Sirikata

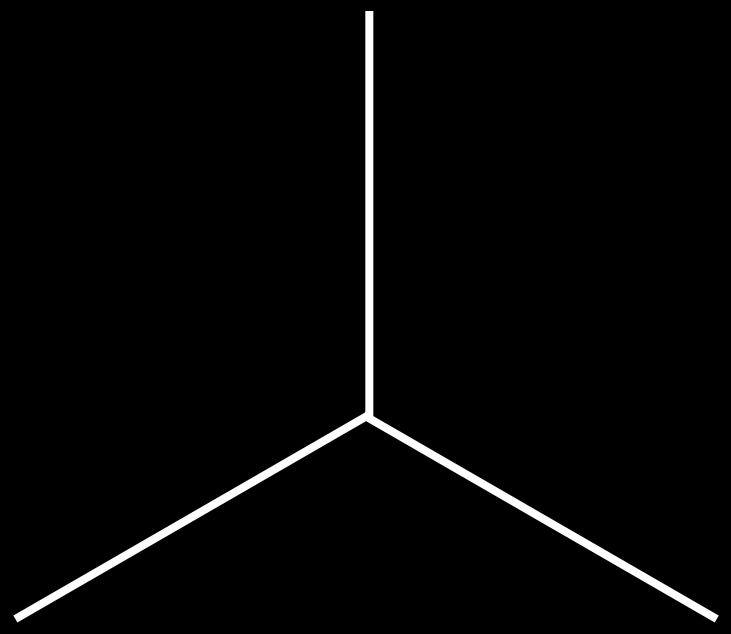
ISTC-VC Virtual Worlds Workshop
August 18, 2011

What Is It?

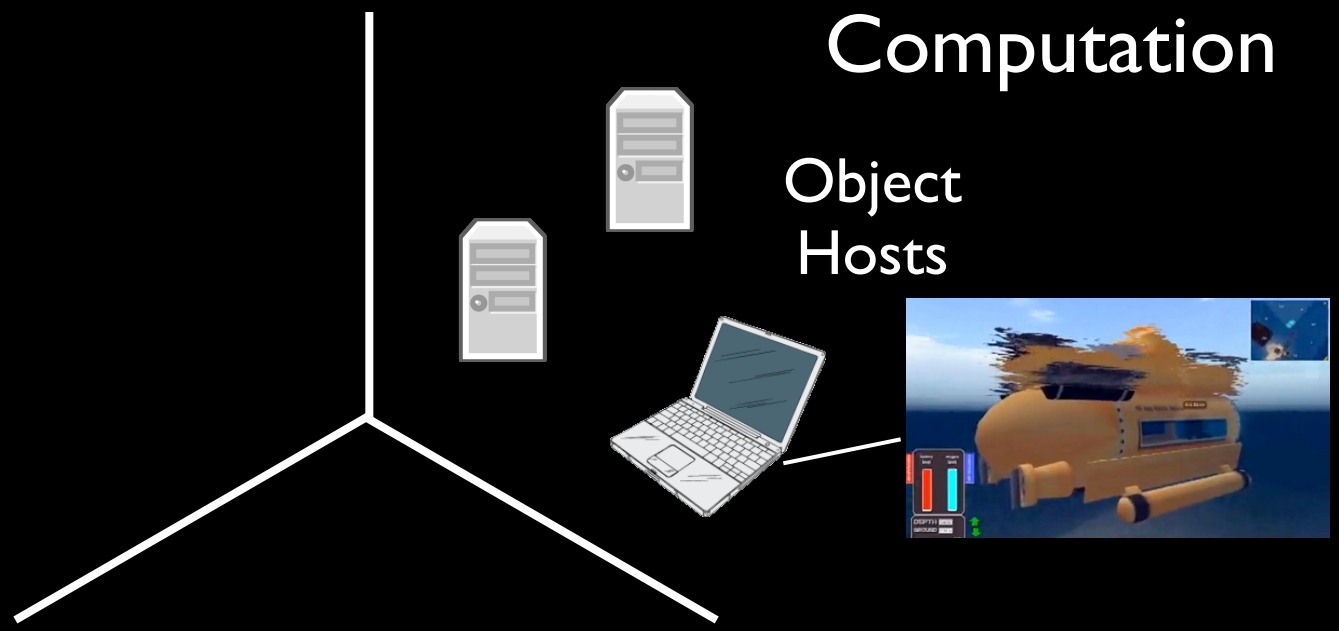
- Software platform for 3D virtual worlds
- Princeton/Stanford collaboration

- Goal: A user should be able to interact with the whole virtual world.
- Goal: Easy application development.

Sirikata Architecture

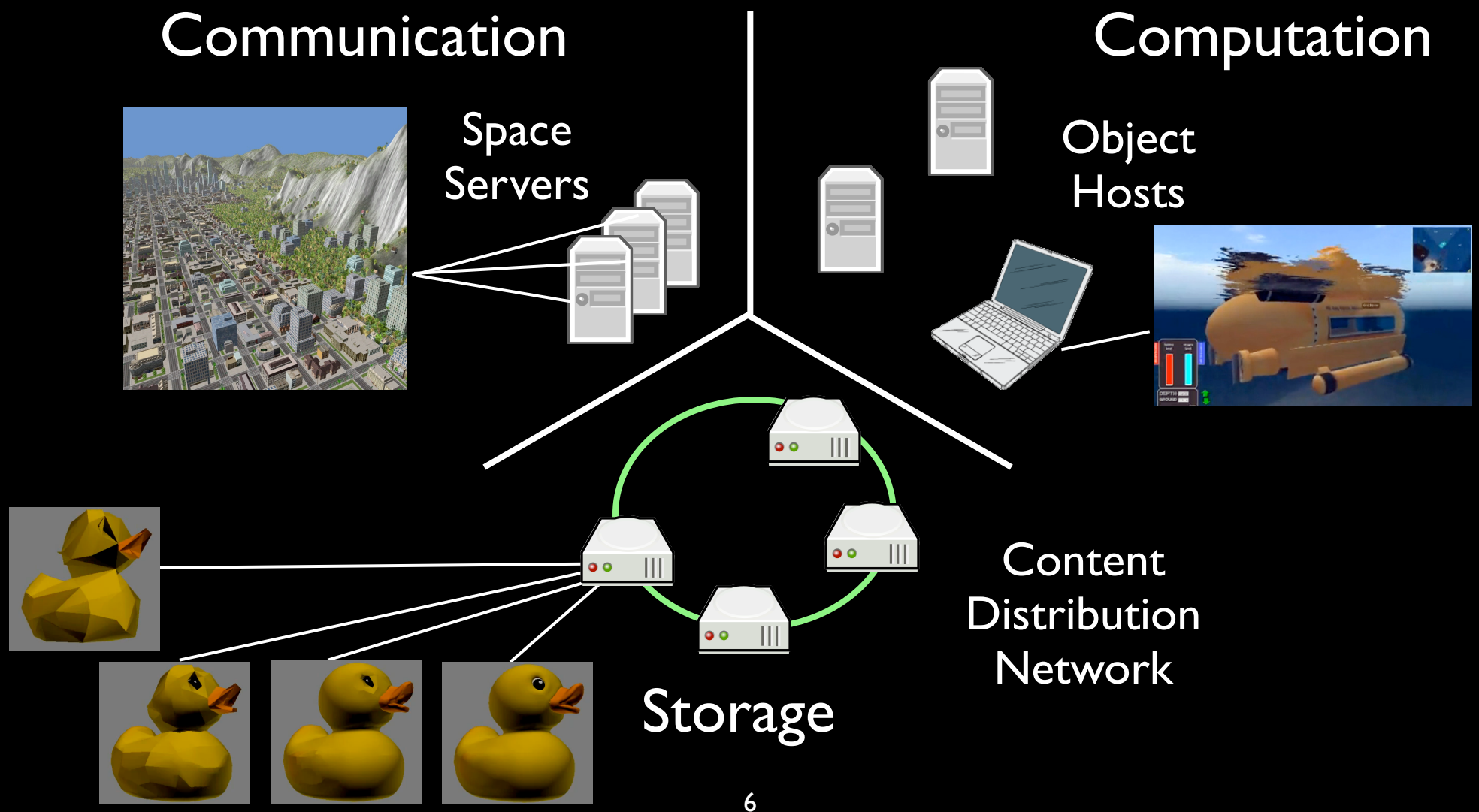


Sirikata Architecture

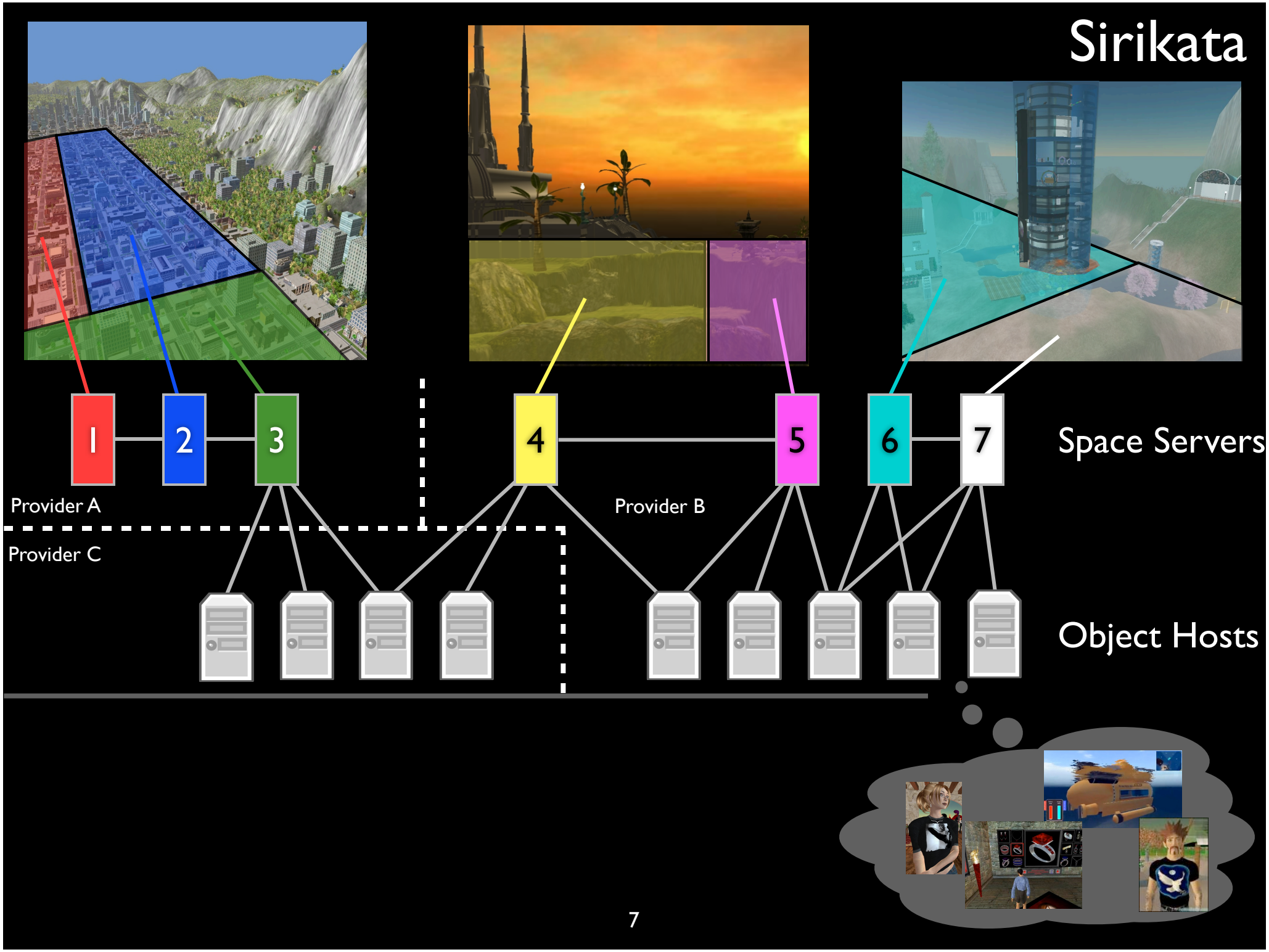


Sirikata Architecture

Sirikata Architecture



Sirikata


1
2
3
4
5
6
7
Space Servers
Provider A
Provider B
Provider C
Object Hosts

What Is It?

- Software platform for 3D virtual worlds
- Princeton/Stanford collaboration

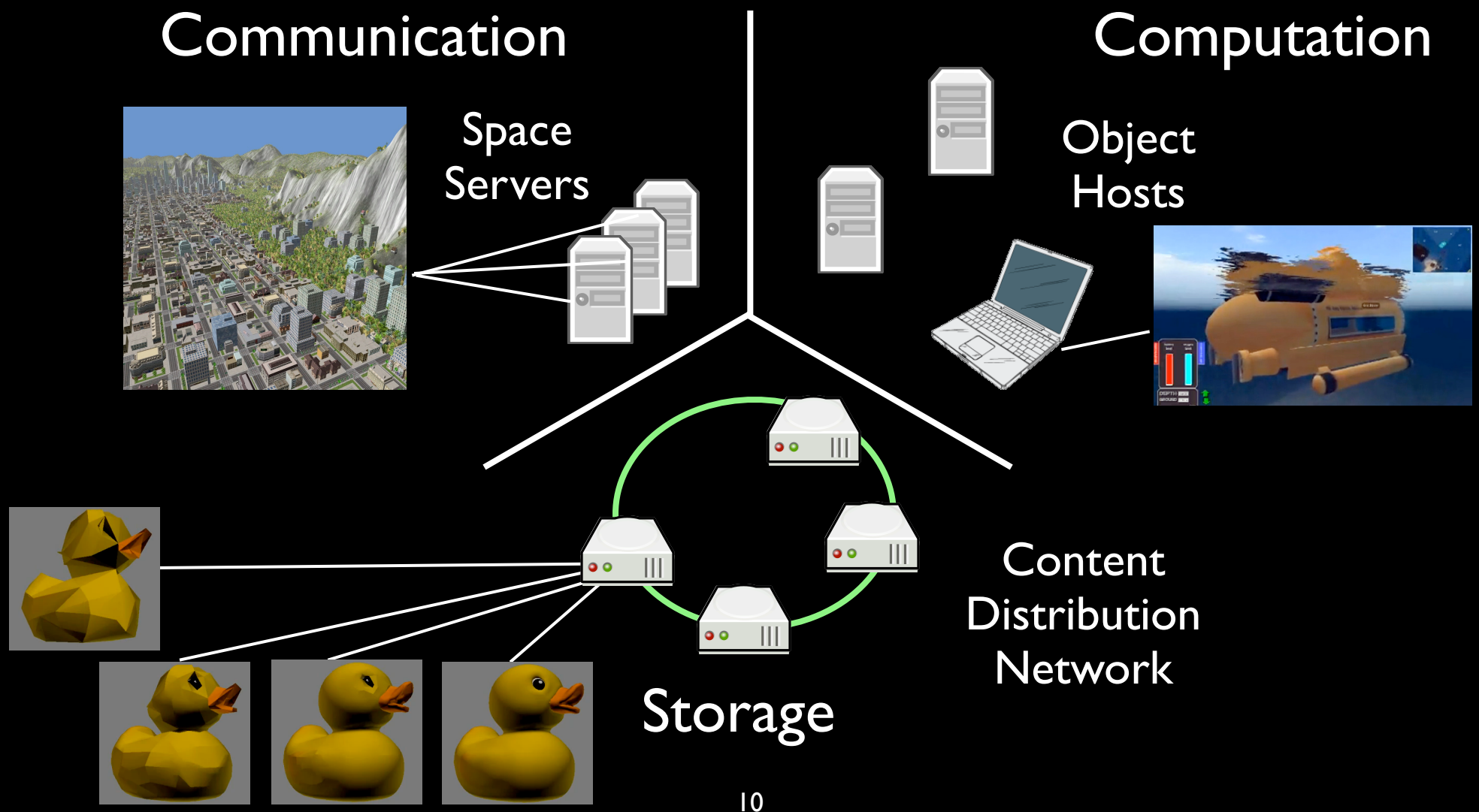
- Goal: A user should be able to interact with the whole virtual world.
- Goal: Easy application development.

