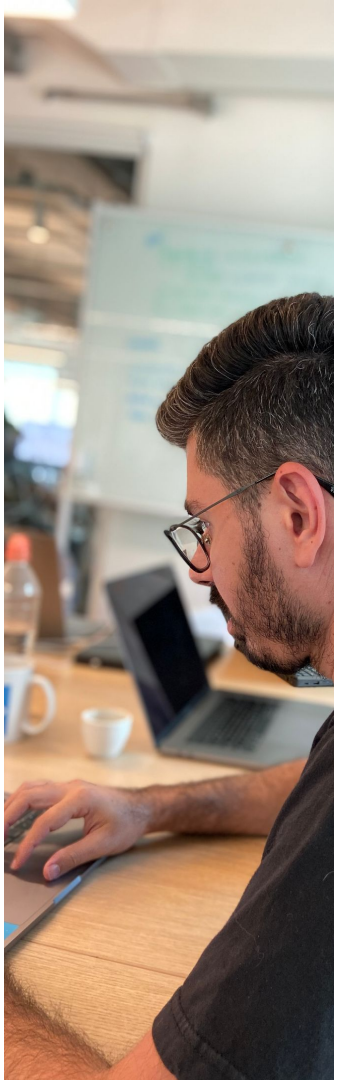
SRE @ Loggi

Como rodamos sistemas em produção em larga escala

Outubro 2022 / Privado

Italo Santos
Senior Engineering Manager

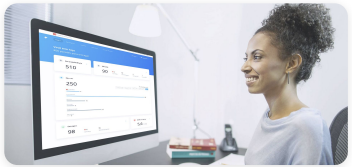
/in/italosantos/



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

Tech Stack

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

Asset Intensity

3

LOW HIGH

- 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

9

LOW HIGH

- 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

6

LOW HIGH

- 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*

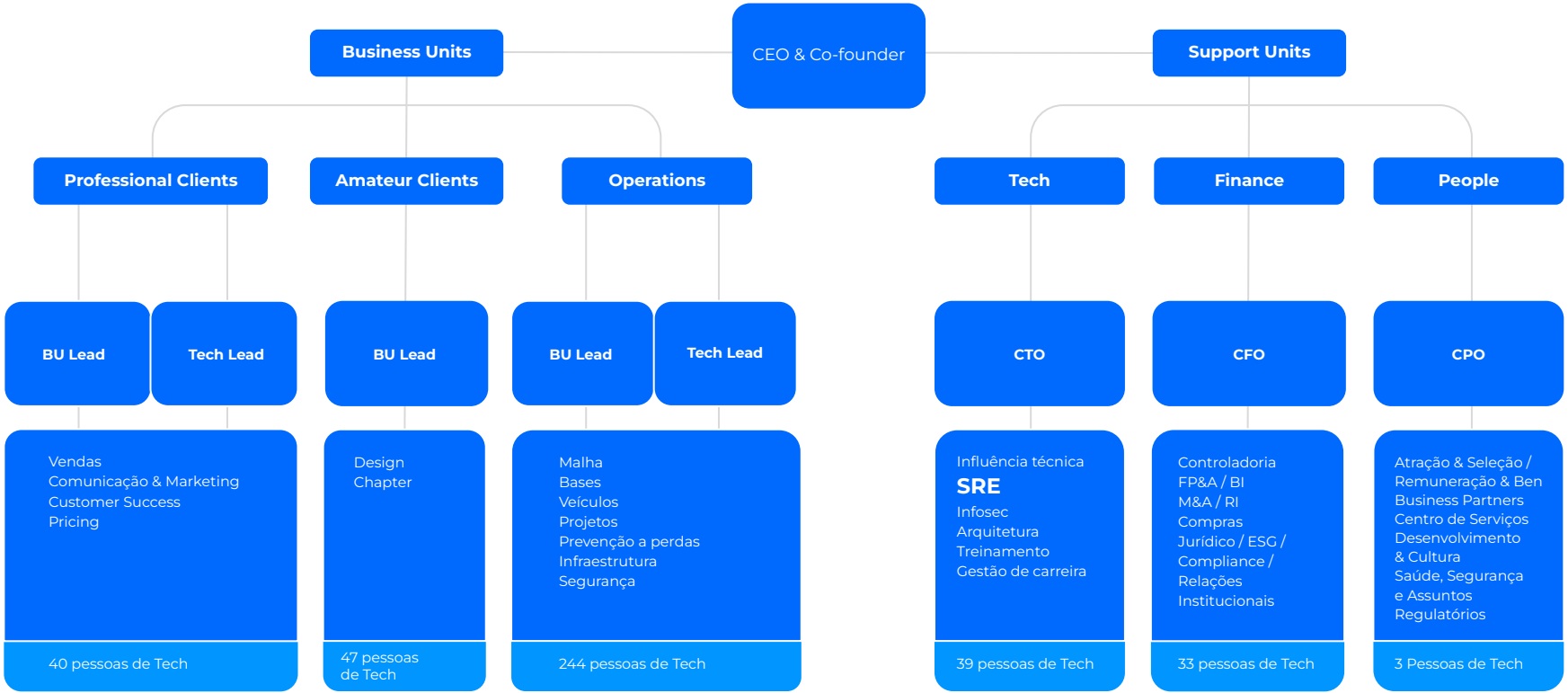
2

LOW HIGH

- Independent contractors
- Loggi Leve: franchises with dedicated fleet

Estrutura Organizacional

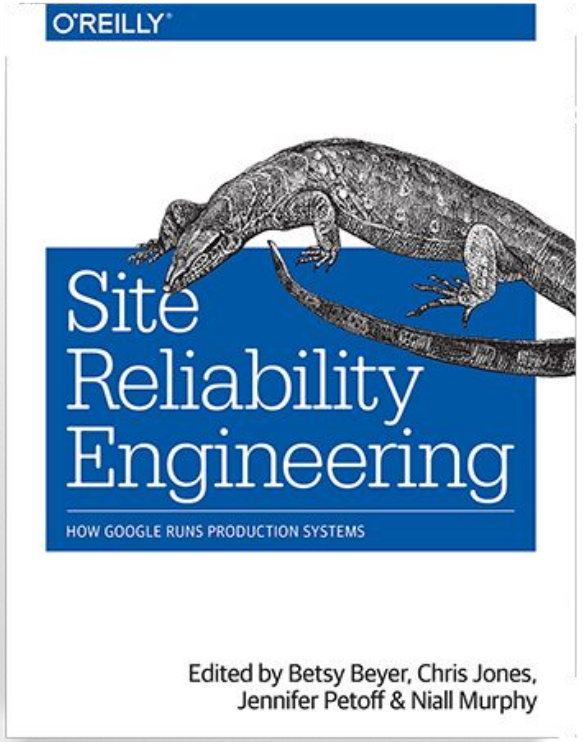
Como a empresa está estruturada



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.*

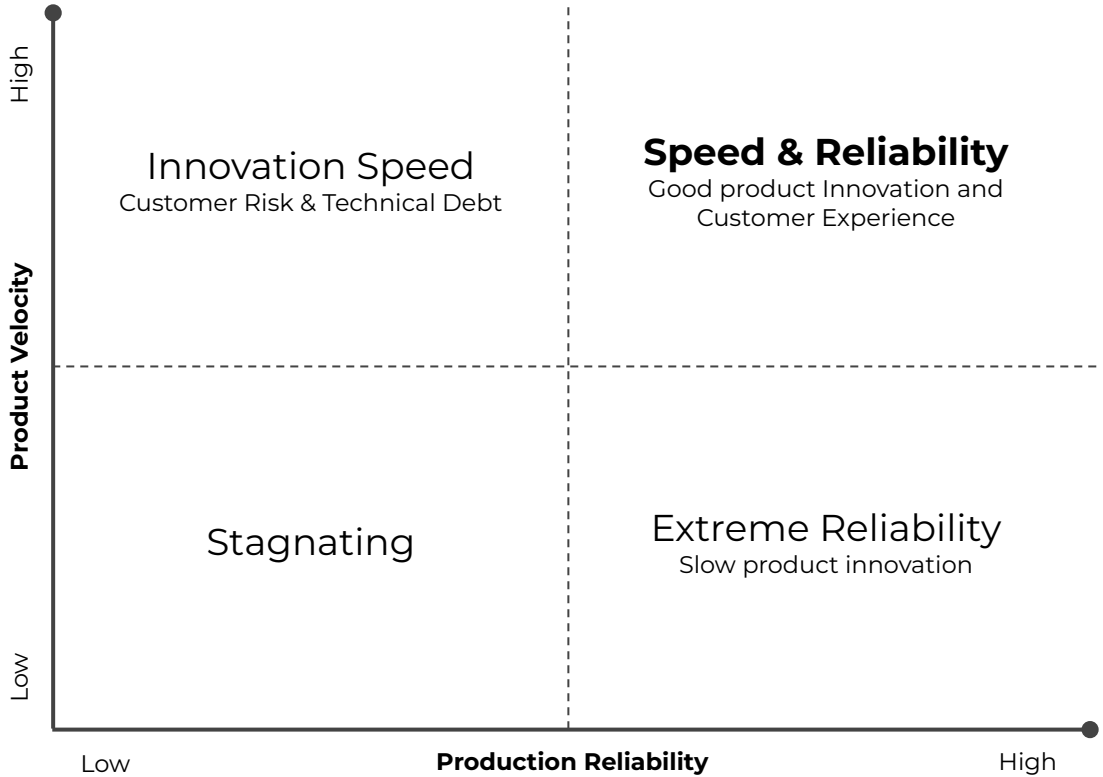


Site reliability engineering - Wikipedia

	SRE	DevOps
Nature	Construir um conjunto de métodos, métricas e princípios para melhorar a cooperação e a entrega **sem perder a qualidade**	Um conjunto de filosofias que permitem o pensamento cultural e a **colaboração para reduzir os silos da organização**
Goal	**Minimizar os riscos de negócio**	Colaboração como para **preencher o gap entre desenvolvimento e operação**
Focus	Melhorar a **disponibilidade e resiliência** dos sistemas	Melhorar a **velocidade de entrega** dos times
Team	Tem experiência **operacional e de desenvolvendo** sistemas	**Misto de pessoas,** incluindo QA, desenvolvedores, SREs, entre outras..

