



# Lemon Tutorial

## Lemon Overview

Miroslav Siket, Dennis Waldron

<http://cern.ch/lemon>

CERN-IT/FIO-FD

---



# Tutorial

---

- Why?
  - Number of services is expanding. More to monitor every day.
- For whom?
  - Service managers to configure monitoring of their services
  - Developers to simplify their life when writing sensors
  - Site managers to setup their monitoring instances



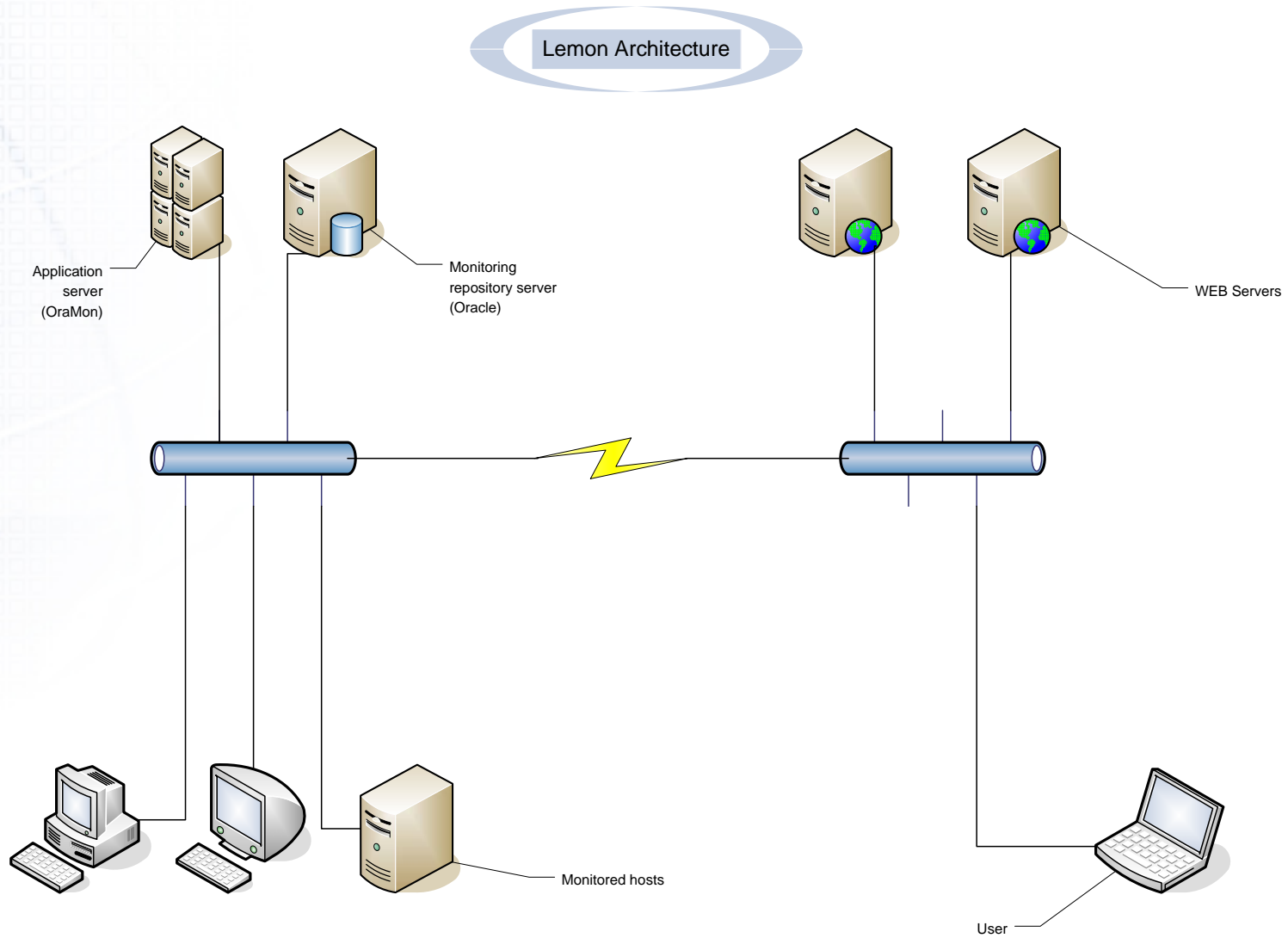
# Tutorial Outline

---

- Architecture
- Writing sensors
- Running and configuring Agent
- Using lemon tools
- Running Lemon server(s)
- Running and configuring web interface
- Running alarm system



