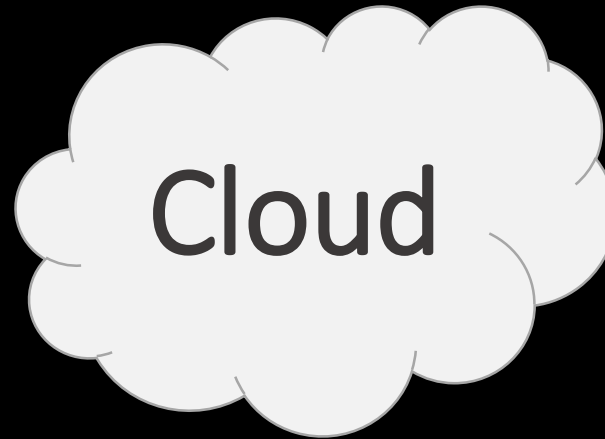
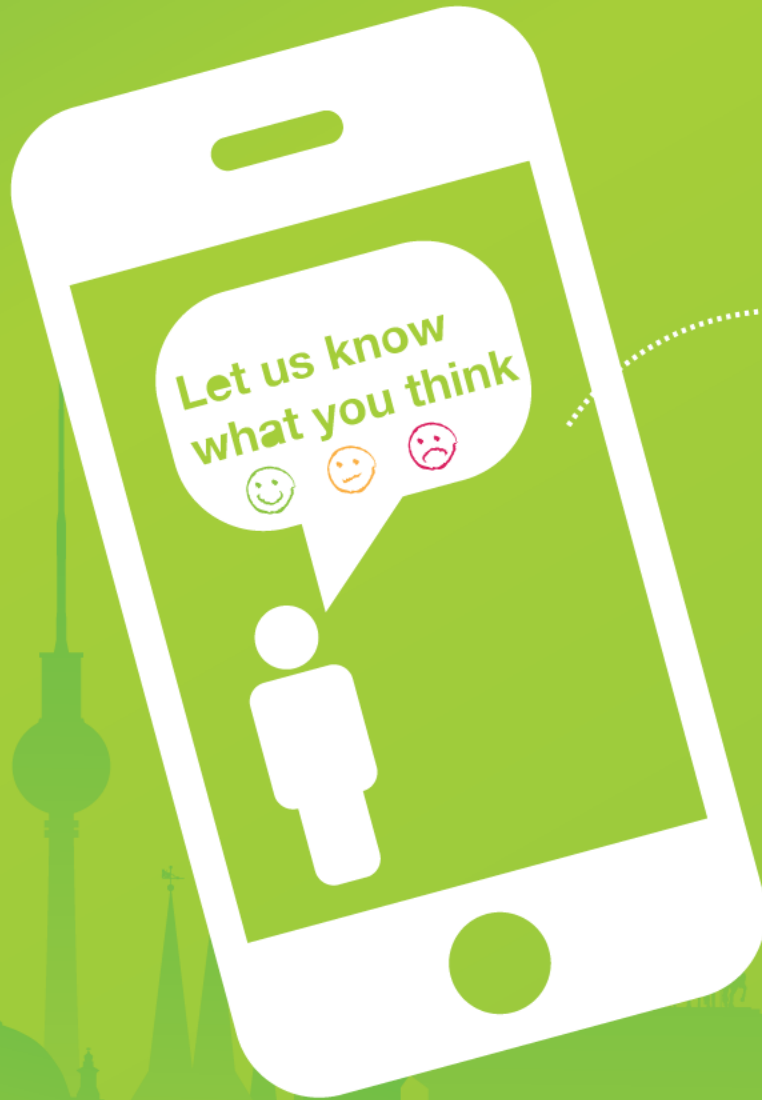


Architecting for the



 @axelfontaine



**Click 'engage'
to rate sessions
and ask questions**



About Axel Fontaine



- Founder and CEO of Boxfuse
- Over 15 years industry experience
- Continuous Delivery expert
- Regular speaker at tech conferences
- JavaOne RockStar in 2014

 @axelfontaine



Flyway

flywaydb.org



boxfuse

boxfuse.com

about questions

POLL:

what type of **infrastructure** are you running on?

- On Premise
- Colocation
- Root Server
- Cloud

what is special about the cloud ??



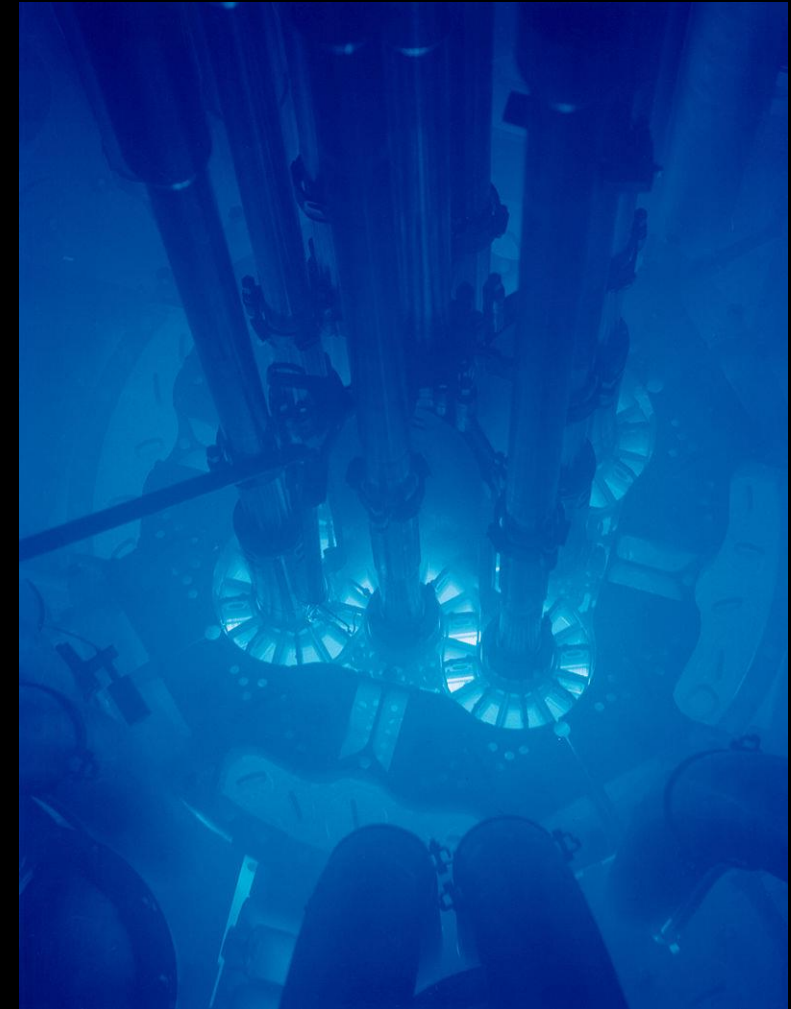
*Every day, AWS adds
enough server capacity
to power the whole \$7B
enterprise Amazon.com
was in 2004.
Weekends included.*





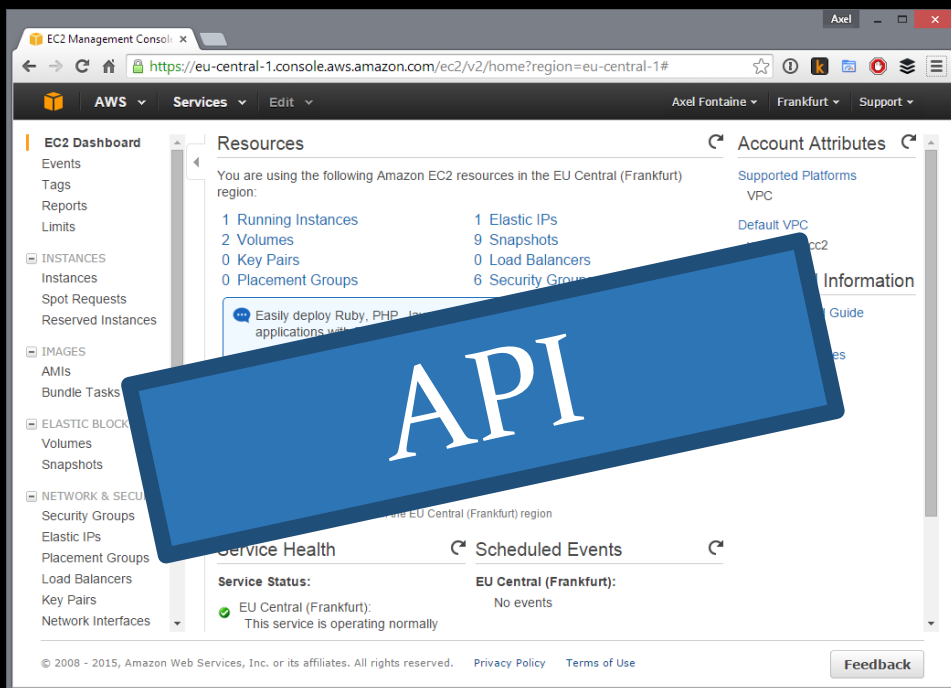
Control Plane

"RIAN archive 341194 Kursk Nuclear Power Plant" by RIA Novosti archive image #341194, merged by yakovlev, CC-BY-SA 3.0, Licensed under CC BY-SA 3.0 via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:RIAN_archive_341194_Kursk_Nuclear_Power_Plant.jpg#mediaviewer/File:RIAN_archive_341194_Kursk_Nuclear_Power_Plant.jpg

