

AUTO SCOUT 24

Enterprise Fast Lane

Transforming to Microservices in the Cloud

GOTO Amsterdam | 14.06.2017 | Christian Deger | @cdeger



Christian Deger
Chief Architect
christian.deger@scout24.com
@cdeger

2,4 Million Vehicles



Microservices in the cloud adoption?







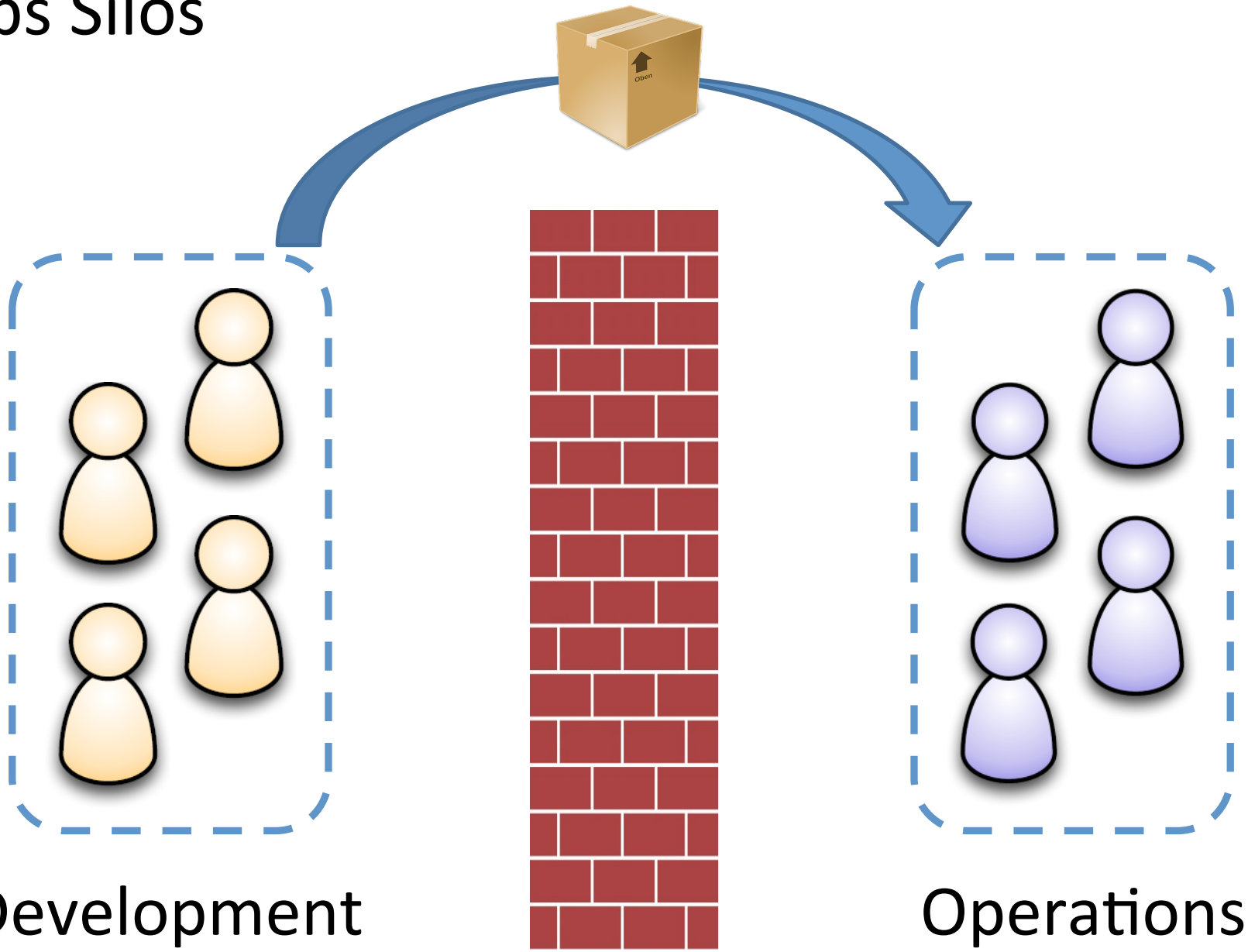


2000 Servers

2 Data Centers

MTBF optimized

Dev and Ops Silos

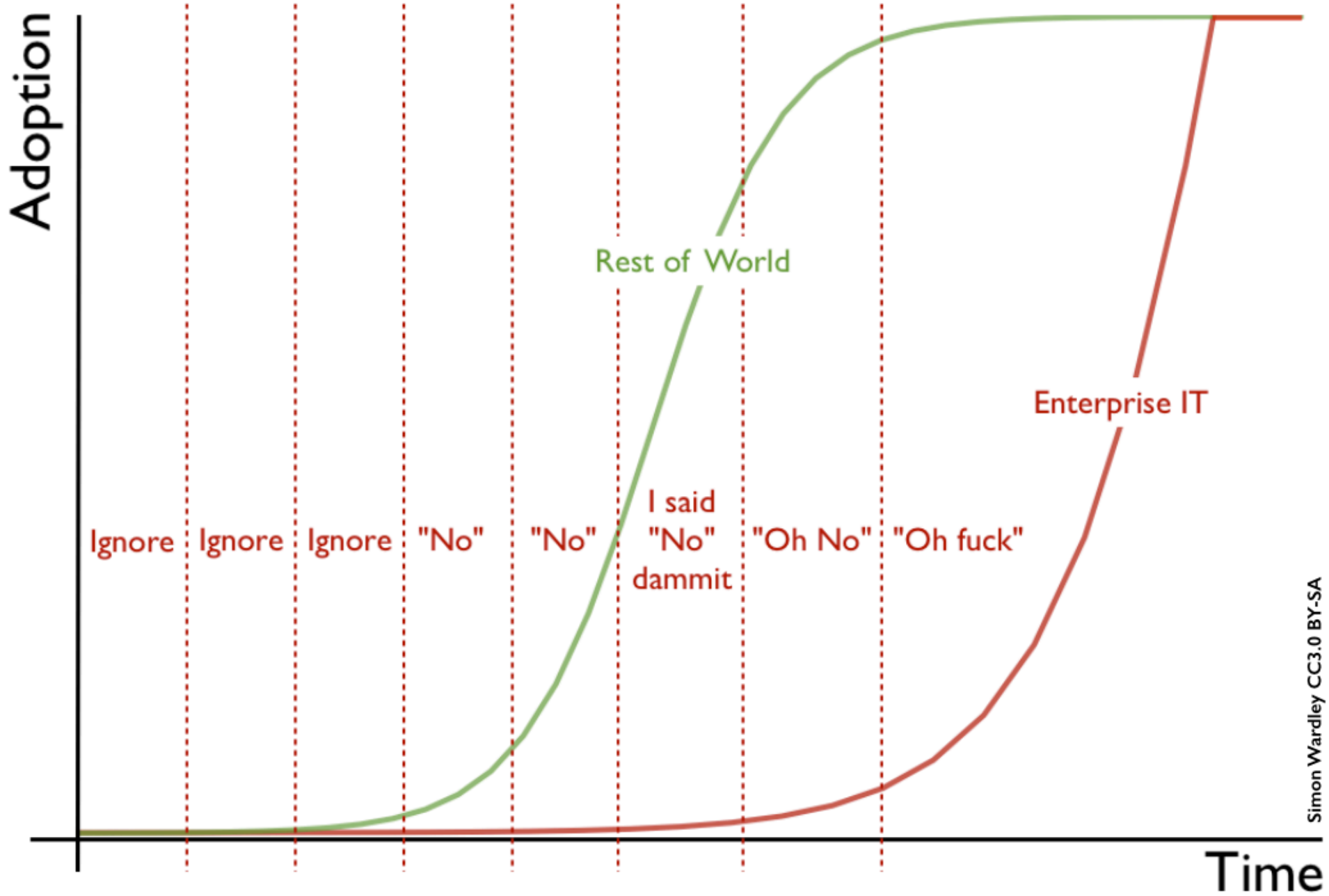


Development
“Change”

Operations
“Stability”


WORKED FINE IN DEV
OPS PROBLEM NOW
memegenerator.net

Enterprise IT Adoption Cycle




New
CEO

Do you attract



Talent?

What does a

21st Century

tech company

look like?




