LHC Alarm Service (LASER)

Mark Buttner, 25 May 2009

Summary

- Scope
- Architecture & Technical requirements
- Key concepts
- Sending alarms
- Recommendations
- Demo

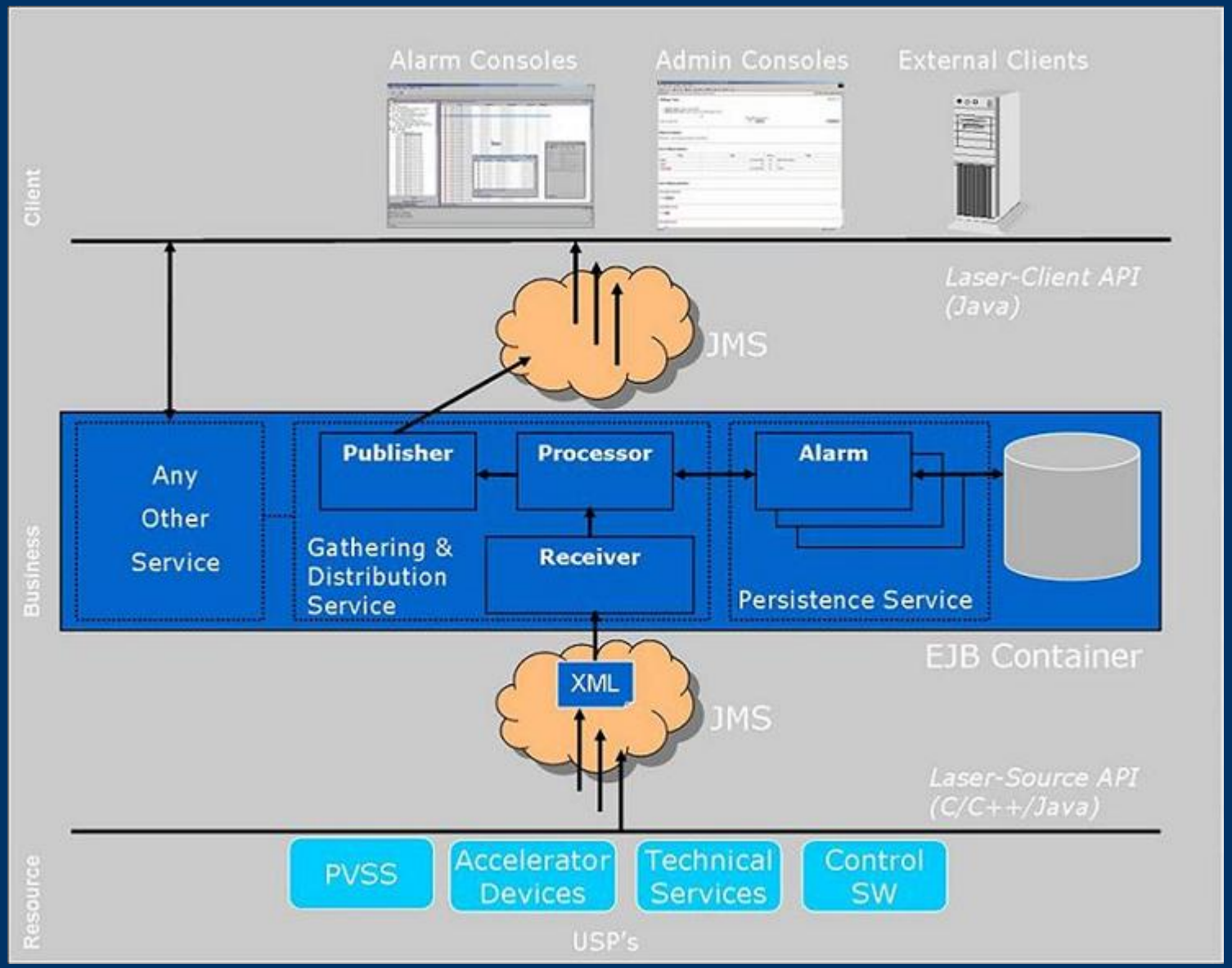
Scope

- The Alarm Team delivers an alarm service for the operation of the accelerator chain and technical services. This includes alarms from the PS complex, SPS, LHC, the CERN technical services (provided by TS), and surveillance (provided by IT) of networks or hosts supporting the above.
- The current implementation is from the LASER project (LHC Alarm SERvice Project). The LASER project addresses the analysis, design and procurement of the alarms management service for the future LHC control system

Summary

- Scope
- Architecture & Technical requirements
- Key concepts
- Sending alarms
- Recommendations
- Demo

Architecture



Technical requirements

- Source
  - Java API
  - C/C++ API for Linux and LynxOS 4
- Middle tiers
  - Oracle database
  - Sonic MQ JMS broker
  - OC4J container
- Client (GUI)
  - Any system running Java/Swing

Statistics

- Actual load
  - ~200'000 alarm definitions in database
  - 150-250 alarm messages/minute (in average)
  - >10'000 alarm messages/minute (peeks)
  - Min. 10 consoles open
- Scalability (tested with DIAMON)
  - 1000 messages / minute
  - 60 consoles open
  - ==> no problems observed

Summary

- Scope
- Architecture & Technical requirements
- Key concepts
- Sending alarms
- Recommendations
- Demo

Key concepts

- Alarm definition
  - Static (in database):
    - System name, identifier, problem description
    - Priority
    - Cause, Action, Consequence
    - Responsible person
    - Notification
    - Categories
  - Dynamic (sent by source):
    - Active or terminated
    - Timestamps
    - User properties

Key concepts

- Alarm source
  - A programme written by information providers
  - Based on APIs provided by LASER
  - Sends:
    - Events (alarm activation, termination)
    - List of active alarms for synchronization

Key concepts

- Alarm processing
  - Filtering incoming messages:
    - Source and alarm definition known
    - Oscillation control
  - Surveillance of alarm sources
  - Reduction, based:
    - Source requests (node reduction)
    - Multiplicity (number of similar alarms)
  - Distribution to categories
  - Archiving

Key concepts

- User actions (Console)
  - Configuration
    - Subscription to categories, filters
    - Dynamic grouping
  - Working with alarms
    - Acknowledge
    - Mask
    - Inhibit
    - Search
    - ...

Summary

- Scope
- Architecture & Technical requirements
- Key concepts
- Sending alarms
- Recommendations
- Demo

Sending alarms: the API

- Java example with the “laser-source-ext” API:
  - ...
  - AlarmSourceManager asm = AlarmSourceFactory.getSourceManager();
  - ...
  - asm.setSourceName(“MY_SOURCE_ID"));
  - ...
  - AlarmInstance alarm =
    builder.clear().setProblemKey(“ALARM_DEF_KEY”).getInstance();
  -
  - manager.activate(alarm);
  - ...
  - manager.terminate(alarm);
  - ...

Sending alarms: Existing sources

- CMW devices (through AlarmMon)
- Computer alarms (input from Diamon)
- Network alarms (input from Spectrum)
- ...

Summary

- Scope
- Architecture & Technical requirements
- Key concepts
- Sending alarms
- Recommendations
- Demo

Recommendations

- Use alarm system only for alarms
  - Limited number per time/operator
  - An alarm requires and action in response
  - An alarm should have a short lifecycle
- Define processes
  - How alarms are defined
  - Who does what when an alarm becomes activate
- Monitoring of the system

Demo