InGRID: A Generic Autonomous Expert System for Grid Nodes Real Time Monitor and Control

Francesc Pérez Picazo (Telefónica I+D)

June 2007. KTH, Stockholm

www.eu-egee.org

Telefónica Investigación y Desarrollo

PIC port d'informació científica

Information Society and Media

EGEE and gLite are registered trademarks

INDEX

- INTRODUCTION
- DESIGN CONSIDERATIONS
- ARCHITECTURE
- DECISIONS MAKER
- CASE OF USE
- CONCLUSIONS

INTRODUCTION

- **INTRODUCTION**
- DESIGN CONSIDERATIONS
- ARCHITECTURE
- DECISIONS MAKER
- CASE OF USE
- CONCLUSIONS

INTRODUCTION

- InGrid project started with EGEE, beyond a collaboration between Telefónica I+D and PIC (South-West ROC).
- The project studies the problem of managing a Grid resource centre and how to improve this task using an expert system.

PIC
port d'informació
científica

Telefónica
Telefónica
Investigación y Desarrollo

INTRODUCTION

- **Resource center monitoring and operation:**
  - Computing centres must provide a high **Quality Of Service** (QoS), and to accomplish this QoS, an optimal monitoring and operation is required.
  - Many open source monitoring tools are used, such as Nagios, Ganglia, etc.
    - They are excellent at **displaying** the raw service incidences.
    - They can generate alarms and filter them based on **simple dependencies**, before notifying to the operator.
  - Even though, when a large number of alarms appear simultaneously, the operator can get overloaded.

➔ **HUMAN STILL HAVE LIMITATIONS**

INTRODUCTION

- **Operator's human limitations could be:**
  - Delay due to a work overload.
  - Fatigue.
  - Subjectivity.
  - Lack of experience.
  - Inattentiveness.
  - High cost for training of the operator.
- **To overcome these limitations, we propose InGRID**
- **InGRID will:**
  - Manage the large amount of monitoring information
  - Act as a **virtual operator** automatically executing actions on services to restore their regular state.
  - If the action succeeds, as any intervention of the operator is required, this leads to a **decrease of the operator workload**.

DESIGN CONSIDERATIONS

- INTRODUCTION
- **DESIGN CONSIDERATIONS**
- ARCHITECTURE
- DECISIONS MAKER
- CASE OF USE
- CONCLUSIONS

DESIGN CONSIDERATIONS

- **Modularity**
  - InGRID should be **adaptable** to any monitoring tool.
    - It will be **independent** of the monitor, and **complement** it.
  - We only require that the monitor publishes the status data somewhere can be parsed.
- **Integration**
  - Each monitoring tool has its set of particular **functionalities.**
- **Scalability**
  - A huge **load of monitoring data** will be generated.
- **Autonomy**
  - Resource centres are required to be **24x7x365 available**.
    - Only few of them can afford to do this with the attendance of at least one operator.
  - InGRID will respond automatically to those well defined incidences without the operator attendance.

DESIGN CONSIDERATIONS

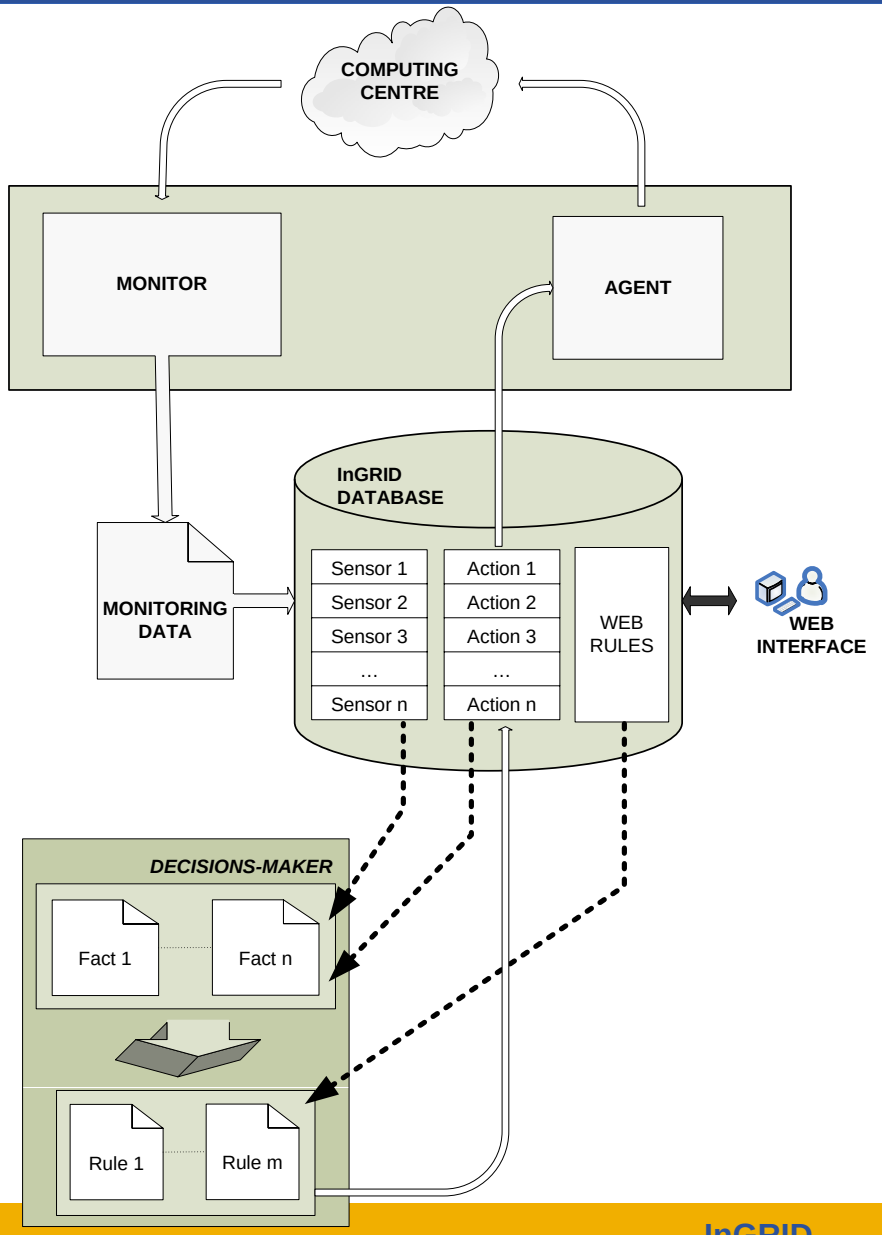
- **Real-Time actions**
  - Once an incidence is detected, the expert system must find out which **action** is required.
  - Decisions will be made in **real-time**.
- **Easy and clear to administrate.**
  - The hardest work with any expert system is the **administration.**
    - The administrator of the expert system must define an automatic procedure and design a set of rules for each service monitored.
  - To facilitate this administration task, InGRID will provide a **clear** graphical interface.

ARCHITECTURE

- INTRODUCTION
- DESIGN CONSIDERATIONS
- **ARCHITECTURE**
- DECISIONS MAKER
- CASE OF USE
- CONCLUSIONS

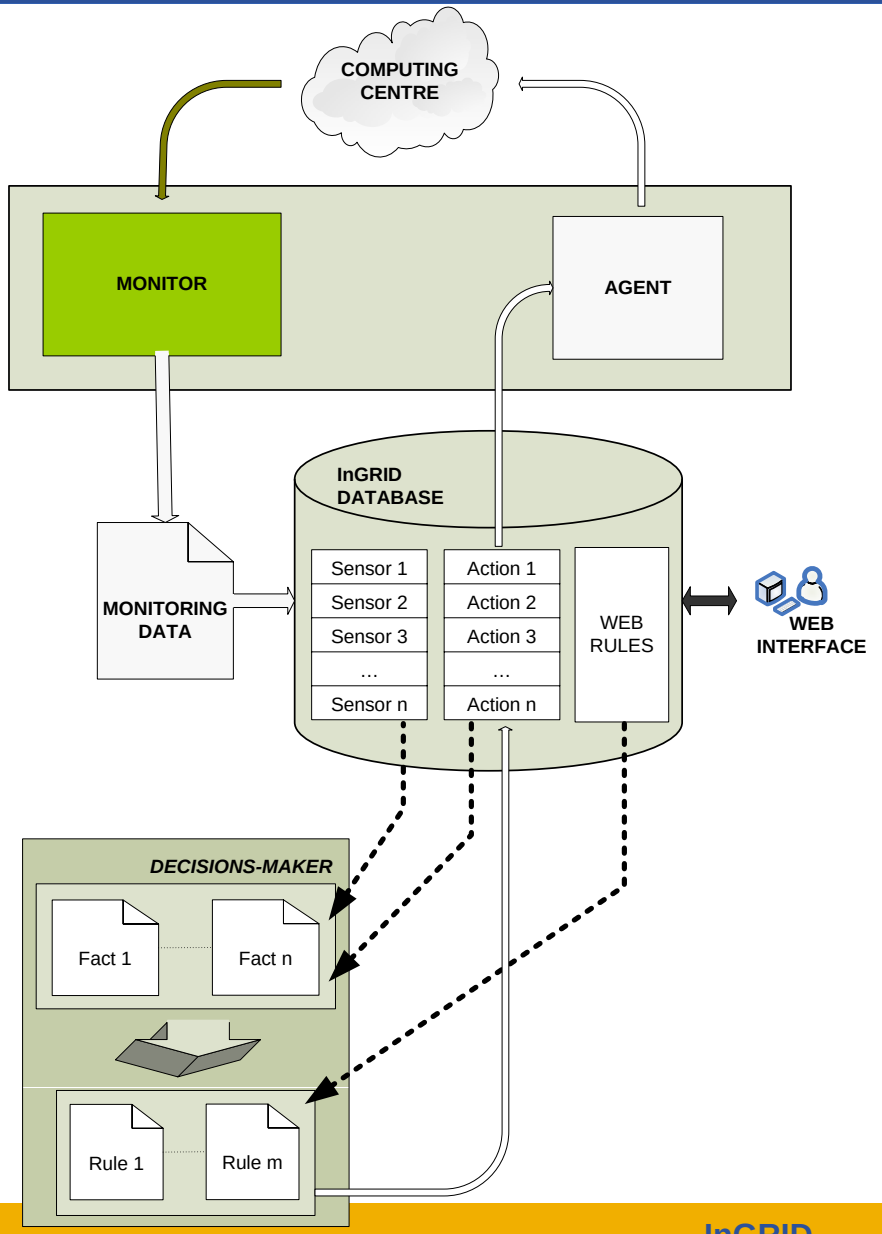
WORKFLOW

- **The InGRID workflow contains three main phases:**
  - Information Collection:
    - the expert system collects the monitored status of each element being supervised.
  - Decision Making:
    - the system analyses the information collected, and makes decisions according to defined rules.
  - Correcting:
    - each decision corresponds to an action that must be carried out.



