(My) Vision of where we are going

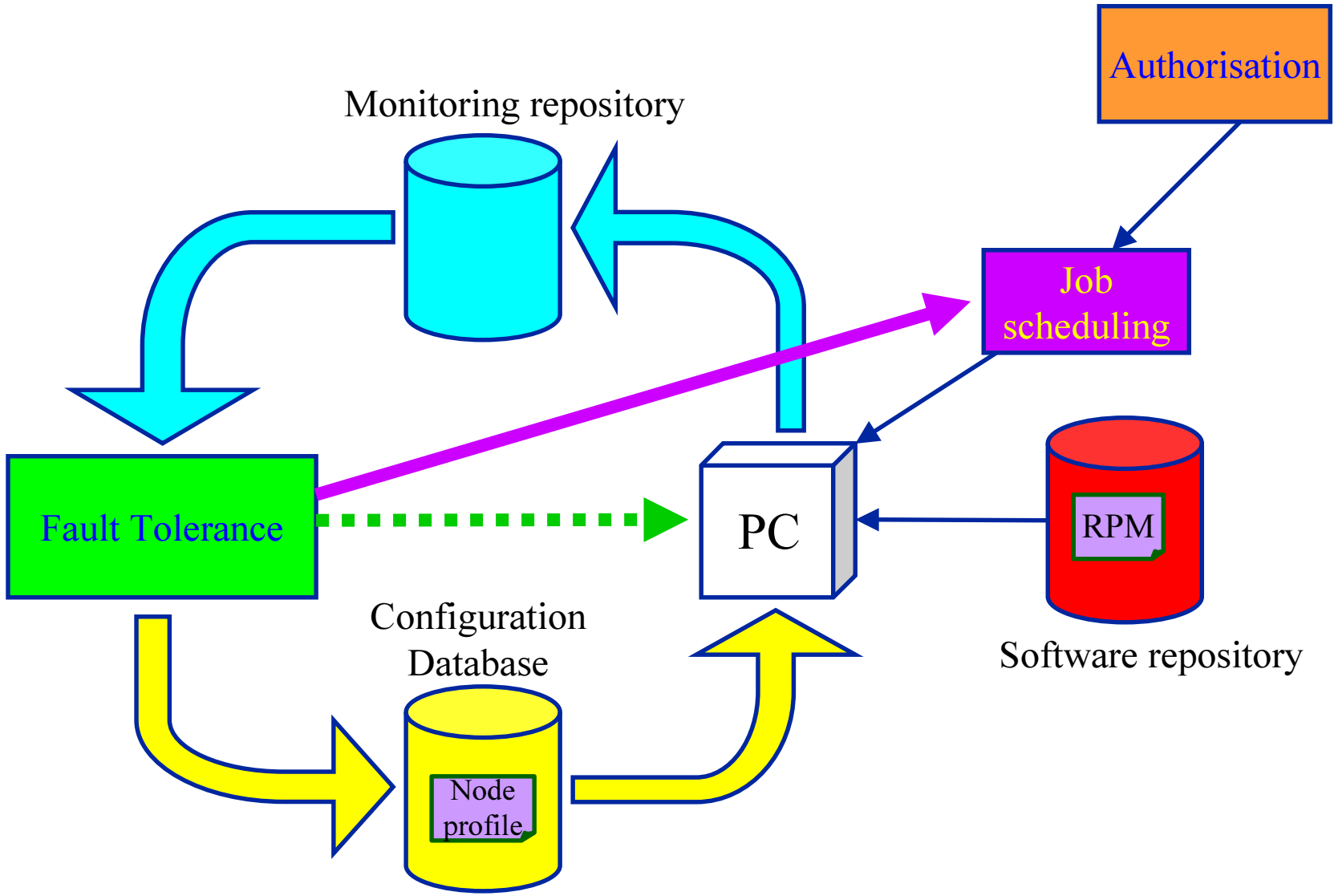
WP4 workshop, 10/12/2002

Olof Bärring

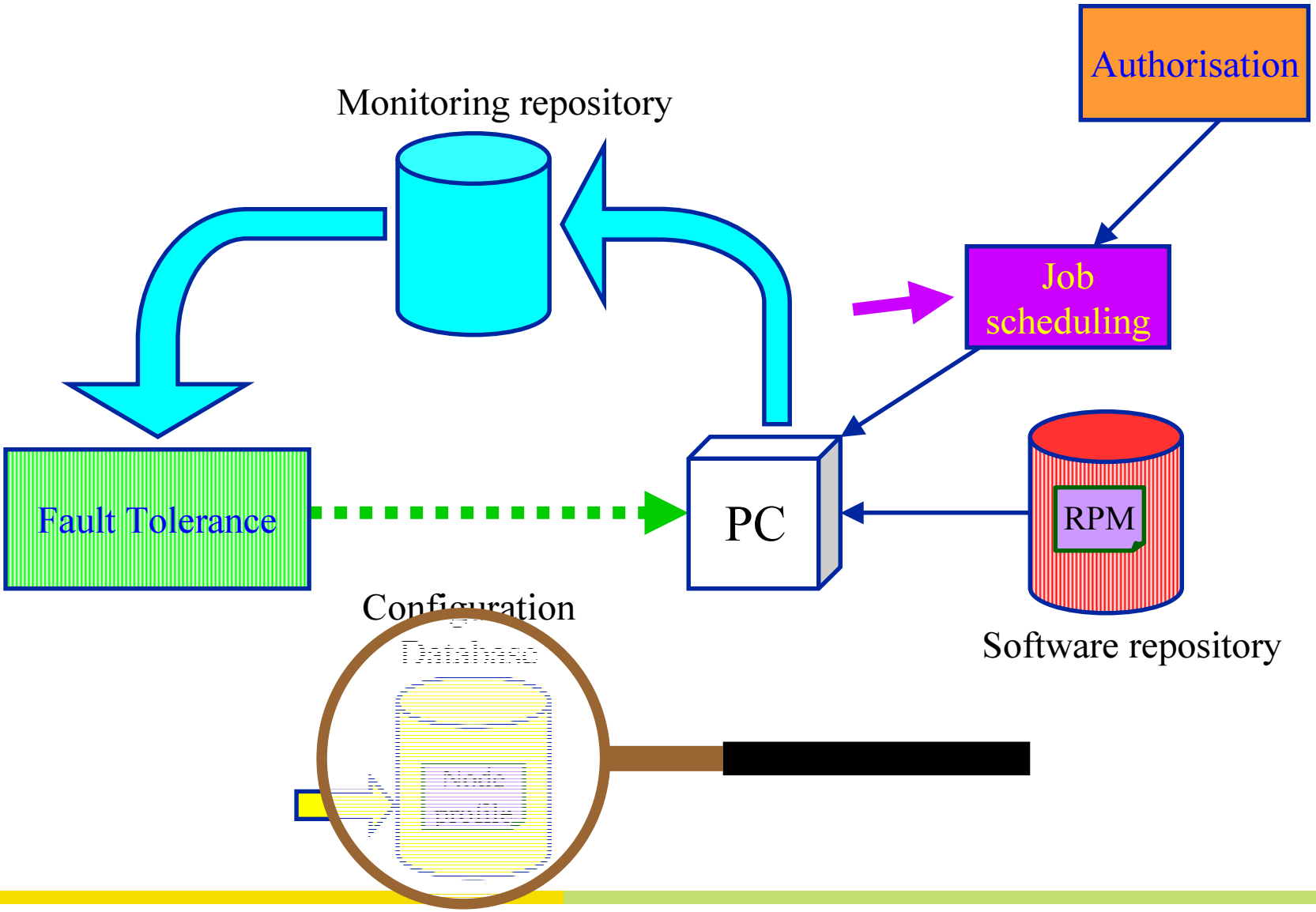
Outline

- Fabric automation architecture
- Today and tomorrow subsystem by subsystem
- Important steps towards tomorrow
- How to use this workshop

