

**BEYOND BASICS:  
MASTERING COMPOSITE SLO**

**WHO AM I**

## WHO AM I

- Aleksey Shirokikh

## WHO AM I

- Aleksey Shirokikh
- sre-team@semrush

## WHO AM I

- Aleksey Shirokikh
- sre-team@semrush
- mostly interested in monitoring and measurements

# COMPANY

As of Fall 2023, the Semrush database contained over:

# COMPANY

As of Fall 2023, the Semrush database contained over:

- 142 geographic databases

# COMPANY

As of Fall 2023, the Semrush database contained over:

- 142 geographic databases
- 25,3 billion keywords



# COMPANY

As of Fall 2023, the Semrush database contained over:

- 142 geographic databases
- 25,3 billion keywords
- 808 million domains

# COMPANY

As of Fall 2023, the Semrush database contained over:

- 142 geographic databases
- 25,3 billion keywords
- 808 million domains
- 500TB of raw website traffic data for 190 countries and regions

# COMPANY

As of Fall 2023, the Semrush database contained over:

- 142 geographic databases
- 25,3 billion keywords
- 808 million domains
- 500TB of raw website traffic data for 190 countries and regions
- 43 trillion backlinks

# COMPANY

As of Fall 2023, the Semrush database contained over:

- 142 geographic databases
- 25,3 billion keywords
- 808 million domains
- 500TB of raw website traffic data for 190 countries and regions
- 43 trillion backlinks
- 1 billion Google Ads

**PLAN**

# PLAN

- Sli/slo intro

# PLAN

- Sli/slo intro
- sli/slo and a user

# PLAN

- Sli/slo intro
- sli/slo and a user
- Moving to composite slo



# PLAN

- Sli/slo intro
- sli/slo and a user
- Moving to composite slo
- Tool readiness

# PLAN

- Sli/slo intro
- sli/slo and a user
- Moving to composite slo
- Tool readiness
- Final thoughts

# PLAN

- Sli/slo intro
- sli/slo and a user
- Moving to composite slo
- Tool readiness
- Final thoughts
- QA

# BRIEF INTRODUCTION TO SLO

- Gives a nice overview on performance of a product from a dev-team perspective
- Alerts, in case product promises are about to be broken.
- Approach to communicate reliability stuff across the company

# DEFINITION OF BASIC SLOS

```
sum(  
  increase(  
    http_requests_count{  
      host="billing.semrush.com", status=~"5..|499"} [28d]  
    )  
  )  
/  
sum(  
  increase(  
    http_requests_count{  
      host="billing.semrush.com"} [28d]  
    )  
)
```

# USEFUL TYPES

- Availability

# USEFUL TYPES

- Availability
- Latency

# USEFUL TYPES

- Availability
- Latency
- Quality



# USEFUL TYPES

- Availability
- Latency
- Quality
- other



**BACKEND**



**FRONTEND**



**API's**



**KITCHEN**

# KITCHEN

- measure from kitchen

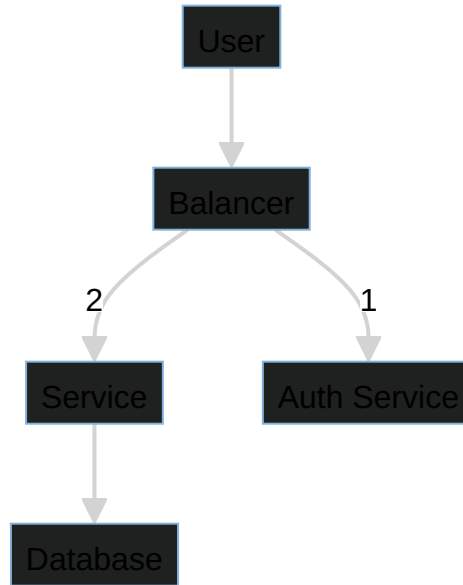
## KITCHEN

- measure from kitchen
- measure from customer in a hall

## KITCHEN

- measure from kitchen
- measure from customer in a hall
- measure for home delivery

