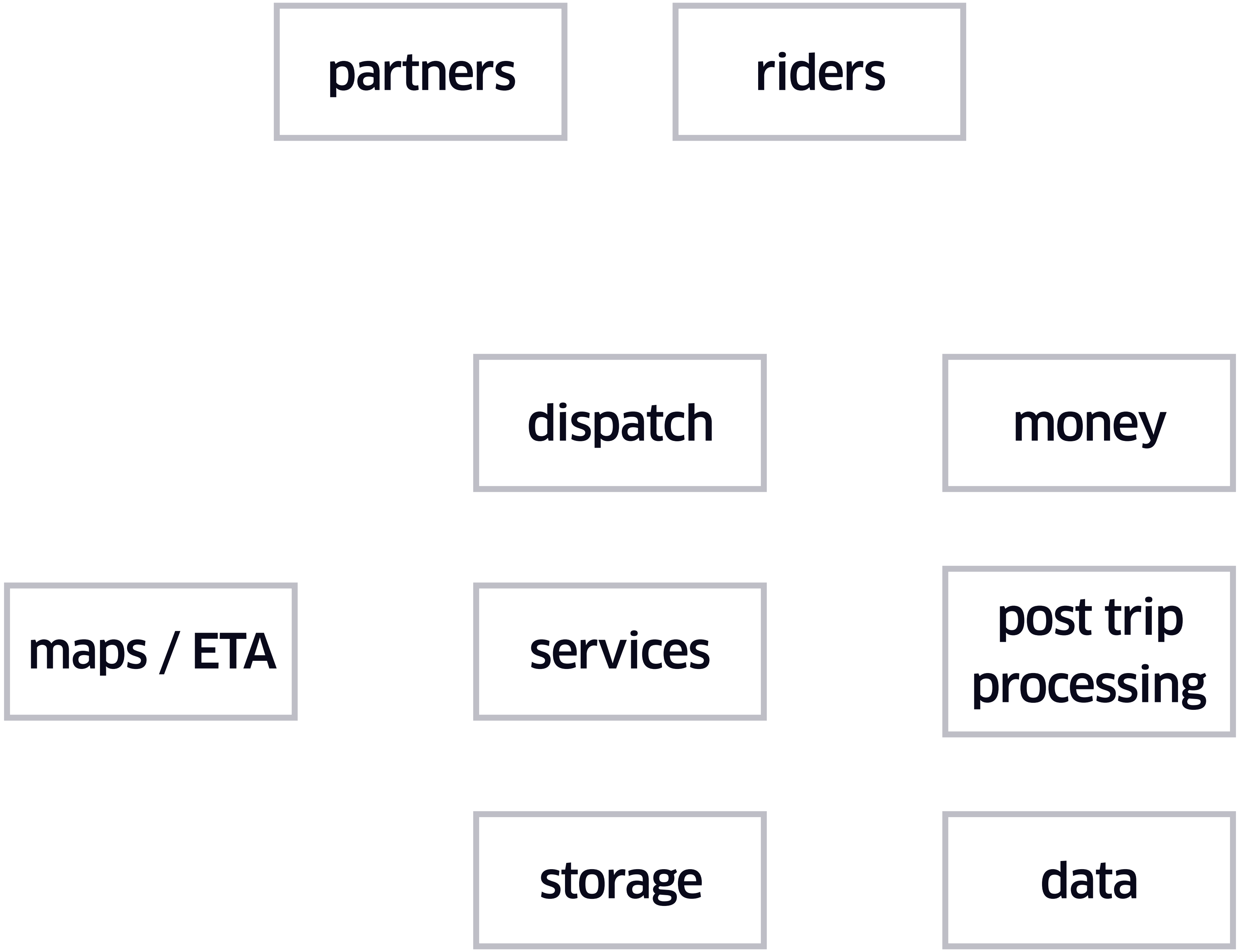
DISTRIBUTED SYSTEMS AT UBER

MATT RANNEY

UBER

OPEN SOURCE



partners

riders

dispatch

money

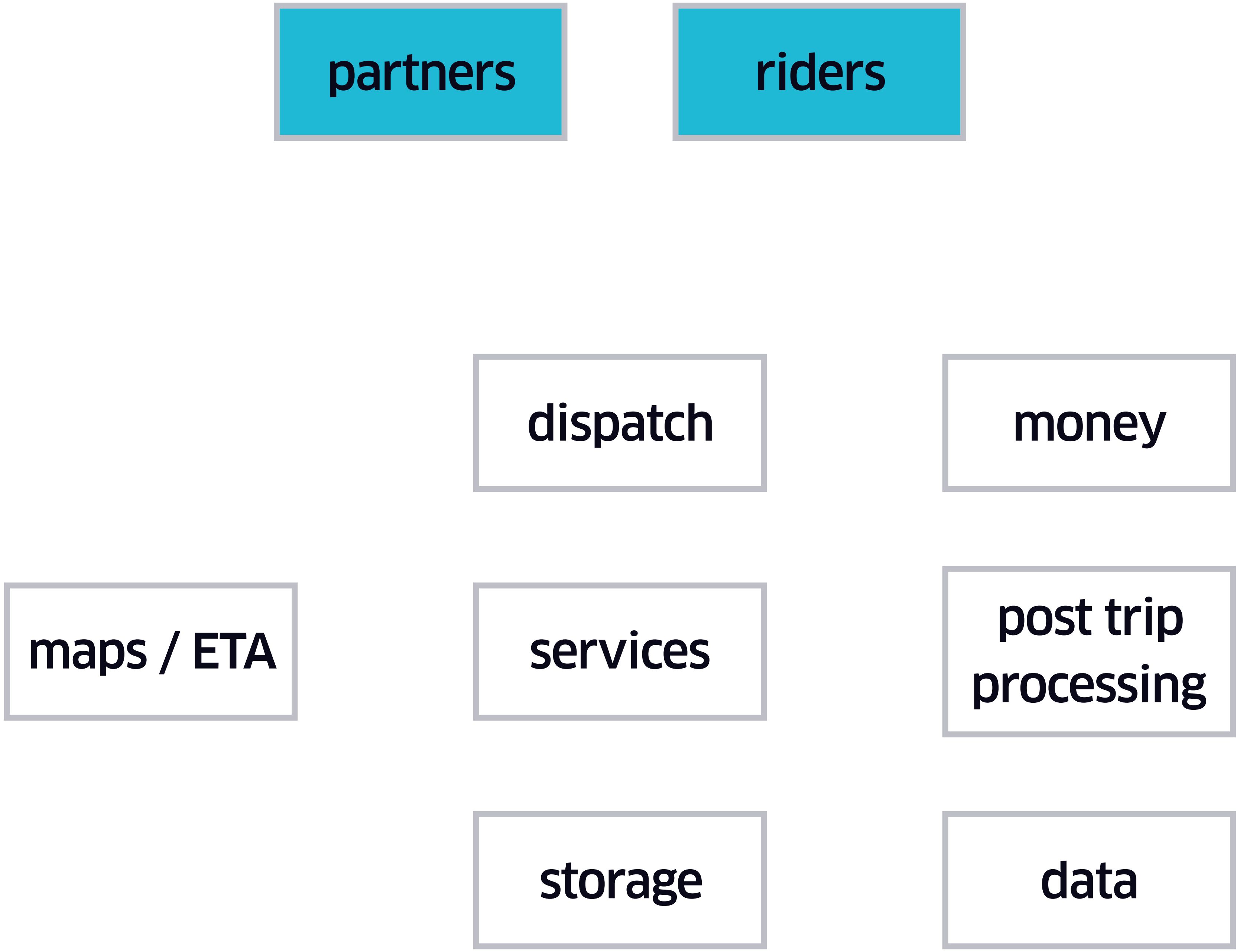
maps / ETA

services

post trip processing

storage

data



partners

riders

dispatch

money

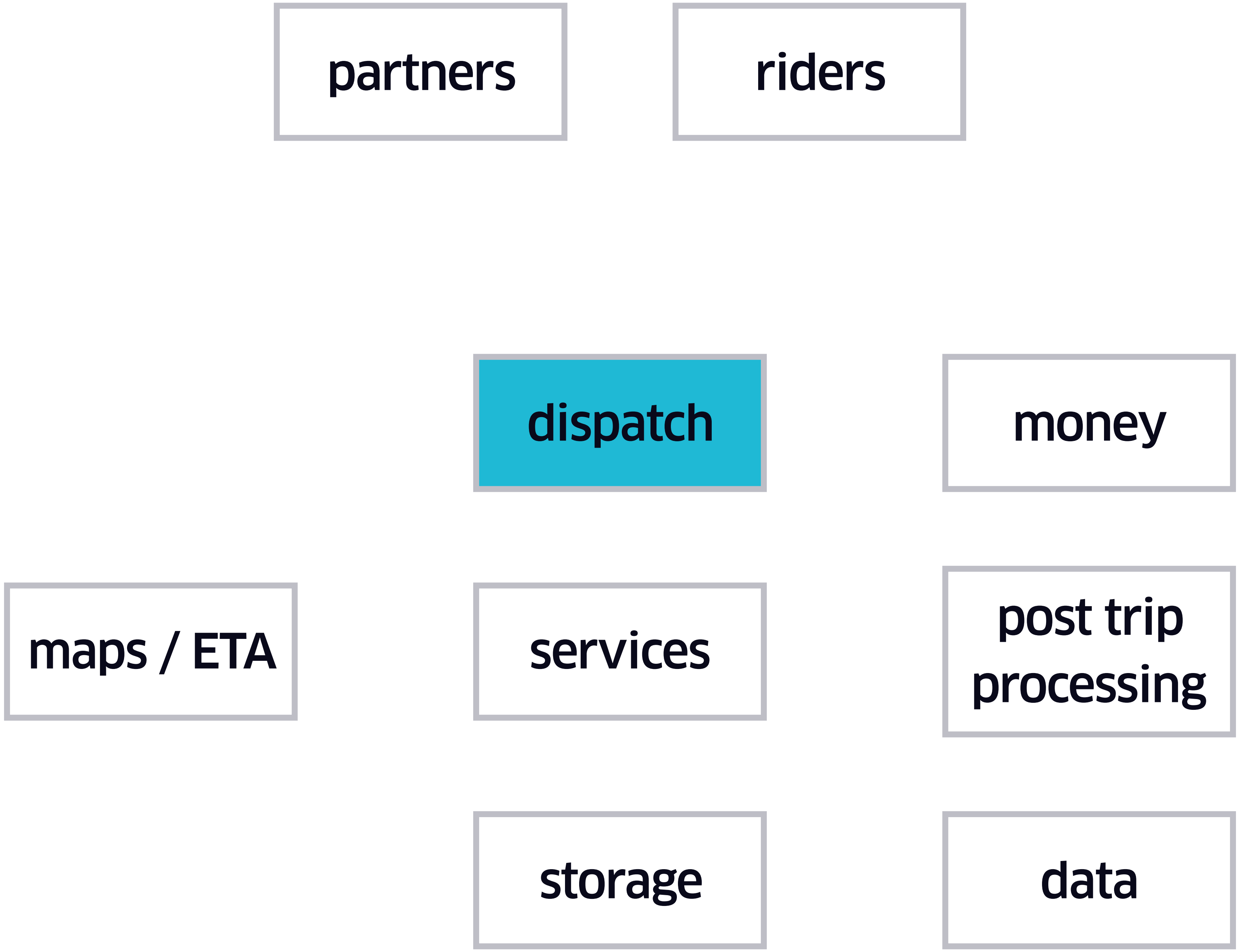
maps / ETA

services

post trip processing

storage

data



partners

riders

dispatch

money

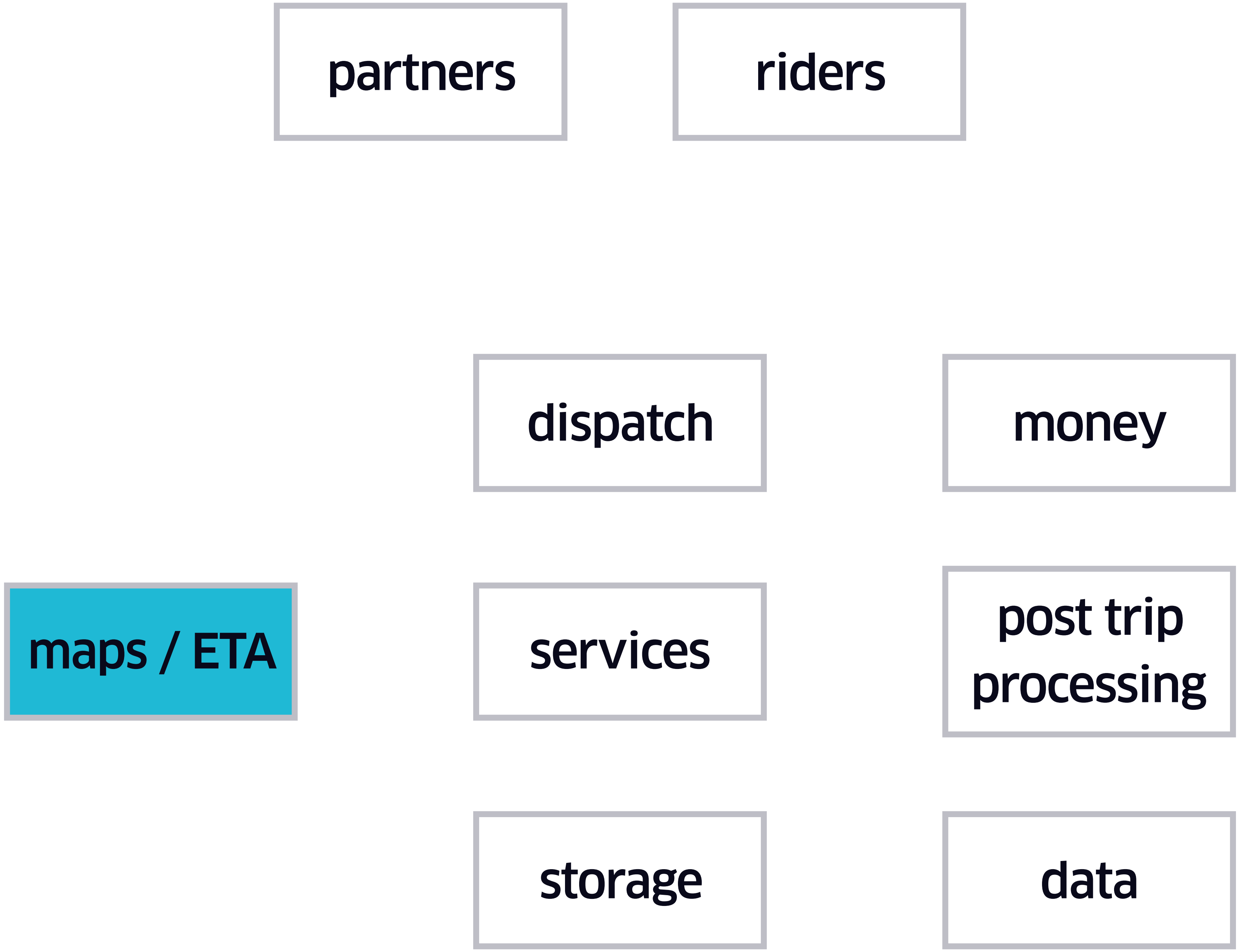
maps / ETA

services

post trip processing

storage

data



partners

riders

dispatch

money

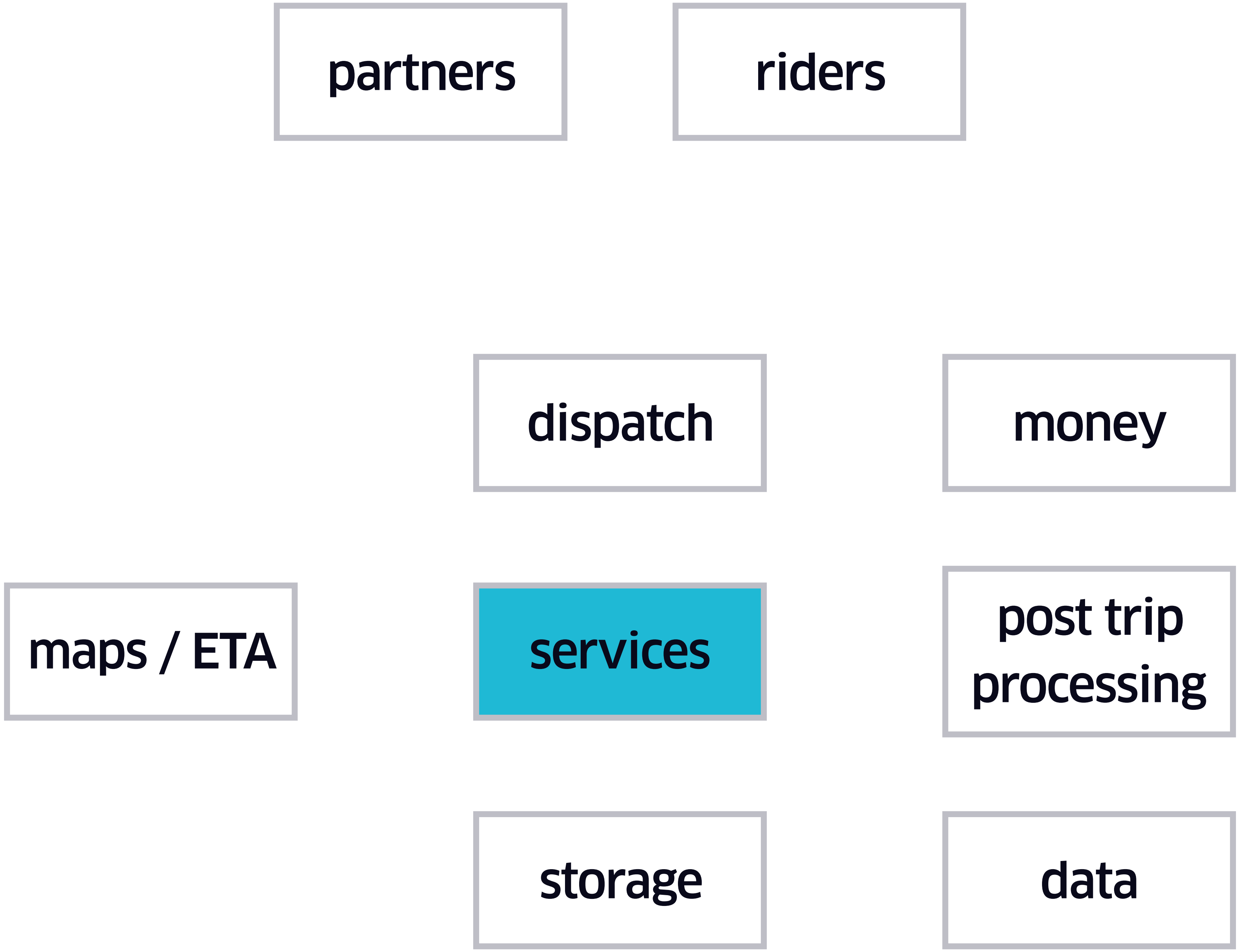
maps / ETA

services

post trip processing

storage

data



partners

riders

dispatch

money

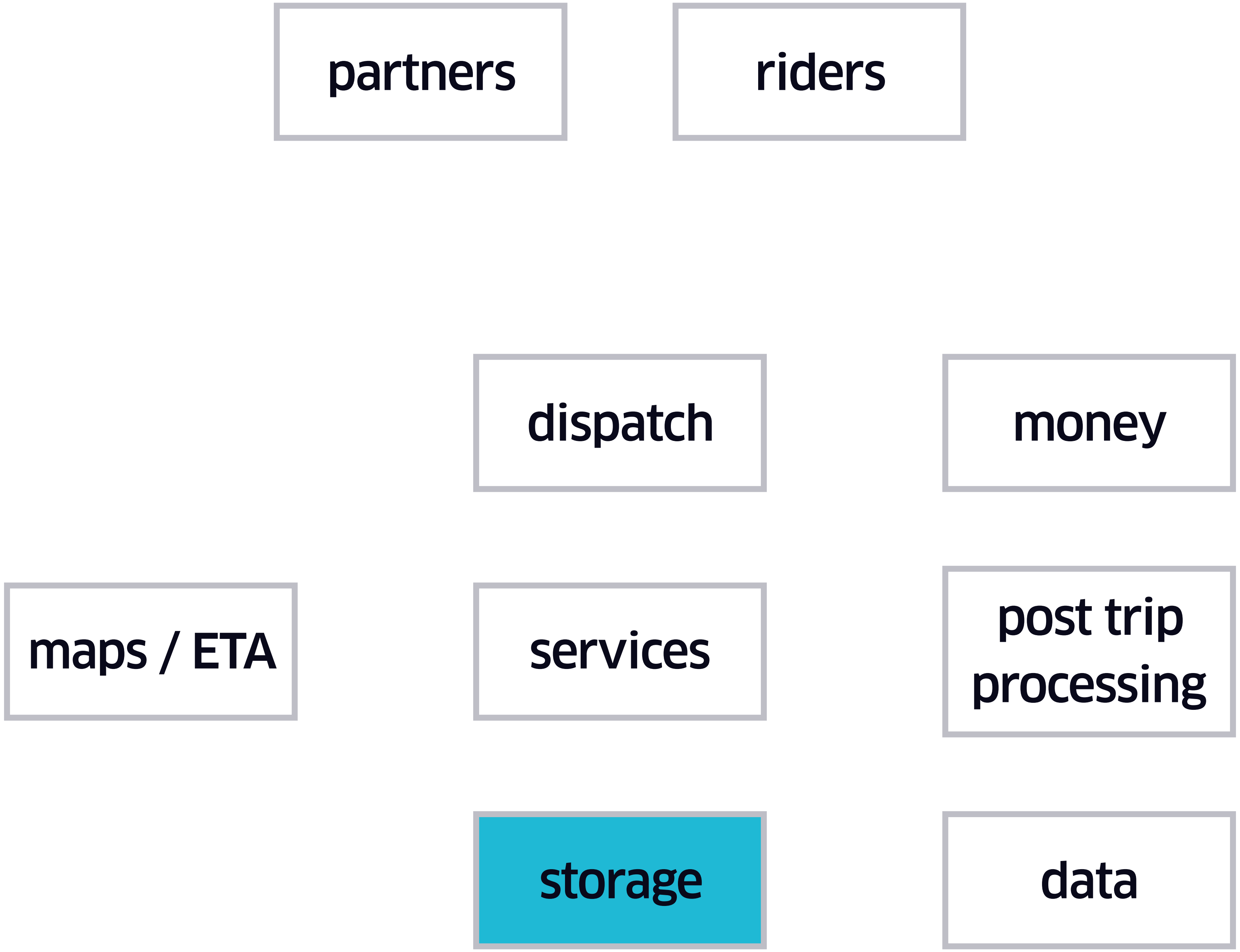
maps / ETA

services

post trip processing

storage

data



partners

riders

dispatch

money

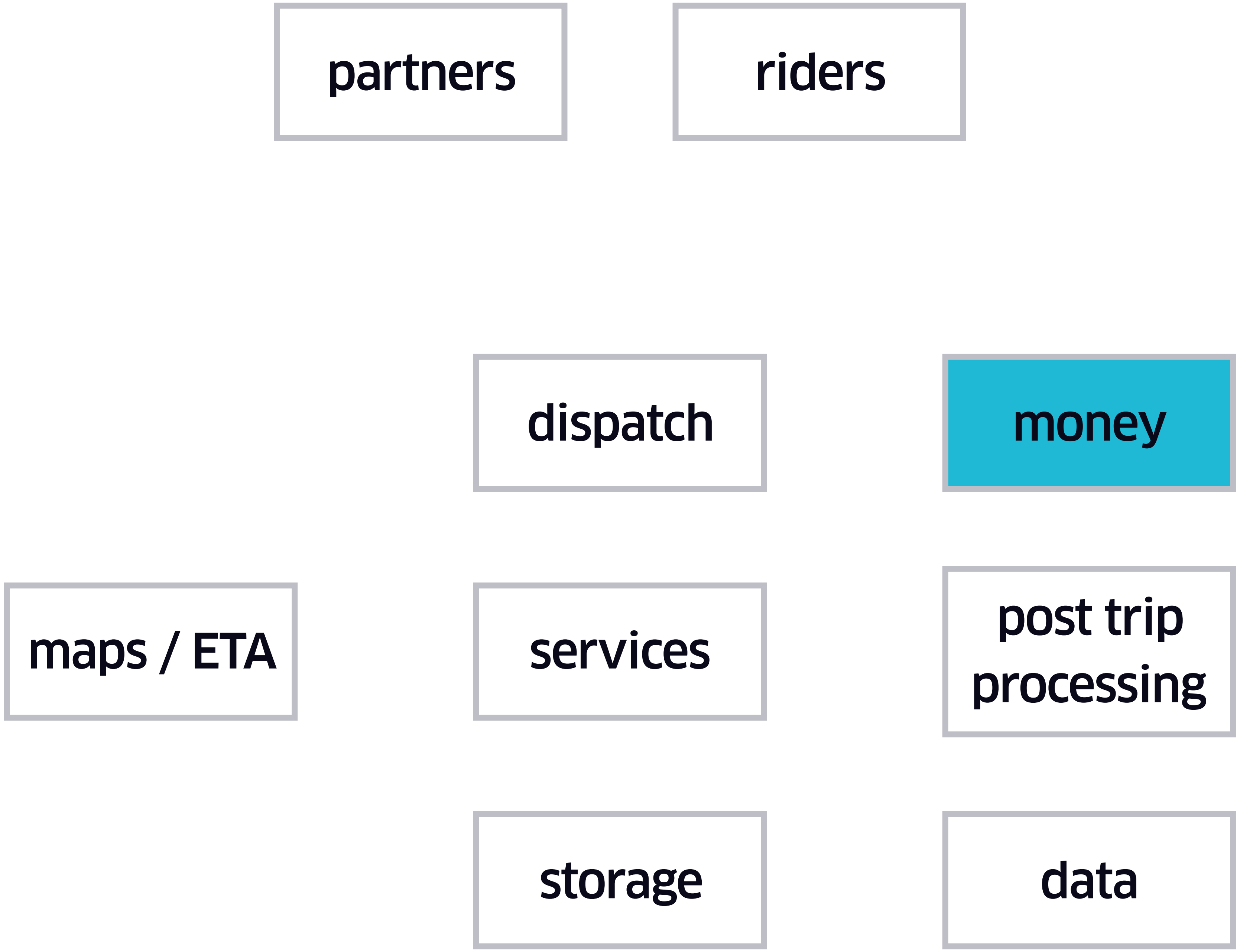
maps / ETA

services

post trip processing

storage

data



partners

riders

dispatch

money

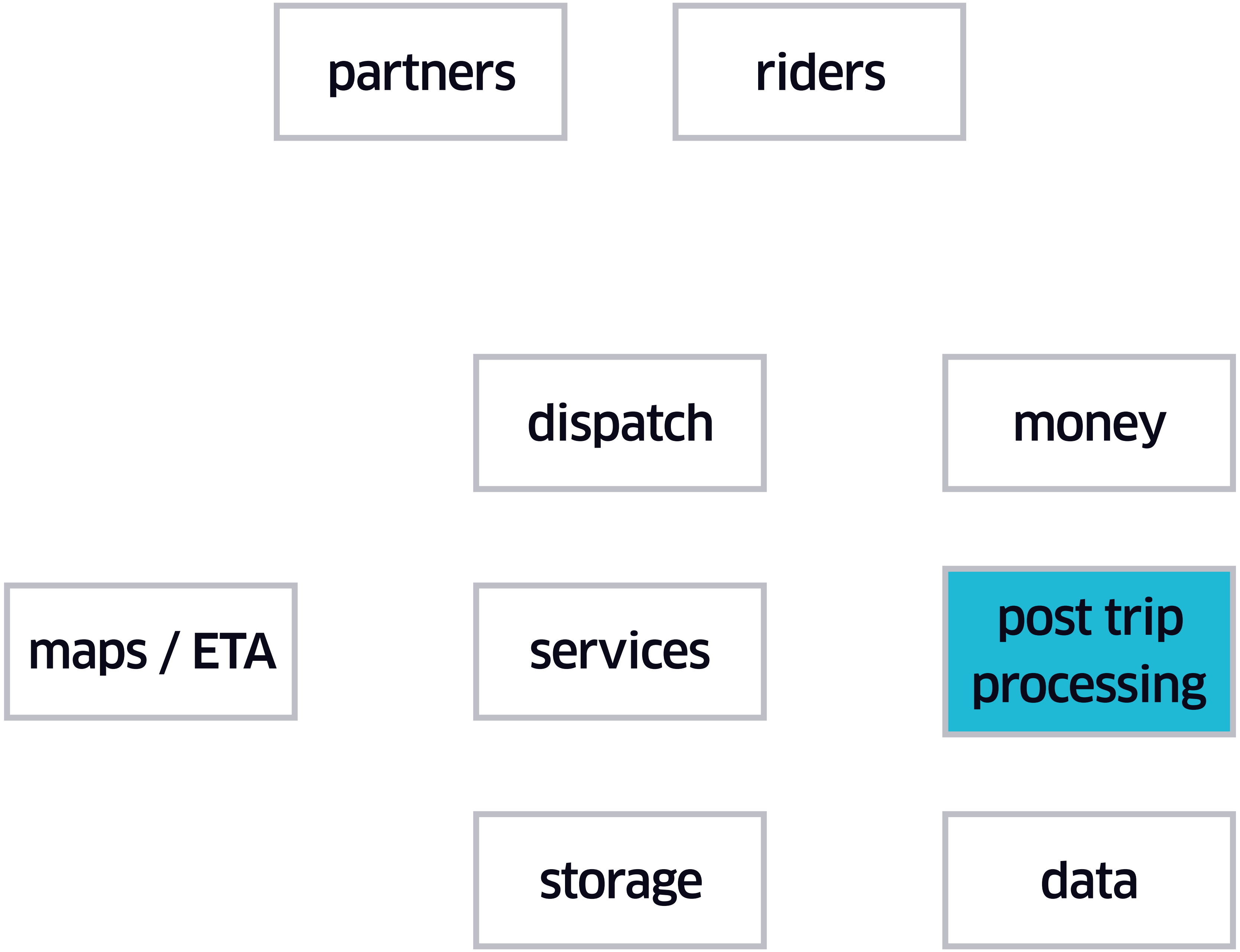
maps / ETA

services

post trip processing

storage

data



partners

riders

dispatch

money

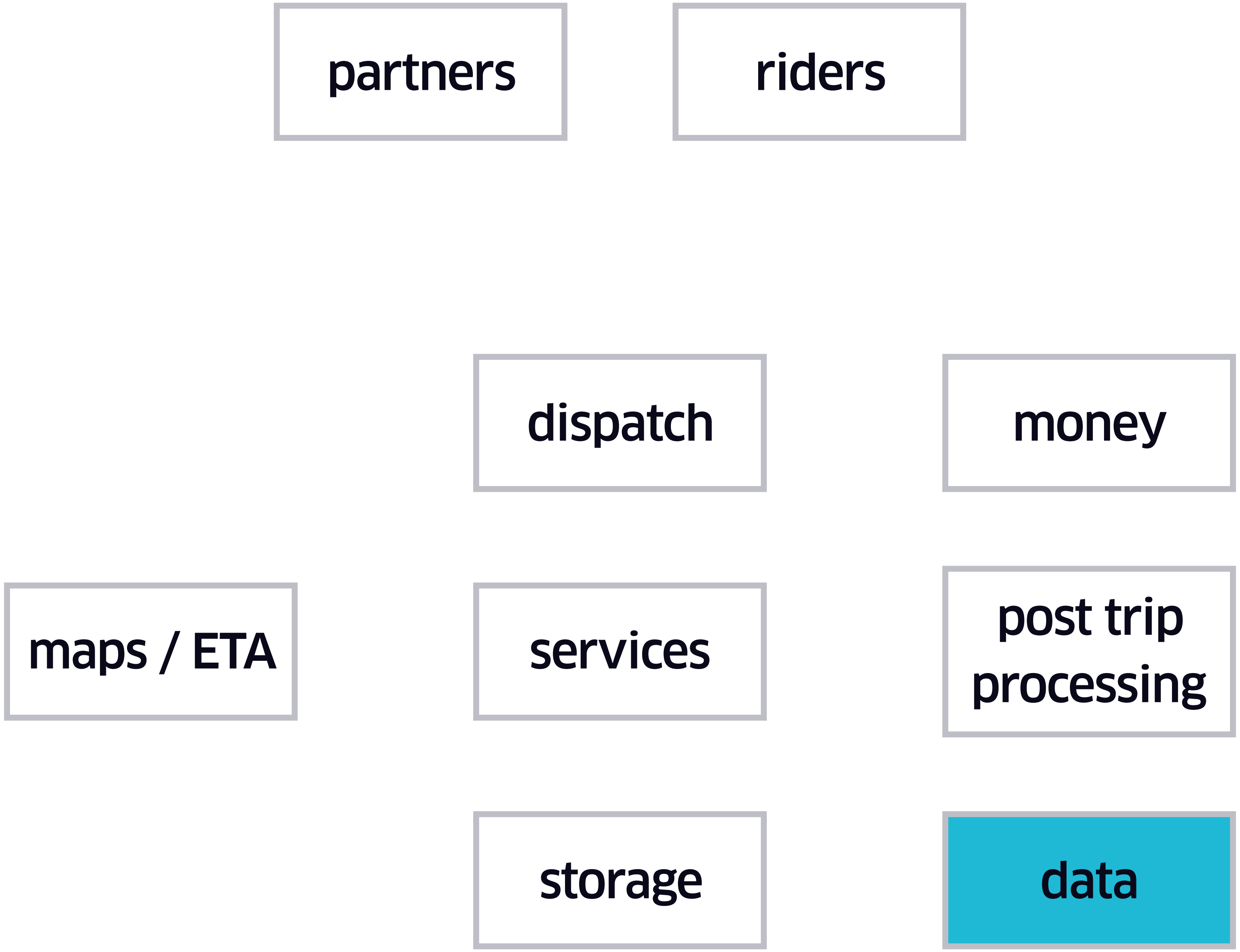
maps / ETA

services

post trip processing

storage

data



partners

riders

dispatch

