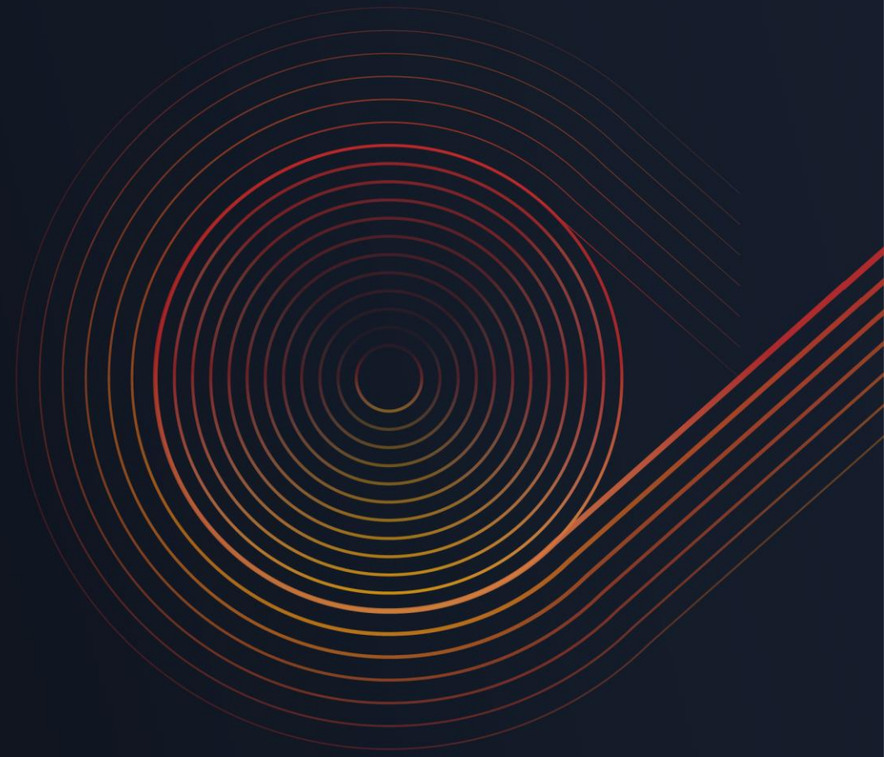
aws modern apps

How operations change as your organization embraces event-driven architectures

Danilo Poccia
Chief Evangelist, EMEA
Amazon Web Services

@danilop



Agenda

- Why event-driven architectures
- What’s different about modern operations
- Why observability matters
- What about teams
- Case study ☺

Why event-driven architectures?

aws modern apps

“A complex system that works is invariably found to have evolved from a simple system that worked.”

Gall’s Law

aws modern apps

“Amazon S3 is intentionally built with a minimal feature set. The focus is on simplicity and robustness.”

– Amazon S3 Press Release, March 14, 2006

Amazon Simple Storage Service (S3)

8 → more than 200 microservices

Mai-Lan Tomsen Bukovec
VP and GM, Amazon S3

aws modern apps

What is an “event” ?

“something that happens”

Events tell us a fact

Immutable time-series

Time	What
2020 09 21 08 07 06	CustomerCreated
2020 09 21 08 07 09	OrderCreated
2020 09 21 08 07 13	PaymentSuccessful
2020 09 21 08 07 17	CustomerUpdated
. . .	. . .

Source - “Event.” Merriam-Webster.com Dictionary, Merriam-Webster, https://www.merriam-webster.com/dictionary/event. Accessed 29 Sep. 2020.

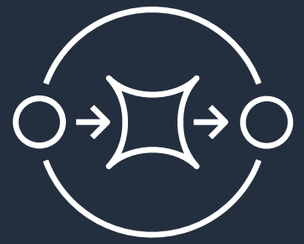
aws modern apps

Commands vs Events

Command	Event
Has an intent Directed to a target Personal communication	It’s a fact For others to observe Broadcast one to many
”CreateAccount” “AddProduct”	”AccountCreated” “ProductAdded”

Connecting AWS event sources

Messaging



Amazon Simple Queue Service (SQS)

Queues

Amazon Simple Notification Service (SNS)

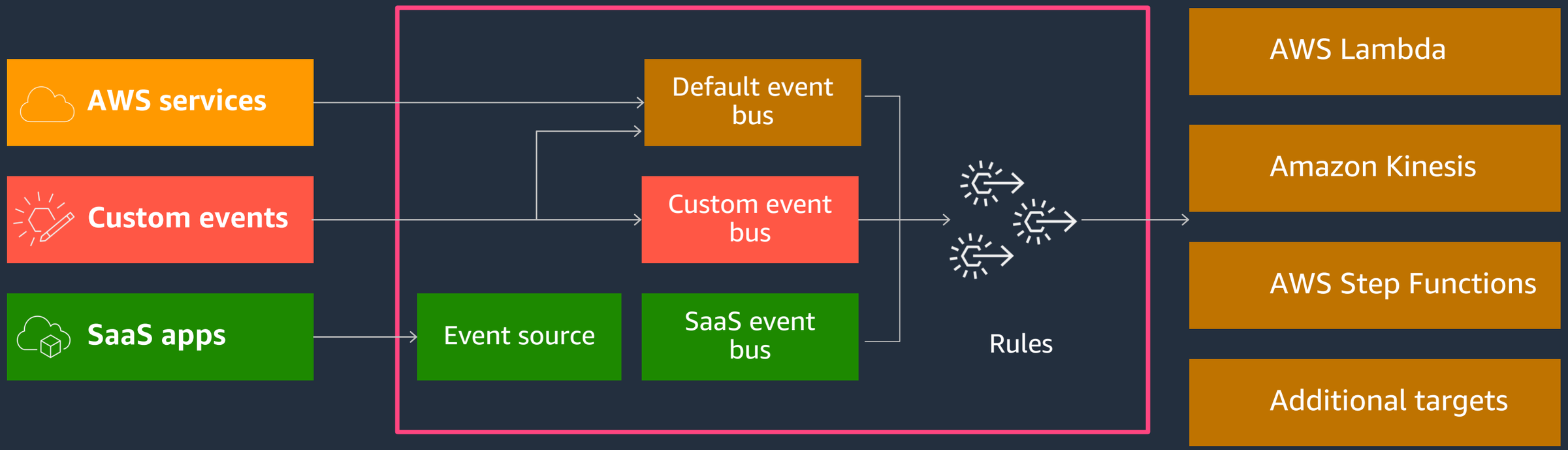
Pub/Sub

Amazon EventBridge

Event Bus

aws modern apps

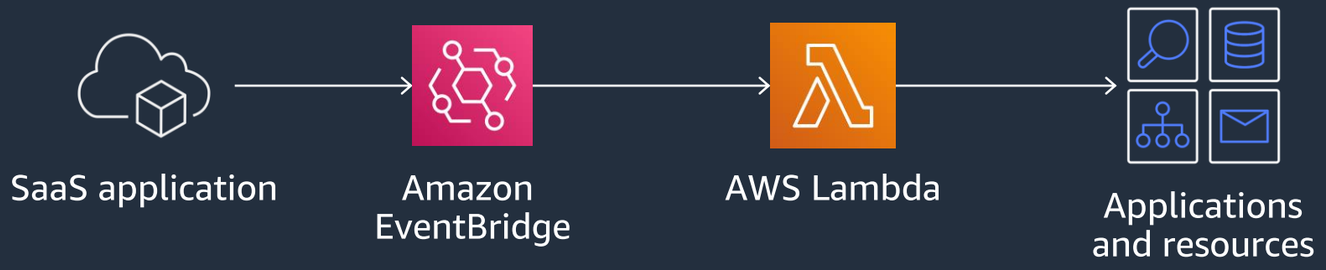
Amazon EventBridge



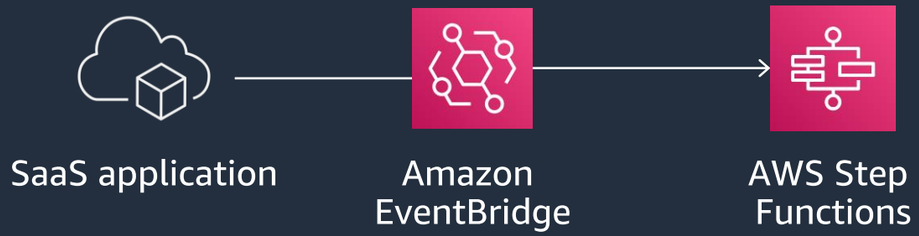
aws modern apps

Common use cases

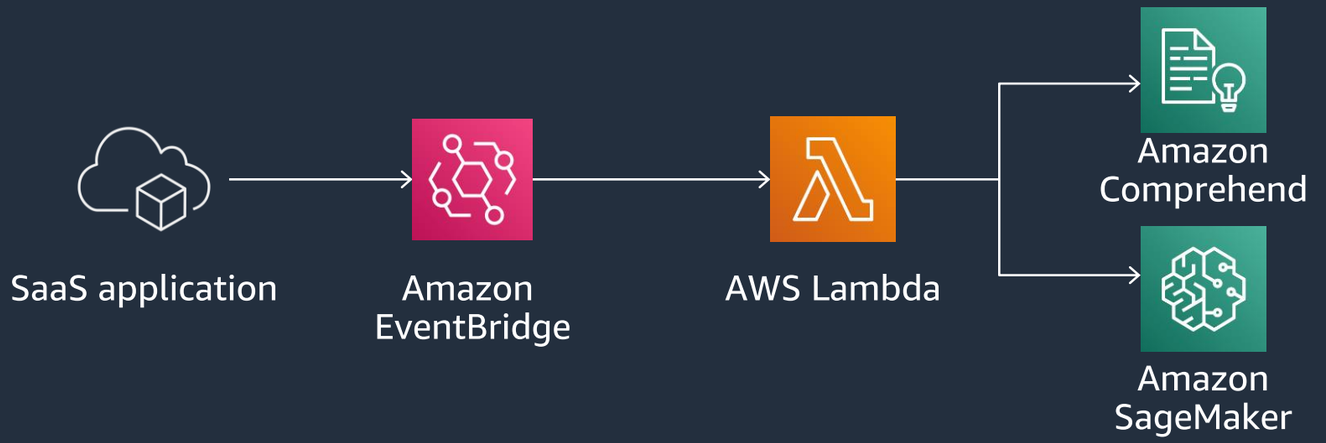
Take action



Run workflows



Apply intelligence



aws modern apps

Schema registry and discovery

Amazon EventBridge

Source of truth for sharing schema

Explicitly publish and discover

Integrations for JetBrains and VS Code

Language bindings for Java, Python, and TypeScript

aws modern apps

What’s different about modern operations?

aws modern apps

What’s different about modern operations?

Traditional	Modern
Central control	Decoupled teams
Periodic software release	Continuous delivery
Physical hardware	Virtual or ephemeral
Manual tasks	Automation via code

What are the approaches to operations?

Free for all

Fast dev time, but high risk to legal & app reliability

Chaos

Guardrails

Fast time & low risk to the business

Win win

Central control

Low risk but very slow to release

Dependencies & time lags

aws modern apps

What are guardrails?

Guardrails are mechanisms, such as processes or practices, that reduce both the occurrence and blast radius of undesirable application behavior

What are some real-world guardrails?

Security
& compliance

Cost
management

Deployment

Provisioning

Monitoring

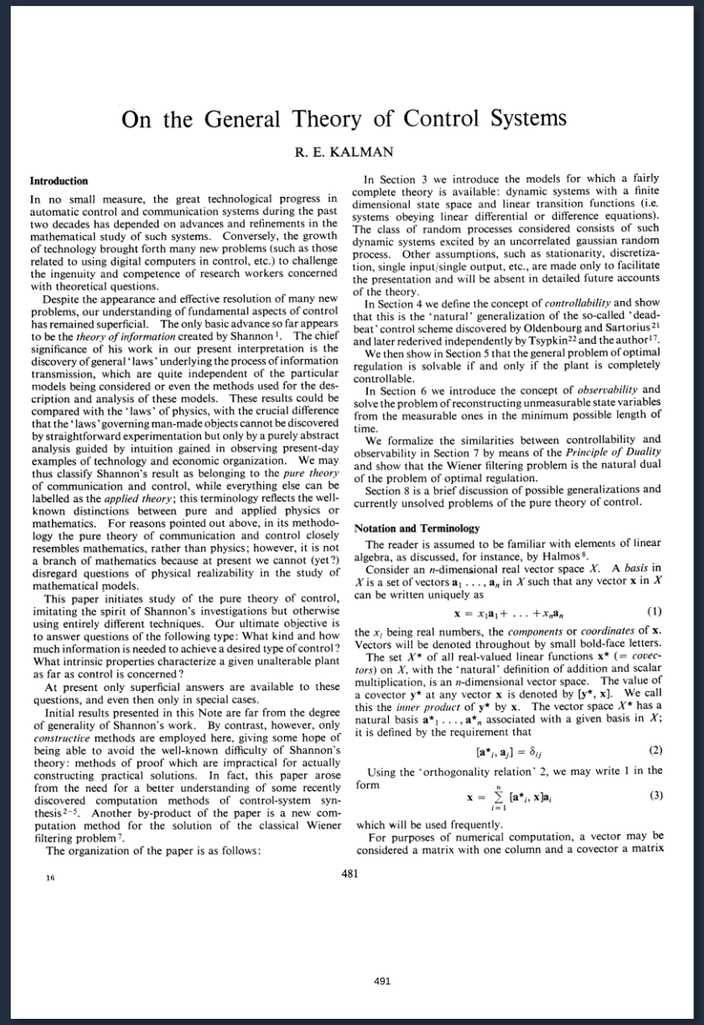
aws modern apps

Why observability matters?

aws modern apps

Observability in Control Theory

1961-62



On the General Theory of Control Systems

R. E. KALMAN

Introduction

In no small measure, the great technological progress in automatic control and communication systems during the past two decades has depended on advances and refinements in the mathematical study of such systems. Conversely, the growth of technology brought forth many new problems (such as those related to using digital computers in control, etc.) to challenge the ingenuity and competence of research workers concerned with theoretical questions.