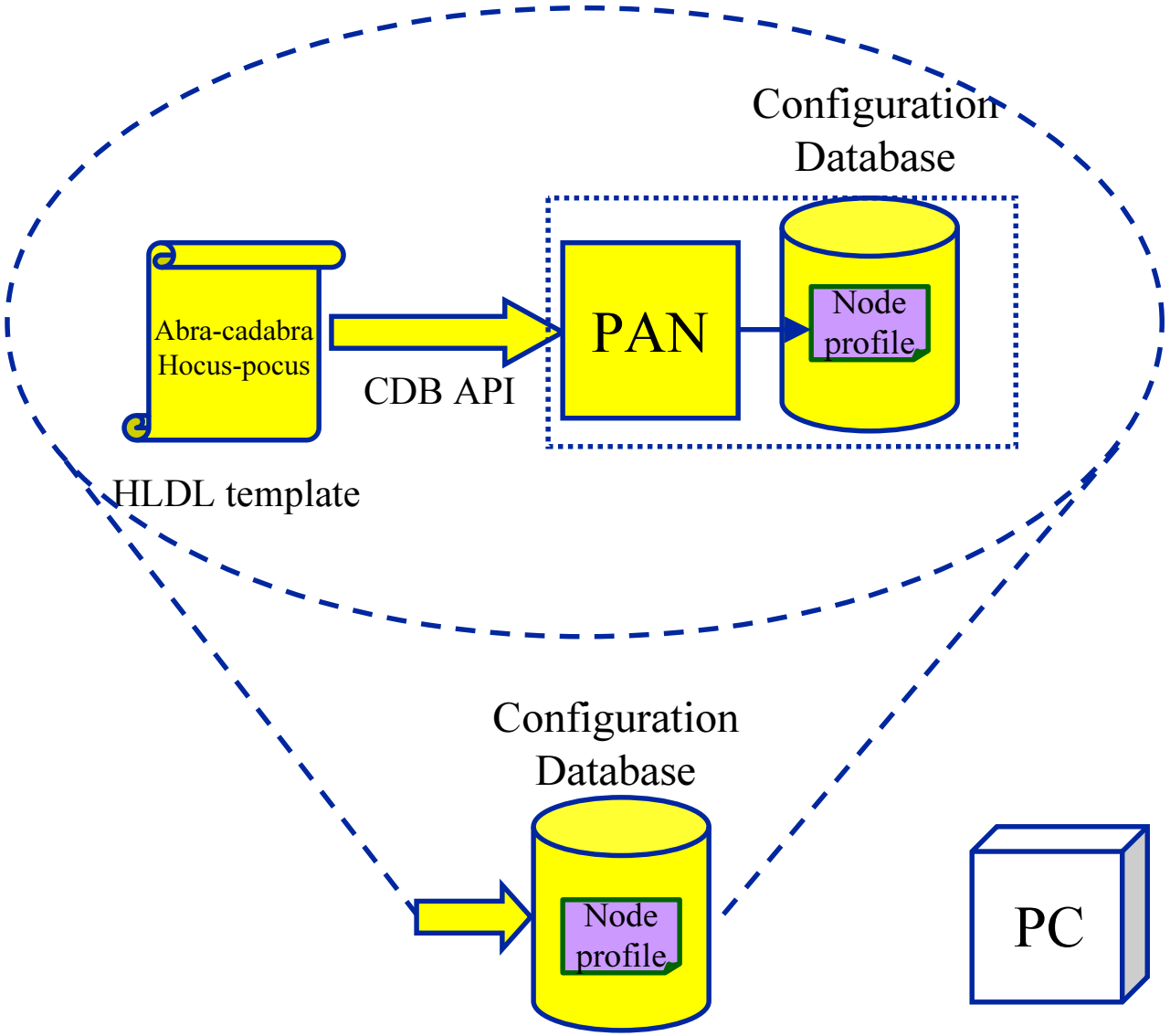
Automation Architecture



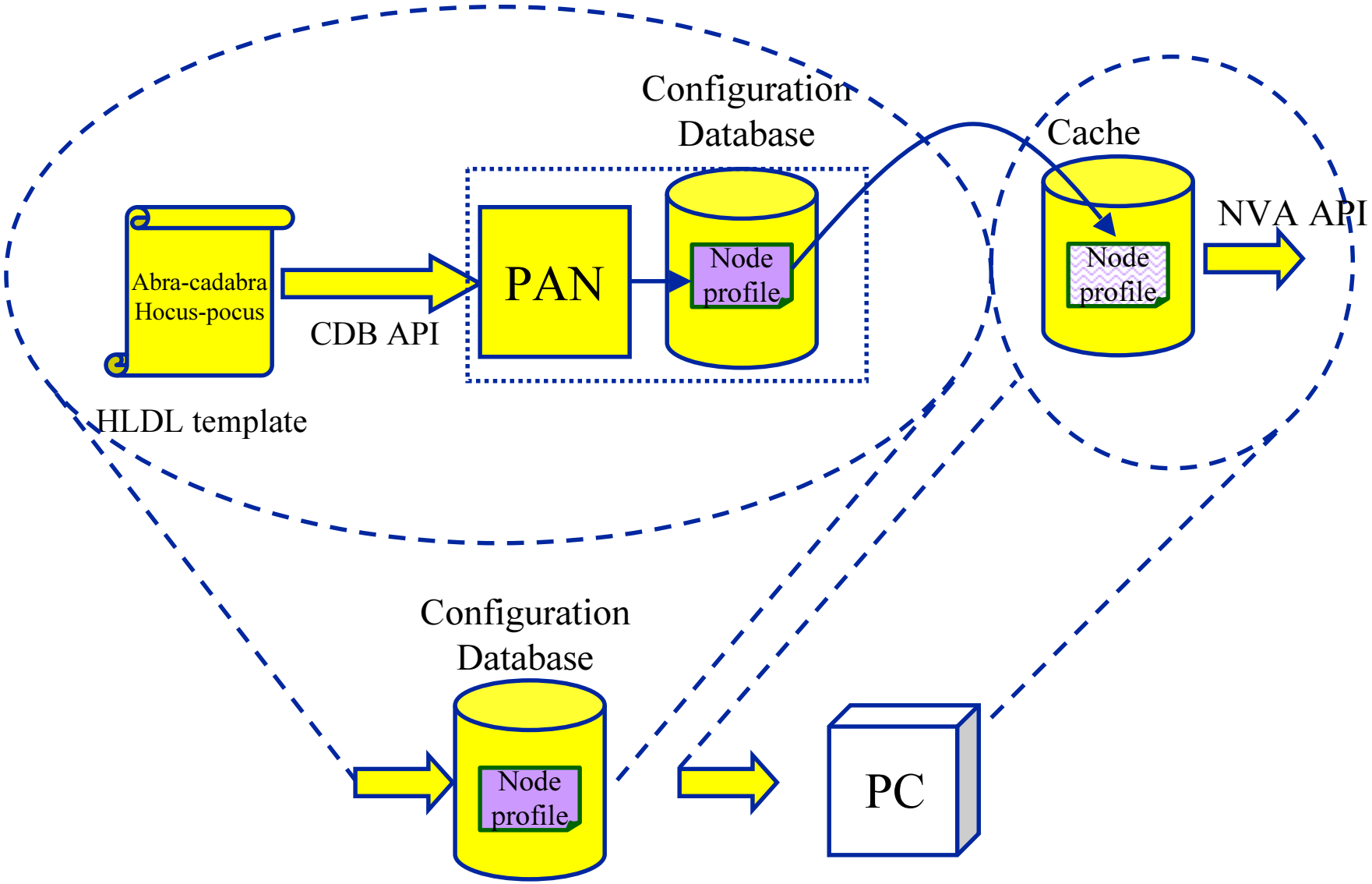
Today

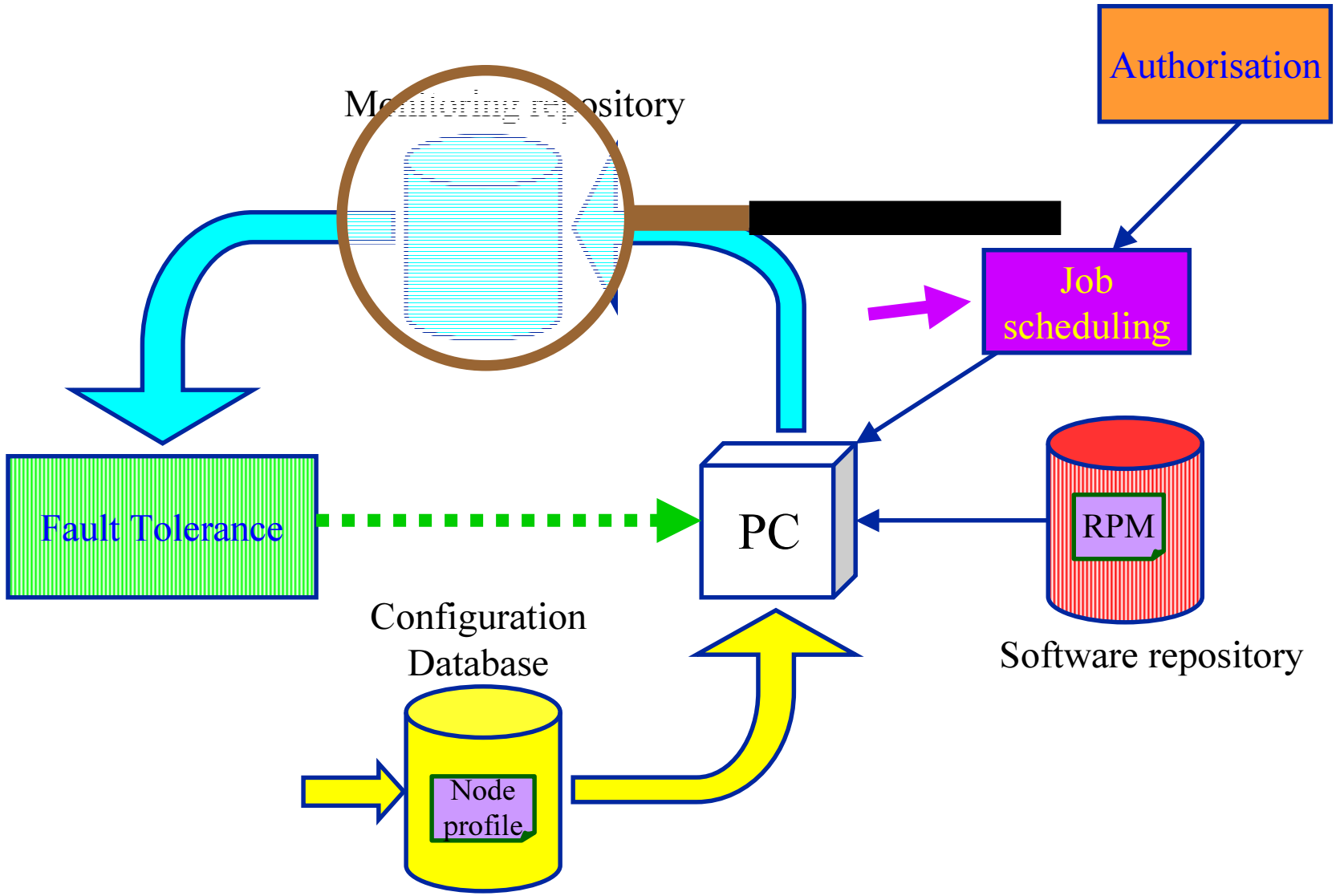

Authorisation
Monitoring repository
