SERVICE AVAILABILITY™ FORUM

Open Specifications for Service Availability

Introduction to the Service Availability Forum

Contents

- ***Introduction***
- ***Quick AIS Specification overview***
- ***AIS Dependability services***
- ***AIS Communication services***
- ***Programming model***
- ***DEMO***

SERVICE AVAILABILITY™ FORUM

Design of dependable services

- ***Two parts of functionality***
  - Business logic
    - Implements the service
  - Common functionality
    - Communication, management, fault tolerance…
    - **This is where SA Forum AIS comes into picture**

Construction of a service

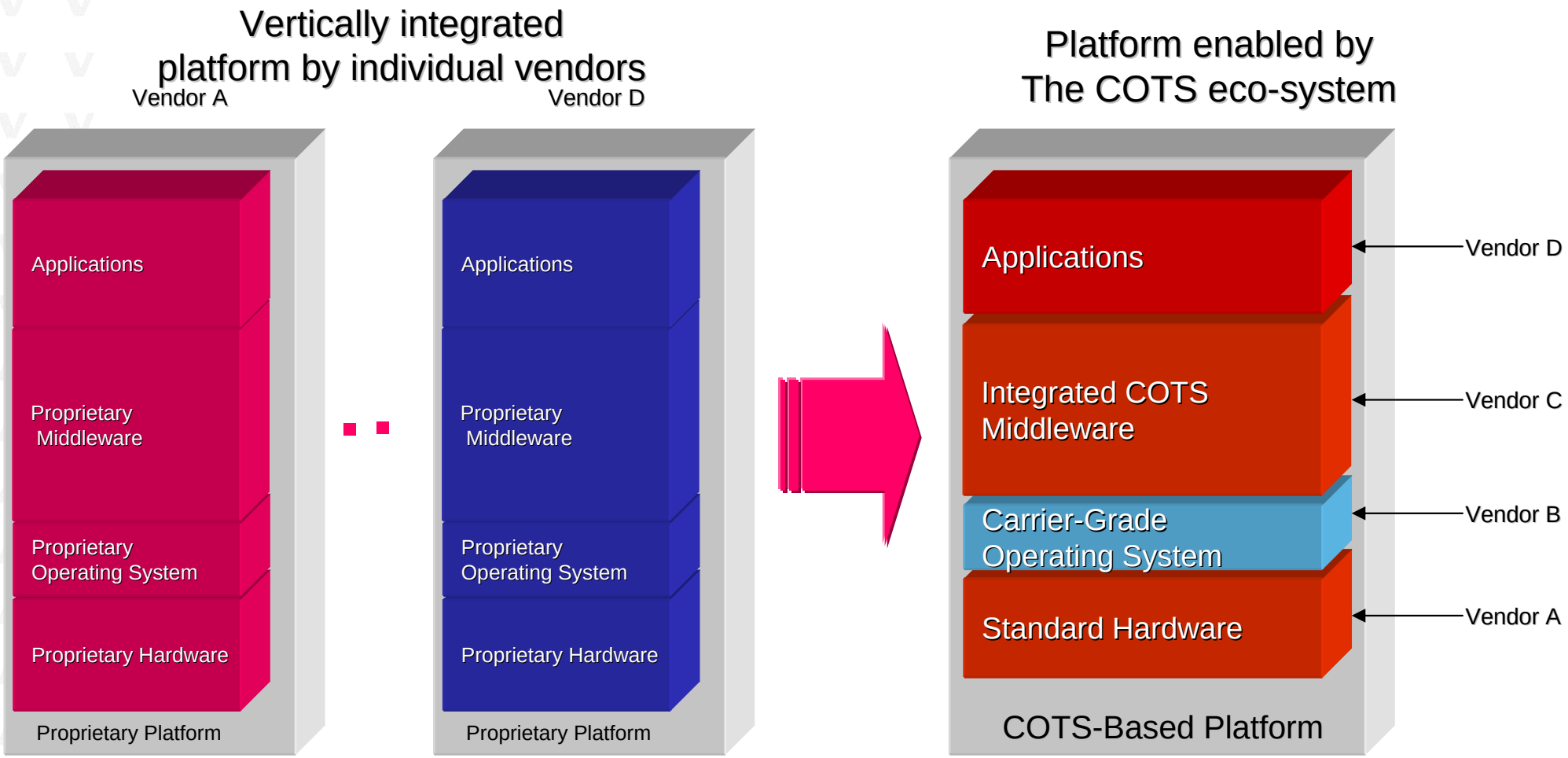
- ***Define the functionality***
- ***Plan the architecture***
  - Components, roles
- ***Integrate fault management***
  - Service state monitoring
  - Management
- ***Plan recovery***
- ***Communication between components***

Construction of a service

- ***Define the functionality***
- ***Plan the architecture (AMF)***
  - Components, roles
- ***Integrate fault management (AMF)***
  - Service state monitoring
  - Management
- ***Plan recovery (CKPT)***
- ***Communication between components (MSG, EVT)***

The Transition

COTS adoption is accelerating transition from vertical to horizontal industry model



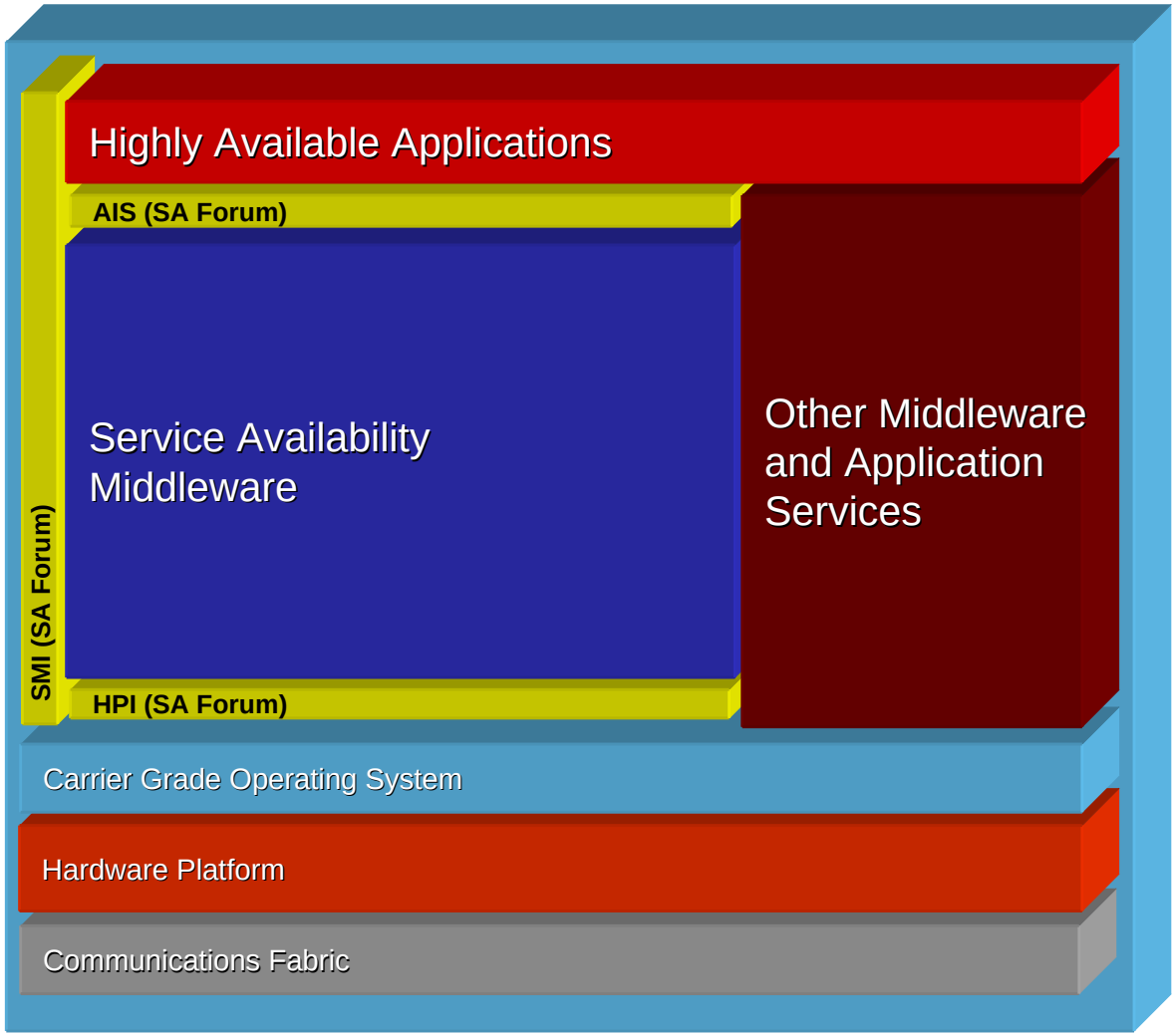
SERVICE AVAILABILITY™ FORUM

SA Forum Standard Interface Specifications

Application Interface Specification (AIS)

