## MHAT I MISH I KNEWBEFORE SCALING UBER TO 1,000 SERVICES **MATT RANNEY**





## MHAT I MISH I KNEWBEFORE SCALING UBER TO 1,000 SERVICES **MATT RANNEY**





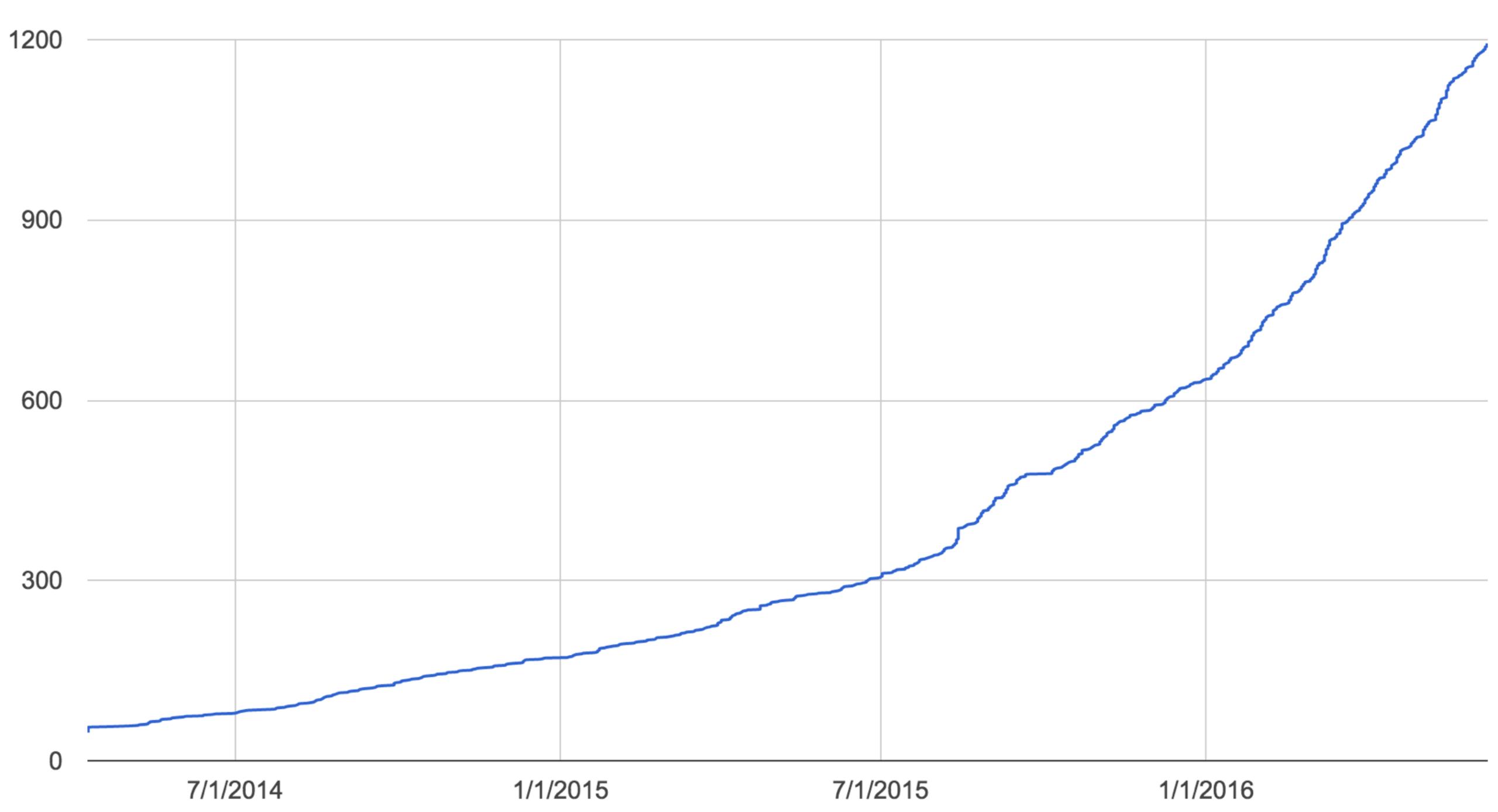
#### **As of June 2016:**

- **Countries: 70**
- Employees: 7,000-

# Uber Cities Worldwide: 400+

### LIFE LESSONS

#### total services



## WHY MICROSERVICES?

### **Own your Uptime** Use the "Best" tool for the job

- Move and Release Independently



## WHAT ARE THE COSTS?

What if it breaks? **Operational complexity** 

- Now you have a distributed system



### MICROSERVICES

Immutable? Append Only?

## LESS OBVIOUS COSTS

### **Everything is a tradeoff** You can build around problems Might trade complexity for politics You get to keep your biases



