Google

SRE Fundamentals: Developing & calculating SLOs

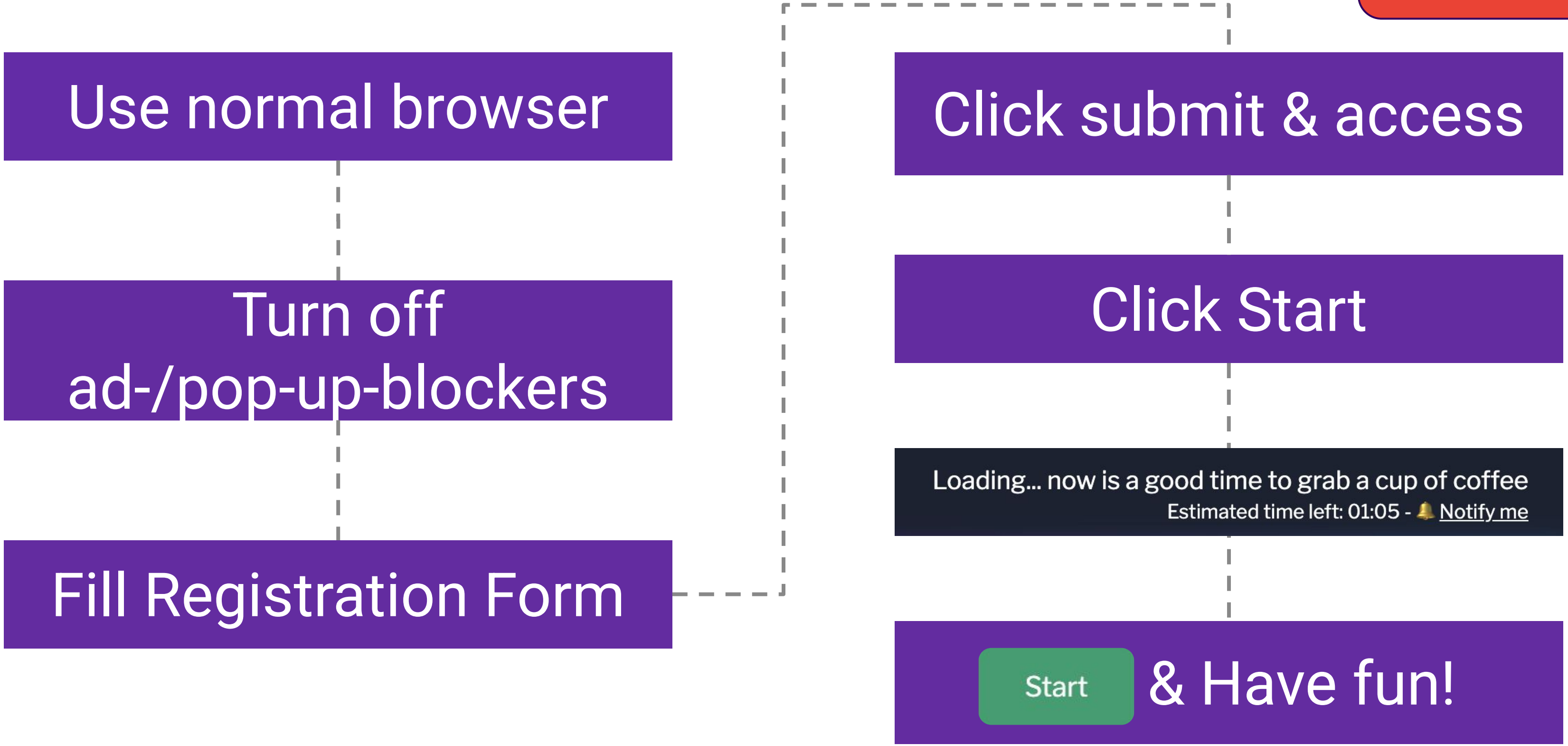
February 4th, 2025

Jose Andrade

CE SRE Global Lead

Starting your personal lab

JOIN THE WORKSHOP
dtdg.co/srelab



Use normal browser

Turn off ad-/pop-up-blockers

Fill Registration Form

Click submit & access

Click Start

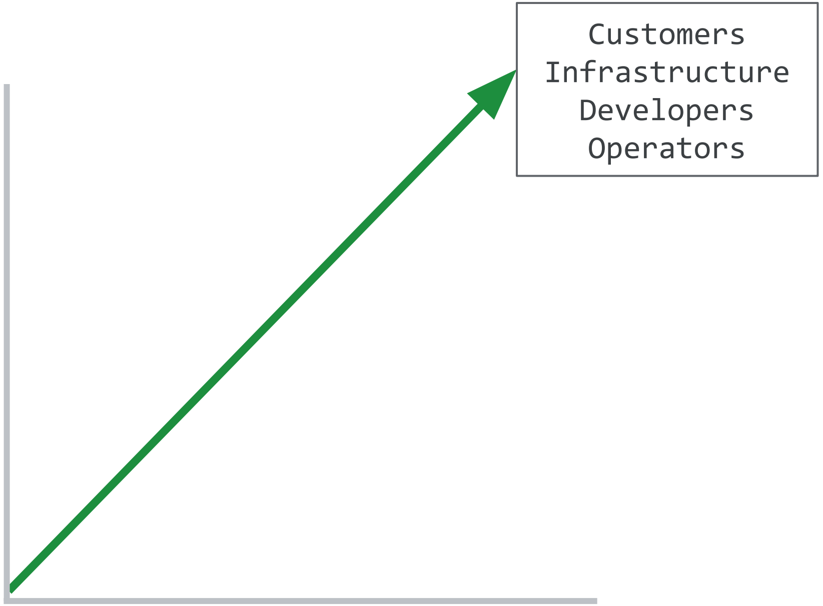
Loading... now is a good time to grab a cup of coffee
Estimated time left: 01:05 - 🔔 Notify me

Start & Have fun!

DATADOG | Google Cloud

Why SRE?

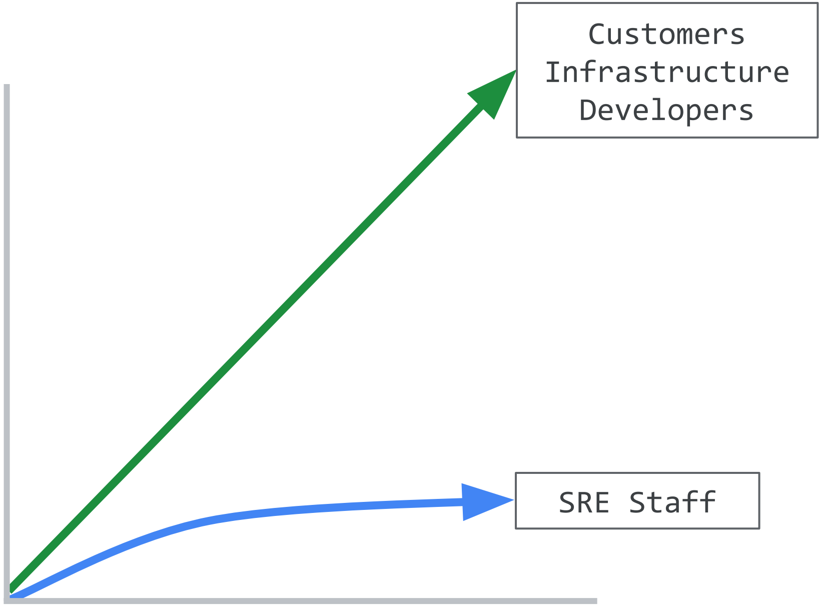
Why SRE? Scaling operations problem



Linear scaling

The number of operators needs to scale proportionally with the size and scope of any product they maintain.

Why SRE? Scaling operations problem



Sublinear scaling solution

- Automation -> Self-healing
- Standardized tooling
- Community of practice
- Shared responsibility

> “SRE is what happens when you ask a software engineer to design an operations team.”

Benjamin Treynor Sloss
Vice President of 24x7 Engineering, Google

What is SRE?

Key Principles

Principle #1

The **Most Important Feature** of Any System is its **Reliability**

Principle #2

Our Monitoring Doesn’t Decide Our Reliability - **Our Users Do**

Principle #3

Well engineered software can only get you to **99.9%**

Well engineered operations -> **99.99%**

Well engineered business -> **99.999%**

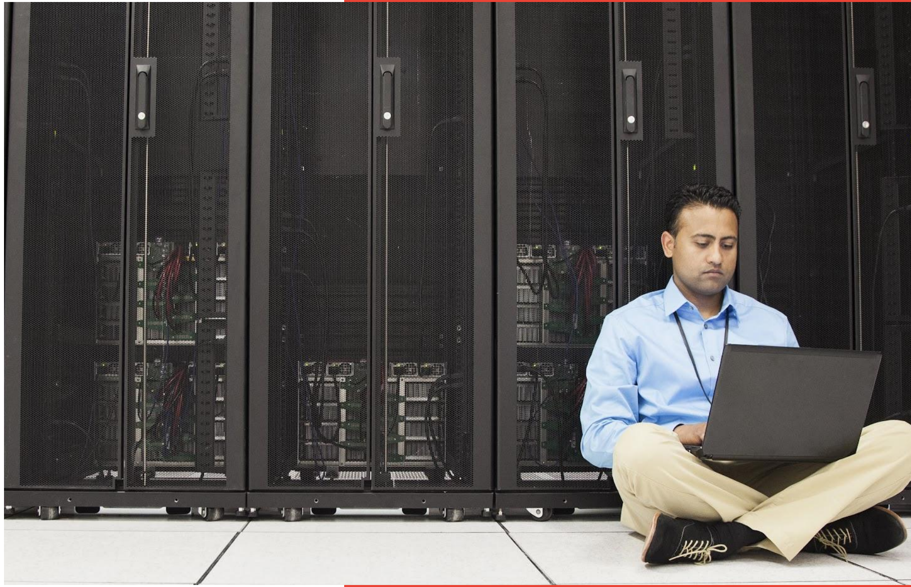
> “100% is the wrong reliability target for basically everything.”

Benjamin Treynor Sloss
Vice President of 24x7 Engineering, Google

At Google, we constantly enhance our services with new features and add new systems.

Incidents and outages are inevitable given the velocity of change.

John Lunney & Sue Lueder
Site Reliability Engineers



What SRE is not...

✗ SREs are not System Administrators.

✗ SREs are not Desktop Support Technicians for developers.

✗ SREs are not customer support.

✗ The SRE model doesn't apply fully to COTS applications.

✗ SRE is not for every application. Only business-critical ones.

