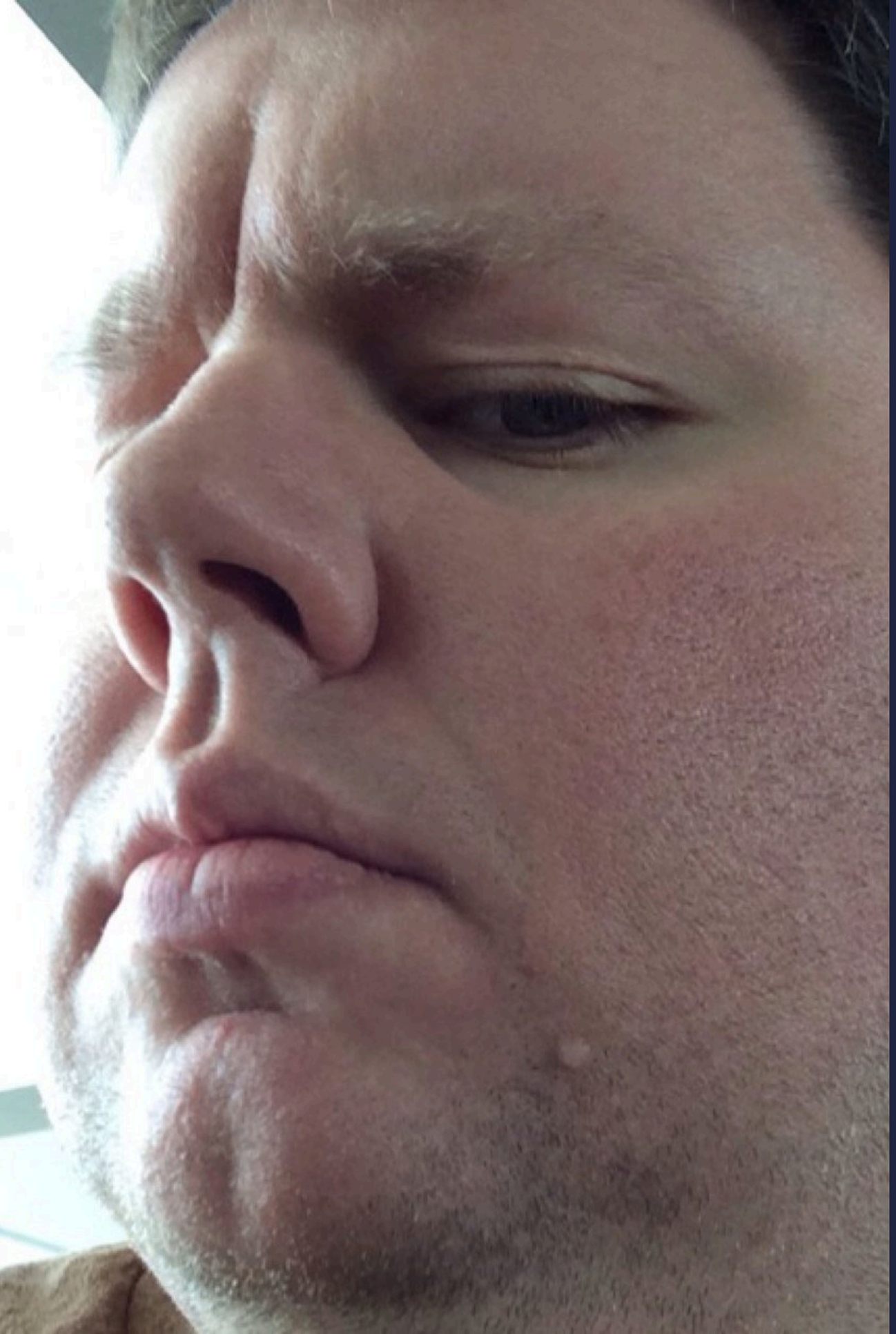
CLOUD FOUNDRY DIEGO

MODULAR AND EXTENSIBLE SUBSTRUCTURE FOR MICROSERVICES

CLOUD FOUNDRY™



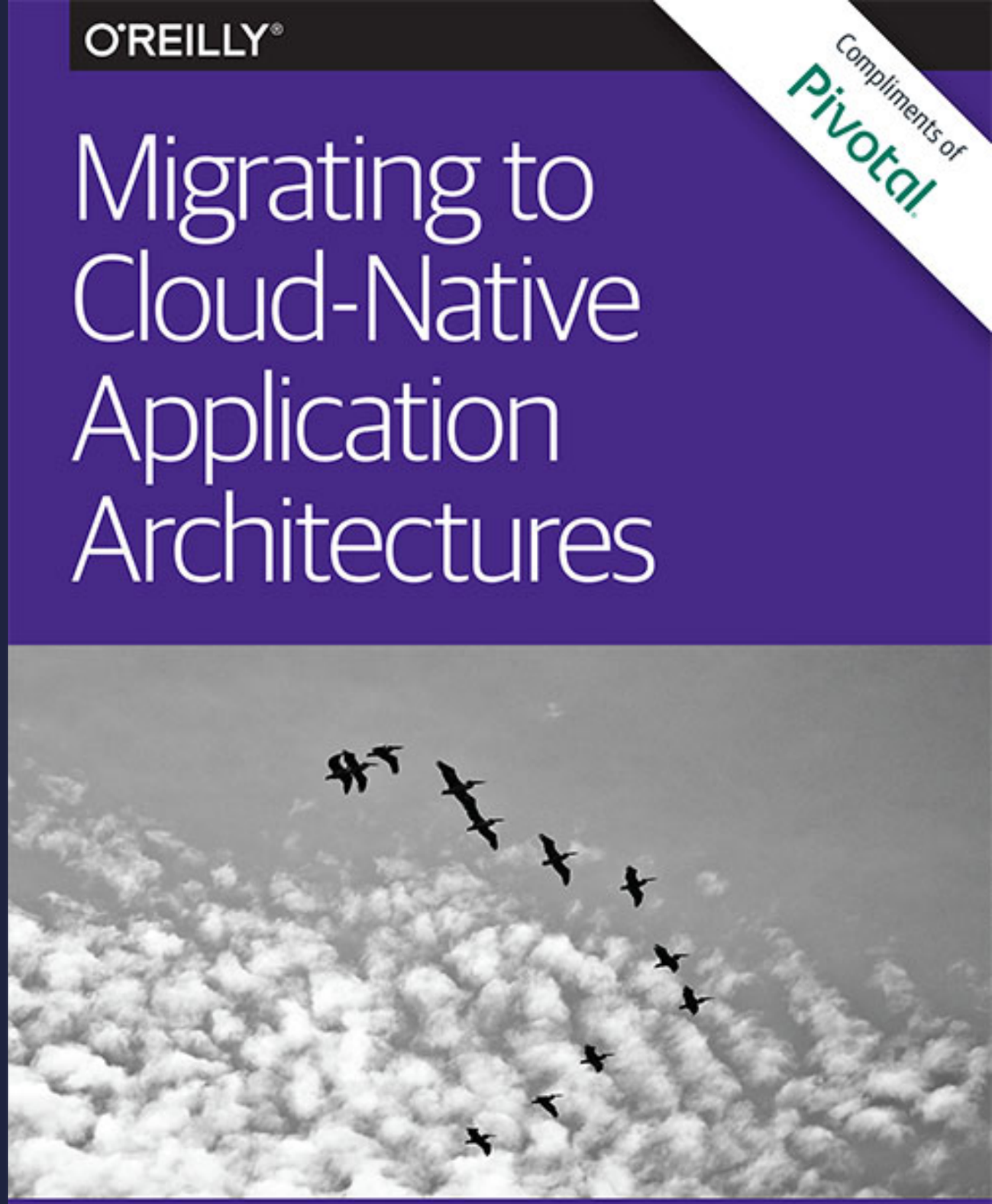
ME

Matt Stine @mstine
Principal Software Engineer
Pivotal
mstine@pivotal.io

I WROTE A LITTLE CLOUD BOOK....

FREE - Compliments of Pivotal

http://bit.ly/cloud-native-book




MICROSERVICES?
.com/shop/sharpwriter

Loosely coupled service oriented architecture with bounded contexts...

– Adrian Cockcroft

YOU MUST BE THIS TALL TO USE MICROSERVICES

- RAPID PROVISIONING
- BASIC MONITORING
- RAPID APPLICATION DEPLOYMENT
- DEVOPS CULTURE

http://martinfowler.com/bliki/MicroservicePrerequisites.html

A SYMBIOTIC RELATIONSHIP

PLATFORM FEATURES

- Environment Provisioning
- On-Demand Scaling
- Failover/Resilience
- Routing/Load Balancing
- Data Service Operations (BOSH)
- Monitoring

But we need MOAR features...

HARD

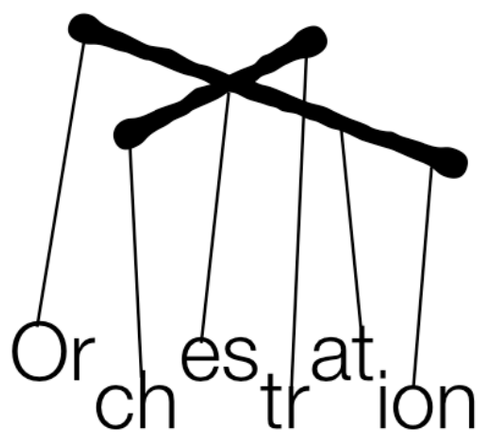
TO ADD NEW FEATURES

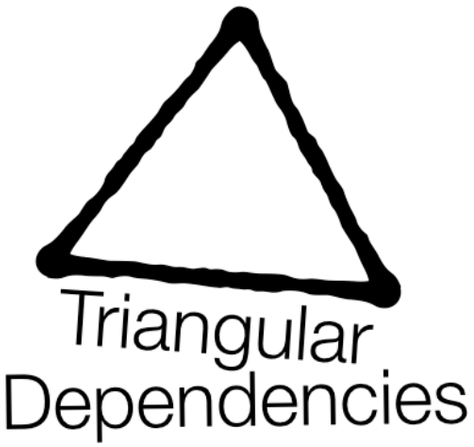
HARD

TO MAINTAIN EXISTING FEATURES

WHY IS IT HARD?

Tight Coupling





Poor separation of concerns

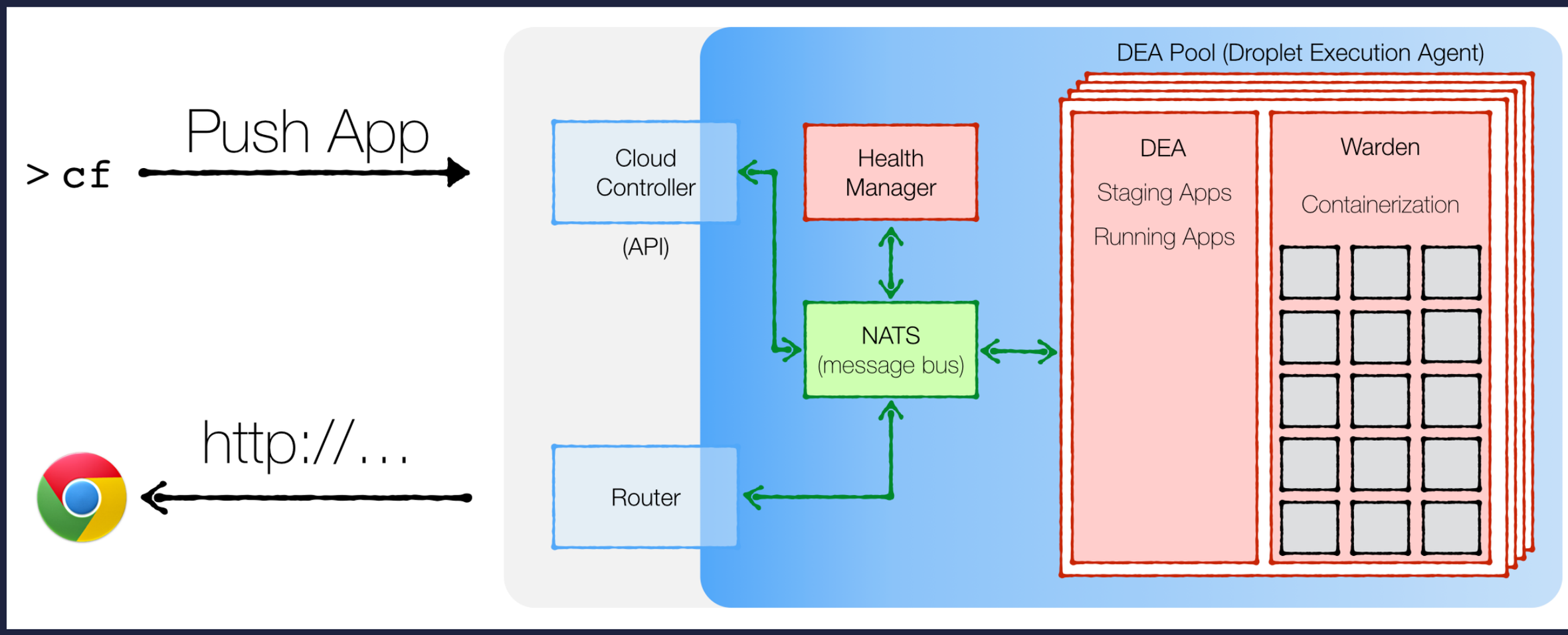
Domain Specific
(app, app, app, app)