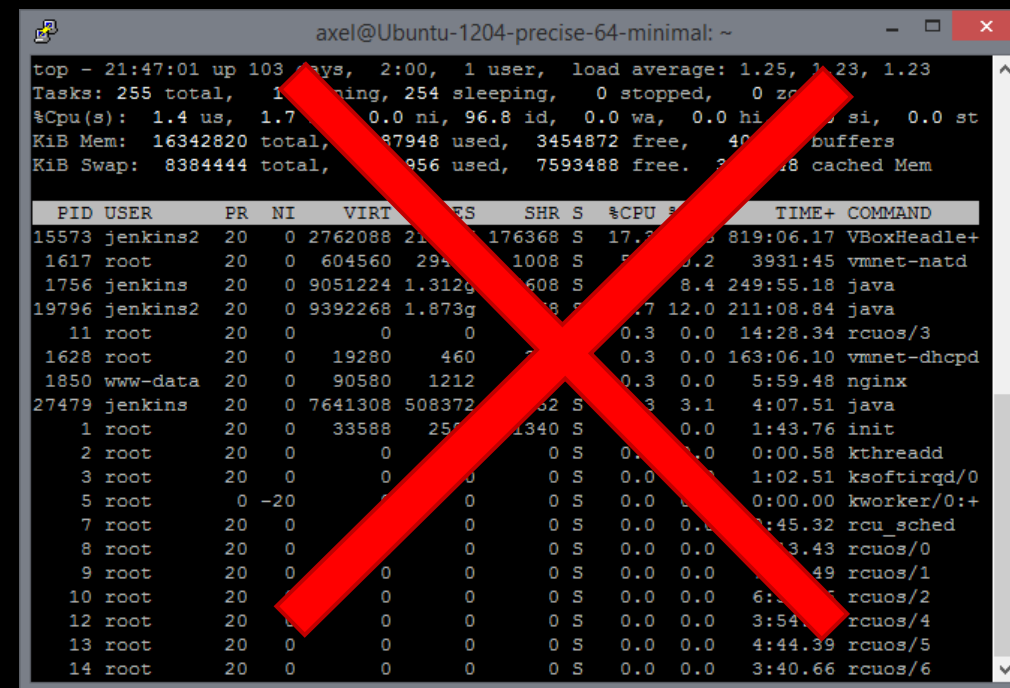


Data Plane

"Advanced Test Reactor" by Argonne National Laboratory - originally posted to Flickr as Advanced Test Reactor Core, Idaho National Laboratory, uploaded using Flickr Commons. Licensed under CC BY-SA 2.0 via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:Advanced_Test_Reactor.jpg#mediaviewer/File:Advanced_Test_Reactor.jpg



Control Plane

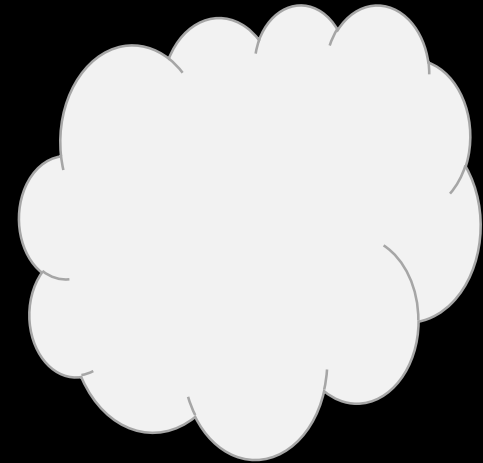


Data Plane

benefits of the cloud

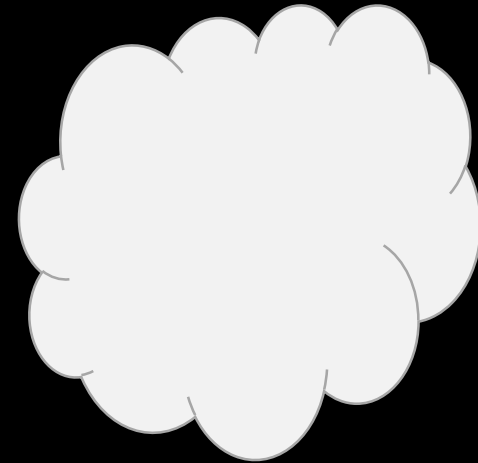
- ✓ Shift to a world of abundance
(no more resource scarcity)
- ✓ Clean Control Plane/Data Plane split
with API-based provisioning
- ✓ Cost-based Architectures
with the ability to turn infrastructure off

moving to the cloud



lift & shift

(= the **naïve** approach)



lift & shift

(= the **naïve** approach)

Congratulations! You now have:

- A more expensive Hetzner/OVH
- Lots of (too much?) trust in your cloud provider
- Potential legal trouble due to data privacy laws

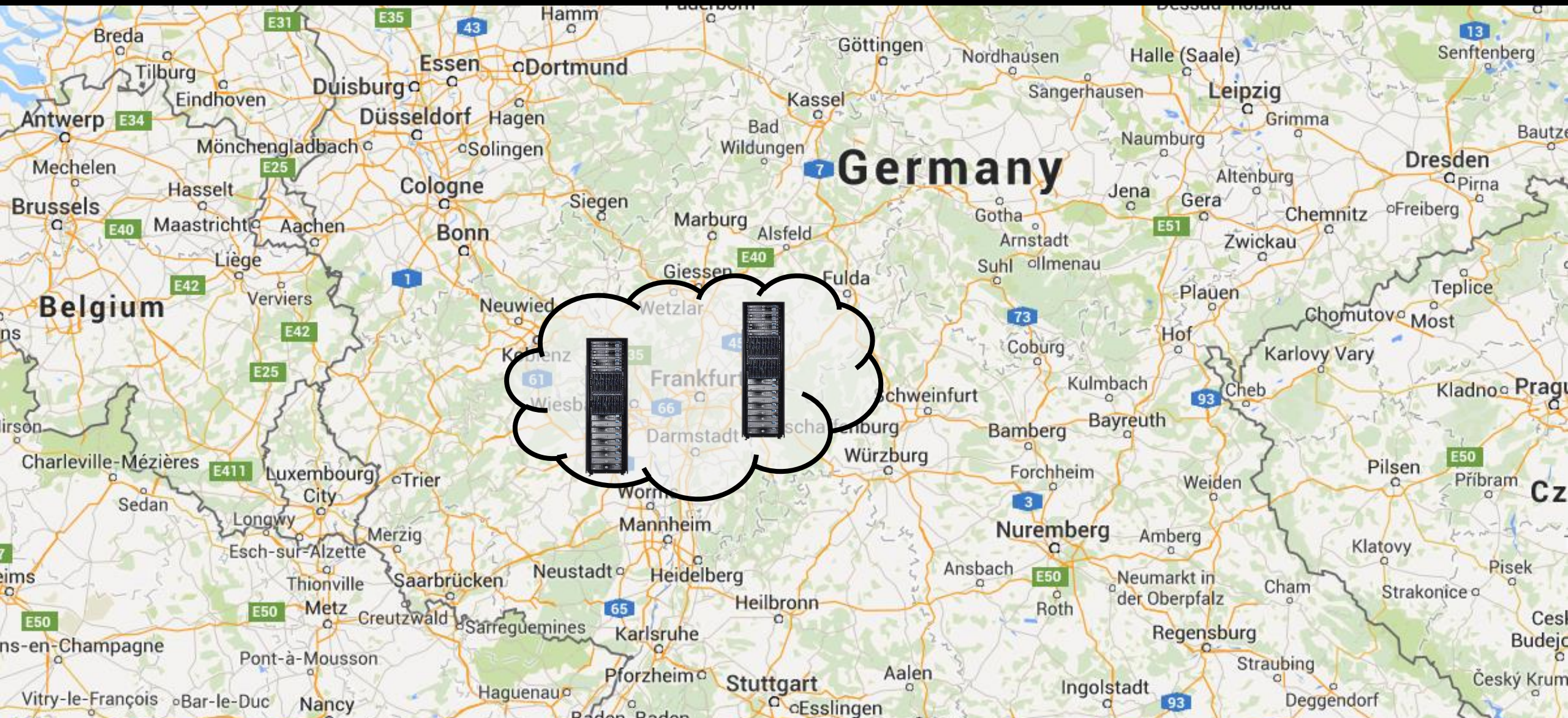


understanding the cloud

regions



availability zones



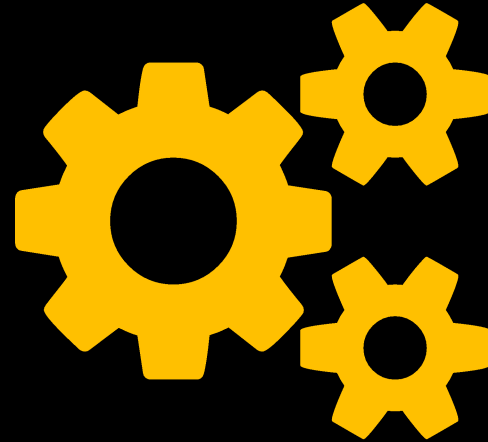
building blocks



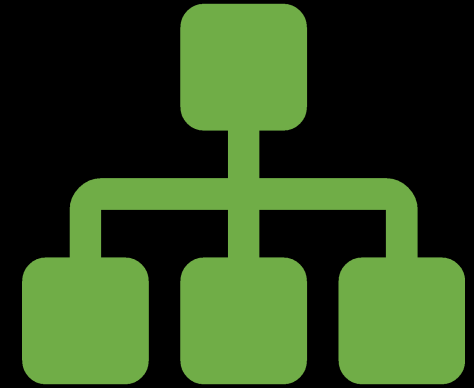
building blocks



Storage



Compute



Network

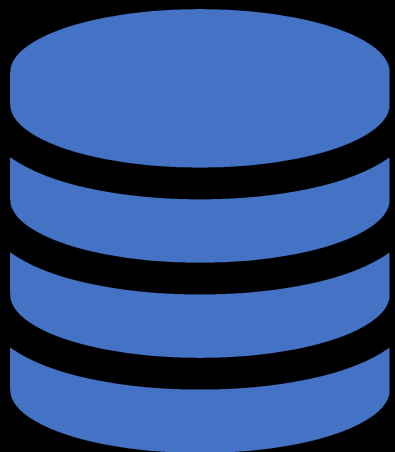
Security



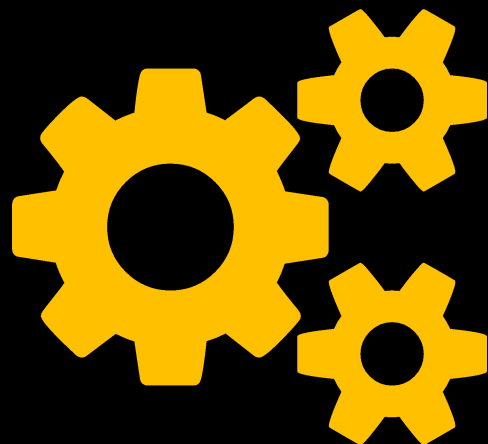
The hard Truth about Security

1. Always breakable with infinite time & resources
2. Must make it more complicated/expensive to break than it's worth (use defense in depth!)
3. Has a usability cost
4. Almost always about the data