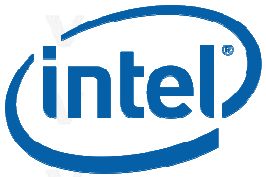
Hardware Platform Interface (HPI)

Systems Management Interfaces (SMI)



SERVICE AVAILABILITY™ FORUM

SA Forum Members



MOTOROLA

ERICSSON

RadiSys
THE POWER OF WE

SYBASE

TietoEnator
Building the Information Society

EMERSON
Network Power
Formerly Artesyn Technologies

HUAWEI

montavista

solid.

Continuous Computing

Industrial Technology Research Institute

PIGEON POINT SYSTEMS

Sun microsystems

OpenClovis

Alcatel·Lucent

tail-f

ENEA

WIND RIVER

GOAHEAD

ESO TECHNOLOGIES

IBM

NTT

NEC Empowered by Innovation

PERFORMANCE TECHNOLOGIES

ORACLE

NORTEL NETWORKS™

Sonus NETWORKS

hp invent

FUJITSU COMPUTERS SIEMENS

NOKIA
CONNECTING PEOPLE

FUJITSU THE POSSIBILITIES ARE INFINITE

embeddedMIND™

SERVICE AVAILABILITY™ FORUM

THE AIS SPECIFICATION

Using SA Forum's specifications to build highly available services

SERVICE AVAILABILITY™ FORUM

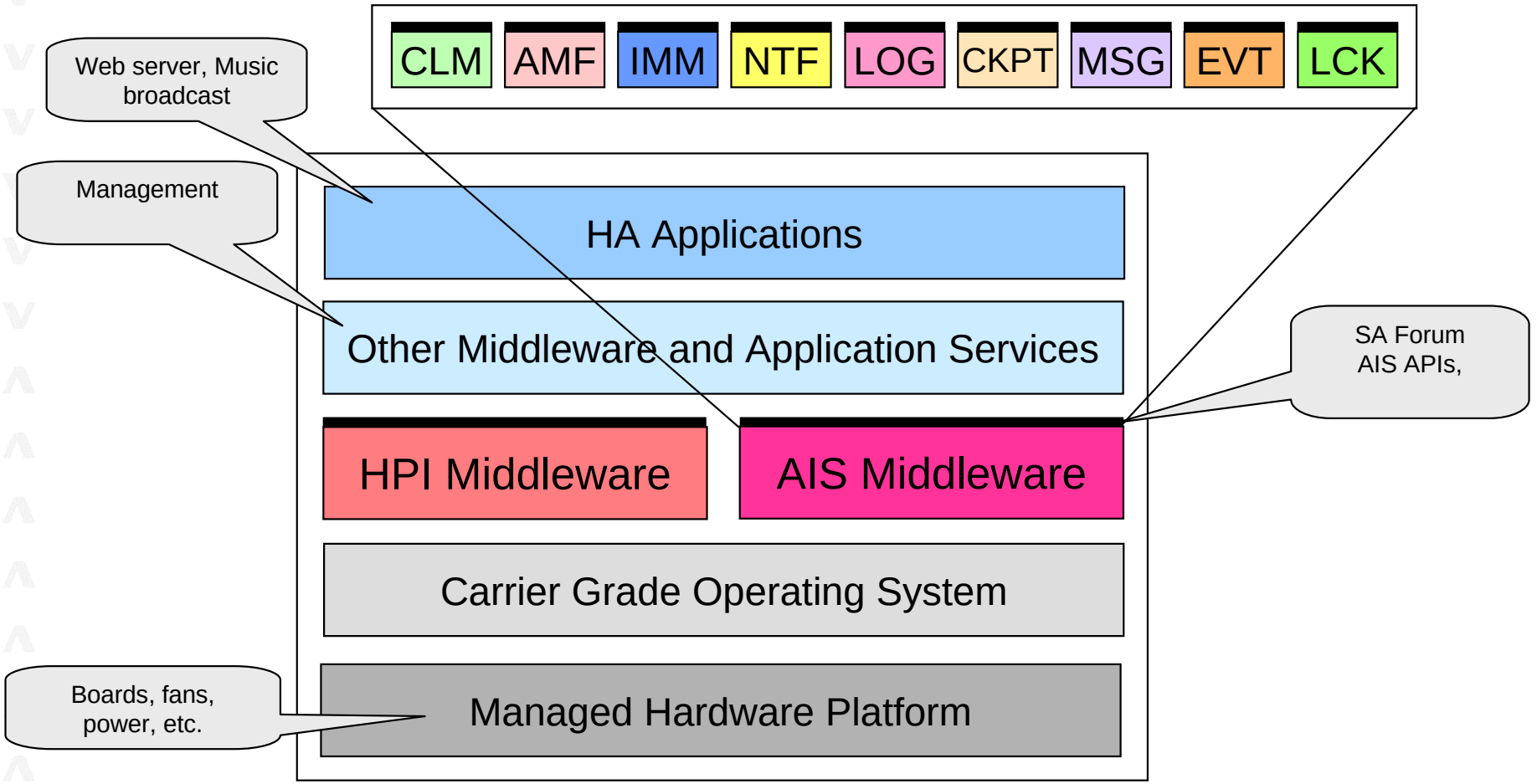
Application Interface Specification

- ***The Application Interface Specification (AIS) is a set of open standard interface specifications***

- ***The AIS specifies***
  - the Application Programming Interface (API) for HA middleware services
  - service entities and their behavior (life cycle, administrative operations, functionality)

- ***Example for specification/implementation***
  - HW interface: SATA
  - HW implementation: manufactured mainboard
  - HW user: the HDD (Samsung, Western Digital, Seagate, Hitachi…)

