

Customizable and Extensible Deployment of Mobile/Cloud Applications

Irene Zhang

Adriana Szekeres Dana Van Aken Isaac Ackerman,
Steven D. Gribble Arvind Krishnamurthy Henry M. Levy

University of Washington

Once upon a time, applications were..



single user, single platform, and single node.

Today, applications are ...



multi-user, multi-platform and multi-node

Application programmers face new challenges.



How do I coordinate data and computation
across nodes and platforms?

How do I hide performance
limitations and failures?



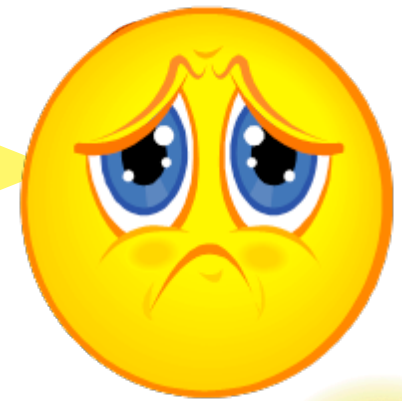
How do I manage different
programming environments
and hardware resources?

Application programmers face new challenges.



How do I coordinate data and computation across nodes and platforms?

How do I hide performance limitations and failures?



How do I manage different programming environments and hardware resources?

These sound like distributed systems problems!



Application programmers face new challenges.



How do I coordinate data and computation across nodes and platforms?

How do I hide performance limitations and failures?



How do I manage different programming environments and hardware resources?



These sound like distributed systems problems!



**I ♥ MY
SYSTEMS
PROGRAMMER**

Application programmers have different requirements.



I need reliable RPC.

No, I want caching!



I need distributed objects.

I need code-offloading.

But, I want replication!



Application programmers have different requirements.



I need reliable RPC.

No, I want caching!



I need distributed objects.

I need code-offloading.

But, I want replication!



They are left implementing distributed deployment tasks!

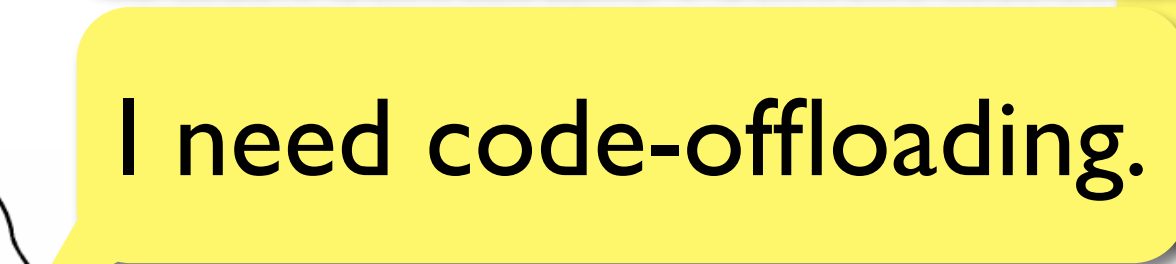
Application programmers have different requirements.



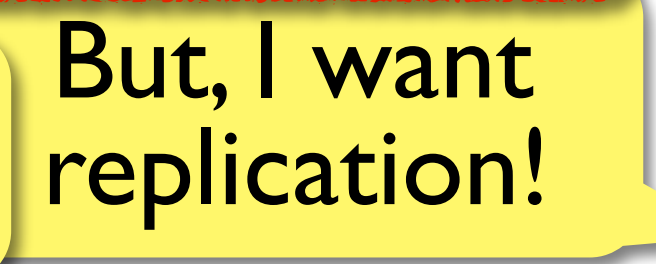
I need reliable RPC.



**PROFESSIONAL SYSTEMS
PROGRAMMER REQUIRED.
DO NOT ATTEMPT AT HOME.**



I need code-offloading.



But, I want replication!



**They are left implementing
distributed deployment tasks!**

Sapphire

**A new programming system
for deploying
mobile/cloud applications.**

Our Goals

- 1. Separate application logic from deployment code.**
- 2. Allow programmers to easily choose and change application deployment.**

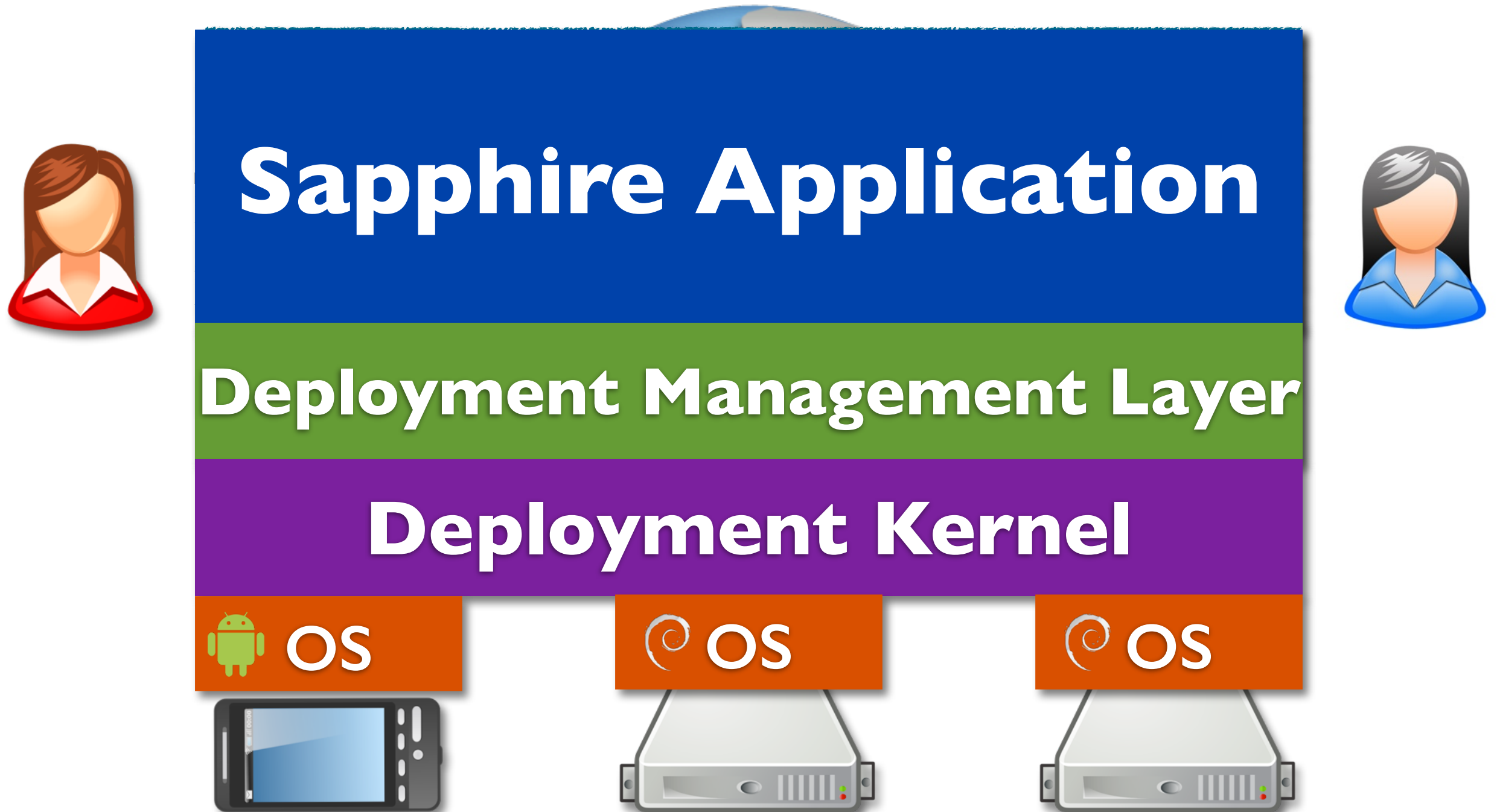
Our Solution

**A new system architecture
that supports
pluggable and extensible
deployment managers.**

Outline

1. Sapphire Architecture
2. Deployment Managers
3. Experience and Evaluation

Sapphire Architecture



Sapphire Application

Partitioned into ***Sapphire Objects***, which:

- Run in a single address space with RPC.
- Execute anywhere and move transparently.
- Provide a unit of distribution for deployment managers.

