OpenSAF™

The Open Service Availability Framework

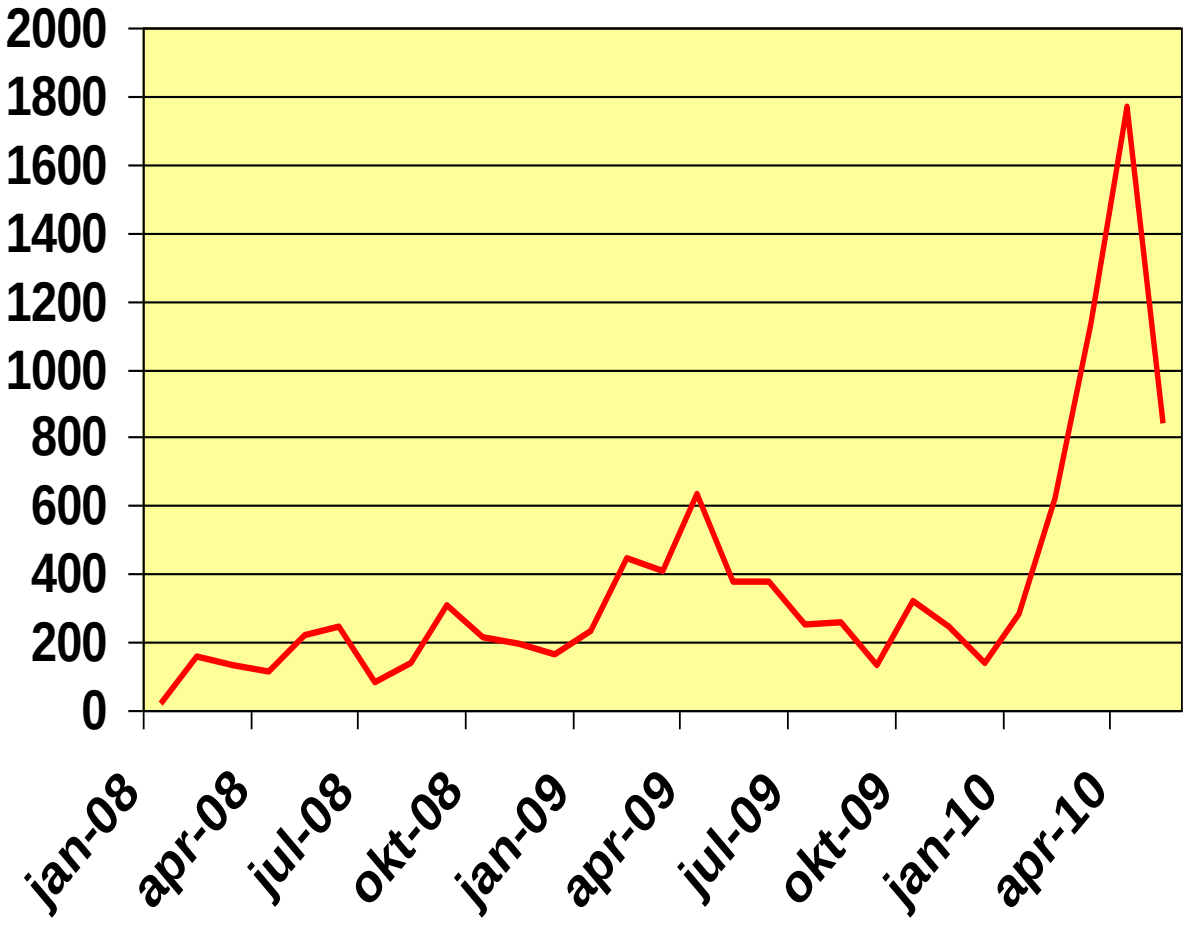
OpenSAF Release 4 Overview
"The Architecture Release"

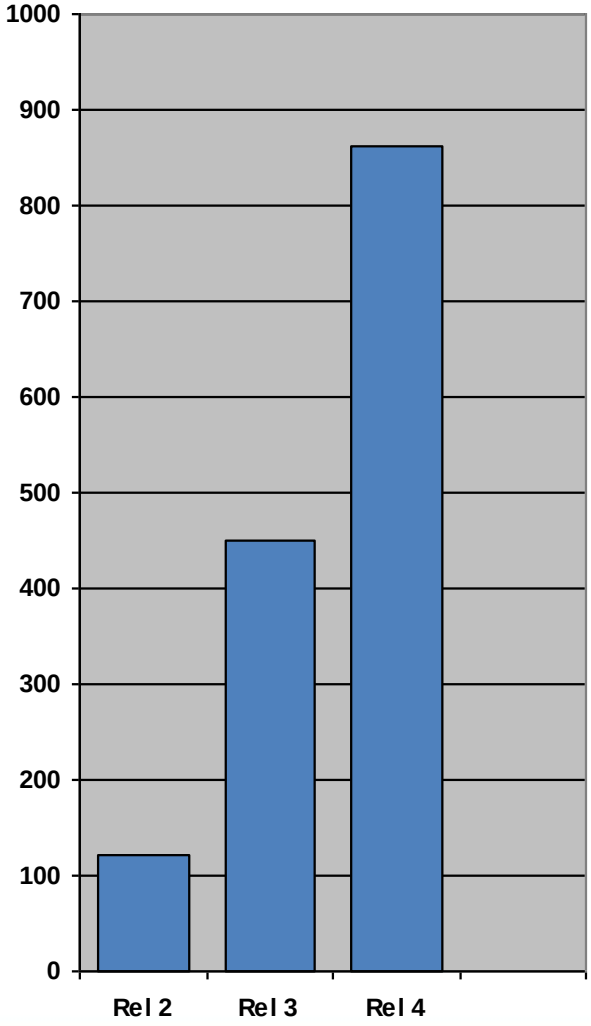
Mario Angelic
Expert Middleware Architectures, Ericsson
OpenSAF TLC
mario.angelic@ericsson.com

