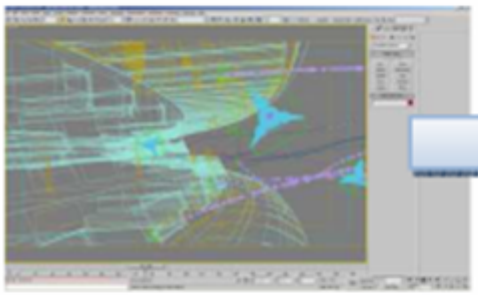
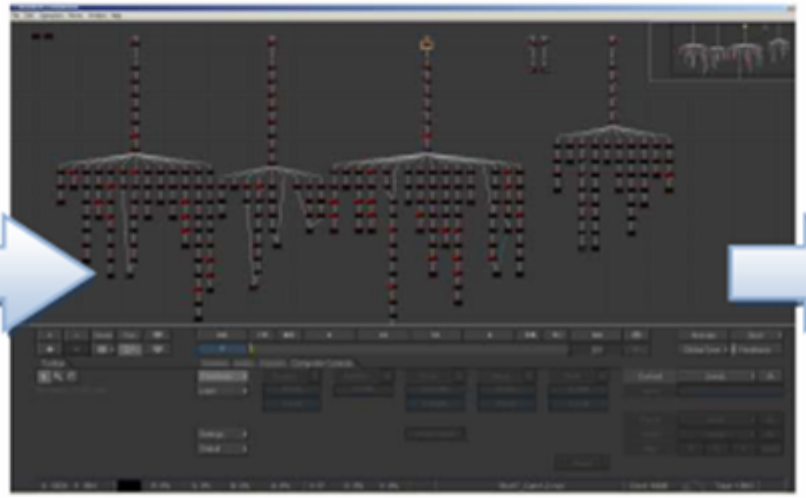
# Seminar on Scalability in Distributed Systems