# Architecture





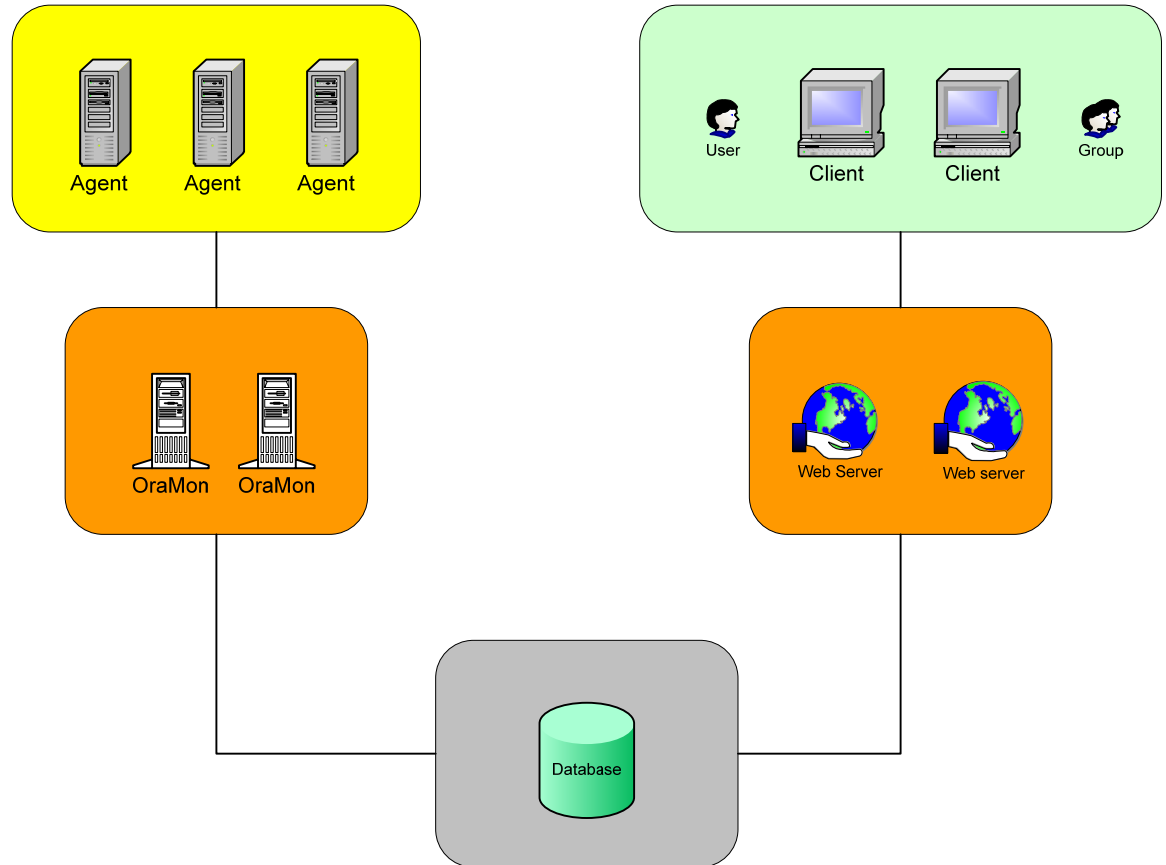
# Architecture II

Three layers:

Data producing/consuming

Data manipulation

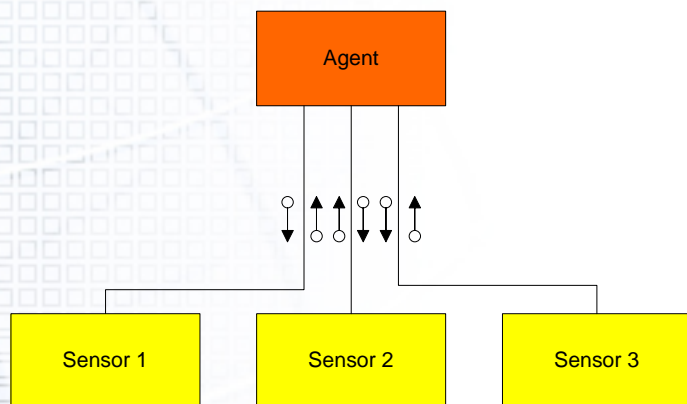
Data Storage





# Client side

## Agent



- forks sensors and communicate with them using custom protocol over a bi-directional “pipes”
- configures **metric instances** of **metric classes** of a **sensor** and pulls for **metrics**
- checks on status of sensors
- agent sends data to servers using TCP or UDP
- monitors itself with internal MSA sensor
- caches data locally

Default Linux client distribution comes with the agent, linux and file sensors.

Footprint: agent - 5.5MB and 0.02% of CPU utilization\*

core sensors (Linux, file, exception) – 10MB, 0.2% of CPU\*

parseLog – 9.4MB

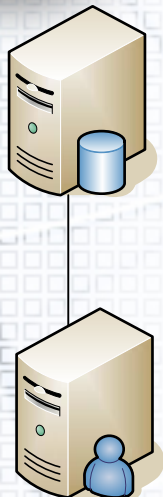
Currently C++ and perl APIs available.

\* i386, SLC3/4, RHES3/4 – average over CERN CC



# Server side

Two implementations:



- **Oracle based – OraMon**
  - optimized for high performance and for large Computer Centers
  - runs on Oracle 9i+ (with alarms system on 10g)
  - validation of metric samples, metadata information
- **Flat files based – FlatMon (edg-fmon-server)**
  - uses OS files for storing data
  - for smaller sites (scalable to 1000 machines max.)

General features:

- multithreaded UDP/TCP server
- built in authentication mechanism

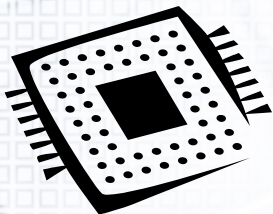


# Server side - planning



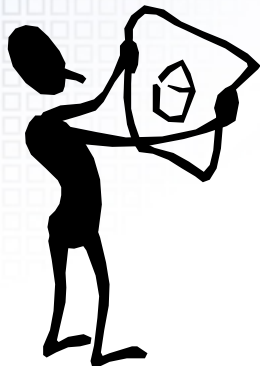
## Space considerations

- About 400kB of data per machine/day (Oracle Enterprise edition with compression) – 700kB without compression (XE, Standard)
- About 1.2MB for FlatMon per machine per day



## CPU considerations

- Dual PIV, 3GHz, 4GB of memory with Oracle DB server + OraMon requires about 15% CPU for 4000 monitored machines
- Adding Alarm system on Oracle requires additional 5% of CPU
- FlatMon saturates the above machine with 1000 monitored hosts
- OraMon/FlatMon require about 105MB of memory



## Functionality considerations

- FlatMon does not provide metric checks and has no metadata concept
- Lemon Alarm System (LAS) runs on Oracle as PL/SQL procedures and requires Oracle 10g – integrated with OraMon schema in Oracle database
- For HA architecture, use Oracle RAC and multiple OraMon servers

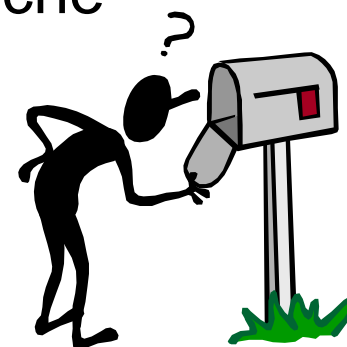




# User/administration tools

## Lemon-cli

- Retrieving monitoring data from the local machine cache
- Allows retrieving data from the server
- Currently uses SOAP interface (to be retired soon)



## Lemon-host-check

- Checks status of the machine based on the values of exceptions
- Checks status of the monitoring agent and sensors
- Manages status of exceptions





