



SRE @ Loggi

Como rodamos sistemas em produção em larga escala



Outubro 2022 / Privado



Italo Santos
Senior Engineering Manager

[in/in/italosantos/](https://in.linkedin.com/in/italosantos/)





A Loggi está **conectando o Brasil**,
reinventando logística com **tecnologia**

Criando a rede logística do futuro

Aproveitando o poder da tecnologia em todas as etapas



1. Client pickup

- Loggi One mobile app for senders
- Loggi One web app for senders
- API integration for senders
- Driver mobile app



- Independent contractor for pickup
- Loggi Leve: franchisees with dedicated fleets



2. Cross-docking

- XD App mobile app for operators
- Warehouse Management System (WMS)
- Integration with sorters and IOT



- Cross-docking warehouses leased on Loggi' balance sheet
- Operators are mainly employees



3. Mini hub

- XD App mobile app for operators
- Integration with sorters and IOT



- Some urbans hubs are leased and operated directly by Loggi
- Loggi Leve: franchisees own and operate their hubs



4. Last-mile

- Driver mobile app
- Loggi One web app for recipients
- Proprietary route optimization *algos*

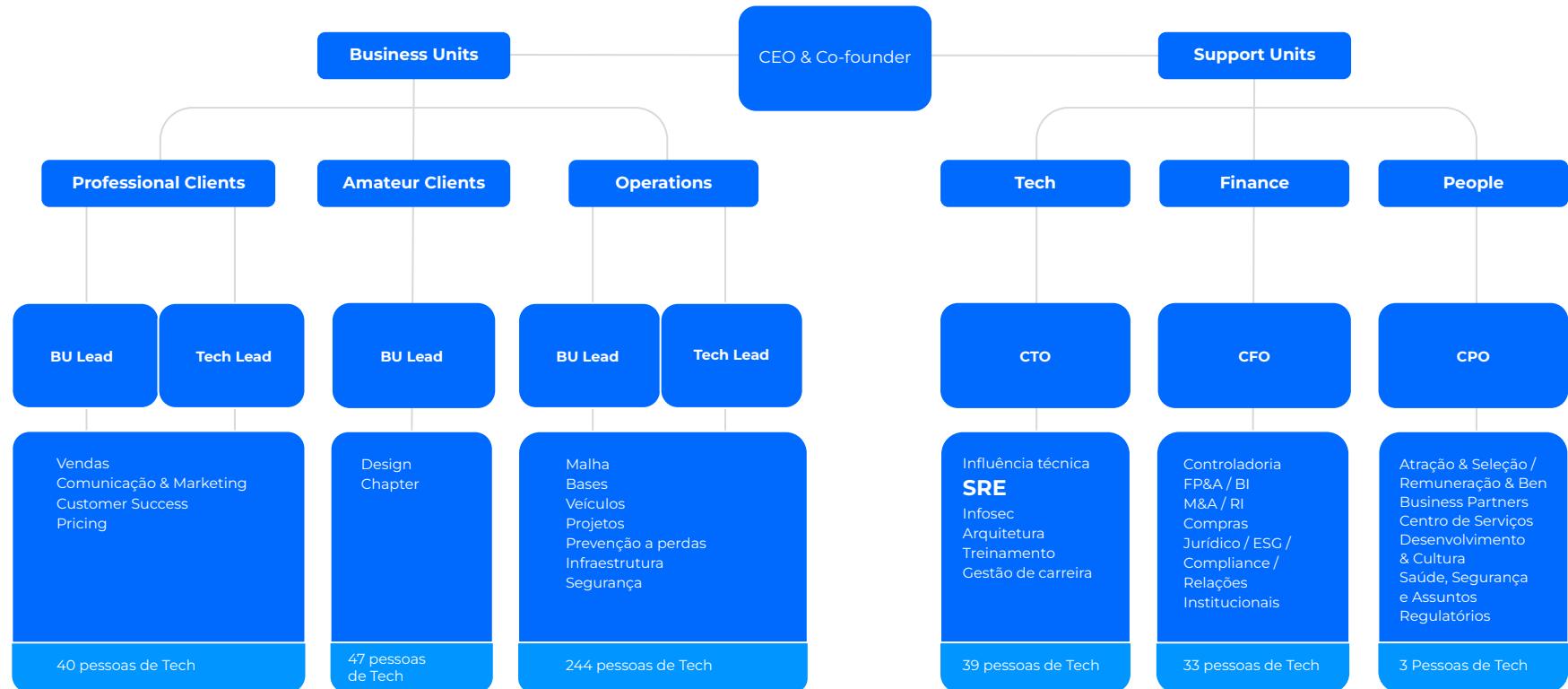


- Independent contractors
- Loggi Leve: franchises with dedicated fleet



Estrutura Organizacional

Como a empresa está estruturada



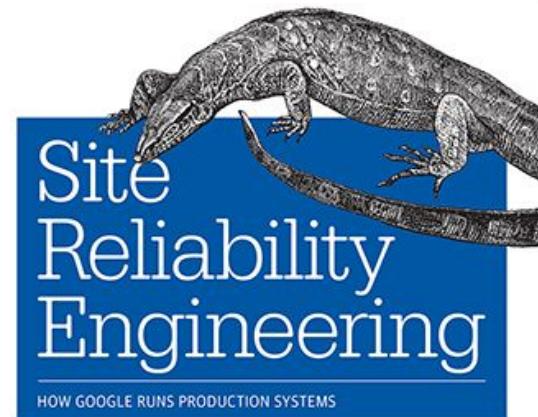


Site Reliability Engineering

Is a set of **principles and practices** that incorporates aspects of software engineering and applies them to infrastructure and operations problems.

The main goals are to create **scalable and highly reliable** software systems.

O'REILLY®



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



SRE

Nature

Construir um conjunto de métodos, métricas e princípios para melhorar a cooperação e a entrega **sem perder a qualidade**

Goal

Minimizar os riscos de negócio

Focus

Melhorar a **disponibilidade e resiliência** dos sistemas

Team

Tem experiência **operacional e de desenvolvimento** sistemas



DevOps

Um conjunto de filosofias que permitem o pensamento cultural e a **colaboração para reduzir os silos da organização**

Colaboração como para **preencher o gap entre desenvolvimento e operação**

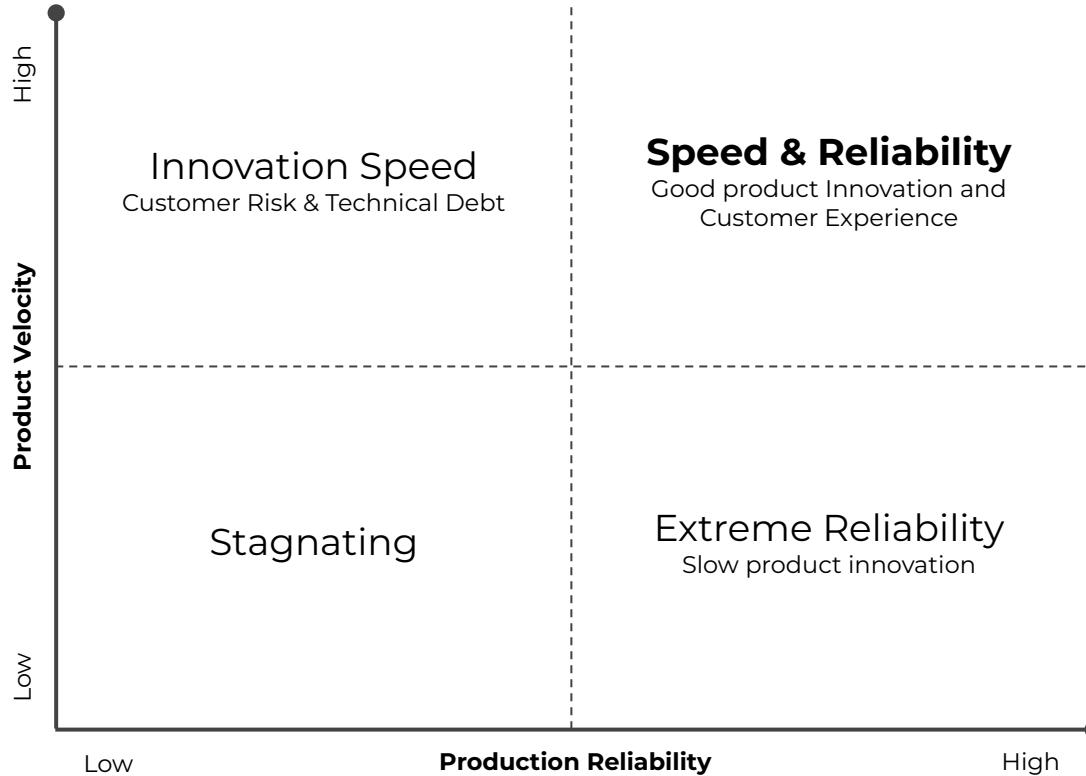
Melhorar a **velocidade de entrega** dos times

Misto de pessoas, incluindo QA, desenvolvedores, SREs, entre outras..



