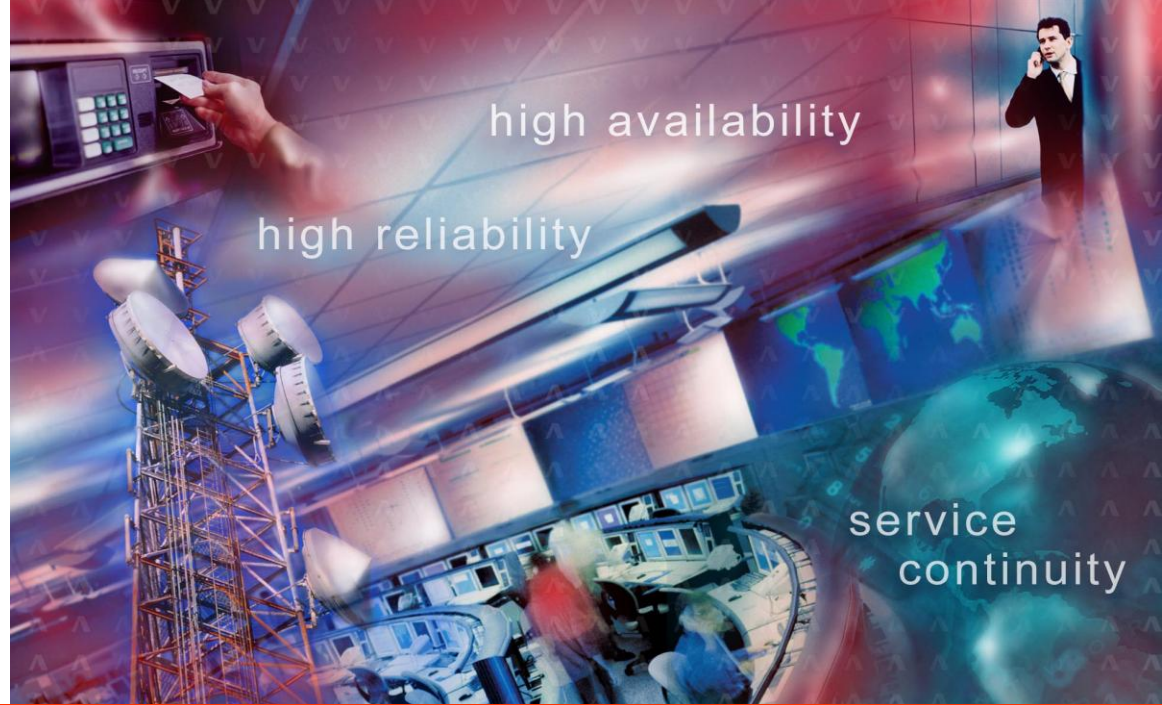**SERVICE AVAILABILITY FORUM**™

Open Specifications for Service Availability

high availability

high reliability

service continuity

Introduction to the Service Availability Forum

Contents

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**

SERVICE AVAILABILITY™
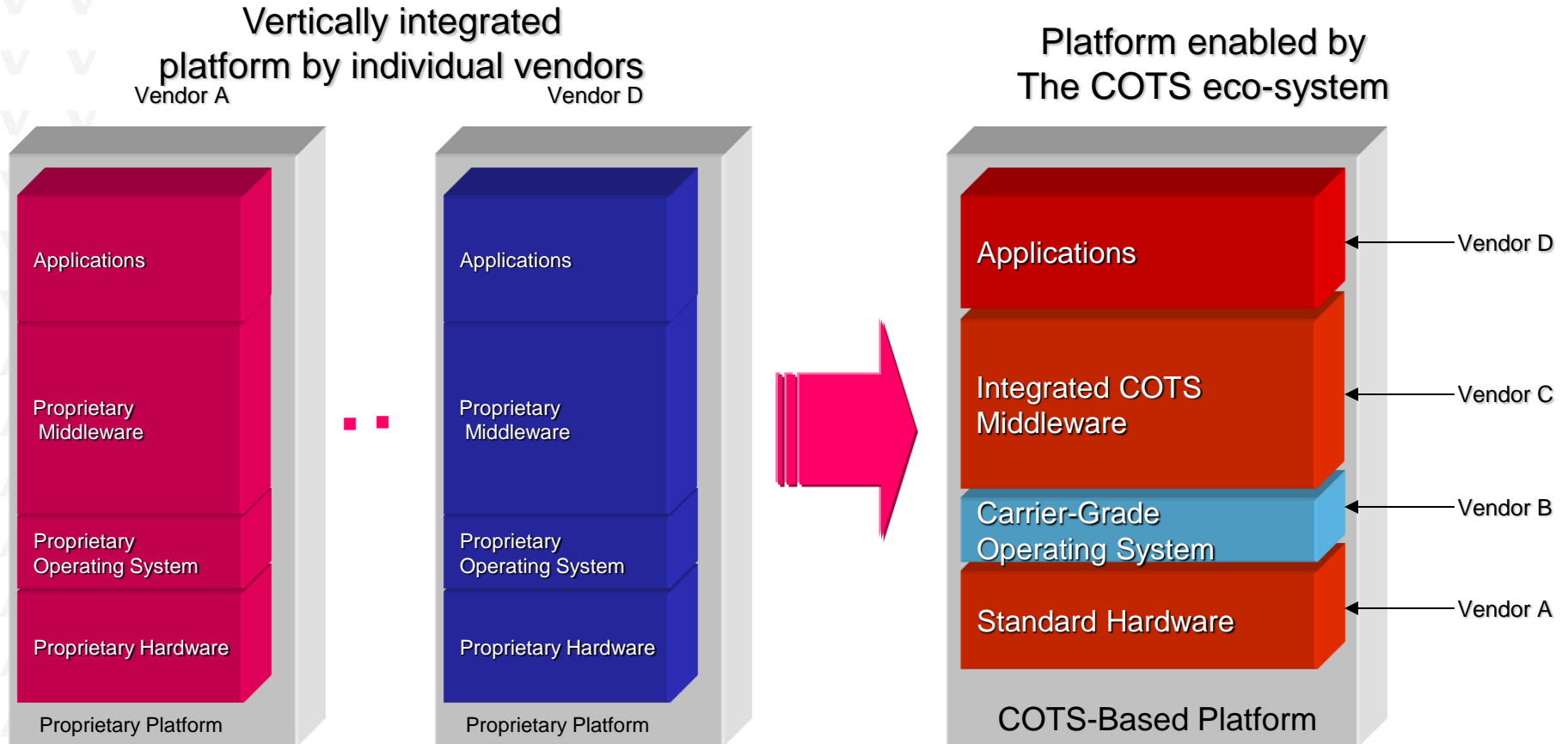**FORUM**

Construction of a service

➢ *Define the functionality*

➢ *Plan the architecture*

  ➤ Components, roles

➢ *Integrate fault management*

  ➤ Service state monitoring

  ➤ Management

➢ *Plan recovery*

➢ *Communication between components*

SERVICE AVAILABILITY FORUM™

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)*

SERVICE AVAILABILITY ™
FORUM

The Transition

COTS adoption is accelerating transition from vertical to horizontal industry model

Vertically integrated platform by individual vendors

Platform enabled by The COTS eco-system

Vendor A

Vendor D

Applications	Applications
Proprietary Middleware	Proprietary Middleware
Proprietary Operating System	Proprietary Operating System
Proprietary Hardware	Proprietary Hardware
Proprietary Platform	Proprietary Platform

Applications ← Vendor D

Integrated COTS Middleware ← Vendor C

Carrier-Grade Operating System ← Vendor B

Standard Hardware ← Vendor A

COTS-Based Platform
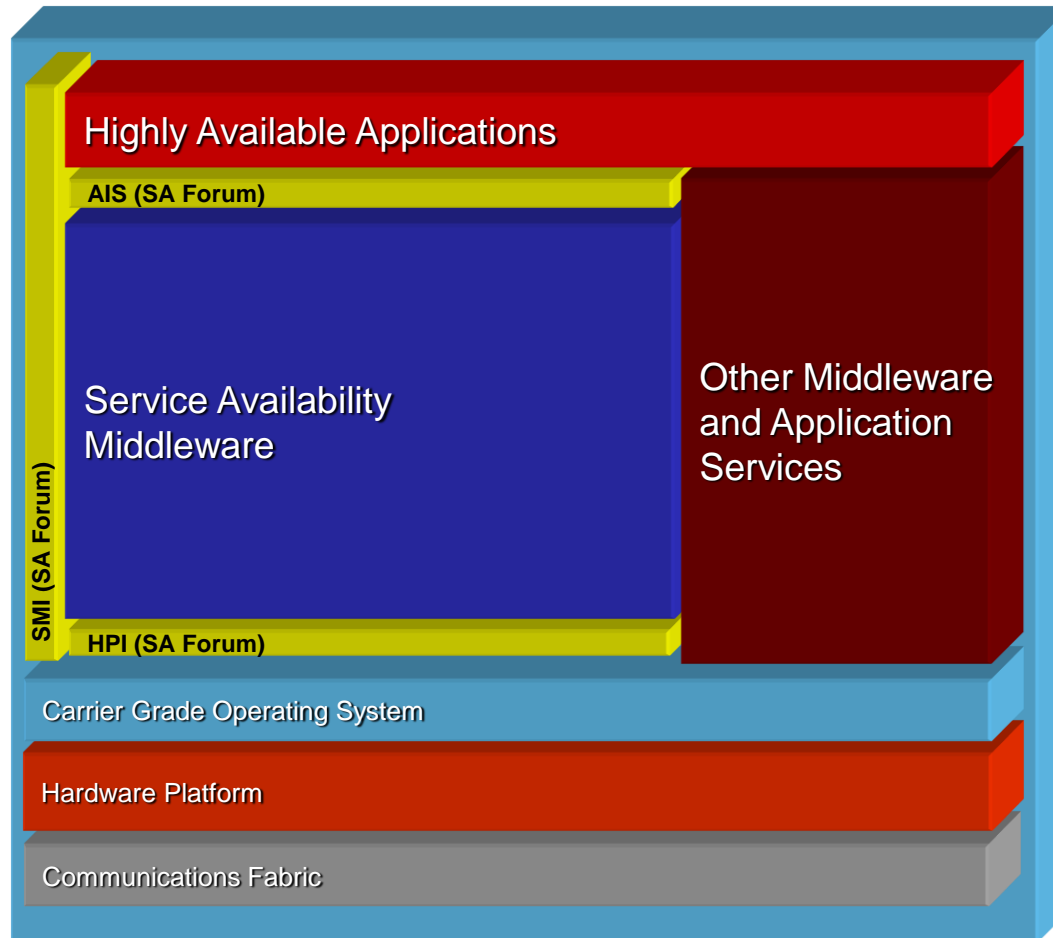
SERVICE
AVAILABILITY™
FORUM