Despite the appearance and effective resolution of many new problems, our understanding of fundamental aspects of control has remained superficial. The only basic advance so far appears to be the *theory of information* created by Shannon[1]. The chief significance of his work in our present interpretation is the discovery of general 'laws' underlying the process of information transmission, which are quite independent of the particular models being considered or even the methods used for the description and analysis of these models. These results could be compared with the 'laws' of physics, with the crucial difference that the 'laws' governing man-made objects cannot be discovered by straightforward experimentation but only by a purely abstract analysis guided by intuition gained in observing present-day examples of technology and economic organization. We may thus classify Shannon's result as belonging to the *pure theory* of communication and control, while everything else can be labelled as the *applied theory*; this terminology reflects the well-known distinctions between pure and applied physics or mathematics. For reasons pointed out above, in its methodology the pure theory of communication and control closely resembles mathematics, rather than physics; however, it is not a branch of mathematics because at present we cannot (yet?) disregard questions of physical realizability in the study of mathematical models.

This paper initiates study of the pure theory of control, imitating the spirit of Shannon's investigations but otherwise using entirely different techniques. Our ultimate objective is to answer questions of the following type: What kind and how much information is needed to achieve a desired type of control? What intrinsic properties characterize a given unalterable plant as far as control is concerned?

At present only superficial answers are available to these questions, and even then only in special cases.

Initial results presented in this Note are far from the degree of generality of Shannon's work. By contrast, however, only *constructive* methods are employed here, giving some hope of being able to avoid the well-known difficulty of Shannon's theory: methods of proof which are impractical for actually constructing practical solutions. In fact, this paper arose from the need for a better understanding of some recently discovered computation methods of control-system synthesis[2-5]. Another by-product of the paper is a new computation method for the solution of the classical Wiener filtering problem[7].

The organization of the paper is as follows:

In Section 3 we introduce the models for which a fairly complete theory is available: dynamic systems with a finite dimensional state space and linear transition functions (i.e. systems obeying linear differential or difference equations). The class of random processes considered consists of such dynamic systems excited by an uncorrelated gaussian random process. Other assumptions, such as stationarity, discretization, single input/single output, etc., are made only to facilitate the presentation and will be absent in detailed future accounts of the theory.

In Section 4 we define the concept of *controllability* and show that this is the 'natural' generalization of the so-called 'deadbeat' control scheme discovered by Oldenbourg and Sartorius[21] and later rederived independently by Tsypkin[22] and the author[17].

We then show in Section 5 that the general problem of optimal regulation is solvable if and only if the plant is completely controllable.

In Section 6 we introduce the concept of *observability* and solve the problem of reconstructing unmeasurable state variables from the measurable ones in the minimum possible length of time.

We formalize the similarities between controllability and observability in Section 7 by means of the *Principle of Duality* and show that the Wiener filtering problem is the natural dual of the problem of optimal regulation.

Section 8 is a brief discussion of possible generalizations and currently unsolved problems of the pure theory of control.

Notation and Terminology

The reader is assumed to be familiar with elements of linear algebra, as discussed, for instance, by Halmos[8].

Consider an n-dimensional real vector space X. A *basis* in X is a set of vectors $\mathbf{a}_1, \ldots, \mathbf{a}_n$ in X such that any vector $\mathbf{x}$ in X can be written uniquely as

$$\mathbf{x} = x_1\mathbf{a}_1 + \ldots + x_n\mathbf{a}_n \tag{1}$$

the x_i being real numbers, the *components* or *coordinates* of $\mathbf{x}$. Vectors will be denoted throughout by small bold-face letters.

The set X^* of all real-valued linear functions $\mathbf{x}^*$ (= *covectors*) on X, with the 'natural' definition of addition and scalar multiplication, is an n-dimensional vector space. The value of a covector $\mathbf{y}^*$ at any vector $\mathbf{x}$ is denoted by $[\mathbf{y}^*, \mathbf{x}]$. We call this the *inner product* of $\mathbf{y}^*$ by $\mathbf{x}$. The vector space X^* has a natural basis $\mathbf{a}^*_1, \ldots, \mathbf{a}^*_n$ associated with a given basis in X; it is defined by the requirement that

$$[\mathbf{a}^*_i, \mathbf{a}_j] = \delta_{ij} \tag{2}$$

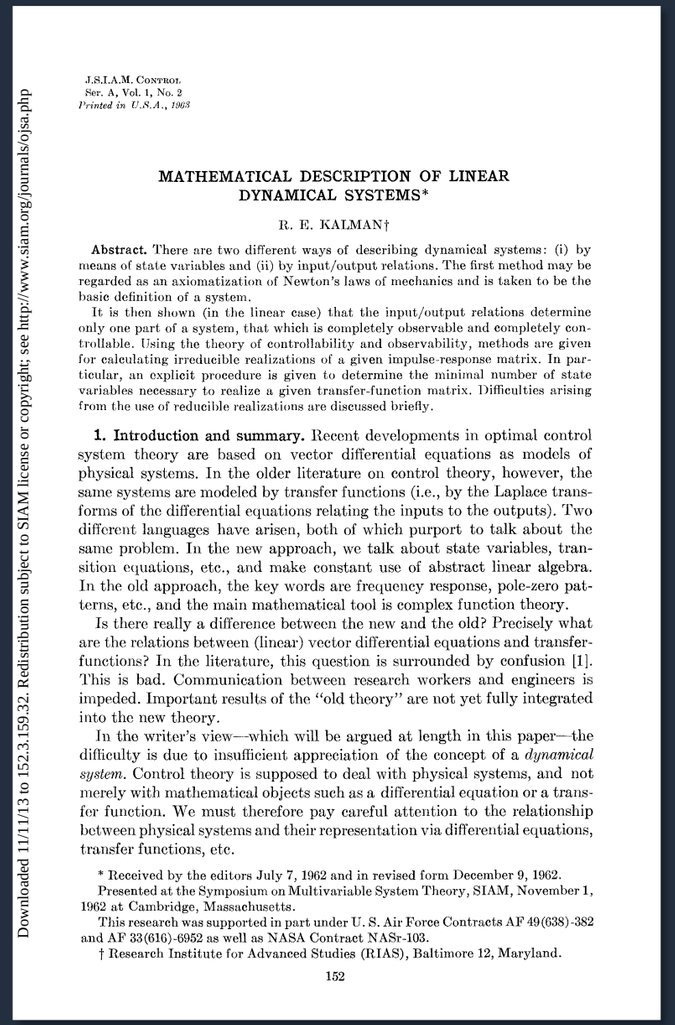
Using the 'orthogonality relation' 2, we may write 1 in the form

$$\mathbf{x} = \sum_{i=1}^{n} [\mathbf{a}^*_i, \mathbf{x}]\mathbf{a}_i \tag{3}$$

which will be used frequently.

For purposes of numerical computation, a vector may be considered a matrix with one column and a covector a matrix

16 481 491



J.S.I.A.M. CONTROL
Ser. A, Vol. 1, No. 2
Printed in U.S.A., 1963

MATHEMATICAL DESCRIPTION OF LINEAR DYNAMICAL SYSTEMS*

R. E. KALMAN†

Abstract. There are two different ways of describing dynamical systems: (i) by means of state variables and (ii) by input/output relations. The first method may be regarded as an axiomatization of Newton's laws of mechanics and is taken to be the basic definition of a system.

It is then shown (in the linear case) that the input/output relations determine only one part of a system, that which is completely observable and completely controllable. Using the theory of controllability and observability, methods are given for calculating irreducible realizations of a given impulse-response matrix. In particular, an explicit procedure is given to determine the minimal number of state variables necessary to realize a given transfer-function matrix. Difficulties arising from the use of reducible realizations are discussed briefly.

1. Introduction and summary. Recent developments in optimal control system theory are based on vector differential equations as models of physical systems. In the older literature on control theory, however, the same systems are modeled by transfer functions (i.e., by the Laplace transforms of the differential equations relating the inputs to the outputs). Two different languages have arisen, both of which purport to talk about the same problem. In the new approach, we talk about state variables, transition equations, etc., and make constant use of abstract linear algebra. In the old approach, the key words are frequency response, pole-zero patterns, etc., and the main mathematical tool is complex function theory.

Is there really a difference between the new and the old? Precisely what are the relations between (linear) vector differential equations and transfer-functions? In the literature, this question is surrounded by confusion [1]. This is bad. Communication between research workers and engineers is impeded. Important results of the "old theory" are not yet fully integrated into the new theory.

In the writer's view—which will be argued at length in this paper—the difficulty is due to insufficient appreciation of the concept of a *dynamical system*. Control theory is supposed to deal with physical systems, and not merely with mathematical objects such as a differential equation or a transfer function. We must therefore pay careful attention to the relationship between physical systems and their representation via differential equations, transfer functions, etc.

* Received by the editors July 7, 1962 and in revised form December 9, 1962.
Presented at the Symposium on Multivariable System Theory, SIAM, November 1, 1962 at Cambridge, Massachusetts.
This research was supported in part under U. S. Air Force Contracts AF 49(638)-382 and AF 33(616)-6952 as well as NASA Contract NASr-103.
† Research Institute for Advanced Studies (RIAS), Baltimore 12, Maryland.

152

Kalman R. E., "On the General Theory of Control Systems", Proc. 1st Int. Cong. of IFAC, Moscow 1960 1481, Butterworth, London 1961.
Kalman R. E., "Mathematical Description of Linear Dynamical Systems", SIAM J. Contr. 1963 1 152

aws modern apps

Observability

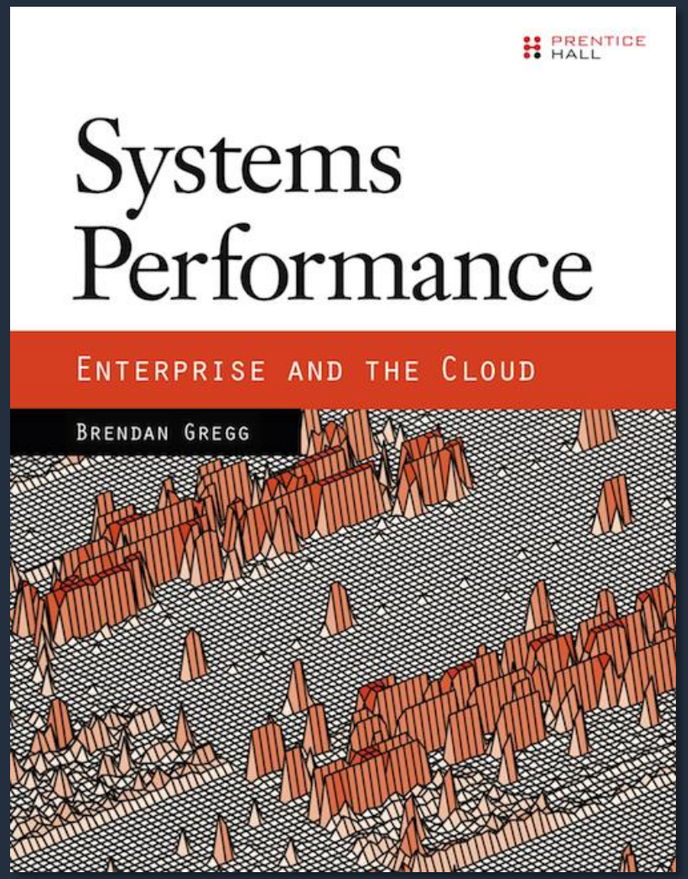
In control theory, observability is a measure of how well internal states of a system can be inferred from knowledge of its external outputs.

