

Building Microservice Architectures with Go

Matt Heath, Mondo

#yow2015



@mattheath



mondo



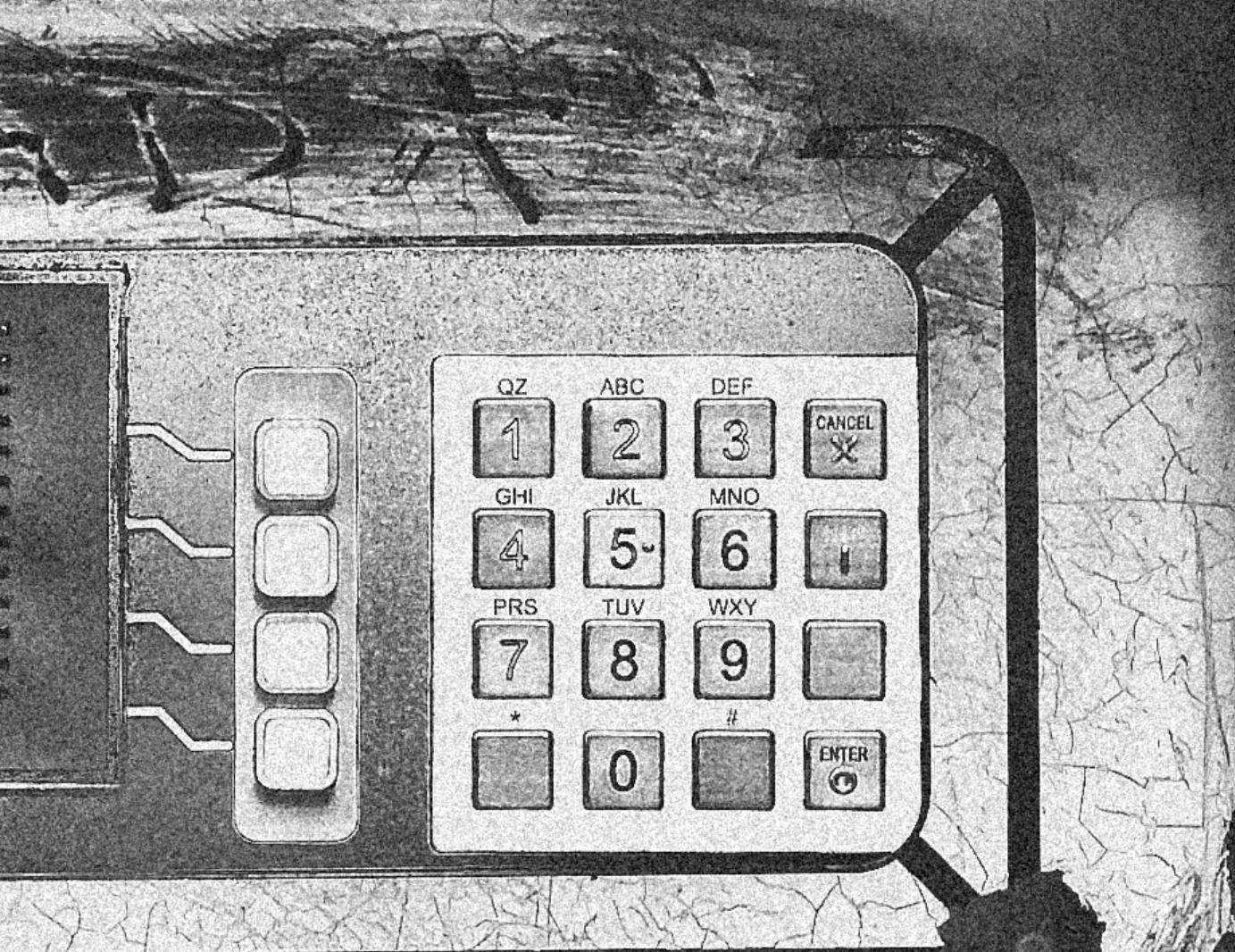
ATM

VISA

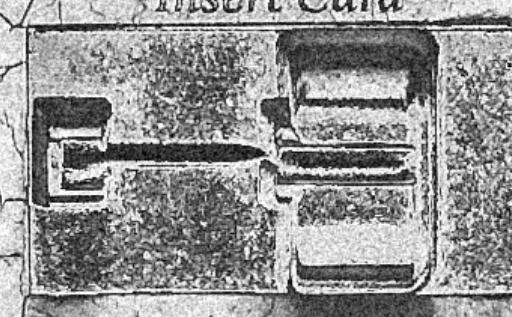
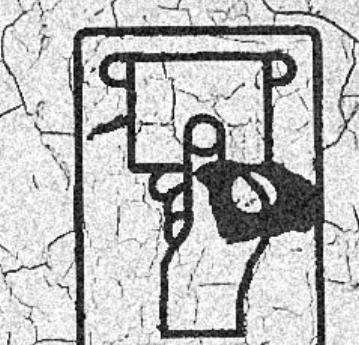
MasterCard



Maestro



To report suspicious activity or operational problems please contact:
ATM Help Desk
1-888-486-4949
PLUS transactions are sponsored by
RBS Citizens N.A.



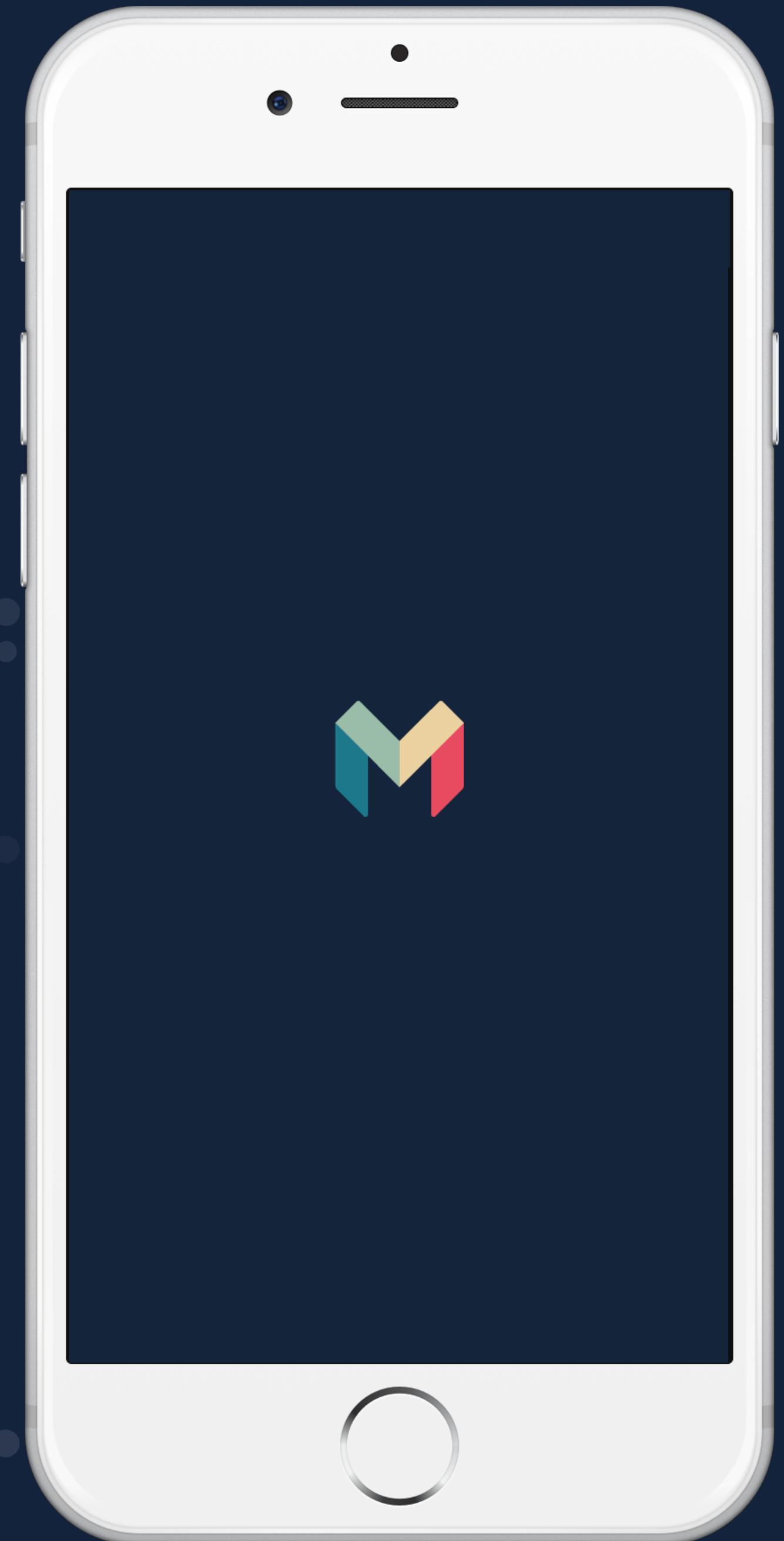
Rece...

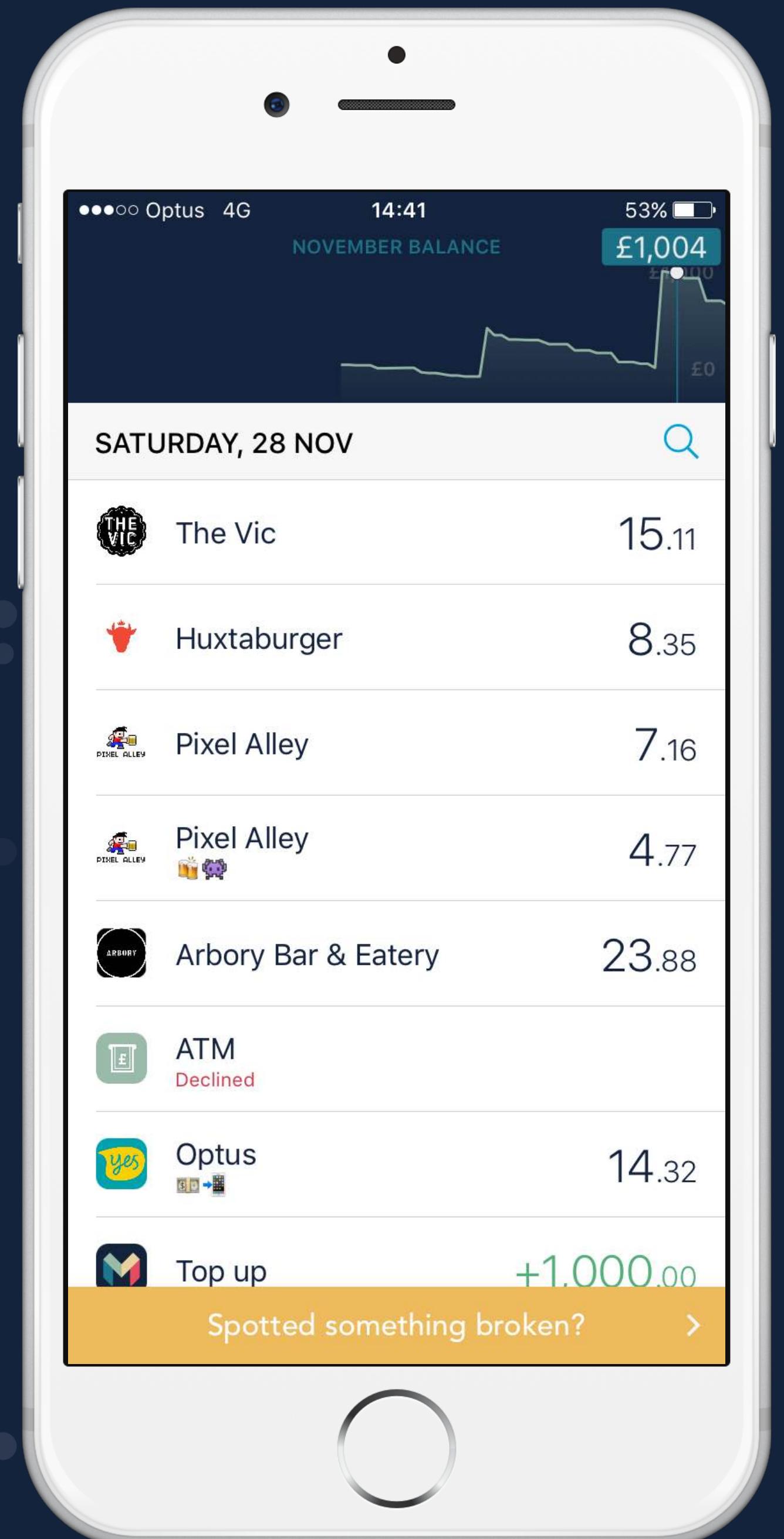
1895

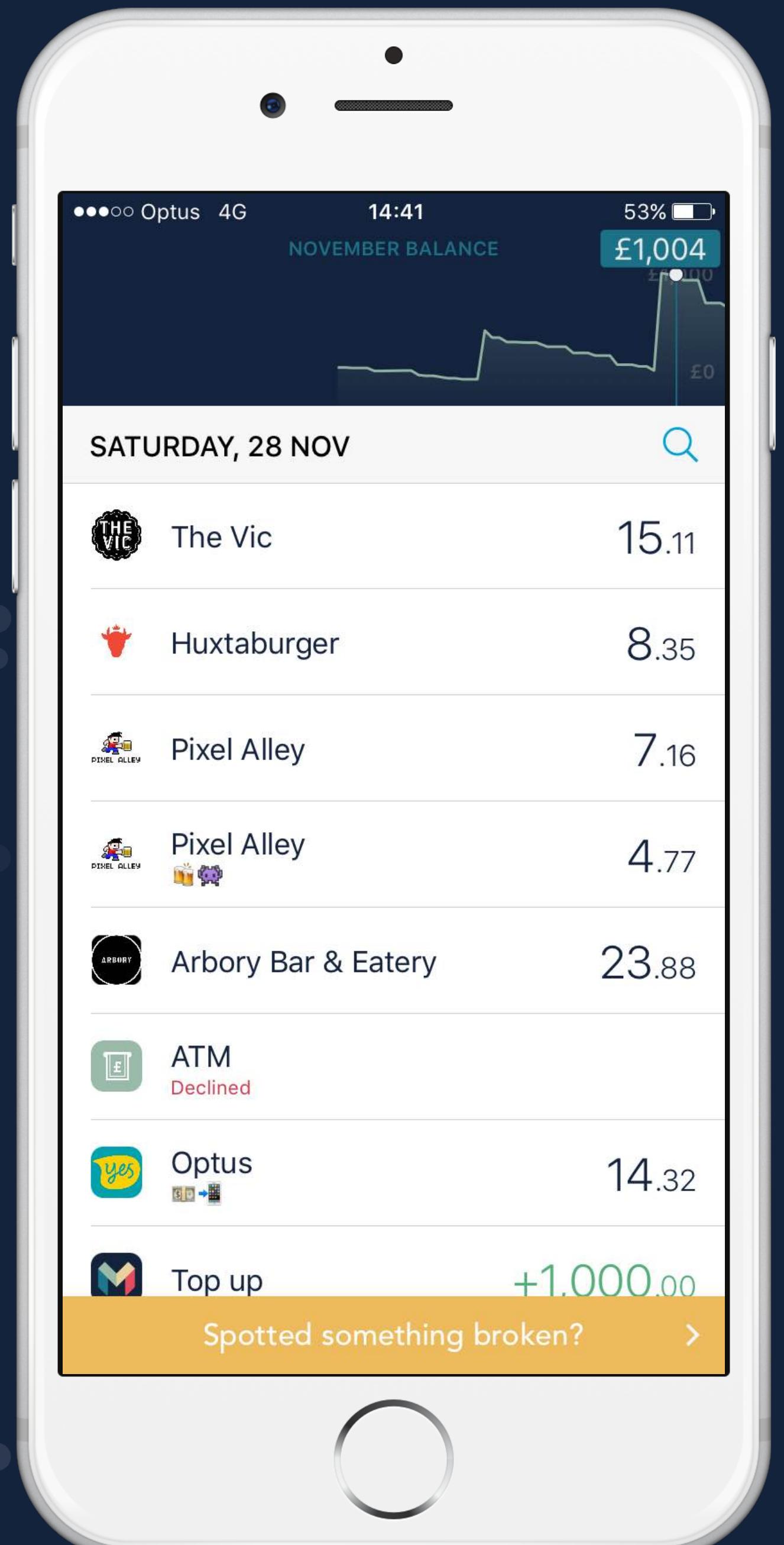
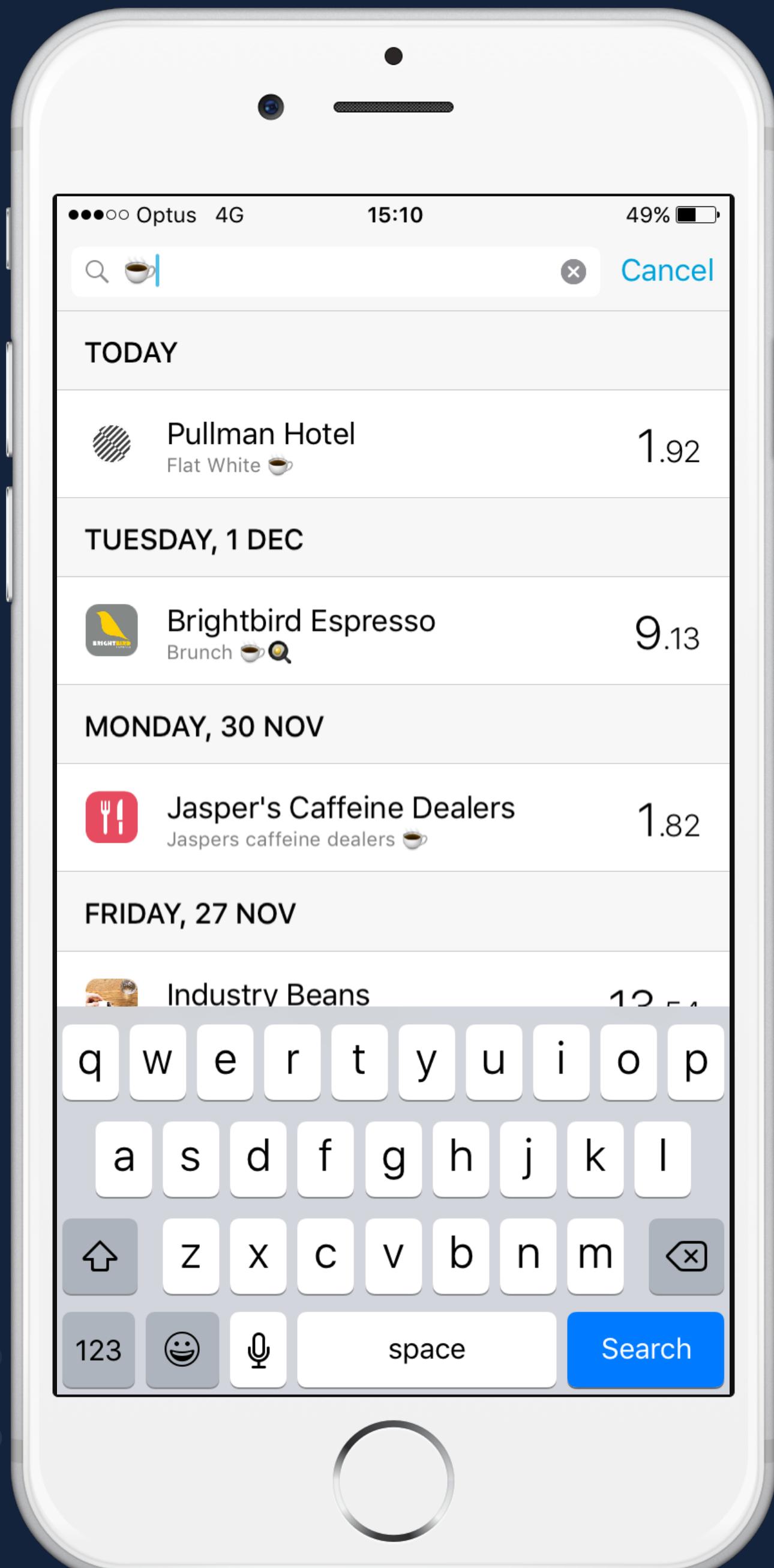


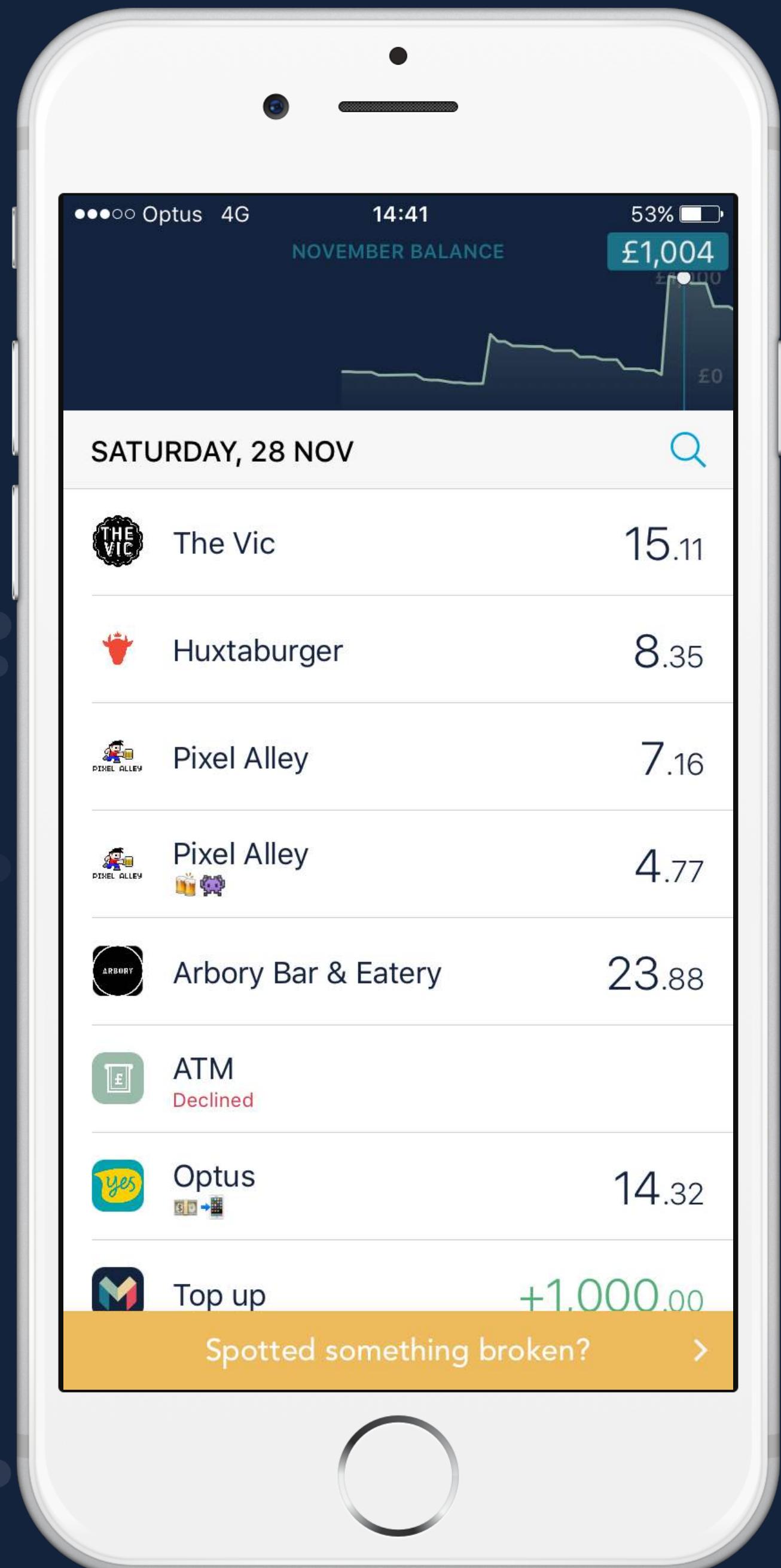
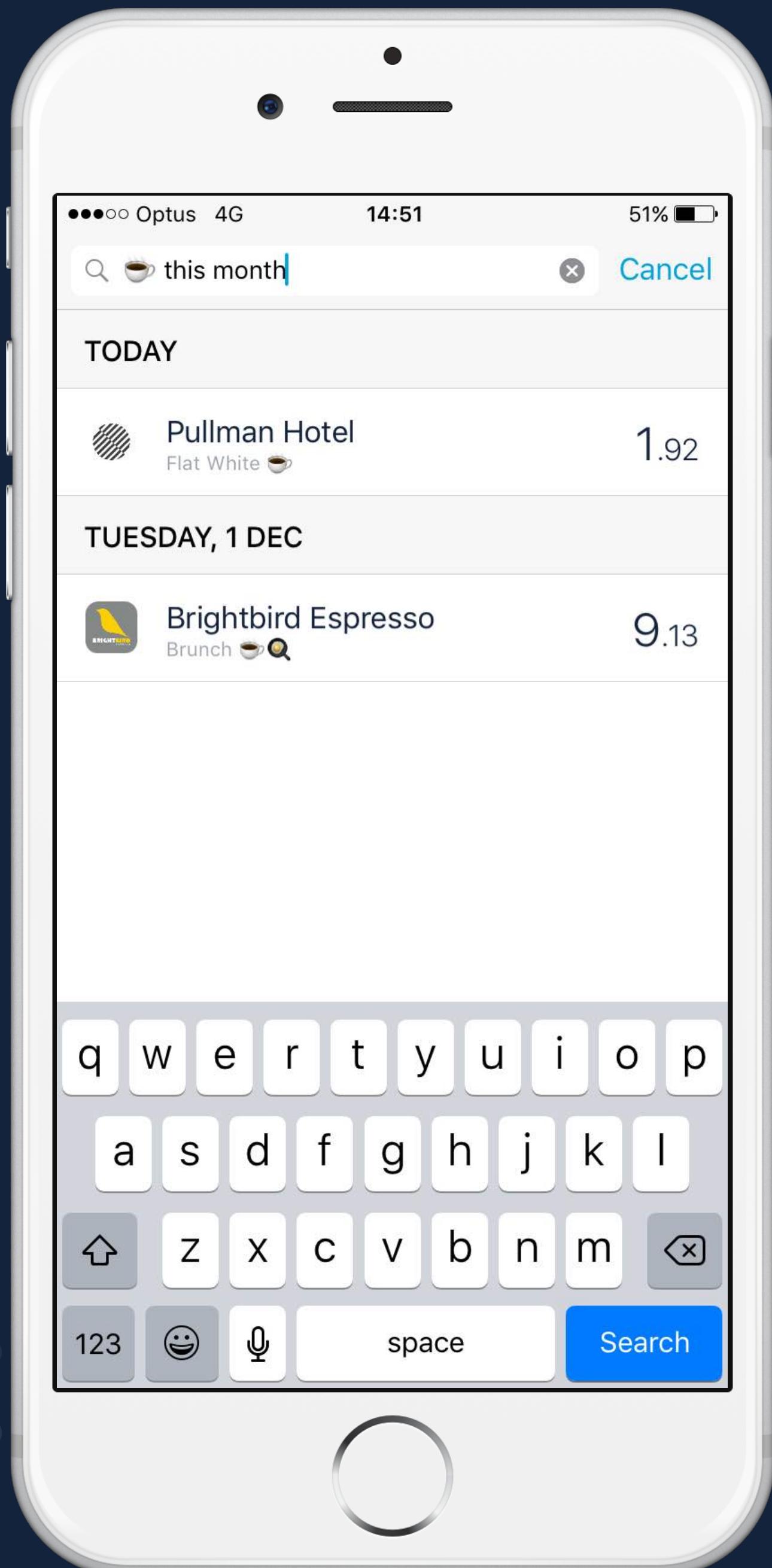


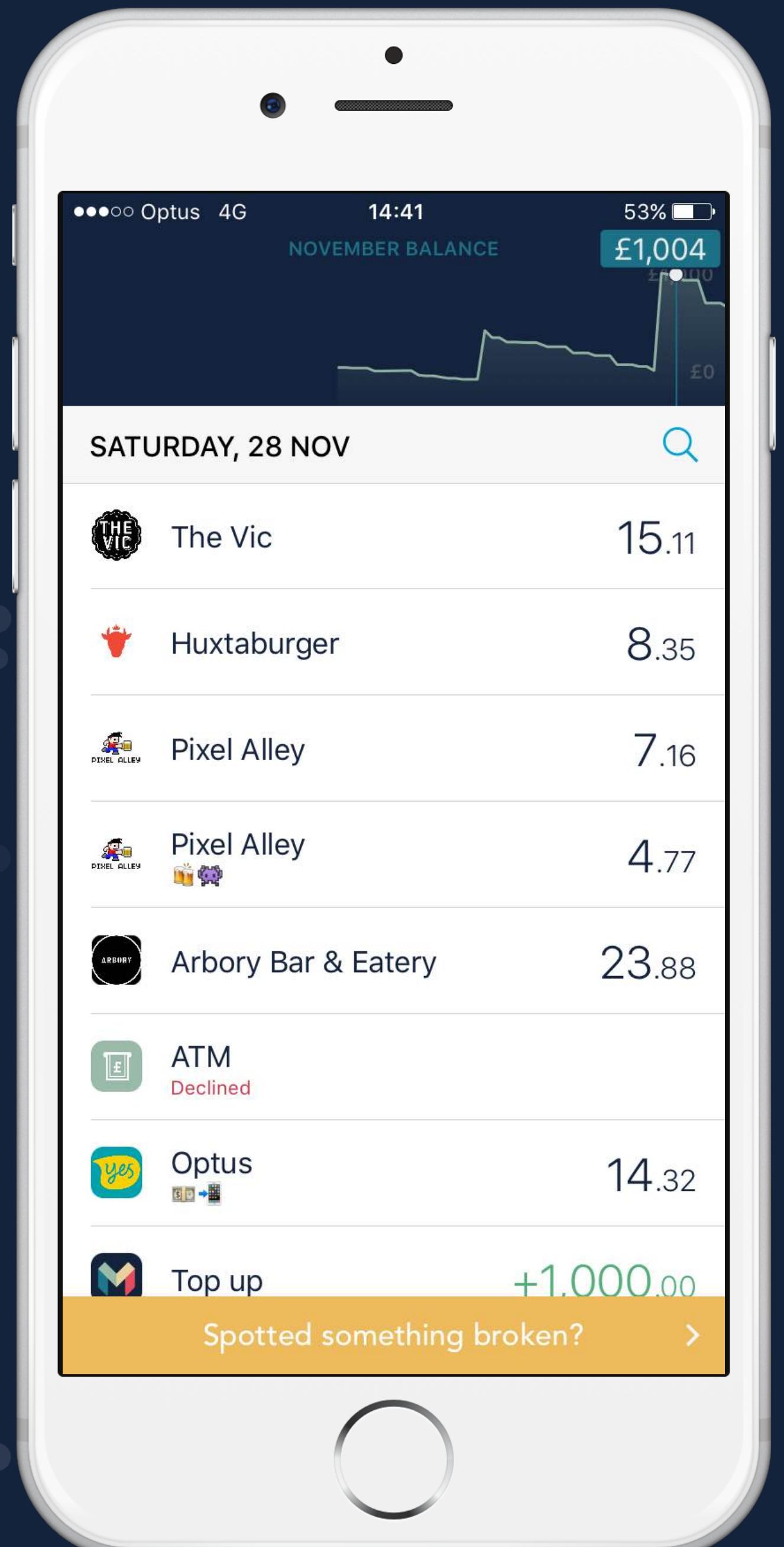
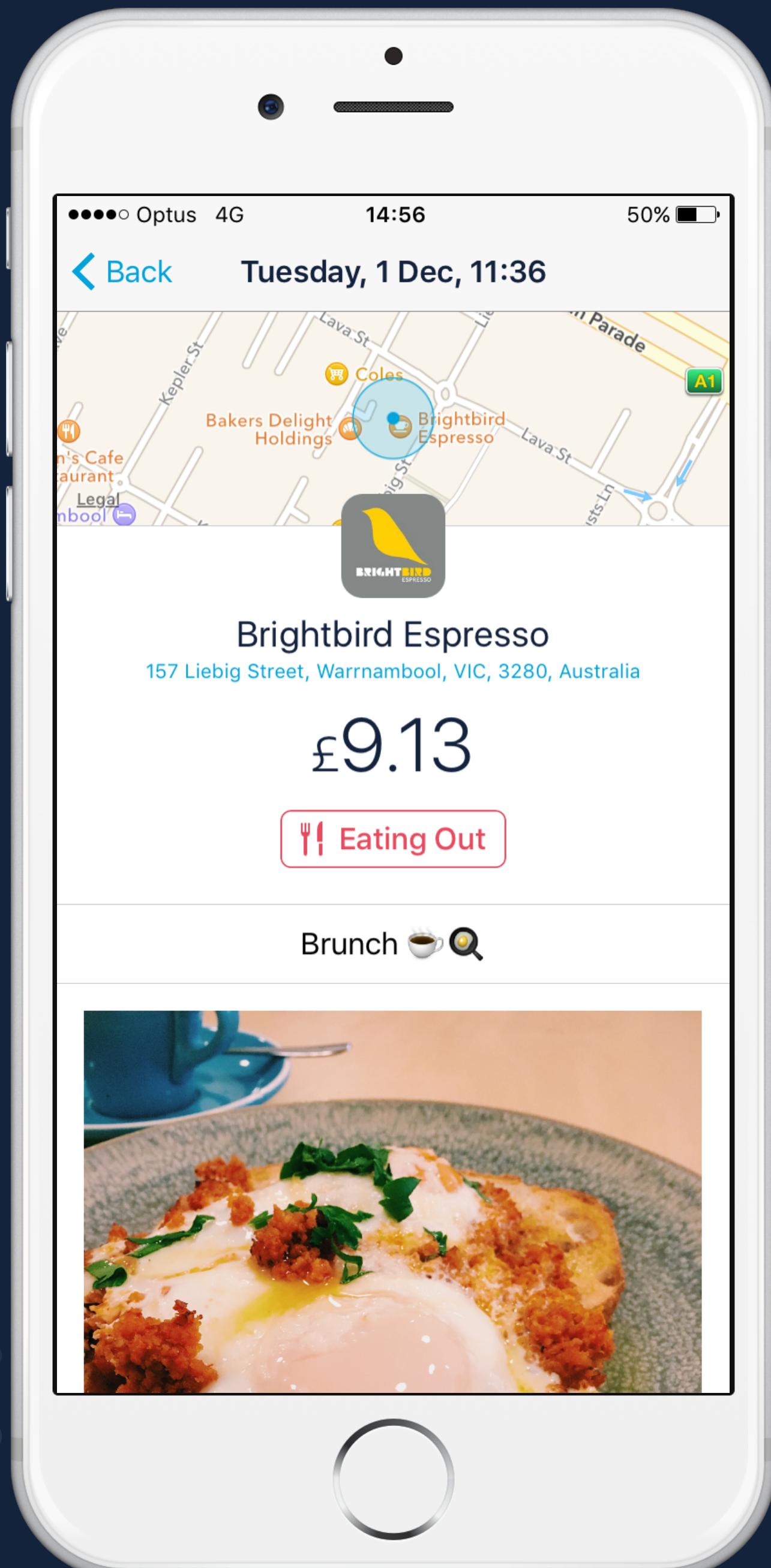