SA Forum Standard Interface Specifications

Application Interface Specification (AIS)

Hardware Platform Interface (HPI)

Systems Management Interfaces (SMI)

SERVICE AVAILABILITY™ FORUM

SA Forum Members

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

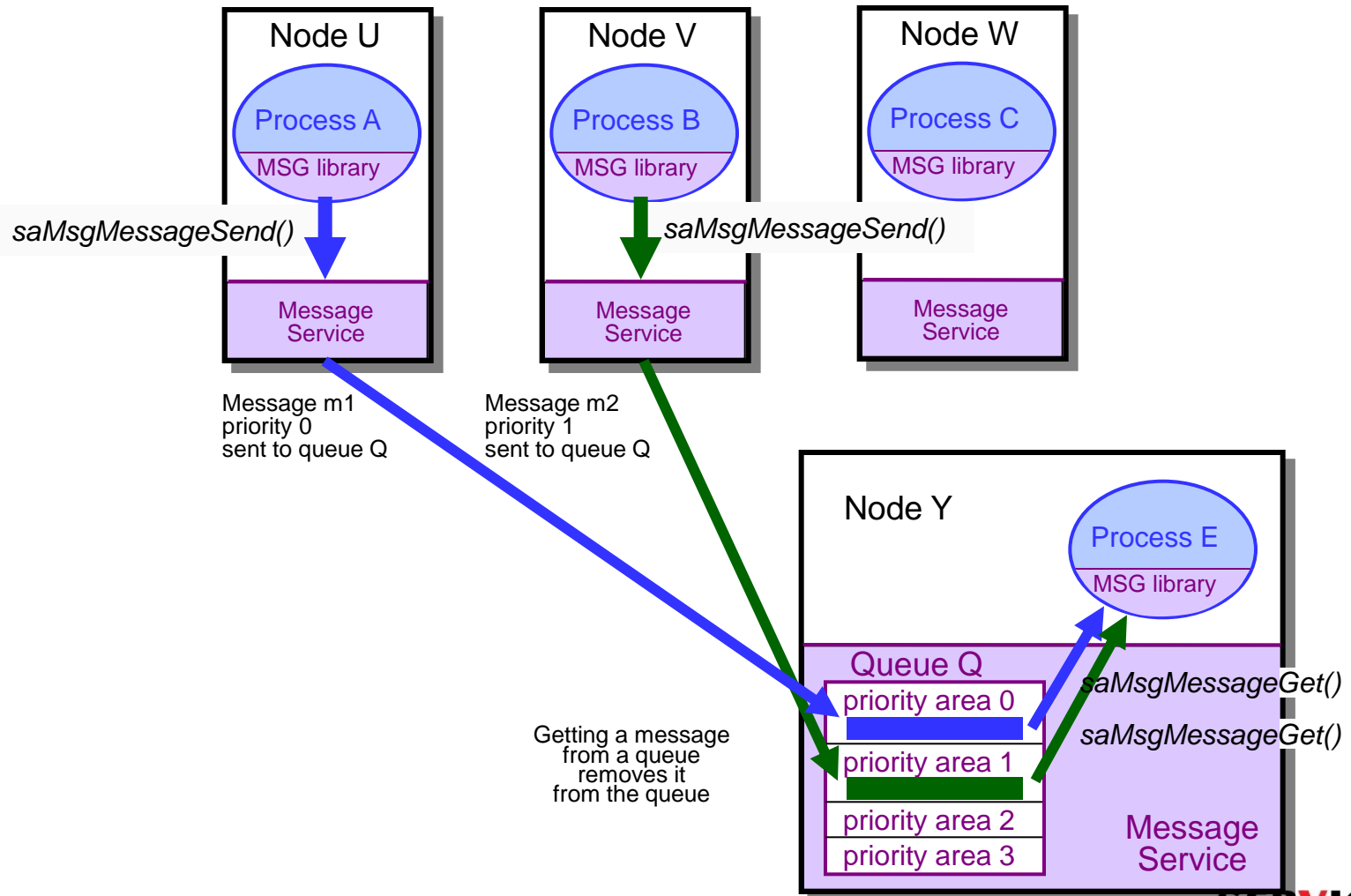 ➢ HW user: the HDD (Samsung, Western Digital, Seagate, Hitachi…)

SERVICE AVAILABILITY
FORUM ™

Application Interface Specification

Copyright© 2006 Service Availability™ Forum, Inc

SERVICE AVAILABILITY™ FORUM
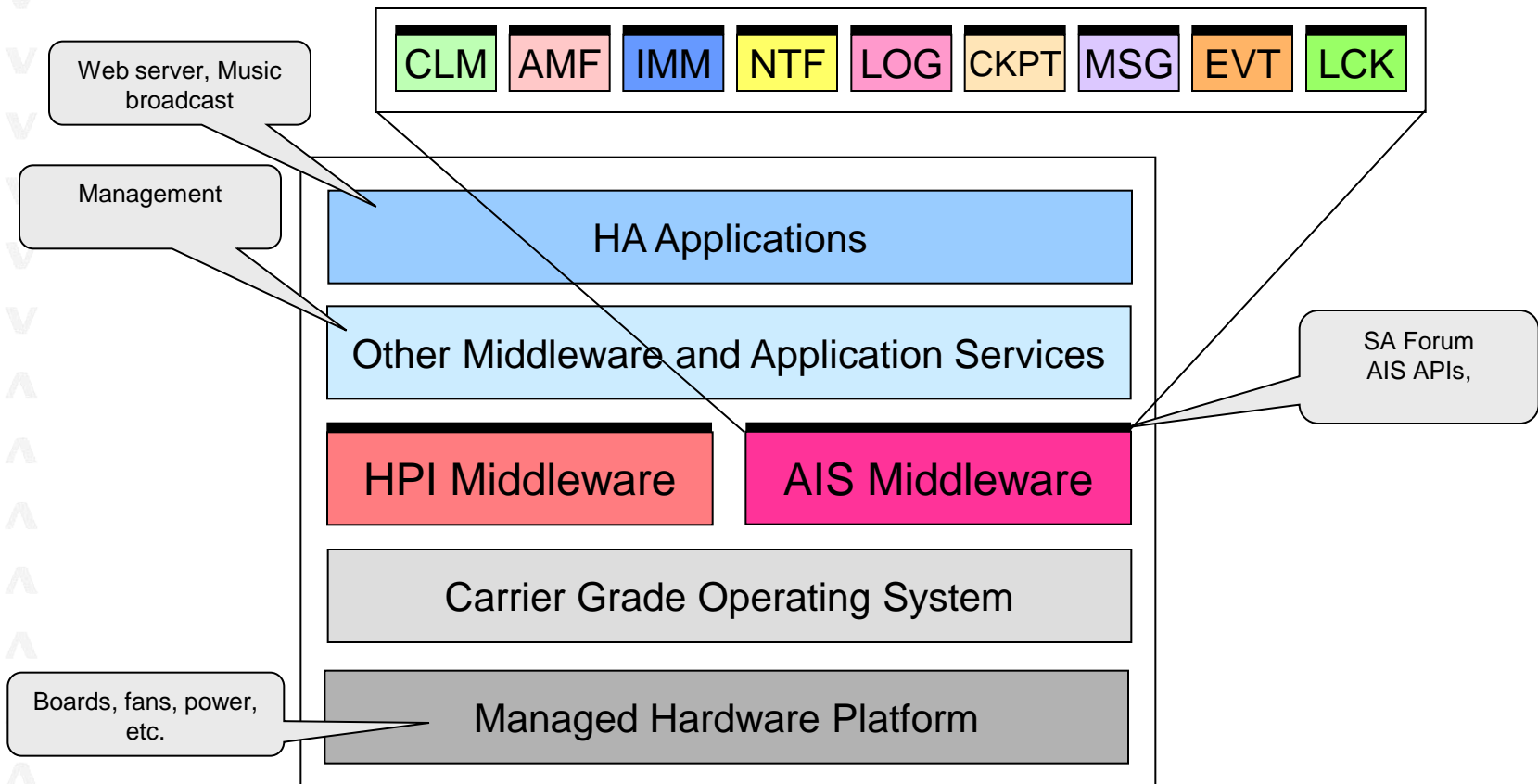
Application Interface Specification