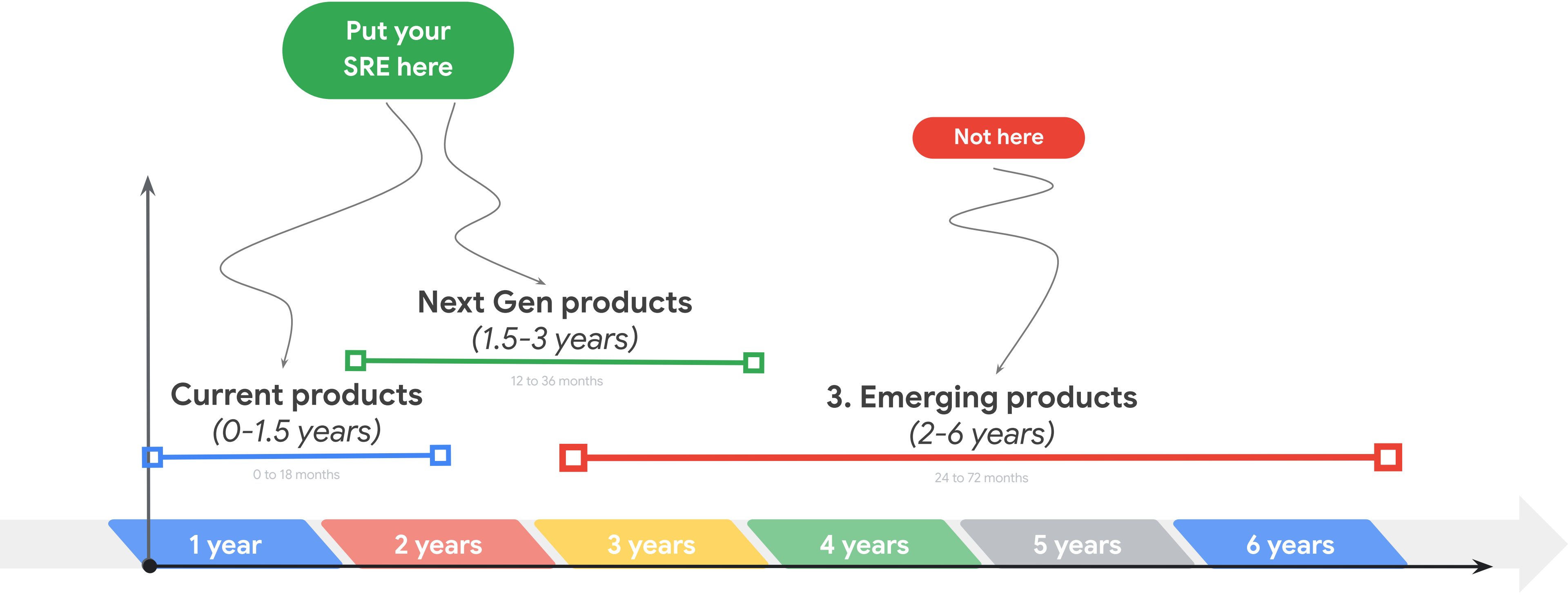
✗ SREs are not application developers (those are SWEs).

✗ SREs don't "own" the applications (SWEs do).

✗ SREs don't work in a vacuum.

- Other teams **must** help defend reliability

Where to put SRE?



Baghai, M., Coley, S., & White, D. (2000). In The alchemy of growth: Practical insights for building the enduring enterprise. Perseus Books.

SRE Practices

Areas of practice

Observability

- Logs, metrics, and traces
- Paging *vs.* ticketing
- Involve humans for serious threats to SLO
- Triggers, actions

Capacity Planning

- Isolation strategies
- Organic growth
- Inorganic growth:
  - BFCM
  - COVID-19
- Buffer capacity

Change Management

- Progressive rollouts
- Efficient rollbacks
- Remember: ~70% of outages are caused by changes

Emergency Response

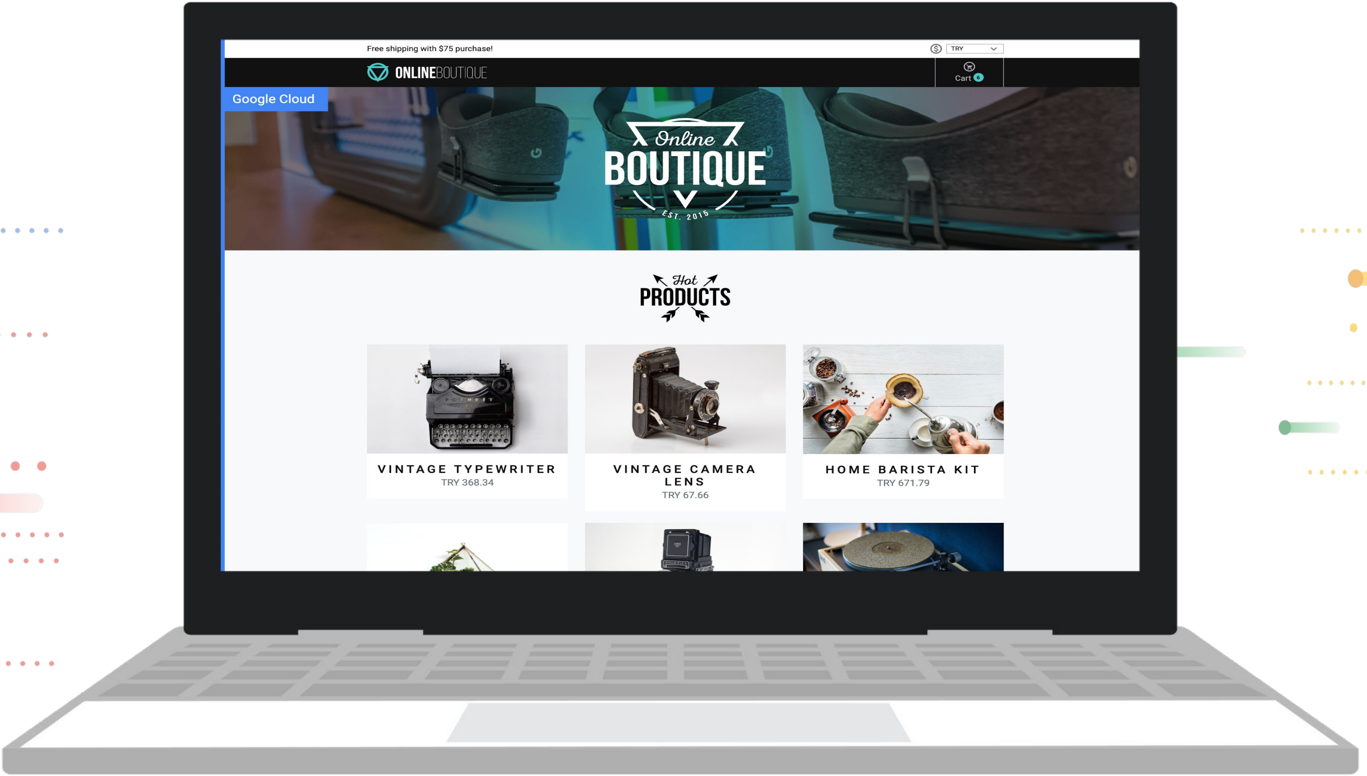
- Clear outage thresholds
- Pre-defined RACI
- Playbooks & documentation

Culture

- Psychological safety
- Blamelessness
- Data-driven

SRE Vocabulary

Online Boutique



Free shipping with $75 purchase!

TRY

ONLINEBOUTIQUE

Cart

Google Cloud

Online BOUTIQUE EST. 2015

Hot PRODUCTS

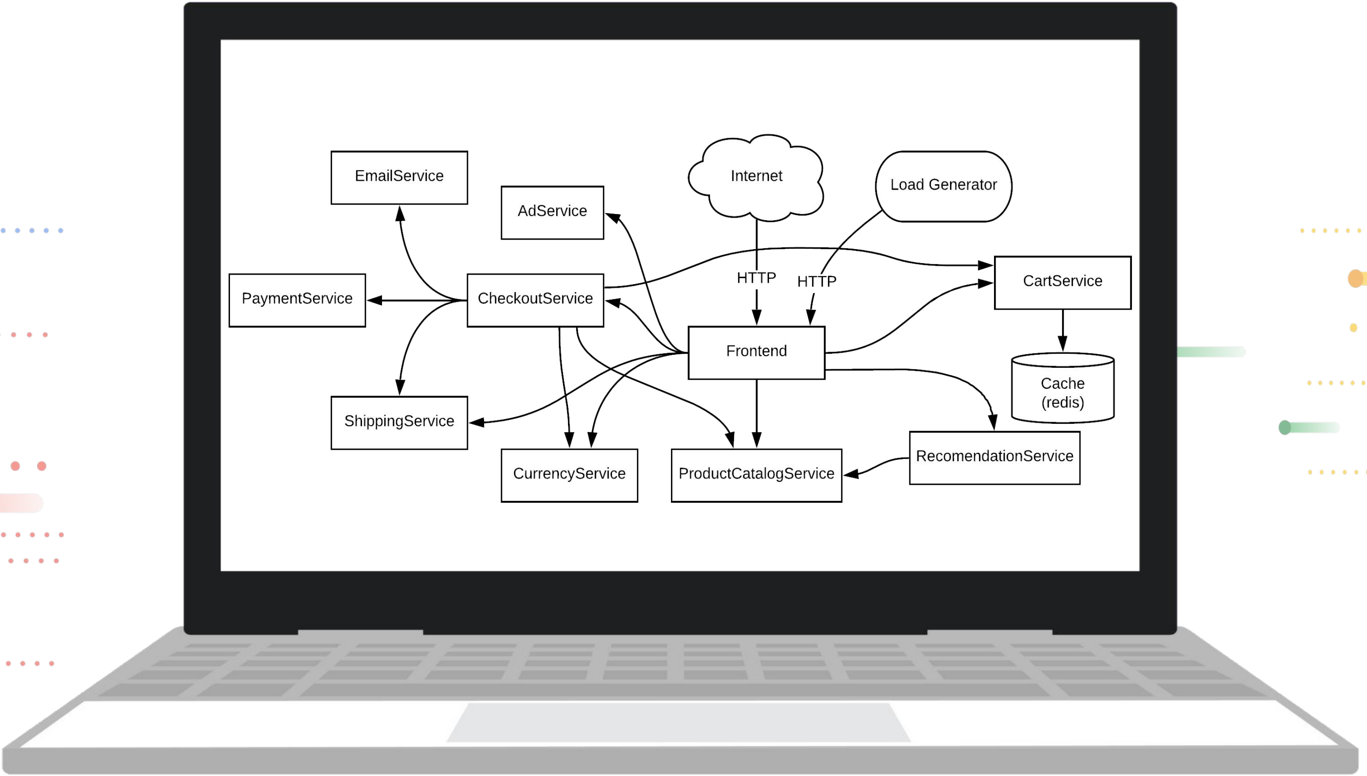
VINTAGE TYPEWRITER
TRY 368.34

VINTAGE CAMERA LENS
TRY 67.66

HOME BARISTA KIT
TRY 671.79

https://github.com/GoogleCloudPlatform/microservices-demo

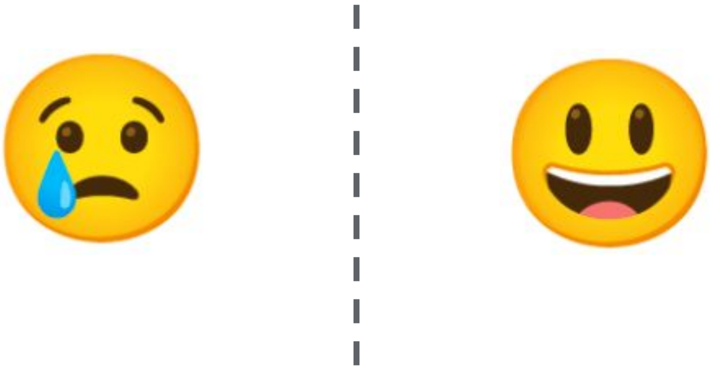
Online Boutique



https://github.com/GoogleCloudPlatform/microservices-demo

What is “Reliability”?

