Building Serverless microservices

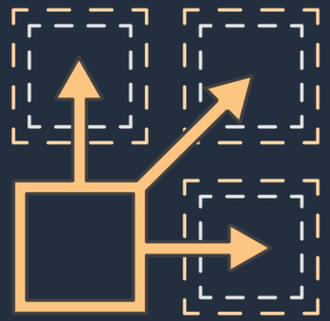
Cloud Native Application modernization journey

Anurag
Solutions Architect

Building the bridge but not advancing modernization

Pillars of Modernization

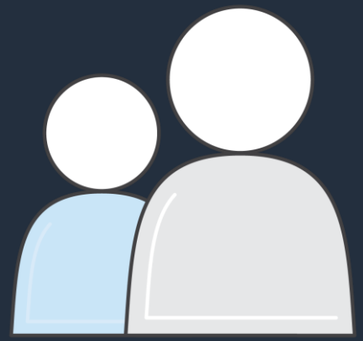
1



Technology & Architecture

Independent business functions

2



People, Process, & Culture

Organized for Value

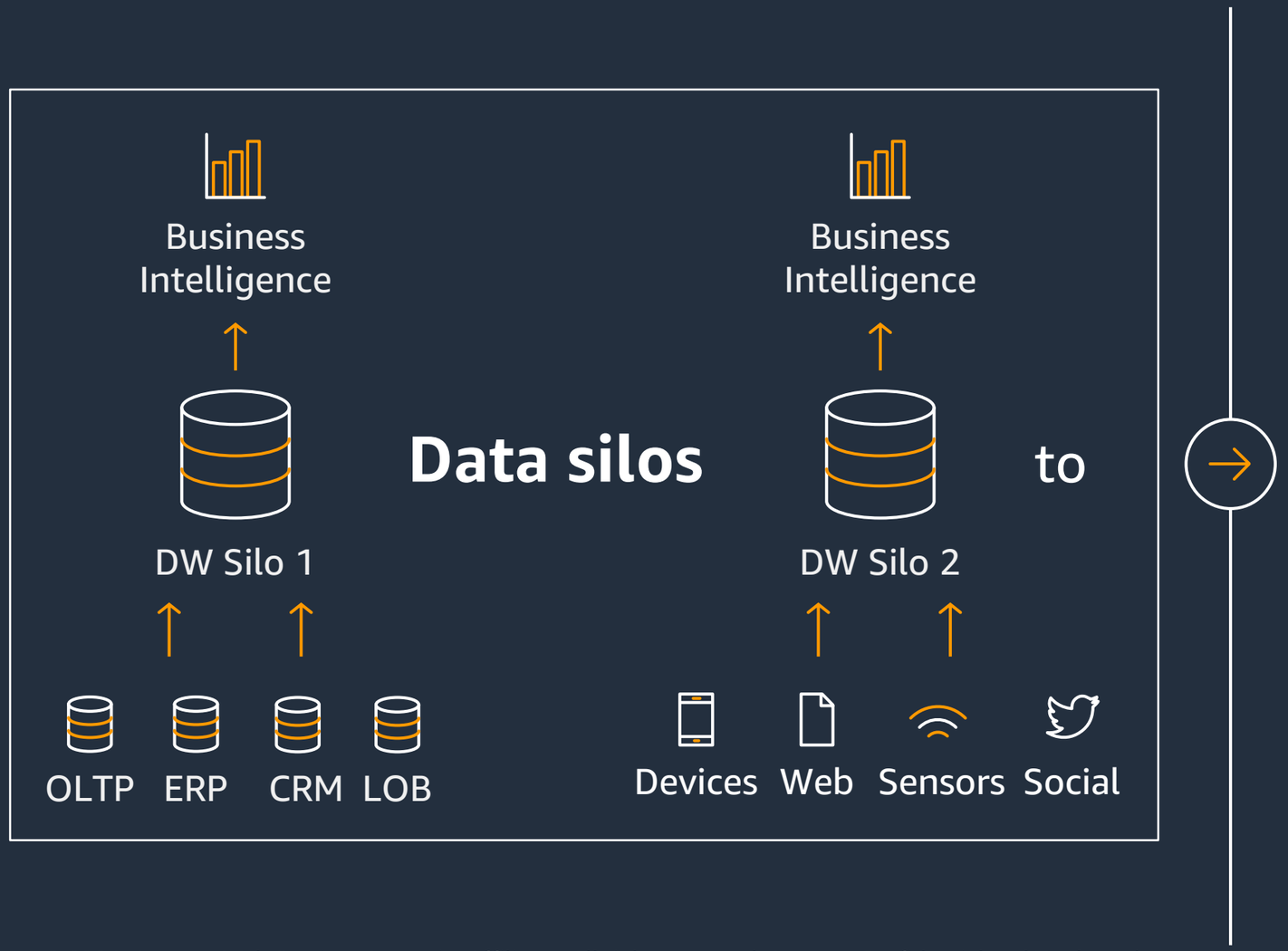
3

Ops & Governance at Scale

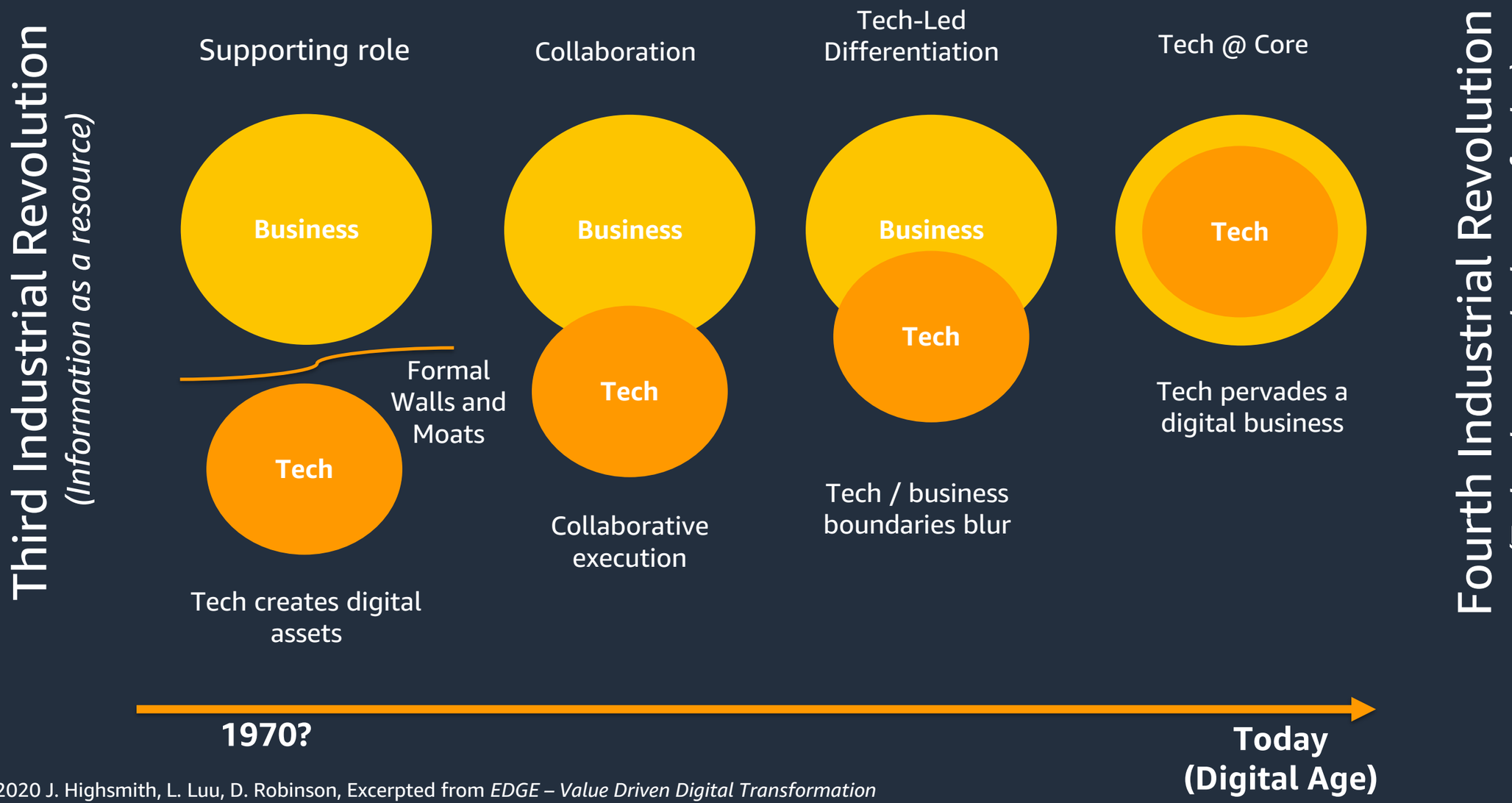