Sapphire Architecture

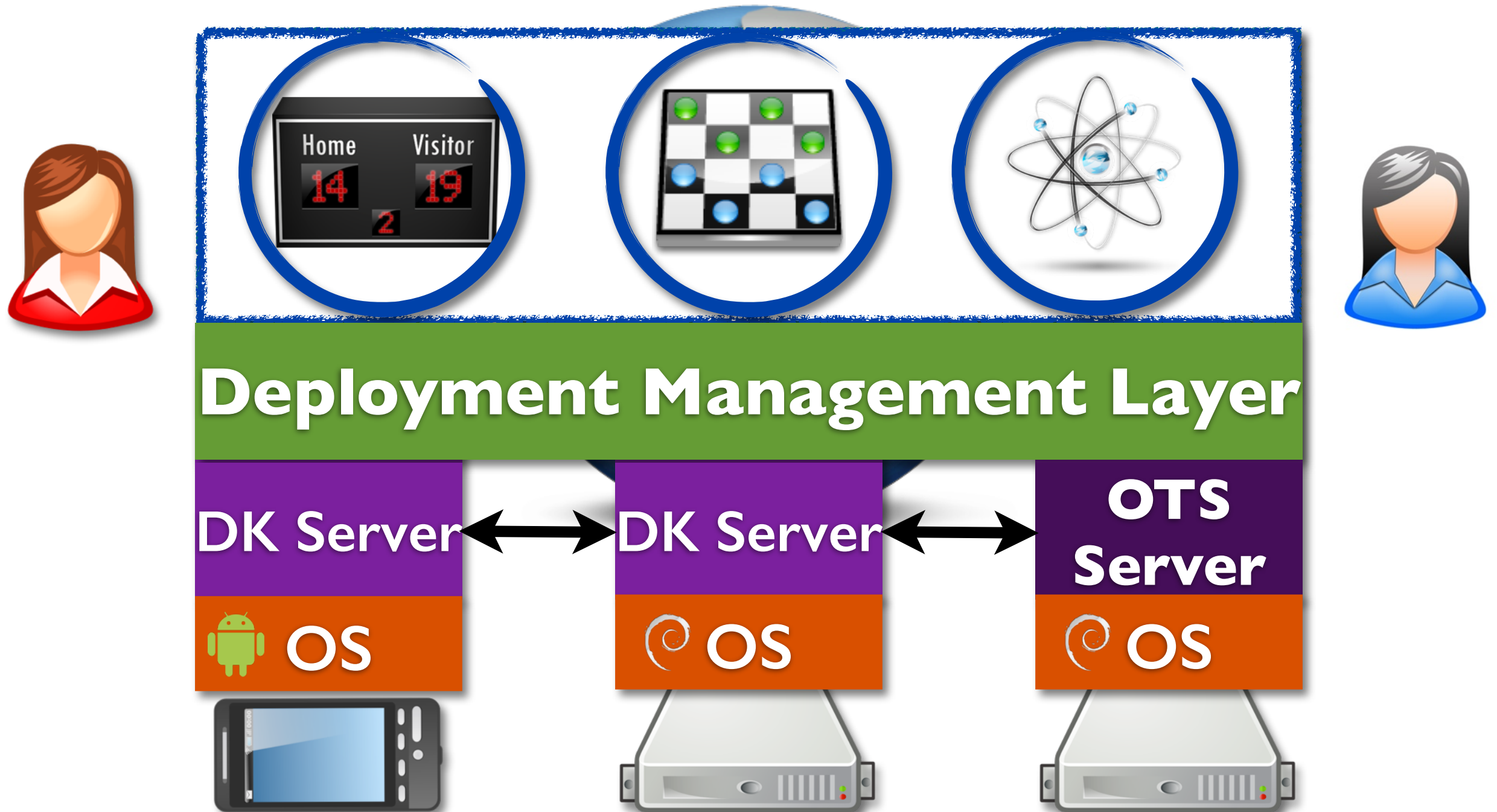


Deployment Kernel

Provides ***best-effort distribution services***, including:

- Sapphire object tracking, mobility and replication.
- Making and routing RPC to Sapphire objects.
- Managing, distributing and running deployment managers.

Sapphire Architecture

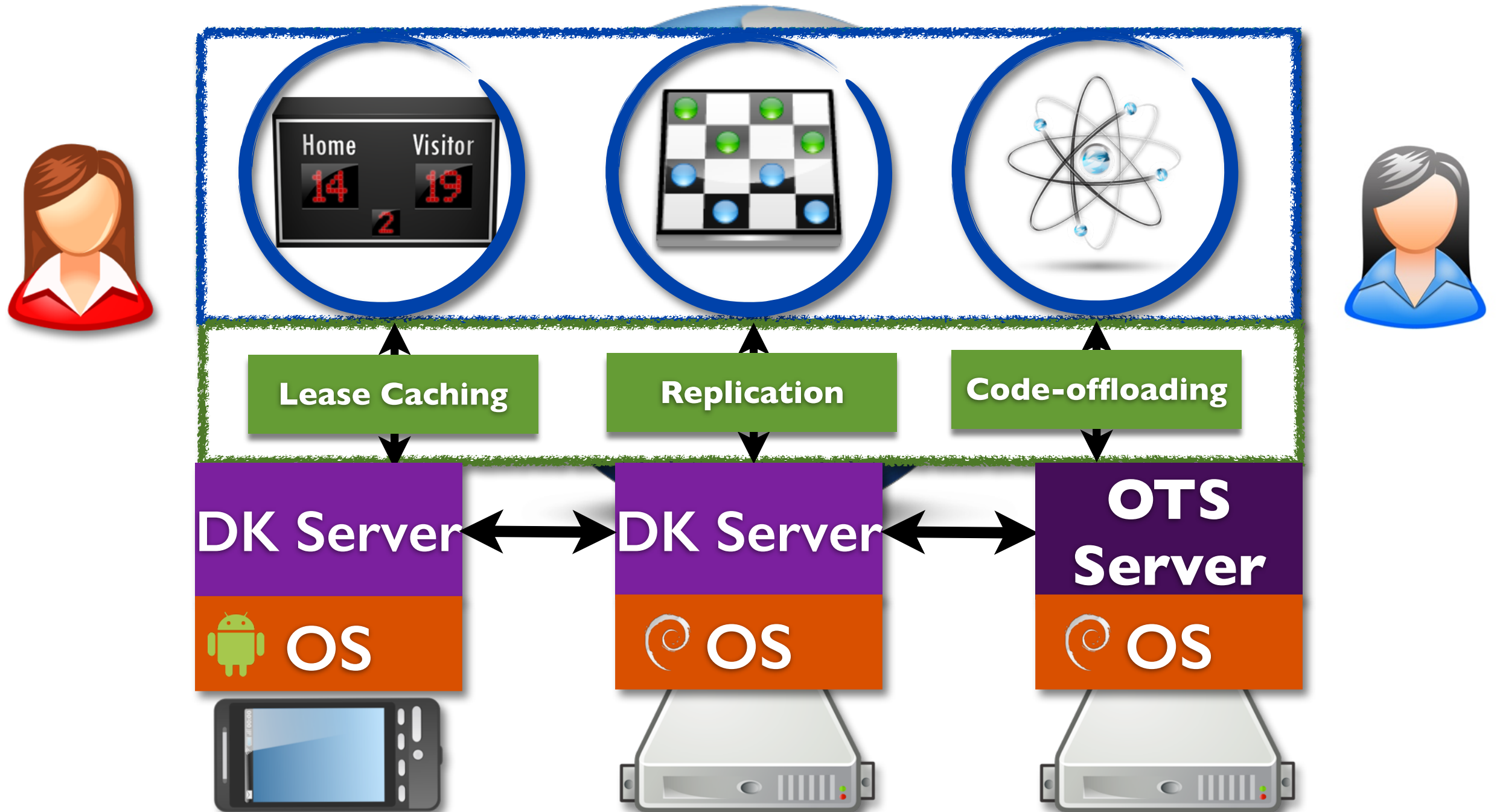


Deployment Management Layer

Consists of ***deployment managers***, which:

- Extend the functions and guarantees of the deployment kernel.
- Interpose on Sapphire object events.
- Easy to choose and change without modifying the application.

Sapphire Architecture



Sapphire Deployment Manager Library

Primitives	Caching	Serializability	Checkpoint	Replication	Mobility	Scalability
Immutable	Explicit Caching	Serializable RPC	Explicit Checkpoint	RSM-Cluster	Explicit Migration	LoadBalanced Frontend
AtLeastOnce RPC	Lease Caching	Locking Transactions	Periodic Checkpoint	RSM-Geo	Dynamic Migration	Scale-up Frontend
Keep In Place	Writethrough Caching	Optimistic Transactions	DurableRPC	RSM-P2P	Explicit Code-offload	LB Master-slave
Keep On Device	Consistent Caching		Durable Transactions		Code-offload	
Keep In Cloud						

Sapphire Deployment Manager Library

Primitives	Caching	Serializability	Checkpoint	Replication	Mobility	Scalability
Immutable	Explicit Caching	Serializable RPC	Explicit Checkpoint	RSM-Cluster	Explicit Migration	LoadBalanced Frontend
AtLeastOnce RPC	Lease Caching	Locking Transactions	Periodic Checkpoint	RSM-Geo	Dynamic Migration	Scale-up Frontend
Keep In Place	Writethrough Caching	Optimistic Transactions	DurableRPC	RSM-P2P	Explicit Code-offload	LB Master-slave
Keep On Device	Consistent Caching		Durable Transactions		Code-offload	
Keep In Cloud						

Extensible with the Sapphire Deployment Manager API!