Application Interface Specification



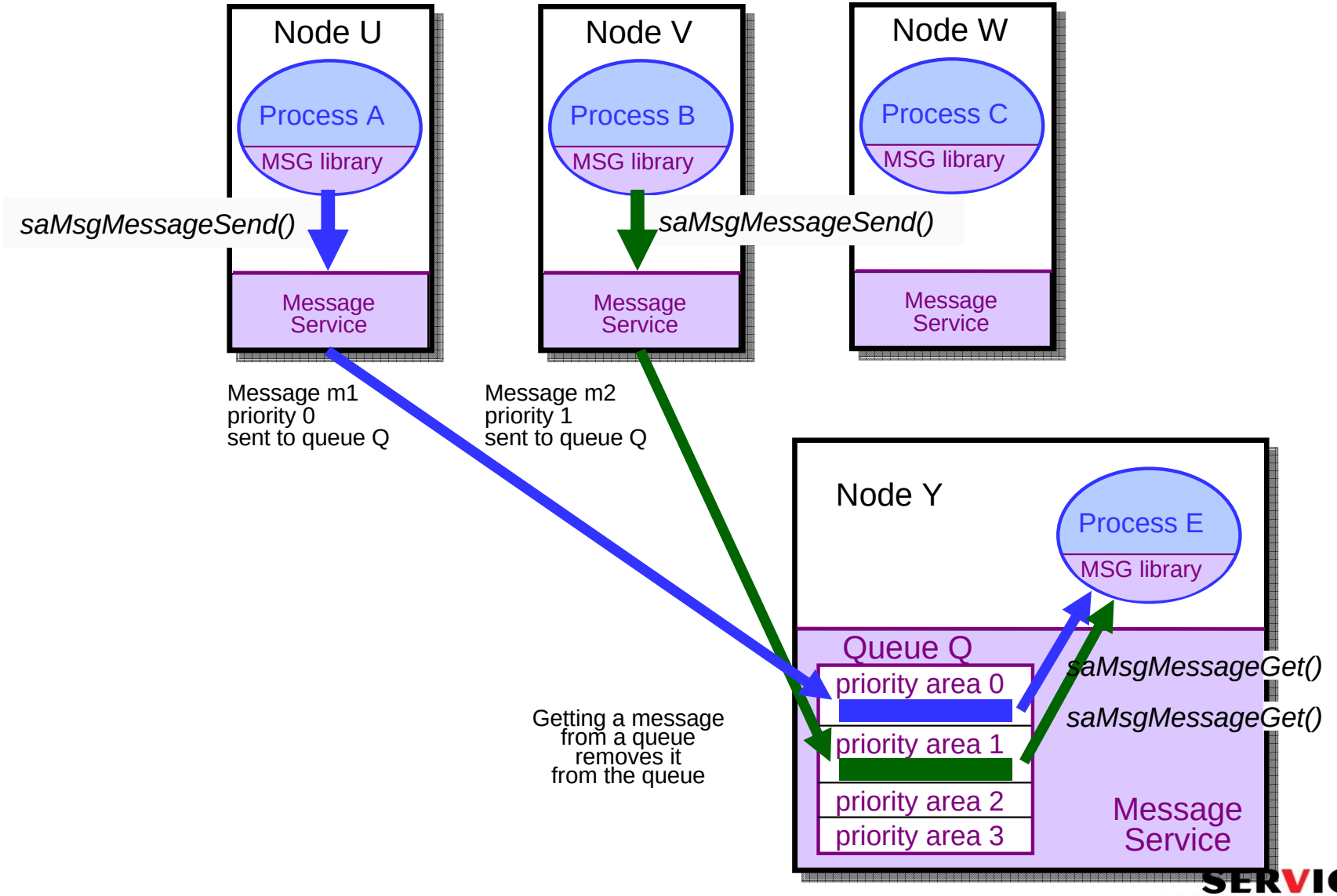
SERVICE AVAILABILITY™ FORUM

Application Interface Specification

- ***The AIS is divided into the following parts or areas***

- ***Administration***
  - Software Management Framework
  - Information Model Management Service
  - Cluster Membership Service
  - Notification Service
- ***Dependability***
  - Availability Management Framework
  - Checkpoint Service
- ***Communication***
  - Event Service
  - Message Service
- ***Miscellaneous***
  - Lock Service
  - Log Service

Example service - Message Queues



THE AIS SPECIFICATION

Dependability Services

SERVICE AVAILABILITY™ FORUM

SA Forum's approach to making applications HA

- ***Availability Management Framework principles***
  - Integrate best practices into the middleware,
  - Provide means for the software developer to influence behavior
  - Let the middleware control the application (like a Marionette)

SERVICE AVAILABILITY™ FORUM

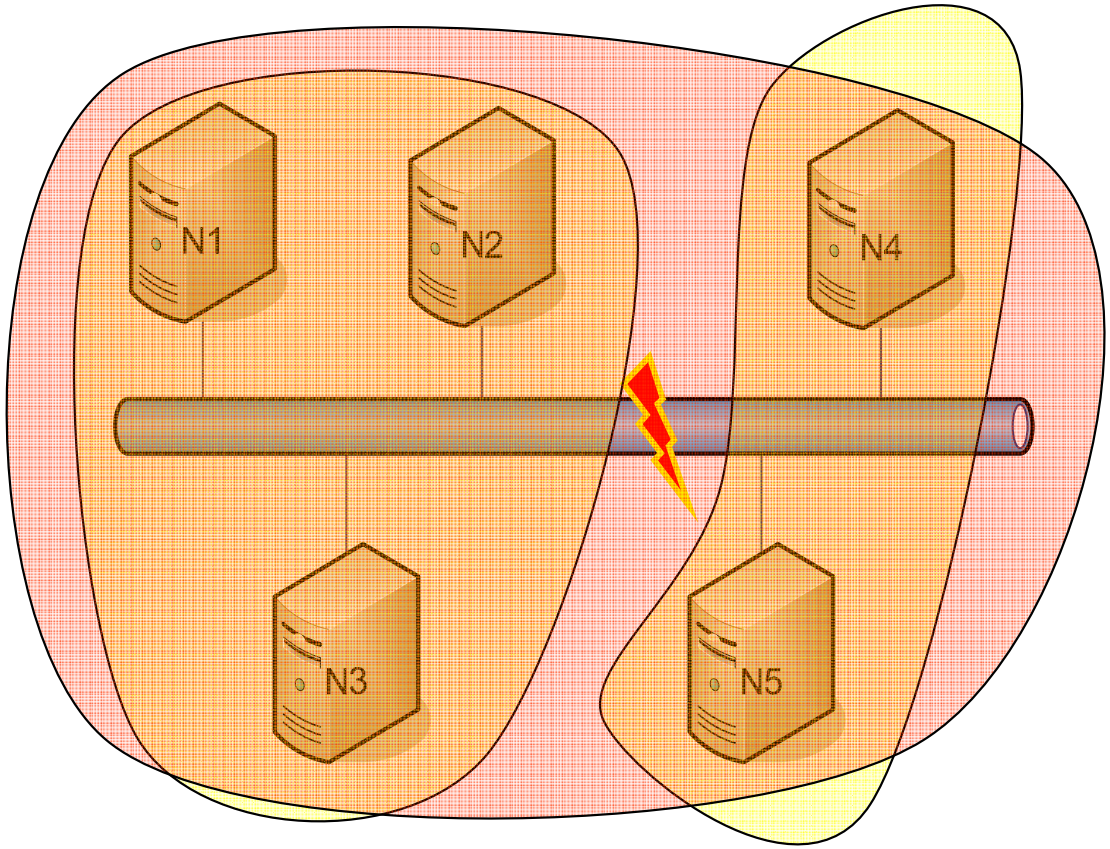
SA Forum's approach to making applications HA

- **Fault prevention**
  - Reduce the probability of system failure to an acceptably low value
- **Fault tolerance**
  - Provide service in spite of faults
- **Fault removal**
  - Diagnostics, monitoring, repair at development and runtime
- **Fault forecasting / prediction**
  - Estimating failures and their effects
  - E.g. software aging

SERVICE AVAILABILITY™ FORUM

Server Clustering

- ***Notions***
  - node
  - link
  - cluster
  - partition
- ***Functionalities of cluster middleware***
  - Error detection
  - Error handling
  - Notification
  - Reconfiguration



