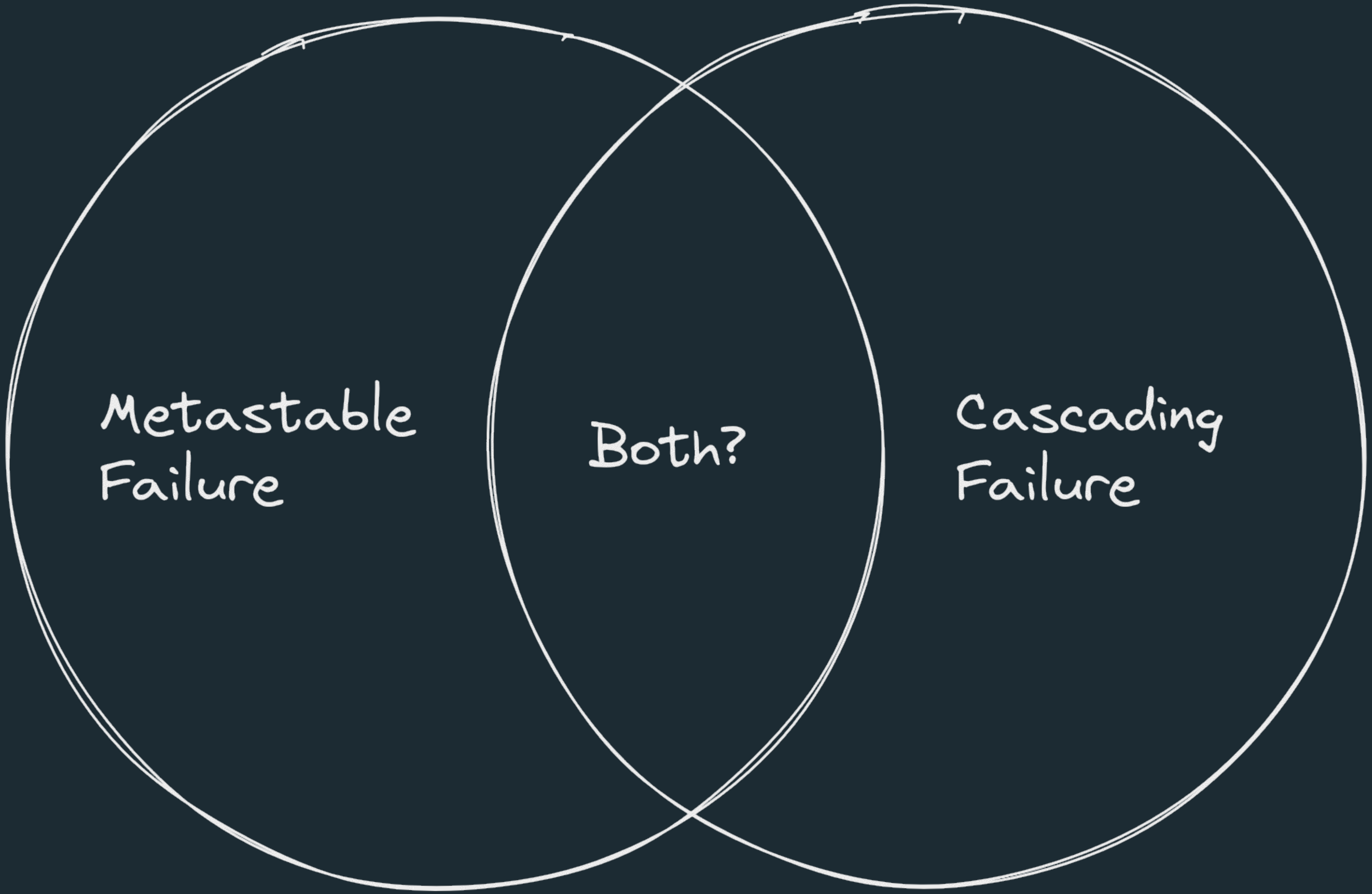
“We’re Still Down”

A metastable failure tale

Kyle Lexmond
Production Engineer
SRECon 2023 Americas,
March 2023

What is a metastable failure?

1. Trigger causes system to enter a bad state, and stay in bad state even after trigger is removed
2. While in the bad state, the system is effectively unusable
3. Sustaining effect that prevents the system from self recovering