Users are happy because we are meeting their implicit and explicit expectations



Glossary of terms

CUJ

critical **user journey**: specific steps that a user takes to accomplish a goal

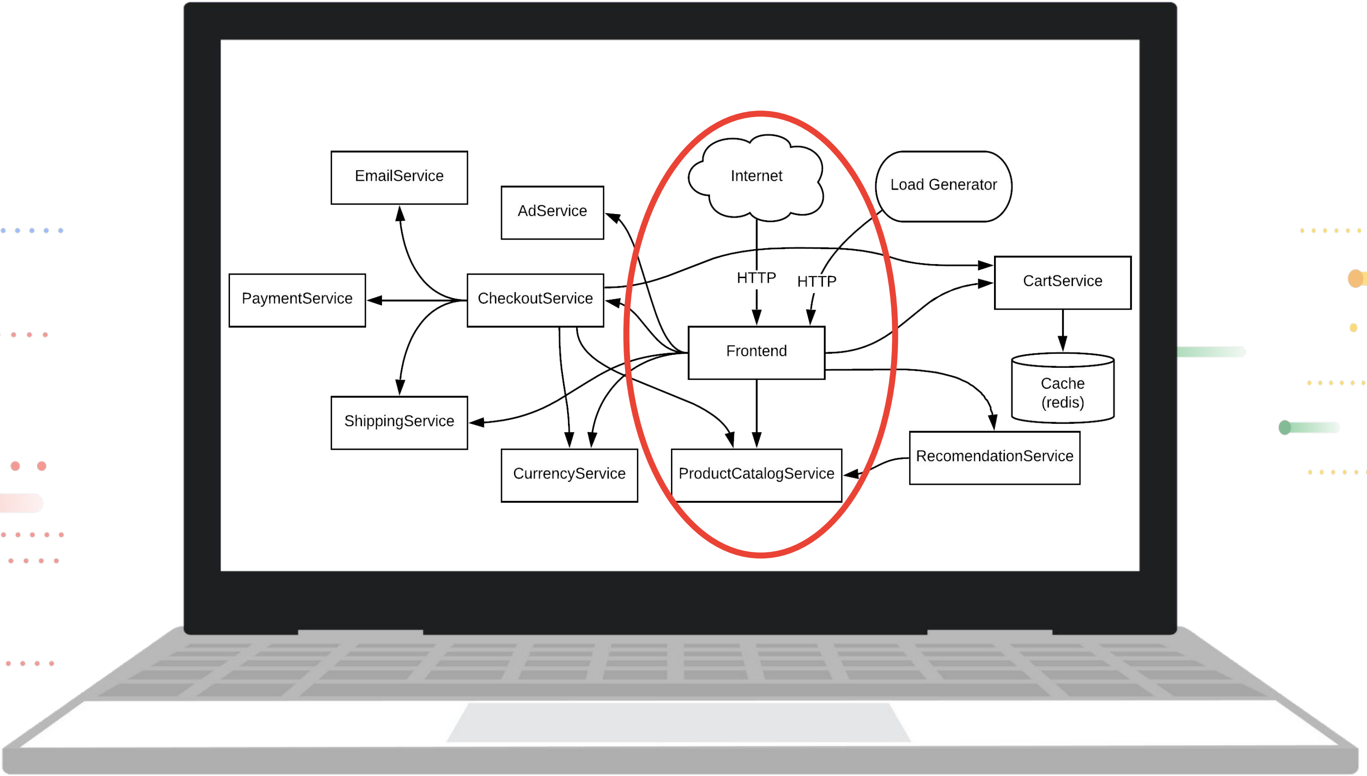
Critical User Journey (CUJ)

User interacts with Service to achieve Goal

Critical User Journey (CUJ)

As a shopper I want to see items for sale in the store.

Online Boutique



https://github.com/GoogleCloudPlatform/microservices-demo

Glossary of terms

CUJ	SLI
critical **user journey**: specific steps that a user takes to accomplish a goal	service level **indicator**: a well-defined measure of success

Service Level Indicators (SLI)

Quantitative and carefully-defined as seen in the following equation:

$$\text{SLI} : \left(\frac{\text{good events}}{\text{valid events}} \right)$$

Monitoring systems may (and should) capture a large number of potential SLIs, but most are not immediately useful to back objectives

SLI Types

SLI Menu

Request / Response	Availability Latency Quality
Data Processing	Coverage Correctness Freshness Throughput
Storage	Throughput Latency

Service Level Indicator (SLI)

Good responses from the catalog

Service Level Indicator (SLI)

SLI Type (consider the SLI Menu):

Request/Response:

Availability, Latency, Quality

SLI Specification: an assessment of service outcome that you think matters to users

SLI Implementation: a way to measure the SLI specification

Service Level Indicator (SLI)

Good responses from the catalog

SLI Type: Availability

SLI Specification: Requests to the ProductCatalogService that return HTTP response code 2xx, 3xx, or 4xx (excl. 429)... divided by valid Requests (eg: 5xx included)

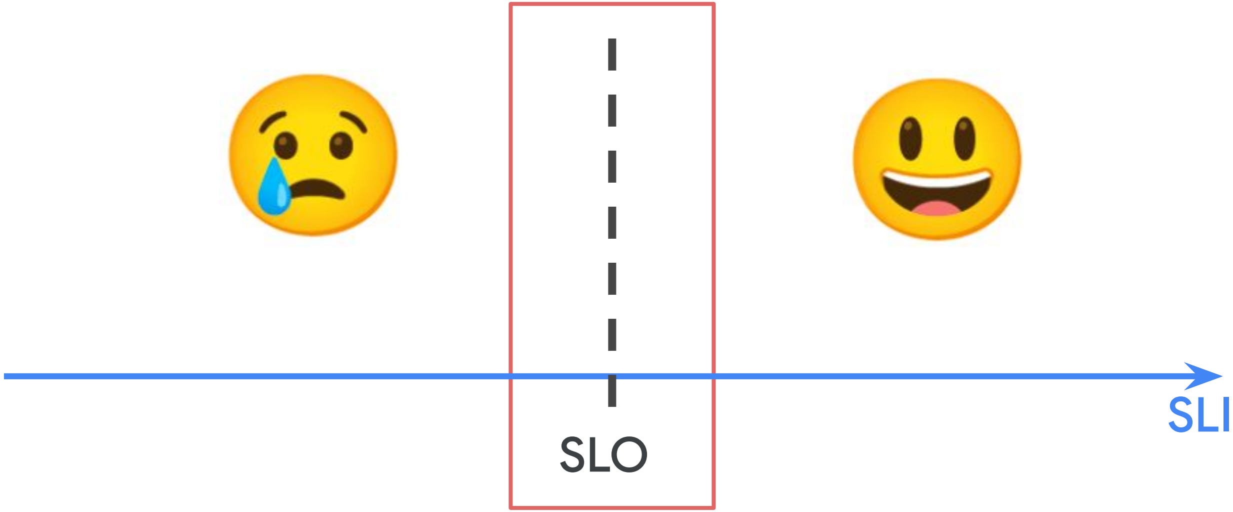
SLI Implementation: Identify HTTP response codes from web logs

Glossary of terms

CUJ	SLI	SLO
critical **user journey**: specific steps that a user takes to accomplish a goal	service level **indicator**: a well-defined measure of success	service level **objective**: a top-line target for fraction of successful interactions

What is “Reliability”?

Users are happy because we are meeting their implicit and explicit expectations



Service Level Objective (SLO)

The catalog must return 99.9% good responses

Service Level Objective (SLO)

The catalog must return 99.9% good responses

An SLO must contain a target and a compliance window

Service Level Objective (SLO)

The catalog must return 99.9% good responses

An SLO must contain a target and a compliance window

99.9% of Product Catalog Service requests in the past 28 days are successful

CUJ, SLI, and SLO

CUJ: As a shopper I want to see items for sale in the store.

SLI Type: Availability

SLI Specification: Requests to the Product Catalog Service that return HTTP response code 2xx, 3xx, or 4xx (excl. 429)... divided by valid Requests (eg: 5xx included)

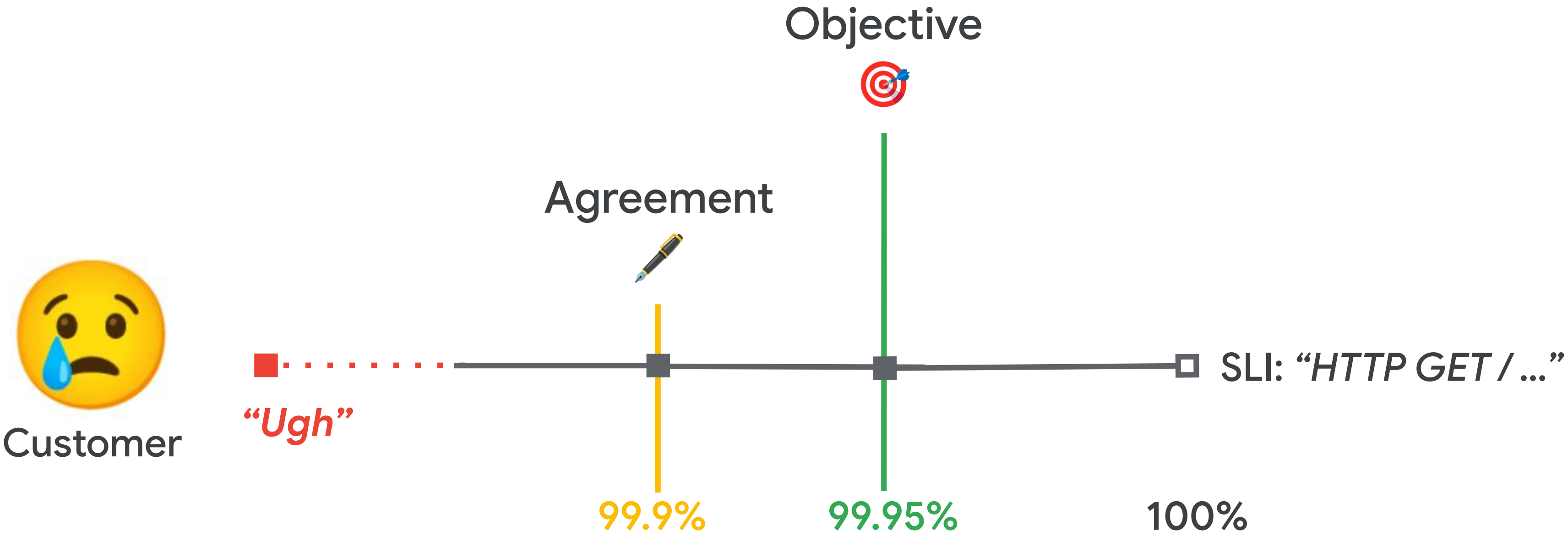
SLI Implementation: Identify HTTP response codes from web logs

SLO: 99.9% of Product Catalog Service requests in the past 28 days are successful

Glossary of terms

CUJ	SLI	SLO	*SLA*
critical **user journey**: specific steps that a user takes to accomplish a goal	service level **indicator**: a well-defined measure of success	service level **objective**: a top-line target for fraction of successful interactions	*service level **agreement**: business consequences*

