UNBREAKABLE:

LEARNING TO BEND BUT NOT BREAK AT NETFLIX

NETFLIX ORIGINAL

UNBREAKABLE KIMMY SCHMIDT

91% Match 2017 TV-14 3 Seasons

Season 4 Coming May 30

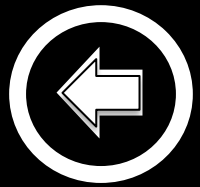
S1:E1 "Kimmy Goes Outside!"

Imprisoned by a cult leader as a teenager, Midwesterner Kimmy is freed after 15 years. The first thing she decides to do is move to New York.

▶ NEXT EPISODE + MY LIST

Crea[illegible] Robert Carlock


NETFLIX
24:40
Unbreakable Kimmy Schmidt S1:E1 Kimmy Goes Outside!



Whoops, something went wrong…

Netflix Streaming Error

We're having trouble playing this title right now. Please try again later or select a different title.

pili #LoSiento #SS7inArgentina
@piligarciac

Follow

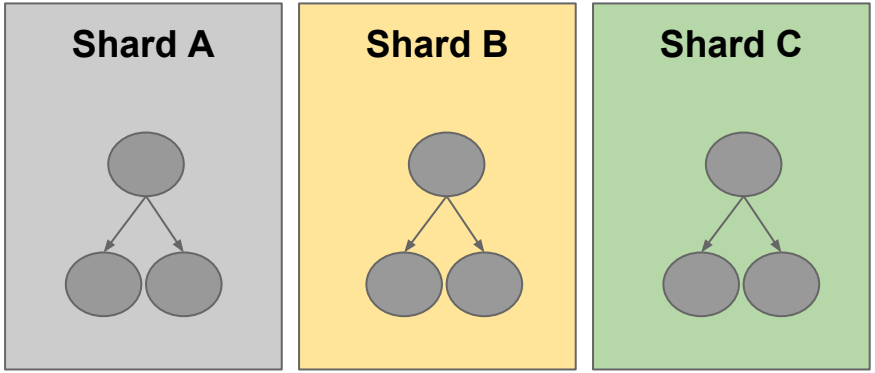
What's up with Netflix? My Kimmy Schmidt! #netflixdown

3:10 PM - 29 Sep 2016

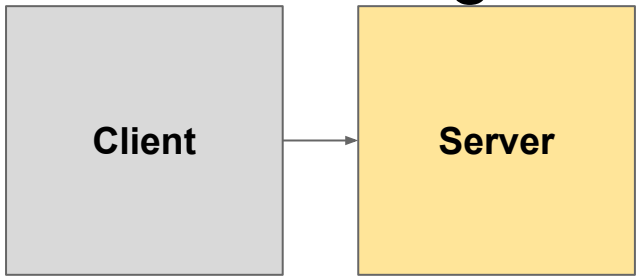
7 Likes

7

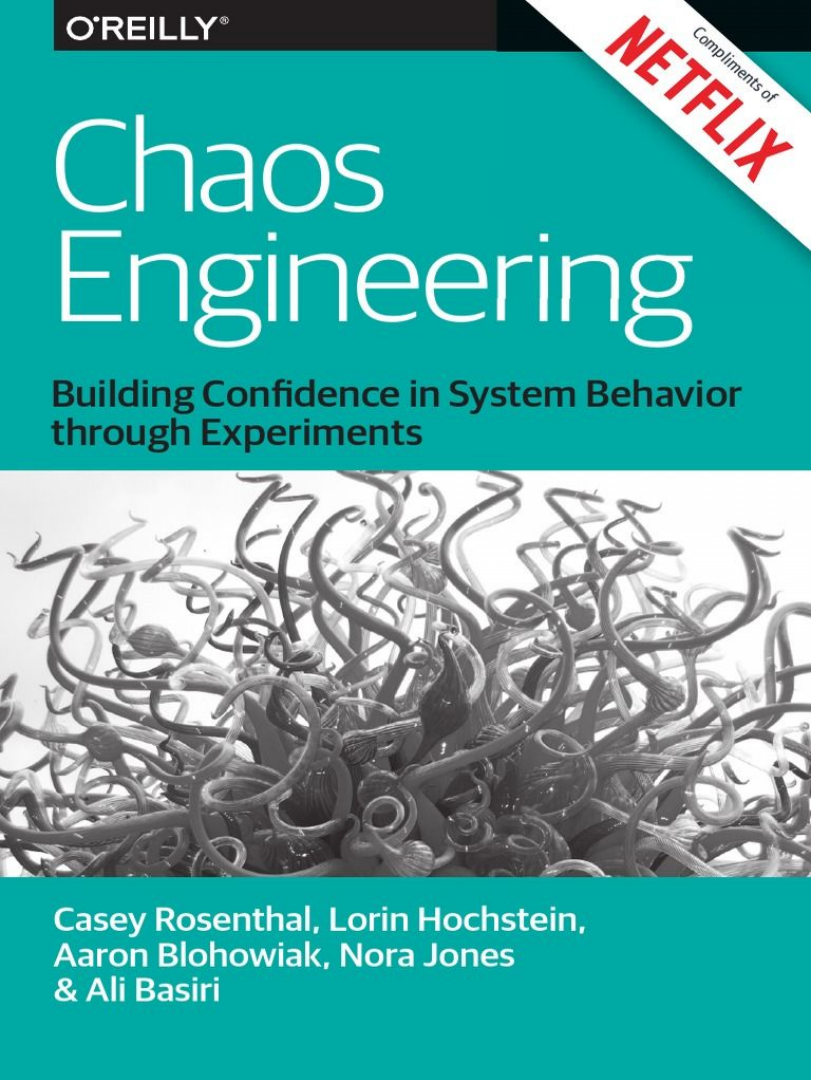
Functional Sharding



RPC tuning

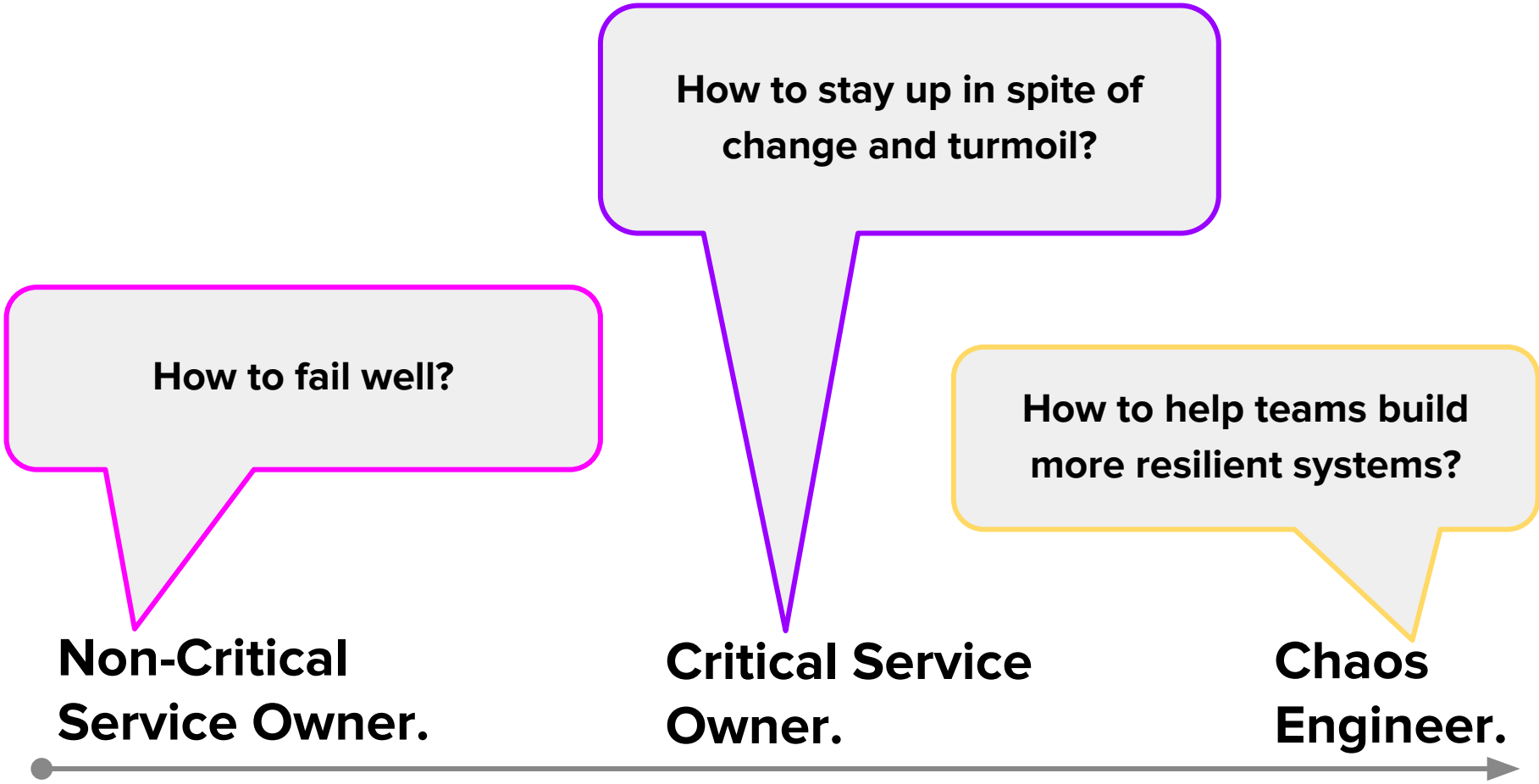


Bulkheads & Fallbacks


O'REILLY®
Compliments of
NETFLIX
Chaos Engineering
Building Confidence in System Behavior through Experiments
Casey Rosenthal, Lorin Hochstein, Aaron Blohowiak, Nora Jones & Ali Basiri


Chaos Monkey


ChAP

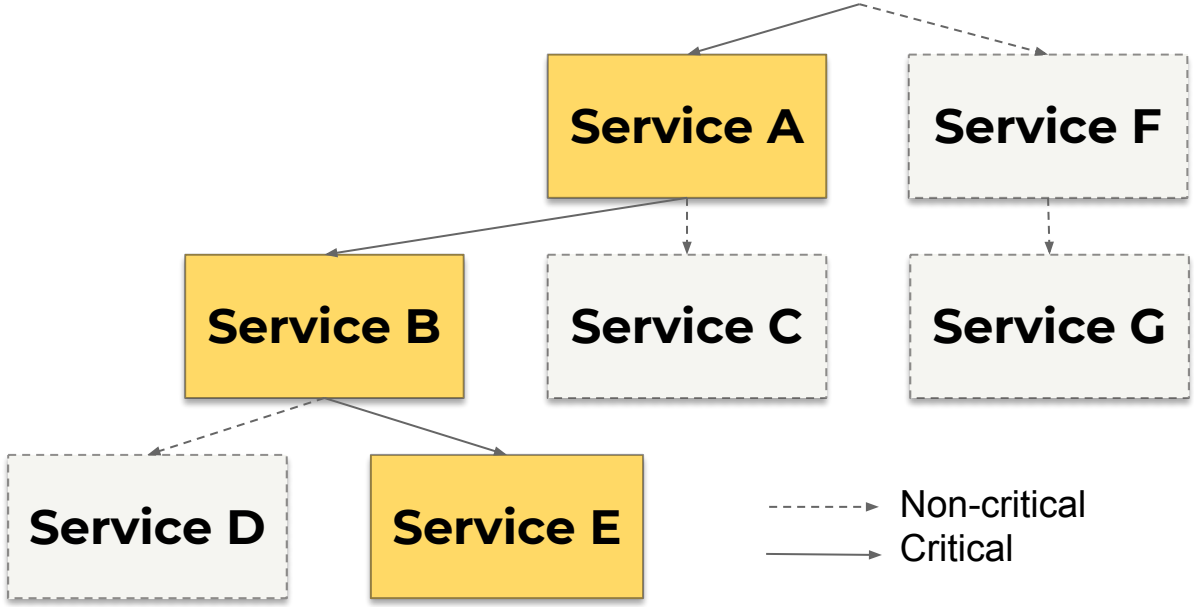

How to fail well?
How to stay up in spite of change and turmoil?
How to help teams build more resilient systems?
Non-Critical Service Owner.
Critical Service Owner.
Chaos Engineer.