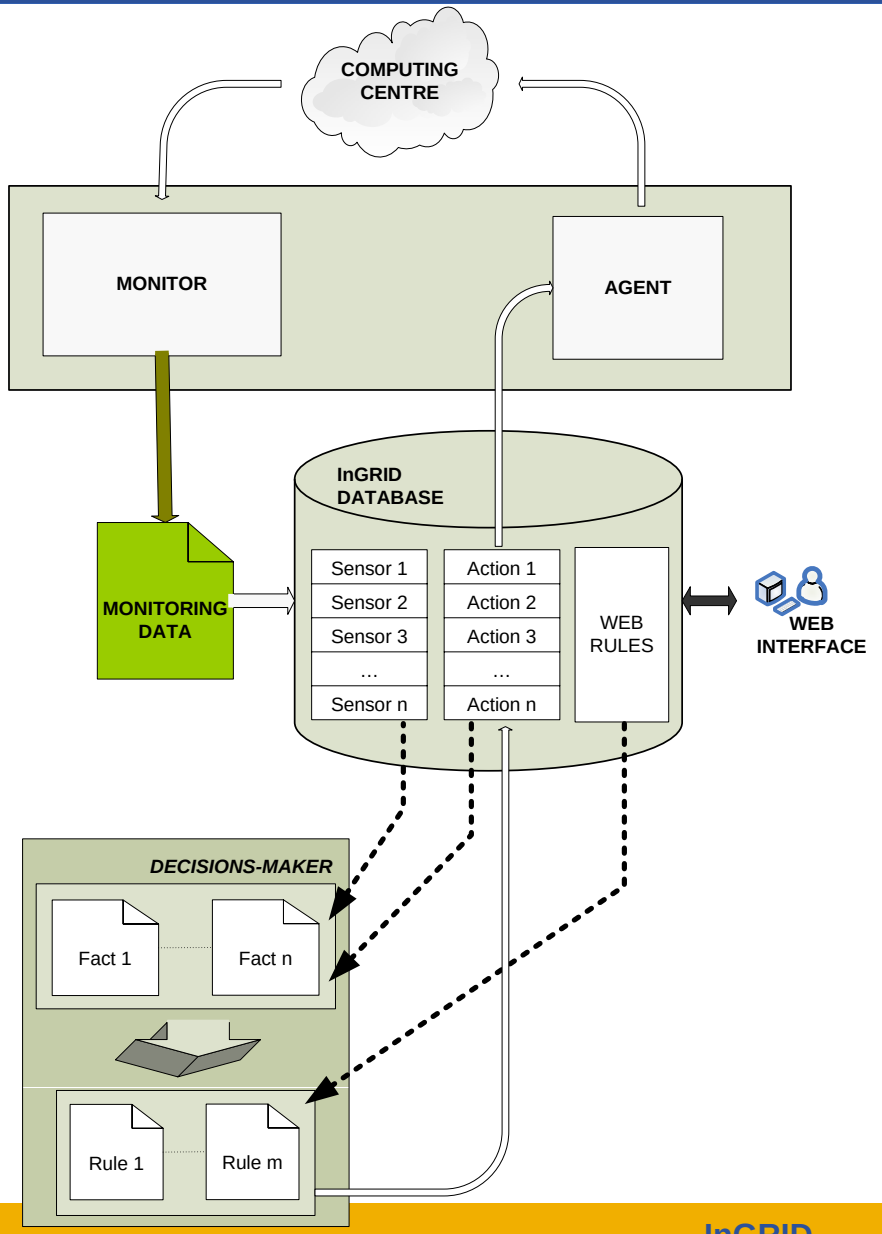
MONITORING (I)

The **Monitor** collects information about the status of each sensor.



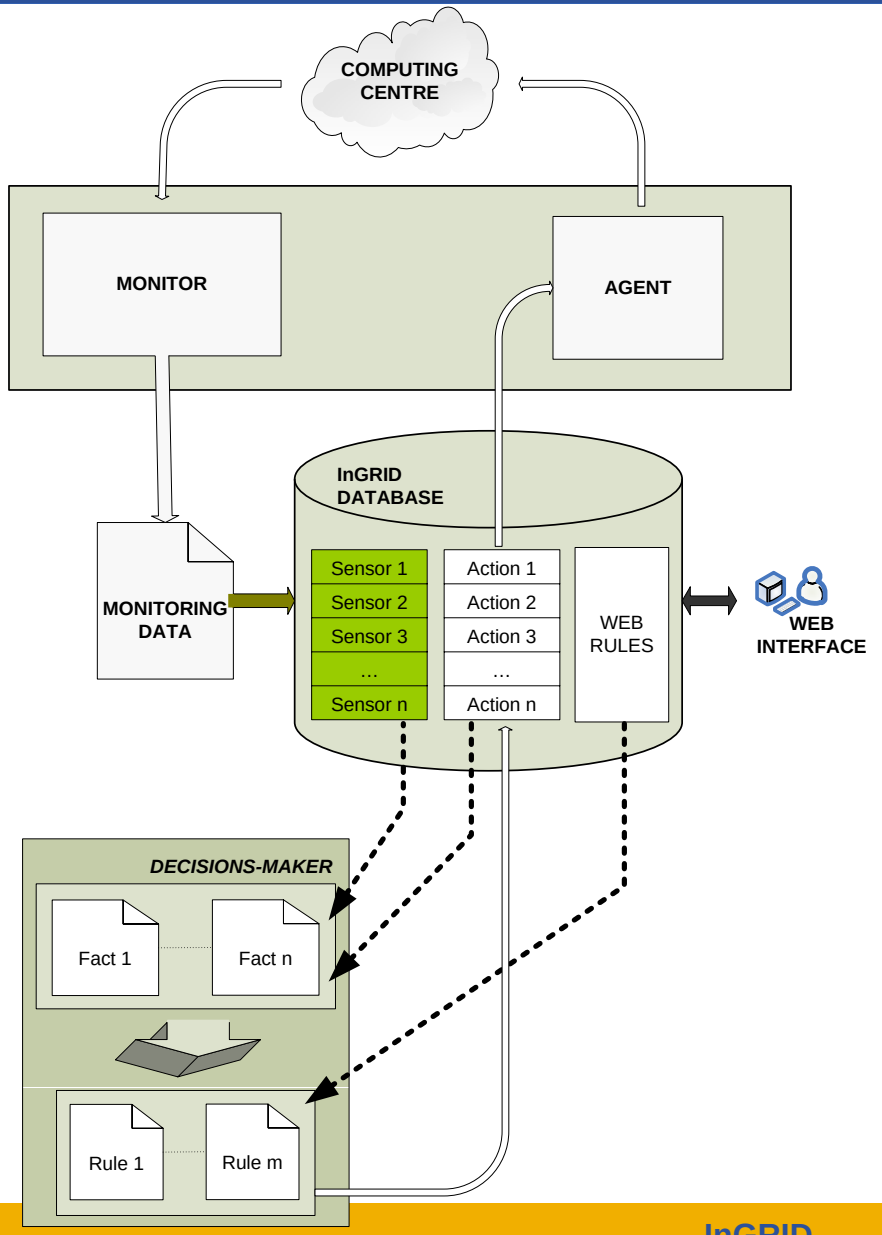
MONITORING (II)

The monitoring information collected must be stored into an output database or text file...



MONITORING (III)

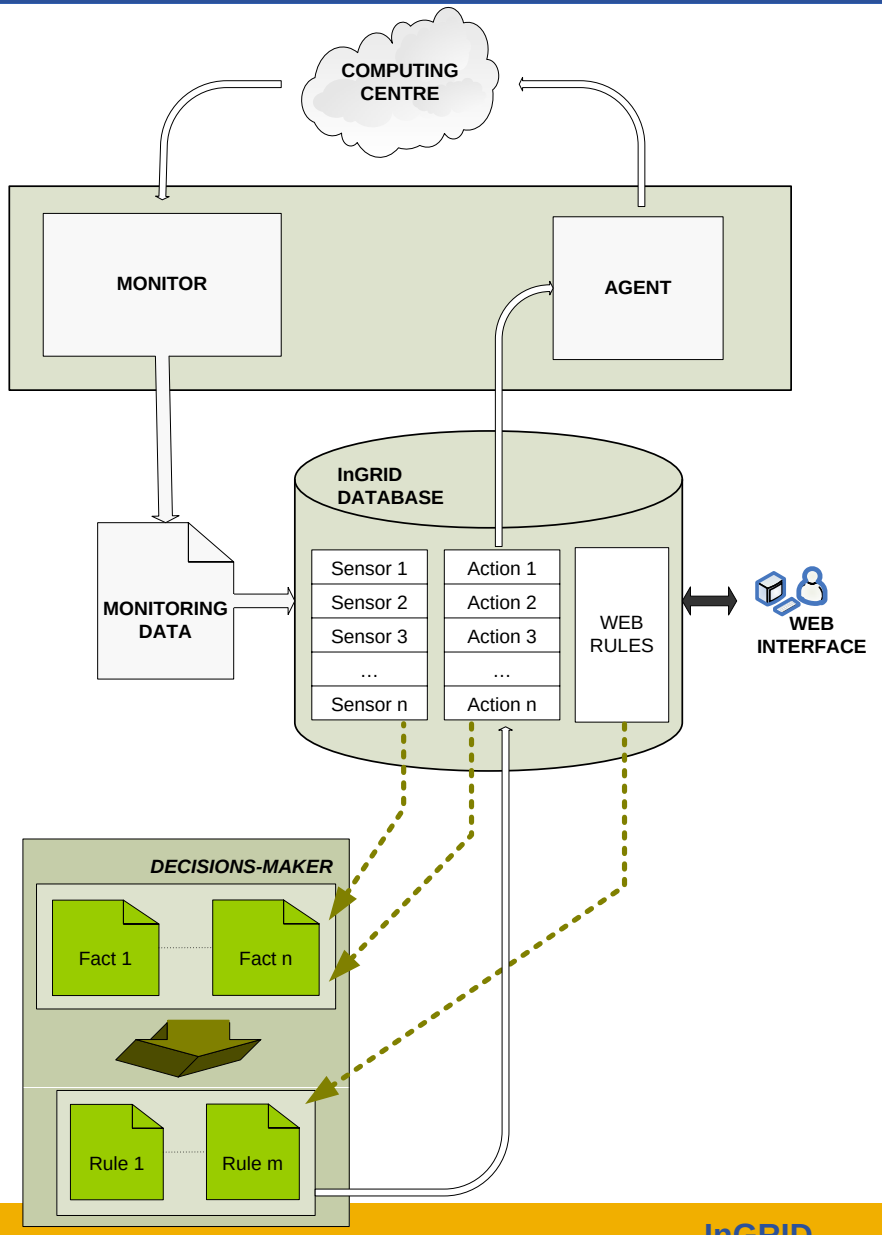
…so it can be dumped into InGRID database.



