Reconstructing Netflix

Prof. Yair Amir

Raghuram SV, Aditya Rao, Kunal Lillaney

600.667 Advanced Distributed Systems & Communication
Johns Hopkins University

Netflix Facts

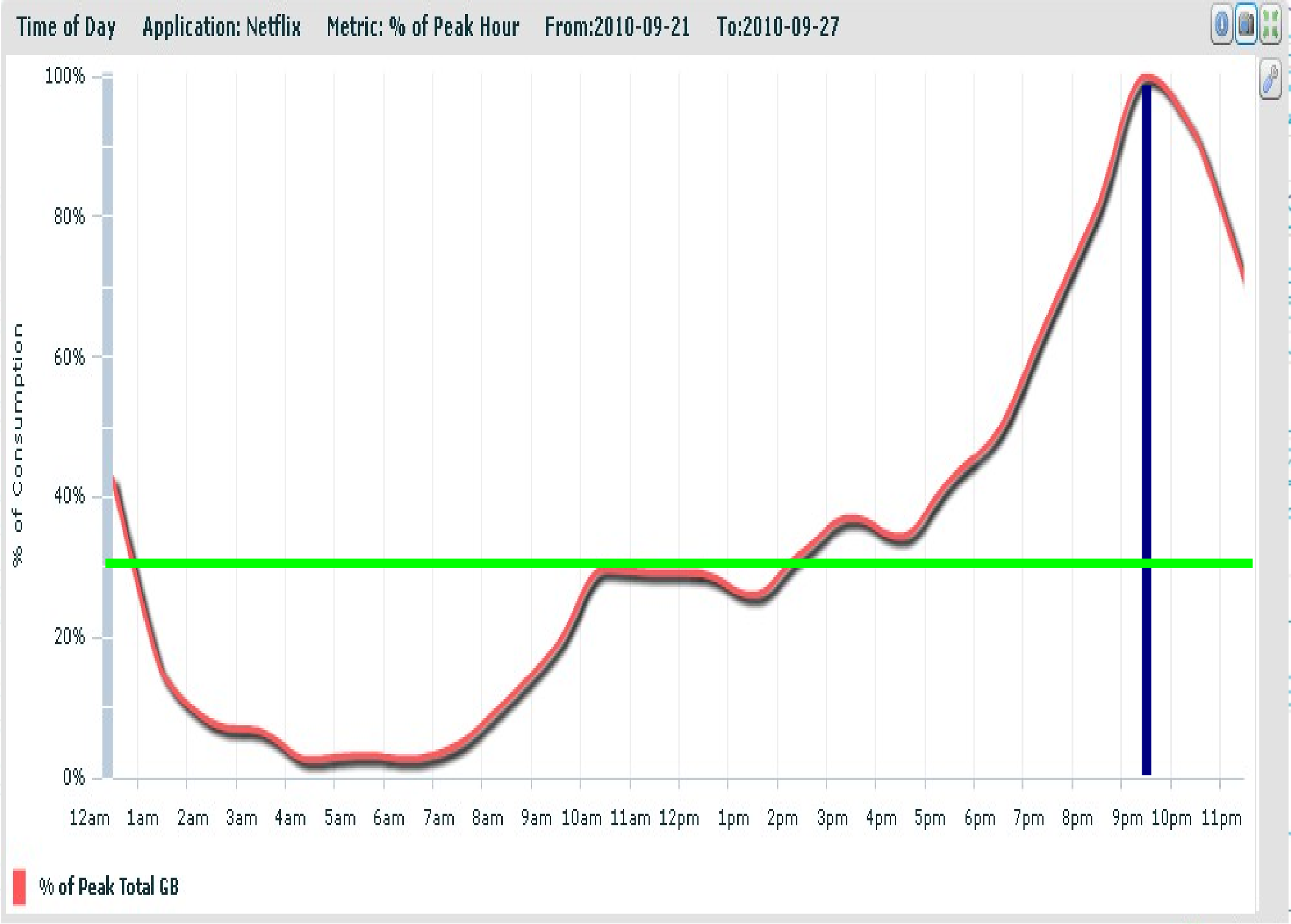
- Internet has overtaken cable TV as preferred medium for delivering video content
- Netflix, in 2012, ~30 million subscribers ~30% downstream traffic in the US.
- Exclusively uses Amazon AWS for storage,web-services and CDN's for delivery
- Revenue of $3.2B , Profit of $200M = $3B Expenses

Our Project

- What if we build our own system?
- Would it be easier to rent?
- What technologies would we require?
- What challenges would we face given the huge number of users?
- How would this system look?