Presentation Outline

- Background Information
  - OpenSAF Release 3 Architecture Overview
- Release 4, “Architecture Release”
  - Functionality
  - Architecture Improvement
    - Streamlining
    - Modularity
    - Architecture alignment

First, a chart

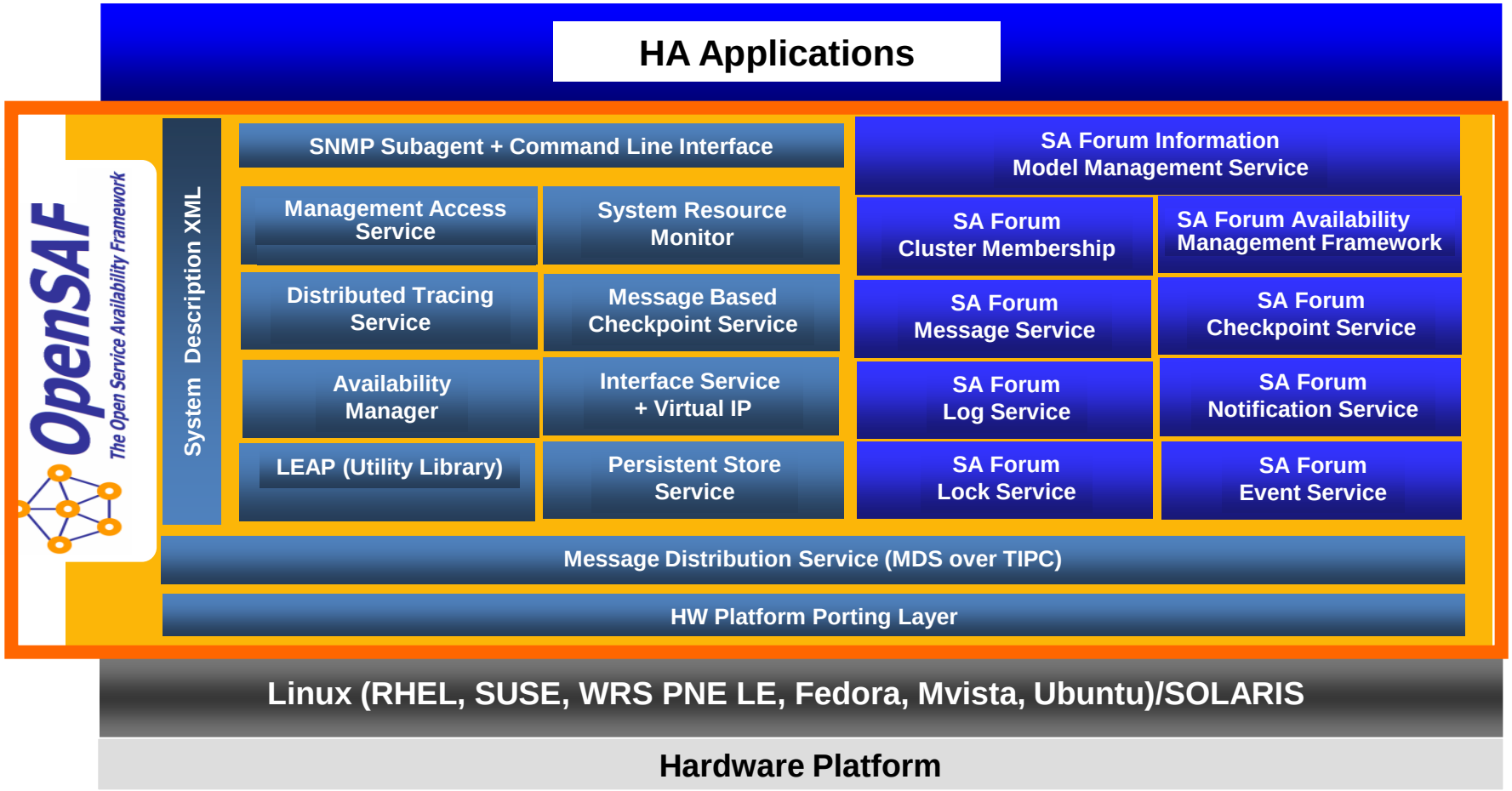




Presentation Outline

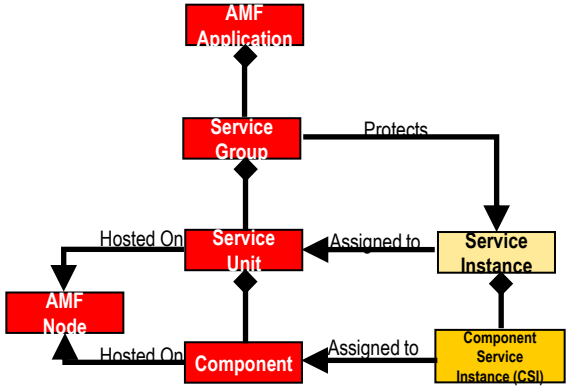
- Background Information
  - **OpenSAF Release 3 Architecture Overview**
- Release 4, “Architecture Release”
  - Functionality
  - Architecture Improvement
    - Streamlining
    - Modularity
    - Architecture alignment