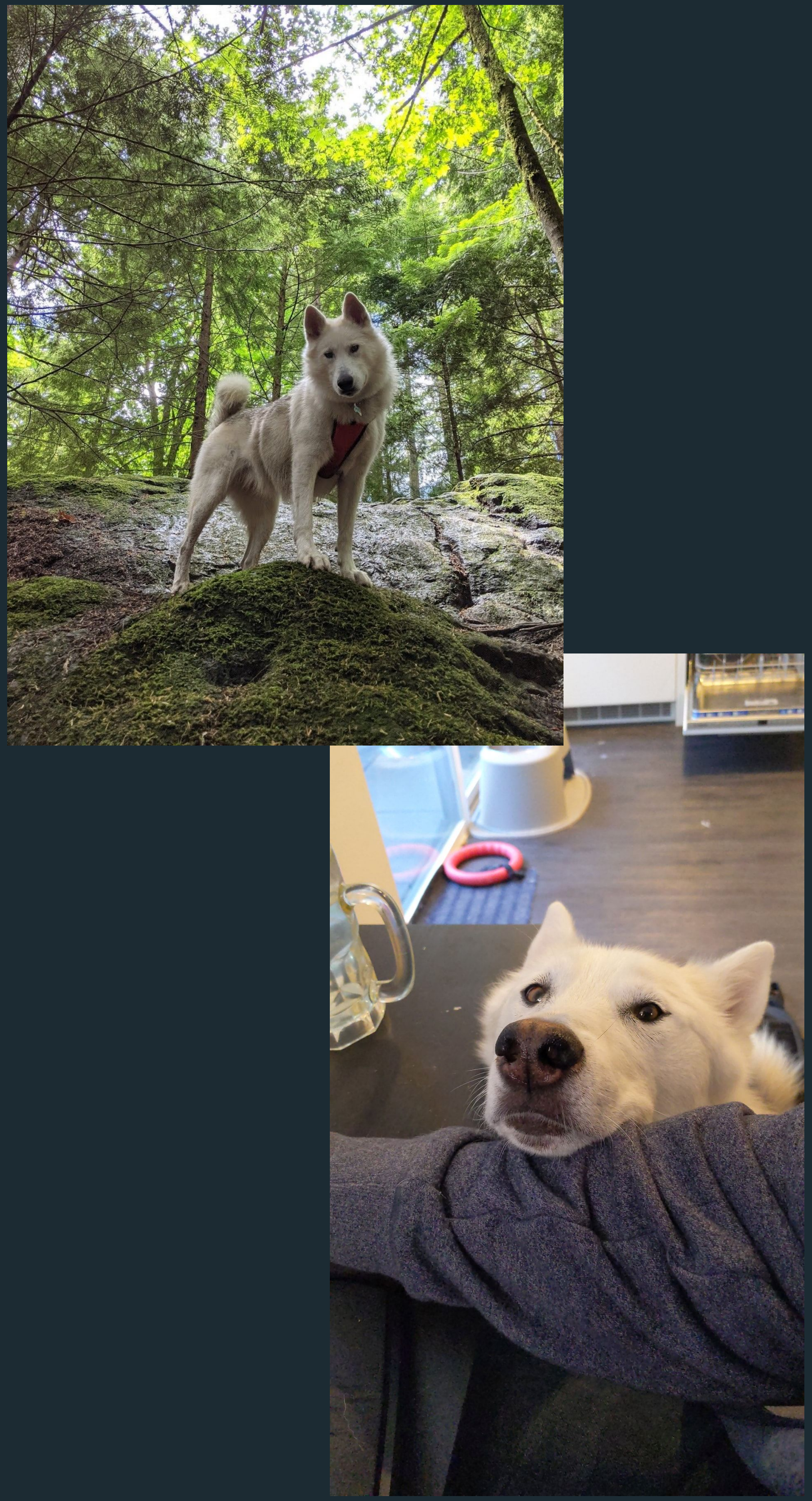

Metastable Failure
Both?
Cascading Failure

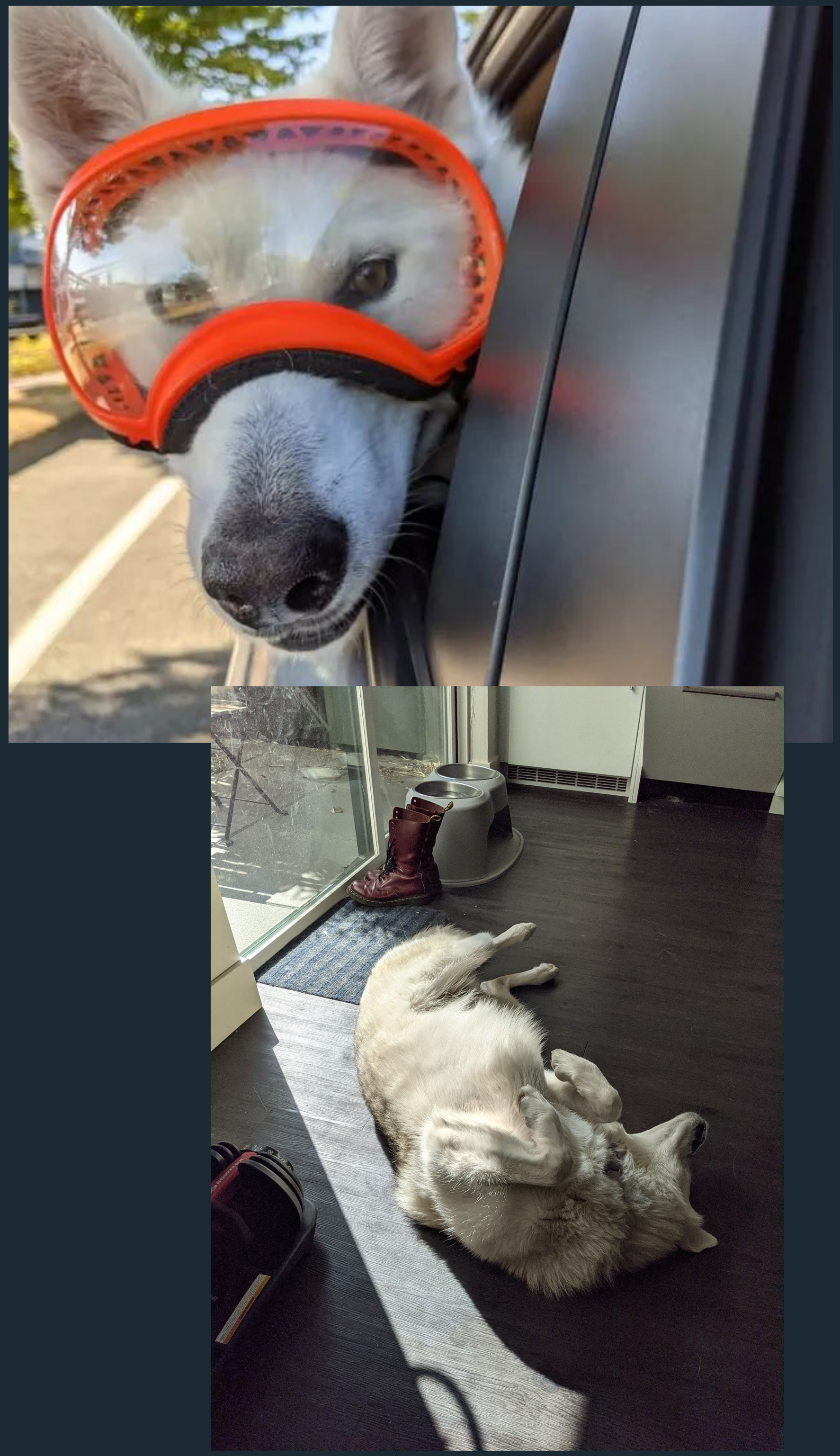
My Experience

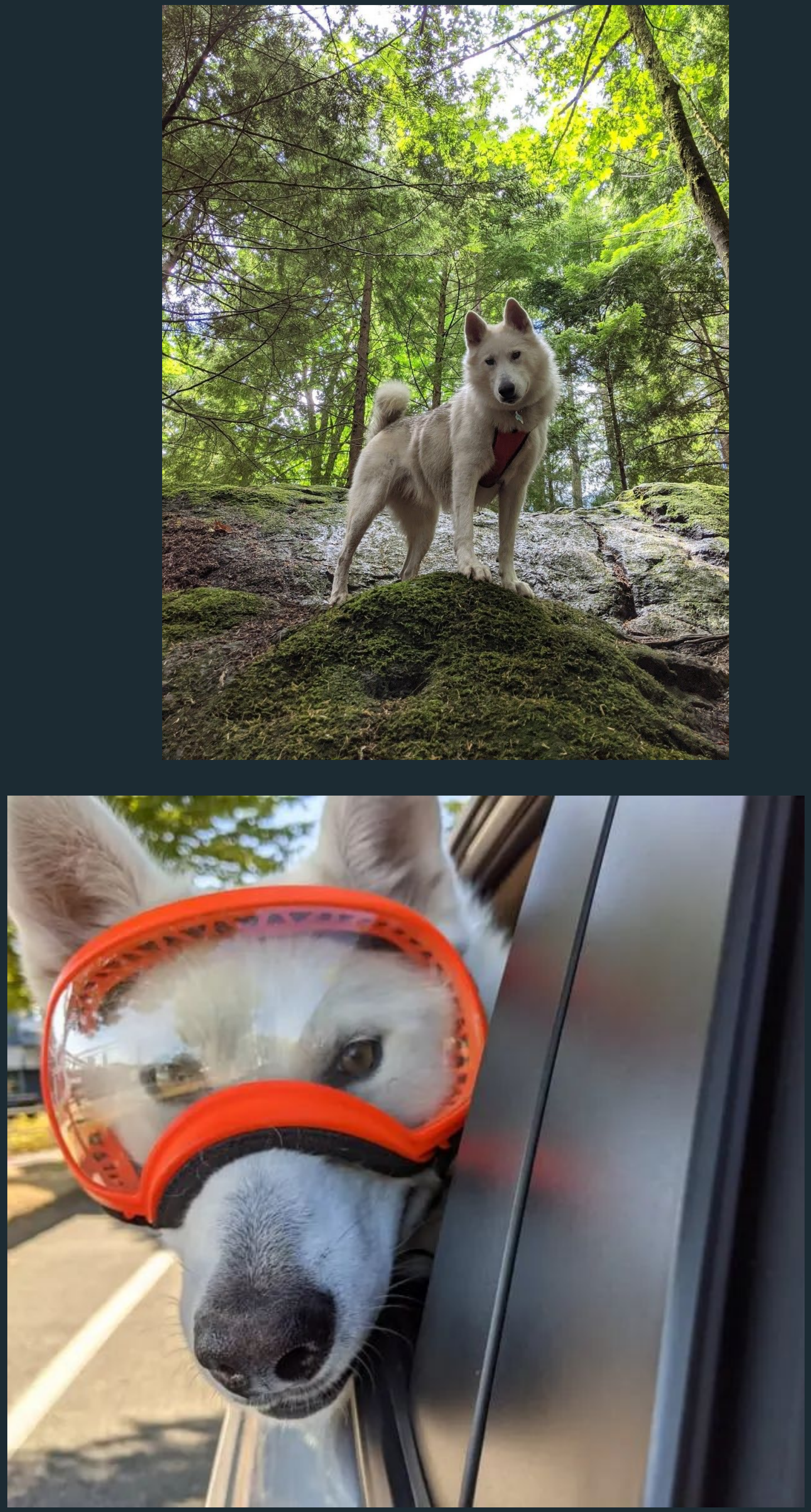
- Peak DR Load Test scheduled
- Alerts flooded the CDN team channel
- Focus on getting machines back up