## LANGUAGES

# Hard to share code

- Hard to move between teams
- WIWIK: Fragments the culture

### HTTP/REST gets complicated JSON needs a schema **RPCs are slower than PCs**

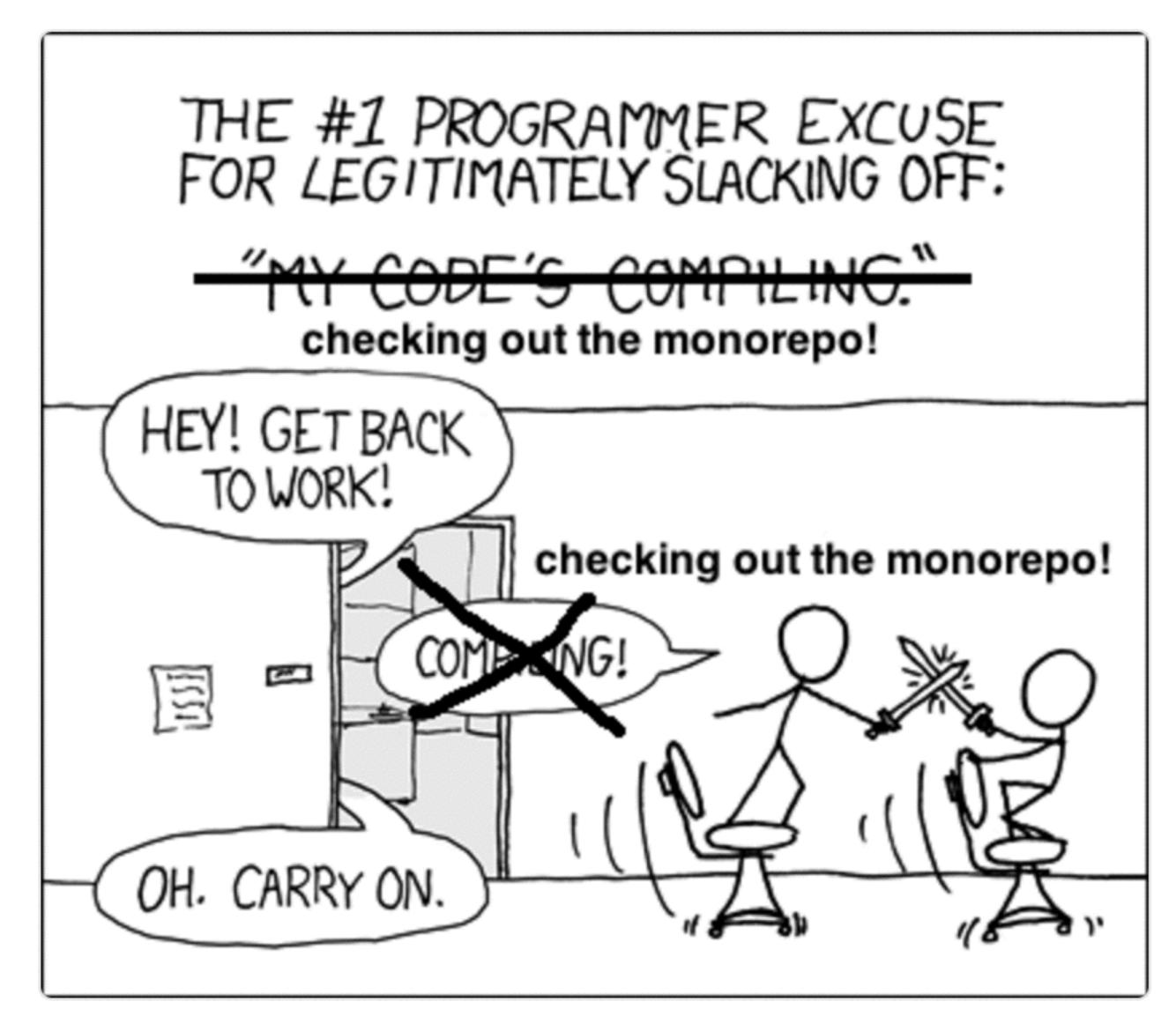
## RP(C

- WIWIK: servers are not browsers

### HOW MANY REPOS



#### here I updated the XKCD comic for you



#### **APRIL 2016**

7005 total 1074 personal 374 conf

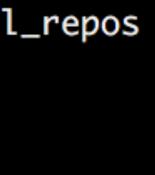
#### MAY 2016

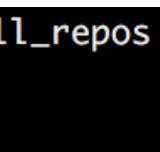
mjr:~\$ perl -ne '\$c++; \$p++ if /personal/; \$conf++ if /config/; END { print "\$c total\n\$p personal\n\$conf conf\n";}' all\_repos 8263 total 1137 personal 407 conf

#### **JUNE 2016**

mjr:\$ perl -ne '\$c++; \$p++ if /personal/; \$conf++ if /config/; END { print "\$c total\n\$p personal\n\$conf conf\n";}' all\_repos 8760 total 1170 personal 427 conf

#### mjr:~\$ perl -ne '\$c++; \$p++ if /personal/; \$conf++ if /config/; END { print "\$c total\n\$p personal\n\$conf conf\n";}' all\_repos