DECISION MAKING (I)

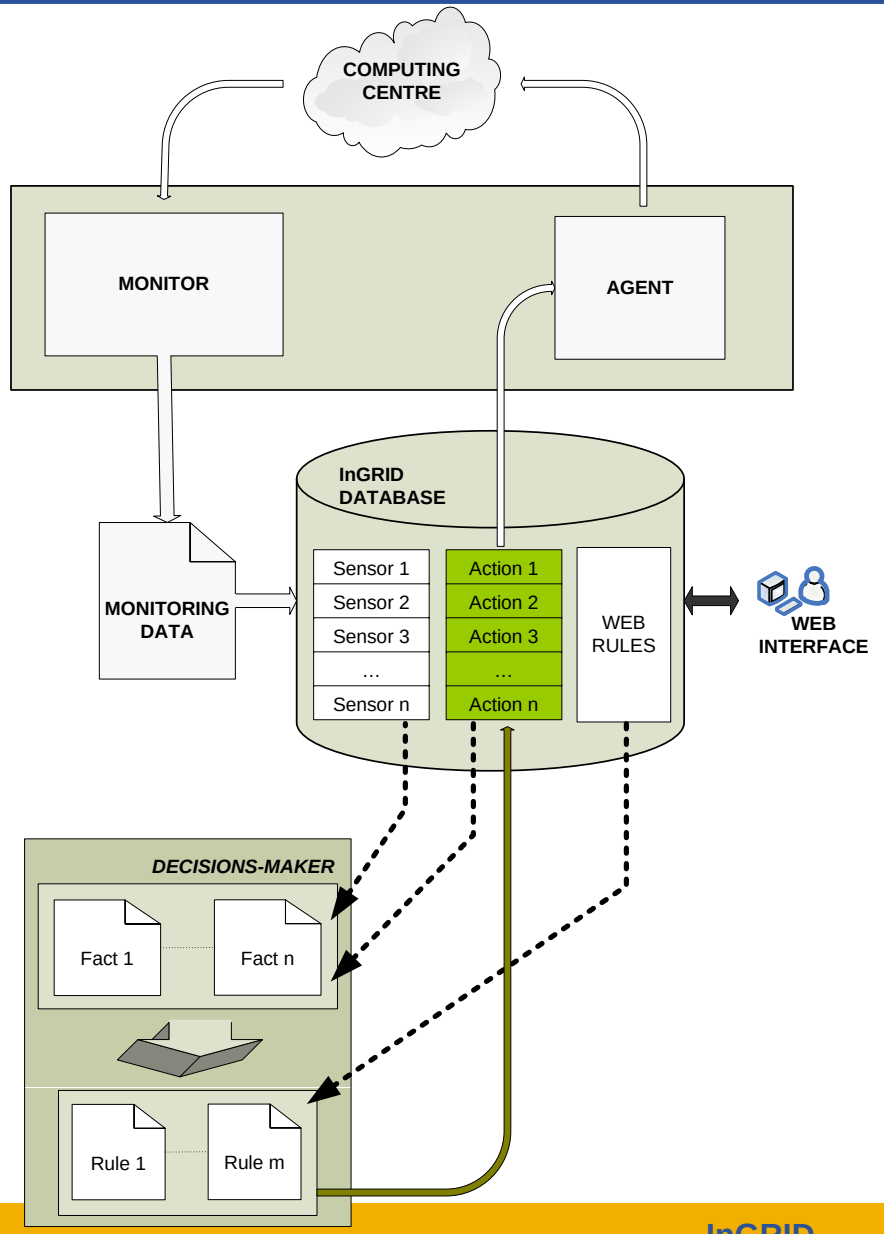
Data stored in InGRID database feeds the Decisions Maker internal memory.

The facts are questioned against the web rule conditions.



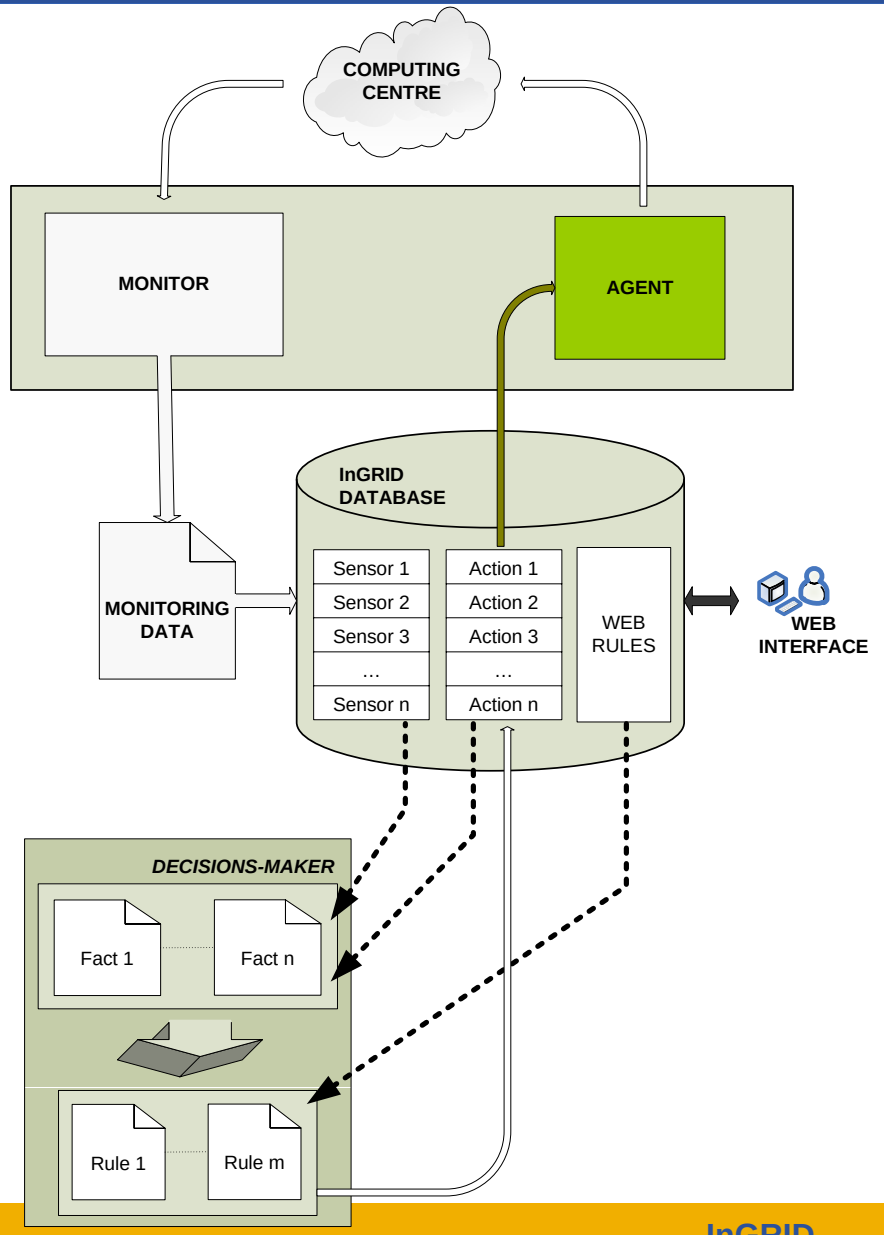
DECISION MAKING (II)

When the facts and the web rule conditions match, a rule is activated, and an action is scheduled.



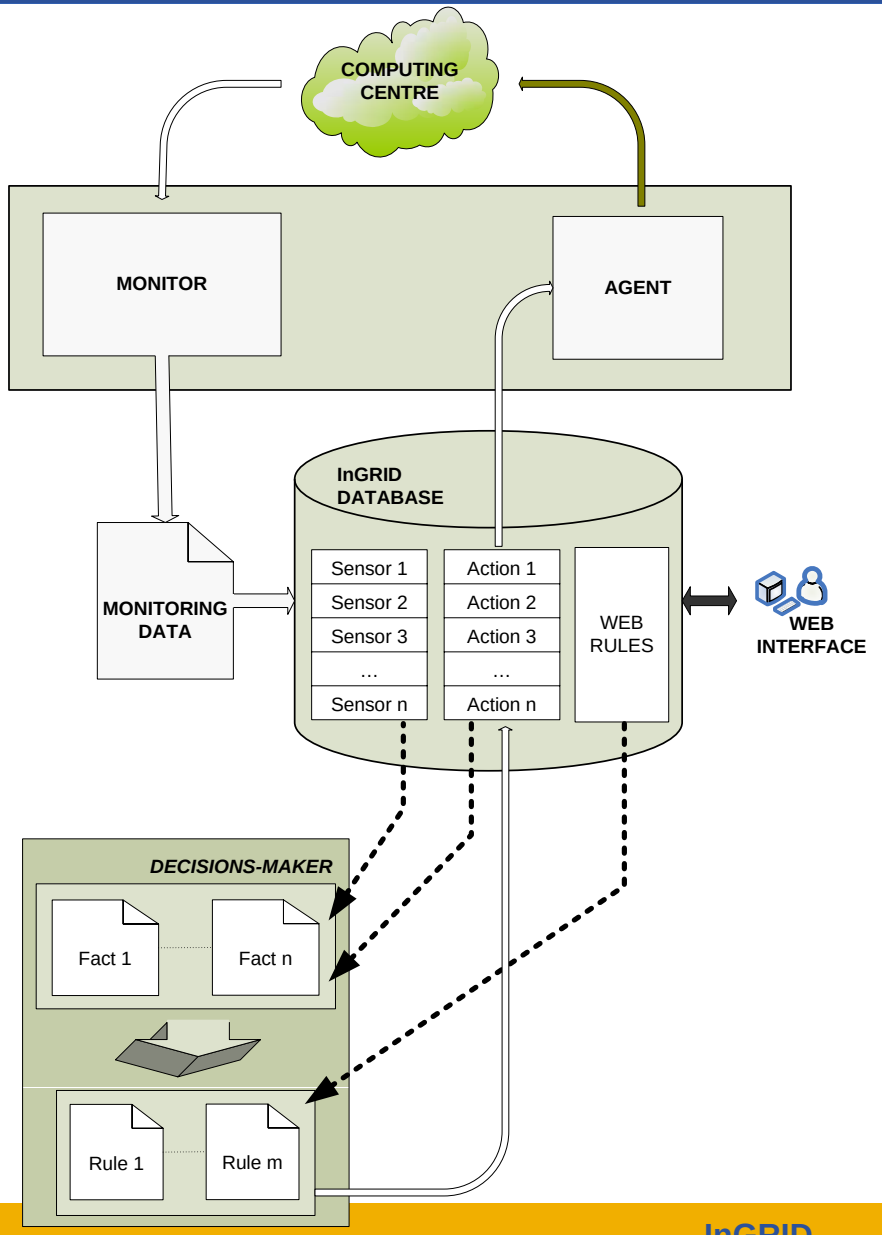
CORRECTING (I)