NETFLIX


LinkedIn


airbnb


SOUNDCLOUD


Spotify®

Great Design

Universally Connected

Mobile First

Instant Business Value

Massive Data Insight

Highly Available


Hmm, we are
good, but not great


Reboot
everything

.NET / Windows to JVM / Linux

Monolith to Microservices

Data center to AWS

Devs + Ops to Collaboration culture

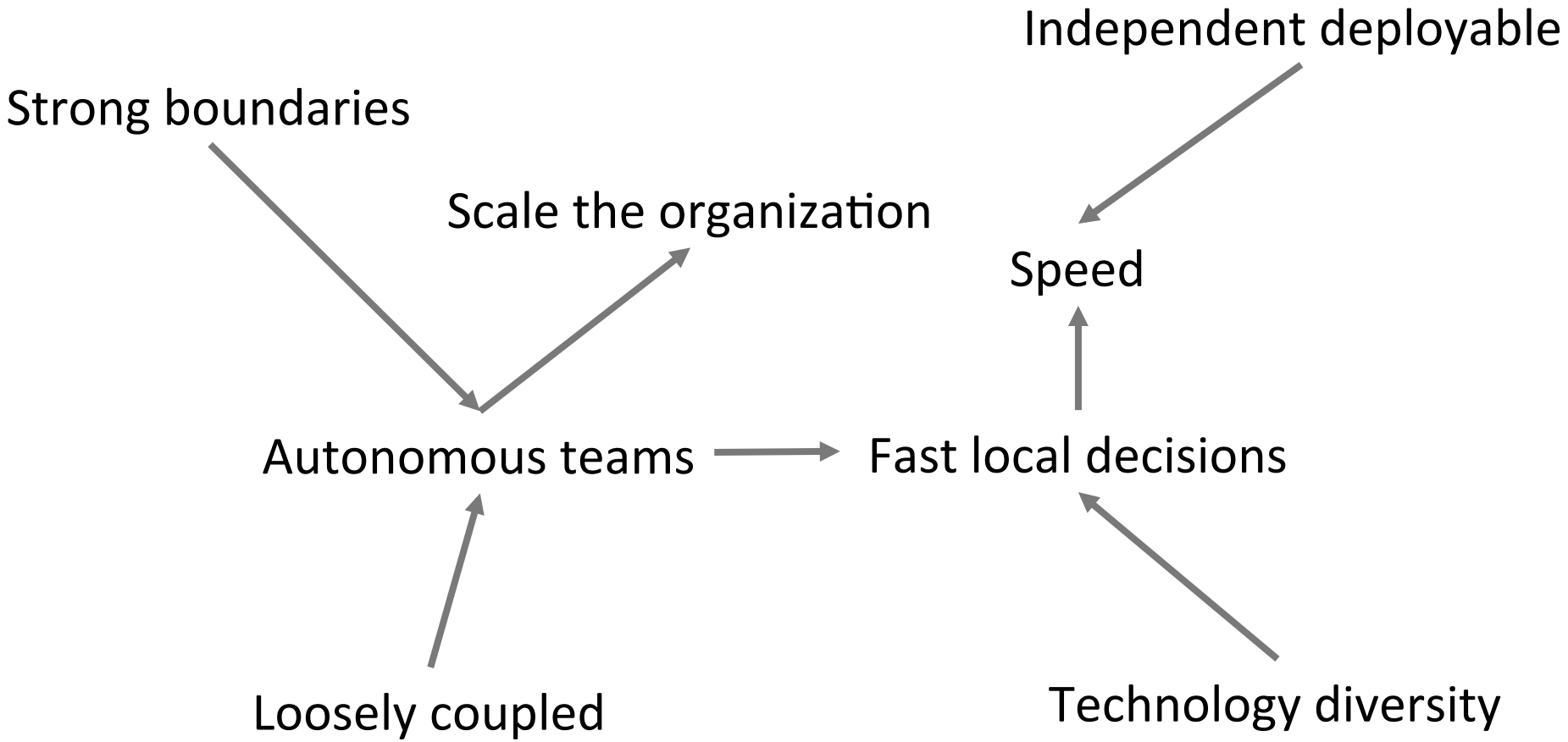
Involve product people



Microservices in the cloud adoption?



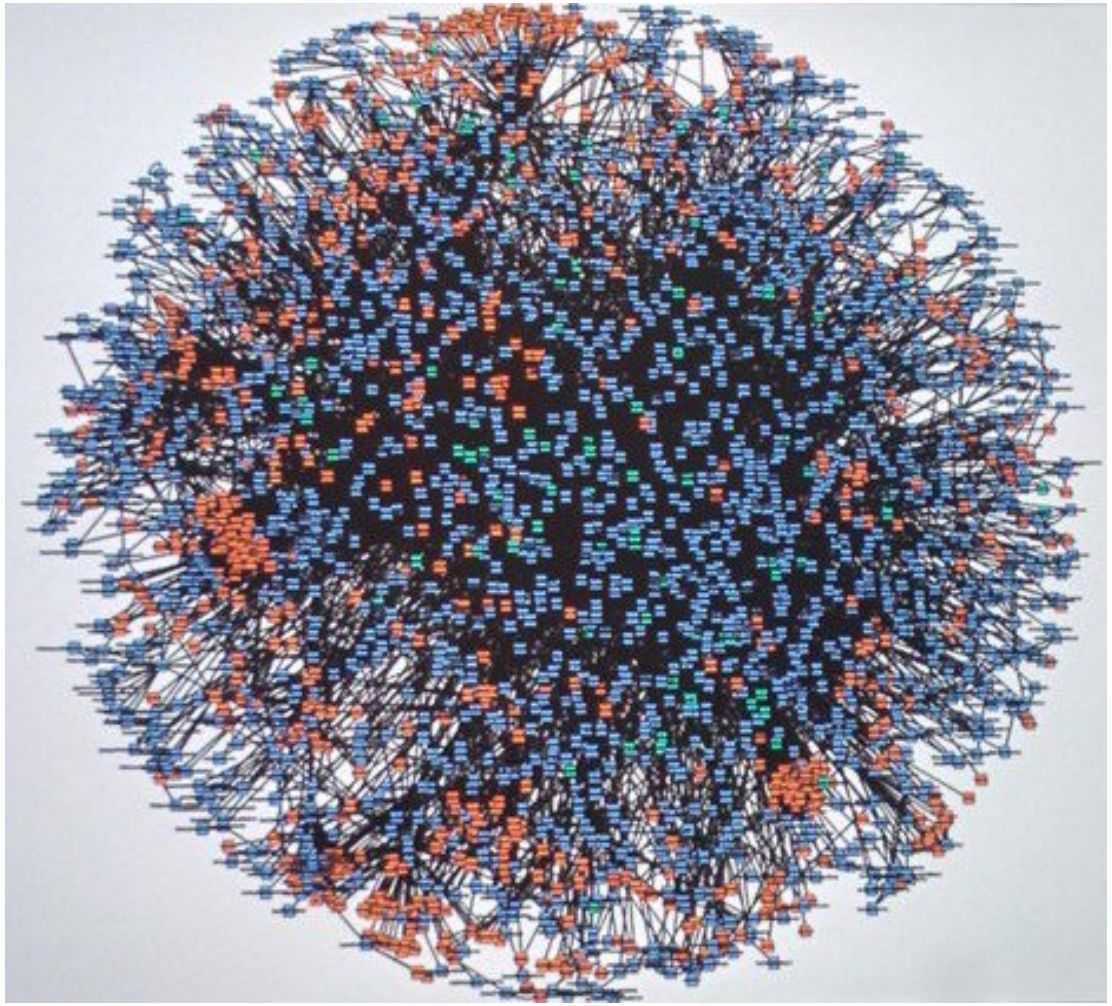
Why Microservices?



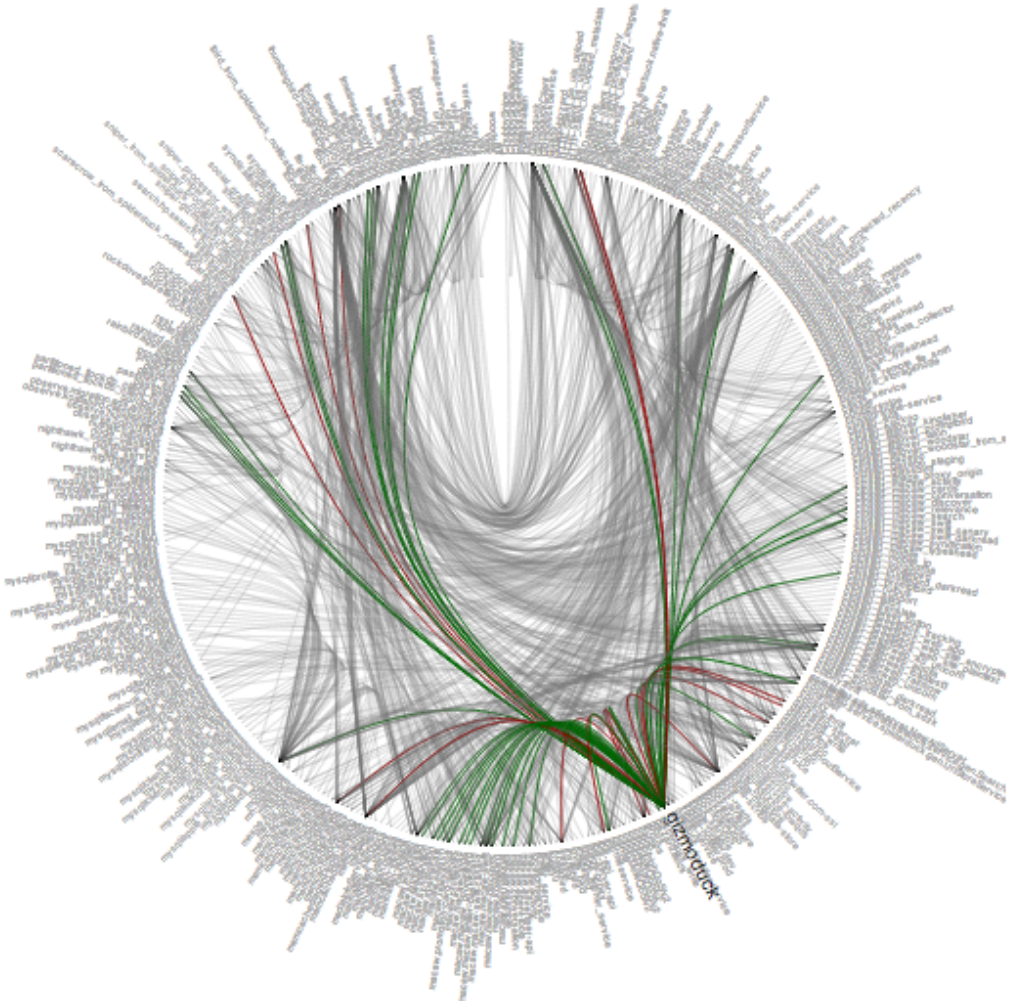
Loosely coupled service oriented architecture with bounded contexts.
—Adrian Cockcroft