➤ *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

SERVICE
AVAILABILITY™
FORUM

Example service - Message Queues

Node U

Process A

MSG library

saMsgMessageSend()

Message Service

Message m1
priority 0
sent to queue Q

Node V

Process B

MSG library

saMsgMessageSend()

Message Service

Message m2
priority 1
sent to queue Q

Node W

Process C

MSG library

Message Service

Node Y

Process E

MSG library

Queue Q

priority area 0

priority area 1

priority area 2

priority area 3

saMsgMessageGet()

saMsgMessageGet()

Getting a message
from a queue
removes it
from the queue

Message Service

SERVICE
AVAILABILITY™
FORUM

13

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
- ➢ <u>**Fault tolerance**</u>
  - ➤ <u>Provide service in spite of faults</u>
- ➢ **Fault removal**
  - ➤ Diagnostics, monitoring, repair at development and runtime
- ➢ **Fault forecasting / prediction**
  - ➤ Estimating failures and their effects
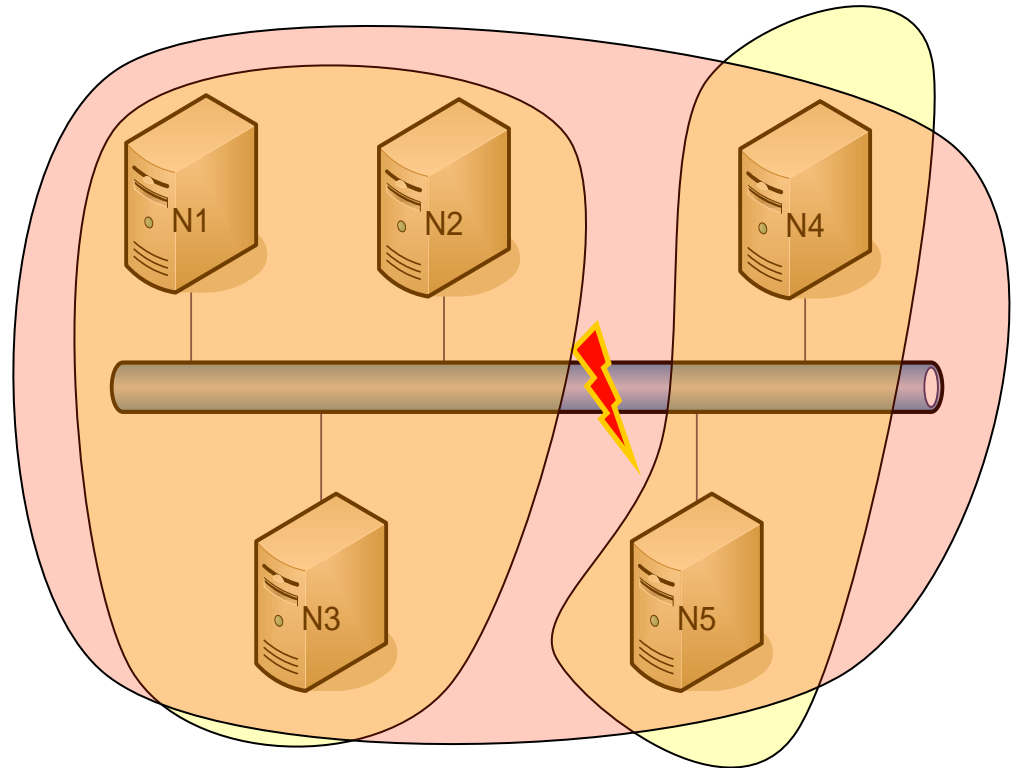 - ➤ E.g. software aging

SERVICE AVAILABILITY™ FORUM

Server Clustering

➤ *Notions*

- ❯ node
- ❯ link
- ❯ cluster
- ❯ partition

➤ *Functionalities of cluster middleware*

- ❯ Error detection
- ❯ Error handling
- ❯ Notification
- ❯ Reconfiguration

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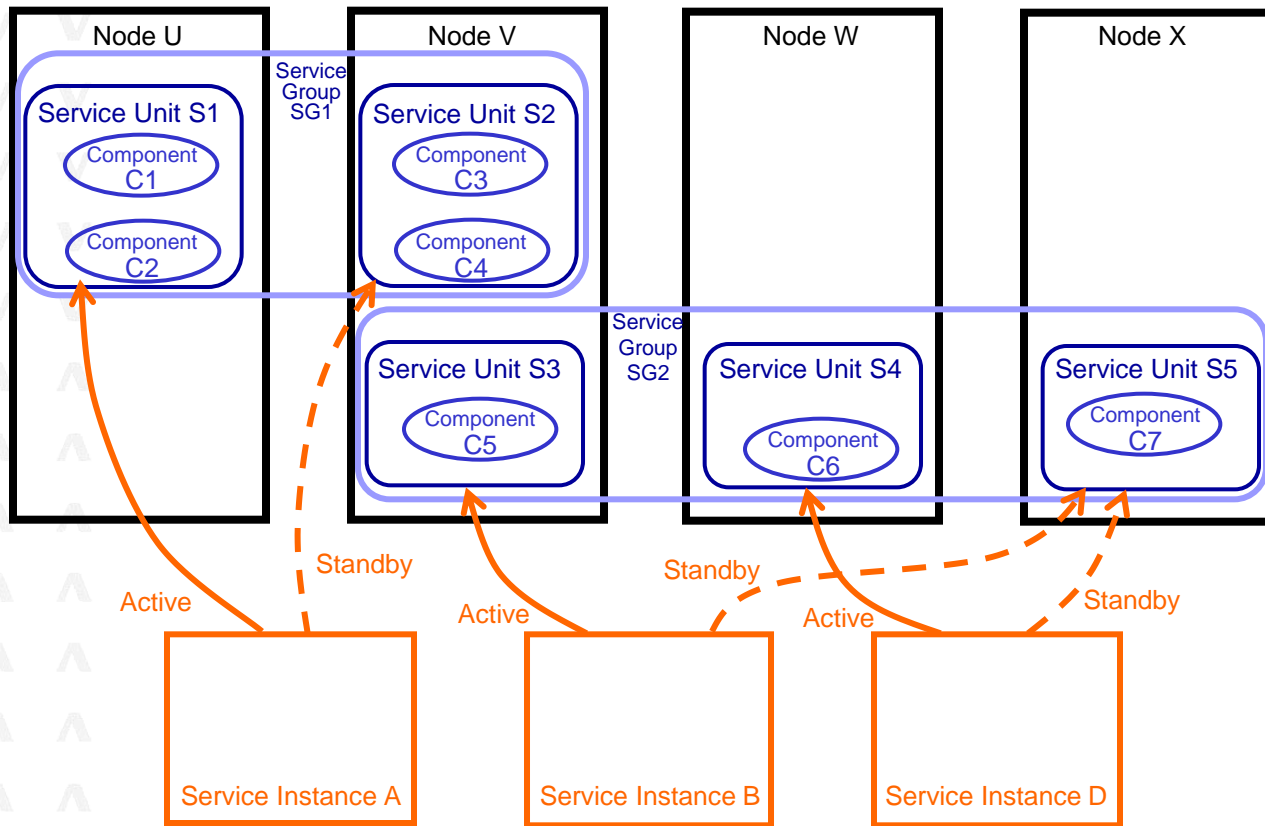
- ***HW***
  - Redundant power supply
- ***SW***
  - Process replicas, failover, switchover…
- ***Clusters – hybrid solutions***
  - Process replicas on different nodes
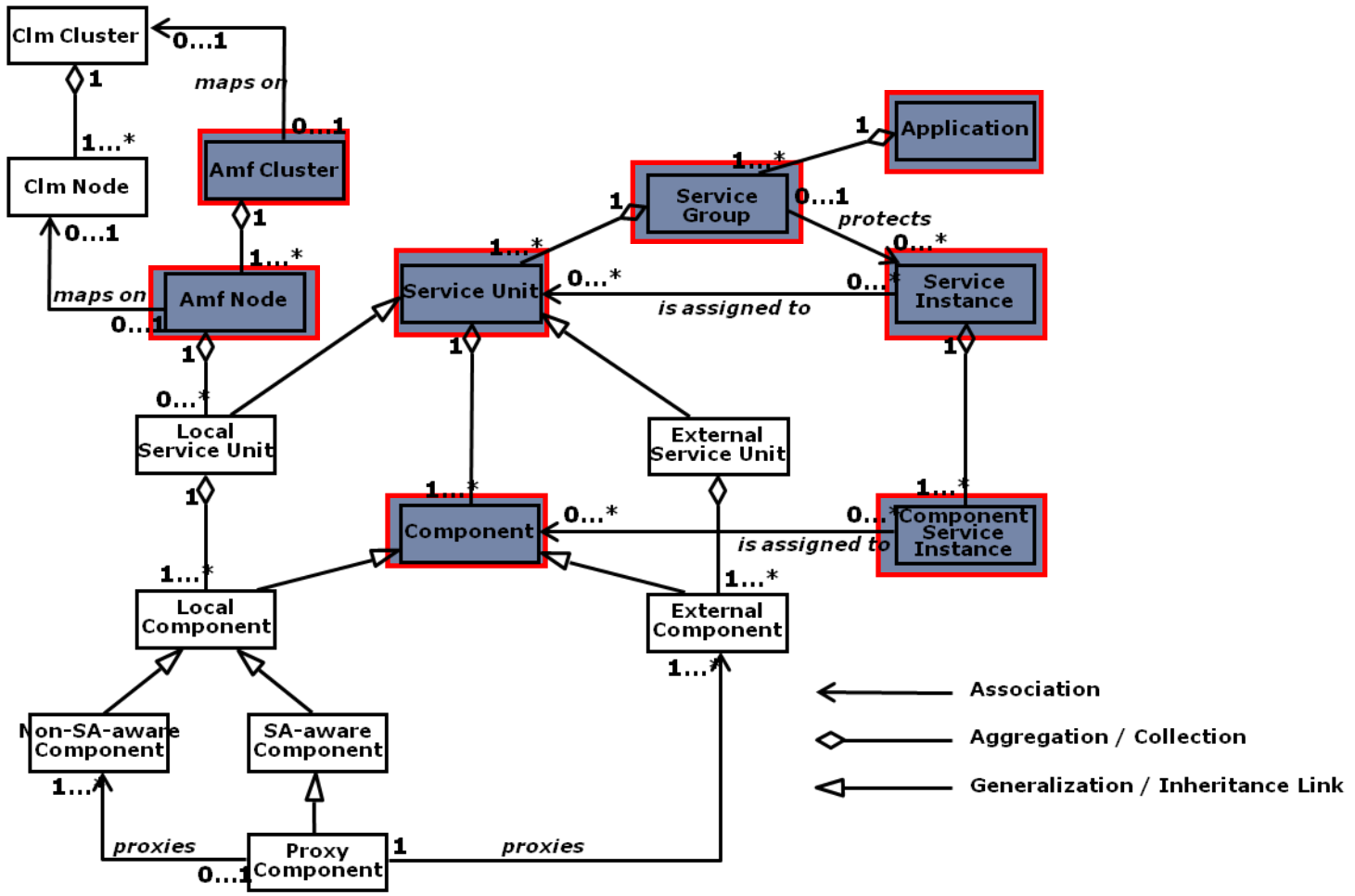
AMF System Model Example

