riak cs

riak

# Reid Draper

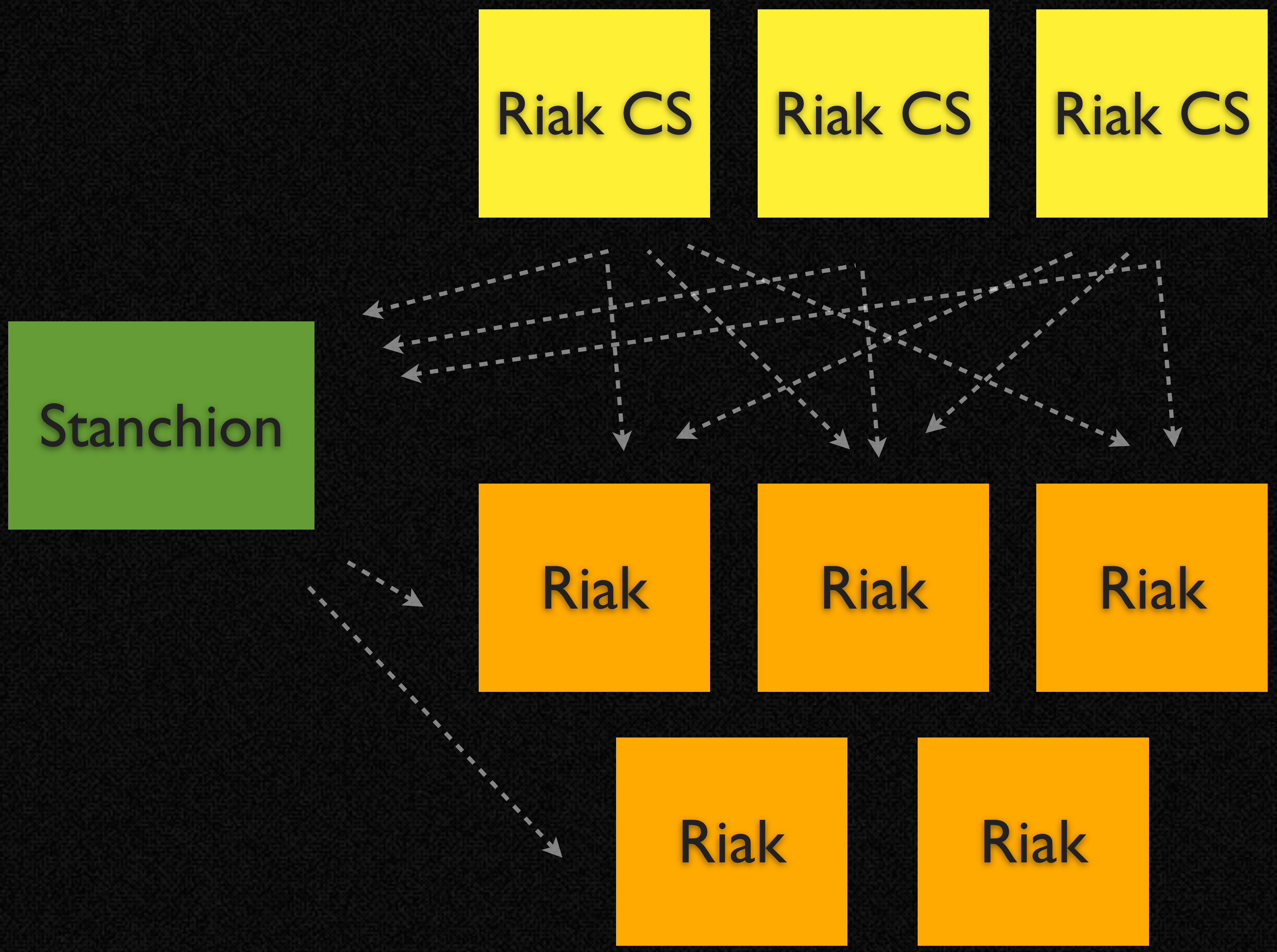
@reiddraper
reid@basho.com

Riak CS is an Erlang
application
that exposes the S3 API on
top of Riak

# Apache 2 Licensed

(as of yesterday!)

