
aws
SUMMIT
INDIA | MAY 25, 2023

BMARC001

Enterprise Architectures - State of the Union

Mani Chandrasekaran
Principal Solutions Architect – India and South Asia
AWS India

Agenda

- The value of architecture patterns
- Common integration patterns
- Application patterns – Serverless and Containers
- What next?

AWS Global Infrastructure

Why customers choose AWS

Most experience

16

years helping millions of customers

Global reach & high availability

99

availability zones spanning 31 geographic regions

Security & compliance

300+

security features

Customer obsession & innovation

200+

service offerings

AWS's Infrastructure is

3x More Energy Efficient

than the median of surveyed U.S. enterprise data centers

Improve TCO

111

price reductions since 2006

Machine learning

81%

of all deep learning is running on AWS[1]

Ecosystem

4,500

software listings from 1,400 ISVs

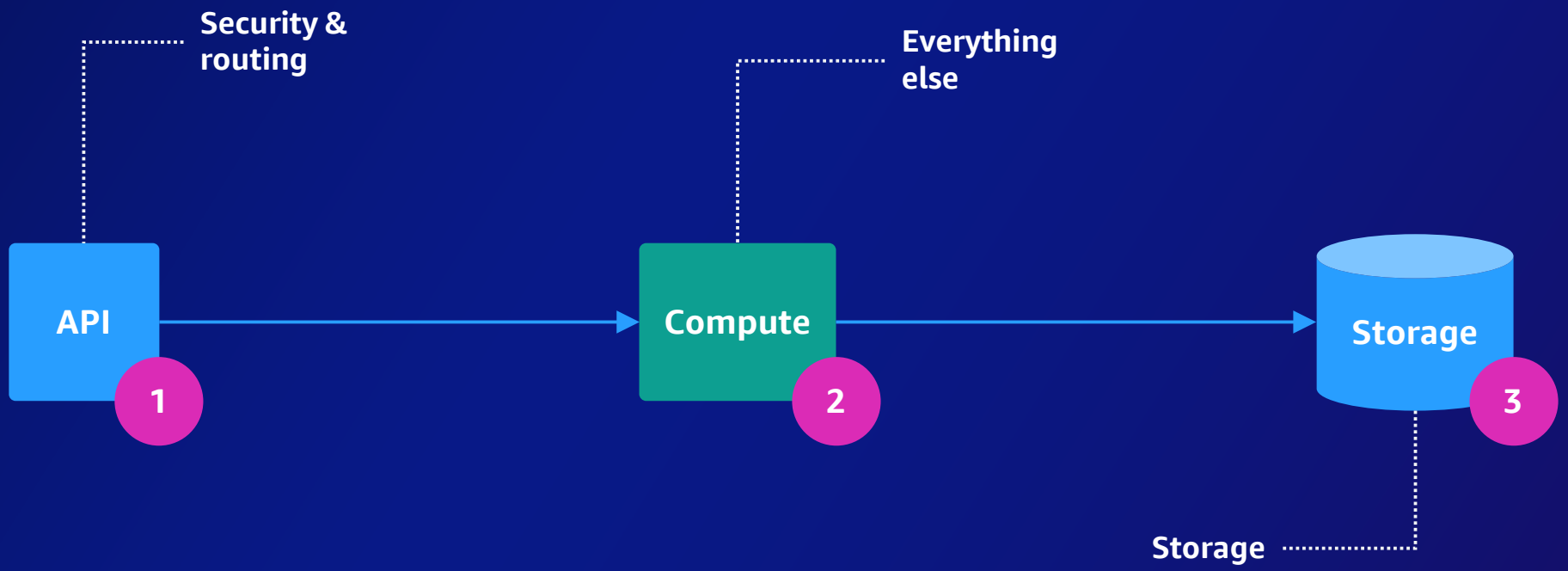
AWS Global Infrastructure

31 Launched Regions, 99 Availability Zones and 450+ Points of Presence

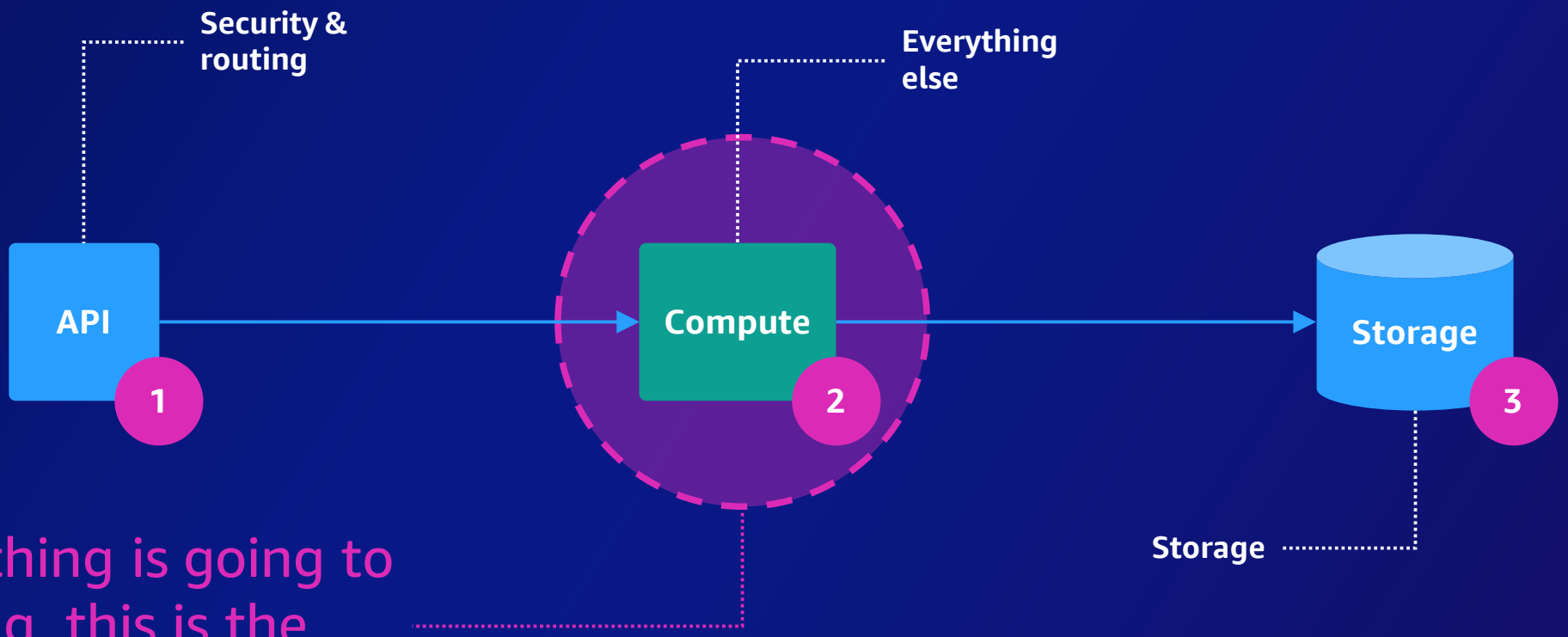
Latest Region & number of Availability Zones (AZs) can be found here [AWS global infrastructure page]