Job scheduling
Fault Tolerance
PC
RPM
Configuration
Software repository

Configuration: Today

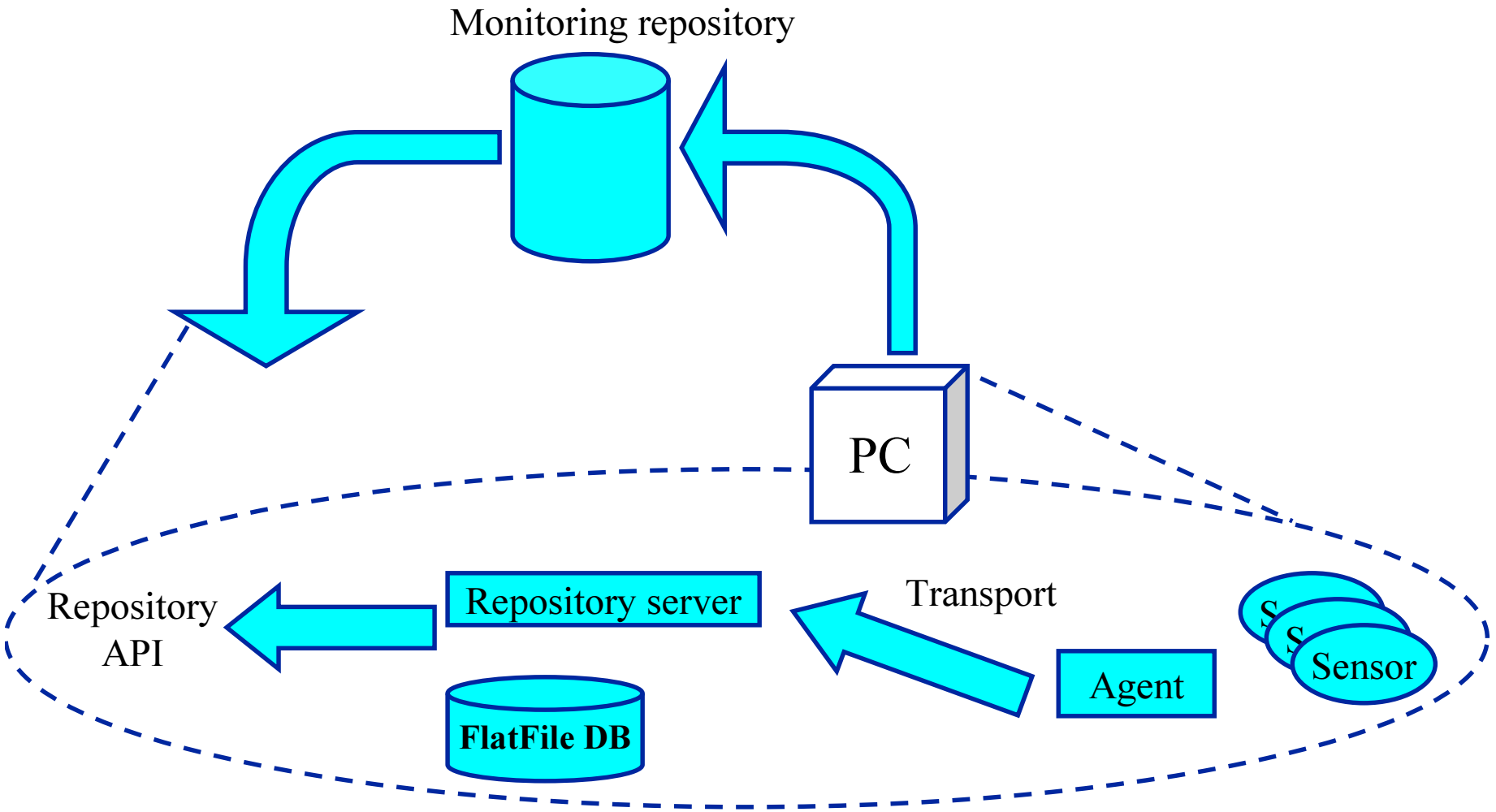


Configuration: … and tomorrow (February)