Outline

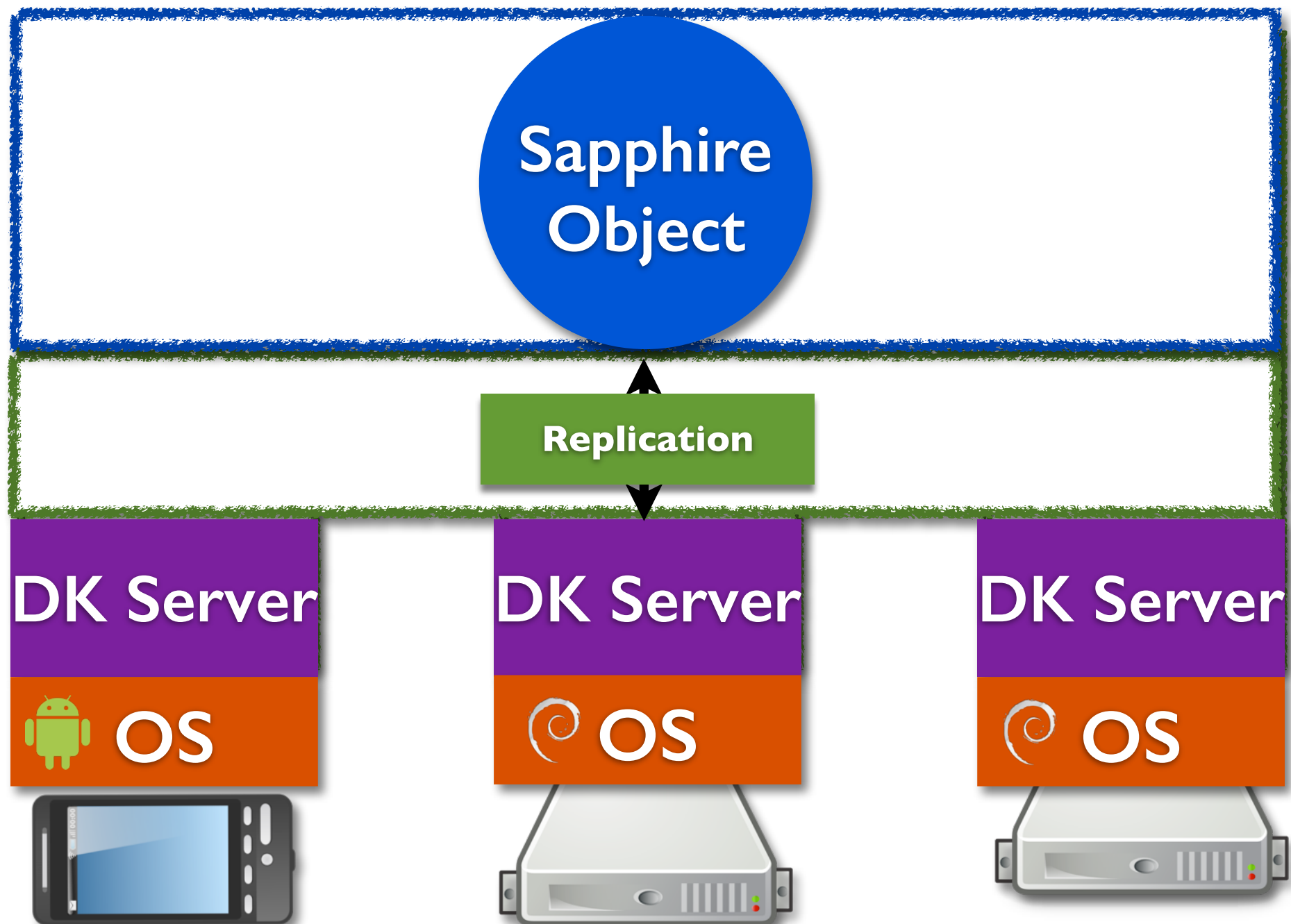
1. Sapphire Architecture
2. Deployment Managers
3. Experience and Evaluation

Deployment Manager API

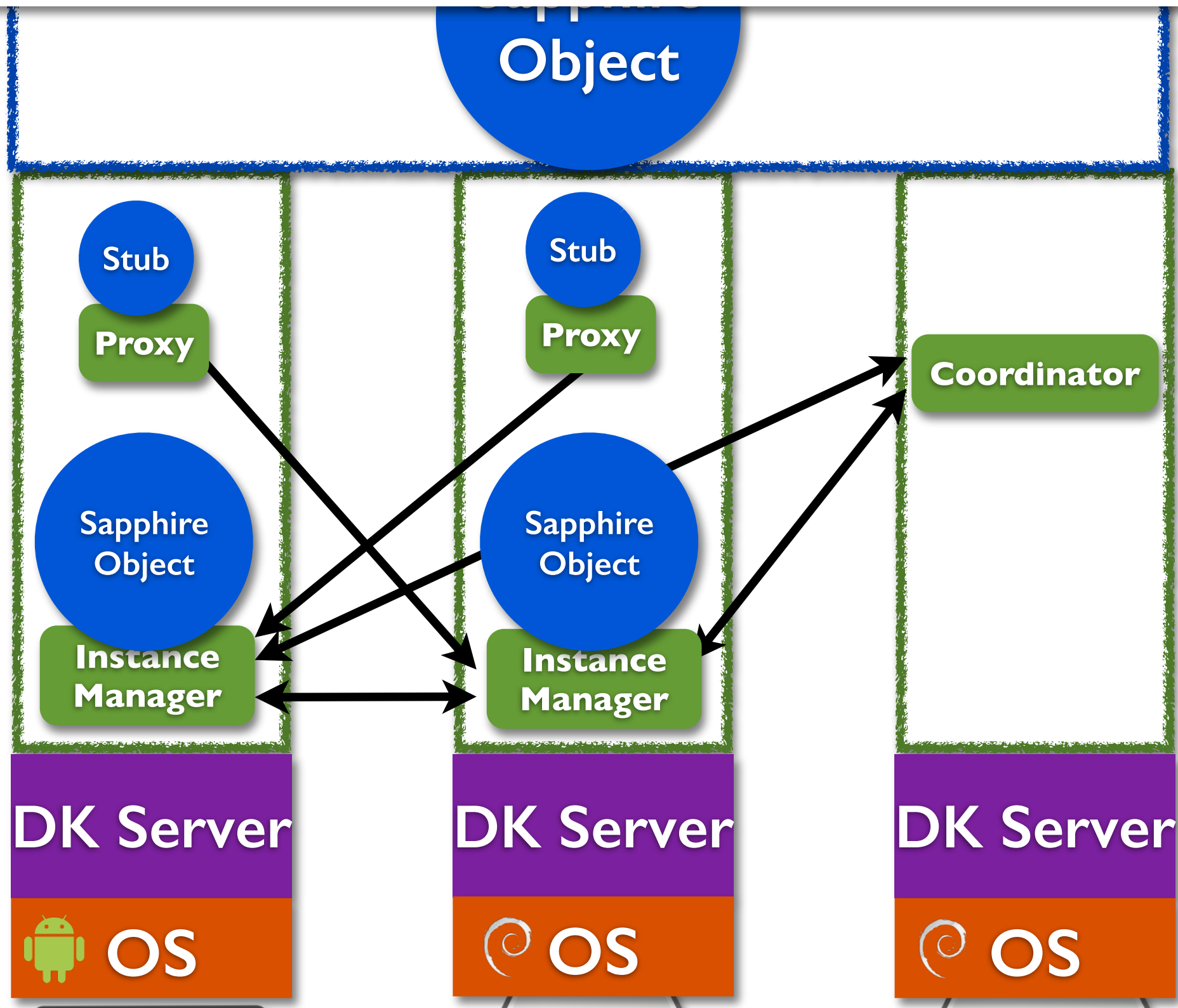
Deployment manager components, which the Sapphire kernel creates, deploys and invokes:

- **Instance Manager:** Co-located with the Sapphire Object.
- **Proxy:** Co-located with remote references to the Sapphire Object.
- **Coordinator:** Co-located with fault-tolerant Object Tracking Service (OTS).