Platform  Specific

From https://www.youtube.com/watch?v=10kmVTFhfLY

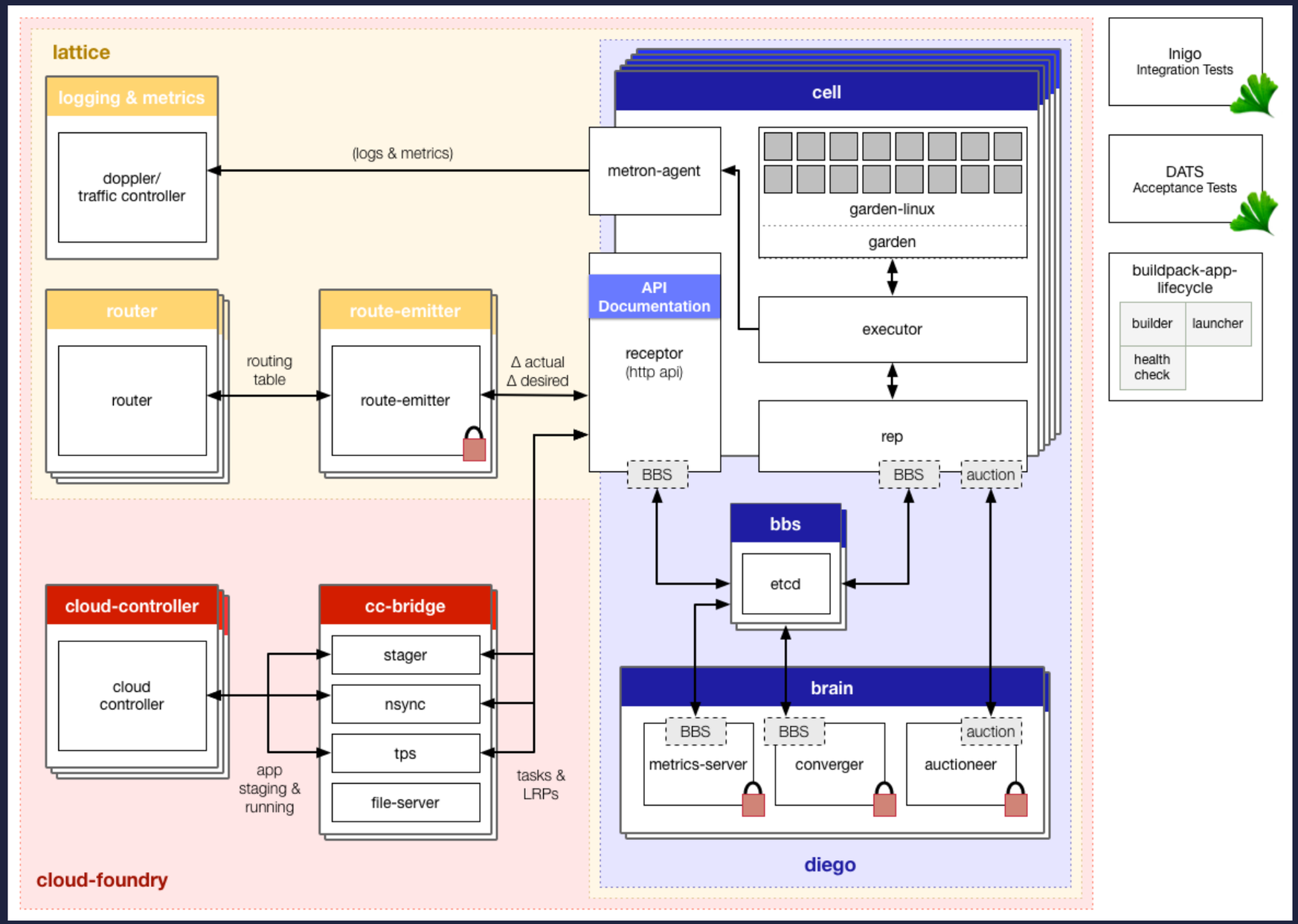
ENTER THE REWRITE

WHAT?

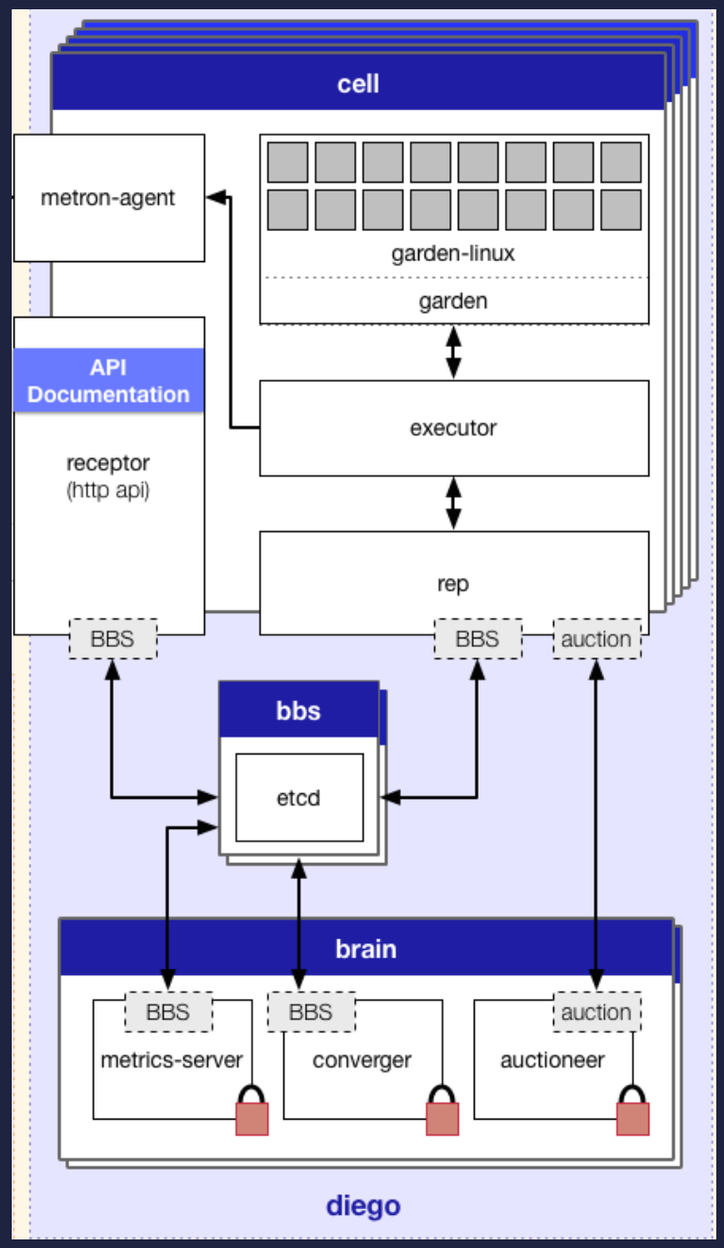


From https://www.youtube.com/watch?v=10kmVTFhfLY

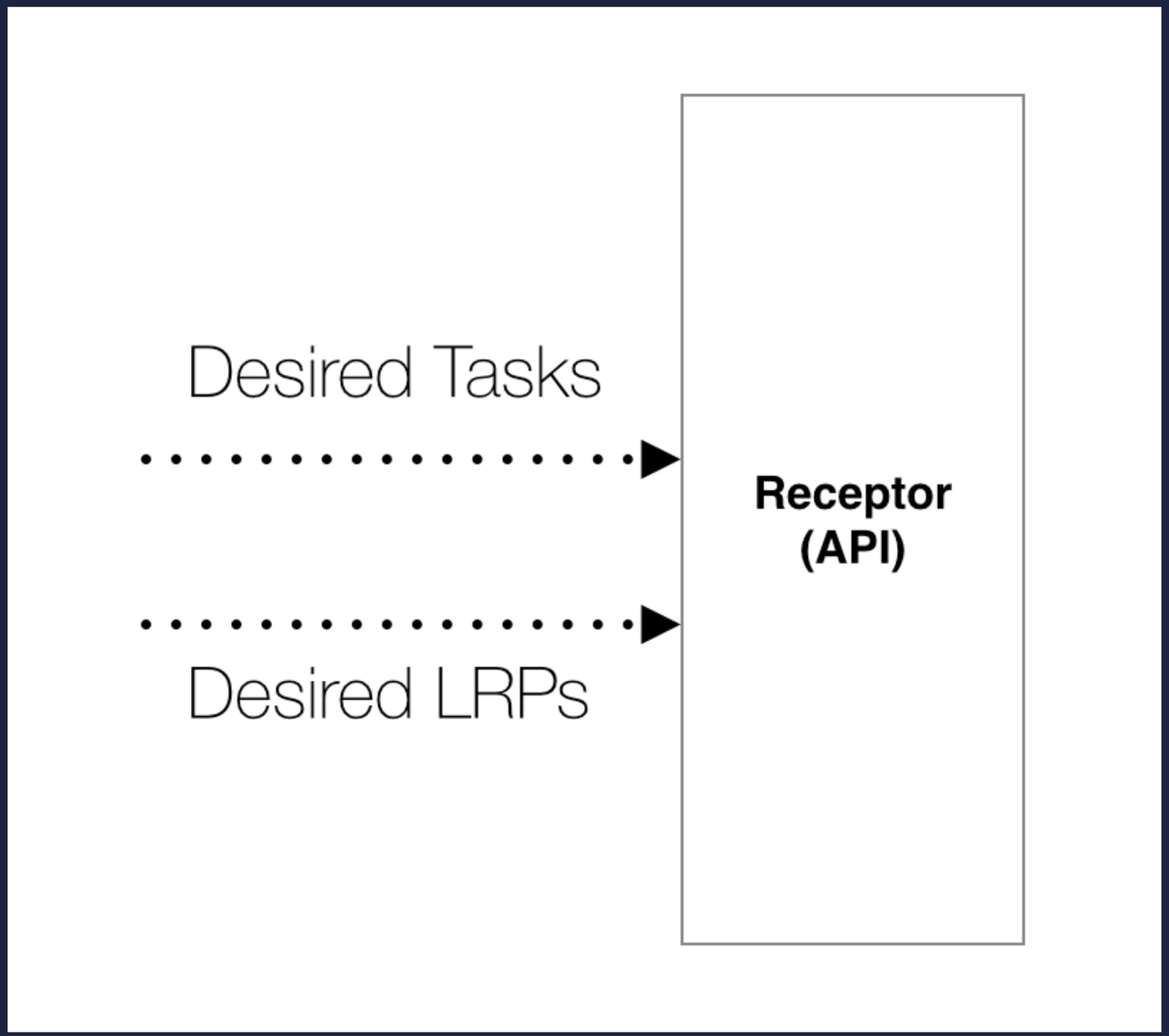
THE RESULT:


lattice
logging & metrics
doppler/
traffic controller
(logs & metrics)
metron-agent
cell
garden-linux
garden
router
routing
table
route-emitter
Δ actual
Δ desired
API
Documentation
receptor
(http api)
executor
rep
BBS
BBS
auction
bbs
etcd
brain
BBS
metrics-server
BBS
converger
auction
auctioneer
cloud-controller
cloud
controller
app
staging &
running
cc-bridge
stager
nsync
tps
file-server
tasks &
LRPs
cloud-foundry
diego
Inigo
Integration Tests
DATS
Acceptance Tests
buildpack-app-
lifecycle
builder
launcher
health
check

THIS IS DIEGO



MEET THE RECEPTOR



TASKS
ONE-OFF WORK

LRPS
LONG-RUNNING PROCESSES

DESIRED TASK

```
{
    ...

    "rootfs": "docker:///docker-org/docker-image",
    "env": [
        {"name": "ENV_NAME_A", "value": "ENV_VALUE_A"},
        {"name": "ENV_NAME_B", "value": "ENV_VALUE_B"}
    ],

    "cpu_weight": 57,
    "disk_mb": 1024,
    "memory_mb": 128,
    "privileged": true,

    "action":  ACTION(s) TO RUN,

    ...
}
```

ACTIONS

- **RunAction:** run proc in container
- **DownloadAction:** fetches and extract archive
- **UploadAction:** POST file from container to URL
- **ParallelAction:** run multiple actions in parallel
- **SerialAction:** runs multiple actions in order

