MA-01

Reduce your operational burden to deploy containers

Andreas Lindh (he/him)
Specialist Solutions Architect, Containers
Amazon Web Services

Alberto López (he/him)
Director of Engineering
Acast

Container infrastructure challenges

Challenge
Run containers

Containers infrastructure plumbing

Containers infrastructure plumbing

Containers infrastructure plumbing

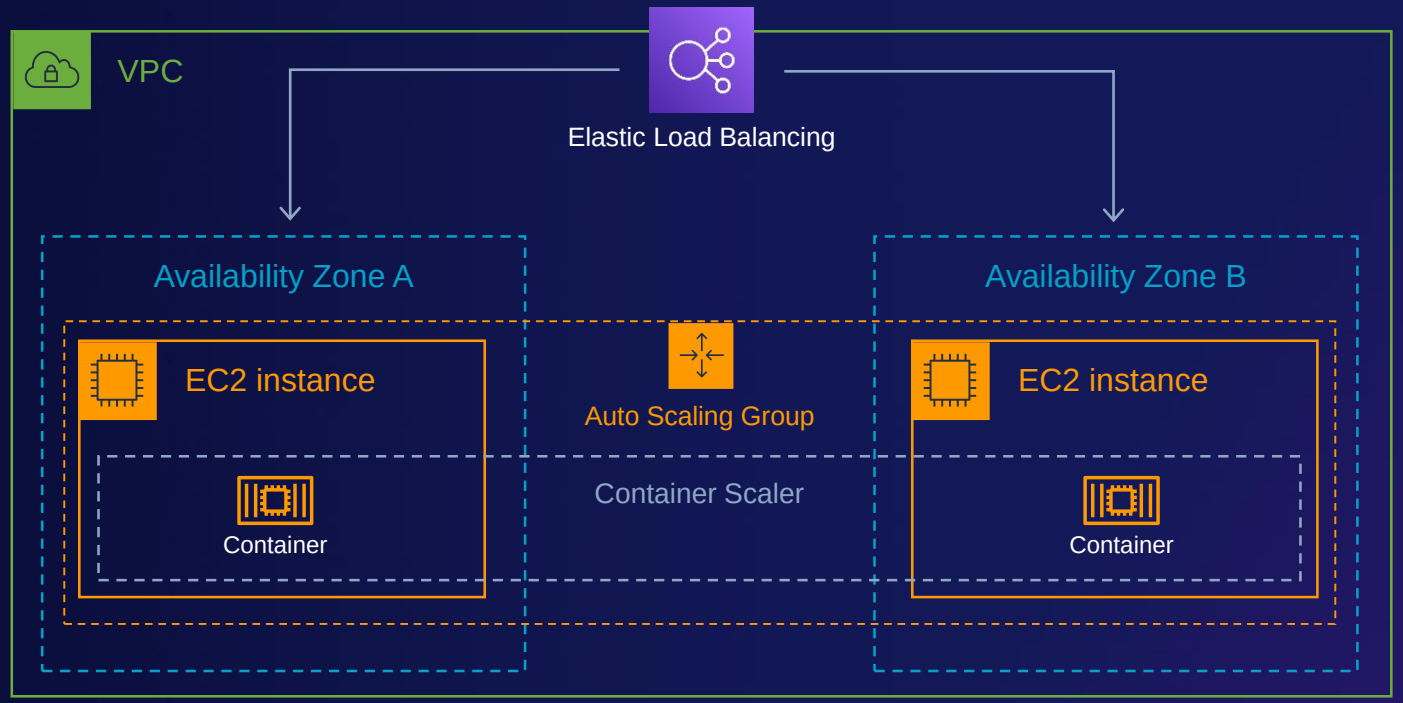
Challenge
Run many containers

Containers infrastructure plumbing

Amazon Elastic Container Service (Amazon ECS)

Amazon Elastic Kubernetes Service (Amazon EKS)

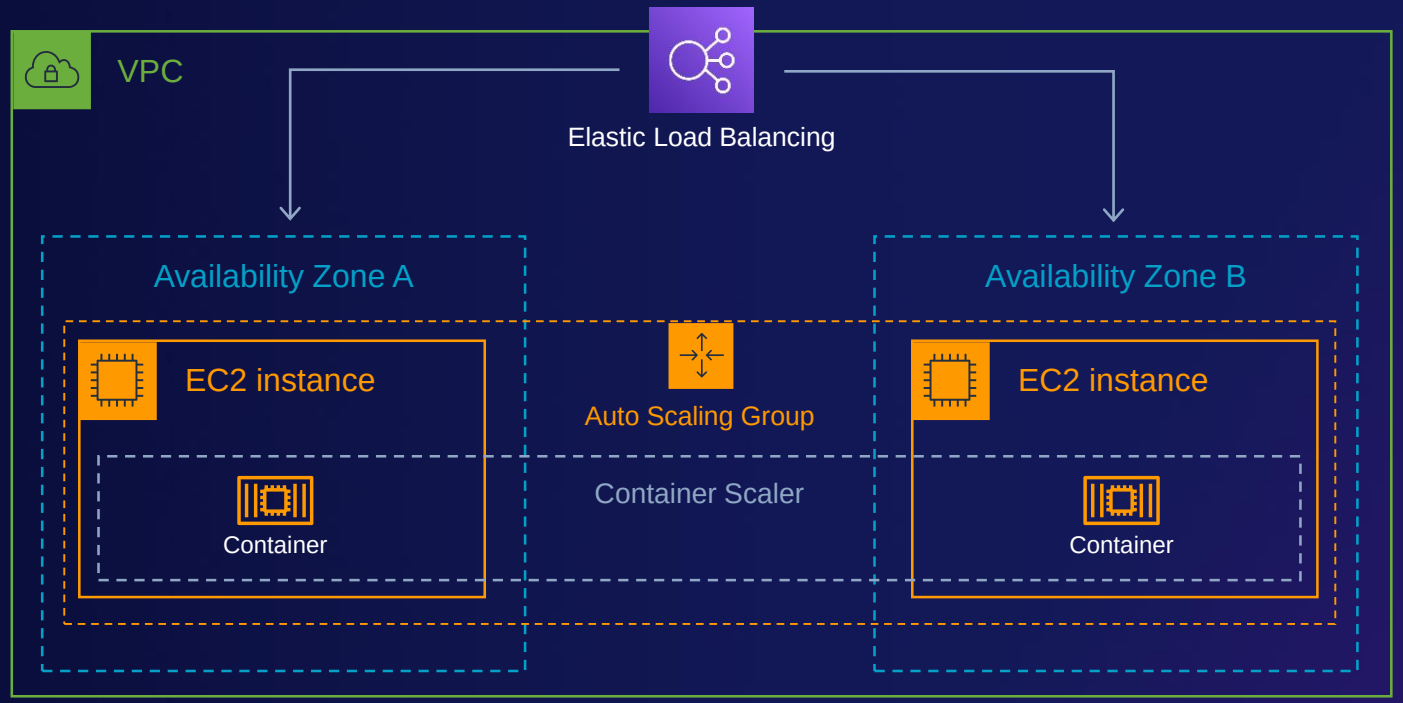
Containers infrastructure plumbing



Amazon Elastic Container Service (Amazon ECS)

Amazon Elastic Kubernetes Service (Amazon EKS)