Automate, Enable, & Self-service

Modernization is the refactoring of legacy technology by combining modern infrastructure, architecture, organization patterns together to maximize resiliency, engineering efficiency, and business agility.

Value from Data, deserves it's own dedicated talk

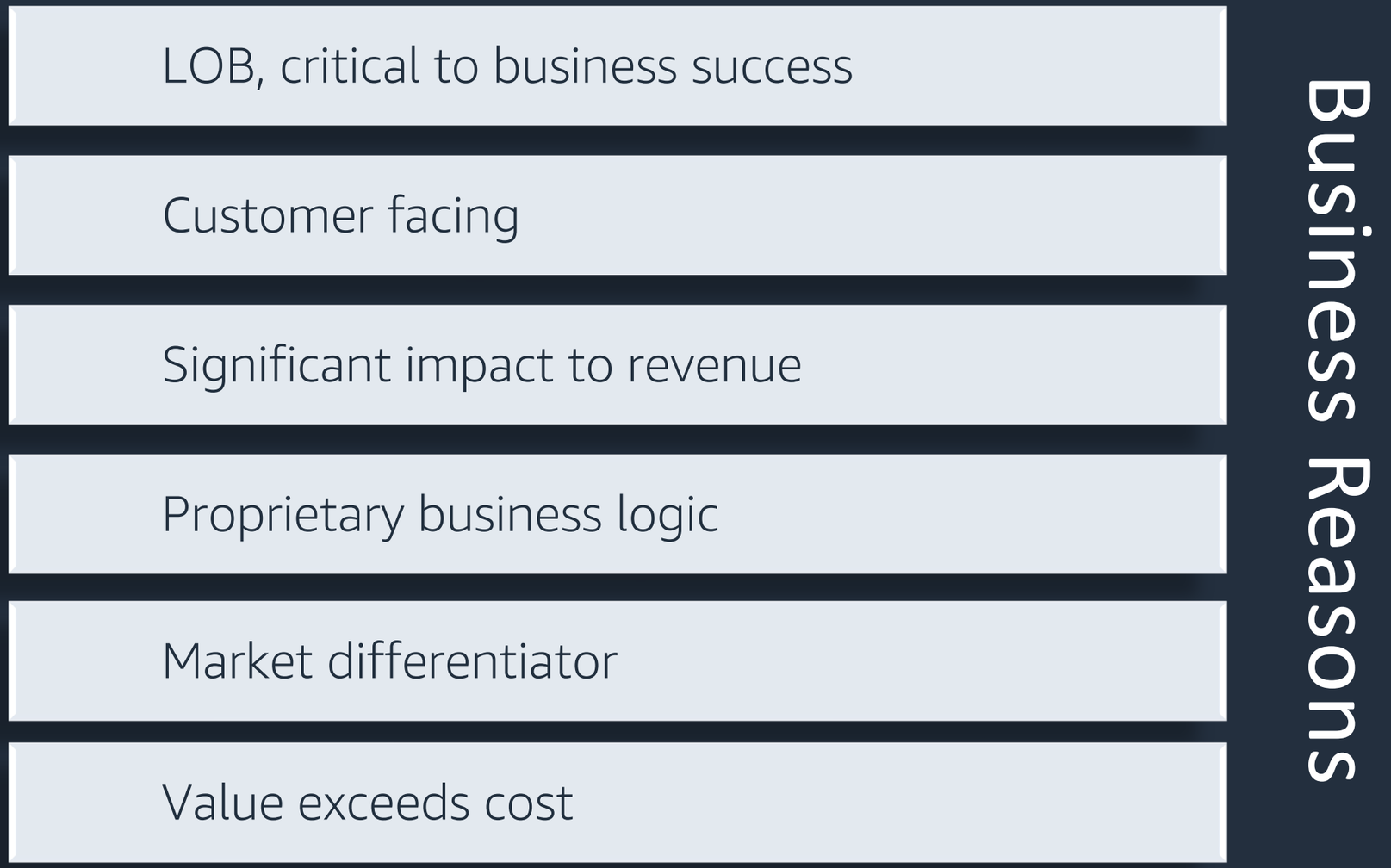


Tech is the Strategic Differentiator



Source: © 2020 J. Highsmith, L. Luu, D. Robinson, Excerpted from *EDGE – Value Driven Digital Transformation*

