GOAHEAD

*embedded*MIND
OpenSAFfire Integrated Solution

Tomasz Mikolajczyk
GoAhead Software

OpenSAFfire Integrated Solution

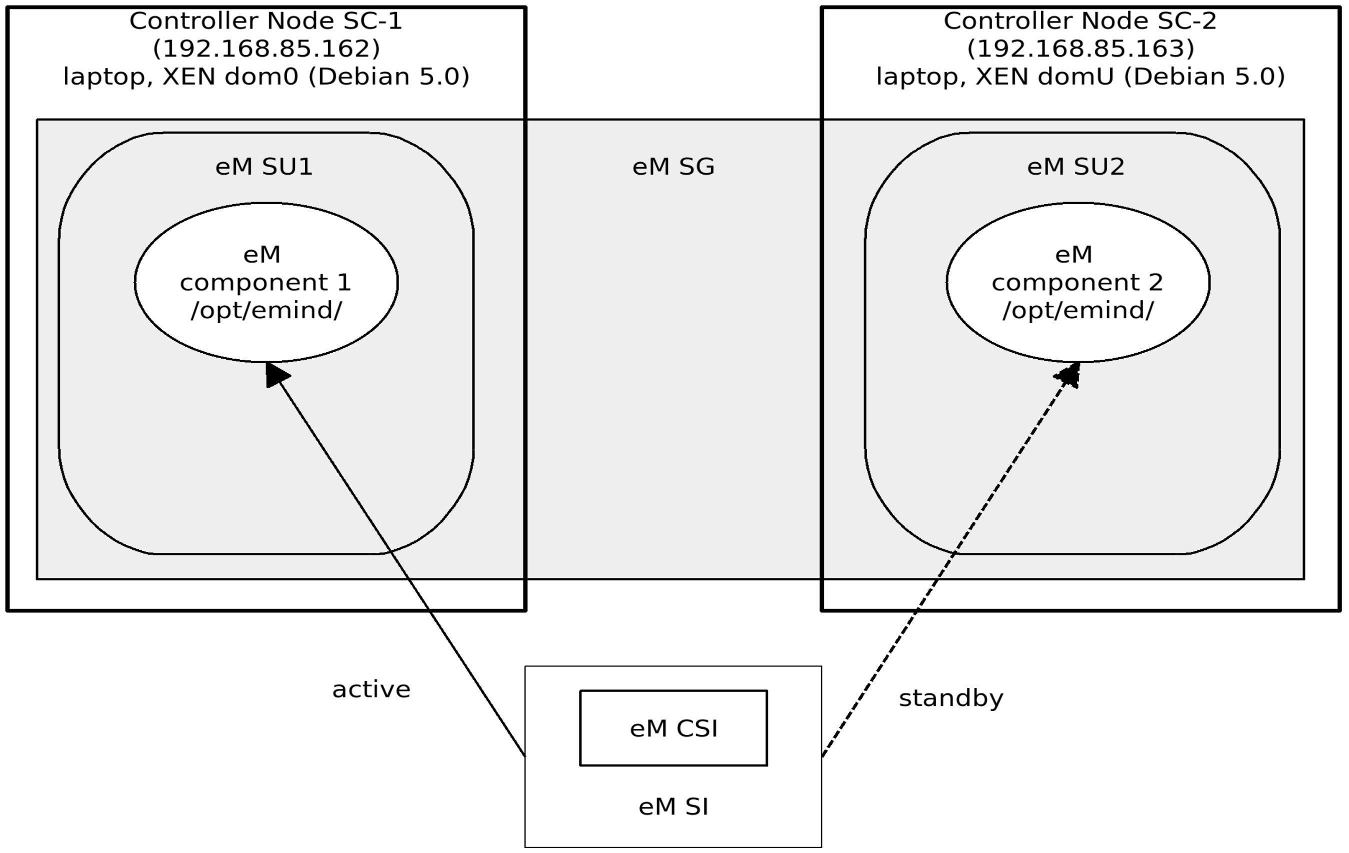
- embeddedMIND (eM) is a **M**anagement **I**nfrastructure for **N**etworked **D**evices.
- Providing a Management Layer for a cluster.
- Operating on the cluster through the standard management protocols:
  - CLI
  - NETCONF
  - WEB/XML
  - SNMP
- Creating a cluster's model using MINDConstructor Modeling GUI tool (Eclipse-based IDE).
- Integration with the forthcoming OpenSAFfire 6.0 (GoAhead distributed version of OpenSAF 4.0).

Integrated Services

- AMF Service
- LOG Service
- NTF Service
- IMM Service

Demo Cluster Structure

eM component working in the 2N-redundancy model.



