Platforms in the Cloud

Dan Sanderson, Google
December 4, 2014


O'REILLY
Programmi
Google
App E
with P
BUILD & RUN SCALABLE PYTHO
ON GOOGLE'S INFRASTRUCTUR
O'REILLY
Programming
Google
App Engine
with Java
BUILD & RUN SCALABLE JAVA APPS ON GOOGLE'S INFRASTRUCTURE
Dan Sanderson

Software as a Service (SaaS)

Gmail, Google Docs, NetSuite, SugarCRM

Infrastructure as a Service (IaaS)

Google Compute Engine, Amazon EC2; networking, storage

Software as a Service (SaaS)

Gmail, Google Docs, NetSuite, SugarCRM

Platform as a Service (PaaS)

Google App Engine, Heroku, Microsoft Azure

Infrastructure as a Service (IaaS)

Google Compute Engine, Amazon EC2; networking, storage

All problems in computer science can be solved by another level of indirection.

— David Wheeler

- web applications
- managed servers
- automatic scaling

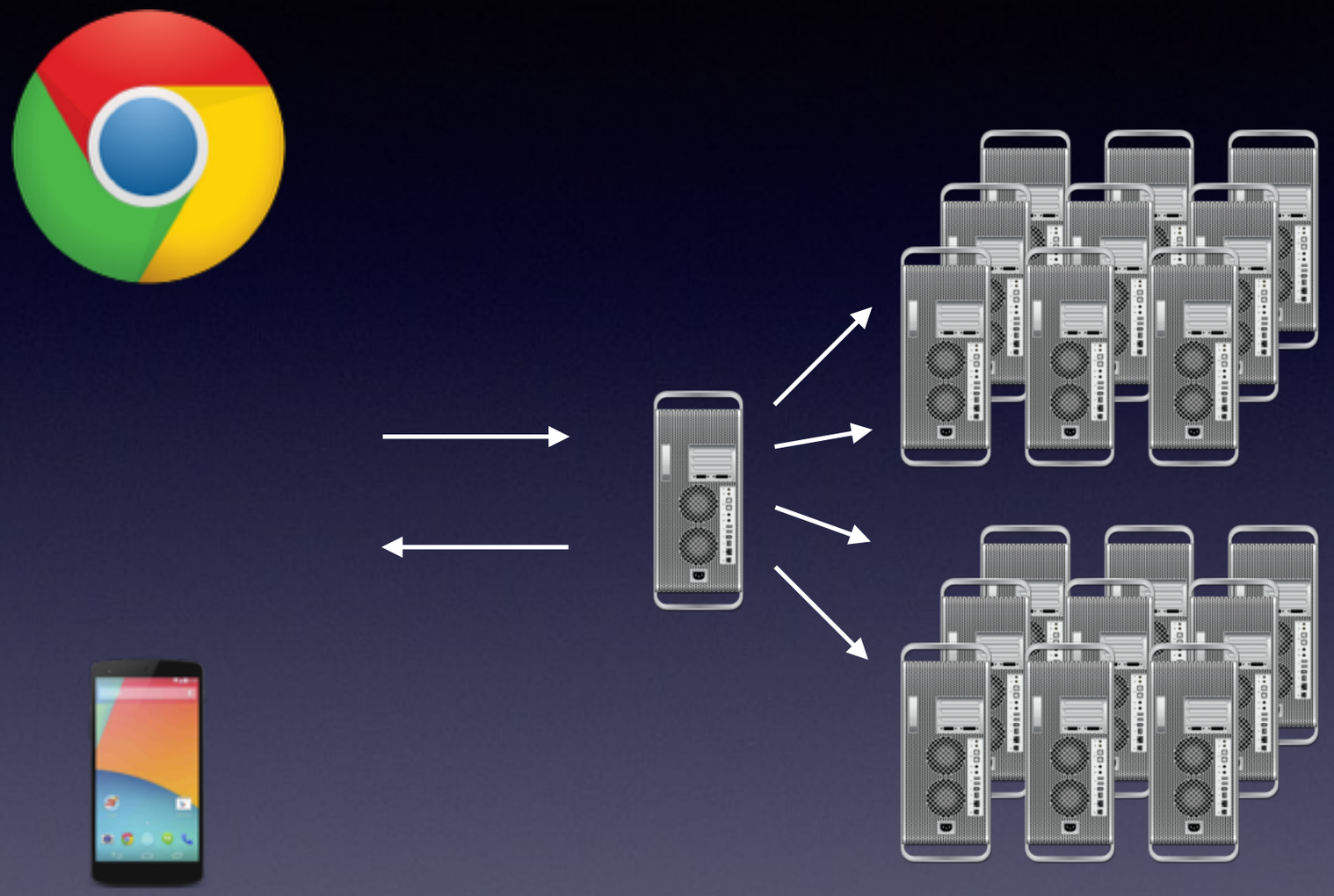
Why PaaS?