money

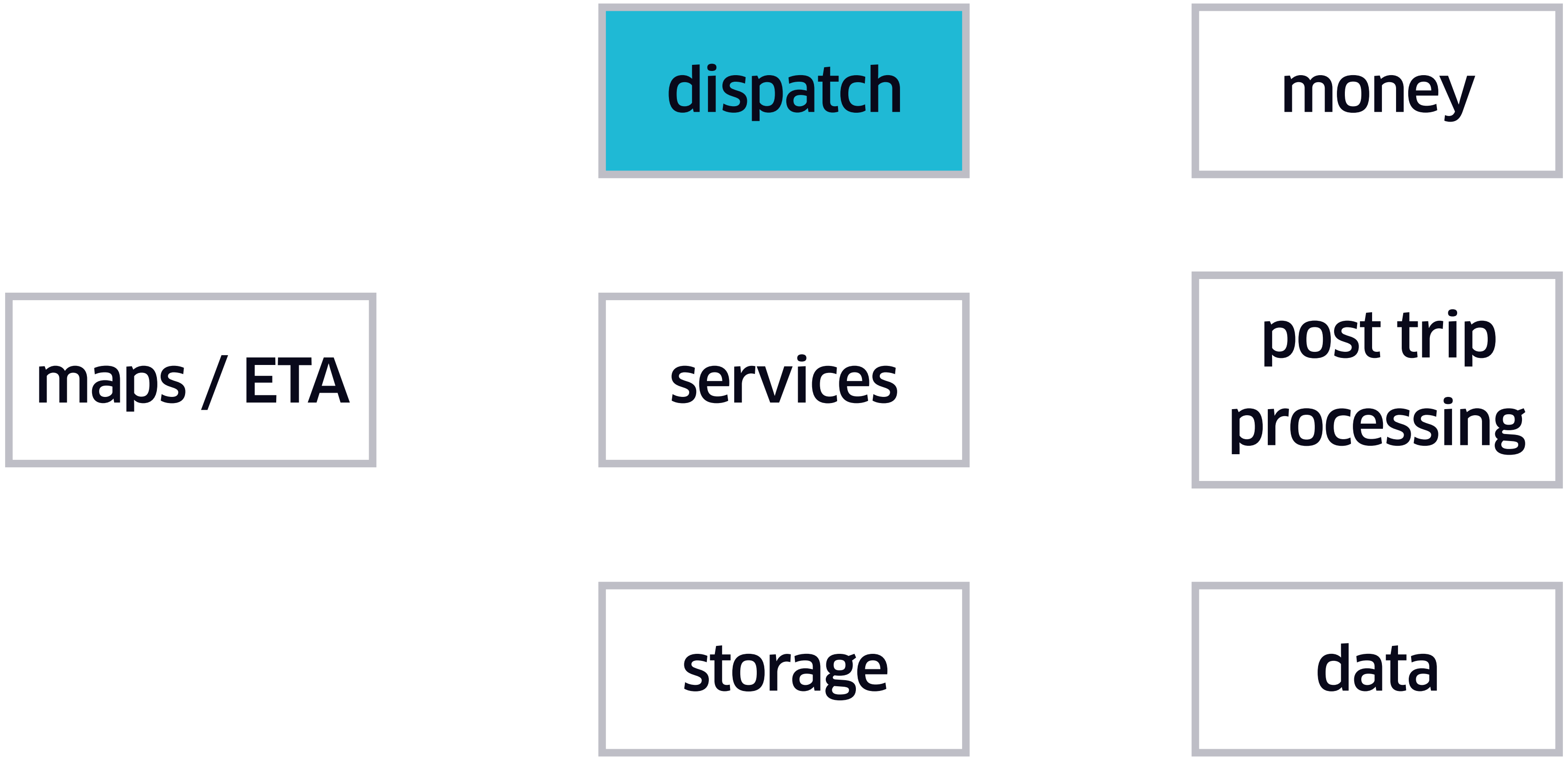
maps / ETA

services

post trip processing

storage

data



dispatch

money

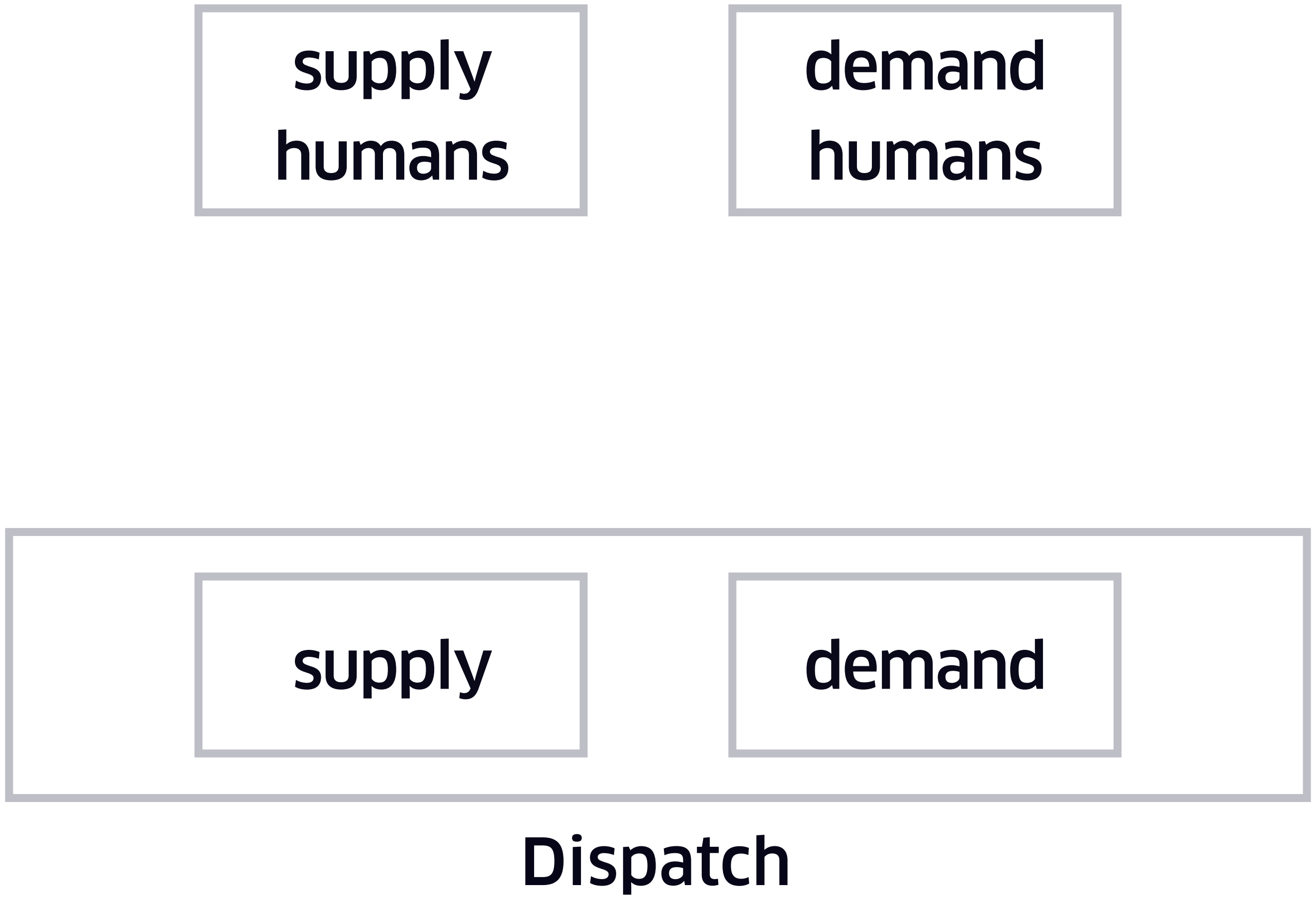
maps / ETA

services

post trip processing

storage

data



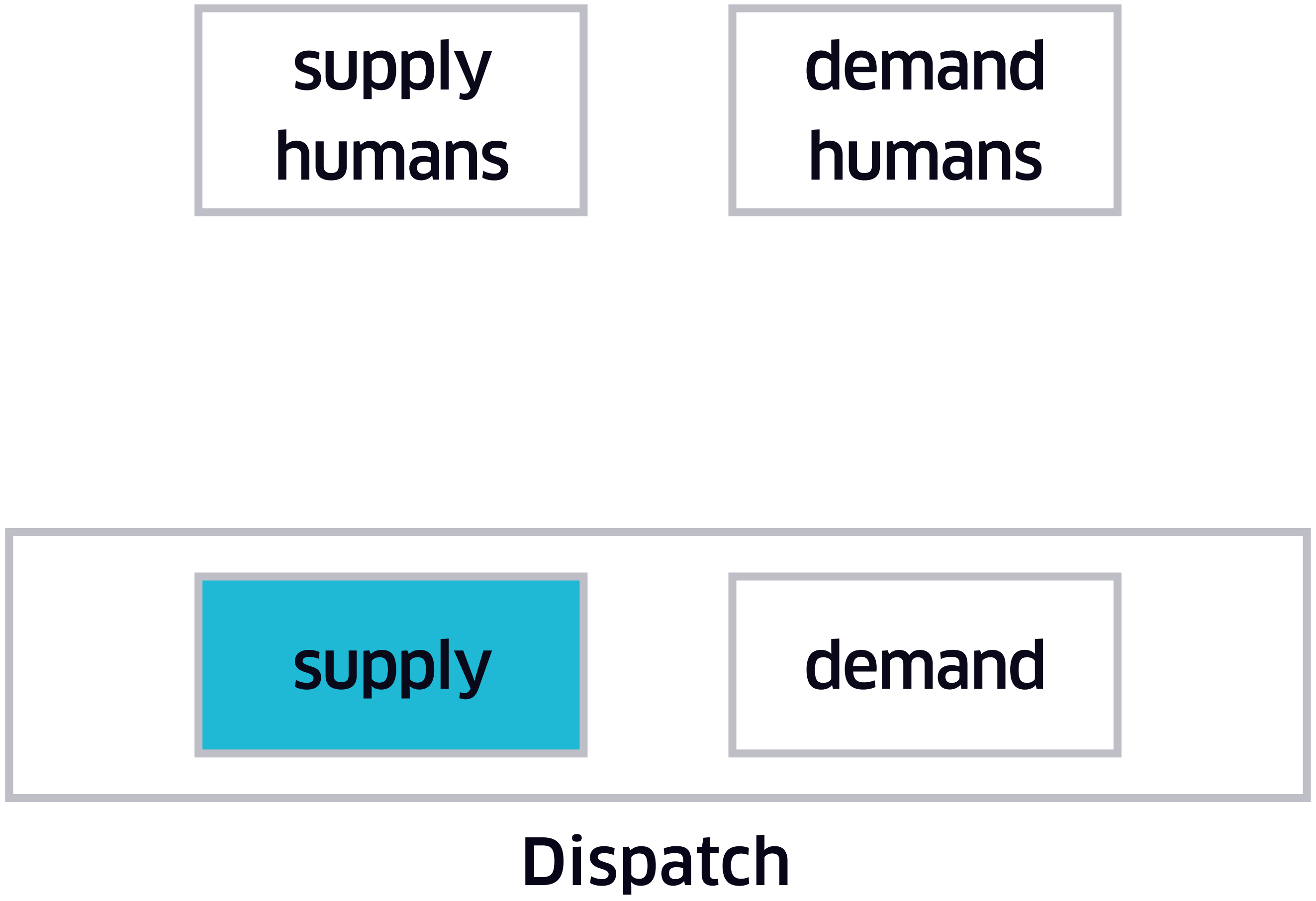
supply humans

demand humans

supply

demand

Dispatch



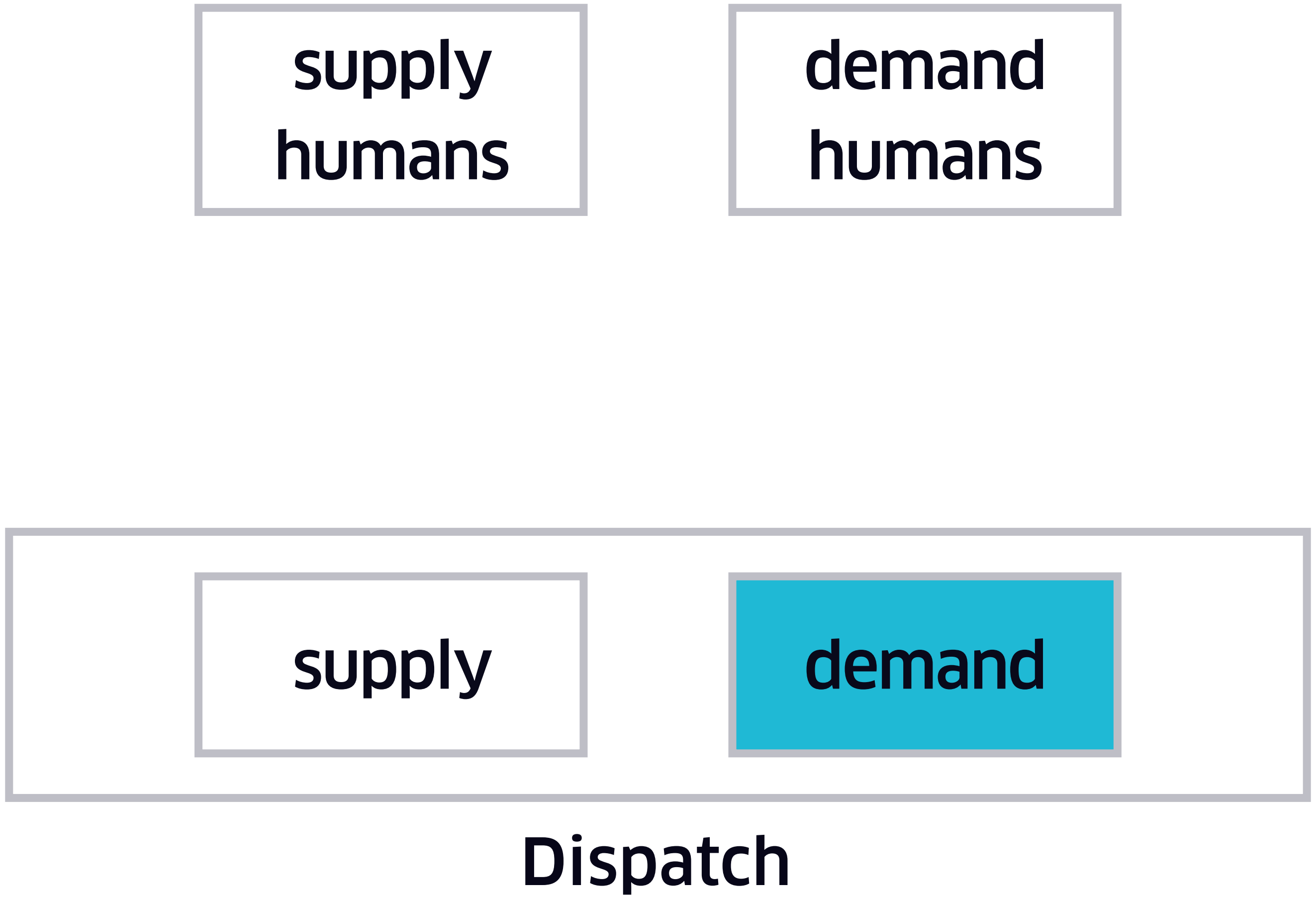
supply humans

demand humans

supply

demand

Dispatch



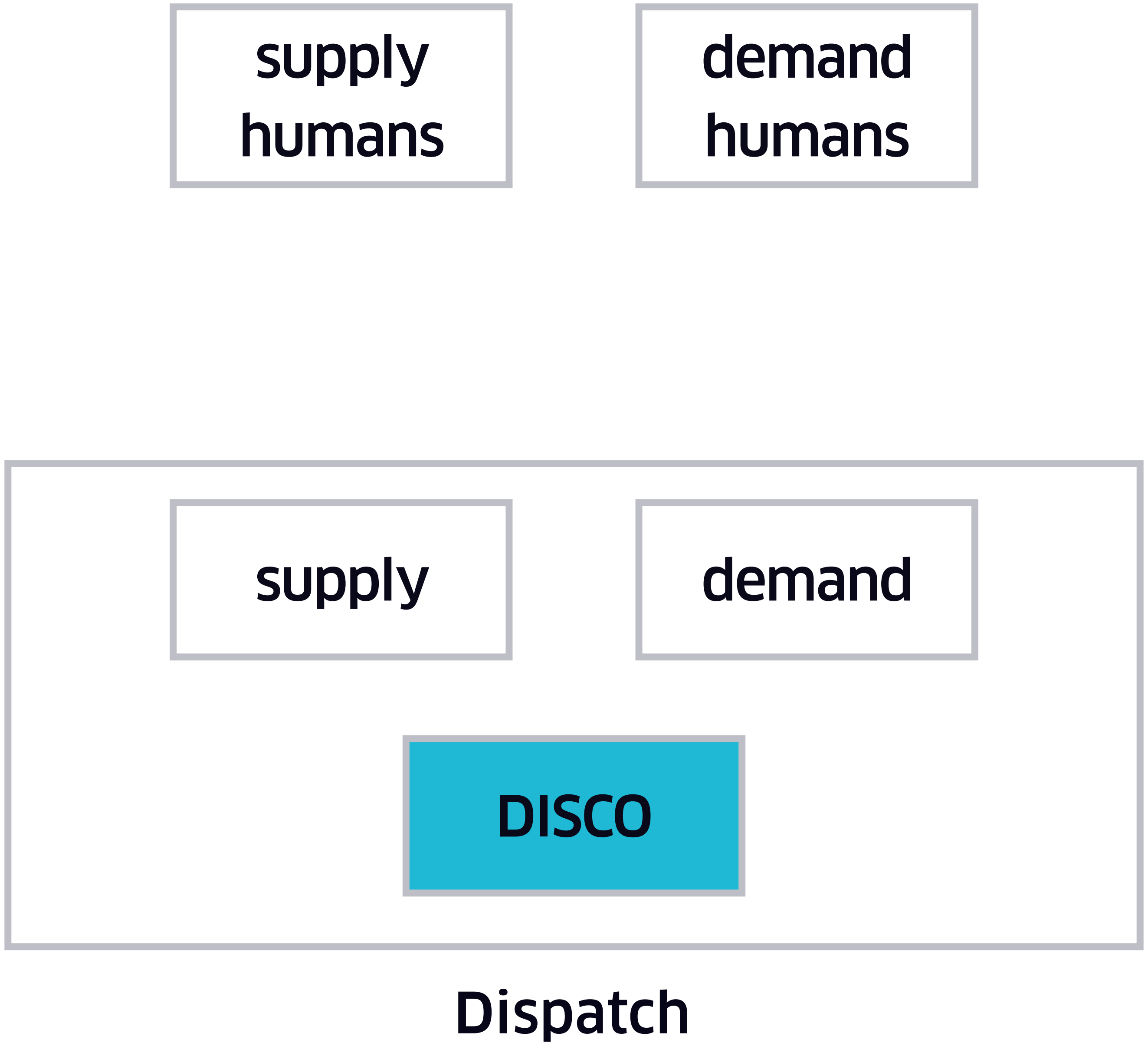
supply humans

demand humans

supply

demand

Dispatch



supply humans

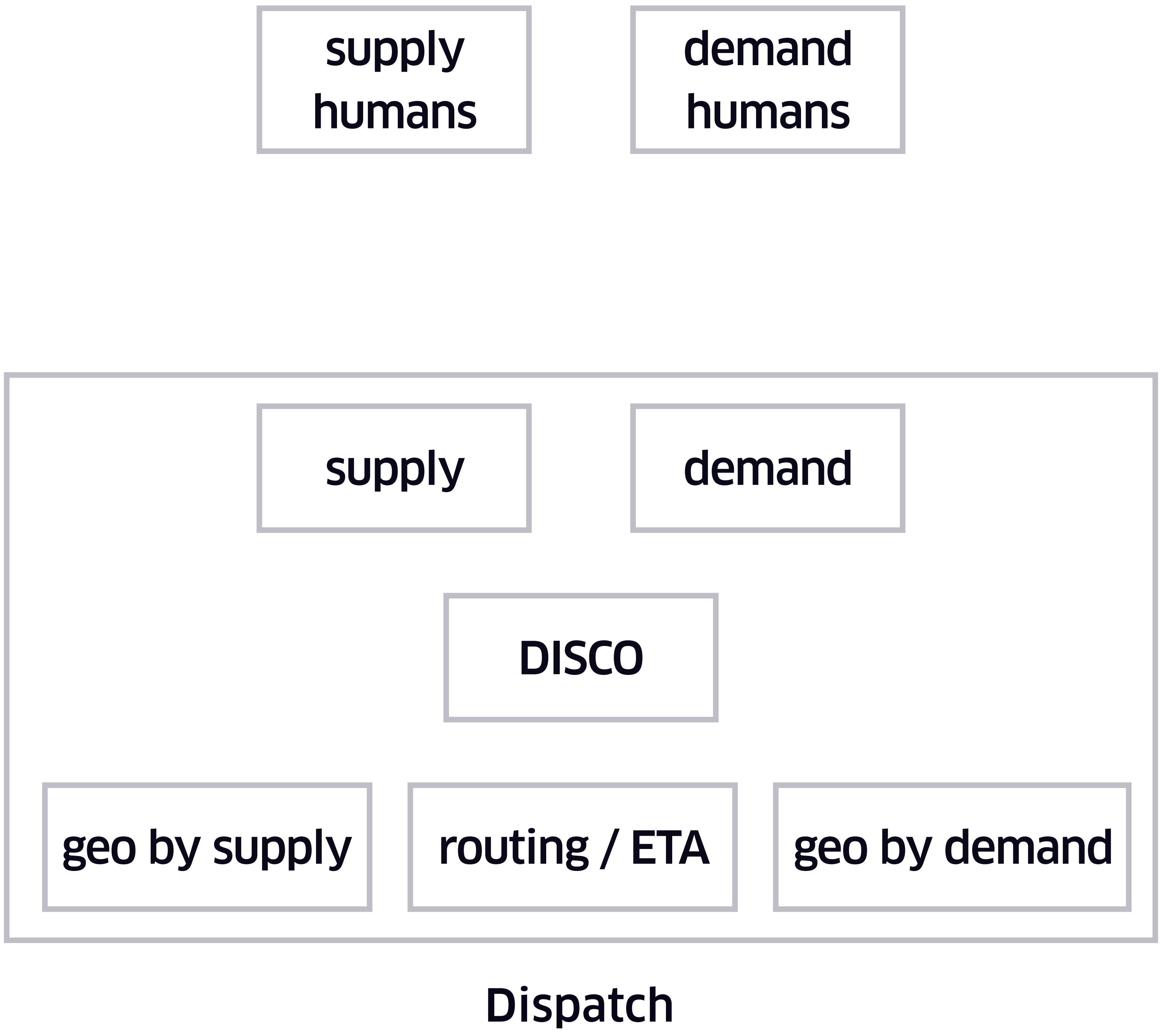
demand humans

supply

demand

DISCO

Dispatch



supply humans

demand humans

supply

demand

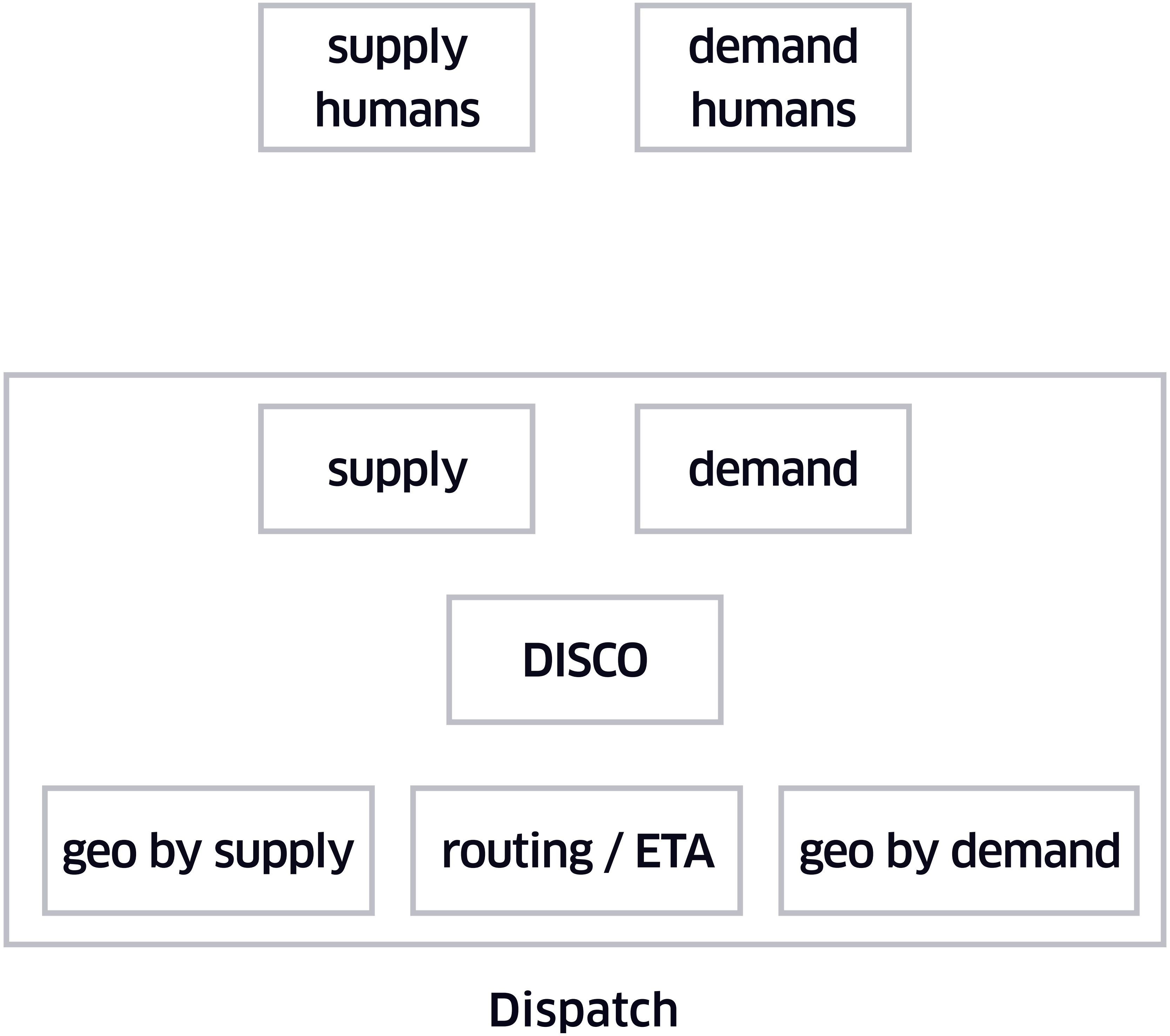
DISCO

geo by supply

routing / ETA

geo by demand

Dispatch



supply humans

demand humans

supply

demand

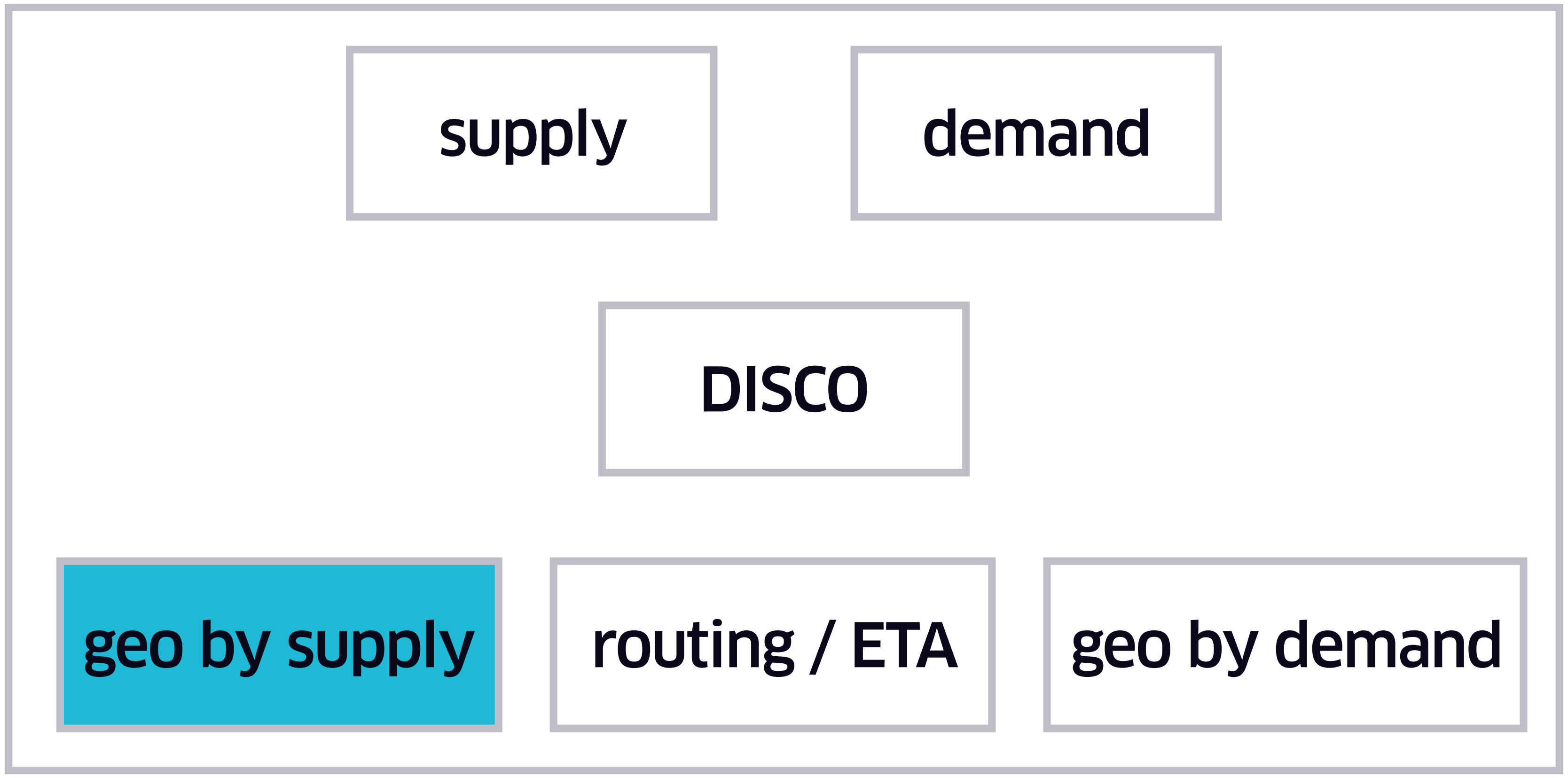
DISCO

geo by supply

routing / ETA

geo by demand

Dispatch



supply

demand

DISCO

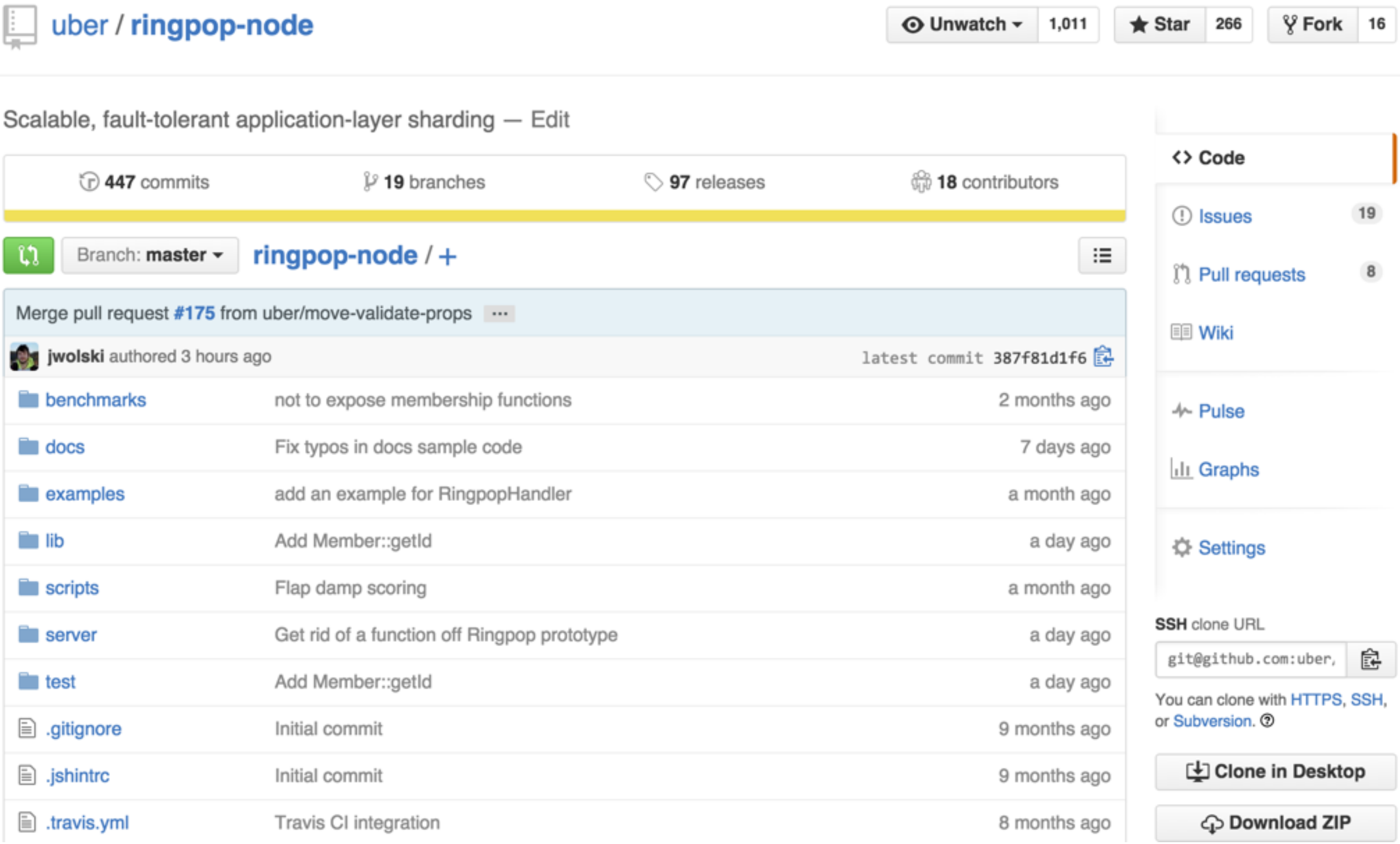
geo by supply

routing / ETA

geo by demand

Dispatch

SCALING NODE.JS



uber / ringpop-node

Scalable, fault-tolerant application-layer sharding — Edit

447 commits 19 branches 97 releases 18 contributors

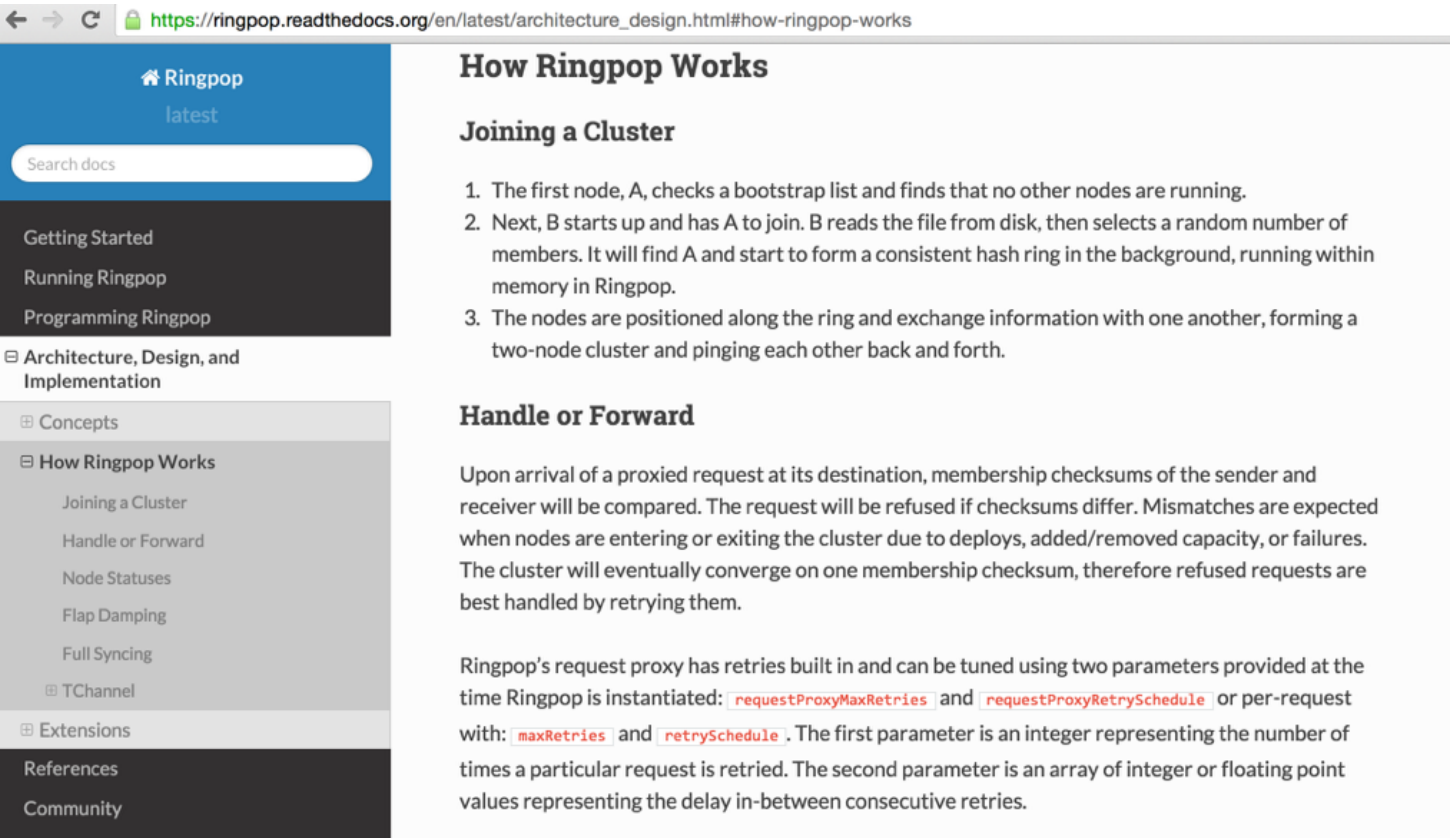
