Put SLOs in your business teams' hands

Martin Piegay / Damien Bertau

Hello

Today's agenda

01

Quick Intro to SLIs/SLOs/Error budget

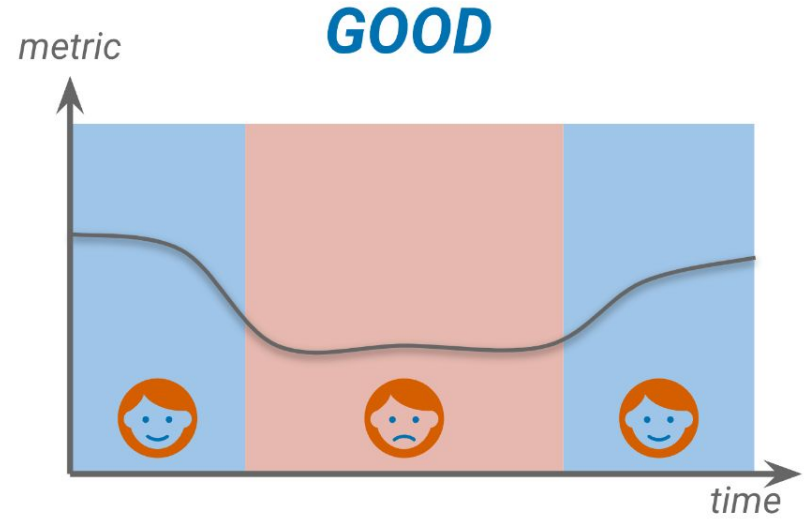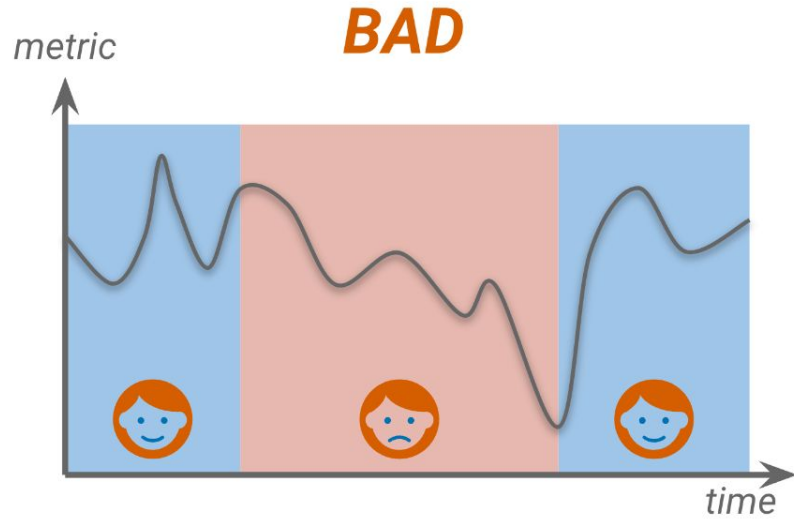# Why do we need SLOs ?