https://en.wikipedia.org/wiki/Observability

Don’t build a network of connected “black boxes”

Observability is a developer responsibility

Understand performance…



Systems Performance *by Brendan Gregg*

aws modern apps

Understand performance… and latency…

Event	Latency	Scaled
1 CPU cycle	0.3 ns	1 s
Level 1 cache access	0.9 ns	3 s
Level 2 cache access	2.8 ns	9 s
Level 3 cache access	12.9 ns	43 s
Main memory access (DRAM, from CPU)	120 ns	6 min
Solid-state disk I/O (flash memory)	50–150 µs	2–6 days
Rotational disk I/O	1–10 ms	1–12 months
Internet: San Francisco to New York	40 ms	4 years
Internet: San Francisco to United Kingdom	81 ms	8 years
Internet: San Francisco to Australia	183 ms	19 years
TCP packet retransmit	1–3 s	105–317 years
OS virtualization system reboot	4 s	423 years
SCSI command time-out	30 s	3 millennia
Hardware (HW) virtualization system reboot	40 s	4 millennia
Physical system reboot	5 m	32 millennia

Systems Performance *by Brendan Gregg*

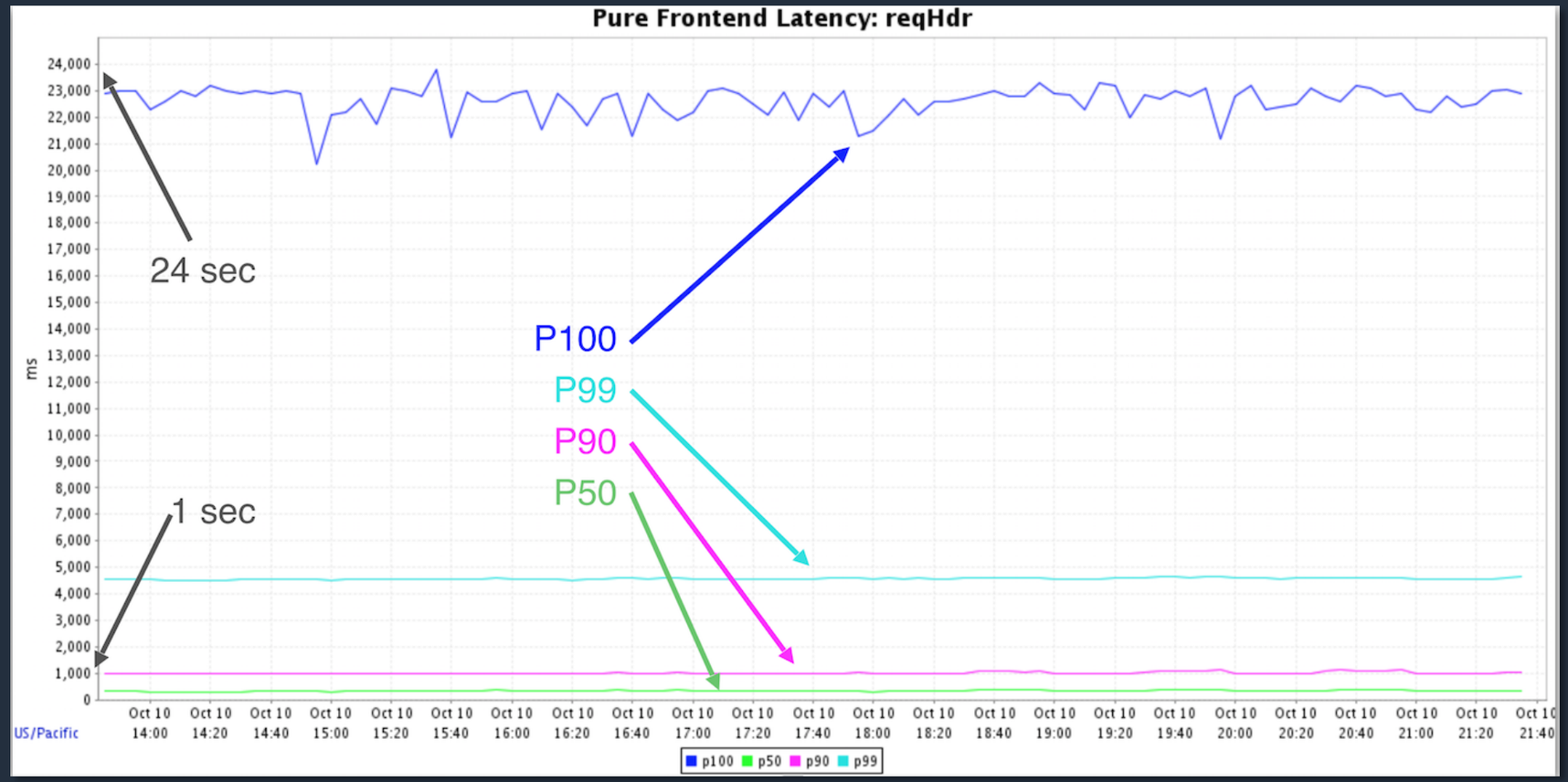
Understand performance… and latency… and percentiles!