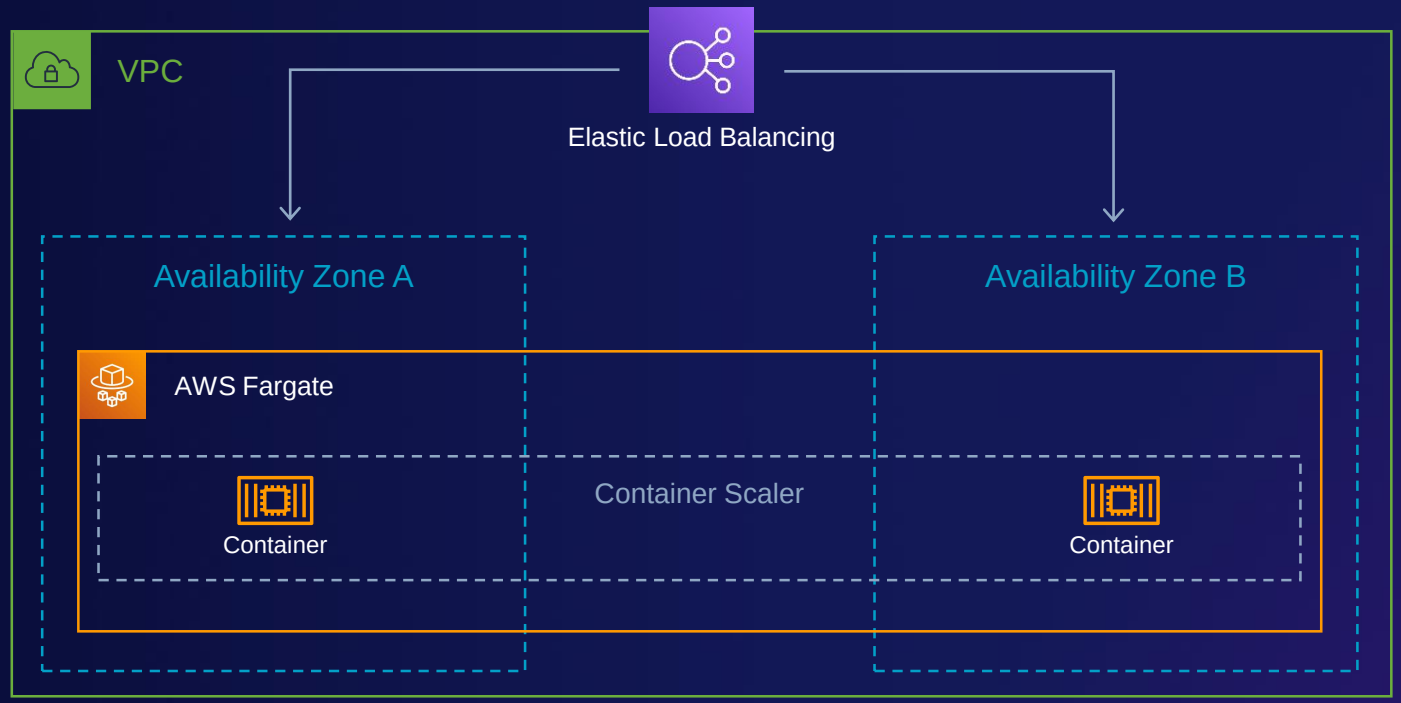
Containers infrastructure plumbing



Amazon Elastic Container Service (Amazon ECS)

Amazon Elastic Kubernetes Service (Amazon EKS)

Containers infrastructure plumbing



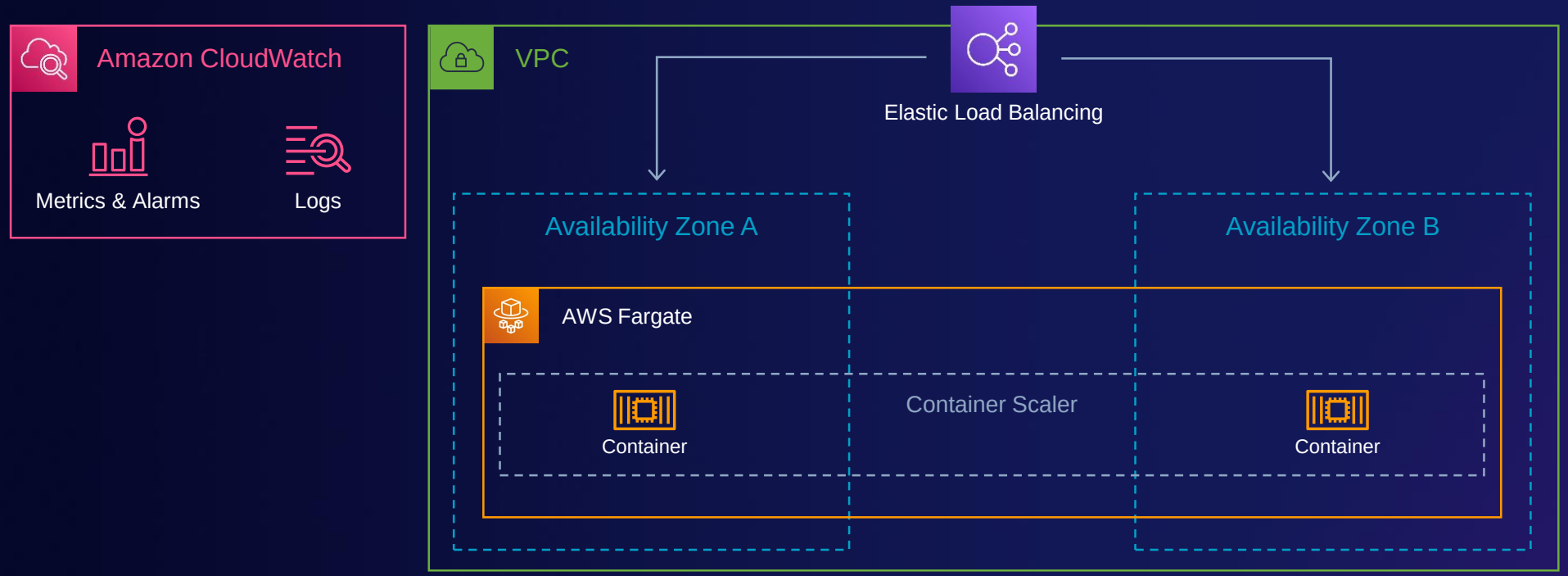
Amazon Elastic Container Service (Amazon ECS)

Amazon Elastic Kubernetes Service (Amazon EKS)

Fargate is a serverless compute engine for containers

Challenge
How is the system behaving?

Containers infrastructure plumbing

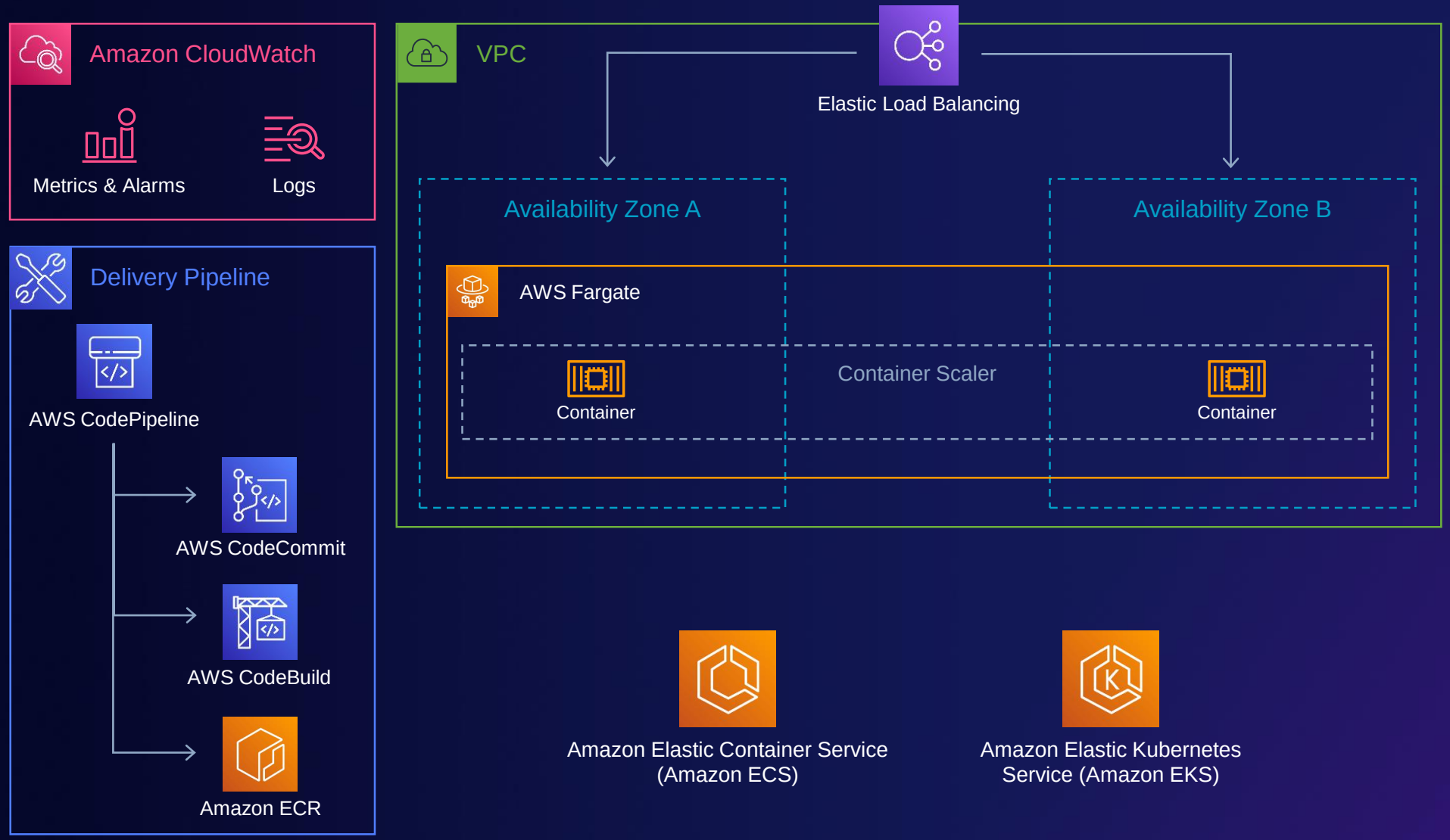


Amazon Elastic Container Service (Amazon ECS)