1 A way to measure **reliability** from **user** point of view

2 A **common language** for all your teams
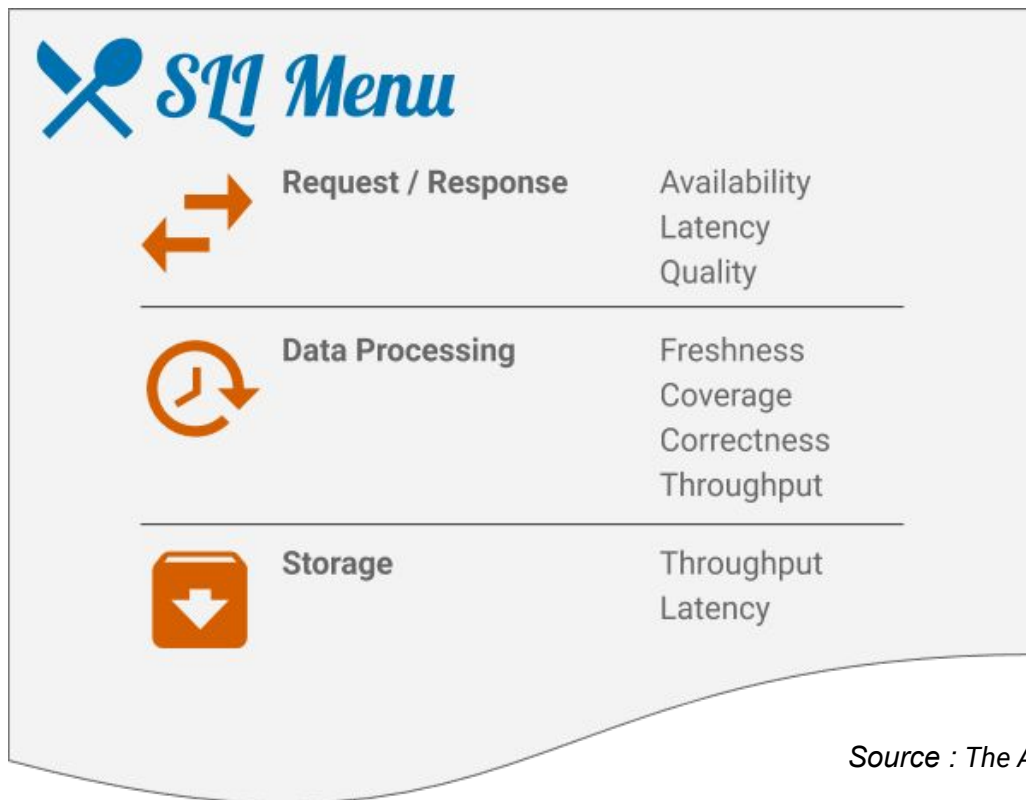
3 A way to fight **Alert-Fatigue**

Service Level Indicator

- A **metric** that measure **one aspect** of a service's reliability

- meaningful for the service consumer

- computed in an automated way

Source : The Art of SLOs – Slides (Google)

6

Types of SLIs

SLI Menu

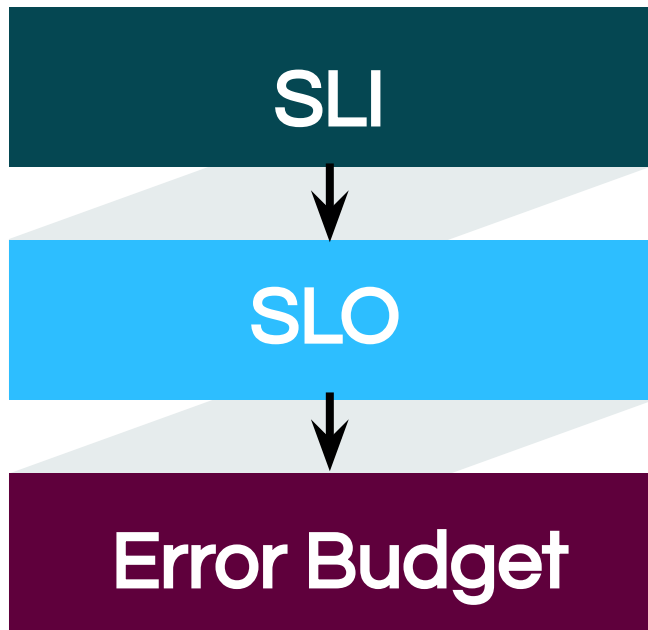|	Request / Response	Availability Latency Quality
	Data Processing	Freshness Coverage Correctness Throughput
	Storage	Throughput Latency

Source : The Art of SLOs – Slides (Google)

7

Service Level Objectives

- A **target** for one SLI

- Over a **measurement window**

- It represents the consumer expectations

SLI, SLO and Error Budget
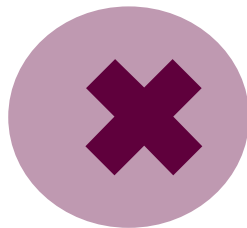
SLI

My workload is **Up** or **Down** ?

SLO

Objective set to **99.9%** up over **30 days**.

Error Budget

Allowed downtime is **43 min**
(0,1% of 30 days)

Why it can go wrong ?