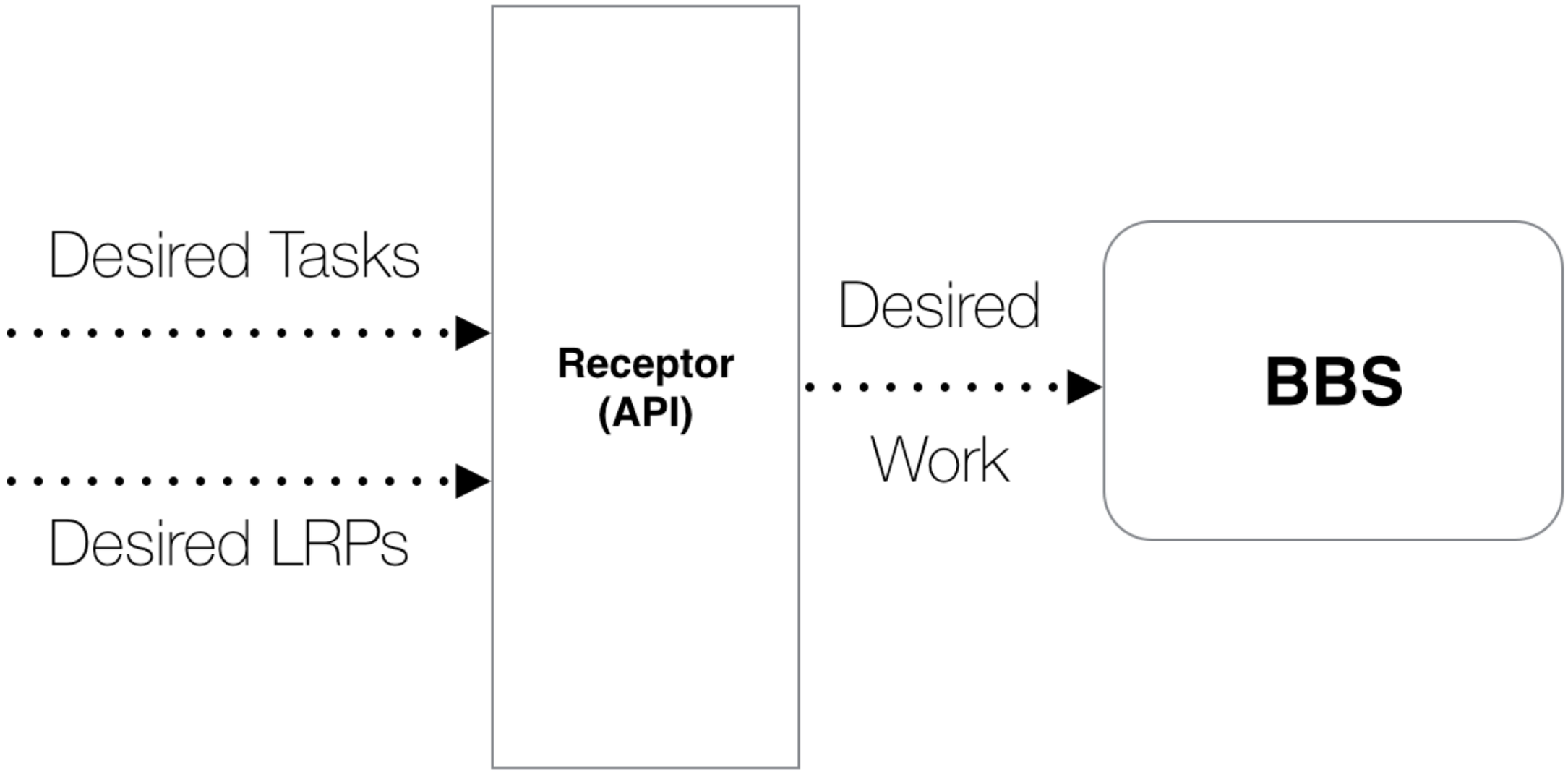
MOAR ACTIONS

- EmitProgressAction: wraps action and logs progress
  - TimeoutAction: wrap action and fail if timed out
- TryAction: wrap action and ignore errors

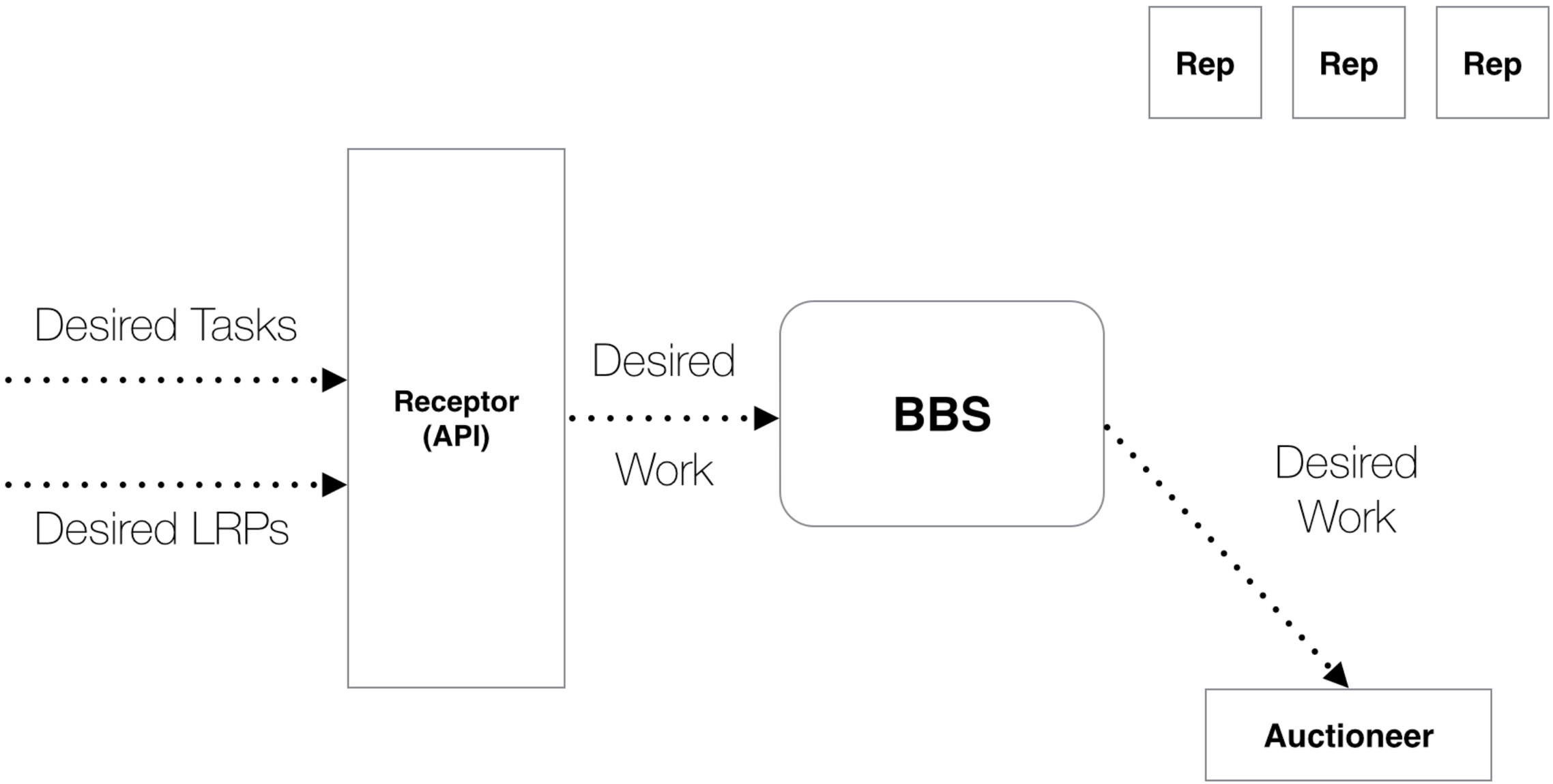
DESIRED LRP

```
{
    ...

    "instances": 17,

    "rootfs": "VALID-ROOTFS",

    "env": [
        {"name": "ENV_NAME_A", "value": "ENV_VALUE_A"},
        {"name": "ENV_NAME_B", "value": "ENV_VALUE_B"}
    ],

    "cpu_weight": 57,
    "disk_mb": 1024,
    "memory_mb": 128,
    "privileged": true,

    "setup": ACTION,
    "action":  ACTION,
    "monitor": ACTION,
    "start_timeout": N seconds,

    "ports": [8080, 5050],

    ...
}
```