Service Criticality

Driver_free_car.jpg, CC BY-SA 3.0, BP63Vincente 2015, Wikimedia

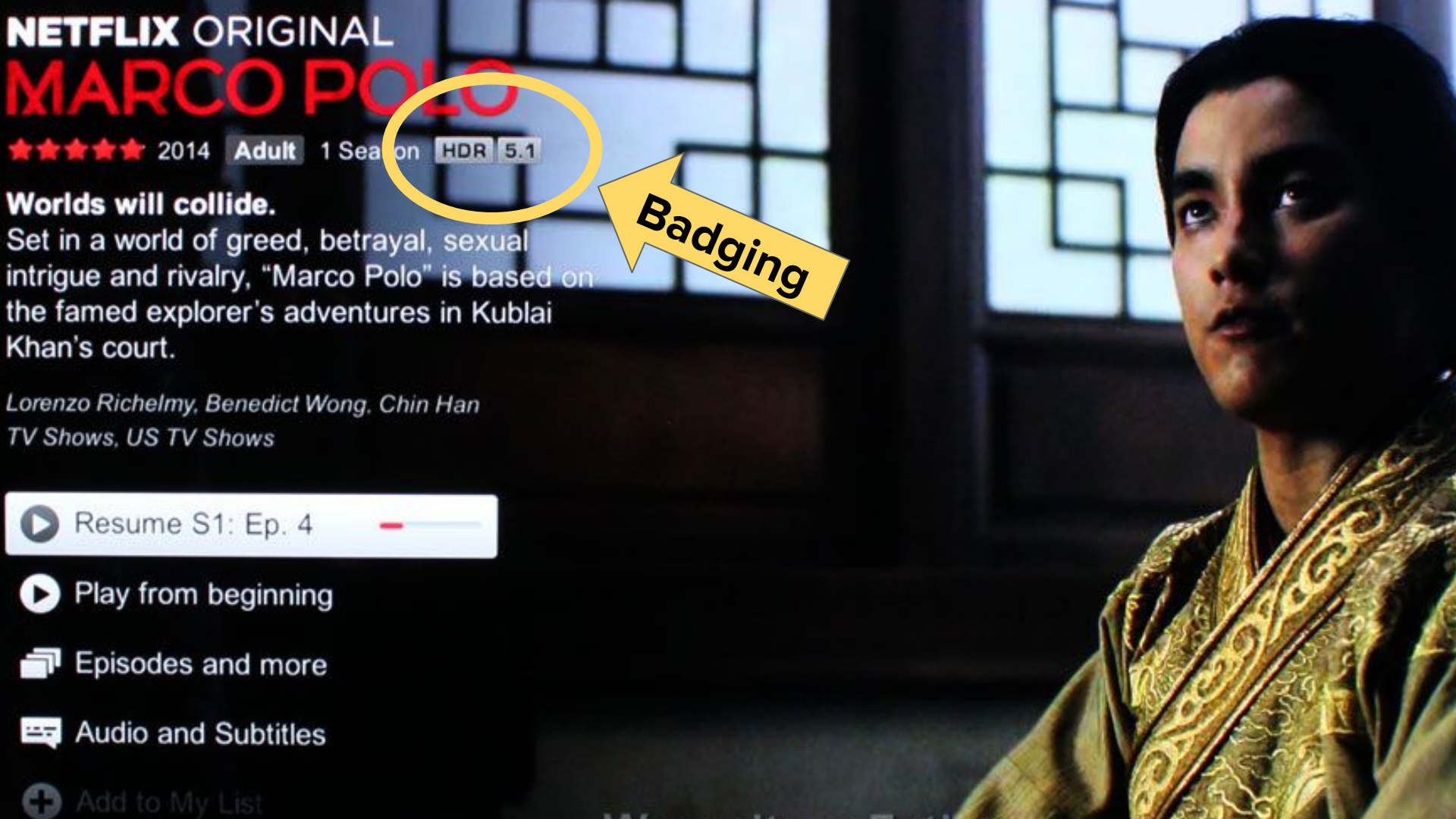
Service Criticality

KPI = Playback Starts Per Second (**SPS**)




If you're confident
on the outside,


Non-Critical Service Owner.
Critical Service Owner.
Chaos Engineer.


NETFLIX ORIGINAL
MARCO POLO
2014
Adult
1 Season
HDR
5.1
Worlds will collide.
Set in a world of greed, betrayal, sexual intrigue and rivalry, “Marco Polo” is based on the famed explorer’s adventures in Kublai Khan’s court.
Lorenzo Richelmy, Benedict Wong, Chin Han
TV Shows, US TV Shows
Resume S1: Ep. 4
Play from beginning
Episodes and more
Audio and Subtitles
Add to My List
Badging

My service is non-critical, who needs Chaos?

How do you *know* your service is non-critical?

HYSTRIX

DEFEND YOUR APP

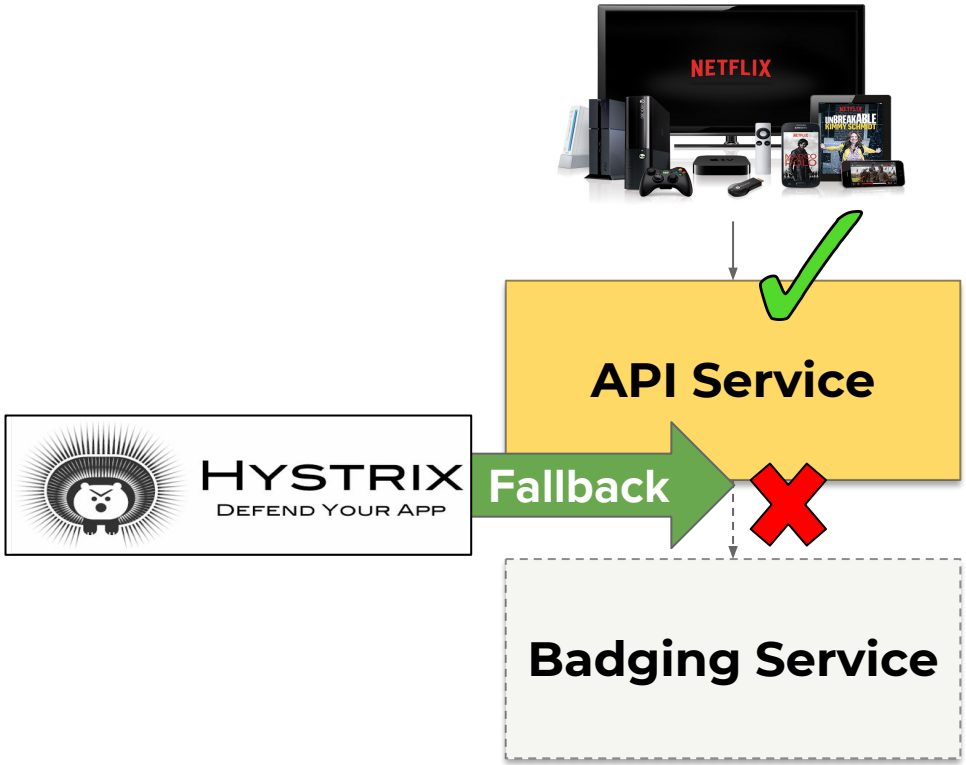
https://github.com/Netflix/Hystrix

Insights
Timeouts

Circuit Breakers

Bulkheads
Fallbacks

Badging Service (Non-Critical)



Svenn Amish
@amishschool

Follow

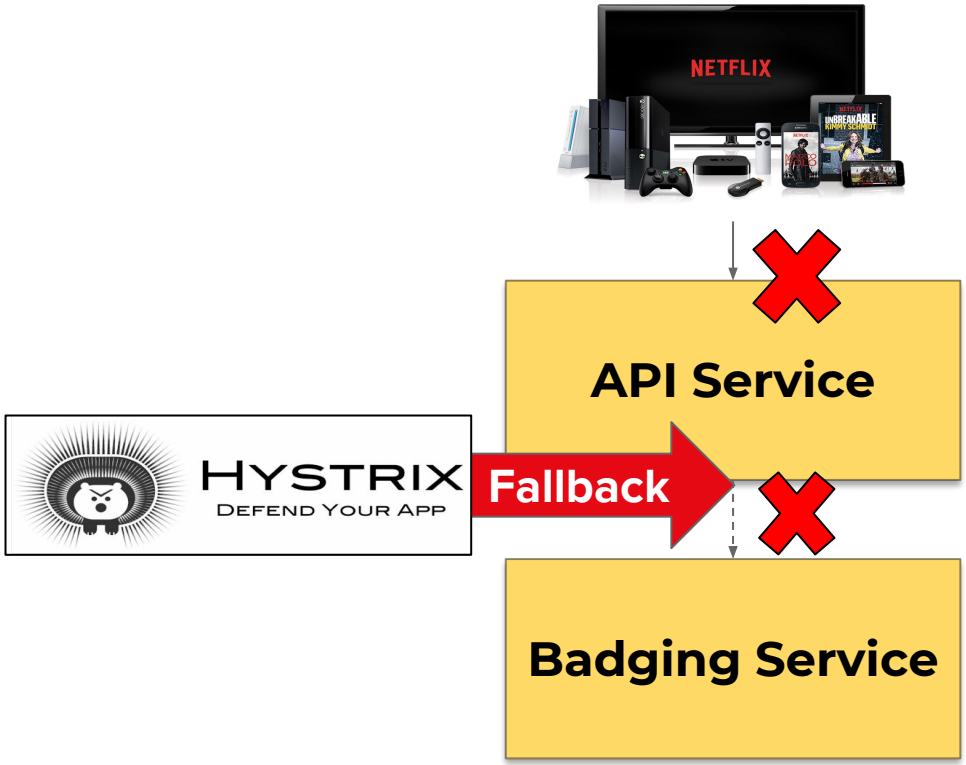
With Netflix down I had to make small talk with my kids with questions like "how was school?" and "what's your name again?"

RETWEETS 8 LIKES 26

9:50 PM - 21 Oct 2016

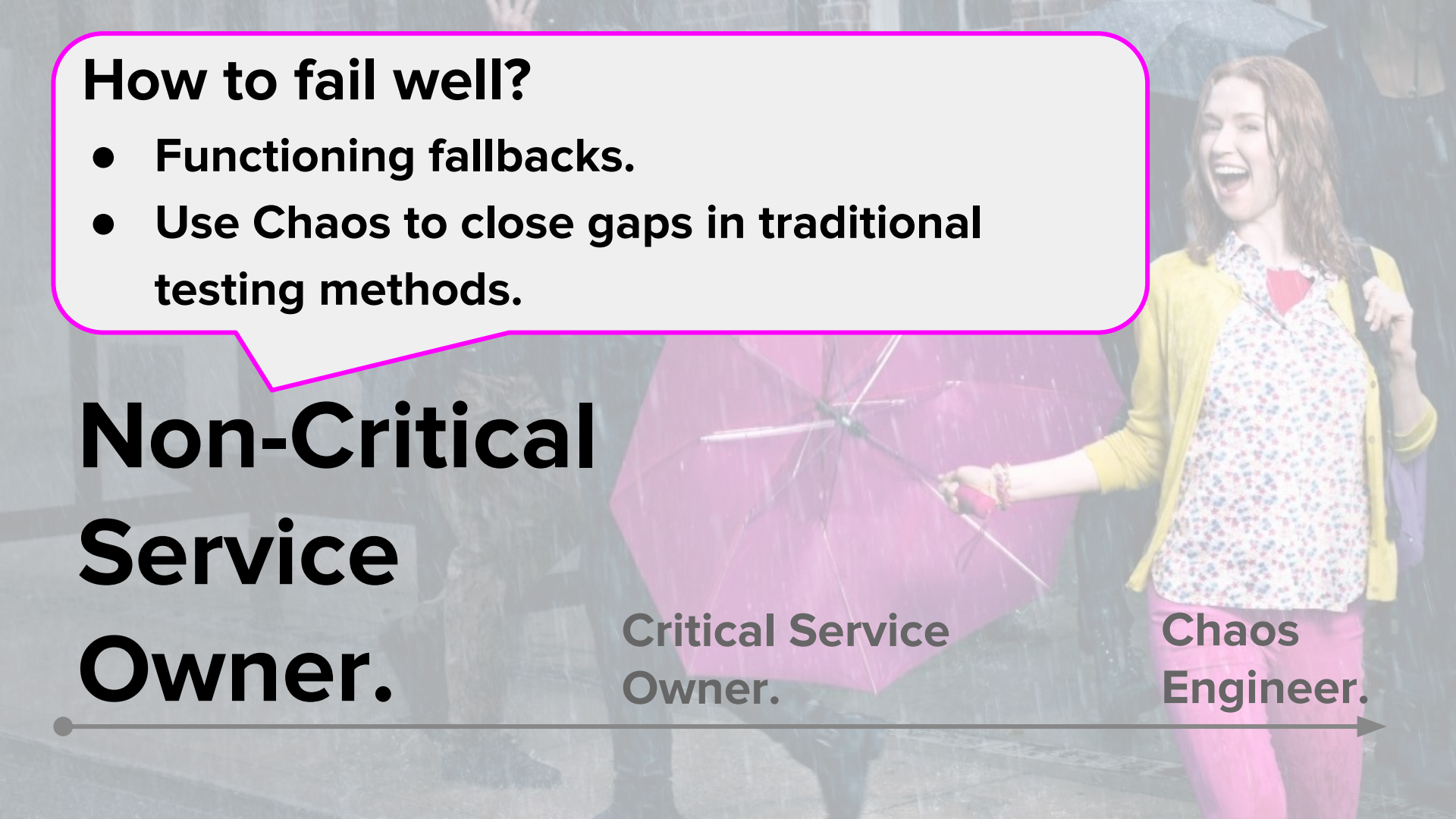
8 26

Surprise! Badging is Critical!



Gaps in Traditional Testing

- Environmental factors may differ between test and production (config, data, etc.)
- Systems behave differently under load than they do in a single unit or integration test
- Users react differently to failures than you expect.



How to fail well?

- Functioning fallbacks.
- Use Chaos to close gaps in traditional testing methods.

Non-Critical Service Owner.

Critical Service Owner.

Chaos Engineer.

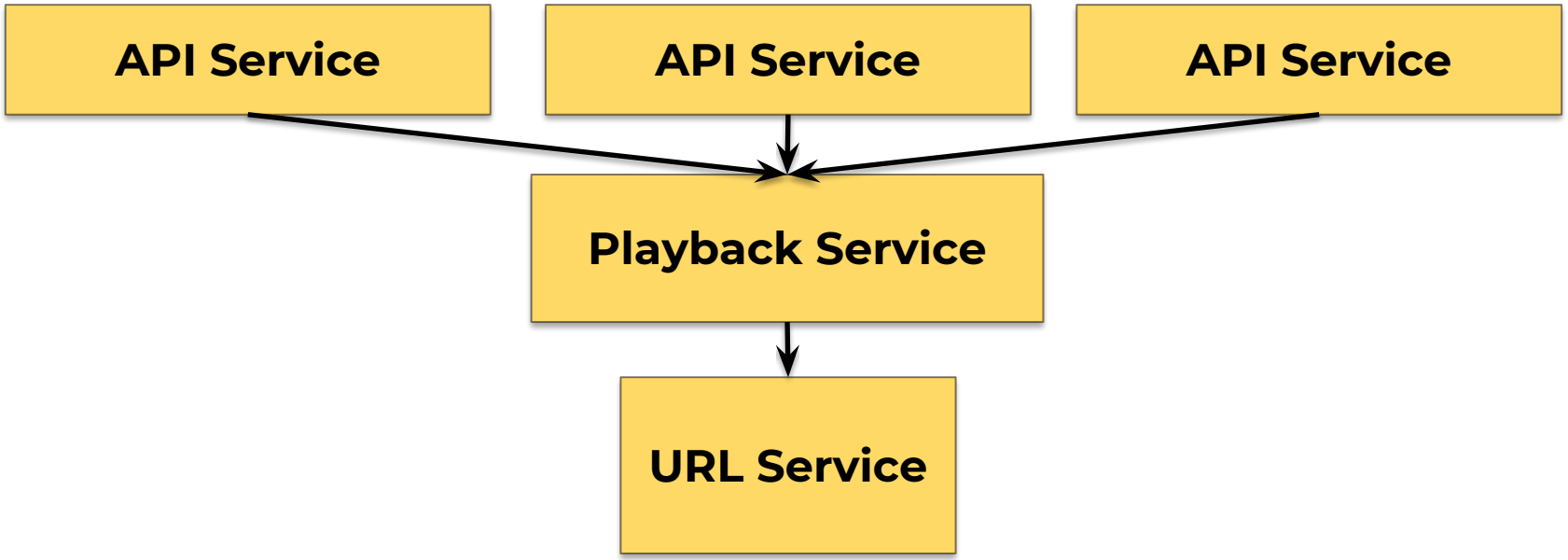

Critical
Service
Owner.
Non-Critical
Service Owner.
Chaos
Engineer.

