Driving Continuous Improvement with Service Levels

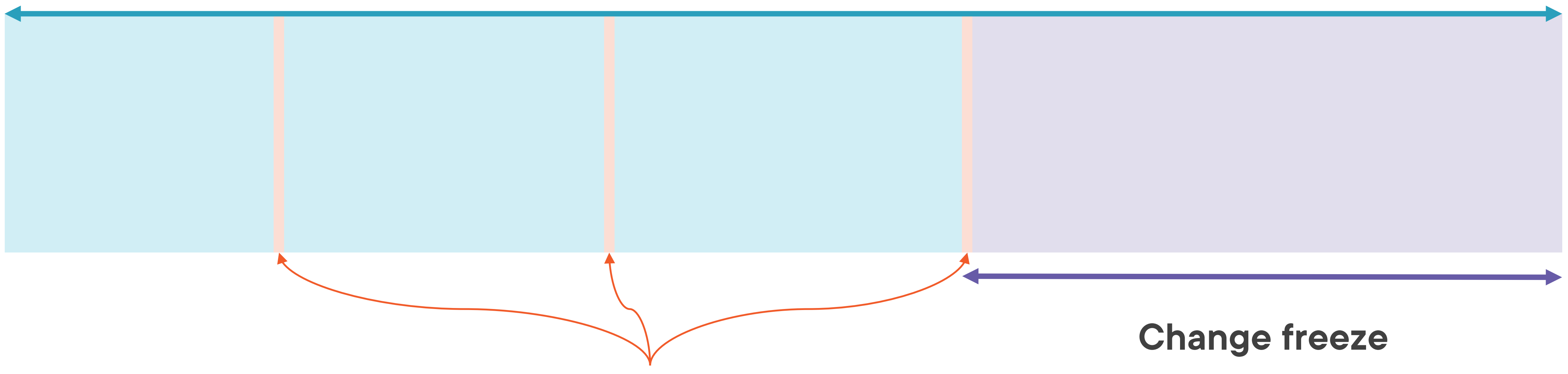
Elton Stoneman
Consultant & Trainer

@EltonStoneman   blog.sixeyed.com

Service Level Objectives

Response time **99.9% of requests within 500ms**

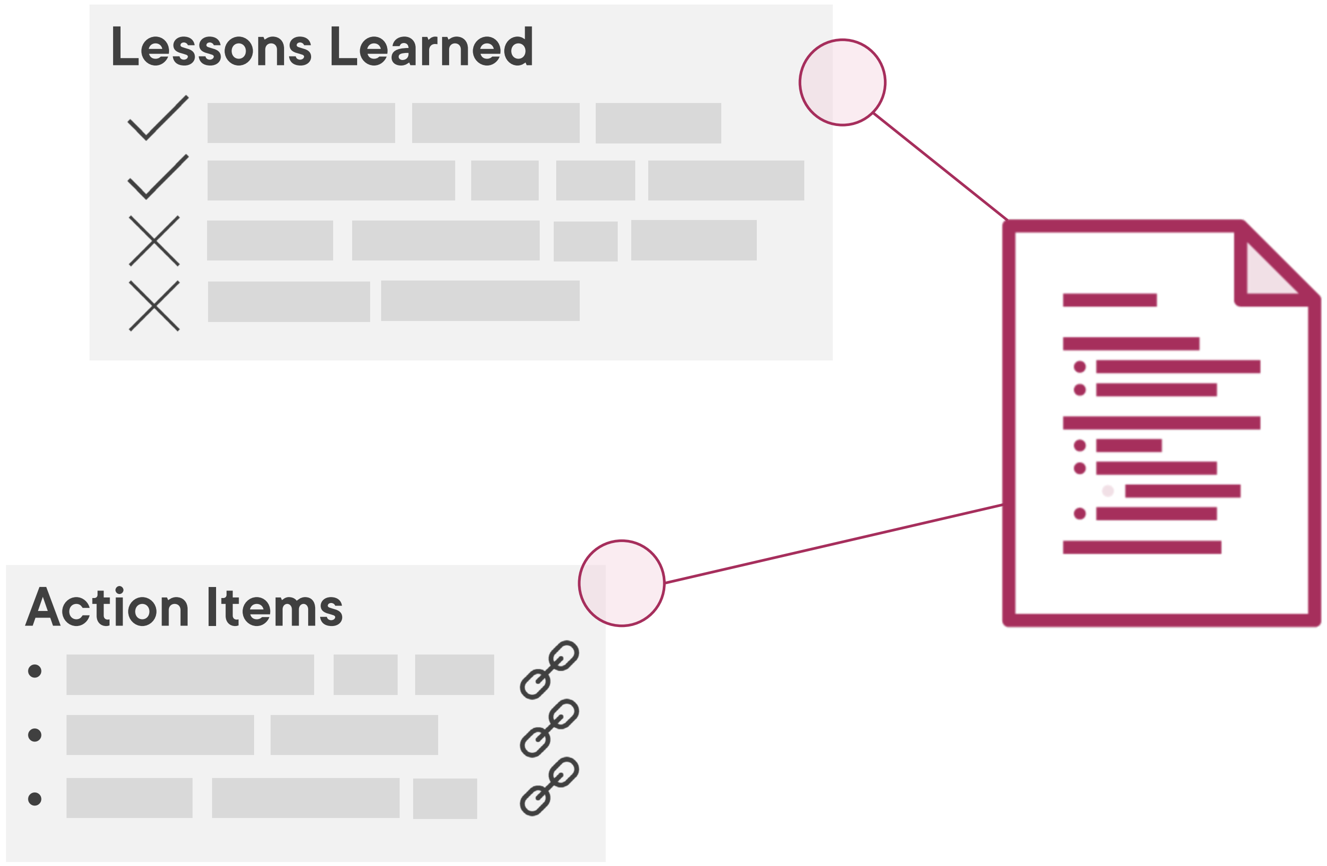
14 days = 20,160 minutes | Error Budget = 20 minutes