Workload Selection Criteria



Workload Selection Criteria

Old technology, no support

Performance & scalability issues

Difficult to extend capabilities

Lack of skillset, lost knowledge

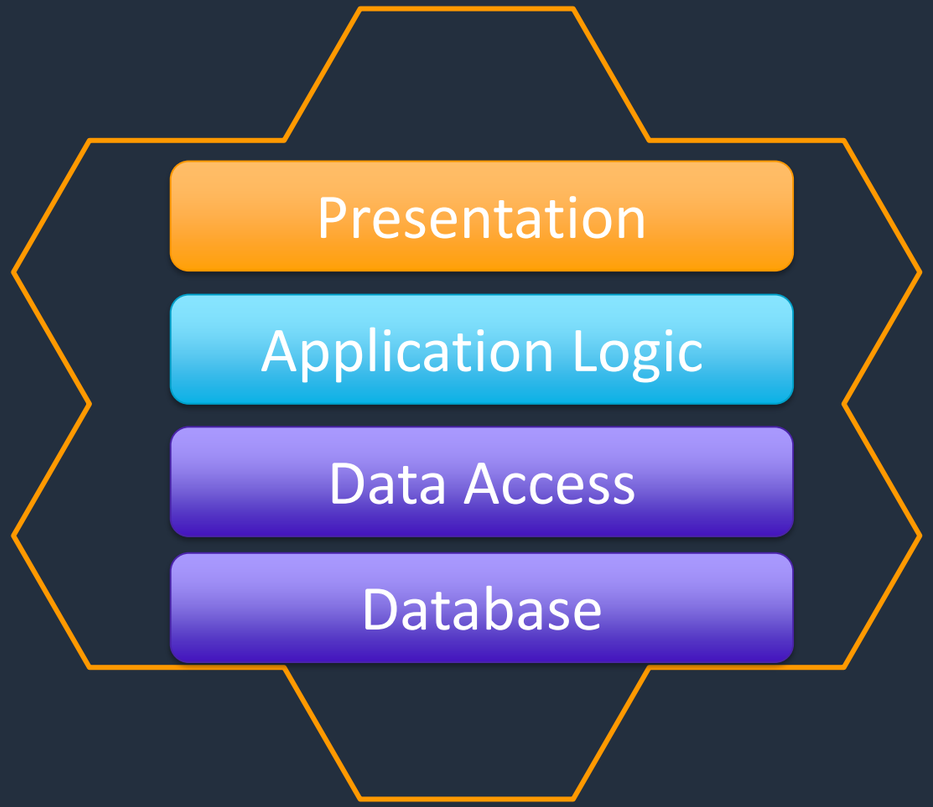
Too many bugs, spaghetti code

Expensive to sun, difficult to integrate

Technical Reasons

Monolithic Architectures

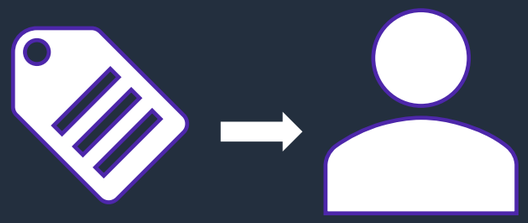
Large changes inherits large risk thereby slows releases and feedback



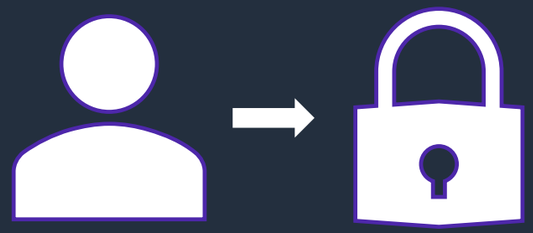
Monolithic application
Does everything
Shared release pipeline
Rigid scaling
High impact of change
Hard to adopt new technologies
Long deployment processes
Hard to iterate and extend

Repeatable tasks are executed manually

Compounding effects of resource drain on repetitive tasks



Service request with manual resolution



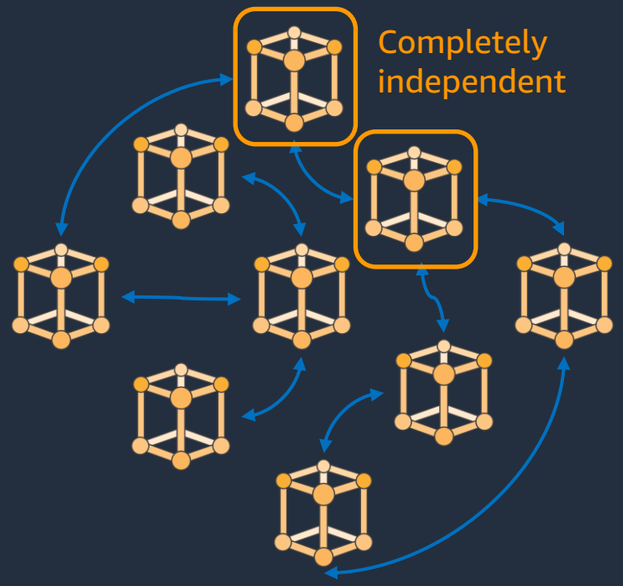
Security approval over security guardrails

Manual application testing and deployment

Microservices

Decoupled business capabilities with decentralized governance and data management