client

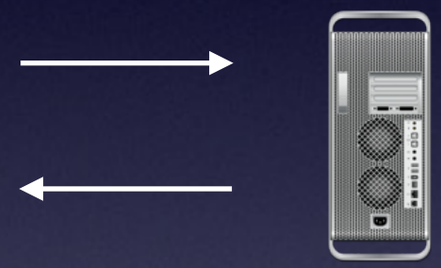
server

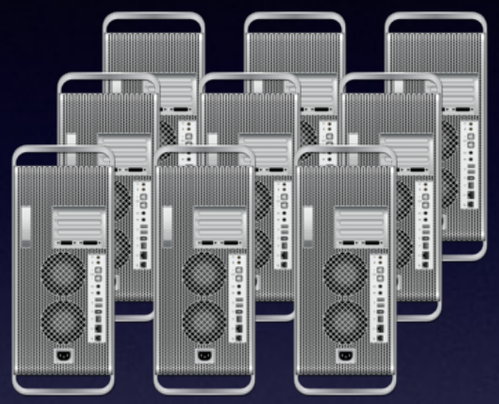
request

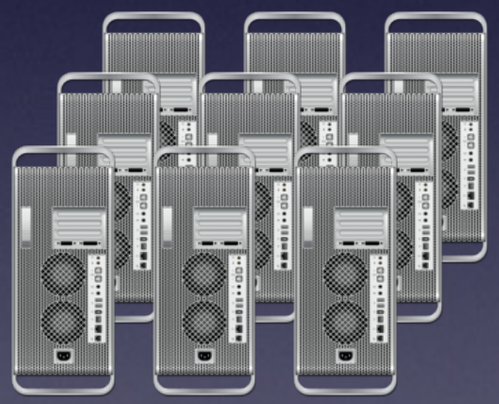
response



load balancer



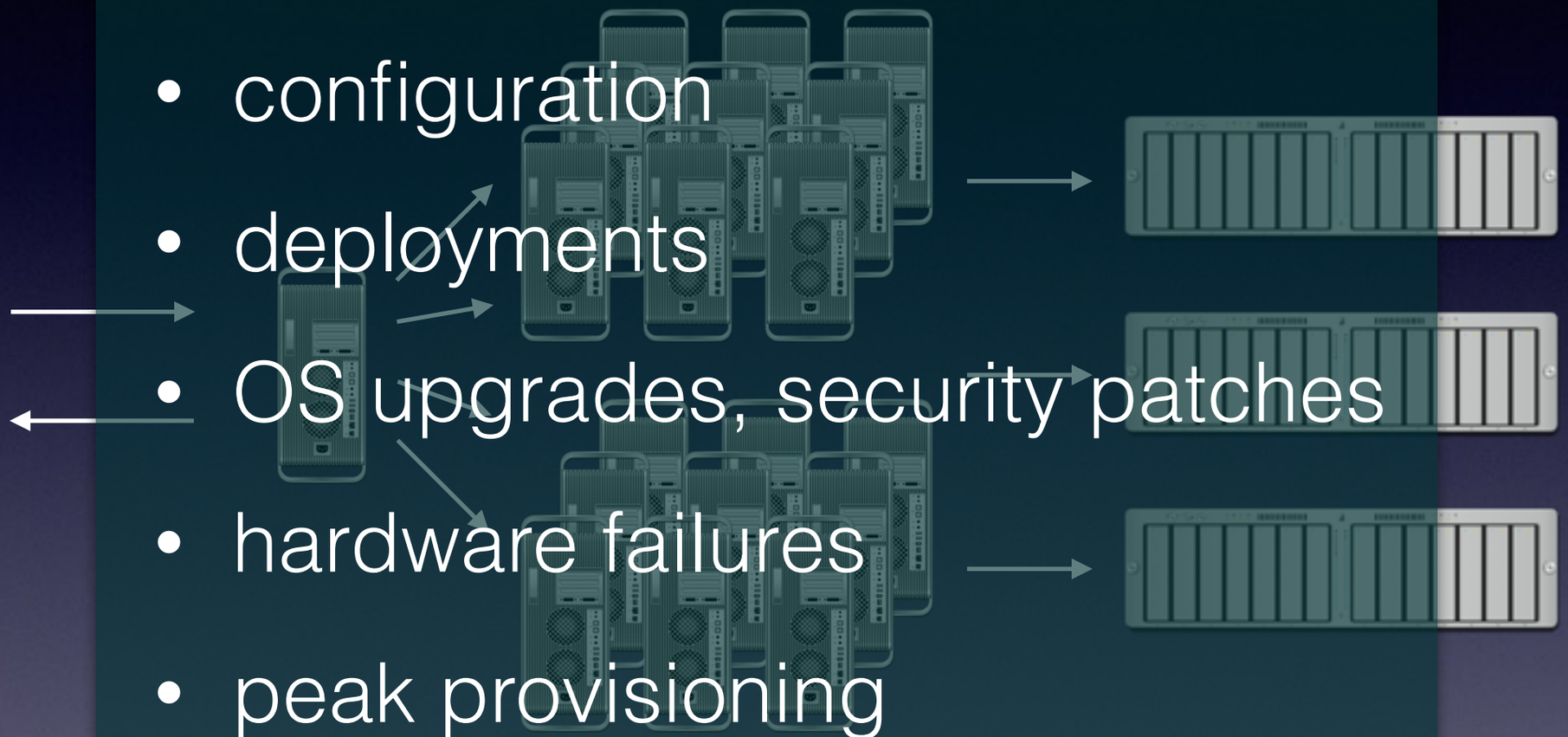




storage

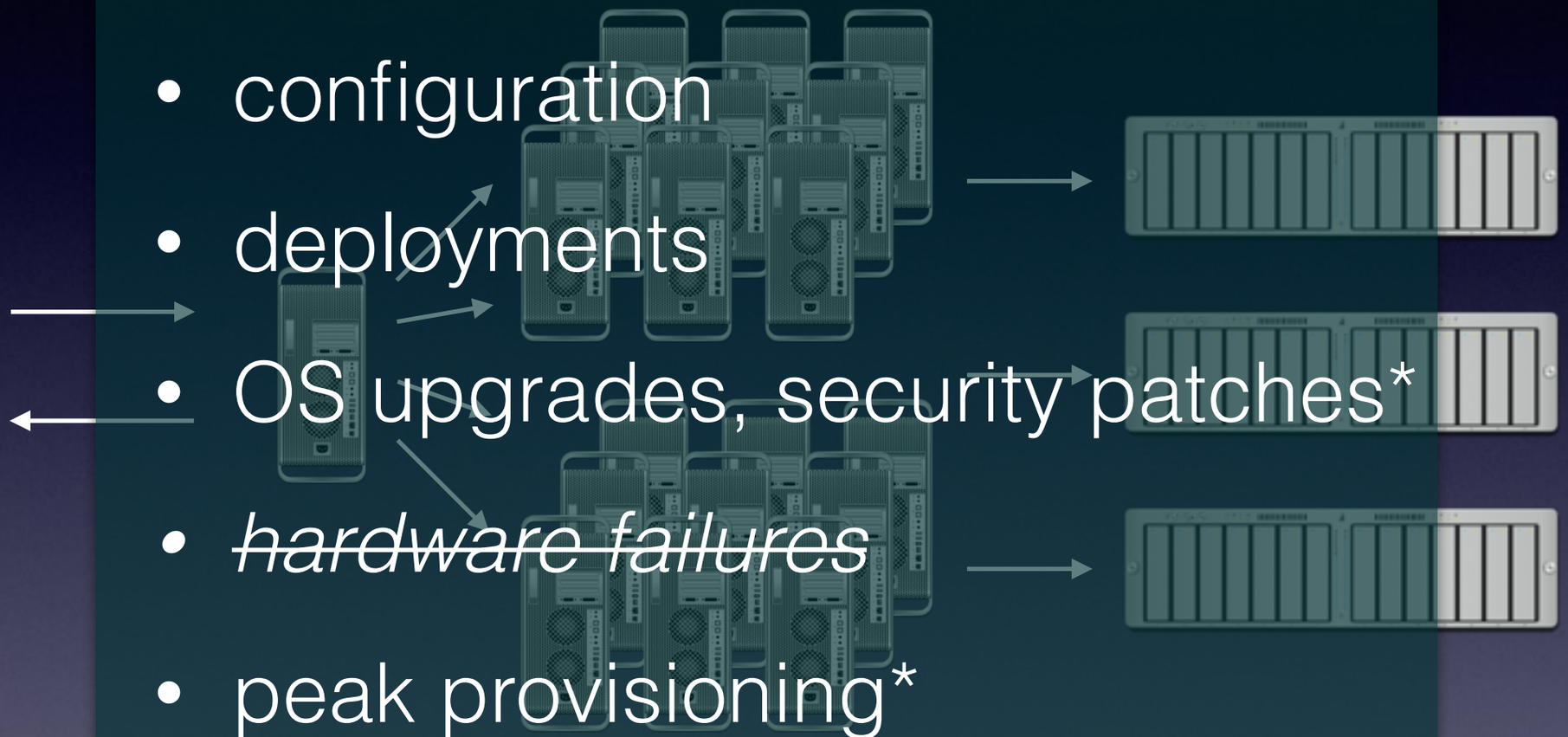
Self Hosted

- configuration
- deployments
- OS upgrades, security patches
- hardware failures
- peak provisioning



