Service Levels, Monitoring and Alerting

Elton Stoneman

Freelance Consultant and Trainer

@EltonStoneman | blog.sixeyed.com

"The Business"

Ops Team

Why not 100%?

99% availability! Agreed?

What happens at 98%?

IT Management

Service Level Objectives

Success rate 99.9% of requests have 2xx response

Response time 90% of requests within 0.5 seconds

Response time 99% of requests within 2 seconds

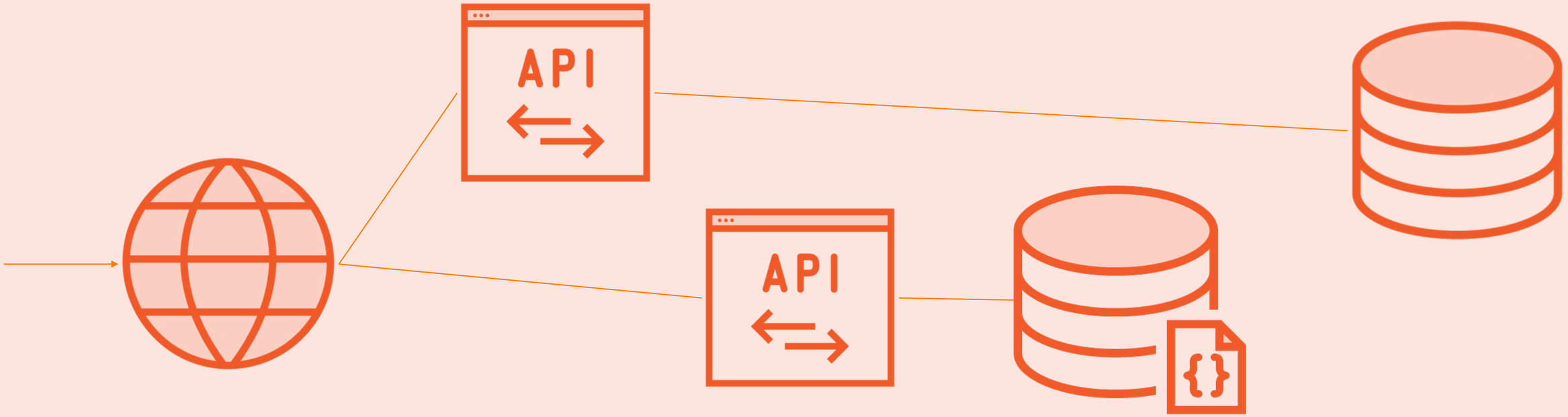
Service Level Objectives

Service Level Objectives

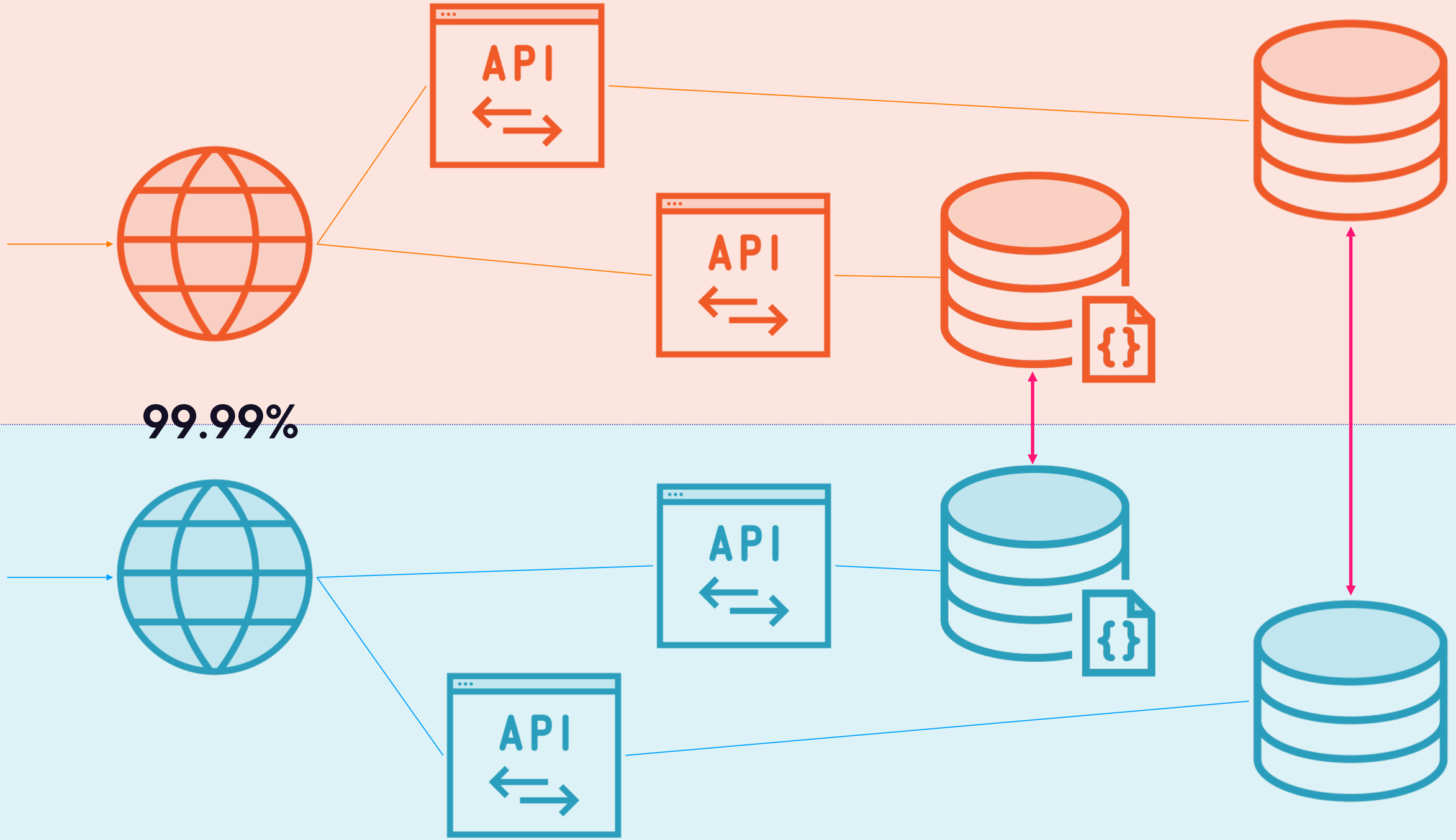
Response time 99% of requests within 2 seconds

