

ThoughtWorks®

INNOVATION WORKSHOP

CLOUD NATIVE

Outcome over output

about me

AGENDA

- 1 ***What is cloud native?***
- 2 ***Why cloud native organization?***
- 3 ***Principles***
- 4 ***Patterns***
- 5 ***Practices***

ICE BREAK



what “cloud native” is in your city?

ThoughtWorks®

CLOUD NATIVE

THE TWELVE FACTORS

I. Codebase

One codebase tracked in revision control, many deploys

II. Dependencies

Explicitly declare and isolate dependencies

III. Config

Store config in the environment

IV. Backing services

Treat backing services as attached resources

V. Build, release, run

Strictly separate build and run stages

VI. Processes

Execute the app as one or more stateless processes

VII. Port binding

Export services via port binding

VIII. Concurrency

Scale out via the process model

IX. Disposability

Maximize robustness with fast startup and graceful shutdown

X. Dev/prod parity

Keep development, staging, and production as similar as possible

XI. Logs

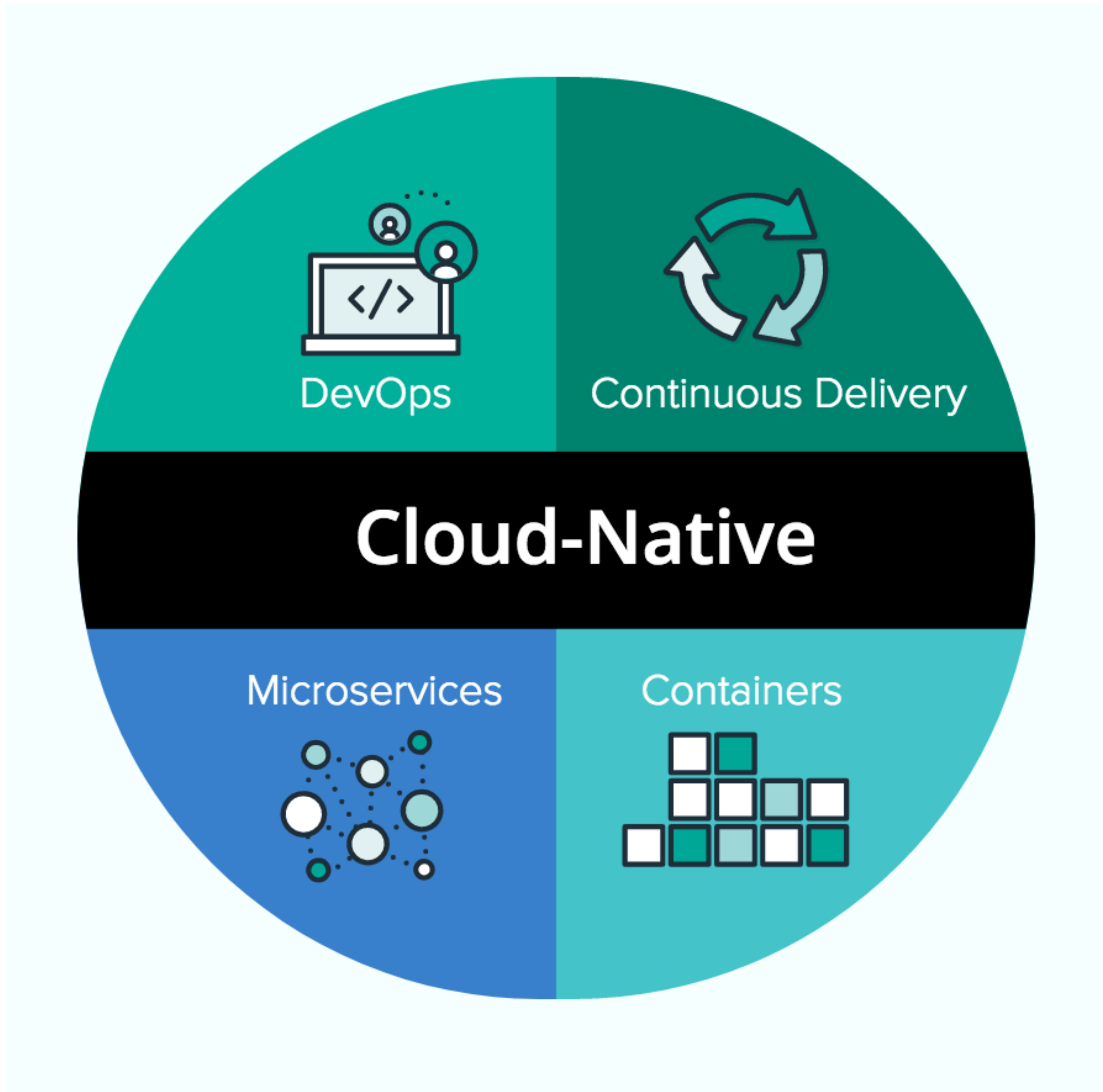
Treat logs as event streams

XII. Admin processes

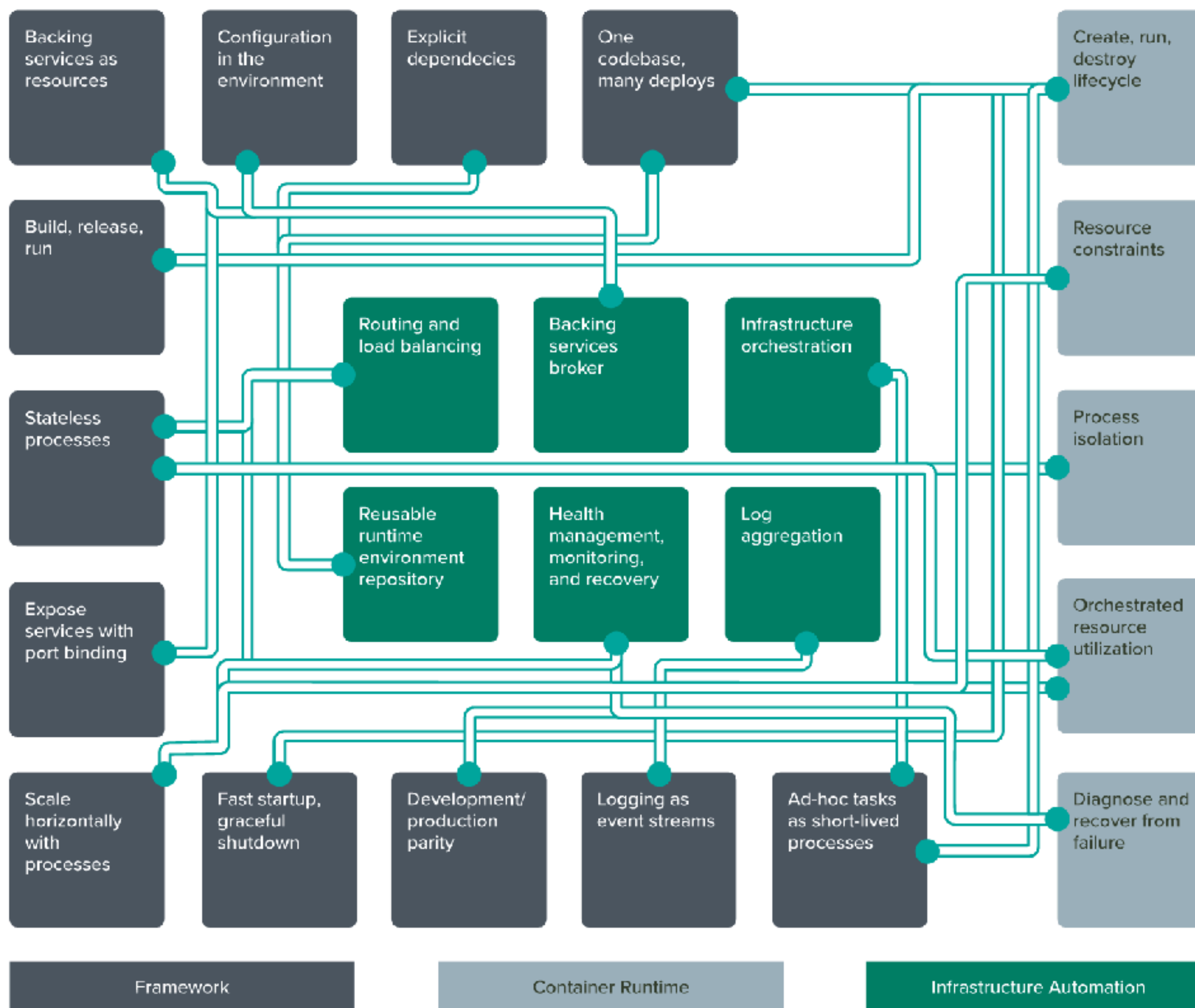
Run admin/management tasks as one-off processes

11 MINUTE READ

Cloud-Native Applications: Ship Faster, Reduce Risk, Grow Your Business



The Cloud-Native Architecture: What You Need To Be Cloud-Native



ThoughtWorks®

CLOUD NATIVE ORGANIZATION

~

NATIVE



A young child with dark hair is looking down at a smartphone held in their hands. The child is wearing a dark-colored top. The background is a warm, yellowish glow. On the left side of the image, there are several overlapping teal and light blue circular shapes of varying sizes, creating a graphic overlay. The text 'MOBILE NATIVE' is centered over the child's face.

MOBILE NATIVE



CLOUD NATIVE

ThoughtWorks®

云原生的价值

~

why every enterprise loves cloud?

simplicity

flexibility

elasticity

velocity

def accumulate(simplicity, flexibility, elasticity, velocity): Any



return responsiveness

The Benefits

By running its applications on AWS, 3M HIS can easily scale up or down without needing to acquire and provision servers months in advance. “The server-provisioning process used to take us 10 weeks or more, but it’s down to minutes using AWS.” says Austin. “That gives us the flexibility to support the scaling events we experience several times per day. We pay for what we need when we need it. We no longer need to pay for and manage unnecessary peak compute capacity. We recently had a customer request a large volume of processing, and we simply spun up an additional 200 instances of our service on AWS. When the large workload completed, we just turned off those instances.”

ThoughtWorks®

PRINCIPLES

~

Code

Clean code

DDD

Microservices

SBA...

and then...

Cloud

Software as a Service: Strategic Backgrounder

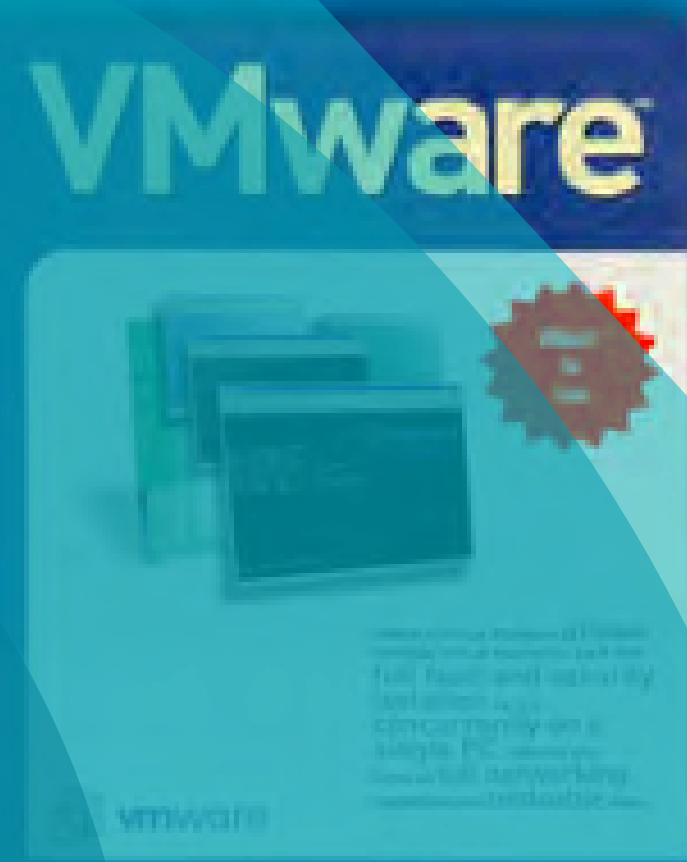
February 2001



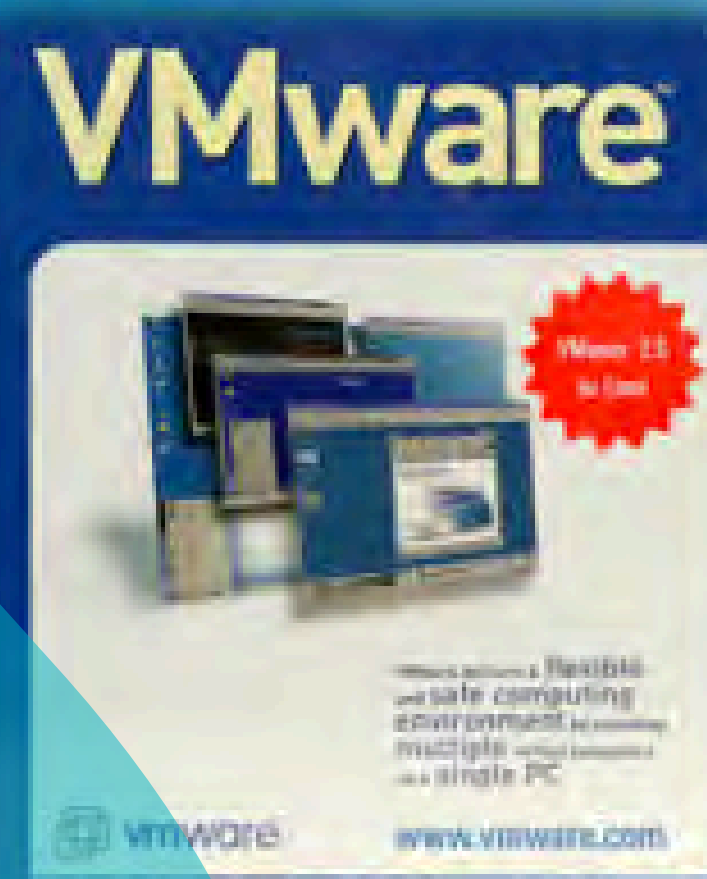
Software & Information Industry Association
1730 M Street NW Suite 700
Washington, DC 20036
+1.202.452.1600
www.siiia.net

© Copyright 2000, Software & Information Industry Association
All rights reserved. Reproduction prohibited without permission of SIIA.

