OVERVIEW OF xTCA SOFTWARE

Service Availability Forum (SA Forum) Hardware Platform Interface (HPI) Application Interface Specification (AIS)

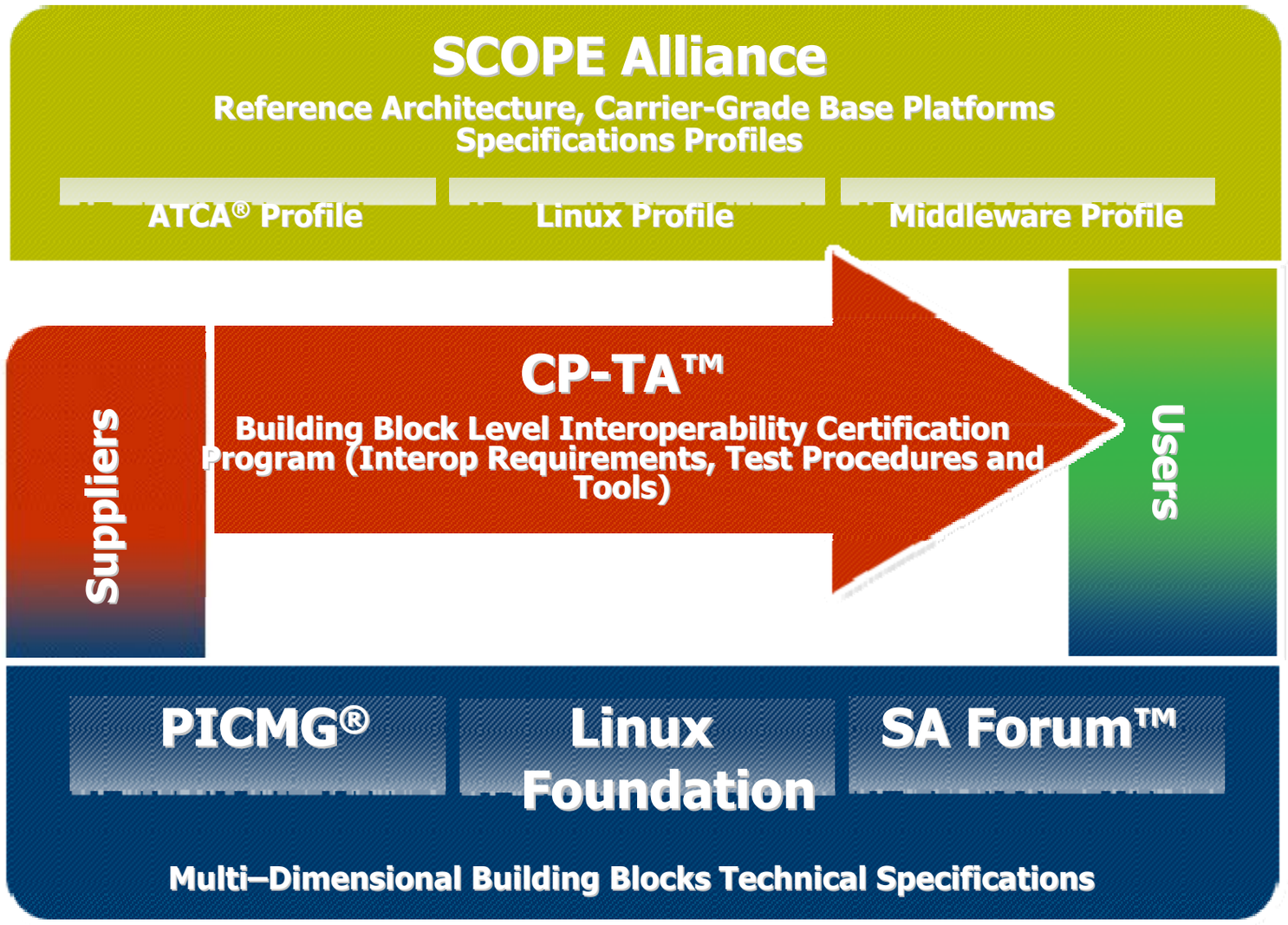
Artem Kazakov
KEK/SOKENDAI, Japan
kazakov@gmail.com

RT2009 ATCA Workshop, IHEP, Beijing, May 10 2009

AGENDA

- **Open specifications**
- **SA Forum overview**
- **SA Forum specifications**
  - Hardware Platform Interface
  - Application Interface Specification

Special Interest Group (SIG) Eco-System Landscape



Courtesy of J. Fryer

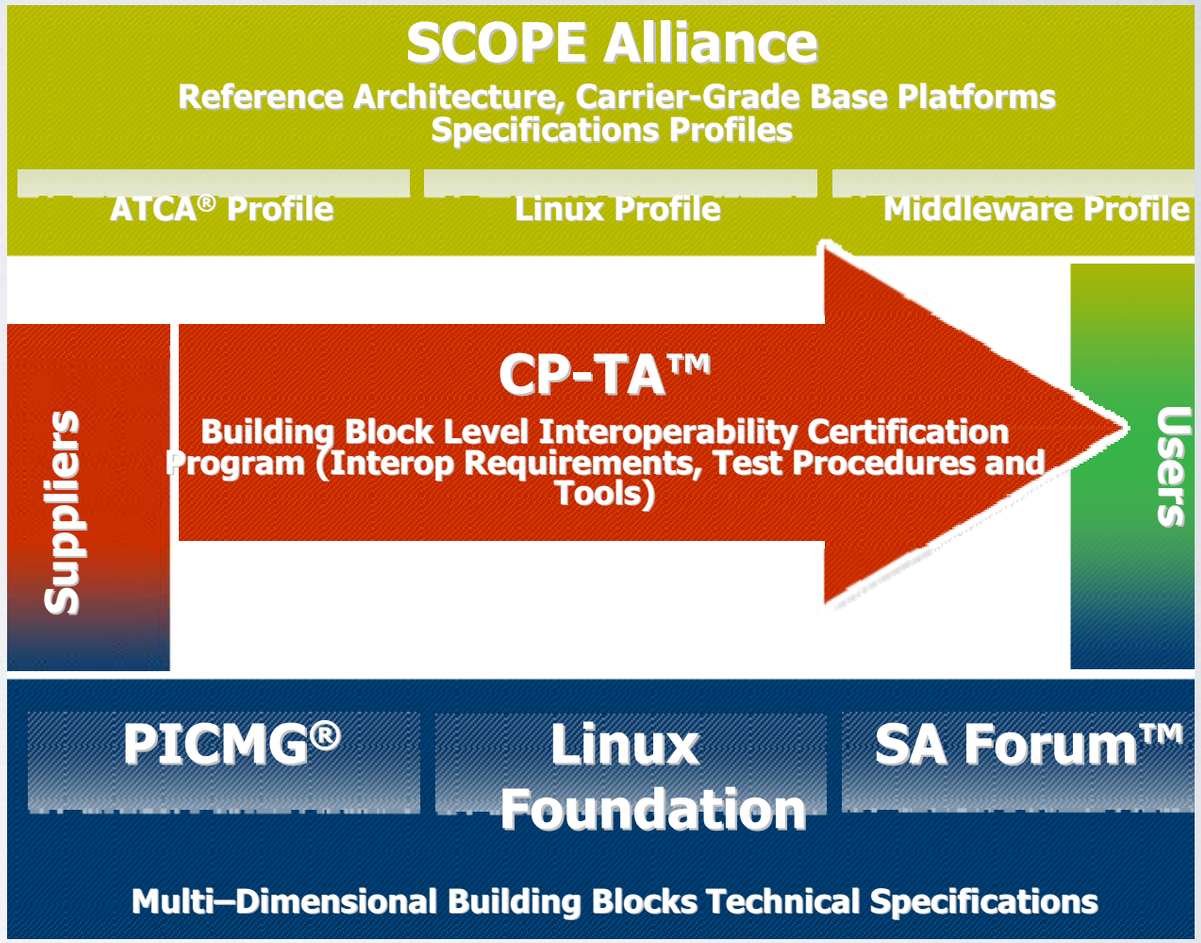
- **SCOPE alliance -** SCOPE is an association of Network Equipment Providers aimed at accelerating the deployment of Carrier Grade Base Platforms for service provider applications.

- **CP-TA** - Communications Platform Trade Association - CP-TA is focused on defining and managing interoperability testing for PICMG's Advanced Telecom Computing Architecture (AdvancedTCA)

- **PICMG** - PCI Industrial Computer Manufacturers Group

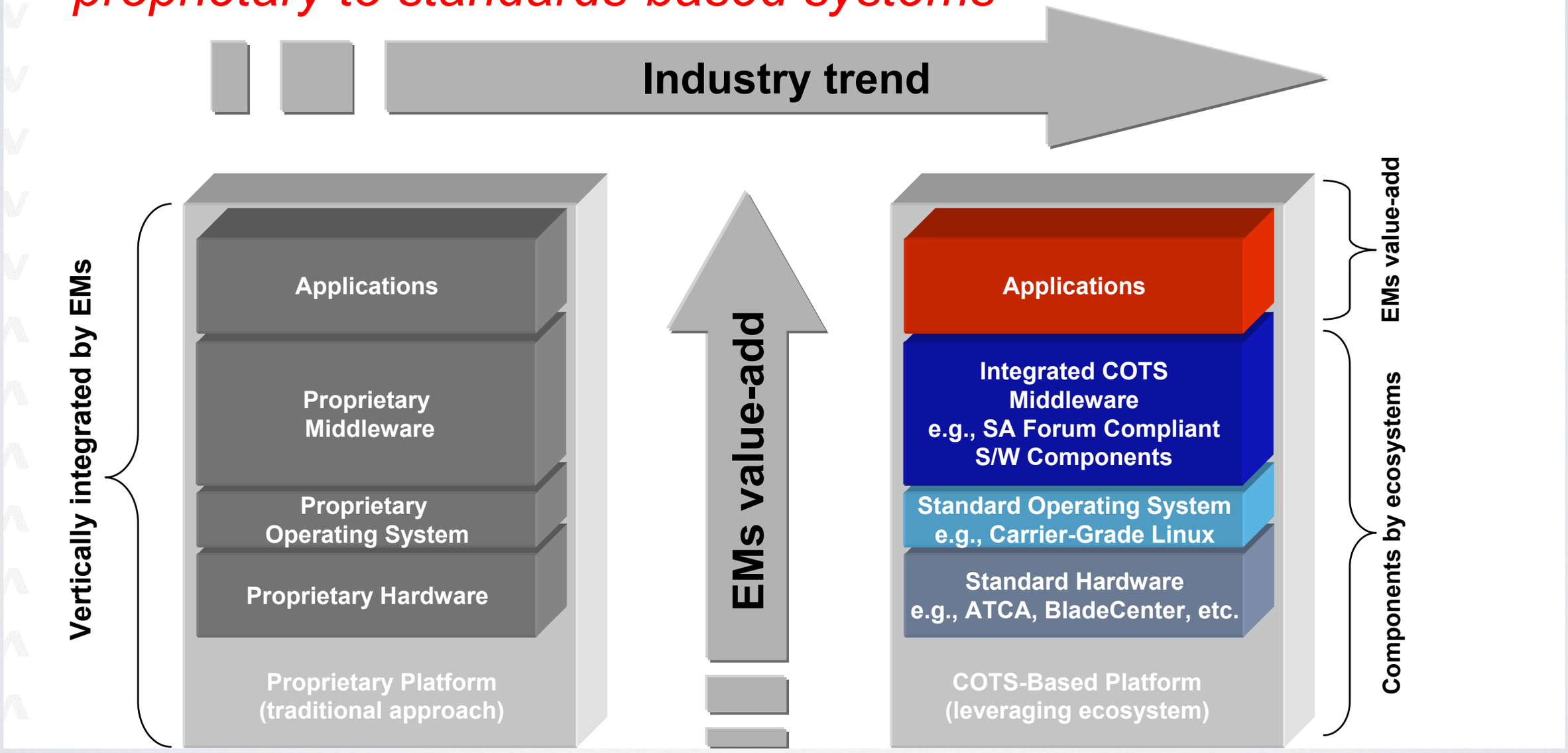
- **Linux Foundation** - helps to close the gap between open-source and proprietary platforms.

- **SAF** - service availability forum



Role of Open Specifications

COTS adoption is accelerating transition from a vertical to horizontal industry model
AND key open specifications are catalyzing a move from proprietary to standards-based systems



Courtesy of J. Fryer

SERVICE AVAILABILITY FORUM OVERVIEW

The Service Availability Forum™ is a consortium of industry-leading communications and computing companies working together to develop and publish high availability and management software interface specifications.

SERVICE AVAILABILITY FORUM

Mission

- Foster an ecosystem to enable the use of COTS building blocks in the creation of high service availability network infrastructure products, systems and services

Objectives

- Develop and publish standard interfaces for carrier grade systems that provide
  - High Availability
  - Service Continuity
- Promote and facilitate their adoption by the industry

SA Forum Members

Alcatel·Lucent
Astute Networks
MŰEGYETEM 1782
Université Concordia University
Continuous Computing
embeddedMIND
EMERSON Network Power
ENEA
ERICSSON
FUJITSU COMPUTERS SIEMENS
FUJITSU
GOAHEAD
hp invent
HUAWEI
IBM
Industrial Technology Research Institute
intel
Designing The Future KDDI
montavista
MOTOROLA
NEC
Nokia Siemens Networks
NORTEL NETWORKS
NTT
OpenClovis
ORACLE
PERFORMANCE TECHNOLOGIES
PIGEON POINT SYSTEMS
RadiSys THE POWER OF WE
Sonus NETWORKS
Sun microsystems
SYBASE
tail-f
TietoEnator
virtualLogix
WIND RIVER
xeround The Telco Database Company™
SERVICE AVAILABILITY™

Courtesy of J. Fryer

SA Forum Membership Categories

Adopter Member

- Eligible to register SA Forum-compliant products
- Eligible to use SA Forum logo and SA Forum association to promote products
- Inclusion in SA Forum marketing collateral, recruitment presentations and press releases
- Company name listed in Public area of SA Forum Web site as a member
- Participate in SA Forum-related Industry events as a SA Forum Member
- $2,000 annual commitment

Contributor Members (additional benefits)

- Eligible to attend Member meetings and participate in work groups
- Early access to draft specifications
- Eligible to develop, review and comment on specifications
- Access to the member-secured area of the Web site
- $10,000 annual commitment

Associate Members

- Open to Academic Institutions
- Participate in specification workgroups (teleconferences, email communication)
- Review and comment on preliminary drafts of specifications
- No cost to participate
- Requires Promoter Member sponsorship and Board Approval to join

Promoter Members (additional benefits)

- Note: Promoters can only be added by unanimous vote of Board of Directors
- Eligible for Board of Director and Officer positions
- Voting rights for bylaw changes & officer elections
- Voting rights for the ratification of final specifications
- $25,000 annual commitment

SERVICE AVAILABILITY™ FORUM

Courtesy of J. Fryer

SERVICE AVAILABILITY SOLUTION

System is available for use 99.999% of the time or more

No loss of service continuity during fault recovery or administrative actions

What end-users expect

The need to achieve Service Availability™ in a standard way inspired the founding of the Service Availability Forum (SA Forum)

To provide High Availability and Service Continuity
from COTS components
that include manageable redundancy

HIGH AVAILABILITY

- **Availability**
  - A measure of the uptime of a system
  - The probability that the system is able to provide service when requested

$$Availability = \frac{MTFB}{MTBF+MTTR}$$

- **High Availability**
  - at least 99.999%

Availability	Downtime per year
90%	36.5 days
99%	3.7 days
99.9%	9 hours
99.99%	53 min
99.999%	5 min
99.9999%	32 sec

SERVICE CONTINUITY

- **Service continuity** attempts to **maintain end-user sessions** in spite of failures of individual systems or components, and in spite of fault recovery, maintenance or system management actions
- **The state** of the application session for each user **must be preserved** during switchover and failover
- Service continuity is provided with the cooperation of the application software
- Thus **standard interfaces are required** to ensure portability of the applications across diverse hardware and software platforms

REDUNDANCY

$$Availability = \frac{MTFB}{MTBF+MTTR}$$

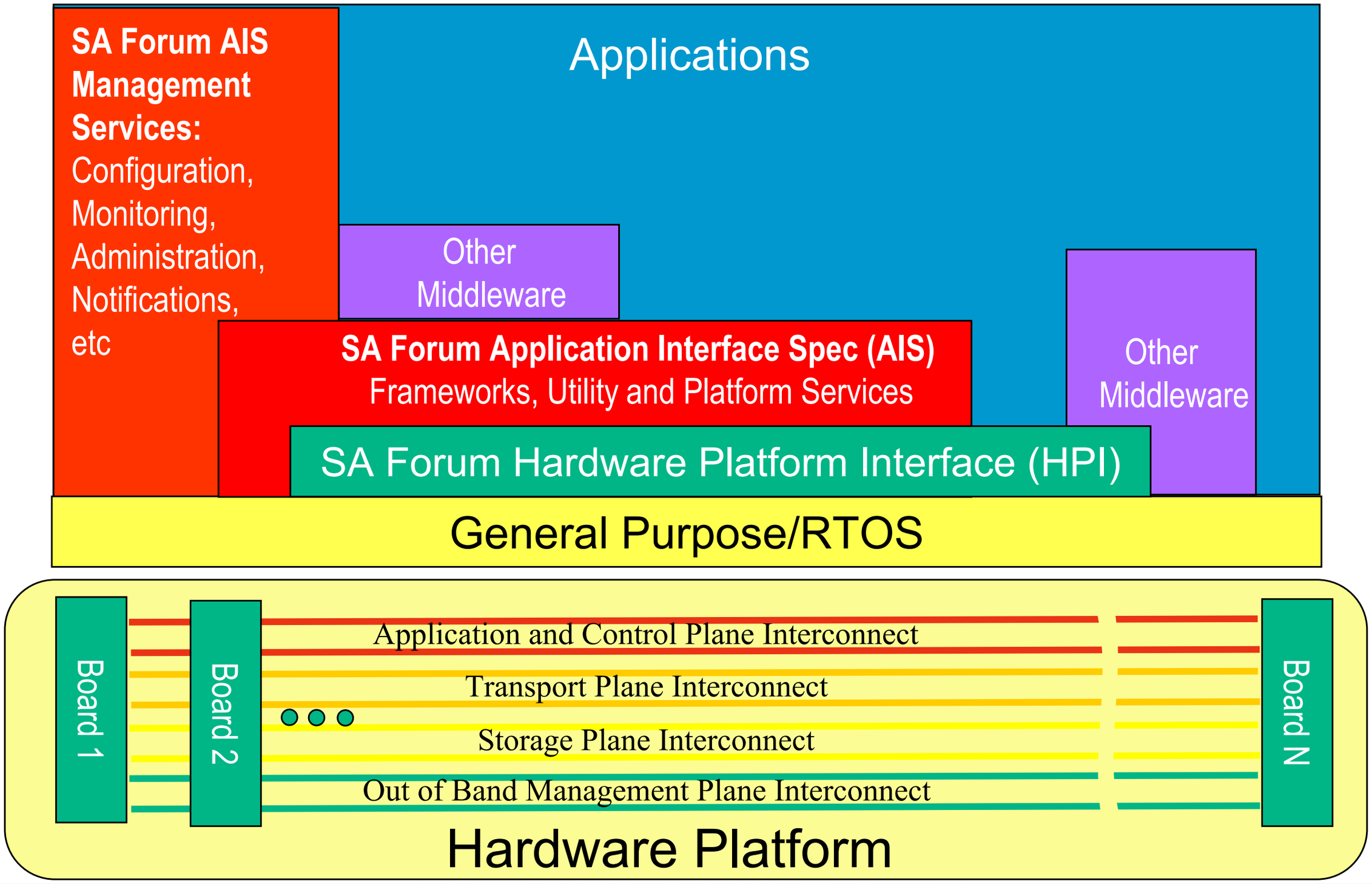
✦ Problem

- Real system has many hardware and software components, thus it is hard to achieve high availability by increasing MTBF

✦ Solution

- use redundant components to achieve high availability by decreasing MTTR to close to 0, from hours or minutes to a few milliseconds

Basic Architecture View



Courtesy of J. Fryer

CURRENT STATUS

- **Specification work**
  - HPI: Solid, mature specification
  - AIS: relatively new, but solid set
- **Implementation**
  - HPI is a commercial success
  - AIS has several free implementations
  - Some vendors have internal HPI and AIS implementation programs

Industry Adoption

HPI is already a commercial success

Hardware Platform Providers

- **Augmentix**
- **Continuous Computing**
- **Emerson ECC**
- **Hewlett Packard**
- **IBM**
- **Intel**
- **Kontron**
- **Performance Technologies**
- **Pigeon Point**
- **RadiSys**
- **Sun Microsystems**
- **And others**

Middleware Providers

- **ENEA**
- **GoAhead Software**
- **Others**

Open Source Implementation

- **OpenHPI**

HPI Tester

- **Polaris**

SERVICE

Courtesy of J. Fryer

Industry Adoption

AIS implementations available in the market

Pre-Integrated Platform Providers

- **Continuous Computing**
- **Emerson Network Power-EC**
- **Fujitsu Siemens Computers**
- **Hewlett Packard**
- **IBM**
- **RadiSys**
- **Sun Microsystems**
- **Others**

Middleware Providers

- **ENEA**
- **GoAhead Software**
- **OpenClovis**
- **Emerson Network Power- EC**

Open Source Implementations

- **OpenAIS**
- **OpenClovis**
- **OpenSAF**

Courtesy of J. Fryer

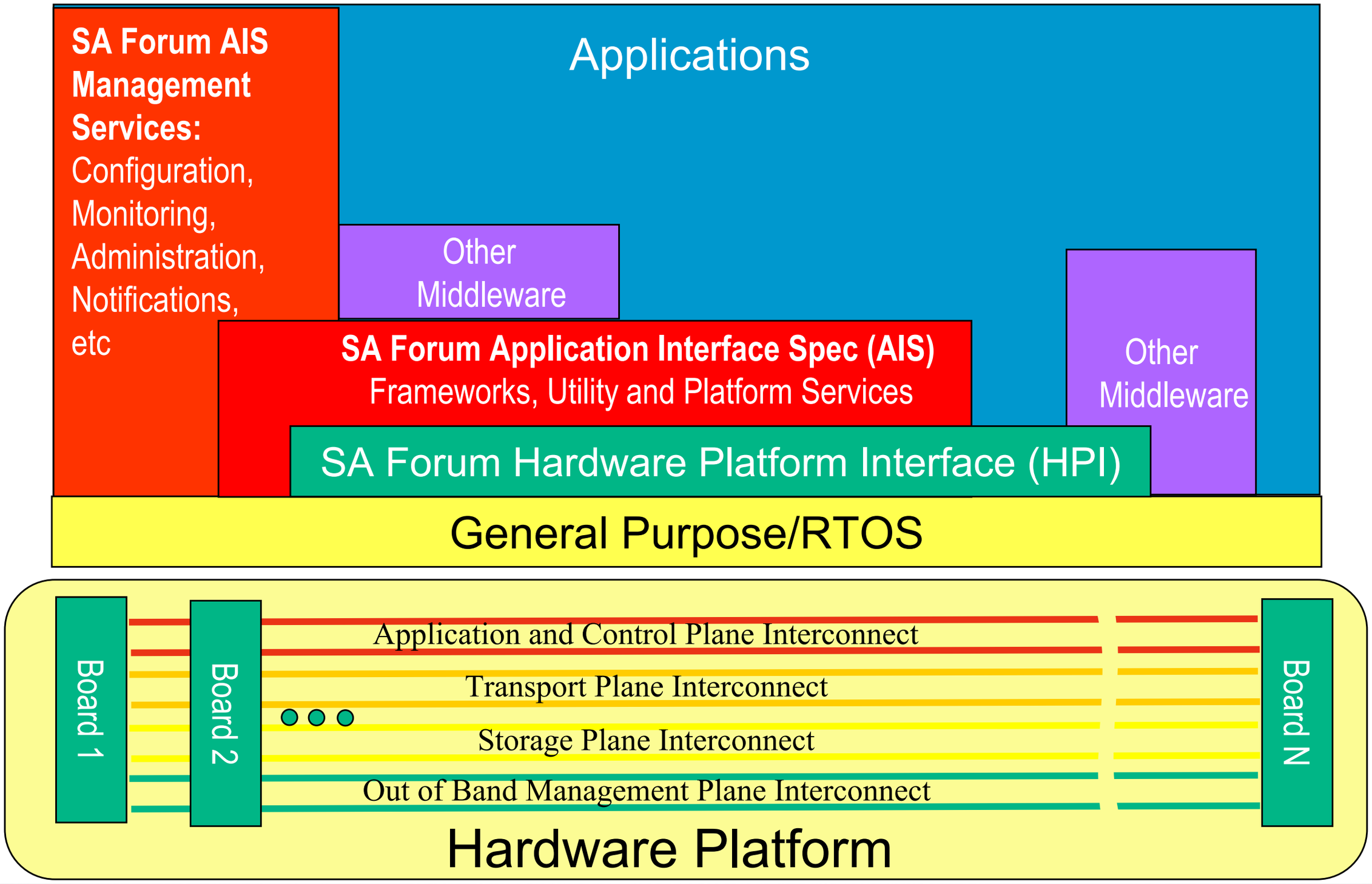
ROAD AHEAD

- **Attract application providers**
  - Create "critical-mass" for the platform
- **Create example applications**
  - Demonstrate application portability and platform features
  - Establish best practices

SERVICE AVAILABILITY FORUM SPECIFICATIONS OVERVIEW

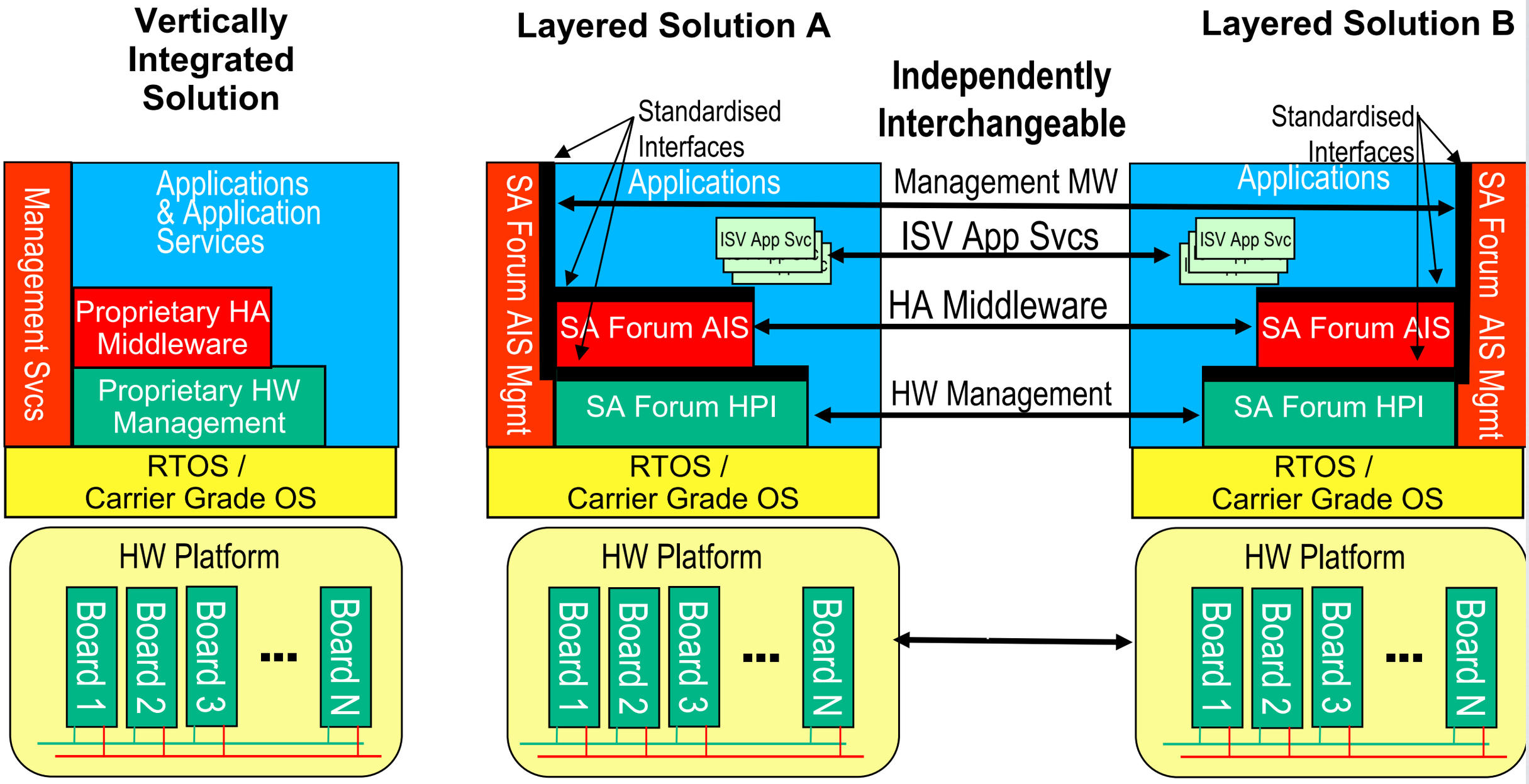
- Architecture
- Hardware Platform Interface
- Application Interface Specification

Basic Architecture View



Courtesy of J. Fryer

SA Forum Specifications for Portability



Courtesy of J. Fryer

Hardware Platform Interface (HPI)

- *APIs for fine grained low level access to hardware:*
  - Inventory
    - HPI accessible hardware managed by domains found in
      - HPI Domain presence table
    - Within each domain is a Resource Presence Table describing discovered management resources
  - HPI resource functions
    - Each resource has a Resource Data Record Repository for each of its managed entities. These contain
      - Instrument capabilities, inventory data (e.g. manufacturer, serial no, version, asset tag etc)
    - Instrumentation:
      - Sensor, Control, Inventory Data Repository, Watchdog Timer, Annunciator (LEDs), Diagnostics and Firmware Upgrade
    - Management Capabilities :
      - Hot-swap, configuration, load, reset, and power management
  - HPI per domain: events, event logs and alarm table

Application Interface Specification (AIS)

- *Platform Services*
  - Platform Management service provides a convenient abstraction to monitor state changes in groups of entities comprising hardware, hypervisors and operating systems
  - A cluster management service that provides a consistent view of the healthy configured nodes in a cluster
- *Basic management services: Standard interfaces for applications and middleware:*
  - Configuration & runtime management information
  - Administrative commands
  - Notification for alarms, state changes, object lifecycle changes, attribute changes and security alarms
  - Logging of alarms, notifications, system messages and application specific streams
  - Access security functions for the AIS and HPI services
- *Frameworks*
  - Availability management of applications and middleware
  - Software management for managing HW, OS, middleware and application upgrades taking service availability into account
- *Utility Services provide common services required in distributed HA systems including*
  - Checkpointing, Event distribution, distributed locks, message passing, name lookup and timer services

OVERVIEW OF **H**ARDWARE **P**LATFORM **I**NTERFACE (**HPI**)

Reasons for HPI
Basic concepts
Example of usage

WHY HPI?

- **Hardware platforms used for High Service Availability applications are rich in platform management capabilities**
  - Monitoring and controlling the hardware platform is essential for High Service Availability
  - Hardware platform differences lead to non-portable applications
- **SA Forum has developed the Hardware Platform Interface (HPI) specification to address these issues**
  - The HPI is a programmatic interface that enables management middleware applications to monitor and control platform hardware
  - The HPI allows COTS building blocks to be integrated into coherent systems that provide High Service Availability

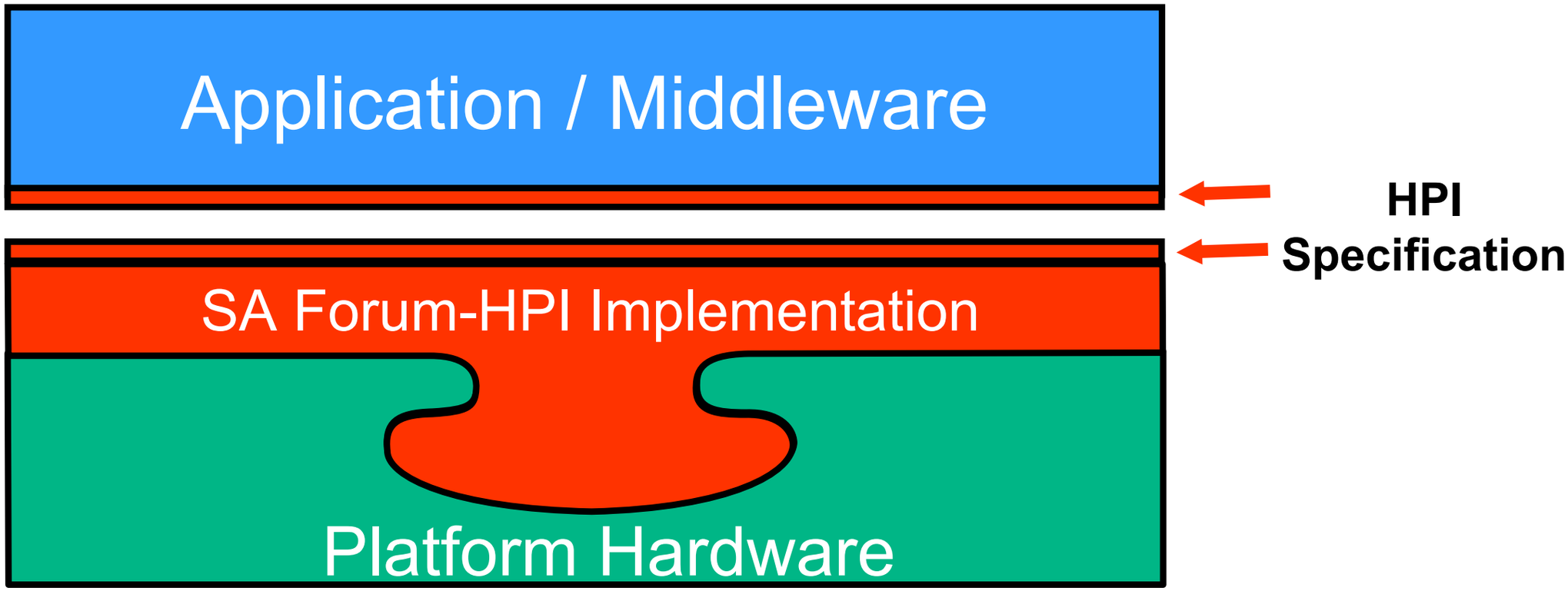
What is HPI?