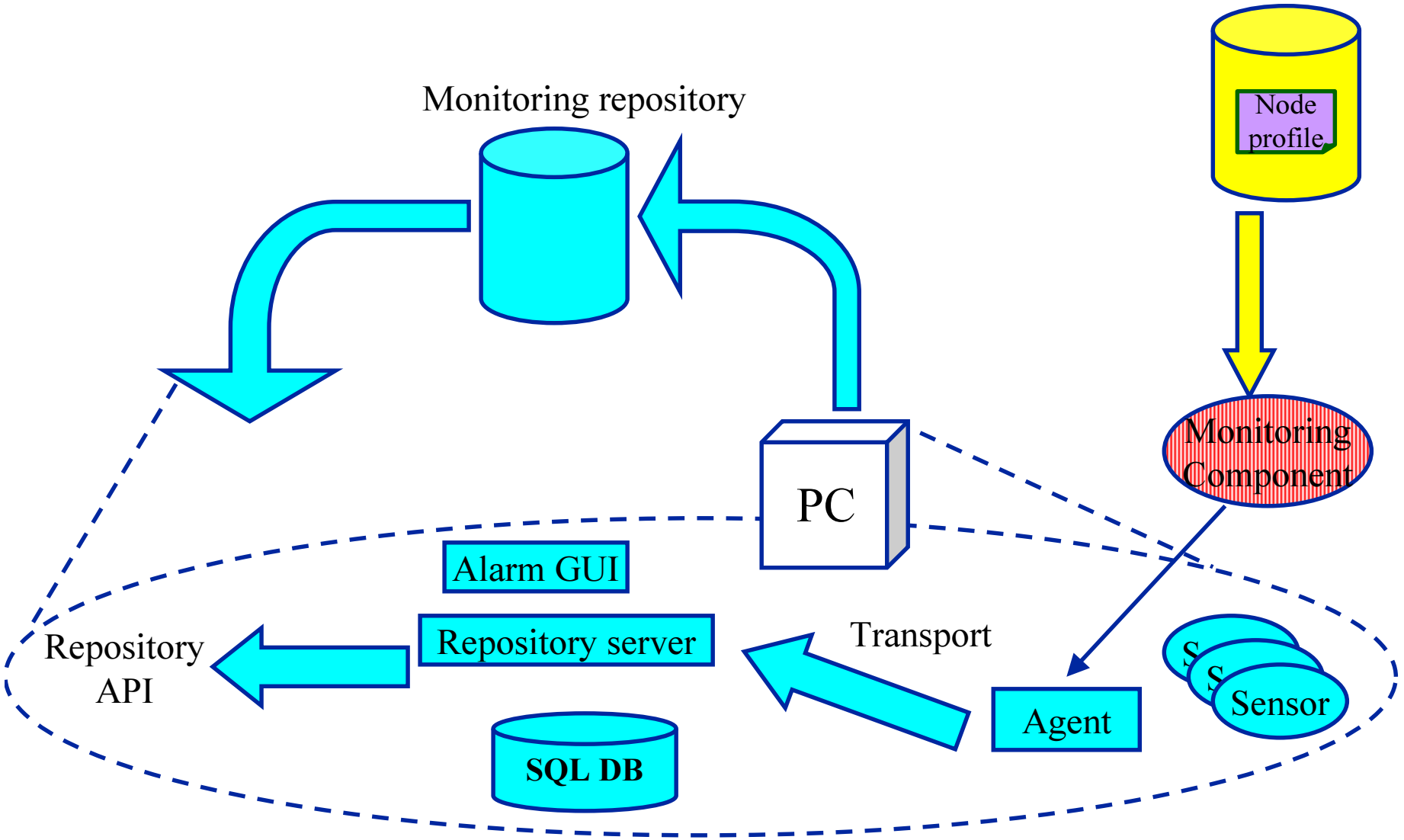


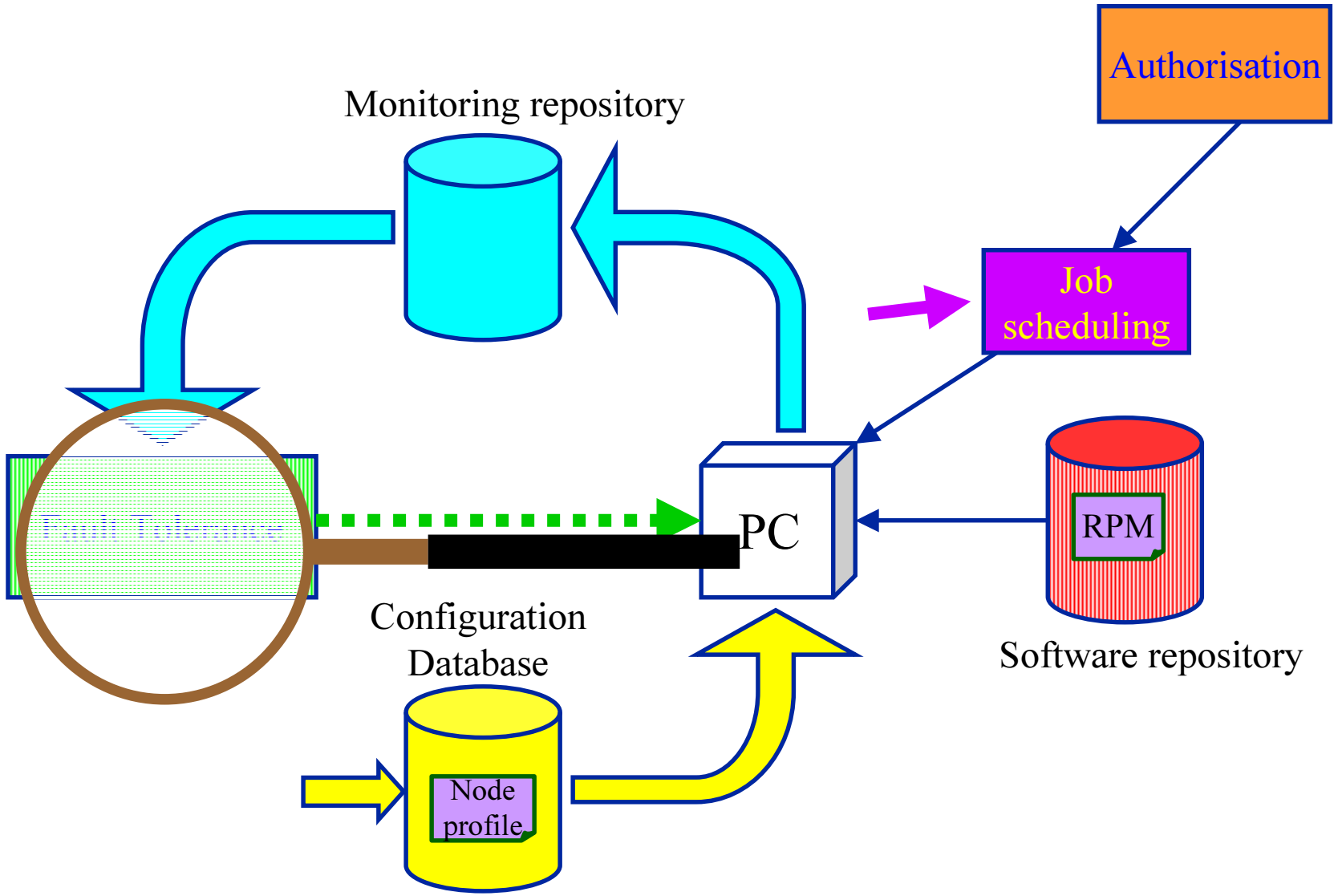

Authorisation
Job scheduling
Fault Tolerance
PC
RPM
Software repository
Configuration Database
Node profile