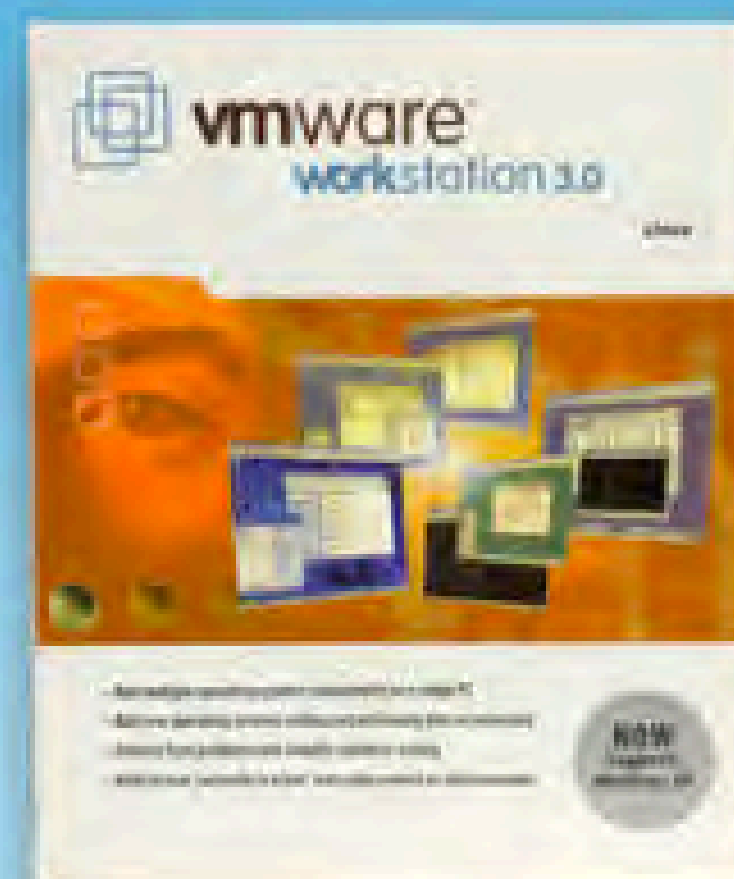
VMware Workstation Major Product Release Timeline.