Releasing
new (buggy)
features

Inevitable
Failures
(hardware, network,
providers)

Risky
experiments

Our Implementation of HTTP SLOs

Deployment Plan

1

Base

Deploy technical SLOs on HTTP endpoints

2

Combine

Deploy feature SLOs

3

Policy

Ensure SLO reviews are in place

HTTP SLOs

On **every** HTTP endpoints

- SLI

  - **Datadog Monitor** on Latency / Availability (/ Quality)

- Monitor based **Datadog SLO**

  - 2 windows: 7d / 30d

- Burn Rate Alerts

  - Datadog Monitors on **Error Budget** Consumption

Example of HTTP SLOs

[Publish][inventory] availability post_/experience/return-flow

Status & History   Alerts   Corrections

SLO (profile=silver) for [Publish][inventory] - **request_route:**corpool-inventory-return-flow

Created by Jean-Baptiste Favre on Dec 20, 2022, 3:38 pm

QUERY

```
9205682 && 8681904
```

TAGS ✎ managedby:**terraform** squad:**publish** service:**inventory** team:**team_**corpool_core slo:**availability** env:**prod** request_route:corpool-inventory-return-flow profile:**silver**

	Past 7 Days	Past 30 Days Primary
TARGET (SLO)	99.57%	99.8%
STATUS	**99.09%**	**99.65%**
ERROR BUDGET	**-110%** (47m 39s)	**-72%** (1h 2m)

History

Past 7 Days   **Past 30 Days**

Overall Uptime		**99.65%**
	Feb 26 Mar 5 Mar 12 Mar 19	30 Days Error Budget
[Publish][inventory] availability corpool-inventory-re...		**99.65%** -1h 2m
[Publish][inventory] Deployment status		**100.00%** 1h 26m

The overall SLO status is calculated as the uptime percentage across all monitors or monitor groups, unless specific groups have been selected. If no specific groups are selected, the UI displays the five groups with the worst statuses. See our documentation ⧉ for more details.

Deployment: Terraform Module

Benefits of a Terraform Module

- Ease adoption: Automate the creation

- Standardization

  - Name Convention

  - Metrics: APM / Istio

  - Profiles: Objective/Error Budget

- Easy to rollout changes and update

Name	timeframe	SLO target	Error budget
platinum	30d	99.95	0 day(s) 0h21m55s
platinum	7d	99.89	0 day(s) 0h10m57s
gold	30d	99.9	0 day(s) 0h43m50s
gold	7d	99.78	0 day(s) 0h21m55s
silver	30d	99.8	0 day(s) 1h27m40s
silver	7d	99.57	0 day(s) 0h43m50s
bronze	30d	99	0 day(s) 7h18m18s
bronze	7d	97.83	0 day(s) 3h39m9s
steel	30d	95	1 day(s) 12h31m28s
steel	7d	89.13	0 day(s) 18h15m44s
tin	30d	90	3 day(s) 1h2m56s
tin	7d	78.26	1 day(s) 12h31m28s
zinc	30d	75	7 day(s) 14h37m21s
zinc	7d	45.65	3 day(s) 19h18m41s

16

```
module "slos" {
  source = "git@github.com:blablacar/terraform-modules.git//datadog/slo?ref=slo-v1.1.7"

  team = local.teams["sre"]
  config = {
    alerting      = { priority = 1, recipients = ["@pagerduty-sre"] }
    name          = "Inventory"
    app_selector = "service:inventory,env:prod"
    watch_deployment = true
    routes = [
      {
        resources = ["resource_name:route1selector","resource_name:route2selector"]
        latency = {
          base_metric      = "trace.servlet.request"
          threshold        = 0.7
          slo_profile      = "silver"
        }
      }
    ]
  }
  tags = ["service:inventory", "env:prod"]
}
```

Next Step: Display SLOs from a product point of view

- How to make it meaningful for non tech teams ?

- Gather availability per feature (like : signup, booking, search...)

- List all endpoints per feature, then build a combined SLO from these endpoints.

03

Feature SLOs

Feature SLO : How to compute target ?