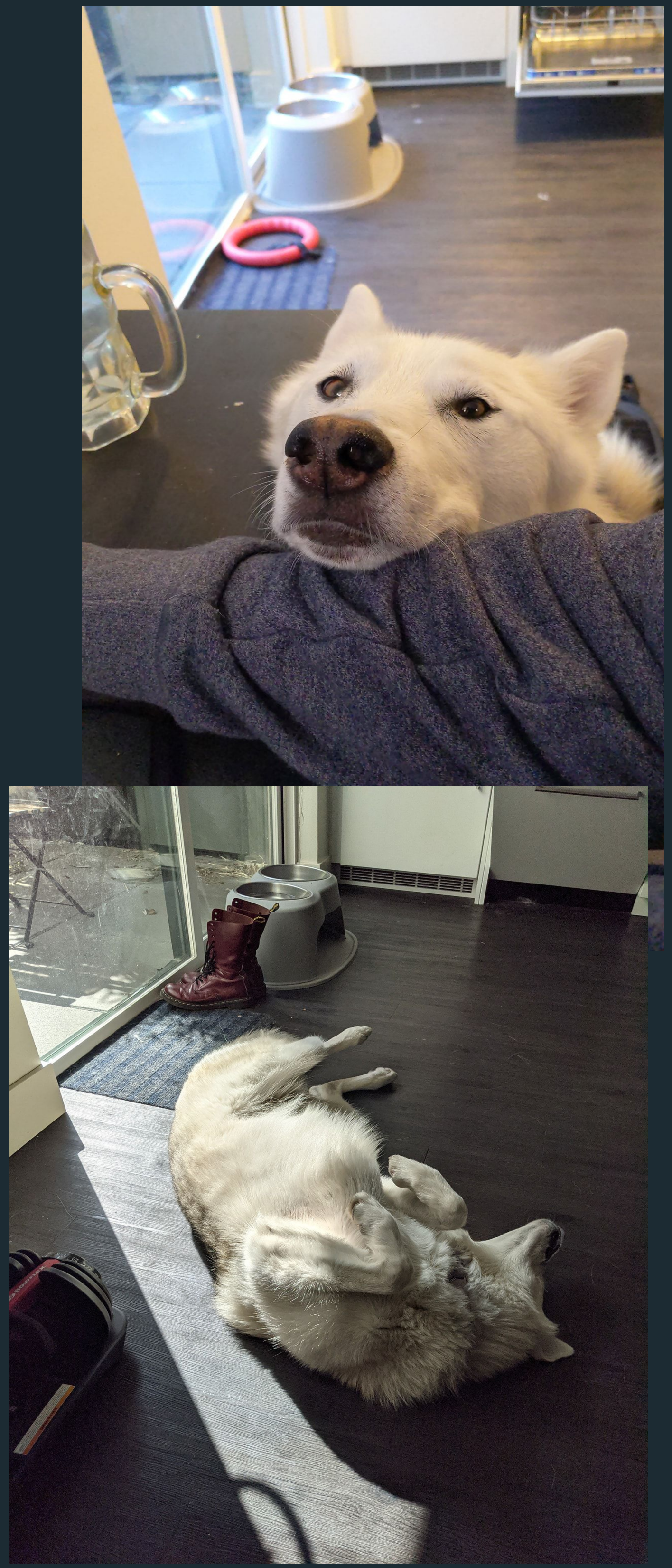
Architecture

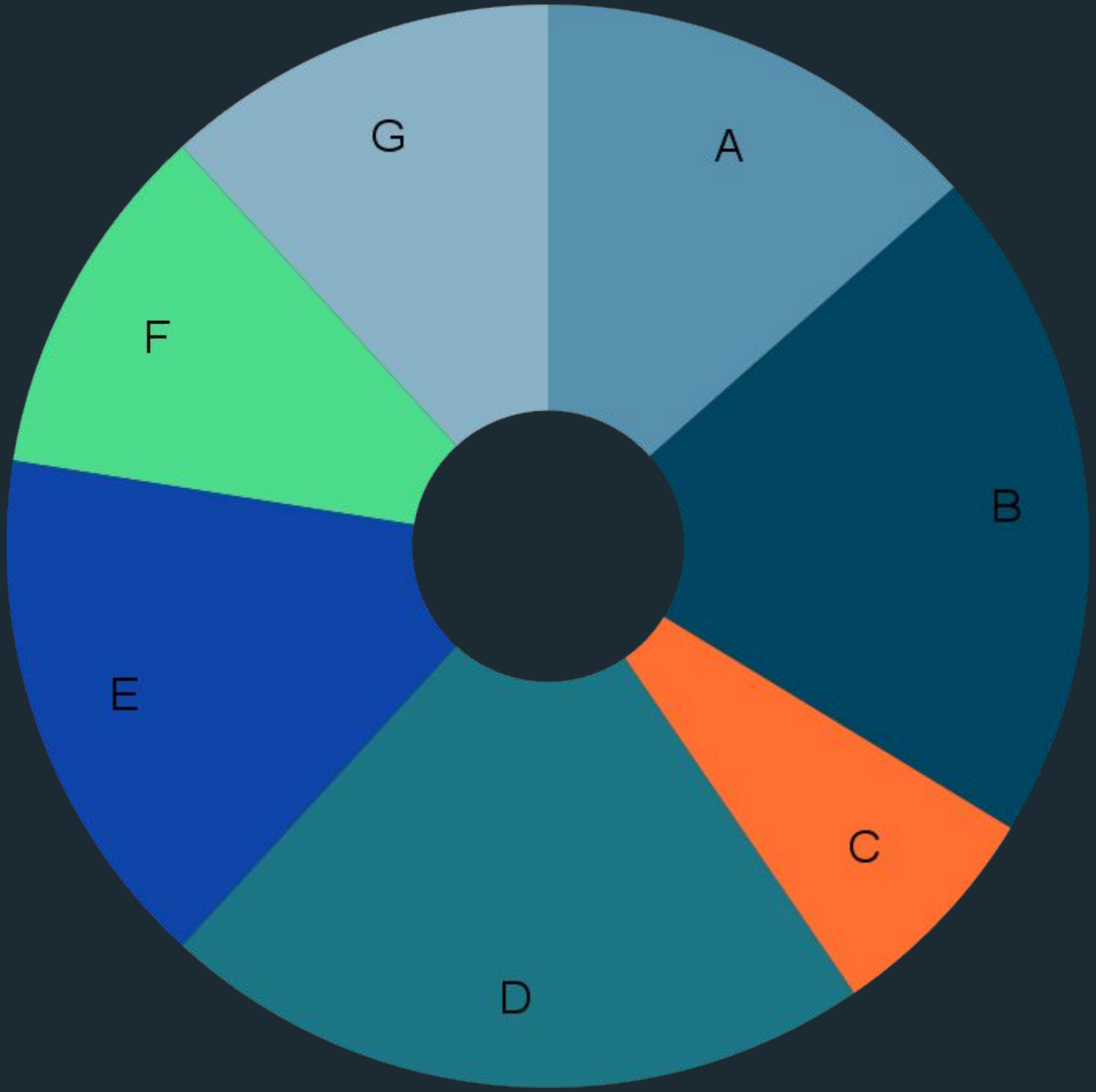
Uniqueness

