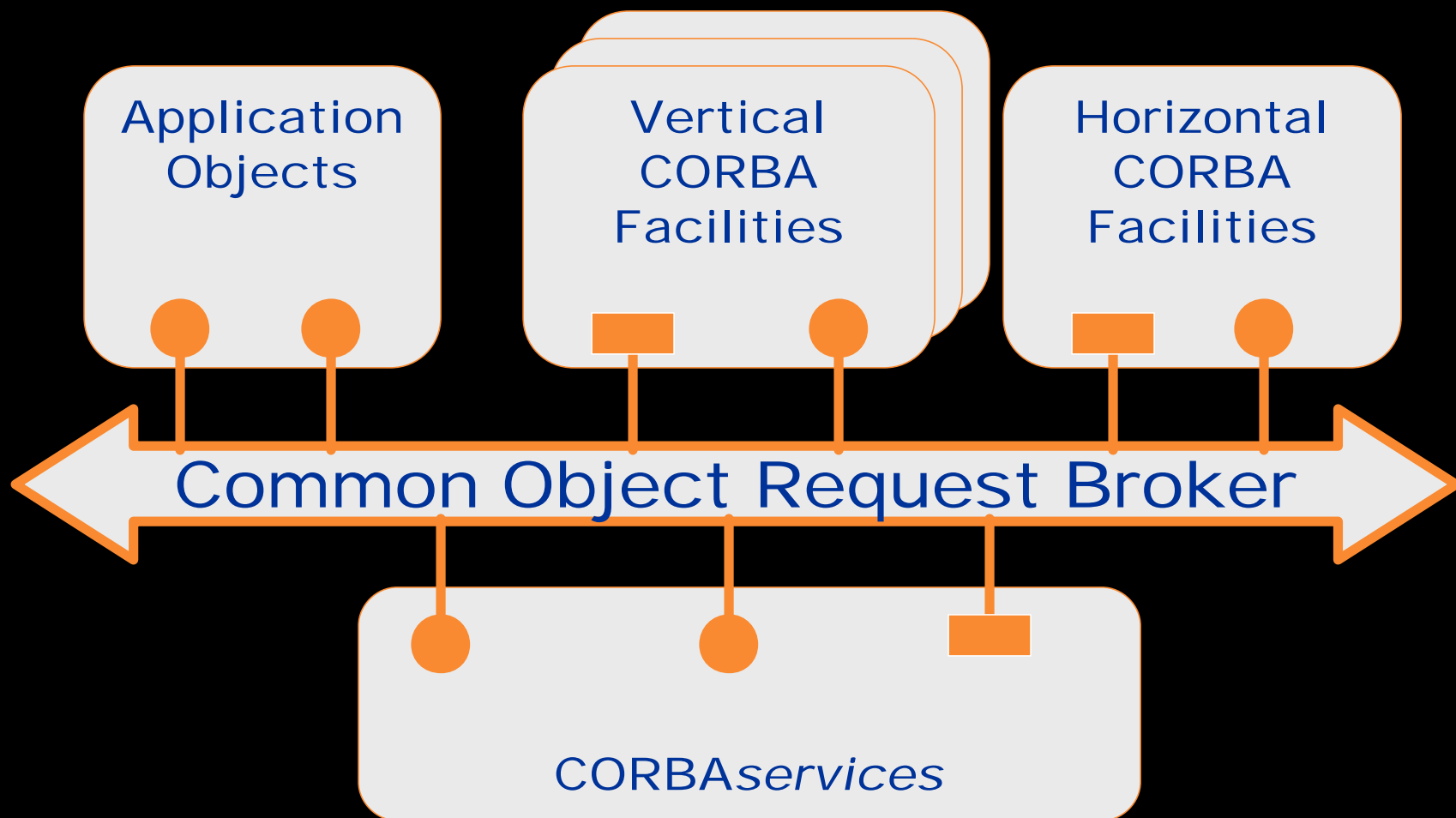




Fine Grained CORBA Services to Build Scalable DRT&E Architectures

Victor Giddings
Objective Interface Systems, Inc.
victor.giddings@ois.com

Object Model Architecture



- ◆ “The CORBA Services component standardizes the life cycle management of objects.”
- ◆ Defined in a set of specifications:
 - Additional Structuring Mechanisms for the OTS
 - Collection Service
 - Concurrency Service
 - Enhanced View of Time
 - Event Service
 - Externalization Service
 - Licensing Service
 - Life Cycle Service
 - Lightweight Log Service
 - Management of Event Domains
 - Naming Service
 - Notification Service
 - Persistent State Service
 - Property Service
 - Query Service
 - Relationship Service
 - Security Service
 - Telecoms Log Service
 - Time Service
 - Trading Object Service
 - Transaction Service

- ◆ Most CORBA*services* implementations have been monolithic servers
 - ❑ Single standalone process or program
 - ❑ On first examination, makes some sense
 - Interface, implementation, and execution environment delivered in one package
 - Persistence, recovery, etc. features bundled into server
 - ❑ Reinforced by resolve_initial_references mechanism

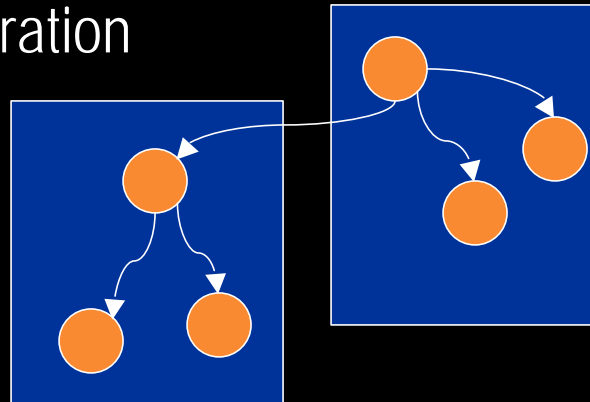
The_Event_Channel =

```
CosEventChannelAdmin::EventChannel::_narrow(  
