image credit http://eyepluscamera.files.wordpress.com/

Université Nice Sophia Antipolis

Fabien Hermenier

Cloud compting 101

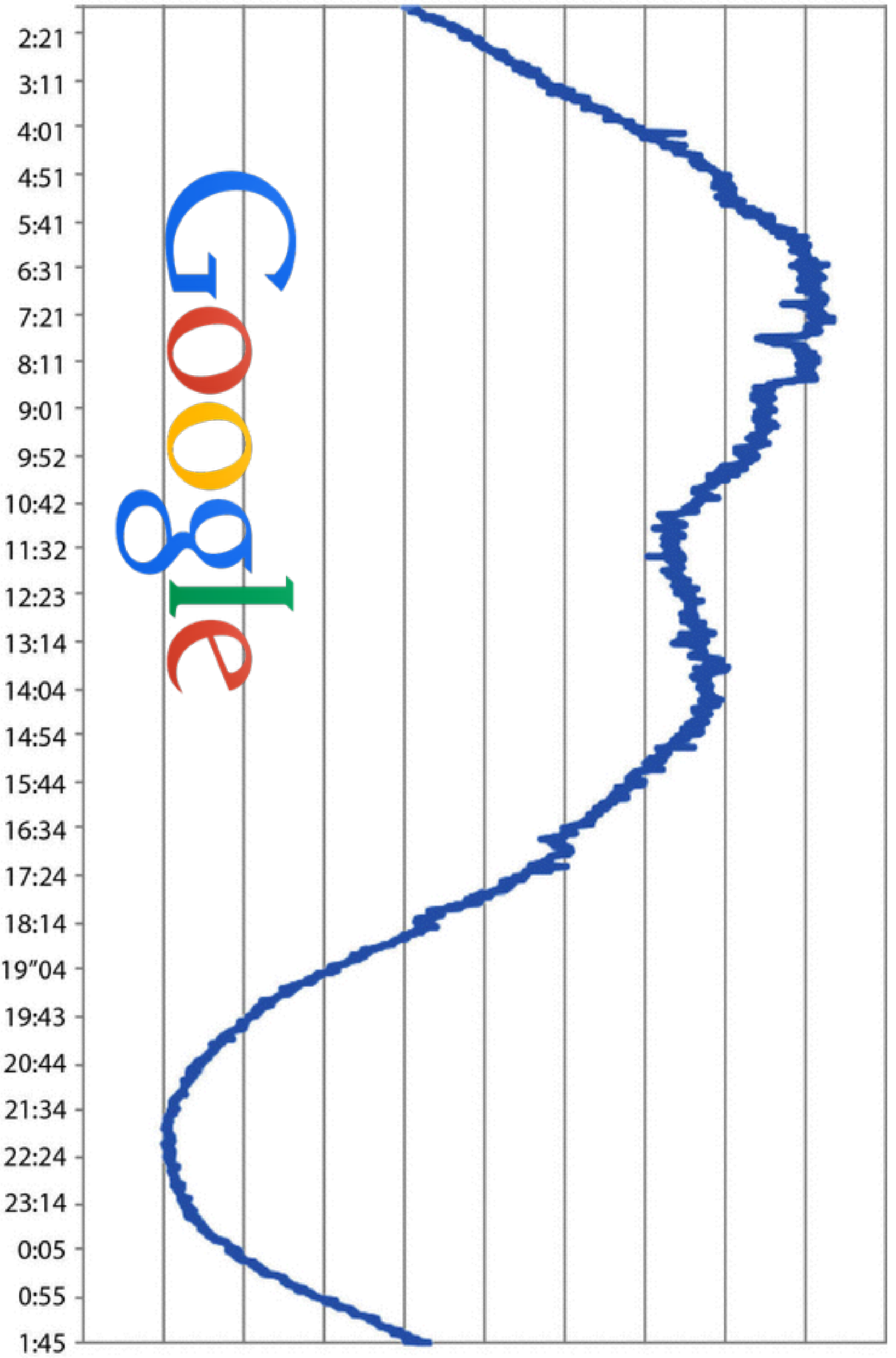
was cloud computing needed ?

Mainframes

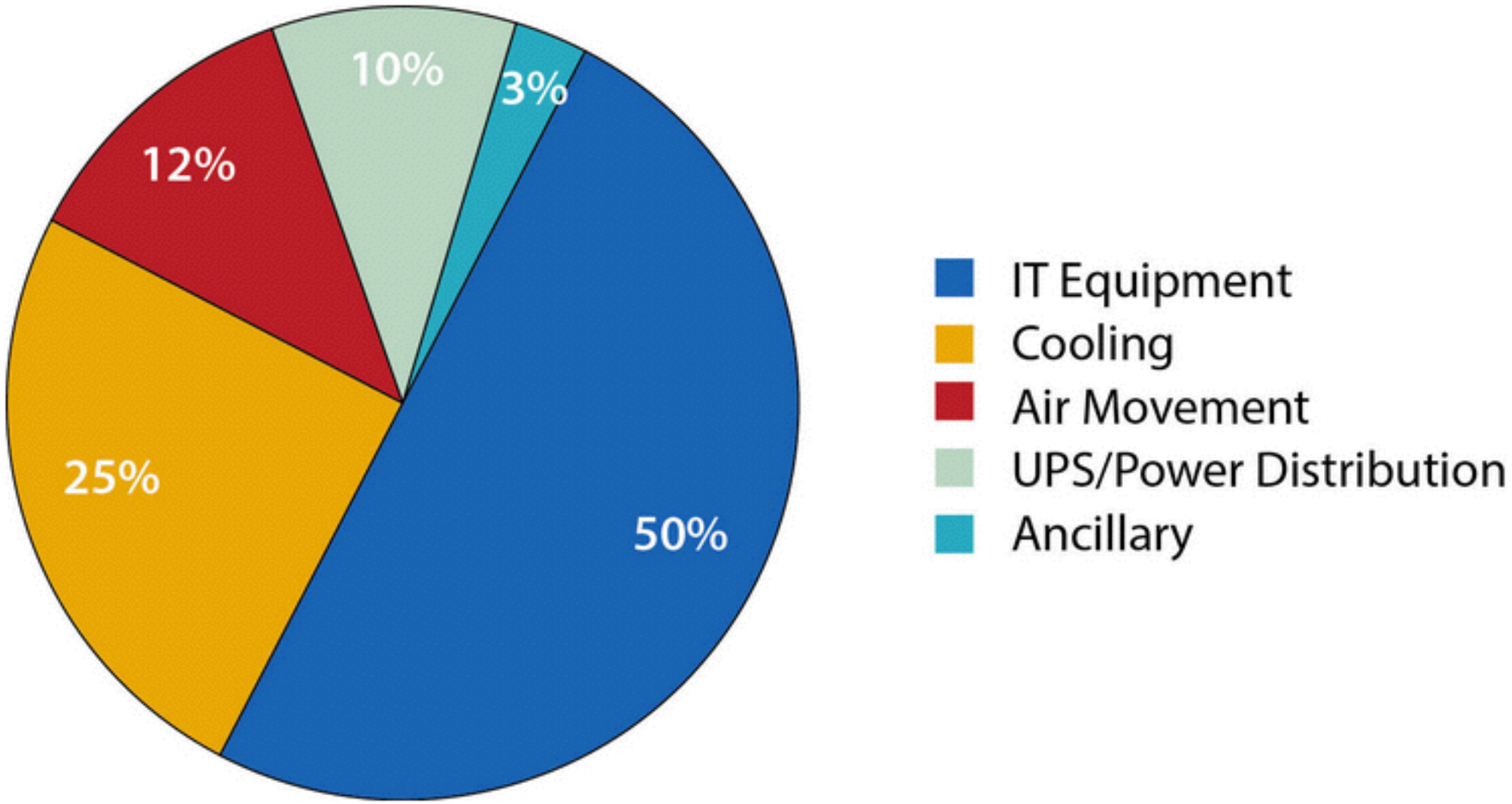
Then  came with affordable PCs

Then we spread out the load for security, performance, manageability

Then we bought tons of servers to support load spikes


2:21
3:11
4:01
4:51
5:41
6:31
7:21
8:11
9:01
9:52
10:42
11:32
12:23
13:14
14:04
14:54
15:44
16:34
17:24
18:14
19"04
19:43
20:44
21:34
22:24
23:14
0:05
0:55
1:45
Google

Amazon X-mas 2013
426 items sold each second



Where is energy spent ?

episode 0
rise of the cloud

National Institute of Standards and Technology
U.S. Department of Commerce

2011

> “Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction.”

1

On-demand self-services

self-provisioning,
no human intervention

Software Infrastructure

Application Development

Application Servers

Application Stacks

Big Data

Databases & Caching

Network Infrastructure

Operating Systems

Security

Developer Tools

Issue & Bug Tracking

Monitoring

Source Control

Testing

Business Software

Business Intelligence

Financial Services

Collaboration

Content Management

CRM

eCommerce

Education & Research

High Performance

Featured Products

Postgres Plus Cloud Database Advanced
EnterpriseDB Corporation
$0.26 to $20.24/hr for software

PHP 5.5 - Zend Server Developer Editi...
Zend Technologies USA,...
$0.03/hr for software
Free Trial

GoldDisk Plus - DISA STIG Windows 200...
SteelCloud
$49.00/mo + $0.00 to $0.06/hr for software + Charges for EC2 with Windows

Popular Products

Operating Systems

Amazon Linux AMI (HVM / 64-bit)
Amazon Web Services
$0.013 to $8.14/hr incl EC2 charges

CentOS 7 (x86_64) with Updates HVM
Centos.org
$0.00/hr for software

Ubuntu Server 14.04 LTS (HVM)
Canonical Group L...
$0.00/hr for software

2

broad network access

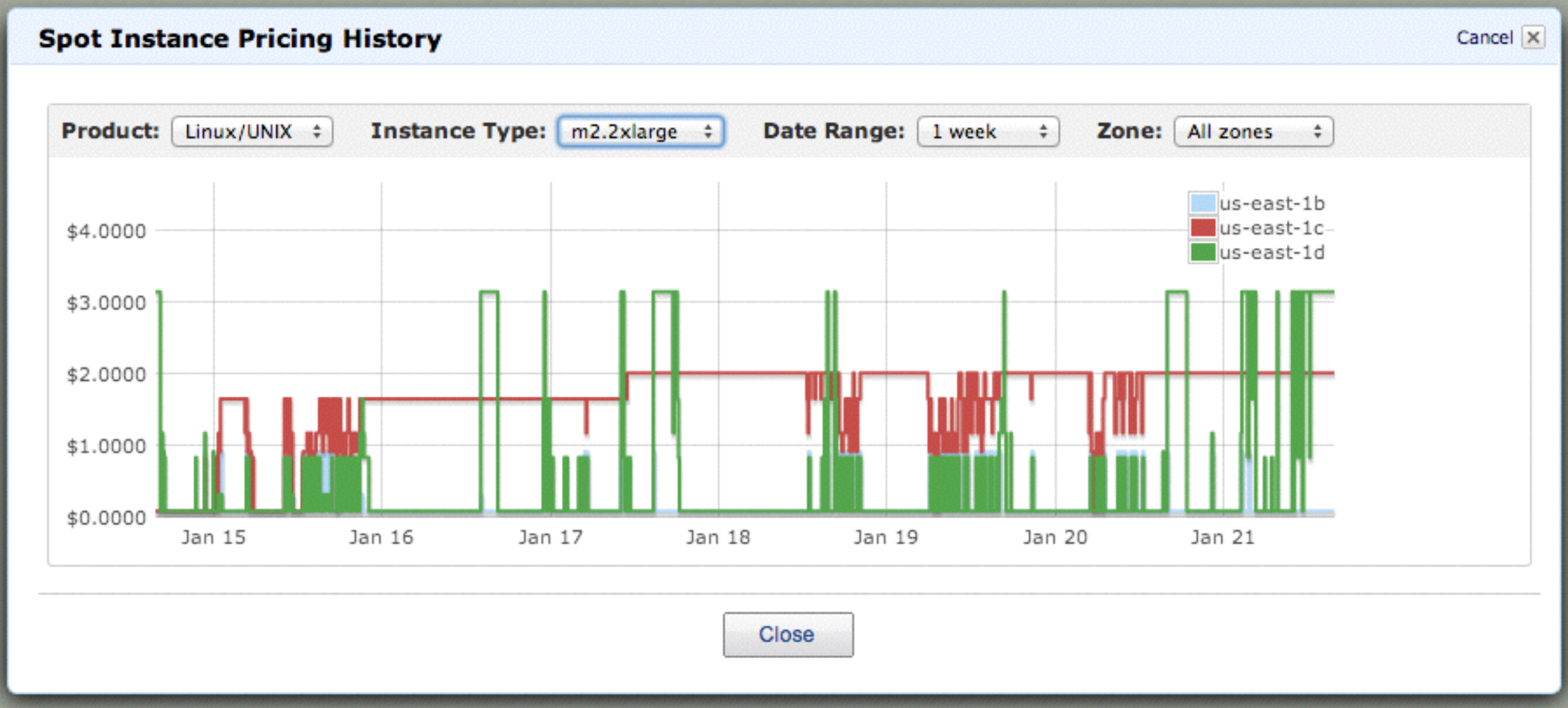
availability over the network
standard mechanisms

3 resource pooling

multi-tenant
virtual or physical resources
on-demand allocation
location independance

reserved instances (yearly based)
on-demand instances (hourly based)
hotspot instances (market based)