Product Evolution vs Reliability

Speed-Reliability compromise

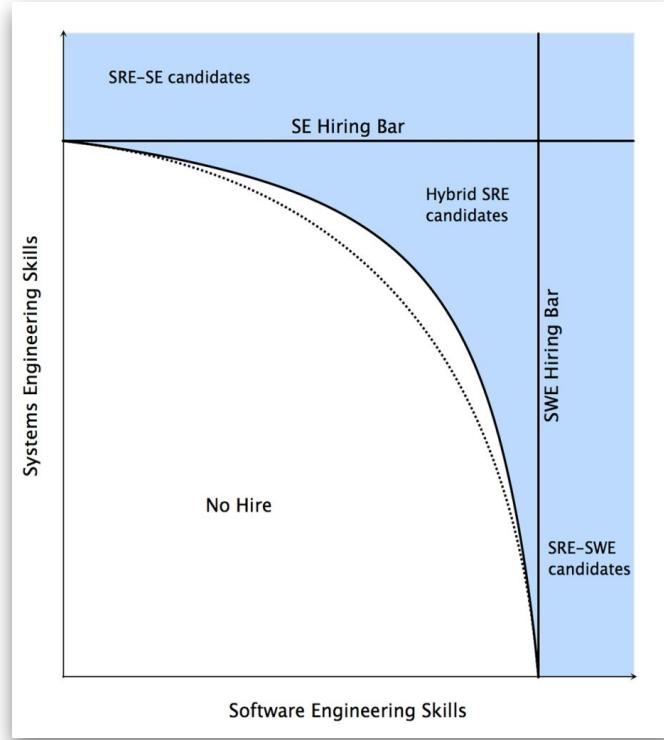




Keys to SRE

Main goals to build SRE culture

- *Hire only coders*
- *Have an SLA for your service*
- *Measure and report performance against SLA*
- *Use Error Budgets and gate launches on them*
- *Common staffing pool for SRE and DEV*
- *Cap SRE operational load at 50%*
- *Share 5% of ops work with DEV team*
- *Oncall teams at latest 8 people or 6x2*
- *Maximum of 2 events per oncall shift*
- *Post Mortem for every event*
- *Post Mortems are blameless and focus on process and technology, not people*



[SYSADMIN - Hiring Site Reliability Engineers](#)



!=





Characteristics

Most important skill for an SRE

Think in **large scale systems**

Problem Solver

Forward thinkers

Data-Driven Debugger

Good **communication** skills

Automation-Oriented

Prioritization sense



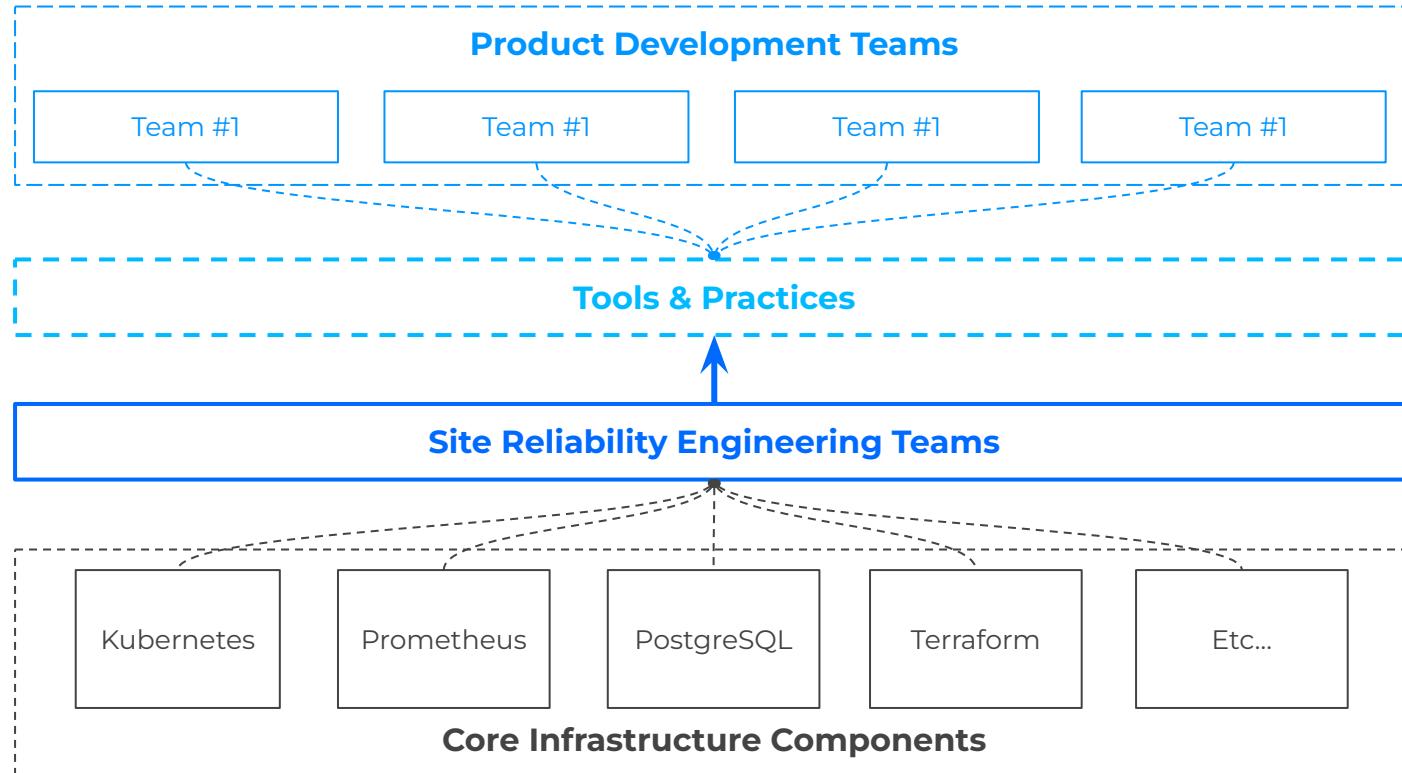
Quickly **adapt**

Always **learning new things**



How are we organized

The interaction with other engineering teams





The Journey...

How we made our way until now

Design a strong infrastructure

Deployment Pipeline & Rotation

Basic monitoring infrastructure

Incident Response

SRE Oncall Rotation

Blameless Postmortems

Foundation

Improve & Growth

Scale



Foundation

How build the base SRE practices

SRE - Postmortem - SRE - Confluence

logidev.atlassian.net/wiki/spaces/DEVOPS/pages/329482250/SRE+-+Postmortem... Share ...

SRE - Postmortem

Introduction

As described in *Site Reliability Engineering*, [Chapter 15 - Postmortem Culture: Learning from Failure](#):

A postmortem is a **written record of an incident**, its impact, the actions taken to mitigate or resolve it, the root cause(s), and the follow-up actions to prevent the incident from recurring.

Proposal

The cost of failure is education - Devin Caraway

As quoted above, the main purpose of a Postmortem is learning, whether it's about the systems being managed, the process being followed, or how the organization runs during a crisis. Additional goals, including identification and implementation of system, or process improvements, may be realized.

The true value of postmortems comes from helping institutionalize a positive culture around frequent and iterative improvement.

Writing process

During incident response, the team is **100% focused on restoring service**. They can't, and should not, be wasting time and energy on thinking about how to do something more optimally, not performing a deep dive on figuring out the root cause of an incident.

The Release Manager model

Published in Loggi

Italo Santos · Sep 29, 2020 · 9 min read · Listen

The Release Manager model

Scaling monolith deployment process with continuous delivery

N owadays the **Continuous Integration** and **Continuous Delivery** (i.e.: CI/CD) practices are widely used by most of the technology companies and the goal of these practices is to speed up product development.