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

Node U

Service Group SG1

Service Unit S1
- Component C1
- Component C2

Node V

Service Unit S2
- Component C3
- Component C4

Service Group SG2

Service Unit S3
- Component C5

Node W

Service Unit S4
- Component C6

Node X

Service Unit S5
- Component C7

Active
Standby
Active
Standby
Active
Standby

Service Instance A

Service Instance B

Service Instance D

Similarly, for service group SG2

SERVICE AVAILABILITY FORUM

The SA Forum Information Model

Fault Management

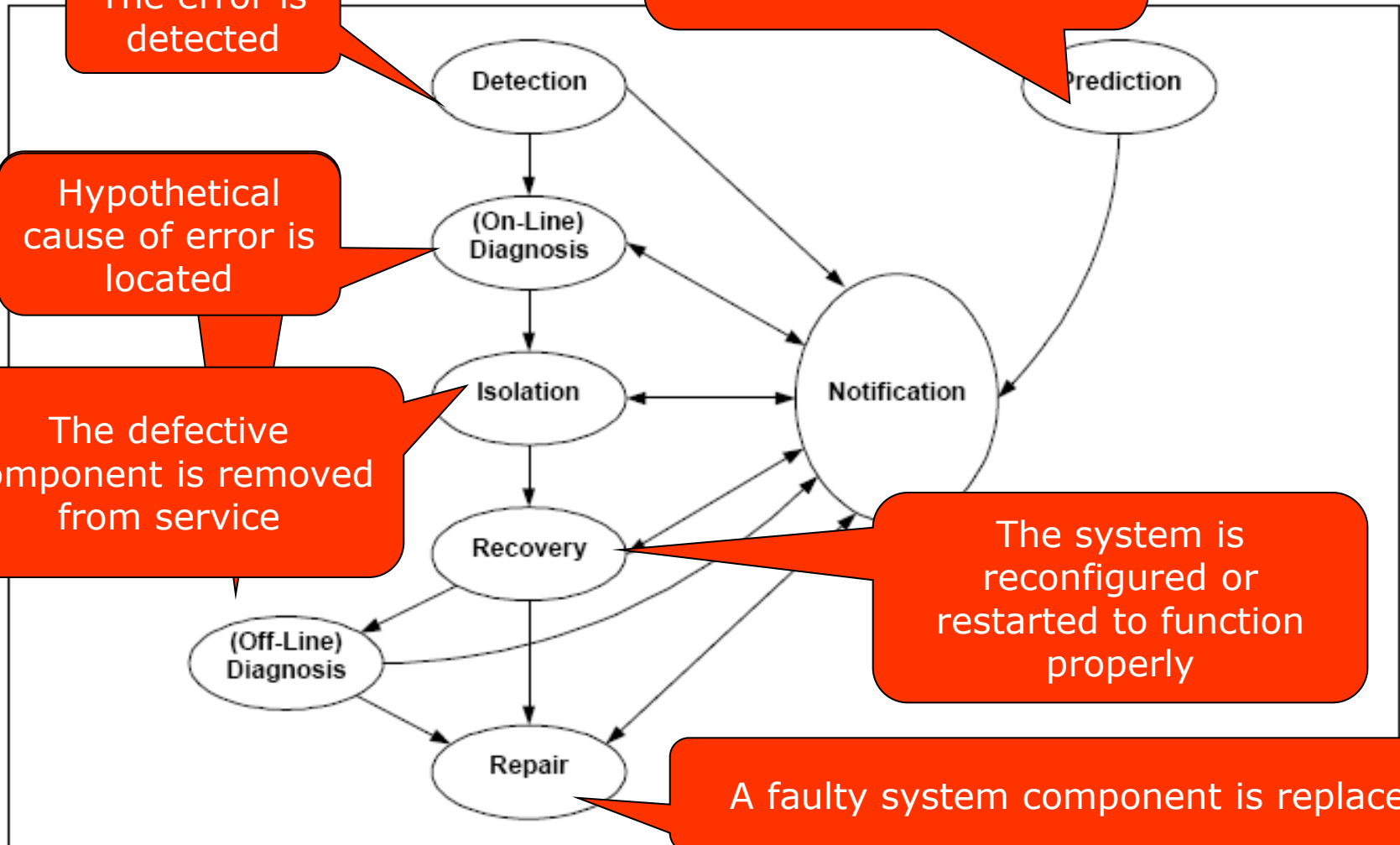An alternate form of fault detection, including built-in diagnosis

The error is detected

Hypothetical cause of error is located

The defective component is removed from service

The system is reconfigured or restarted to function properly

A faulty system component is replaced

Detection

Prediction

(On-Line) Diagnosis

Isolation

Notification

Recovery

(Off-Line) Diagnosis

Repair

AVAILABILITY
FORUM

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

> ## *Passive monitoring*

    Needs the change of the component

    ➤ 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

SERVICE AVAILABILITY
FORUM

Checkpoint Service

➢ *Example of checkpointing, and restoration from a checkpoint after a fault, for a collocated checkpoint*
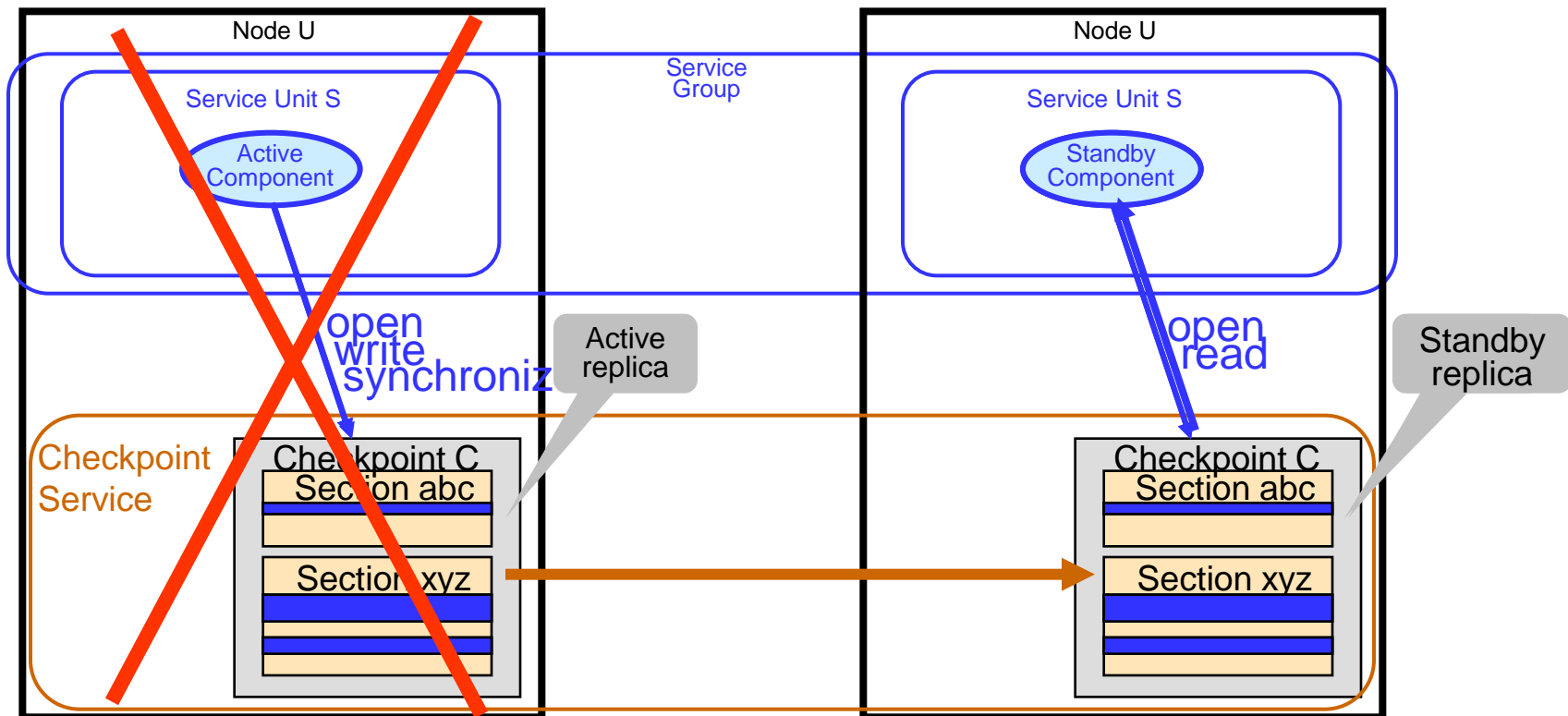
Methods of recovery