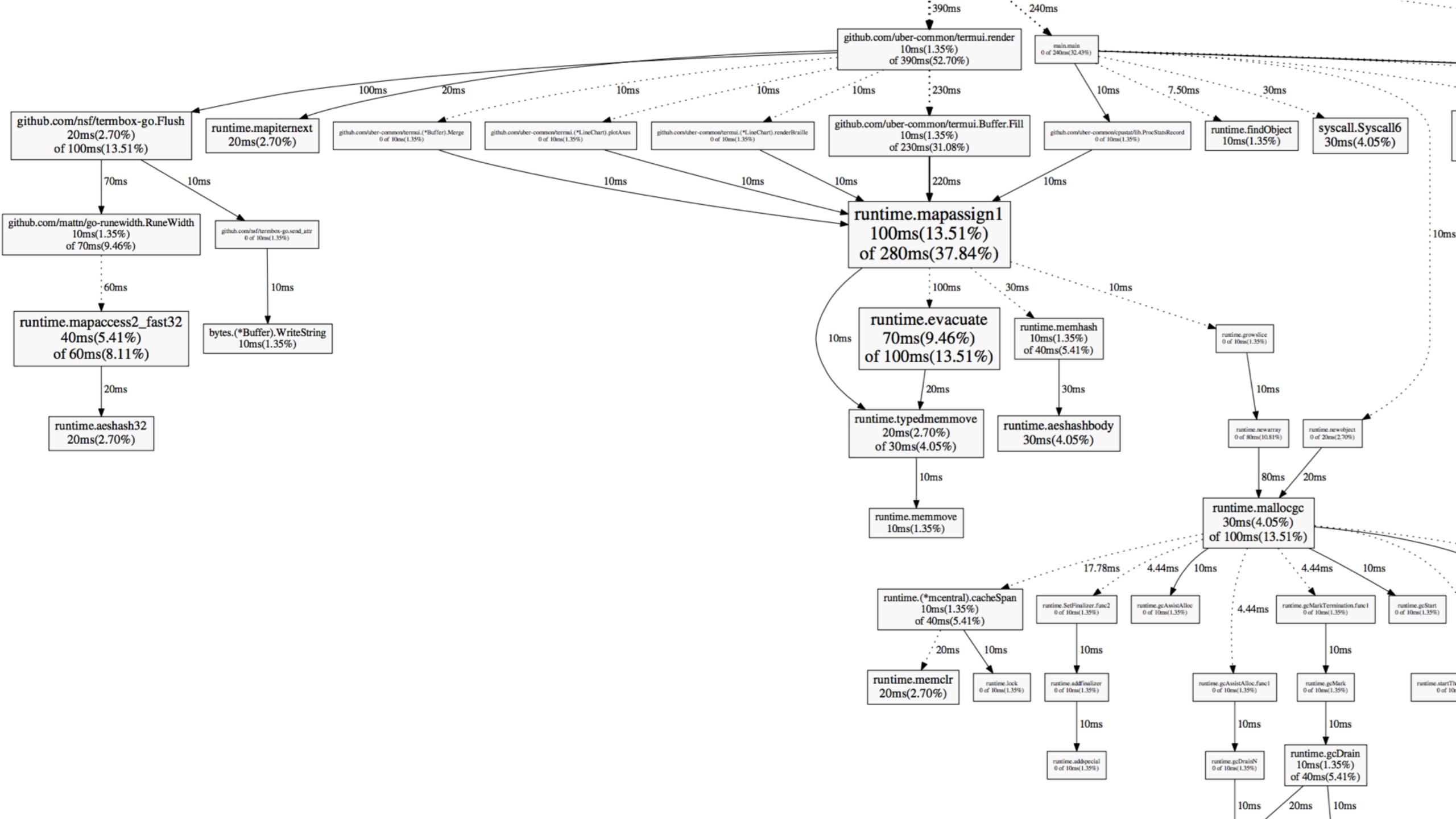


## OPERATIONAL

### What happens when things break? Can other teams release your service? Understand a service in the larger context

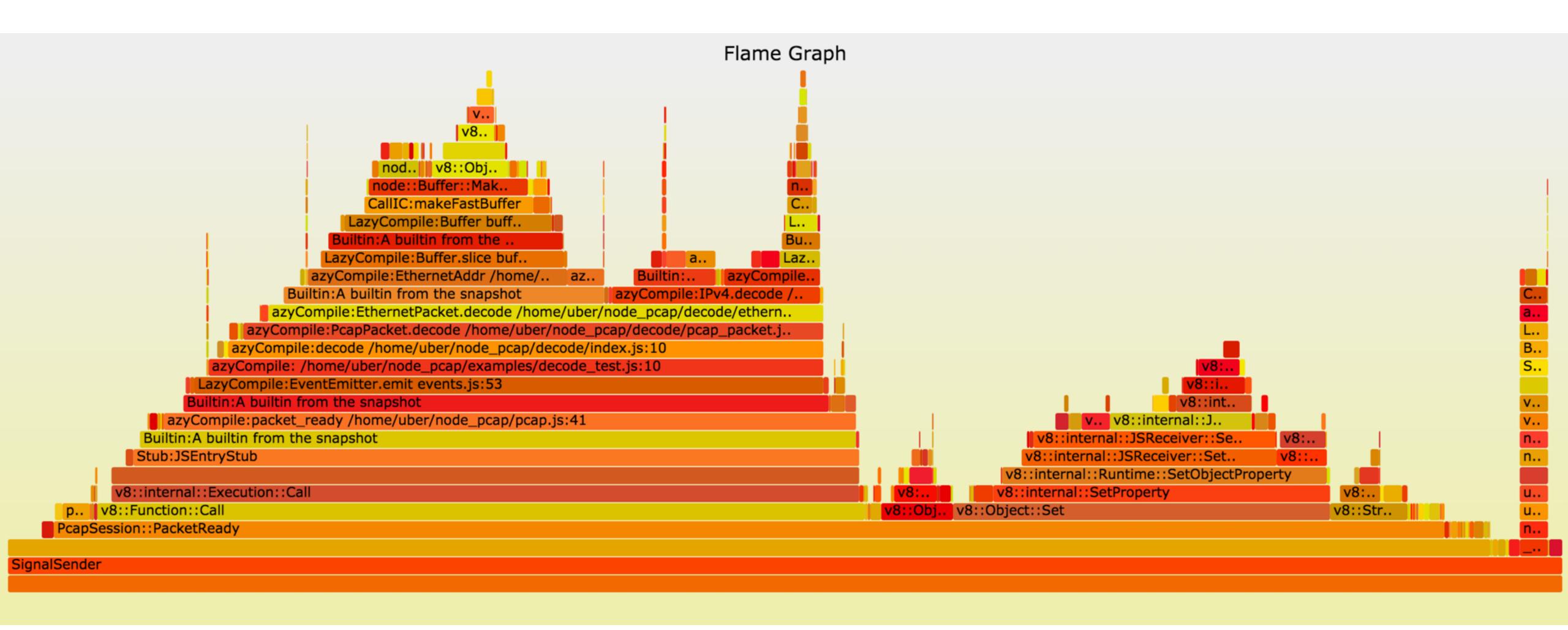
### PERFORMANCE

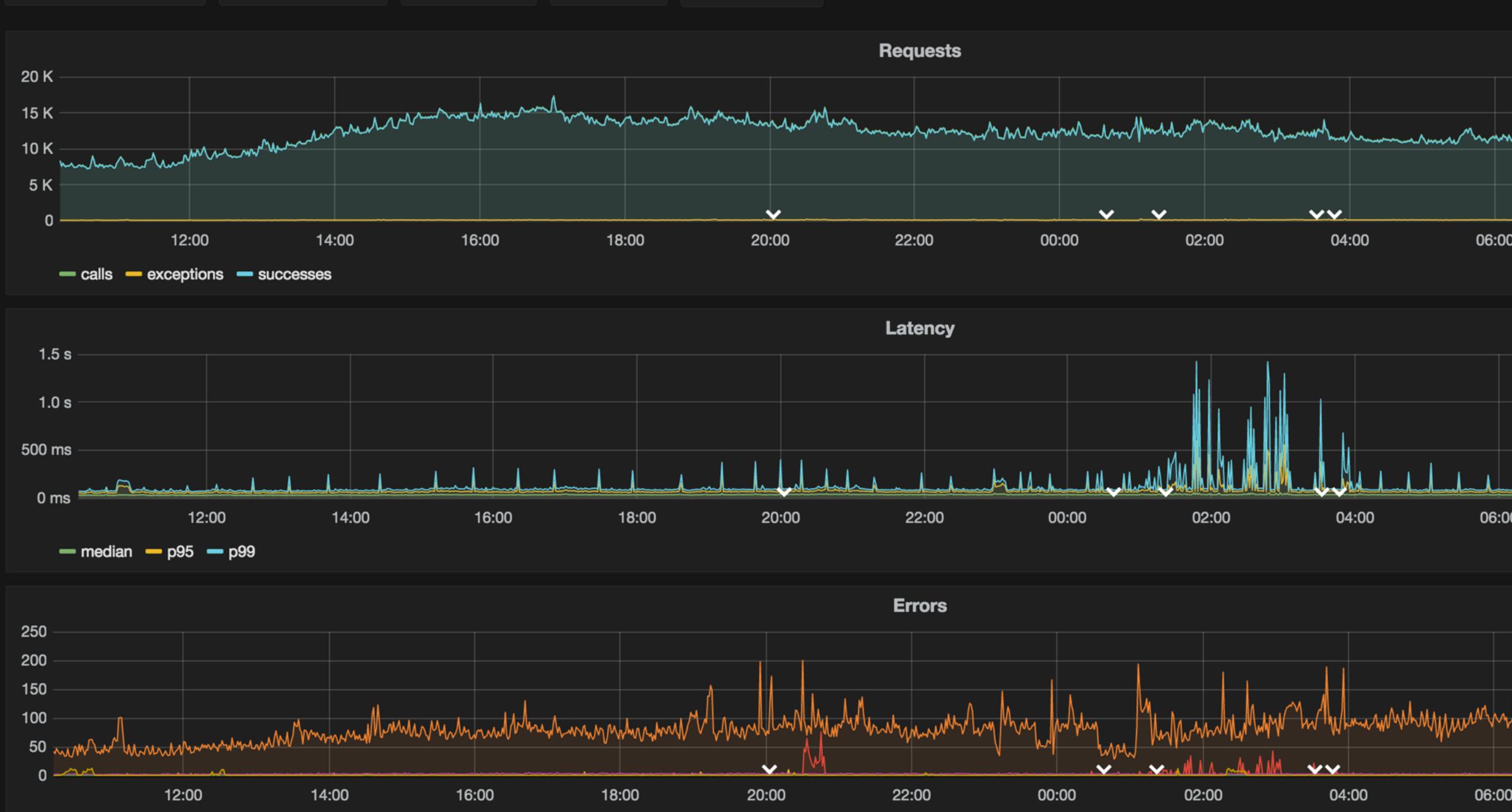
#### **Depends on language tools**



									en	
						syscall.Syscall6			encod.	syscall.Syscall6
			runtime.cgocall_errno	en.		syscall.mount		syscall.Syscall6	encodi	syscall.mount
			github.com/docker/d	enc		syscall.Mount		syscall.mount	encodin.	syscall.Mount
			g github.com/docker/d	enco		github.com/dock		syscall.Mount	encodin.	github.com/docke
			g github.com/docker/d	enco		github.com/dock		github.com/docker	e encoding.	github.com/docke
			g github.com/docker/d	enco.		github.com/dock.		github.com/docker	encoding/js.	github.com/docke
			github.com/docker/dock	encodi.	cr	github.com/dock		github.com/docker	github.com/dock	github.com/docke
			github.com/docker/docke	github.co	cr	github.com/dock		github.com/docker/	github.com/dock	github.com/docker/
			github.com/docker/docke	github.co	io	github.com/dock		github.com/docker/	github.com/dock	github.com/docker/
			github.com/docker/docke	github.co	gi	github.com/dock		github.com/docker/	github.com/dock	github.com/docker/