- *Defines an API for abstract access to the platform's management capabilities by a management application ("HPI User")*
  - API: C headers (standard) and libraries (provided by a vendor).
  - Mostly used remotely, but remote access is vendor specific.
- *Provides an access to the hardware:*
  - Configuration: what hardware components are in the system;
  - Inventory: serial numbers, vendors…
  - Hot Swap: on-line insertion and extraction of field replaceable units;
  - System's well-being: temperature, voltage, fan speed, displays, LEDs…
  - Power on and off, resets, hardware watchdogs
  - Utilities: Firmware Upgrade, Diagnostics
- *Abstract, support various form factors/architectures – mapping is needed*
- *Permits OEM differentiation*

Courtesy of J. Fryer

Reason for HPI

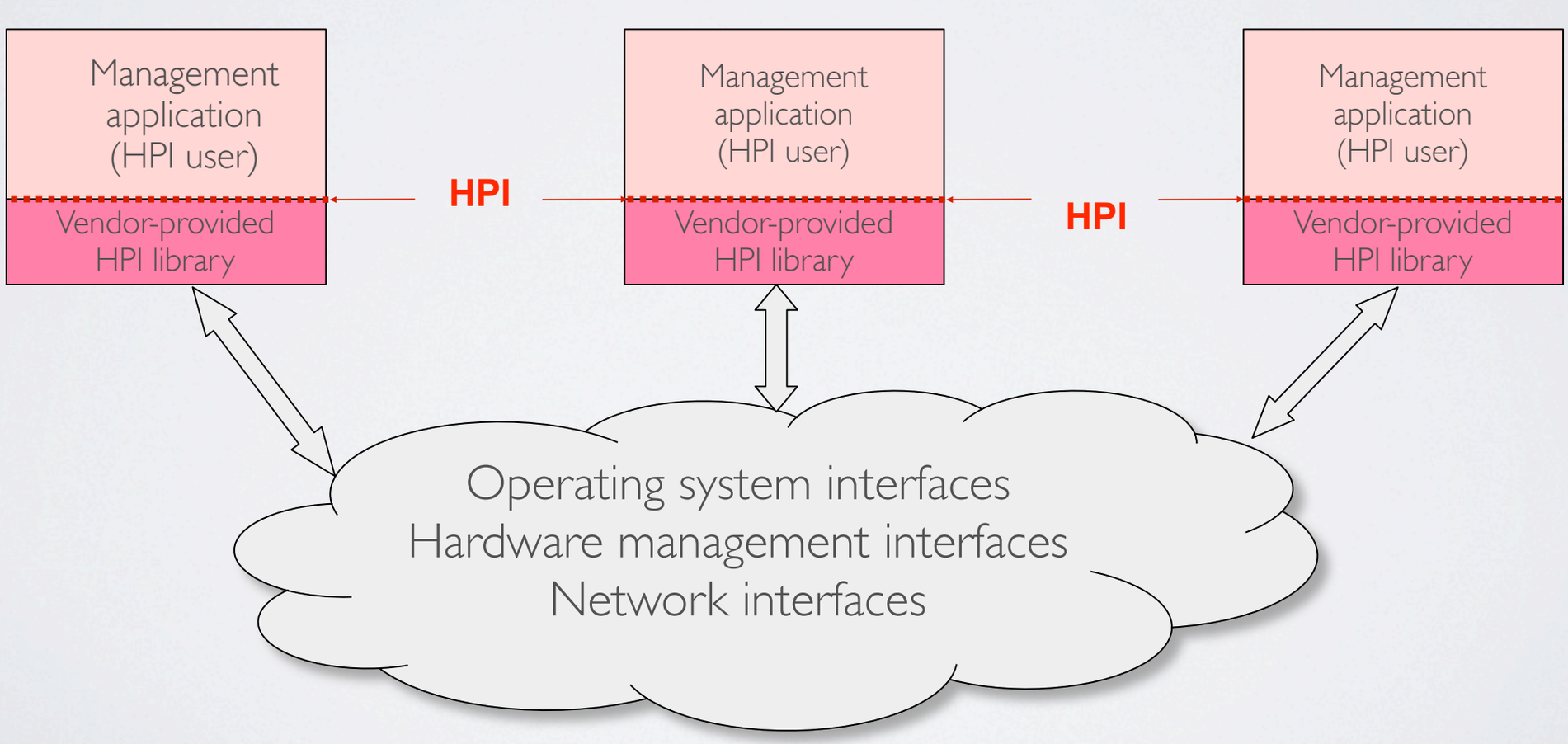
- *Hides platform specific or proprietary features from middleware and applications*
- *Allows the use of Commercial Off The Shelf middleware building blocks*
- *HPI eliminates the proprietary application/middleware interface, allowing for faster porting of applications to multiple platforms.*



Courtesy of J. Fryer

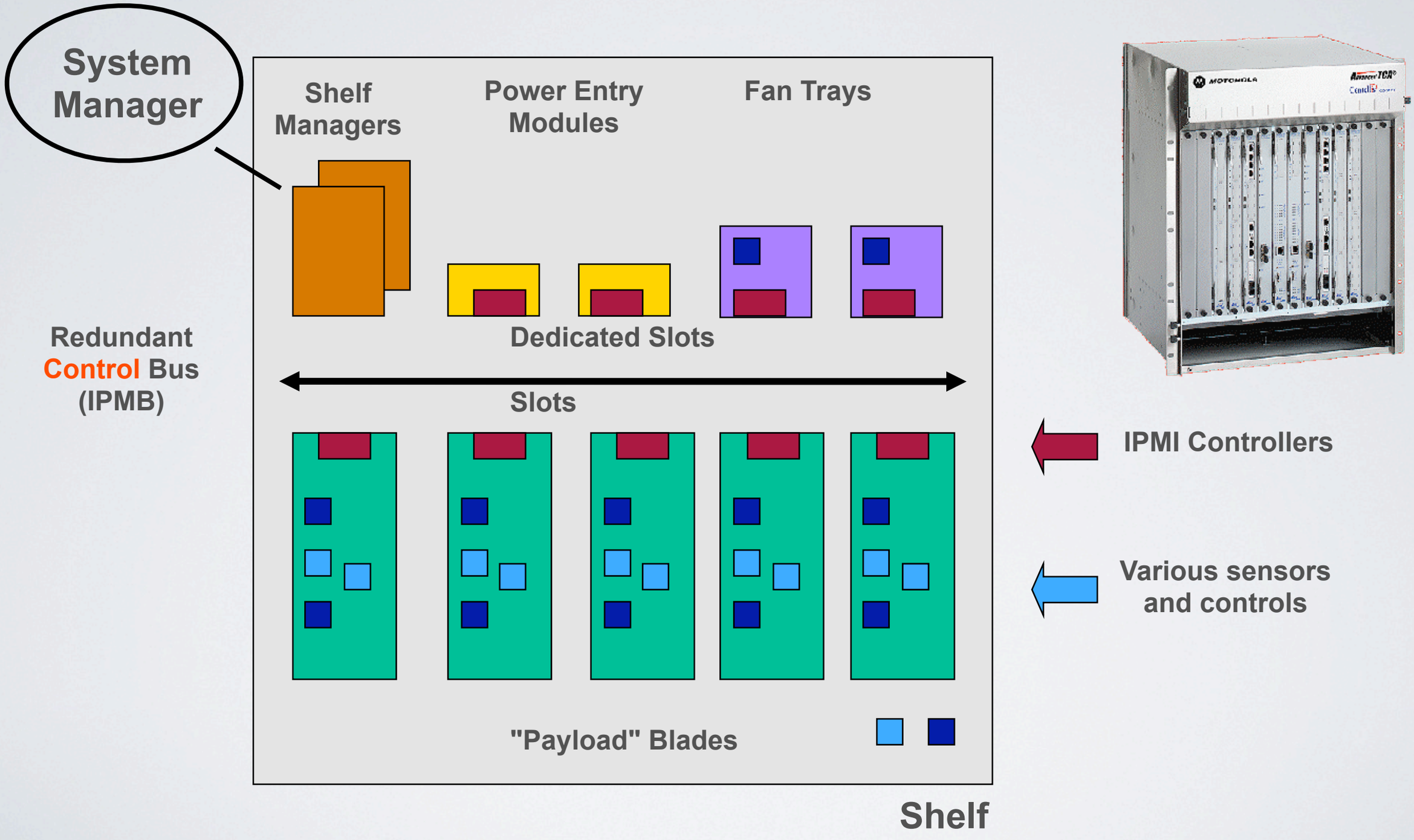
HARDWARE PLATFORM INTERFACE

- The Hardware Platform Interface (HPI) is defined as a library API
  - Library functions are specified in the C language
  - The HPI neither specifies nor implies a library implementation



Copyright© 2006 Service Availability™ Forum, Inc

EXAMPLE ATCA BLADED SERVER CONTROL ARCHITECTURE



Copyright© 2006 Service Availability™ Forum, Inc

HPI BASIC CONCEPTS

- **Domain**
- **Session**
- **Resource**
- **Entity**

Copyright© 2006 Service Availability™ Forum, Inc

DOMAIN AND SESSION

✦Domain

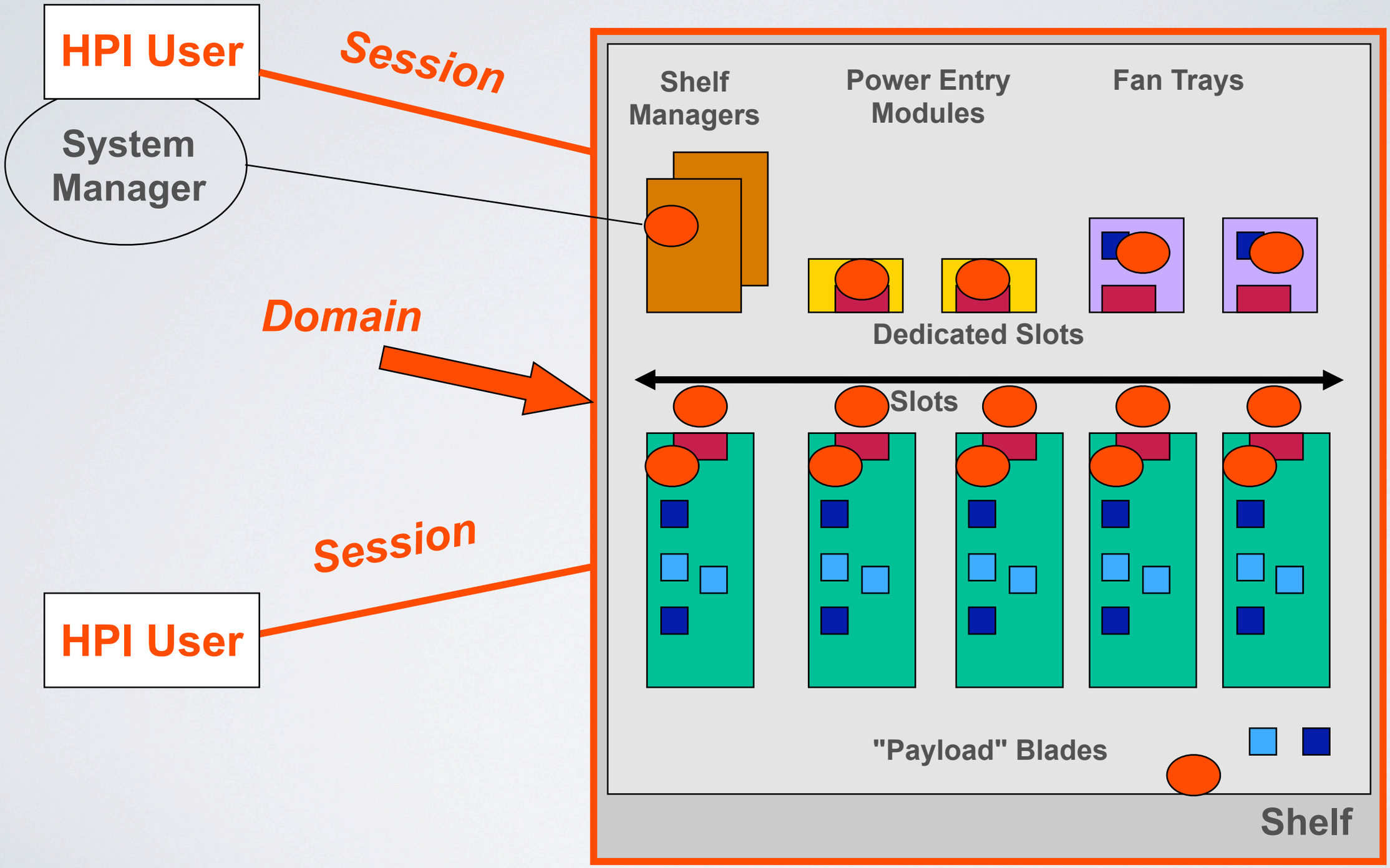
- A domain is a grouping of resources and a set of associated capabilities
- The HPI provides flexible domain / resource configurations

✦Session

- A session confines the user, so that it can access only resources in a single domain
- A user can access a domain only by opening a session for that domain.
  - The user provides the domain identifier
  - The user obtains a session identifier in return
- A user may open multiple sessions on a single domain, and multiple sessions on multiple domains

Copyright© 2006 Service Availability™ Forum, Inc

EXAMPLE DOMAINS AND SESSIONS IN ATCA



Copyright© 2006 Service Availability™ Forum, Inc

RESOURCE AND ENTITY

- **Resource**
  - Resources represent the management access to the components of the system
  - A resource provides access to management information about, and management control over, some of the components of the system, called entities
  - A resource is modeled as a set of **Resource Data Records (RDRs)** that contain information about the management instruments associated with the resource

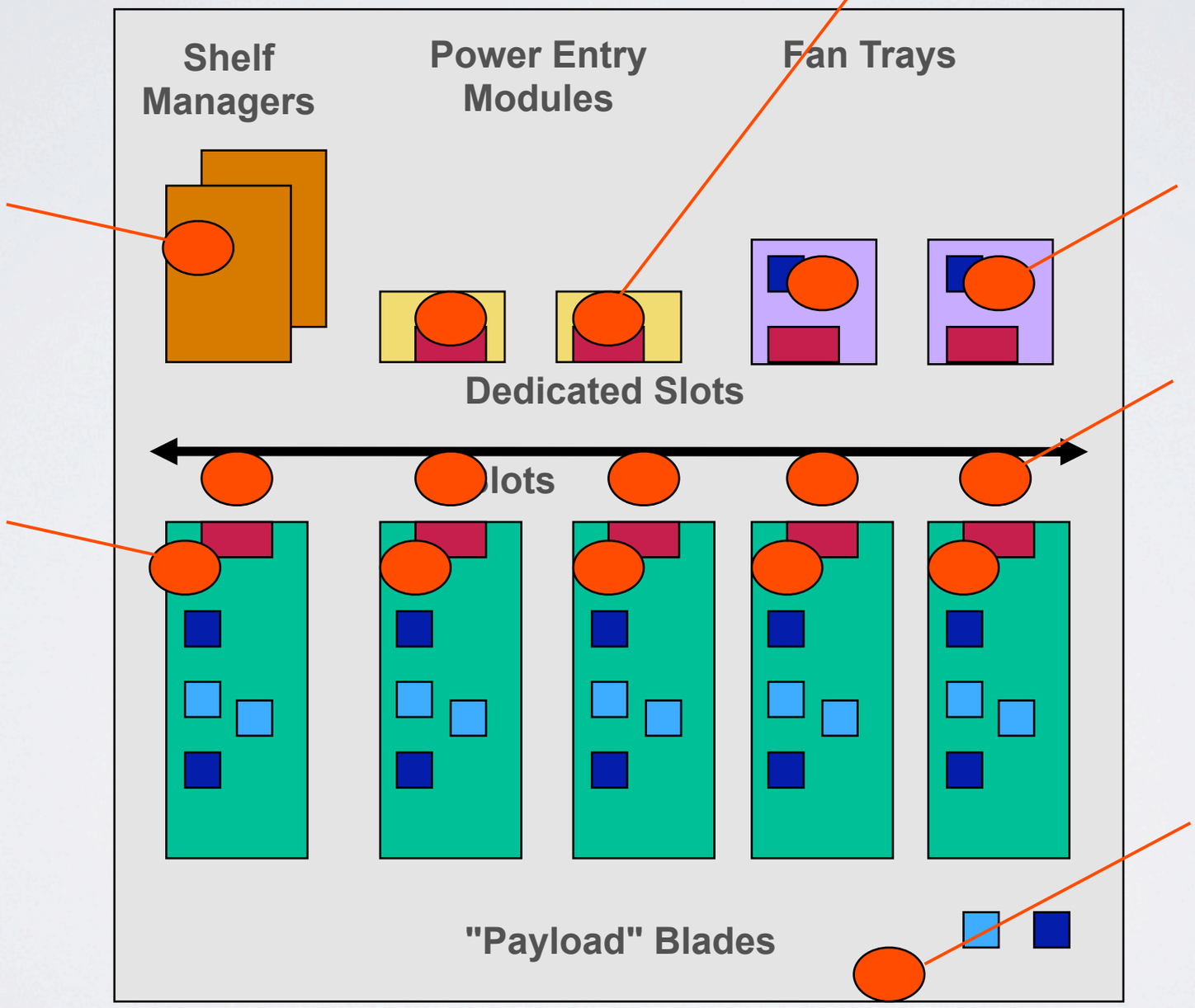
Copyright© 2006 Service Availability™ Forum, Inc

EXAMPLE RESOURCES IN ATCA

FRU Resource: PEM

FRU Resource: Fan:
C: LED
S: IPMB-0
S: Inlet temperature
S: Voltage
C: Fan speed
…

Shelf Manager Resource:
S: Redundancy
…



Slot Resources:
S: Slot State Sensor
C: FRU Activation
S: Max Power Capability
S: Assigned Power

Blade Resource:
S: Temperature
S: Voltage
I: Inventory
….

Shelf Resource:
C: Geographical Address
I: Shelf Inventory
…

Copyright© 2006 Service Availability™ Forum, Inc

RESOURCE AND ENTITY

- **Entity**
  - Entities are manageable physical components of the system
  - An entity is uniquely named by an **entity path** that identifies the component in terms of its physical containment within the system

Copyright© 2006 Service Availability™ Forum, Inc

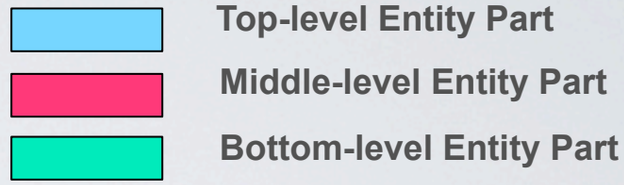
MAPPING OF ENTITIES AND ENTITY PATHS

- An **entity** is a physical hardware component
- A **HPI entity path** defines the location of the entity in the system by first identifying the entity (Board, Fan Tray, etc), then the Slot that the entity occupies, then further containment (Shelf, Chassis, etc)
  - **Top Level** – Defines elements of the entity path from the ATCA Chassis on up, given in an implementation-specific configuration file
  - **Middle Level** – Represents both Shelf and FRU building blocks, mandatory for all HPI Resources representing physical components (FRUs and Slots) in a HPI-managed ATCA Shelf, maps to site types (Slots) and locations in the ATCA Shelf
  - **Bottom Level** – Represents the physical functional entities of the Shelf, i.e., FRUs (ATCA Front Boards, Fans, PEMs), mezzanine boards (PMCs, AMCs), alarm modules (fixed or field replaceable, and vendor-specific OEM components)

Entity Path	Top Level	Shelf, Chassis, Rack, etc
	Middle Level	Slot
	Bottom Level	Board, Fan Tray, etc

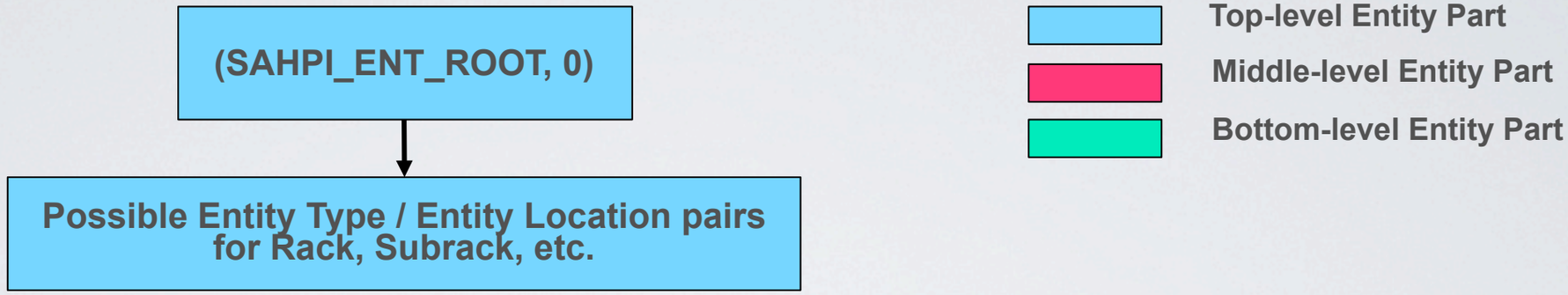
Copyright© 2006 Service Availability™ Forum, Inc

EXAMPLE ENTITY PATH TREE



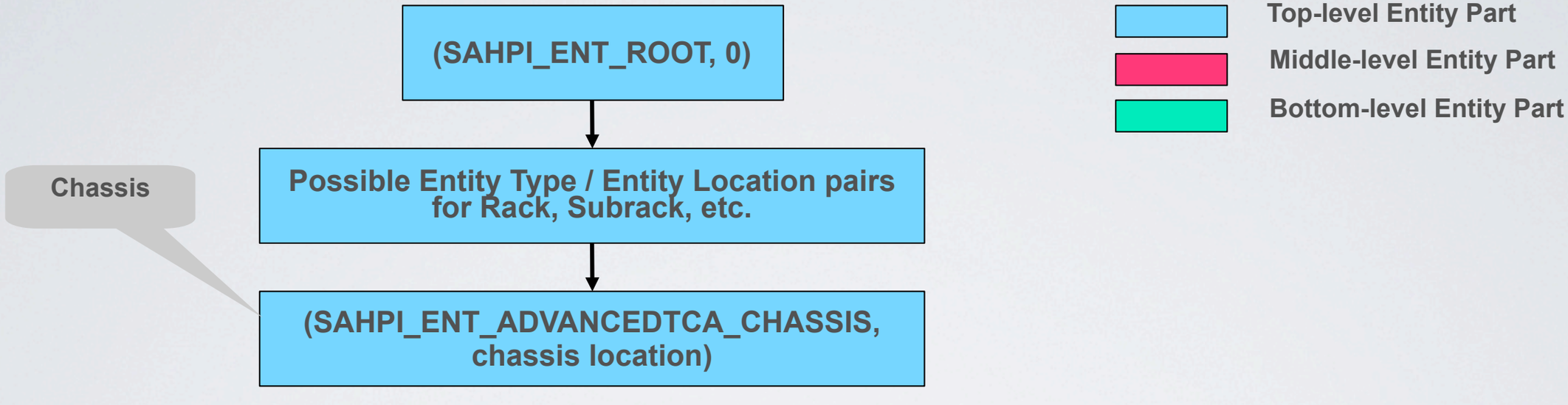
Copyright© 2006 Service Availability™ Forum, Inc

EXAMPLE ENTITY PATH TREE



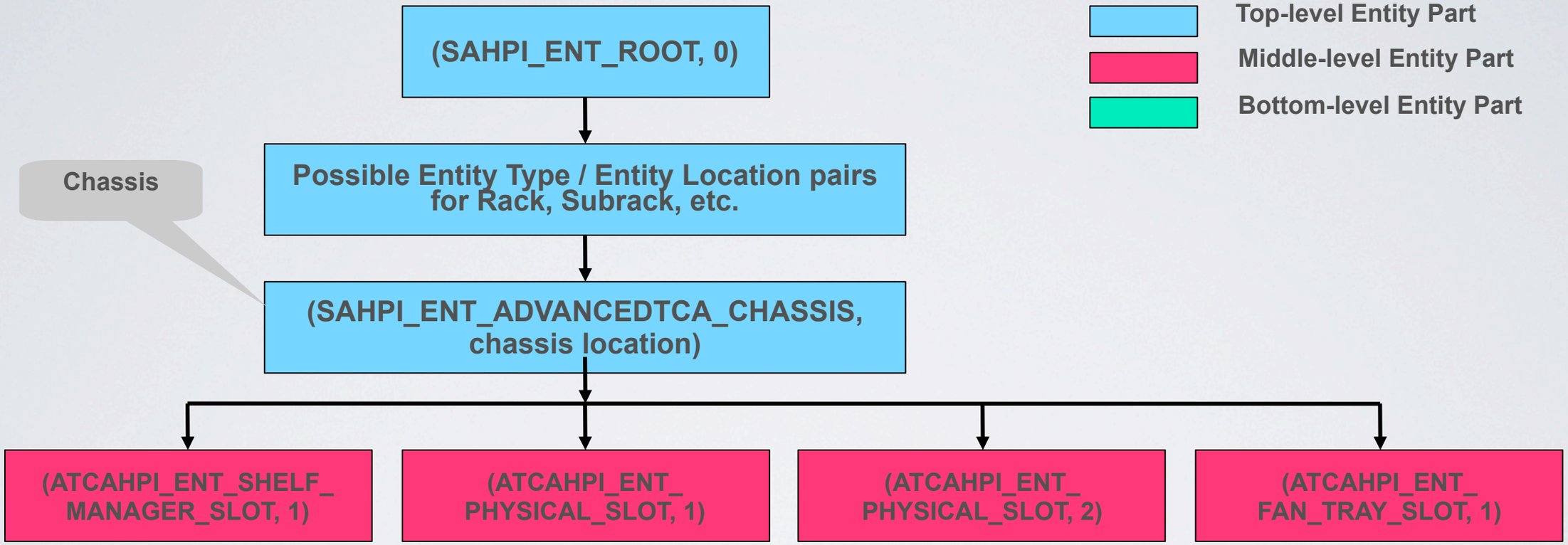
Copyright© 2006 Service Availability™ Forum, Inc

EXAMPLE ENTITY PATH TREE



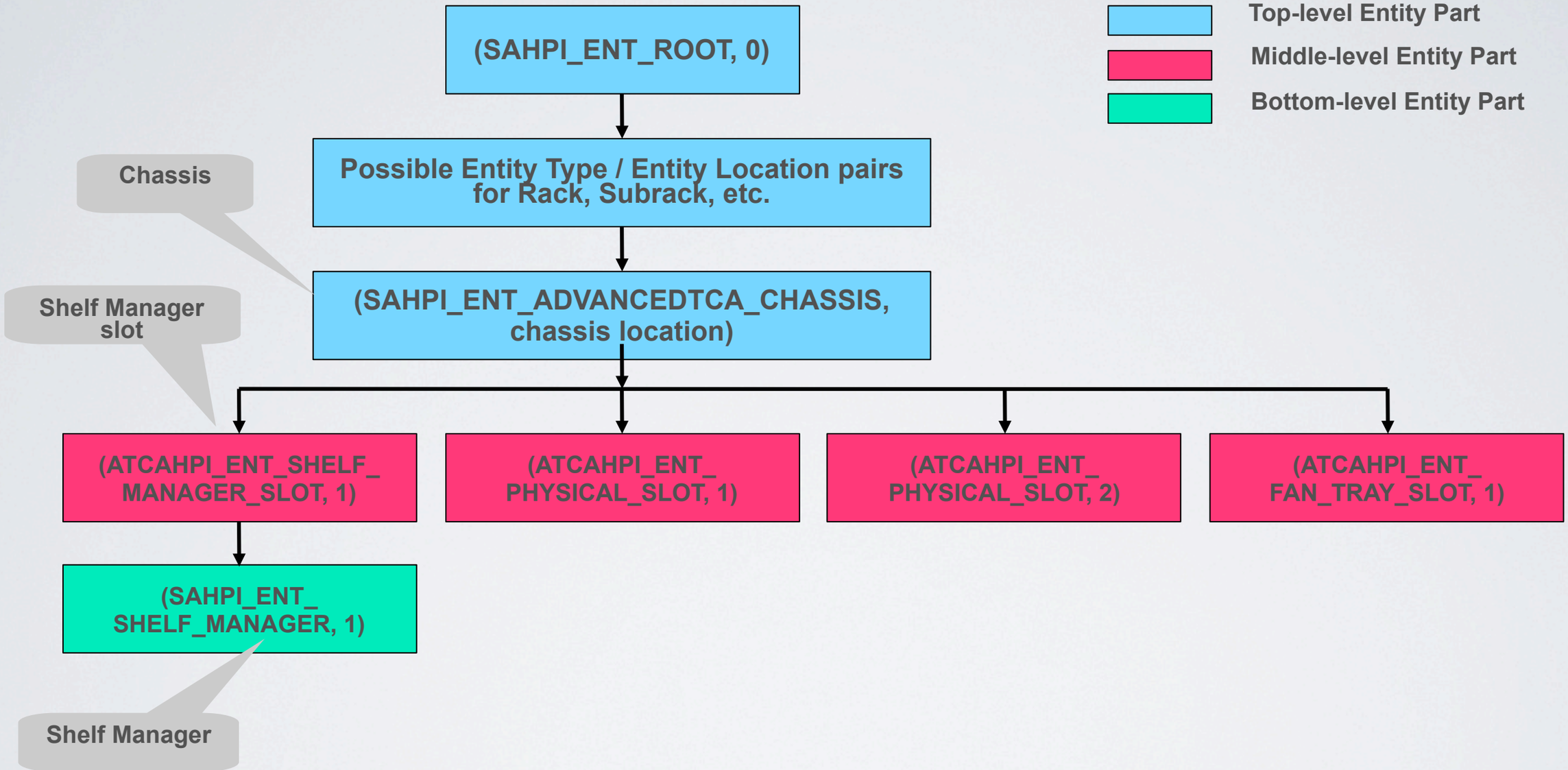
Copyright© 2006 Service Availability™ Forum, Inc

EXAMPLE ENTITY PATH TREE



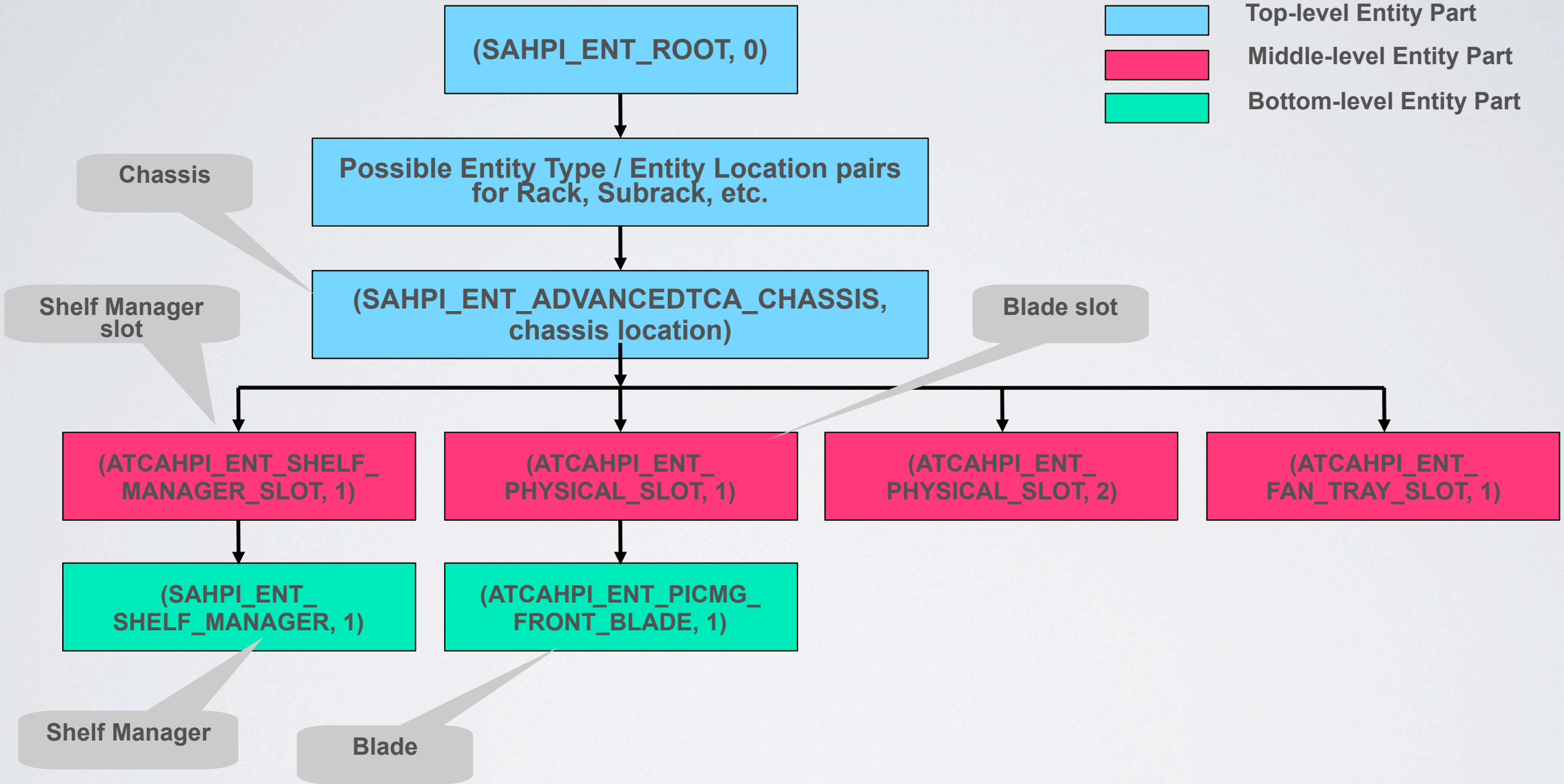
Copyright© 2006 Service Availability™ Forum, Inc

EXAMPLE ENTITY PATH TREE



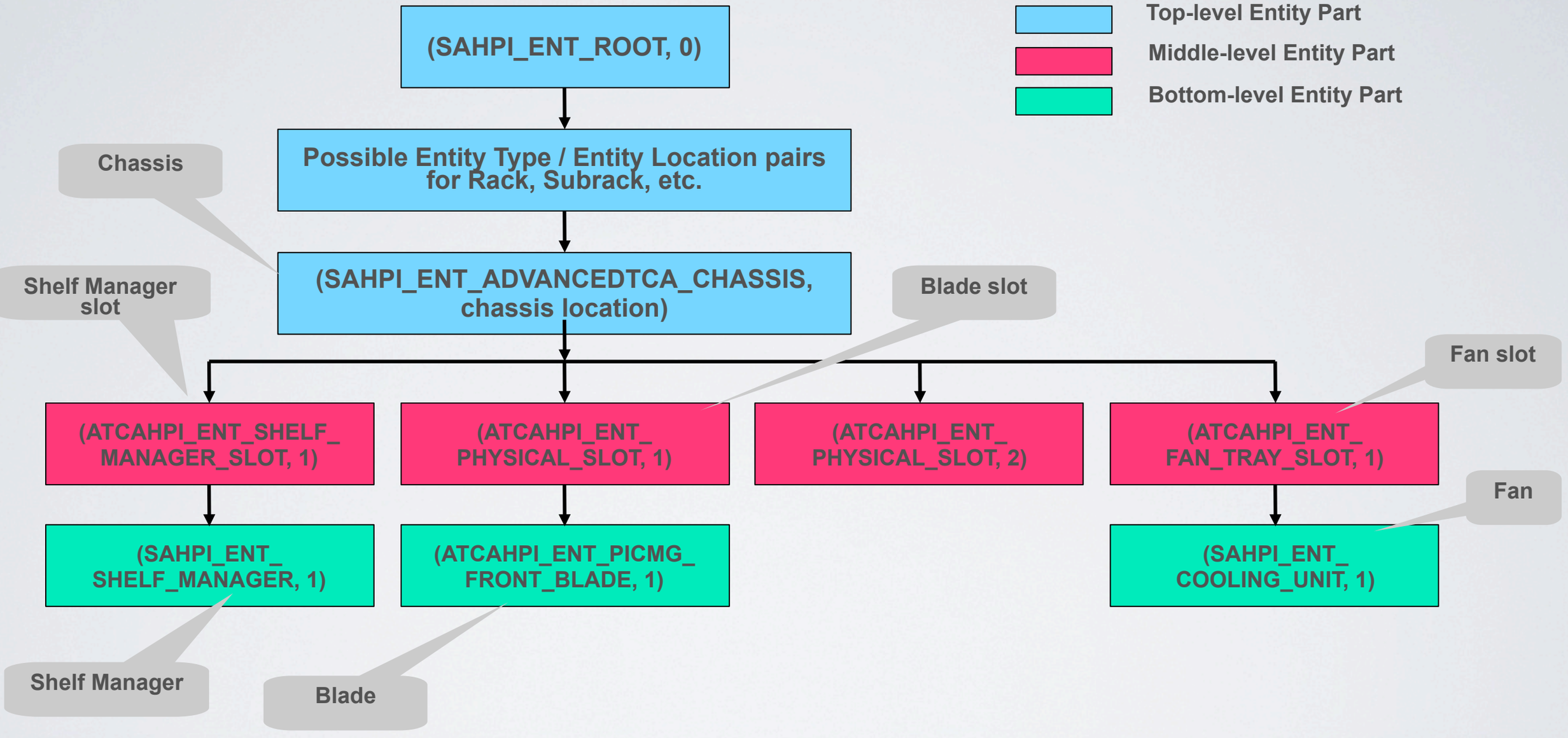
Copyright© 2006 Service Availability™ Forum, Inc

EXAMPLE ENTITY PATH TREE



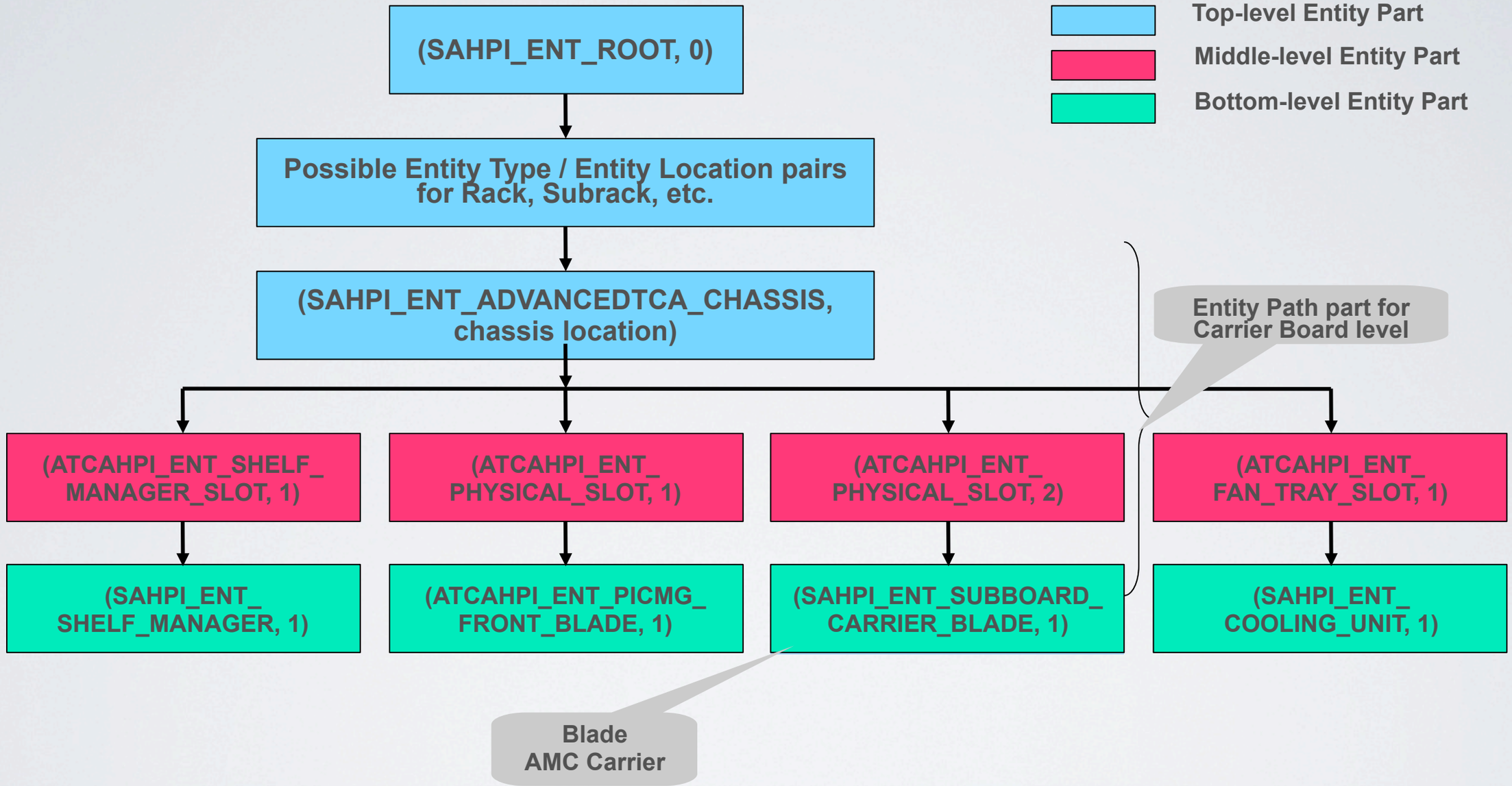
Copyright© 2006 Service Availability™ Forum, Inc