# Seminar on Scalability of Distributed Systems

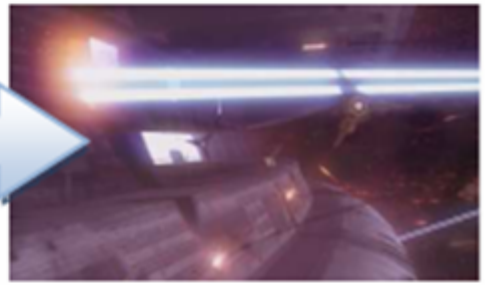
# Distributed Rendering with 3DSMAX



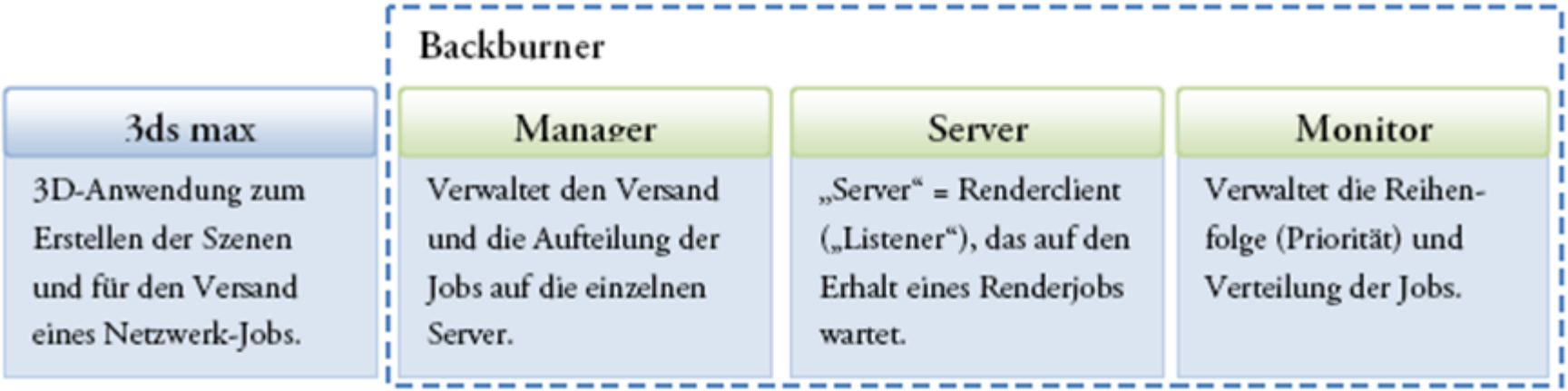
3D-Szene

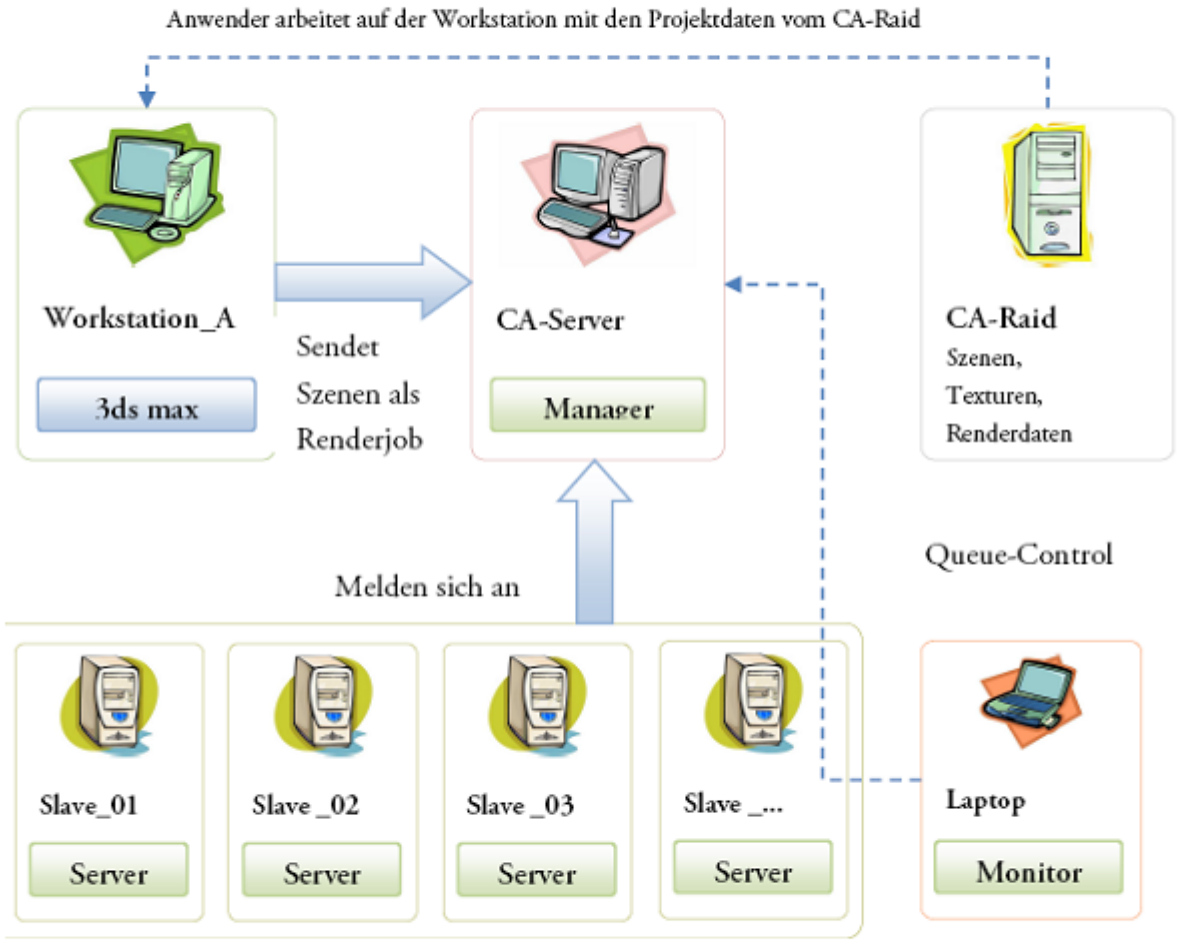


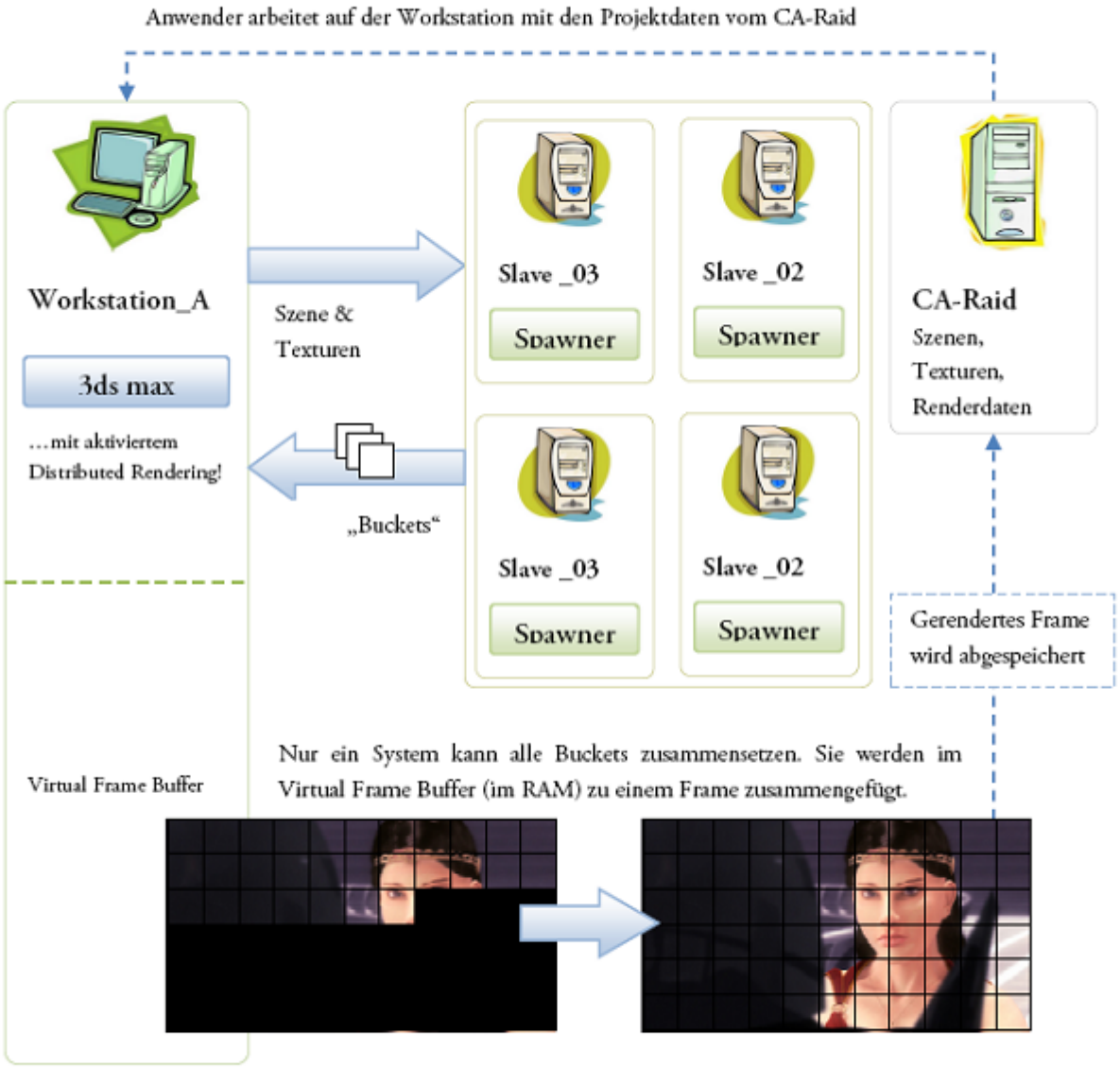
Compositing der Layer



Final

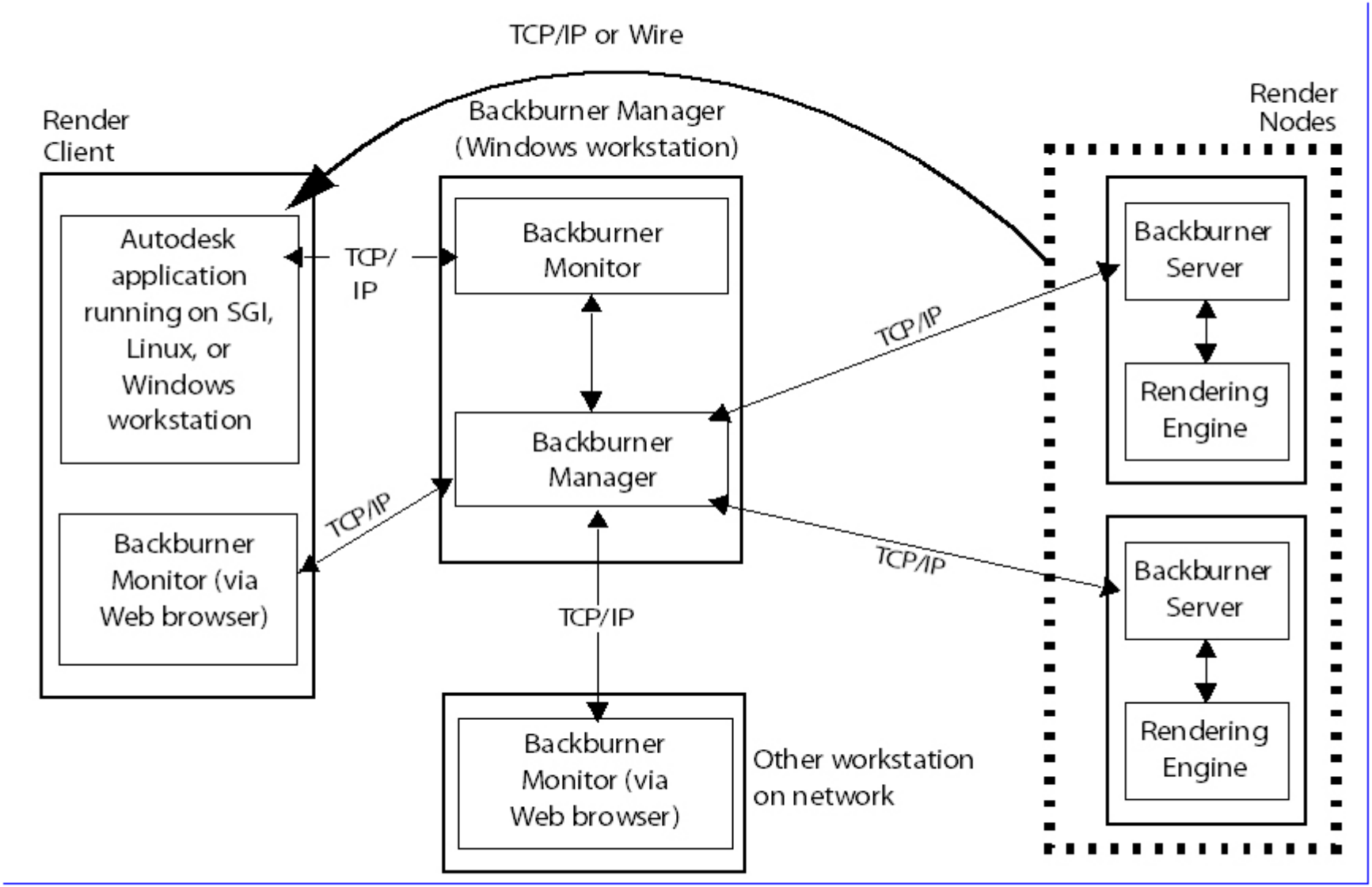

Backburner
3ds max
3D-Anwendung zum Erstellen der Szenen und für den Versand eines Netzwerk-Jobs.
Manager
Verwaltet den Versand und die Aufteilung der Jobs auf die einzelnen Server.
Server
„Server" = Renderclient („Listener"), das auf den Erhalt eines Renderjobs wartet.
Monitor
Verwaltet die Reihenfolge (Priorität) und Verteilung der Jobs.

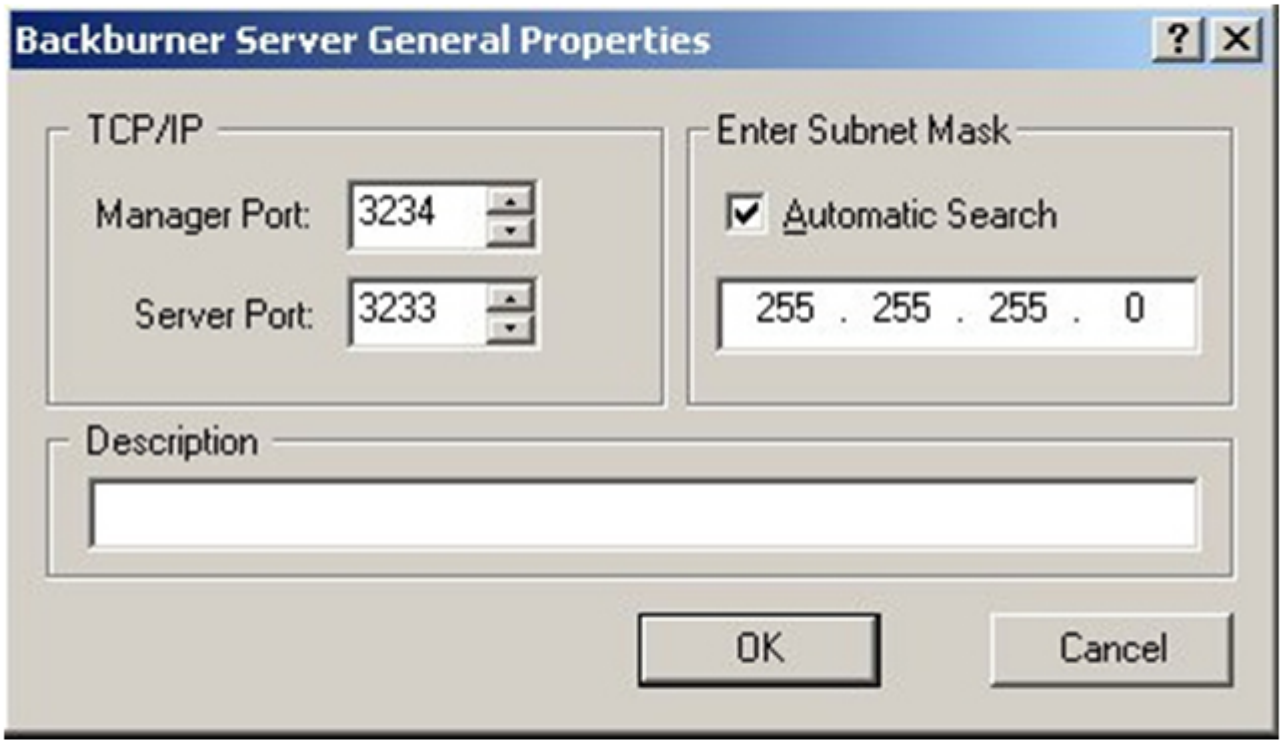

Anwender arbeitet auf der Workstation mit den Projektdaten vom CA-Raid
Workstation_A
3ds max
Sendet
Szenen als
Renderjob
CA-Server
Manager
CA-Raid
Szenen,
Texturen,
Renderdaten
Queue-Control
Melden sich an
Slave_01
Server
Slave _02
Server
Slave _03
Server
Slave _...
Server
Laptop
Monitor

