Evolving systems design

from unreliable rpc to resilience with **Linkerd**

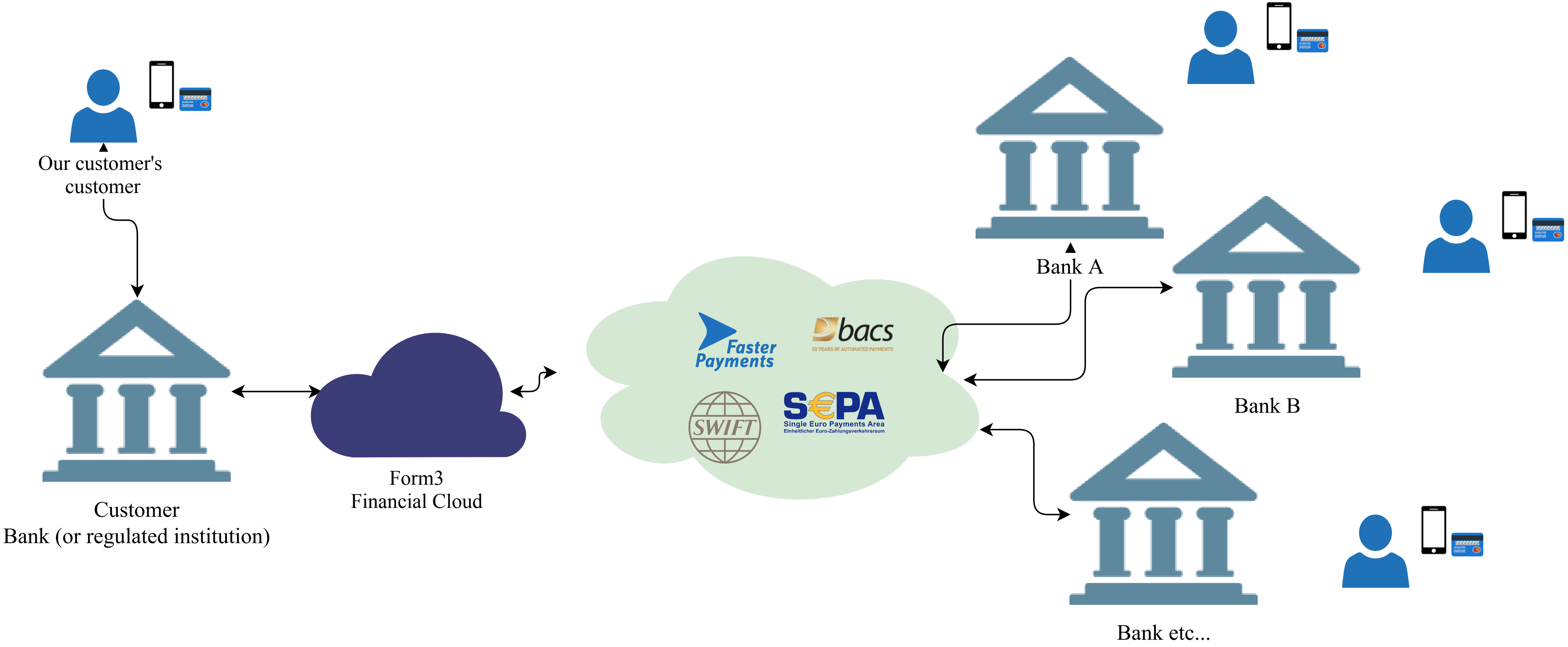
Edward Wilde

8 May, 2018

FORM3
FINANCIAL CLOUD

FORM3

FINANCIAL CLOUD


Faster Payments
bacs
50 YEARS OF AUTOMATED PAYMENTS
SWIFT
SEPA
Single Euro Payments Area
Einheitlicher Euro-Zahlungsverkehrsraum

Our customer's customer

Customer
Bank (or regulated institution)

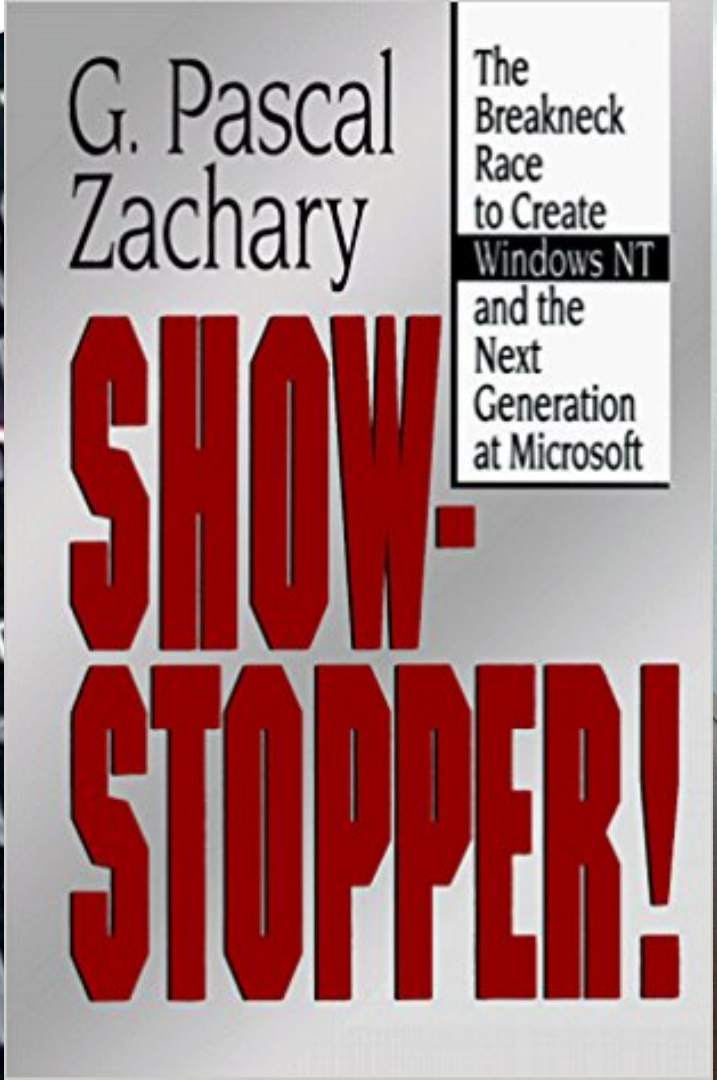
Form3
Financial Cloud

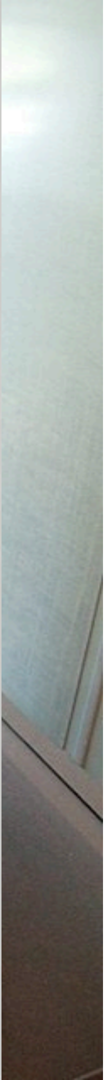
Bank A

Bank B

Bank etc...

"A gem of a story that has never been so well told"
NEW YORK TIMES
DEALER
LIGHT
Xerox PARC
of the Co
MIC
HI


G. Pascal Zachary
SHOW-STOPPER!
The Breakneck Race to Create Windows NT and the Next Generation at Microsoft



A story about unreliable RPC

- System 1

- Increased reliability
- Reduced overall latency

But….