SLO vs SLA

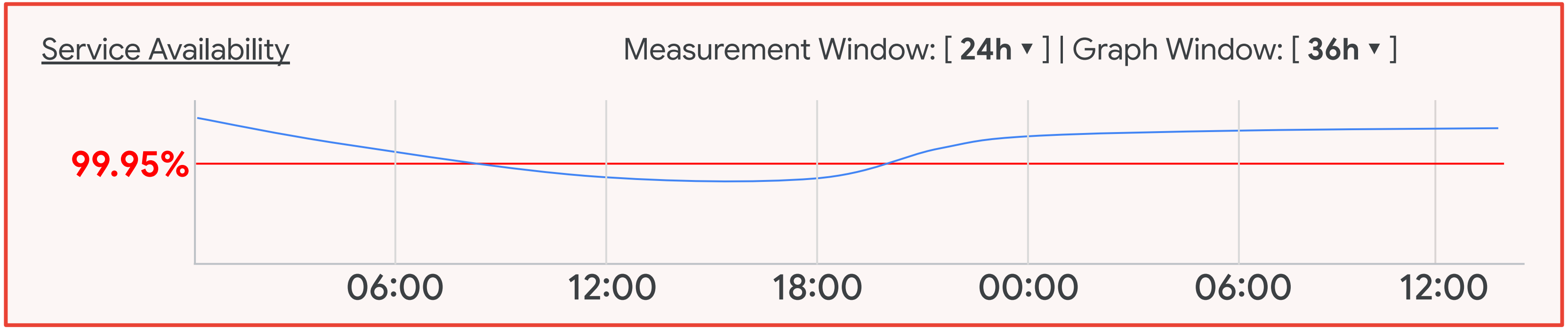


Glossary of terms

CUJ	SLI	SLO	Error Budget	*SLA*
critical **user journey**: specific steps that a user takes to accomplish a goal	service level **indicator**: a well-defined measure of success	service level **objective**: a top-line target for fraction of successful interactions	proportion of **“affordable” unreliability**; one minus the SLO To be continued...	*service level* ***agreement:*** *business consequences*

Example

Documenting SLOs



Service Availability

Measurement Window: [**24h** ▾] | Graph Window: [**36h** ▾]

Target: 99.95% Details

Owner: someuser@example.com (Product Manager)

Contacts: otheruser@example.com (Tech Lead), service-sre@example.com (SRE)

Status: Signed off 2017-09-24. Paging service-sre.

Rationale: Drops below 99.95% availability correlate with increased support tickets and user complaints.

References:

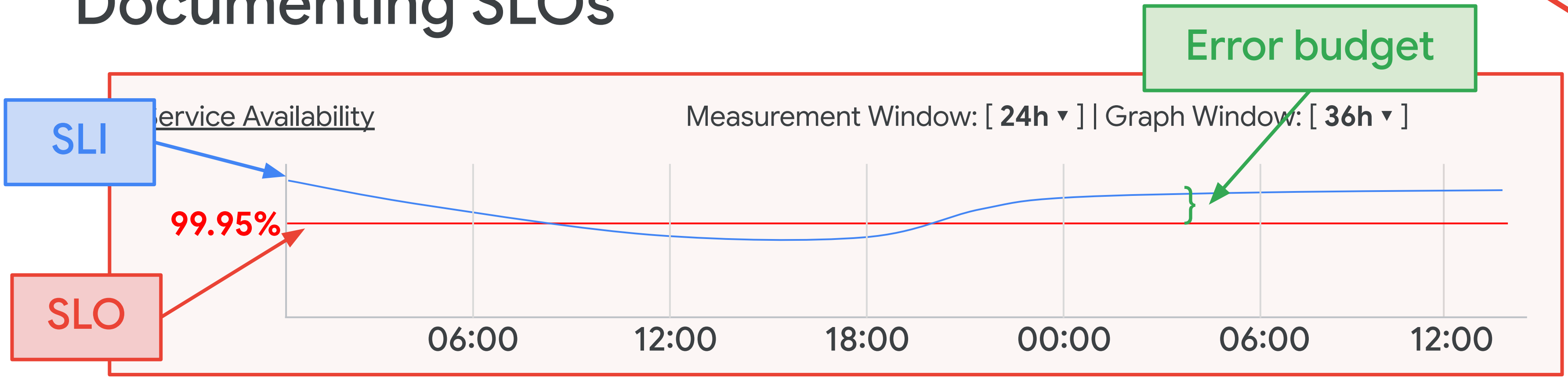
- https://corp.example.com/teams/service-sre/postmortems/2017-09-18-database-write-failures

Changelog:

- **2017-09-24**: Previous availability SLO of 99.9% is too low. A recent outage resulted in lots of complaints on Twitter and our support teams missing their response time SLAs due to overload.

Documenting SLOs

Example



Service Availability

Measurement Window: [**24h** ▾] | Graph Window: [**36h** ▾]

Include CUJ, SLI, SLO

Target: 99.95% Details

Owner: someuser@example.com

Contacts: otheruser@example.com (Tech Lead), service-sre@example.com (SRE)

Status: Signed off 2017-09-24. Paging service-sre.

Rationale: Drops below 99.95% availability correlate with increased support tickets and user complaints.

References:

- https://corp.example.com/teams/service-sre/postmortems/2017-09-18-database-write-failures

Changelog:

- **2017-09-24**: Previous availability SLO of 99.9% is too low. A recent outage resulted in lots of complaints on Twitter and our support teams missing their response time SLAs due to overload.

Calculating SLOs

Q: Can you build 99.99% services on 99.9% infra?

Q: Can you build 99.99% services on 99.9% infra?

Yes.

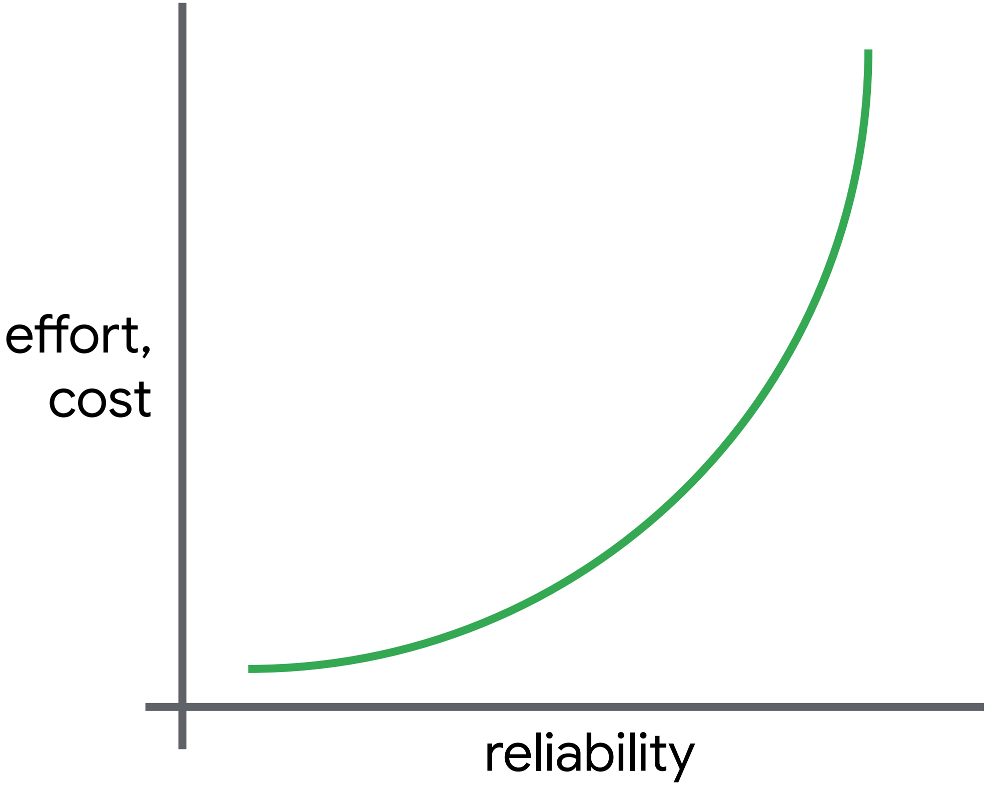
You can build

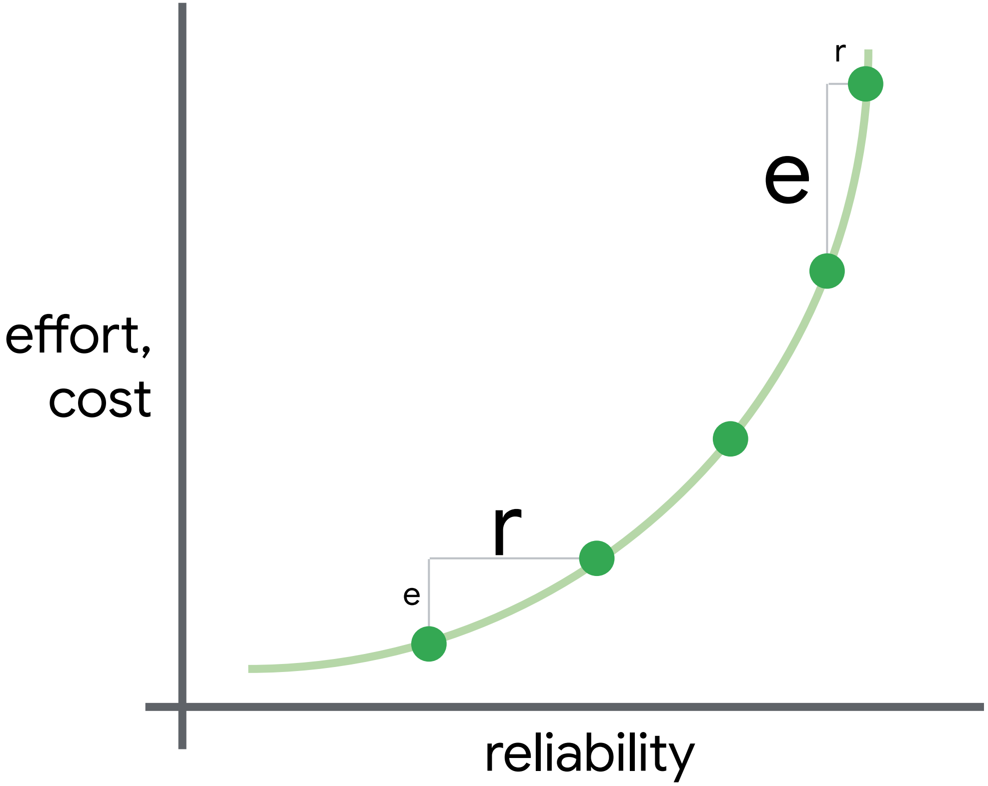
more reliable things

on top of

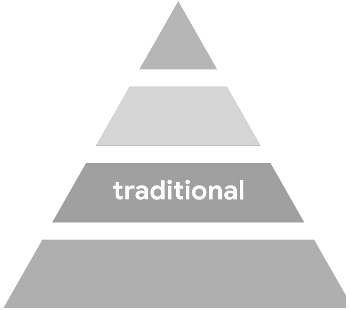
less reliable things.