SERVICE AVAILABILITY™ FORUM

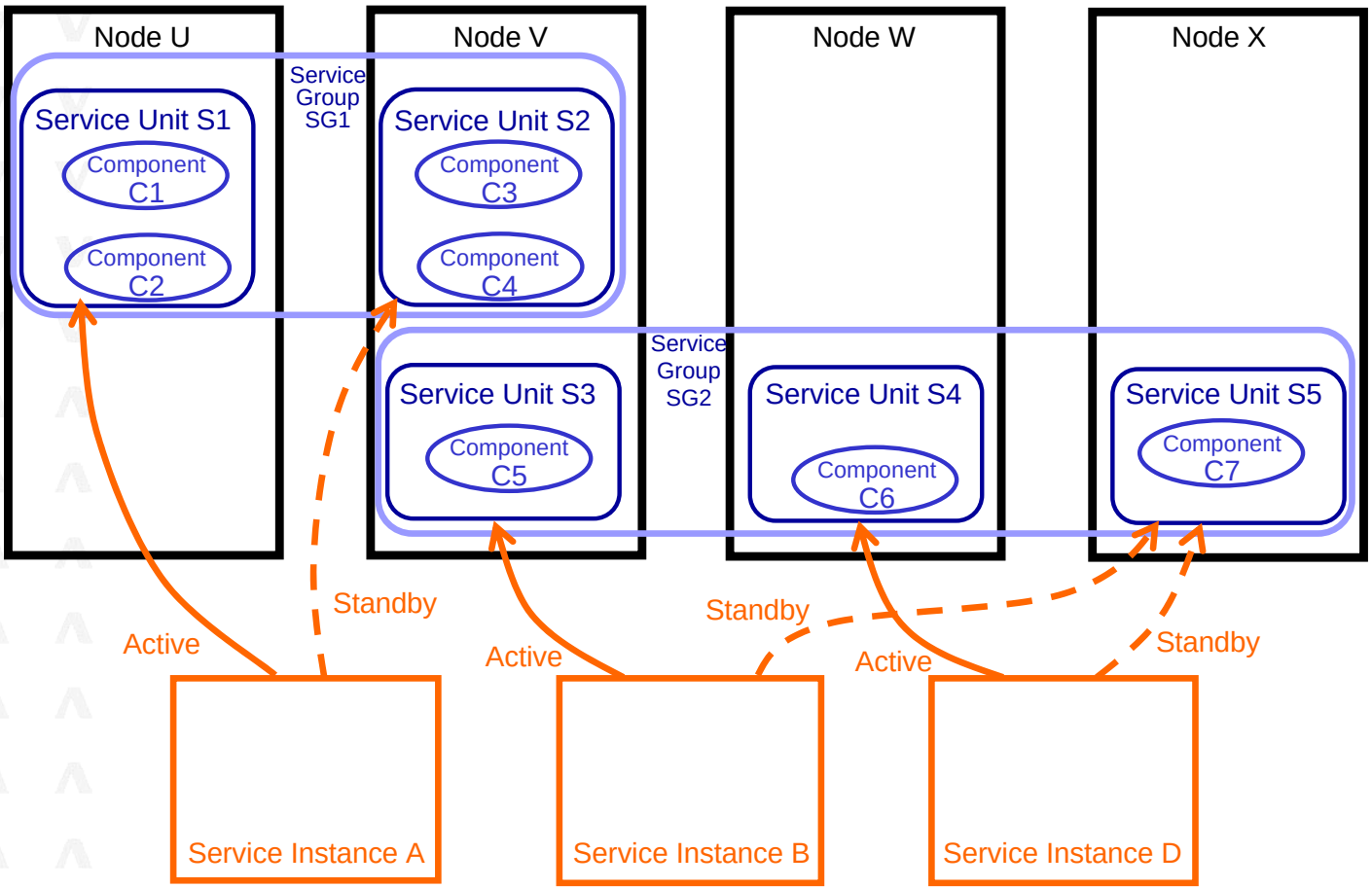
Server Clustering 2.

- ***Categories of clusters***
  - HA clusters
    - Improve the availability of services
  - Load-balancing clusters
    - Share the workload among the nodes
  - High-Performance Clusters (HPC)
    - Scientific computing
  - Grid computing
    - Many independent jobs

HW and SW based Fault Tolerance

- ***HW***
  - Redundant power supply
- ***SW***
  - Process replicas, failover, switchover…
- ***Clusters – hybrid solutions***
  - Process replicas on different nodes

AMF System Model Example



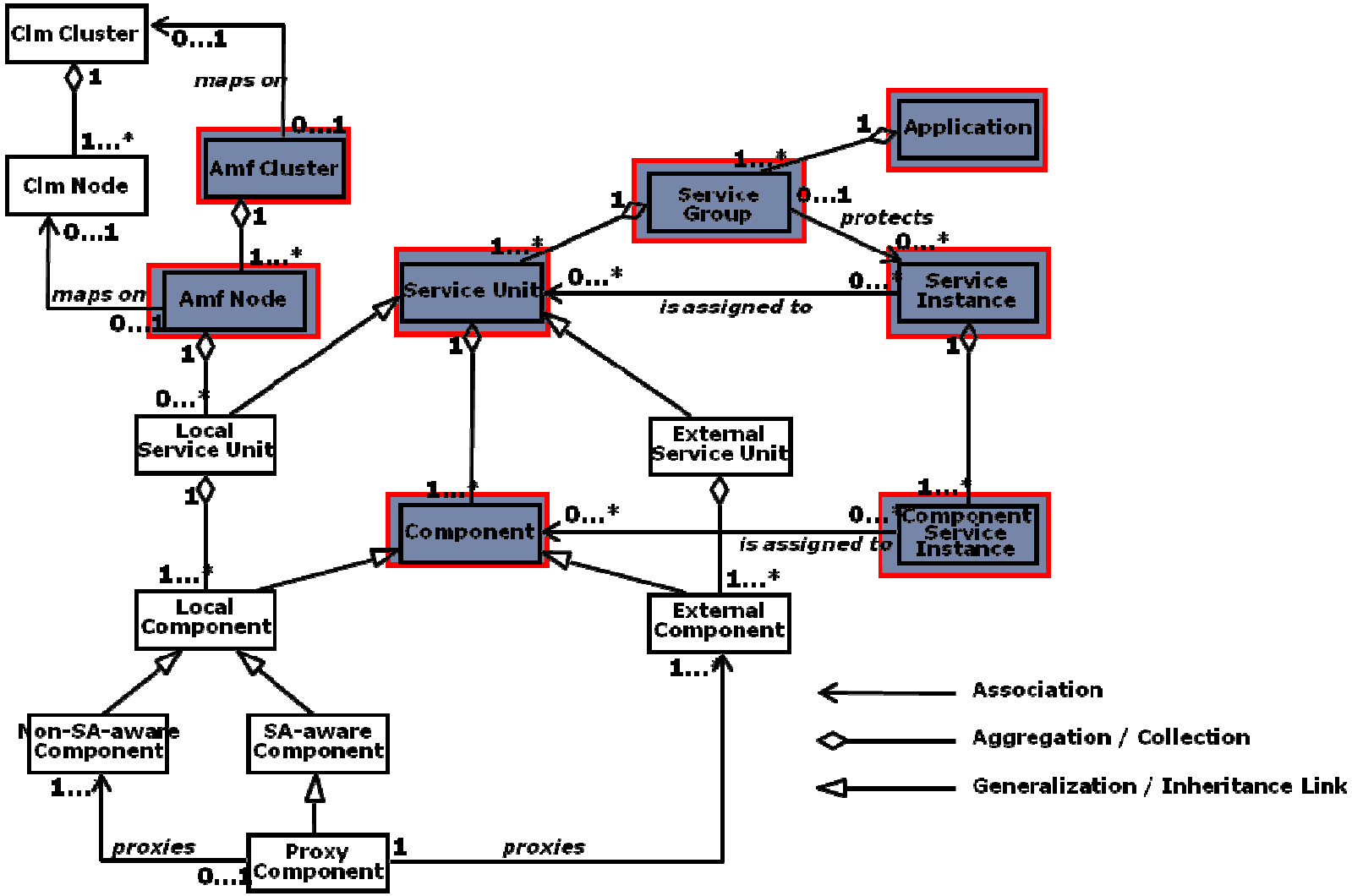
Service group SG1 supports a single service instance A, and service group SG2 supports two service instances B and D

On behalf of A, service unit S1 is assigned the active HA state and service unit S2 is assigned the standby HA state

Service unit S1 contains two components C1 and C2, and service unit S2 contains two components C3 and C4