			github.com/docker/docke	github.co	gi	github.com/dock	g	jithub.com/docker/d	github.com/dock	github.com/docker/
		sy	github.com/docker/docker	github.com	gi	github.com/docker/d	locke	r/daemon.(*Daemon).Cr	reate	github.com/docker/do
		sy	github.com/docker/docker	github.com	git	github.com/docker/d	locke	r/daemon.(*Daemon).Co	ontainerCreate	github.com/docker/do
		git	github.com/docker/docker	github.com	git	github.com/docker/do	ocker	r/api/server/router/local.	(*router).pos	github.com/docker/do
		git	github.com/docker/docker	github.com	git	github.com/docker/do	ocker	r/api/server/router/local.	*router.(gith	github.com/docker/do
		git	github.com/docker/docker/ap	/server.func·004						
		git	github.com/docker/docker/ap	/server.func.003	3					
		git	github.com/docker/docker/ap	/server.func.002	2					
			github.com/docker/docker/ap	/server.func.001						
		git	github.com/docker/docker/ap	/server.func.007	7					
		git	net/http.HandlerFunc.ServeH	ТР						
		githu	github.com/gorilla/mux.(*Rou	ter).ServeHTTP						
		githu	net/http.serverHandler.Servel	ITTP						
scan	r 🚺 🚺	githu	net/http.(*conn).serve							
GC	System	runtime.g	joexit							