What Is It?

- Software platform for 3D virtual worlds
- Princeton/Stanford collaboration

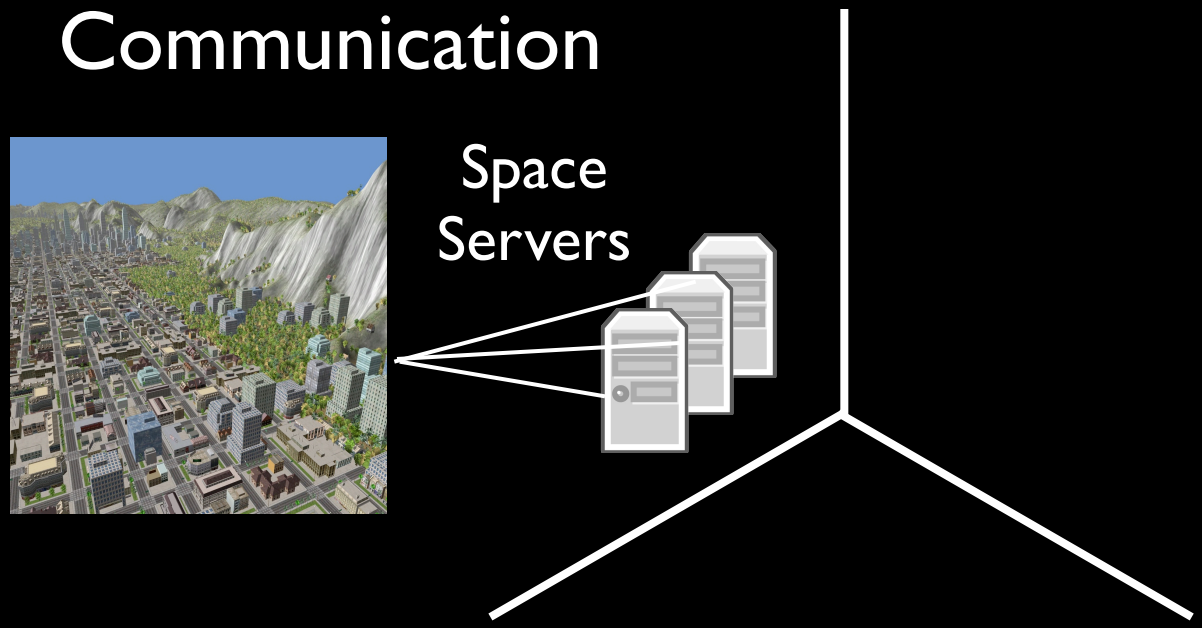
- Goal: A user should be able to interact with the whole virtual world.
- Goal: Easy application development.

