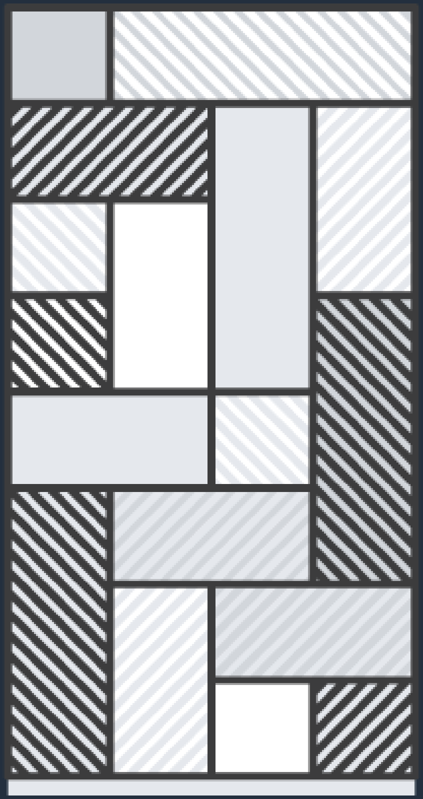
DevOps at Amazon

Emil Lerch
Principal DevOps Specialist

A look back at development at Amazon…

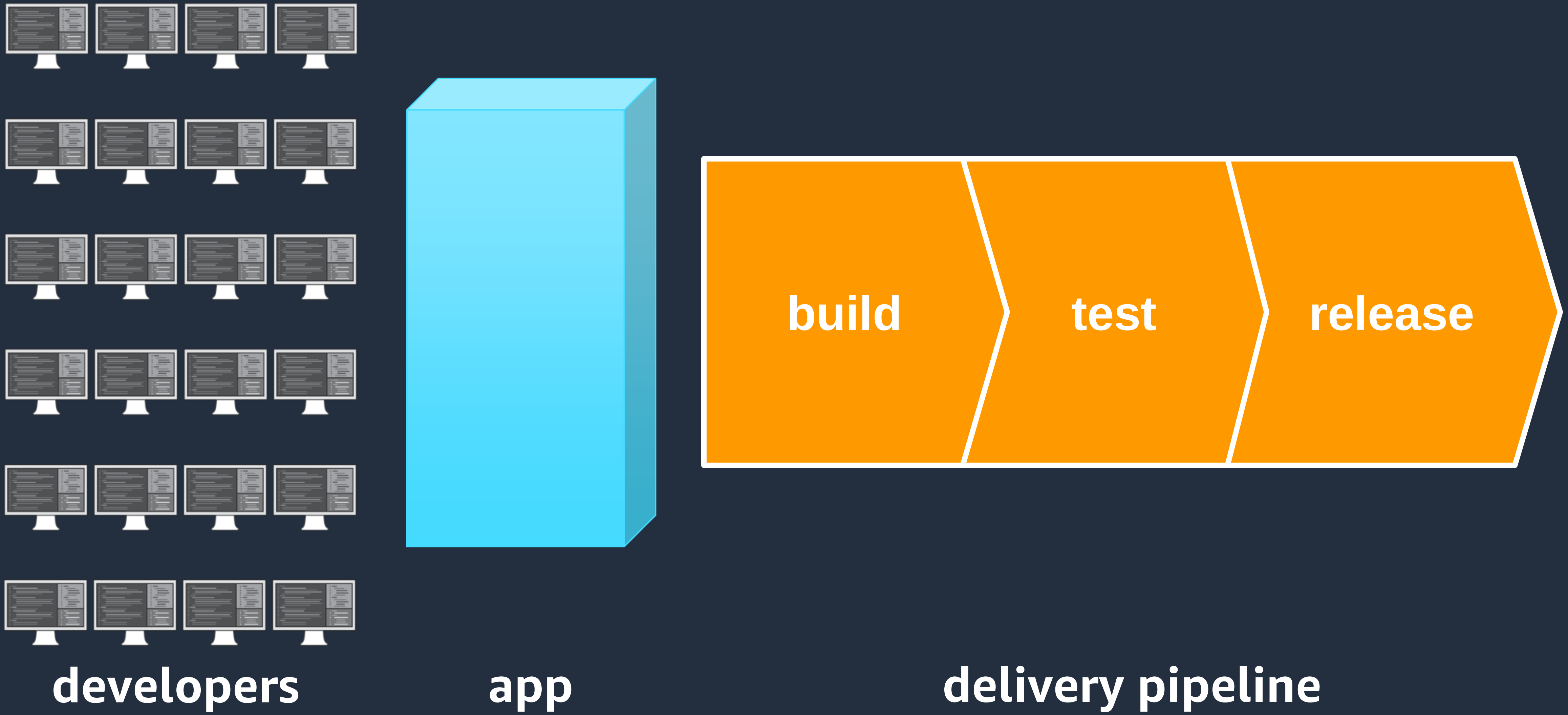
https://secure.flickr.com/photos/pixelthing/15806918992/