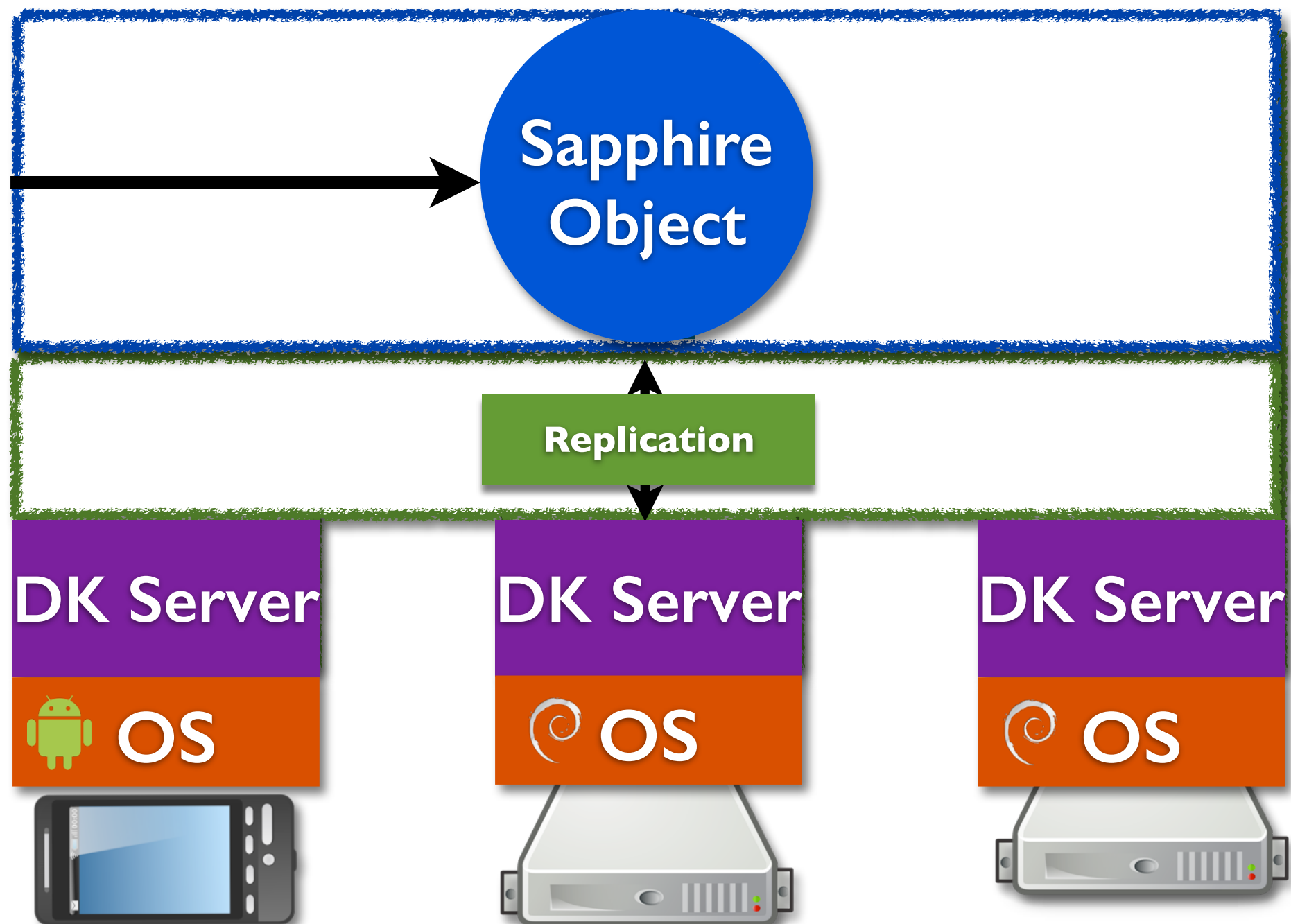
Deployment Manager Architecture



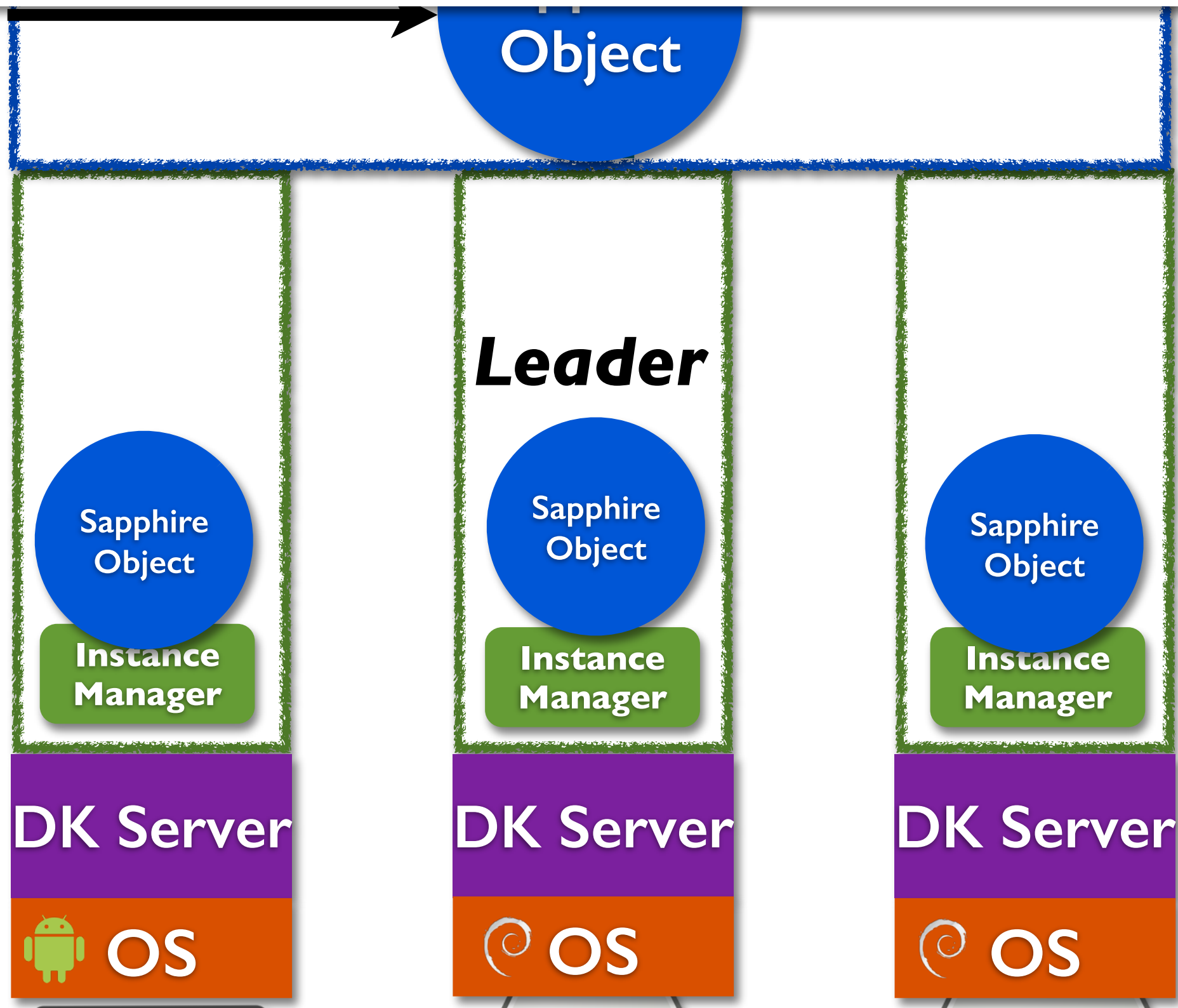
Deployment Manager Architecture



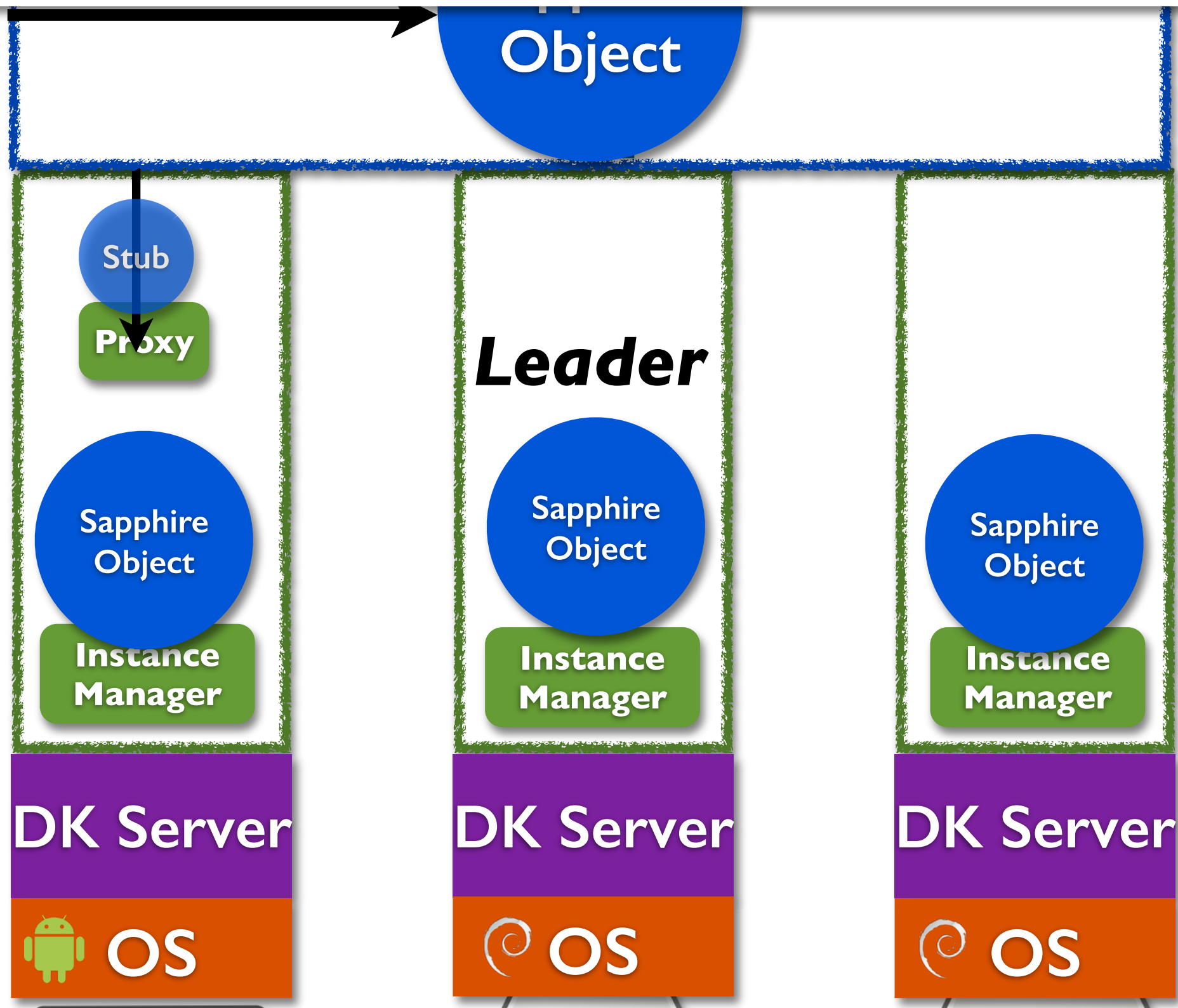
Replicating a Sapphire Object