8 minutes @ 600ms

- Baseline performance borderline
- Release degrades performance

- Product optimization
- Pipeline optimization



Service Levels

SLI

Service Level
Indicator

SLO

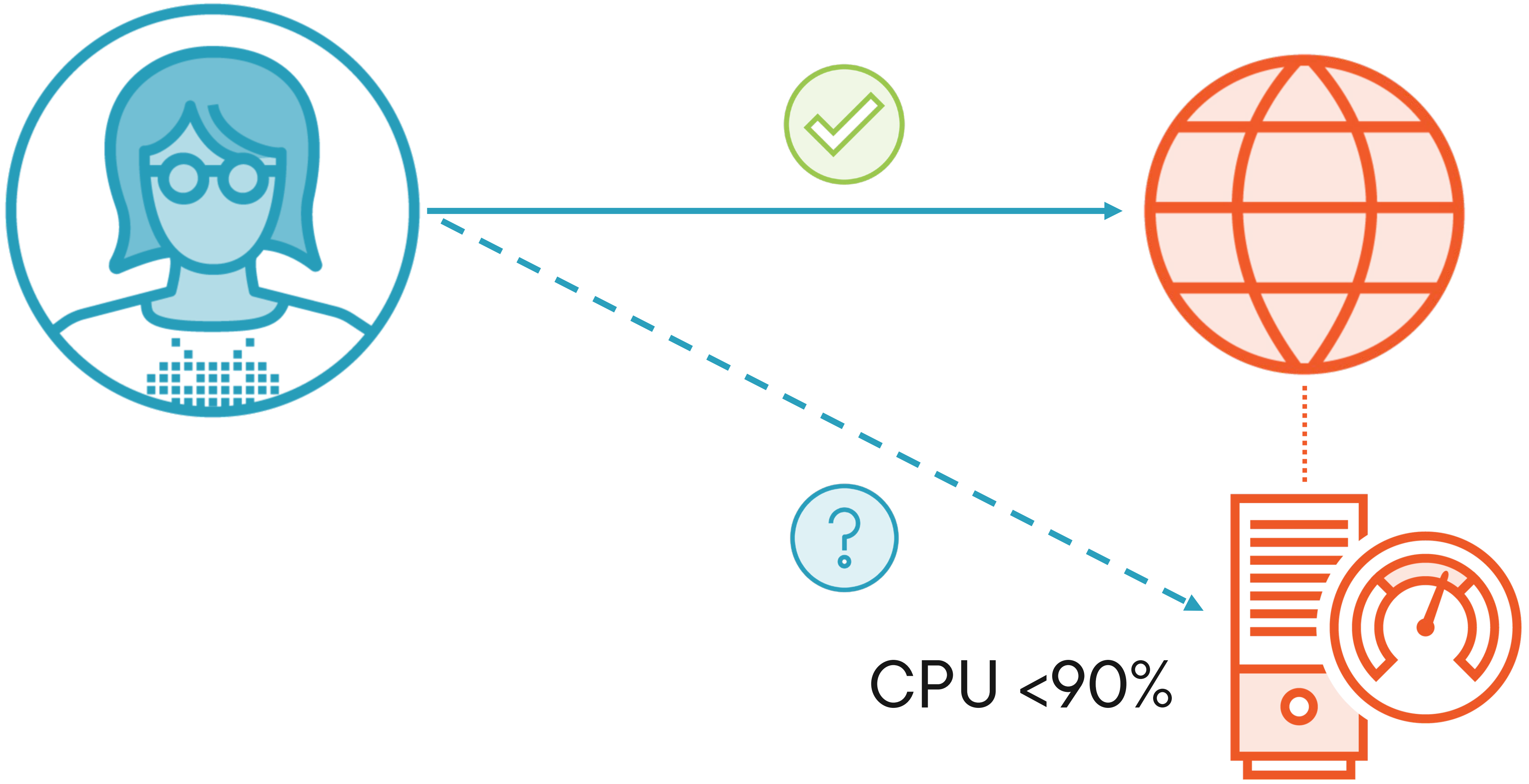
Service Level
Objective

SLA

Service Level
Agreement

SLI	SLO	SLA
X is true	*for Y period of time*	*... or else*

Matthew Flaming, New Relic


CPU <90%

Designing SLOs

Customer journeys

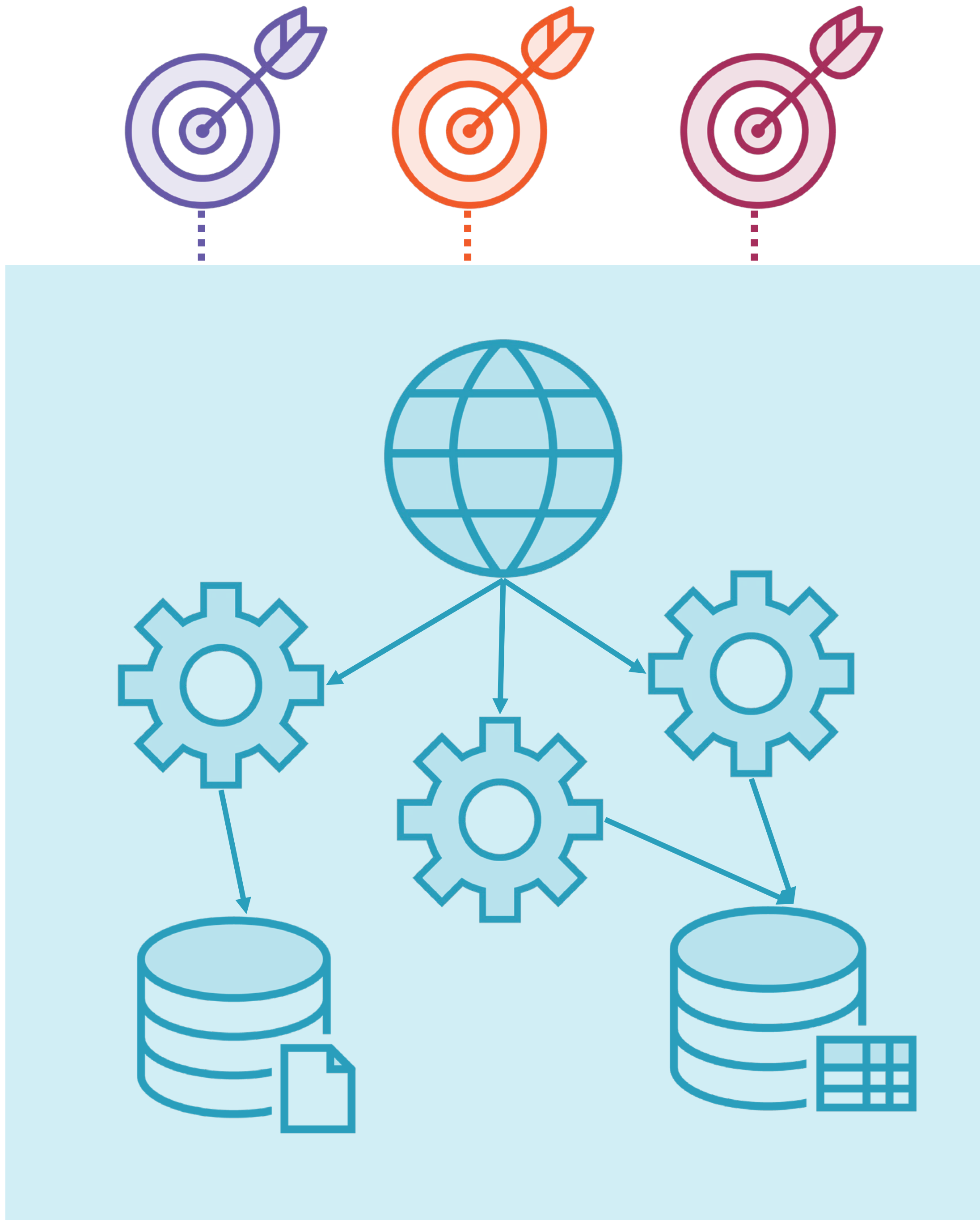
Improvable targets

Measure breaches

GLOBOTICKET

A Globomantics Company

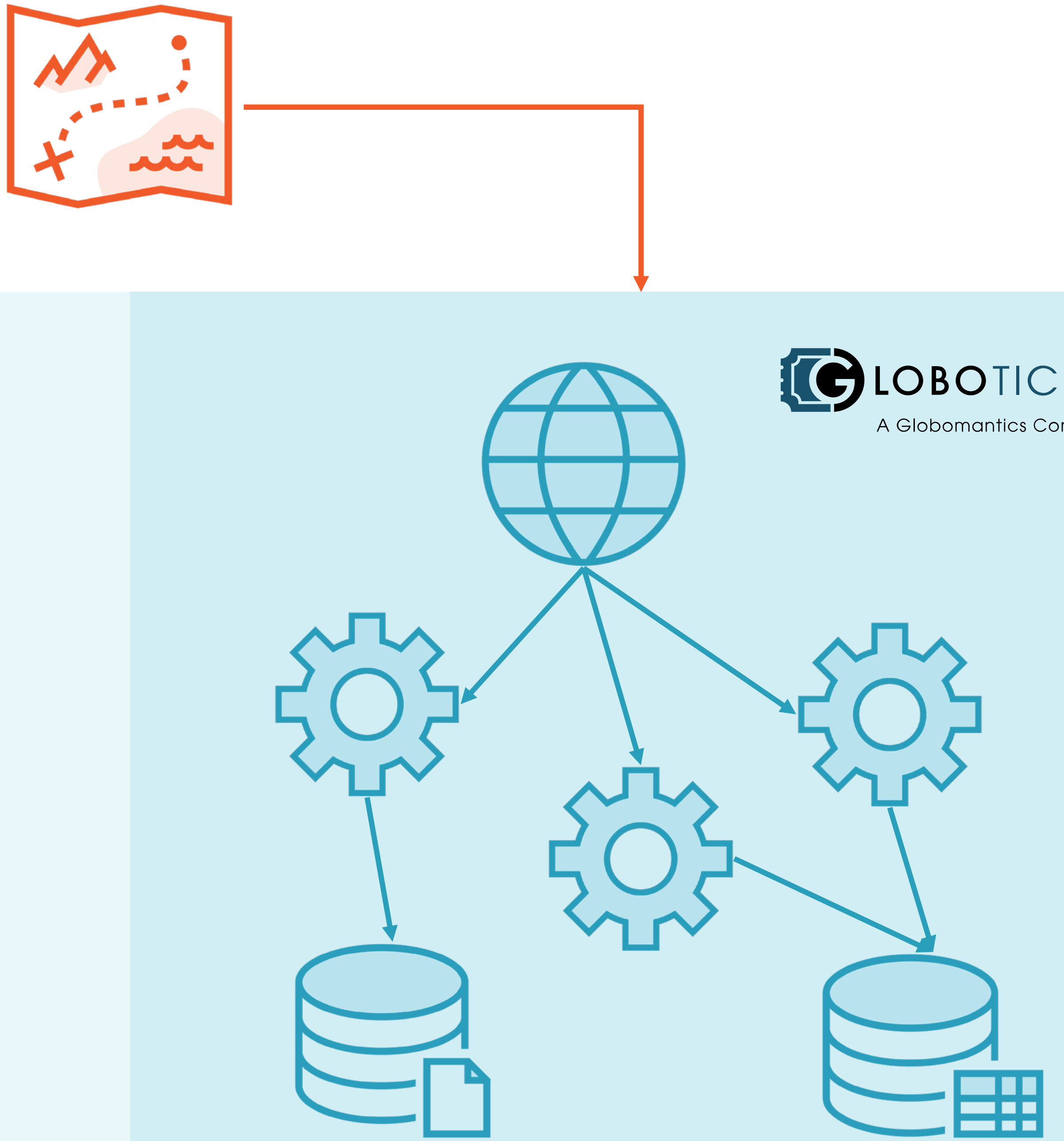
Amila

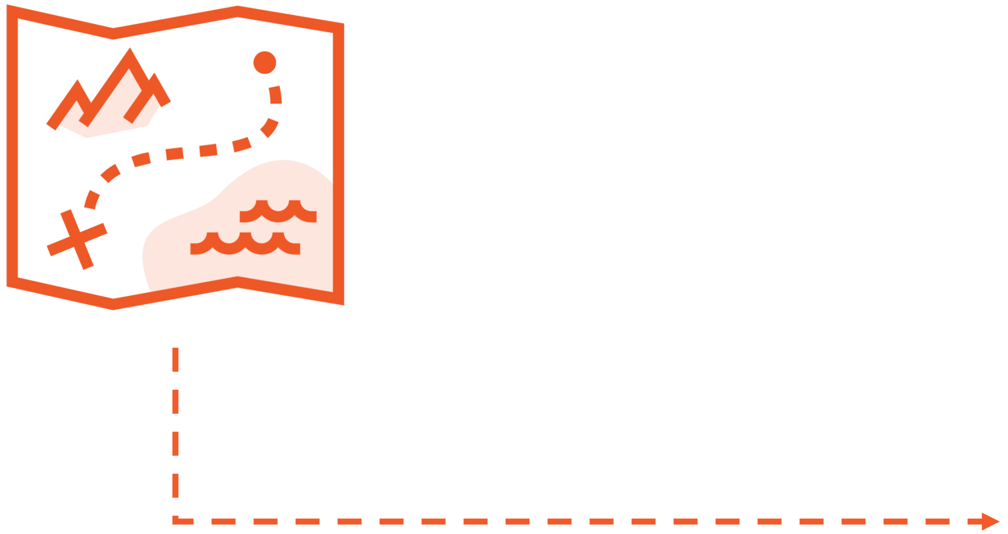


Scenario: Designing SLOs and SLIs

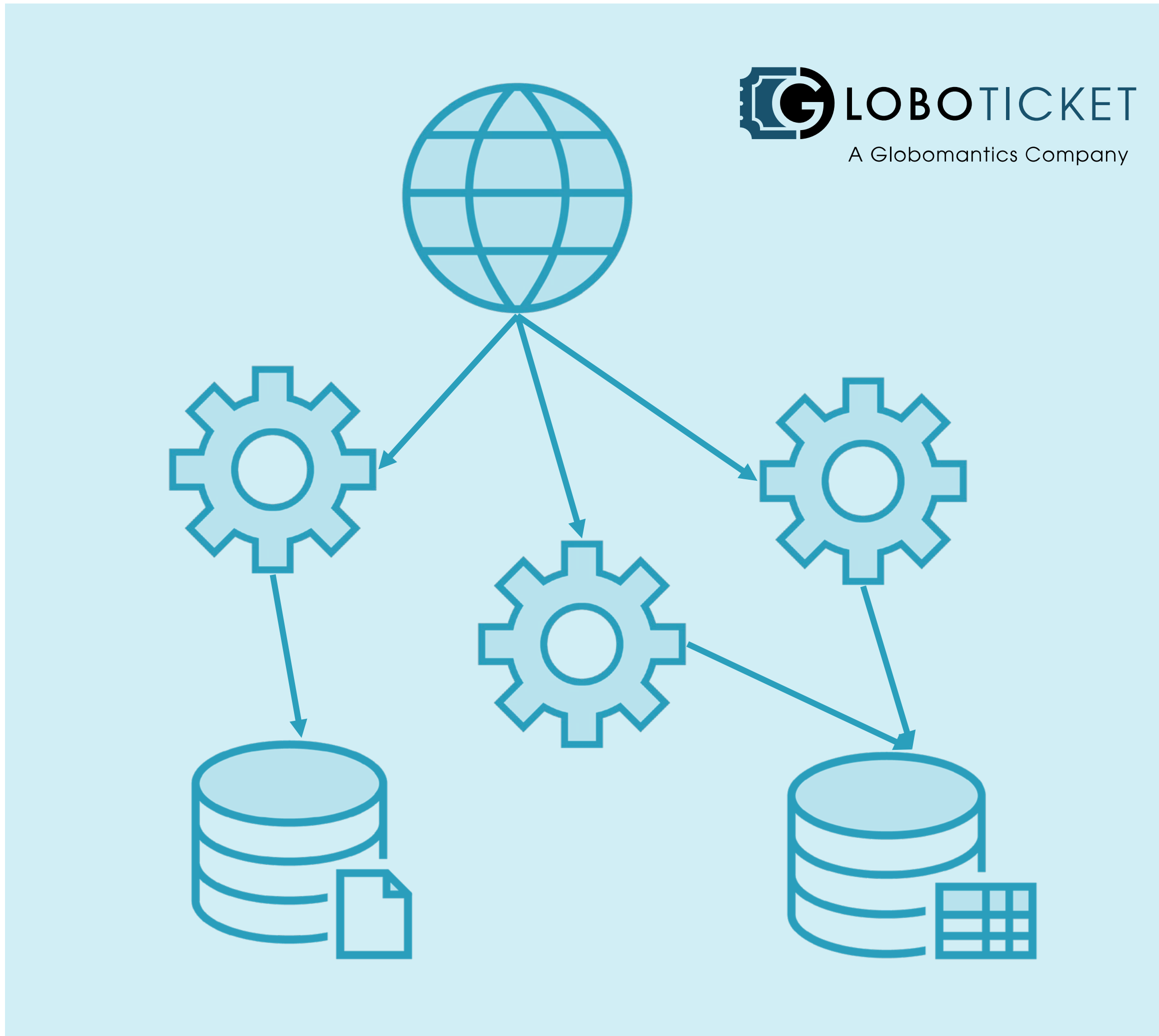
Amila

What are the most important user journeys?





- Event search
- Event details
- Reserve tickets
- Buy tickets

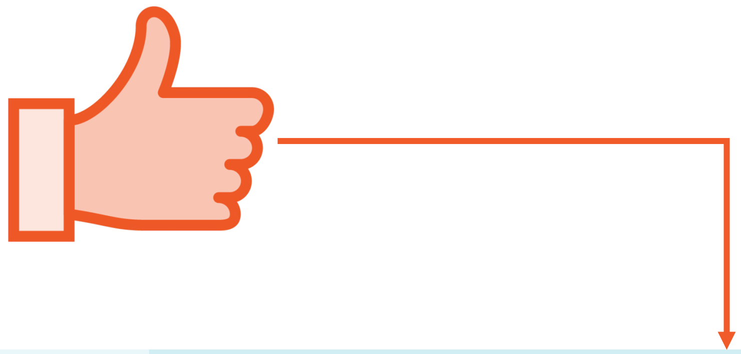


Prioritized User Journeys

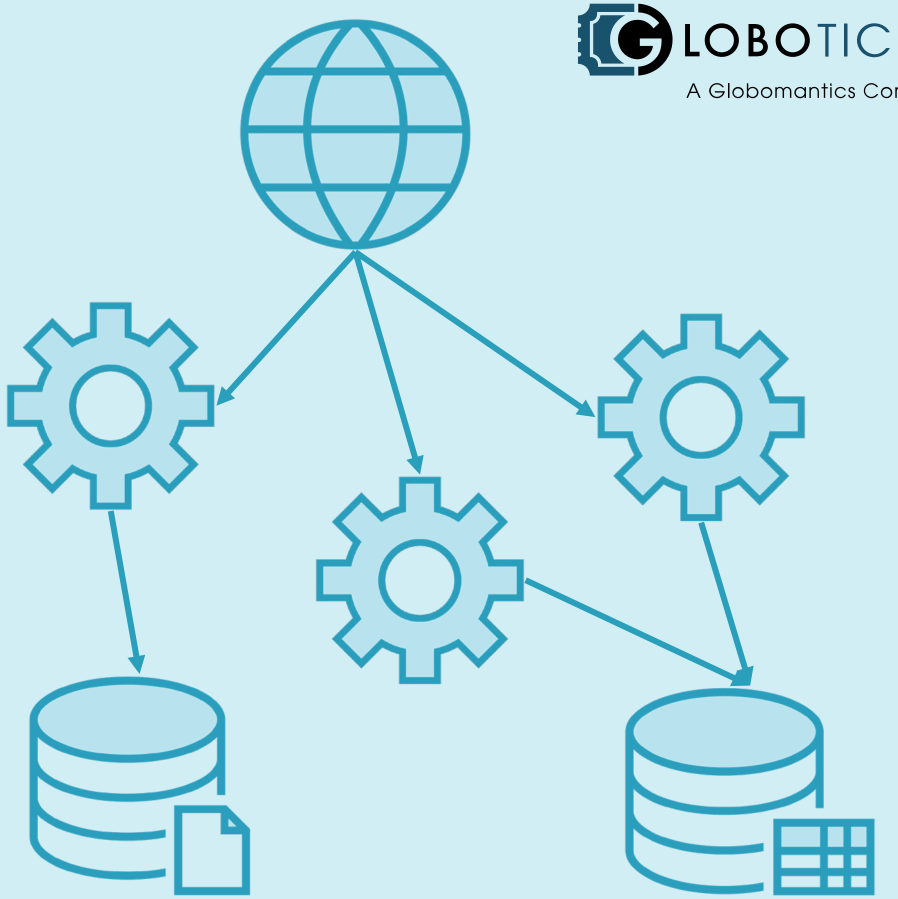
		Customer Impact	*Business Impact*
#1	**Buy tickets**	☆☆☆	☆☆☆
#2	**Event search**	☆☆☆	☆☆
#3	**Event details**	☆☆	☆
#4	**Reserve tickets**	☆	☆

Prioritized SLOs

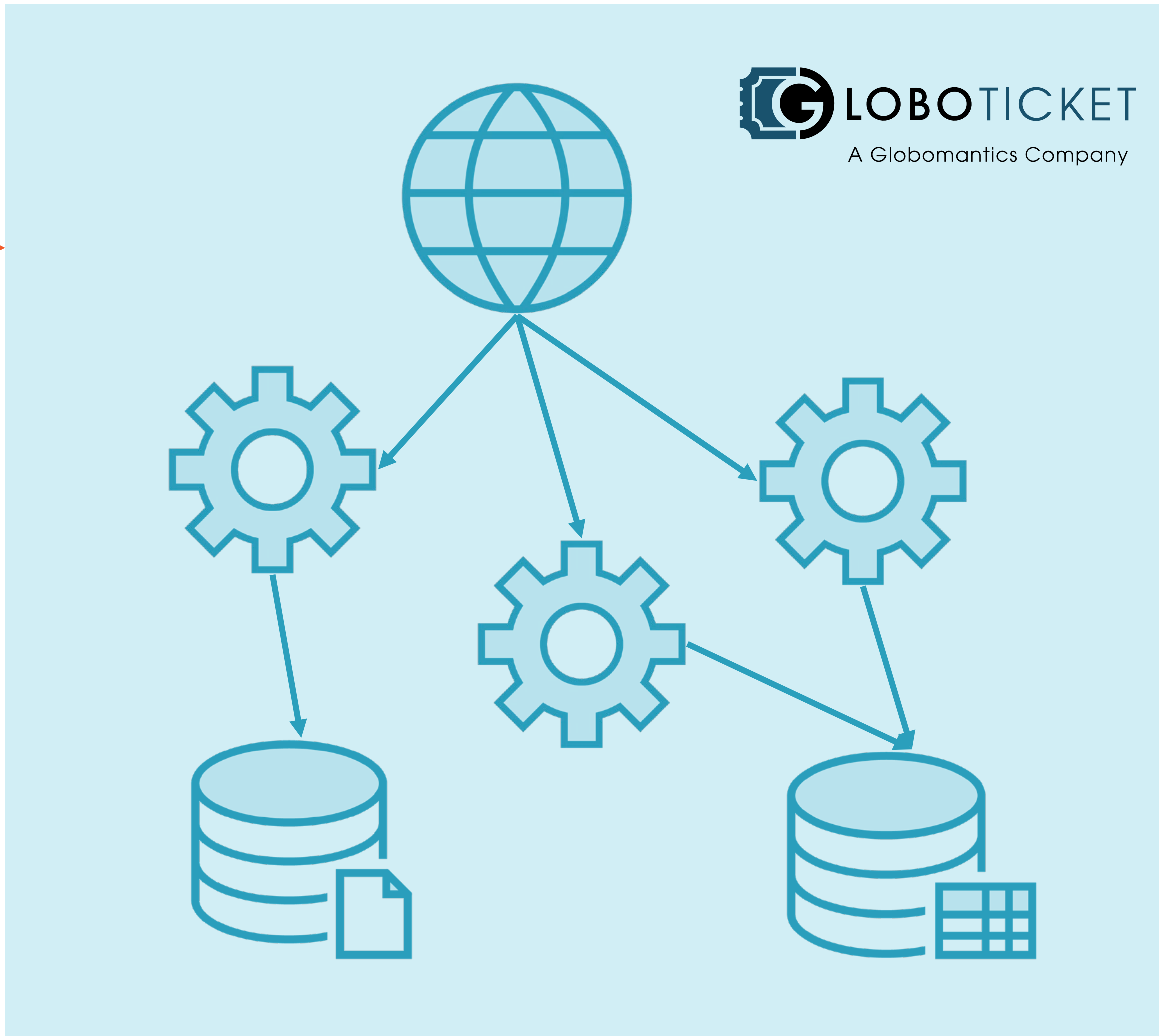
		Customer Impact	*Business Impact*
#1	**Buy tickets**	☆☆☆	☆☆☆
#2	**Event search**	☆☆☆	☆☆
#3	**Event details**	☆☆	☆
#4	**Reserve tickets**	☆	☆



What is a "good" buy-ticket experience?