P100

P99

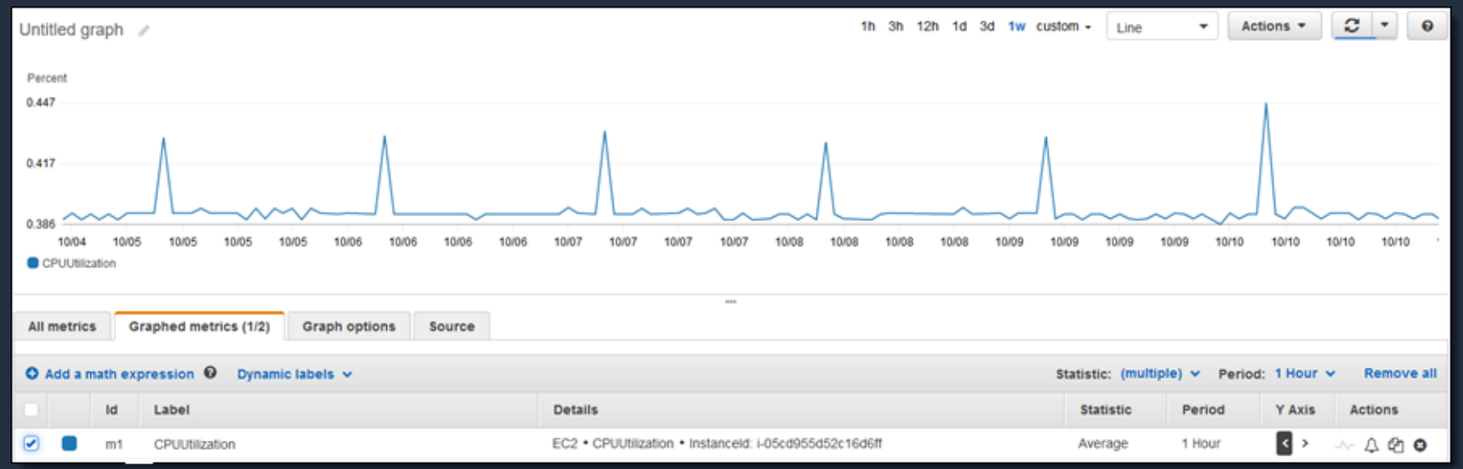
P90

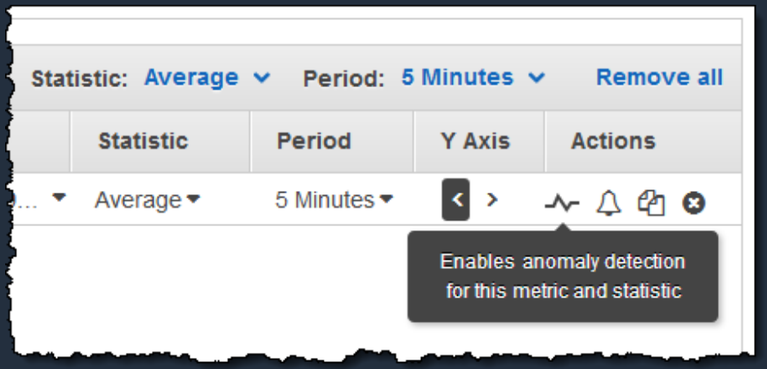
P50

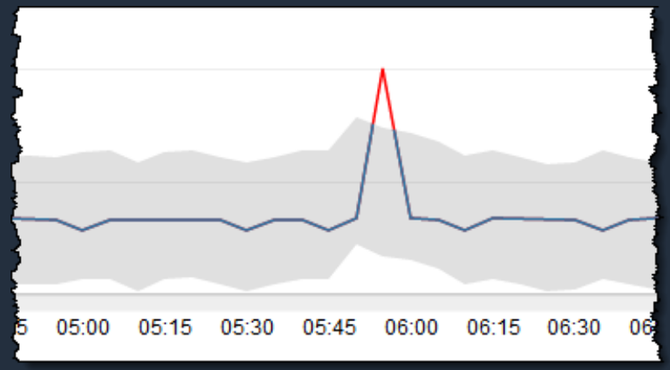


aws modern apps

Using Amazon CloudWatch Anomaly Detection alarms

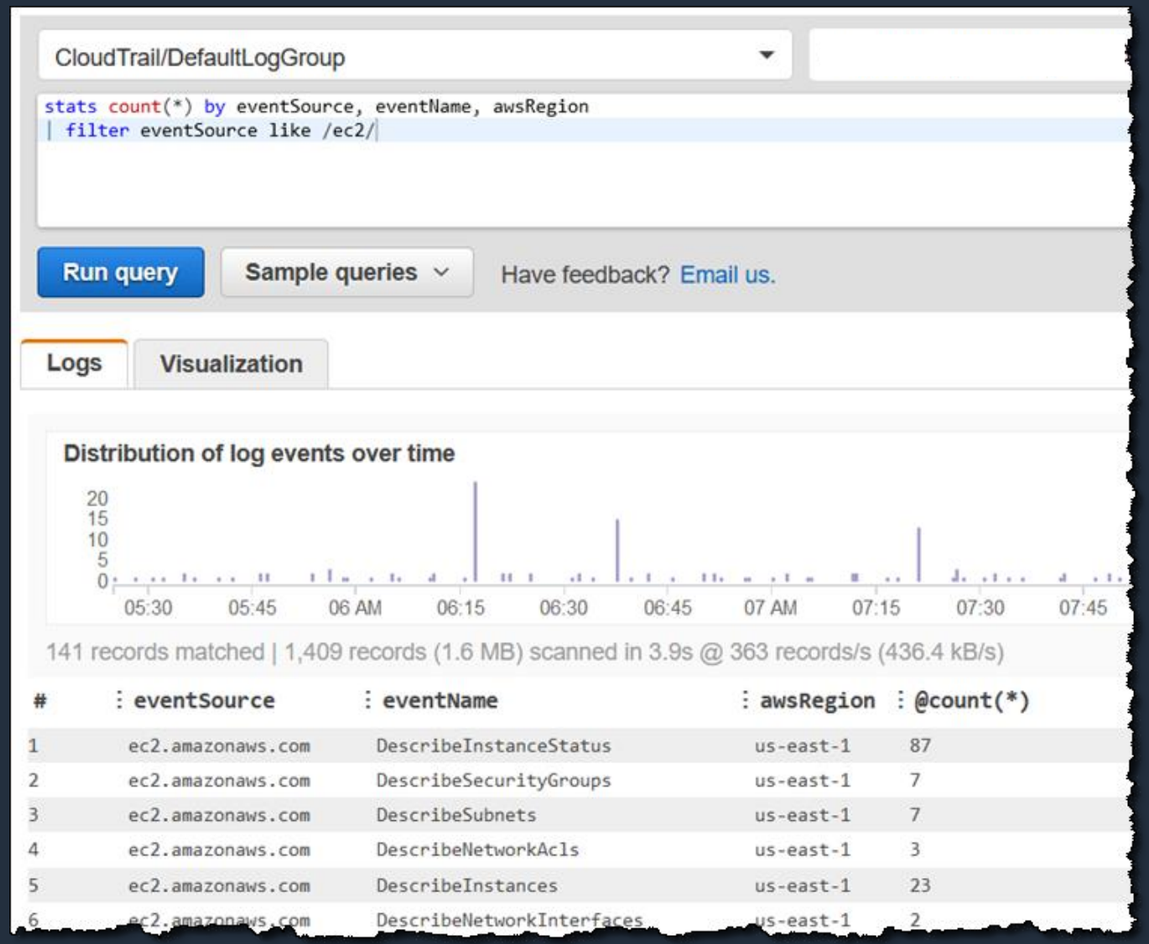


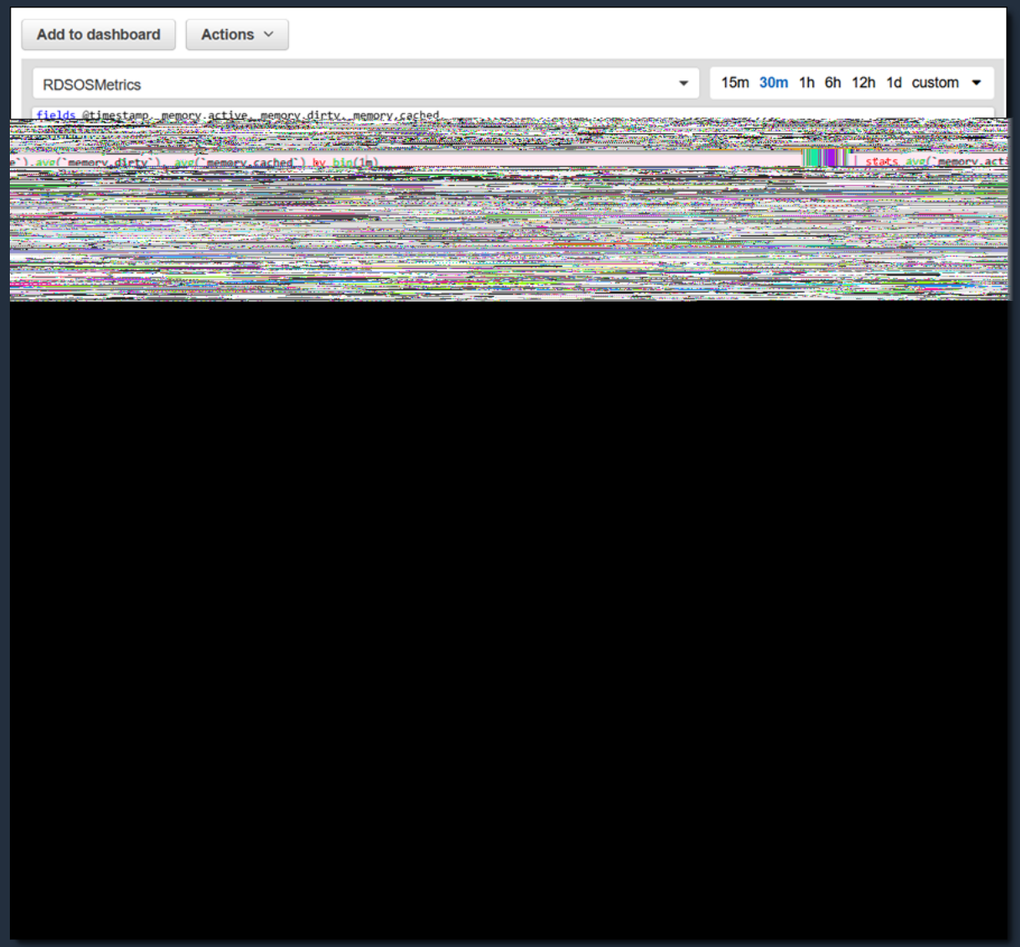




aws modern apps

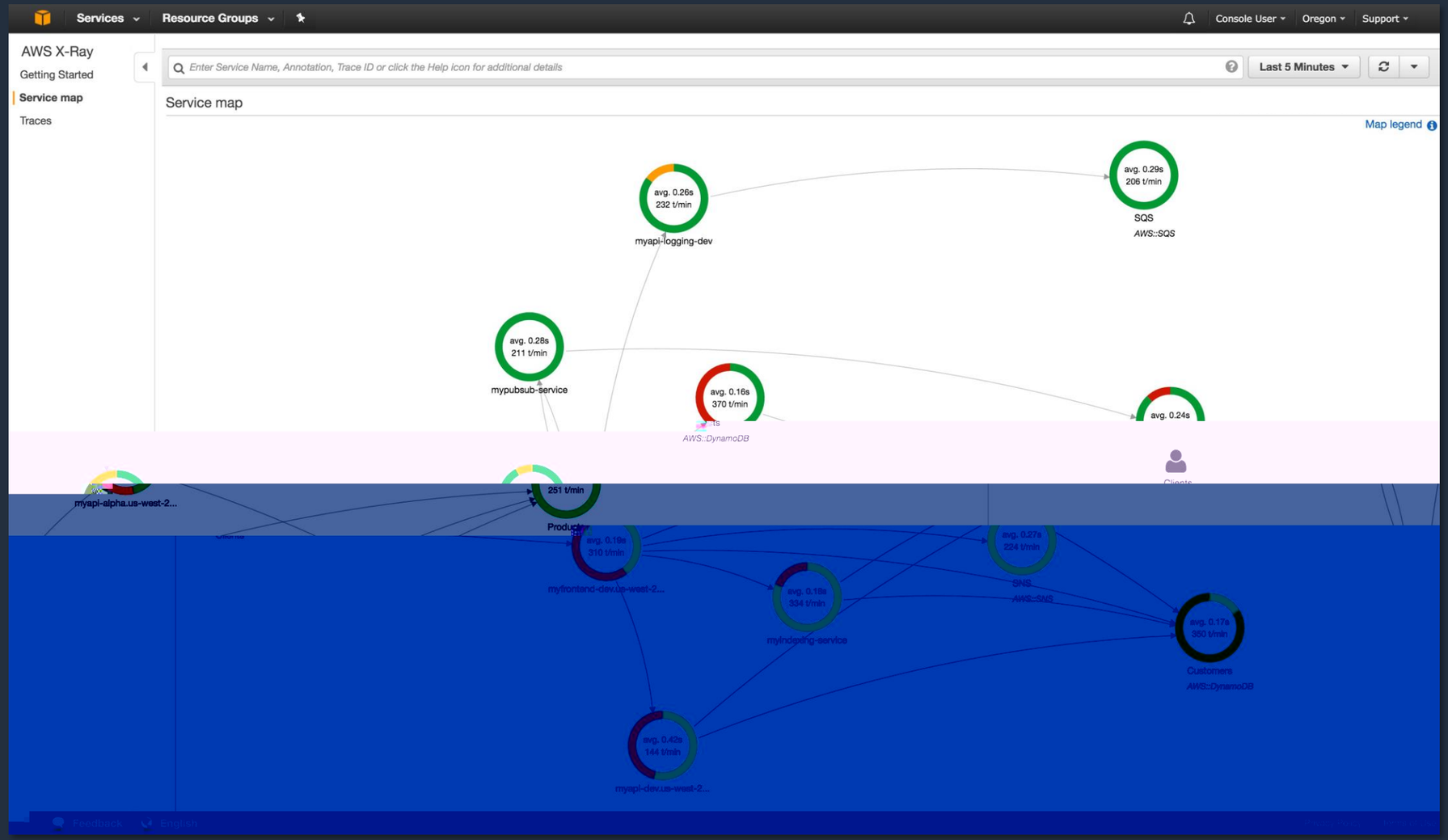
Understanding Data – Amazon CloudWatch Logs Insights





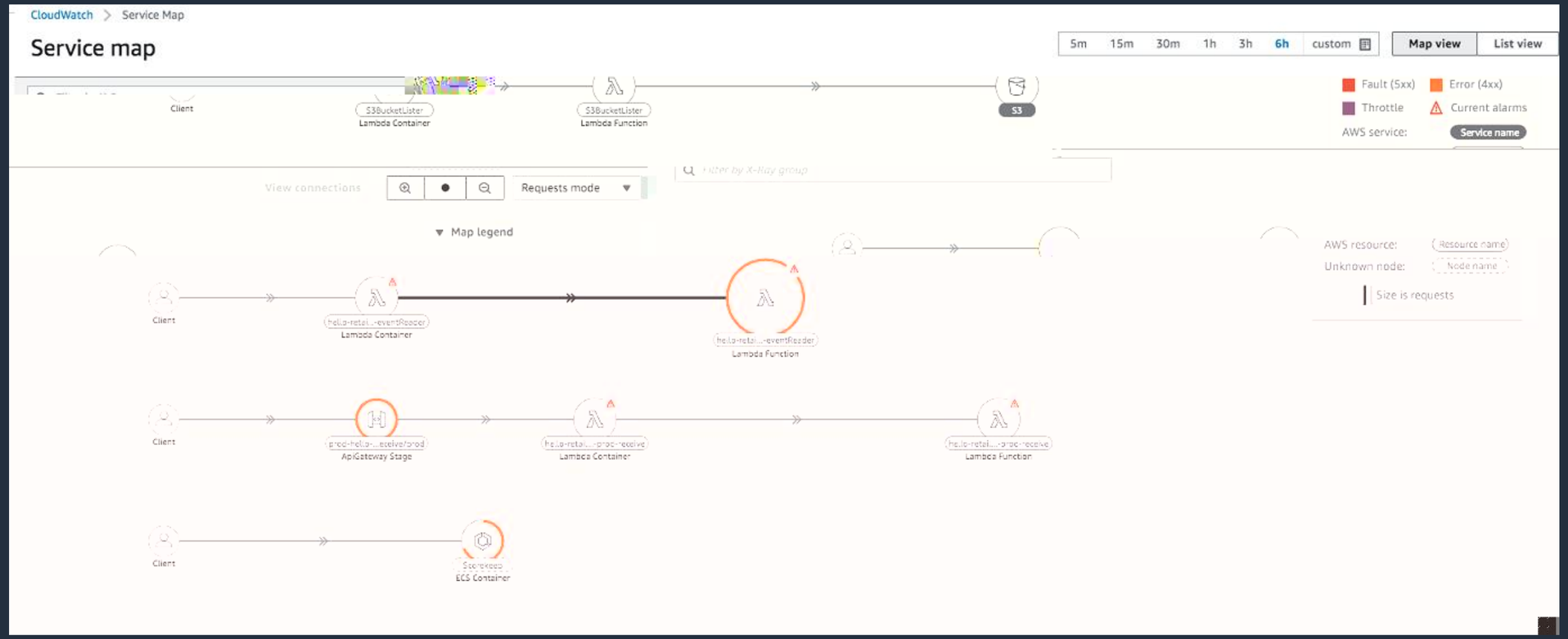
aws modern apps

End-to-end tracing – AWS X-Ray service map



aws modern apps

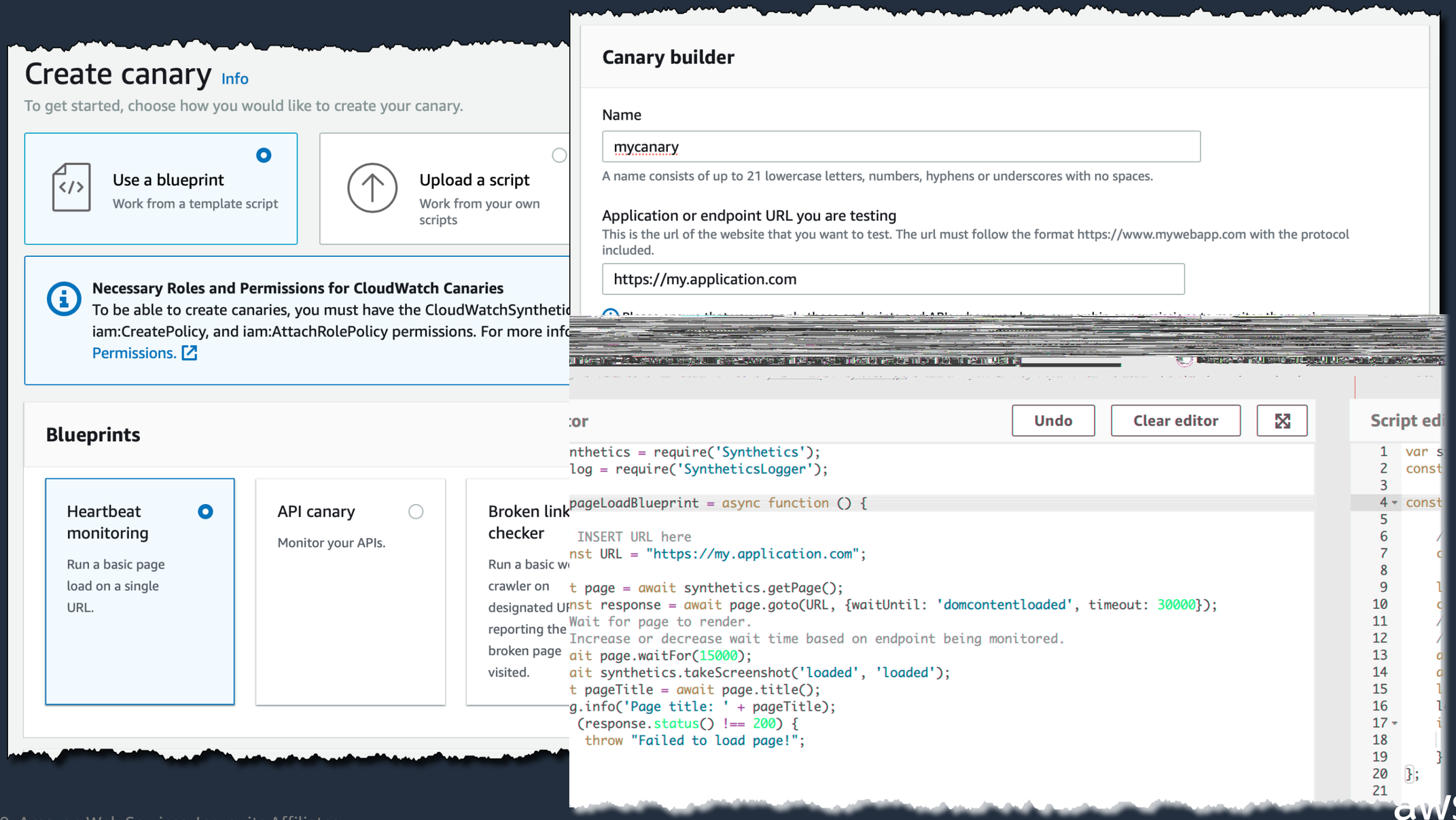
CloudWatch ServiceLens



CloudWatch metrics and logs + AWS X-Ray traces

aws modern apps

Amazon CloudWatch Synthetics – Monitor endpoints



aws modern apps

What about teams?