ringpop-node / +

Merge pull request #175 from uber/move-validate-props ...

jwolski authored 3 hours ago latest commit 387f81d1f6

benchmarks	not to expose membership functions	2 months ago
docs	Fix typos in docs sample code	7 days ago
examples	add an example for RingpopHandler	a month ago
lib	Add Member::getId	a day ago
scripts	Flap damp scoring	a month ago
server	Get rid of a function off Ringpop prototype	a day ago
test	Add Member::getId	a day ago
.gitignore	Initial commit	9 months ago
.jshintrc	Initial commit	9 months ago
.travis.yml	Travis CI integration	8 months ago

You can clone with HTTPS, SSH, or Subversion.



How Ringpop Works

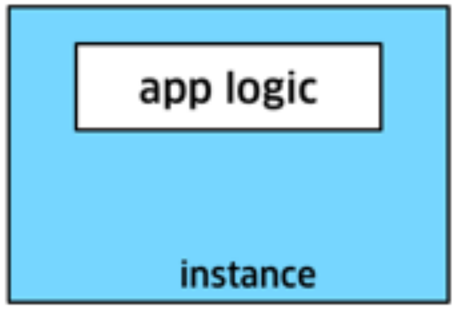
Joining a Cluster

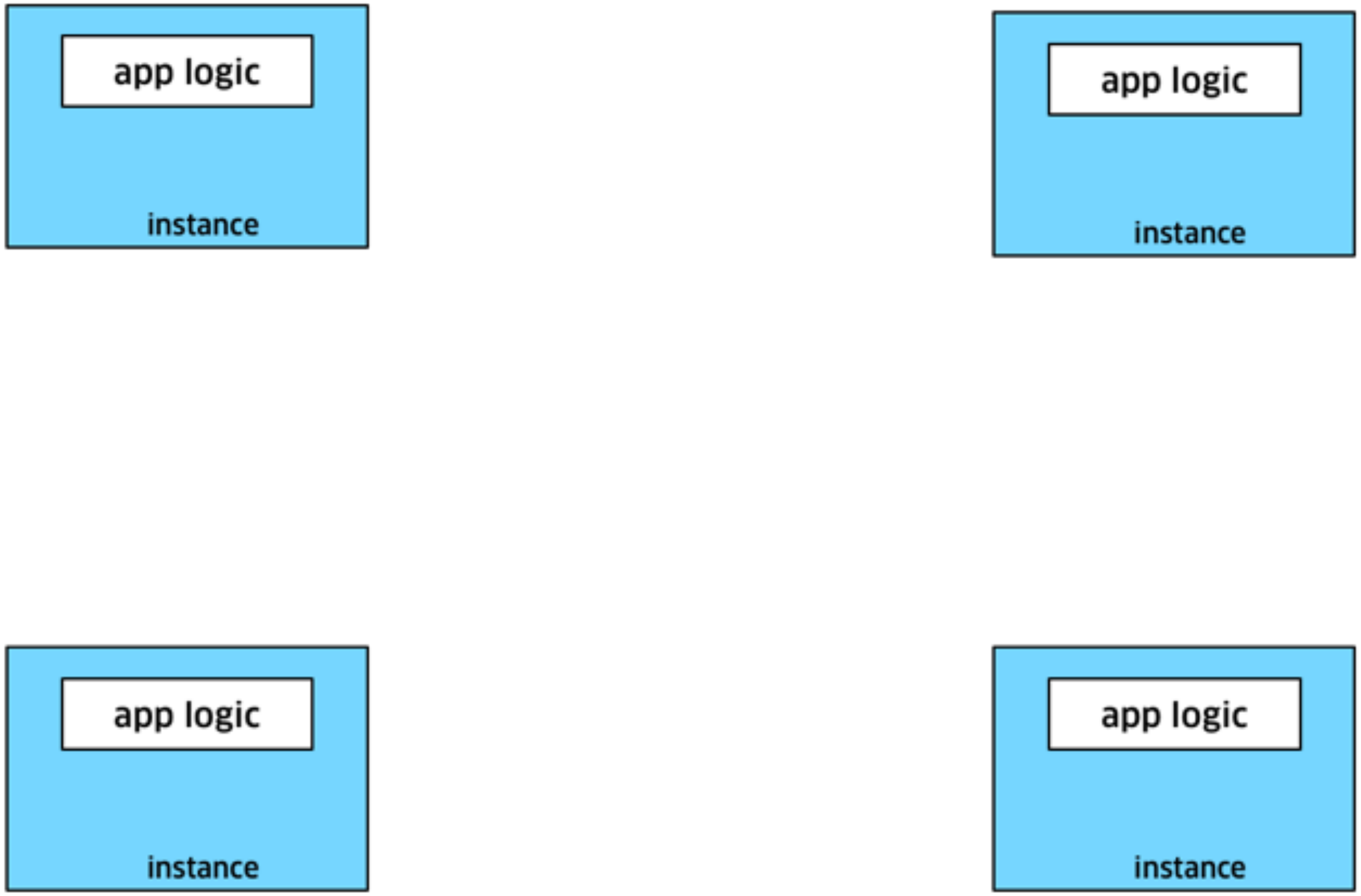
1. The first node, A, checks a bootstrap list and finds that no other nodes are running.
2. Next, B starts up and has A to join. B reads the file from disk, then selects a random number of members. It will find A and start to form a consistent hash ring in the background, running within memory in Ringpop.
3. The nodes are positioned along the ring and exchange information with one another, forming a two-node cluster and pinging each other back and forth.