Sirikata Architecture

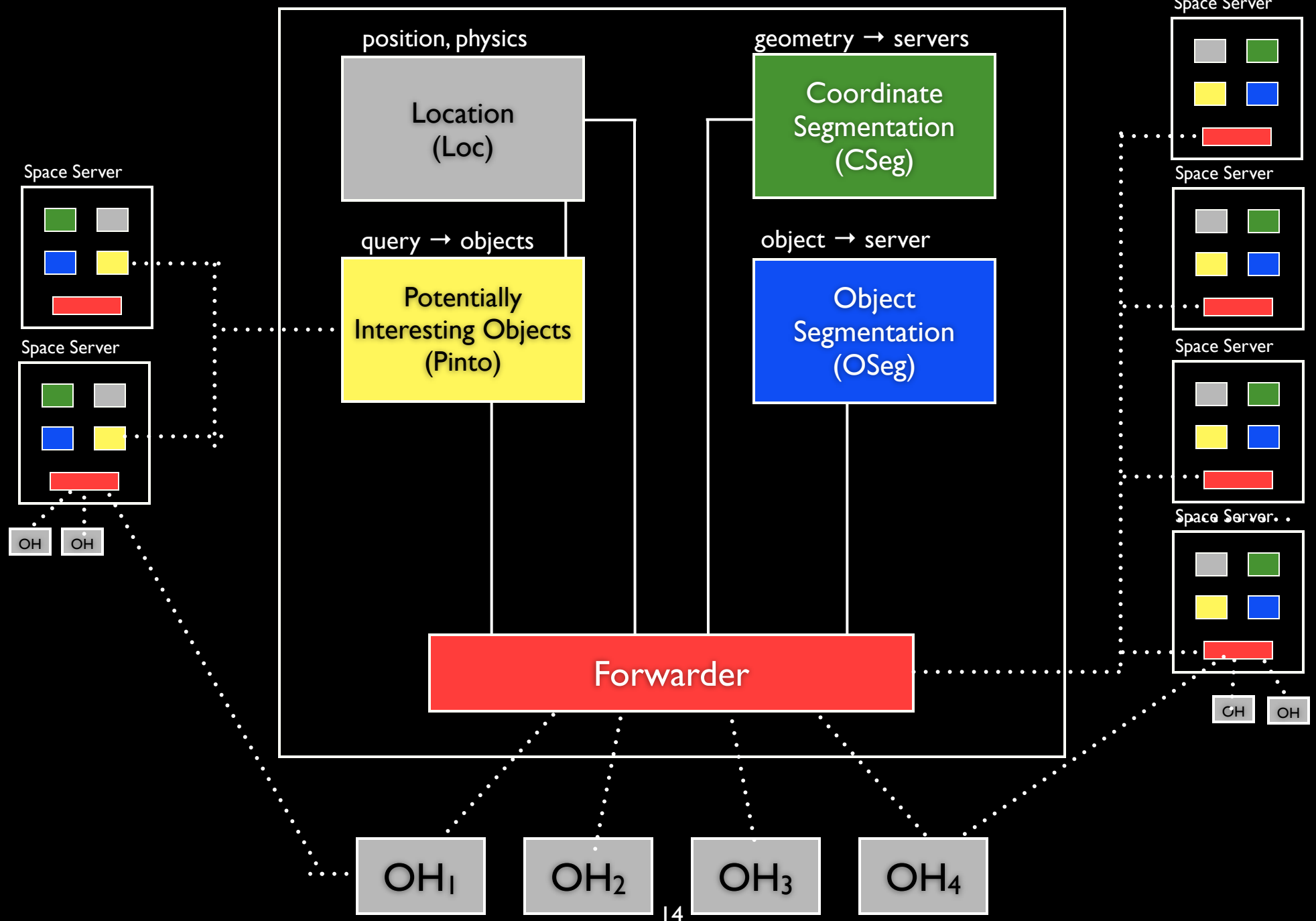


Sirikata Architecture

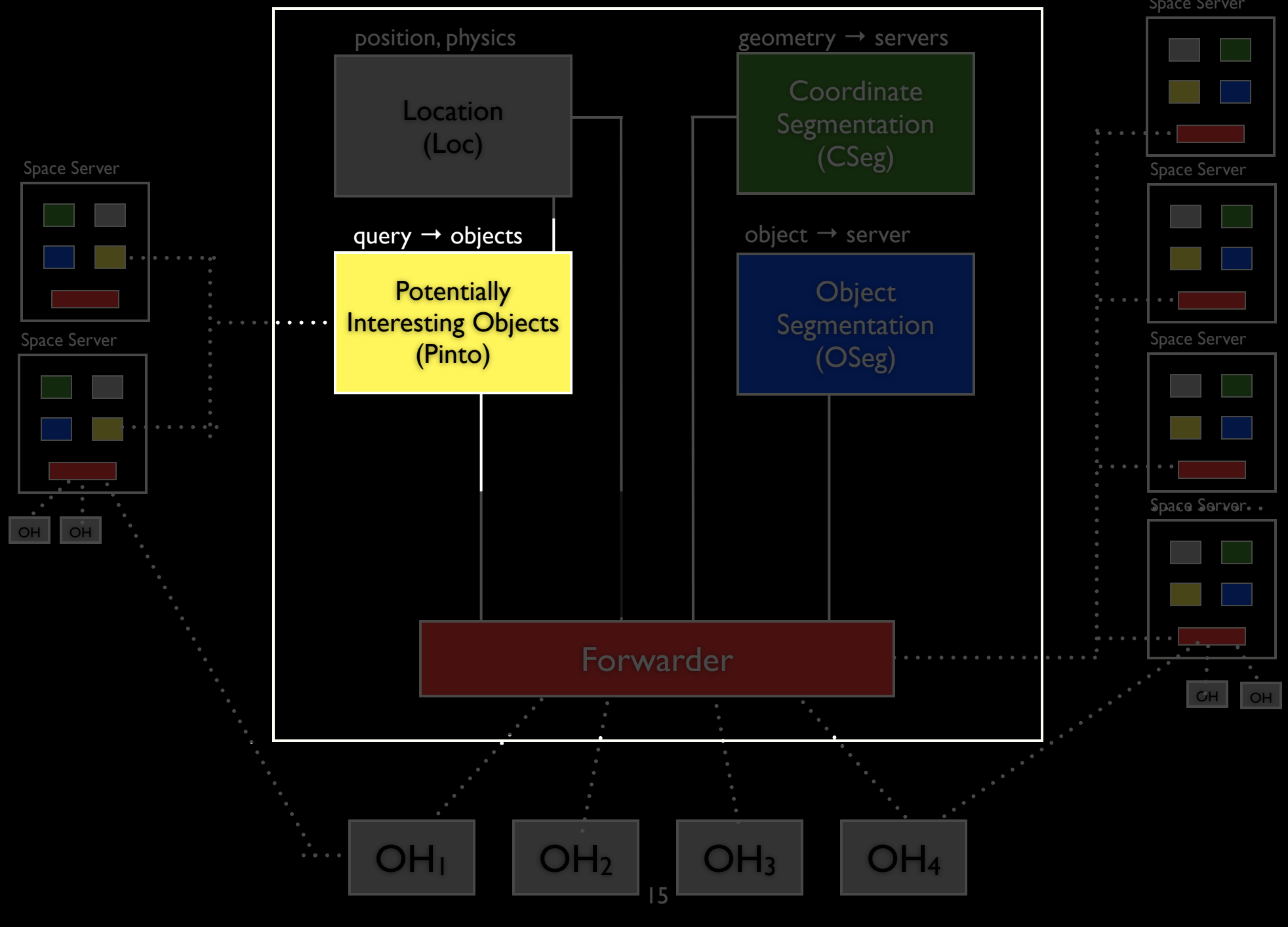
Communication



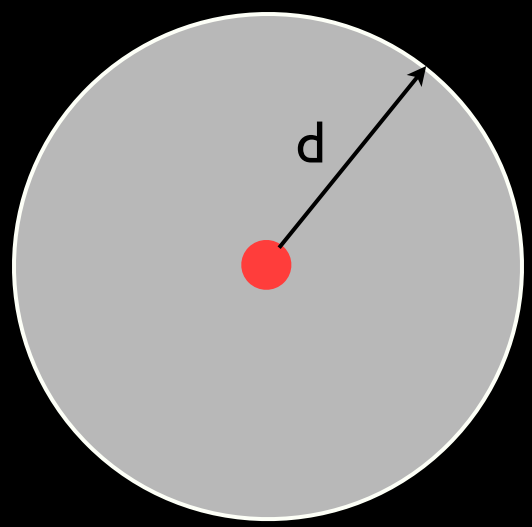
Space Server



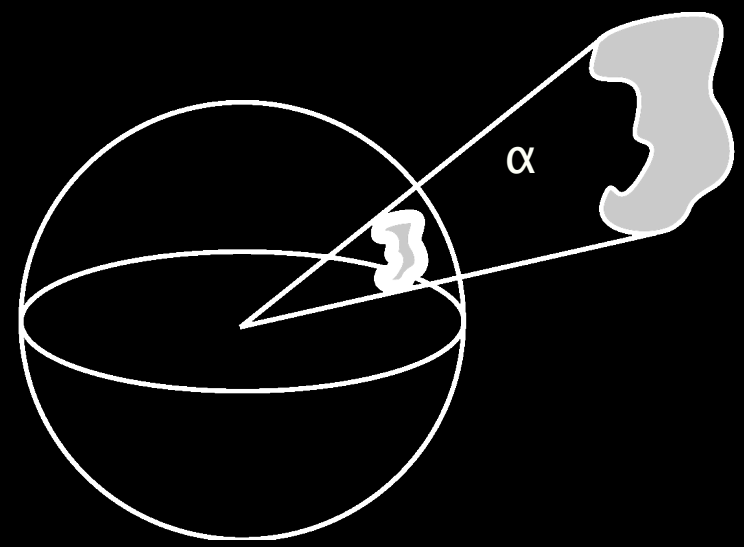
Space Server



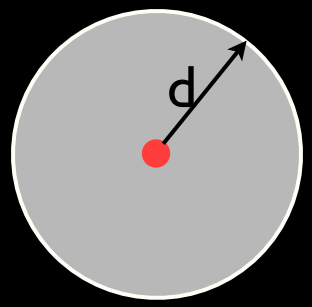
Solid Angles



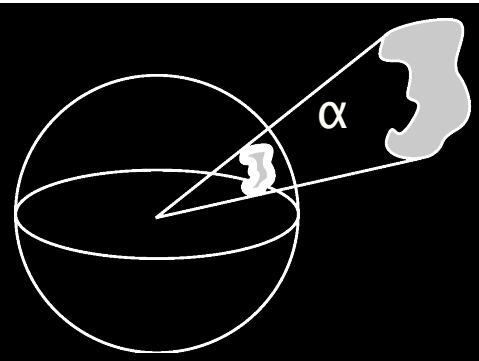
distance



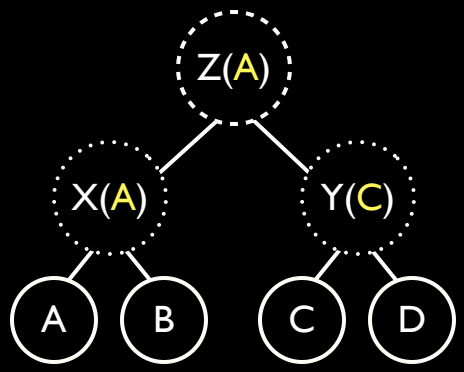
solid angle



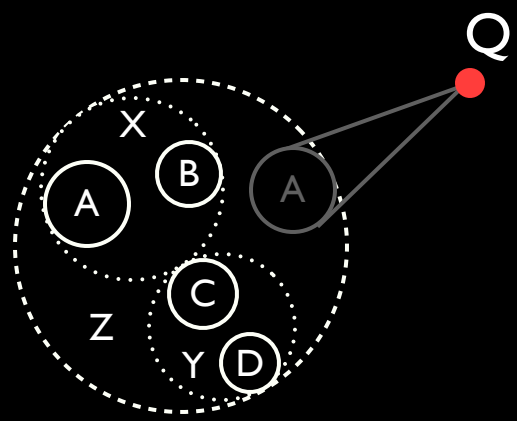
distance



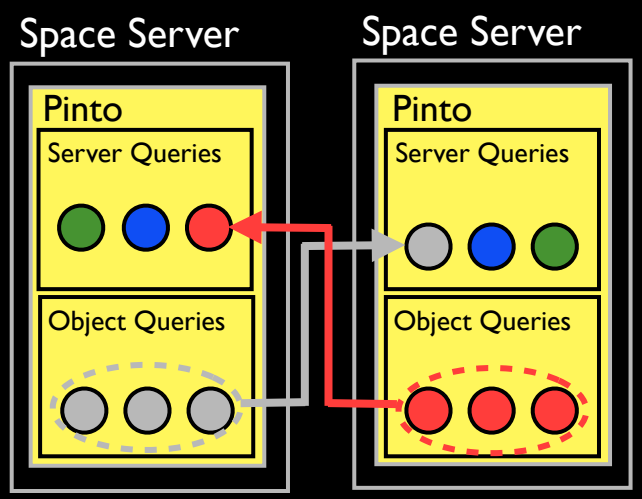
solid angle

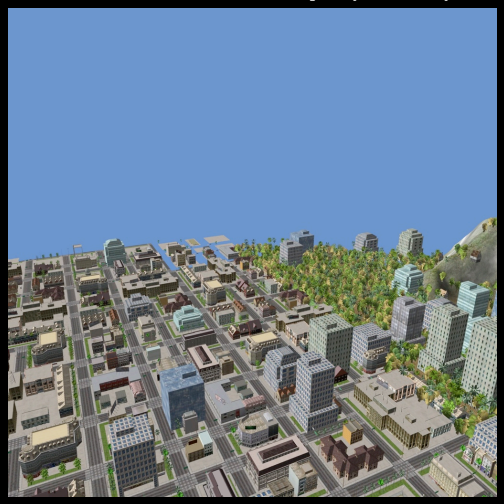


bounding volume hierarchy (BVH)

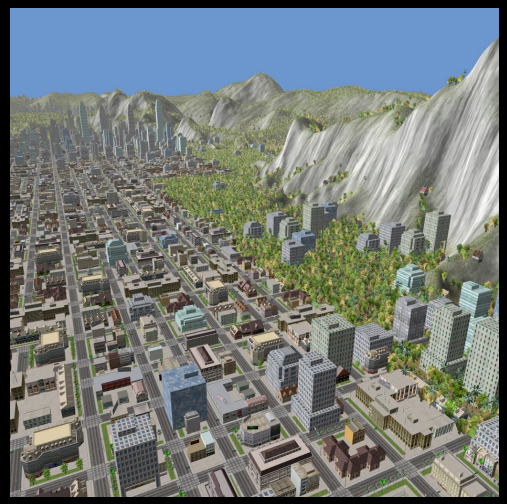


BVH geometry

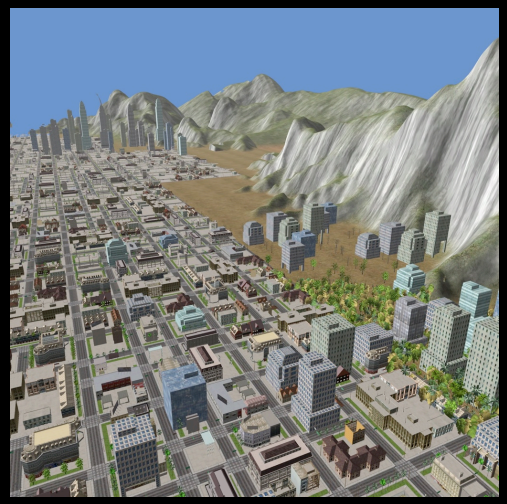




distance (20,000)



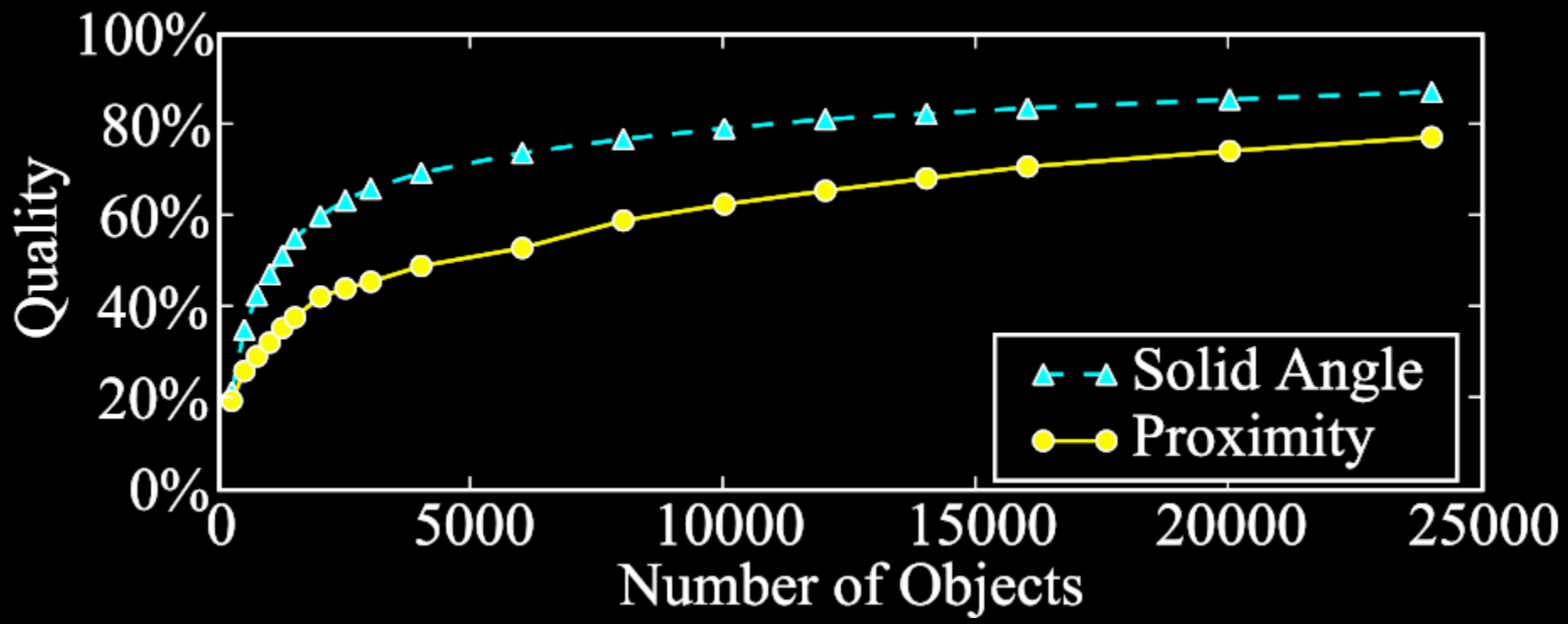
ideal (10,000,000)[17]



angle (20,000)

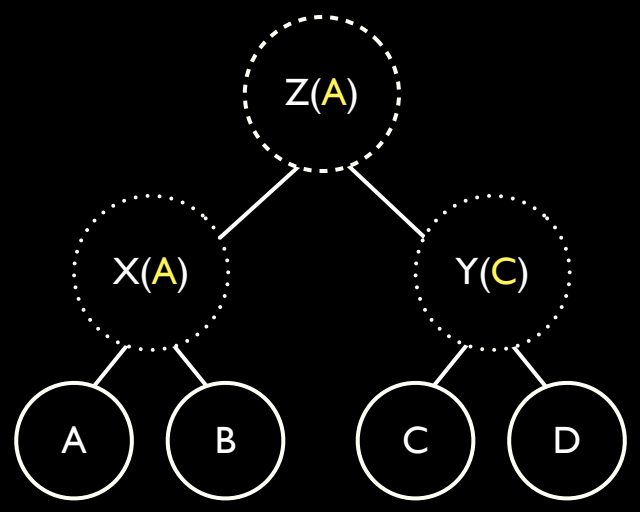
Pinto

Query Comparison

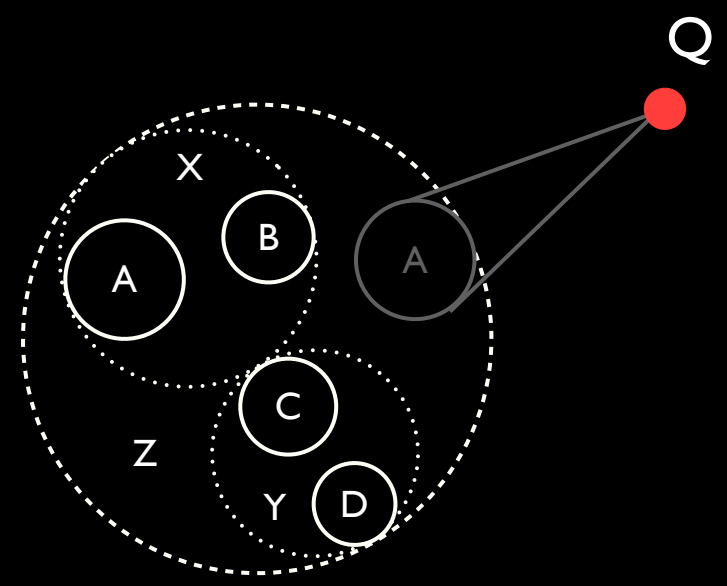


- For same # of objects, better quality
- For same quality, fewer objects

LBVH: Largest Bounding Volume Hierarchy

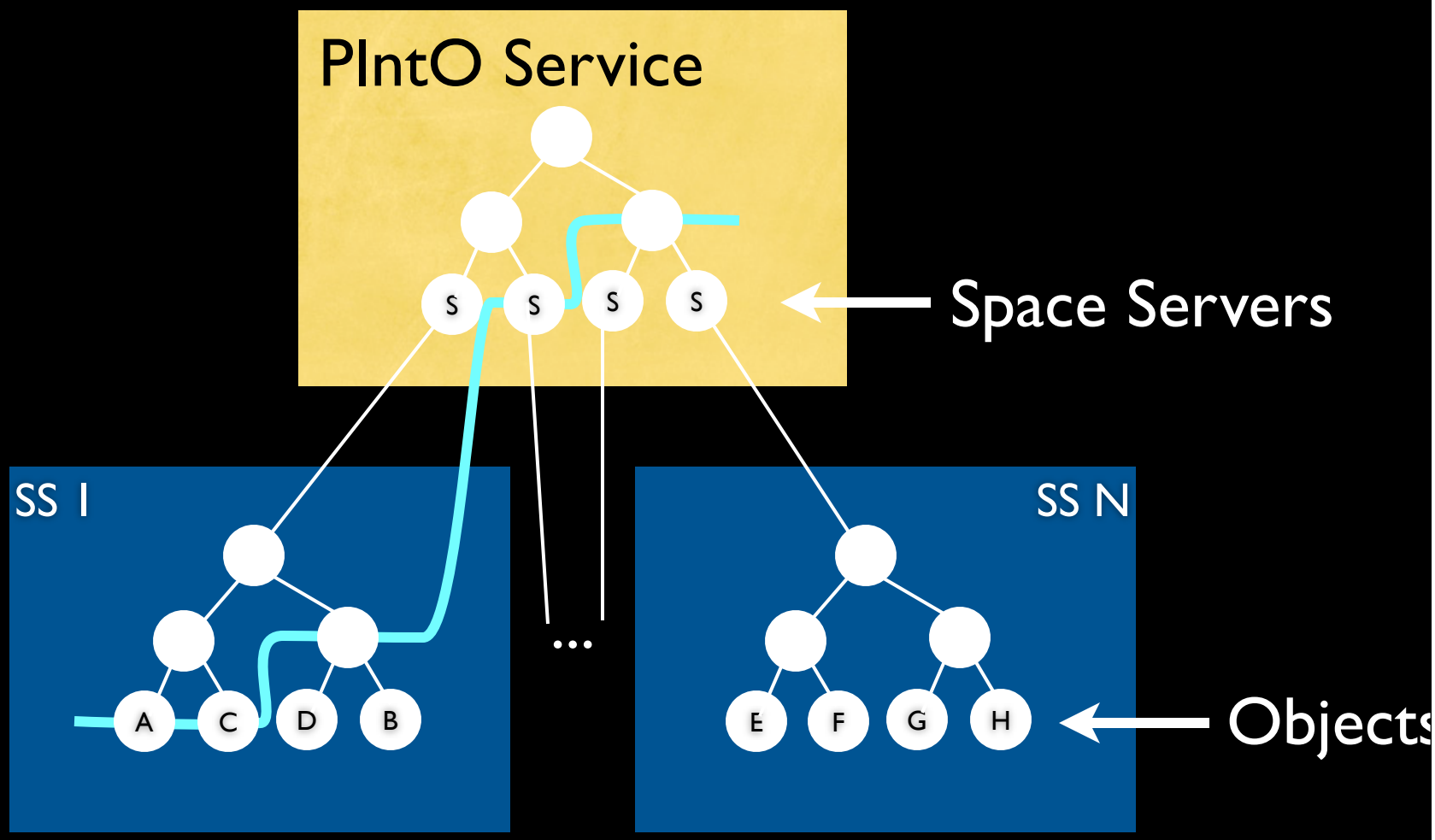


bounding volume hierarchy (BVH)