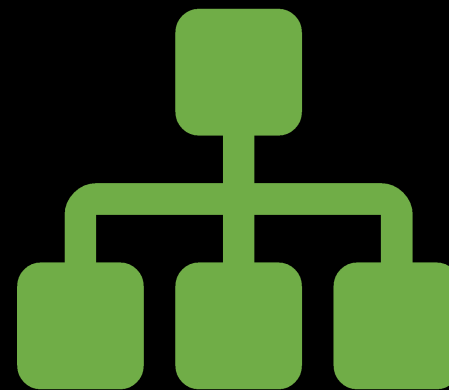
the 3 states of data



Data at Rest



Data in Use



Data in Motion



*Trusting your neighbors
is good. But it's even
better to put a good
lock on the door.*

Werner Vogels
CTO of an online book shop

Data in Motion



TLS / SSL

Data in Use & at Rest



Client-side
encryption

Client-side encryption



- ✓ Encrypt sensitive & personally identifiable data
- ✓ Use different Encryption key for each field/record
- ✓ Encrypt Encryption Key using Key encrypting Key
- ✓ Secure & Rotate the Key encrypting Key

Key Management



In App
€



KMS
€€



HSM
€€€€€€

Querying Encrypted Data

Id	Encrypted
123	#!azw\b
456	67ftf6&)

Other
clear text
field

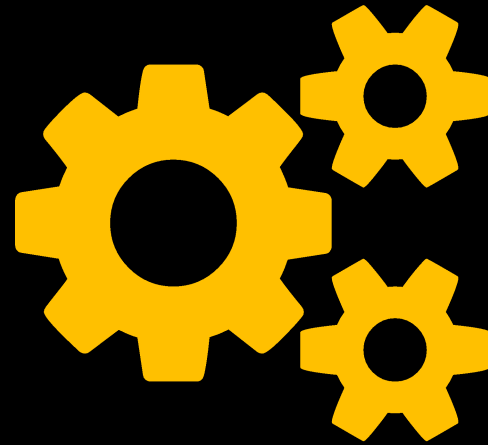
Hmac	Encrypted
5841545832	#!azw\b
0219237127	67ftf6&)

Exact Match
=> Hmac

Low Fi	Encrypted
48.5	#!azw\b
37.2	67ftf6&)

Range
=> Lower fidelity

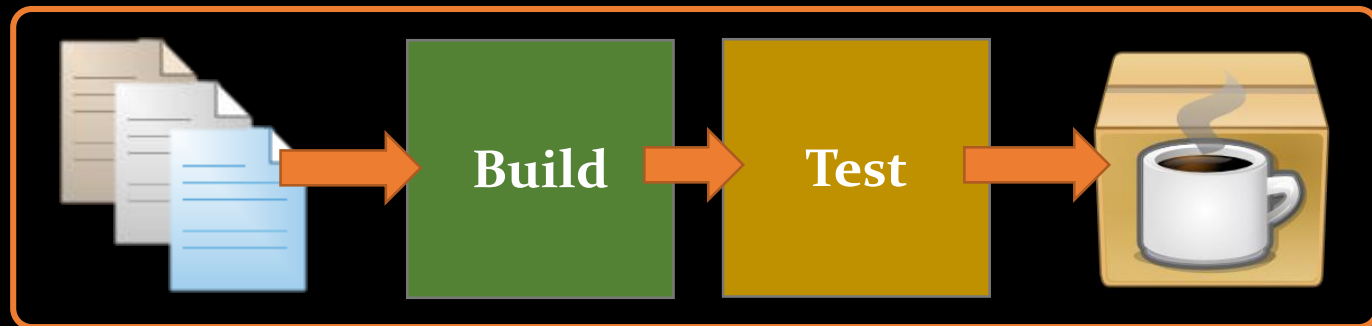
=> Use transparent persistence layer converters!

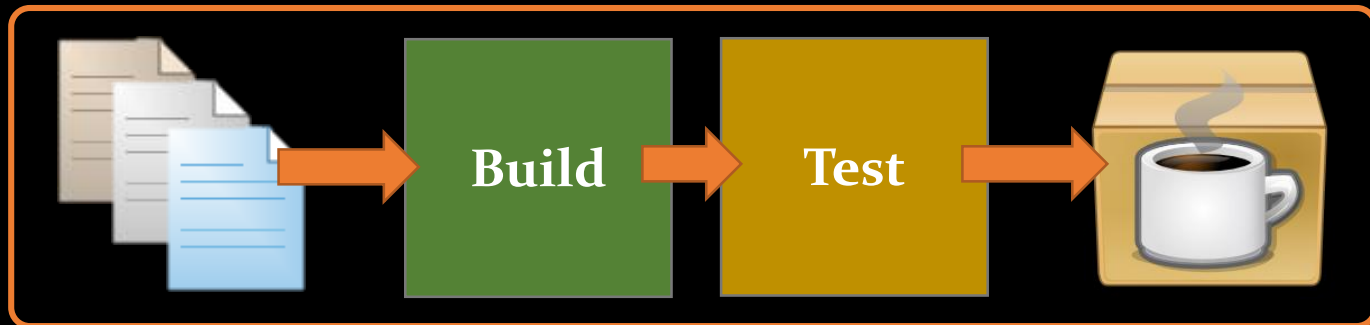


Compute

POLL:
which level of **automation** are you at?

- Build
- Unit Tests
- Continuous Integration
- Acceptance Tests
- Continuous Deployment (Code)
- Continuous Deployment (Code + DB + Configuration)
- Infrastructure

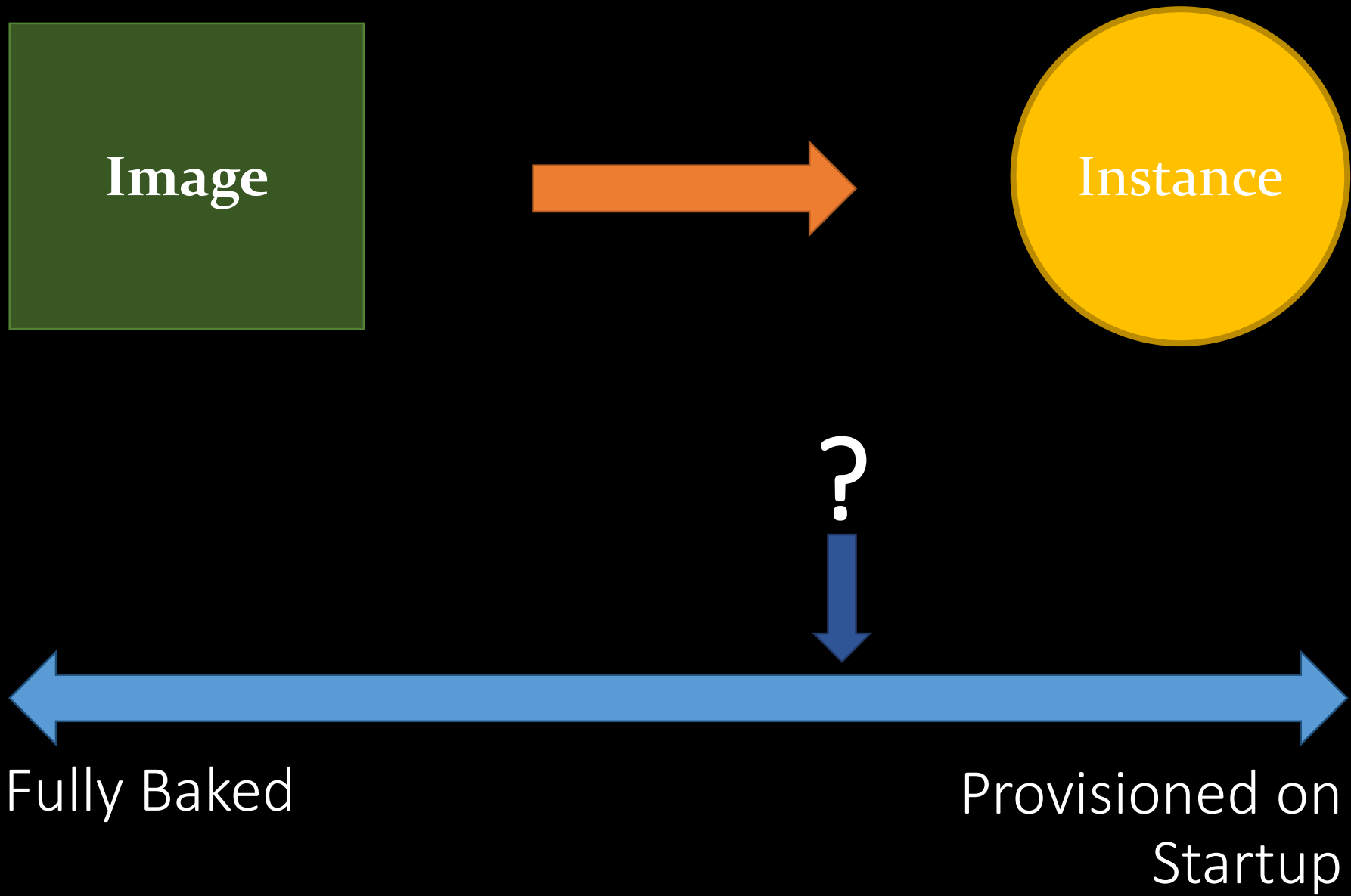




- One **immutable** unit
- **Regenerated** after every change
- **Promoted** from Environment to Environment