IaaS / Managed Servers

- configuration
- deployments
- OS upgrades, security patches*
- ~~*hardware failures*~~
- peak provisioning*



Google App Engine

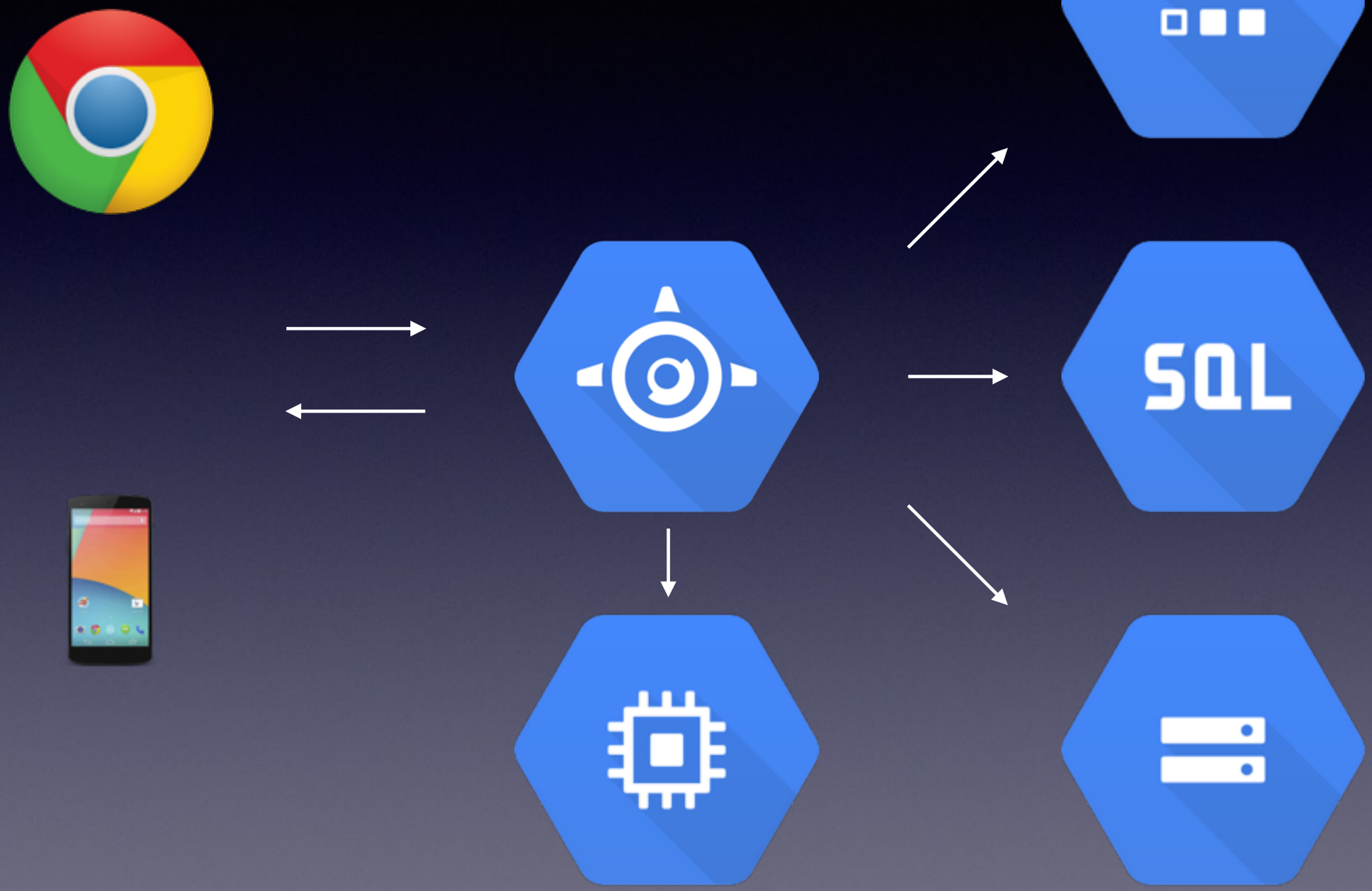
Google App Engine

- Easy deployment
- No servers to manage, no OS to update;
  App Engine does this for you
- Pay for what you use
  - Instance hours, storage, bandwidth,
    service calls