GOAHEAD

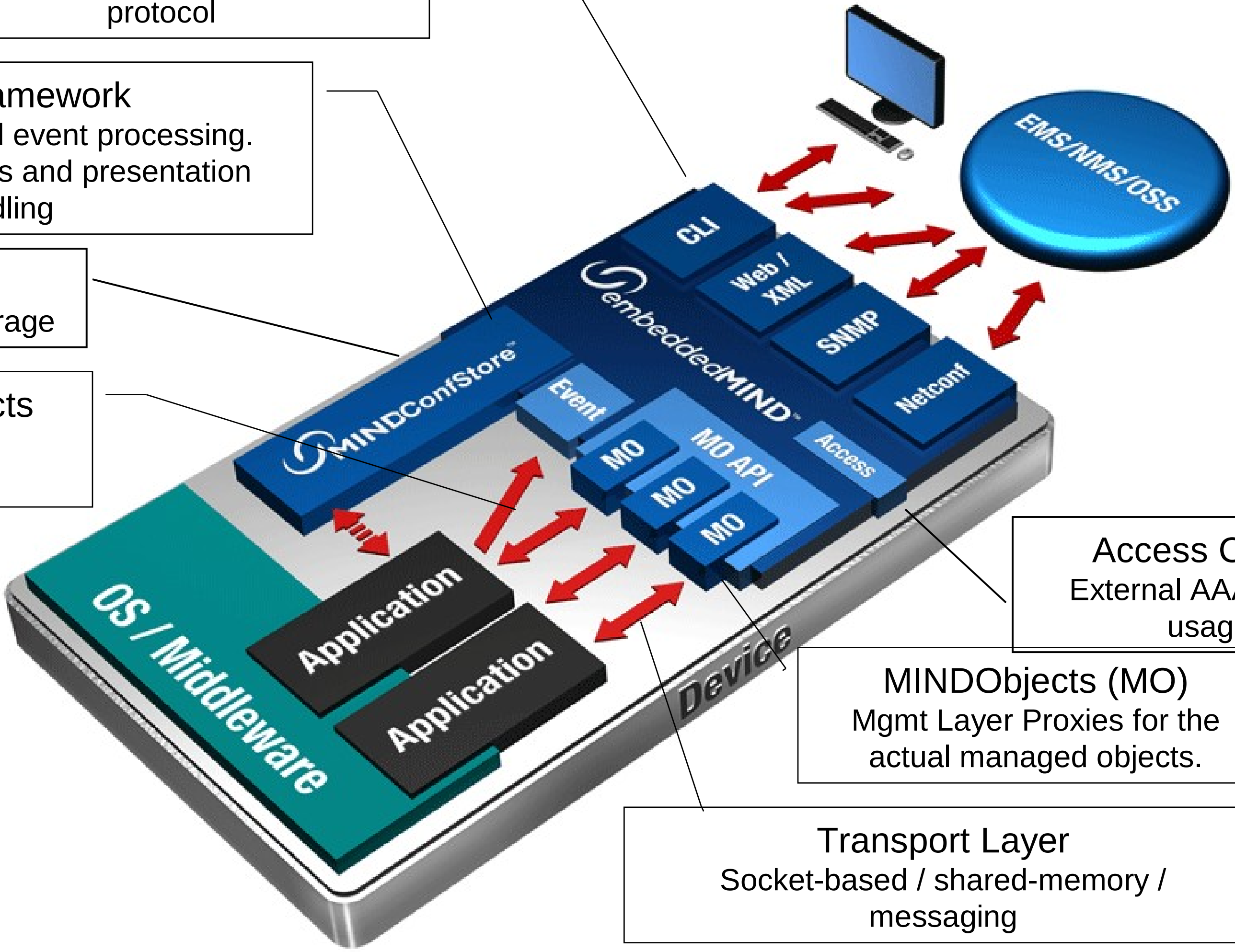
System Architecture

MINDAgents
Makes MO available via corresponding management protocol

MINDFramework
Unified request and event processing. Transaction, access and presentation handling

Database
Configuration Storage

Managed Objects
Implementation
[Validation]

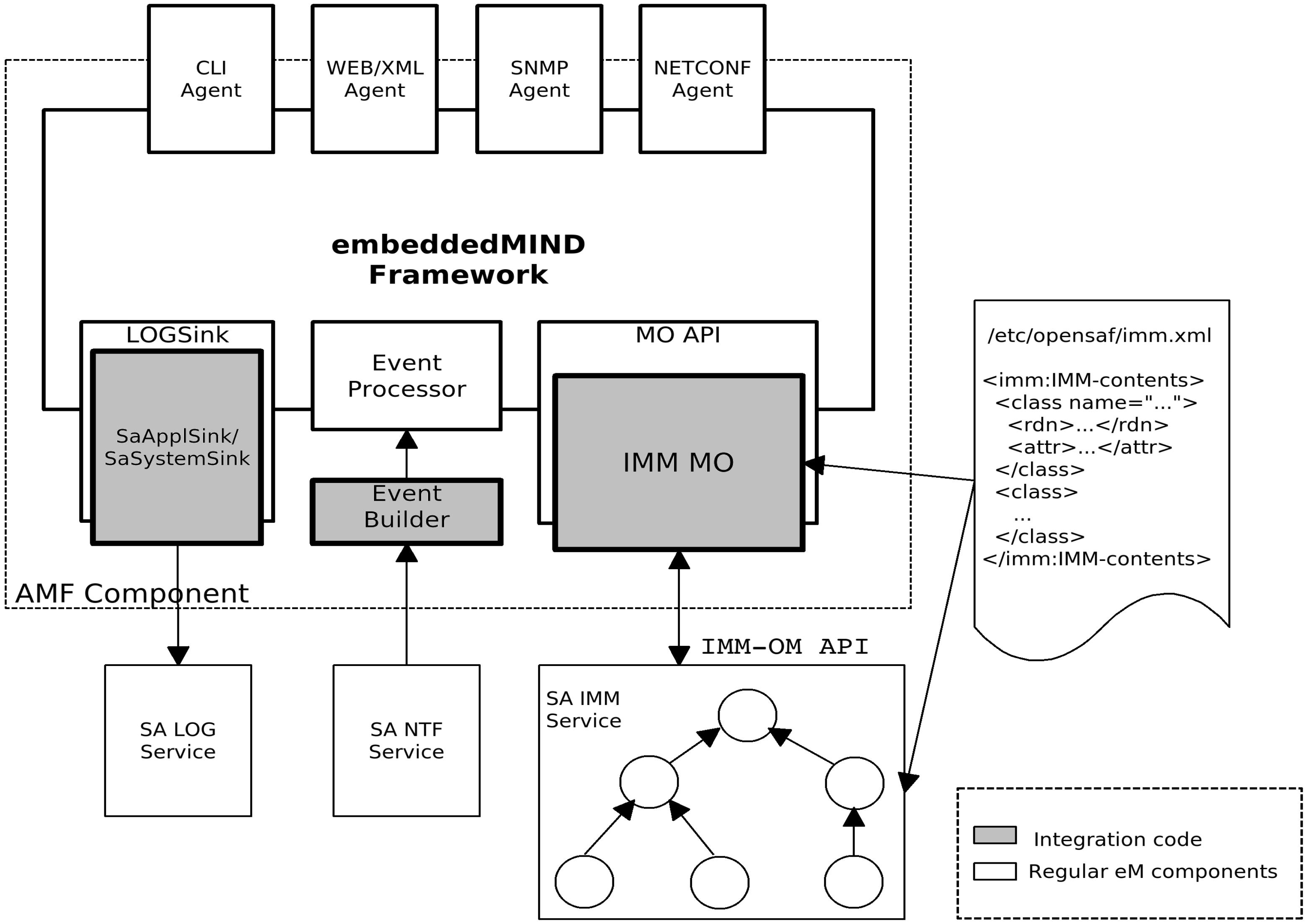


Access Control
External AAA service usage.

MINDObjects (MO)
Mgmt Layer Proxies for the actual managed objects.