Microservices

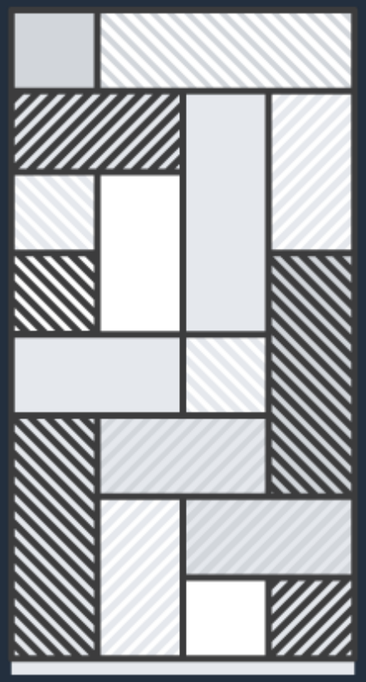


Re-usable modules of code
Built and deployed independently
Organized around **business capabilities**
Own their **domain logic**
Independent **SLA**
Optimizes around the cloud’s elasticity

Development transformation at Amazon.com

1
2

1994-2001



Monolithic architecture +
hierarchical organization

2002+

Decoupled services +
Two-pizza teams

We are witnessing a paradigm shift

75% *of organizations use or plan to use serverless technologies within the next two years.*[1]

What do we mean when Aeries say serverless?

- ✓ No server management
- ✓ Flexible, automated scaling
- ✓ 3X more native integrations (Lambda has 47 native integrations)
- ✓ Automated high availability

Serverless-First is the decision to opt for serverless technologies in your application as a first choice.