Architecture patterns in enterprise architectures

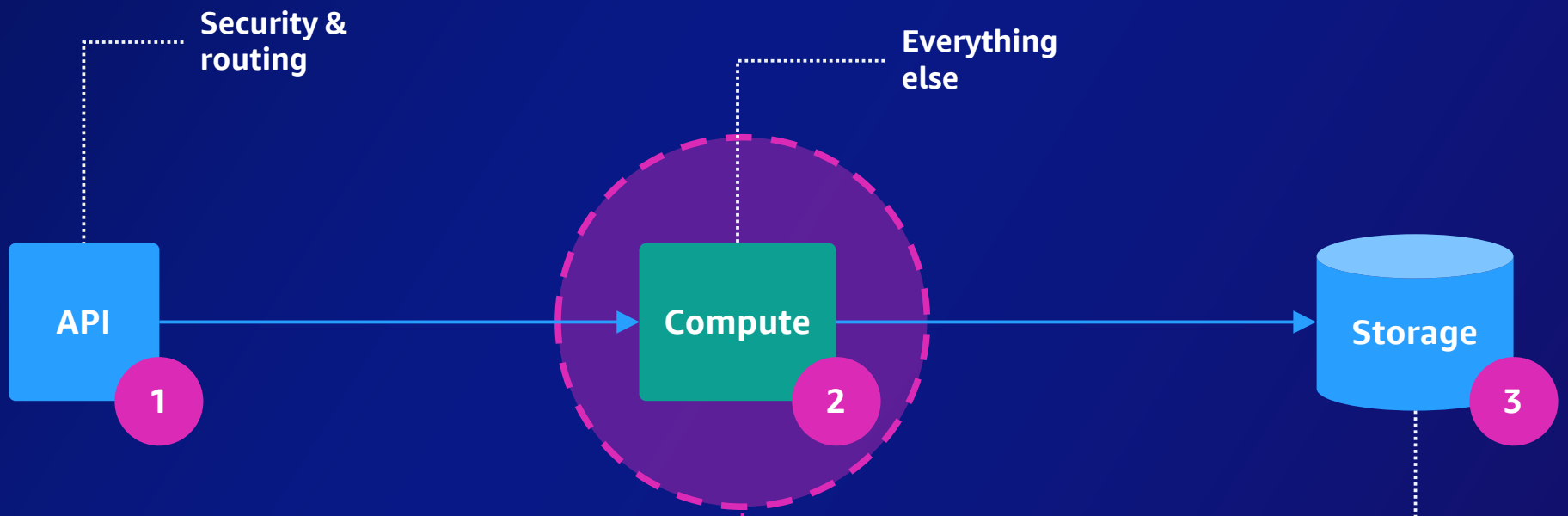
Application elements



Application elements



Application elements

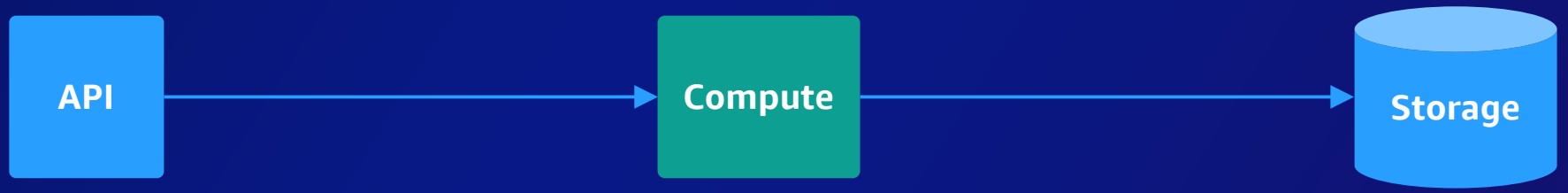


If something is going to go wrong, this is the most probable place

Why?

Because it is MY code!

How do we improve this?



Synchronous systems

- Are inherently tightly coupled
- Problems downstream can immediately impact upstream
- Retries from upstream callers can amplify problems

Photo: Dirk Fröhner

Synchronous systems

- Are inherently tightly coupled
- Problems d
  immediatel
- Retries fron
  amplify pro

…and some things simply take too much time to wait, or are asynchronous by nature!

Photo: Dirk Fröhner

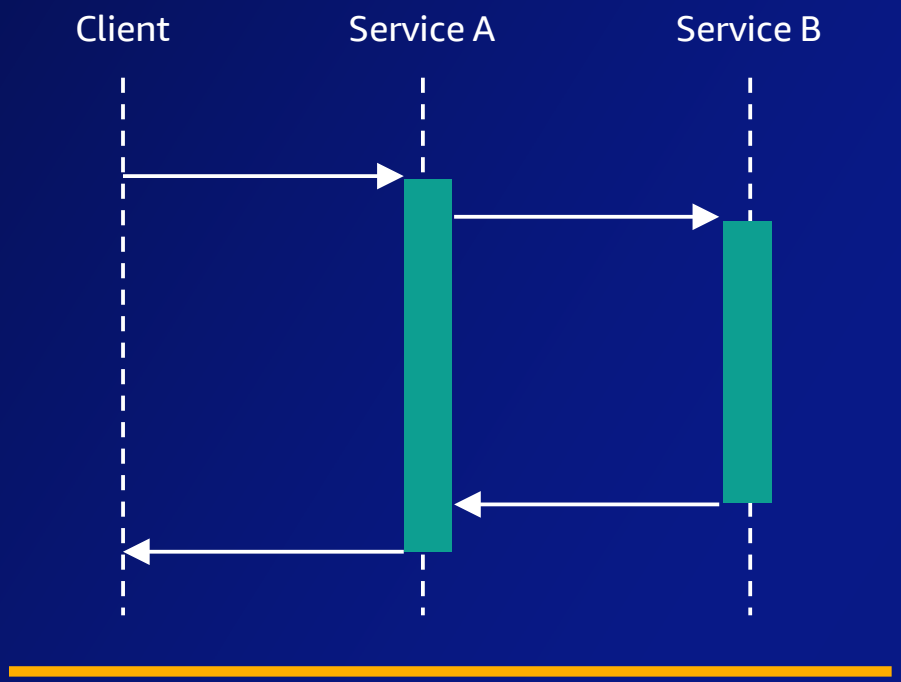
> “If your application is cloud-native, or large-scale, or distributed, and doesn’t include a messaging component, that’s probably a bug”