Microservices are small, autonomous services that work together.
—Sam Newman

“Death Star” Diagrams

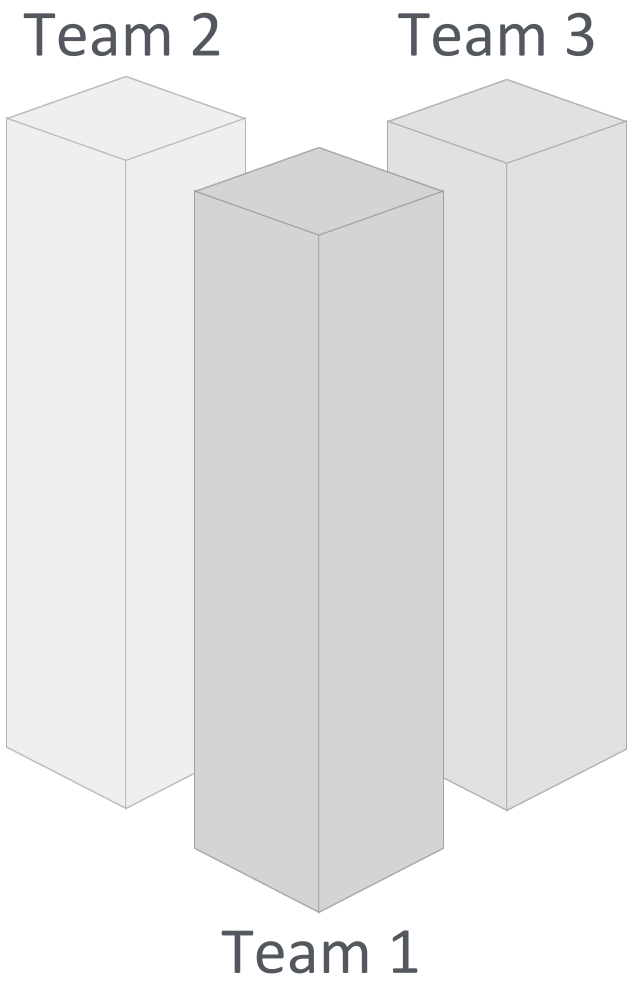


Amazon 2008

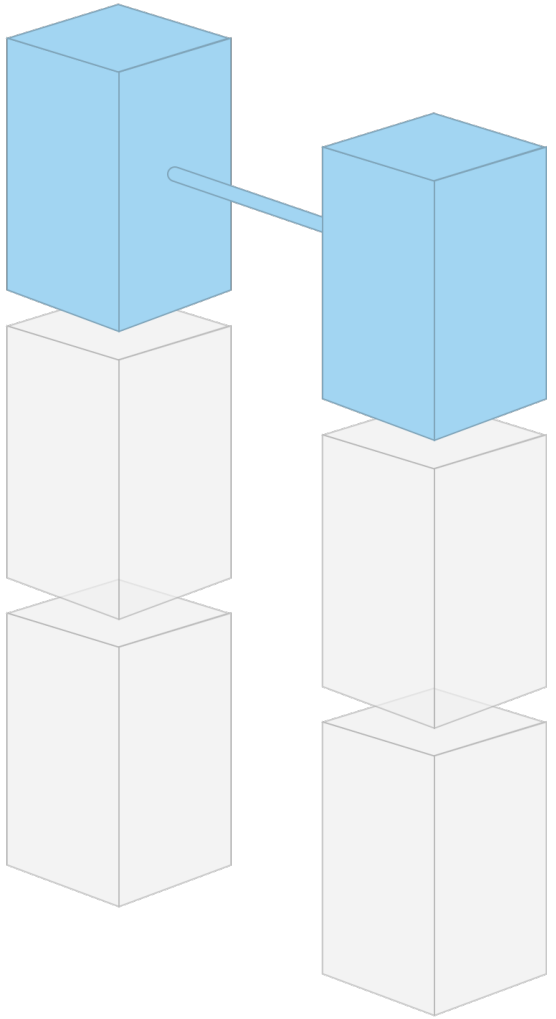


Twitter 2013

Self-Contained Systems = Microservices Flavor



One business capability is owned, built and run as an SCS by one team.



Self-Contained System are vertical slices integrated at the UI.

same

direction



STRATEGIC GOALS



Goals of the business side

REDUCE TIME TO MARKET

Establish fast feedback loops to learn, validate and improve. Remove friction, hand-offs and undifferentiated work.

SUPPORT DATA-DRIVEN DECISIONS

Provide relevant metrics and data for user and market insights. Validate hypothesis for problems worth solving.

MOBILE FIRST

Start small and use device capabilities.

BEST TALENT

Autonomy, Purpose and Mastery: We know why we do things, we decide how to approach them and deliberately practice our skills.

COST EFFICIENCY

Run your segment in the right balance of cost and value.

ONE SCOUT IT

Foster collaboration. Harmonize and standardize tools. Pull common capabilities into decoupled platform services.

ARCHITECTURAL PRINCIPLES



High-Level Principles

ORGANIZED AROUND BUSINESS CAPABILITIES

Build teams around products not projects. Follow the domain and respect bounded contexts. Make boundaries explicit. Inverse Conway Maneuver.

ELIMINATE ACCIDENTAL COMPLEXITY

Strive to keep it simple. Don't over-engineer. Focus on necessary domain complexity.

LOOSELY COUPLED

By default avoid sharing and tight coupling. No integration database. Don't create the next monolith.

MACRO AND MICRO ARCHITECTURE

Clear separation. Autonomous micro services within the rules and constraints of the macro architecture.

SECURITY, COMPLIANCE AND DATA PRIVACY

Build with least privilege and data privacy in mind. Know your threat model. Limit blast radius.

AWS FIRST

Favor AWS platform service over managed service, over self-hosted OSS, over self built solutions.

DESIGN AND DELIVERY PRINCIPLES



Tactical measures

YOU BUILT IT, YOU RUN IT

The team is responsible for shaping, building, running and maintaining its products. Fast feedback from live and customers helps us to continuously improve.

CROSS-FUNCTIONAL TEAMS

Engineers from all backgrounds work together in collaborative teams as engineers and share responsibilities. No silos.

AUTONOMOUS TEAMS

Make fast local decisions. Be responsible. Know your boundaries. Share findings.

BE BOLD

Go into production early. Value monitoring over tests. Fail fast, recover and learn. Optimize for MTTR not MTBF.

DATA-DRIVEN / METRIC-DRIVEN

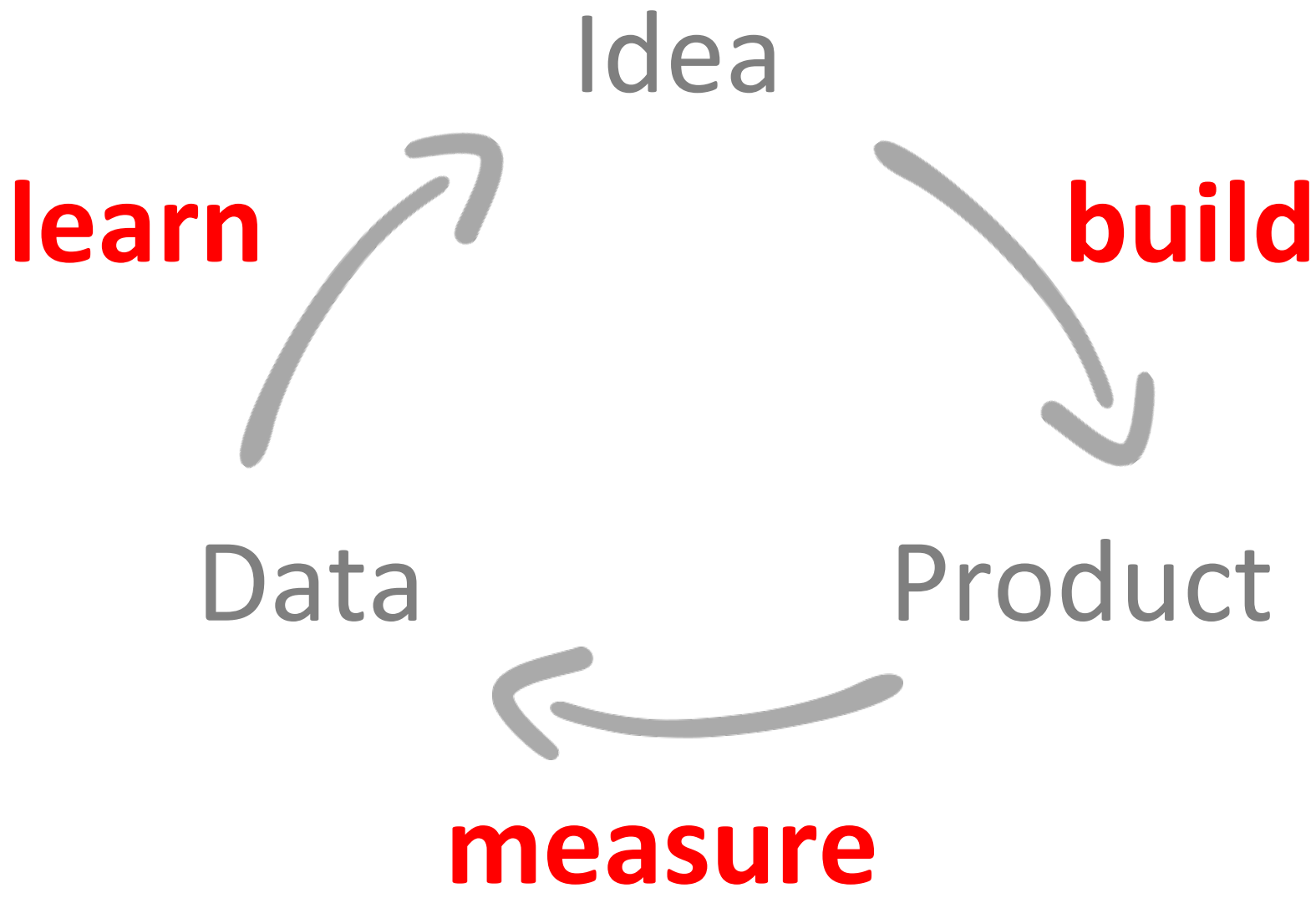
Collect business and operational metrics. Analyze, alert and act on them.

INFRASTRUCTURE AS CODE

Automate everything: Reproducible, traceable, auditable and tested. Immutable servers.