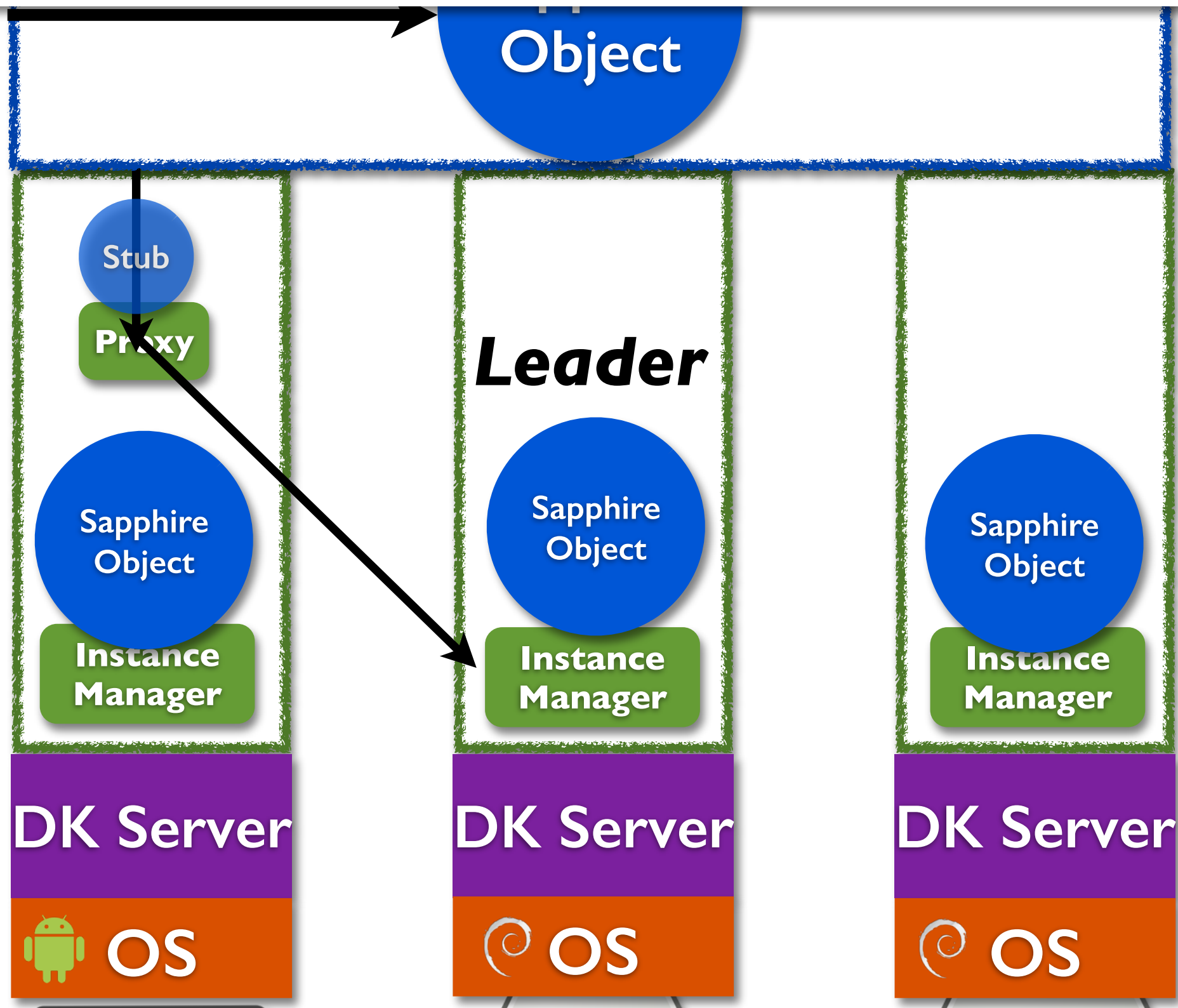
Replicating a Sapphire Object



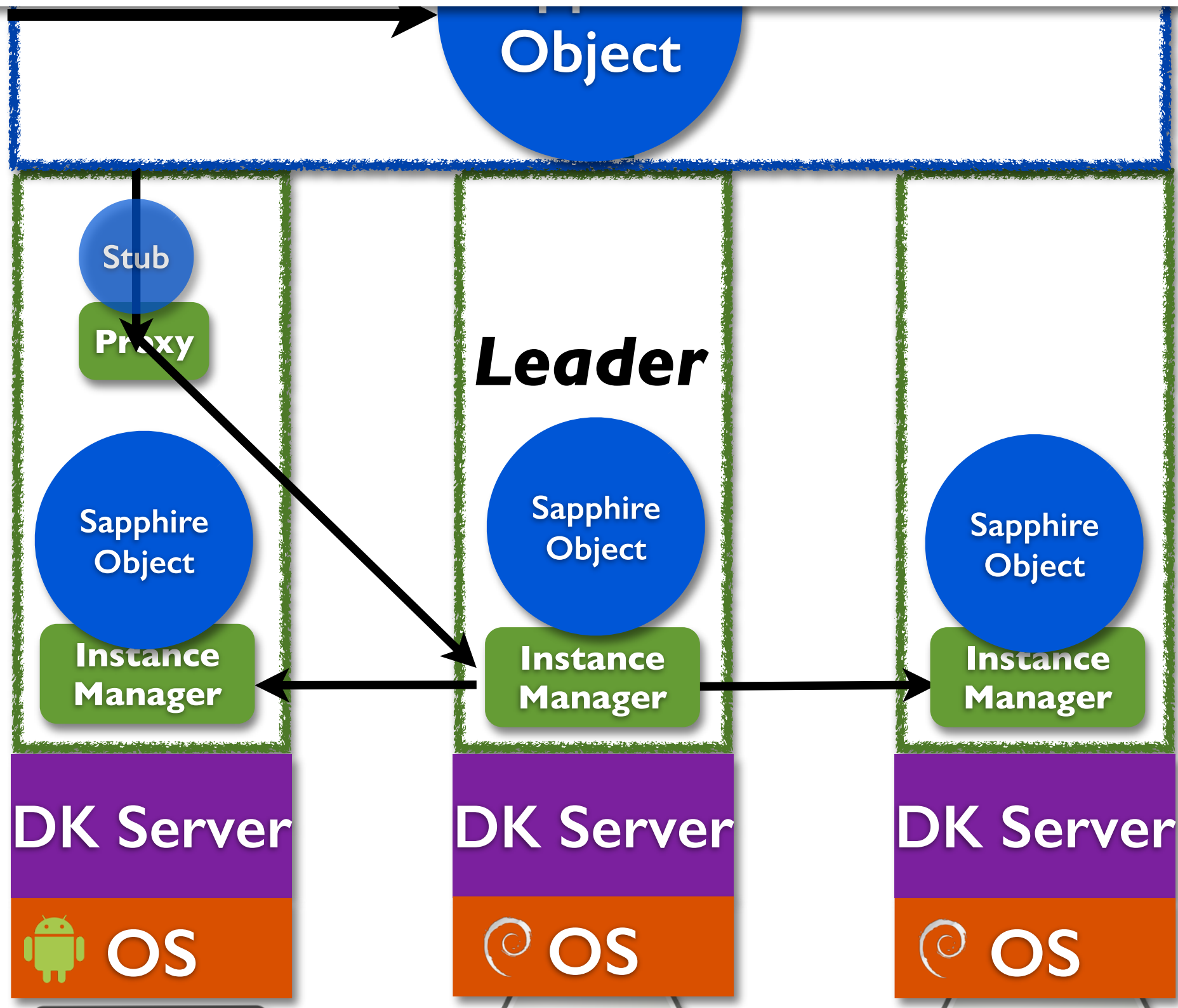
Replicating a Sapphire Object



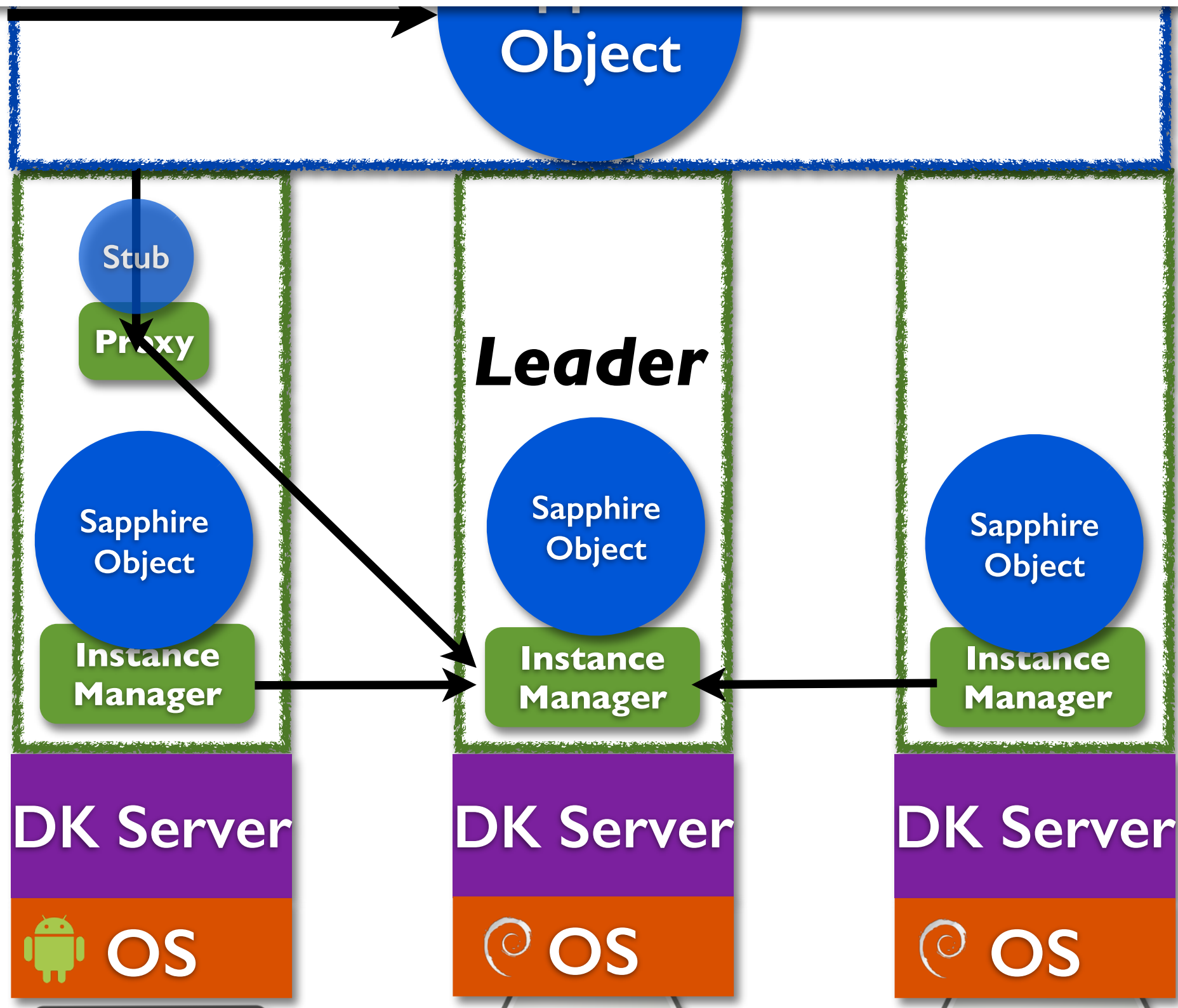
Replicating a Sapphire Object



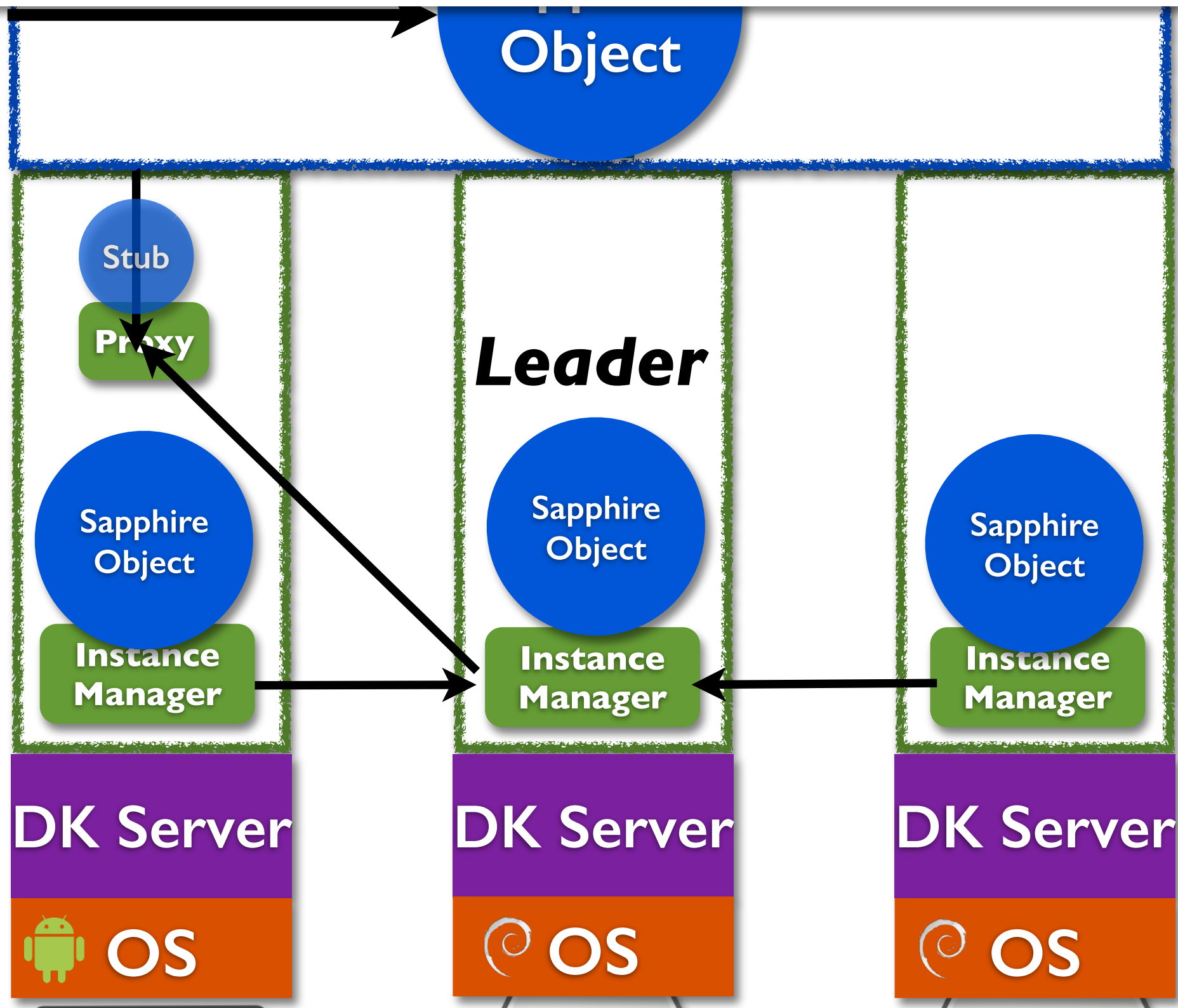