Handle or Forward

Upon arrival of a proxied request at its destination, membership checksums of the sender and receiver will be compared. The request will be refused if checksums differ. Mismatches are expected when nodes are entering or exiting the cluster due to deploys, added/removed capacity, or failures. The cluster will eventually converge on one membership checksum, therefore refused requests are best handled by retrying them.

Ringpop's request proxy has retries built in and can be tuned using two parameters provided at the time Ringpop is instantiated: `requestProxyMaxRetries` and `requestProxyRetrySchedule` or per-request with: `maxRetries` and `retrySchedule`. The first parameter is an integer representing the number of times a particular request is retried. The second parameter is an array of integer or floating point values representing the delay in-between consecutive retries.


app logic
instance



app logic

instance

app logic

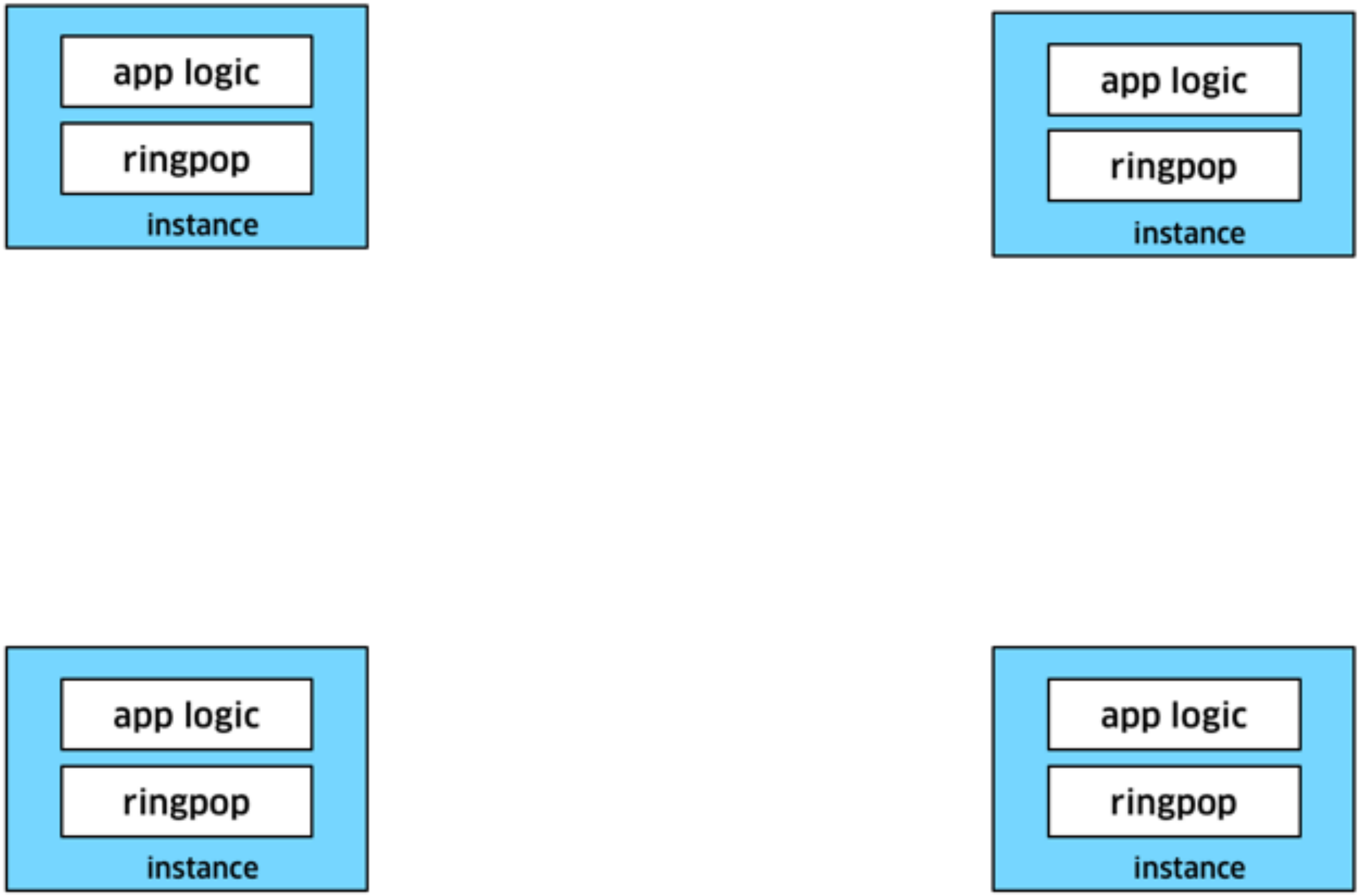
instance

app logic

instance

app logic

instance



app logic

ringpop

instance

app logic

ringpop

instance

app logic

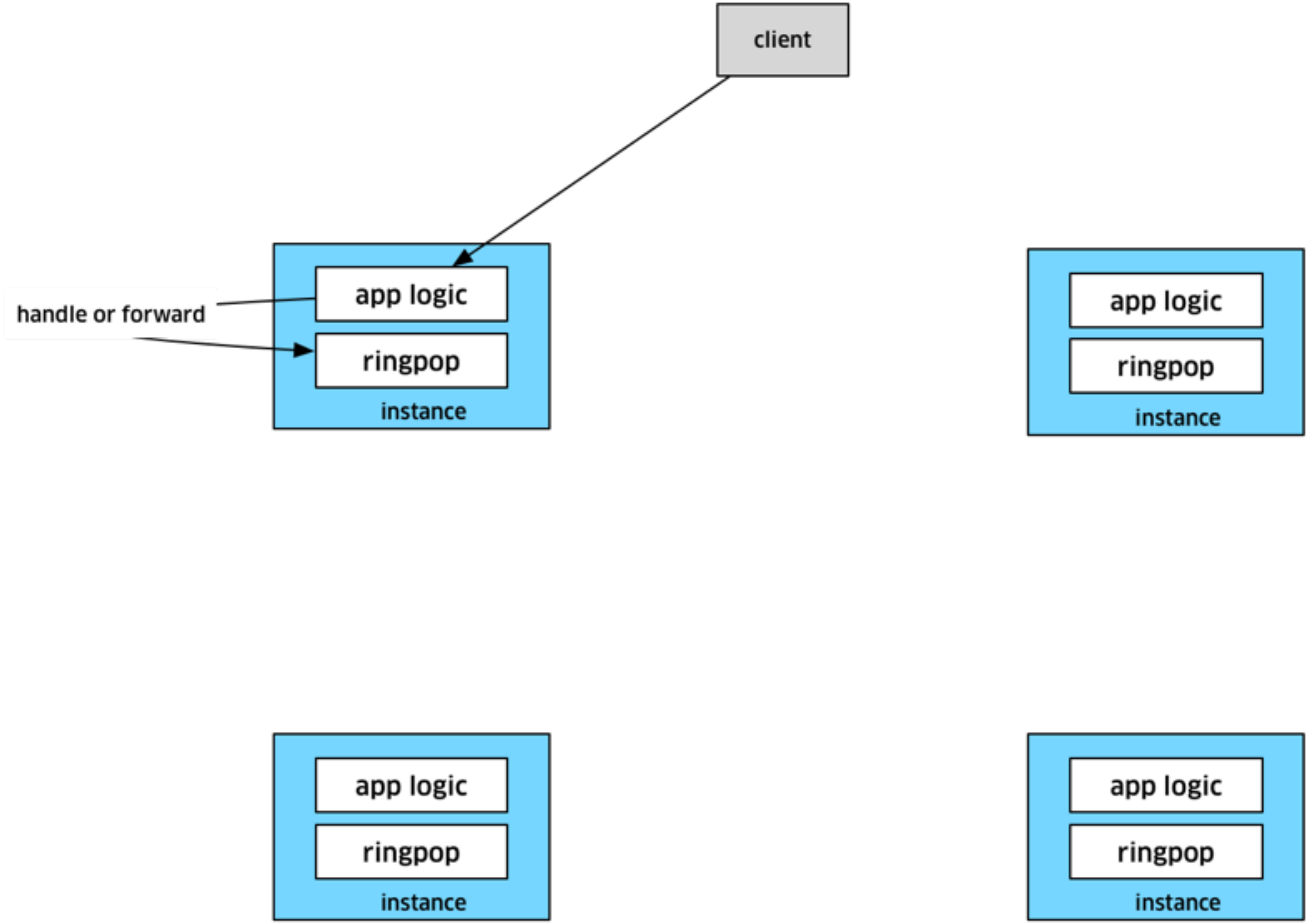
ringpop

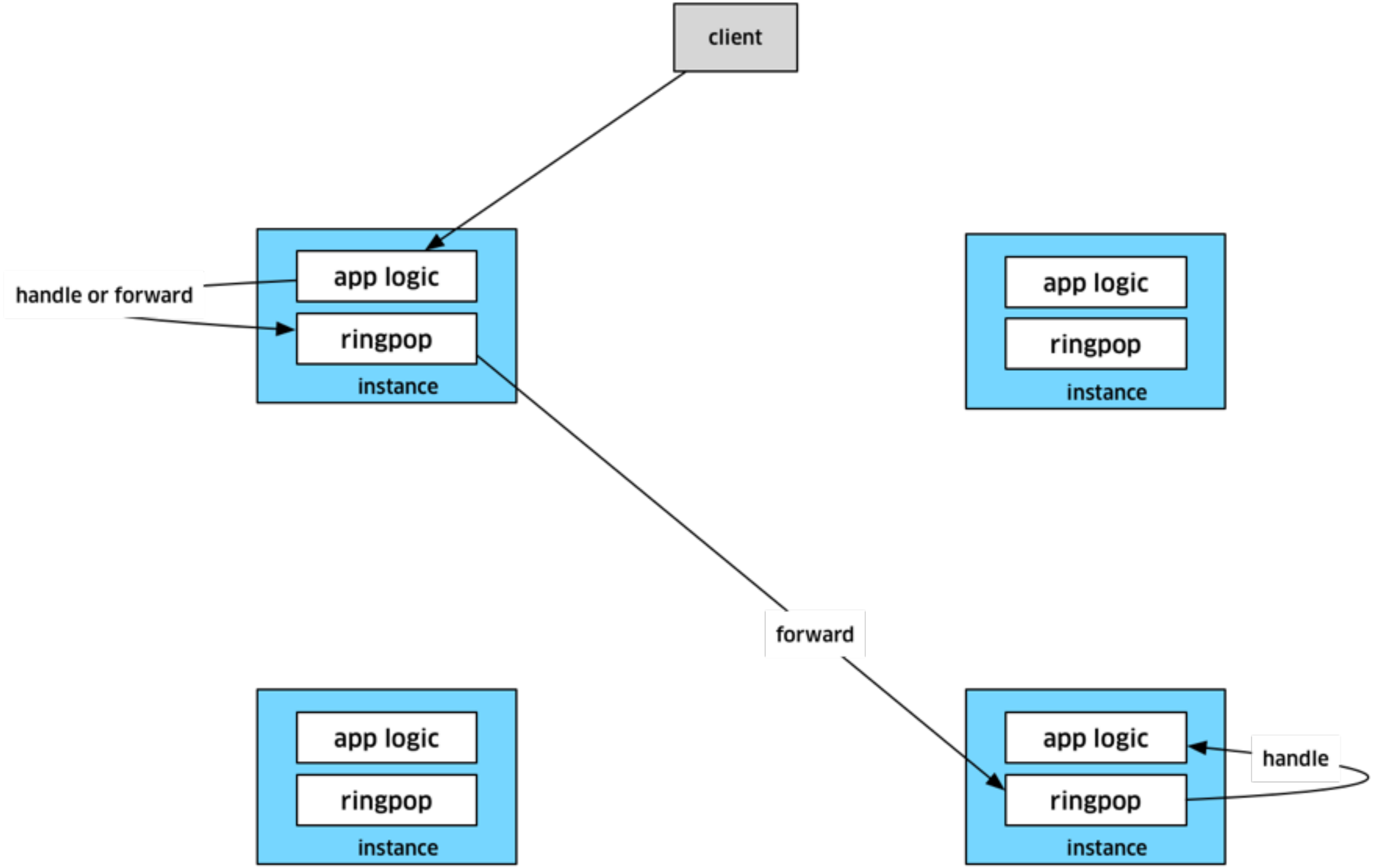
instance

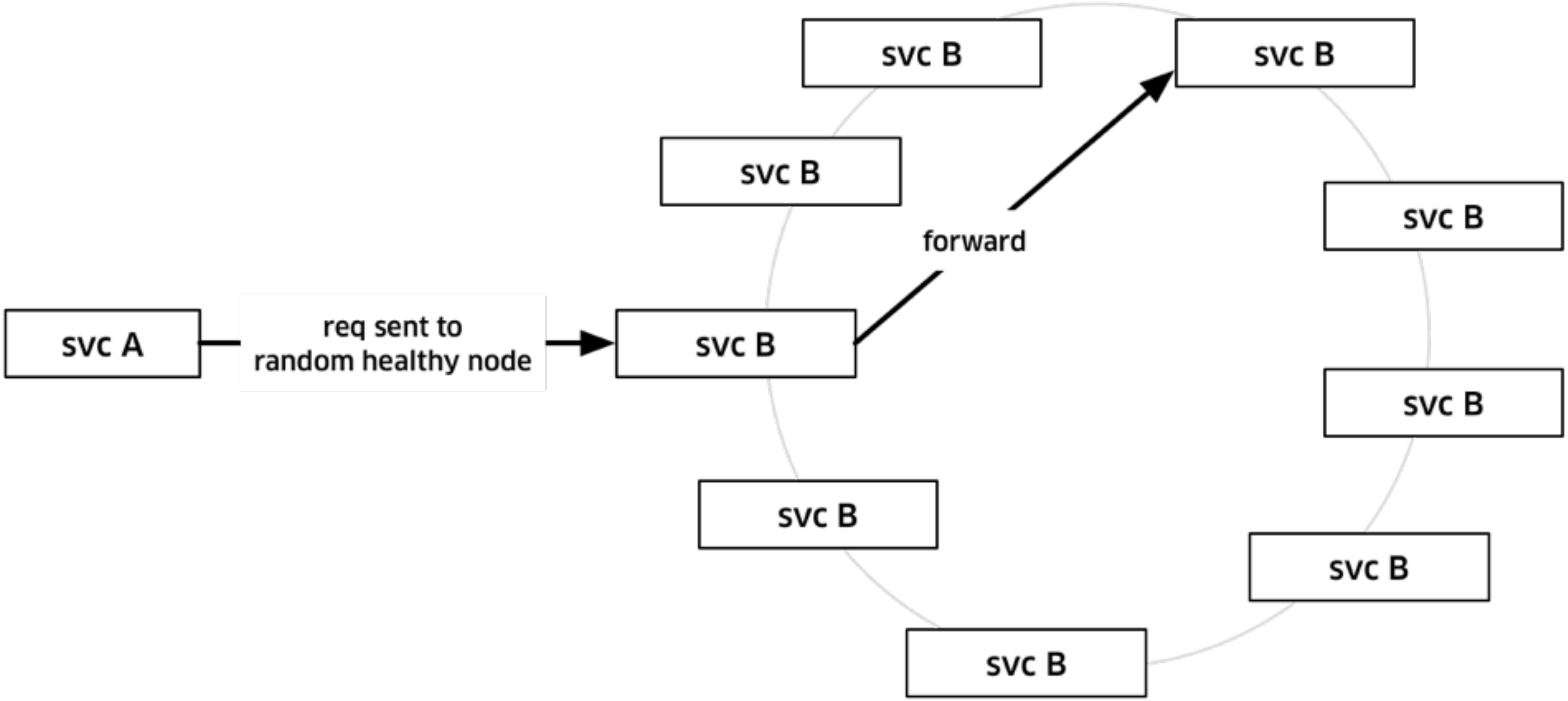
app logic

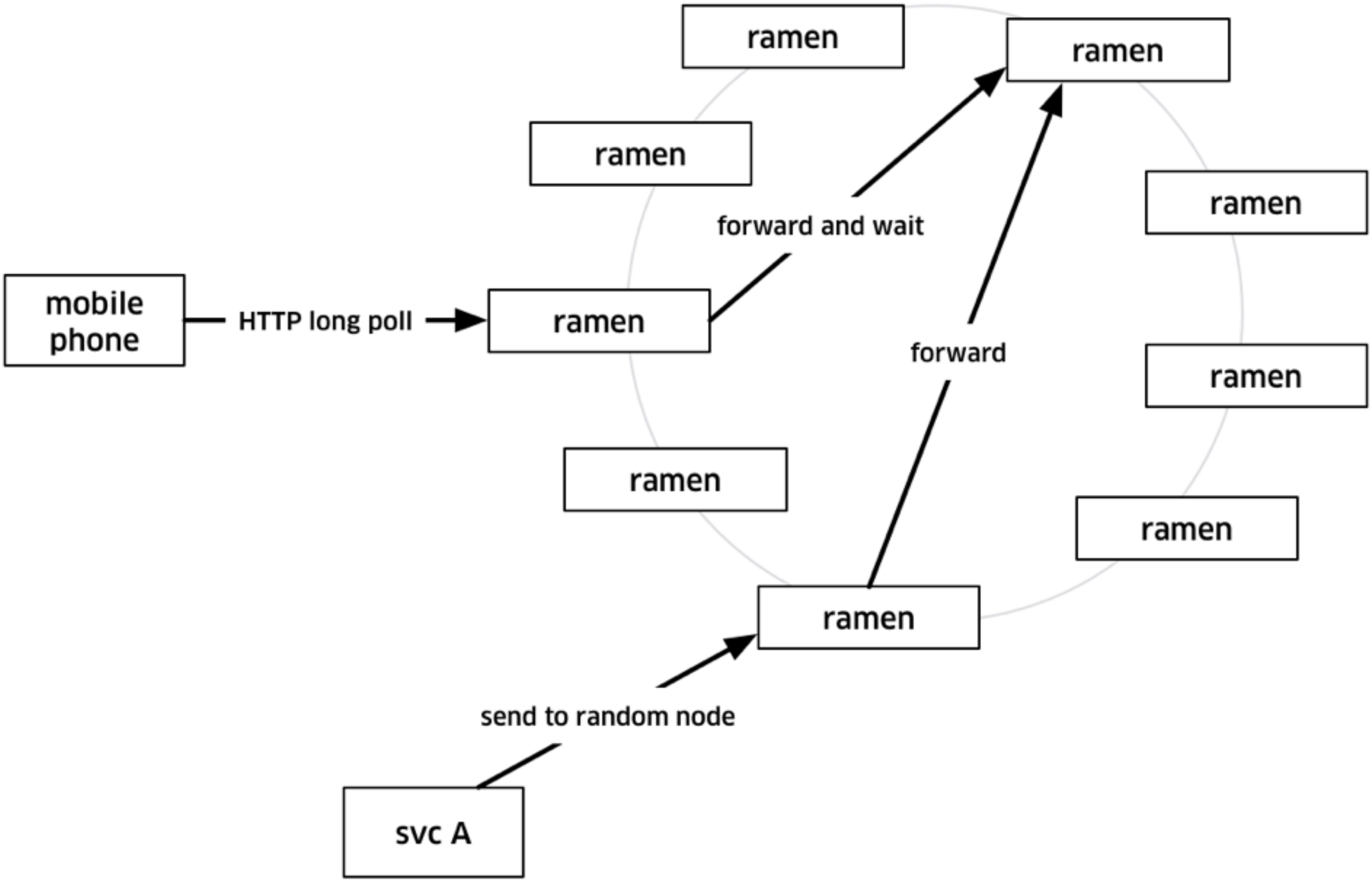
ringpop

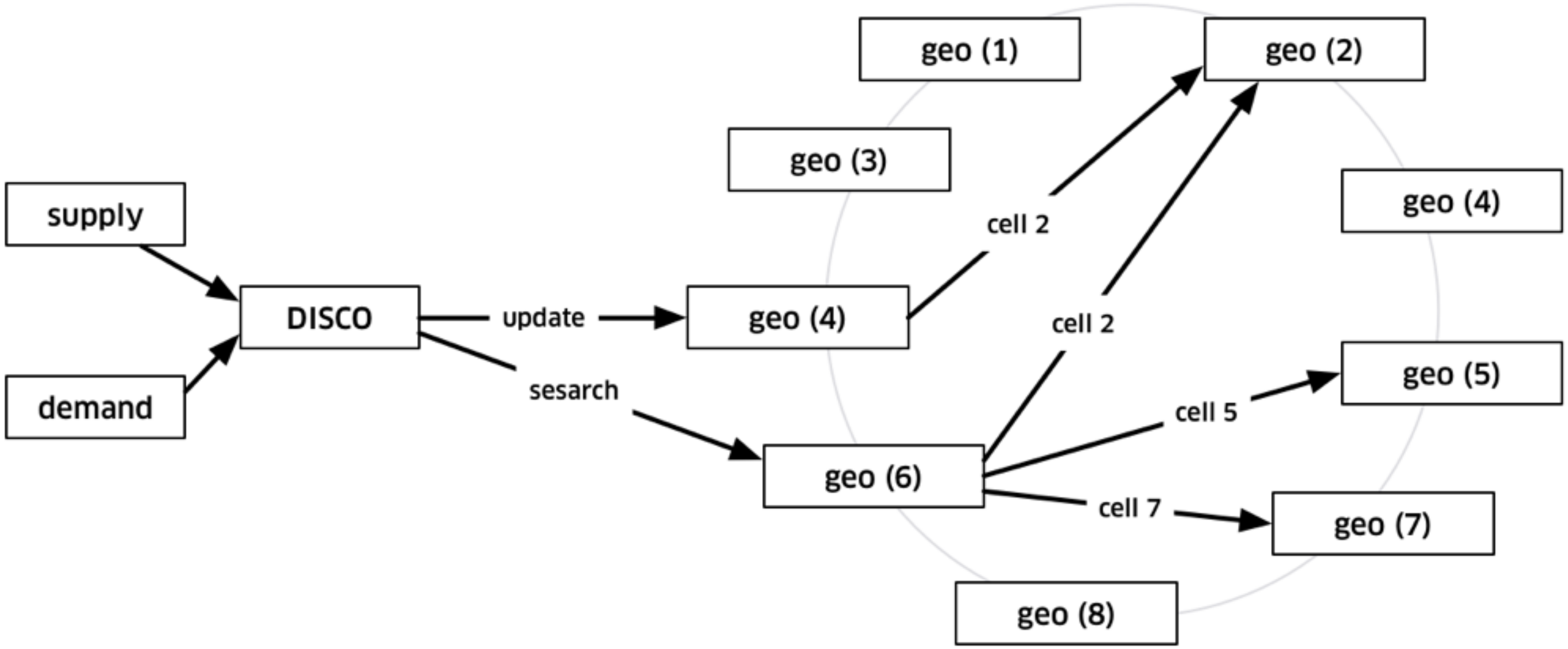
instance


client
handle or forward
app logic
ringpop
instance
app logic
ringpop
instance
app logic
ringpop
instance
app logic
ringpop
instance


client
handle or forward
app logic
ringpop
instance
app logic
ringpop
instance
forward
app logic
ringpop
instance
app logic
handle
ringpop
instance

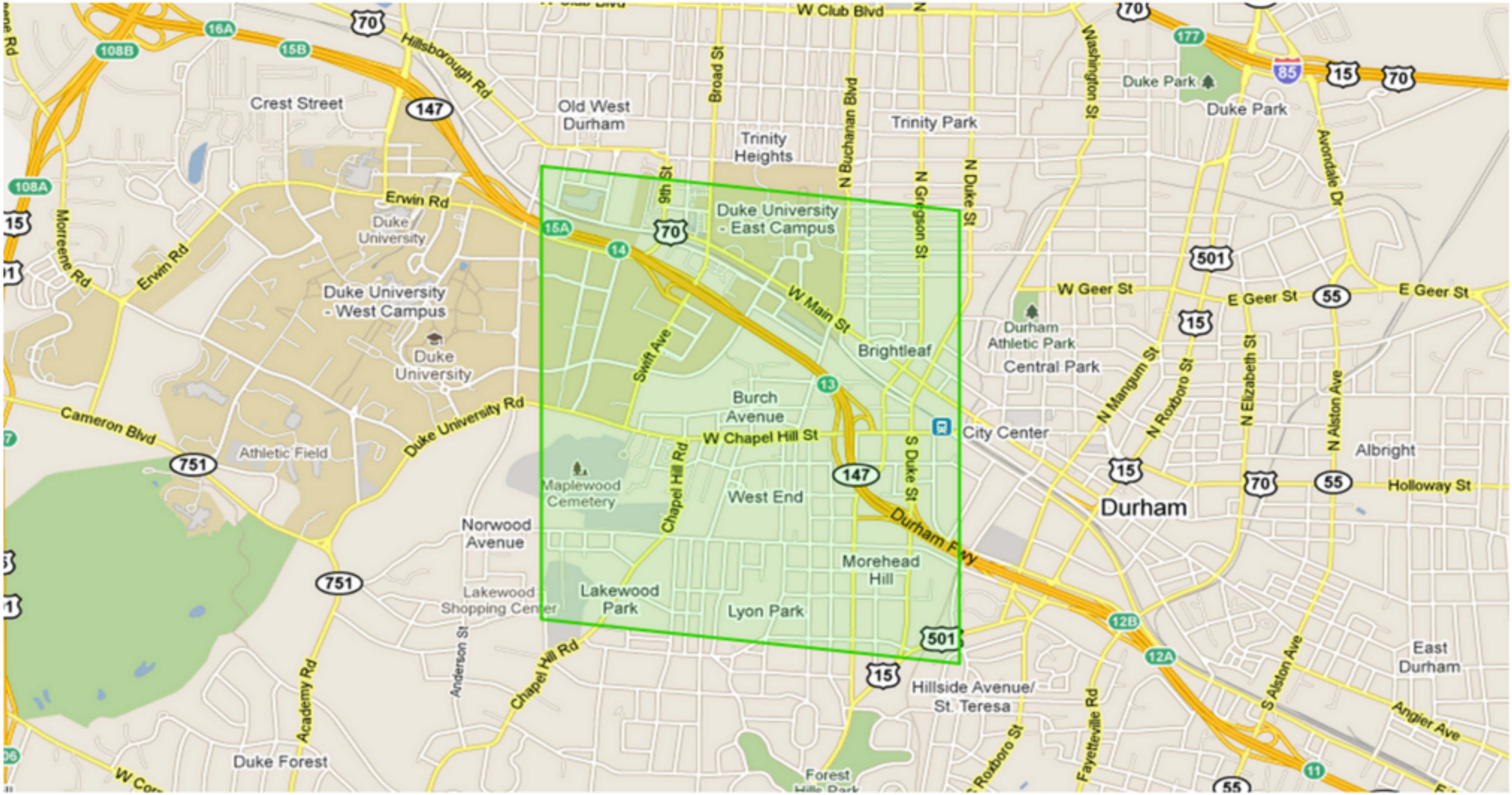

svc A
req sent to
random healthy node
svc B
forward
svc B
svc B
svc B
svc B
svc B
svc B
svc B
svc B


ramen
ramen
ramen
ramen
mobile phone
HTTP long poll
ramen
forward and wait
forward
ramen
ramen
ramen
ramen
send to random node
svc A


supply
demand
DISCO
update
sesarch
geo (1)
geo (2)
geo (3)
geo (4)
cell 2
cell 2
geo (4)
geo (5)
cell 5
geo (6)
cell 7
geo (7)
geo (8)