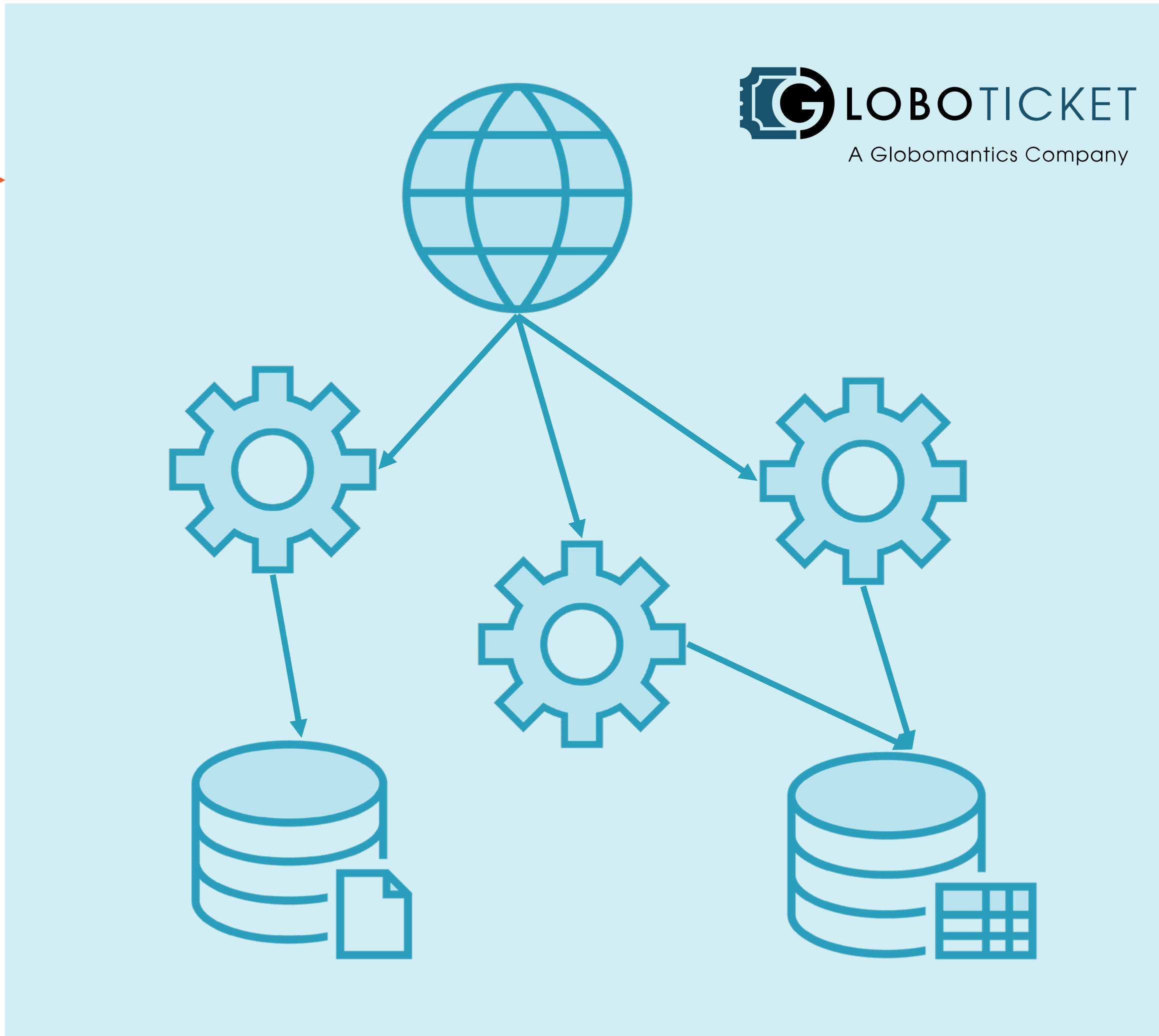


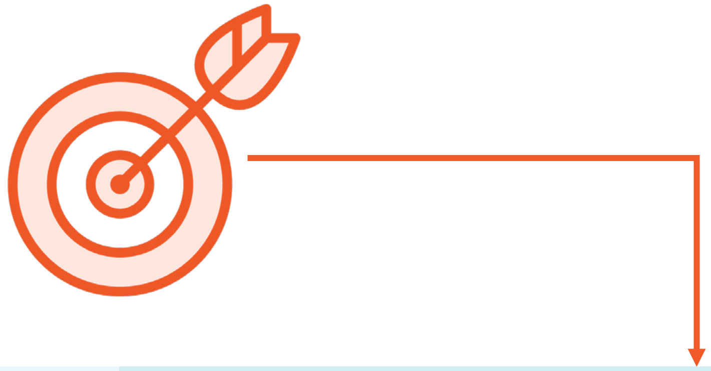
GLOBOTICKET
A Globomantics Company



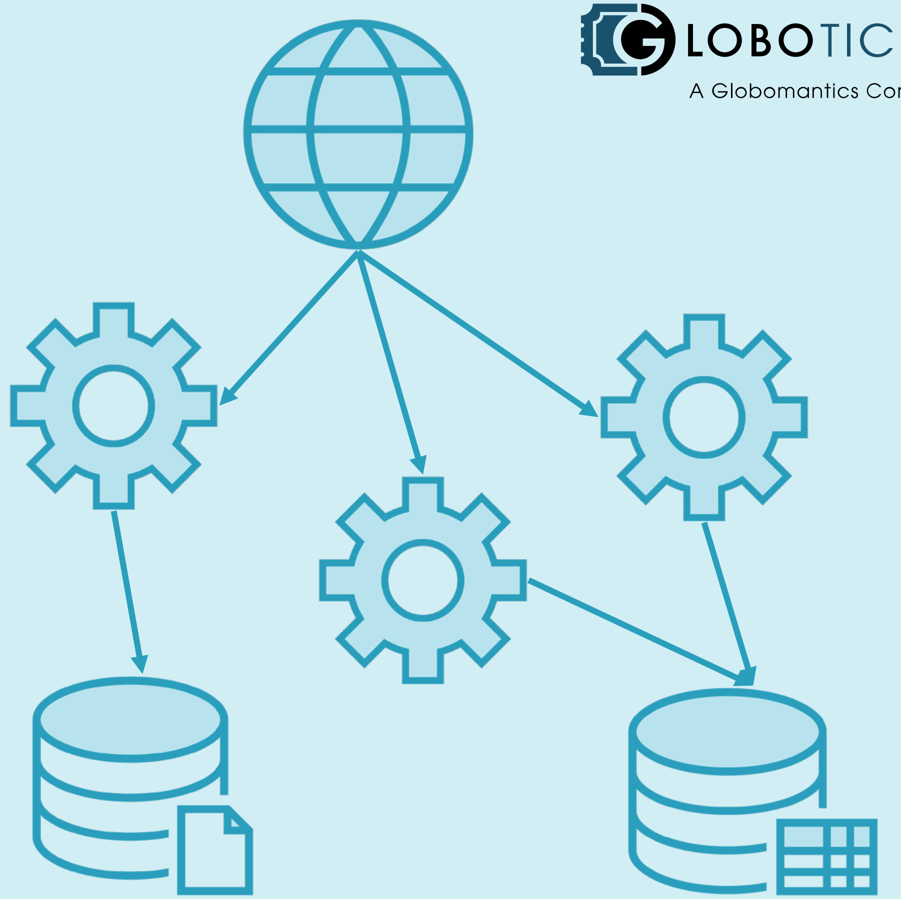
- Correctly and quickly
- Latency & status tracked
- Histogram for latency

- 95th percentile
- Service performance
- Client performance



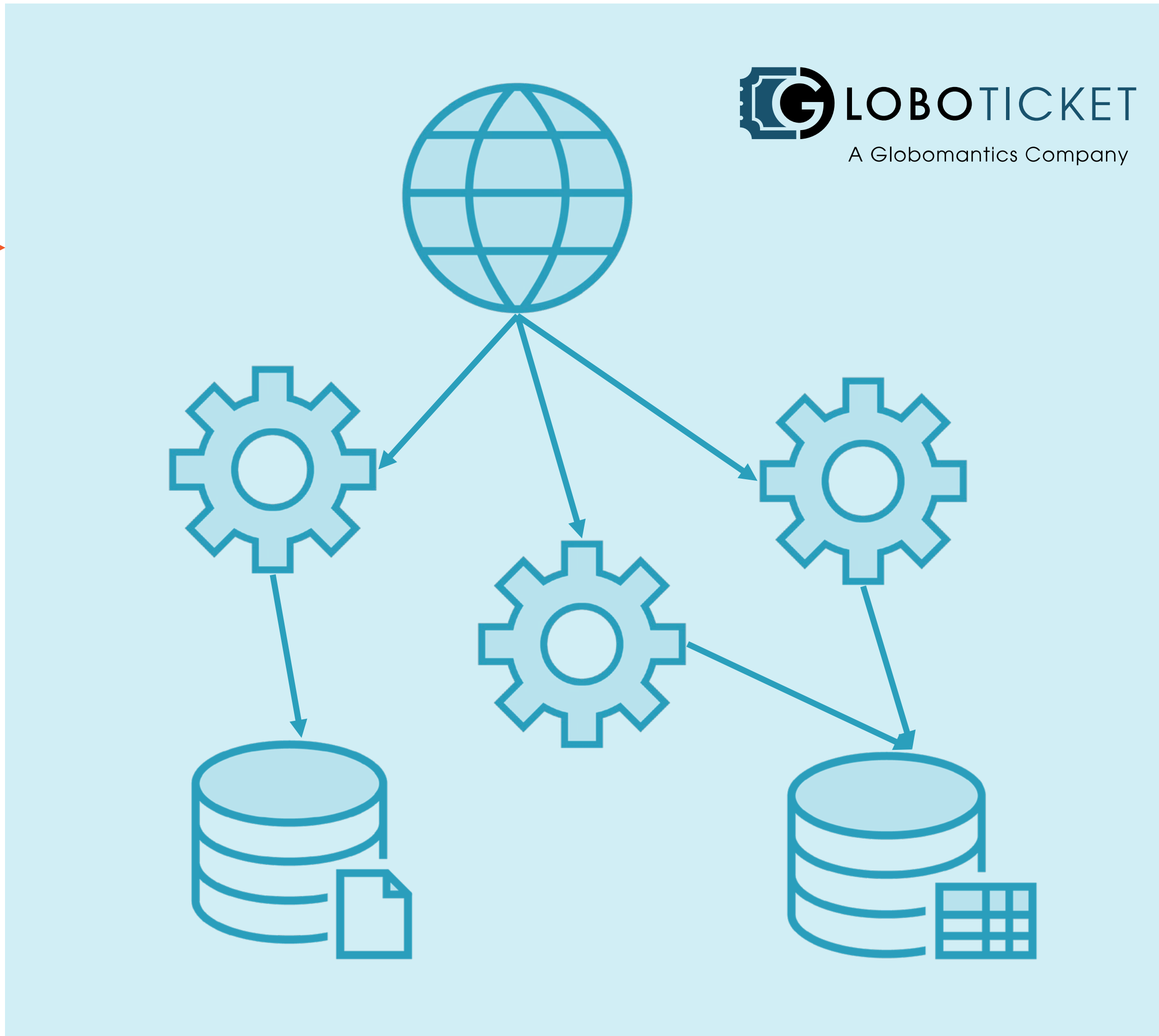


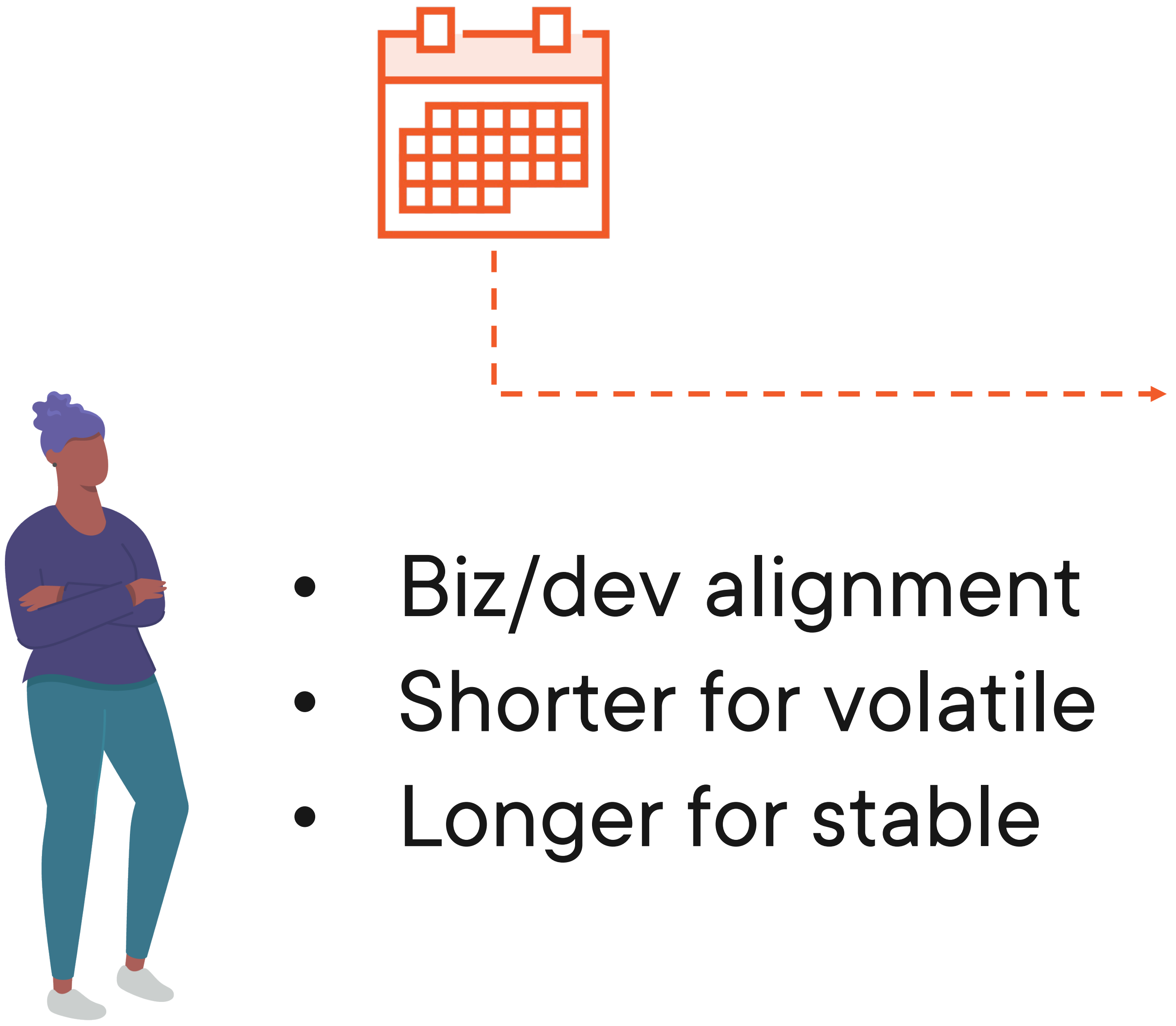
What SLO is achievable and desirable?



