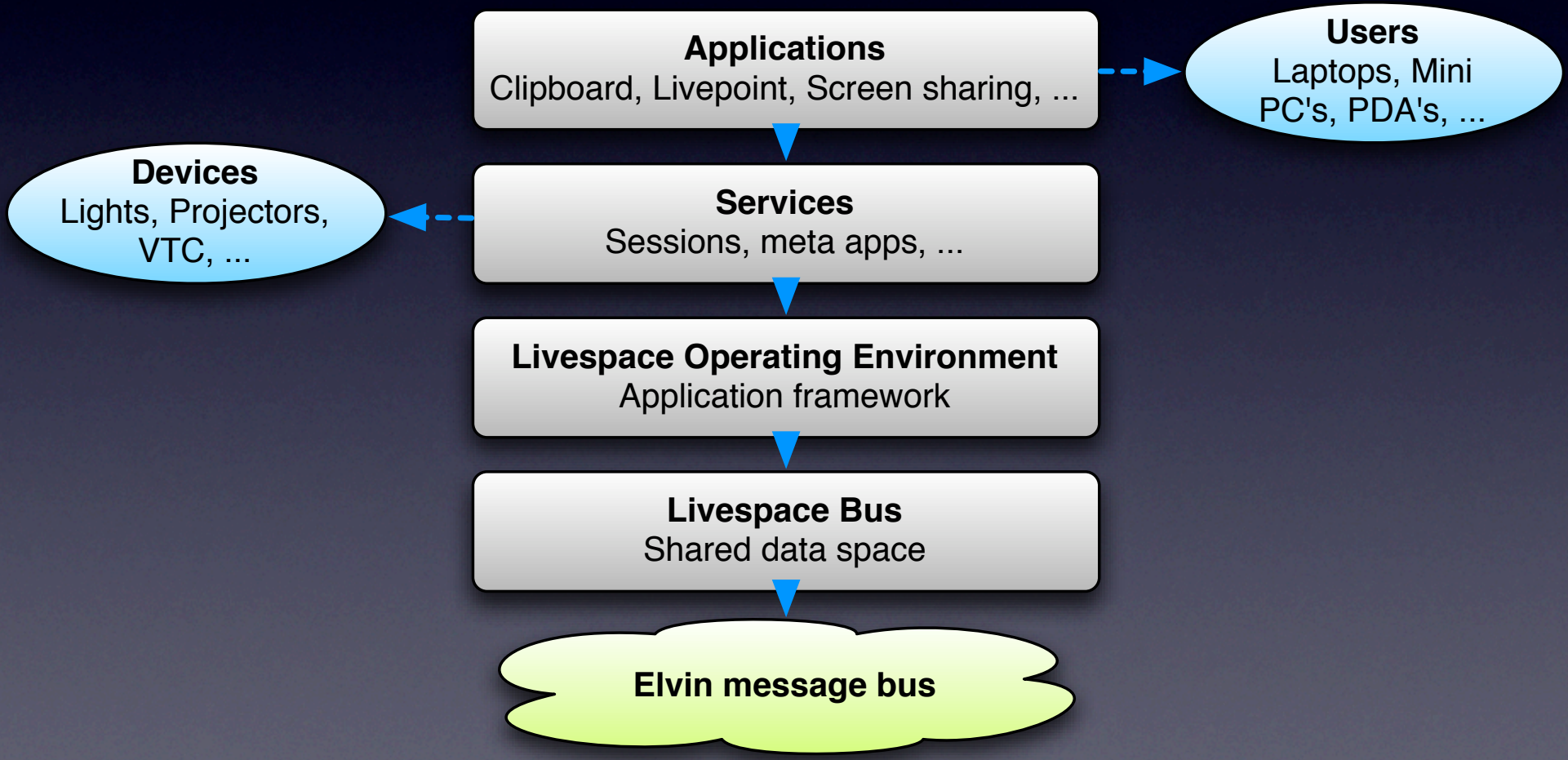
The Livespace Bus

Part I

The Livespace Bus

The Livespace bus is the core communication system underlying a Livespace

Where The Bus Fits



History

- Where the concepts underlying the Livespace architecture came from

History

- Livespaces started life as AUSPLANS
- Originally built with iROS, ODSI, Breeze and other bits and pieces
- This had lots of problems for any number of reasons
- But the root cause of most of them were due to the fact that...

