OpenSAF – A Standardized HA Solution

LinuxCON – Edinburgh, UK – 2013-10-21

**Anders Widell**
Ericsson AB

Outline

- What are OpenSAF and SA Forum?

- What is Service Availability?

- Simple Use Case: Web server

- The OpenSAF architecture

- Information model & model-based management

- New features in OpenSAF 4.3 and 4.4

- Roadmap and future

OpenSAF and SA Forum

- SA Forum is a standardisation body formed in 2001 by a group of companies in the communication and computer industry

- The SAF specifications describe APIs for 14 services in a framework for Service Availability

- OpenSAF is the only open source (LGPLv2.1) implementation of the SAF specifications

Service Availability

- The probability that a system provides its service at a (randomly chosen) point in time

- Telecom requirement is often 99.999% (five nines) availability – which allows a total downtime of around five minutes per year

- Planned downtime (hardware and software updates, system maintenance) is included in the total downtime
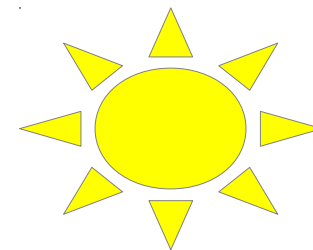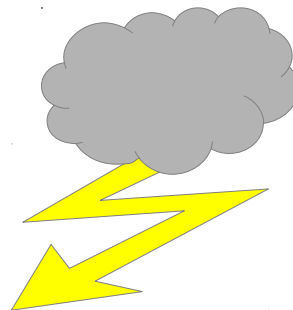
SAF Services

- AMF - Availability Management Framework

- CLM - Cluster Membership Service

- IMM - Information Model Management Service

- LOG - Log Service

- NTF - Notification Service

 Optional services include: CKPT (checkpoint), EVT (event), MSG (message queue), SMF (software management), ...
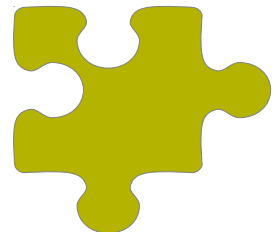
The SAF APIs