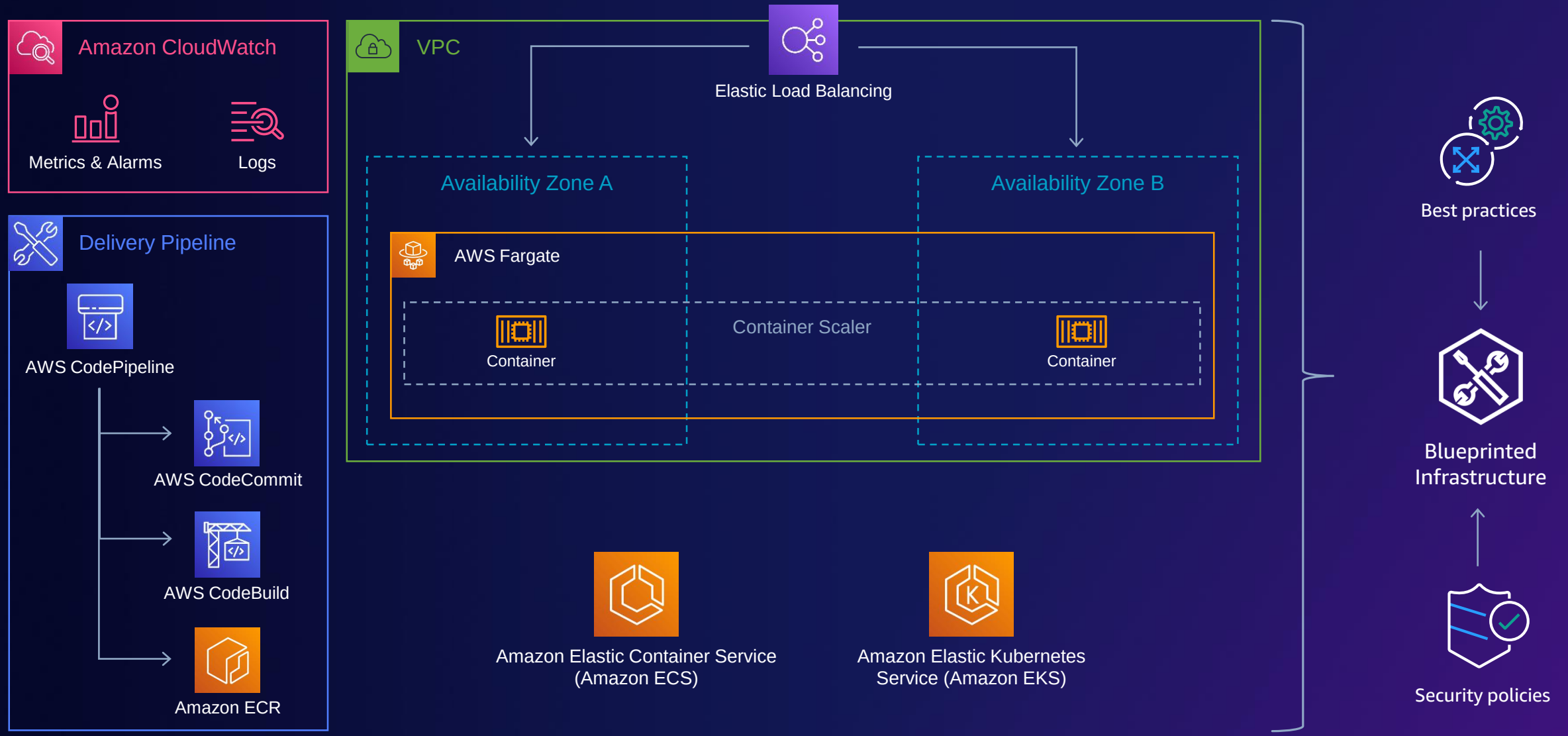
Amazon Elastic Kubernetes Service (Amazon EKS)

Challenge
Automation

Containers infrastructure plumbing



Containers infrastructure plumbing



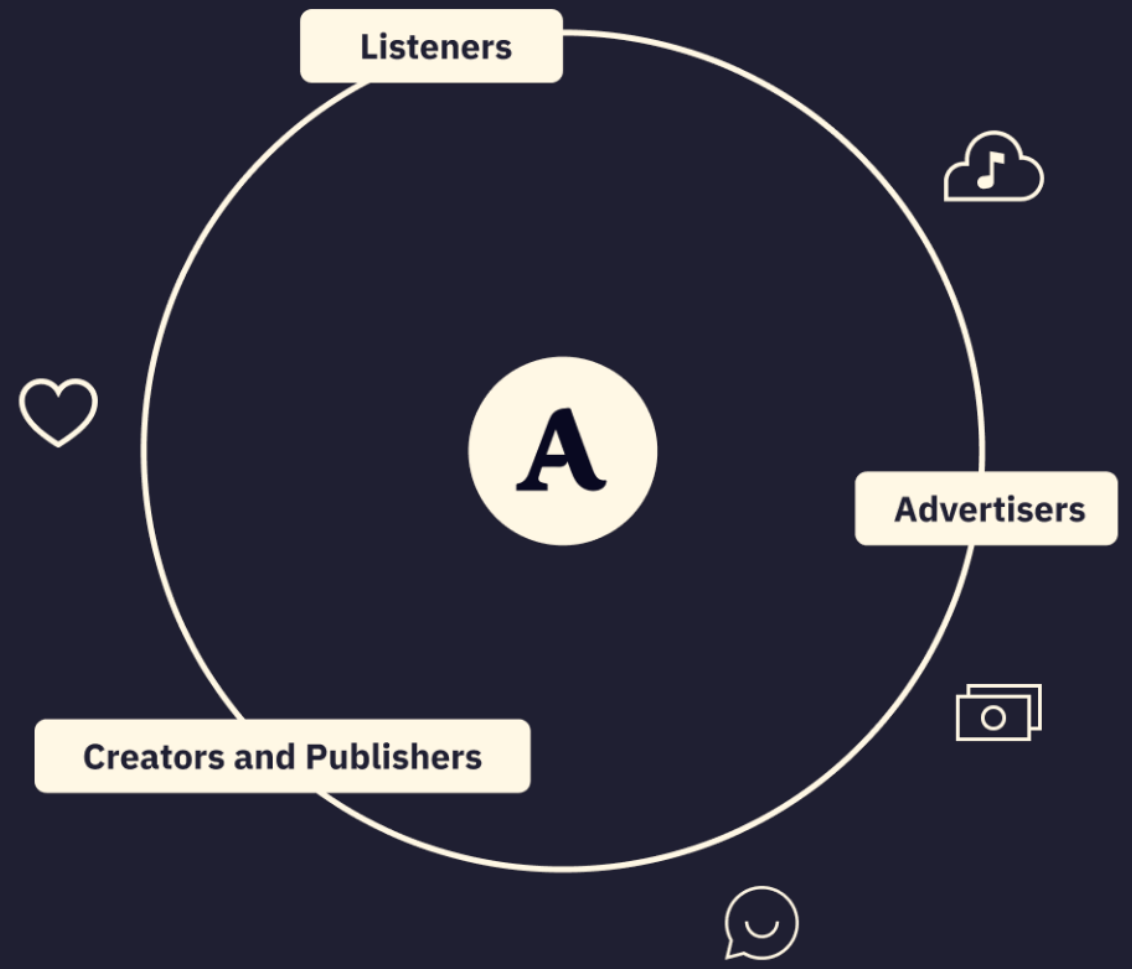
Customer story – Acast

Who are we?

Who are we?