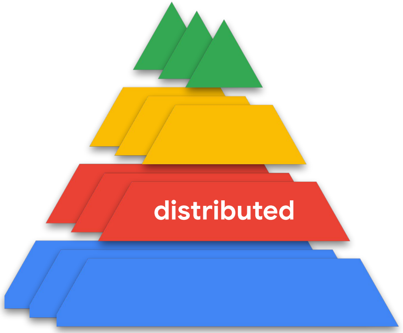
Remember RAID?


effort,
cost
reliability


r
e
effort,
cost
r
e
reliability

Component Reliability





Component reliability

- Software **inherits** reliability from the base
- Lower levels **must be** more reliable

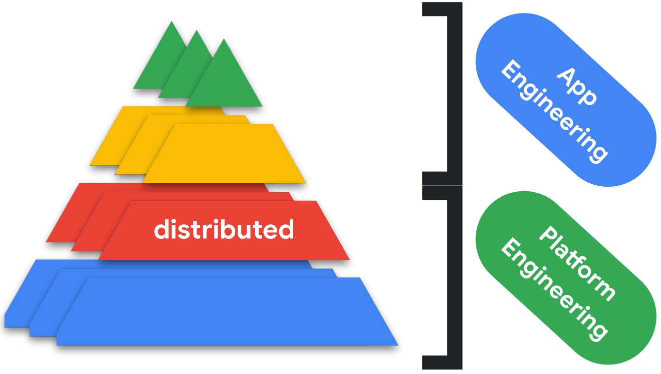
Scalable reliability

- Software **improves** reliability
- Lower levels **can be** less reliable

(*) Result of commoditization of distributed systems

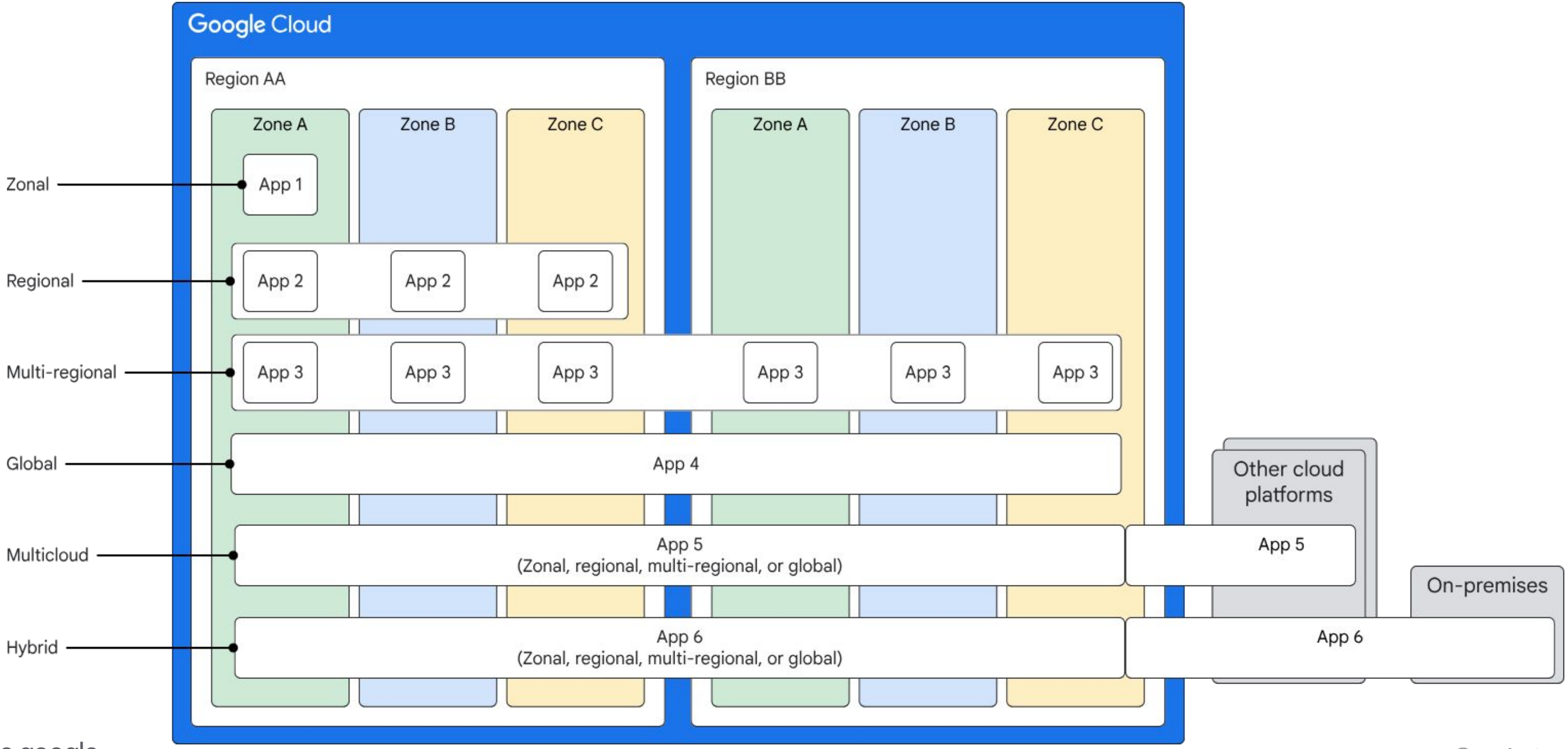
Separate Apps from Platforms

- ✗ Don't force / demand / set % adoption targets!
- ✗ Don't start with the most critical apps!
- ✗ Don't waterfall !

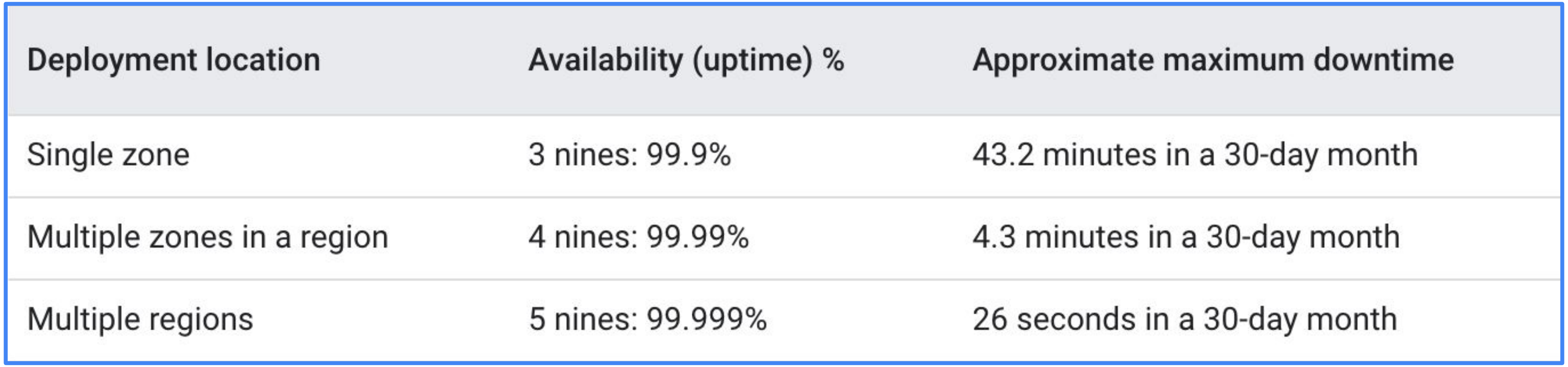


- ✓ Do let teams adopt the platform at their own pace
- ✓ Do celebrate early-adopters publicly, share wins
- ✓ Do listen to dev teams as your customers

The six deployment archetypes

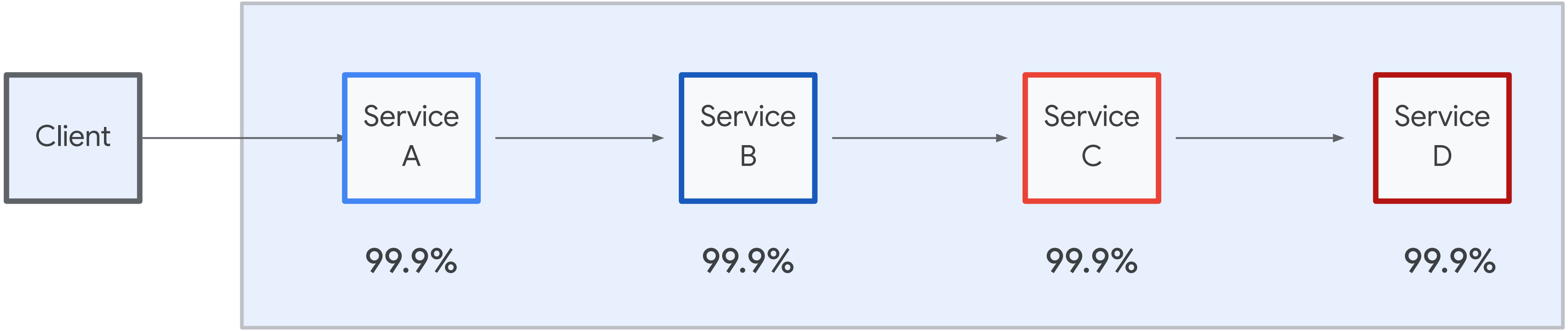


Google Infrastructure is designed to support the following target levels of availability for most customer workloads



Deployment location	Availability (uptime) %	Approximate maximum downtime
Single zone	3 nines: 99.9%	43.2 minutes in a 30-day month
Multiple zones in a region	4 nines: 99.99%	4.3 minutes in a 30-day month
Multiple regions	5 nines: 99.999%	26 seconds in a 30-day month