- C language APIs for applications running in the framework

- Bindings for Java and Python exist

- Backwards compatibility ensured when new API versions are introduced

- Applications don't have to use C APIs – for simple use cases, a shell script (resource agent) can be sufficient
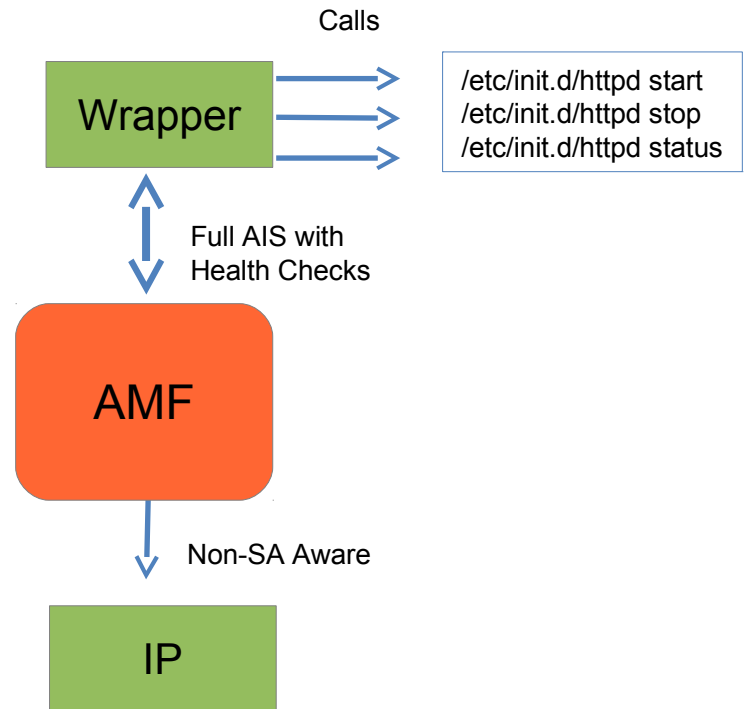
OpenSAF and Availability Spectrum
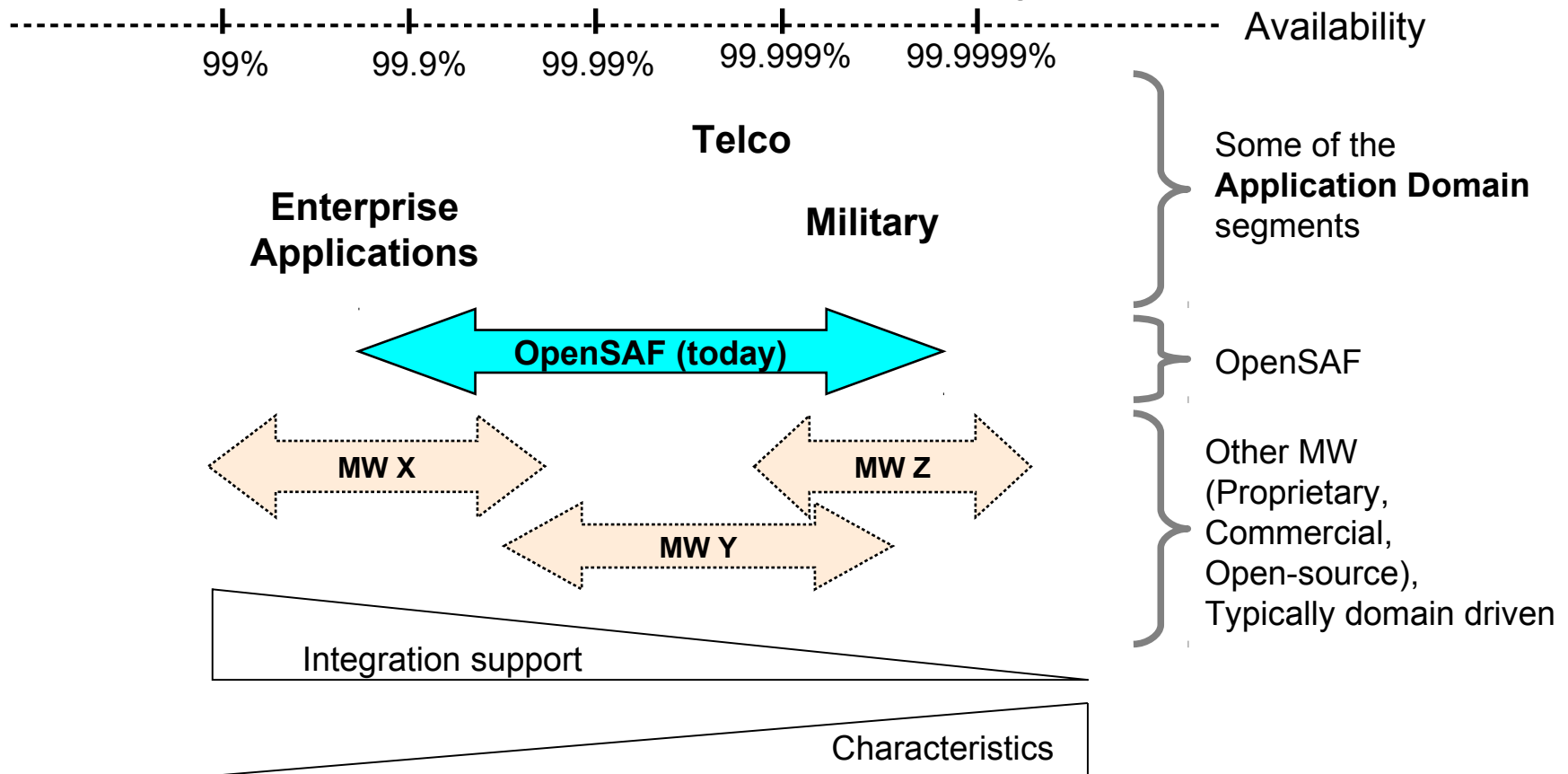
Simple Use Case: Web server