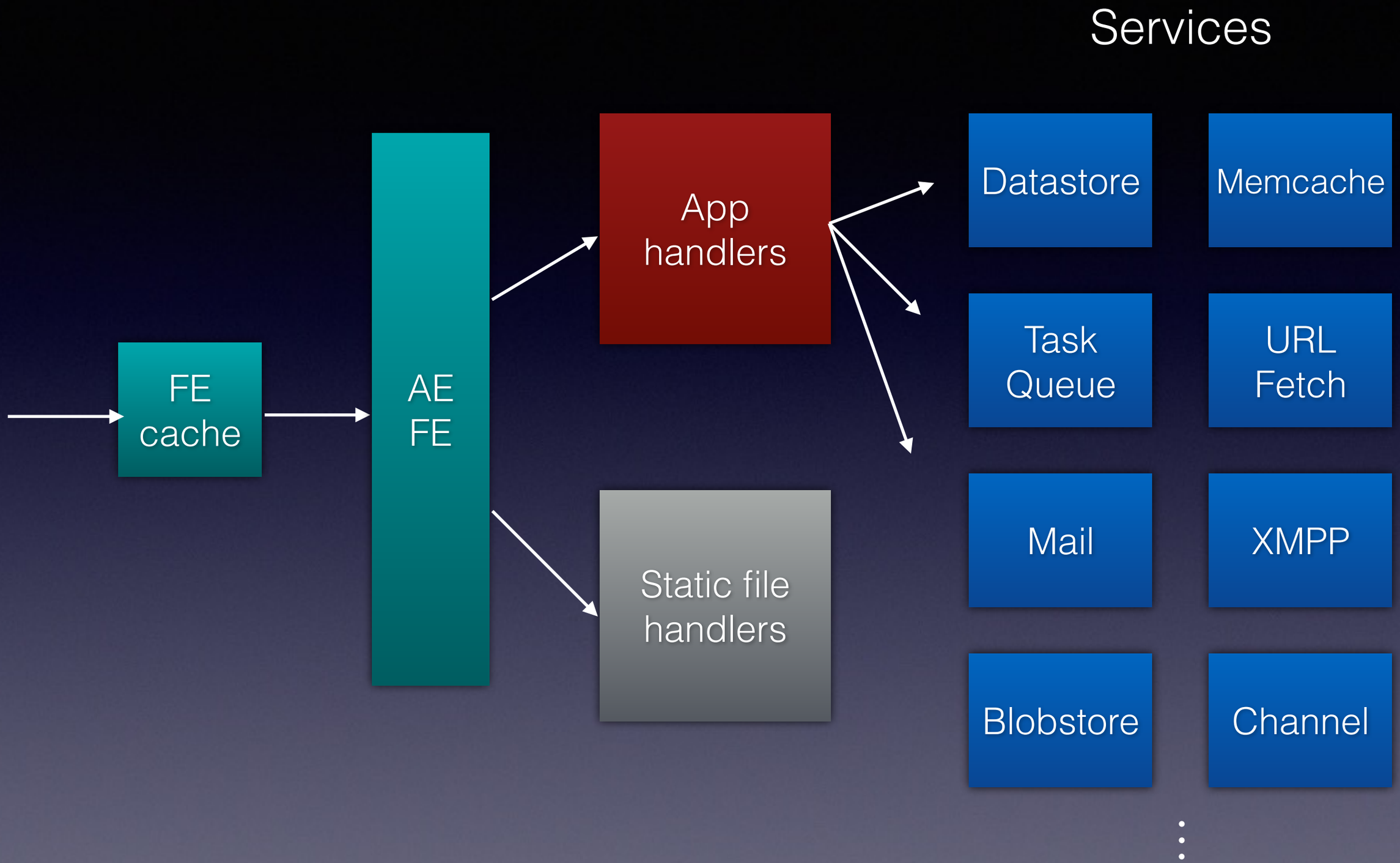
Google App Engine

- Built on Google infrastructure
- Based on Google’s internal best practices
- Based on standard technologies


SQL


SQL

App Engine Architecture


Services
FE cache
AE FE
App handlers
Static file handlers
Datastore
Memcache
Task Queue
URL Fetch
Mail
XMPP
Blobstore
Channel

Instances and Request Handlers

Instances and Request Handlers

- Request handlers are ephemeral:
  now you see them, now you don’t
- Can’t rely on data persistence between requests
- Use storage services to persist data