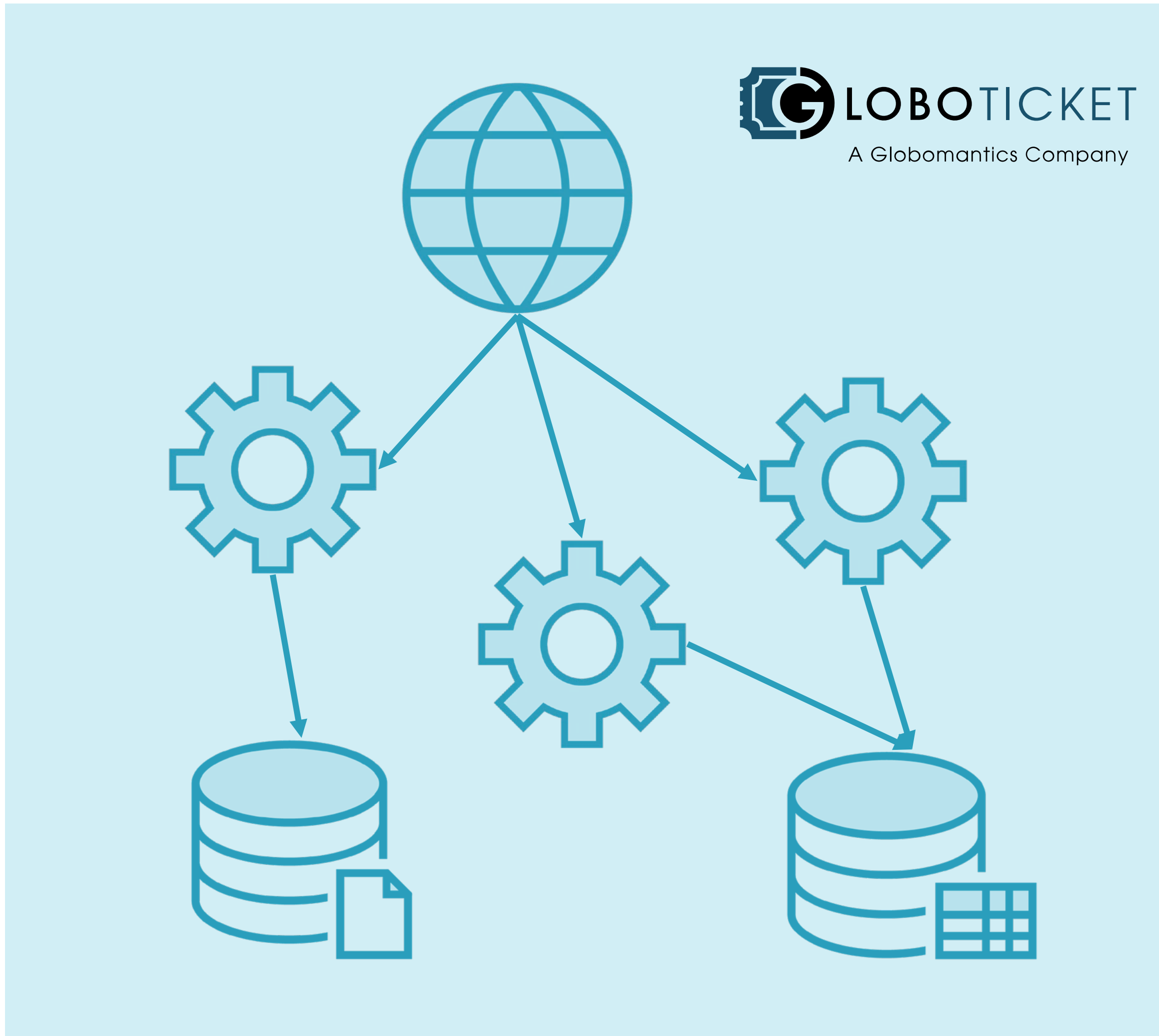
GLOBOTICKET
A Globomantics Company

- Data-driven targets
- 99.9% success rate
- 500ms latency





- Biz/dev alignment
- Shorter for volatile
- Longer for stable



99.9% of ticket purchases
complete **successfully**
within **500ms**
over a **one-month** period

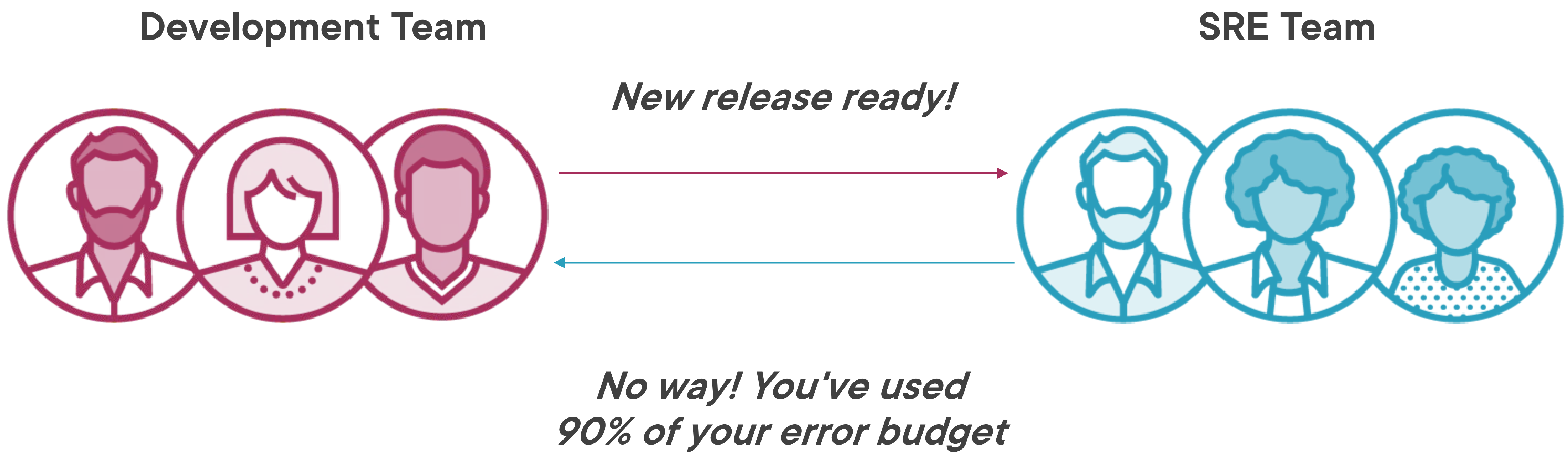
Understanding the Goals of SLOs

Reactive SLOs

"Just about" good enough

Minimizing intervention

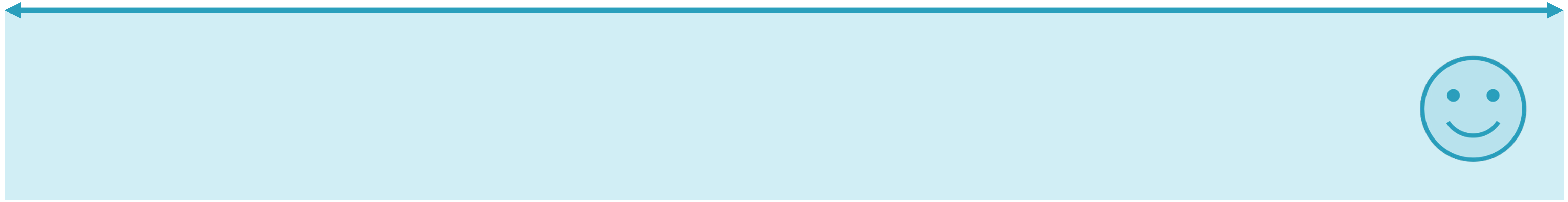
Hiding underlying issues


Development Team
SRE Team
New release ready!
No way! You've used 90% of your error budget

Reactive SLOs

Response time **99% of home page requests within 900ms**

14 days = 20,160 minutes | Error Budget = 201 minutes



No breaches for 427 weeks!

Proactive SLOs

Positive customer experience

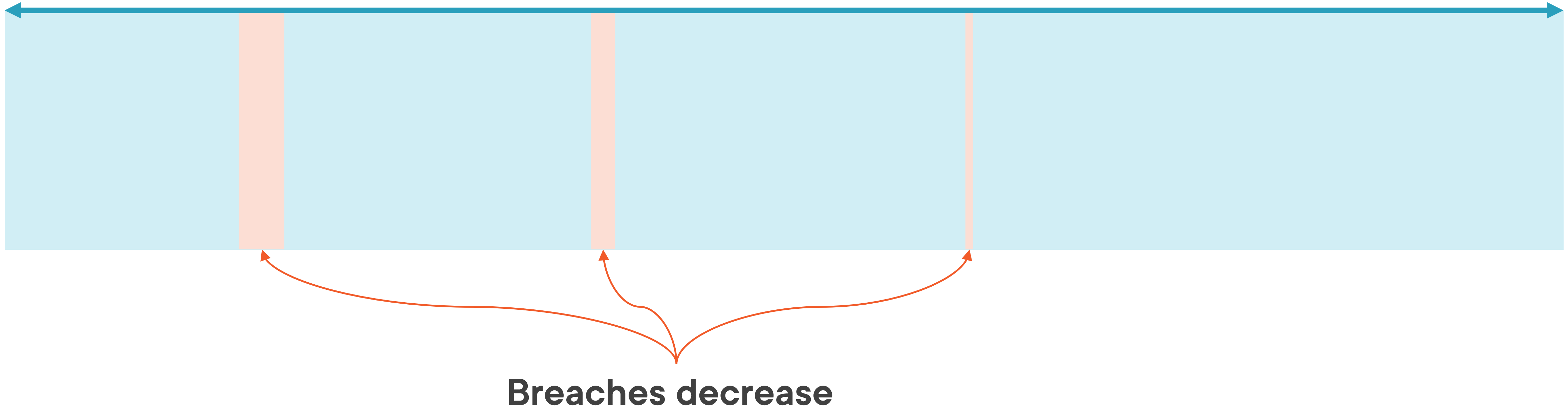
Error-budget tracking

Driving improvement

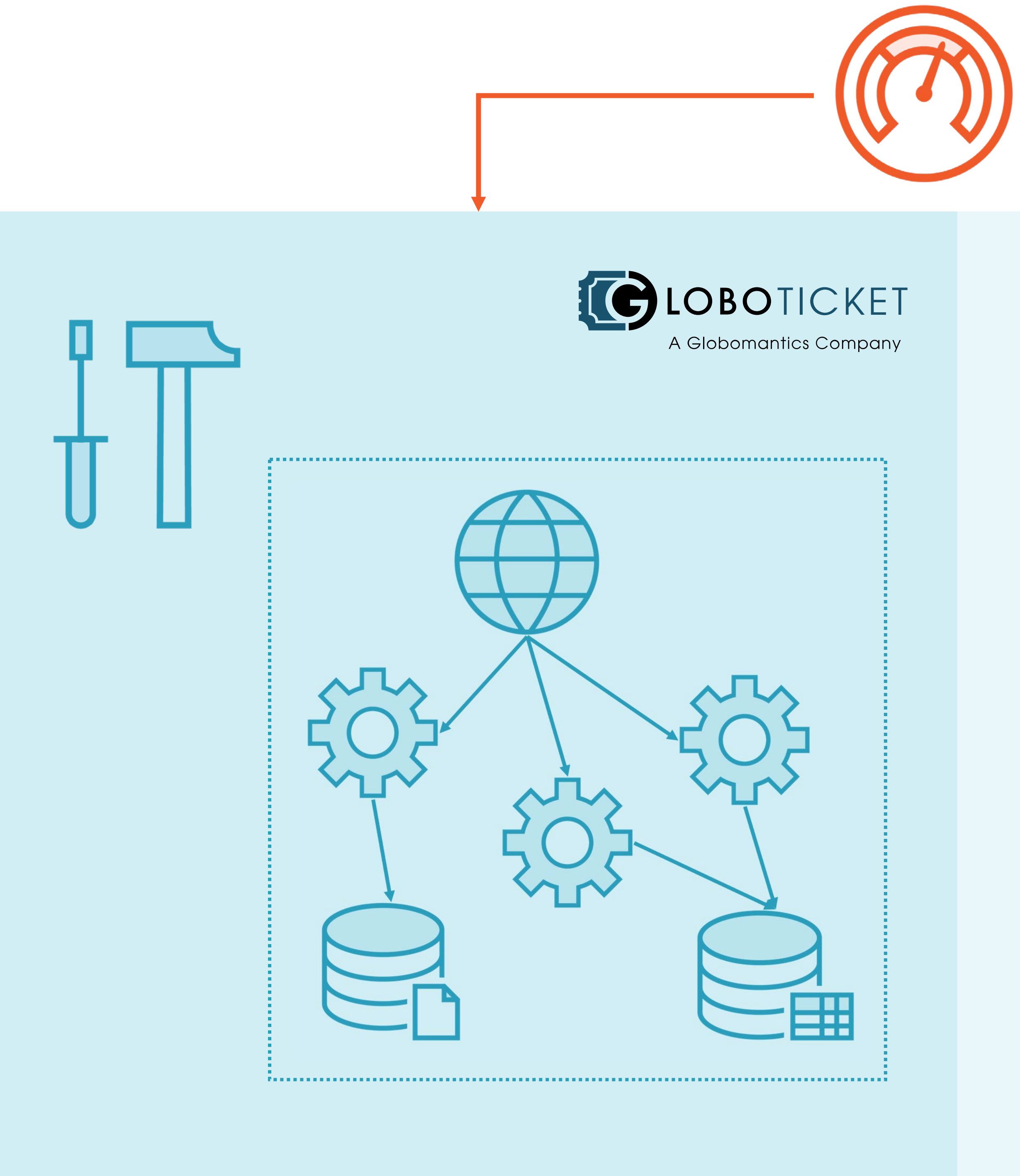
Proactive SLOs

Ticket purchase **99.9% successful within 550ms**

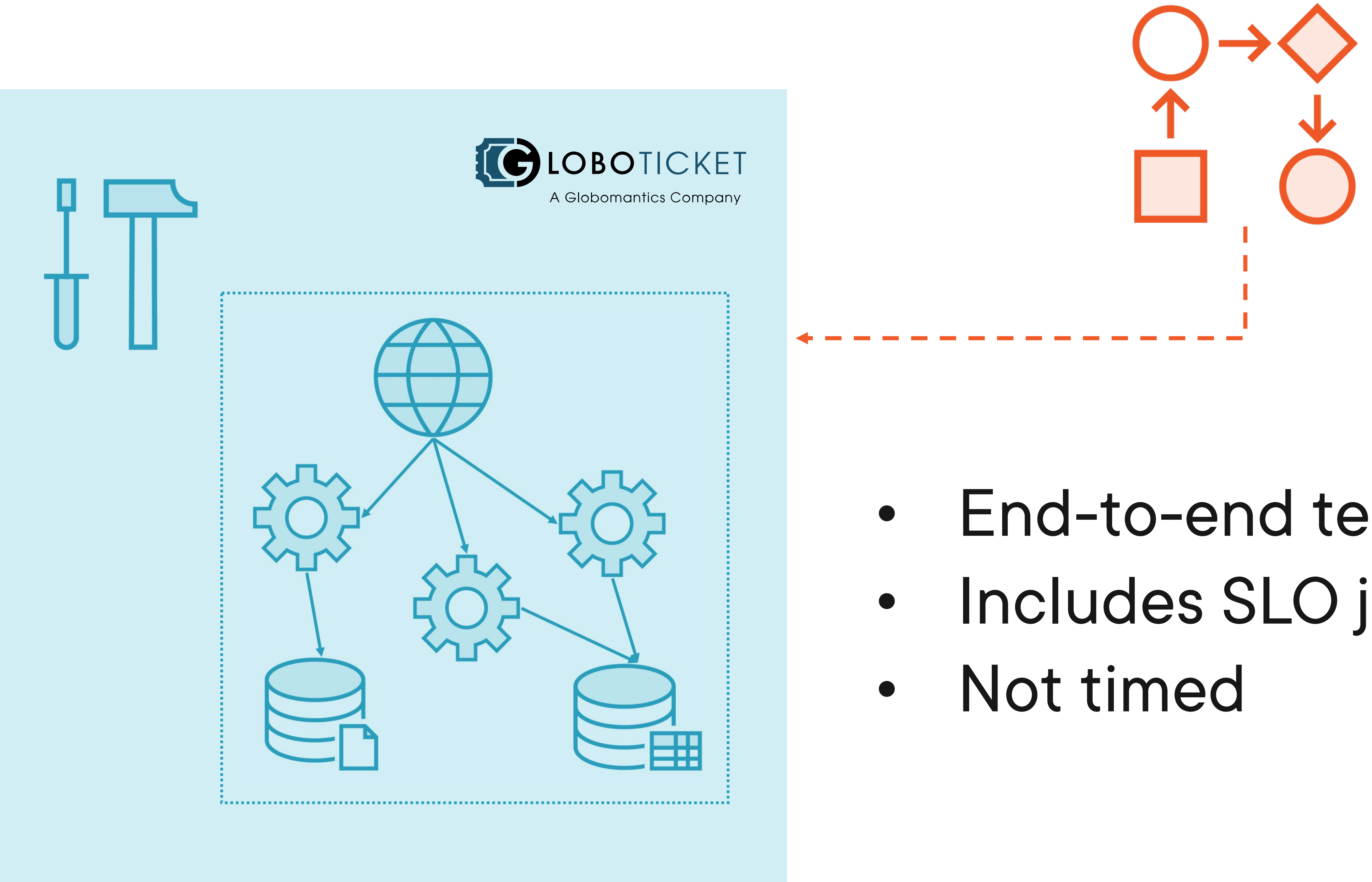
1 month ≈ 43,200 minutes | Error Budget ≈ 43 minutes