- Requirements
  - Static content => No shared disk needed
  - Migrating IP address
  - Lifecycle management
  - Health Monitoring
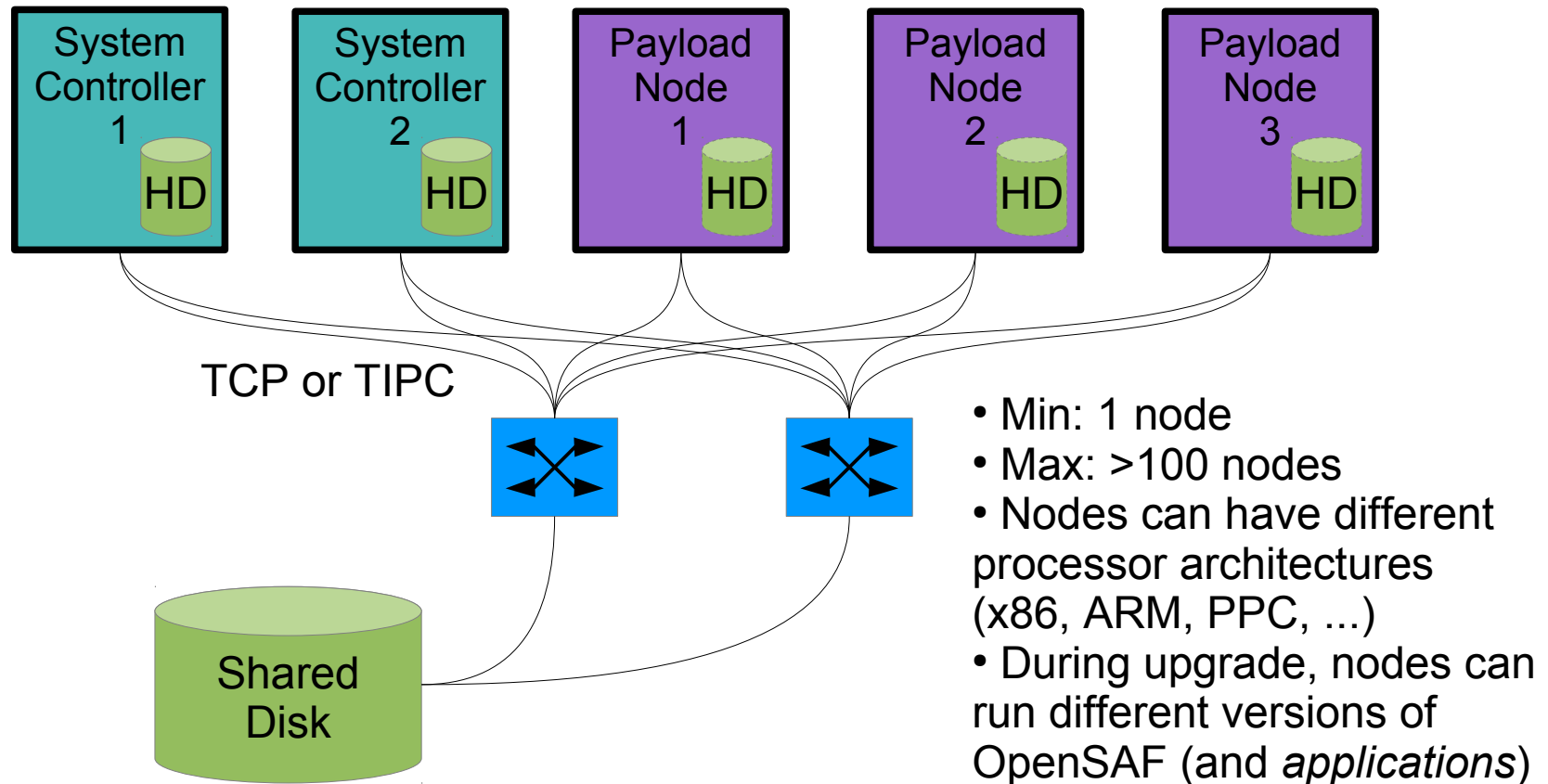 - No altering of the code
  - Cold standby
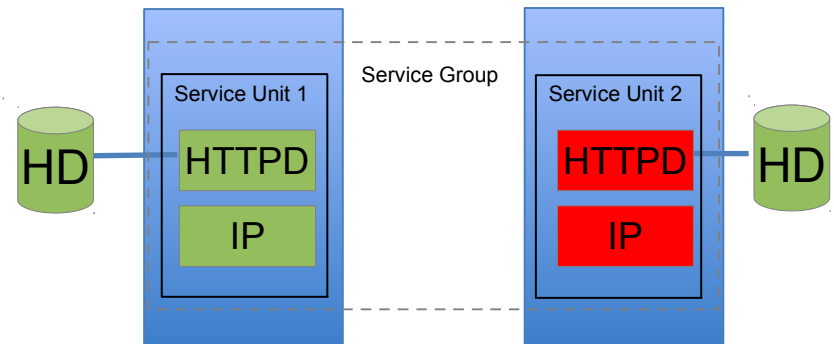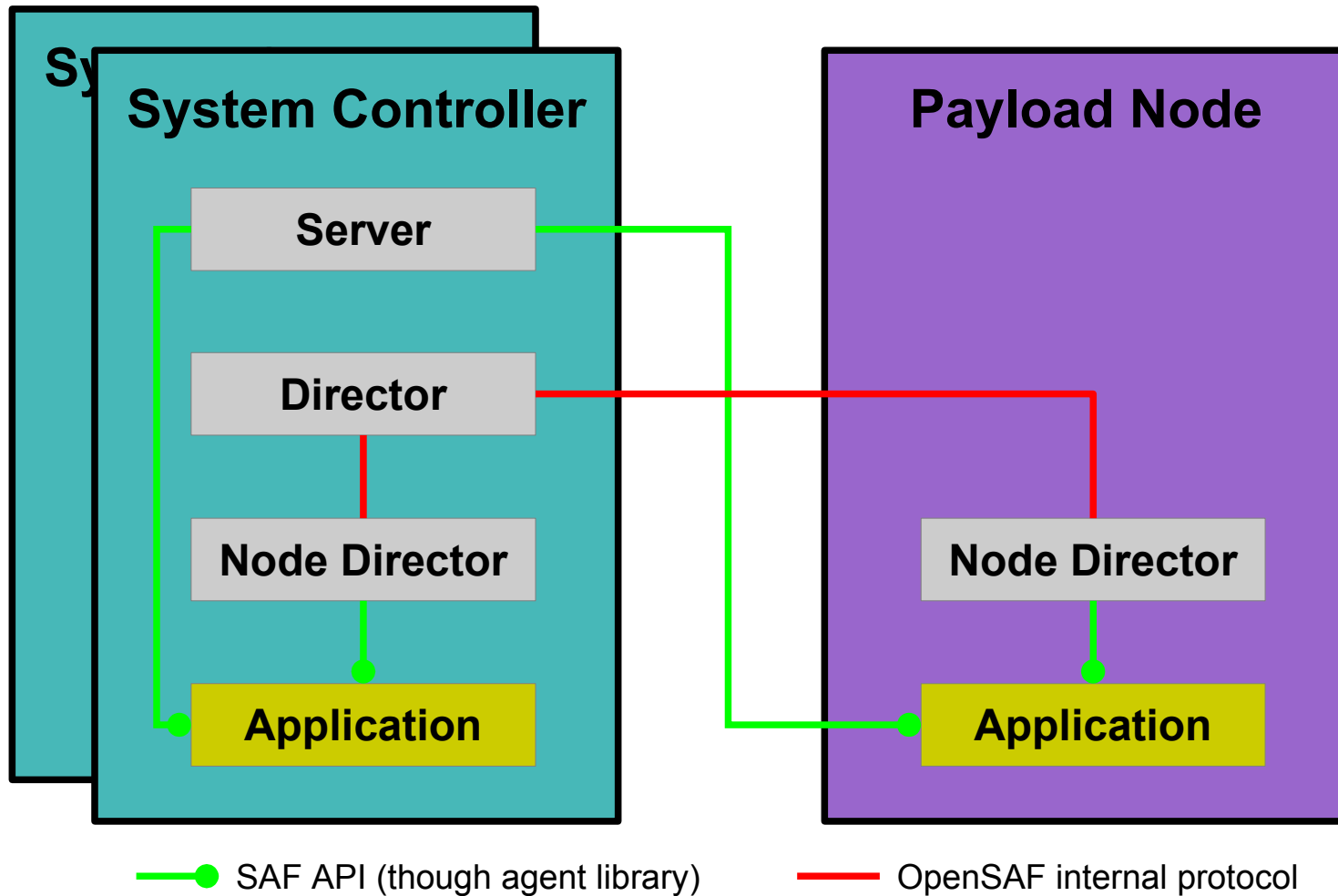
Simple Use Case: Web server