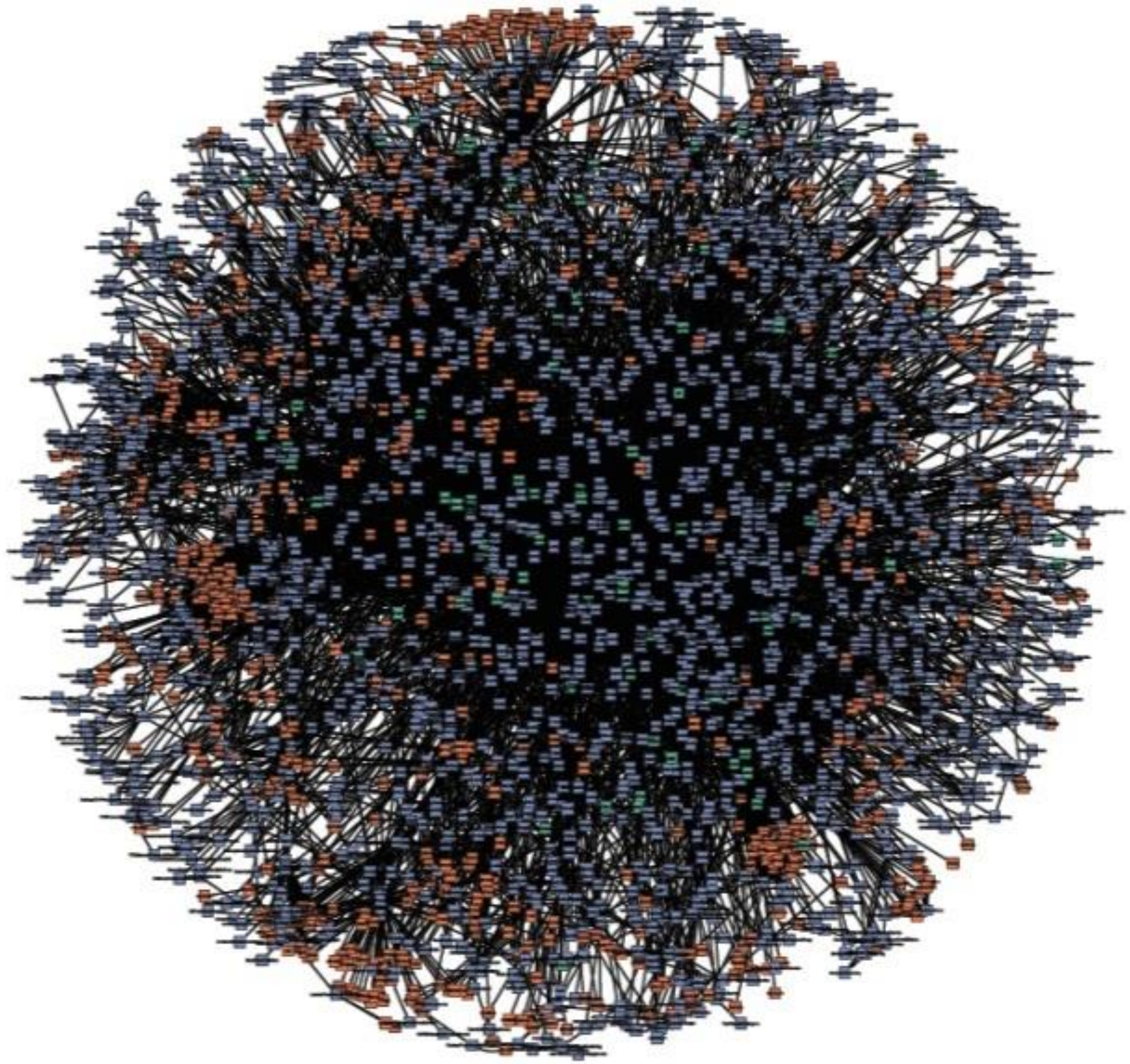
2001



monolithic application
+
monolithic teams

Monolith development lifecycle





Single-purpose

Connect only through APIs

Connect over HTTPS

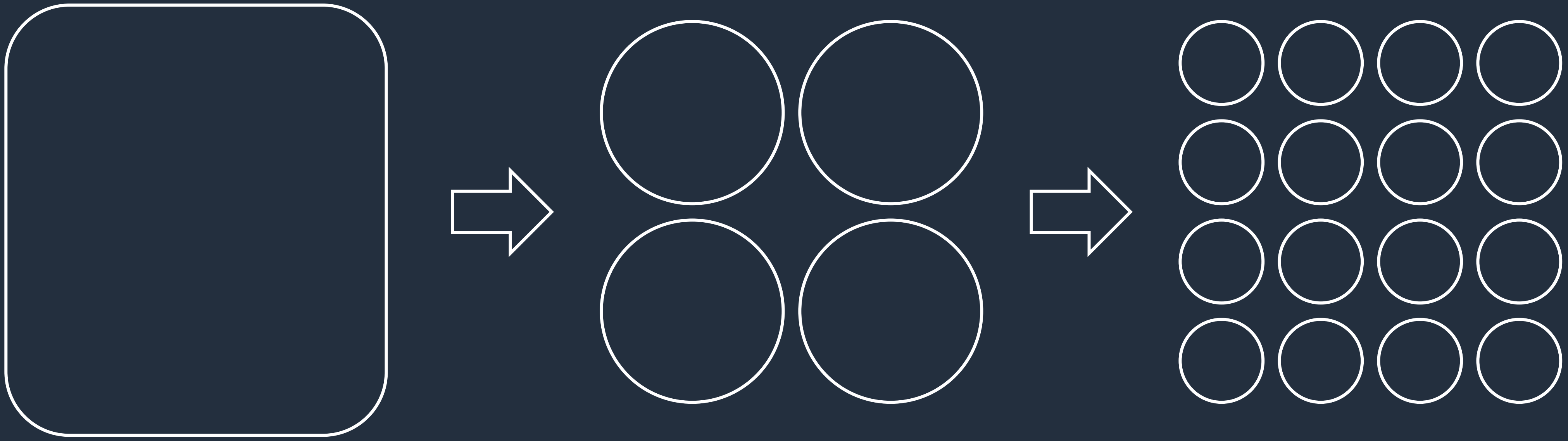
Largely “black boxes” to each other

“Microservices”

Isn’t this just SOA rebranded?

https://www.flickr.com/photos/pagedooley/15461138759/

Monolithic vs. SOA vs. Microservices



Monolithic
Single Unit

SOA
Coarse-grained

Microservices
Fine-grained

Microservices vs. SOA

Microservices

Many very small components

Business logic lives inside of single service domain

Simple wire protocols(HTTP with XML/JSON)

API driven with SDKs/Clients