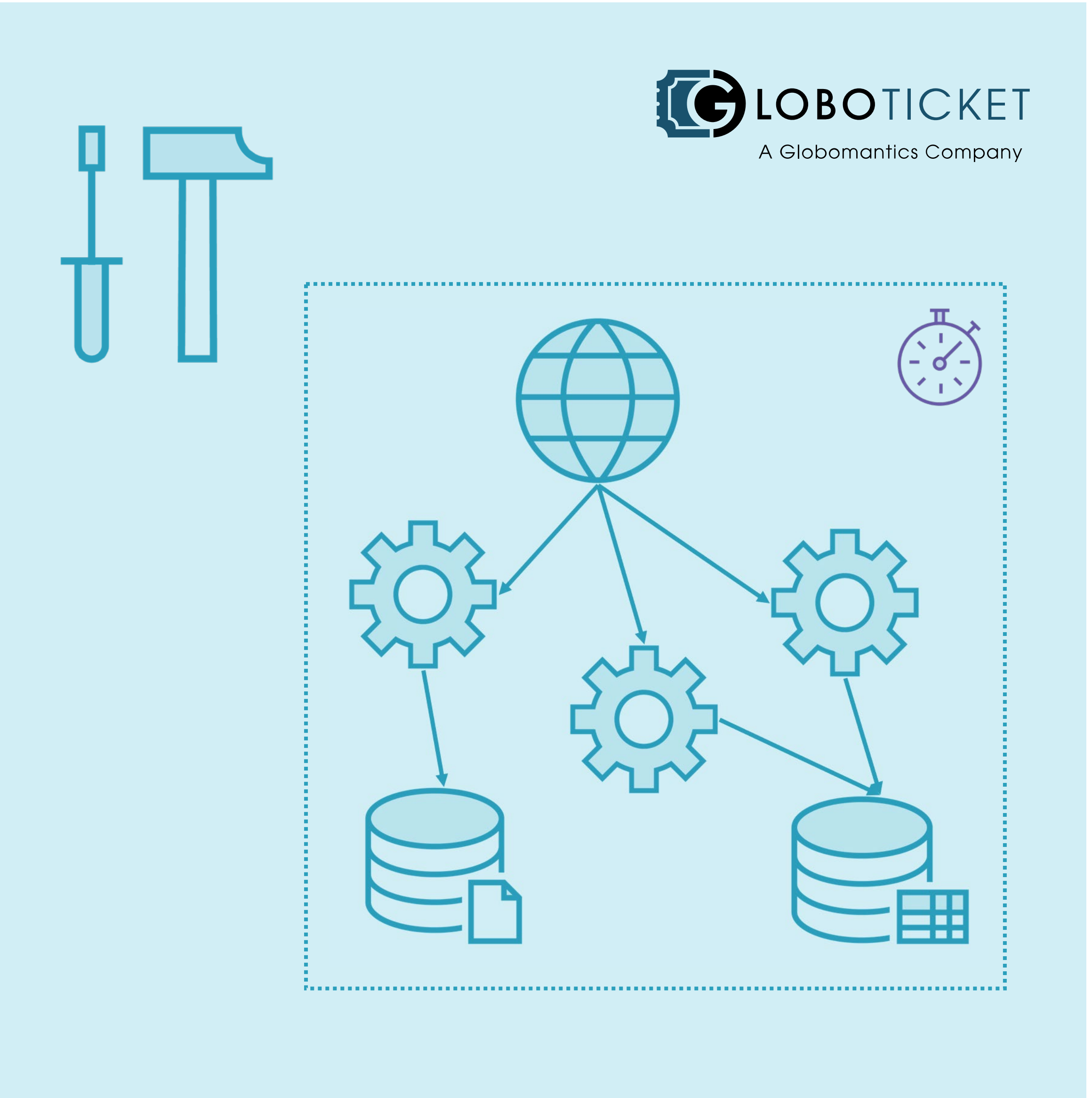
Scenario: Evolving SLOs Over Time

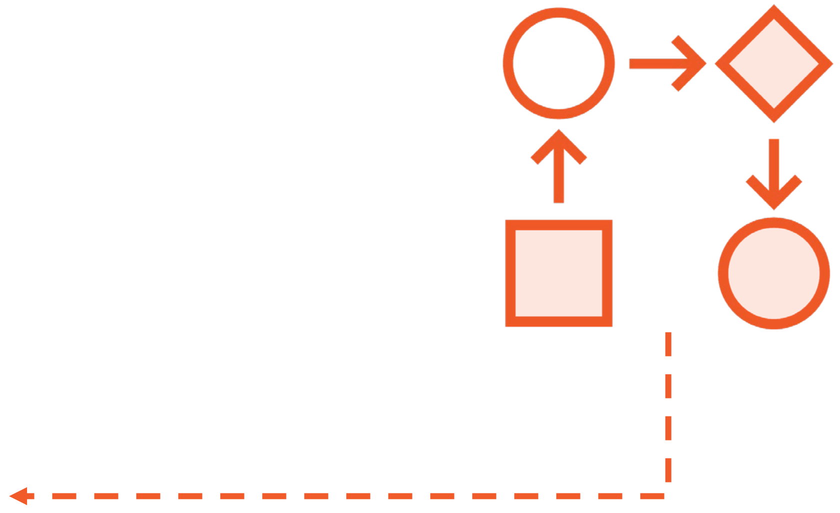


Can the product devs measure the SLO?

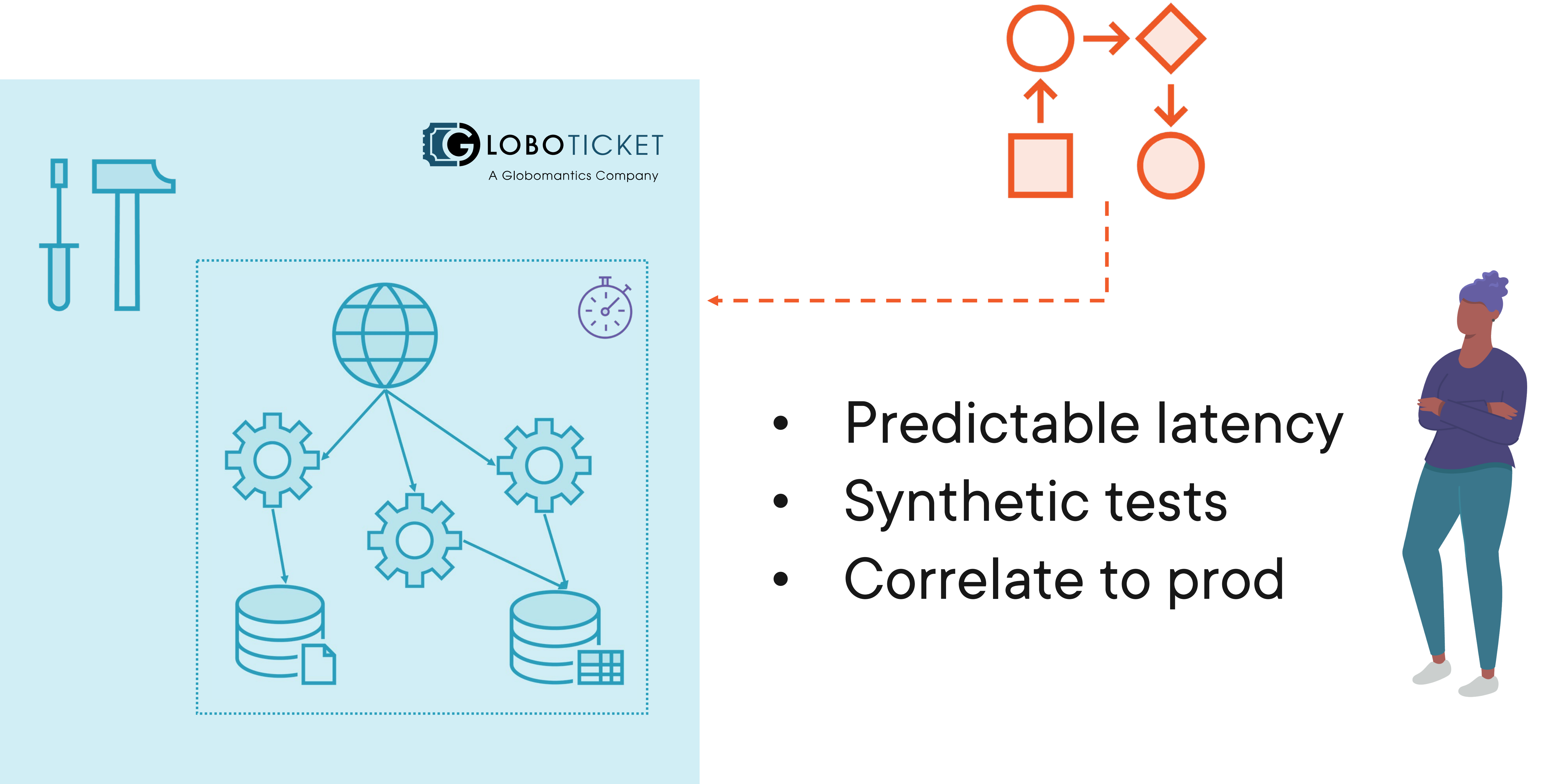


- End-to-end testing
- Includes SLO journey
- Not timed

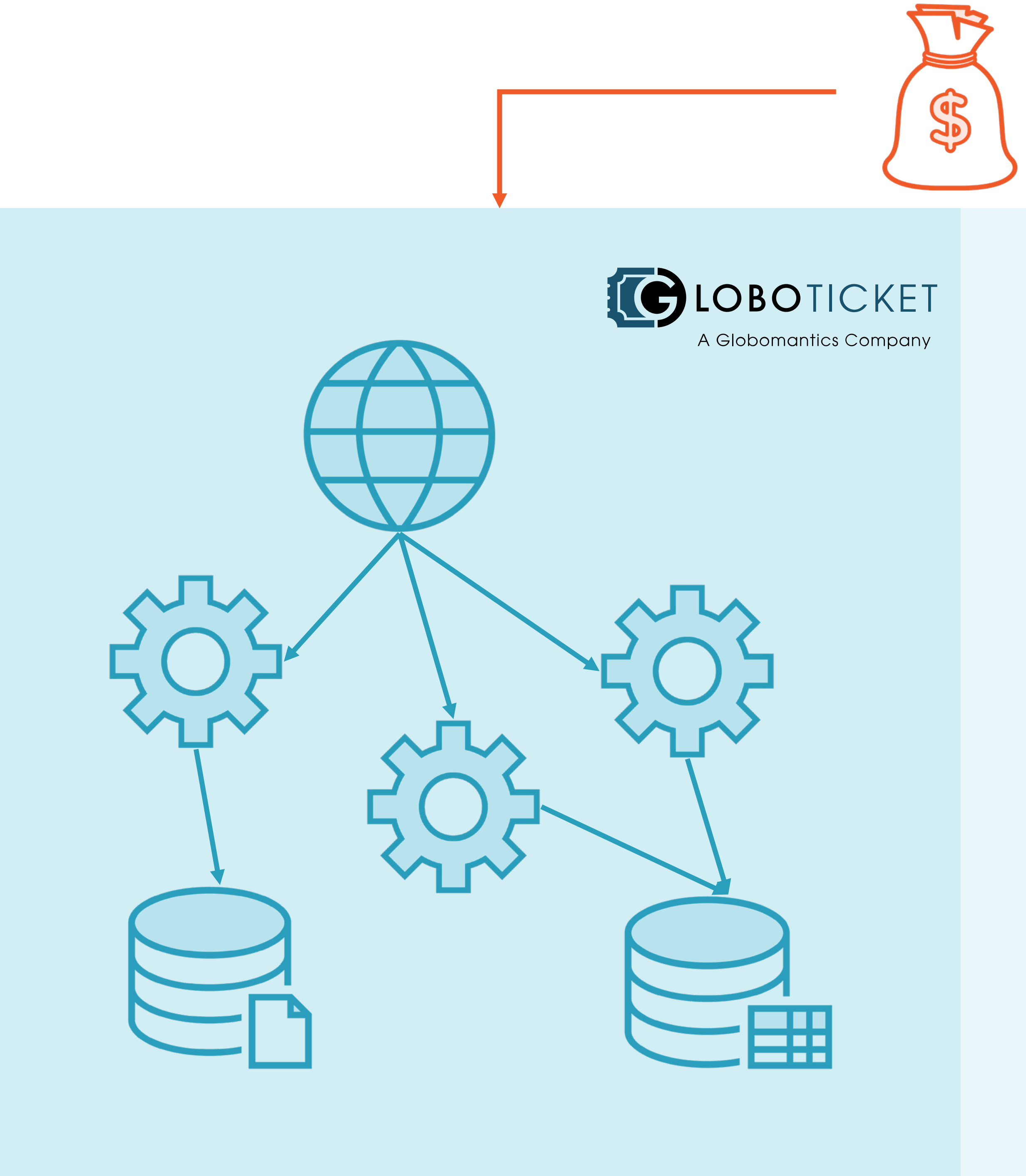




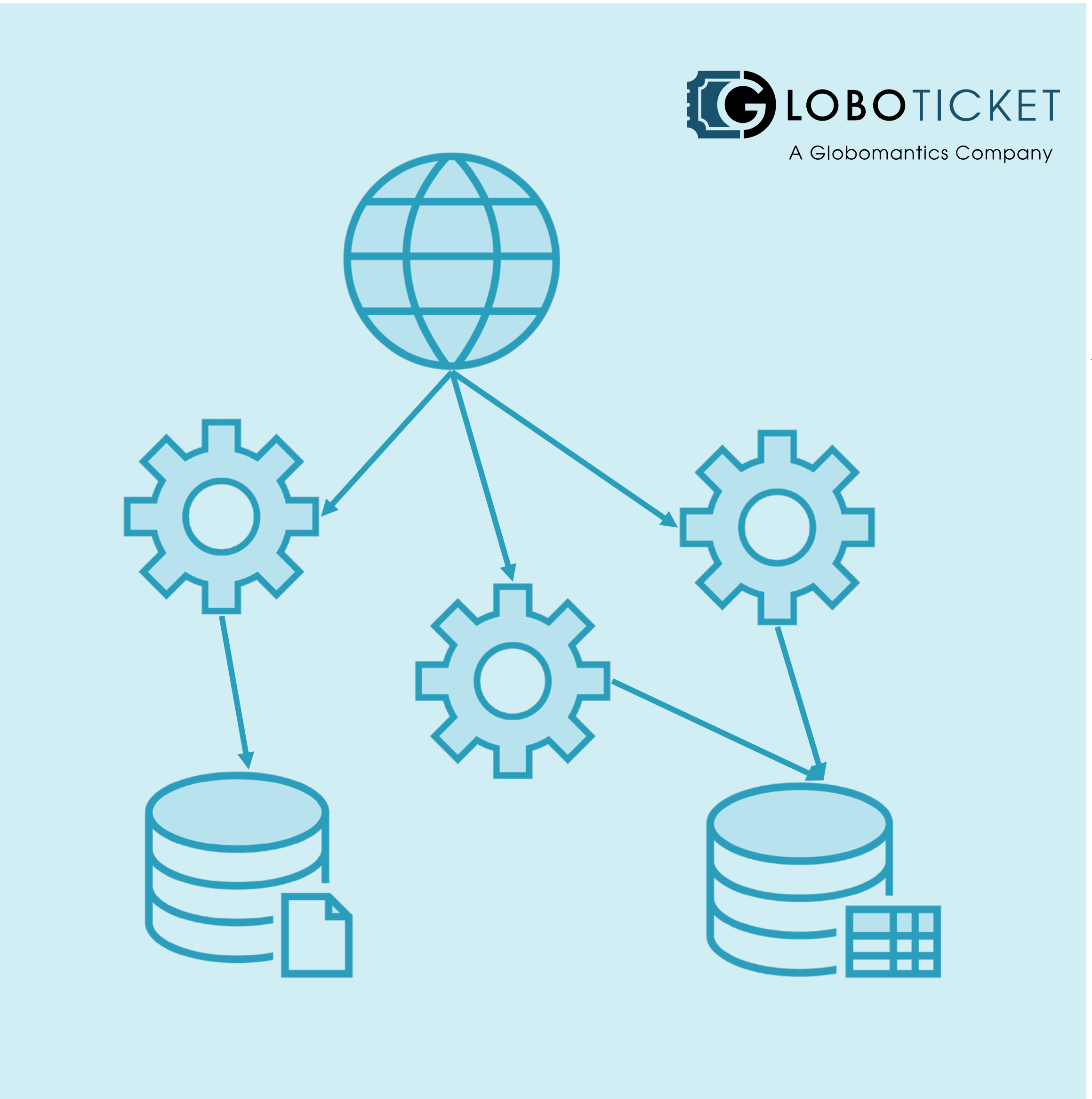
- Include timings
- Collect baseline
- Add SLO test