orb->resolve_initial_references("EventService"));
```

- Only one
- Statically configured by Object URL or proprietary means

Fine-Grained CORBA *services*

- ◆ But, services are specified in “fine-grained” manner
 - ❑ Service interfaces are (fine-grained) CORBA objects
 - Location-transparent
 - Could be in same process as client of service
 - ❑ Services include features for collaboration or composition
 - Name Service – federation



- Event/Notification
 - ❖ Separation of administrative interfaces from producer and consumer interfaces
 - ❖ Setup deliberately more complicated than needed – in order to support “joining” of event channels



Fine-Grained CORBA services "Obvious" Advantages

- ◆ **Embedded systems may not have**
 - ❑ Resources for separate processes
 - ❑ Processes (single address space O/Ses)
 - ❑ Disks for persistent storage
- ◆ **Fine-Grained services can be delivered as library-based implementations**
 - ❑ Can be integrated with ORB-provided and user-selected
 - Alternate transports
 - Real-time CORBA feature usage
 - ❖ Banded connections
 - ❖ Priority propagation
 - Security
 - ❑ "Collocation" optimizations



Fine-Grained CORBA Services Example Implementation

- ◆ ORBexpress Names: Embedded Objects & ORBexpress Events: Embedded Objects
 - ❑ Library-based implementations
 - ❑ Relatively small footprint