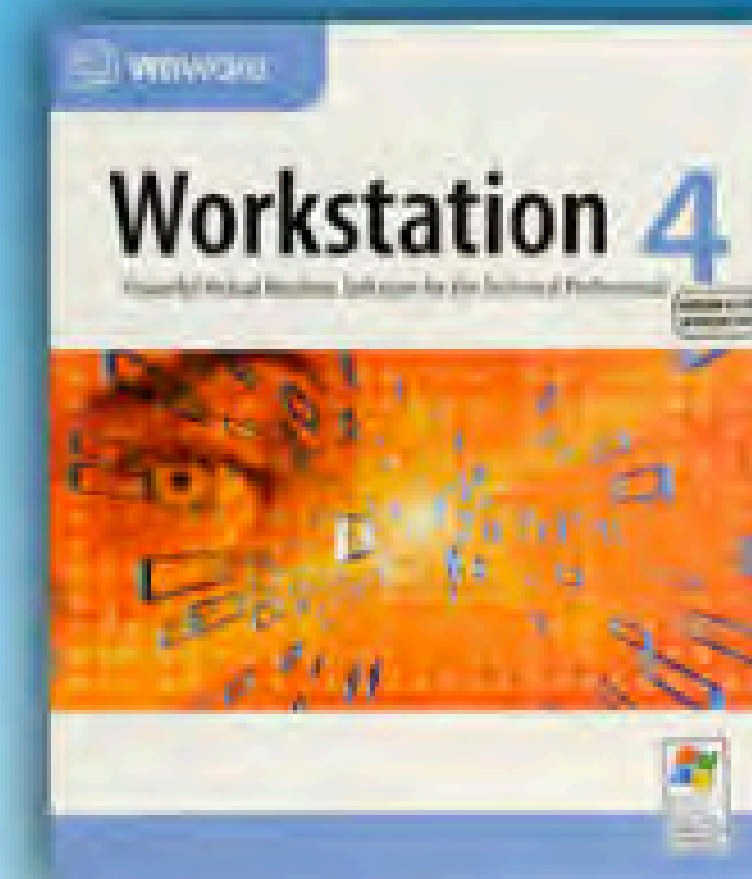
1999



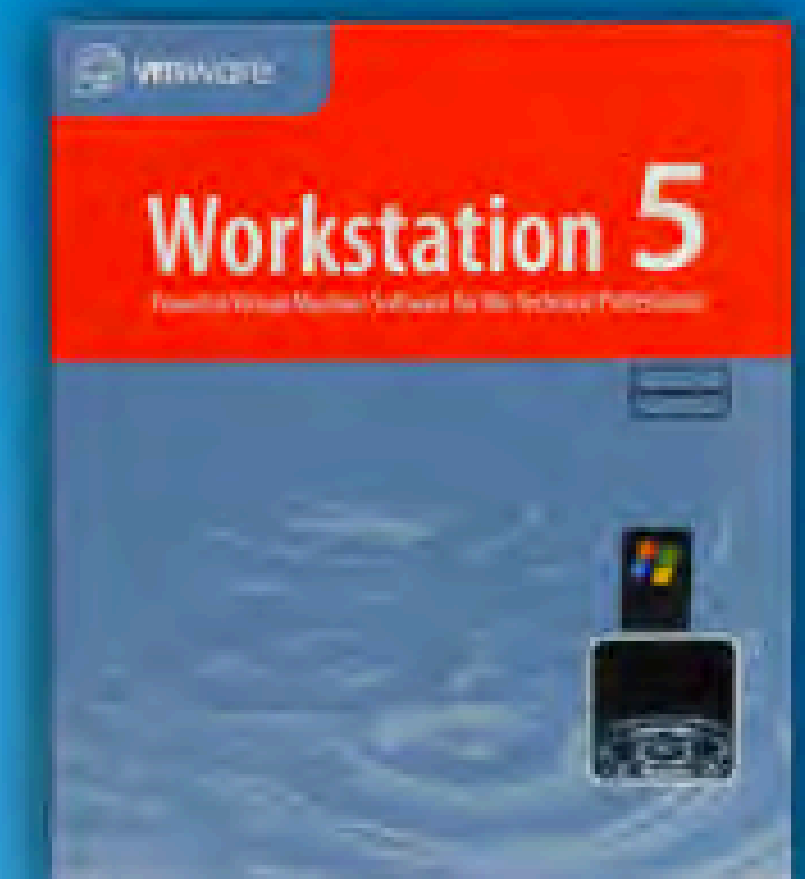
2000



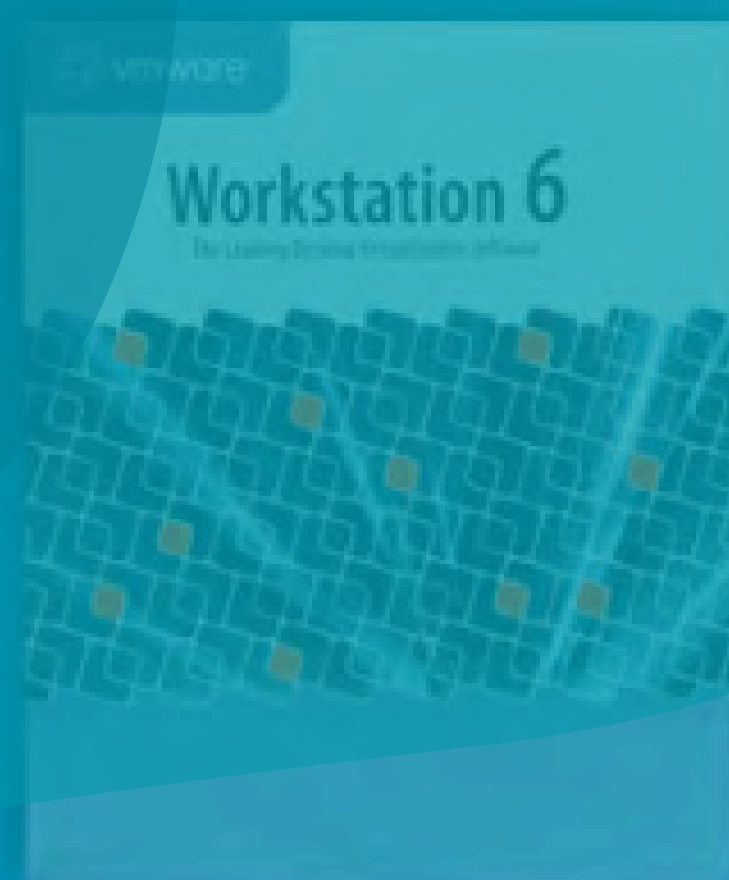
2001



2003



2005



2007



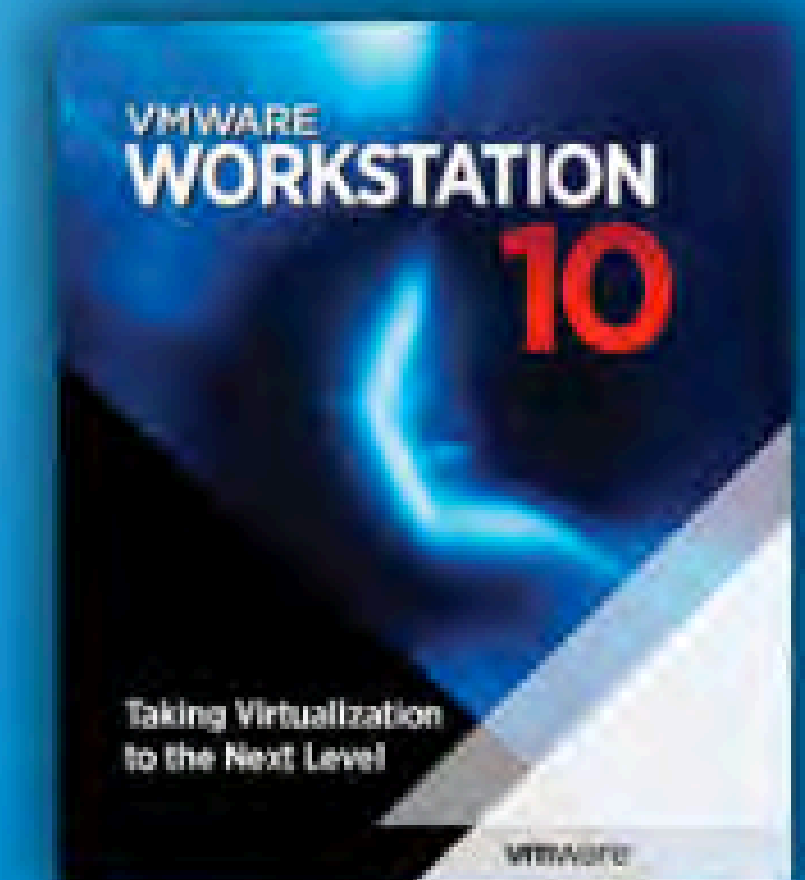
2009



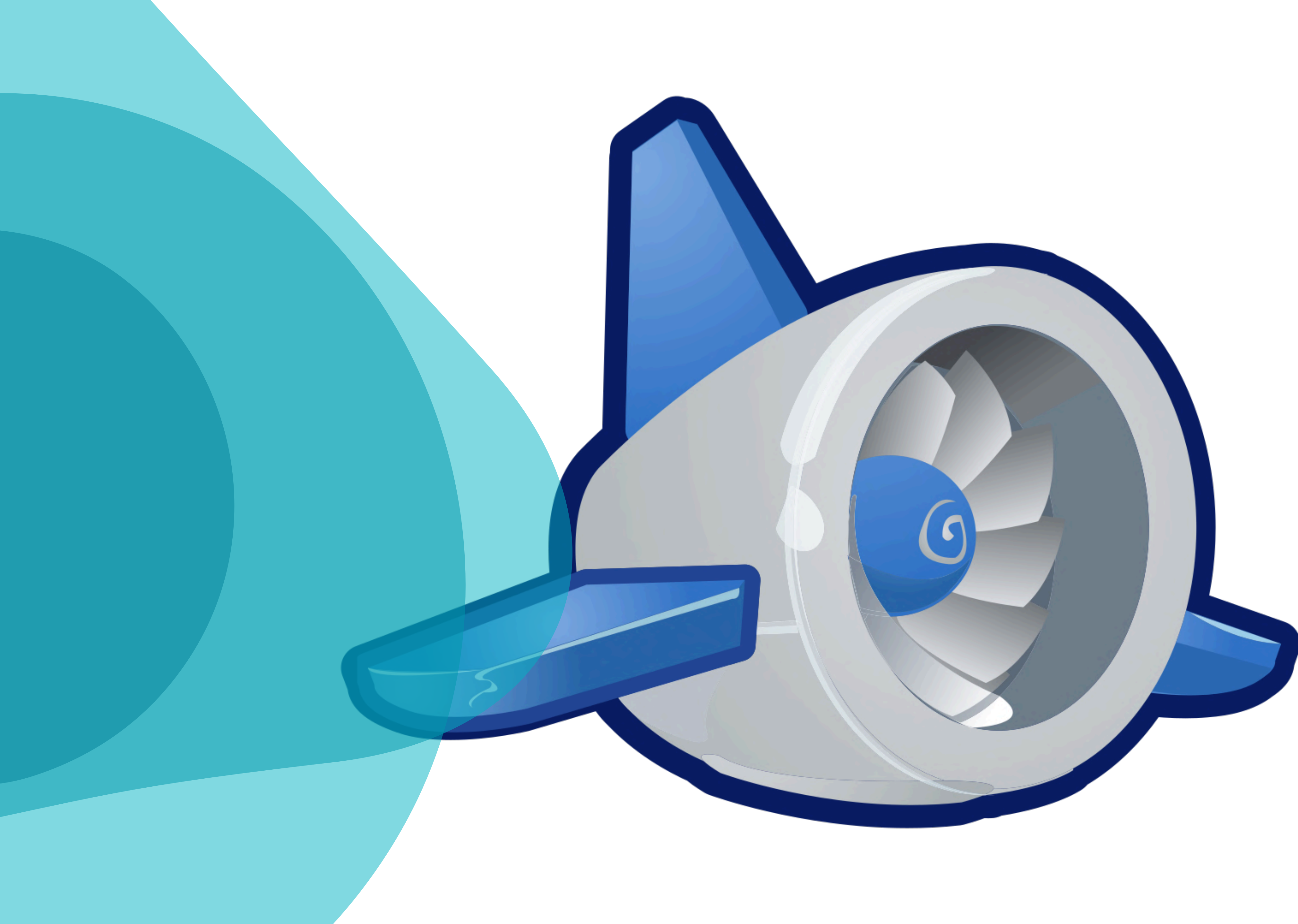
2011



2012



2013



IaaS

PaaS

SaaS

CaaS

BaaS

AaaS

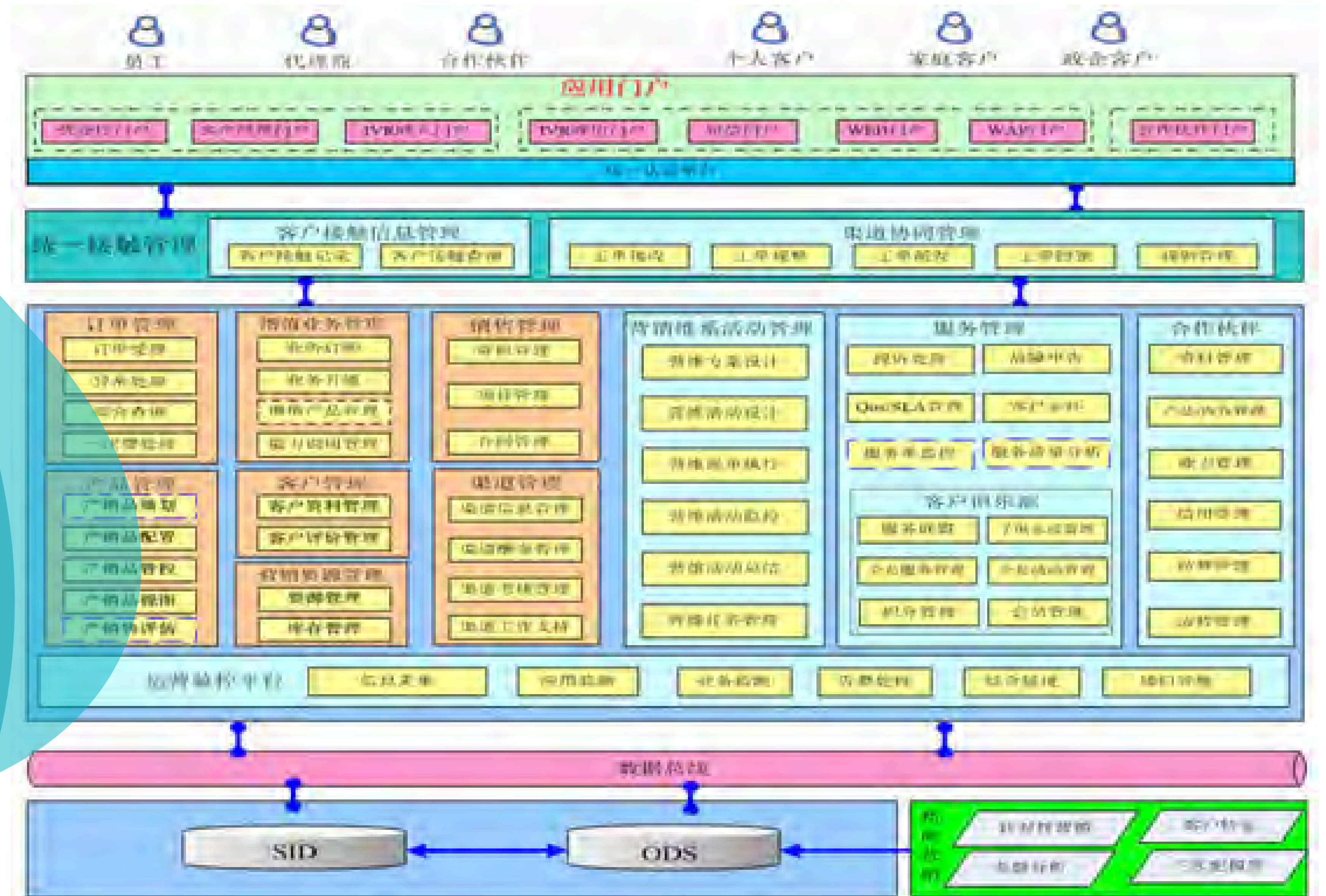
Virtualization

abstract



abstract in different level

you must recall this ...



it's not abstract, it's just a crab

abstract in production level

```
apiVersion: v1
kind: Pod
metadata:
  name: frontend
spec:
  containers:
  - name: db
    image: mysql
    resources:
      requests:
        memory: "64Mi"
        cpu: "250m"
      limits:
        memory: "128Mi"
        cpu: "500m"
  - name: wp
    image: wordpress
    resources:
      requests:
        memory: "64Mi"
        cpu: "250m"
      limits:
        memory: "128Mi"
        cpu: "500m"
```

```
version: '3'
services:
  redis:
    image: redis:alpine
    deploy:
      resources:
        limits:
          cpus: '0.001'
          memory: 50M
        reservations:
          cpus: '0.0001'
          memory: 20M
```


ubiquitous language

*cloud native system should implement the
abstraction around **production** itself*

*cloud native system should implement the
abstraction around **business outcome***



Melvin Conway



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

— Melvin Conway, 1968

*cloud native **organization** should
implement the abstraction and
decomposition around **business outcome***

How?

ThoughtWorks®

PATTERNS

~

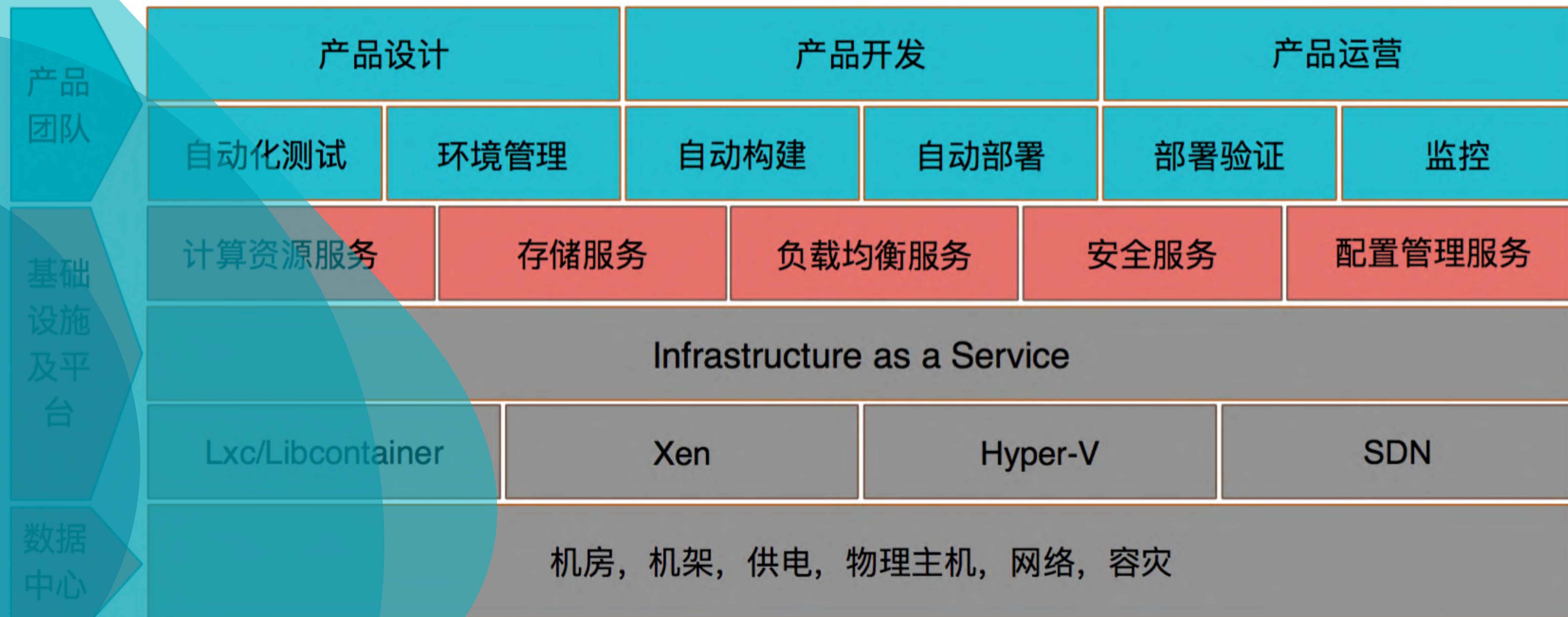
P1: INFRASTRUCTURE INDEPENDENT

migrate from AWS to Google Cloud or Azure?

build local environment

cloud native == independent from cloud

in organization perspective...