BVH geometry

PIntO Conceptual Model



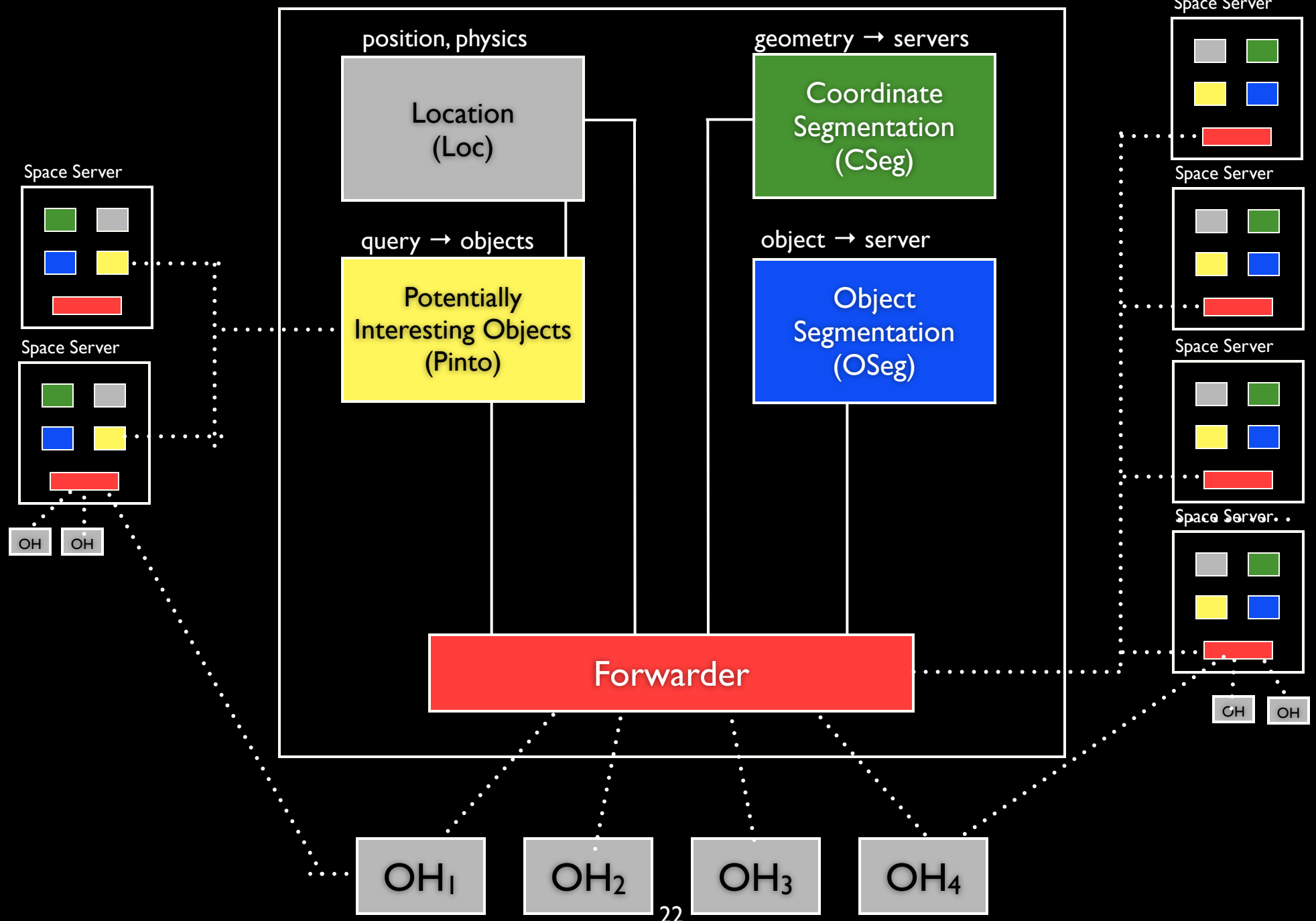
Distance

Ideal

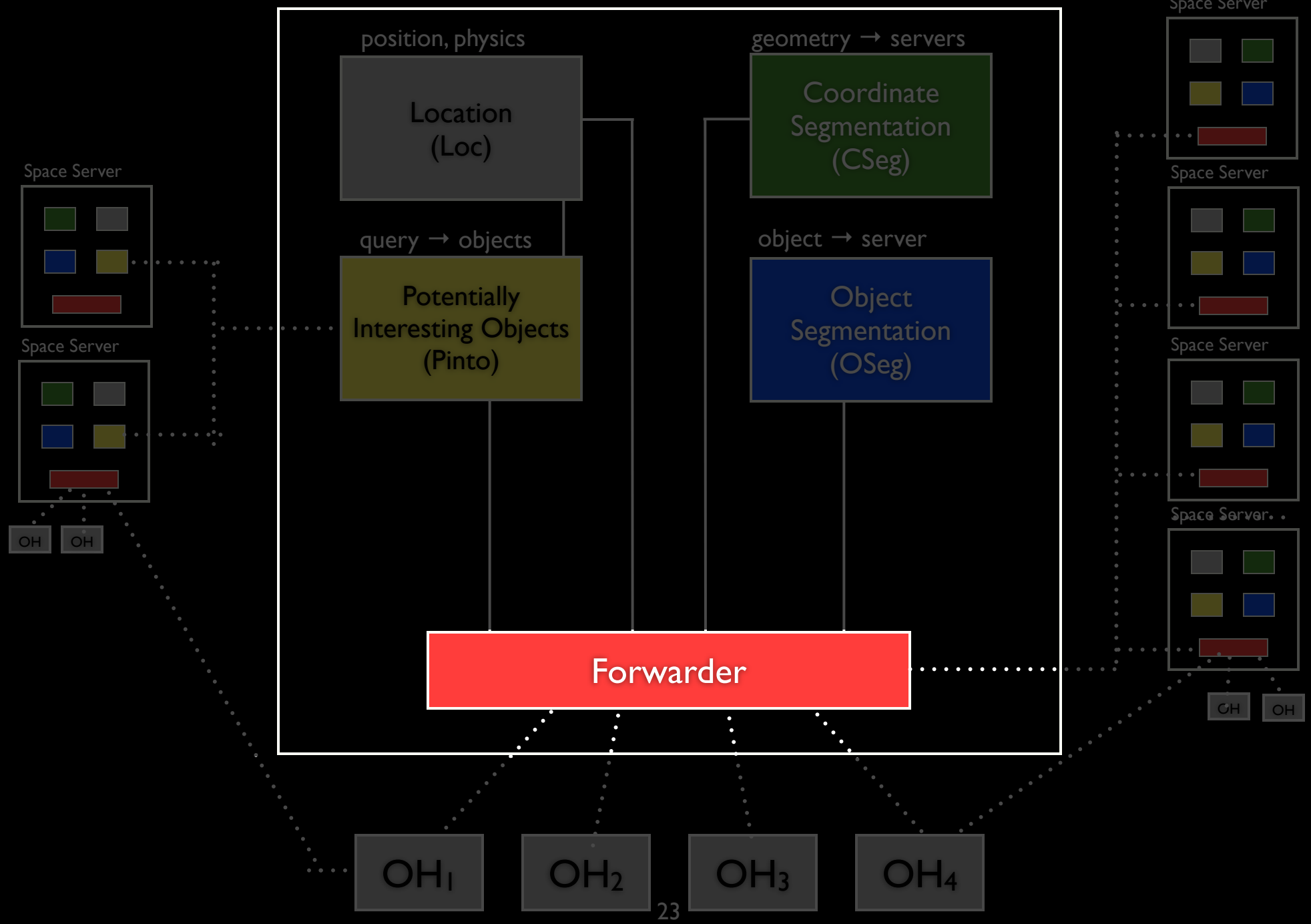
Cuts

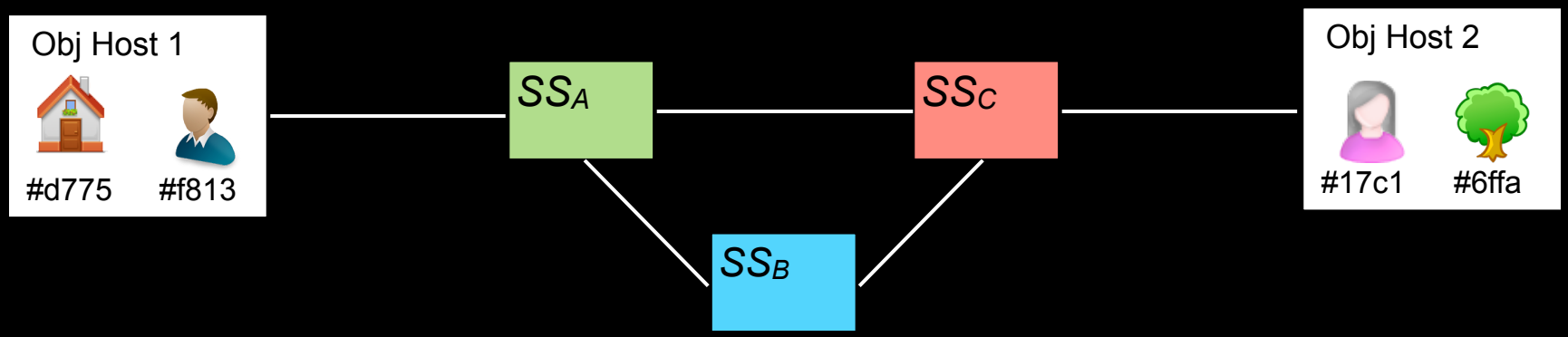
Aggregates

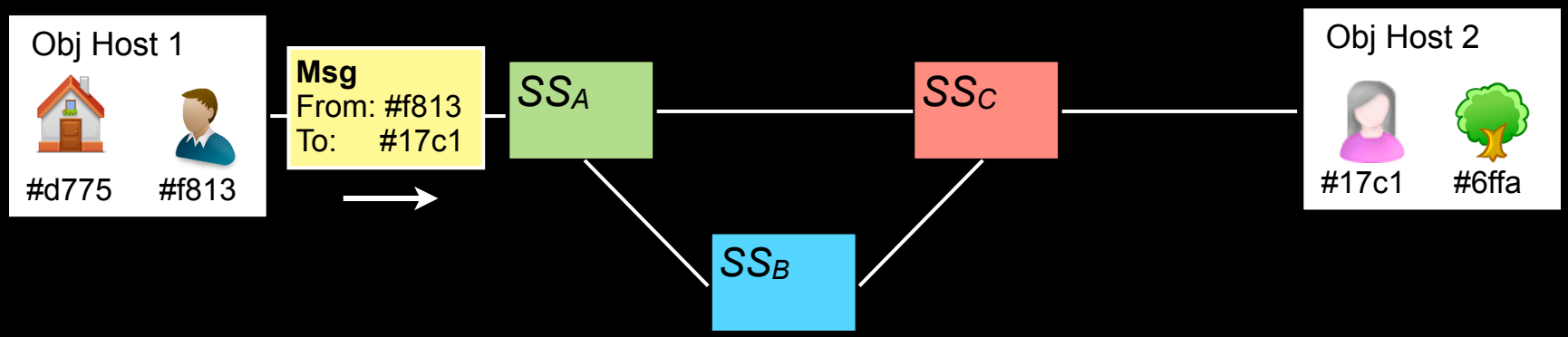
Space Server

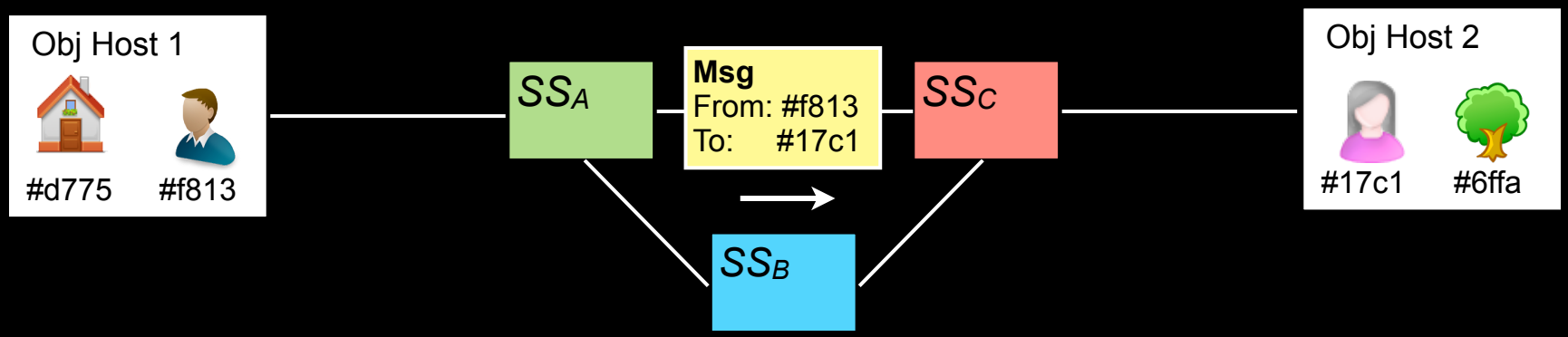


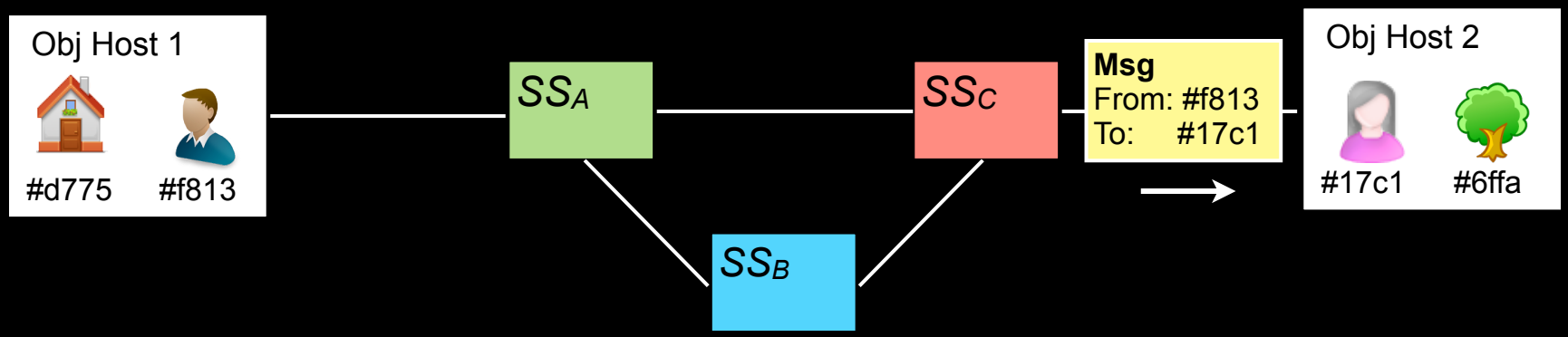
Space Server



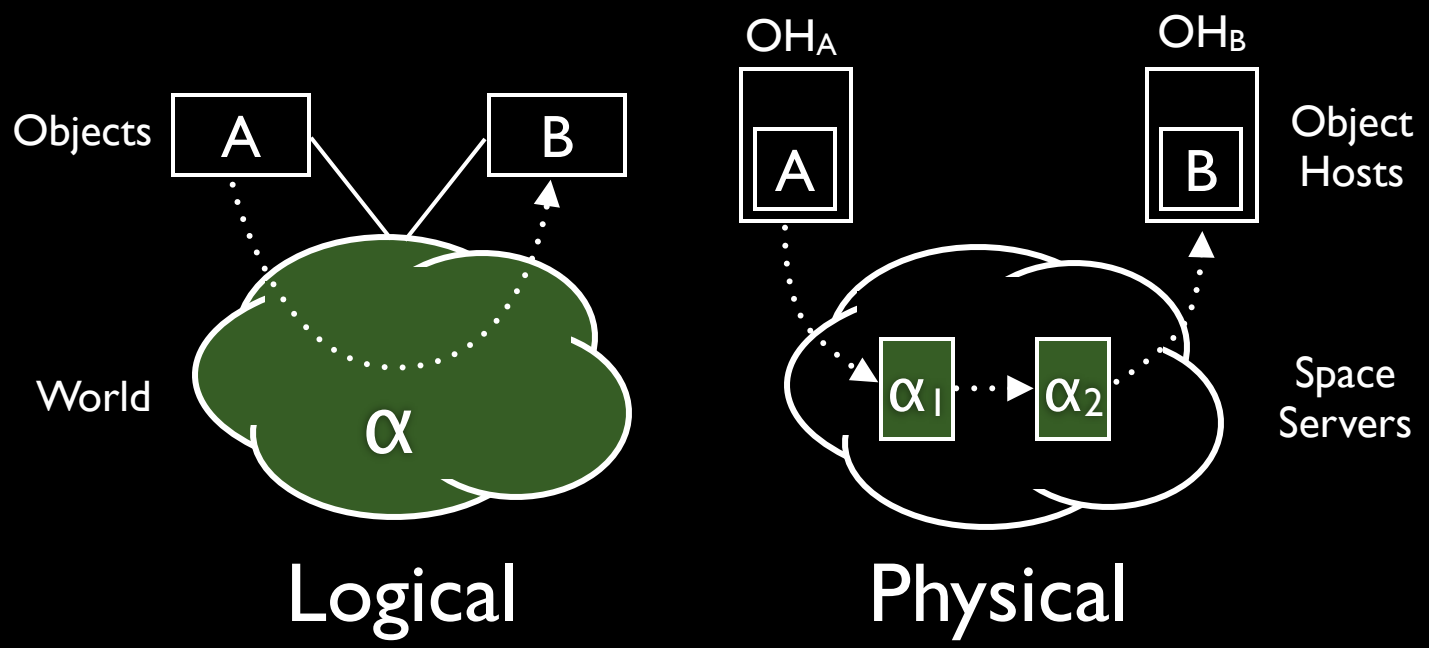

Obj Host 1
#d775
#f813
SS_A
SS_C
SS_B
Obj Host 2
#17c1
#6ffa


Obj Host 1
#d775
#f813
Msg
From: #f813
To: #17c1
SSA
SSC
SSB
Obj Host 2
#17c1
#6ffa