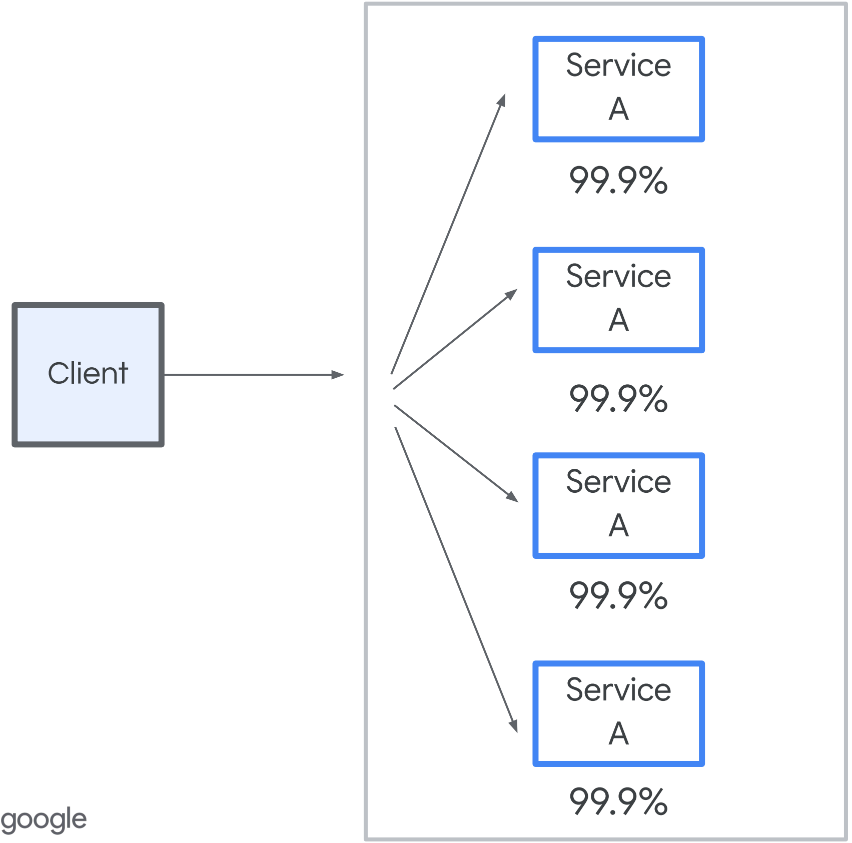
Serial Service Architecture



0.999 x 0.999 x 0.999 x 0.999

99.6% SLO

Parallel Service Architecture



$1 - (0.001)^4$

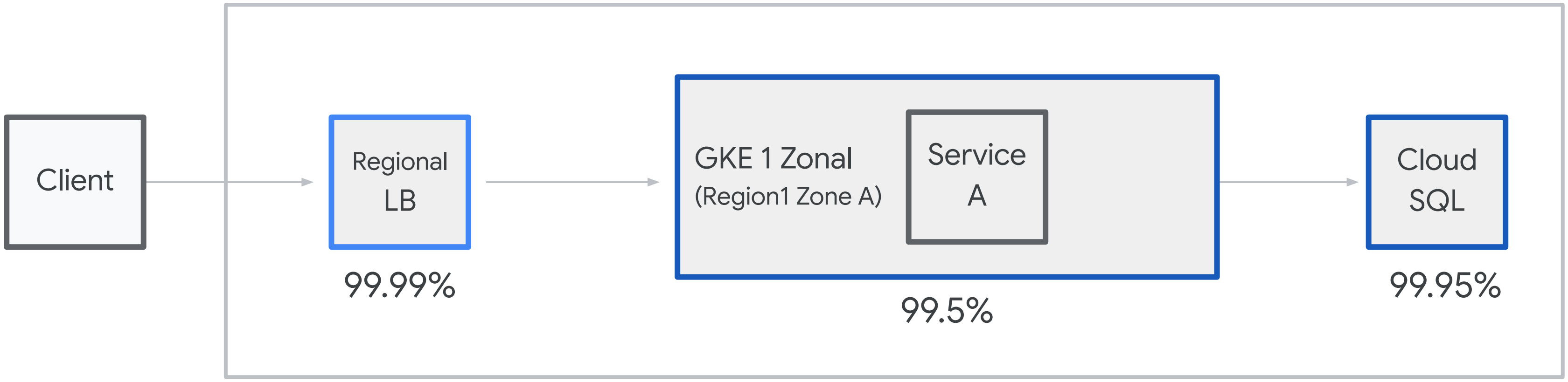
99.999999999% SLO

or 11 nines

This is strictly mathematical and does not include any dependent variables like network, LBs, capacity planning, connectivity, and other dependent services

Building Reliable Platforms on GCP

Single zonal GKE cluster with Cloud SQL



0.9999 x 0.995 x 0.9995

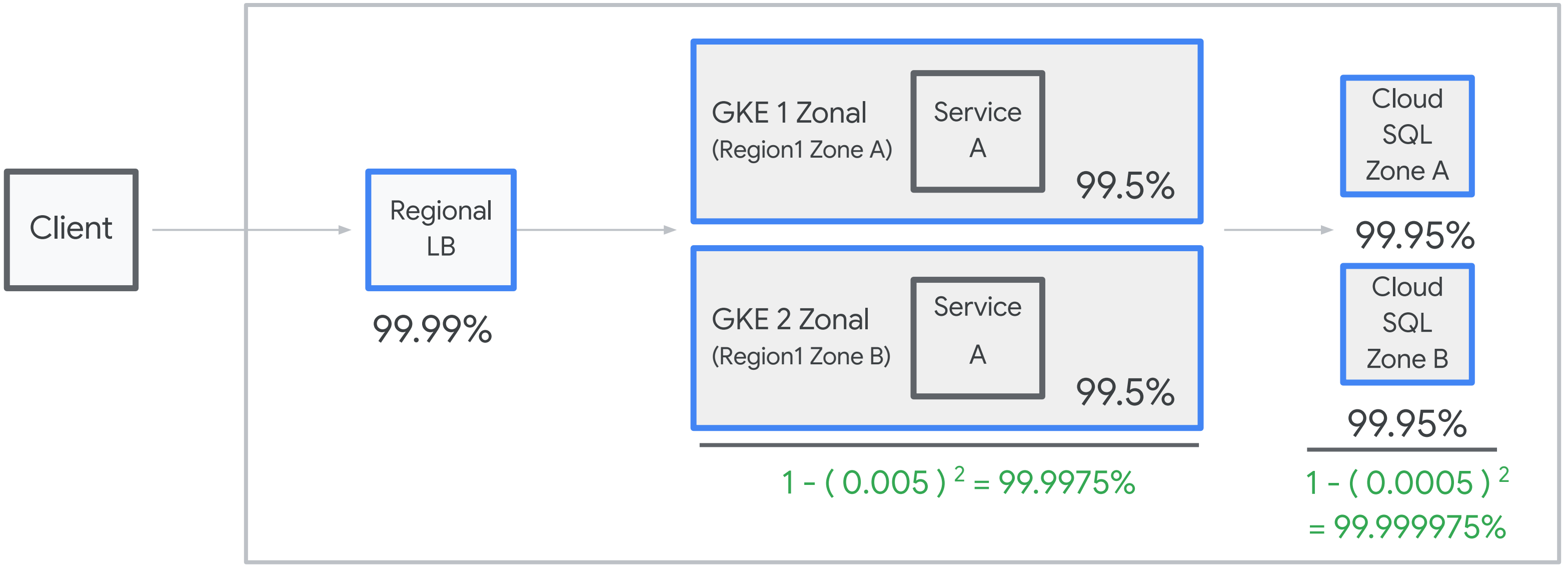
99.44% SLO

[GCP SLAs]

sre.google

Google Cloud

Active passive zone with Cloud SQL HA



$1 - (0.005)^2 = 99.9975\%$

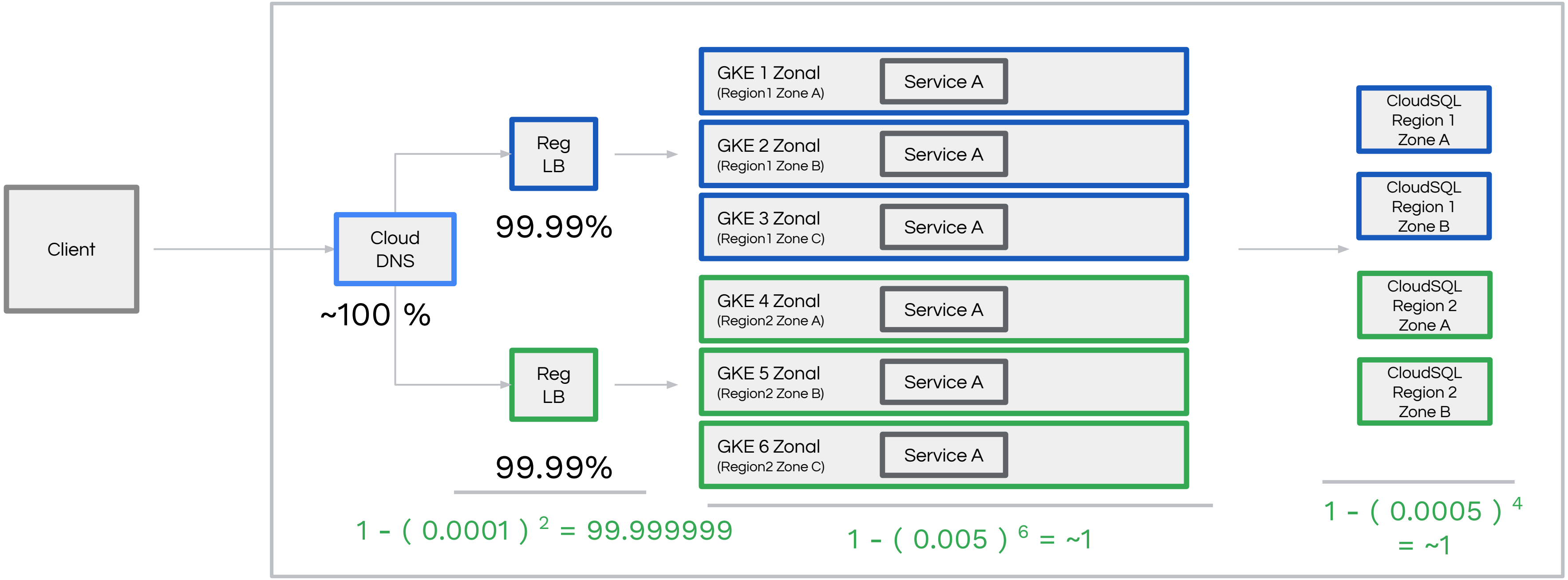
$1 - (0.0005)^2 = 99.999975\%$

$0.9999 \times 0.999975 \times 0.99999975$

99.98% SLO

GCP SLAs

Active passive regions with Cloud SQL HA



Client

Cloud DNS

~100 %

Reg LB

99.99%

Reg LB

99.99%

GKE 1 Zonal (Region1 Zone A) Service A

GKE 2 Zonal (Region1 Zone B) Service A

GKE 3 Zonal (Region1 Zone C) Service A

GKE 4 Zonal (Region2 Zone A) Service A

GKE 5 Zonal (Region2 Zone B) Service A

GKE 6 Zonal (Region2 Zone C) Service A

CloudSQL Region 1 Zone A

CloudSQL Region 1 Zone B

CloudSQL Region 2 Zone A

CloudSQL Region 2 Zone B

$1 - (0.0001)^{2} = 99.999999$

$1 - (0.005)^{6} = \sim 1$

$1 - (0.0005)^{4} = \sim 1$

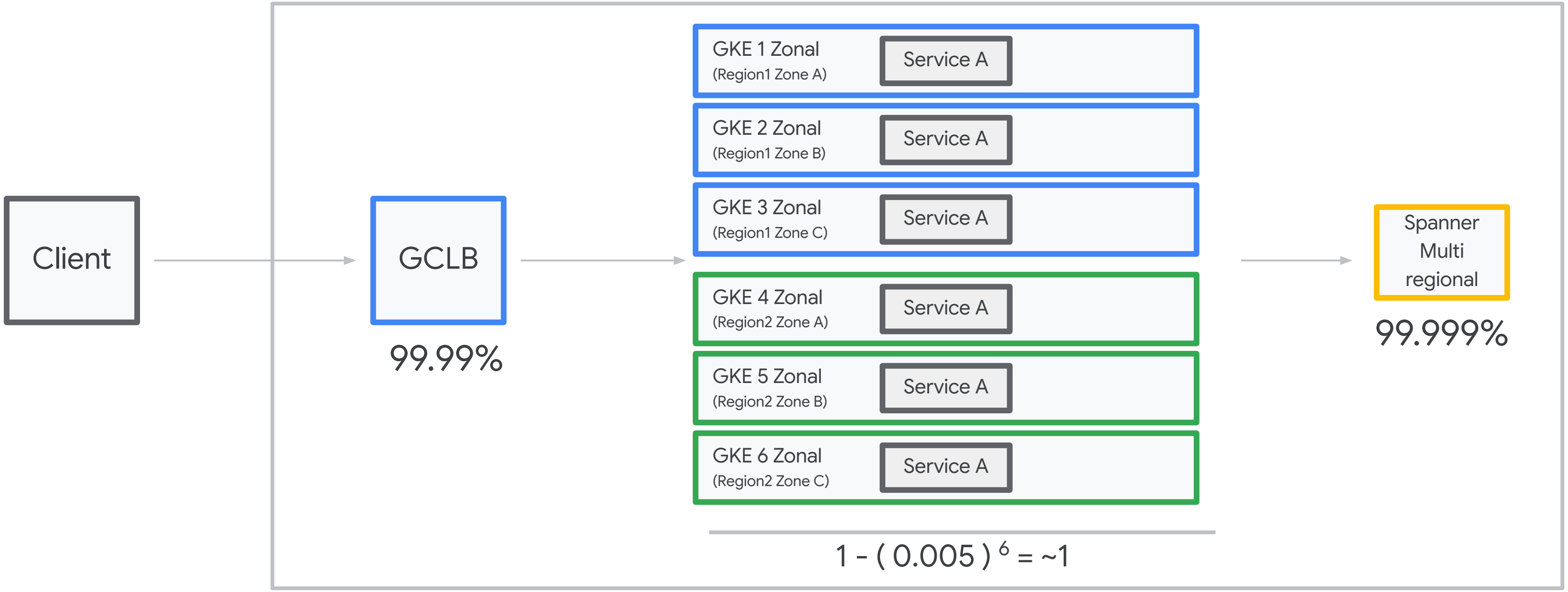
$\sim 1 \times 0.99999999 \times \sim 1 \times \sim 1$