Request handler

Request handler

Request handler

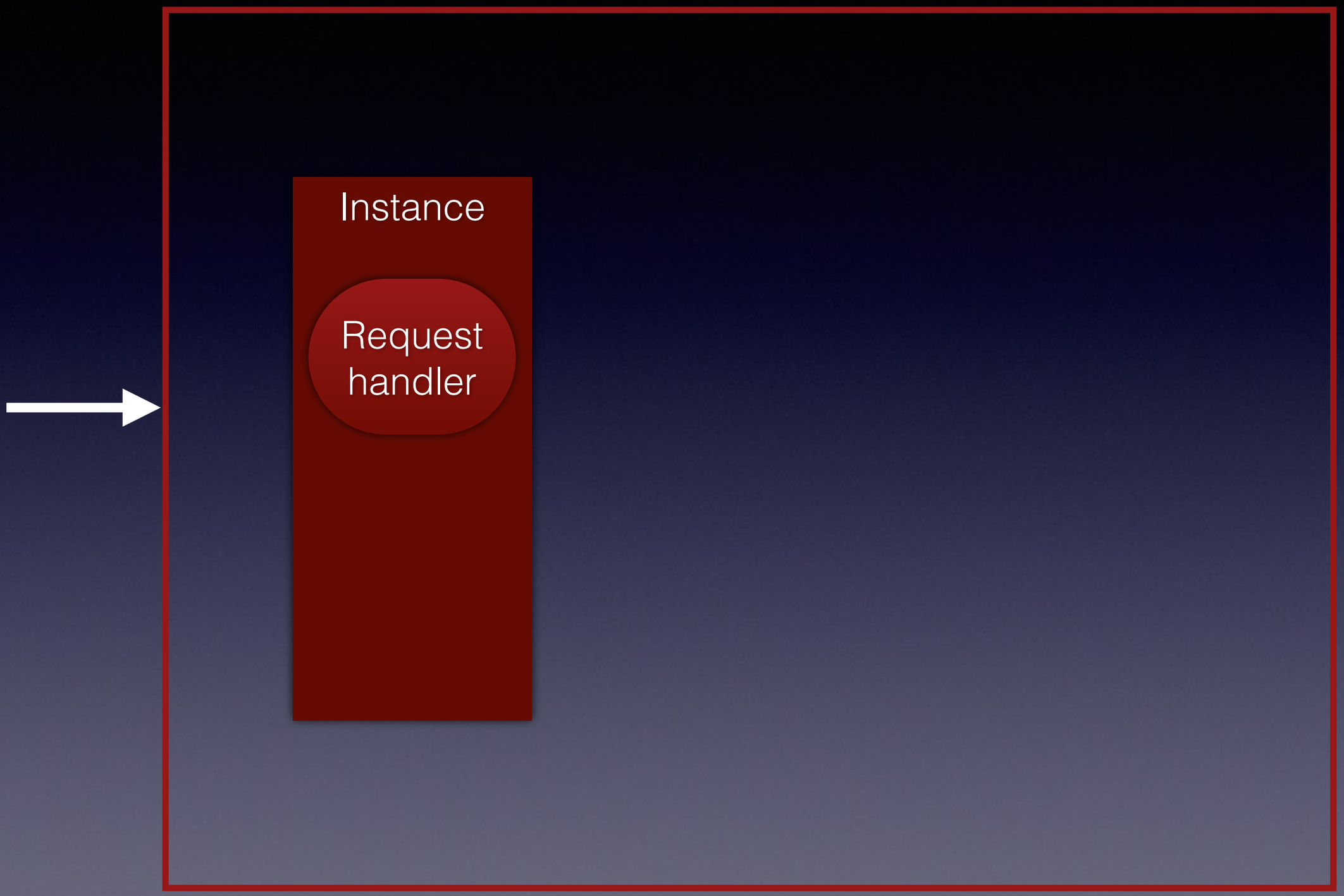
Request handler

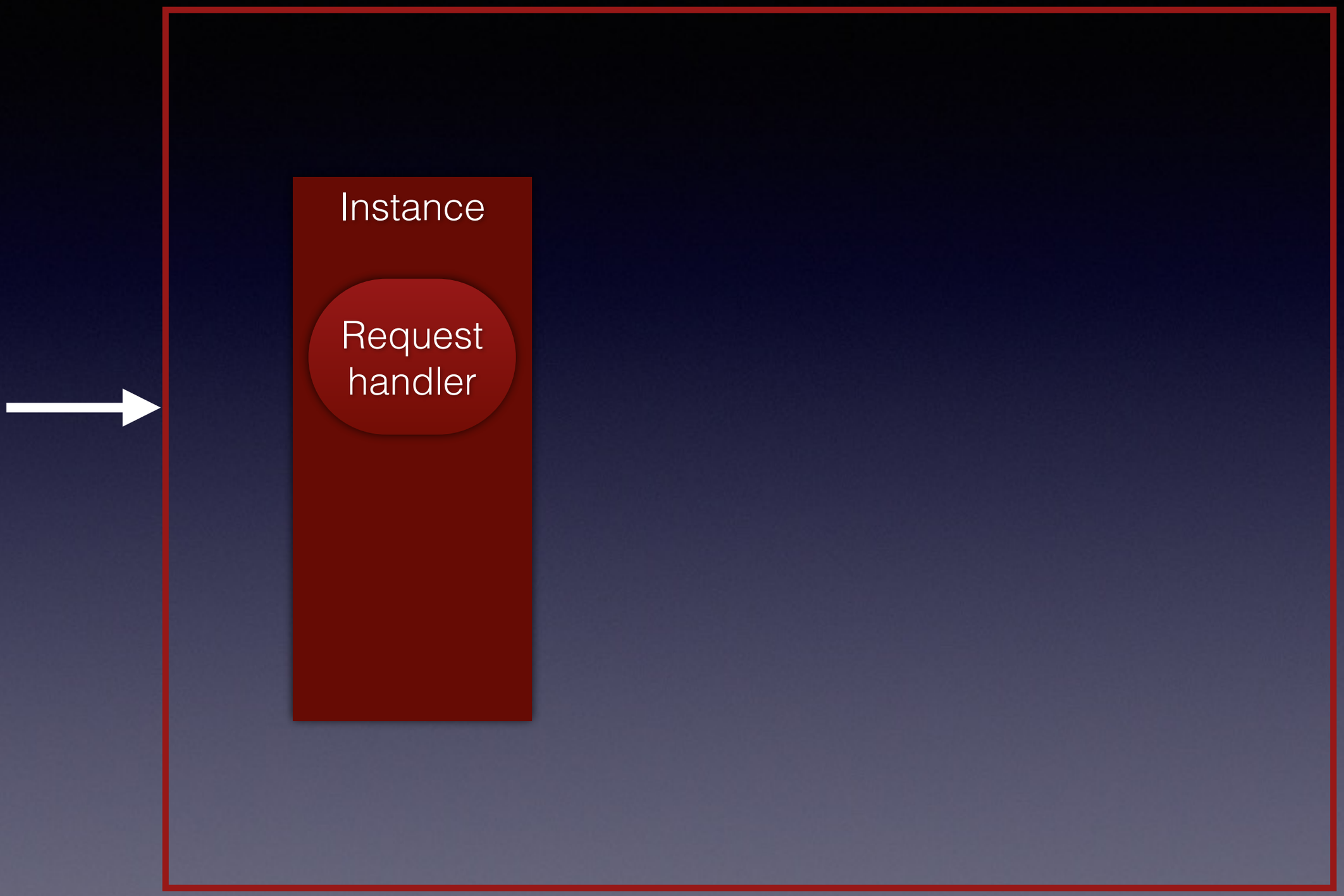
...