Transport Layer
Socket-based / shared-memory / messaging

Integration Architecture Diagram



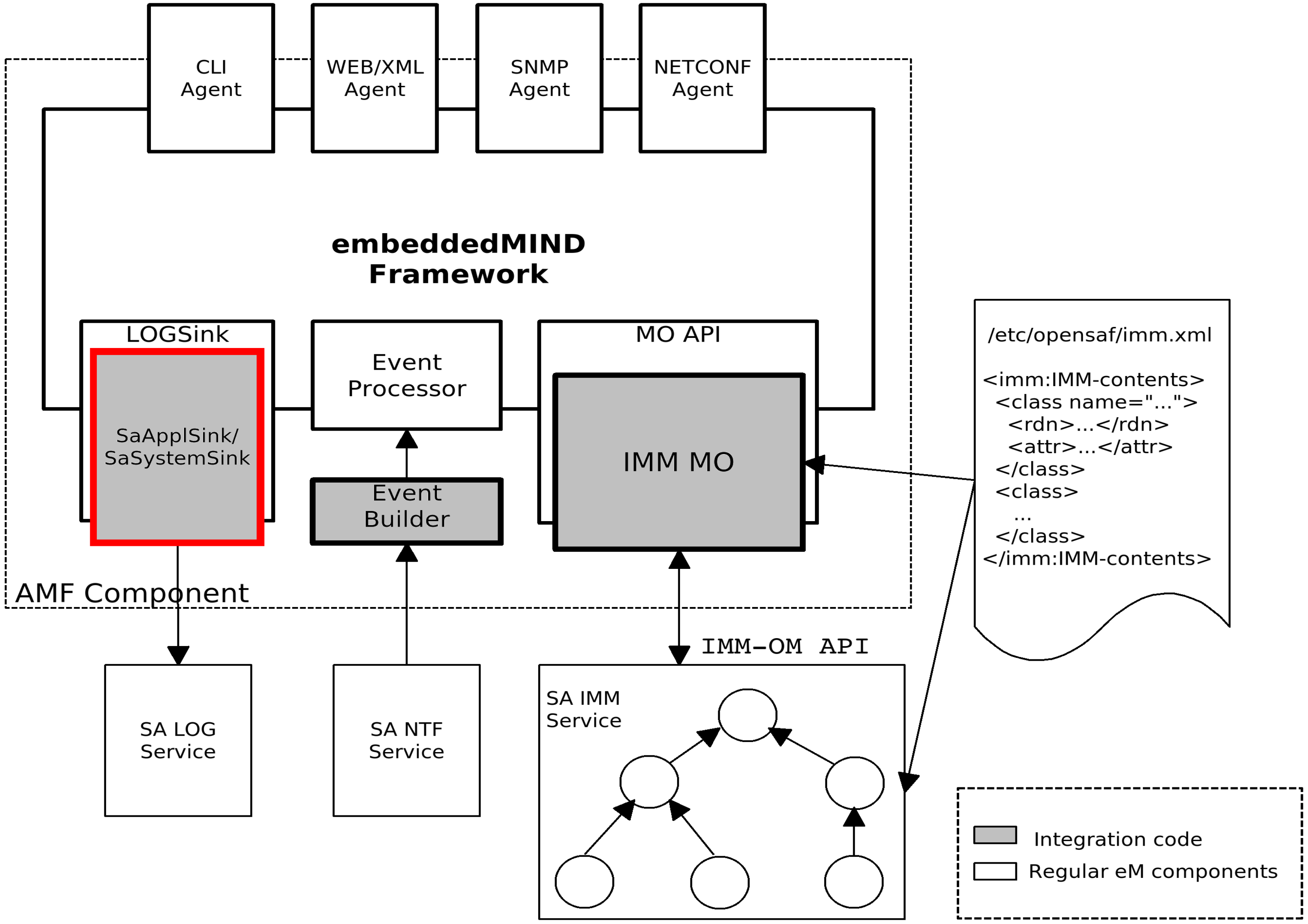
```
/etc/opensaf/imm.xml

<imm:IMM-contents>
  <class name="...">
    <rdn>...</rdn>
    <attr>...</attr>
  </class>
  <class>
    ...
  </class>
</imm:IMM-contents>
```

Integration with AMF

- Providing eM as a highly available management layer component.
- Working as a local SA-aware component.
- Using 2-N redundancy model as a default environment.
- Providing the IMM model part describing the eM component.

NOTE: eM does not preserve the state of managed applications.

Integration with LOG Service (1/4)



Integration with LOG Service (2/4)

Making meaningful information about the operation of eM available in the SA Log service.

Achieved by writing eM log messages to various SA Log Streams:

- SA System Log Stream:

  */var/log/opensaf/saflog/saLogSystem_** files.

- A dedicated eM SA Application Log Stream:

  */var/log/opensaf/saflog/eMIND_** files.

A SA-aware implementation of the eM LogSink:

- SASystemLogSink
- SAApplicationLogSink

Integration with LOG Service (3/4)

eM log data priorities:

eM Log Data Priority	SA Log Data Severity
prioCritical	SA LOG SEV CRITICAL
prioError	SA LOG SEV ERROR
prioWarning	SA LOG SEV WARNING
prioInfo	SA LOG SEV INFO
prioDebug	SA LOG SEV NOTICE

eM log filtering mechanism:

- Filtering log data using different filtering criteria (adjustable at runtime), for instance:
  - Priority filter
  - Module filter
  - Custom filter

GOAHEAD

Integration with LOG Service (4/4)