Product Evolution vs Reliability

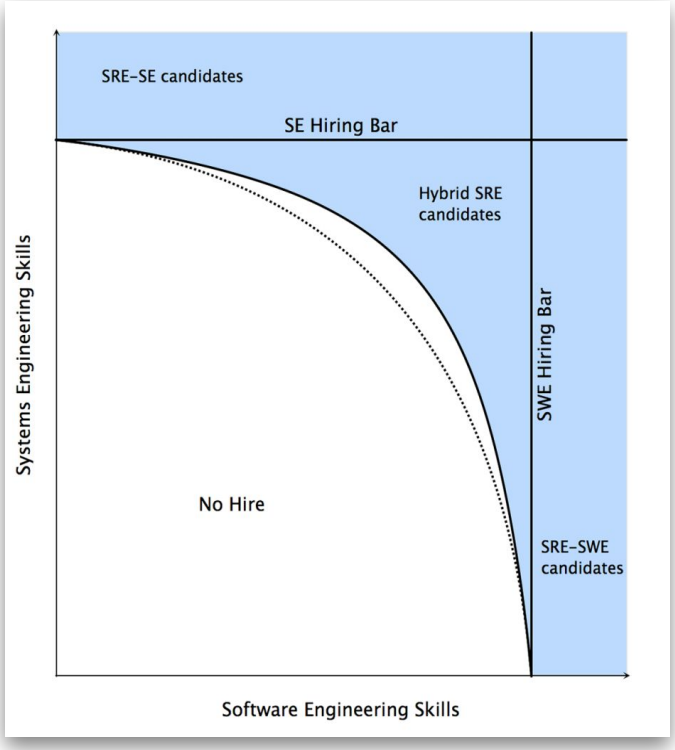
Speed-Reliability compromise



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*



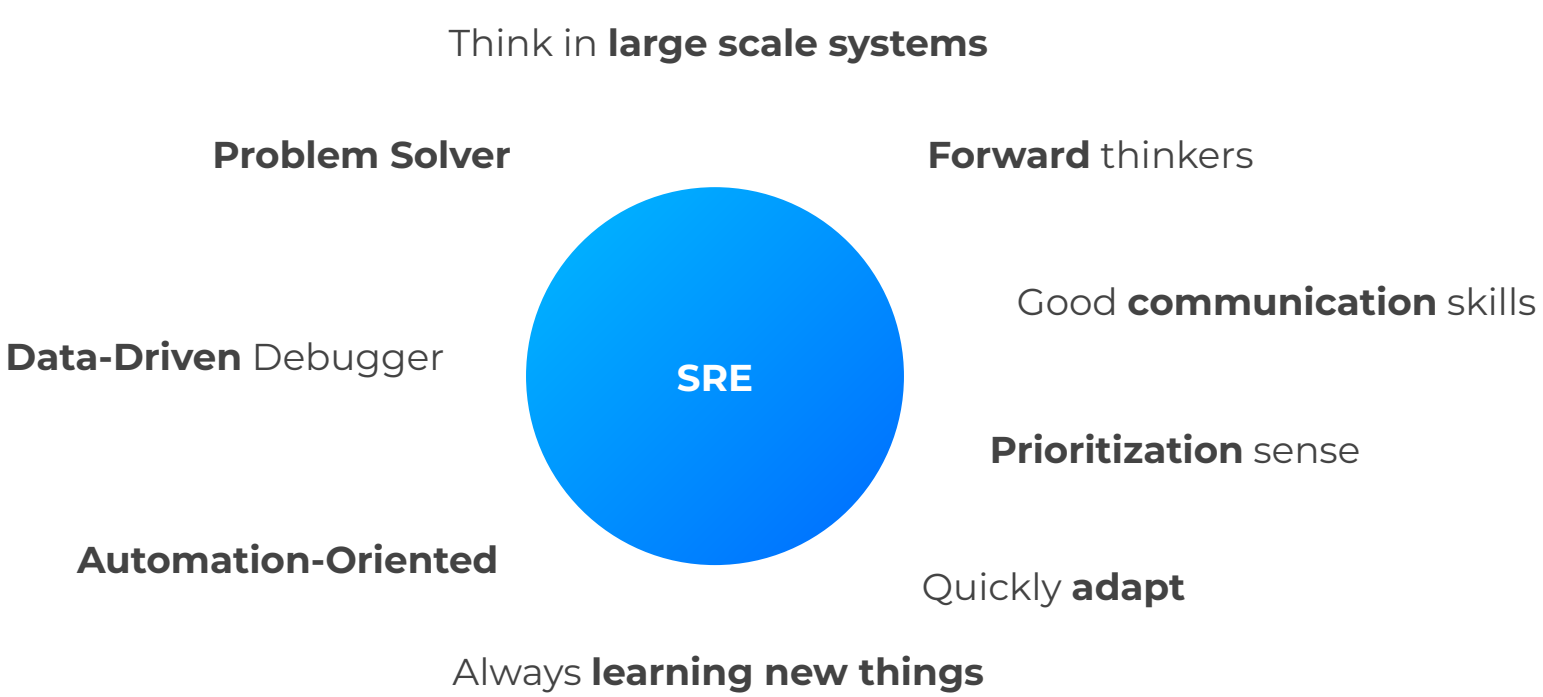
SYSADMIN - Hiring Site Reliability Engineers

Keys to SRE | USENIX - Ben Treynor

!=

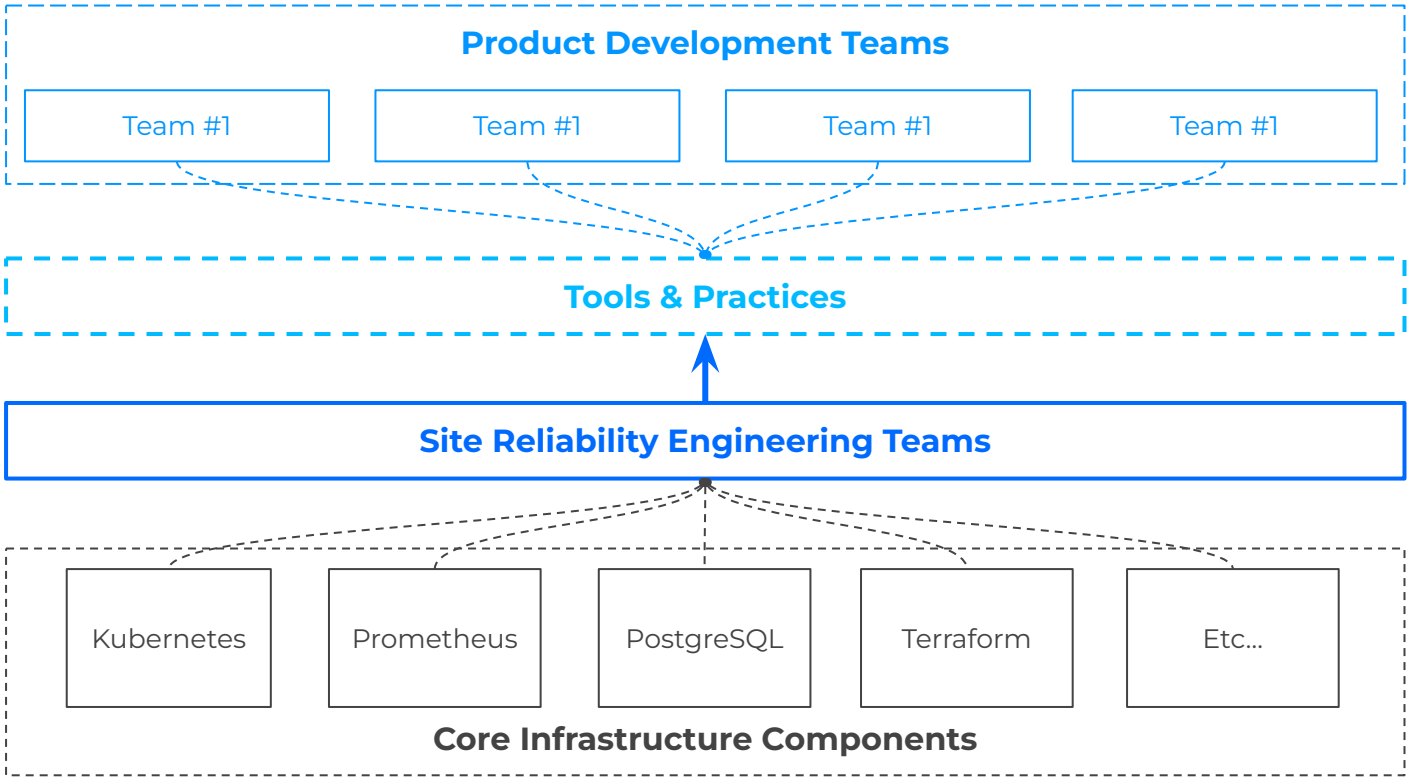
Characteristics

Most important skill for an SRE



How are we organized

The interaction with other engineering teams



The Journey...