SOA:

Fewer more sophisticated components

Business logic can live across domains

Enterprise Service Bus like layers between services

Middleware

Two-pizza teams

Full ownership

Full accountability

Aligned incentives

"DevOps"

How do Two Pizza Teams work?

We call them "Service teams"

Own the "primitives" they build:

- Product planning (roadmap)
- Development work
- Operational/Client support work

"You build it, you run it"

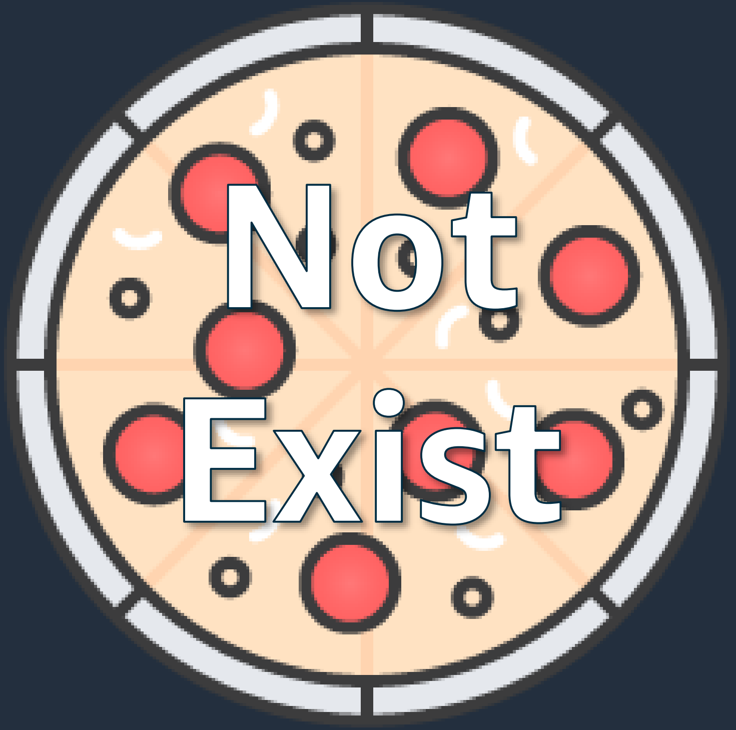
Part of a larger concentrated org (Amazon.com, AWS, Prime, etc)

Who Does QA?

Who Does On Call?

Image By: Chris Munns – munns@amazon.com

What does Ops Do?



What about Ops/QA/Etc?