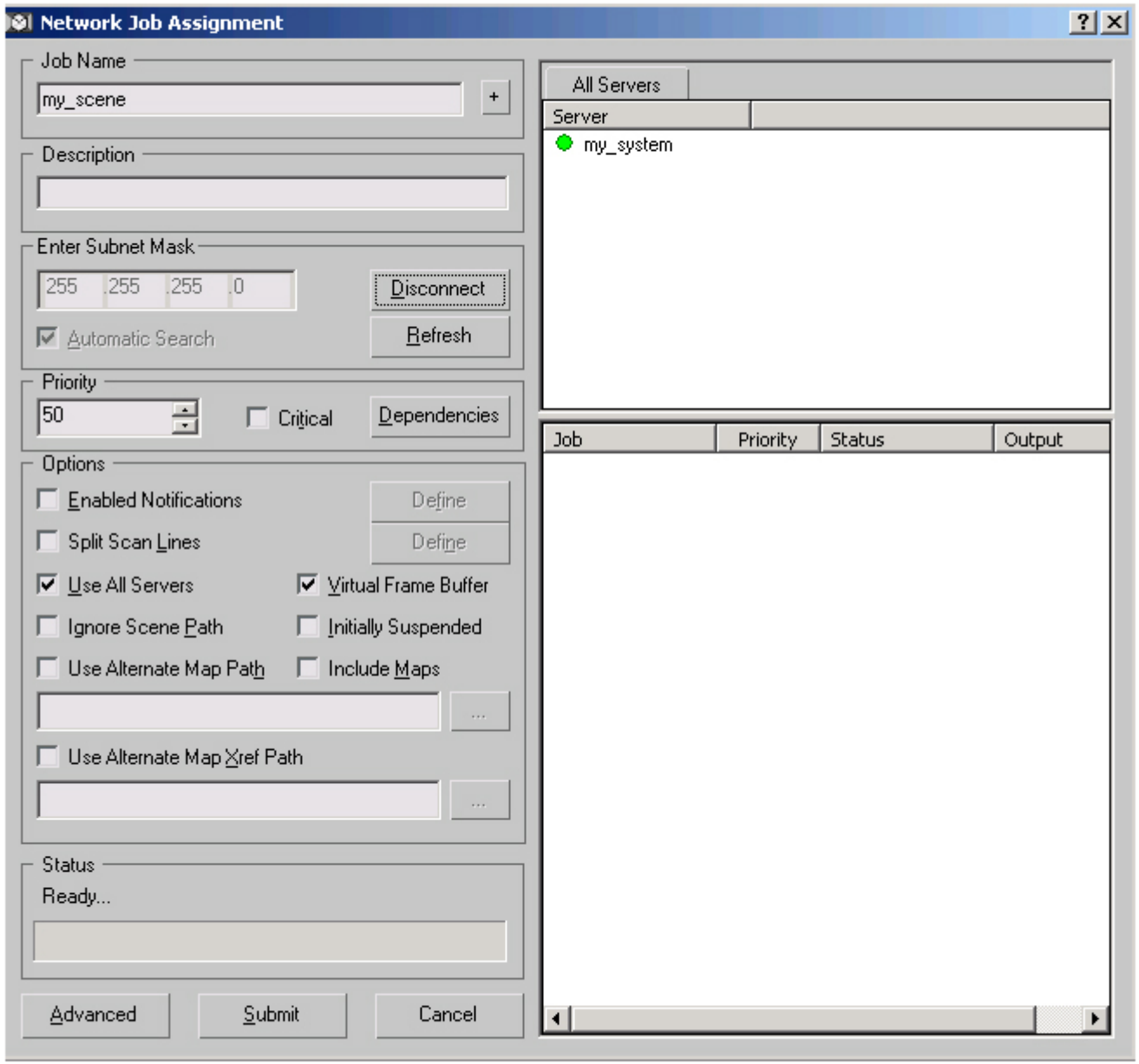

Anwender arbeitet auf der Workstation mit den Projektdaten vom CA-Raid
Workstation_A
3ds max
...mit aktiviertem Distributed Rendering!
Szene & Texturen
„Buckets“
Slave _03
Spawner
Slave _02
Spawner
Slave _03
Spawner
Slave _02
Spawner
CA-Raid
Szenen, Texturen, Renderdaten
Gerendertes Frame wird abgespeichert
Virtual Frame Buffer
Nur ein System kann alle Buckets zusammensetzen. Sie werden im Virtual Frame Buffer (im RAM) zu einem Frame zusammengefügt.

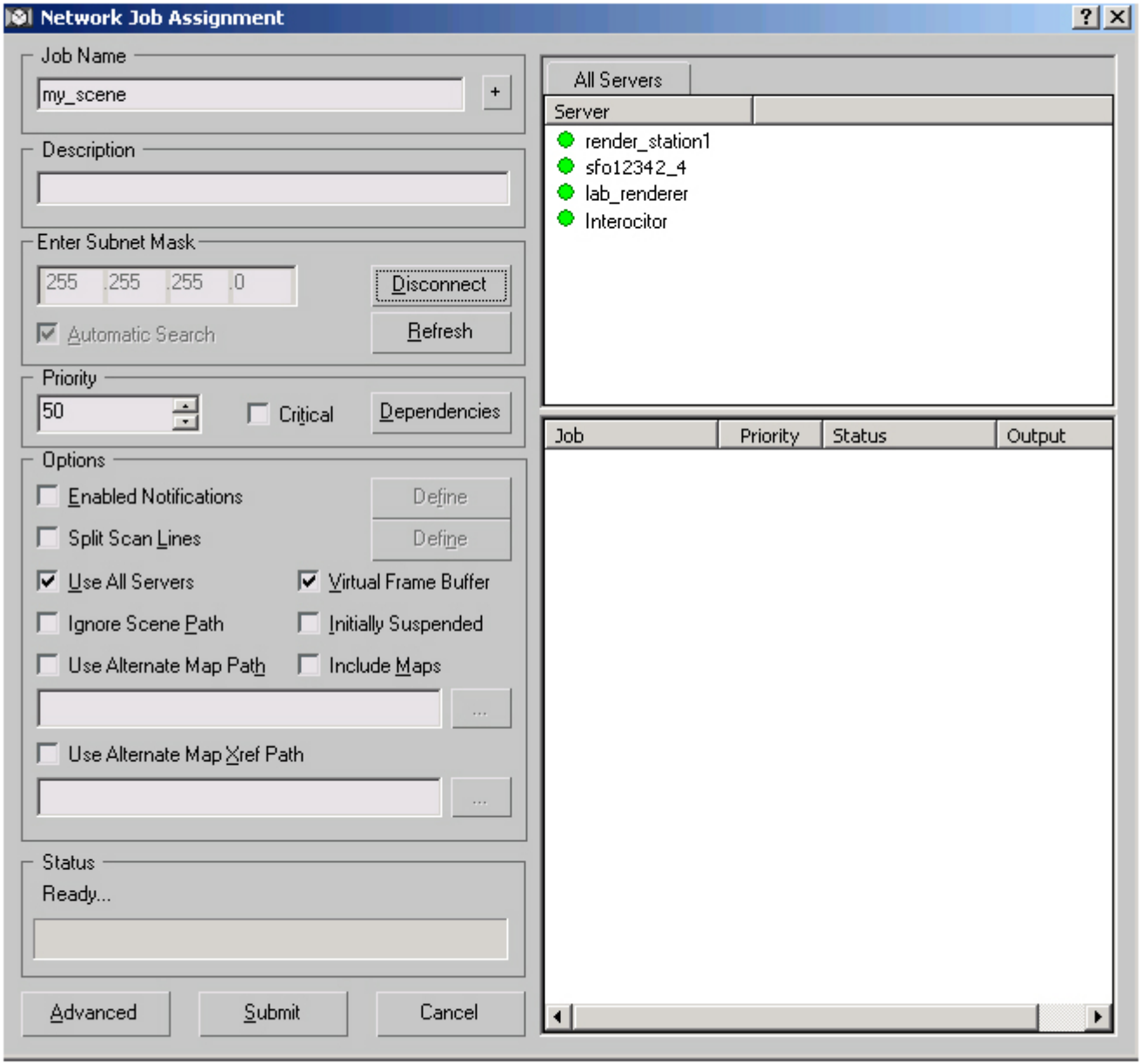
*Valentin Schwind*
Online: www.valisoft.com
Nightfall-Grafiken: nightfall.valisoft.com

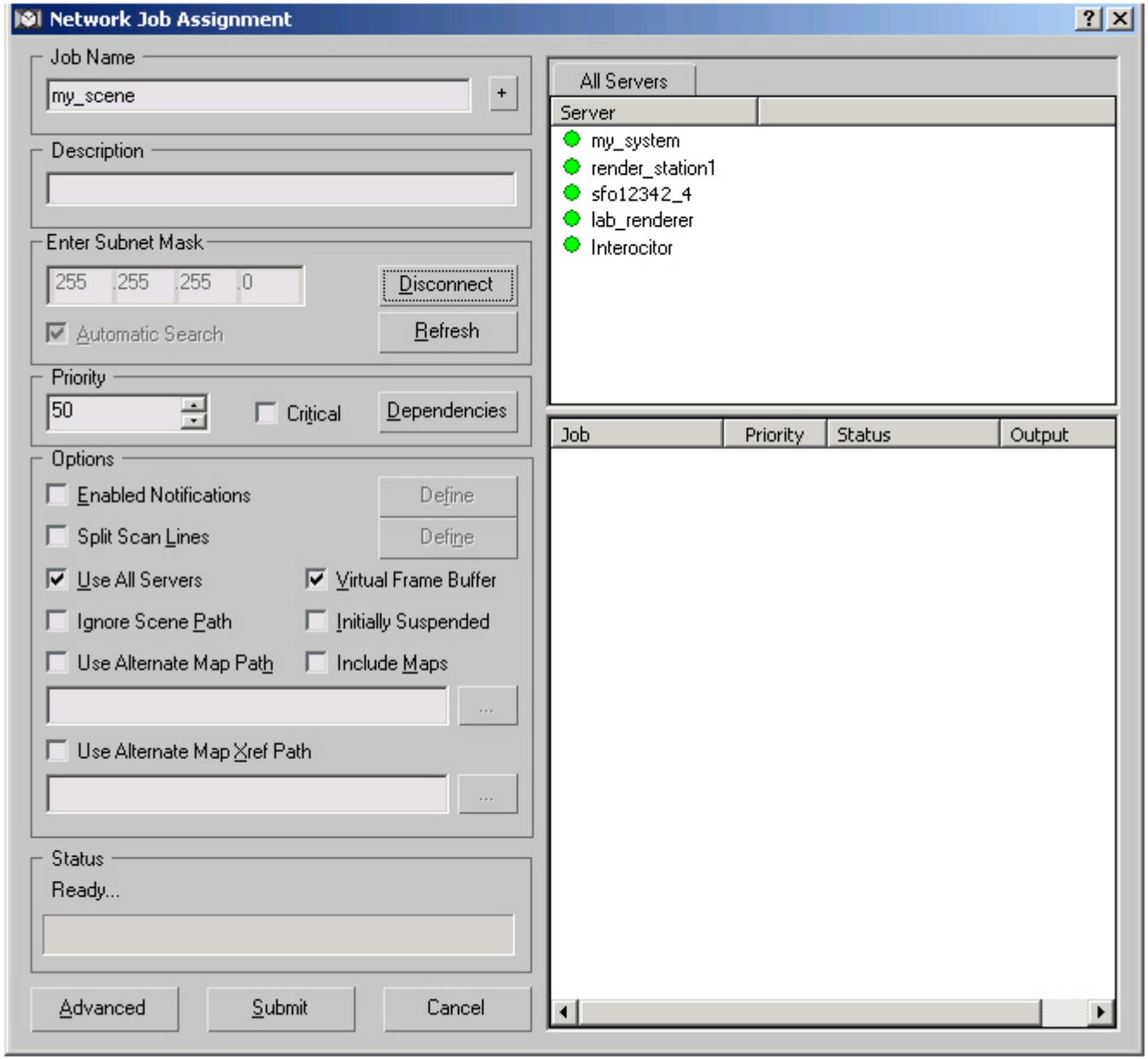
Netzwerkrendering & Distributed
Rendering in verteilten Systemen