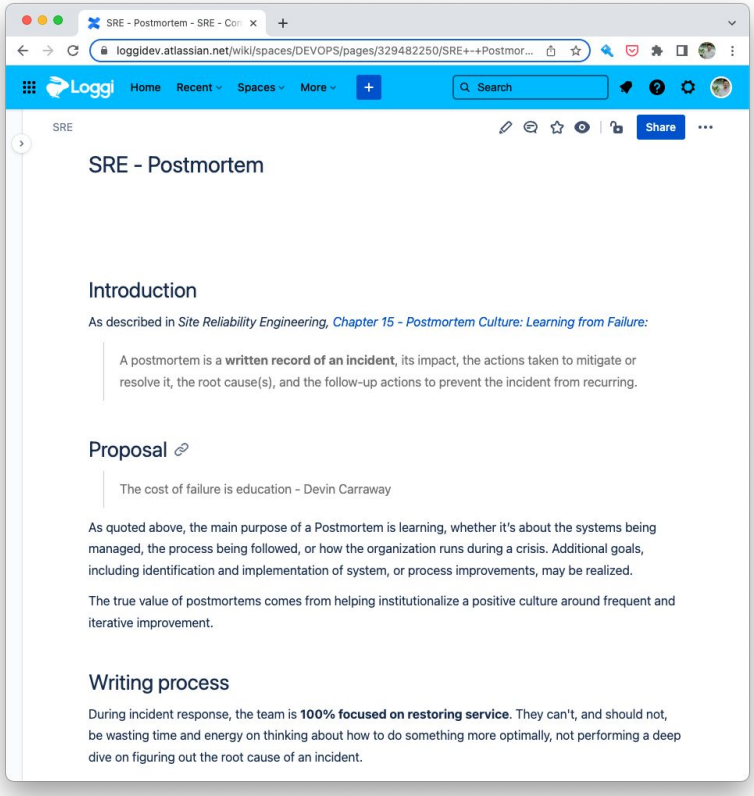
How we made our way until now

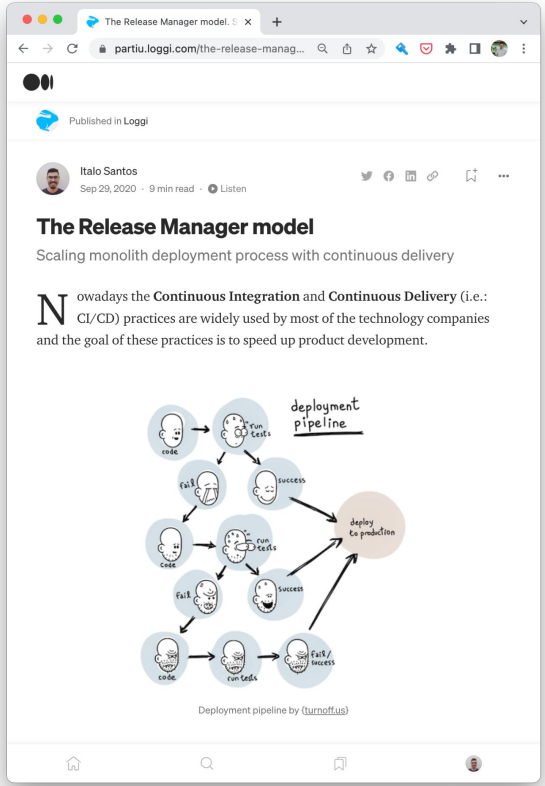
- Design a strong infrastructure
- Deployment Pipeline & Rotation
- Basic monitoring infrastructure
- Incident Response
- SRE Oncall Rotation
- Blameless Postmortems

Foundation | **Improve & Growth** | **Scale**

Foundation

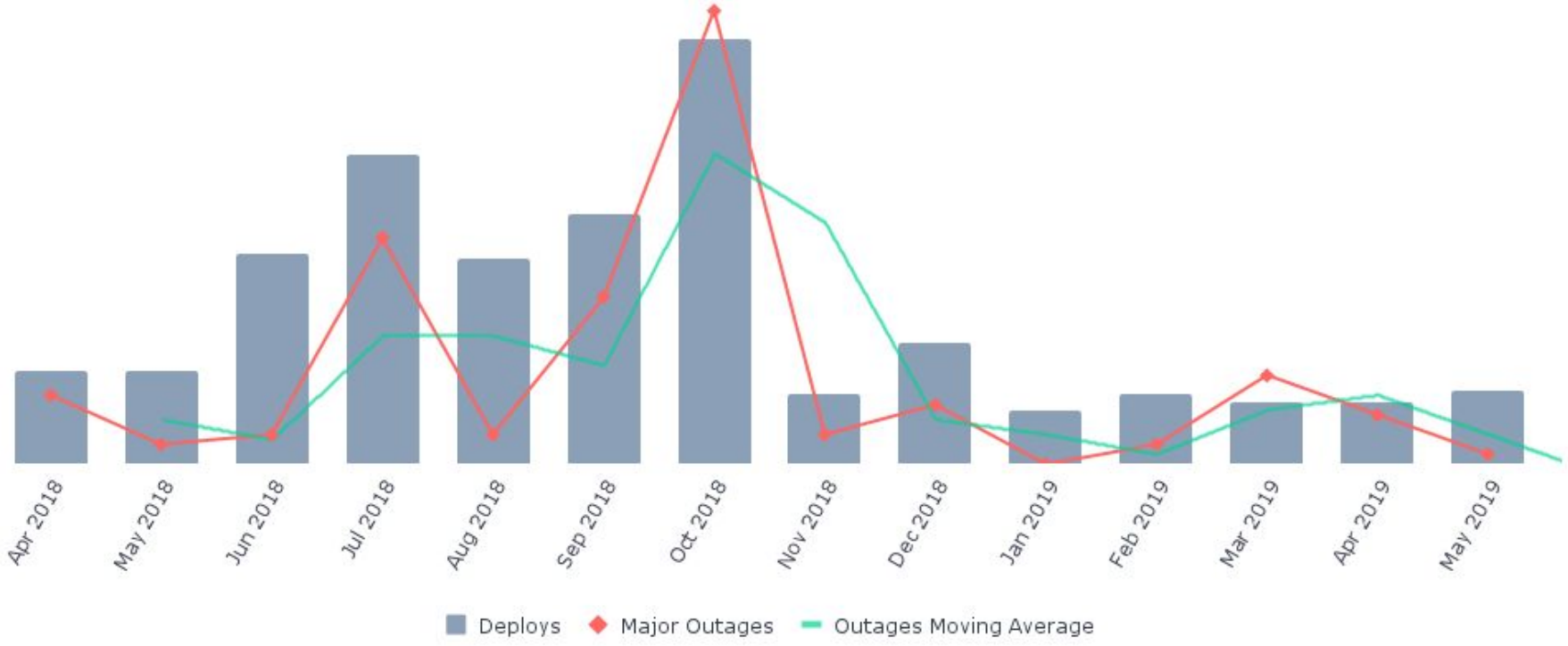
How build the base SRE practices





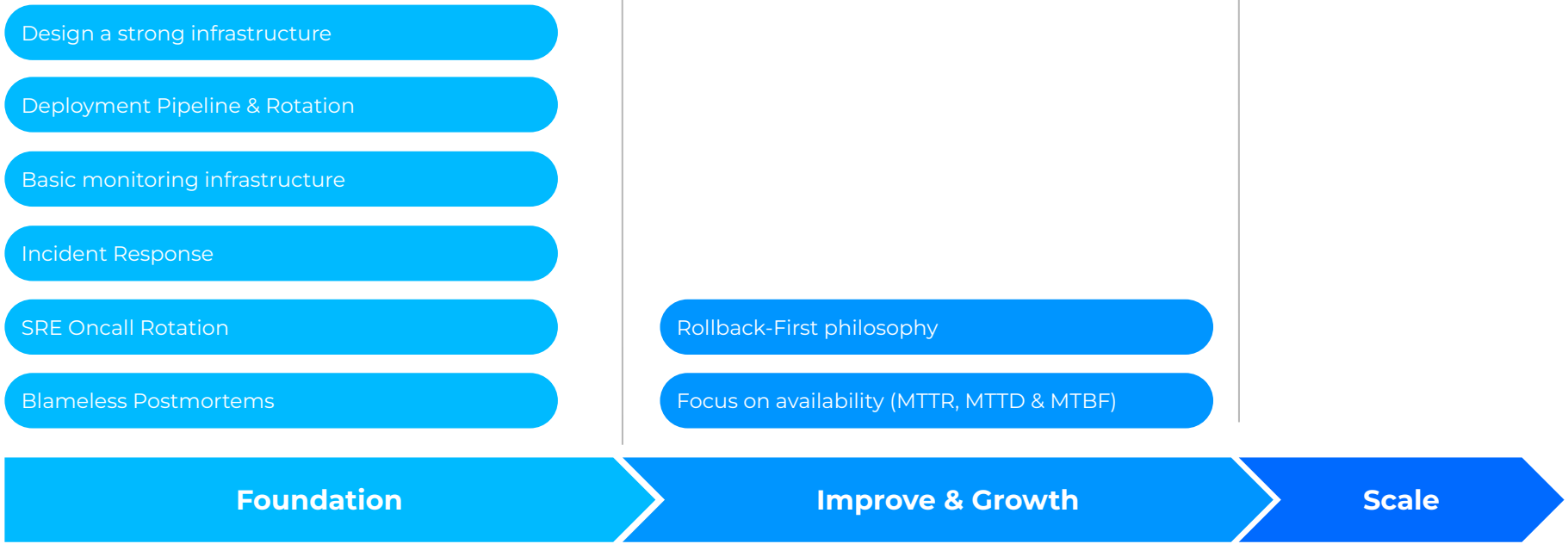
Foundation

How build the base SRE practices



The Journey...