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 AVAILABILITY
FORUM

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*
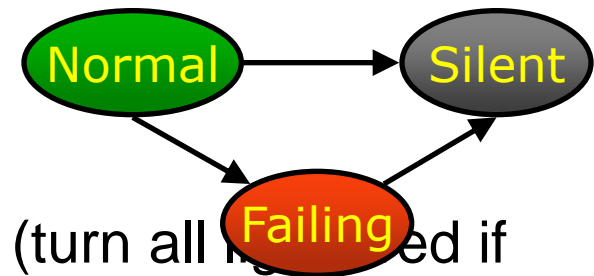 - Service is carried on in a degraded state, probably with degraded functionality

SERVICE AVAILABILITY
FORUM

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

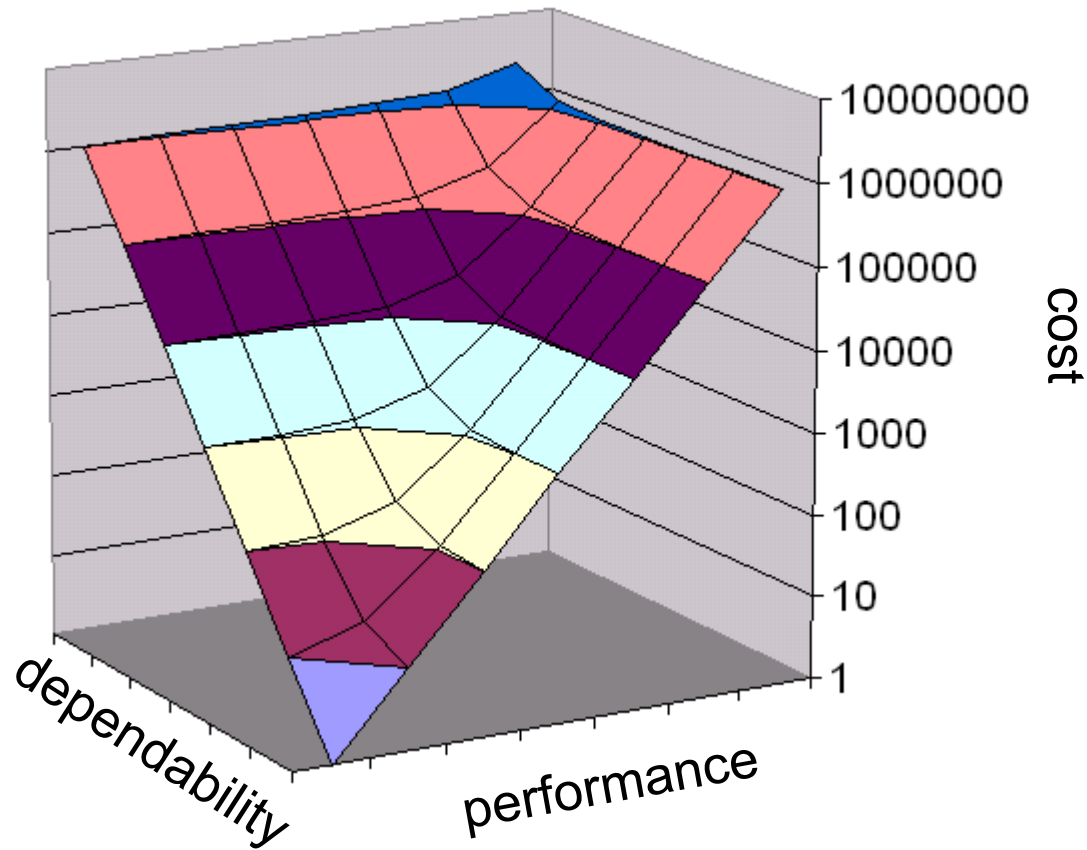   ➤ E.g. train traffic control systems (turn all lights red if failure happens)

Normal → Silent

Failing

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

SERVICE
AVAILABILITY™
FORUM

Requirements in today's systems

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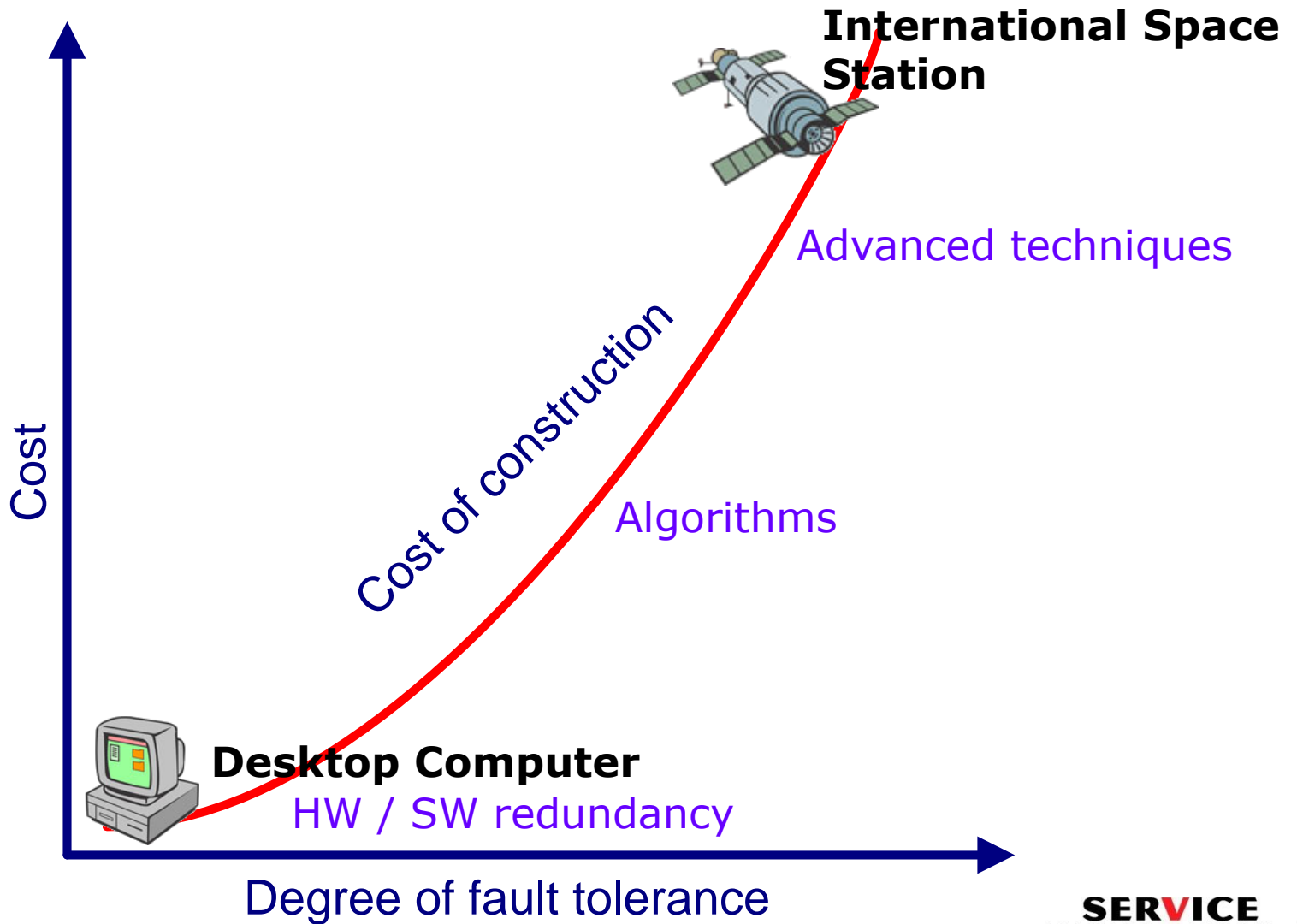- ➢ *Dependable services*
  - ➤ Highly Available
  - ➤ Reliable

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

FT vs. Cost

Cost (y-axis)

International Space Station

Advanced techniques

Cost of construction

Algorithms

Desktop Computer

HW / SW redundancy

Degree of fault tolerance (x-axis)