Protect your service
(and your customers)

How can I decrease the blast radius of failures?

How about functional sharding!

Playback Service Architecture



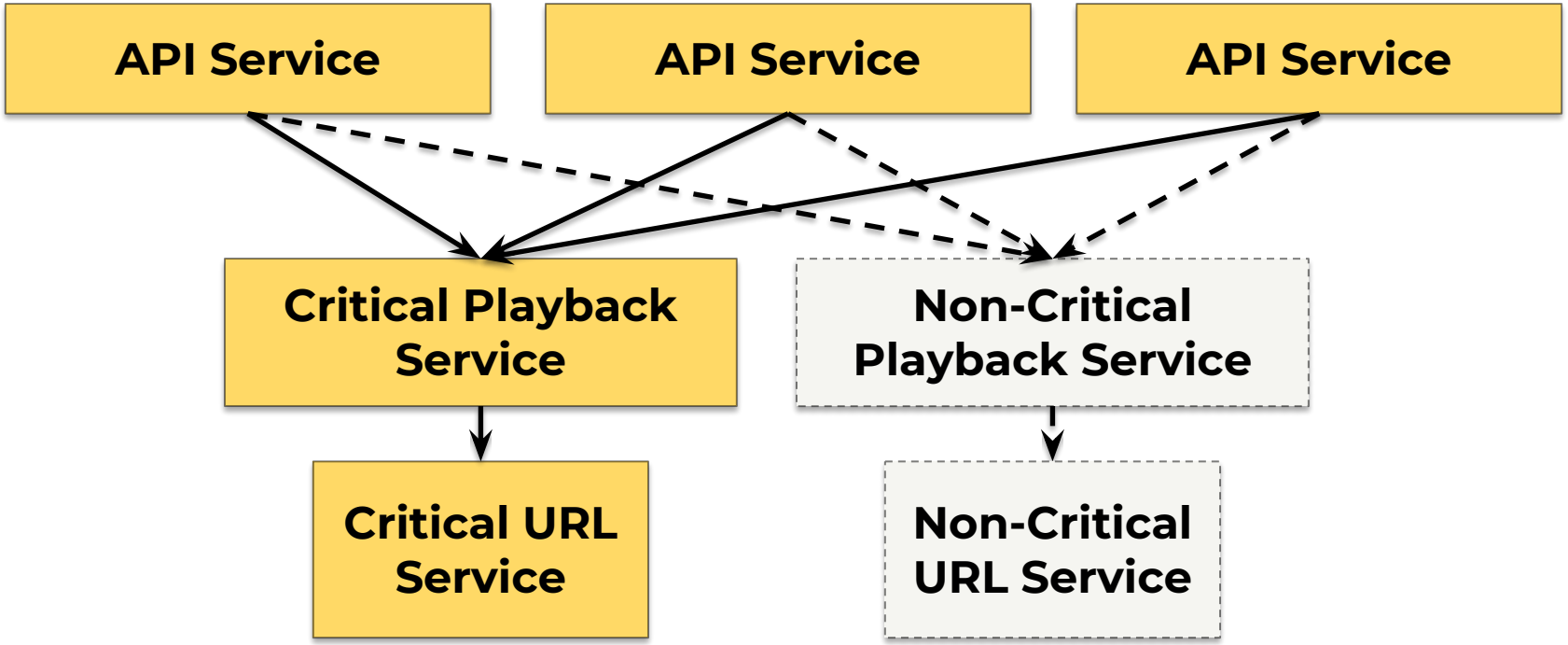
CRITICAL

Customer Streaming Impact

NON-CRITICAL

Experience or Performance Impact

Playback Service Functional Shards

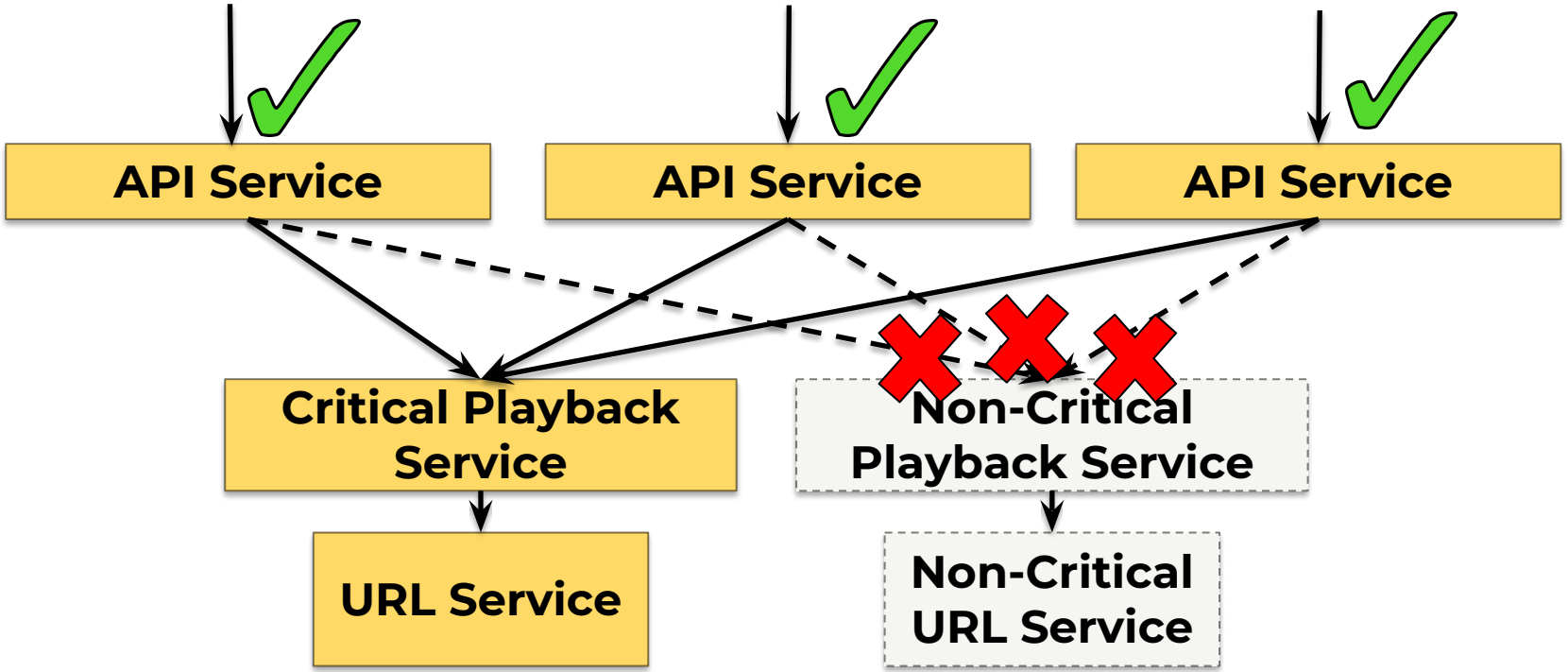


STAND BACK

I'M GOING TO TRY SCIENCE

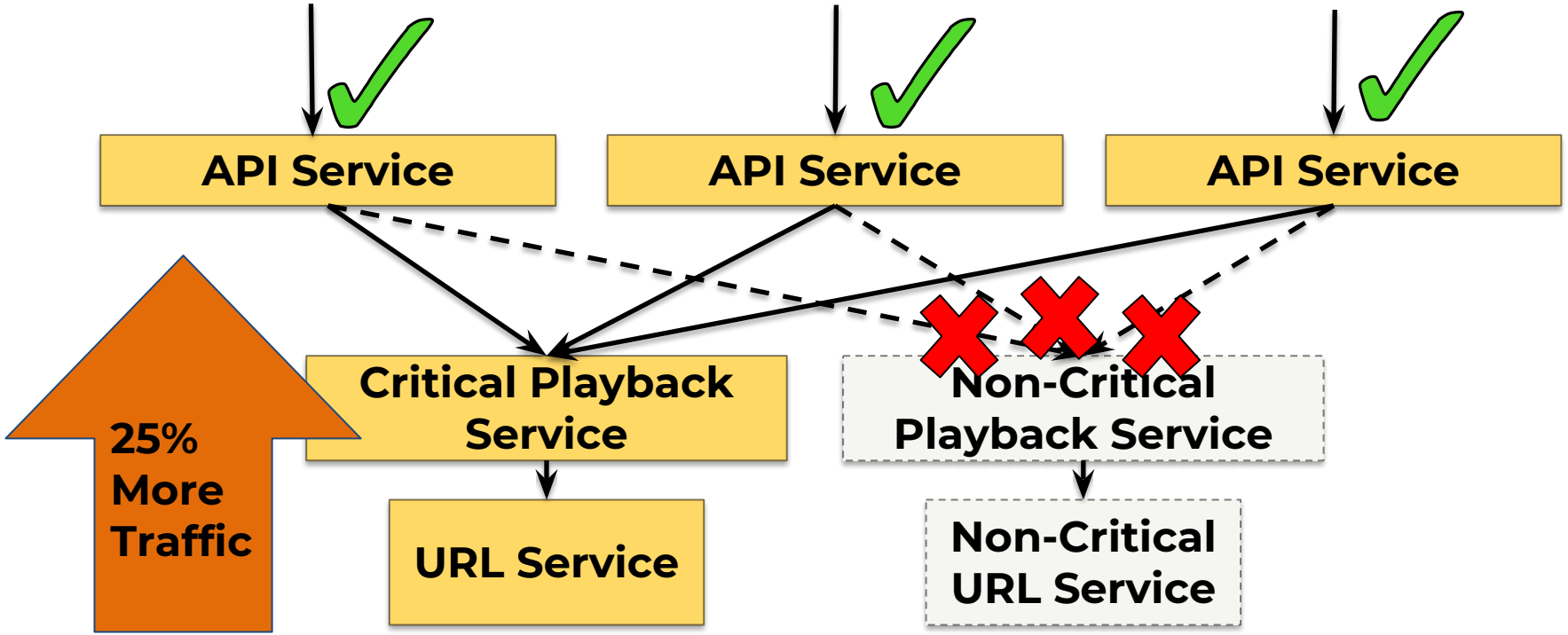
CC BY-NC 2.5, Randall Munroe, xkcd.com

Experimenting with Shards




NETFLIX

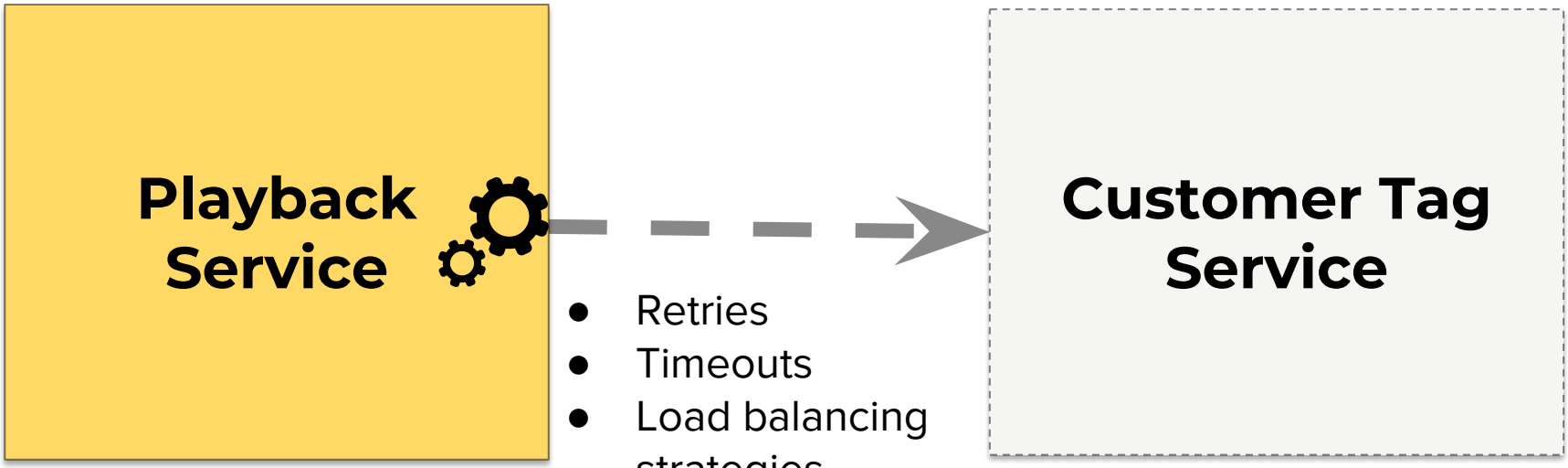
Customer Behavior Insights



How do I confirm my system is tuned properly?

Inject latency, of course!