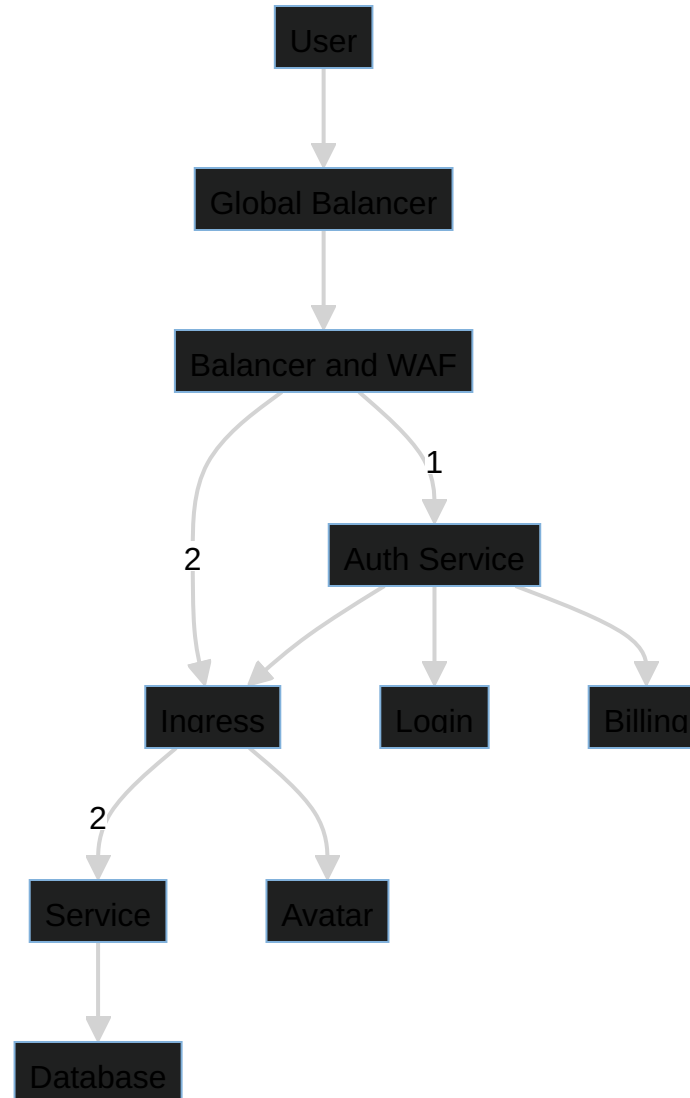
# EXAMPLE SERVICE DIAGRAM





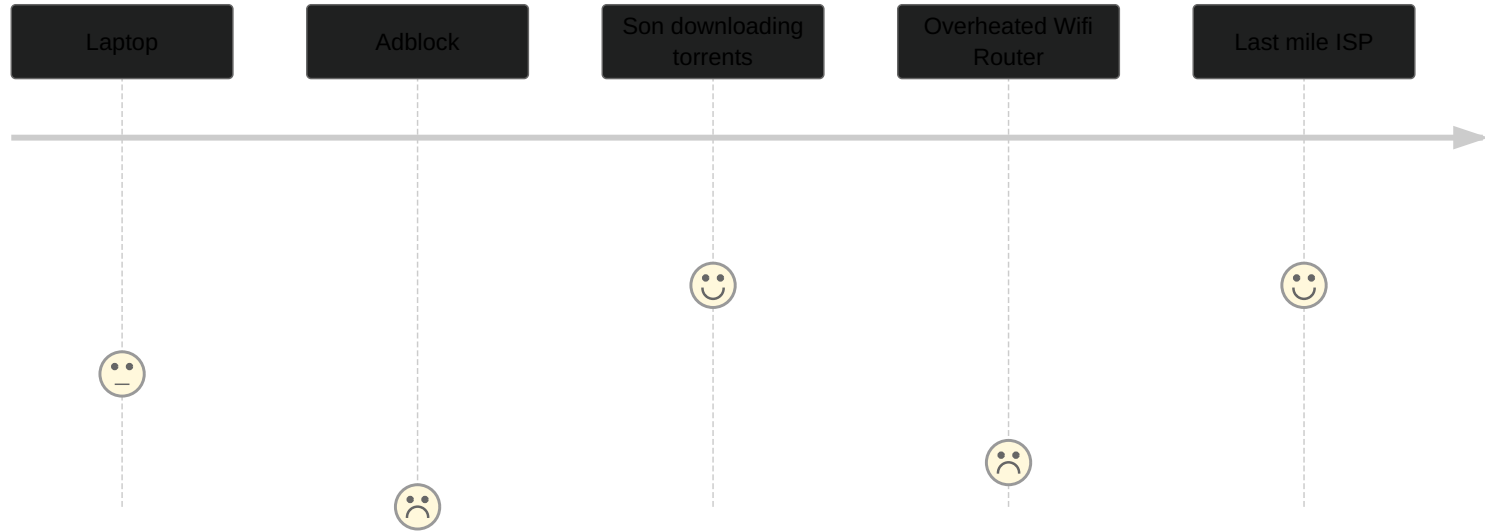
# WHY IT IS NOT A WHOLE STORY

User perspective differs from service



# USER ITSELF

User's env



# MEASURE FROM FRONTEND

Possible to measure from

- Google Analytics
- Newrelic
- SpeedCurve
- ....

