

# *Plans for LASER & DIAMON*

6 May 2010

Mark Buttner / Marek Misiowiec  
On behalf of LASER/DIAMON team

# Overview

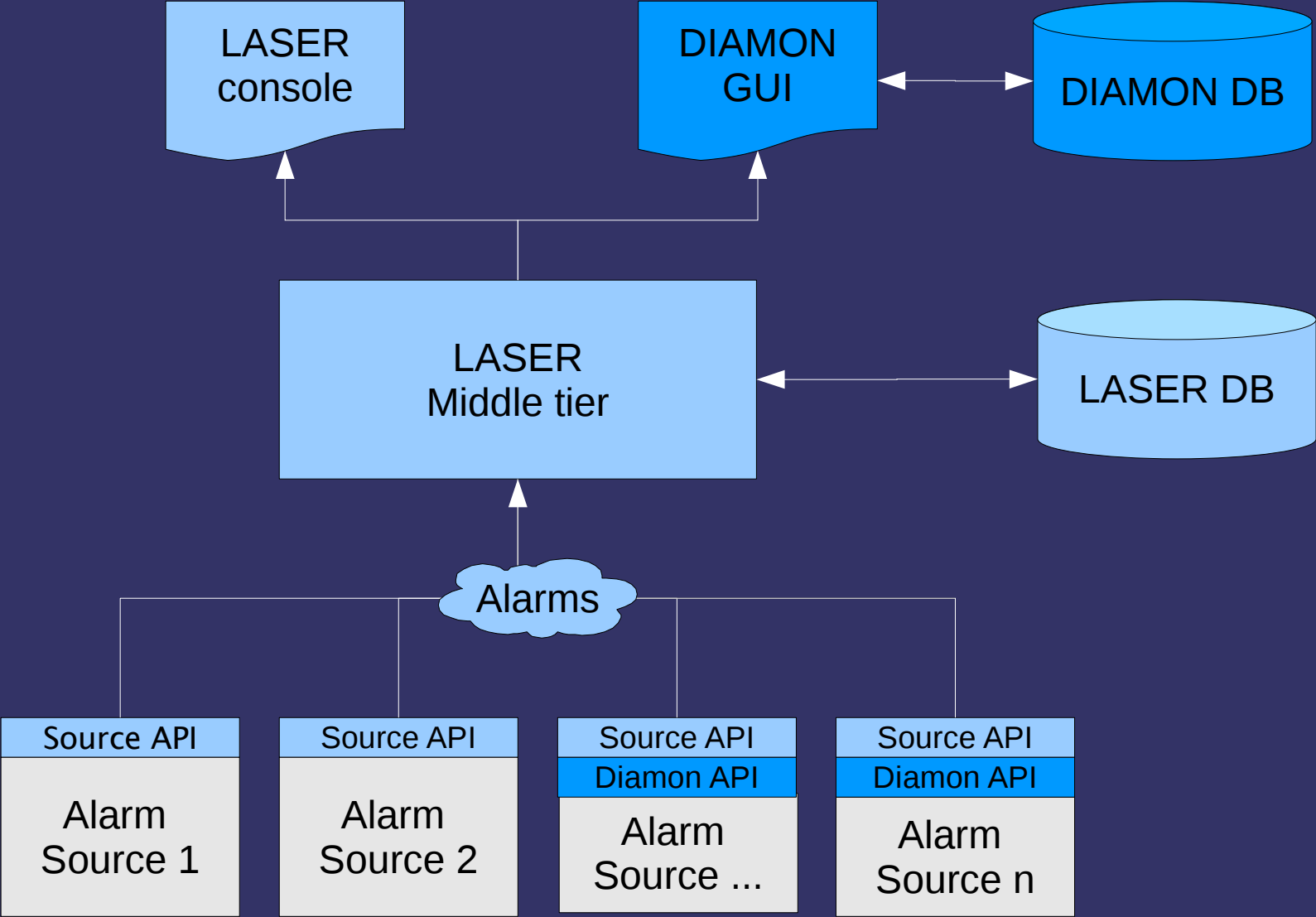
- ➔ Part I: Reasons for a major change
- ➔ Part II: Proposed design
- ➔ Part III: Options for implementation

# Overview Part I

- ⇒ Reasons for a major update
  - Actual LASER/DIAMON design
  - Data handling issue
  - Technology
  - Missing functionality
- ⇒ Scope of the project

# *Actual LASER/DIAMON design*

# Actual LASER / DIAMON design



# *LASER / DIAMON design issues*

- ➔ Middle-tier acts as target for sources
  - Third party sources decide on volume
  - No simple way to disconnect
- ➔ Data issue: Only status information is sent
  - Status changes decided by the source program
  - Communication only when problem is detected (LASER)
  - Justification comes optionally in user properties
  - No influence on source behavior (threshold changes ...)
- ➔ Monitoring based on alarm system, should be alarms based on monitored data

# *LASER / DIAMON design issues* (cont.)

## ➔ Consequences

- Middle-tier can only do technical filtering
- Unjustified alarm flood can overload the system (mainly for LASER)
- Business rules for status changes vary from source to source
- Monitored data only available on request