Amazon EC2 HotSpot instances



bid over the market price to get the instance

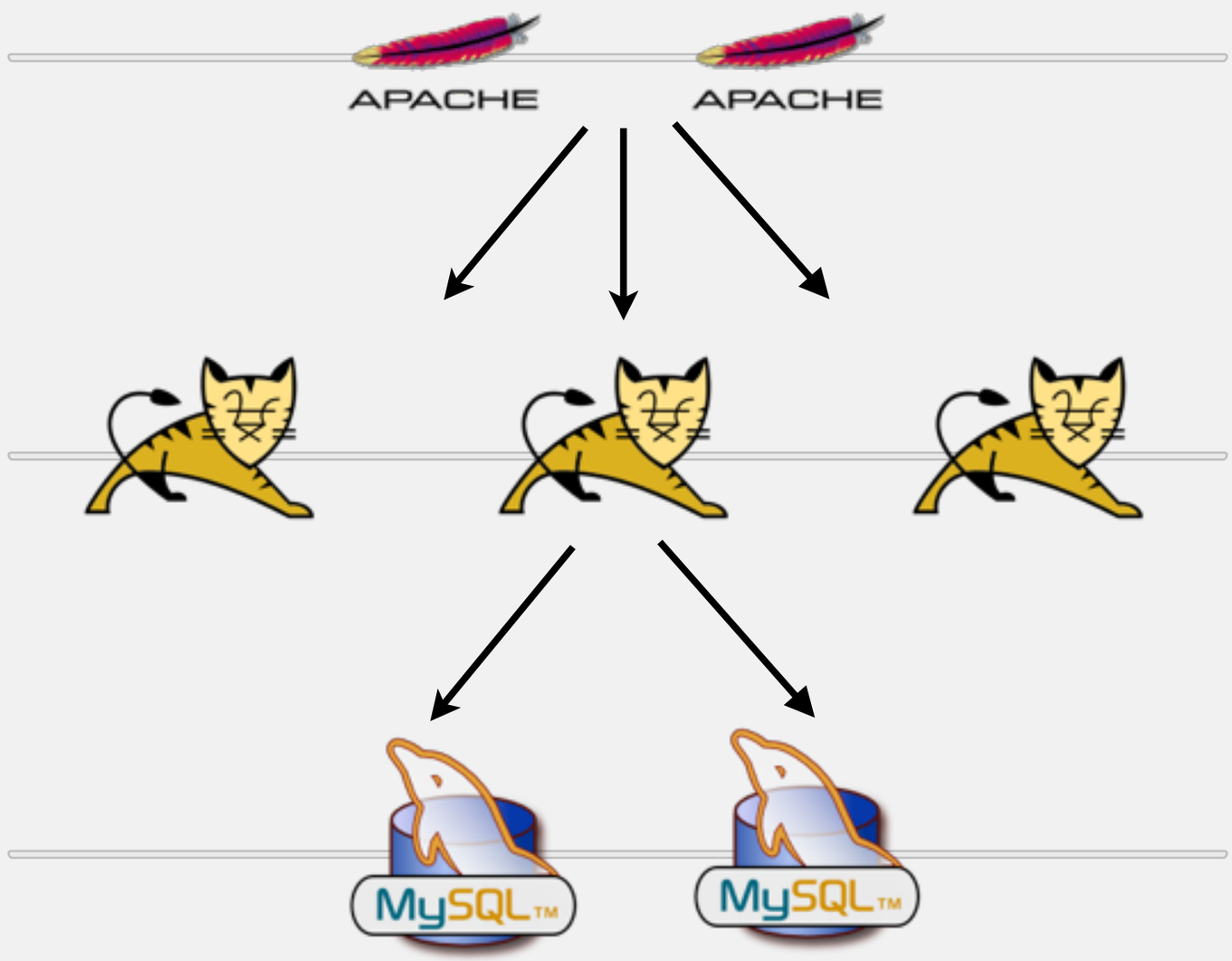
rapid elasticity

fast (de-)allocation of resources
scale to infinity

4

vertical elasticity

Tiers 1

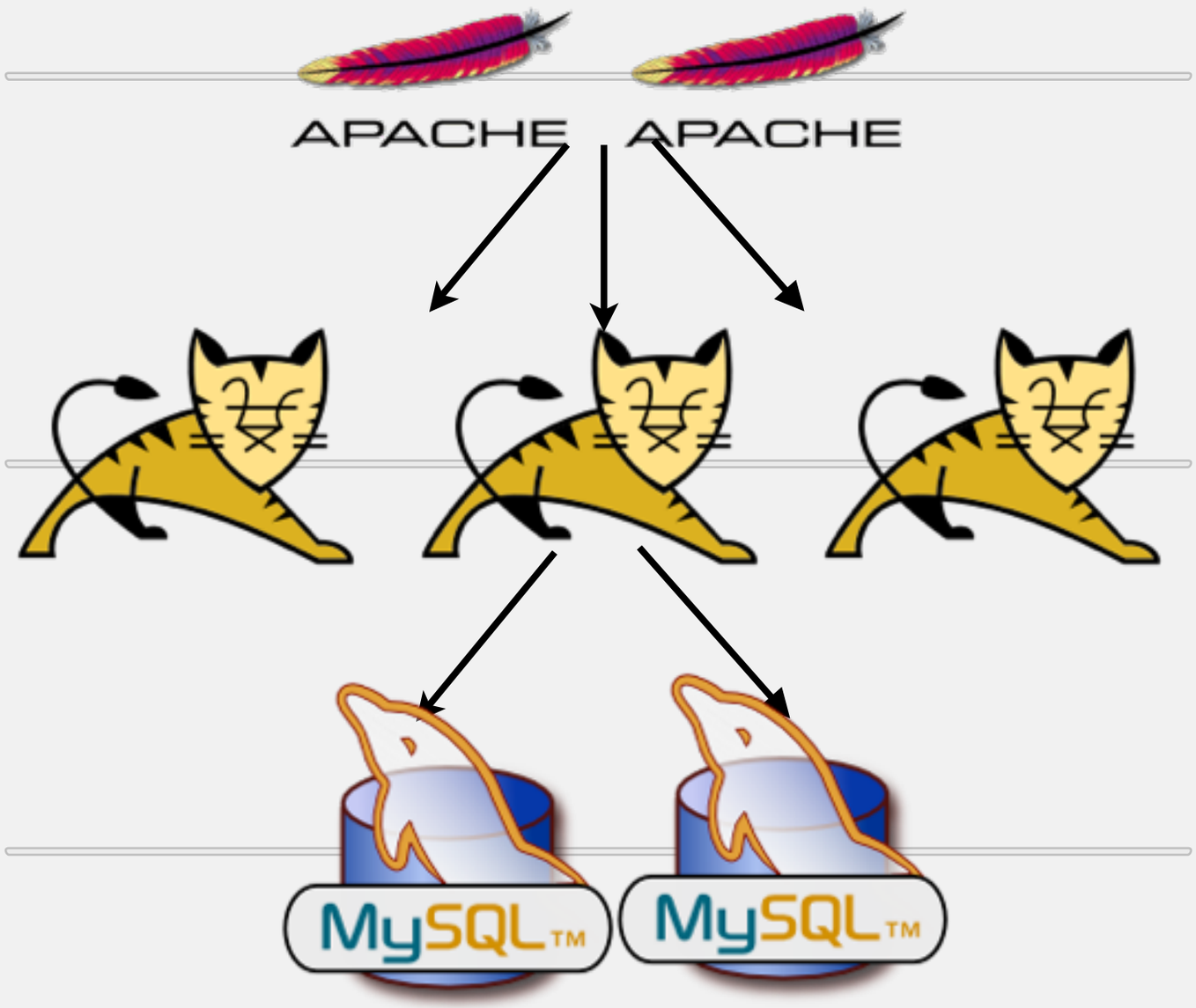


Tiers 2

Tiers 3

vertical elasticity

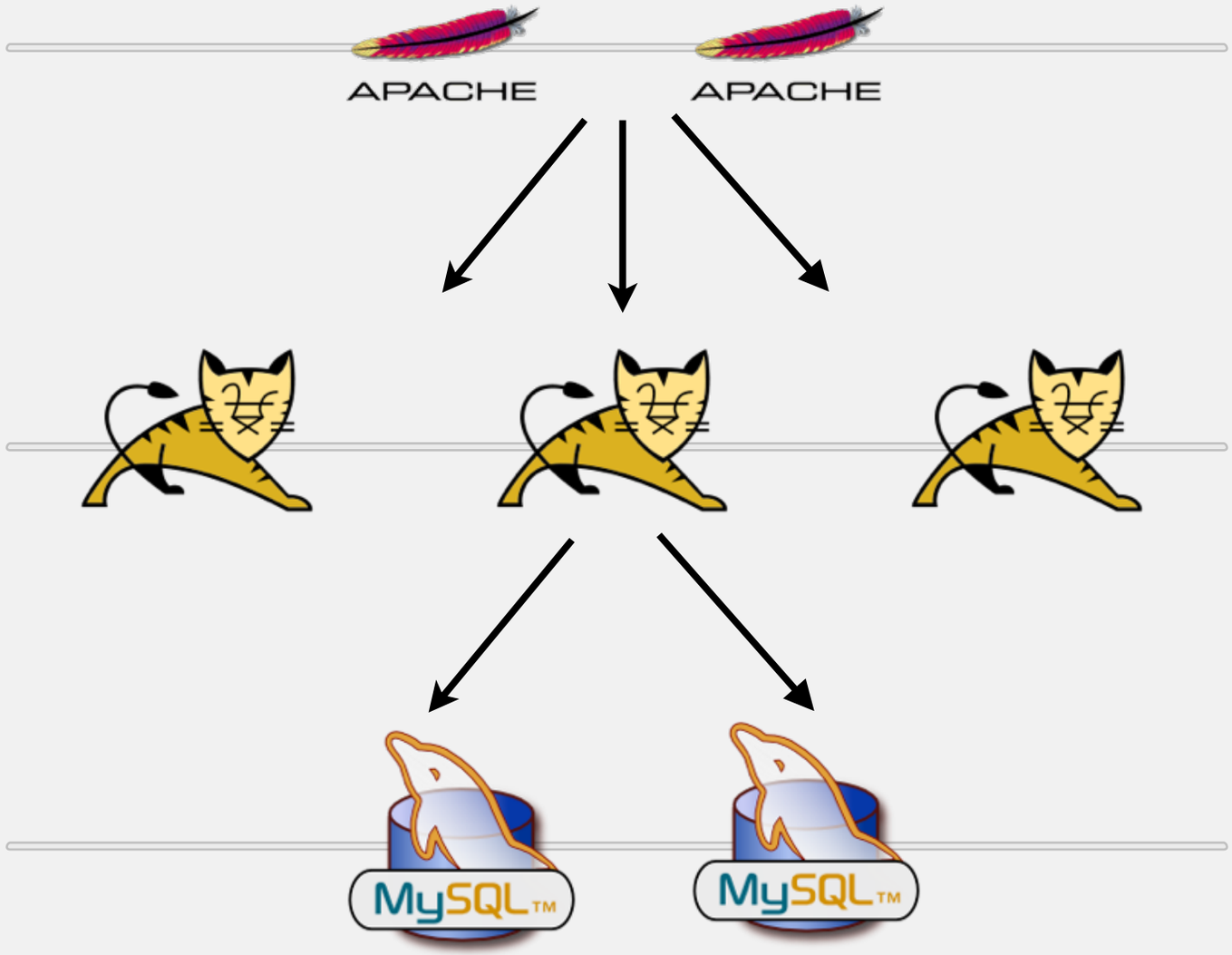
Tiers 1



Tiers 2

Tiers 3

Tiers 1

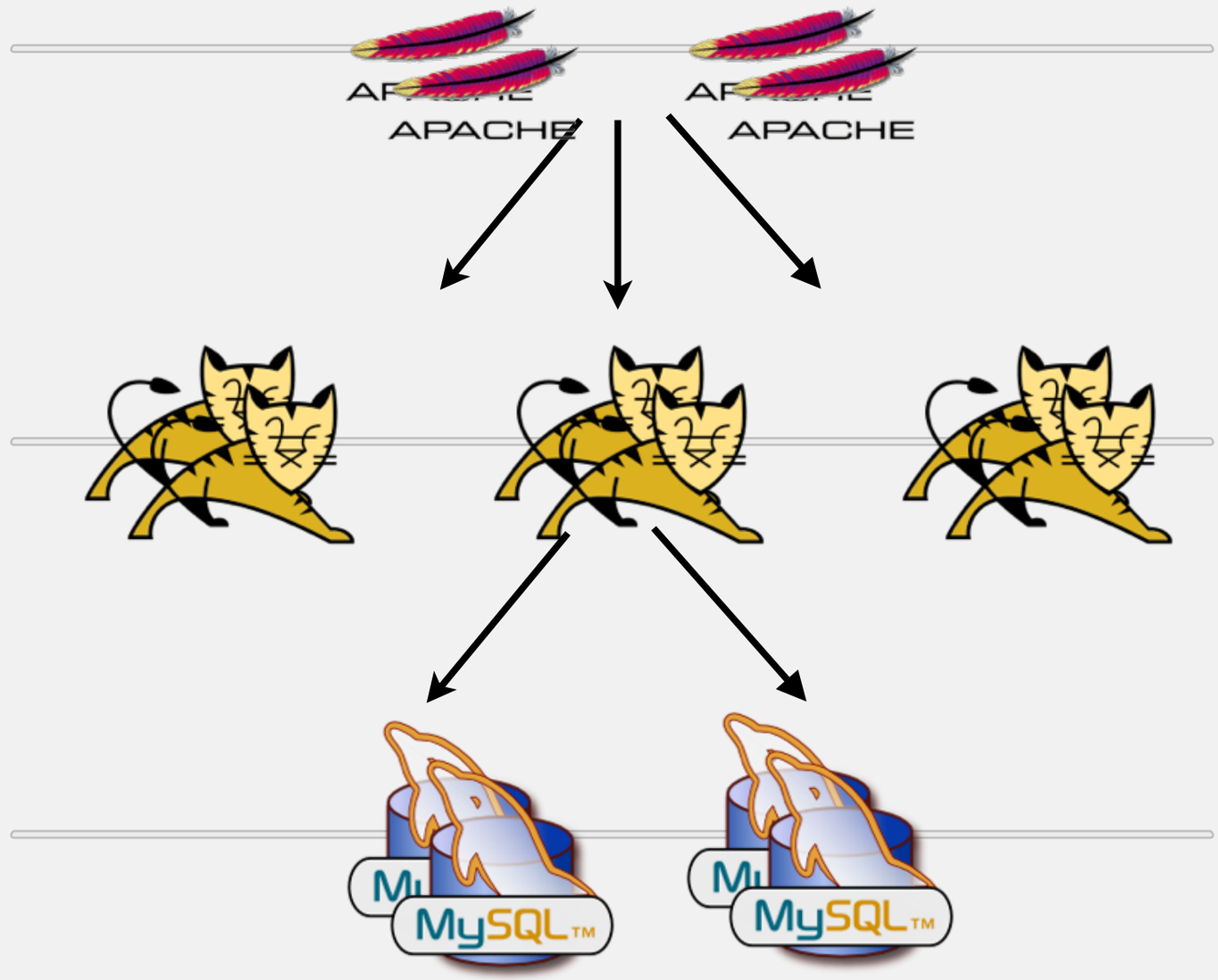


Tiers 2

Tiers 3

horizontal elasticity

Tiers 1



Tiers 2

Tiers 3

horizontal elasticity

5 measured service

metering capabilities
transparent reporting

RunAbove

Hourly | Monthly

Cloud Sandbox

$ 0.004	$ 0.010
M	L
1 Core	1 Core
2 GB RAM	4 GB RAM

LABS | Get 1 month Free!

Steadfast Resources

$ 0.014	$ 0.028	$ 0.056
S	M	L
1 Core	1 Core	2 Cores
2 GB RAM	4 GB RAM	8 GB RAM

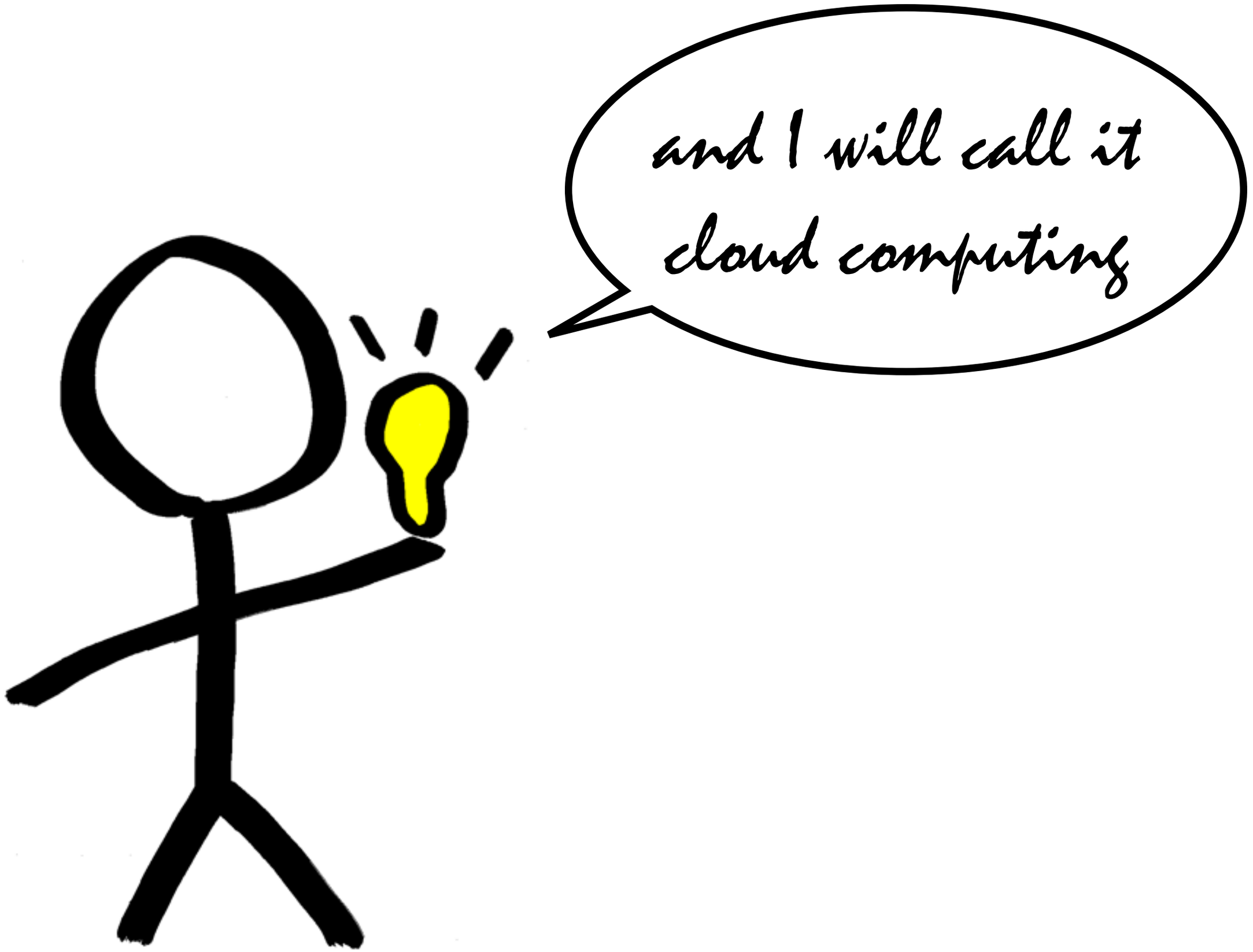
Launch Now !

1 VM/HOST

$ 0.110	$ 0.140
XL 3	XL 4
4 Cores	6 Cores
16 GB RAM	24 GB RAM

Launch Now !

More Info


and I will call it
cloud computing

Cloud Computing

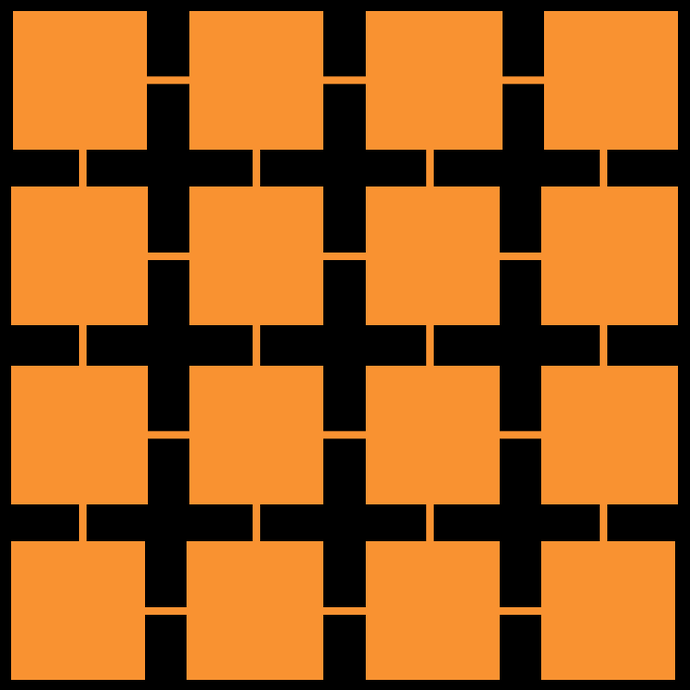
origins

“

If computers of the kind I have advocated become the computers of the future, then computing may someday be organized as a public utility just as the telephone system is a public utility... The **computer utility** could become the basis of a new and important industry.

”

John McCarthy, 1961



cluster computing

80s

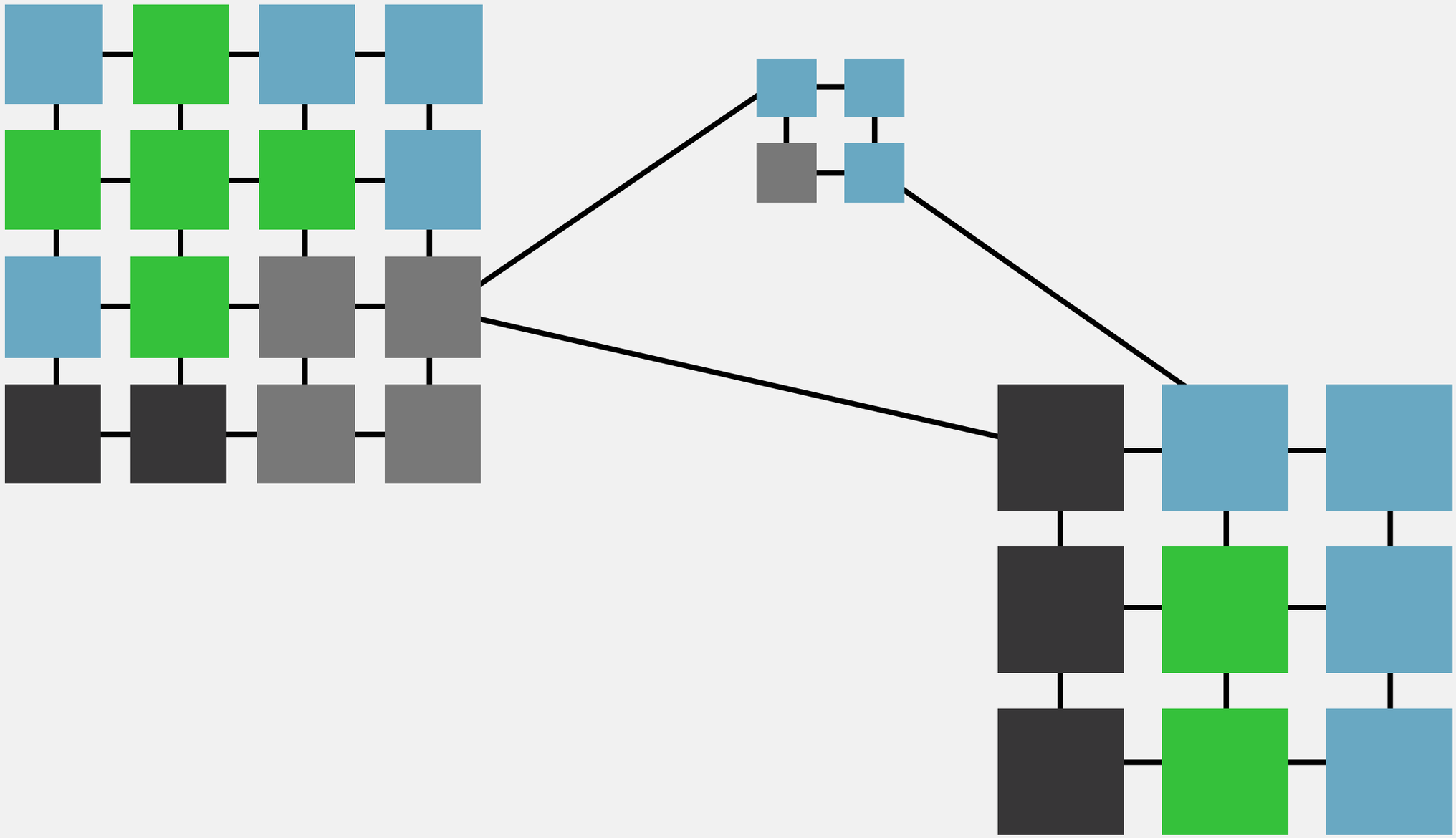
loosely coupled co-located servers
single tenant
non-interactive workload
rigid jobs

Cloud or not ?

- ☐ on demand self-services
- ☐ broad network access
- ☐ resource pooling
- ☐ rapid elasticity
- ☐ measured service

Cloud or not ?

