INNOVATE

MODERN APPLICATIONS EDITION

27&28 October 2021

Breaking down the monolith with containers

Jason Umiker

Senior Specialist Solutions Architect - Containers
Amazon Web Services

Agenda

- Why containers?
- Why decouple monoliths into microservices?
- Popular decoupling patterns with containers
- Decoupling with containers for security
- Practical decoupling tips
- A (brief) overview of our container services

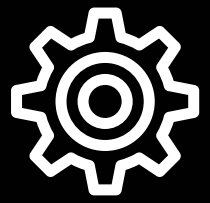
Why containers?

Applications aren’t just code, they have dependencies

Code

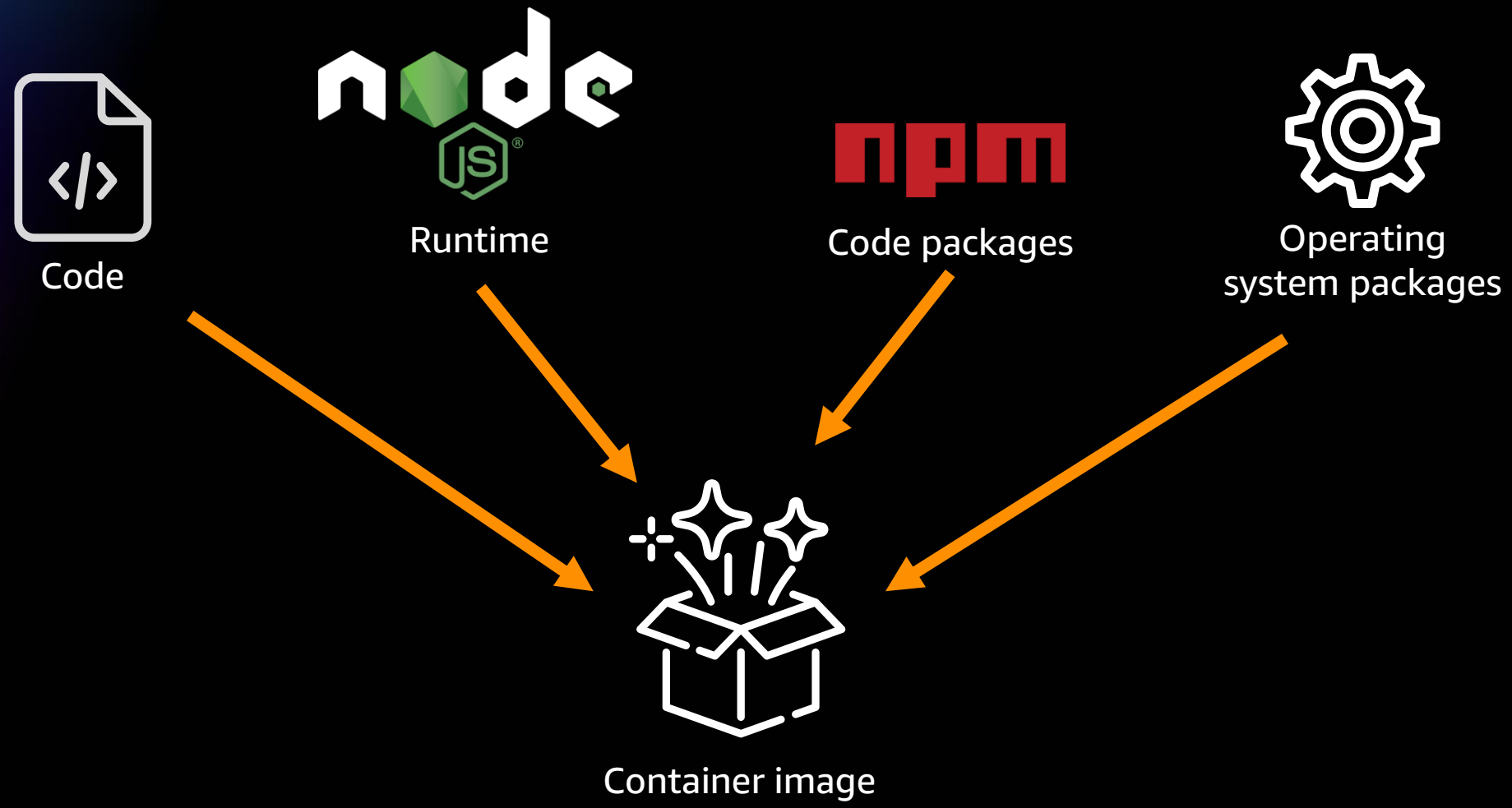
Runtime

Code packages



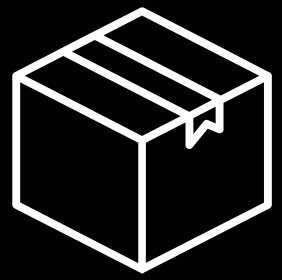
Operating system packages

Containers turn applications into one deployable artifact



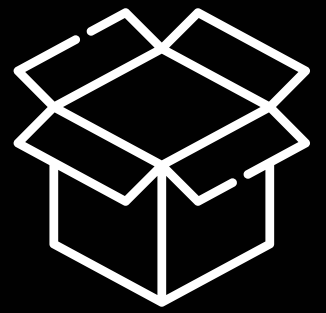
Build

Gather the app and its dependencies. Create an immutable container image



Push

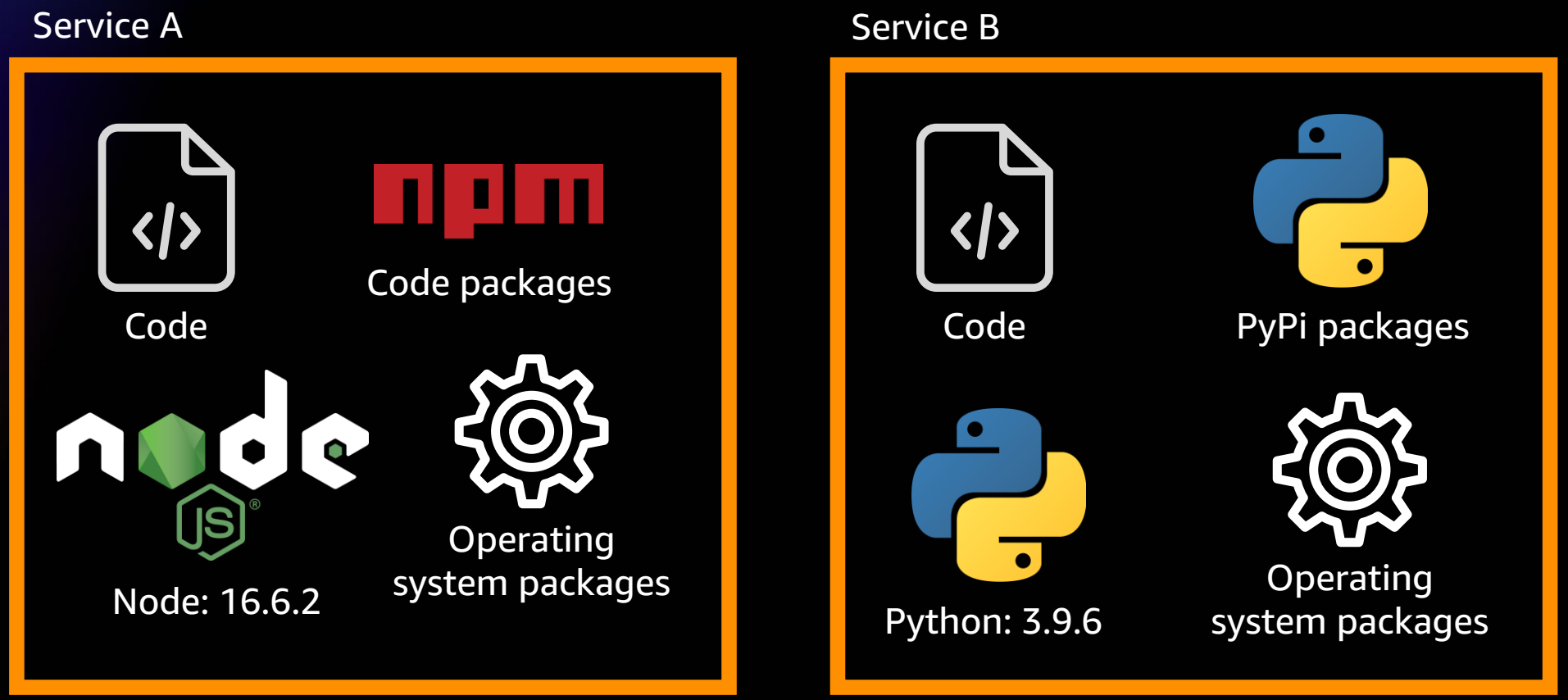
Store the container image in a registry so it can be downloaded to compute