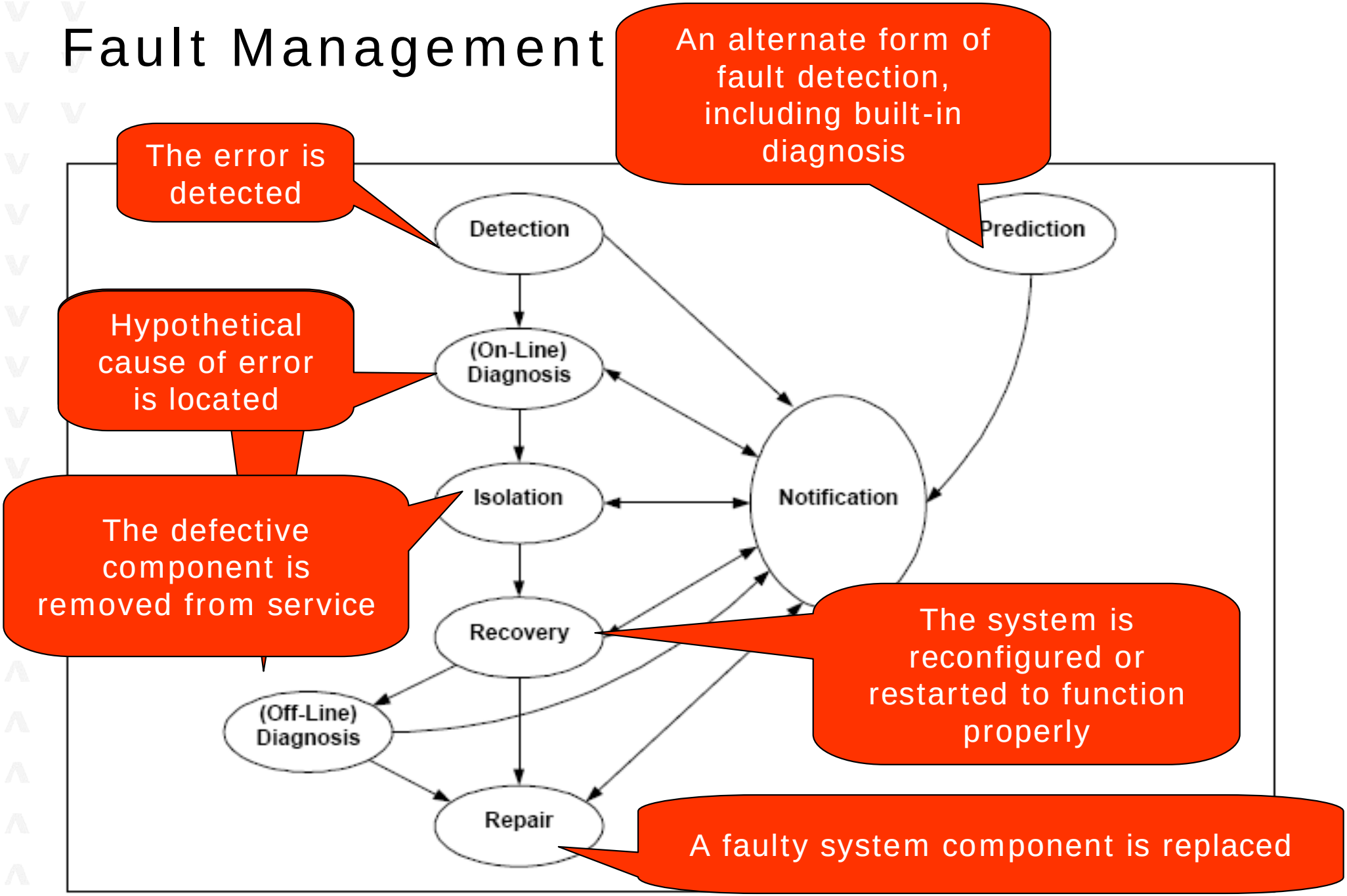
Similarly, for service group SG2

The SA Forum Information Model



SERVICE AVAILABILITY™ FORUM

Fault Management



Error detection methods

- ***Mechanisms***
  - Interface check (e.g. illegal instruction, illegal parameter, insufficient access rights)
- ***Ad-hoc methods***
  - Acceptability testing
  - Error checking codes
  - Timing checks (watchdog)
  - Diagnostic analysis (in idle time)
  - Substitute back (integrate → derive)

AMF error detection mechanisms

Needs the change of the component

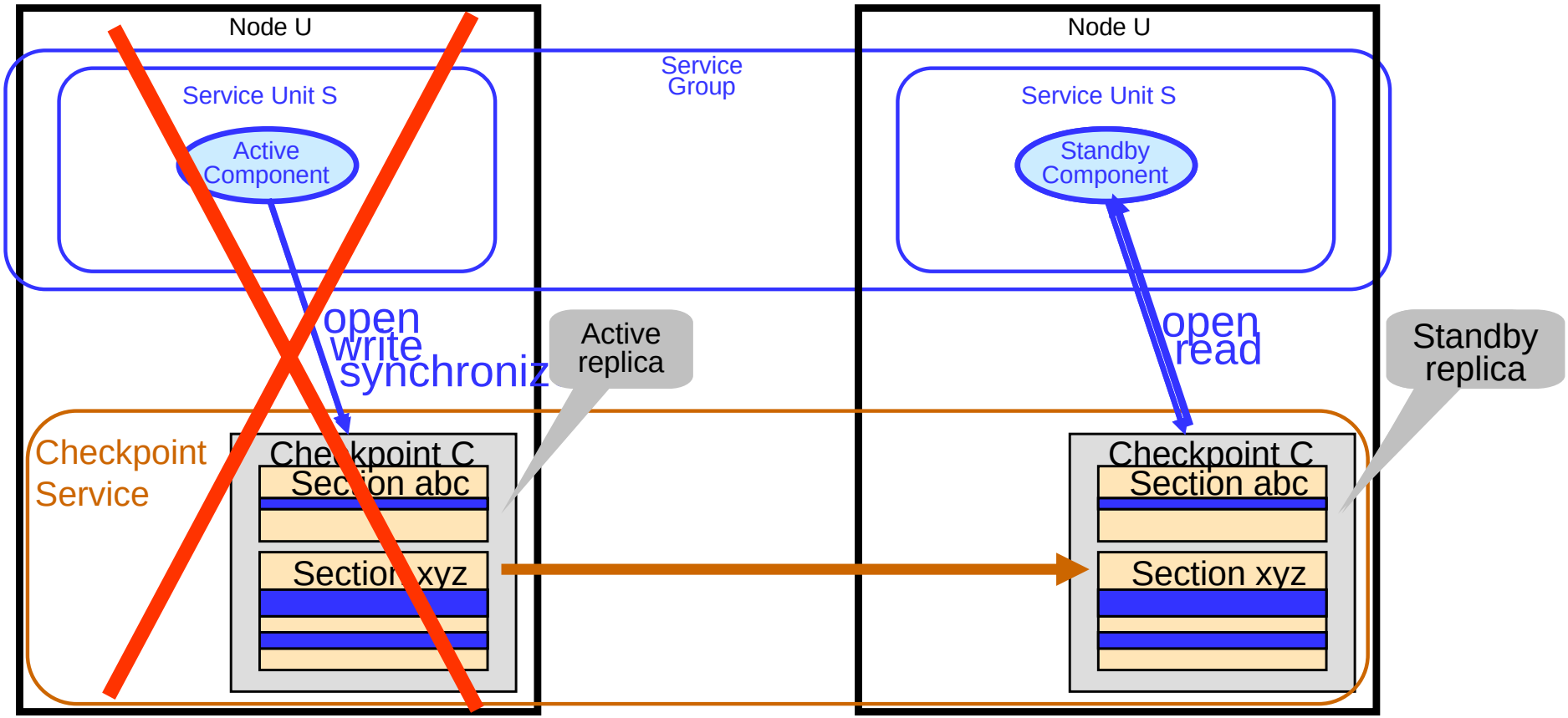
- ***Passive monitoring*** ✗
  - Using OS functionality
  - Currently only crash of a process
- ***External active monitoring*** ✗
  - External entity is used to monitor
- ***Internal active monitoring*** ✓
  - Healthchecks
  - Types
    - Pull: AMF invoked
    - Push: Component invoked

Planning recovery - checkpointing

- ***Define the variables to be saved***
  - State variables
- ***Normal operation***
  - Open checkpoint
  - Save variables regularly
- ***On failure***
  - Read checkpoint
  - Continue normal operation

Checkpoint Service

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



SERVICE AVAILABILITY™ FORUM

Methods of recovery

Goal: Recovery from errors before failures.