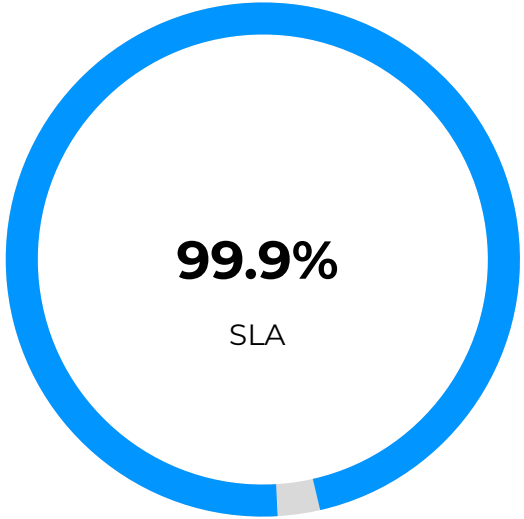
How we made our way until now



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

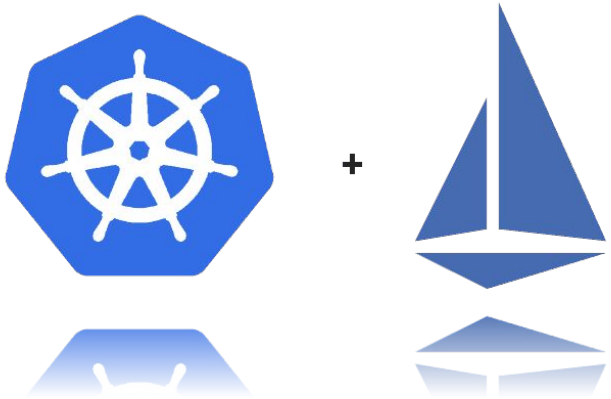
- Design a strong infrastructure
- Deployment Pipeline & Rotation
- Basic monitoring infrastructure
- Incident Response
- SRE Oncall Rotation
- Blameless Postmortems

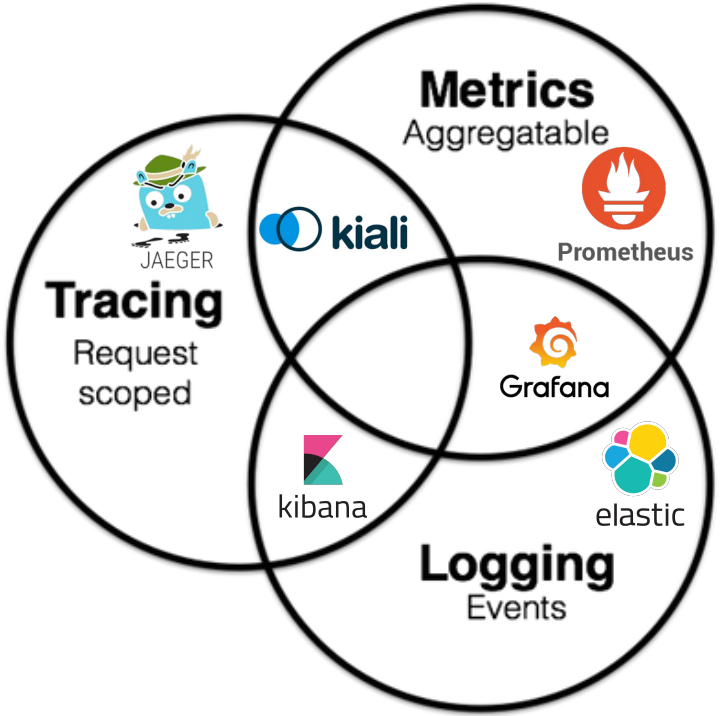
- Reliable & Scalable Infrastructure
- Improve observability tools
- Rollback-First philosophy
- Focus on availability (MTTR, MTTD & MTBF)

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

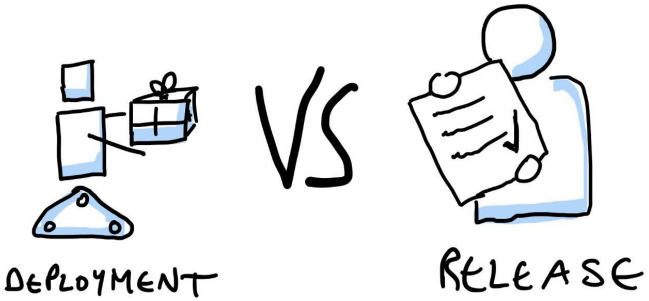
- Design a strong infrastructure
- Deployment Pipeline & Rotation
- Basic monitoring infrastructure
- Incident Response
- SRE Oncall Rotation
- Blameless Postmortems

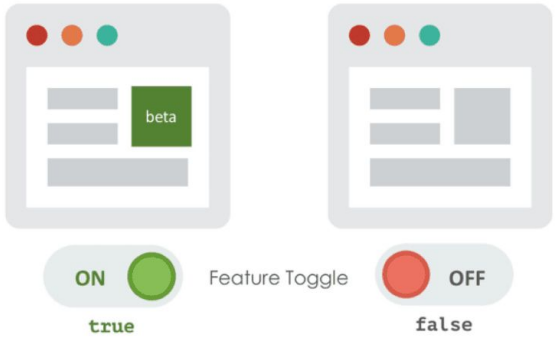
- Canary Deployments
- Feature Flags
- Reliable & Scalable Infrastructure
- Improve observability tools
- Rollback-First philosophy
- Focus on availability (MTTR, MTTD & MTBF)

Foundation → **Improve & Growth** → **Scale**

Improve & Growth

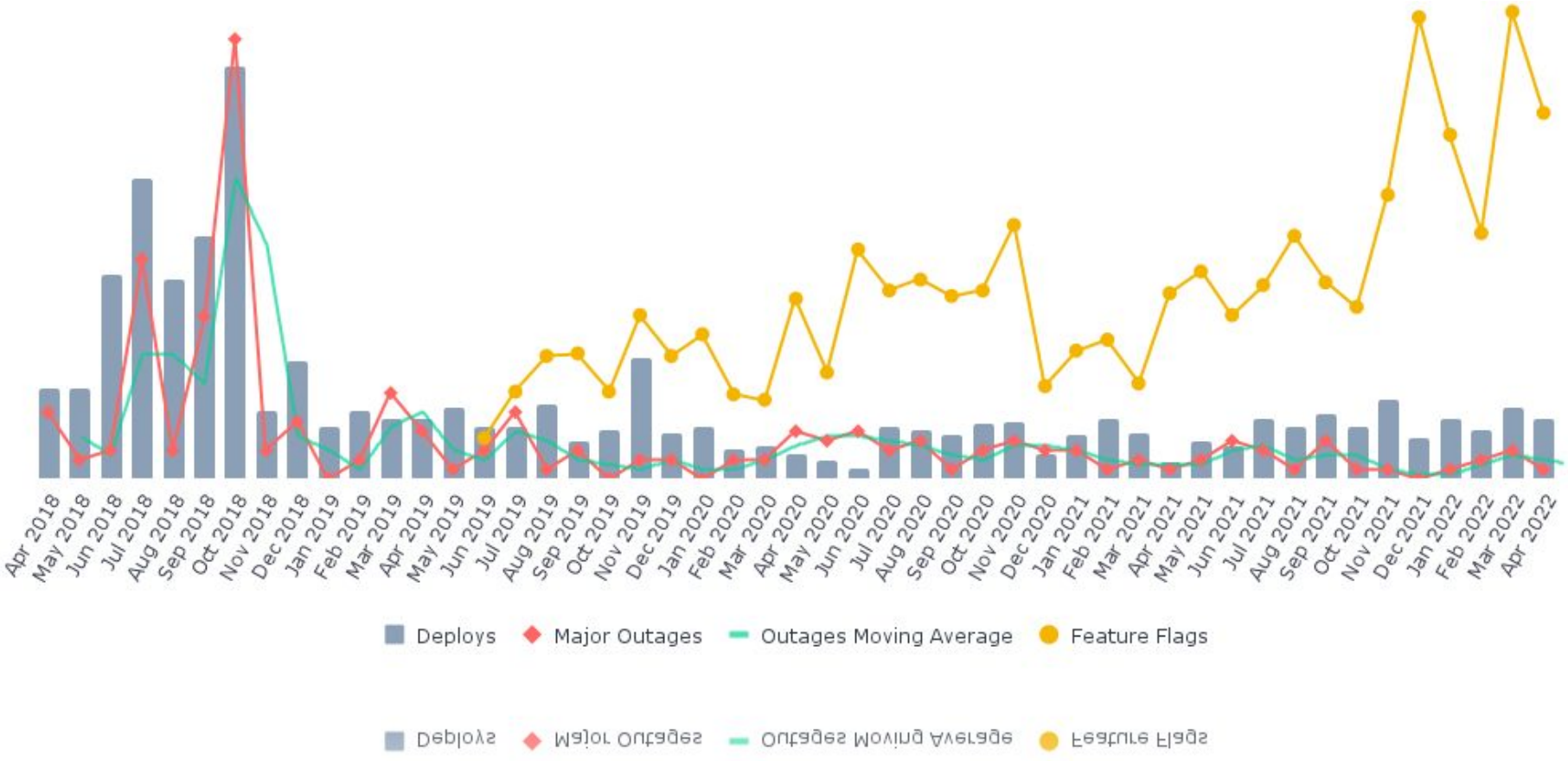
Decouple deploy from release and introduce the feature flags concept





Improve & Growth

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



The Journey...