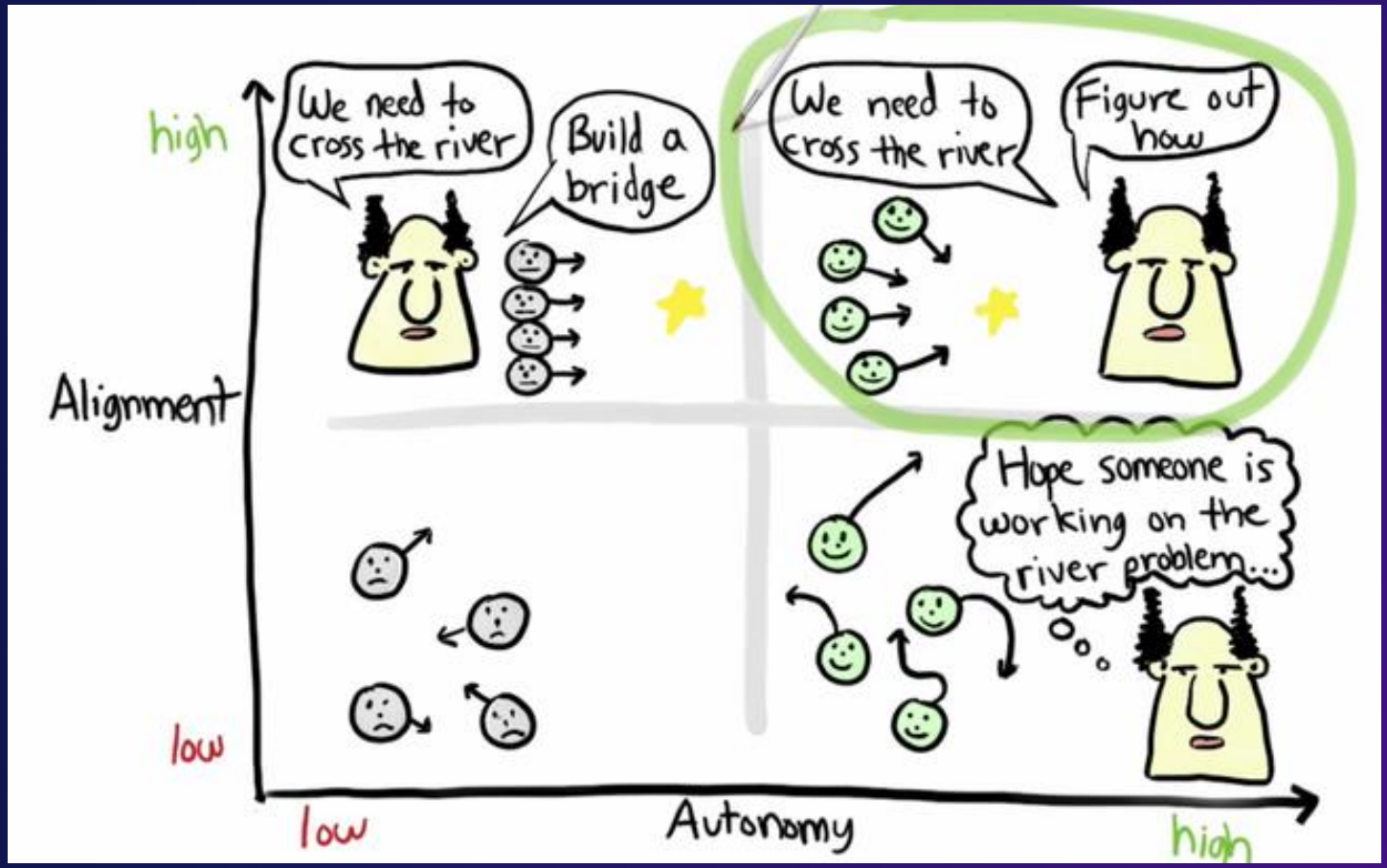
Acast is the world's leading independent podcast company. We help creators and advertisers of all sizes to reach listeners in the most immersive environment in the world

Acast mission: We help more podcasters find valuable audiences and make more money.

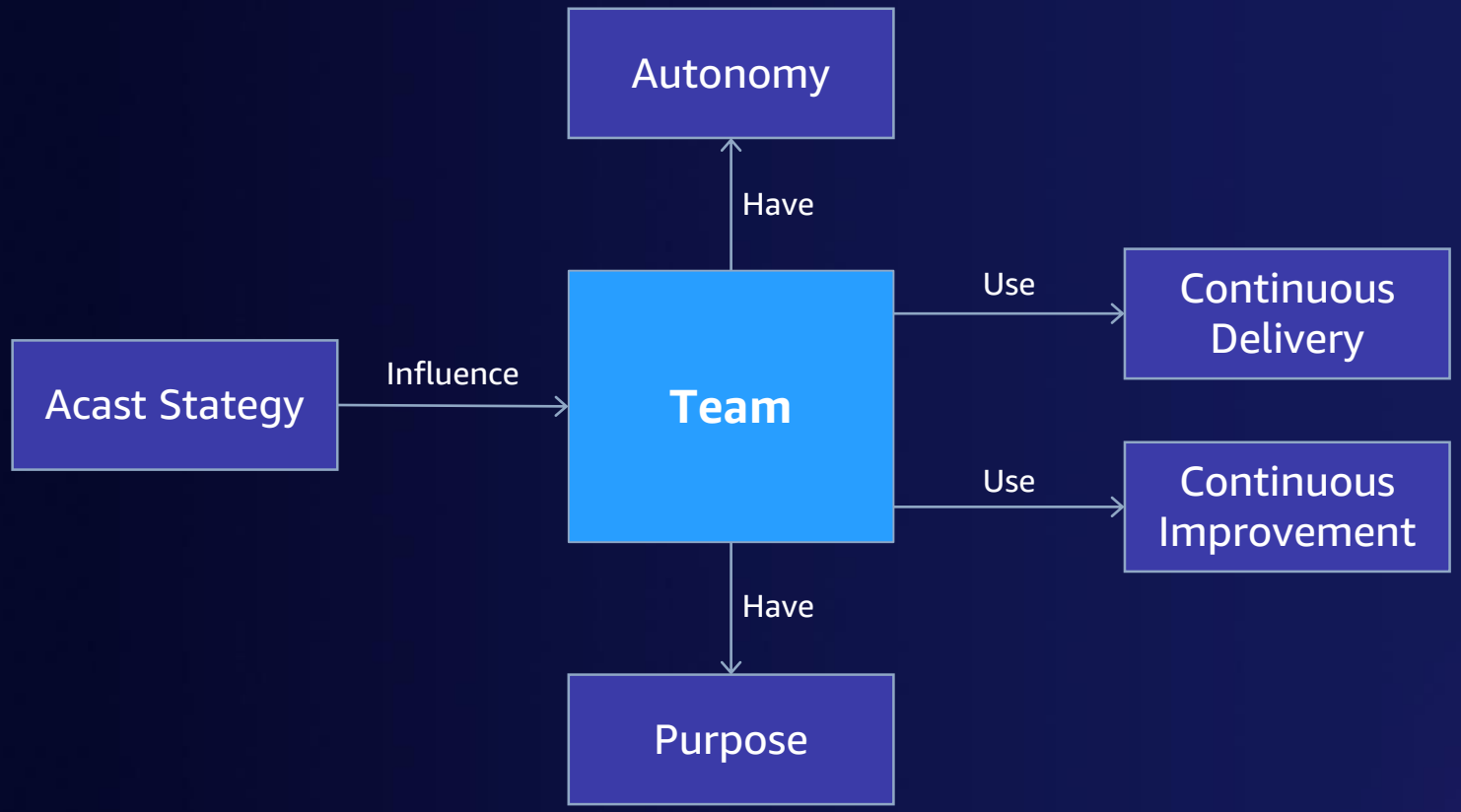


From sales driven to product driven

- From start-up to scale-up
- Output vs Outcome
- Inverse Conway’s maneuver
- Autonomous teams
- Low operational burden



Our core principles today



Teams are autonomous, purpose driven teams. They have the right competences to execute within their purpose.