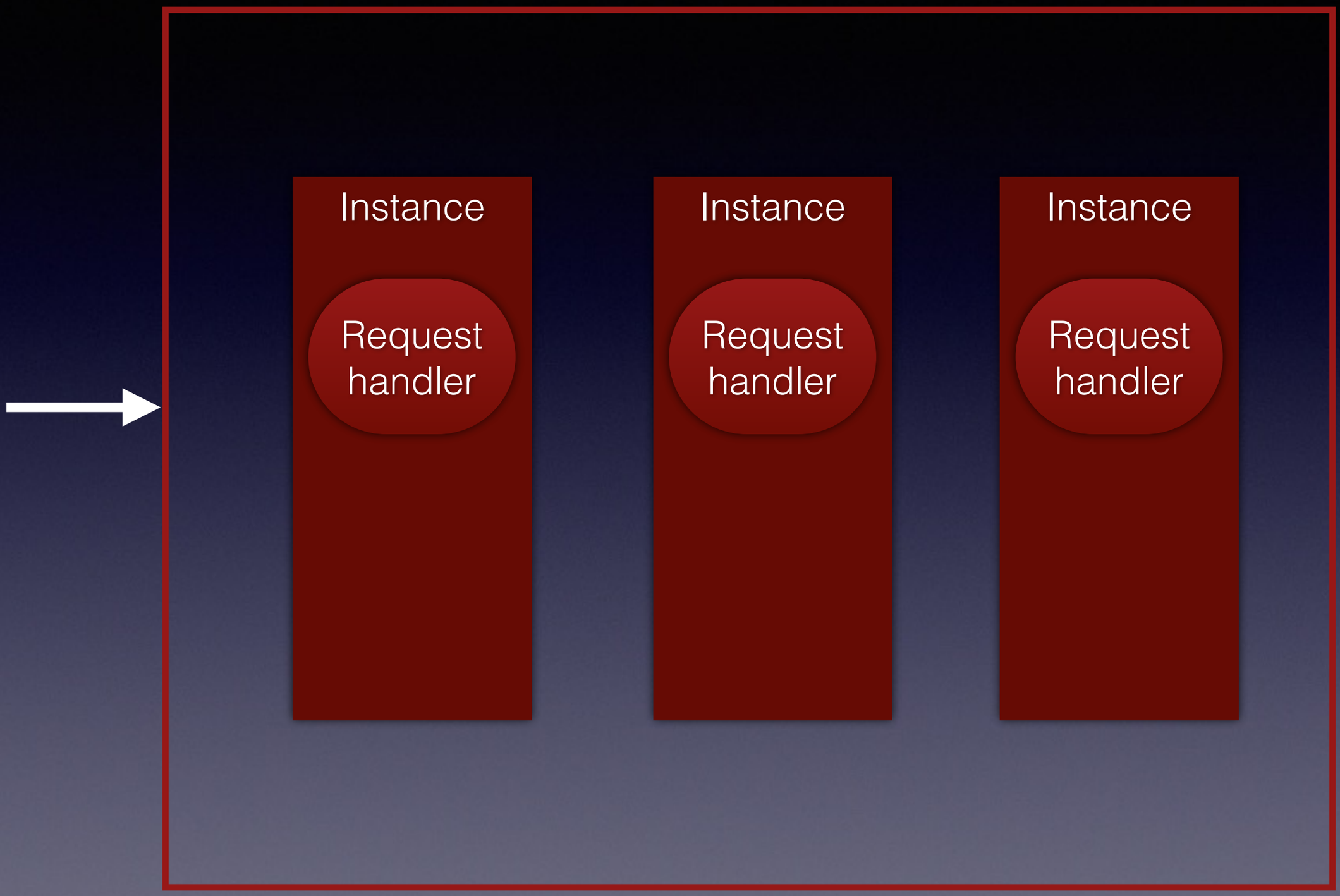
...