Availability Targets

api gateway — 99,0%

backend — 99,9%

system uptime — **99% * 99,9% = 98,9%**
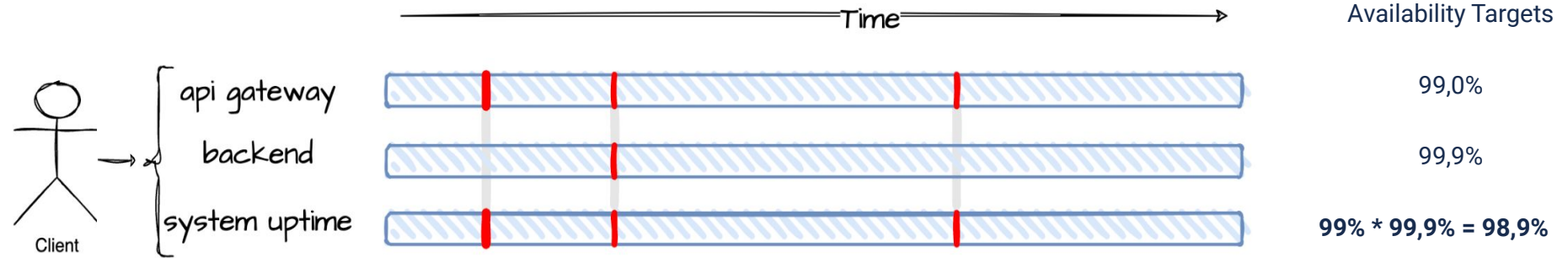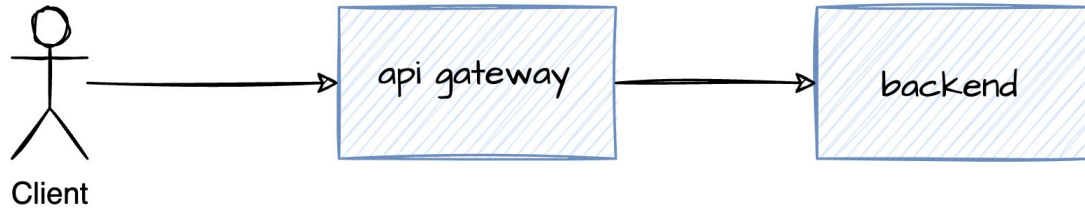
The feature availability SLO is simply calculated by multiplying the availability of the two systems