Obj Host 1
#d775
#f813
SSA
Msg
From: #f813
To: #17c1
SSC
SSB
Obj Host 2
#17c1
#6ffa


Obj Host 1
#d775
#f813
SS_A
SS_C
SS_B
Msg
From: #f813
To: #17c1
Obj Host 2
#17c1
#6ffa

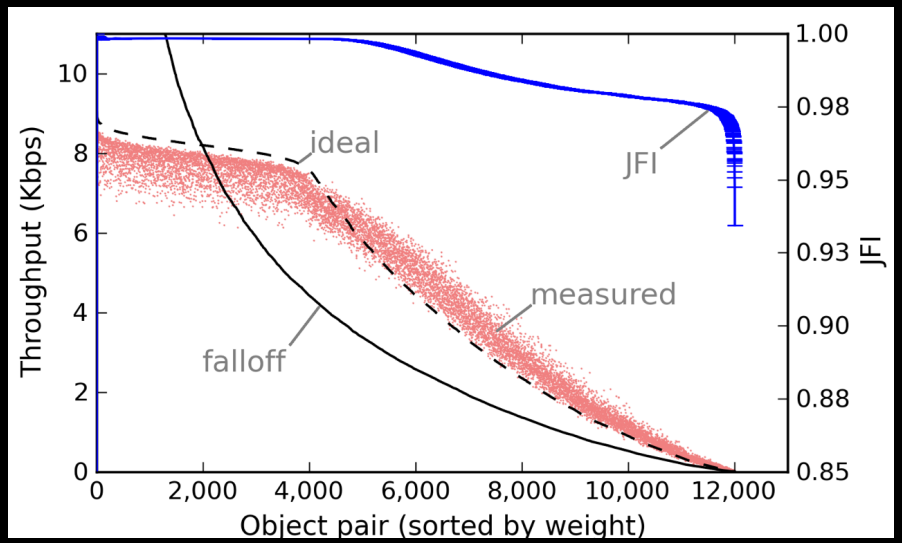
Forwarder



$$w_i = F(R_s, R_d) = \int \int f(p_s, p_d)\, d_d\, d_s$$

$$r = d(p_s, p_d)$$

$$f(r) = \frac{1}{(sr + \rho)^2 \cdot \log^2(sr + \rho)}$$

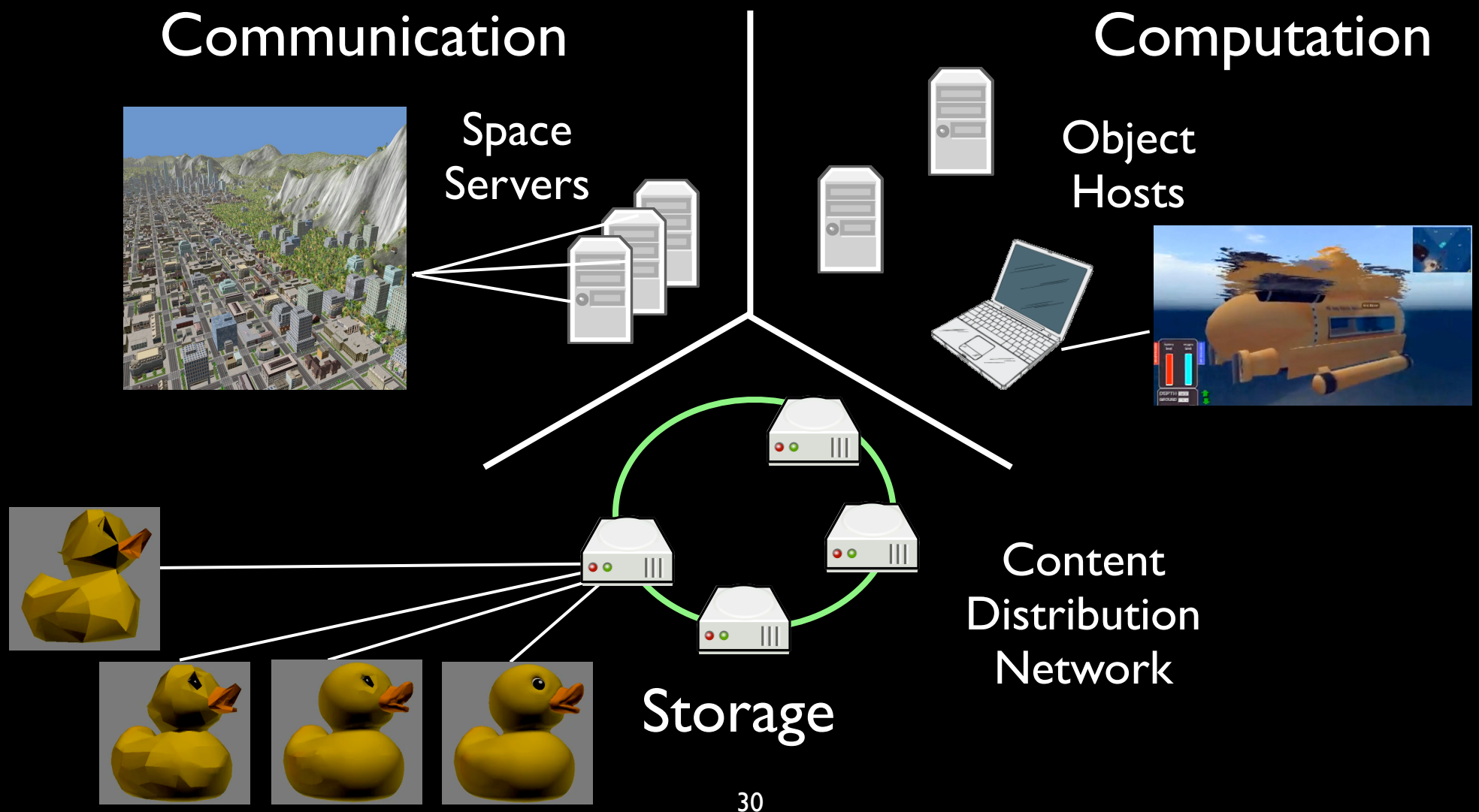


What Is It?

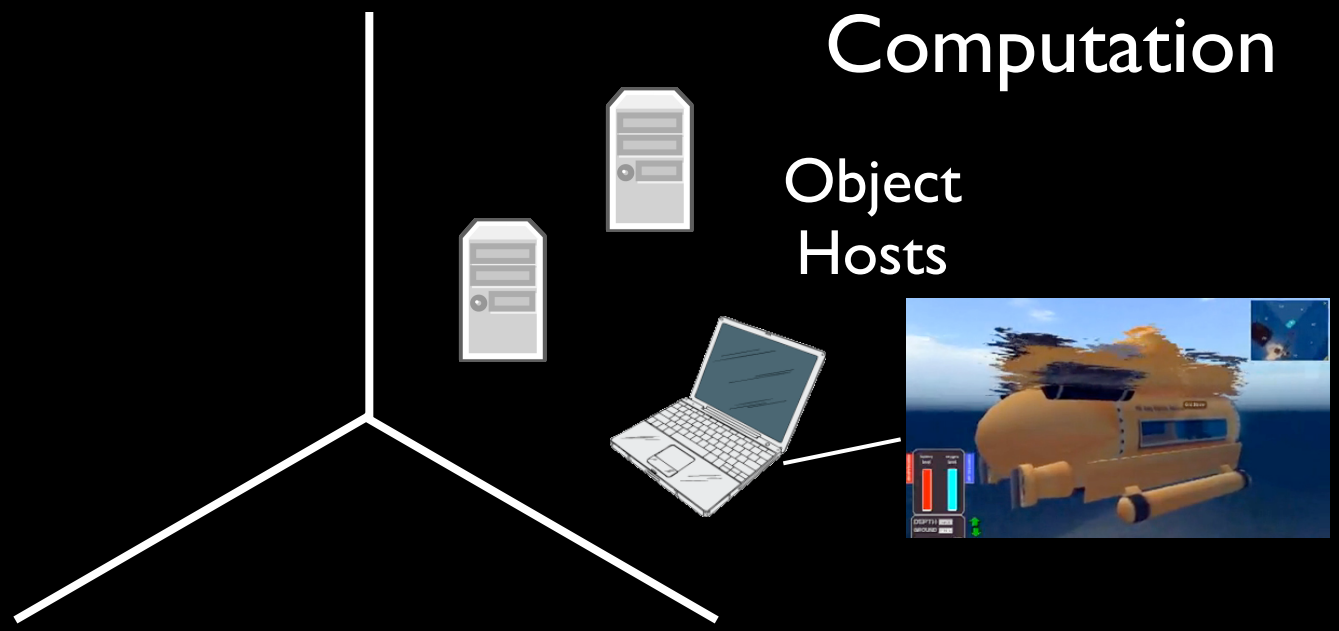
- Software platform for 3D virtual worlds
- Princeton/Stanford collaboration

- Goal: A user should be able to interact with the whole virtual world.
- Goal: Easy application development.