One S2 Cell

Id: 0x89ace41000000000 (0b10001001101011001110010000010000...), Level: 12



Source: Geometry on the Sphere: Google's S2 Library

S2 Cells - Stats

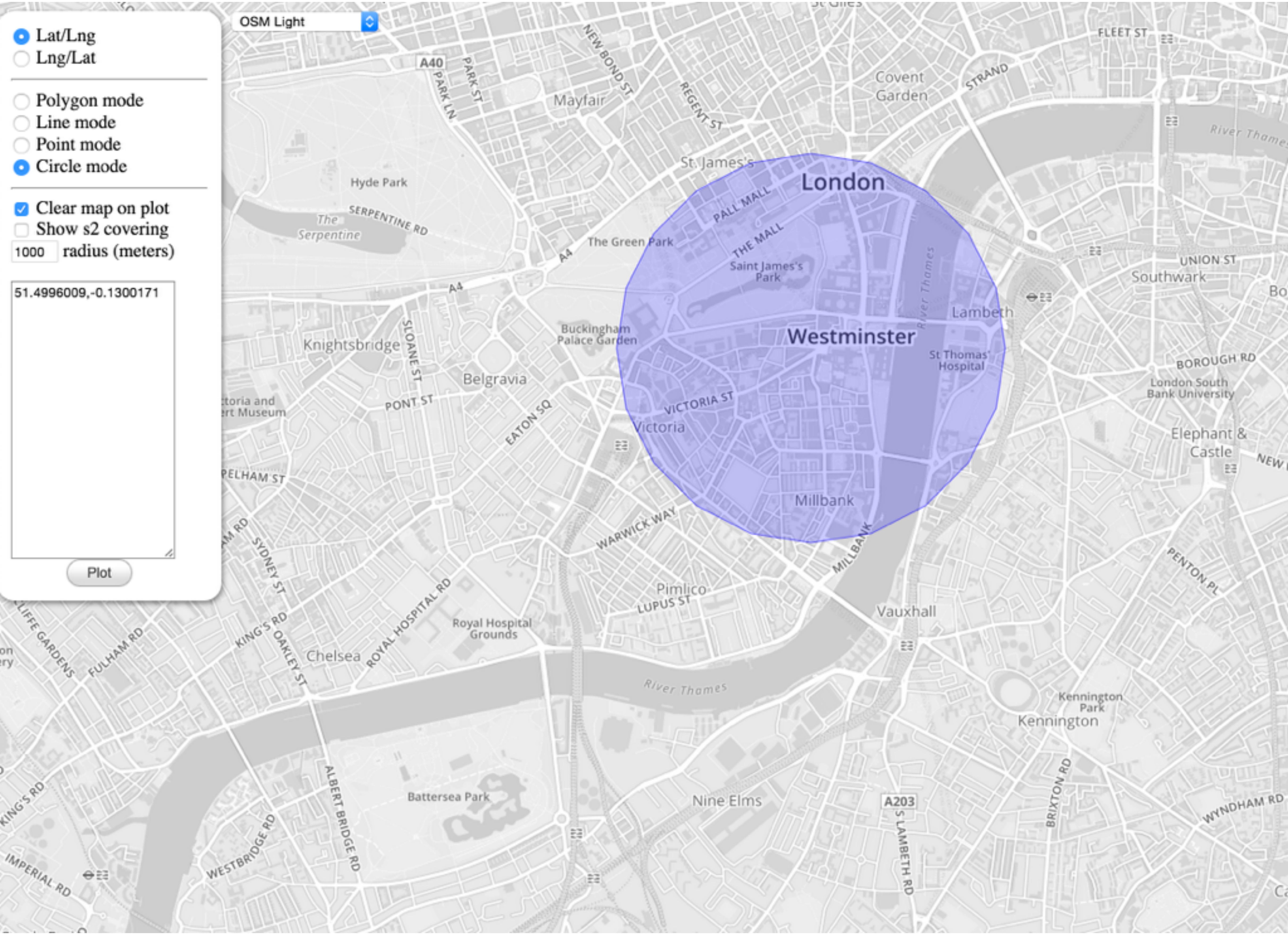
Level	Min Area	Max Area
0	85,011,012 km^2	85,011,012 km^2
1	21,252,753 km^2	21,252,753 km^2
12	3.31 km^2	6.38 km^2
30	0.48 cm^2	0.93 cm^2

⇧

smallest cell

Every cm^2 on Earth can be represented using a 64-bit integer.

Source: Geometry on the Sphere: Google's S2 Library



OSM Light

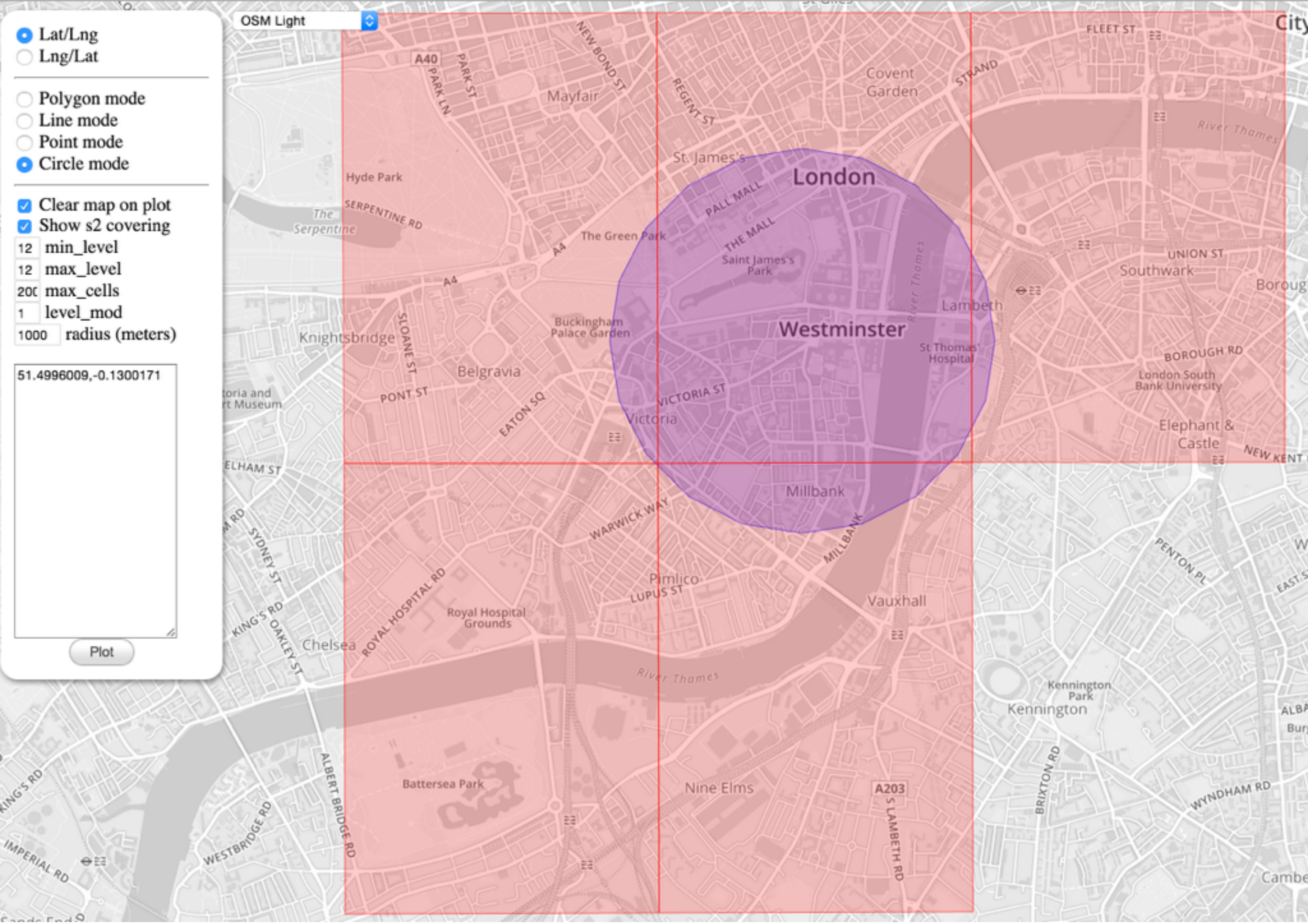
● Lat/Lng
○ Lng/Lat

○ Polygon mode
○ Line mode
○ Point mode
● Circle mode

☑ Clear map on plot
□ Show s2 covering
1000 radius (meters)

51.4996009,-0.1300171

Plot



OSM Light

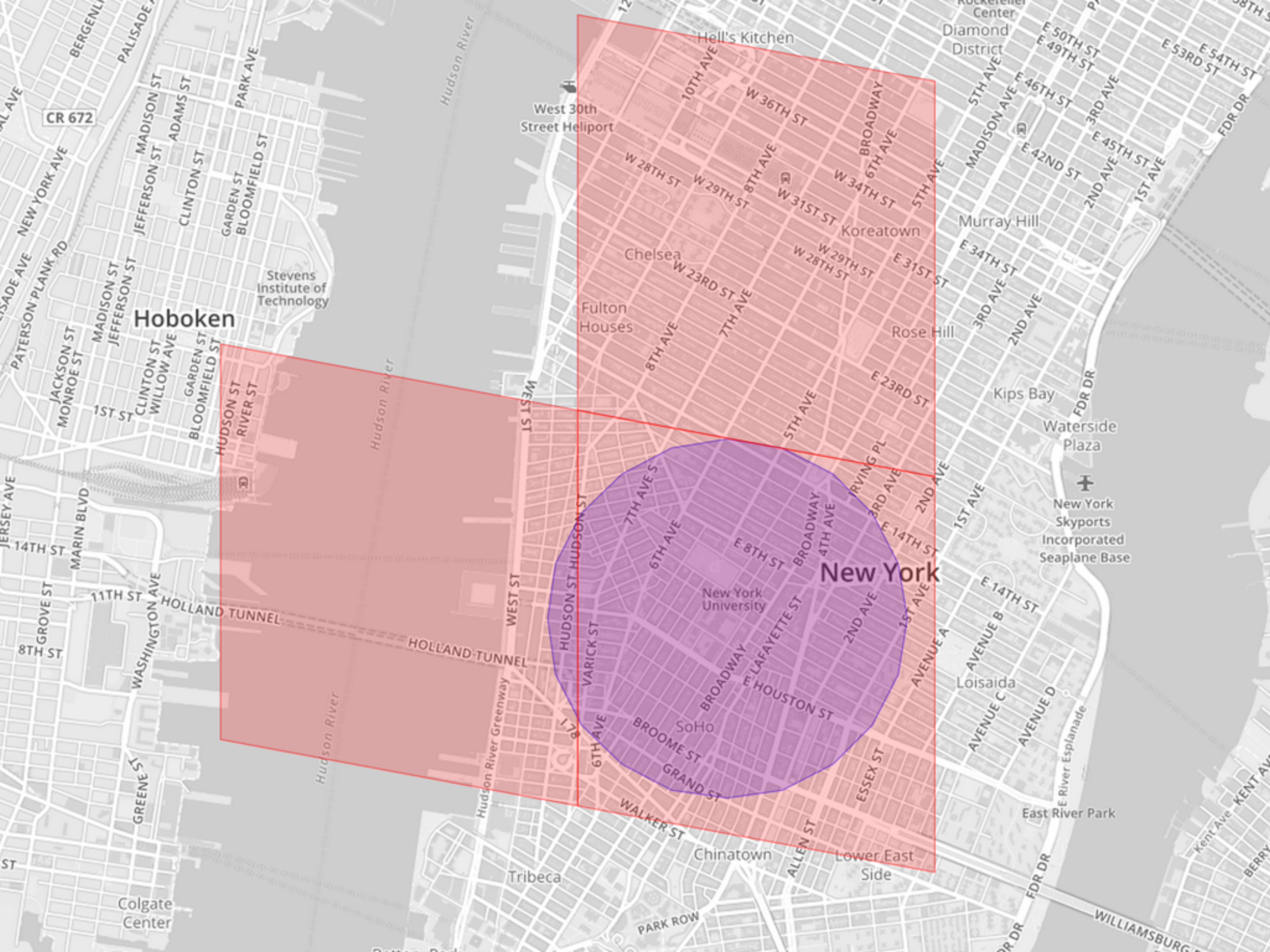
● Lat/Lng
○ Lng/Lat

○ Polygon mode
○ Line mode
○ Point mode
● Circle mode

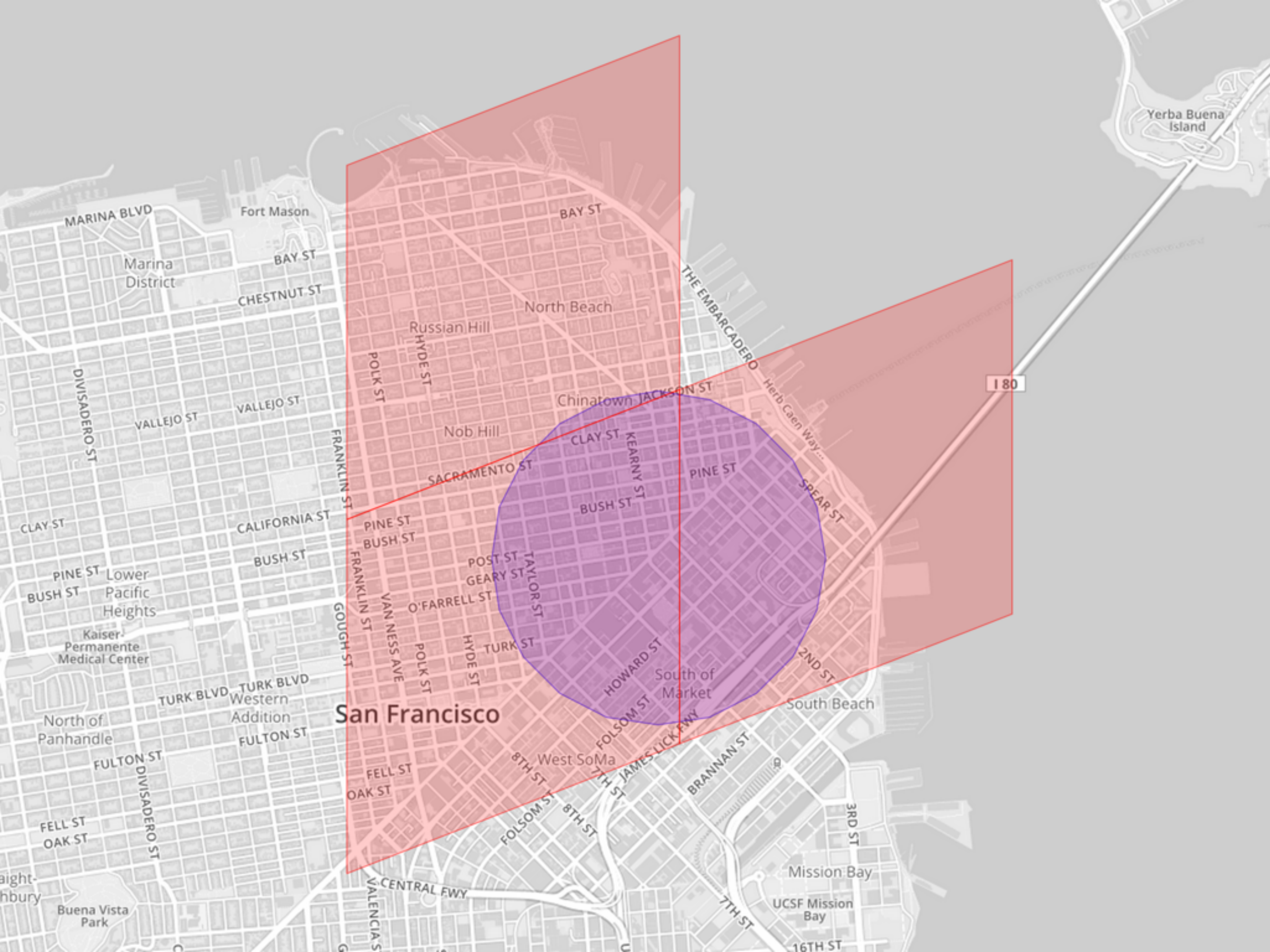
☑ Clear map on plot
☑ Show s2 covering
12 min_level
12 max_level
200 max_cells
1 level_mod
1000 radius (meters)

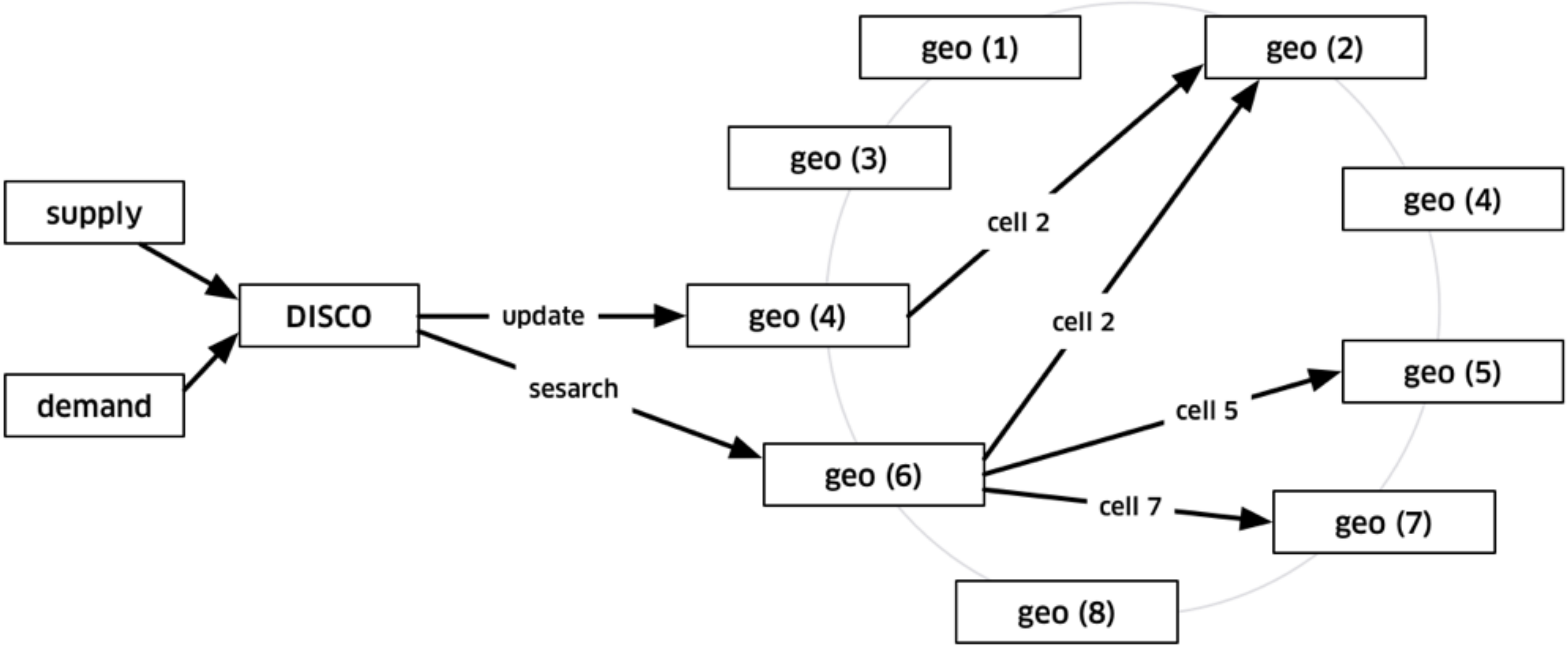
51.4996009,-0.1300171

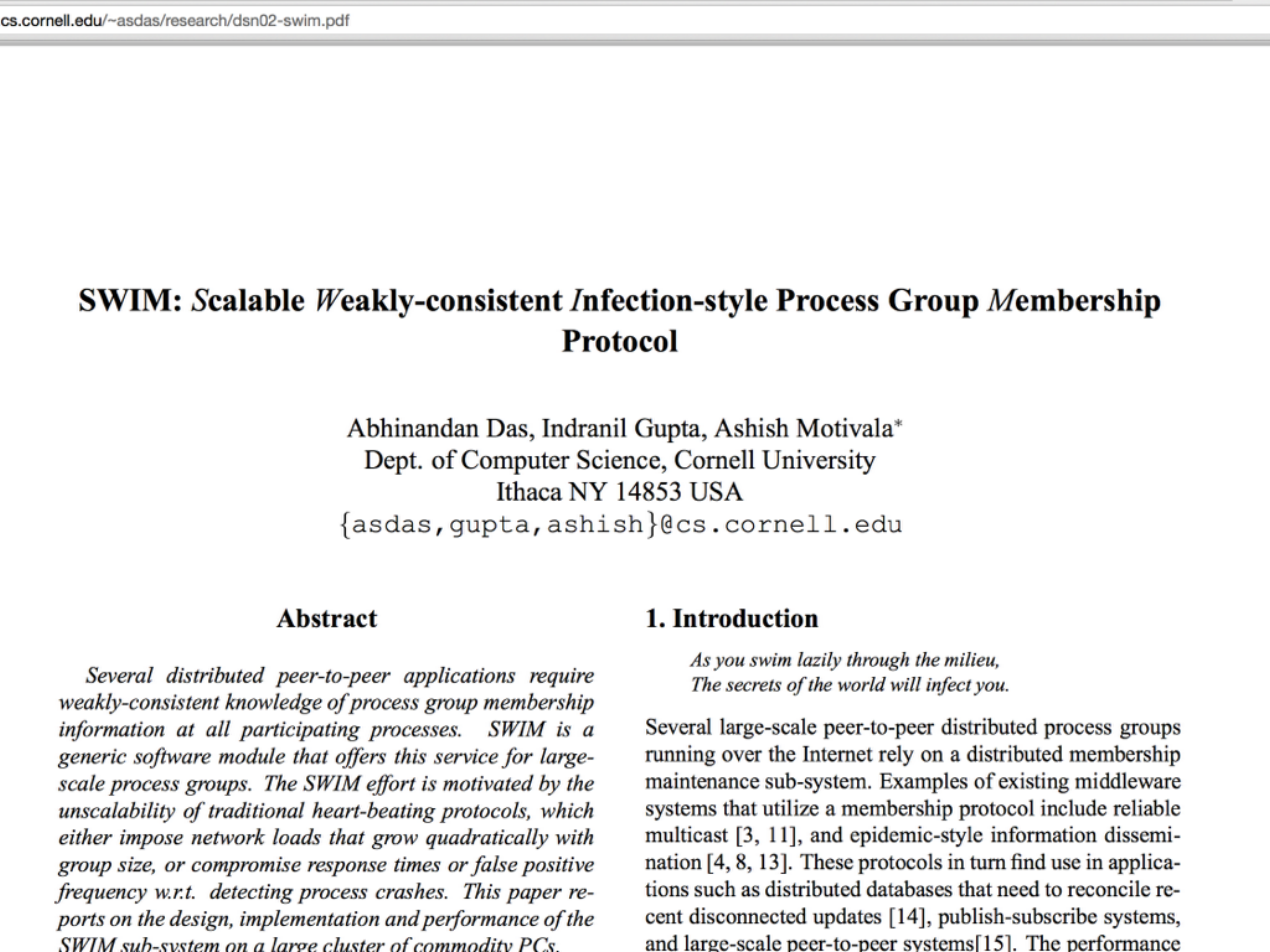
Plot



Hell's Kitchen
Diamond District
Rockefeller Center
West 30th Street Heliport
Hoboken
Stevens Institute of Technology
Chelsea
Koreatown
Murray Hill
Fulton Houses
Rose Hill
Kips Bay
Waterside Plaza
New York Skyports Incorporated Seaplane Base
New York
New York University
SoHo
Loisaida
East River Park
Chinatown
Lower East Side
Tribeca
Colgate Center
Hudson River
Hudson River Greenway
HOLLAND TUNNEL
WILLIAMSBURG
CR 672
W 36TH ST
W 34TH ST
W 31ST ST
W 29TH ST
W 28TH ST
W 23RD ST
E 50TH ST
E 49TH ST
E 46TH ST
E 45TH ST
E 42ND ST
E 53RD ST
E 54TH ST
E 34TH ST
E 31ST ST
E 23RD ST
E 14TH ST
E 8TH ST
E HOUSTON ST
BROOME ST
GRAND ST
WALKER ST
PARK ROW
10TH AVE
8TH AVE
7TH AVE
7TH AVE S
6TH AVE
5TH AVE
BROADWAY
MADISON AVE
3RD AVE
2ND AVE
1ST AVE
4TH AVE
IRVING PL
LAFAYETTE ST
AVENUE A
AVENUE B
AVENUE C
AVENUE D
FDR DR
ESSEX ST
ALLEN ST
VARICK ST
HUDSON ST
WEST ST
I 78
HUDSON ST
RIVER ST
BLOOMFIELD ST
GARDEN ST
WILLOW AVE
CLINTON ST
JEFFERSON ST
MADISON ST
MONROE ST
JACKSON ST
ADAMS ST
PARK AVE
PALISADE AVE
BERGENLINE
NEW YORK AVE
PATERSON PLANK RD
JERSEY AVE
MARIN BLVD
WASHINGTON AVE
GROVE ST
GREENE ST
1ST ST
14TH ST
11TH ST
8TH ST
E River Esplanade
Kent Ave
KENT AVE
BERRY