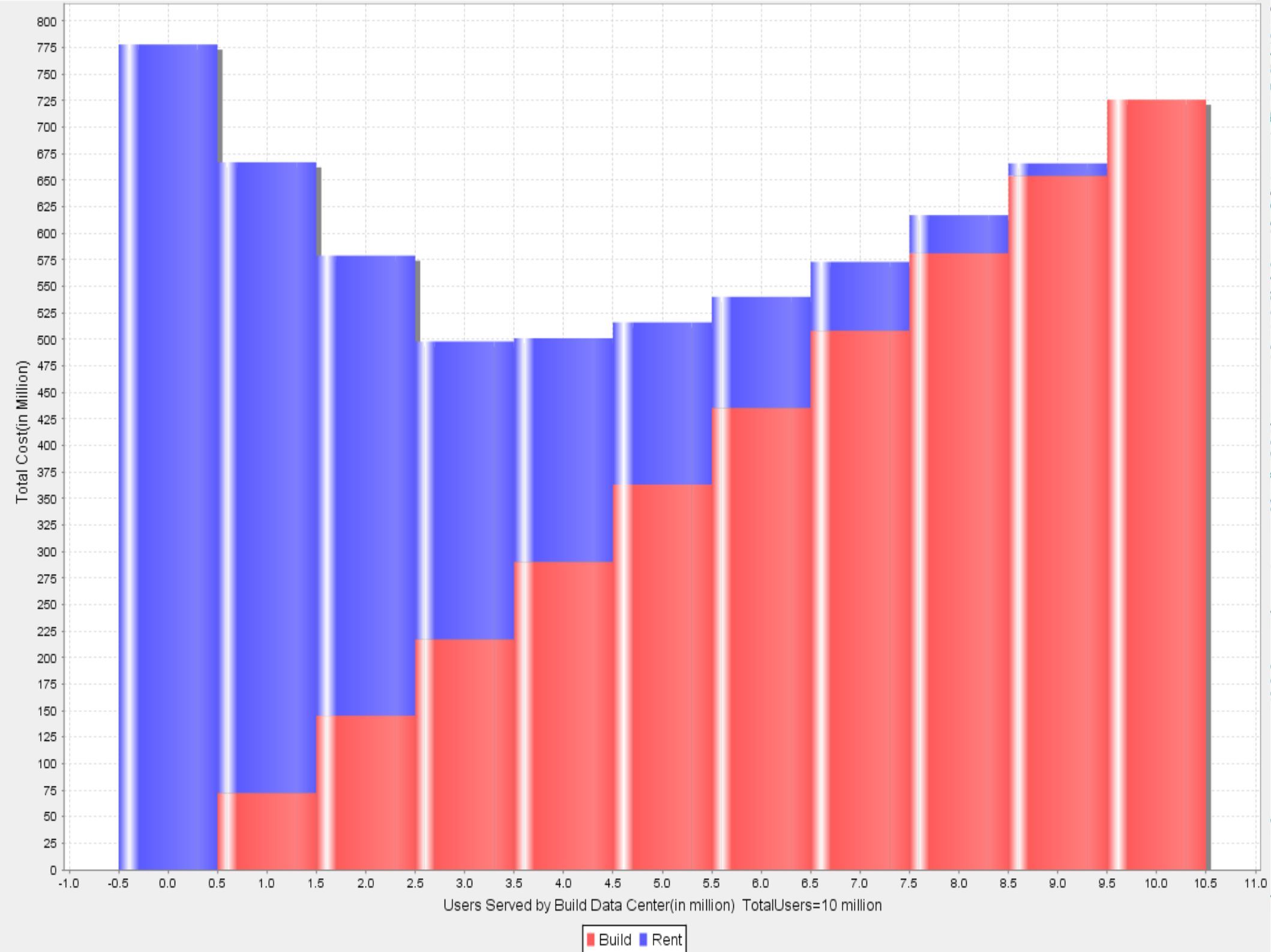
Demystifying the Cost

- Proposed system with 30 million subscribers
- Average User – 80 hours/month
- 1.5Mbps for each user (HD 1080p 24fps)
- Peak of 10 million users
- What is the best way to go?