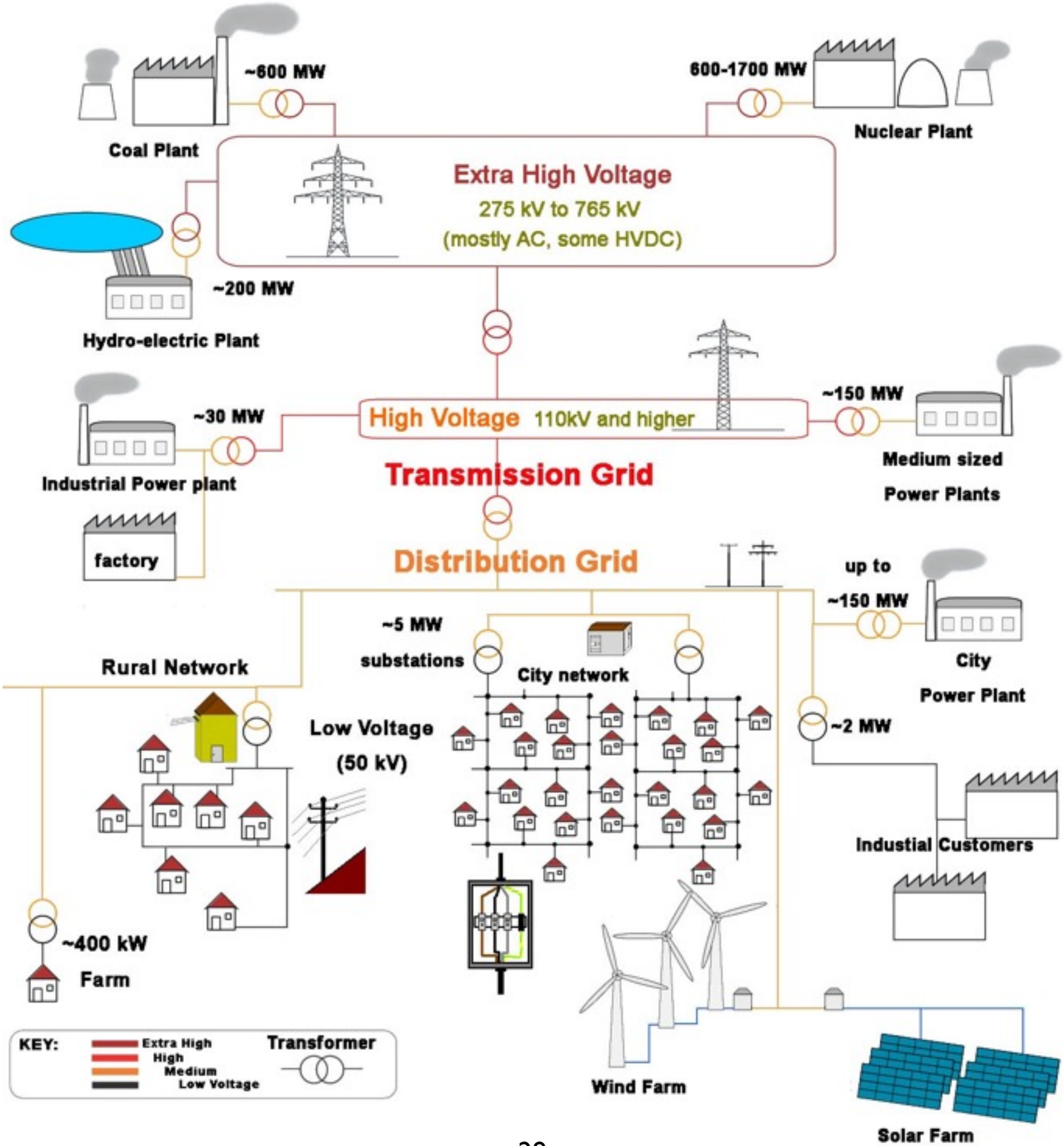
- [ ] on demand self-services
- [x] broad network access
- [ ] resource pooling
- [ ] rapid elasticity
- [x] measured service



grid computing

Ian Foster et al. 2001

Power Grid Analogy



Power grid

multiple providers

heterogeneous sources

multiple clients

abstract source

live consumption

location

doing * at

Computing grid

virtual organisation

heterogeneous hw.

multiple applications

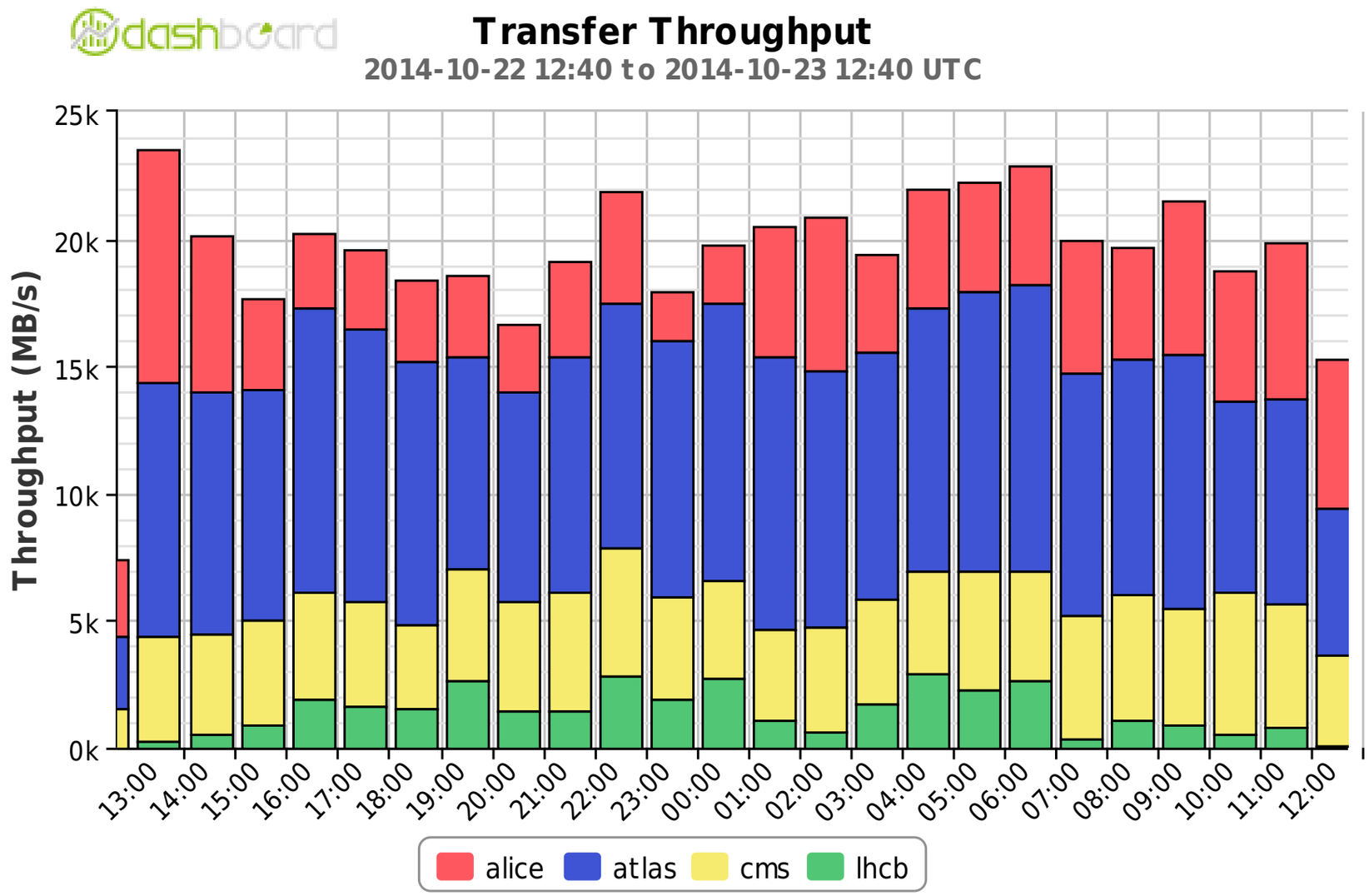
abstract resources

batch jobs

independence

large scale

Worldwide LHC Computing grid



170 centres to analyse 30 PB / year

Cloud or not ?

- ☐ on demand self-services
- ☐ broad network access
- ☐ resource pooling
- ☐ rapid elasticity
- ☐ measured service

Cloud or not ?

- [ ] on demand self-services
- [x] broad network access
- [x] resource pooling
- [ ] rapid elasticity
- [x] measured service

Client 1

Client 2

Client 3

Application Service Provider

95+

remote access to
dedicated applications

service oriented

pay as you go

Cloud or not ?

- ☐ on demand self-services
- ☐ broad network access
- ☐ resource pooling
- ☐ rapid elasticity
- ☐ measured service

Cloud or not ?

- ☐ on demand self-services
- ☑ broad network access
- ☑ resource pooling (not real hw resources)
- ☐ rapid elasticity
- ☑ measured service

2002 computers on demand.

Deploy full custom stacks (OS to applications)

emulab

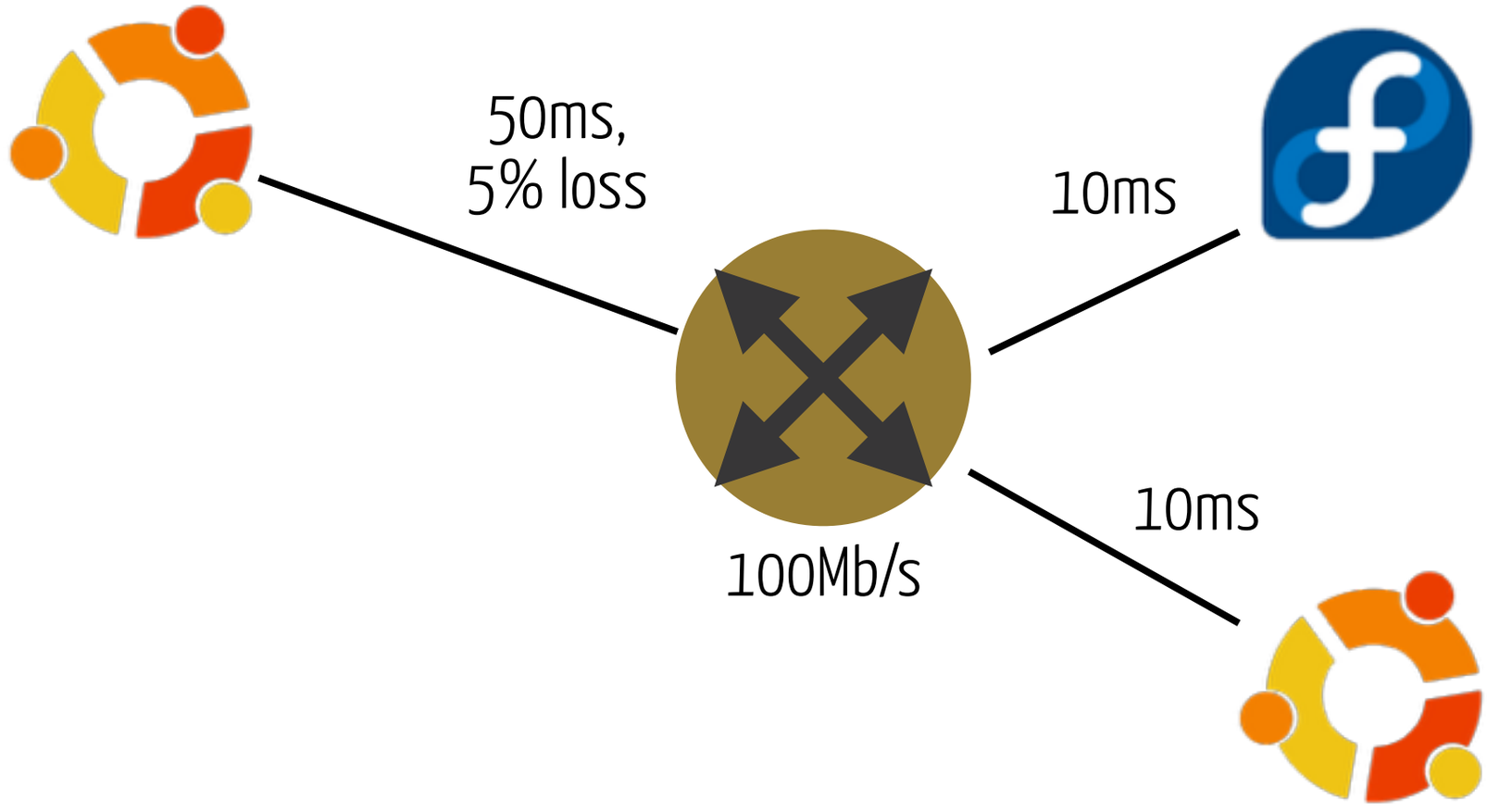
PLANETLAB

to (re)deploy reproducible network experiments

multi-tenant,
(limited on purpose) resource pooling,

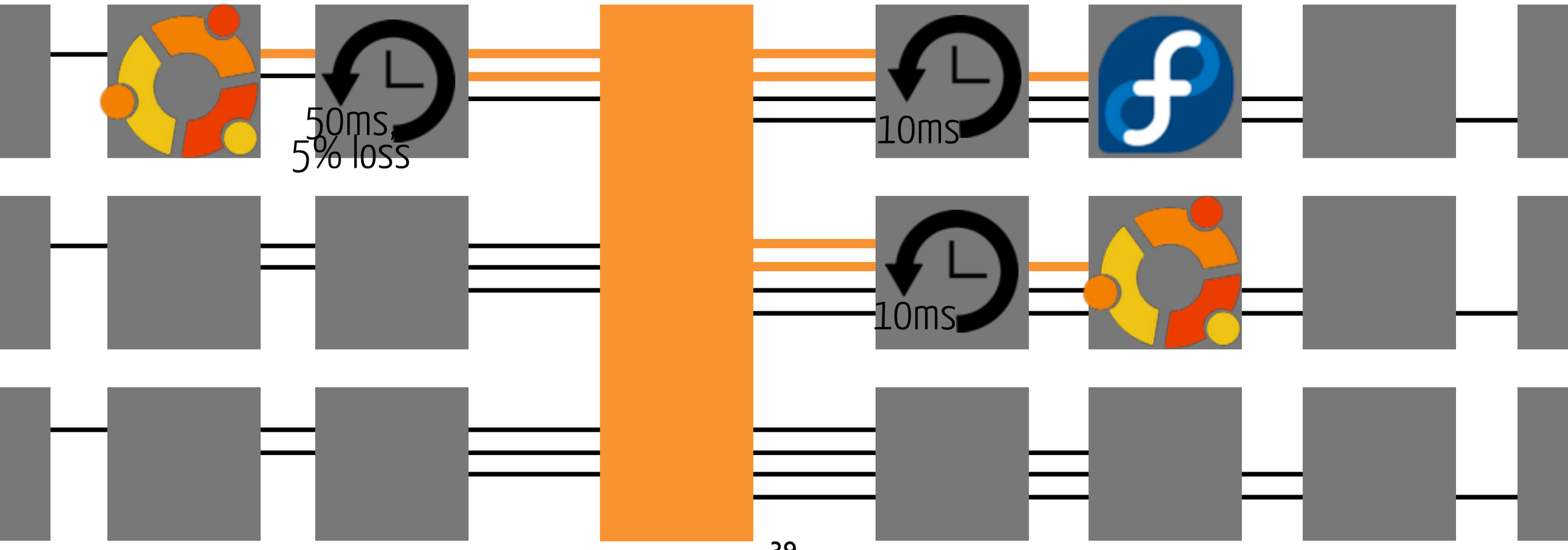
emulab

to (re)deploy reproducible network experiments



emulab

to (re)deploy reproducible network experiments



Cloud or not ?

- ☐ on demand self-services
- ☐ broad network access
- ☐ resource pooling
- ☐ rapid elasticity
- ☐ measured service

Cloud or not ?

- [x] on demand self-services
- [x] broad network access
- [x] resource pooling
- [ ] rapid elasticity
- [x] measured service

2001+

Service **O**riented **A**rchitecture

composable unassociated, loosely coupled units

exponential grows since 2001
private and public services to support its growth

Two pizza rule

If a team can’t be fed by two pizzas then it is to big

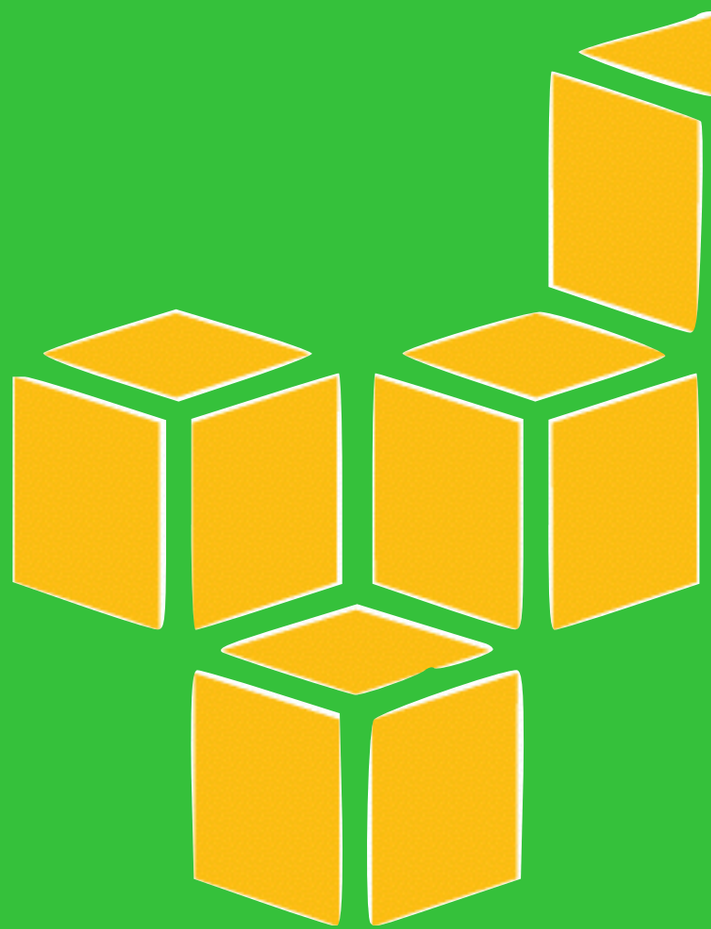
- Jeff Bezos (founder/ CEO of amazon.com)

800 x 

tons of API, microservices devoted to automation, flexibility, on-demand services for public and private use

2006

scalable web services for other
websites or client-side applications



SOAP & REST over HTTP
pay as you go
elastic *-oriented services

*data, network or computation

Cloud !

- ☑ on demand self-services
- ☑ broad network access
- ☑ resource pooling
- ☑ rapid elasticity
- ☑ measured service

RECAP

“I have a dream, it was about Utility Computing”

John McCarthy - 1961

web
+ grid computing
+ resources on demand
+ service oriented architectures

cloud computing (2006)


can we talk about
cloud computing now ?

?aaS

SaaS
Service as a Service

web access to commercial sw.
"one to many" model
customers don't handle upgrades
API for integration

Users.messages

For Users.messages Resource details, see the resource representation page.

Method	HTTP request	Description
URIs relative to https://www.googleapis.com/gmail/v1/users, unless otherwise noted		
delete	`DELETE /userId/messages/id`	Immediately and permanently deletes the specified message. This operation cannot be undone. Prefer `messages.trash` instead.
get	`GET /userId/messages/id`	Gets the specified message.
insert	`POST https://www.googleapis.com/upload/gmail/v1/users/userId/messages` and `POST /userId/messages`	Directly inserts a message into only this user's mailbox similar to `IMAP APPEND`, bypassing most scanning and classification. Does not send a message.
list	`GET /userId/messages`	Lists the messages in the user's mailbox.
modify	`POST /userId/messages/id/modify`	Modifies the labels on the

Plans and pricing

GitHub is free to use for public projects. Collaborate on private repositories with any of our paid plans.

Sign up now

Personal plans

Display estimated prices in EUR

For individuals looking to share their own projects and collaborate with others.

	Free $0/month	Micro $7/month	Small $12/month	Medium $22/month	Large $50/month
Collaborators	Unlimited	Unlimited	Unlimited	Unlimited	Unlimited
Public repositories	Unlimited	Unlimited	Unlimited	Unlimited	Unlimited
Private repositories	0	5	10	20	50

Do you offer education discounts?

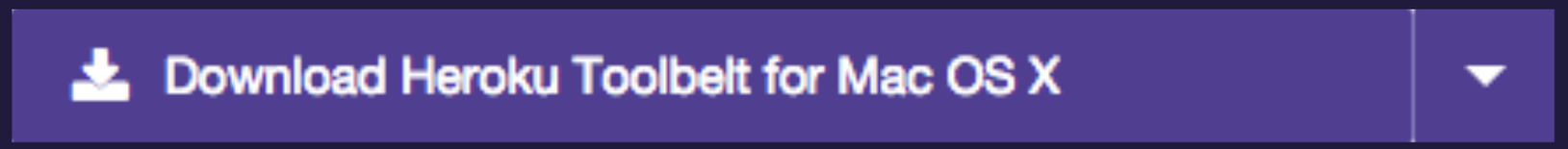
Yes, we offer free and discounted plans to students, teachers, and schools. You can find more information and apply on our education site.