The Agent listens for actions scheduled by the Decisions Maker…



CORRECTING (II)

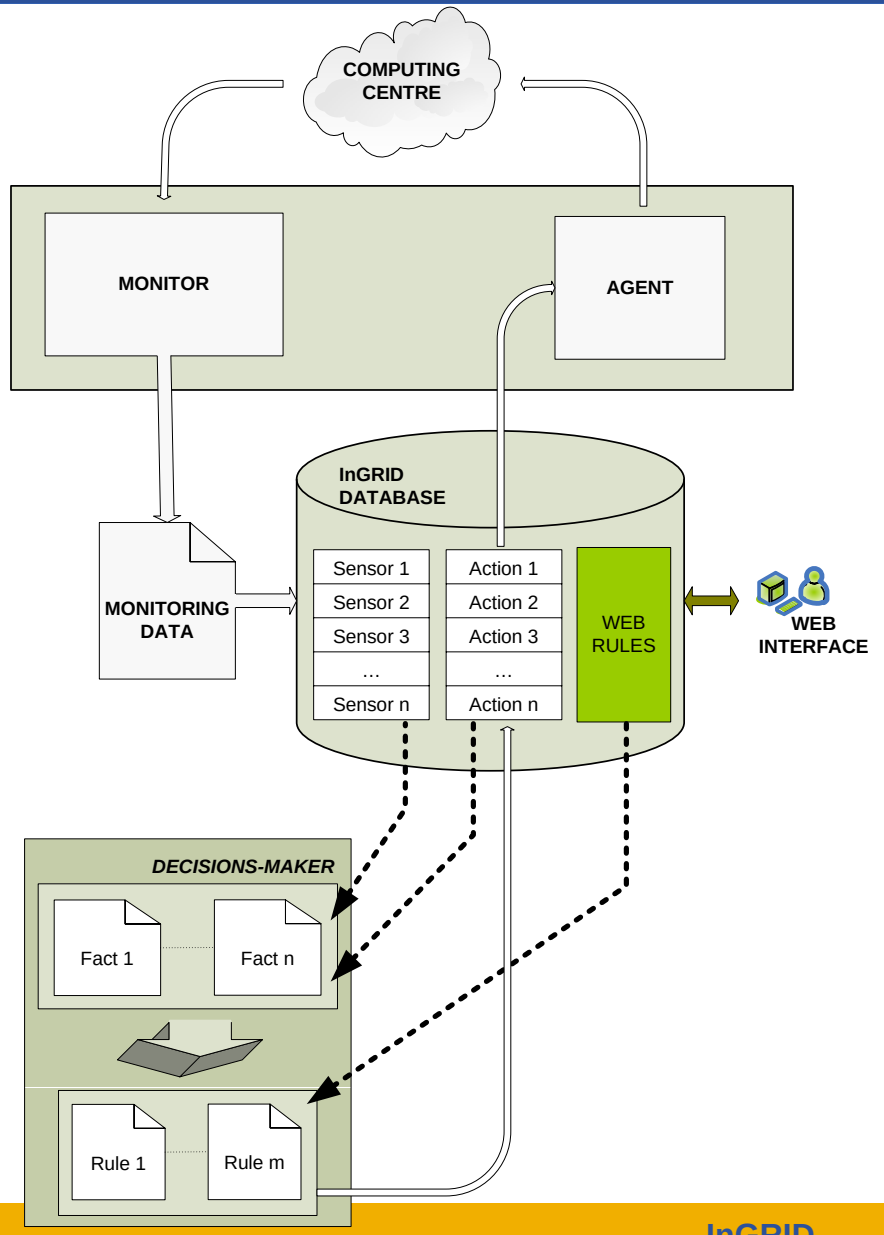
…and executes them.



ADMINISTRATION

The operator uses the web interface to administrate the expert system and configure the Web Rules.

Both Monitor and Agent are independent modules from the Decisions Maker. The communication between them is kept through the database.



DECISIONS MAKER

- INTRODUCTION
- DESIGN CONSIDERATIONS
- ARCHITECTURE
- **DECISIONS MAKER**
- CASE OF USE
- CONCLUSIONS

CLIPS (I)

- **The Decisions Maker is the core of the expert system.**
- **This module is based on a rule-based programming, which is one of the most commonly used techniques for developing expert systems.**

"**IF** clause, **THEN** action"

Rule-based programming

CLIPS

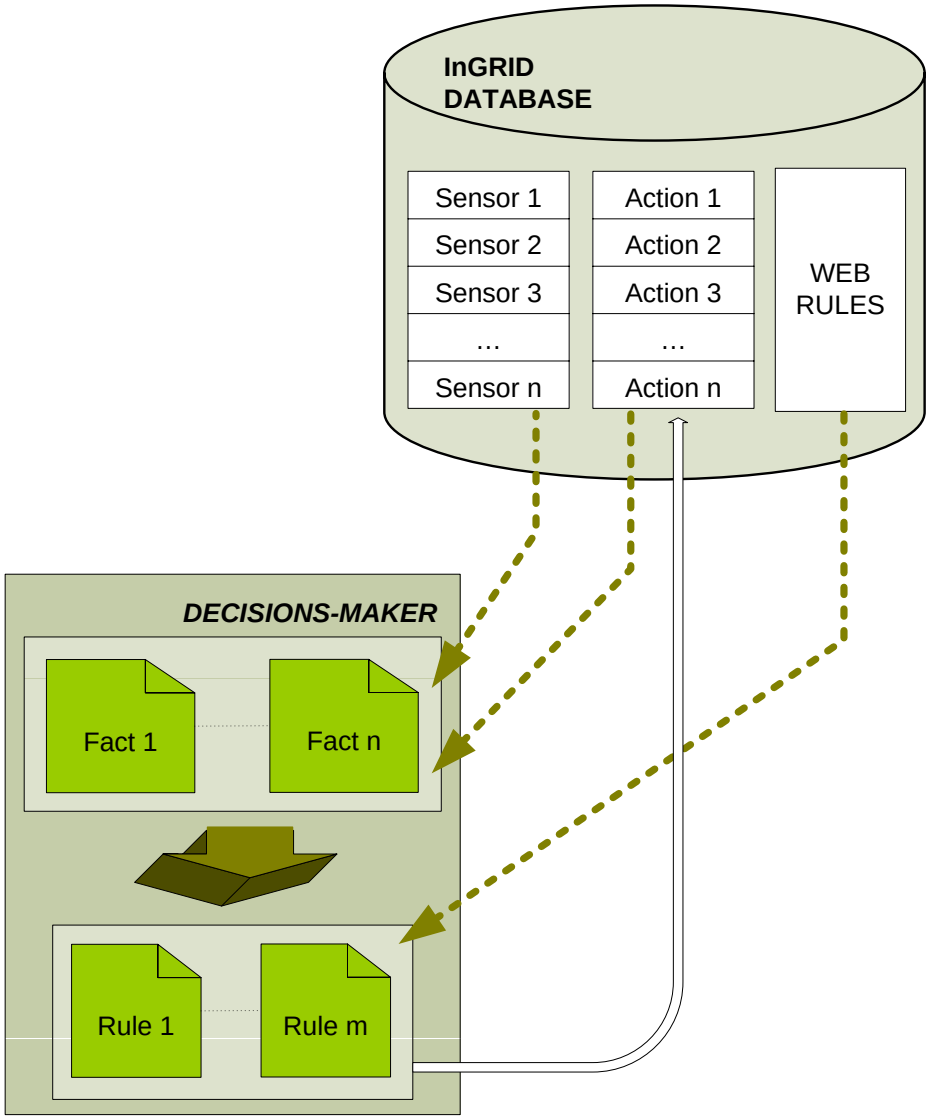
CLIPS (II)

- **CLIPS is a productive public domain development tool.**
- **Provides a complete environment for the construction of rule based expert systems.**
- **Key features:**
  - Knowledge Representation
    - cohesive tool for handling a wide variety of knowledge.
  - Portability
    - written in C for portability and speed.
  - Integration/Extensibility
    - embedded within procedural code, called as a subroutine.
  - Fully Documented
    - extensive documentation.
  - Low Cost
    - maintained as public domain software.

MEMORY FEEDING

Data stored in InGRID database feeds the internal memory of the Decisions Maker.

The **Facts** are composed by the monitoring data and the list of actions scheduled.
The **Web Rules** are programmed using the web interface.