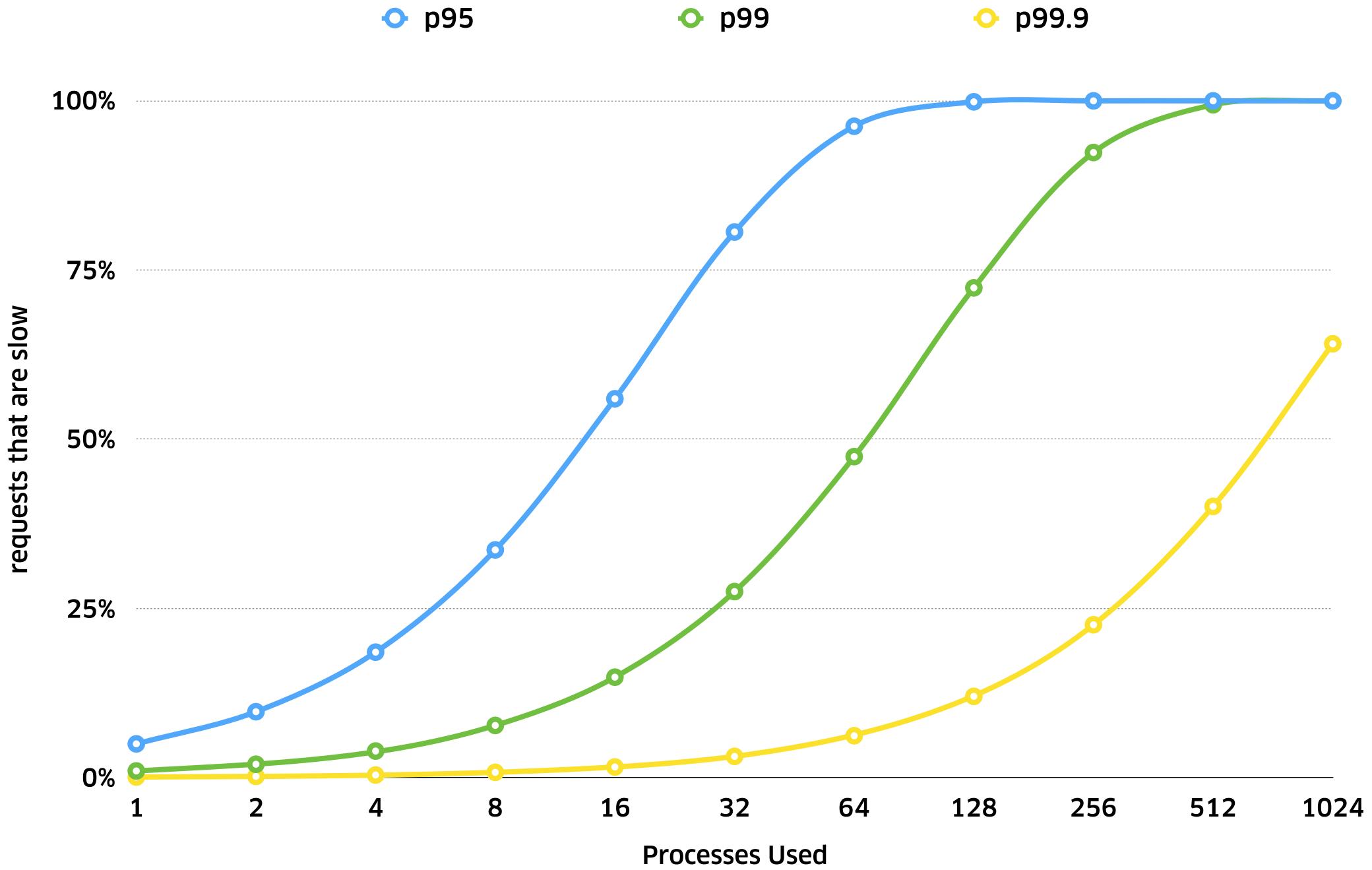
## PERFORMANCE

#### Doesn't matter until it does Probably want at least simple perf requirements WIWIK: "good" not required, but "known" is

### FANOUT

### 1ms avg, 1000ms p99 **use 1: 1% at least 1000ms** use 100: 63% at least 1000ms $1.0 - 0.99^{100} = 0.634 = 63.4\%$

- overall latency ≥ latency of slowest



• p99

◆ p99.9

## TRACING

#### Lots of ways to get this Best way to understand fanout

Se	rvices		7	756.000ms	1.512s	2.268s	3.024s	
-	rtapi	3.775s : /rid	ers/:rideruuid/pickup	•	•	•	•	
-	passport	·3.000ms : r	esolveregion					
-	cn	·3.000ms : r	esolveregion					
-	on	· 162.000ms	: getclient					
-	halyard	. 58	3.000ms : gettreatmentresult					
-	optic	. 62	2.000ms : /client/:uuid/ping					
-	geospatial	· 6.	.000ms : supply.rpc.multique	ery				
-	paxon	. 3.	.000ms : /eyeball/:					
	ueta	. 3	3.000ms : /v2/eta/predict-m	апу				
-	onedirection	. 4	4.000ms : /fitted_multi					
-	onedirection	. 3	3.000ms : /fitted_multi					
-	ueta	. 3	2.000ms : /v2/eta/predict-m	апу				
-	ultron		4.000ms : /classify					
-	ultron		3.000ms : /classify					
-	api		3.085s : verifypaymentprof	ile				
-	demand							230.000ms :
-	optic							
-	optic							
-	demand							
-	trident							
-	on							
-	passport							