Yerba Buena Island
MARINA BLVD
Fort Mason
BAY ST
Marina District
BAY ST
CHESTNUT ST
North Beach
Russian Hill
THE EMBARCADERO
POLK ST
HYDE ST
I 80
DIVISADERO ST
VALLEJO ST
VALLEJO ST
Chinatown
JACKSON ST
Herb Caen Way...
Nob Hill
CLAY ST
KEARNY ST
FRANKLIN ST
SACRAMENTO ST
PINE ST
SPEAR ST
BUSH ST
CLAY ST
CALIFORNIA ST
PINE ST
BUSH ST
BUSH ST
PINE ST
Lower Pacific Heights
BUSH ST
POST ST
GEARY ST
TAYLOR ST
FRANKLIN ST
O'FARRELL ST
GOUGH ST
VAN NESS AVE
Kaiser-Permanente Medical Center
POLK ST
HYDE ST
TURK ST
HOWARD ST
2ND ST
South of Market
TURK BLVD
TURK BLVD
Western Addition
San Francisco
South Beach
North of Panhandle
FOLSOM ST
FULTON ST
JAMES LICK FWY
FULTON ST
BRANNAN ST
West SoMa
8TH ST
7TH ST
FELL ST
OAK ST
DIVISADERO ST
FOLSOM ST
8TH ST
FELL ST
OAK ST
3RD ST
Haight-Ashbury
Mission Bay
CENTRAL FWY
VALENCIA ST
Buena Vista Park
UCSF Mission Bay
7TH ST
16TH ST


supply
demand
DISCO
update
sesarch
geo (1)
geo (2)
geo (3)
geo (4)
geo (4)
cell 2
cell 2
geo (5)
cell 5
geo (6)
cell 7
geo (7)
geo (8)



cs.cornell.edu/~asdas/research/dsn02-swim.pdf

SWIM: *S*calable *W*eakly-consistent *I*nfection-style Process Group *M*embership Protocol

Abhinandan Das, Indranil Gupta, Ashish Motivala*
Dept. of Computer Science, Cornell University
Ithaca NY 14853 USA
{asdas,gupta,ashish}@cs.cornell.edu

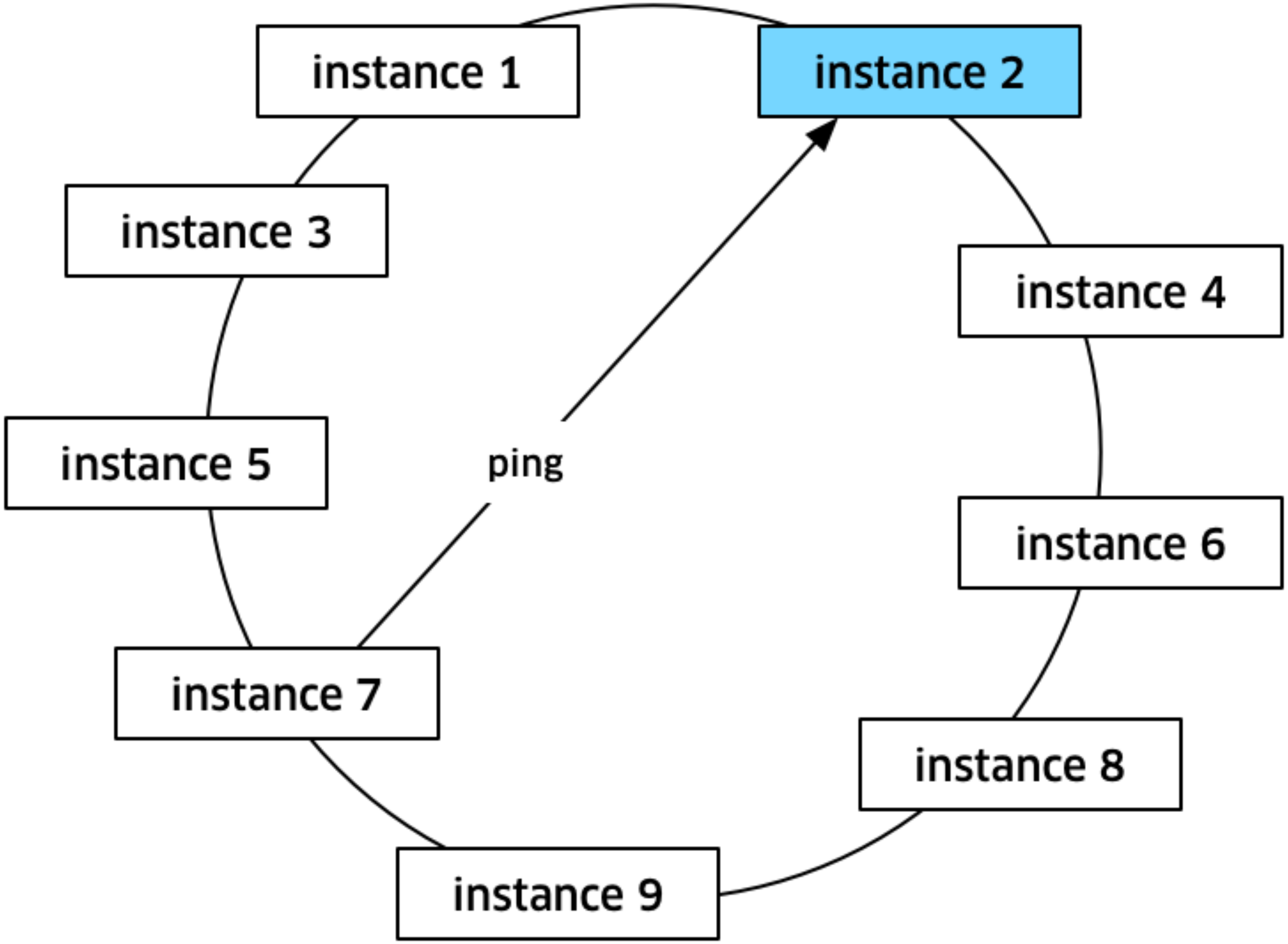
Abstract

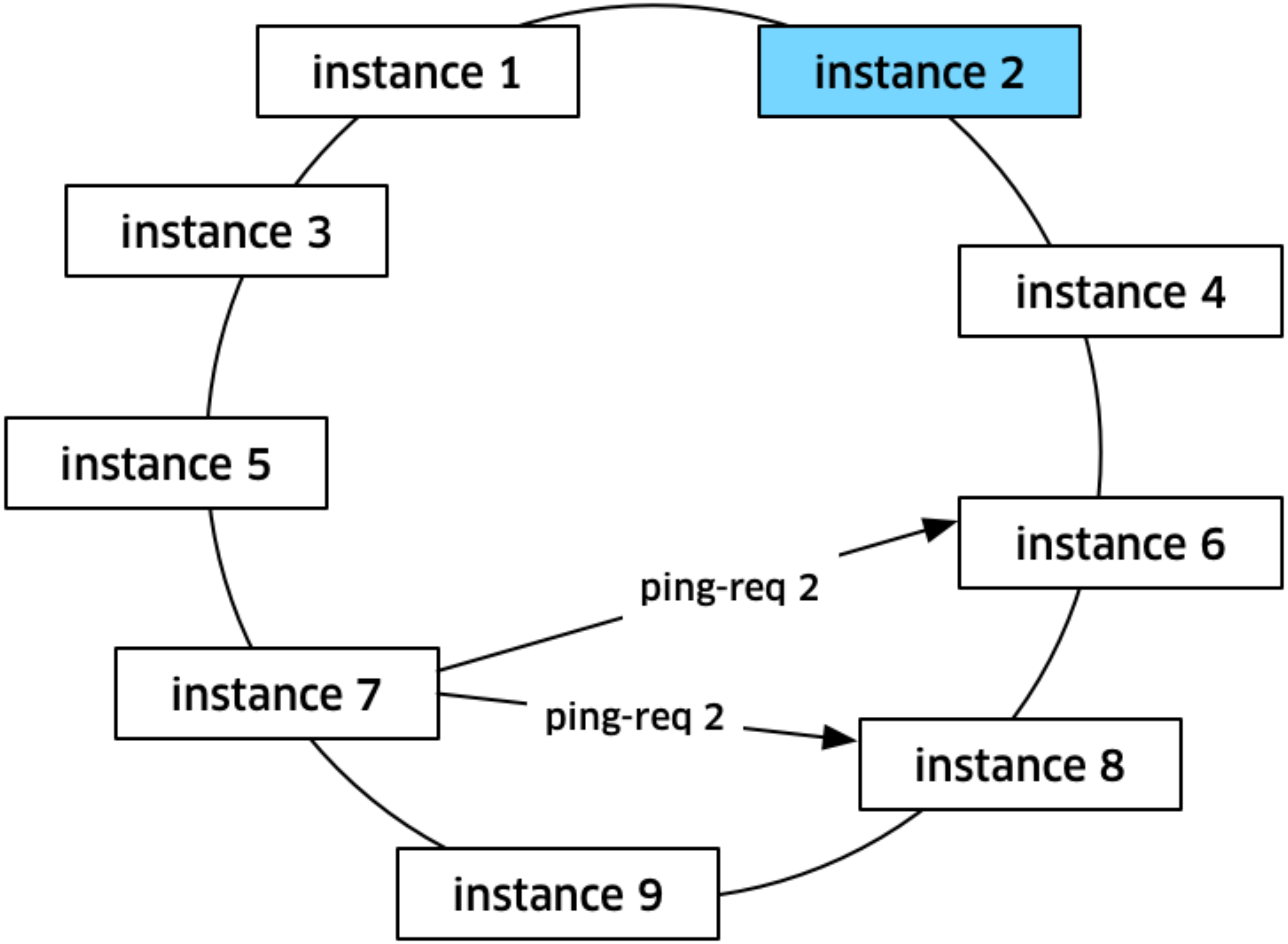
Several distributed peer-to-peer applications require weakly-consistent knowledge of process group membership information at all participating processes. SWIM is a generic software module that offers this service for large-scale process groups. The SWIM effort is motivated by the unscalability of traditional heart-beating protocols, which either impose network loads that grow quadratically with group size, or compromise response times or false positive frequency w.r.t. detecting process crashes. This paper reports on the design, implementation and performance of the SWIM sub-system on a large cluster of commodity PCs.

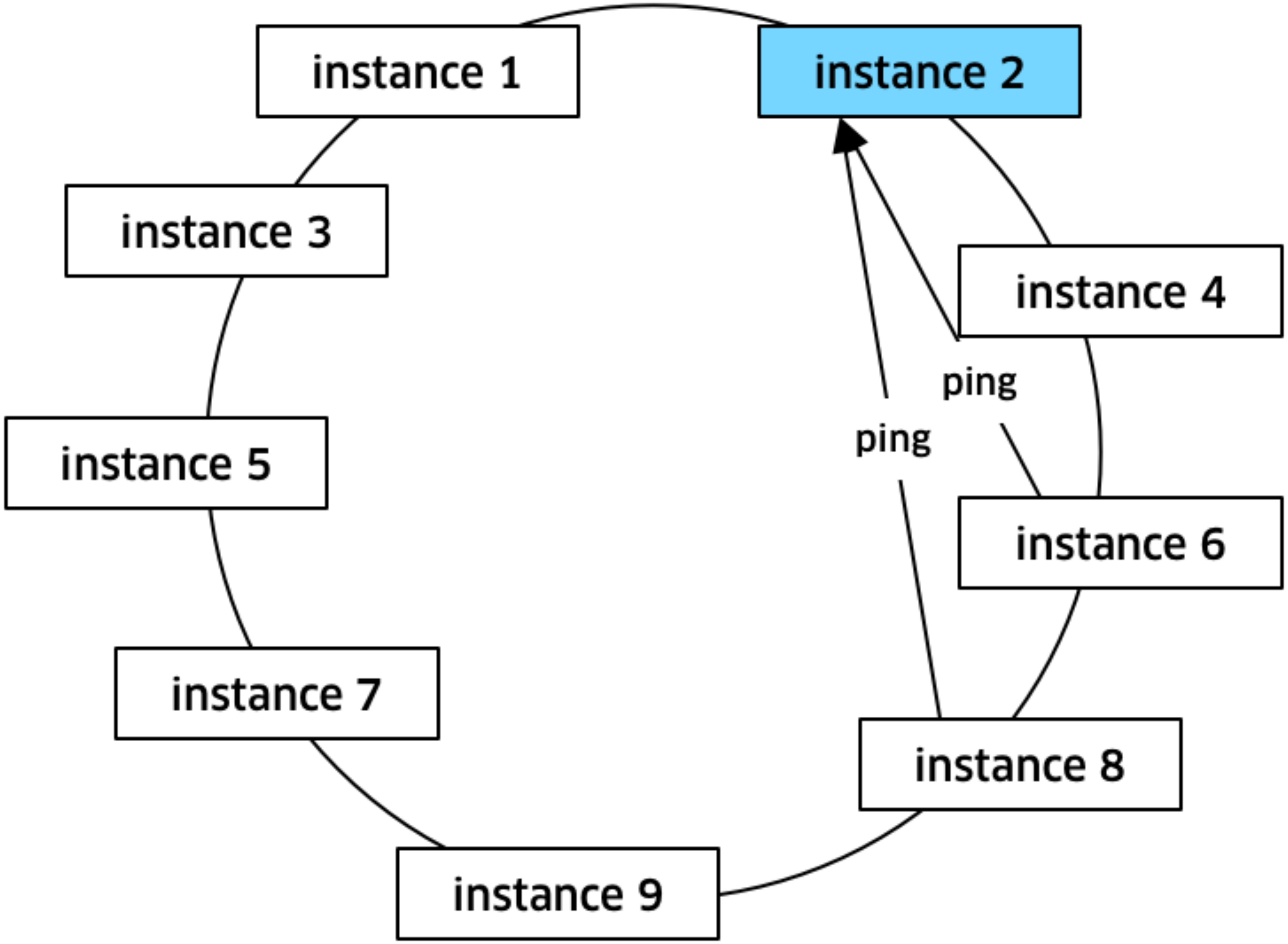
1. Introduction

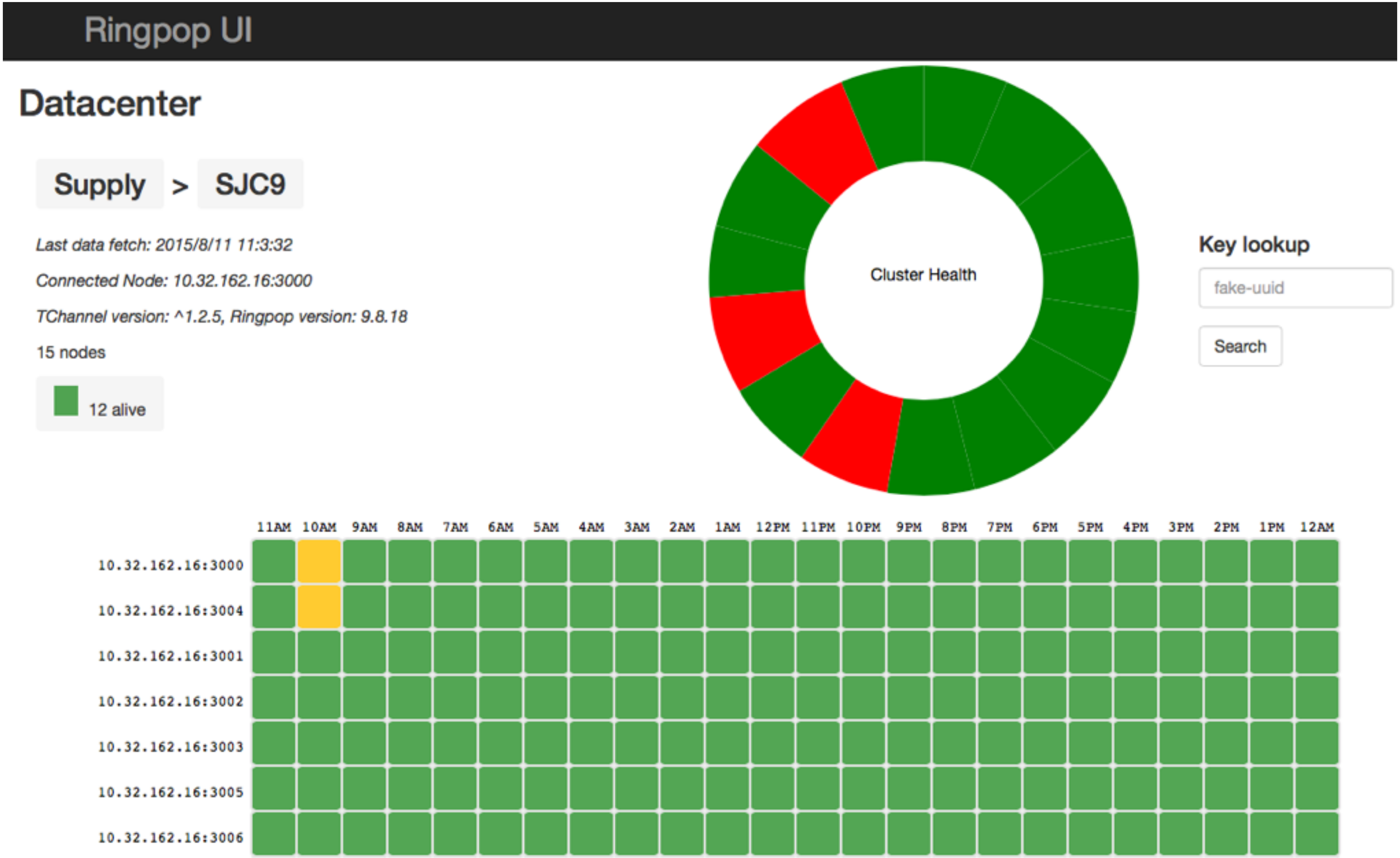
As you swim lazily through the milieu,
The secrets of the world will infect you.

Several large-scale peer-to-peer distributed process groups running over the Internet rely on a distributed membership maintenance sub-system. Examples of existing middleware systems that utilize a membership protocol include reliable multicast [3, 11], and epidemic-style information dissemination [4, 8, 13]. These protocols in turn find use in applications such as distributed databases that need to reconcile recent disconnected updates [14], publish-subscribe systems, and large-scale peer-to-peer systems[15]. The performance


instance 1
instance 2
instance 3
instance 4
instance 5
ping
instance 6
instance 7
instance 8
instance 9


instance 1
instance 2
instance 3
instance 4
instance 5
instance 6
ping-req 2
instance 7
ping-req 2
instance 8
instance 9


instance 1
instance 2
instance 3
instance 4
ping
ping
instance 5
instance 6
instance 7
instance 8
instance 9



Ringpop UI

Datacenter

Supply > SJC9

Last data fetch: 2015/8/11 11:3:32

Connected Node: 10.32.162.16:3000

TChannel version: ^1.2.5, Ringpop version: 9.8.18

15 nodes

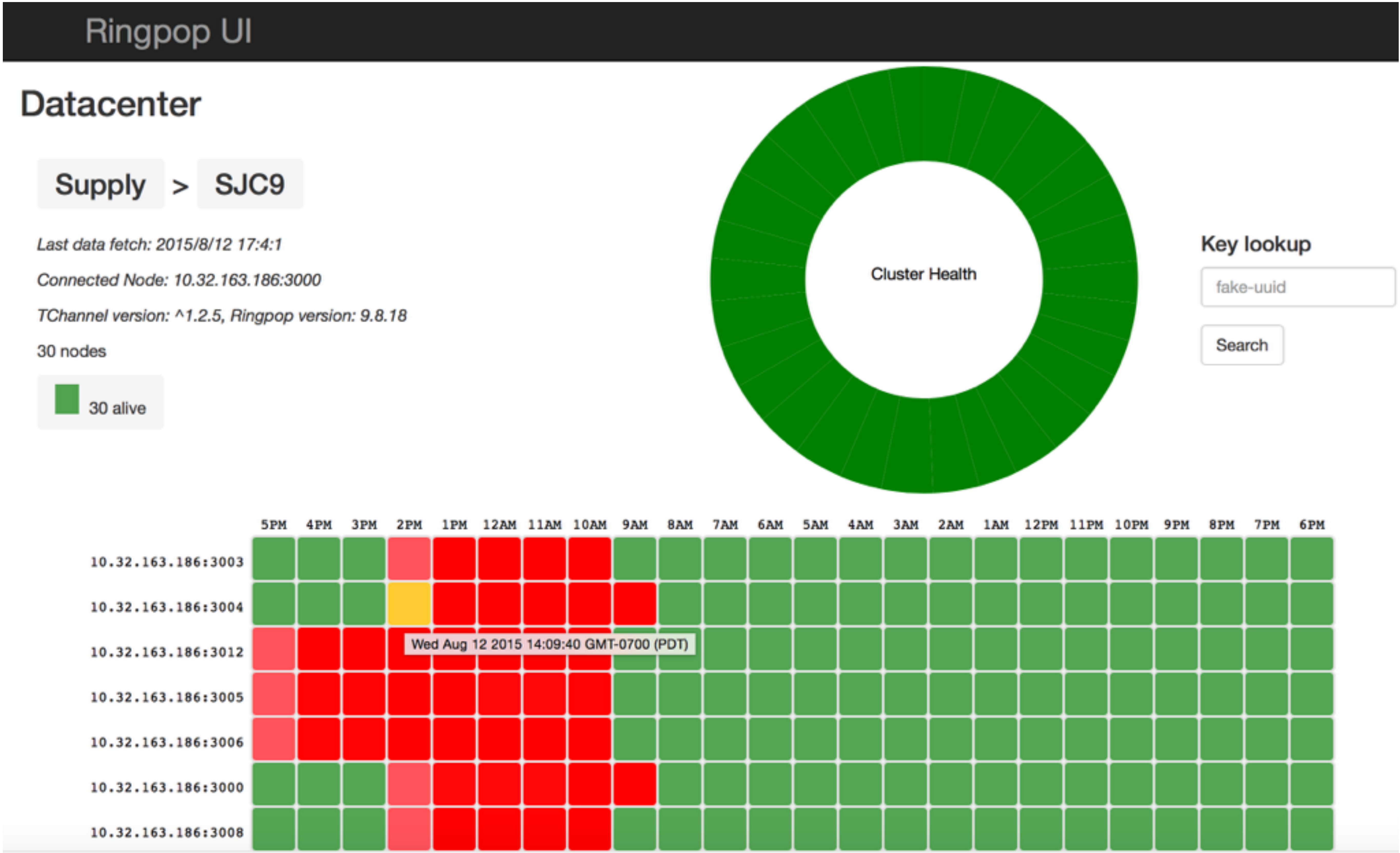
🟩 12 alive

Key lookup

fake-uuid

Search

	11AM	10AM	9AM	8AM	7AM	6AM	5AM	4AM	3AM	2AM	1AM	12PM	11PM	10PM	9PM	8PM	7PM	6PM	5PM	4PM	3PM	2PM	1PM	12AM
10.32.162.16:3000	🟩	🟨	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
10.32.162.16:3004	🟩	🟨	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
10.32.162.16:3001	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
10.32.162.16:3002	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
10.32.162.16:3003	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
10.32.162.16:3005	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
10.32.162.16:3006	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩



Ringpop UI

Datacenter

Supply > SJC9

Last data fetch: 2015/8/12 17:4:1

Connected Node: 10.32.163.186:3000

TChannel version: ^1.2.5, Ringpop version: 9.8.18

30 nodes

🟩 30 alive

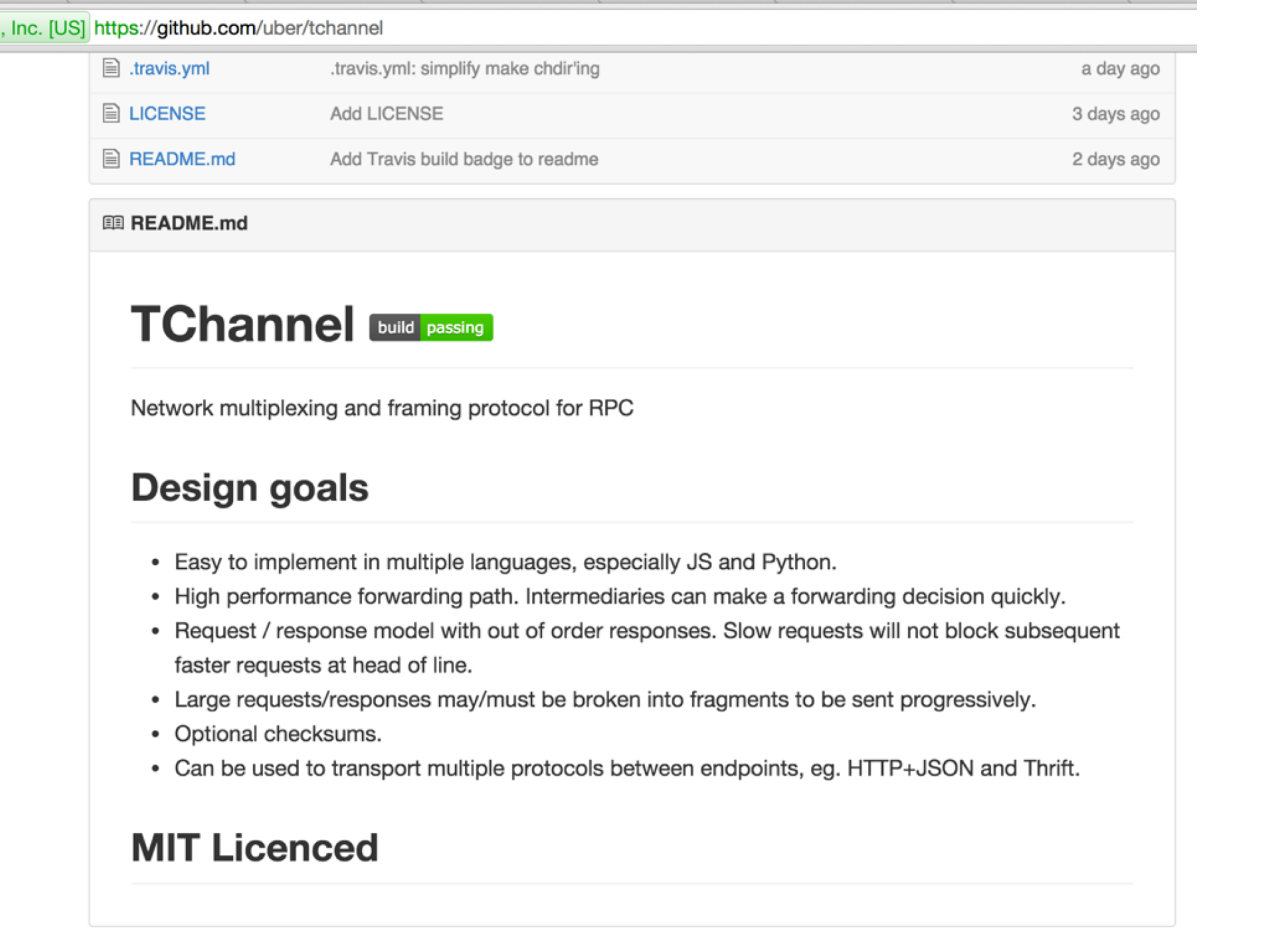
Key lookup

fake-uuid

Search

	5PM	4PM	3PM	2PM	1PM	12AM	11AM	10AM	9AM	8AM	7AM	6AM	5AM	4AM	3AM	2AM	1AM	12PM	11PM	10PM	9PM	8PM	7PM	6PM
10.32.163.186:3003	🟩	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
10.32.163.186:3004	🟩	🟩	🟩	🟨	🟥	🟥	🟥	🟥	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
10.32.163.186:3012	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
10.32.163.186:3005	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
10.32.163.186:3006	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
10.32.163.186:3000	🟩	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
10.32.163.186:3008	🟩	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩

Wed Aug 12 2015 14:09:40 GMT-0700 (PDT)



, Inc. [US] https://github.com/uber/tchannel

.travis.yml	.travis.yml: simplify make chdir'ing	a day ago
LICENSE	Add LICENSE	3 days ago
README.md	Add Travis build badge to readme	2 days ago

README.md

TChannel build passing

Network multiplexing and framing protocol for RPC

Design goals

- Easy to implement in multiple languages, especially JS and Python.
- High performance forwarding path. Intermediaries can make a forwarding decision quickly.
- Request / response model with out of order responses. Slow requests will not block subsequent faster requests at head of line.
- Large requests/responses may/must be broken into fragments to be sent progressively.
- Optional checksums.
- Can be used to transport multiple protocols between endpoints, eg. HTTP+JSON and Thrift.