Serverless to accelerate modernization

COMPUTE


AWS Lambda


AWS Fargate

DATA STORES


Amazon S3


Amazon Aurora Serverless


Amazon DynamoDB

INTEGRATION


Amazon EventBridge


Amazon API Gateway


Amazon SQS


Amazon SNS

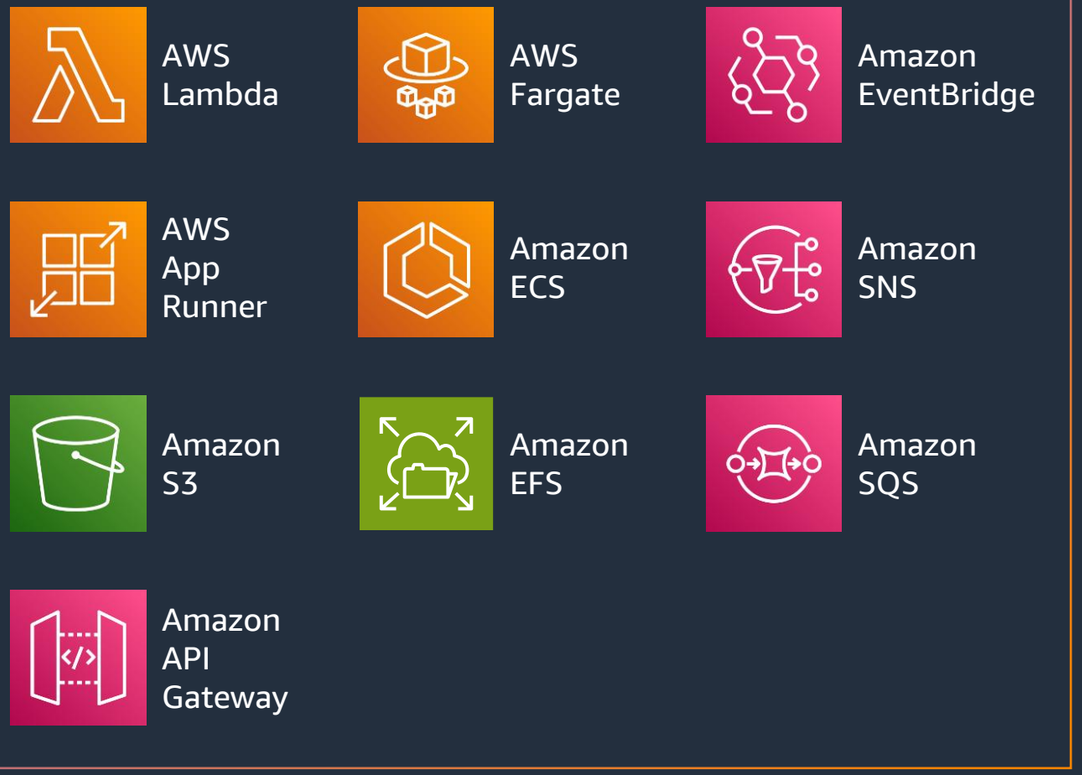

AWS Step Functions


AWS AppSync

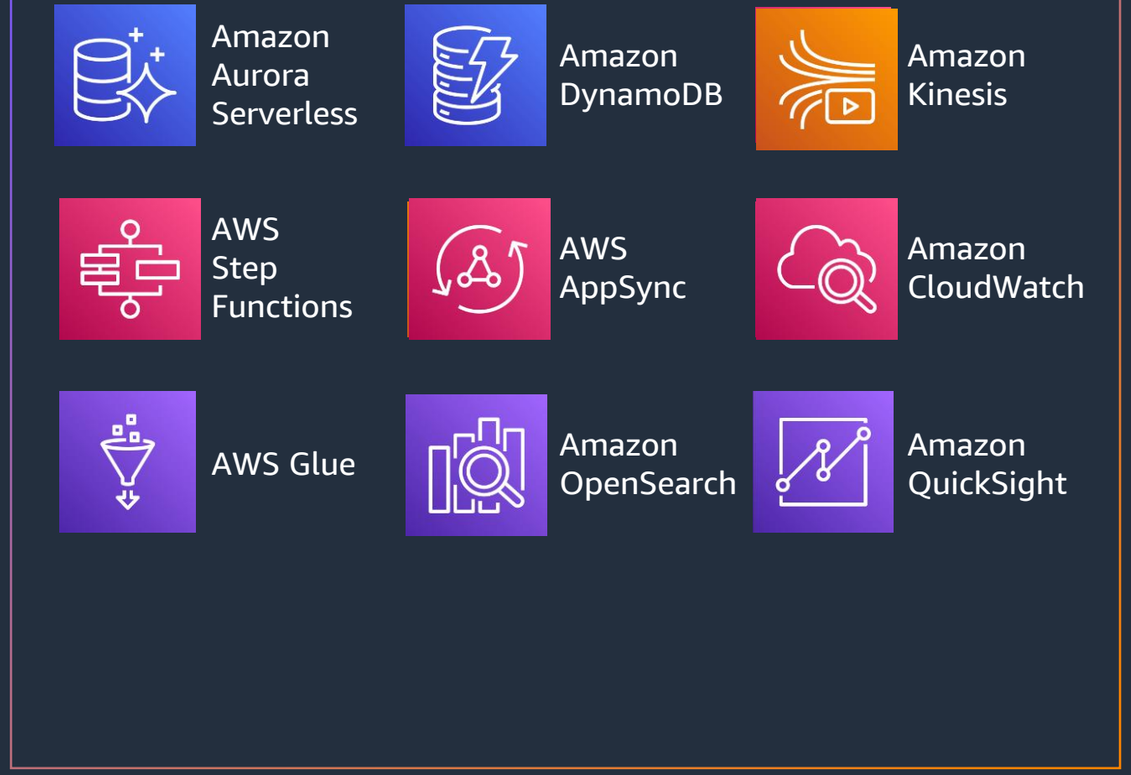
AWS serverless spectrum

AWS OFFERS THE WIDEST PORTFOLIO OF SERVERLESS SERVICES FOR RUNNING AND BUILDING MODERN APPS

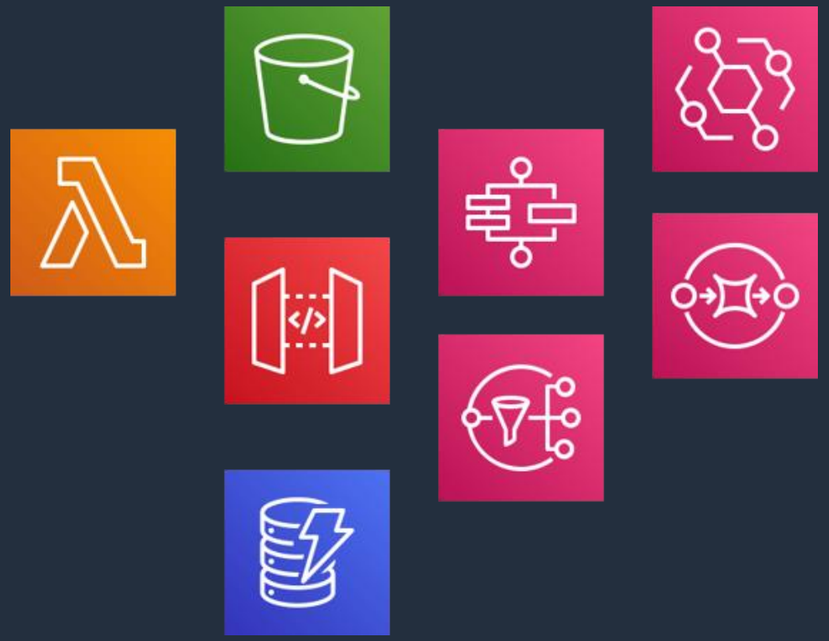
From Primitives to...
(Examples: compute, containers, buses)

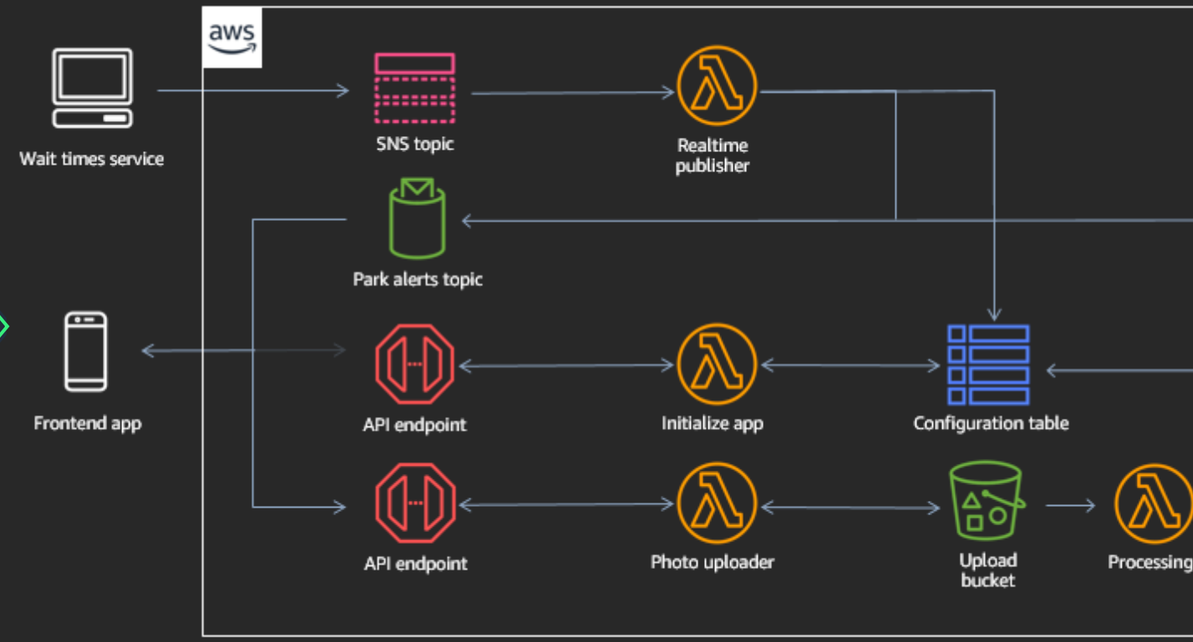


...Peripherals
(Examples: databases, analytics, workflows)