Everyone exists on a "service team" focused on their primitive(s):

SDE's focused on developing

PM's focused on product direction

$\left.\right\}$ Most "2 pizza" teams are just these 2 roles

TPM's help drive development

SE's focused on infra/tooling

SDET's focused on test excellence throughout the organization

Some folks are shared across the org, some on individual teams

Boy, that sounds like a lot of freedom?

It is! Teams are empowered and also held to high standards:

Thorough onboarding/training

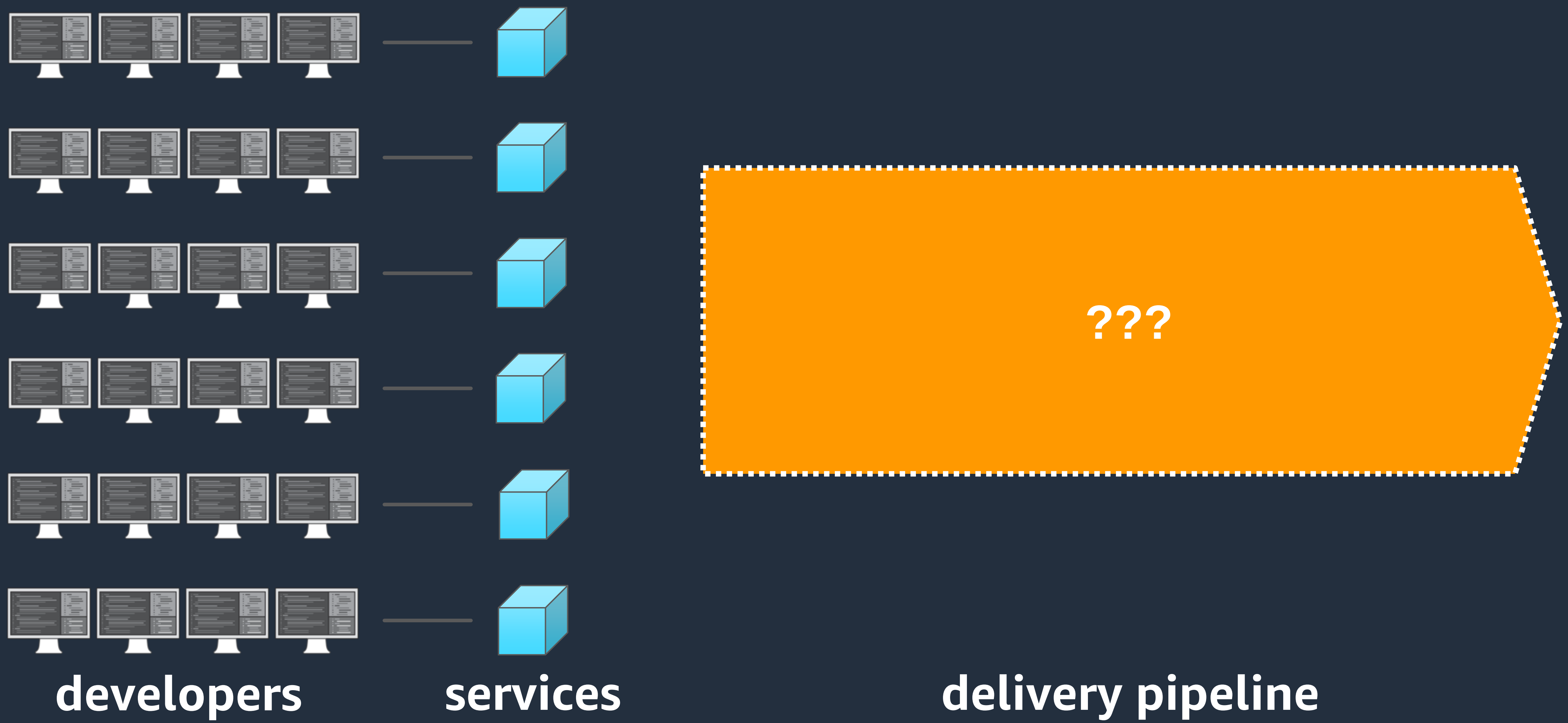
Patterns/practices defined at scale and with 20+ years of organizational knowledge

Regular technical and business metric reviews

Regular sharing of new tools, services, technologies, etc, by internal subject matter experts