PaaS Platform as a Service

jailed runtime available to host applications

generic or provider-specific APIs

no control over the environment




```
$ heroku login
…

$ git clone https://github.com/heroku/java-getting-started.git
$ cd java-getting-started
```

```
$ heroku create
Creating warm-eyrie-9006... done, stack is cedar-14
http://warm-eyrie-9006.herokuapp.com/ | git@heroku.com:warm-eyrie-9006.git
Git remote heroku added

$ git push heroku master
…
       http://warm-eyrie-9006.herokuapp.com/ deployed to Heroku

$ heroku ps:scale web=1
```

Scale & estimate your dyno cost

1X	2X	Performance Dynos
512MB RAM 1x CPU Share	**1024MB RAM** **2x CPU Share**	6GB RAM Superior Quality-of-Service Low latency, high throughput
$0.05/dyno-hour	**$0.10/dyno-hour**	$0.80/dyno-hour

20

Summary

Dynos	$682.50
Databases	$0.00
Support	$0.00
Add-ons	$0.00

$682.50
Estimated monthly cost

Sign up for free

IronMQ from $0/mo

Highly available elastic message queuing service.

Plans

Lite	Free
Starter	**$29/mo**
Dev	$129/mo
Pro	$499/mo

Requests per month	10M
Unlimited Queues	✔
High Availability	✔
Persistent Messages	✔
Push Queues	✔

IaaS Infrastructure as a Service

low-level resources to deploy arbitrary software stacks
complete control over its network, storage and OS

EC2 Dashboard
Events
Tags
Reports
Limits

INSTANCES
Instances
Spot Requests
Reserved Instances

IMAGES
AMIs
Bundle Tasks

ELASTIC BLOCK STORE
Volumes
Snapshots

NETWORK & SECURITY
Security Groups
Elastic IPs
Placement Groups
Load Balancers
Key Pairs
Network Interfaces

Resources

You are using the following Amazon EC2 resources in the EU West (Ireland) region:

2 Running Instances
9 Volumes
1 Key Pair
0 Placement Groups
0 Elastic IPs
6 Snapshots
2 Load Balancers
6 Security Groups

Easily deploy Ruby, PHP, Java, .NET, Python, Node.js & Docker applications with Elastic Beanstalk.
Hide

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Launch Instance

Note: Your instances will launch in the EU West (Ireland) region

Service Health

Service Status:

EU West (Ireland):
This service is operating normally

Availability Zone Status:

eu-west-1a:
Availability zone is operating normally

Scheduled Events

EU West (Ireland):

No events

Account Attributes

Supported Platforms

Default VPC
vpc-e1b57684

Additional Information

Getting Started Guide
Documentation
All EC2 Resources
Forums
Pricing
Contact Us

AWS Marketplace

Find **free software trial** products in the AWS Marketplace from the EC2 Launch Wizard.
Or try these popular AMIs:

Vyatta Virtual Router/Firewall/VPN

Provided by Vyatta, Inc.
Rating ★★★★☆
Pay by the hour for software and AWS usage
View all Networking

Bills

Date: November 2014

Download CSV | Print

Summary	Amount
AWS Service Charges	**$0.00**
There are no invoices for the selected month.	

+ Expand All

Details		Total
AWS Service Charges		**$0.00**
▶ **Data Transfer**		**$0.00**
▼ **Elastic Compute Cloud**		**$0.00**
EU (Ireland) Region	**Usage**	
Amazon CloudWatch		
$0.00 per alarm-month - first 10 alarms	0.373 Alarms	$0.00
$0.00 per metric-month - first 10 metrics	0.826 Metrics	$0.00
Total:		**$0.00**
Amazon Elastic Compute Cloud running Linux/UNIX		
Region Total:		**$0.00**
▶ **Simple Notification Service**		**$0.00**
▶ **Simple Queue Service**		**$0.00**
▶ **Simple Storage Service**		**$0.00**
▶ **CT to be collected**		**$0.00**

ol' skool IT

you manage

- applications
- runtimes
- integration/security
- database
- servers
- virtualisation
- server HW
- storage
- network

SaaS

applications

runtimes

integration/security

database

servers

virtualisation

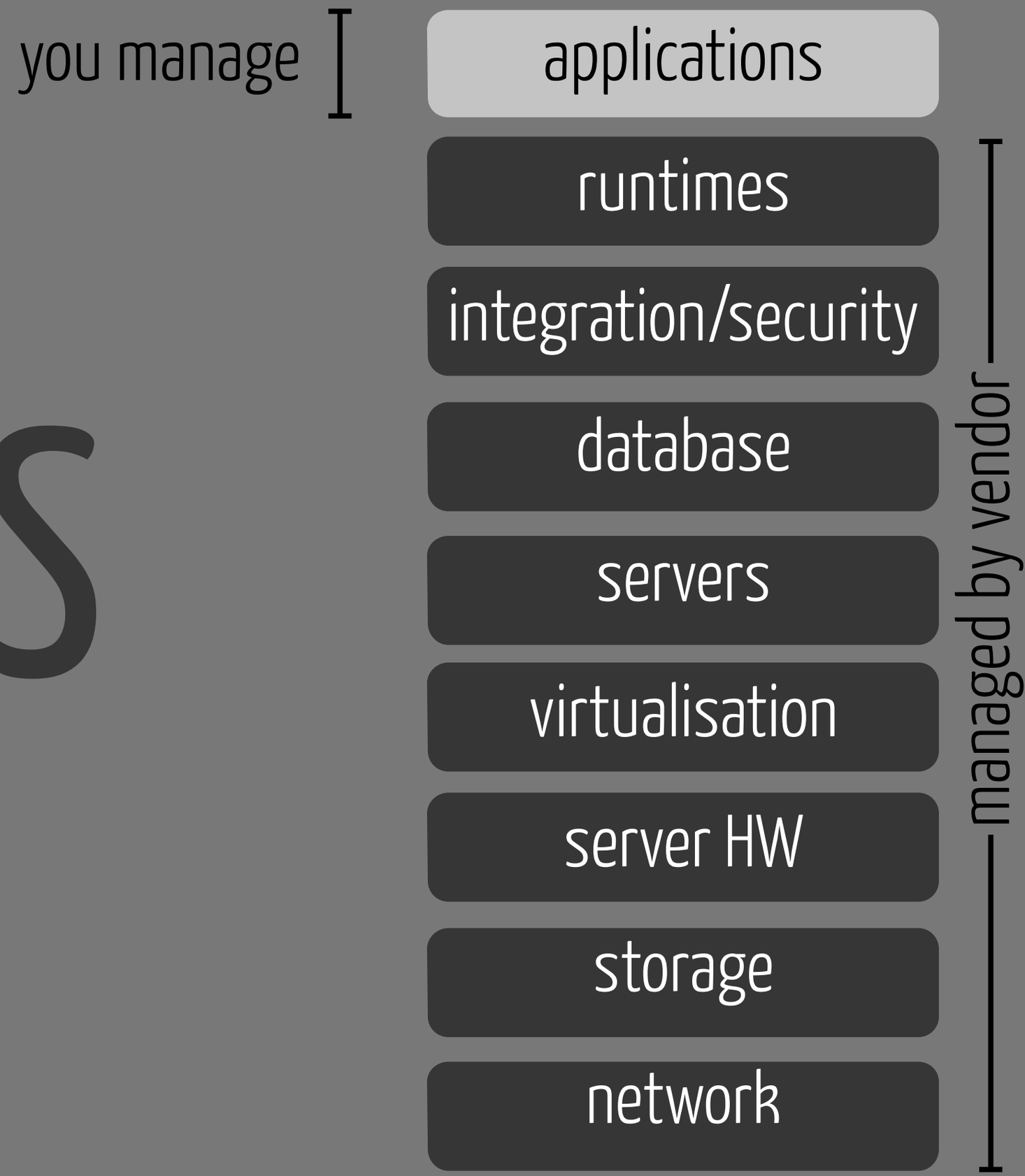
server HW

storage

network

managed by vendor

PaaS


you manage
applications
runtimes
integration/security
database
servers
virtualisation
server HW
storage
network
managed by vendor

Open-source PaaS stacks

OPENSHIFT

CLOUD FOUNDRY™

Cloudify

IaaS

you manage

- applications
- runtimes
- integration/security
- database
- servers

managed by vendor

- virtualisation
- server HW
- storage
- network

Open-source IaaS stacks

EUCALYPTUS (2008+)

OpenNebula (2008+)

openstack™ (2010+)

cloudstack (2012+)

Elastic Beanstalk

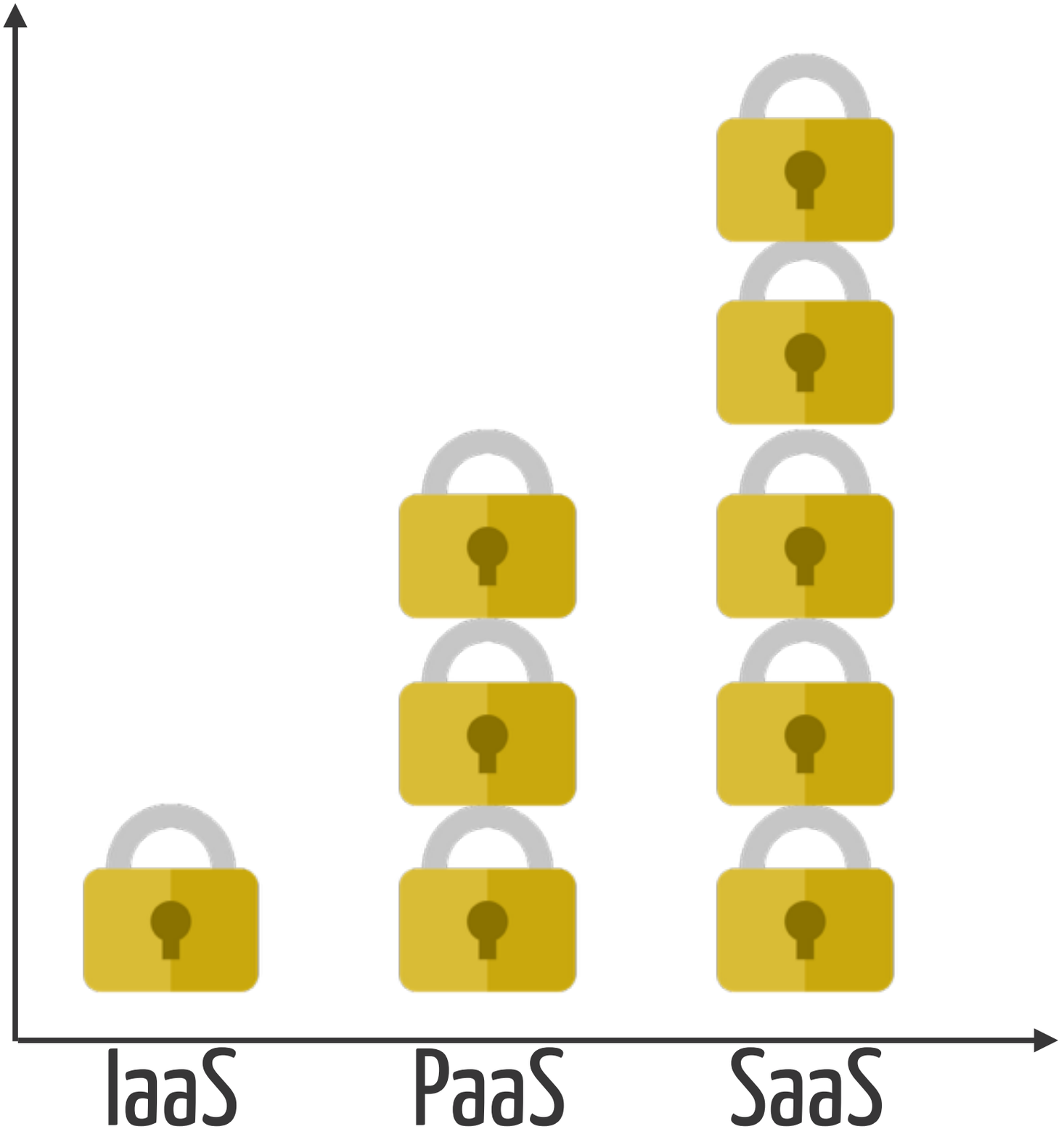
amazon DynamoDB

Cloud Storage

Cloud Datastore

S3

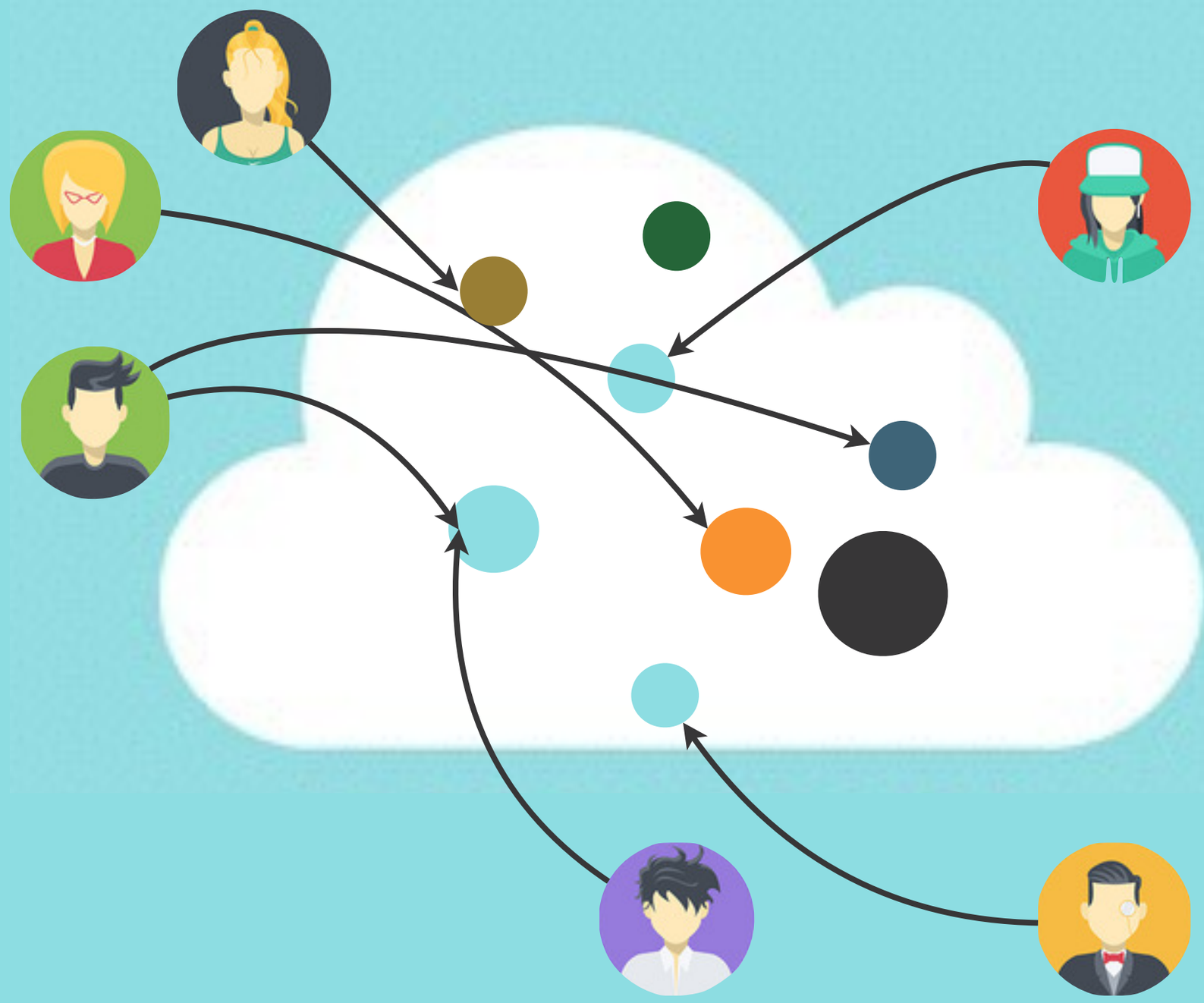
vendor lock-in



Deployment models

public cloud

general availability to everyone



the “real” cloud
reduced costs
trust issues ?

cloud computing
vs.
fog of war

give me your code & data

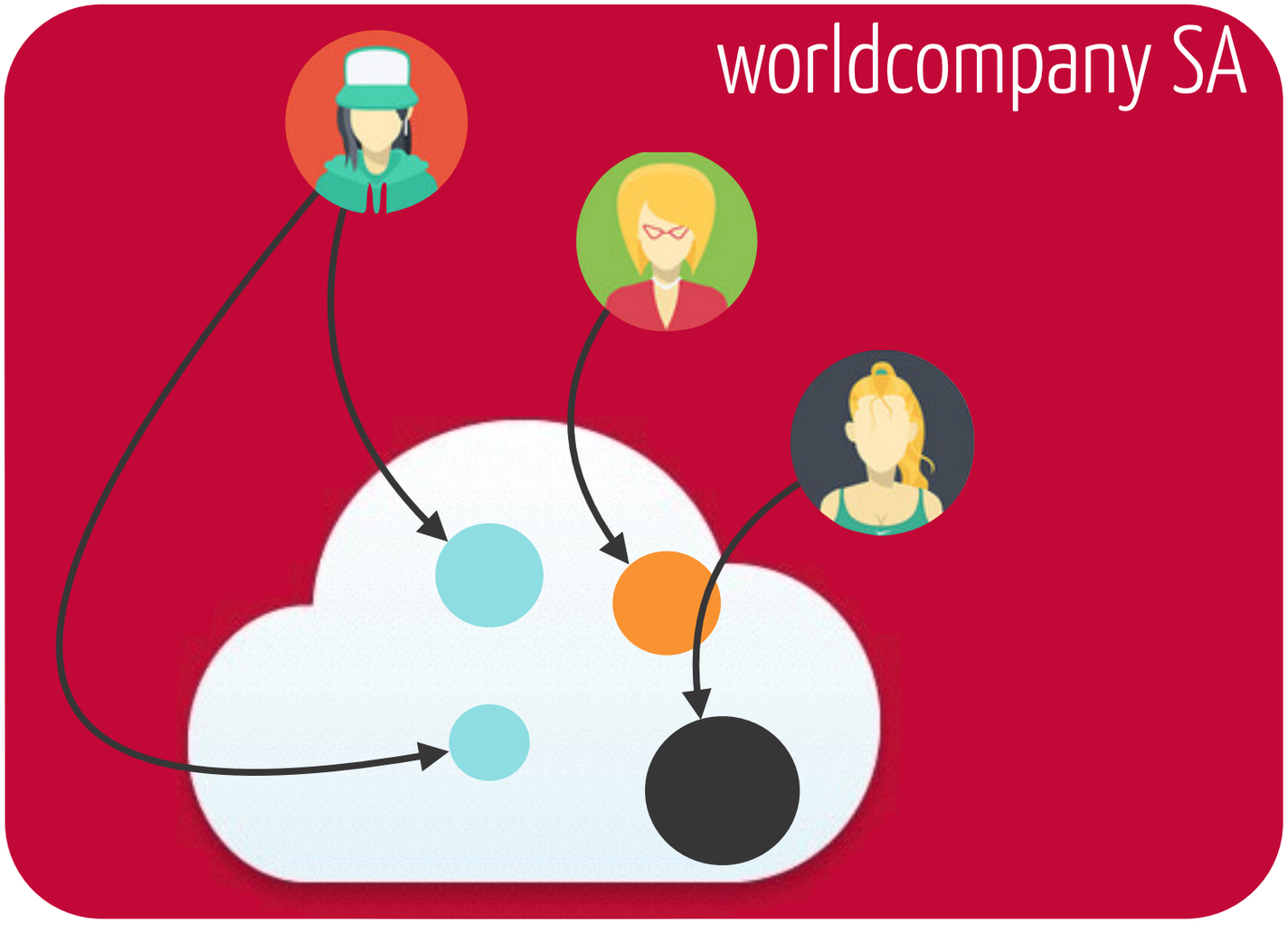
Trust in me

I'm aware  read my mails

what is my  is hacked ?