Browser mvme2604_1@psi

Memory Usage

Tools: 522256

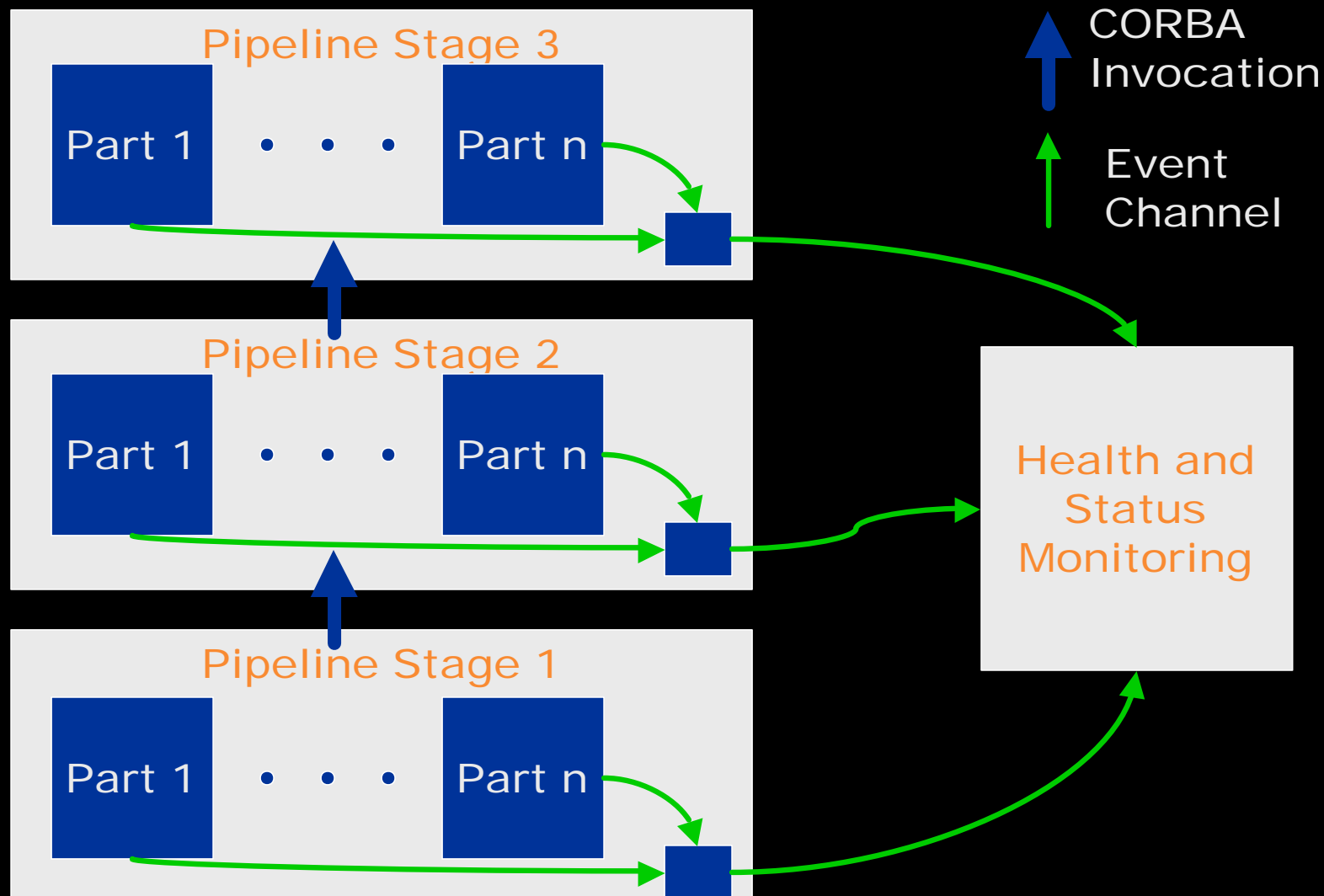
Application: 2665920

ID	NAME	.text	.data	.bss
0x5b8d0	vdWorks	895260	110092	5576
0x661e0	libOEorb.out	291088	360	120
0xed7e0	libOEtcp.out	10568	48	1
0xec260	libOEnames.out	79120	156	0
0xf6458	libOEevents.out	114400	204	0
0xf7000	Counter.out	22592	76	1
Total:		1413028	114352	36698