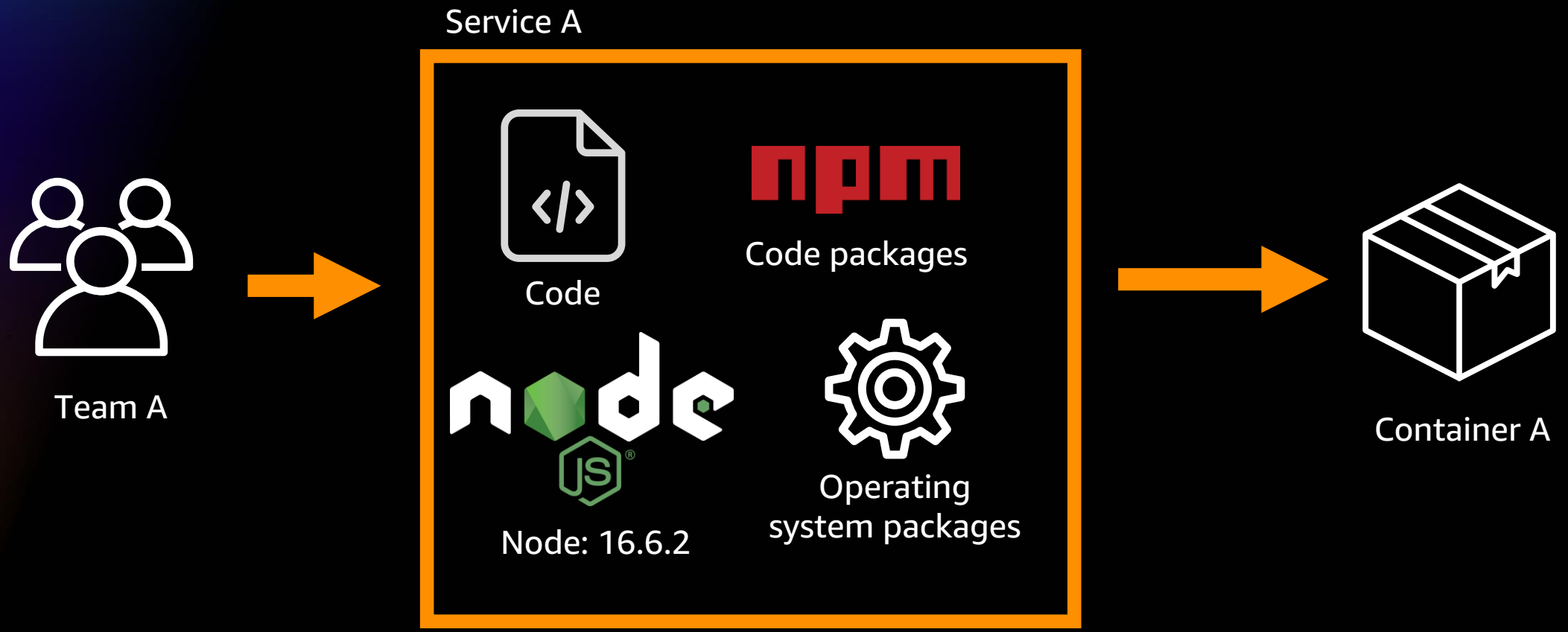
Run

Download image to compute, unpack it, and run it in an isolated environment

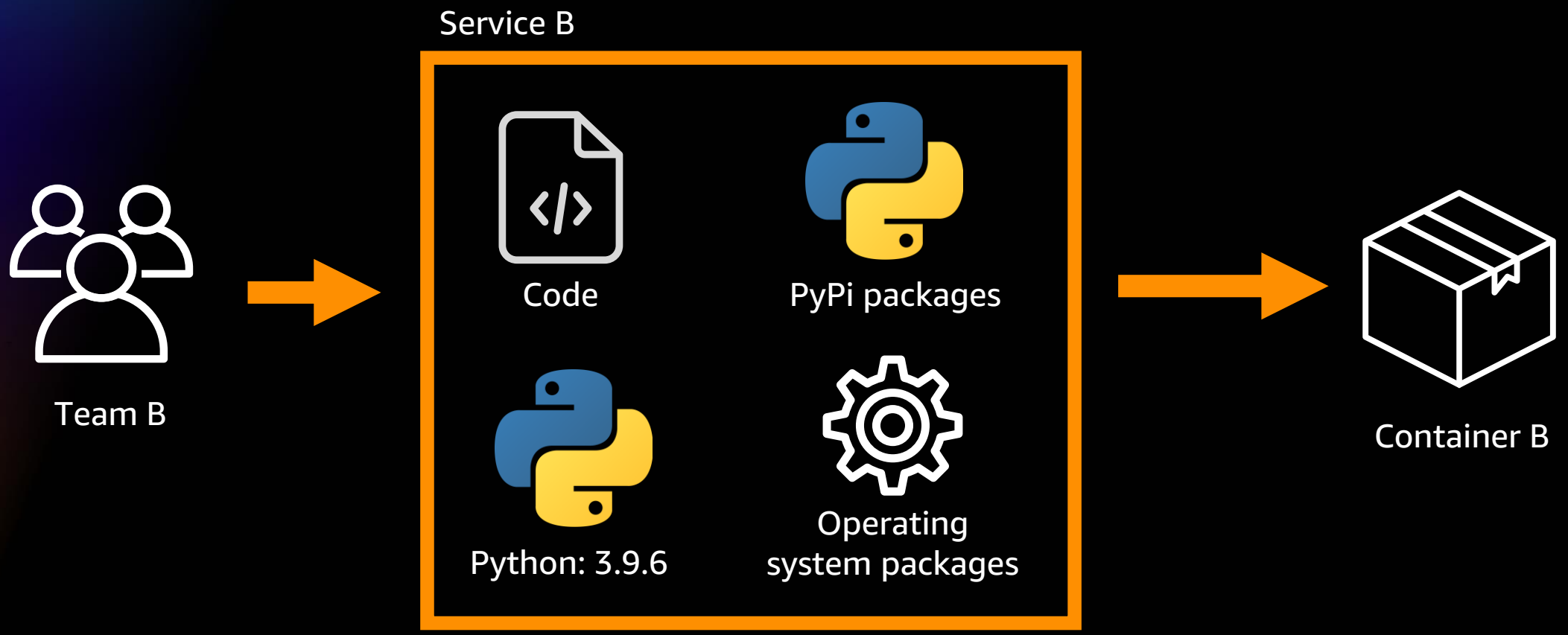
Breaking a monolith is scary because more services mean more dependencies



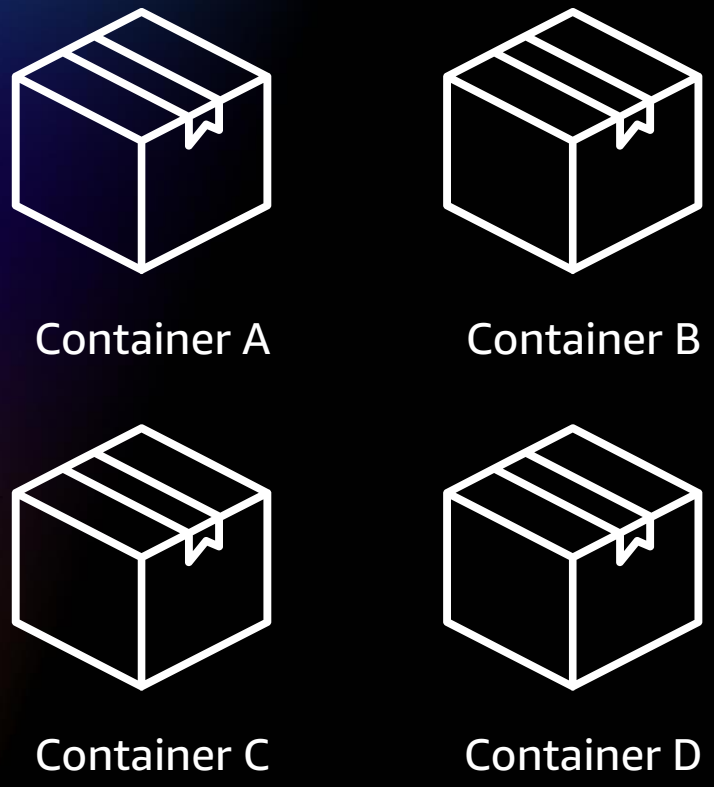
Containers make dependencies a decentralized job



