

# NIMI and Network Monitor Integration

## ***Assumptions and Goals***

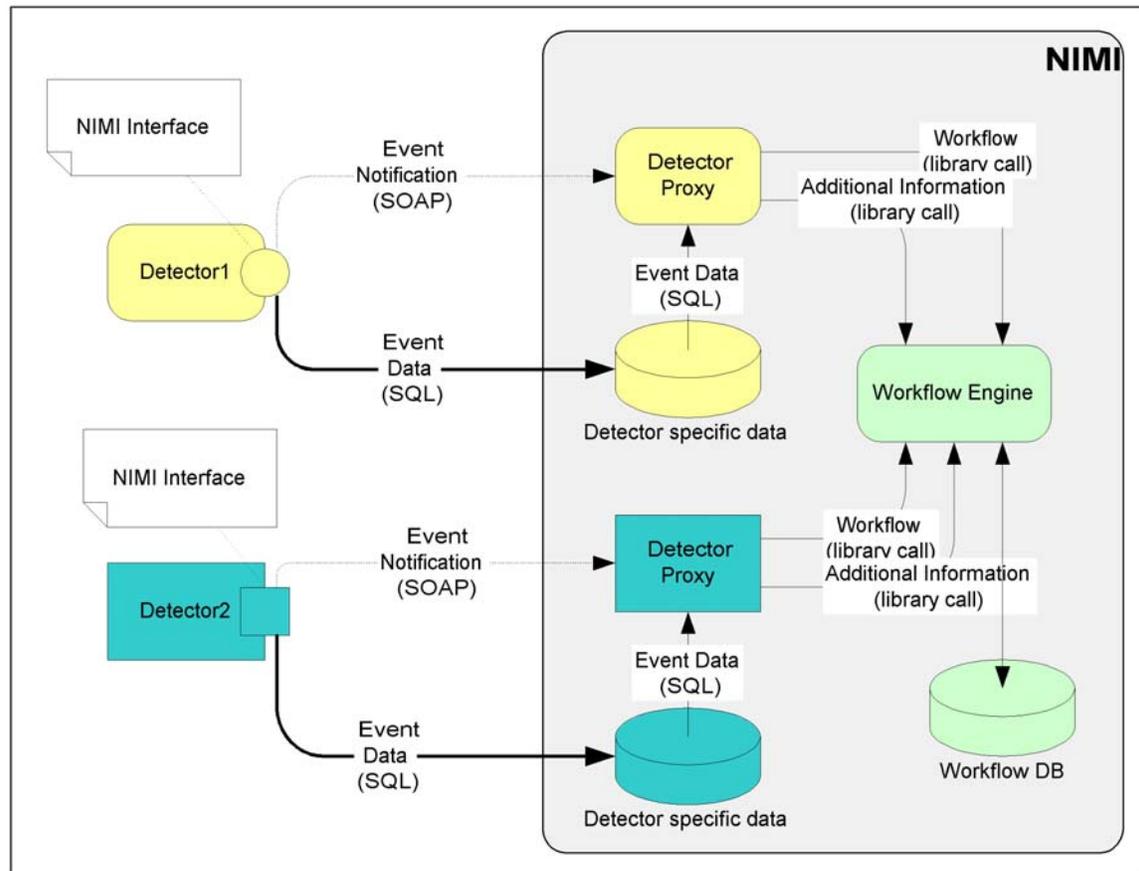
There are several independent network monitors (*detectors*) on site. They are monitoring network activity and are capable of detecting one or more *conditions*. Every time a condition is detected, the detector enters a *detector event* into NIMI database in detector-specific format.

NIMI operates its own workflow management system (*workflow engine*) where some or all conditions detected by the detectors have to be tracked as *issues* or *events*. Events are represented in the Workflow Engine as *workflow elements* in detector-independent way. However, workflow element has a reference to detector-specific bundle of data associated with corresponding detector event. Workflow behavior is determined by detector-independent portion of workflow element, but detector-specific information is needed for complete description of the event for the administrator or user.

The goal is to provide common architecture solution which would be used to integrate existing and future network monitors in such a way that:

- NIMI and the Detector maintain as much independence as possible in the sense that future modifications of one has as little effect on the other as possible;
- Data stored by the detector is usable not only by the workflow engine, but also by other NIMI-based applications and by the Detector itself if necessary;
- The architecture is efficient in terms of overall system performance and response time;

## Architecture Solution



Integration of a detector with NIMI is performed through 2 major components, NIMI Interface and Detector Proxy.

### NIMI Interface

NIMI Interface is a piece of software that reads detector event data produced by the detector and

- Stores the data into NIMI database using (secured) SQL access and
- Generates event notification sent to Detector Proxy as a (secured) SOAP message

### Detector Proxy

Detector Proxy consists of 2 parts, SOAP Server and Detector Proxy Library.

SOAP Server receives notifications from the detector. The notification carries only a reference to the event previously stored into the database by the NIMI Interface. Upon the receipt of the notification, the SOAP Server reads event data from the database, compiles it and calls Workflow Engine interface library to create new workflow element.

Also, SOAP Server can be used for subsequent notifications about future changes of the detected condition.

Detector Proxy Library is used by the Workflow Engine to obtain detector-specific event data associated with the workflow element. This is done later during the lifetime of the workflow element, for example to generate event details portion of e-mail message sent to the user.