Icons made by Freepik from www.flaticon.com are licensed under CC BY 3.0

Build-Measure-Learn feedback loop



Conway's Law

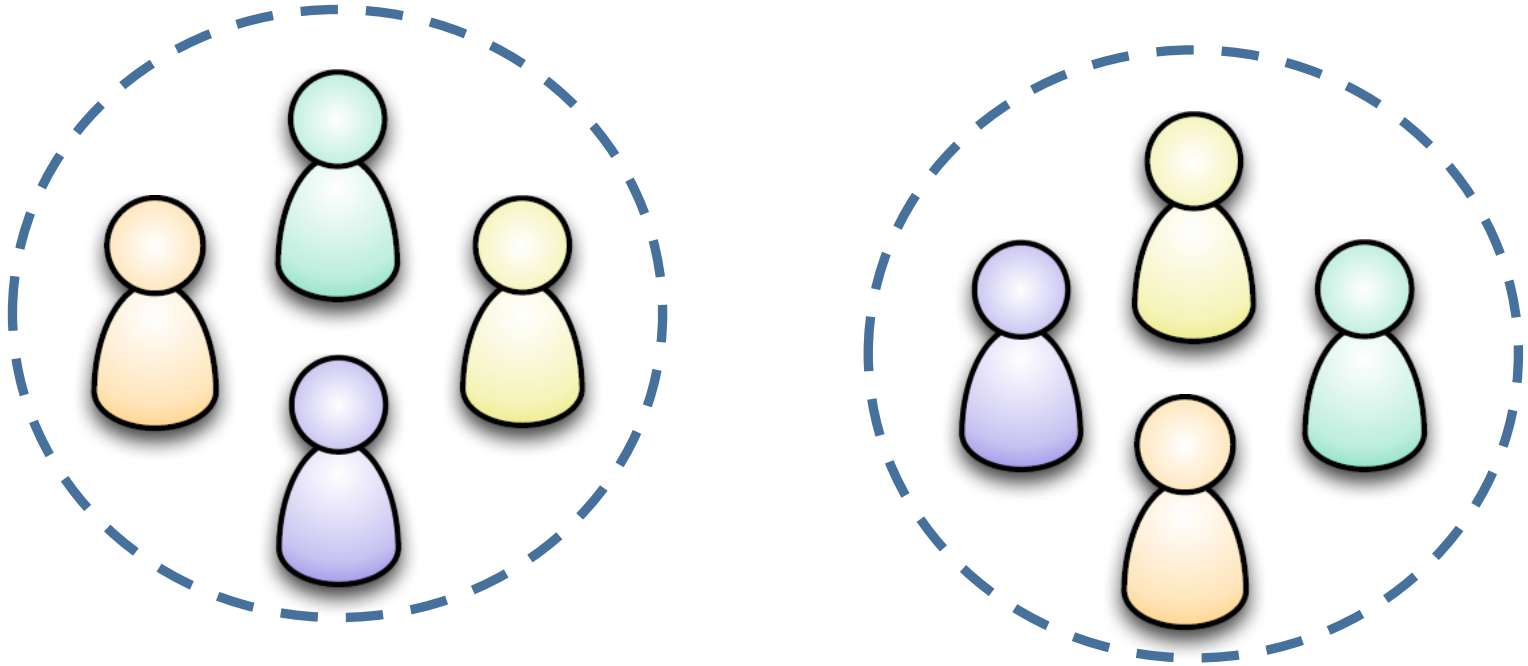
"organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations"



Autonomous teams

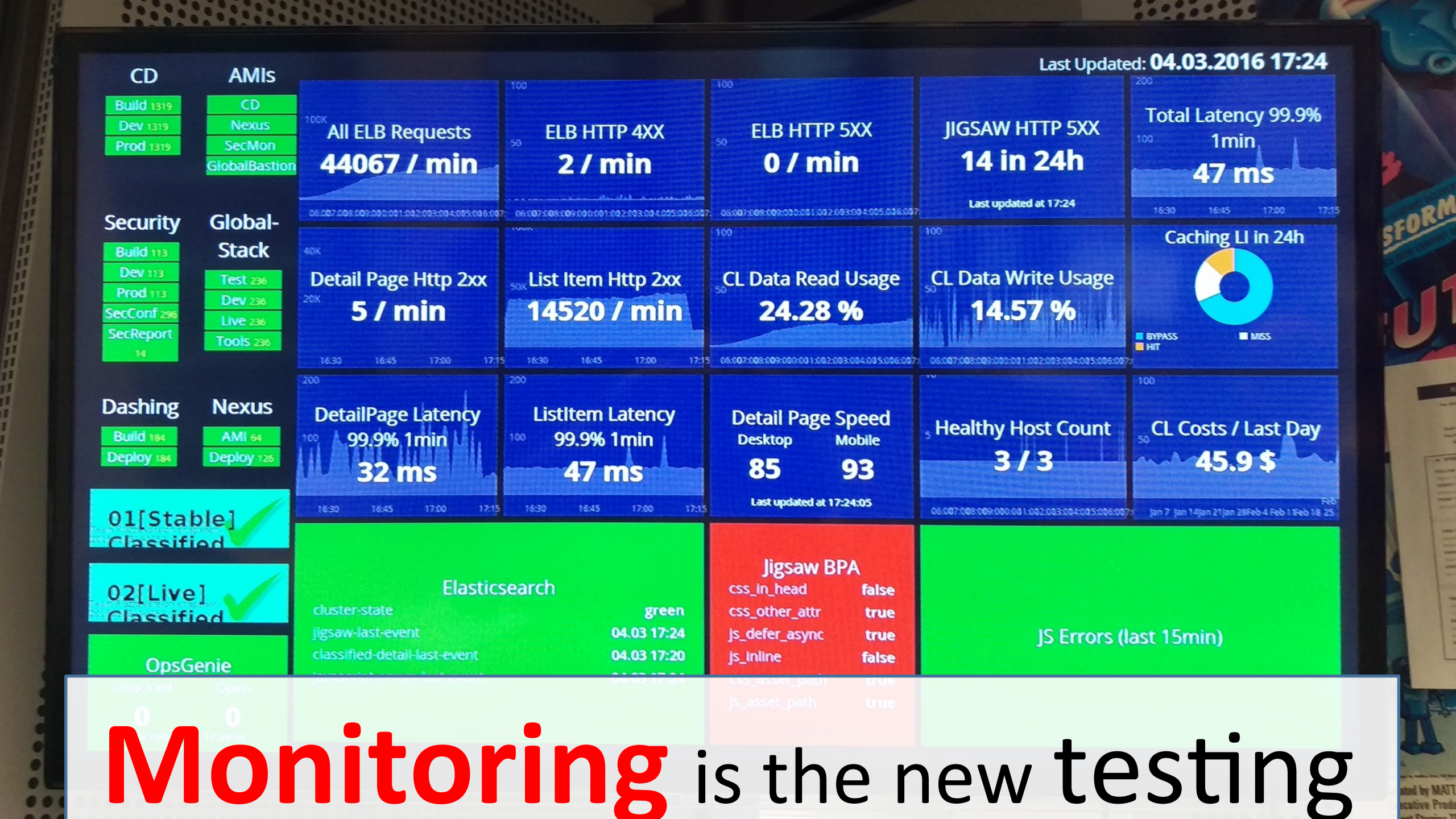
organized around

business capabilities




collaboration
culture


You build it,
you run it.



Monitoring is the new testing


Follow the
trail

Templates

AutoScout24/uber-**template**

Shell ★ 0 0

Updated on 18 May

AutoScout24/tatsu-service-**template**

Scala ★ 0 0

Template to Bootstrap a new Service

Updated 5 days ago

AutoScout24/**template**-lambda-scala

Scala ★ 0 0

Template for scala-based lambda

Updated on 20 May

AutoScout24/frontend-stack-**template**

JavaScript ★ 0 0

Updated 3 days ago

- Faster bootstrapping
- Copied not inherited
- Collect and share best practices

Guilds



Self-organizing; common interests; across teams

Macro Architecture, Infrastructure, Frontend, QA…

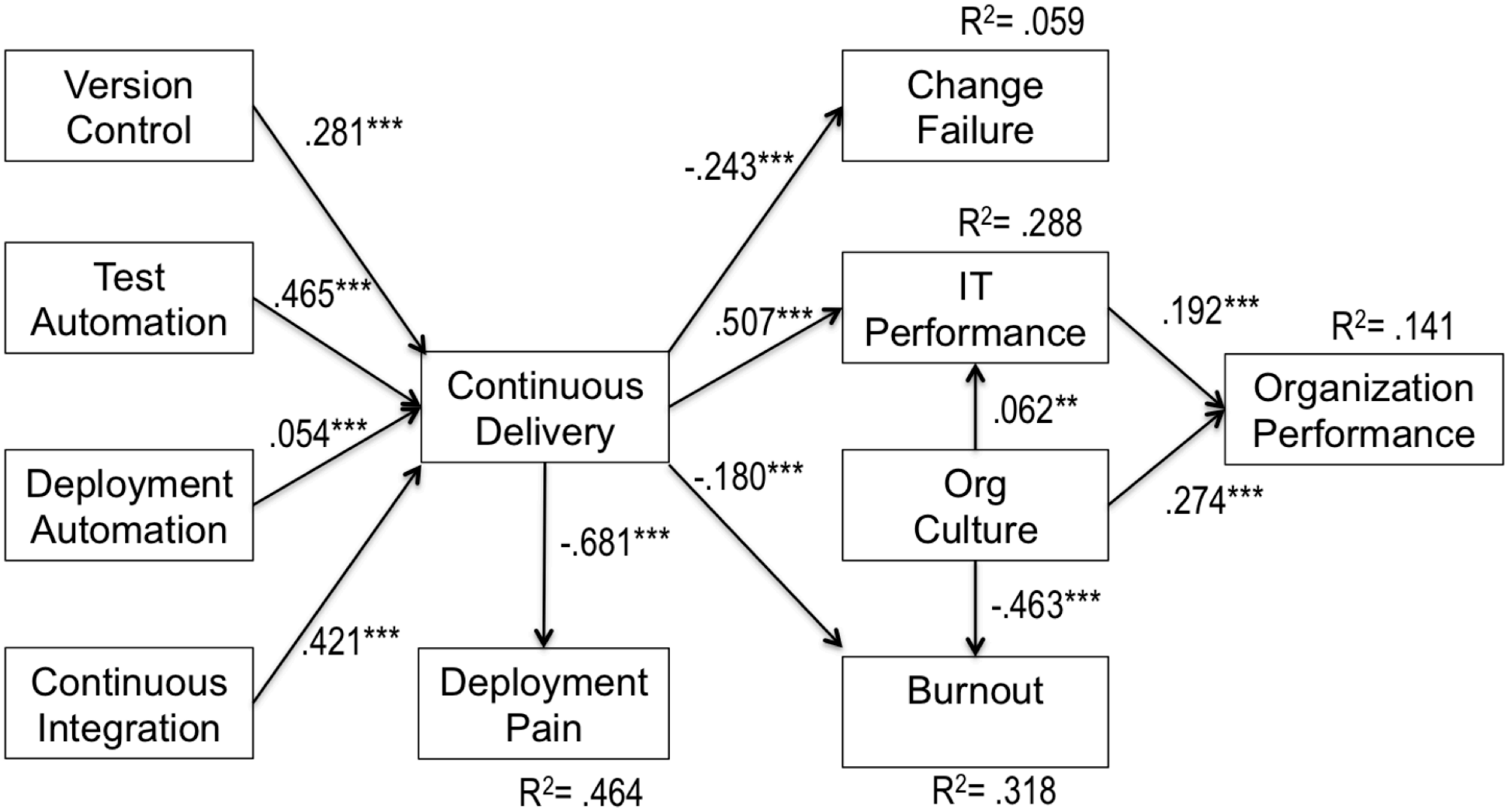
Beware of mandelbrot teams


Continuous
Delivery

DevOps Survey



DevOps Science



Forsgren, Nicole and Humble, Jez, The Role of Continuous Delivery in IT and Organizational Performance (October 27, 2015). Forsgren, N., J. Humble (2016). "The Role of Continuous Delivery in IT and Organizational Performance." In the Proceedings of the Western Decision Sciences Institute (WDSI) 2016, Las Vegas, NV. . Available at SSRN: http://ssrn.com/abstract=2681909 or http://dx.doi.org/10.2139/ssrn.2681909

Delivery Pipeline – Data Center



Application **code** in one repository per service.