# *Data handling issue*



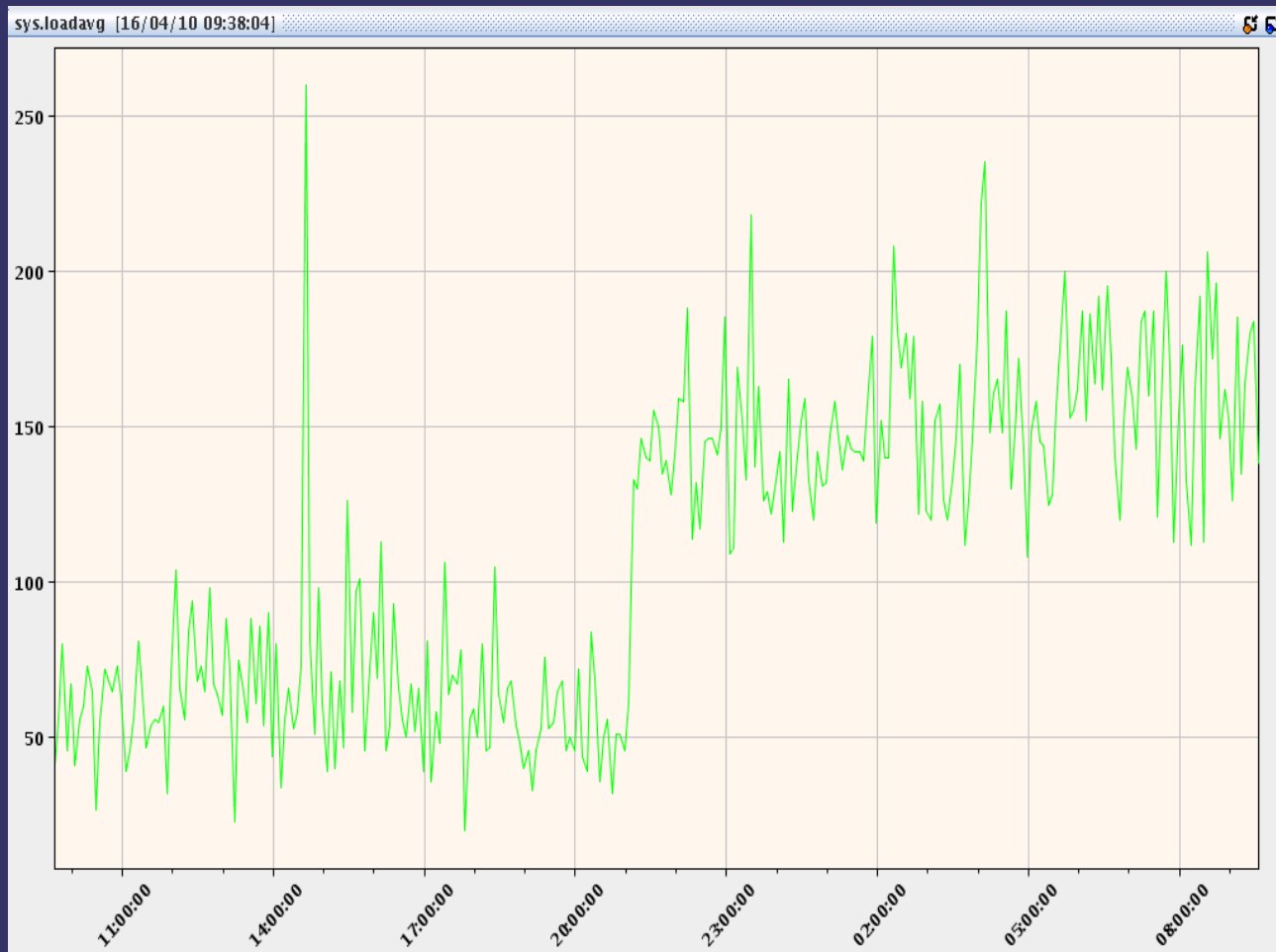
# *Data handling issue*

- ➔ Problem: Data processing is done in user source programs
- ➔ ... it should be done by the LASER / DIAMON middle-tier.

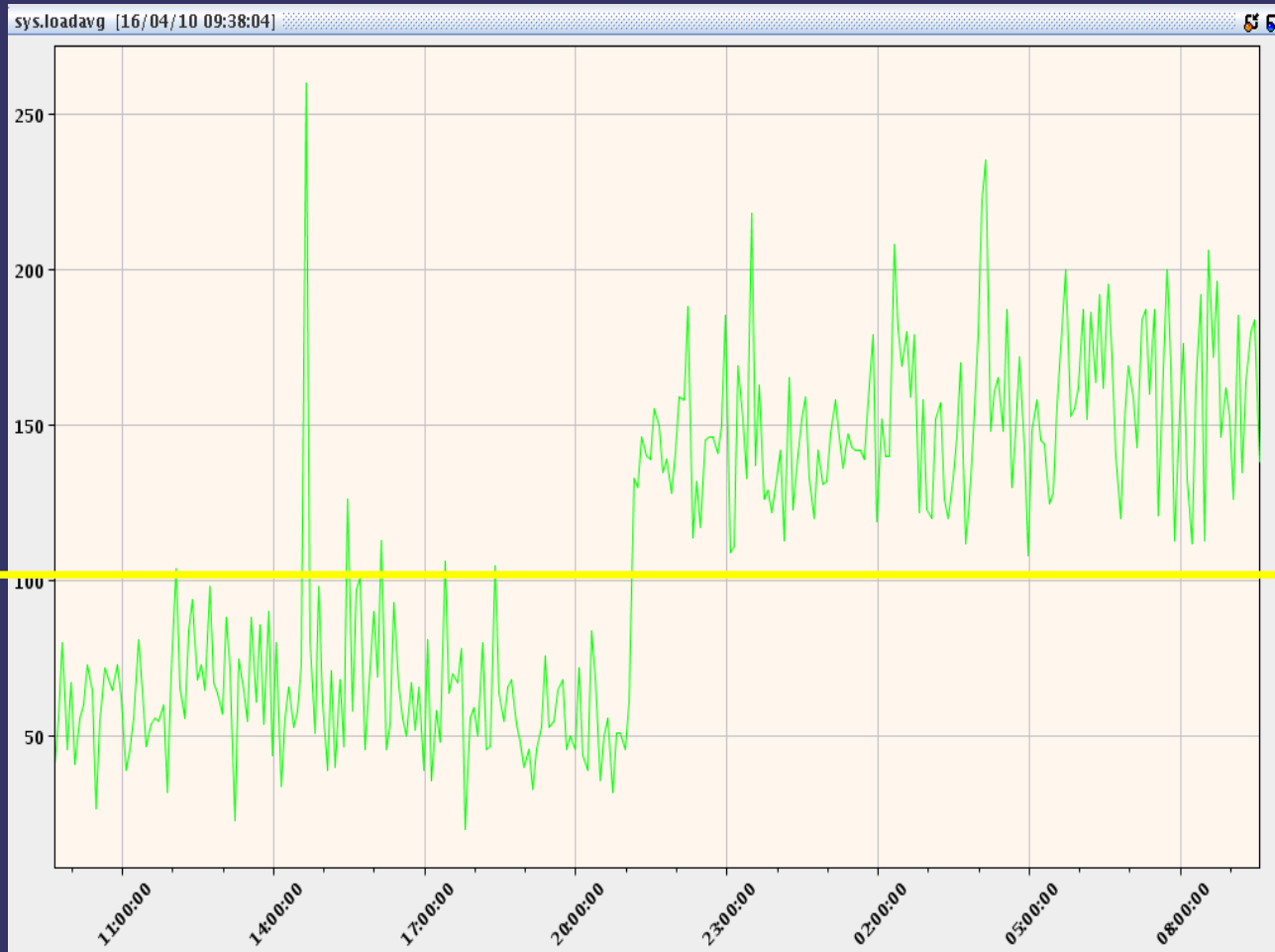
# Examples based on real data

The screenshot displays the Diamon console interface. The title bar reads "Diamon console 1.9.15 - MBUTTNER - file:/afs/cern.ch/user/m/mbuttner/diamon/newconfig.xml". The interface includes a menu bar (File, Edit, View), a search bar, and a "logged as mbuttner" indicator. A tree view on the left shows a hierarchy of services: Root, Central services, CMW, DIAMON, cs-ccr-diam1, CLIC cs-ccr-diam1, DMN D, JMS jm, PING c, cs-ccr-diam3, CLIC cs-ccr-diam3, BOT CMW central ag, PING cs-ccr-diam3, JMS, LASER, LOGGING, OASIS, TIM, AD (ADE), CNGS, PS (CPS), CTF, and CRYO. A context menu is open over "CLIC cs-ccr-diam1", listing options: Error(s) (0), Warning(s) (0), Actions(s), Redo & refresh, Documentation, History (with a sub-menu), Notification(s), Show report, and Show fields. The History sub-menu is open, showing time intervals: Last 1 hour(s), Last 2 hour(s), Last 3 hour(s), Last 6 hour(s), Last 12 hour(s), Last 24 hour(s), and Last 48 hour(s). A tooltip for "Last 24 hour(s)" reads "View the history of this item for the last 24 hour(s)". The main panel shows a list of services: CMW, DIAMON, JMS, LASER, LOGGING, OASIS, and TIM. The "CLIC cs-ccr-diam1" service is selected and highlighted in green. Below the list, the "CLIC cs-ccr-diam1" details are shown, including tabs for General, Details, and Host. The "General" tab is active, displaying buttons for "Repair", "Restart CLIC", and "Reload CFG", along with a "Menu" button and a refresh icon. The status bar indicates "Last message received: Fri Apr 16 08:24:31 CEST 2010" and "No problems found". The bottom status bar shows "08:24:27 - performing automatic login..."