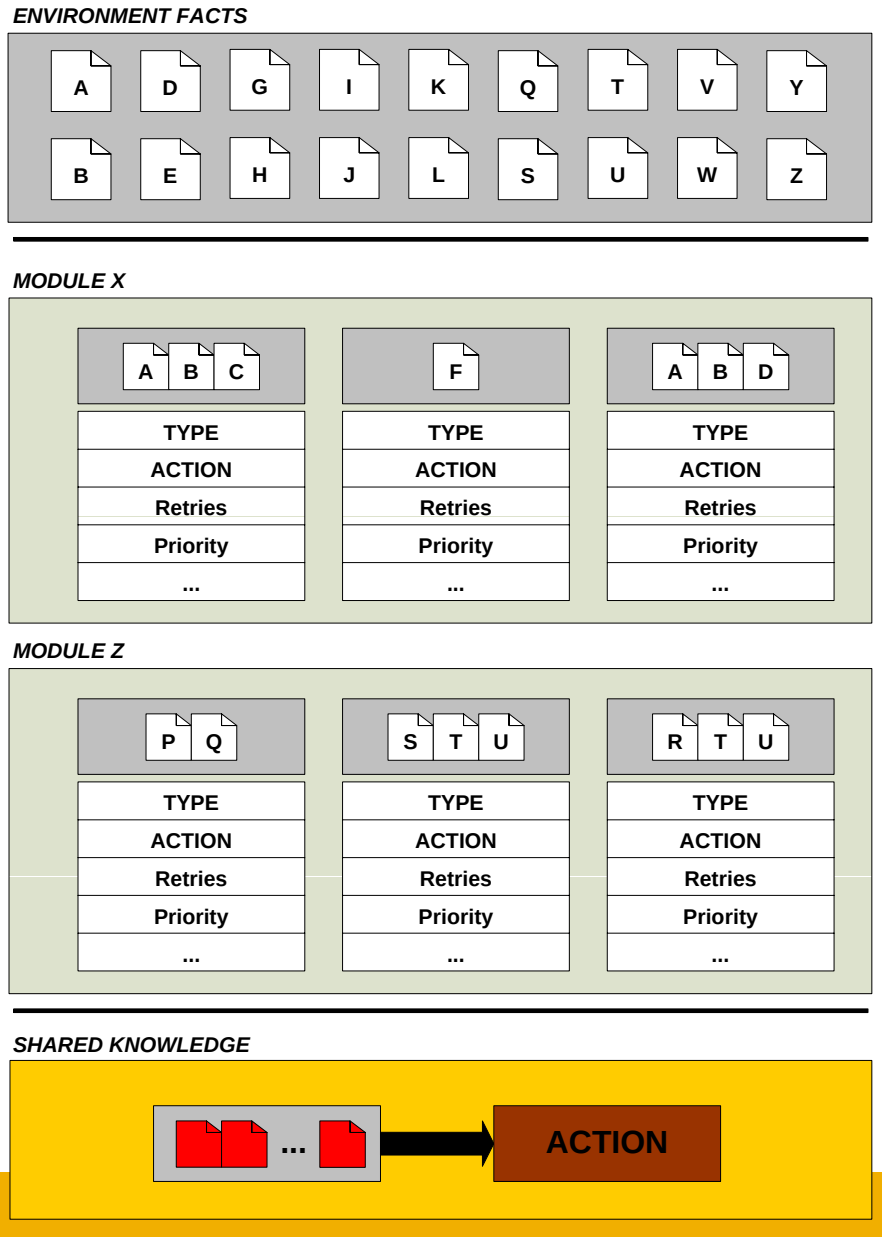
WEB RULES (I)

- **The Web Rules specify a set of patterns to find, and the actions to be performed when these patterns are found.**
- **A Web Rule is composed of an if" component (antecedent), and a "then" component (consequent).**
  - The antecedent is a set of **patterns** (conditions) that specifies the **facts** (or monitoring data) which cause the rule to be applicable.
  - The consequent is the set of **actions** to be executed when the rule is applicable.
- **To represent all kind of events that may appear in real situations, InGRID considers four types of rules depending on the state of the elements supervised.**

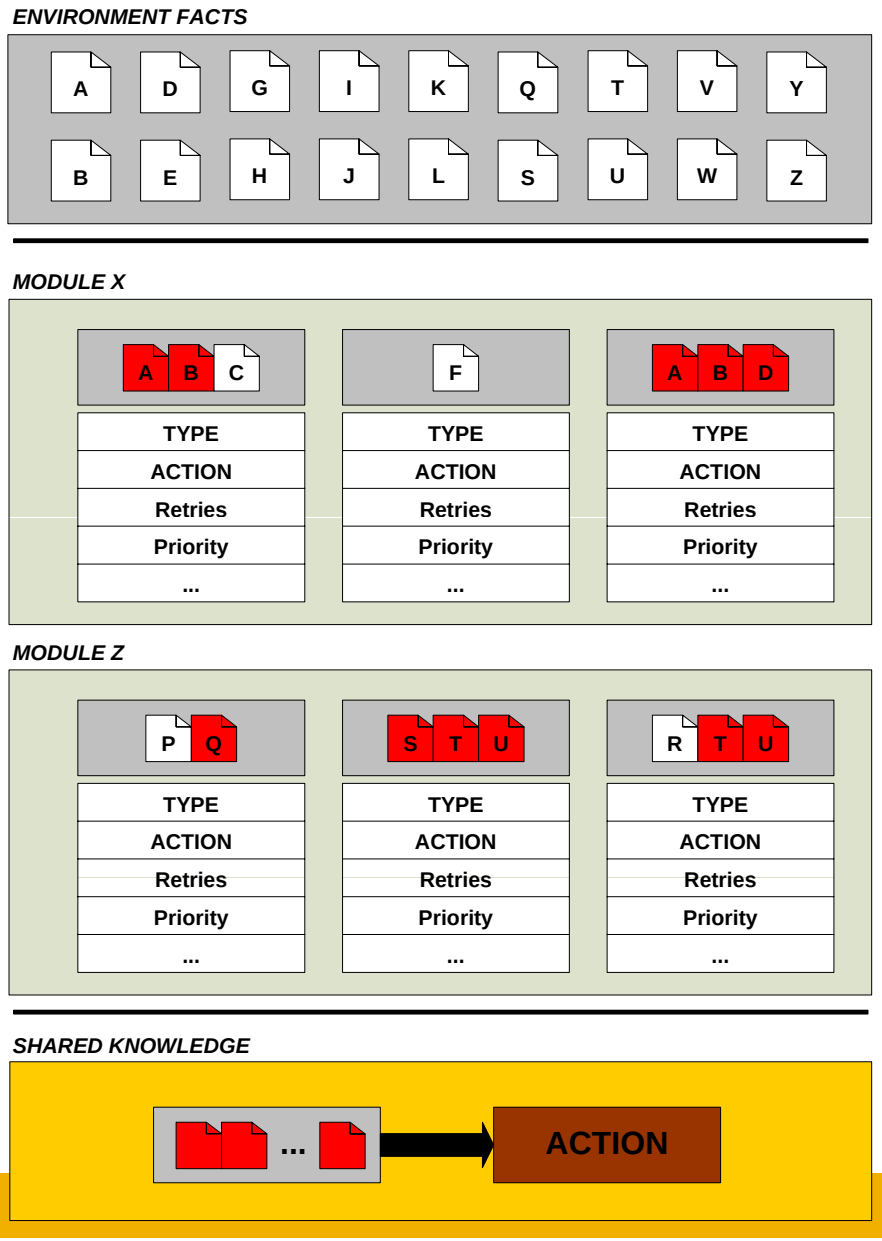
WEB RULES (II)

STABLE	**OK** state	**No action** is required
UNSTABLE	**hard error** state	An **immediate action** is required, independently of the state of any other element.
DOWN	**soft error** state	Previously, **dependencies** must be checked. The action will be schedule only if this error is the real cause of the problem
NOTIFY	-	**Notifies** by email the incident to the operator, without trying to schedule any action.

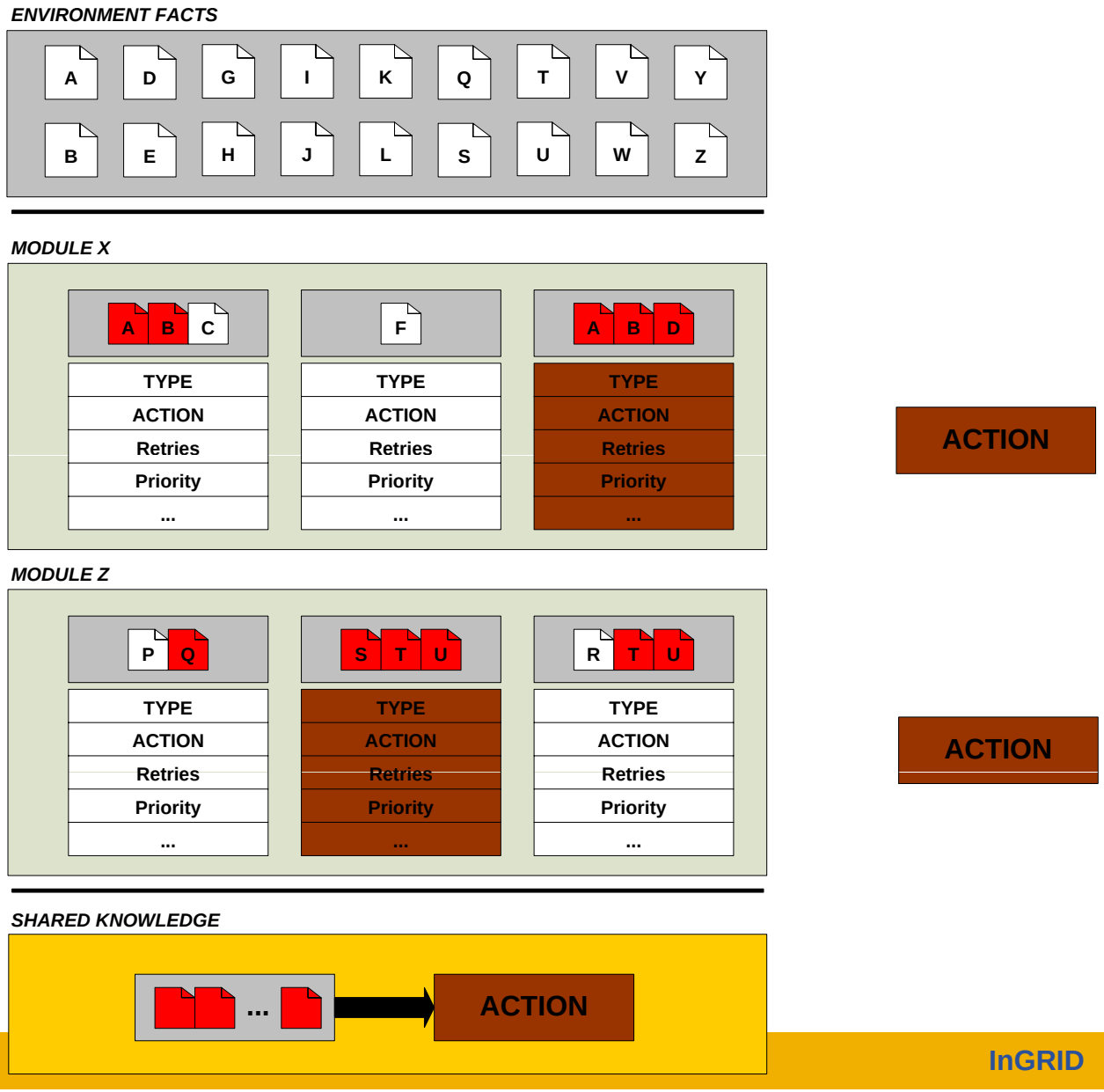
MATCHING


ENVIRONMENT FACTS
A D G I K Q T V Y
B E H J L S U W Z
MODULE X
A B C
F
A B D
TYPE
ACTION
Retries
Priority
...
MODULE Z
P Q
S T U
R T U
SHARED KNOWLEDGE
...
ACTION