Example of the Search Feature

Feature SLO : How to combine complex dependencies
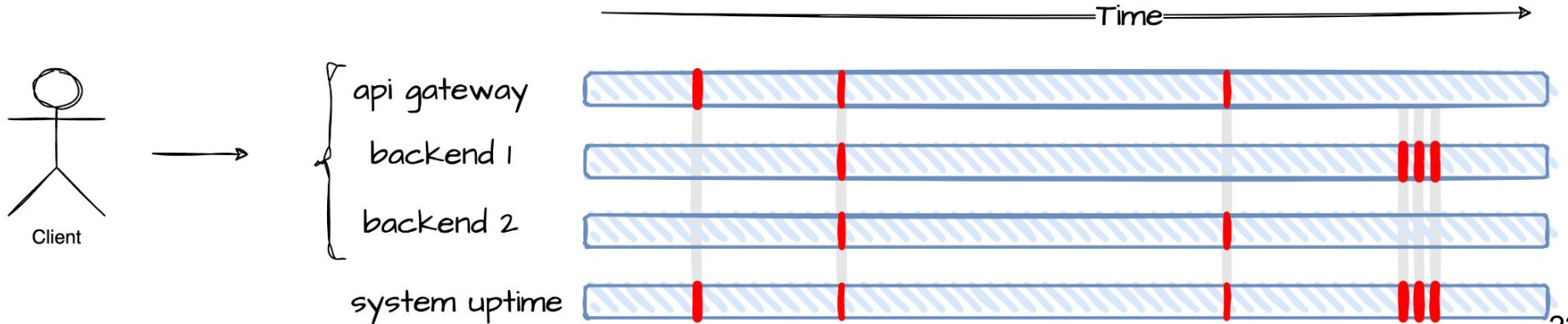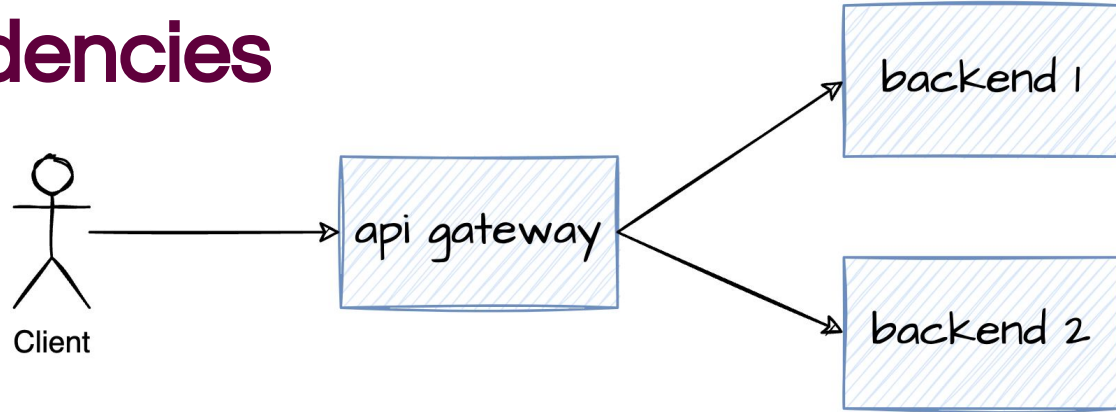
Feature SLO in Datadog
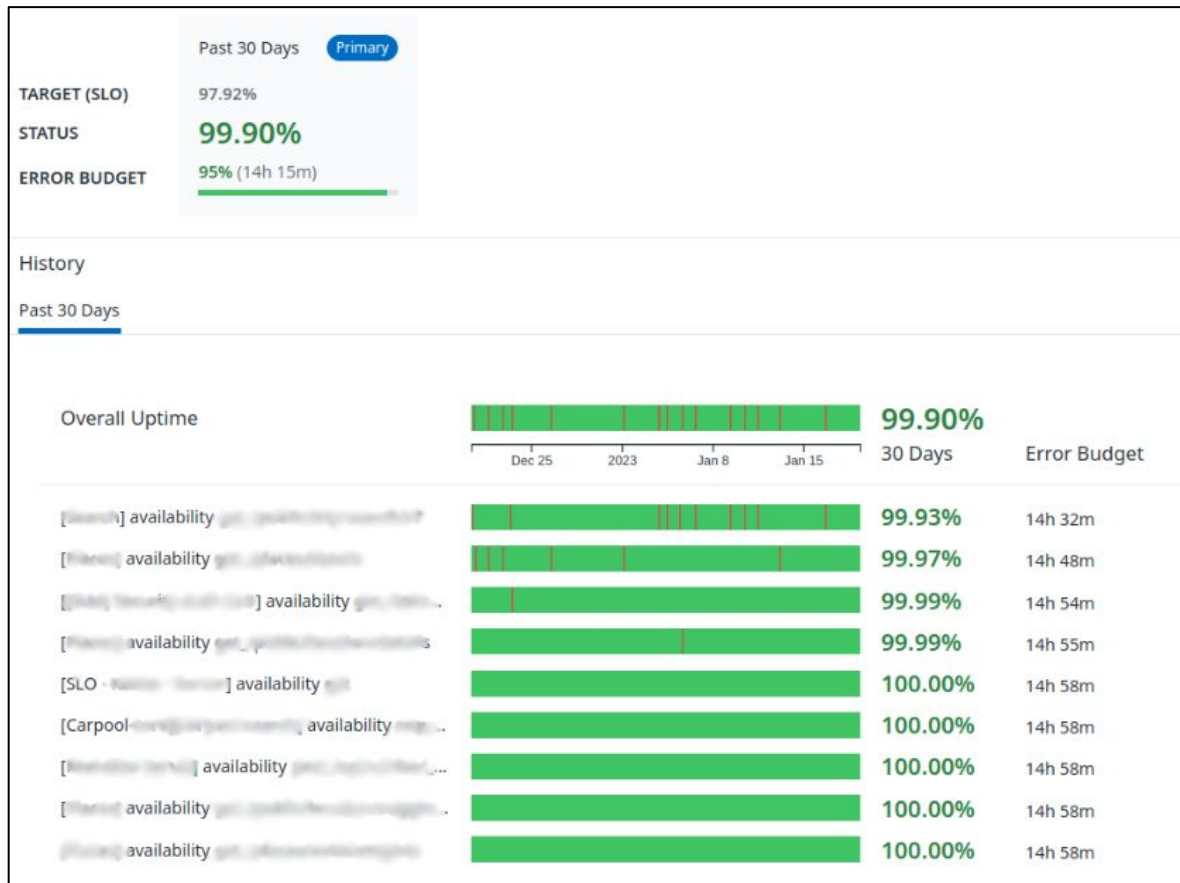
Feature SLO : How to publish them ?