Containers make dependencies a decentralized job



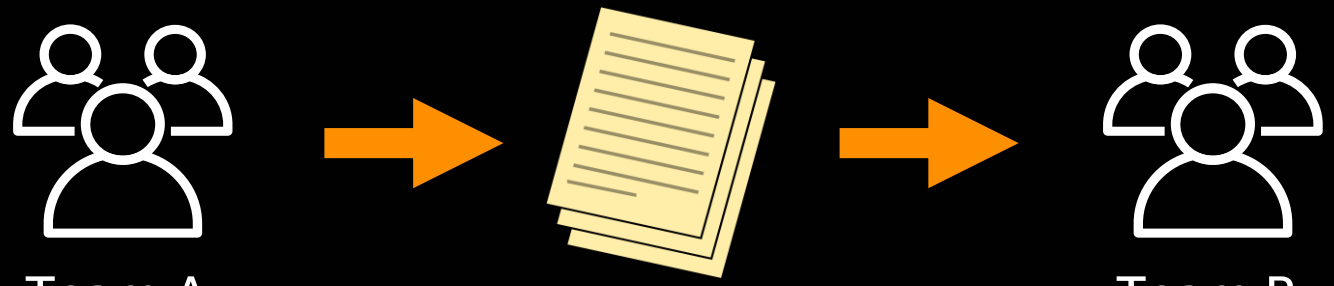
Infrastructure is now agnostic to container contents



Different containers can be handled the same way using the same tooling

Why decouple monoliths into microservices?

Decoupling your services = decoupling your teams



API = Contract
w/SLAs,
versioning, etc.

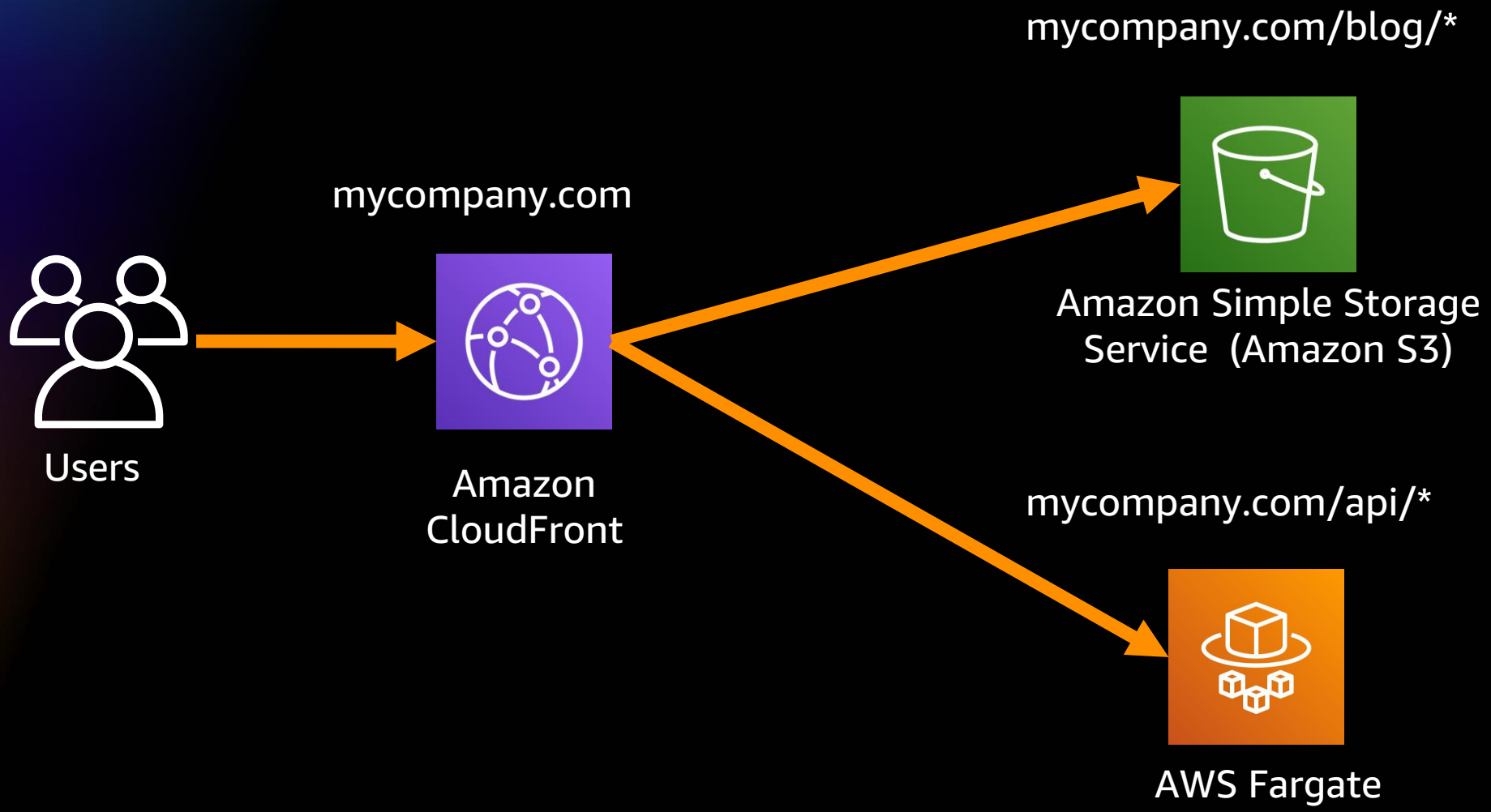
“Smaller teams working on smaller codebases tend to be more productive.”

“Can we make a change to a microservice and deploy it without having to deploy any other?”

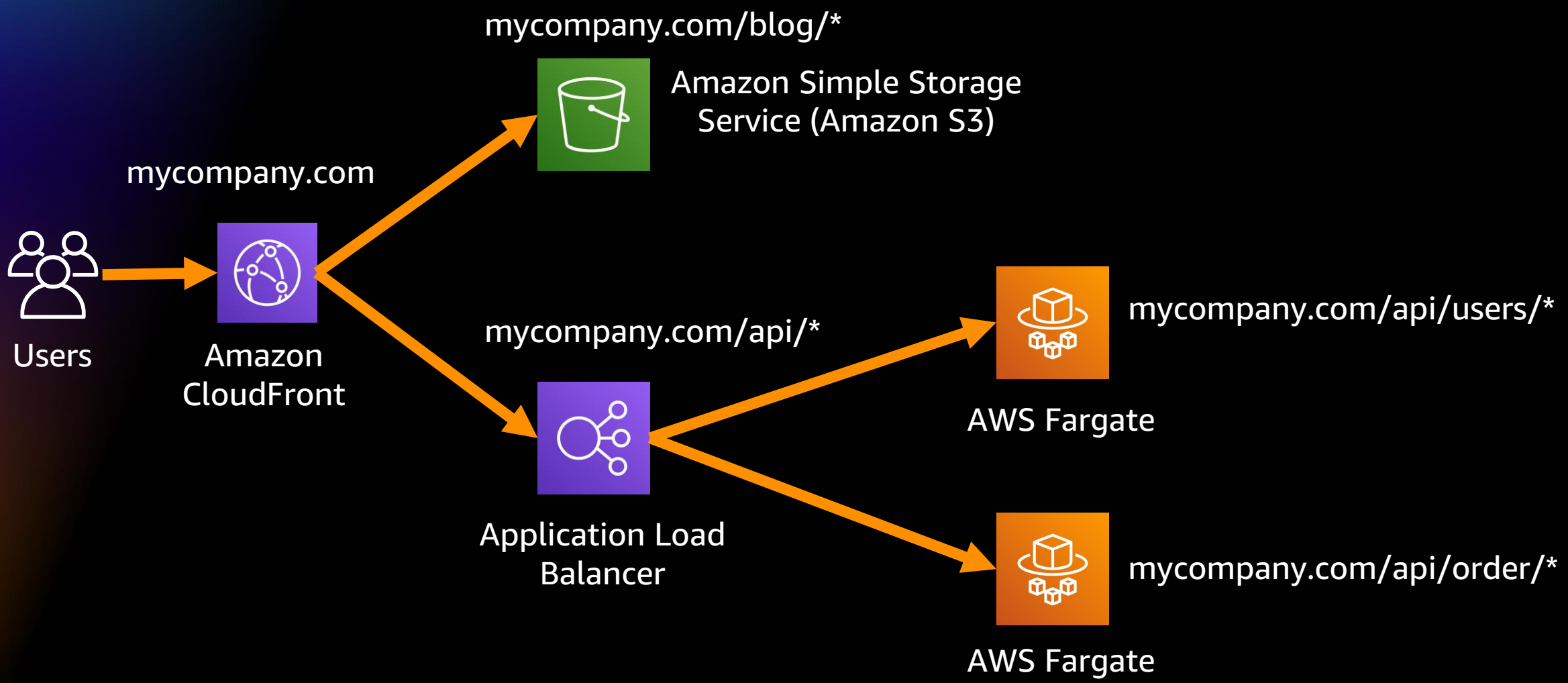
“Can we make a change to a microservice and deploy it without needing to have a meeting?”