- Did not manage to reduce tail-latency
- Did not manage to protect services

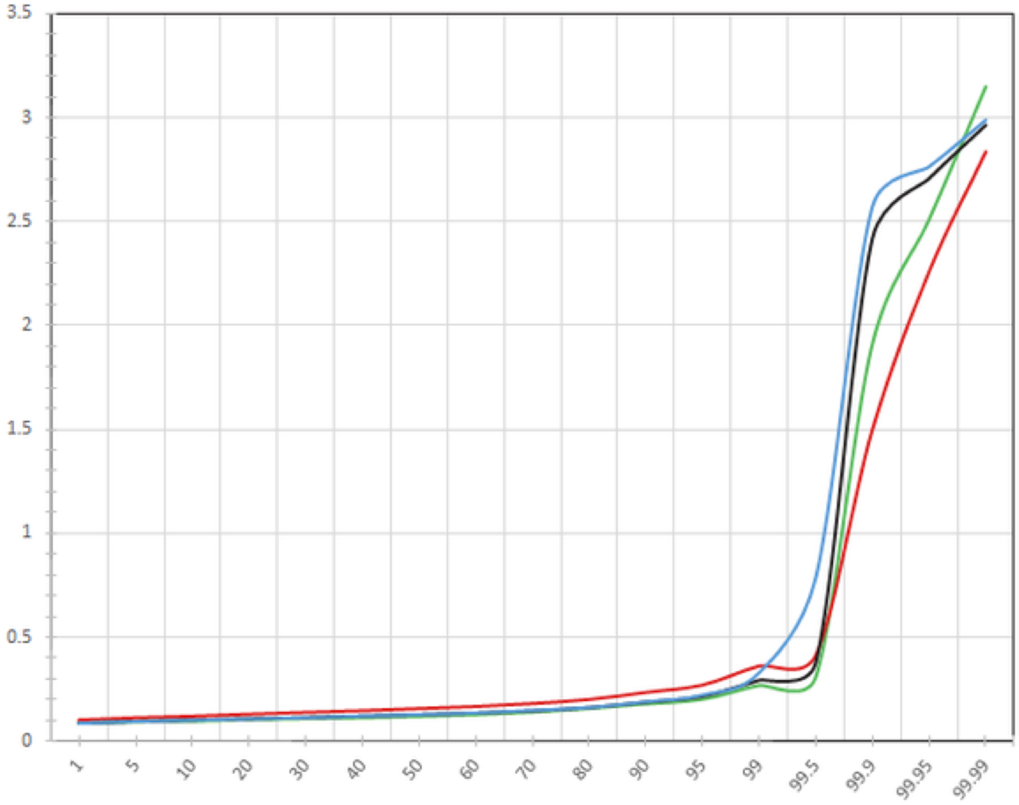
- System 2

- Increased reliability
- Reduced overall latency
- Protected services

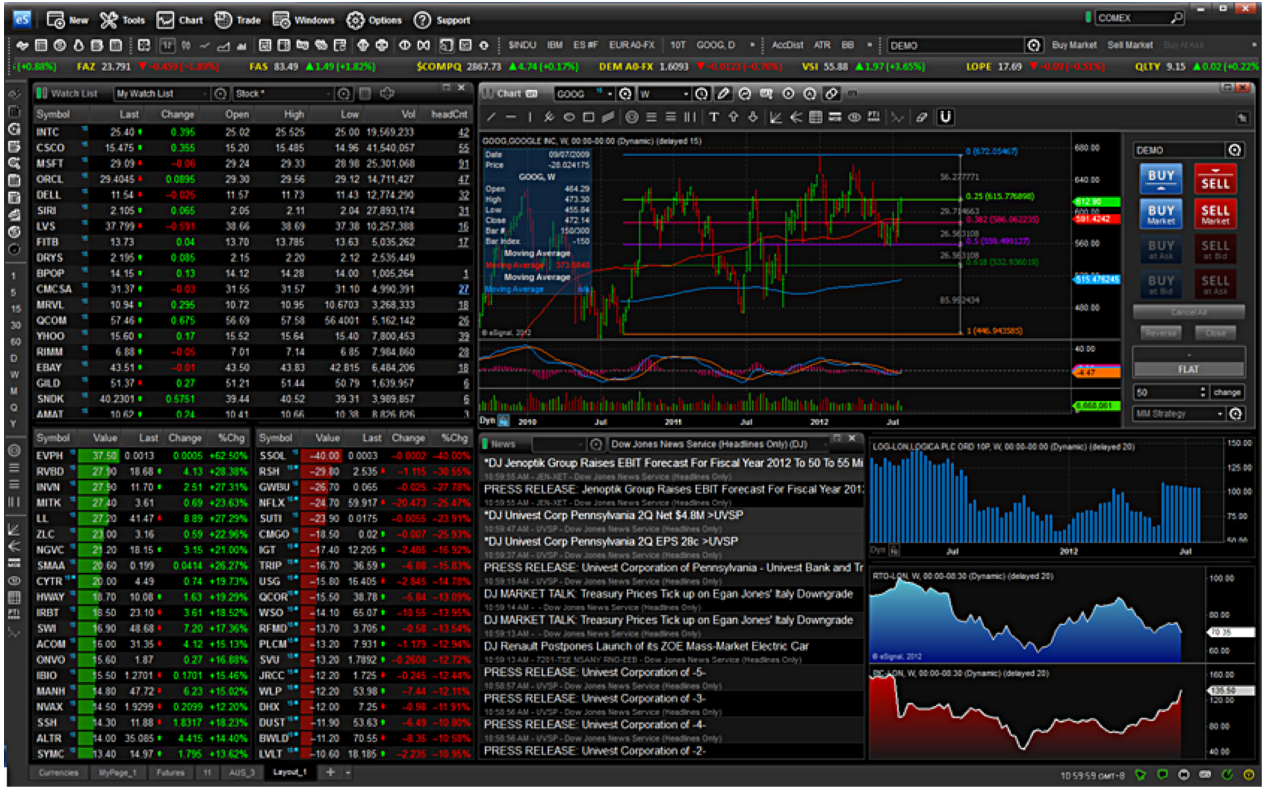
But…

- Still need to work more on tail-latency

Tail latency



System 1



Symbol	Last	Change	Open	High	Low	Vol	headCnt
INTC	25.40	0.395	25.02	25.525	25.00	19,569,233	42
CSCO	15.475	0.355	15.20	15.485	14.96	41,540,057	55
MSFT	29.09	-0.06	29.24	29.33	28.98	25,301,068	91
ORCL	29.4045	0.0895	29.30	29.56	29.12	14,711,427	47
DELL	11.54	-0.025	11.57	11.73	11.43	12,774,290	32
SIRI	2.105	0.065	2.05	2.11	2.04	27,893,174	31
LVS	37.799	-0.591	38.66	38.69	37.38	10,257,388	16
FITB	13.73	0.04	13.70	13.785	13.63	5,035,262	17
DRYS	2.195	0.085	2.15	2.20	2.12	2,535,449	
BPOP	14.15	0.13	14.12	14.28	14.00	1,005,264	1
CMCSA	31.37	-0.03	31.55	31.57	31.10	4,990,391	27
MRVL	10.94	0.295	10.72	10.95	10.6703	3,268,333	18
QCOM	57.46	0.675	56.69	57.58	56.4001	5,162,142	26
YHOO	15.60	0.17	15.52	15.64	15.40	7,800,453	39
RIMM	6.88	-0.05	7.01	7.14	6.85	7,984,860	28
EBAY	43.51	-0.01	43.50	43.83	42.815	6,484,206	18
GILD	51.37	0.27	51.21	51.44	50.79	1,639,957	6
SNDK	40.2301	0.5751	39.44	40.52	39.31	3,989,857	6
AMAT	10.62	0.24	10.41	10.66	10.38	8,826,826	3

Symbol	Value	Last	Change	%Chg
EVPH	37.50	0.0013	0.0005	+62.50%
RVBD	27.90	18.68	4.13	+28.38%
INVN	27.50	11.70	2.51	+27.31%
MITK	27.40	3.61	0.69	+23.63%
LL	27.20	41.47	8.89	+27.29%
ZLC	23.00	3.16	0.59	+22.96%
NGVC	21.20	18.15	3.15	+21.00%
SMAA	20.60	0.199	0.0414	+26.27%
CYTR	20.00	4.49	0.74	+19.73%
HWAY	18.70	10.08	1.63	+19.29%
IRBT	18.50	23.10	3.61	+18.52%
SWI	16.90	48.68	7.20	+17.36%
ACOM	16.00	31.35	4.12	+15.13%
ONVO	15.60	1.87	0.27	+16.88%
IBIO	15.50	1.2701	0.1701	+15.46%
MANH	14.80	47.72	6.23	+15.02%
NVAX	14.50	1.9299	0.2099	+12.20%
SSH	14.30	11.88	1.8317	+18.23%
ALTR	14.00	35.085	4.415	+14.40%
SYMC	13.40	14.97	1.795	+13.62%

Symbol	Value	Last	Change	%Chg
SSOL	-40.00	0.0003	-0.0002	-40.00%
RSH	-29.80	2.535	-1.115	-30.55%
GWBU	-26.70	0.065	-0.025	-27.78%
NFLX	-24.70	59.917	-20.473	-25.47%
SUTI	-23.90	0.0175	-0.0055	-23.91%
CMGO	-18.50	0.02	-0.007	-25.93%
IGT	-17.40	12.205	-2.485	-16.92%
TRIP	-16.70	36.59	-6.88	-15.83%
USG	-15.80	16.405	-2.845	-14.78%
QCOR	-15.50	38.78	-5.84	-13.09%
WSO	-14.10	65.07	-10.55	-13.95%
RFMD	-13.70	3.705	-0.58	-13.54%
PLCM	-13.20	7.931	-1.179	-12.94%
SVU	-13.20	1.7892	-0.2608	-12.72%
JRCC	-12.20	1.725	-0.245	-12.44%
WLP	-12.20	53.98	-7.44	-12.11%
DHX	-12.00	7.25	-0.98	-11.91%
DUST	-11.90	53.63	-6.49	-10.80%
BWLD	-11.20	70.55	-8.35	-10.58%
LVLT	-10.60	18.185	-2.235	-10.95%

*DJ Jenoptik Group Raises EBIT Forecast For Fiscal Year 2012 To 50 To 55 Mi
10:59:55 AM - JEN-XET - Dow Jones News Service (Headlines Only)

PRESS RELEASE: Jenoptik Group Raises EBIT Forecast For Fiscal Year 201:
10:59:55 AM - JEN-XET - Dow Jones News Service (Headlines Only)

*DJ Univest Corp Pennsylvania 2Q Net $4.8M >UVSP
10:59:47 AM - UVSP - Dow Jones News Service (Headlines Only)

*DJ Univest Corp Pennsylvania 2Q EPS 28c >UVSP
10:59:37 AM - UVSP - Dow Jones News Service (Headlines Only)

PRESS RELEASE: Univest Corporation of Pennsylvania - Univest Bank and Tr
10:59:15 AM - UVSP - Dow Jones News Service (Headlines Only)

DJ MARKET TALK: Treasury Prices Tick up on Egan Jones' Italy Downgrade
10:59:14 AM - - Dow Jones News Service (Headlines Only)

DJ MARKET TALK: Treasury Prices Tick up on Egan Jones' Italy Downgrade
10:59:13 AM - - Dow Jones News Service (Headlines Only)

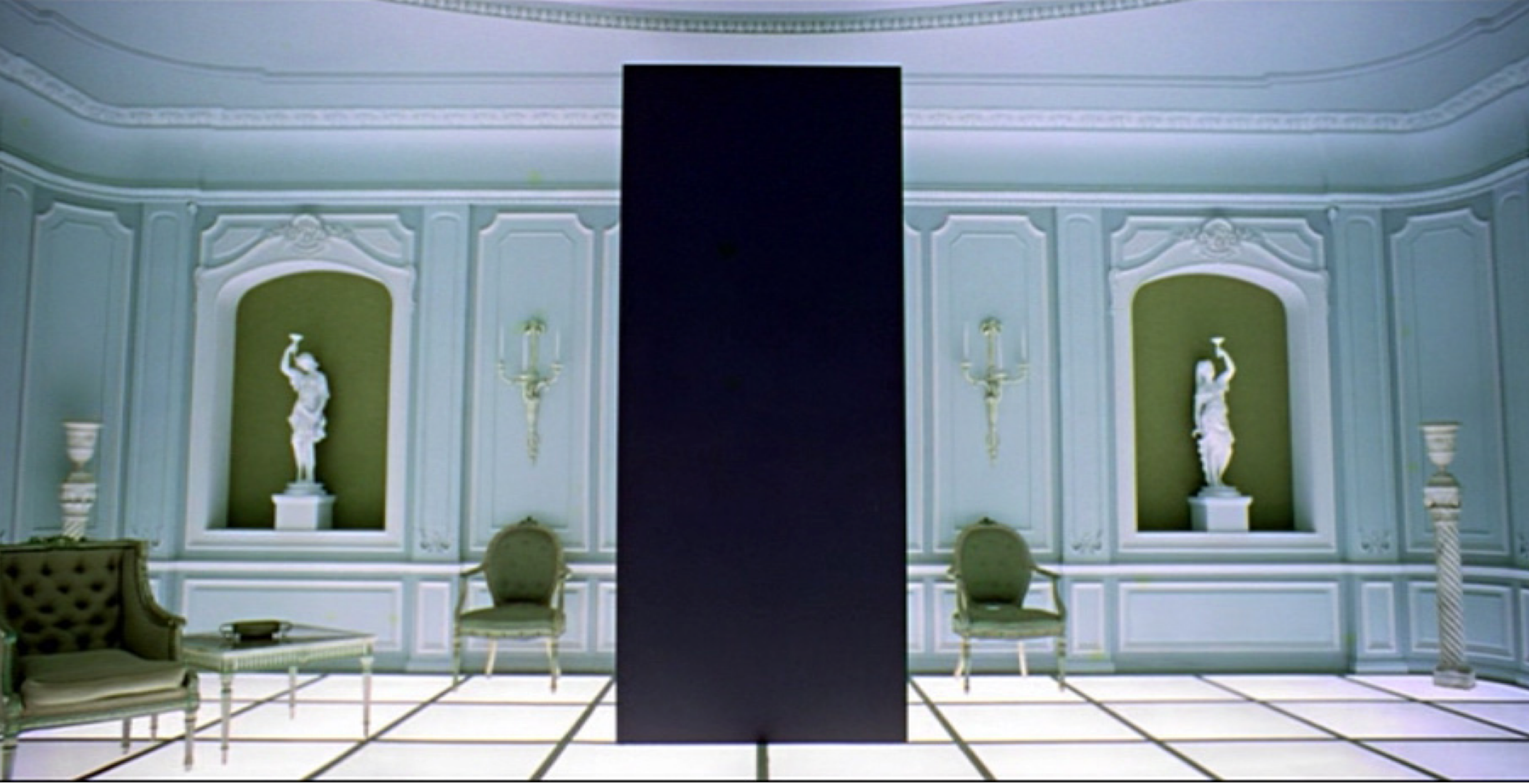
DJ Renault Postpones Launch of its ZOE Mass-Market Electric Car
10:59:13 AM - 7201-TSE NSANY RNO-EEB - Dow Jones News Service (Headlines Only)

PRESS RELEASE: Univest Corporation of -5-
10:58:57 AM - UVSP - Dow Jones News Service (Headlines Only)