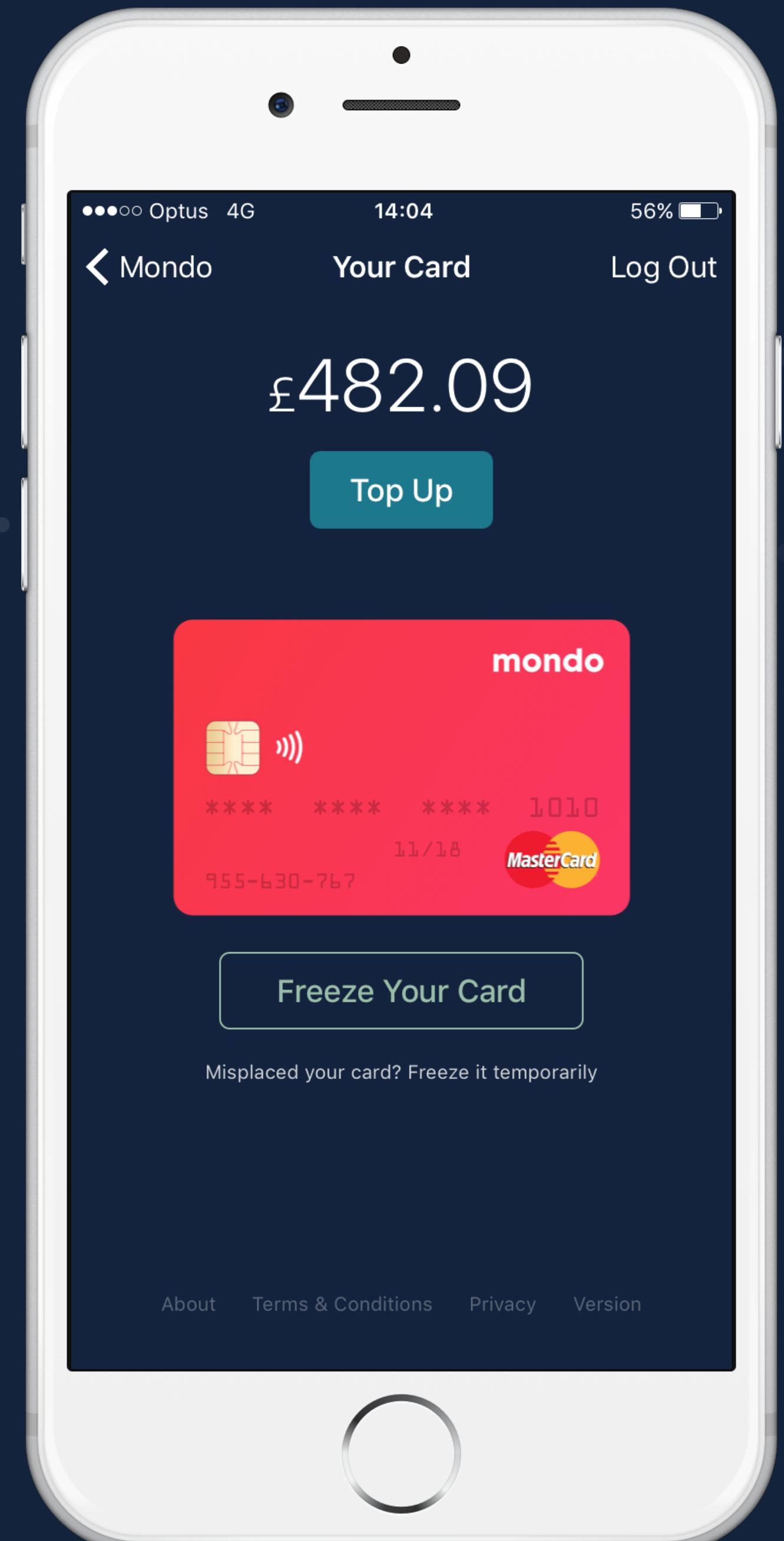
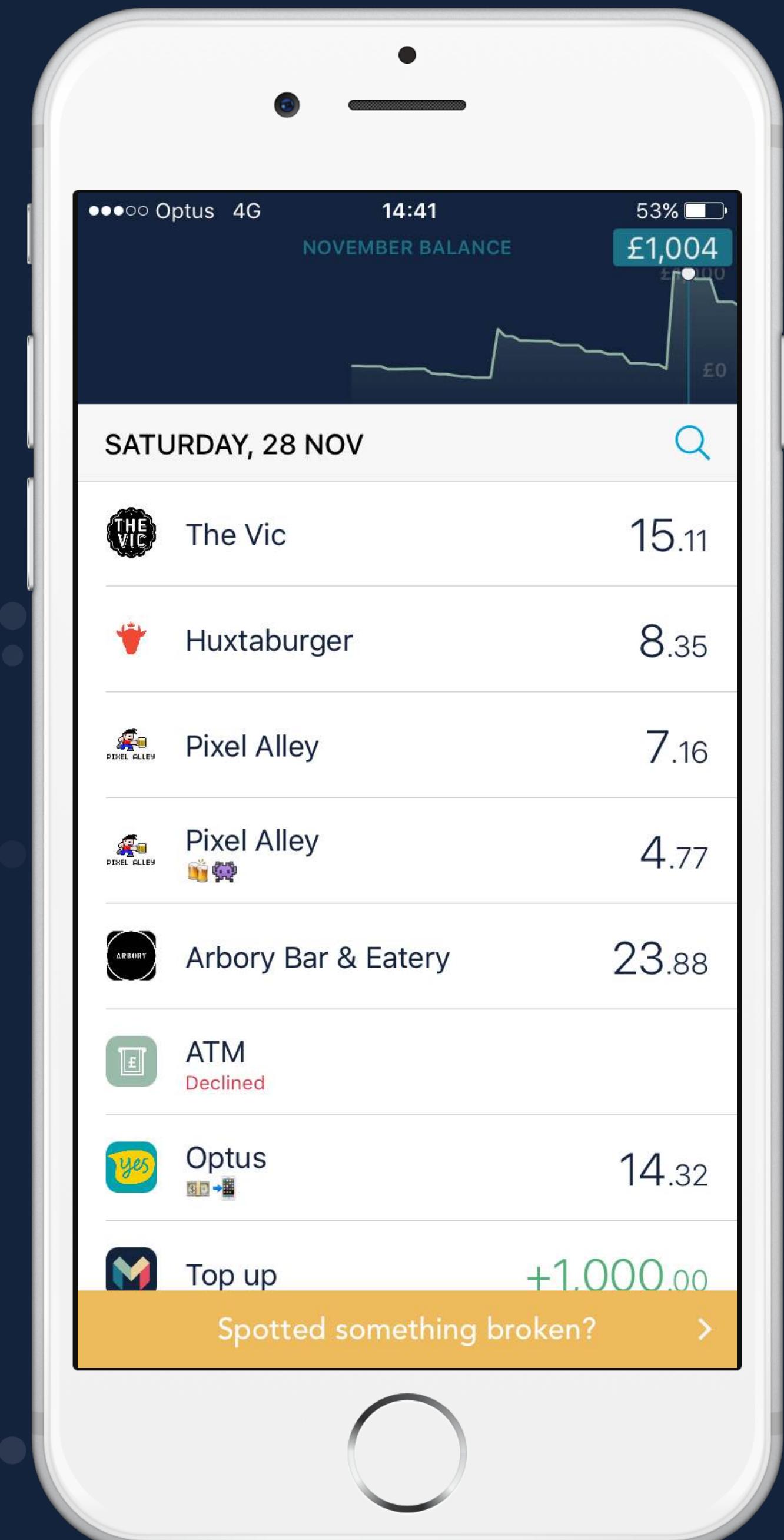
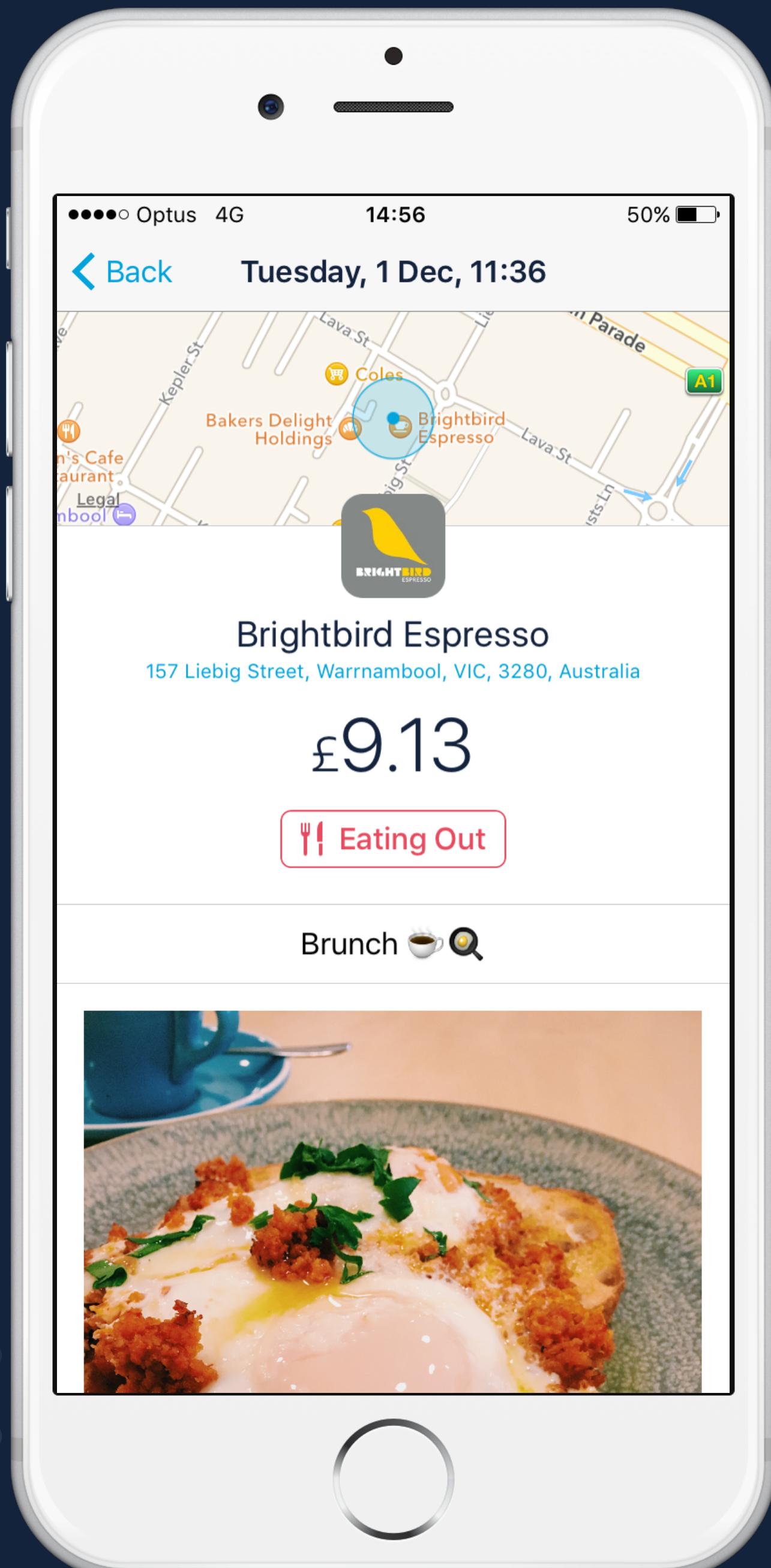


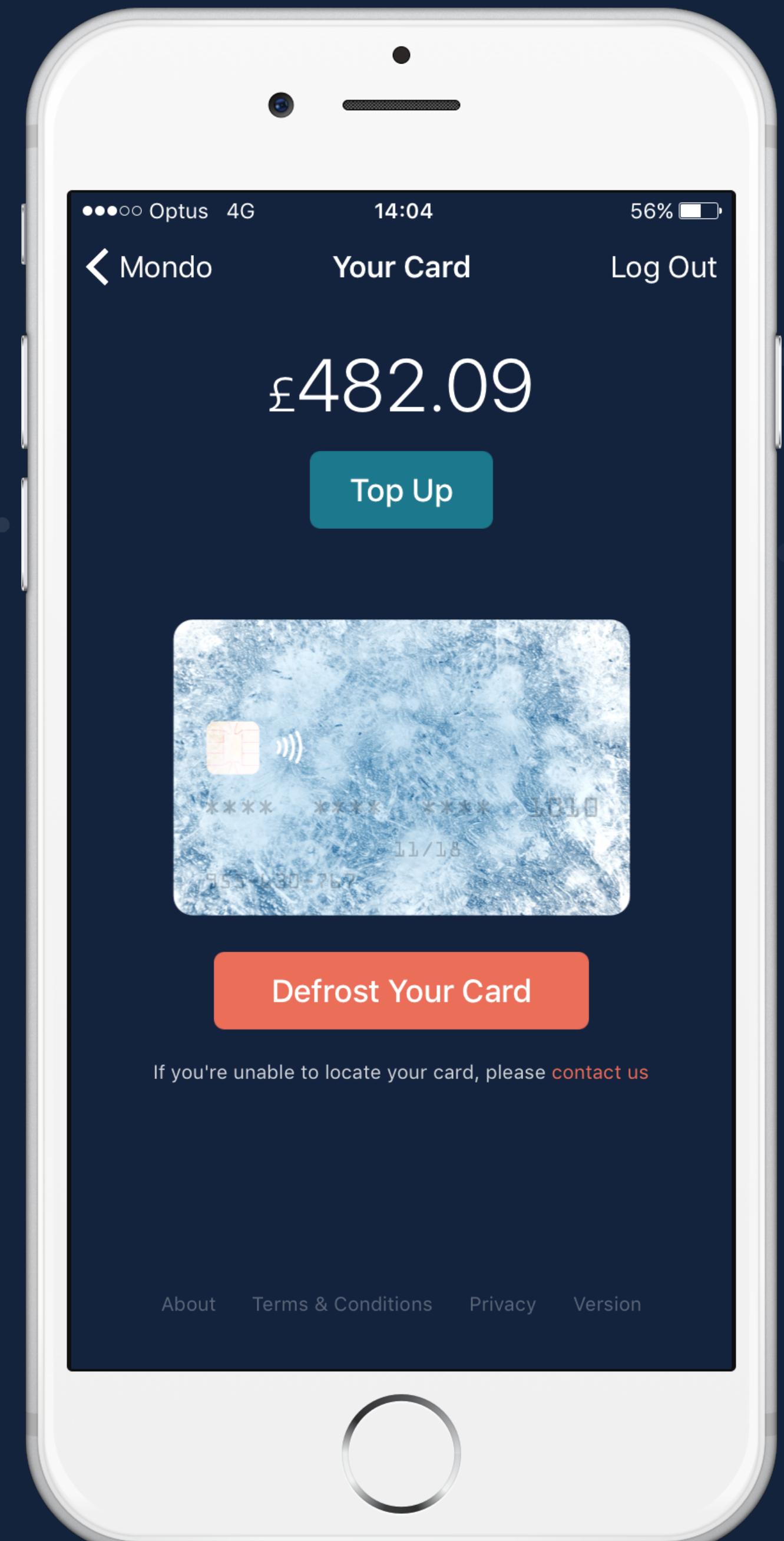
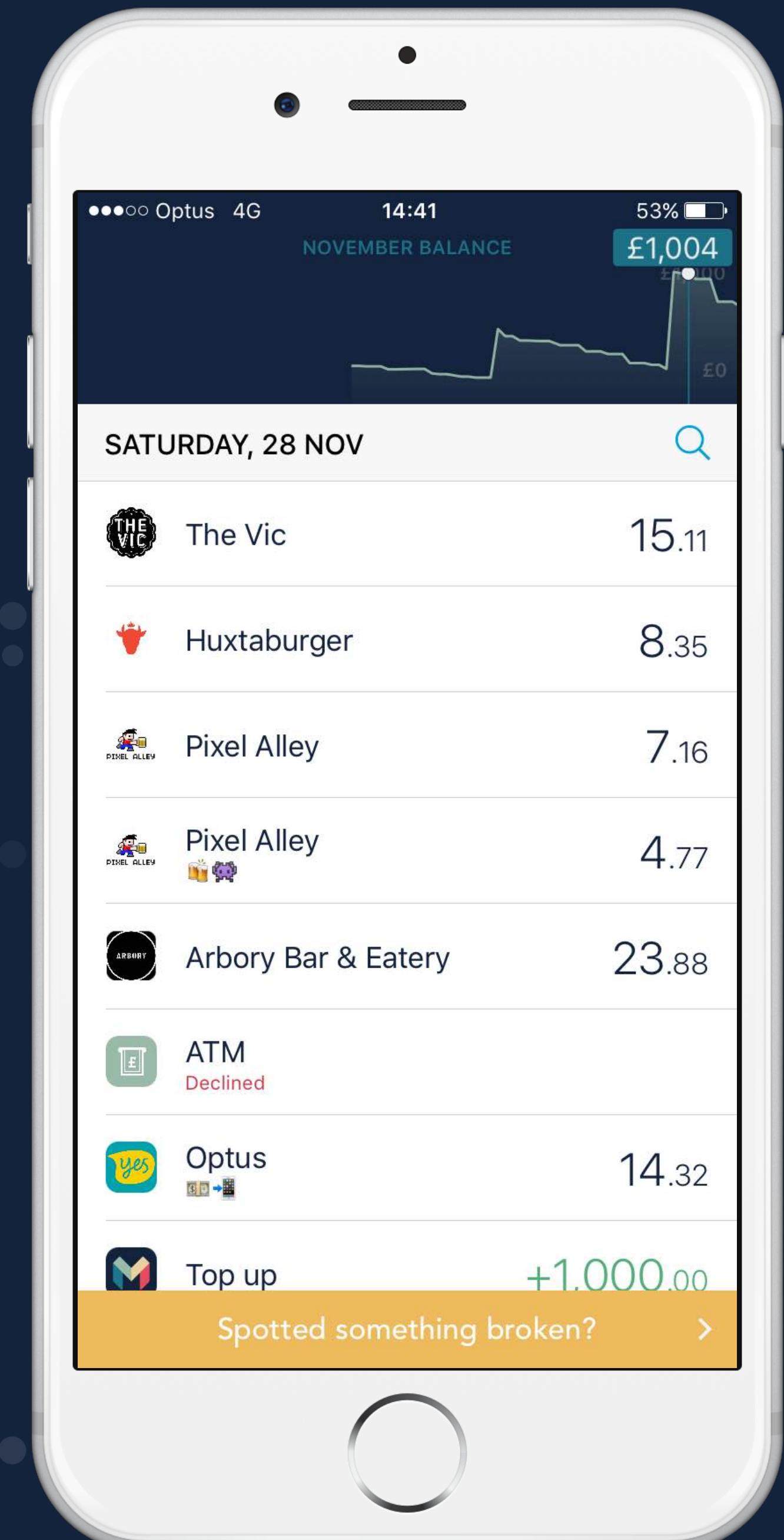
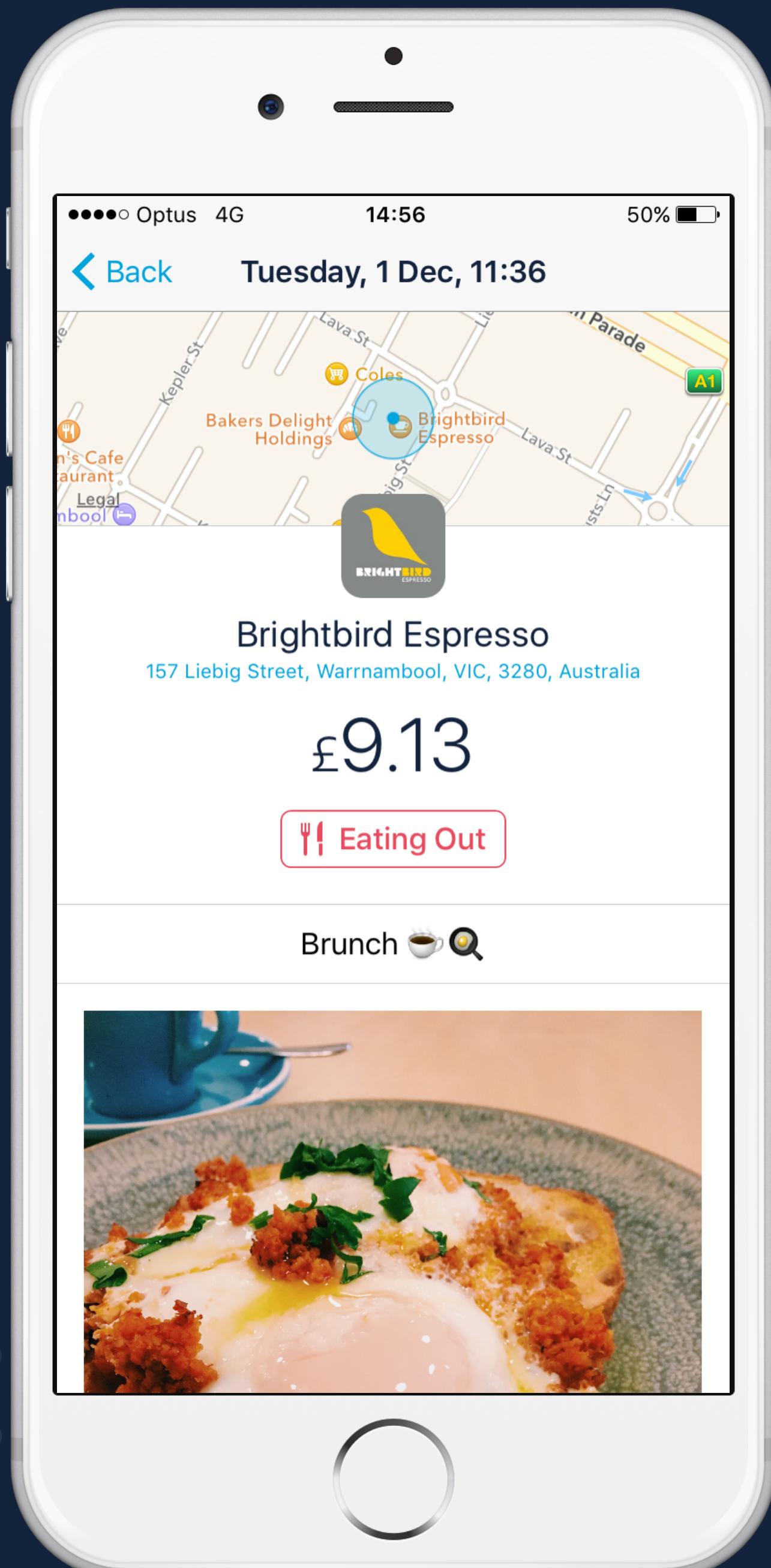


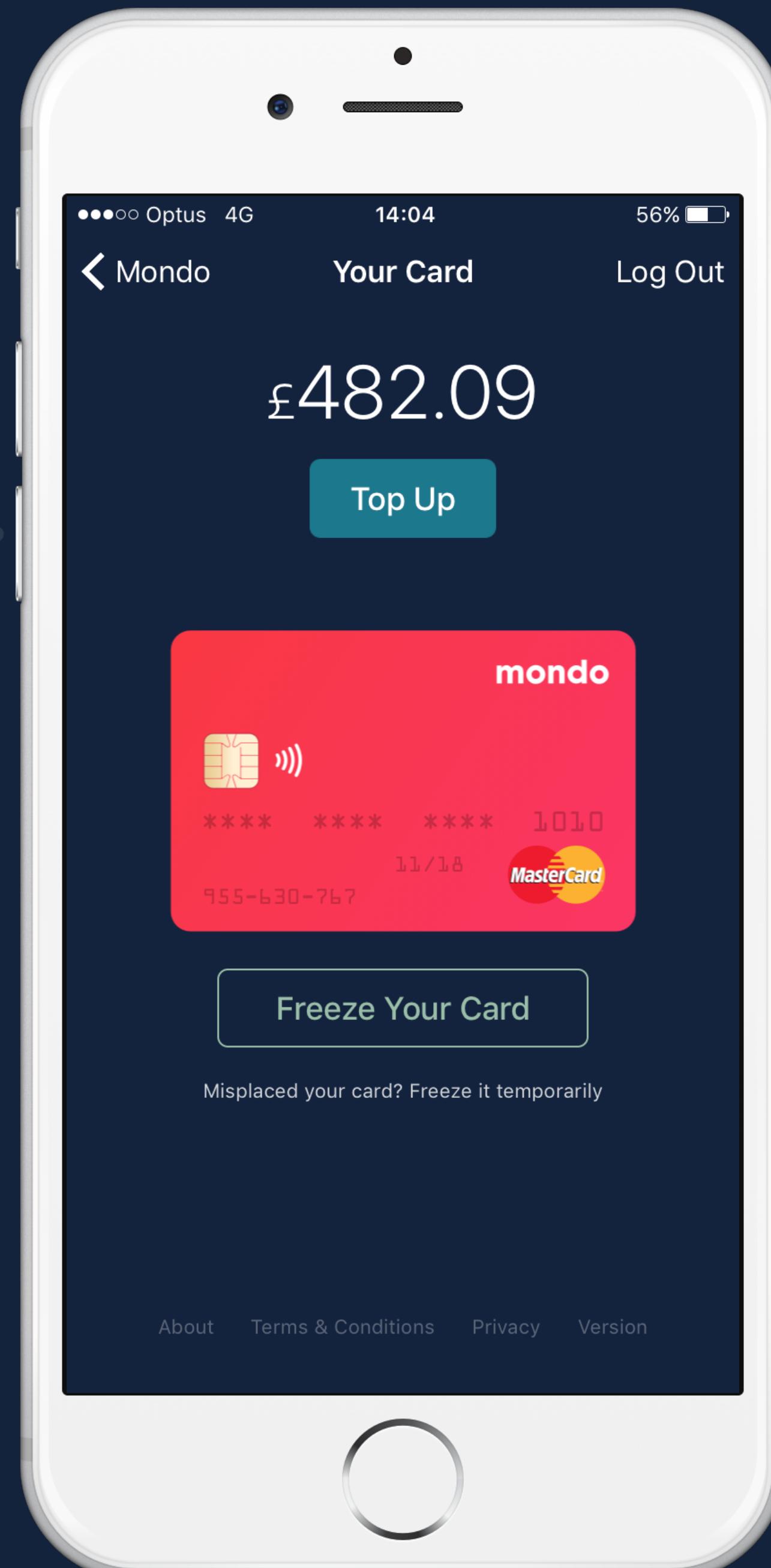
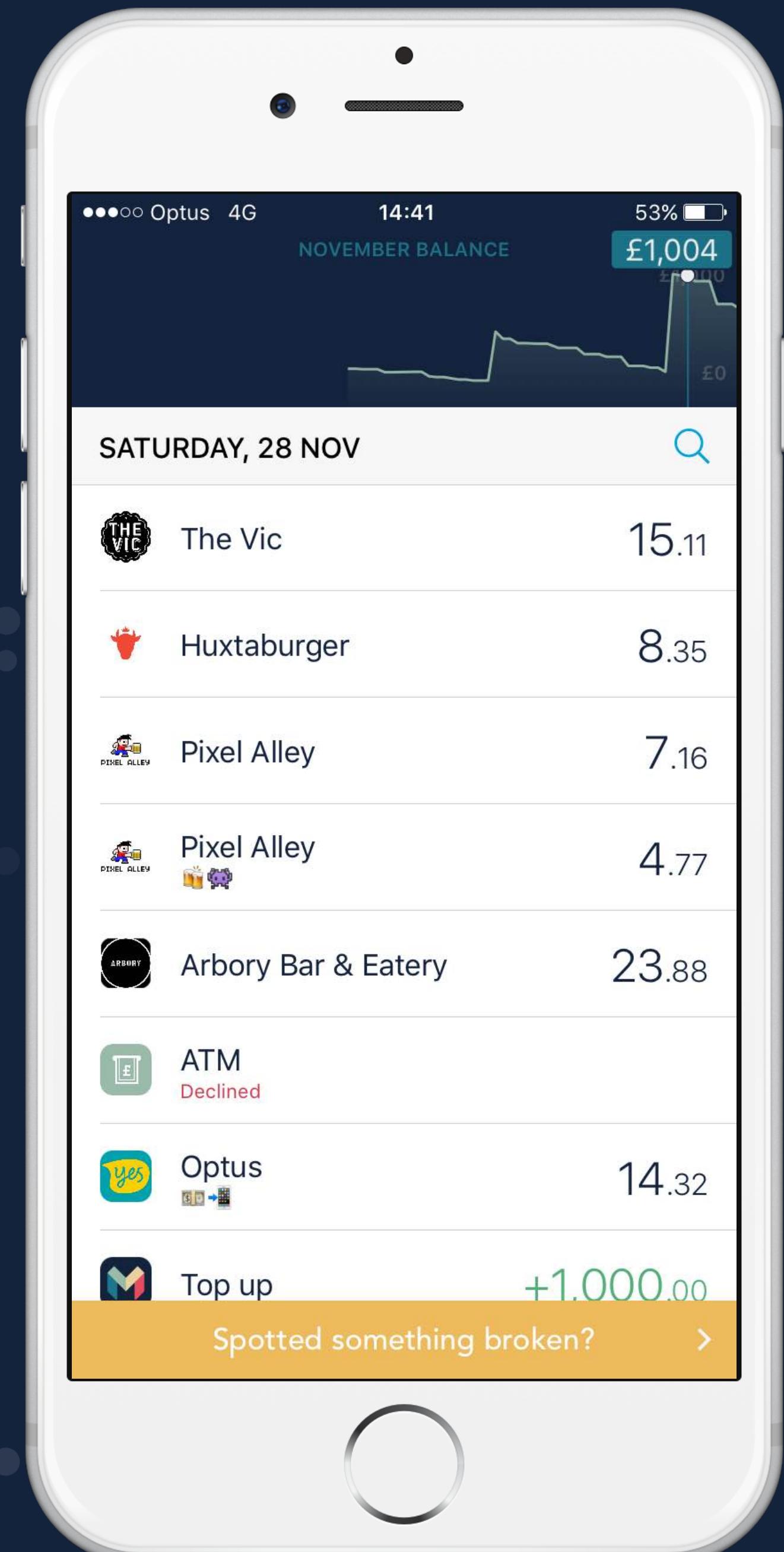
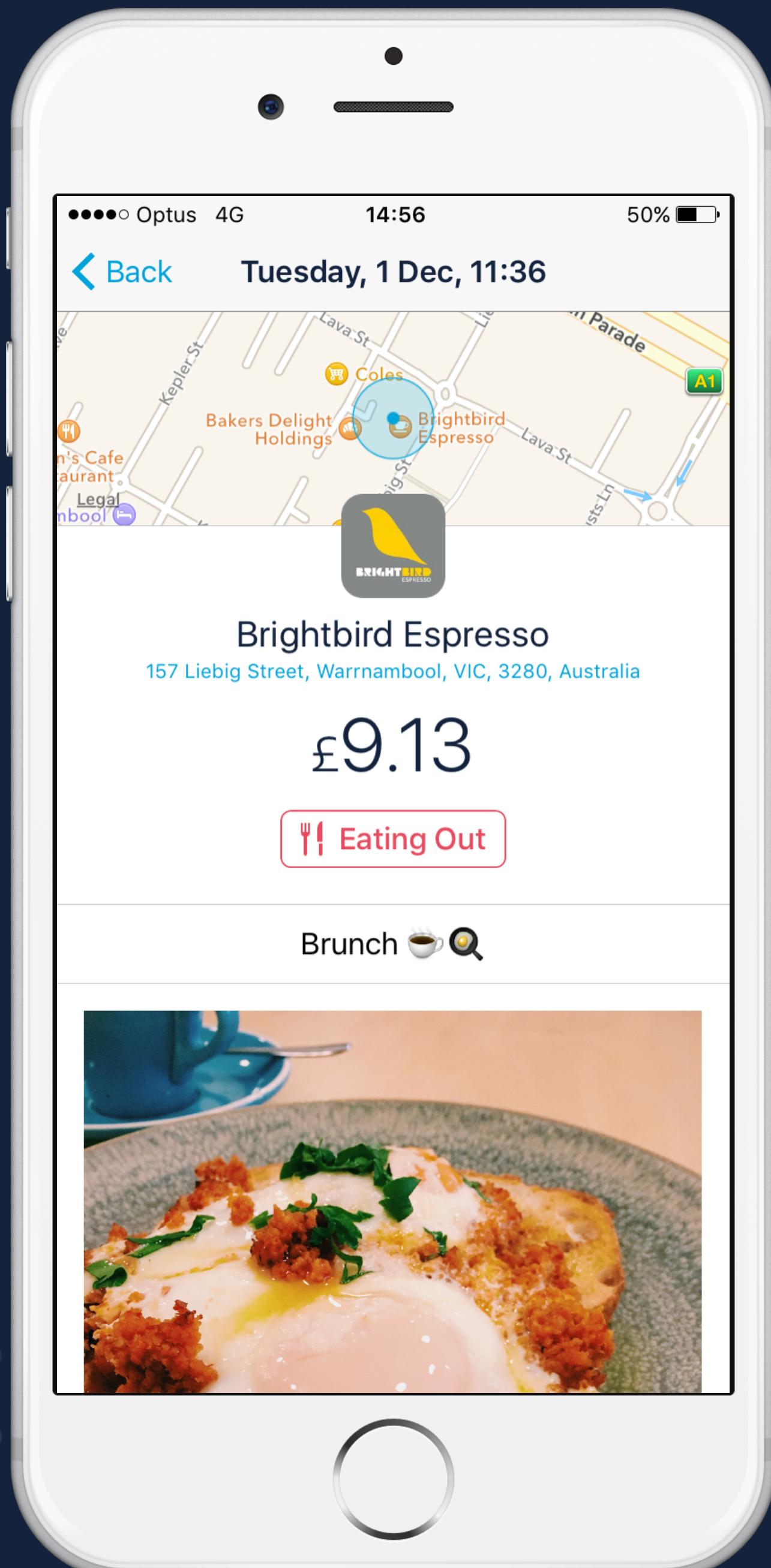


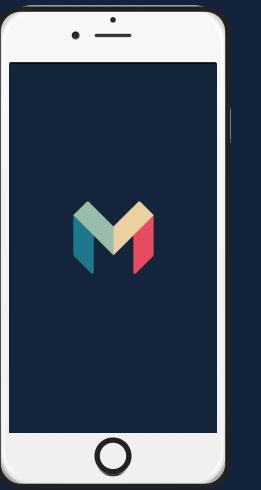












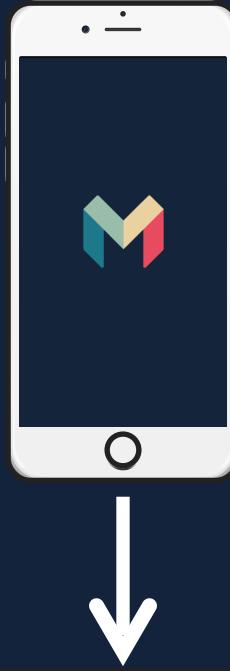
?