EXAMPLE ENTITY PATH TREE



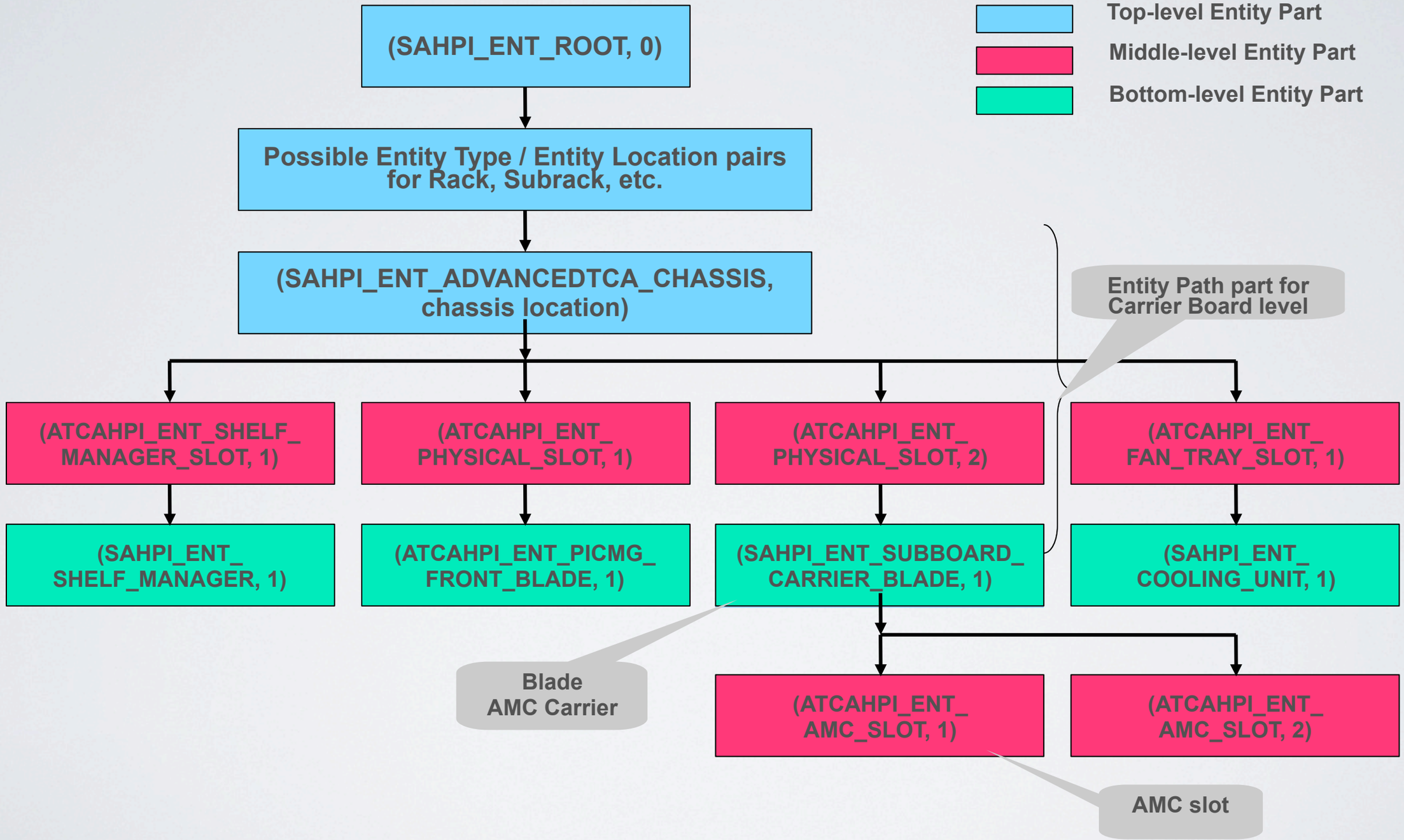
Copyright© 2006 Service Availability™ Forum, Inc

EXAMPLE ENTITY PATH TREE



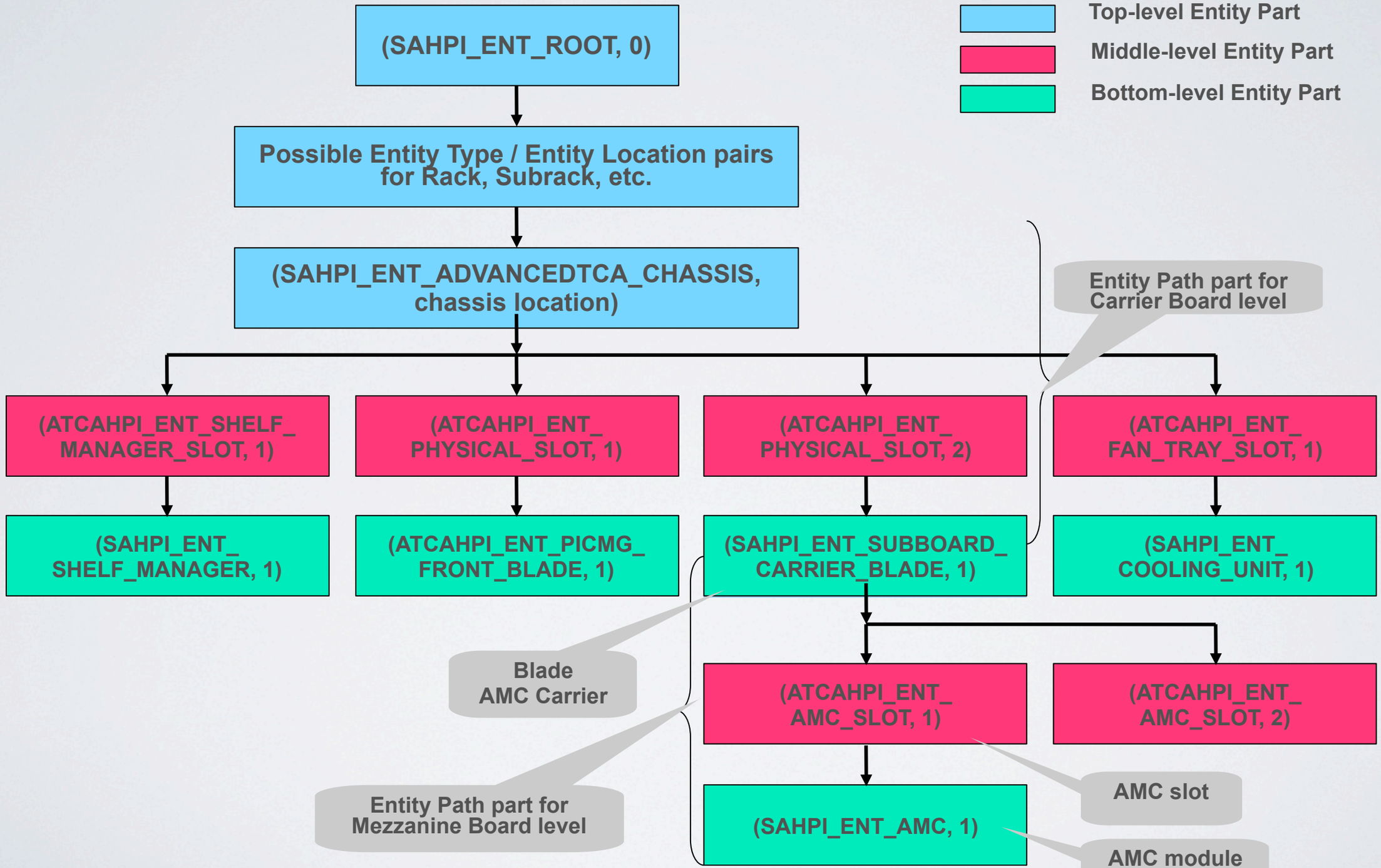
Copyright© 2006 Service Availability™ Forum, Inc

EXAMPLE ENTITY PATH TREE



Copyright© 2006 Service Availability™ Forum, Inc

EXAMPLE ENTITY PATH TREE



Copyright© 2006 Service Availability™ Forum, Inc

EXAMPLE ENTITY PATH IN ATCA

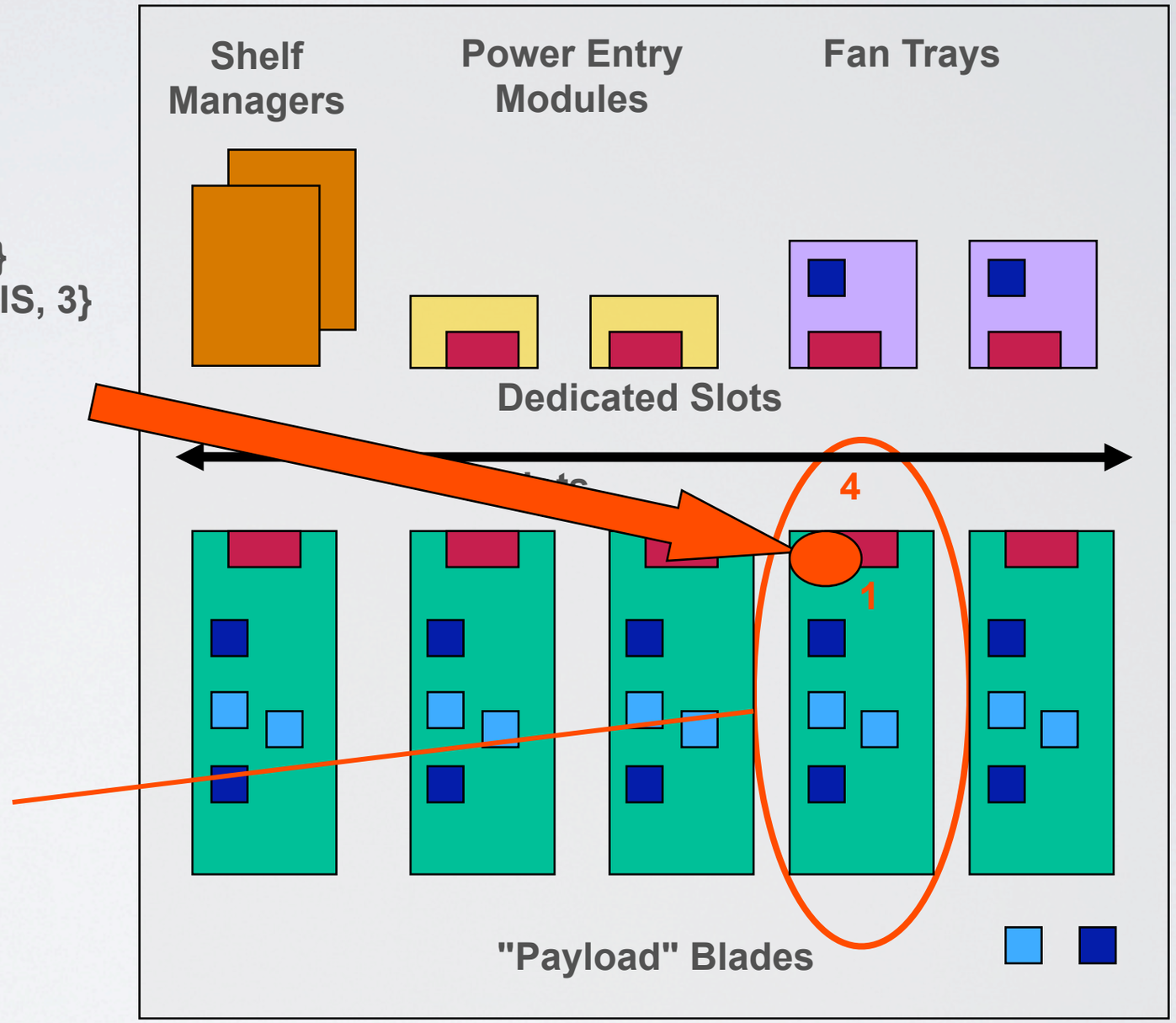
Blade 1 in slot 4 in chassis 3:

{
{SAHPI_ENT_SBC_BLADE, 1}
{SAHPI_ENT_PHYSICAL_SLOT, 4}
{SAHPI_ENT_ADVANCEDTCA_CHASSIS, 3}
{SAHPI_ENT_ROOT, 0}
}

Entity Type **Entity Location**

"Address" of this resource

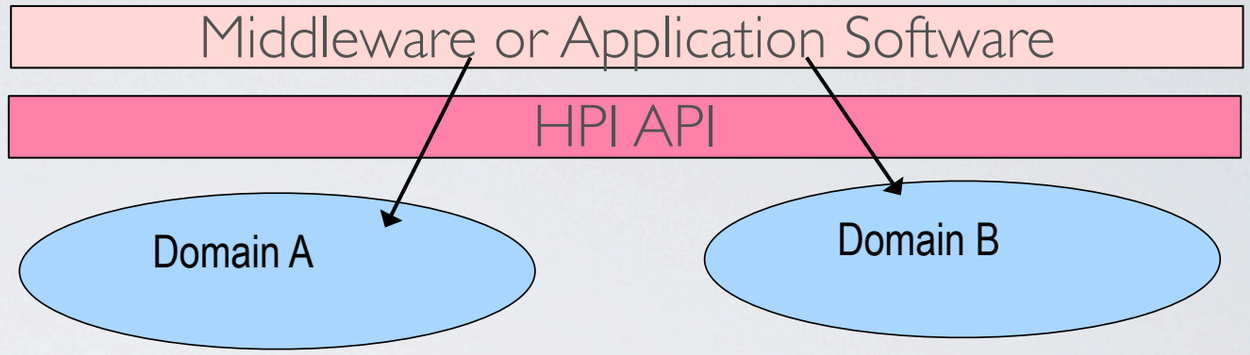
All its Management Instruments share the same "address"



Copyright© 2006 Service Availability™ Forum, Inc

HARDWARE PLATFORM INTERFACE

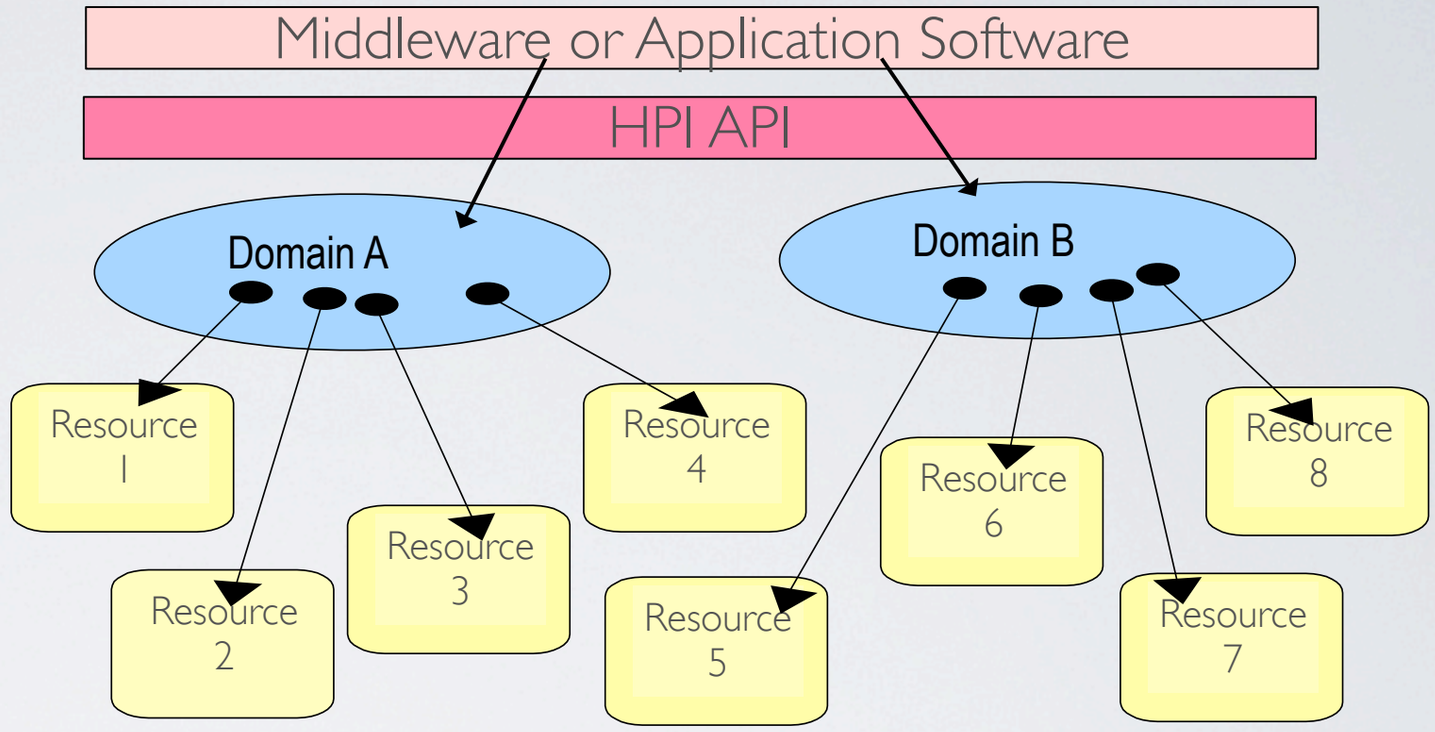
- Users open **sessions** for **domains** via HPI library functions



Copyright© 2006 Service Availability™ Forum, Inc

HARDWARE PLATFORM INTERFACE

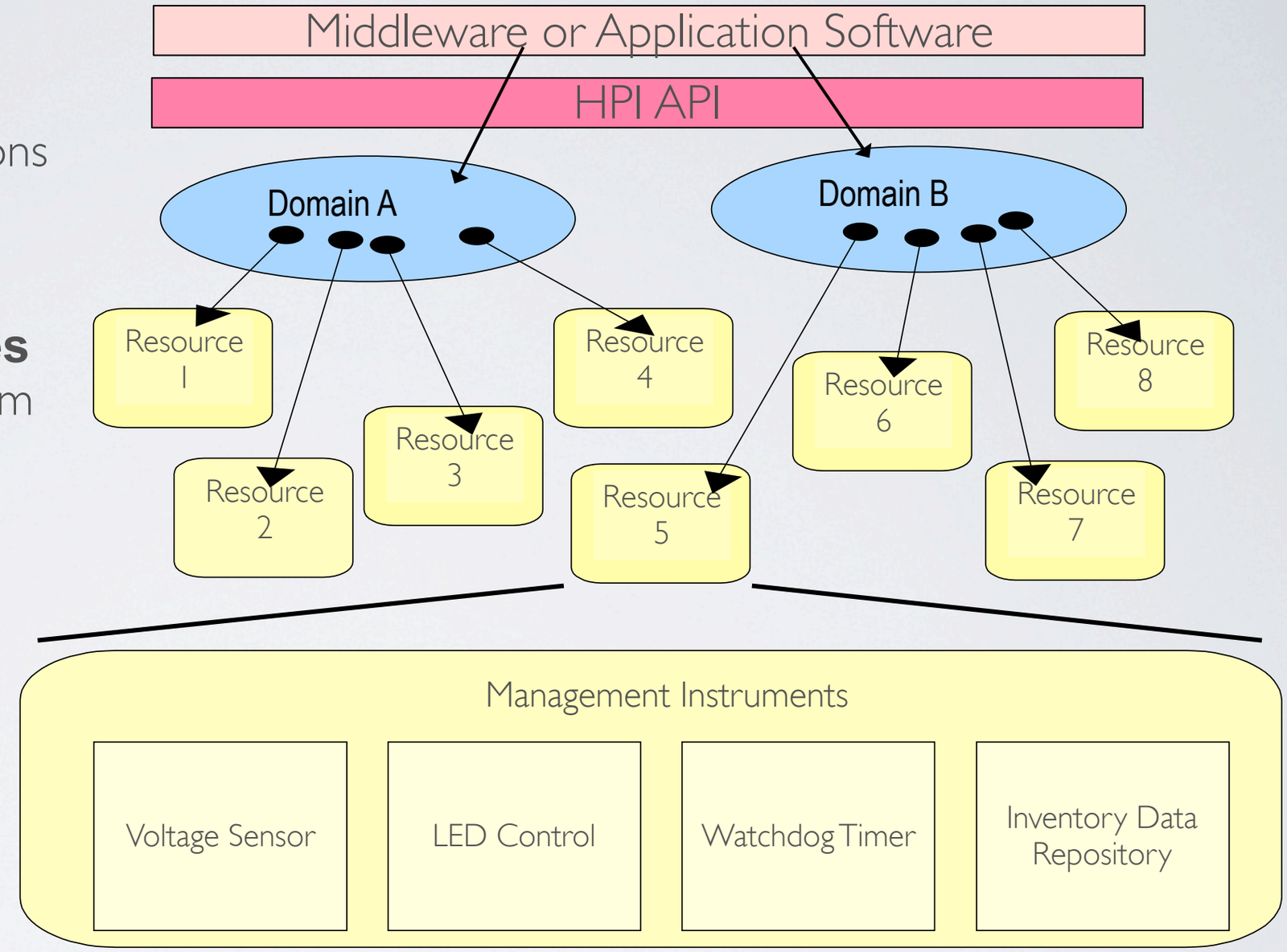
- Users open **sessions** for **domains** via HPI library functions
- **Domains** contain **resources** representing parts of the platform management infrastructure



Copyright© 2006 Service Availability™ Forum, Inc

HARDWARE PLATFORM INTERFACE

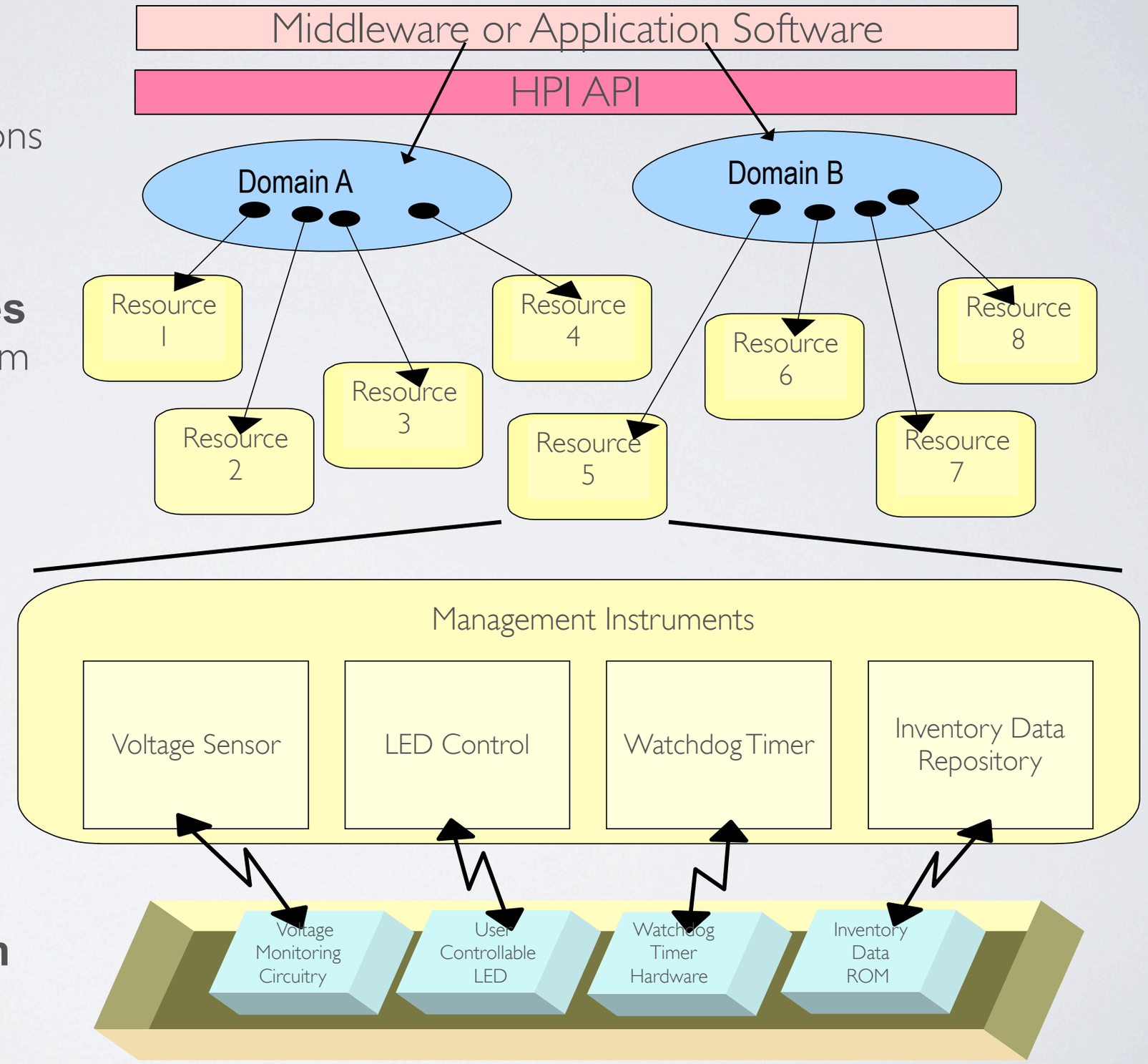
- Users open **sessions** for **domains** via HPI library functions
- **Domains** contain **resources** representing parts of the platform management infrastructure
- **Resources** have **management capabilities** accessible by users



Copyright© 2006 Service Availability™ Forum, Inc

HARDWARE PLATFORM INTERFACE

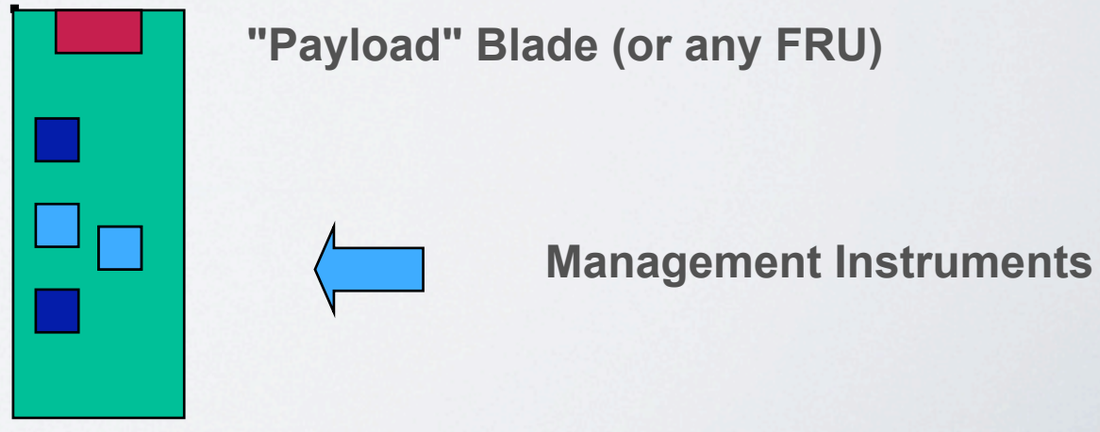
- Users open **sessions** for **domains** via HPI library functions
- **Domains** contain **resources** representing parts of the platform management infrastructure
- **Resources** have **management capabilities** accessible by users
- **Resource management capabilities** map to **platform management hardware**



Copyright© 2006 Service Availability™ Forum, Inc

GENERALIZED MAPPING

- A **HPI Resource** is a logical representation of a physical entity within an AdvancedTCA platform, such as:
  - AdvancedTCA FRU
  - Shelf
  - Slot within a Shelf
  - FRU in a Shelf
- Each **HPI Domain** corresponding to a ATCA Shelf has a **HPI Resource Presence Table (RPT)**, which contains a Resource for each entity in the Shelf
- Each **HPI Resource** has an associated **HPI Resource Data Record (RDR) Repository**, which provides access to its **HPI Management Instruments**:
  - Sensors
  - Controls
  - Inventory Data Repositories
  - Watchdog Timers
  - Annunciators



Copyright© 2006 Service Availability™ Forum, Inc

HPI DATA STORES

- **Domain Reference Table (DRT)**
- **Resource Presence Table (RPT)**
- **Resource Data Record (RDR) Repository**
- **Inventory Data Repository (IDR)**
- **Domain and Resource Event Logs**
- **Domain Alarm Table (DAT)**

Copyright© 2006 Service Availability™ Forum, Inc

DOMAIN REFERENCE TABLE

- The Domain Reference Table (DRT) for a domain provides information about other domains associated with the given domain, including
  - Peer domains in a peer architecture
  - Child domains in a tier architecture
- There is one DRT per domain
- Each associated domain has a DRT entry in the DRT for the given domain
- The DRT is built and maintained by the HPI implementation
  - The DRT is dynamically updated as the system configuration changes. An event is generated when a domain is added to/ removed from the DRT

Copyright© 2006 Service Availability™ Forum, Inc

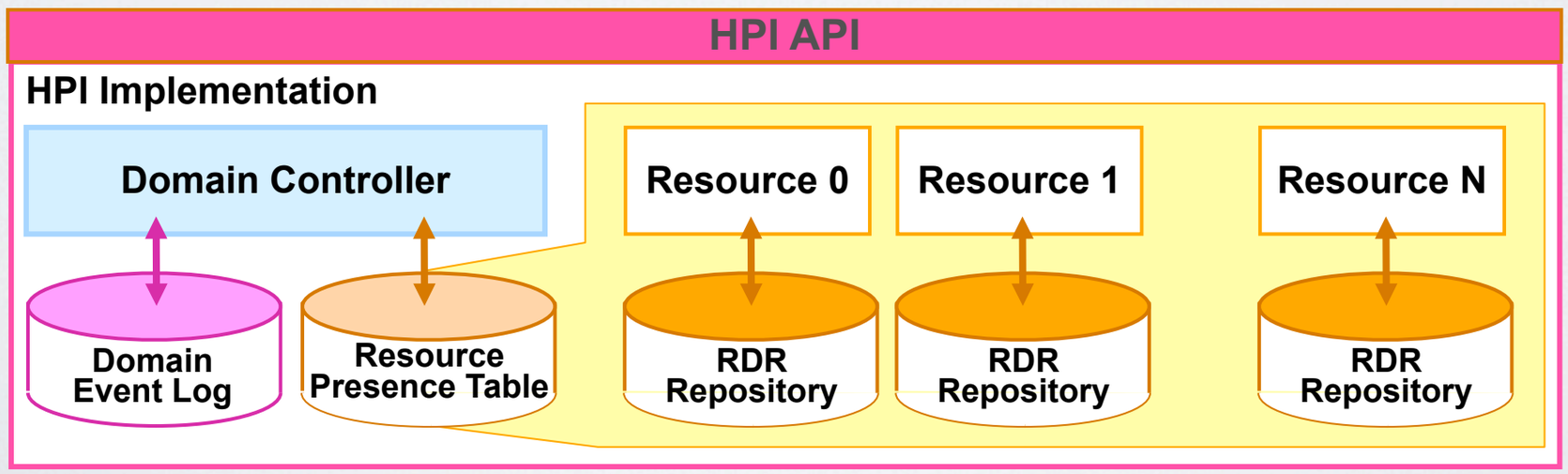
RESOURCE PRESENCE TABLE

- The Resource Presence Table (RPT) reflects the current resources in a domain
- There is one RPT per domain
- Each RPT entry represents a resource
  - A RPT entry contains vendor product data to identify the resource
    - IDs, versions, etc.
  - A RPT entry defines standard capabilities of the resource
    - Managed hot swap, watchdog timer, event log, etc.
- The RPT is built and maintained by the HPI implementation
  - The RPT is dynamically updated as resources are added to/removed from system, as in Hot Swap

Copyright© 2006 Service Availability™ Forum, Inc

RESOURCE DATA RECORD REPOSITORY

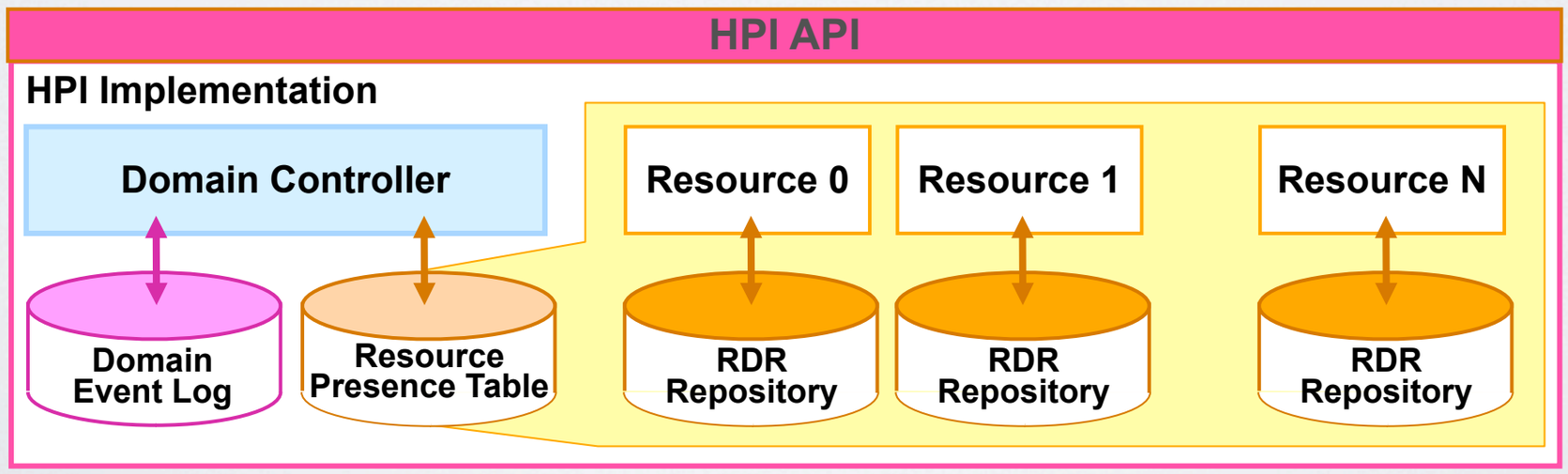
- The Resource Data Record (RDR) Repository is a database containing RDRs
- The RDRs provide static information describing the various management instruments, namely
  - Sensors
  - Controls
  - Watchdog Timers
  - Inventory Data Repositories
  - Annunciators
- There is one RDR Repository for each RPT entry



Copyright© 2006 Service Availability™ Forum, Inc

RESOURCE DATA RECORD REPOSITORY

- The HPI Resource APIs access the Resource Data Record (RDR) Repository
  - Each resource that contains a management instrument (sensor, control, watchdog timer, inventory data repository, or annunciator) must have an associated RDR Repository
  - Because most resources contain management instruments, most will contain a RDR Repository
- The HPI model uses a distributed RDR Repository, where each resource maintains its own local RDR Repository
  - A local RDR Repository is a database containing records that describe management instruments
  - The records in the various local RDR Repositories have common fields for record type and naming information
- A local RDR Repository holds information for all entities that are managed by the resource
  - All sensors, controls, watchdogs, Inventory Data Repositories, and annunciators present in a resource must be specified in the local RDR Repository



Copyright© 2006 Service Availability™ Forum, Inc

INVENTORY DATA REPOSITORY

- The Inventory Data Repository (IDR) is a collection of Inventory Data Areas
  - An Inventory Data Area is a category of inventory data (manufacturer name, product name, product version, model number, serial number, part number, asset tag, etc)
- A resource that has the Inventory Data capability set in its RPT entry contains at least one IDR
- The IDR contains information that relates to a single entity
- A IDR RDR in the RDR Repository can be used to locate an available Inventory Data Repository

Copyright© 2006 Service Availability™ Forum, Inc

EVENT LOGS

- The HPI supports two kinds of event logs
  - Domain Event Log – Maintained for events collected in the domain by the domain controller
  - Resource Event Log – Maintained by an individual resource
- Exactly which events are placed in these logs, and how long the events remain there, is implementation-specific
- An Event Log entry contains two timestamps:
  - Event Log Timestamp - Time that event is placed in the Event Log
  - Event Timestamp - Approximation to when event actually occurred
- Event Log entries are ordered chronologically by Event Log Timestamp

Copyright© 2006 Service Availability™ Forum, Inc

DOMAIN ALARM TABLE

- The Domain Alarm Table (DAT) contains an entry for each alarm in the domain
  - A HPI implementation adds/deletes alarms to/from the DAT as it detects the presence/absence of corresponding conditions
  - A HPI user may also add/delete alarms to/from the DAT to reflect user-detected conditions
- The DAT contains an entry corresponding to each of the following conditions within the domain or one of its resources
  - Significant asserted sensor event state
  - Significant resource failure
  - OEM alarm condition
  - User-defined alarm condition

Copyright© 2006 Service Availability™ Forum, Inc

HPI API

- The HPI defines 79 function calls that are logically grouped into 16 sets
  - Implementation global functions
  - Domain functions
    - Sessions
    - Discovery
    - Events and Event Logs
    - Alarms
  - Resource functions
    - Discovery
    - 10 sets of functions for the management capabilities that might be contained in a resource