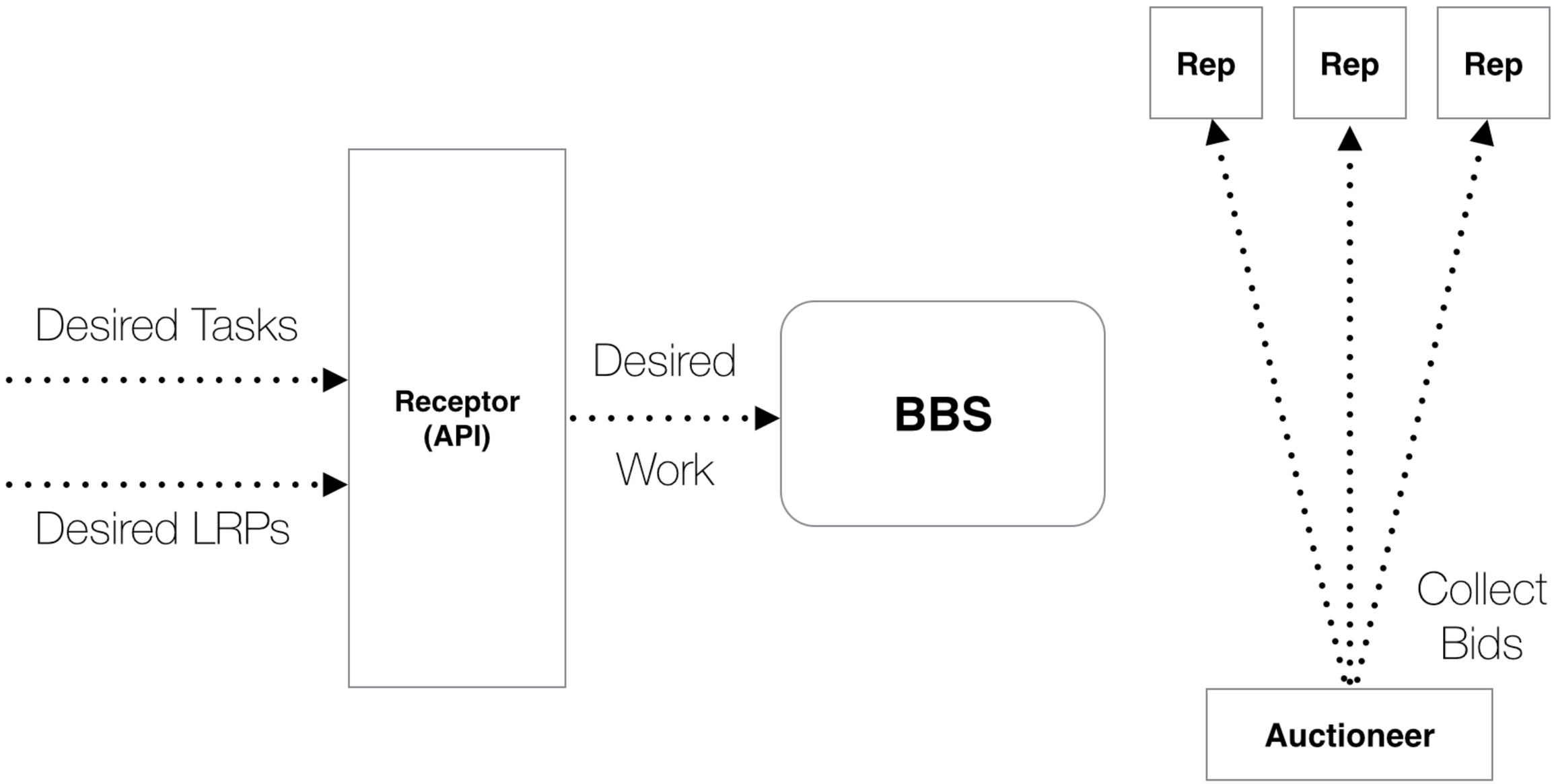
MEET THE BBS



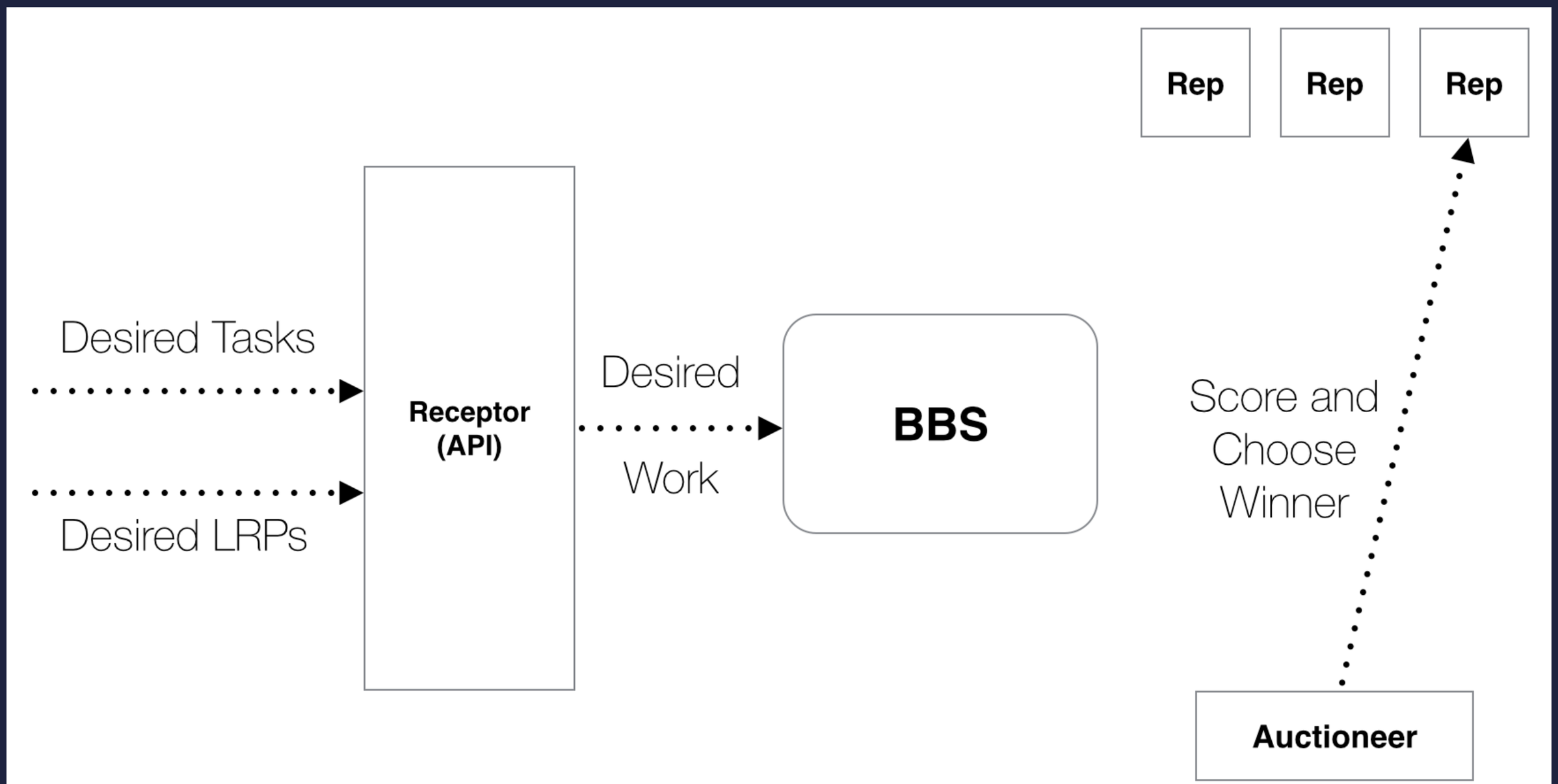
LET'S HAVE AN AUCTION



LET'S HAVE AN AUCTION



LET'S HAVE AN AUCTION



INSIDE THE CELL

Rep

Tasks
&
LRPs

Executor

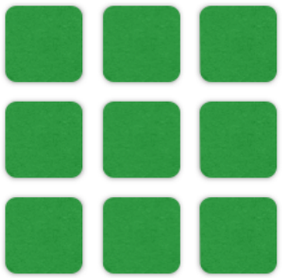
Actions

Garden

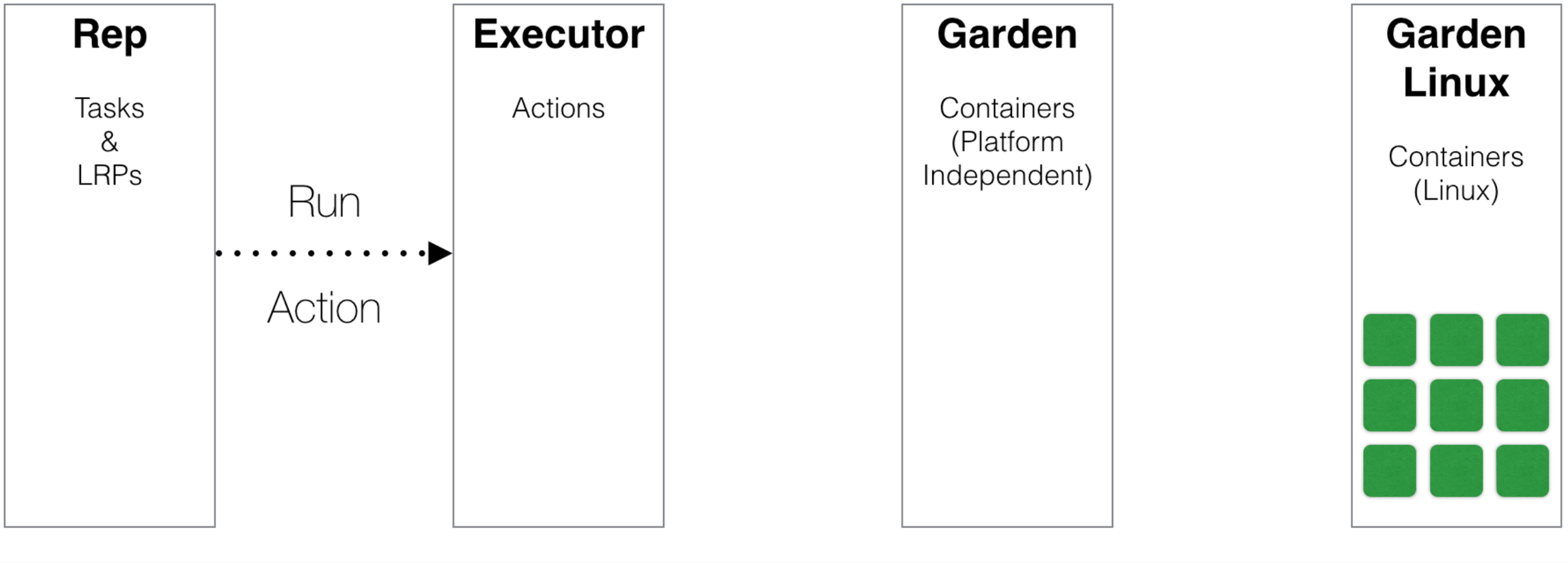
Containers
(Platform
Independent)

Garden Linux

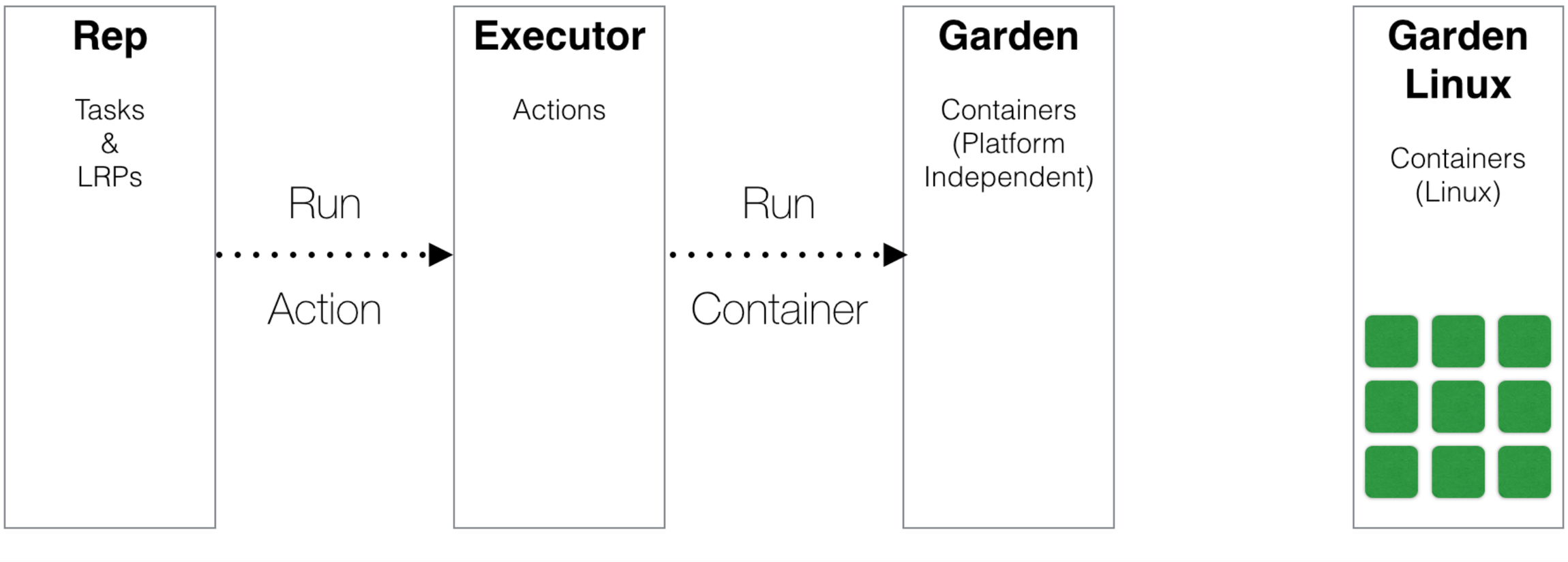
Containers
(Linux)