Sirikata Architecture



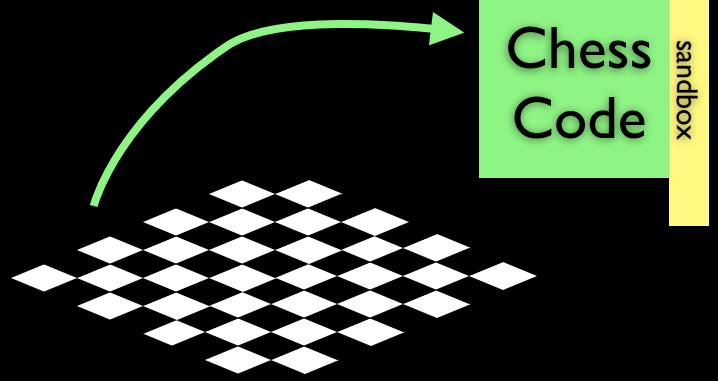
Sirikata Architecture

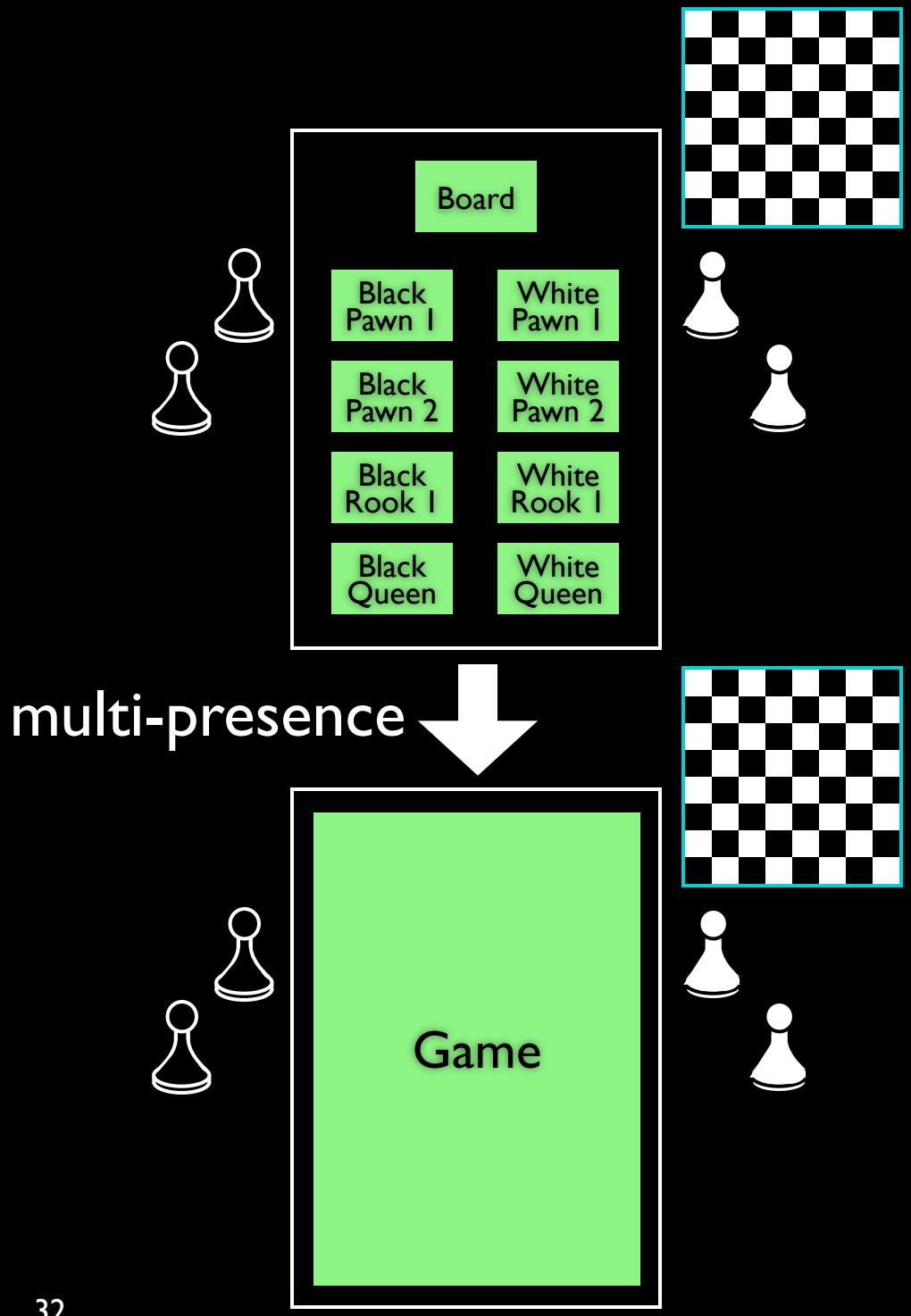


Emerson,

a language for virtual worlds

sandboxing





Survival Game: Zombie

Only one zombie attacks at a time

Zombie Game: Basic Actor Model

Zombie 1 script

Zombie 2 script

Zombie 3 script

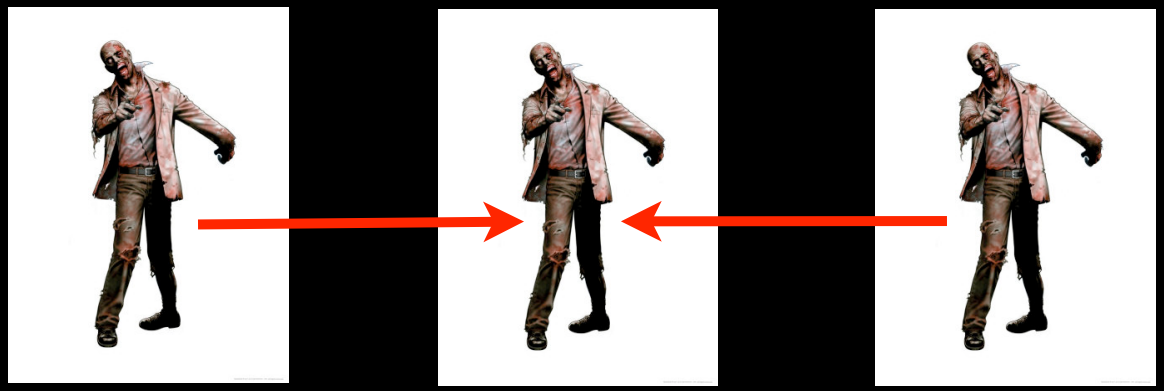
Zombie Game: Basic Actor Model

Zombie 1 script

Zombie 2 script

Zombie 3 script

Zombie Game: Basic Actor Model



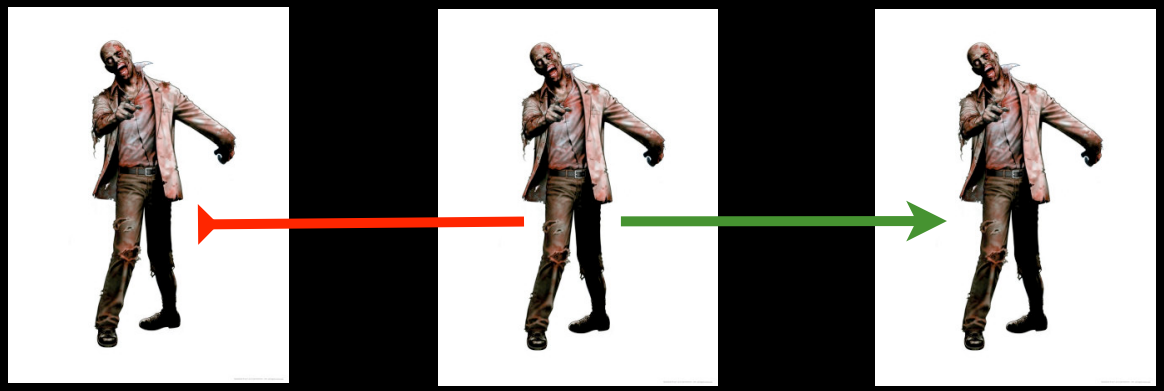
Lock Request

Zombie 1 script

Zombie 2 script

Zombie 3 script

Zombie Game: Basic Actor Model



Lock Granted

Zombie 1 script

Zombie 2 script

Zombie 3 script

Zombie Game: Basic Actor Model

Zombie Attack!

Zombie 1 script

Zombie 2 script

Zombie 3 script

Zombie Game: Multi-Presence

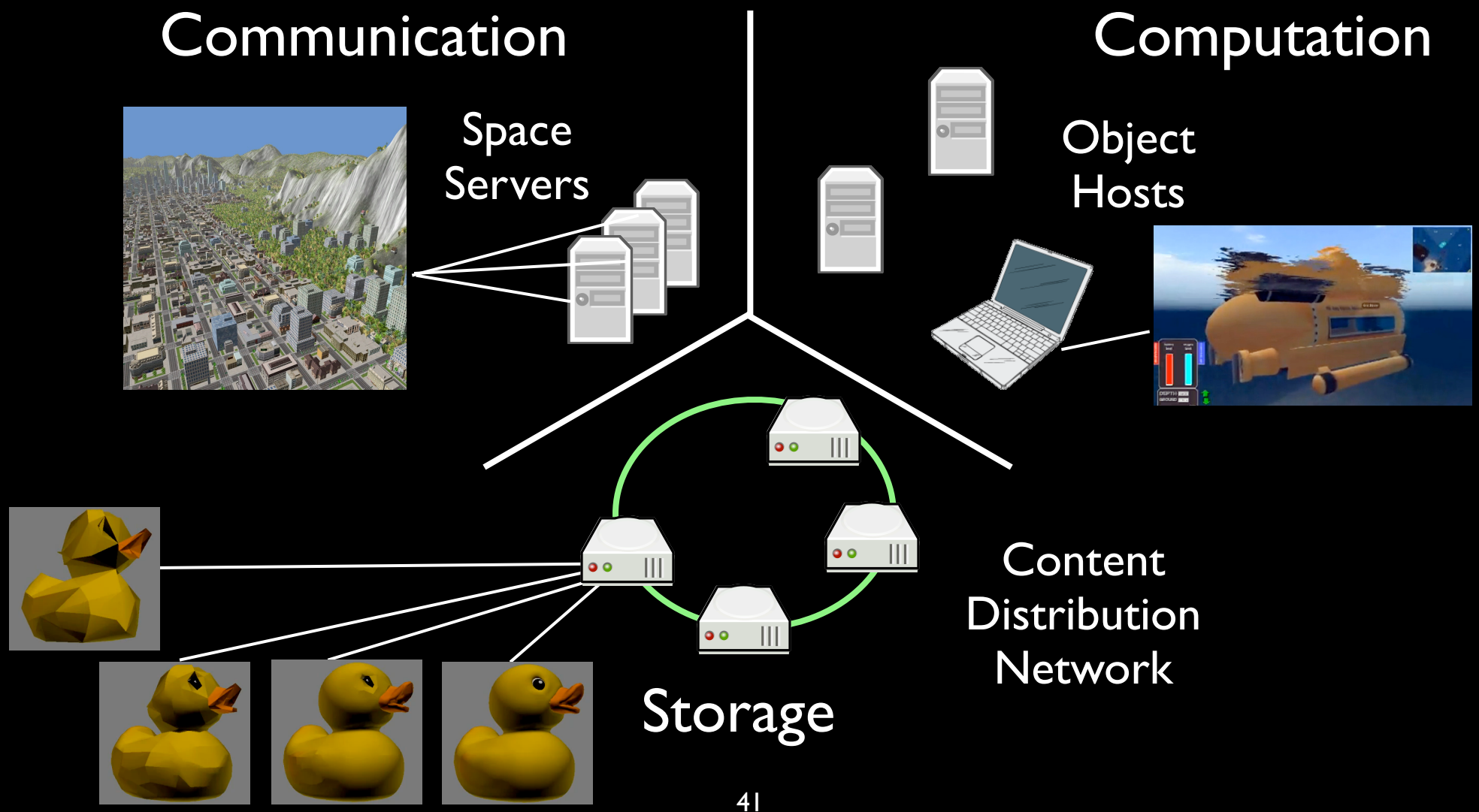
Zombies Script

Zombie Game: Multi-Presence

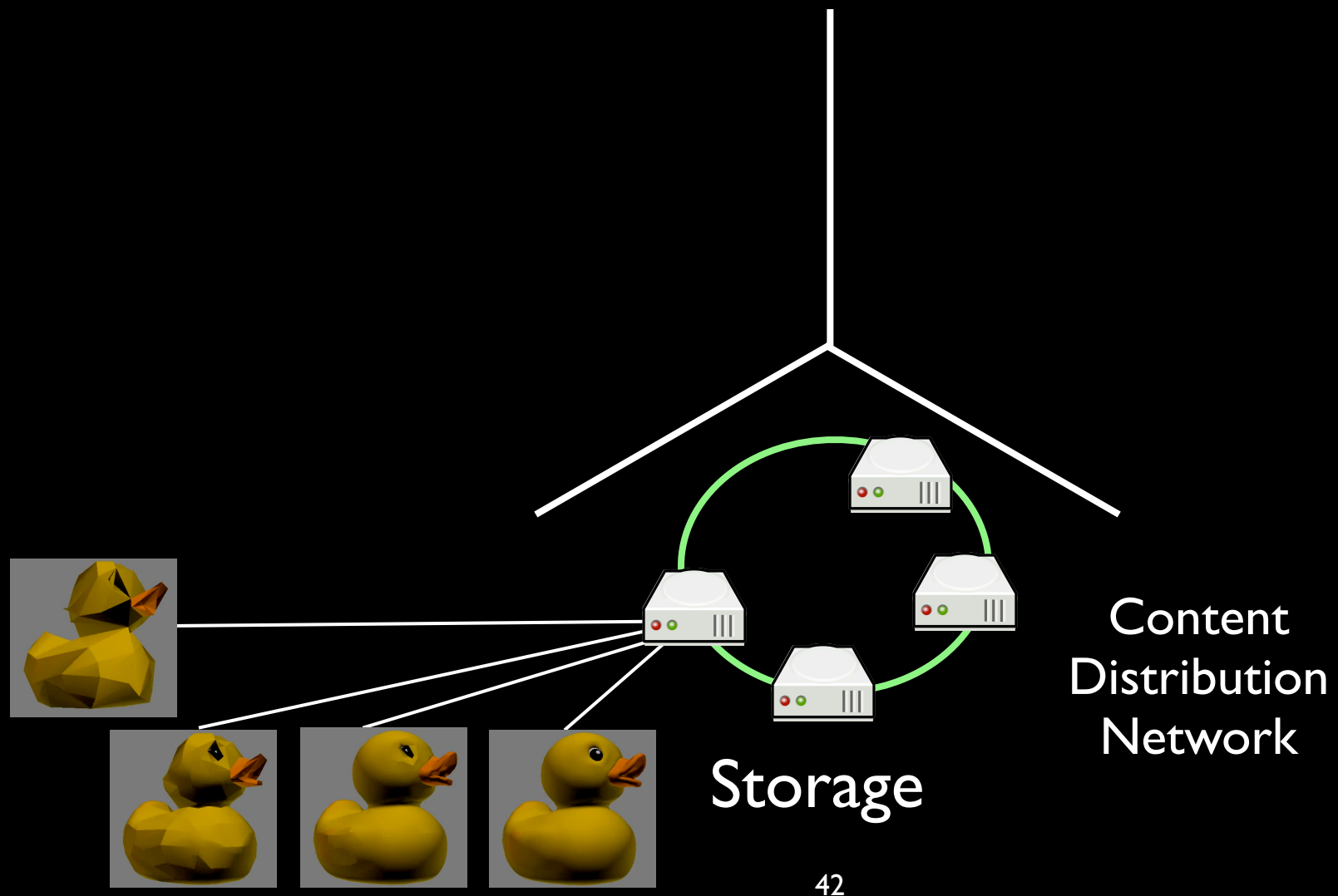
Zombies Script

Zombie A attack!

Sirikata Architecture



Sirikata Architecture

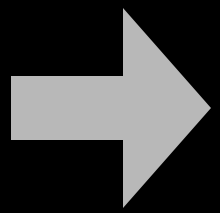


Sirikata CDN

- http://open3dhub.com
  - meerkat:///jterrace/Seagull.dae/optimized/Seagull.dae
- Authentication through OAuth
- Wide-area replication through Cassandra
- DONAR for mapping to nearest replica
- Caching to reduce latency