- ***Trivial methods***
  - Retry operation
  - Restart the application (cold, warm)
  - Restart the system
- ***Techniques***
  - Forward recovery
  - Backward recovery
  - Compensation

Service continuity

Goal: maintain the service continuity

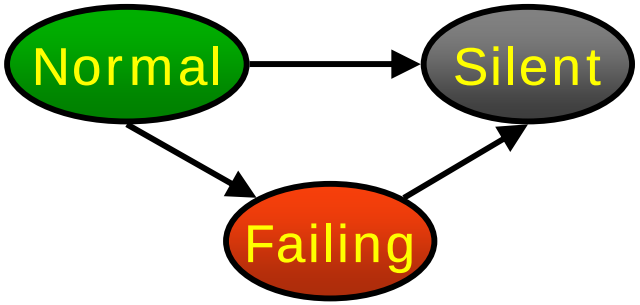
Type of error	Related actions
Transient	•Ignore (recovery handles them)
Permanent	•Reconfiguration •Failover •Switchover •Fail-back •“Graceful degradation”

Actions

- ***Failover***
  - The service is removed from the failed entity and assigned to a healthy entity
- ***Switchover***
  - The service is removed from a healthy entity to another healthy entity
- ***Fail-back***
  - The failed entity is repaired and service is assigned to it again
- ***Graceful degradation***
  - Service is carried on in a degraded state, probably with degraded functionality

System level error handling patterns

- ***Fail-silent***
  - If failed then no messages are sent out until repaired
  - E.g. in distributed systems



- ***Fail-stop***
  - Stop operation on failure
  - E.g. train traffic control systems (turn all lights red if failure happens)

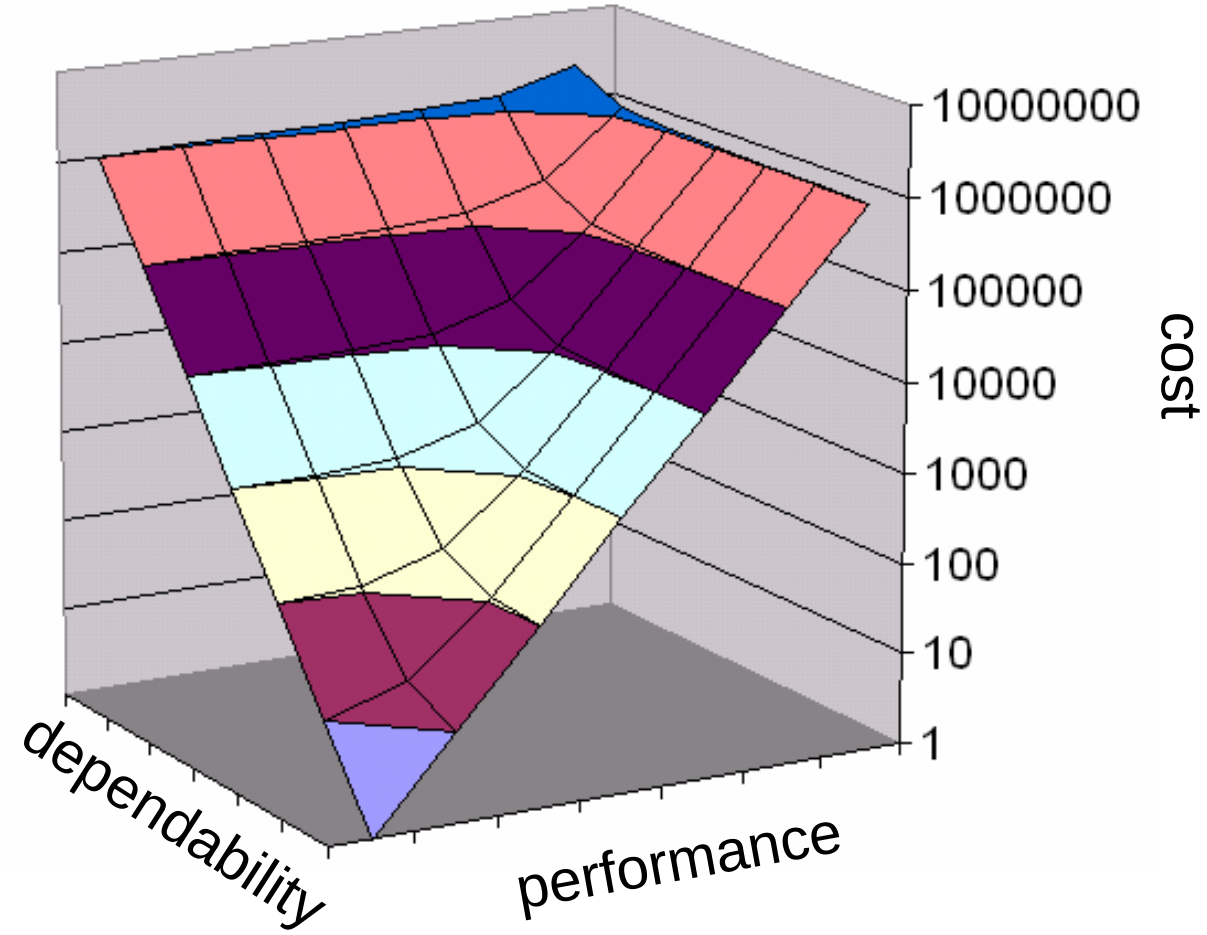
Francisco V. Brasileiro et. al. Implementing Fail-Silent Nodes for Distributed Systems (1996)

Requirements in today's systems

- ***Cost efficiency***
- ***Flexibility***
- ***Dependable services***
  - Highly Available
  - Reliable

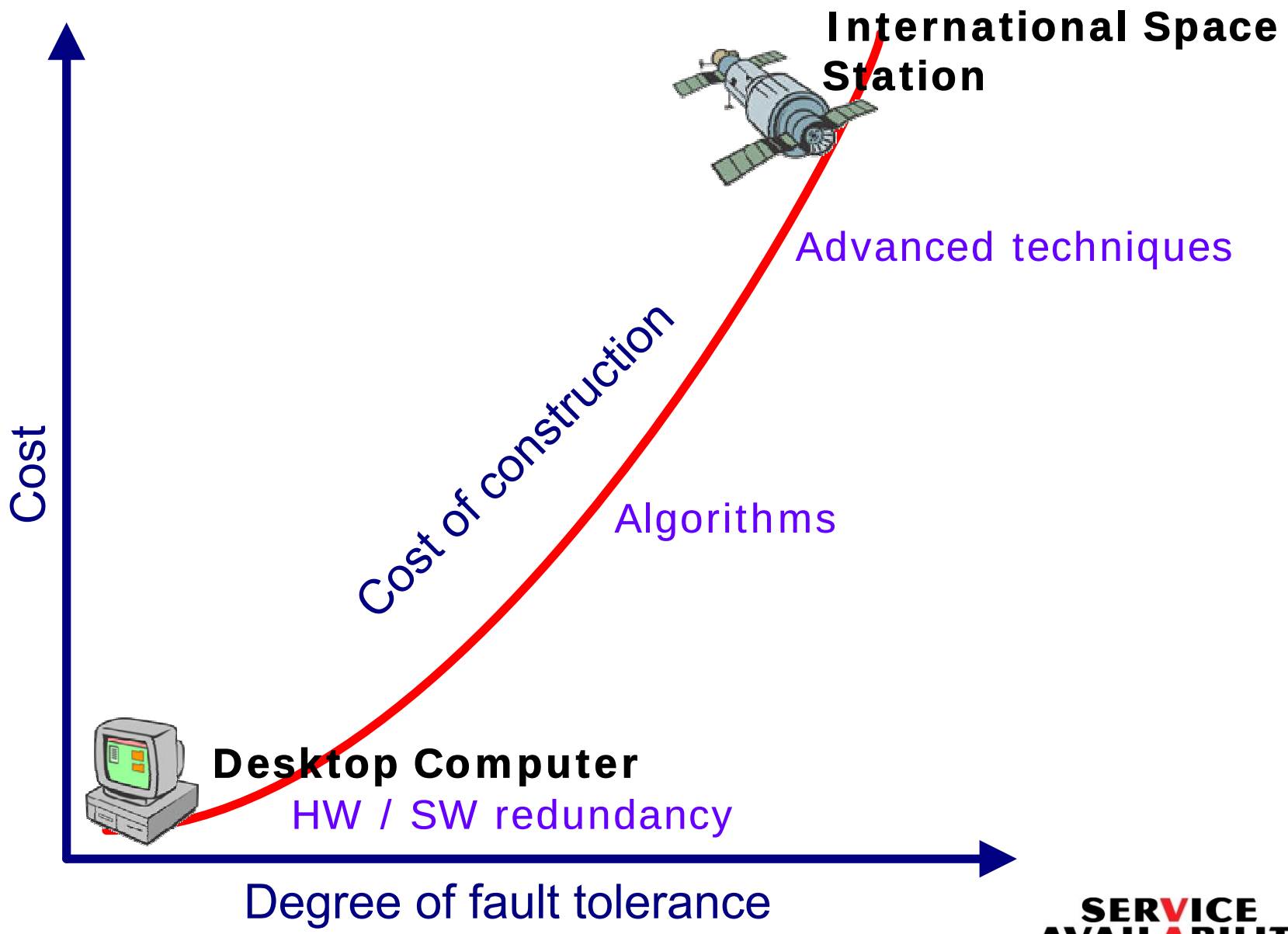
Availability	Outage duration per year
0.99999	~5 mins
0.9999	~52 mins
0.999	~8 hrs
0.99	~3 days

Dependability and performance vs. costs



SERVICE AVAILABILITY™ FORUM

FT vs. Cost



SERVICE AVAILABILITY™ FORUM

THE AIS SPECIFICATION

Programming model

SERVICE AVAILABILITY™ FORUM

Programming model

- ***Library life cycle***
  - Initialize, finalize, dispatch events
- ***Service specific APIs***
  - Callbacks
  - Request functions

Programming model - AMF

- ***Library life cycle***
  - *saAmfInitialize(), saAmfFinalize(), saAmfDispatch()*
- ***Service specific APIs***
  - Callbacks
    - *saAmfCSISetCallbackT()*
  - Request functions
    - *saAmfHAStateGet()*

Programming model – how to use it?