APPLICATION



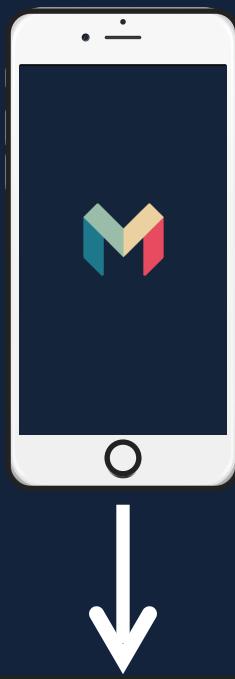
DATABASE



APPLICATION



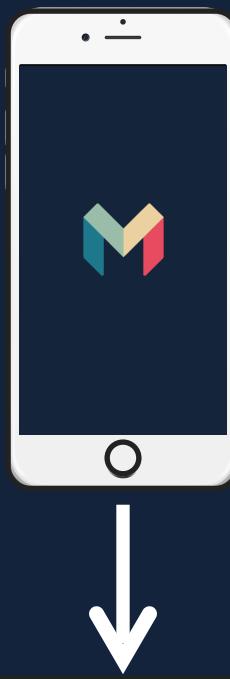
DATABASE



APPLICATION



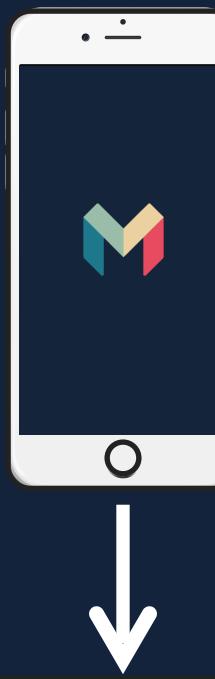
DATABASES



APPLICATION

SEARCH

DATABASES

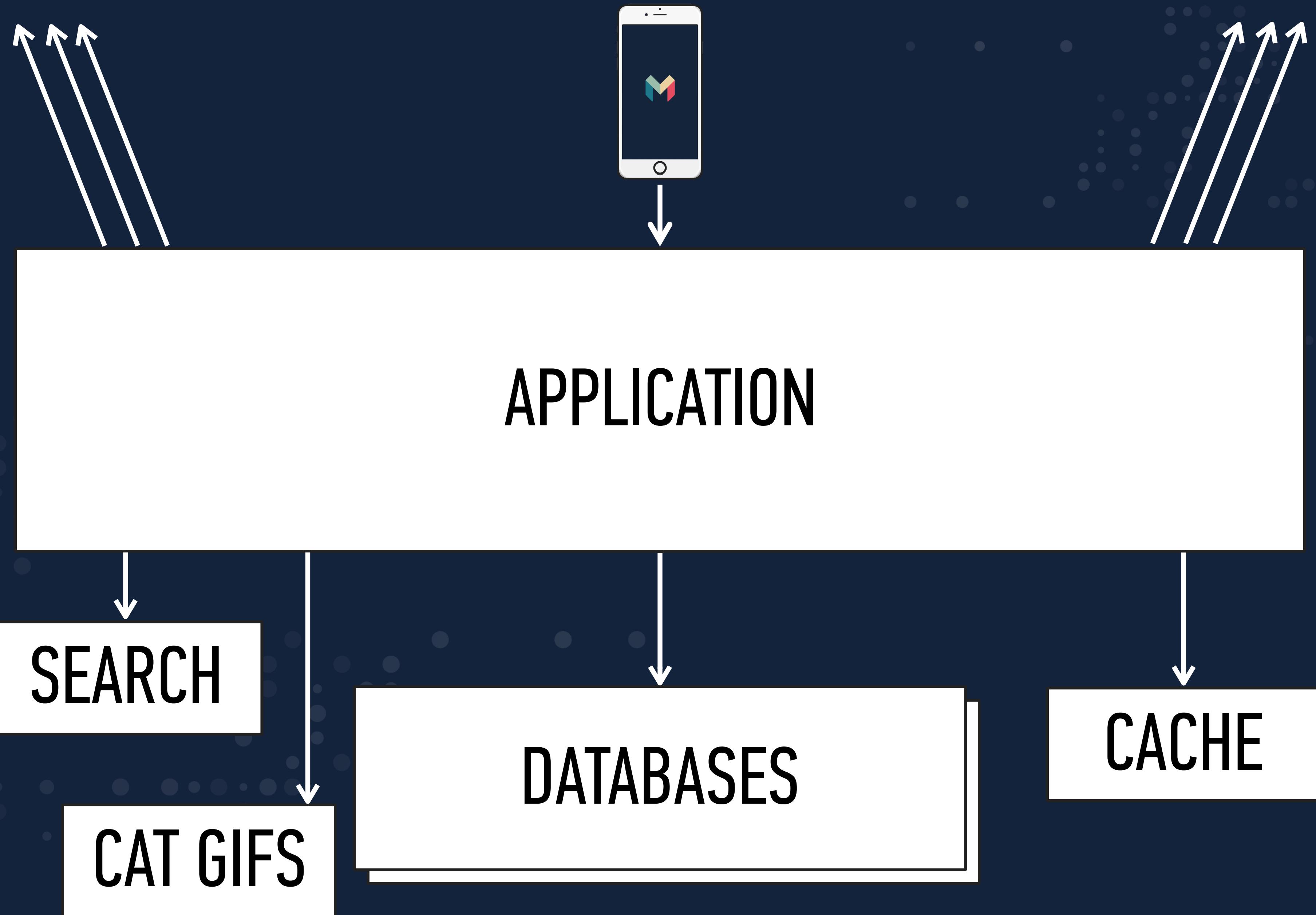


APPLICATION

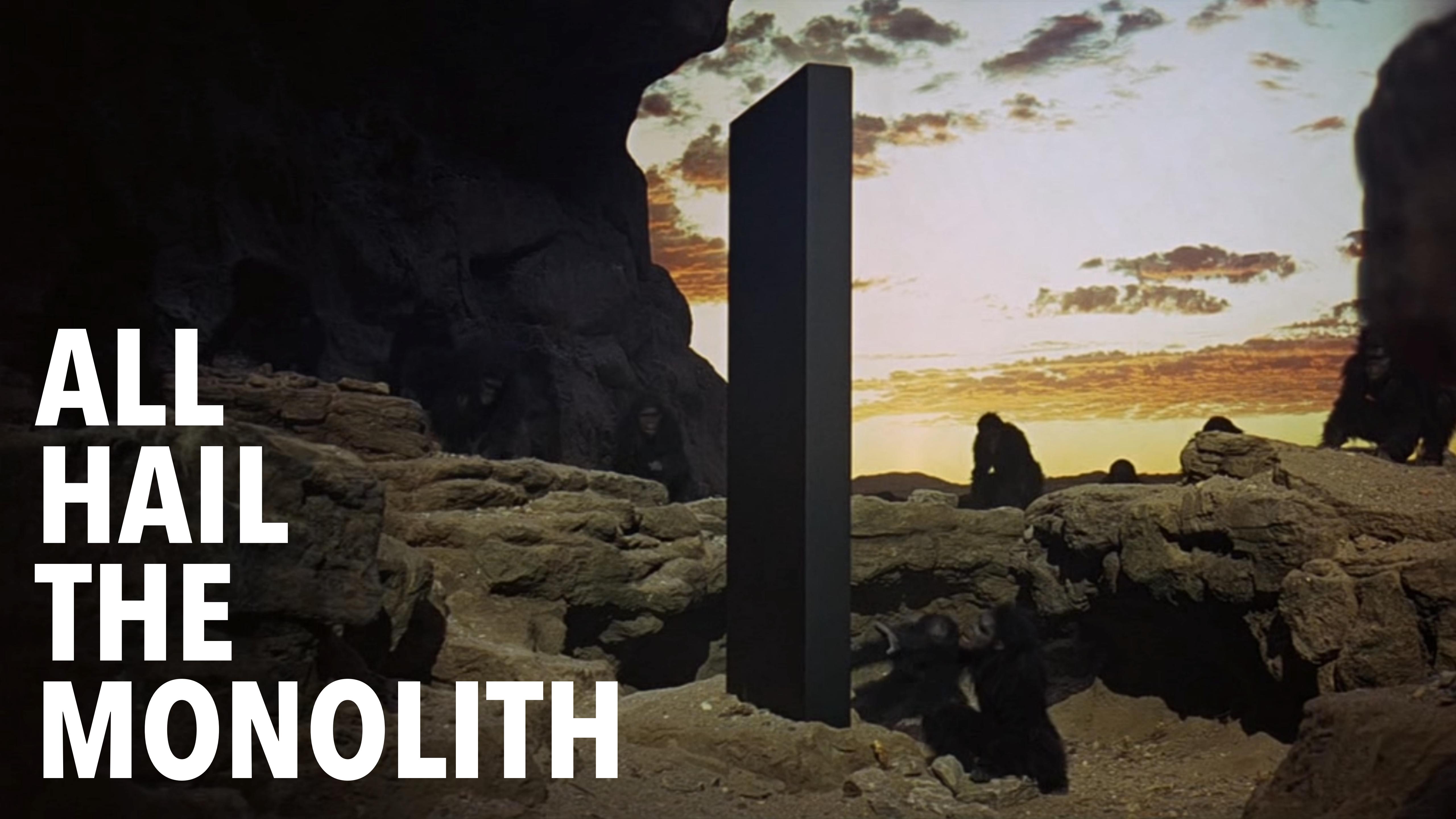
SEARCH

DATABASES

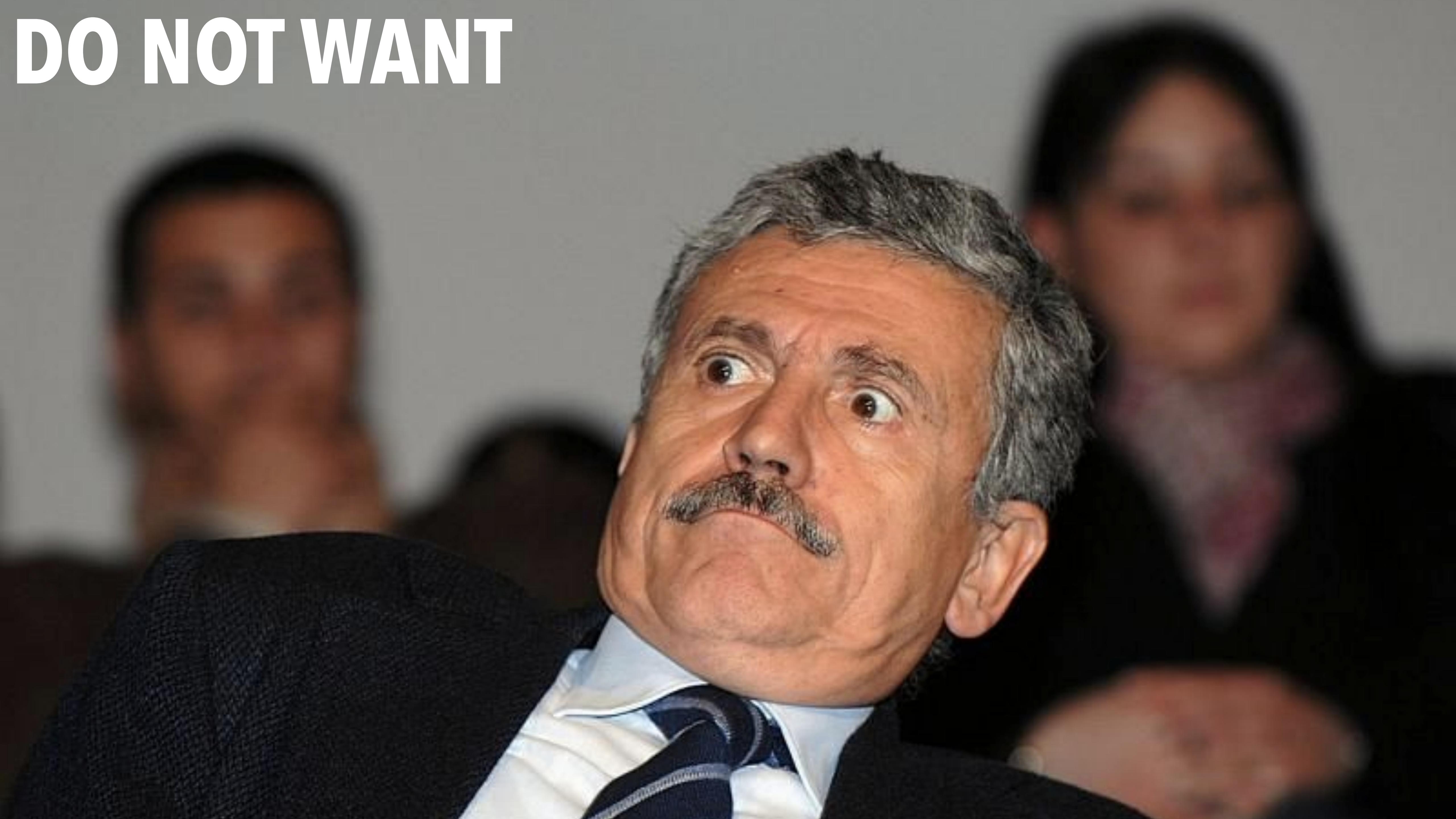
CACHE

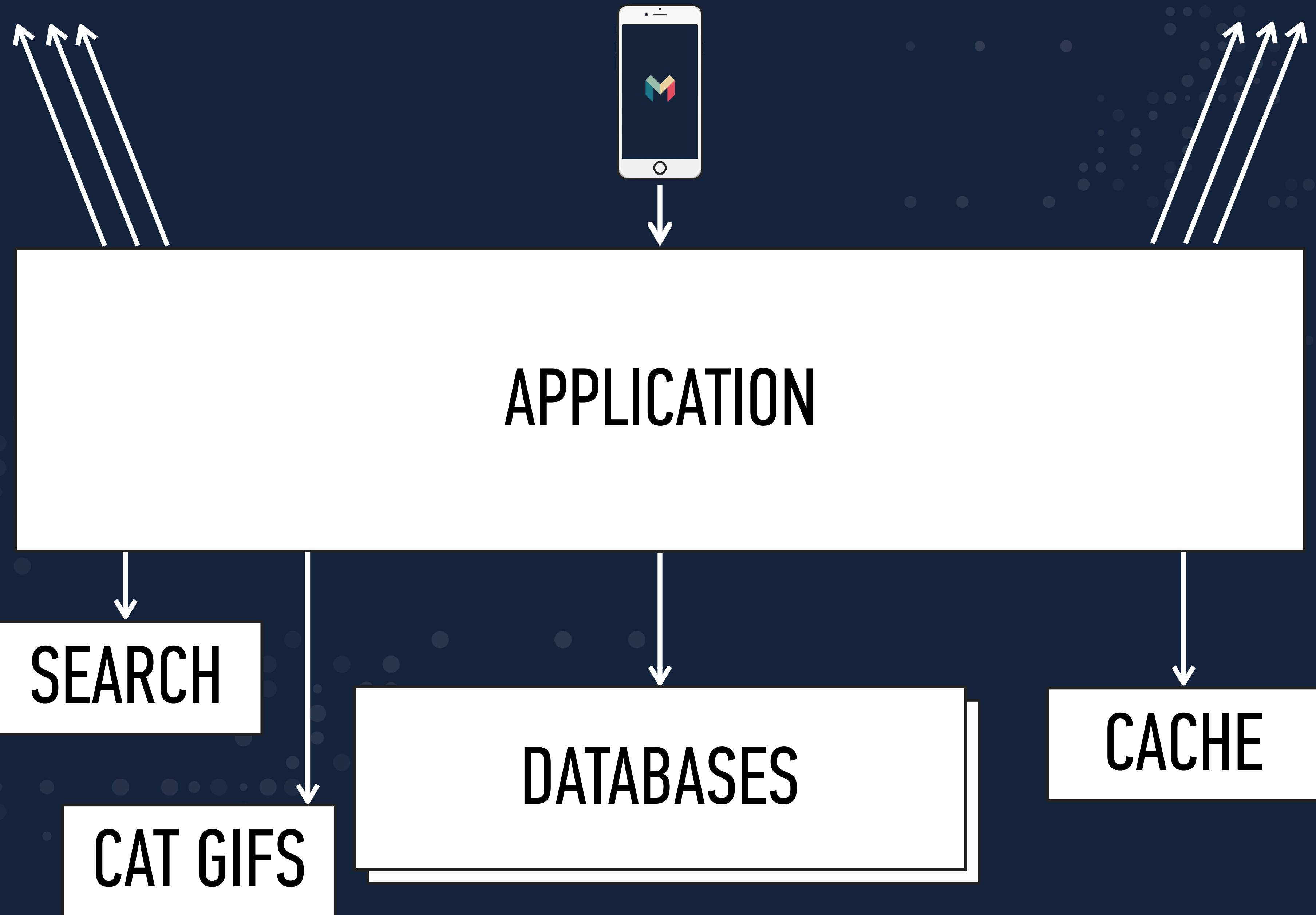


**ALL
HAIL
THE
MONOLITH**

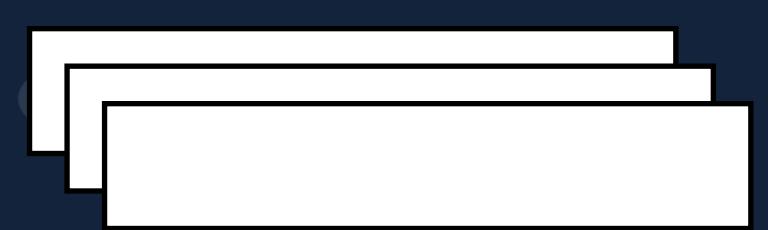


DO NOT WANT

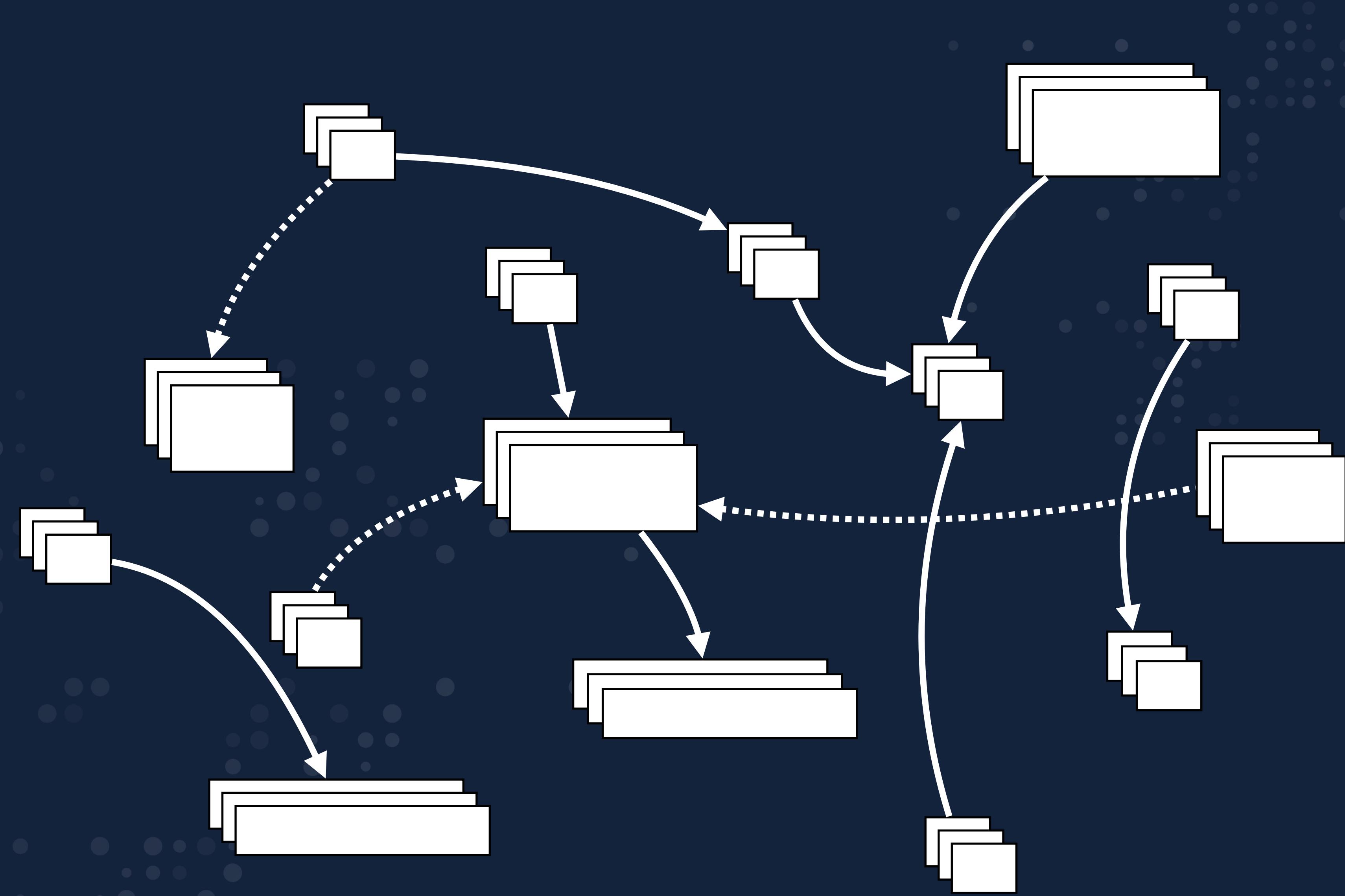




APPLICATION







Why Microservices at Mondo?

Single Responsibility Principle

Bounded Context

Well defined Interfaces

Composability

Why not start with Microservices?