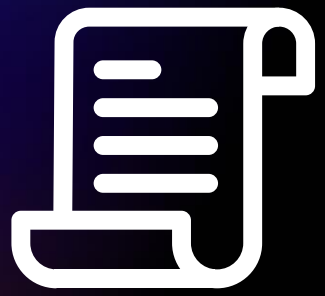
Popular decoupling patterns with containers

Decouple traffic: One domain, multiple services



Decouple API into microservices





Listener rule

Up to 100 rules

Match on host

Hostname == mycompany.com
Hostname == api.mycompany.com

Match on path

Path == /api/users
Path == /api/orders

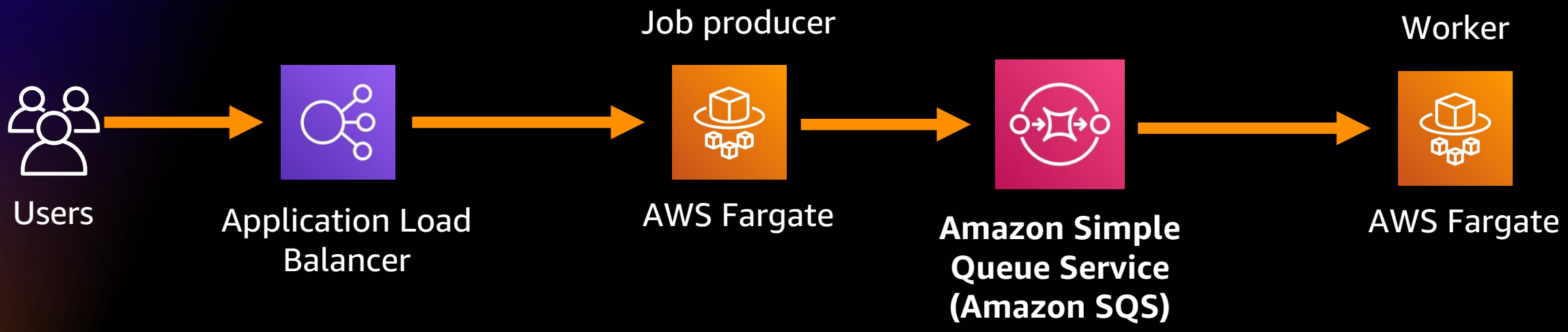
Match on query string

?utm_source==bot

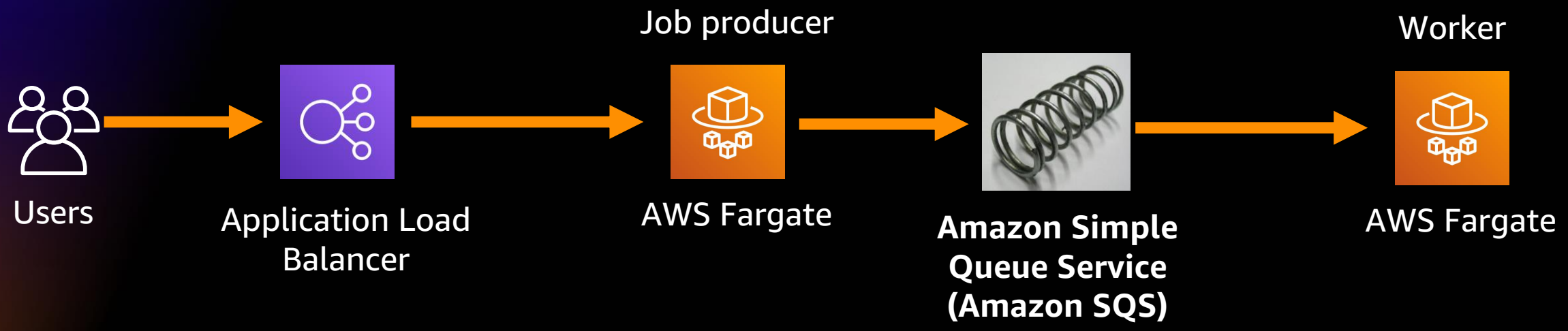
Match on header

Version == 1.0.0
User-Agent == mobile

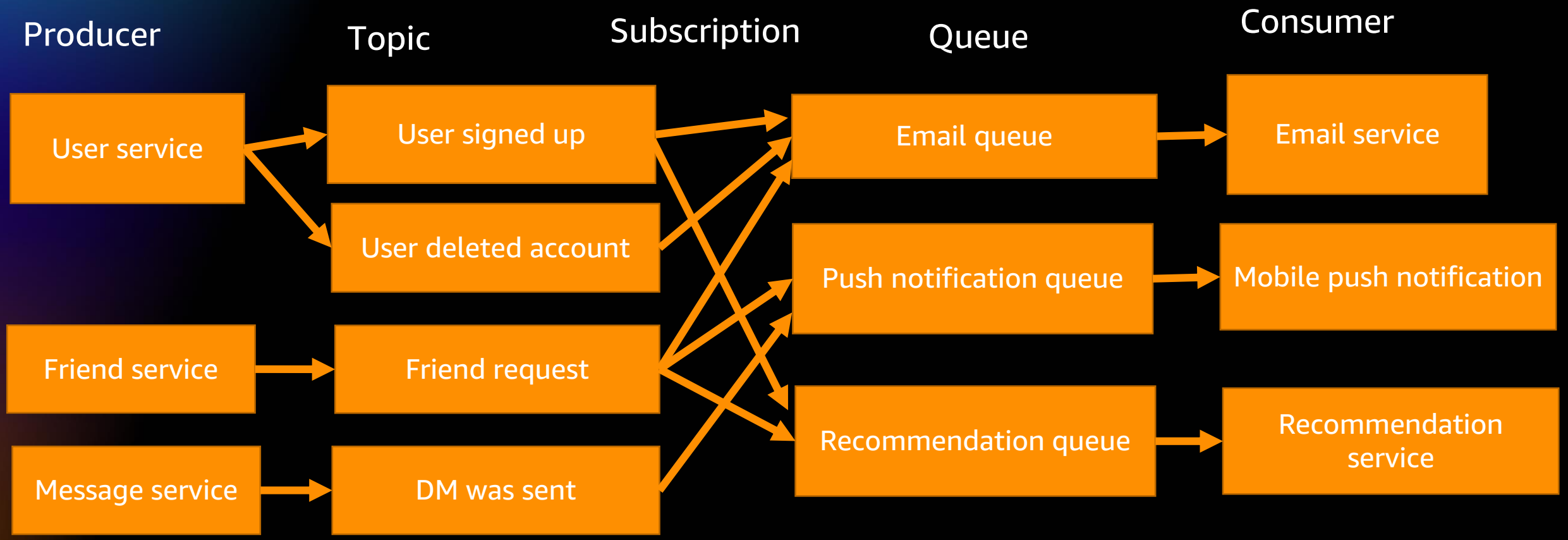
Decouple background workers



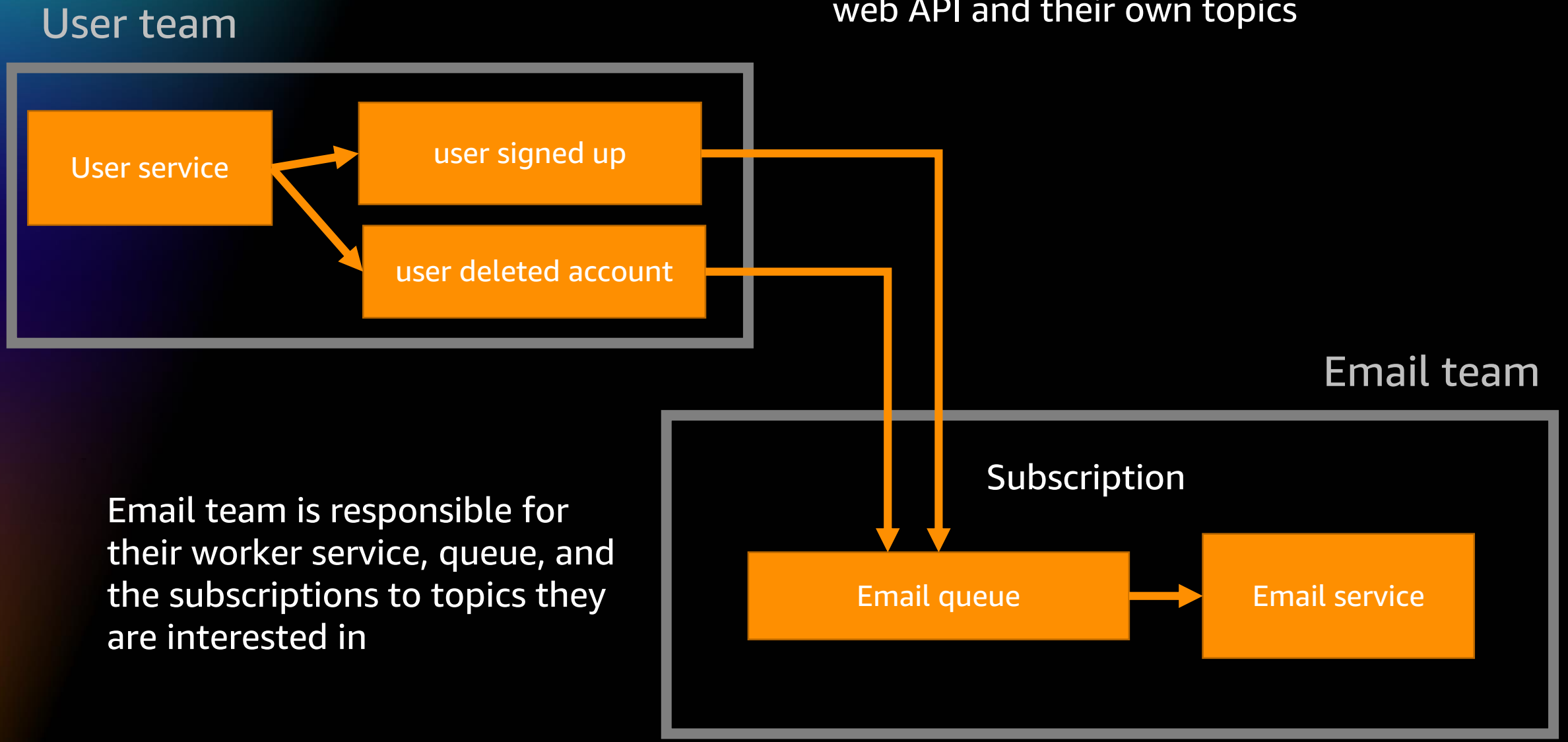
Decouple background workers



Use topics and queues for more complicated business logic



User team is responsible for web API and their own topics



Email team is responsible for their worker service, queue, and the subscriptions to topics they are interested in