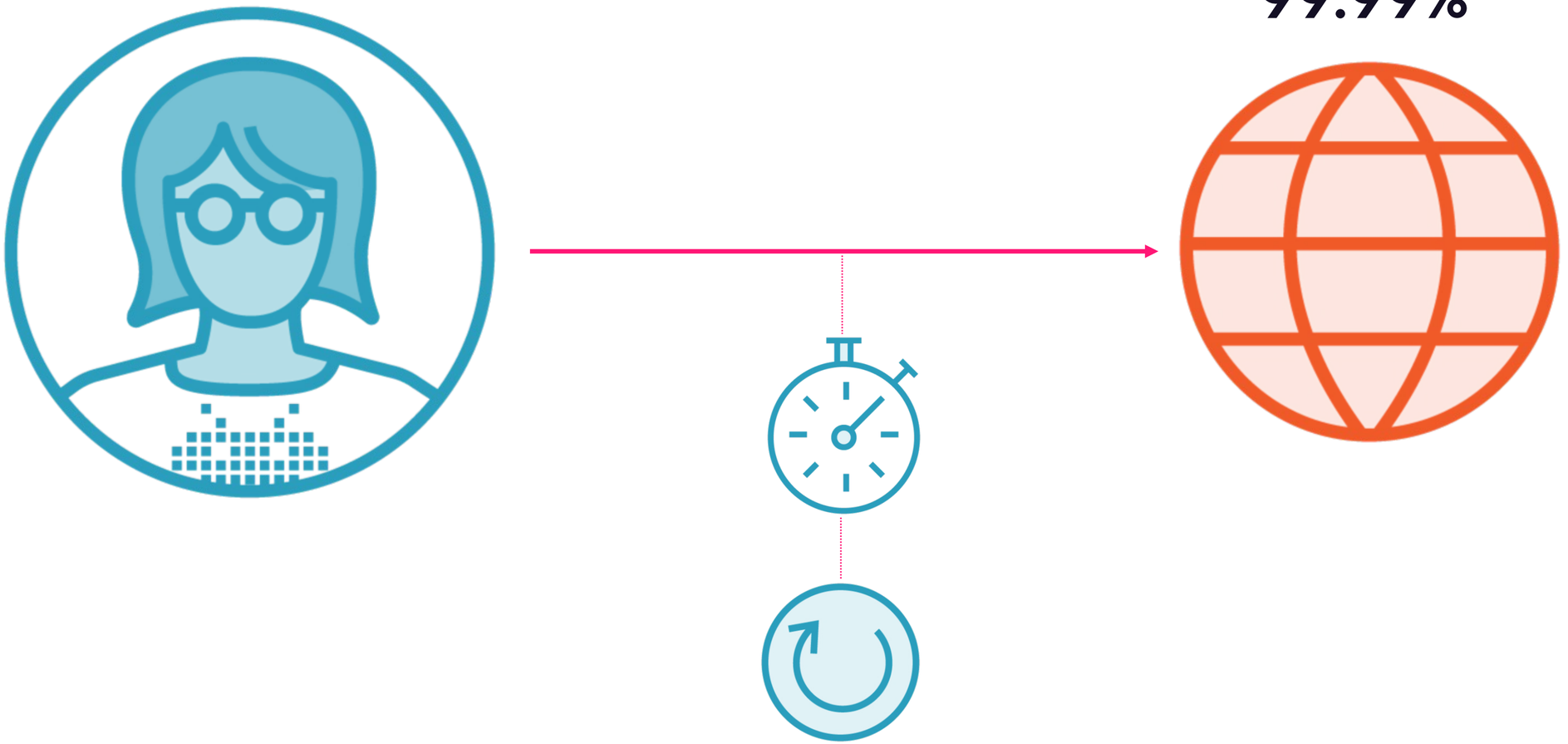
28 days =
40,320 minutes

SLO	*Time before breach*
99%	403 minutes (~ 7hours)
99.9%	40 minutes
99.99%	4 minutes
99.999%	0.4 minutes (~24 seconds)



99.9%


API
API
{}
99.99%
API
API
{}


99.99%

Service Level Objectives

Response time 99.9% of requests within 5 seconds

28 days

Normal operation
99.9% within 5 seconds
40,280 minutes

- Change window
- Product updates
- Configuration

Error budget
0.1% outside of 5 seconds
40 minutes

Service Level Objectives

Response time

97% of requests within 5 seconds

28 days

Normal operation

97% within 5 seconds

39,110 minutes

Error budget

3% outside of 5 seconds

1,210 minutes

Service Level Objectives

Response time 99.99% of requests within 5 seconds

28 days

Normal operation
99.99% within 5 seconds
40,316 minutes

Error budget
0.01% outside of 5 seconds
4 minutes

Error budget policy

- Formal document
- Agreed between product, dev and SRE

Enacted on SLA breach

- Prioritize reliability fixes
- Feature freeze - only reliability fixes
- Change freeze - only security patches

Contracted balance

- Change velocity
- Product reliability

Defining Service Level Indicators and Service Level Objectives

Service Level Measurement

Success rate
"Availability"

SLO: 99.9% of requests succeed
SLI: web server status codes

Response time
"Latency"