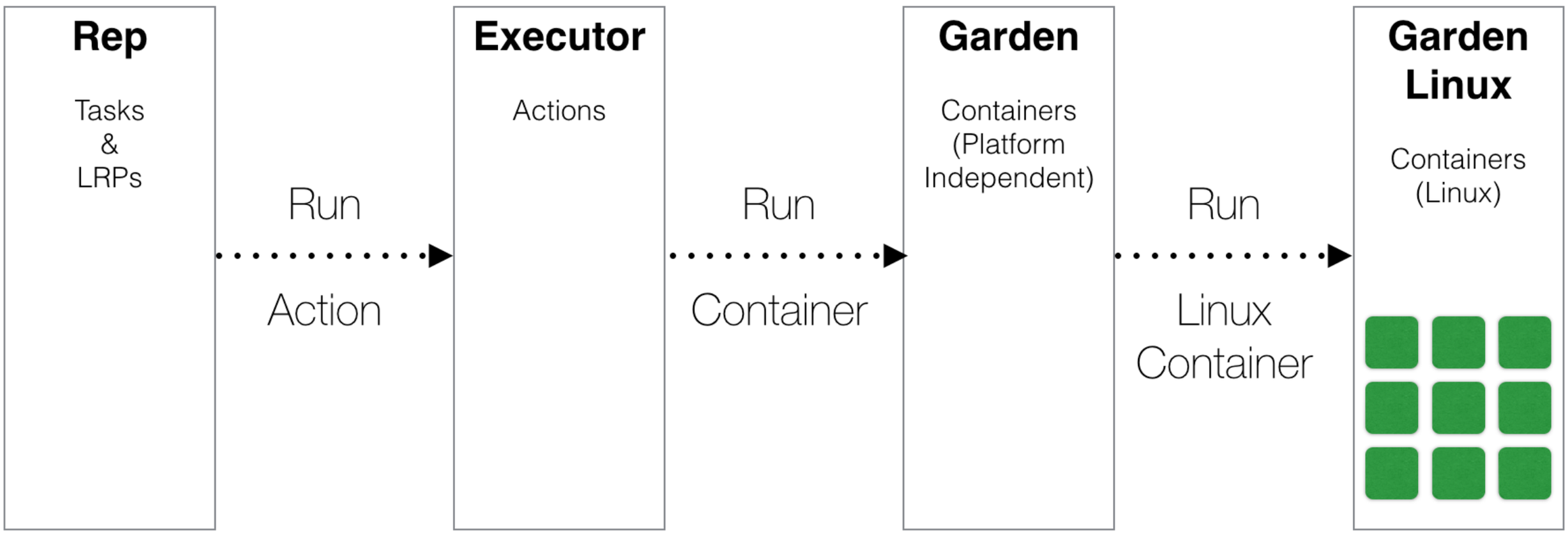
INSIDE THE CELL



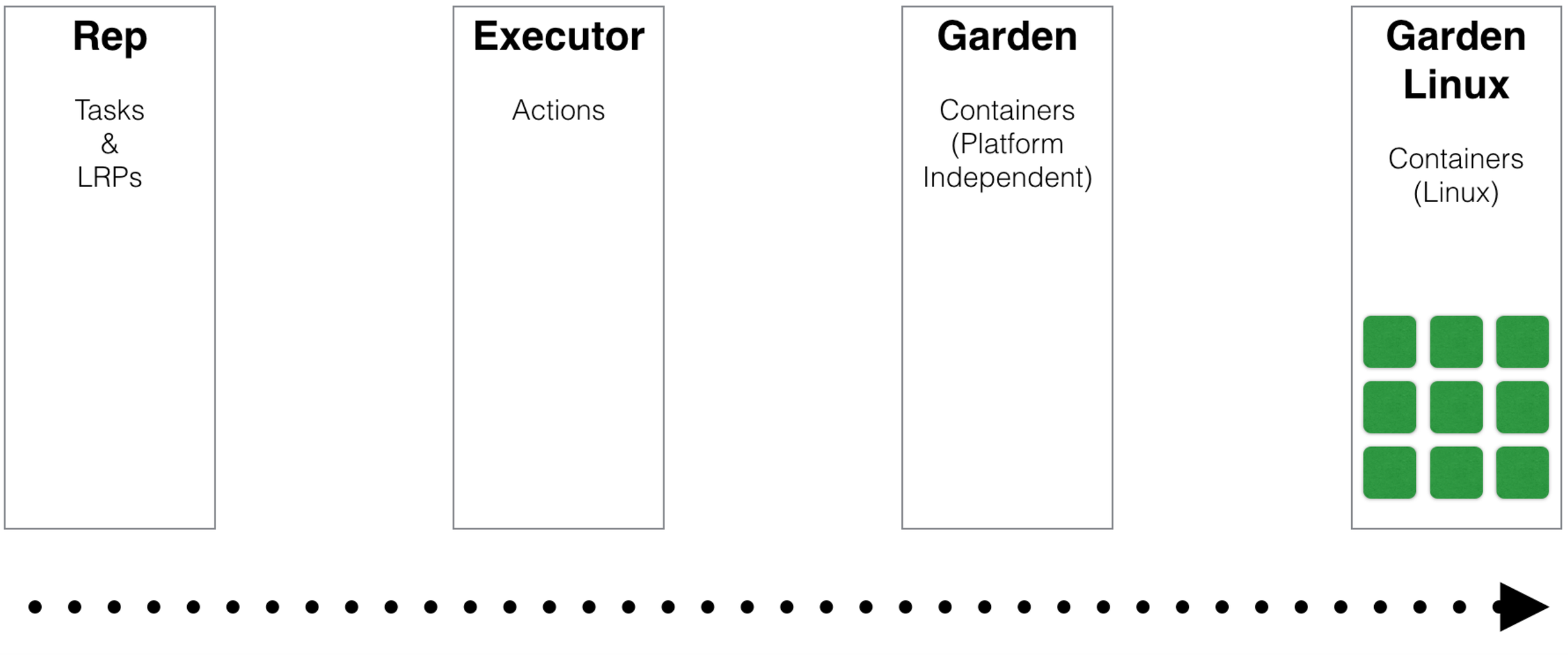
INSIDE THE CELL



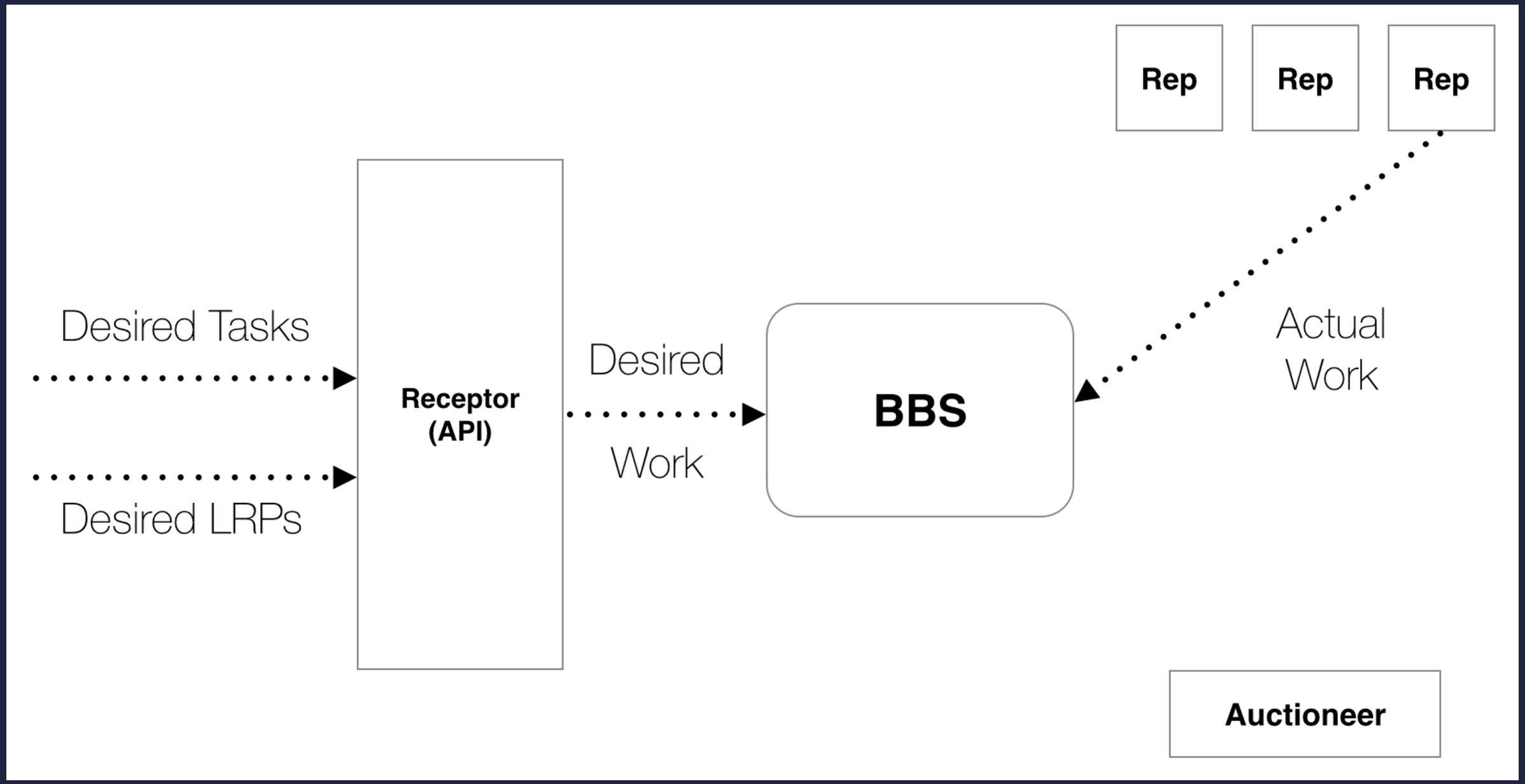
INSIDE THE CELL



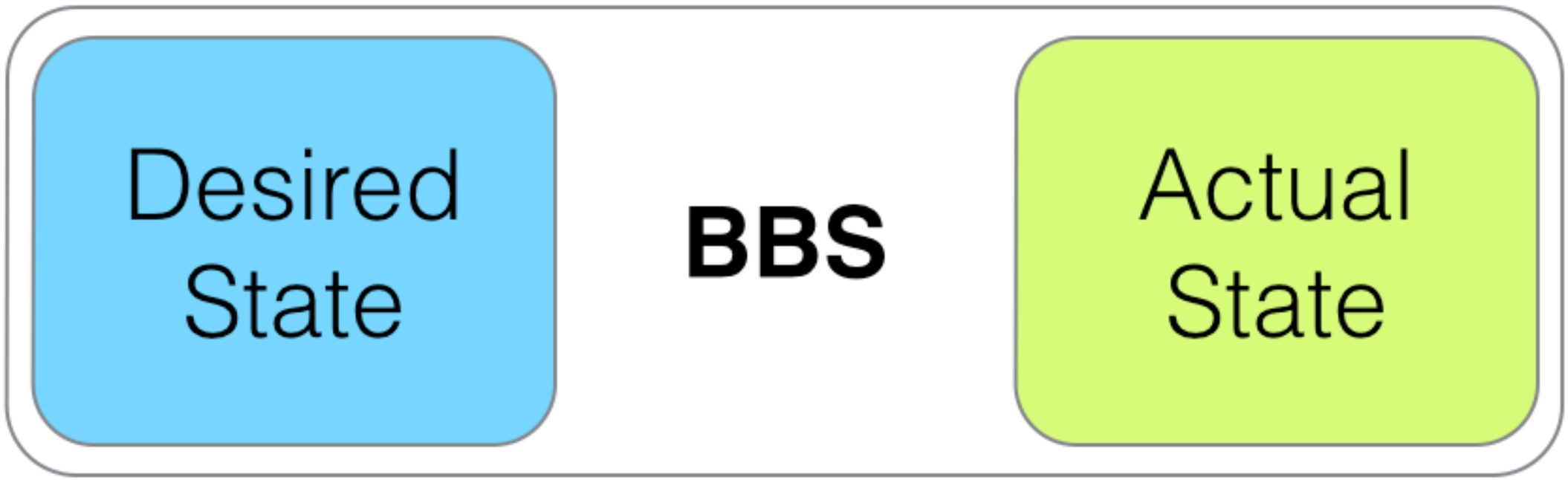
SPECIFICITY GRADIENT



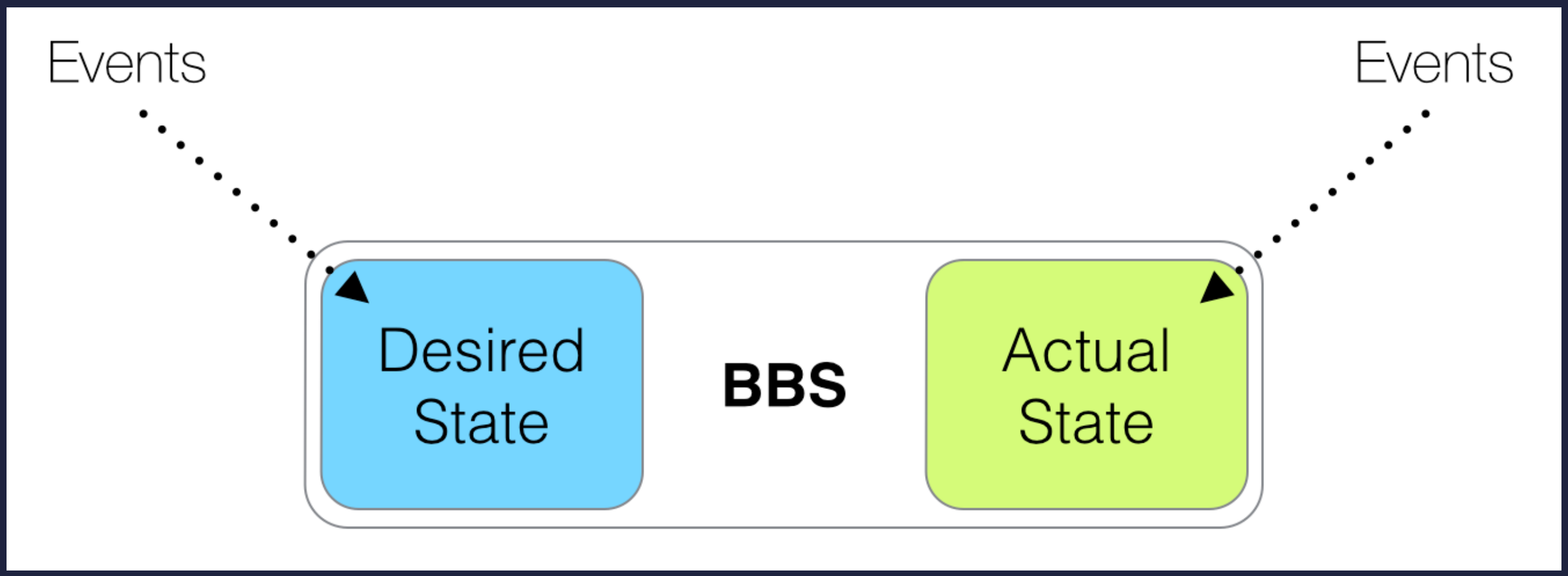
REP RESENTING THE ACTUAL WORK



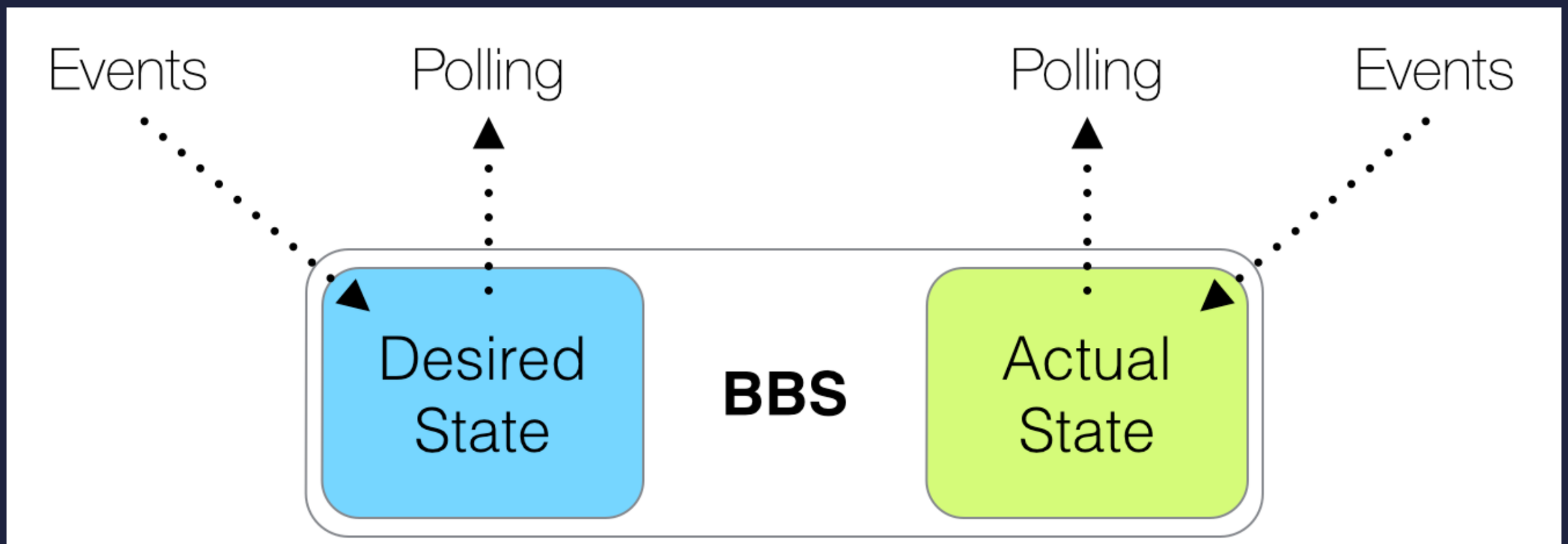
WHAT IS THE TRUTH?



