University Autonoma of Barcelona (**UAB**)

- Established in 1968

- 40.000 Students (8.000 graduate students)

- 3.000 Teachers (Professors, Associate Professors and Assistant Professors)

- 2.500 Students in Computer Science

CAOS Department

- 3 Professors
- 10 Associate Professors
- 1 Post Doc
- 23 Research Assistant (Graduate Students)

More Information: http://www.caos.uab.es

CAOS Research Activities

Parallel Applications

Support Tools for Parallel Programming

Management of Parallel and Distributed Systems

Computer Architecture: Processors and Networks

- Advanced Processors Design (Branch Prediction and Adaptive Cache)
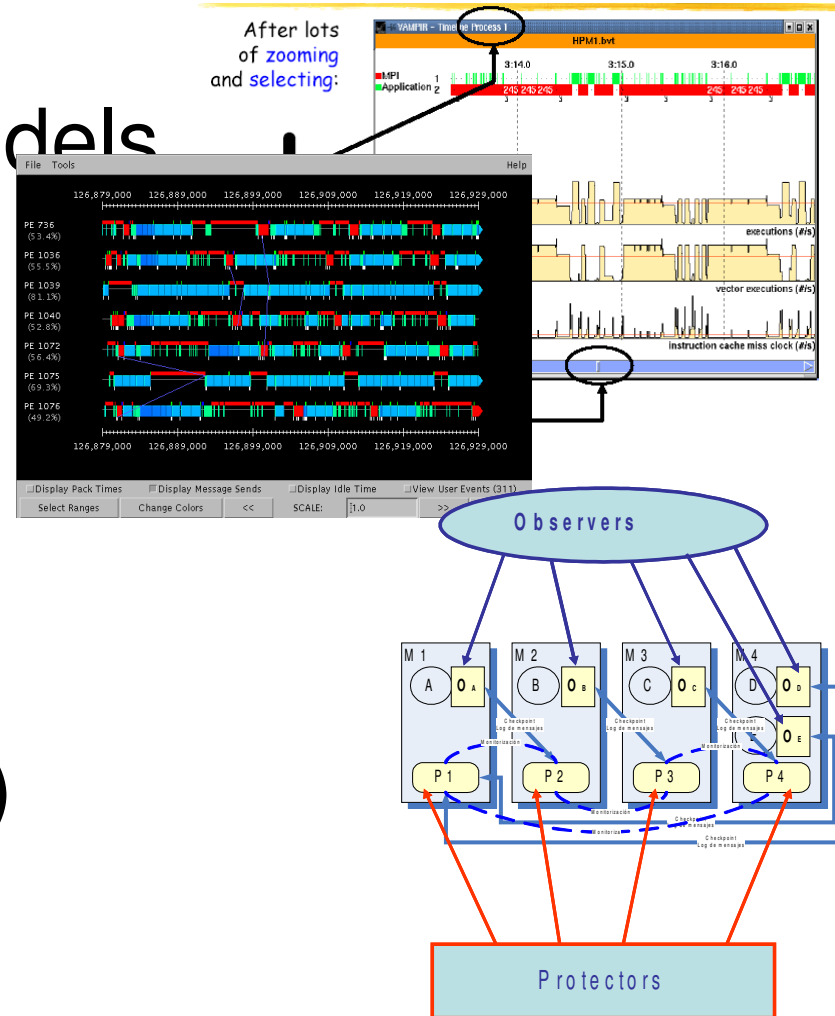
- Interconnection Networks
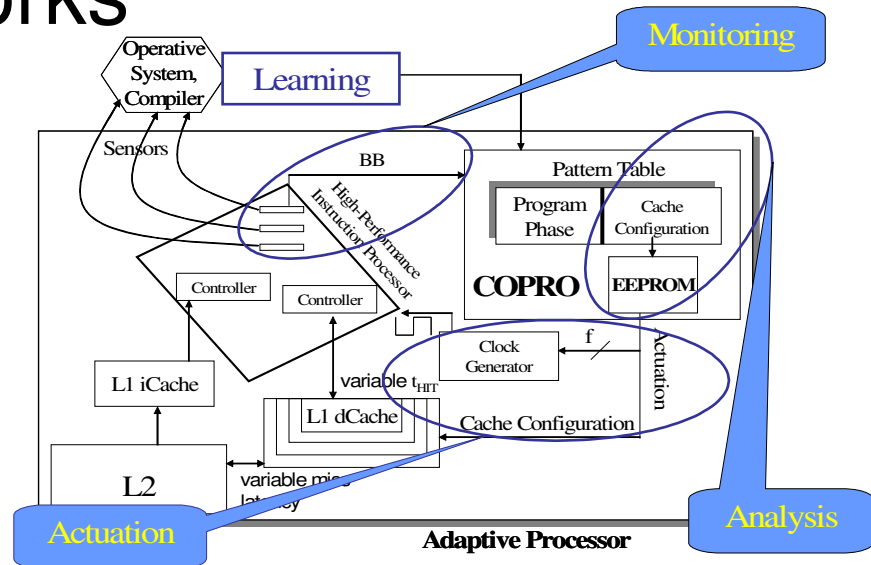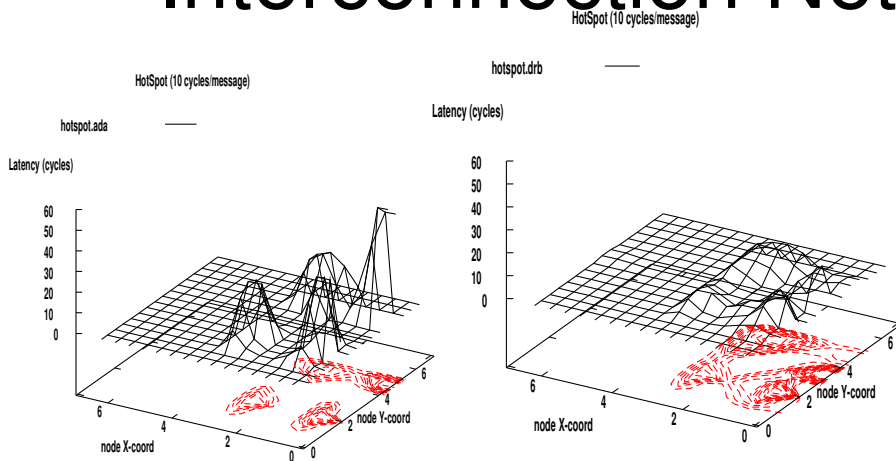
Management of Parallel and Distributed Systems

