

Web Infrastructures

RES, Lecture 8

Olivier Liechti

heig-vd

Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

Systemic Qualities

The Notion of Systemic Quality

- How do we specify the **requirements** of a system?
 - We always build systems to deliver some sort of "**service**" (web, messaging, access control, customer management, etc.).
 - We first have to specify "**what**" the system should do. In other words, we have to specify **functional requirements**.
 - We also have to specify "**how**" the system should behave, i.e. what qualities it should exhibit. These are the **non-functional requirements**.
- Non-functional requirements characterize **systemic qualities, or "-ilities"**:
 - There are lots of different systemic qualities. Depending on the system, some are more important than others.
 - Choices made when defining the system architecture have a large impact on the systemic qualities.
 - Your life as an architect will be to deal with **trade-offs** in addressing conflicting systemic qualities.

System
Vehicle

Functional requirements
Move people around

Non-functional requirements
Performance
Capacity
Reliability
Cost
Aesthetics
Ease of use



Different systemic qualities often create **opposing forces**.

Defining the “right” architecture for a system means finding the right **balance** between these forces.



Some Systemic Qualities...

- **Response time**
 - Measures the time required to present a result to the user
 - Important for the end-user
- **Throughput**
 - Measures the number of requests that can be processed in a given time frame
 - Important for the service provier
- **Scalability**
 - Measures how easy/costly it is to adapt the system in order to handle additional load
 - Ideally: linear scalability. "2 x more users => 2 x more servers"
- **Availability**
 - Measures the percentage of time during which the system can be used
 - 99% availability = average unavailability of 3.65 jours per year, i.e 1 hour and 41 minutes per week.
- **Security**
- **Manageability**
 - Measures the ease of operating the system: how easy is it to monitor it, to detect issues, to upgrade, etc.

Specifying/Measuring Qualities

- **Performance**

- Response time in ms
- Throughput in requests/seconds

- **Scalability**

- Can we sustain a bigger load with additional resources? How much will it **cost** and how much **time** do we need?