OpenSAF 3.0 Key Features

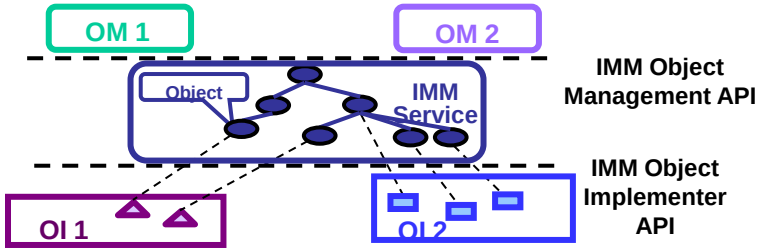


OpenSAF 3.0 key services

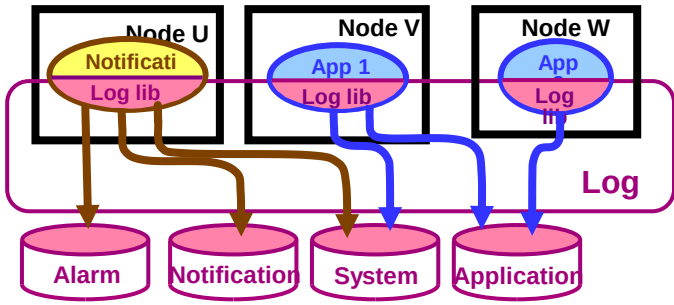
- Availability Management Framework
  - Manages **redundant service providers** for each **service**
    - instantiate, terminate and monitor **service providers**
    - Dynamically (re)assing **services** to **service providers**
    - Model driven



- Information Model Management Service
  - Allows **objects** of the **Information Model** to be created, accessed, and managed by system management applications

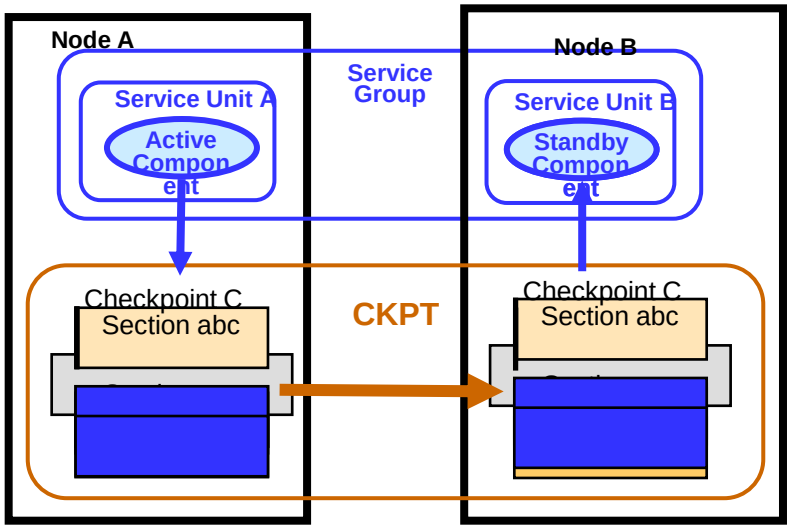


- Log Service
  - Enable application to express and forward **log records** through well-known **log streams** that lead to particular output destinations such as named files

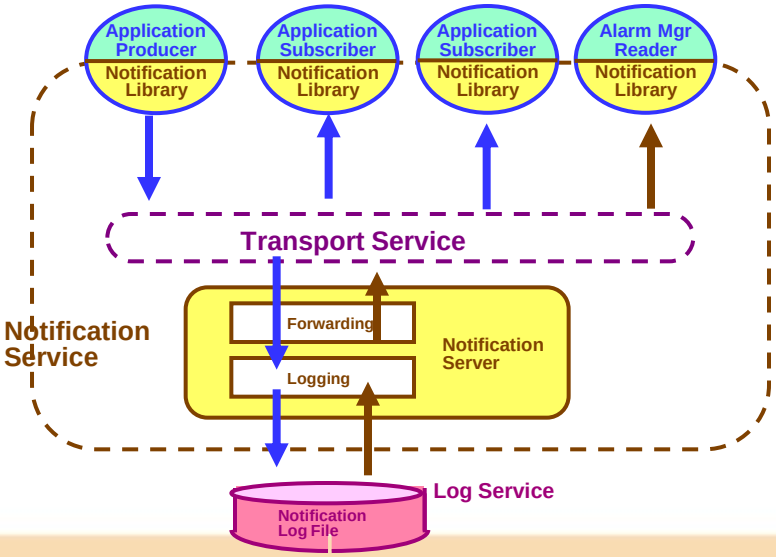


OpenSAF 3.0 key services

- Checkpoint Service
  - Manages checkpoints that a process uses to save its state to 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.