SERVICE AVAILABILITY FORUM

THE AIS SPECIFICATION

Communication Services

SERVICE AVAILABILITY FORUM

Application Interface Specification

Copyright© 2006 Service Availability™ Forum, Inc

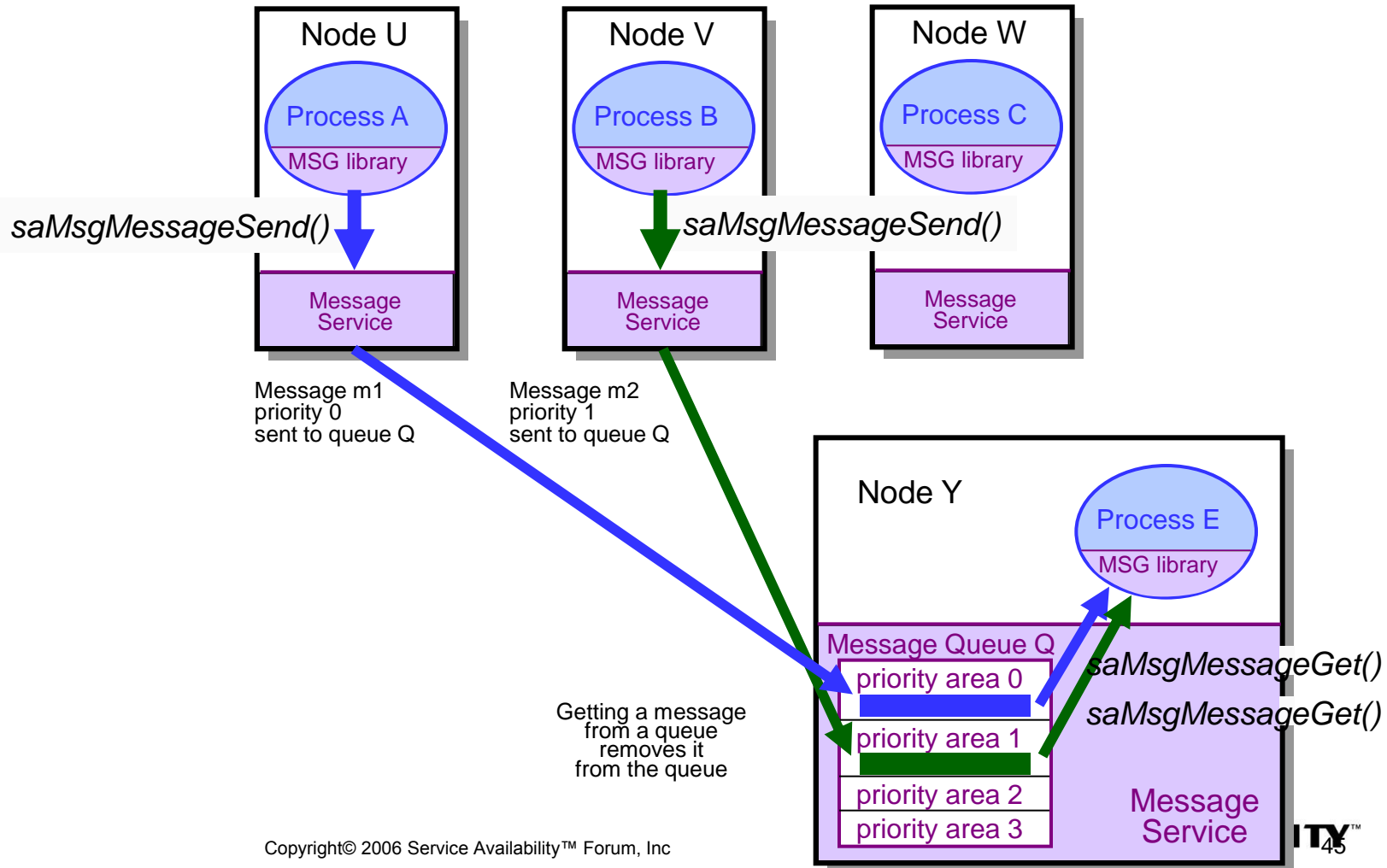**SERVICE AVAILABILITY FORUM**
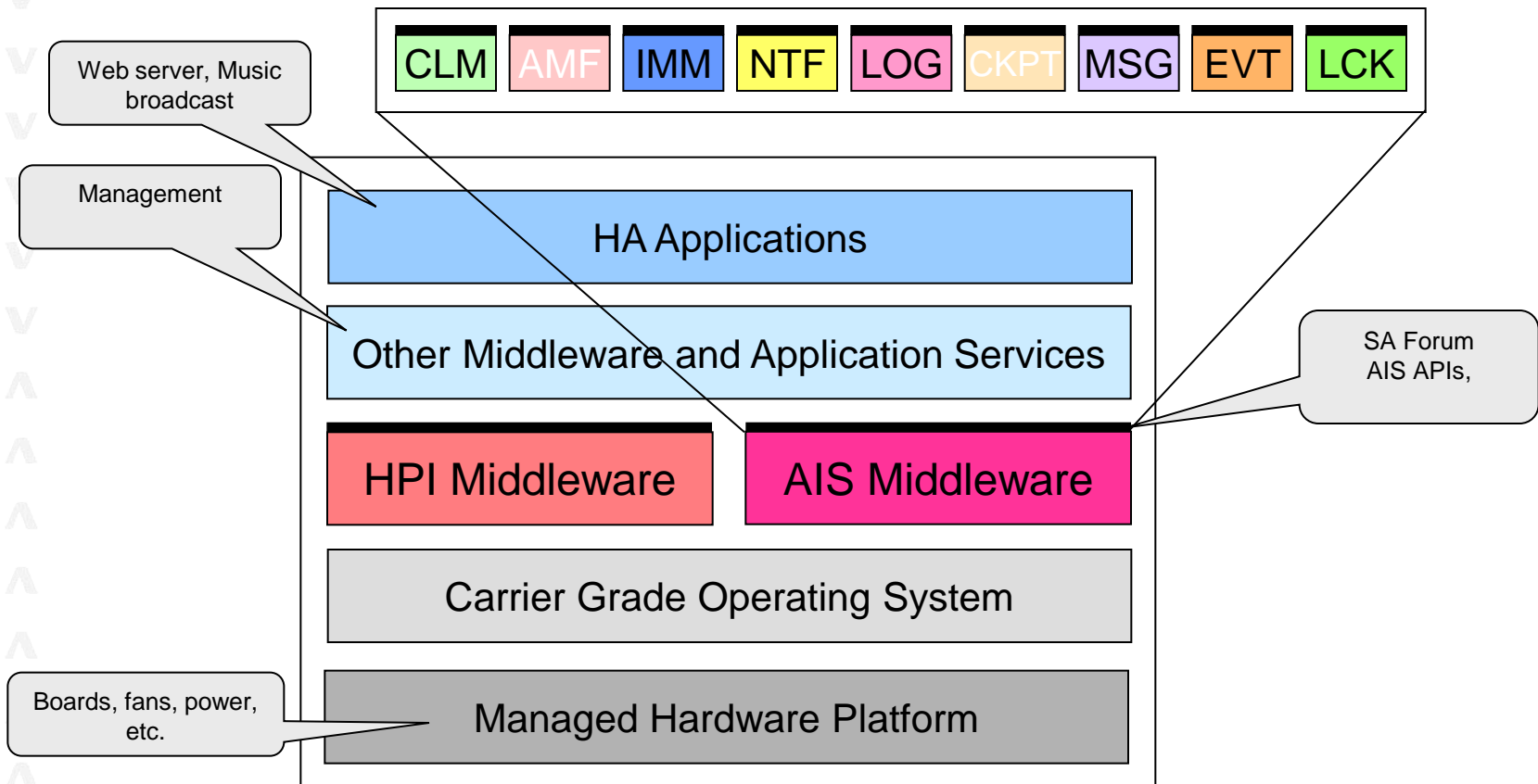
Communication

> ## *Message (MSG)*
>> ➤ Recommended for many to one scheme
>>
>> ➤ Example: collect sensor data
>
> ## *Event (EVT)*
>> ➤ Many to Many (publish – subscribe) scheme
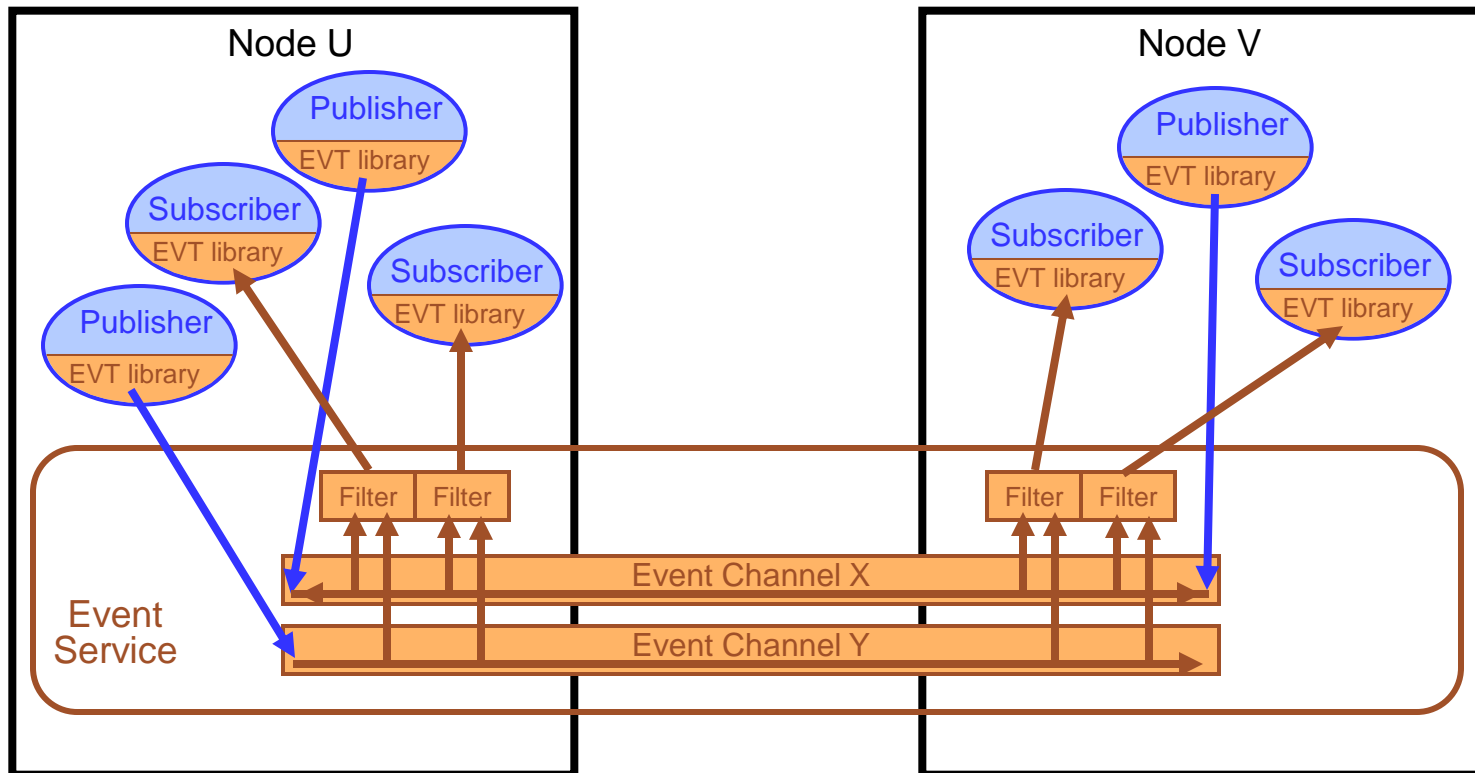>>
>> ➤ Example: Sentinels and villagers