- Low Cost Parallel Machine

- Performance Modelling and Prediction of Multiclusters
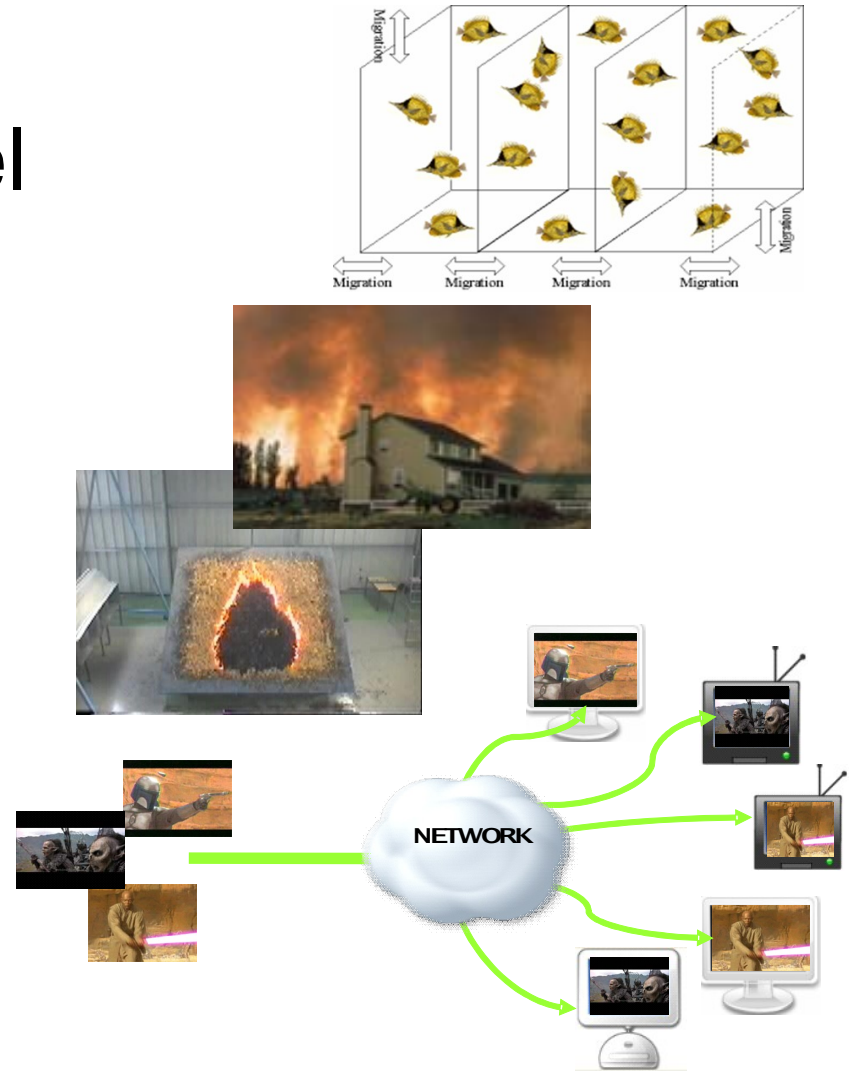
- Grid Resource Management