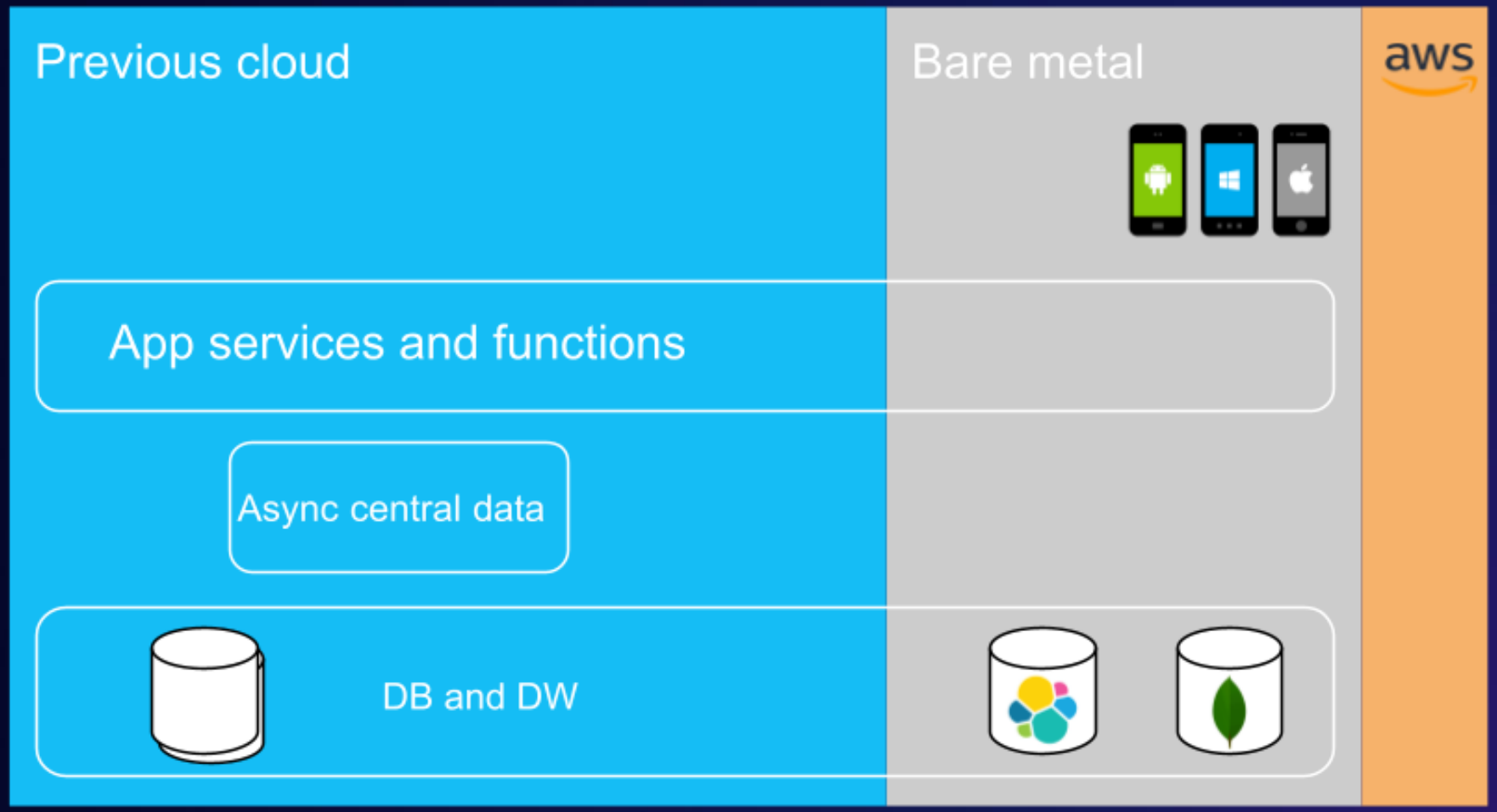
The team's roadmap is influenced by Acast's strategy and the teams are expected to continuously deliver value in small increments.

Teams are also expected to never settle with status quo, and always strive to get better and continuously improve.

Acast's technology journey

Acast's technology journey

2018 – CLOUD AGNOSTIC



Managed Runtime (.NET CLR)