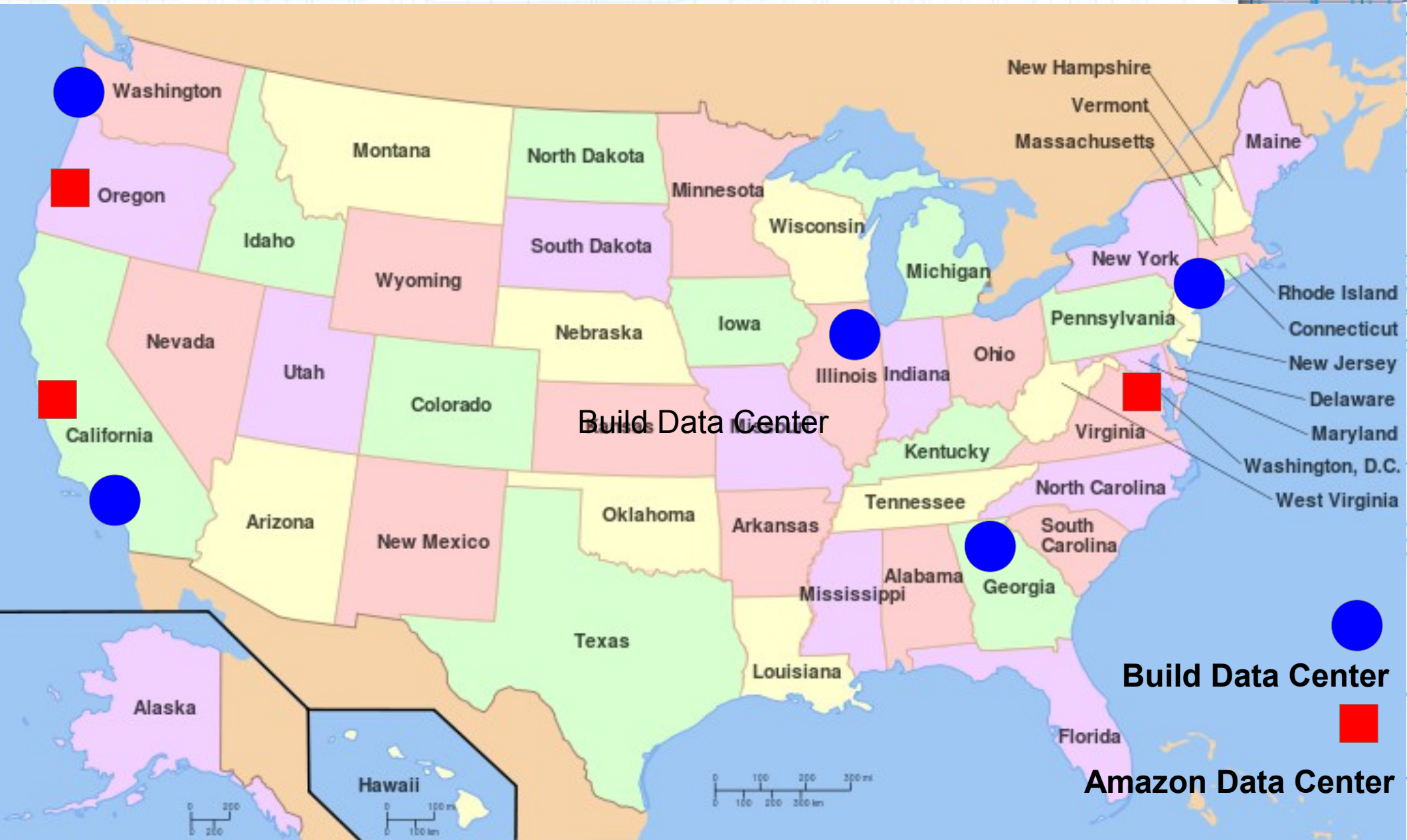

Time of Day
Application: Netflix
Metric: % of Peak Hour
From:2010-09-21
To:2010-09-27
100%
80%
60%
40%
20%
0%
% of Consumption
12am
1am
2am
3am
4am
5am
6am
7am
8am
9am
10am
11am
12pm
1pm
2pm
3pm
4pm
5pm
6pm
7pm
8pm
9pm
10pm
11pm
% of Peak Total GB

Assumptions we made

- Machine Buy + Maintenance for 5 years - $3000
- 1.5Mbps dedicated line - $6/month
- Amazon 1GB data cost - $0.04
- Move reserved instances across Amazon data centers
- Look at the graph and play with the values