Deployment **package** as artifact.

Deliver package to **servers**

Delivery Pipeline – AWS



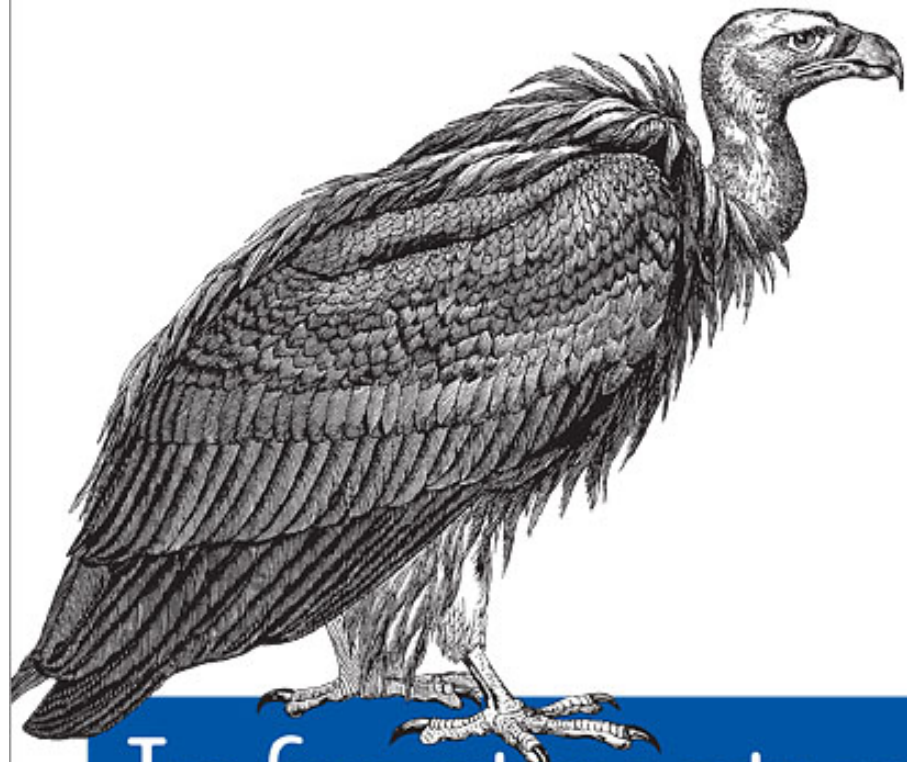
Application **code** and **infrastructure** specification in one repository per service.

Deployment **package** and **infrastructure** declaration as artifact.

1. Create or update service infrastructure.

2. New instances pull down package and start application.

O'REILLY®

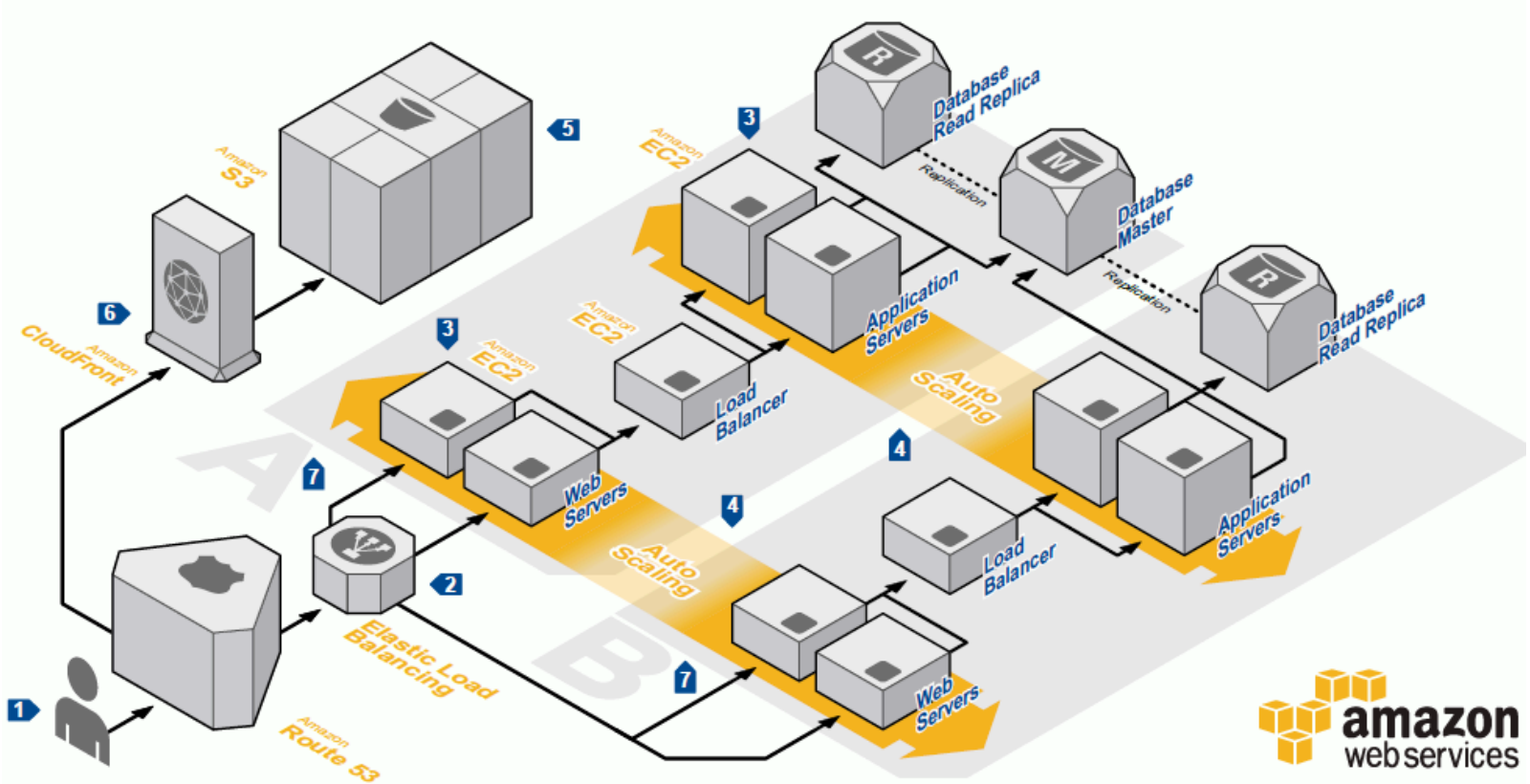


Infrastructure as Code

MANAGING SERVERS IN THE CLOUD

Kief Morris

Unlimited Infrastructure with APIs




Cattle,
not pets

Separate

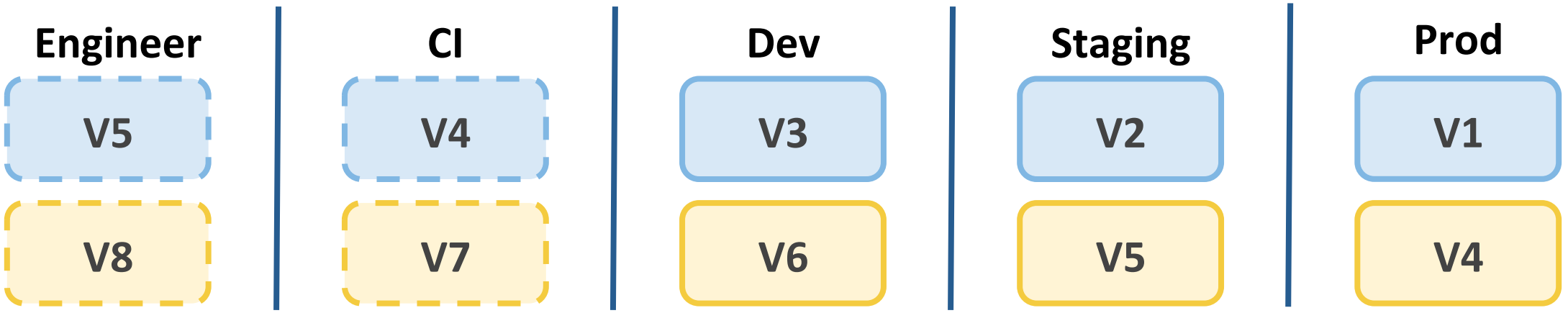
code deployment

from

feature release



How many environments?



Which versions on staging?