Decouple scheduled tasks from the monolith

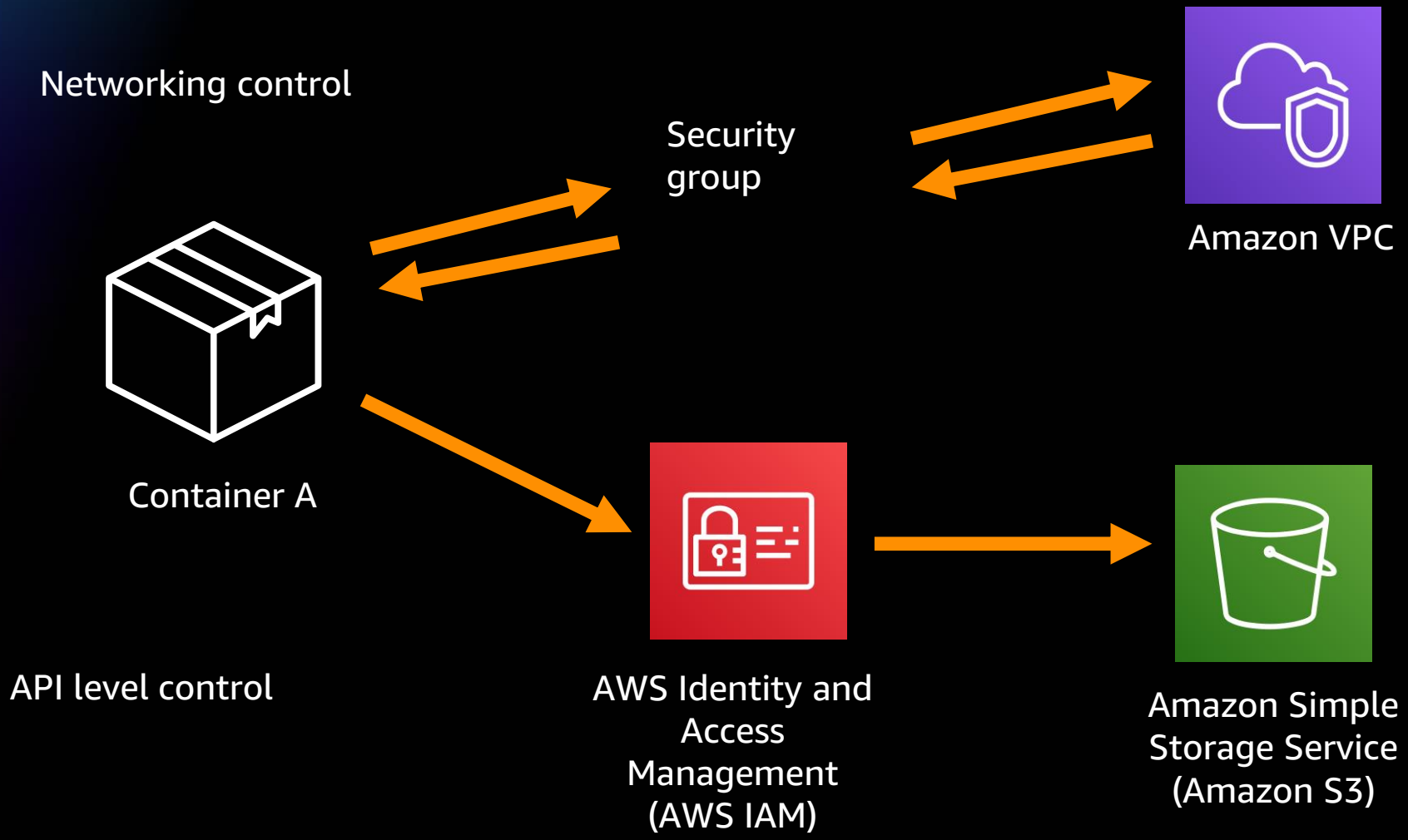
Amazon EventBridge

Every day at 5:00

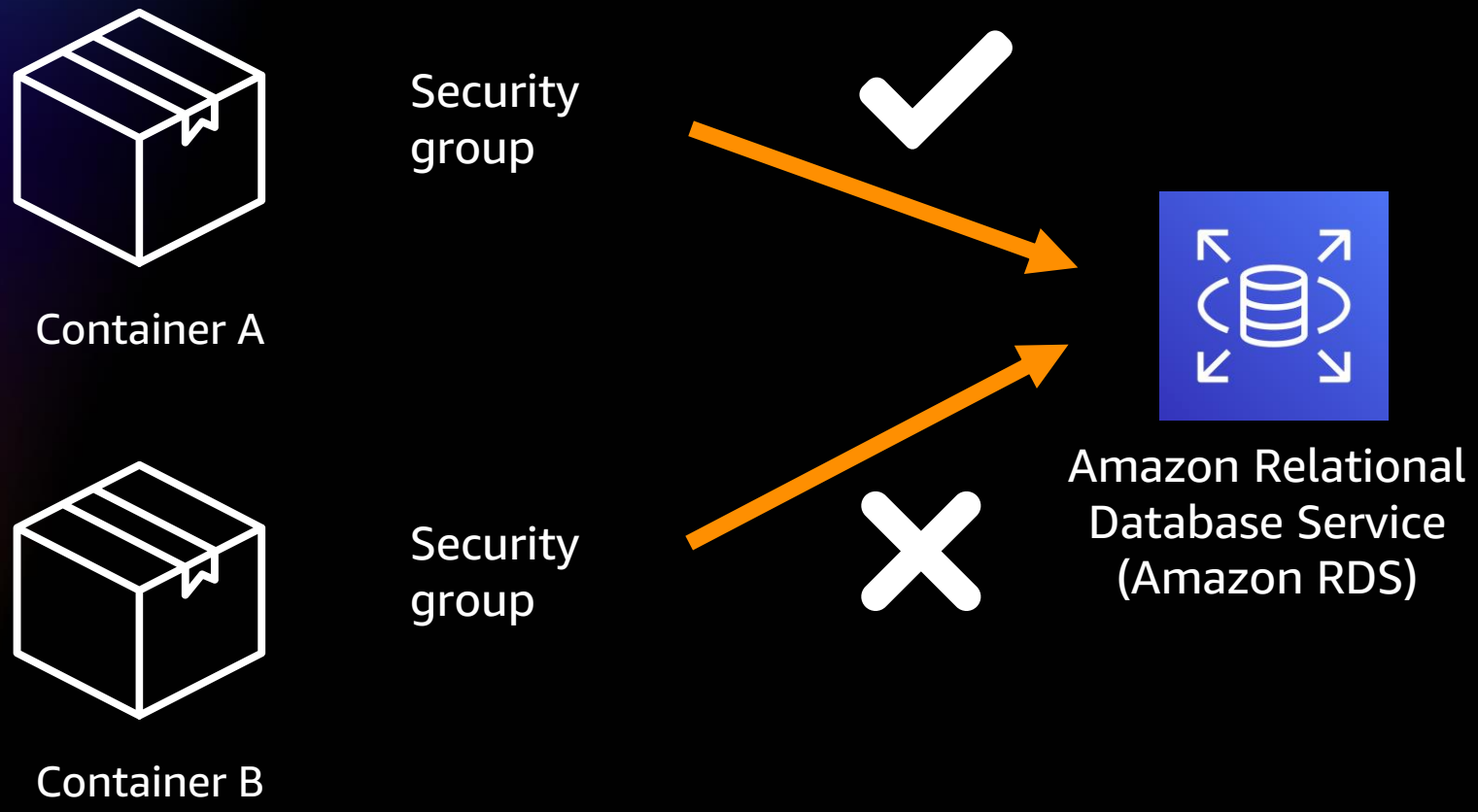
AWS Fargate

Decoupling with containers for security

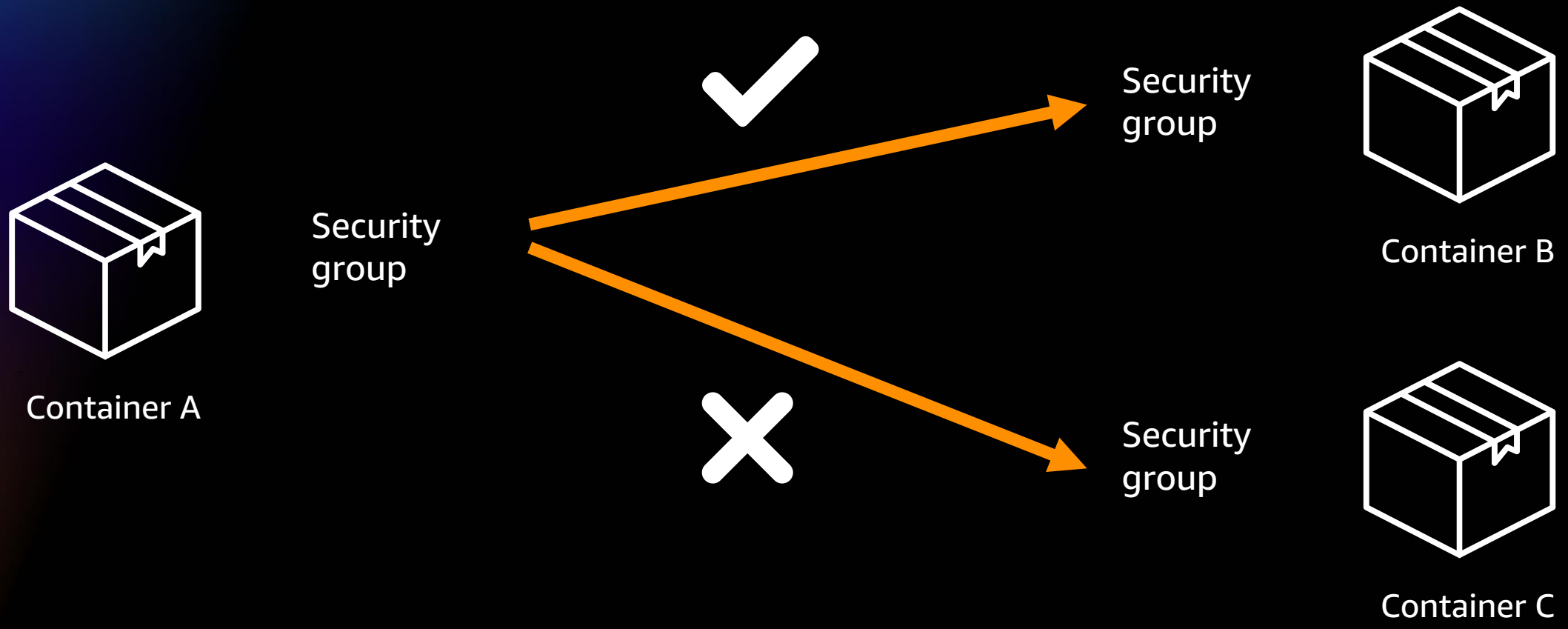
Containers can have their own security groups and IAM roles



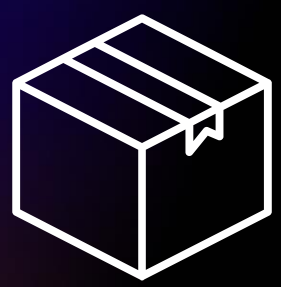
Control access to databases on a service by service basis



Control access from one service to another



Control API level actions for services

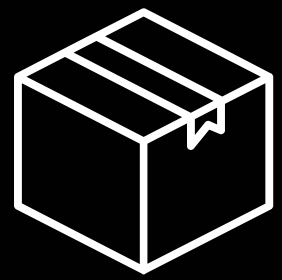


Job producer

IAM Role

Amazon Simple Queue Service (Amazon SQS)

IAM Role



Job consumer

Producer is allowed to add items to queue, but not read queue

Consumer is allowed to read queue but not add jobs

Practical decoupling tips

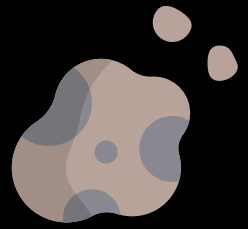
Practical decoupling: from monolith to microservice

Functioning monolith

Trying to break things up too fast is a recipe for disaster

Kind of works?

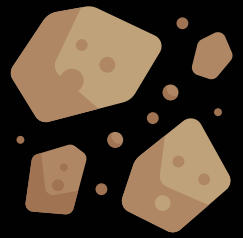
Buggy service



Semi-functioning microservice

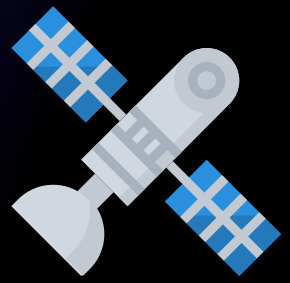
Boom!