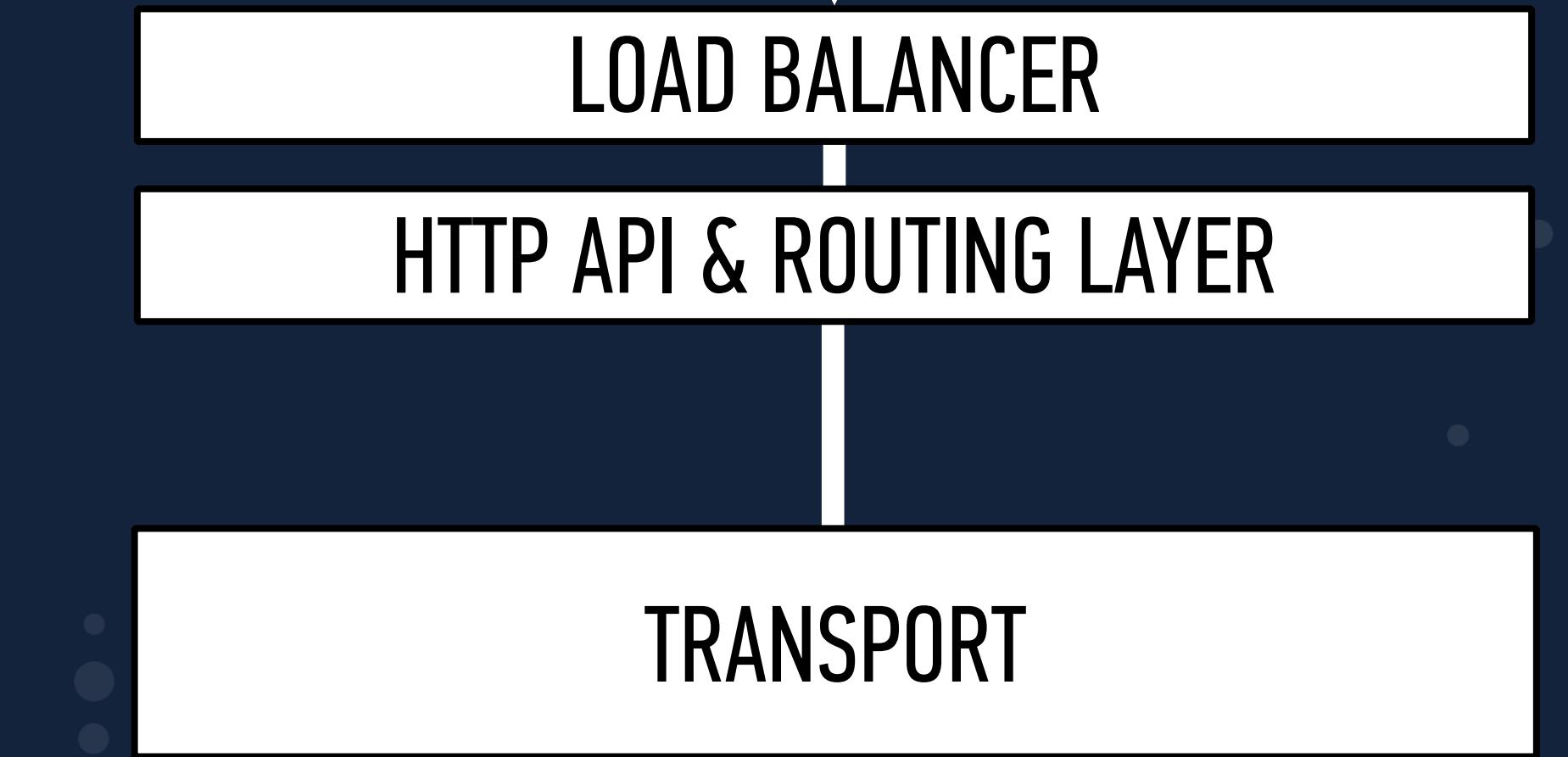


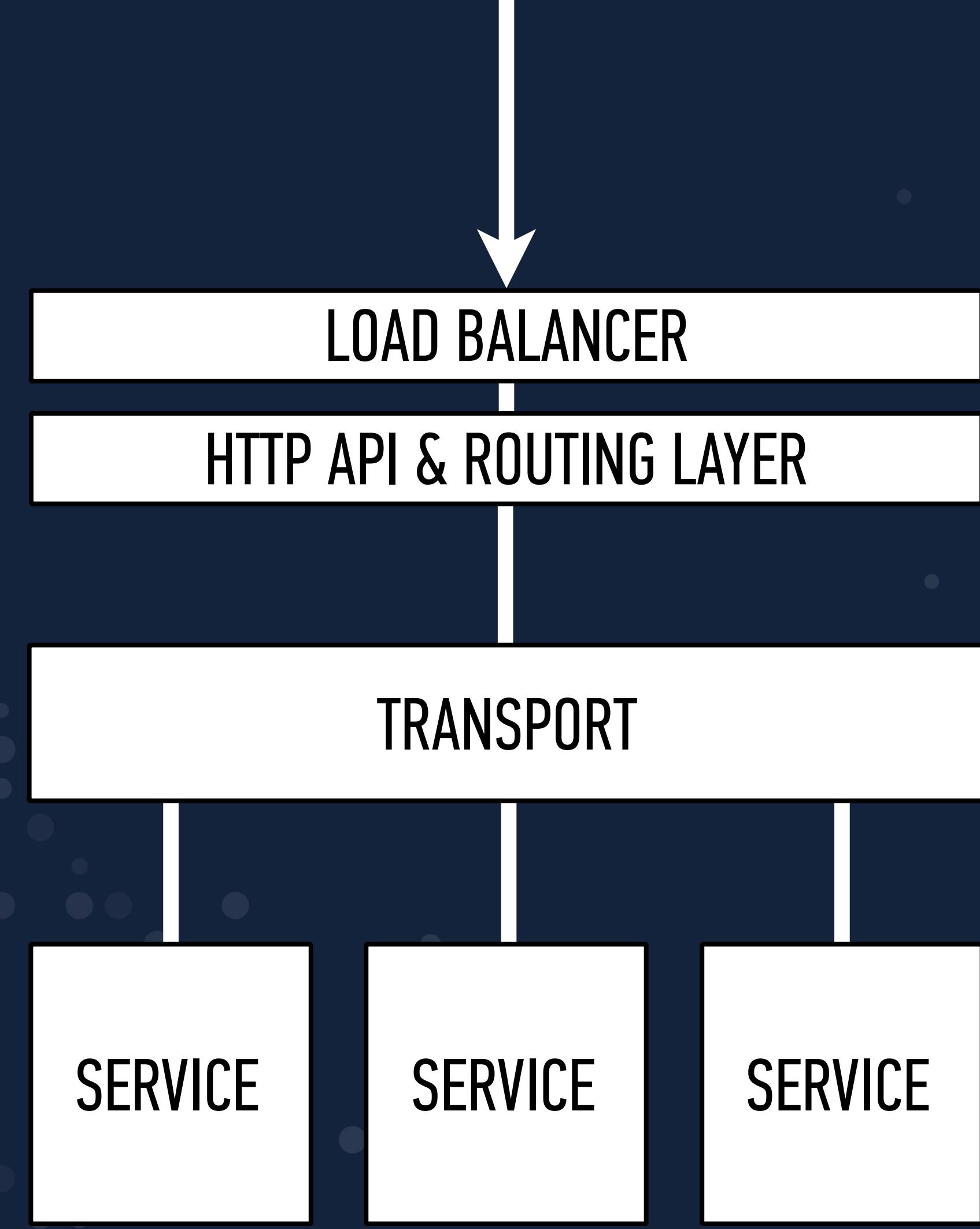
LOAD BALANCER

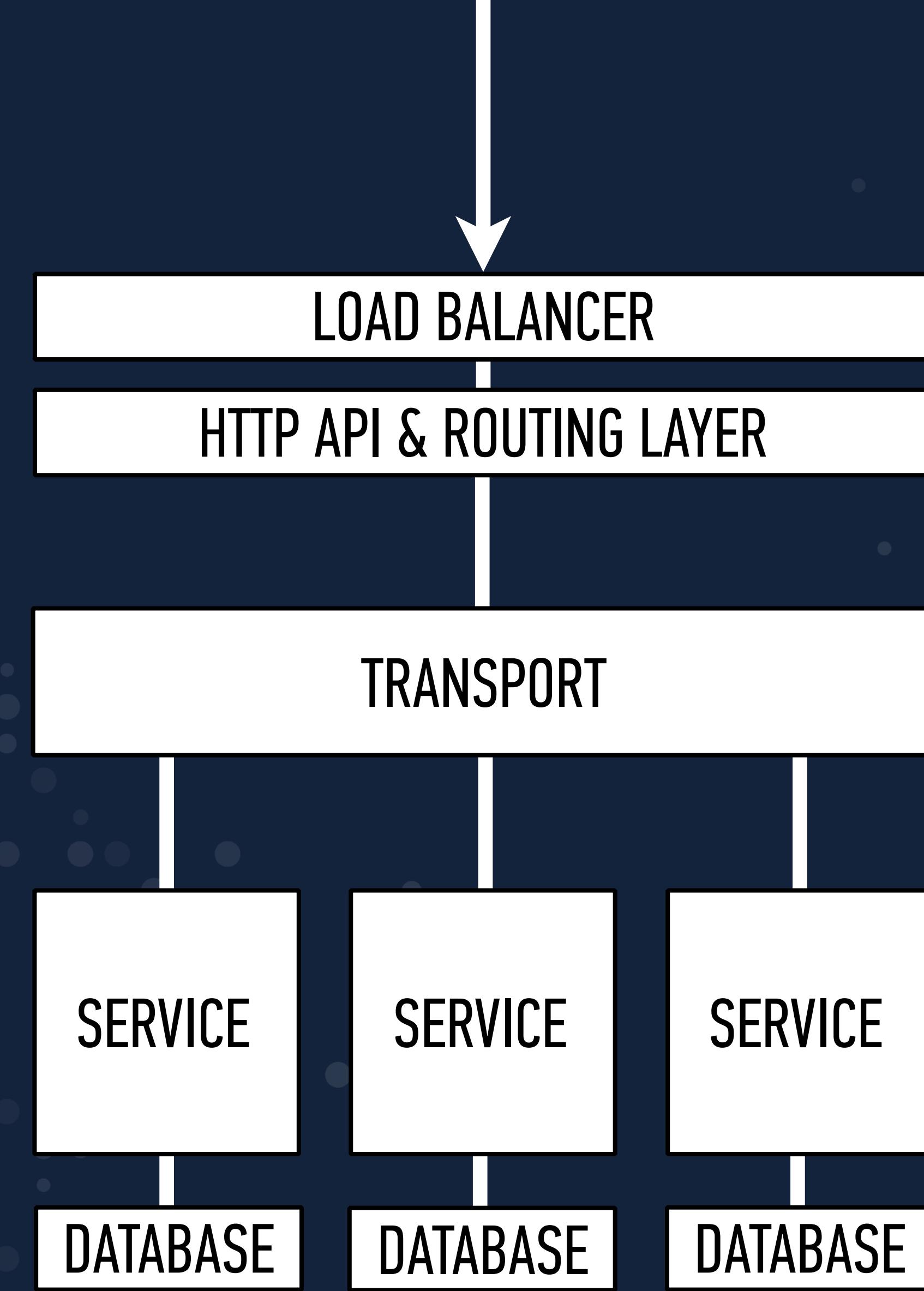


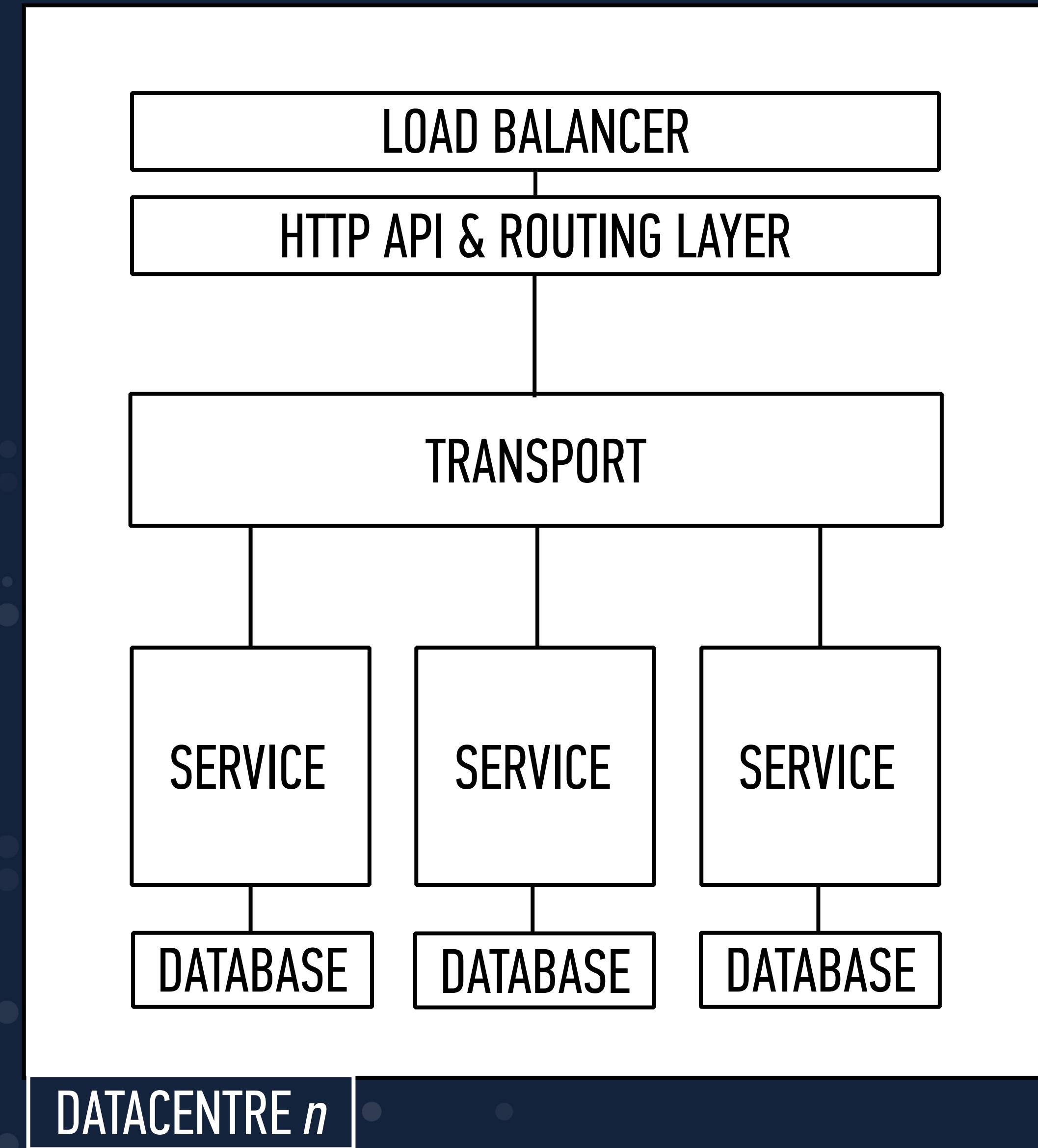
LOAD BALANCER

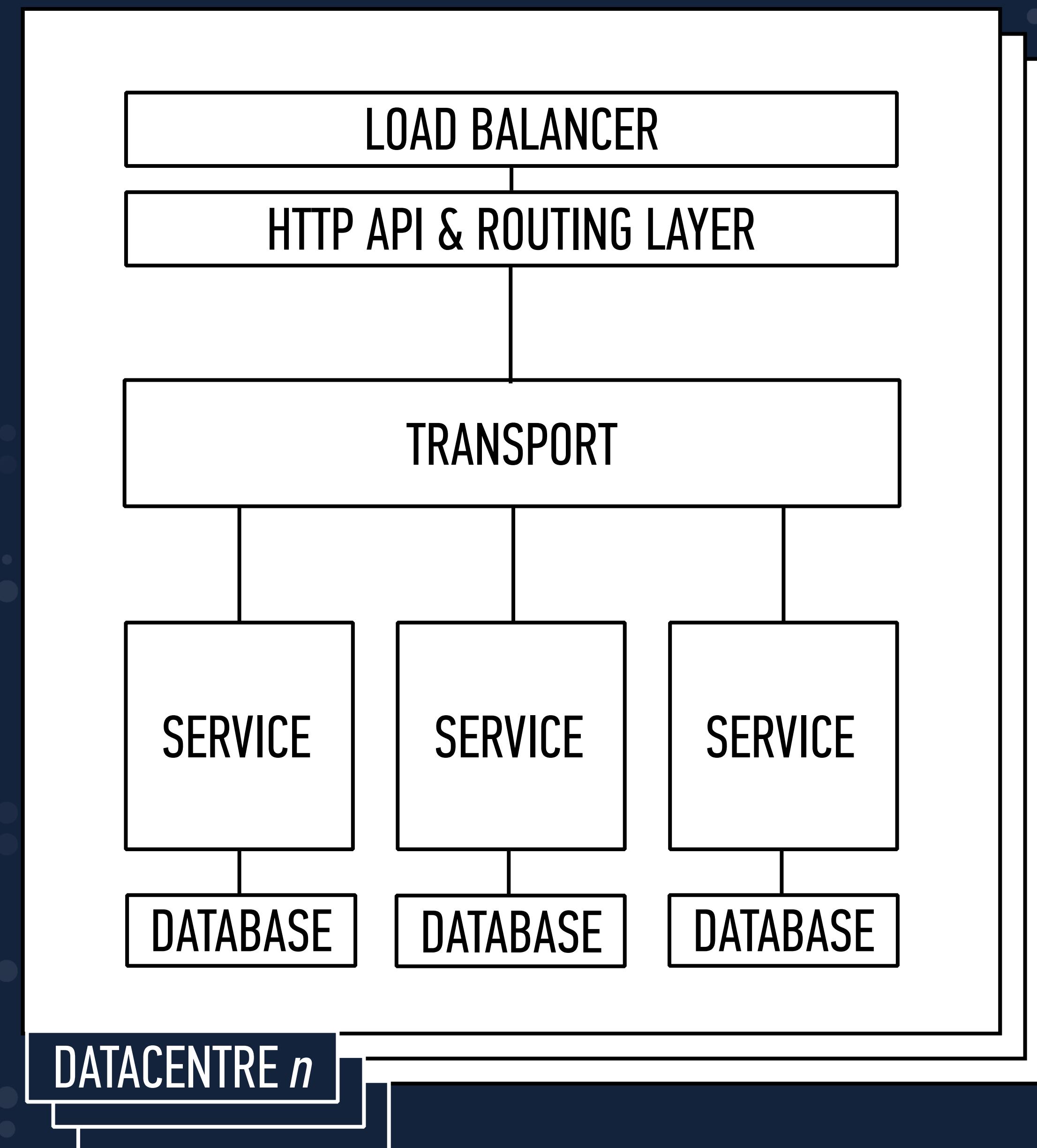
HTTP API & ROUTING LAYER













**Simple
Static typing
Static linking**

Concurrency Interfaces

stdlib
eg. Networking

Go Kit
micro
gRPC
Kite

mondough/typhon
mondough/mercury

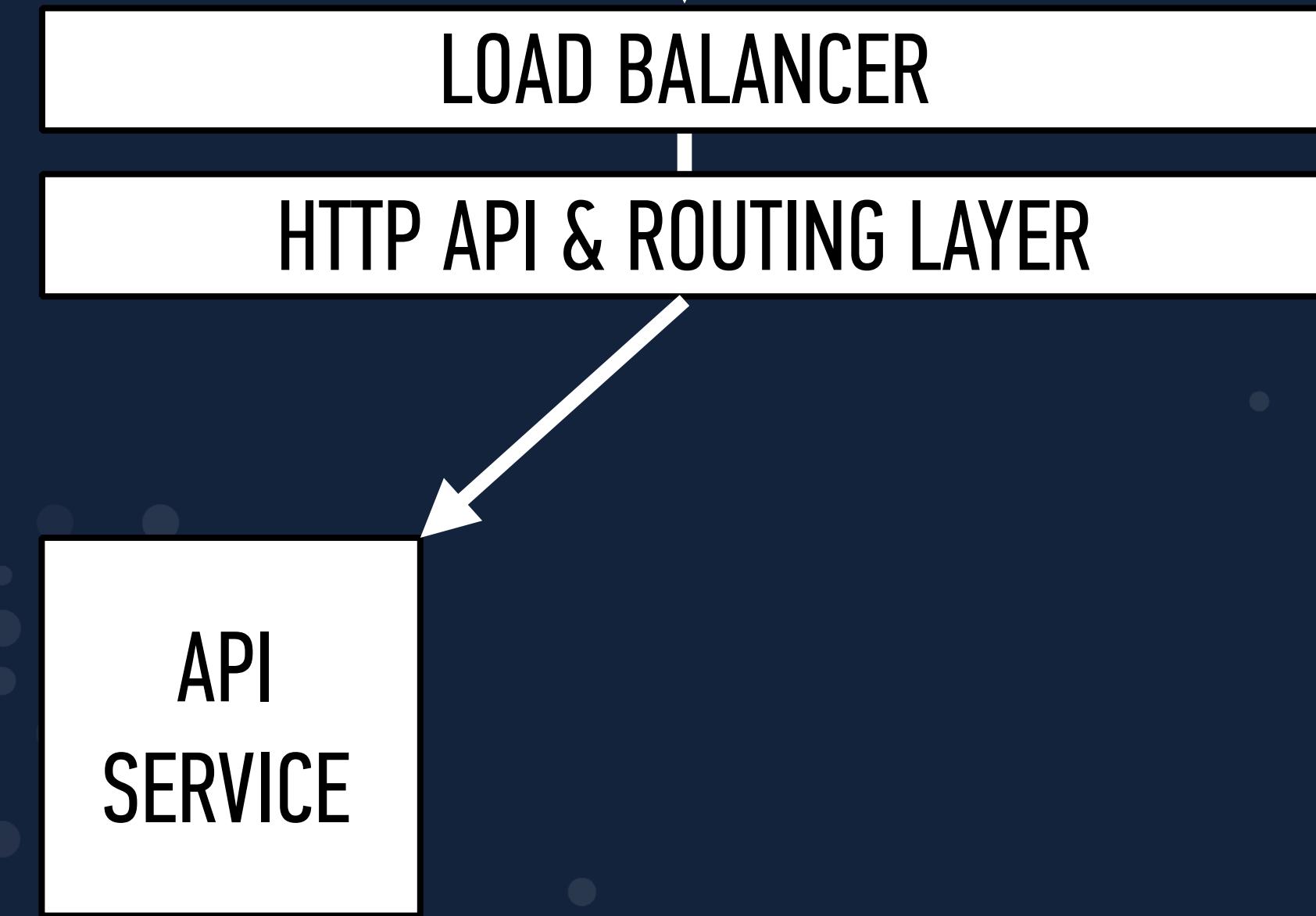


LOAD BALANCER

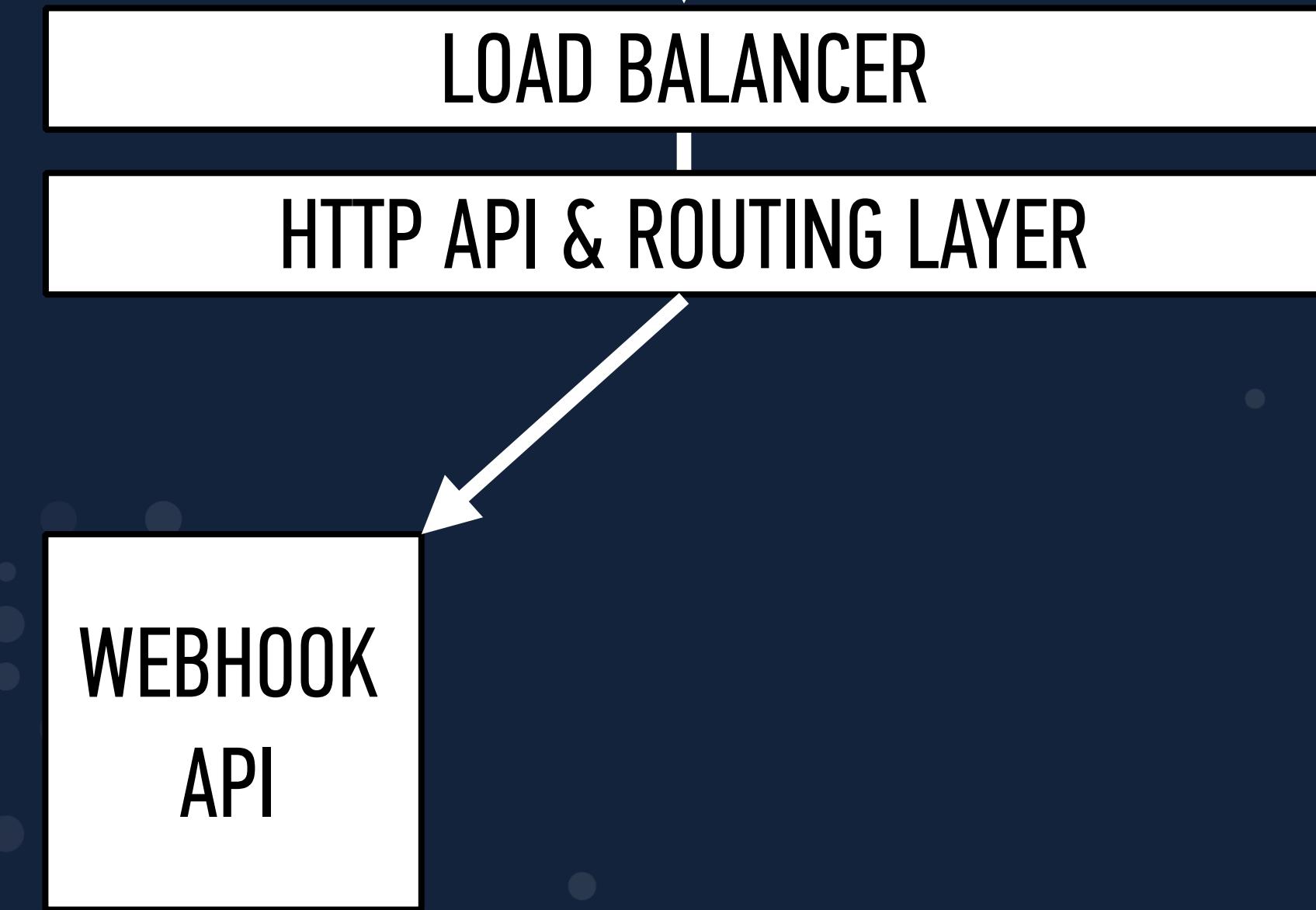


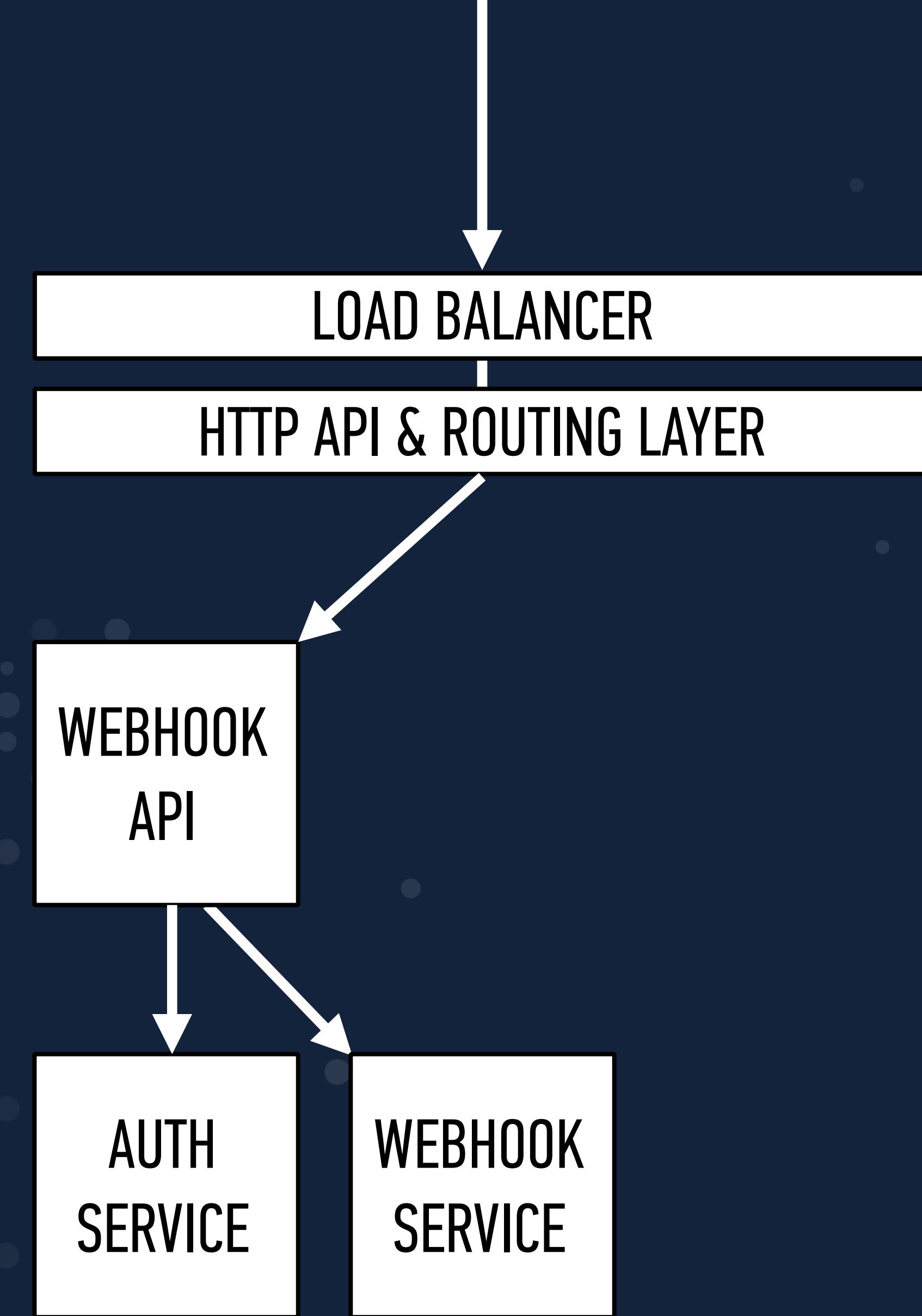
LOAD BALANCER

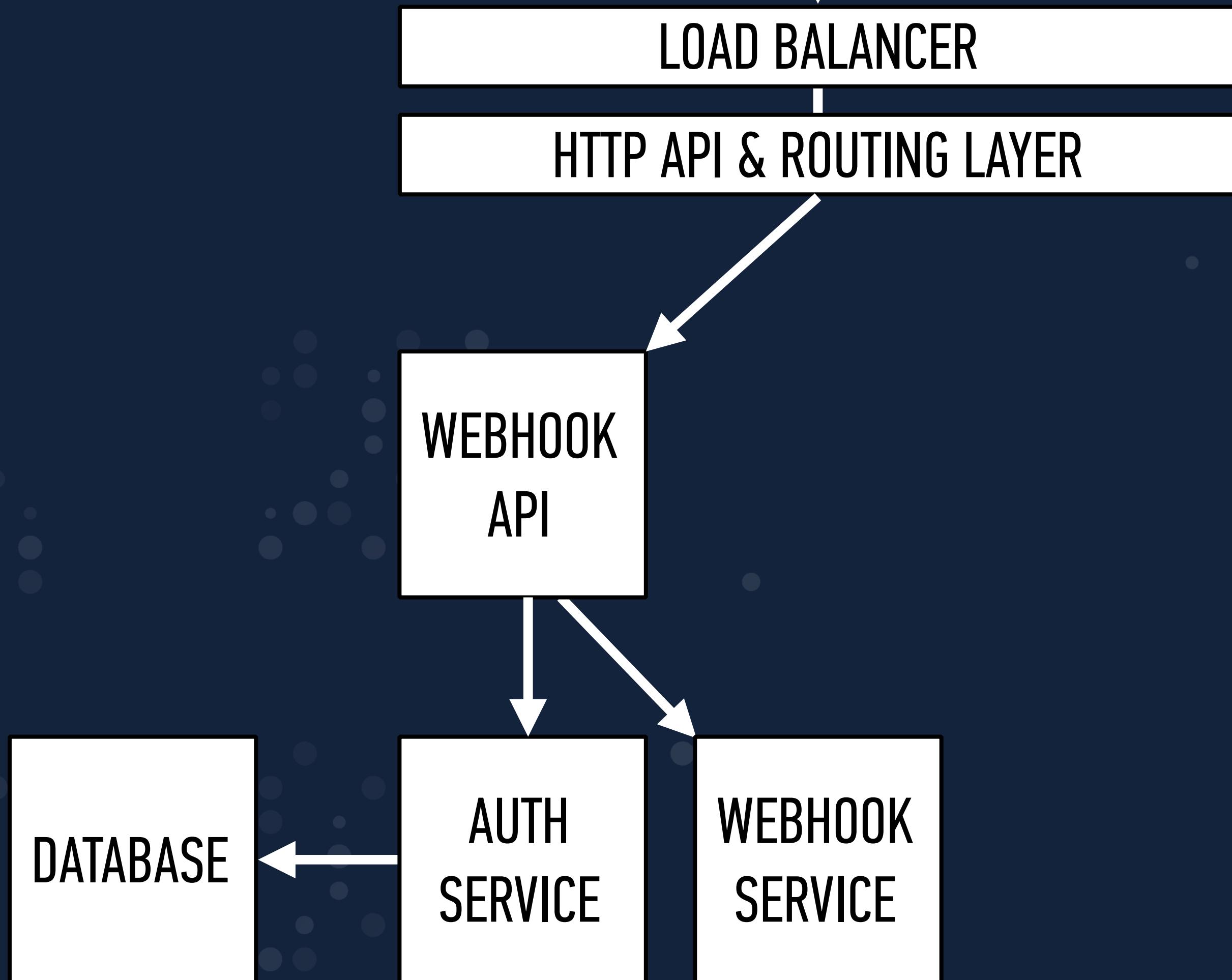