- Suggested Solution
  - Wrapper component for httpd interacts with AMF
  - Use httpd init script for lifecycle and health checks
  - PID supervision
  - IP part is non-saf aware component

Calls

Wrapper

/etc/init.d/httpd start
/etc/init.d/httpd stop
/etc/init.d/httpd status

Full AIS with
Health Checks

AMF

Non-SA Aware

IP

OpenSAF Clusters

System Controller 1 HD	System Controller 2 HD	Payload Node 1 HD	Payload Node 2 HD	Payload Node 3 HD

TCP or TIPC

Shared Disk

- Min: 1 node
- Max: >100 nodes
- Nodes can have different processor architectures (x86, ARM, PPC, ...)
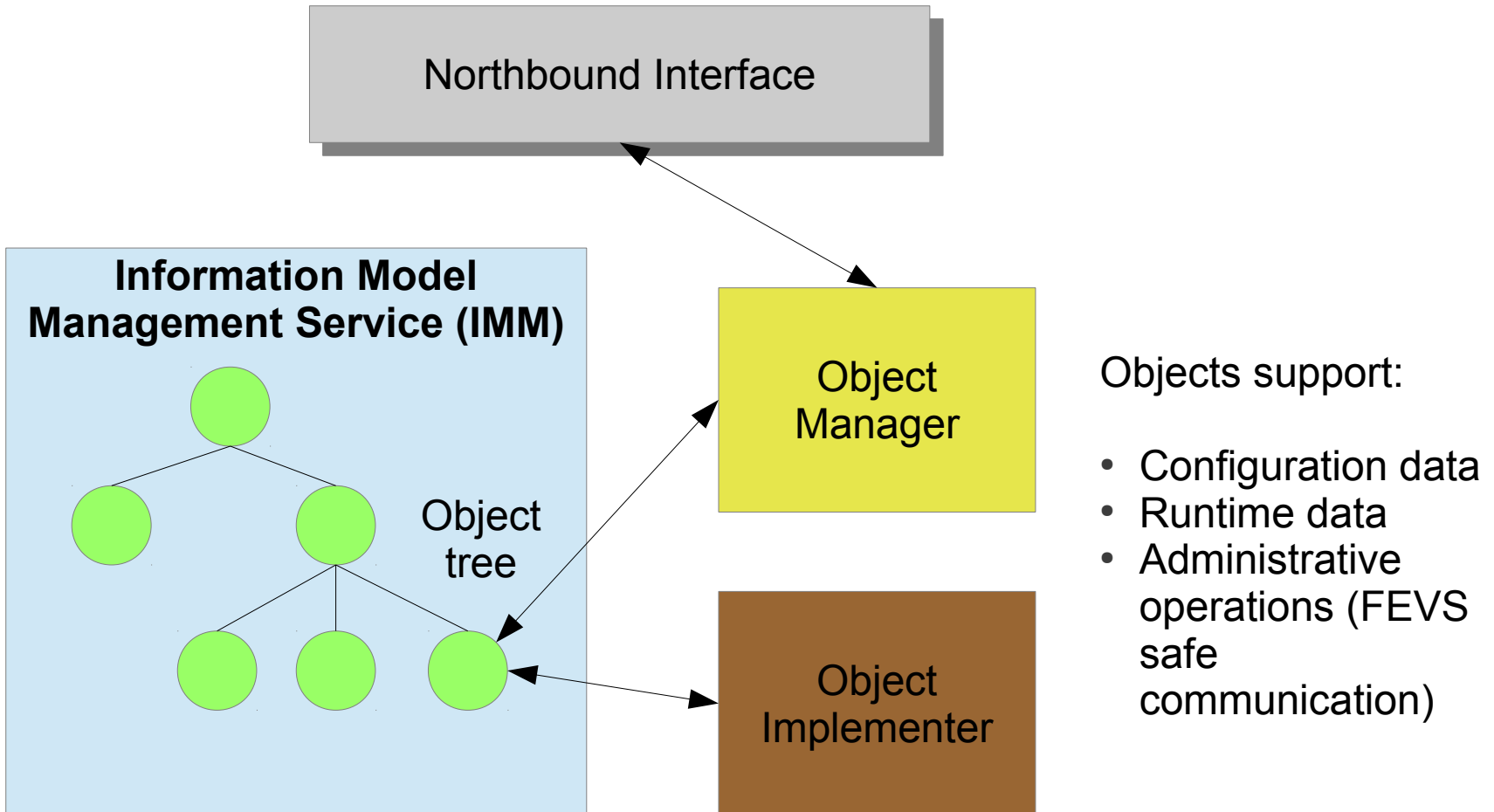- During upgrade, nodes can run different versions of OpenSAF (and *applications*)