99.999999% SLO

GCP SLAs

Global with Cloud Spanner (multi regional)

Archetype 5.1

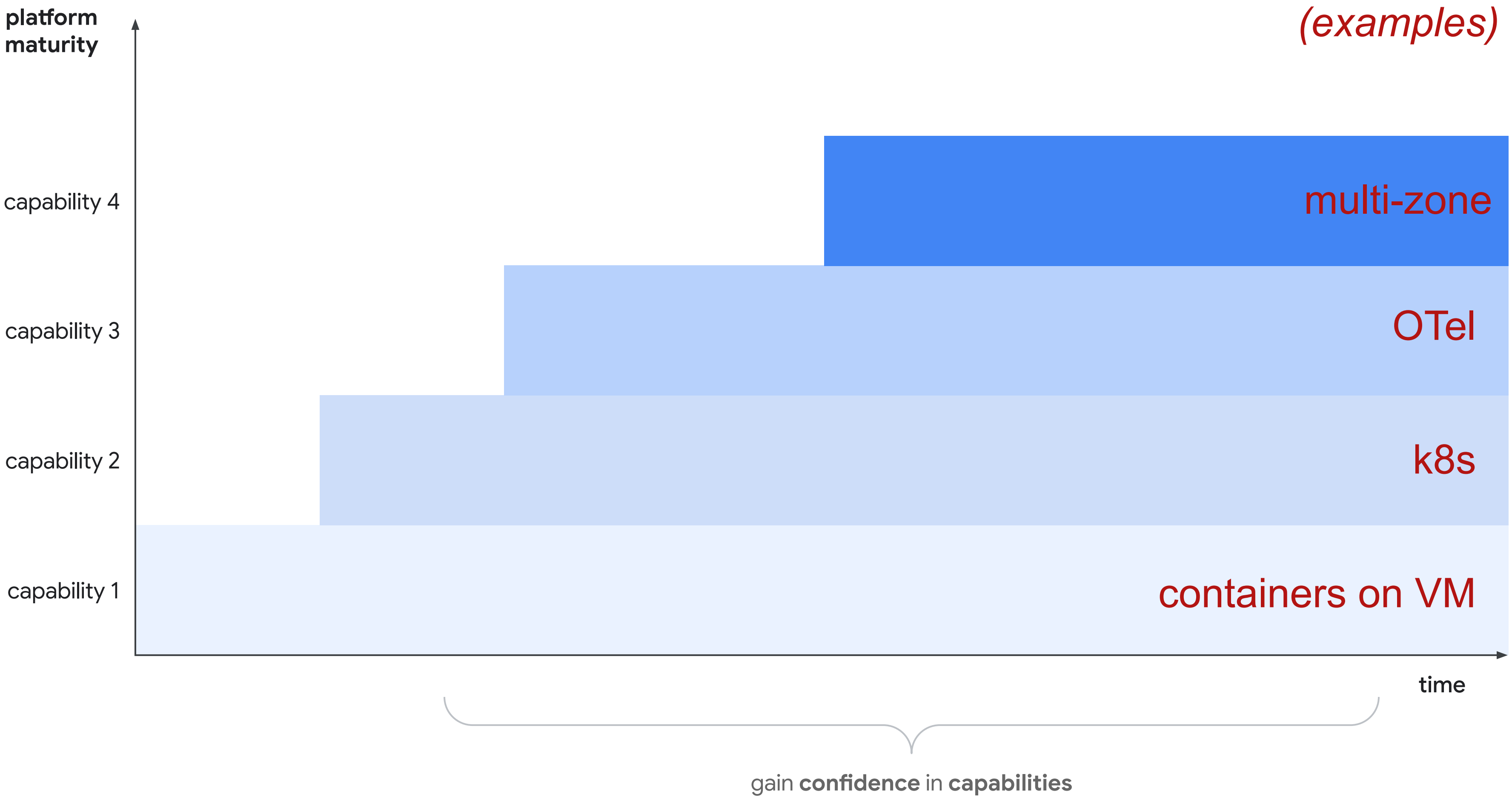


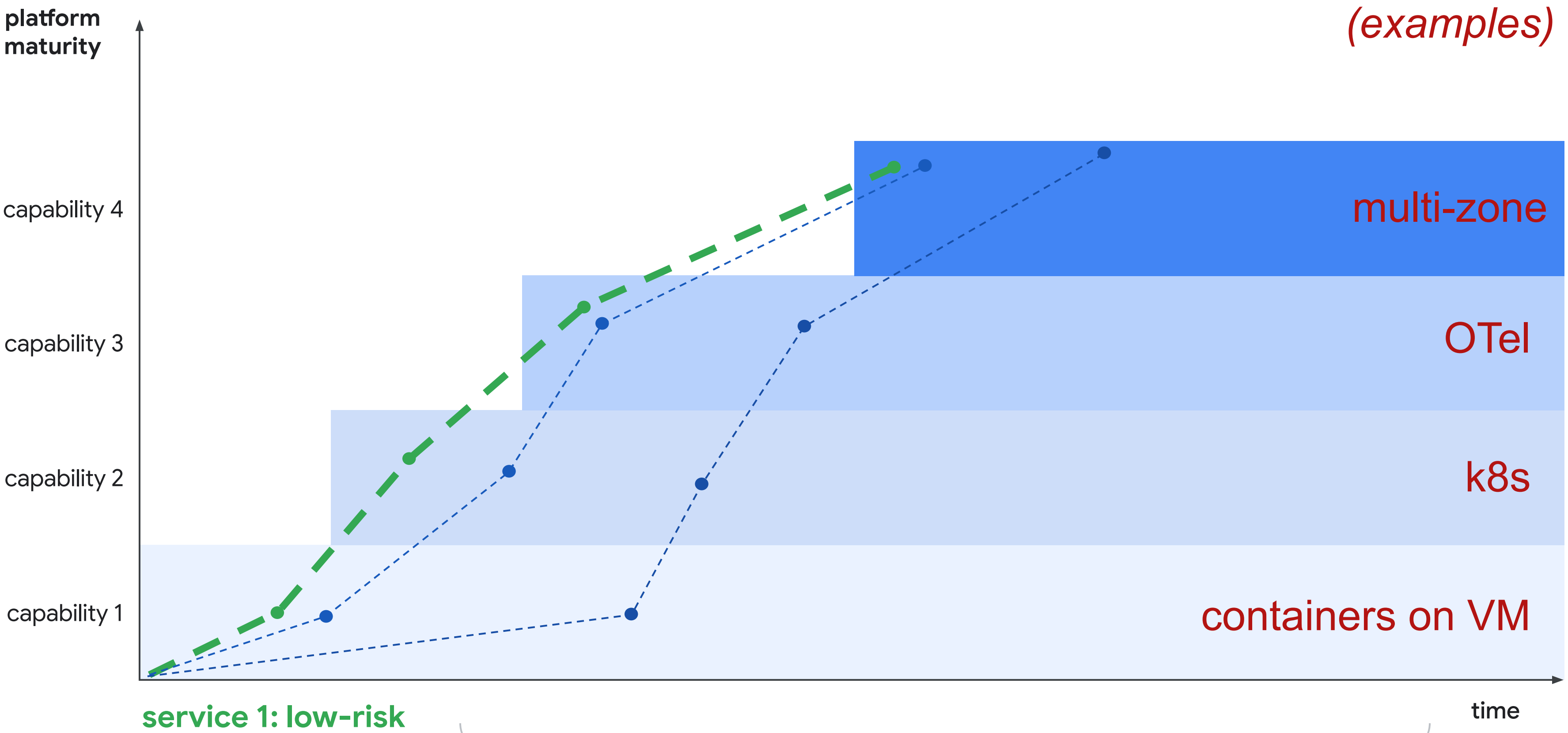
0.9999 x 1 x 0.99999

99.99% SLO

GCP SLAs

Platform capabilities

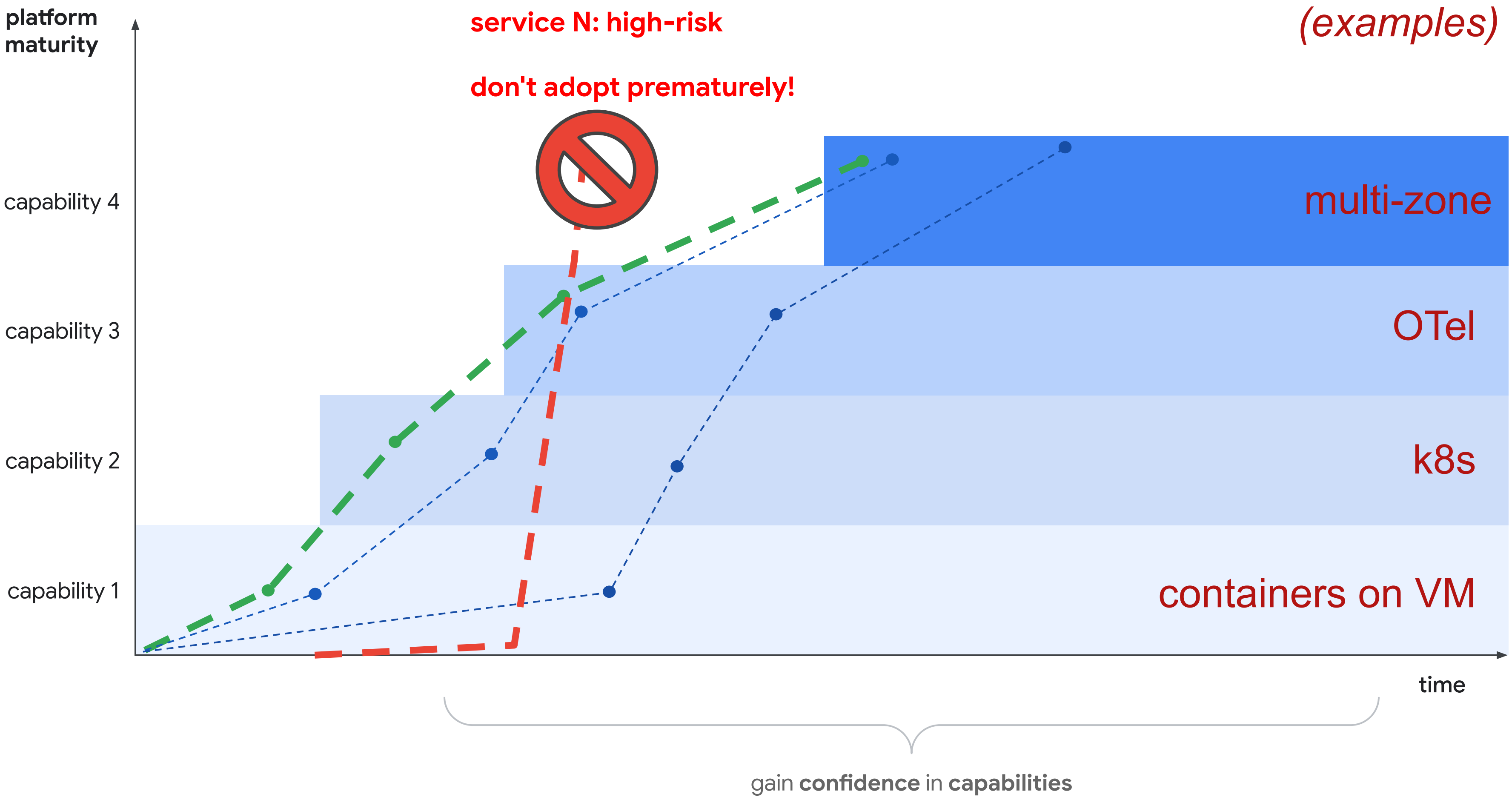

platform maturity
(examples)
capability 4
multi-zone
capability 3
OTel
capability 2
k8s
capability 1
containers on VM
time
gain **confidence** in **capabilities**



service 1: low-risk

early adoption, slow progress

gain **confidence** in **capabilities**



service N: high-risk

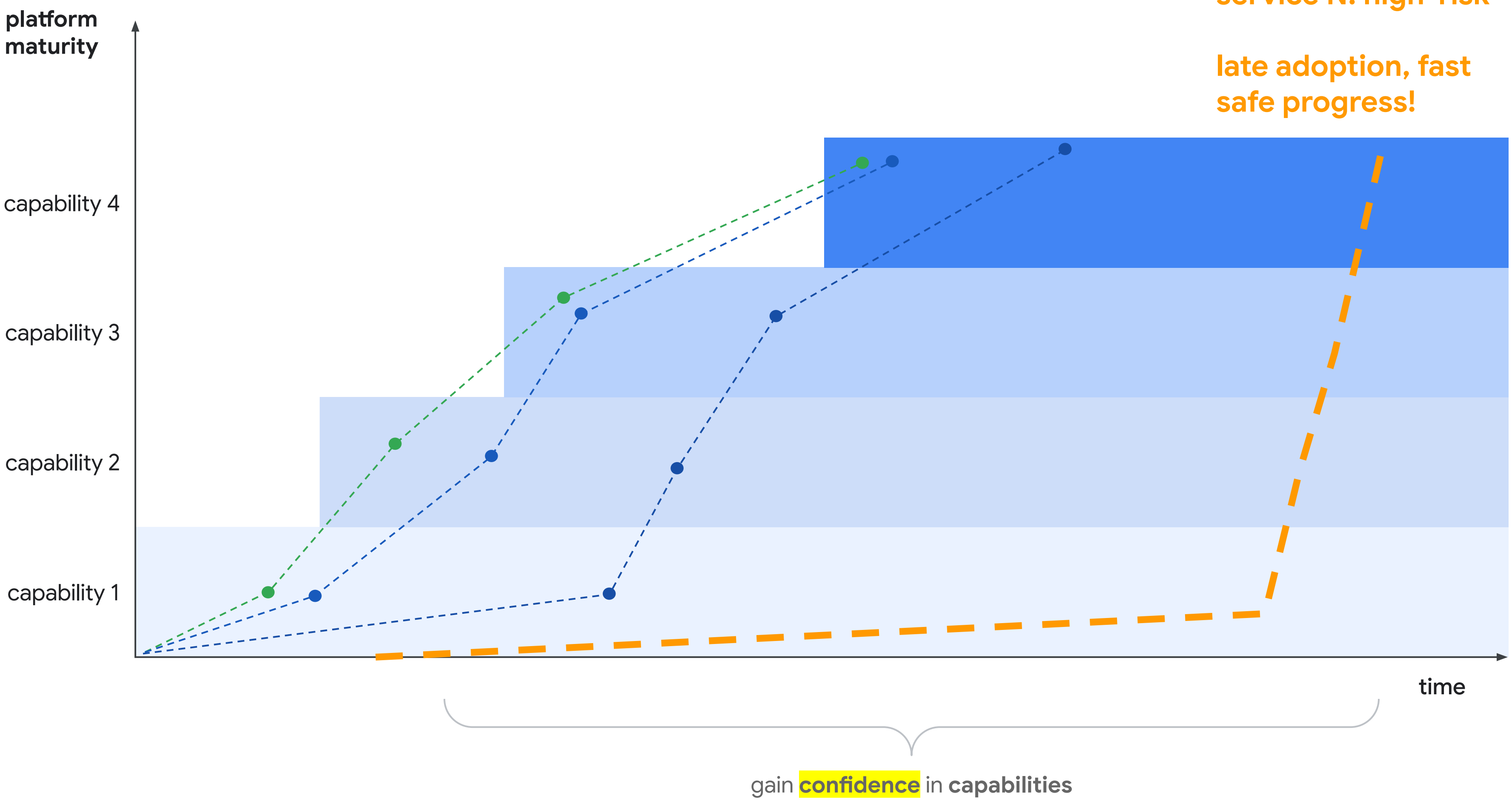
don't adopt prematurely!

(examples)

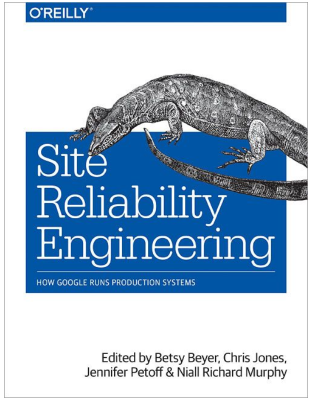
gain **confidence** in **capabilities**

service N: high-risk

late adoption, fast safe progress!



Additional resources

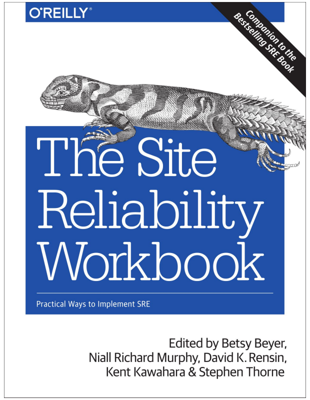


O'REILLY

Site Reliability Engineering

HOW GOOGLE RUNS PRODUCTION SYSTEMS

Edited by Betsy Beyer, Chris Jones, Jennifer Petoff & Niall Richard Murphy



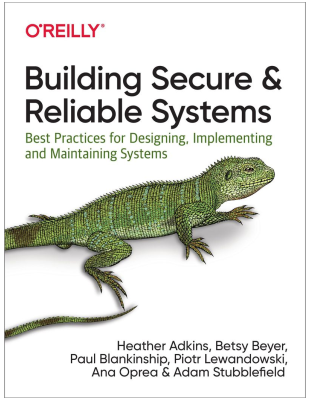
O'REILLY

Companion to the Bestselling SRE Book

The Site Reliability Workbook

Practical Ways to Implement SRE

Edited by Betsy Beyer, Niall Richard Murphy, David K. Rensin, Kent Kawahara & Stephen Thorne



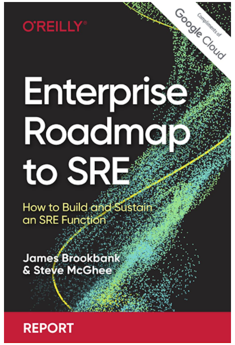
O'REILLY

Building Secure & Reliable Systems

Best Practices for Designing, Implementing and Maintaining Systems

Heather Adkins, Betsy Beyer, Paul Blankinship, Piotr Lewandowski, Ana Oprea & Adam Stubblefield

These books and reports can be found at https://sre.google/



O'REILLY

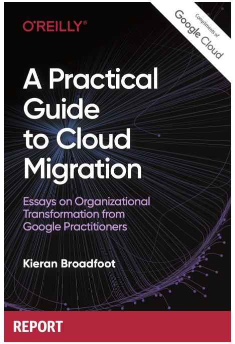
Compliments of Google Cloud

Enterprise Roadmap to SRE