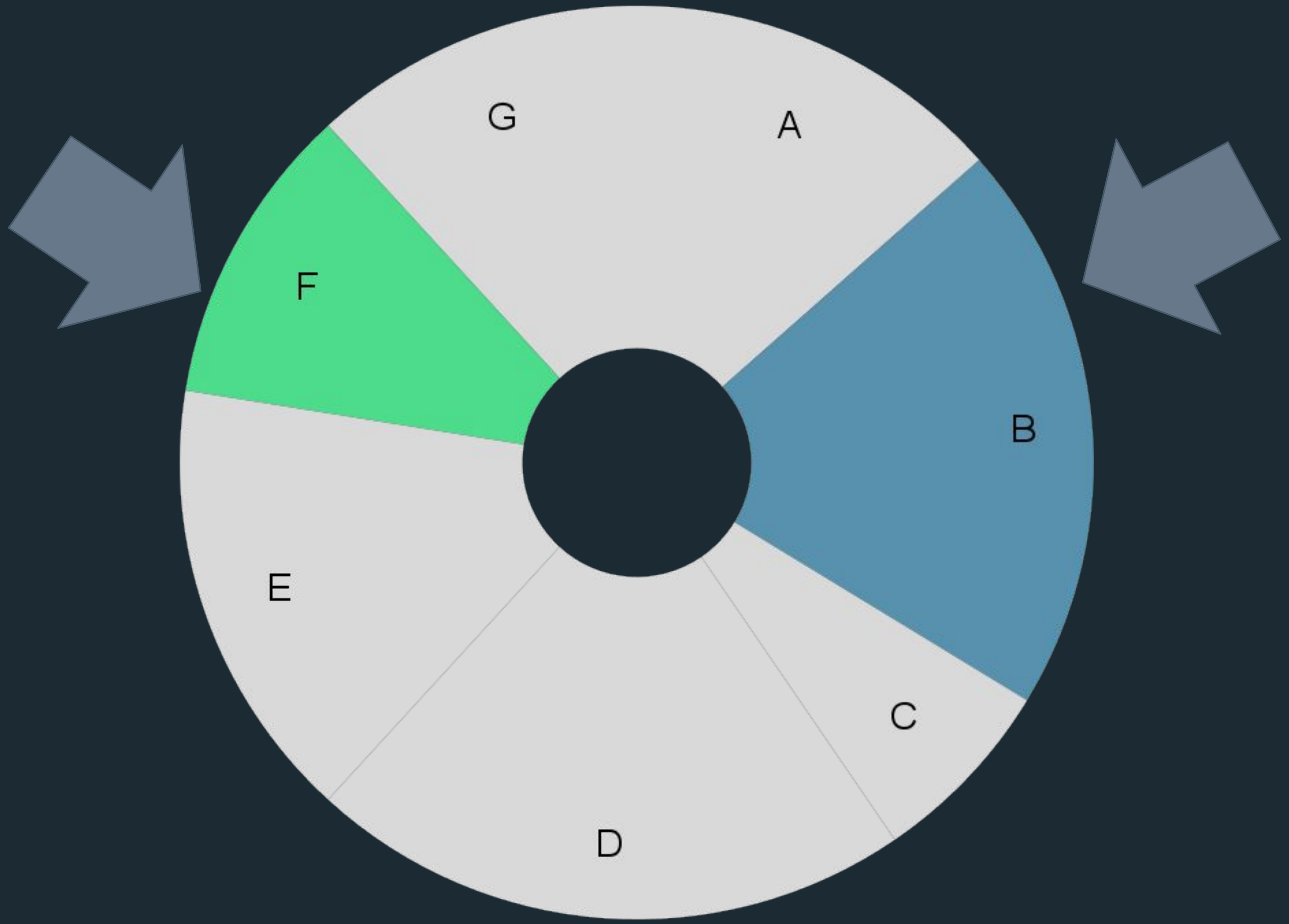

Region 1


Region 2

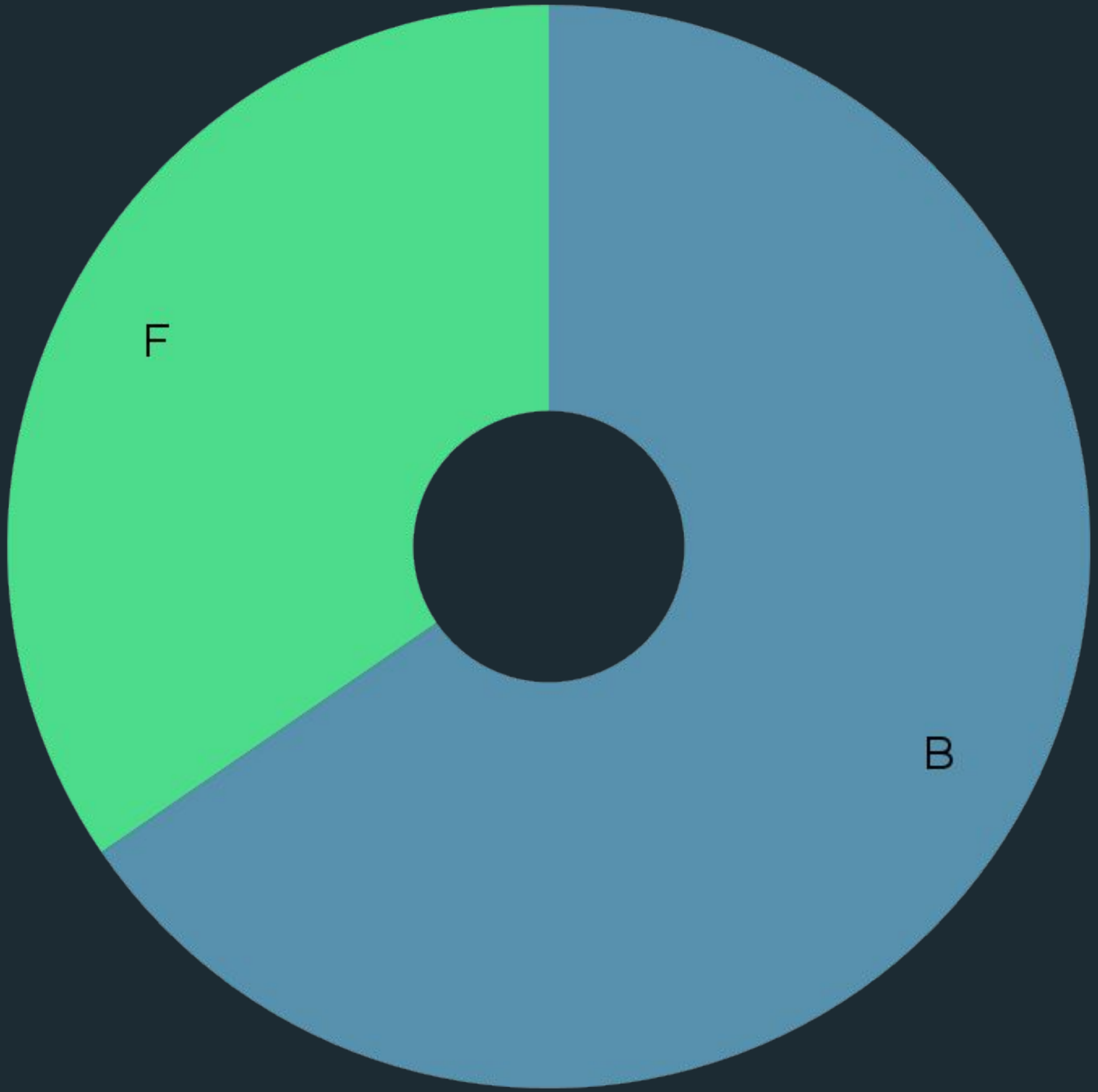

Region 3