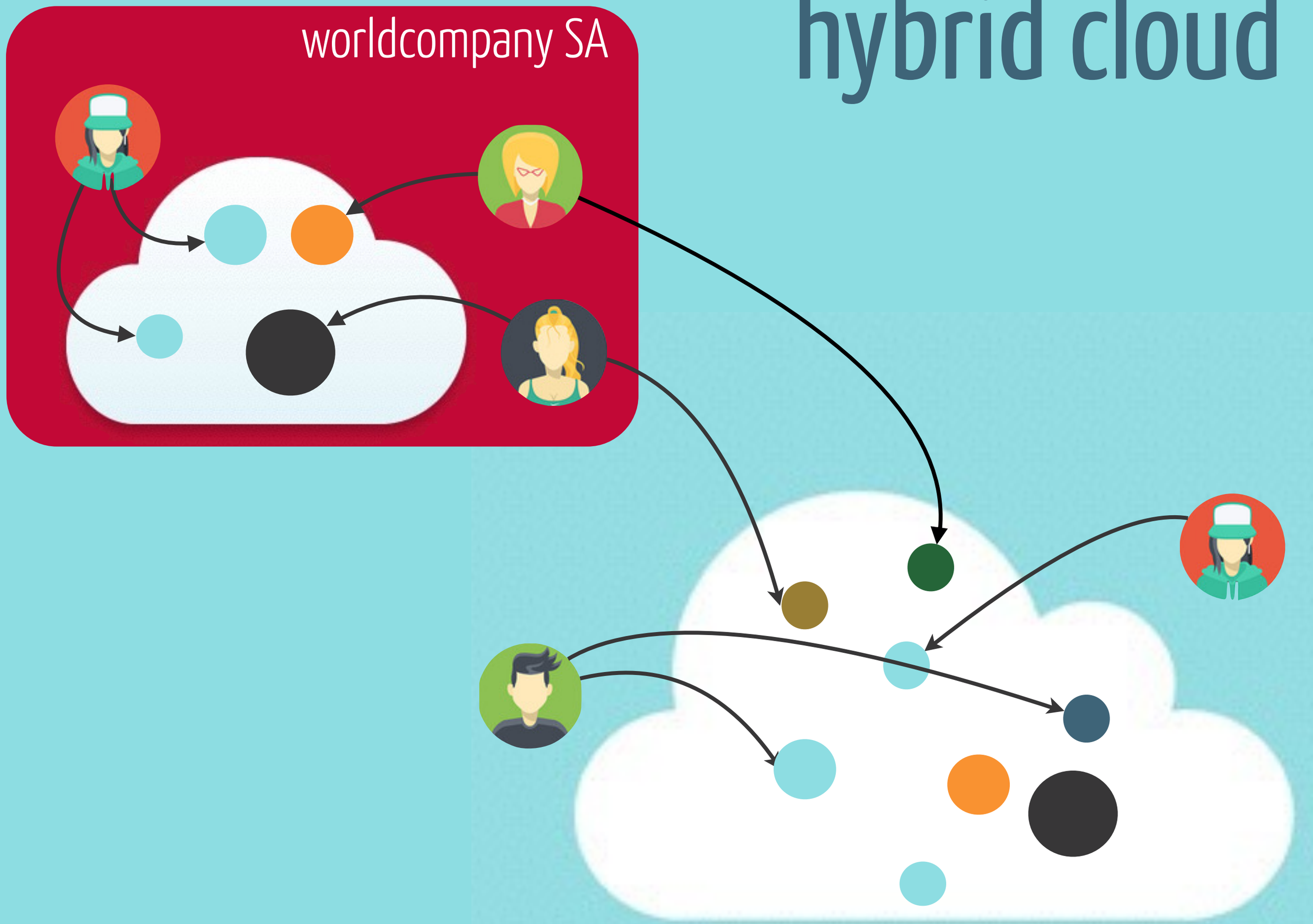
private cloud

self hosted cloud



might reduce TCO
stronger trust
better manageability

hybrid cloud



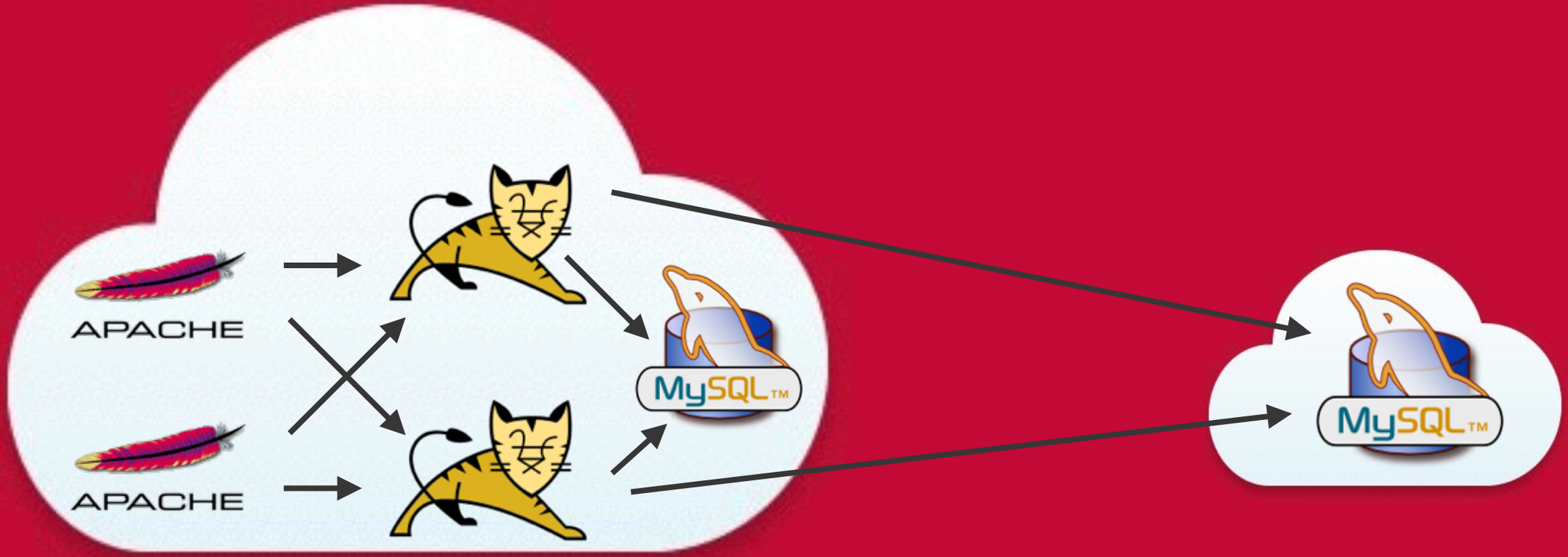
LB

multi-clouds

you spread your application
avoid Single Point of Failures*
take the benefits of each cloud

inter-clouds

they outsource your components
agreements between the providers
"cloud of clouds" (federation)

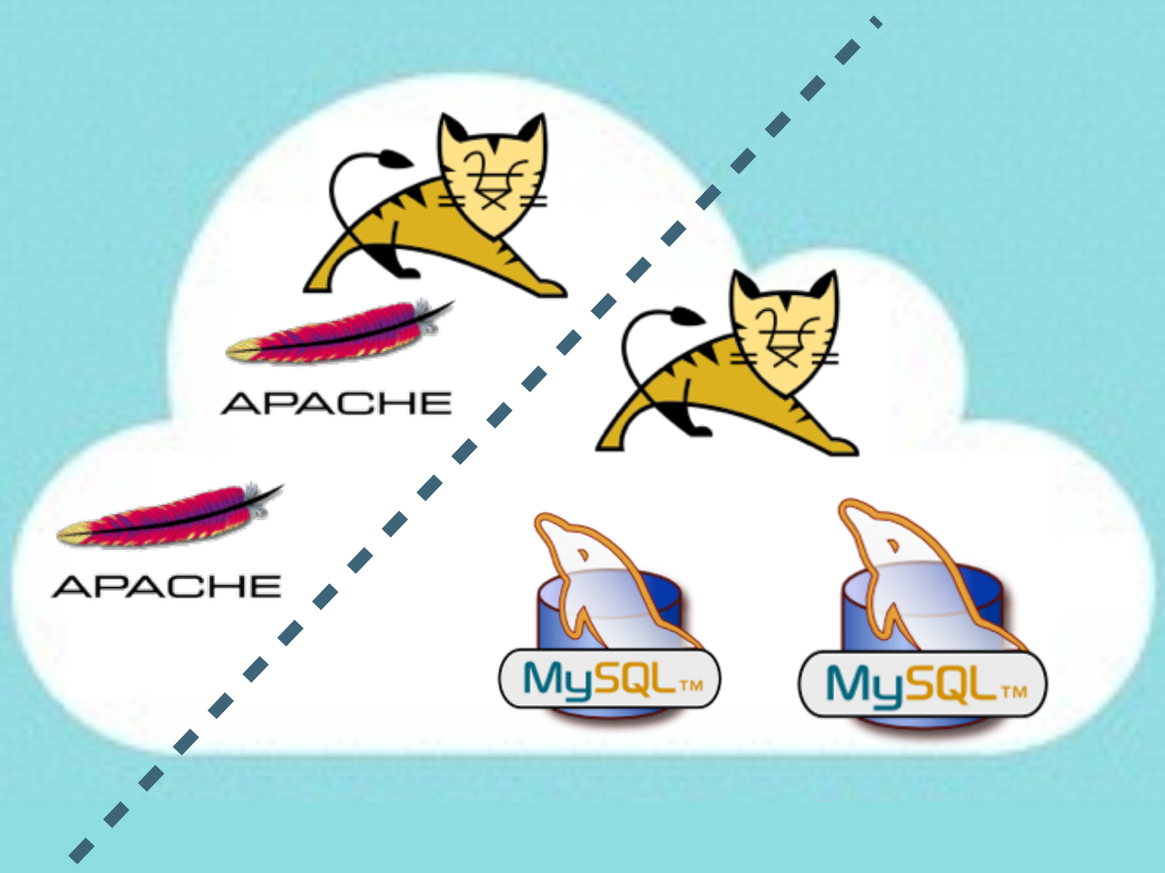


distributed clouds

back to volunteer computing
(Boinc, cloud@home, ...)

community cloud

private cloud by and for
multiple organizations



“

Assume you could start with super reliable servers (MTBF of 30 years)
Build computing system with 10 thousand of those
Watch one fail per day

Things will crash. Deal with it!

”

Dean Keynote, LADIS 2009

Typical first year for a new google cluster

~0.5 overheating (power down most machines in <5 mins, ~1-2 days to recover)
~1 PDU failure (~500-1000 machines suddenly disappear, ~6 hours to come back)
~1 rack-move (plenty of warning, ~500-1000 machines powered down, ~6 hours)
~1 network rewiring (rolling ~5% of machines down over 2-day span)
~20 rack failures (40-80 machines instantly disappear, 1-6 hours to get back)
~5 racks go wonky (40-80 machines see 50% packetloss)
~8 network maintenances (4 might cause ~30-minute random connectivity losses)
~12 router reloads (takes out DNS and external vips for a couple minutes)
~3 router failures (have to immediately pull traffic for an hour)
~dozens of minor 30-second blips for dns
~1000 individual machine failures
~thousands of hard drive failures
slow disks, bad memory, misconfigured machines, flaky machines, etc.
Long distance links: wild dogs, sharks, dead horses, drunken hunters, etc.

“A **distributed system** is one in which the failure of a computer you didn't even know existed can render your own computer unusable”

Leslie Lamport

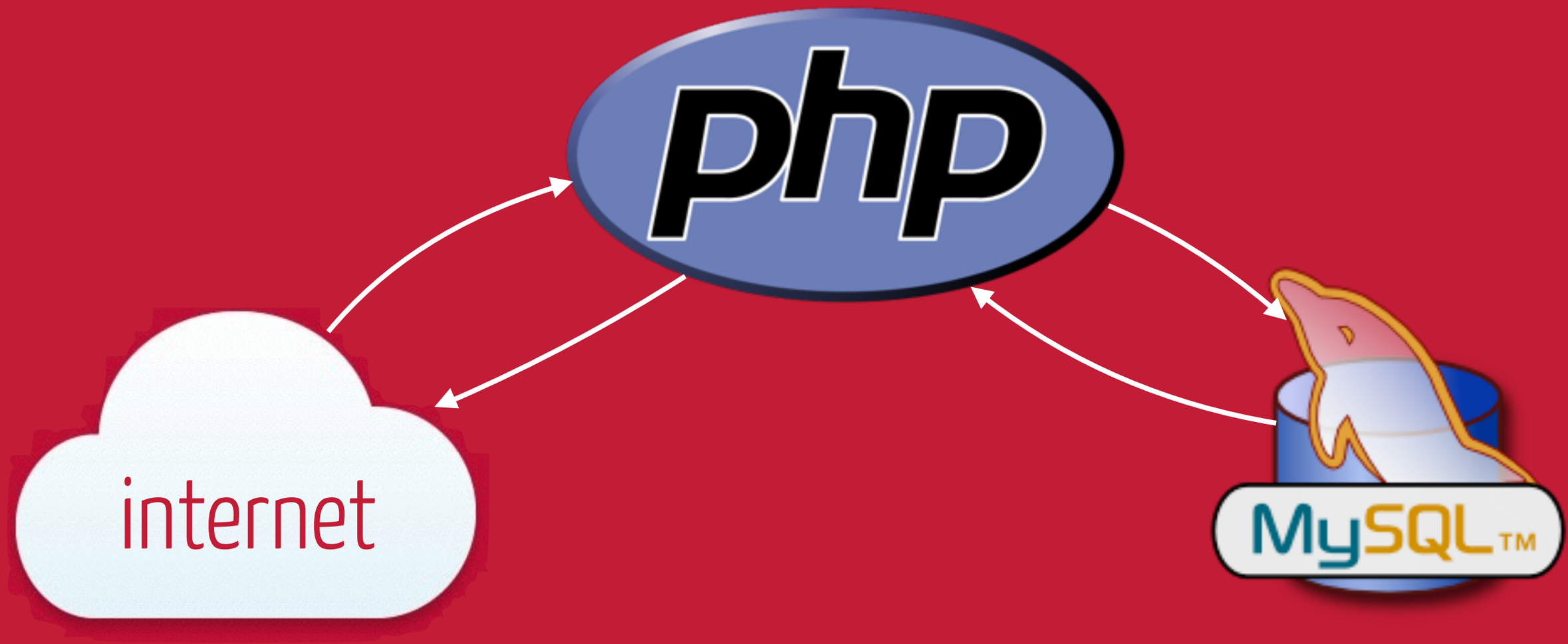
Building fault tolerant services

at every level

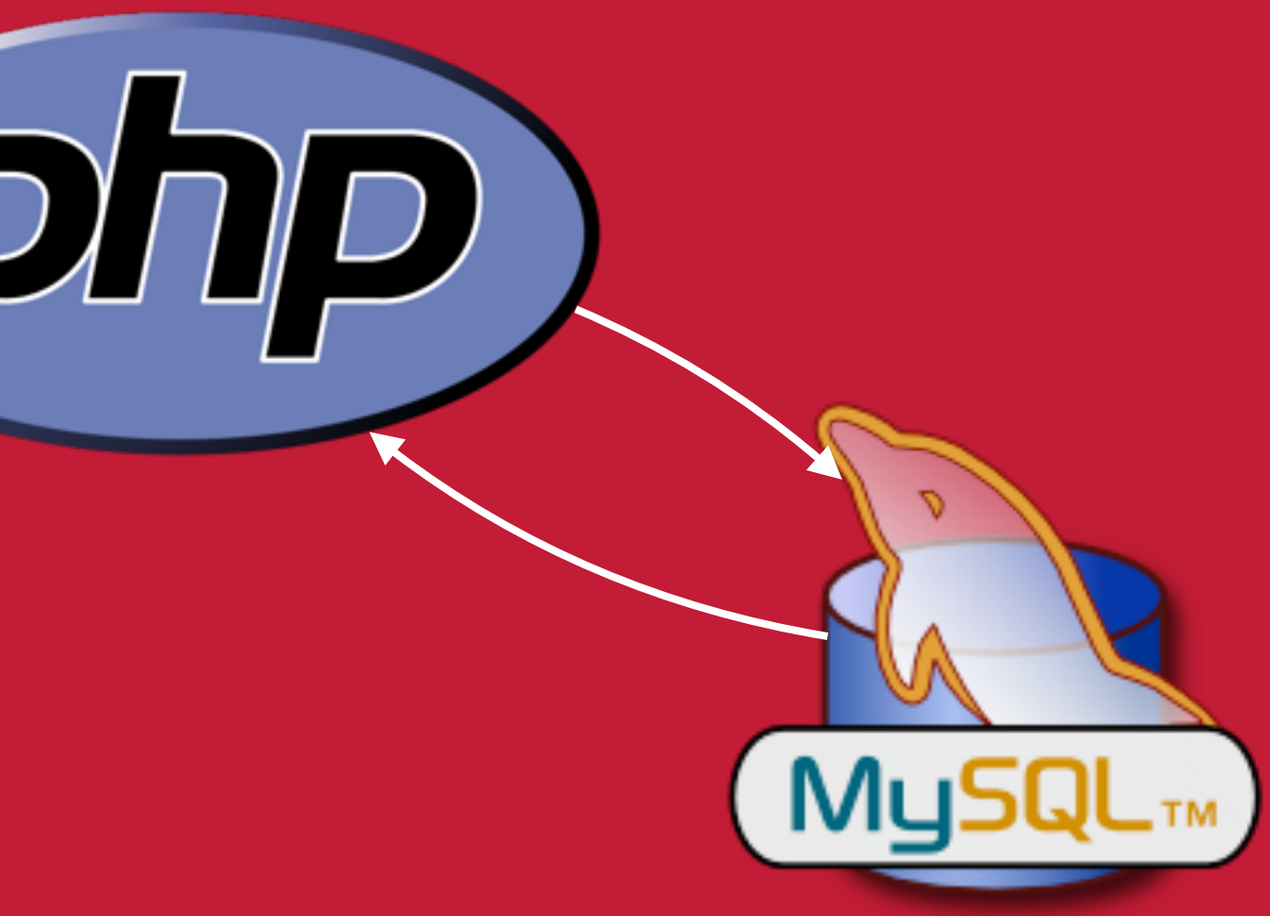
be pessimistic

deal with failures
deal with inconsistency

prehistorical times

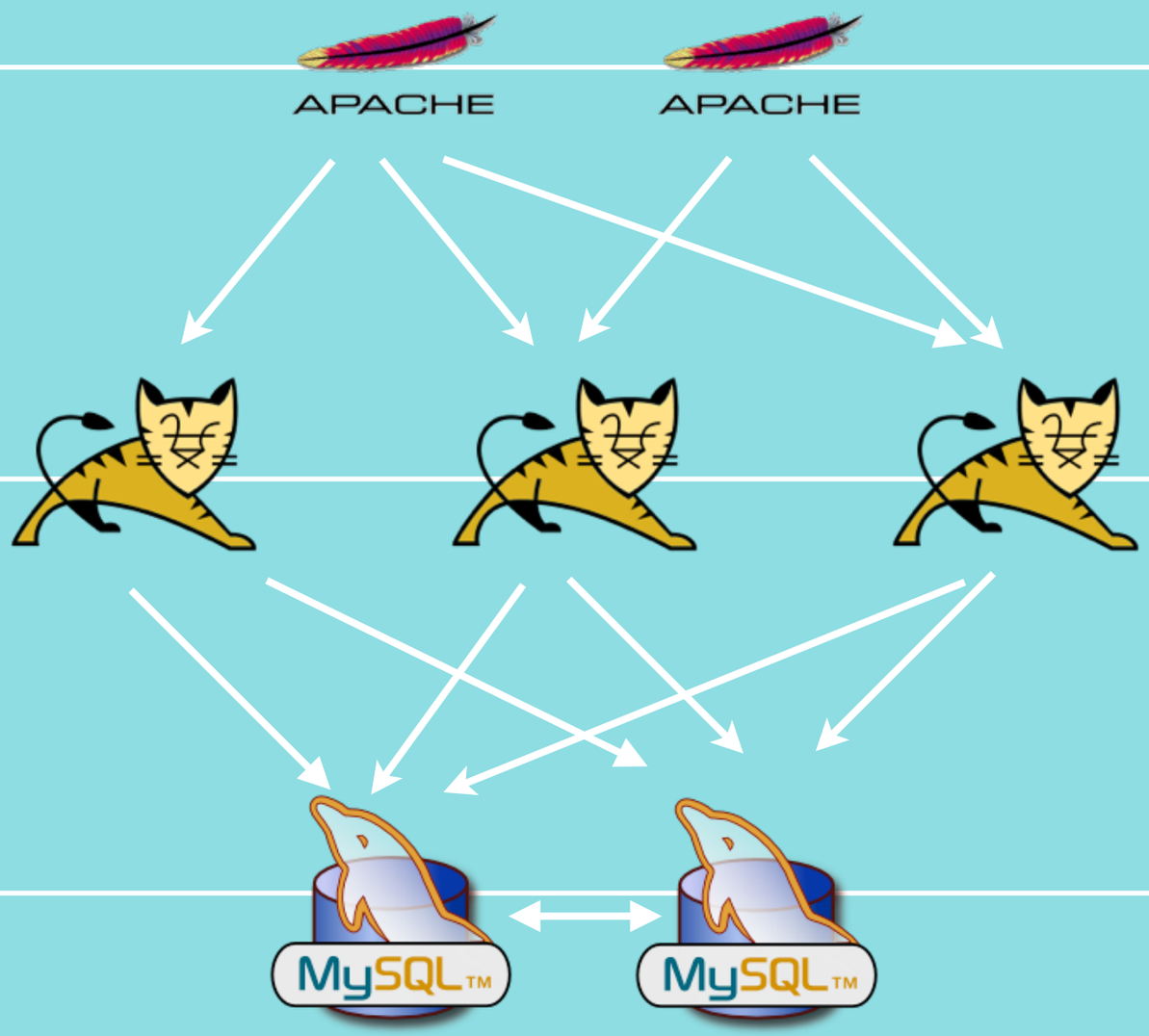


prehistorical times



2 **S**ingle **P**oint **o**f **F**ailure
limited scalability
not scalable online
not upgradable online