MATCHING

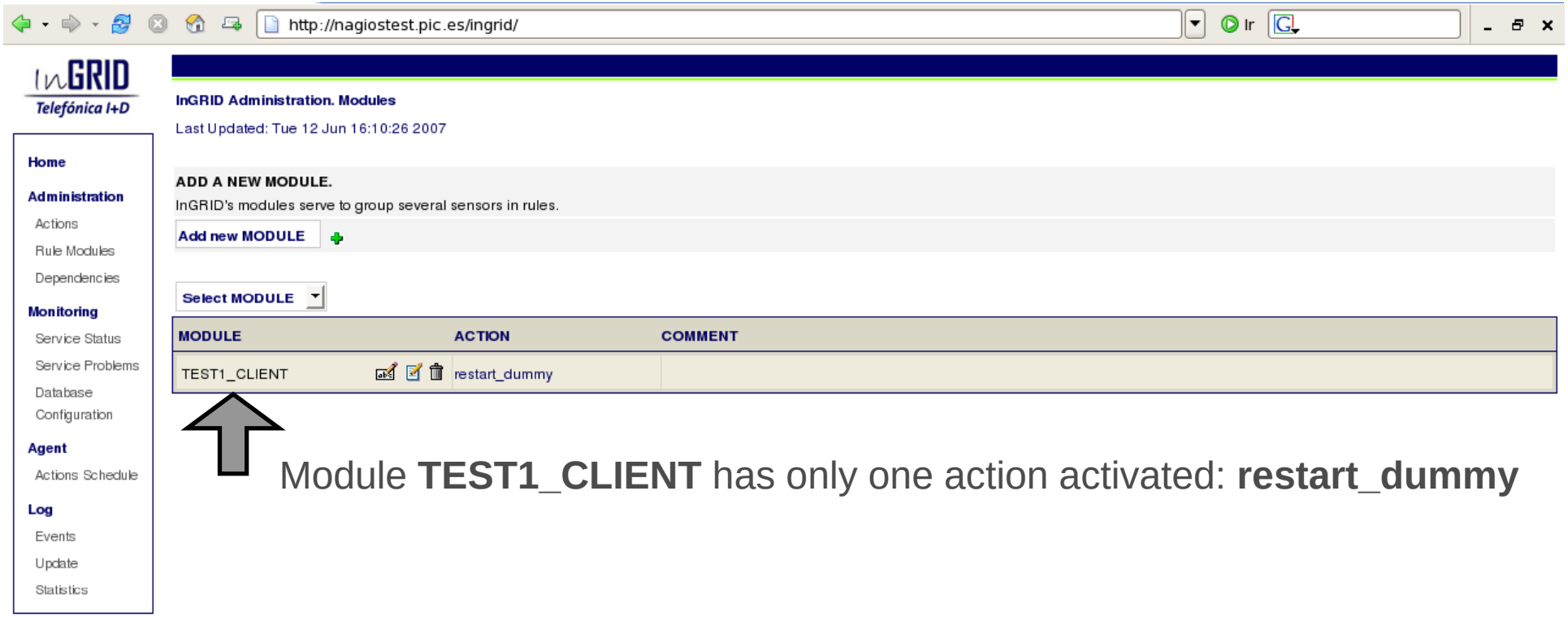

ENVIRONMENT FACTS
A
D
G
I
K
Q
T
V
Y
B
E
H
J
L
S
U
W
Z
MODULE X
A B C
F
A B D
TYPE
ACTION
Retries
Priority
...
MODULE Z
P Q
S T U
R T U
SHARED KNOWLEDGE
...
ACTION

MATCHING

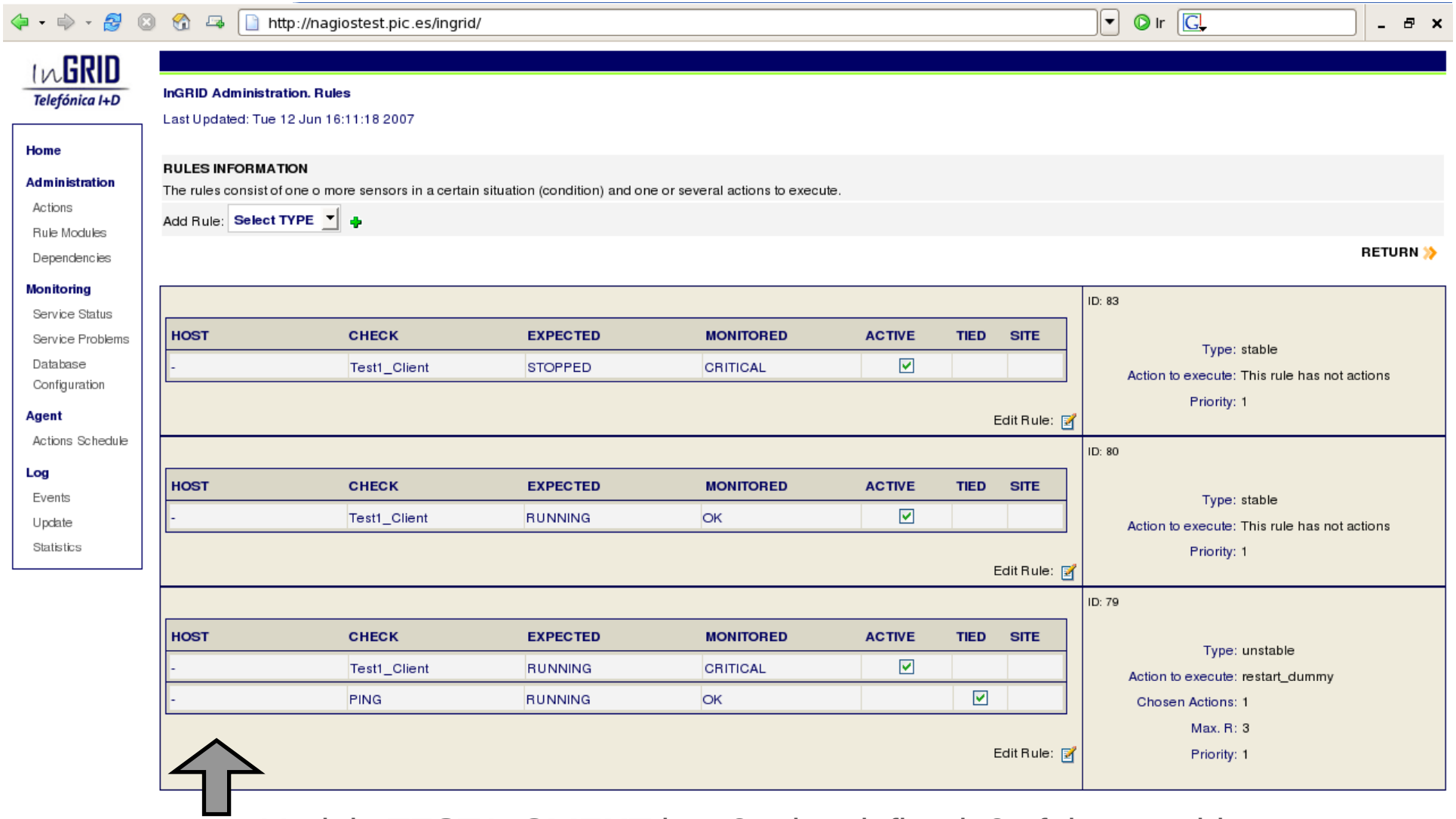

ENVIRONMENT FACTS
A D G I K Q T V Y
B E H J L S U W Z
MODULE X
A B C
TYPE
ACTION
Retries
Priority
...
F
TYPE
ACTION
Retries
Priority
...
A B D
TYPE
ACTION
Retries
Priority
...
ACTION
MODULE Z
P Q
TYPE
ACTION
Retries
Priority
...
S T U
TYPE
ACTION
Retries
Priority
...
R T U
TYPE
ACTION
Retries
Priority
...
ACTION
SHARED KNOWLEDGE
...
ACTION

CASE OF USE

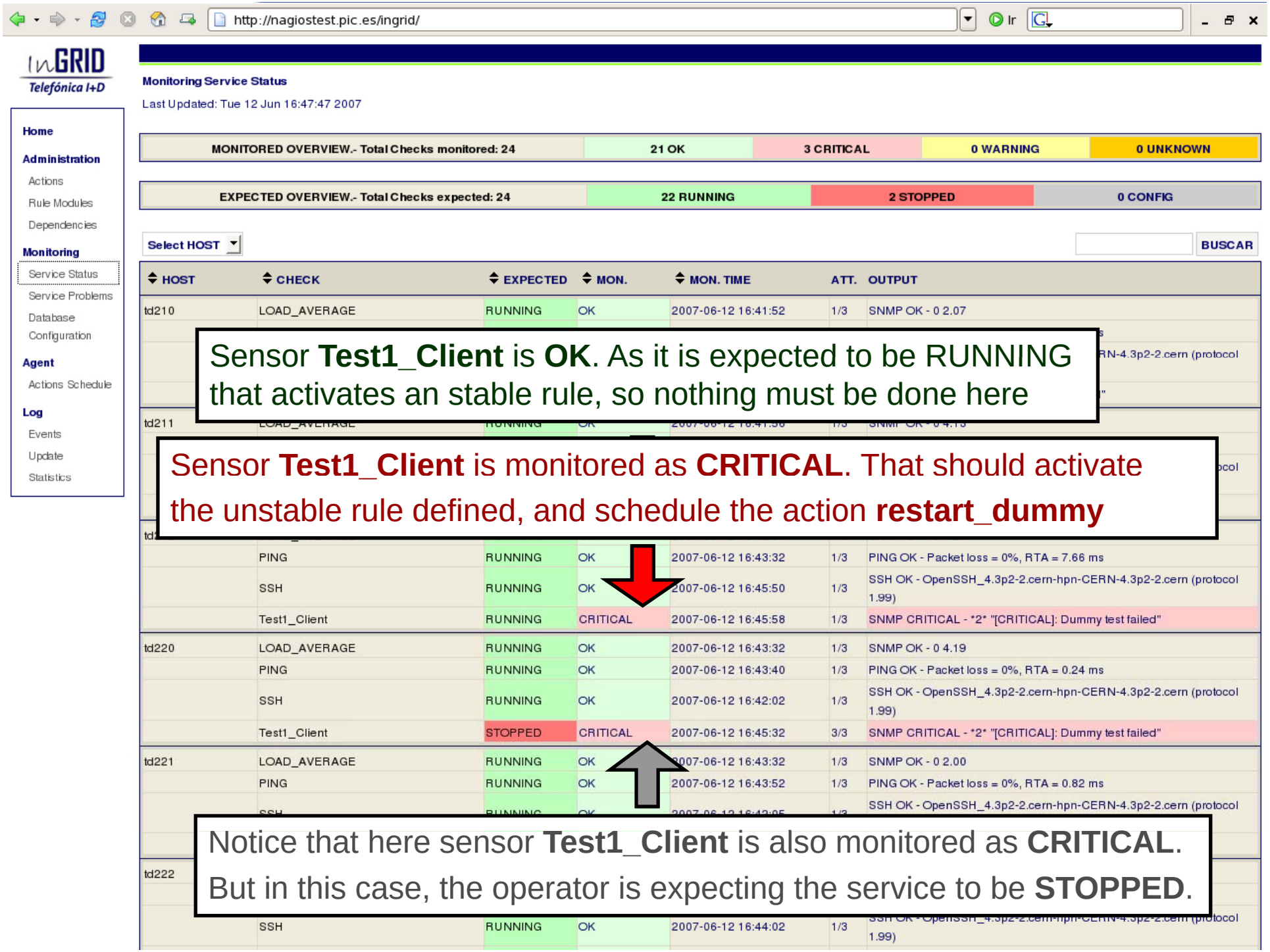
- INTRODUCTION
- DESIGN CONSIDERATIONS
- ARCHITECTURE
- DECISIONS MAKER
- **CASE OF USE**
- CONCLUSIONS