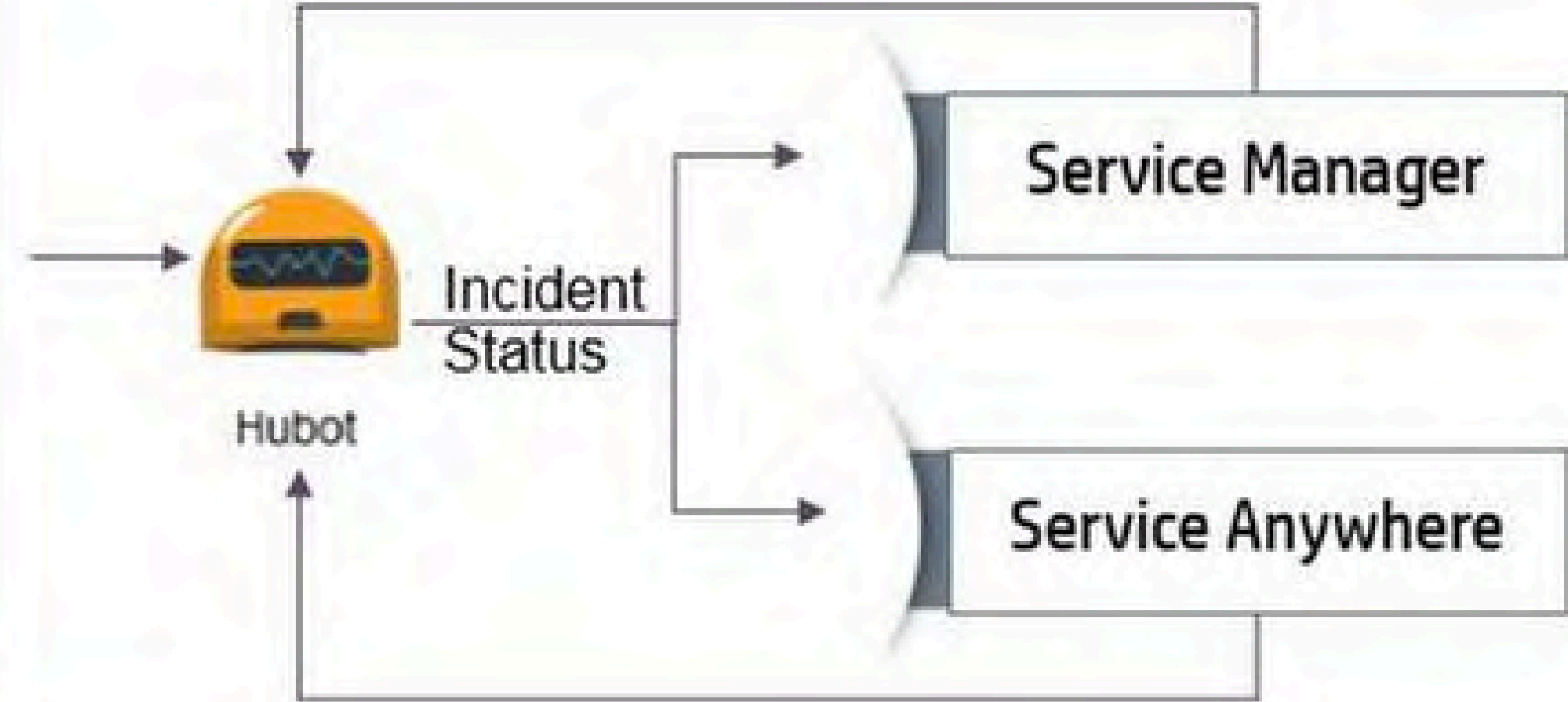
亚马逊没有传统意义上的“运维”部门，“运维”就是开发和运营其云平台的团队，提供从资源到安全的各种公共服务。但为每个产品创建、配置和权限管理这些服务都给了产品开发团队，以自助的方式完成。

每个产品从设计、开发到运维都是由一个团队负责，这样就很容易对每一个产品的交付运营成本及价值做出衡量。

chatops


Hewlett Packard
Enterprise

ChatOps with HPE Service manager / Service Anywhere

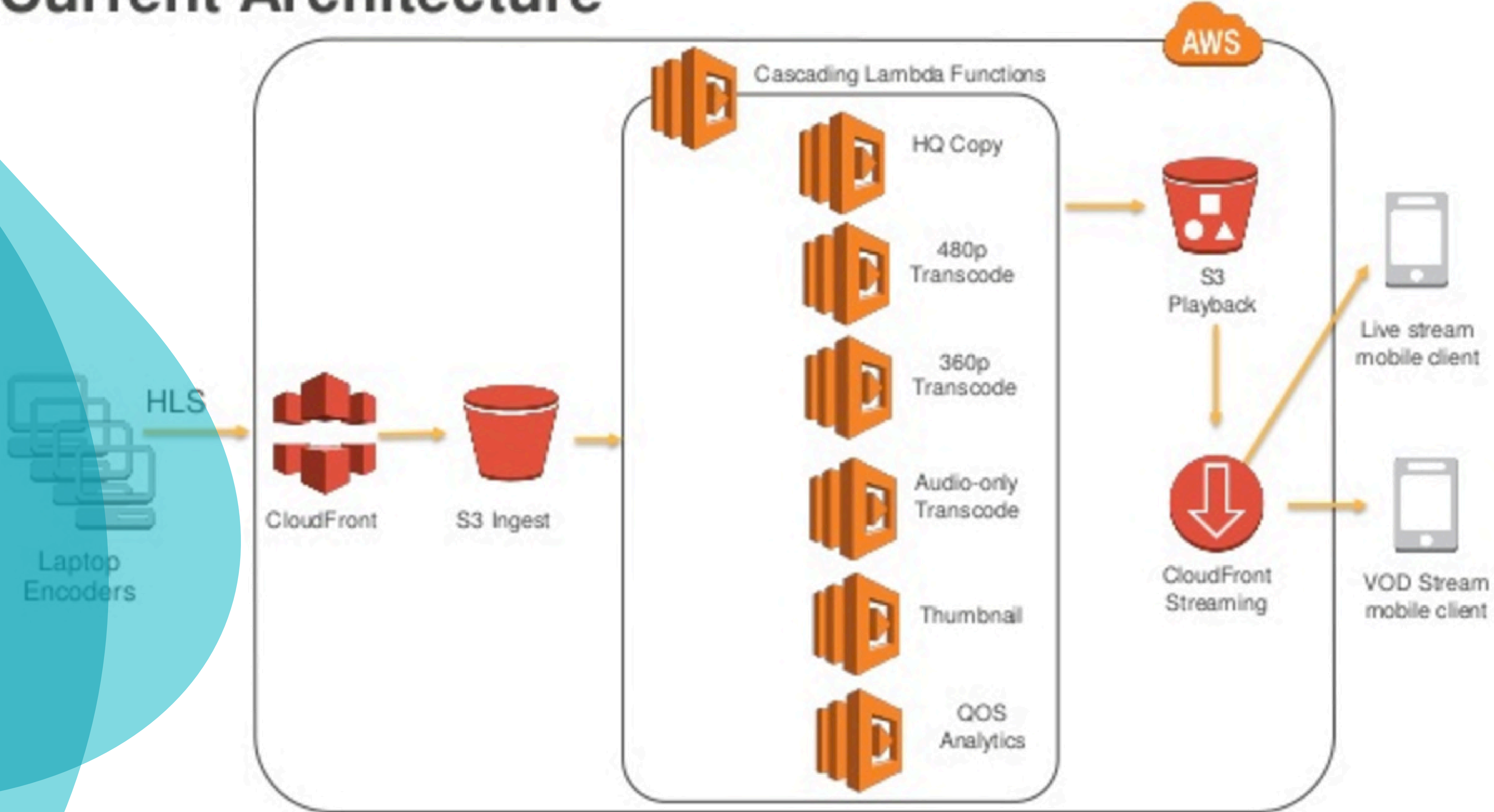


no ops...