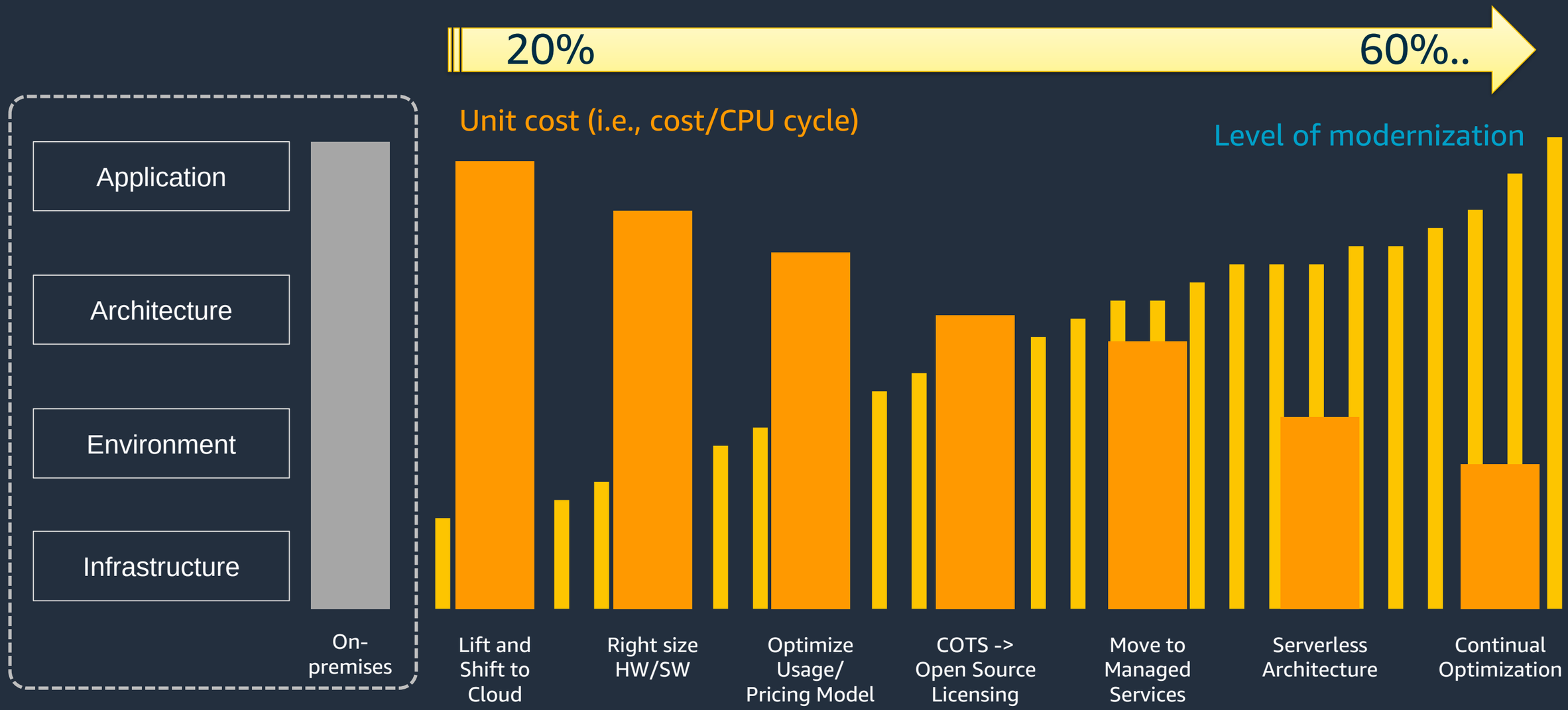


Small pieces, loosely joined



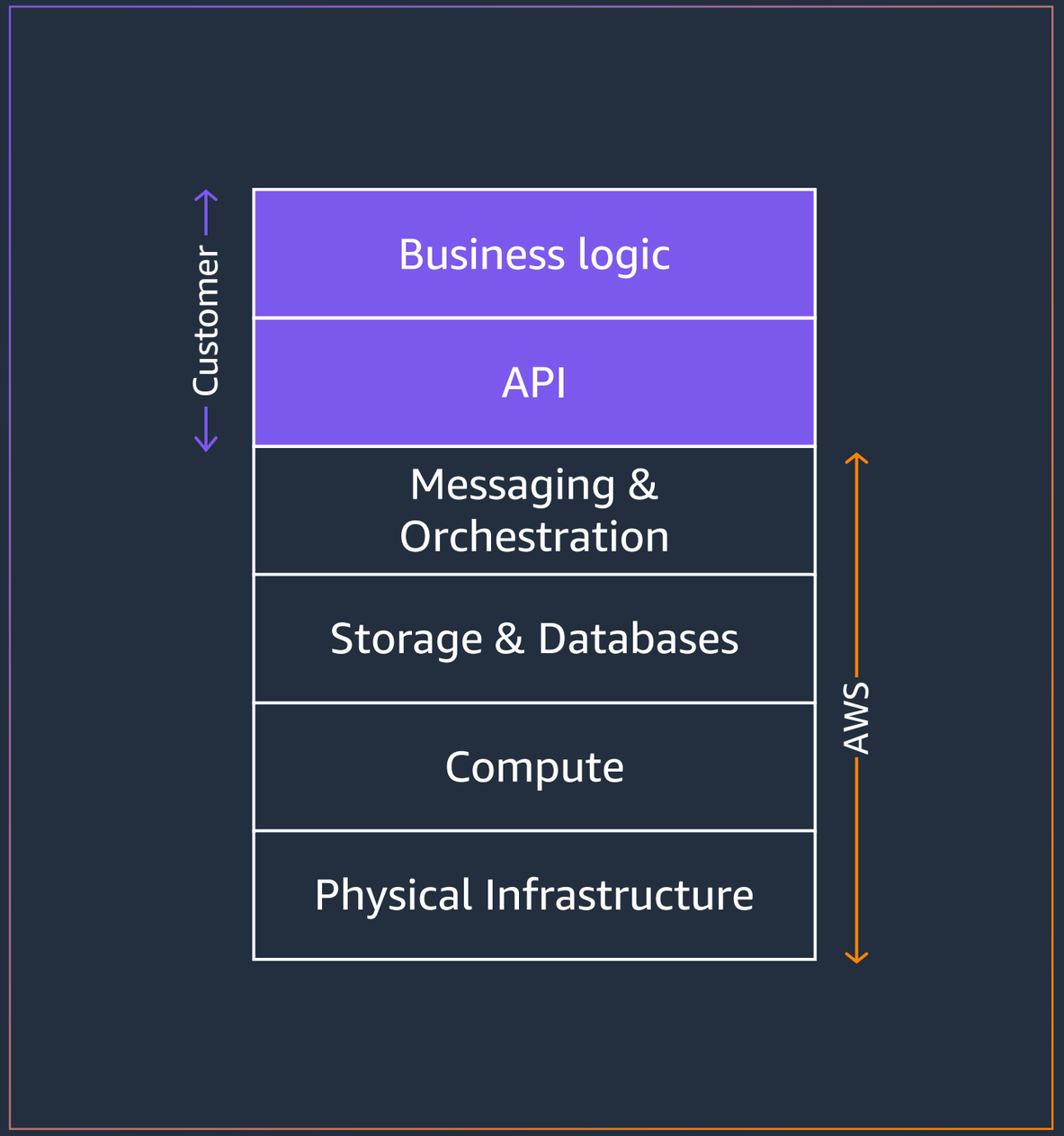


Modernization delivers economic value



What is Serverless?