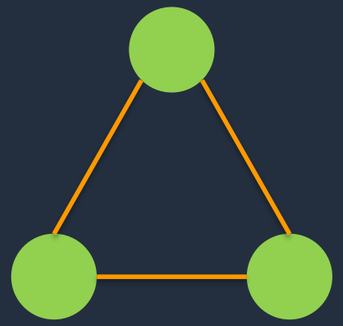
aws modern apps

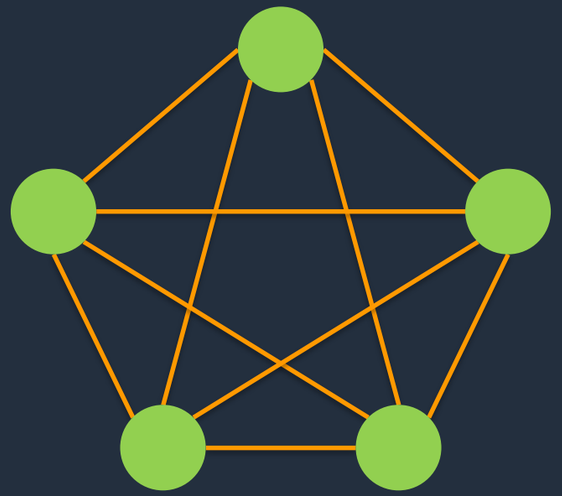
You Build It, You Run It

“This brings developers into contact with the day-to-day operation of their software. It also brings them into day-to-day contact with the customer.”

– Werner Vogels
CTO, Amazon.com

Team size & communication paths

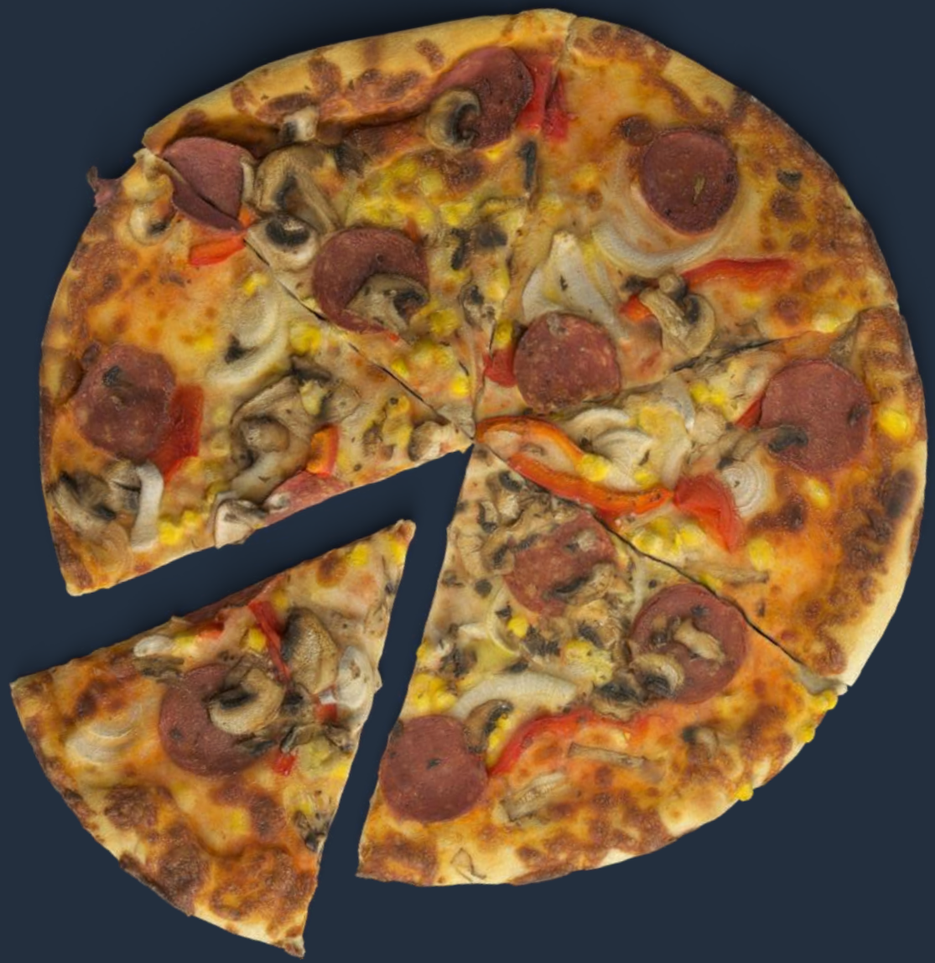




$$\text{Communication paths in a team of } N \text{ people} = \frac{N(N-1)}{2}$$

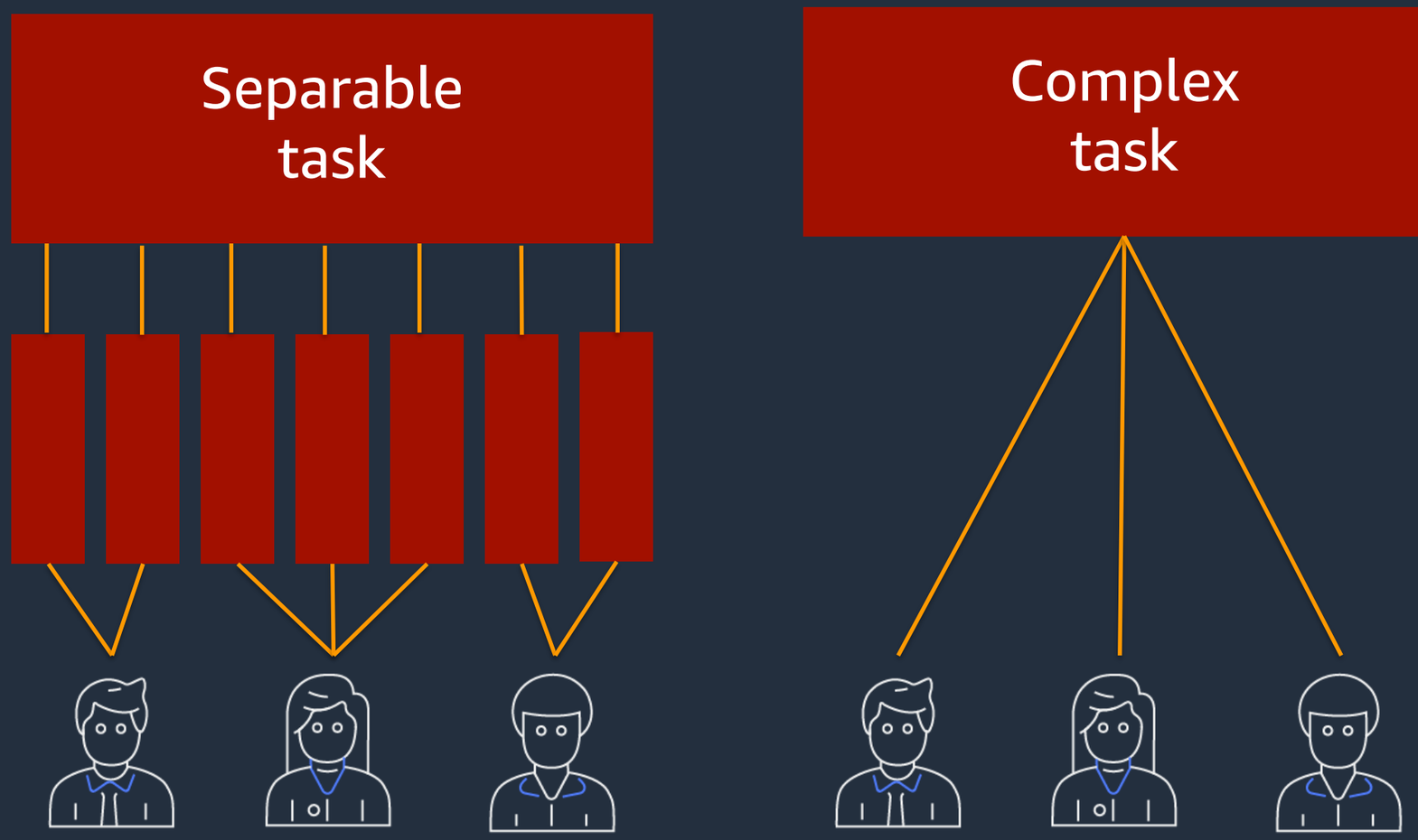
Photo by Kristina Bratko on Unsplash

Two pizza teams



aws modern apps

Separable vs complex tasks



aws modern apps

Ability as a collection of cognitive tools

Adam

Abilities = 5

For example:

A – mobile development on iOS

B – back end development in Java

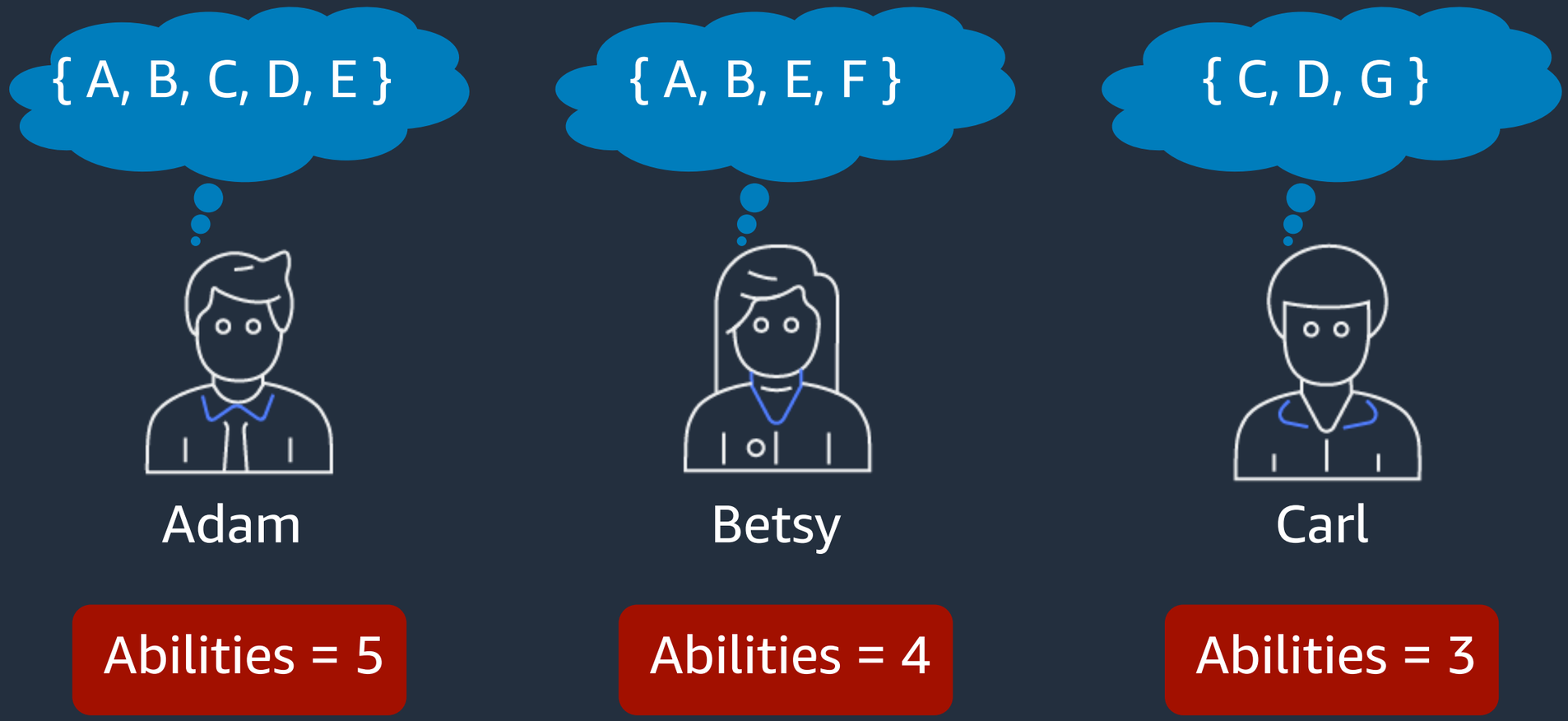
C – data analytics in Python

D – complex SQL queries

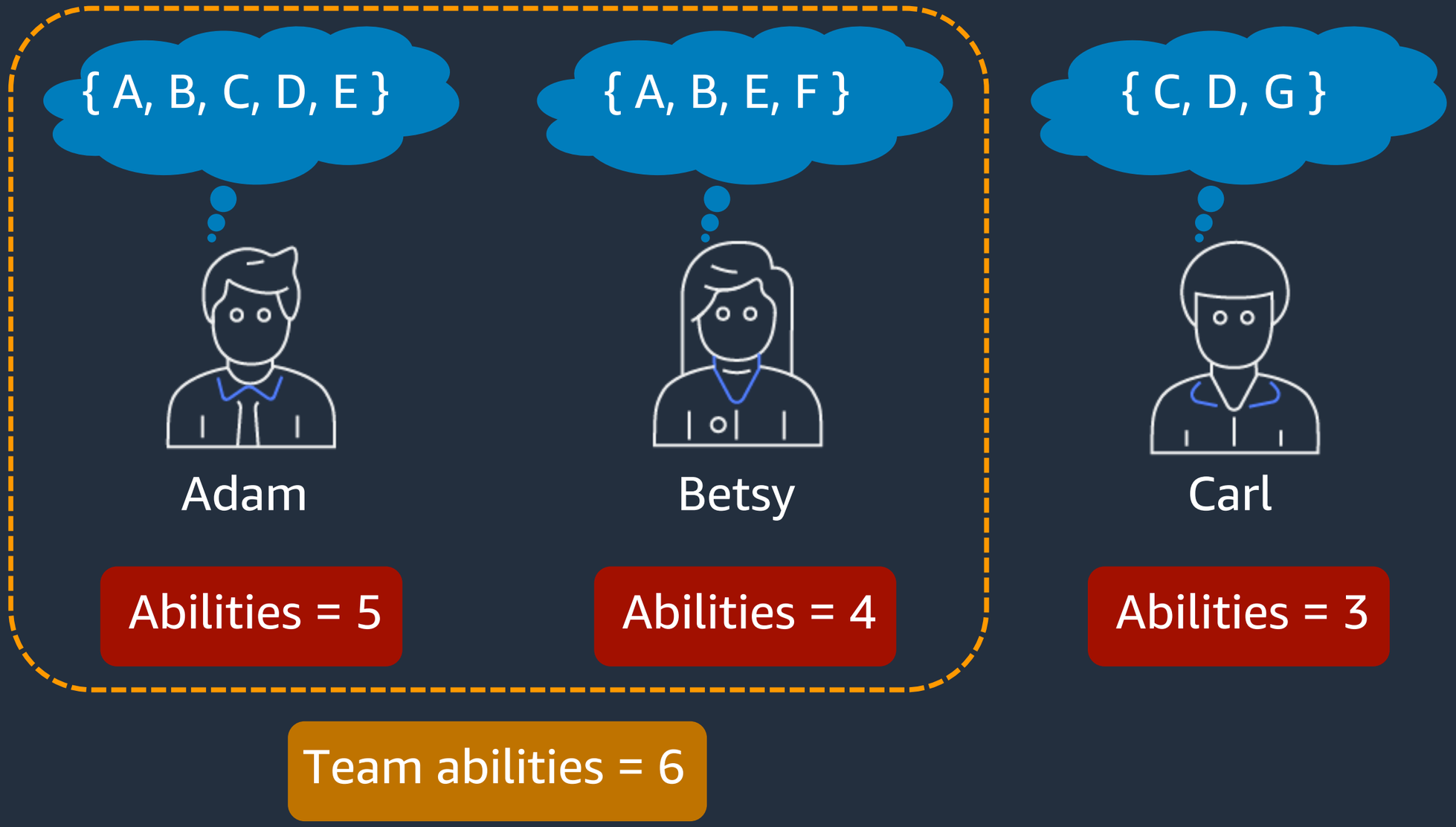
E – deep learning

aws modern apps

Ability as a collection of cognitive tools

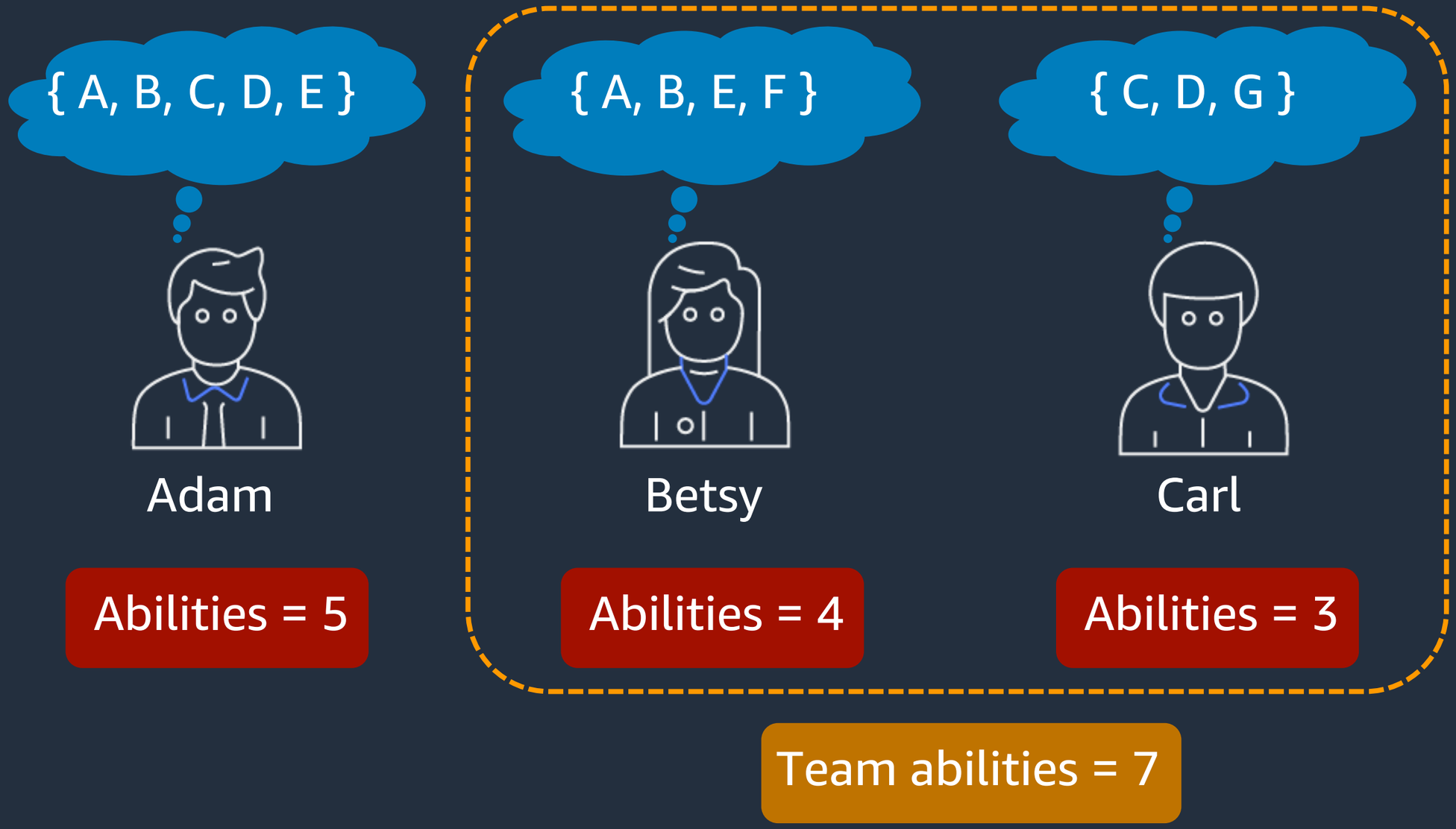


Diversity bonus model



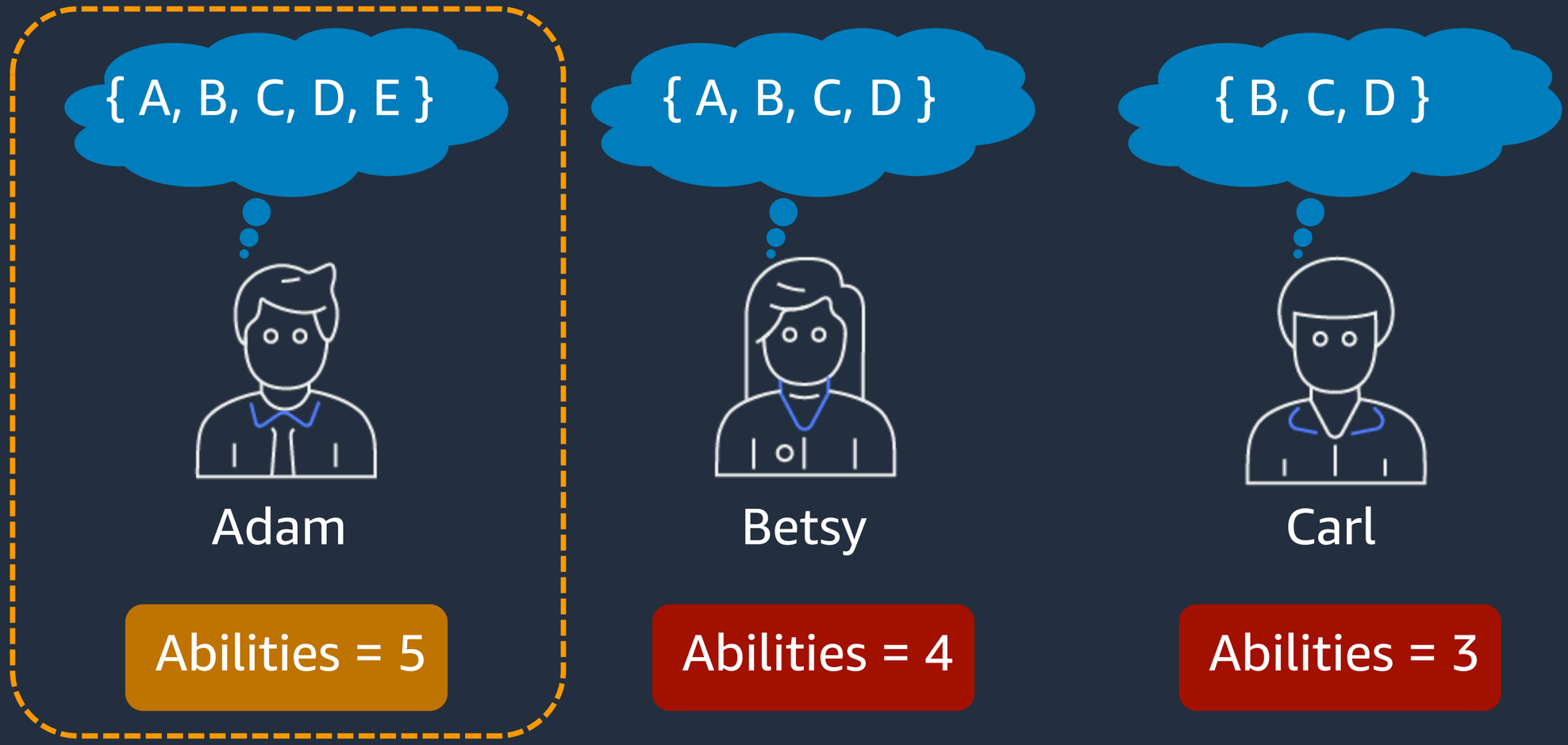
aws modern apps

Diversity bonus model



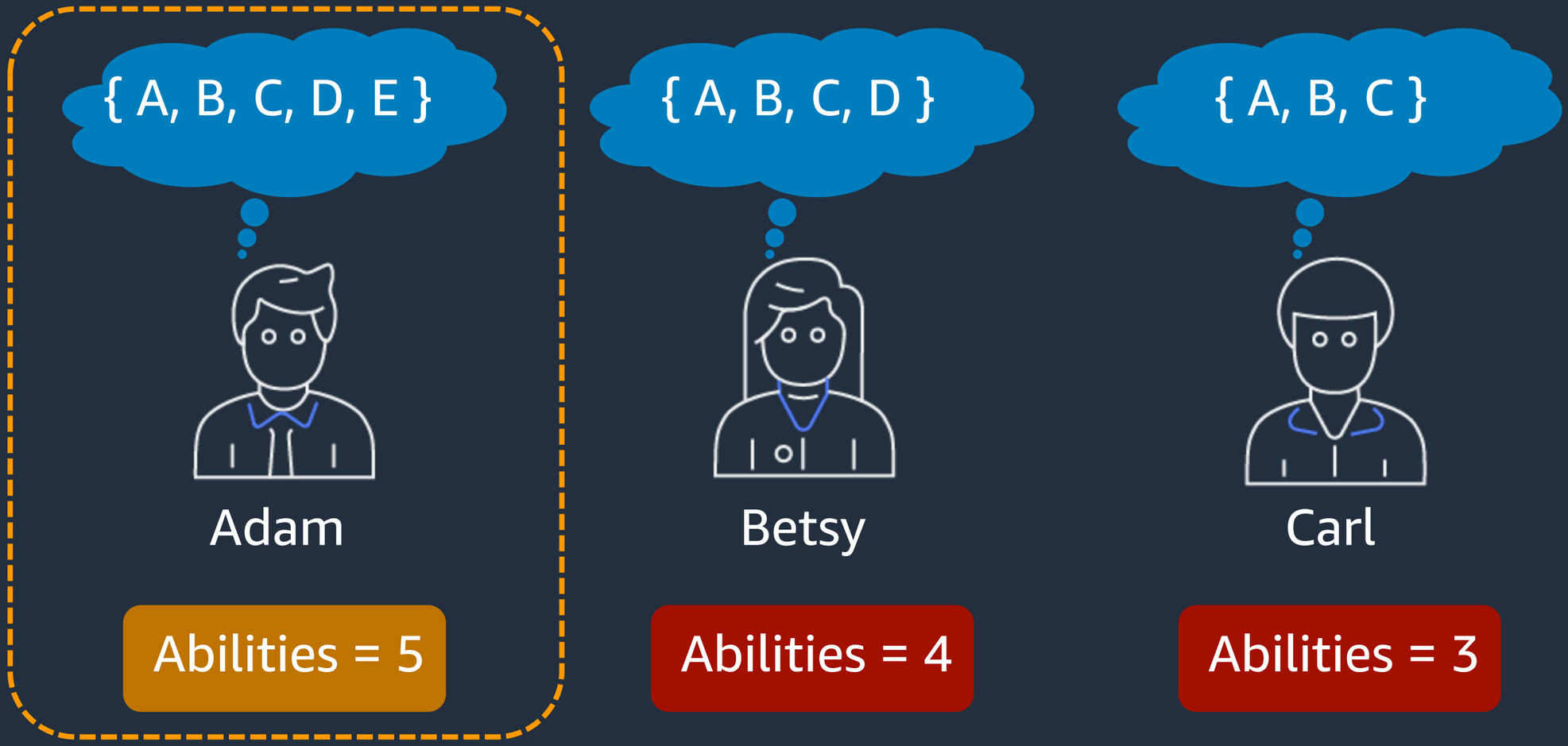
aws modern apps

No diversity, no bonus – Beware hiring managers



aws modern apps

Some cognitive tools must be learned in order



aws modern apps

Small world networks & creativity

2,092 people who worked on
474 musicals from 1945 to 1989

Thursday Oct 13 2005 11:31 AM AJS v111n2 090090 VSJ

Collaboration and Creativity: The Small World Problem[1]

Brian Uzzi
Northwestern University

Jarrett Spiro
Stanford University

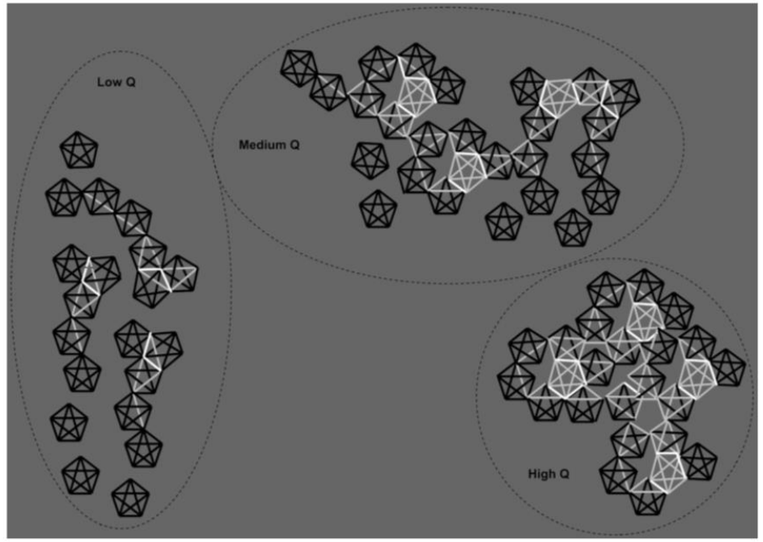
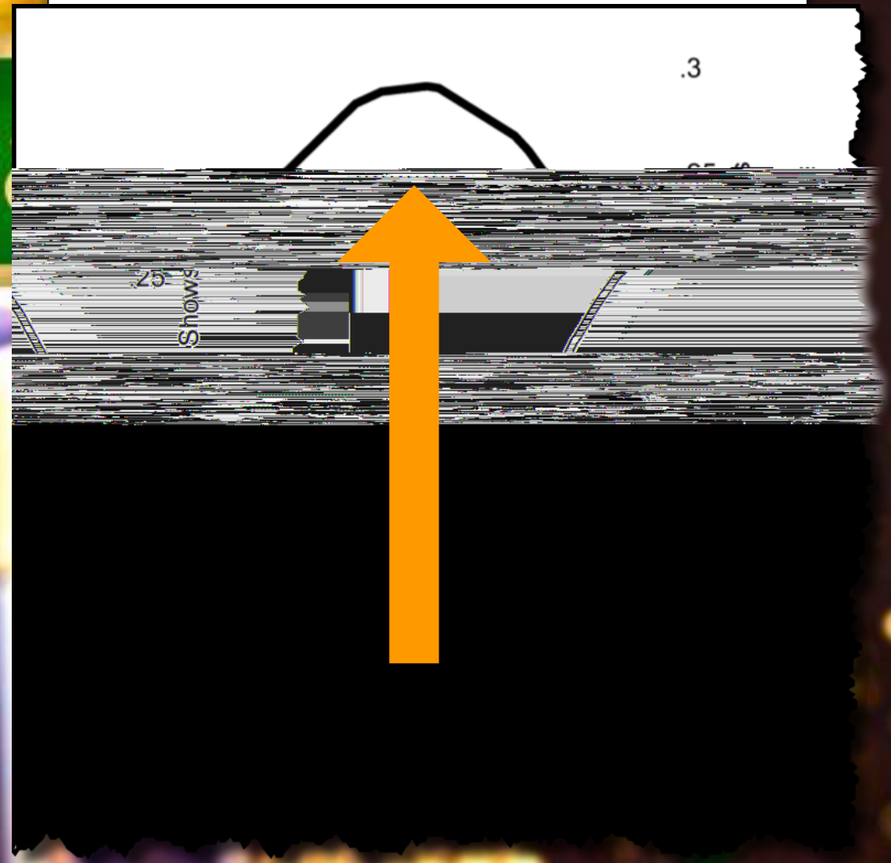


FIG. 3.—Variation in small world network structure. Figure is illustrative but based on the actual structure of our network data for various years. Each inset reflects the structure that is multiplied many times to create a large global network for three levels of Q. When Q is low there are few links between teams (cliques), and the ties that make up these links are not disprotionately made up of repeated and third-party ties as represented by the white (repeated tie) and gray (third-party tie) links. This topology has low connectivity and cohesion. As Q tends toward a high level, there are many betwee- team links, and these links are disproportionately made up of repeated and third-party ties—there is high connectivity and cohesion in the network's topology. At medium levels of Q the small world network has an intermediate amount of connectivity and cohesion.



Source of study -https://www.jstor.org/stable/10.1086/432782?seq=1

What can we build?