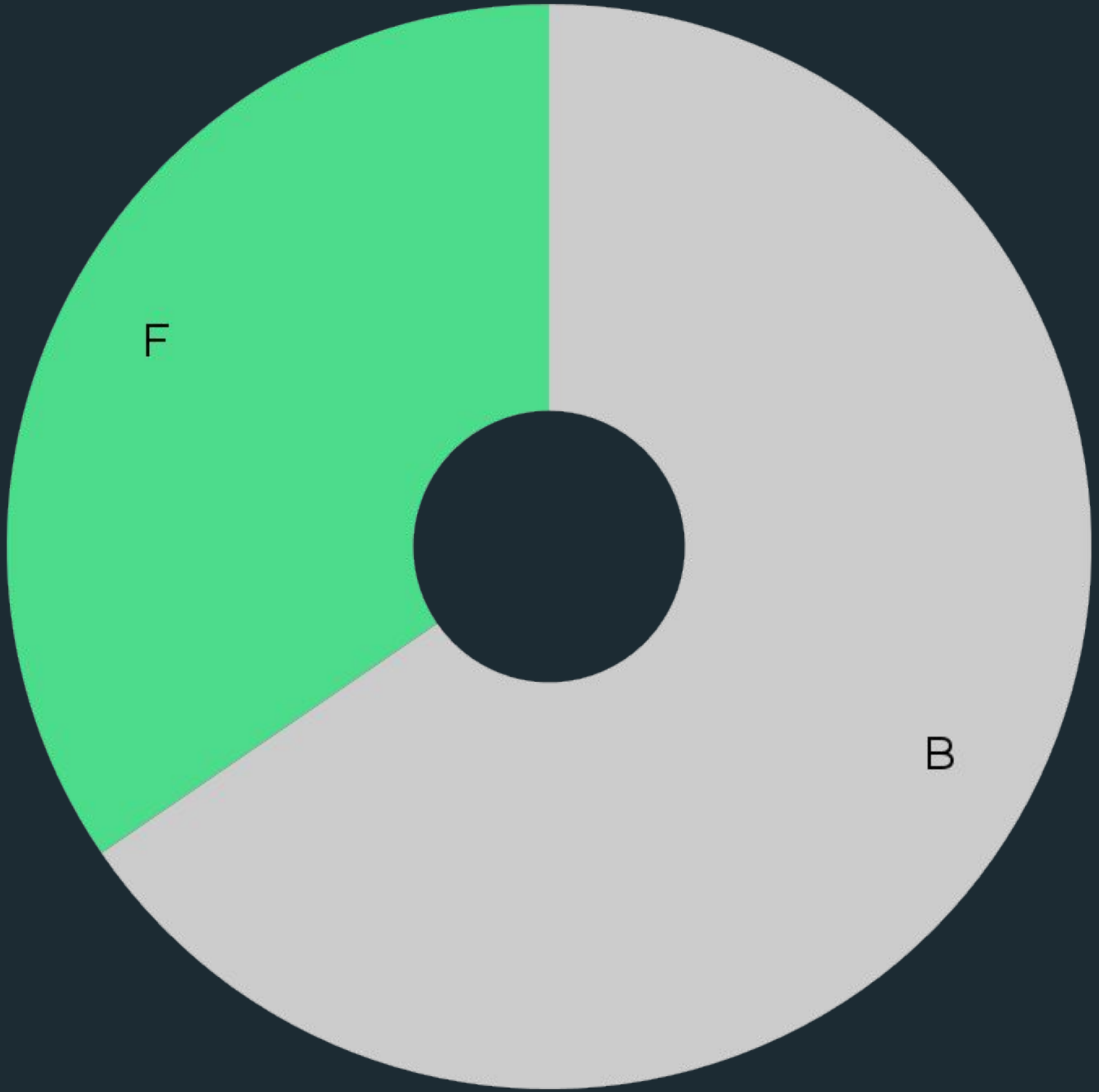

Region 4


G
A
F
B
E
C
D

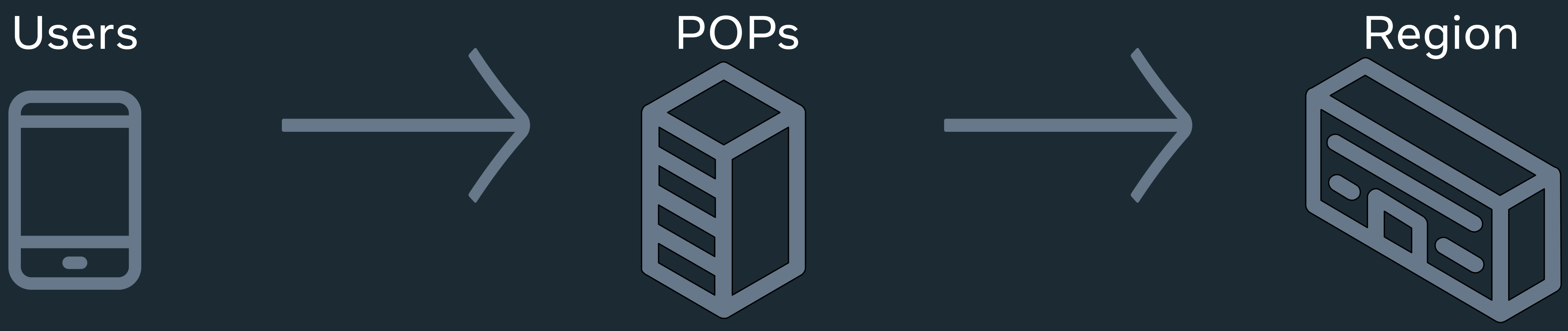

G
A
F
B
E
C
D

?