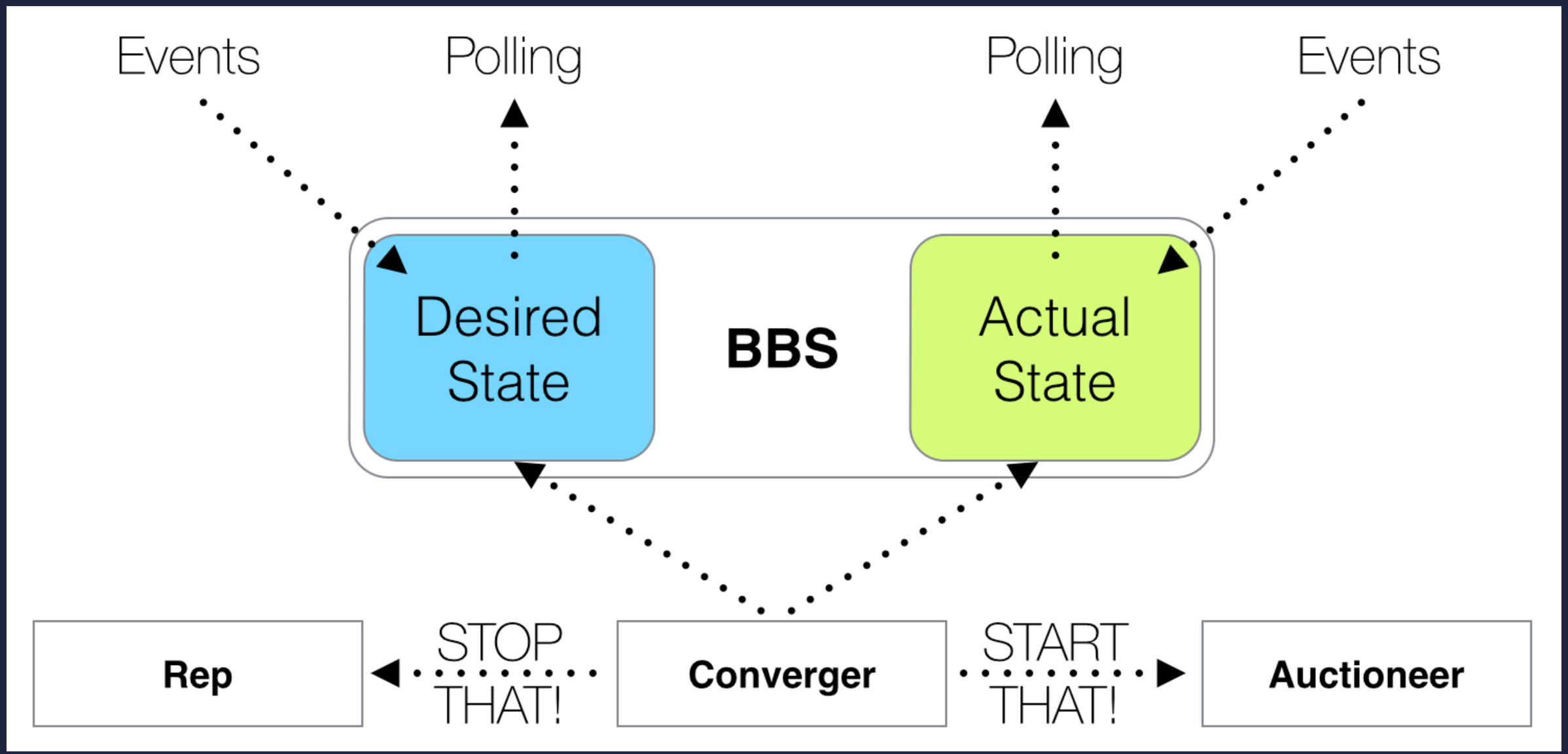
WHAT IS THE TRUTH?



WHAT IS THE TRUTH?

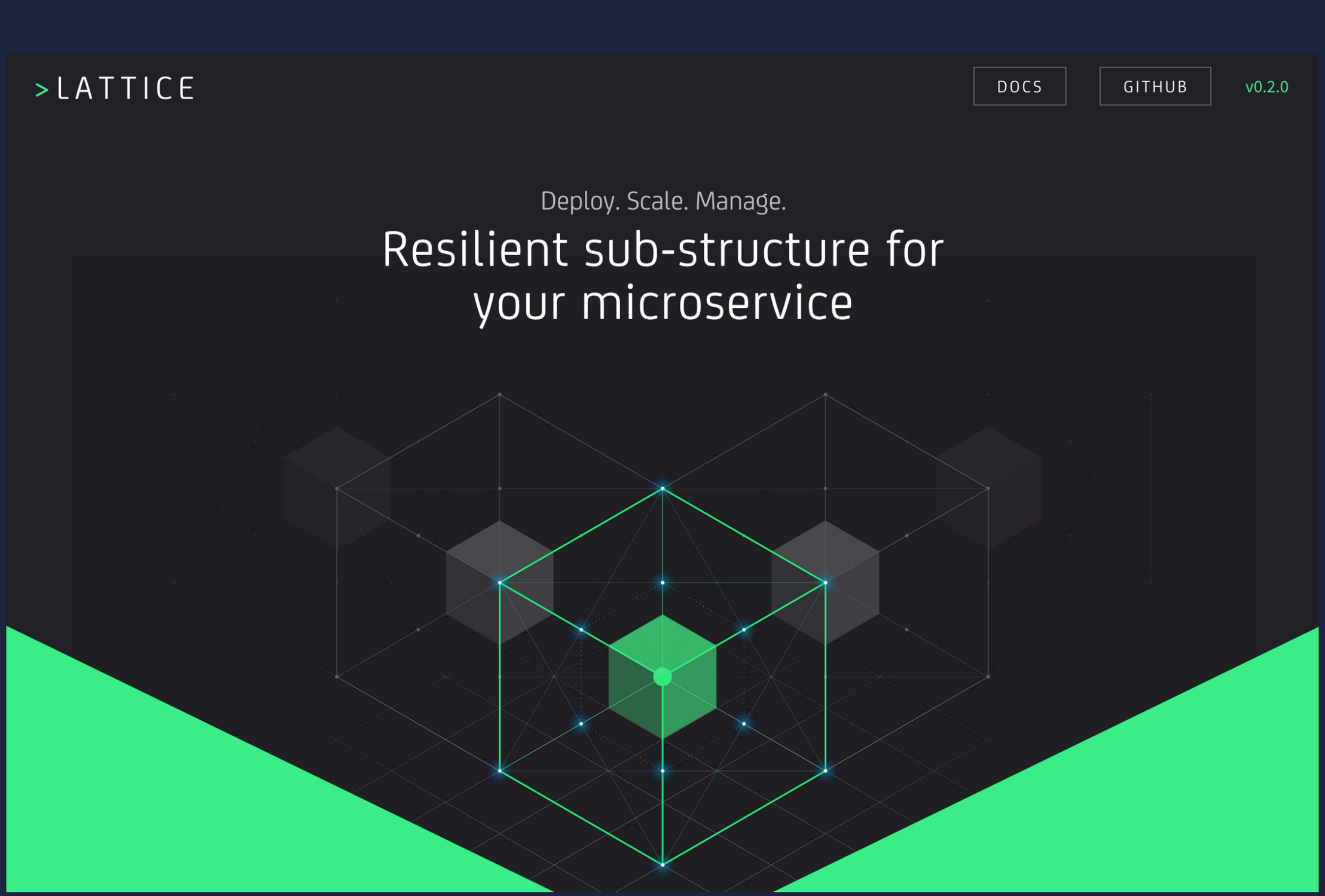


CONVERGENCE



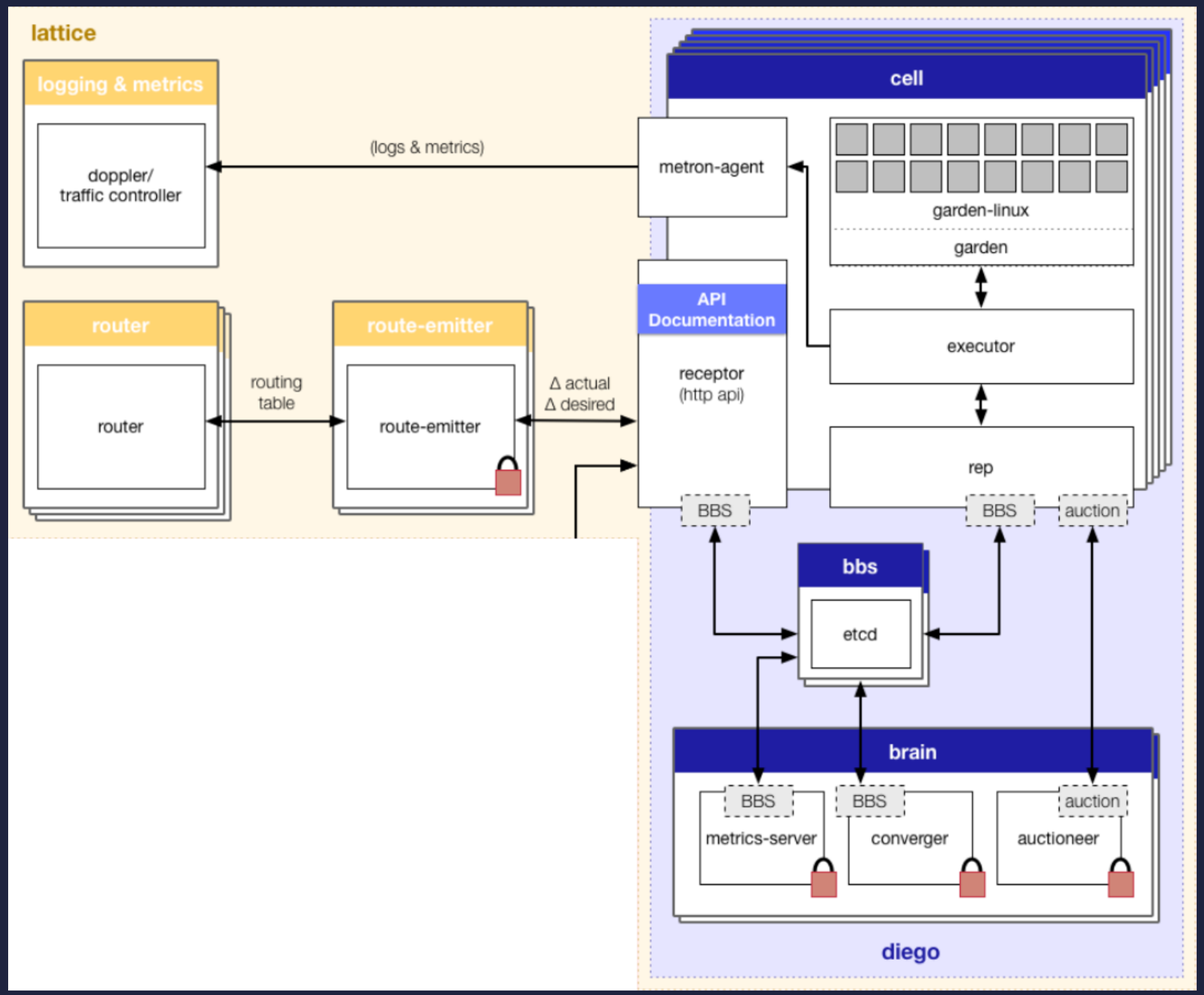
PLAY WITH DIEGO?


DEPLOYING CLOUD FOUNDRY IS HARD
WALLPAPERSWIDE.COM



https://lattice.cf

THIS IS LATTICE



LATTICE + DIEGO

- Adds CF GoRouter (dynamic HTTP routing)
- Adds Doppler (log/metrics aggregation)
- LOCAL: vagrant up
- CLOUD: terraform apply (AWS, Google, DigitalOcean)

DEMO

WHAT ELSE?