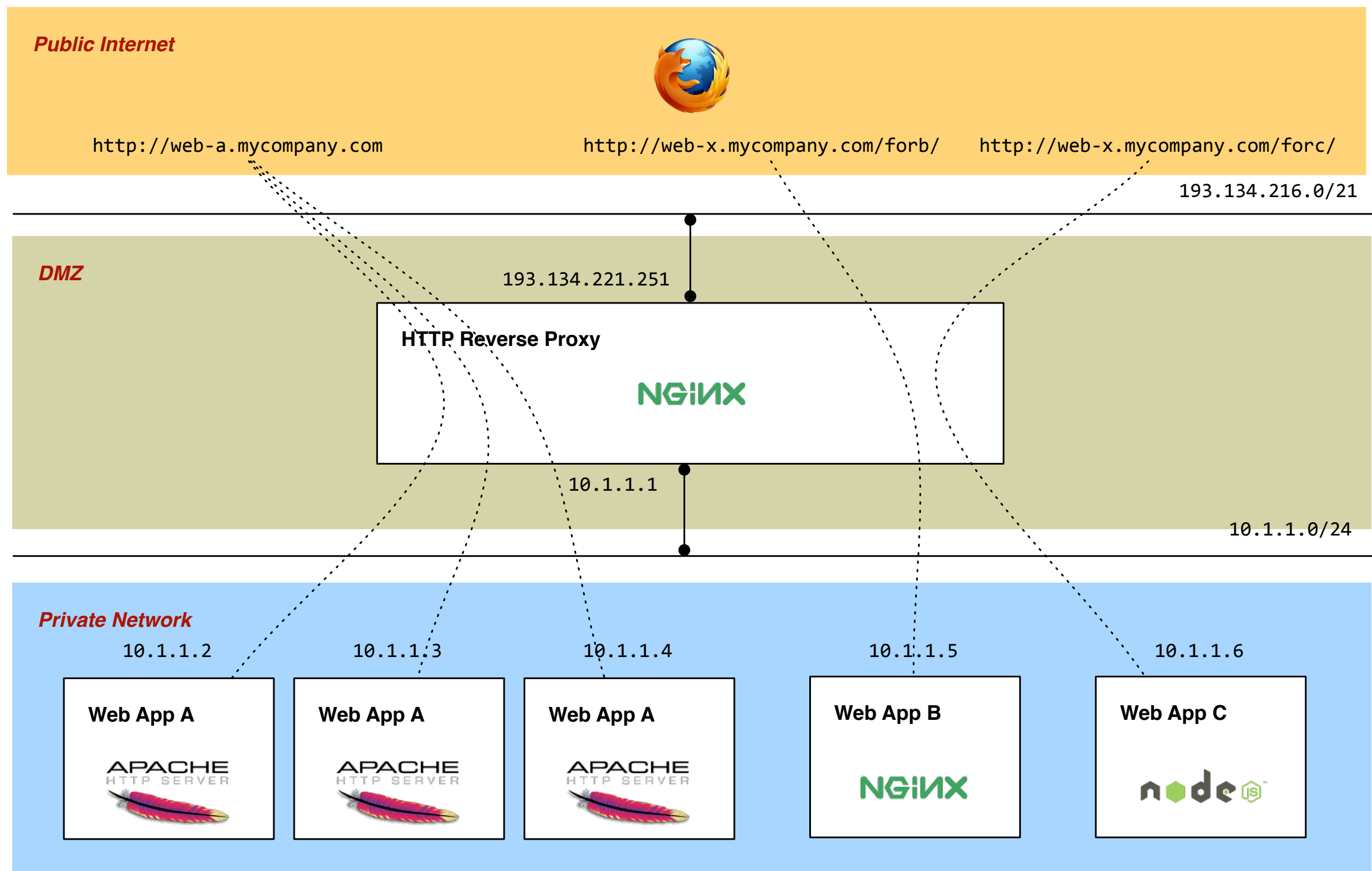
- **Availability**

- Over a given period
- **Percentage**, ration between the time when the service was available over the total time
- Towards the “**five nines**” (99.999% uptime)
- $\text{Availability} = \text{Mean Time Between Failures} / (\text{Mean Time Between Failures} + \text{Mean Time To Repair})$



If this is your web “infrastructure”, what are its systemic qualities?

HTTP Infrastructure



The Role of the Reverse Proxy

- An HTTP proxy that is “close” to the server
- Forwarding requests to the “appropriate” server
- Balancing requests between several “equivalent” servers (load balancing)
- **Sticky sessions!**

```
ProxyRequests Off
```