Monitoring: Today

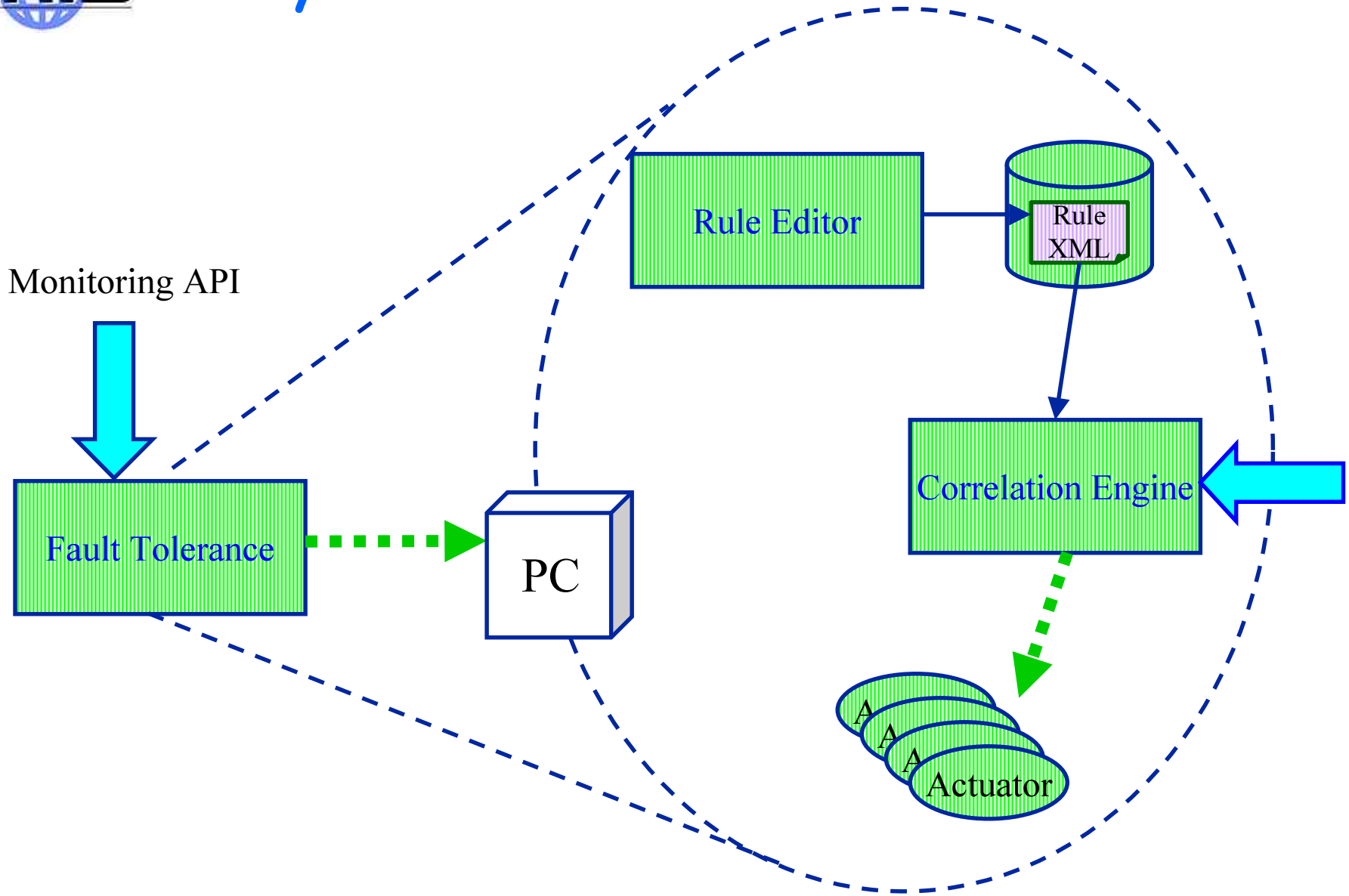


Monitoring:
… and tomorrow (March)