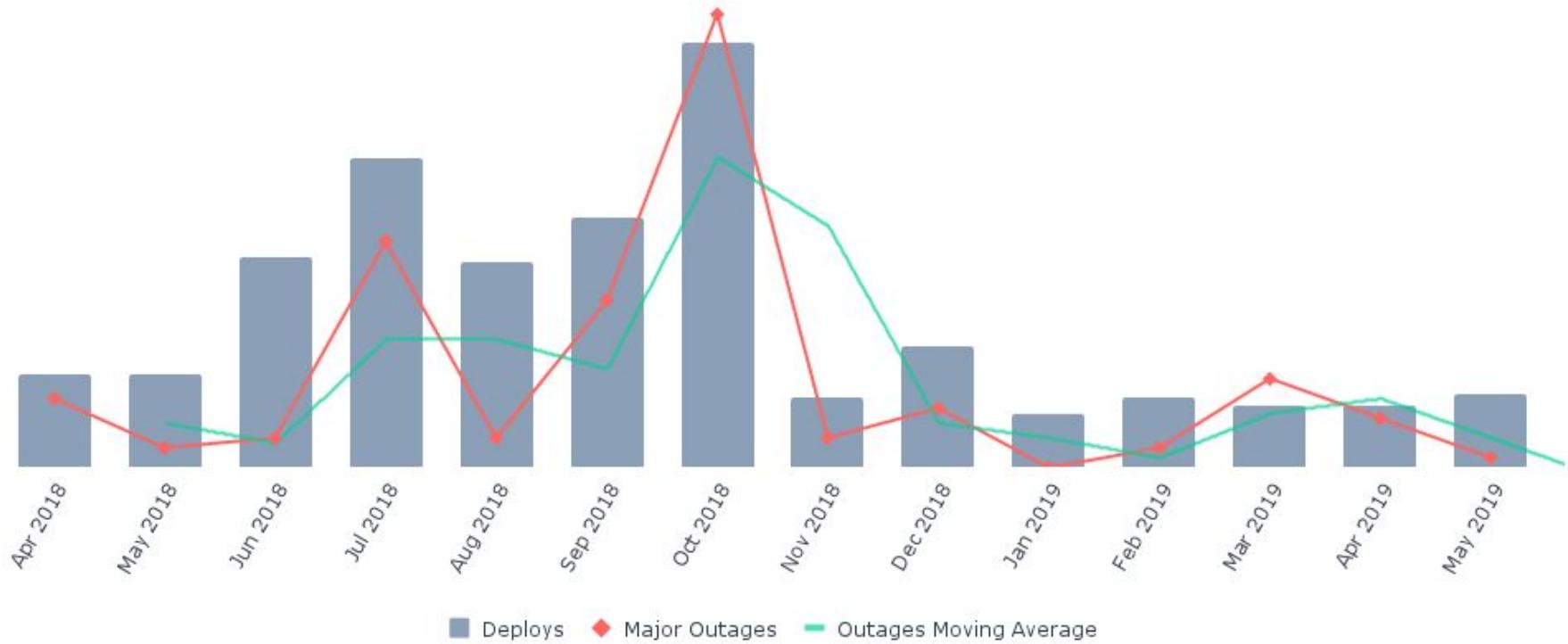
The diagram illustrates a deployment pipeline. It shows multiple parallel branches of code being processed. Each branch starts with a person icon labeled "code", which leads to another person icon labeled "run tests". From there, the path splits into two parallel branches: one leading to a person icon labeled "success" and another leading to a person icon labeled "fail". Both the "success" and "fail" paths then converge into a single path that leads to a large circular node labeled "deploy to production". Arrows indicate the flow from "code" to "run tests", from "run tests" to "success" or "fail", and finally from both "success" and "fail" to "deploy to production".

Deployment pipeline by ([turnoff.us](#))



Foundation

How build the base SRE practices





The Journey...

How we made our way until now

Design a strong infrastructure

Deployment Pipeline & Rotation

Basic monitoring infrastructure

Incident Response

SRE Oncall Rotation

Blameless Postmortems

Rollback-First philosophy

Focus on availability (MTTR, MTTD & MTBF)

Foundation

Improve & Growth

Scale



How we measure Availability

We count as **lack of availability** any period of time where any of the three major Loggi subsystems (Consumer apps, X-docking and Last-mile) **fail for over 30% of the transactions** due to instabilities in our platform.

This is closely related to "code red" events, which are automatically triggered whenever we **lose over 30 minutes of availability** according to the same criteria.





The Journey...

How we made our way until now

Design a strong infrastructure

Deployment Pipeline & Rotation

Basic monitoring infrastructure

Incident Response

SRE Oncall Rotation

Blameless Postmortems

Reliable & Scalable Infrastructure

Improve observability tools

Rollback-First philosophy

Focus on availability (MTTR, MTTD & MTBF)

Foundation

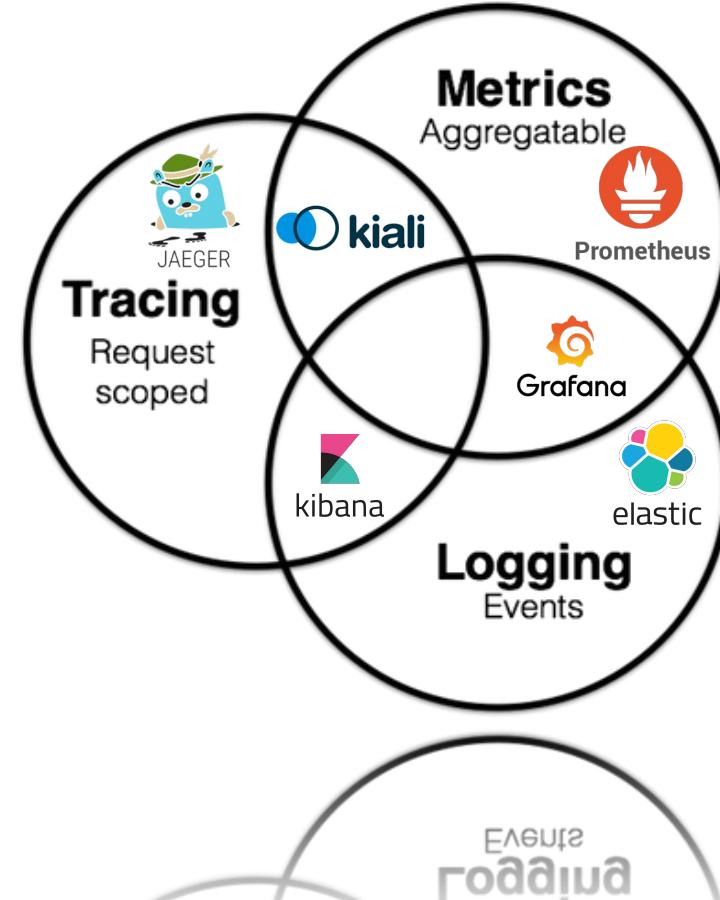
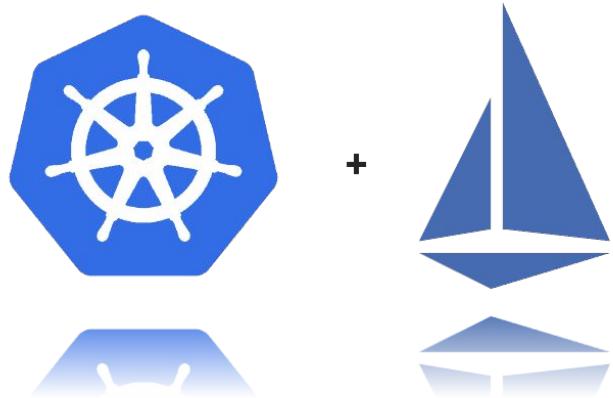
Improve & Growth

Scale



Improve & Growth

The Kubernetes + Istio gives us more scalability and observability, together with Elasticsearch Stack





The Journey...