SLO: 90% of requests within 0.5 sec
SLI: web server response times

Service Level Indicators

		SLI
 **Availability**	5,000 requests 4,800 successful	96% successful
 **Latency**	5,000 requests ~~Average 1.5s~~ 4,000 within 0.5s 600 within 2s 300 within 5s	80% within 0.5s 92% within 2s 98% within 5s

Mapping SLIs to SLOs

		Current SLI	*Target SLO*
	**Availability**	5,000 requests 4,800 successful	96% successful	99% successful
	**Latency**	5,000 requests 4,000 within 0.5s 600 within 2s 300 within 5s	80% within 0.5s 92% within 2s 98% within 5s	90% within 0.8s 95% within 2s

Service Level Period

Response time

99.9% of requests within 5 seconds

14 days

Error budget consumption

Service Level Period

Response time

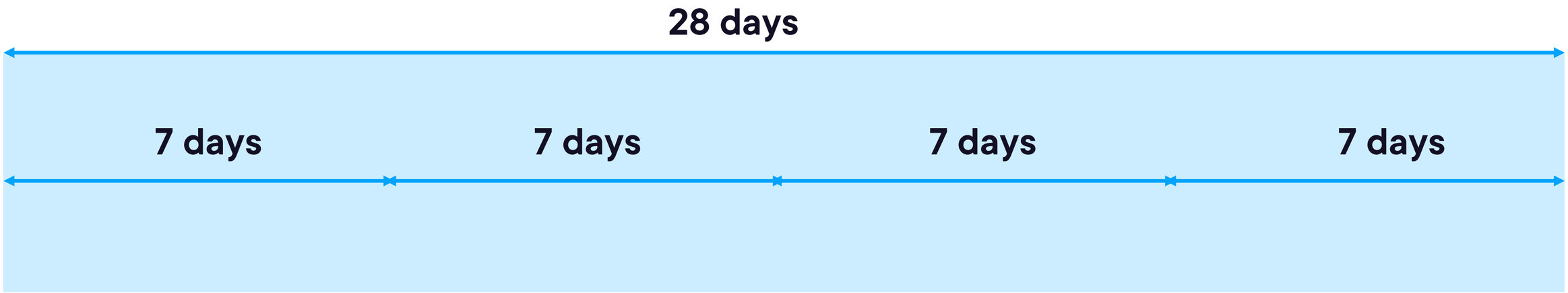
99.9% of requests within 5 seconds

84 days

Error budget consumption

Service Level Period

Response time 99.9% of requests within 5 seconds



Monitoring Service Level Indicators

Four Golden Signals

Latency

Job processing time

Response generation time

Traffic

Length of message queue