Classic Mistake: Build per Environment



- ✓ Every Instance 100% identical
- ✓ Fastest startup
- ✓ Launch always succeeds

NETFLIX



Fully Baked

Most people



Provisioned on
Startup



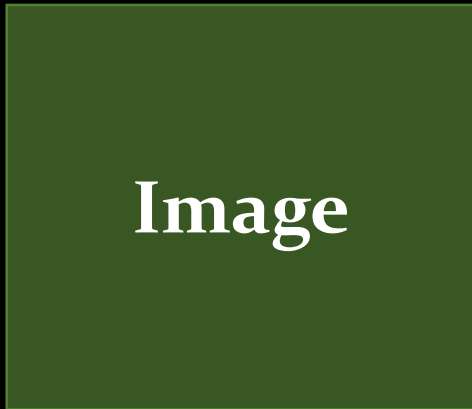
- ✓ One immutable unit
- ✓ Regenerated after every change
- ✓ Promoted from environment to environment

NETFLIX

Most people



- ✓ One immutable unit
- ✓ Regenerated after every change
- ✓ Promoted from environment to environment

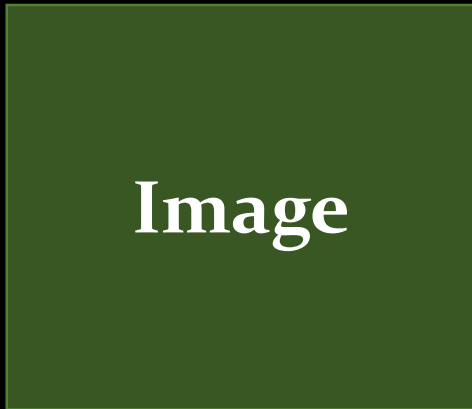


Fully Baked

- ✓ One immutable unit
- ✓ Regenerated after every change
- ✓ Promoted from environment to environment



- ✓ One immutable unit
- ✓ Regenerated after every change
- ✓ Promoted from environment to environment



Fully Baked

keep your instances **stateless**



Fully Baked



high uptime is a liability



axel@Ubuntu-1204-precise-64-minimal: ~



```
axel@Ubuntu-1204-precise-64-minimal:~$ uptime -p  
up 14 weeks, 5 days, 2 hours, 47 minutes  
axel@Ubuntu-1204-precise-64-minimal:~$
```

**The longer an instance is up,
the harder it becomes to recreate exactly
(and it will fail eventually!)**

Focus **shift**

Instance



Service

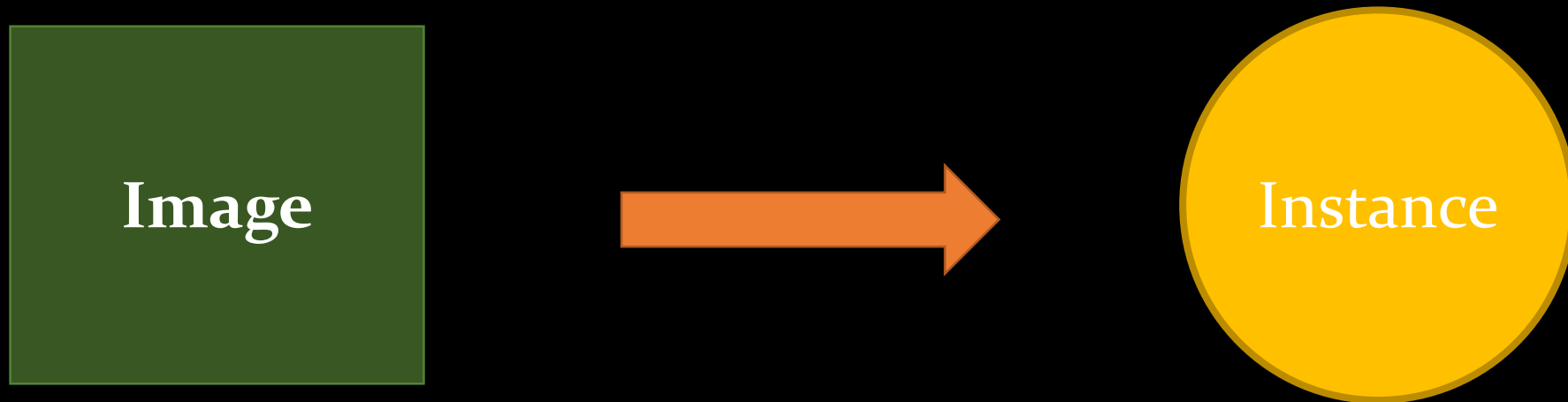
Individual instances become **disposable**

Treat servers like **cattle** instead of pets

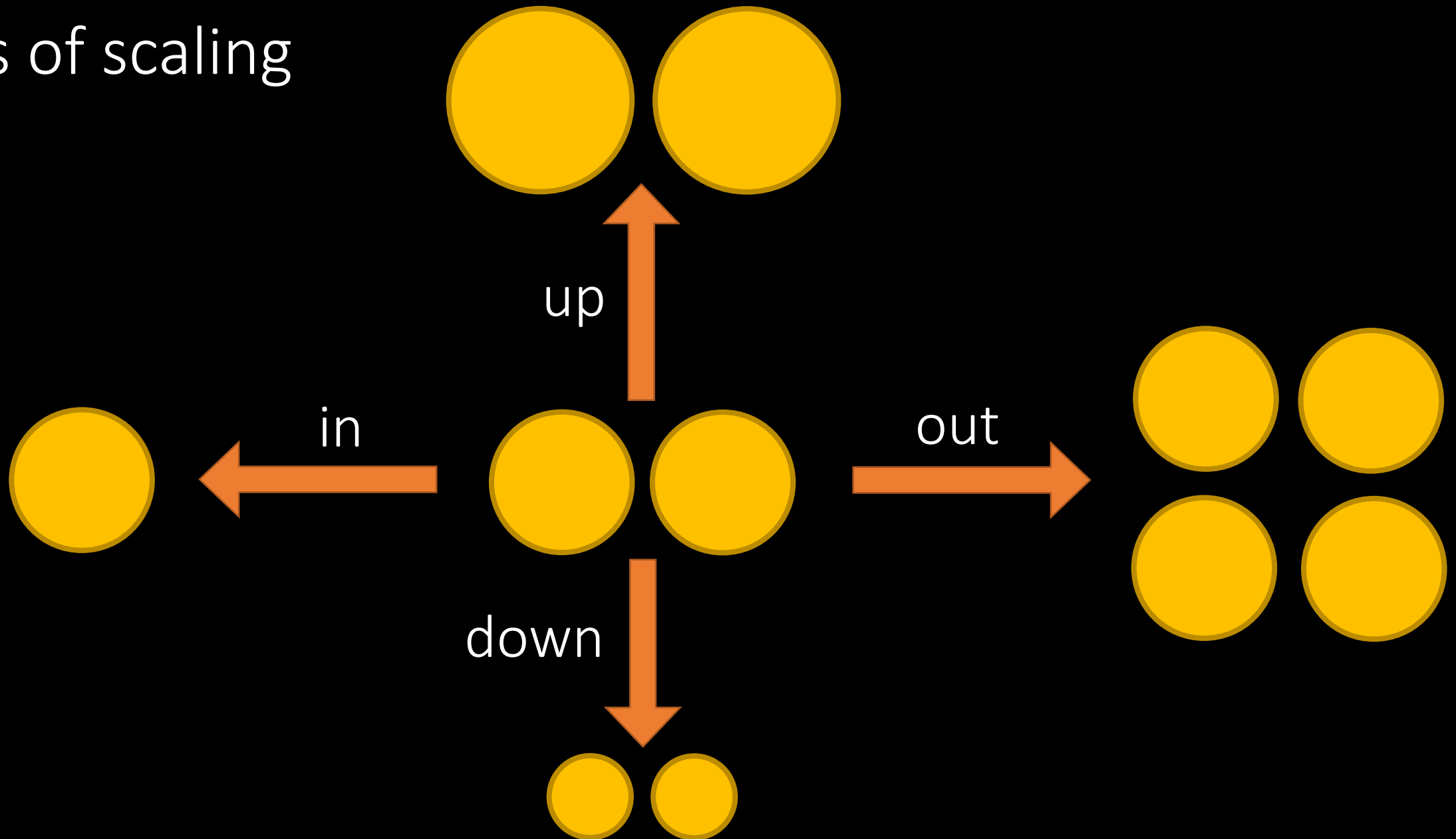


What are the implications ???