Tim Bray

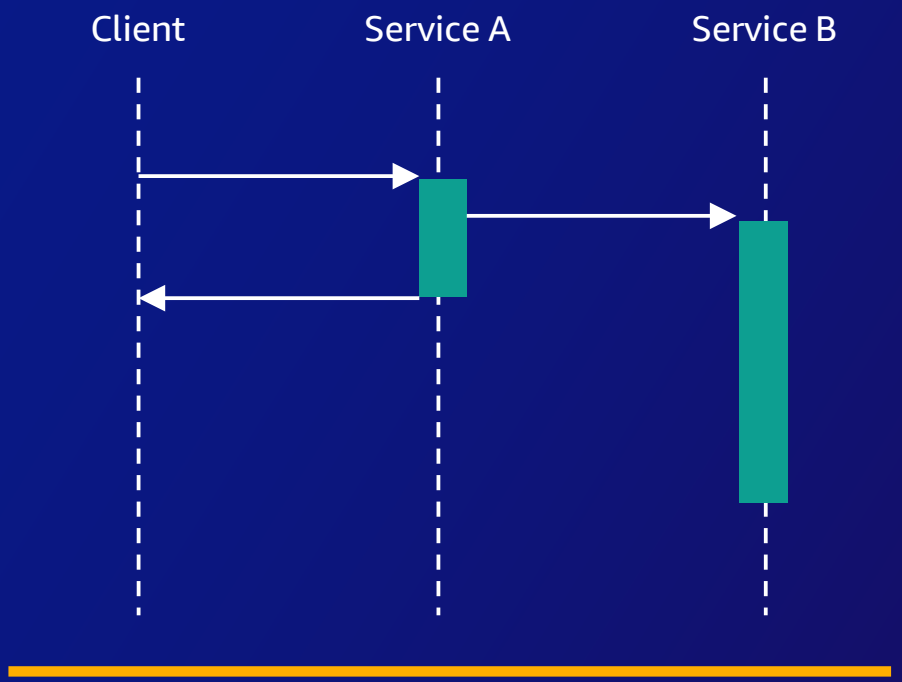
General-purpose internet-software geek

Thinking asynchronously

Think more asynchronously



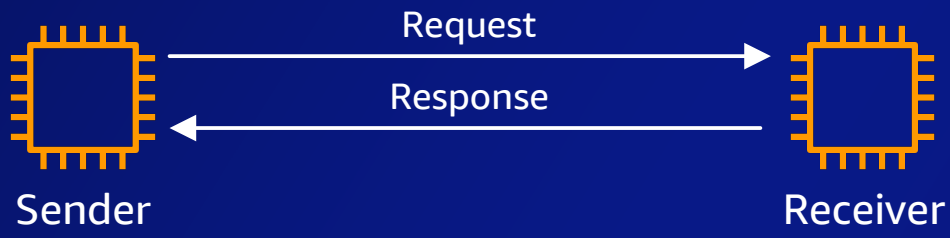
Synchronous commands



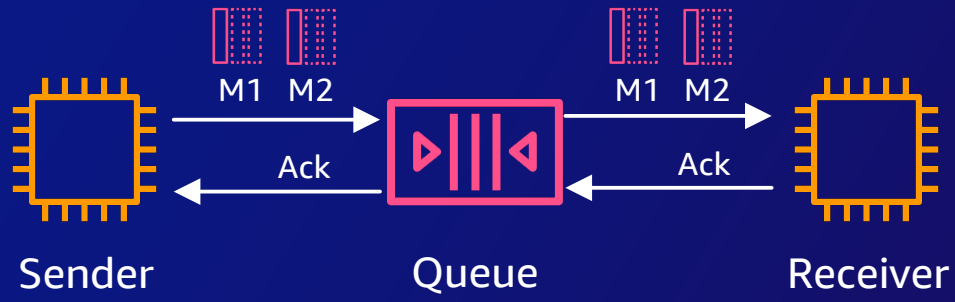
Asynchronous events

If you don’t need a response, execute asynchronously

Synchronous



Asynchronous



Integration patterns

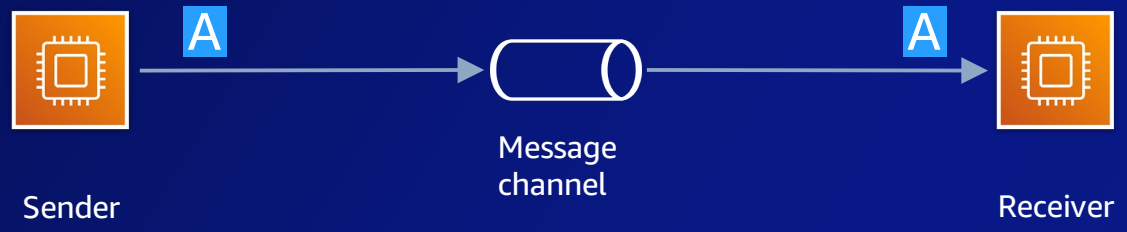
In modern cloud applications, integration isn't an afterthought. It's an integral part of the application architecture and the software delivery lifecycle.

Gregor Hohpe

Author of *Enterprise Integration Patterns*, *Cloud Strategy*, and *The Software Architect Elevator*

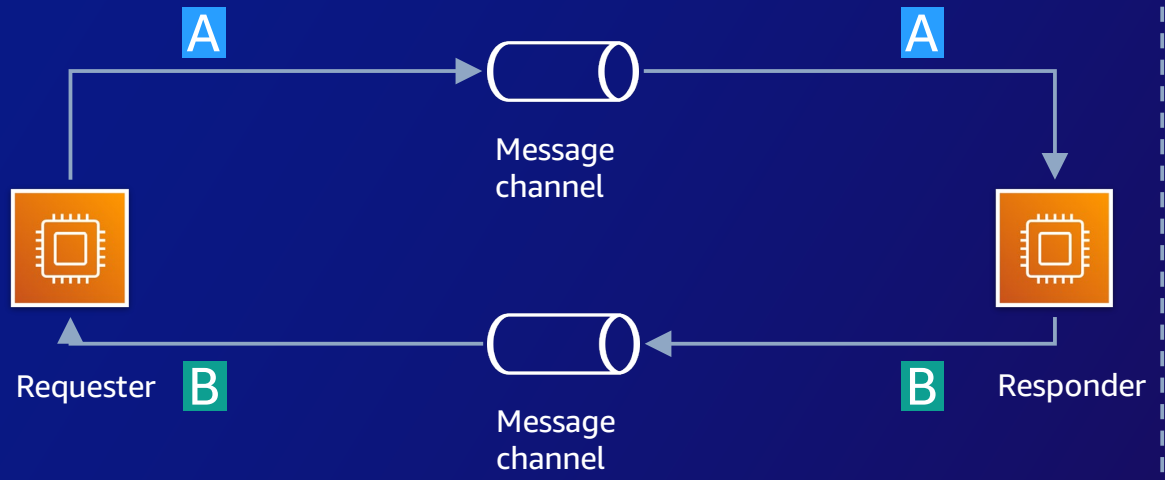
Message exchange

One-way



No response expected

Request-response



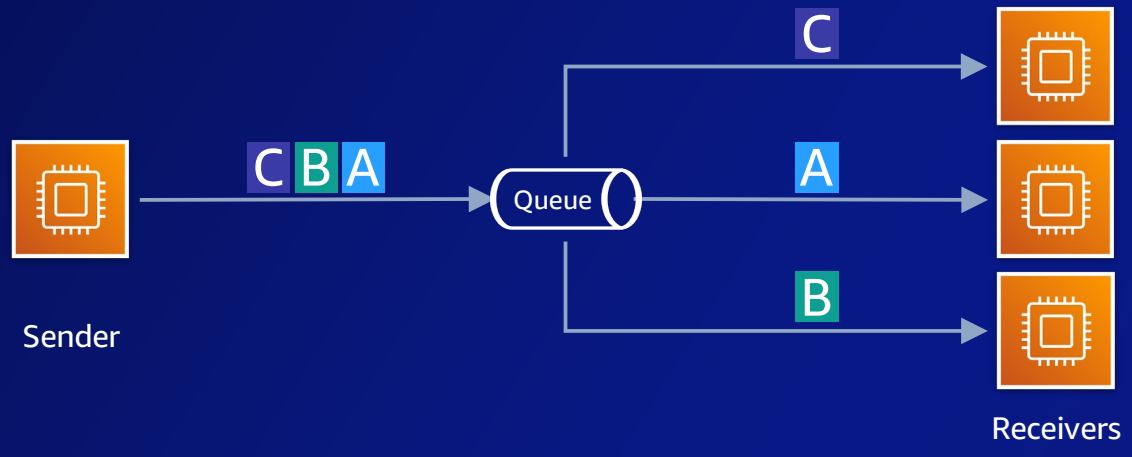
Response expected

Return address

Correlation ID

Message channels

Point-to-point (queue)