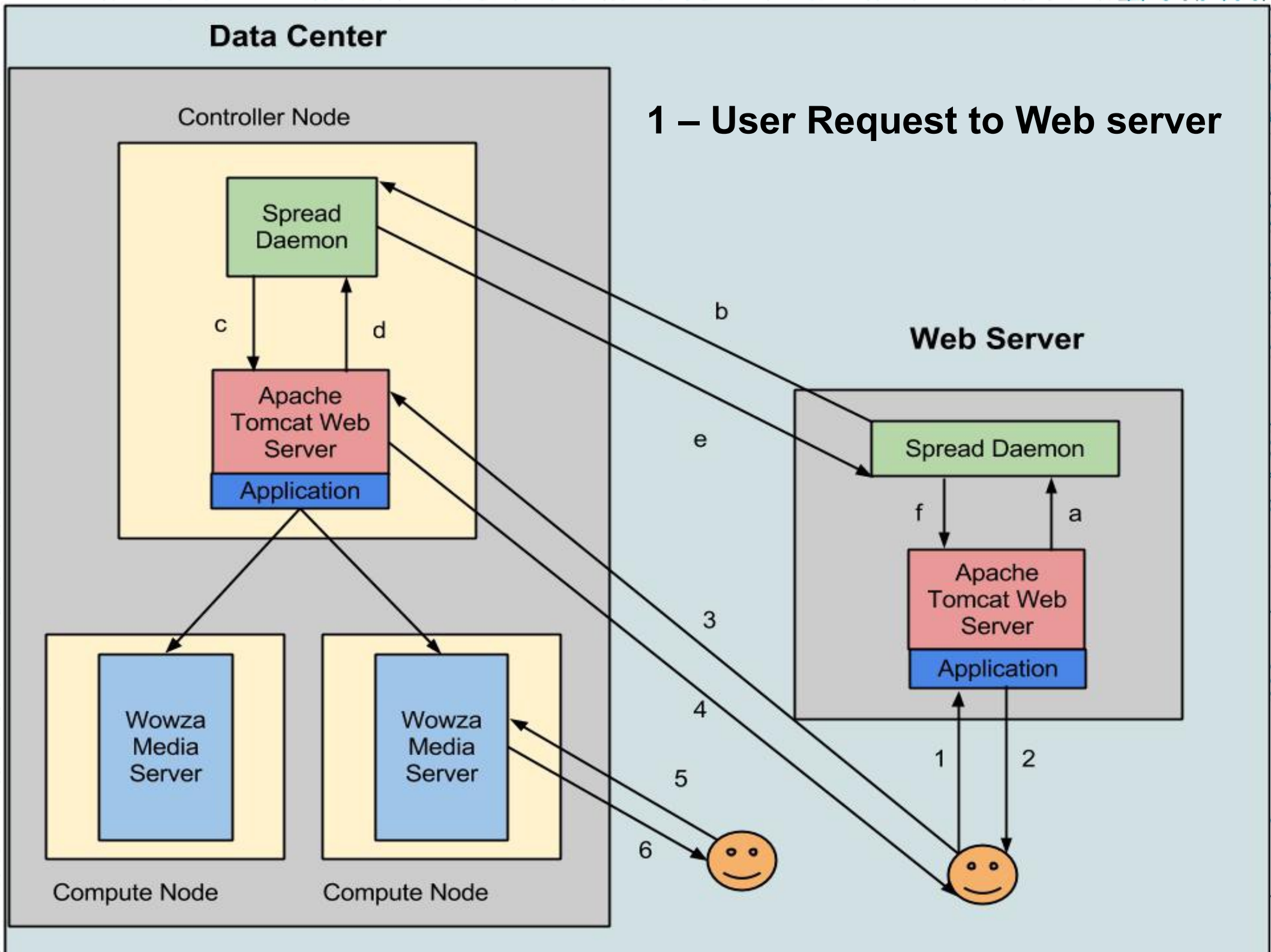

Total Cost(in Million)
800
775
750
725
700
675
650
625
600
575
550
525
500
475
450
425
400
375
350
325
300
275
250
225
200
175
150
125
100
75
50
25
0
-1.0
-0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
8.5
9.0
9.5
10.0
10.5
11.0
Users Served by Build Data Center(in million) TotalUsers=10 million
Build
Rent