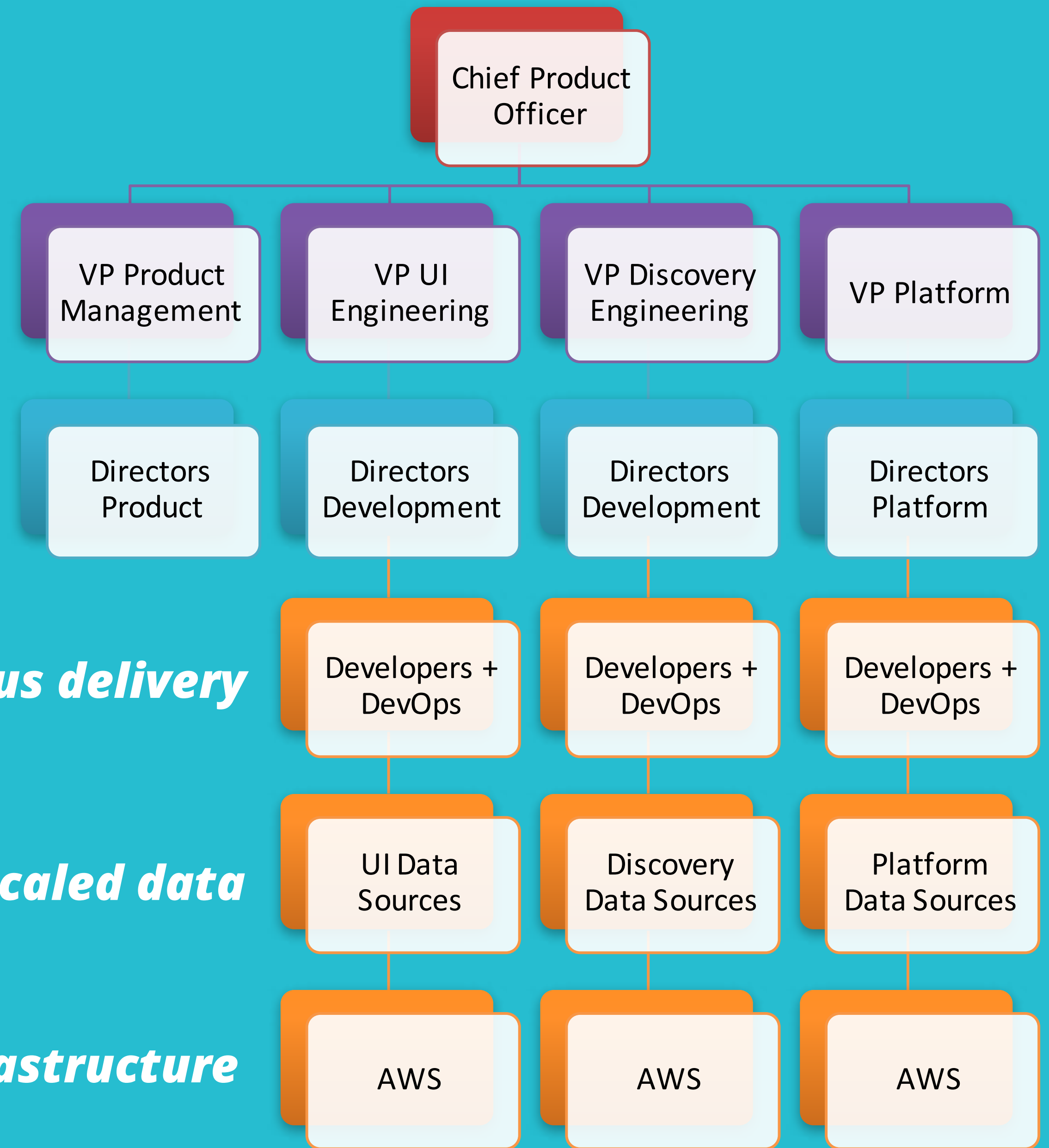
P2: CLOUD FIRST

cloud as the default infrastructure

Current Architecture



organization around the cloud



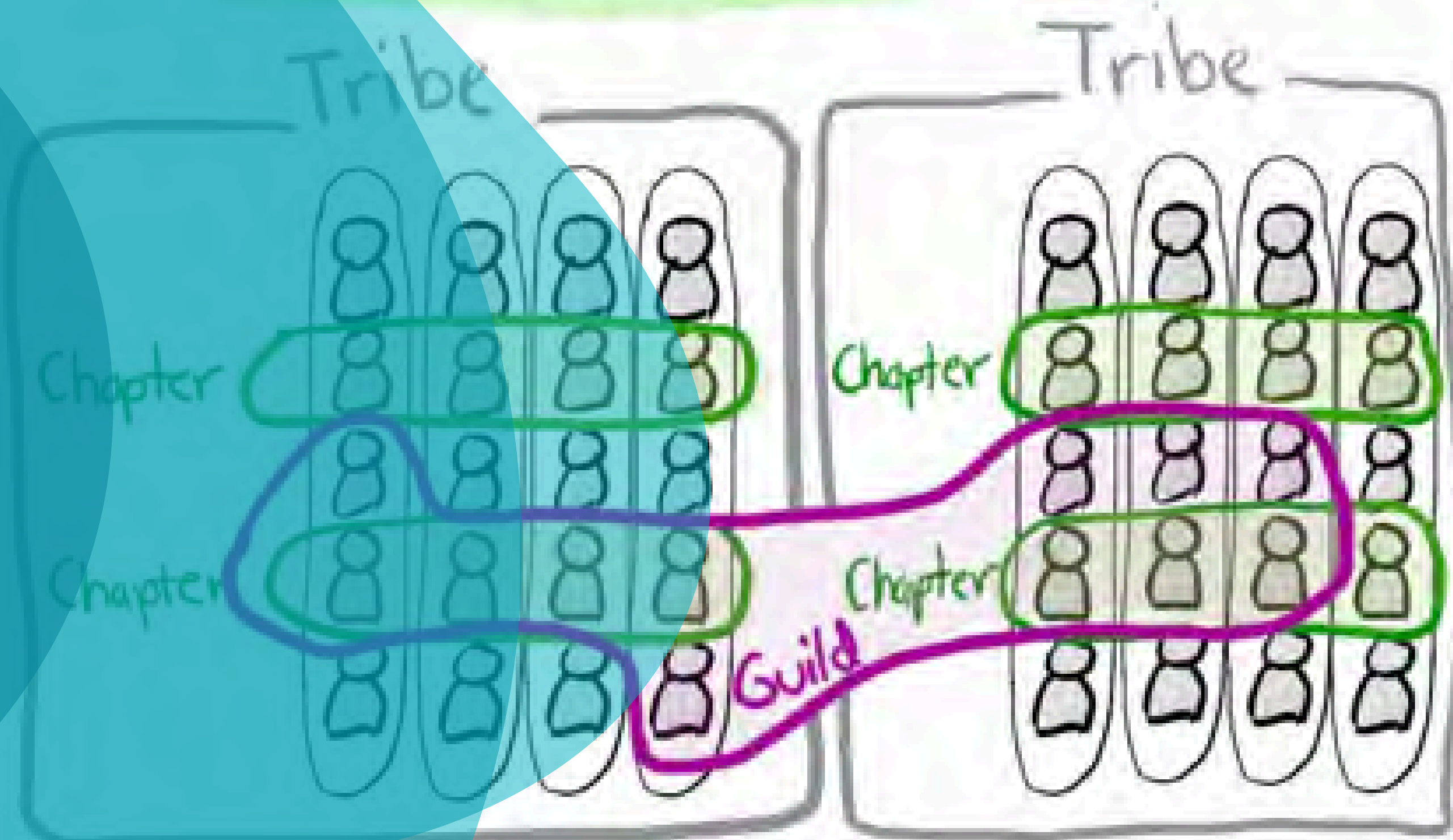
Code, independently updated continuous delivery

Denormalized, independently updated and scaled data

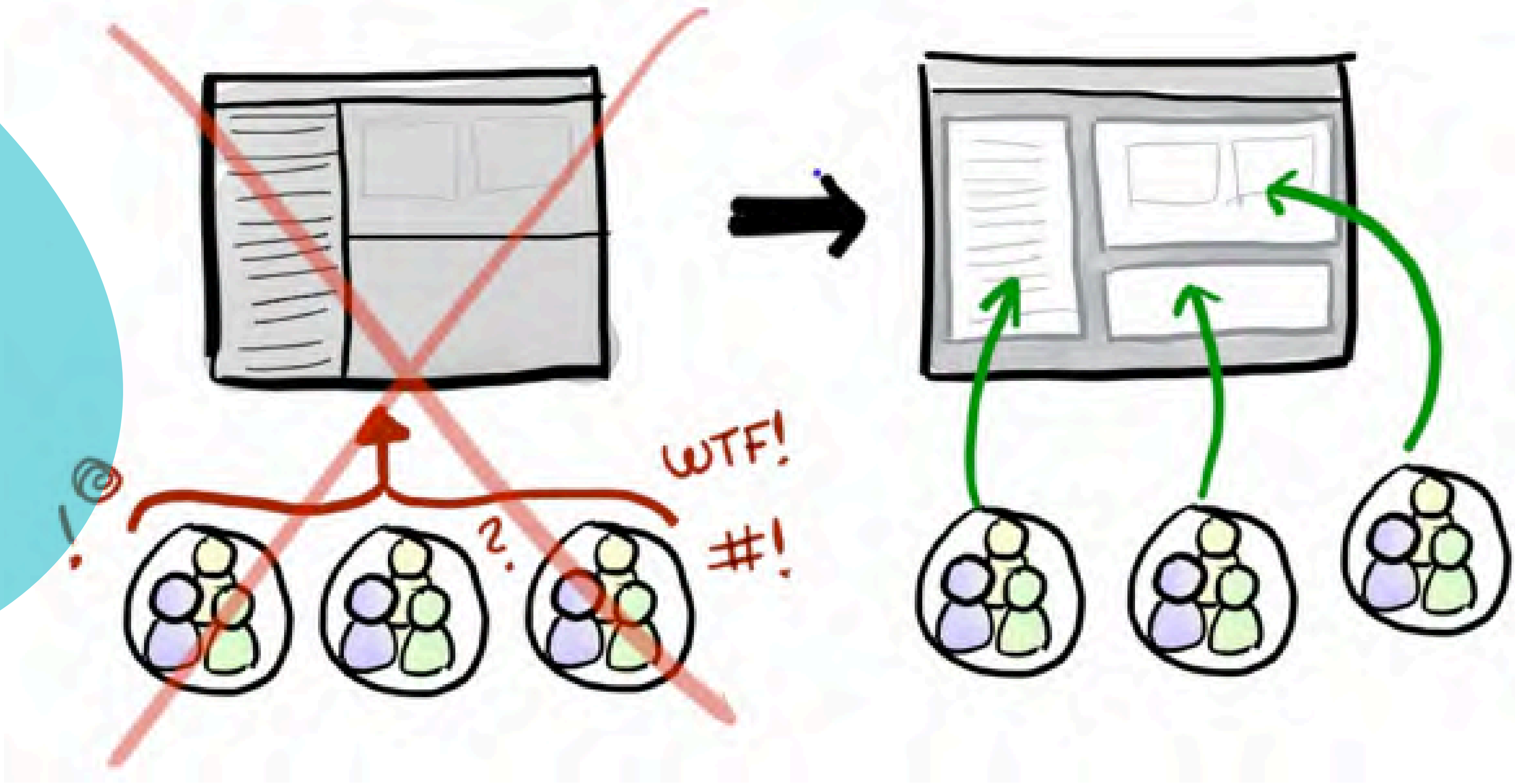
Cloud, self service updated & scaled infrastructure

P3: MICROSERVICES ORIENTED

Community > Structure

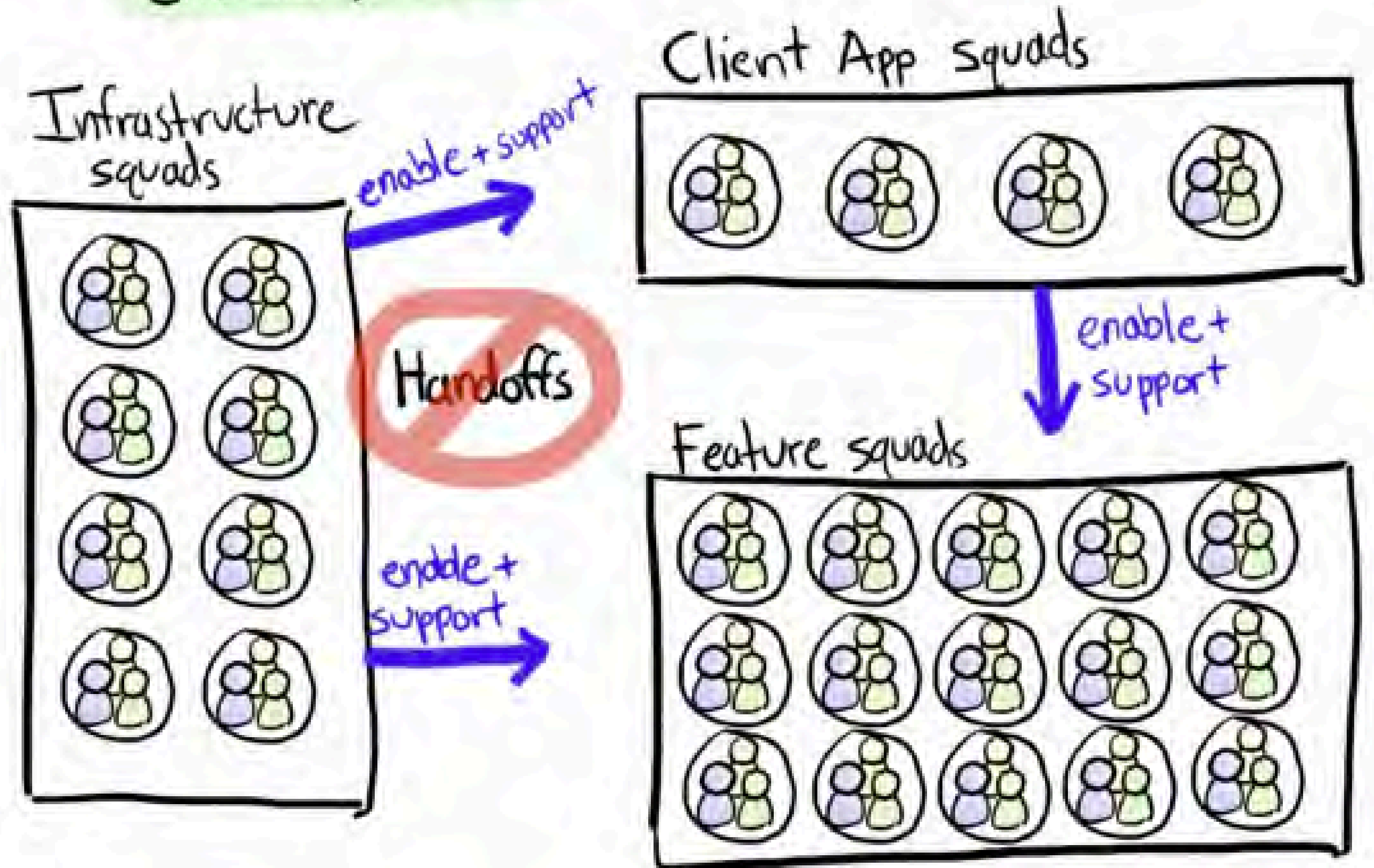


Decoupled releases



Self-service model

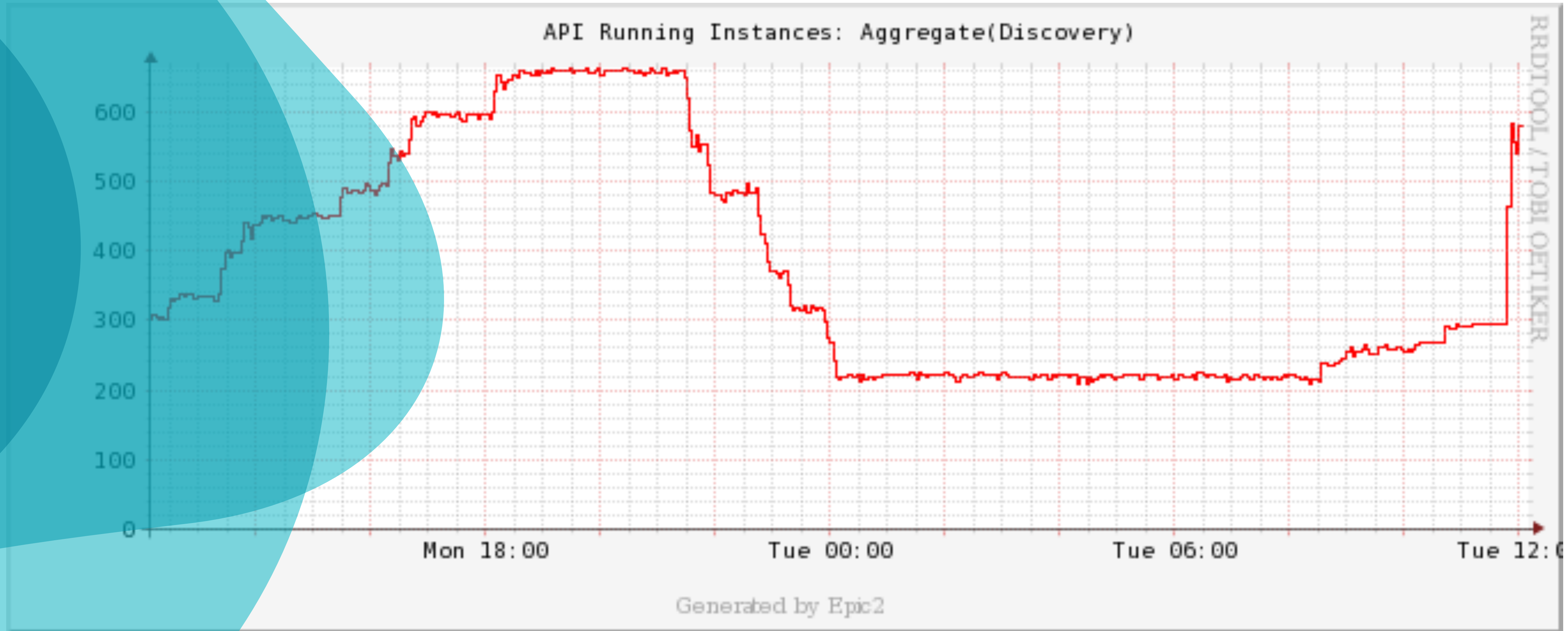
enable > serve



P4: MAX CLOUD ROI



***Largest services are autoscaled
Average lifetime of an instance is 36 hours***



P5: OPEN ECOSYSTEM

api market



open source ecosystem

P6: BUILD SECURITY IN

内建安全应用开发

Build **SecurityIn**[®]