MICROSERVICES

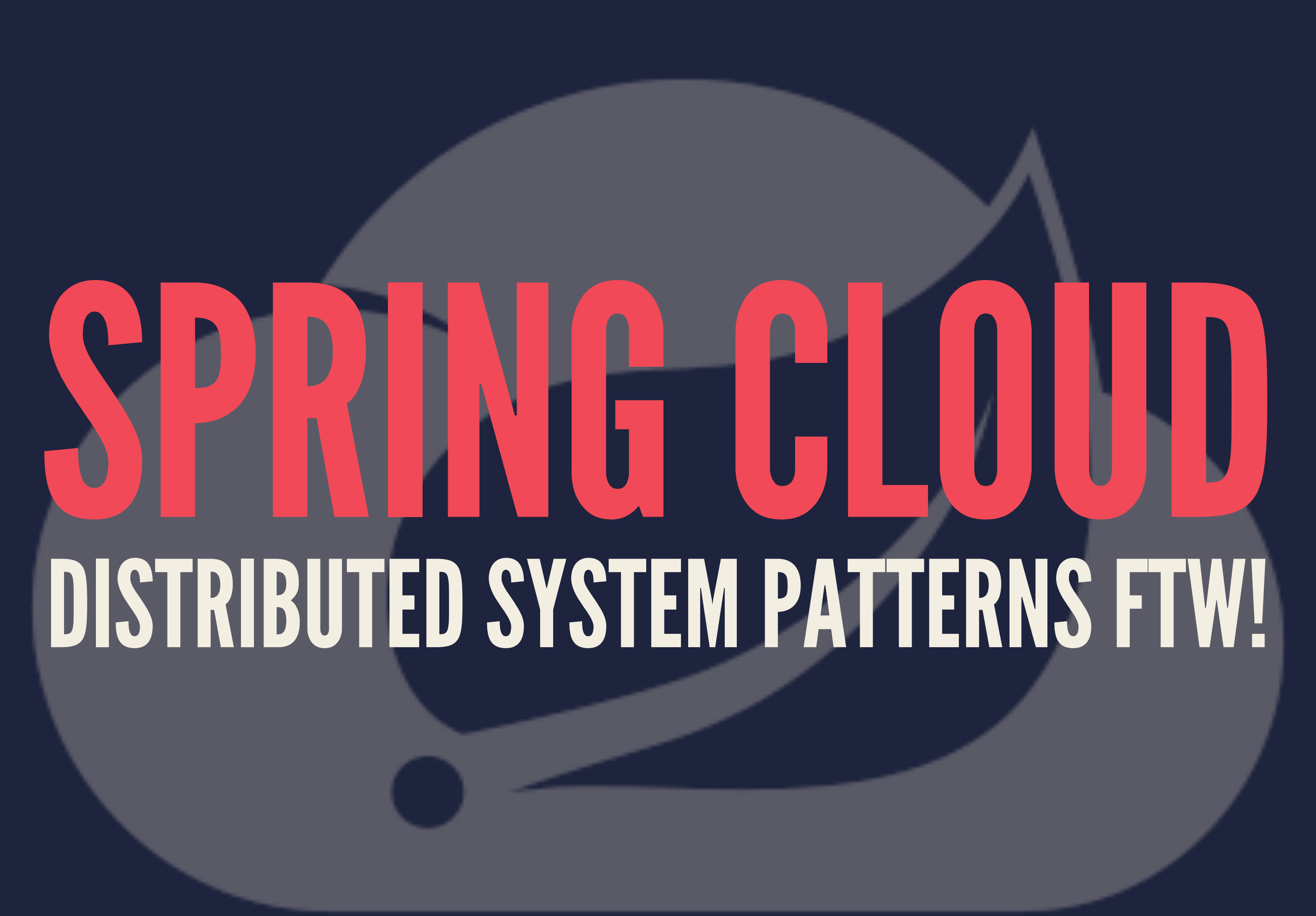

NO MICROSERVICE
IS AN ISLAND

CHALLENGES OF DISTRIBUTED SYSTEMS

- Configuration Management
- Service Registration & Discovery
- Routing & Load Balancing
- Fault Tolerance (Circuit Breakers!)
- Monitoring
- Concurrent API Aggregation & Transformation

NETFLIX

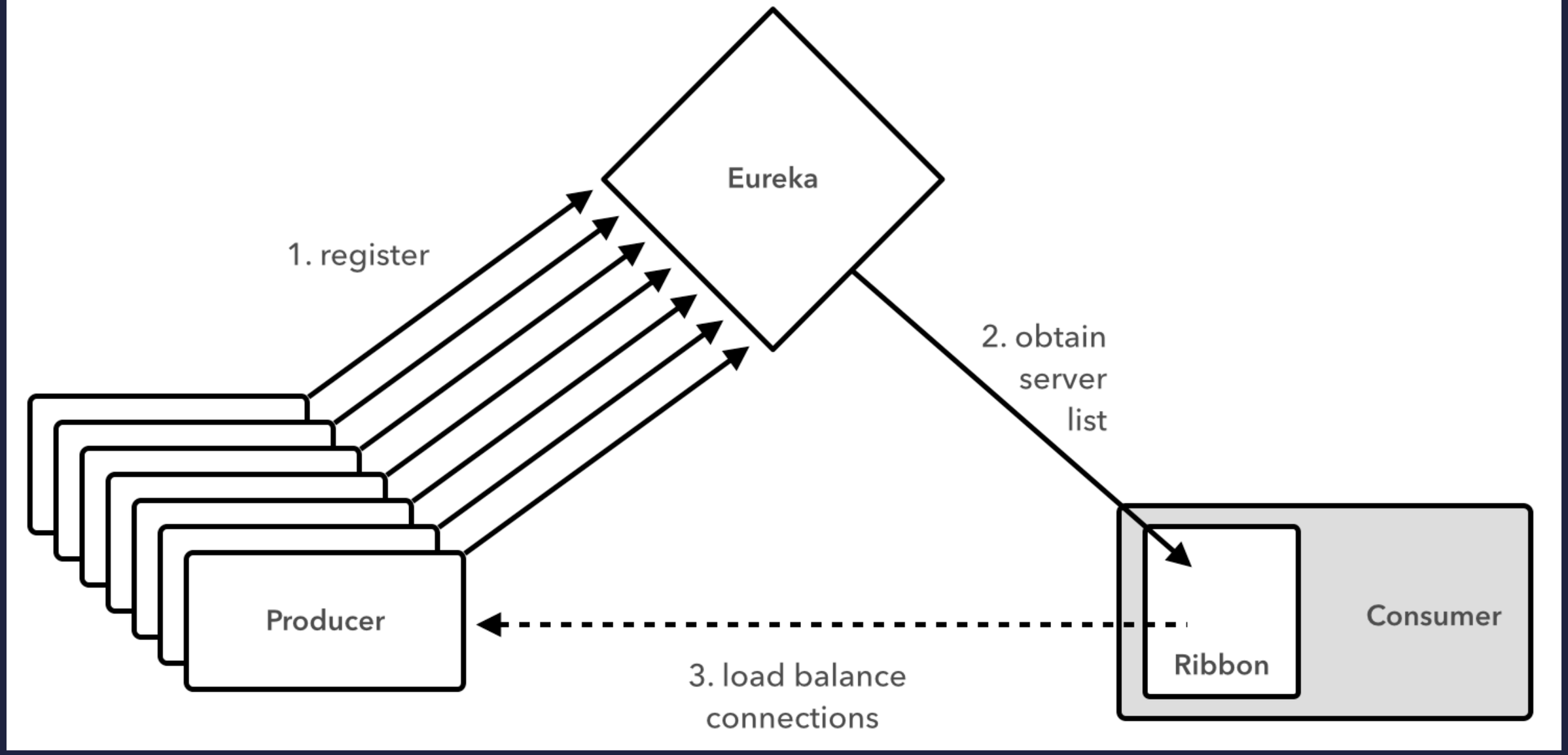
OSS


SPRING CLOUD
DISTRIBUTED SYSTEM PATTERNS FTW!

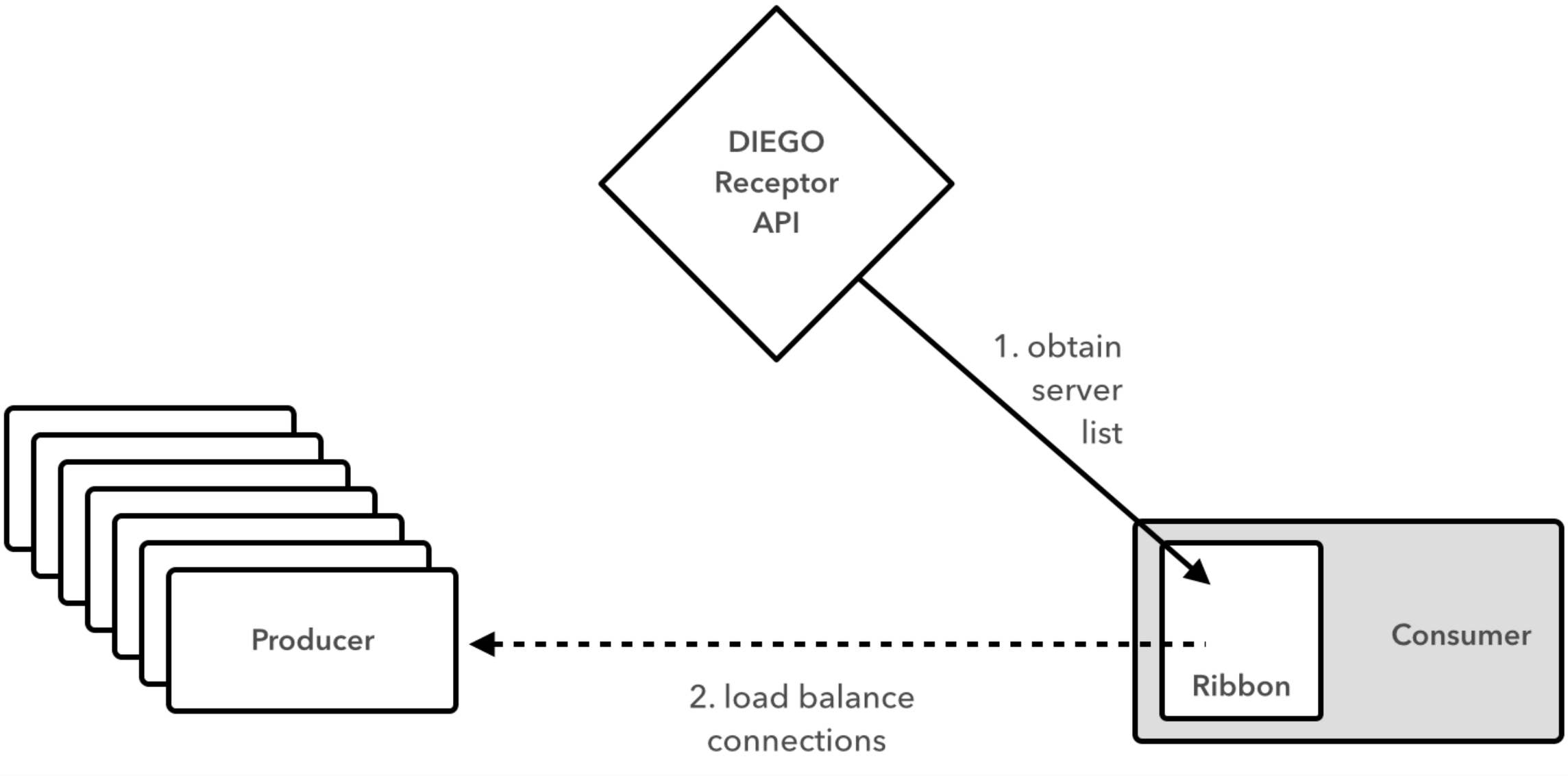
SPRING CLOUD LATTICE

- https://github.com/spring-cloud/spring-cloud-lattice
- https://spring.io/guides/gs/spring-cloud-and-lattice