How we made our way until now

Design a strong infrastructure

Deployment Pipeline & Rotation

Basic monitoring infrastructure

Incident Response

SRE Oncall Rotation

Blameless Postmortems

Canary Deployments

Feature Flags

Reliable & Scalable Infrastructure

Improve observability tools

Rollback-First philosophy

Focus on availability (MTTR, MTTD & MTBF)

Foundation

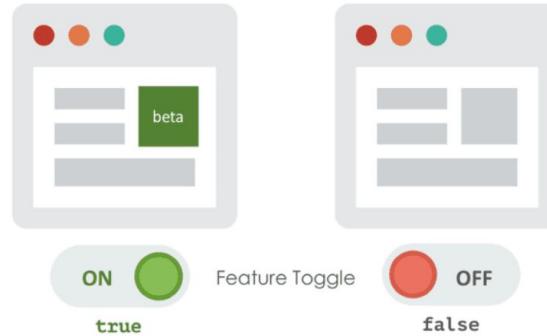
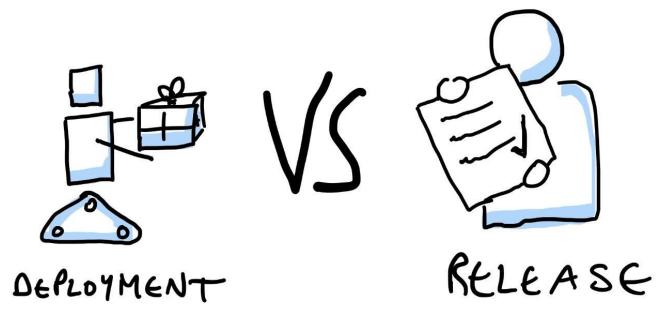
Improve & Growth

Scale



Improve & Growth

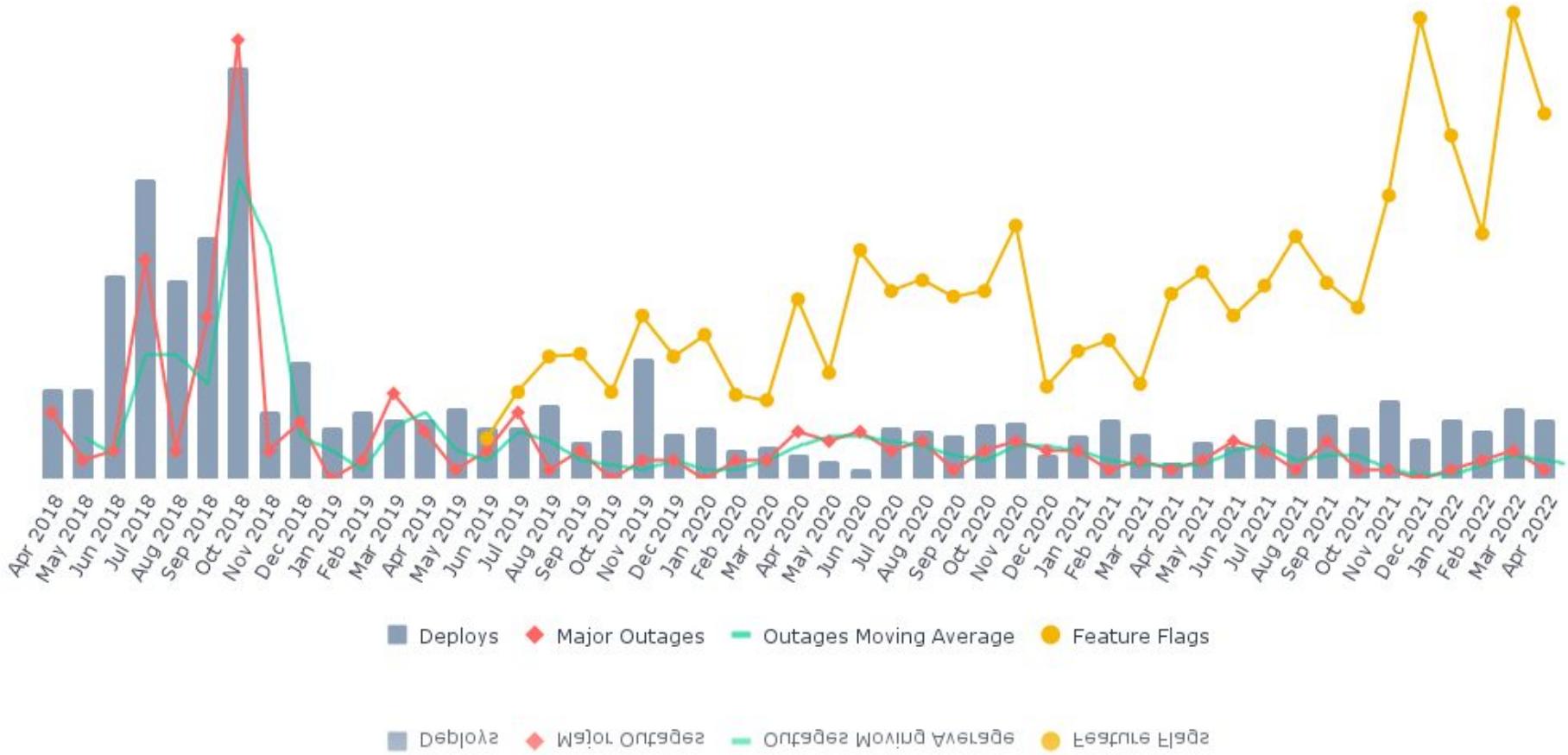
Decouple deploy from release and introduce the feature flags concept





Improve & Growth

We've increased the number of production changes with less deploys





The Journey...

How we made our way until now

Design a strong infrastructure

Deployment Pipeline & Rotation

Basic monitoring infrastructure

Incident Response

SRE Oncall Rotation

Blameless Postmortems

Systems Protection

SLO + Error Budget

Canary Deployments

Feature Flags

Reliable & Scalable Infrastructure

Improve observability tools

Rollback-First philosophy

Focus on availability (MTTR, MTTD & MTBF)