Elastic on Searchly

ELK on-prem

DBs in IBM

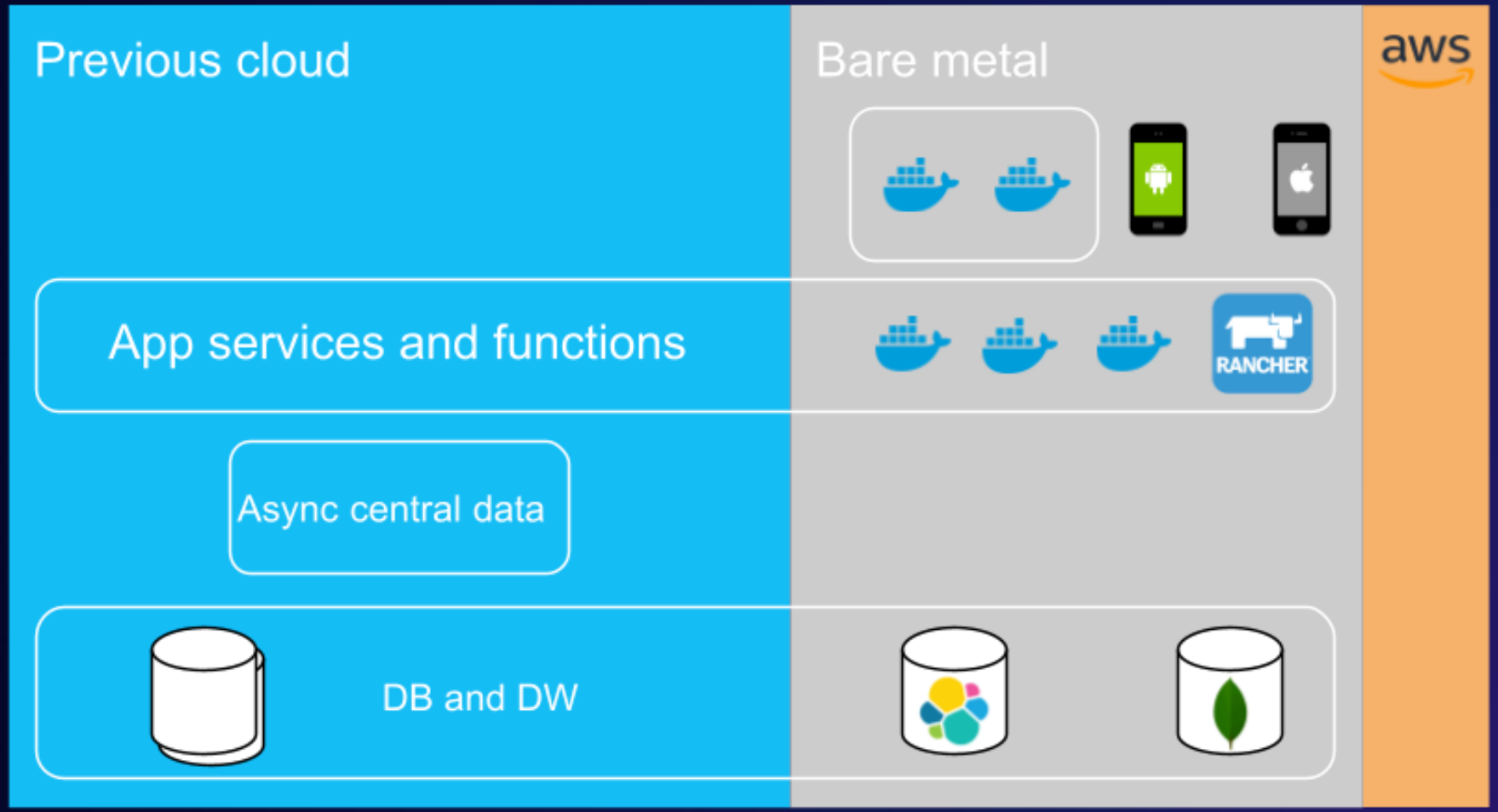
Frontend monolith

Backend microservices

CloudFront CDN in AWS

Acast's technology journey

2019 – CLOUD AGNOSTIC



Deprecation of the previous cloud starts

Buying/renting capacity externally

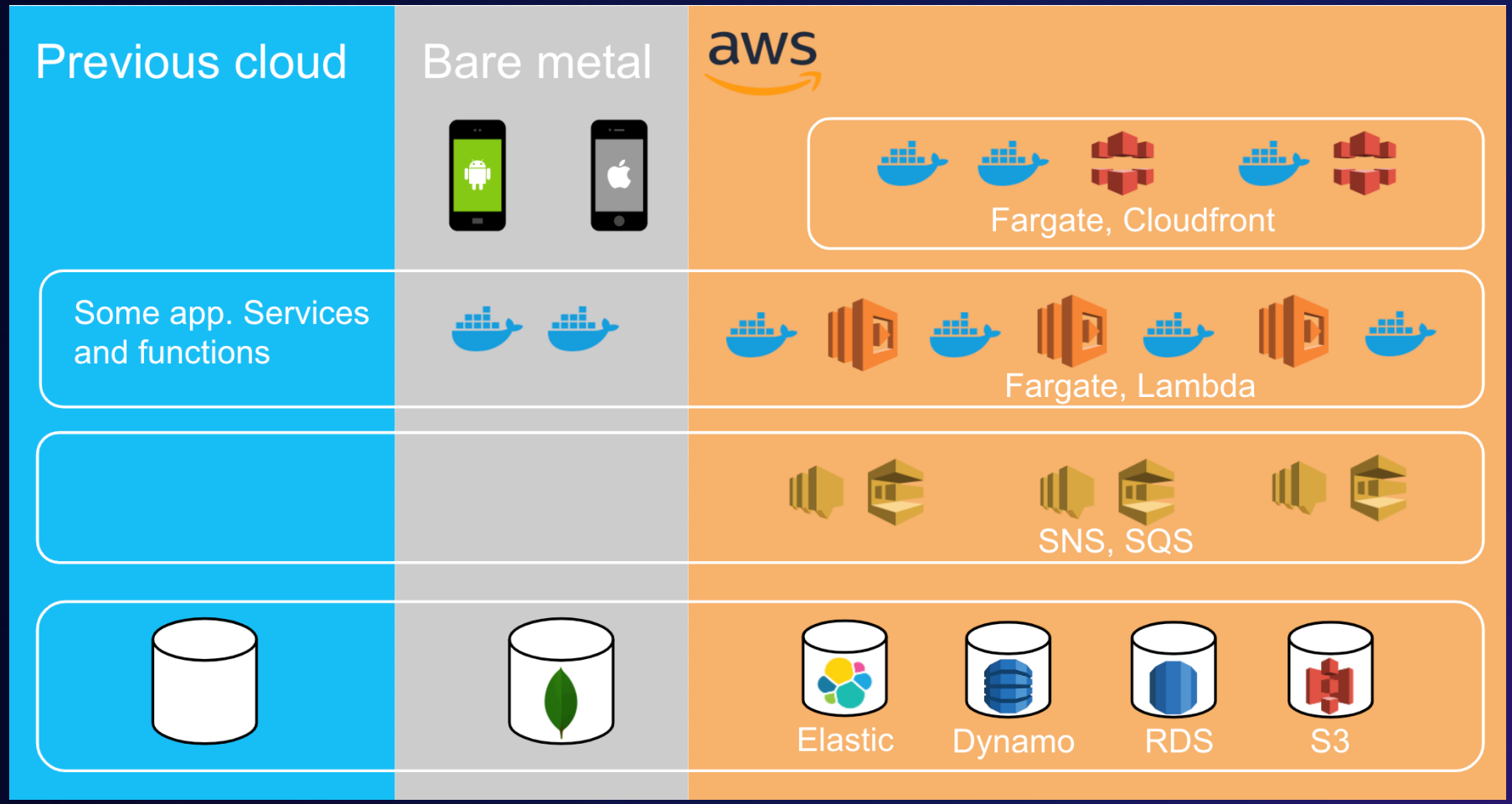
Managed Docker in Rancher

Decoupled frontend

CloudFront CDN in AWS

Acast's technology journey

2020 – CLOUD NATIVE WITH AWS



Deprecation of the previous cloud continues

Bare metal EOL closing in

Elastic moved to AWS OpenSearch

Docker containers with AWS Fargate

Pub-sub with SNS/SQS

Data storage to AWS RDS, Dynamo and S3

CloudFront CDN in AWS

Acast's technology journey

2018

- ✚ Good support for Dotnet
- ▬ Low support for NodeJS
- ▬ Low support for Docker
- ▬ Lots of issues

2019

- ✚ Cheap capacity
- ✚ Rancher + Docker
- ▬ High operational cost
- ▬ Black box for teams
- ▬ Manual scaling

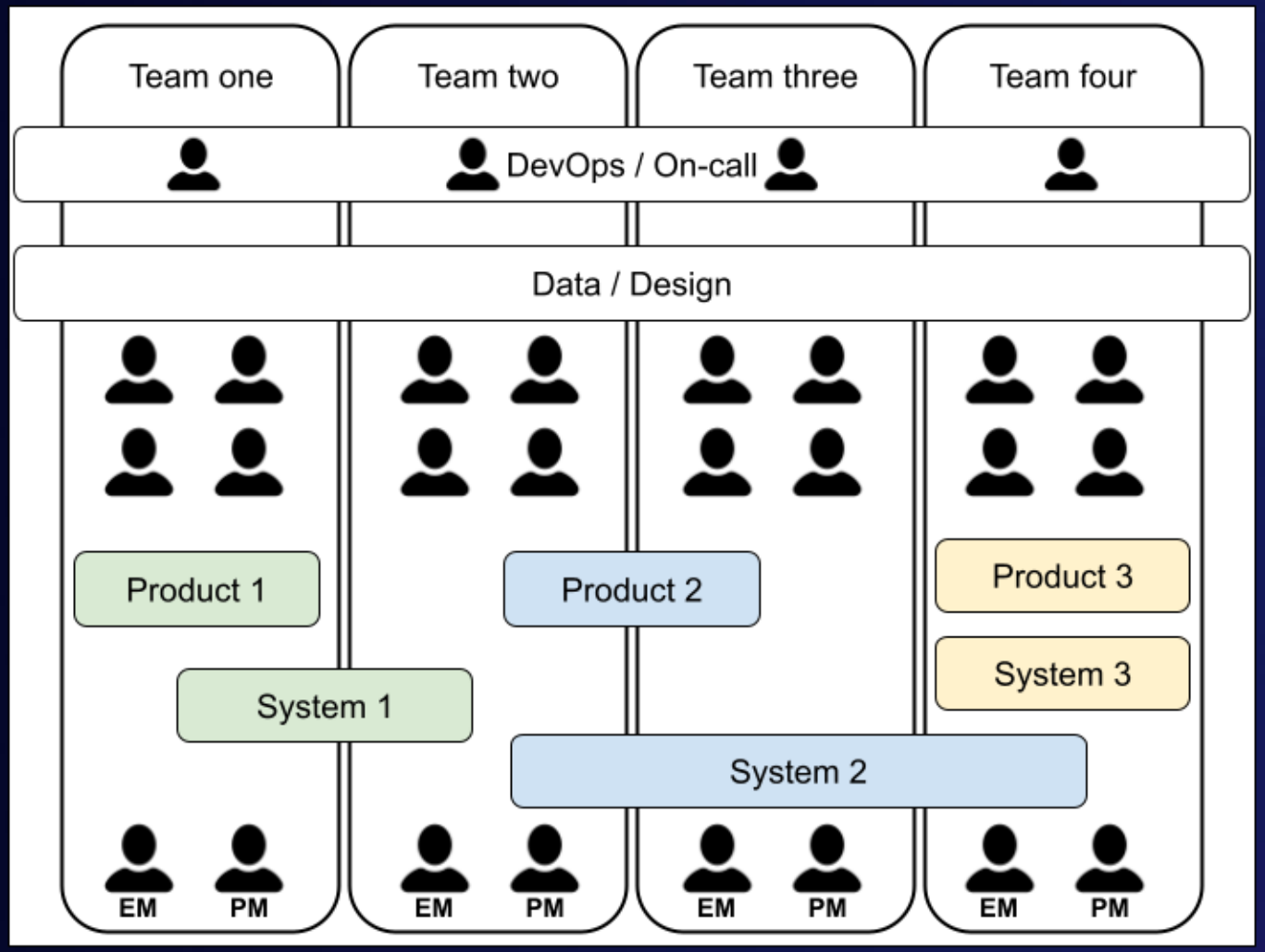
2020

- ✚ Infra as code
- ✚ Operations in teams
- ✚ Fully-managed by AWS
- ✚ Scalable
- ✚ Enables focus on products