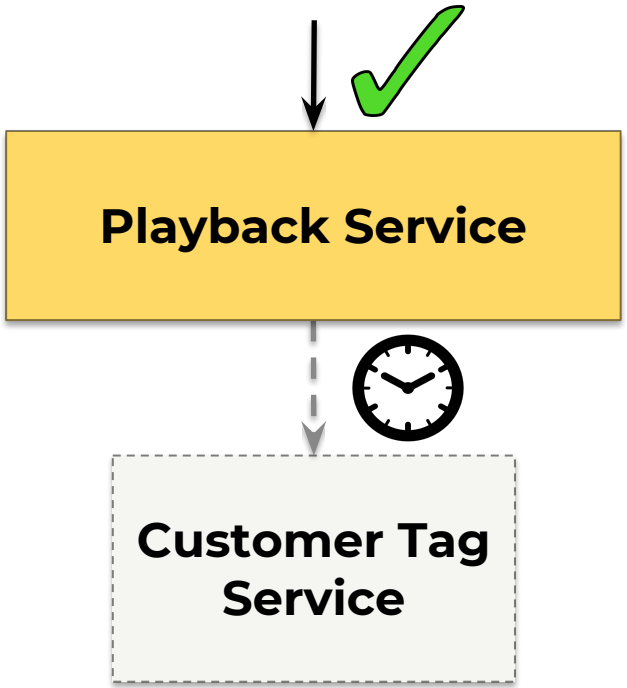
Dependency Tuning



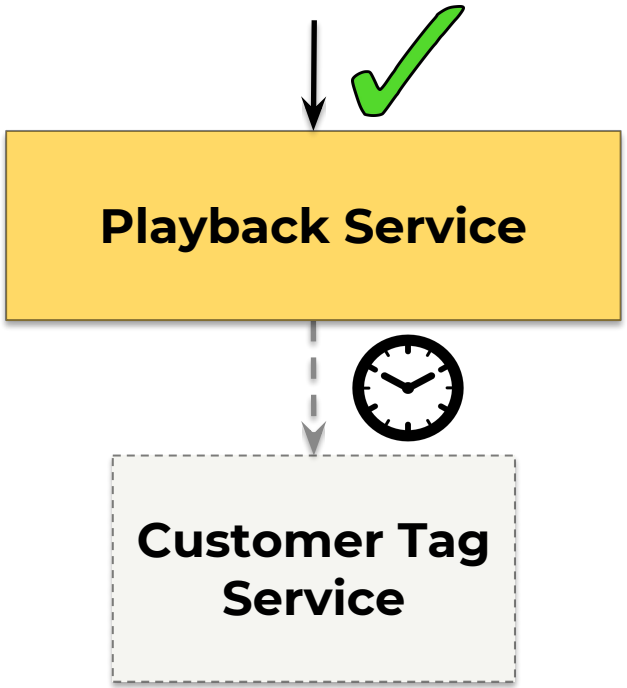
- Retries
- Timeouts
- Load balancing strategies
- Concurrency limits
- Circuit breakers

Calendar*, CC BY 2.0, Dafne Cholete 2011, Flikr

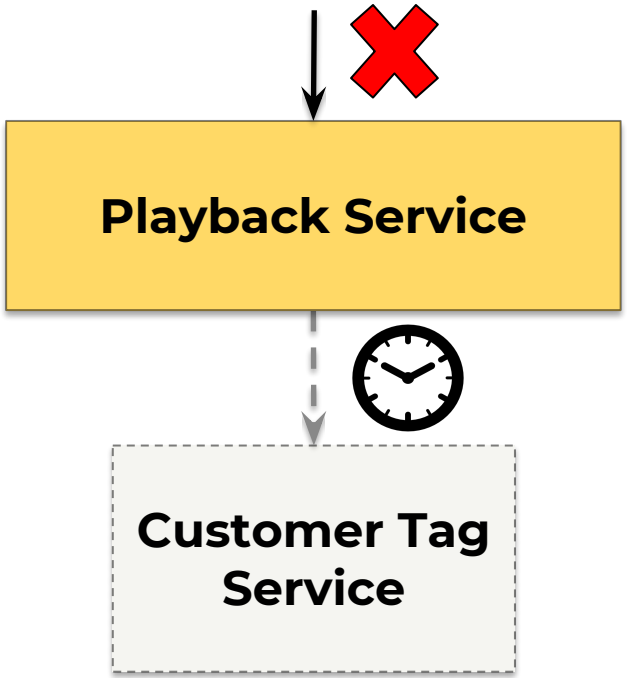
Playback Service → Customer Tag Service



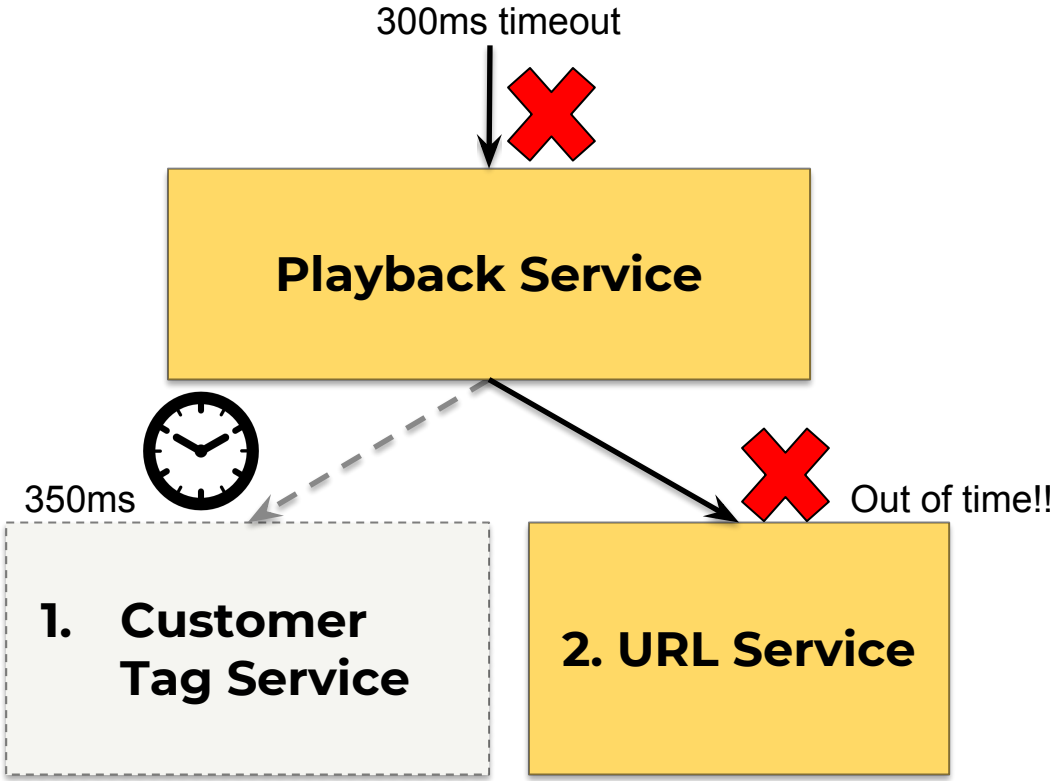
Latency Injection - Round 1



Latency Injection - Round 2

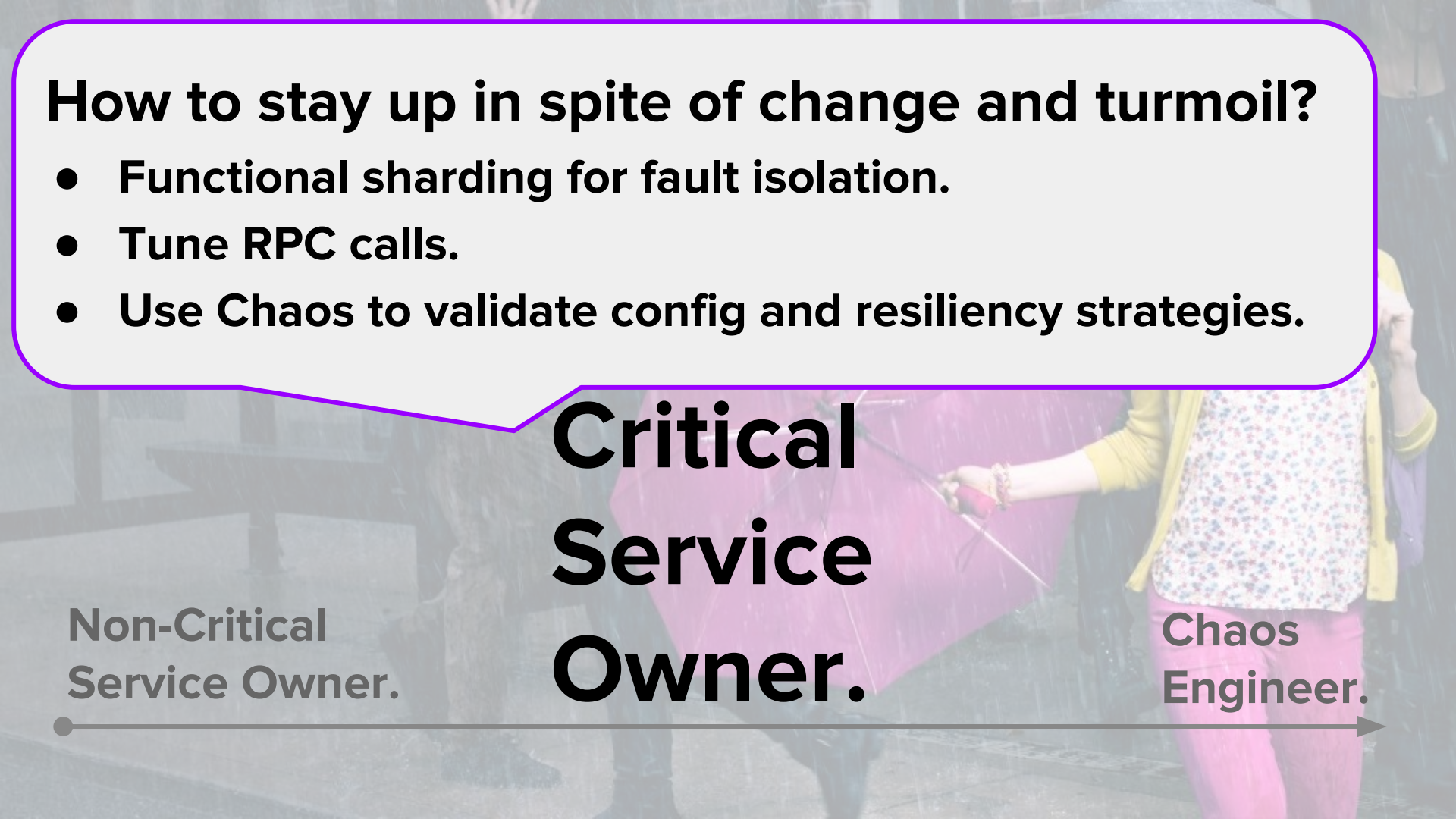


Latency Injection - Round 2



Continuous Experimentation FTW!

- Fewer changes between experiments make it easier to isolate the regression.
- Fine-grained experiments scope the investigation *(as opposed to outages where there are lots of red-herrings)*.



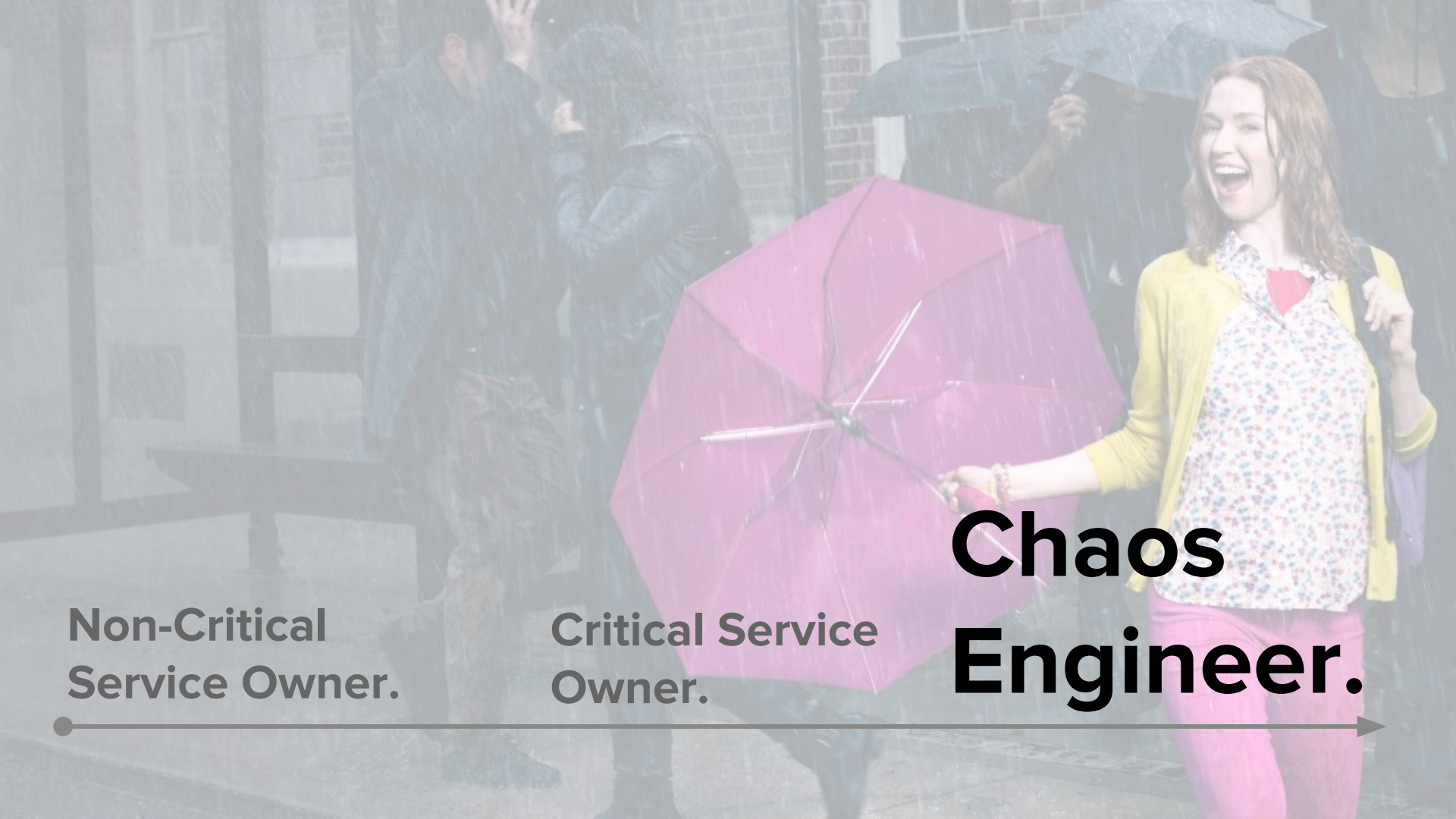
How to stay up in spite of change and turmoil?

- Functional sharding for fault isolation.
- Tune RPC calls.
- Use Chaos to validate config and resiliency strategies.

Non-Critical Service Owner.

Critical Service Owner.

Chaos Engineer.


Non-Critical
Service Owner.
Critical Service
Owner.
Chaos
Engineer.

How do you help teams build more resilient systems?