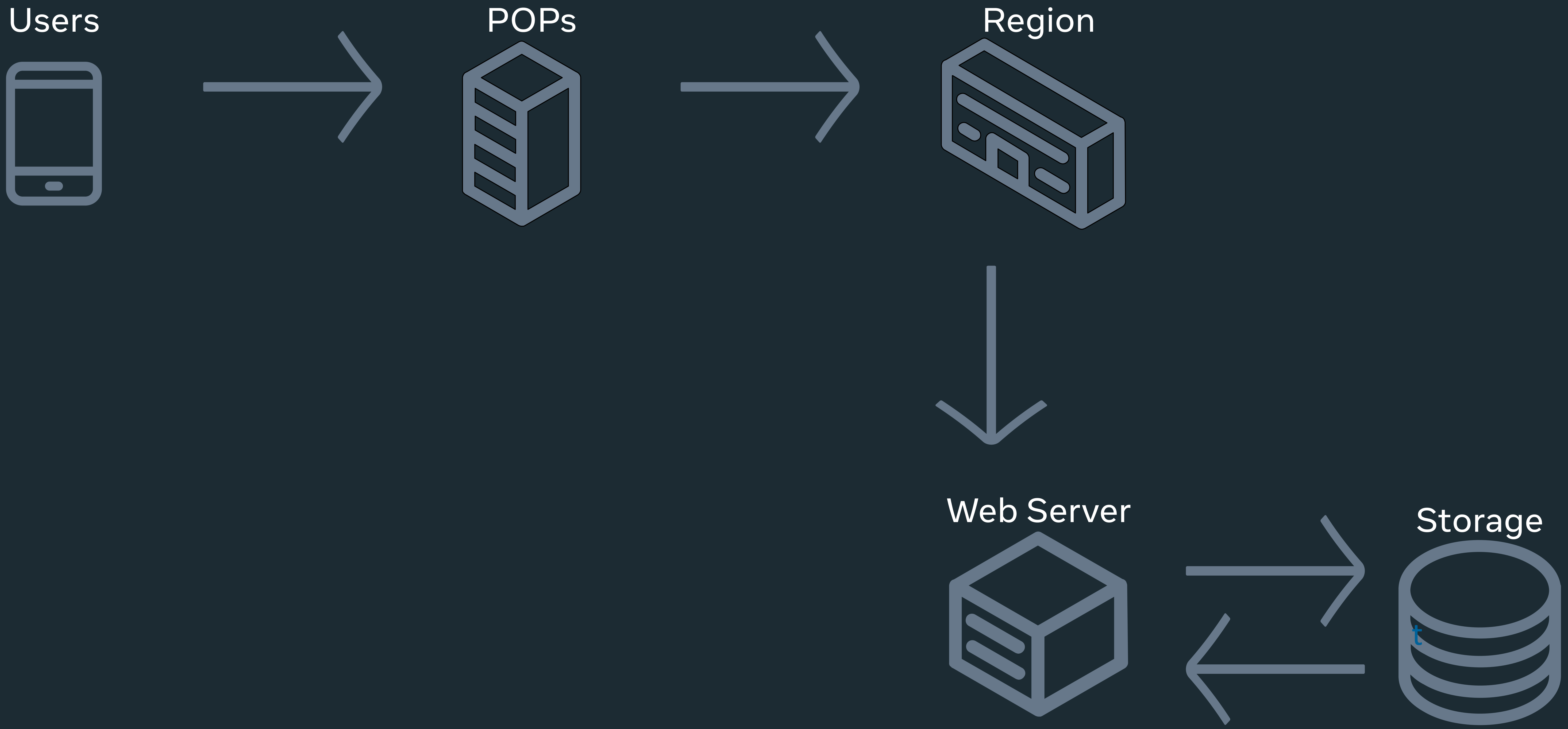

F
B


F
B

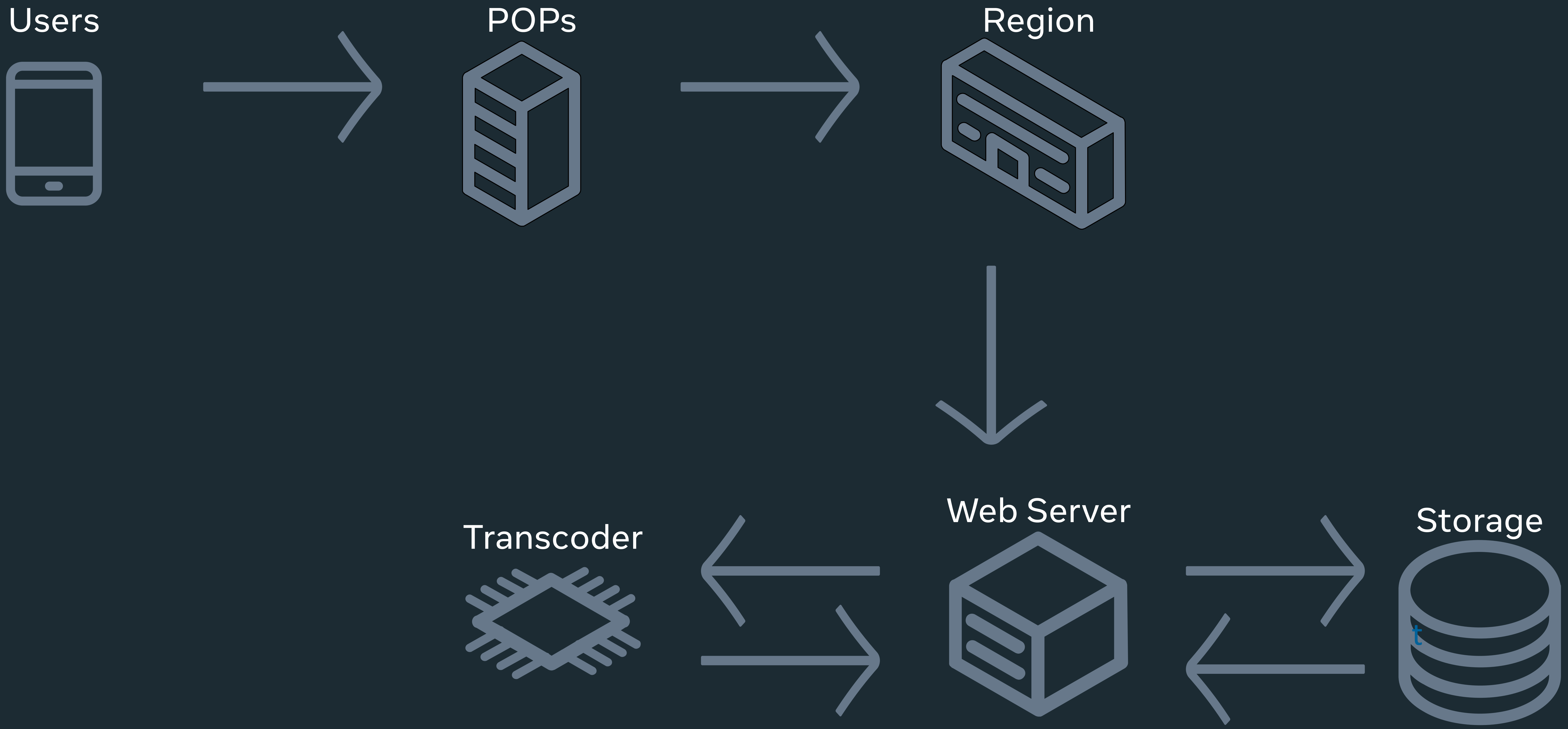
CDN Request Flow



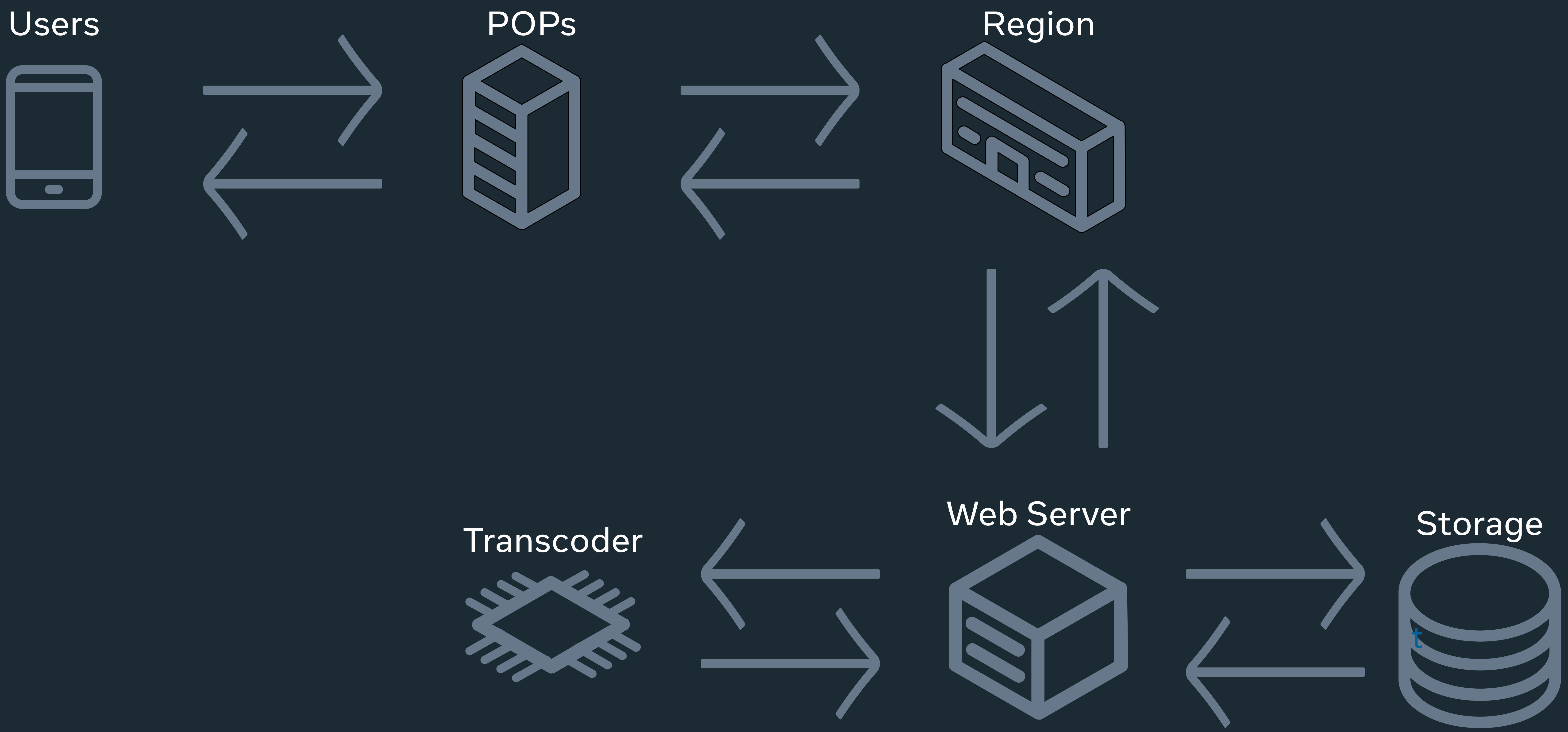
CDN Request Flow



CDN Request Flow



CDN Request Flow



Back to the Story

Recap

- Machines are in a cycle of failing/healthy/failing
- More than one region is impacted
- Cascading failure?