TCP/IP or Wire
Render Client
Backburner Manager (Windows workstation)
Render Nodes
Autodesk application running on SGI, Linux, or Windows workstation
TCP/IP
Backburner Monitor
Backburner Manager
Backburner Monitor (via Web browser)
TCP/IP
TCP/IP
TCP/IP
TCP/IP
Backburner Server
Rendering Engine
Backburner Server
Rendering Engine
Backburner Monitor (via Web browser)
Other workstation on network

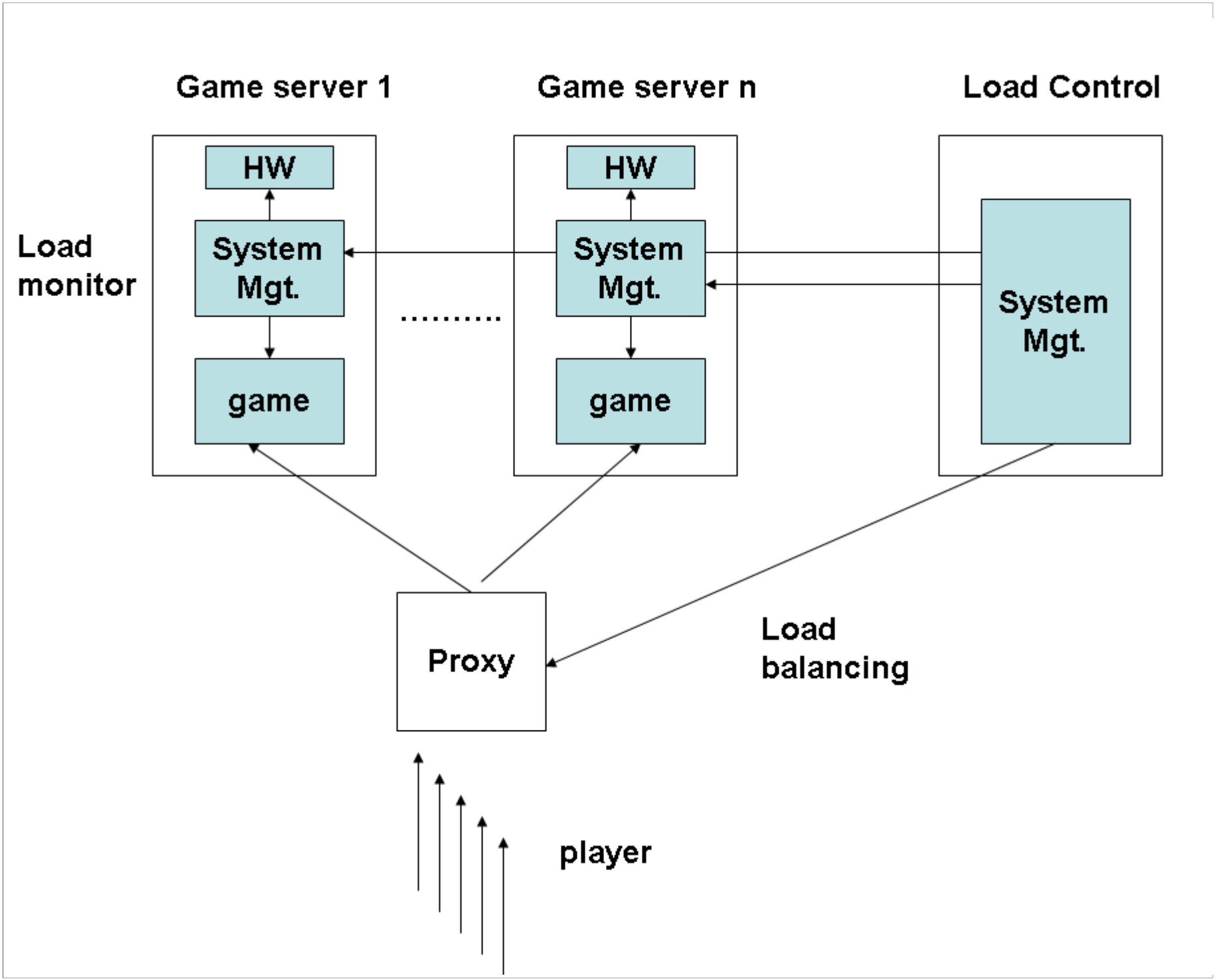

Backburner Server General Properties
TCP/IP
Manager Port:
3234
Server Port:
3233
Enter Subnet Mask
Automatic Search
255 . 255 . 255 . 0
Description
OK
Cancel

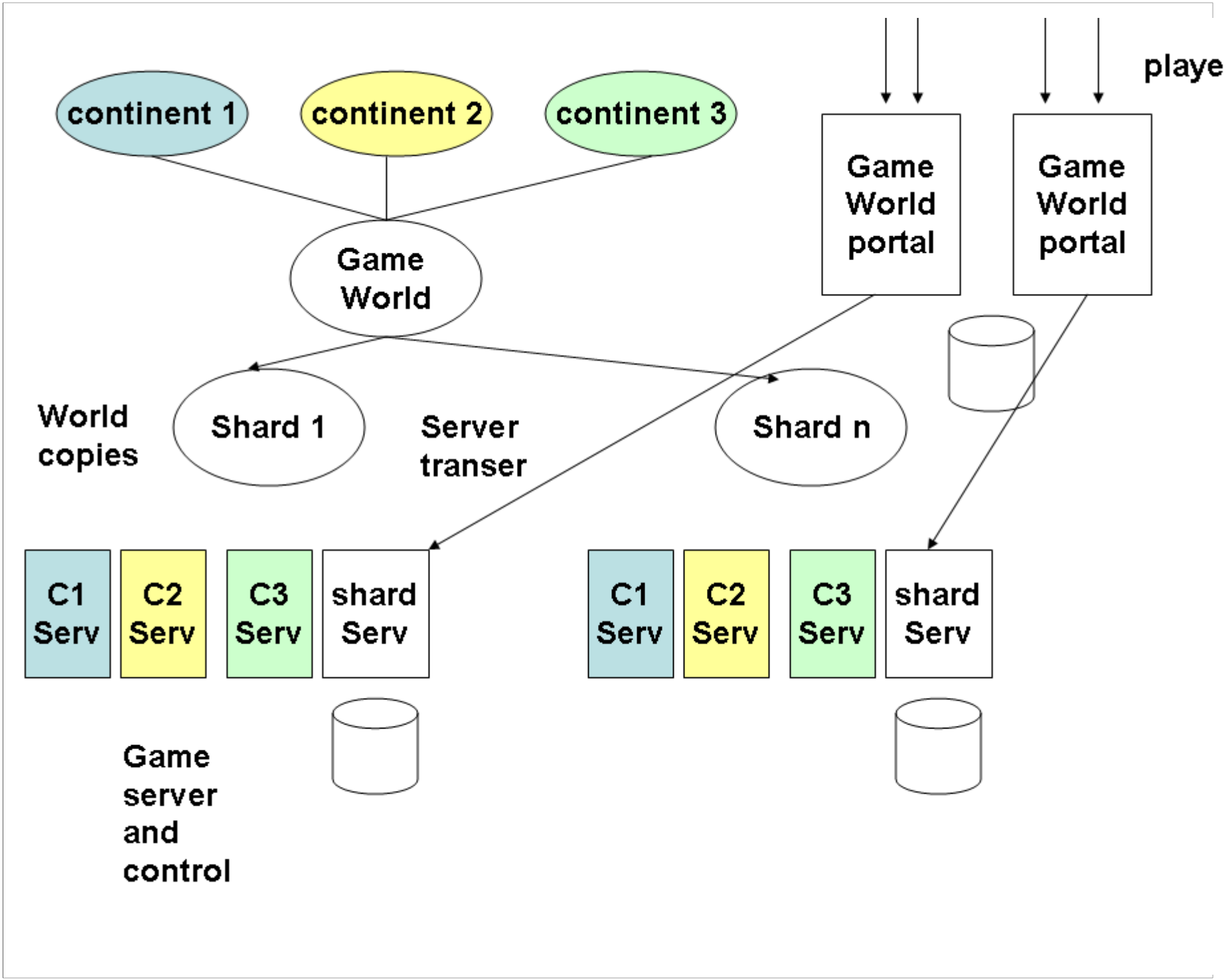

Network Job Assignment
Job Name
my_scene
+
Description
Enter Subnet Mask
255 .255 .255 .0
Disconnect
Automatic Search
Refresh
Priority
50
Critical
Dependencies
Options
Enabled Notifications
Define
Split Scan Lines
Define
Use All Servers
Virtual Frame Buffer
Ignore Scene Path
Initially Suspended
Use Alternate Map Path
Include Maps
...
Use Alternate Map Xref Path
...
Status
Ready...
Advanced
Submit
Cancel
All Servers
Server
my_system
Job
Priority
Status
Output