# Configuration management

---

## At CERN we use Quattor Configuration Database

- Configuration is stored in hierarchical templates per domain/cluster/node
- NCM framework is used to download configuration XML profile to nodes
- NCM components are used:
  - For agent/sensors configuration – using [fmonagent](#) component
  - For server configuration (metadata) – using [oramonsver](#) component

## For smaller sites with homogeneous structures

- Use default agent and sensor rpms from Lemon
- Use rpms for custom sensors/settings



# Lemon RRD framework

- User front-end for visualization and caching monitoring data
- Two layers
  - Pre-processing – consumes monitoring data and creates rrd files per machine/cluster/... (aging, averages) - [lemonmrd](#)
  - Visualization – using rrd files for fast visualization or direct access to the monitoring repository – [status web pages](#)
- Different plugins/options available:
  - Synoptic display of the Computer Center (XML driven)
  - Lemon Alarm GUI
  - Quattor .tpl file browser, ...

## Requirements

- Web server with PHP (v5+ if want to use LAS)
- rrdtool rpm
- 500kB space per machine's rrd file



# Automatic recovery actions and alarms

- **Sensor exception**
  - For defined values of measured metrics an actuator is called with predefined action
  - An example: ssh daemon dead – action `/sbin/service sshd start`
  - Definition: metric X, field Y `<op>` reference value Z => call actuator
    - `<op>` can be `==, <, >, regexp, range, +,-,*,/` etc..
  - Each occurrence is logged in the Monitoring Repository
  - Already about 230 predefined exceptions with automatic recovery actions
  - Exceptions are base for alarms in Lemon Alarm System
  - Allow multi-valued metrics and on-behalf metrics
  - Allow corrective actions (actuators) up to n-times or within given time window
  - Allow distinguishing of the alarm state (failed actuator, silenced,...)
  - Example:
    - `(10004:7 > 100 && (10005:3 - 34:5)>100:56)`
    - On behalf: `(soap_srvx:302:1 > 10)`



# Lemon Alarm System

Newest addition to Lemon

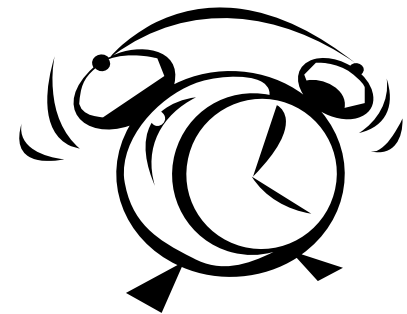
Build on top of the OraMon schema in Oracle database

Comes in two pieces:

- **PL/SQL** stored procedures (requires Oracle 10g) to consume exceptions and to produce alarms
- **GUI** – web based interface based on AJAX – part of LRF

## Features

- Reduction of alarms (by type or by node/cluster)
- Possibility to hide/inhibit alarms
- Access control
- History tracking
- Future: notifications, RSS feeds





# Software distribution

## RPM

- direct download from <http://lemon.web.cern.ch/lemon/downloads.shtml> or at <http://linuxsoft.cern.ch/lemon/>
- YUM setup with [/etc/yum.repos.d/lemon.repo](#)

```
[lemon]
name=Lemon
baseurl=http://linuxsoft.cern.ch/lemon/linux/RPMS/i386/sl4/stable/
enabled=1
gpgcheck=1
gpgkey=http://linuxsoft/lemon/RPM-GPG-KEY-lemon
```
- APT setup with [/etc/apt/sources.list.d/lemon.list](#)

```
# Lemon stable
rpm http://linuxsoft.cern.ch/lemon linux/RPMS/i386/sl4 lemon_stable_sl4
```

## Source code

- CVS

```
CVSROOT=:pserver:anonymous@isscvms.cern.ch:/local/reps/elfms
```



# Future and additional information

---

## Things not covered/under development

- XML gateway with API to several languages (C++, perl, python, java,...)
- Python Sensor API
- LAS notification, RSS feeds
- Encryption of data between agent and server
- Authentication for user access
- Service views for LRF

Check Web pages: <http://cern.ch/lemon> for additional information