Public sharing of COEs; "Correction of Errors" our post-mortem process/tool

Missing tools



Self-service

Technology-agnostic

Encourage best practices

Single-purpose services

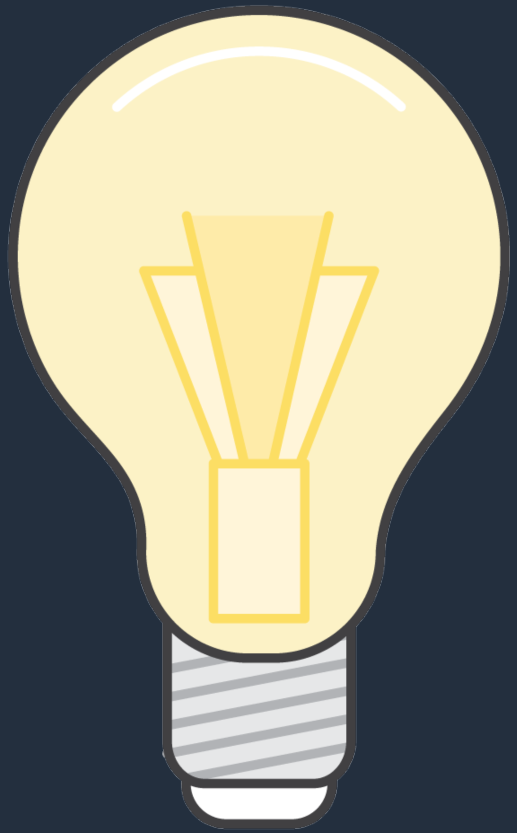
Deployment service

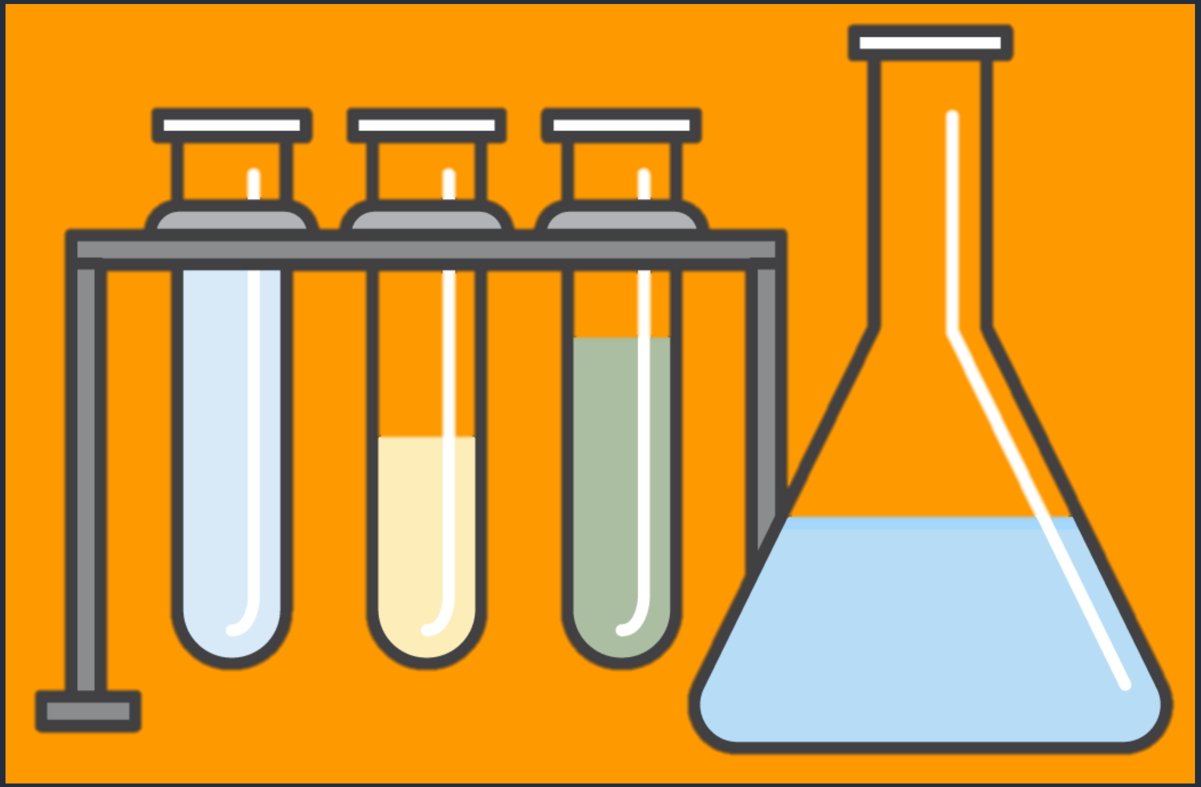
No downtime deployments

Health checking

Versioned artifacts and rollbacks

Things went much better under this model and teams were developing features faster than ever, but we felt that we could still improve.





In 2009, we ran a study to find out where inefficiencies might still exist. We found that many teams were still being slowed down by manual processes and work flows.