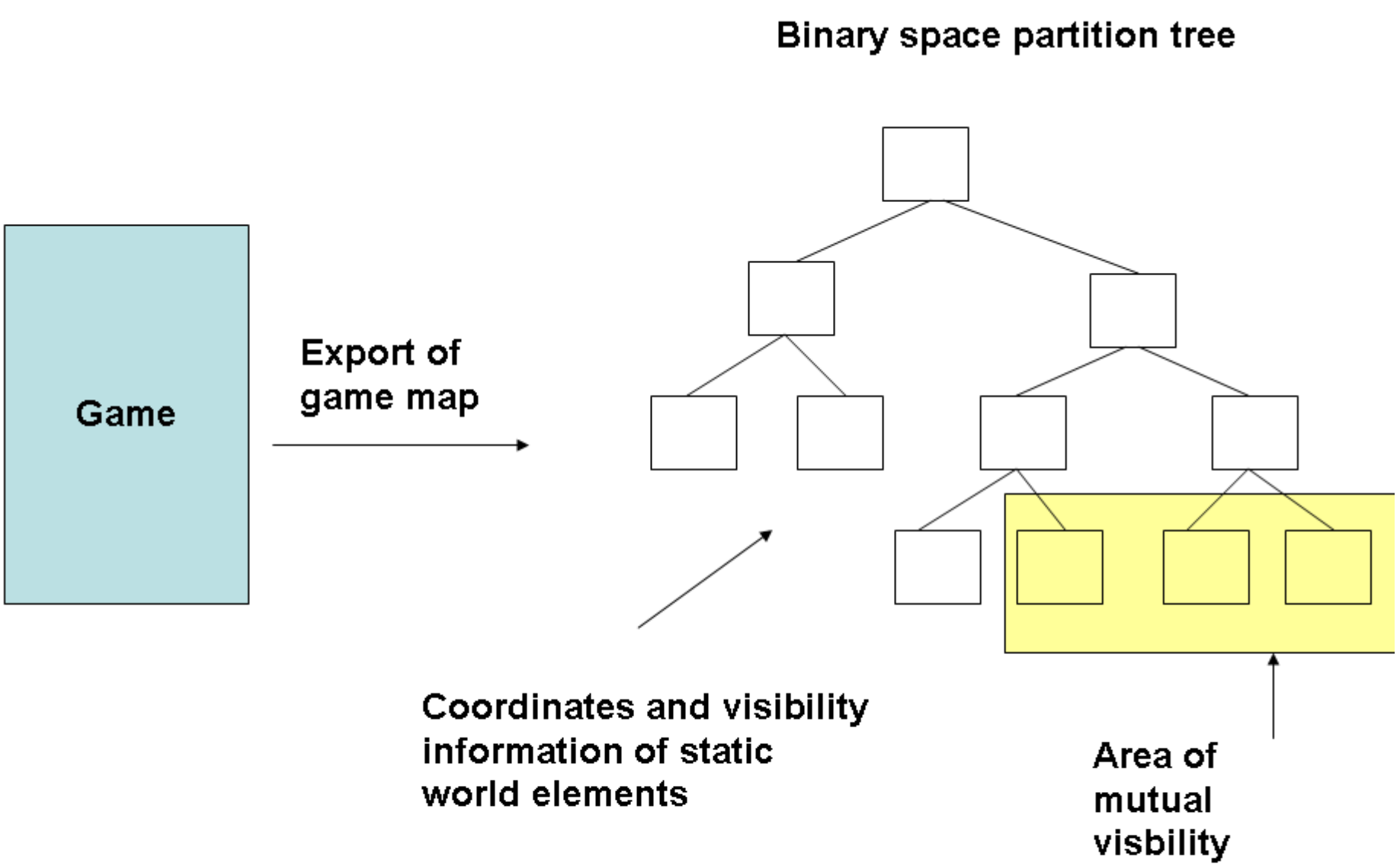

Network Job Assignment
Job Name
my_scene
Description
Enter Subnet Mask
255 255 255 0
Disconnect
Automatic Search
Refresh
Priority
50
Critical
Dependencies
Options
Enabled Notifications
Define
Split Scan Lines
Define
Use All Servers
Virtual Frame Buffer
Ignore Scene Path
Initially Suspended
Use Alternate Map Path
Include Maps
Use Alternate Map Xref Path
Status
Ready...
Advanced
Submit
Cancel
All Servers
Server
render_station1
sfo12342_4
lab_renderer
Interocitor
Job
Priority
Status
Output


Network Job Assignment
Job Name
my_scene
+
Description
Enter Subnet Mask
255 255 255 0
Disconnect
Automatic Search
Refresh
Priority
50
Critical
Dependencies
Options
Enabled Notifications
Define
Split Scan Lines
Define
Use All Servers
Virtual Frame Buffer
Ignore Scene Path
Initially Suspended
Use Alternate Map Path
Include Maps
...
Use Alternate Map Xref Path
...
Status
Ready...
Advanced
Submit
Cancel
All Servers
Server
my_system
render_station1
sfo12342_4
lab_renderer
Interocitor
Job
Priority
Status
Output

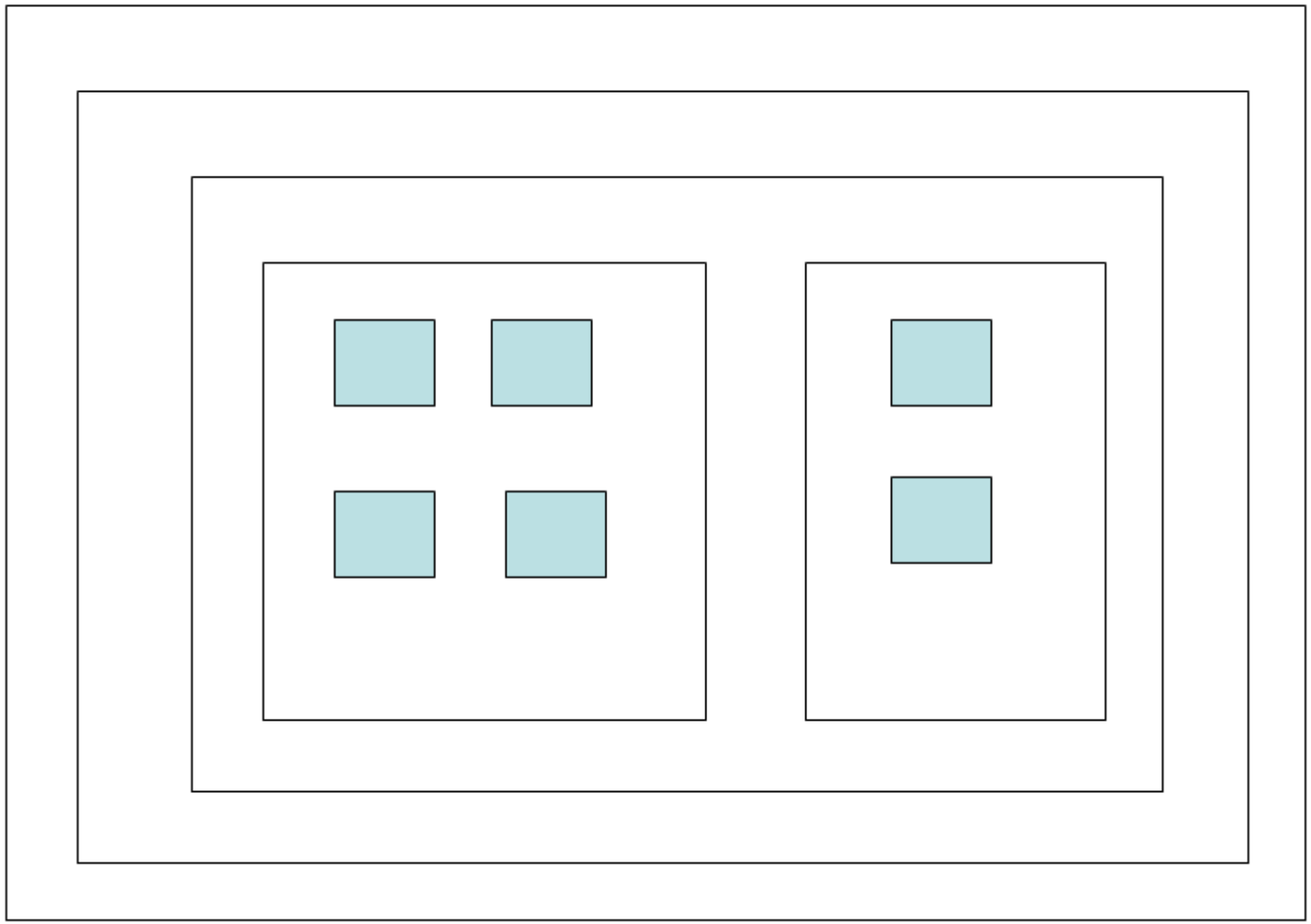
# MMOGs: scalability, partitioning and transparency at real-time speed


Game server 1
Game server n
Load Control
HW
HW
Load
monitor
System
Mgt.
System
Mgt.
System
Mgt.
.....
game
game
Proxy
Load
balancing
player


player
continent 1
continent 2
continent 3
Game World portal
Game World portal
Game World
World copies
Shard 1
Server transer
Shard n
C1 Serv
C2 Serv
C3 Serv
shard Serv
C1 Serv
C2 Serv
C3 Serv
shard Serv
Game server and control