Foundation

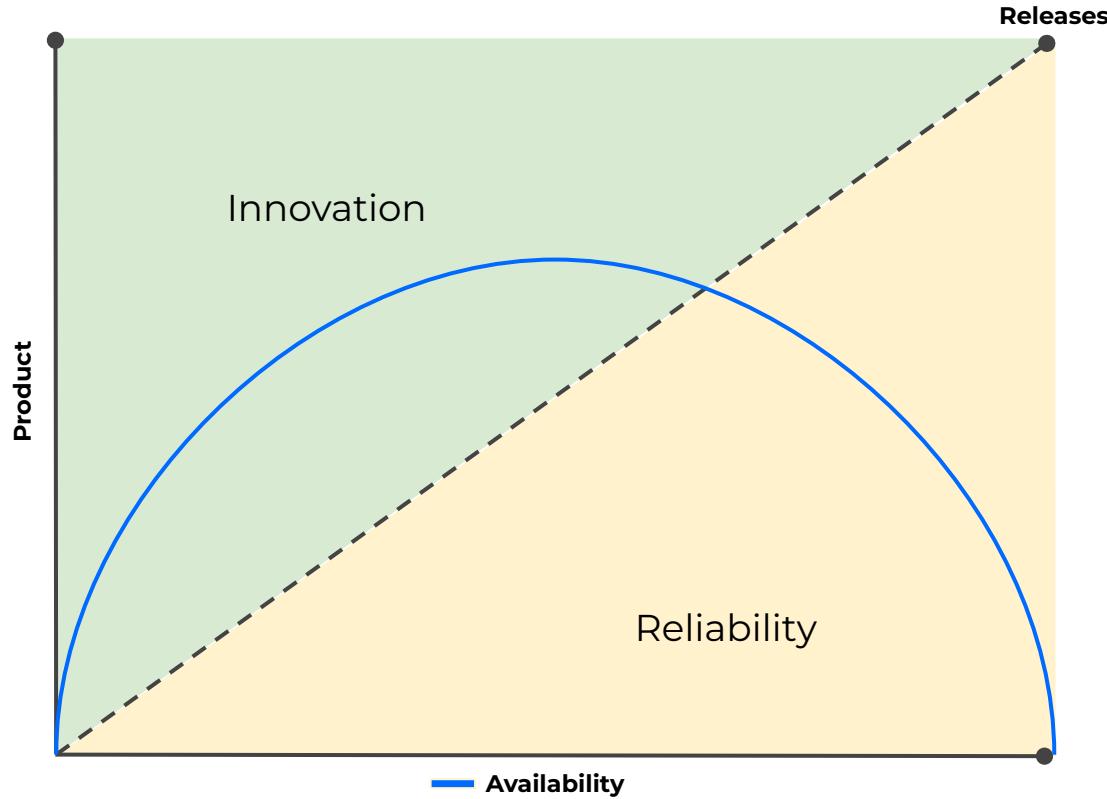
Improve & Growth

Scale



Improve & Growth

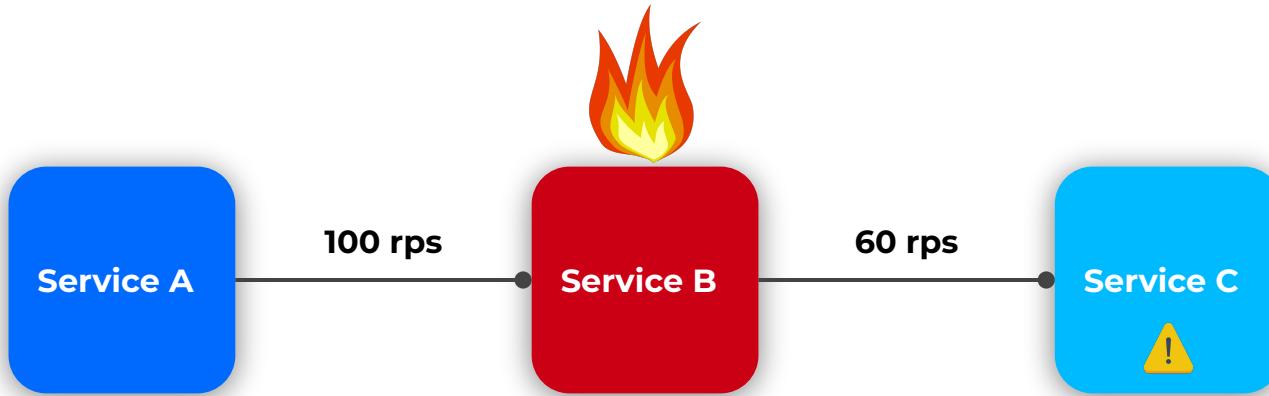
SLO + Error Budget implementation





Improve & Growth

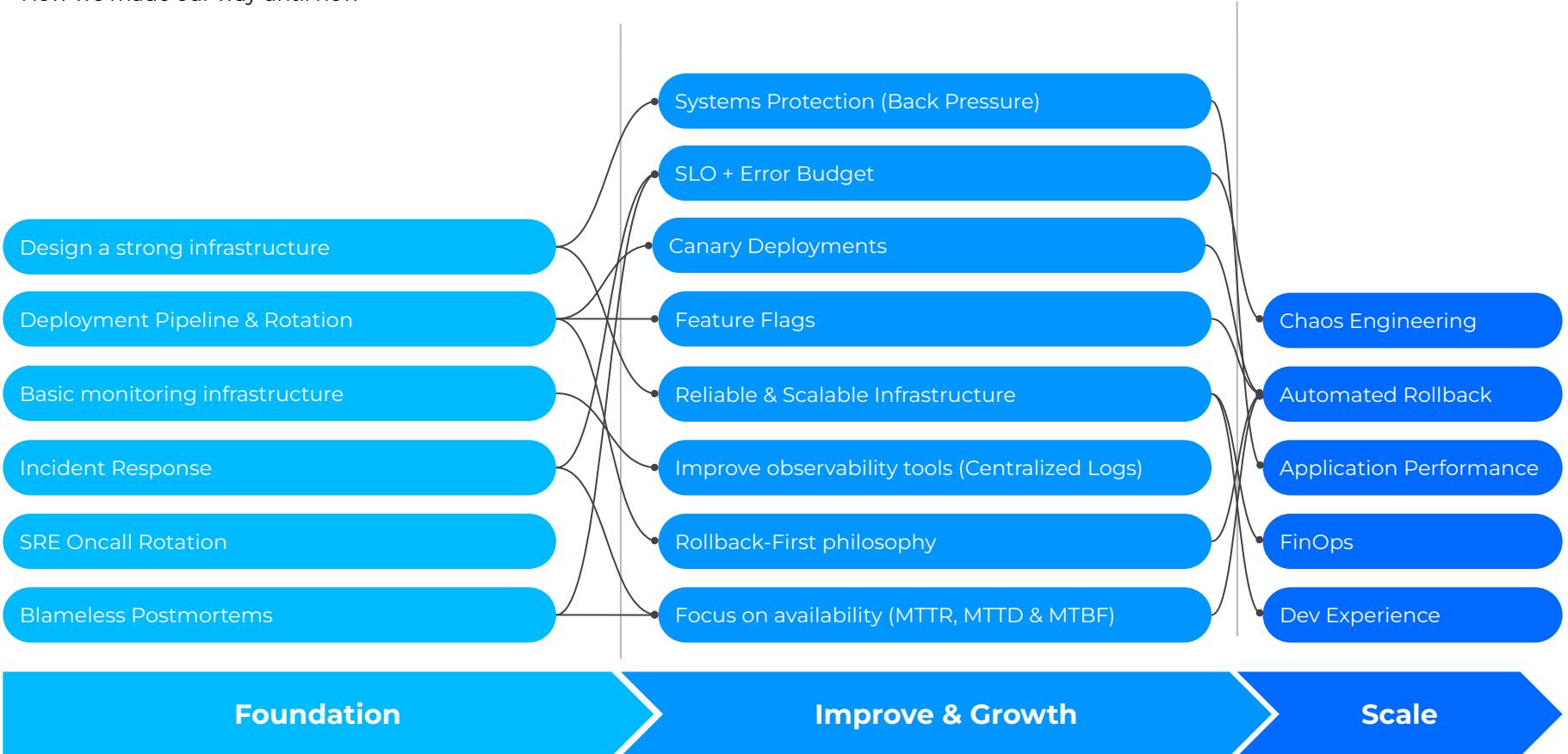
Protecting systems with back pressure





The Journey...