scaling



types of scaling



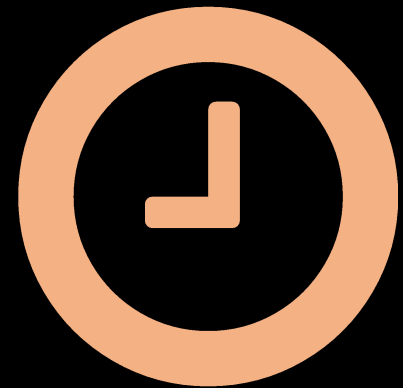
scaling triggers for different types of services



sync
=> load

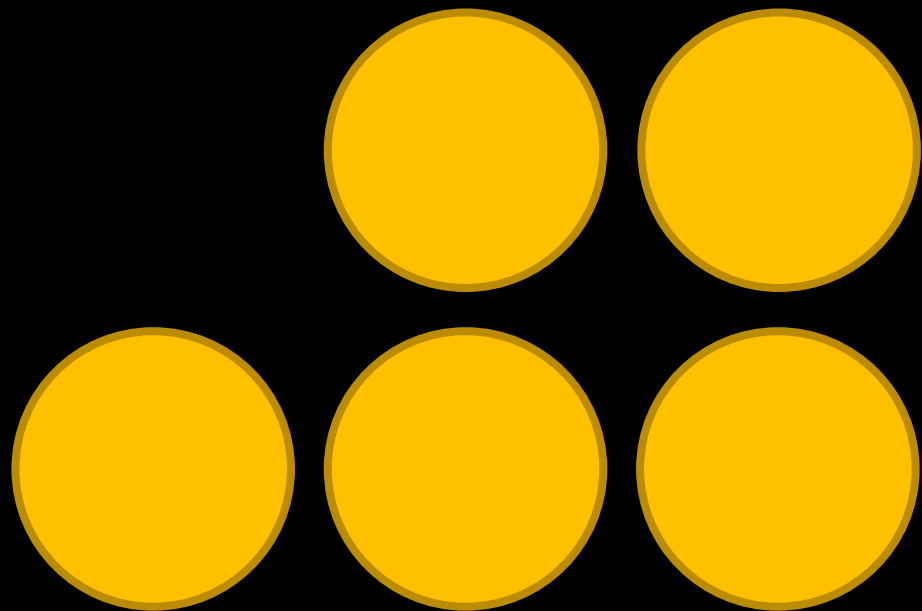


async
=> queue depth

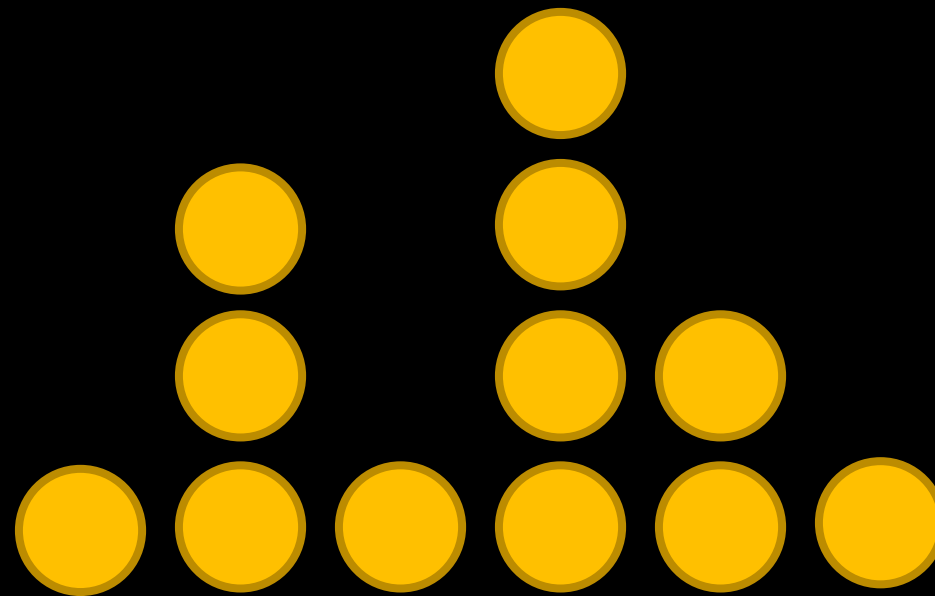


cron
=> time

scaling & costs

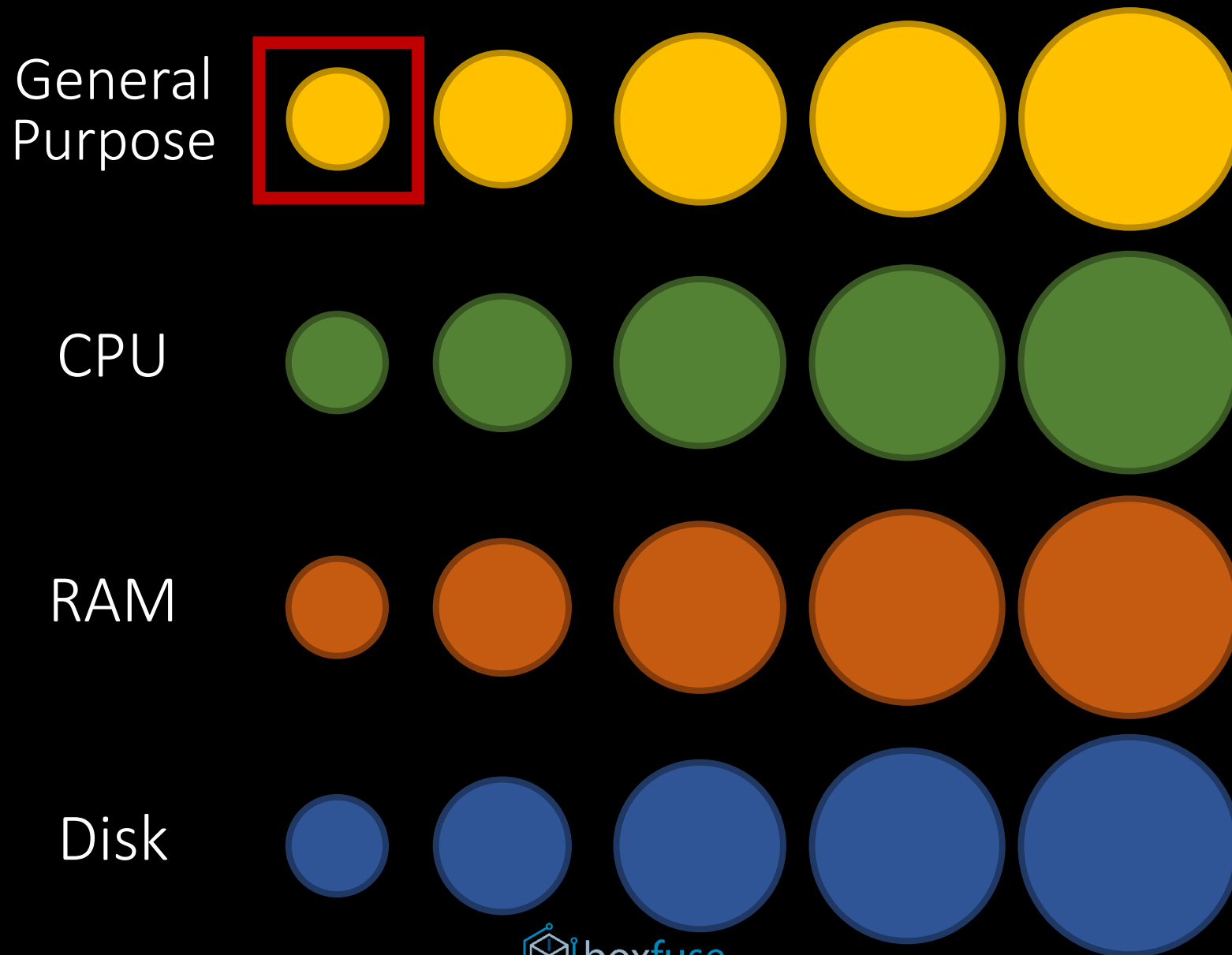


VS

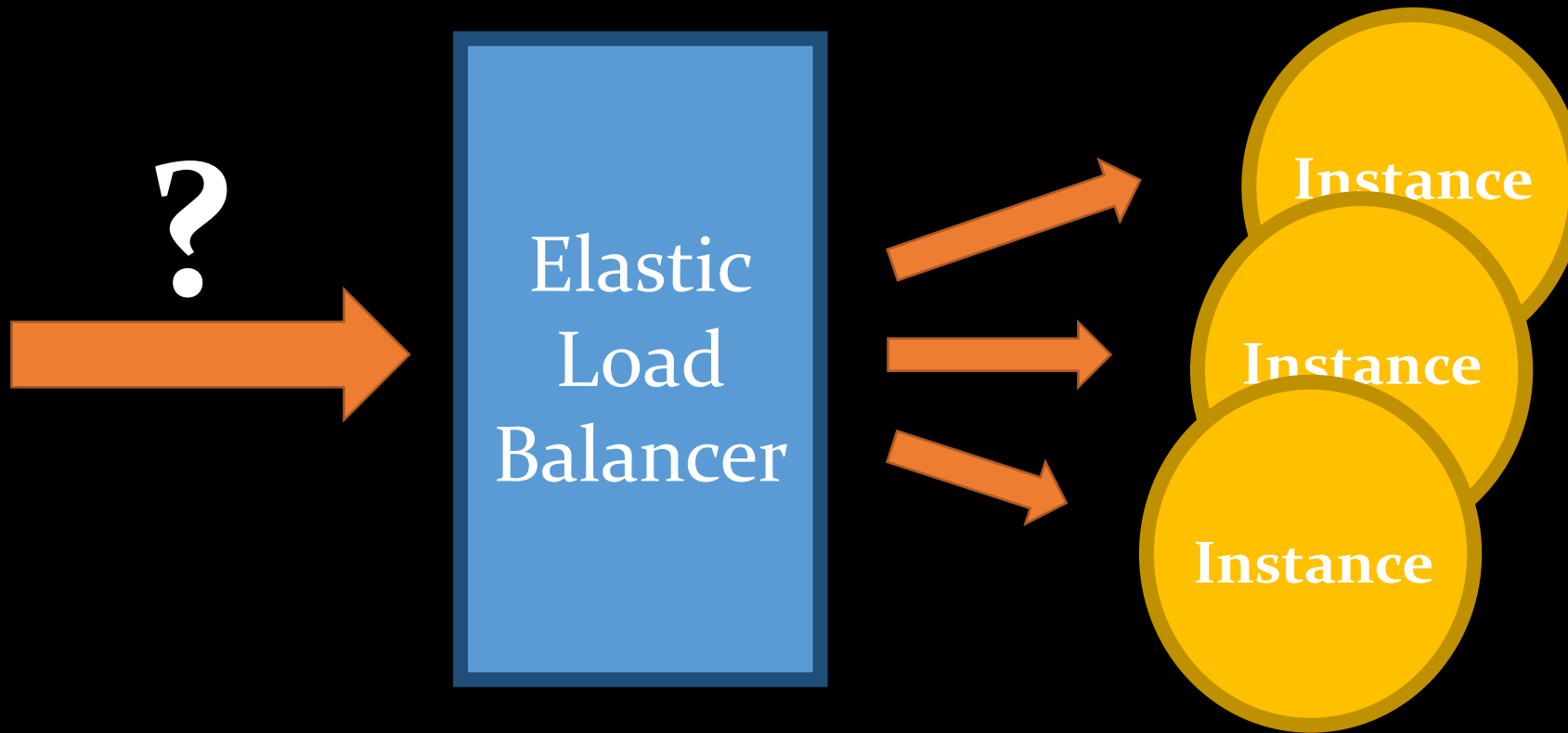


prefer smaller granularity

instance types



How to solve **service discovery** ?