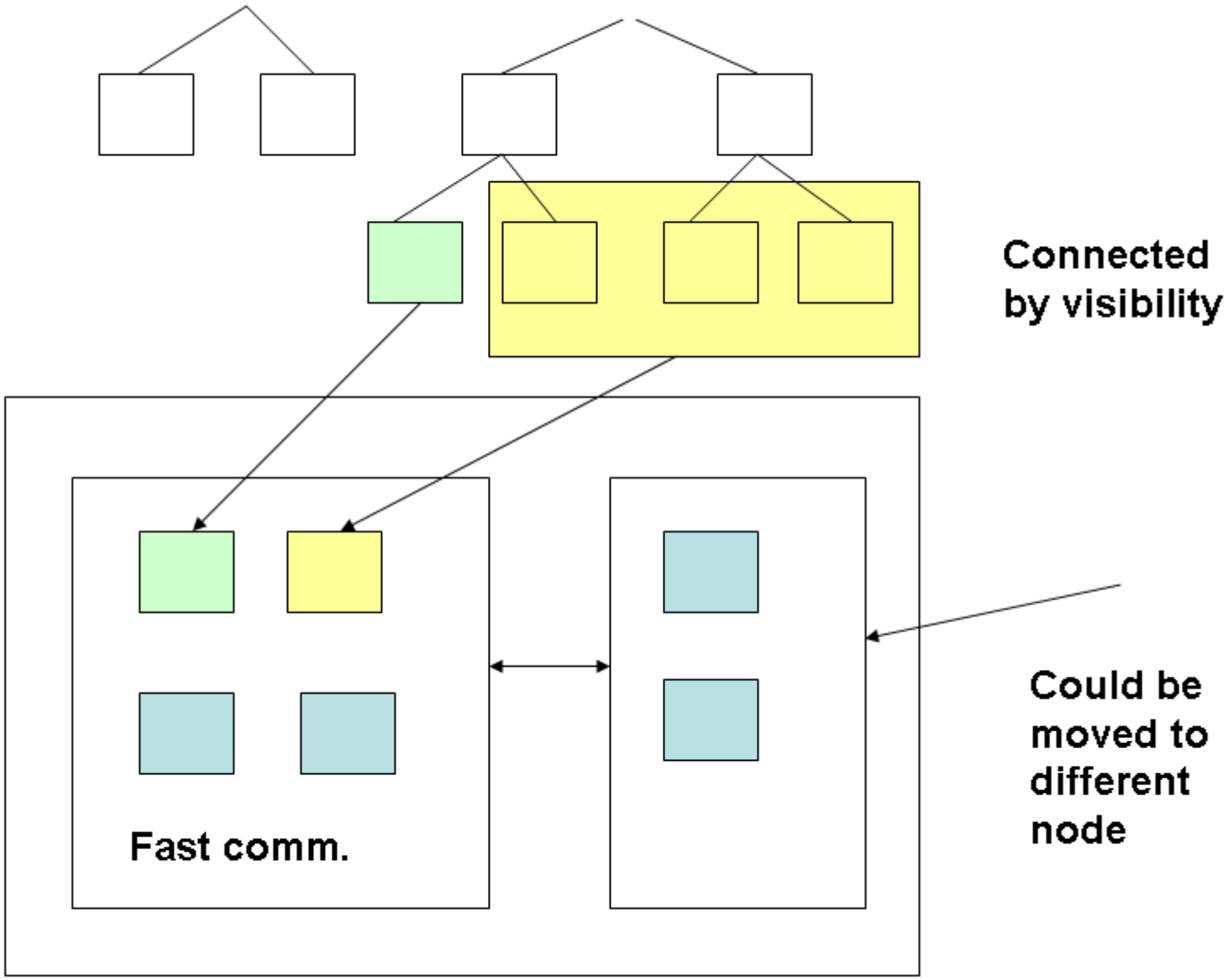

Binary space partition tree
Game
Export of
game map
Coordinates and visibility
information of static
world elements
Area of
mutual
visbility