Is this a Metastable Failure?

- All 3 criteria met

- The DR test was reverted
- But we were failing a lot of requests
- Regions would get hit by the thundering herd on individual recovery

Recovery Attempts

Physical Recovery

- Power cycling machines
- Some machines ran out of memory and crashed
- Thundering herd on region recovery

Load shedding

- Tighten rate limits
- Disable background content fetch in apps
- Add additional web servers

Load shedding

- Tighten rate limits
- Disable background content fetch in apps
- Add additional web servers

- Changed healthchecks to ignore reality

Why did that work?

- Forced connections to be spread across machines
- No more “lasering” healthy instances
- Prevented the thundering herd

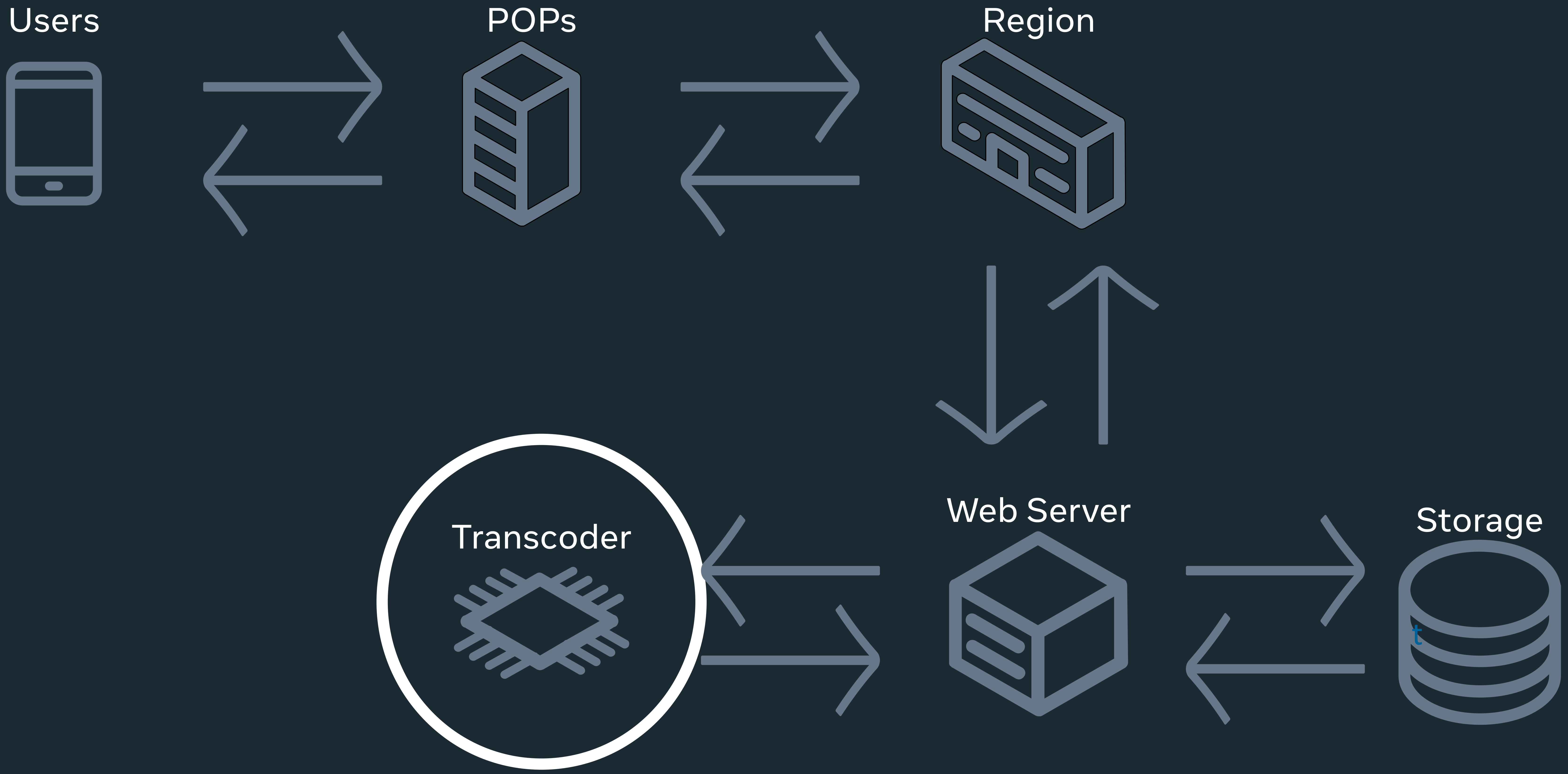
- Broke the sustaining effect

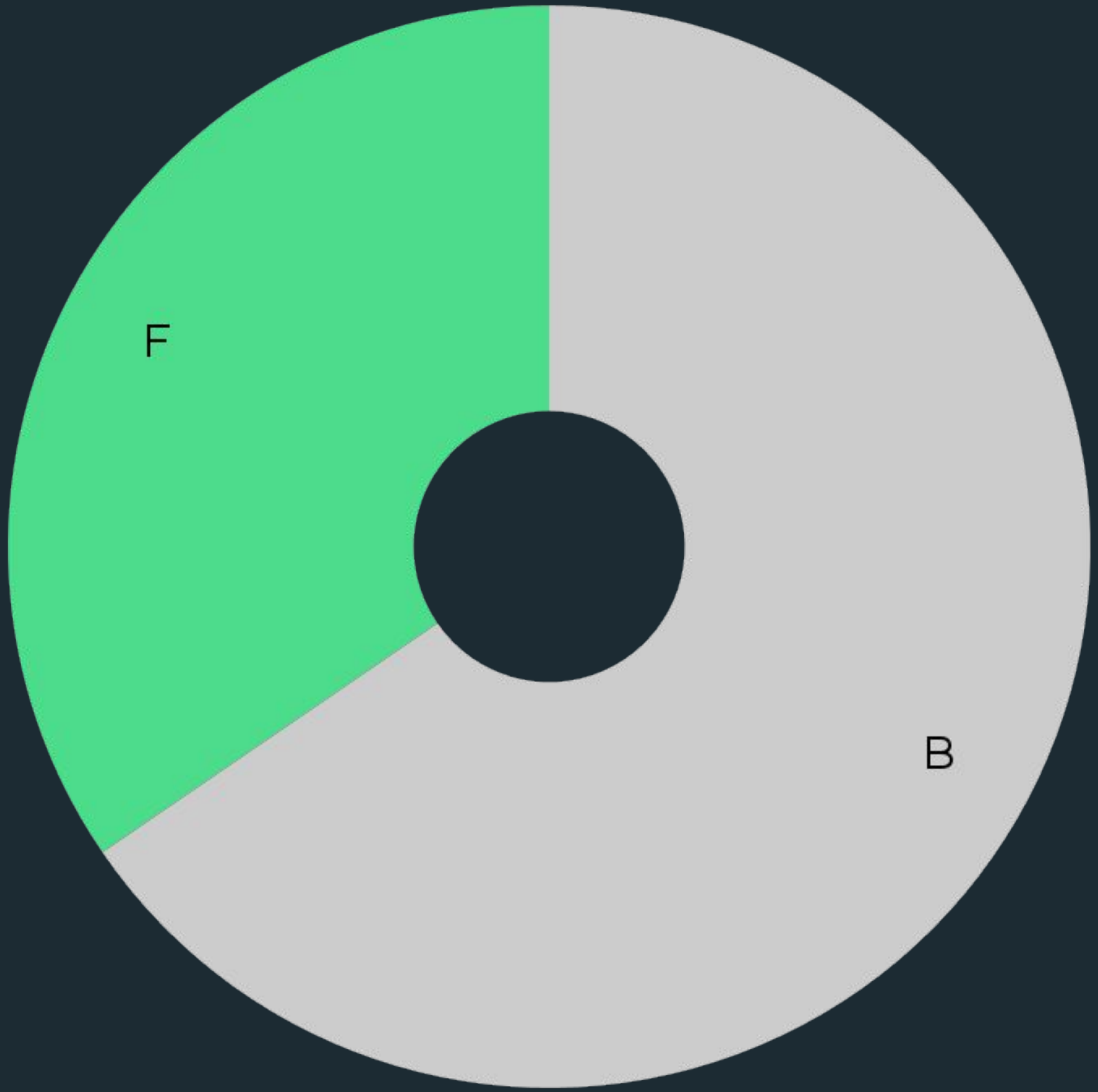
“Many load-balancing systems use a health check to send requests only to healthy instances, though you ***might need to turn that behavior off during an incident*** to avoid focusing all the load on brand-new instances as they are brought up.”