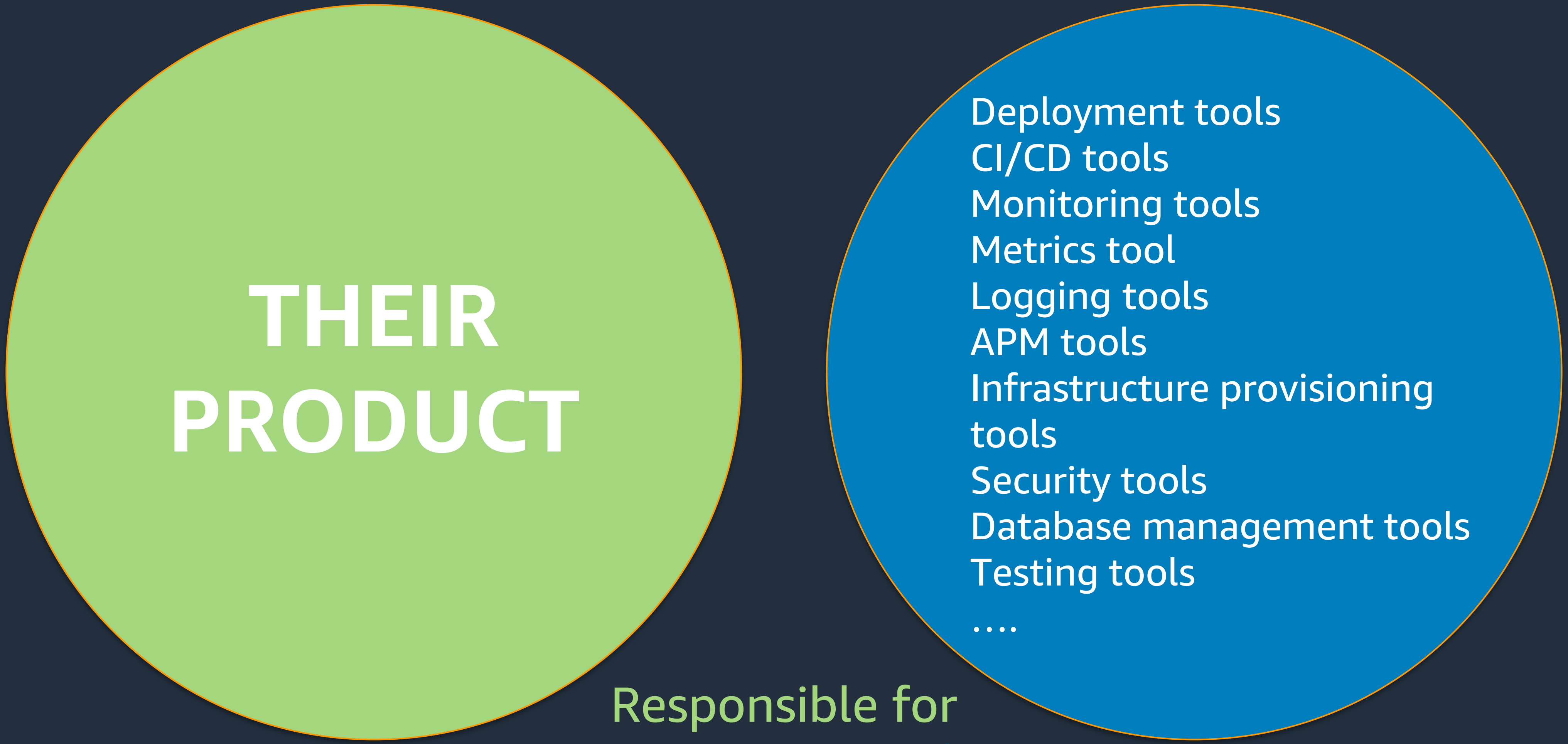
Pipelines

Automated actions and transitions; from check-in to production

Development benefits:

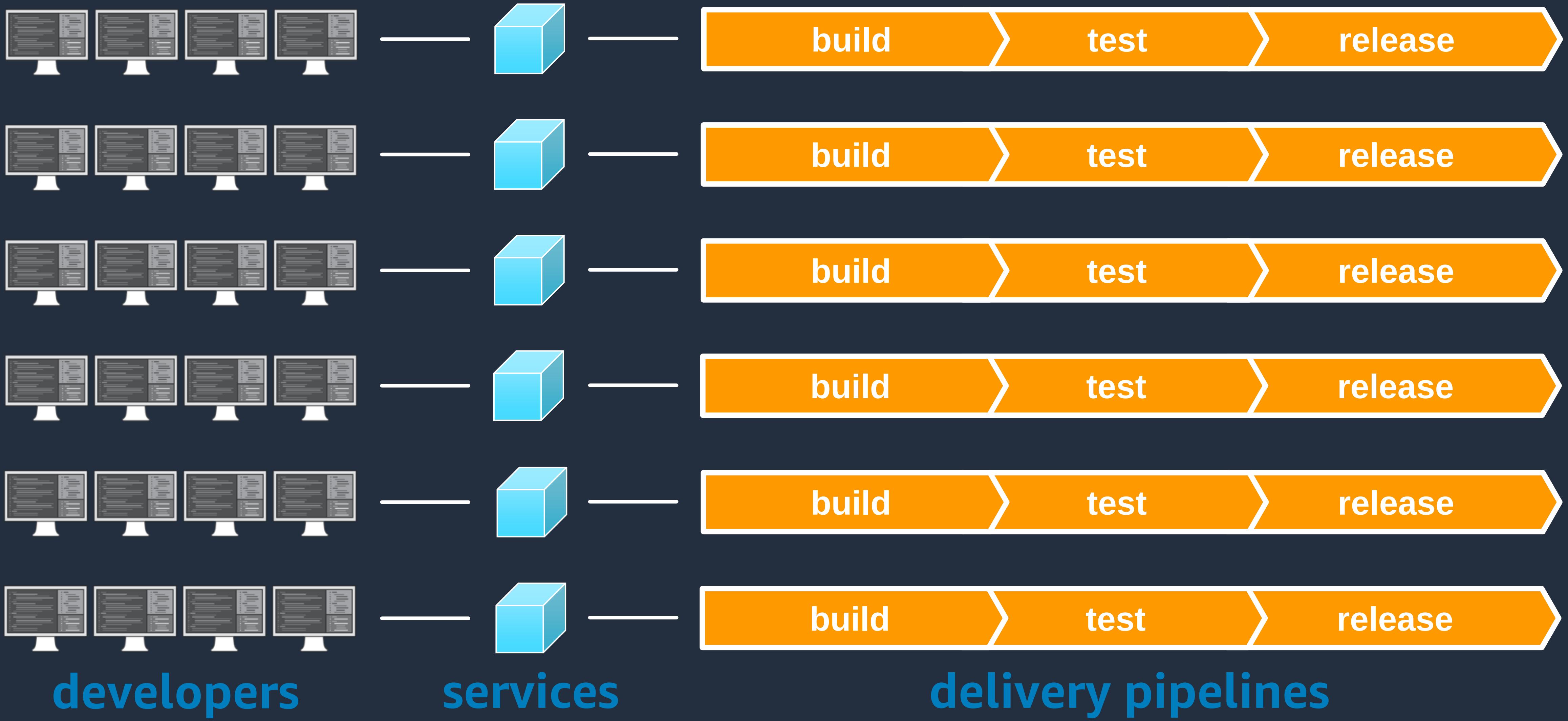
- Faster
- Safer
- Consistent & Standardized
- Visualization of the process

2 Pizza Team Responsibility Venn Diagram



Responsible for
Not responsible for

Microservice development lifecycle



This has continued to work out really well:

Every year at Amazon, we perform a survey of all our software developers. The 2014 results found only one development tool/service could be correlated statistically with happier developers:

Our pipelines service!

continuous delivery == happier developers

Thousands of teams
× Microservice architecture
× Continuous delivery
× Multiple environments

= 50 million deployments a year*

*2014 number

What is DevOps?

Cultural Philosophy ✚ Practices ✚ Tools

What is DevOps?

Cultural Philosophy

Practices

Tools

What is DevOps?

Cultural Philosophy

Practices

Tools

- Tearing down barriers
  - Between teams
  - Mid-process
- Enable the smart people you are spending time and money hiring to make smart decisions
- Assigning ownership, accountability, responsibility to the people doing the work, aka “you build it, you run it”
- Reducing responsibility to the most directly involved individuals
- Increase visibility to the big picture and the results of work being done

What is DevOps?

Cultural Philosophy

Practices

Tools

- Continuous Integration
  - Application testing/QA work applied throughout the development
- Continuous Delivery
  - Automated deployment capabilities of code across environments
- Infrastructure as Code
  - No hand carved infrastructure
- Self-service environments
  - Remove procurement blockers for basic needs
- Microservices
  - Break down complicated monolithic applications in to smaller ones