# Example 1: Simple case



# Example 1: Threshold

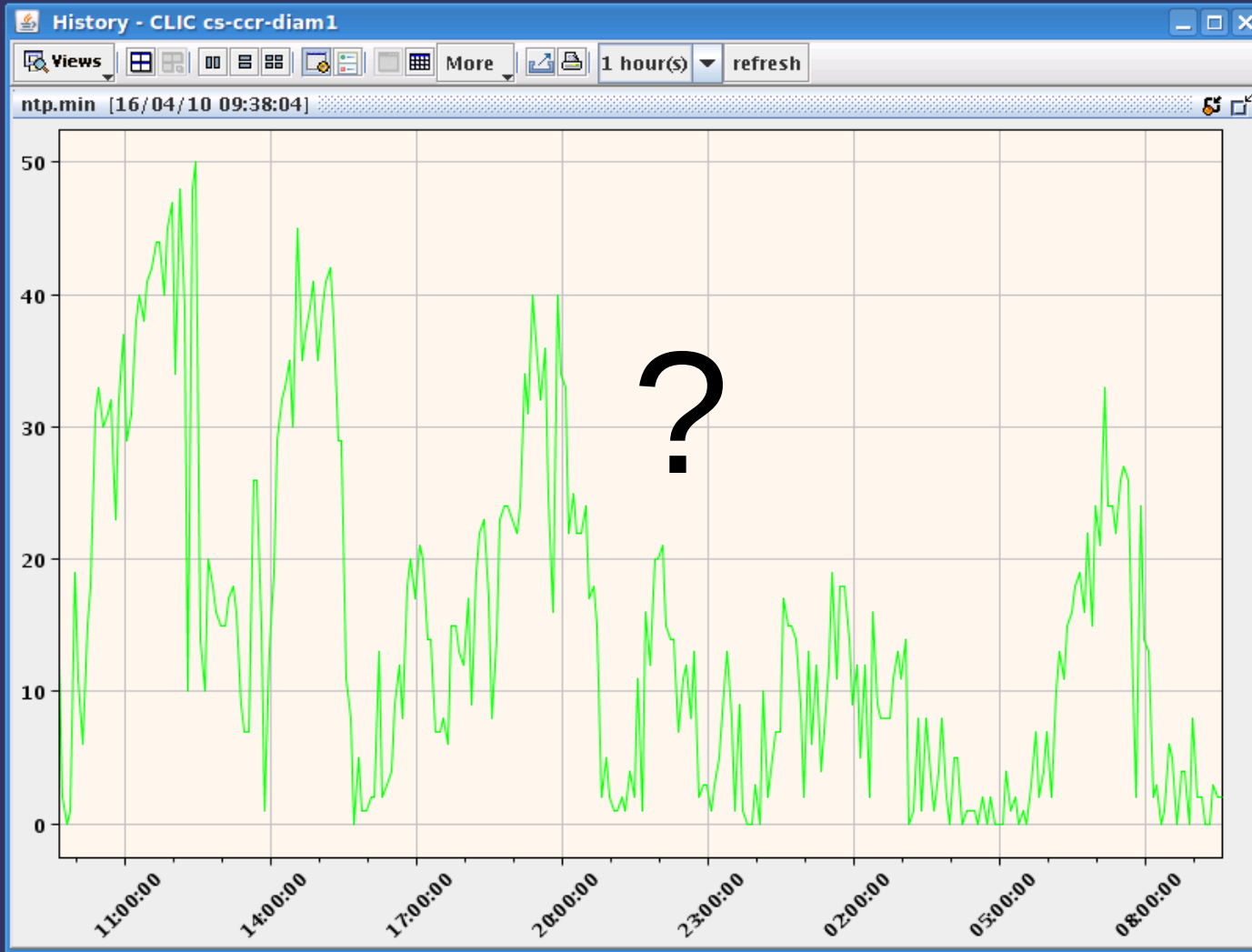


Threshold

Alarm on

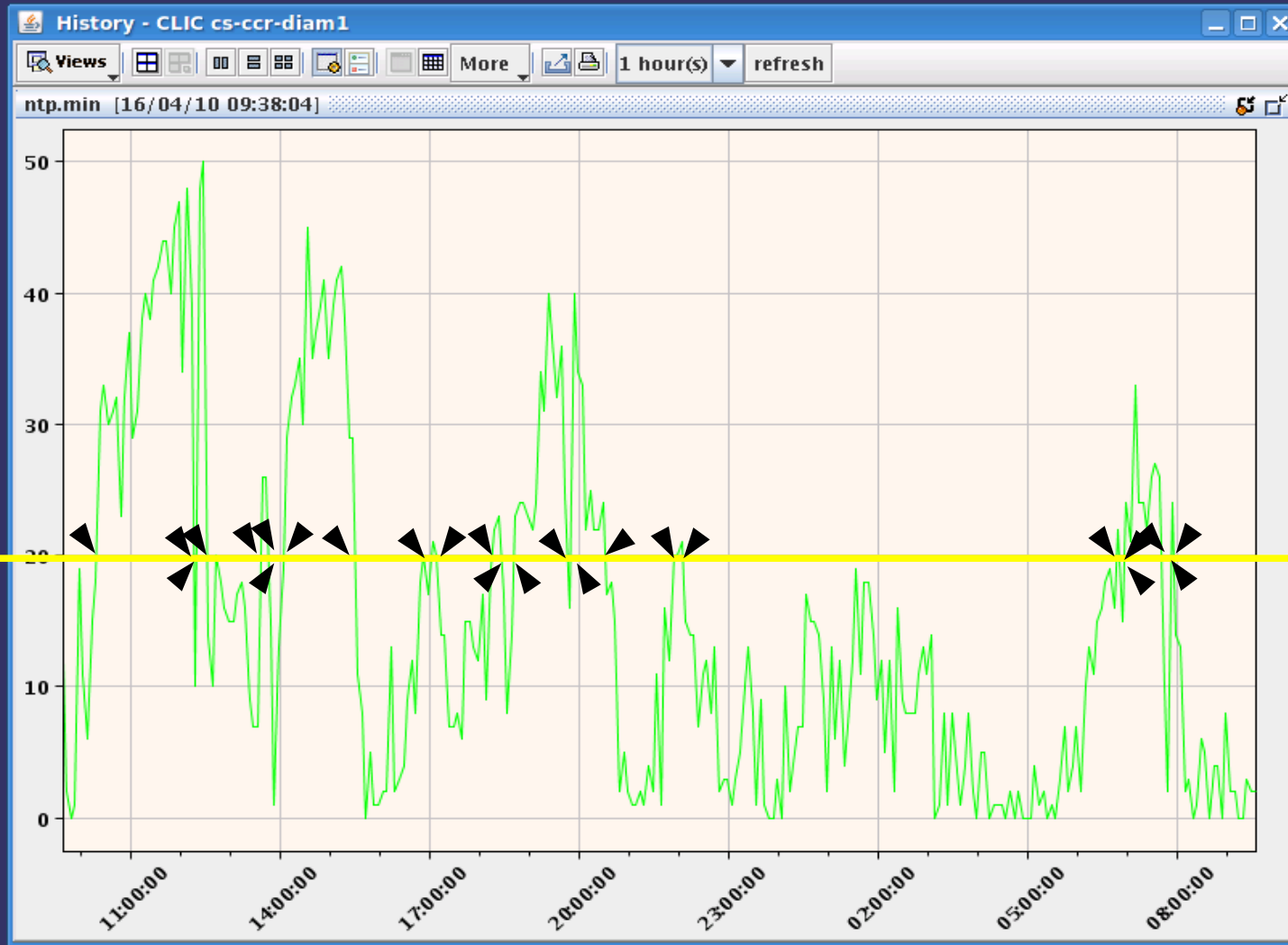
Alarm off

# Example 2: More difficult



# Example 2: Single threshold

(result: 25 state changes)