OpenSAF Architecture

Information Model

IMM Persistent Backend (PBE)

System Controller 1

Local disk
imm1.db (2 PBEs)

System Controller 2

Local disk
imm2.db (2 PBEs)

Initial
Configuration
imm.xml (no PBE)

Shared disk
imm.db (1 PBE)

Three different ways to configure the persistent backend:

· No PBE – Initial configuration is reloaded at cluster restart
· 1 PBE – Shared disk is used
· 2 PBEs* – Each SC has a local copy of the PBE
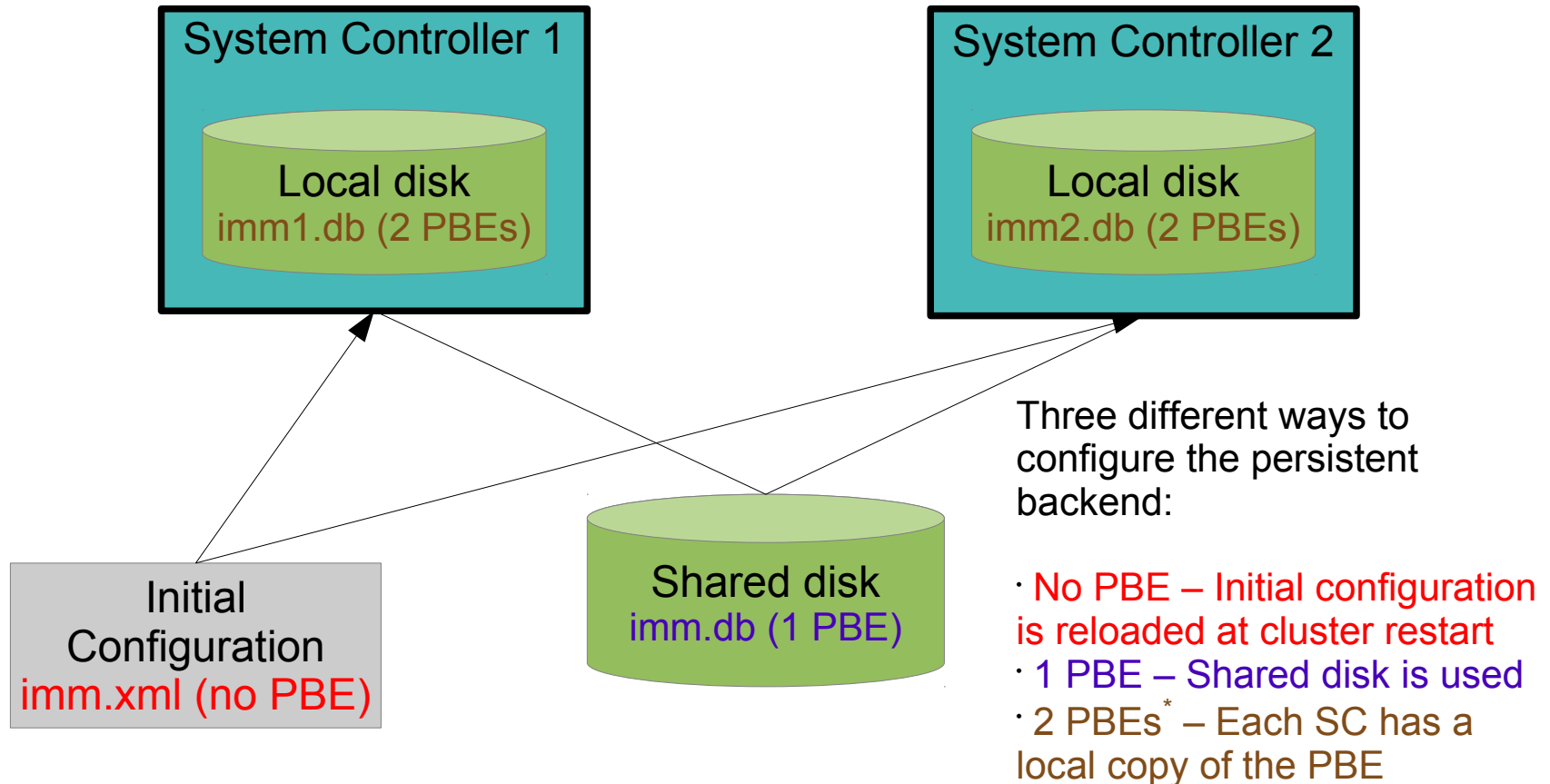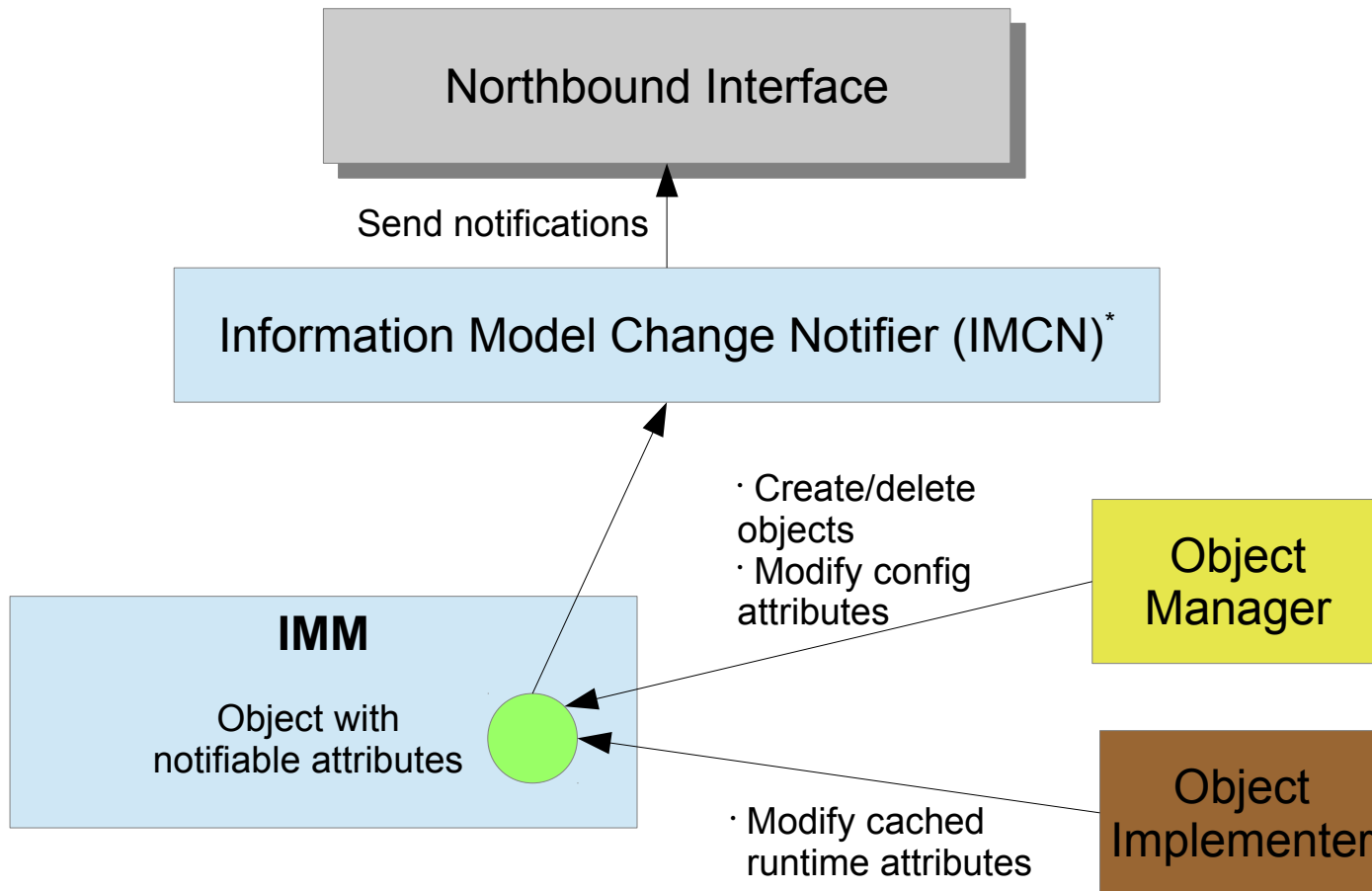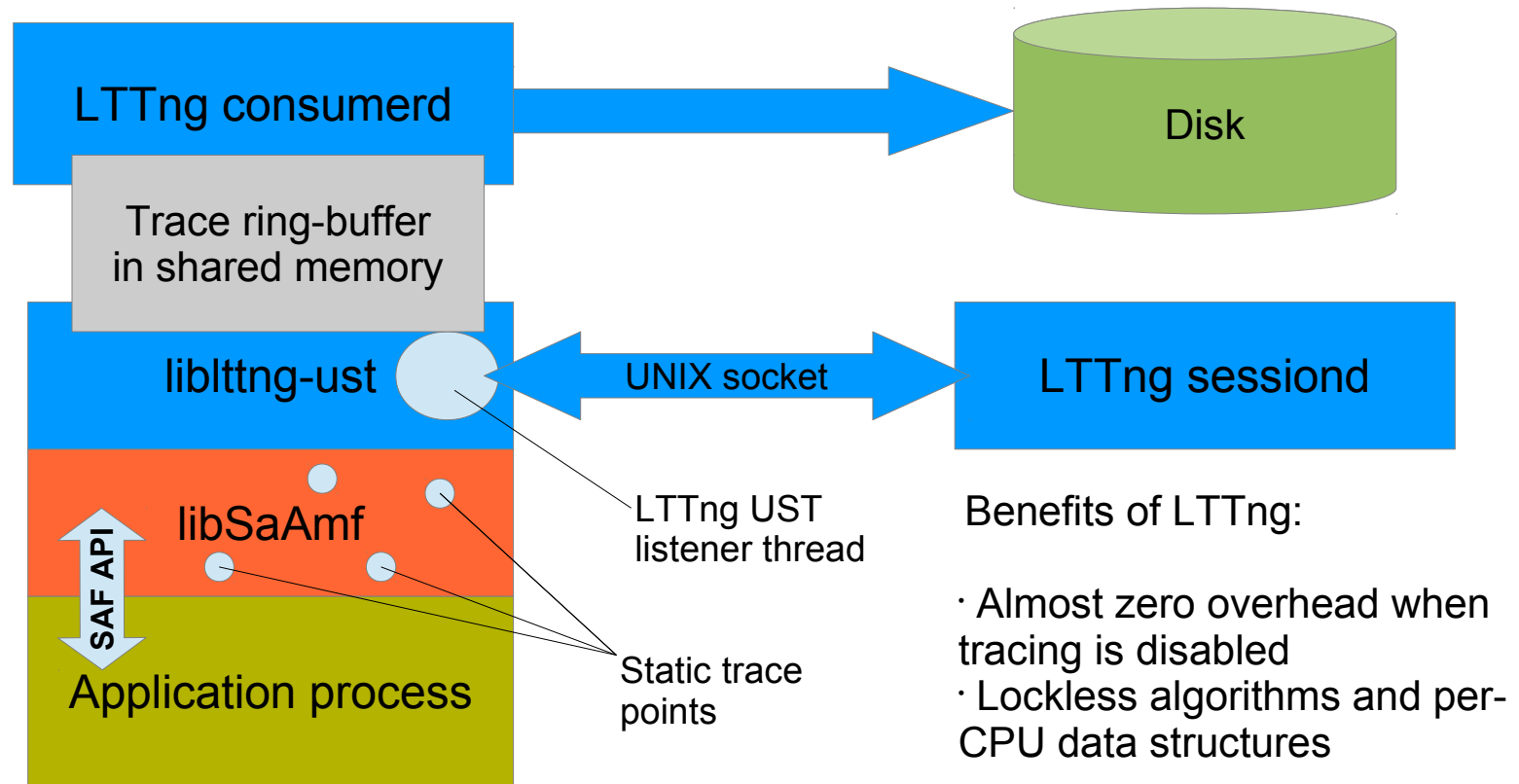
* 2 PBEs is a new feature in OpenSAF 4.4
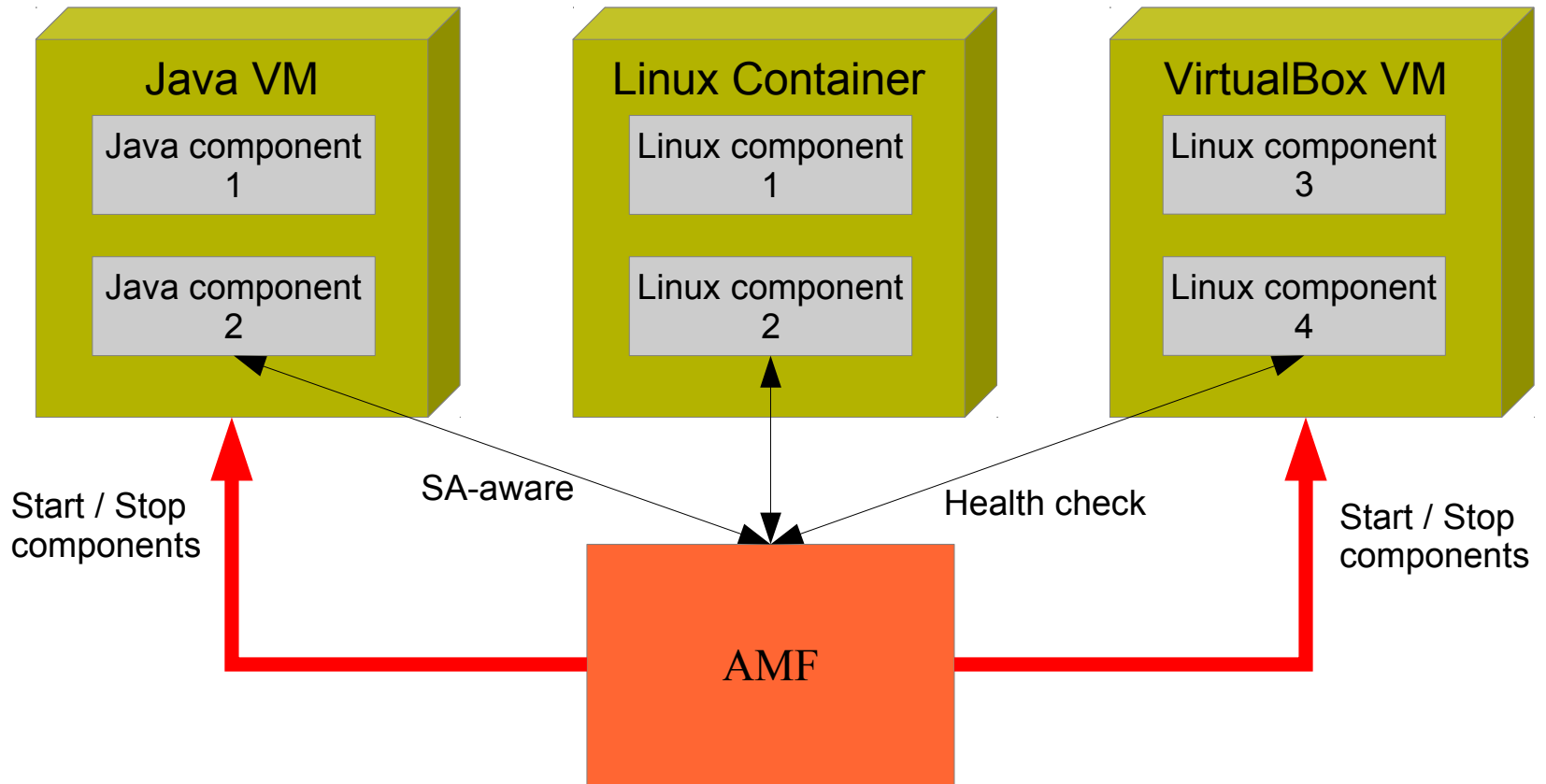
Information Model Notifications