How we made our way until now



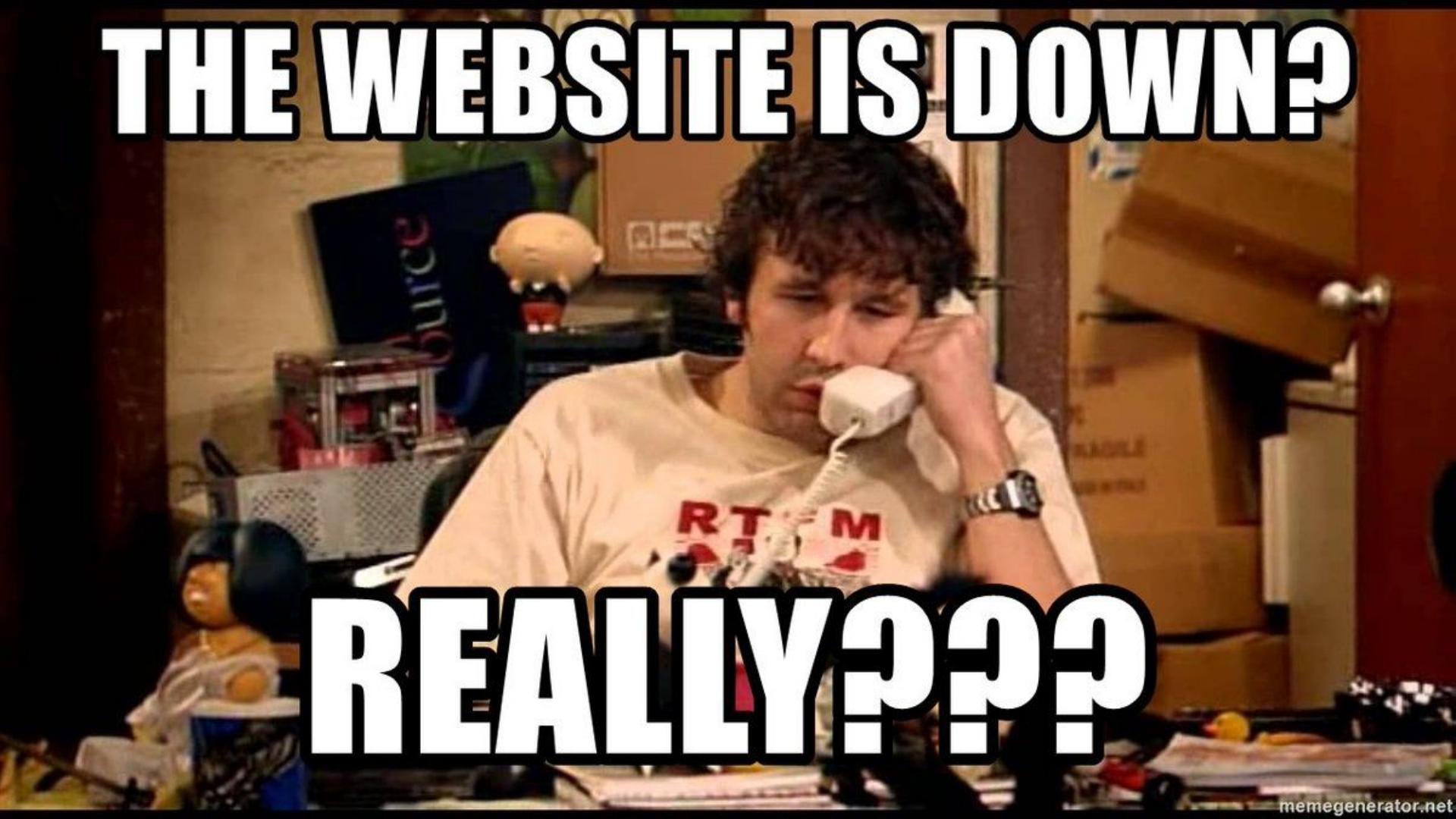
Foundation

Improve & Growth

Scale

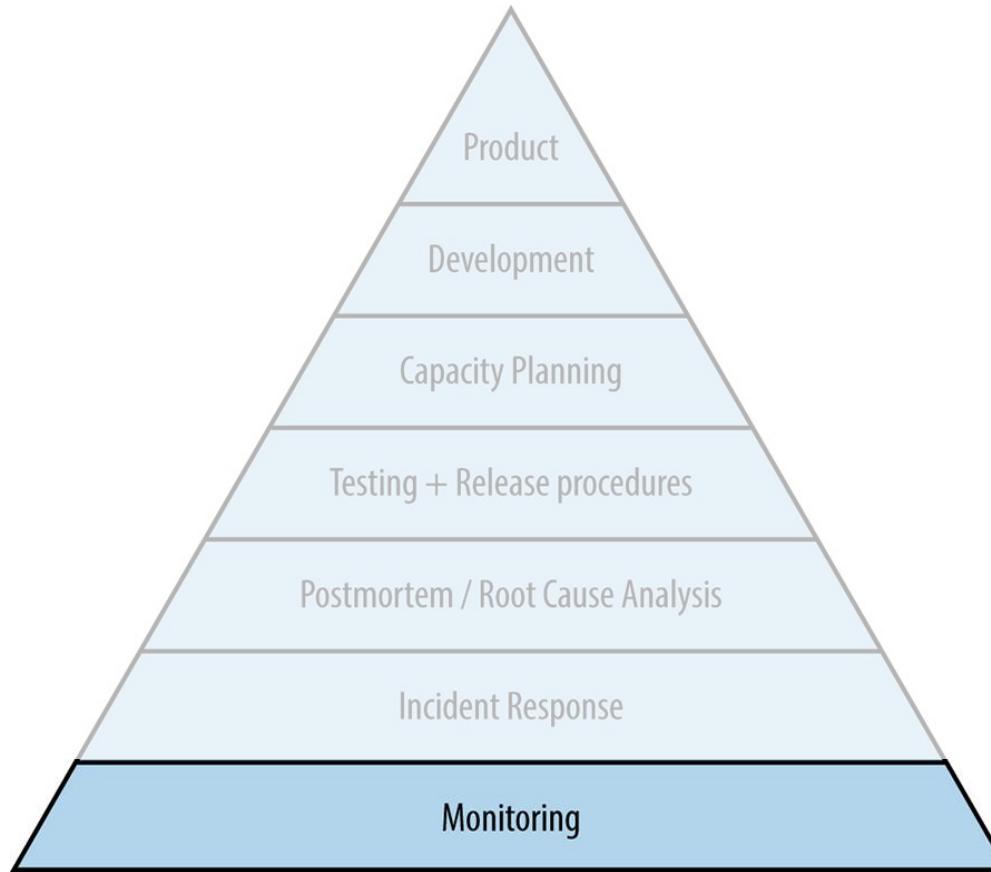
THE WEBSITE IS DOWN?

REALLY???



SRE Fundamentals

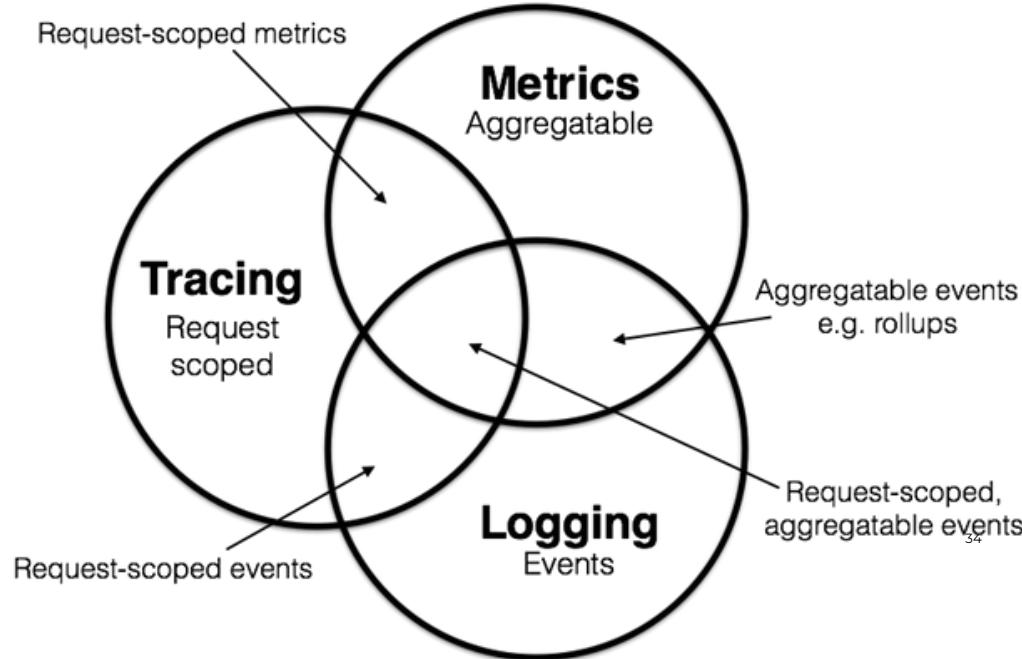
Service Reliability Hierarchy





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

Observability Pillars

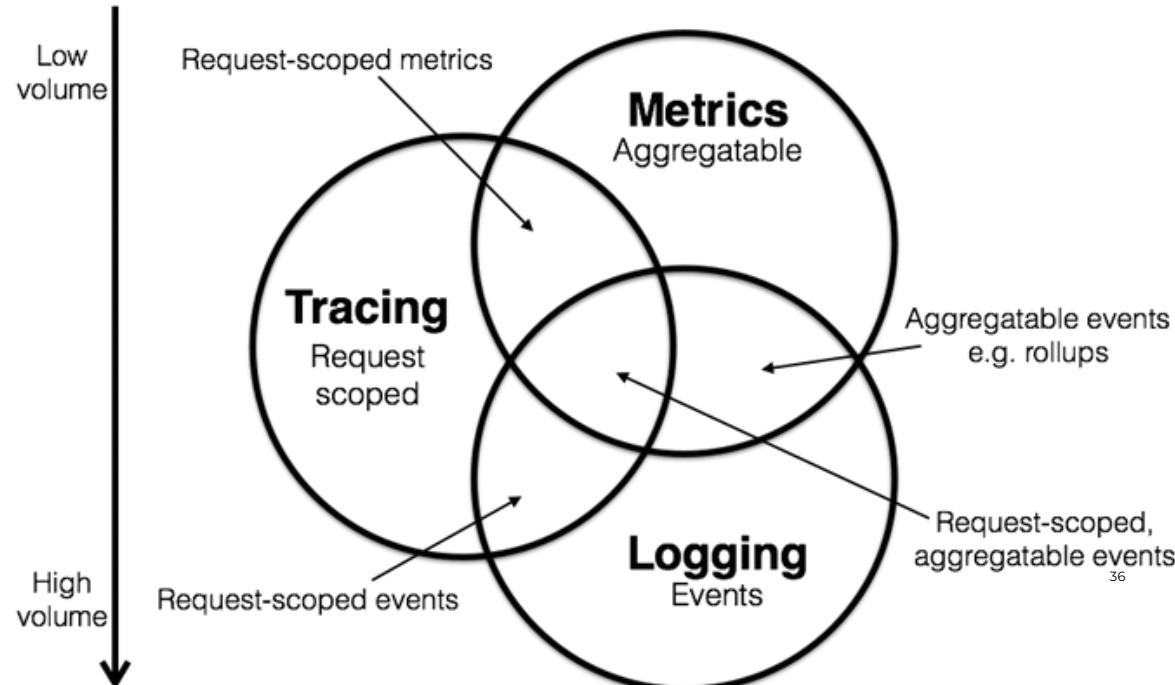




GET



Observability is data!