Making It Easy

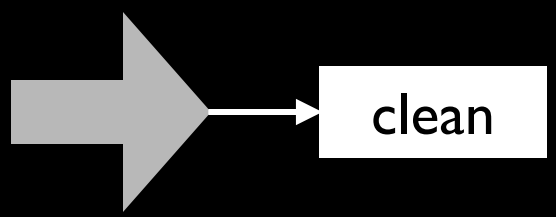
Model upload



Sirikata CDN

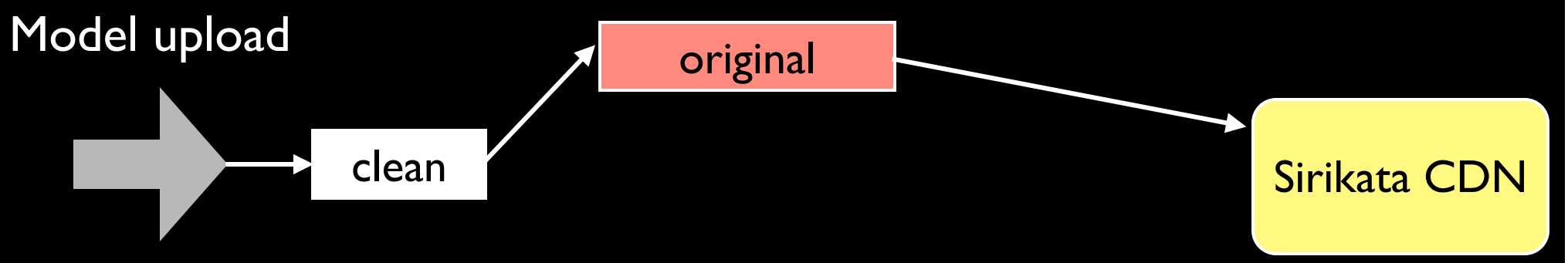
Making It Easy

Model upload

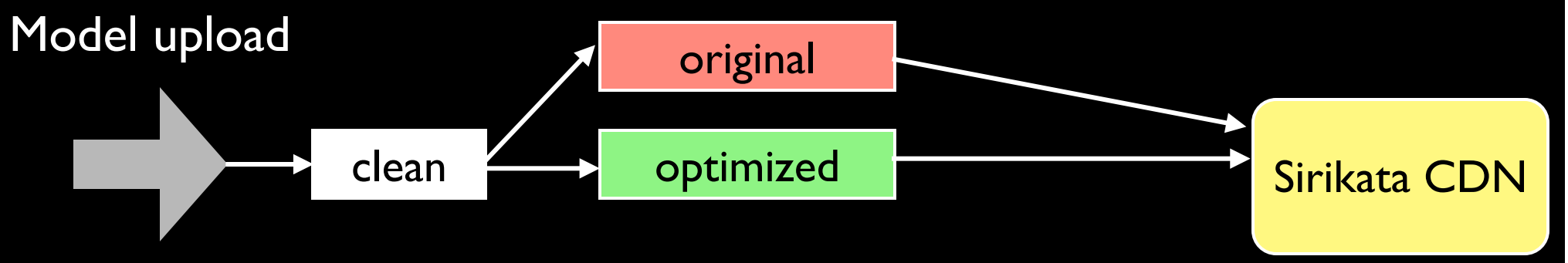


Sirikata CDN

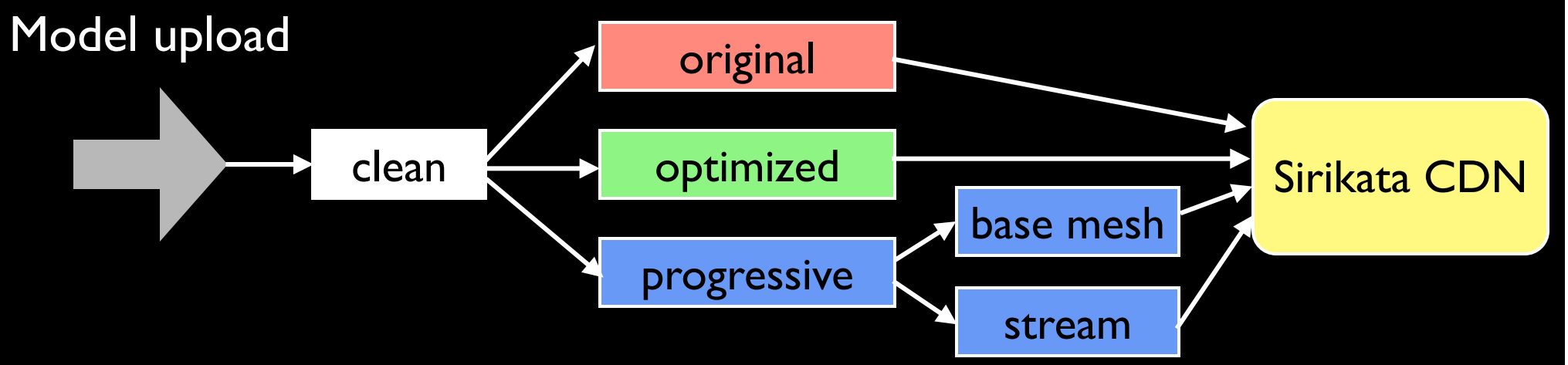
Making It Easy



Making It Easy



Making It Easy



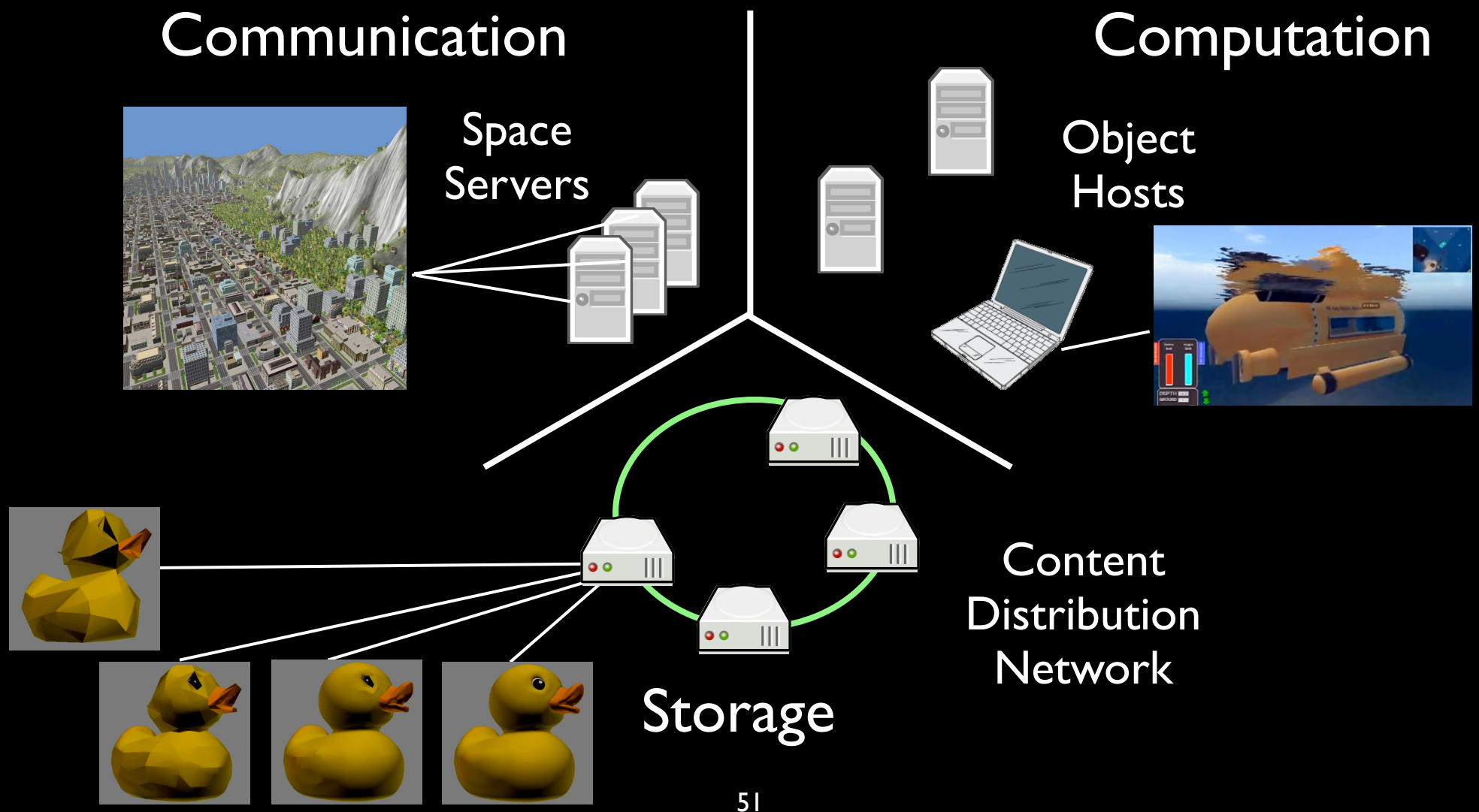
Content Pipeline

- Check COLLADA model is valid
- Asynchronously upload optimized model
  - Reduces materials, atlases textures, generates triangles
  - Average draw calls reduced from 13 to 5
- Asynchronously upload progressive model
  - Base mesh at low detail
  - Download progressive stream to increase detail
  - Allows fine-grained client control of system resources

Content Pipeline

- Check COLLADA model is valid
- Asynchronously upload optimized model
  - Reduces materials, atlases textures, generates triangles
  - Average draw calls reduced from 13 to 5
- Asynchronously upload progressive model
  - Base mesh at low detail
  - Download progressive stream to increase detail
  - Allows fine-grained client control of system resources

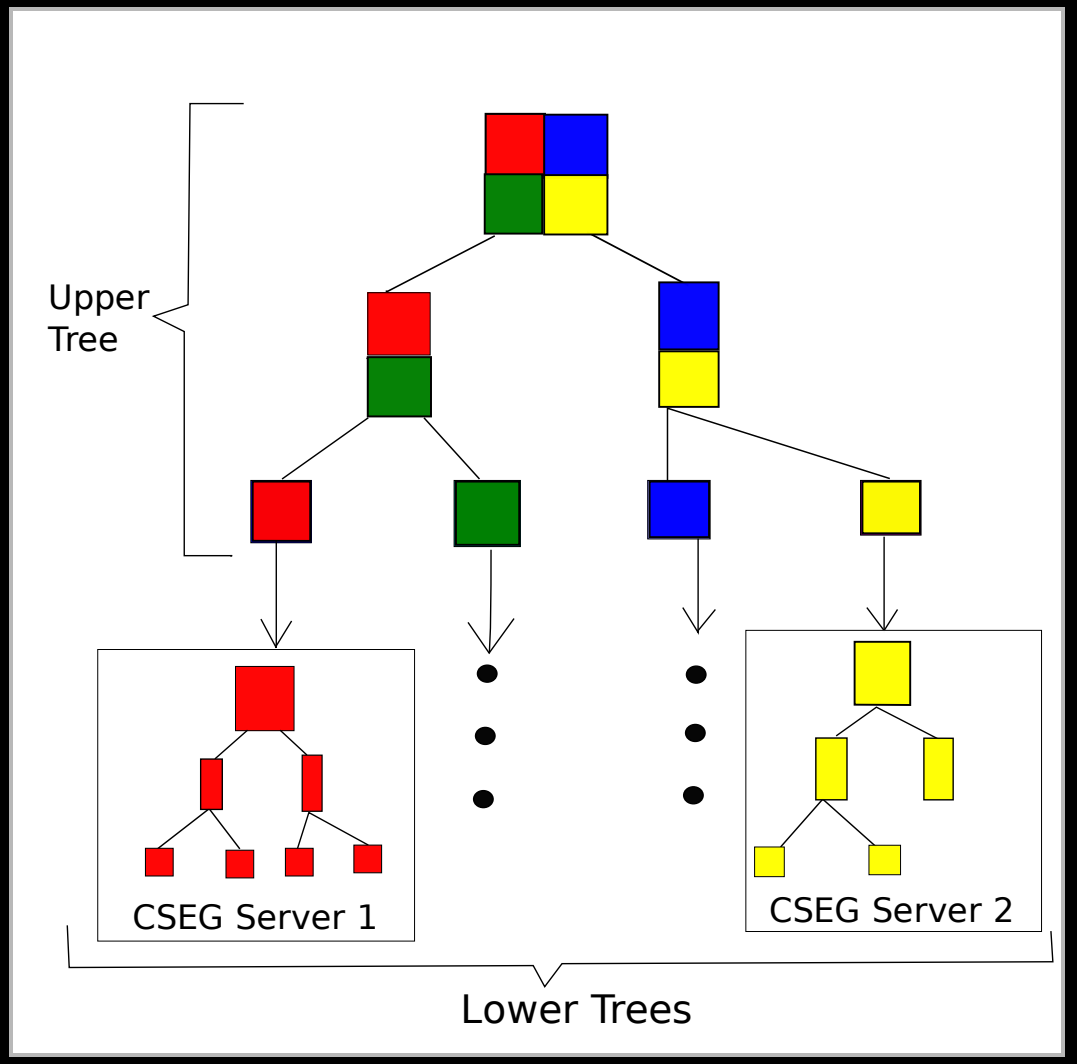
Sirikata Architecture



Questions

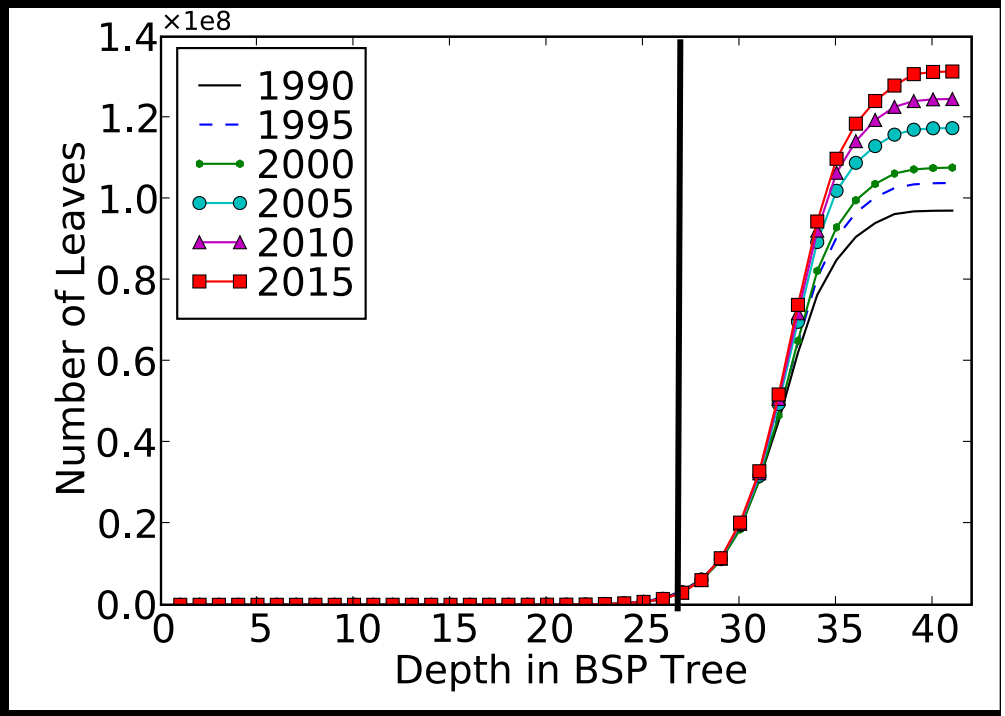
Coordinate Segmentation

- Distributed kd-tree
  - Hierarchically divides space into regions
- Tree "cut" into two levels
  - Replicated, highly stable upper tree
  - Much more dynamic lower trees managed by CSEG servers
- Load balancing done at CSEG servers based on number of avatars



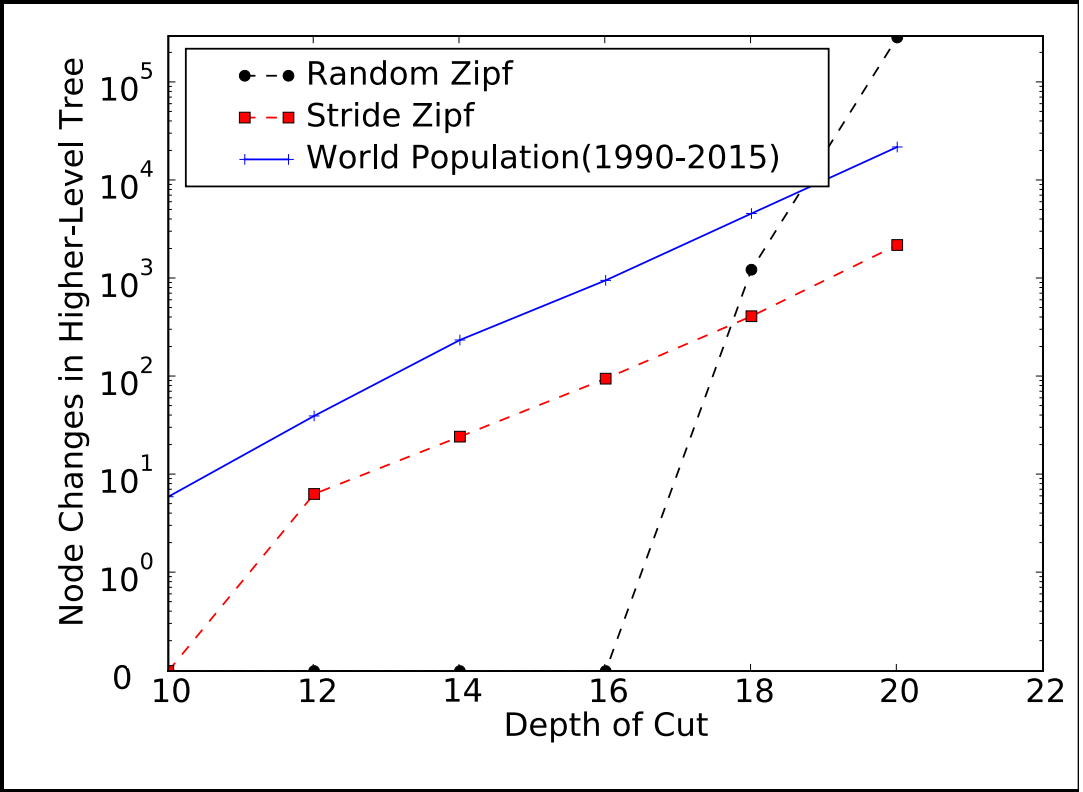
Evaluation

- kd-tree is "split-axis"
  - Not perfect, but still pretty good...
  - Graph characterizes a kd-tree over world population from 1990-2015
    - Depth remains bounded by 41; deeper than a perfect tree by 50%
    - Average depth remains about the same