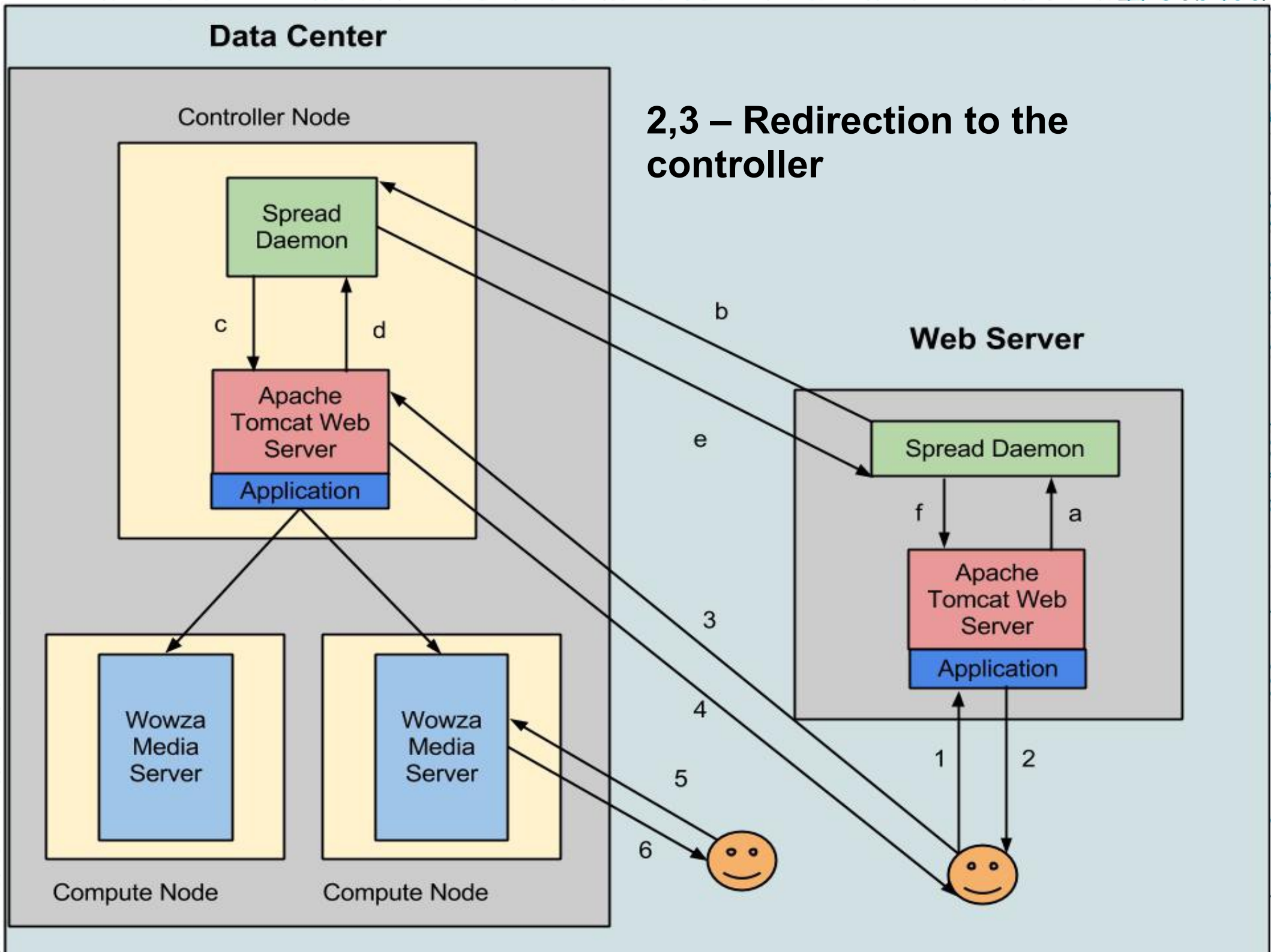
How we would deploy it

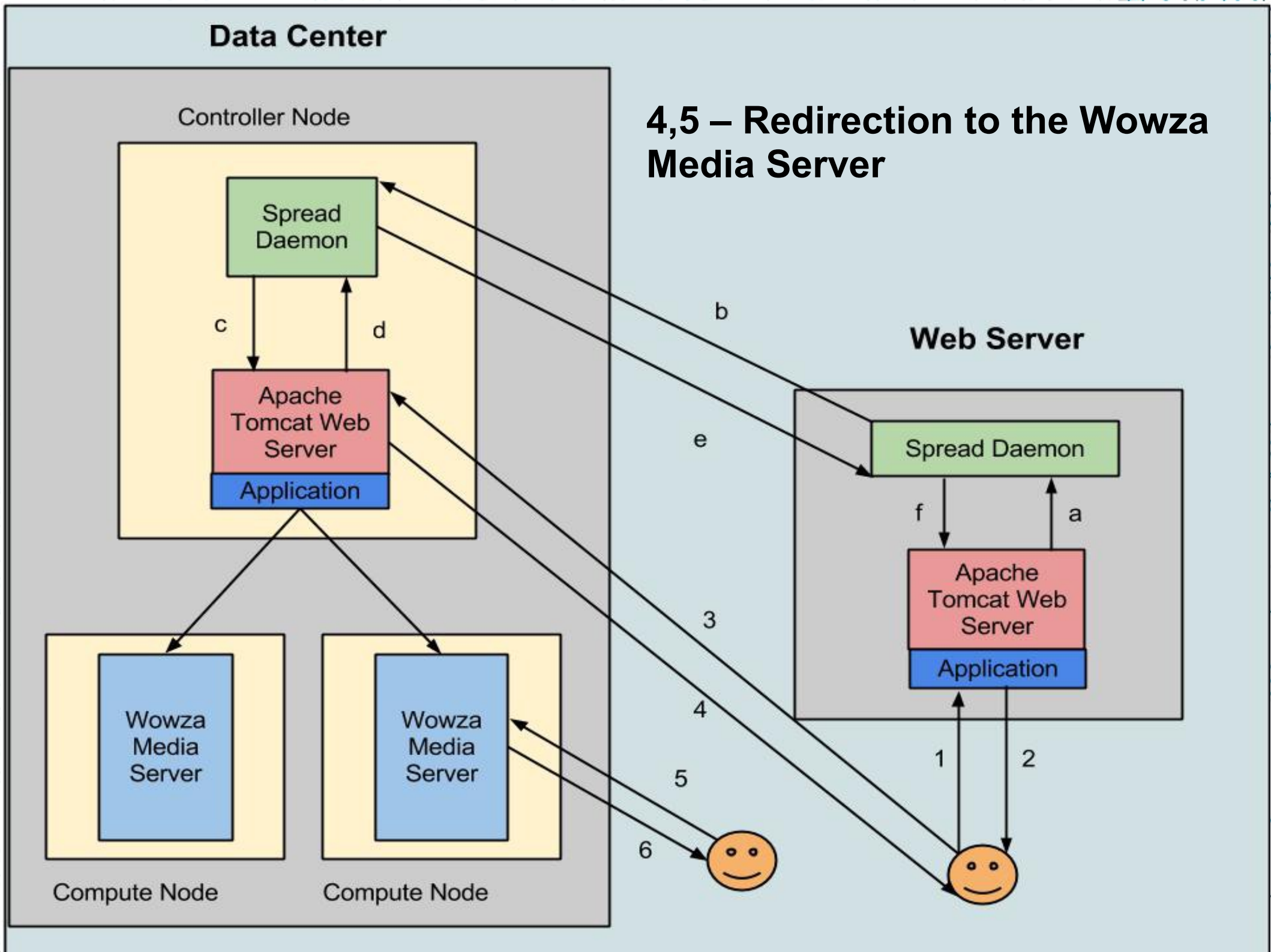


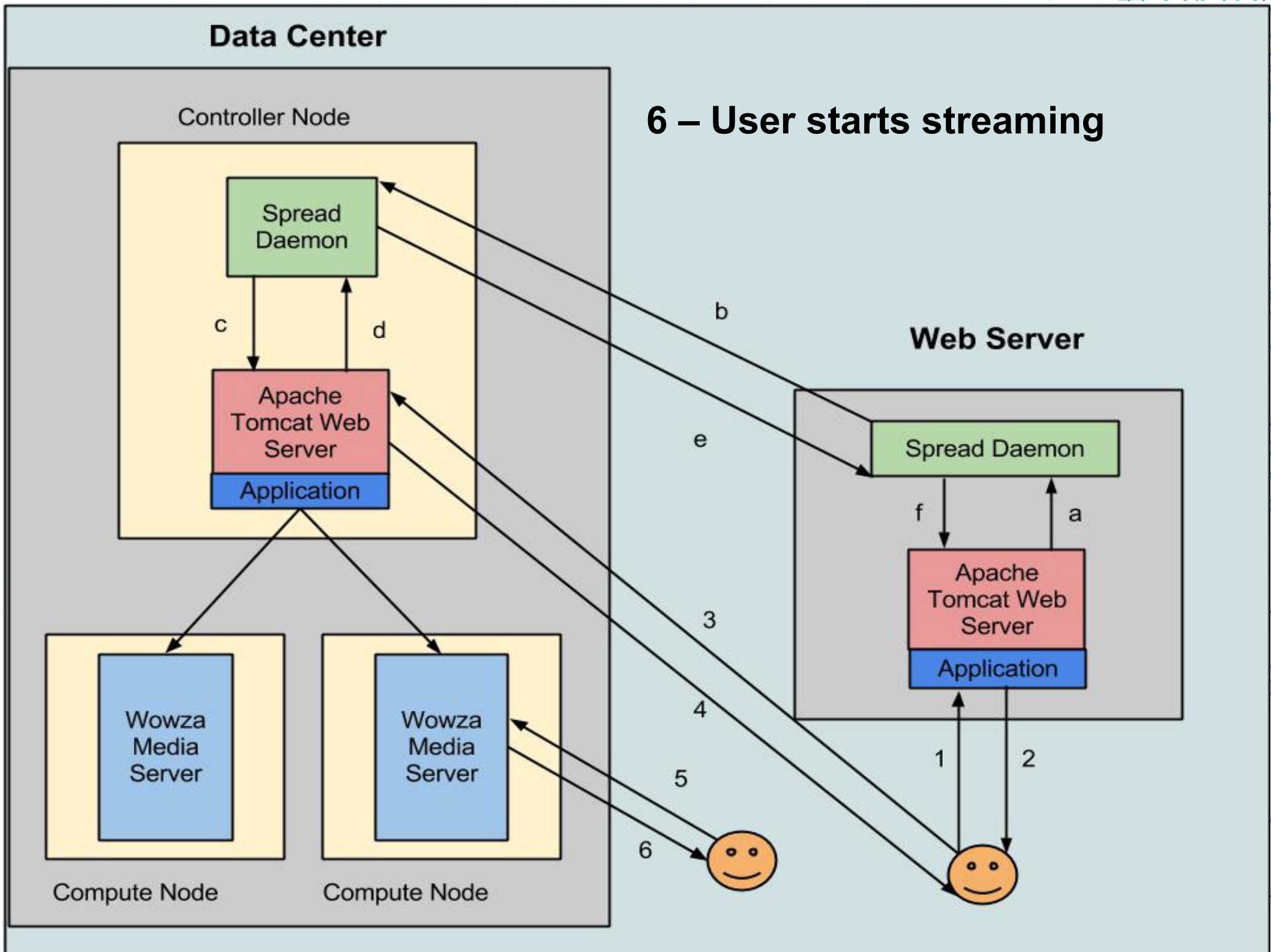
System Architecture

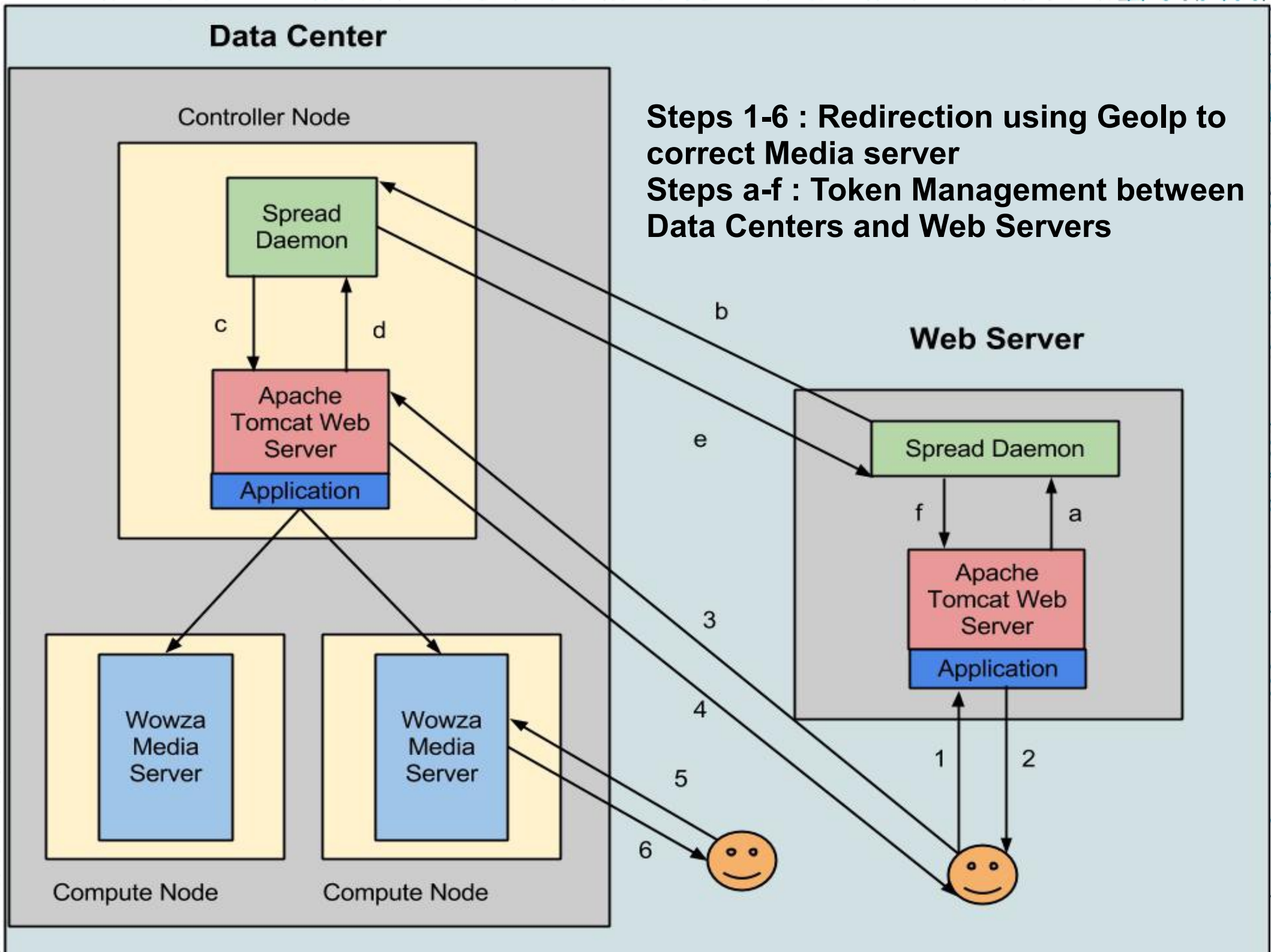
- Two Components – Web Server & Data Center
- Data Center – Build and Rent


Data Center
Controller Node
Spread Daemon
c
d
Apache Tomcat Web Server
Application
b
e
Web Server
Spread Daemon
f
a
Apache Tomcat Web Server
Application
3
4
5
6
1
2
Wowza Media Server
Wowza Media Server
Compute Node
Compute Node
1 – User Request to Web server


Data Center
Controller Node
Spread Daemon
c
d
Apache Tomcat Web Server
Application
Wowza Media Server
Wowza Media Server
Compute Node
Compute Node
2,3 – Redirection to the controller
b
e
Web Server
Spread Daemon
f
a
Apache Tomcat Web Server
Application
3
4
5
6
1
2


Data Center
Controller Node
