



# WP4 Fault Tolerance Task

Software overview



# Objective

- Keep a node in a cluster up and running in case of a fault
- Minimise the need for human intervention
- Simplify the life of an administrator of a large farm of nodes



# Overview

- Fault Tolerance by software
- No specialised hardware used
- Automatically detect and correct errors
- Rule based approach
- Math. Expression used to formulate error/fault conditions



# Software structure

- Three main components of EDG FT daemon:
  - Sensors
  - Decision unit / correlation engine
  - Actuators
- Input data provided by WP4 monitoring system



# Internals

- Arbitrarily complex expressions of various types (integer, boolean) may be formulated
- Boolean expressions used to specify error conditions
- Should an error condition occur a corresponding „actuator“ is launched
- Input data is received from the WP4 monitoring system



# Implementation

- Expression trees are implemented by a templated class hierarchy written in C++
- Data types currently supported: boolean, integer and ANSI C strings. This can be easily extended to arbitrarily complex types (e.g. structs)
- Interface to UNIX OS written in plain C, but mainly accessed through class abstractions
- Repository API of the WP4 Monitoring system used to get input data from the Monitoring system



# Sensors

- „Sensors“ are an abstraction of a data source
- In principle they can provide data of any type (e.g. complex structs, classes, etc...)
- Currently integer, ANSI C string and boolean are implemented
- The software can easily be extended to support further types



# „Virtual“ sensors

- „Virtual sensors“ are an abstraction which provides combined or correlated data from various simple sensors





# Actuators

- Actuators are an abstraction which provide a means to execute a command on a node
- An actuator should perform a repair task, e.g. restarting a daemon or clearing a directory
- Usually, the actuators are system binaries or scripts