EUREKA + RIBBON



LATTICE + RIBBON

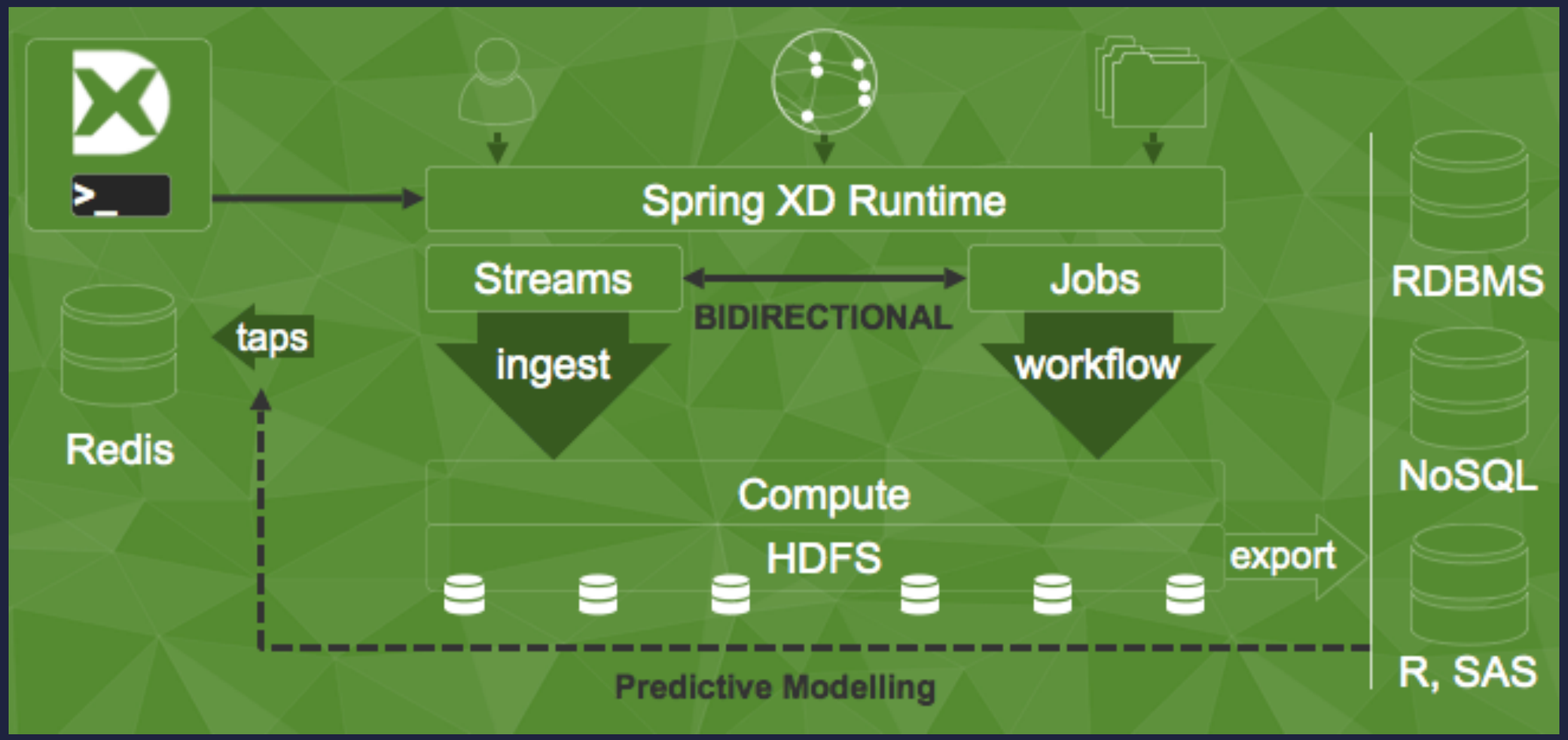


DEMO

DATA

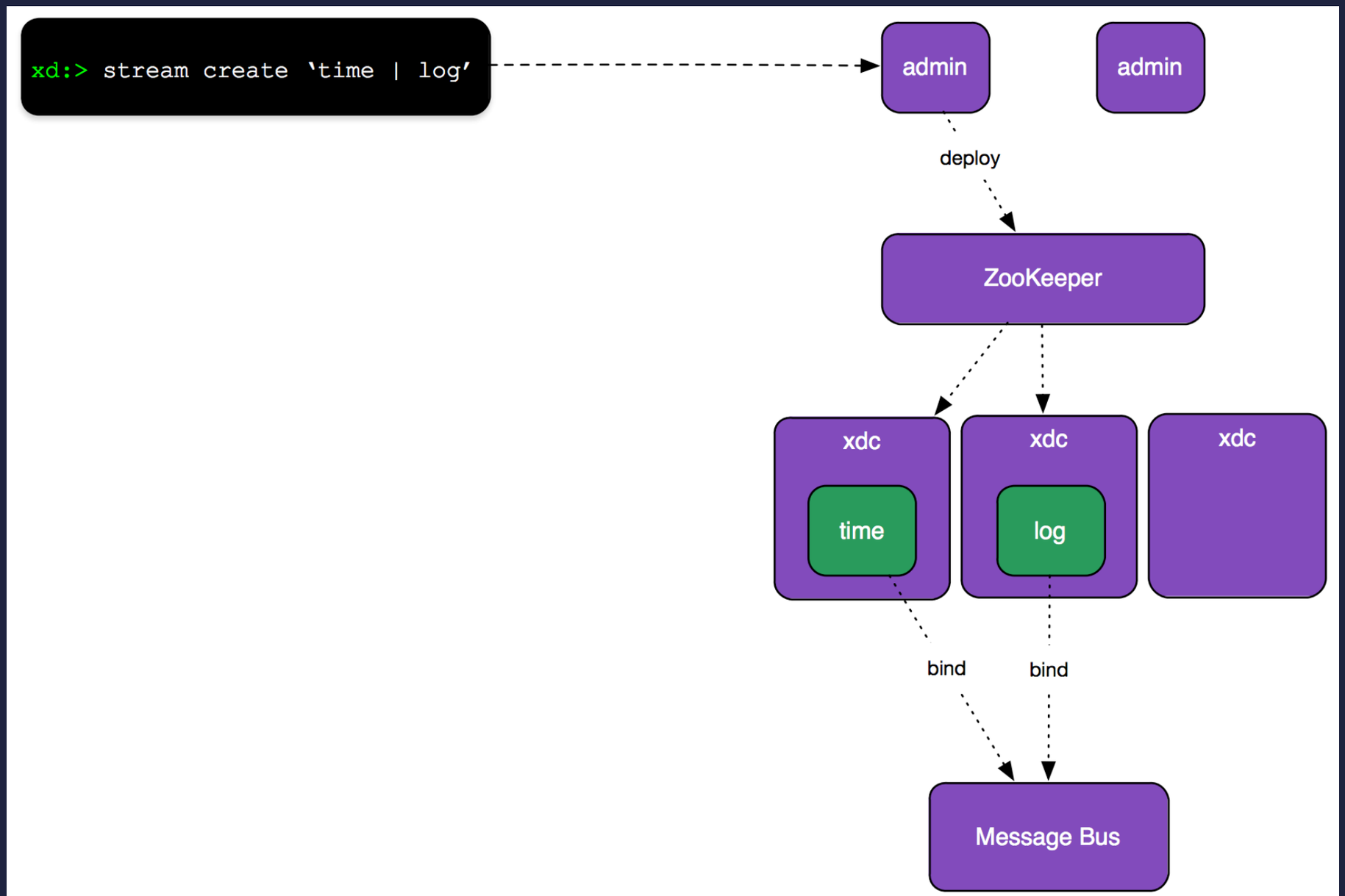
MICROSERVICES?

SPRING XD

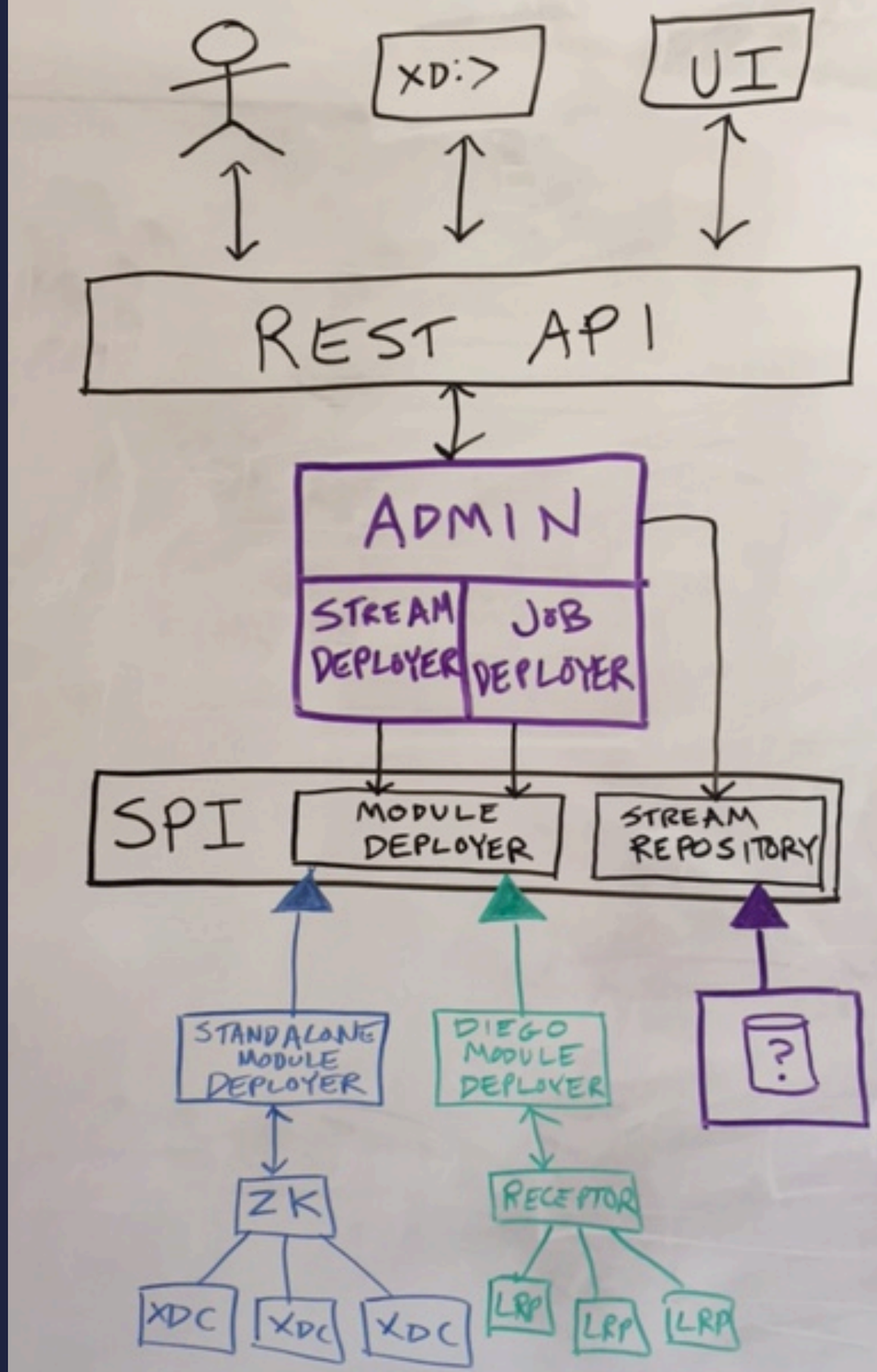


http://projects.spring.io/spring-xd

SPRING XD DISTRIBUTED RUNTIME

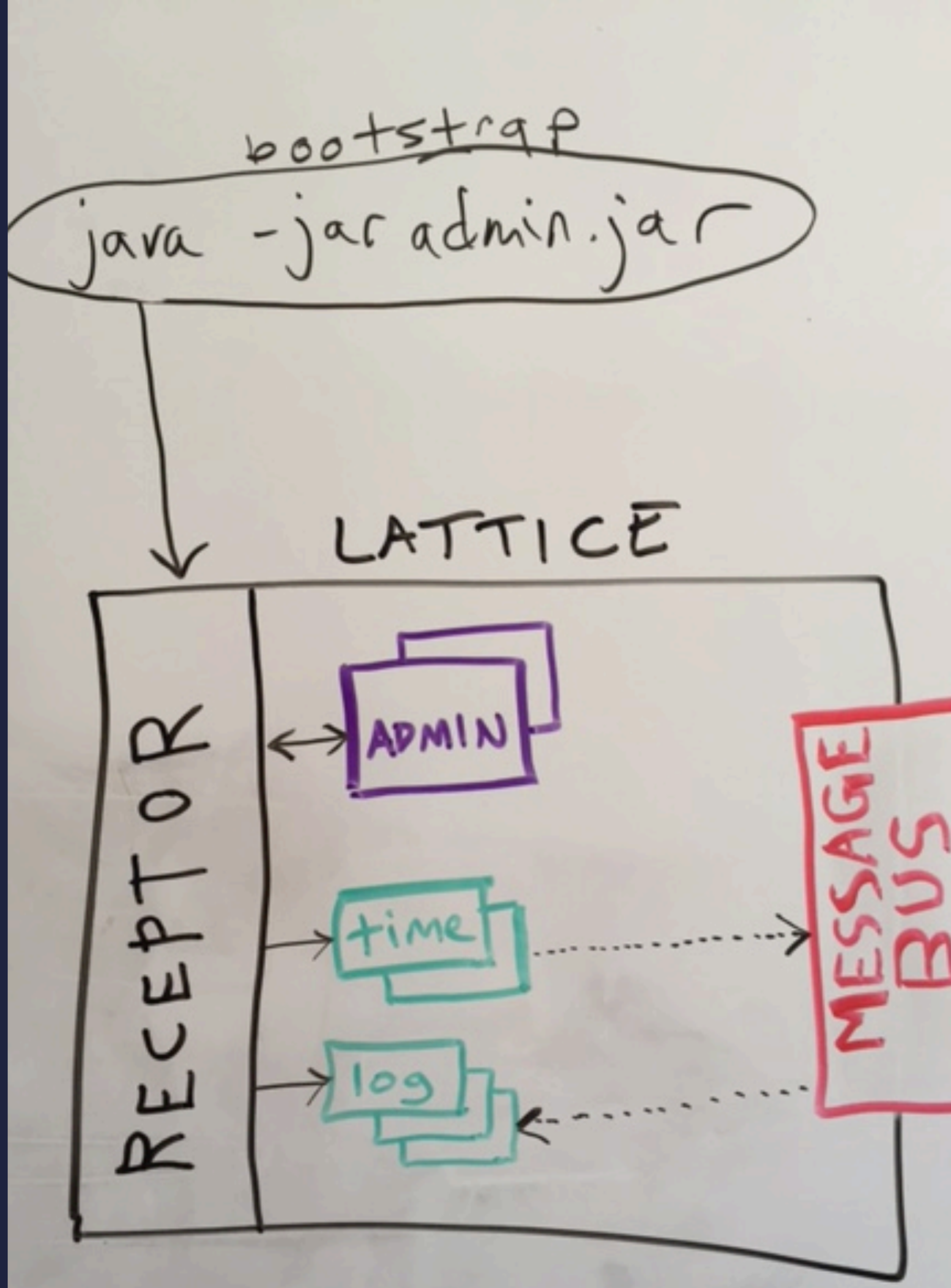


SPRING XD SPI MODEL



SPRING XD

+ LATTICE



DEMO

LEARN MORE!

- Lattice: http://lattice.cf
- Diego Design Notes: https://github.com/cloudfoundry-incubator/diego-design-notes

THANK YOU!