"Observability Isn't a Panacea"

Observability

- Dynamic
- Unpredictable
- Data

Implement
— — — ^s VS →

Monitoring

- Static
- Predictable
- Metrics



mon-i-tor

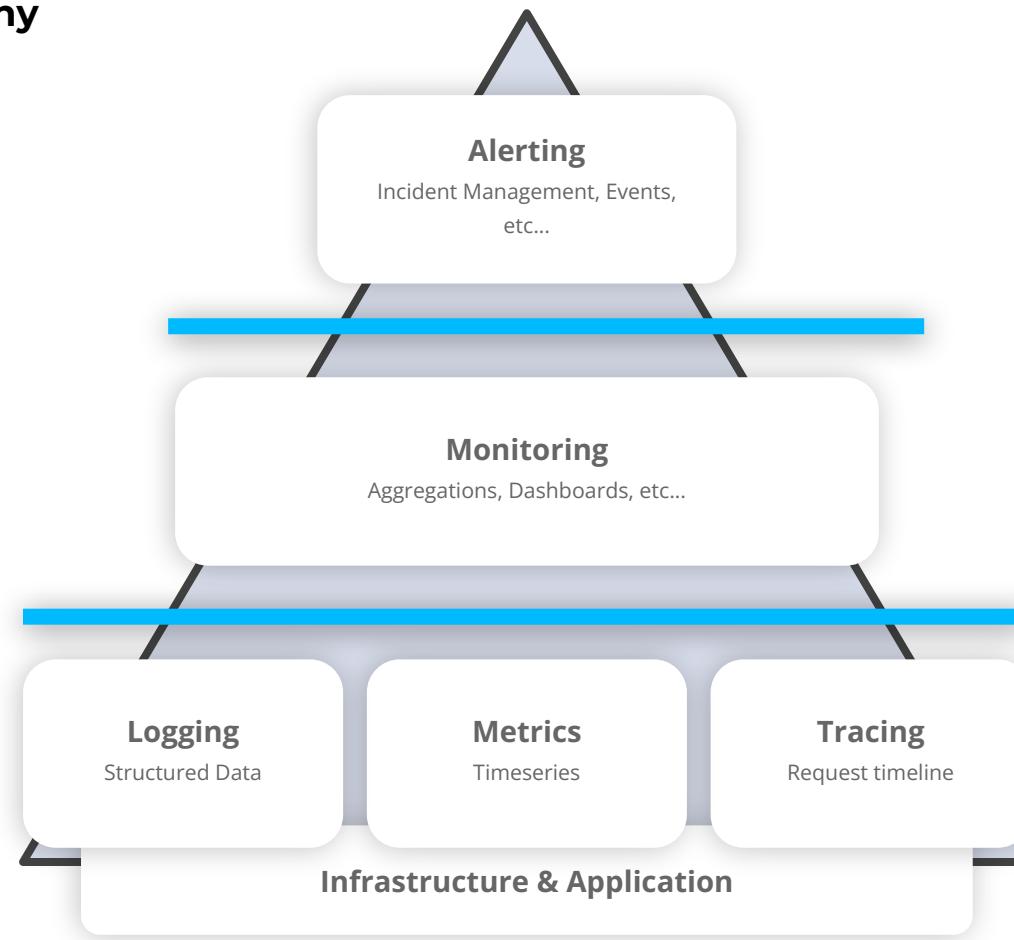
/'mänədər/

verb

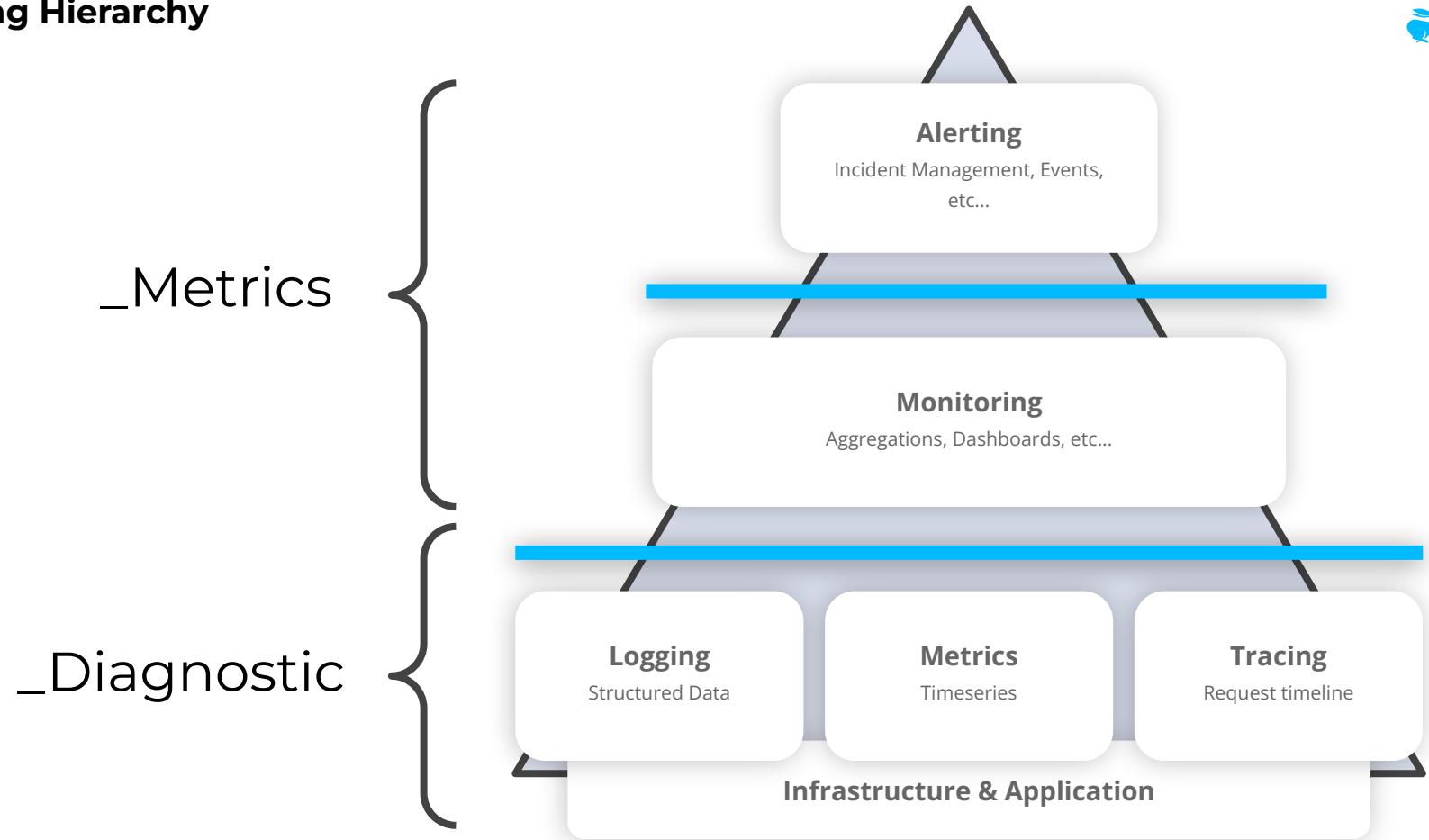
gerund or present participle: monitoring

observe and check the progress or quality of (something) over a period of time;
keep under systematic review.

Monitoring Hierarchy



Monitoring Hierarchy



Metrics

- High Level Overview
- System health & spotting issues
- Look to the present time

vs

Diagnostics

- Log Analysis
- Debugging
- Postmortem & Investigation

Symptom

what?

- High HTTP errors
- System is slow

VS

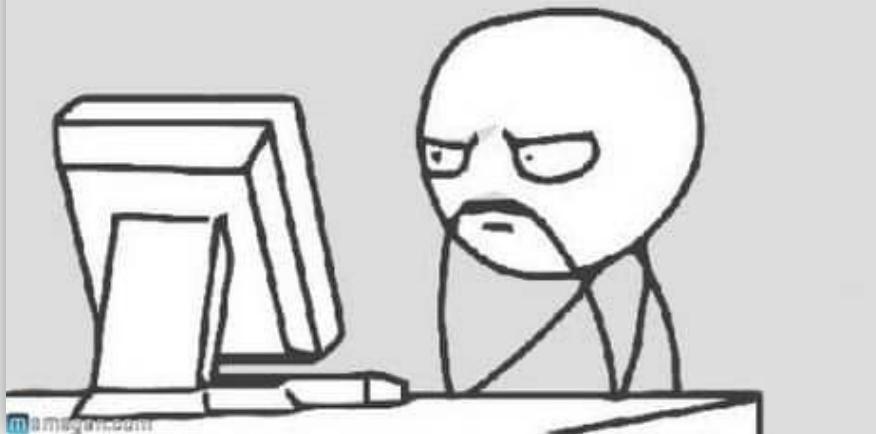
Cause

why?