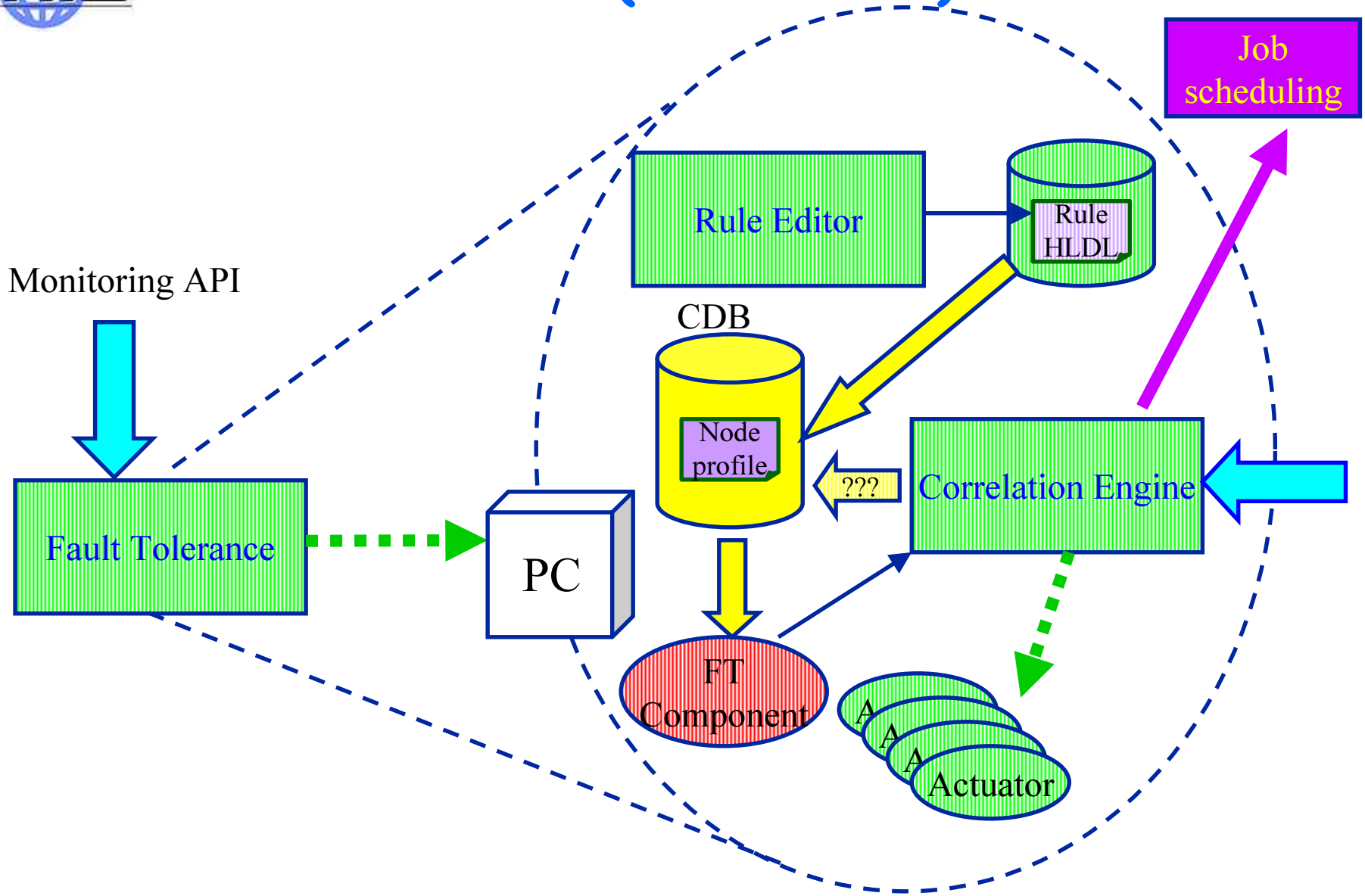


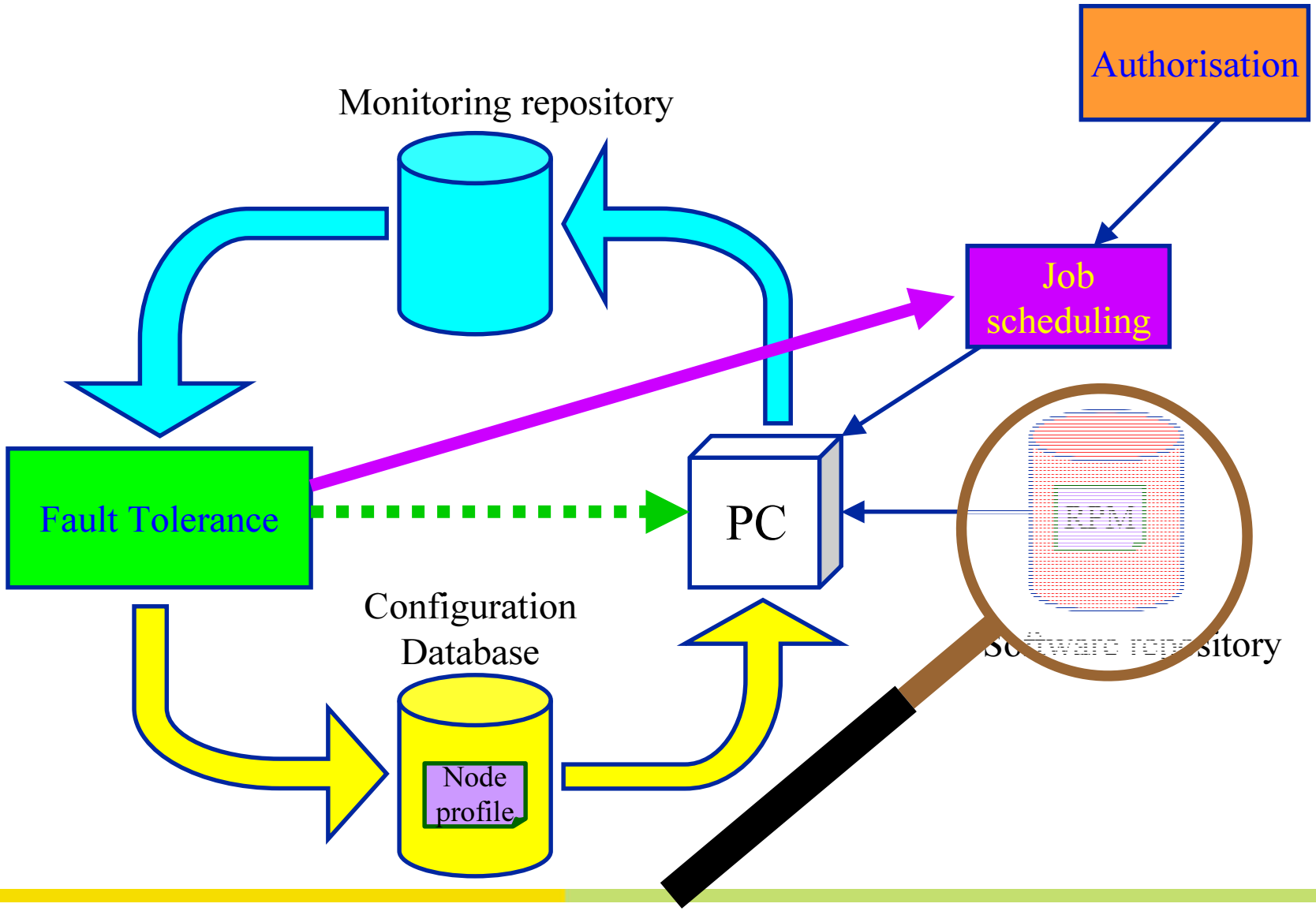

Authorisation
Monitoring repository
Job scheduling
Fabric Fault Tolerance
PC
RPM
Configuration Database
Node profile
Software repository

Fault tolerance: Today

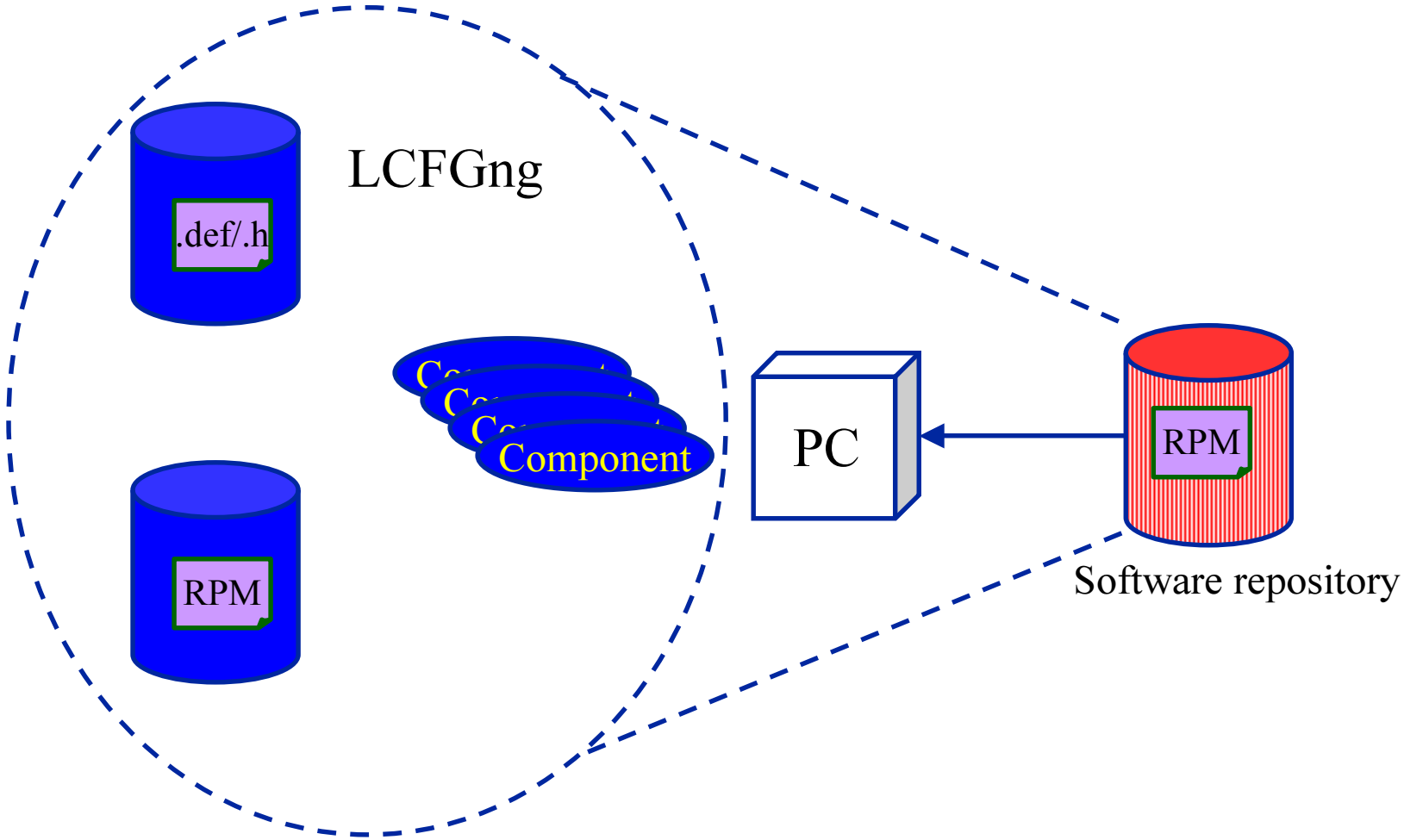


Fault tolerance:
… and tomorrow (summer???)