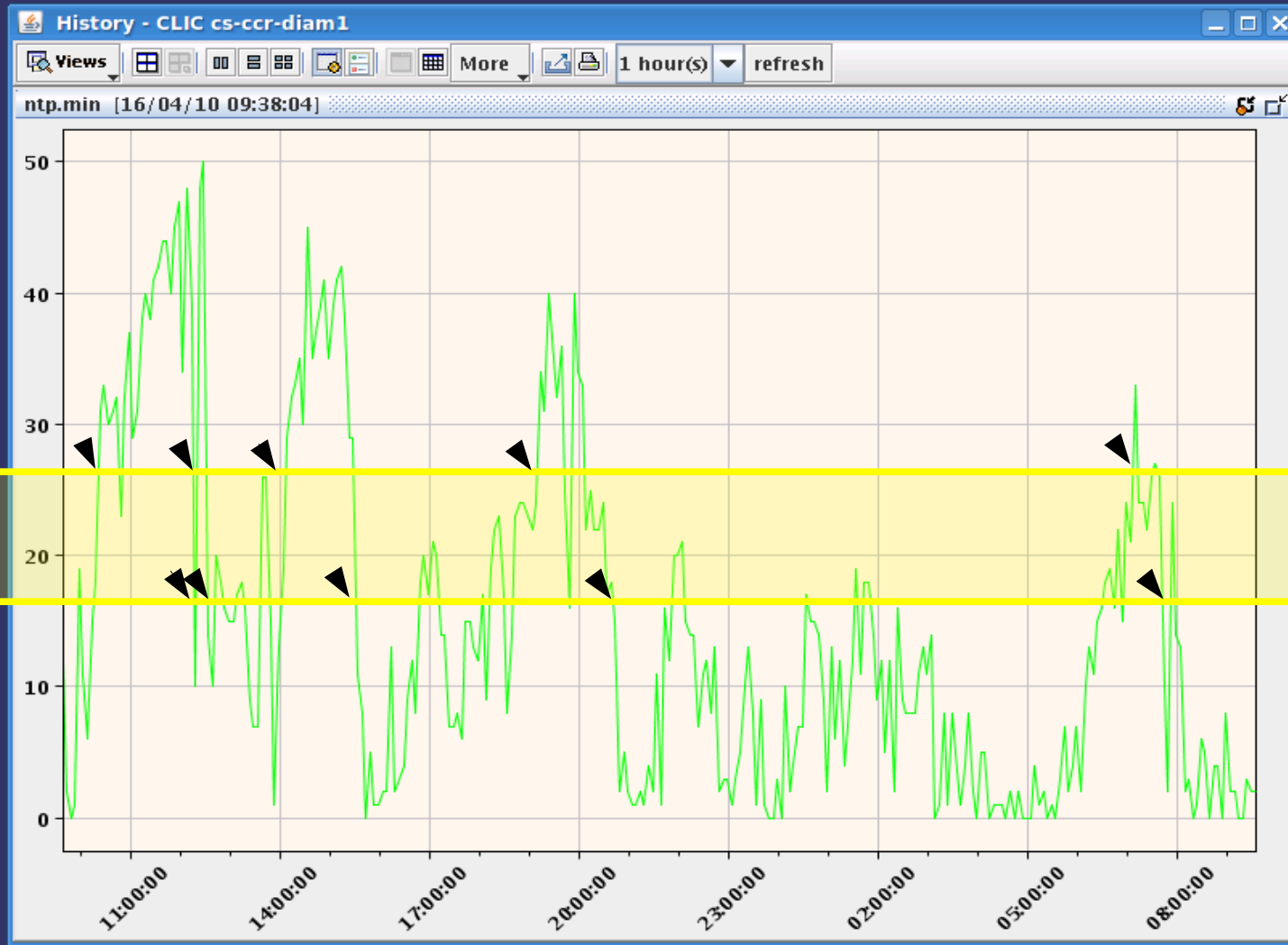
Alarm on

Alarm off

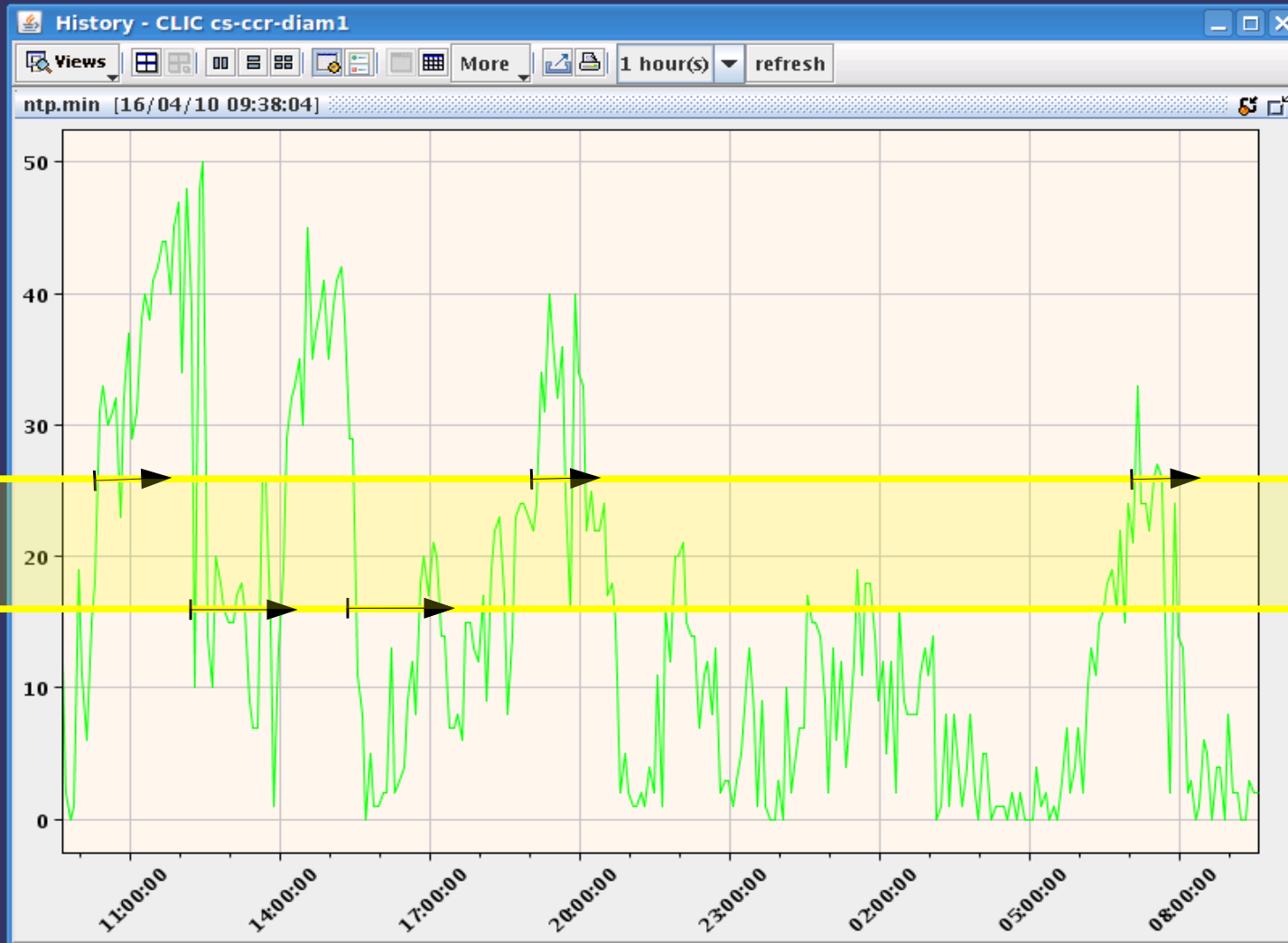
Threshold

# Example 2: Using a dead-band

(result: 10 state changes)



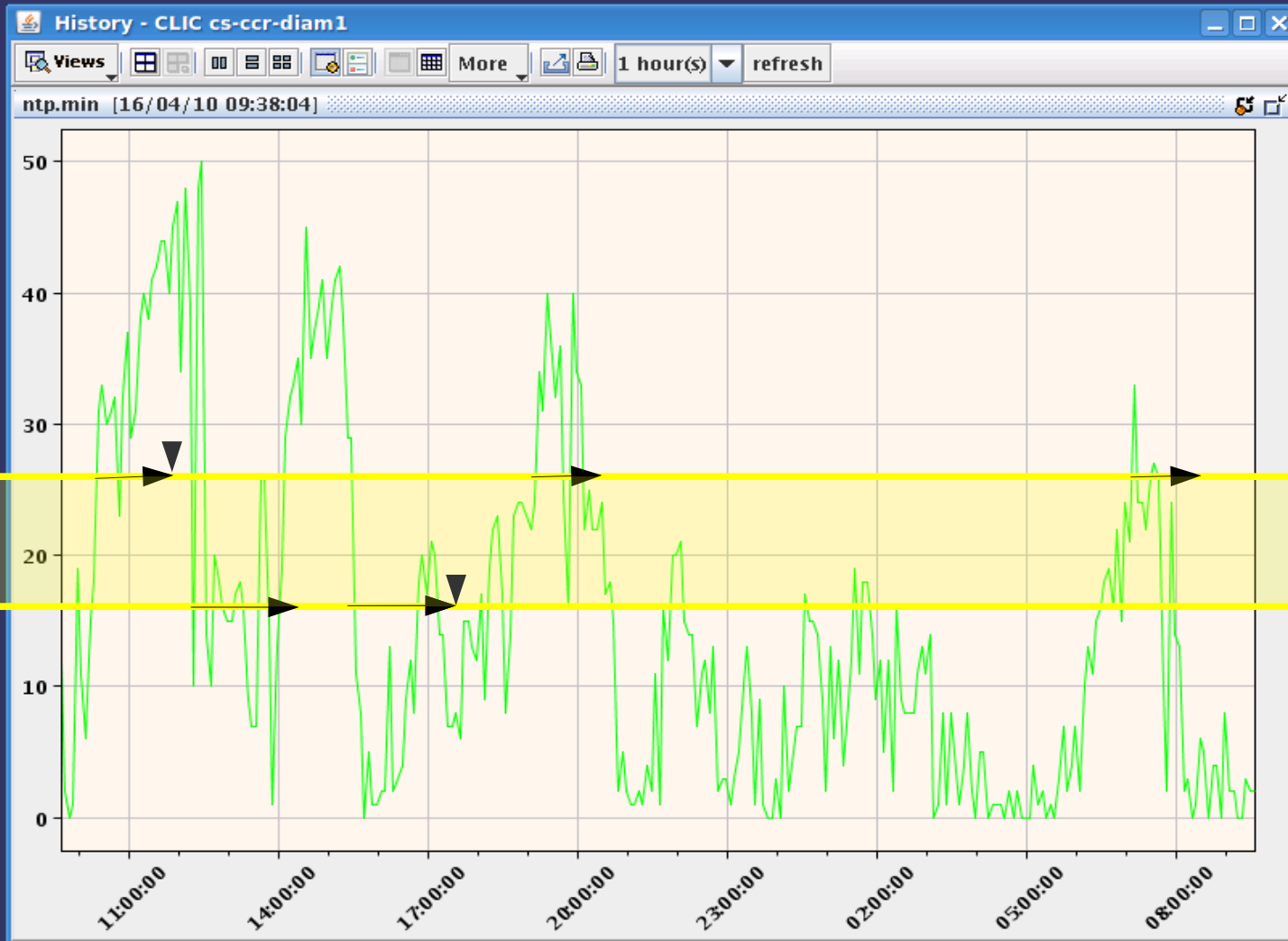
# Example 2: Adding delays





# Example 2: Result

(result: 2 state changes)



# *Data handling issue: Conclusion*

- ⇒ The system should display significant alarms only
- ⇒ To make alarms significant:
  - Use data analysis
  - Tune parameters (threshold, delays)
- ⇒ The data is needed, but once available allows also to:
  - Define alarms dynamically (no source change)
  - Combine data elements into a single alarm

Not reasonable to ask alarm providers to implement the features in their code

# *About technology*

# *About technology*

## ⇒ LASER/DIAMON uses SonicMQ and OC4J:

- Requires a license (Sonic)
- Old versions, difficult to maintain or migrate.
  - OC4J is not supported anymore by Oracle
- Not BE-CO standard (anymore!)

## ⇒ LASER 1 was neglected in favor of LASER 2

- ... which did not reach production level due to changes in org.

LASER/DIAMON should move to more BE-CO standard technologies  
(ActiveMQ, Spring, JAPC, device property model)

# *Missing functionality*

# *User interfaces*

- ⇒ Security restrictions degraded the service
  - No servlets available anymore (statistics)
  - GUIs working only on TN
- ⇒ Need to provide additional UIs
  - Web interface(s) to provide access from outside TN
  - RSS feeds
  - Development kit for GUIs (JViews ?)
- ⇒ Lack of DIAMON / LASER integration