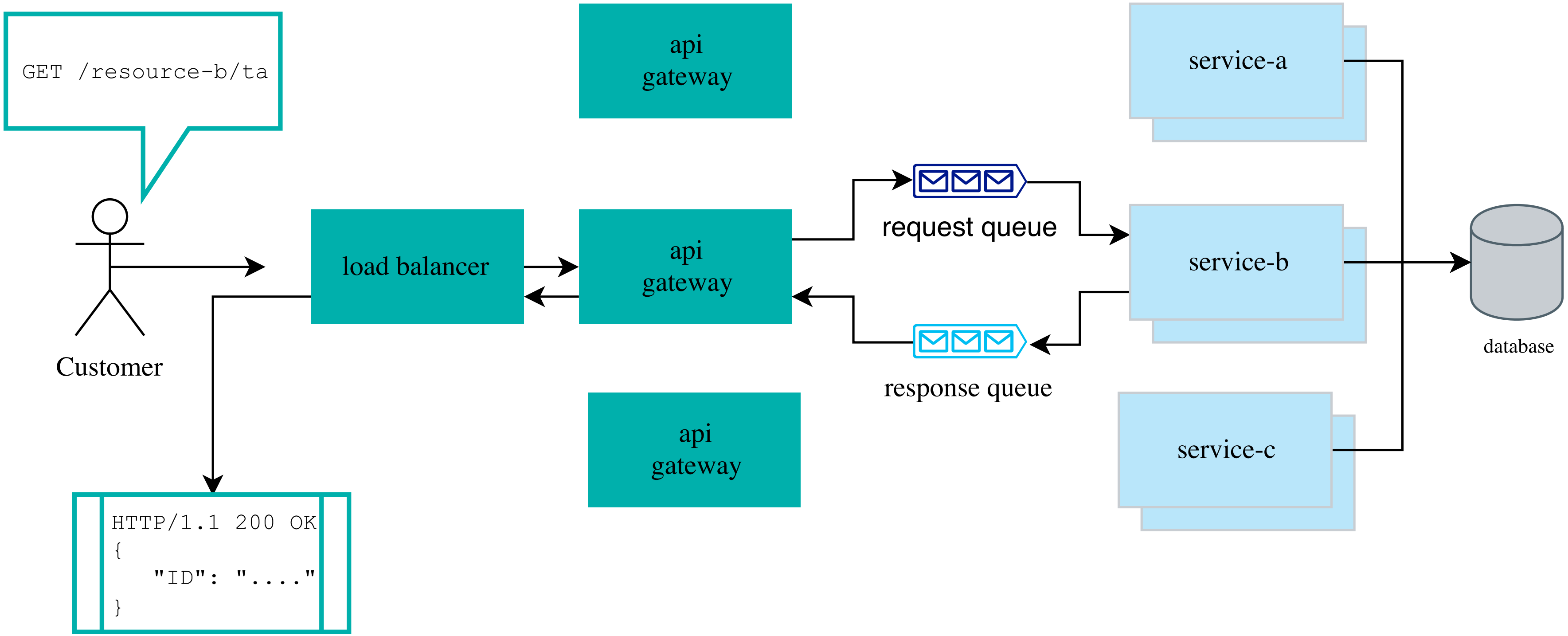
PRESS RELEASE: Univest Corporation of -3-
10:58:56 AM - UVSP - Dow Jones News Service (Headlines Only)

PRESS RELEASE: Univest Corporation of -4-
10:58:56 AM - UVSP - Dow Jones News Service (Headlines Only)

PRESS RELEASE: Univest Corporation of -2-



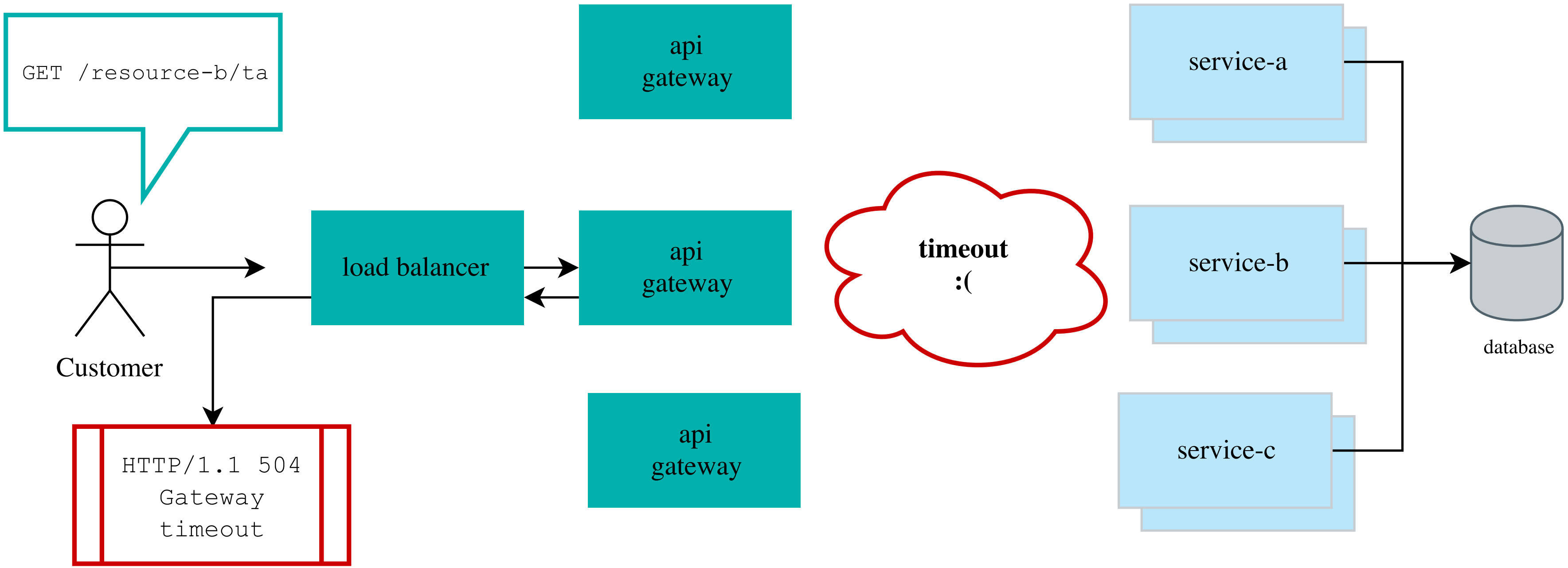
System 1



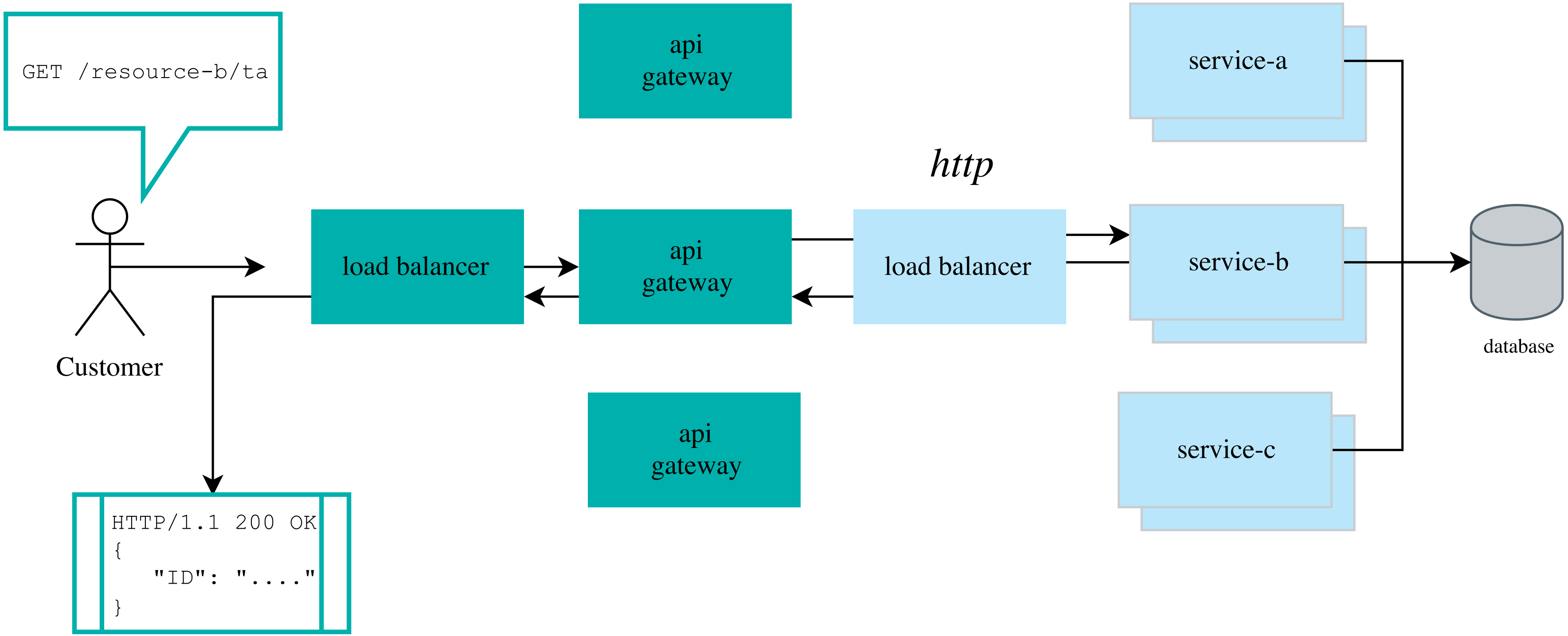
```
GET /resource-b/ta
```

```
HTTP/1.1 200 OK
{
    "ID": "...."
}
```

Timeout!


GET /resource-b/ta
Customer
load balancer
api gateway
api gateway
api gateway
timeout :(
service-a
service-b
service-c
database
HTTP/1.1 504 Gateway timeout