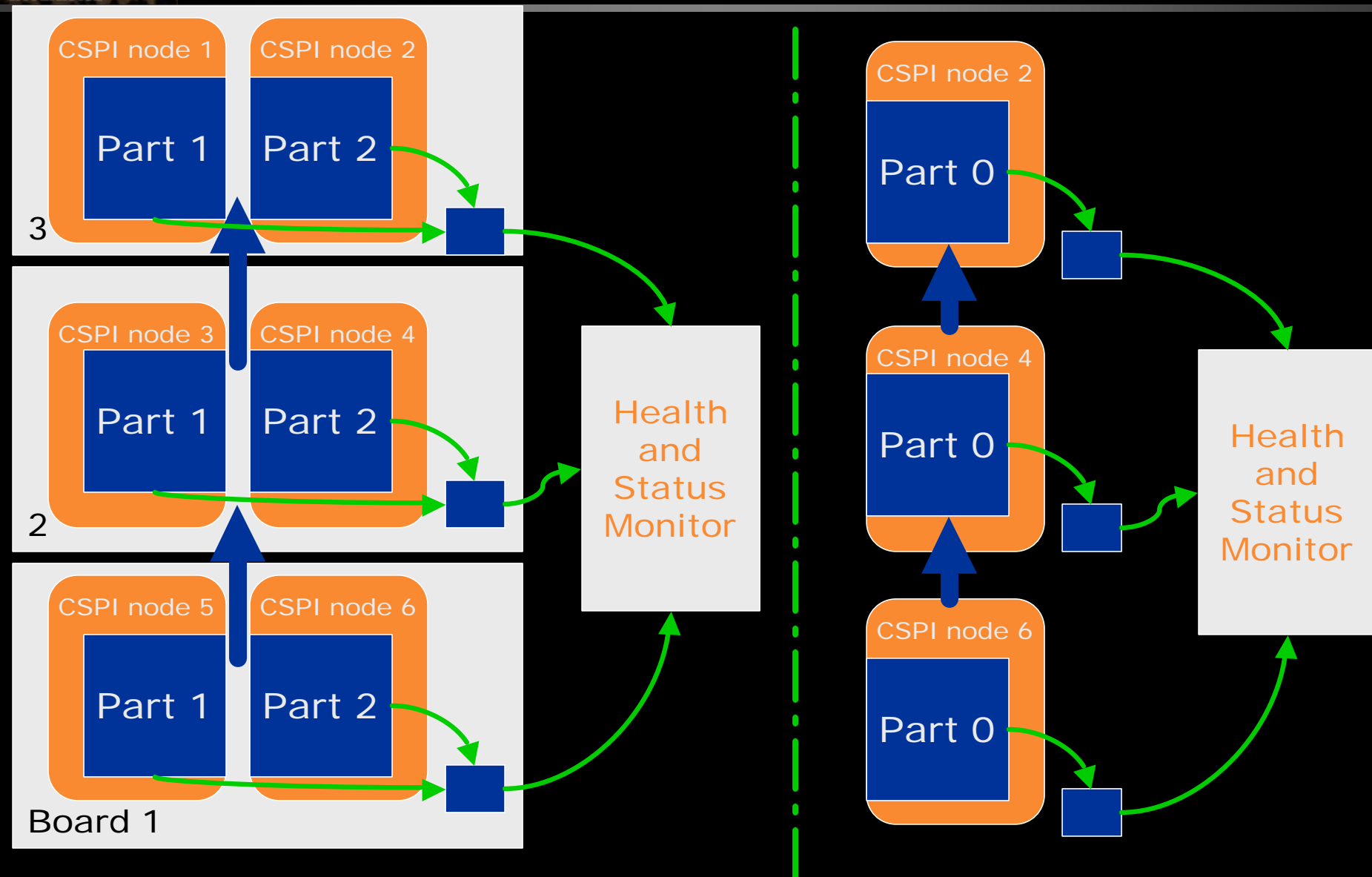
Names Library: 80K

Events Library: 115K