Prod differs anyway: Load, data, patterns



No

staging

environment

Integrate in production

- Consumer driven contracts
- Canary releases
- Shadow traffic
- Semantic monitoring

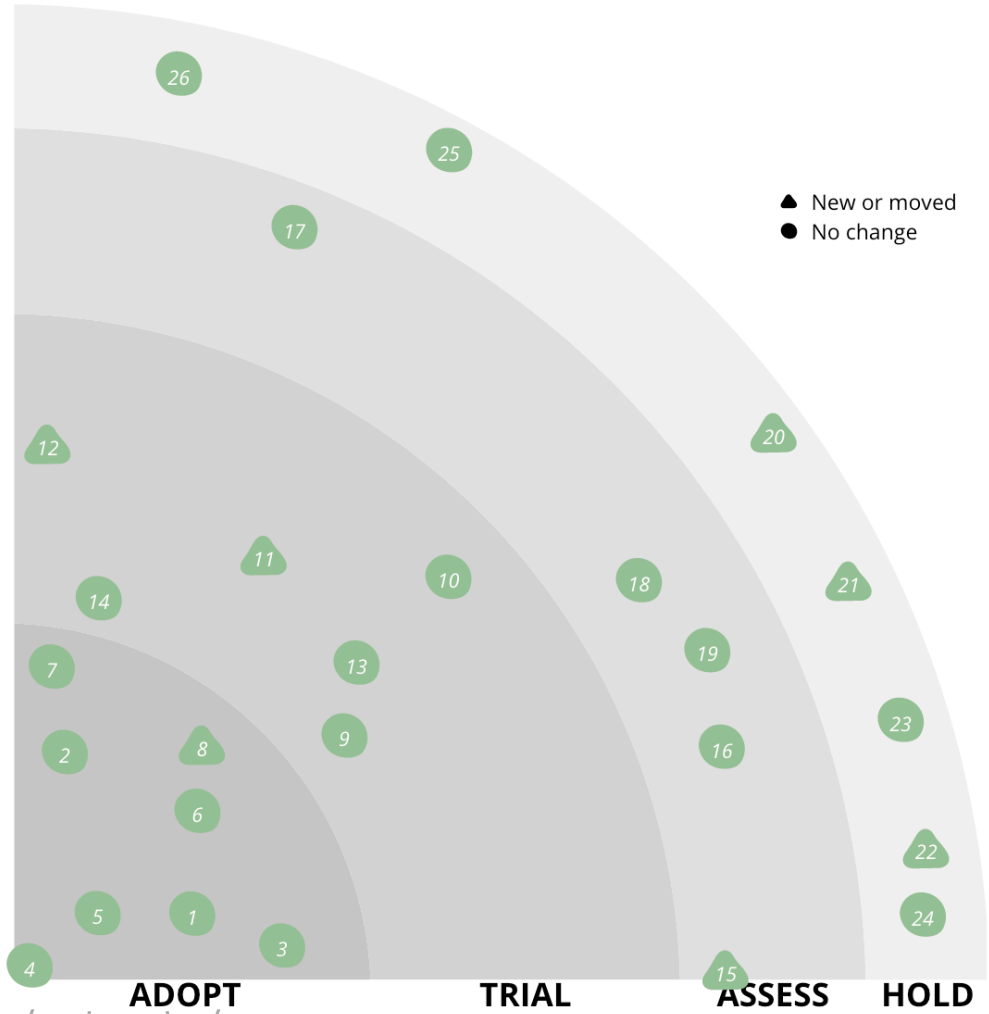


AUTOSCOUT24 RADAR APRIL 2017

Powered by ThoughtWorks®

Techniques | Tools | Platforms | Languages & Frameworks | Print this radar

« Back to Radar home



ADOPT

1. Pre-compute as much as you can
2. Game Day Exercises
3. Feature Toggles 'the proper way'
4. Q.A. in Production
5. Custom Elements
6. External User Journey Monitoring
7. Circuit Breaker
8. Platform engineering product Teams

TRIAL

9. Consumer-Driven Contracts
10. Progressive Webapps
11. Performance Monitoring
12. Logging using STDOUT
13. CQRS/Event Sourcing
14. Business KPI Monitoring

ASSESS

15. Replicate DBTables to Kafka
16. Canary Releases
17. Decouple Build and Deploy Tool
18. Content Security Policy
19. Automated Architecture Diagram

https://autoscout24.github.io/tech-radar/

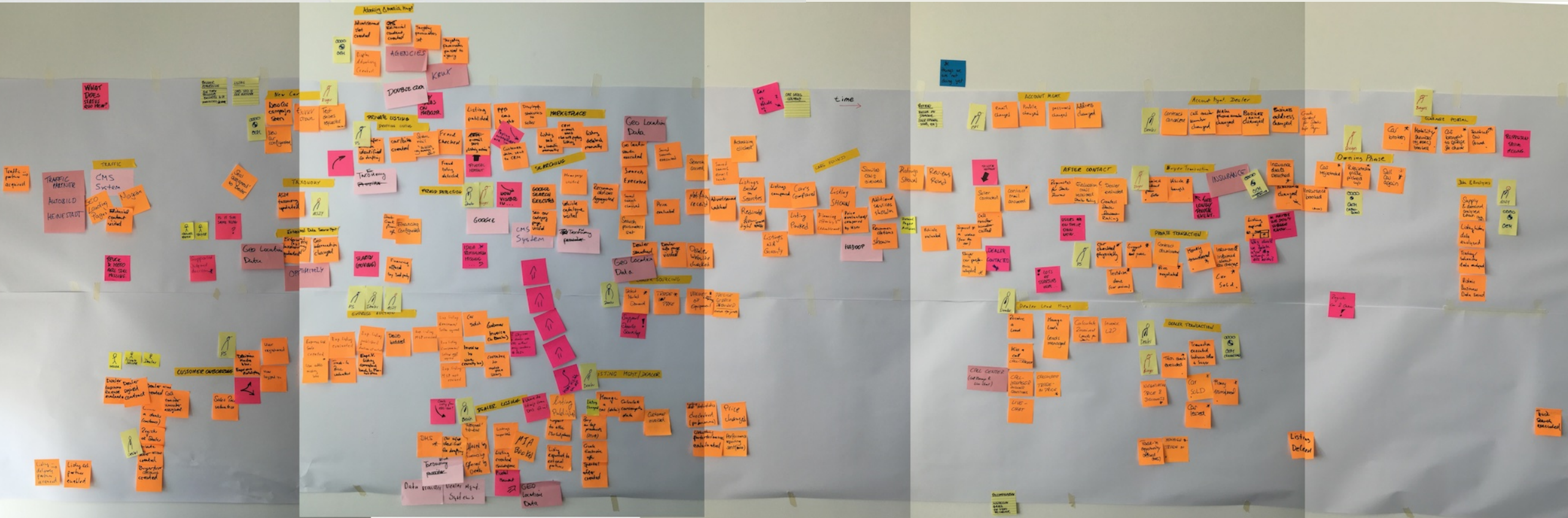
EVENT STORMING

Alberto Brandolini

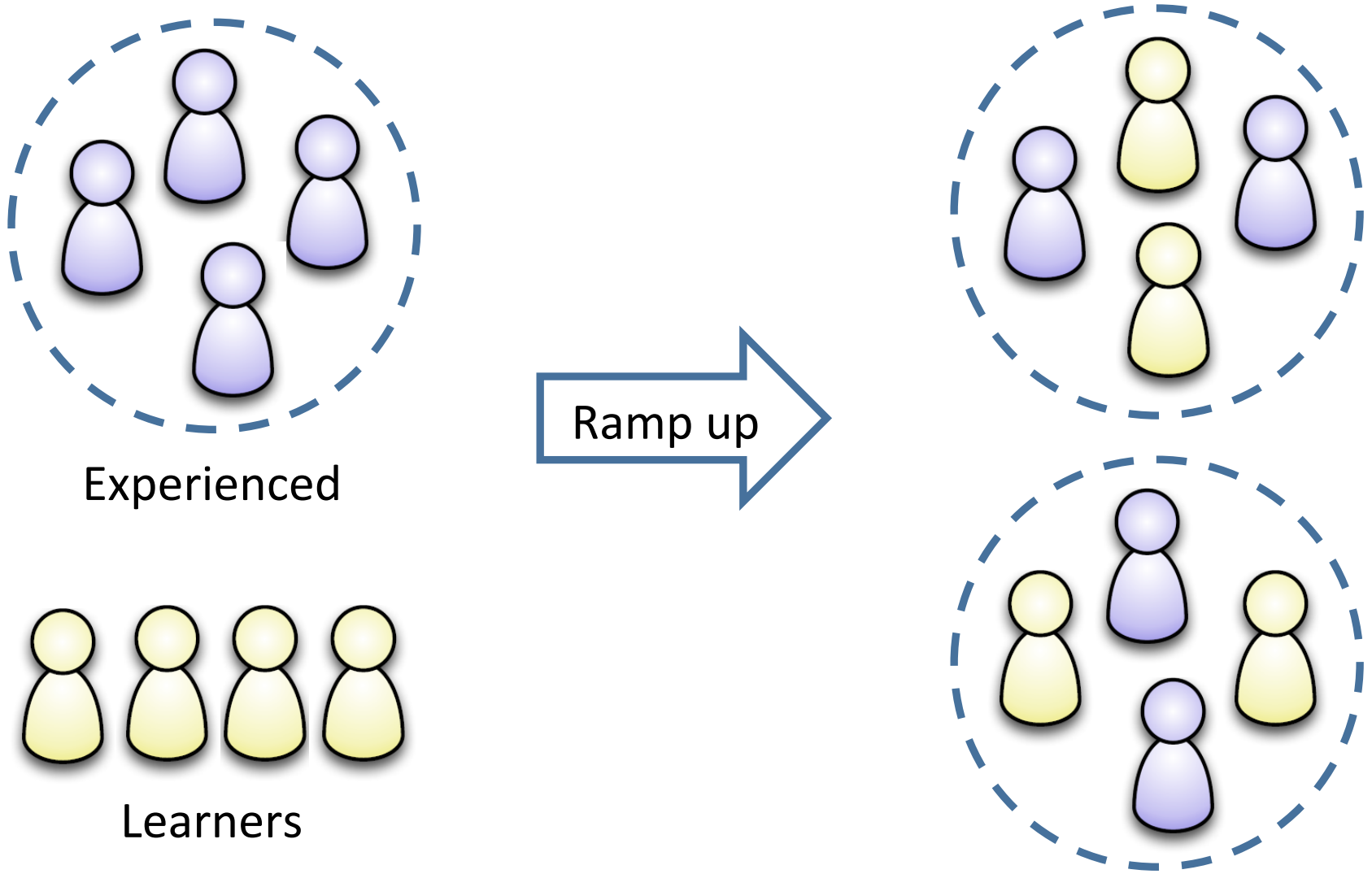
An act of Deliberate Collective Learning