How to Build and Sustain an SRE Function

James Brookbank & Steve McGhee

REPORT



O'REILLY

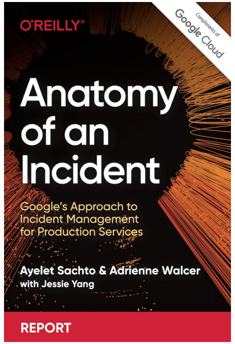
Compliments of Google Cloud

A Practical Guide to Cloud Migration

Essays on Organizational Transformation from Google Practitioners

Kieran Broadfoot

REPORT



O'REILLY

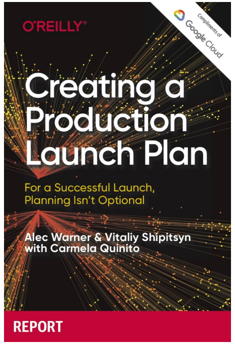
Compliments of Google Cloud

Anatomy of an Incident

Google's Approach to Incident Management for Production Services

Ayelet Sachto & Adrienne Walcer
with Jessie Yang

REPORT



O'REILLY

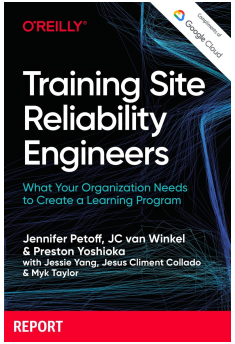
Compliments of Google Cloud

Creating a Production Launch Plan

For a Successful Launch, Planning Isn't Optional

Alec Warner & Vitaliy Shipitsyn
with Carmela Quinito

REPORT



O'REILLY

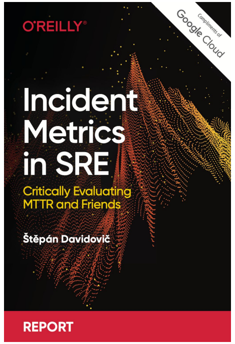
Compliments of Google Cloud

Training Site Reliability Engineers

What Your Organization Needs to Create a Learning Program

Jennifer Petoff, JC van Winkel & Preston Yoshioka
with Jessie Yang, Jesus Climent Collado & Myk Taylor

REPORT



O'REILLY

Compliments of Google Cloud

Incident Metrics in SRE

Critically Evaluating MTTR and Friends

Štěpán Davidovič

REPORT

Next steps

01

Visit sre.google

We have lots of resources available to help organizations get started with SRE practices. The SRE books on the previous slide are ***free*** to read online!

02

Establish SLOs

Developers and business owners should work together to define service-level objectives that can be met most months. Consider starting with one application or major project.

03

Work with Google Cloud

Reach out to your account team. Our SREs from Professional Services can help you stand up an SRE practice.

Thank you!