HTTP API & ROUTING LAYER

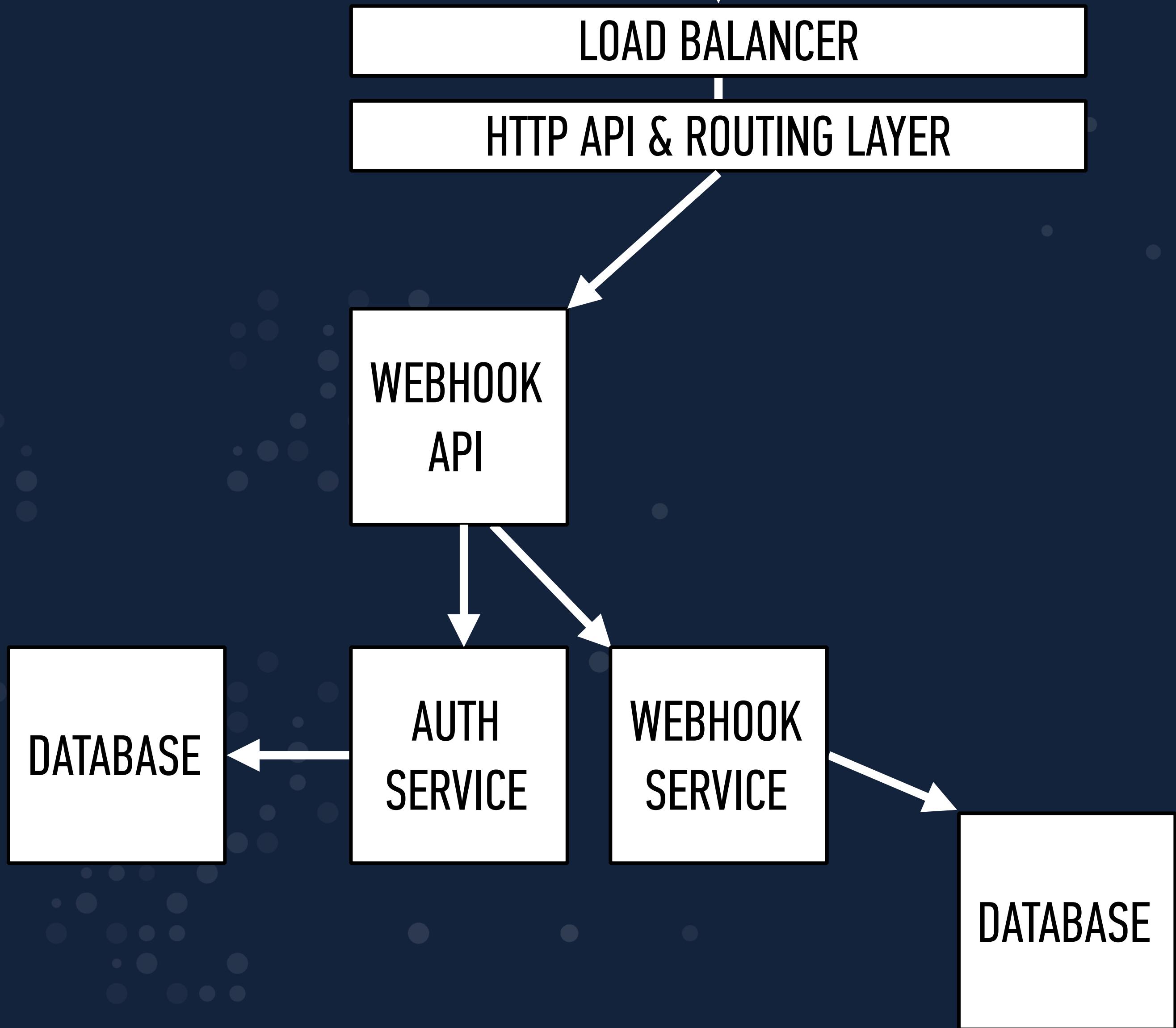


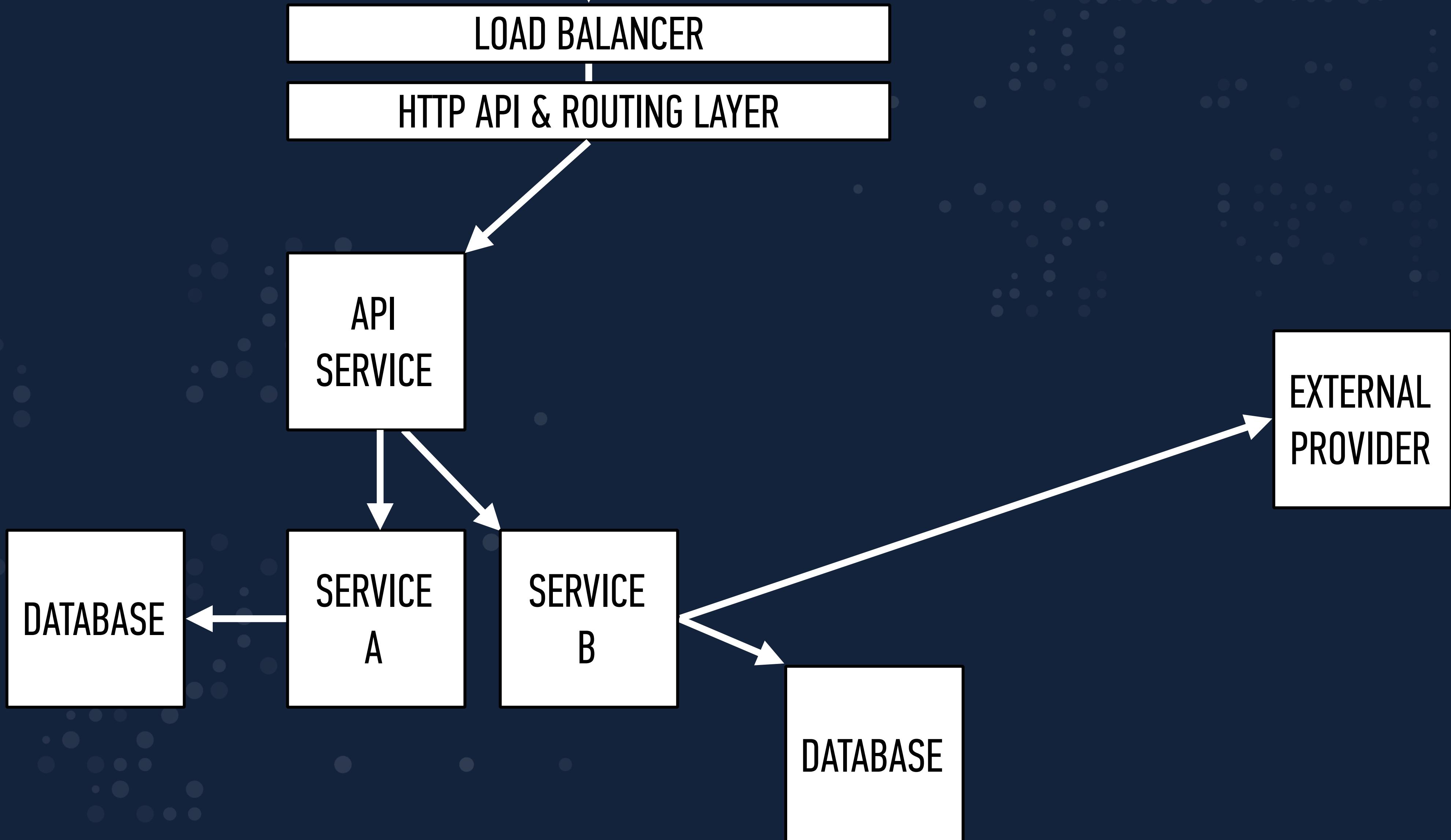
/webhooks → Webhook API



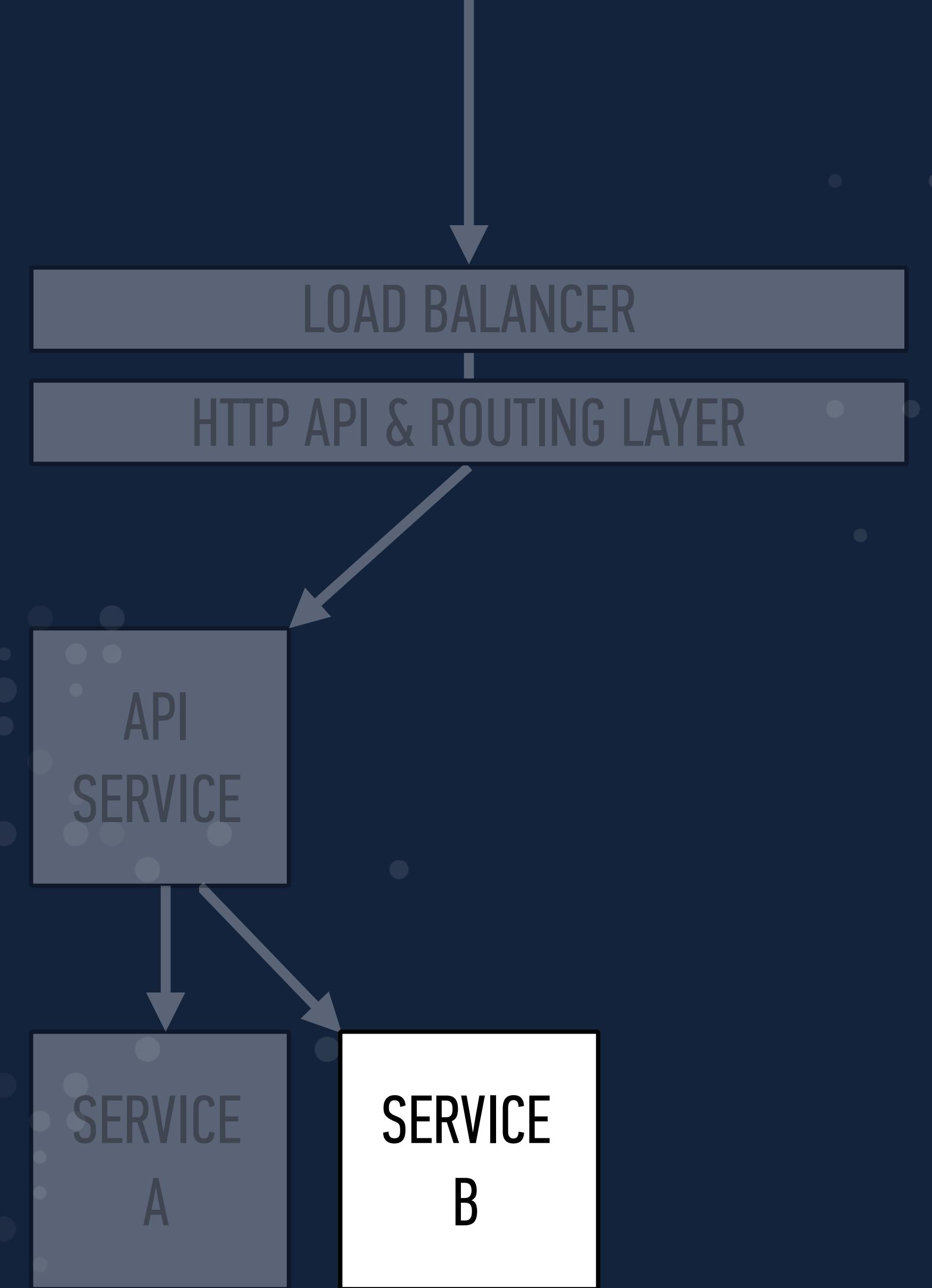


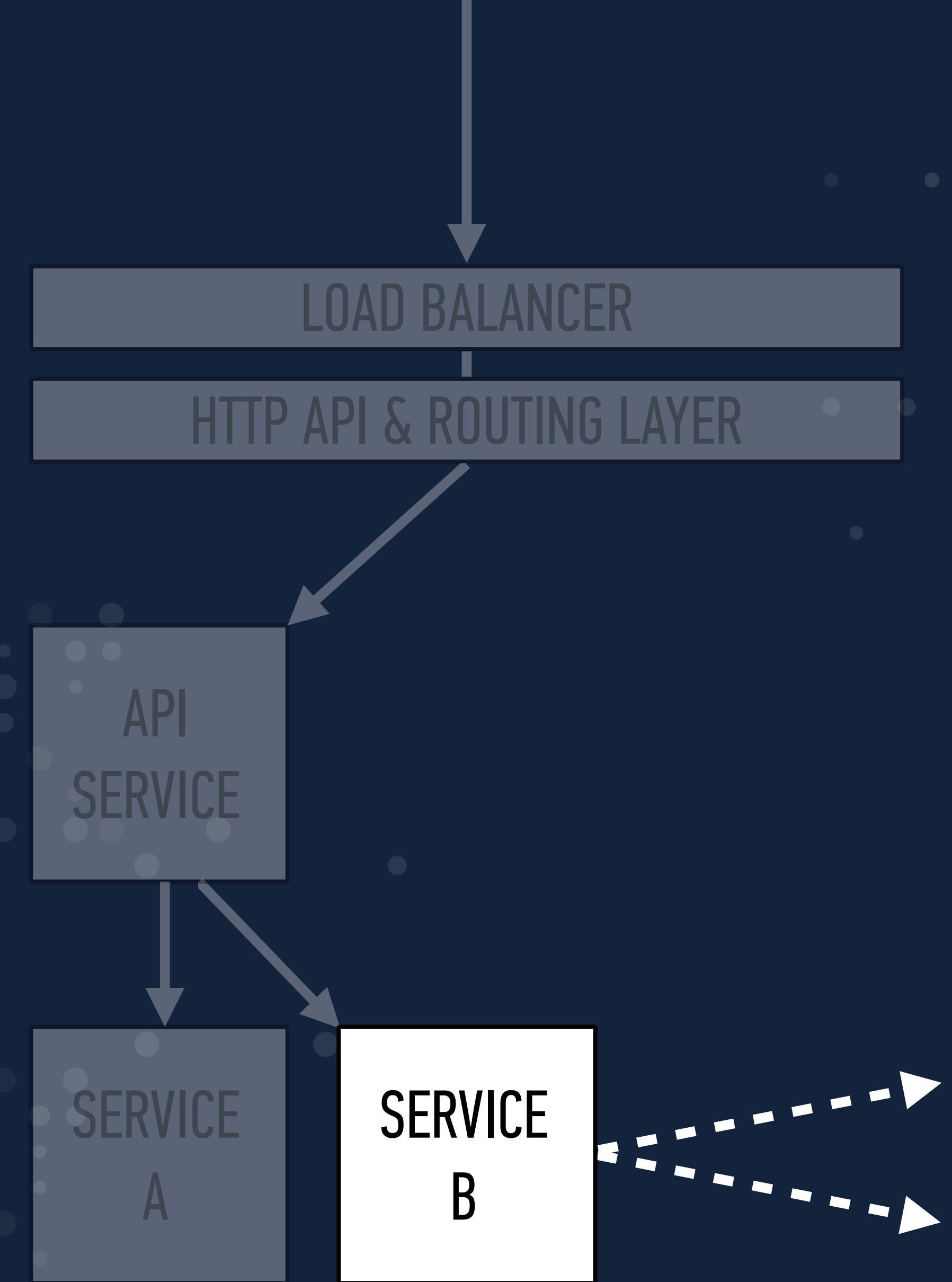


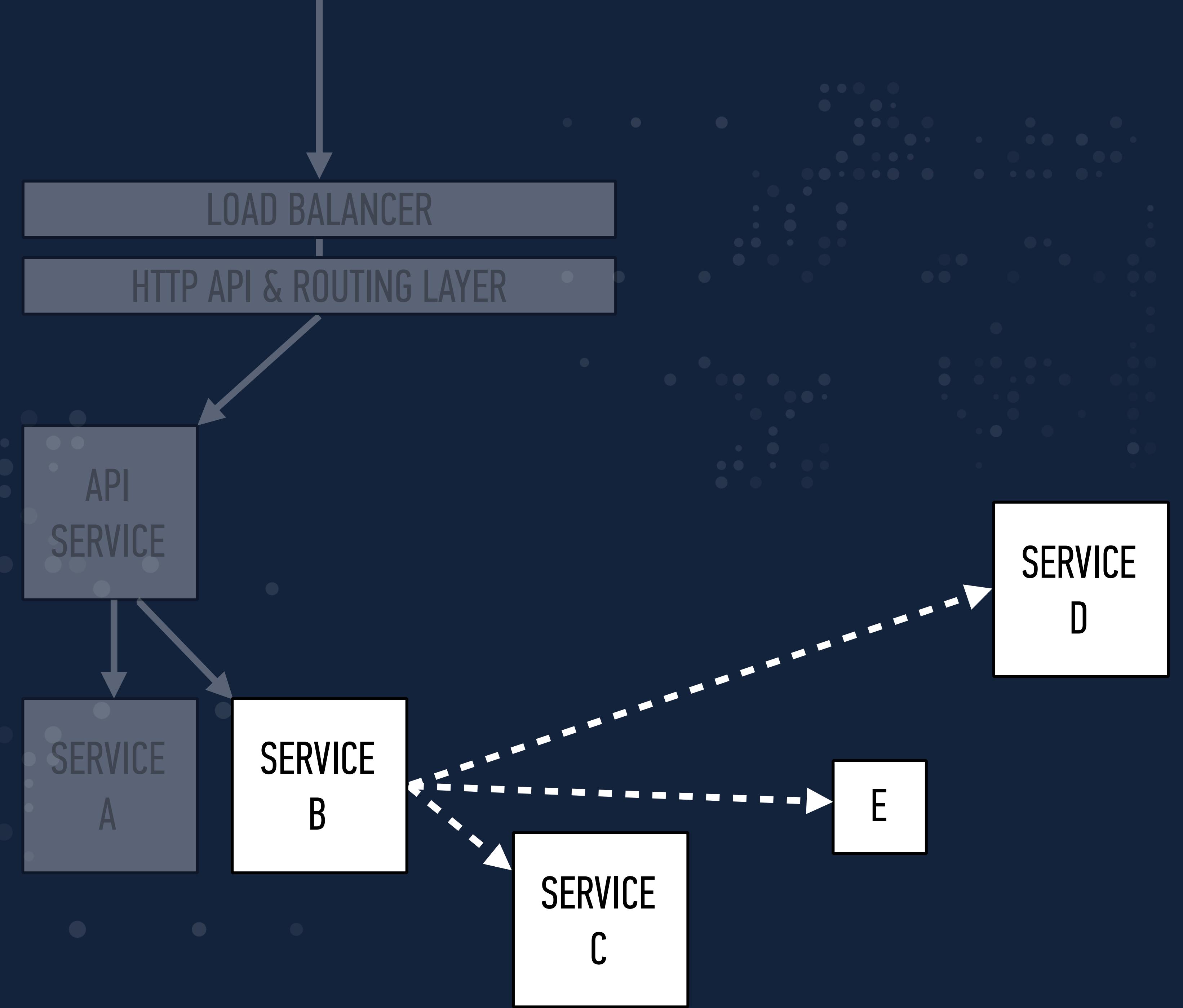


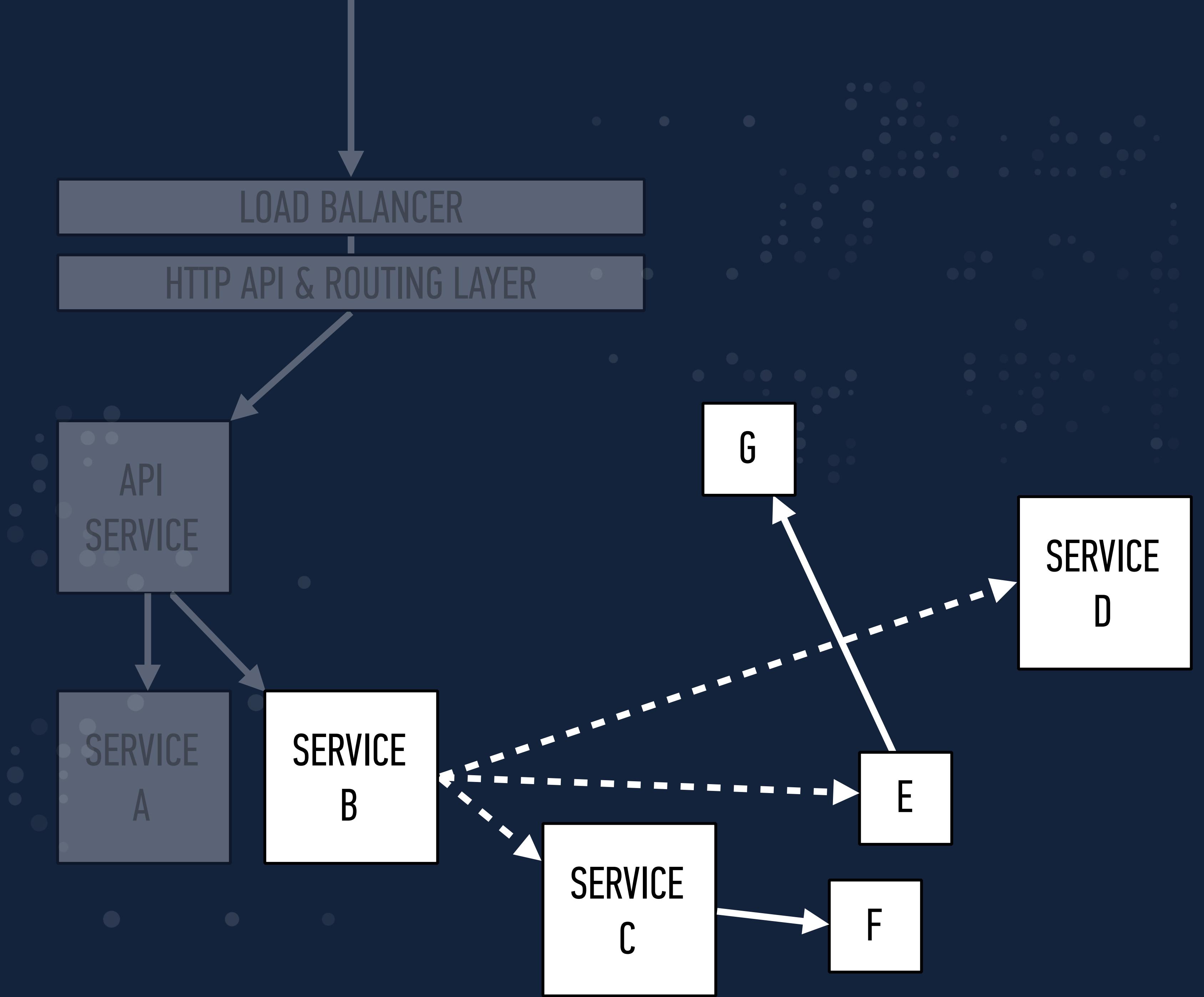


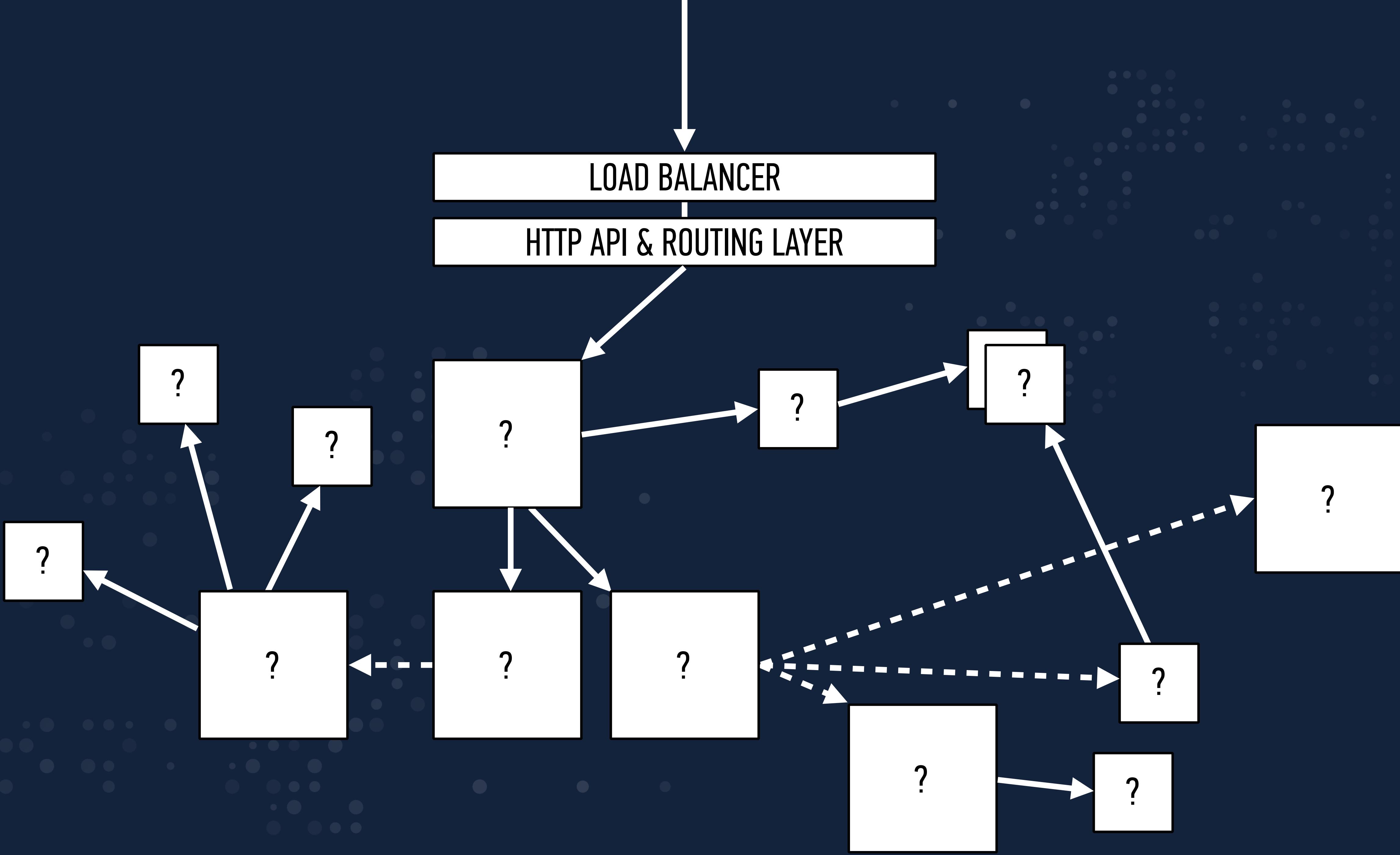
Event Driven Architectures





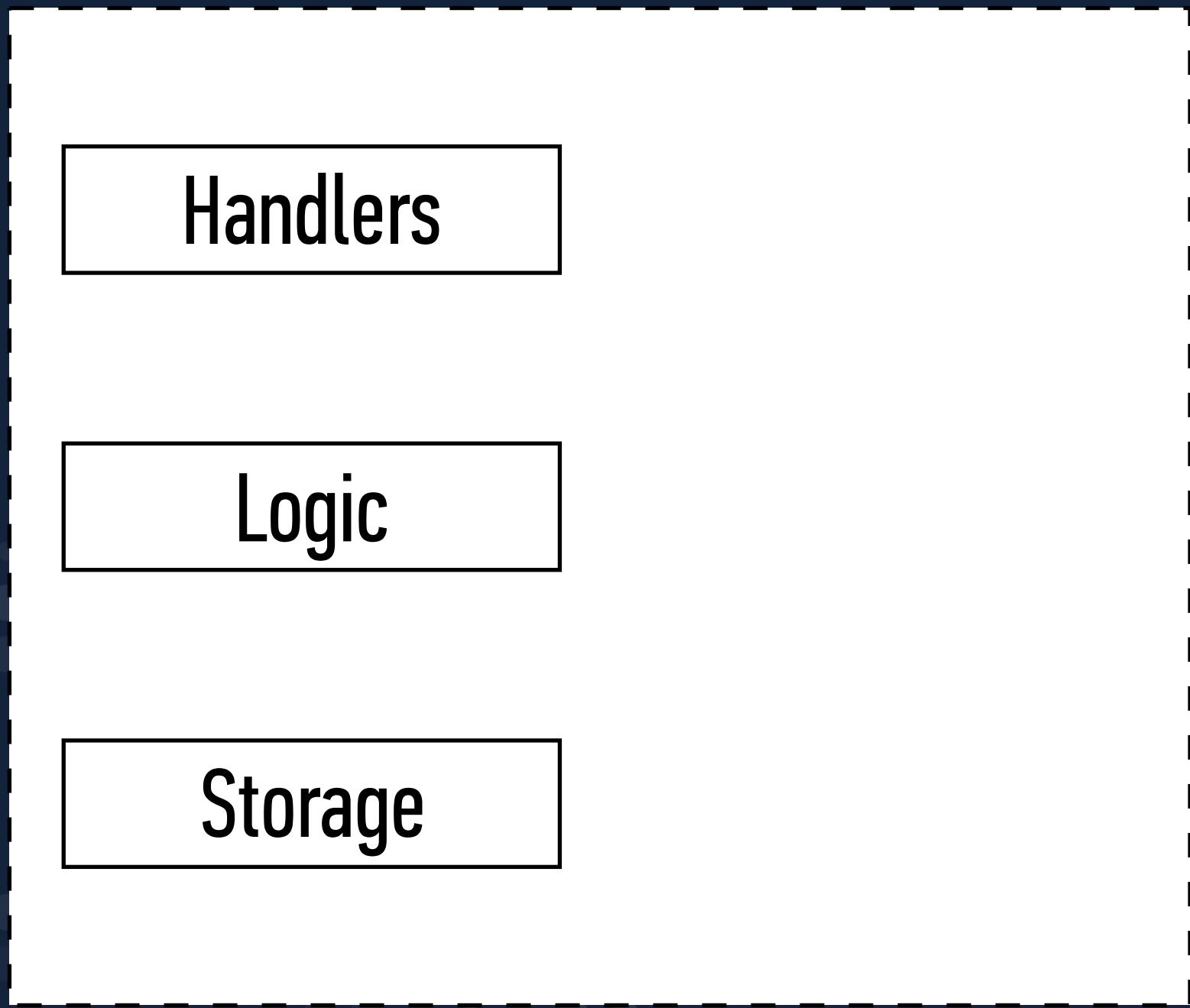




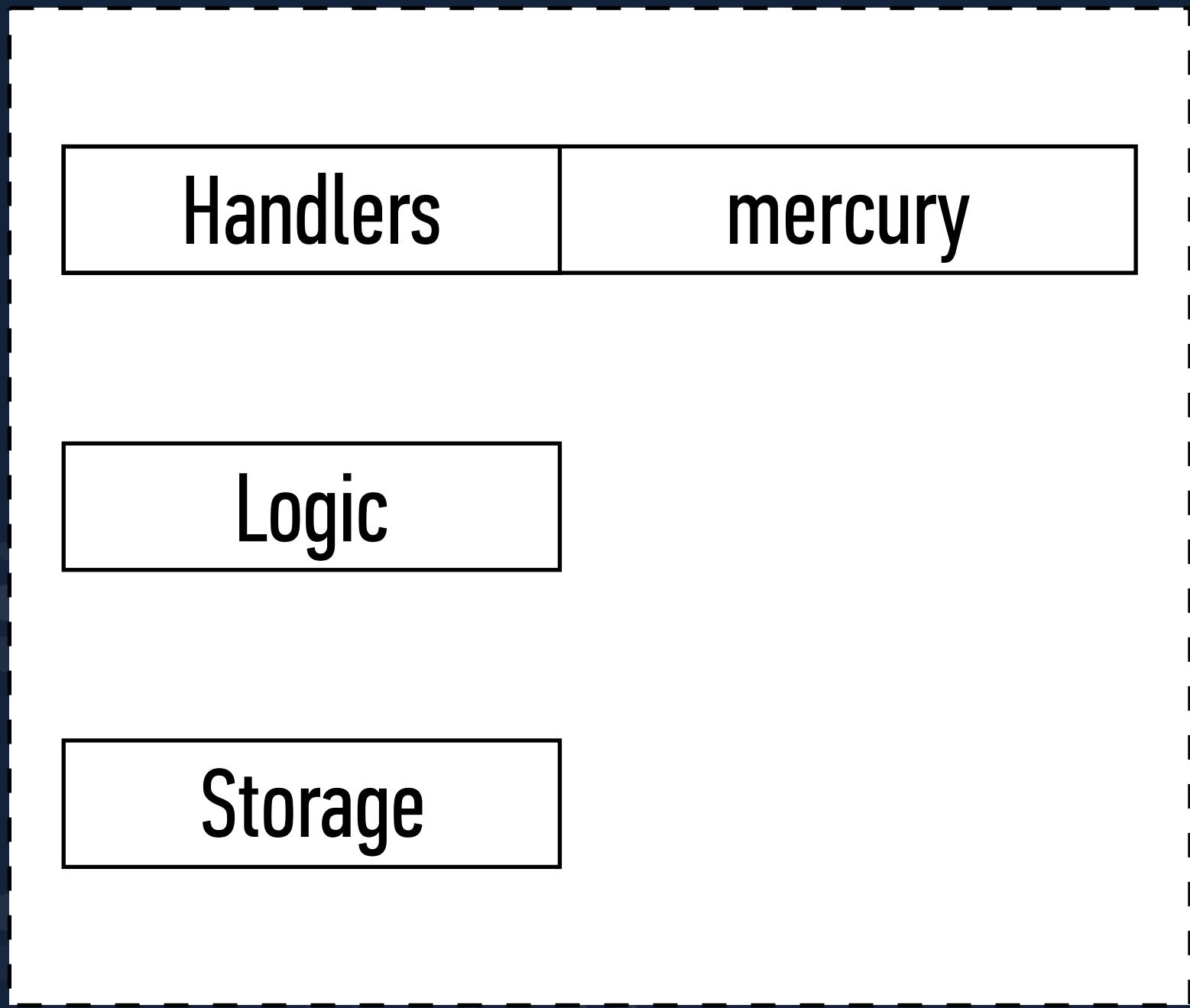




SERVICE



SERVICE



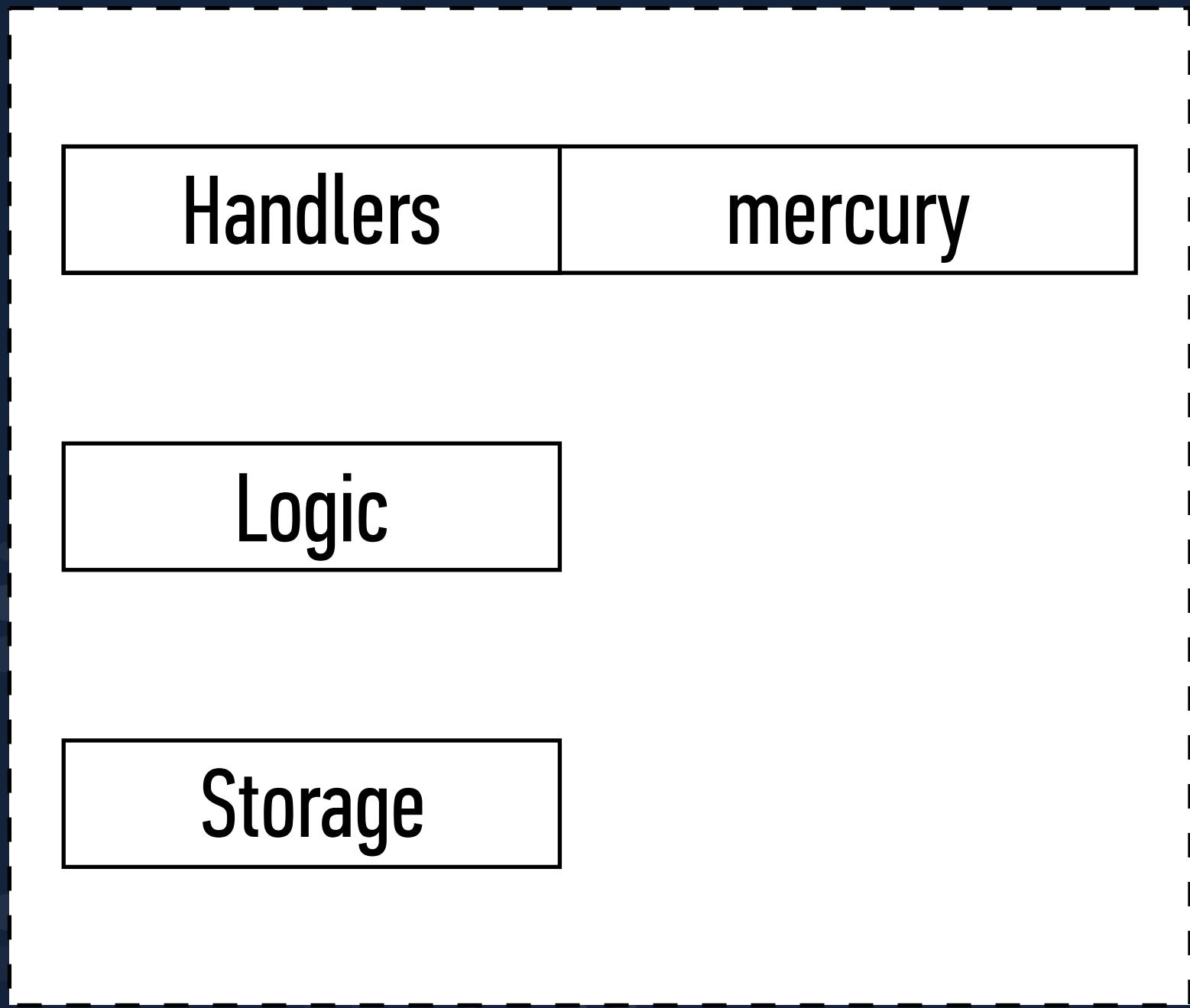
SERVICE

```
type Handler func(request) (response, error)
```