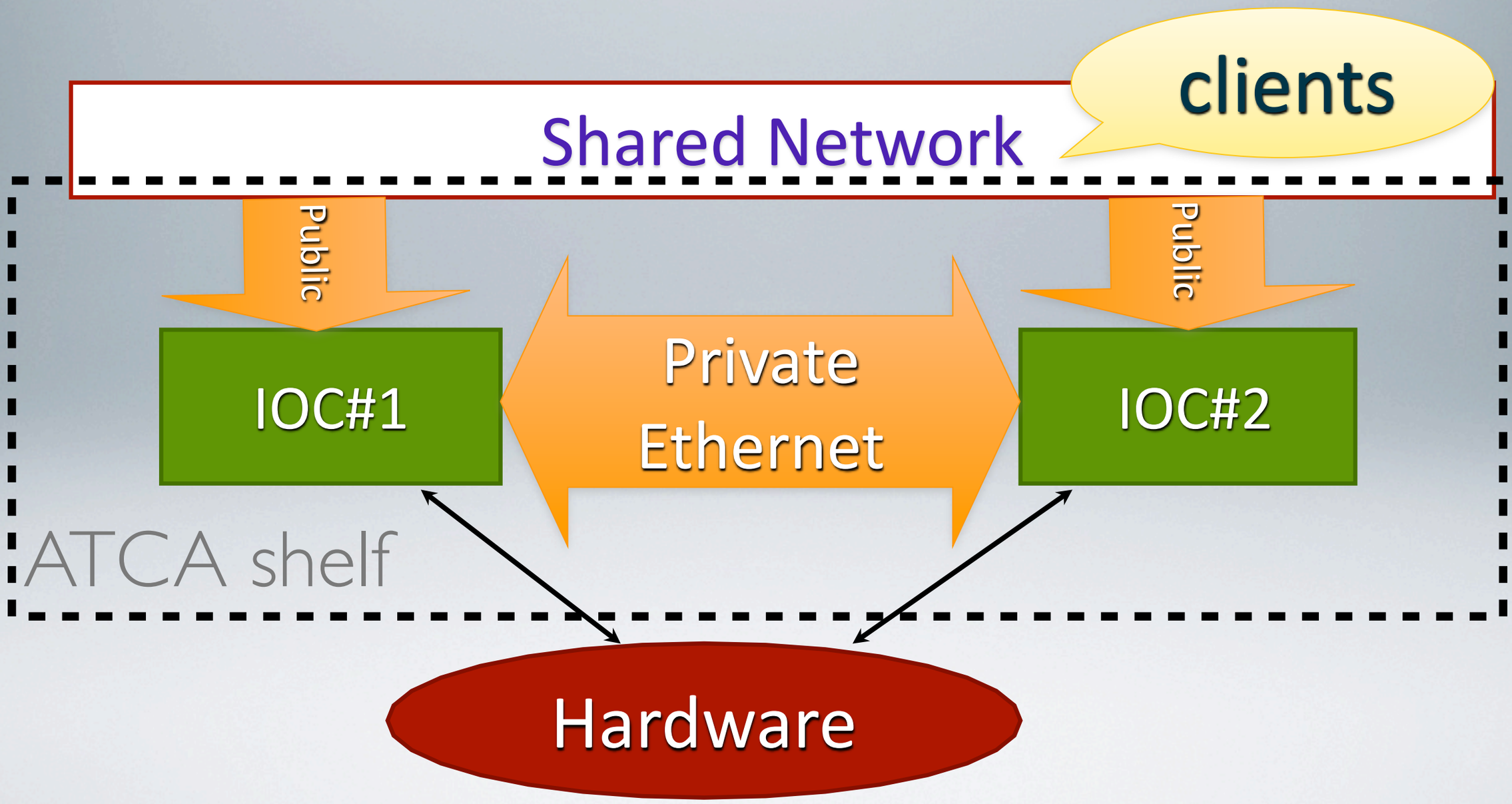
Copyright© 2006 Service Availability™ Forum, Inc

HPI USAGE EXAMPLE: REDUNDANT IOC ACCELERATOR CONTROL SYSTEM

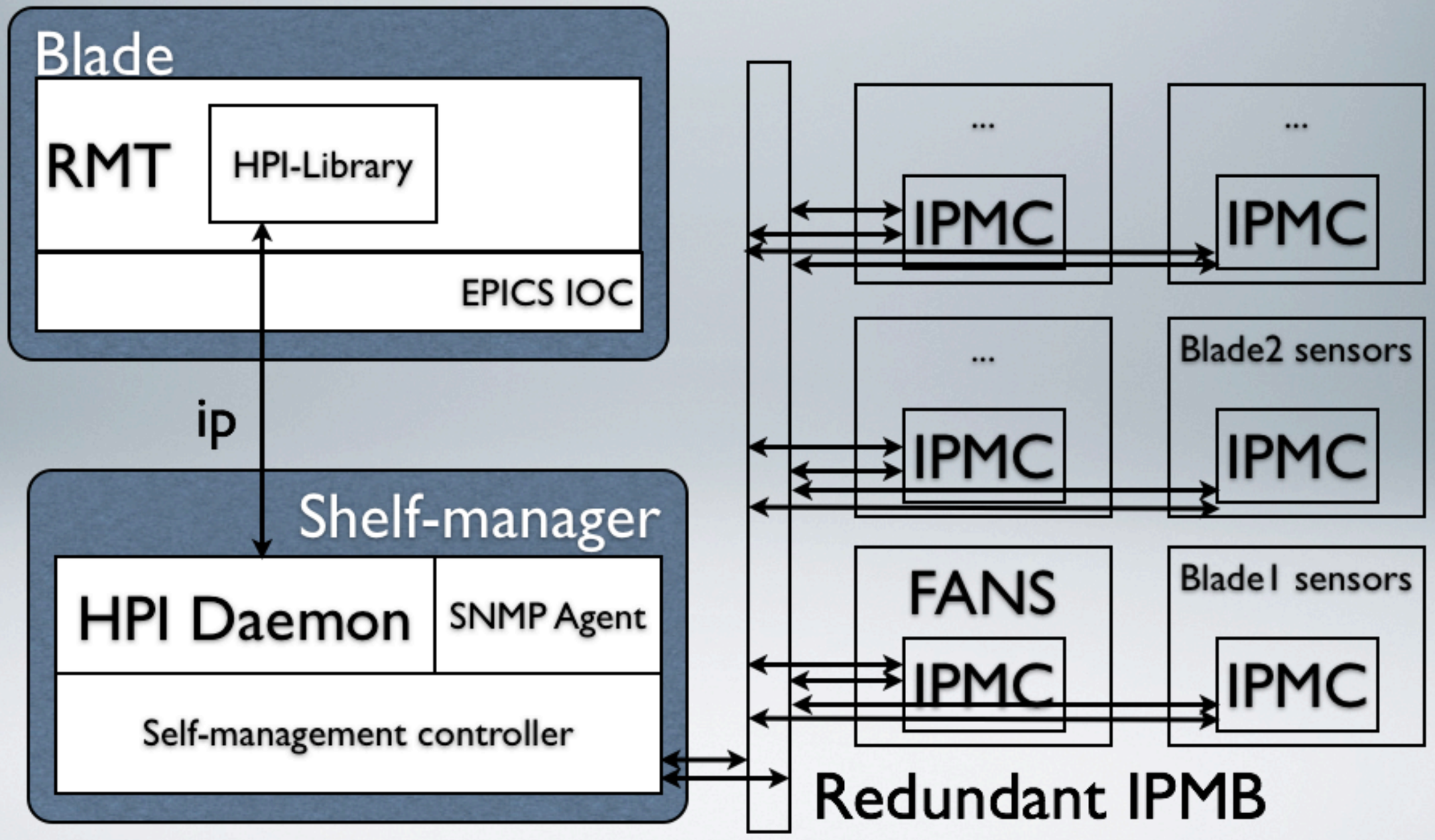
- **Fully redundant ATCA Shelf**
- **2 CPU boards**
- **Each running EPICS Redundant IOC**
- **HPI implementation from the vendor**

- **Red Hat ES 5 Linux Operating System**
- EPICS - Experimental Physics and Industrial Control System
- IOC - Input Output Controller

REDUNDANT EPICS IOC ARCHITECTURE

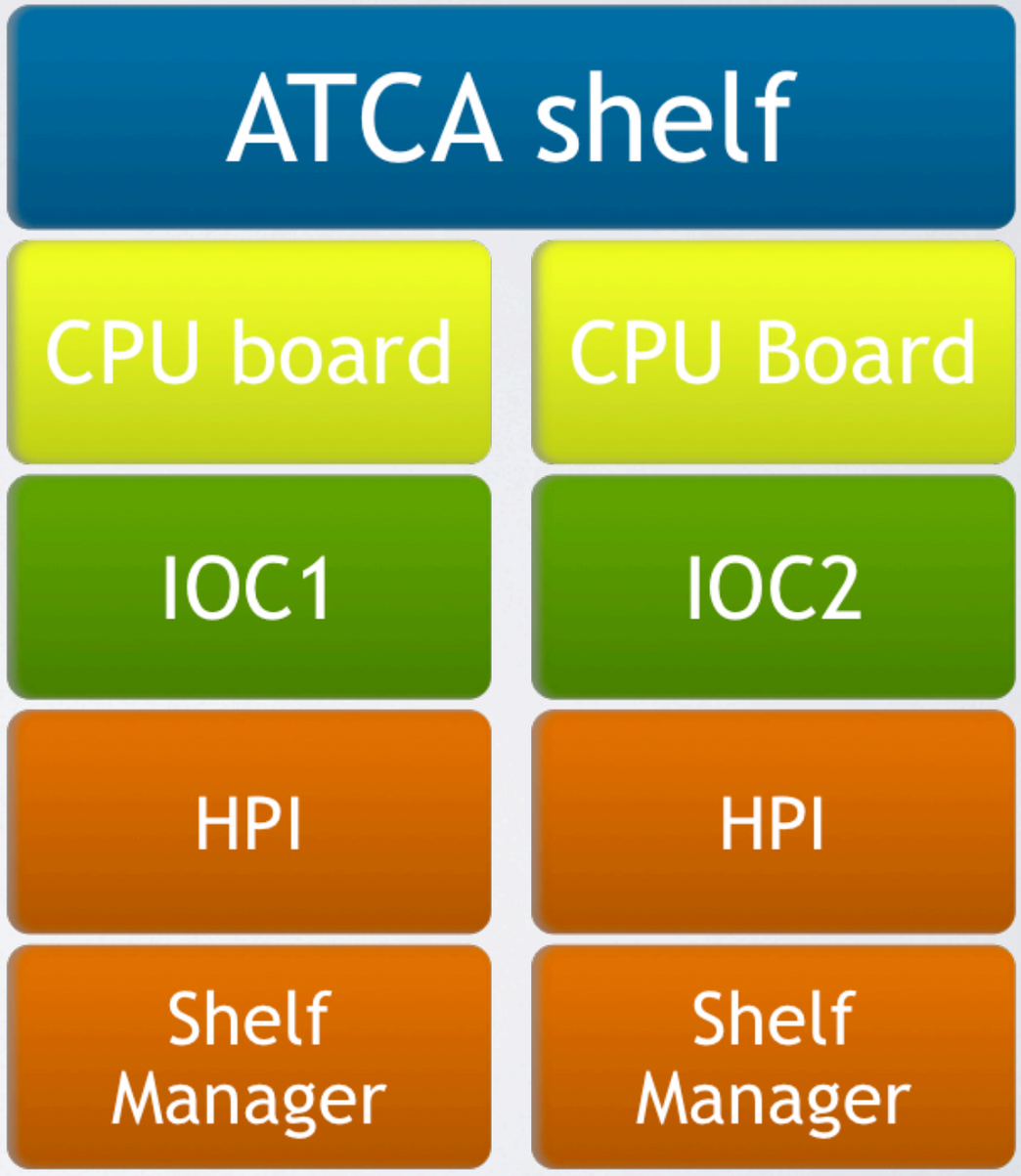


REDUNDANT EPICS IOC ON ATCA



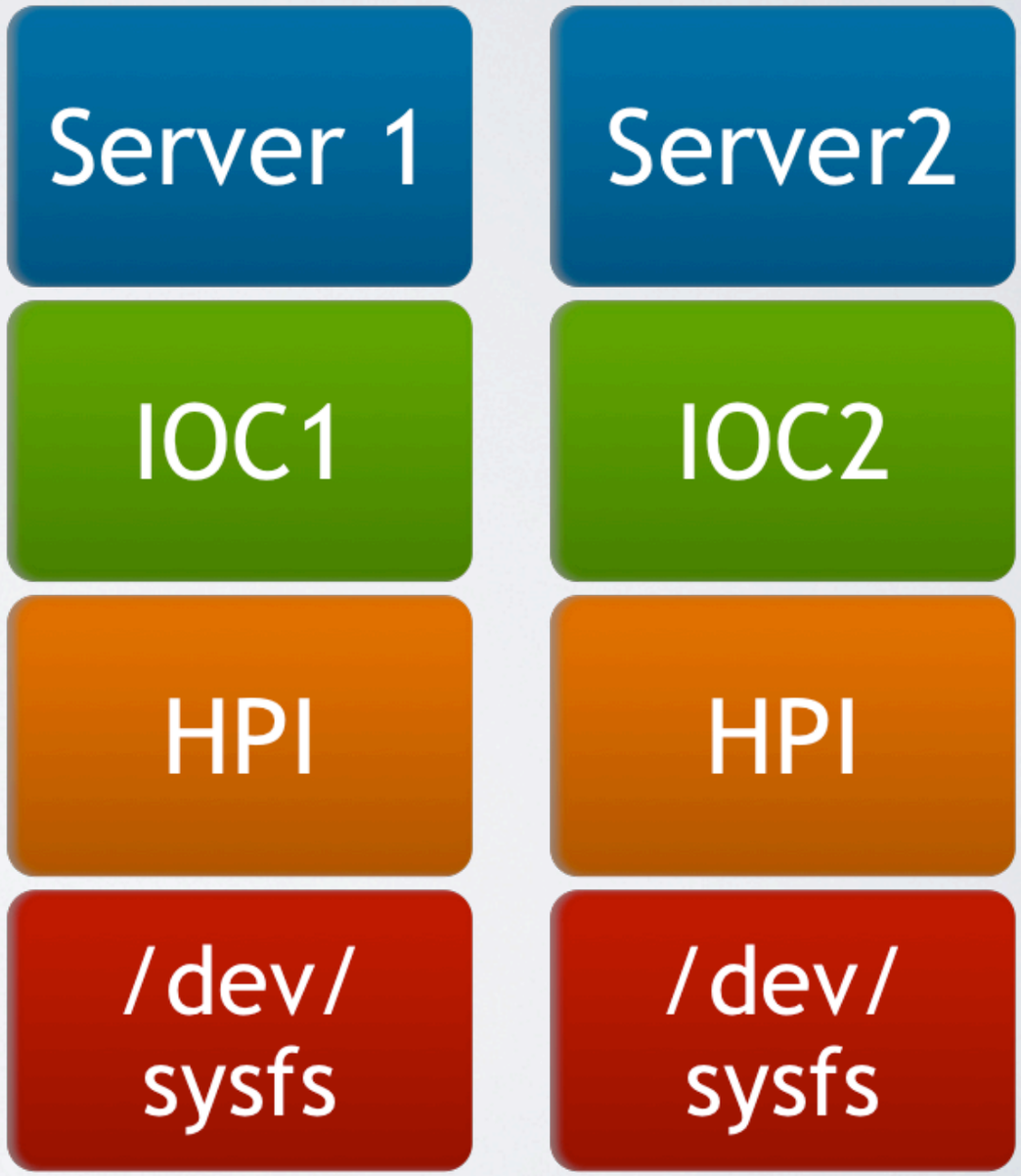
HPI USAGE EXAMPLE – REDUNDANT EPICS IOC

- **HPI is used to monitor the health of each blade and the shelf**
- **This information is used to make decision on failover**



HPI USAGE EXAMPLE – REDUNDANT EPICS IOC

- **HPI is Platform independent**
- **Instead of ATCA we can use "conventional" server PC**
- **OpenHPI has /dev/ sysfs mappings on Linux**



HPI USAGE EXAMPLE – REDUNDANT EPICS IOC

- **HPI allowed us to create a portable implementation at no cost**
- **HPI allowed to avoid “low-level” interfaces**
  - Just changing the configuration file we can access all the possible sensors on all available hardware in the shelf
  - Even the new ones

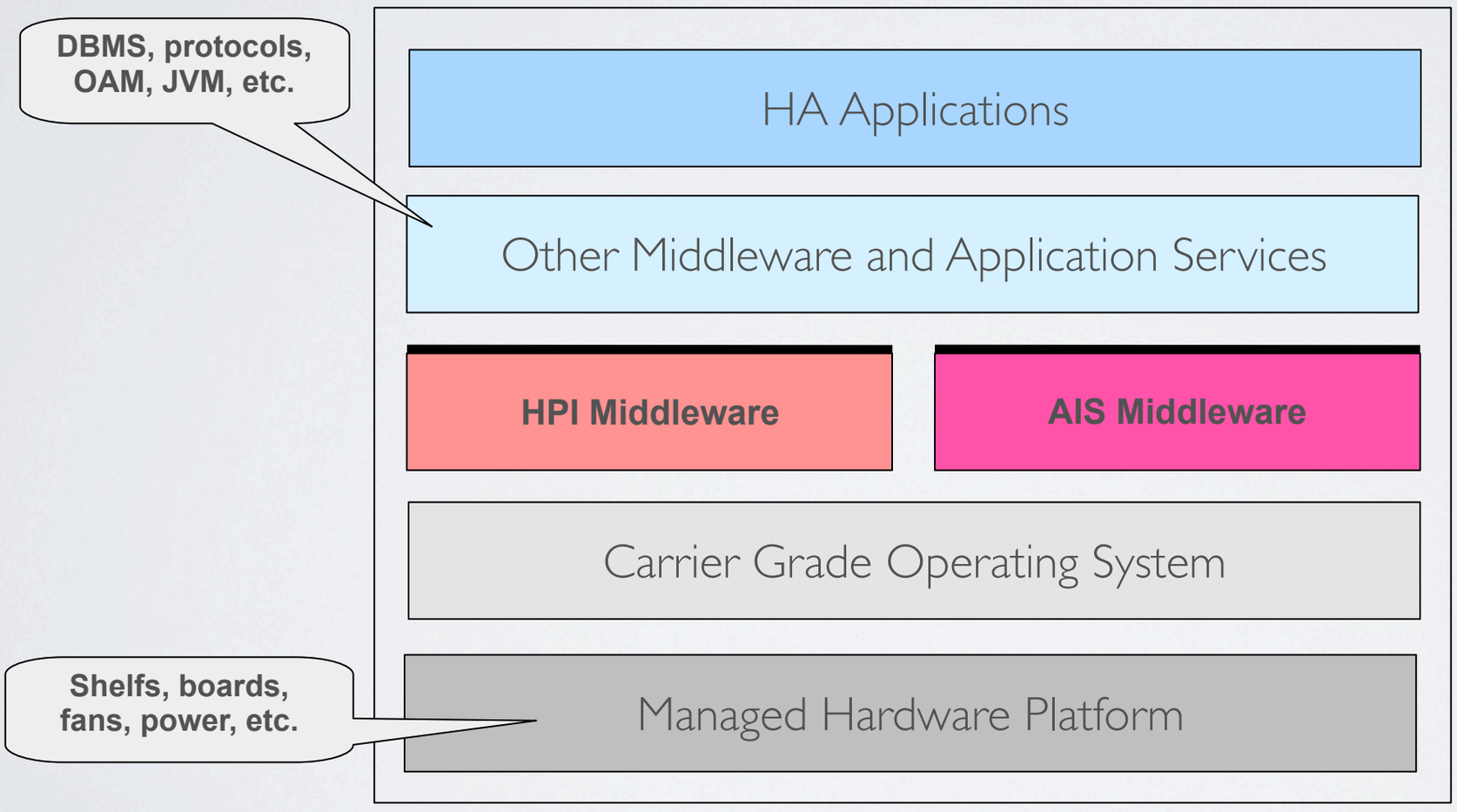
APPLICATION INTERFACE SPECIFICATION OVERVIEW

APPLICATION INTERFACE SPECIFICATION

- The Application Interface Specification (AIS) is a set of open standard interface specifications
- The AIS defines an Application Program Interface (API) for middleware between the applications and the operating system
- The AIS is divided into the following parts or areas:
  - Availability Management Framework
  - Cluster Membership Service
  - Checkpoint Service
  - Event Service
  - Message Service
  - Lock Service
  - Notification Service
  - Log Service
  - Information Model Management Service

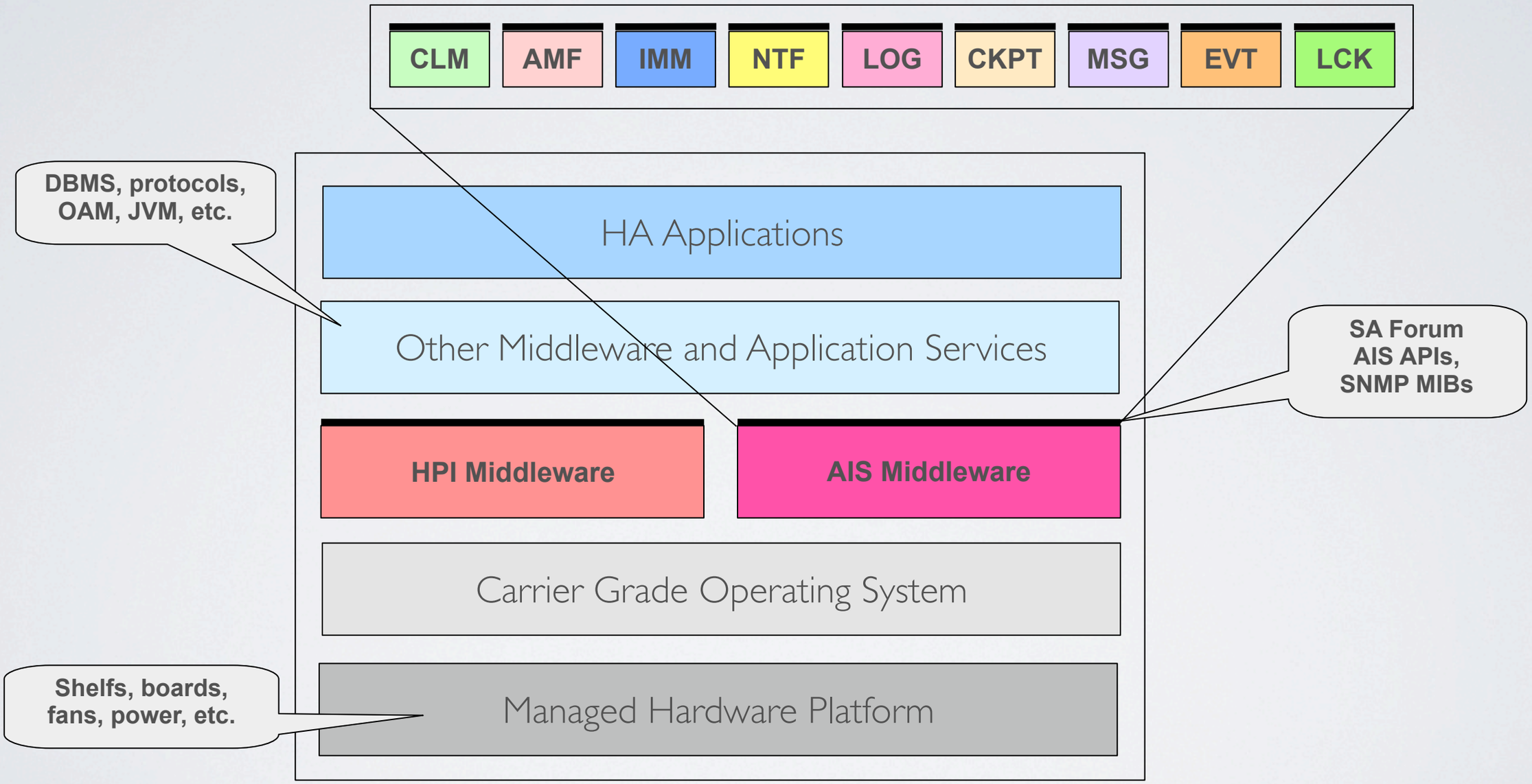
Copyright© 2006 Service Availability™ Forum, Inc

APPLICATION INTERFACE SPECIFICATION



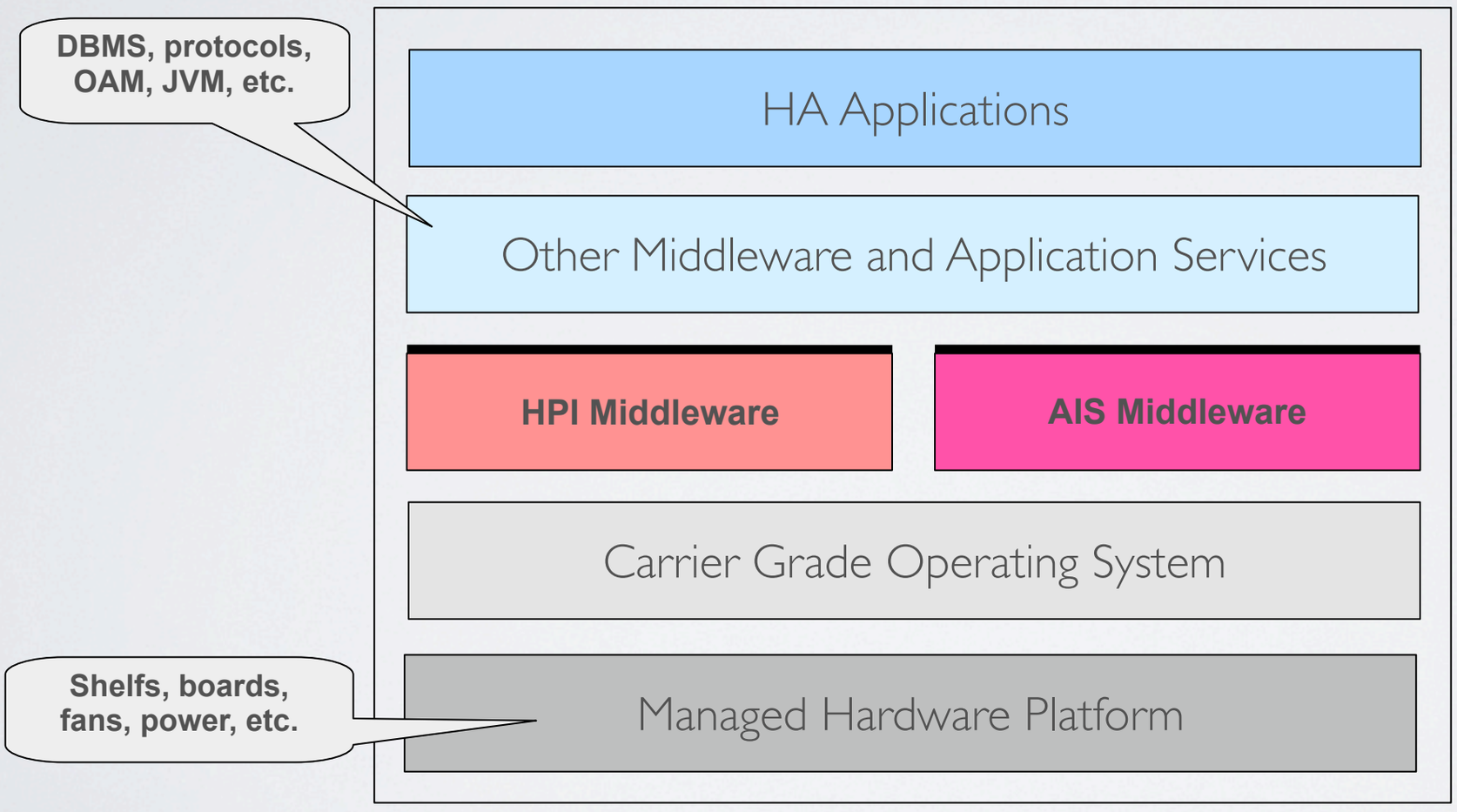
Copyright© 2006 Service Availability™ Forum, Inc

APPLICATION INTERFACE SPECIFICATION



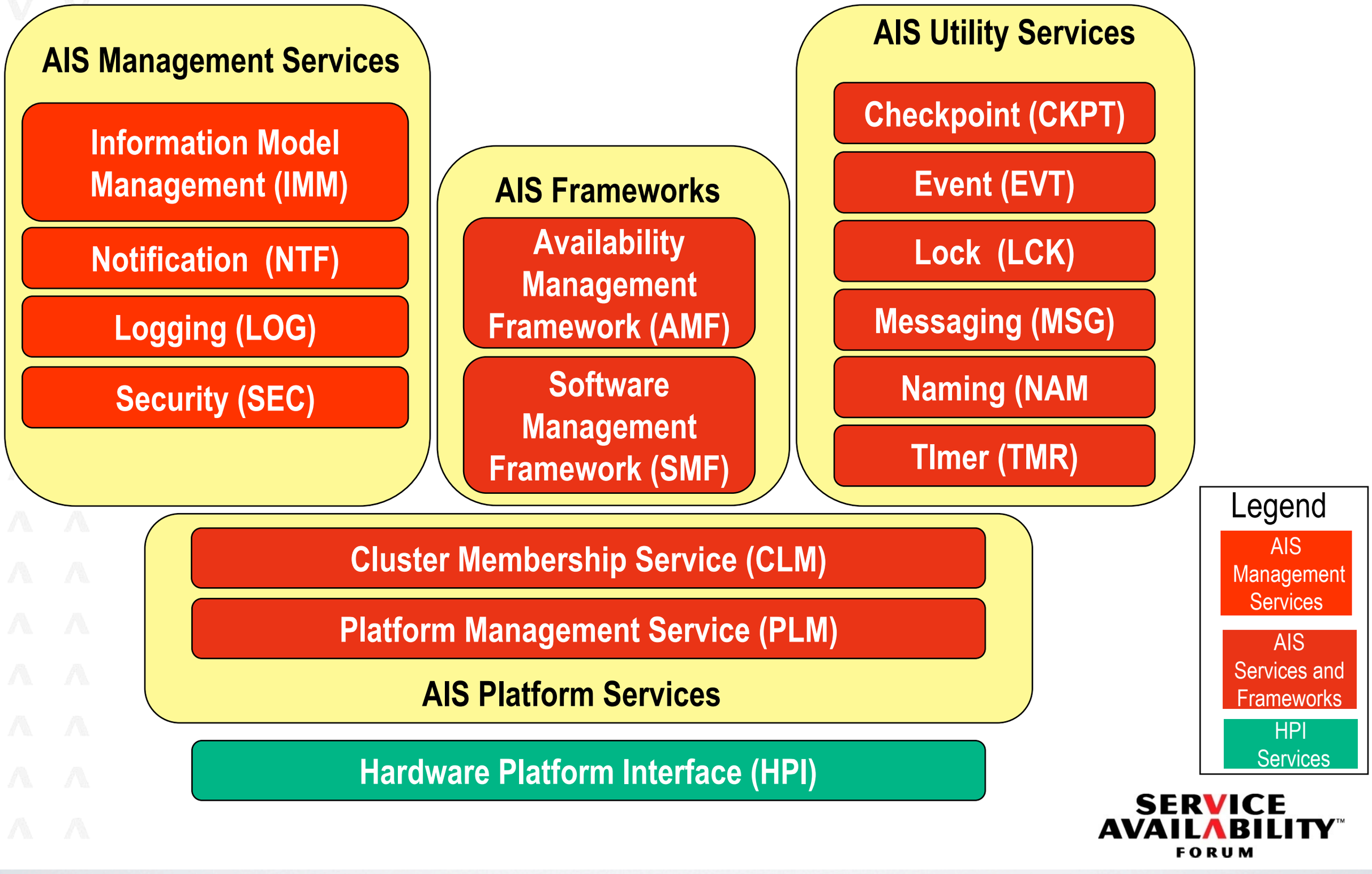
Copyright© 2006 Service Availability™ Forum, Inc

APPLICATION INTERFACE SPECIFICATION



Copyright© 2006 Service Availability™ Forum, Inc

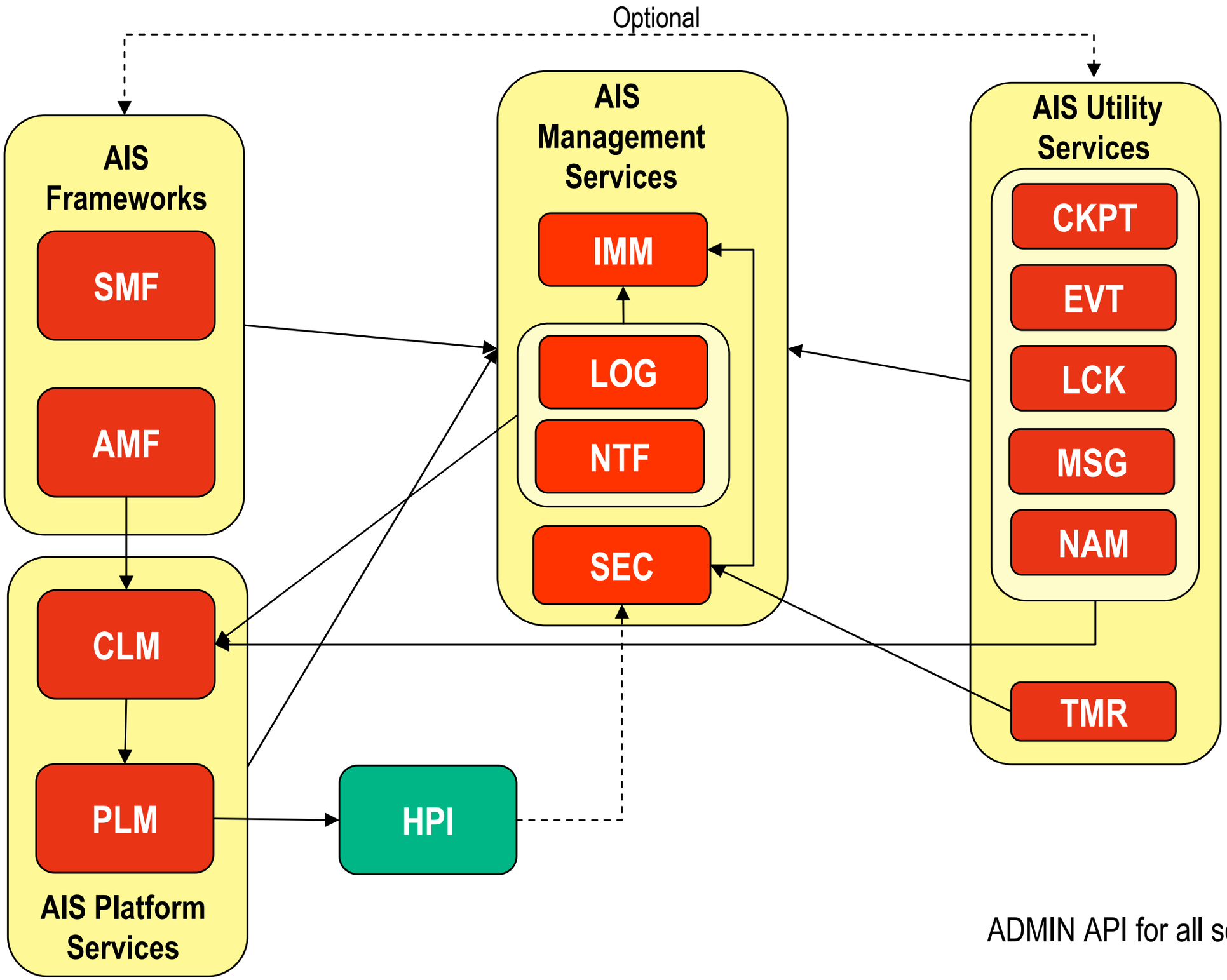
SA Forum Architecture Box diagram



SERVICE AVAILABILITY™ FORUM

Courtesy of J. Fryer

Typical SA Forum Service Dependencies



ADMIN API for all services through IMM

SERVICE AVAILABILITY™ FORUM

Courtesy of J. Fryer

APPLICATION INTERFACE SPECIFICATION

- Each of the AIS Services consists of a client library and a server
- The client library for an AIS Service is linked into each application process that utilizes the service
- No assumptions are made about how the server is implemented, or how the server instances are distributed across the nodes of the cluster
- An implementation of an AIS Service might have
  - A single server in the cluster,
  - A server on each node, or
  - A server within each copy of the library

Copyright© 2006 Service Availability™ Forum, Inc

AVAILABILITY MANAGEMENT FRAMEWORK

- The Availability Management Framework (AMF), sometimes simply called the Framework, is the core of the Application Interface Specification
- The Framework provides service availability by coordinating redundant resources within a cluster to deliver a system with no single point of failure
- The Framework
  - Provides a view of one logical cluster that consists of a number of cluster nodes
  - Coordinates redundant resources within the cluster
  - Monitors the health of components and determines the states of components

Copyright© 2006 Service Availability™ Forum, Inc

AVAILABILITY MANAGEMENT FRAMEWORK

- The Availability Management Framework (AMF) is based on the following logical entities:
  - A component is a logical entity that represents a set of resources to the Availability Management Framework
  - The Availability Management Framework interacts with components
  - Components that are collocated on a node are managed as a service unit which is local to the node
  - Service units are units of redundancy from a deployment perspective
  - A service group consists of one or more identical service units that participate in a redundancy model
  - A service instance is workload assigned to the service unit
  - A service instance is made up of component service instances that correspond to the workload assigned to the components in the service unit
  - An application is a logical entity that contains one or more service groups

Copyright© 2006 Service Availability™ Forum, Inc

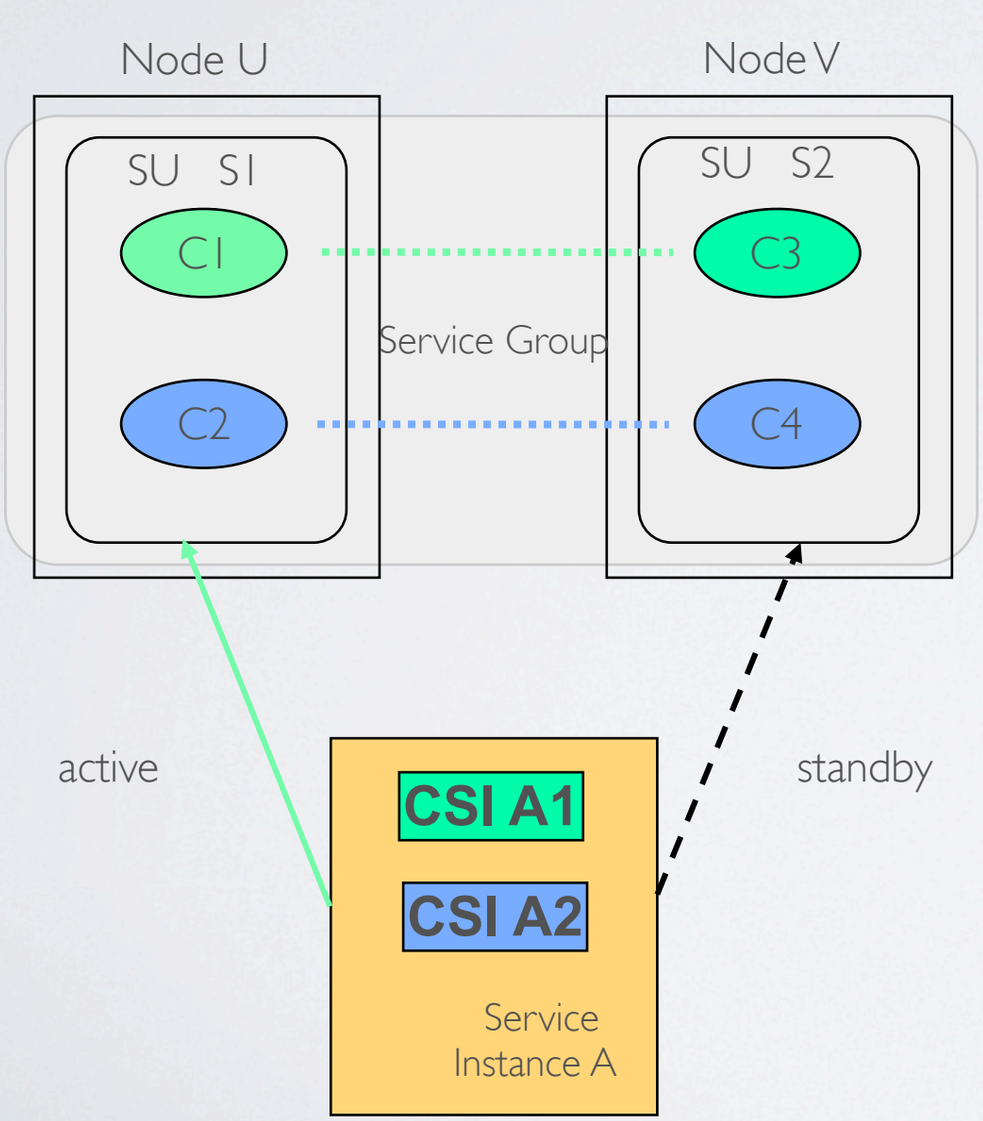
AVAILABILITY MANAGEMENT FRAMEWORK

- Example of the 2N redundancy model (with two service units on different nodes)