Broken implementation

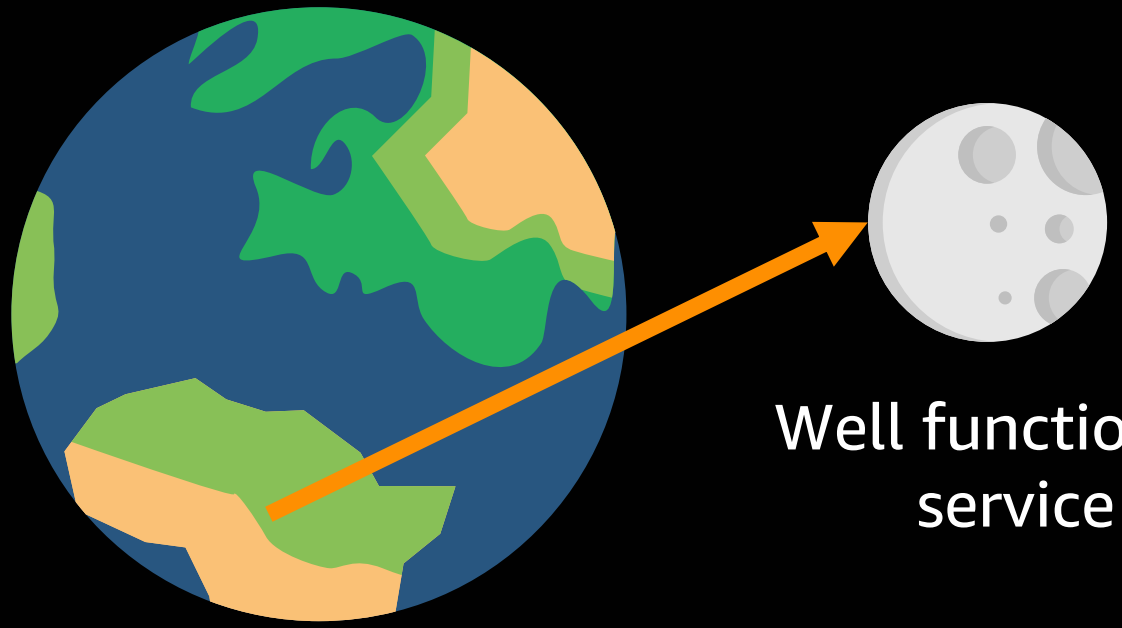


This part works great!

Decouple gradually, leave the central monolith for a while



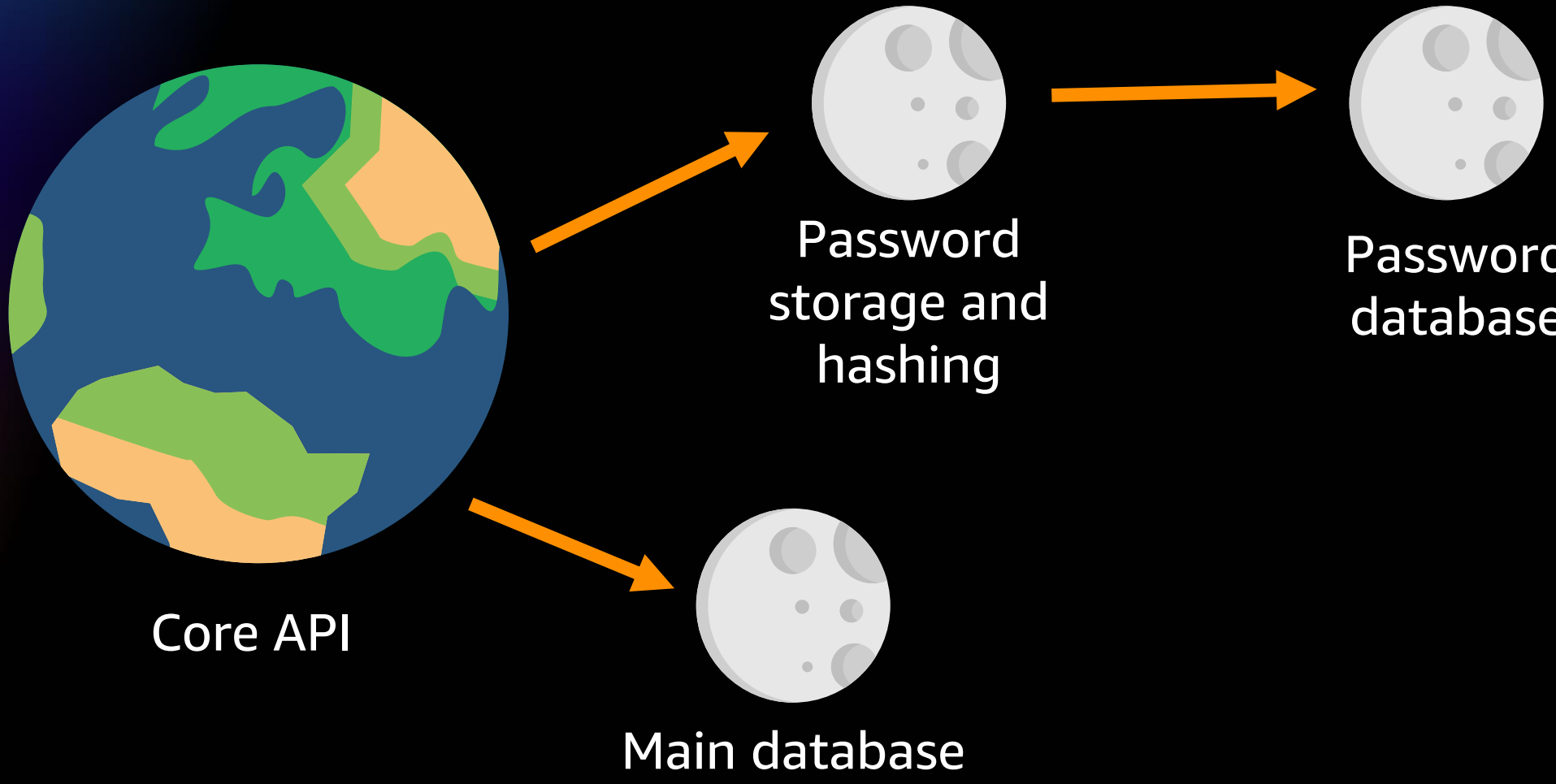
Efficient microservice



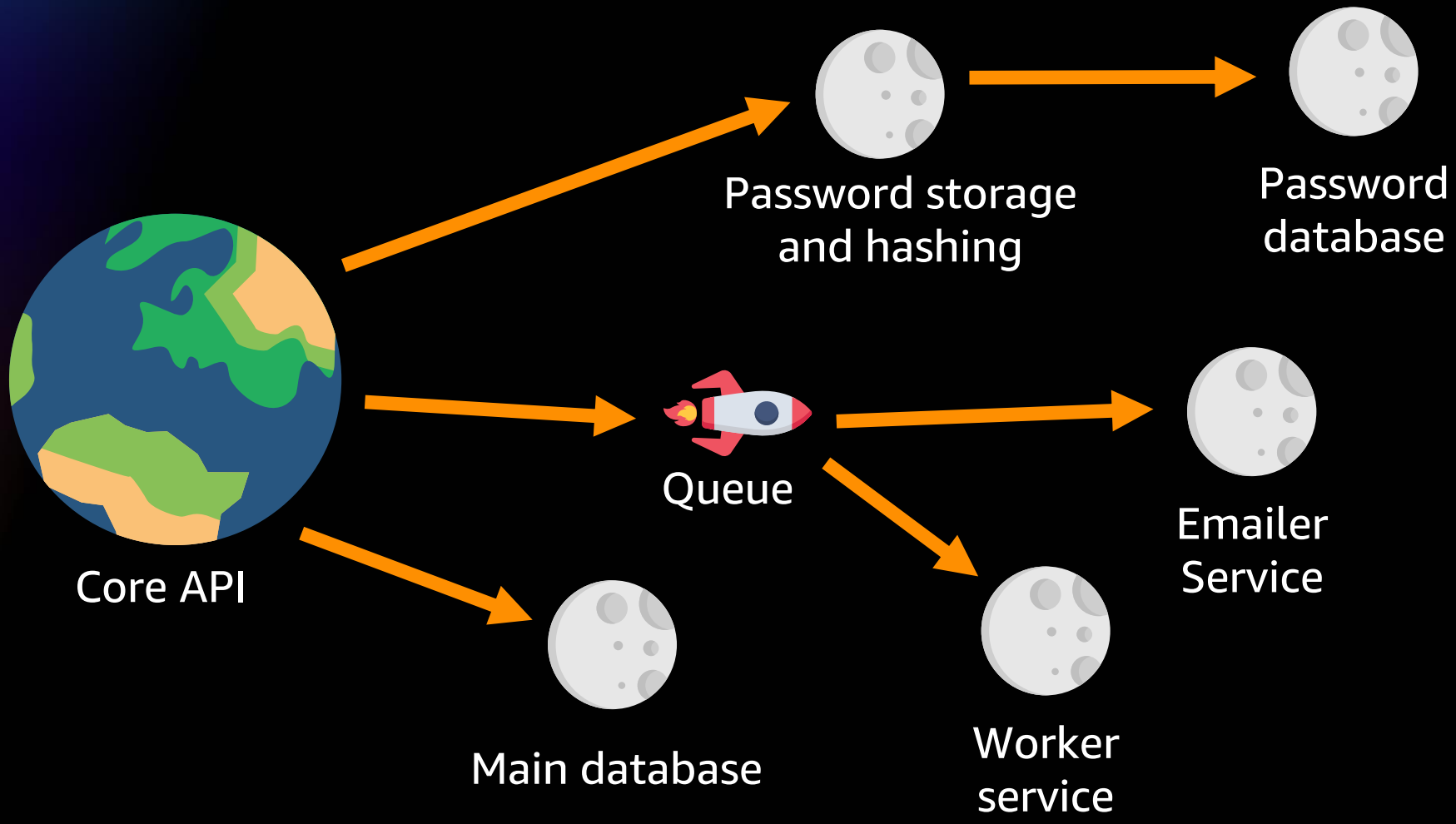
Functioning monolith

Well functioning service

Some practical places to start: User signup



Some practical places to start: User signup



Where to start?