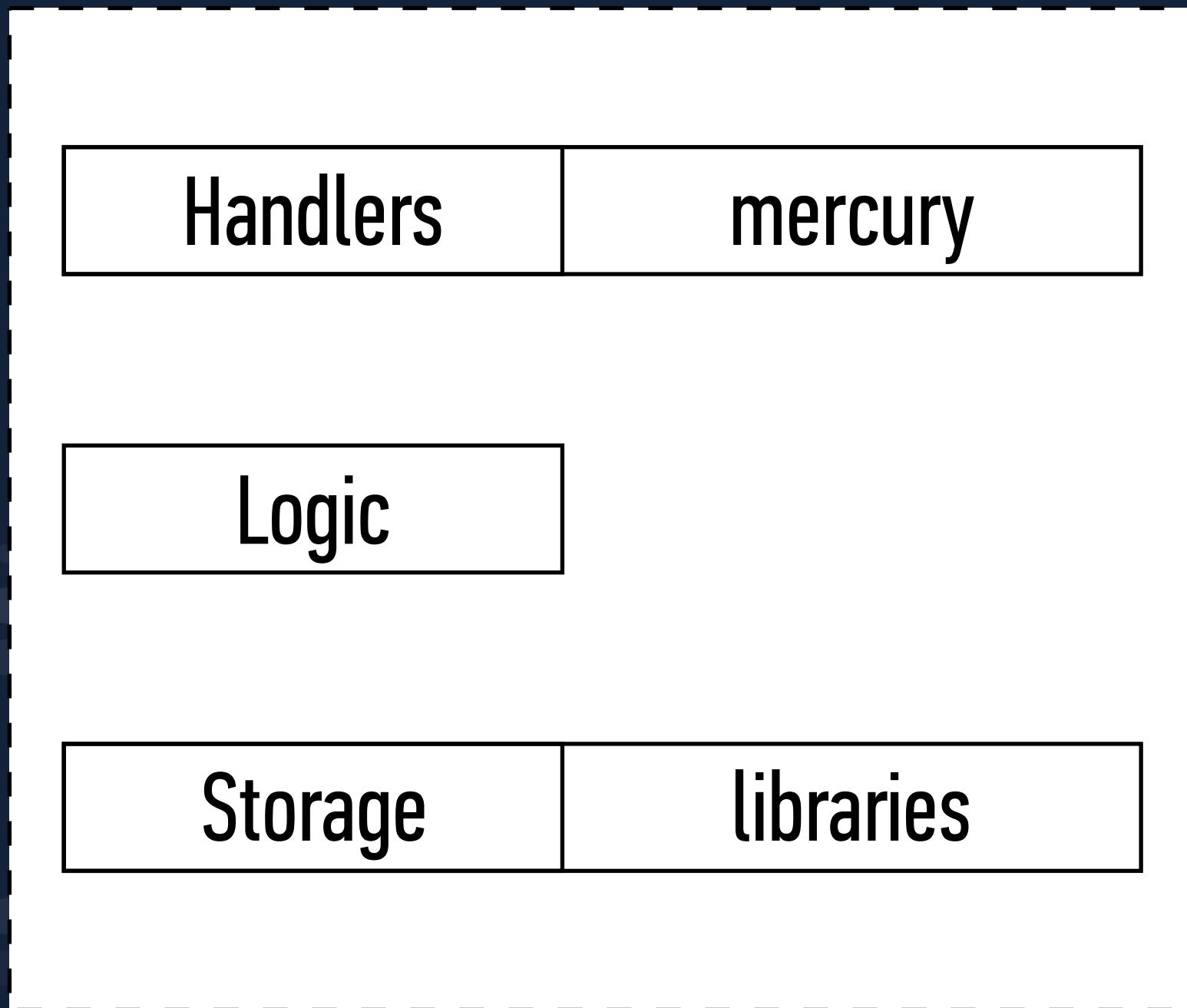
```
type Transport interface {  
    ...  
}
```

```
type Transport interface {  
    Listen(serviceName string, inbound chan<- request) error  
    StopListening(serviceName string) bool  
  
    Respond(req request, resp response) error  
    Send(req request, timeout time.Duration) (response, error)  
  
    ...  
}
```

```
type Transport interface {  
    Listen(serviceName string, inbound chan<-> request) error  
    StopListening(serviceName string) bool  
  
    Respond(req request, resp response) error  
    Send(req request, timeout time.Duration) (response, error)  
  
}
```

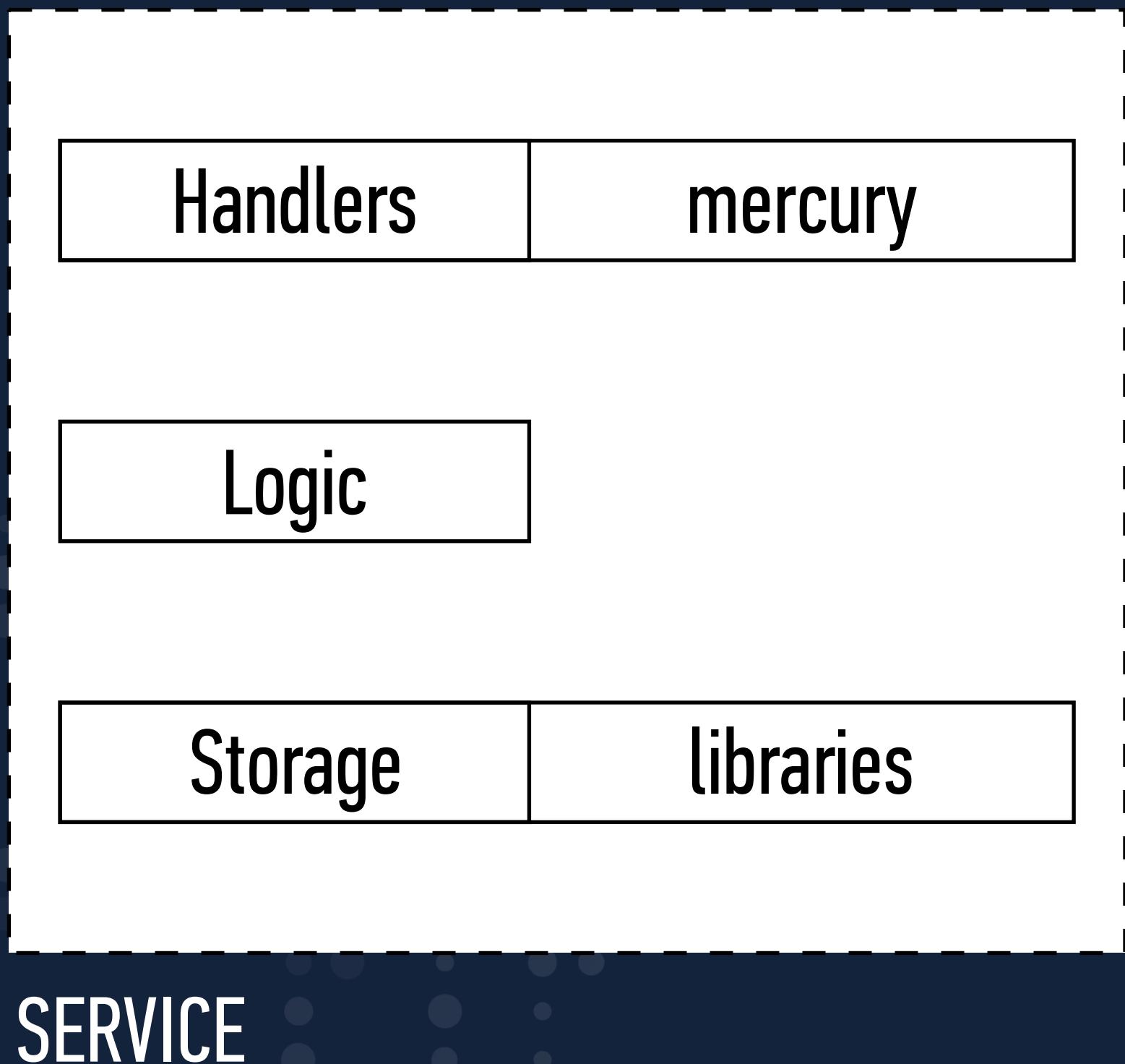


SERVICE



SERVICE

Deployment
Service Discovery
Configuration
Monitoring
Authentication
Authorisation
Storage
Circuit Breaking



MAX GROSS
TARE

30480 KG
67200 LB
2350 KG
5180 LB

MAX PAYLOAD
CU CAP

28130 KG
62020 LB
33.1 CU.M
1170 CU.FT

MAX GROSS
TARE

30480 KG
67200 LB
2350 KG
5180 LB

MAX PAYLOAD
CU CAP

28130 KG
62020 LB
33.1 CU.M
1170 CU.FT

MAX GROSS
TARE

30480 KG
67200 LB
2350 KG
5180 LB

MAX PAYLOAD
CU CAP

28130 KG
62020 LB
33.1 CU.M
1170 CU.FT

MAX GROSS
TARE

30480 KG
67200 LB
2350 KG
5180 LB

MAX PAYLOAD
CU CAP

28130 KG
62020 LB
33.1 CU.M
1170 CU.FT

MAX GROSS
TARE

30480 KG
67200 LB
2350 KG
5180 LB

MAX PAYLOAD
CU CAP

28130 KG
62020 LB
33.1 CU.M
1170 CU.FT

MAX GROSS
TARE

30480 KG
67200 LB
2350 KG
5180 LB

MAX PAYLOAD
CU CAP

28130 KG
62020 LB
33.1 CU.M
1170 CU.FT

MAX GROSS
TARE

30480 KG
67200 LB
2350 KG
5180 LB

MAX PAYLOAD
CU CAP

28130 KG
62020 LB
33.1 CU.M
1170 CU.FT

821197 0
2/G1

MGW.
TARE.

24 000 KGS
52 910 LBS
2 340 KGS
5 159 LBS

NET.
CU.CAP.

21 660 KGS
47 751 LBS
33.1 CU.M
1171 CU.FT

821197 0
2/G1

MGW.
TARE.

24 000 KGS
52 910 LBS
2 340 KGS
5 159 LBS

NET.
CU.CAP.

21 660 KGS
47 751 LBS
33.1 CU.M
1171 CU.FT

821197 0
2/G1

MGW.
TARE.

24 000 KGS
52 910 LBS
2 340 KGS
5 159 LBS

NET.
CU.CAP.

21 660 KGS
47 751 LBS
33.1 CU.M
1171 CU.FT

821197 0
2/G1

MGW.
TARE.

24 000 KGS
52 910 LBS
2 340 KGS
5 159 LBS

NET.
CU.CAP.

21 660 KGS
47 751 LBS
33.1 CU.M
1171 CU.FT

821197 0
2/G1

MGW.
TARE.

24 000 KGS
52 910 LBS
2 340 KGS
5 159 LBS

NET.
CU.CAP.

21 660 KGS
47 751 LBS
33.1 CU.M
1171 CU.FT

821197 0
2/G1

MGW.
TARE.

24 000 KGS
52 910 LBS
2 340 KGS
5 159 LBS

NET.
CU.CAP.

21 660 KGS
47 751 LBS
33.1 CU.M
1171 CU.FT

MAX GROSS
TARE

30480 KG
67200 LB
2350 KG
5180 LB

MAX PAYLOAD
CU CAP

28130 KG
62020 LB
33.1 CU.M
1170 CU.FT

MAX GROSS
TARE

30480 KG
67200 LB
2350 KG
5180 LB

MAX PAYLOAD
CU CAP

28130 KG
62020 LB
33.1 CU.M
1170 CU.FT

MAX GROSS
TARE

30480 KG
67200 LB
2350 KG
5180 LB

MAX PAYLOAD
CU CAP

28130 KG
62020 LB
33.1 CU.M
1170 CU.FT

MAX GROSS
TARE

30480 KG
67200 LB
2350 KG
5180 LB

MAX PAYLOAD
CU CAP

28130 KG
62020 LB
33.1 CU.M
1170 CU.FT

MAX GROSS
TARE

30480 KG
67200 LB
2350 KG
5180 LB

MAX PAYLOAD
CU CAP

28130 KG
62020 LB
33.1 CU.M
1170 CU.FT

MAX GROSS
TARE

30480 KG
67200 LB
2350 KG
5180 LB

MAX PAYLOAD
CU CAP

28130 KG
62020 LB
33.1 CU.M
1170 CU.FT

MAX GROSS
TARE

30480 KG
67200 LB
2350 KG
5180 LB

MAX PAYLOAD
CU CAP

28130 KG
62020 LB
33.1 CU.M
1170 CU.FT

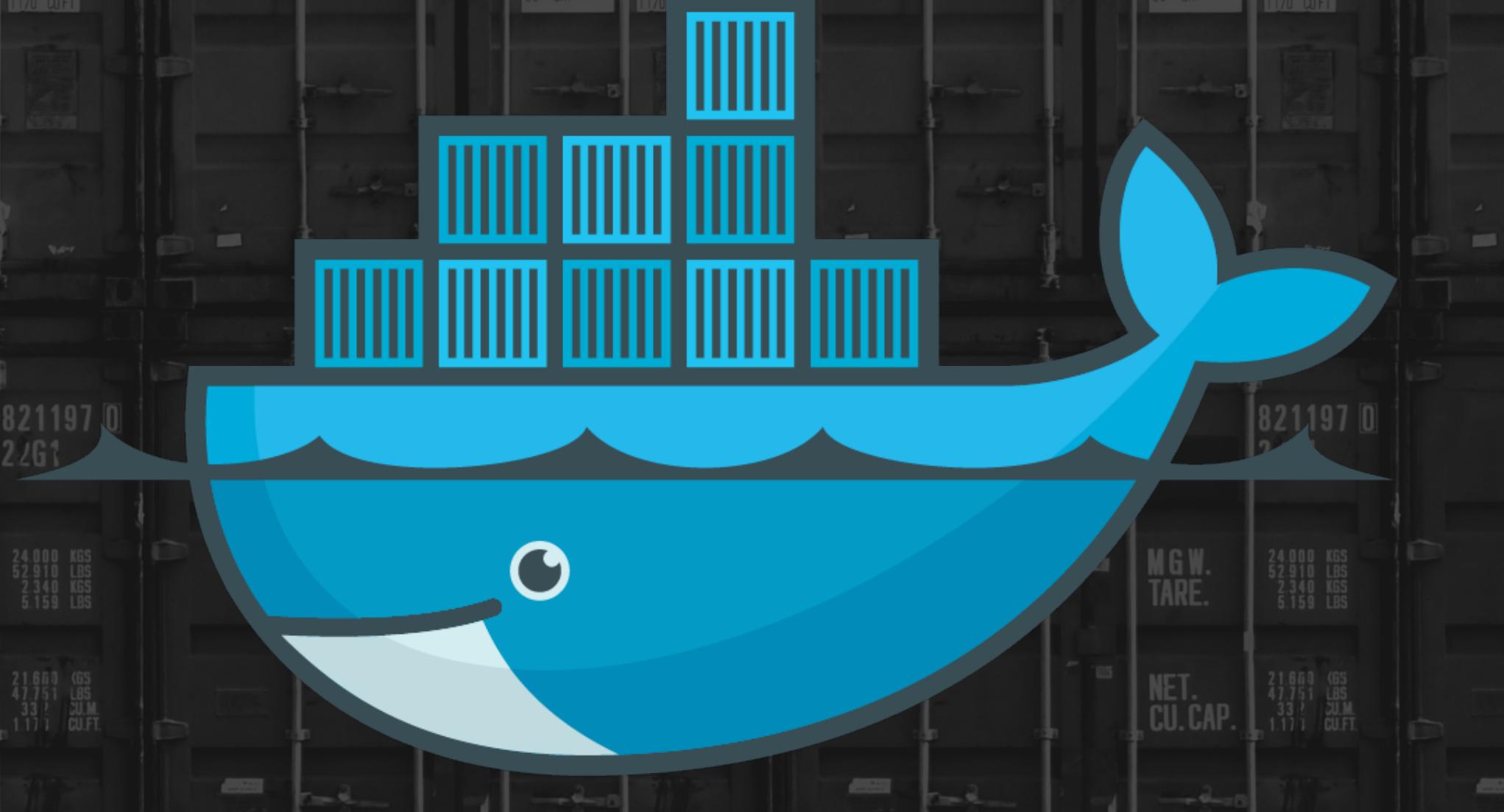


docker

docker

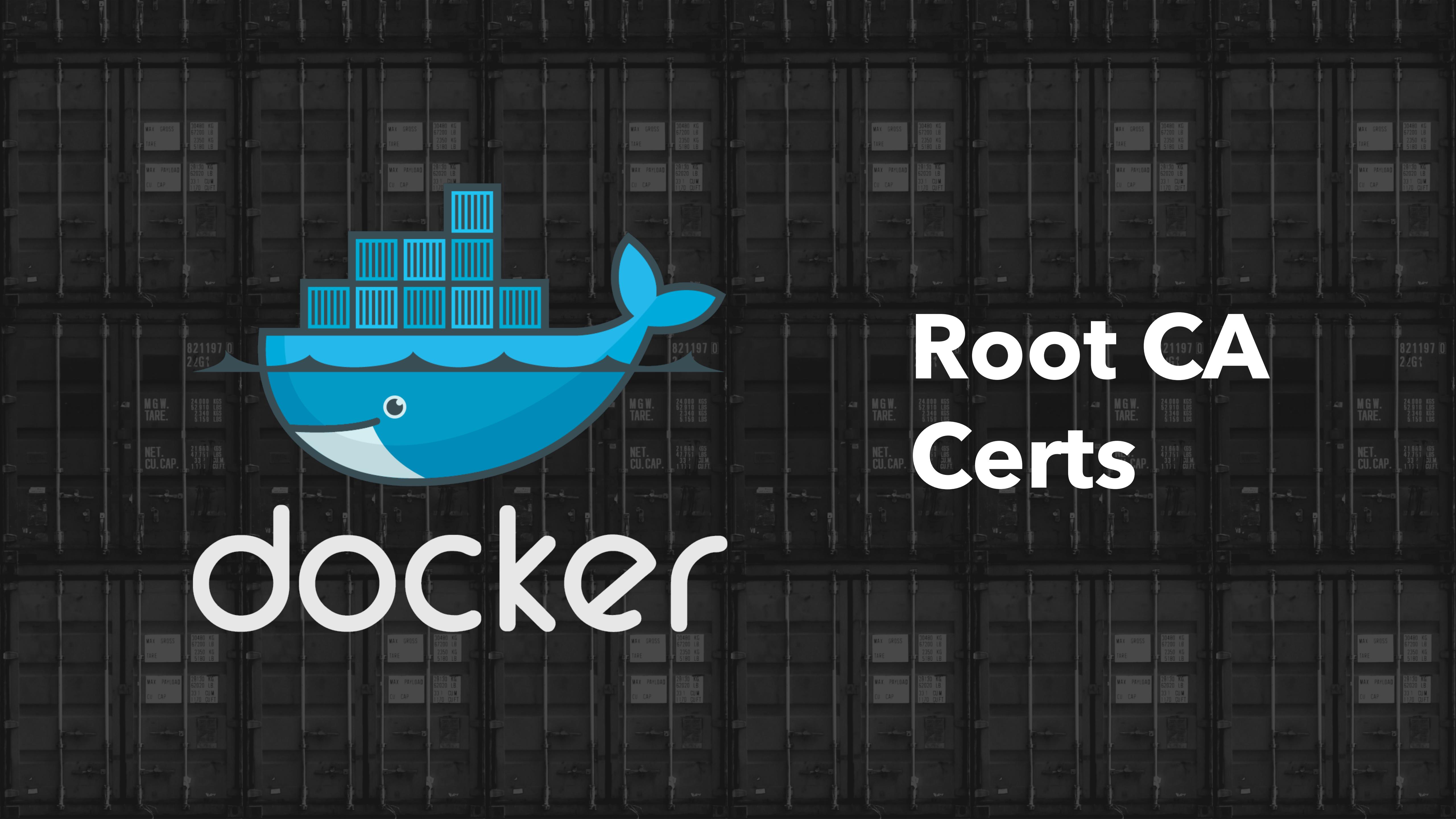


Scratch images



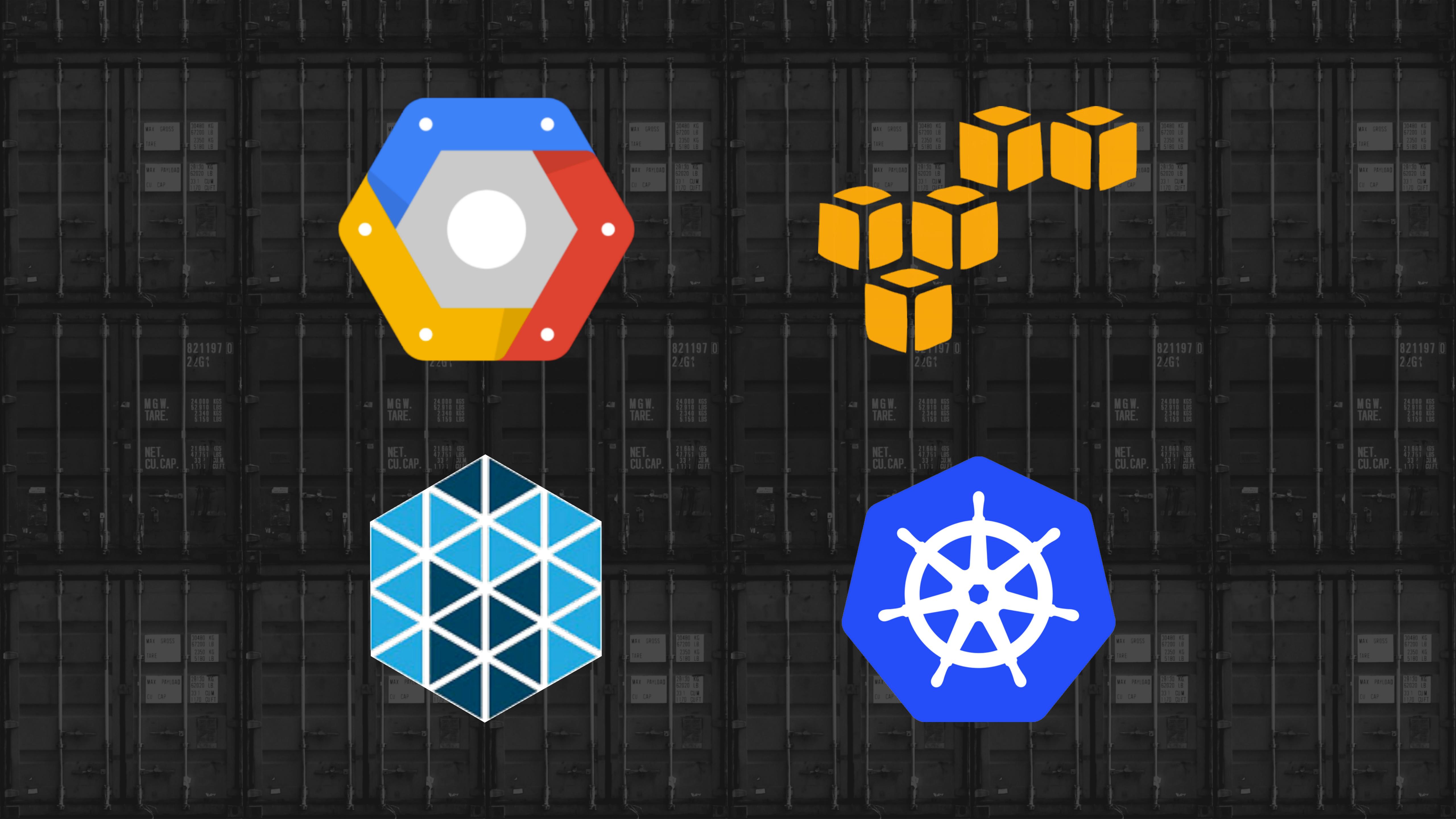
docker

Statically Compiled

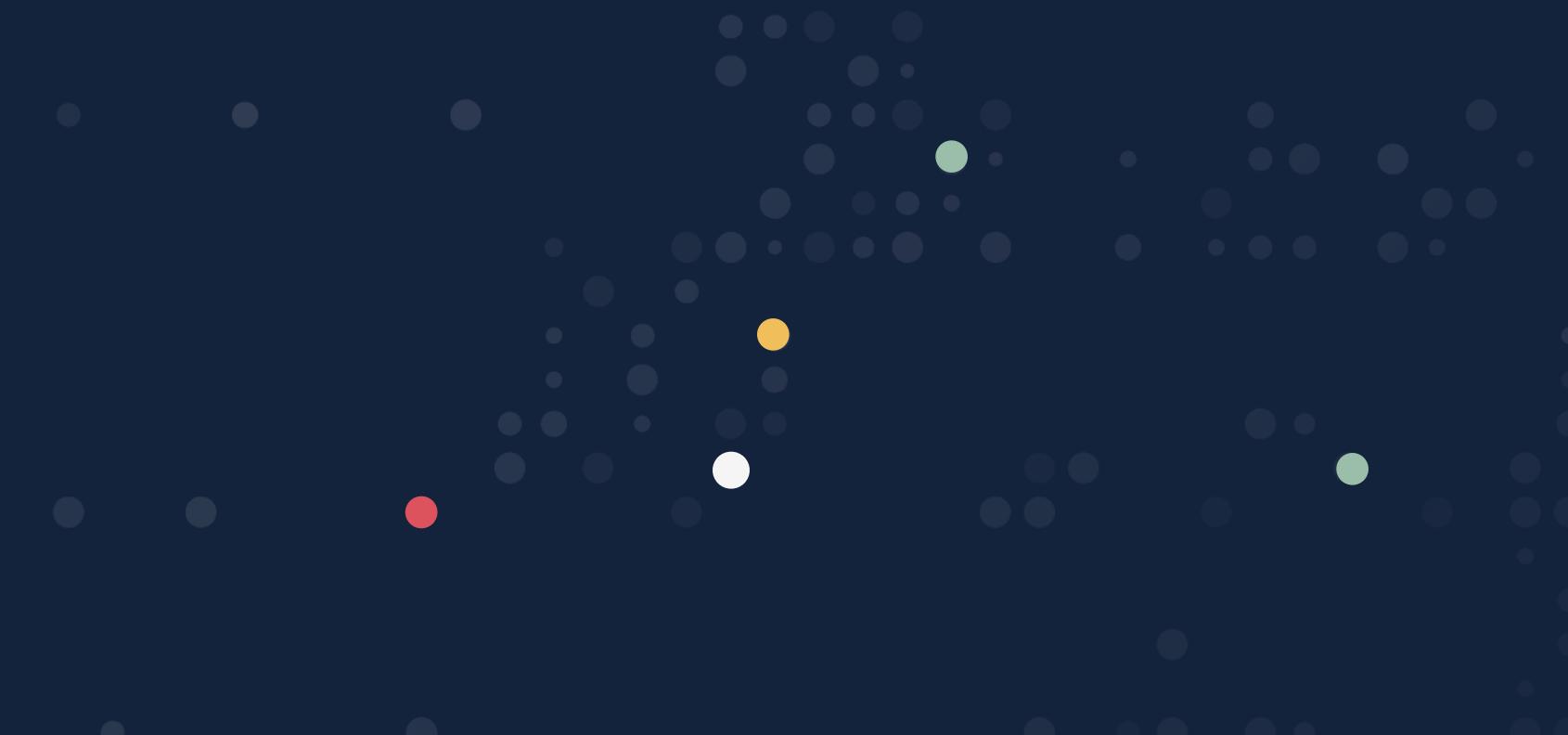


docker

Root CA Certs



Dealing with Complexity



Testing



**Load
Failure
Degradation**



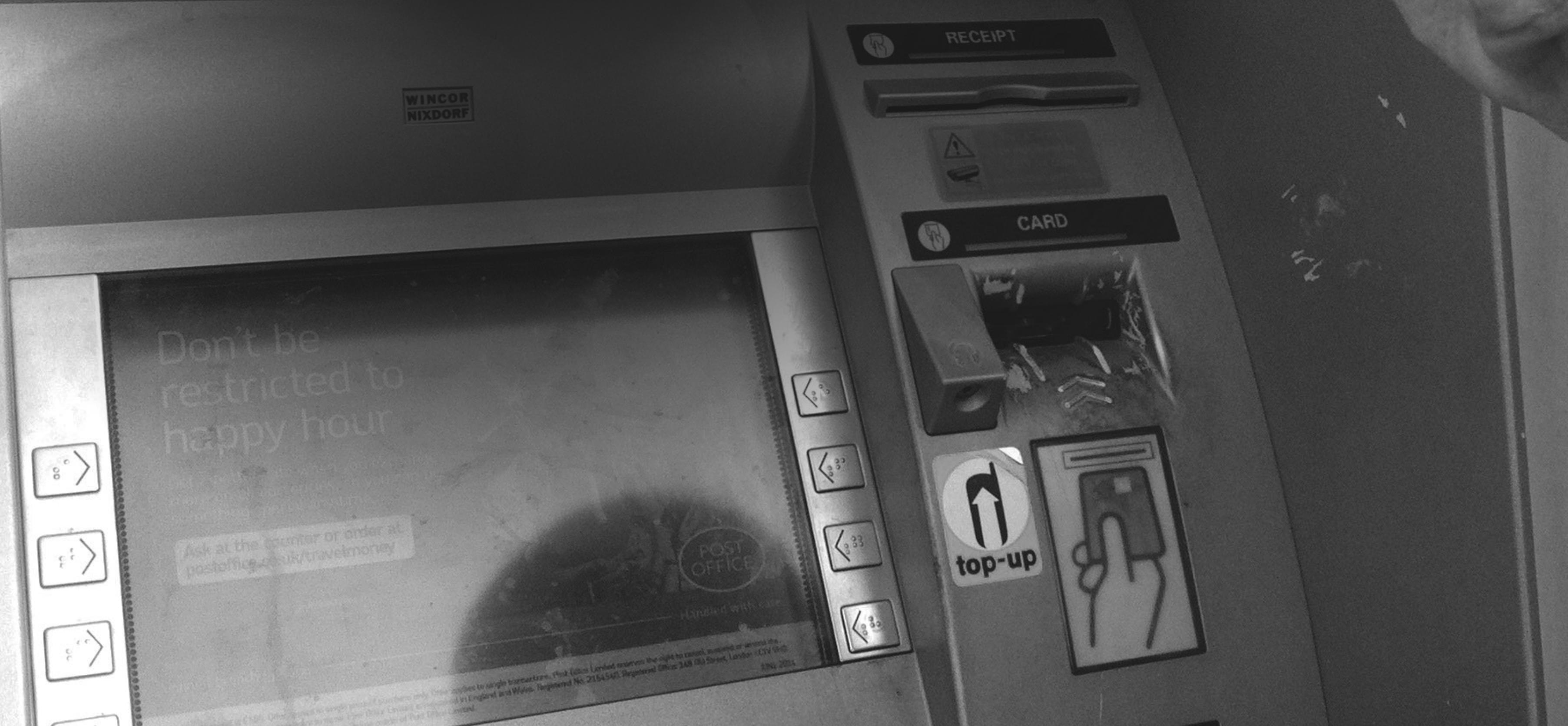
Monitoring



Monitor your Business Logic



customers



In-Band vs Out of Band

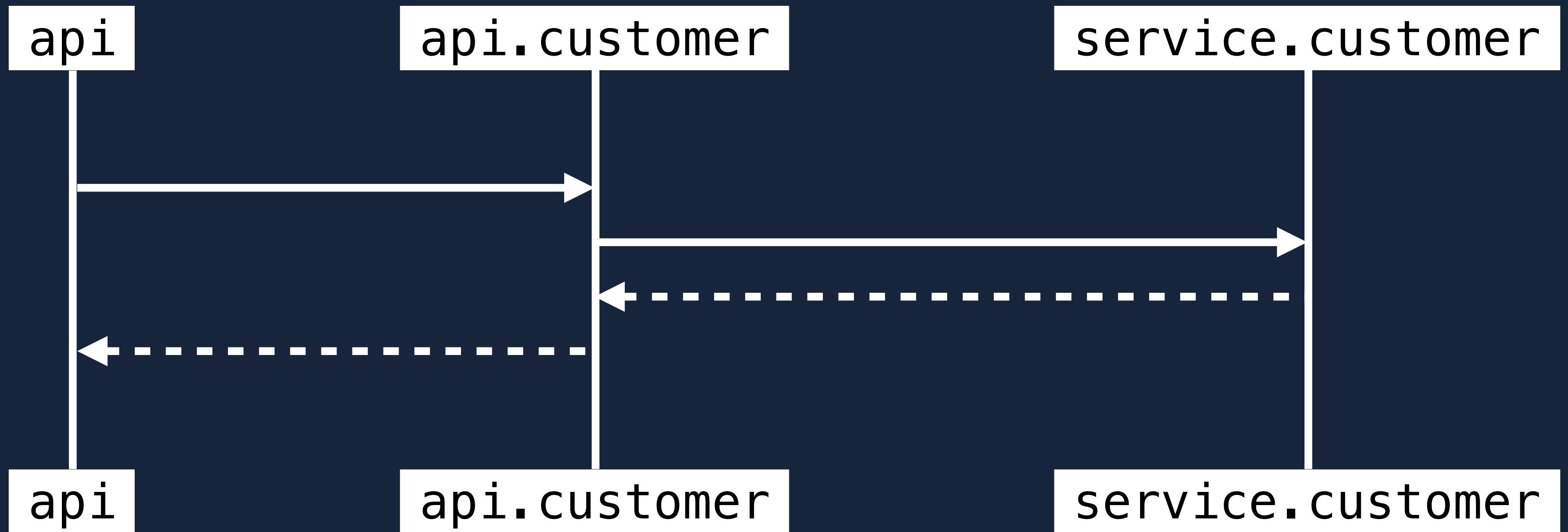


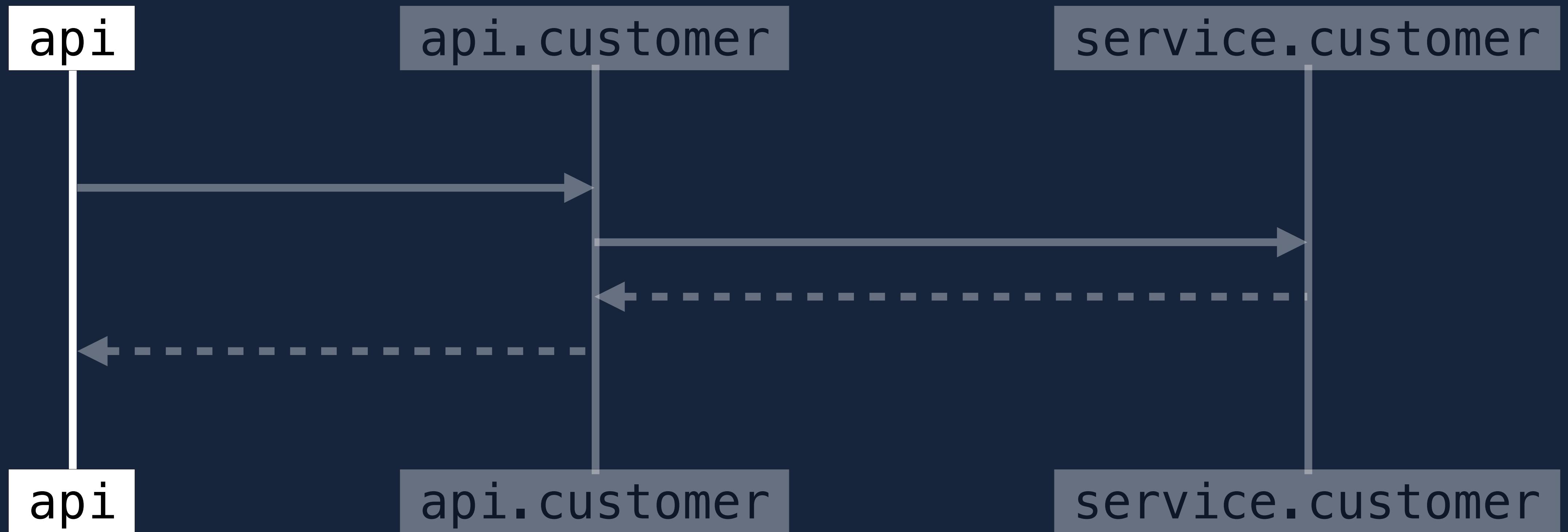


```
type Checker func() (error, map[string]string)
```

