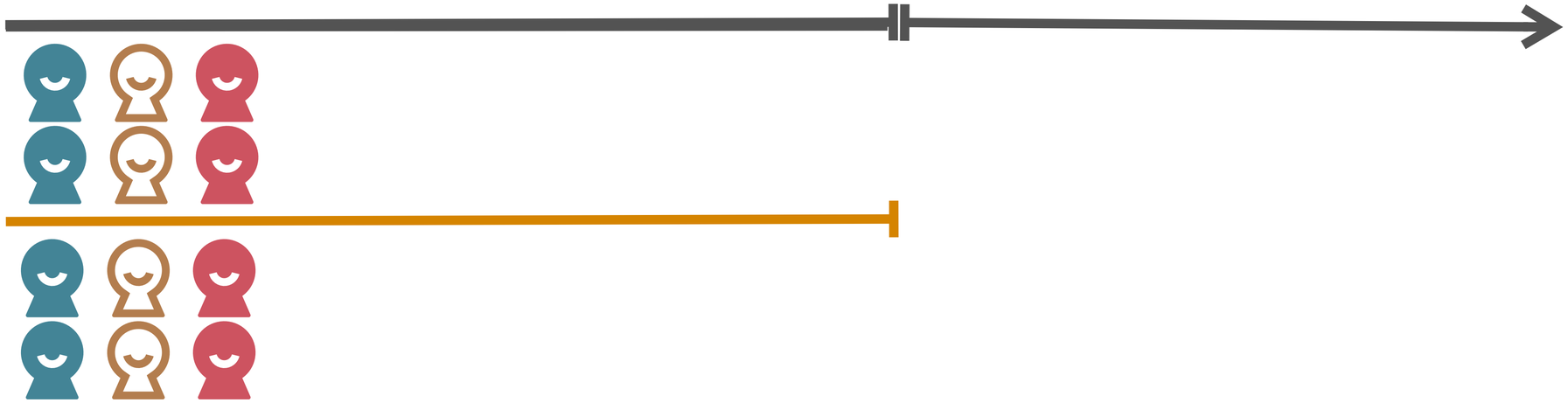




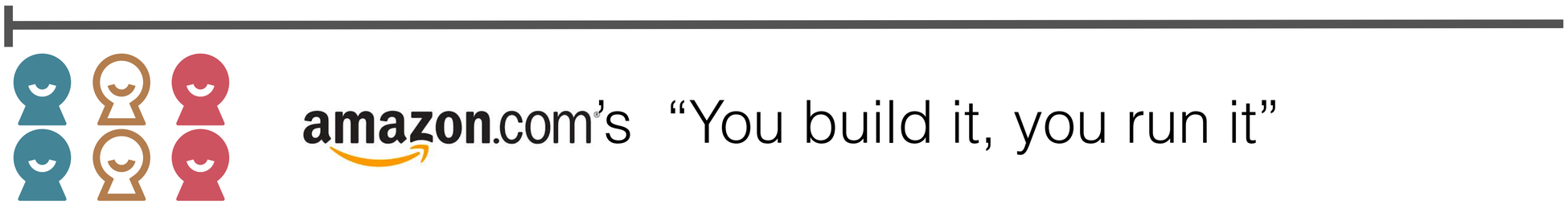
Build teams that look like the architecture you want
(and it will follow).

Products thinking organization

projects:



products:



amazon.com's "You build it, you run it"

Maturity

Thank you!

Rebecca Parsons @rebeccaparsons
Sofia Tania @developingviews