n-tiered apps to the rescue

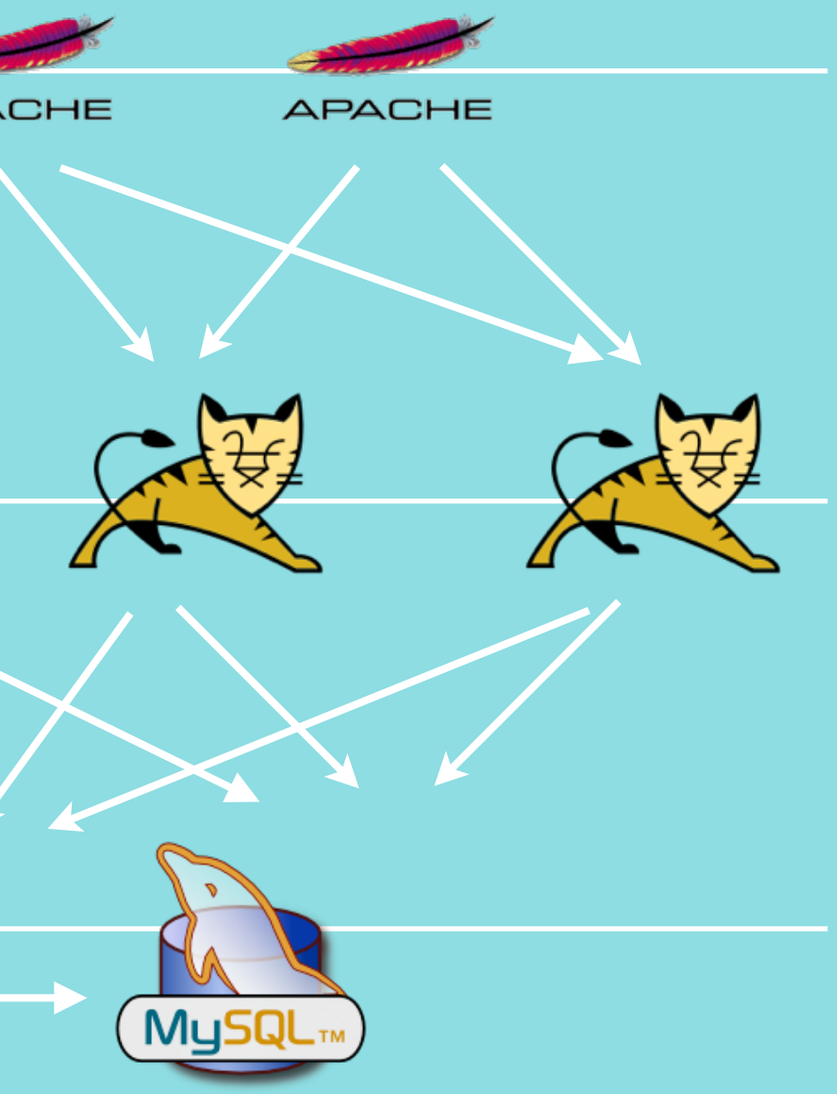


Tiers 1

Tiers 2

Tiers 3

n-tiered apps to the rescue



Tiers 1

Tiers 2

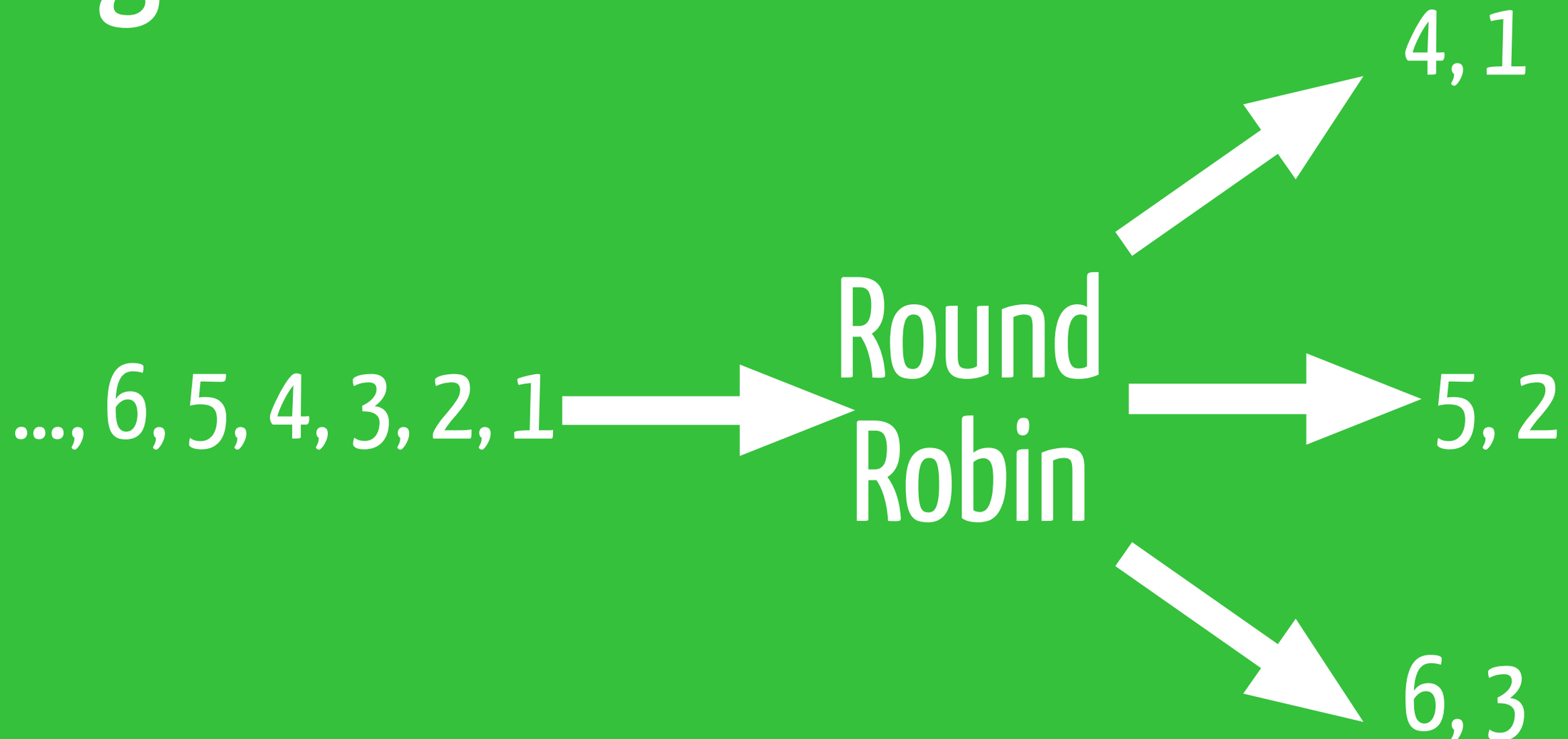
Tiers 3

load balancing
horizontal scalability
upgradable online

spread incoming requests
to backend services

synchronous communications

possible load balancing algorithms



nice on homogeneous nodes

possible load balancing algorithms

6, 5, 3, 1

..., 6, 5, 4, 3, 2, 1 → weighted →

4, 2

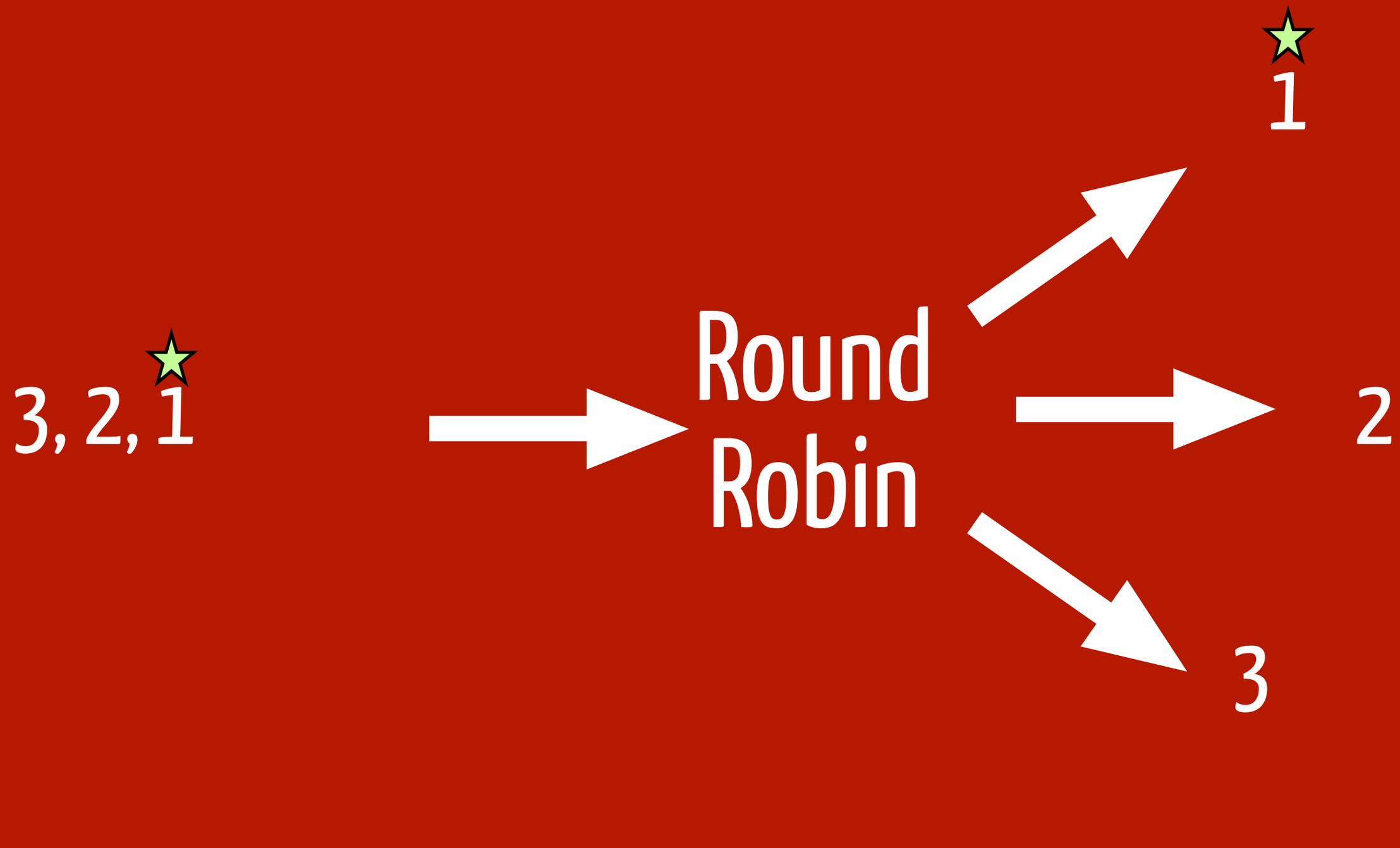
weight the nodes or the requests
depending on their size

“spread incoming requests
to backend services

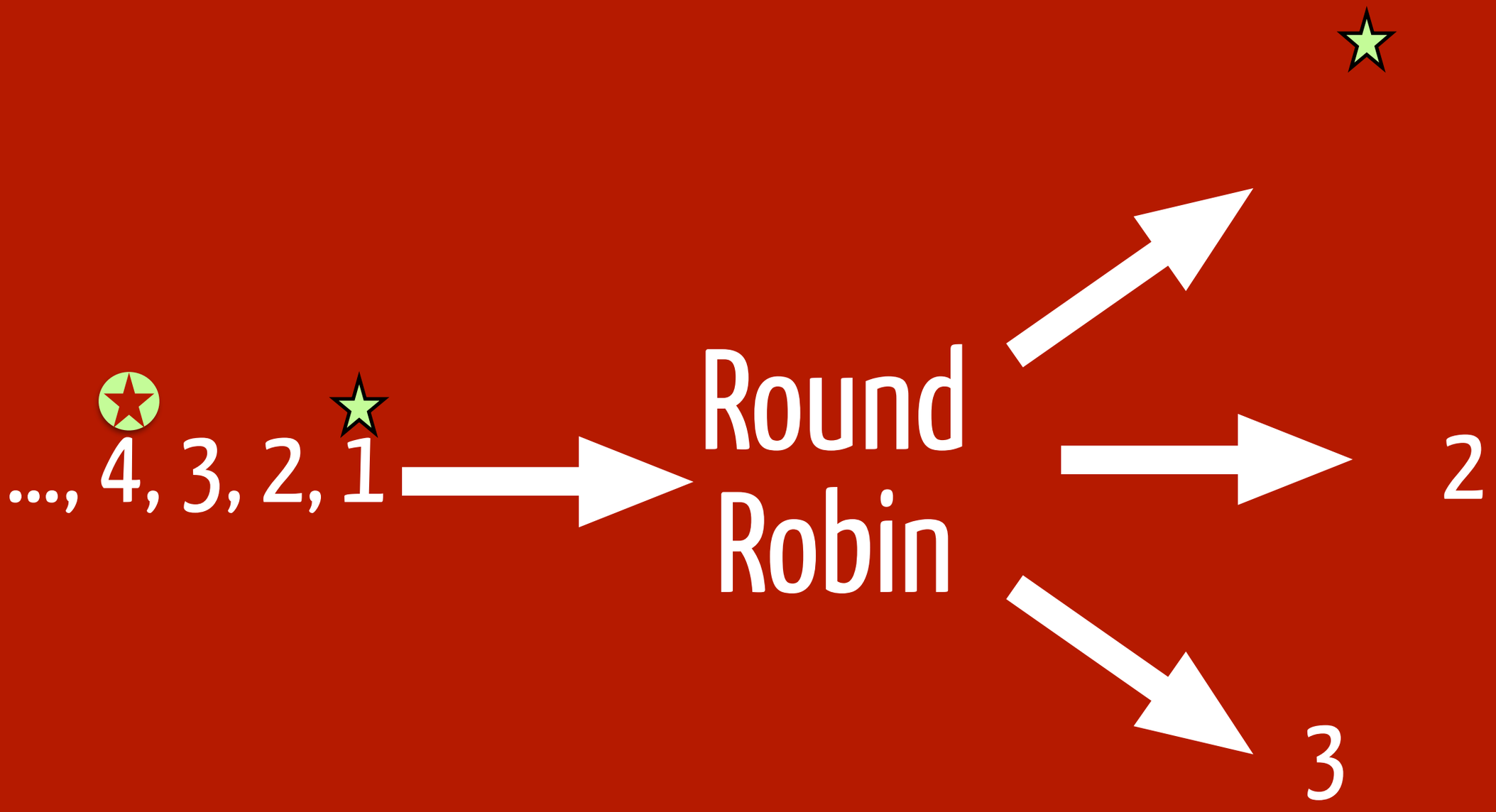
synchronous communications”

is everything balanceable ?