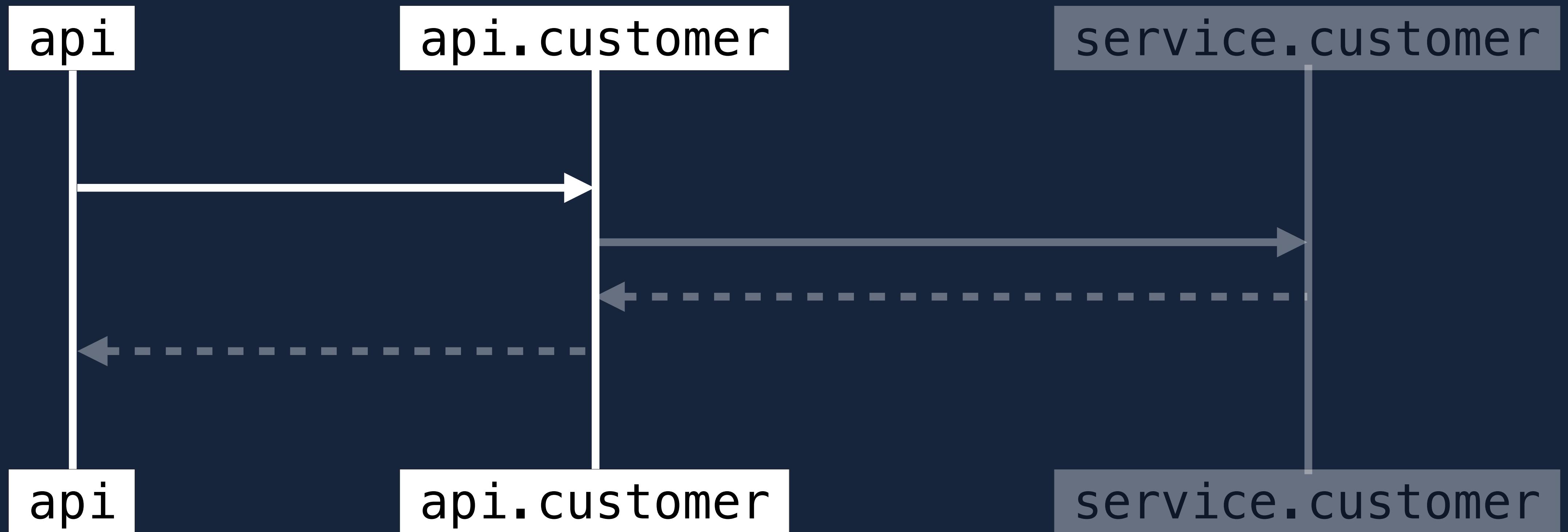
Distributed Tracing

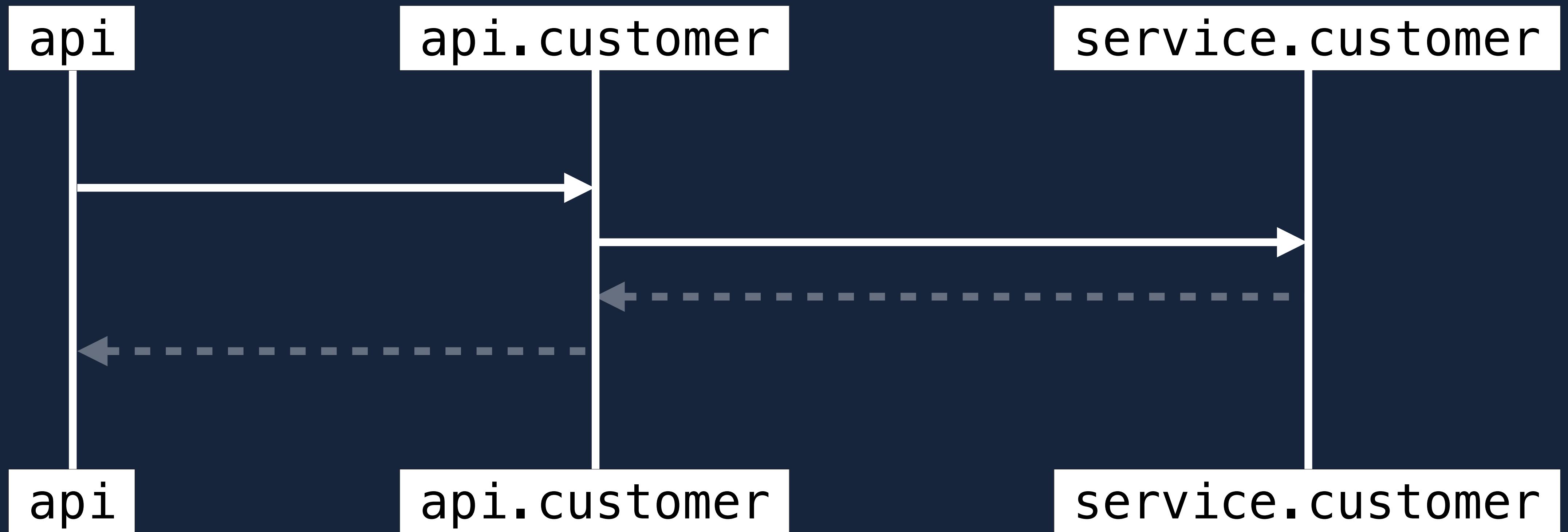






8096820c-3b7b-47ec-bce6-1c239252ab40





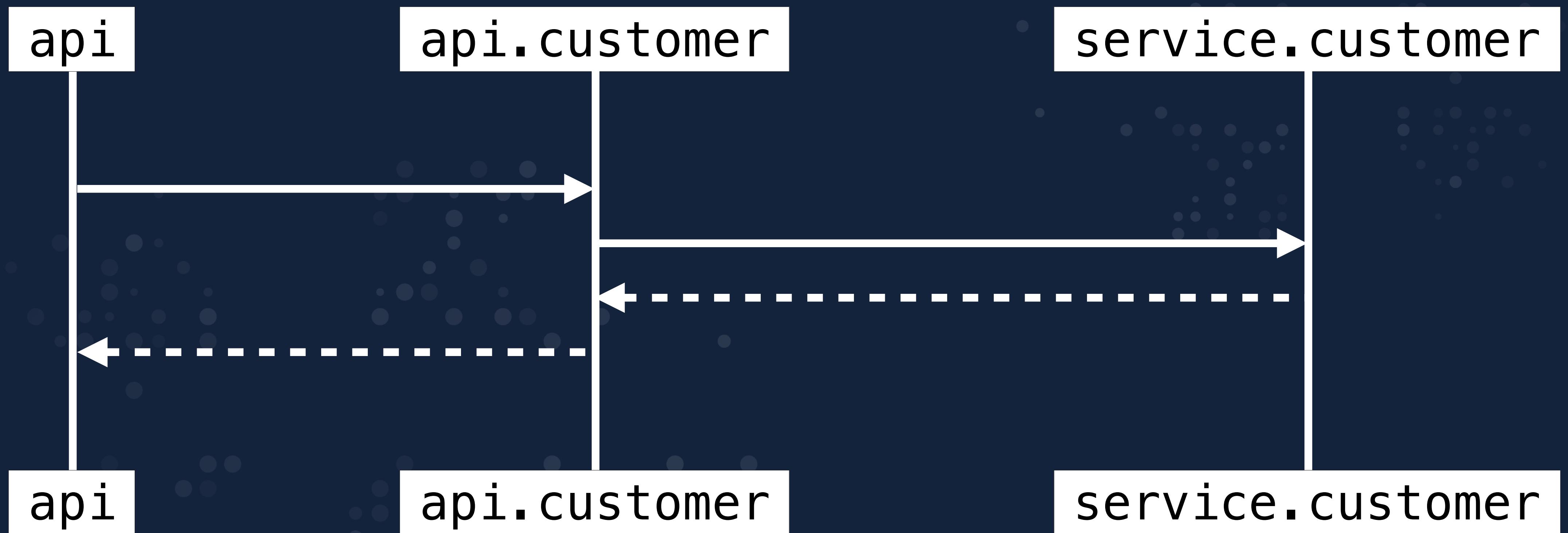
Context Propagation

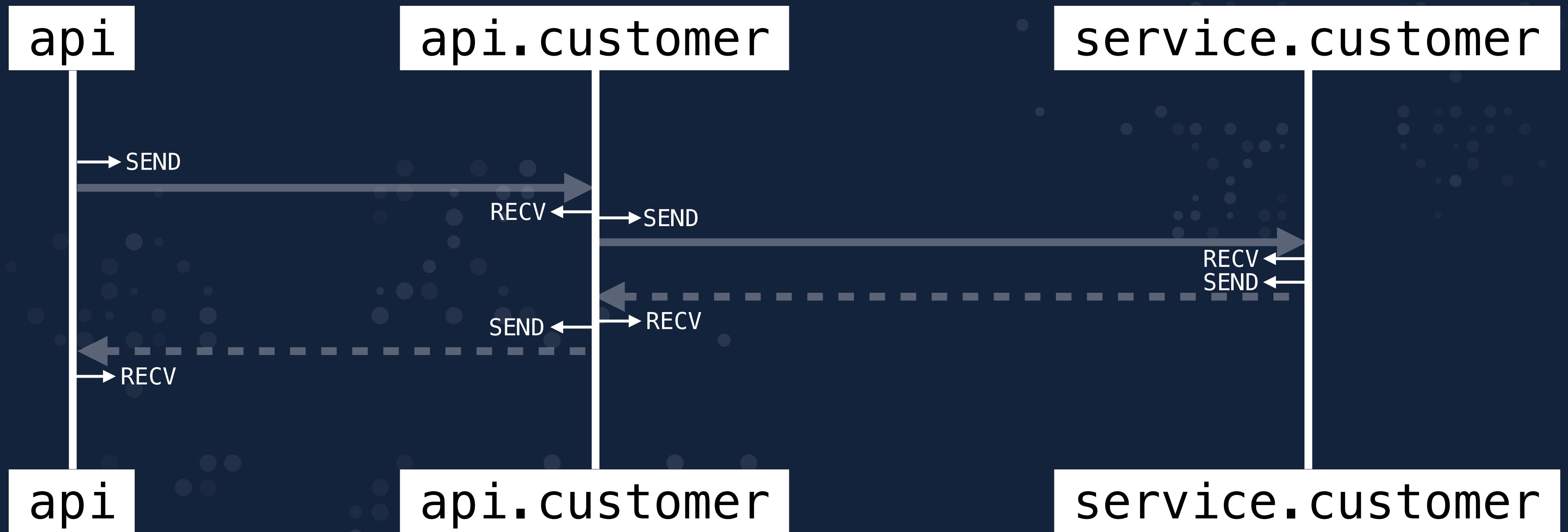
```
package context
```

```
type Context interface {
    Deadline() (deadline time.Time, ok bool)
    Done() <-chan struct{}
    Err() error
    Value(key interface{}) interface{}
}
```

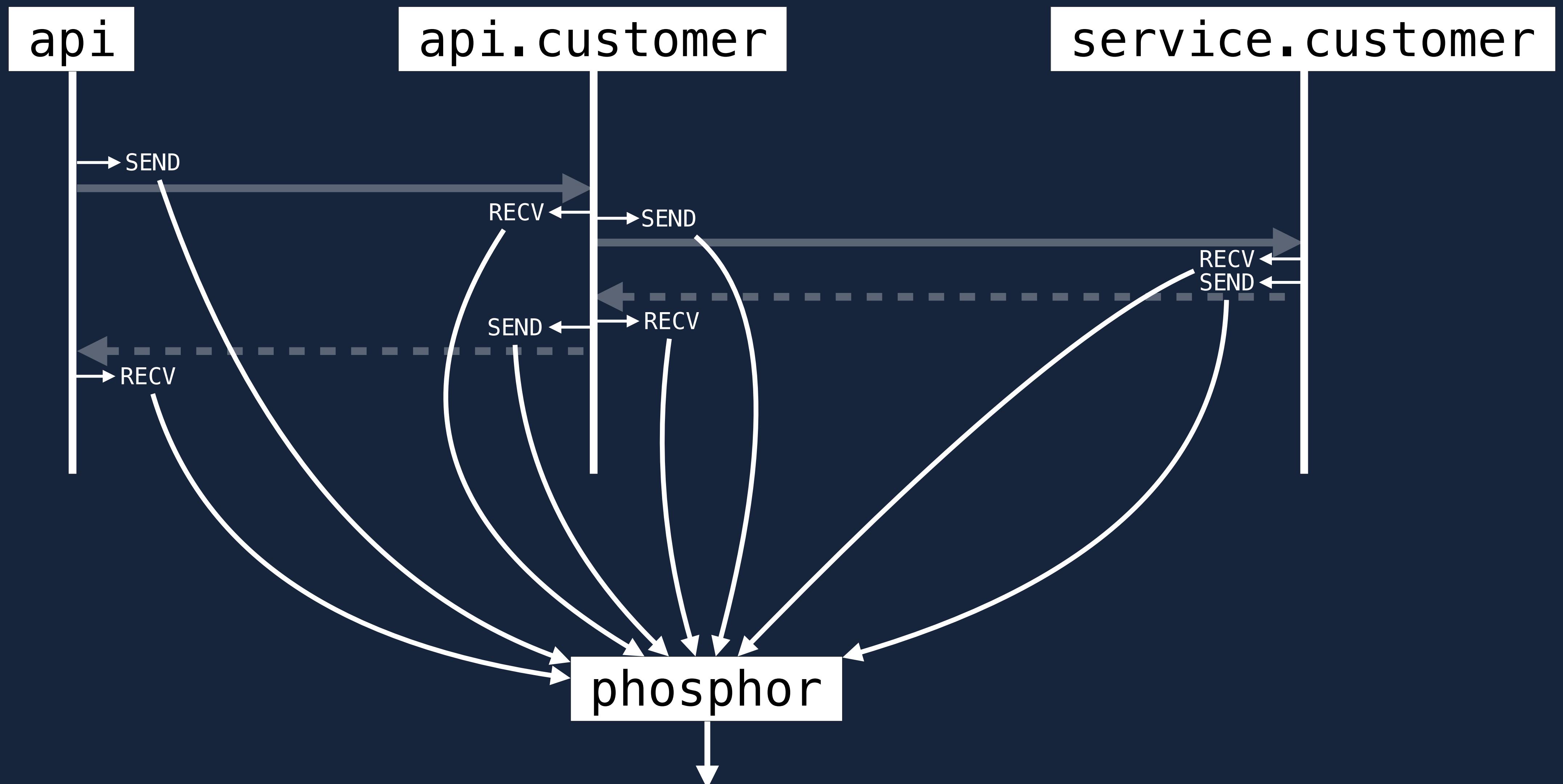
```
package context
```

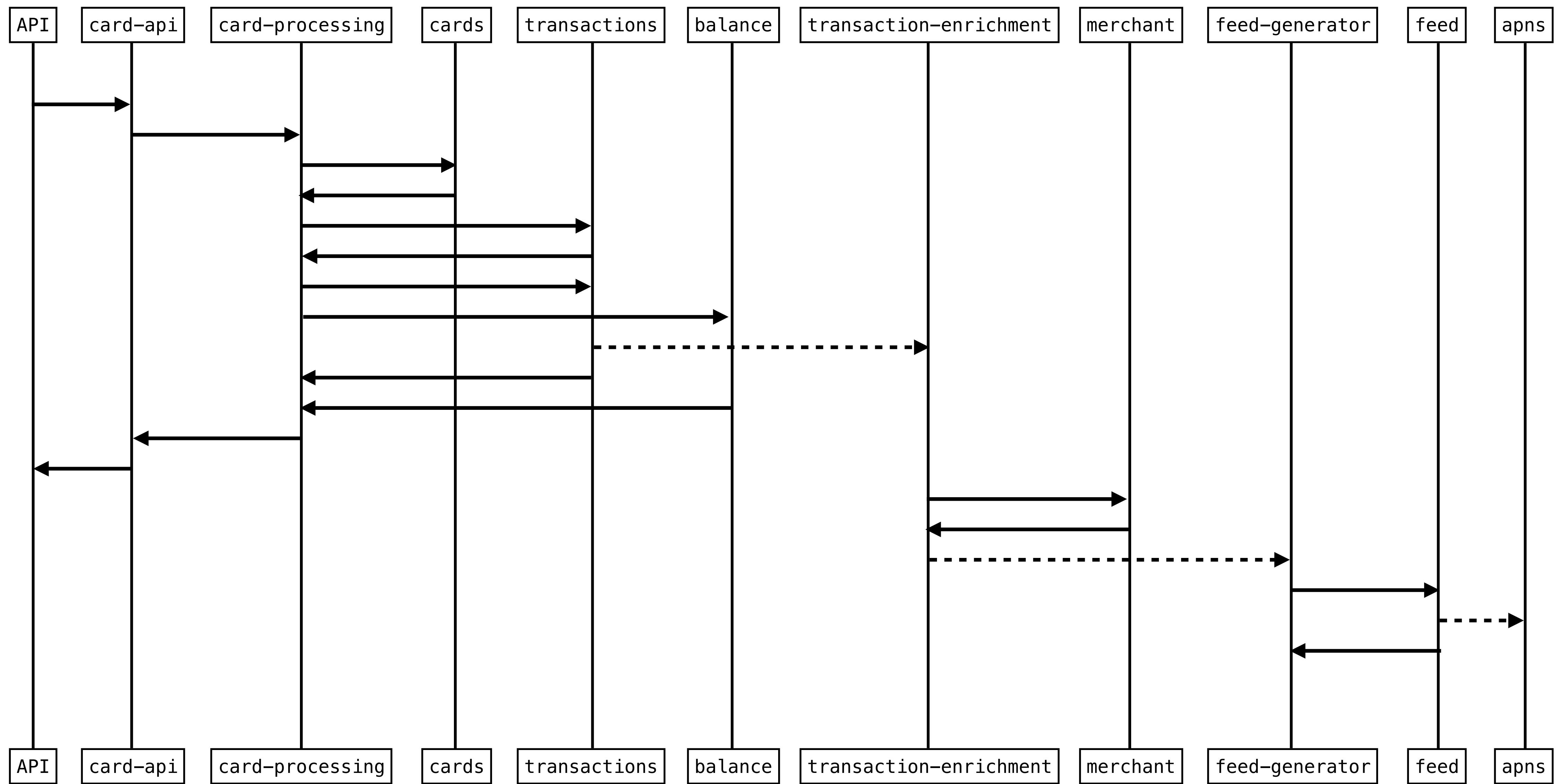
```
type Context interface {
    Deadline() (deadline time.Time, ok bool)
    Done() <-chan struct{}
    Err() error
    Value(key interface{}) interface{}
}
```