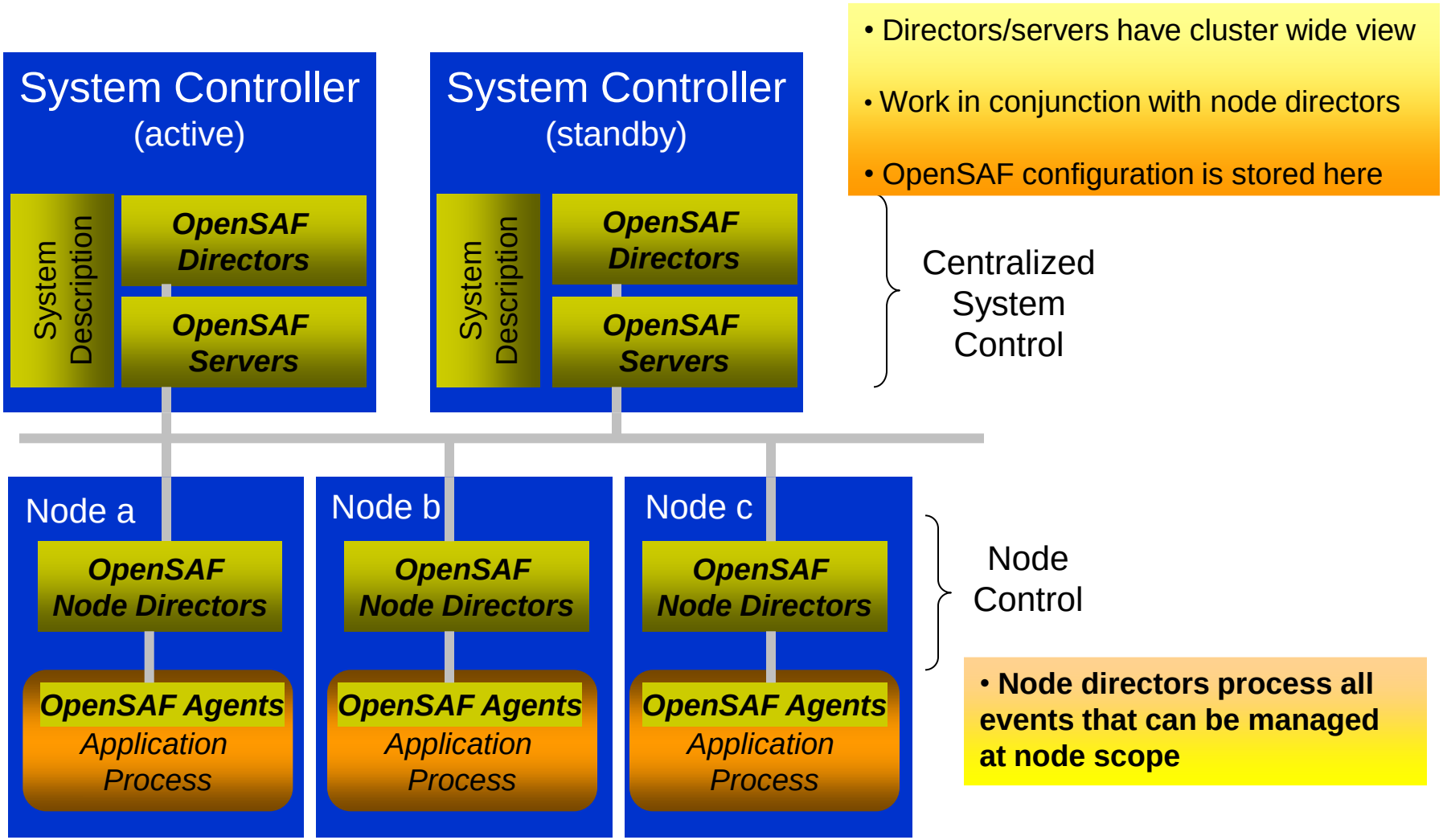
- Notification Service
  - Notification **producers** generate **notifications**
  - Notification **consumers** consume **notifications** generated by **producers**, and can be either of **subscriber** or **reader** type
  - Support for Notification filters



OpenSAF 3.0 key services

- Event Service
  - Publish/subscribe multipoint-to-multipoint communication mechanism based on cluster-wide event channels
- 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
- Message Service
  - Buffered message passing system, for processes on the same or different nodes, that is based on the concept of a message queue.

3/2-Tier OpenSAF Architecture



OpenSAF Basic Architectural Styles

2-Tier: *Server and Agent*	➢ SAF Event Service ➢ SAF Log Service ➢ SAF Notification Service ➢ OpenSAF Distributed Trace Service
3-Tier *Director, Node-Director and Agent*	➢ SAF Availability Management Framework ➢ SAF Cluster Membership Service ➢ SAF Checkpoint Service ➢ SAF Information Model Management Service ➢ SAF Message Service ➢ SAF Lock Service

Release 3 Architecture Issues

- Functional gaps
  - SMF, PLM, IMM Transactional Persistency
- Non-streamlining Architecture
  - Functionally overlapping services
    - Typically between SAF services and OpenSAF legacy services (Example MASv and IMM)
  - Focus on minimum number of core infrastructure services
  - Alignment in configuration and fault management area
  - Consolidation of logging
- Modularity
  - Architecture
  - Packaging

Presentation Outline

- Background Information
  - OpenSAF Release 3 Architecture Overview
- Release 4, “Architecture Release”
  - **Goals**
  - Functionality
  - Architecture Improvement
    - Streamlining
    - Modularity
    - Architecture alignment