Teams’ structure and operational model

Teams' structure and operational model

2018 - 2019



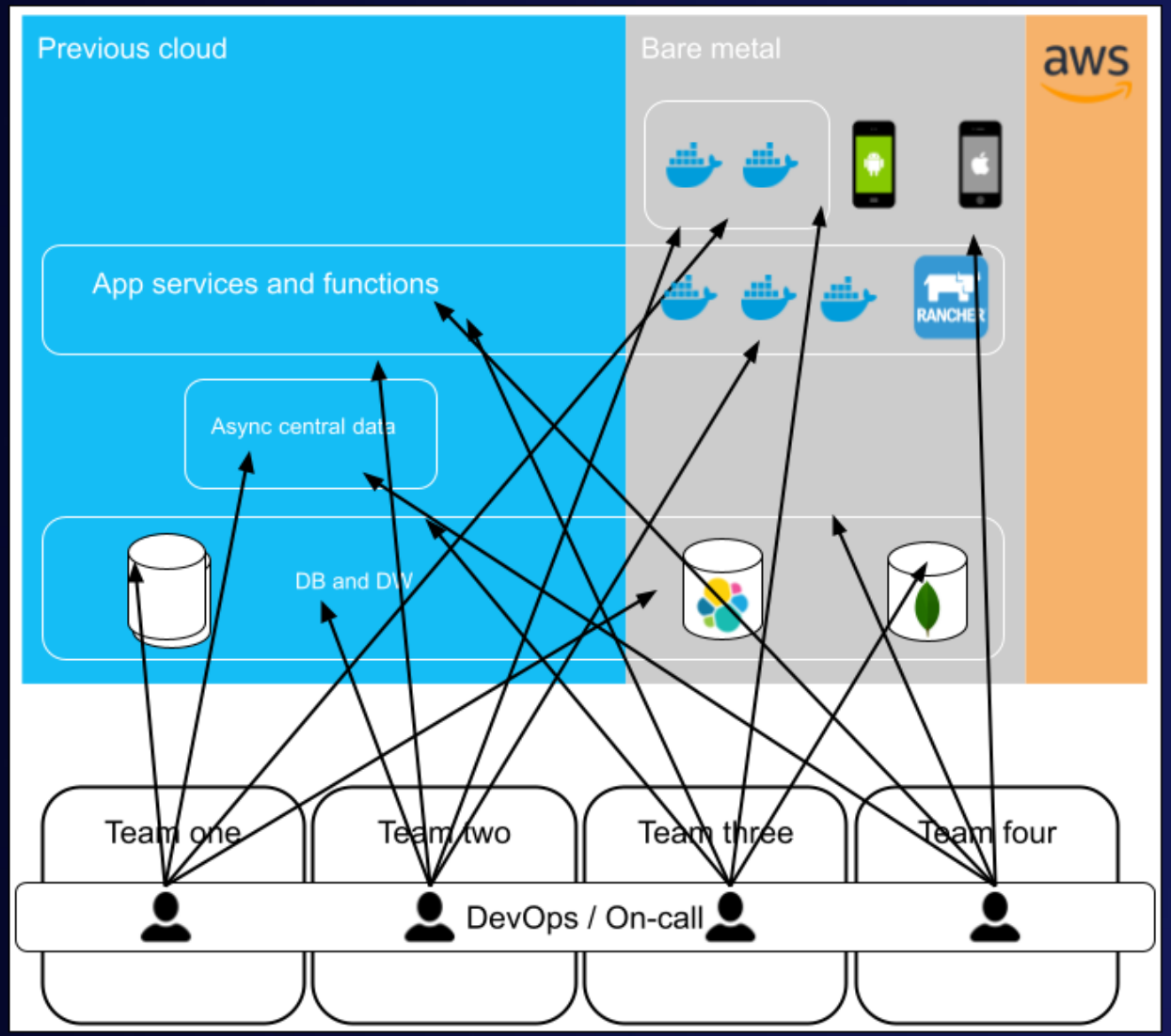
Shared ownership.

Global DevOps/On-call for a selection of runtime systems

Different providers, different deploy pipelines and technologies

Complex change management

Teams' structure and operational model



Too much overhead

Conway's law: our system design does NOT mirror our communication structure.

High coupling, low cohesion.