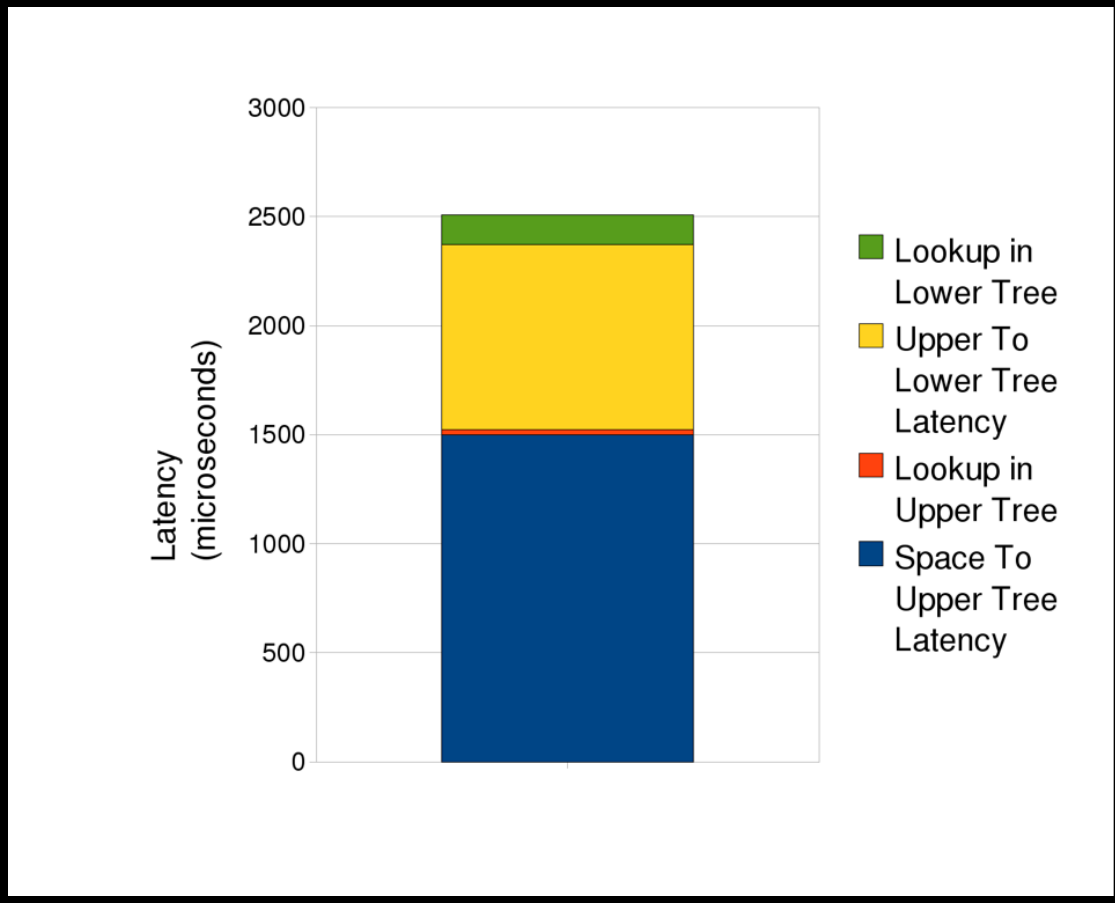
Upper Tree Stability

- With cut at a suitable depth, upper tree remains very stable



Performance

- Currently, latency for a query is around 2.5 ms
  - Most of it is network latency
- With four CSEG servers, currently sustains ~5000 queries per second



Object Segmentation

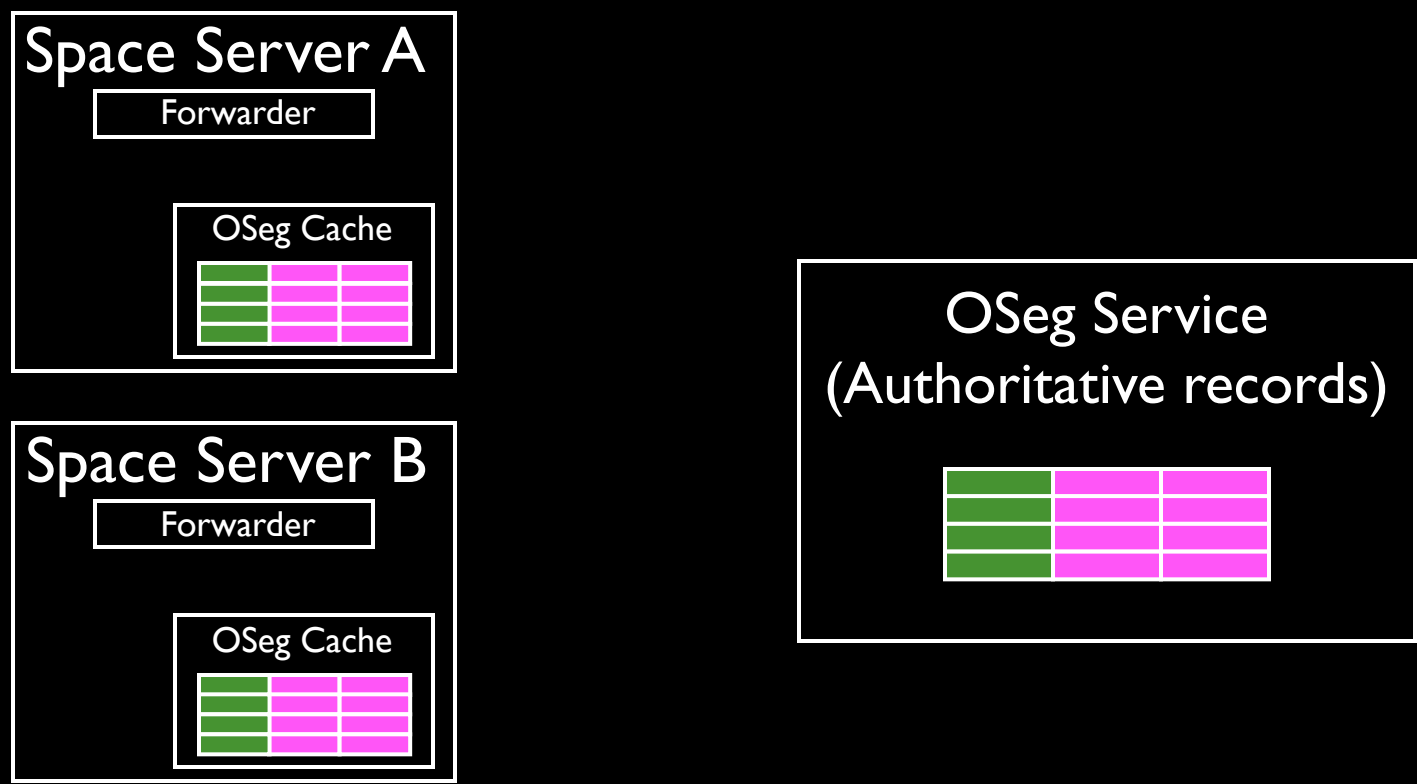
- The space server forwarder must know which space server is authoritative for the recipient.
- OSeg maintains presence to space server mappings:

Presence ID	Server connected to ID	Presence radius

- Each record is ~50B: heavy caching is feasible.
- Stale entries are easy to handle (extra hop).

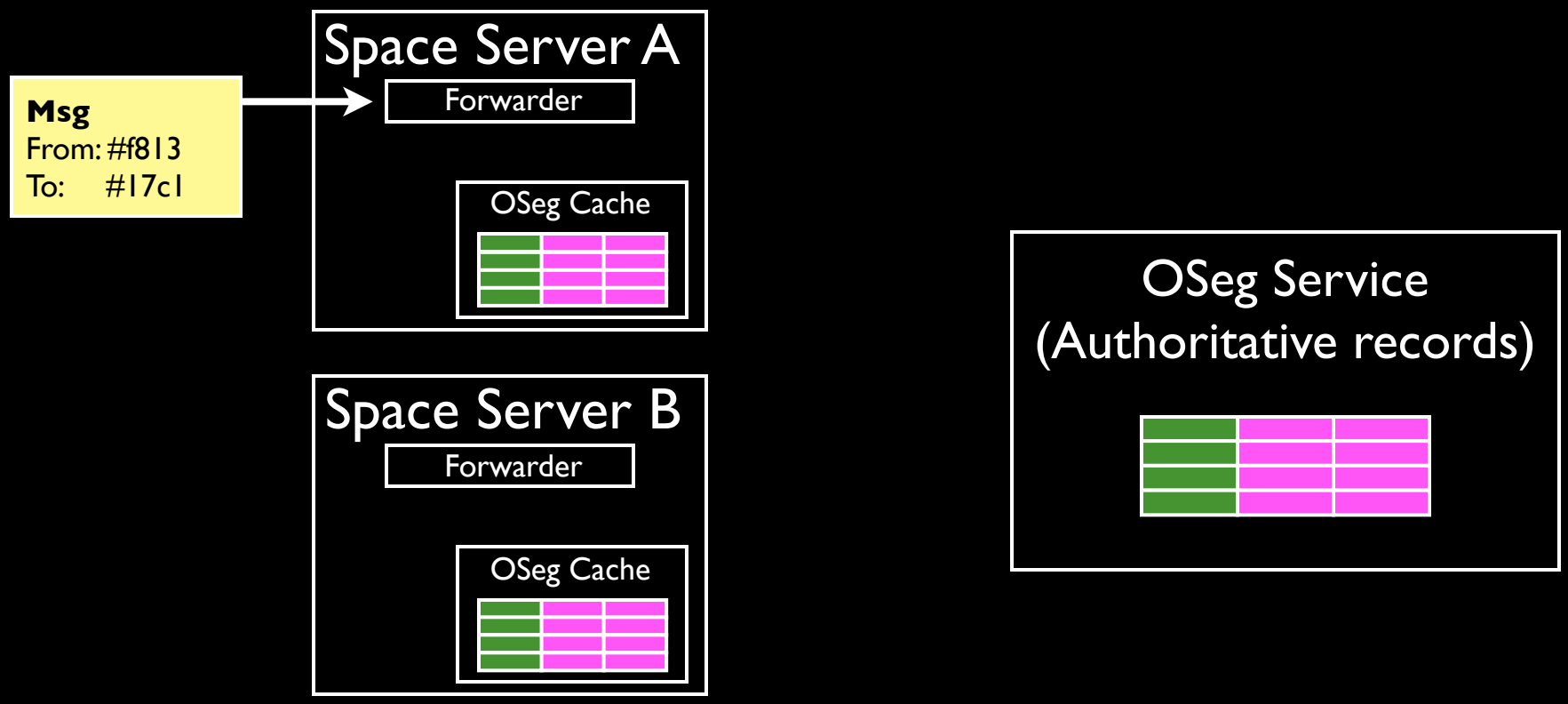
Object Segmentation (OSeg)

Presence ID	Server connected to ID	Presence radius



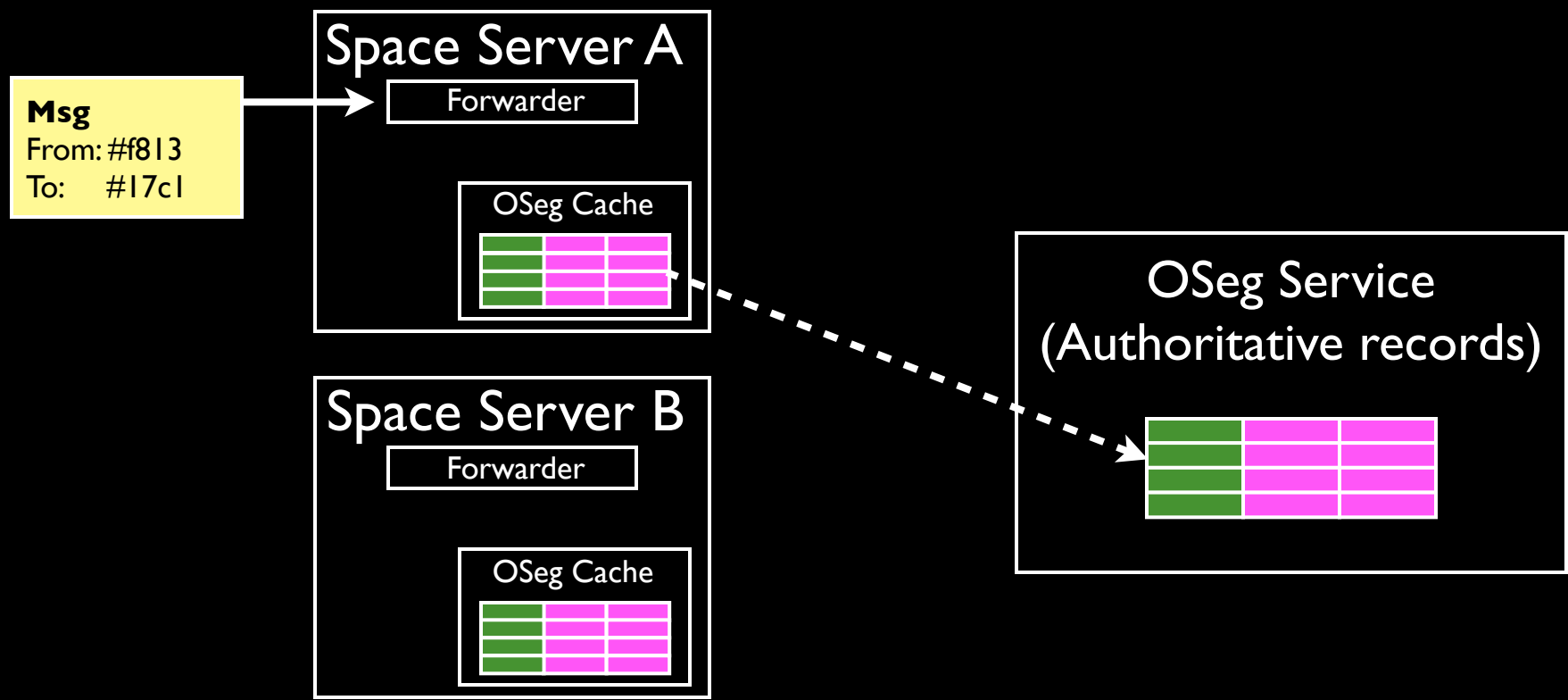
Object Segmentation (OSeg)

Presence ID	Server connected to ID	Presence radius



Object Segmentation (OSeg)

Presence ID	Server connected to ID	Presence radius



Object Segmentation (OSeg)

Presence ID	Server connected to ID	Presence radius