Service         International         22200m           Second         55000ms : Landiates/supply insk         -           Second         37.000ms : supply justimitiquesty         -         -           Second         -         -         -         -         -           Second         -         -         -         -         -         -           Second         -         -         -         -         -         -         -           Second         -						
-         97000ms: supply.gon.ml.tiquery         -           -         Respin         -     <			111.000ms	222.000ms		:
spyy         44.000ms:supply.ukily.msd           supply         35.000ms:supply.ukily.msd           supply         35.000ms:supply.ukily.msd           supply         35.000ms:supply.ukily.msd           supply         35.000ms:supply.ukily.msd           supply         35.000ms:supply.ukily.msd           supply         40.000ms:supply.ukily.msd           supply         30.000ms:supply.ukily.msd           supply         31.000ms:supply.ukily.msd           supply         32.000ms:supply.ukily.msd           supply         32.000ms:supply.ukily.msd           supply         32.000ms:supply.ukily.m	- disco	-555.000ms : /candidates/supply/rank				
sexiy         3500ms: supply.ludi.yead           sexiy         3200ms: supply.ludi.yead           sexiy </th <th></th> <th><ul> <li>37.000ms : supply.rpc.multiquery</li> </ul></th> <th></th> <th></th> <th></th> <th></th>		<ul> <li>37.000ms : supply.rpc.multiquery</li> </ul>				
ksyy         , /2.00m: supp/(ud/)red         , /2.00m: supp/(ud/)red           ksyy         , /2.00m: supp/(ud/)red						
scpy         90000ms: suppi/(uid):ed         .           scpy						
ispy         4000000000000000000000000000000000000				1.1.1		
4 4000ms         41000ms         <		•		1 A A A A A A A A A A A A A A A A A A A		
suppy/         90000mm: suppy/(uui)/read           suppy/         31000mm: suppy/(uui)/read           suppy/         31000mm: suppy/(uui)/read           suppy/         33000mm: suppy/(uui)/read           suppy         33000mm: suppy/(uui)/r		•				
augsy         S1000ms:supply(Judi)red           augsy         S1000ms:		•		1.1		
suppy         31.00mm: suppy(juid):red           suppy         32.00mm: suppy(juid):red           suppy </th <th></th> <th>•</th> <th></th> <th></th> <th></th> <th></th>		•				
auspy         auspy         33.00me: supply.(Luid).read           auspy         35.00me: supply.(Luid).read           auspy         35.00me: supply.(Luid).read           auspy         35.00me: supply.(Luid).read           auspy         33.00me: supply.(Luid).read           auspy         60.00me: supply.(Luid).read           auspy         32.000me: supply.(Luid).read           auspy         32.000me: supply.(Luid).read           auspy         32.000me: supply.(Luid).read           auspy         47.000me: supply.(Luid).read           auspy         48.000me: supply.(Luid).read           auspy         58.000me: supply.(Luid).read           auspy         58.000me: supply.(Luid).read           auspy         58.000me: supply.(Luid).read <tr< th=""><th></th><th>•</th><th></th><th></th><th></th><th></th></tr<>		•				
suppy         s3.000ms: suppy/(uid).read           suppy         s3.000ms: suppy/(uid).read           suppy         supp		•				
-         supply         B5.000ms: supply.[uuid].read           -         supply         B3.000ms: supply.[uuid].read           -         supply         B3.000ms: supply.[uuid].read           -         supply         B2.000ms: supply.[uuid].read           -         supply         S2.000ms: supply.[uuid].read           -         supply         B2.000ms: supply.[uuid].read		•				
suppy         33.000ms: suppy/(uid):read           suppy         33.000ms: suppy/(uid):read           suppy         33.000ms: suppy/(uid):read           suppy         82.000ms: suppy/(uid):read           suppy         48.000ms: suppy/(uid):read           suppy         33.000ms: suppy/(uid):read           suppy         44.000ms: suppy/(uid):read           suppy         44.000ms: suppy/(uid):read           suppy         48.000ms: suppy/(uid):		•				
sppy         9.000000000000000000000000000000000000		•				
skpby         62.000ms: supply.(uuid).read           skpby         64.000ms: supply.(uuid).read           skpby         33.000ms: supply.(uuid).read           skpby         32.000ms: supply.(uuid).read           skpby         46.000ms: supply.(uuid).read           skpby         48.000ms: supply.(uuid).read           skpby         48.000ms: supply.(uuid).read           skpby         48.000ms: supply.(uuid).read           skpby         53.000ms: supply.(uuid).read           skpby         53.000ms: supply.(uuid).read           skpby         53.000ms: supply.(uuid).read           skpby         43.000ms: supply.(uuid).read           skpby         43.000ms: supply.(uuid).read           skpby         53.000ms: supply.(uuid).read <td< th=""><th></th><th>•</th><th></th><th></th><th></th><th></th></td<>		•				
supply         -         46.000ms : supply. (uuid).read           supply         -         33.000ms : supply. (uuid).read           supply         -         32.000ms : supply. (uuid).read           supply         -         32.000ms : supply. (uuid).read           supply         -         -         47.000ms : supply. (uuid).read           supply         -         -         28.000ms : supply. (uuid).read           supply         -         -         28.000ms : supply. (uuid).read           supply         -         -         28.000ms : supply. (uuid).read           supply         -         -         -         28.000ms : supply. (uuid).read           supply         -         -         -         -         -           supply         -         -         -         -		•				
supply         33.000ms : supply.(uuid).read           supply         32.000ms : supply.(uuid).read           supply         32.000ms : supply.(uuid).read           supply         47.000ms : supply.(uuid).read           supply         47.000ms : supply.(uuid).read           supply         47.000ms : supply.(uuid).read           supply         47.000ms : supply.(uuid).read           supply         48.000ms : supply.(uuid).read           supply         48.000ms : supply.(uuid).read           supply         48.000ms : supply.(uuid).read           supply         58.000ms : supply.(uuid).read           supply         58.000ms : supply.(uuid).read           supply         65.000ms : supply.(uuid).read           supply         55.000ms : supply.(uuid).read           supply         48.000ms : supply.(uuid).read           supply         55.000ms : supply.(uuid).read           supply         55.000ms : supply.(uuid).read           supply         65.000ms : supply.(uuid).read           supply         65.000ms : supply.(uuid).read           supply         55.000ms : supply.(uuid).read           supply         64.000ms : supply.(uuid).read           supply         55.000ms : supply.(uuid).read           supply         64.000ms : supply.(		•				
suppy         32.00ms: supply.[uiid].read           suppy         47.000ms: supply.[uiid].read           suppy         28.000ms: supply.[uiid].read           suppy         28.000ms: supply.[uiid].read           suppy         48.000ms: supply.[uiid].read           suppy         48.000ms: supply.[uiid].read           suppy         48.000ms: supply.[uiid].read           suppy         48.000ms: supply.[uiid].read           suppy         53.000ms: supply.[uiid].read           suppy         53.000ms: supply.[uiid].read           suppy         53.000ms: supply.[uiid].read           suppy         53.000ms: supply.[uiid].read           suppy         65.000ms: supply.[uiid].read		*				
suppy       -       47.000ms: supply.(uid).read       28.000ms: supply.(uid).read         suppy       -       48.000ms: supply.(uid).read       48.000ms: supply.(uid).read         suppy       -       48.000ms: supply.(uid).read       48.000ms: supply.(uid).read         suppy       -       48.000ms: supply.(uid).read       53.000ms: supply.(uid).read         suppy       -       -       65.000ms: supply.(uid).read       -         suppy       -       -       -       65.000ms: supply.(uid).read       -         suppy       -       -       -       -       -       -       -         suppy       -<						
supply         -         28.000ms : supply.(uid).read         -           supply         -         48.000ms : supply.(uid).read         -           supply         -         46.000ms : supply.(uid).read         -           supply         -         -         53.000ms : supply.(uid).read         -           supply         -         -         -         -         -           supply         -         -         -         -         -         -           supply         - <td< th=""><th></th><th></th><th></th><th></th><th></th><th></th></td<>						
supply       48.000ms: supply.(uuid).read         supply       48.000ms: supply.(uuid).read         supply       53.000ms: supply.(uuid).read         supply       55.000ms: supply.(uuid).read						
supply       46.000ms: supply.(uuid).read         supply       53.000ms: supply.(uuid).read         supply       65.000ms: supply.(uuid).read         supply       25.000ms: supply.(uuid).read         supply       34.000ms: supply.(uuid).read					4	
supply       53.000ms : supply.(uuid).read         supply       65.000ms : supply.(uuid).read         supply       25.000ms : supply.(uuid).read         supply       34.000ms : supp				- 40		
supply       65.000ms : supply.(uuid).read         supply       25.000ms : supply.(uuid).read         supply       34.000ms : supply.(uuid).read						
supply       . <th></th> <th></th> <th></th> <th></th> <th></th> <th></th>						
supply       34.000ms : supply.(uuid).read         supply       43.000ms : supply.(uuid).read         supply       55.000ms : supply.(uuid).read         supply       55.000ms : supply.(uuid).read         supply       28.000ms : supply.(uuid).read         supply       28.000ms : supply.(uuid).read         supply       28.000ms : supply.(uuid).read         supply       28.000ms : supply.(uuid).read         supply       29.000ms : supply.(uuid).read         supply       29.000ms : supply.(uuid).read						
supply       43.000ms : supply.{uuid}.read         supply       55.000ms : supply.{uuid}.read         supply       28.000ms : supply.{uuid}.read         supply       29.000ms : supply.{uuid}.read         supply       29.000ms : supply.{uuid}.read         supply       29.000ms : supply.{uuid}.read						
supply       5.000ms : supply.(uuid).read         supply       28.000ms : supply.(uuid).read         supply       74.000ms : supply.(uuid).read         supply       29.000ms : supply.(uuid).read						
supply       28.000ms : supply.[uuid].read         supply       74.000ms : supply.[uuid].read         supply       29.000ms : supply.[uuid].read						
supply       74.000ms : supply.{uuid}.read         supply       29.000ms : supply.{uuid}.read         supply       61.000ms : supply.{uuid}.read						
supply       29.000ms : supply.{uuid}.read         supply       61.000ms : supply.{uuid}.read						
- supply - 61.000ms : supply.{uuid}.read						
- soppy .			•			
	- suppry				30.000ms :	anthelia (nono) us