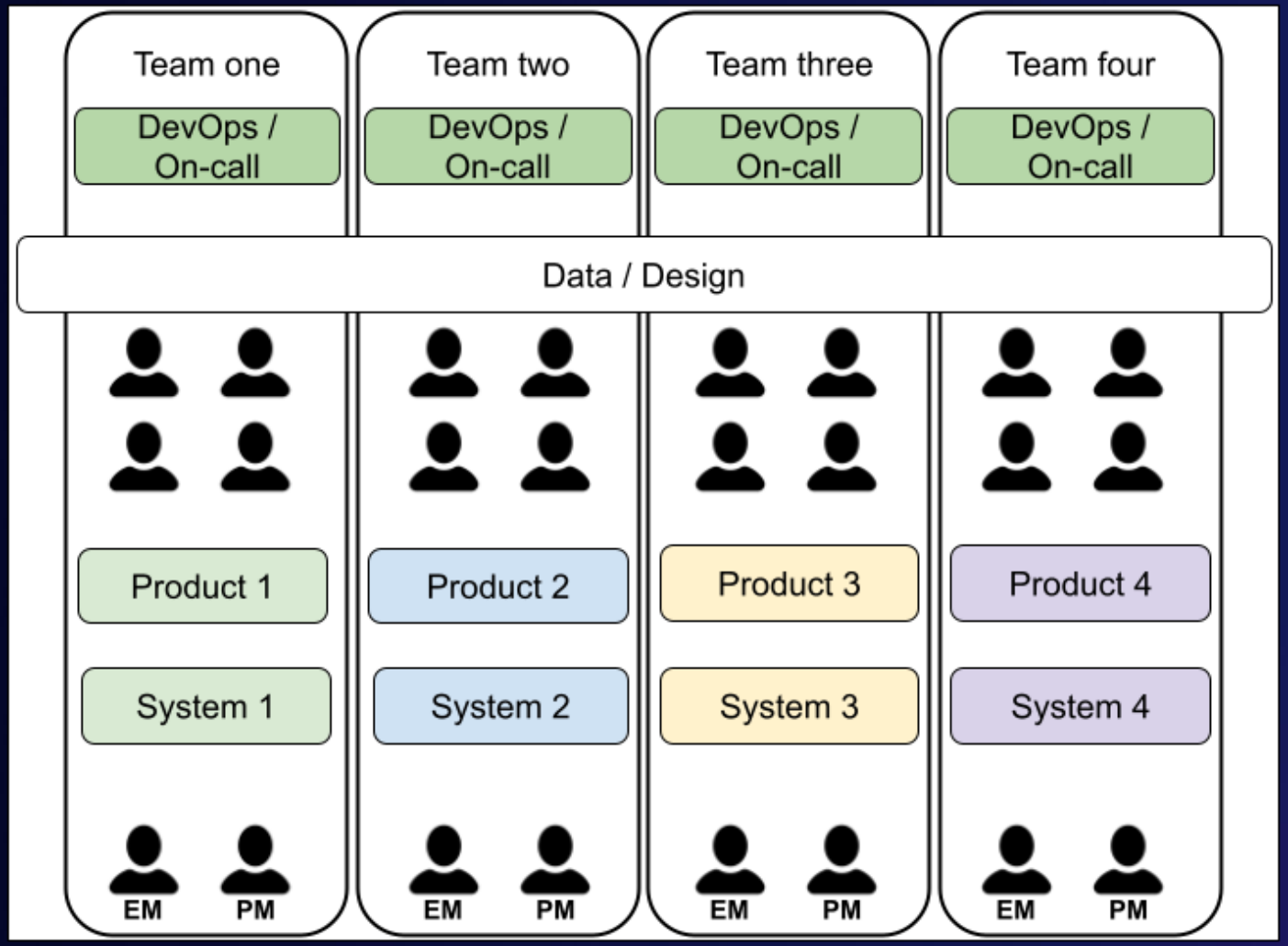
Communication is complex

Operational burden is high and limited to a few people

This operational model does not scale

Teams' structure and operational model

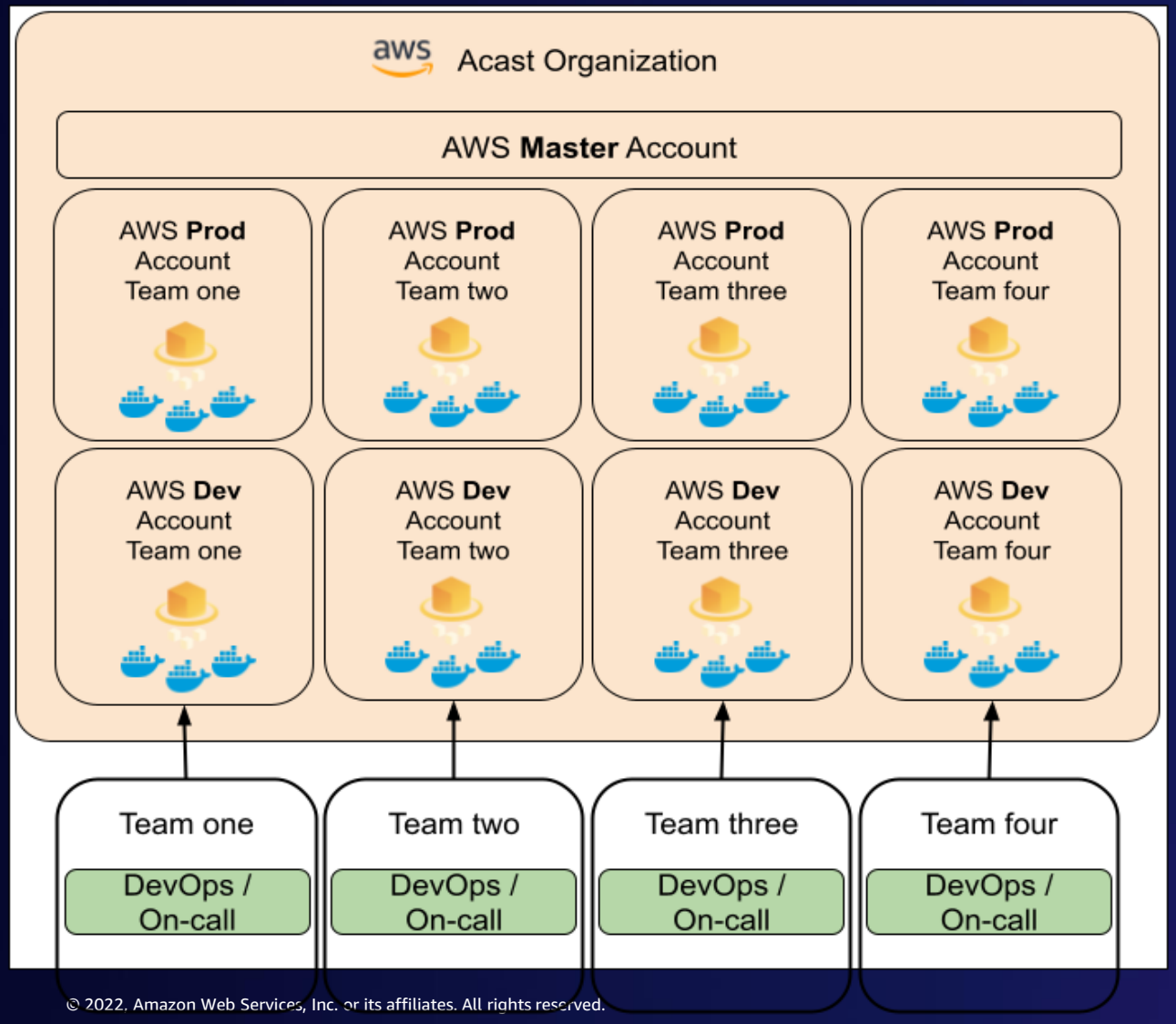
2020 – PRESENT: CONTAINERIZATION AND BOUNDED CONTEXT



- Team autonomy
- End-to-end responsibility
- Clear(er) ownership
- Clear(er) communication structure
- DevOps and on-call in the team. All production systems backed by on-call rotation.

Teams' structure and operational model

2020 – PRESENT: FOCUS, SPEED AND LOW OPERATIONAL BURDEN



- Low operational overhead
- Improved CI/CD
- Increased productivity
- Easier change management
- Easier control/access management

How does ECS, Fargate and CDK help us?

Reducing the burden – ECS, Fargate and CDK

DevOps/infrastructure in the teams
CDK allows backend developers to create, maintain and evolve their infrastructure without having to be experts.

Fargate allows developers to focus on applications, not infrastructure
Pay for what we use – CPU and memory

Technology choices allow for flexibility while quickly delivering value to customers

Acast Tech in numbers today (April 2022)

19

Tech Teams

12

7

50

AWS Accounts

Dev

Prod

Disaster recovery

77

ECS Clusters

95%

5%

793

Running tasks

75%

25%

AWS Cloud Development Kit

AWS Cloud Development Kit

Familiar
Your language
Just classes and methods

Tool Support
AutoComplete
Inline documentation

Abstraction
Sane defaults
Reusable classes

```
class UrlShortener extends Stack {
  constructor(scope: App, id: string, props?: UrlShortenerProps) {
    super(scope, id, props);

    const vpc = new ec2.Vpc(this, 'vpc', { maxAzs: 2 });
    const cluster = new ecs.Cluster(this, 'cluster', { vpc: vpc });
    const service = new patterns.NetworkLoadBalancedFargateService(this, 'sample-app', {
      cluster,
      taskImageOptions: {
        image: ecs.ContainerImage.fromAsset('ping'),
      },
      dom
    });
    // Setup AutoScaling policy
    const scaling = service.service.autoScaleTas
    scaling.scaleOnCpuUtilization('CpuScaling',
      targetUtilizationPercent: 50,
      scaleInCooldown: Duration.seconds(60),
      scaleOutCooldown: Duration.seconds(60)
    });
  }
}
```

domainName
domainZone

(property) patterns.NetworkLoadBalancedServiceBaseProps.domainName?: string | undefined

The domain name for the service, e.g. "api.example.com."

@default

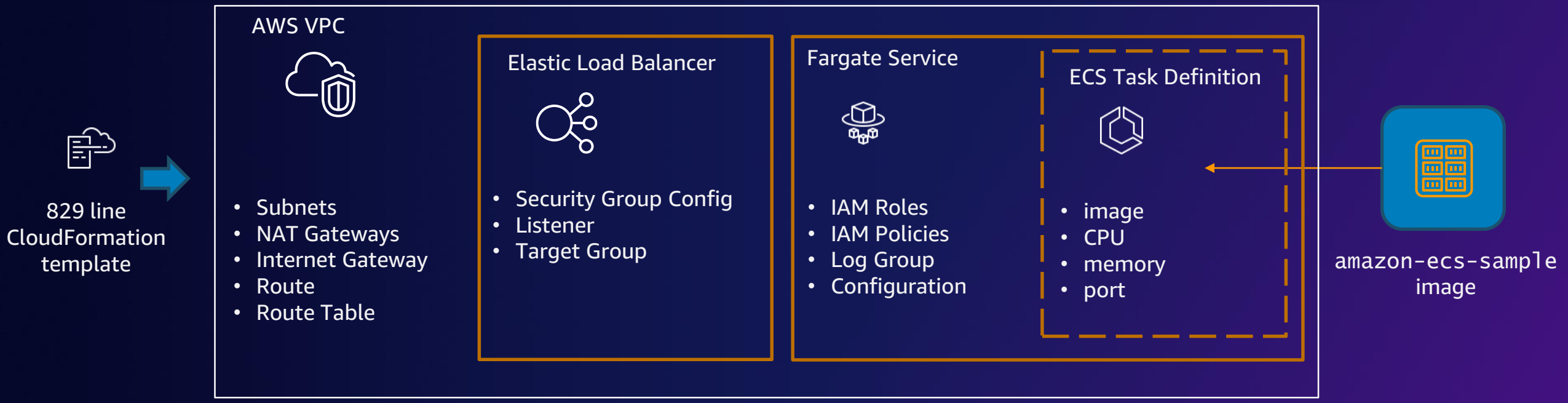
- No domain name.

CDK Constructs: Levels

L3+	Purpose-built constructs	Opinionated abstractions
L2	AWS Constructs	High level service constructs
L1	CloudFormation Resources	Automatically generated

L3

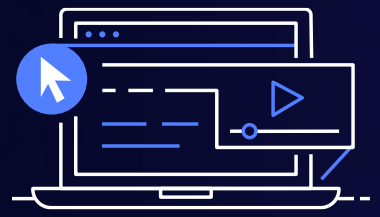
```
const myService = new ecs_patterns.ApplicationLoadBalancedFargateService(this, 'Service', {
  cluster,
  taskImageOptions: {
    image: ecs.ContainerImage.fromRegistry('amazon/amazon-ecs-sample')
  }
});
```



Demo

AWS makes it easy to build with containers

AWS container services simplify the management of your underlying container infrastructure, whether on premises or in the cloud



DIVE DEEP

Workshops

- Amazon ECS workshop
  - https://www.ecsworkshop.com/
- Amazon EKS workshop
  - https://www.eksworkshop.com/
- AWS Cloud Development Kit Workshop
  - https://cdkworkshop.com/

REGISTRY

- Amazon ECR

ORCHESTRATION

- Amazon ECS
- Amazon EKS
- Red Hat OpenShift Service on AWS
- AWS App Runner

COMPUTE

- AWS Fargate
- Amazon EC2
- AWS App Runner

Learn in-demand AWS Cloud skills

AWS Skill Builder

Access **500+ free** digital courses and Learning Plans

Explore resources with a variety of skill levels and **16+** languages to meet your learning needs

Deepen your skills with digital learning on demand

Train now

AWS Certifications

Earn an industry-recognized credential

Receive Foundational, Associate, Professional, and Specialty certifications

Join the **AWS Certified community** and get exclusive benefits

Access **new** exam guides

Please complete the session survey

Thank you!

Andreas Lindh

elindh@amazon.com

@andskli

Alberto López

https://www.linkedin.com/in/albelopez