Photo by Scott Blake on Unsplash

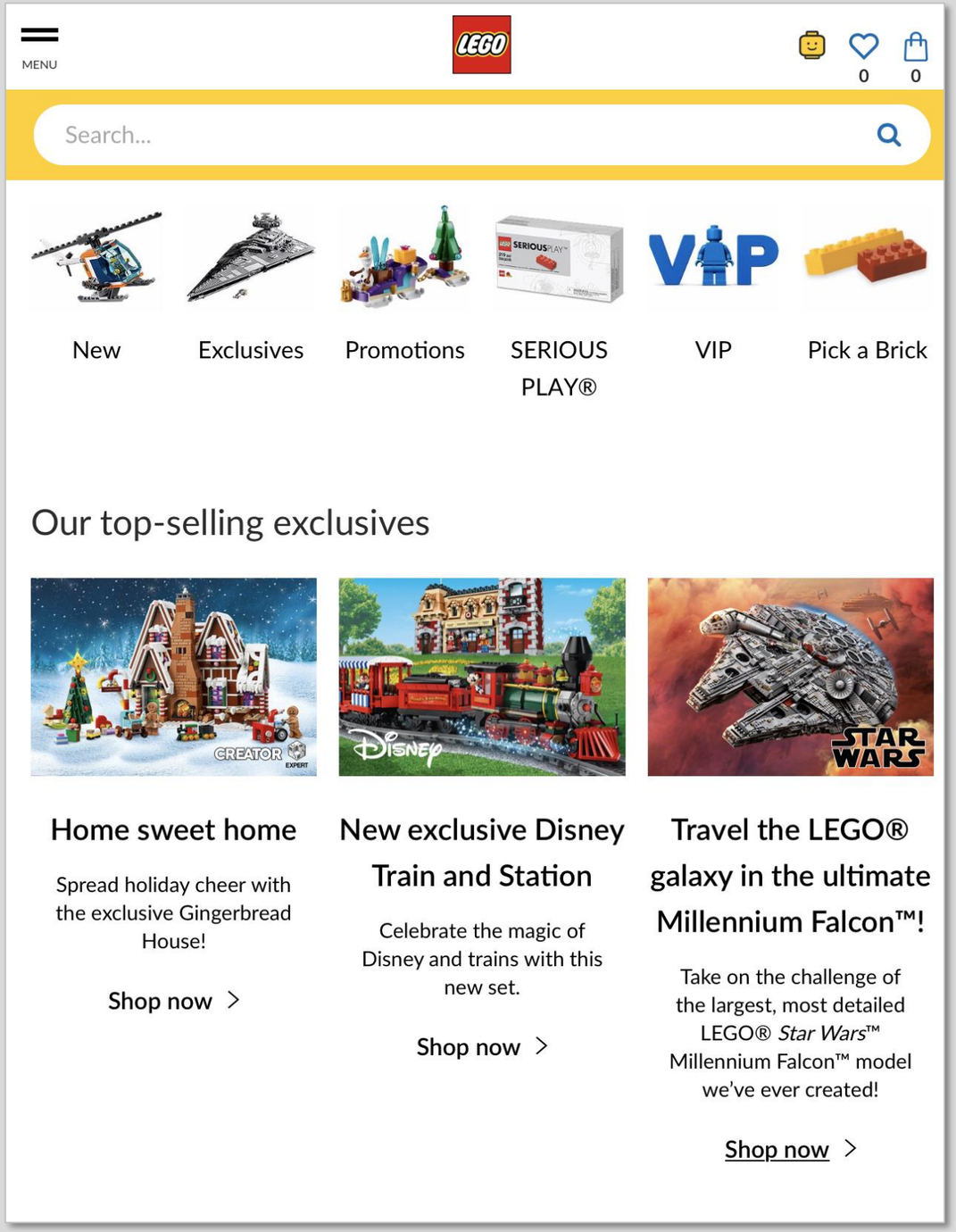
July 10, 2019

shop.LEGO.com

was switched to

serverless

on AWS



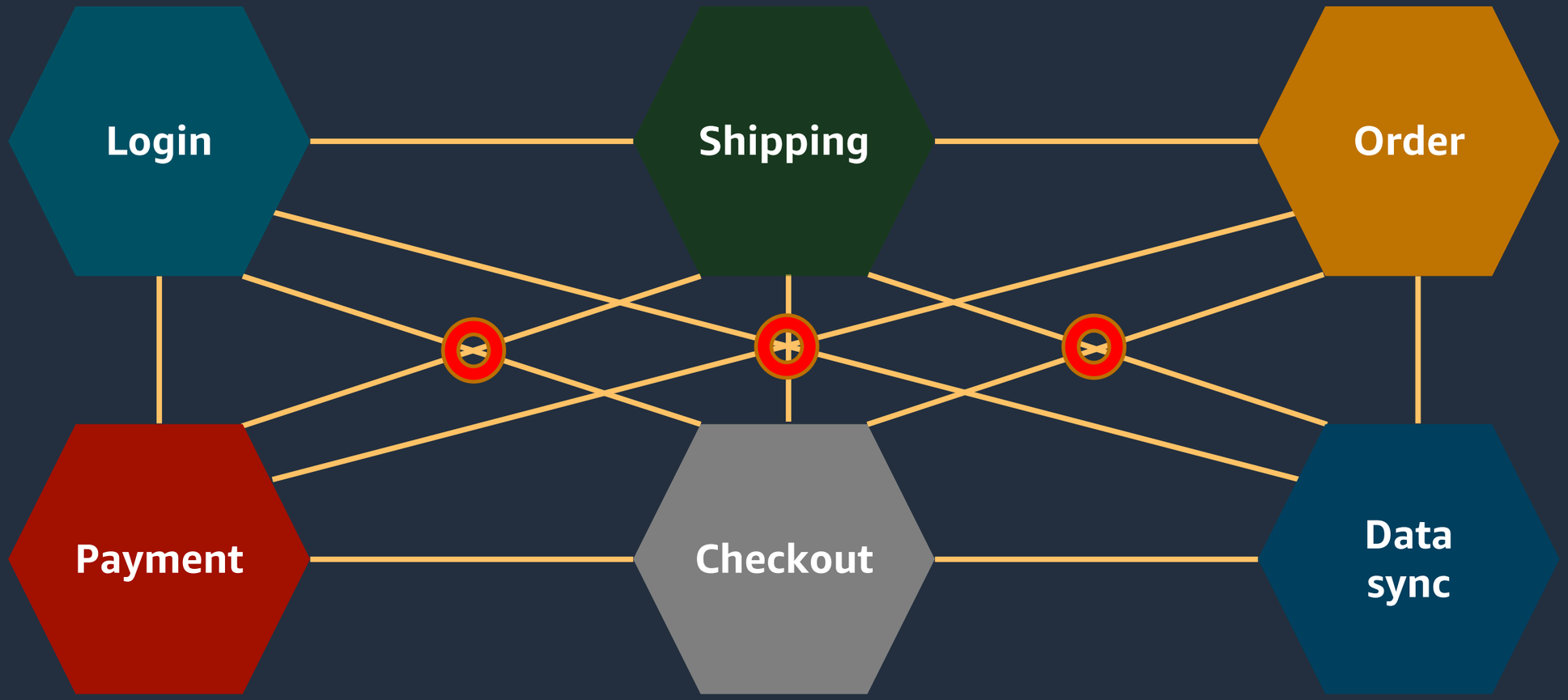
July 11, 2019

The day after,

Amazon EventBridge

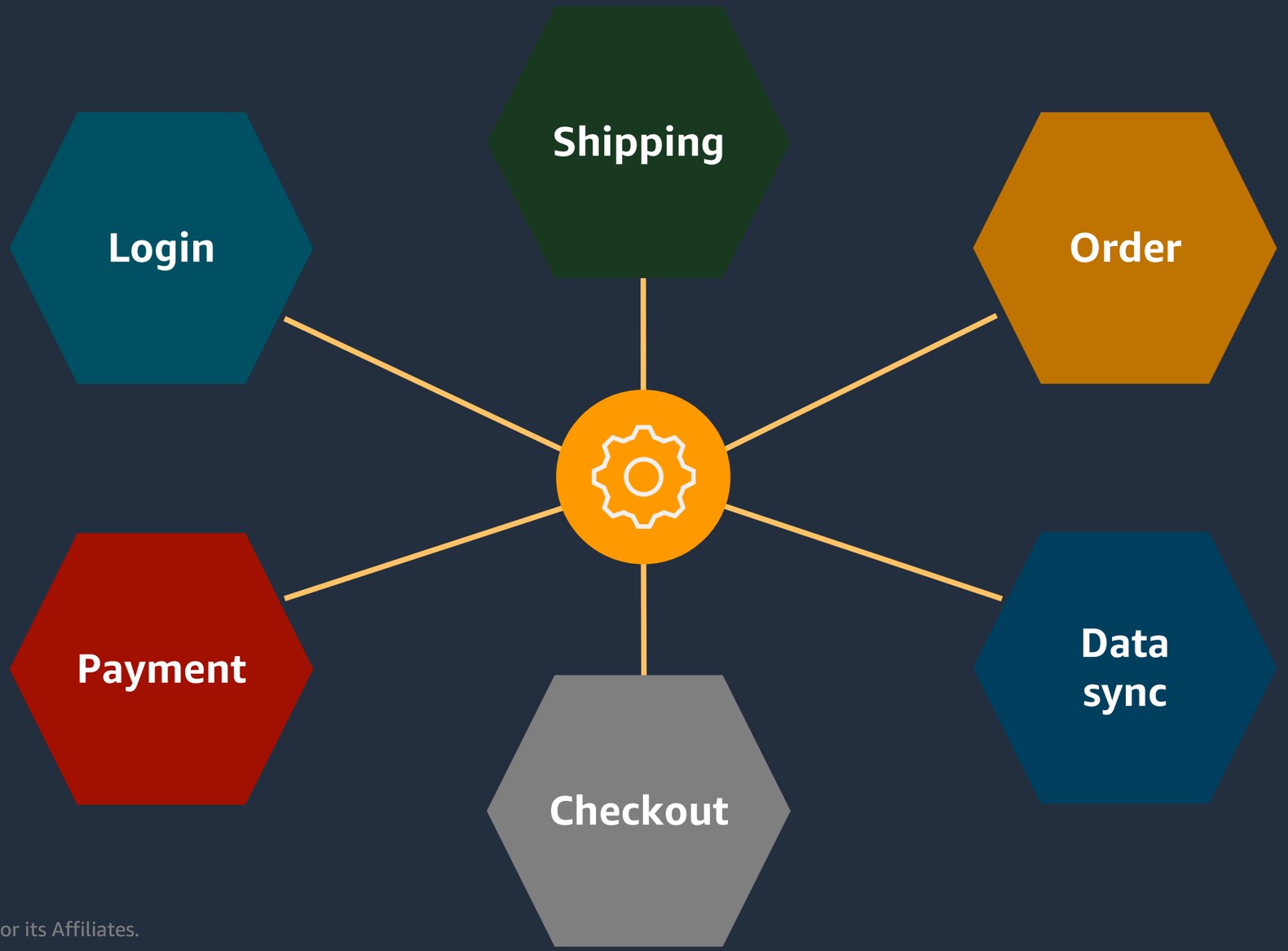
was launched

Checkout event processing



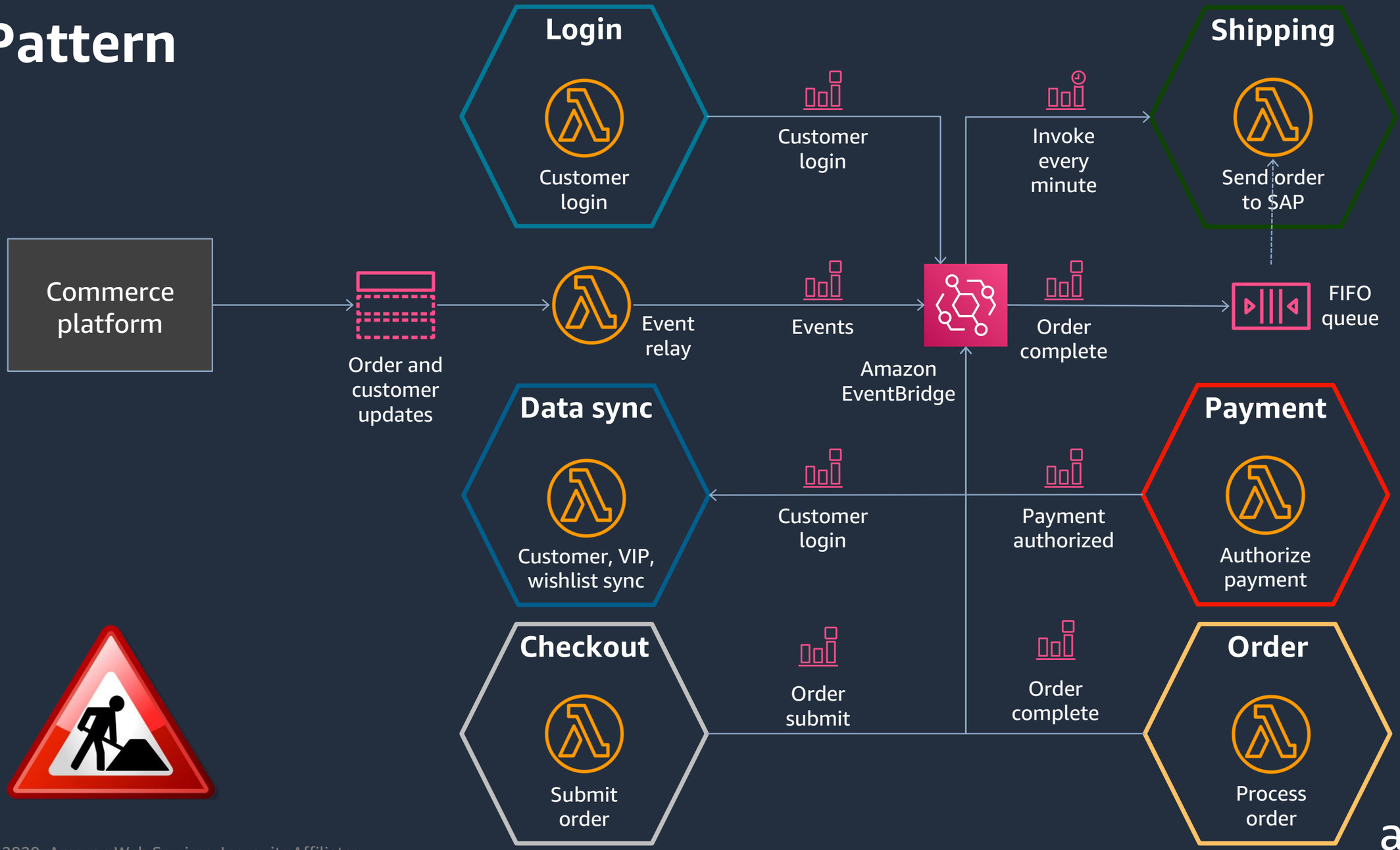
aws modern apps

Hub-and-spoke event bus



aws modern apps

Pattern



aws modern apps

Takeaways

aws modern apps

Takeaways

- Event-driven architectures simplify the integration of distributed architectures
- Guardrails reduce occurrence and blast radius of undesirable application behavior
- Tracing, canaries, and log analytics are fundamental tools to improve observability
- Organize around small, independent, diverse, dynamic teams

Visit the Modern Applications Resource Hub for more resources

Dive deeper with these newly created whitepapers and e-books to accelerate your modernization journey.

- Modern Applications e-book
- Accelerating your AWS journey: Migration & Modernization
- Journey to serverless-first report
- Modernize today with containers on AWS
- … and more!

Visit resource hub »

https://tinyurl.com/aws-modern-apps

aws modern apps

Accelerate Your Modernization Journey

Develop skills in designing, building, and managing modern applications

90% of IT decision makers report cloud skills shortages[1]. A lack of cloud skills impacts modern application development. Start your modern application development journey with AWS Training & Certification.

Take free digital training

With a little time and initiative, learners can enhance their practical cloud knowledge through free digital training. These on-demand courses, which vary in length from 10 minutes to several hours, can help one broaden their understanding of specific subjects such as serverless, containers, and developer tools.

Get live, hands-on, instructor-led training

Whether physical or virtual, classroom training offers more in-depth instruction for people who want to deepen their technical skills. Classes are a mix of presentations, hands-on labs, and group discussions led by experts in their fields. Courses include Developing on AWS and Advanced Developing on AWS.

Quickly ramp up your modern application skills

Independent learning allows people to fill in knowledge gaps and learn new topics at their own pace. There's a wide range of whitepapers, blog posts, videos, webinars, use cases, and peer resources available for IT professionals who want to dive deep into specific technical topics. Learn more.

[1] 451 Research, *Demystifying Cloud Transformation: Where Enterprises Should Start,* September 2019.

aws modern apps

Thank you for attending AWS Modern Applications Online Series

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-apac-marketing@amazon.com


twitter.com/AWSCloud


facebook.com/AmazonWebServices


youtube.com/user/AmazonWebServices


slideshare.net/AmazonWebServices


twitch.tv/aws

aws modern apps

aws modern apps

Thank you!

@danilop