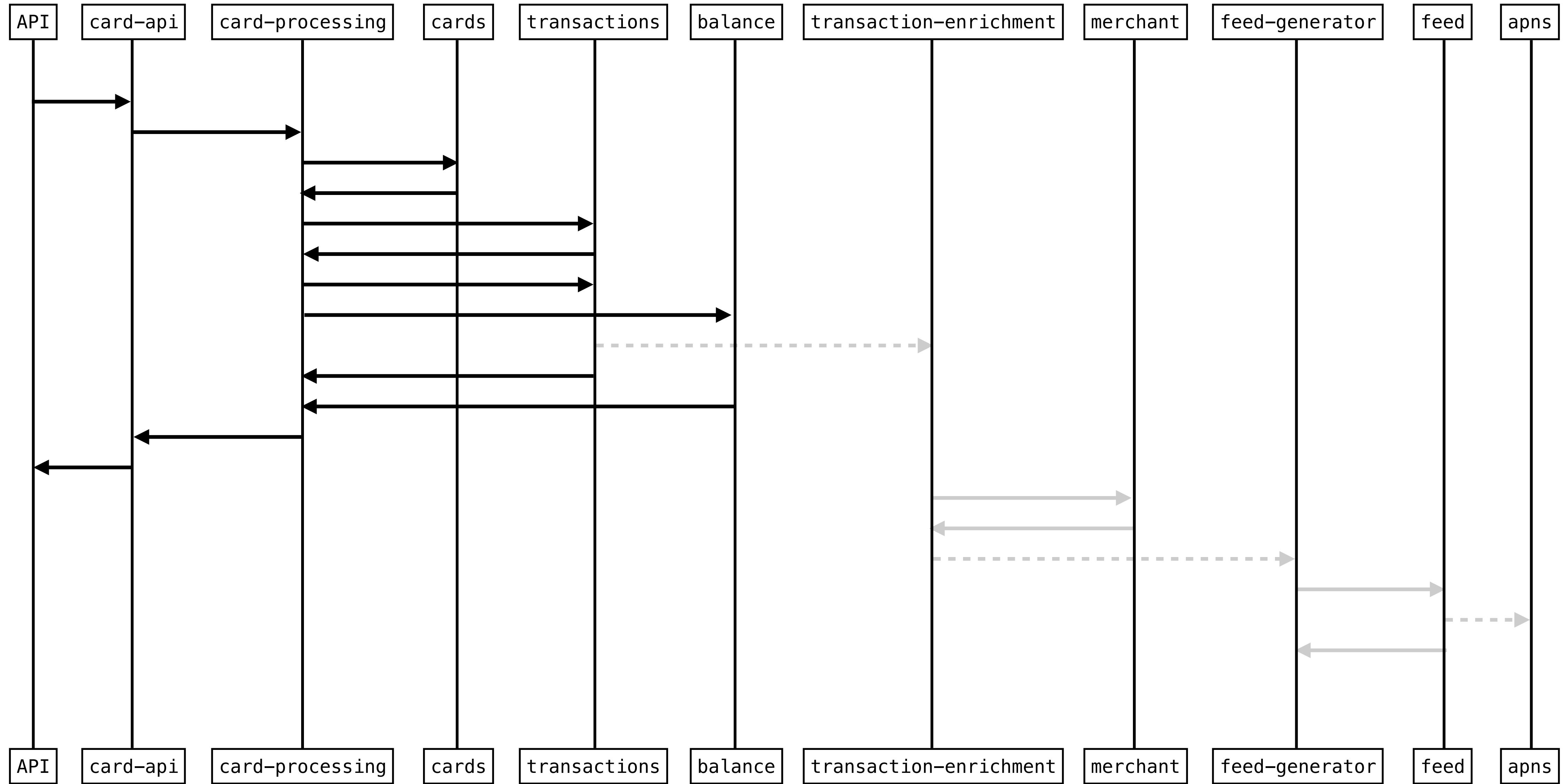


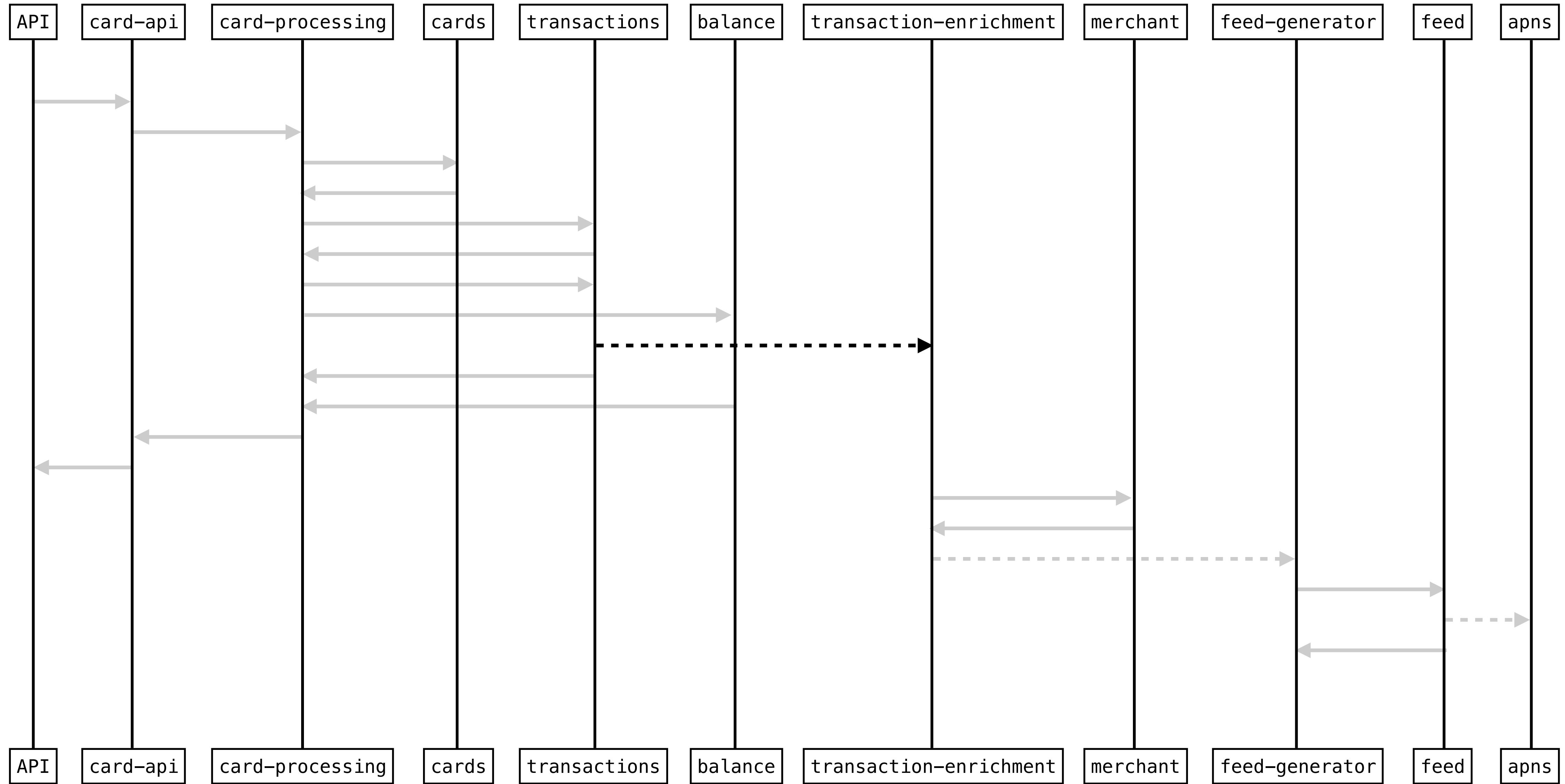


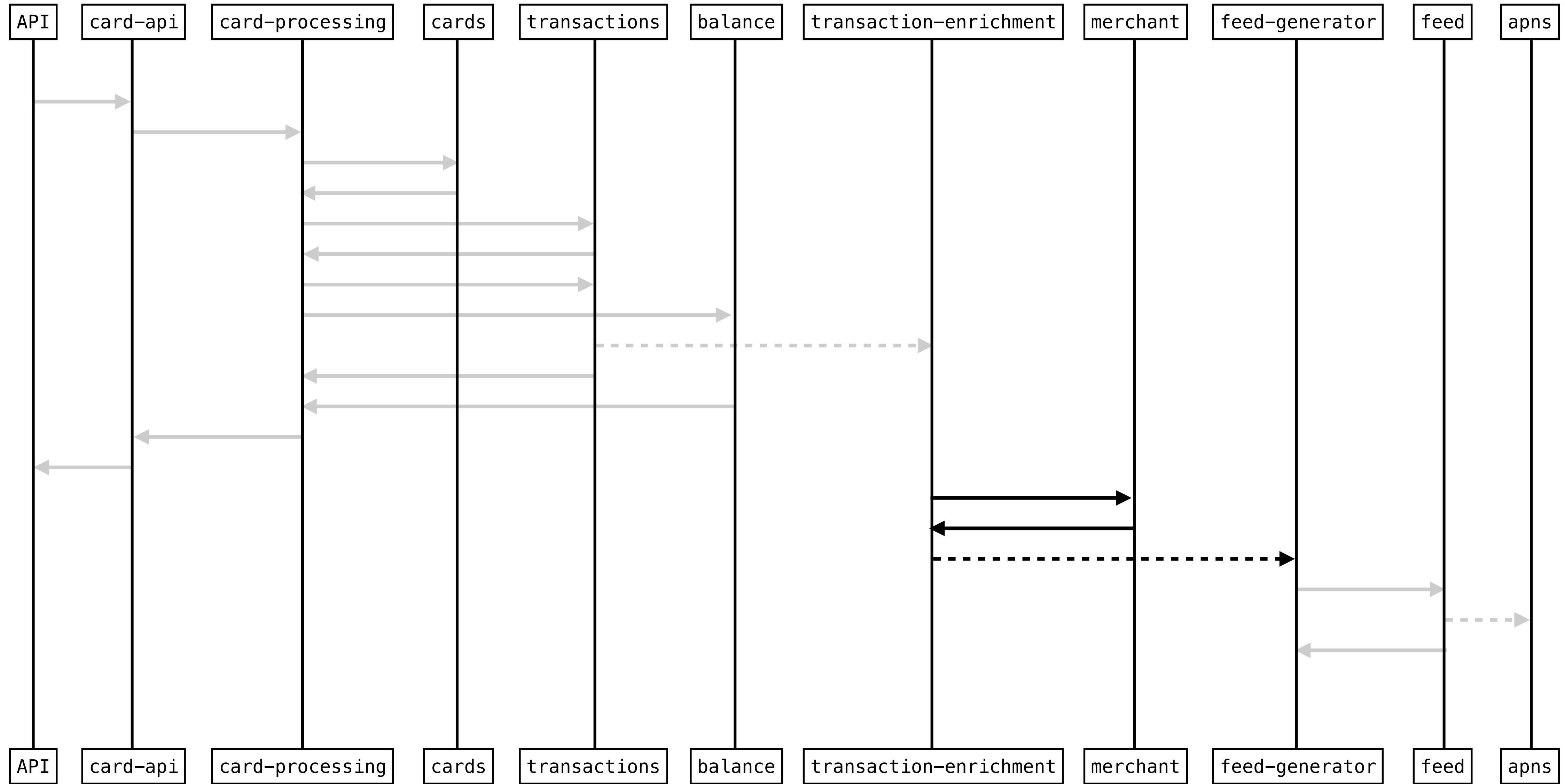
mondough/phosphor

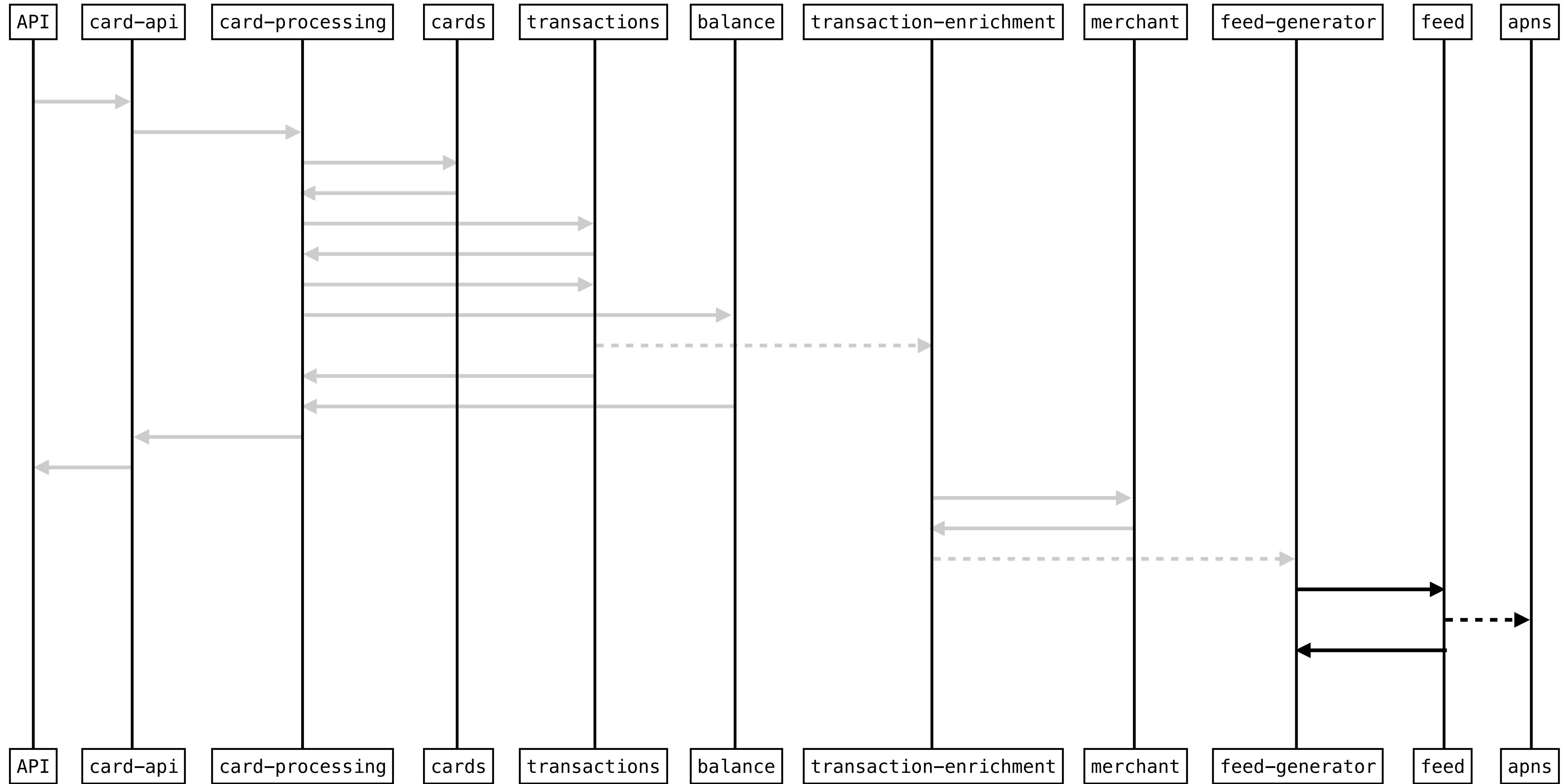




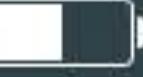








●●●○○ Optus 4G

58% 

11:36

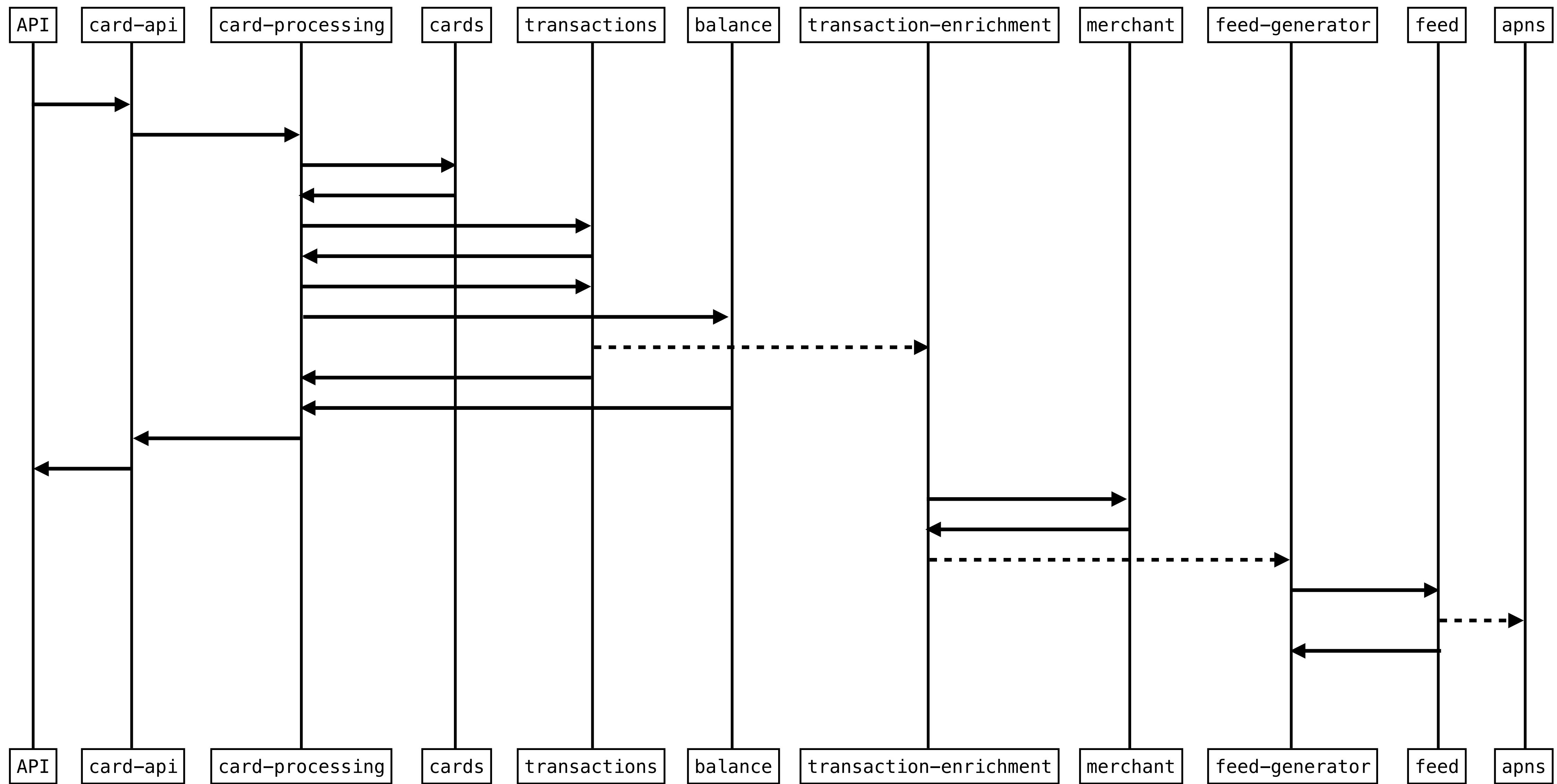
Tuesday, 1 December

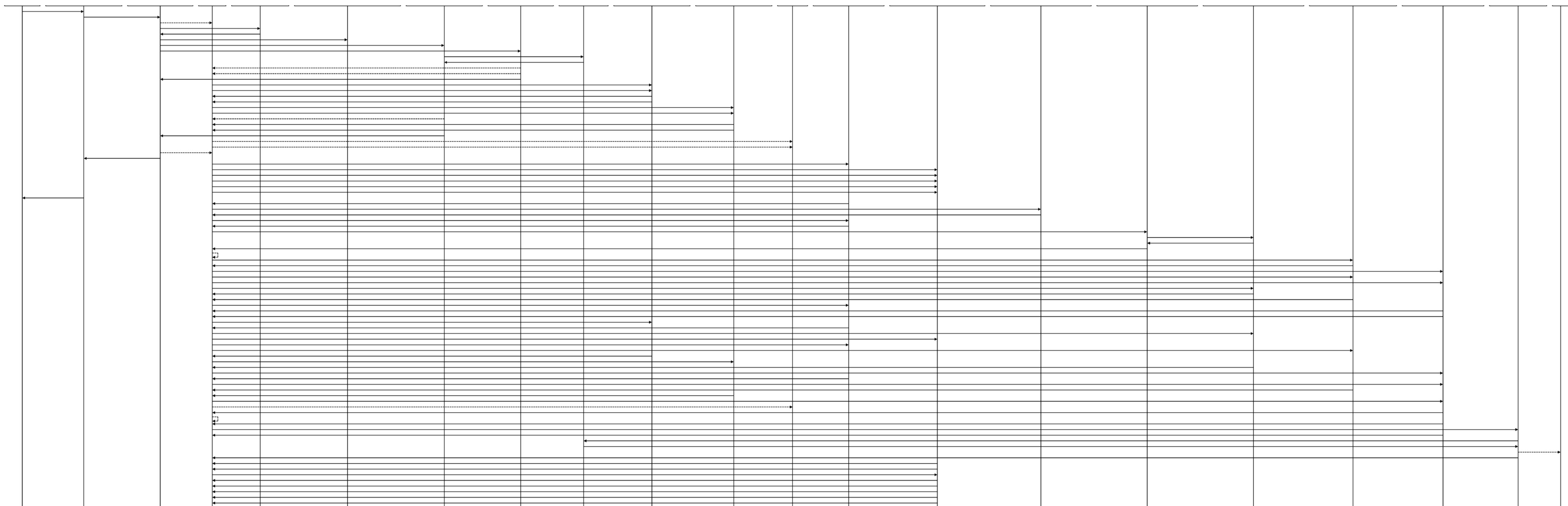


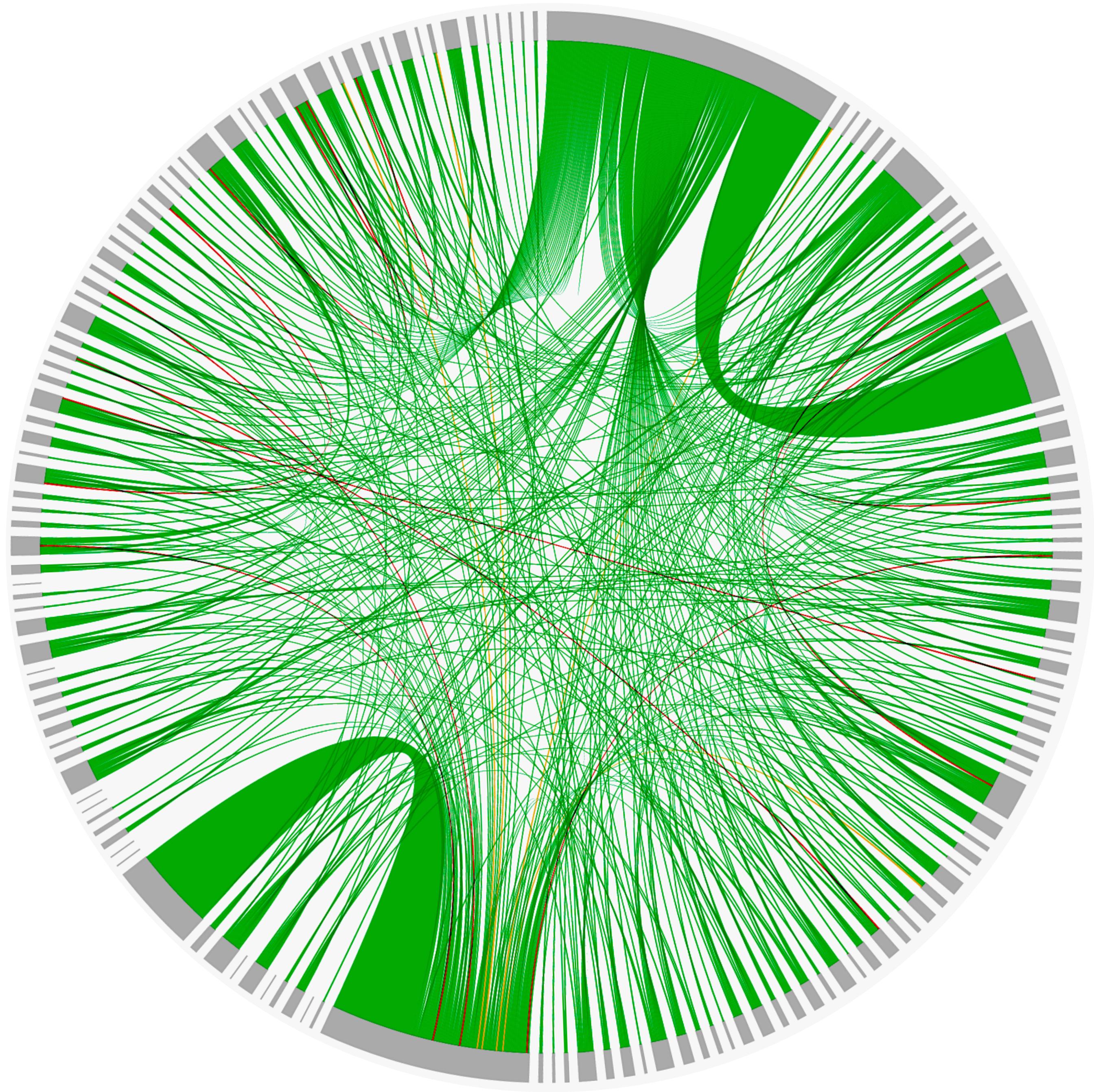
Mondo now

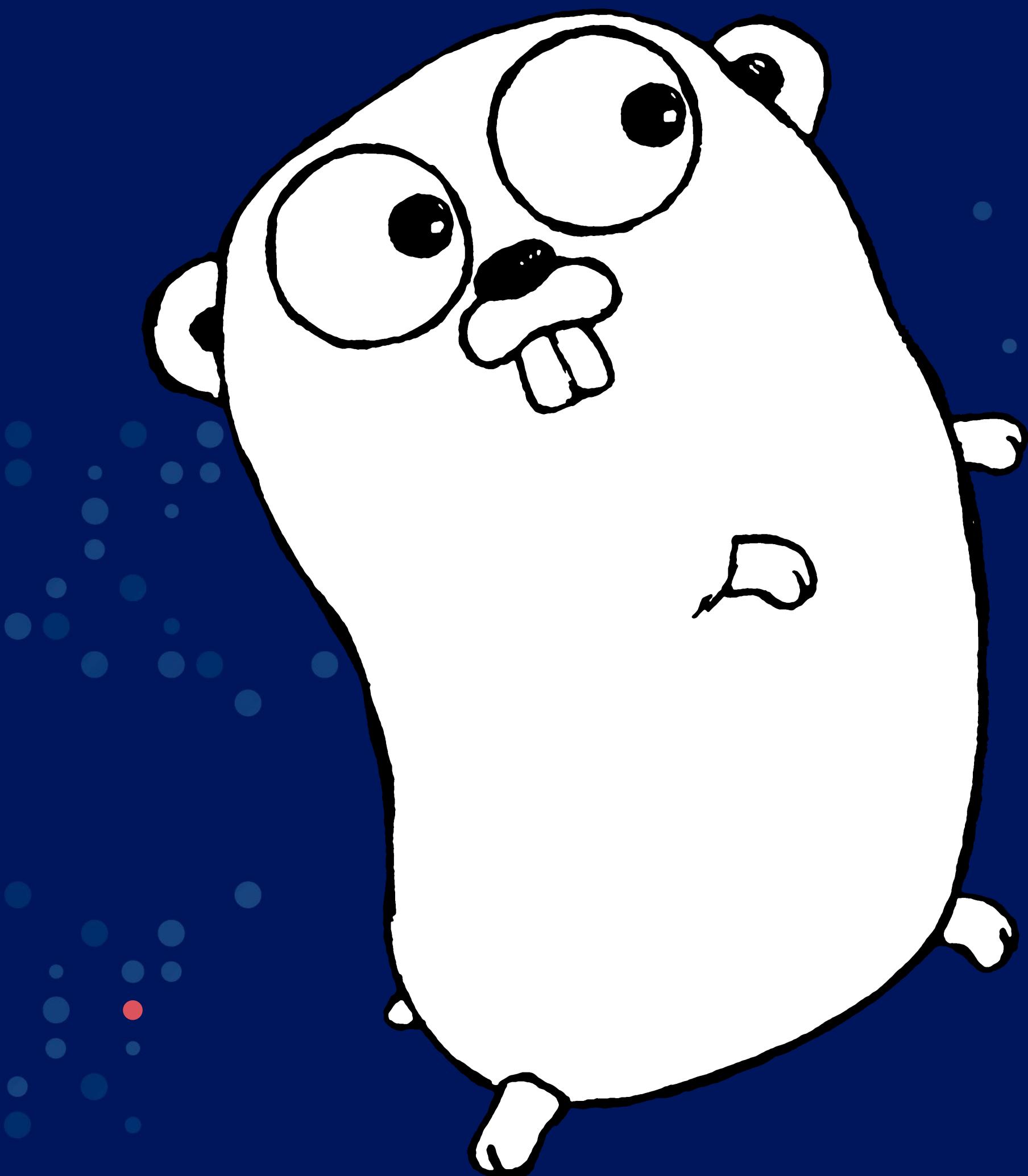
☕ \$19.15 at Brightbird Espresso.
You've spent \$43.55 today.

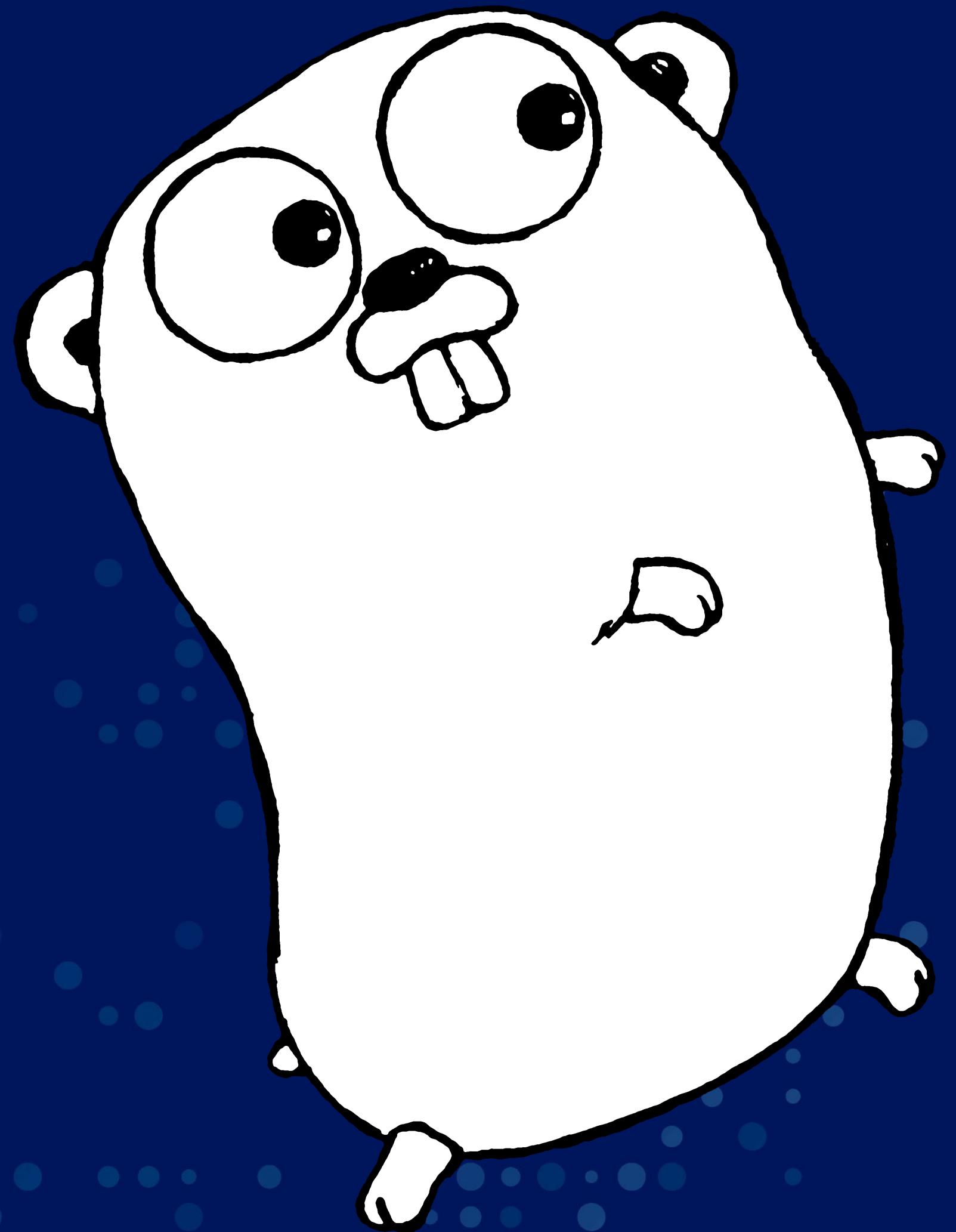
[slide to view](#)



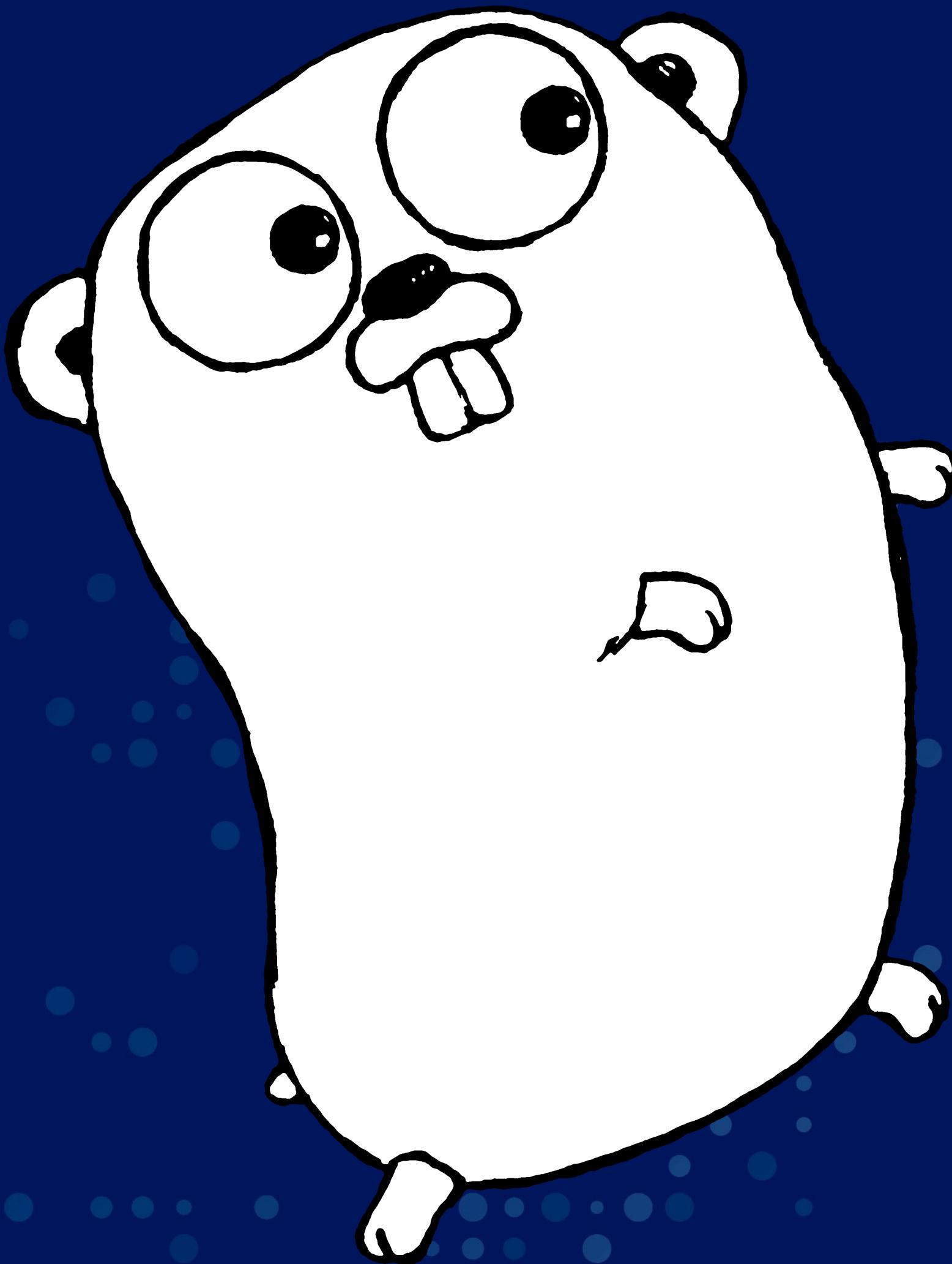




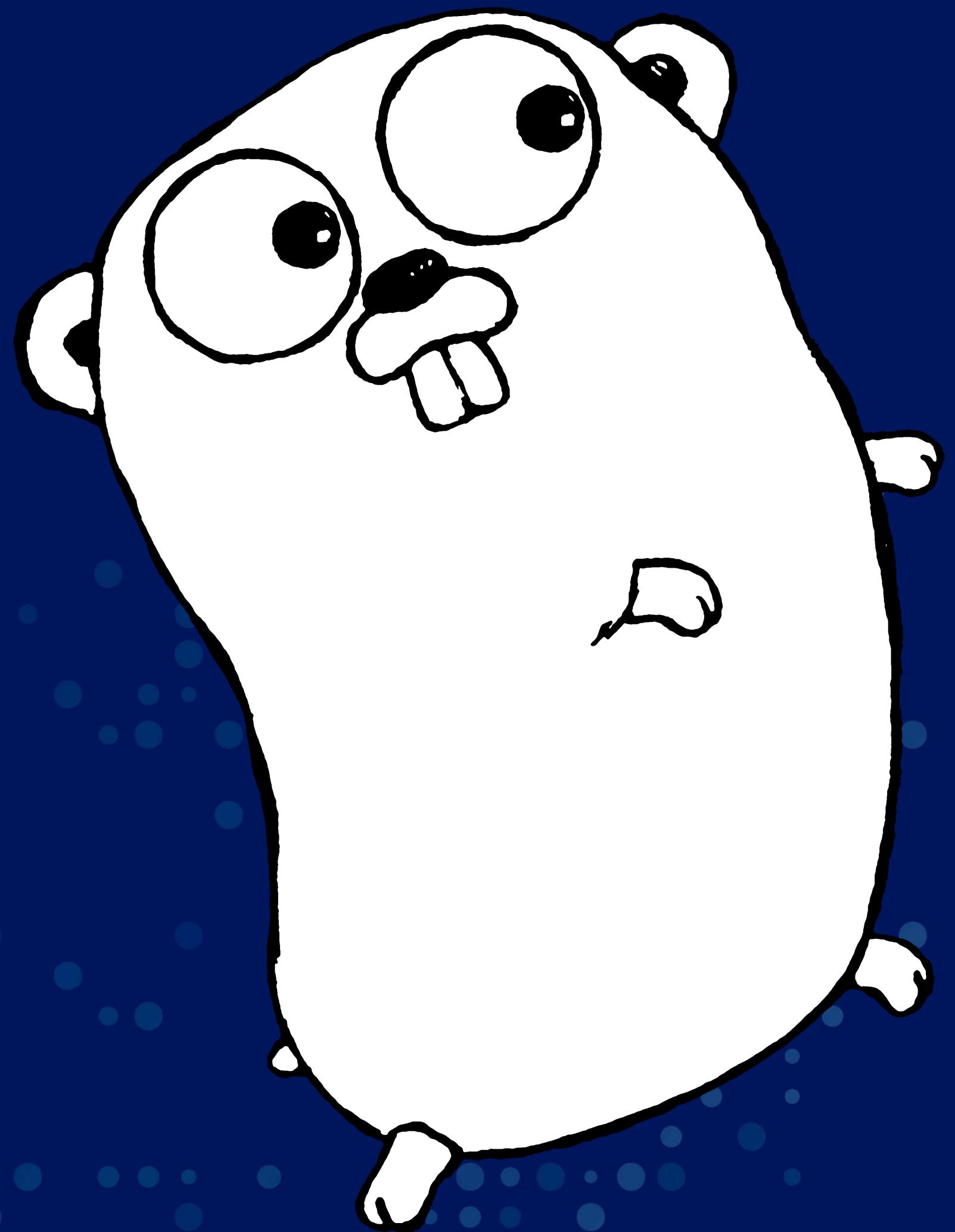




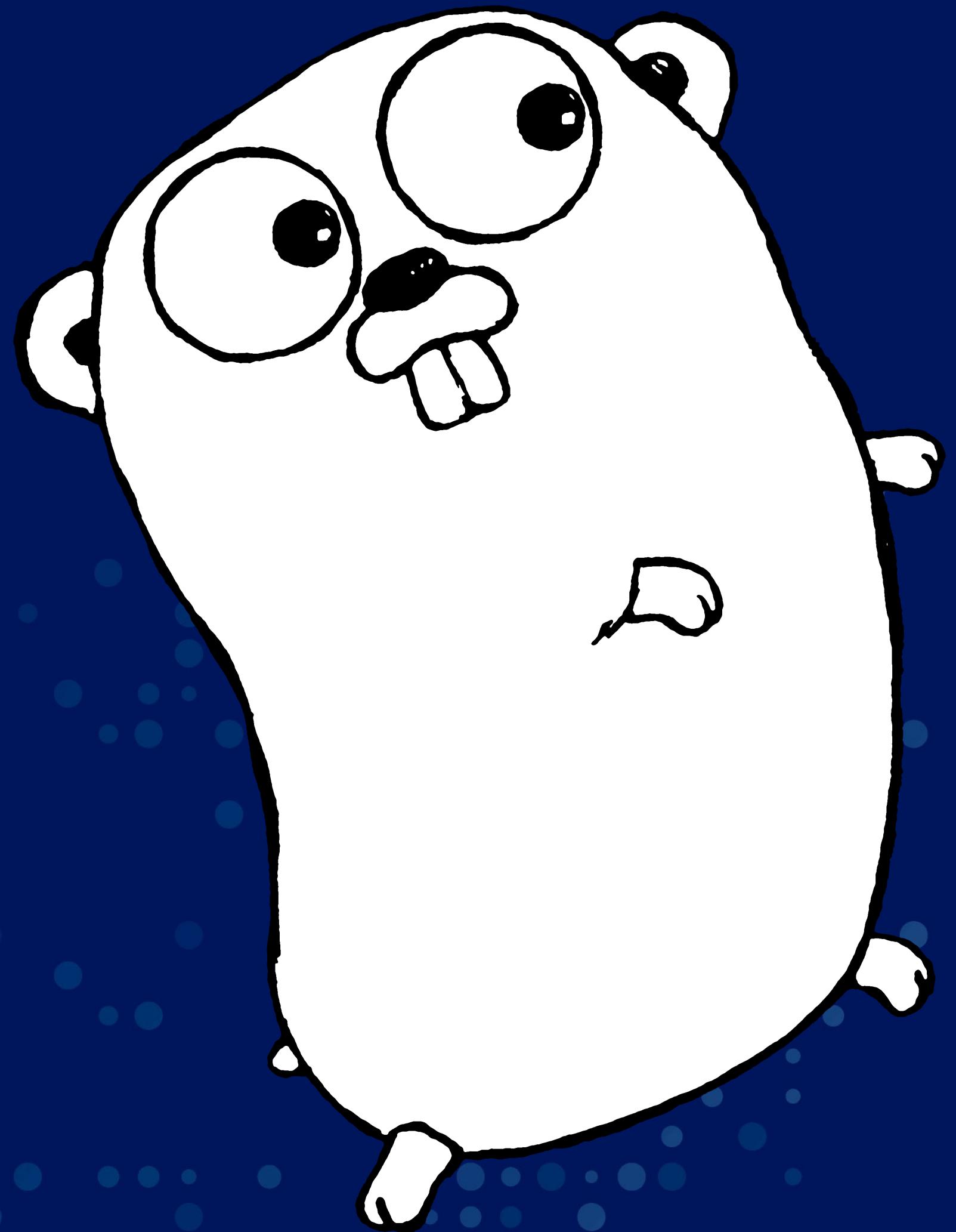
Small
Simple
Easy to learn



Concurrency Interfaces Networking



Downsides?



Starting with
Microservices?

Thanks!

@mattheath
@getmondo

#yow2015



Credits

ATM: Thomas Hawk

IBM System/360: IBM

Absorbed: Saxbald Photography

Orbital Ion Cannon: www.rom.ac

Go Gophers: Renee French

Control Room: NASA

ATM Failure: George Redgrave