Release 4 Goals

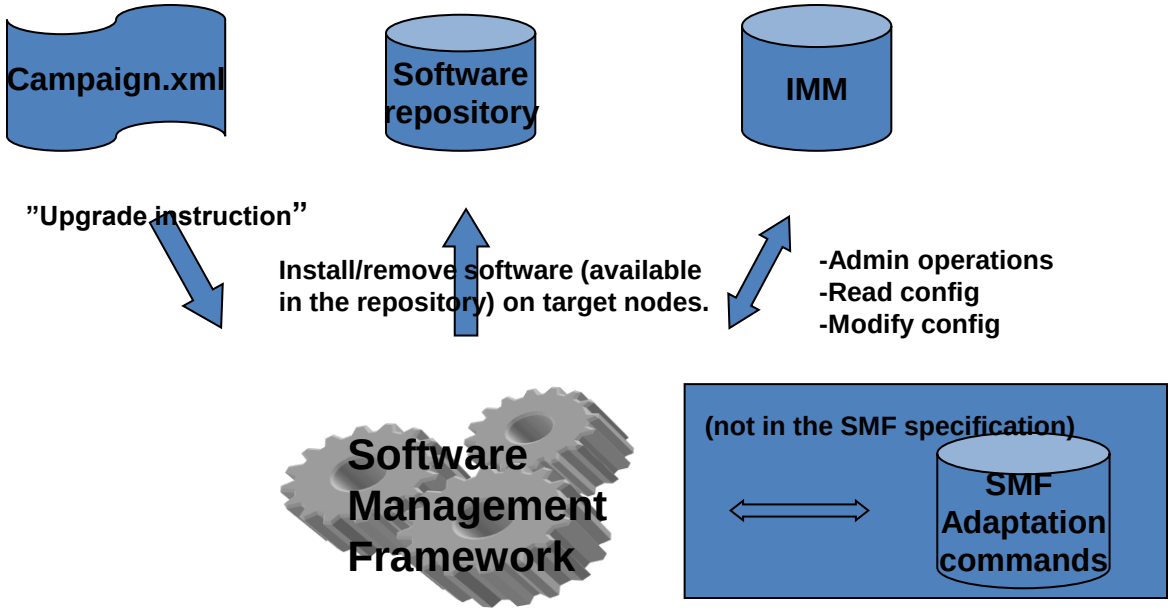
- Close major functional gaps
  - SMF, PLM, IMM Transactional Persistency
- Settle internal architecture
  - Enabler for in-service upgradeability from Release 4
  - Keep basic set of infrastructure services
    - Only those that really add value & needed by SAF services
    - Other infrastructure services for which there exist better open-source alternative are removed (focus on added-value)
- Clearly distinguish between public API and internal infrastructure services
- Deliberate decision to not support in-service upgradeability between Release 3 and Release 4
  - From Release 4 OpenSAF will support in-service upgradeability between releases

Presentation Outline

- Background Information
  - OpenSAF Release 3 Architecture Overview
- Release 4, “Architecture Release”
  - Goals
  - **Functionality**
  - Architecture Improvement
    - Streamlining
    - Modularity
    - Architecture alignment