Distributed Systems

Are

Hard

Eight Fallacies Of Distributed Computing

1. The network is reliable
2. Latency is zero
3. Bandwidth is infinite
4. The network is secure
5. Topology doesn't change
6. There is one administrator
7. Transport cost is zero
8. The network is homogeneous

– Peter Deutsch, Bill Joy, Tom Lyon, James Gosling

Everyone Thinks RPC's Are A Solution

- Remote Procedure Calls (RPC's) seem like a a natural solution
- Every developer understands the model
- Just run a call over the network

RPC's Fall Afoul Of The Eight Fallacies

- Especially affected by
  - Unreliable network
  - Non-zero latency
  - Security
  - Topology changes

Other Ways RPC's Fail

- Synchronous – clients tied to the network
- Large & brittle API – evolution is hard
- One way – hard for a "client" to connect to "server"
- A leaky abstraction – real procedure calls do not randomly fail when someone kicks a cable out of a socket

Why Tuple Spaces Don't (Quite) Work

- Tuple spaces are a good step up
- And they *almost* solve our problems
- Tuples are network events, often used to
  - Find/announce a resource's existence
  - Telling you about a change to a resource
  - Requesting a change (to a resource)

Resources: Find, Update, Monitor

- The resource *find*, *update* and *monitor* pattern is universal in tuple spaces
- But they miss one thing:

  They don't actually define what a resource is

- Clearly the next step is...

Make Everything A Resource

Everything in a Livespace should a

findable,

modifiable,

observable

information resource

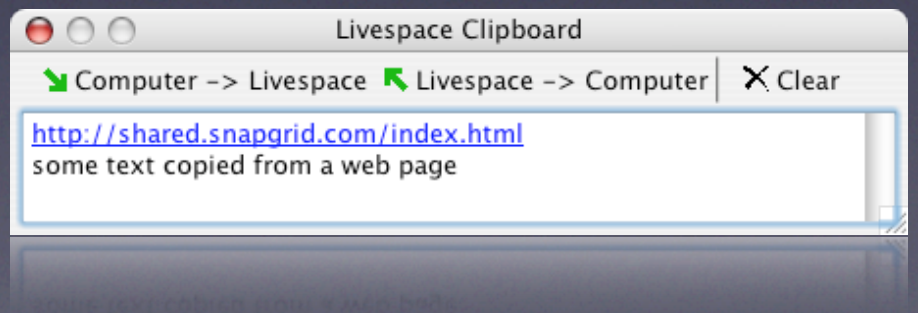
... simple!

It Actually Is Almost That Simple

- All operations in a Livespace are modelled as state changes to distributed resources
- We call these resources *entities*

Simplest Example – The Clipboard

- To set clipboard text
  - Find the `clipboard` entity
  - Set its `text` property