SERVICE AVAILABILITY ™
FORUM

Message Service

➤ *Message Service, Message Service (MSG) Library, and Message Queue*

Node U
Process A
MSG library

saMsgMessageSend()

Message Service

Message m1
priority 0
sent to queue Q

Node V
Process B
MSG library

saMsgMessageSend()

Message Service

Message m2
priority 1
sent to queue Q

Node W
Process C
MSG library

Message Service

Node Y
Process E
MSG library

Getting a message
from a queue
removes it
from the queue

Message Queue Q
priority area 0
priority area 1
priority area 2
priority area 3

saMsgMessageGet()

saMsgMessageGet()

Message Service

ITУ
45

Event Service

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

Copyright© 2006 Service Availability™ Forum, Inc

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

SERVICE
AVAILABILITY™
FORUM

Programming model - AMF

➢ *Library life cycle*

   ➢ *saAmfInitialize(), saAmfFinalize(), saAmfDispatch()*

➢ *Service specific APIs*

   ➢ Callbacks

      ‣ *saAmfCSISetCallbackT()*

   ➢ Request functions

      ‣ *saAmfHAStateGet()*

SERVICE
AVAILABILITY™
FORUM

Programming model – how to use it?

➢ *Initialize service handler*

  ➤ *saAmfInitialize()*
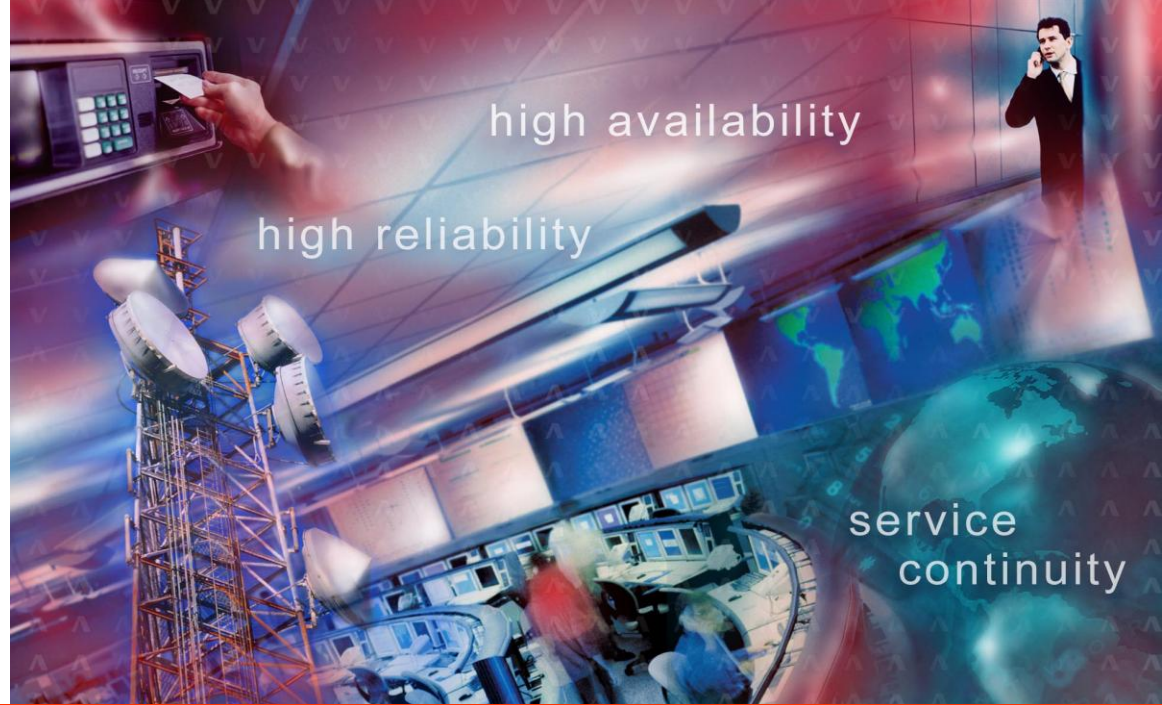
    ‣ Set callbacks

➢ *Get selection object*

  ➤ *saAmfSelectionObjectGet()*

➢ **Start main loop**

  ➤ do {

    ‣ select(amfSelectionObject,…)

    ‣ saAmfDispatch()

  ➤ } while (1)

SERVICE AVAILABILITY™ FORUM

Demo

Highly Available / Fault Tolerant
Music Broadcast Application

Architecture

Clients

Internet

Broadcast Server

Source Server 2

Source Server 1

Icecast

Input buffer

Music stream source application

Music stream source application
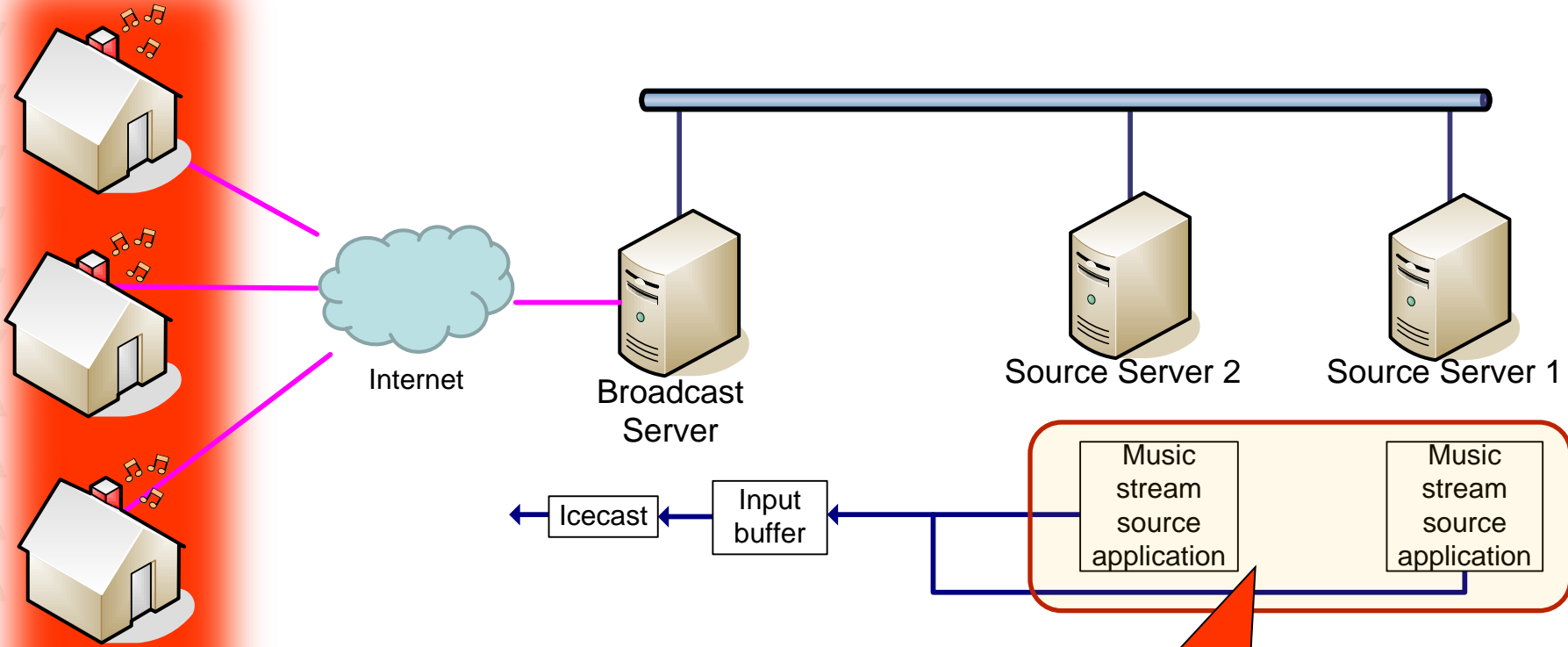
The source application is made HA