Legend:
- Existing component
- Added component

1. Publish feature SLOs KV
2. Runs
3. Get feature SLOs KV
4. Create
5. Create
6. Apply

Components: HashiCorp Consul, Terraform Config (State Output), GitHub Action, Feature SLO Tool, Terraform Config TF file, Terraform Config Pull Request, DATADOG

24
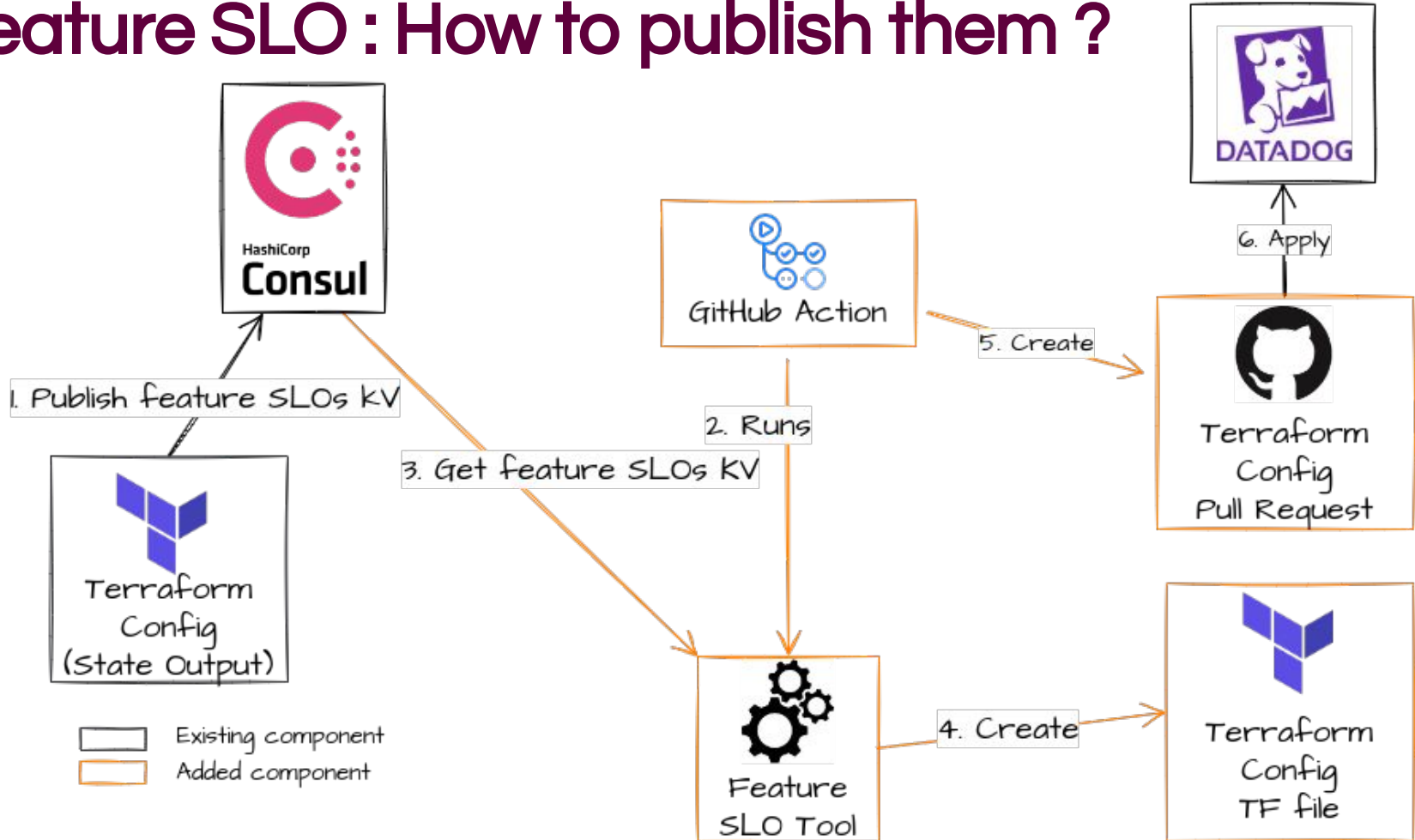
Feature SLO : Monthly Publication

Feature Availability Summary

URL	Pillar ⓘ ▲	Feature	Target	Status	% Δ	Remaining Budget	% Δ
●	Passenger Experience	b	97.42	97.55	1.5% ↑	00:56:35	110.1% ↑
●	Passenger Experience	bo	97.42	96.7	-0.2% ↓	-05:16:07	-48.0% ↓
●	Passenger Experience	bus	99.3	98.65	-1.0% ↓	-04:45:42	-302.8% ↓
●	Passenger Experience	carpool	98.51	99.61	-0.1% ↓	08:00:34	-9.4% ↓
●	Passenger Experience	di	99.5	99.8	-0.2% ↓	02:09:51	-37.7% ↓
●	Passenger Experience	s	97.33	99.33	-0.4% ↓	14:37:32	-15.3% ↓
●	Platform	driver	94.15	100	-	42:44:01	-
●	Platform	driver	96.06	100	-	28:46:52	-
●	Platform	log	99	99.98	0.0% ↓	07:08:08	-1.9% ↓