Requests per second

Errors

Request failures

Response correctness

Saturation

CPU & memory utilization

Network bandwidth

Implementing SLIs

Server Logs

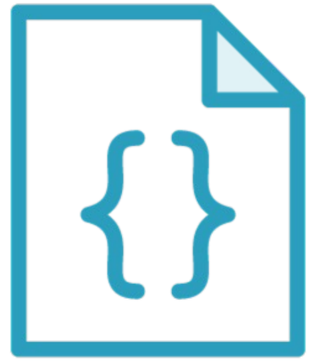
Response duration

No network time

Synthetic Testing

External services

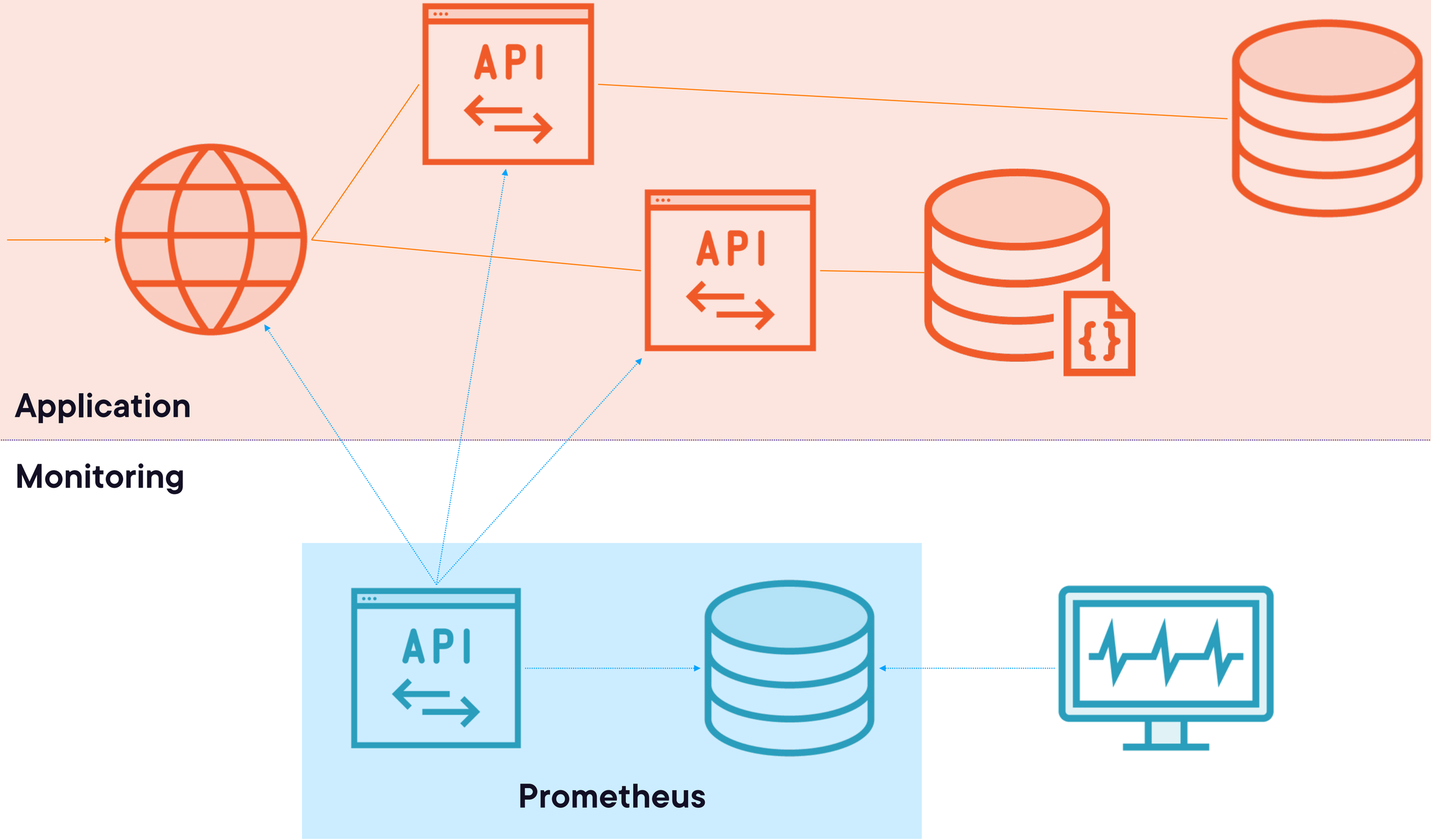
Pingdom/StatusCake

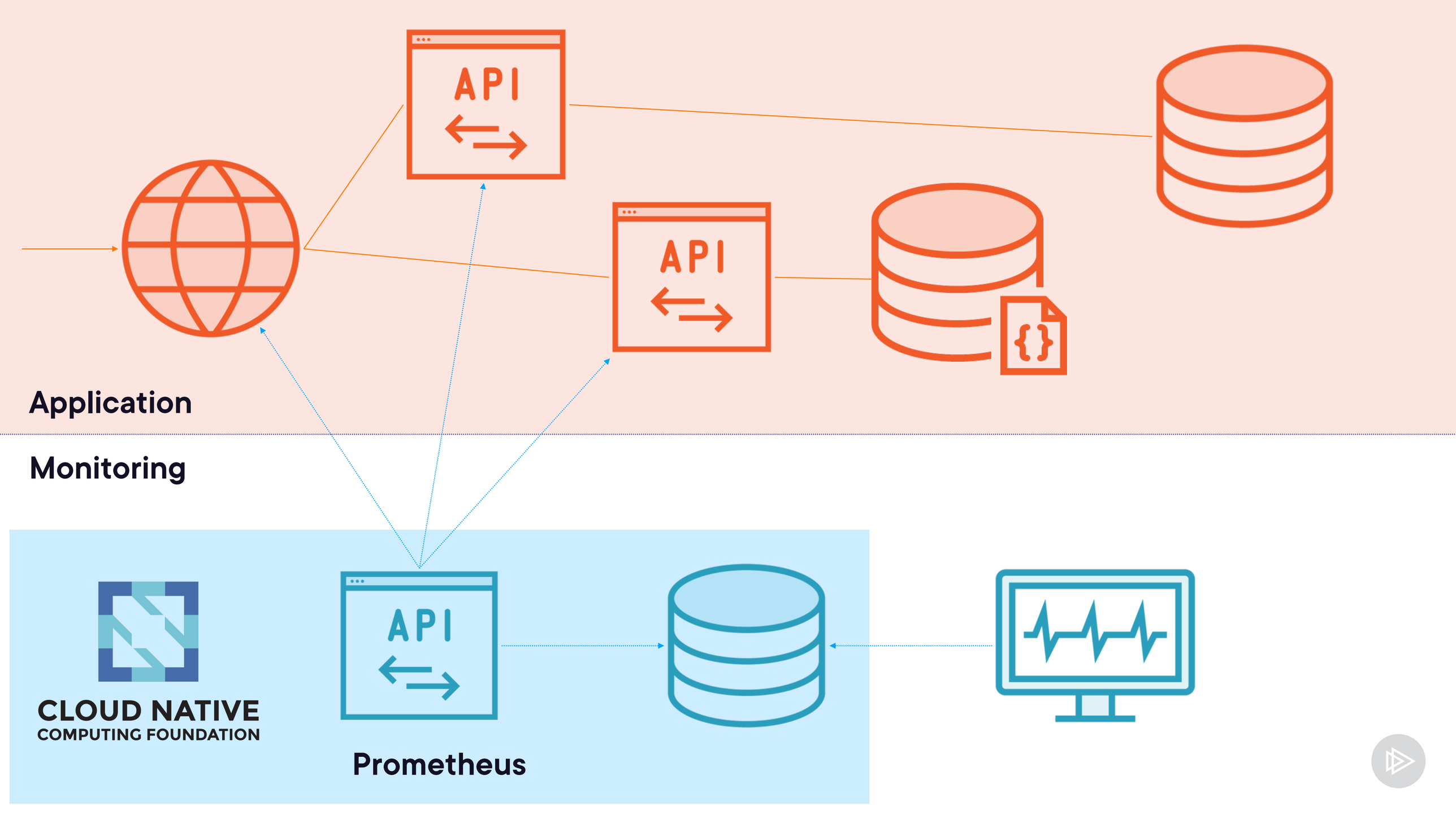


JavaScript Monitoring

Network load time

Browser rendering


API
API
{ }
Application
Monitoring
API
Prometheus


API
API
{}
Application
Monitoring
API
CLOUD NATIVE
COMPUTING FOUNDATION
Prometheus

Implementing Monitoring

Getting Started with Prometheus

Elton Stoneman

Metric	Labels	Value
http_requests_count	{statusCode="200"}	4600
http_requests_count	{statusCode="401"}	360
http_requests_count	{statusCode="500"}	18
http_requests_count	{statusCode="503"}	149

Total HTTP requests: 5,127

Total failures (5xx): 167

Metric	Labels	Value
latency_seconds_bucket	{le="0.25"}	200
latency_seconds_bucket	{le="0.50"}	4300
latency_seconds_bucket	{le="1.00"}	4700
latency_seconds_bucket	{le="5.00"}	5000