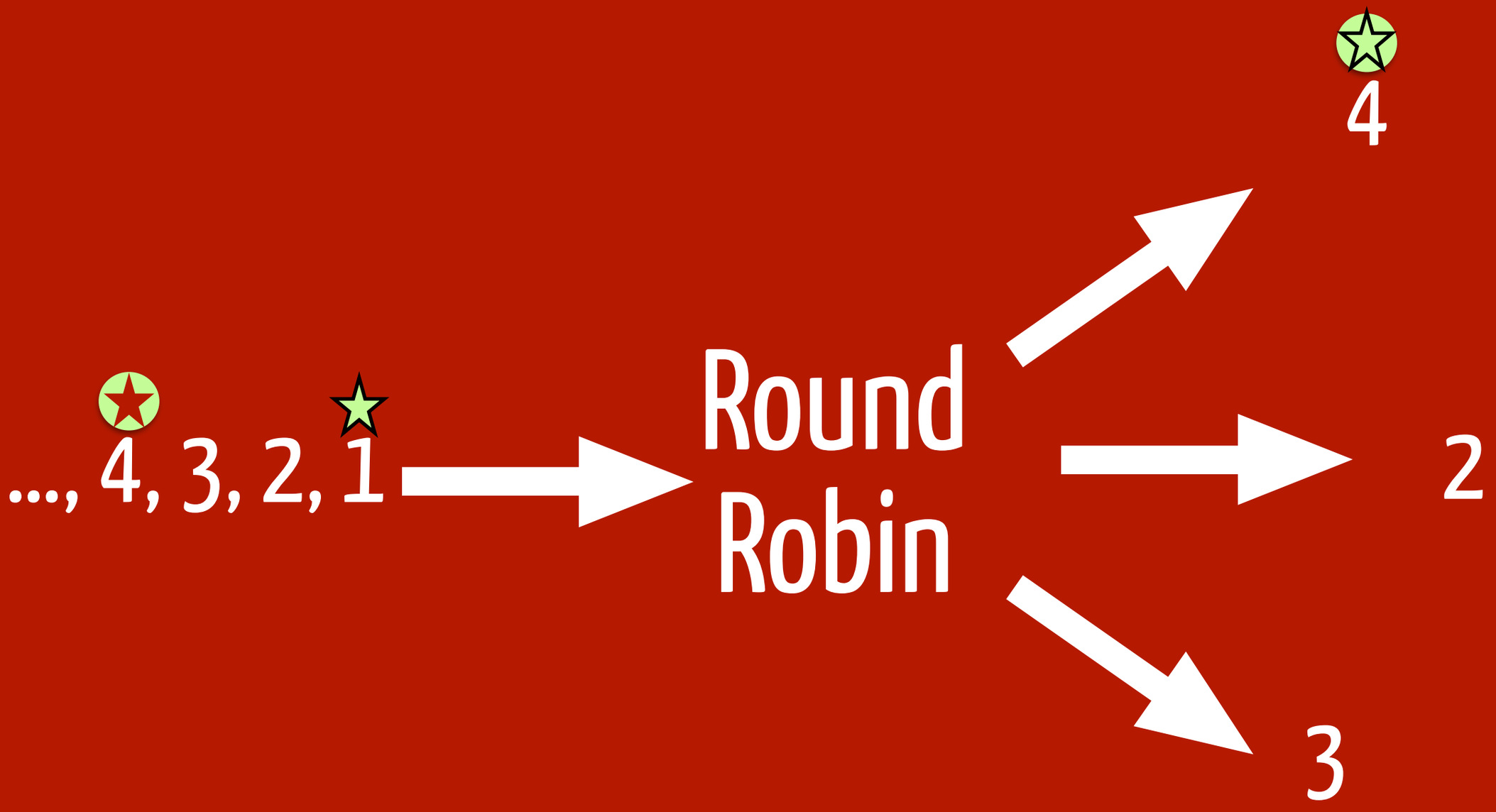
load balancing stateful stuff



load balancing stateful stuff



load balancing stateful stuff



load balancing stateful stuff

achievement unlocked

4

…, 4, 3, 2, 1

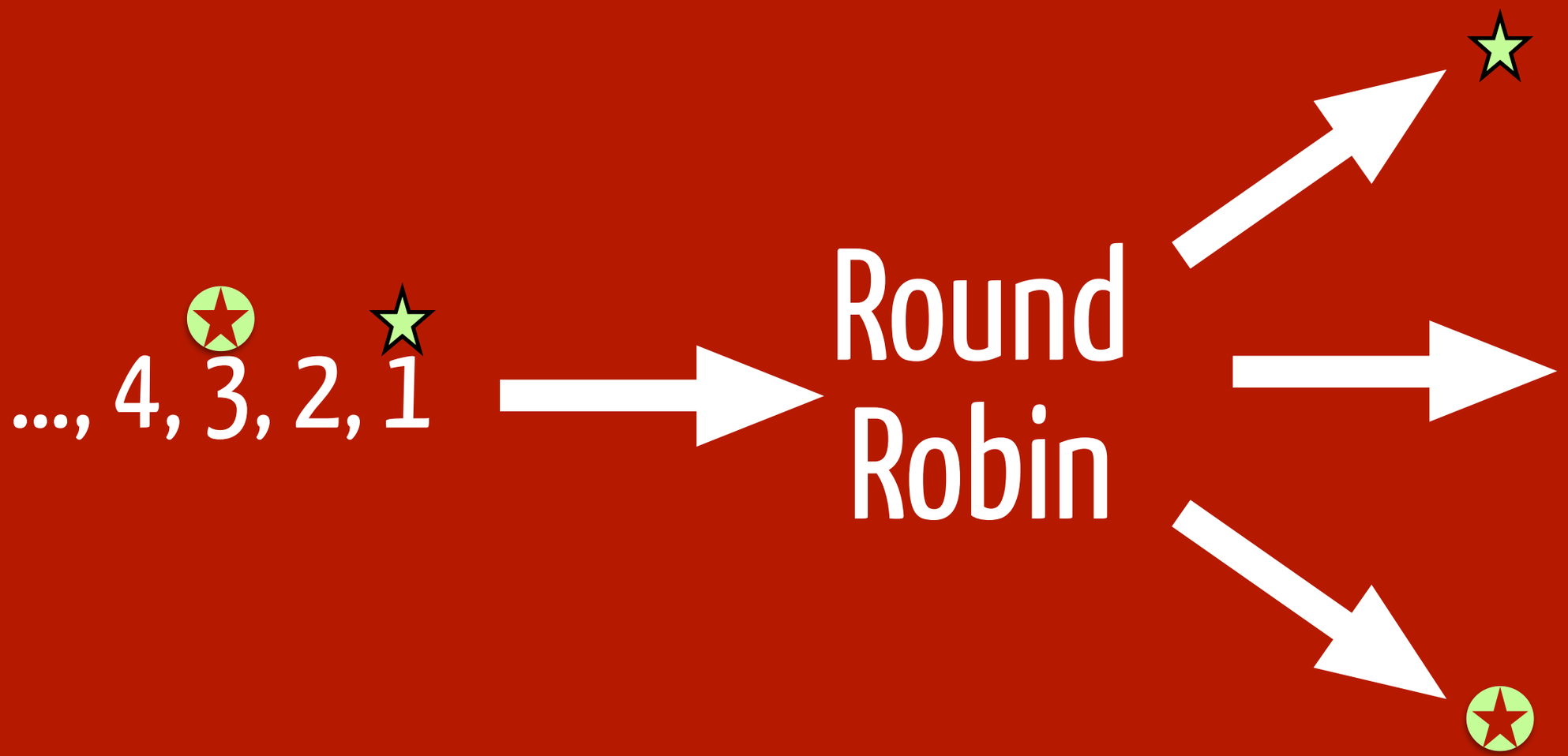
Round Robin

4

2

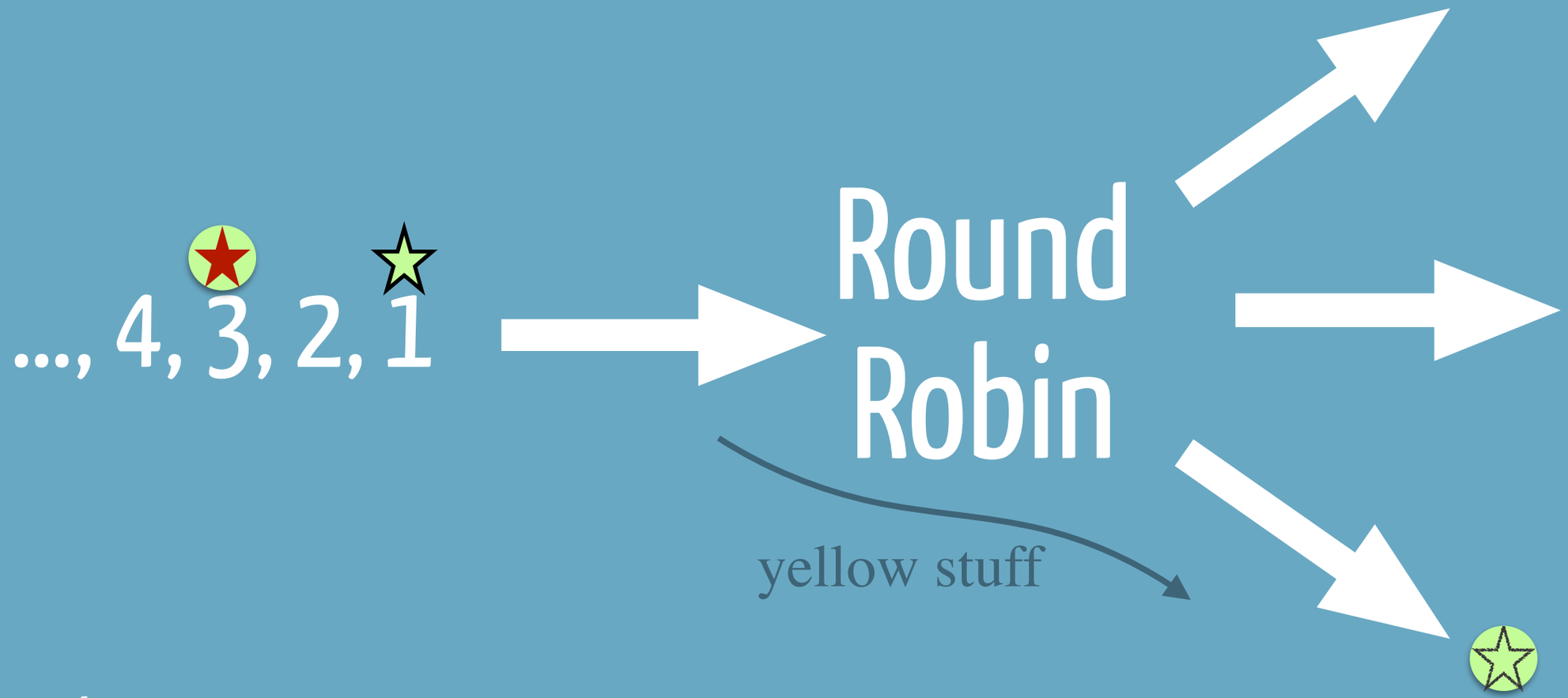
3

load balancing stateful stuff



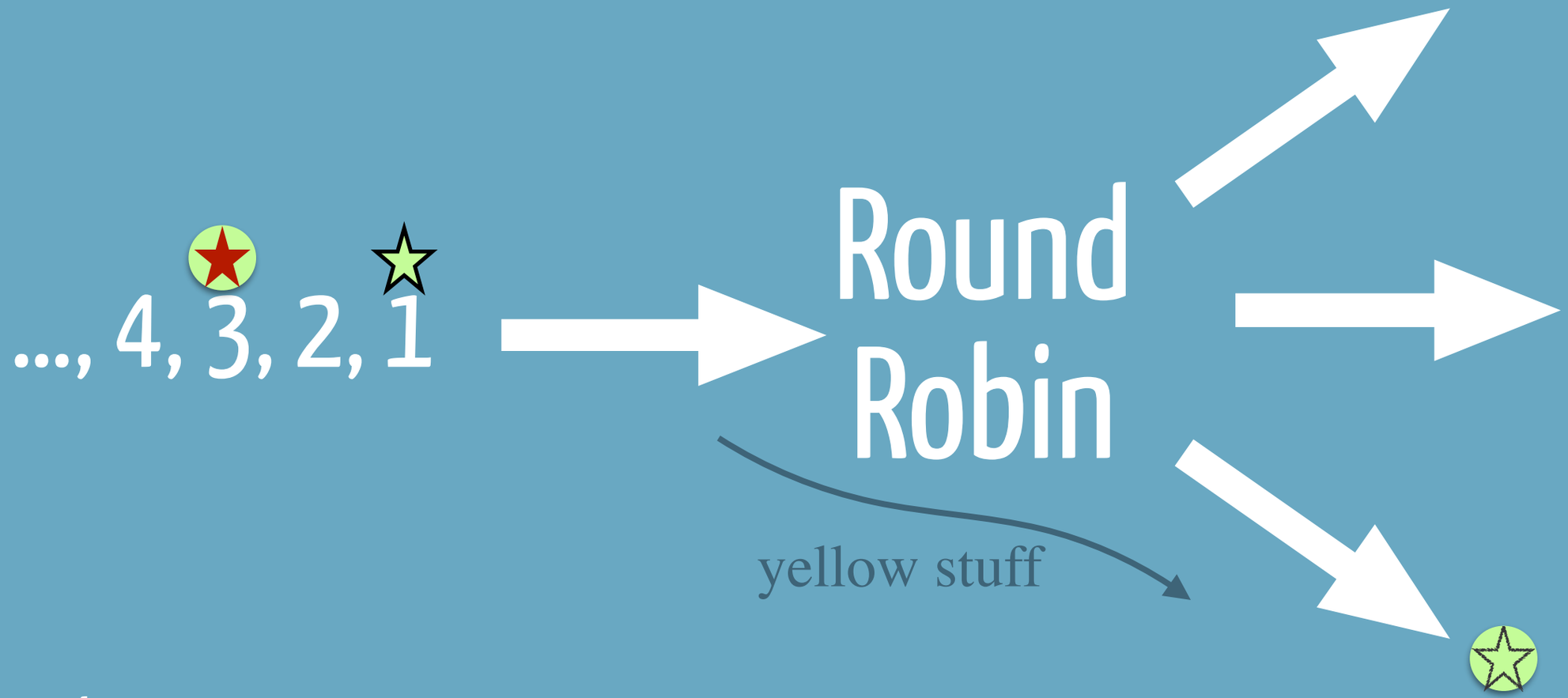
load balancing stateful stuff

load balancing stateful stuff + sticky sessions



IP based

load balancing stateful stuff + sticky sessions



IP based

not resilient to node failures

amazon
elastic load balancing

NGINX

APACHE mod_proxy_balancer

lb

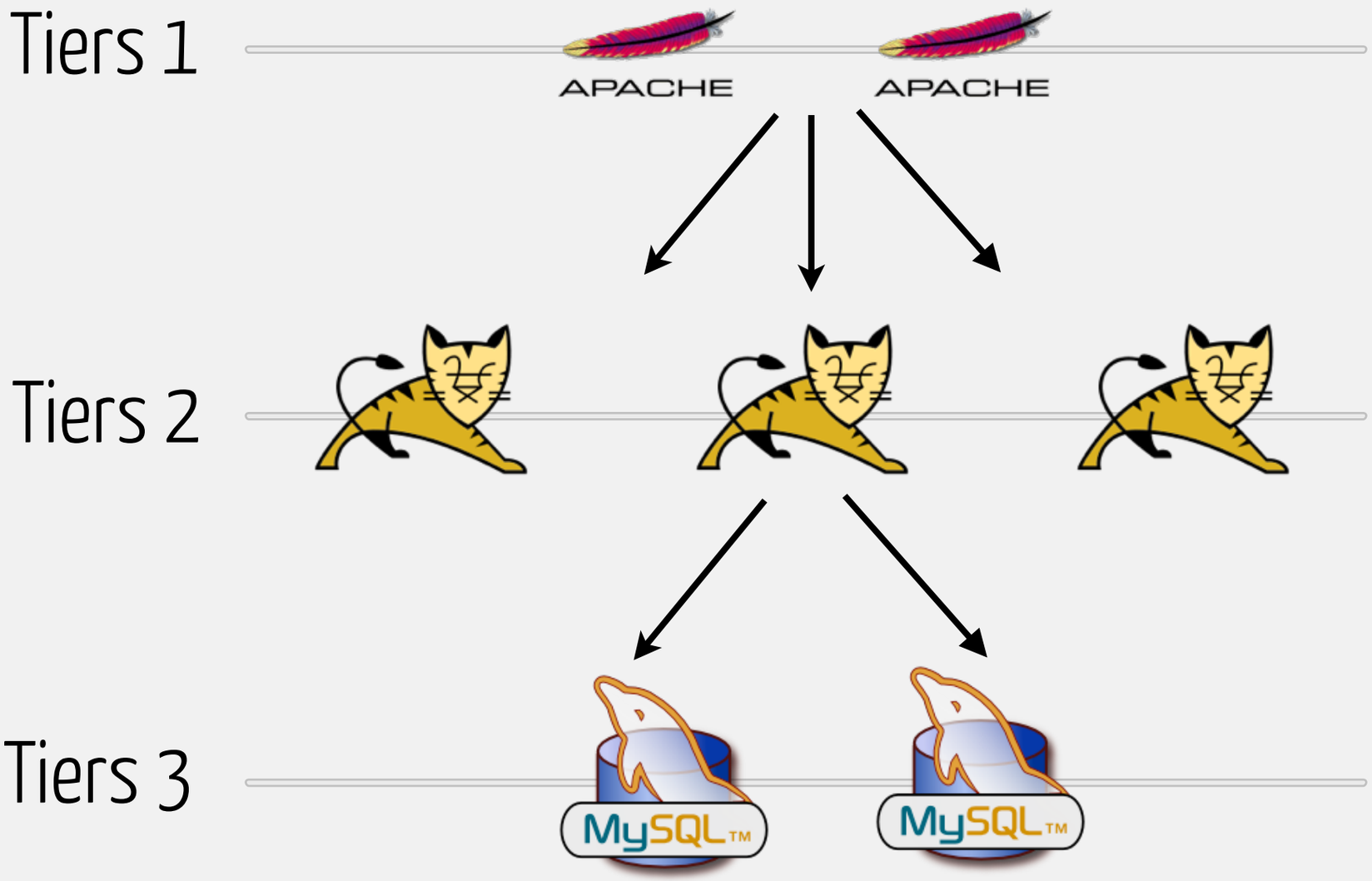
linux virtual server

MariaDB galera

scaling

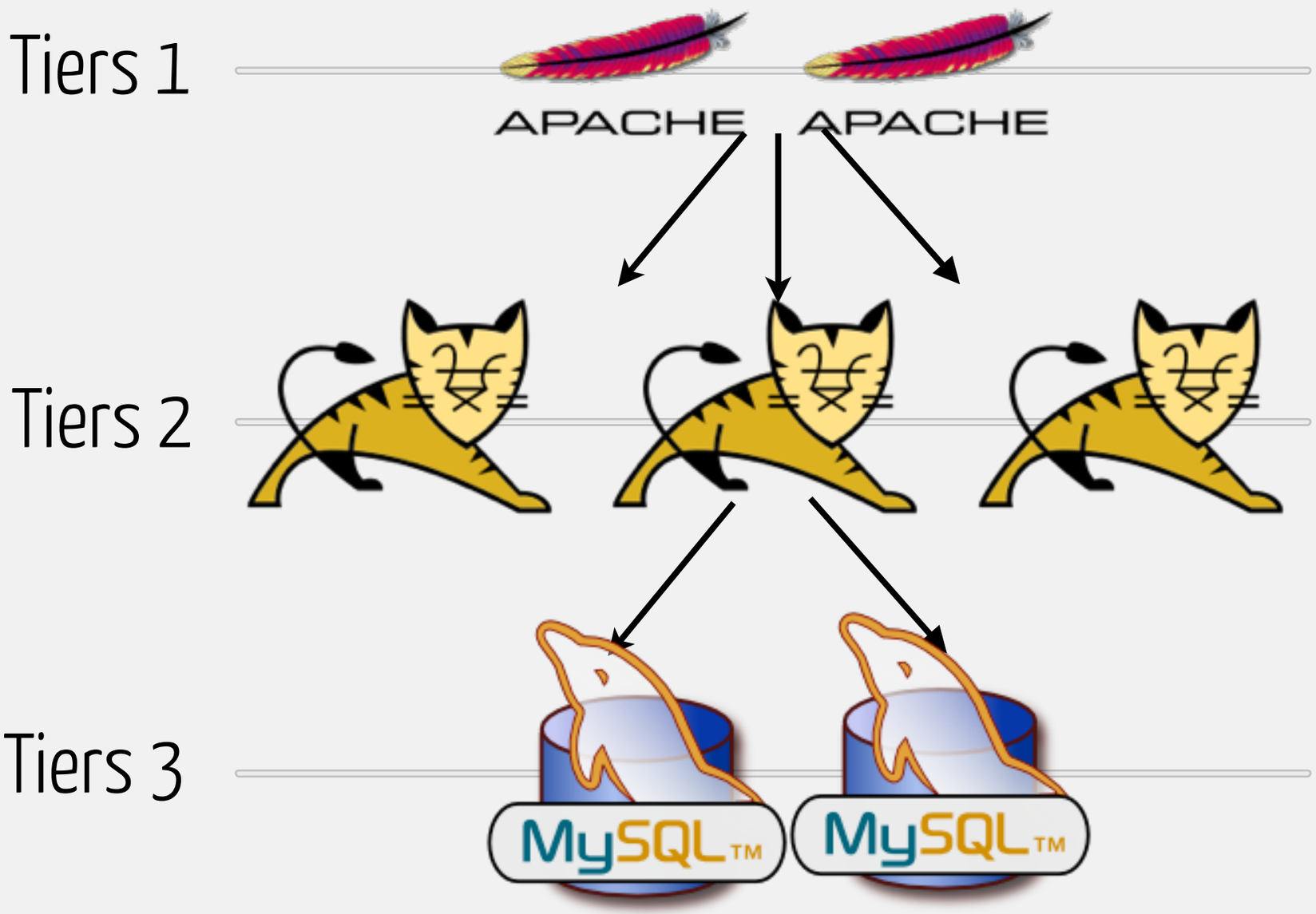
scale up / down

vertical sclaing



scale up / down

vertical sclaing



historical method (more powerful hardware)