●	Platform	passenge	92.27	100	-	56:28:00	-
●	Platform	passenger	97.03	100	-	21:41:43	-
●	Platform	r	98.81	99.17	-	02:37:01	-
●	Platform	reset	99.2	99.23	-0.7% ↓	00:14:47	-95.6% ↓
●	Platform	si	98.61	98.08	0.0% ↑	-03:54:13	6.8% ↑

Last step : SLO Reviews

📅 From the publication, regularly review technical SLO

    Tech Team
        ✖
    Product Team

🔧 If necessary adjusts thresholds/targets

📊 Prioritize reliability topics in the sprints/roadmap

04

Key results

Feature SLO : Summary

Teams are looking for this high level observability

Better visibility of SLOs through the company

Better Product & Tech priorisation
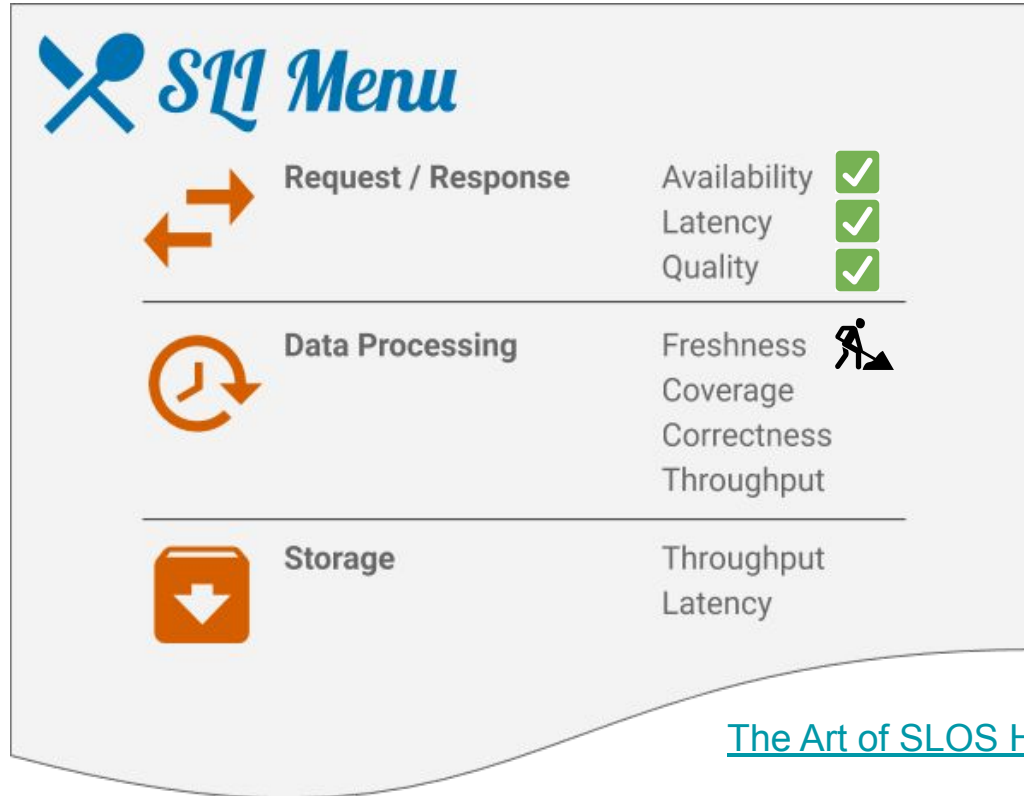
Numbers of Alerts is decreasing

05

Next steps and takeaways

Next steps: add other SLO types

SLI Menu

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

The Art of SLOS Handbook

Next steps: continue to onboard teams

SLO Policy

SLO Reviews

Takeaways

Start simple

Onboard teams

Questions

"Thanks!