内建安全的应用开发是一系列安全原则、最佳安全实践以及安全工具的综合体，它使得企业或者团队在保证交付速度的同时，开发出具备更高安全质量的应用、服务。



尽早识别安全需求、尽早获取安全反馈



充分利用自动化进行安全测试



将安全实践融入到持续交付流程中



共同承担安全职责

INFRASTRUCTURE INDEPENDENT

CLOUD FIRST

MICROSERVICES ORIENTED

MAX CLOUD ROI

OPEN ECOSYSTEM

BUILD SECURITY IN

ThoughtWorks®

PRACTICES

~

1. architecture

abstract in different level

INFRASTRUCTURE INDEPENDENT

CLOUD FIRST

MICROSERVICES ORIENTED

MAX CLOUD ROI

OPEN ECOSYSTEM

BUILD SECURITY IN

cross functional: performance / security

infrastructure: iac

collaboration: interfaces

component: microservices

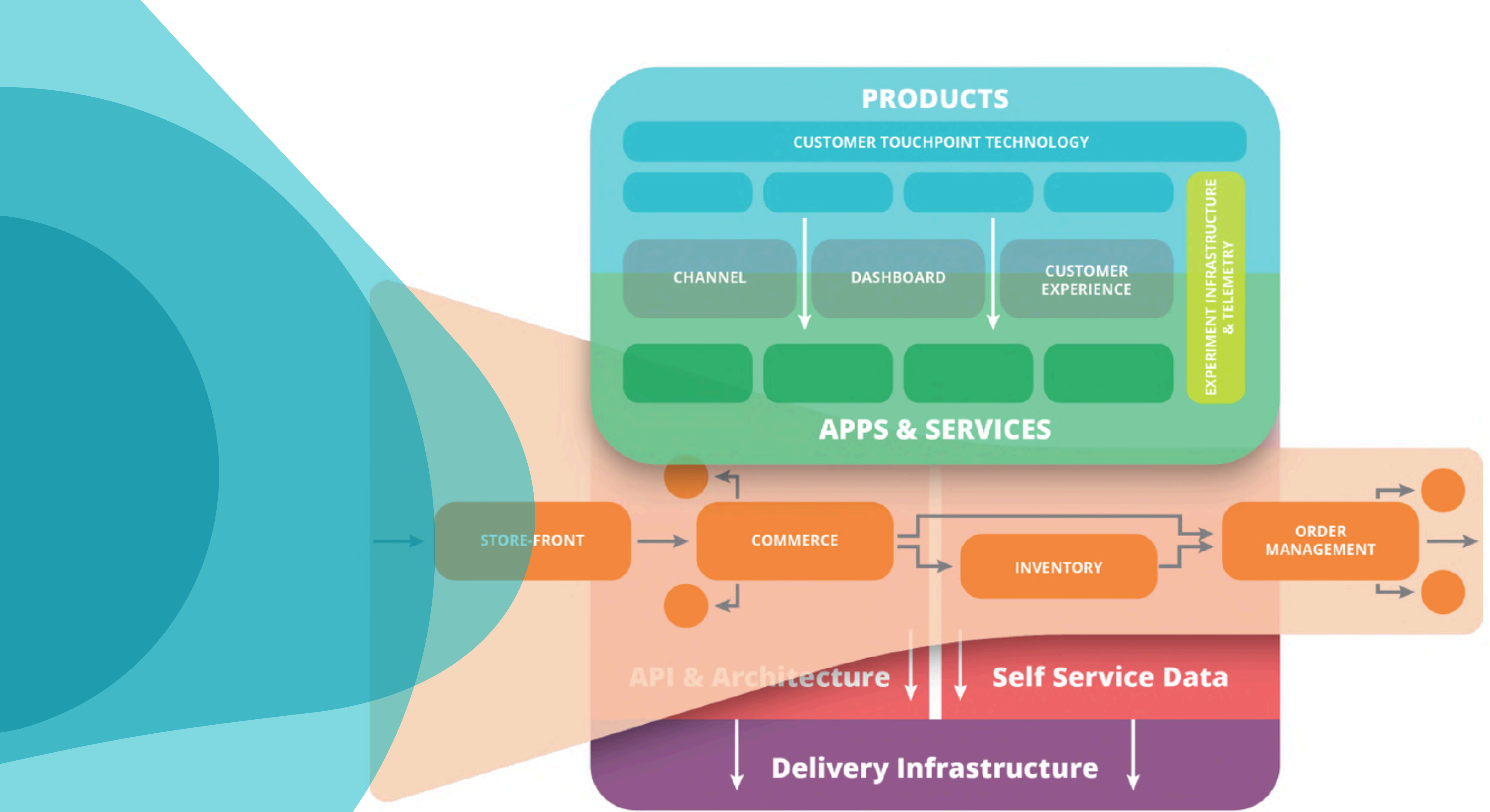
logic modeling: object / function / clean code