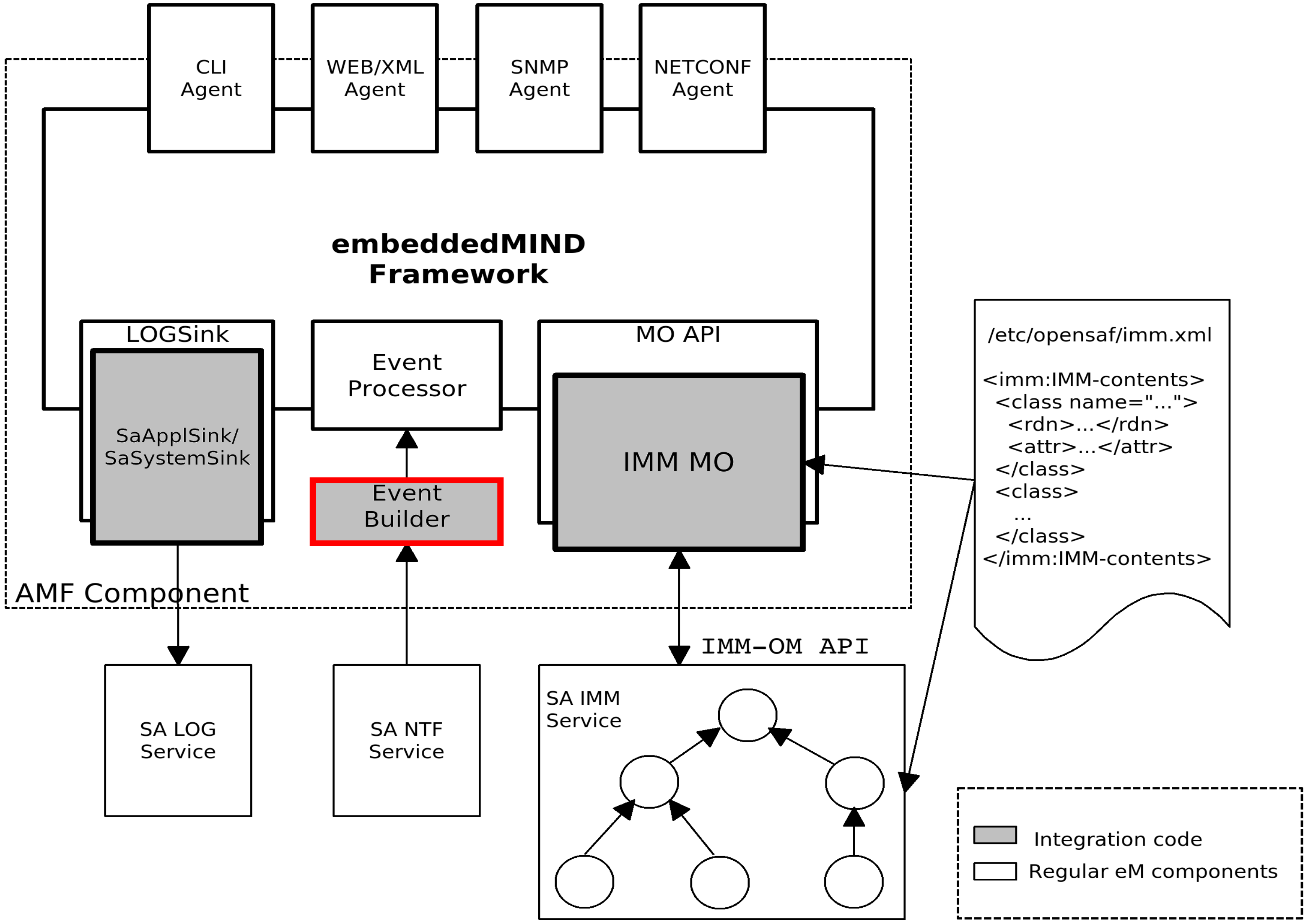
Example usage:

```
void amfCSISetCallback(SaInvocationT inv, const SaNameT* comp, SaAmfHAStateT haState,
                       SaAmfCSIDescriptorT csiDesc)
{
    mainLog.setPriority(embeddedMIND::prioInfo) << "AMF [" << inv << "]: Dispatched 'CSI Set'. COMP name: "
        << (comp ? SaNameT_toString(*comp) : "")
        << ". CSI name: " << csiDesc.csiName << ". HA state: " << enum_toString(ha_state)
        << ". CSI flags: " << csiDesc.csiFlags << std::endl;
    // ...
}
```

'*cat /var/log/opensaf/saflog/saLogSystem_20100517_174850.log*'

```
45 17:49:05 05/17/2010 IN embeddedMIND "AMF [4280287235]: Dispatched 'CSI Set'. COMP name:
safComp=embeddedMIND,safSu=SU1,safSg=embeddedMIND,safApp=embeddedMIND. CSI name: safCsi=embeddedMIND
46 17:49:05 05/17/2010 IN embeddedMIND "CLI Listening on 0.0.0.0:11001, 15000 (eCLI)
```

Integration with NTF Service (1/5)



```
/etc/opensaf/imm.xml

<imm:IMM-contents>
  <class name="...">
    <rdn>...</rdn>
    <attr>...</attr>
  </class>
  <class>
    ...
  </class>
</imm:IMM-contents>
```

Integration with NTF Service (2/5)

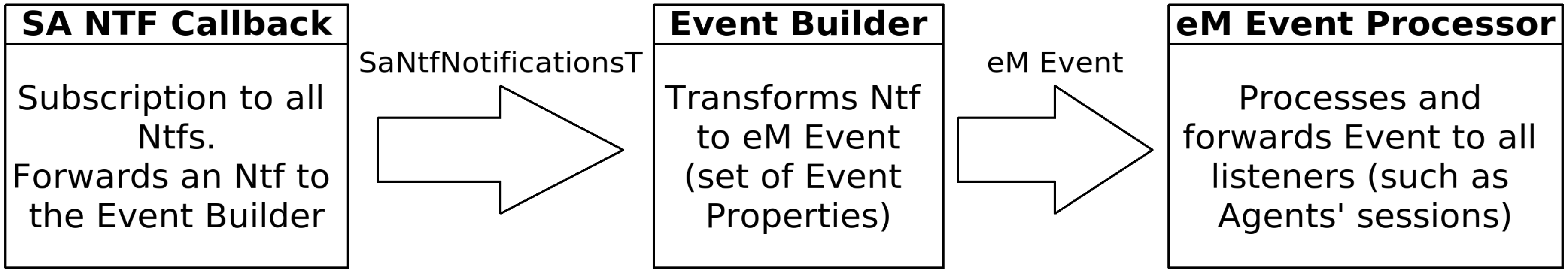
Making SA notifications available through the standard management interfaces (NETCONF, SNMP, CLI, WEB/XML).

Support for all SA Notification types:

- SA NTF TYPE OBJECT CREATE DELETE
- SA NTF TYPE ATTRIBUTE CHANGE
- SA NTF TYPE STATE CHANGE
- SA NTF TYPE ALARM
- SA NTF TYPE SECURITY ALARM

Integration with NTF Service (3/5)

Notification's flow:


SA NTF Callback
Subscription to all Ntfs. Forwards an Ntf to the Event Builder
SaNtfNotificationsT
Event Builder
Transforms Ntf to eM Event (set of Event Properties)
eM Event
eM Event Processor
Processes and forwards Event to all listeners (such as Agents' sessions)

eM Event filtering mechanism:

- Filtering an Event based on listener's filtering criteria (adjustable at runtime).

Integration with NTF Service (4/5)

CLI Agent Session:

- Processing and decorating an event based on a session's event formatter.
- Sending formatted event to a client's remote console (telnet/SSH).

NETCONF Agent Session:

- Processing and decorating an event based on filtering criteria passed in the <*create-subscription/*> command.

WEB Agent:

- Polling from a Web browser for an event.
- Decorating the received event (XML document) based on a customizable JavaScript code.