webmachine lager riakc poolboy
druuid poolboy folsom sext eper


Riak CS
Riak CS
Riak CS
Stanchion
Riak
Riak
Riak
Riak
Riak

immutability
backpressure
webmachine
eqc
503
records

# immutability

manifest

content-type

content-length

md5

blocks-written

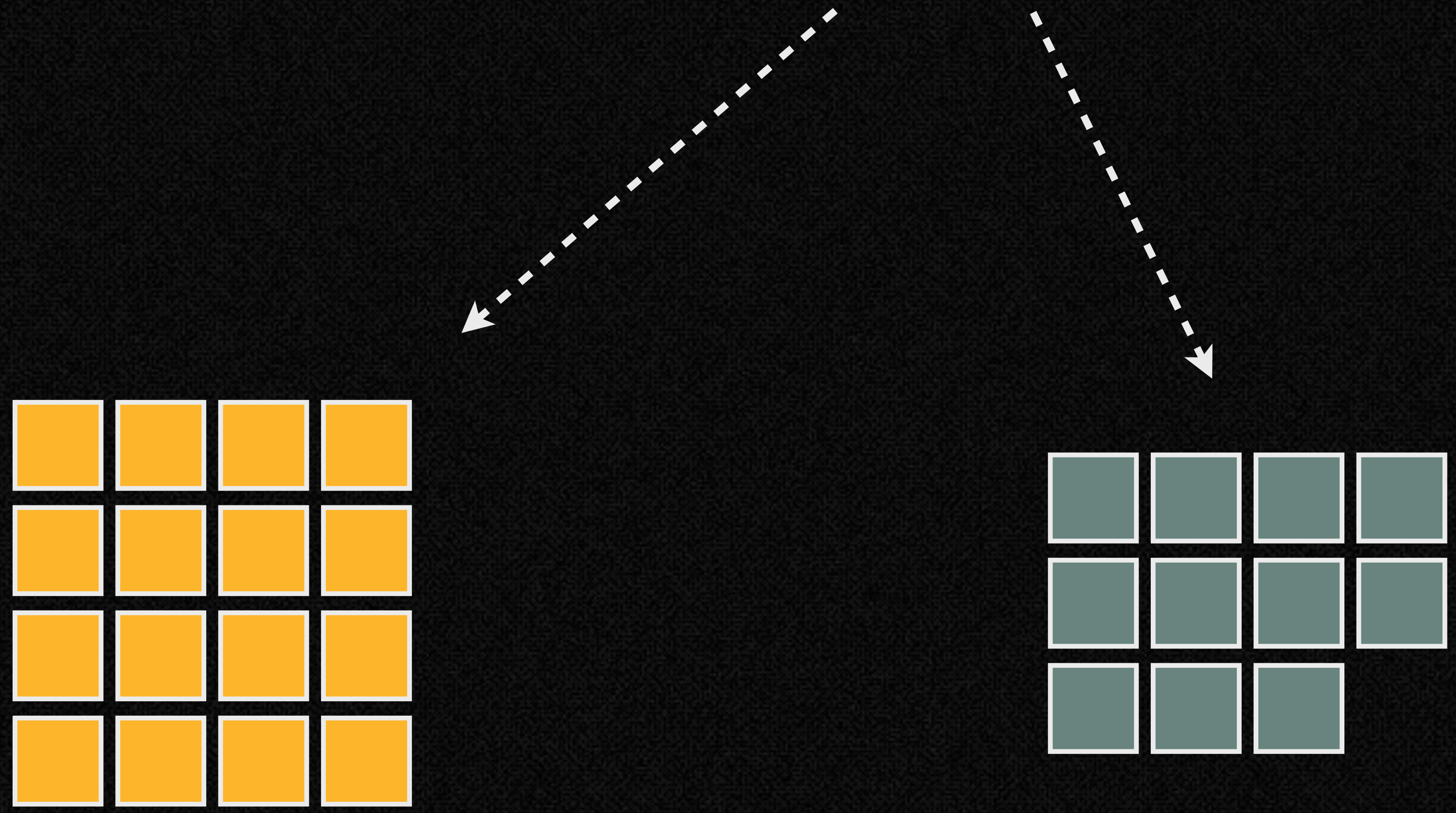
block-size

manifest version

acl

cluster-id

[2fb5bb24, 3e908b88]