- Big Picture
- Design Level

http://eventstorming.com/



First 8 teams: Split teams to share knowledge



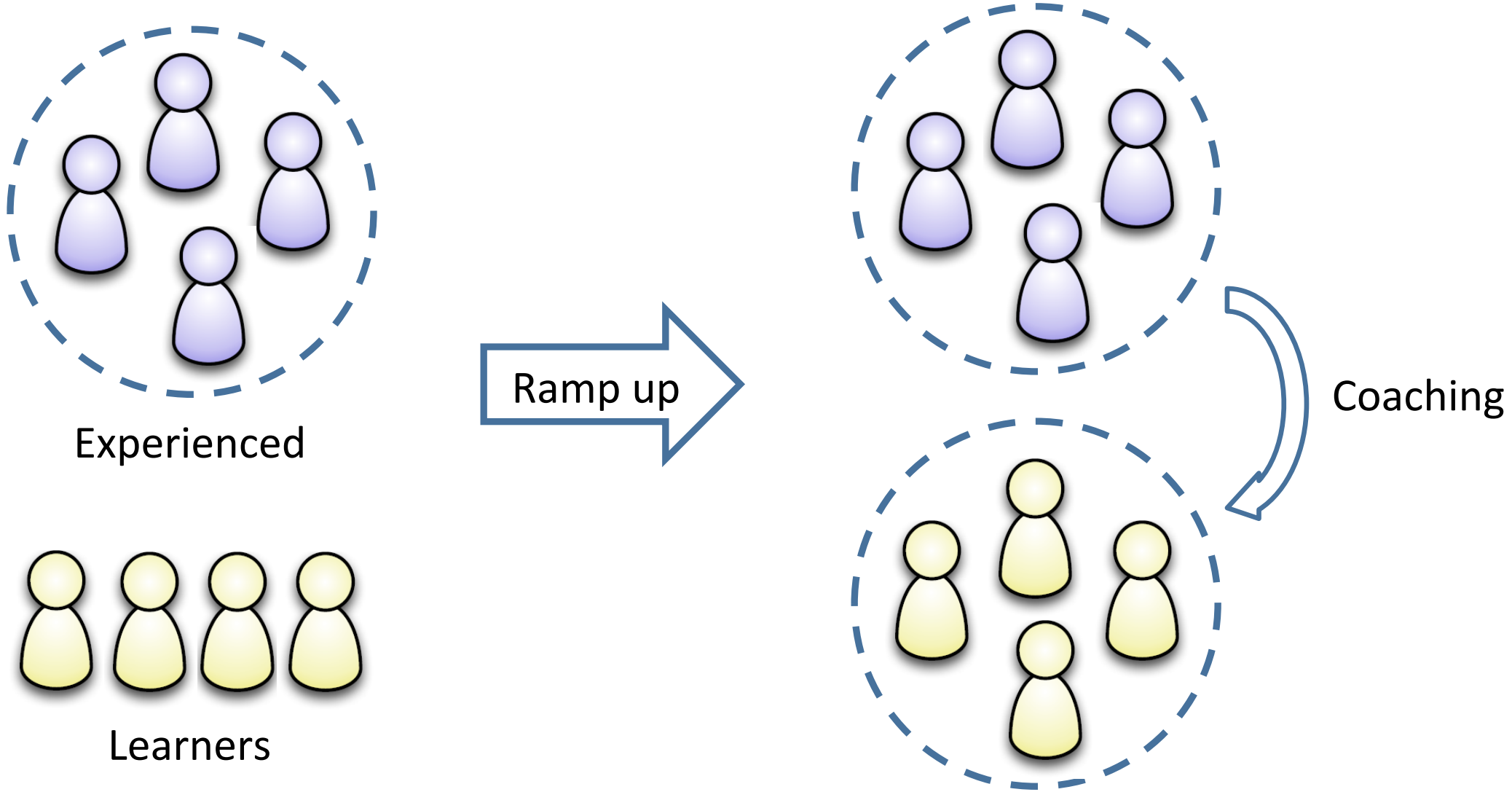
The five keys to a successful Google team

”Who is on a team matters less than how the team members interact, structure their work, and view their contributions.”