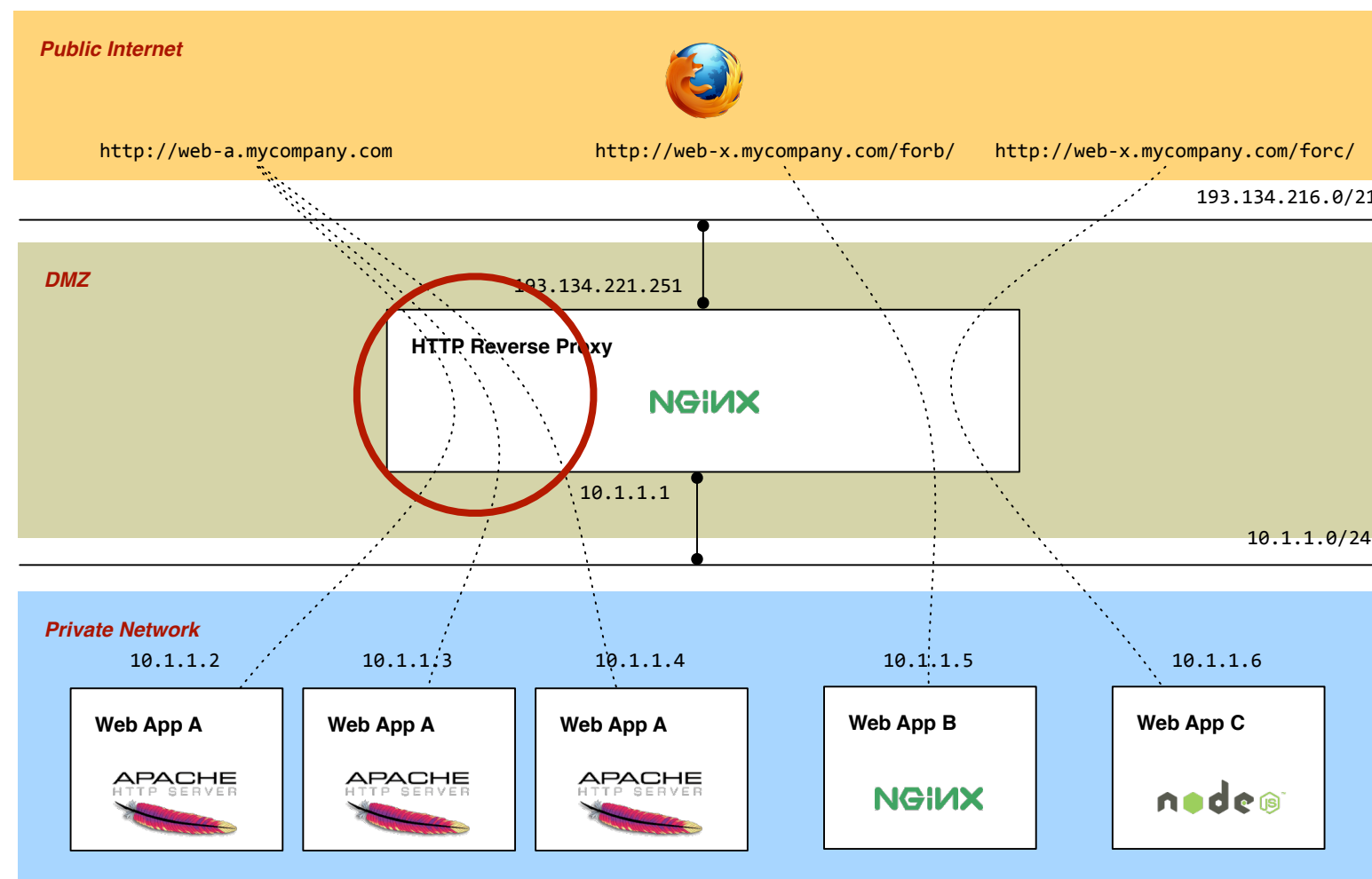
```
<Proxy *>  
Order deny,allow  
Allow from all  
</Proxy>
```

```
ProxyPass /foo http://192.168.1.2:8080/bar  
ProxyPassReverse /foo http://192.168.1.2:8080/bar
```

```
Header add Set-Cookie "ROUTEID=.%{BALANCER_WORKER_ROUTE}e;  
path=/" env=BALANCER_ROUTE_CHANGED  
<Proxy balancer://mycluster>  
BalancerMember http://192.168.1.50:80 route=1  
BalancerMember http://192.168.1.51:80 route=2  
ProxySet stickysession=ROUTEID  
</Proxy>  
ProxyPass /test balancer://mycluster
```