Copyright© 2006 Service Availability™ Forum, Inc

AVAILABILITY MANAGEMENT FRAMEWORK

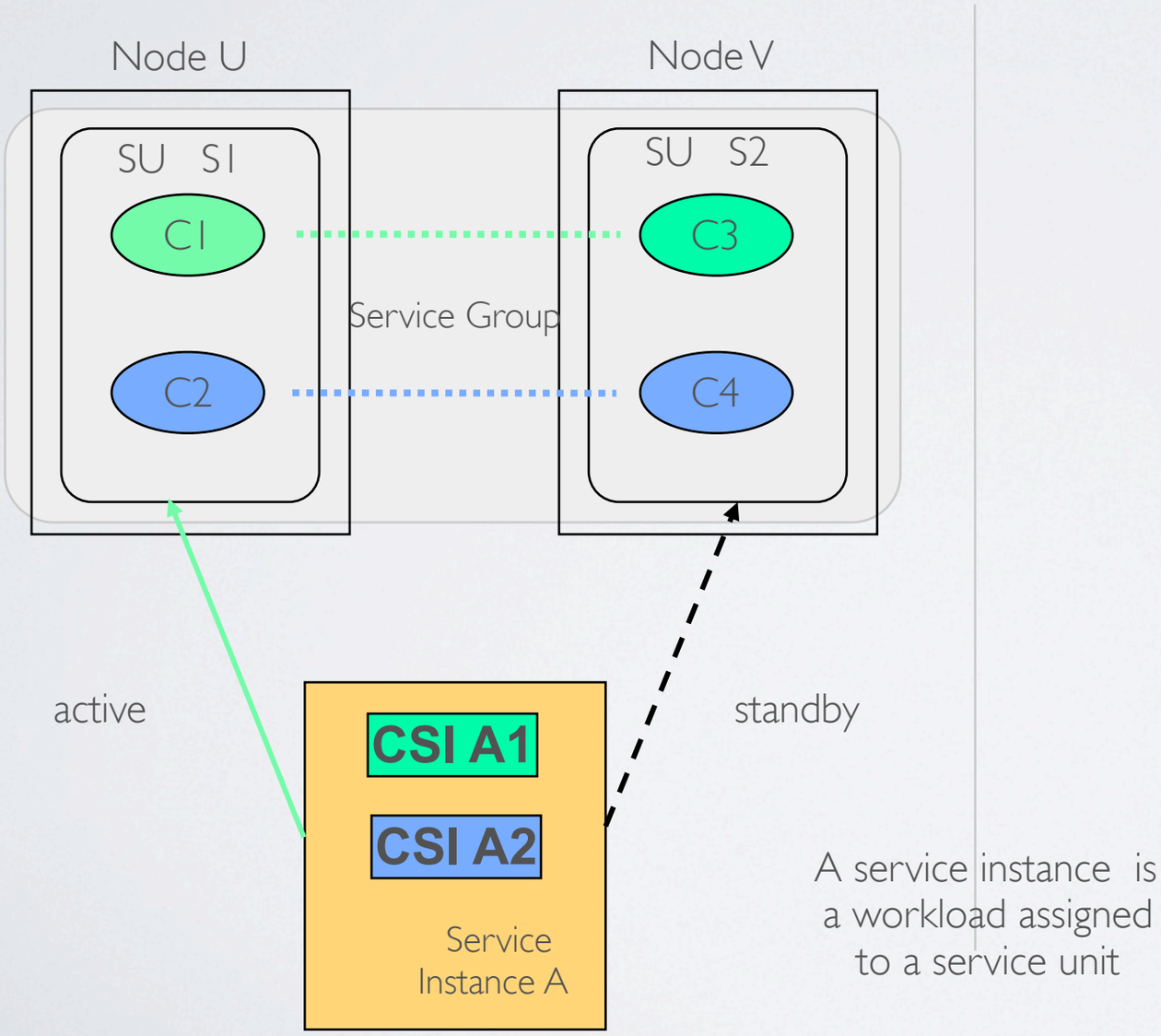
- Example of the 2N redundancy model (with two service units on different nodes)



Copyright© 2006 Service Availability™ Forum, Inc

AVAILABILITY MANAGEMENT FRAMEWORK

- Example of the 2N redundancy model (with two service units on different nodes)



A service instance is a workload assigned to a service unit

Copyright© 2006 Service Availability™ Forum, Inc

AVAILABILITY MANAGEMENT FRAMEWORK

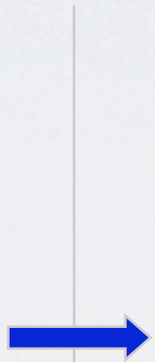
- Example of the 2N redundancy model (with two service units on different nodes)

A service instance is a workload assigned to a service unit

Copyright© 2006 Service Availability™ Forum, Inc

AVAILABILITY MANAGEMENT FRAMEWORK

- Example of the 2N redundancy model (with two service units on different nodes)

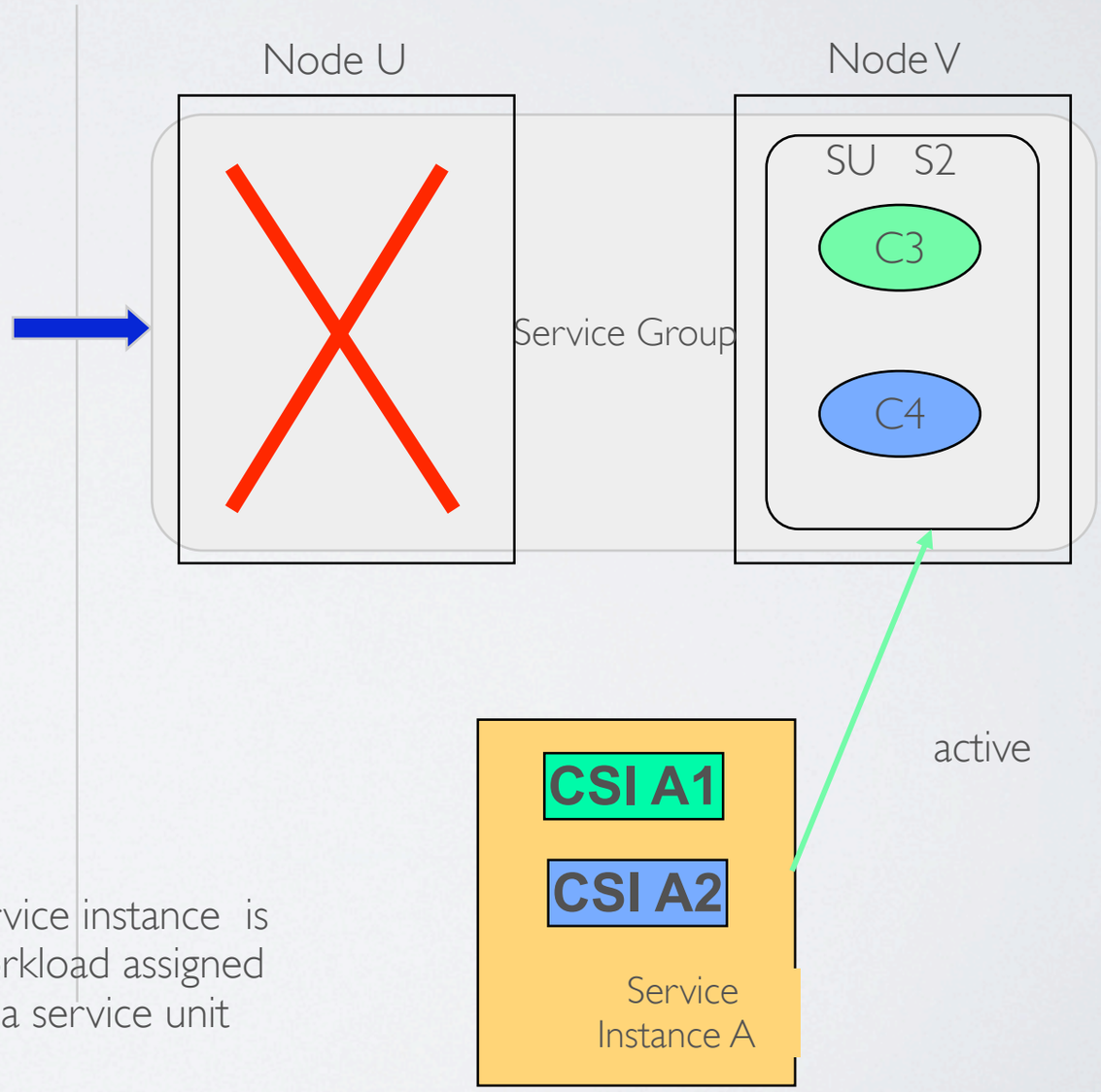


A service instance is a workload assigned to a service unit

Copyright© 2006 Service Availability™ Forum, Inc

AVAILABILITY MANAGEMENT FRAMEWORK

- Example of the 2N redundancy model (with two service units on different nodes)



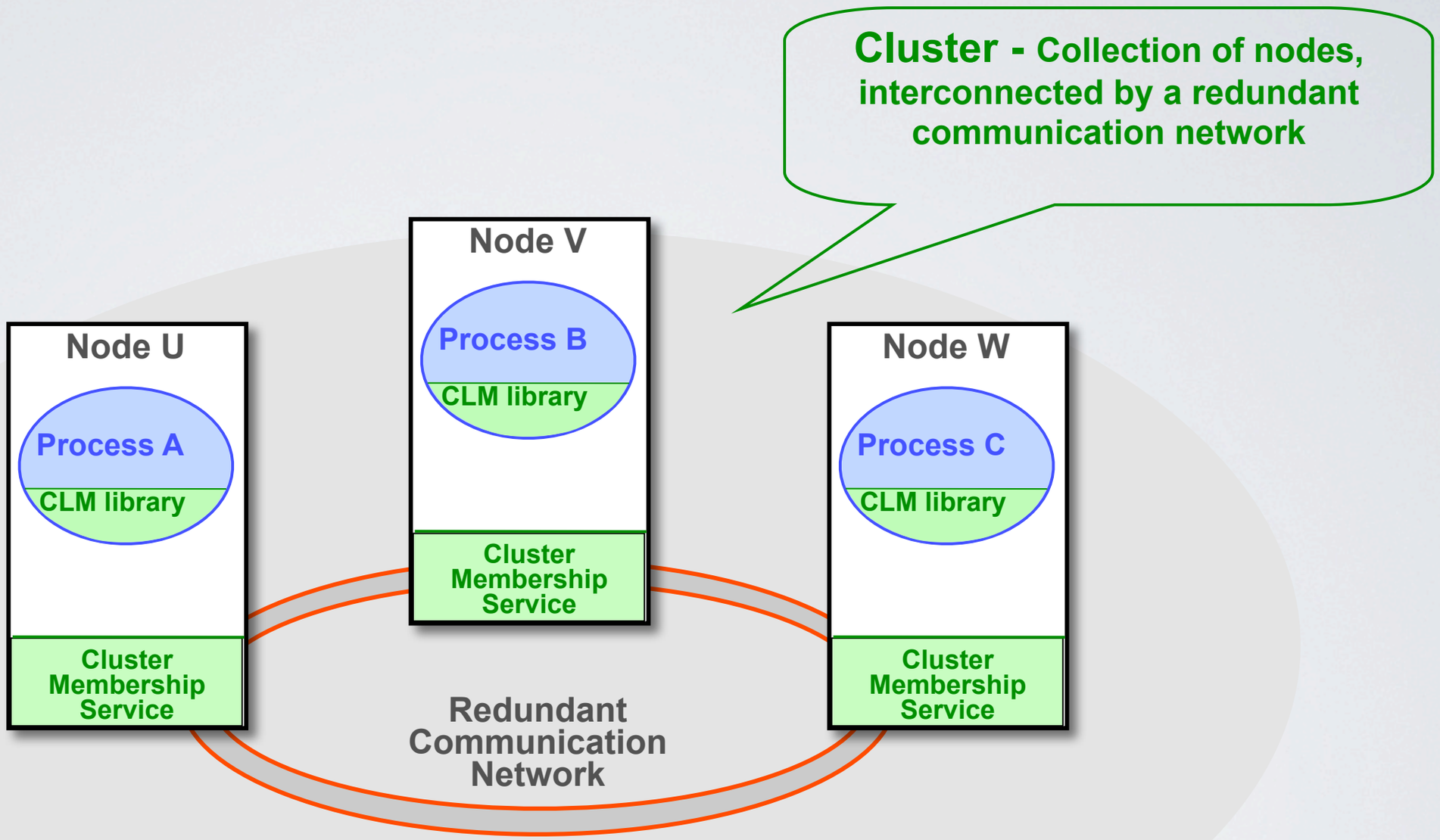
Copyright© 2006 Service Availability™ Forum, Inc

CLUSTER MEMBERSHIP SERVICE

- The Cluster Membership Service provides applications with information about the nodes (the cluster nodes or configured nodes) that have been administratively configured into the cluster
  - The cluster consists of this set of configured nodes
- A member node is a configured node that the Cluster Membership Service has recognized to be healthy and connected to other nodes in the cluster
  - The set of member nodes is the cluster membership, which may change dynamically as nodes join and leave the cluster
- The Cluster Membership Service is the authority that determines whether a cluster node is a member node
- It also allows application processes to retrieve information about the current cluster membership and to register to receive membership change notifications as they occur
- The Cluster Membership Service is used mainly by the AMF and the other AIS services and not directly by the applications

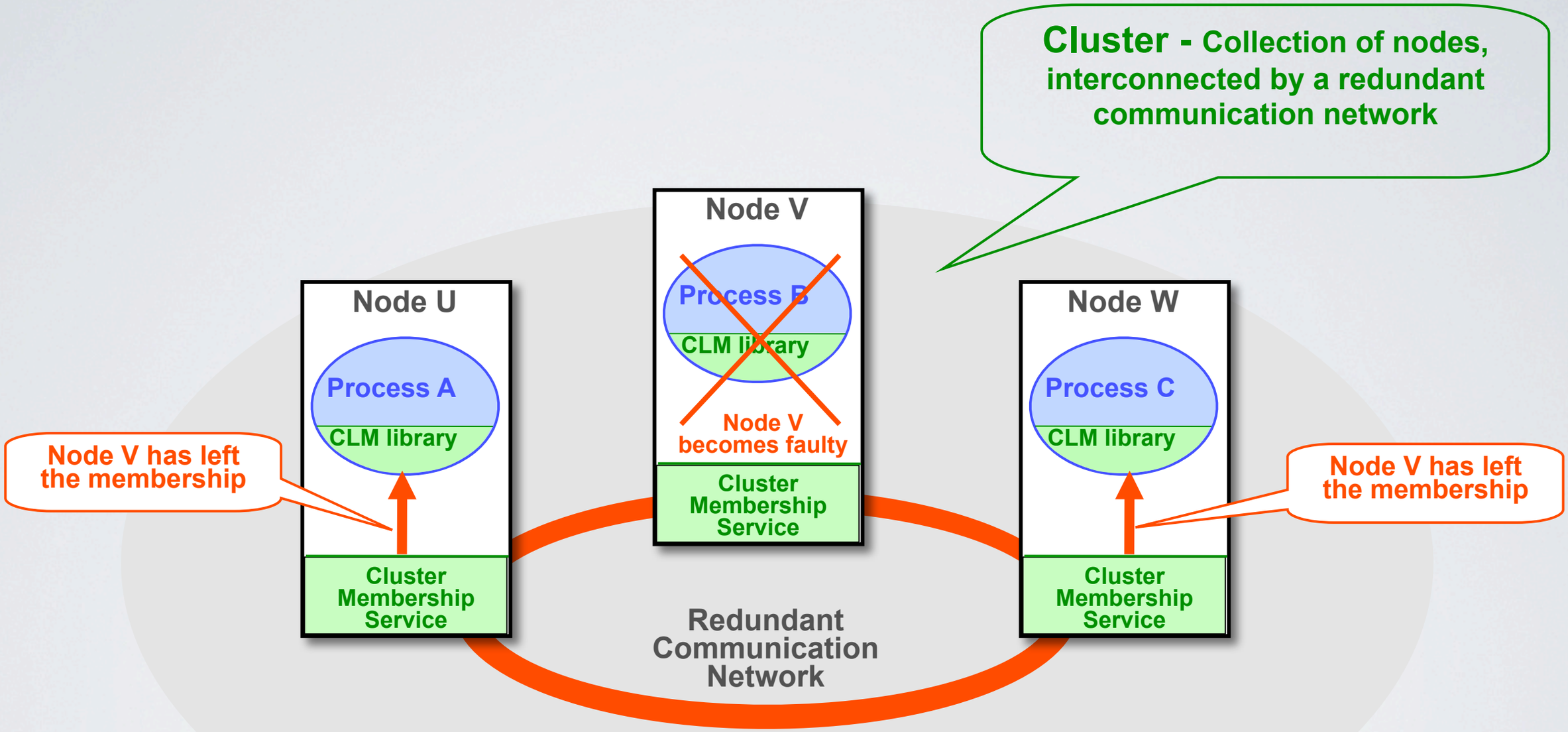
Copyright© 2006 Service Availability™ Forum, Inc

CLUSTER MEMBERSHIP SERVICE



Copyright© 2006 Service Availability™ Forum, Inc

CLUSTER MEMBERSHIP SERVICE



Copyright© 2006 Service Availability™ Forum, Inc

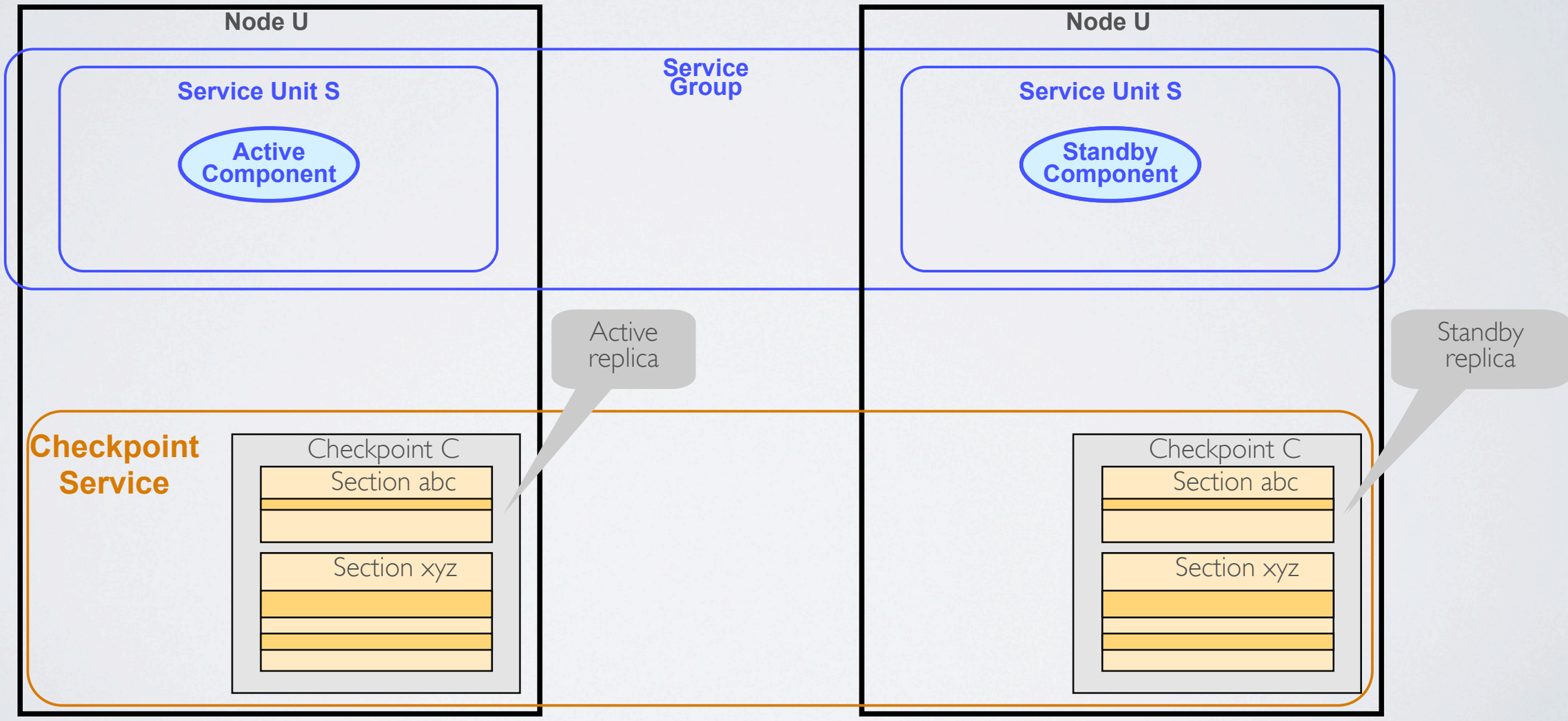
CHECKPOINT SERVICE

- The Checkpoint Service manages a set of entities, called checkpoints, that a process uses to save its state to protect the process against failure and minimize the impact of failure
  - A checkpoint is a cluster-wide entity, with a unique name, that is structured into areas called sections
  - A copy of the data that are stored in a checkpoint is called a checkpoint replica. A checkpoint replica is typically stored in main memory, rather than on disk, for performance reasons
  - A checkpoint may have several checkpoint replicas stored on different nodes in the cluster, to protect against node failure
- When a process writes data to a checkpoint, the Checkpoint Service transfers this data to the checkpoint replicas. Synchronous update and asynchronous update options are provided, and the process can invoke an API function to synchronize the checkpoint replicas
- When a process recovers from a failure with a restart or failover procedure, the Checkpoint Service is used to retrieve the checkpoint data, instantiate the process with the checkpoint, and resume execution of the process from the state recorded before the failure

Copyright© 2006 Service Availability™ Forum, Inc

CHECKPOINT SERVICE

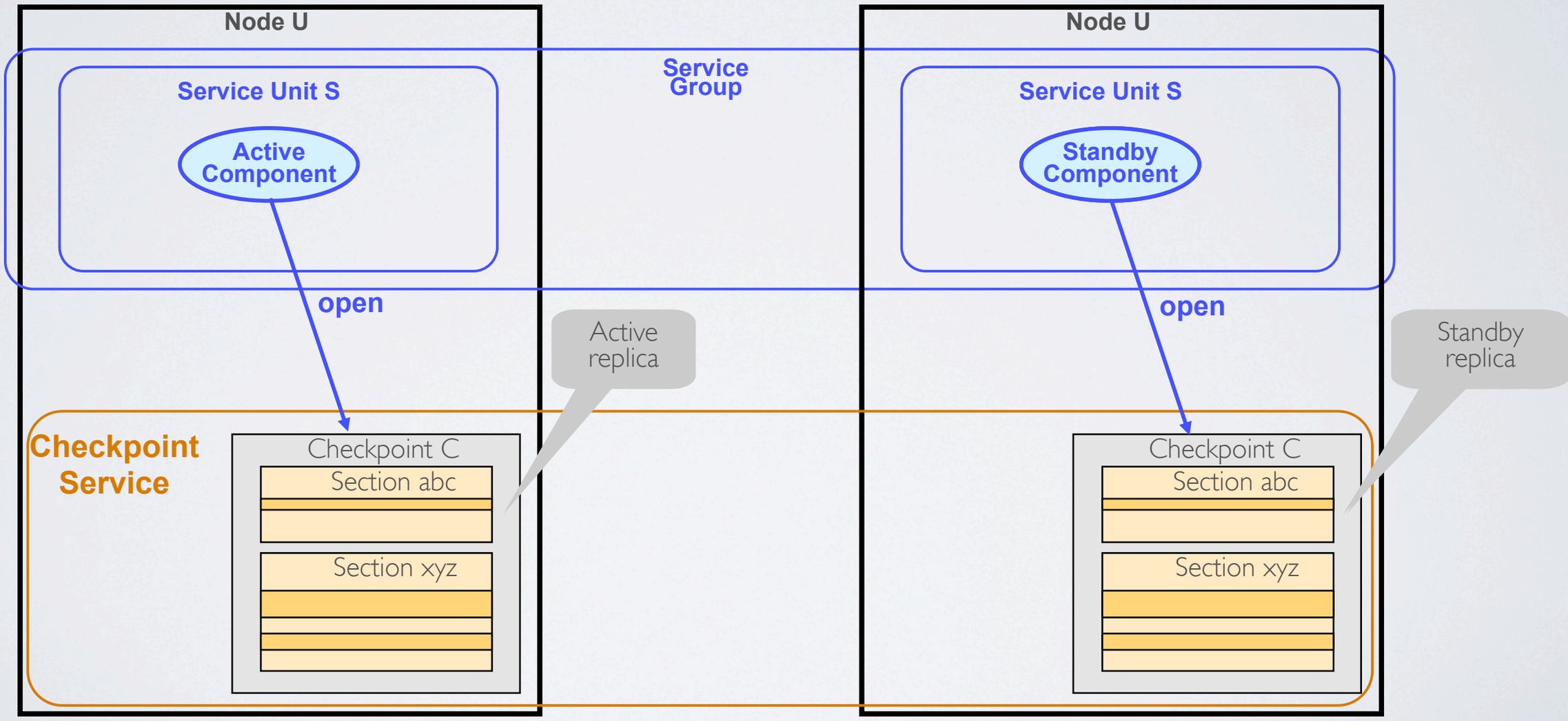
- Example of checkpointing, and restoration from a checkpoint after a fault, for a collocated checkpoint



Copyright© 2006 Service Availability™ Forum, Inc

CHECKPOINT SERVICE

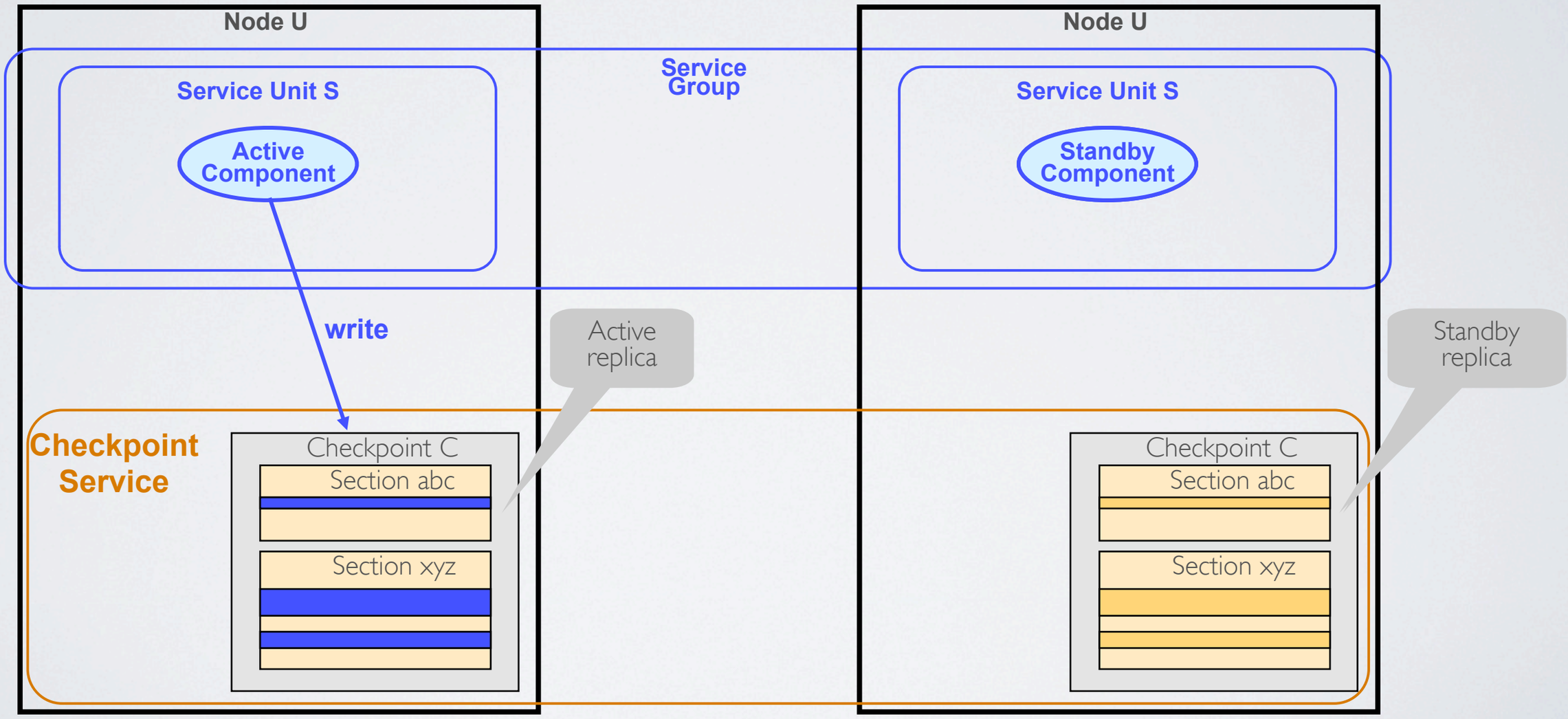
- Example of checkpointing, and restoration from a checkpoint after a fault, for a collocated checkpoint



Copyright© 2006 Service Availability™ Forum, Inc

CHECKPOINT SERVICE

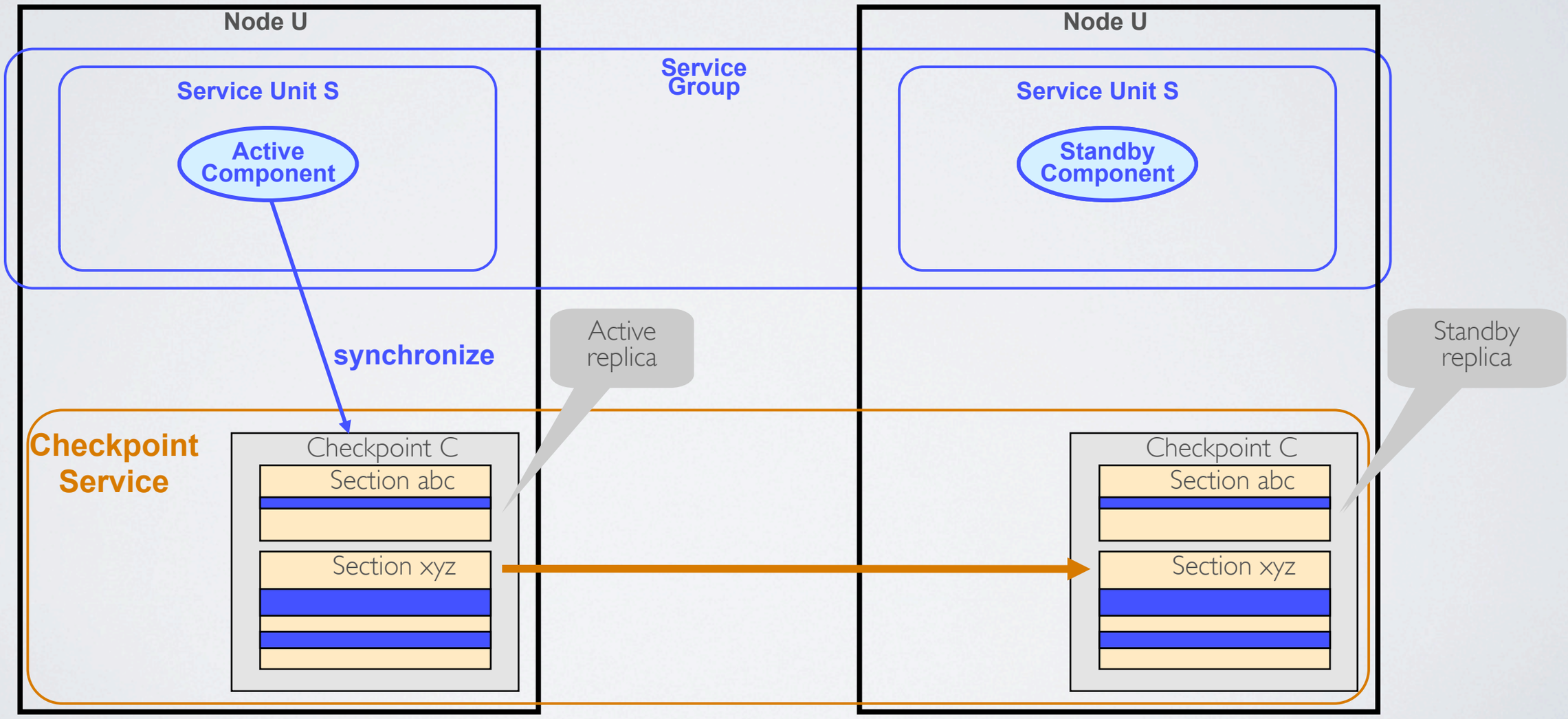
- Example of checkpointing, and restoration from a checkpoint after a fault, for a collocated checkpoint



Copyright© 2006 Service Availability™ Forum, Inc

CHECKPOINT SERVICE

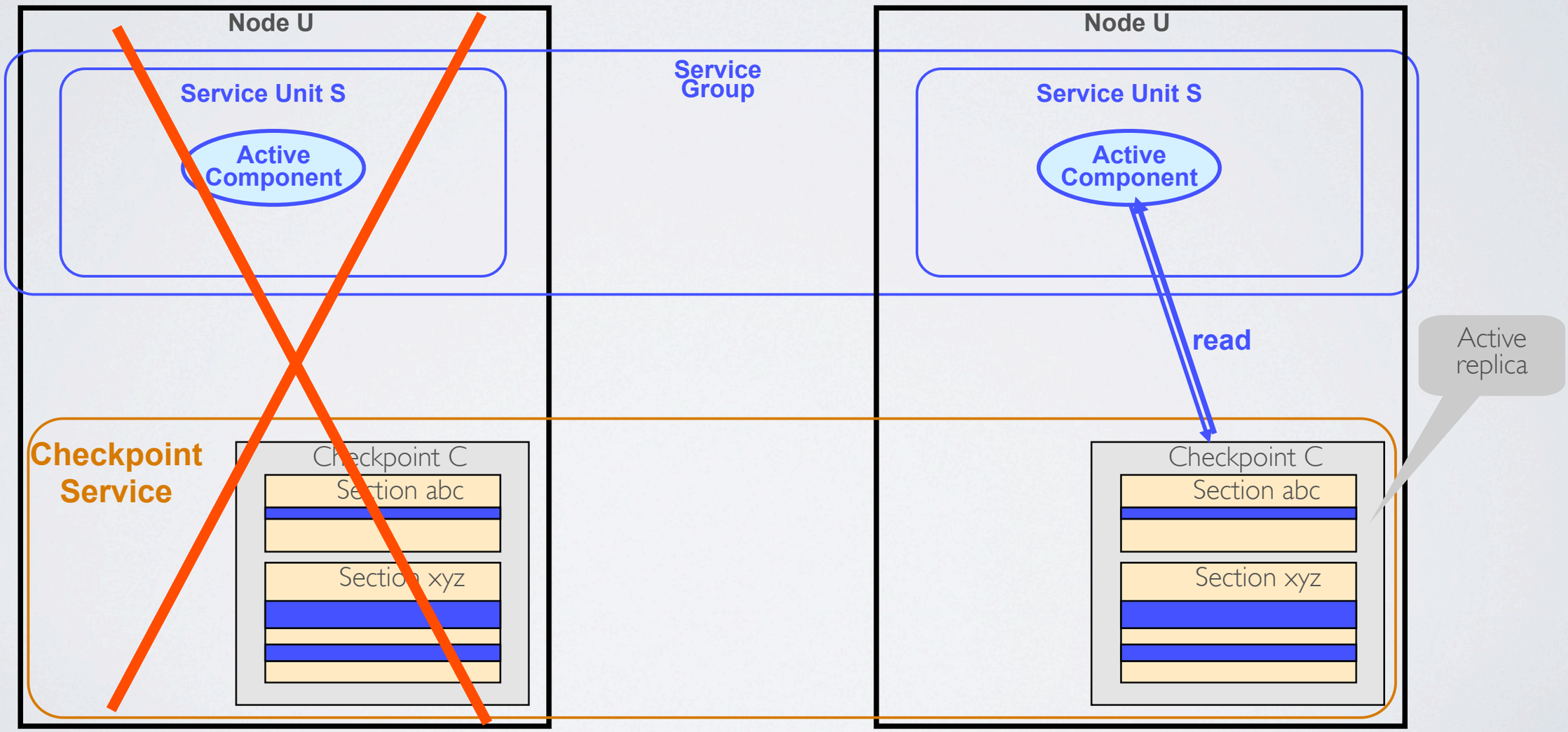
- Example of checkpointing, and restoration from a checkpoint after a fault, for a collocated checkpoint



Copyright© 2006 Service Availability™ Forum, Inc

CHECKPOINT SERVICE

- Example of checkpointing, and restoration from a checkpoint after a fault, for a collocated checkpoint