## Grid node computing and administrative elements

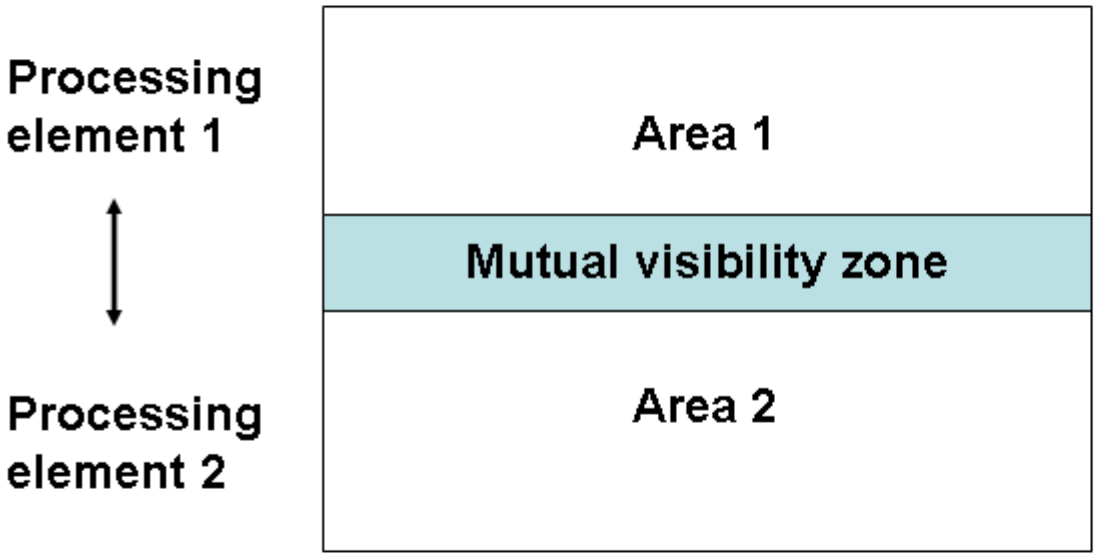


Borders: in-process, inter-process, inter-virtual-node, inter node

## Static bsp to compute grid mapping

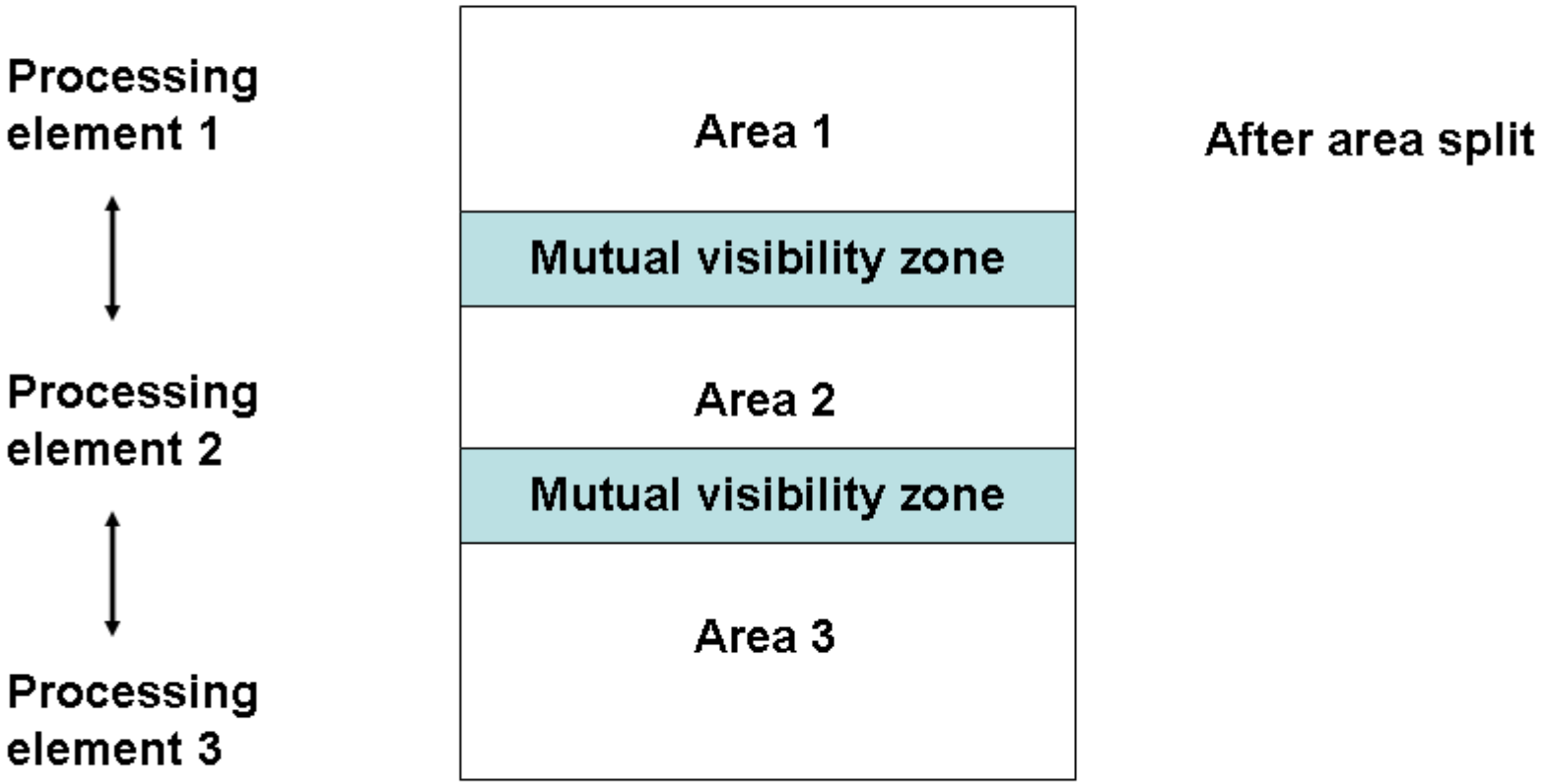


## Dynamic reconfiguration of partitioning based on local inconsistency and static visibility regions

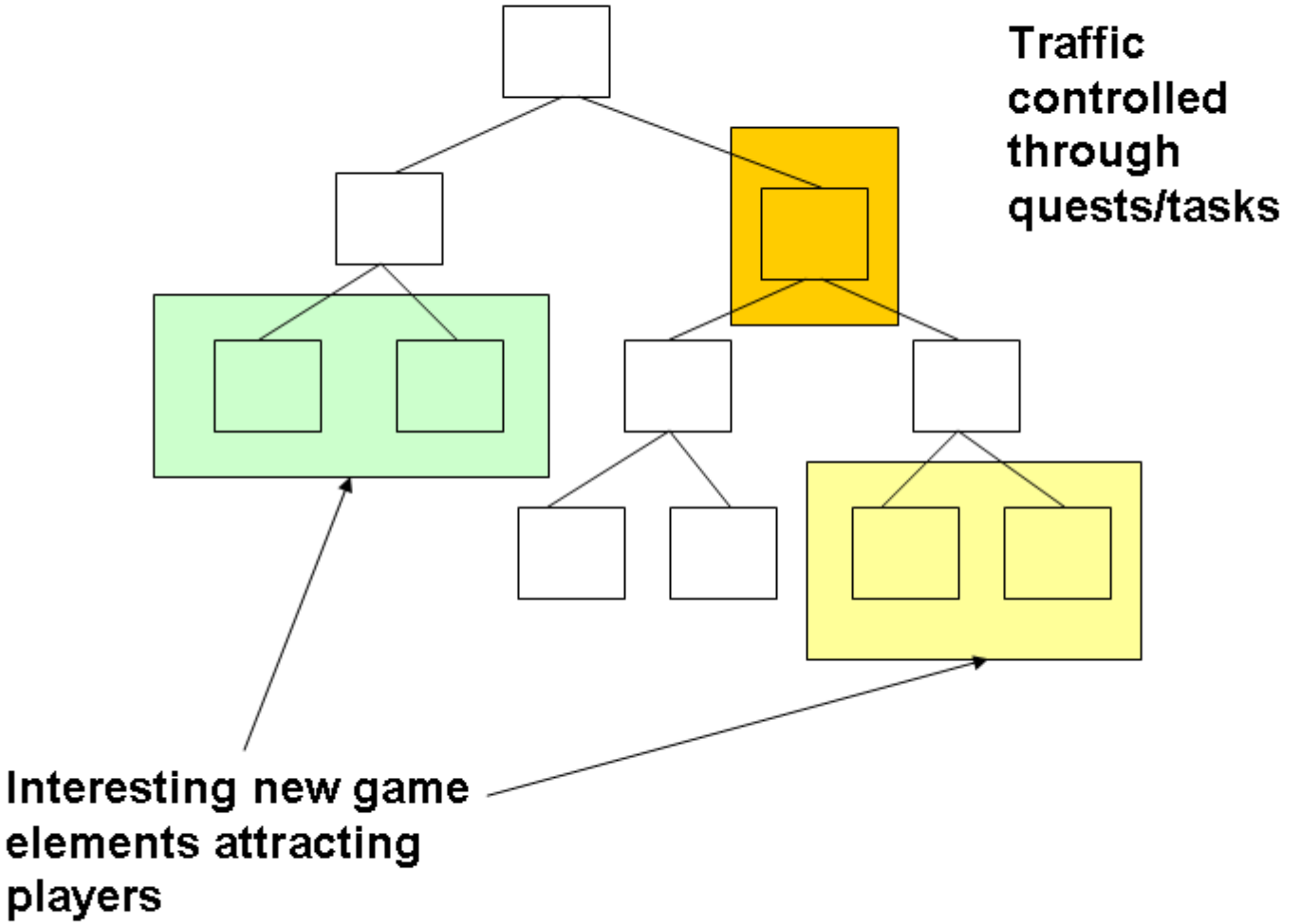



Locally consistent through event propagation

## Dynamic reconfiguration of partitioning based on local inconsistency and static visibility regions



## Content level: game balancing to avoid flash crowds and hotspots



## Croquet peer-to-peer replication architecture

**New secondlife architecture (wikipedia)**

**Reliable messages: only when re-transmit makes sense. If re-transmitted data are outdated, don‘t ask for them. Design a protocol that lets clients re-synchronize**

**Proxying allows hiding of infrastructure and load balancing. Let proxies take over the distribution of world segments to clients. The proxies then have to know wich client sees which part of the world (protocol design issue)**

**Caching only when local space is less valuable than saved latency. This is usually true on coarse grained distributed systems but may be wrong in fine-grained, tightly connected hardware cases (cell chip e.g.)**

**Partitioning creates compute power at the price of increased communication overhead and latency. It can work on several layers, up to the top level content layer. Decide carefully where transparent partitioning is really needed and what the costs are.**

**Lose-coupling of servers allows the transparent addition of new compute power without modifiying existing servers.**

**Watch for logical connectedness (logical dependencies) in your world. A high degree of logical connectedness requires a high communication load in case of partitioning.**

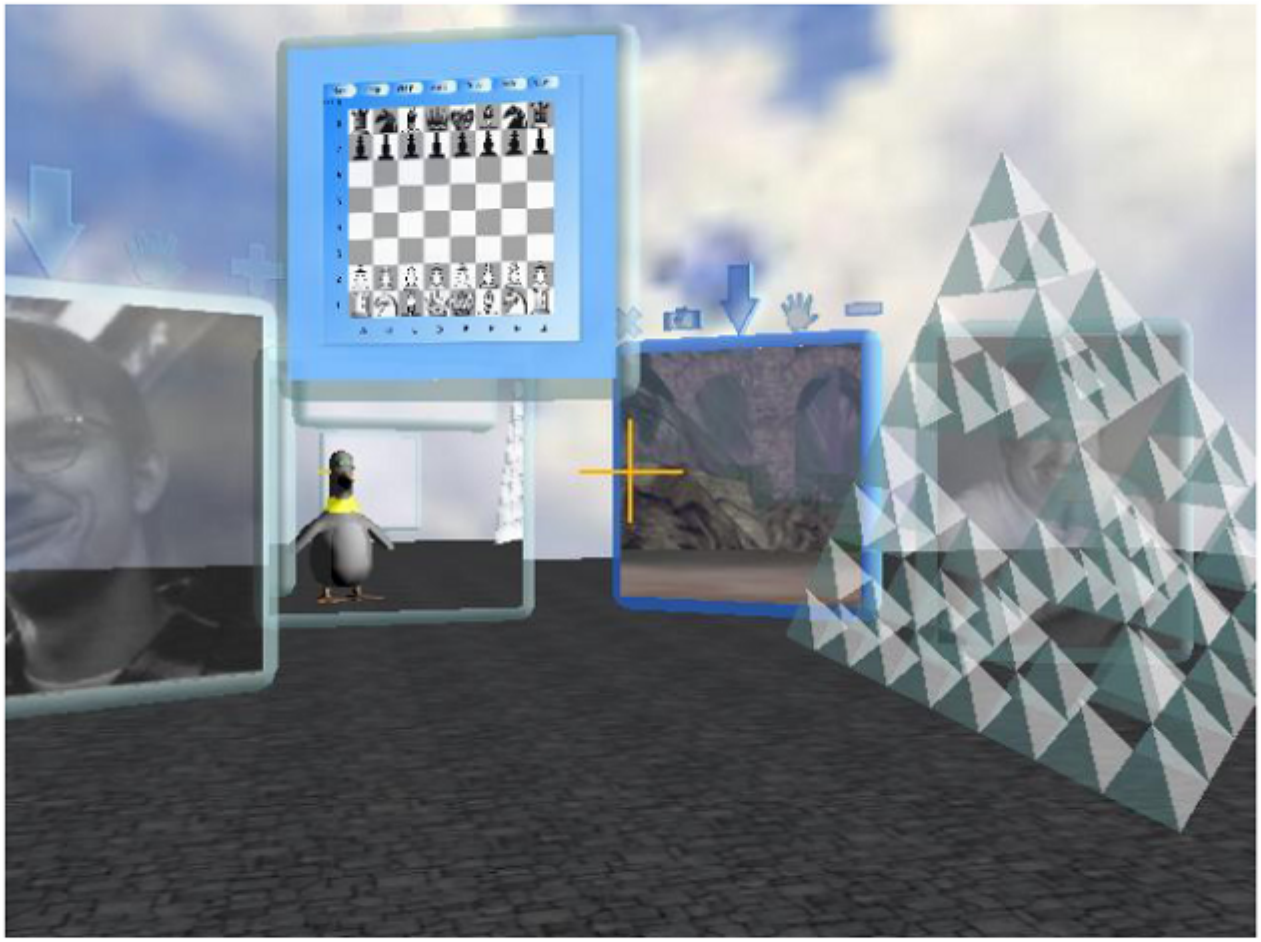
# Croquet: immersive, collaborative virtual worlds

# What is Croquet?

**„Imagine you are a graduate student in astronomy asked to demonstrate your knowledge of Kepler's Laws. You launch a software application on your computer and enter a three-dimensional online world. Inside this environment, you use the drop-down menu to quickly design and deploy a dynamic simulation of the solar system. As your simulation runs, you see your professor enter the lab and move in for a closer look. Your professor downloads a file from his own hard drive into the virtual laboratory and it appears inside a display window he just created with a click of the mouse. Remarkably, you and your professor are now able to see one another make additions and changes to the same document, all the while keeping up a steady banter with the help of network-enabled telephony built into the software system. Your professor is impressed by your work and so he invites his entire introductory astronomy class to a viewing and discussion of your simulation. From across campus, hundreds of students gather inside the virtual lab. The instructor's video image (captured by the web camera on his laptop) is visible to the groups of students he guides through the demonstration. Classmates wander among the planets, adjusting the timing and motion of the celestial machinery, talking with one another over network-enabled telephony, gaining an unprecedented appreciation for Kepler's laws in action. „**