SERVICE AVAILABILITY FORUM

Online radio streaming system
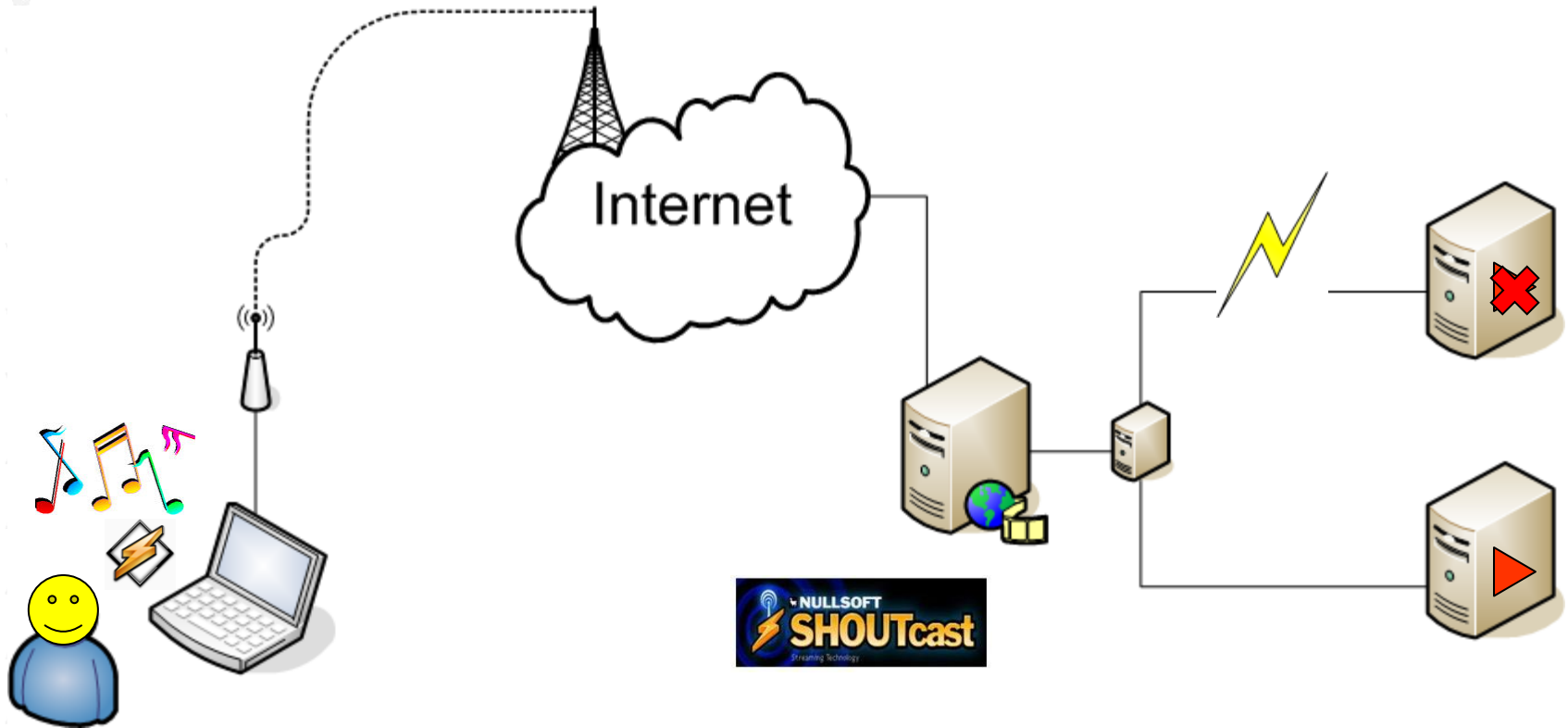
AMF configuration
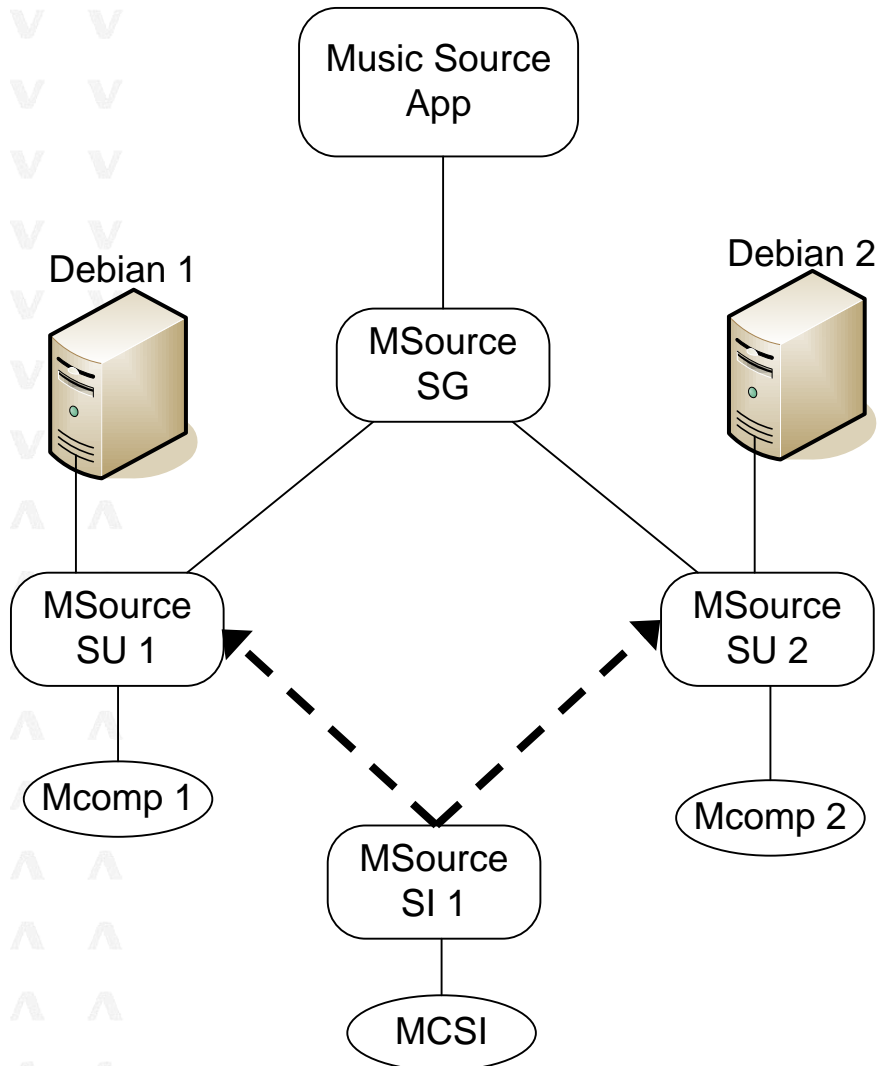
> ➤ *Tolerated failures*
>   - ➤ Component
>   - ➤ Node
>   - ➤ One communication channel

SERVICE AVAILABILITY FORUM

Where should I continue?

MP3 streamer

Middleware

OS

MP3 streamer

Middleware

OS

Position

STATE

Active

Standby

SERVICE AVAILABILITY FORUM™
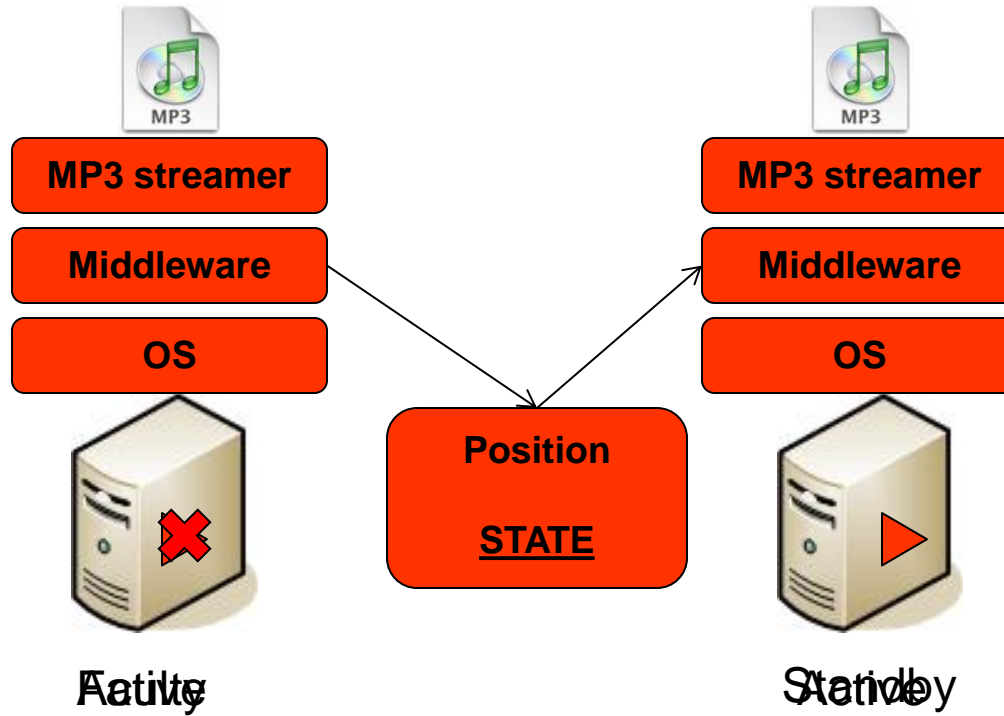
Acknowledgements

➢ *András Kövi*

➢ *Zoltán Micskei*

➢ *István Majzik*

➢ *András Pataricza*

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/