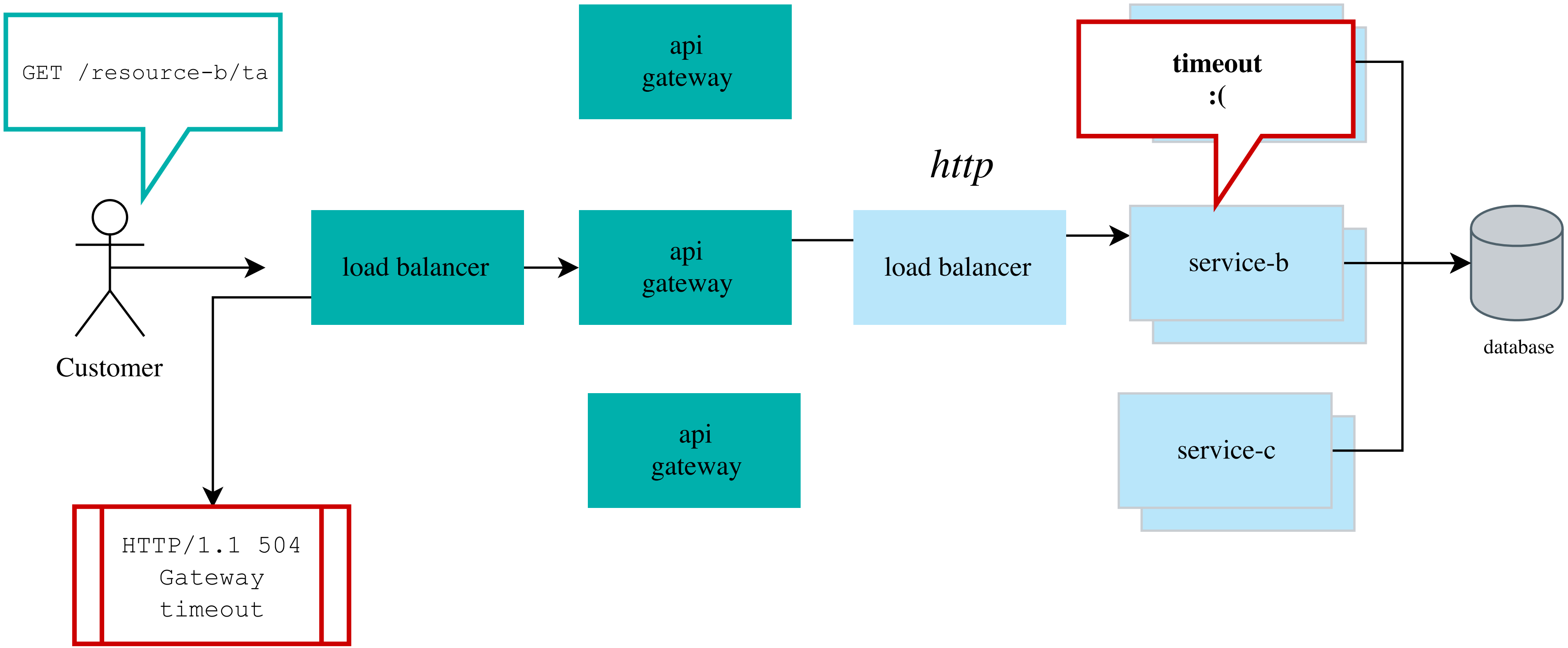
Let’s try http



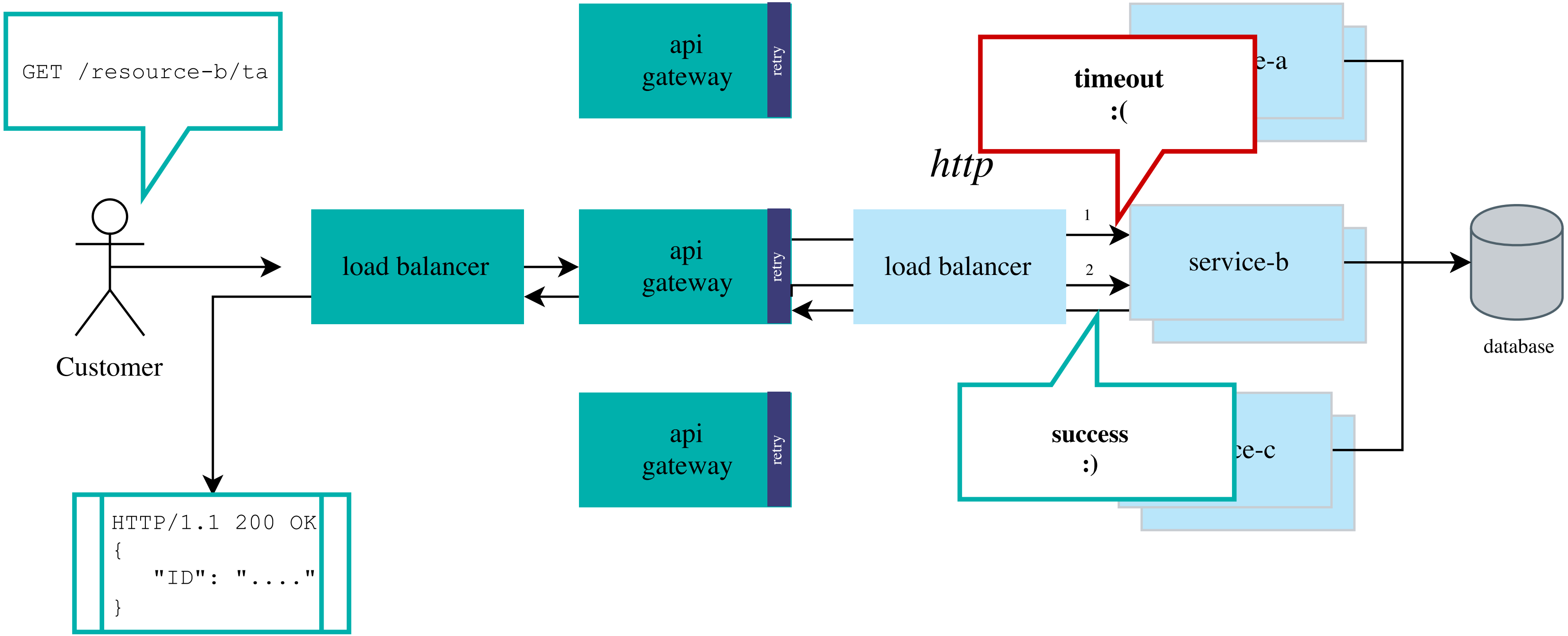
```
GET /resource-b/ta
```

```
HTTP/1.1 200 OK
{
   "ID": "...."
}
```

Oops still timeout

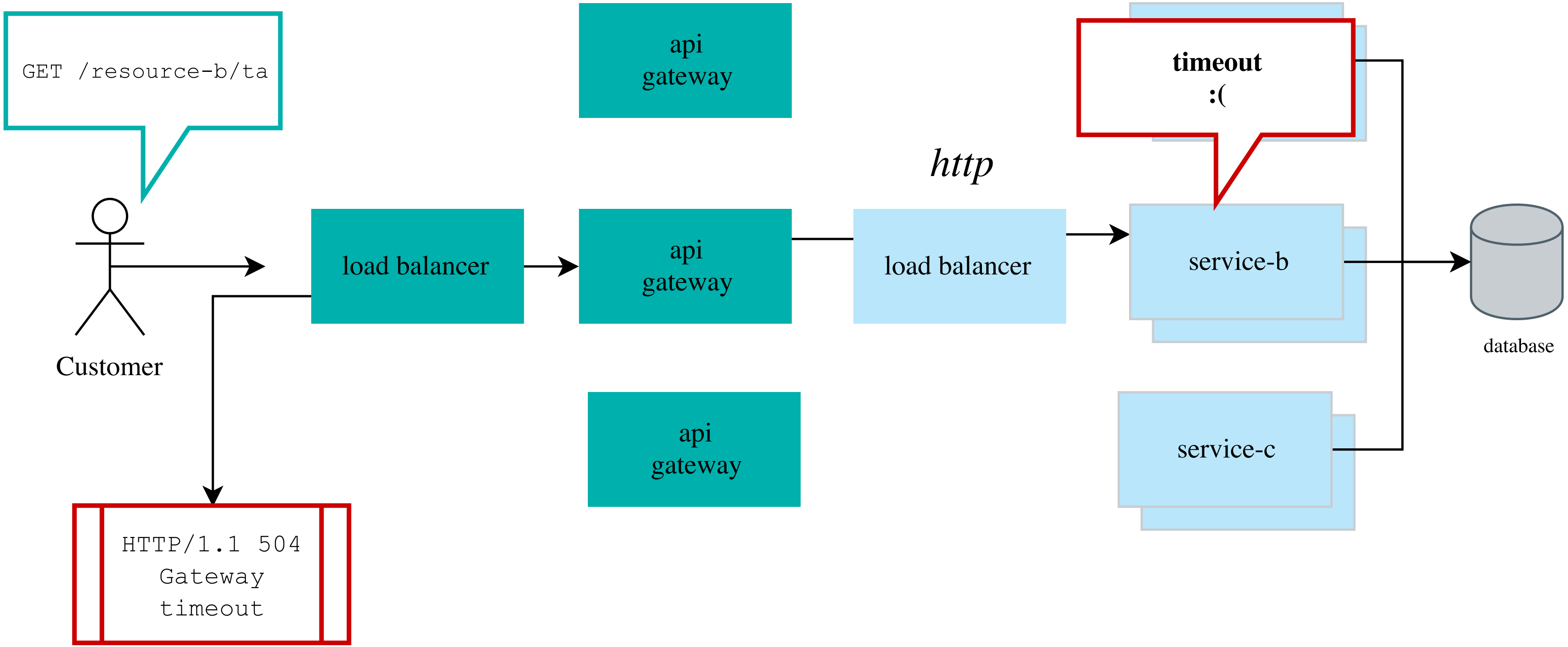


Try that again

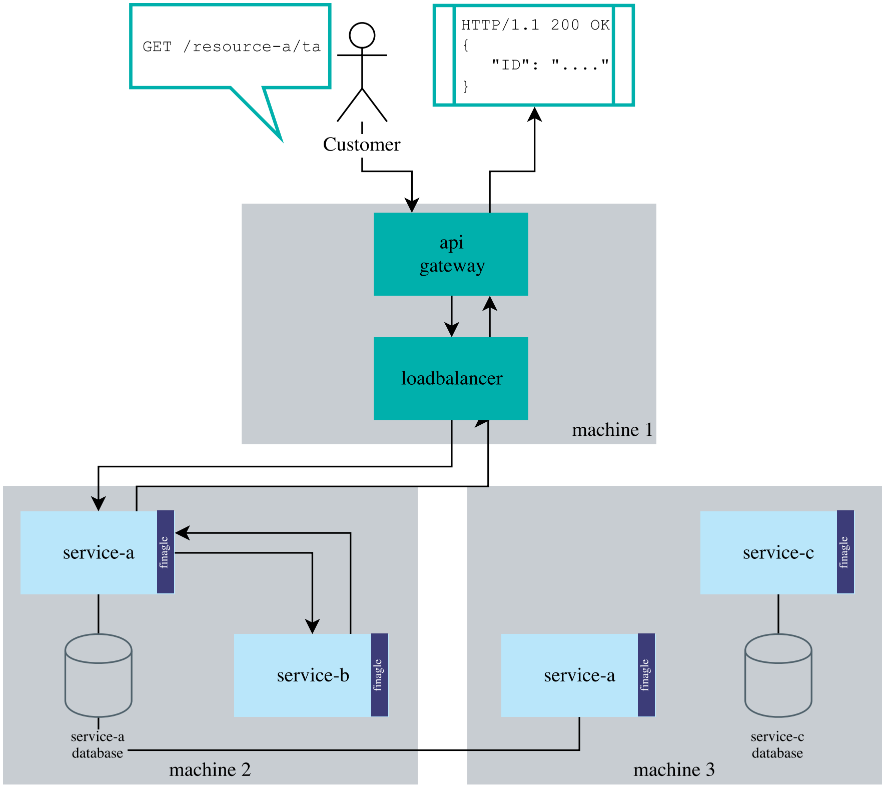


```
HTTP/1.1 200 OK
{
    "ID": "...."
}
```

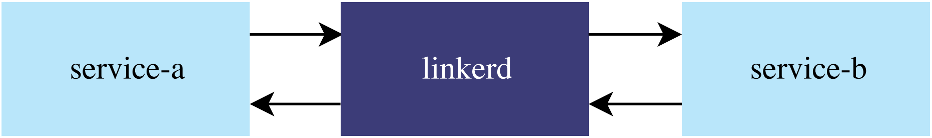
Busy day, more timeout



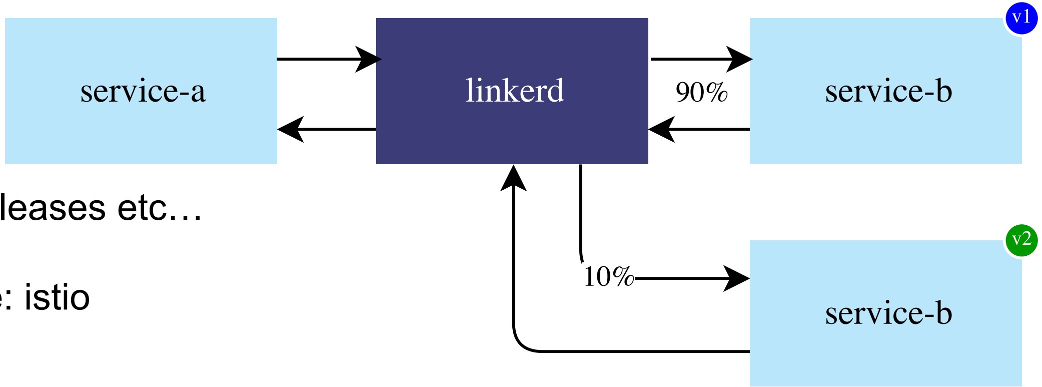
System 2: Service to service communication in a microservices architecture