50th percentile = *PromQL query*

90th percentile = *PromQL query*

Dashboards

- Visualize SLIs
- Four golden signals

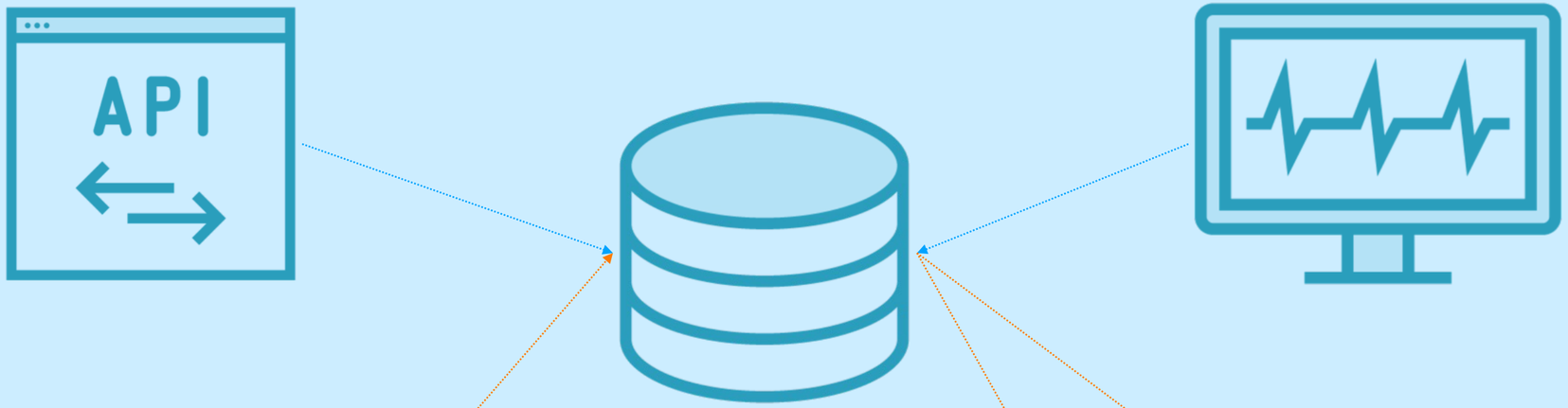
Textual information

- Release versions
- Configuration timestamps

Analysis head-start

- SLO under threat
- Memory saturation high
- Since latest release?