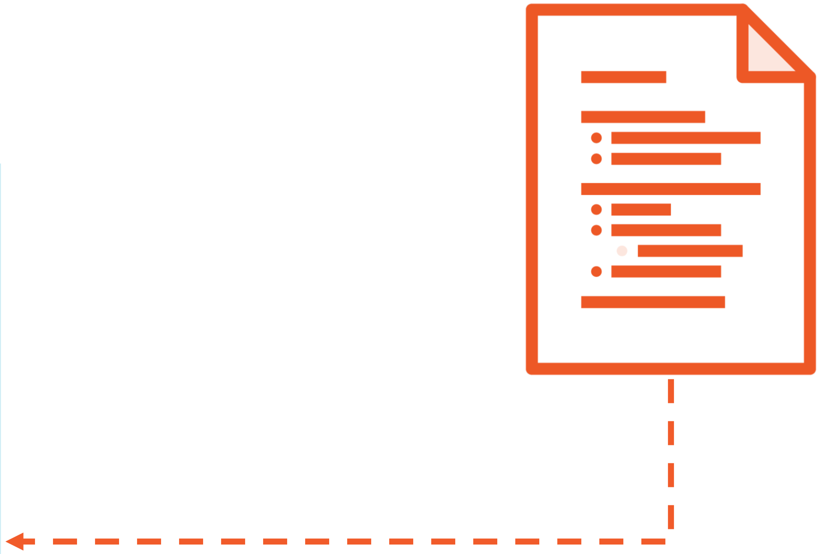


- Predictable latency
- Synthetic tests
- Correlate to prod

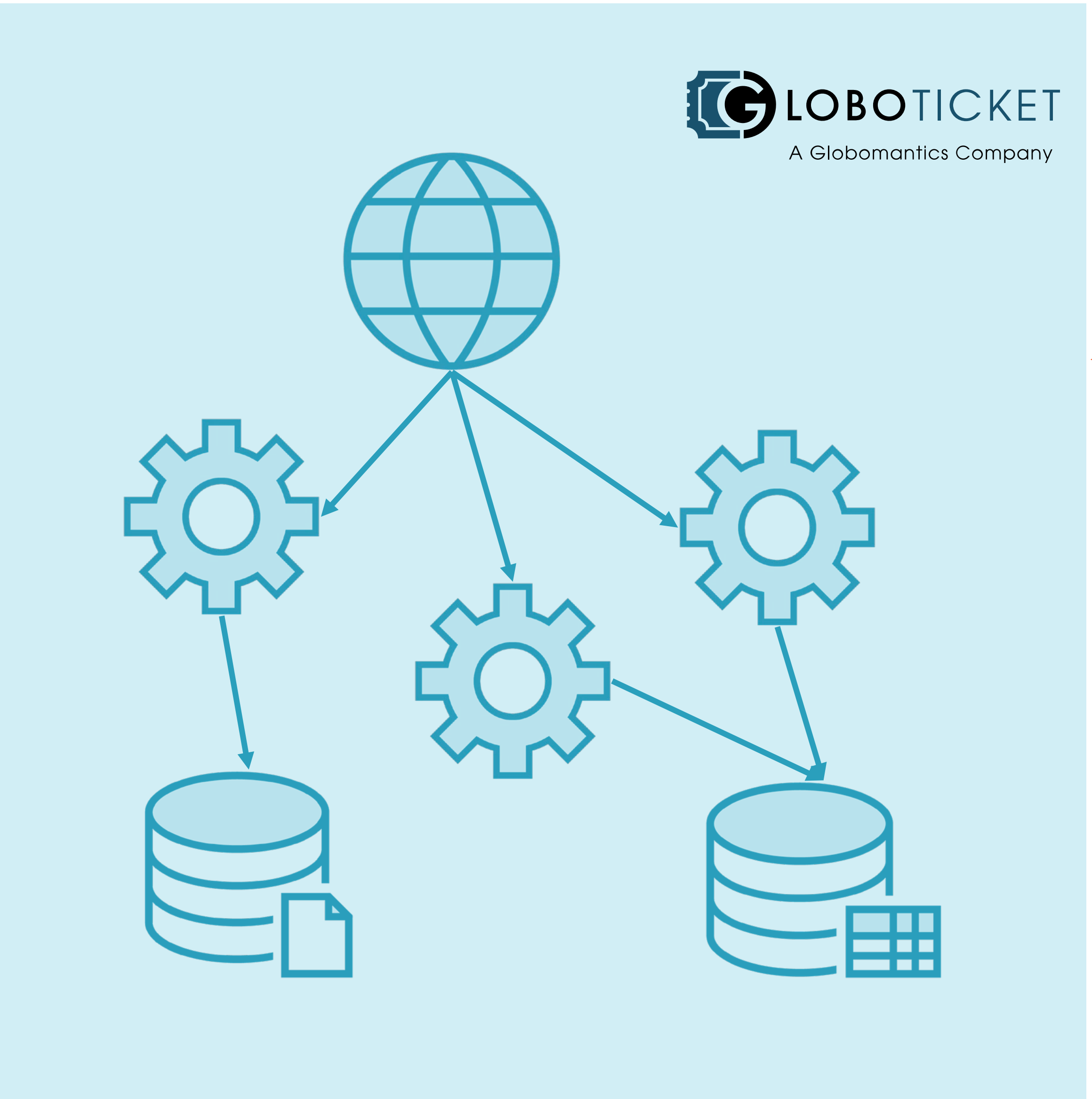


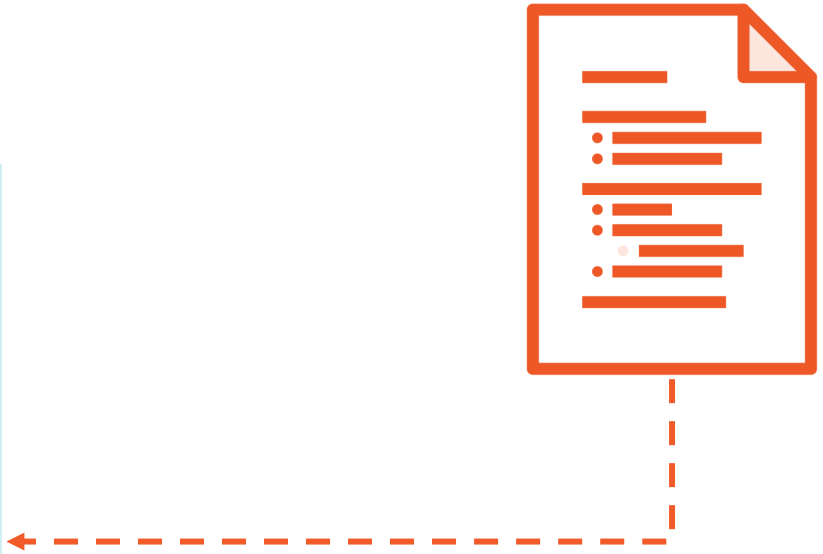
What type of action items from an SLO breach?





- Big issue -> postmortem
- Code or rollout issue
- Product or process actions

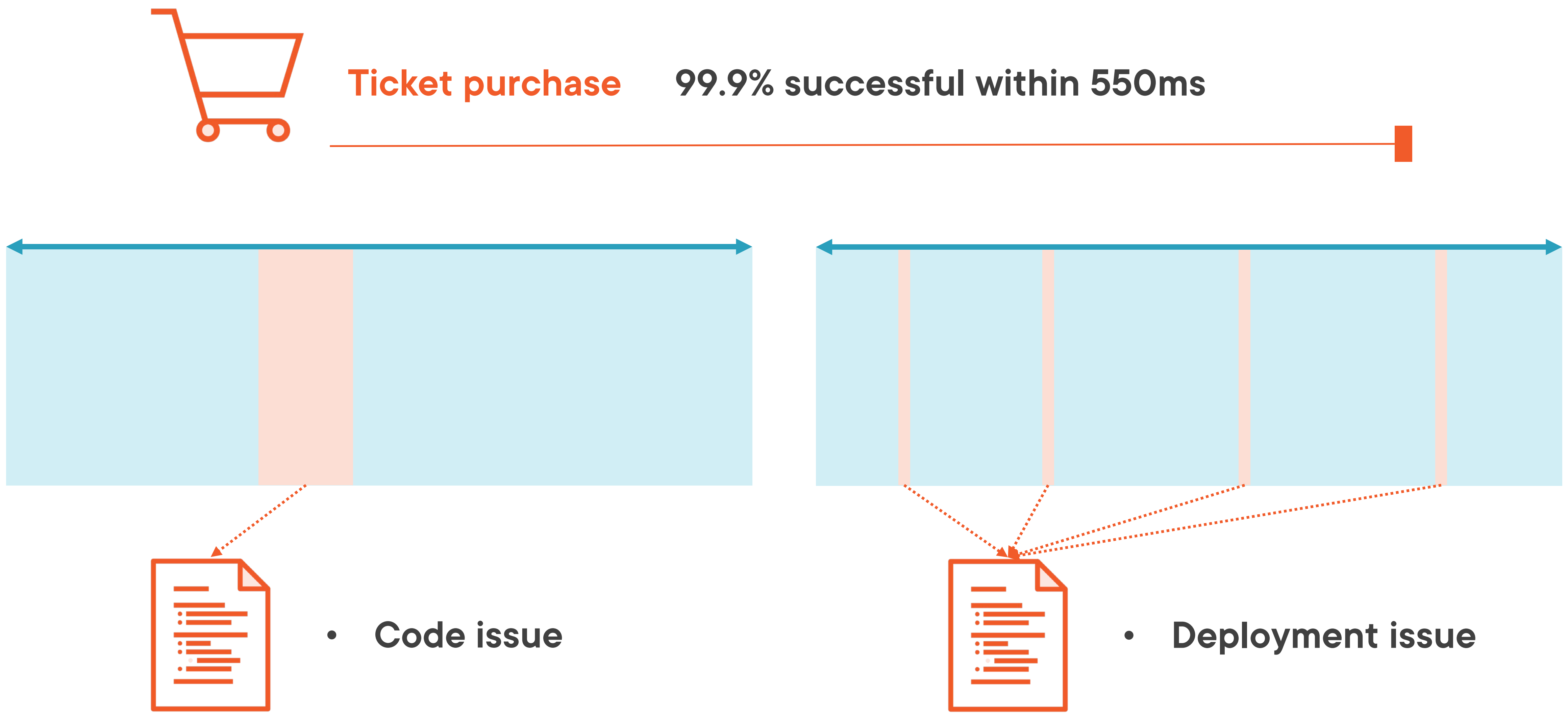




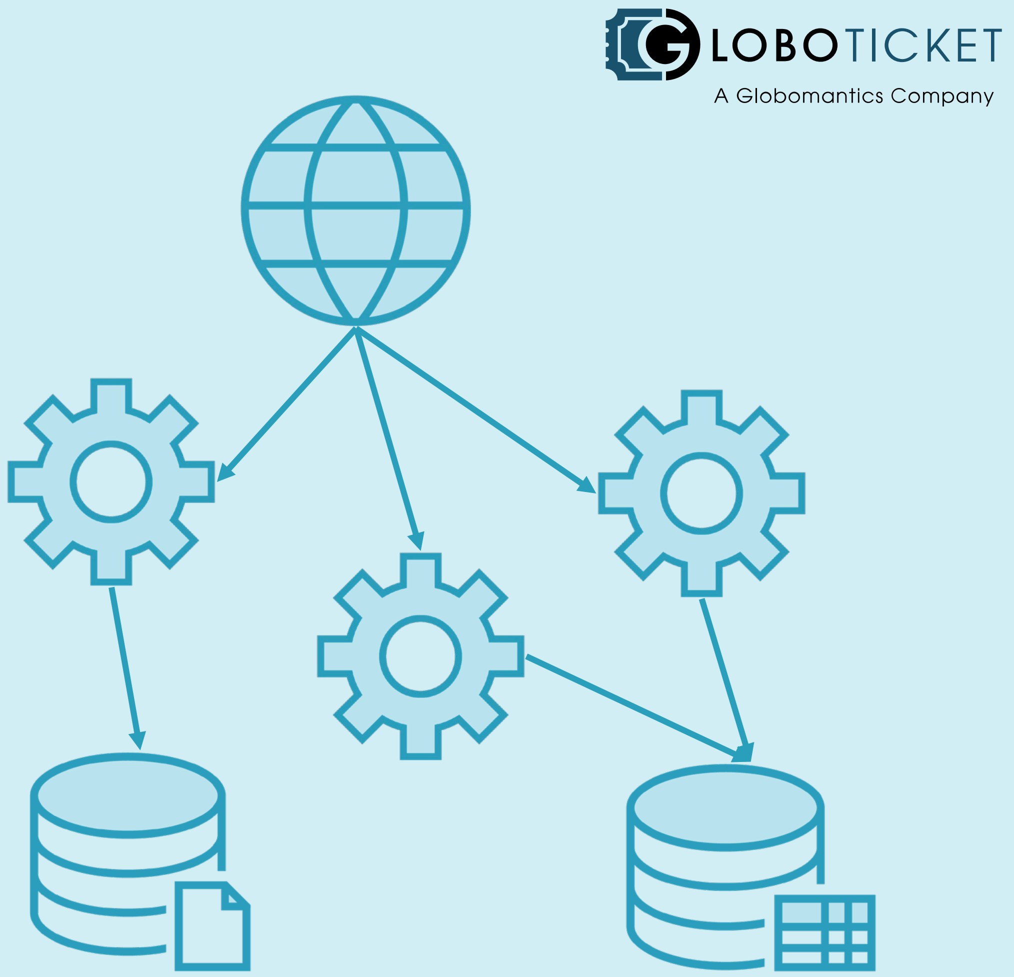
- Error budget policy
- Postmortems:
  - 25% in one issue
  - Budget exhausted

Error Budget Policy

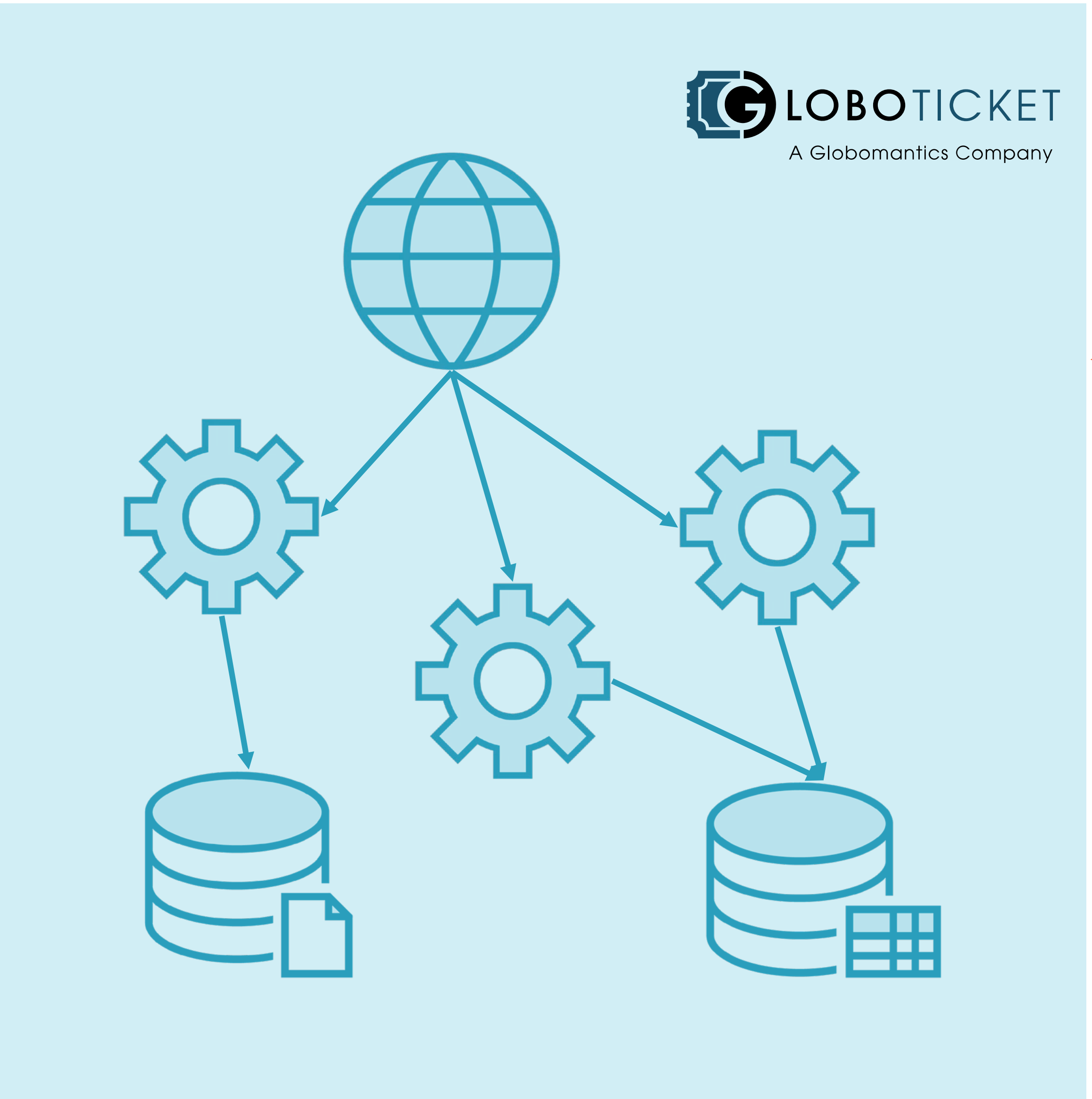
Ticket purchase **99.9% successful within 550ms**