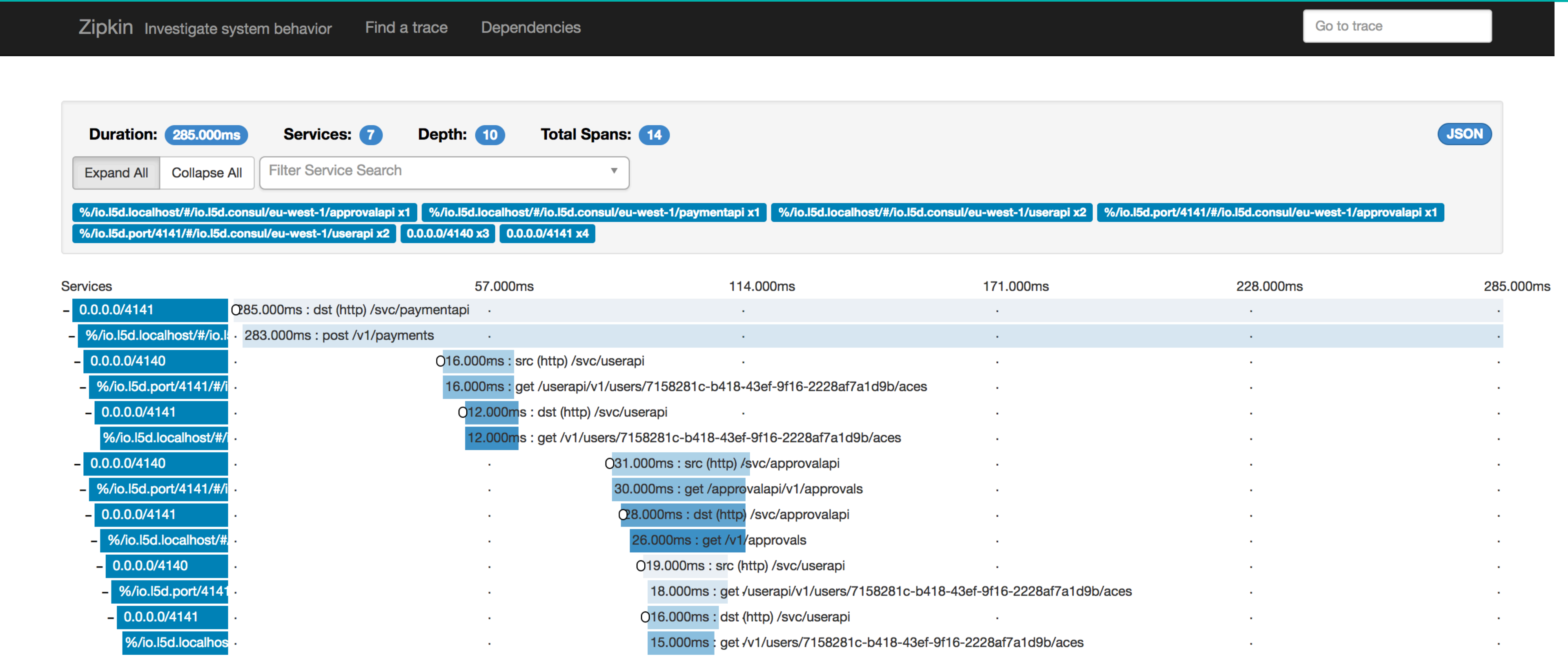
Service mesh



- Retry failures / timeout
- Implements circuit breaker
- Security
- Routing logic & service discovery
- Blue / green deployments: canary releases etc…
- Distributed Tracing
- Other services meshes are available: istio



Tracing



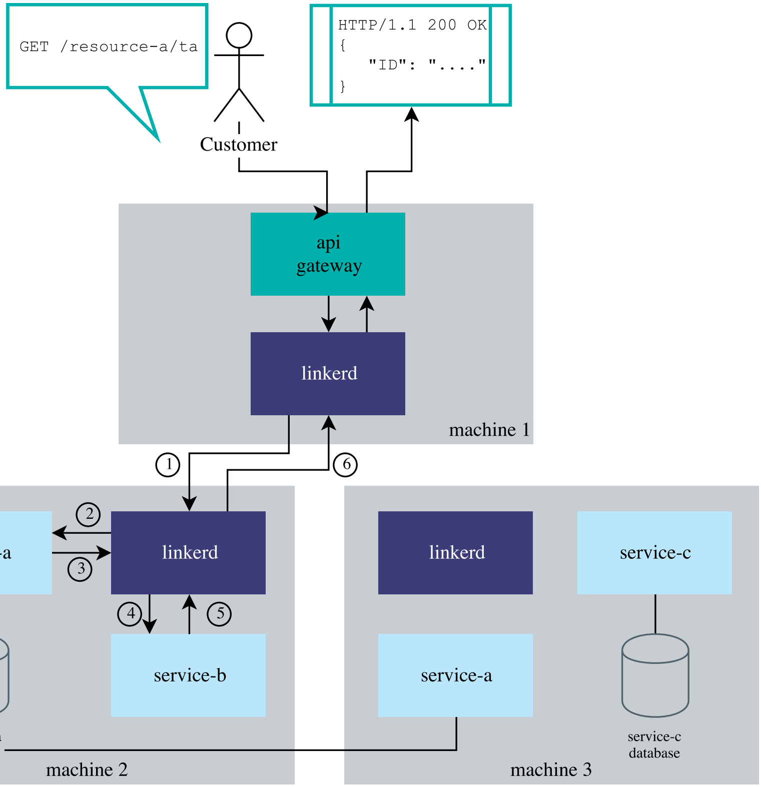
Duration: 285.000ms **Services:** 7 **Depth:** 10 **Total Spans:** 14

%/io.l5d.localhost/#/io.l5d.consul/eu-west-1/approvalapi x1 %/io.l5d.localhost/#/io.l5d.consul/eu-west-1/paymentapi x1 %/io.l5d.localhost/#/io.l5d.consul/eu-west-1/userapi x2 %/io.l5d.port/4141/#/io.l5d.consul/eu-west-1/approvalapi x1 %/io.l5d.port/4141/#/io.l5d.consul/eu-west-1/userapi x2 0.0.0.0/4140 x3 0.0.0.0/4141 x4

Services	57.000ms 114.000ms 171.000ms 228.000ms 285.000ms
0.0.0.0/4141	285.000ms : dst (http) /svc/paymentapi
%/io.l5d.localhost/#/io.l	283.000ms : post /v1/payments
0.0.0.0/4140	16.000ms : src (http) /svc/userapi
%/io.l5d.port/4141/#/i	16.000ms : get /userapi/v1/users/7158281c-b418-43ef-9f16-2228af7a1d9b/aces
0.0.0.0/4141	12.000ms : dst (http) /svc/userapi
%/io.l5d.localhost/#/	12.000ms : get /v1/users/7158281c-b418-43ef-9f16-2228af7a1d9b/aces
0.0.0.0/4140	31.000ms : src (http) /svc/approvalapi
%/io.l5d.port/4141/#/i	30.000ms : get /approvalapi/v1/approvals
0.0.0.0/4141	28.000ms : dst (http) /svc/approvalapi
%/io.l5d.localhost/#	26.000ms : get /v1/approvals
0.0.0.0/4140	19.000ms : src (http) /svc/userapi
%/io.l5d.port/4141	18.000ms : get /userapi/v1/users/7158281c-b418-43ef-9f16-2228af7a1d9b/aces
0.0.0.0/4141	16.000ms : dst (http) /svc/userapi
%/io.l5d.localhos	15.000ms : get /v1/users/7158281c-b418-43ef-9f16-2228af7a1d9b/aces

FORM3 FINANCIAL CLOUD

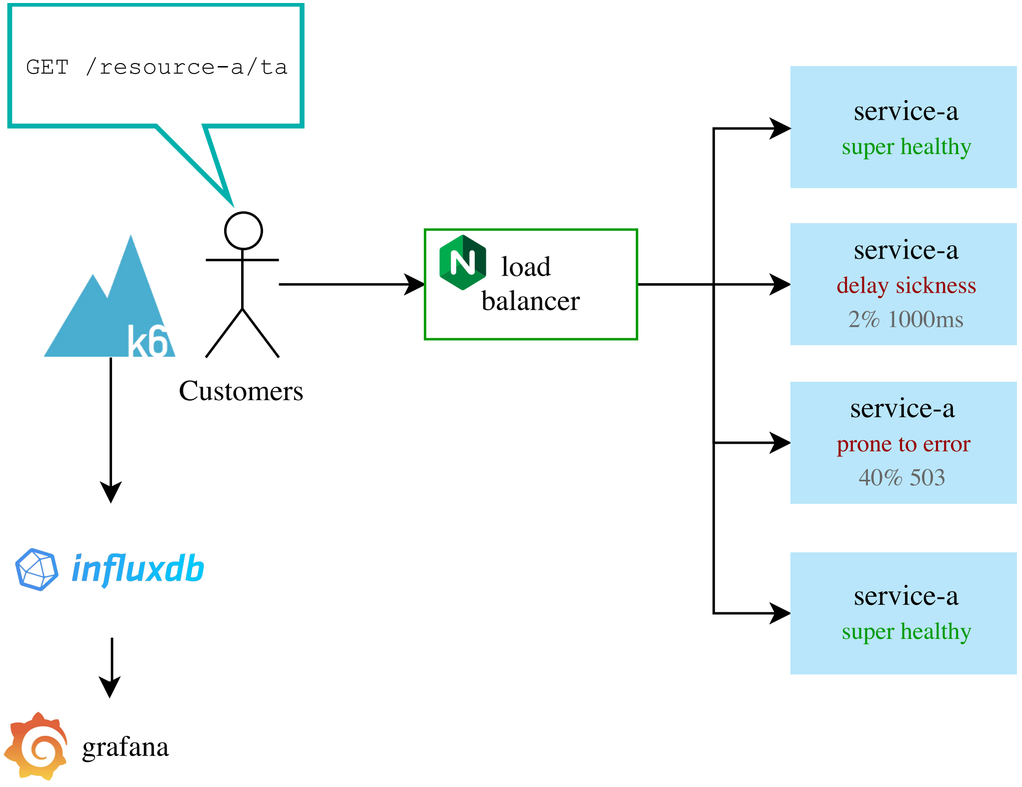
Service mesh in a microservices architecture