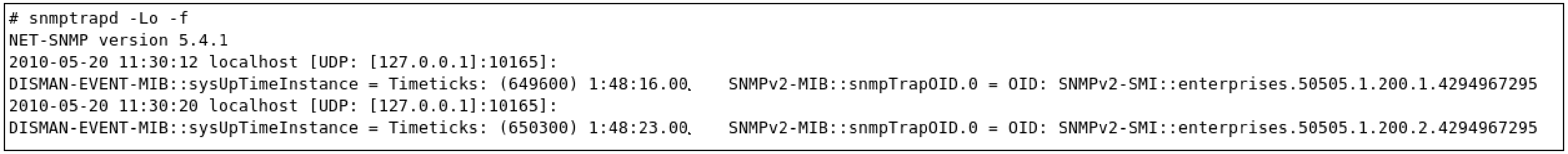
SNMP Agent:

- Event transformation to either SNMP trap or inform.
- Sending trap/inform to all registered trap hosts (for instance, running *Net-SNMP* trap daemon).

Integration with NTF Service (5/5)

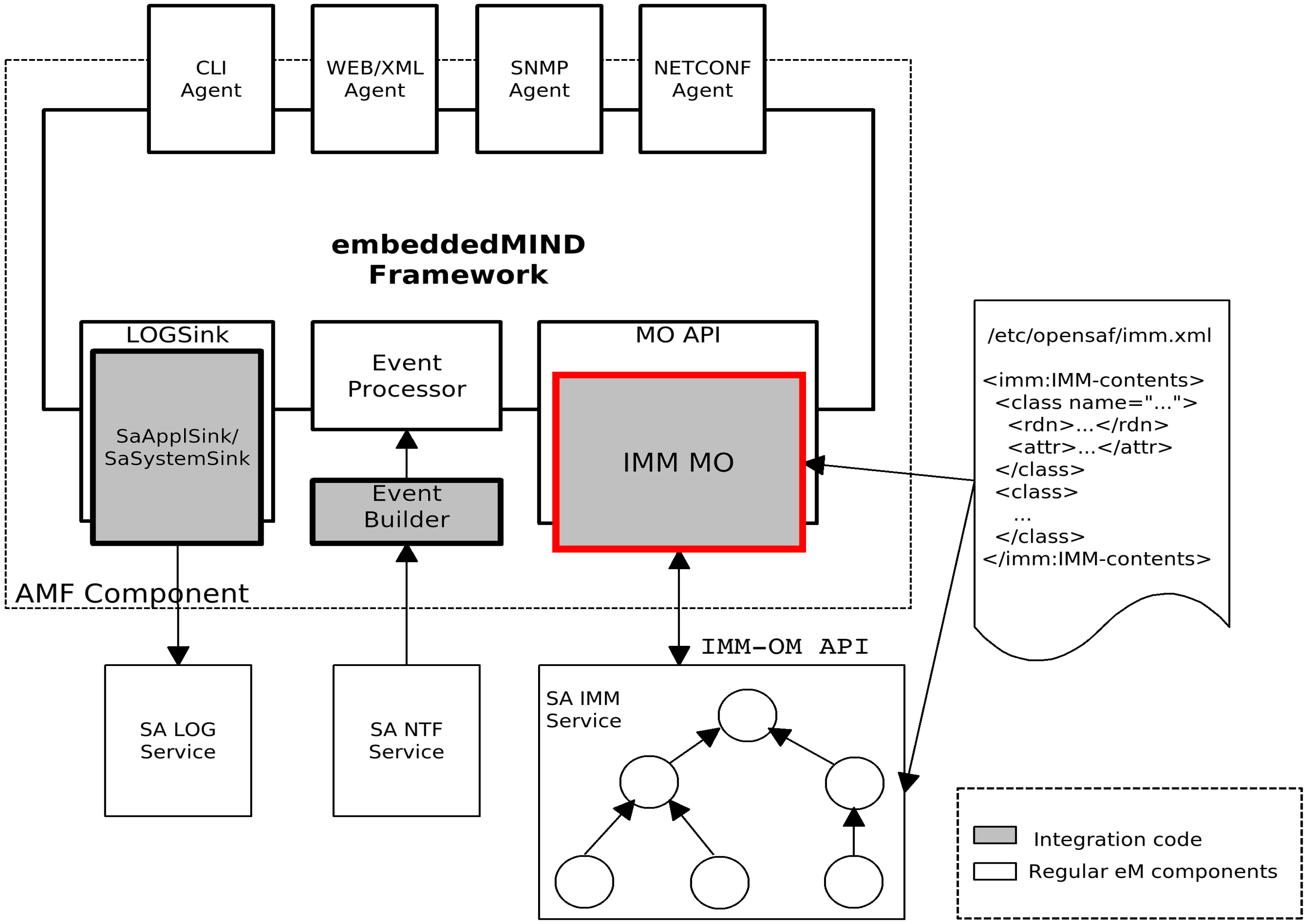
Support for SNMP traps/informs:

- Lack of the SNMP-related information in the NTF service, such as trap OIDs or varbind list.
- Assigning predefined trap OID values to all SA Notification types.
- Event to trap/inform transformation in the SNMP Agent based on the trap's OID.
- Possibility to send SA Notification as the SNMP trap/inform to the SNMP trap host using eM integrated solution.



```
# snmptrapd -Lo -f
NET-SNMP version 5.4.1
2010-05-20 11:30:12 localhost [UDP: [127.0.0.1]:10165]:
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (649600) 1:48:16.00    SNMPv2-MIB::snmpTrapOID.0 = OID: SNMPv2-SMI::enterprises.50505.1.200.1.4294967295
2010-05-20 11:30:20 localhost [UDP: [127.0.0.1]:10165]:
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (650300) 1:48:23.00    SNMPv2-MIB::snmpTrapOID.0 = OID: SNMPv2-SMI::enterprises.50505.1.200.2.4294967295
```

Integration with IMM Service (1/9)



```
/etc/opensaf/imm.xml

<imm:IMM-contents>
  <class name="...">
    <rdn>...</rdn>
    <attr>...</attr>
  </class>
  <class>
    ...
  </class>
</imm:IMM-contents>
```

Integration with IMM Service (2/9)

OpenSAFfire integrated solution as a management application for the IMM service.