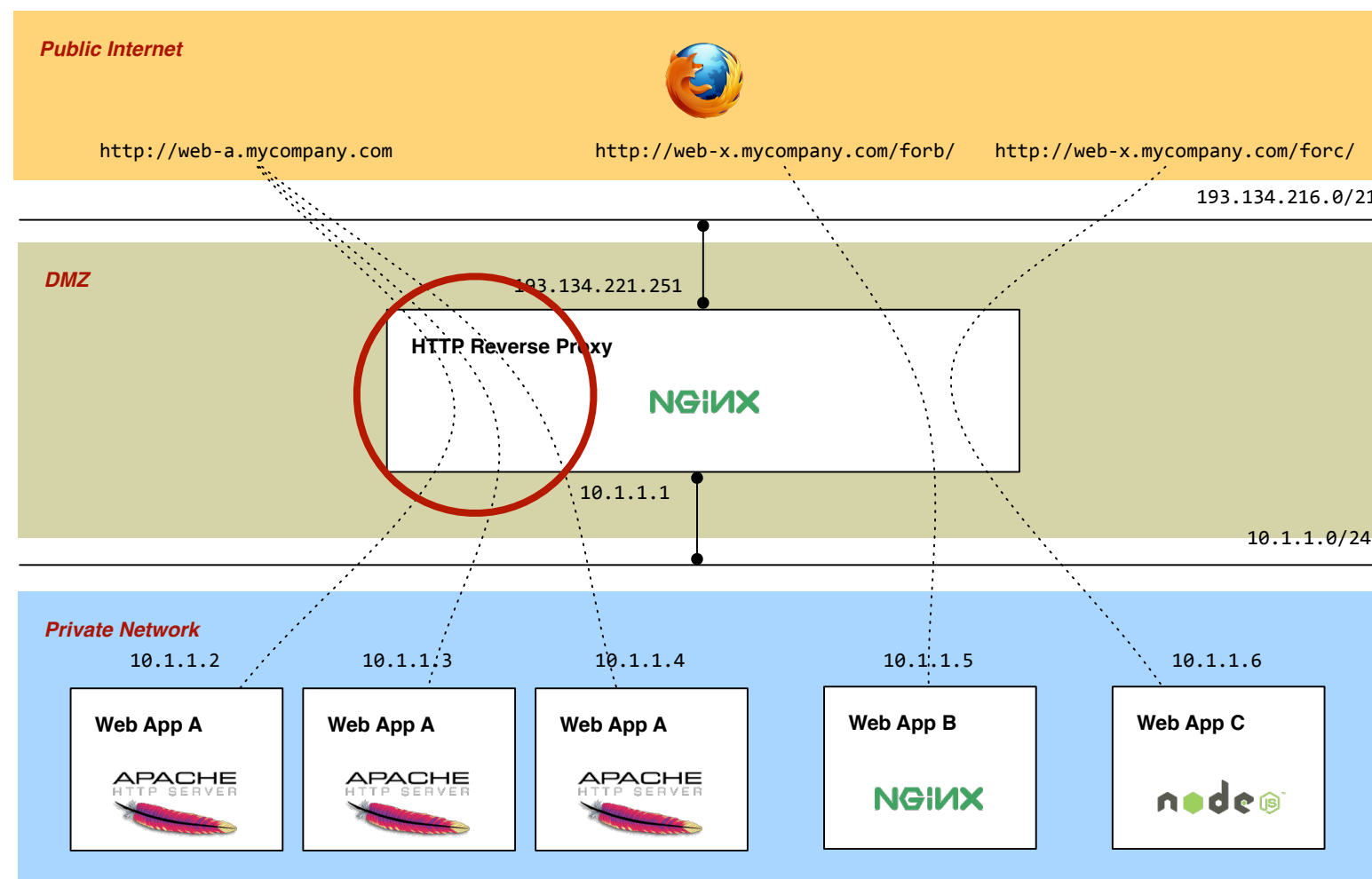

Availability

- What happens if a server (or server component) fails?