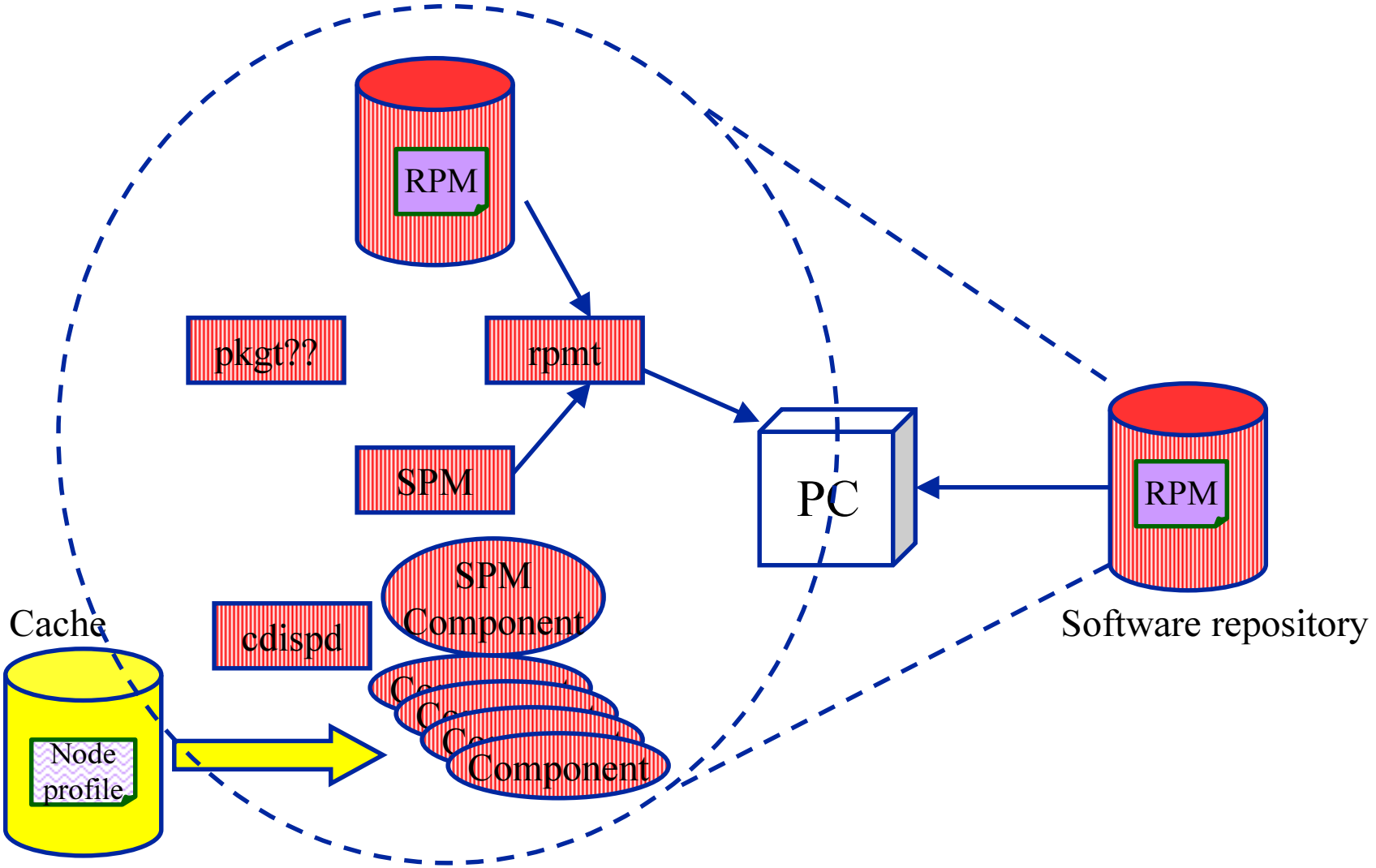


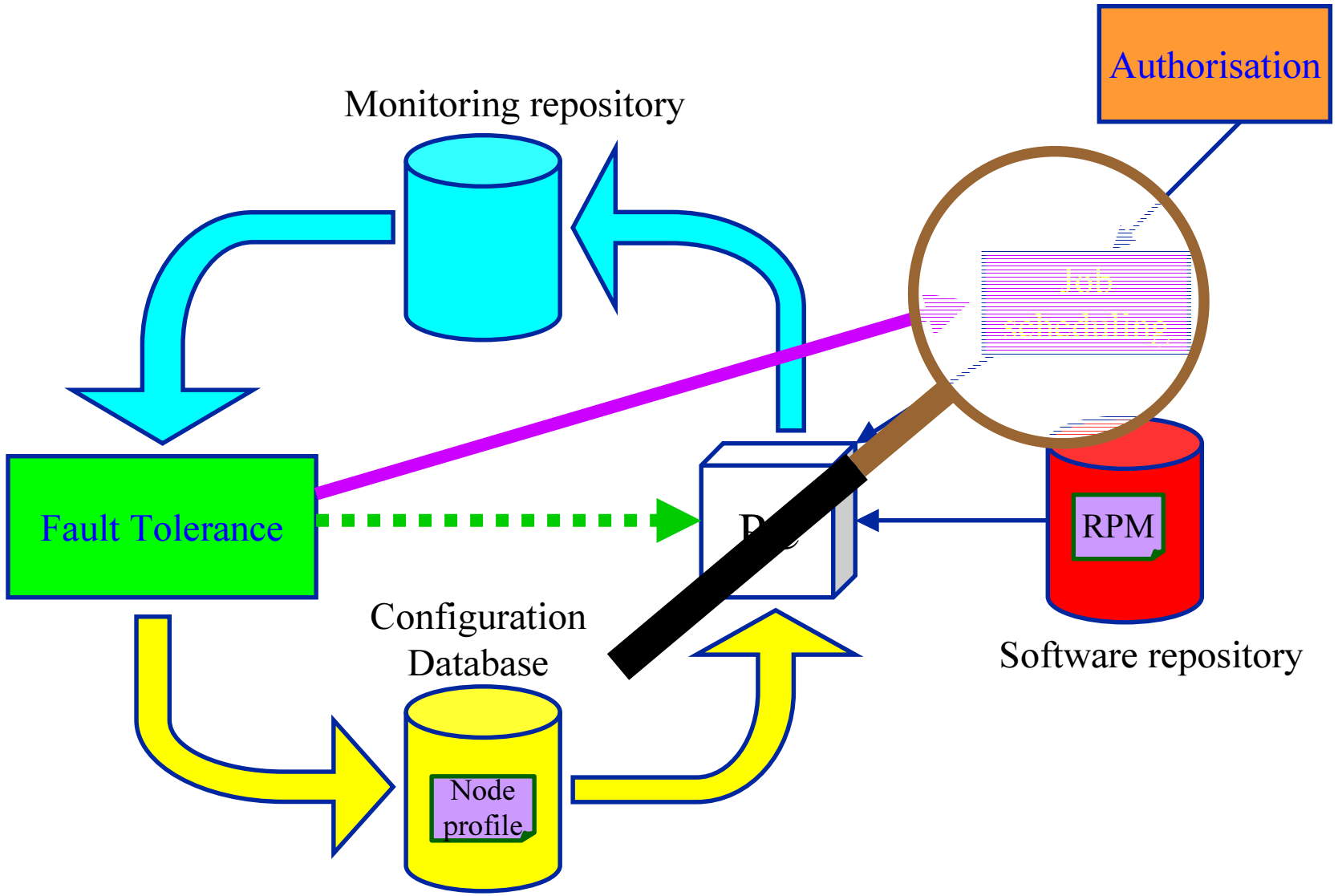

Authorisation
Monitoring repository
Job scheduling
Fault Tolerance
PC
RPM
Configuration Database
Node profile