MIT Licenced

GOALS

- performance
- forwarding
- language support
- proper pipelining
- checksums / tracing
- encapsulation

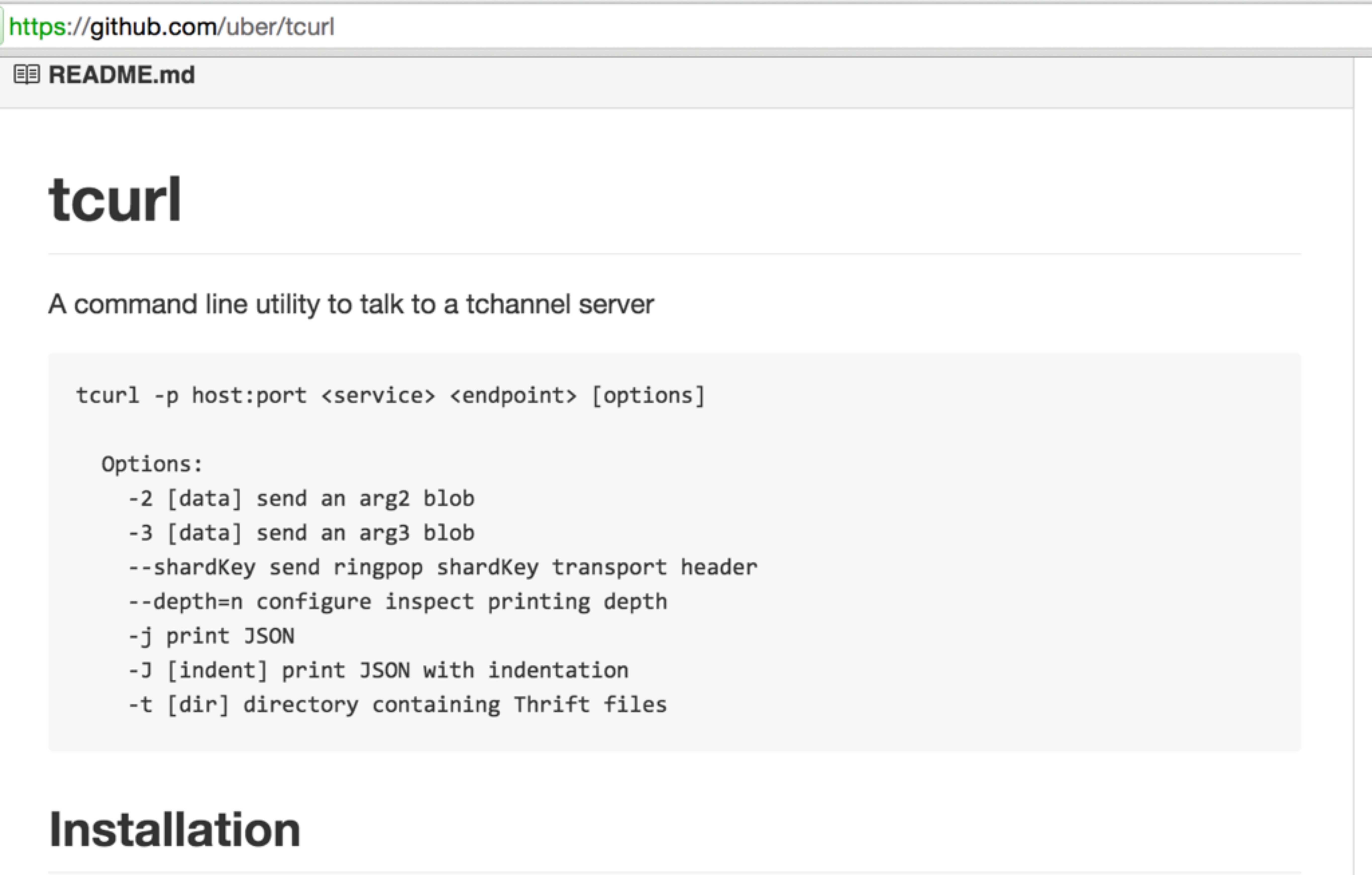
RPC

Getting out of the HTTP and JSON business

HTTP is slow, complex, and inconsistent

JSON is hard to validate and awkward in non-node

Thrift is OK, but generated code is bad



https://github.com/uber/tcurl

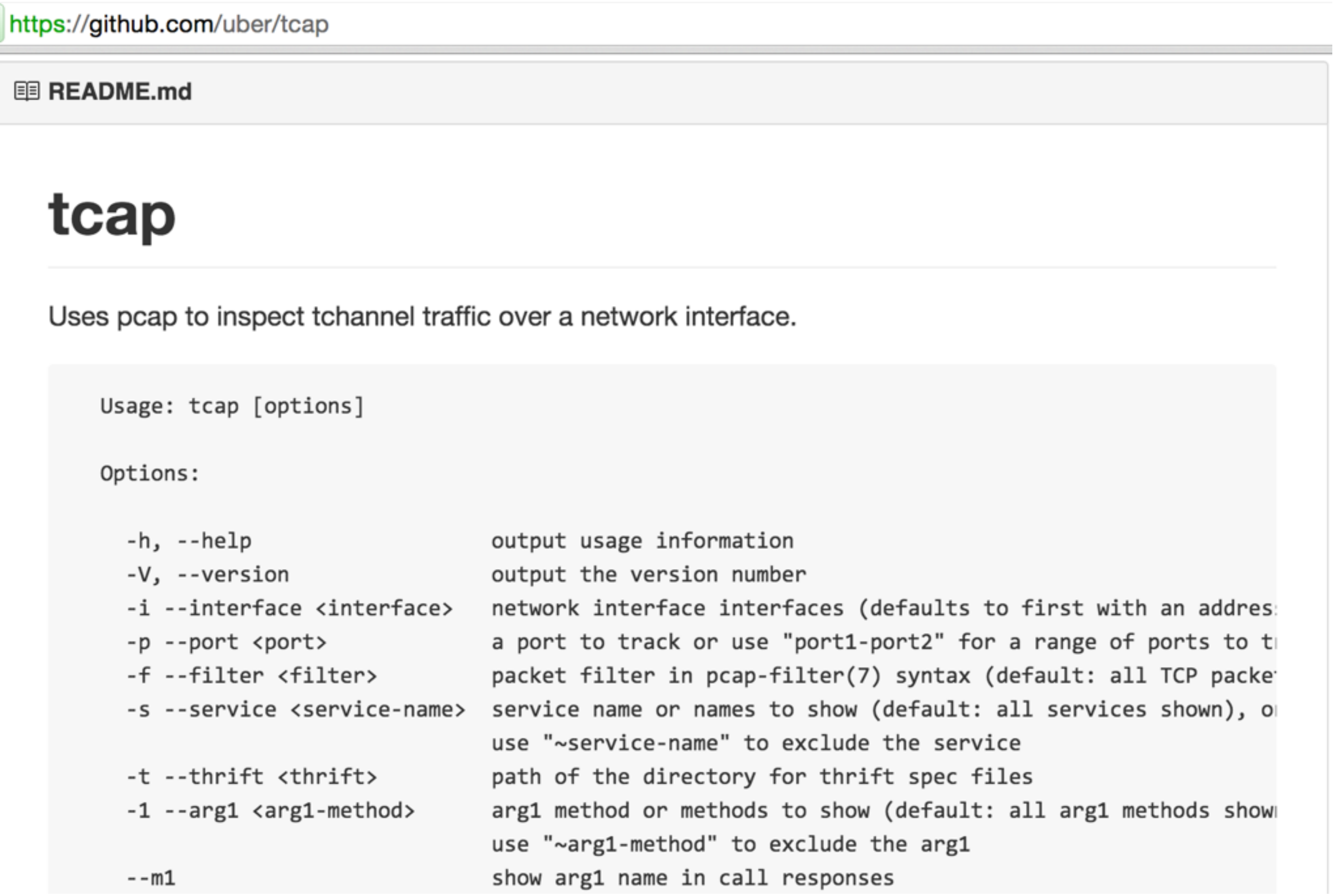
README.md

tcurl

A command line utility to talk to a tchannel server

```
tcurl -p host:port <service> <endpoint> [options]

  Options:
    -2 [data] send an arg2 blob
    -3 [data] send an arg3 blob
    --shardKey send ringpop shardKey transport header
    --depth=n configure inspect printing depth
    -j print JSON
    -J [indent] print JSON with indentation
    -t [dir] directory containing Thrift files
```

Installation



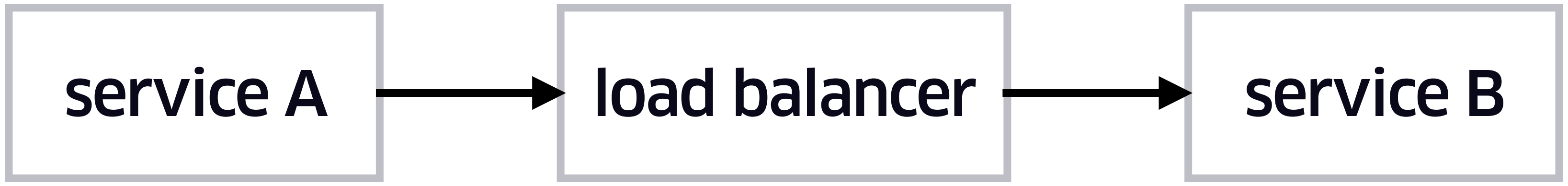
https://github.com/uber/tcap

README.md

tcap

Uses pcap to inspect tchannel traffic over a network interface.

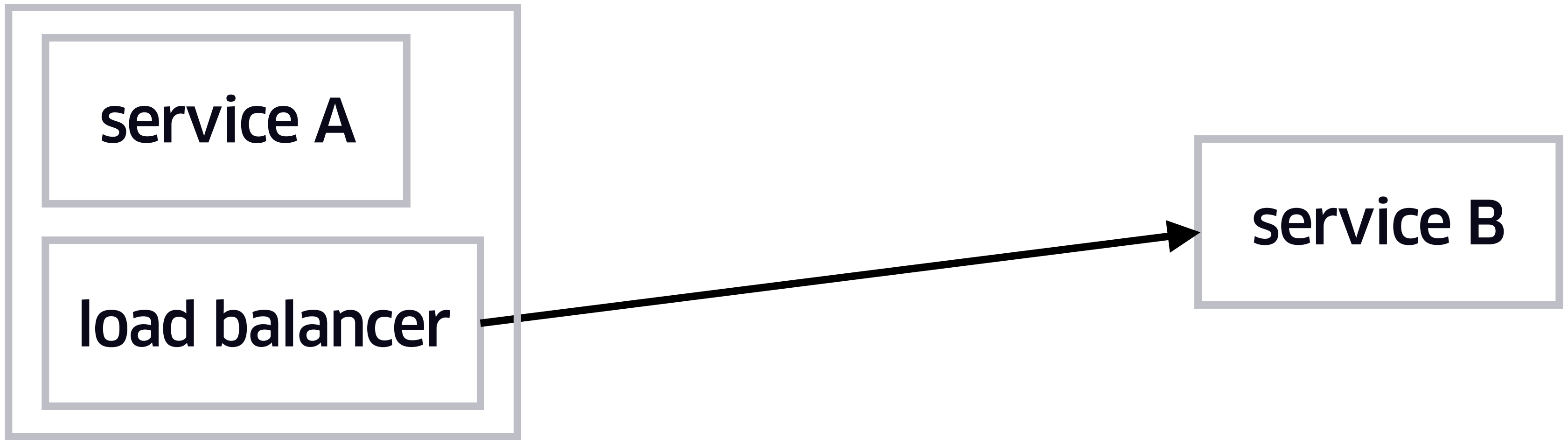
```
Usage: tcap [options]

Options:

  -h, --help                     output usage information
  -V, --version                  output the version number
  -i --interface <interface>     network interface interfaces (defaults to first with an addres
  -p --port <port>               a port to track or use "port1-port2" for a range of ports to t
  -f --filter <filter>           packet filter in pcap-filter(7) syntax (default: all TCP packe
  -s --service <service-name>    service name or names to show (default: all services shown), o
                                 use "~service-name" to exclude the service
  -t --thrift <thrift>           path of the directory for thrift spec files
  -1 --arg1 <arg1-method>        arg1 method or methods to show (default: all arg1 methods show
                                 use "~arg1-method" to exclude the arg1
  --m1                           show arg1 name in call responses
```