mostly cold approaches

easy to implement coldly

hardware bounded

does not address reliability

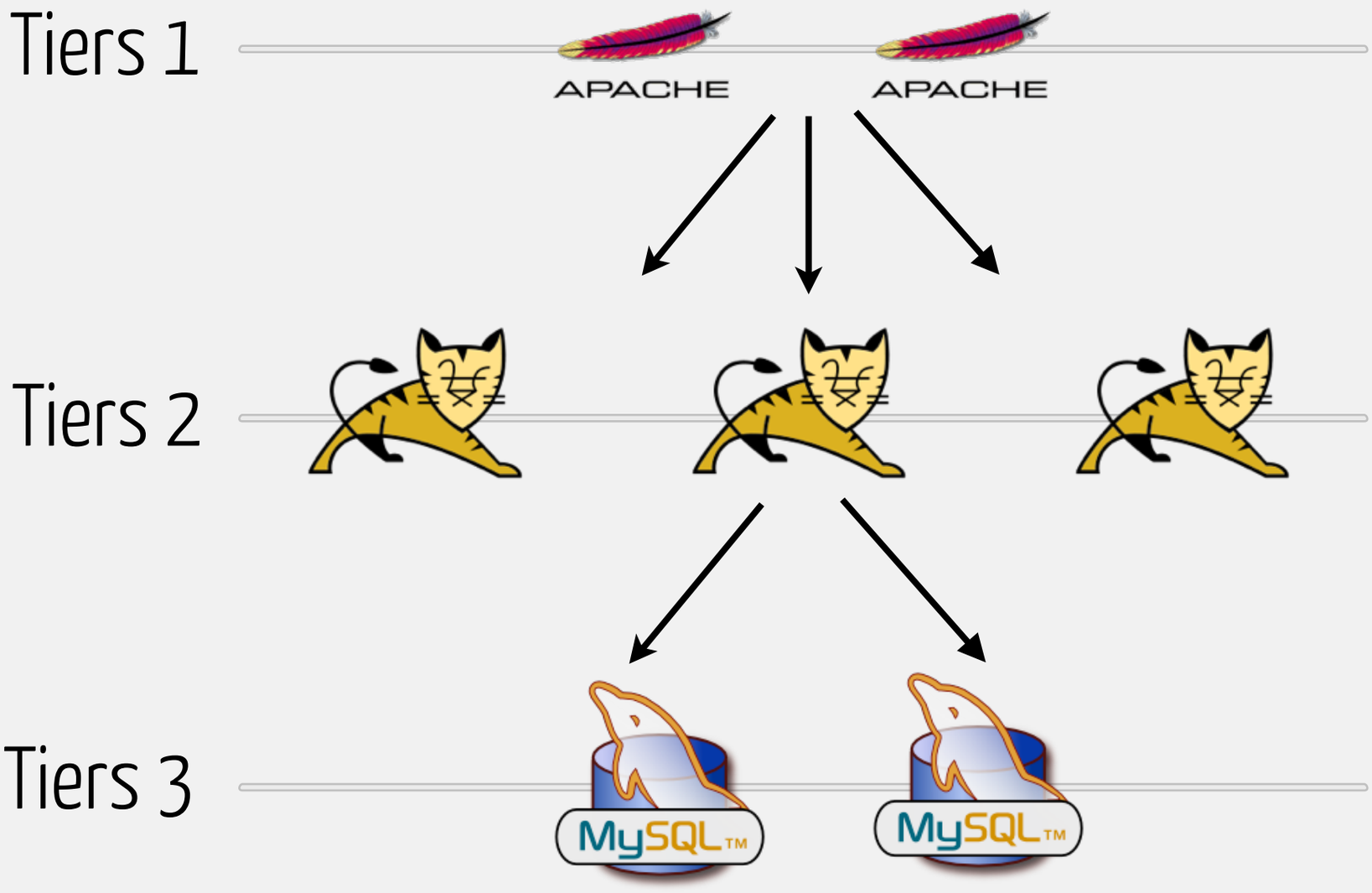
vertical scalin

oVirt support for hot plug CPU

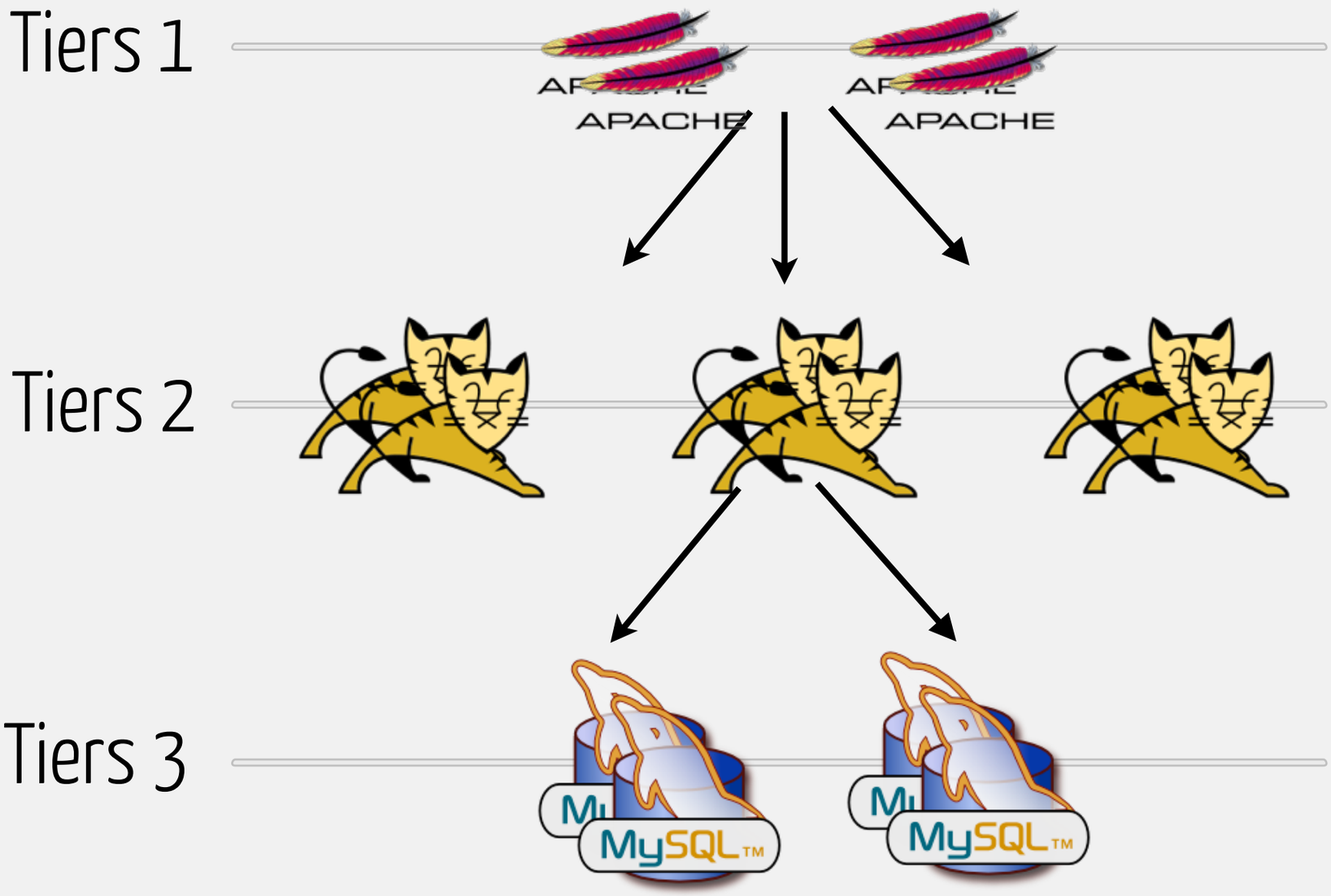
Guest OS Support Matrix

OS	Version	Arch	Plug	Unplug
Red Hat Enterprise Linux 6.3		x86	+	-
Red Hat Enterprise Linux 6.5		x86	+	+
Microsoft Windows Server 2003	All	x86	-	-
Microsoft Windows Server 2003	All	x64	-	-
Microsoft Windows Server 2008	All x86	-	-	
Microsoft Windows Server 2008	Standard, Enterprise	x64	Reboot Required	Reboot Required
Microsoft Windows Server 2008	Datacenter	x64	+	?
Microsoft Windows Server 2008 R2	All	x86	-	-
Microsoft Windows Server 2008 R2	Standard, Enterprise	x64	Reboot Required	Reboot Required
Microsoft Windows Server 2008 R2	Datacenter	x64	+	?
Microsoft Windows Server 2012	All	x64	+	?
Microsoft Windows Server 2012 R2	All	x64	+	?
Microsoft Windows 7	All	x86	-	-
Microsoft Windows 7	Starter, Home, Home Premium, Professional	x64	Reboot Required	Reboot Required
Microsoft Windows 7	Enterprise, Ultimate	x64	+	?
Microsoft Windows 8.x	All	x86	+	?
Microsoft Windows 8.x	All	x64	+	?

what about the runtime ?



horizontal scaling



horizontal scaling

horizontal scaling

not application agnostic — require a load balancer / synchronisation — scale to the infinite in theory — support node failure

Elasticity

optimize performance in live through
scaling requests

static elasticity

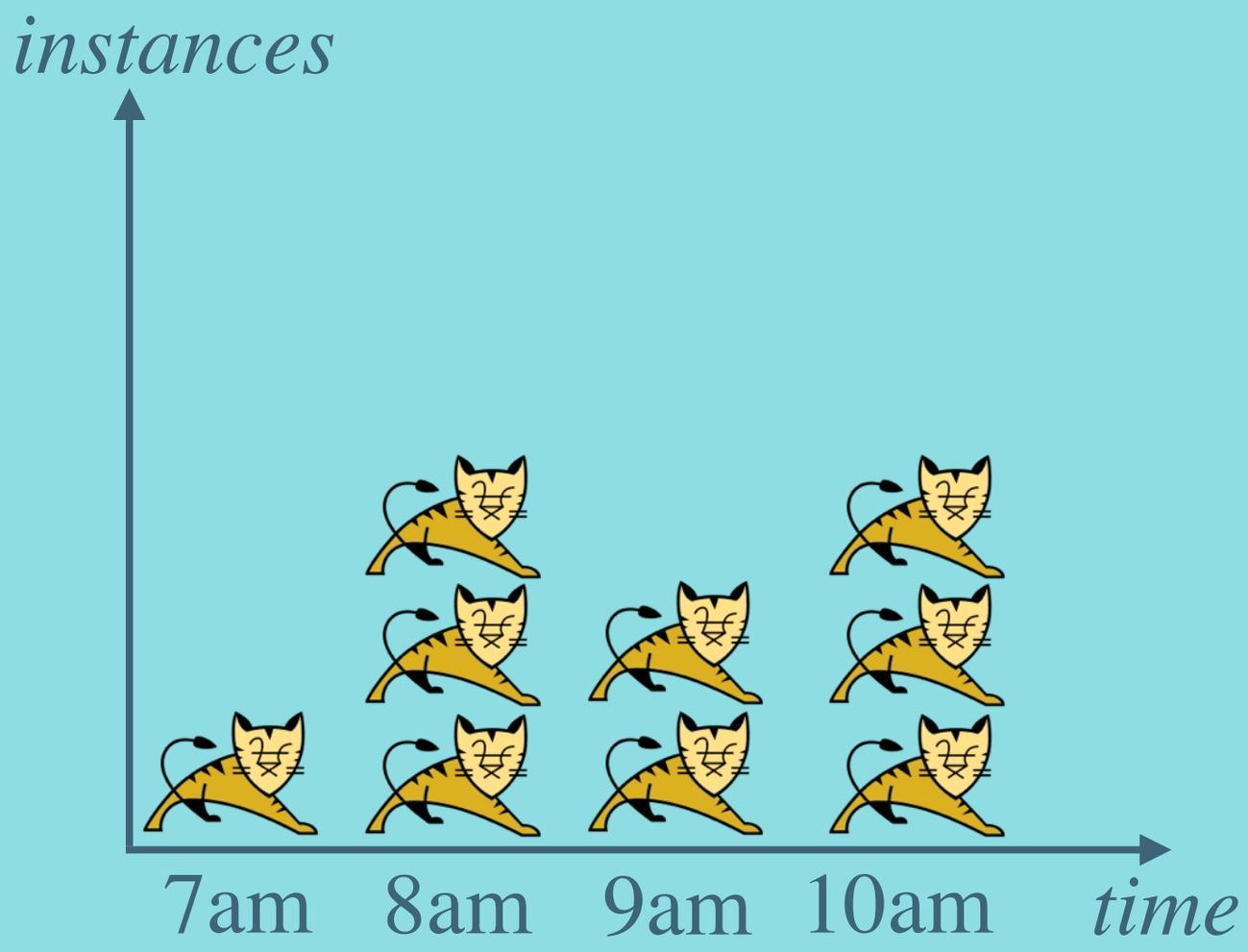
initiated by the administrator

not feedback based

error prone
under/over-estimations

static elasticity

initiated by the administrator



not feedback based

error prone
under/over-estimations

time-driven only ?

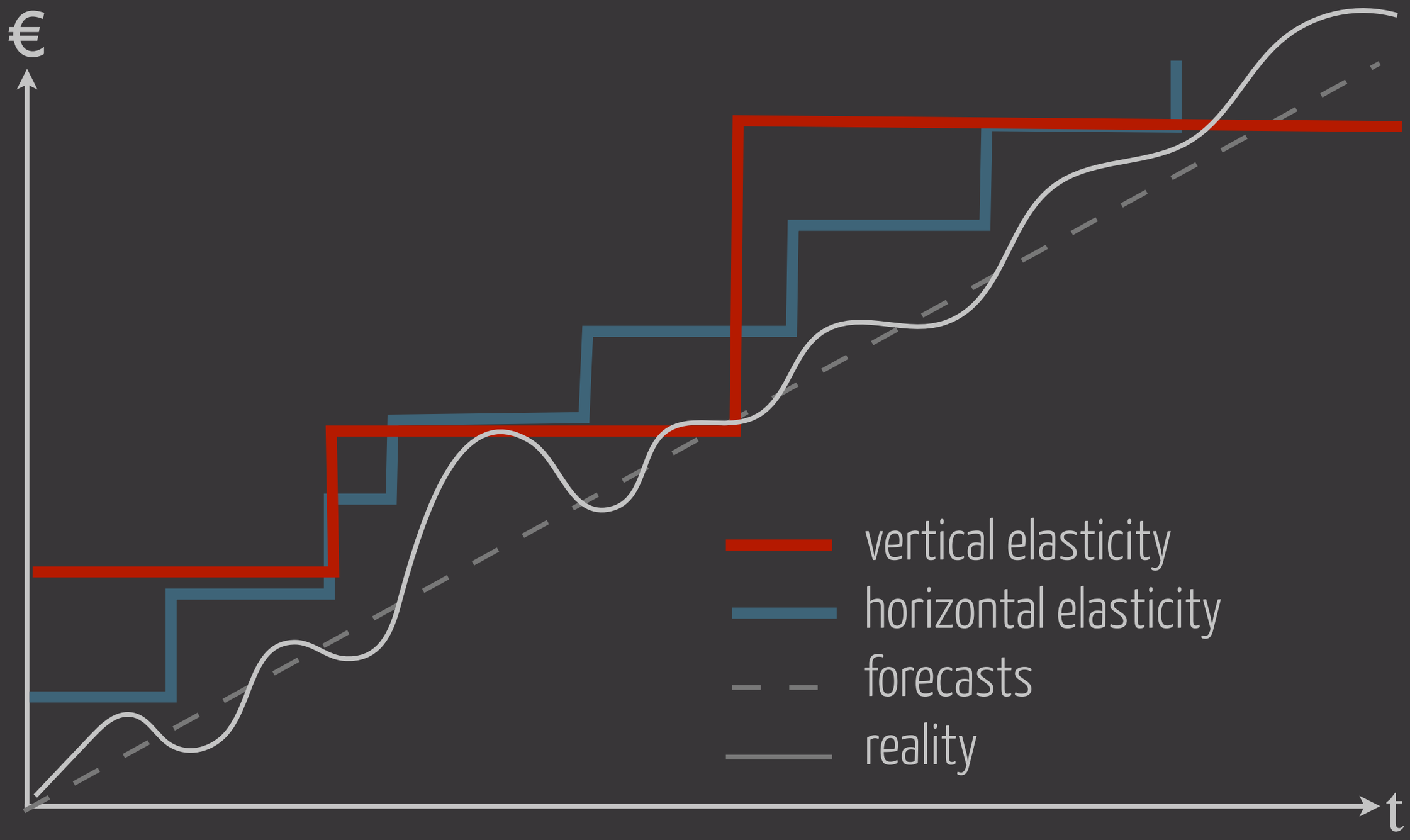
latency-aware elasticity

getting a VM takes up to 5 minutes

think in terms of trends

spare space just in case

static elasticity



dynamic elasticity

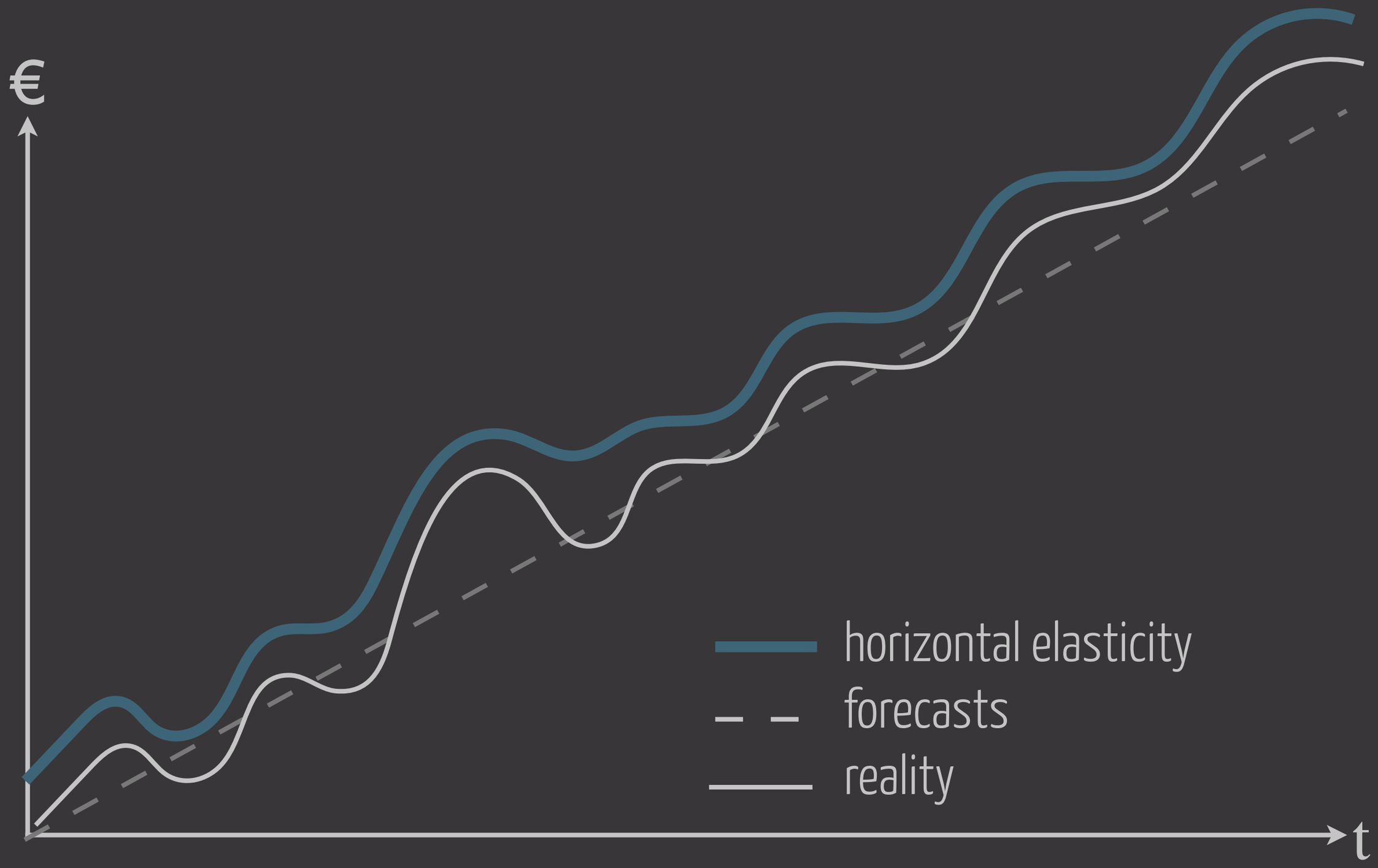
initiated by the app itself

rule based system

feedback from monitoring data

implemented inside/outside the app

dynamic elasticity



Filter: Filter Auto Scaling groups... 1 to 2 of 2 Auto Scaling Groups

Name	Launch Configuration	Instances	Desired	Min	Max	Availability Zones	Default Cc
awseb-e-dnma...	awseb-e-dnmaa76xme-...	1	1	1	4	eu-west-1a, eu-west-1b, eu-...	360
WWWELB	my	1	1	1	10	eu-west-1b	100

Decrease Group Size

Actions

Execute policy when: awsec2-WWWELB-High-CPU-Utilization
breaches the alarm threshold: CPUUtilization < 20 for 300 seconds
for the metric dimensions AutoScalingGroupName = WWWELB

Take the action: Remove 1 instances

And then wait: 100 seconds before allowing another scaling activity

Increase Group Size

Actions

Execute policy when: awsec2-WWWELB-CPU-Utilization
breaches the alarm threshold: CPUUtilization >= 40 for 300 seconds
for the metric dimensions AutoScalingGroupName = WWWELB

Take the action: Add 1 instances

And then wait: 100 seconds before allowing another scaling activity

dynamic elasticity

scale where its matter

monitor each tier to indentify the bottlenecks

scale out apache/tomcat/mysql ?

cost model for elasticity

service cost

instance cost (hourly based)

/!\ don't scale too often

RECAP

CLOUD IS ABOUT

CLOUD IS ABOUT SCALABILITY

CLOUD IS ABOUT RESILIENCY

CLOUD IS ABOUT TRUST