#### Services

#### 1.515s

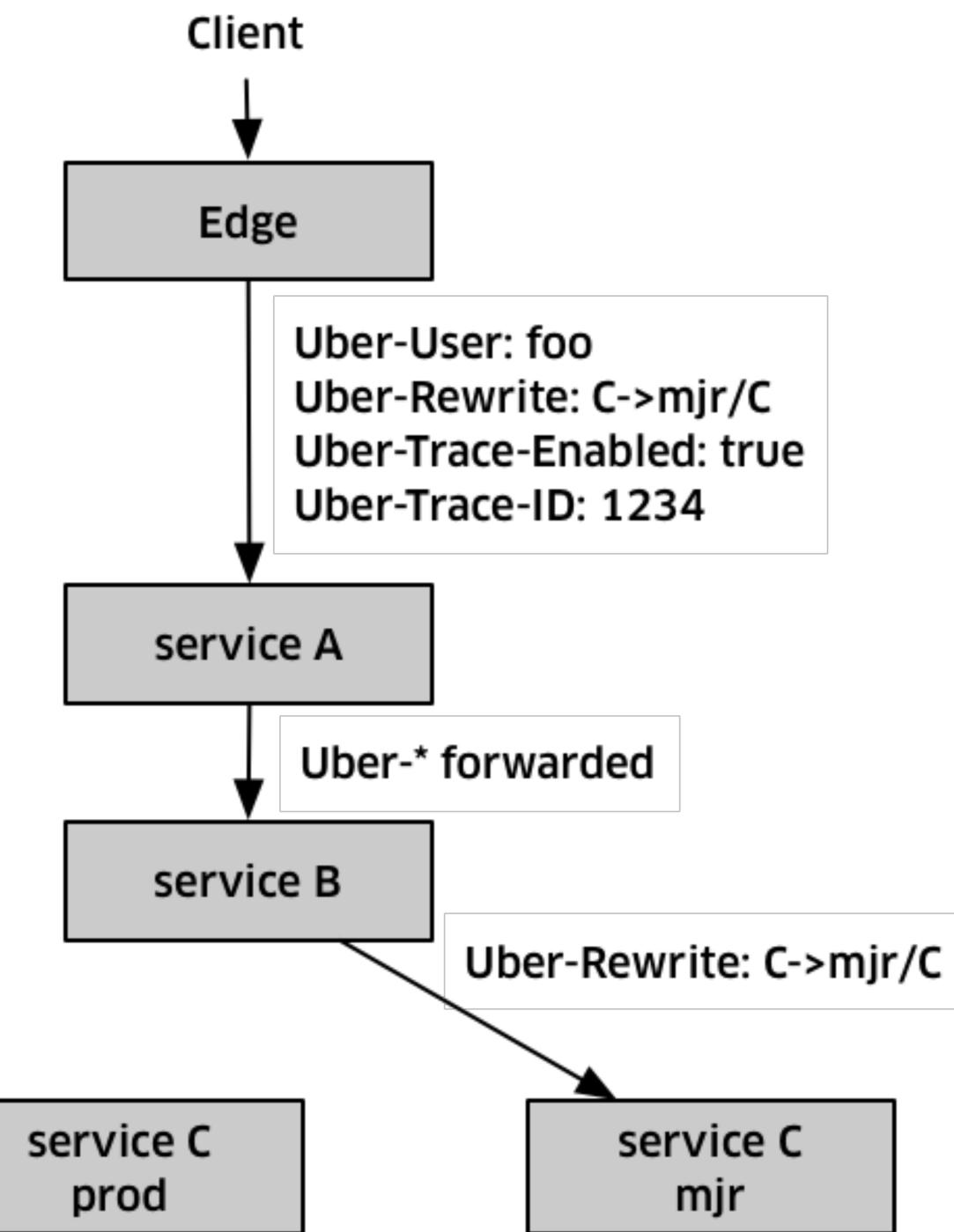
36	rvices		1.0105
-	accountmgmt	7.577s : accountmgmtservice::getallmerchants	· · ·
-	accountmgmt	-58.104ms : sql select	•
-	accountmgmt	-57.771ms : mysqldb:select	•
-	accountmgmt	<ul> <li>180.370ms : sql select</li> </ul>	
-	accountmgmt	<ul> <li>180.120ms : mysqldb:select</li> </ul>	
-	accountmgmt	<ul> <li>5.316ms : sql select</li> </ul>	
-	accountmgmt	<ul> <li>4.976ms : mysqldb:select</li> </ul>	
-	accountmgmt	<ul> <li>1.848ms : sql select</li> </ul>	•
-	accountmgmt	<ul> <li>766µ : mysqldb:select</li> </ul>	•
-	accountmgmt	<ul> <li>1.048ms : sql select</li> </ul>	•
-	accountmgmt	<ul> <li>600µ : mysqldb:select</li> </ul>	•
-	accountmgmt	<ul> <li>1.070ms : sql select</li> </ul>	•
-	accountmgmt	<ul> <li>783µ : mysqldb:select</li> </ul>	•
-	accountmgmt	<ul> <li>940µ : sql select</li> </ul>	•
-	accountmgmt	<ul> <li>624µ : mysqldb:select</li> </ul>	•
-	accountmgmt	<ul> <li>1.130ms : sql select</li> </ul>	· ·
-	accountmgmt	<ul> <li>791µ : mysqldb:select</li> </ul>	•
-	accountmgmt	<ul> <li>2.553ms : sql select</li> </ul>	· ·
-	accountmgmt	<ul> <li>814µ : mysqldb:select</li> </ul>	· ·
-	accountmgmt	<ul> <li>751µ : sql select</li> </ul>	· ·
-	accountmgmt	<ul> <li>495µ : mysqldb:select</li> </ul>	· ·
-	accountmgmt	<ul> <li>956µ : sql select</li> </ul>	· ·
-	accountmgmt	<ul> <li>734µ : mysqldb:select</li> </ul>	· ·
-	accountmgmt	<ul> <li>722µ : sql select</li> </ul>	· ·
-	accountmgmt	<ul> <li>493µ : mysqldb:select</li> </ul>	· ·
-	accountmgmt	<ul> <li>698µ : sql select</li> </ul>	· ·
-	accountmgmt	<ul> <li>469µ : mysqldb:select</li> </ul>	· ·
-	accountmgmt	<ul> <li>692µ : sql select</li> </ul>	· ·
-	accountmgmt	<ul> <li>479µ : mysqldb:select</li> </ul>	· ·
-	accountmgmt	<ul> <li>669µ : sql select</li> </ul>	•
-	accountmgmt	<ul> <li>455µ : mysqldb:select</li> </ul>	· ·
-	accountmgmt	<ul> <li>702µ : sql select</li> </ul>	· ·
-	accountmgmt	<ul> <li>475µ : mysqldb:select</li> </ul>	· ·
-	accountmgmt	<ul> <li>719µ : sql select</li> </ul>	