Support Tools for Parallel Processing

- Parallel Programming Models

- Automatic Performance Evaluation and Dynamic Tuning

- Reliability (Fault Tolerant)

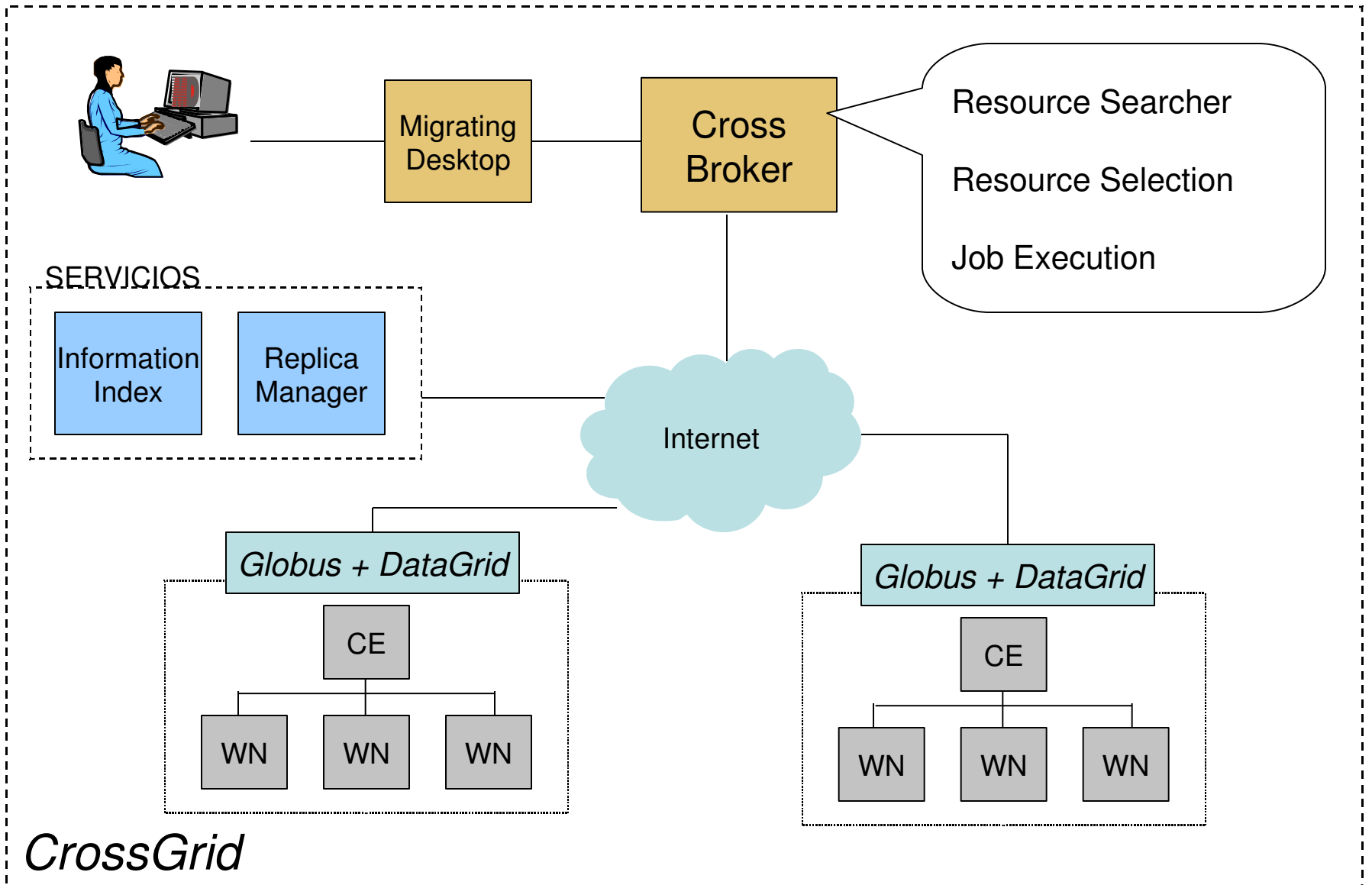# Parallel Applications

- Event Driven Parallel and Distributed Simulation

- Simulation of Forest Fire Propagation

- Video on Demand

Grid Resource Management: CrossBroker

- Grid Resource Management
  - Automatic
  - Transparent
  - Reliable
- Supporting new applications:
  - Interactive
  - Workflows
  - Parallel

CrossBroker

SERVICIOS

Information Index

Replica Manager

Migrating Desktop

Cross Broker

Resource Searcher

Resource Selection

Job Execution

Internet

Globus + DataGrid

CE

WN WN WN

Globus + DataGrid

CE

WN WN WN

CrossGrid

Simulation of Forest Fire Propagation

- S^2F^2M (Statistical System for Forest Fire Management) is a statistical method that improves the prediction of fire propagation by focusing on the imprecision of the input data

- S^2F^2M looks for a behavior pattern of the forest fire, independently of the parameters values

- S^2F^2M presents an option to the resolution of the problem by means of the use of multiple cases instead of doing a prediction from only one parameters set