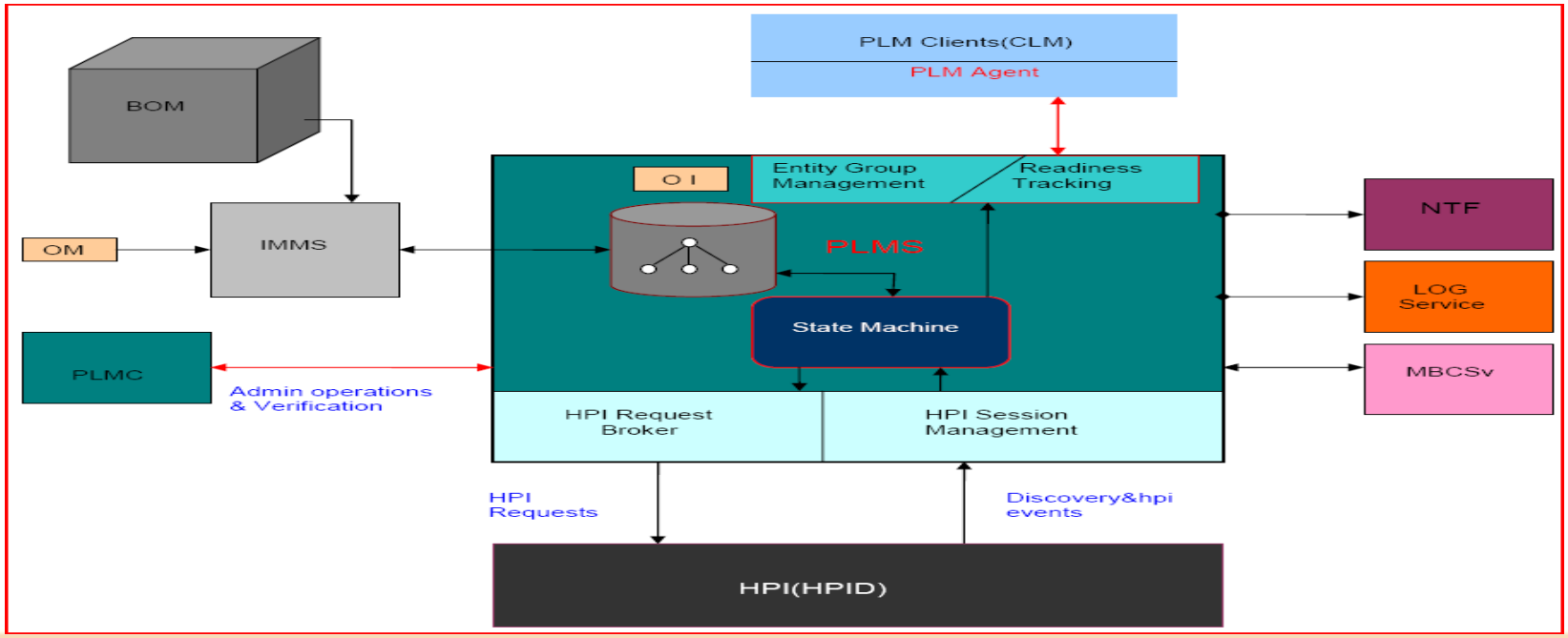
Software Management Framework

- Migrating a target system in operation from one deployment configuration to another (**software upgrade**), is realized following an **upgrade campaign** specification
- Upgrade can be done without loss of service (**rolling upgrade**) or with loss of service (**single-step upgrade**)
- Maintains **software catalog**
  - contains information about the available software entity types in the system, their versions, and references to the **software bundles** that delivered them and to the entities that deploy them.



Platform Management Service

- Provides a logical view of the hardware and low-level software of the system.
  - Low-level software in this sense comprises the operating system and virtualization layers.
- The main logical entities implemented by the PLM Service are:
  - Execution Environment (EE)
    - An EE is a logical entity that represents an environment capable of running software.
  - Hardware Element (HE)
    - An HE is a logical entity that represents any kind of hardware entity, which can be, for instance, a chassis, a blade, or an I/O device.
- PLM maps discovered entities representing HW management and Execution environment and configured once.



IMM Transactional Persistency

- In Release 3 IMM implements in-memory “persistency”, and support for dumping state to a file (“backup”)
  - In case of total cluster restart state is read from last backup)
- In Release 4 a full transactional persistency is implemented
  - Feature is disable by default
  - If enabled
    - during build

      ```
      configure –enable-imm-pbe
      ```

    - In target configuration

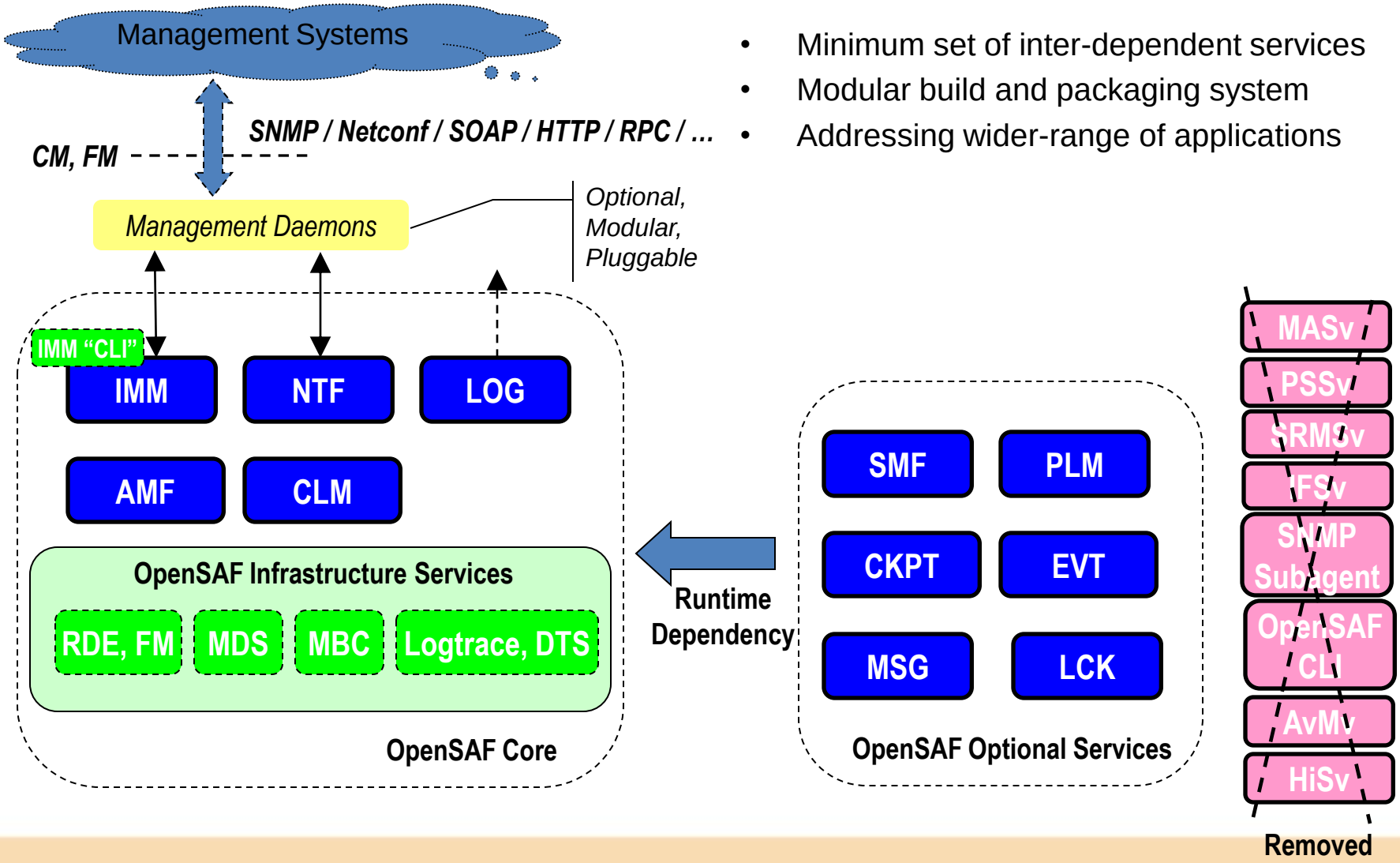
      ```
      immcfg –m –a saImmRepositoryInit=1, safRdn=immManagement,\
           safApp=safImmService
      ```