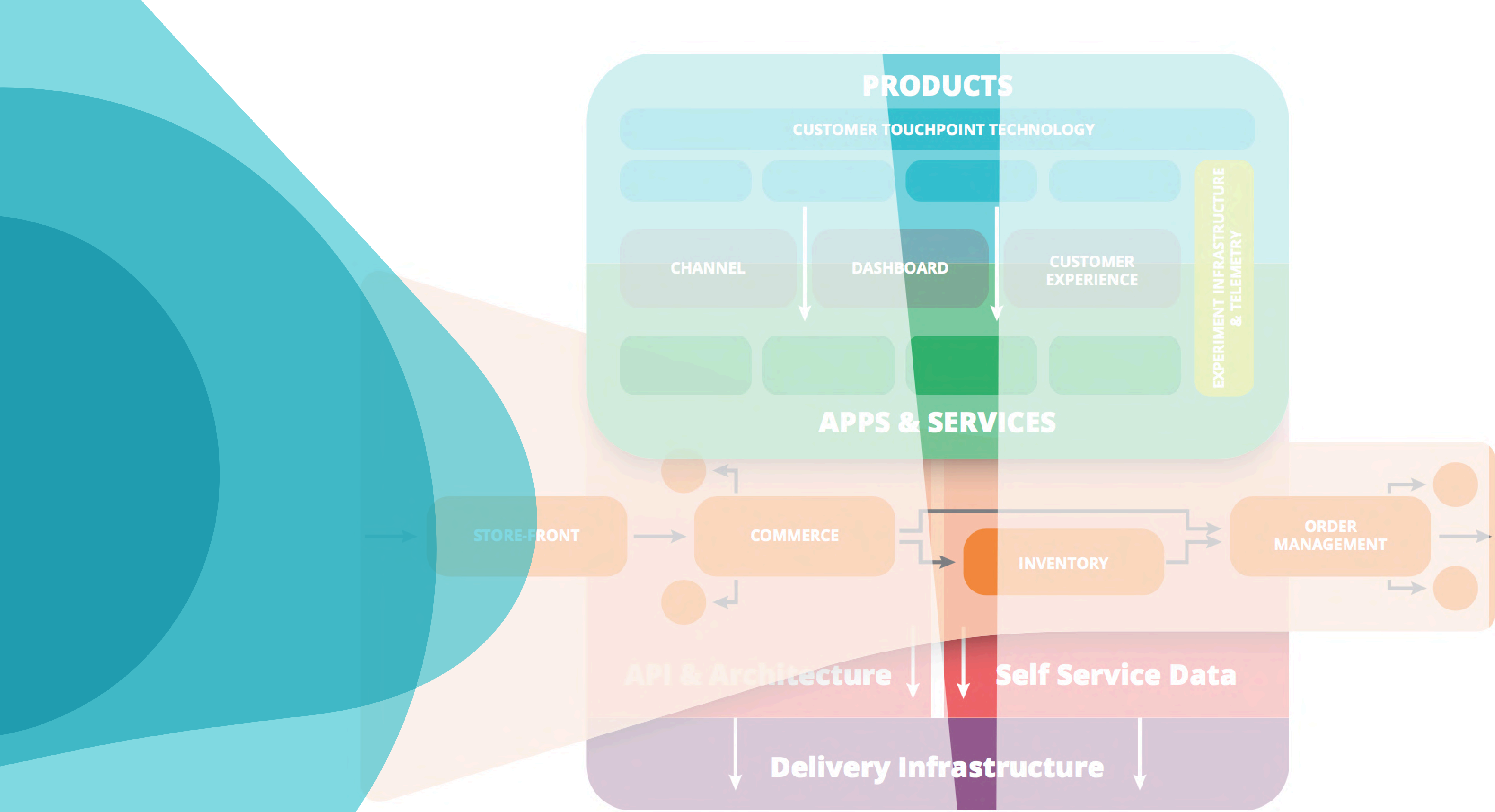
business modeling: DDD

product modeling: service design

build API ecosystem, and make API as product

build platform, starting with a slice





2. Organization

Conway's Law

INFRASTRUCTURE INDEPENDENT

CLOUD FIRST

MICROSERVICES ORIENTED

MAX CLOUD ROI

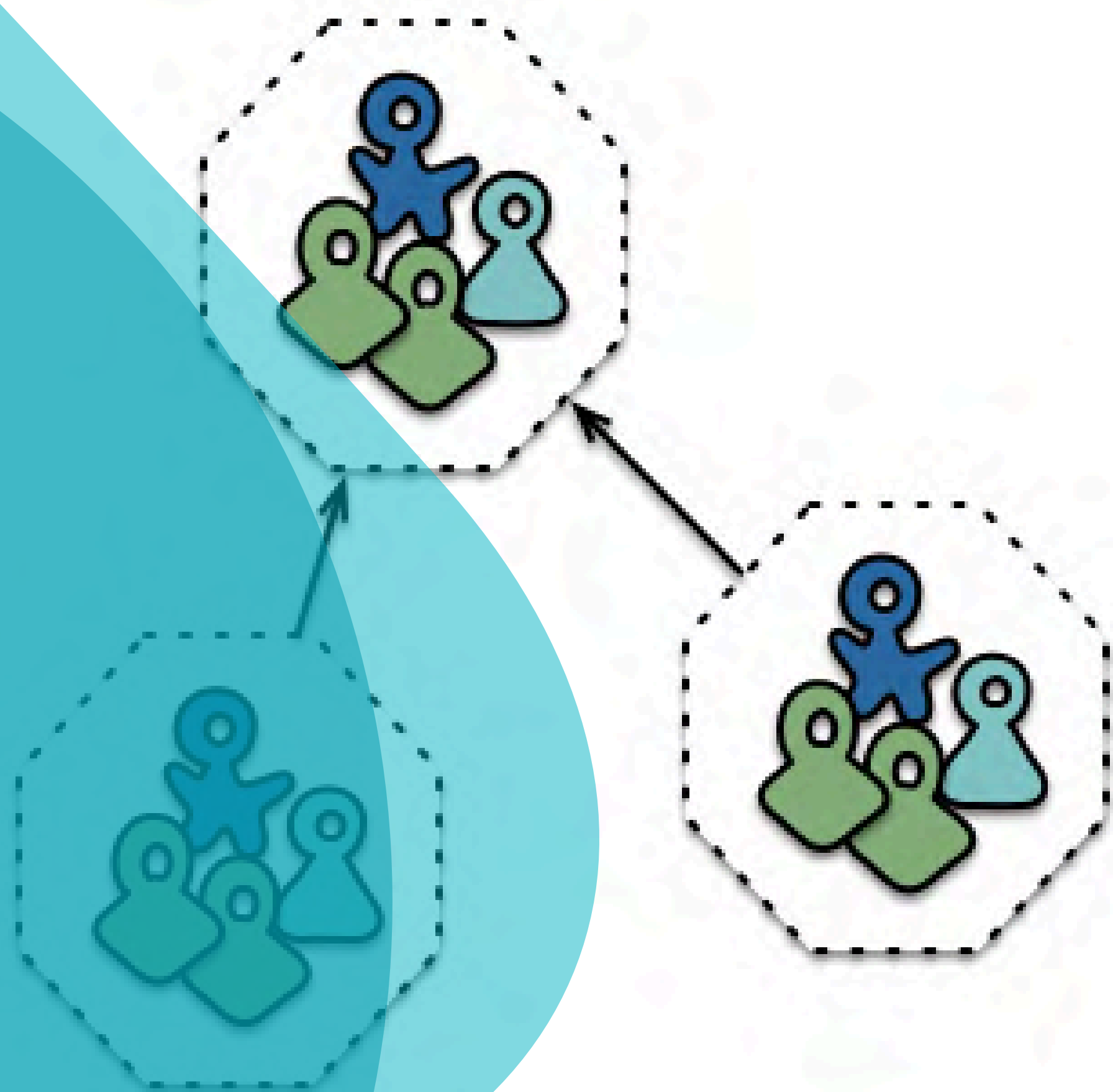
OPEN ECOSYSTEM

BUILD SECURITY IN

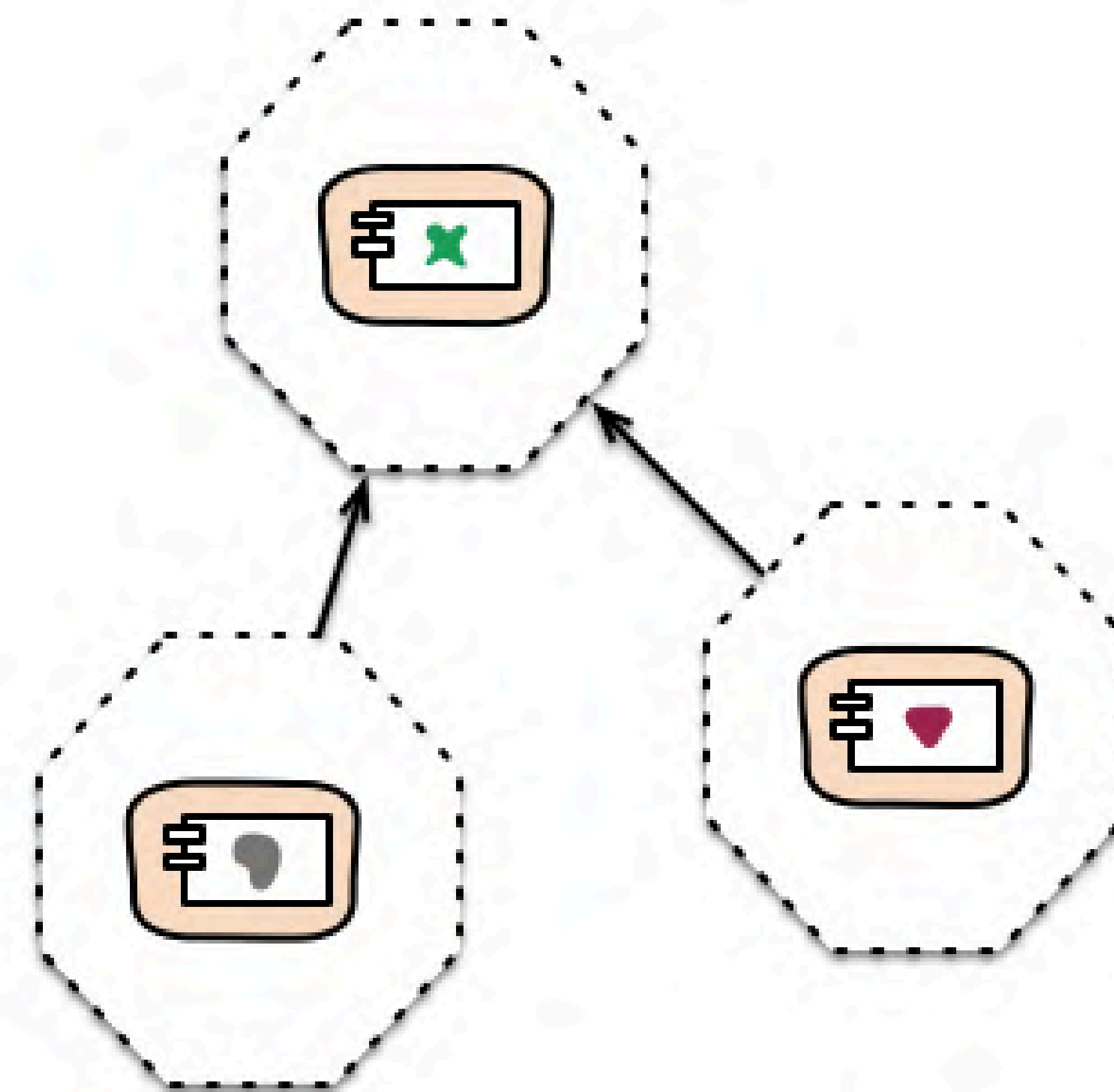
what we have mentioned...

infrastructure independent product teams

and microservices oriented



Cross-functional teams...



**... organised around capabilities
Because Conway's Law**

investment on cloud and measure the ROI

VALUE DRIVEN PORTFOLIO MANAGEMENT (“EDGE”)

Technology driven innovation is a creative journey. In a rapidly changing and uncertain marketplace, it requires a strong shared business vision as a guide. EDGE uses the Lean Value Tree to capture and share this vision.

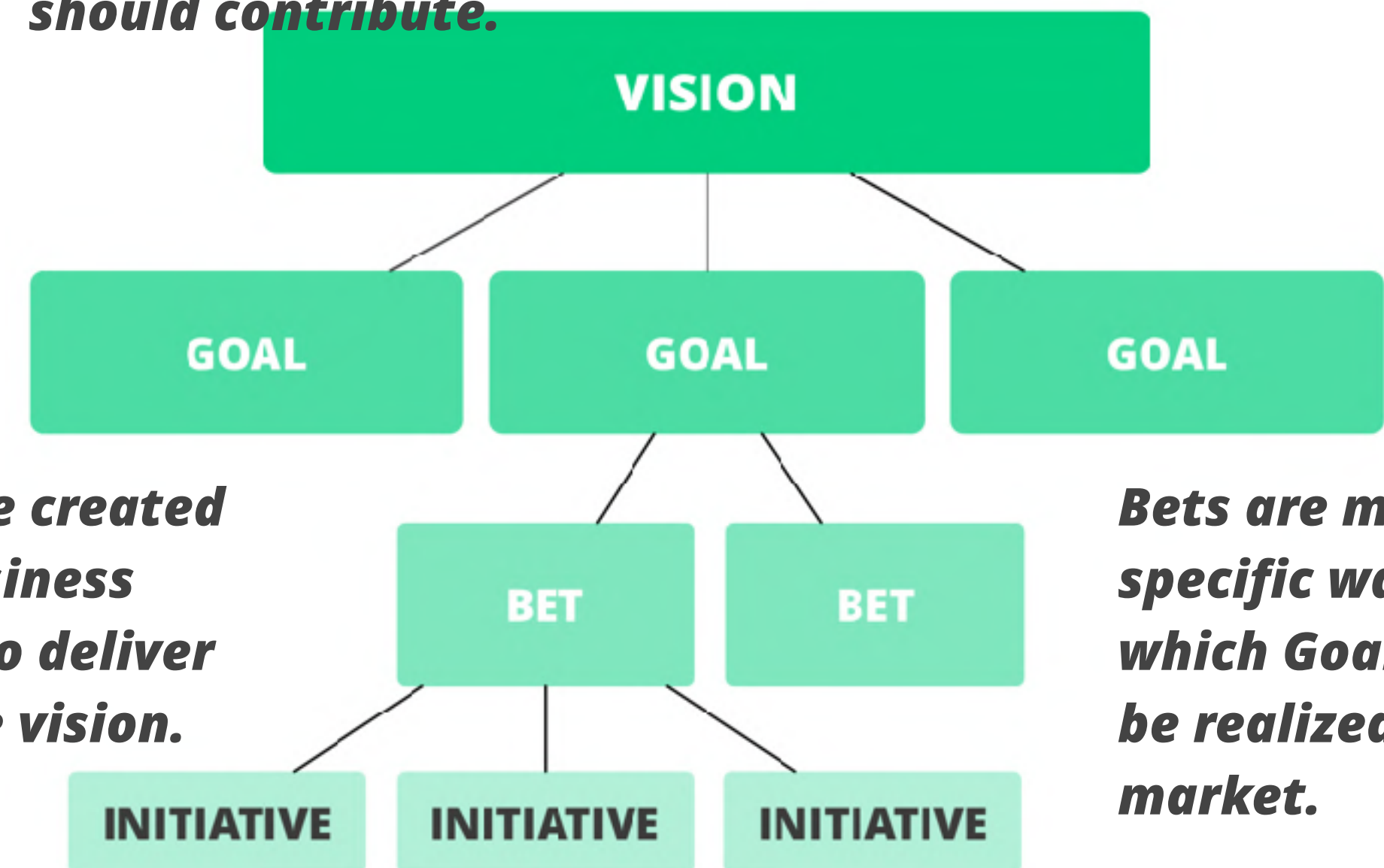
It's not enough to simply deliver projects from a list. The best opportunities are emerging all the time. Investment decisions must be continuously aligned with evolving business priorities and on the ground delivery teams must be able to learn and adjust or pivot as they go.

This is only possible if the Executive Business Vision is clearly understood and can be applied throughout an organization. It must be easy to see how work ties back to strategic direction and priorities.

A tool to facilitate capturing and sharing the organizational Vision and Strategy is the Lean Value Tree (LVT). It is a tree because everything stems from the Executive Business Vision. Everything in the tree is framed in terms of outcomes, so it is clear the value that they will provide to the organization.

Example: Lean Value Tree is a visual tool to facilitate capturing and sharing an organizational vision and strategy.

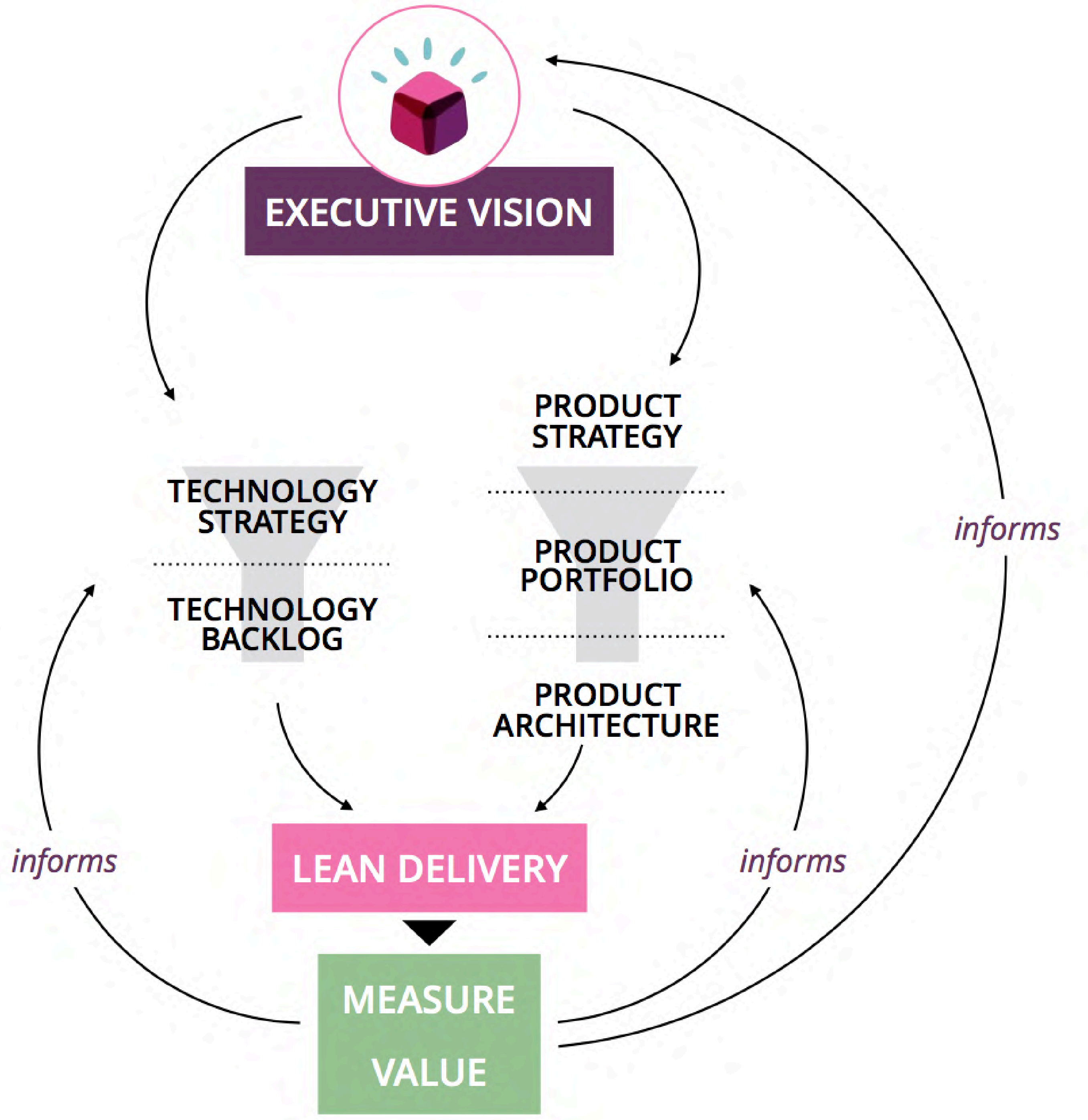
Vision: the overall vision at the top of the tree sets a guiding direction toward which all investments should contribute.