Copyright© 2006 Service Availability™ Forum, Inc

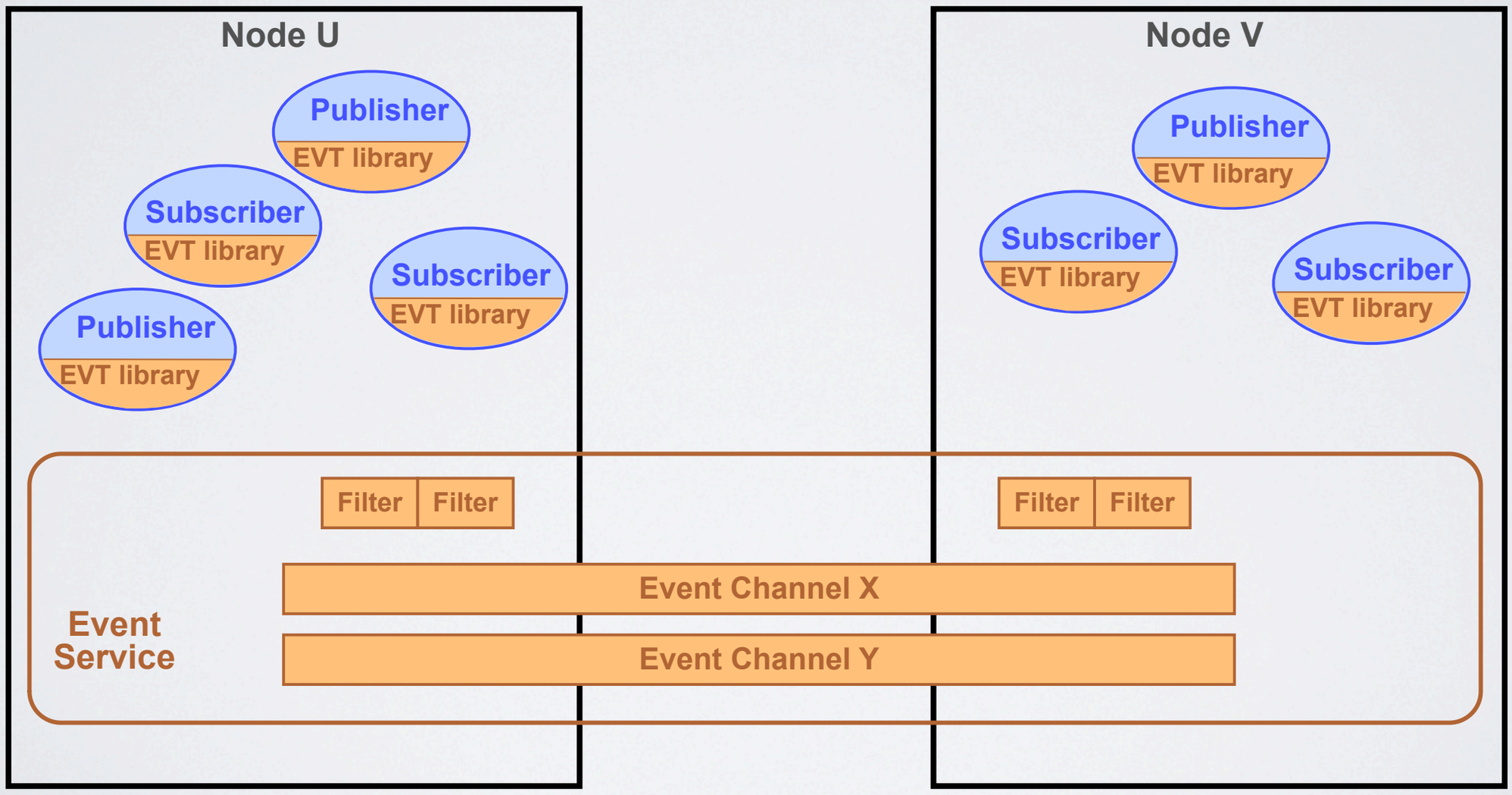
EVENT SERVICE

- The Event Service is a publish/subscribe multipoint-to-multipoint communication mechanism based on cluster-wide event channels
- It allows one or more publishers to communicate asynchronously with one or more subscribers via events over the event channels
  - An event consists of an event header, containing a set of event attributes, and zero or more bytes of event data, contained in a separate free-form data buffer
  - Multiple publishers and multiple subscribers can communicate over the same event channel
  - Individual publishers and individual subscribers can communicate over multiple event channels
  - Subscribers are anonymous, which means that they may join and leave an event channel at any time without involving the publisher(s)
  - A subscriber may supply a filter, through a set of event attributes represented as an array of event patterns, to the Event Service, which then delivers only events that match the filter to the subscriber

Copyright© 2006 Service Availability™ Forum, Inc

EVENT SERVICE

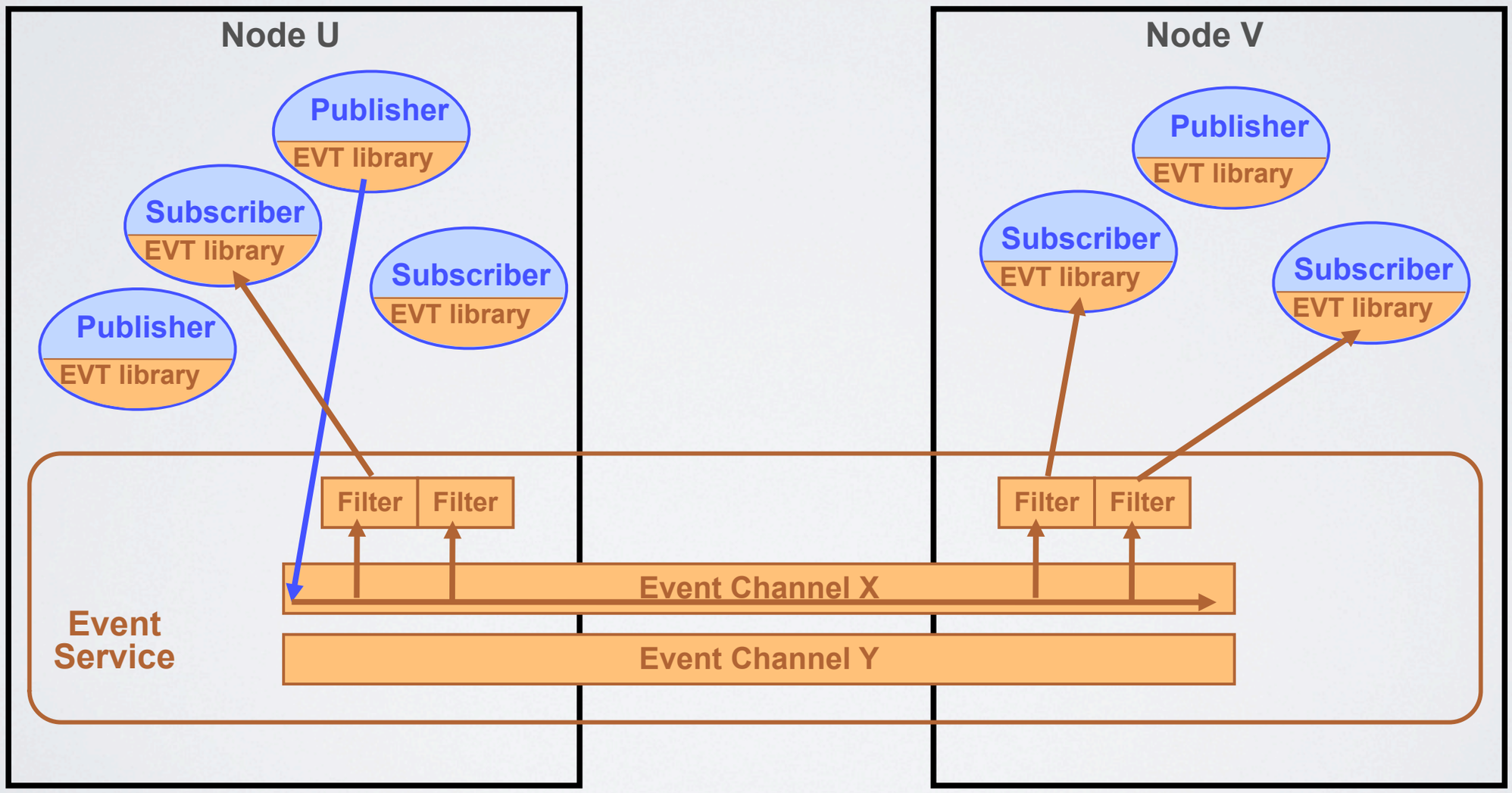
- Example of the Event Service, with publishers and subscribers on two nodes



Copyright© 2006 Service Availability™ Forum, Inc

EVENT SERVICE

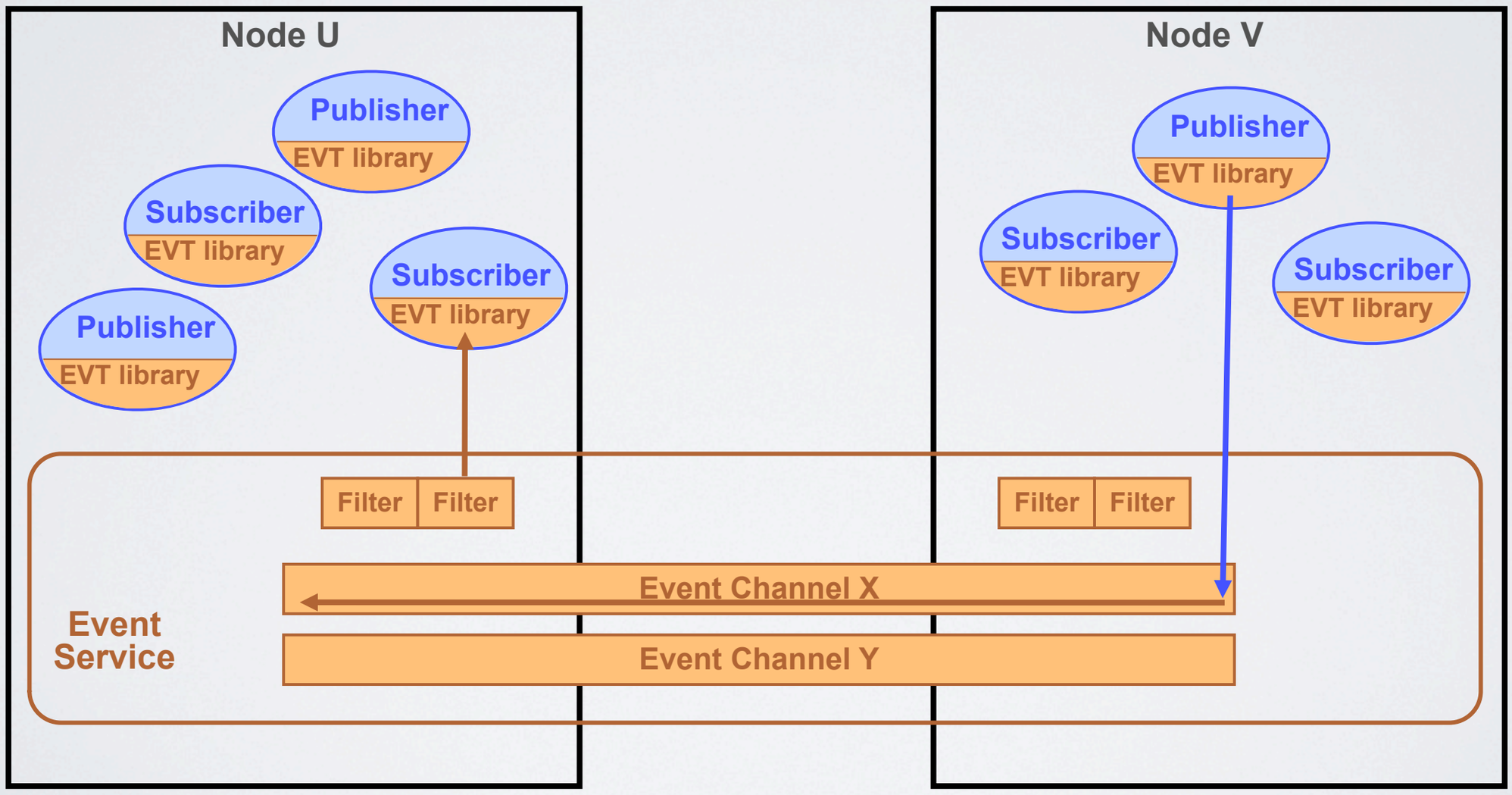
- Example of the Event Service, with publishers and subscribers on two nodes



Copyright© 2006 Service Availability™ Forum, Inc

EVENT SERVICE

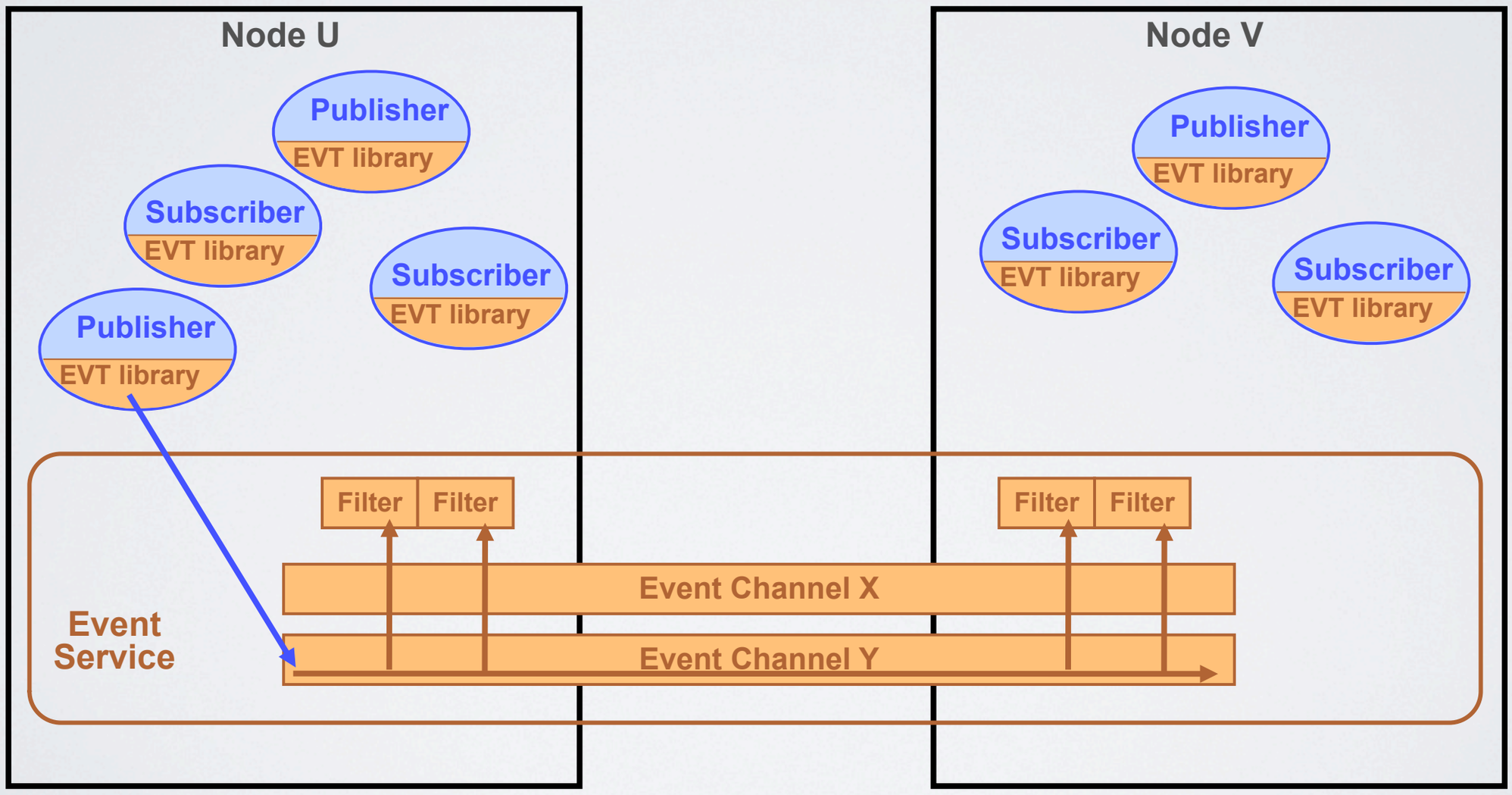
- Example of the Event Service, with publishers and subscribers on two nodes



Copyright© 2006 Service Availability™ Forum, Inc

EVENT SERVICE

- Example of the Event Service, with publishers and subscribers on two nodes



Copyright© 2006 Service Availability™ Forum, Inc

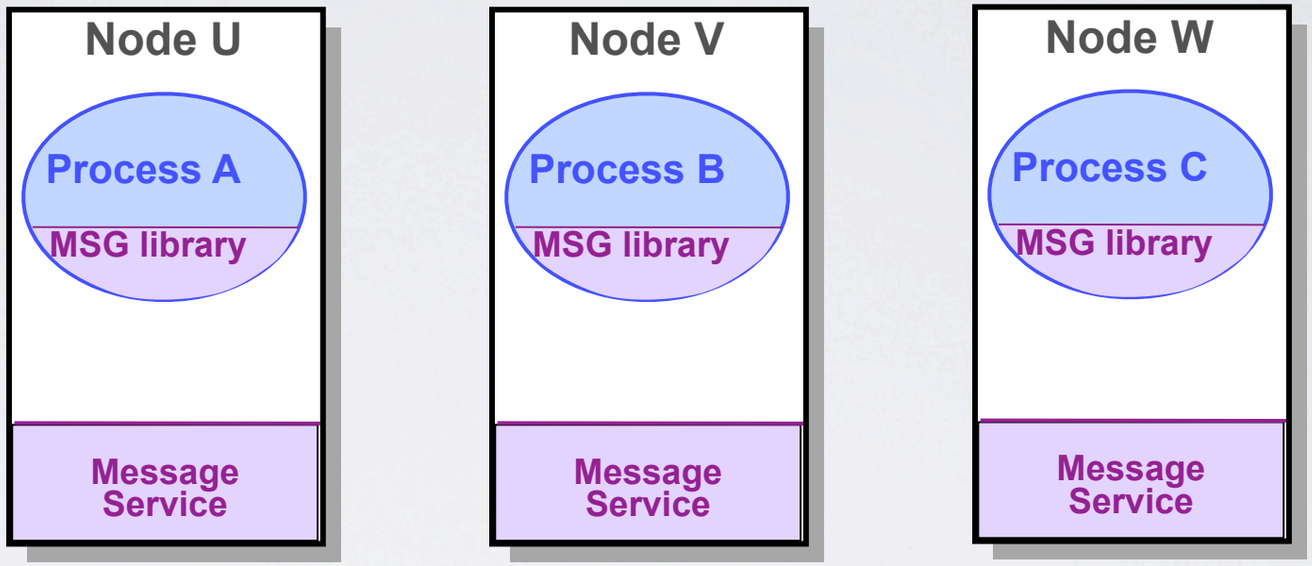
MESSAGE SERVICE

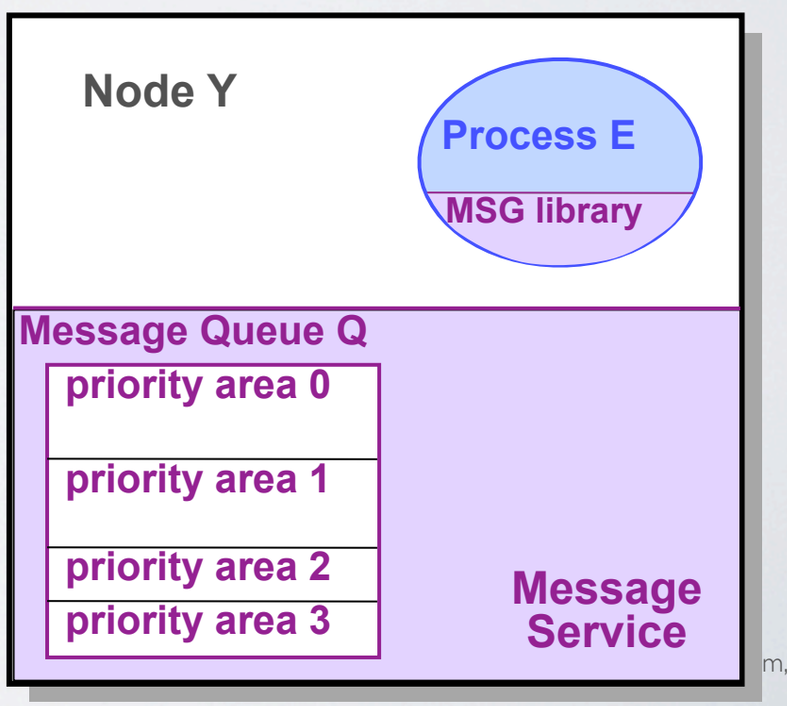
- The Message Service is a buffered message passing system, for processes on the same or different nodes, that is based on the concept of a message queue
  - Messages are written to, and read from, message queues
- A message queue permits multipoint-to-point communication
  - Many processes can write into a message queue, but only one process can open a message queue and read from it at any time
  - A sender process is unaware that a receiver process that fails or switches over has been replaced by another process acting as a standby
- A message queue group permits multipoint-to-multipoint communication
  - Many processes can read from a message queue group at the same time
  - Message queue groups can be used to maintain transparency of the sender process to faults in the receiver processes, and also to distribute workload among message queues in the group

Copyright© 2006 Service Availability™ Forum, Inc

Message Service

- Message Service, Message Service (MSG) Library, and Message Queue

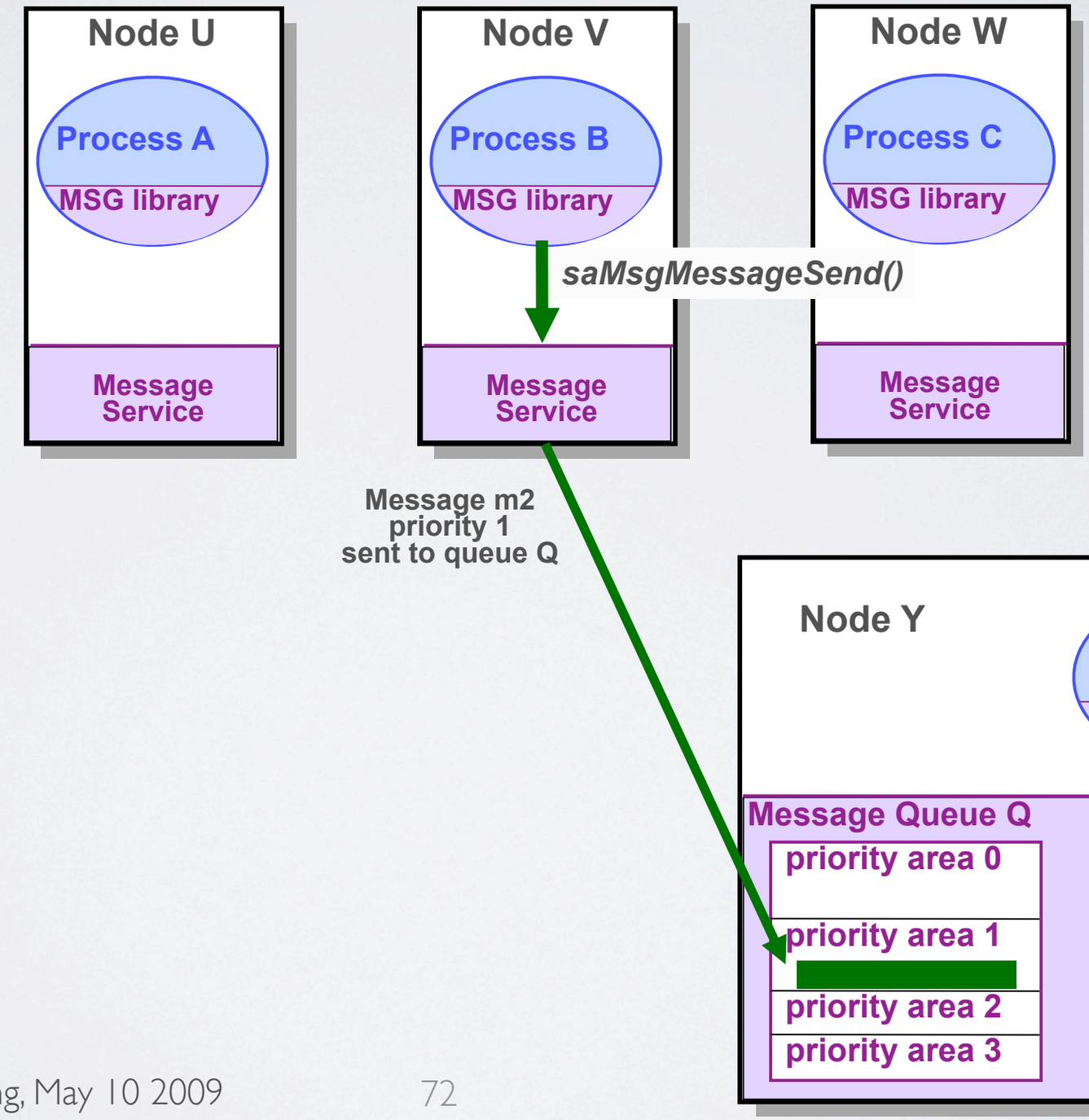




m, Inc

Message Service

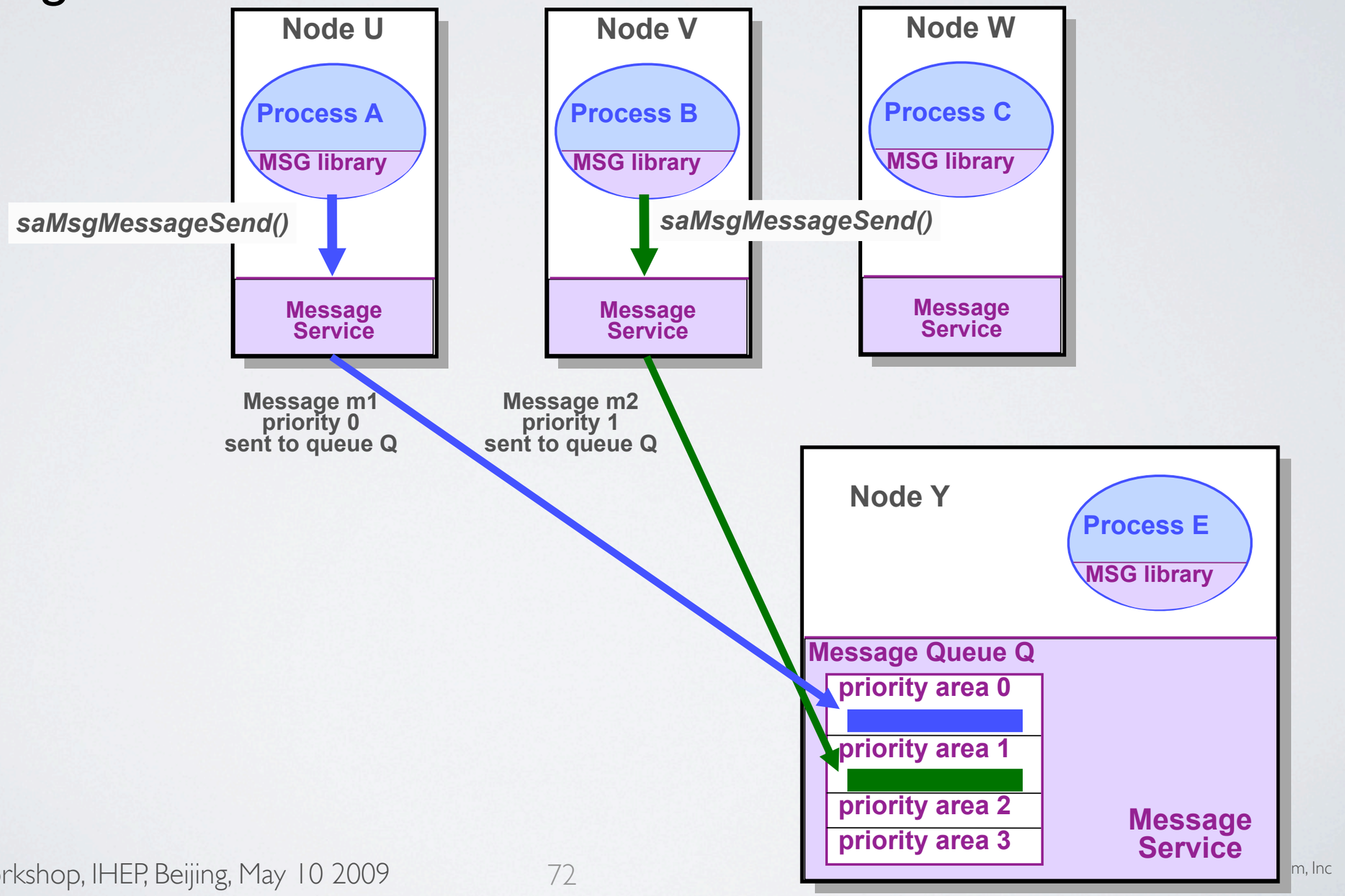
- Message Service, Message Service (MSG) Library, and Message Queue



m, Inc

Message Service

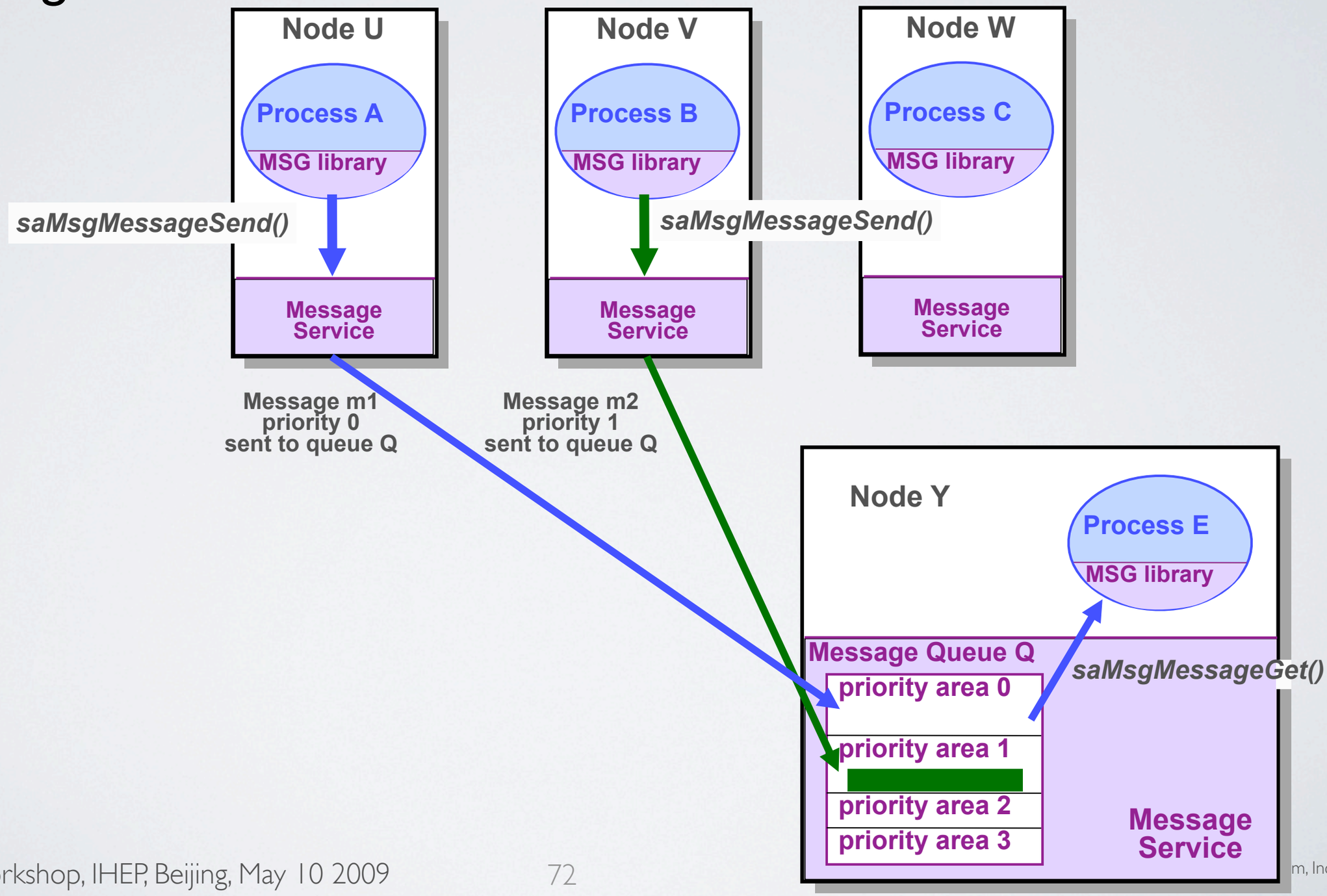
- Message Service, Message Service (MSG) Library, and Message Queue



m, Inc

Message Service

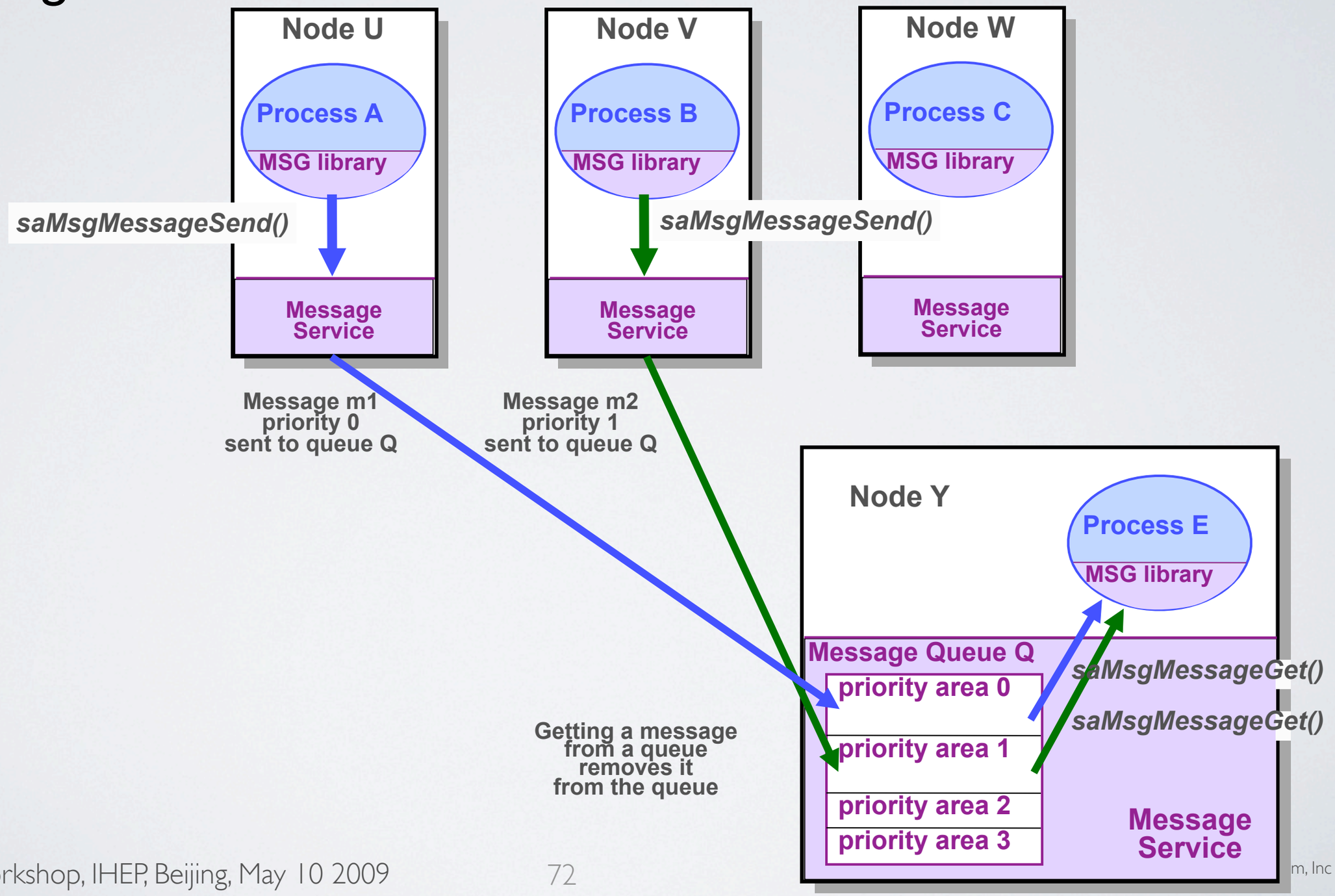
- Message Service, Message Service (MSG) Library, and Message Queue



m, Inc

Message Service

- Message Service, Message Service (MSG) Library, and Message Queue



m, Inc

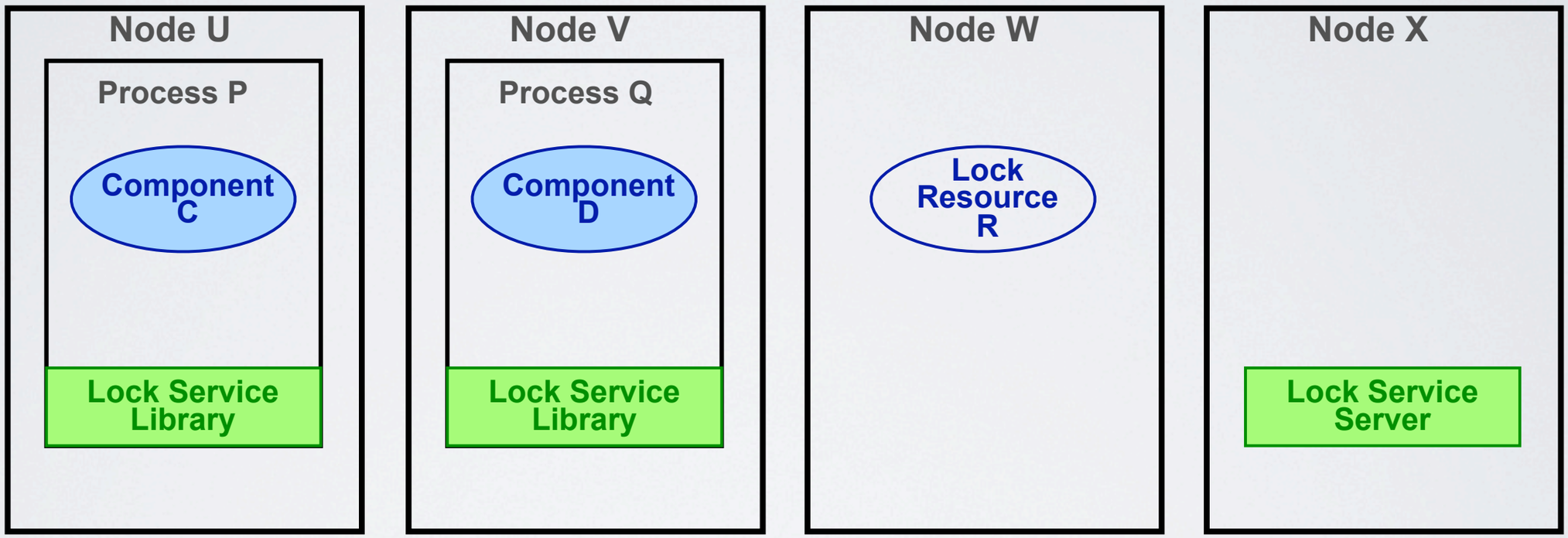
LOCK SERVICE

- The Lock Service is a distributed lock service that allows different application processes on the same or different nodes in the cluster to compete for access to a shared resource in the cluster
- A lock resource is a globally-named resource, access to which is controlled by a lock
- A lock is used to synchronize access to a shared resource by different application processes
- There are two lock modes
  - Exclusive access: Exclusive lock (EX lock)
    Only one process can hold the lock at the same time
  - Shared access: Protected read lock (PR lock)
    One or more processes can hold the lock at the same time
- Optional features
  - Deadlock detection
  - Lock orphaning