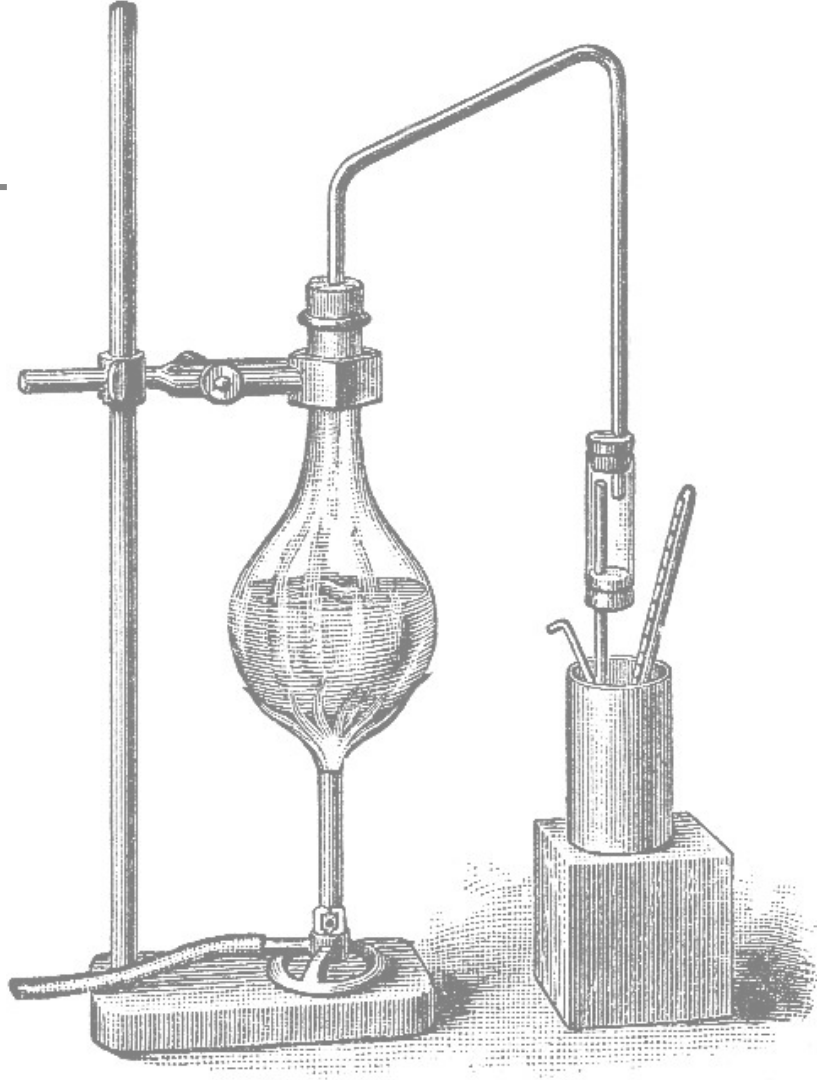
We need to do more of the heavy lifting.

Perhaps the Principles of Chaos can help!

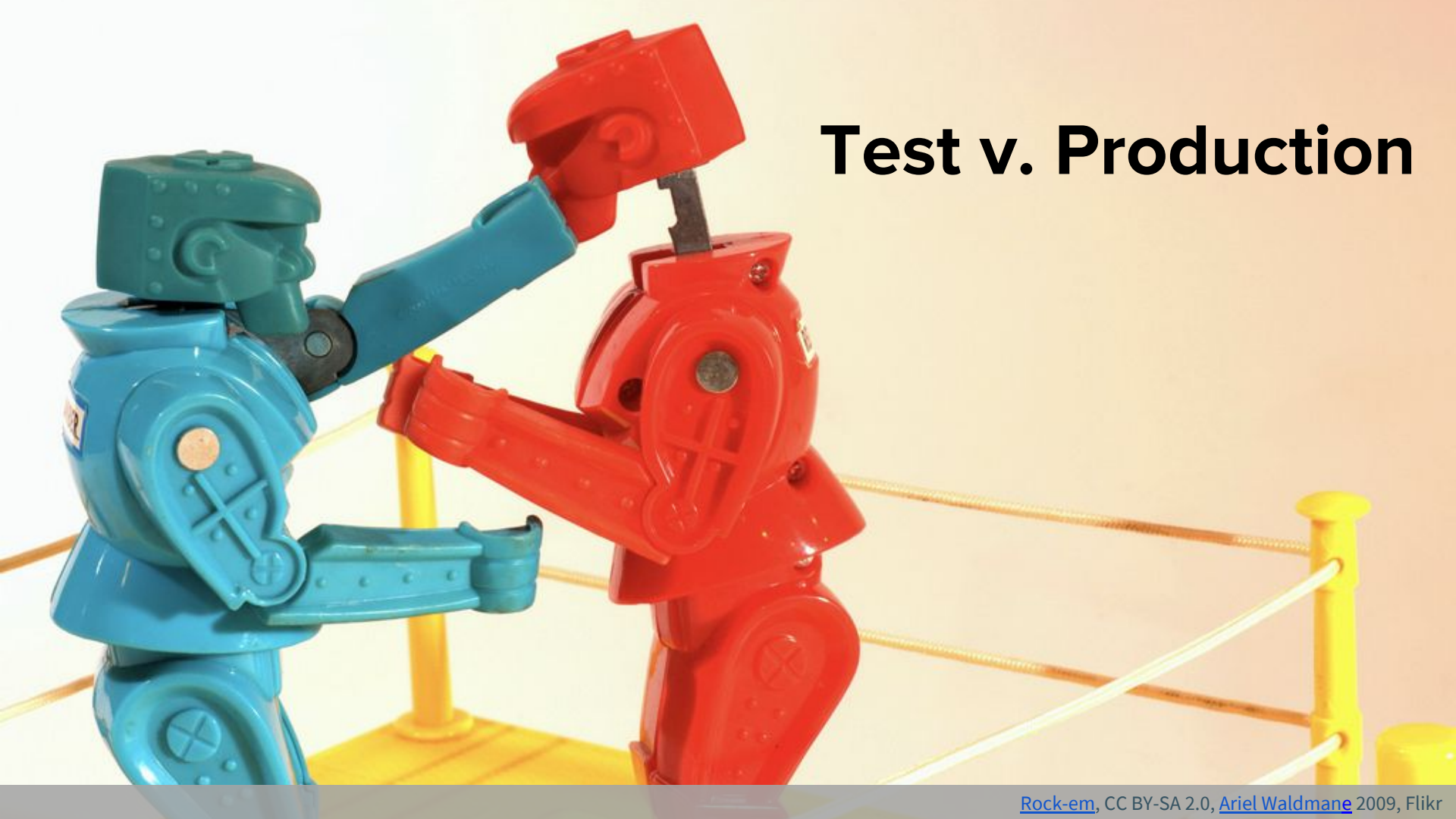
Principles of Chaos

- Minimize Blast Radius
- Build a Hypothesis around Steady State Behavior
- Vary Real-world Events
- Run Experiments in Production
- Automate Experiments to Run Continuously

https://principlesofchaos.org/



Test v. Production

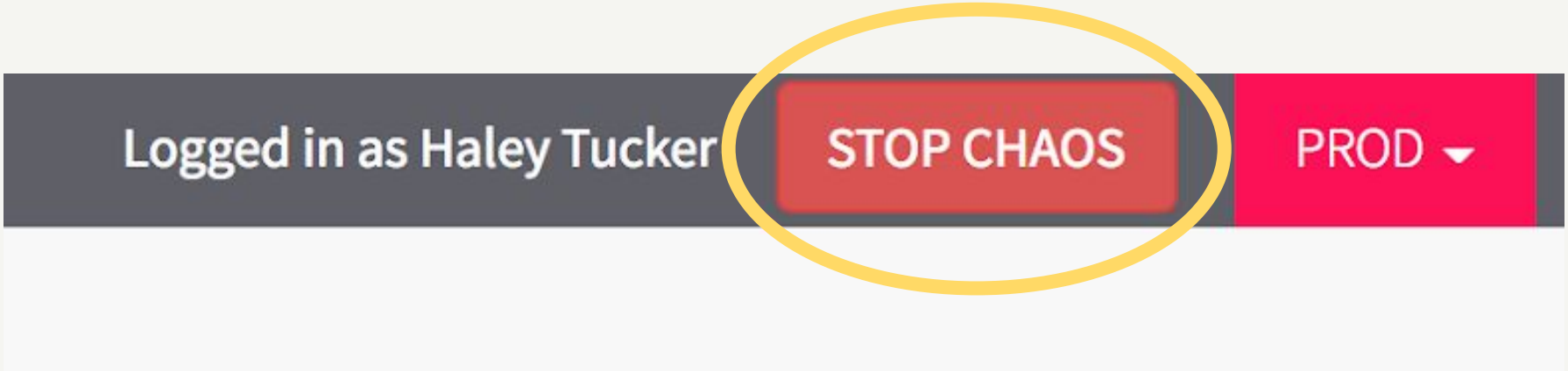


Rock-em, CC BY-SA 2.0, Ariel Waldmane 2009, Flikr

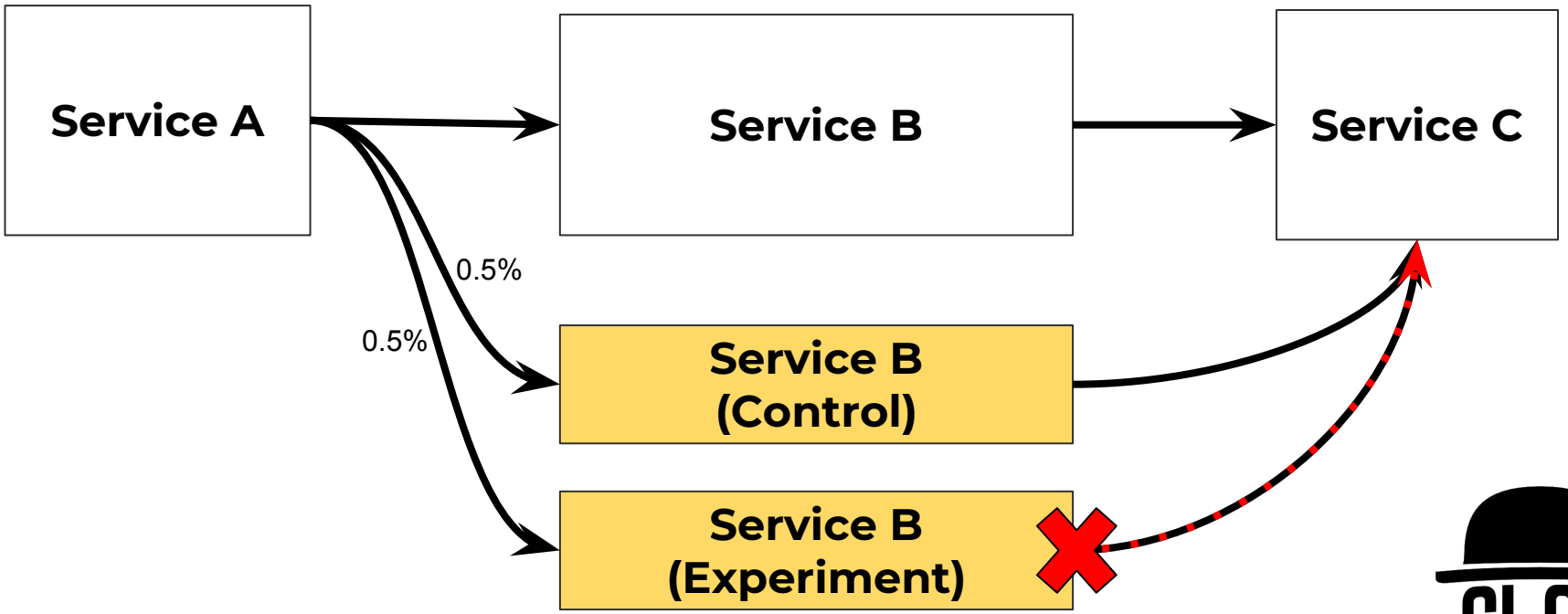
How can we Minimize Blast Radius?

Safety, safety, safety!!