How we made our way until now

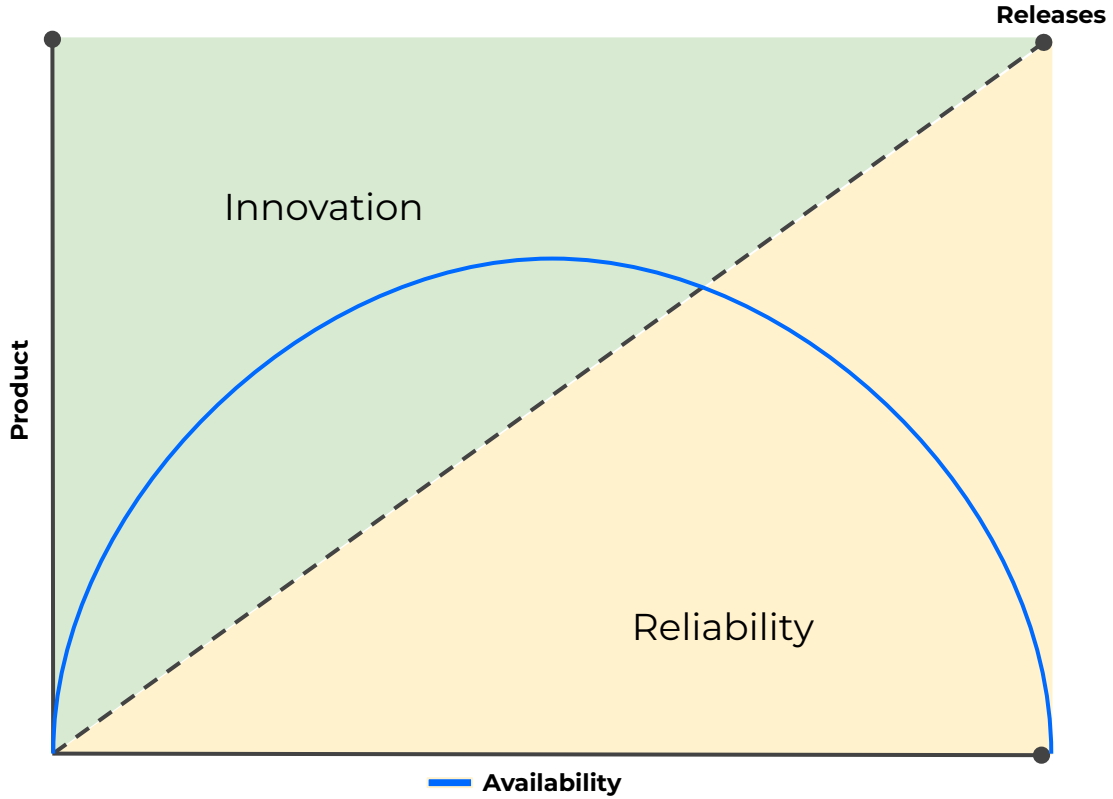
- Design a strong infrastructure
- Deployment Pipeline & Rotation
- Basic monitoring infrastructure
- Incident Response
- SRE Oncall Rotation
- Blameless Postmortems