Replicating a Sapphire Object



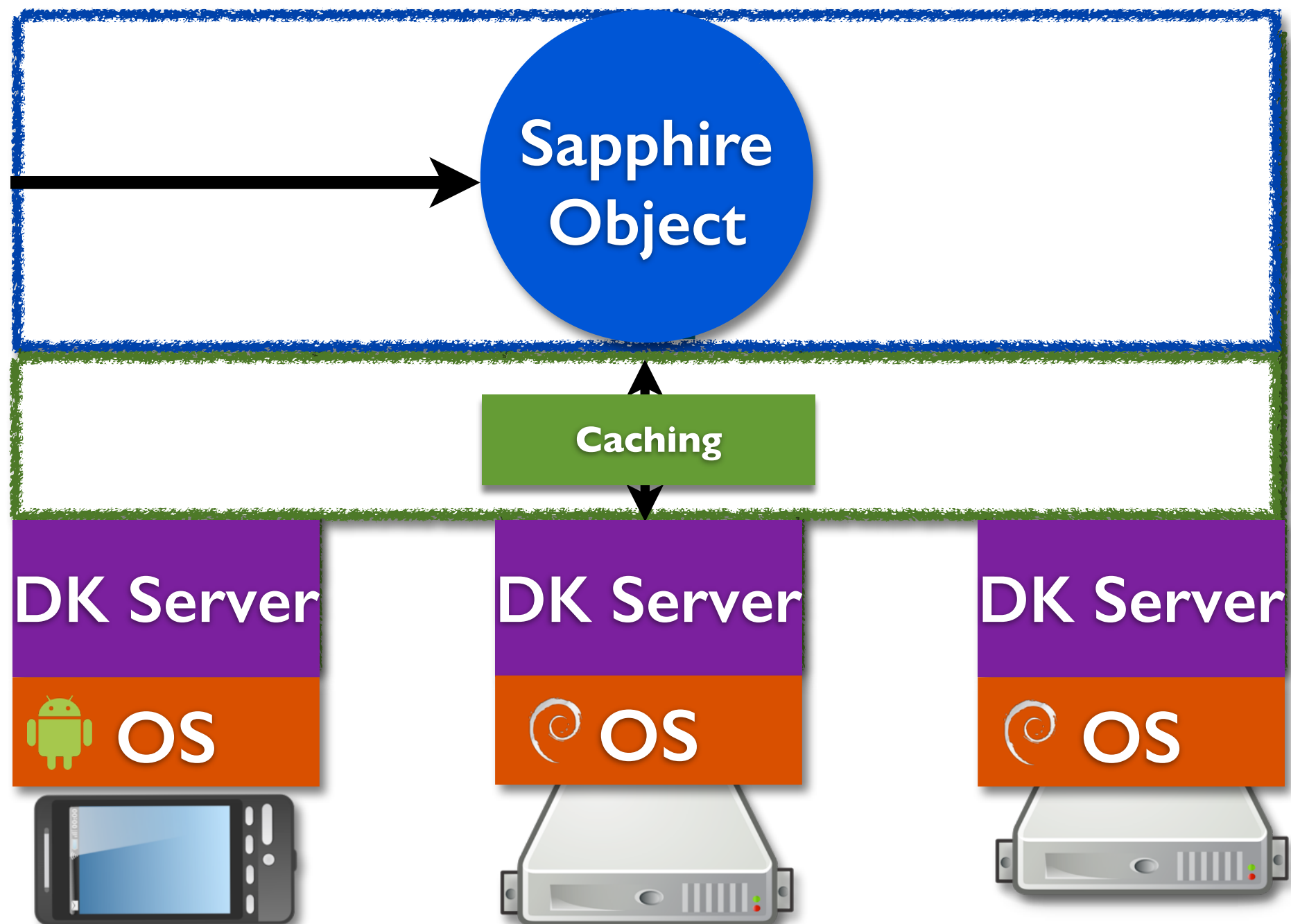
Replicating a Sapphire Object



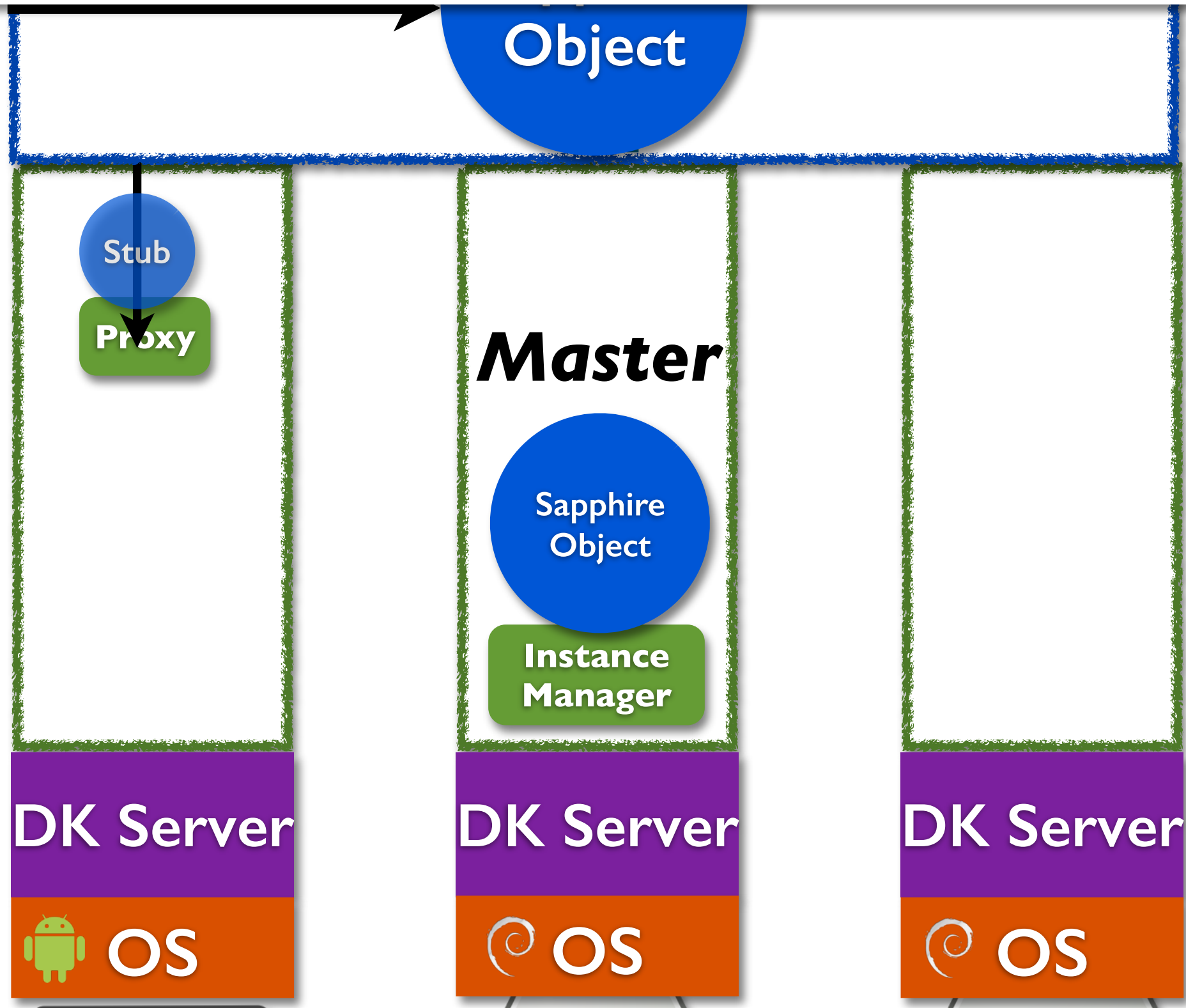
Replicating a Sapphire Object



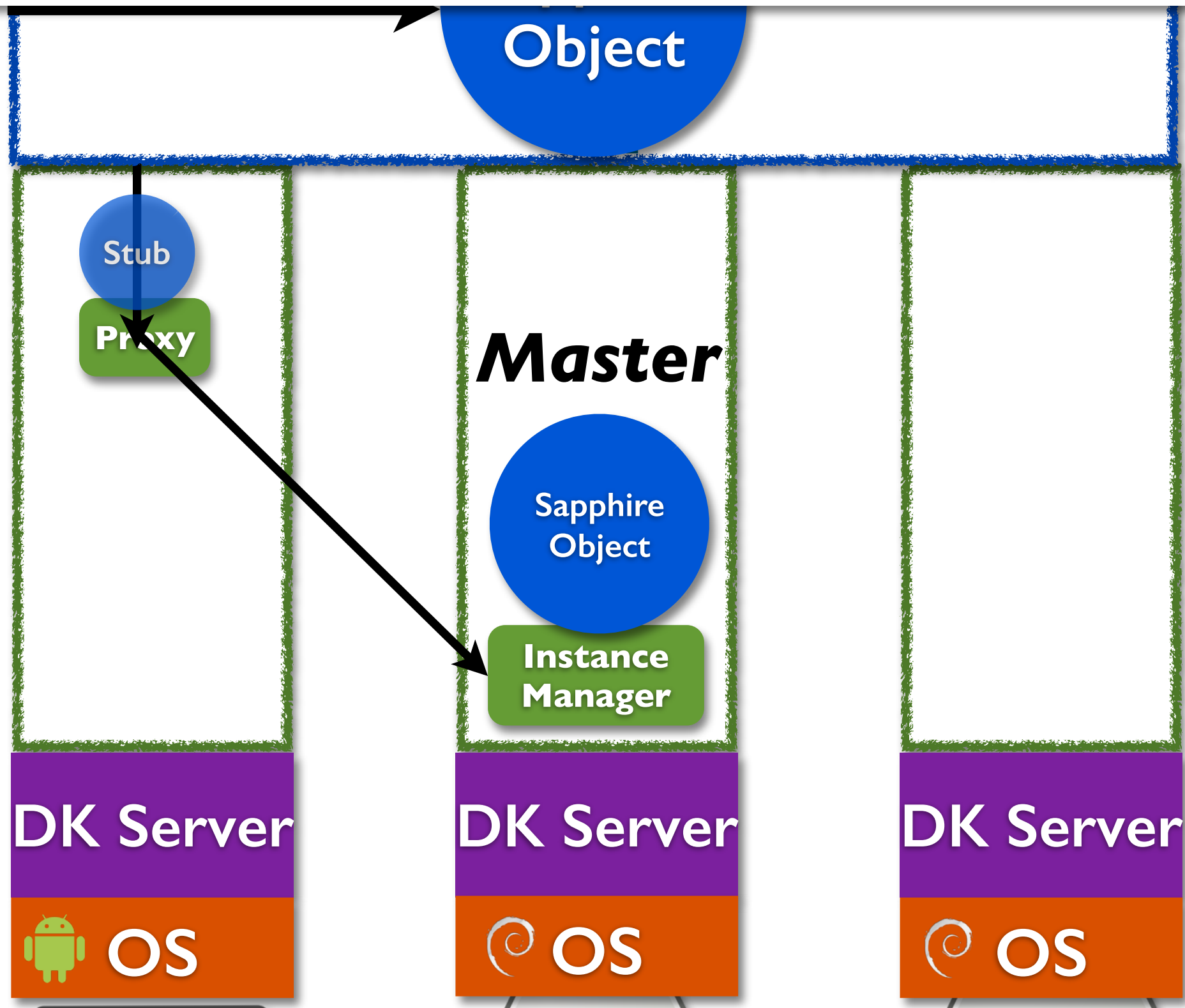