Visibility into the configuration and state information for the objects in the IMM model through the standard management protocols:

- CLI
- NETCONF
- WEB/XML
- SNMP

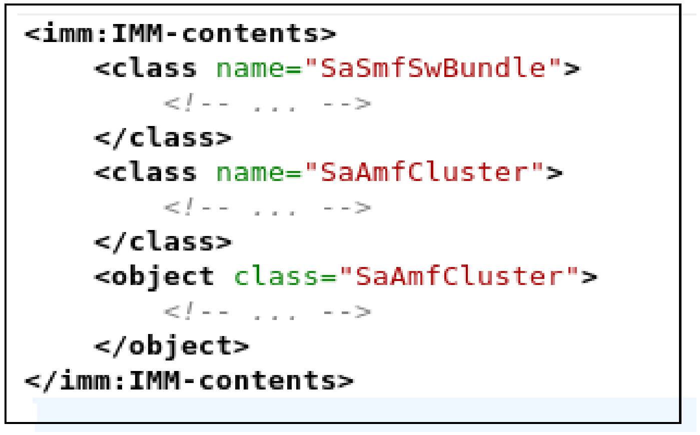
Integration with IMM Service (3/9)

- Translation of the SA IMM model into the eM internal model.
- Implementation of a dedicated IMM MINDObject (*IMMMO*) communicating with the IMM service.
- Loading the IMM model to eM at runtime.
- YANG IMM model.
- Support for SNMP.
- Standalone IMM management application (AMF independent component).

Integration with IMM Service (4/9)

Processing the IMM model:

- IMM model as an input for the eM internal model.
- Creating eM model by reading the cluster's *imm.xml* model file.



```
<imm:IMM-contents>
    <class name="SaSmfSwBundle">
        <!-- ... -->
    </class>
    <class name="SaAmfCluster">
        <!-- ... -->
    </class>
    <object class="SaAmfCluster">
        <!-- ... -->
    </object>
</imm:IMM-contents>
```

- Processing IMM class definition only *(/IMM-contents/class).*

GOAHEAD

Integration with IMM Service (5/9)

Mapping SA IMM types to eM BaseTypes:

SA type	eM BaseType
SaInt32T	Integer
SaUint32T	Unsigned
SaInt64T	Integer64
SaUint64T	Unsigned64
SaTimeT	Time64
SaNameT	String<0, 255>
SaFloatT	Float
SaDoubleT	Double
SaStringT	RawString
SaAnyT	RawData

Mapping an IMM attribute with the *MULTI_VALUE* flag as a sequence of an appropriate BaseType.

Integration with IMM Service (6/9)

IMM MINDObject implementation:

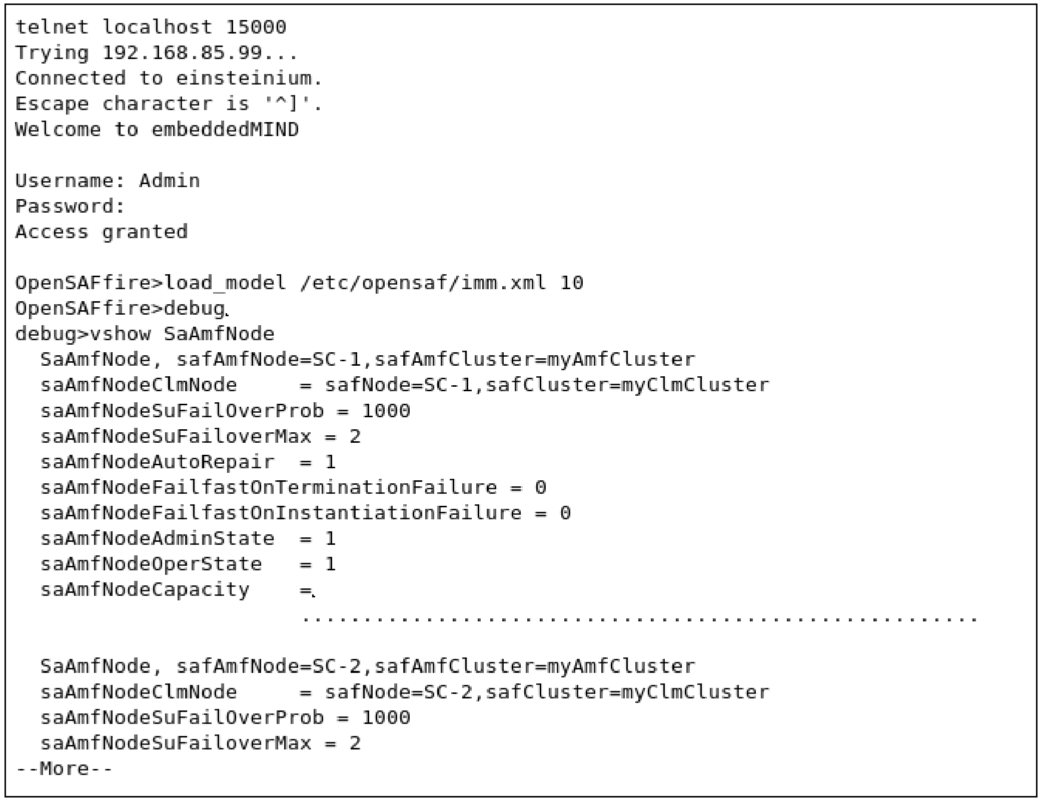
- Basic operations:
  - Getting objects list (instances list) for an IMM class.
  - Getting attribute values of an IMM class' object.
  - Setting attributes of an IMM class' object.
  - Creating/deleting IMM class' objects.
- Operating on the IMM service using SA *IMM-OM* API only.
- A one-to-one mapping between CCB and a MINDObject’s modifying operation.

Note: eM does not caches any data of managed applications.

Integration with IMM Service (7/9)

Loading IMM Model at eM runtime:

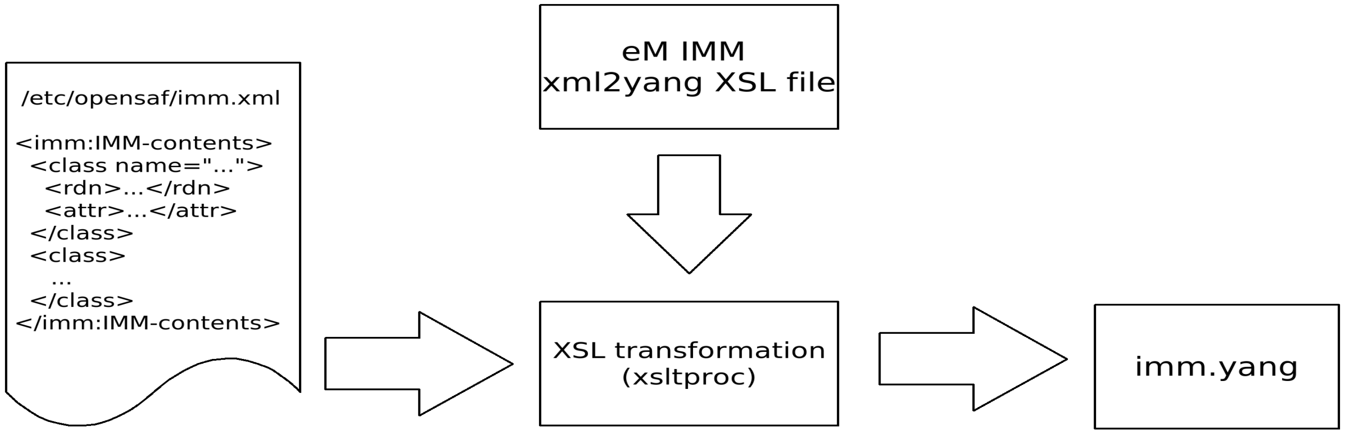
- Auto-loading at eM startup.
- Explicit loading.



```
telnet localhost 15000
Trying 192.168.85.99...
Connected to einsteinium.
Escape character is '^]'.
Welcome to embeddedMIND

Username: Admin
Password:
Access granted

OpenSAFfire>load_model /etc/opensaf/imm.xml 10
OpenSAFfire>debug
debug>vshow SaAmfNode
  SaAmfNode, safAmfNode=SC-1,safAmfCluster=myAmfCluster
  saAmfNodeClmNode      = safNode=SC-1,safCluster=myClmCluster
  saAmfNodeSuFailOverProb = 1000
  saAmfNodeSuFailoverMax = 2
  saAmfNodeAutoRepair   = 1
  saAmfNodeFailfastOnTerminationFailure = 0
  saAmfNodeFailfastOnInstantiationFailure = 0
  saAmfNodeAdminState   = 1
  saAmfNodeOperState    = 1
  saAmfNodeCapacity     =
                        ......................................................

  SaAmfNode, safAmfNode=SC-2,safAmfCluster=myAmfCluster
  saAmfNodeClmNode      = safNode=SC-2,safCluster=myClmCluster
  saAmfNodeSuFailOverProb = 1000
  saAmfNodeSuFailoverMax = 2
--More--
```

Integration with IMM Service (8/9)

Representation of the IMM Model in YANG:



'xsltproc imm_xml2yang.xsl /etc/opensaf/imm.xml > /etc/opensaf/imm.yang'

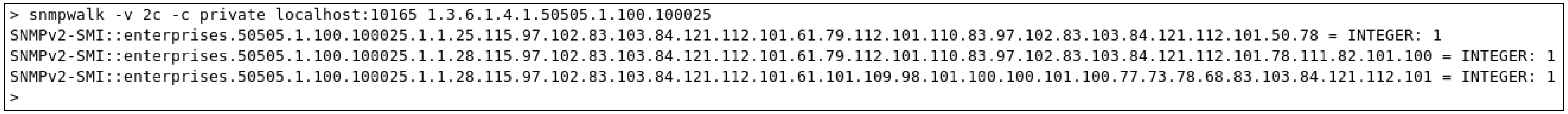
Loading IMM YANG model at eM runtime:

```
OpenSAFfire>load_model /etc/opensaf/imm.yang 10
OpenSAFfire>debug
debug>vshow SaAmfNode
  SaAmfNode, safAmfNode=SC-1,safAmfCluster=myAmfCluster
  saAmfNodeClmNode      = safNode=SC-1,safCluster=myClmCluster
  saAmfNodeSuFailOverProb = 1000
--More--
```

Integration with IMM Service (9/9)

Support for SNMP:

- Lack of the SNMP-related information in the IMM model, i.e. OIDs for the IMM classes and attributes.
- Generating OID values for each IMM class and attribute from the IMM model representation (XML/YANG file).
- Possibility to use an SNMP manager to operate on the IMM service using eM integrated solution.
- Example: *snmpwalk* through the *SaAmfSGBaseType* class:



```
> snmpwalk -v 2c -c private localhost:10165 1.3.6.1.4.1.50505.1.100.100025
SNMPv2-SMI::enterprises.50505.1.100.100025.1.1.25.115.97.102.83.103.84.121.112.101.61.79.112.101.110.83.97.102.83.103.84.121.112.101.50.78 = INTEGER: 1
SNMPv2-SMI::enterprises.50505.1.100.100025.1.1.28.115.97.102.83.103.84.121.112.101.61.79.112.101.110.83.97.102.83.103.84.121.112.101.78.111.82.101.100 = INTEGER: 1
SNMPv2-SMI::enterprises.50505.1.100.100025.1.1.28.115.97.102.83.103.84.121.112.101.61.101.109.98.101.100.100.101.100.77.73.78.68.83.103.84.121.112.101 = INTEGER: 1
>
```

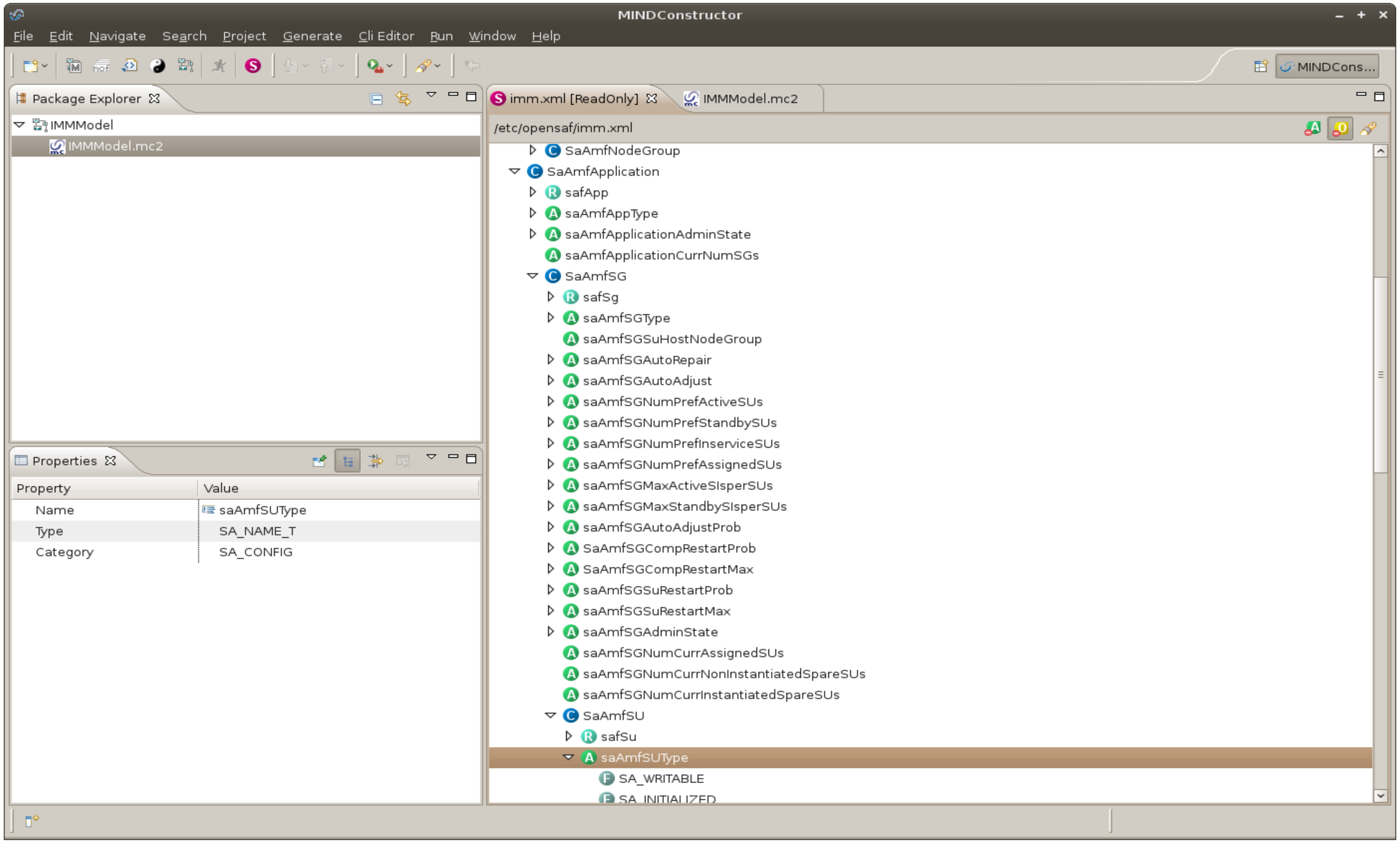
IMM Issues (1/2)