- 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

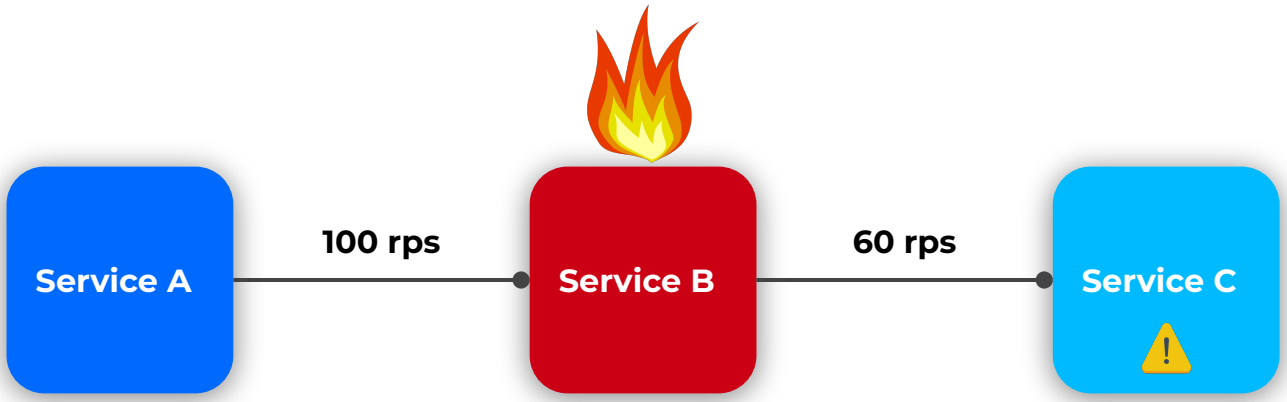


Allowed

Halted

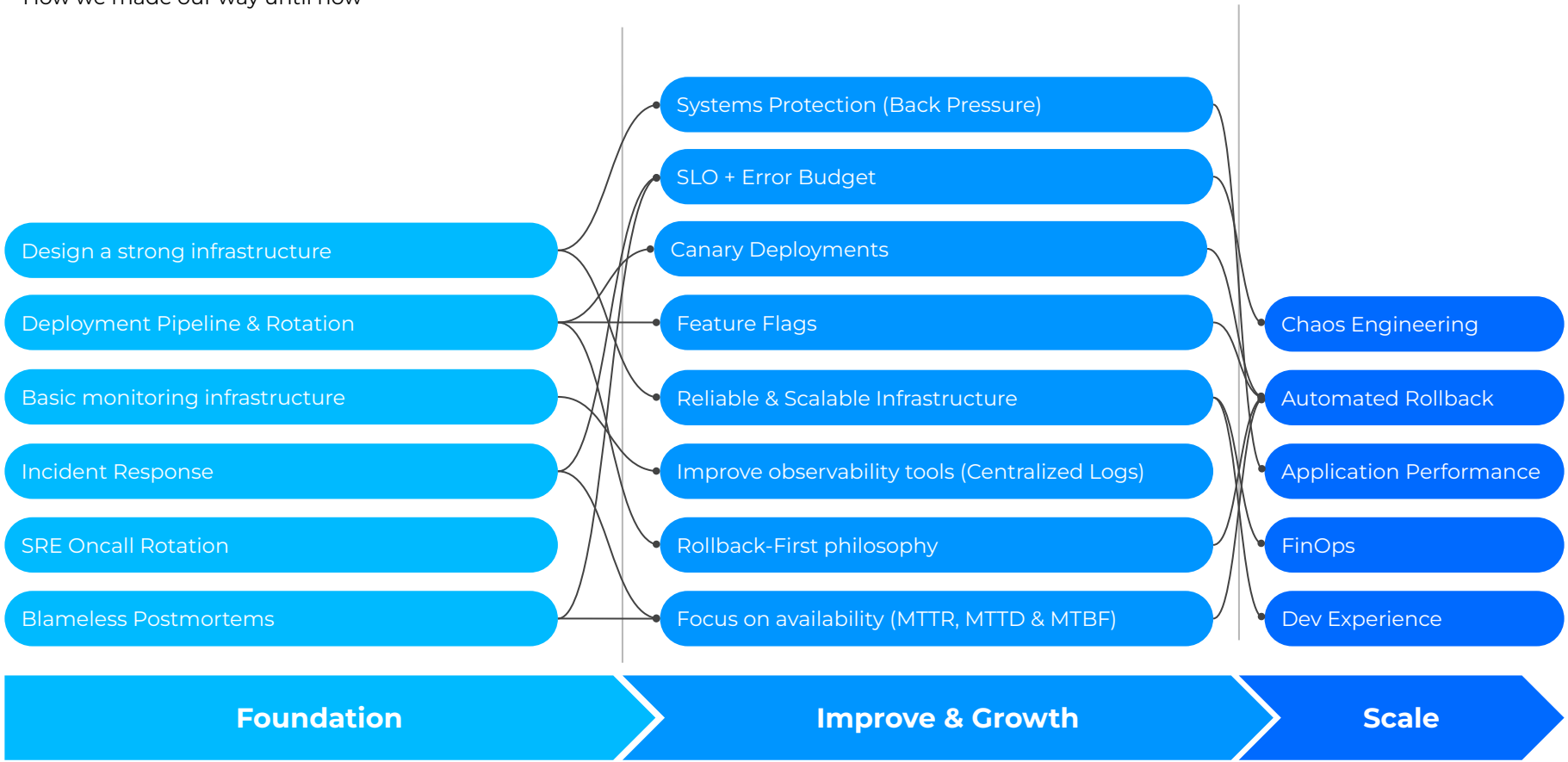
Improve & Growth

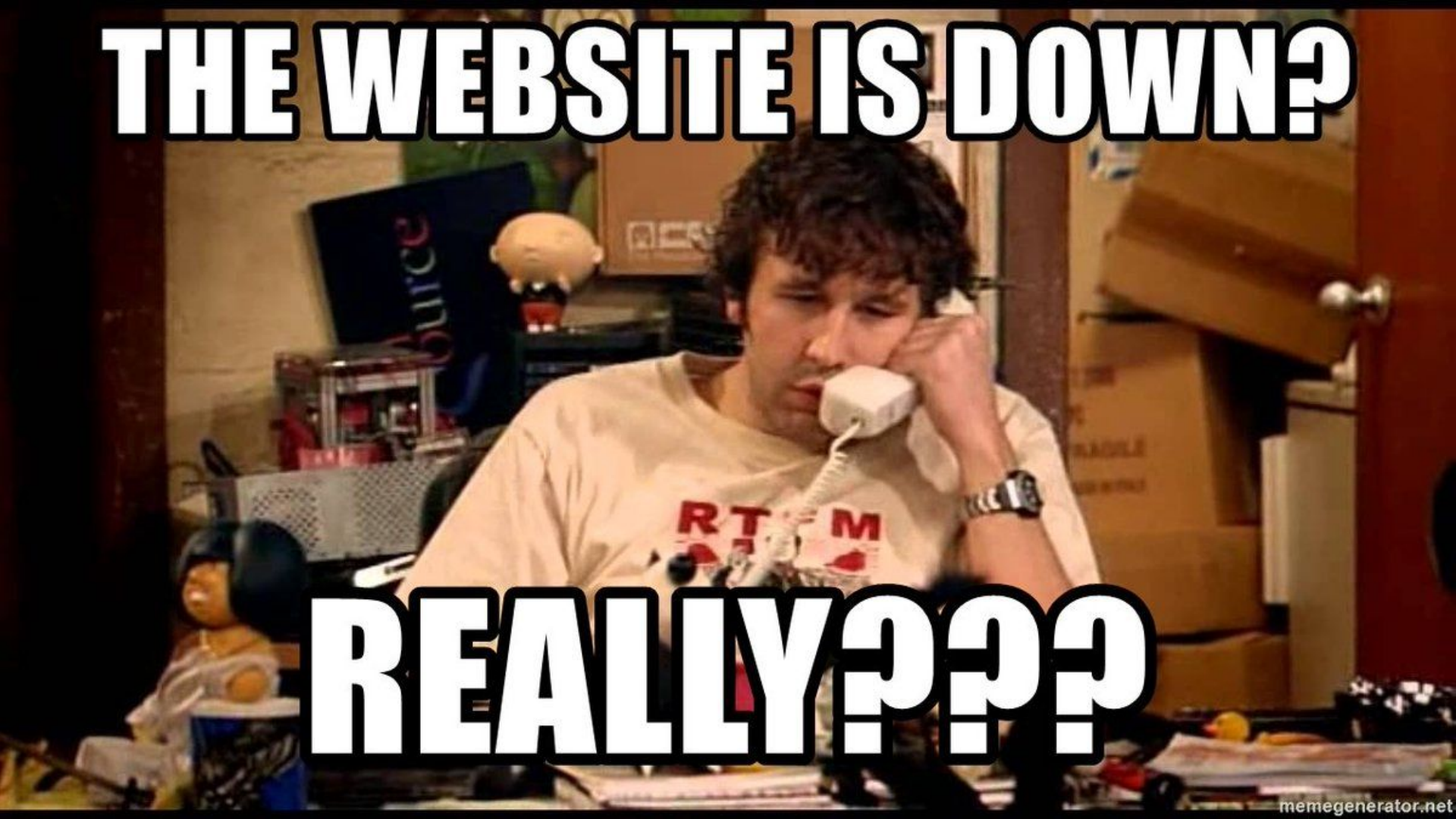
Protecting systems with back pressure



The Journey...