Spread Daemon
c
d
Apache Tomcat Web Server
Application
Wowza Media Server
Wowza Media Server
Compute Node
Compute Node
4,5 – Redirection to the Wowza Media Server
b
Web Server
e
Spread Daemon
f
a
Apache Tomcat Web Server
Application
3
4
1
2
5
6


Data Center
Controller Node
Spread Daemon
c
d
Apache Tomcat Web Server
Application
Wowza Media Server
Wowza Media Server
Compute Node
Compute Node
6 – User starts streaming
b
Web Server
e
Spread Daemon
f
a
Apache Tomcat Web Server
Application
3
4
5
6
1
2


Data Center
Controller Node
Spread Daemon
c
d
Apache Tomcat Web Server
Application
Wowza Media Server
Wowza Media Server
Compute Node
Compute Node
Steps 1-6 : Redirection using Geolp to correct Media server
Steps a-f : Token Management between Data Centers and Web Servers
b
Web Server
e
Spread Daemon
f
a
Apache Tomcat Web Server
Application
3
4
5
6
1
2

GeoIP for correct Datacenter

- Latitude & Longitude for IP
- Euclidean Distance Formula
- Redirect to the closest build data centers
- If build data centers are unavailable, switch to closest rented one

Token for Resource Allocation

Data center

- Each data center has a token pool
- Datacenter has a token directory for each web server
- The allocation is done using first come first serve policy
- Allocation is done in sizes called “batch count”
- Allocates whatever it has if the leftover is less than batch count

Token for Resource Allocation

Web Server

- Request for tokens made in sizes of batch count
- Request a new batch of tokens when about to be exhausted.
- Periodically sends a token request when all tokens are exhausted
- Redirects to a different data center if all tokens are exhausted for one