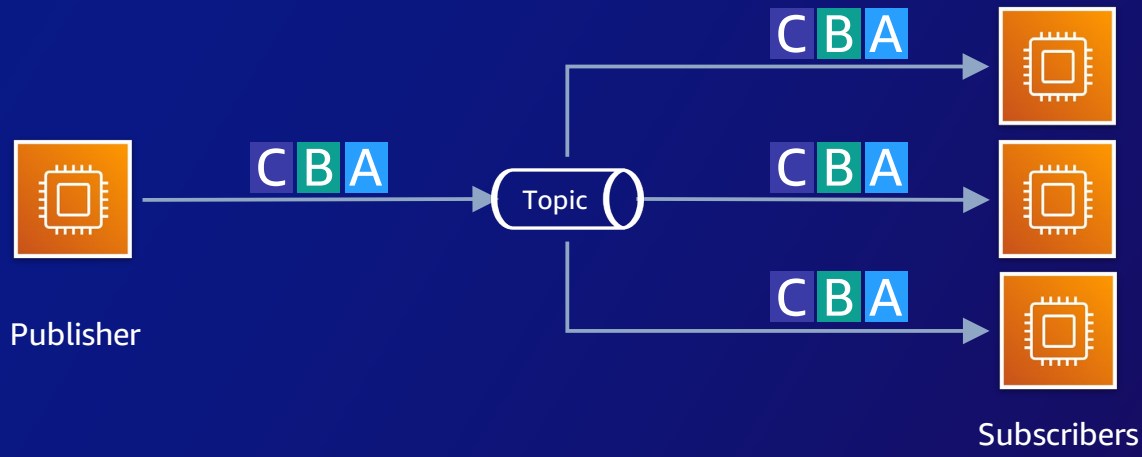


Consumed by one receiver

Easy to scale

Flatten peak loads

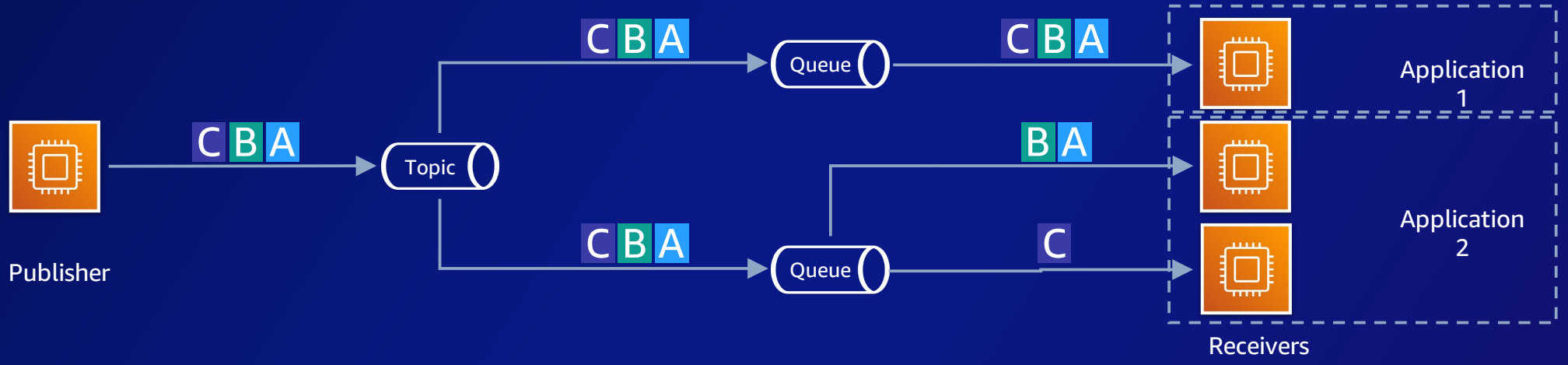
Publish-subscribe (topic)