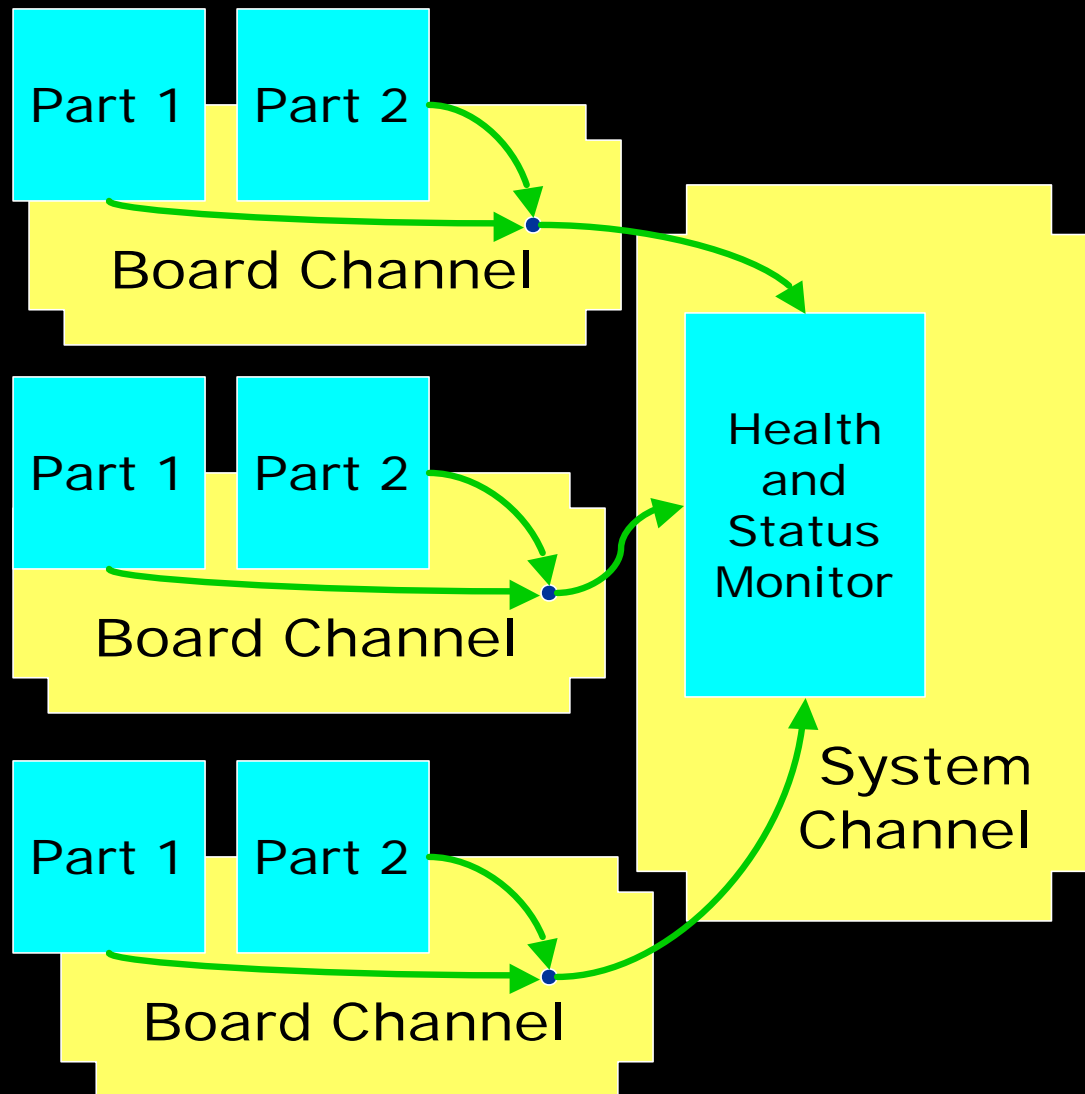
Prototype Application – Functional View