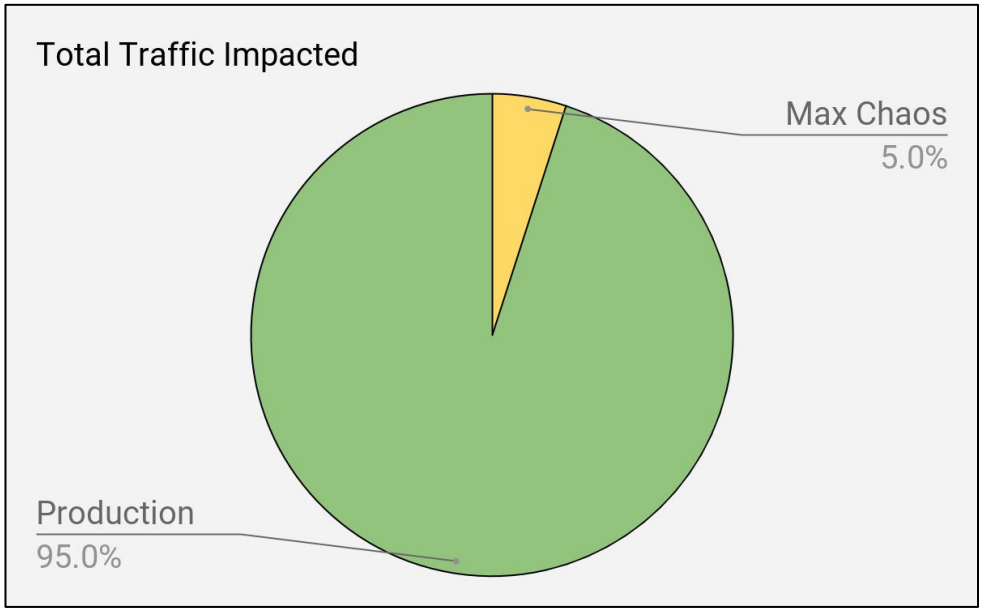
Kill Switch



Canary Strategy



Limit *Impact*

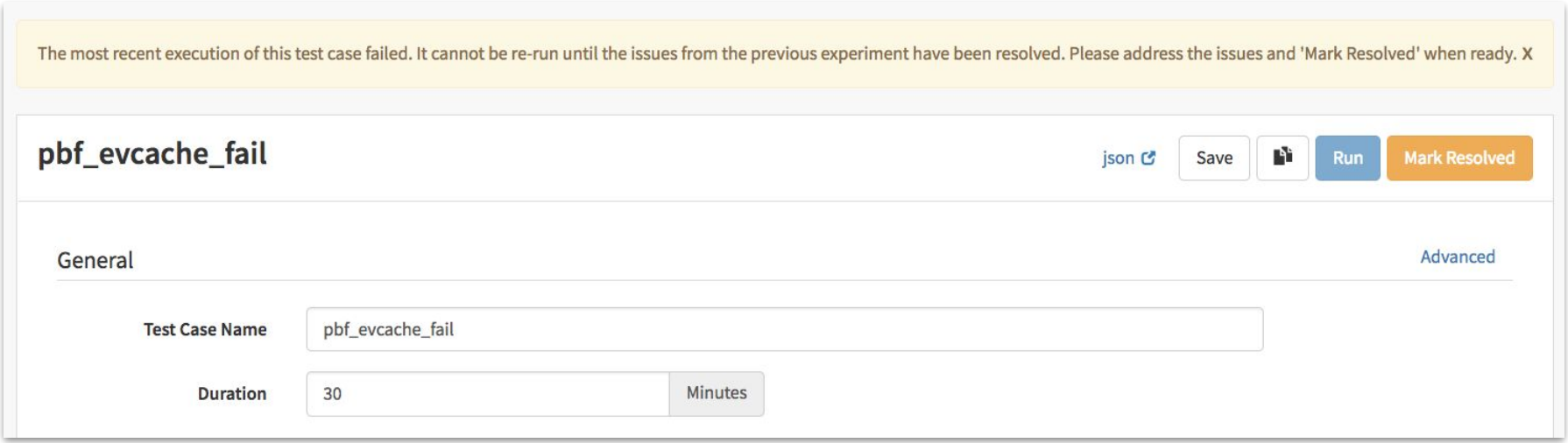


Runs In Progress		
Experiment	**Cluster**	**Status**
Latency	*api-prod*	*In Progress*
Latency	dredd-prod	In Progress
Failure	*api-prod*	*Queued*

Limit *When* Experiments can Run

Safety First during the Holidays

Ensure Failures are Addressed



Fail Open

1. Control errors too high.

2. Errors in chaos code unrelated to the experiment in question.

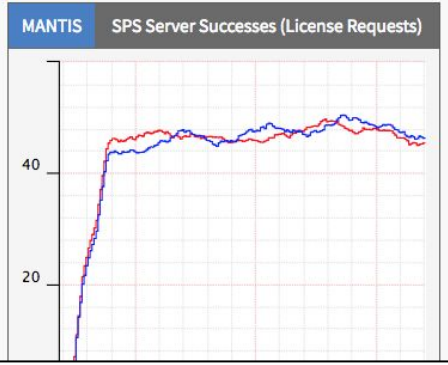
3. Platform components crashing (monitoring, worker nodes, etc).

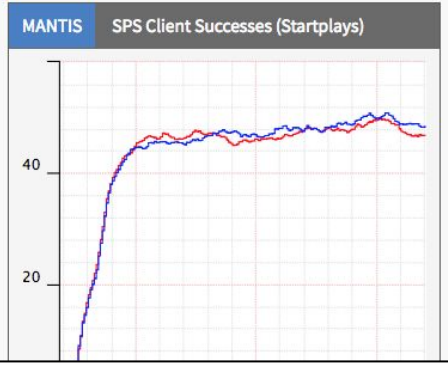
How should we Build a Hypothesis around Steady State?

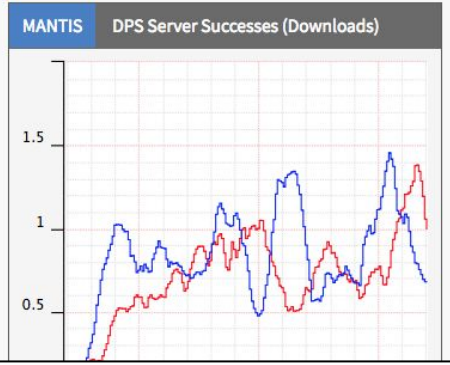
Observability is key!

Add effective monitoring, analysis, and insights.

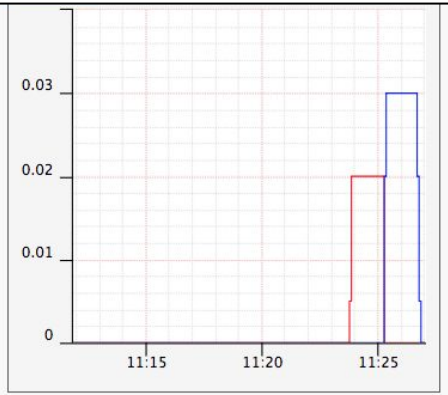
KPIs (real-time)

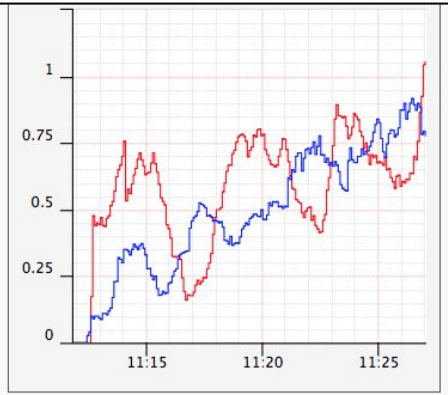


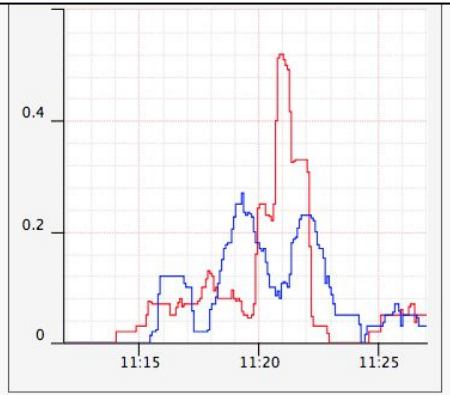




2018-06-04 10:31:55	KPI: SERVER_DPS, Kind: successes, Streams: control: 489, experiment: 729. Time interval: control: 585 seconds, experiment: 584 seconds. Difference exceeded absolute threshold (240 streams) and relative threshold (33%).
2018-06-04 10:31:55	SPS impact detected, stopping run early

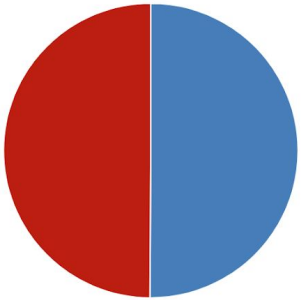




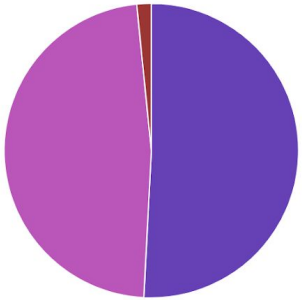


Insights

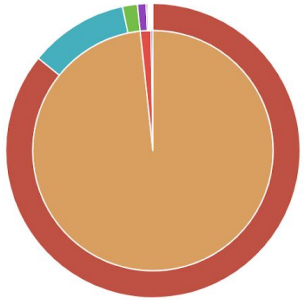
Chap Cluster Breakdown



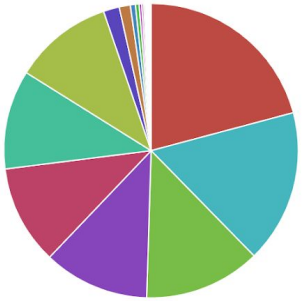
Chap KPI Breakdown



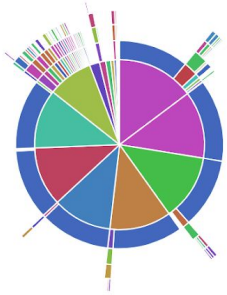
Chap Success/Error Breakdown



Chap Country Breakdown



Chap Device Breakdown

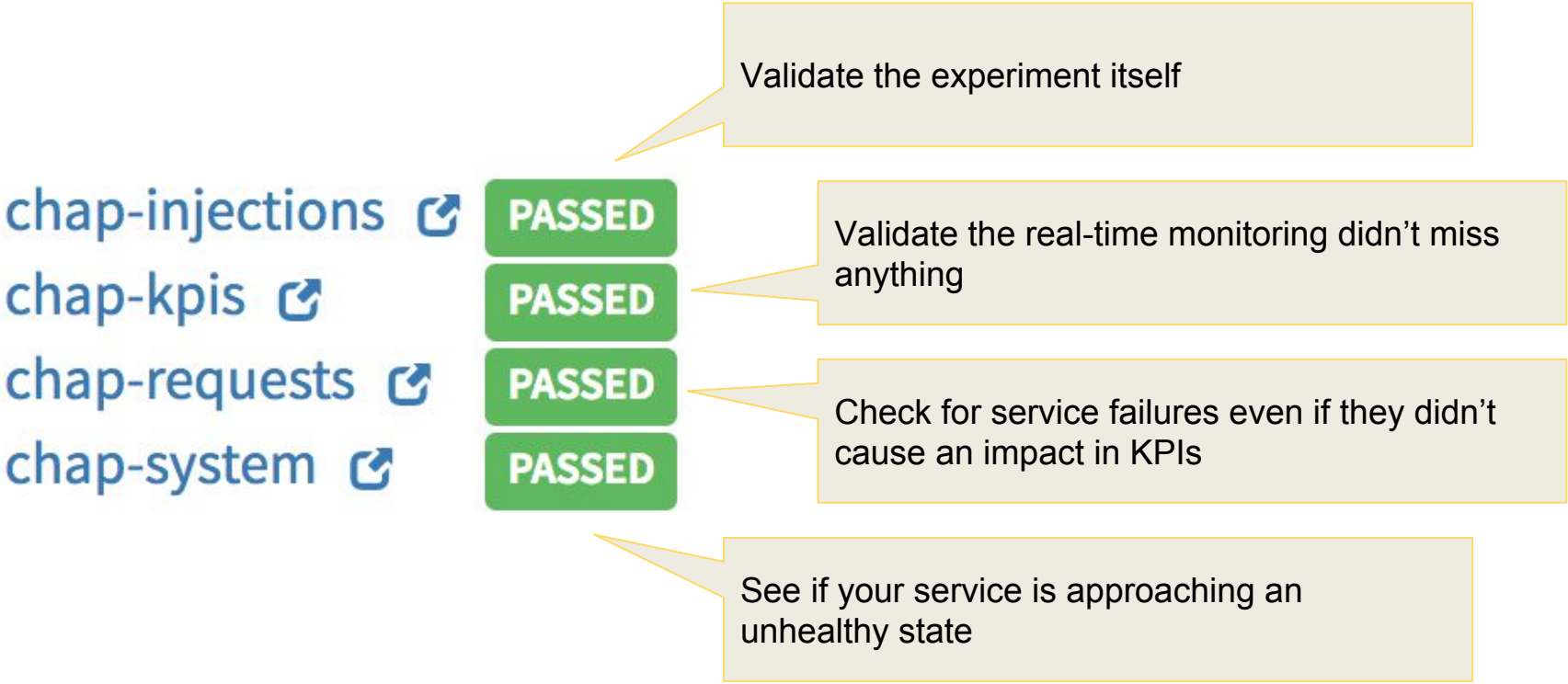


elastic

Automated Canary Analysis (ACA)

https://medium.com/netflix-techblog/automated-canary-analysis-at-netflix-with-kayenta-3260bc7acc69

ChAP ACA Configurations



How do you Vary Real-world Events in an automated fashion?

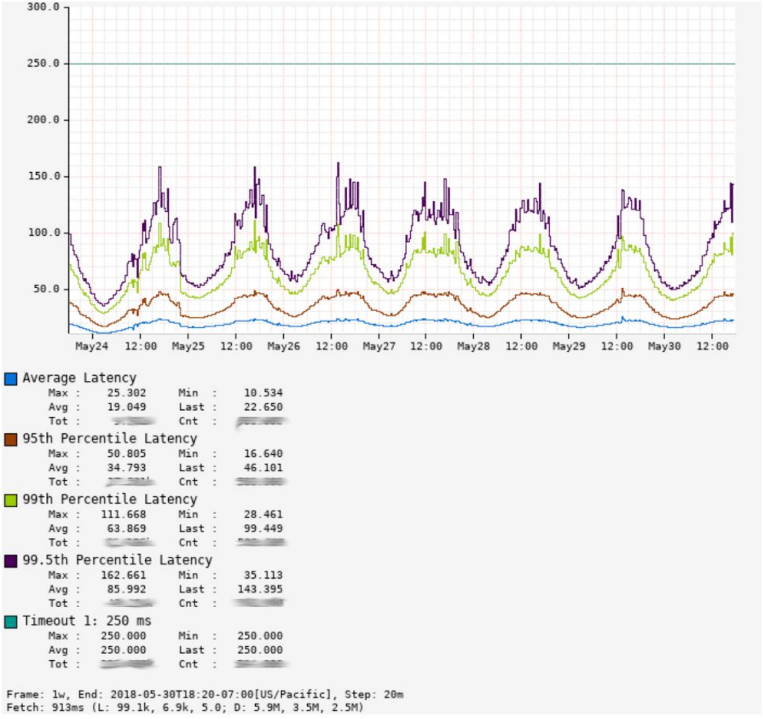
By carefully designing and prioritizing your experiments, of course!