Caching a Sapphire Object



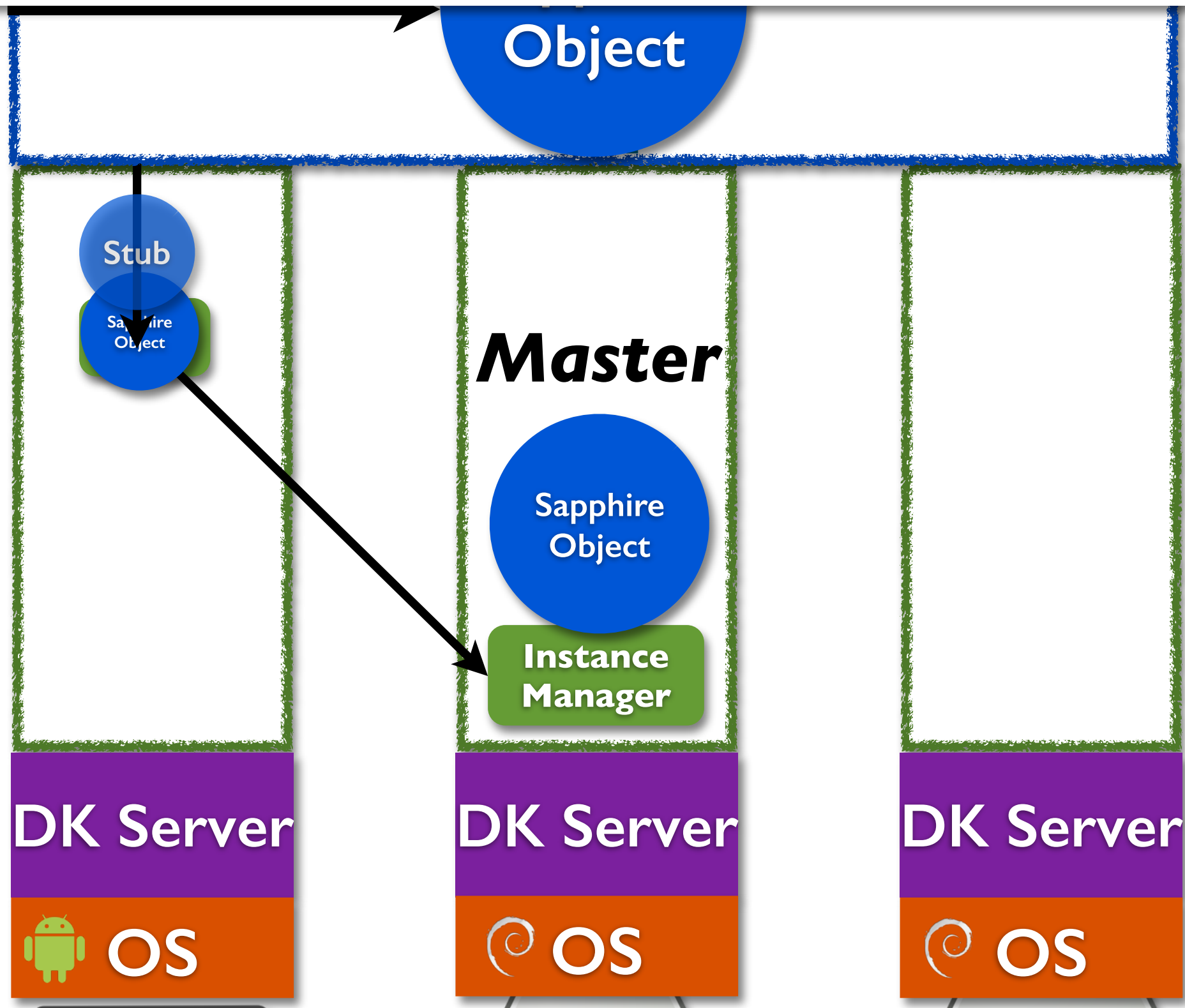
Caching a Sapphire Object



Caching a Sapphire Object



Caching a Sapphire Object



Outline

- 1. Sapphire Architecture**
- 2. Deployment Managers**
- 3. Experience and Evaluation**