Token expiry at Data Center

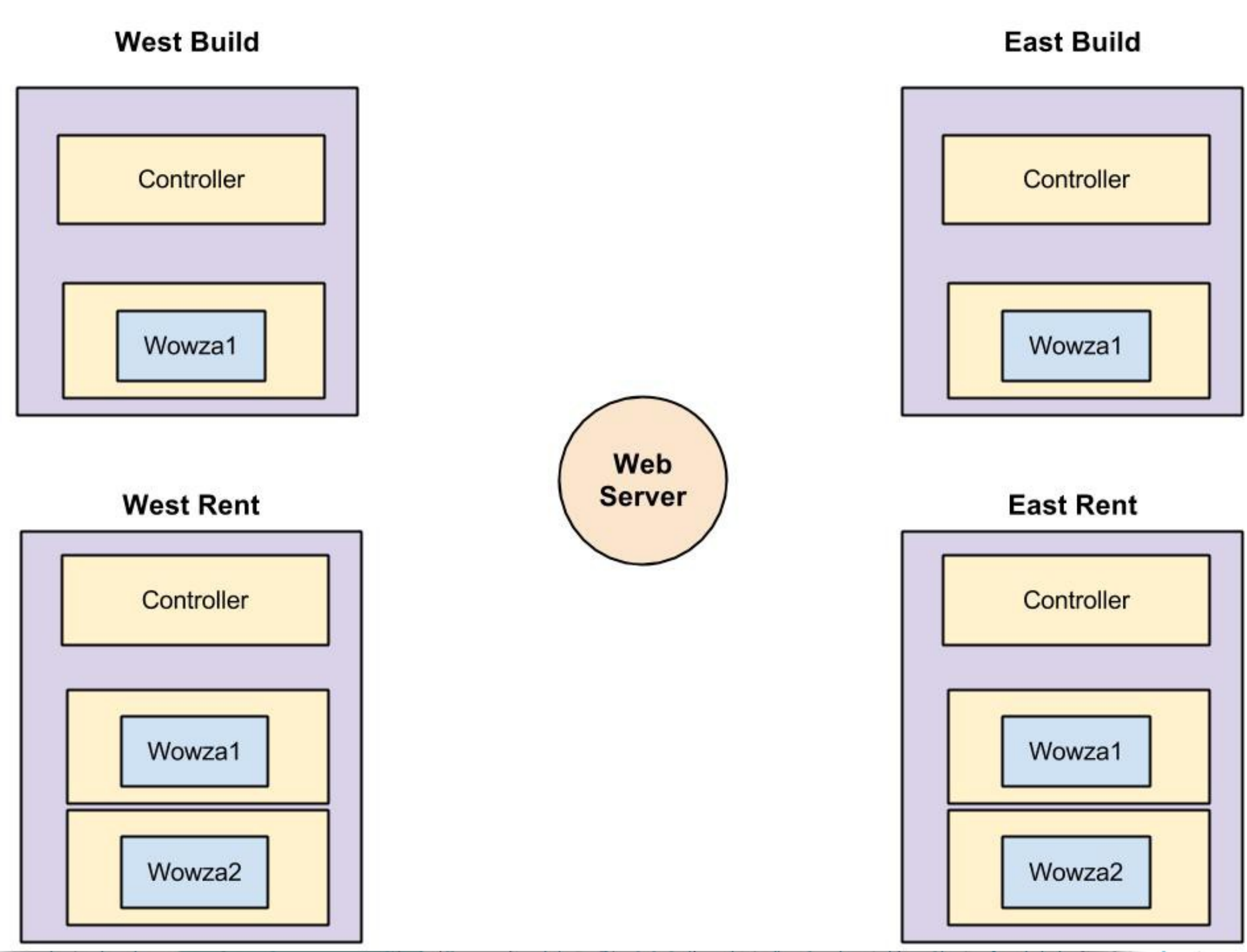
- Unused tokens are expired after a timeout if subsequent tokens are received
- Token pool count is incremented based on the number of unused tokens
- Handling users who have left

Tools used

- OpenStack
- Tomcat Web-Server
- Spread Toolkit
- Maxmind GeoIP
- Wowza Media Server

Demo

- How we deployed the system?
- 1 Web-Server
- East – 1 Build + 1 Rent
- West – 1 Build + 1 Rent
- 1 Wowza per Build – Capacity 4
- 2 Wowza per Rent – Capacity 2*2
- Total Capacity – 16


West Build
Controller
Wowza1
East Build
Controller
Wowza1
Web Server
West Rent
Controller
Wowza1
Wowza2
East Rent
Controller
Wowza1
Wowza2

Demo

- Demo 1 - A sample video streaming
- Demo 2 - Correct allocation of data center
- Demo 3 - Transition from build to rent

Future Work

- Application in the data center auto-scalable
- Login authorization & Access Control
- Fault tolerant & Partition aware

References

- http://gigaom.com/2012/07/03/netflix-june-one-billion-hours/
- http://www.wowza.com/faq
- http://aws.amazon.com/about-aws/globalinfrastructure/
- http://techcrunch.com/2013/04/22/netflix-beats-analyst-estimates-with-29-2-million-us-subscribers-and-1-billion-in-q1-revenue/
- http://www.betterbroadbandblog.com/2010/10/netflix-time-of-day-and-relative-metrics/
- http://dev.maxmind.com/geoip/
- http://www.spread.org/
- http://www.openstack.org/
- http://tomcat.apache.org/

Thank You

Special thanks to the members of the DSN Lab for listening to our long discussions and patiently answering all our questions