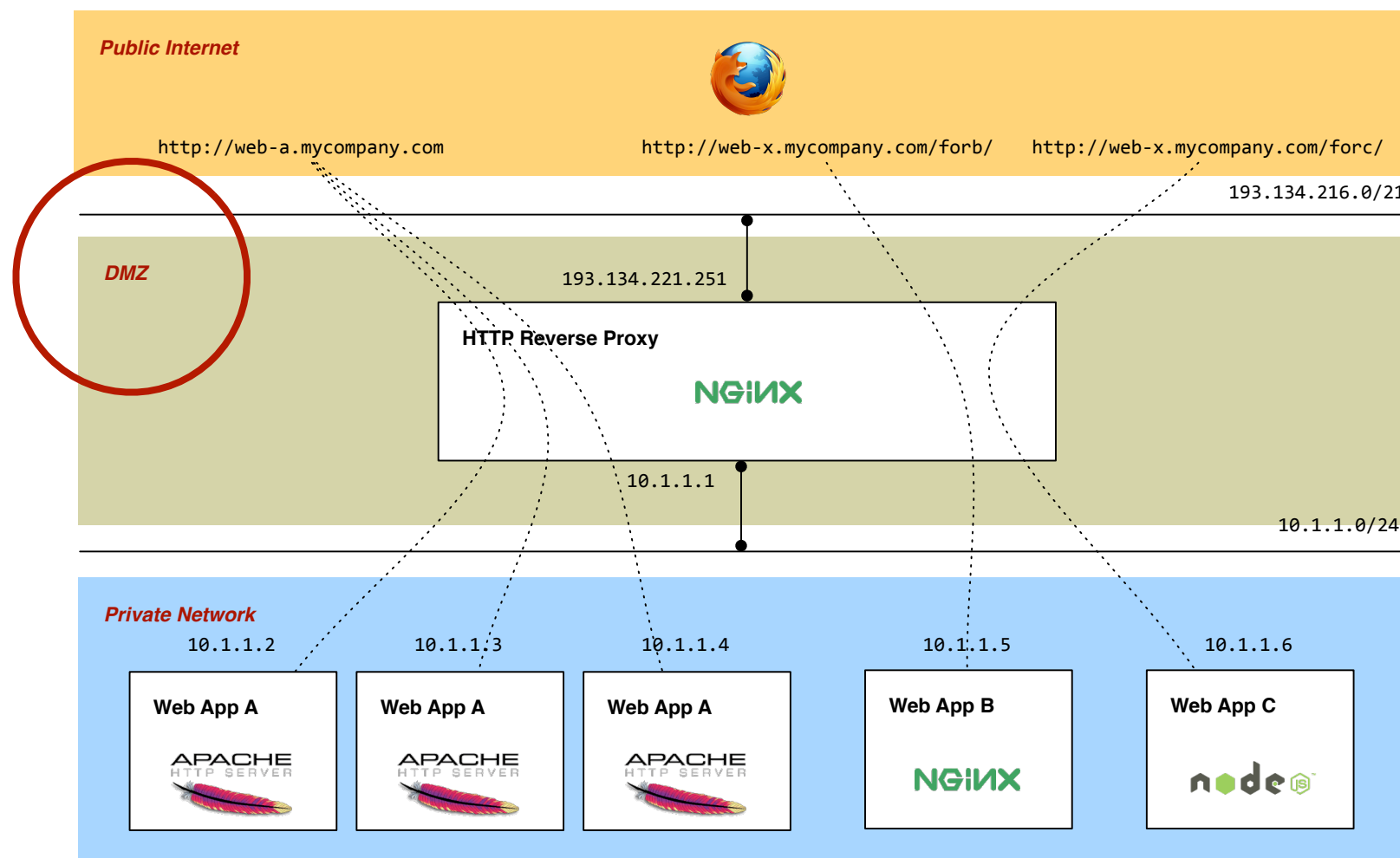


Scalability

- Ability to evolve in order to sustain a bigger load, in quick and economical manner.
- Horizontal vs Vertical Scalability
- Elasticity



Security



Virtualization and Provisioning Technologies

Run a guest OS on top of your OS



<https://www.virtualbox.org/>

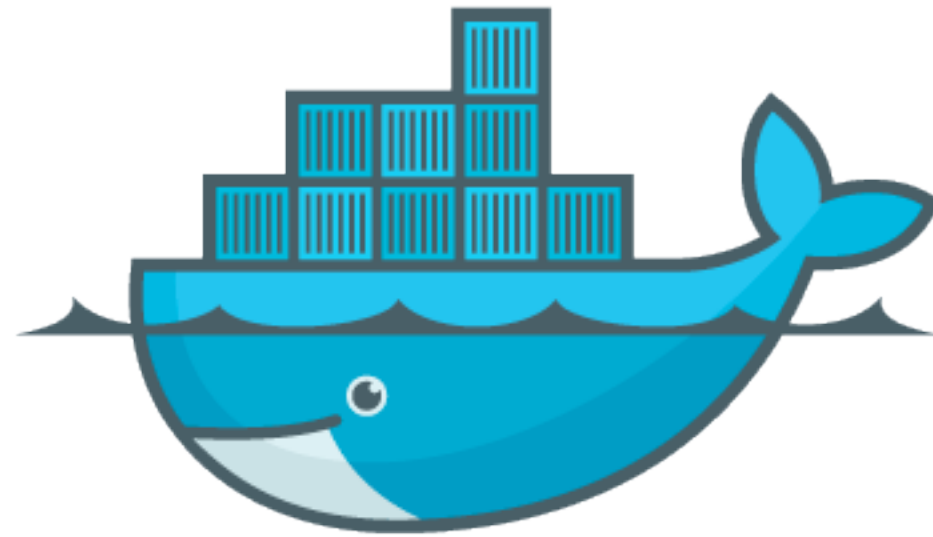
Manage your VMs in a reproducible way



VAGRANT

<http://www.vagrantup.com/>

*Run lightweight containers on top of your
Linux OS*



docker

<https://www.docker.io/>



Vagrant controls Virtual Box



Virtual Box runs a box on which Docker has been installed and is running