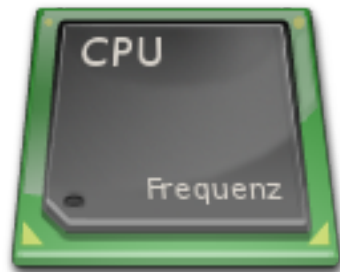
Experimental Setup



Dell Server

Nexus 7

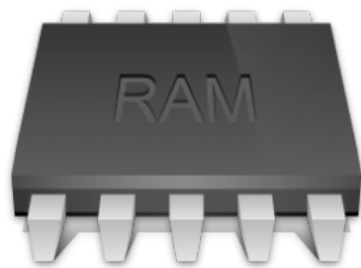
Nexus S



8-core
Intel Xeon
2GHz

4-core
ARM Cortex A9
1.3GHz

1-core
ARM Cortex A8
1GHz



8GB

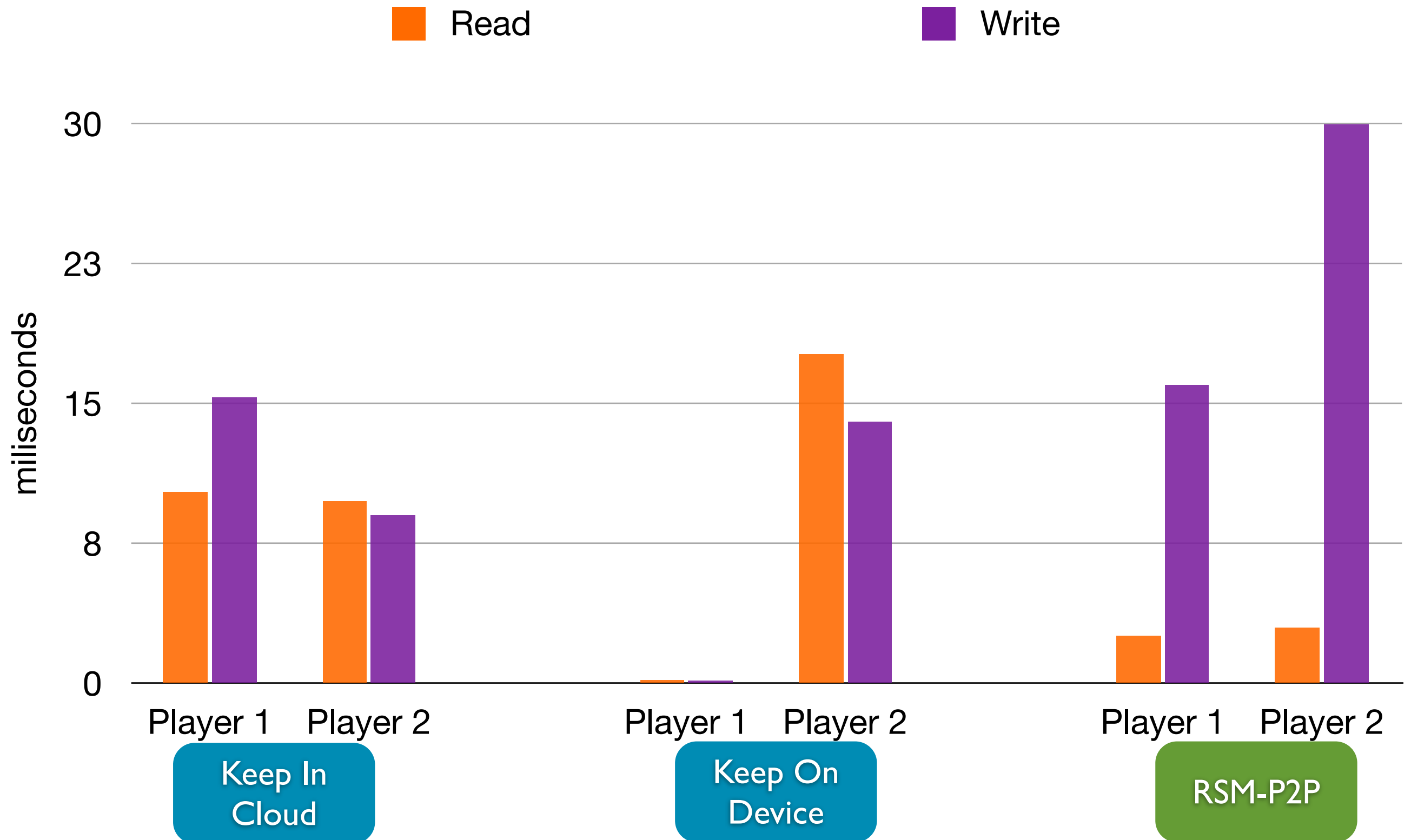
1GB

512MB

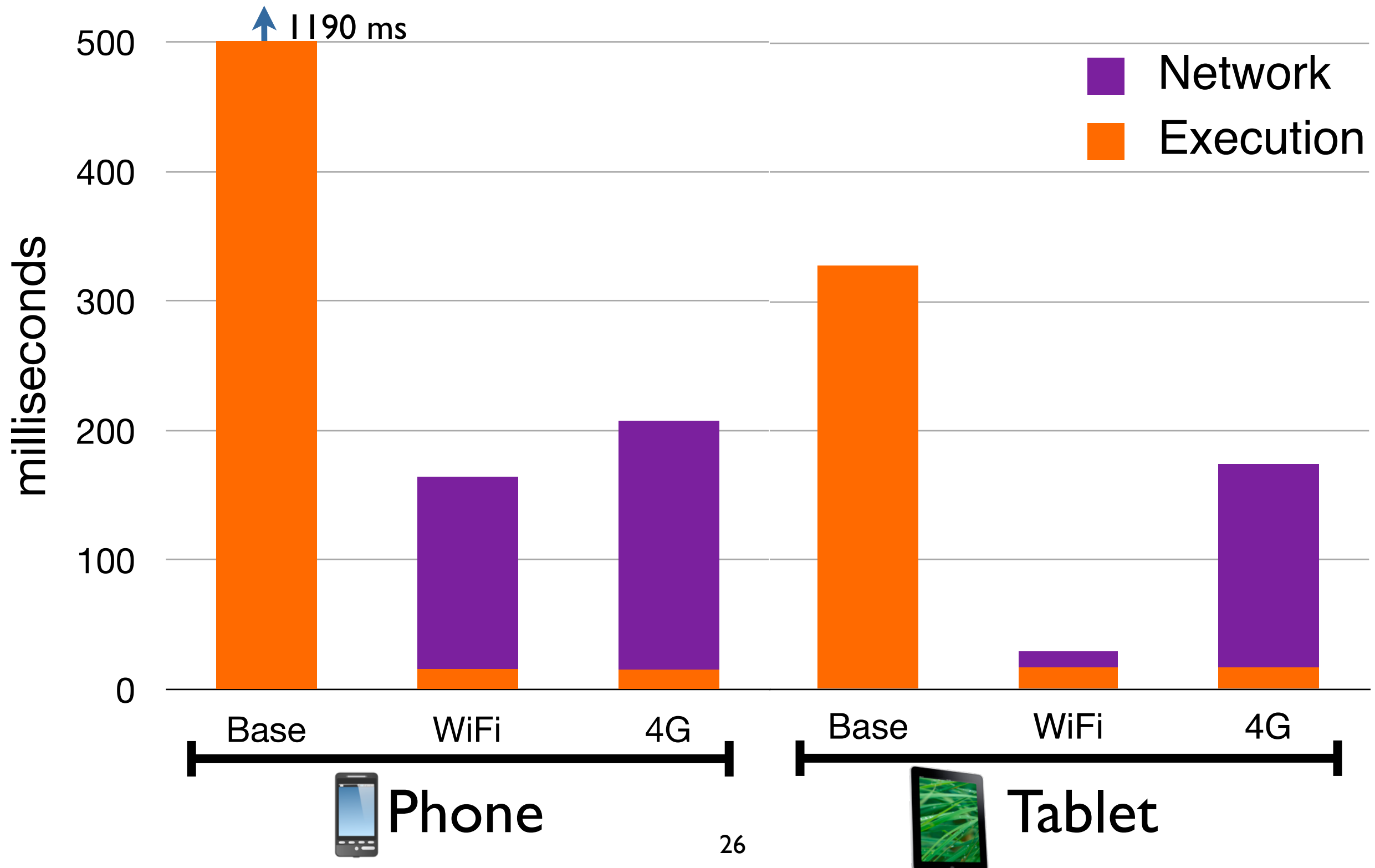
OS



Peer-to-Peer Multiplayer Game



Code-offloading for Physics Engine



Summary

Modern applications implement difficult distributed deployment tasks.

Sapphire is a new programming system for deploying mobile/cloud applications.

Deployment managers makes it easy to choose, change and build deployment.