Installation&Maintenance: Today

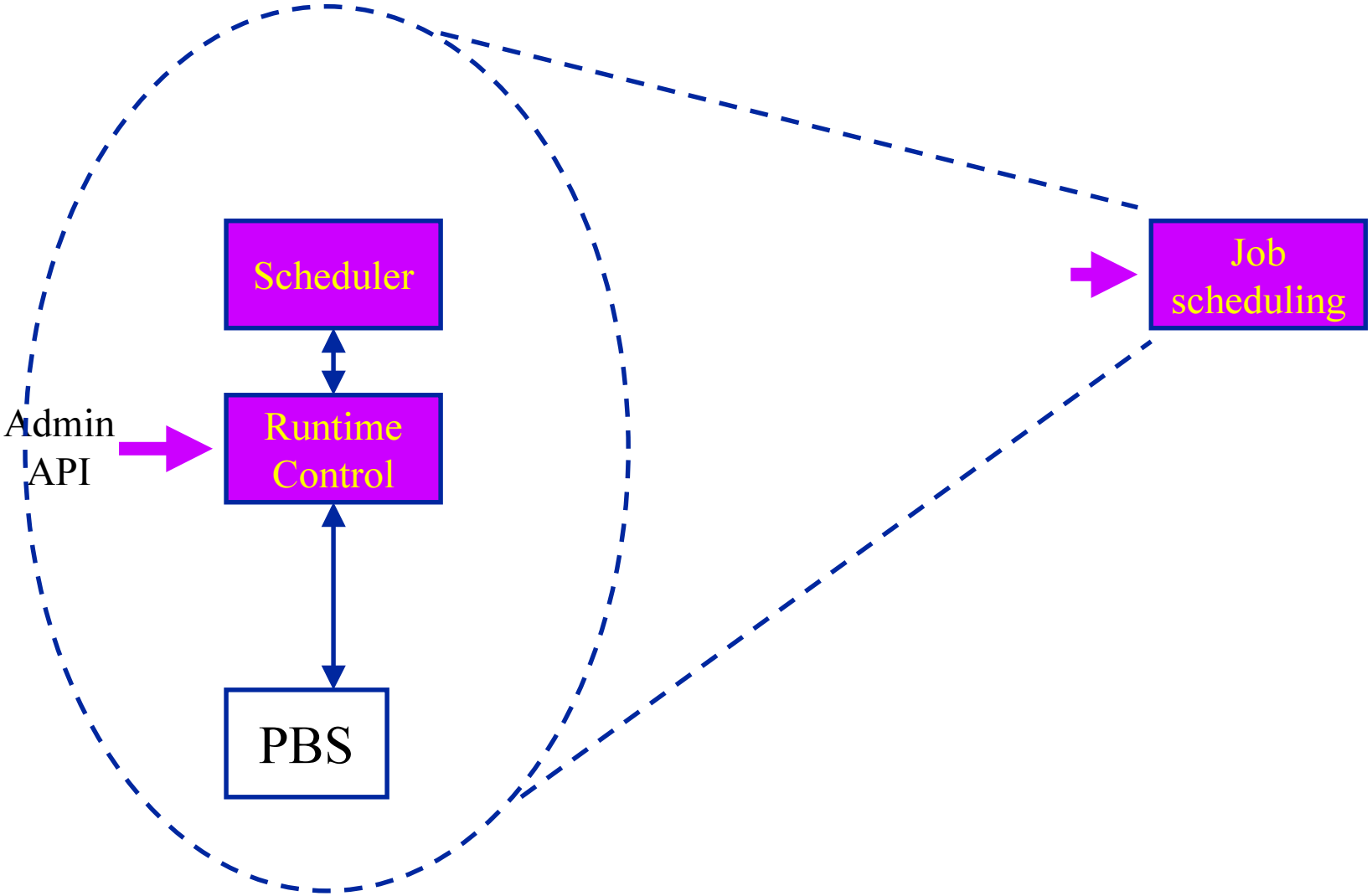


Installation&Maintenance: … and tomorrow (April)

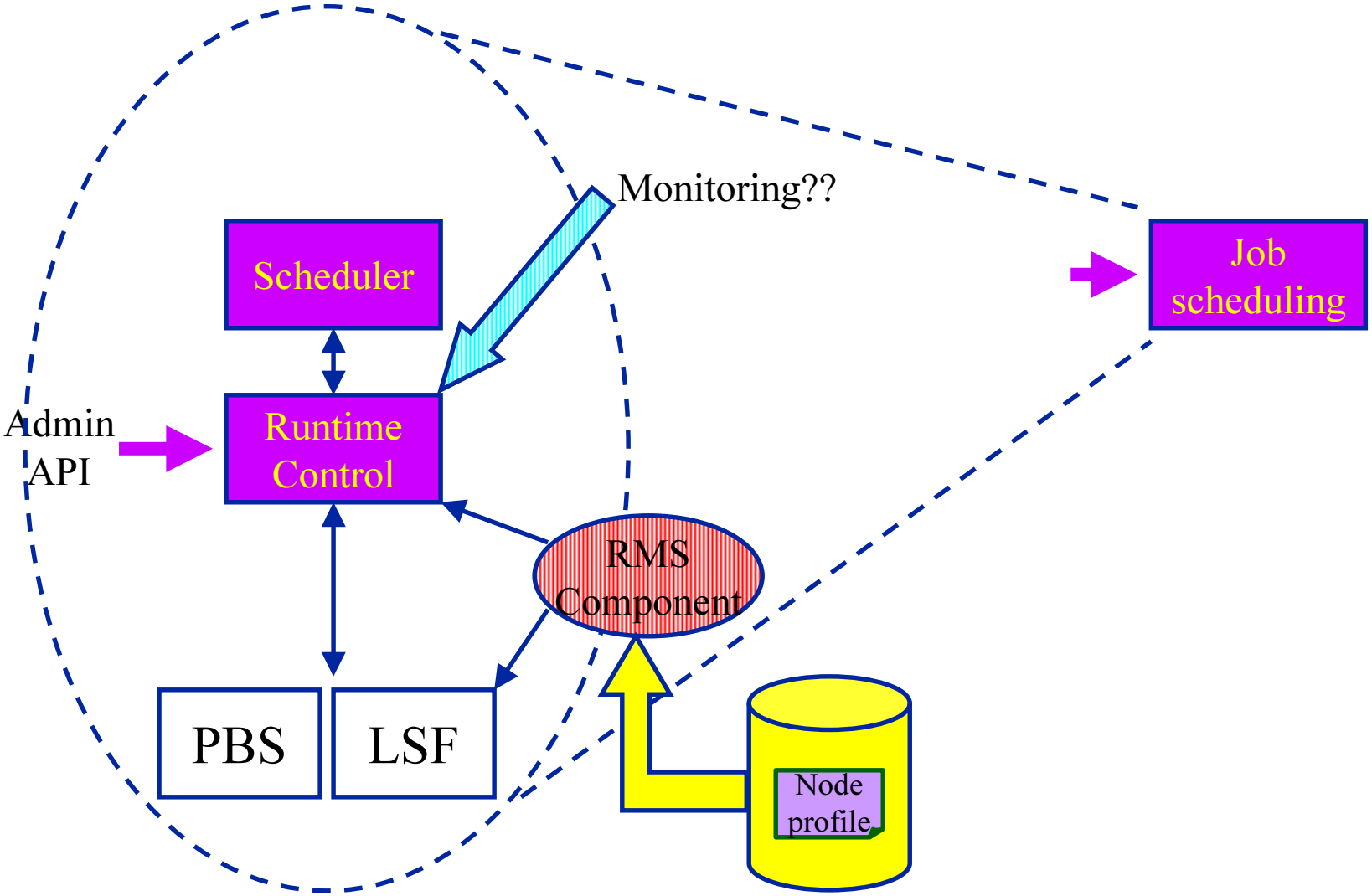



Authorisation
Monitoring repository
Fault Tolerance
RPM
Configuration
Database
Node
profile
Software repository

Resource mgmt Today



Resource mgmt
… and tomorrow (spring??)



Important steps towards tomorrow

- WP4 subsystems need to be
  - Configured
  - Monitored
  - Repaired

Important steps towards tomorrow (1)

- WP4 subsystem needs to be configured
  - Identify what potentially needs to be (re)configured
  - Define your configuration parameters and how they should fit into the global schema
  - Write HLDL templates for your subsystem (learn from exercises this week)
  - Write "configuration components" that calls the NVA API and generates configuration files for your services

Important steps towards tomorrow (2)

- WP4 subsystem needs to be monitored
  - Identify what can go wrong → define the subsystem health metrics
  - Implement a sensor that measures the health metrics
  - Configure the monitoring subsystem to become aware of your sensor/metrics. How is this done best?
    - Keep the configuration together with your subsystem and "link" it to the monitoring?
    - Or should the configuration be added directly to the monitoring HLDL template?

Important steps towards tomorrow (3)

- WP4 subsystem needs to recover from unhealthy states
  - Determine how to recover the subsystem from the identified set of unhealthy states
  - Implement actuator scripts that performs the repairs
  - Define the rule that links your health metrics to your recovery actuator

Important steps towards tomorrow (3)

- WP4 subsystem needs to recover from unhealthy states
  - Determine how to recover the subsystem from the identified set of unhealthy states
  - Implement actuator scripts that performs the repairs
  - Define the rule that links your health metrics to your recovery actuator

Open issues (to think over during the exercises)

- Who launches the recovery actuator scripts? I see two cases:
  - Repair that do not involve a configuration change, e.g. restart daemon; clean /tmp;
  - Repair that do involve a configuration change, e.g. service client reconfiguration when a central service falls over
- Desired state versus actual state duality: how to enter a reference to the desired state in a FT rule?
  - FT rules naturally reference actual state through monitoring metrics. How can the same be done for the desired state?

How to use this workshop

- The objective for this workshop to facilitate the integration of WP4 subsystems. It is *not* for you to show how nice your software is written and how well your task has been working, so
  - Teachers: please make sure to focus your exercises on important interfaces and how to use them. Leave out cool features unless the time allows for it
  - Participants: use the exercises to understand how to interface your subsystem. Identify potential problems with the interfaces from your subsystem point of view. Save those issues for the Friday brainstorming session
- In the brainstorm in Friday: discuss possible open issues or problems identified during the hands-on exercises.