## **BLACKBOX CHECKS**

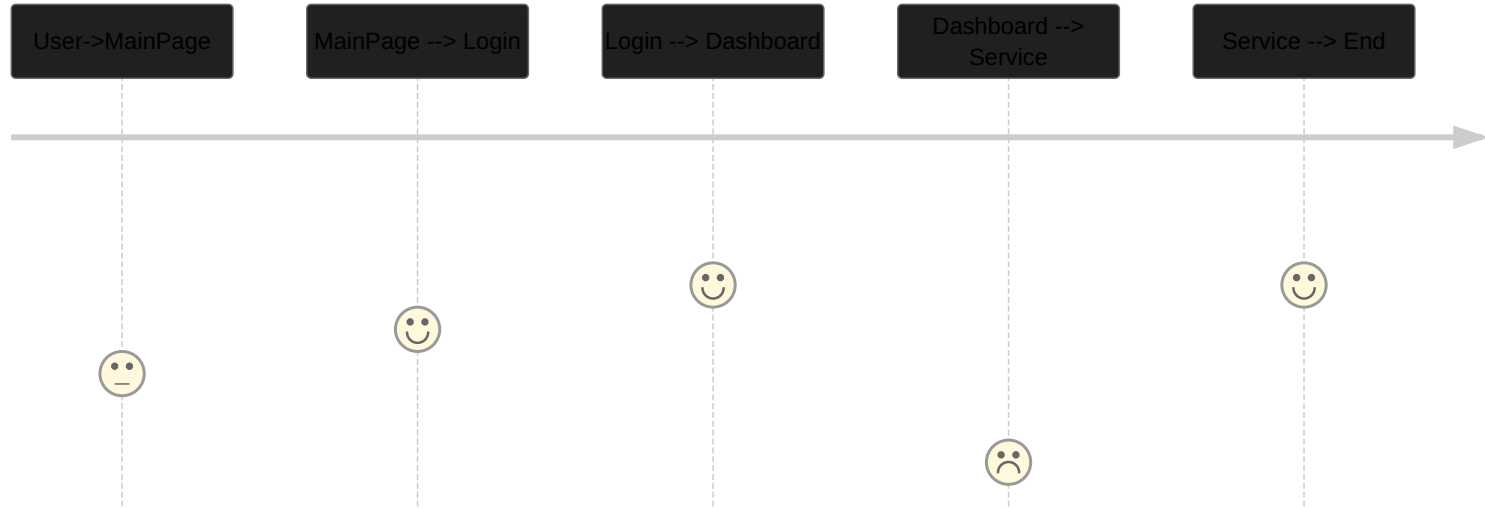
once a minute check, you know...

**SO WE NEED A BEST OF BOTH WORLDS**

**COMPOSITE SLO**



# User scenario





- Scenario -- chain of pages that user must follow to achieve it's goal

# User scenario



- Scenario -- chain of pages that user must follow to achieve it's goal
- Critical user journey -- parts of the system needed to provide service

# User scenario



# Critical User Journey



# **HOW TO DEFINE YOUR OWN COMPOSITE SLO**

## **HOW TO DEFINE YOUR OWN COMPOSITE SLO**

- start with product owner and ask a questions.

## HOW TO DEFINE YOUR OWN COMPOSITE SLO

- start with product owner and ask a questions.
- what is the most valuable in terms of money page in your product



## HOW TO DEFINE YOUR OWN COMPOSITE SLO

- start with product owner and ask a questions.
- what is the most valuable in terms of money page in your product
- from your side take a completely different approach. Find most popular route