3.031s

•	•
· · ·	
· ·	
· ·	
· · ·	
·	
	•
•	•
	•
•	•
· ·	•
	•
•	•
· ·	
· · ·	
· ·	
· ·	
· ·	
	•
	•
	•
	•
•	•
· ·	•
· ·	•
-	
· · · · · · · · · · · · · · · · · · ·	•

## TRACING

#### Probably want sampling WIWIK: cross-lang context propagation



## LOGGING

### Need consistent, structured logging Multiple languages makes this hard Logs are not for humans

zap godoc reference build passing coverage 99% 4

Blazing fast, structured, leveled logging in Go.

#### Installation

go get -u github.com/uber-go/zap

#### Structure

Zap takes an opinionated stance on logging and doesn't provide any printf -style helpers. Rather than logger.Printf("Failed to fetch URL %s (attempt %v), sleeping %s before retry.", url, tryNum, sleepFor), Zap encourages the more structured

```
logger.Info("Failed to fetch URL.",
 zap.String("url", url),
 zap.Int("attempt", tryNum),
 zap.Duration("backoff", sleepFor),
```

This a bit more verbose, but it enables powerful ad-hoc analysis, flexible dashboarding, and accurate message bucketing. In short, it helps you get the most out of tools like ELK, Splunk, and Sentry. All log messages are JSON-serialized, though PRs to support other formats are welcome.

For compatibility with the standard library and bark, zap provides the zwrap.Standardize and zbark.Barkify wrappers. Both are slower than the core zap logger, but faster than the libraries they replace.



Log a message using a logger that already has 10 fields of context:

Library	Time	<b>Bytes Allocated</b>	<b>Objects Allocated</b>
🗲 zap	231 ns/op	0 B/op	0 allocs/op
logrus	8532 ns/op	3438 B/op	61 allocs/op
go-kit	6874 ns/op	2486 B/op	48 allocs/op
log15	20462 ns/op	4118 B/op	70 allocs/op
apex/log	13886 ns/op	2384 B/op	48 allocs/op

Log a static string, without any context or printf -style formatting:

Library	Time	<b>Bytes Allocated</b>	<b>Objects Allocated</b>		
🗲 zap	222 ns/op	0 B/op	0 allocs/op		
standard library	565 ns/op	32 B/op	2 allocs/op		
logrus	3085 ns/op	1336 B/op	26 allocs/op		
go-kit	1061 ns/op	624 B/op	13 allocs/op		
log15	5462 ns/op	1351 B/op	23 allocs/op		
apex/log	3009 ns/op	584 B/op	11 allocs/op		

## 

#### Logging floods can amplify problems WIWIK: Accounting

### Need to test against production Without breaking metrics Preferably all the time WIWIK: all systems need to handle "test" traffic

### LOAD TESTING

### FAILURE TESTING

#### WIWIK: people won't like it

## MIGRATIONS

### Old stuff still has to work What happened to immutable? WIWIK: mandates are bad

## **OPEN SOURCE**

### Build/buy tradeoff is hard Commoditization WIWIK: this will make people sad

#### Services allow people to play politics Company > Team > Self

### POLTICS

### TRADEOFFS

#### Everything is a tradeoff Try to make them intentionally