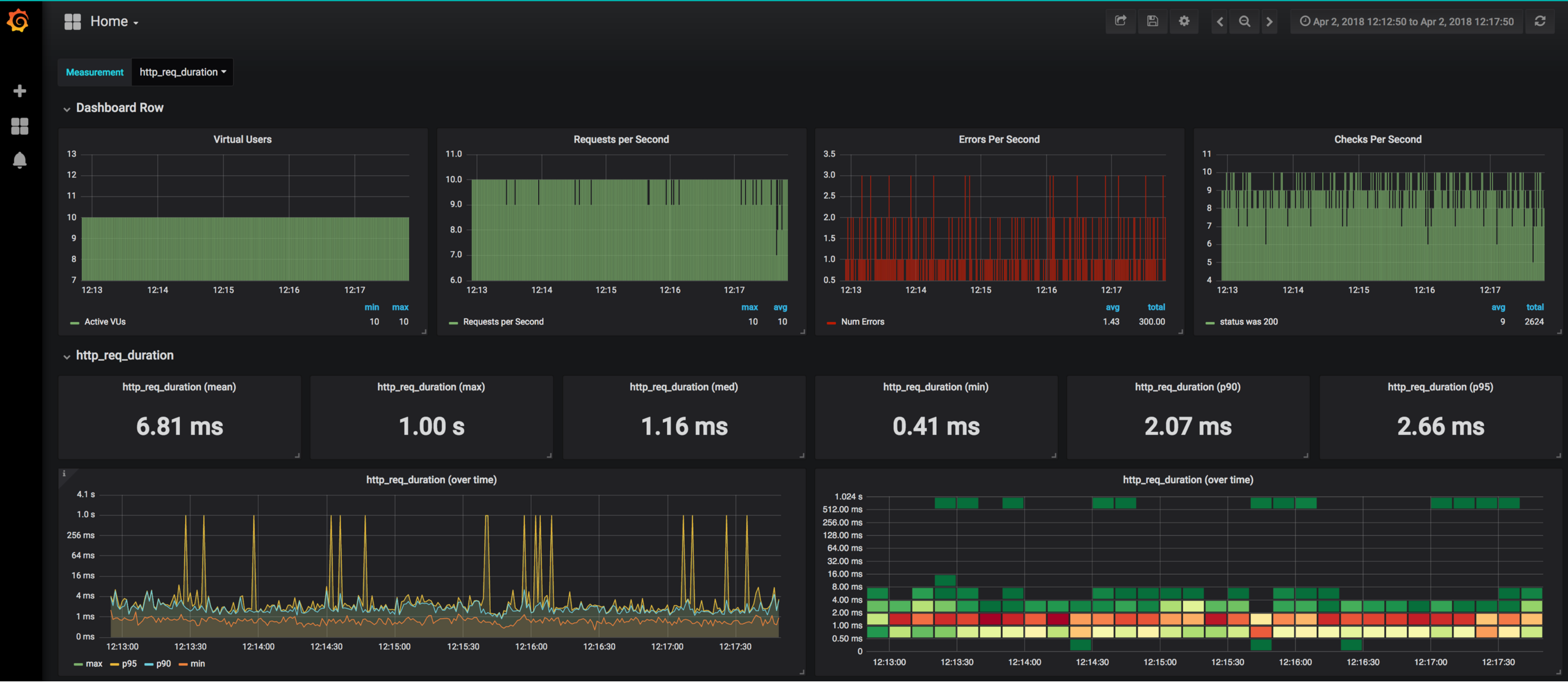
```
GET /resource-a/ta
```

```
HTTP/1.1 200 OK
{
   "ID": "...."
}
```

Demo: system-1



System 1: Response times



Home

Apr 2, 2018 12:12:50 to Apr 2, 2018 12:17:50

Measurement: http_req_duration

Dashboard Row

Virtual Users

	min	max
Active VUs	10	10

Requests per Second

	max	avg
Requests per Second	10	10

Errors Per Second

	avg	total
Num Errors	1.43	300.00

Checks Per Second

	avg	total
status was 200	9	2624

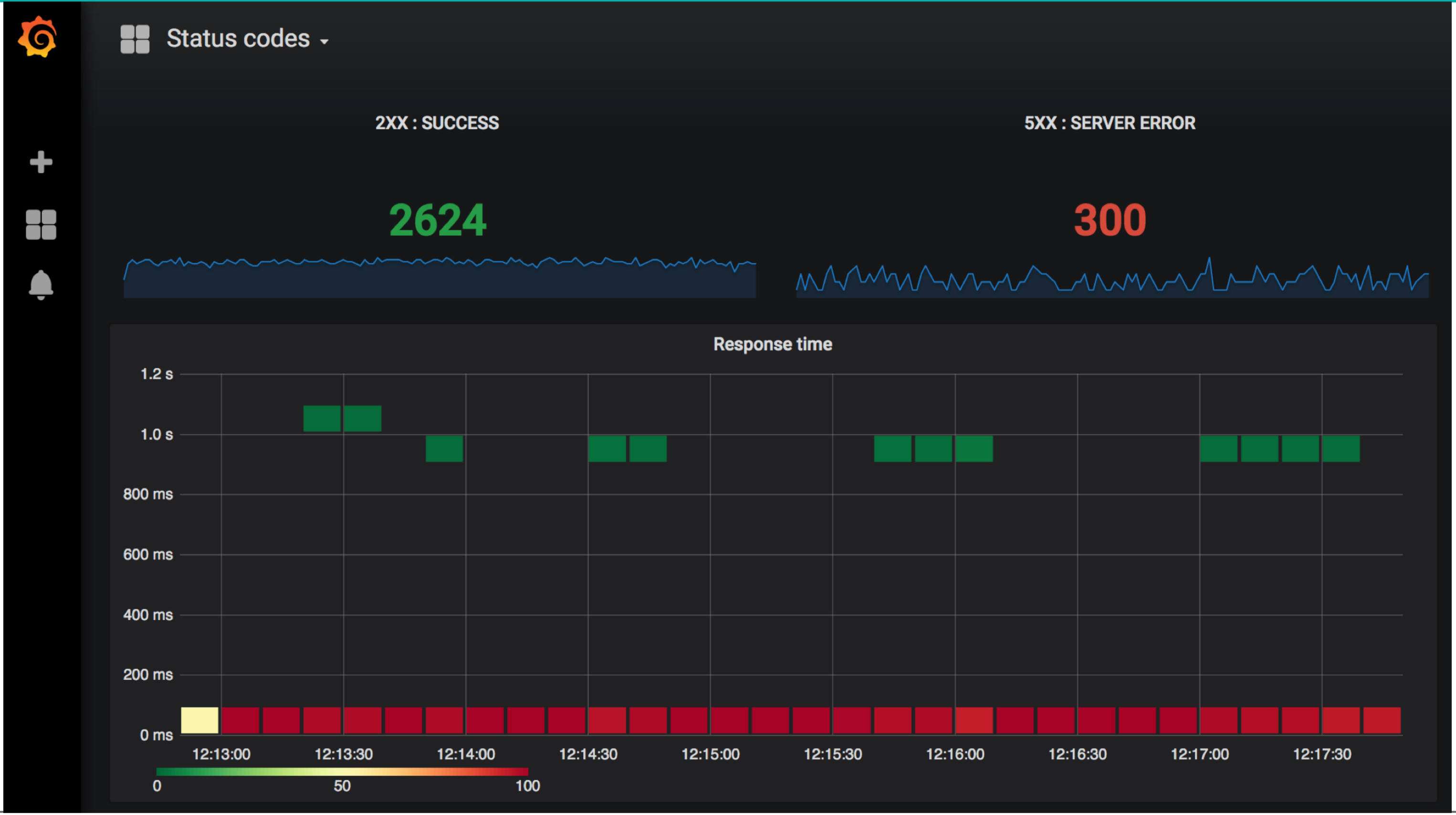
http_req_duration

http_req_duration (mean)	http_req_duration (max)	http_req_duration (med)	http_req_duration (min)	http_req_duration (p90)	http_req_duration (p95)
6.81 ms	1.00 s	1.16 ms	0.41 ms	2.07 ms	2.66 ms

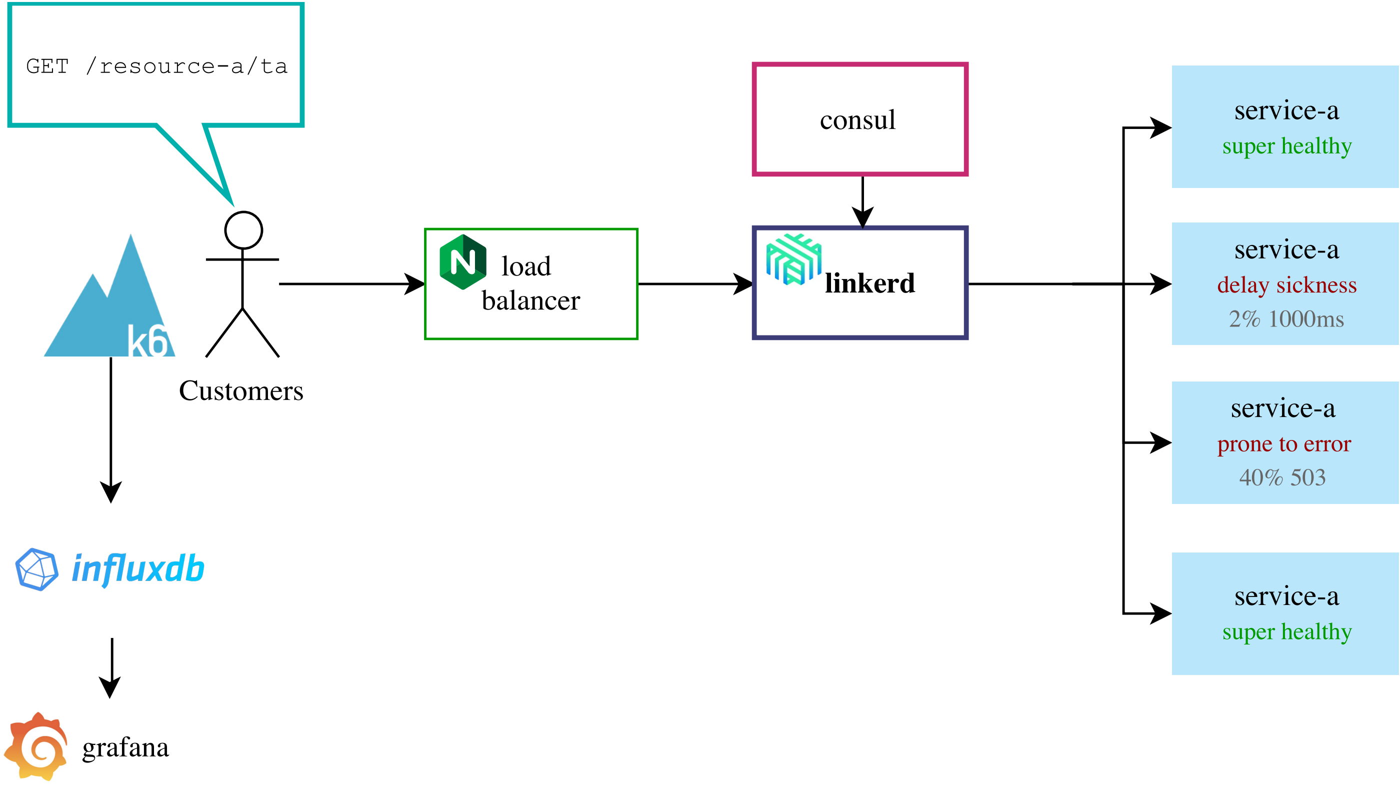
http_req_duration (over time)

http_req_duration (over time)