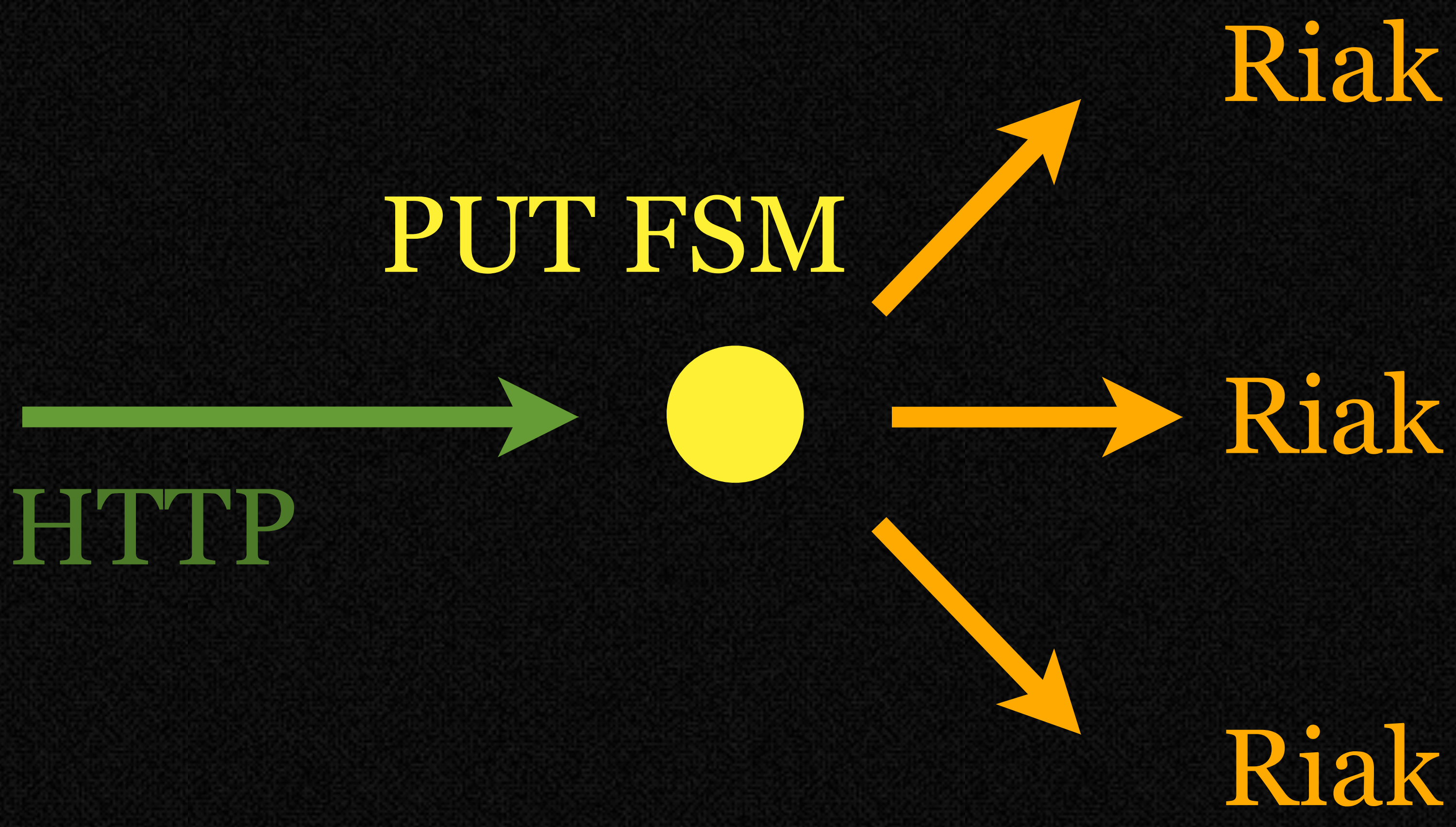


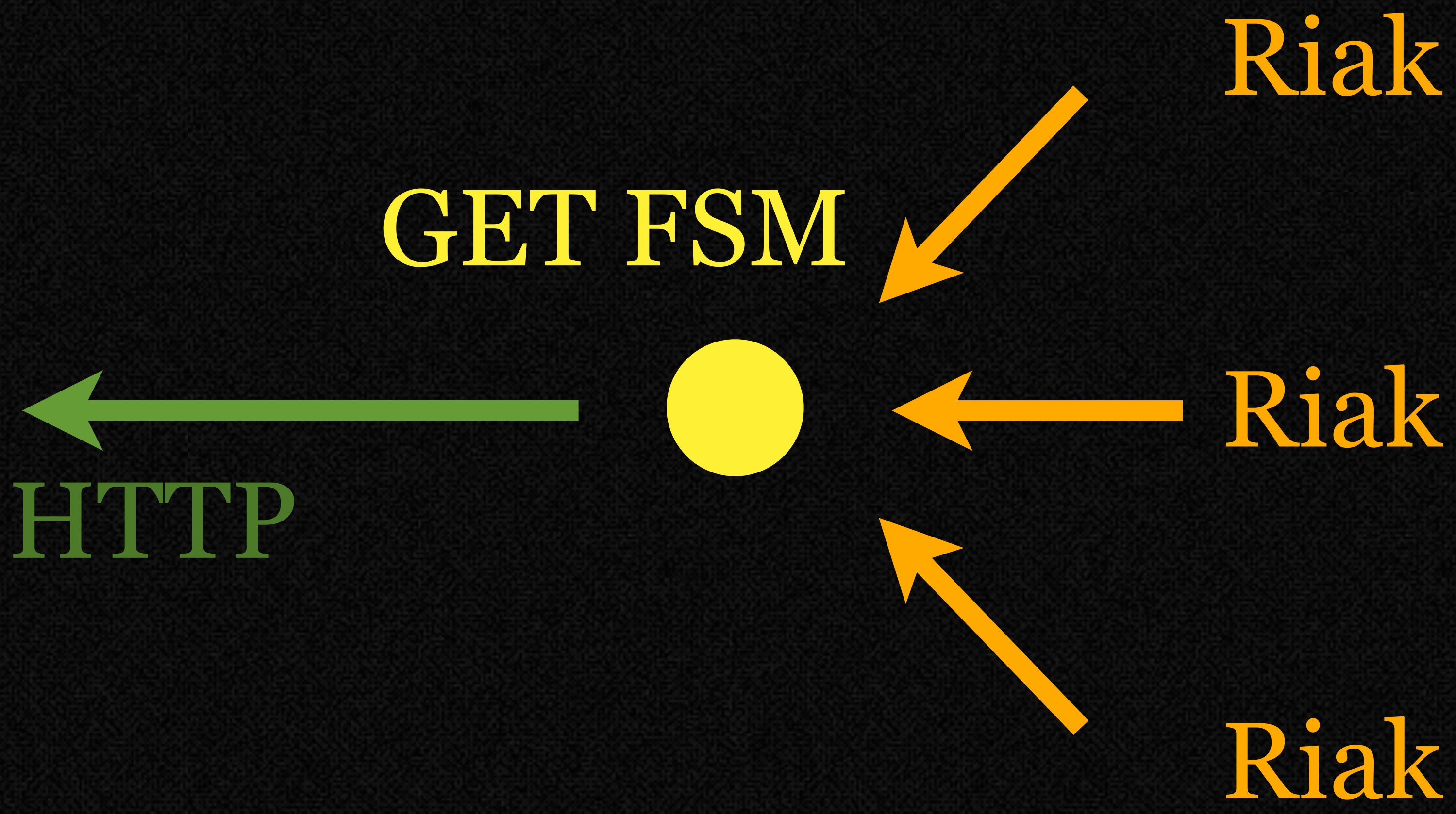
correct or not found

# cache forever

99% of data is immutable

# backpressure


Riak
PUT FSM
Riak
HTTP
Riak


Riak
GET FSM
Riak
HTTP
Riak

# webmachine

webmachine is great if you have a RESTful API s3 is not

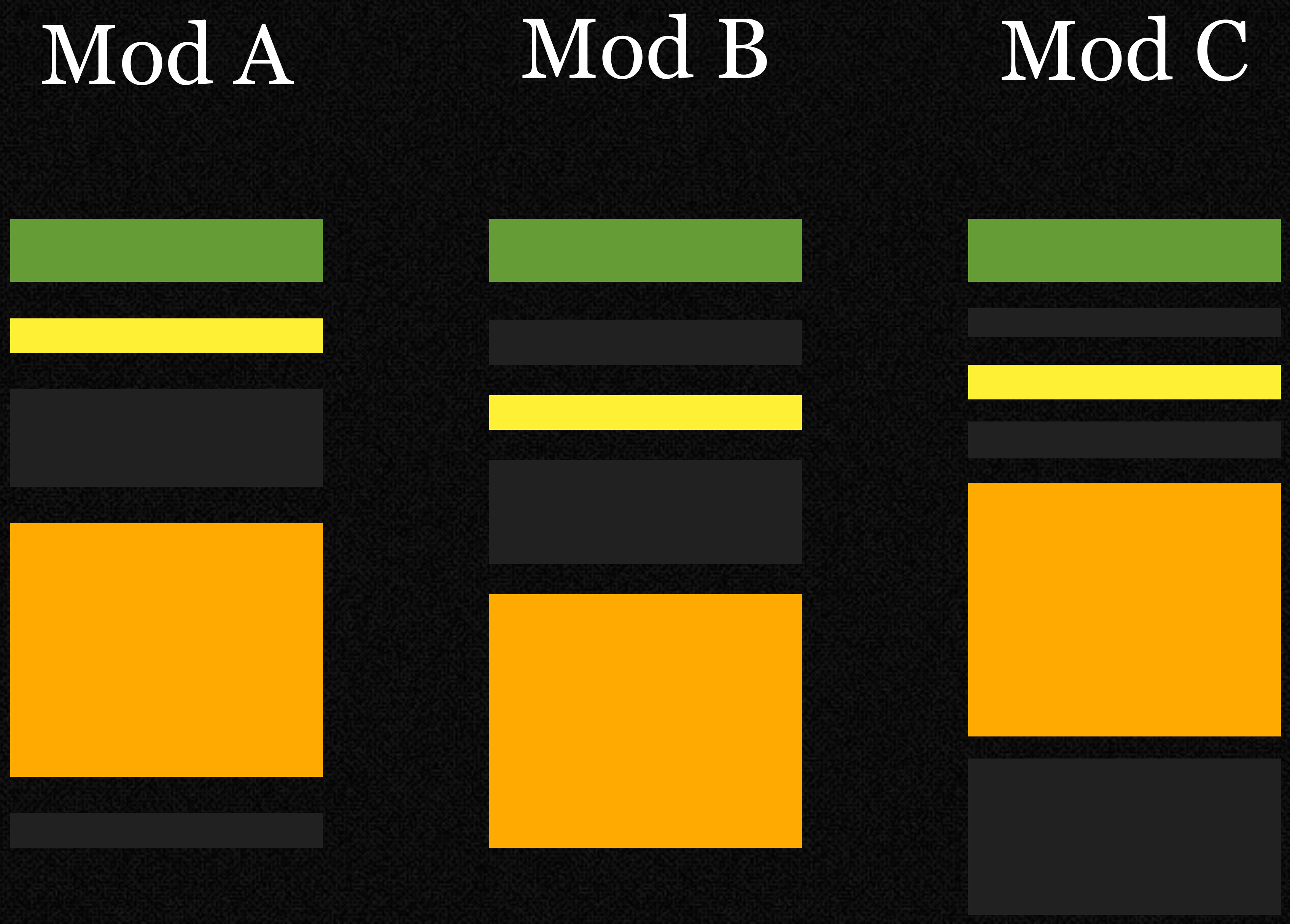
originally this was painful, now we rewrite requests

```
/foo
/buckets/foo
```

```
/foo?policy
/buckets/foo/policy
```

```
/foo/bar
/buckets/foo/objects/bar
```

```
/foo/bar?acl
/buckets/foo/objects/bar/acl
```



Mod A

Mod B

Mod C

# custom default resource funs

```
dict:merge(F, Resource,
        Defaults).
```

# module_info/1

# testing conflict resolution

(with EQC)

# binary `resolve` function

# associative and commutative

```
1 + (2 + (3 + (4 + (5 + 6))))
              ==
6 + (5 + (4 + (3 + (2 + 1))))
```

```
lists:foldl(Fun, L) ==
lists:foldr(Fun, L).
```

# HTTP 503

(or how I learned to stop worrying and say ‘no’)

# poolboy pools Riak connections

503 if you can’t check out a connection during `service_available`

separate pools for resources with different ‘costs’

# record versioning

upgrade/1

```
upgrade(A=#v1{}) ->
  upgrade(upgrade_v1(A);
upgrade(A=#v2{}) ->
  upgrade(upgrade_v2(A);
upgrade(A=#v3{}) ->
  A.
```

# read path:

upgrade

resolve

Kelly McLaughlin Scott Lystig Fritchie
Jared Morrow Andy Gross Bryan Fink
UENISHI Kota Shunichi Shinohara
Christopher Meiklejohn Kazuhiro Suzuki
Jordan West

Questions?