- ***Initialize service handler***
  - *saAmfInitialize()*
    - Set callbacks
- ***Get selection object***
  - *saAmfSelectionObjectGet()*
- **Start main loop**
  - do {
    - select(amfSelectionObject,…)
    - saAmfDispatch()
  - } while (1)

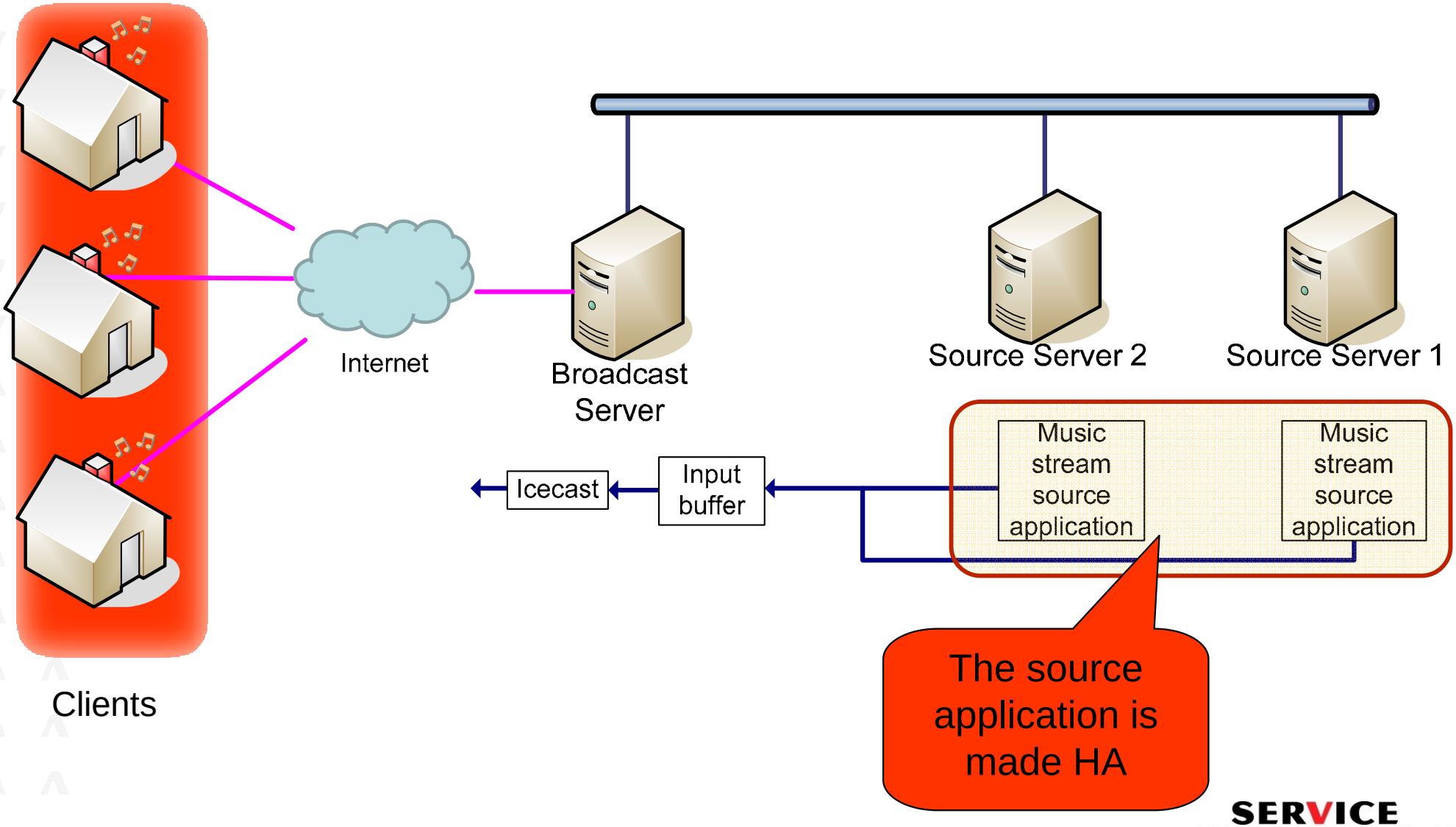
SERVICE AVAILABILITY™ FORUM

Open Specifications for Service Availability

Demo

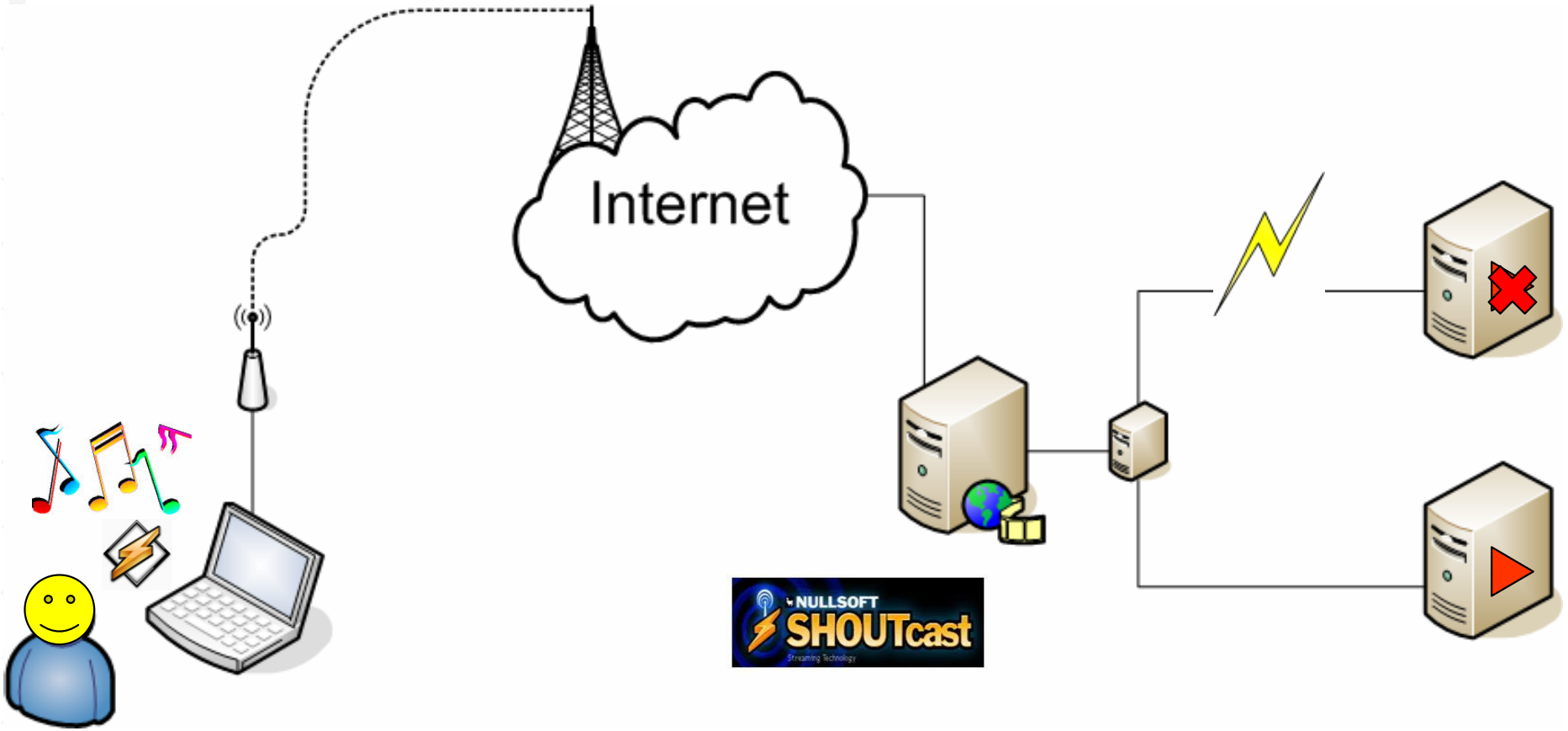
Highly Available / Fault Tolerant
Music Broadcast Application