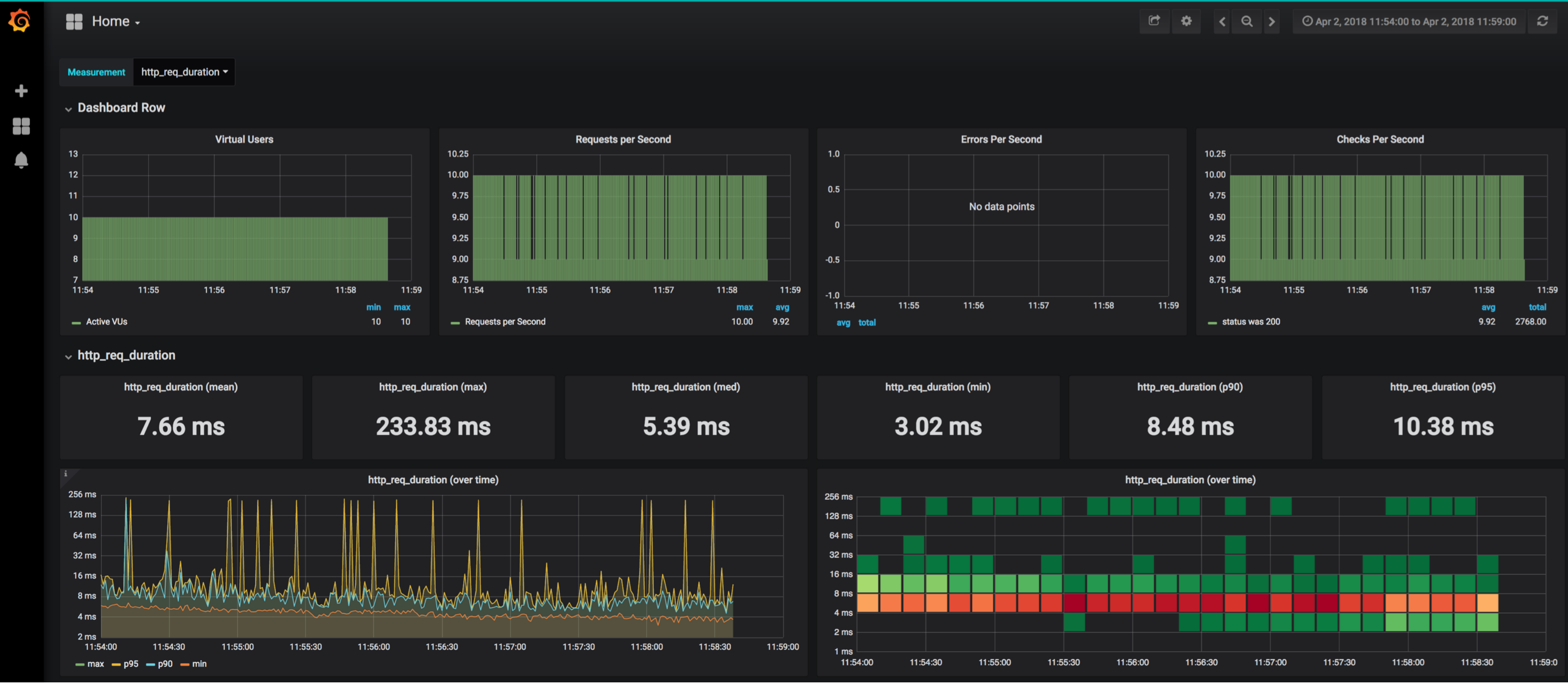
System 1: Response times



Demo: system-2



System 2: Response times



Home

Apr 2, 2018 11:54:00 to Apr 2, 2018 11:59:00

Measurement http_req_duration

Dashboard Row

Virtual Users

	min	max
Active VUs	10	10

Requests per Second

	max	avg
Requests per Second	10.00	9.92

Errors Per Second

No data points

avg total

Checks Per Second

	avg	total
status was 200	9.92	2768.00

http_req_duration

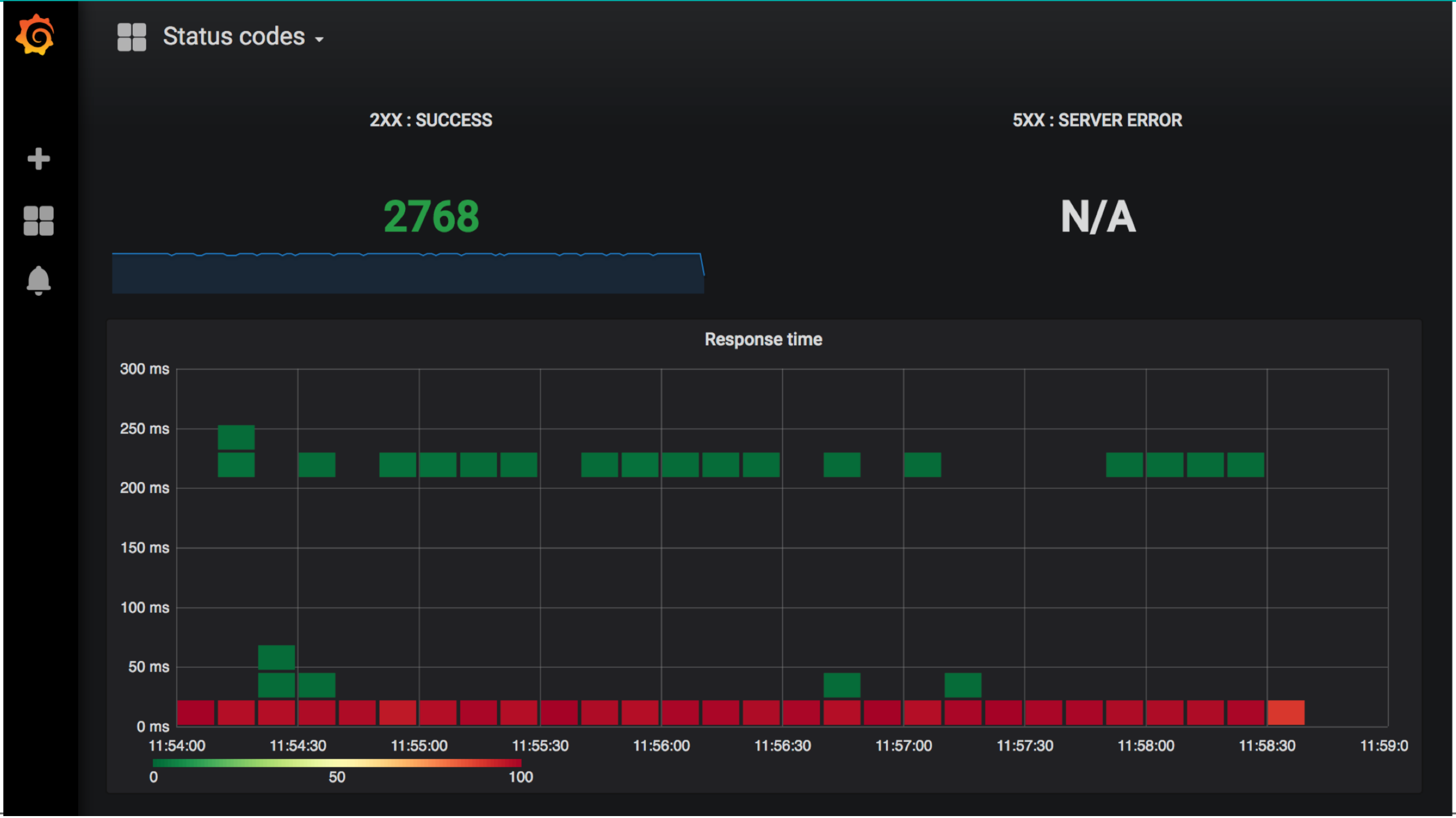
http_req_duration (mean)	http_req_duration (max)	http_req_duration (med)	http_req_duration (min)	http_req_duration (p90)	http_req_duration (p95)
7.66 ms	233.83 ms	5.39 ms	3.02 ms	8.48 ms	10.38 ms

http_req_duration (over time)

max p95 p90 min

http_req_duration (over time)

System 2: Response times



Status codes

2XX : SUCCESS

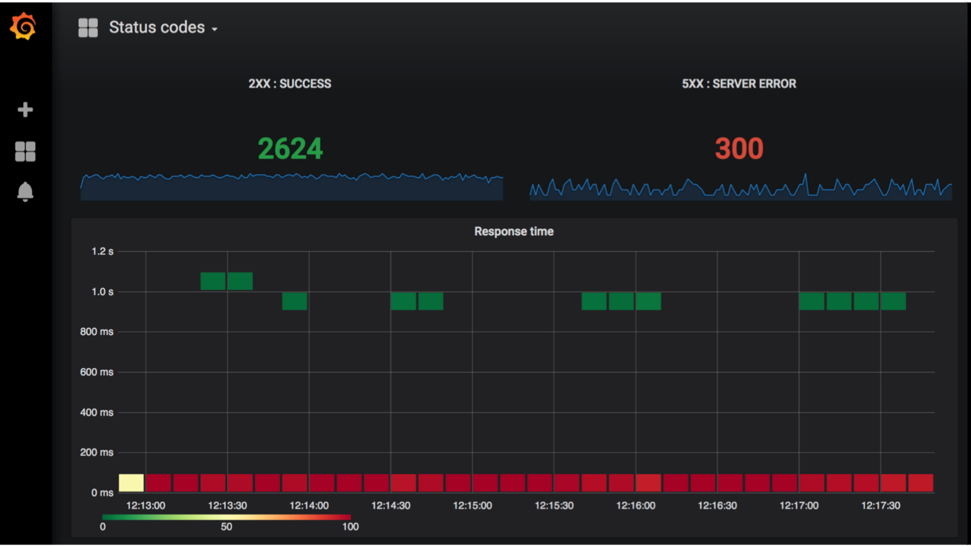
2768

5XX : SERVER ERROR

N/A

How do the 2 demos compare?