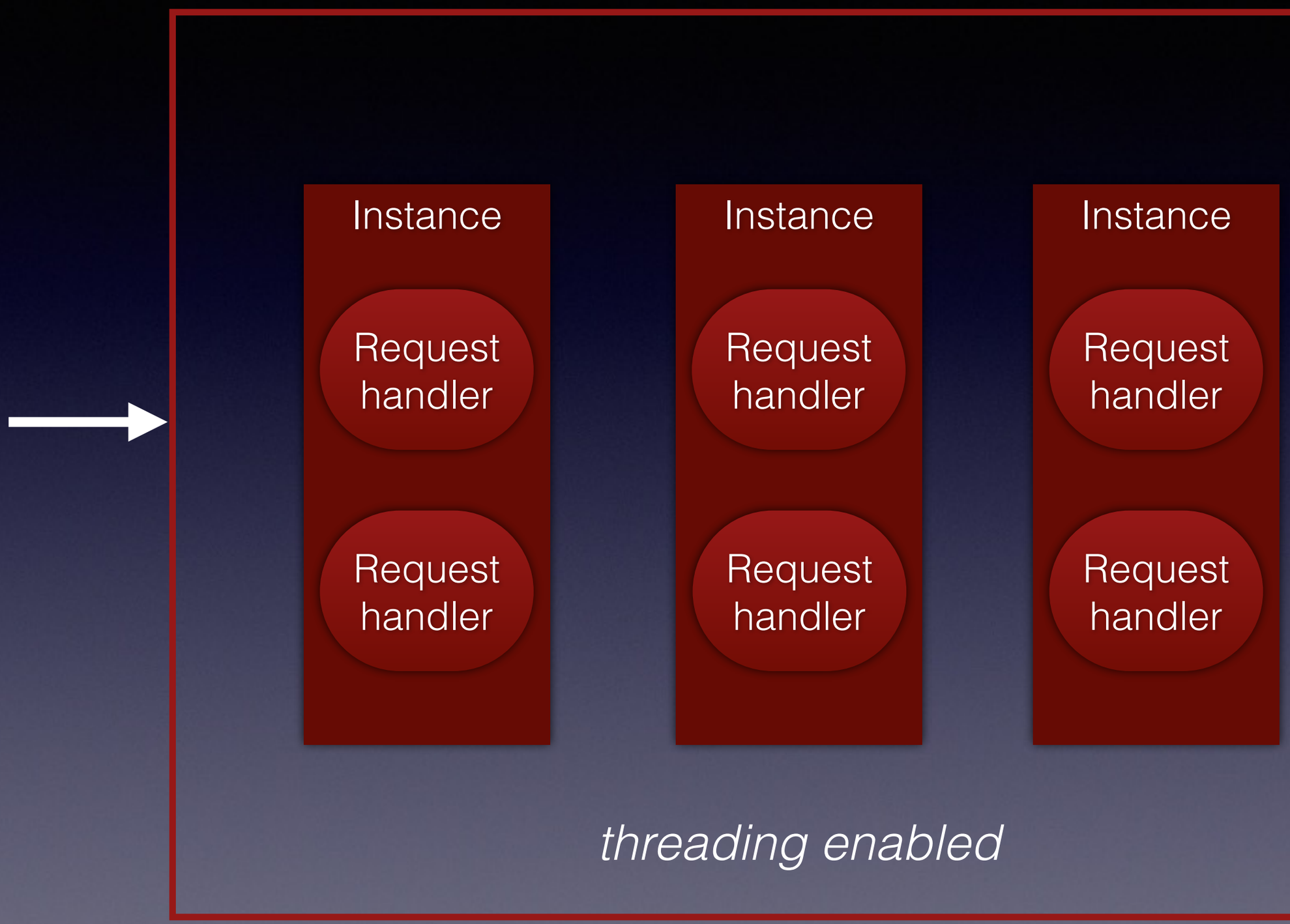
Instances and Request Handlers

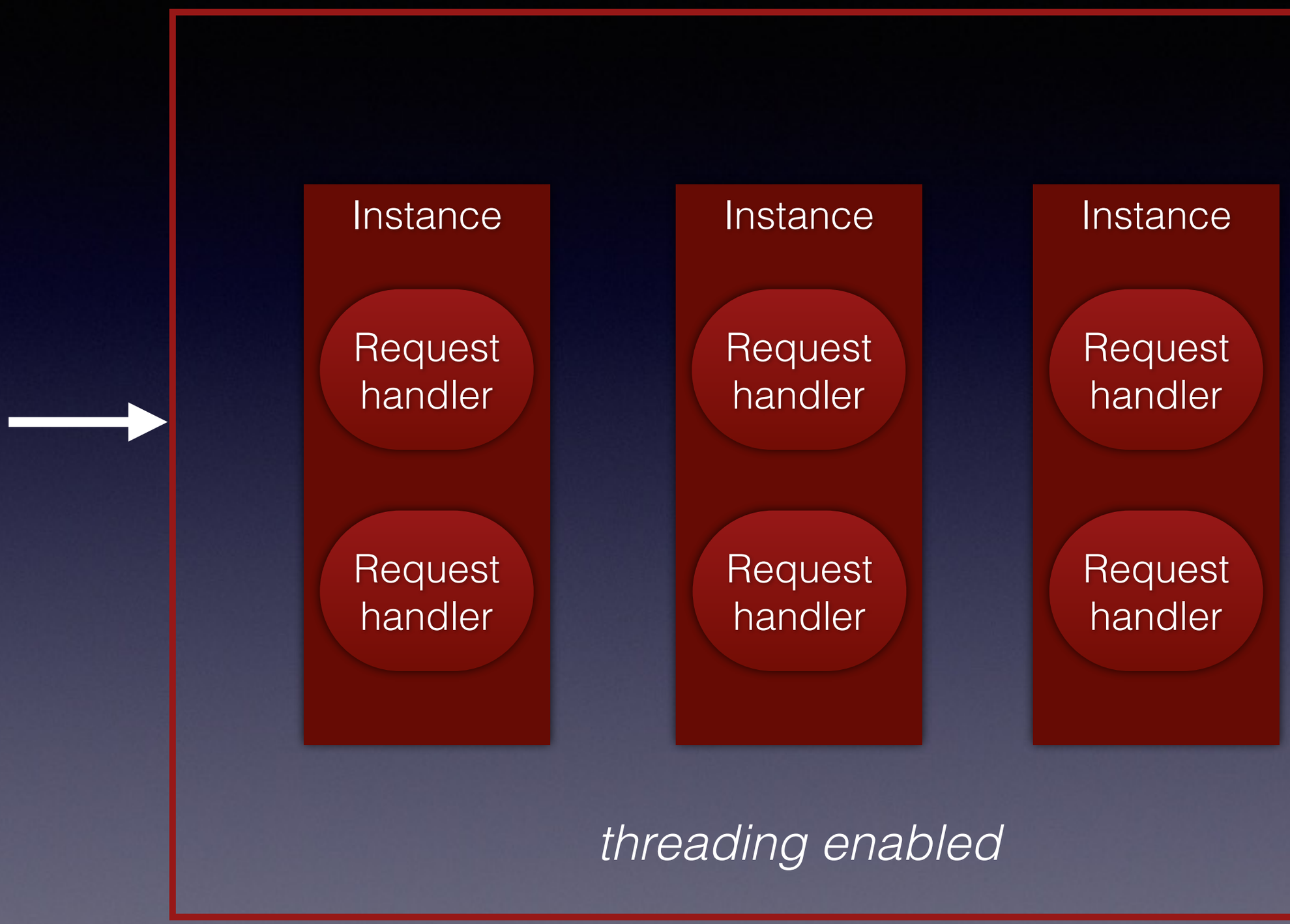
- In practice, app initialization is expensive
- An app instance is long running, can handle multiple requests in its lifetime
- Environment initialized; instance memory loaded
- Started and stopped as needed
- Can’t rely on a single user’s session to go to the same instance