Copyright© 2006 Service Availability™ Forum, Inc

LOCK SERVICE

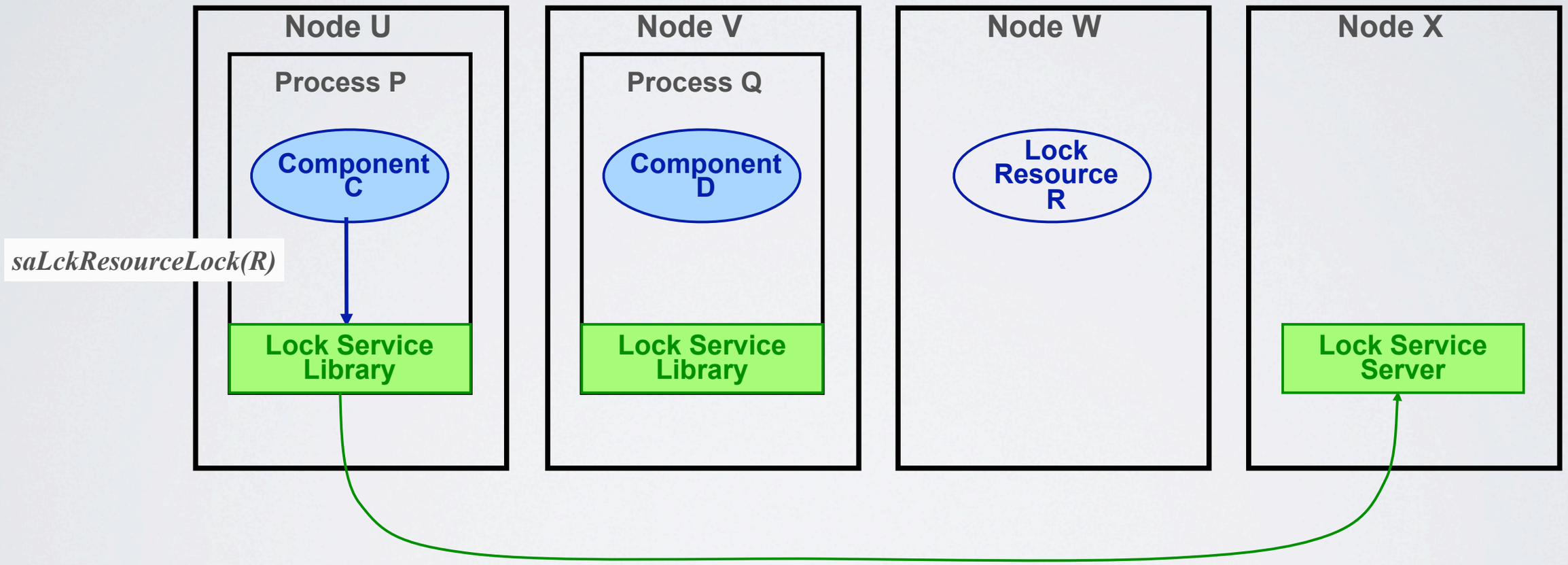
- The Lock Service Library linked into application processes and a Lock Service Server



Copyright© 2006 Service Availability™ Forum, Inc

LOCK SERVICE

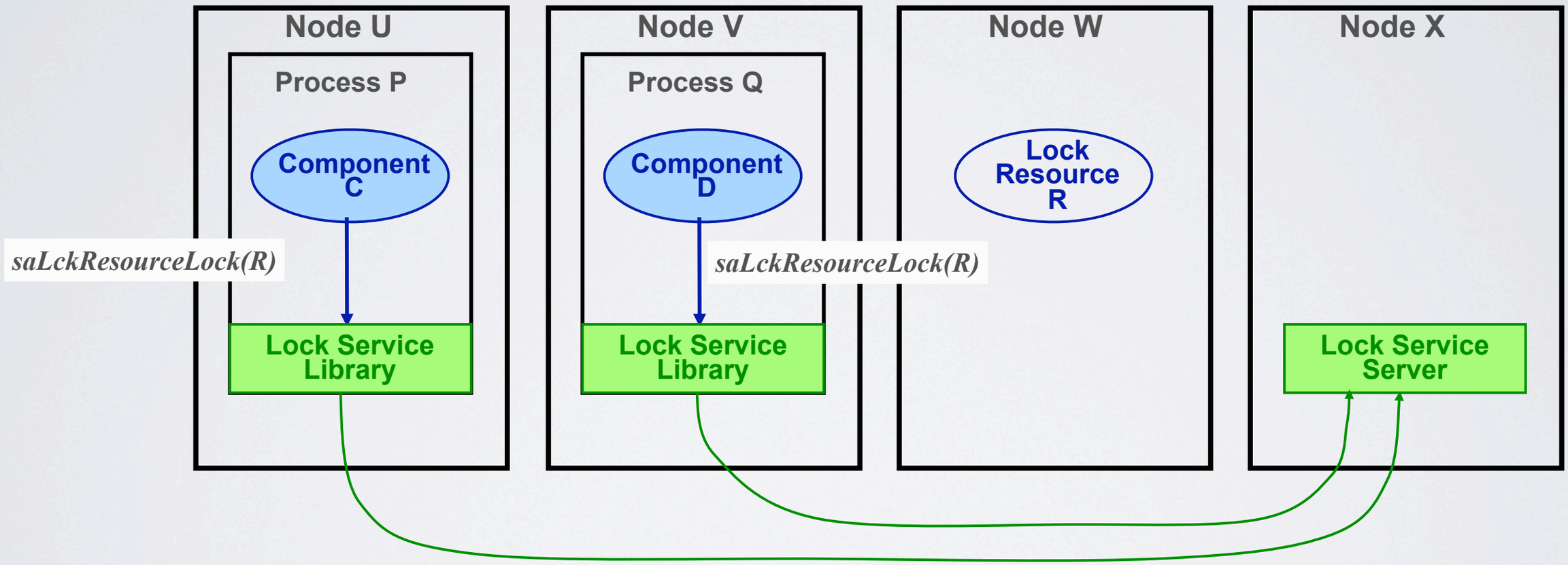
- The Lock Service Library linked into application processes and a Lock Service Server



Copyright© 2006 Service Availability™ Forum, Inc

LOCK SERVICE

- The Lock Service Library linked into application processes and a Lock Service Server



Copyright© 2006 Service Availability™ Forum, Inc

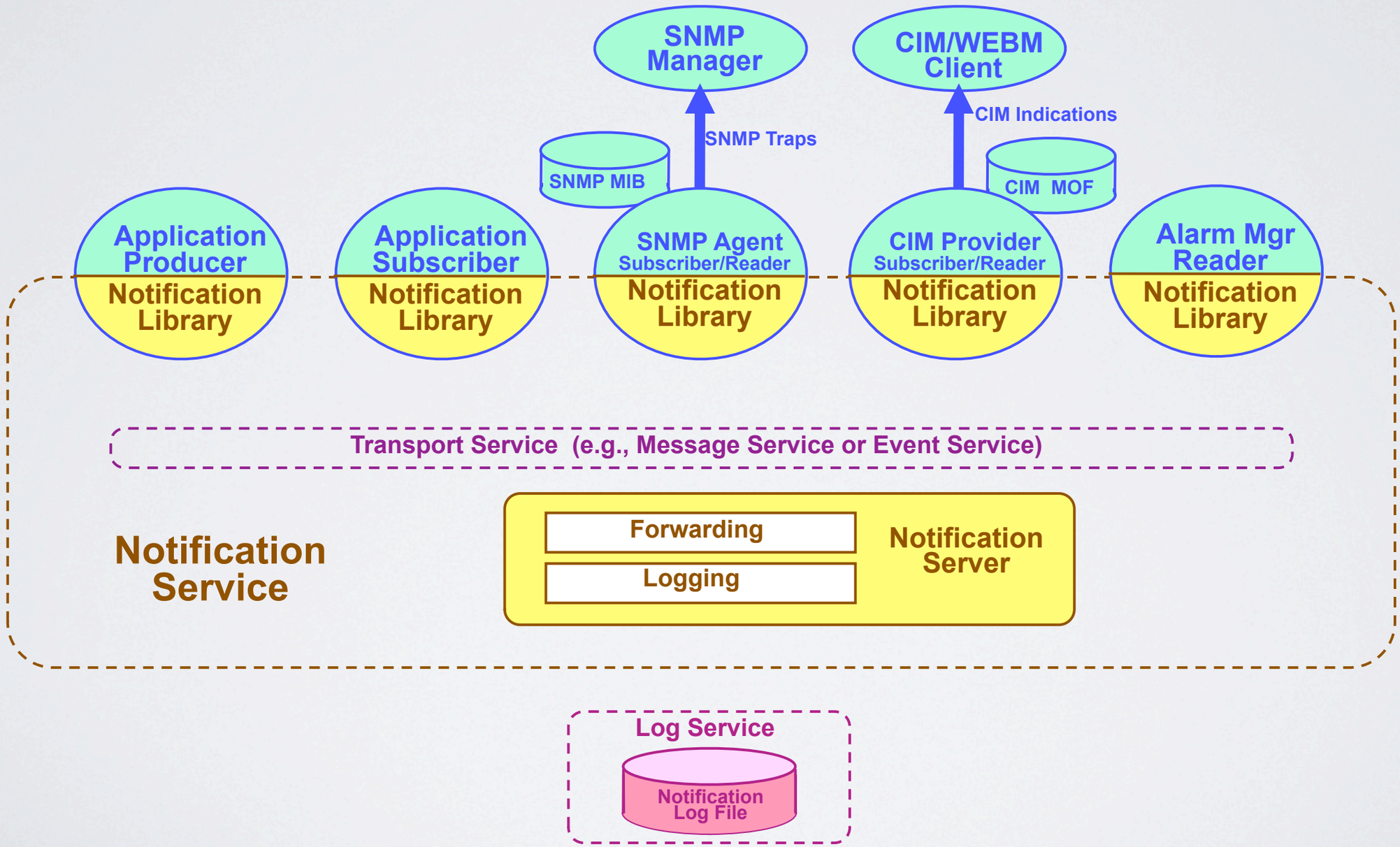
NOTIFICATION SERVICE

- The Notification Service reports incidents or changes in statuses of entities as notifications
- Notifications are data objects that are categorized into the following types
  - Alarm
  - State Change
  - Object Create / Delete
  - Attribute Change
  - Security Alarm
- Notification producers generate notifications and supply them to the Notification Service
- Notification consumers consume notifications generated by producers, and can be one of the following types or both
  - A subscriber that receives notifications from the Notification Service as they occur (push interface)
  - A reader that retrieves historical notification entries from the persistent Notification Log (pull interface)
- Notification filters can be used to
  - Reduce the number of notifications that a consumer receives
  - Allow a user application to specify the notifications in which it is interested

Copyright© 2006 Service Availability™ Forum, Inc

NOTIFICATION SERVICE

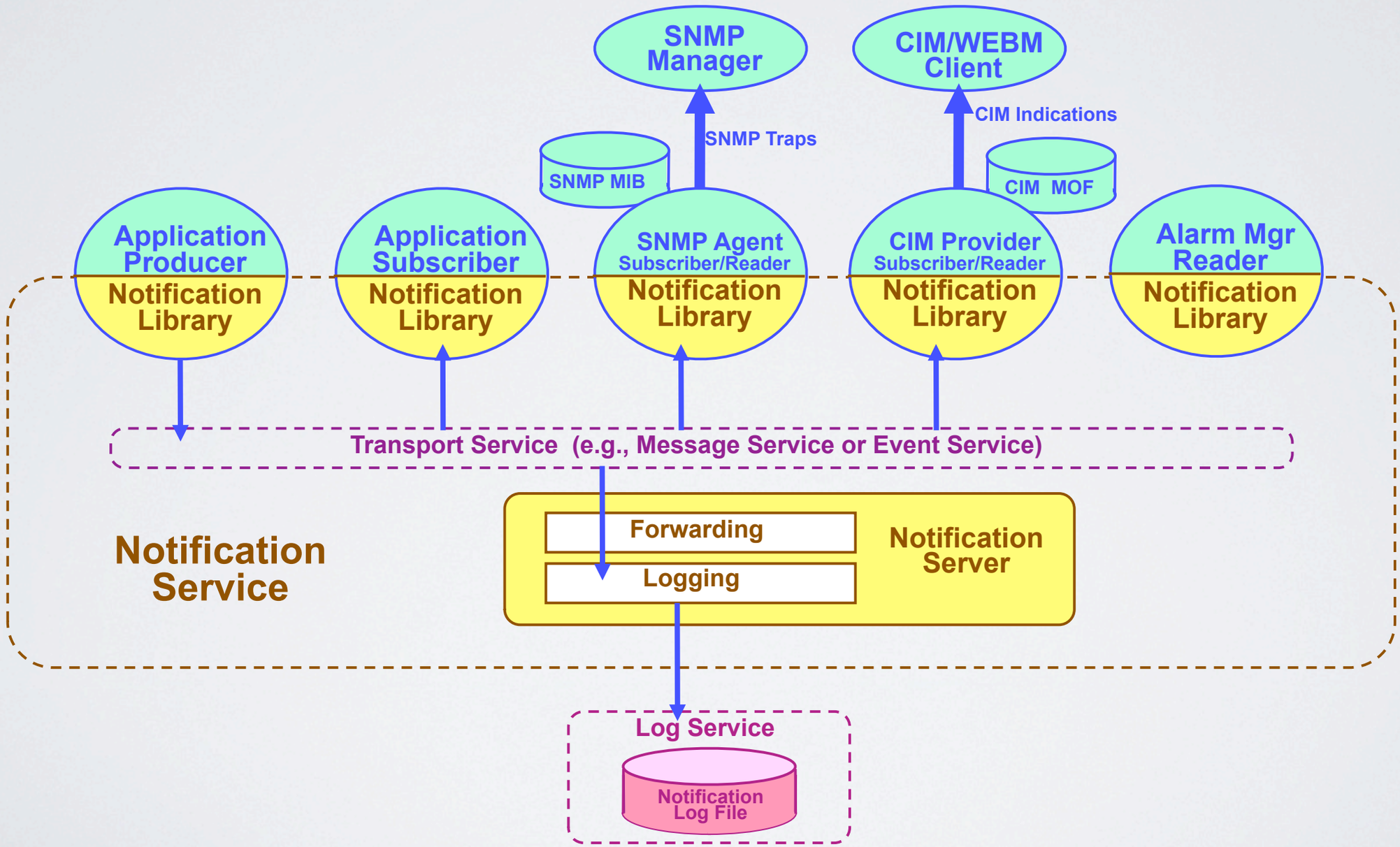
- The Notification Library and the Notification Server and the interactions with other entities in the system



Copyright© 2006 Service Availability™ Forum, Inc

NOTIFICATION SERVICE

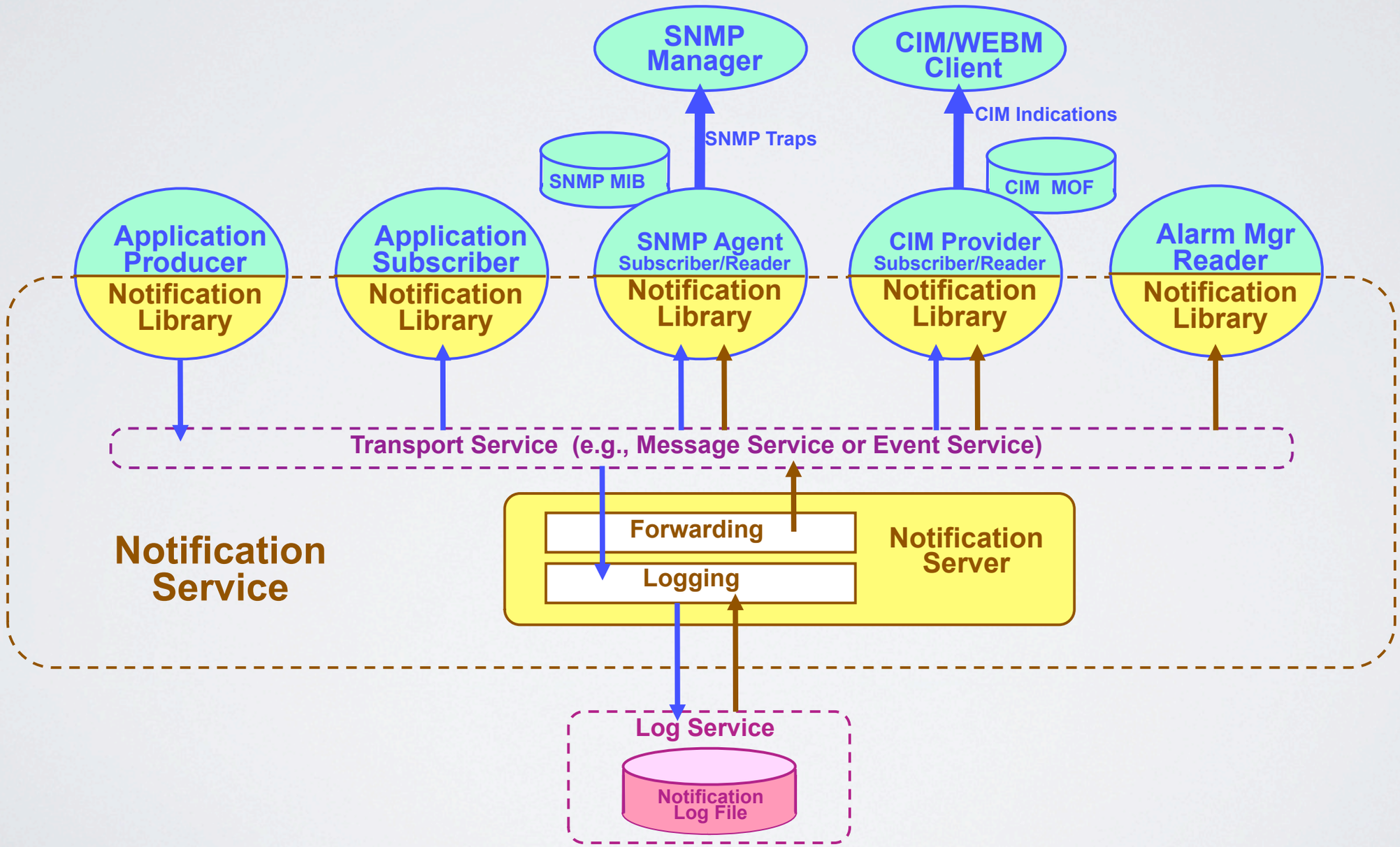
- The Notification Library and the Notification Server and the interactions with other entities in the system



Copyright© 2006 Service Availability™ Forum, Inc

NOTIFICATION SERVICE

- The Notification Library and the Notification Server and the interactions with other entities in the system



Copyright© 2006 Service Availability™ Forum, Inc

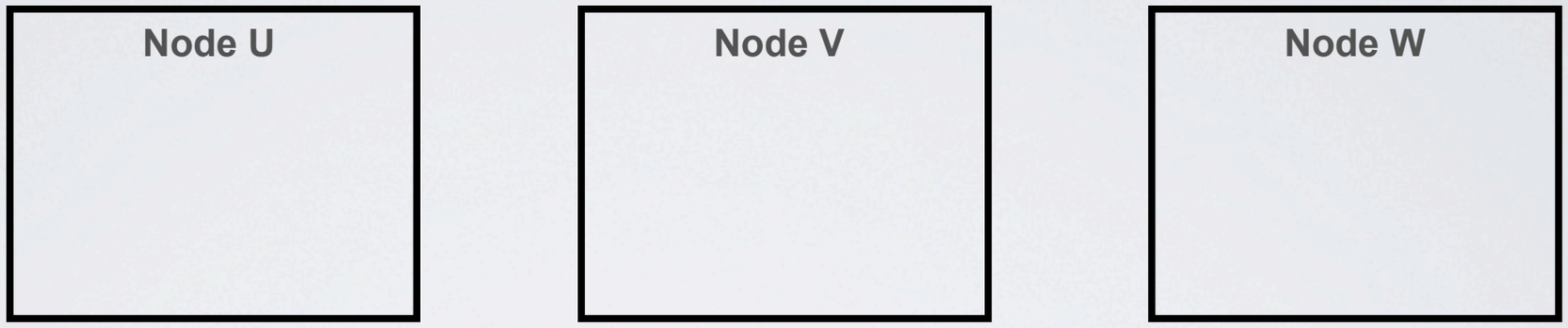
LOG SERVICE

- The Log Service records cluster-significant information in a permanent repository that can be used for subsequent analysis by system administrators and automated tools
- The Log Service enables application to express and forward log records through well-known log streams that lead to particular output destinations such as named files
  - Once at the output destination, a log record is subject to configurable and public output formatting rules
- A log configuration file is a publicly readable file that explains the properties associated with a log stream, such as how the log record data are formatted in the log file(s)
- The Log Service supports four types of log streams
  - Alarm log stream for ITU X.733 and ITU X.736 based log records
  - Notification log stream for ITU X.730 and ITU X.731 based log records
  - System log stream for system-relevant log records
  - Application log streams for application-specific log records

Copyright© 2006 Service Availability™ Forum, Inc

LOG SERVICE

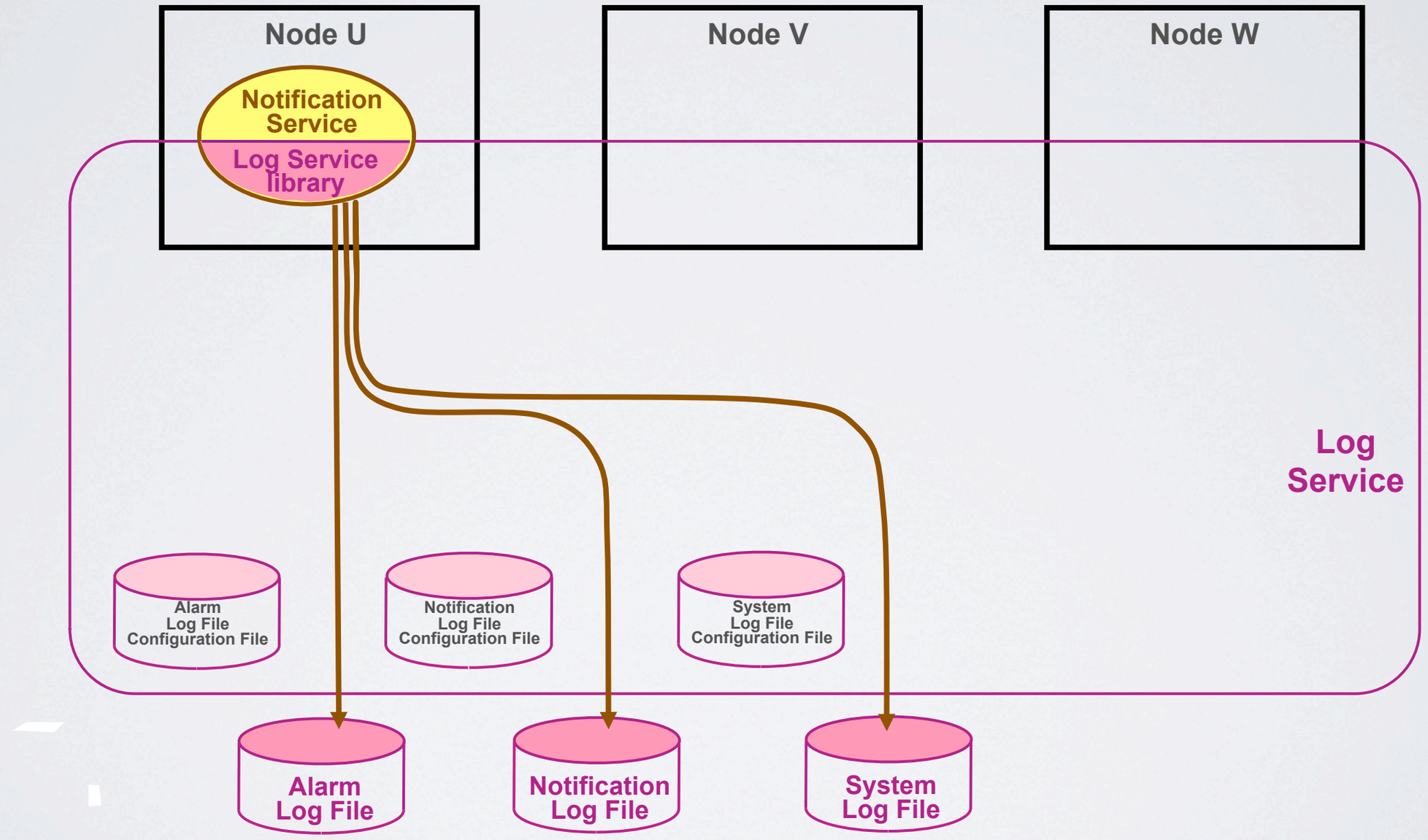
- The Log Service library, log files, and log file configuration files



Copyright© 2006 Service Availability™ Forum, Inc

LOG SERVICE

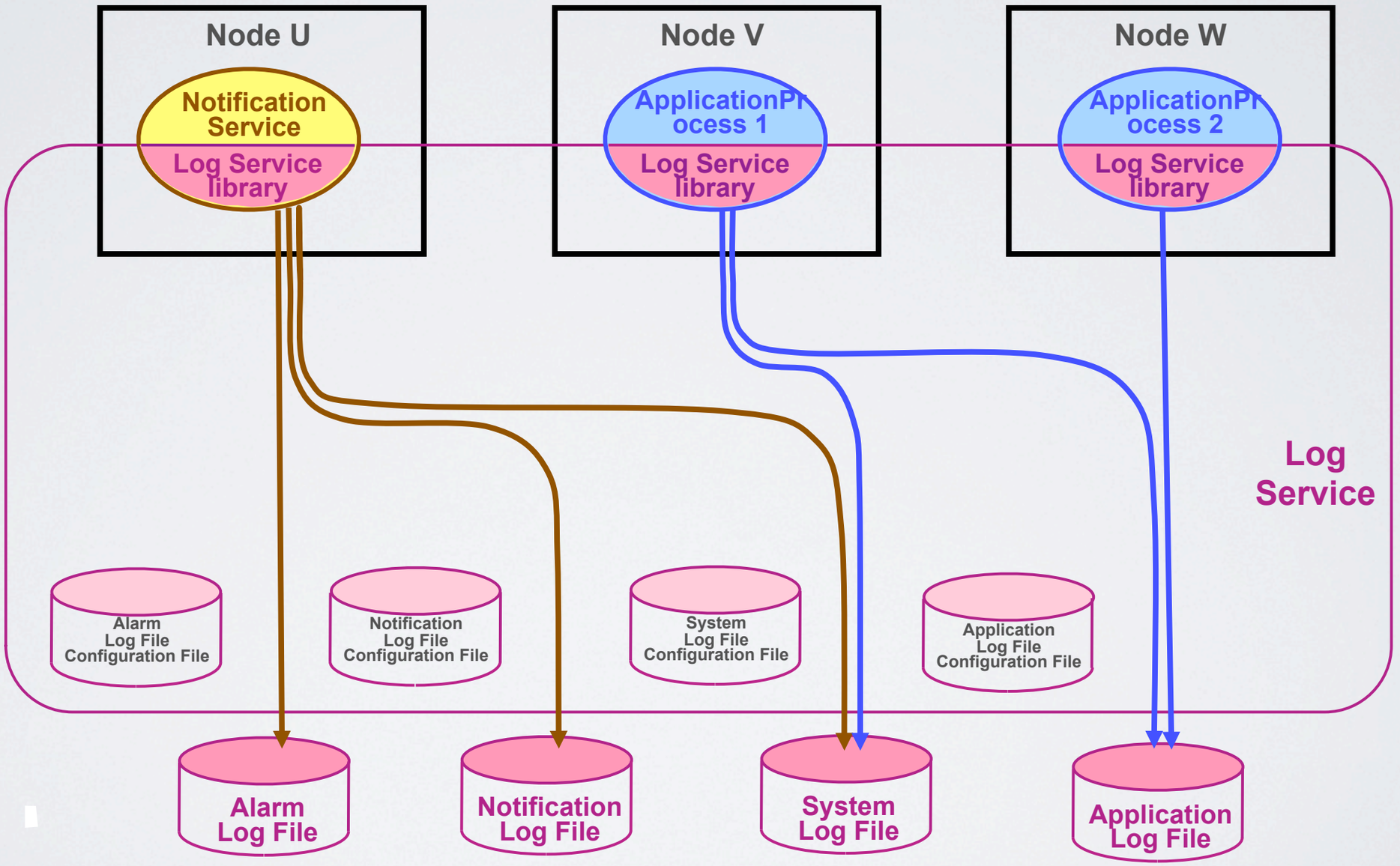
- The Log Service library, log files, and log file configuration files



Copyright© 2006 Service Availability™ Forum, Inc

LOG SERVICE

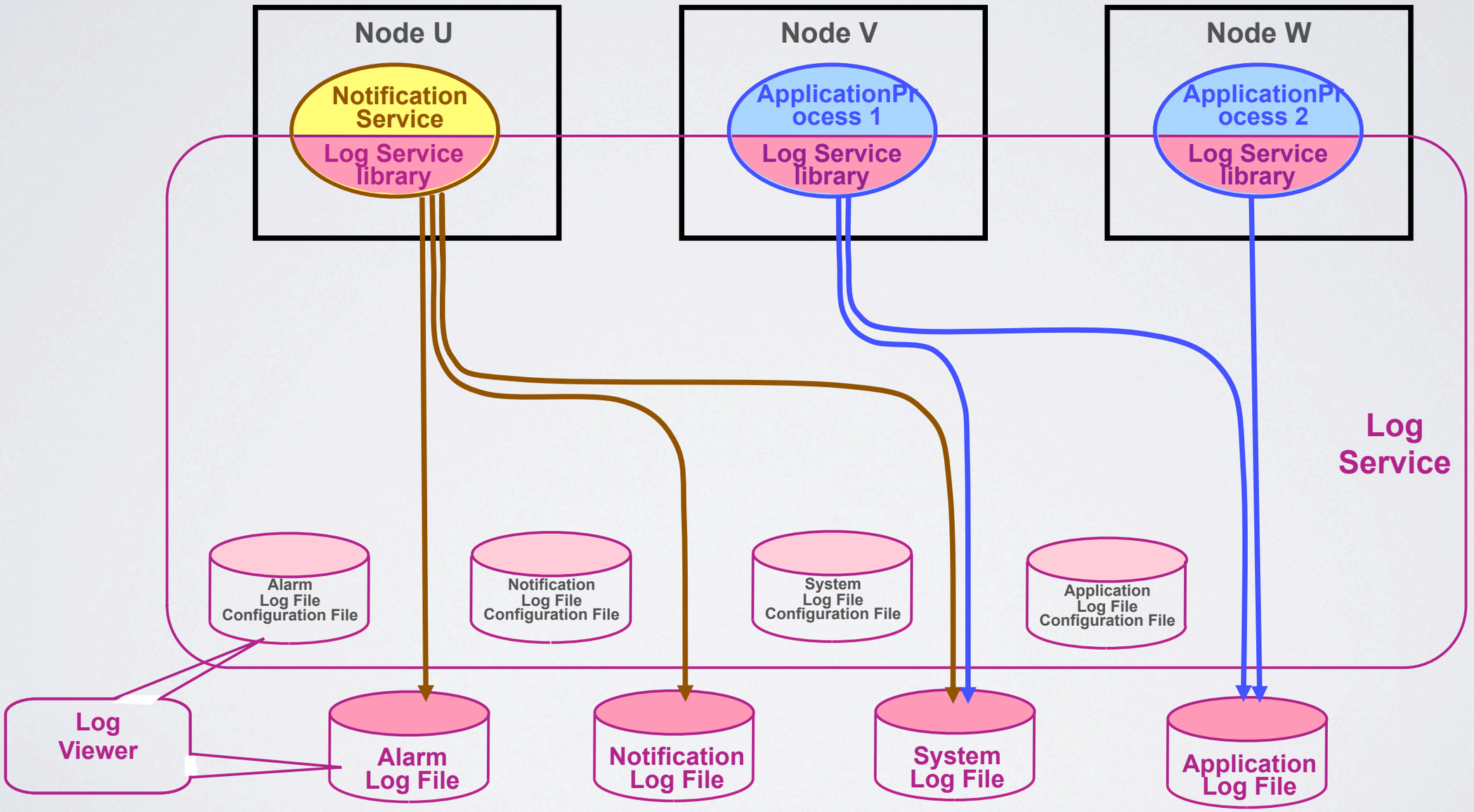
- The Log Service library, log files, and log file configuration files



Copyright© 2006 Service Availability™ Forum, Inc

LOG SERVICE

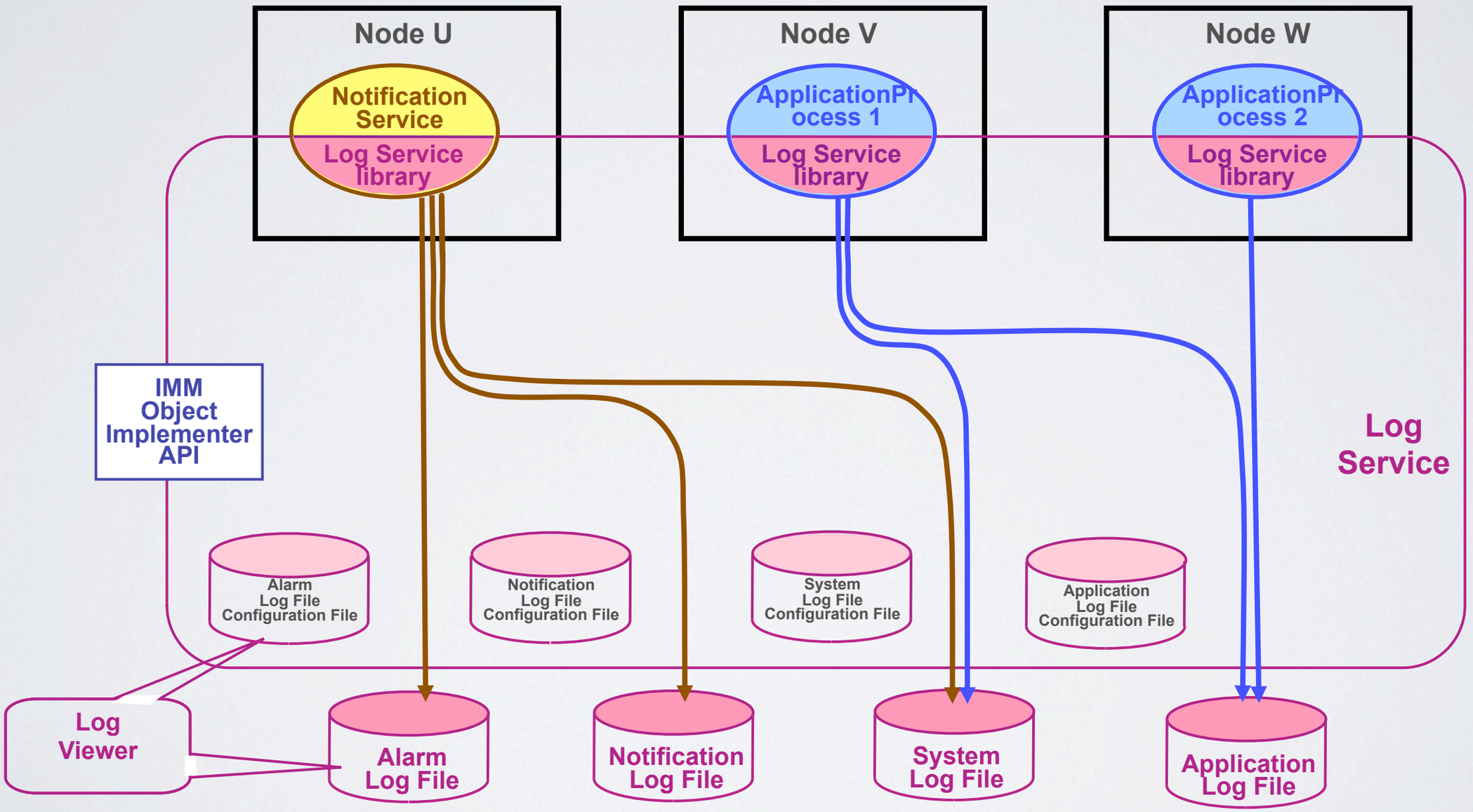
- The Log Service library, log files, and log file configuration files