**MAIN QUESTION IS AGGREGATION LOGIC**

Strategy

# MAIN QUESTION IS AGGREGATION LOGIC

Strategy

- $\min(\text{slo})$

# MAIN QUESTION IS AGGREGATION LOGIC

Strategy

- min(slo)
- events

# MAIN QUESTION IS AGGREGATION LOGIC

## Strategy

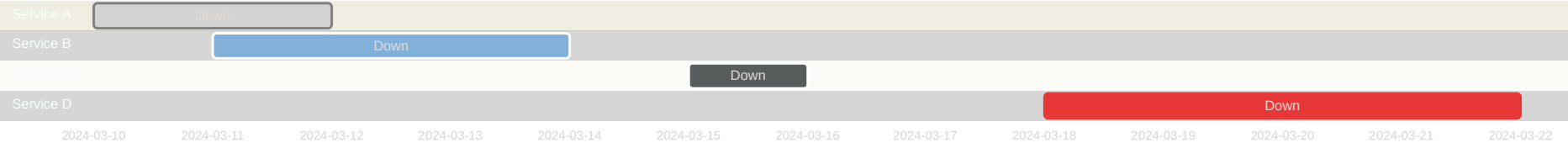
- min(slo)
- events
- time slices

## MINIMAL SLO METHOD

Lets assume we have 4 slo.  
Lets just use min for aggregation.  
Or even avg

# BUT

Services Downtime Timeline



## EVENTS METHOD

```
sum (errors_q for every slo involved)
/  
sum (total_q for every slo involved)
```



**BUT**

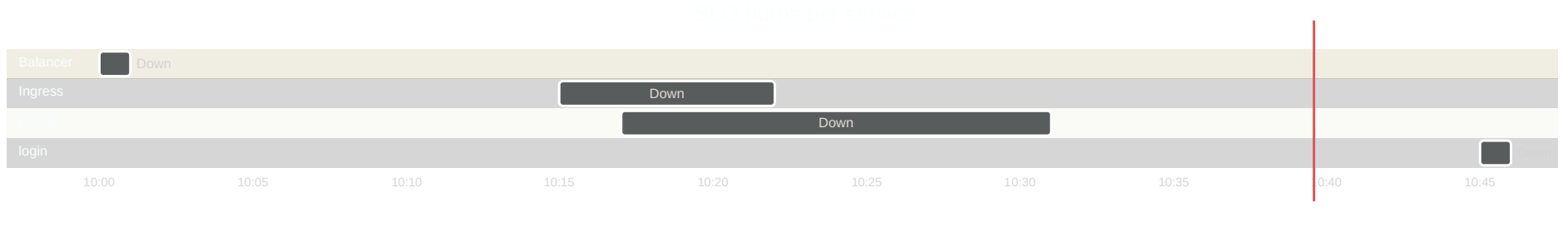
**BUT**

- totals are counted multiple times

## EVENTS METHOD. IMPROVED

```
sum ("errors_q for every slo involved")  
/  
"total_q for one actually done the job"
```

# TIMESLICE METHOD. RAW



System B



System C



System A



## DISADVANTAGES OF TIMESLICE

- more than 99.99 is not measurable

```
28 days * 24 hours * 60 minutes = 40320 minutes  
(1 / 40320) * 100 ~= 99.9975
```

- mouthful to explain

**BUT**

**BUT**

- Serial



**BUT**

- Serial
- Parallel Dependencies

# **SUPERBOSS**

- n out of k
- 3 out of 4

timeslice method. better

components:

- name: top

anyOf:

- service: front-balancer

slo: requests-availability

components:

- name: clusters

anyOf:

- name: cluster 1

allOf:

- service: auth-proxy

slo: requests-availability-cl1

version: 2

- service: dashboard

slo: requests-availability-cl2

SLO: front-balancer

```
and (  
  clusters(  
    cl1(auth-proxy and dashboard)  
    or  
    cl2(auth-proxy and dashboard  
  )  
)  
)
```

```
1 and (clusters(cl1(1 and 0) or cl2(1 and 1)))
```

## TOOLS USED

prometheus, gather data  
Victoria metrics, storage and metricsql

<https://github.com/slok/sloth>

# SLOTH

```
slos:
  - name: "http-availability"
    objective: 99.99
    description: "Common SLO based on availability for HTTP re
sli:
  events:
    error_query: sum(rate(http_request_duration_seconds_co
      {job="myapp", code=~"(5..|429)"}[{{.window}}]
    ))
    total_query: sum(rate(http_request_duration_seconds_co
      {job="myapp"}[{{.window}}]
    ))
  alerting:
    ...
```

## METRICSQL

- share\_gt\_over\_time
- share\_le\_over\_time
- share\_eq\_over\_time



# DEBRIEF

- SLO for the service
- Composite SLO for user
  - min/avg/max (-)
  - events (?)
  - timeslices (!)
- tools
  - prometheus
  - victoria metrics
  - <https://github.com/slok/sloth>

**YOUR THOUGHTS ?**

## Links:

- <https://github.com/slok/sloth>
- <https://sre.google/workbook/alerting-on-slos/#6-multiwindow-multi-burn-rate-alerts>
- <https://blog.alexewerlof.com/p/composite-slo>
- [https://docs.nobl9.com/Guides/SLO\\_Guides/composite-guide/](https://docs.nobl9.com/Guides/SLO_Guides/composite-guide/)