Consumed by all subscribers

Message channels

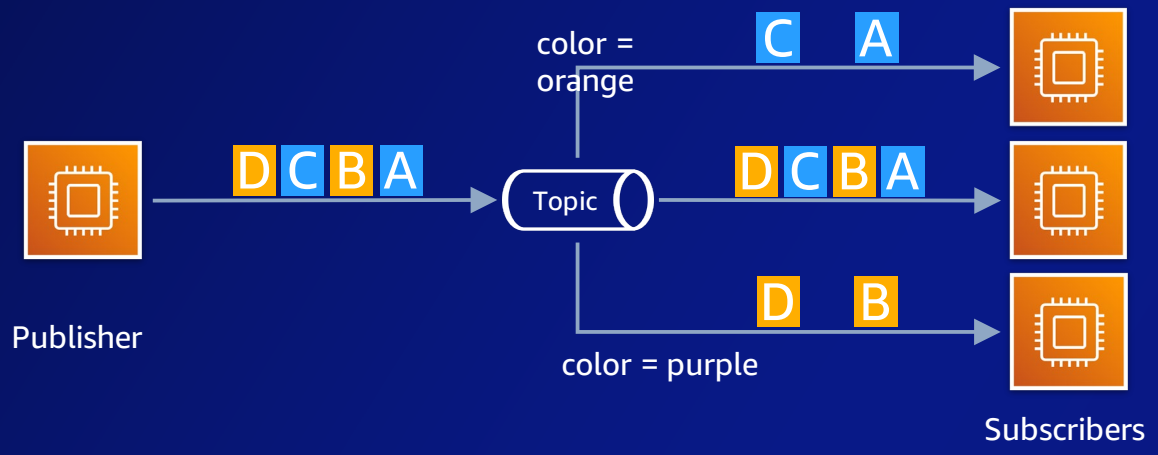
Topic-queue-chaining



Allows fan-out and receiver scale-out at the same time

Message routing

Message filter

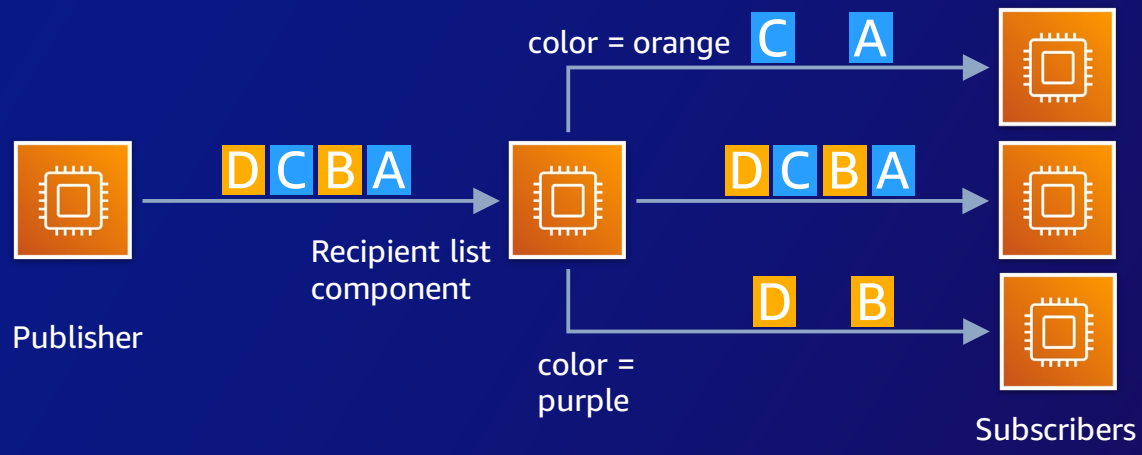


Receive only a relevant subset of messages

Controlled by subscriber

Publisher remains completely unaware

Recipient list



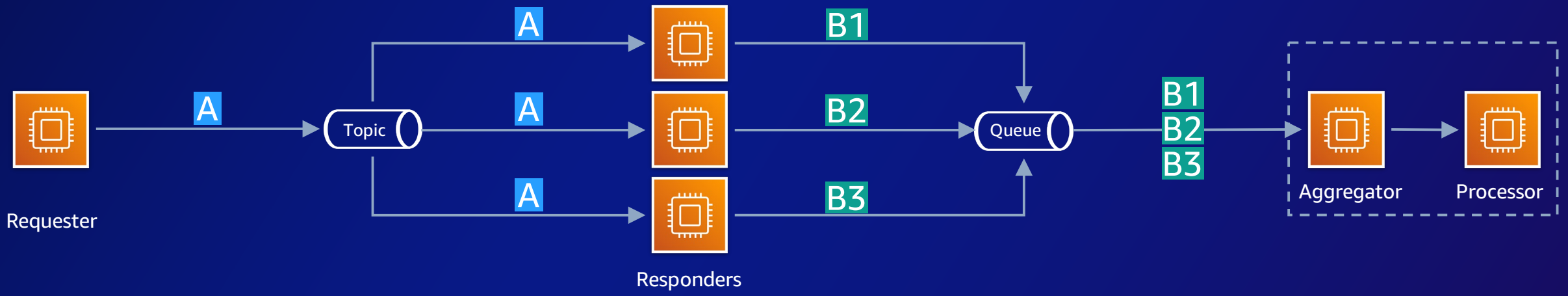
Send only a relevant subset of messages to a subscriber

Controlled by publisher or separate component

Potentially adds coupling

Message routing

Scatter-gather



How to distribute a request across potentially interested/relevant parties and capture their individual responses?

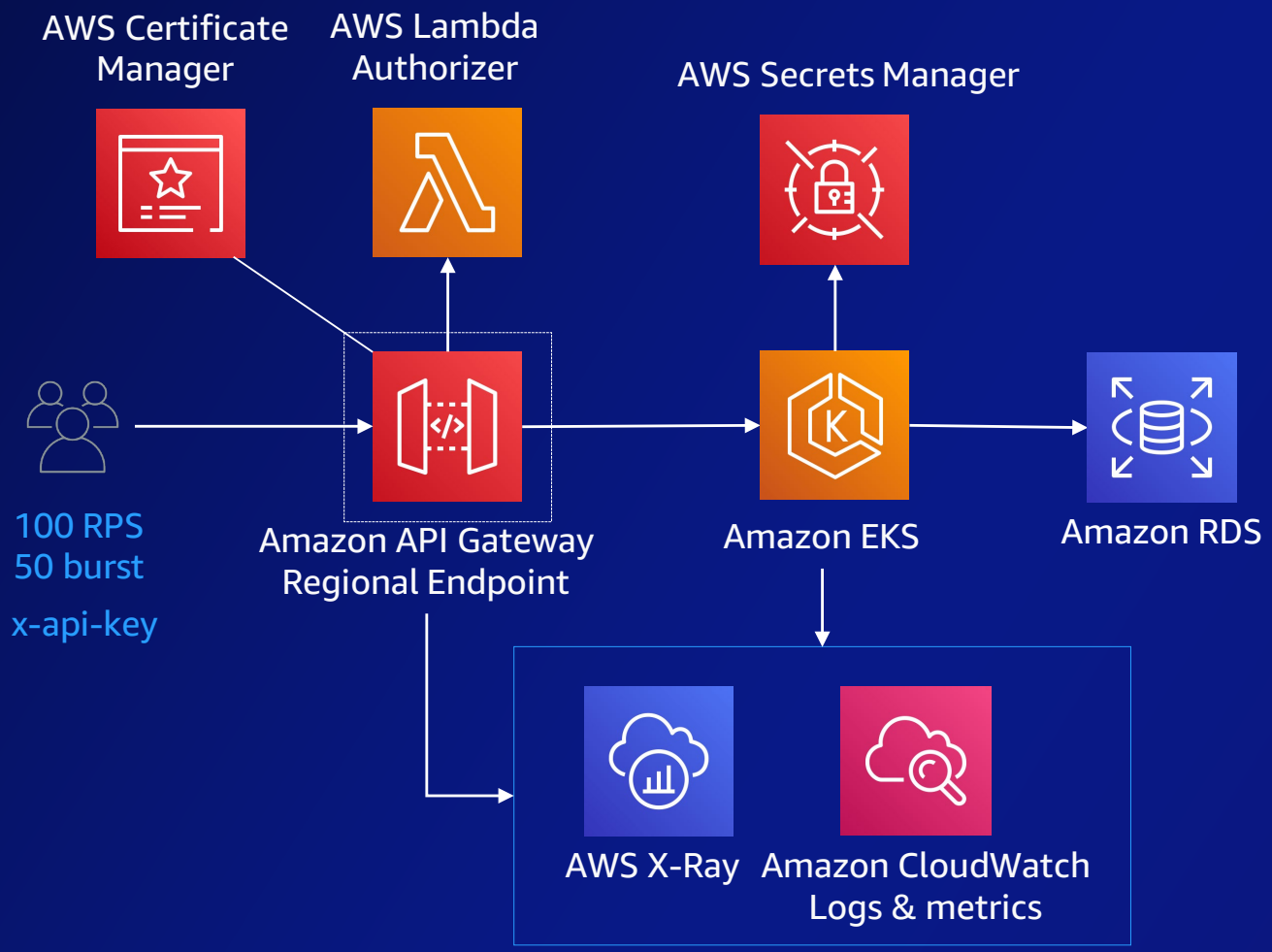
Election or parallel processing scenarios, i.e., search for best response or accumulate responses