System 1

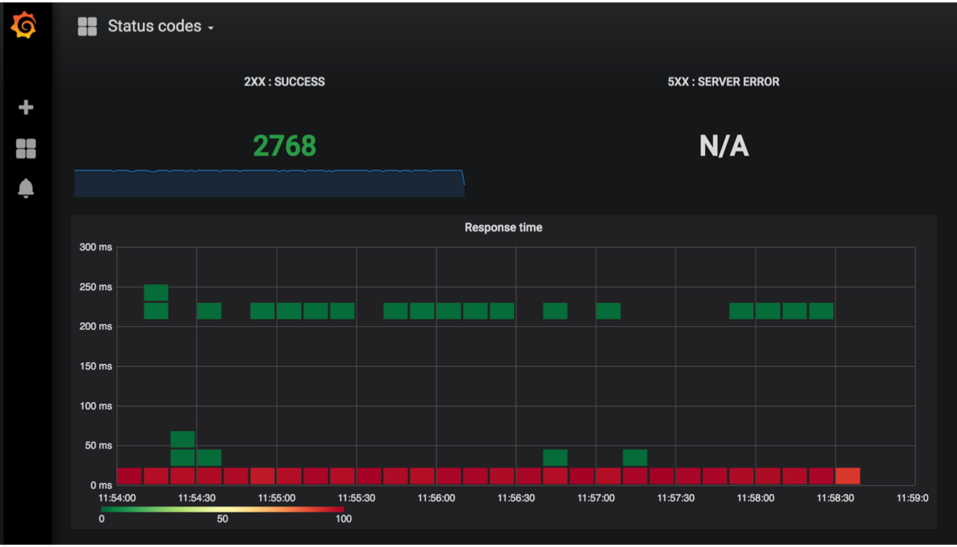


Status codes

2XX : SUCCESS
2624

5XX : SERVER ERROR
300

System 2

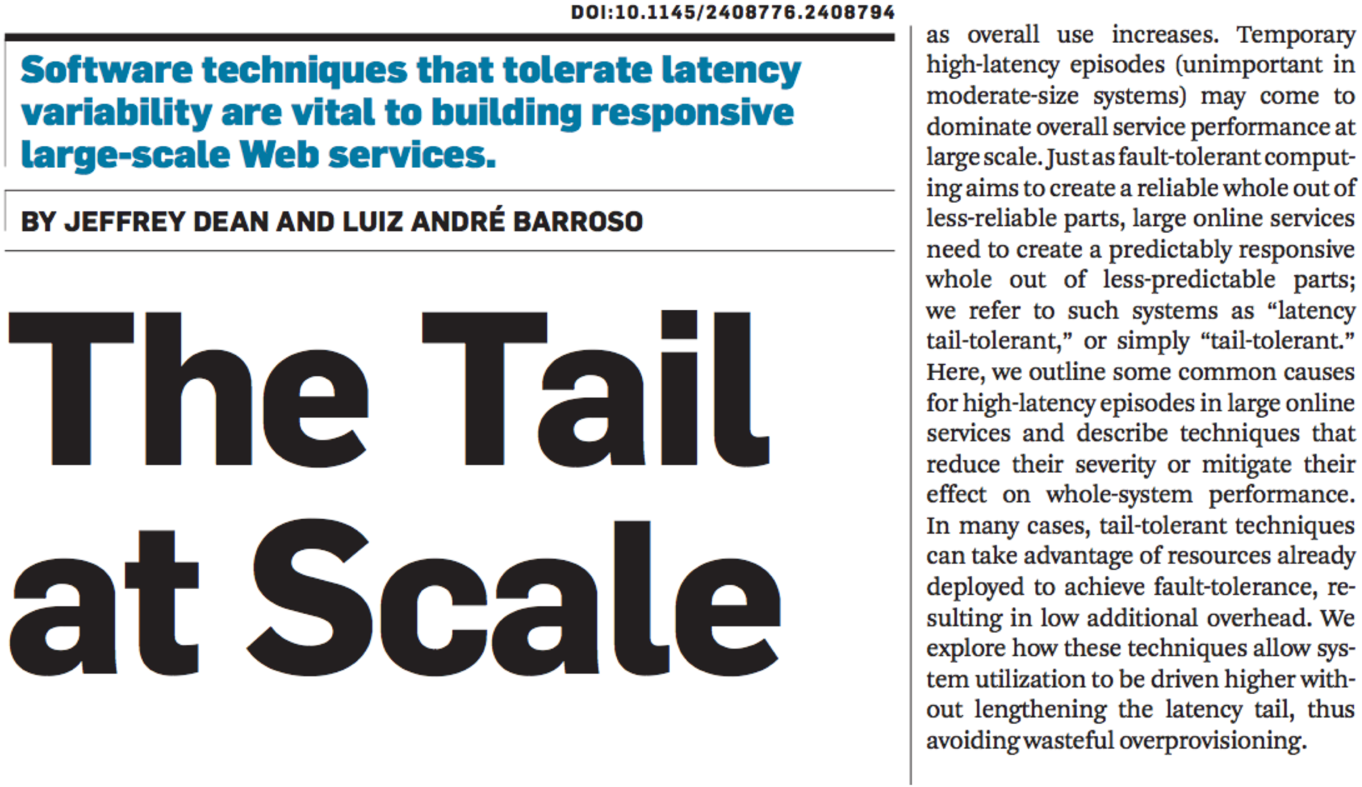


Status codes

2XX : SUCCESS
2768

5XX : SERVER ERROR
N/A

Further improving tail latency



DOI:10.1145/2408776.2408794

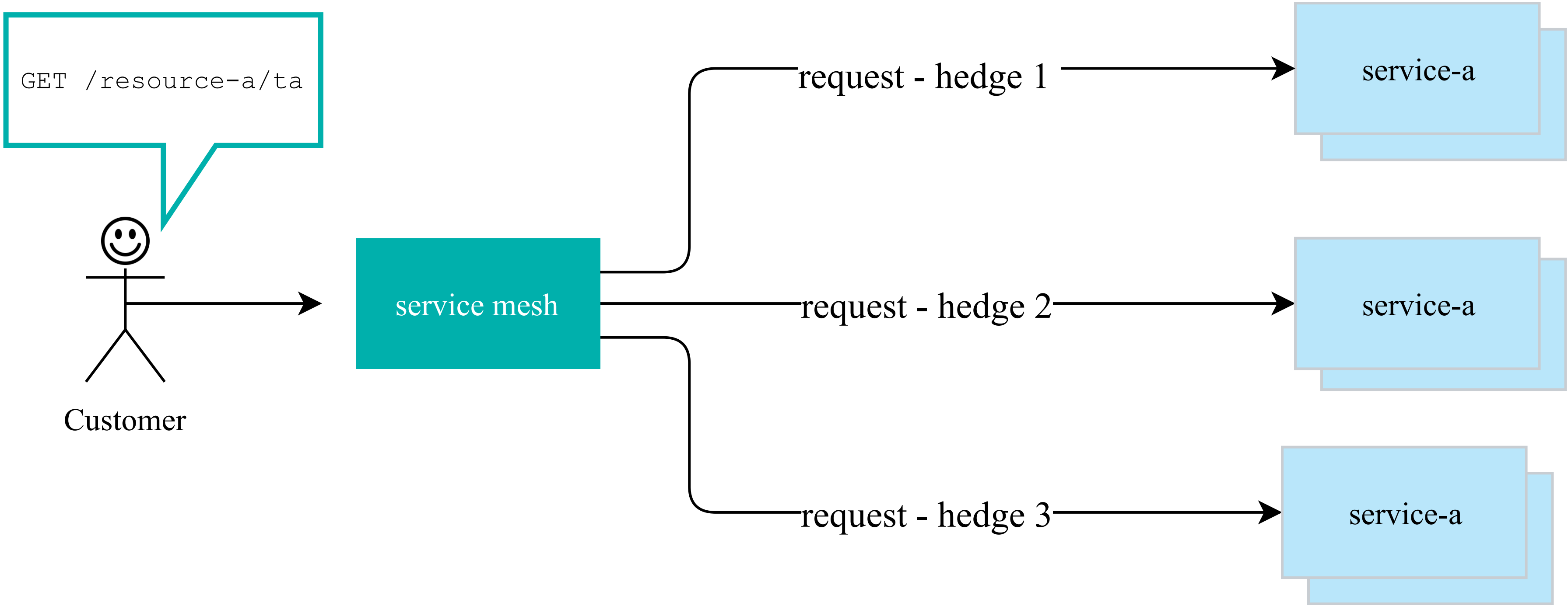
Software techniques that tolerate latency variability are vital to building responsive large-scale Web services.

BY JEFFREY DEAN AND LUIZ ANDRÉ BARROSO

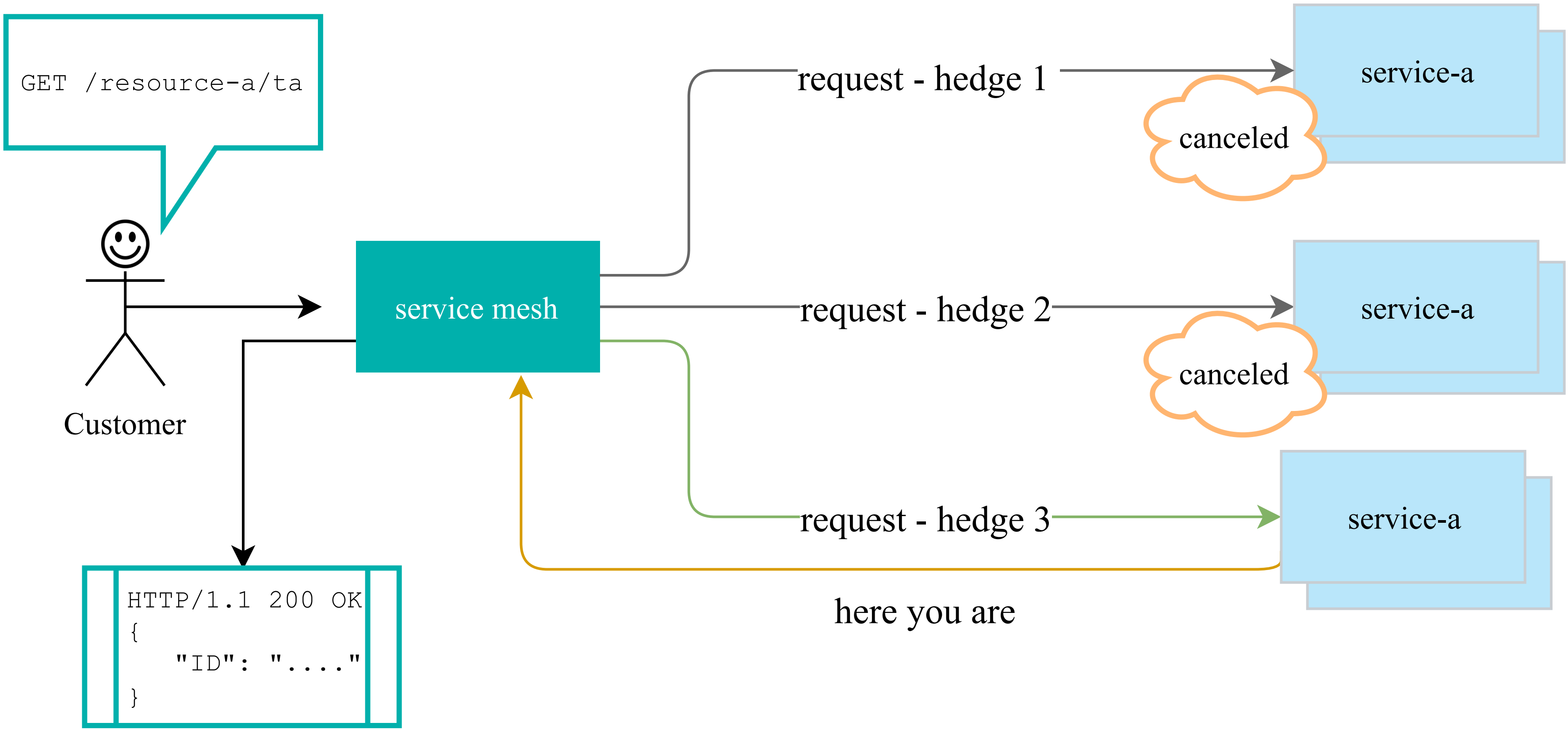
The Tail at Scale

as overall use increases. Temporary high-latency episodes (unimportant in moderate-size systems) may come to dominate overall service performance at large scale. Just as fault-tolerant computing aims to create a reliable whole out of less-reliable parts, large online services need to create a predictably responsive whole out of less-predictable parts; we refer to such systems as "latency tail-tolerant," or simply "tail-tolerant." Here, we outline some common causes for high-latency episodes in large online services and describe techniques that reduce their severity or mitigate their effect on whole-system performance. In many cases, tail-tolerant techniques can take advantage of resources already deployed to achieve fault-tolerance, resulting in low additional overhead. We explore how these techniques allow system utilization to be driven higher without lengthening the latency tail, thus avoiding wasteful overprovisioning.

Hedge requests



Hedge requests



```
GET /resource-a/ta
```

```
HTTP/1.1 200 OK
{
    "ID": "...."
}
```

Wrap it up

We have used Linkerd in this demo to

- Reduce error rates using retries
- Minimize tail latencies using timeouts

Additionally at Form3 we use it to:

- Increase security &
- Help diagnose problems using tracing

Thank you

service mesh yay!

WE'RE HIRING!

https://github.com/ewilde/kubecon ← this talk

https://form3.tech

and/or

working