Serverless services simplify the management and scaling of cloud applications by shifting undifferentiated operational tasks to the cloud provider so **development teams can focus on writing code** that solve business problems



Why serverless

TAKE FULL ADVANTAGE OF THE CLOUD TO MODERNIZE APPLICATIONS AND ACCELERATE INNOVATION

No infrastructure provisioning, no management

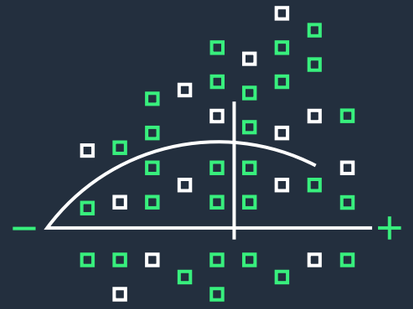
Automatic scaling

Pay for value

Highly available and secure

Possibilities with Serverless?

IT Automation



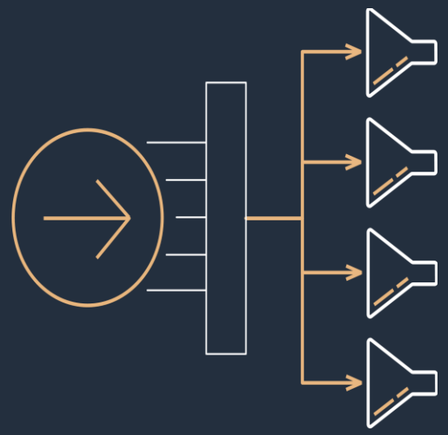
Data processing

Web applications

Machine Learning

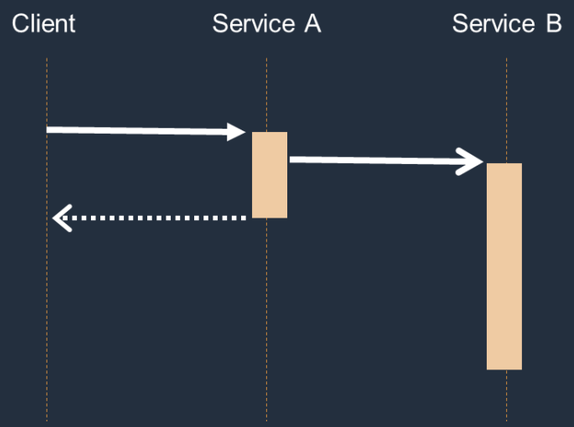
Event-driven architecture in Serverless

1



Event Routers

Abstract producers and consumers from each other



Asynchronous Events

Improve responsiveness and reduce dependencies

Event Stores

Buffer messages until services are available to process

Trends of event-driven architectures

WHY CUSTOMERS ARE MOVING TO EVENT-DRIVEN APPLICATIONS

1

Speed & agility

Move faster. Build and deploy services independently.

2

Resiliency

Loosely coupled systems can run and fail independently.

3

Scalability

Minimize waiting time through async and parallel processing.