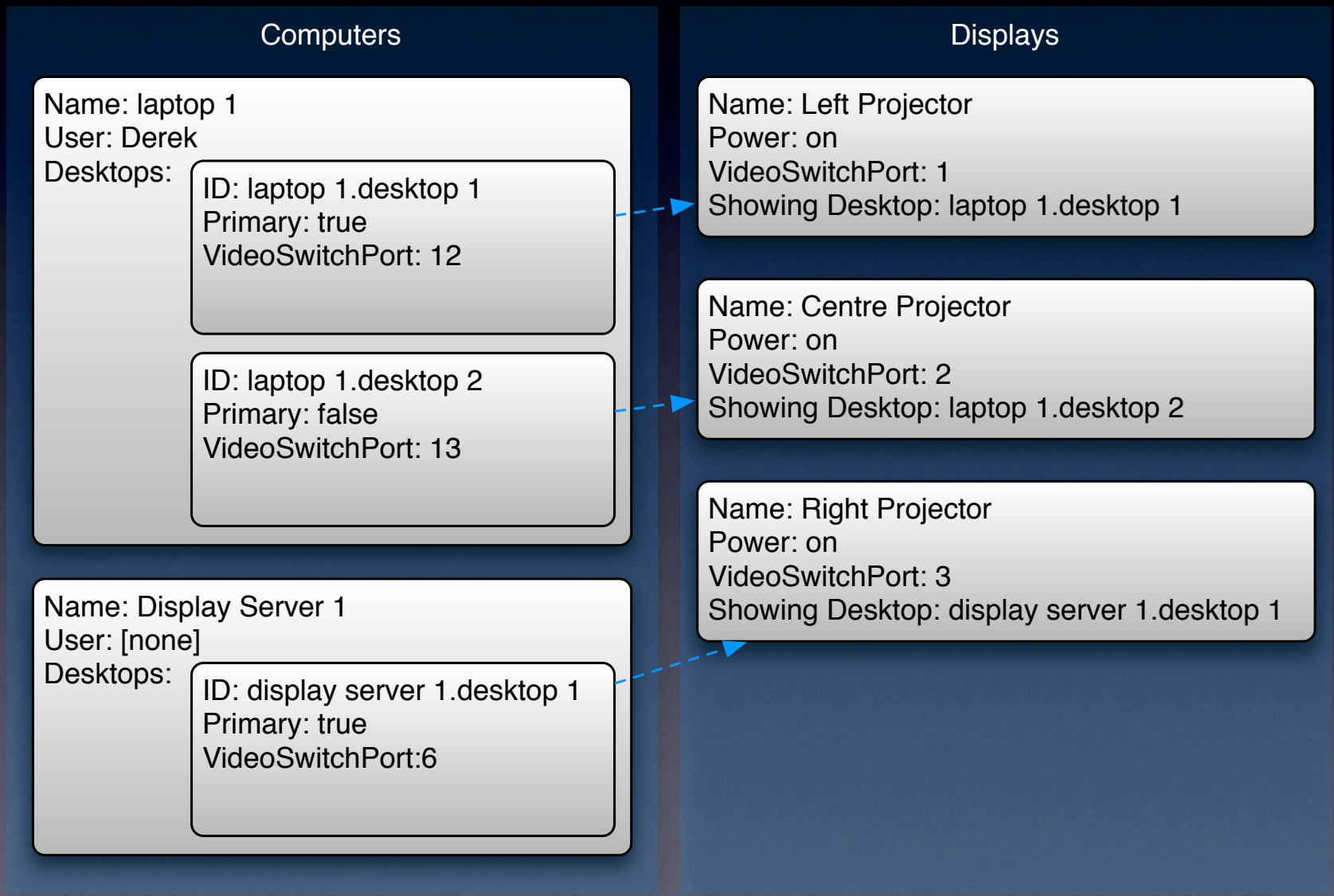
Another Example – The Command Line Service

- Allows commands to be executed on a host
- An example of an active service
- Set the `computer` entity's `commandLine.command` property to execute a command on that computer
- Result is echoed in other properties