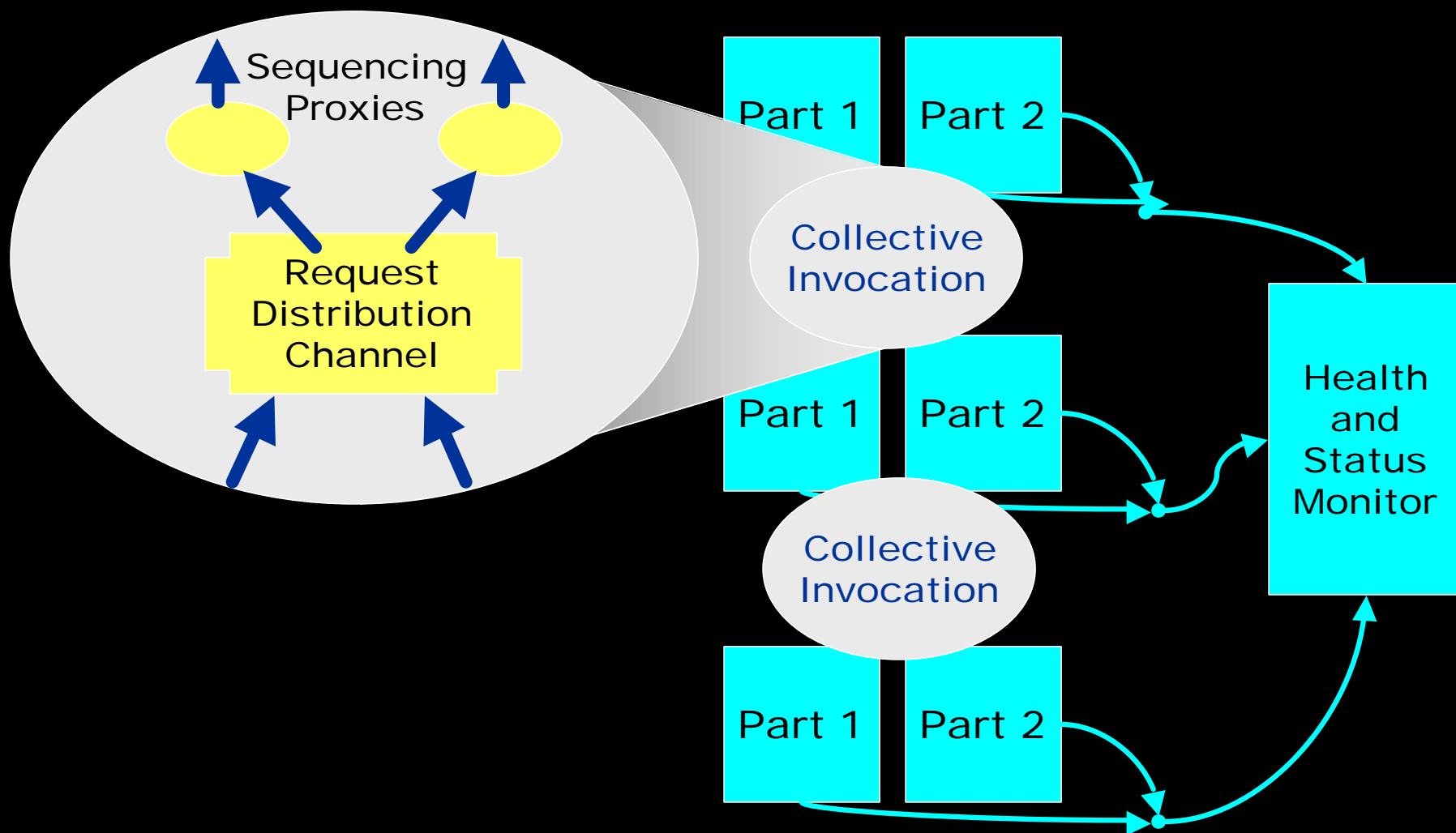
Prototype Application - Two Physical Configurations