Application patterns – Serverless and Containers

REST API pattern

REST API

REST API with Containers



- Enable access logs, execution logs for your APIs
- API Gateway is well integrated with AWS X-Ray and CloudWatch
- Instrument your backend services to capture X-Ray traces and create metrics with CloudWatch Embedded Metric Format or Prometheus Exposition Format
- Regulate inbound access rates
- Authorize consumers. Enforce API keys
- Secure end points with SSL certificates
- Manage secrets with AWS Secrets Manager
- Regional endpoints support HTTP2

REST API with AWS Lambda

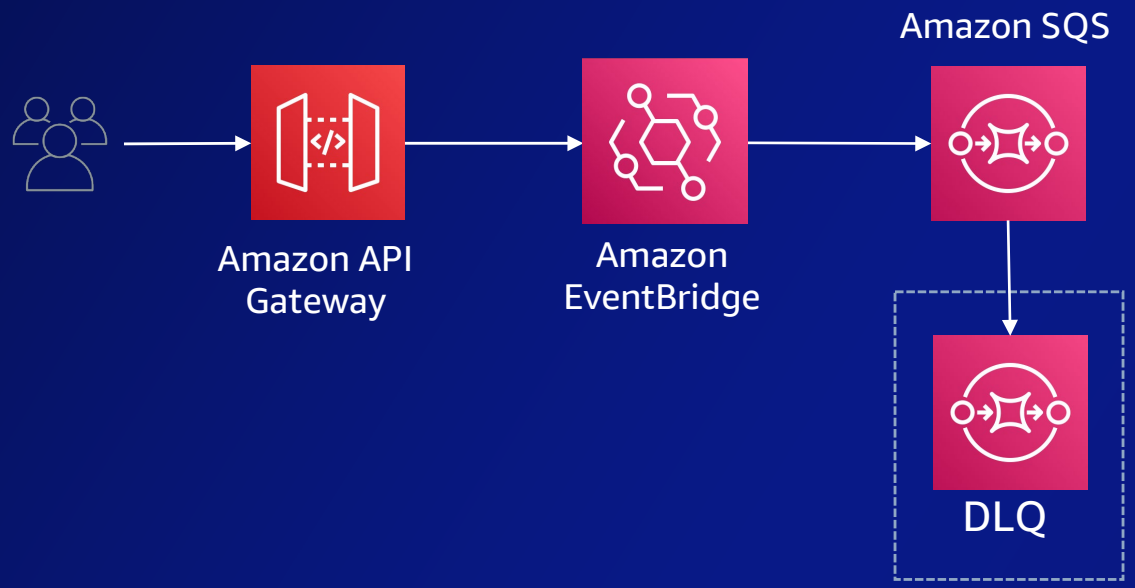
Best practices

- Enable access logs, structure logs and instrument your code
- Create metrics async with CloudWatch Embedded Metric Format
- Regulate inbound access rates
- Authorize consumers. Manage secrets with AWS Secrets Manager
- On-demand tables support up to 40K read/write request units
- Regional endpoints support HTTP2

OPERATIONS | RELIABILITY | SECURITY | PERFORMANCE | COST

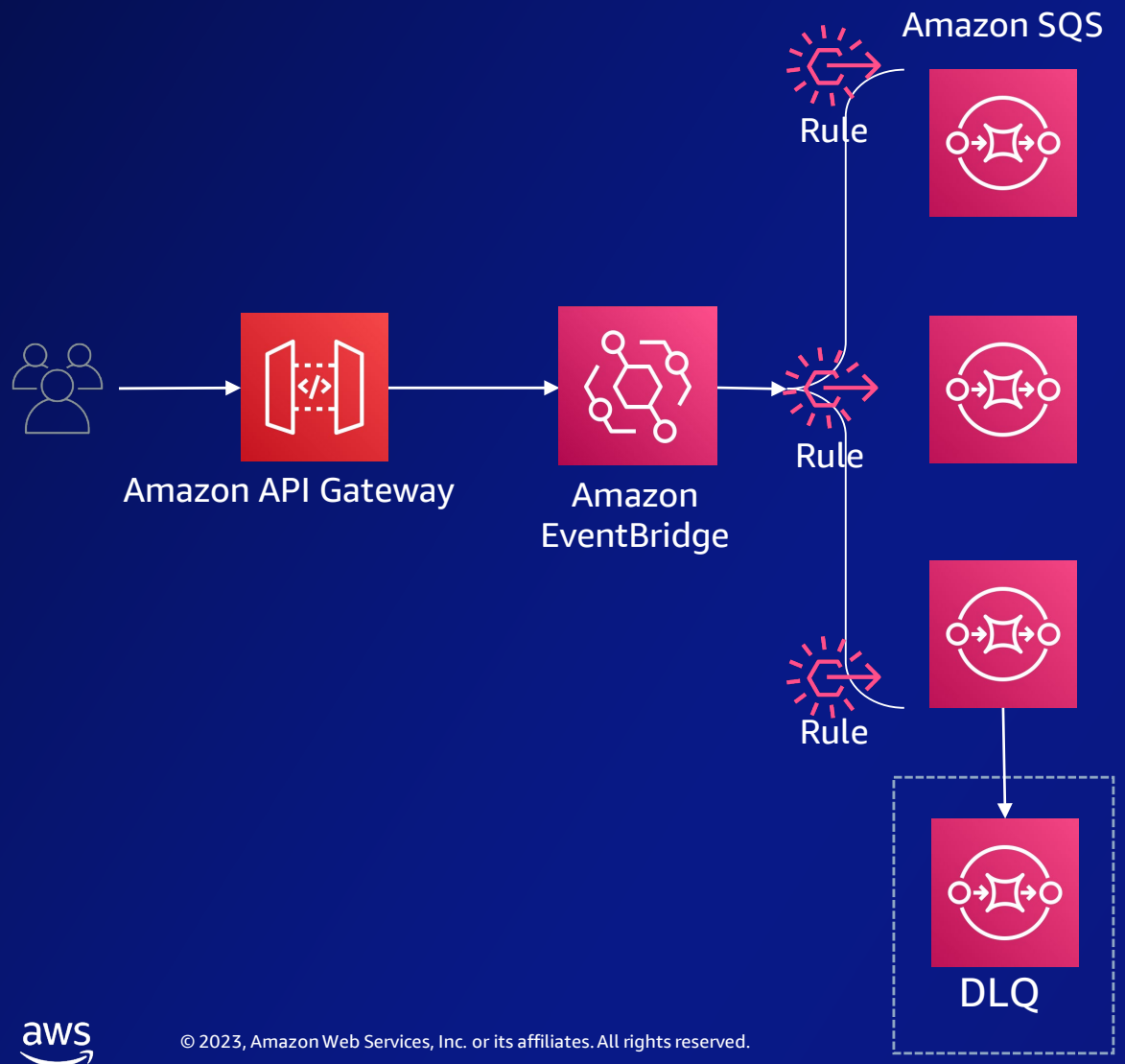
Fan-out pattern

Fan-out pattern for message processing



- API Gateway can integrate with AWS services directly
- EventBridge is a pub/sub service to decouple producer and consumer
- Publish notifications directly to EventBridge from API Gateway
- Integrate with Amazon SQS for higher durability, batching, and DLQ