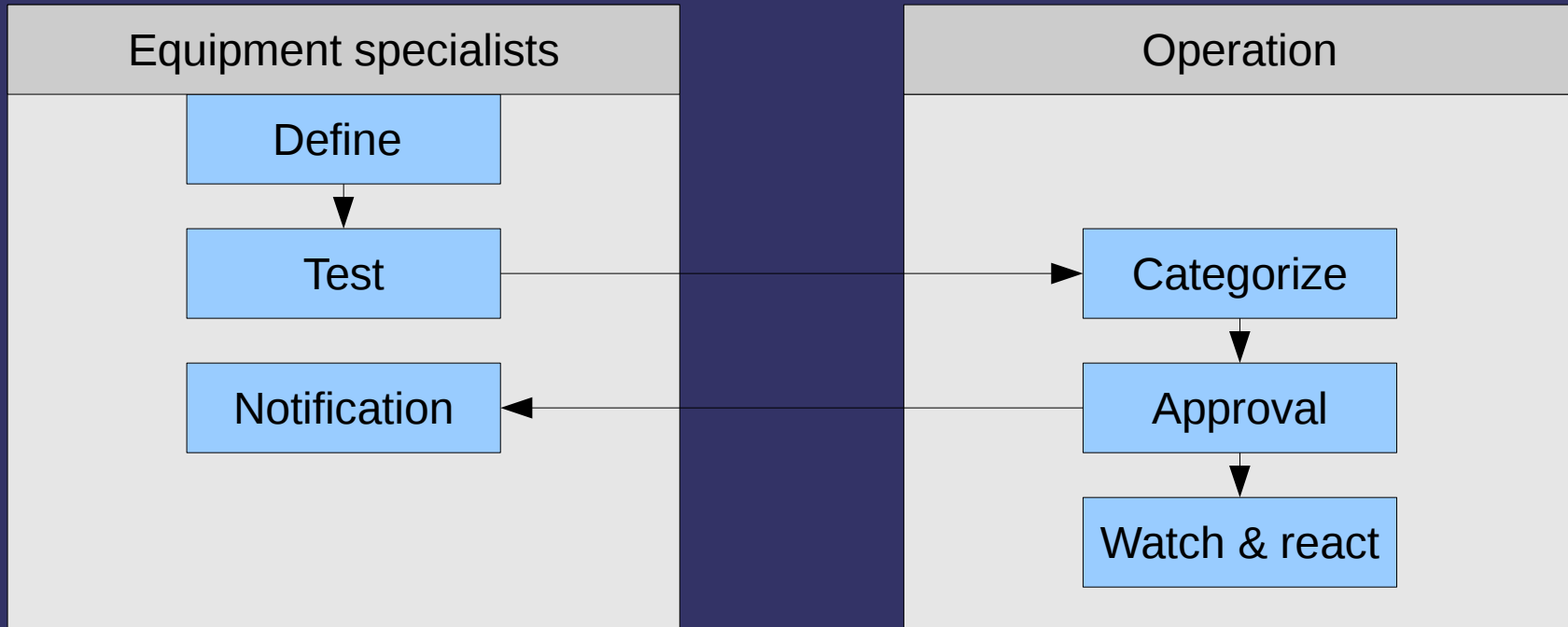
# *Provide tools for data management*

- ➔ **More statistics must be available for operations:**
  - Long living alarms
  - Oscillating alarms
  - Alarms per day / shift
  - Display of masked/inhibited alarms
  - Short living alarms (problem when used with auto-terminate)
- ➔ **Configuration management**
  - data browser
  - tool to categorize alarms
  - APIs / interfaces for alarm providers

Replace weekly restart for alarm definition import!

# Work flow

- ➔ Need guarantee that alarms are monitored



- ➔ Alarm test procedure and tools to
  - Evaluate alarm parameters
  - Check parameter changes against historical data
- ➔ Define clear responsibilities:
  - LASER/DIAMON team: develop, operate and support
  - Equipment: expose data, parametrize alarms
  - Operation: monitor, react, adjust parameters



# Miscellaneous

- ⇒ Integrate CMW admin
  - ... and follow their plans to go “device/property”
- ⇒ Provide support for Windows monitoring
- ⇒ Better integration with other tools like
  - wreboot
    - Give the possibility to start a given service “anywhere”
    - Check that a service is not started more than once
    - Centralize logging actions for services
    - Unique approach for all systems (not servers <> FECs)
    - Better control of processes started at boot time
  - Remote reset
    - Detect “à priori” if the action is possible

# *Scope of the project*

# *Summary of current situation*

- ⇒ Important enhancements are needed
- ⇒ The logic is distributed over many data / alarm sources
  - For LASER: 200 instances of type TIM, DIAMON, Spectrum, Alarm monitors, ENS, SPS-RF, SPS-Power converters, PVSS Vacuum, PVSS cryo, Nodal, SPS radio-protection, ...)
  - For DIAMON: 1'500 instances of type CLIC, CMW, SNMP, ...
- ⇒ Technology is non-standard and outdated

# *Scope of the project*

- ⇒ Rewrite or replace (evolution is not enough)
- ⇒ More technical than functional
  - First implement new design, than provide new features
- ⇒ Should possibly include auxiliary tools

# *Part II Design*

*(presentation by Marek)*

# *Part III Options for implementation*

# *Overview of part III*

- ➔ Options for implementation
- ➔ Project milestones
- ➔ Conclusion

# *Implementation options*

- ➔ Rewrite using existing components
- ➔ Build on existing app (INCA)
- ➔ Common project with TIM



# *Option 1: Rewrite based on existing components (JAPC etc.)*

- Pro's:

- It will best suit our needs
- Full control over design and functionality

- ⇒ Con's:

- High development workload
- Reduced benefit from others experience

# *Option 2: Build on existing app (INCA)*

- Pro's:

- Same or similar data acquisition
- Project in the same group, same building

- ⇒ Con's:

- Not in the same domain (INCA is controls, not alarms/monitoring)
- Project completed or close to be completed

Approach mutually exclusive with option 3 (TIM)

# *Option 3: Common project with TIM*

- Pro's:

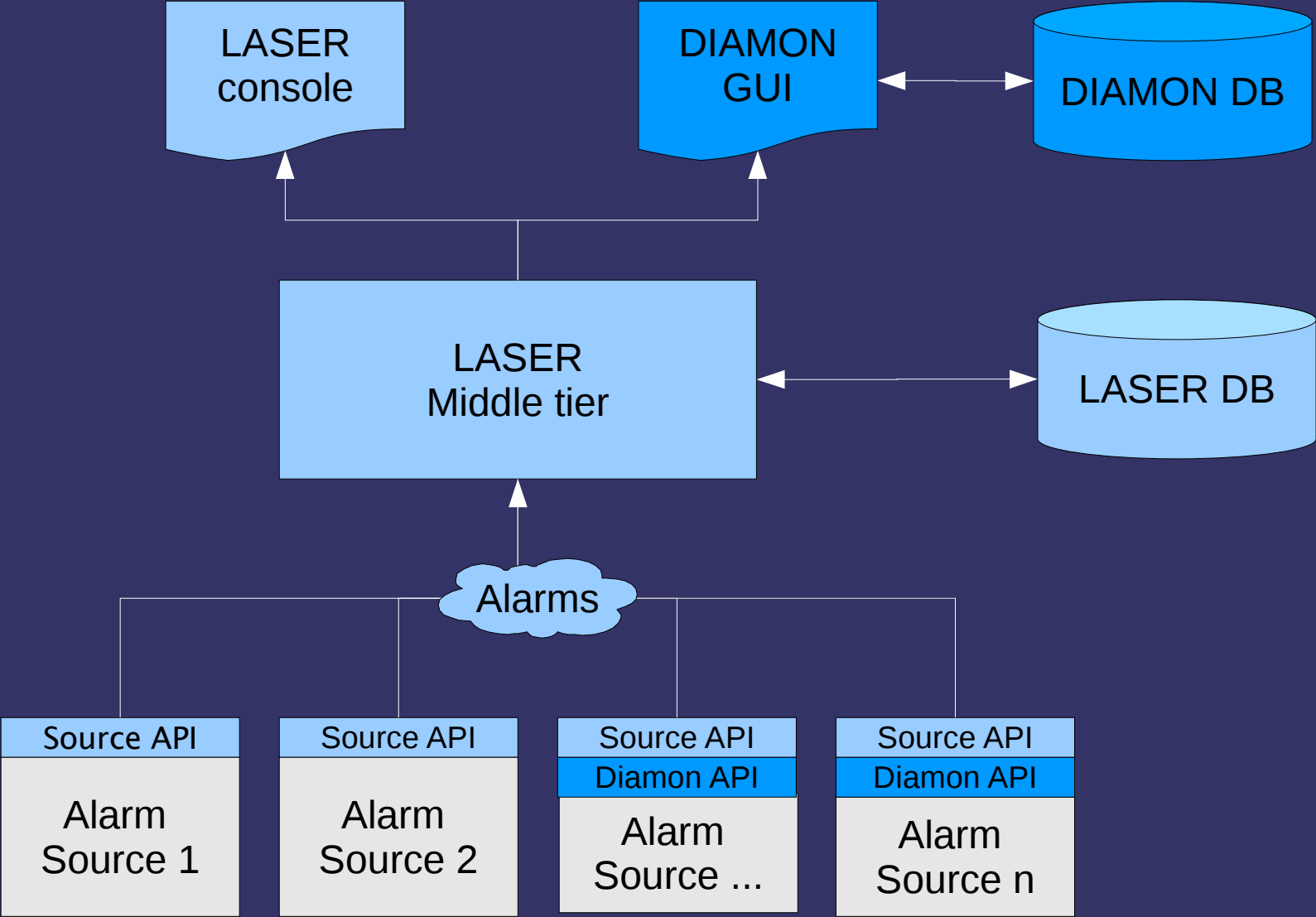
- Similar data acquisition
- Same domain (monitoring and alarms)
- Same technical situation (move from OC4J to Spring)
- Possible sharing of development and support

- ⇒ Con's:

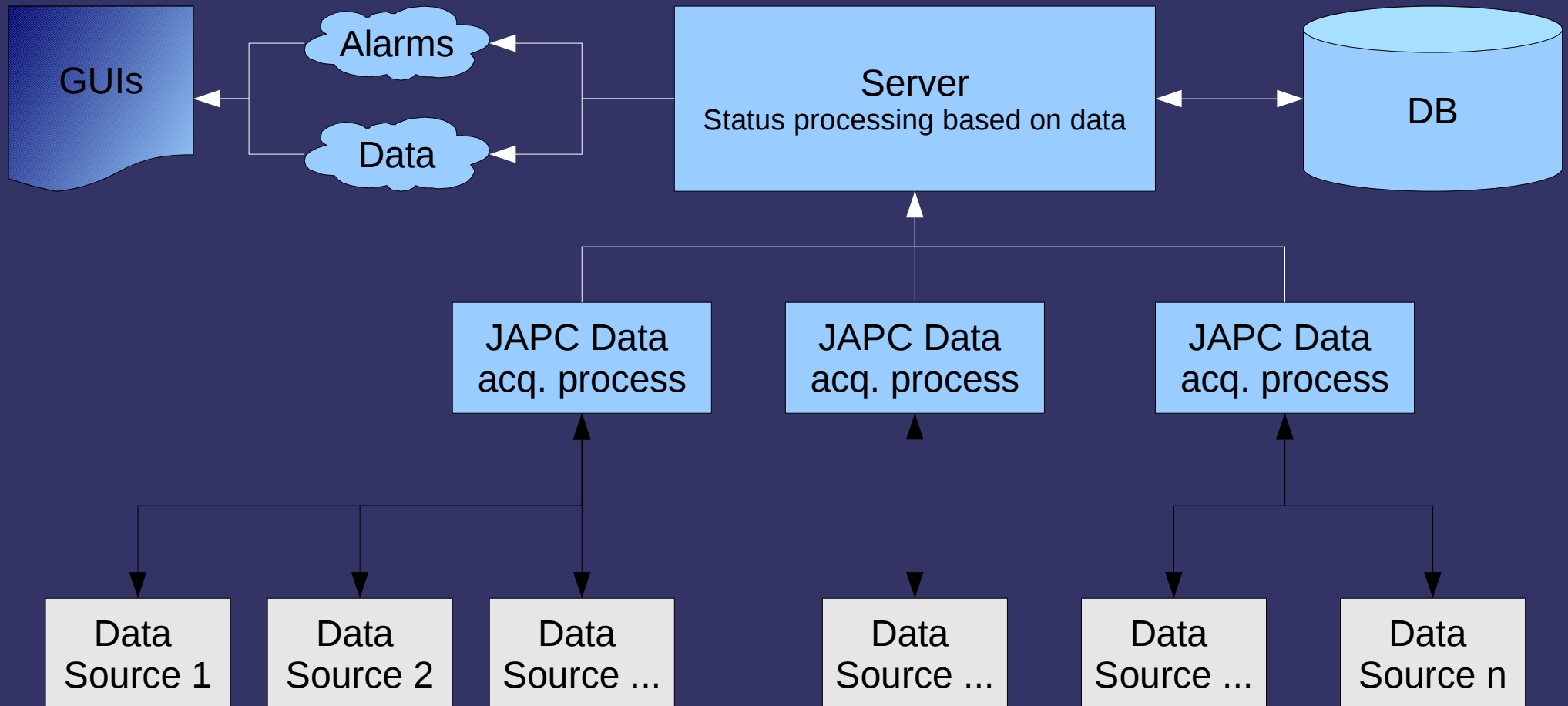
- Not in the same department
- Not always common views on technology
- Previous attempt blocked on resources and organization
- Additional constraints from existing TIM

Approach mutually exclusive with option 2 (INCA)