Complex Example – Forwarding A Screen

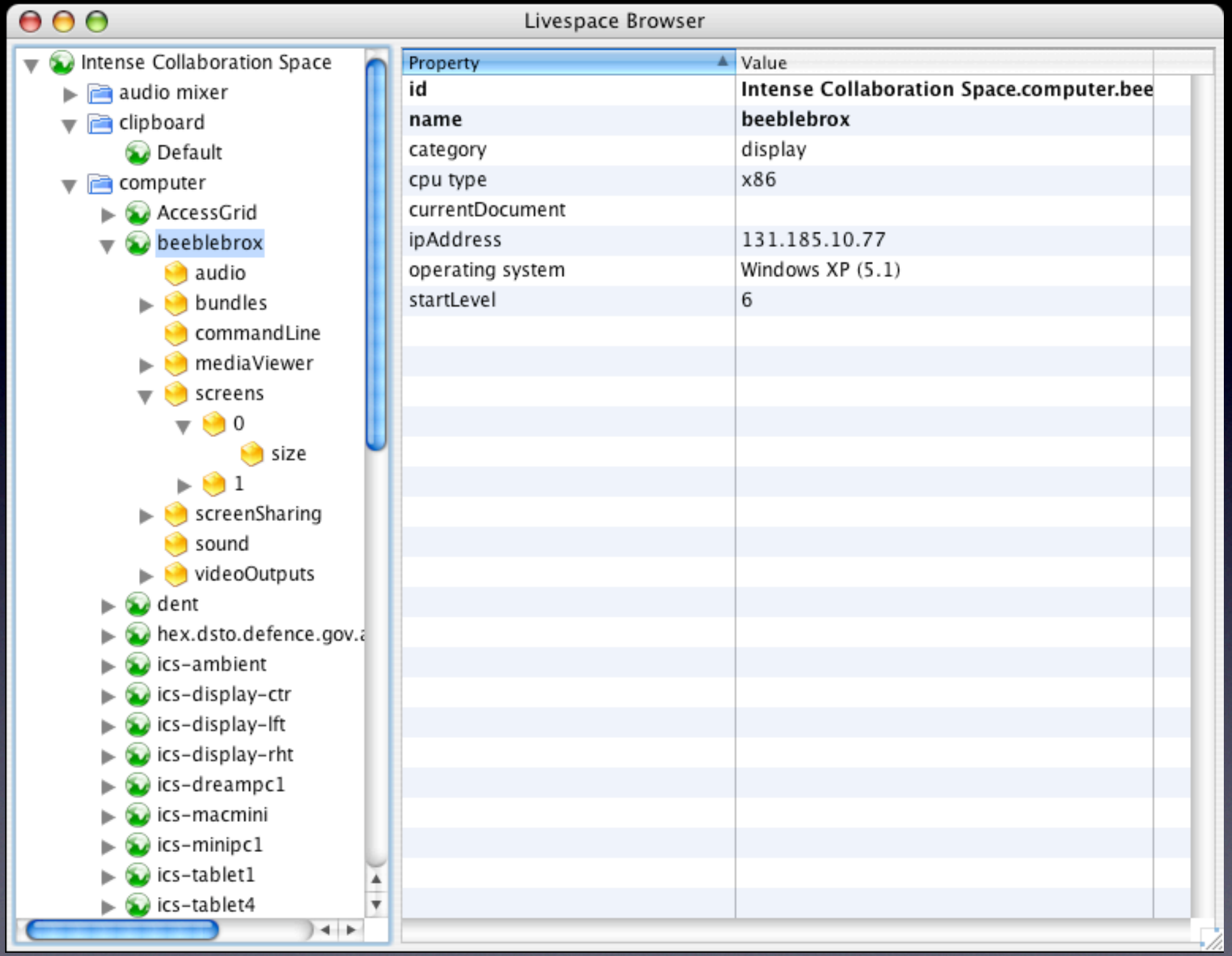
- Find the computer's `screen` entity
- Find the `display` entity
- Set the display's `videoSource` property