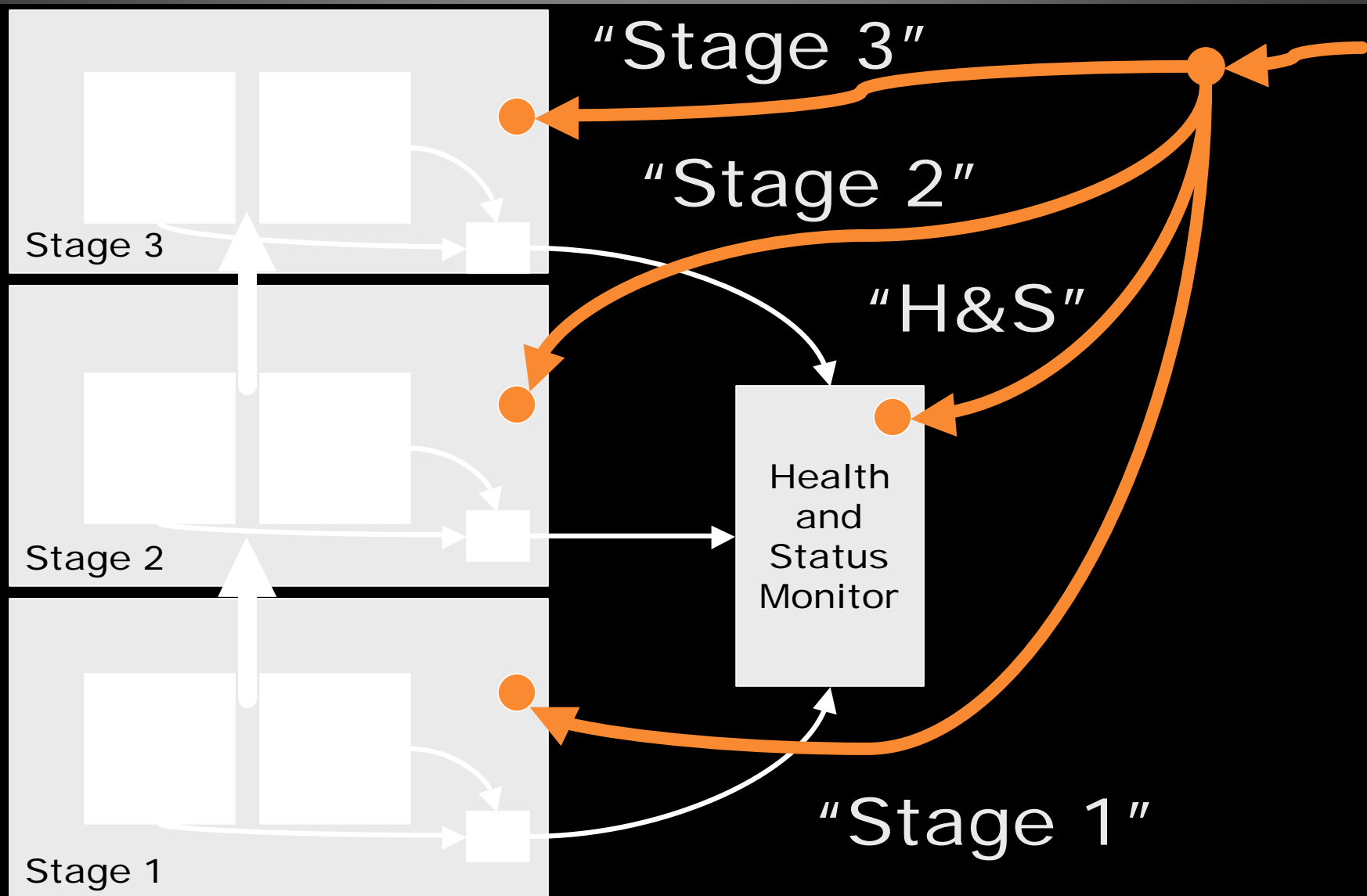
Prototype Application Using "Chained" Event Channels