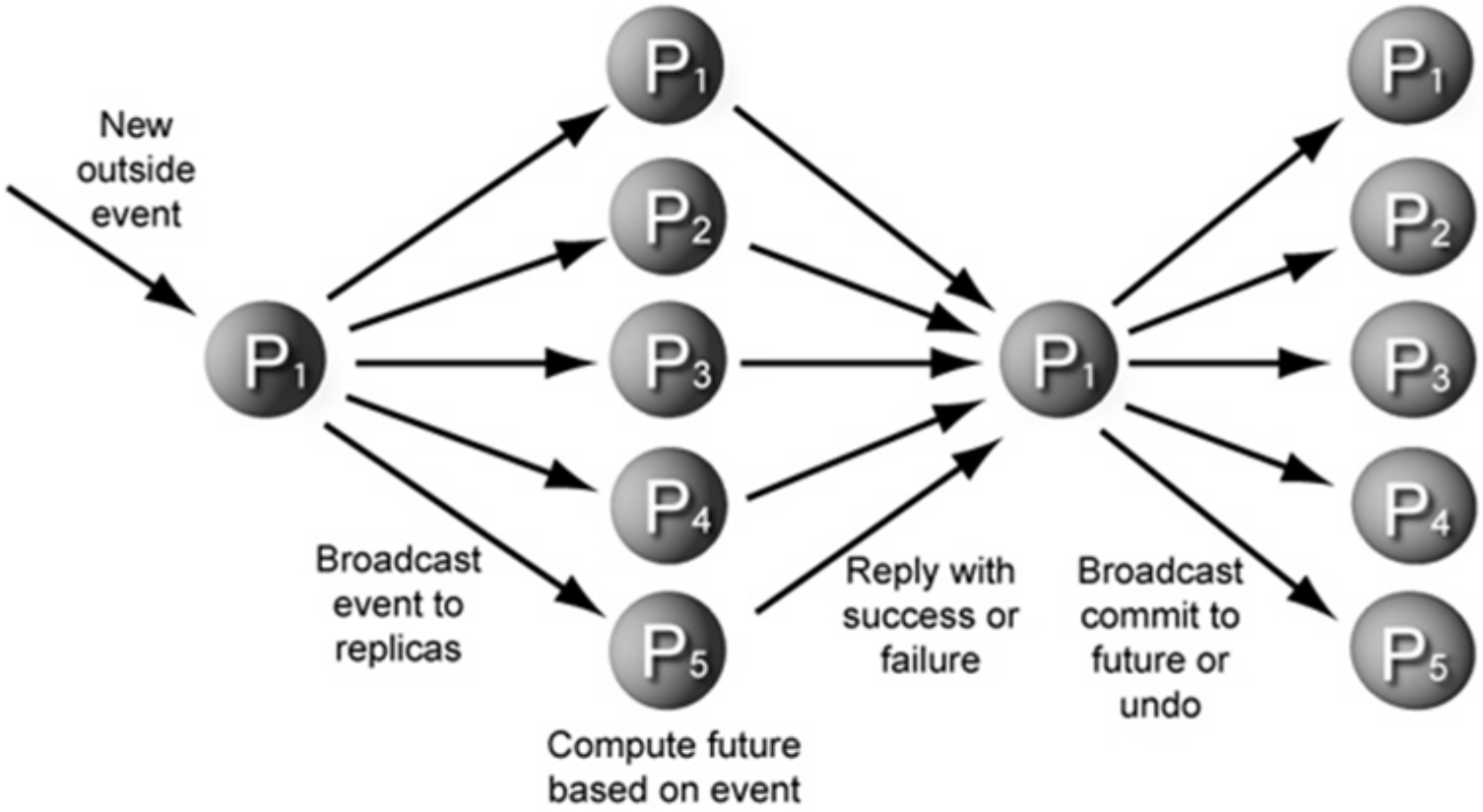
M.Lombardi, Standing on a Plateau..

# Replicated Collaborative Objects and Activities



User portals are shared (replicated) across machines. (Alan Kay et.al, Croquet – A collaboration system)

# Teatime



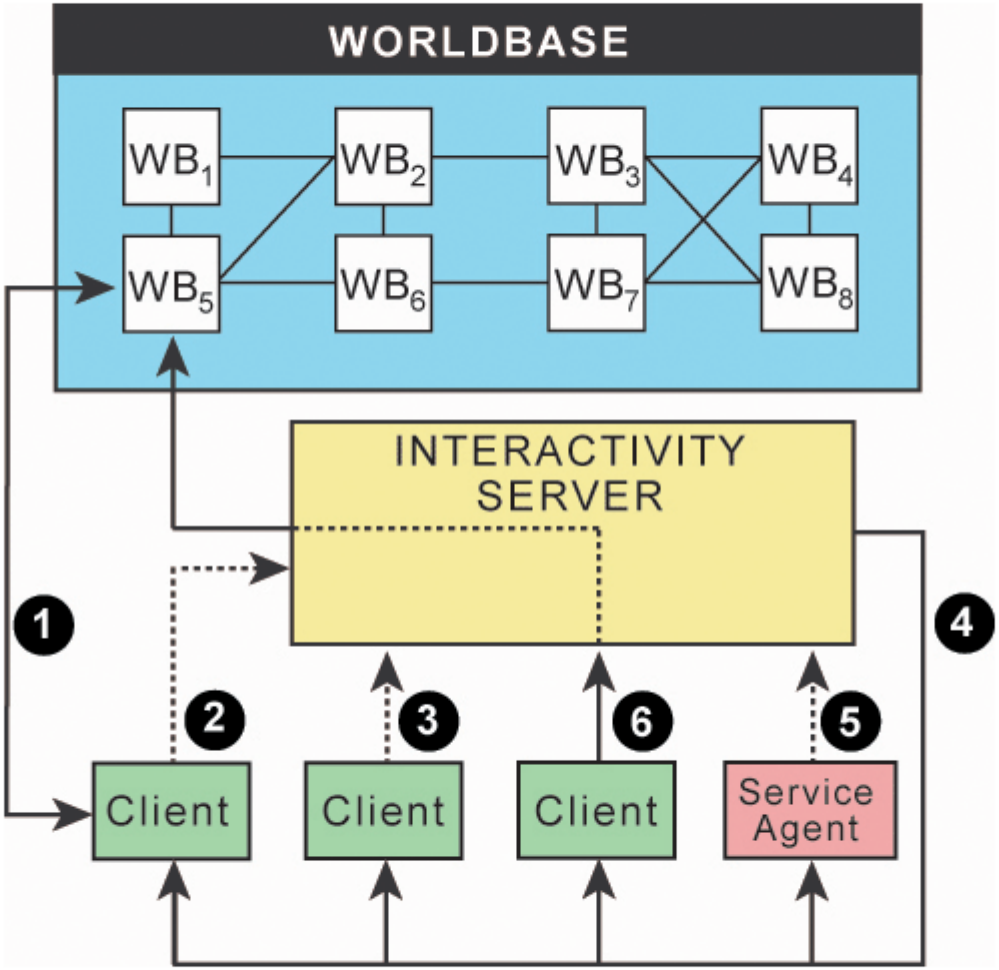

Teatime is the distributed objects framework within Croquet. It provides replication of behavior, synchronization etc.

# Key Elements and Assumptions

- *A coordinated universal timebase embedded in communications protocol*
- *Replicated, versioned objects - unifying replicated computation and distribution of results*
- *Replication strategies - that separate the mechanisms of replication from the behavioral semantics of objects*
- *Deadline-based scheduling extended with failure and nesting*
- *A coordinated distributed two-phase commit that is used to control the progression of computations at multiple sites, to provide resilience, deterministic results, and adaptation to available resources. Uses distributed sets*
- *Low latency network*
- *local rendering and massive computing power*

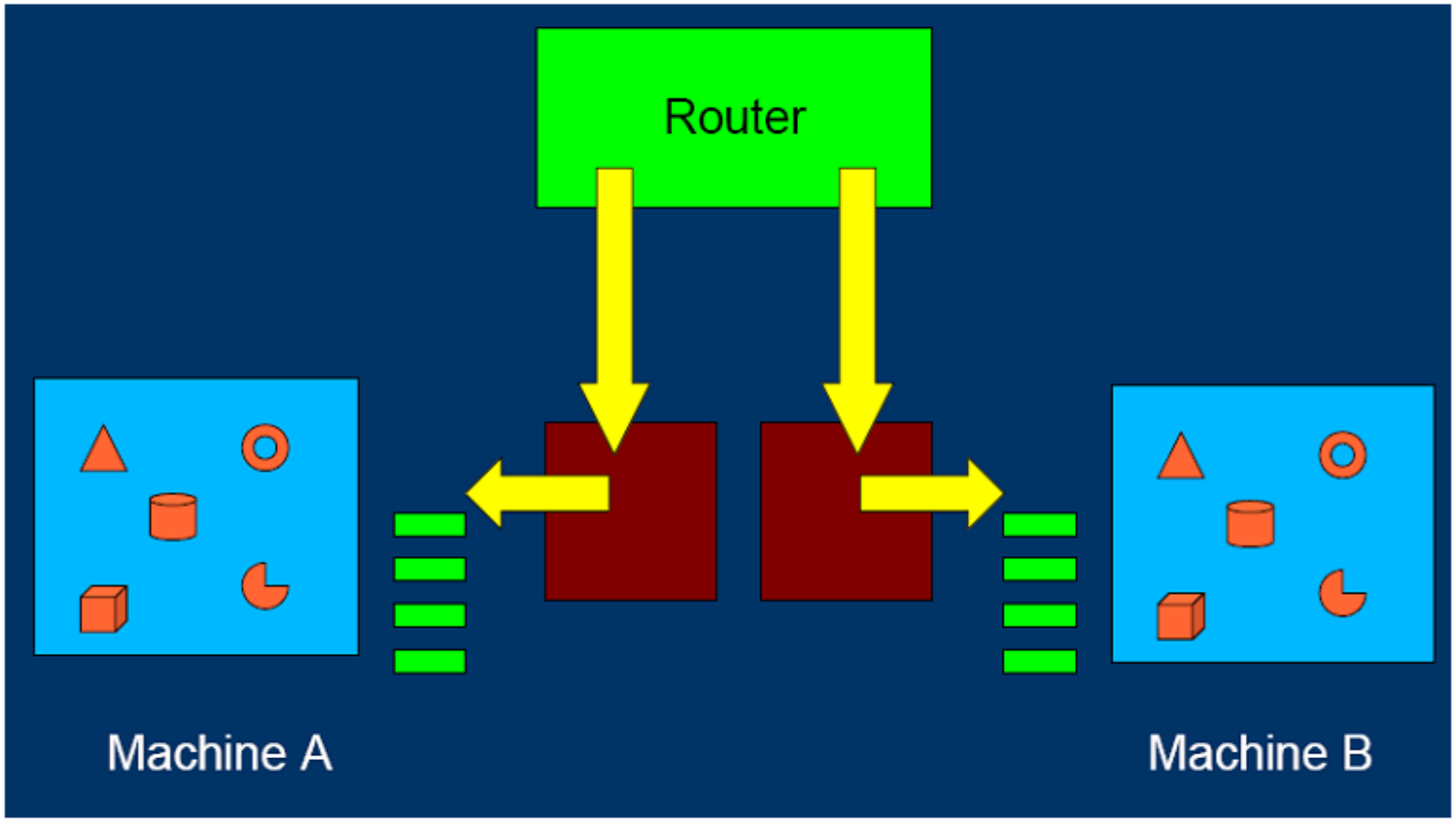
From: Croquet Architecture and David P.Reeds presentation (see ressources)

# World Structure




Event distribution needs to be carefully minimized.

# Routers, Controllers, Time based Replication




Islands are replicated. Controllers interface Islands with Routers.
(From Croquet Programmers Guide, see Ressources)

# WOW-Architecture


Auth
CS
CS
DC