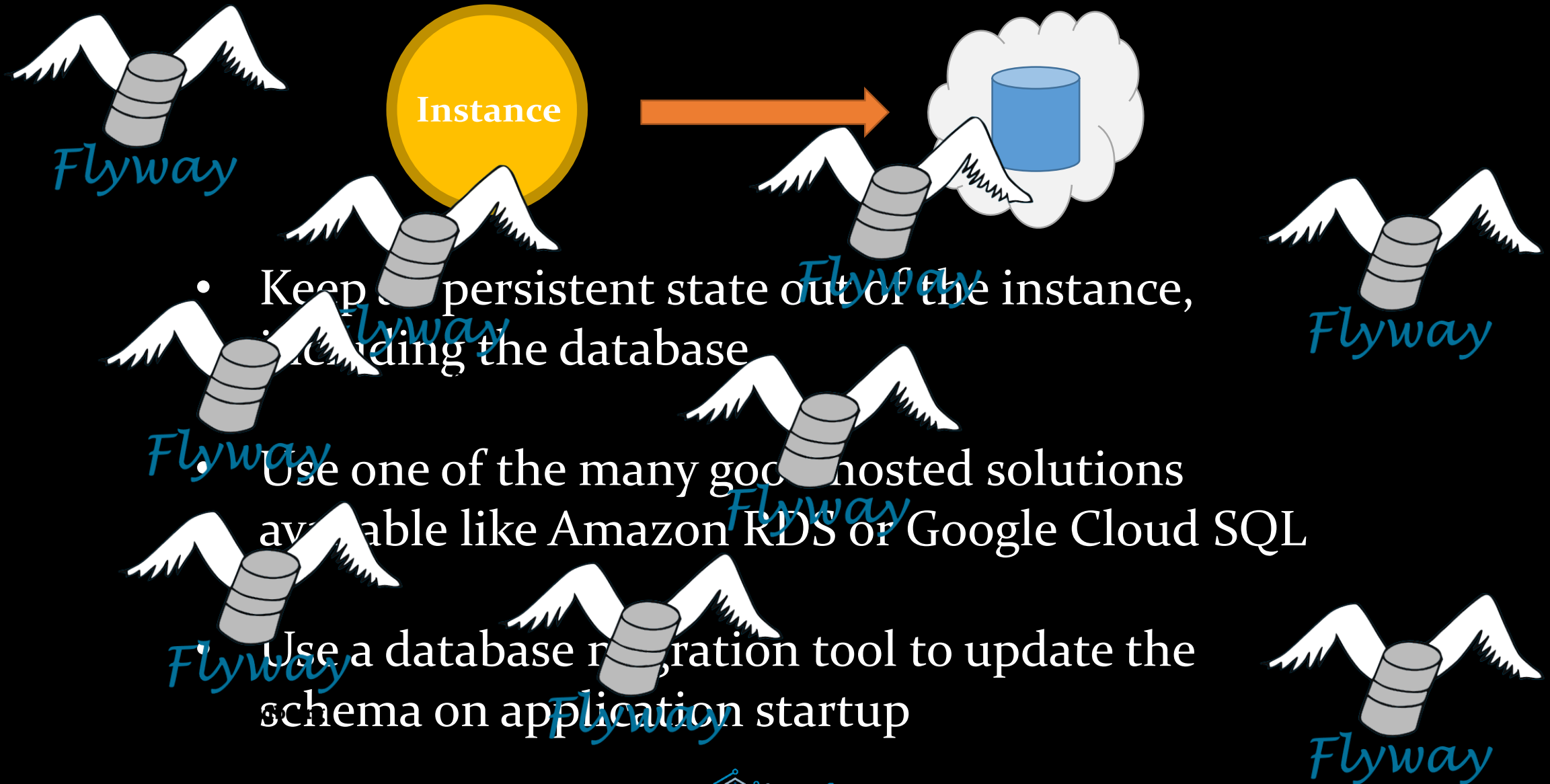
Use a stable entry point with an internal registry

what about configuration ???

- Bake as much configuration as possible for all environments directly in the Image
- Use environment detection and auto-configuration
- Pass remaining configuration at startup and expose it as environment variables

Key	Value
JDBC_URL	jdbc:...
ENV	prod

what about the database ???





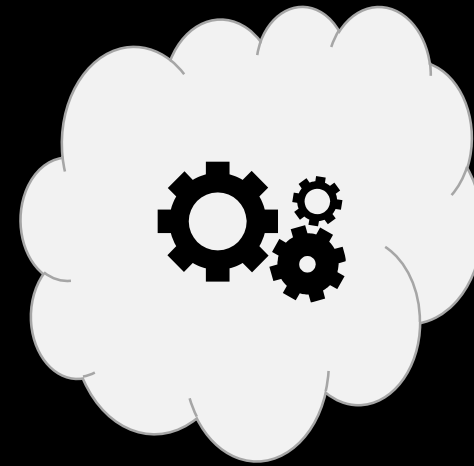
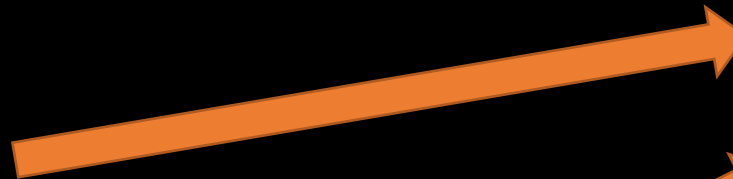
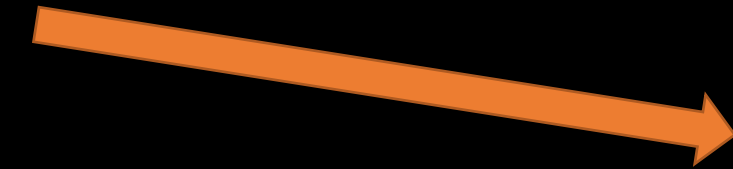
what about the logs ???
~~ssh me@myserver1~~
~~tail -f server.log~~



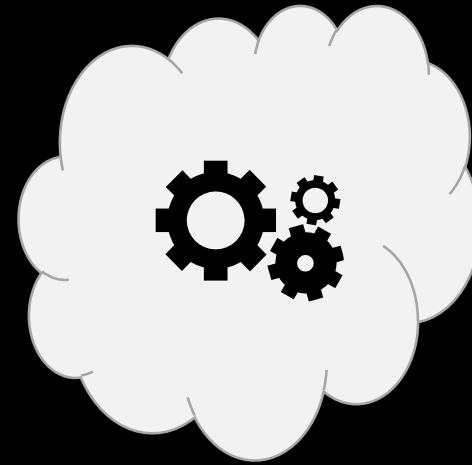
~~ssh me@myserver2~~
~~tail -f server.log~~



~~ssh me@myserver3~~
~~tail -f server.log~~



log server



Ship logs to a **central log server**

where they can be

- aggregated
- stored and backuped
- indexed
- searched through a nice web UI

Many good hosted solutions

- Loggly
- Logentries
- Papertrail
- ...