Understand the Service Under Test

Dependency Insights:

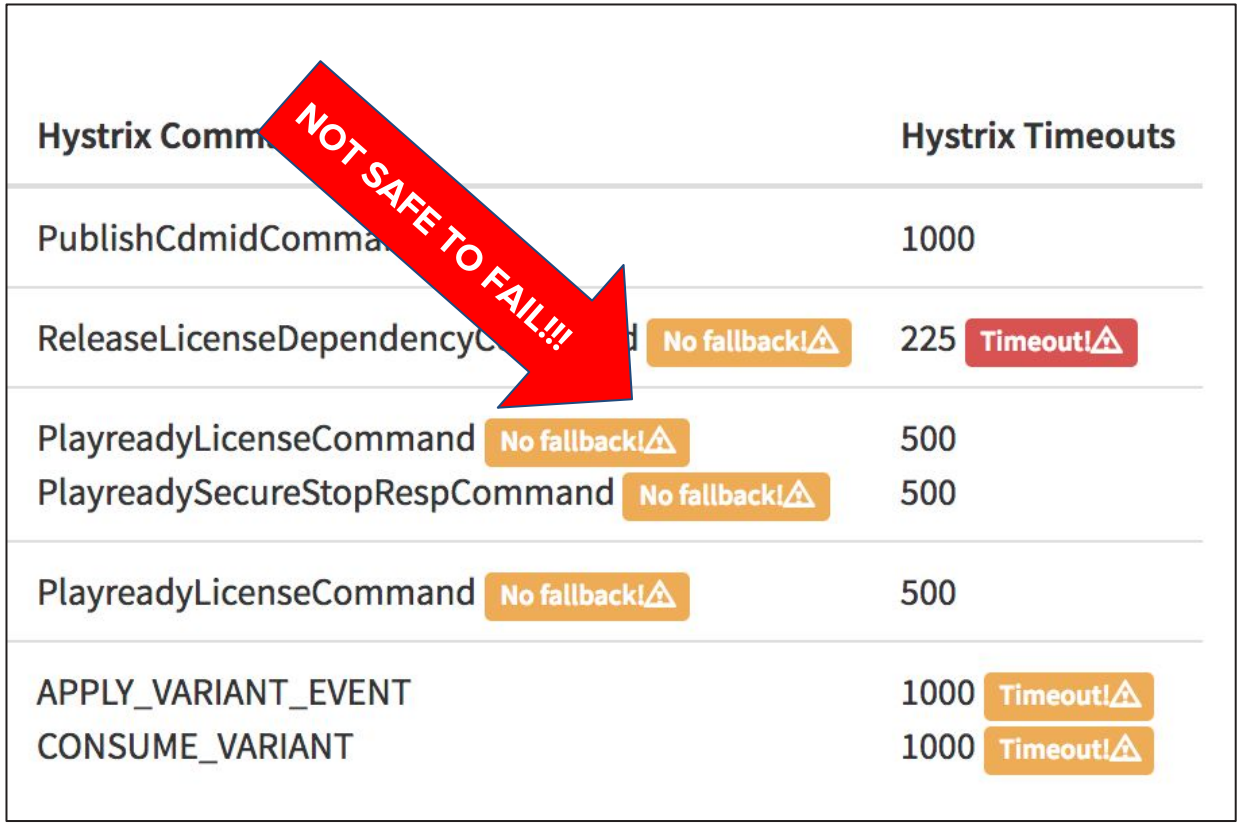
- Timeouts
- Retries
- % of Requests Involved
- Requests Per Second
- Latency
- Hystrix Commands
  - Fallbacks
  - Timeouts

	Service Name	NIWS Client Name	NIWS App Name	Read Timeout Sequence	Max Auto Retries	Max Auto Retries Next Server	Max RPS	Average RPS	Hystrix Commands	Hystrix Timeouts
+	dhs	dhs-client	dhs	300	0	1	[redacted]	[redacted]	PublishCdmidCommand	1000
+	laseoffline	laseoffline-client	laseoffline	900	0	0	[redacted]	[redacted]	ReleaseLicenseDependencyCommand No fallback!	225 Timeout!
+	playready	iis-client	playready	250	0	1	[redacted]	[redacted]	PlayreadyLicenseCommand No fallback! PlayreadySecureStopRespCommand No fallback!	500 500



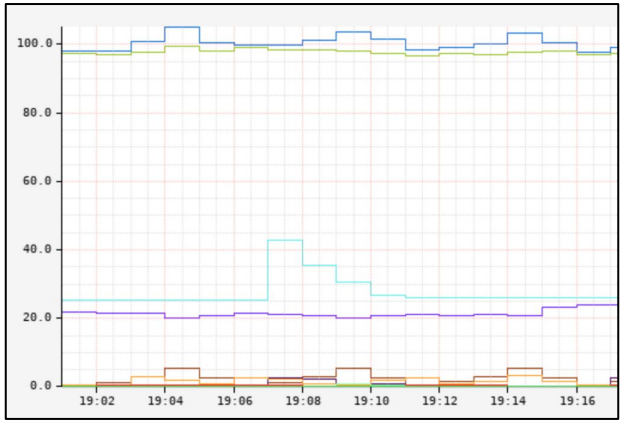
+	simone	simoneclient	simone	2000	0	1	[redacted]	[redacted]	APPLY_VARIANT_EVENT CONSUME_VARIANT	1000 Timeout! 1000 Timeout!

Evaluate Safety



Can more automation eventually lead to fewer experiments?

Prioritize Experiments



Traffic Percentage

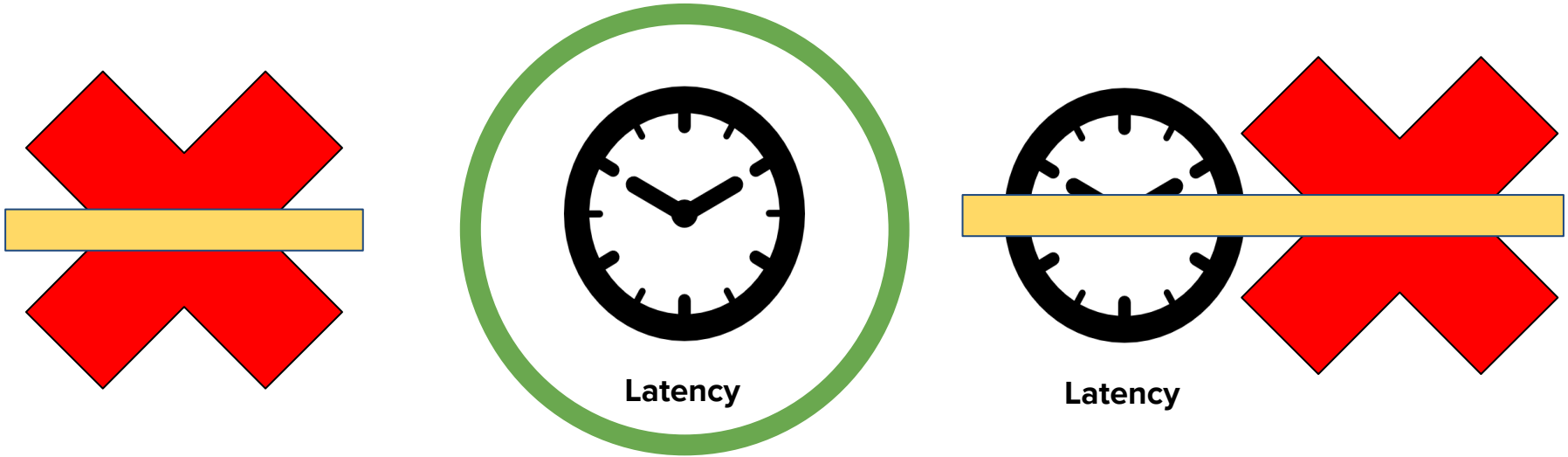
Retries

Latency

Experiment Type

Aging

Generate Experiments



Is it time to Run Experiments in Production?

Here we go!


ATM
NEW YORK

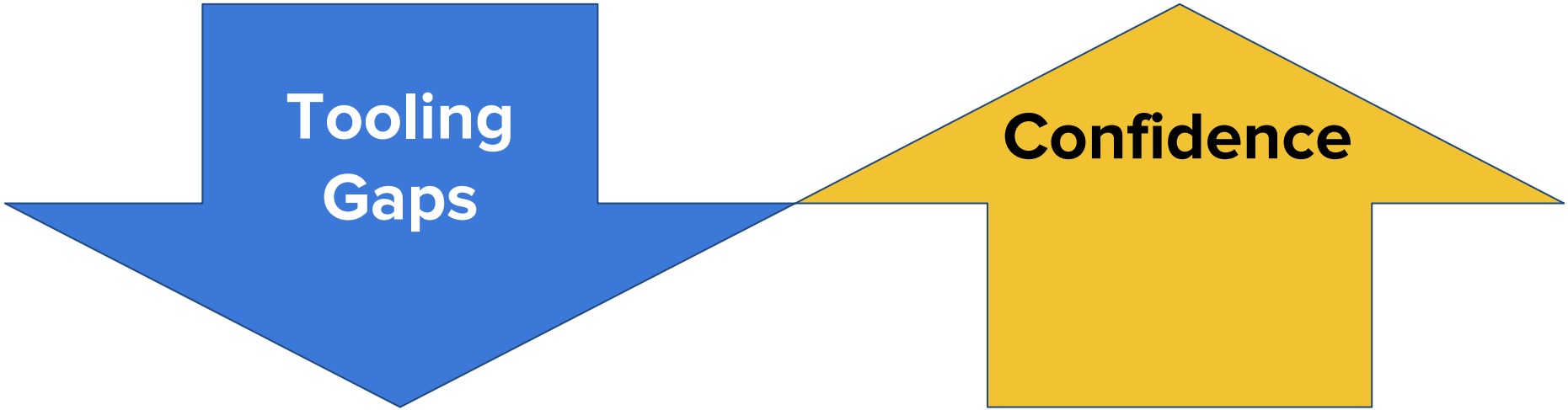
What happened?

14

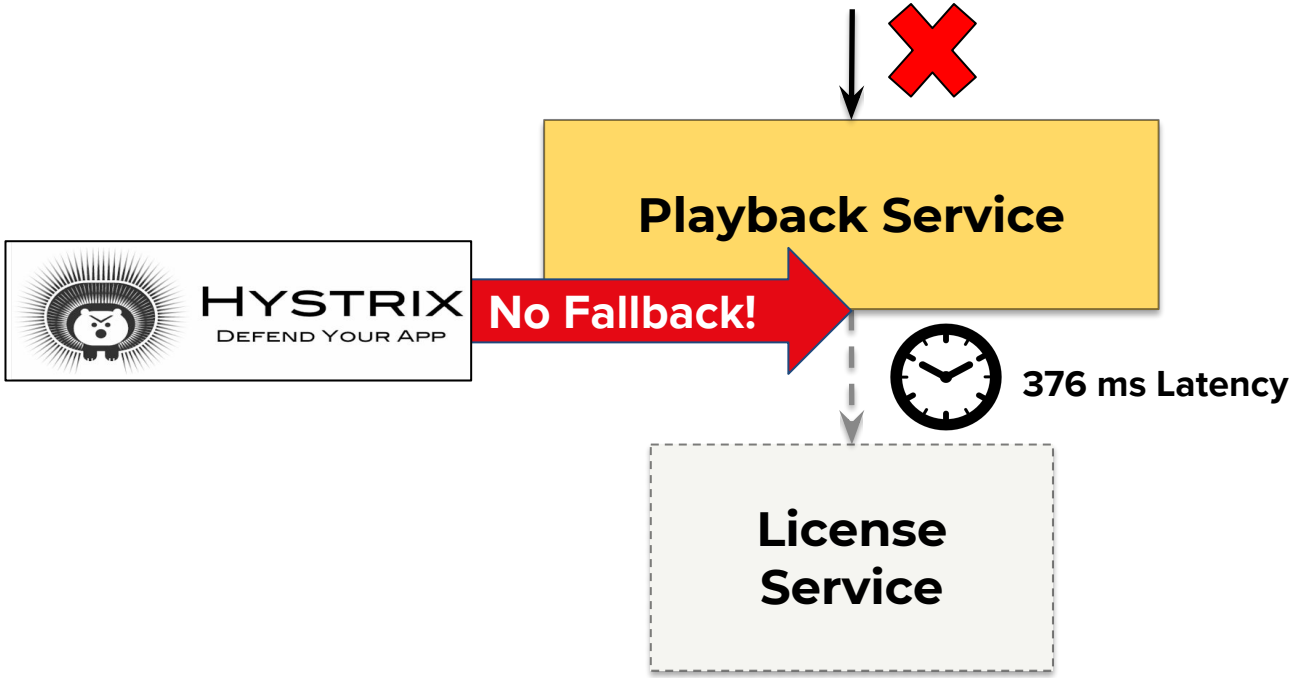
Vulnerabilities

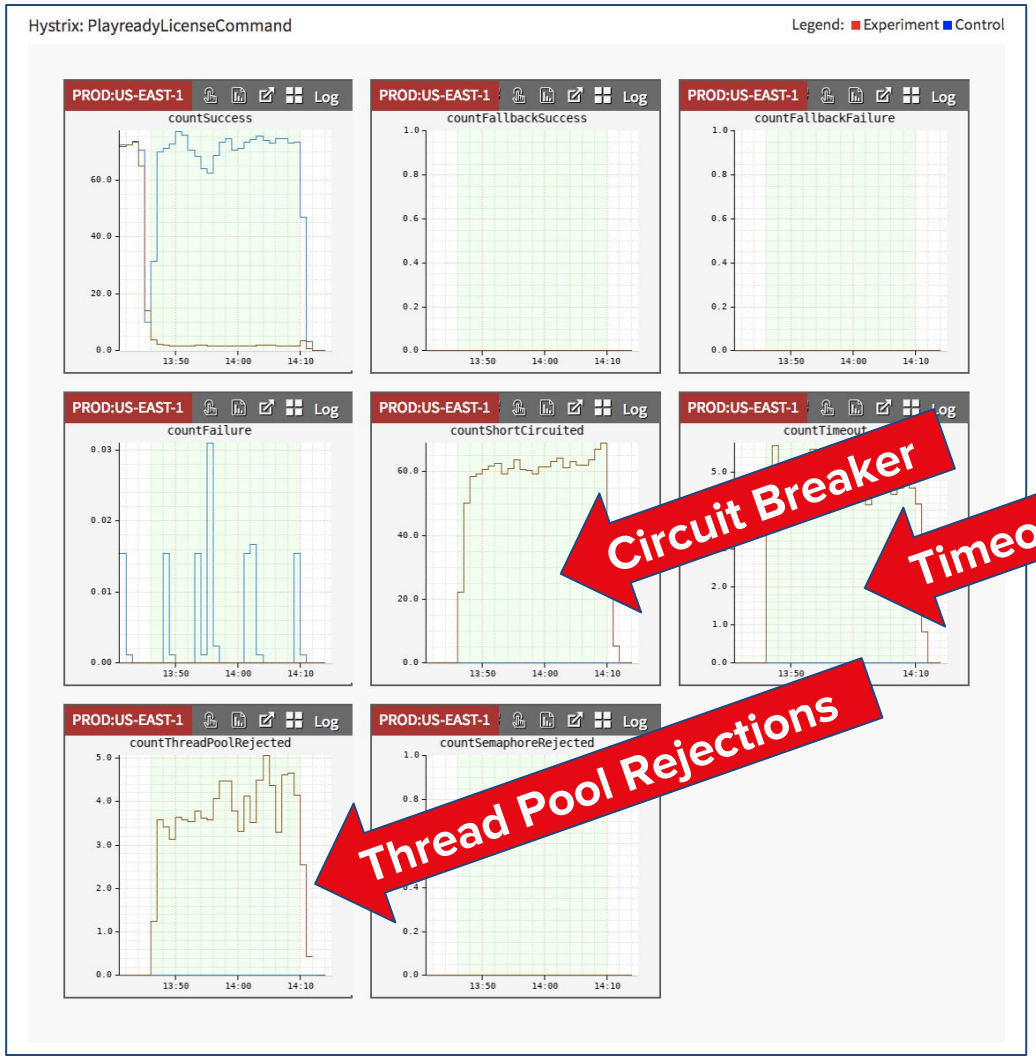
0

Outages



Example Finding





88.85% of cluster traffic

10 threads

HYSTRIX
DEFEND YOUR APP

Fully validated fix in tool before rollout!