AWS Lambda

Event-driven function-as-a-service

Serverless Architecture

Event Source

Changes in data state

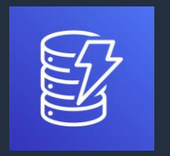
Requests to endpoints

Changes in resource state

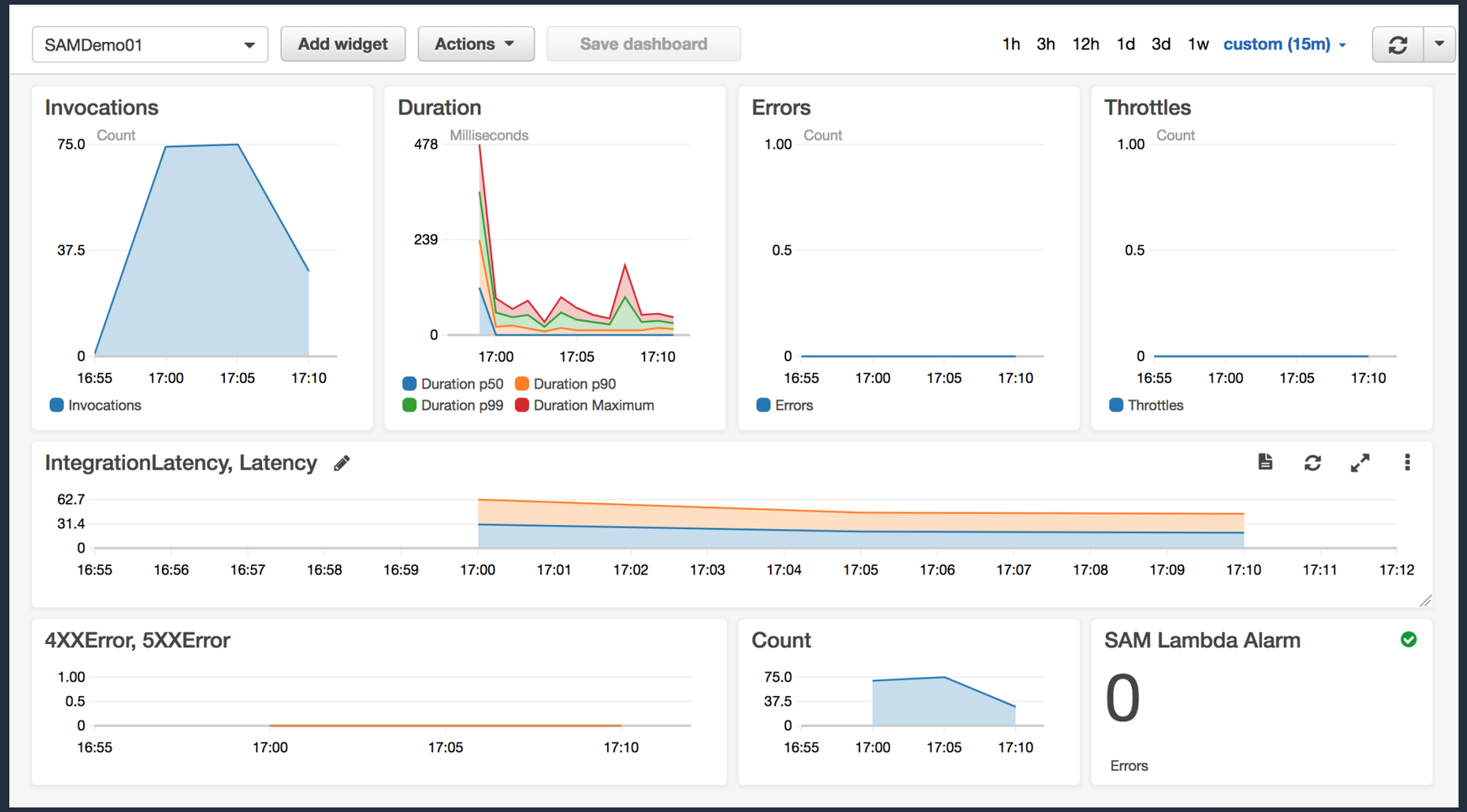
Function

Node.js
Python
Java
C#
Go
Ruby
Bring Your Own

Services / Other

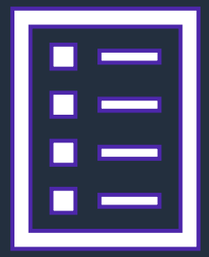


Built in monitoring

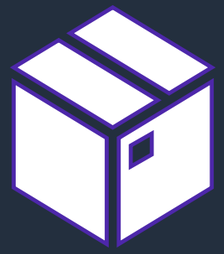


Productivity through automation, self-service, and guardrails

3



Baselined services catalog with automated provisioning



Implement security guardrails to enforce baseline and increase agility

CICD with gates that static code reviews, automate testing, and security scanning

Modernization Pathways

Move to Cloud Native Architecture

Agile, scalable apps built on containers, serverless and microservices

Move to Managed Cloud Services

Deploy applications rapidly and operate reliably at scale with managed services

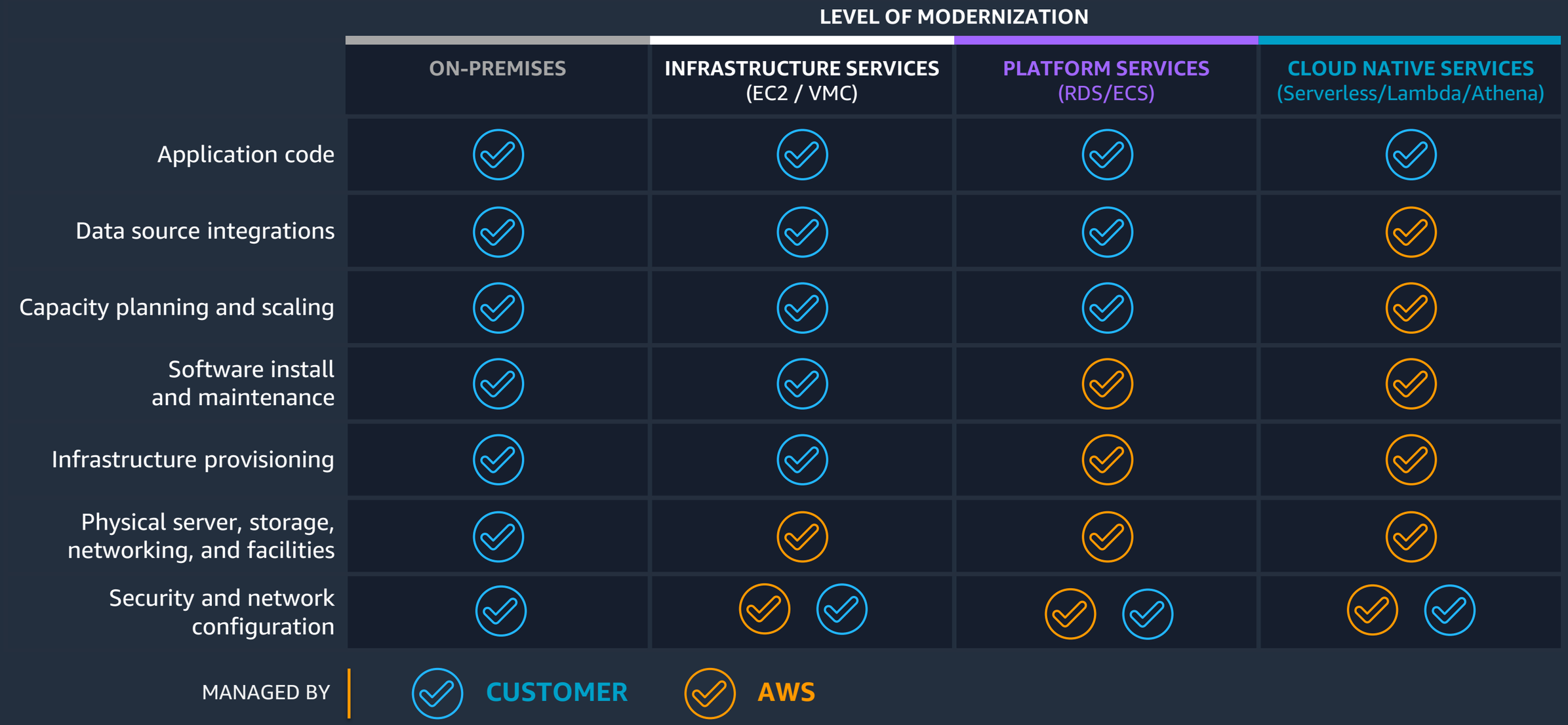
Move to Managed Databases

Open source, fit for purpose, highly scalable databases

Move to Open Source

Freedom from proprietary licensed software with open source technology

Efficiency moves up with Modernization



How to get Started ?
Thank You

aws.amazon.com/cloud-migration

1. Ensure executive sponsorship
2. Identify high-business value workloads
3. Move quickly on small projects and learn
4. Give serious consideration to bringing in a modernization advisory expert