Fan-out pattern for message processing



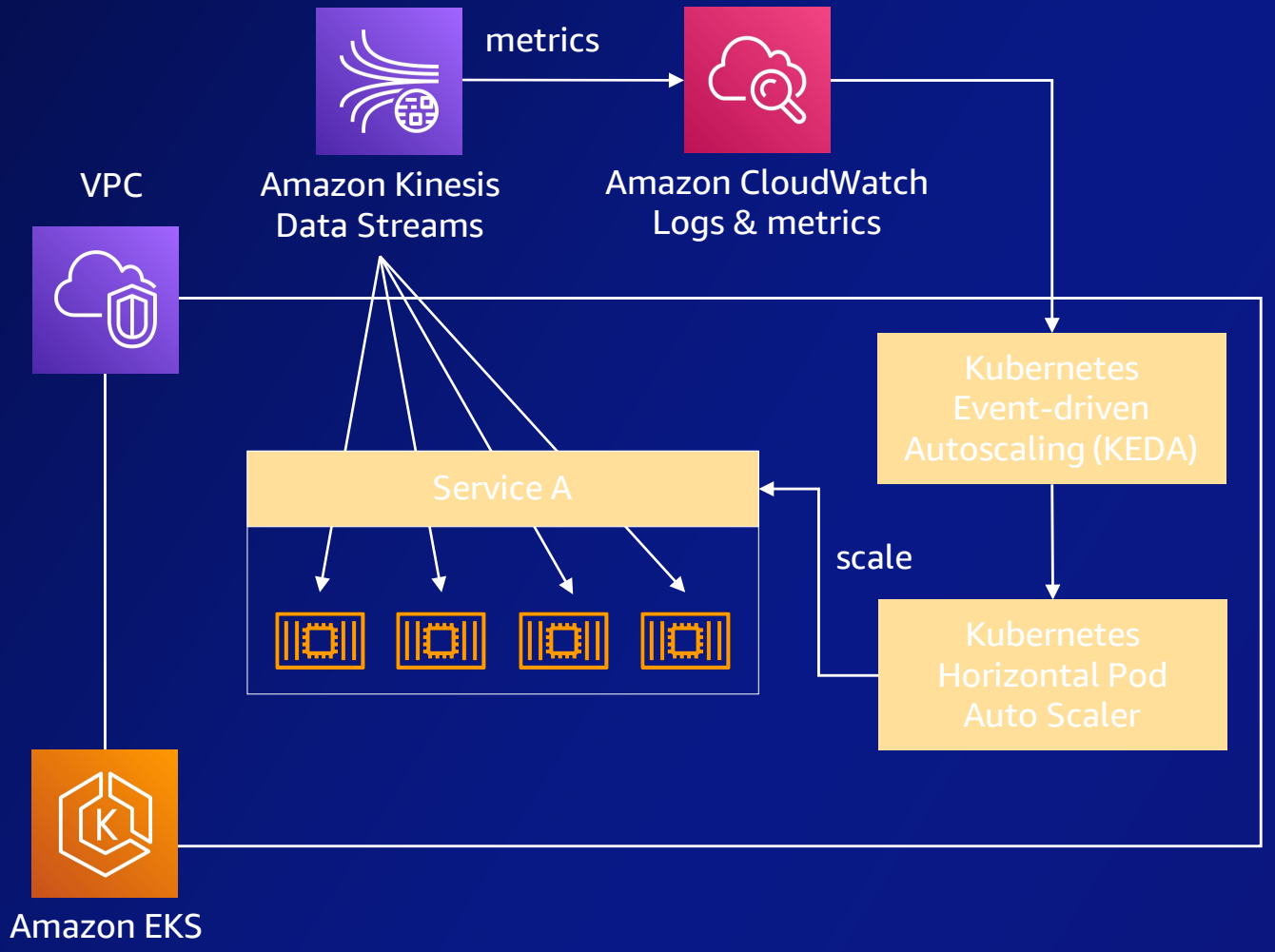
- API Gateway can integrate with AWS services directly
- Publish notifications directly to EventBridge
- Integrate with Amazon SQS for higher durability, batching, and DLQ
- Use EventBridge rules to separate messages into different SQS queues

Event streaming pattern

Event streaming

- Kinesis Data Streams is used to collect and process large streams of data records in real time.
- Use API Gateway REST API as a Kinesis proxy to write data directly into a specified stream
- Kinesis Data Stream application (*consumer*) reads data from the streams using Kinesis Client Library (KCL)
- Consumer applications are run in ECS/EKS

Event streaming



- Kinesis Data Streams application (*consumer*) reads records from the stream using KCL
- Maximum consumers scale out is equal to the number of *shards* in the data stream
- Consumers may be scaled dynamically based on CloudWatch metrics for Kinesis Data Streams
- KEDA and HPA can be used to scale the number of pods

Data orchestration pattern

Use cases

File/video/image processing

Take collection of video files and convert them to sizes and formats that are ideal for different devices

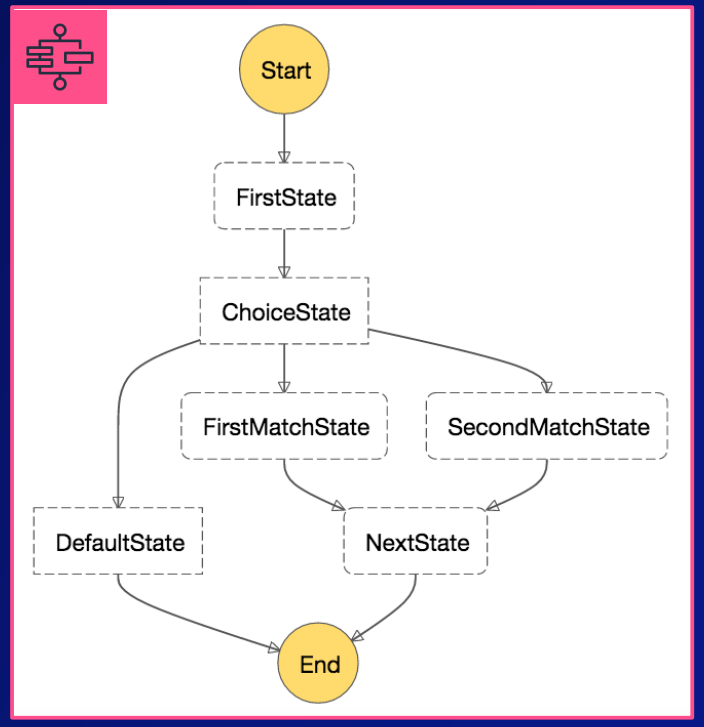
ETL/ELT pipelines

Combine sales opportunity records with marketing data to produce business intelligence report