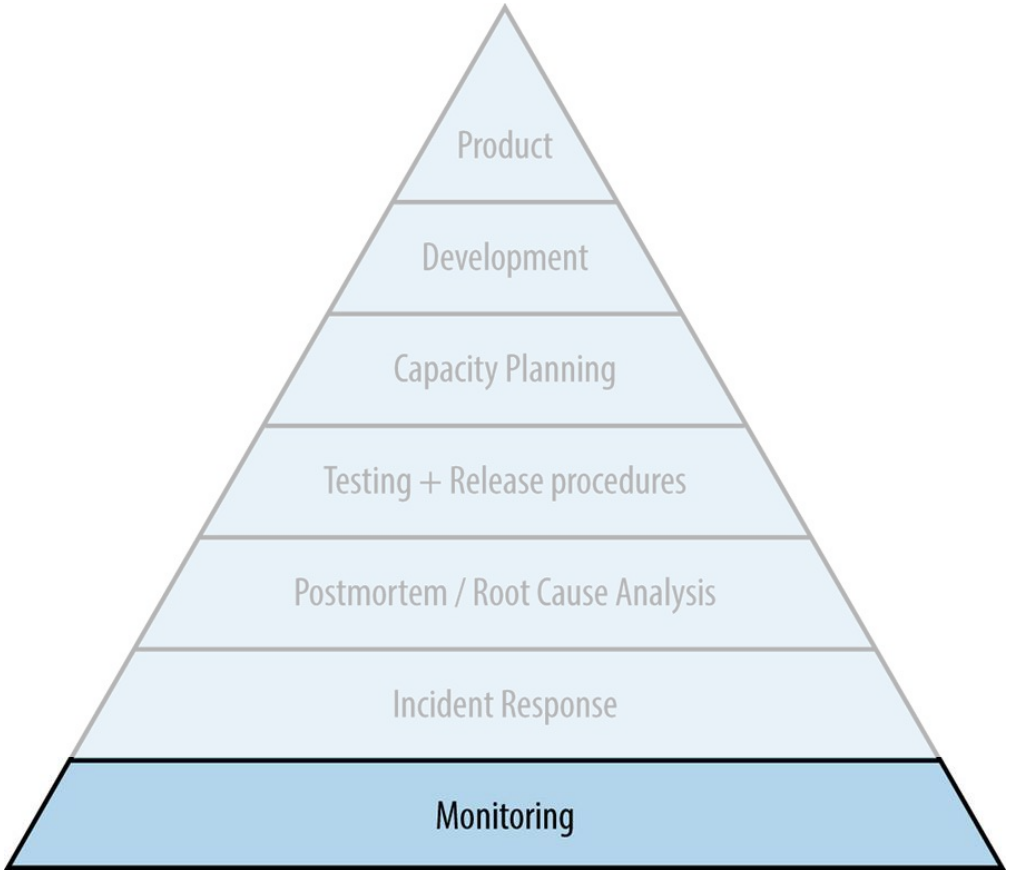
How we made our way until now




THE WEBSITE IS DOWN?
REALLY???
memegenerator.net

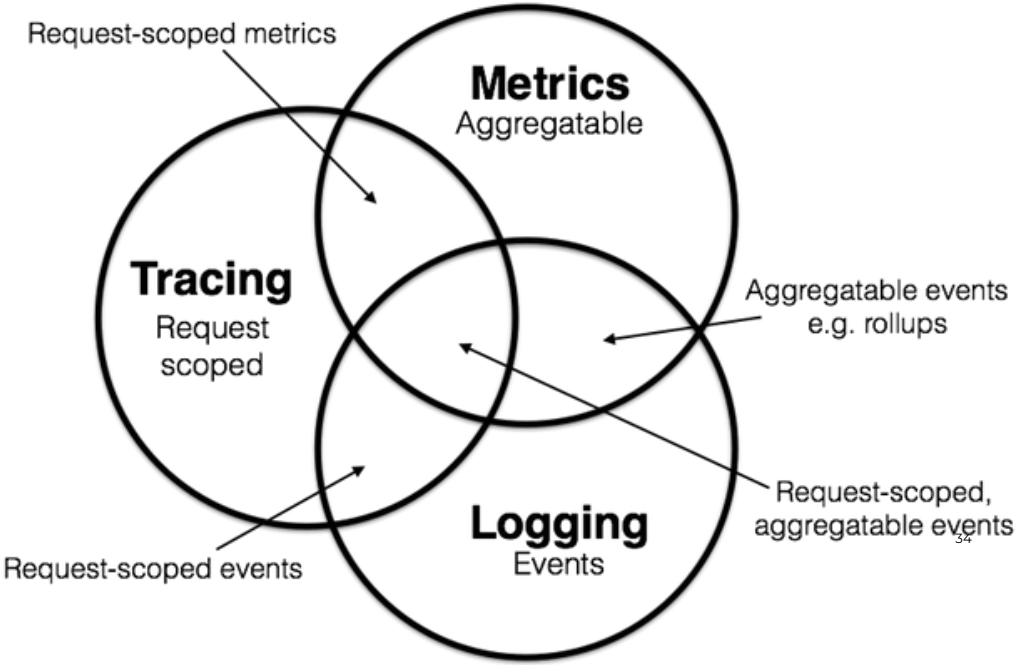
SRE Fundamentals

Service Reliability Hierarchy



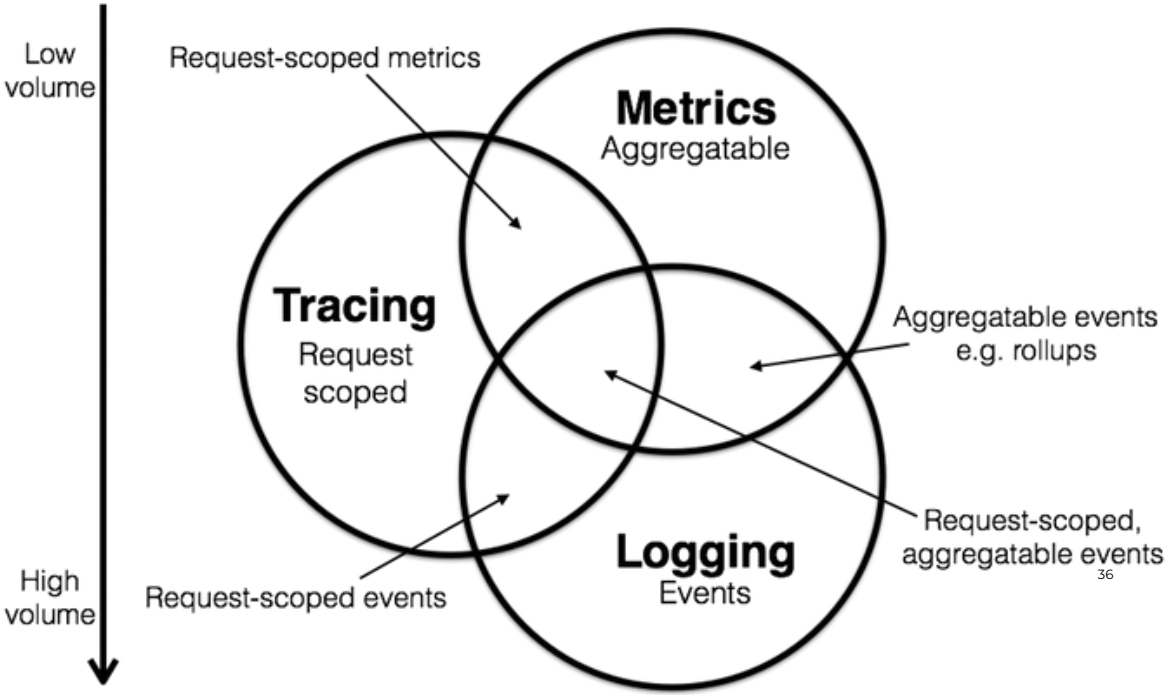
*“In control theory, **observability** is a measure of how well internal states of a system can be inferred by knowledge of its external outputs…”*

Observability Pillars




GET
ALL THE DATA!

Observability is data!



"Observability Isn’t a Panacea"

Implements

Observability vs Monitoring

- Dynamic
- Unpredictable
- Data

- Static
- Predicable
- Metrics

mon-i-tor

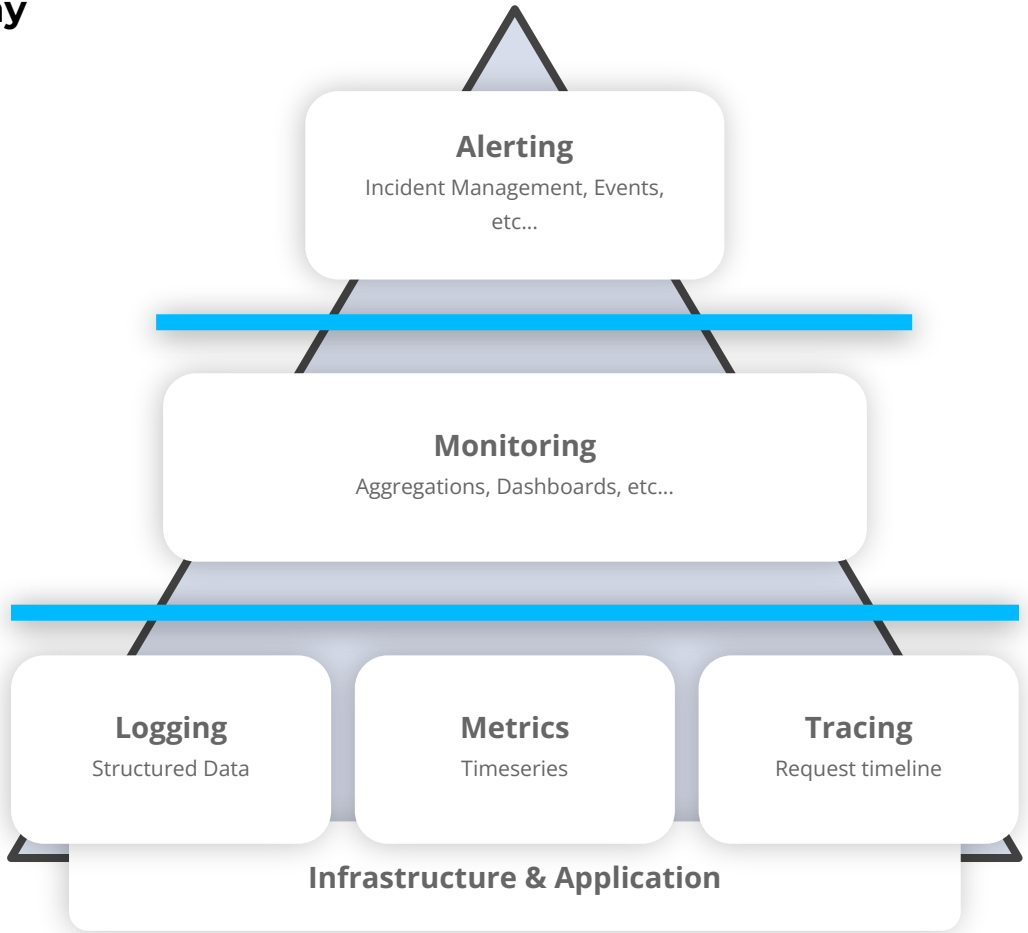
/'mänədər/

verb

gerund or present participle: monitoring

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

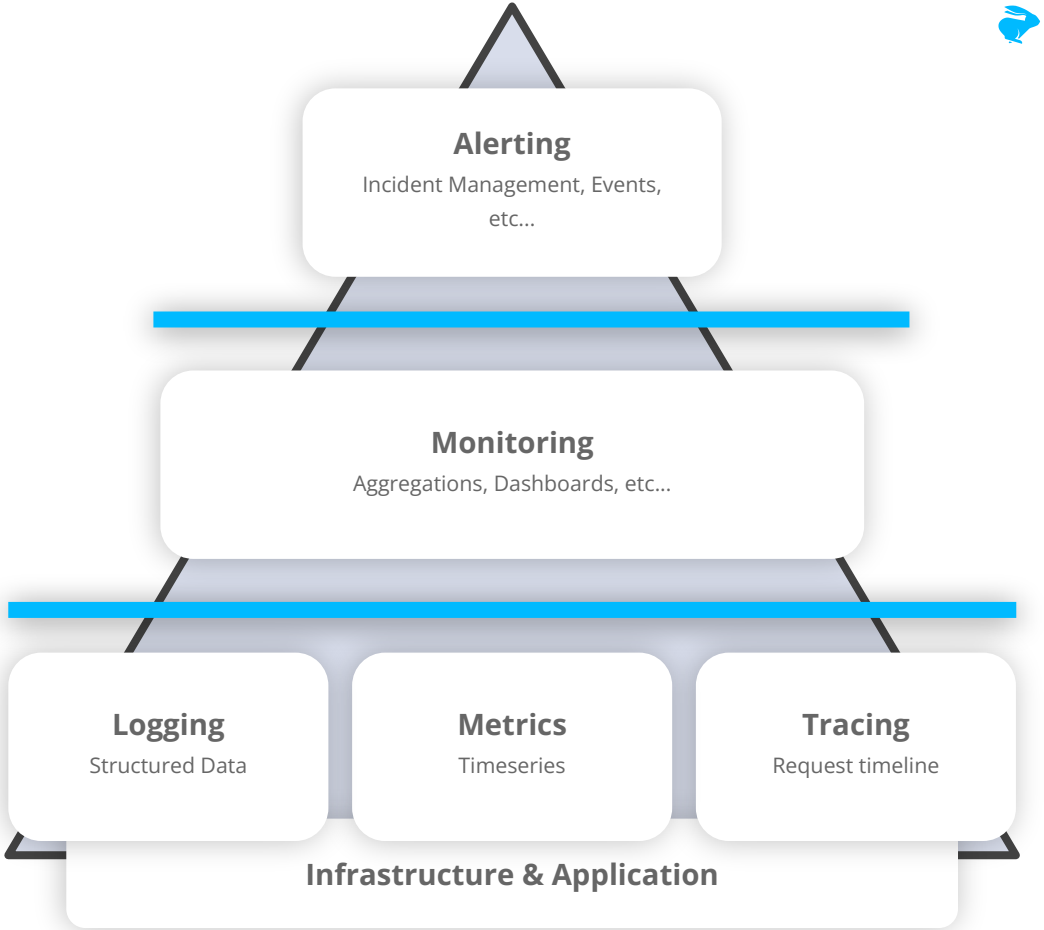
Monitoring Hierarchy



Monitoring Hierarchy

_Metrics

_Diagnostic



Metrics

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

vs

Diagnostics

- Log Analysis
- Debugging
- Postmortem & Investigation

Symptom

what?