# Actual LASER / DIAMON design



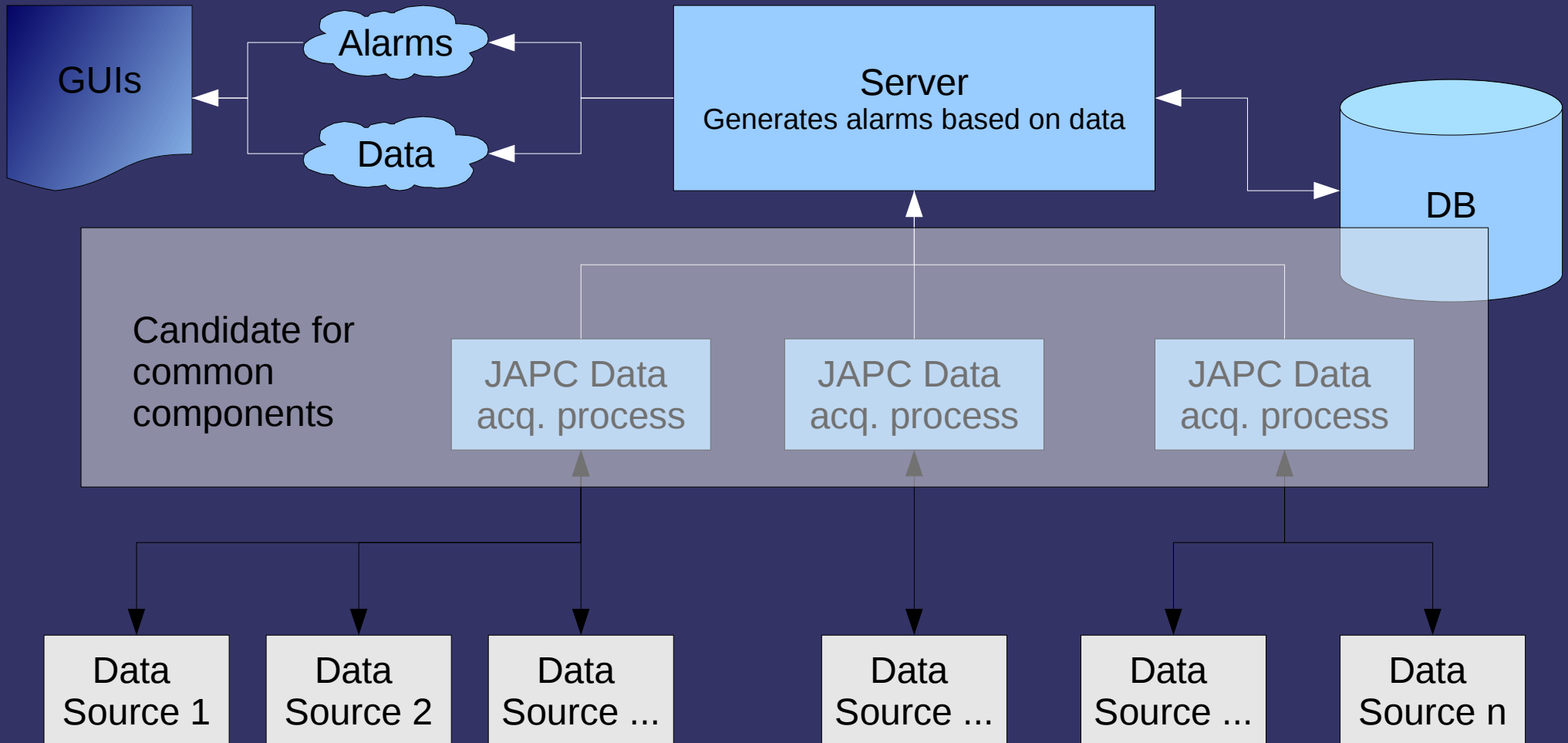
# Summary of proposed design



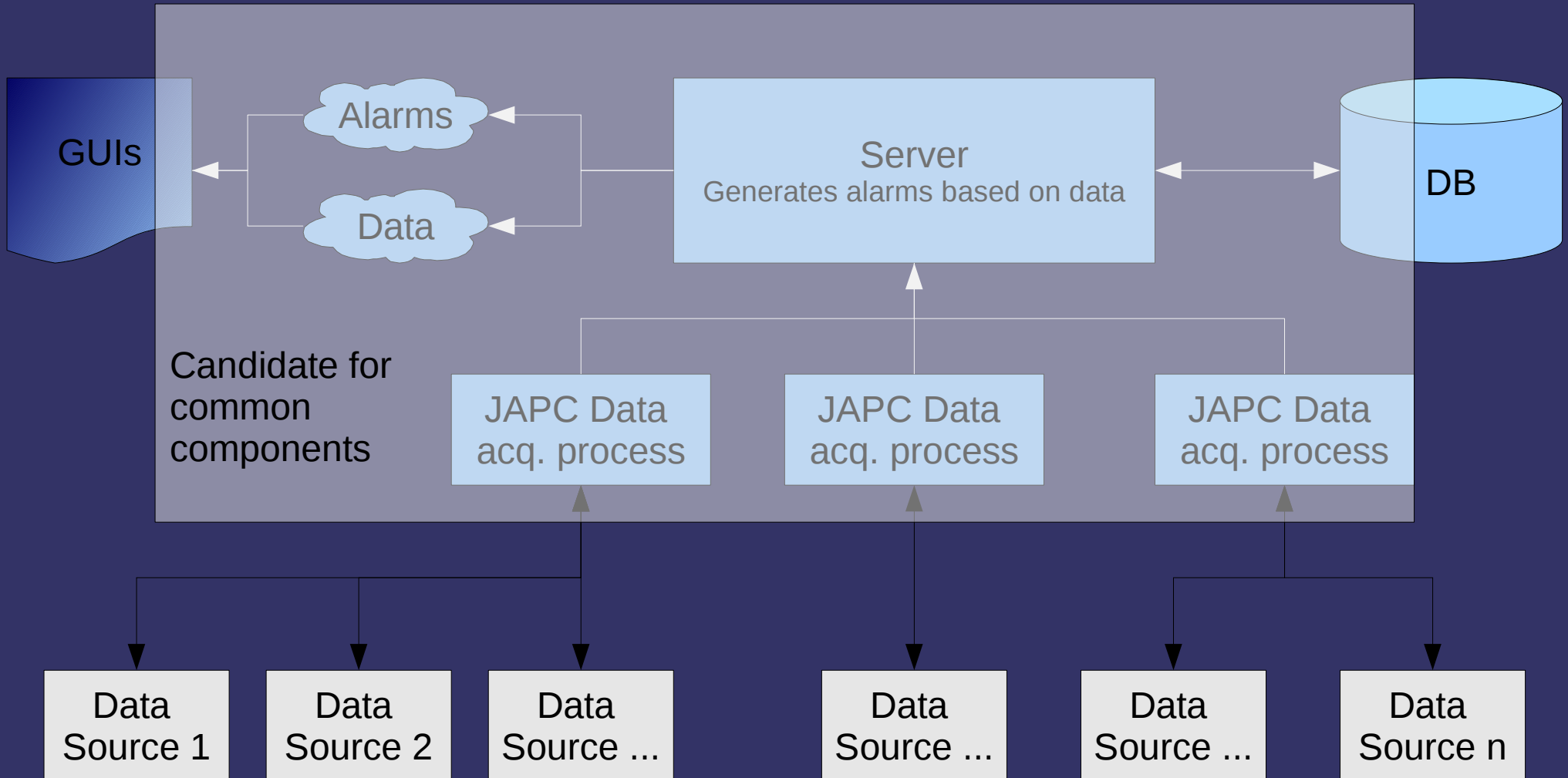
# *Benefits*

- ⇒ DIAMON and LASER become one product
- ⇒ Communication is controlled by subscription
- ⇒ Non-intrusive (no alarm/mon. Component in sources)
- ⇒ Standard technology is used
- ⇒ Data acq. allows alarm quality improvement

# The INCA option



# The TIM option





# *Milestones*

# Milestones

## ⇒ May 2010

- start prototyping with TIM and INCA

## ⇒ June 2010

- Make decision for TIM / INCA or other options
- Collect additional user requirements

## ⇒ December 2010

- Version 1.0 for // run

## ⇒ Production at next shutdown (end 2011 ?)

# *Conclusion*

## ⇒ Objectives

- Update of technology
- Merge LASER/DIAMON into a single application
- Process data in the alarms/monitoring middle-tier

## ⇒ Change in philosophy

- “Process data” instead of “concentrate alarms”
- Gives the possibility to provide better functionality