Module **TEST1_CLIENT** has only one action activated: **restart_dummy**

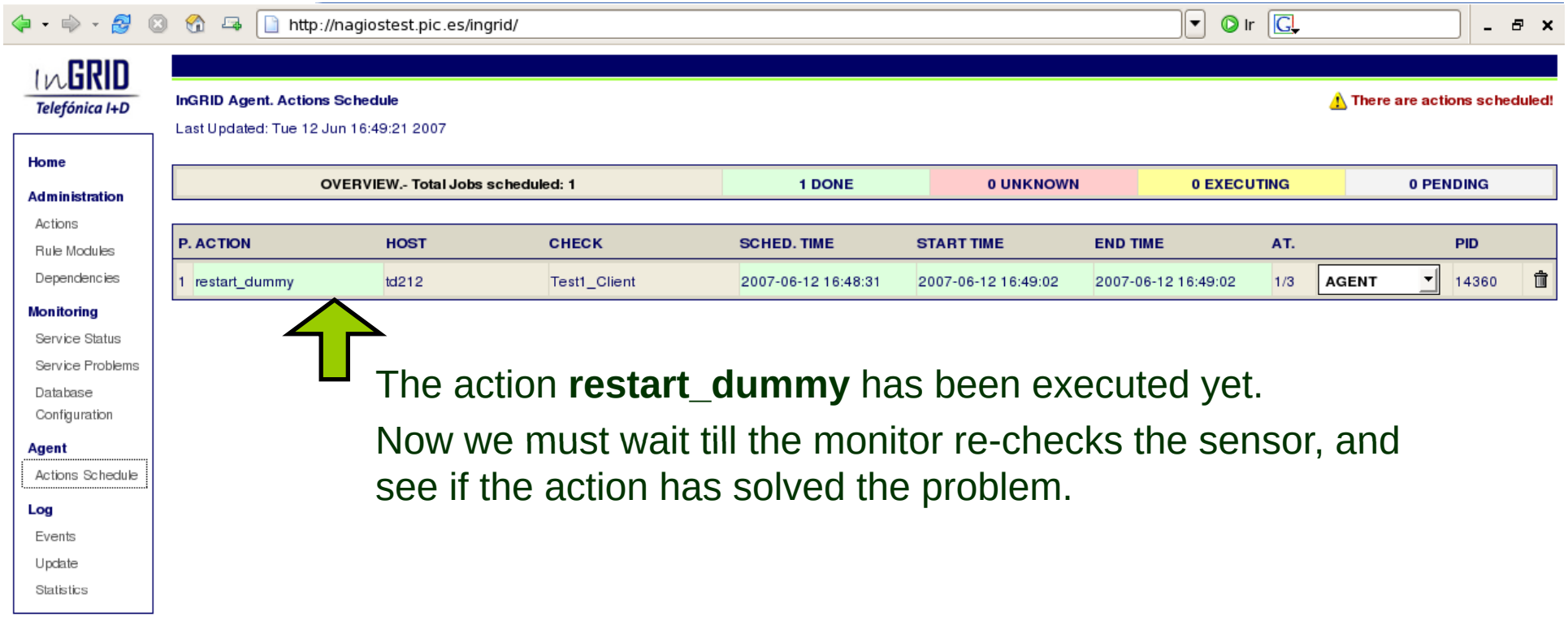


Module **TEST1_CLIENT** has 3 rules defined: 2 of them stable, and 1 rule unstable, which activates the action **restart_dummy**.


http://nagiostest.pic.es/ingrid/
InGRID
Telefónica I+D
Home
Administration
Actions
Rule Modules
Dependencies
Monitoring
Service Status
Service Problems
Database Configuration
Agent
Actions Schedule
Log
Events
Update
Statistics
Monitoring Service Status
Last Updated: Tue 12 Jun 16:47:47 2007
MONITORED OVERVIEW.- Total Checks monitored: 24
21 OK
3 CRITICAL
0 WARNING
0 UNKNOWN
EXPECTED OVERVIEW.- Total Checks expected: 24
22 RUNNING
2 STOPPED
0 CONFIG
Select HOST
BUSCAR
HOST
CHECK
EXPECTED
MON.
MON. TIME
ATT.
OUTPUT
td210
LOAD_AVERAGE
RUNNING
OK
2007-06-12 16:41:52
1/3
SNMP OK - 0 2.07
td211
PING
RUNNING
OK
2007-06-12 16:43:32
1/3
PING OK - Packet loss = 0%, RTA = 7.66 ms
SSH
RUNNING
OK
2007-06-12 16:45:50
1/3
SSH OK - OpenSSH_4.3p2-2.cern-hpn-CERN-4.3p2-2.cern (protocol 1.99)
Test1_Client
RUNNING
CRITICAL
2007-06-12 16:45:58
1/3
SNMP CRITICAL - *2* "[CRITICAL]: Dummy test failed"
td220
LOAD_AVERAGE
RUNNING
OK
2007-06-12 16:43:32
1/3
SNMP OK - 0 4.19
PING
RUNNING
OK
2007-06-12 16:43:40
1/3
PING OK - Packet loss = 0%, RTA = 0.24 ms
SSH
RUNNING
OK
2007-06-12 16:42:02
1/3
SSH OK - OpenSSH_4.3p2-2.cern-hpn-CERN-4.3p2-2.cern (protocol 1.99)
Test1_Client
STOPPED
CRITICAL
2007-06-12 16:45:32
3/3
SNMP CRITICAL - *2* "[CRITICAL]: Dummy test failed"
td221
LOAD_AVERAGE
RUNNING
OK
2007-06-12 16:43:32
1/3
SNMP OK - 0 2.00
PING
RUNNING
OK
2007-06-12 16:43:52
1/3
PING OK - Packet loss = 0%, RTA = 0.82 ms
SSH OK - OpenSSH_4.3p2-2.cern-hpn-CERN-4.3p2-2.cern (protocol
td222
SSH
RUNNING
OK
2007-06-12 16:44:02
1/3
1.99)
Sensor Test1_Client is OK. As it is expected to be RUNNING that activates an stable rule, so nothing must be done here
Sensor Test1_Client is monitored as CRITICAL. That should activate the unstable rule defined, and schedule the action restart_dummy
Notice that here sensor Test1_Client is also monitored as CRITICAL. But in this case, the operator is expecting the service to be STOPPED.

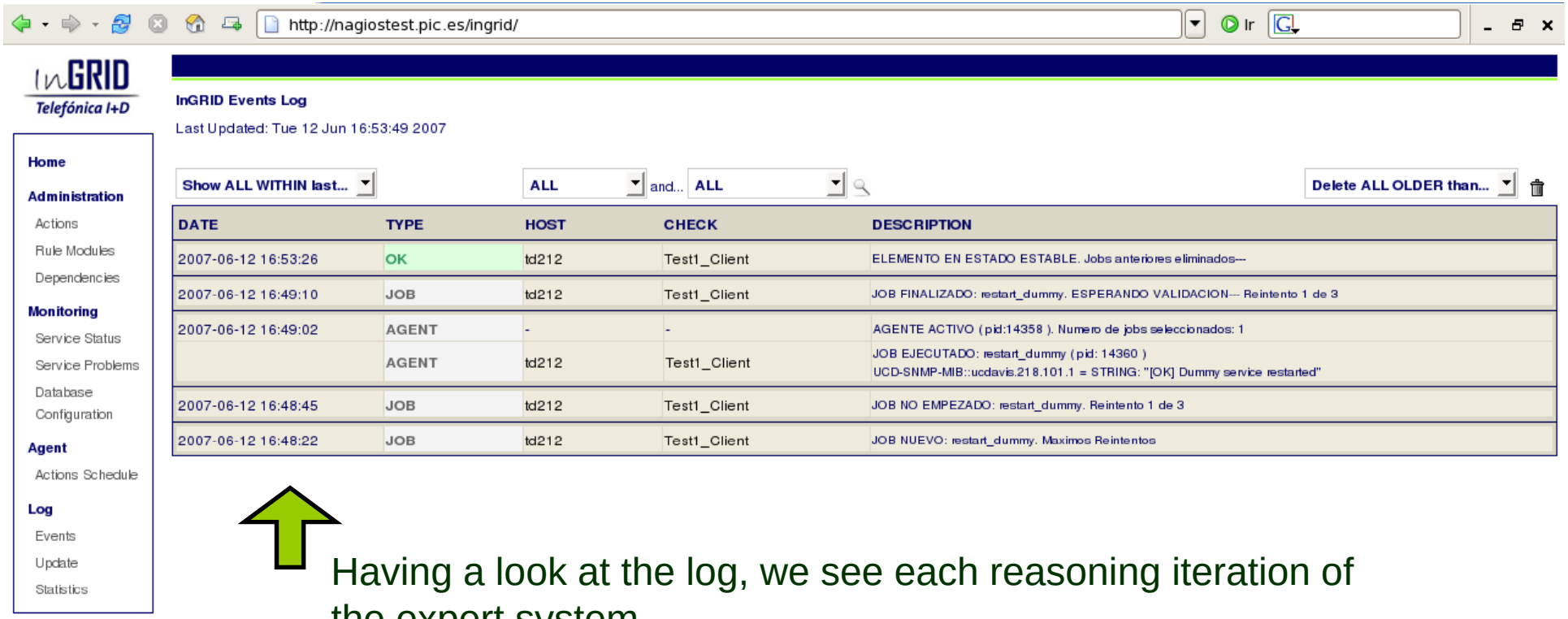
The rule for **Test1_Client** has been activated and the action **restart_dummy** has been scheduled.