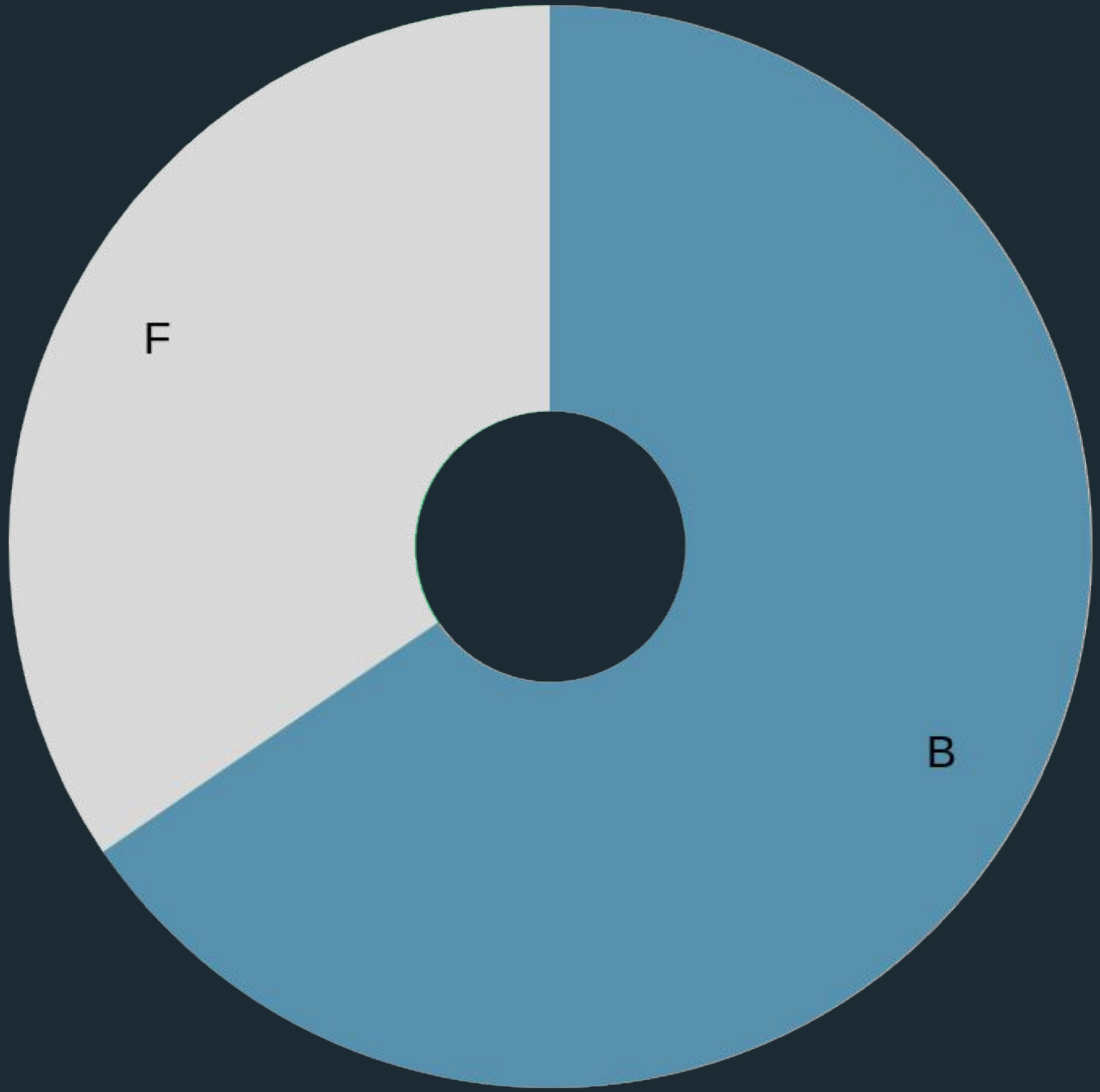
– Laura Nolan

What happened?

CDN Request Flow




F
B


F
B

Hindered & Helped

Hindered

Delayed realization of incident scale

Different failure modes confused us

Healthchecks were affected by load shedding, causing traffic oscillations

Helped

Failed system isolated from other services

Stale cached content continued to be served

Design Changes

Healthcheck Improvements

- Below a threshold, spread load evenly instead of overloading remaining machines

“Panic Threshold”

… if the percentage of available hosts in the cluster becomes too low, Envoy will disregard health status and balance either amongst all hosts or no hosts.

This is known as the panic threshold.

– Panic threshold, Envoy Proxy Documentation

https://www.envoyproxy.io/docs/envoy/latest/intro/arch_overview/upstream/load_balancing/panic_threshold

“Target group health: Unhealthy state actions”

When the healthy targets in a zone fall below the threshold, the load balancer sends traffic to all targets that are available to the load balancer node, including unhealthy targets.

This increases the chances that a client connection succeeds, especially when targets temporarily fail to pass health checks, and reduces the risk of overloading the healthy targets.

> **– Target group health: Unhealthy state actions, AWS ALB documentation**
>
> **https://docs.aws.amazon.com/elasticloadbalancing/latest/application/target-group-health.html**

Healthcheck Improvements

- Short term history, not only most recent check
- Prevent the health checks from being subject to load shedding

“The most important request that a server will receive is a ping request from a load balancer.”

– David Yanacek

CDN Design

- Rebalance routing weights to reduce magnitude of traffic shifts under DR
- Periodic automated calculation of weights based on demand & capacity

Miscellaneous

- New automation to support tier isolation
- Stronger container-level isolation & resource limits
- Accelerated effort to add autoscaling the transcoder & web servers

Takeaways

Metastable failures occur when a system fails under an increase in load, and can’t self-recover due to a sustaining effect

Load shed by disabling healthchecks & force “normal” traffic spread

Spread traffic across all services if there’s a large % of failures to avoid overloading surviving services

Questions?

THANK YOU FOR YOUR TIME

@lightweaver@hachyderm.io

SCHWINN

Meta