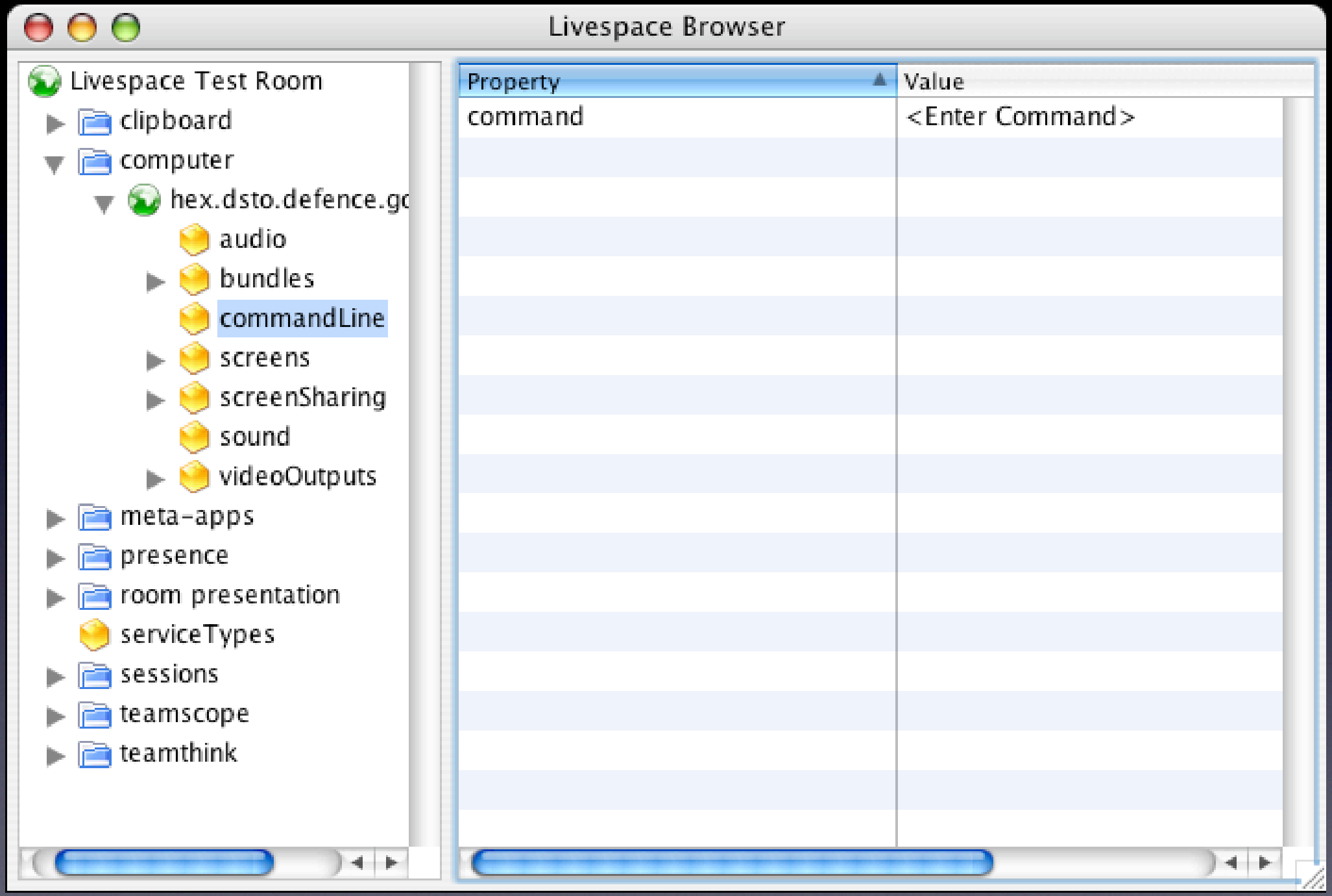
Example – Forwarding A Screen

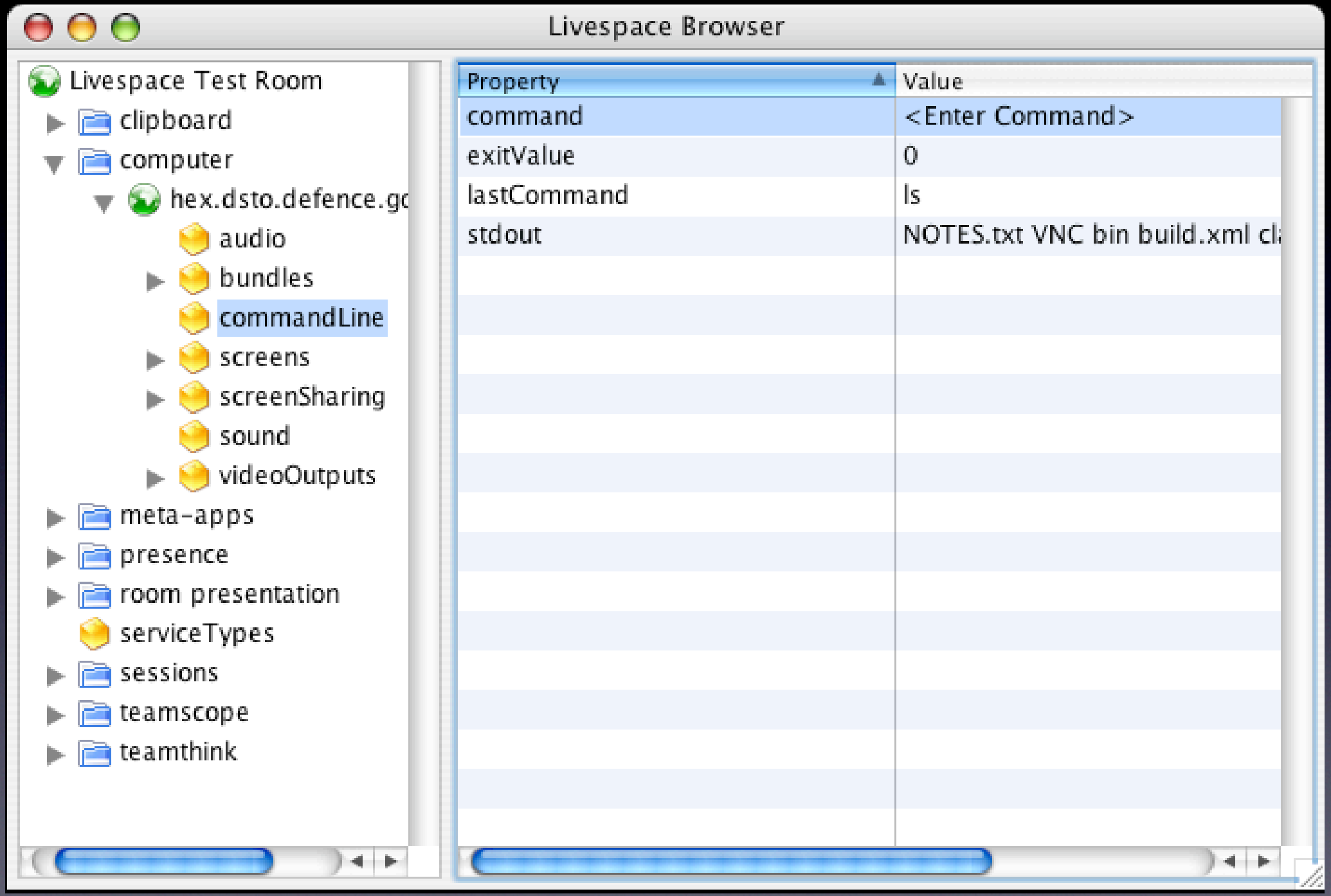


The Browser

- The *everything-is-an-entity* approach makes it easier to see what's going on
- The Livespace Browser is a general exploration & debugging tool for entities


Livespace Browser
Intense Collaboration Space
audio mixer
clipboard
Default
computer
AccessGrid
beeblebrox
audio
bundles
commandLine
mediaViewer
screens
0
size
1
screenSharing
sound
videoOutputs
dent
hex.dsto.defence.gov.a
ics-ambient
ics-display-ctr
ics-display-lft
ics-display-rht
ics-dreampc1
ics-macmini
ics-minipc1
ics-tablet1
ics-tablet4
Property
Value
id
Intense Collaboration Space.computer.bee
name
beeblebrox
category
display
cpu type
x86
currentDocument
ipAddress
131.185.10.77
operating system
Windows XP (5.1)
startLevel
6


Livespace Browser
Livespace Test Room
clipboard
computer
hex.dsto.defence.gc
audio
bundles
commandLine
screens
screenSharing
sound
videoOutputs
meta-apps
presence
room presentation
serviceTypes
sessions
teamscope
teamthink
Property
Value
command
<Enter Command>


Livespace Browser
Livespace Test Room
clipboard
computer
hex.dsto.defence.gc
audio
bundles
commandLine
screens
screenSharing
sound
videoOutputs
meta-apps
presence
room presentation
serviceTypes
sessions
teamscope
teamthink
Property
Value
command
<Enter Command>
exitValue
0
lastCommand
ls
stdout
NOTES.txt VNC bin build.xml cl

Contrast To RPC

- An RPC screen forwarding approach would need API's for:
  - Setting a screen forward
  - Deleting a screen forward
  - Querying what screens are forwarded
  - Querying what screens can be forwarded
  - Adding/removing a screen forwarding listener
  - ...

End Of Part 1