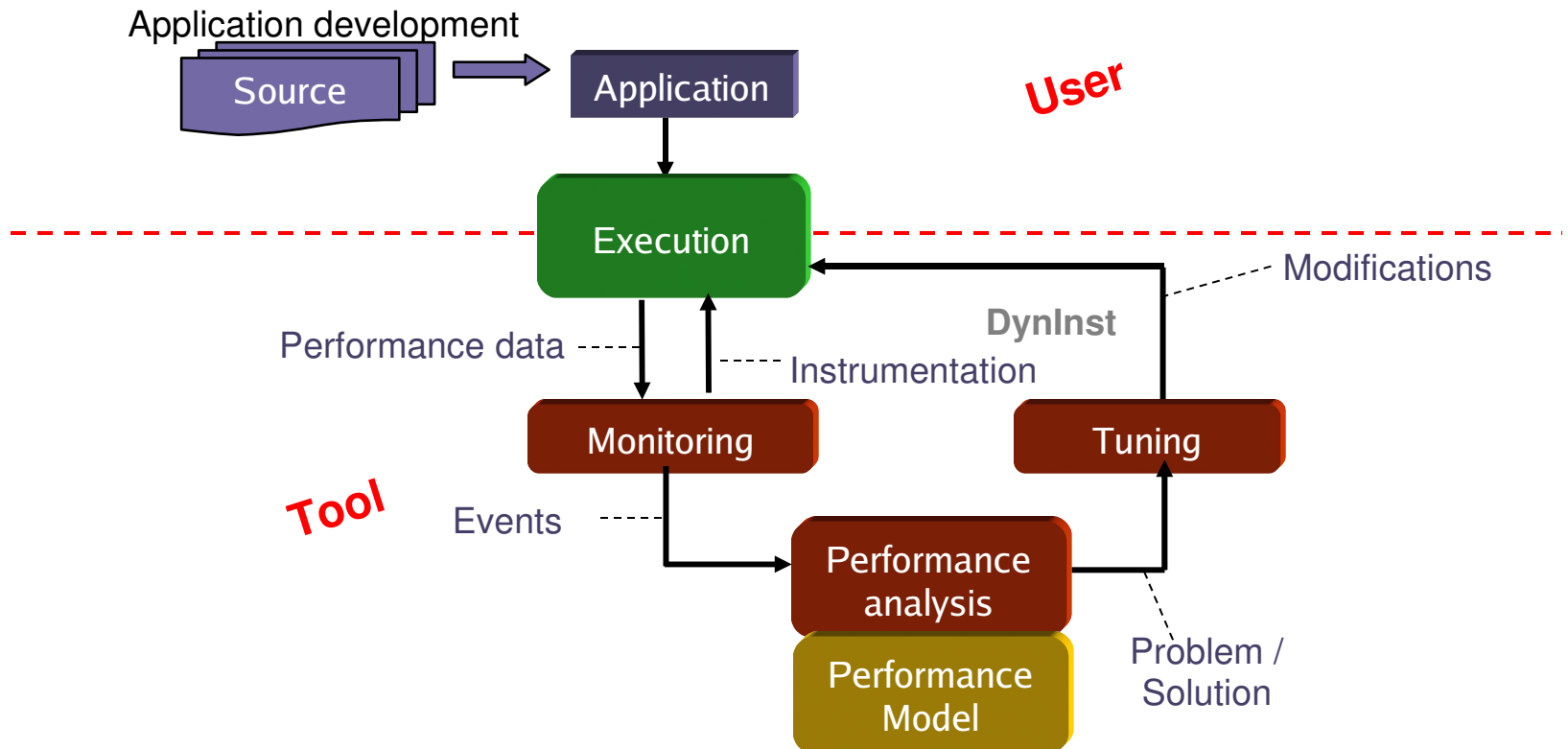# Simulation of Forest Fire Propagation