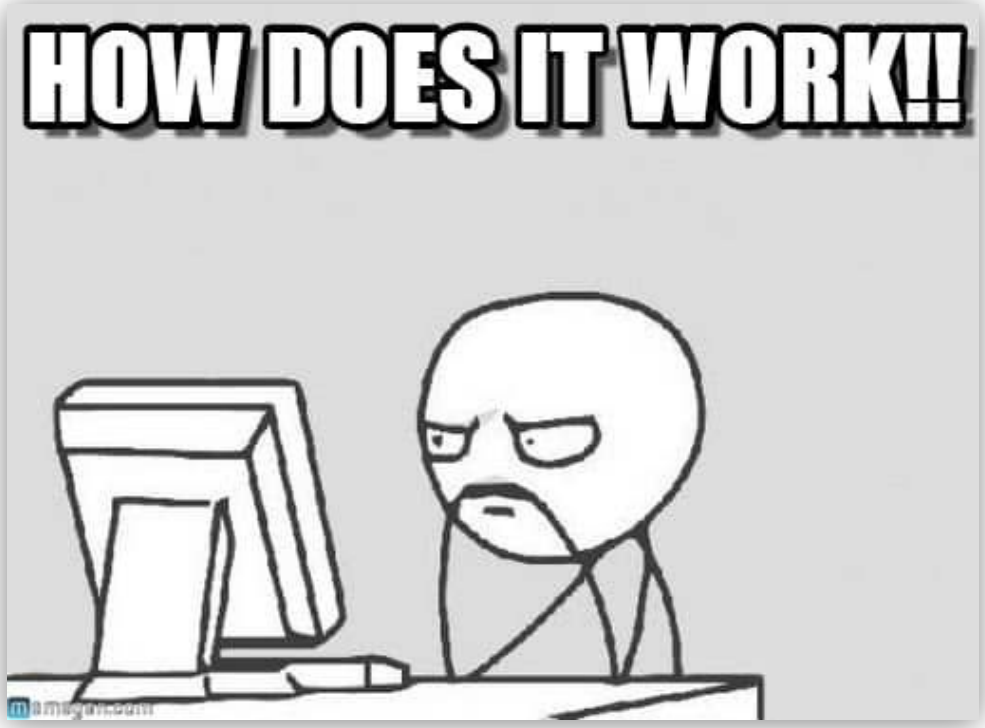
- High HTTP errors
- System is slow

vs

Cause

why?

- Database is down
- Network congestion


HOW DOES IT WORK!!

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, **E**rror & **D**uration

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

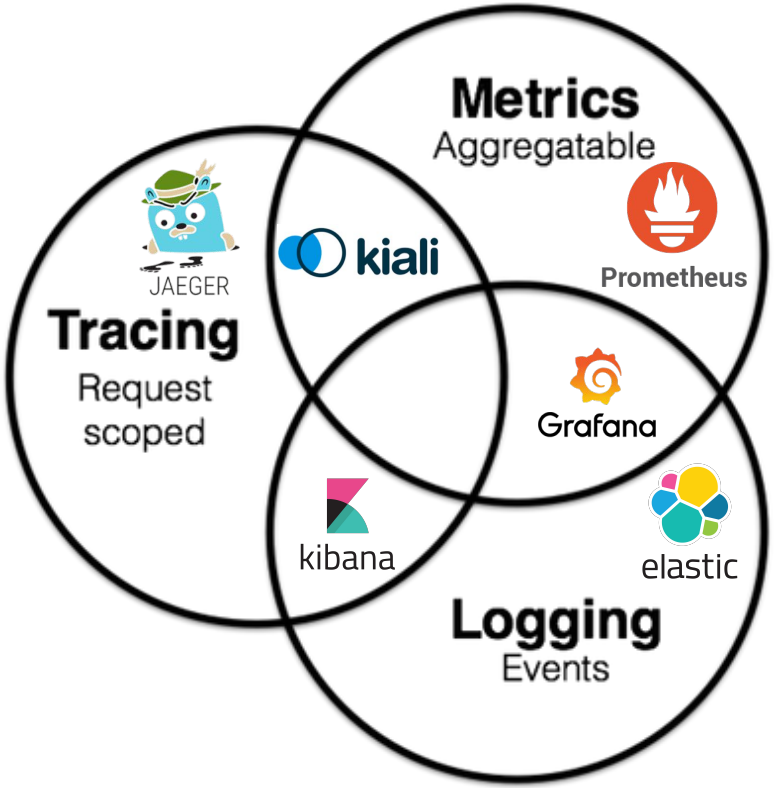
The USE method

Utilization, **S**aturation & **E**rror

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

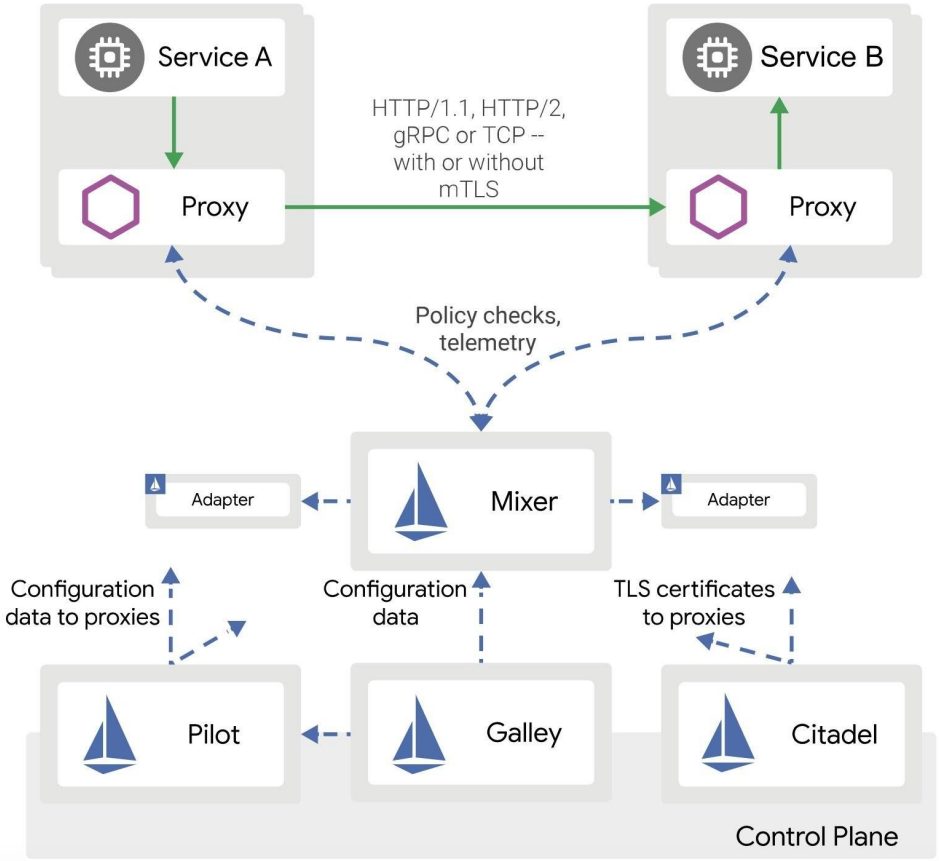
Observability Ecosystem

What tools we use



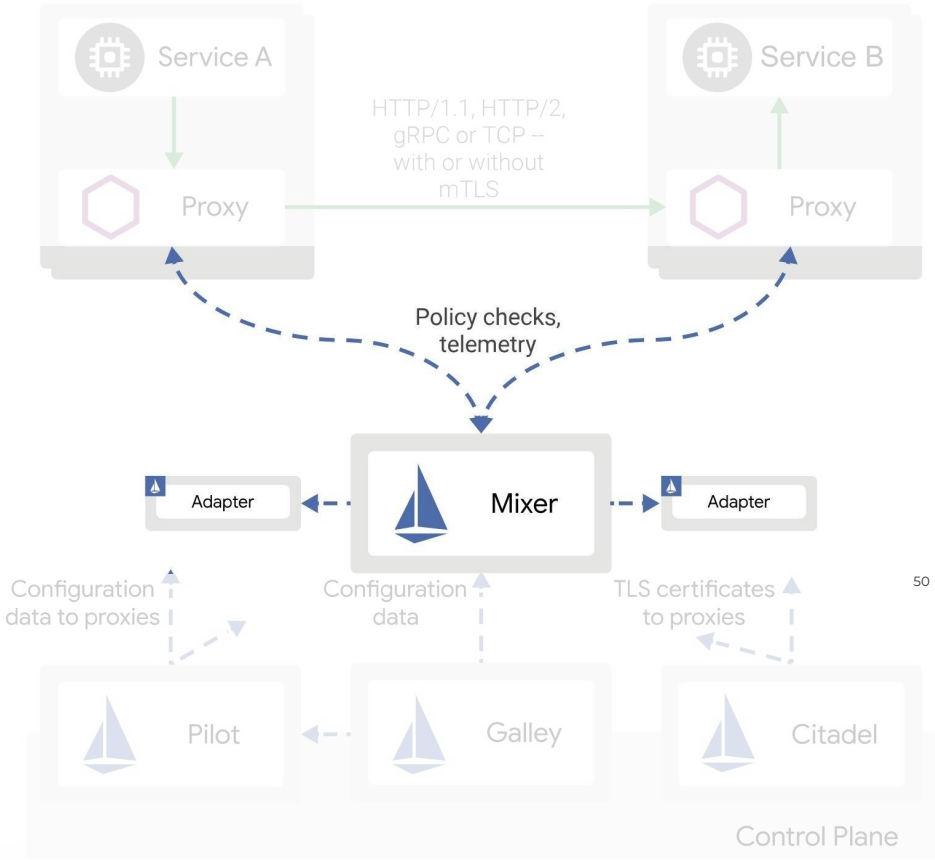
Istio & Observability

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.com