Copyright© 2006 Service Availability™ Forum, Inc

LOG SERVICE

- The Log Service library, log files, and log file configuration files



Copyright© 2006 Service Availability™ Forum, Inc

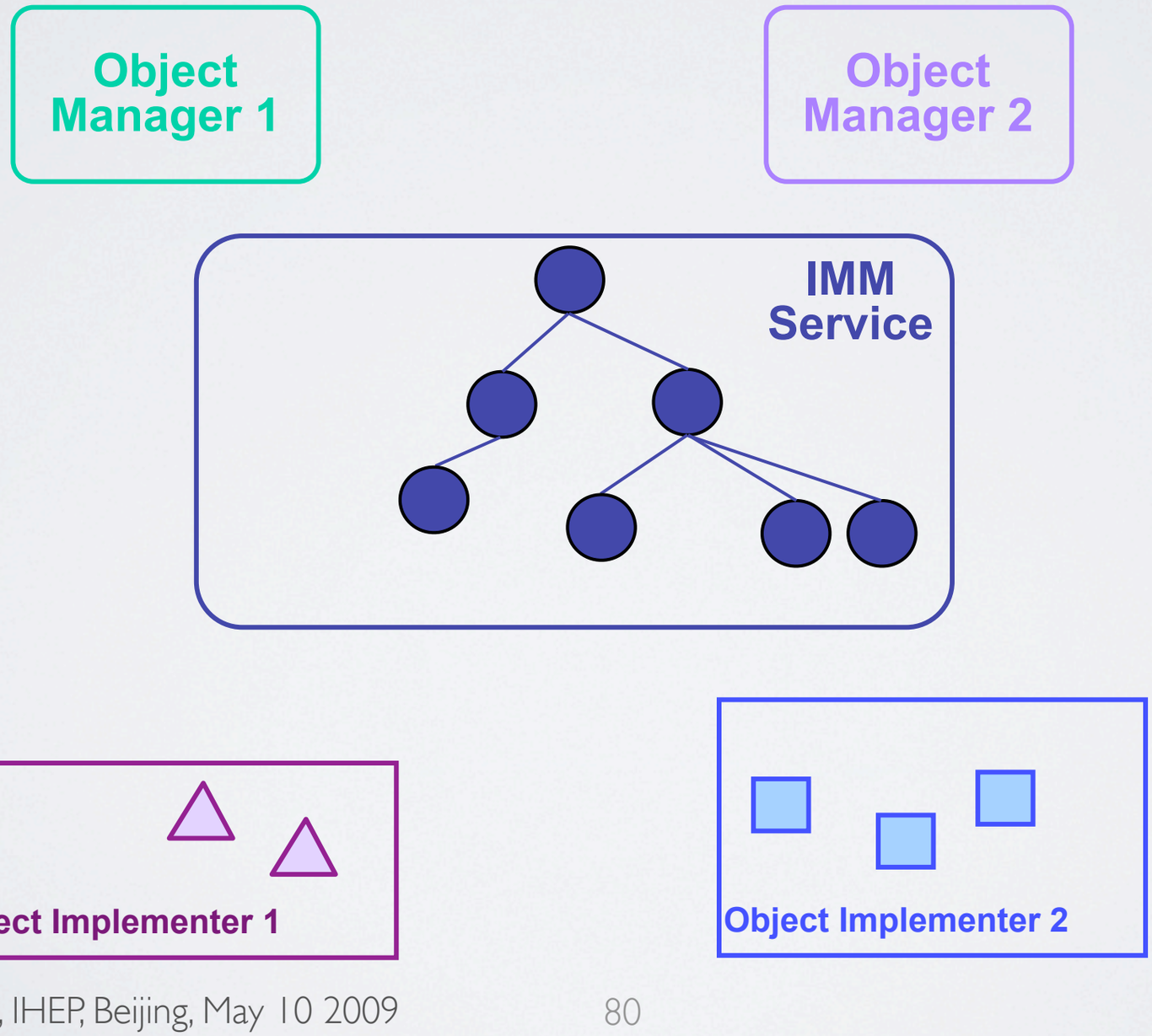
INFORMATION MODEL MANAGEMENT SERVICE

- The Information Model Management (IMM) Service allows objects of the Information Model (IM) to be created, accessed, and managed by system management applications
- Example of entities that can be represented by IM objects include:
  - Components managed by the Availability Management Framework
  - Checkpoints provided by the Checkpoint Service
  - Message queues provided by the Message Service
- The IM objects are provided with attributes and administrative operations (i.e., operations that can be performed on the represented entities through system management interfaces)
- The Information Model Management Service provides APIs that allow Object Managers, such as System Management applications, to create, access, and manage the IM objects
- Subsequently, the Information Model Management Service delivers the requested operations to the appropriate Object Implementers (e.g., AIS services or applications) that implement these objects for execution

Copyright© 2006 Service Availability™ Forum, Inc

INFORMATION MODEL MANAGEMENT SERVICE

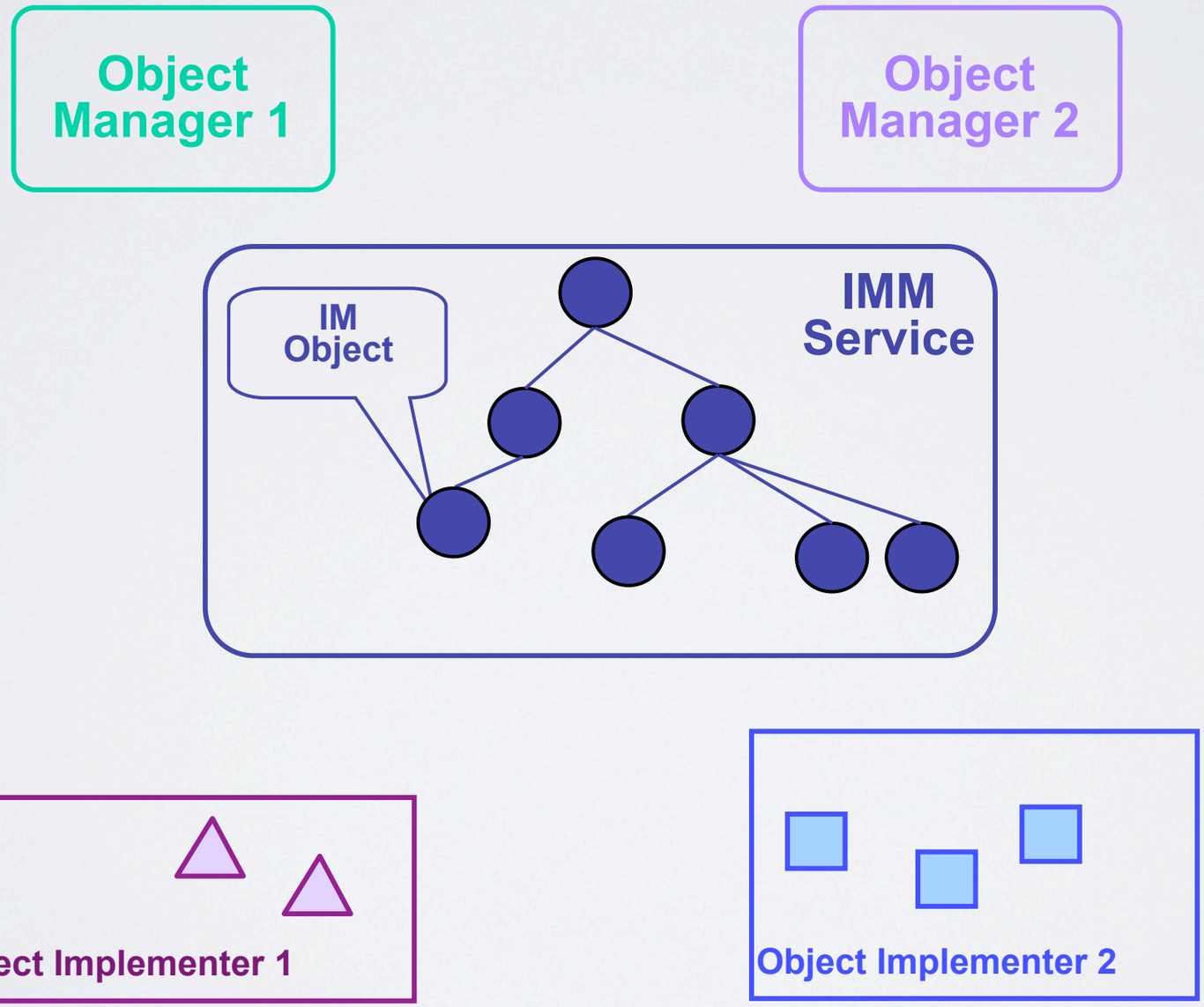
- The Information Model Management (IMM) Service, Information Management (IM) Objects, Object Managers, and Object Implementers



Copyright© 2006 Service Availability™ Forum, Inc

INFORMATION MODEL MANAGEMENT SERVICE

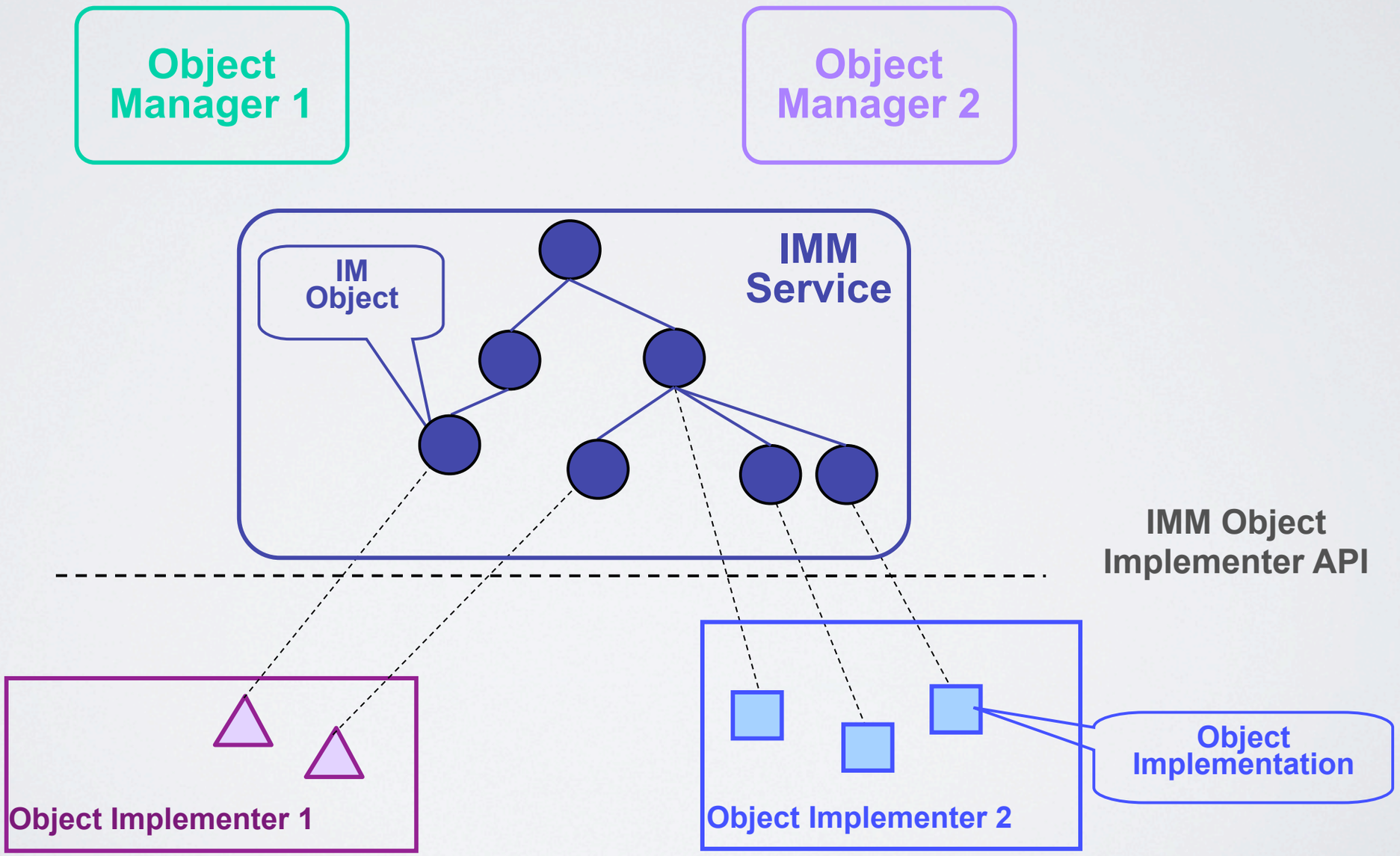
- The Information Model Management (IMM) Service, Information Management (IM) Objects, Object Managers, and Object Implementers



Copyright© 2006 Service Availability™ Forum, Inc

INFORMATION MODEL MANAGEMENT SERVICE

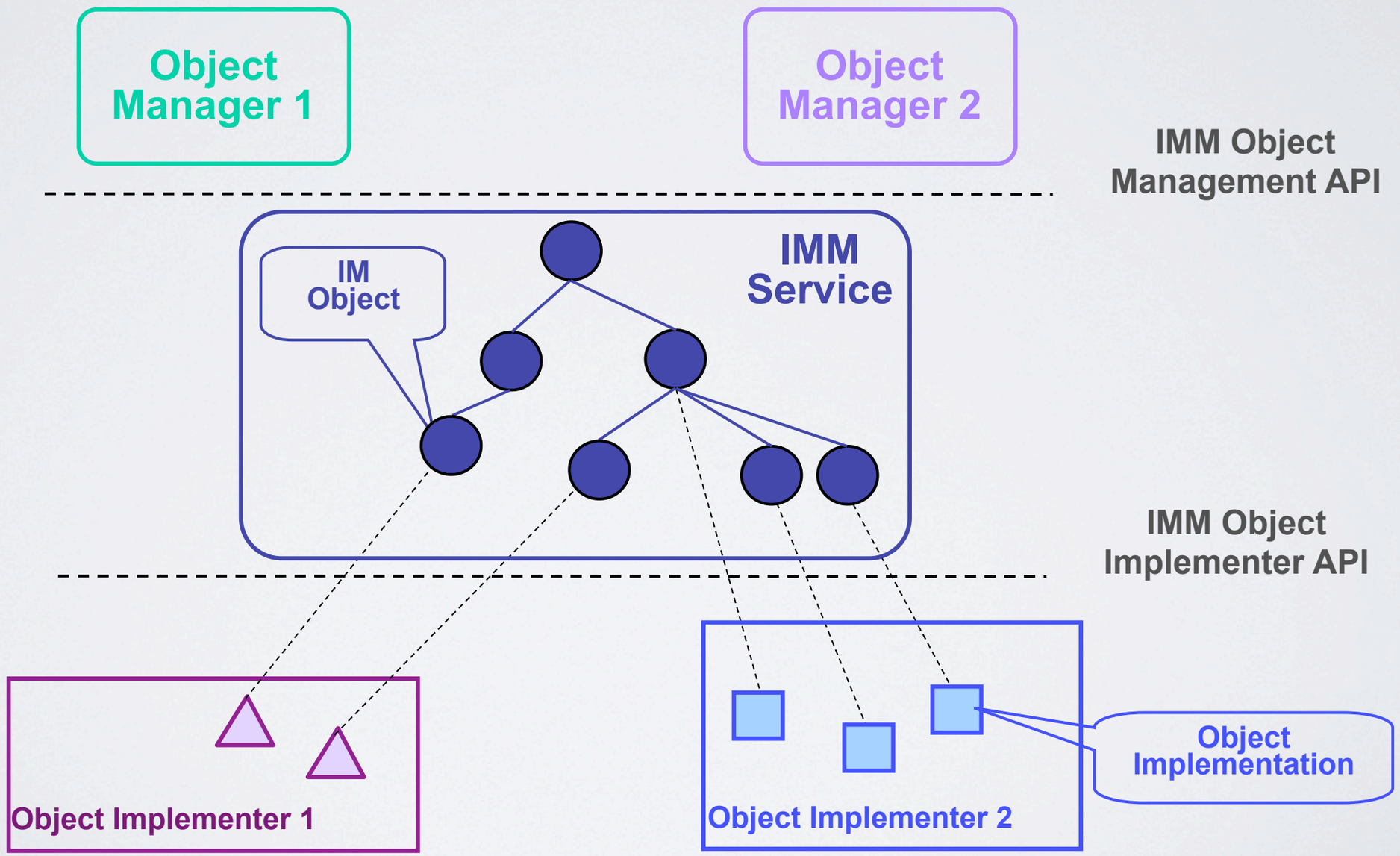
- The Information Model Management (IMM) Service, Information Management (IM) Objects, Object Managers, and Object Implementers



Copyright© 2006 Service Availability™ Forum, Inc

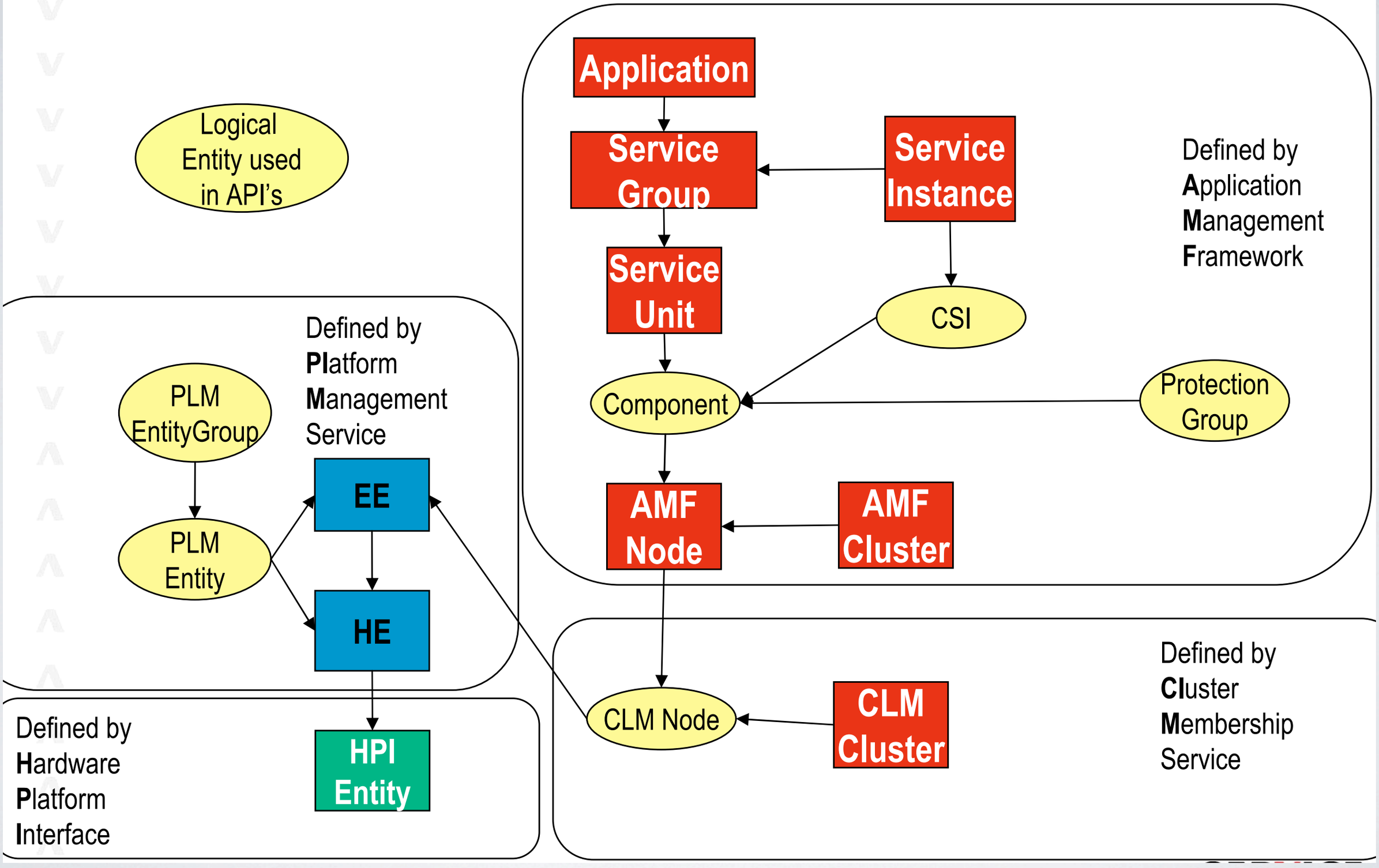
INFORMATION MODEL MANAGEMENT SERVICE

- The Information Model Management (IMM) Service, Information Management (IM) Objects, Object Managers, and Object Implementers



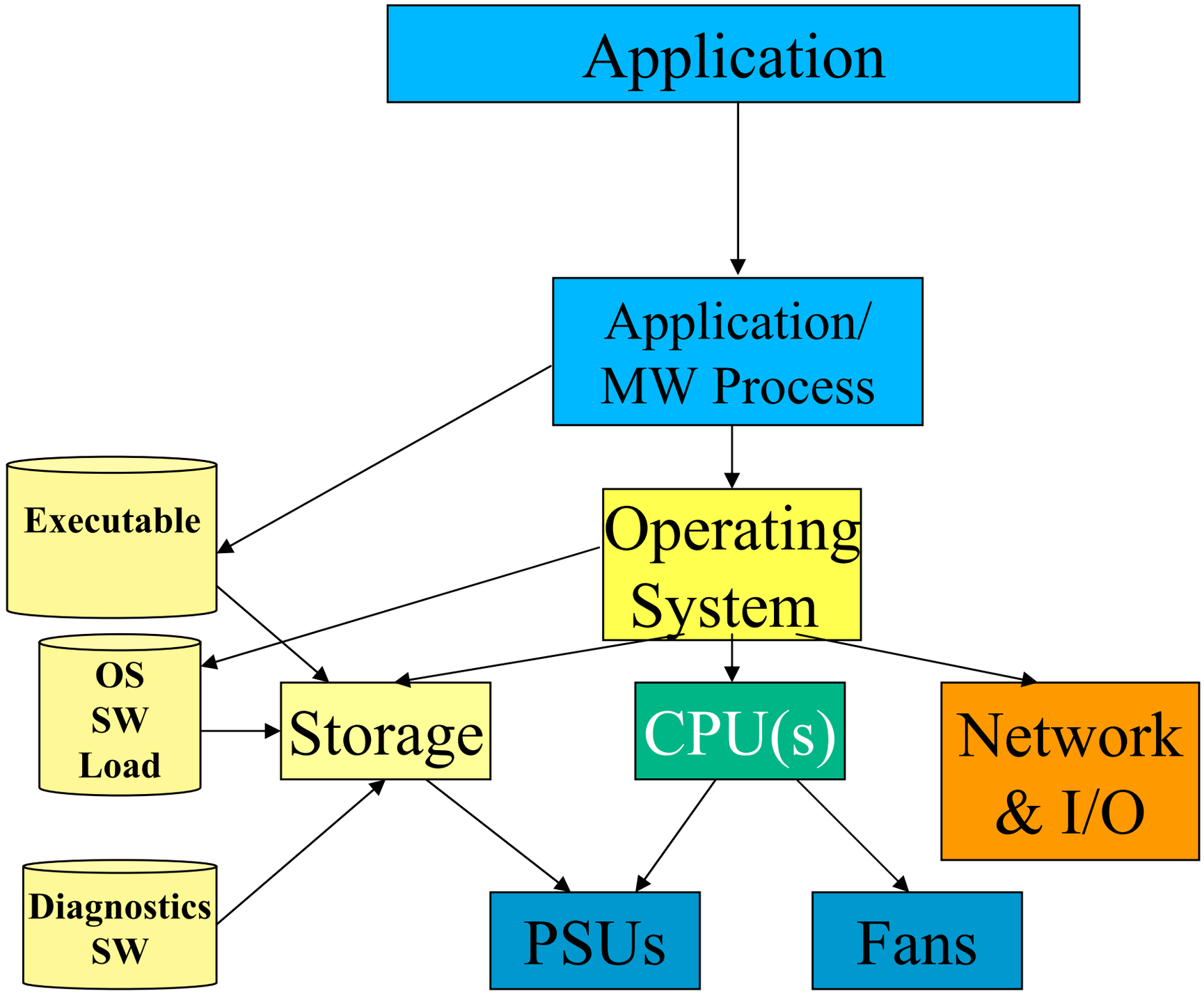
Copyright© 2006 Service Availability™ Forum, Inc

SA Logical Entities Dependency Relationships



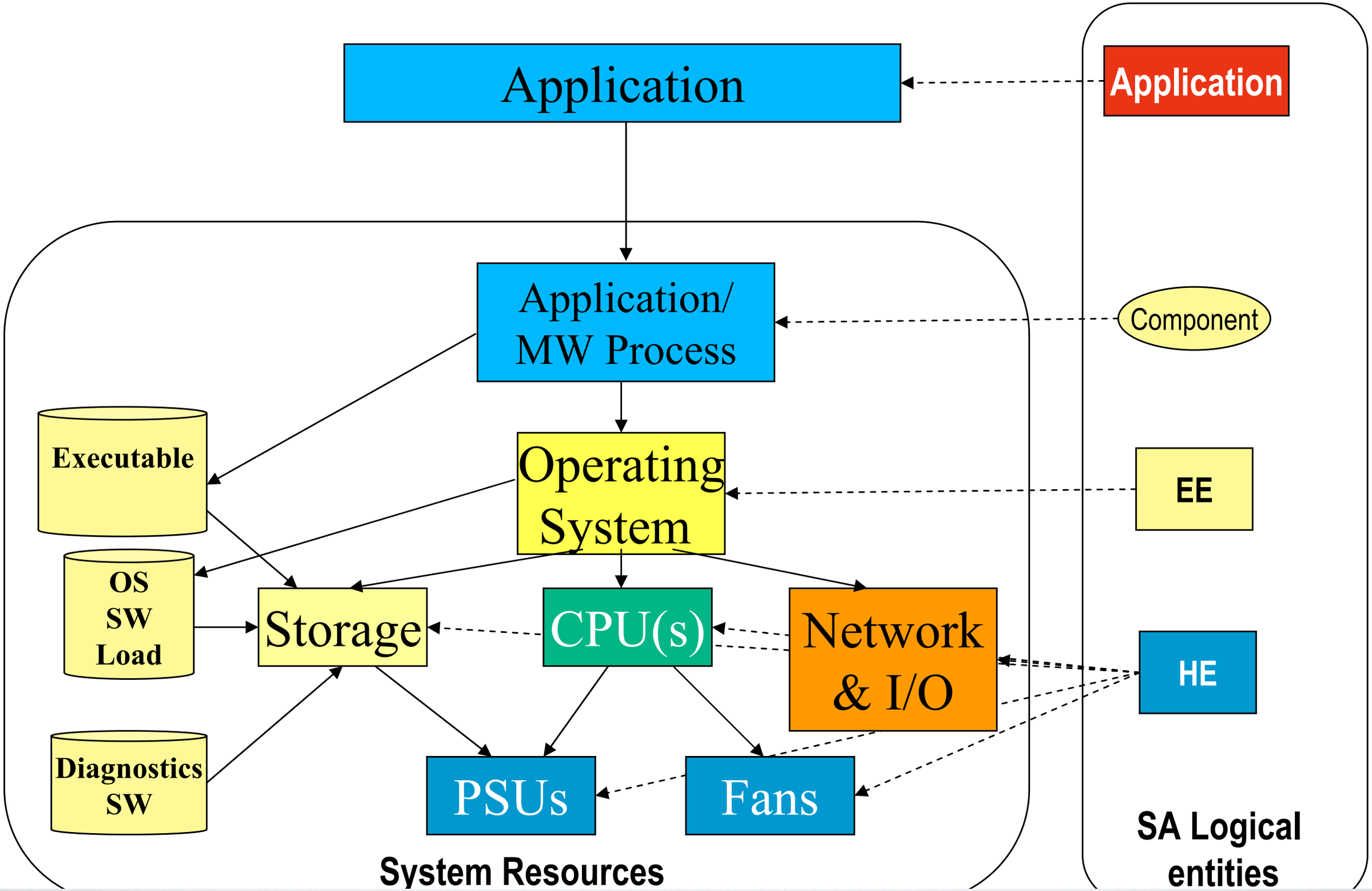
Courtesy of J. Fryer

Resource Dependency Graph



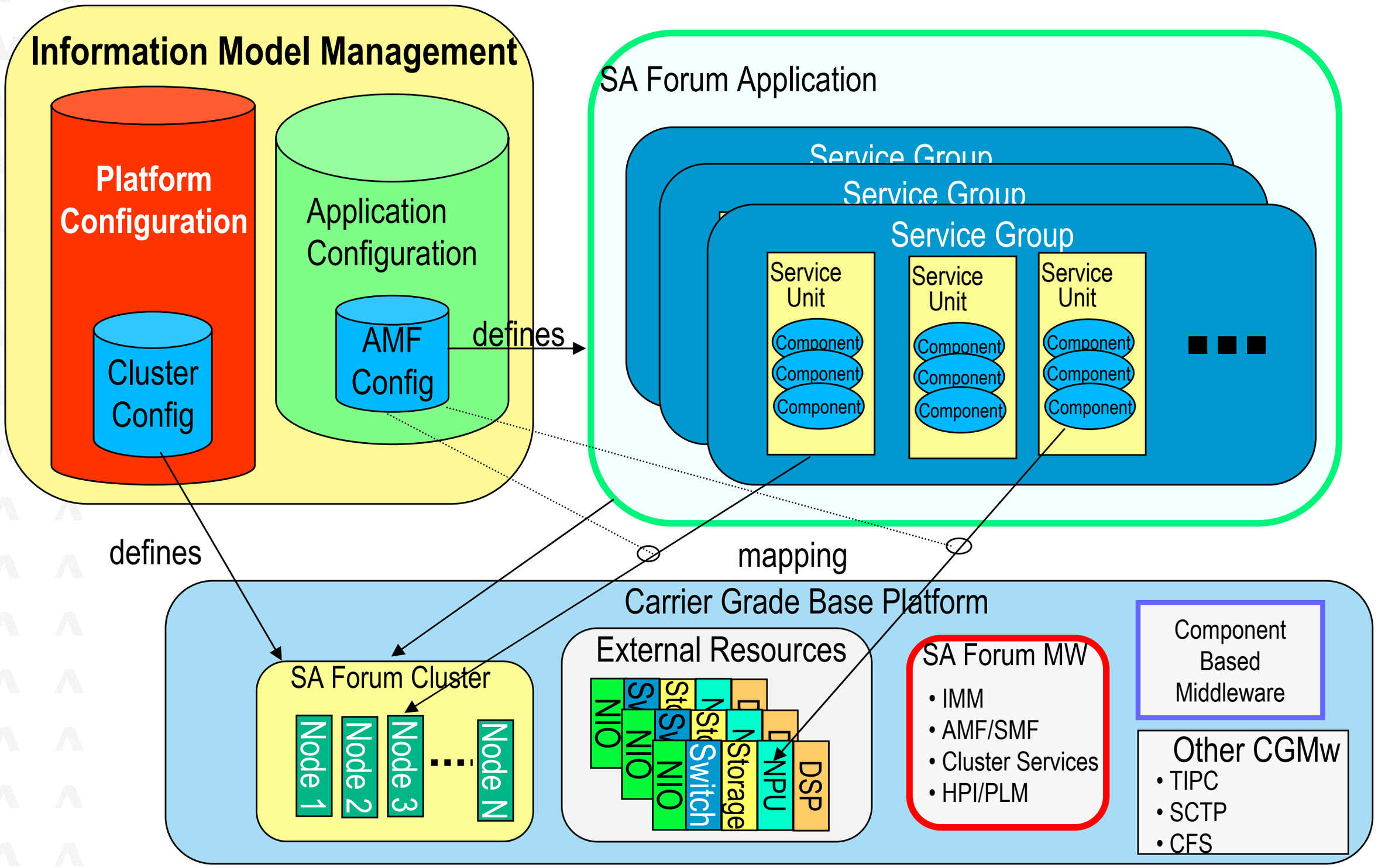
Courtesy of J. Fryer

Logical Entity to Resource Mapping

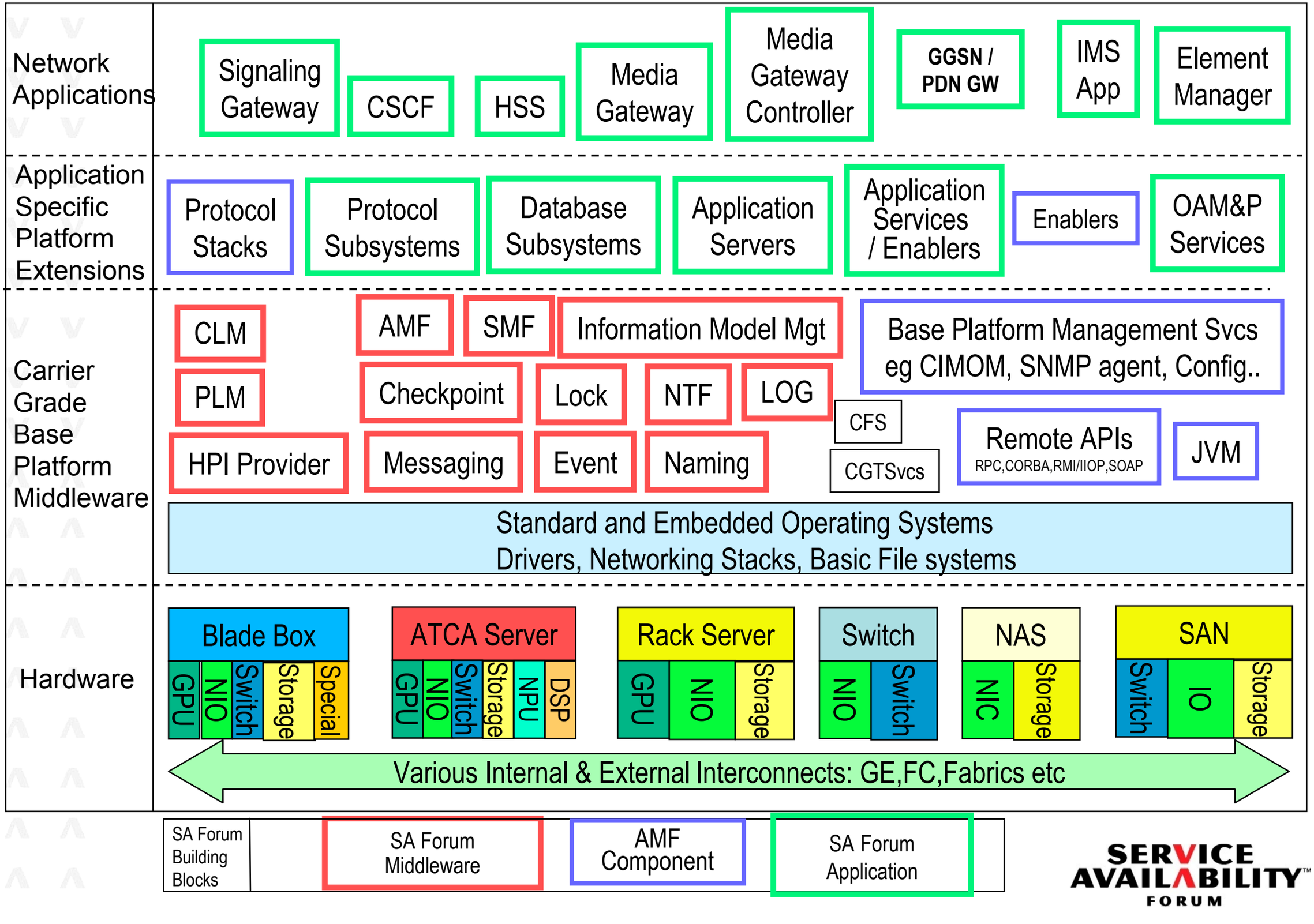


Courtesy of J. Fryer

SA Forum Building Blocks



SA Forum in a Carrier-Grade Base Platform



SERVICE AVAILABILITY™ FORUM

Courtesy of J. Fryer

OPEN SOURCE PROJECTS

- **OpenSAF is an Open Source Project established to develop a High Availability middleware base platform consistent with the Service Availability Forum™ (SA Forum) specifications**
  - www.opensaf.org
  - LGPL license

- **OpenClovis Application Service Platform is the run time middleware that serves as a foundation on which to build a carrier grade system. The OpenClovis middleware is also aligned with the SAForum's AIS and HPI specifications**
  - www.openclovis.org
  - GPL license

- **OpenAIS is an implementation of the Service Availability Forum Application Interface Specification (AIS)**
  - www.openais.org
  - new BSD license

OPEN SOURCE PROJECTS

- **OpenHPI, an open source implementation of HPI in Linux, is available at:**
  - http://openhpi.sourceforge.net
- **OpenHPI SNMP Subagent, an open source implementation of the sub-agent using HPI MIB, is available at:**
  - http://openhpi.sourceforge.net
- **Compliance Test Suite, used to certify that an implementation complies with the SA Forum specifications, is available at:**
  - http://saftest.sourceforge.net

SA FORUM CONTACTS

- **General Information:**
  - www.saforum.org
- **SA Forum Specifications**
  - www.saforum.org/specification
- **Joining the SA Forum**
  - www.saforum.org/join
- **Key Administration Contacts:**
  - Morgan Seibert
    - SA Forum Administration
    - Phone: 503.619.0576
    - admin@saforum.org
  - Lori Zielinski
    - SA Forum PR
    - Phone: 503.619.0852
    - marketing@saforum.org
    - SA Forum Contacts

SA FORUM DOCUMENTS

- **SA Forum Extended Training Material is available for download at:**
  - http://www.saforum.org/extended_training_material_download.html
- **SA Forum Specifications are available for download at:**
  - http://www.saforum.org/specification_download.html
- **More information can be found at:**
  - http://www.saforum.org/downloads/specification_white_paper.pdf

SUMMARY

- SIG Ecosystem, Role of open specifications
- Service Availability Forum
  - Mission, Architecture, High Availability, Service Continuty
- Hardware Platform Interface
  - Architecture, Domains, Sessions, Resources, Entities, HPI data stores
  - HPI usage example - EPICS redundant IOC on ATCA
- Application Interface Specification
  - Architecture, Information Model
  - Application Management Framework, Cluster Membership, Checkpoint, Message, Event, Lock, Notification, Log, IMM Services

THANK YOU!