Now we must wait till the Agent executes the action.



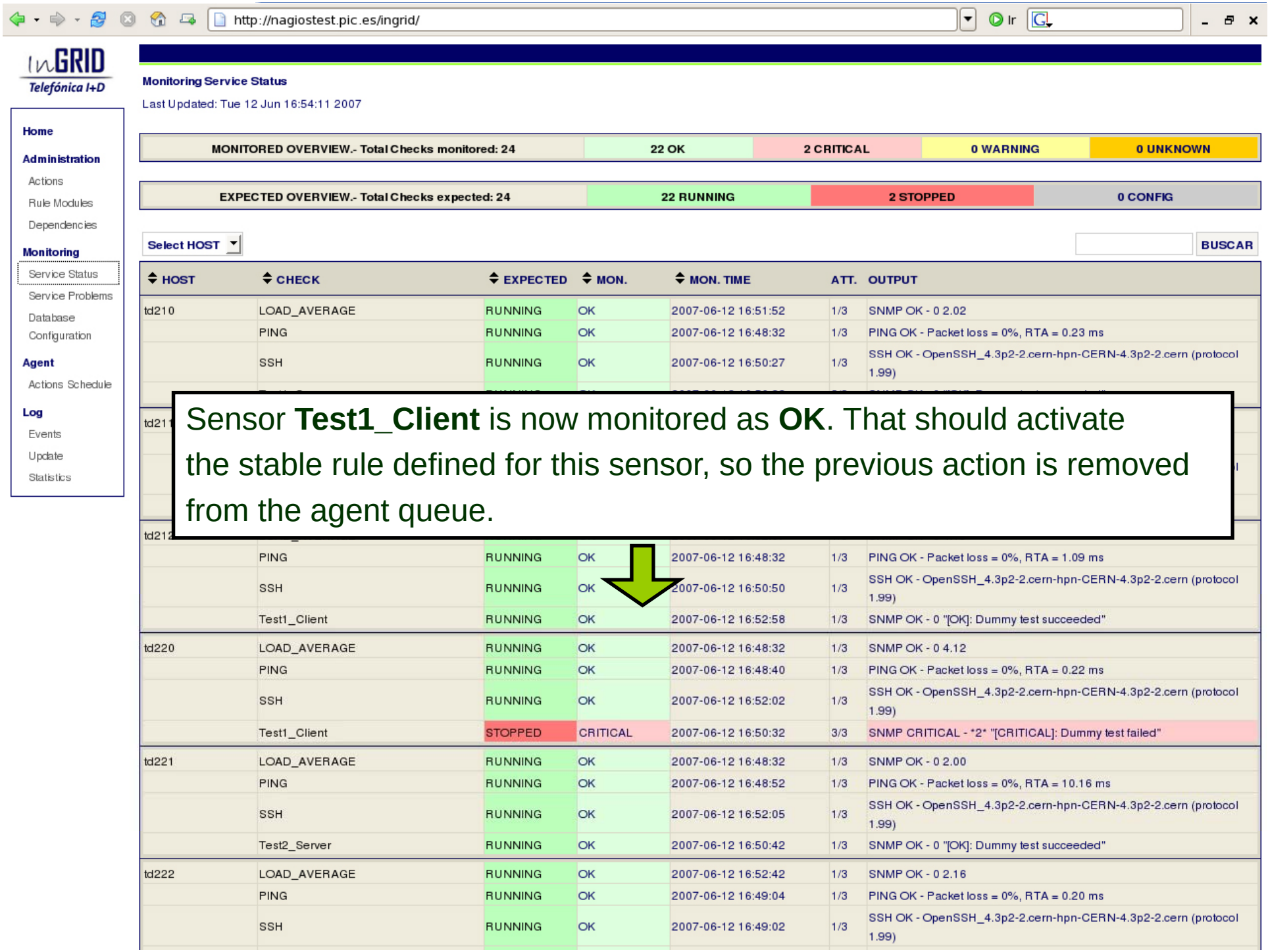
The action **restart_dummy** has been executed yet.

Now we must wait till the monitor re-checks the sensor, and see if the action has solved the problem.

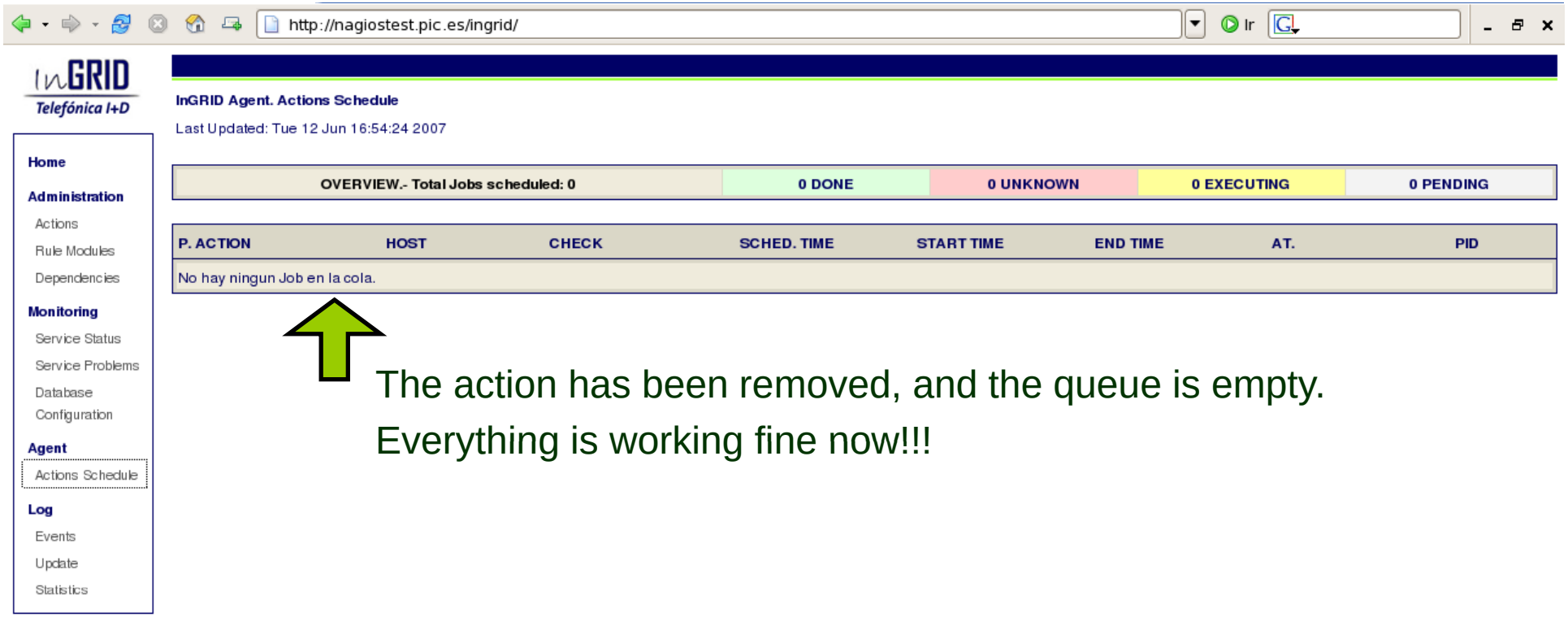


Having a look at the log, we see each reasoning iteration of the expert system.

First, the action **restart_dummy** is scheduled, the agent executes it successfully, and the monitor re-checks the sensor with an **OK** status.



Sensor **Test1_Client** is now monitored as **OK**. That should activate the stable rule defined for this sensor, so the previous action is removed from the agent queue.


http://nagiostest.pic.es/ingrid/
Ir
InGRID
Telefónica I+D
Home
Administration
Actions
Rule Modules
Dependencies
Monitoring
Service Status
Service Problems
Database Configuration
Agent
Actions Schedule
Log
Events
Update
Statistics
InGRID Agent. Actions Schedule
Last Updated: Tue 12 Jun 16:54:24 2007
OVERVIEW.- Total Jobs scheduled: 0
0 DONE
0 UNKNOWN
0 EXECUTING
0 PENDING
P. ACTION
HOST
CHECK
SCHED. TIME
START TIME
END TIME
AT.
PID
No hay ningun Job en la cola.
The action has been removed, and the queue is empty.
Everything is working fine now!!!

CONCLUSIONS

- INTRODUCTION
- DESIGN CONSIDERATIONS
- ARCHITECTURE
- DECISIONS MAKER
- CASE OF USE
- **CONCLUSIONS**

CONCLUSIONS

- **The main objective of InGRID is:**
  - to minimize the large amount of incidents that the operator must attend, and
  - to try to restore automatically the major number of services without the intervention of the operator.

- **InGRID is being tested over some production services of PIC (South-West ROC).**
- **We use PIC feedback to improve InGRID with new features.**

PIC
port d'informació
científica

Telefónica
TELEFÓNICA
INVESTIGACIÓN Y DESARROLLO

CONCLUSIONS

- **During first tests, we detected some outage problems:**
  - Some of the incidences occurred were false errors caused by a monitoring or executing problems (SNMP not responding).
  - Also a large number of incidences were caused by a huge load average of the host monitored, which became into timeouts.
  - During high loads, a host usually returns critical status for all checks.
  - We have tried to minimize these problems monitoring also SNMP and the Load Average of each host, and using dependencies.
- **InGRID demonstrates that may detect the real cause of the problem when:**
  - service dependencies are fine tuned, and
  - the adequate sensors and rules are configured.

CONCLUSIONS

- **When an incidence is caused by a standalone service, independent of any other element, InGRID has demonstrated a great performance when acting on it.**
  - Usually, the only necessary action to do is to restart the service.
- **Standalone services may run in a large number of hosts.**
  - To manage them, the administrator just needs to create only one rule for that generic service.
  - But, if one wants to make distinct rules according to whether the service is running in a particular host, or a group of hosts, or even a site, InGRID permits to associate a rule to these kinds of groups.

QUESTIONS?

Thanks for your attention!!

Francesc Pérez

franp@tid.es

Telefónica
Telefónica Investigación y Desarrollo