Service To Clone

Region: eu-west-1

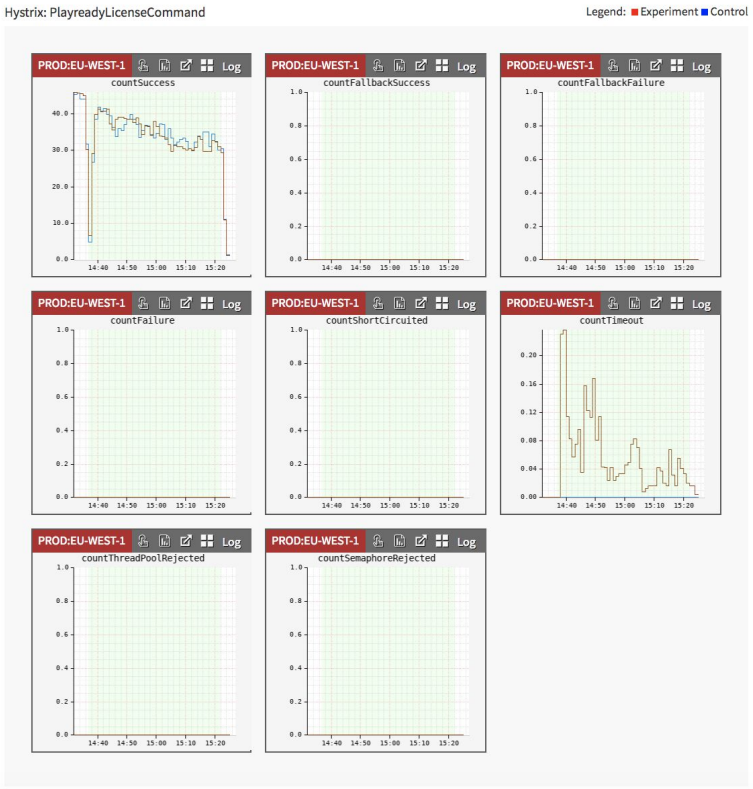
App: dredd

Cluster: dredd-noncritical

Target: Newest ASG | Previous ASG | Oldest ASG

Properties

Name	Value
iis-client.niws.client.ReadTimeout	150,250
nf.dependency.circuit.PlayreadyLicenseCommand.exe…	400
nf.dependency.threadPool.PLAYREADY_LICENSE.core…	20

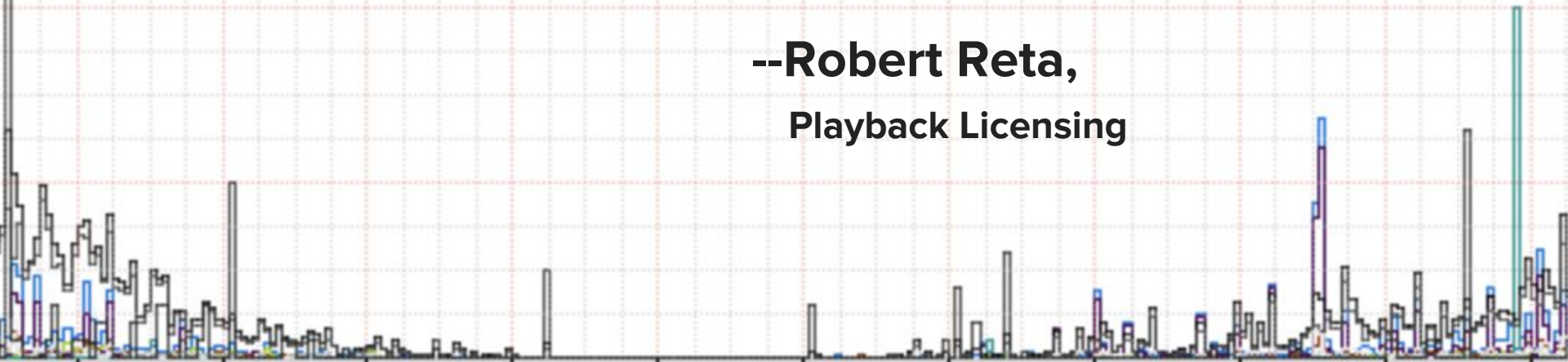


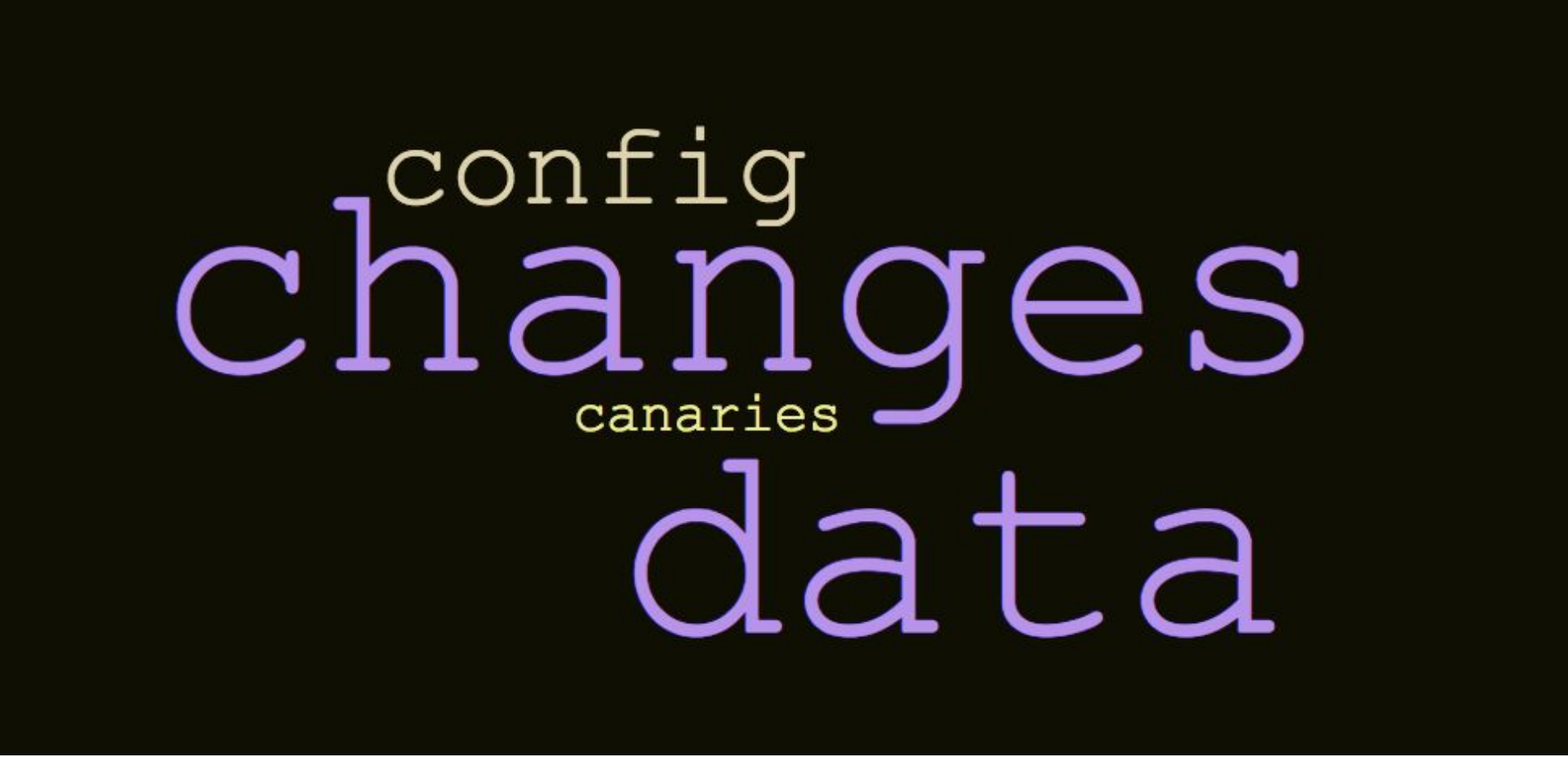
After a day's worth of data, the results are looking fantastic.

Every negative metric [for that Hystrix command] had a drastic improvement, and some by an order of magnitude.

--Robert Reta,

Playback Licensing





What else can be safer?

How do you help teams build more resilient systems?

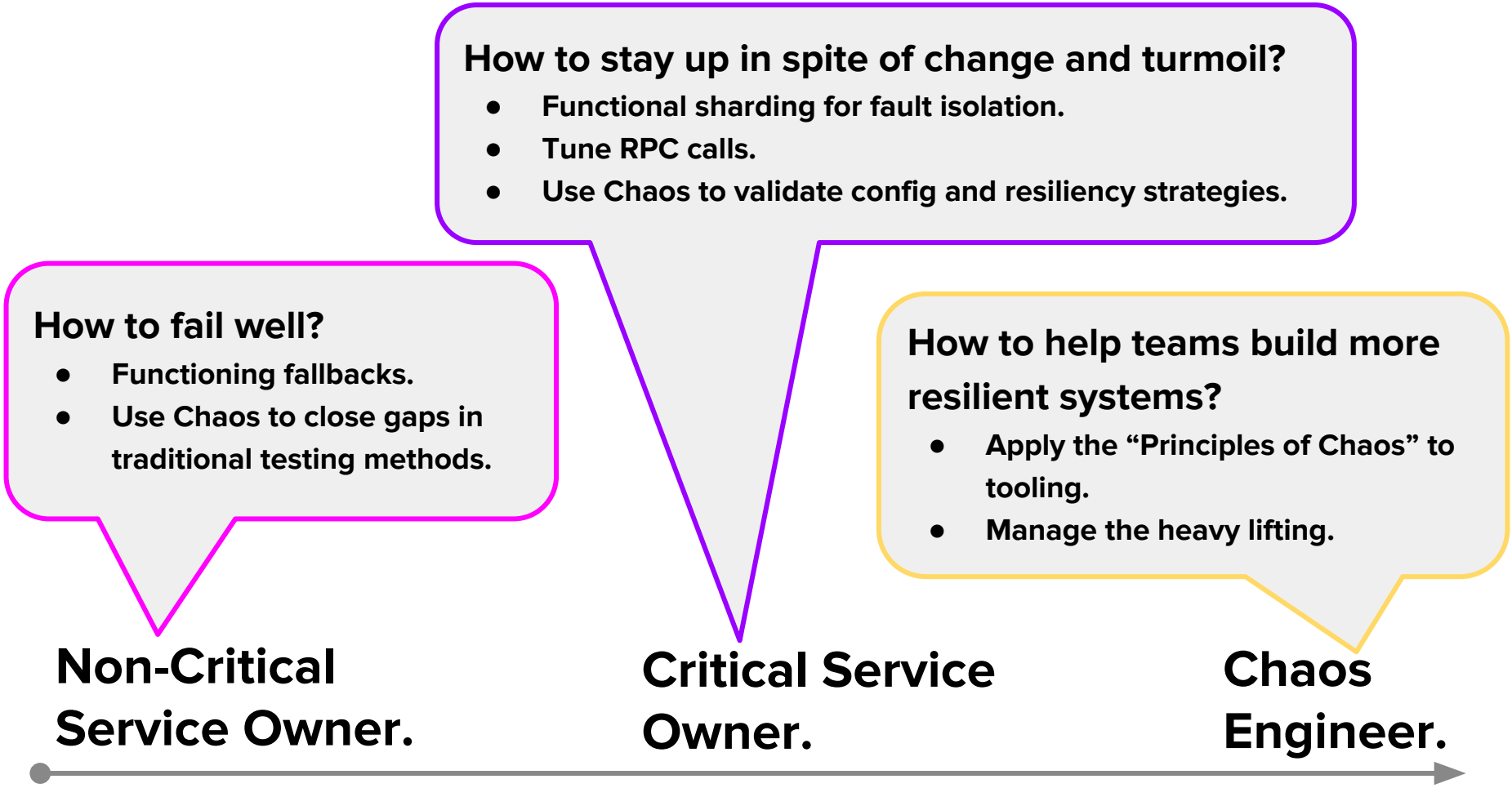
- **Apply the “Principles of Chaos” to tooling.**
- **Manage the heavy lifting.**

Non-Critical Service Owner.

Critical Service Owner.

Chaos Engineer.

You Must be This Tall to Ride?


How to stay up in spite of change and turmoil?
• Functional sharding for fault isolation.
• Tune RPC calls.
• Use Chaos to validate config and resiliency strategies.
How to fail well?
• Functioning fallbacks.
• Use Chaos to close gaps in traditional testing methods.
How to help teams build more resilient systems?
• Apply the “Principles of Chaos” to tooling.
• Manage the heavy lifting.
Non-Critical Service Owner.
Critical Service Owner.
Chaos Engineer.

You Can Either Curl Up In A Ball And Die...

Or You Can Stand Up And Say, “We’re Different. We’re The Strong Ones, And You Can’t Break Us!”

Haley Tucker

Senior Software Engineer

Chaos Engineering

@hwilson1204