=> Think about **data privacy!**

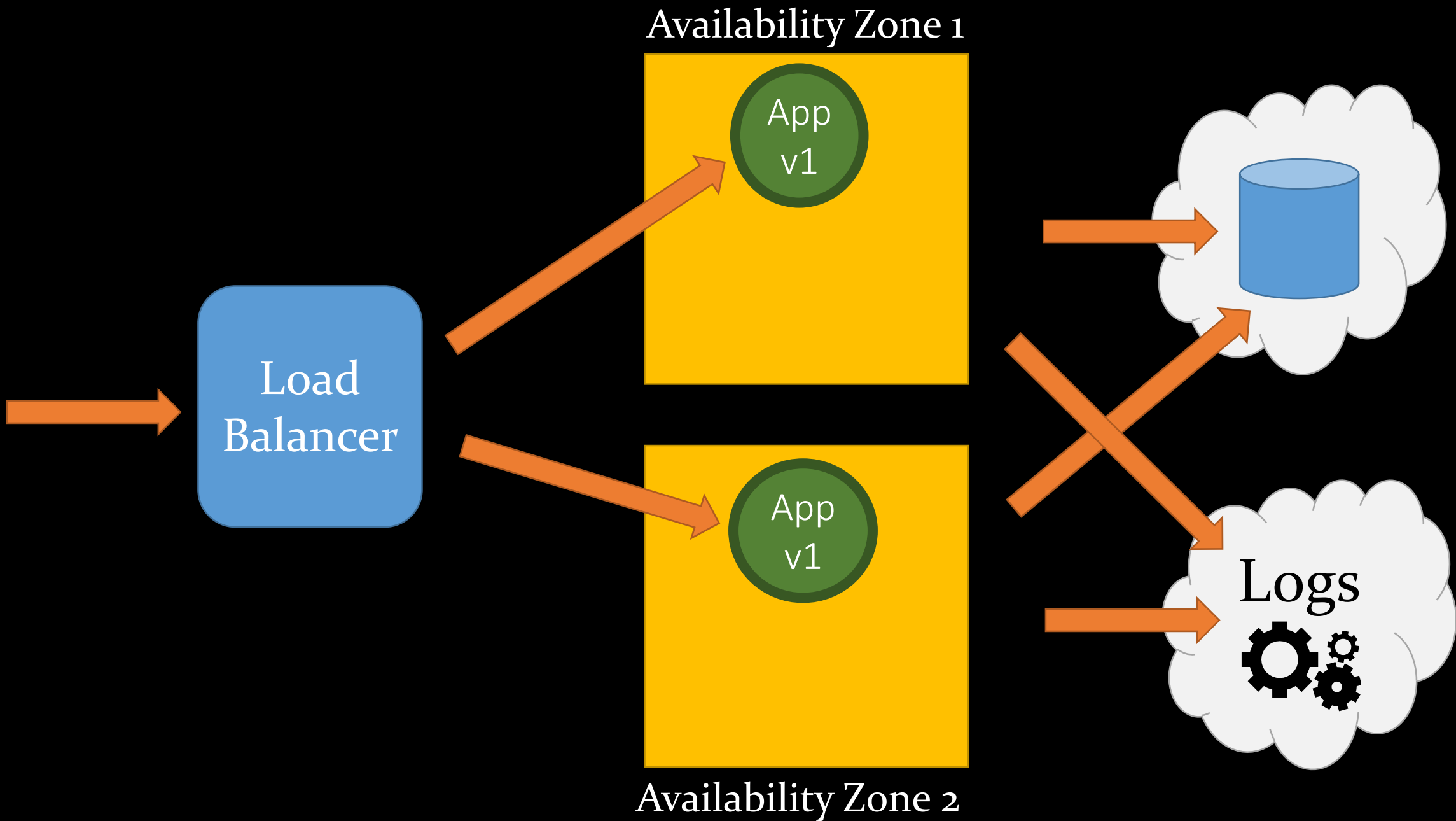
what about sessions ???

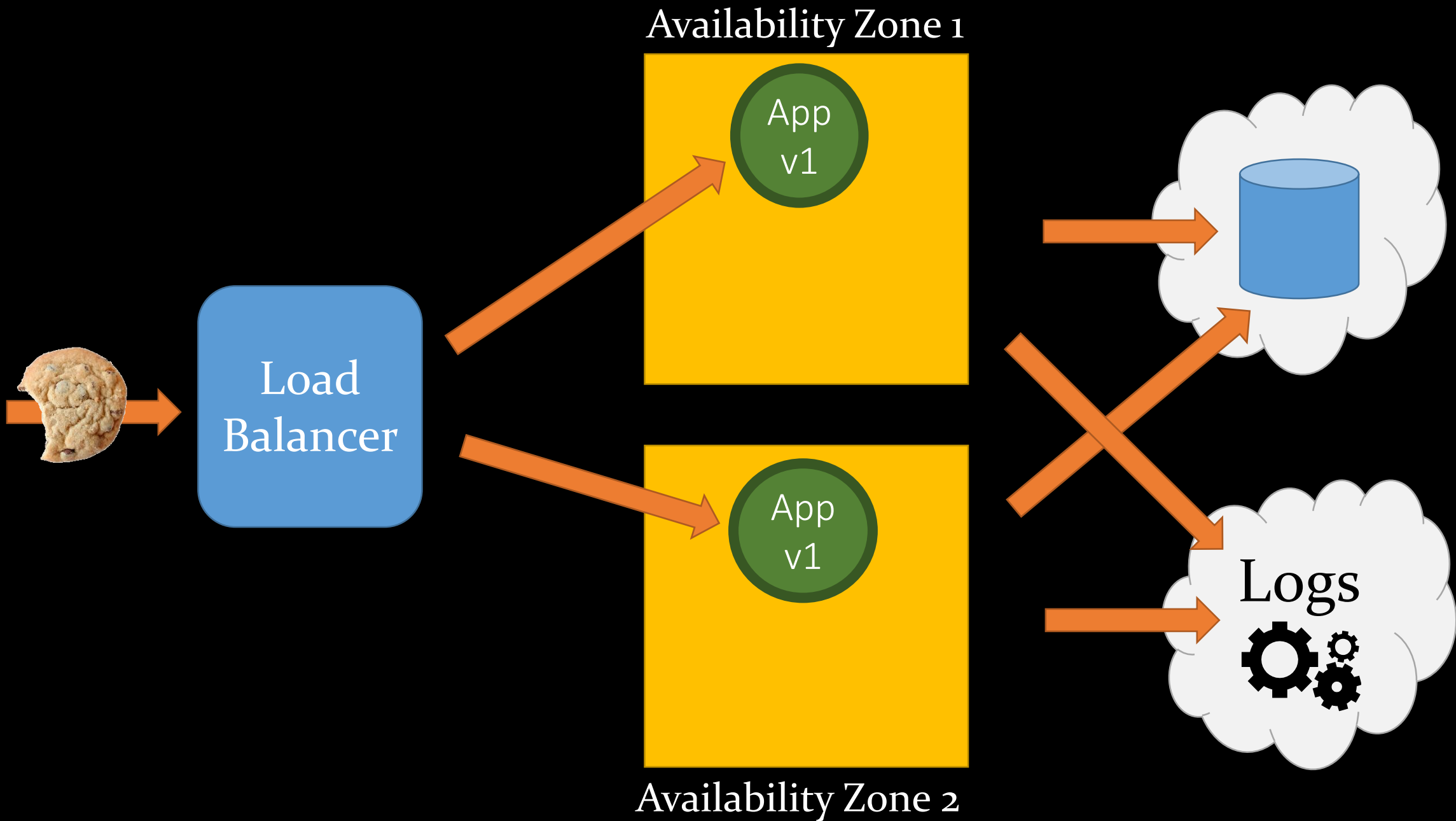


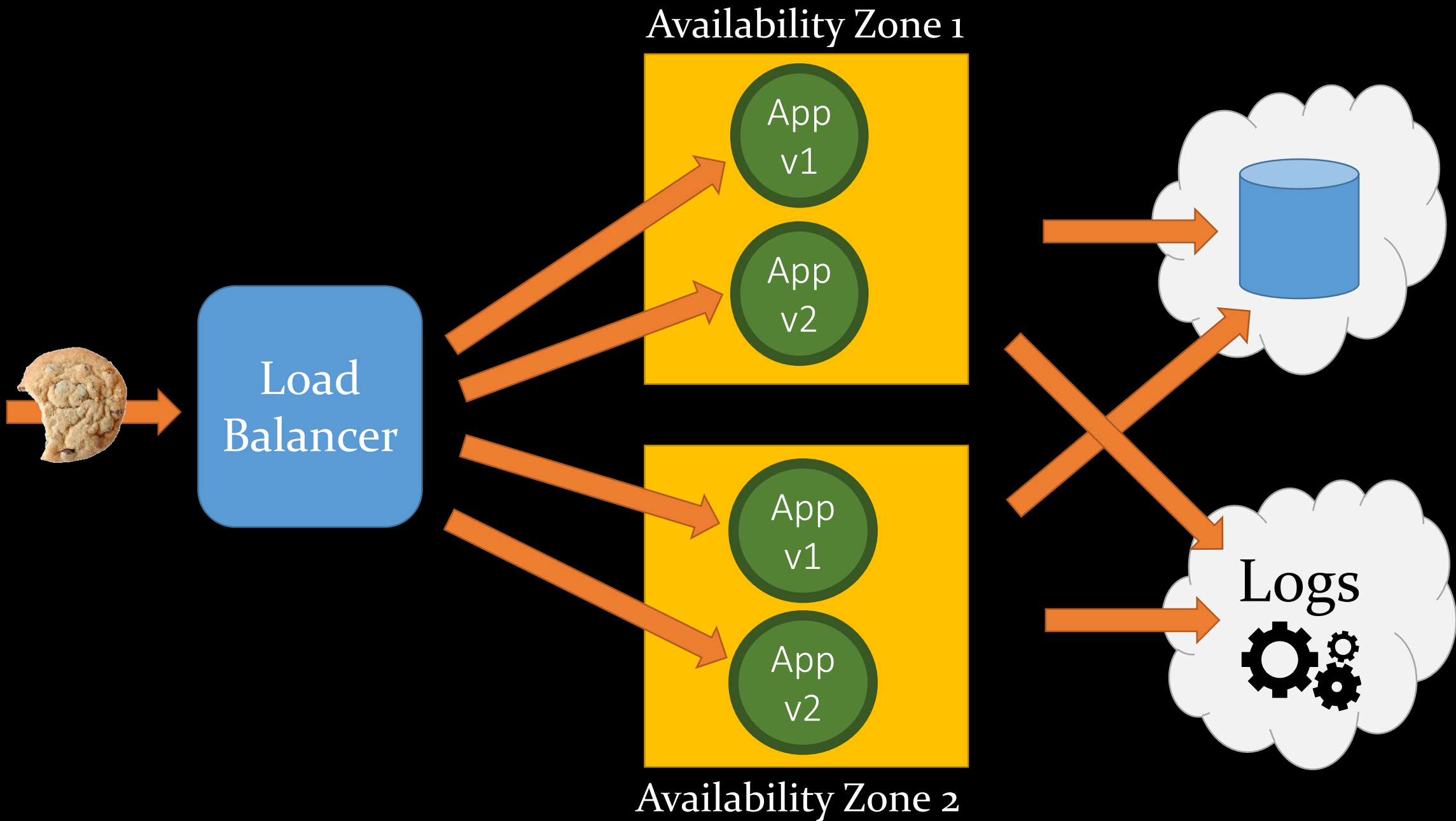
Keep session in an encrypted and signed cookie