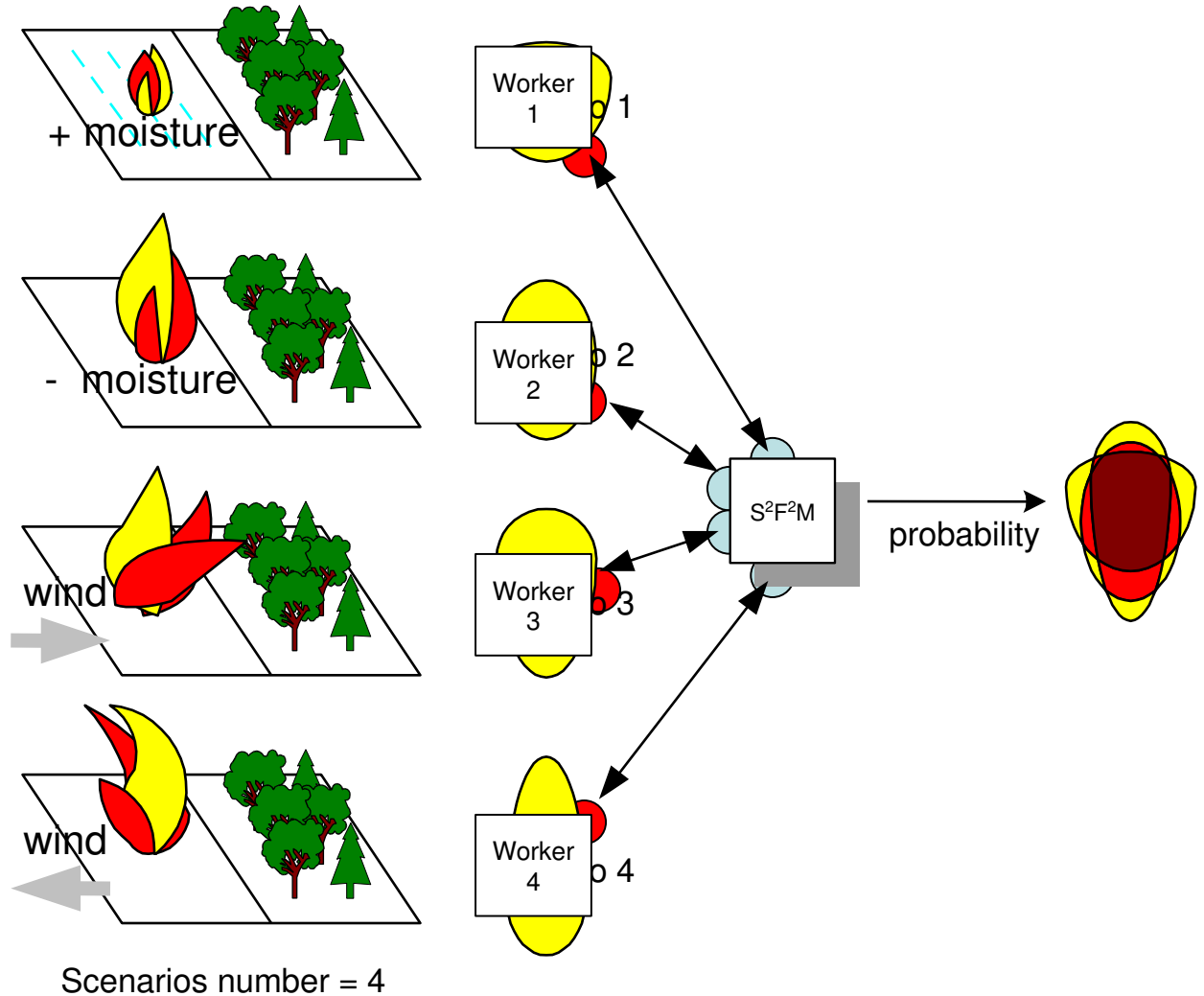
Simulation of Forest Fire Propagation