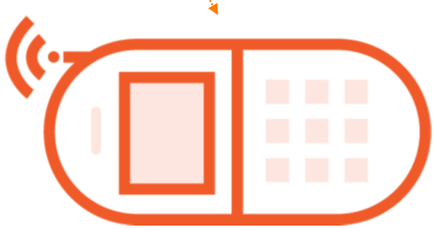
Alerting on Service Level Objectives



Monitoring

Alerting

- Timeliness
- Accuracy
- Signal-to-noise



Precision

- Only trigger on significant events

Recall

- Trigger on all significant events

Detection time

- Time to trigger the alert

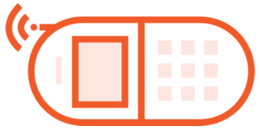
Reset time

- Time to stop alerting

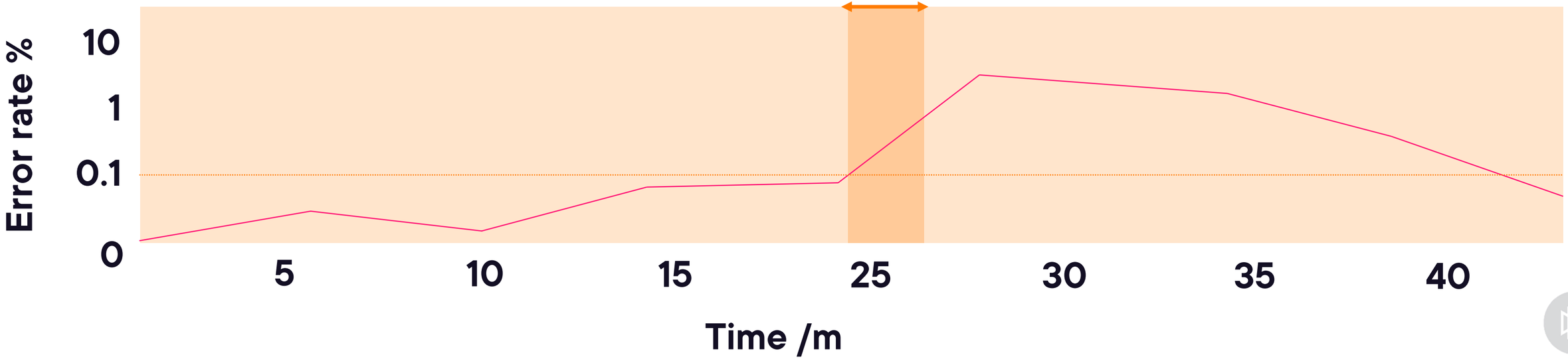
Alerting on SLO Threshold

Availability

SLO: 99.9% success rate



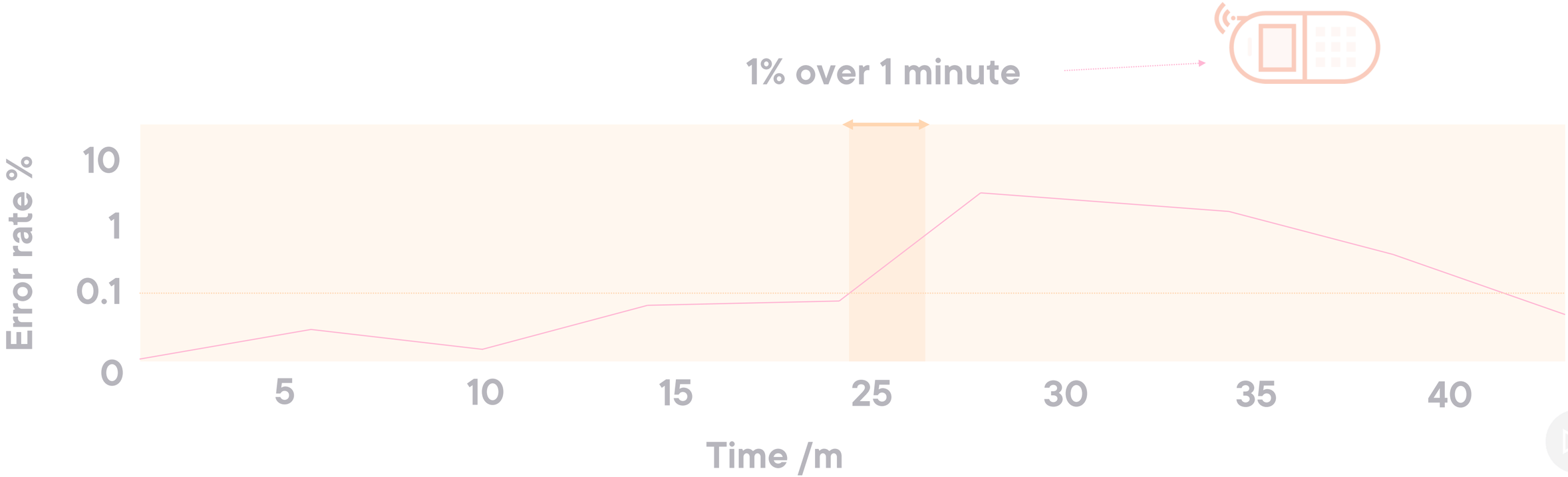
1% over 1 minute



Alerting on SLO Threshold