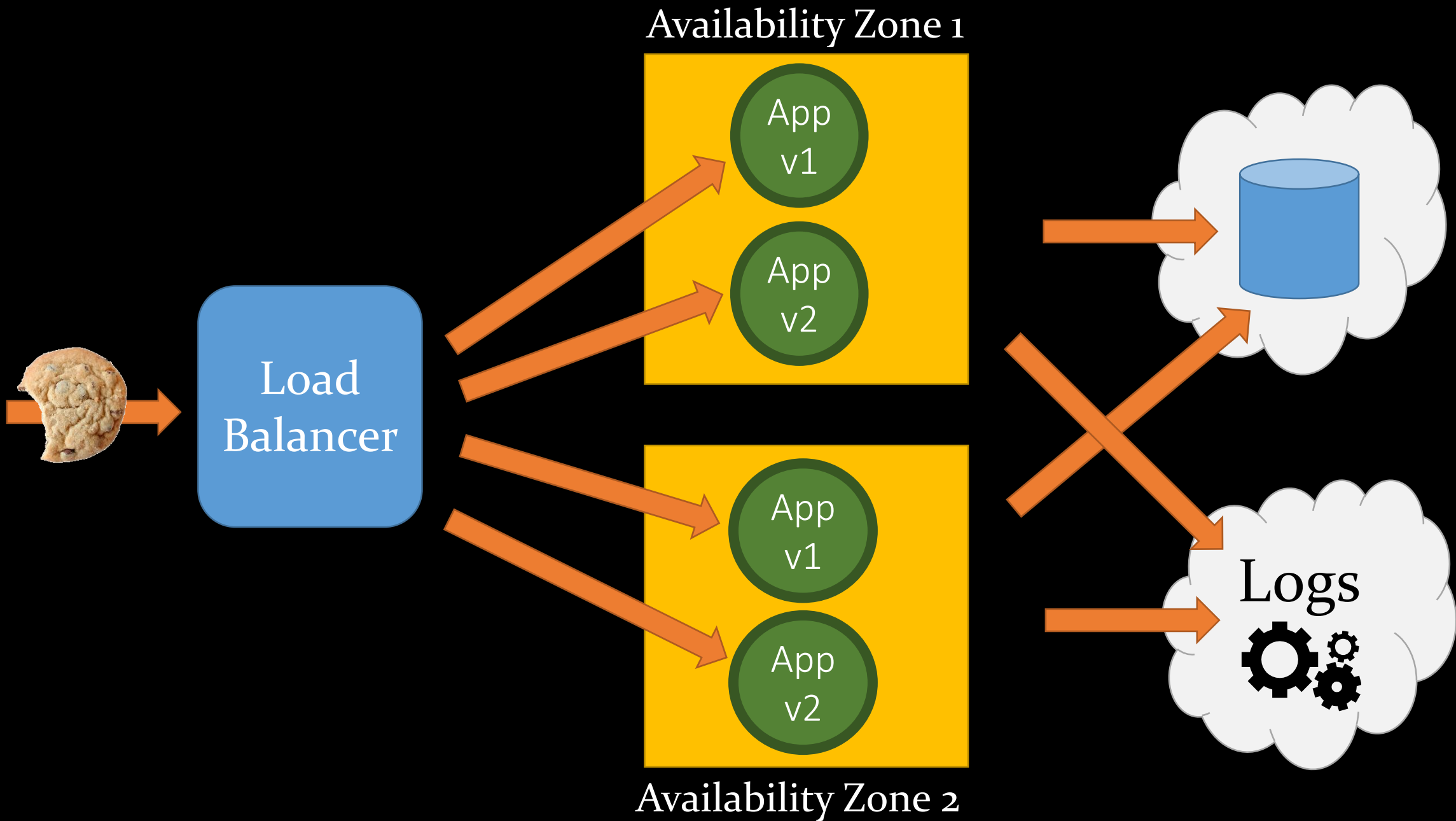
- avoids session timeouts
- avoids server clustering & session replication
- avoids sticky sessions & server affinity

what about rolling out new versions ???









what about containers ???

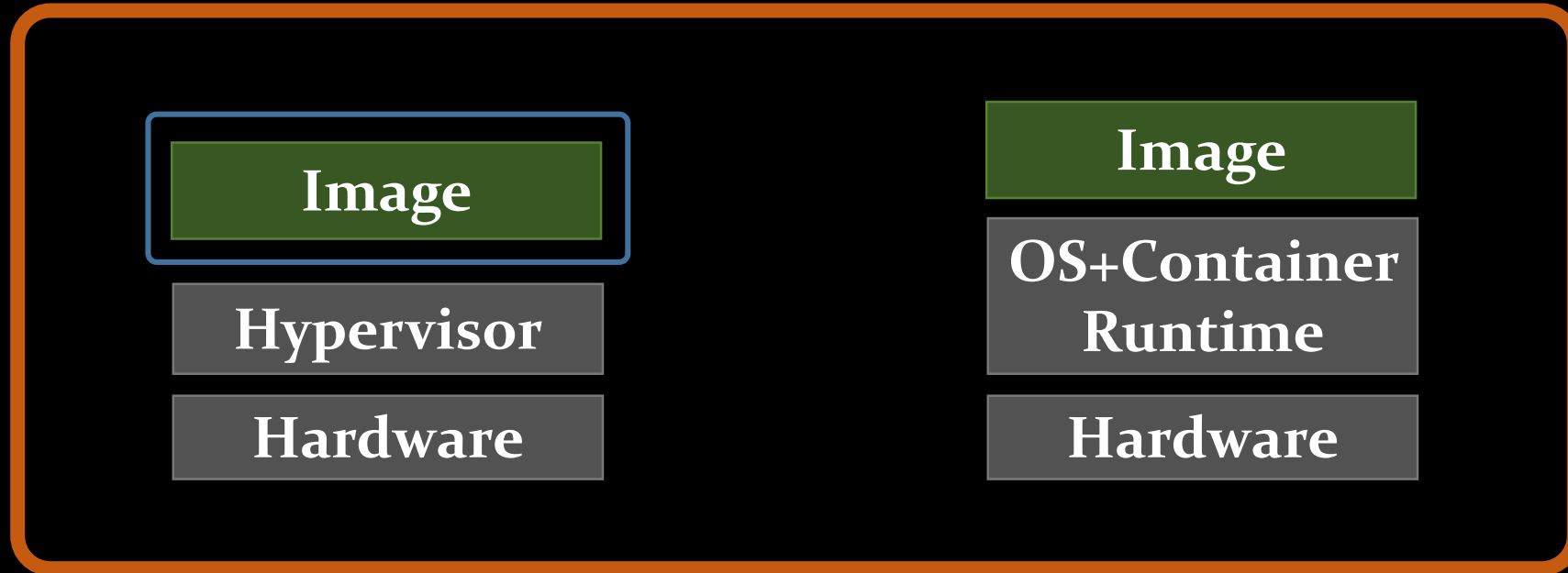
understanding modern CPUs



Both Intel and AMD have hardware support for virtualization

- isolation
- performance

on prem



VM

Container

*your
responsibility*

cloud

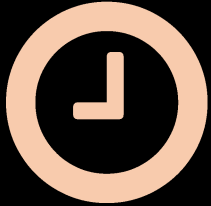
Only makes sense if
you cannot afford
8.75€/month
granularity



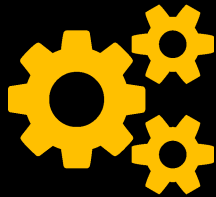
**your
responsibility**



container
images



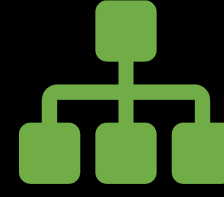
container
scheduling



containers



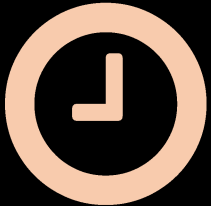
container
volumes



container
networking



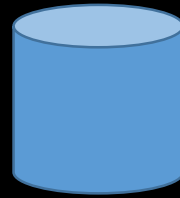
machine
images



instance
scheduling



instances



instance
volumes



instance
networking

**cloud
responsibility**

cloud

Only makes sense if
you cannot afford

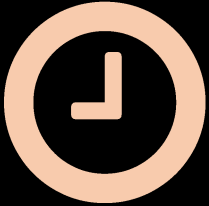
0.01€/hour
granularity



your
responsibility



container
images



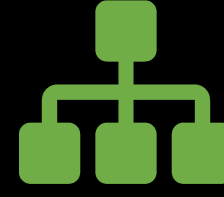
container
scheduling



containers



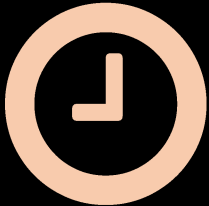
container
volumes



container
networking



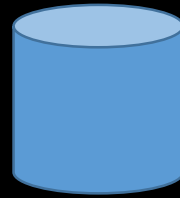
machine
images



instance
scheduling



instances



instance
volumes



instance
networking

cloud
responsibility

summary

- ✓ Put a good **lock on the door** (use encryption!)
- ✓ Use **fully baked** images (build once!)
- ✓ Treat servers like **cattle** (disposable!)



boxfuse

boxfuse.com

- Fully baked images generated in seconds (not minutes or hours)
- Optimized for JVM apps (Spring Boot, Dropwizard, Tomcat, TomEE, ...)
- Minimal images just 1% of size of regular OS (measured in MB not GB)
- Images work on VirtualBox & AWS (environment parity from dev to prod)
- Zero downtime updates on AWS (fully automatic blue/green deployments)

final disclaimer



no animals were harmed
while making this talk 😊



Please

Remember to rate session

Thank you!



Follow us on Twitter @GOTOber

www.gotober.com

 @axelfontaine

Thanks !



boxfuse.com