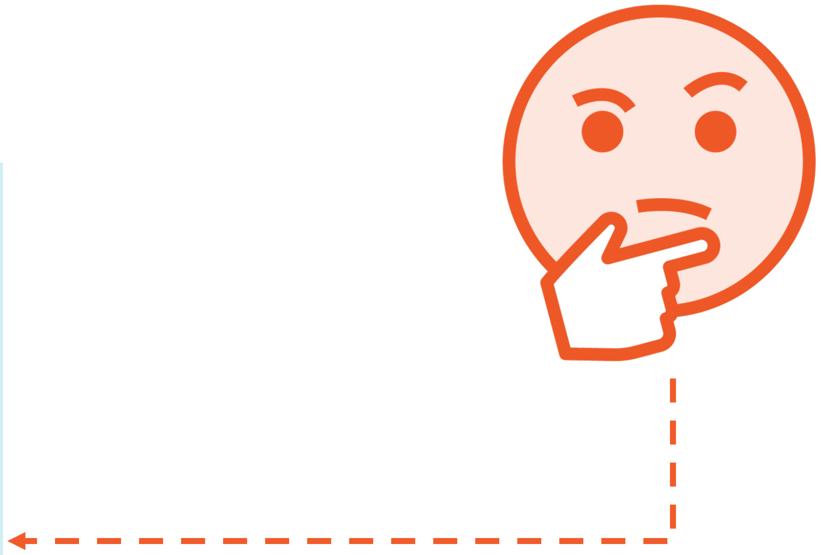


- **Code issue**
- **Deployment issue**



What if the SLO is never breached?

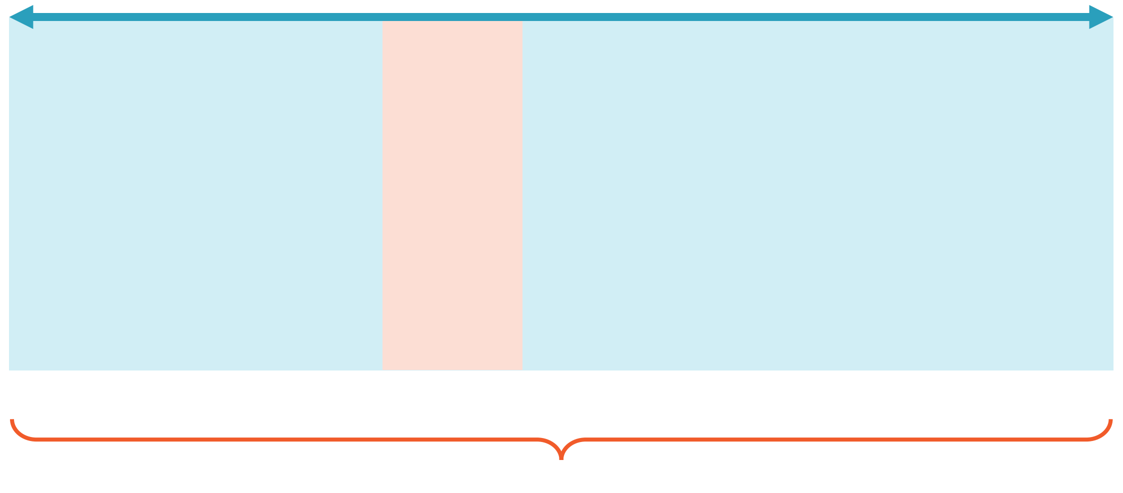




- Not ambitious enough
- No failures
- No improvement

SLO Reviews

Ticket purchase **99.9% successful within 550ms**



Error budget not consumed

- **Reduce latency target**
- **Improve success target**
- **Measure at 99th percentile**

Using SLOs in Feedback Loops

Attainable and Desirable SLOs

Ticket purchase successful within

- 99.9% @ 500ms
- Good for customers
- Achievable for SRE team

- 99.5% @ 800ms
- OK for customers
- Easy for SRE team

Running at SLO Performance

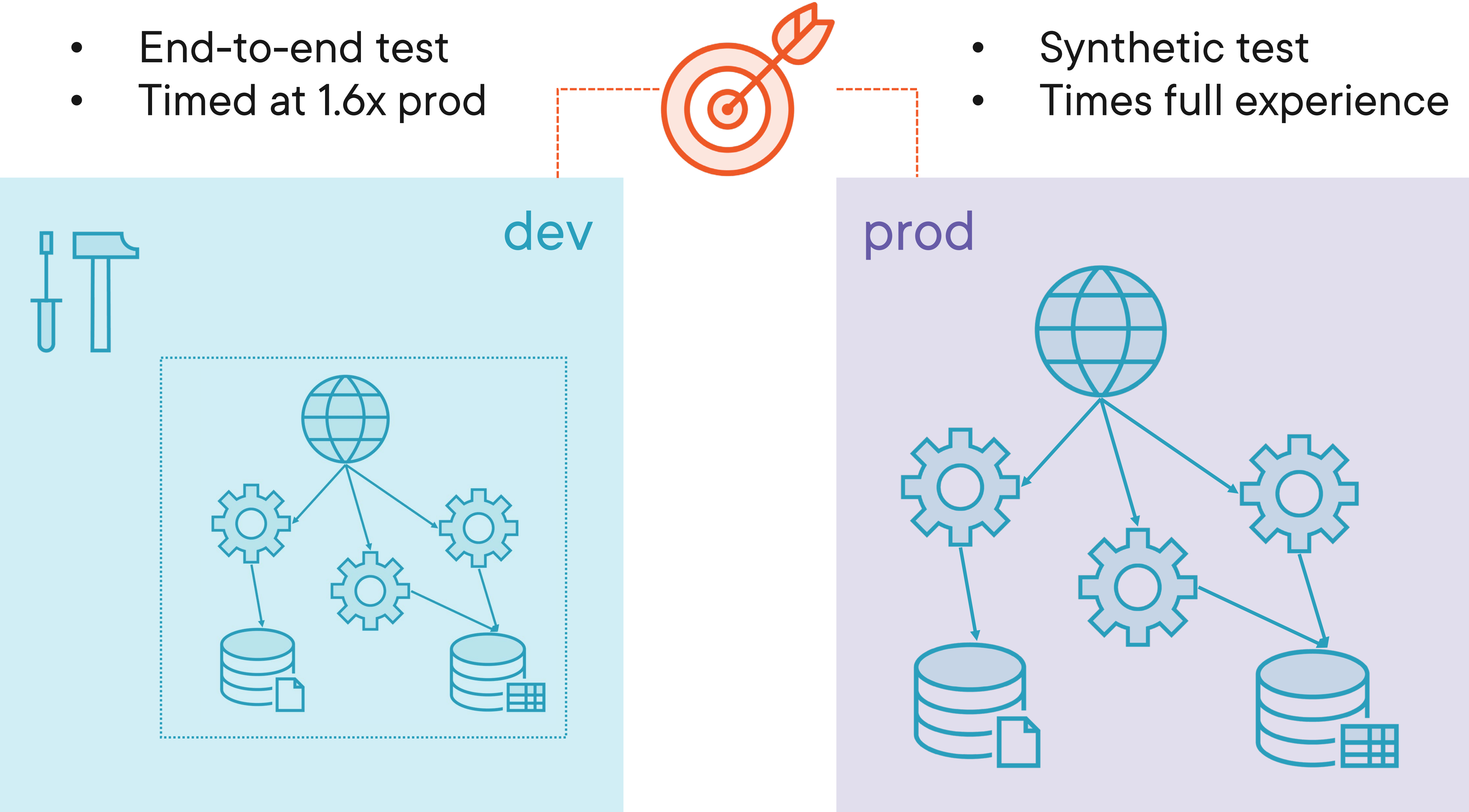
Reduce scale

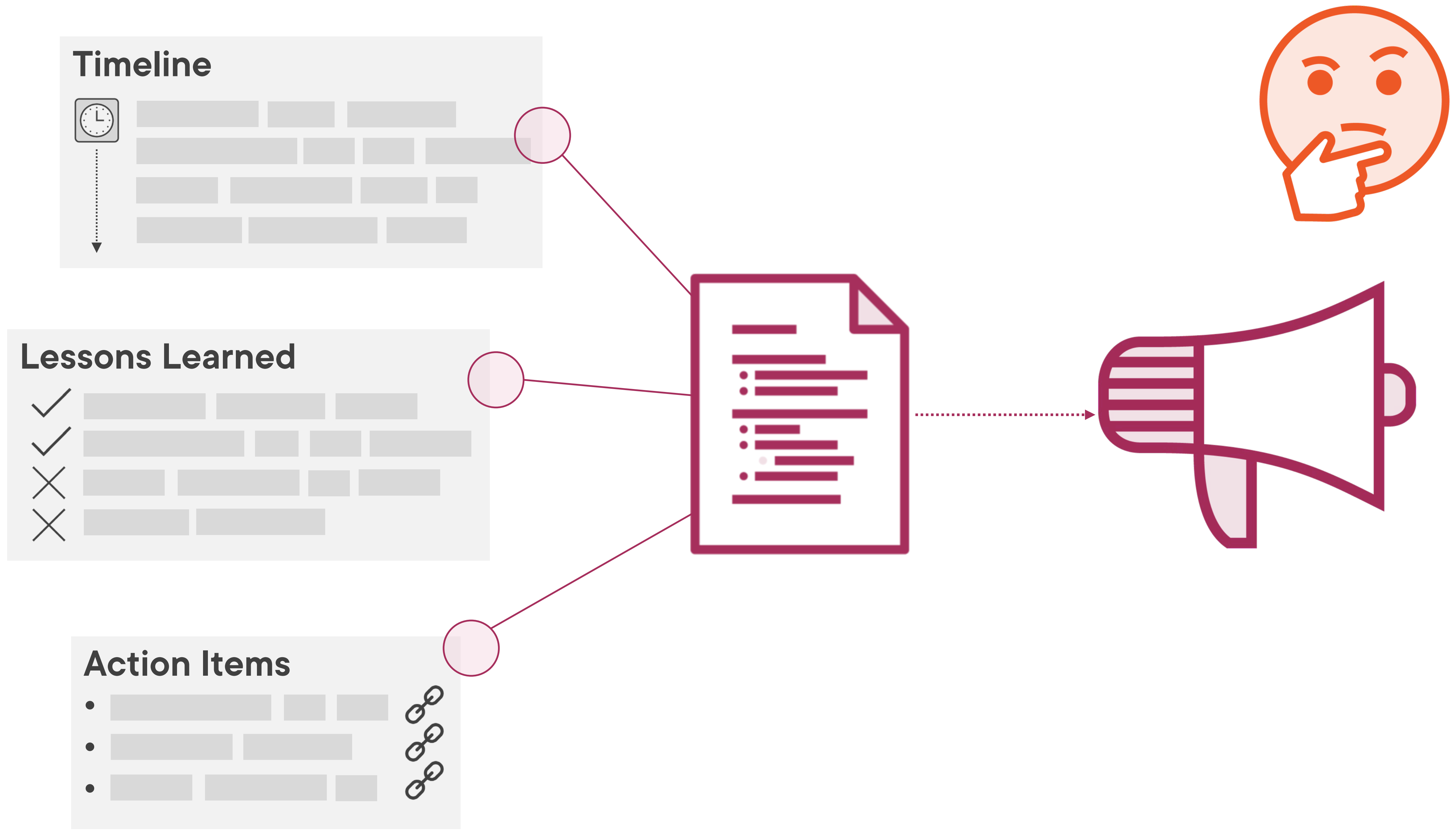
Synthetic outage

Monitor reaction

- End-to-end test
- Timed at 1.6x prod

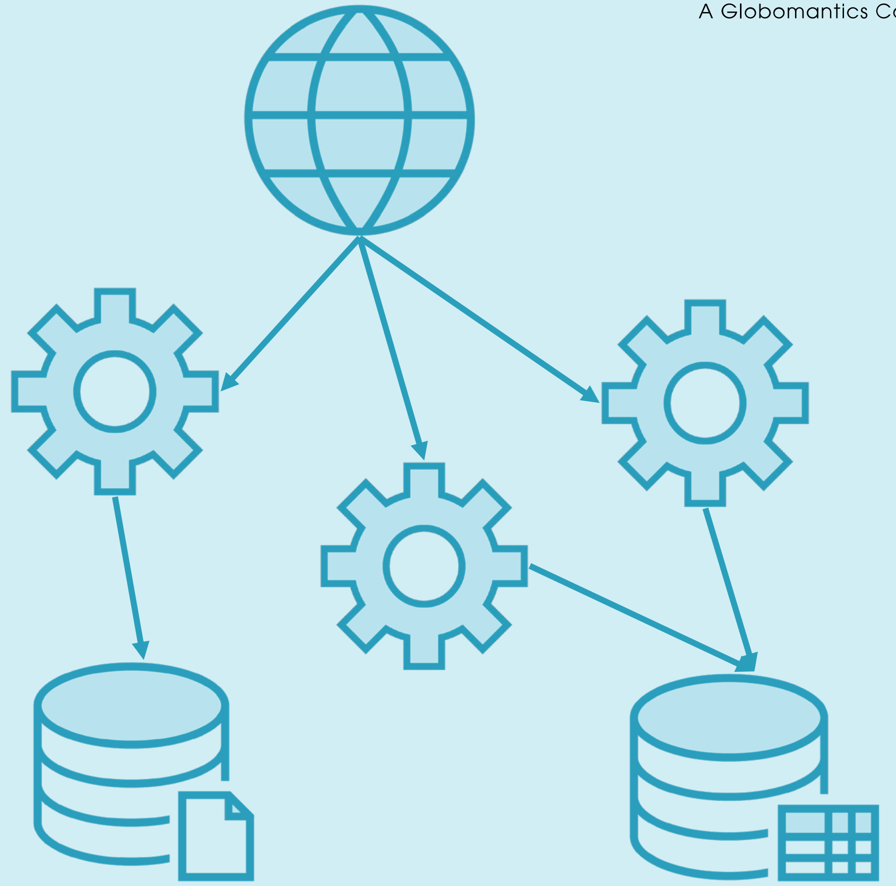
- Synthetic test
- Times full experience