Availability SLO: 99.9% success rate

Period	28 days
Expected load	1M requests
SLO window	10,000 errors



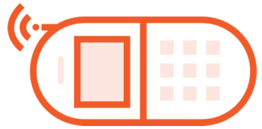
Alerting on SLO Threshold

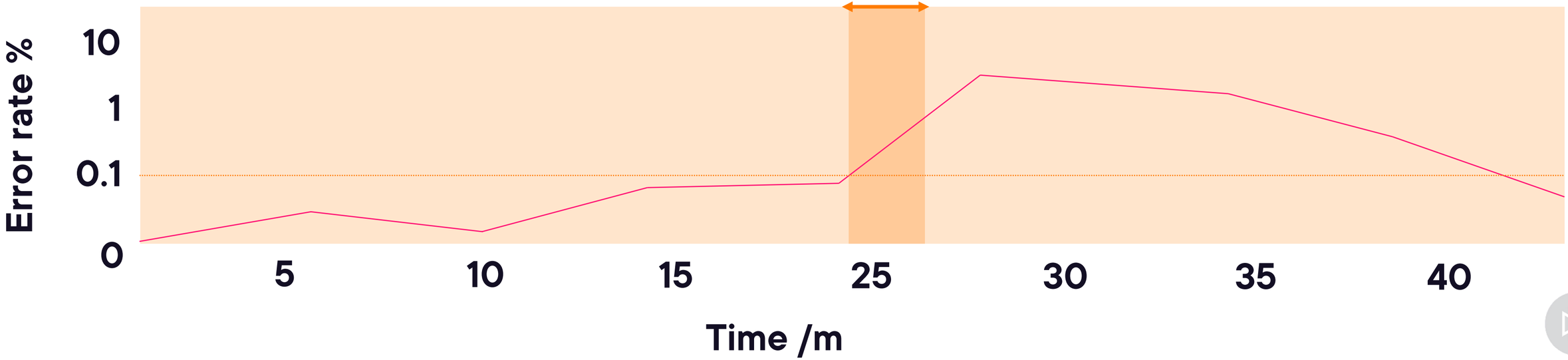
Availability

SLO: 99.9% success rate

Period	28 days
Expected load	1M requests
SLO window	10,000 errors

1% over 1 minute
= 1% of 25 req /s
= 2.5 requests



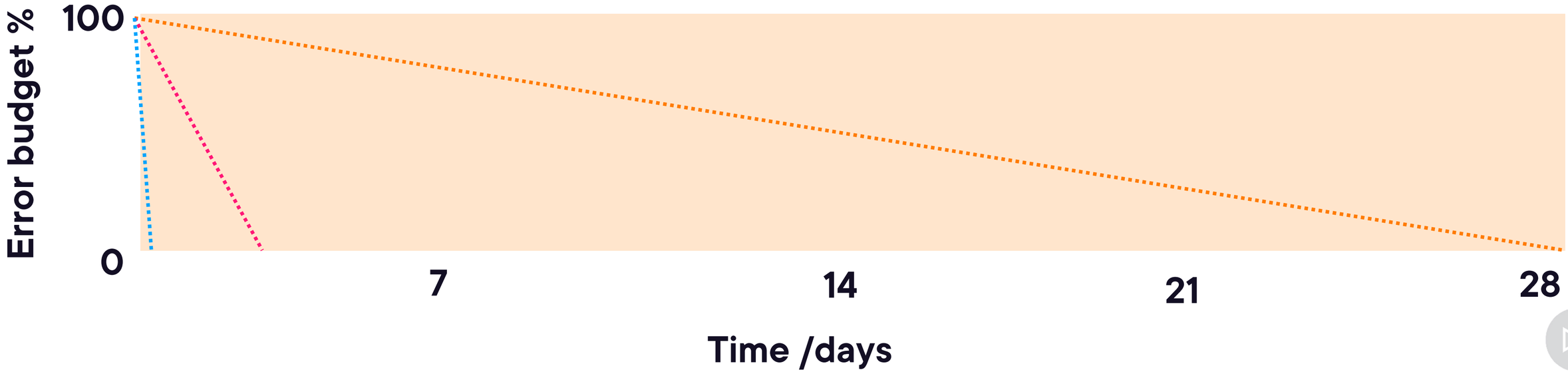


Alerting on Error Budget Burn

Availability

SLO: 99.9% success rate

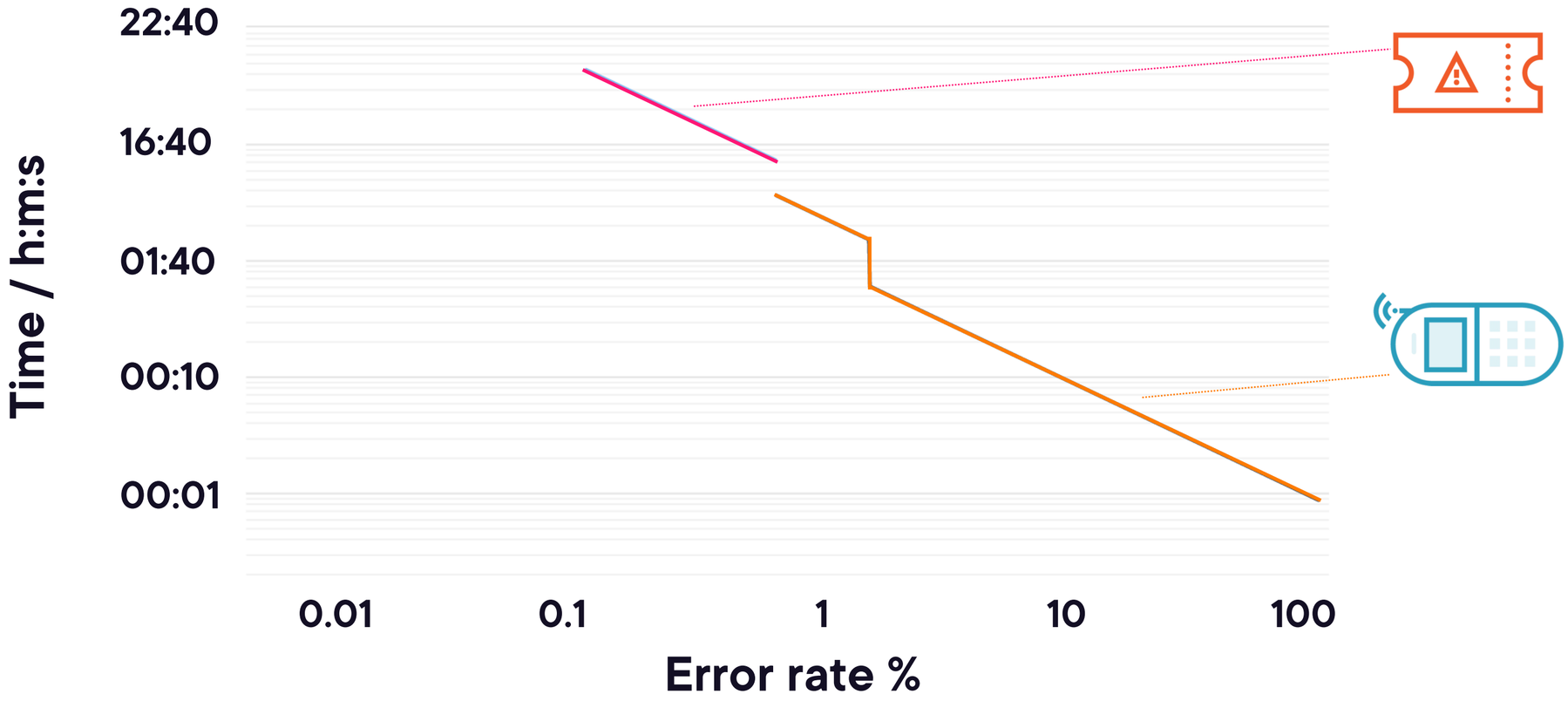
Error rate	*Burn rate*
0.1%	1
1%	10
100%	1,000