    - IMM Directors use SQLite to store configuration persistently on File System

Presentation Outline

- Background Information
  - OpenSAF Project & Technical Overview
- Release 4, “Architecture Release”
  - Functionality
  - **Architecture Improvement**
    - Streamlining
    - Modularity
    - Architecture alignment
- Beyond Release 4

OpenSAF 4.0 Architecture



- Minimum set of inter-dependent services
- Modular build and packaging system
- Addressing wider-range of applications

Build System

- Adapted to support optional services during build phase (**`configure`**)
- Each services packaged in own RPMs
  - 3-tier => <service>-nodedirector, <service>director, <service>-libs
  - 2-tier => <service>-server, <service>-libs
  - “Meta”-packages: opensaf-controller & opensaf-payload
- Many changes to adjust build system to different functional content and structure of OpenSAF

IMM alignment

- All services changed to use IMM for configuration instead of MASv
  - AMF (B.04 model), EVT, CKPT, LCK, DTSv
- IMM Node DIrector “resurection” support

NTF usage alignment

- AMF adapted to send all notification via NTF
  - Previously used EVT service

NTF improvements

- NTF Filtering
- Discarded notification support

AMF

- AMF adapted to use IMM and NTF
- SCP split in two processes (amfd & amfnd)
- AMF B.04 compliant model
  - Note: Still AMF B.01 level API
- Relaying on CLM for cluster membership
- Streamlined “heartbeating” and MW daemon supervision
- Local AMF Monitor (per node) supervising AvND

CLM

- Background: In Release 3 CLM functionality was bundled with AMF in AvSv service
- In Release 4 CLM was “lifted out” as standalone services
- Uplifted to latest release of CLM specification
- Relaying on TIPC

IMM “CLI”

- Small set of Linux shell commands to manipulate IMM content:
  - **`immcfg`**
  - **`immadm`**
  - **`immlist`**
  - **`immfind`**
  - **`immdump`**
- Useful for testing (scripting)
  - Possible to do any IMM changes, queries without “heavy” management application

IMM CLI: immcfg

SC-1

```
# immcfg --help

NAME
        immcfg - create, delete or modify IMM configuration object(s)

SYNOPSIS
        immcfg [options] [object DN]...

DESCRIPTION
        immcfg is an IMM OM client used to create, delete an IMM or modify attribute(s) for IMM object(s)
        The default operation if none specified is modify.
        When creating or modifying several objects, they have to be of the same class.
OPTIONS
        -a, --attribute name[+|-]=value [object DN]...
        -c, --create-object <class name> [object DN]...
        -d, --delete-object [object DN]...
        -h, --help                      this help
        -m, --modify-object [object DN]...
        -v, --verbose (only valid with -f/--file option)
        -f, --file <imm.xml file containing classes and/or objects>
        --delete-class <classname> [classname2]...

EXAMPLE
        immcfg -a saAmfNodeSuFailoverMax=7 safAmfNode=Node01,safAmfCluster=1
                change one attribute for one object
        immcfg -c SaAmfApplication -a saAmfAppType=Test safApp=myTestApp1
                create one object setting one initialized attribute
        immcfg -d safAmfNode=Node01,safAmfCluster=1
                delete one object
        immcfg -d safAmfNode=Node01,safAmfCluster=1 safAmfNode=Node02,safAmfCluster=1
                delete two objects
        immcfg -a saAmfNGNodeList+=safAmfNode=PL_2_6,safAmfCluster=myAmfCluster safAmfNodeGroup=PLs,safAmfCluster=myAmfCluster
                add a value to an attribute
        immcfg -a saAmfNGNodeList-=safAmfNode=PL_2_6,safAmfCluster=myAmfCluster safAmfNodeGroup=PLs,safAmfCluster=myAmfCluster
                remove a value from an attribute
#
```

IMM CLI: immadm

SC-1

```
# immadm --help

NAME
        immadm - perform an IMM admin operation

SYNOPSIS
        immadm [options] [object DN]...

DESCRIPTION
        immadm is a IMM OM client used to ....

OPTIONS
        -h, --help
                this help
        -o, --operation-id <id>
                numerical operation ID (mandatory)
        -p, --parameter <p>
                parameter(s) to admin op
                Parameter syntax: <name>:<type>:<value>
                Value types according to imm.xsd.
                Valid types: SA_INT32_T, SA_UINT32_T, SA_INT64_T, SA_UINT64_T
                        SA_TIME_T, SA_NAME_T, SA_FLOAT_T, SA_DOUBLE_T, SA_STRING_T

EXAMPLE
        immadm -o 1 -p saAmfNodeSuFailoverMax:SA_INT32_T:7 safAmfNode=Node01,safAmfCluster=1
#
```