When deciding what parts of your app to spin out of the monolith you should look for transactions that:

- Can be made asynchronous
  - vs. a monolith that is mainly synchronous
- Have well above average response times
  - For example, just that transaction could be re-written in Rust/Go vs. Python
- Have different resource requirements or scaling needs
  - For example, just that transaction of the app could benefit from expensive GPUs

Overview of AWS container services

Operating containers at scale is challenging

Security

Do we have vulnerabilities on our hosts?

Operating containers at scale is challenging

Security

Do we have vulnerabilities on our hosts?

Maintenance

How are we handling ongoing AMI management, logging, & monitoring?

Operating containers at scale is challenging

Security

Do we have vulnerabilities on our hosts?

Maintenance

How are we handling ongoing AMI management, logging, & monitoring?

Capacity

Is the size of our cluster properly sized and can we scale as-needed?

Operating containers at scale is challenging

Security

Do we have vulnerabilities on our hosts?

Maintenance

How are we handling ongoing AMI management, logging, & monitoring?

Capacity

Is the size of our cluster properly sized and can we scale as-needed?

Cost

Are we being efficient with our spend?

Operating containers at scale is challenging

Security

Do we have vulnerabilities on our hosts?

Maintenance

How are we handling ongoing AMI management, logging, & monitoring?

Capacity

Is the size of our cluster properly sized and can we scale as-needed?

Cost

Are we being efficient with our spend?

Focus

Do we spend more time on our infrastructure than our applications?

Choosing your container environment

Amazon ECS

Powerful simplicity

- Fully managed containers orchestration
- Opinionated solution for containers
- Reduced time to build and deploy
- Fewer decisions needed

Choosing your container environment

Amazon ECS

Powerful simplicity

- Fully managed containers orchestration
- Opinionated solution for containers
- Reduced time to build and deploy
- Fewer decisions needed

Amazon EKS

Open flexibility

- If you are invested in Kubernetes
- Vibrant ecosystem and community
- Consistent open-source APIs
- Easier to run K8s resiliently and at-scale

Choosing your container environment

Amazon ECS

Powerful simplicity

- Fully managed containers orchestration
- Opinionated solution for containers
- Reduced time to build and deploy
- Fewer decisions needed

Amazon EKS

Open flexibility

- If you are invested in Kubernetes
- Vibrant ecosystem and community
- Consistent open-source APIs
- Easier to run K8s resiliently and at-scale

AWS Fargate

Serverless

- No servers to manage
- Pay only for resources when used
- Eliminate capacity planning
- Supports both Amazon EKS and Amazon ECS

Choosing your container environment

Amazon ECS

Powerful simplicity

- Fully managed containers orchestration
- Opinionated solution for containers
- Reduced time to build and deploy
- Fewer decisions needed

Amazon EKS

Open flexibility

- If you are invested in Kubernetes
- Vibrant ecosystem and community
- Consistent open-source APIs
- Easier to run K8s resiliently and at-scale

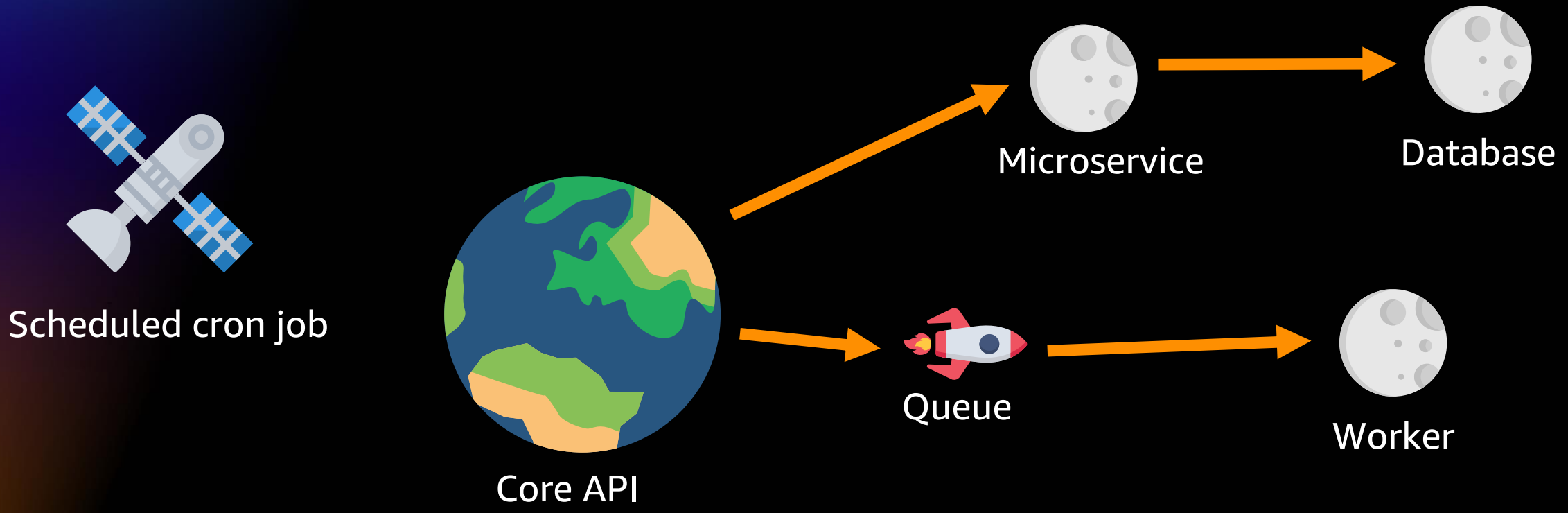
AWS Fargate

Serverless

- No servers to manage
- Pay only for resources when used
- Eliminate capacity planning
- Supports both Amazon EKS and Amazon ECS

And many customers run a mix of all three!

Decouple workloads responsibly - And it is okay to have a central monolith!



Visit the Modern Applications Resource Hub for more resources

Dive deeper with these resources to help you develop an effective plan for your modernization journey.

- Build modern applications on AWS e-book
- Build mobile and web apps faster e-book
- Modernize today with containers on AWS e-book
- Adopting a modern Dev+Ops model e-book
- Modern apps need modern ops e-book
- Determining the total cost of ownership: Comparing Serverless and Server-based technologies paper
- Continuous learning, continuous modernization e-book
- … and more!

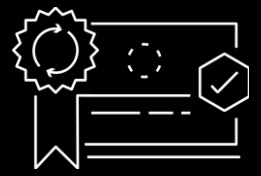
https://bit.ly/3yfOvbK

Visit resource hub »

AWS Training and Certification

Accelerate modernization with continuous learning

Free digital courses, including:
Architecting serverless solutions
Getting started with DevOps on AWS



Earn an industry-recognized credential:
AWS Certified Developer – Associate
AWS Certified DevOps – Professional

Hands-on classroom training (available virtually) including:
Running containers on Amazon Elastic Kubernetes Service (Amazon EKS)
Advanced developing on AWS

Create a self-paced learning roadmap
AWS ramp-up guide - Developer
AWS ramp-up guide - DevOps

Take Developer and DevOps training today

Learn more about Modernization training for you and your team

Thank you for attending AWS Innovate Modern Applications Edition

We hope you found it interesting! A kind reminder to **complete the survey.**
Let us know what you thought of today's event and how we can improve the event experience for you in the future.


aws-apj-marketing@amazon.com


twitter.com/AWSCloud


facebook.com/AmazonWebServices


youtube.com/user/AmazonWebServices


slideshare.net/AmazonWebServices


twitch.tv/aws

Thank you!