Alerting on Error Budget Burn

Availability

SLO: 99.9% success rate



Multiple Burn Rate Calculator

Source code

SLO
99.9

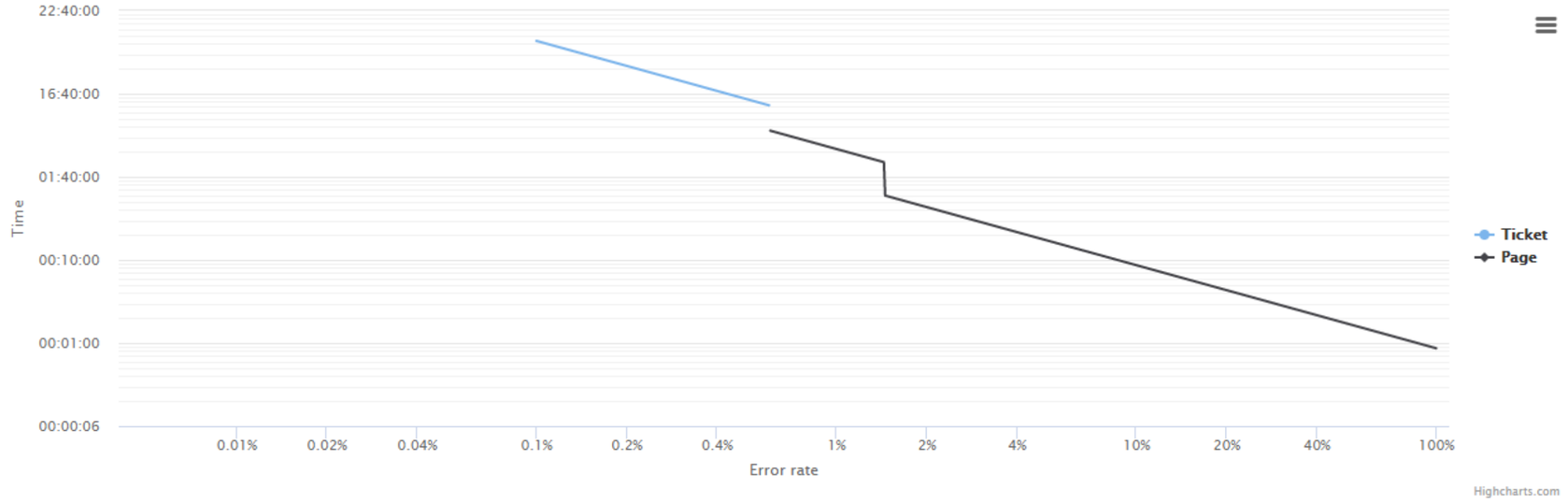
SLO Window (days)
30

Error Budget Consumption thresholds

Time	Budget consumption (%)	Burn rate
1h	2	14.4
6h	5	6
1d	10	3
3d	10	1

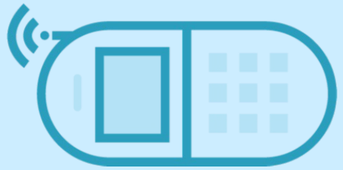
https://is.gd/1oILuV

Detection time



Source: is.gd/mJwaoa

```
expr: (job:slo_errors_per_request:ratio_rate1h{job="x"} > (14.4*0.001)
      and
       job:slo_errors_per_request:ratio_rate5m{job="x"} > (14.4*0.001))
    or
      (job:slo_errors_per_request:ratio_rate6h{job="x"} > (6*0.001)
      and
       job:slo_errors_per_request:ratio_rate30m{job="x"} > (6*0.001))
severity: page
```



```
expr: (job:slo_errors_per_request:ratio_rate24h{job="x"} > (3*0.001)
      and
       job:slo_errors_per_request:ratio_rate2h{job="x"} > (3*0.001))
    or
      (job:slo_errors_per_request:ratio_rate3d{job="x"} > 0.001
      and
       job:slo_errors_per_request:ratio_rate6h{job="x"} > 0.001)
severity: ticket
```

The SLO error rate is 0.1%.

Owners get paged if their backend has returned more than 1.44% 5xx responses over the last 1h and over the last 5m.

They also get paged if it has returned more than 0.6% 5xx responses over the last 6d and over the last 30m.

Björn Rabenstein, SoundCloud `https://is.gd/EhtIPT`

Summary

Monitoring with service levels

- Service Level Objectives
- Service Level Indicators

Error budget policy

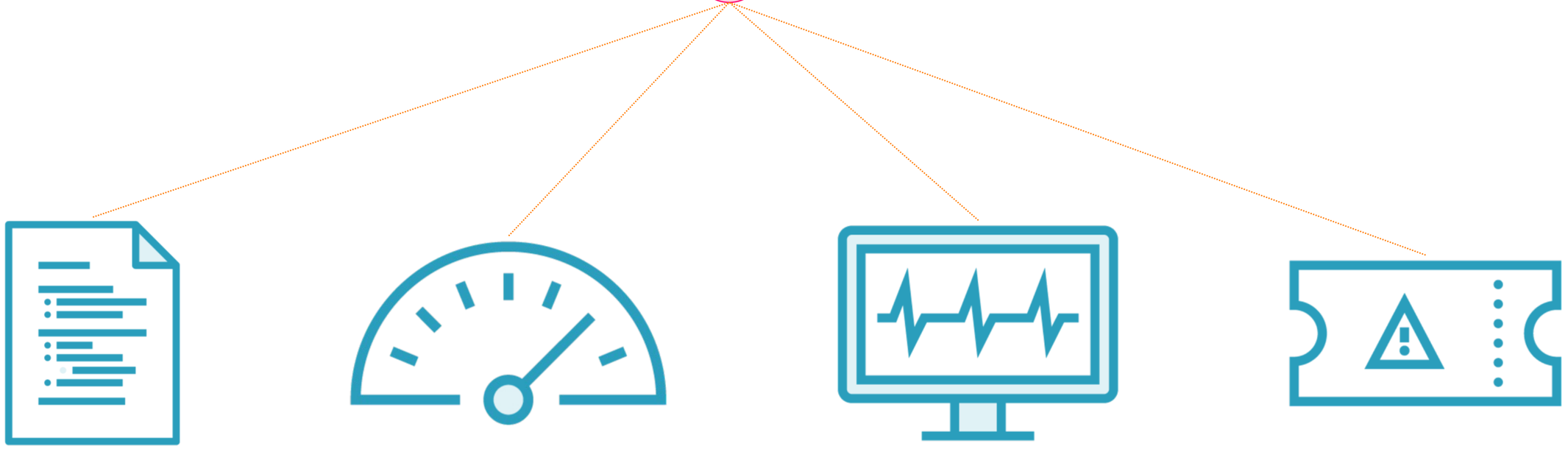
- Agreed between business, dev & SRE
- Enacted if SLOs are not met
- Feature or full change freeze

Alerting

- Parameters for timeliness & accuracy
- Error budget burn rate
- Multiple time windows

Response time

SLO: 99.9% of requests within 5 seconds



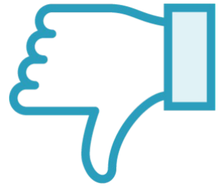
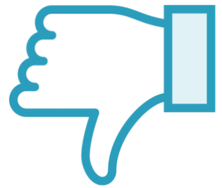
SLO Review

Accuracy

User Experience

Toil

SLO Review

SLO Met	User Satisfaction	SRE Toil	Action
✓			Relax SLO
✗			Relax SLO
✗			Product design

Enhanced SLOs

Success rate 99.9% of requests have 2xx response

Response time 95% of requests within 2 seconds

Enhanced SLOs

Success rate

99.9% of requests have 2xx response

Response time

95% of requests within 2 seconds

95% of homepage within 0.7 seconds

95% of search within 1.5 seconds

Enhanced SLOs

Success rate

99.9% of requests have 2xx response

99.99% of checkouts have 2xx response

Response time

95% of requests within 2 seconds

95% of homepage within 0.7 seconds

95% of search within 1.5 seconds

Up Next:

Incident Management: On-call and Postmortems