* IMCN is a new feature in OpenSAF 4.3
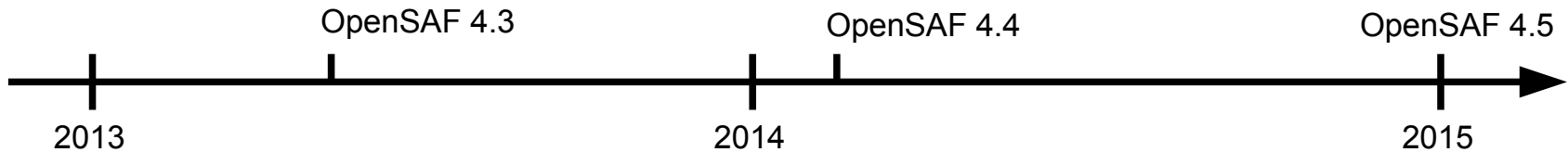
SAF API Trace[*] Using LTTng

* SAF API Trace is a new feature in OpenSAF 4.4

Container* Components

Java VM

Java component 1

Java component 2

Linux Container

Linux component 1

Linux component 2

VirtualBox VM

Linux component 3

Linux component 4

Start / Stop components

SA-aware

Health check

Start / Stop components

AMF

* Container / contained components is on the wish list for future releases

Roadmap

OpenSAF 4.3   OpenSAF 4.4   OpenSAF 4.5

2013   2014   2015

OpenSAF 4.4

- IMM two persistent backends
- IMM reference integrity checks
- SAF API trace using LTTng
- OpenSAF unit files for systemd
- LOG support systems without shared disk
- CKPT callback when active replica is updated

OpenSAF 4.5 and beyond

- AMF container / contained components
- AMF component readiness state
- CLM Enhanced cluster management
- NAM Implement SAF Naming Service
- Utilise Linux control groups
- Man pages (OpenSAF tools & SAF APIs)
- Complete Java SAF API bindings
- "Pythonic" Python SAF API bindings
- C++ SAF API bindings