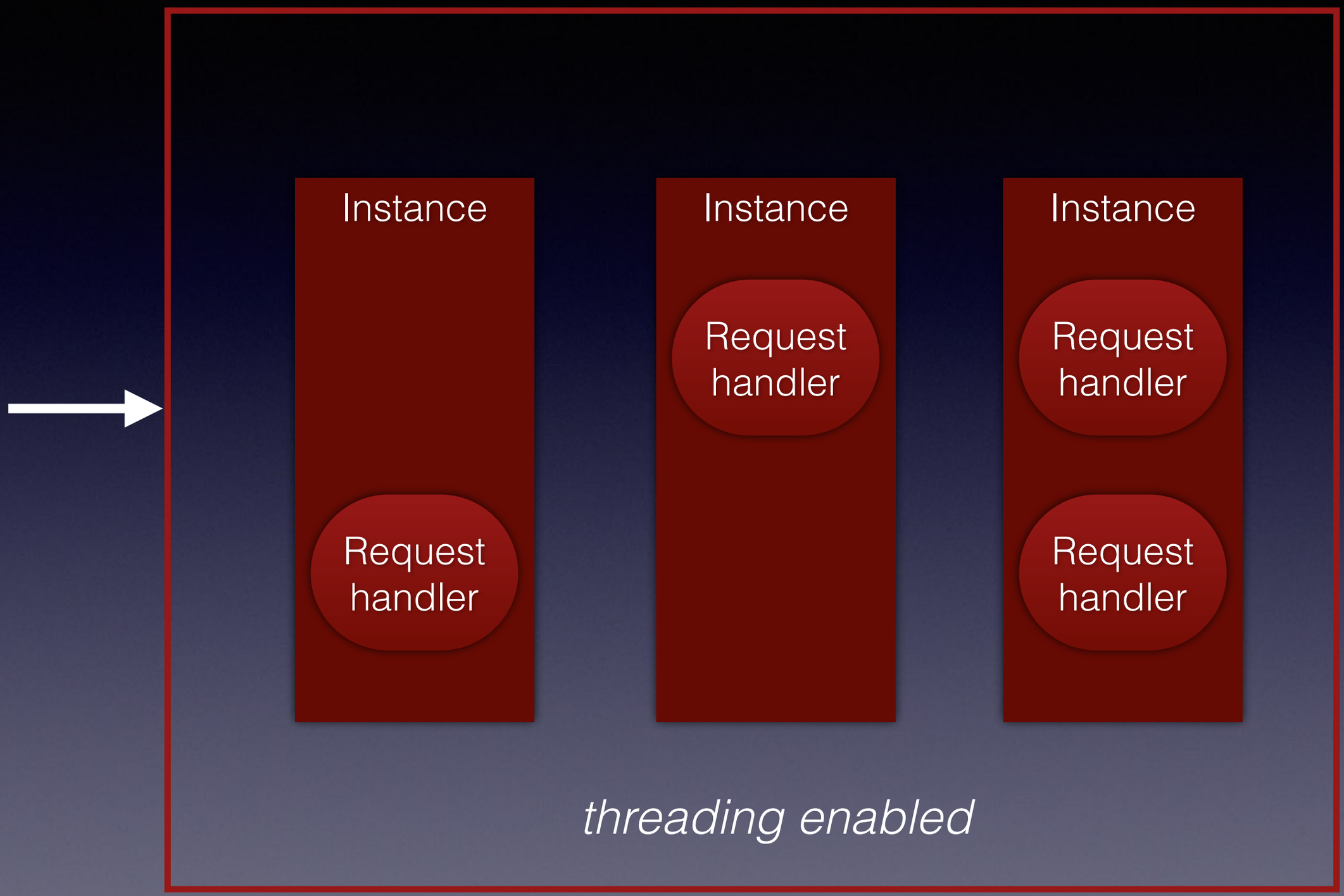

Instance
Request handler
...


Instance
Request handler
...


Instance
Request handler
Instance
Request handler
Instance
Request handler


Instance
Request handler
Request handler
Instance
Request handler
Request handler
Instance
Request handler
Request handler
...
threading enabled


Instance
Request handler
Request handler
Instance
Request handler
Request handler
Instance
Request handler
Request handler
...
threading enabled


Instance
Request handler
Instance
Request handler
Instance
Request handler
Request handler
...
threading enabled

Runtime Environments

- Sandboxing
  - Data isolation
  - Performance isolation
- Sandboxing → scalability

Runtime Environments

- Limits
  - Request timer
  - Restricted access to filesystem, sockets
  - More performance isolation: RAM, CPU
  - Data sizes: requests, responses, API calls, storage objects
- Limits → scalability

Runtime Environments

Python

Java

Go

PHP

Services

- Features with their own scalable infrastructure
- Architecturally distinct from the runtime environments
- Synchronous and asynchronous calling APIs
- Data storage, communication, data processing

Google Cloud Datastore

Google Cloud Datastore

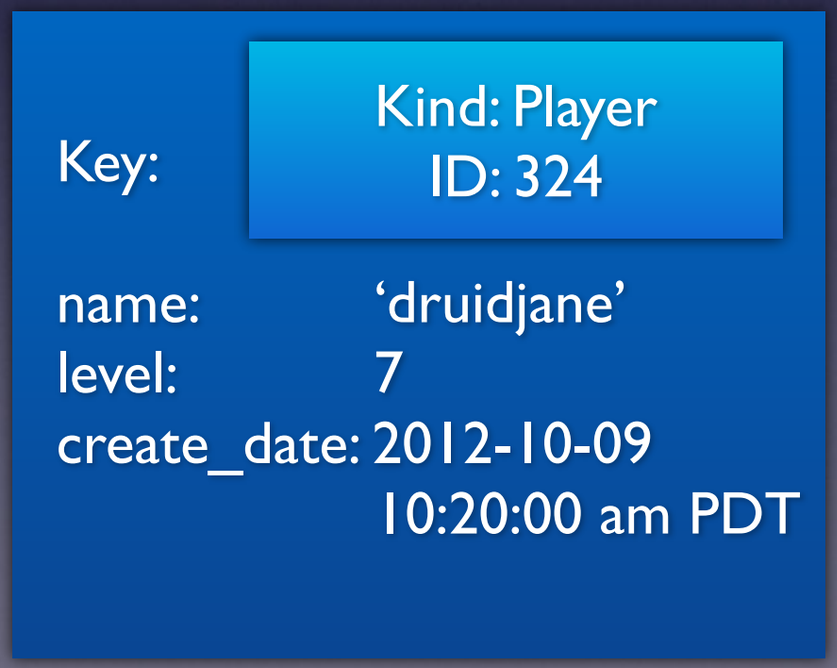
- Scalable object storage
- Based on high-powered key-value storage ("BigTable"); see also "MegaStore"
- Named properties, typed values
- "Schemaless;" data modeling in app code
- Replication using Paxos

Google Cloud Datastore

- entities
- keys: kind, ID, […]
- properties, typed values

Google Cloud Datastore

```
p3 = Player(name='druidjane',
            level=7,
            create_date=now)
p3key = p3.put()
```



Google Cloud Datastore

```
p3key = ndb.Key('Player', 324)
p3 = p3key.get()

if p3.level > 5:
# . . .
```

Key: Kind: Player, ID: 324

name: 'druidjane'
level: 7
create_date: 2012-10-09 10:20:00 am PDT

Google Cloud Datastore

```
class Player(ndb.Model):
  name = ndb.StringProperty()
  level = ndb.IntegerProperty()
  create_date = ndb.DateTimeProperty()
```

```
p1 = Player()
p1.level = 7
p1.put()

p2 = Player()
p2.level = ‘warrior’  # BadValueError
p2.put()
```

Google Cloud Datastore

- queries
  - kind (Player)
    - property filters (level > 7),
      property sort order (creation_date ascending)
- indexes
  - key, property (Player : level)
  - custom indexes

Google Cloud Datastore

Every query is served by reading rows from an index.

Indexes are updated as data is updated.

Google Cloud Datastore

Query speed is proportional to
the size of the result set,
not the size of the data set.

Google Cloud Datastore

- transactions
- local vs. global transactions
- datastore transaction locality = *entity groups*
  - defined by the key
- strong consistency vs. eventual consistency

From PaaS to IaaS

- automatic scaling
- streamlined runtime
- optimized for small units of computation

Managed VMs

- automatic scaling
- full virtual machines
- more flexible software

- manual scaling
- full virtual machines
- suitable for large units of computation

Google Container Engine (GKE)

Thank you!

cloud.google.com

ae-book.appspot.com

Programming Google App Engine with Python, … with Java
Early Access available now

Dan Sanderson
profiles.google.com/dan.sanderson