Architecture



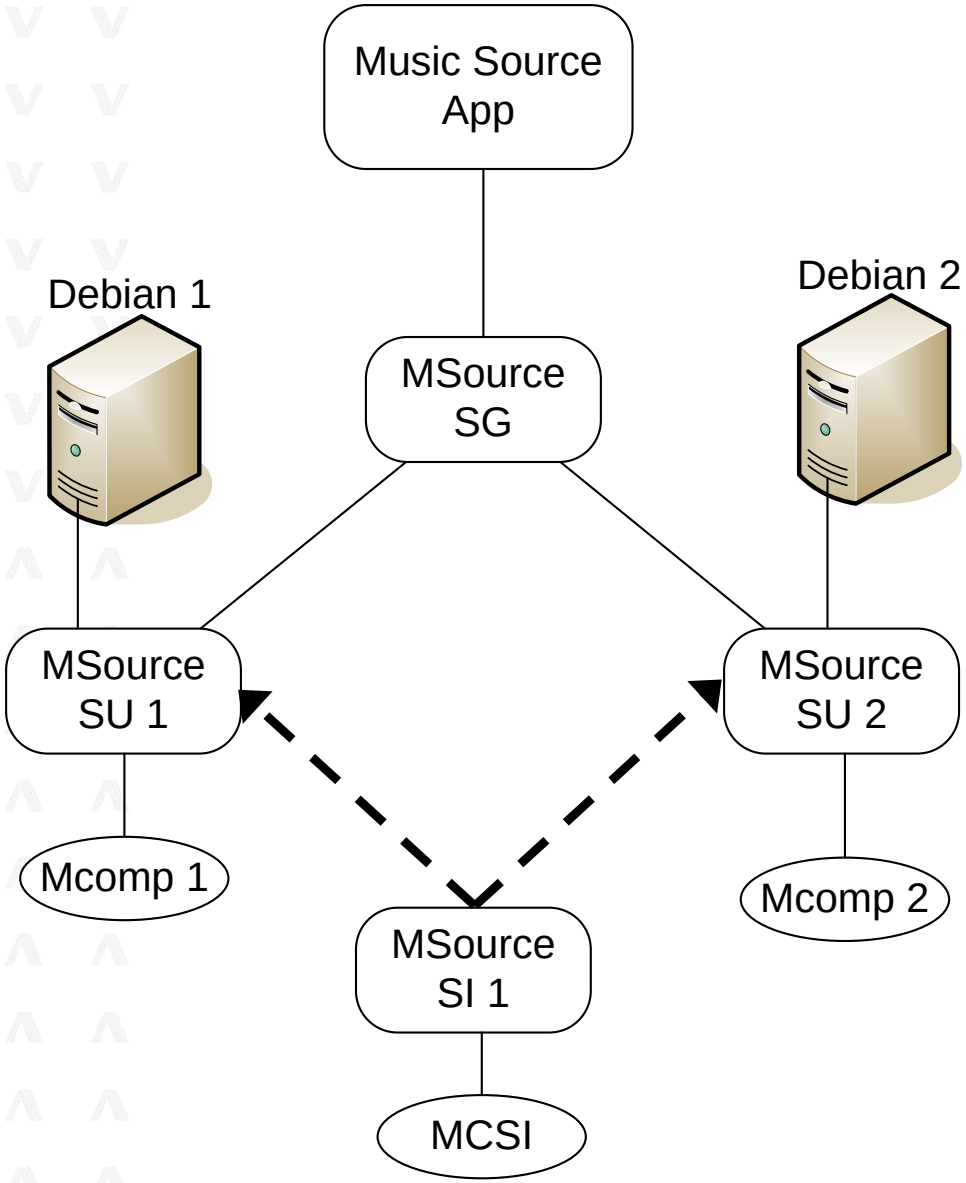
SERVICE AVAILABILITY™ FORUM

Online radio streaming system



SERVICE AVAILABILITY™ FORUM

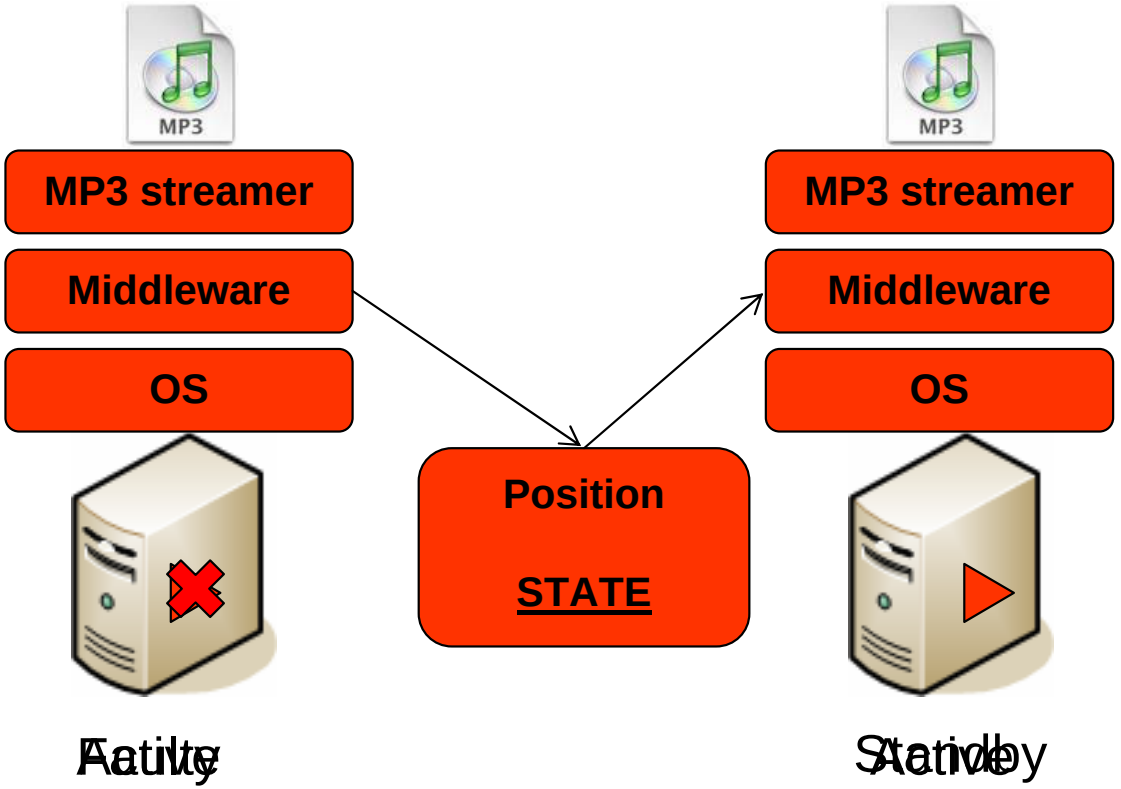
AMF configuration



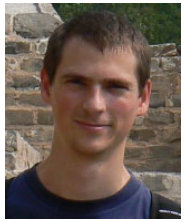
- ***Tolerated failures***
  - Component
  - Node
  - One communication channel

SERVICE AVAILABILITY™ FORUM

Where should I continue?



Acknowledgements

- ***András Kövi*** 
- ***Zoltán Micskei,*** 
- ***István Majzik*** 
- ***András Pataricza*** 

SERVICE AVAILABILITY™ FORUM

References

- ***Service Availability Forum http://saforum.org***
  - Application Interface Specification
    - http://www.saforum.org/specification/AIS_Information/
  - Education Material
    - http://www.saforum.org/education/

- ***Fault Tolerance Research Group – (BME MIT)***
  - http://www.inf.mit.bme.hu/FTSRG/

SERVICE AVAILABILITY™ FORUM