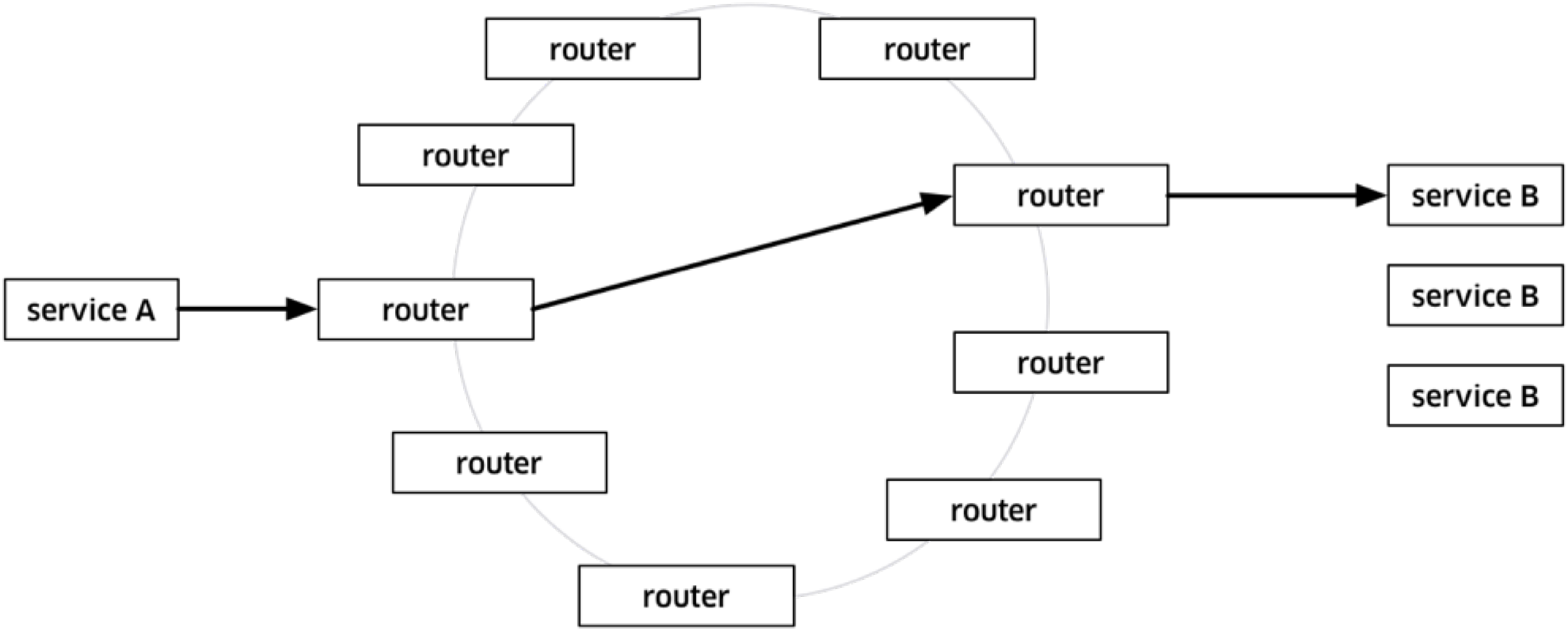
service A

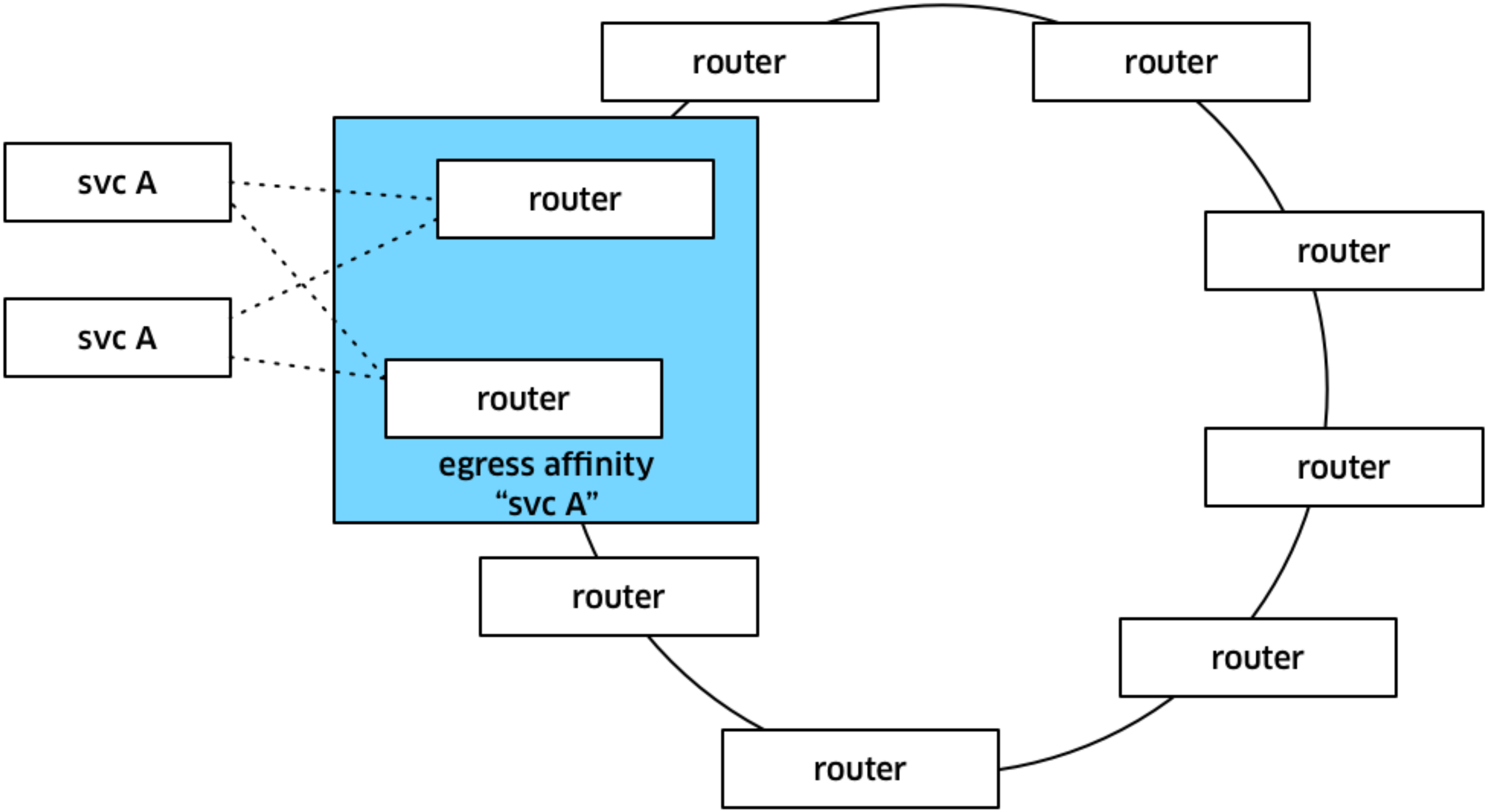
load balancer

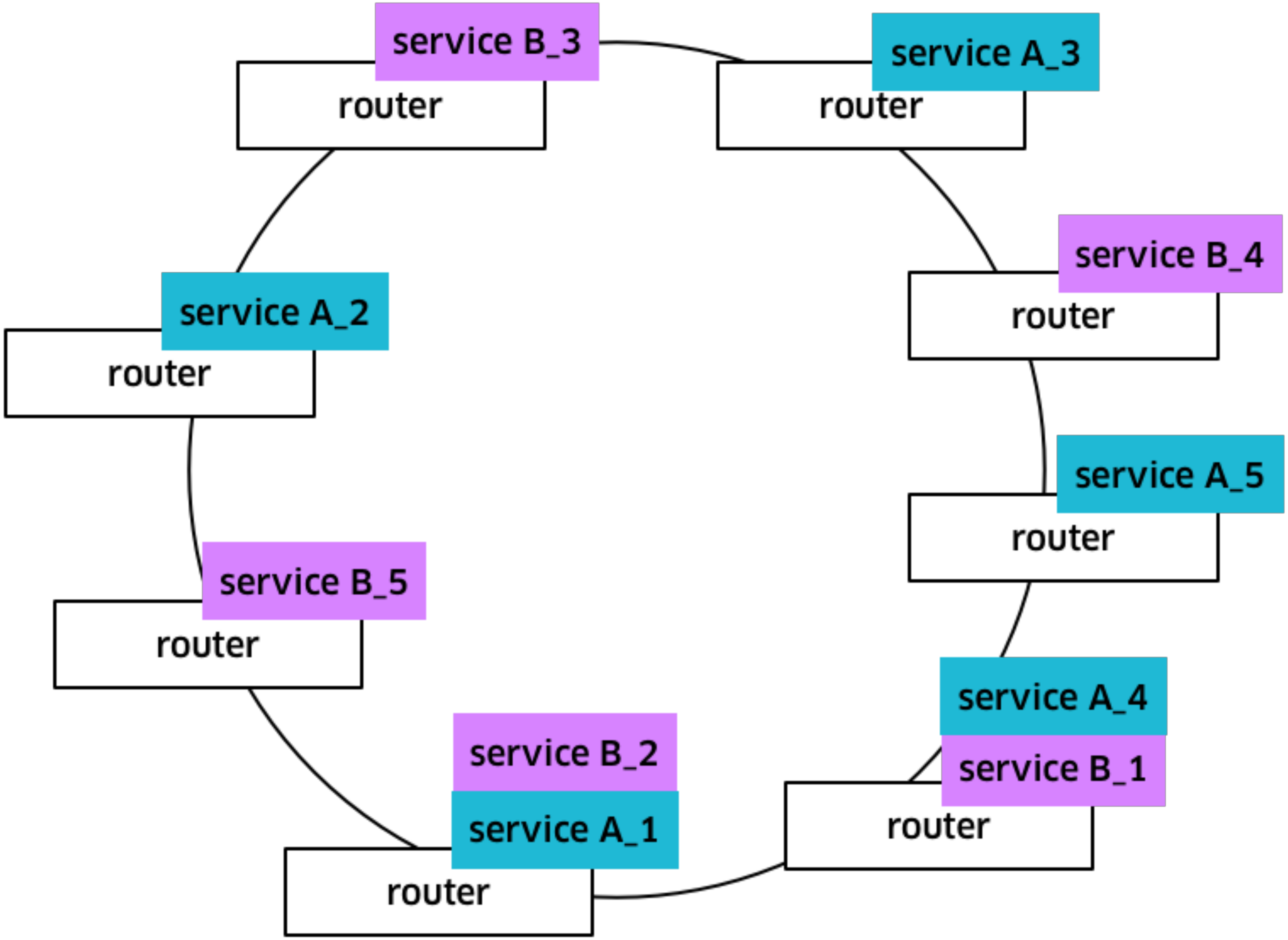
service B

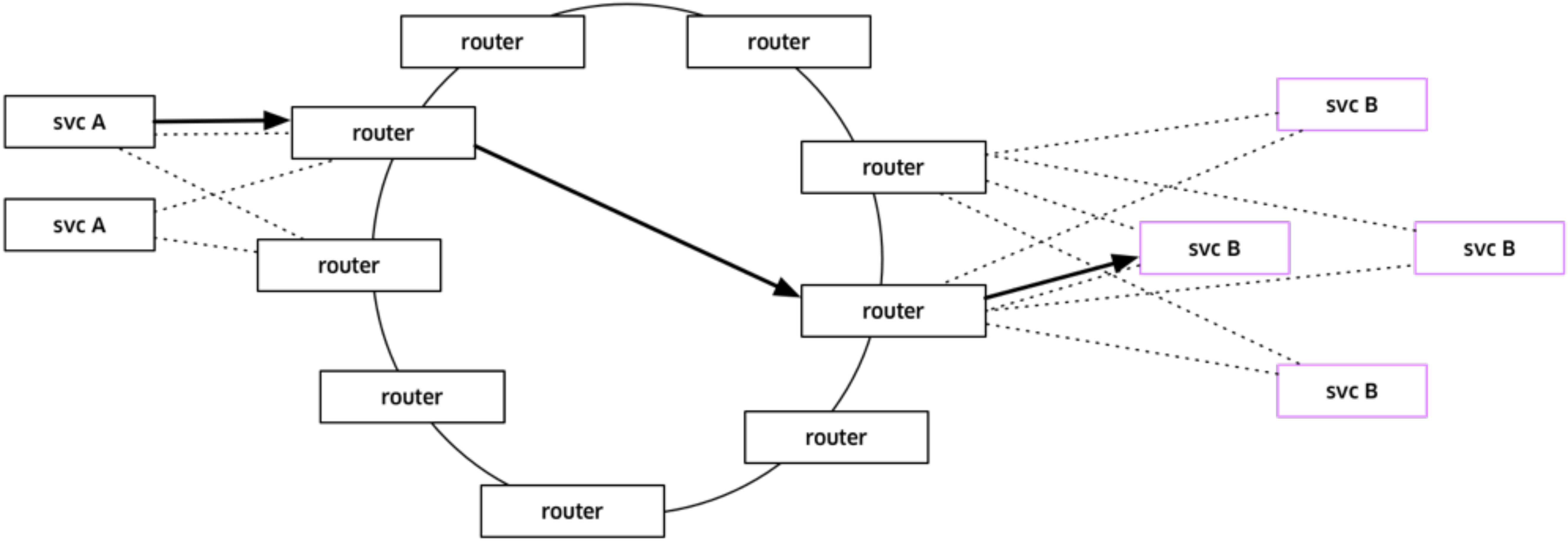

service A
load balancer
service B

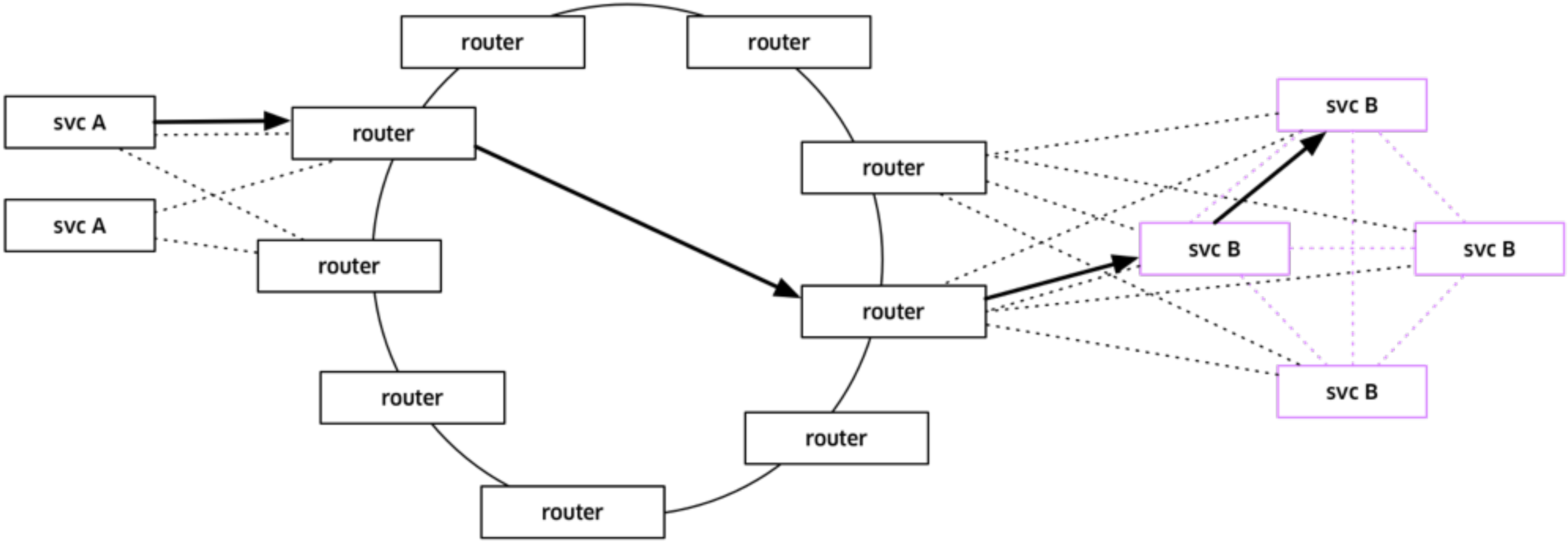
HYPERBAHN

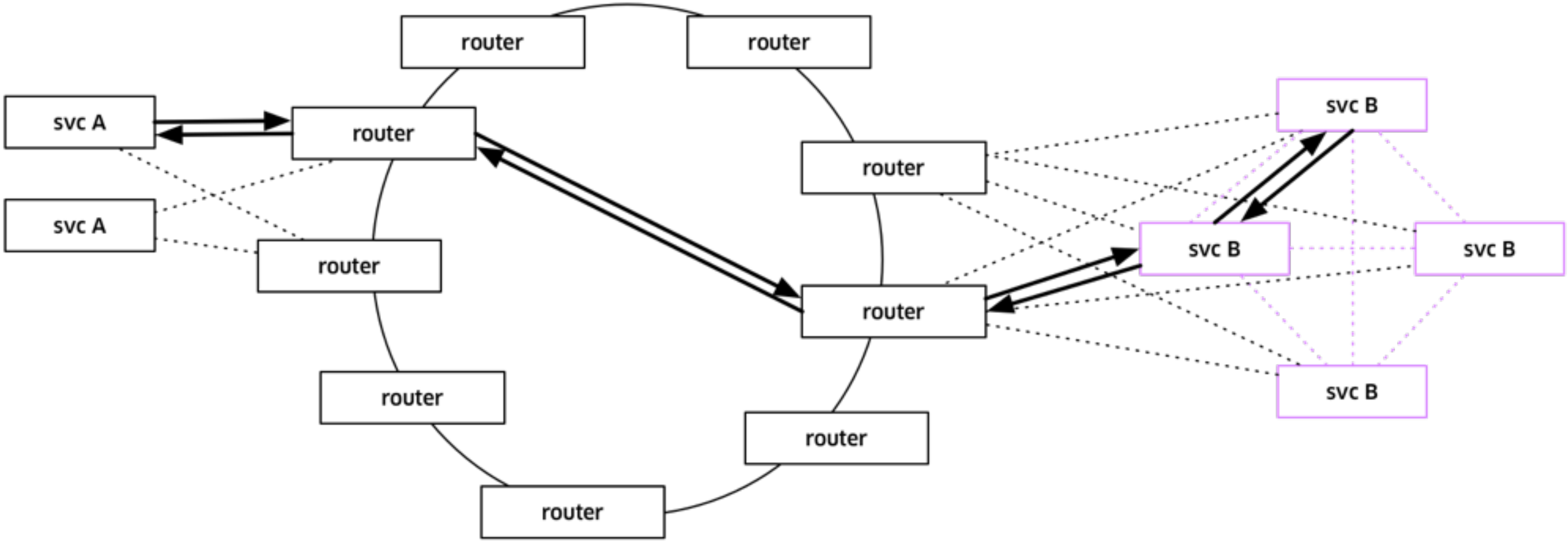

router
router
router
router
service B
service A
router
service B
router
service B
router
router
router


router
router
svc A
router
router
svc A
router
router
egress affinity
"svc A"
router
router
router
router


service B_3
router
service A_3
router
service B_4
router
service A_2
router
service A_5
router
service B_5
router
service A_4
service B_1
router
service B_2
service A_1
router


svc A
svc A
router
router
router
router
router
router
router
router
router
svc B
svc B
svc B
svc B


svc A
svc A
router
router
router
router
router
router
router
router
router
svc B
svc B
svc B
svc B


router
router
svc A
router
router
svc B
svc A
router
svc B
svc B
router
router
svc B
router
router

HYPERBAHN

scalable registry and health checks

zipkin tracing

circuit breaking

rate limiting

failure testable

AVAILABILITY

everything retryable

everything killable

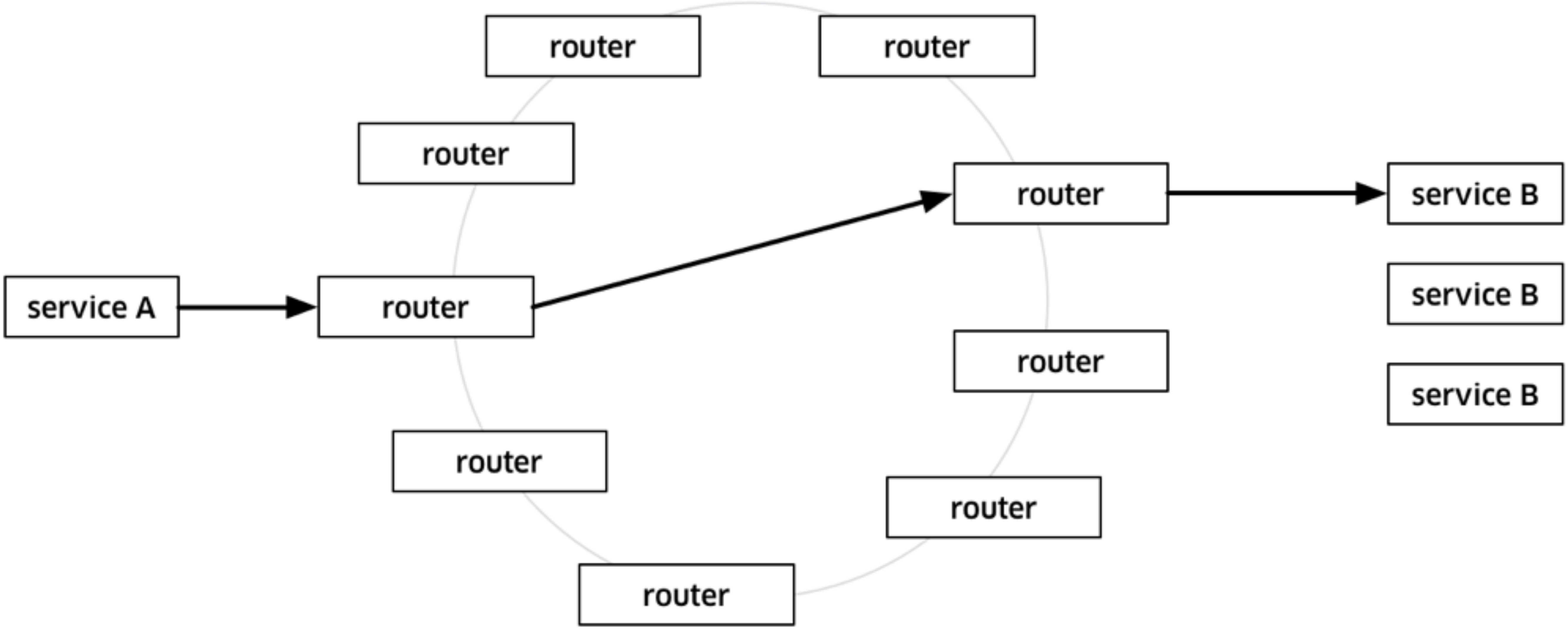
crash only

CULTURAL CHANGES

no pairs

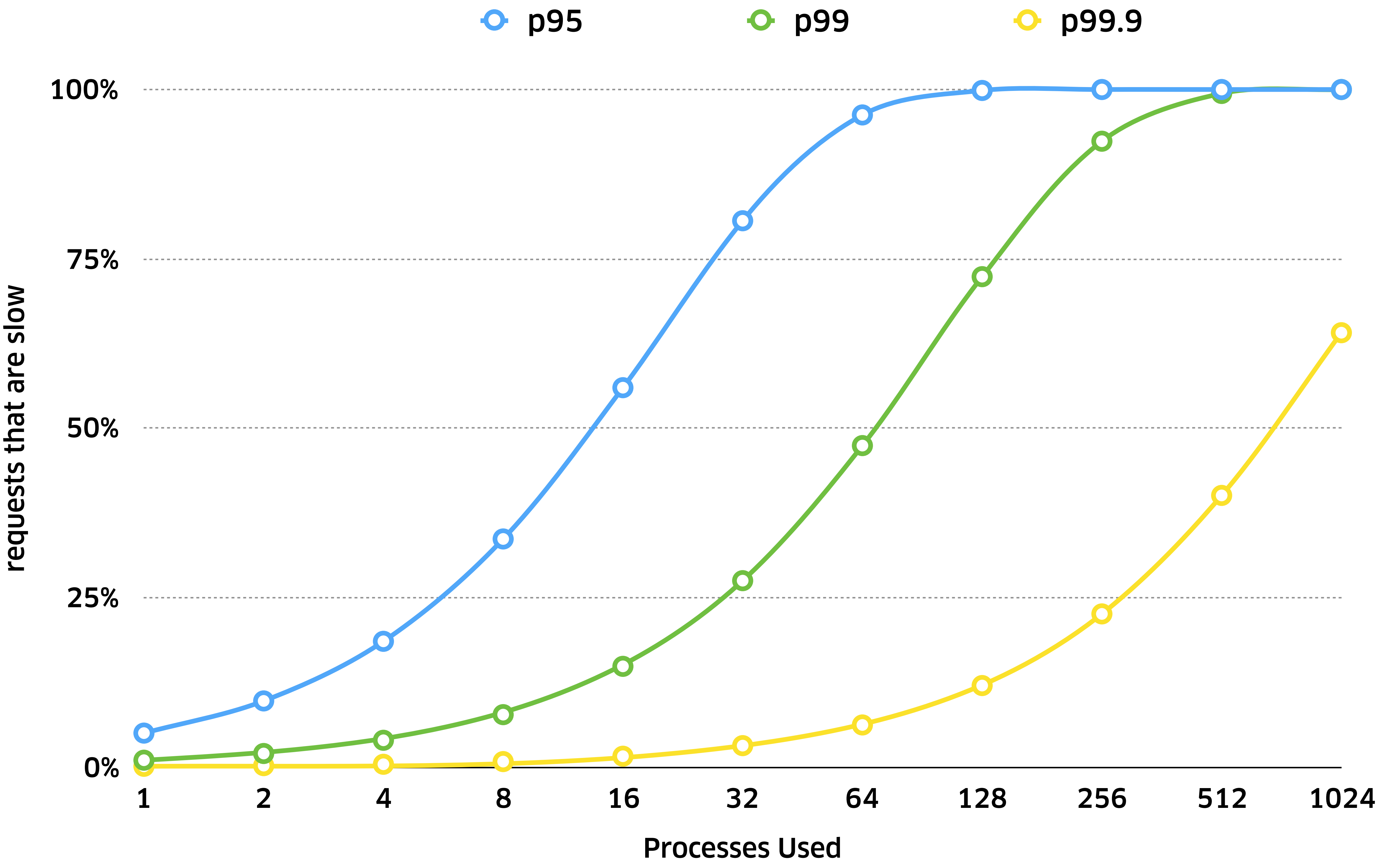
kill everything

even databases


router
router
router
router
service B
service A
router
service B
router
service B
router
router
router

LATENCY

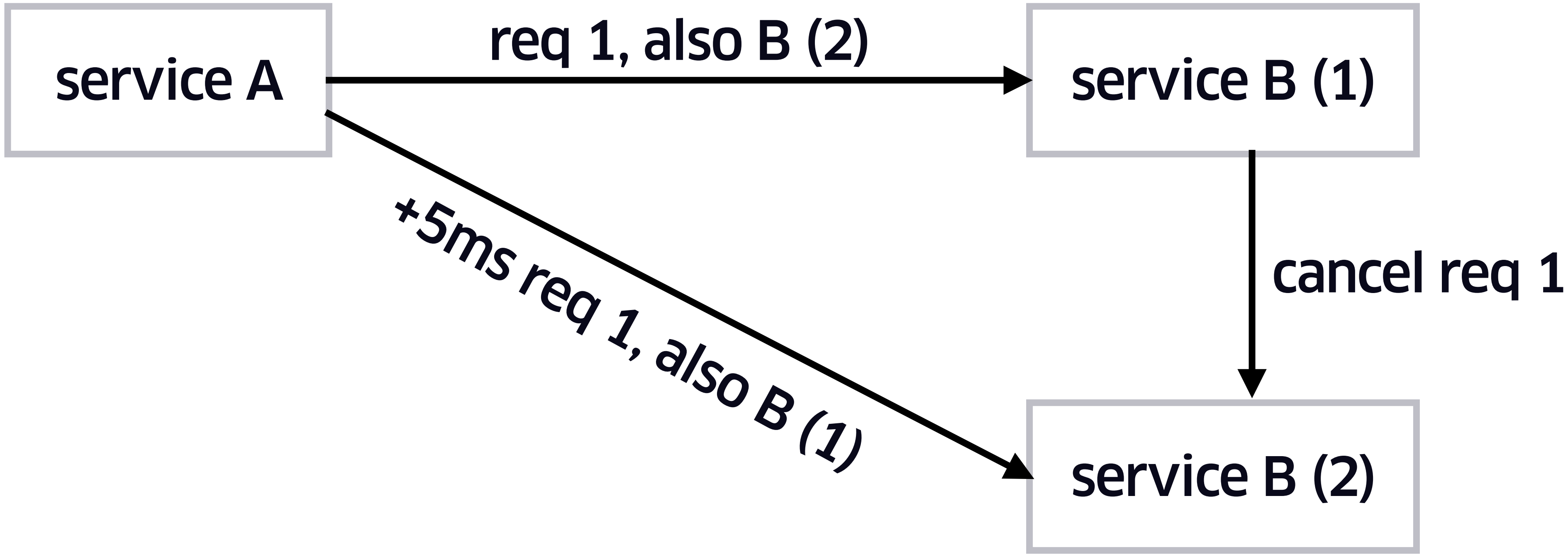
- overall latency ≥ latency of slowest component
- 1ms avg, 1000ms p99
  - use 1: 1% at least 1000ms
  - use 100: 63% at least 1000ms
  - 1.0 - 0.99^100 = 0.634 = 63.4%


p95
p99
p99.9
100%
75%
50%
25%
0%
requests that are slow
1
2
4
8
16
32
64
128
256
512
1024
Processes Used

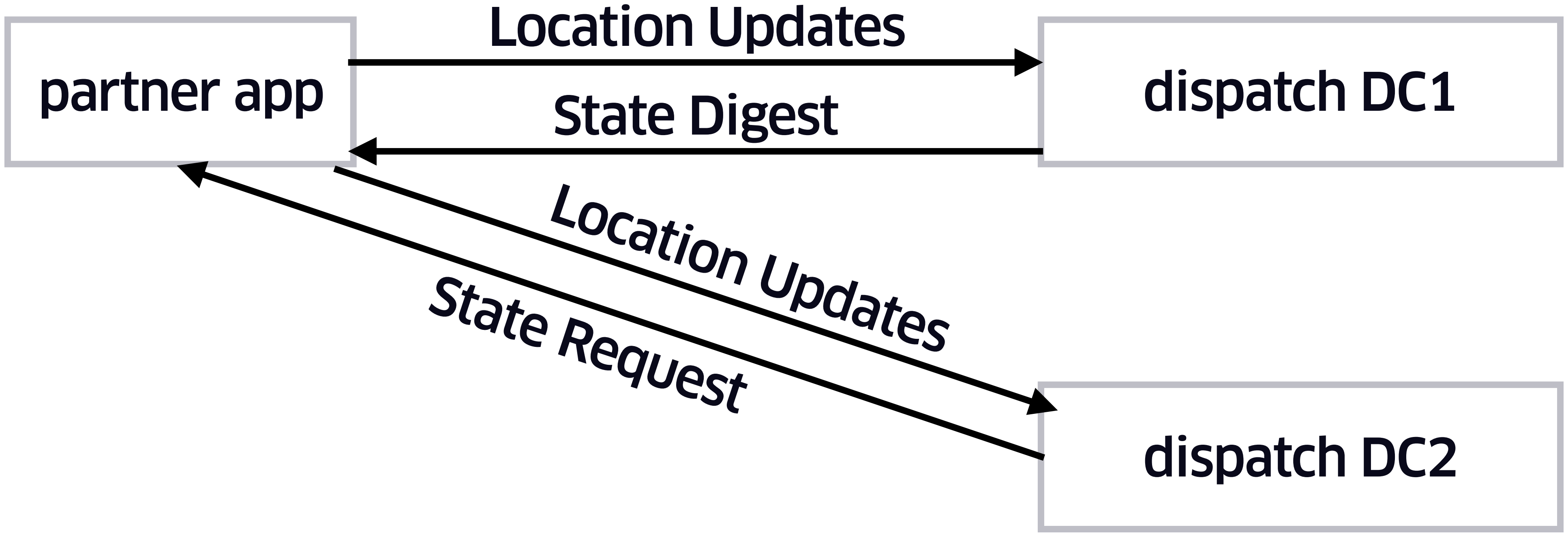
Google™

Achieving Rapid Response Times in Large Online Services

Jeff Dean
Google Fellow
jeff@google.com


service A
req 1, also B (2)
service B (1)
cancel req 1
+5ms req 1, also B (1)
service B (2)

DATACENTER FAILURE



partner app

dispatch DC1

dispatch DC2

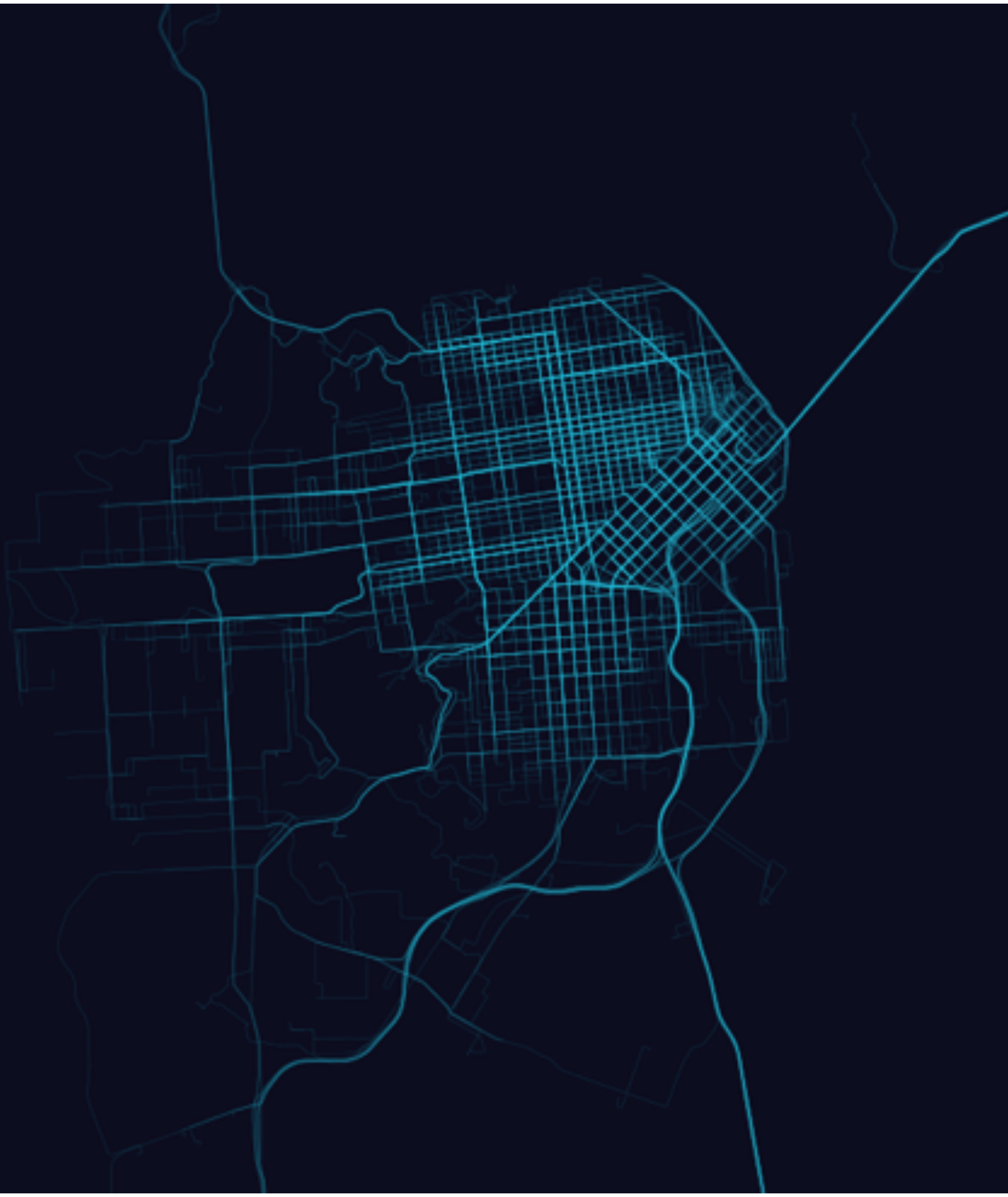
Location Updates

State Digest

Location Updates

State Request

THANKS



UBER