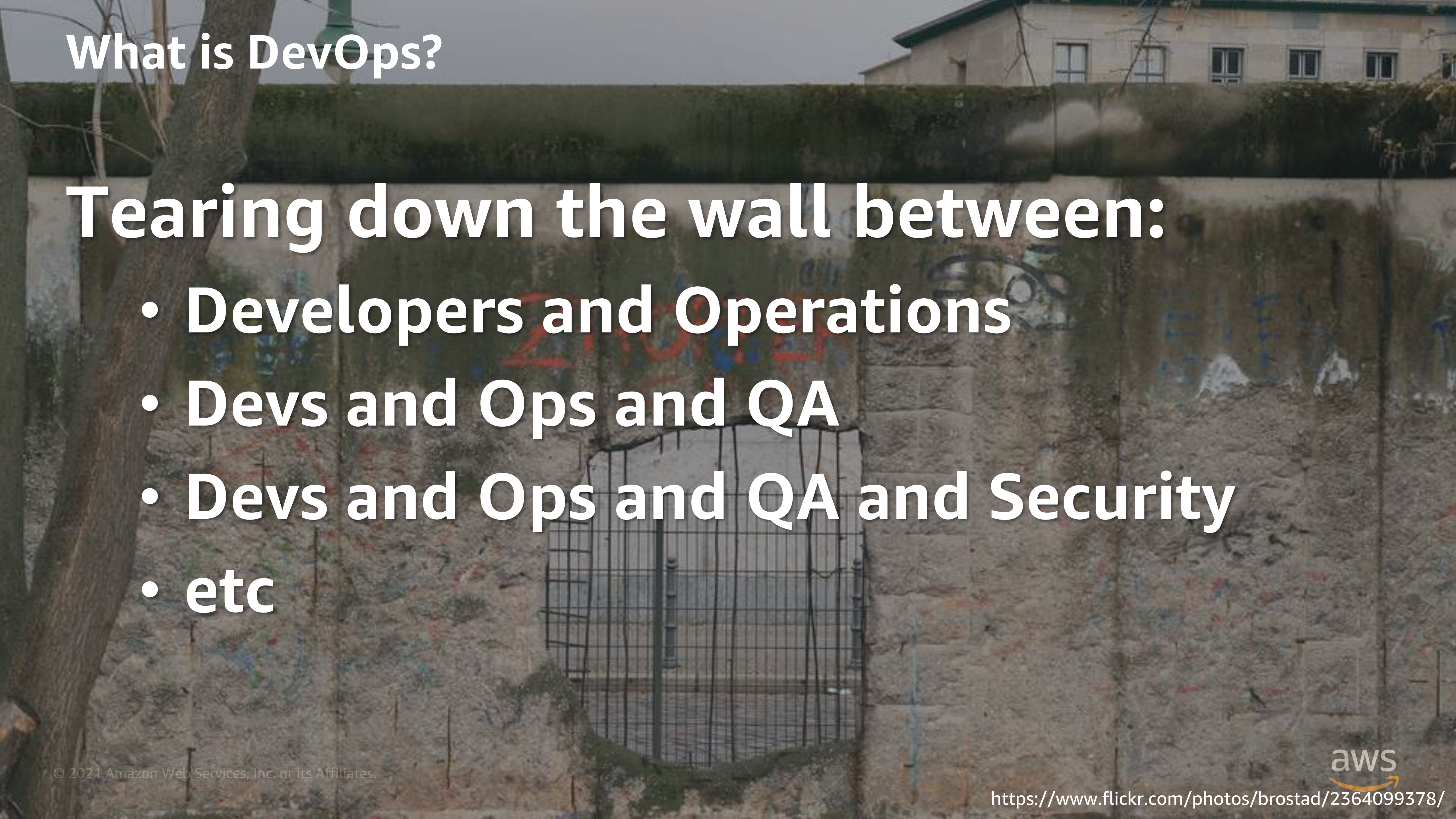
What is DevOps?

Cultural Philosophy

Practices

Tools

- Automated development pipeline tooling
    - Application testing frameworks
    - Code review/feedback tools
    - Automated static analysis
- Consistent and predictable application management & configuration management tools
- Consistent infrastructure measurement tools
    - Metrics
    - Logging
    - Monitoring
    - APM
- Security analysis and management tools



What is DevOps?

Tearing down the wall between:

- Developers and Operations
- Devs and Ops and QA
- Devs and Ops and QA and Security
- etc

https://www.flickr.com/photos/brostad/2364099378/

Teams that adopt modern software practices are more agile and higher performing

Teams who automate software delivery with continuous delivery:

DEPLOYMENT FREQUENCY	Weekly–monthly	→	Hourly–daily
CHANGE LEAD TIME	1–6 months	→	1–7 days
CHANGE FAILURE RATE	46–60%	→	0%–15%

Source: 2019 DORA State of DevOps report

Fully-automated processes

Teams who automate software delivery with continuous delivery:

CHANGE MANAGEMENT EFFECTIVENESS	3 times more effective
RESTORE SERVICE AFTER INCIDENT LESS THAN A DAY	77% of teams with evolved DevOps processes
FULLY REMEDIATE SECURITY VULNERABILITY LESS THAN A DAY	60% of teams with evolved DevOps processes
SELF-SERVICE/ EMPLOYEE INVOLVEMENT	13% of employees more likely to understand and enjoy the process

Source: 2019 DORA State of DevOps report