- Database is down
- Network congestion



HOW DOES IT WORK!!



© 2014 mzlnc.com



What to Monitoring?

Good set of metrics used for monitoring purposes

The Four Golden Signals

Latency, Traffic, Error & Saturation

- Request time
- Total number of requests
- Failed requests
- Service busy

The RED method

Request, Error & Duration

- Requests per seconds
- Failed requests
- Distribution time of requests

The USE method

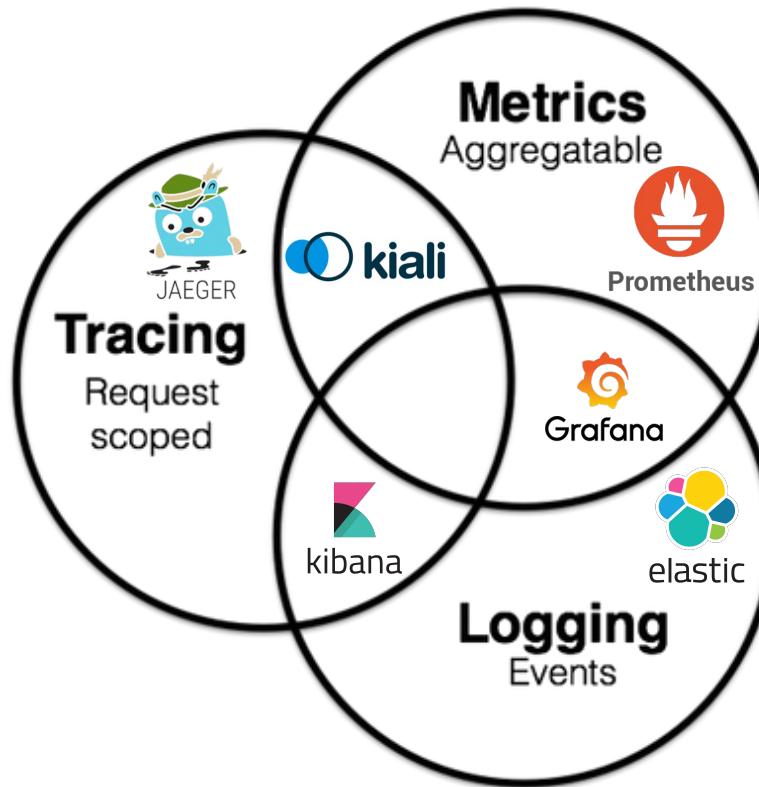
Utilization, Saturation & Error

- Resources (CPU, disks, etc...)
- Percent utilization of resource
- Overload / Queued resource
- Count of error events



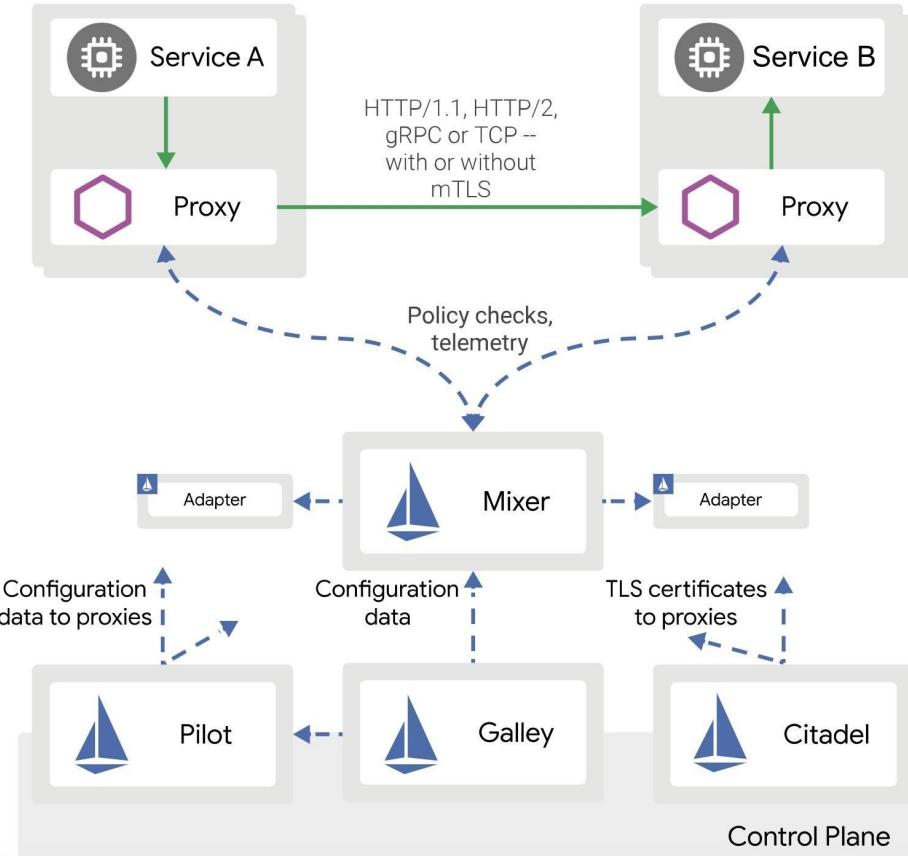
Observability Ecosystem

What tools we use



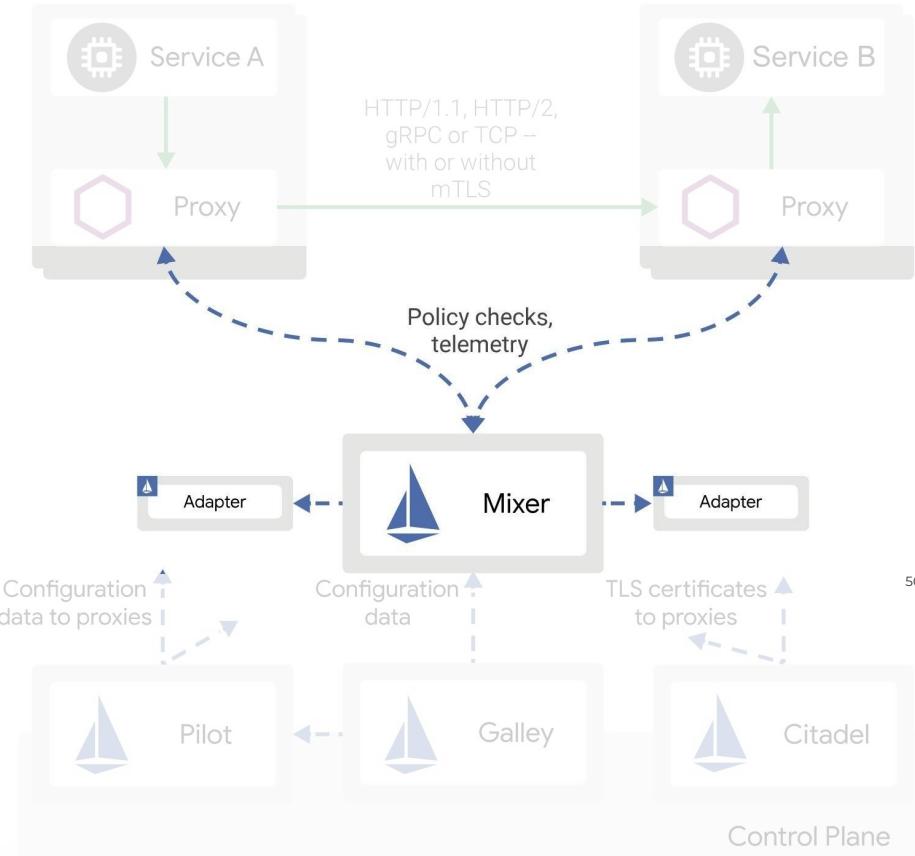
Istio & Observability

Built-in observability infrastructure



Istio & Observability

Built-in observability infrastructure



Metrics

Beyond the infrastructure metrics



Istio Metrics

- Default Metrics
 - HTTP & HTTP/2
 - GRPC traffic
 - TCP
- Service Labels
- Custom Metrics
- Observability tools
 - Kiali
 - Jaeger

Micronaut Micrometer

- Prometheus Support
 - JVM metrics
 - Web metrics
 - Uptime
- Custom metrics

Django Prometheus

- Prometheus Support
 - HTTP metrics
 - Database metrics
 - Models (migrations)
 - Cache metrics
- Custom metrics



Loggi

Thank you



loggi.com