Batch processing and HPC

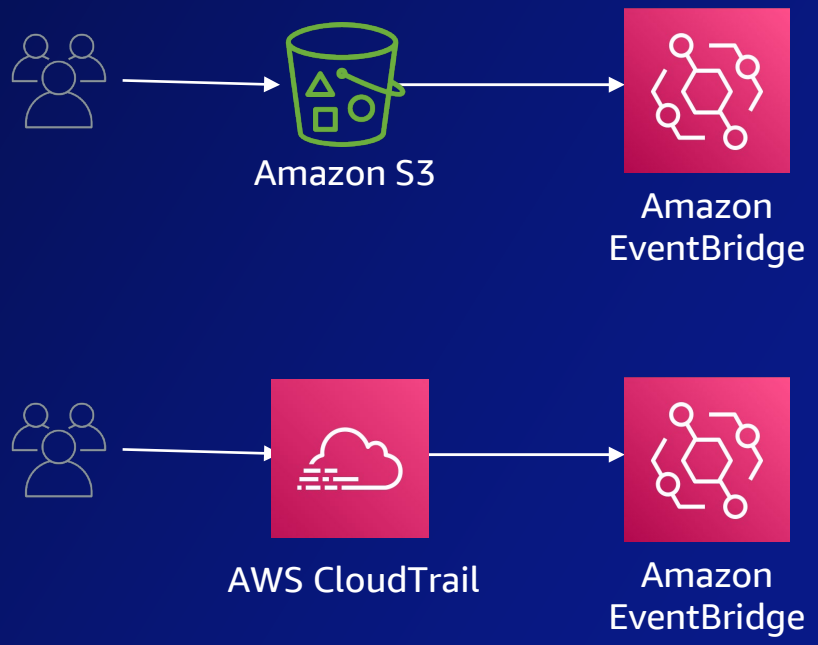
Build a genomic secondary analysis pipeline that processes raw whole genome sequences into variant calls

Orchestrate with AWS Step Functions



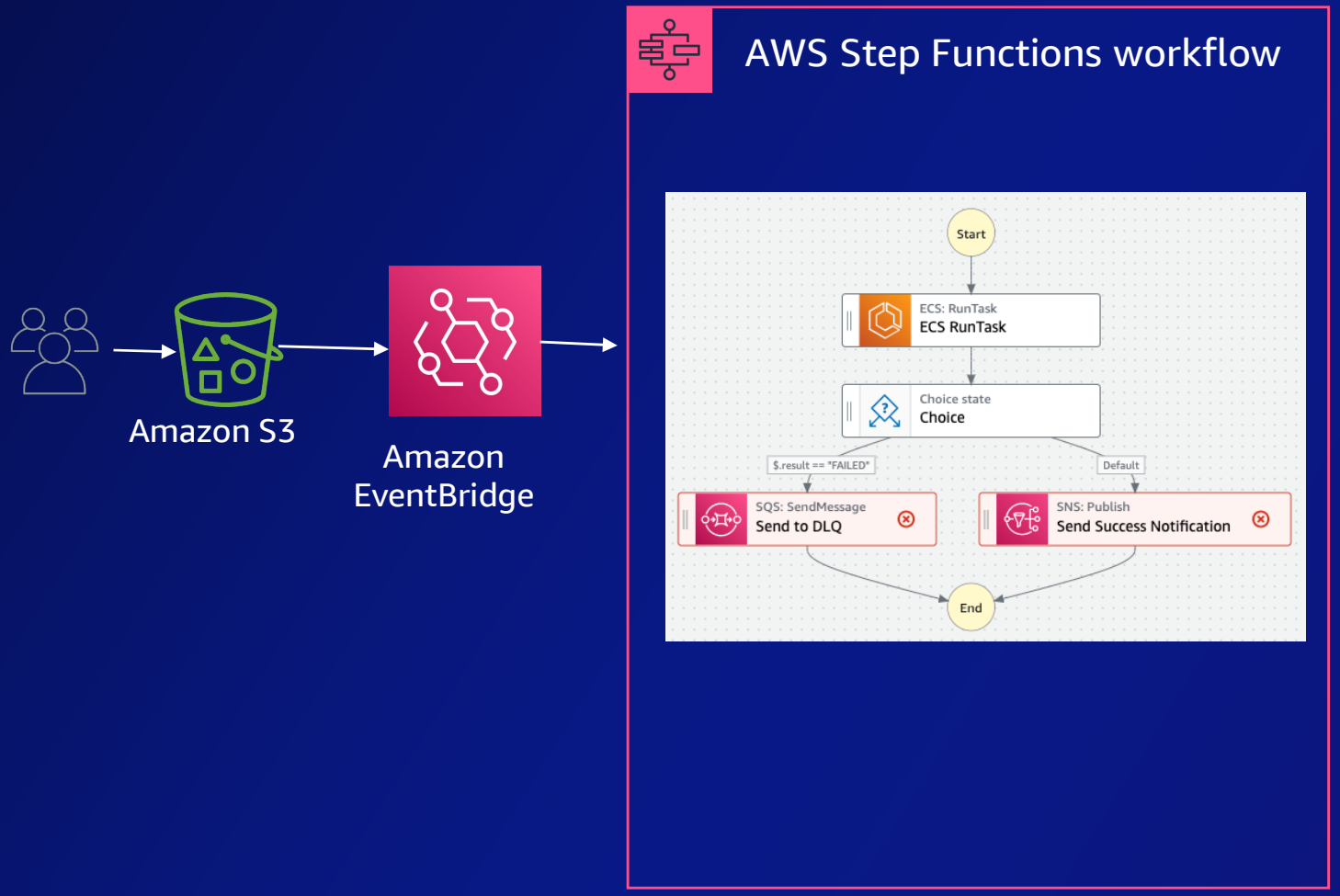
- Highly scalable and Serverless
- Low code orchestration
- Manages state of different activities
- Dependency management
- Integrates with 10k+ AWS API directly
- Built-in try/catch

Orchestrate on events from AWS



- Response to events from 110+ AWS services
- Example: Set Amazon S3 file notification for EventBridge when a new file is created

Orchestrate on events from AWS



- Response to events from 110+ AWS services
- Example: Set Amazon S3 file notification for EventBridge when a new file is created
- Set Rule in EventBridge EventBus to route to AWS Step Functions
- Build workflows with process the file, handle failures and success notifications
- 10k+ direct API calls including running task/pod in EKS and ECS

What next?

Start with existing patterns and modify

There are many sources for Cloud Native Architecture Patterns

- CDK Patterns - https://cdkpatterns.com
- Serverlessland Patterns Collection - https://serverlessland.com/patterns
- Construct Hub - https://constructs.dev
- AWS Compute Blog - https://aws.amazon.com/blogs/compute/
- AWS Samples - https://github.com/aws-samples/serverless-patterns
- Asynchronous Messaging Workshop - https://github.com/aws-samples/asynchronous-messaging-workshop

Create well-architected patterns

OPERATIONS	Understand the health and lifecycle of your application
RELIABILITY	Build resiliency and protect non-serverless resources
SECURITY	Focus on managing security boundaries and AppSec
PERFORMANCE	Optimize for low, steady, and/or peaks
COST	Factor in development, people, opportunity, and maintenance cost

Choose your pattern and go build!

skillbuilder.aws

Your time is now

Build in-demand cloud skills your way

Thank you!

Please complete the session survey

Mani Chandrasekaran

Principal Solutions Architect – India and South Asia

AWS India

manikach@amazon.com

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