Goals are created from business drivers to deliver upon the vision.

Bets are more specific ways in which Goals might be realized in the market.

Initiatives are actions that deliver tangible value. Initiatives have a clear customer need and business opportunity defined.



BUILD YOUR OWN TECHNOLOGY RADAR

Technology radar is a great format to visualize the technology portfolio in a certain enterprise.

BYOR is stand for "Build Your Own Radar" which encourage enterprises to visualize their technique portfolio in a radar format

Build your own technology radar

Building reliable digital operations

Making decisions with an algorithmic rule engine

Ryan Alexander

THE RADAR

TECHNIQUES

ADOPT

1. Consumer-driven contract testing
2. Pipelines as code
3. Threat Modeling

TRIAL

4. APIs as a product
5. Bug bounties
6. Data Lake
7. Hosting PII data in the EU
8. Lightweight Architecture Decision Records
9. Reactive architectures
10. Serverless architecture

ASSESS

11. Client-directed query
12. Container security scanning
13. Consent Security Policies
14. Differential privacy
15. Micro frontends
16. OWASP ASVS
17. Unikernels
18. VR beyond gaming

HOLD

19. A single CI instance for all teams
20. Anemic REST
21. Big Data envy
22. Cloud lift and shift

PLATFORMS

ADOPT

23. Docker
24. HSTS
25. Linux security modules

TRIAL

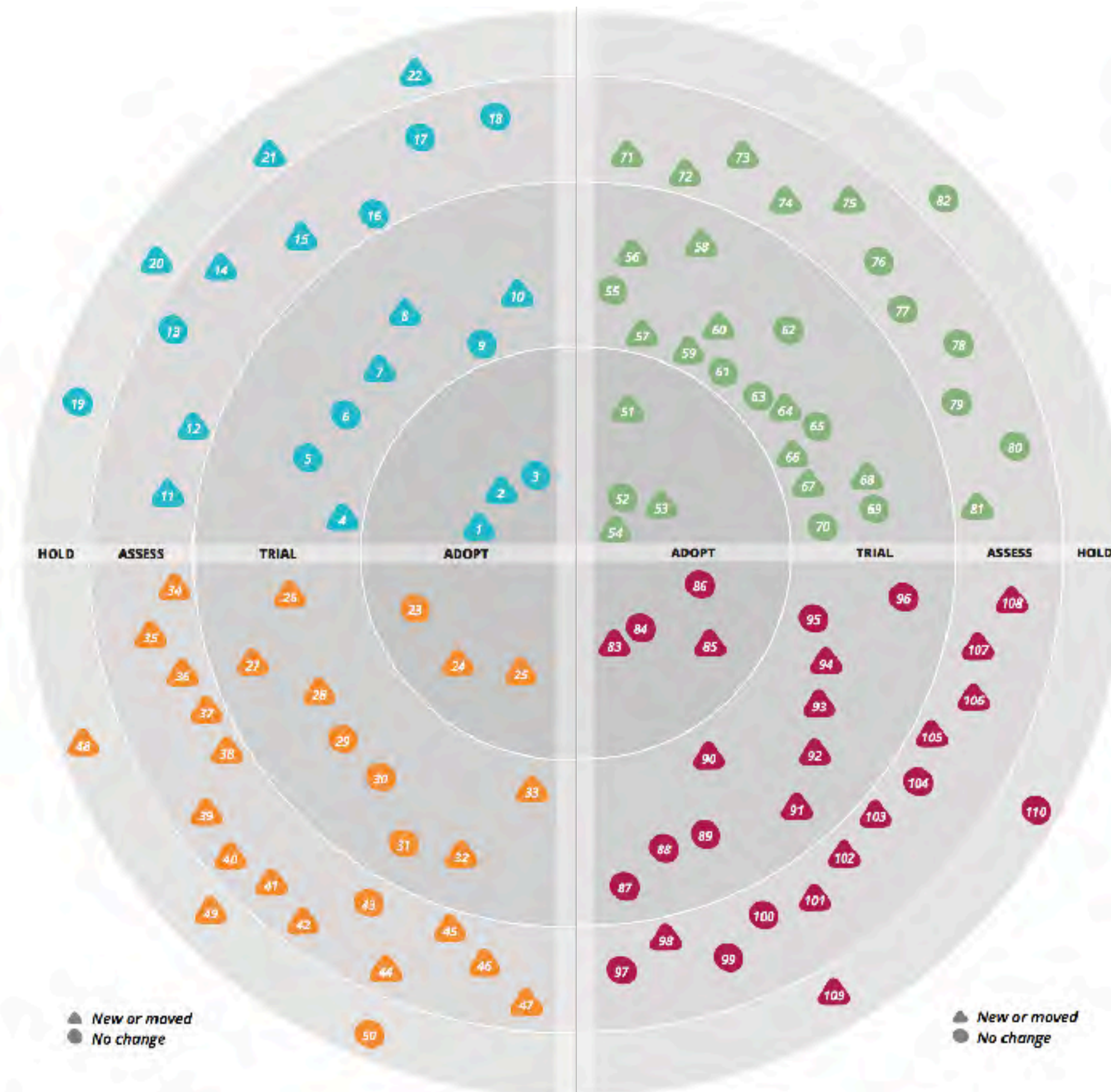
26. Apache Mesos
27. AuthN
28. AWS Lambda
29. Kubernetes
30. Pivotal Cloud Foundry
31. Rancher
32. Reagent
33. Unity beyond gaming

ASSESS

34. .NET Core
35. Amazon API Gateway
36. Apache Flink
37. AWS Application Load Balancer
38. Cassandra carefully
39. Elixir
40. Ethereum
41. HoloLens
42. ImpisStack
43. Nomad
44. Nuance Mix
45. OpenVR
46. TaroTool
47. xit.ai

HOLD

48. CMS as a platform
49. Overambitious API gateway
50. Superficial private cloud



THE RADAR

TOOLS

ADOPT

51. Babel
52. Consul
53. Grafana
54. Packer

TRIAL

55. Apache Kafka
56. Express
57. Istio
58. Galen
59. HashiCorp Vault
60. SOXServer
61. Let's Encrypt
62. Load Impact
63. OWASP Dependency-Check
64. PalTy
65. SonarSpoc
66. Tailsman
67. Terraform
68. tmate
69. Webpack
70. Zipkin

ASSESS

71. Android-x86
72. arius
73. Bottled Water
74. Clojure.soc
75. F5SnoShot/Testcase
76. Grosp
77. LambdaCD
78. Pinpoint
79. Riser
80. Repsheet
81. SkillTeam

HOLD

82. Jenkins as a deployment pipeline

LANGUAGES & FRAMEWORKS

ADOPT

83. Ember.js
84. React.js
85. Redux
86. Spring Boot

TRIAL

87. Butterknife
88. Dagger
89. Dagger
90. Dixir
91. Enzyme
92. Immutable.js
93. Phoenix
94. Quick and Nimble
95. React Native
96. Robolectric

ASSESS

97. Aurelia
98. ECMAScript 2017
99. Elm
100. GraphQL
101. JUMP
102. Physical Web
103. Rapidoid
104. Richards
105. ReSwift
106. Three.js
107. Vue.js
108. WebRTC

HOLD

109. AngularJS
110. jQuery



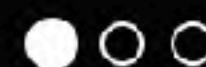
open source ecosystem

JAX 2015 Award Industry Awards!

Netflix is honored to receive the Jury's choice award for Innovation at [JAX 2015 conference](#).

We would like to thank all of those who contribute to the Netflix open source community including our Netflix developers, all external contributors, and our active user base.

Netflix Open Source won the JAX Special Jury Award. Jury member Neal Ford was quoted as saying "that architecture is cool again, that it can be used as a business differentiator, and when done right it is a huge advantage. Netflix showed the power of internalizing DevOps into their architecture; all architectures will do this in the future."



Netflix Open Source Software Center

Netflix is committed to open source. Netflix both leverages and provides open source technology focused on providing the leading Internet television network. Our technology focuses on providing immersive experiences across all internet-connected screens. Netflix's deployment technology allows for continuous build and integration into our worldwide deployments serving



SoundCloud

Berlin, Germany <http://backstage.soundcloud.com>

Repositories

People 37

Search repositories...

Type: All

Language: All

lightcycle

LightCycle lets self-contained classes respond to Android's lifecycle events

Java ★ 670 🍴 45 Updated 4 days ago



sketchy-core

A framework for reducing text-based spam and other malicious user activity. Battle-tested in production at SoundCloud.

Scala ★ 233 🍴 15 Updated 10 days ago



lhm

Online MySQL schema migrations

Ruby ★ 1,417 🍴 140 Updated 18 days ago

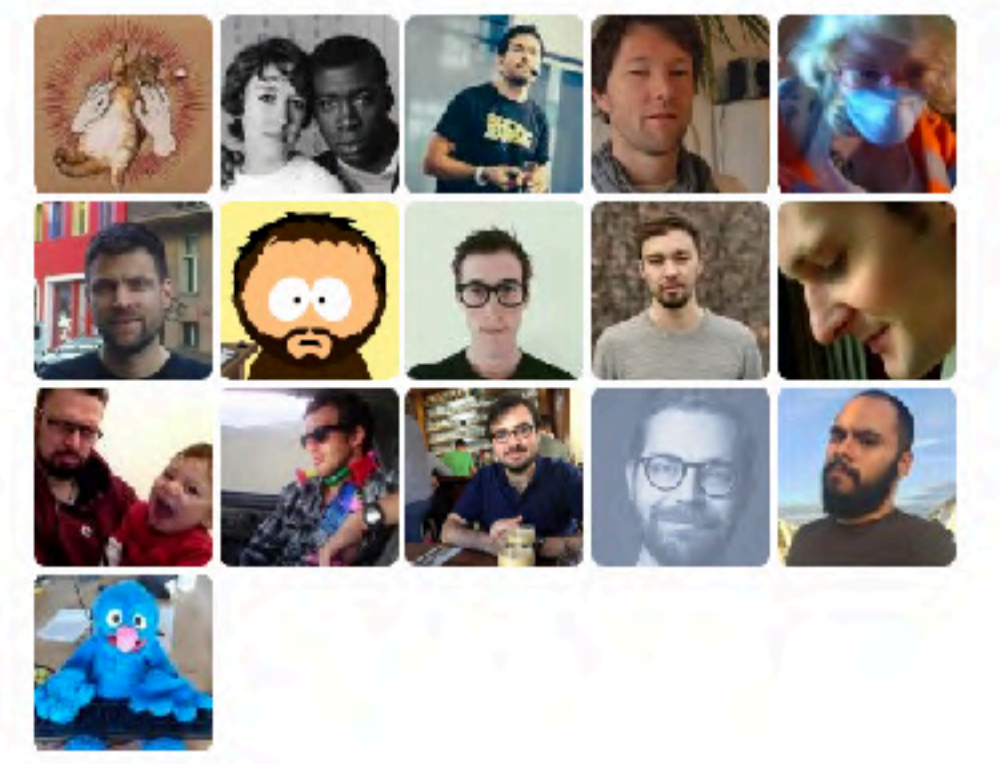


Top languages

- Ruby
- JavaScript
- Java
- Go
- Objective-C

People

37 >



KEY TAKEAWAYS

~

*cloud native system should implement the
abstraction around **business outcome***

*cloud native **organization** should
implement the abstraction and
decomposition around **business outcome***

INFRASTRUCTURE INDEPENDENT

CLOUD FIRST

MICROSERVICES ORIENTED

MAX CLOUD ROI

OPEN ECOSYSTEM

BUILD SECURITY IN

value focused, outcome over output

ThoughtWorks®

Q&A

~

THANK YOU

*Questions or feedback:
David Wang: davidw@thoughtworks.com*



ThoughtWorks®