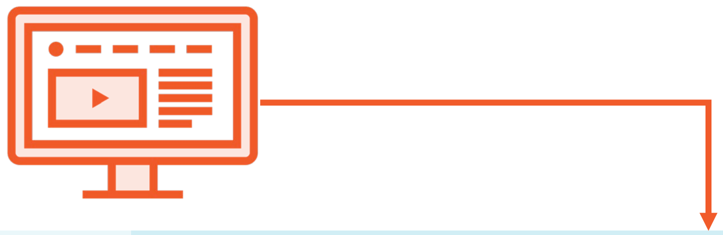

Timeline
Lessons Learned
Action Items

Scenario: Monitoring SRE Performance

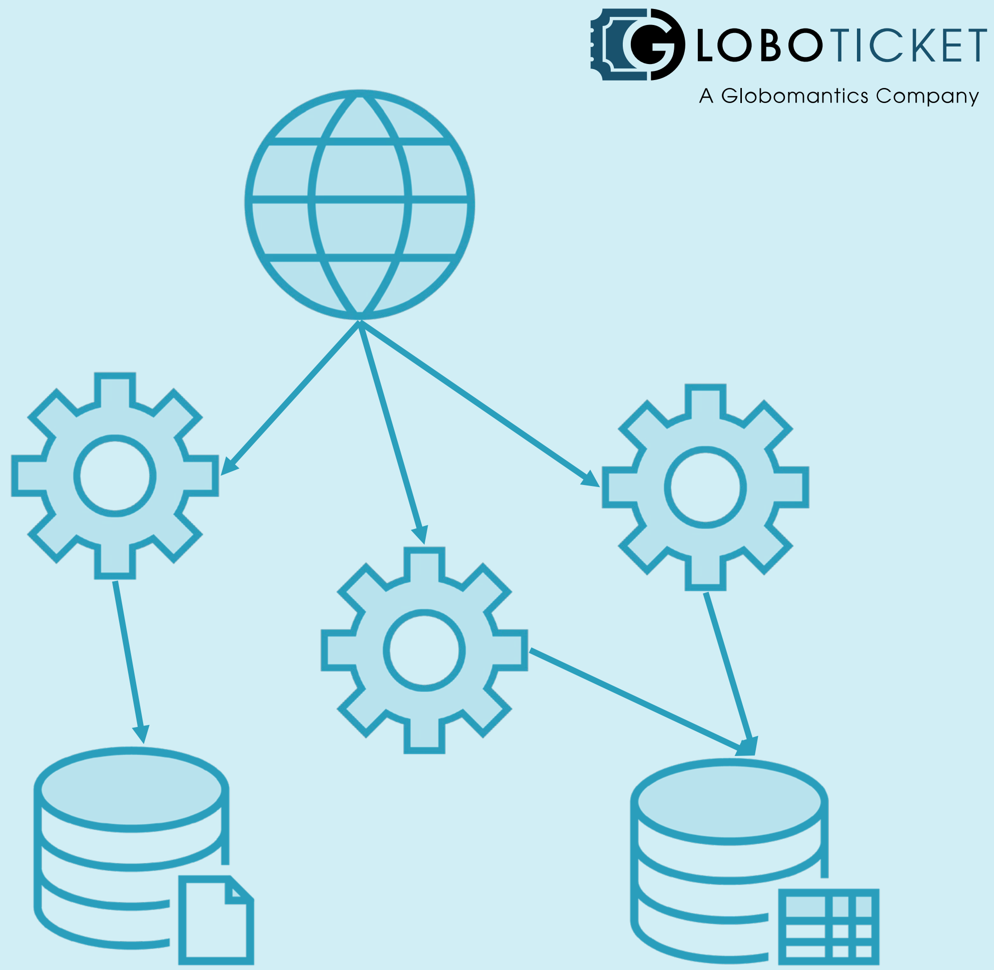
How do you monitor the performance of the SRE team?



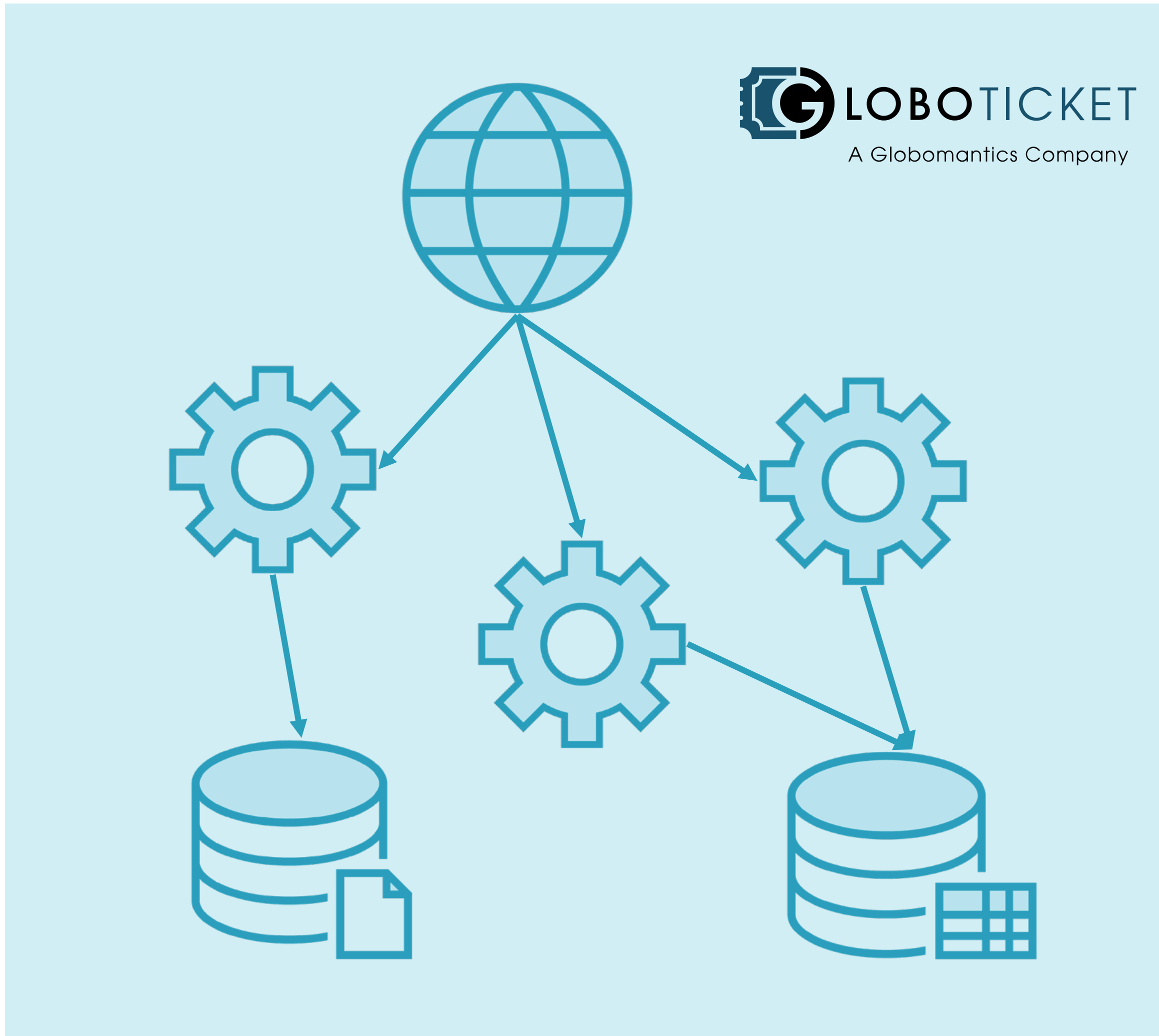
GLOBOTICKET
A Globomantics Company



How can you drive improvement from this data?

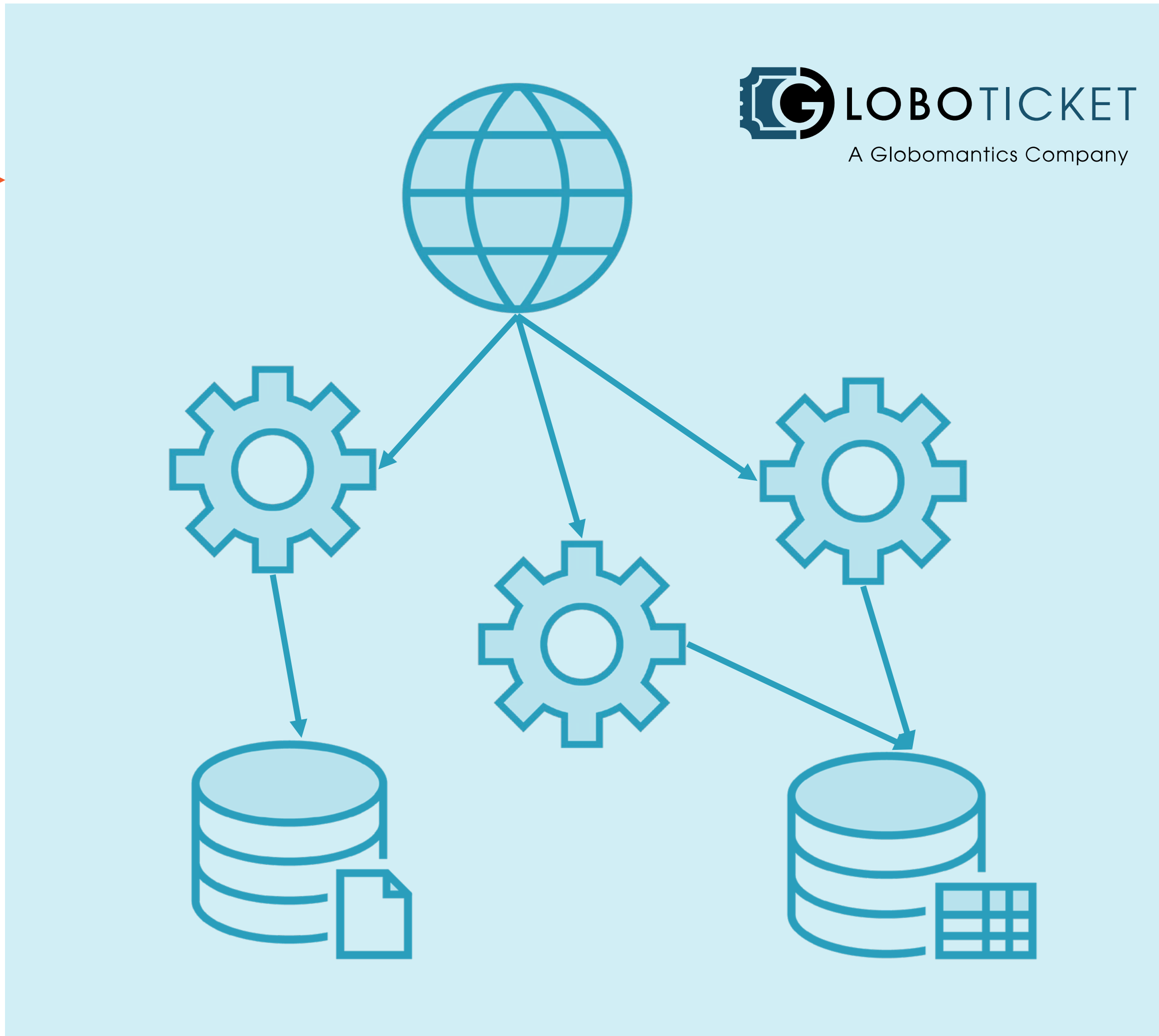


- Monitor incidents
- Record MTBF and MTTR
- Trend product & process

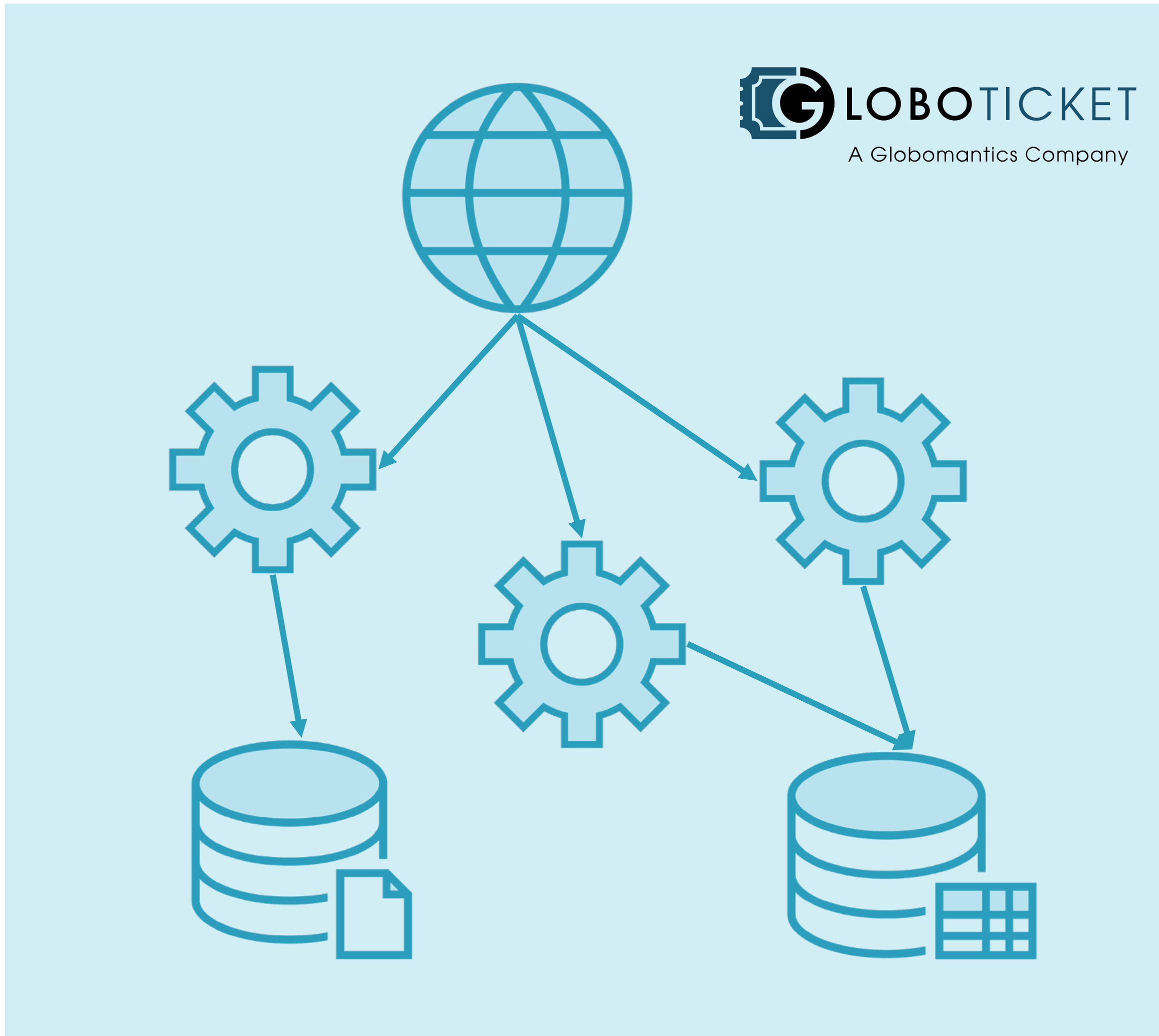


MTTR

- Repair
- Recovery
- Respond
- Resolve



- SRE backlog
- Tied to SLOs
- Otherwise?



Module Summary

Measuring the Measurements

Application performance

SLO breaches and recovery

Incident management

Product Backlog

	Effort	*Team*	*Priority*	*SLO*
Safe restarts	**3 days**	**Dev**	**1**	***x***
Health checks	**2 days**	**SRE**	**2**	***y***
Caching	**5 days**	**SRE**	**3**	***z***

Summary

Desiging service levels

- Customer-focused SLOs
- Meaningful SLIs
- Achievable and desirable

Evolving SLOs over time

- End-to-end and synthetic tests
- Review and tighten goals

Driving improvement with SLOs

- Monitoring MTBF and MTTR
- Product feedback loops

Course Layout

Architecture

Observability

Improvement

Technical Deep Dives

Prometheus, Jenkins, Kubernetes...

Elton Stoneman

We're Done!

So…

- **Please leave a rating**
- **Follow** @EltonStoneman **on Twitter**
- **And watch my other courses ☺**