- No definition of the object class hierarchy (in terms of naming, not inheritance).
- Class instance access through IMM OM:
  - No way to directly access instances of a given class
    - Search based on the class name attribute value required.
  - No way to access instances in a user defined order
    - Issue for SNMP walks.
- SNMP MIBs
  - IMM OM interface does not map well into published SA Forum MIBs.
  - Issue for the management layer because either the management layer needs to define custom MIBs or introduce some kind of mapping.

IMM Issues (2/2)

- No definition of the administrative operations, if any, supported by each object class.
- Lack of features:
  - Non-string keys
    - Representing DNs in form of (always) strings makes managing IMM through SNMP very difficult.
  - Multi-value keys.
  - Extensible types (lack of IPAddress, MACAddress, Enumeration, etc.).

MIND*Constructor* GUI IDE tool (1/2)



IMMModel
IMMModel.mc2

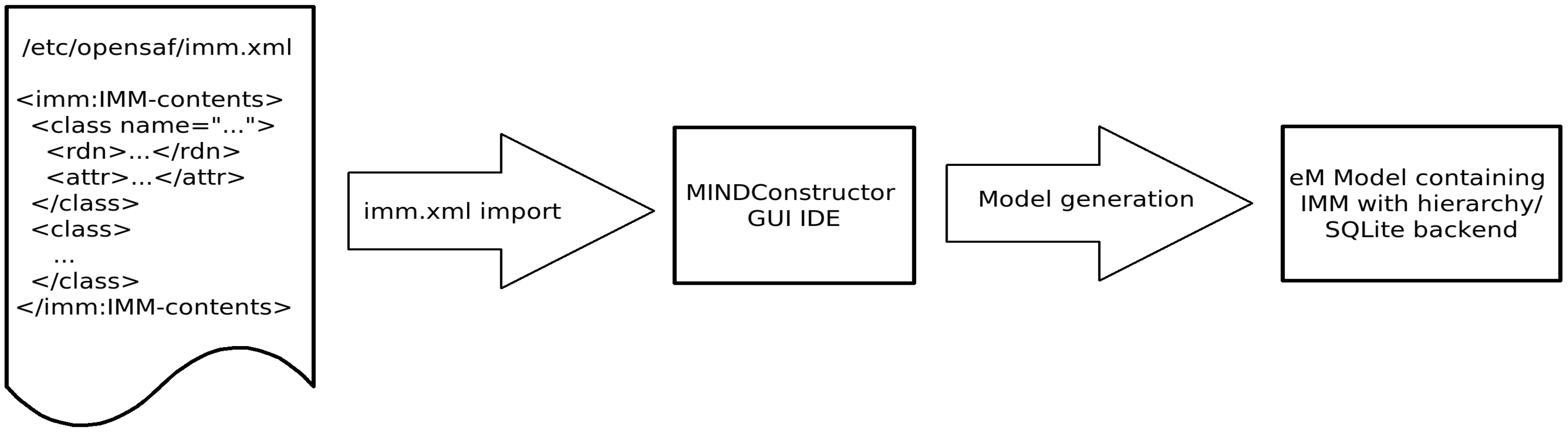
Property	Value
Name	saAmfSUType
Type	SA_NAME_T
Category	SA_CONFIG

/etc/opensaf/imm.xml

```
  ▷ C SaAmfNodeGroup
▽ C SaAmfApplication
  ▷ R safApp
  ▷ A saAmfAppType
  ▷ A saAmfApplicationAdminState
    A saAmfApplicationCurrNumSGs
  ▽ C SaAmfSG
    ▷ R safSg
    ▷ A saAmfSGType
      A saAmfSGSuHostNodeGroup
    ▷ A saAmfSGAutoRepair
    ▷ A saAmfSGAutoAdjust
    ▷ A saAmfSGNumPrefActiveSUs
    ▷ A saAmfSGNumPrefStandbySUs
    ▷ A saAmfSGNumPrefInserviceSUs
    ▷ A saAmfSGNumPrefAssignedSUs
    ▷ A saAmfSGMaxActiveSIsperSUs
    ▷ A saAmfSGMaxStandbySIsperSUs
    ▷ A saAmfSGAutoAdjustProb
    ▷ A SaAmfSGCompRestartProb
    ▷ A SaAmfSGCompRestartMax
    ▷ A saAmfSGSuRestartProb
    ▷ A saAmfSGSuRestartMax
    ▷ A saAmfSGAdminState
      A saAmfSGNumCurrAssignedSUs
      A saAmfSGNumCurrNonInstantiatedSpareSUs
      A saAmfSGNumCurrInstantiatedSpareSUs
    ▽ C SaAmfSU
      ▷ R safSu
      ▽ A saAmfSUType
          F SA_WRITABLE
          F SA_INITIALIZED
```

MIND*Constructor* GUI IDE tool (2/2)

Generating hierarchical IMM model using MIND*Constructor*:



Q&A

GOAHEAD

Thank you

Tomasz Mikolajczyk
tmikolajczyk@goahead.com

embeddedMIND
embeddedMIND@goahead.com
http://www.goahead.com