IMM CLI: immfind

SC-1

```
# immfind --help

NAME
        immfind - search for IMM objects

SYNOPSIS
        immfind [path ...] [options]

DESCRIPTION
        immfind is an IMM OM client used to find IMM objects.
        All objects or objects of a certain class can be searched for.

OPTIONS
        -c, --class=NAME
                only search for objects of the specified class
        -s, --scope=SCOPE
                specify search scope, valid scopes: sublevel subtree
        -h, --help
                this help

EXAMPLE
        immfind
                search for all objects
        immfind safApp=myApp
                search for all objects rooted under safApp=myApp
        immfind safApp=myApp -s sublevel
                search for all objects rooted under safApp=myApp scope sublevel
        immfind safApp=myApp --scope subtree
                search for all objects rooted under safApp=myApp scope subtree
        immfind -c SaAmfApplication
                search for all objects of class SaAmfApplication
# 
```

IMM CLI: immlist

SC-1

```
# immlist --help

NAME
        immlist - list IMM objects

SYNOPSIS
        immlist [options] <object name> [object name]

DESCRIPTION
        immlist is an IMM OM client used to print attributes of IMM objects.

OPTIONS
        -a, --attribute=NAME
        -h, --help - display this help and exit
        -p, --pretty-print=<yes|no> - select pretty print, default yes

EXAMPLE
        immlist -a saAmfSUPresenceState safApp=OpenSAF
        immlist safApp=myApp1 safApp=myApp2
        immlist --pretty-print=no saAmfSUPresenceState safApp=OpenSAF
# 
```

IMM CLI: immdump

SC-1

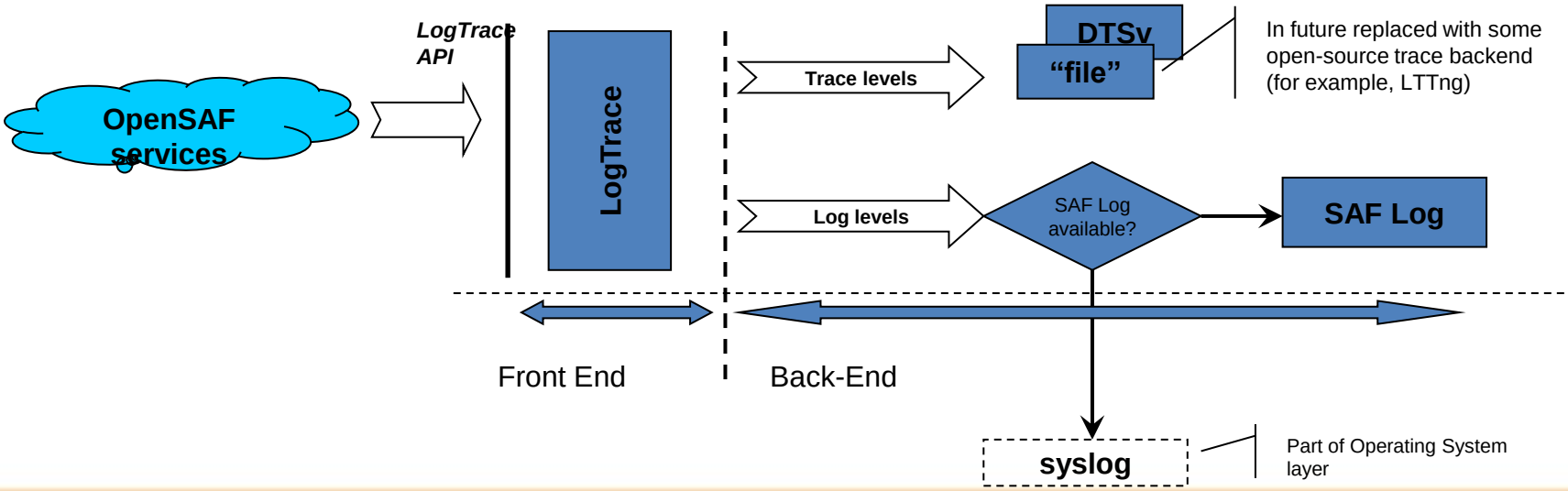
```
# immdump --help

NAME
        immdump - dump IMM model to file

SYNOPSIS
        immdump <file name>

DESCRIPTION
        immdump is an IMM OM client used to dump, write the IMM model to file

OPTIONS
        -h, --help
                this help

        -p, --pbe   {<file name>}
                Instead of xml file, generate/populate persistent back-end database/file

EXAMPLE
        immdump /tmp/imm.xml
#
```

Configuration usability support

- Support for Middleware (OpenSAF)
- Initial configuration adapted to cluster size
  - Done in modular way
    - Each service delivers own configuration template fragment
    - Configuration is merged (depending on installed services)
    - Configuration is “instantiated” (adjusted to cluster size)
- Extending the cluster by N blades
- Shrinking the cluster by N blades

Logging Improvements

- In Release 3, OpenSAF have several means of logging information:
  - Stdout redirected to files
  - Per service log files
  - Using DTS service
  - Using syslog
  - Using LogTrace
- In Release 4, focus on:
  - Using LogTrace (mapping to SAF Log, syslog, trace backend)
- Still some work to do for subsequent release



Beyond Release 4

Focus areas:

a) **“Architecture” => Release 4**

b) “Usability”

c) “Ecosystem”

***Note**: Areas will be covered in presentation “OpenSAF Project Roadmap” tommorow.*

Questions ?

Thank You!

For more information:

http://devel.opensaf.org