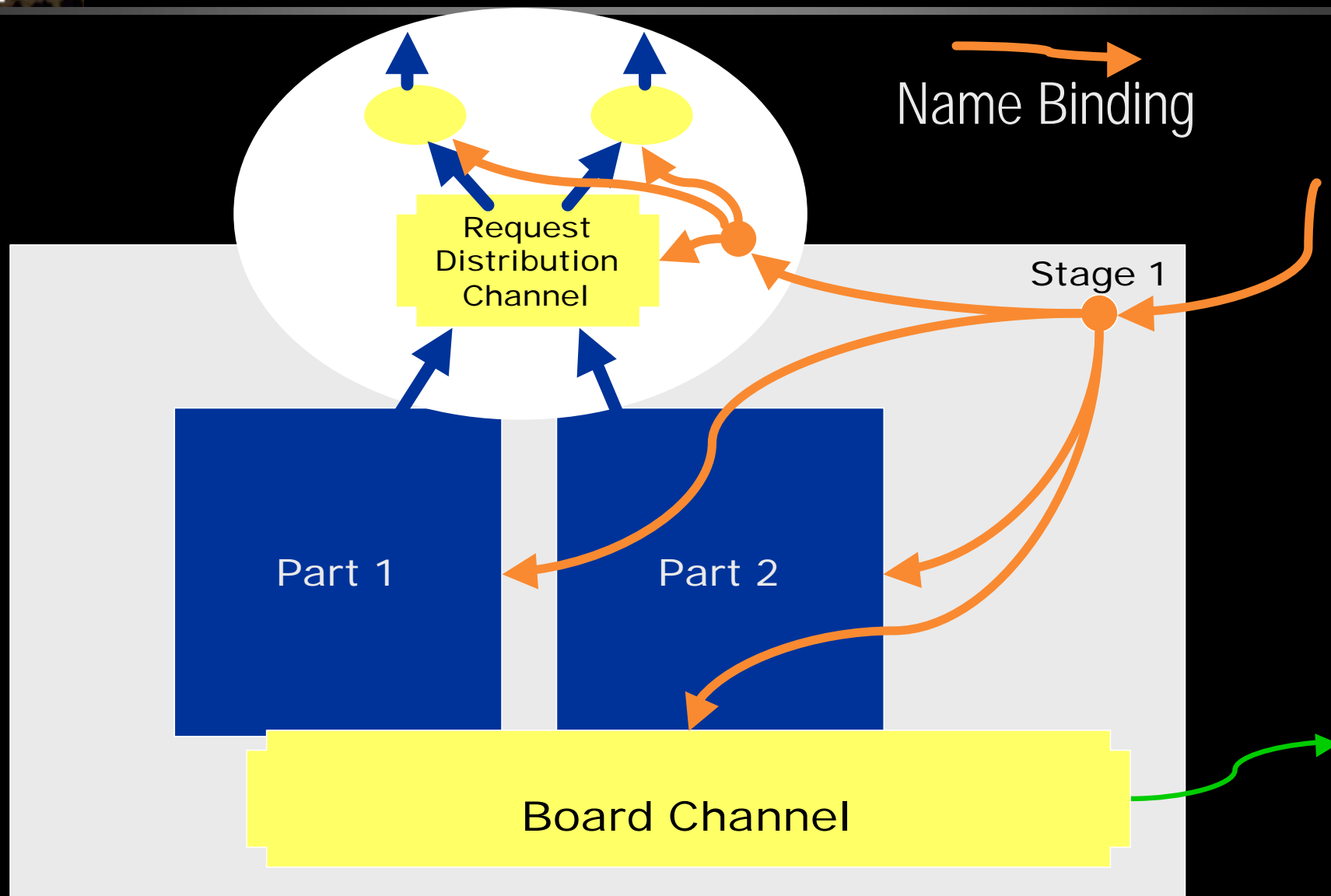
Prototype Application Using "Local" Event Channels



Using Fine-Grained Name Service to Reflect Components



Using Fine-Grained Name Service for Sub-Components



"Architectural Glue" in Prototype Application

◆ Fine-grained event service

- ❑ "Chained" event channels – on-board channel feeds system channel
- ❑ "Local" event channels – single-typed channel provides data transfer for collective invocation

◆ Fine-grained name service

- ❑ "Federated" fine-grained name services reflect
 - Application structure
 - Application deployment
 - Sub-component structure and implementation
- ❑ Simple naming convention allows reflection and dynamic query



Fine-Grained CORBA services

What is Missing?

- ◆ **Library-based Implementations**
- ◆ **Factory interfaces**
 - ❑ Name Service - can't create a NamingContext without a reference to a NamingContext
 - ❑ Event Service – no factory for EventChannel (addressed in Notification Service)
- ◆ **Location controls**
 - ❑ new_context creates NamingContext at same location as "parent" NamingContext
 - ❑ ConsumerAdmin, SupplierAdmin and Proxies will be created collocated with EventChannel
 - ❑ Deployment and Configuration Specification??