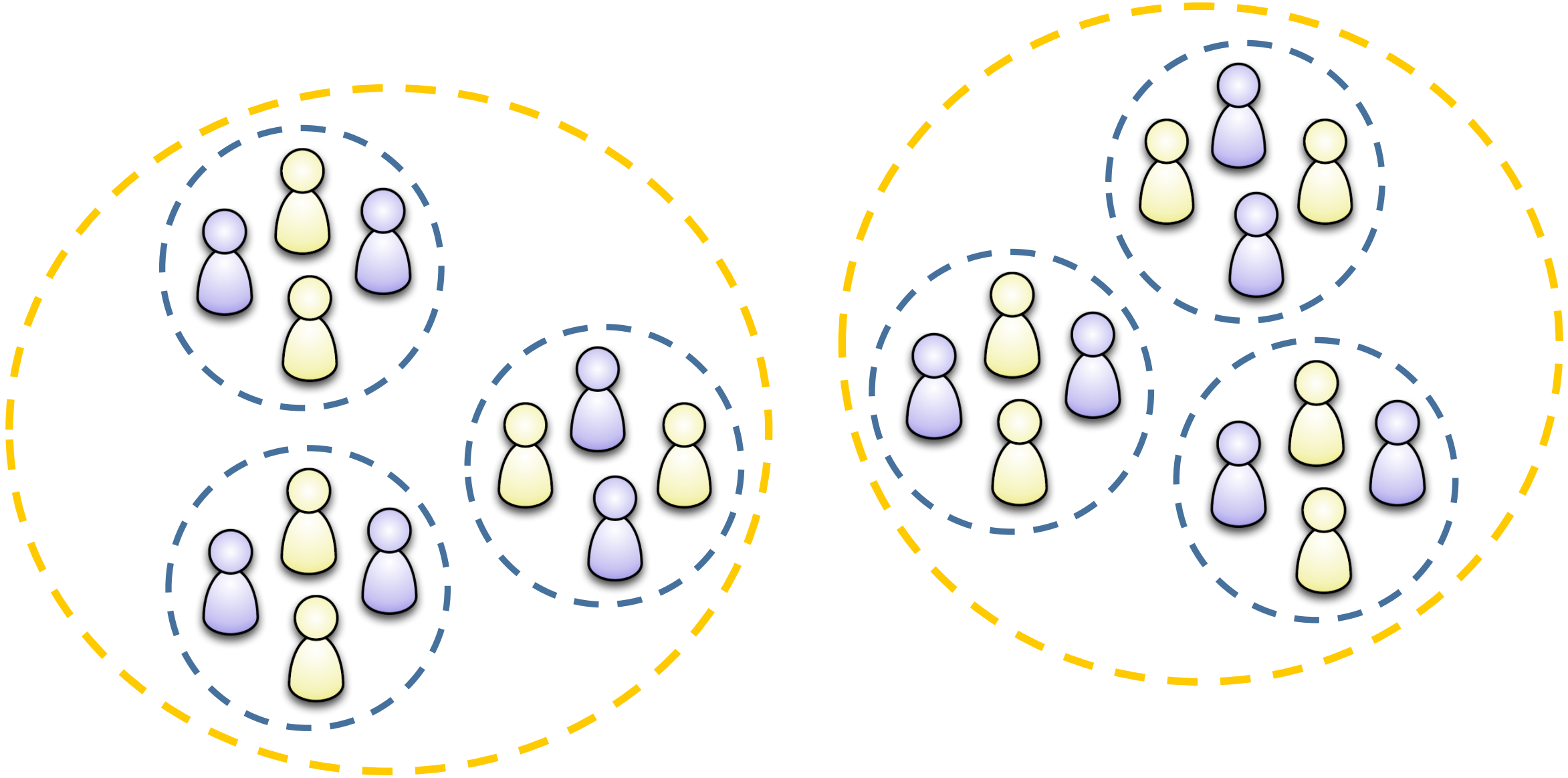


https://rework.withgoogle.com/blog/five-keys-to-a-successful-google-team/

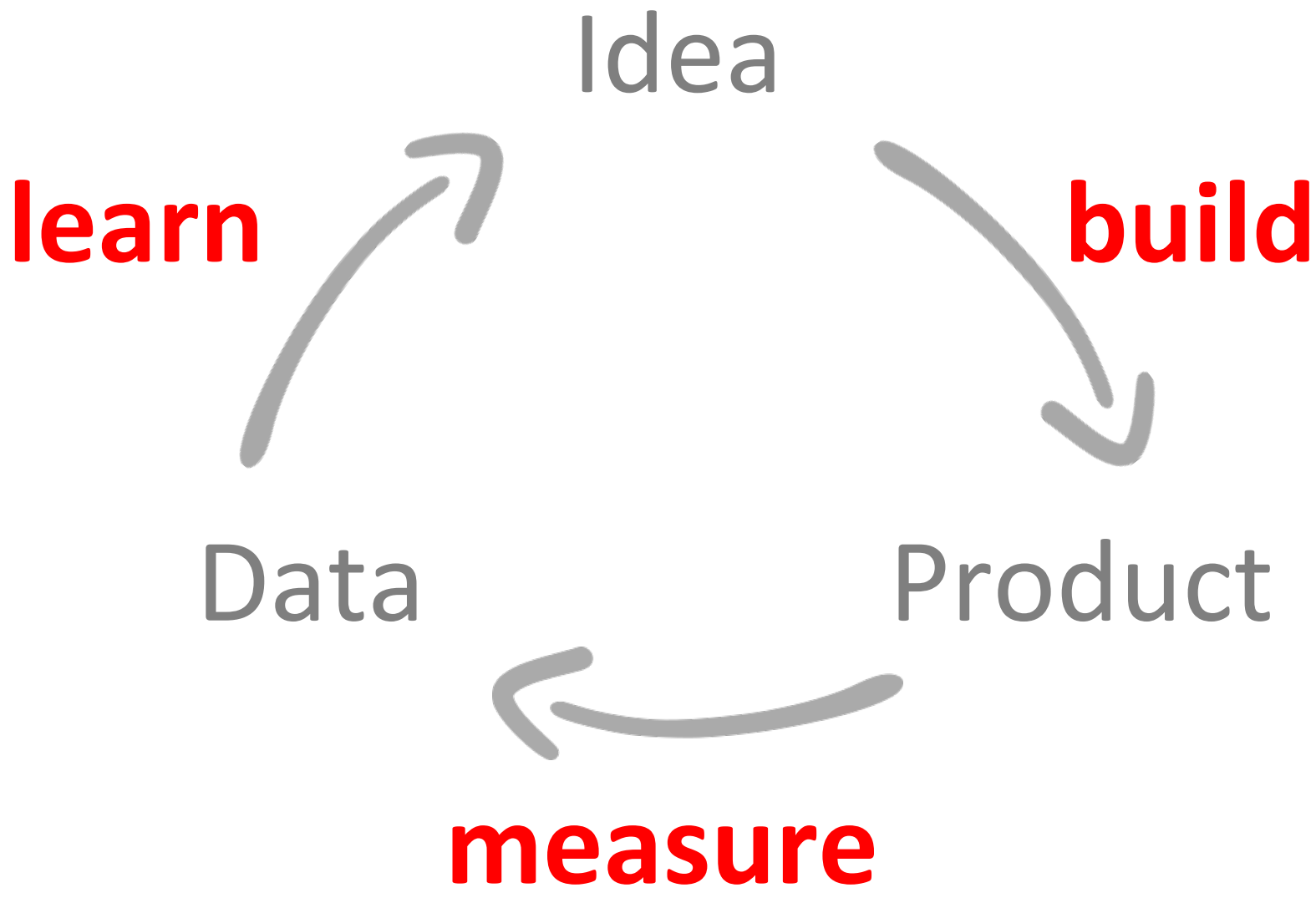
Plan for next teams: Coach don’t split



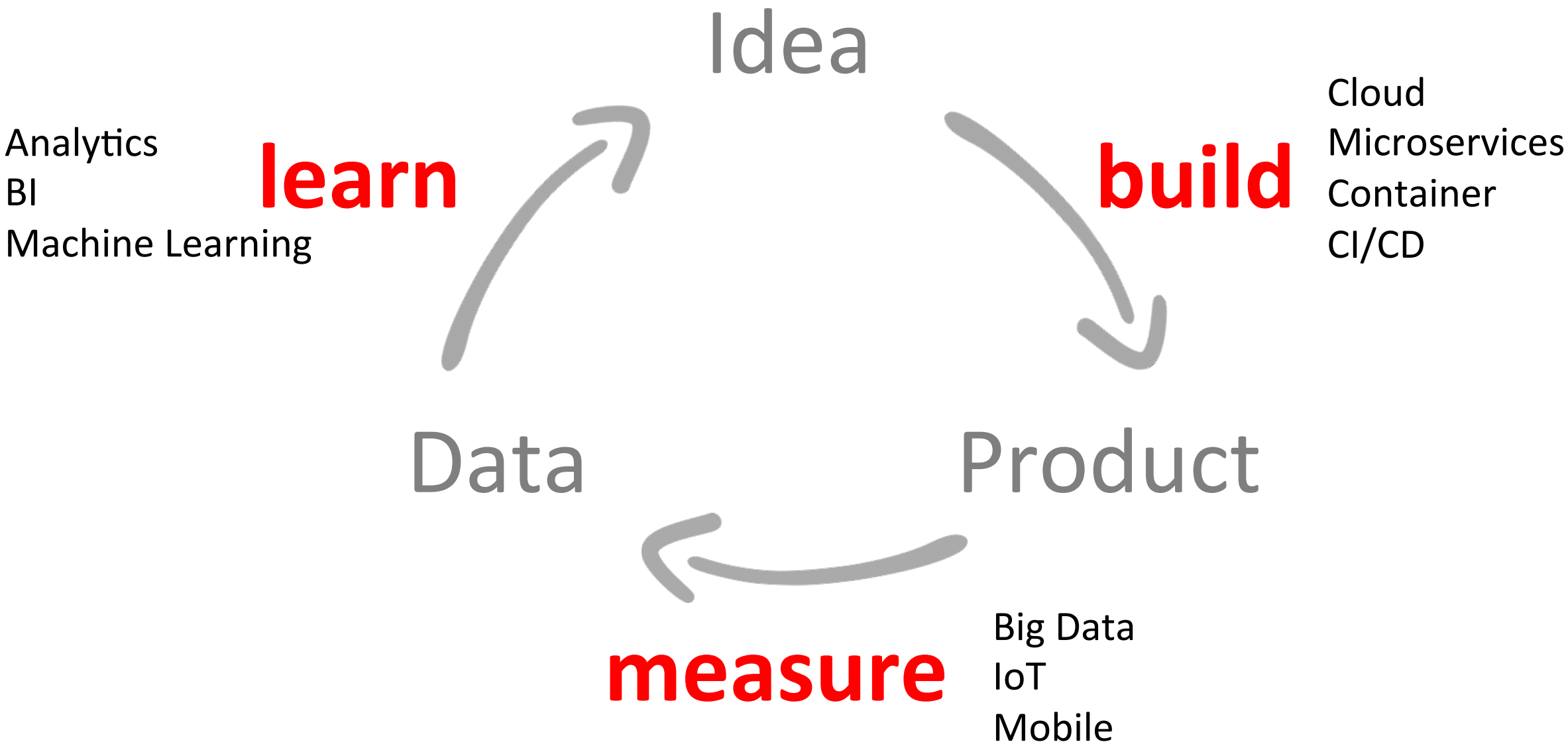
Tribes/Segments/Departments



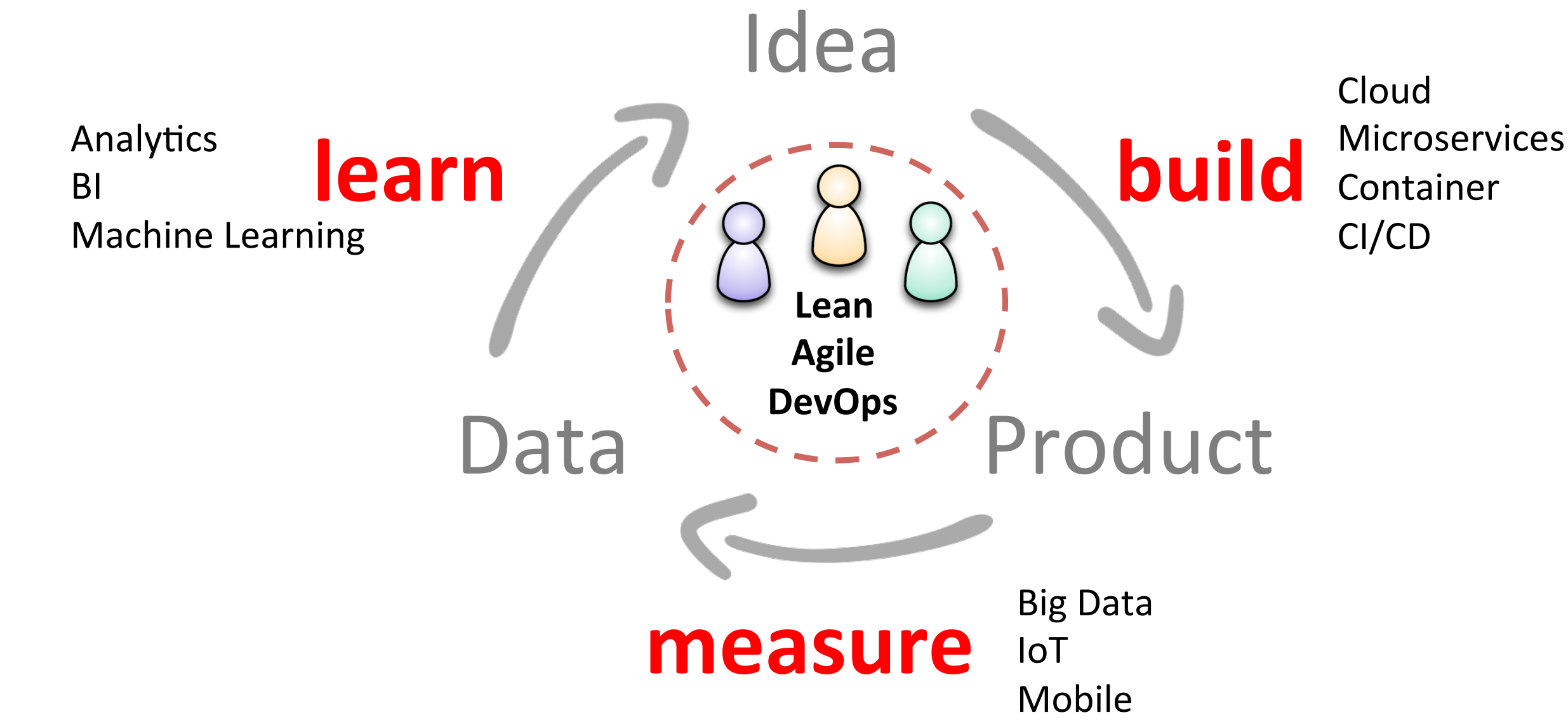
Build-Measure-Learn feedback loop



New technologies enable technical agility…



…and culture enables organizational agility




TWENTY QUESTIONS
DICK
HERB
VAN
FLORENCE
ALDO RAY

?

Picture Credits

"HotWheels - '69 Ford Torino Talladega“ by Leap Kye, licensed under CC BY-ND 2.0

Enterprise IT Adoption Cycle by Simon Wardley under CC BY-SA 3.0

And the future is private by Simon Wardley under CC BY-SA 3.0

Leosvel et Diosmani by Ludovic Péron under CC BY-SA 3.0

Wandergeselle by Sigismund von Dobschütz under CC BY-SA 3.0

Puzzling by Bernd Gessler (Own work) CC BY-SA 3.0

Werner Vogels by Guido van Nispen under CC BY 2.0

Bezos’ Iconic Laugh by Steve Jurvetson under CC BY 2.0