- In real experiments, the terrain is divided into dedicated plots with regular shape and dimensions separated by firewalls to limit fire spread and to keep it inside desired boundaries in each burn
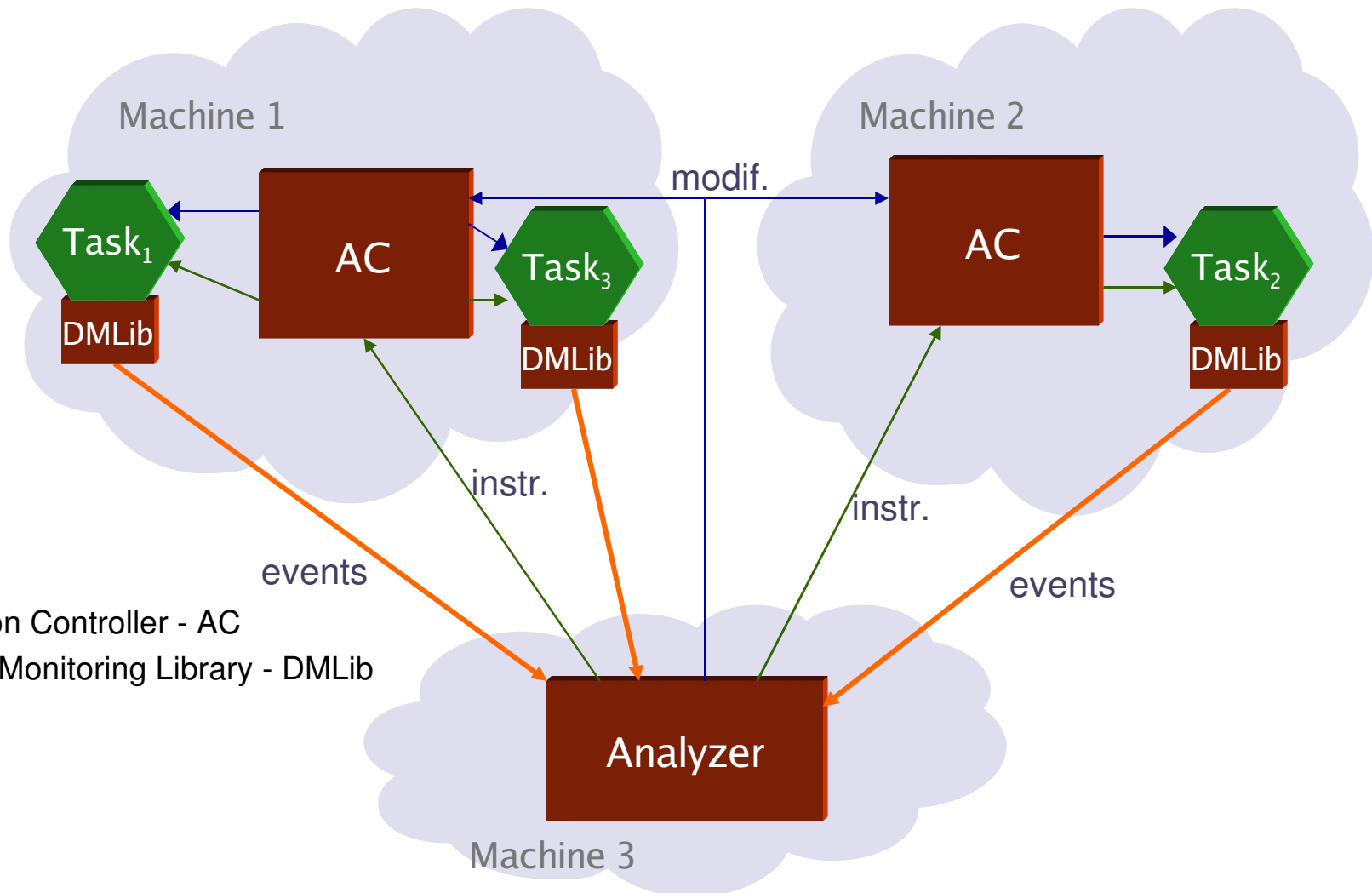
Dynamic Tuning

– Dynamic automatic tuning of parallel/distributed applications

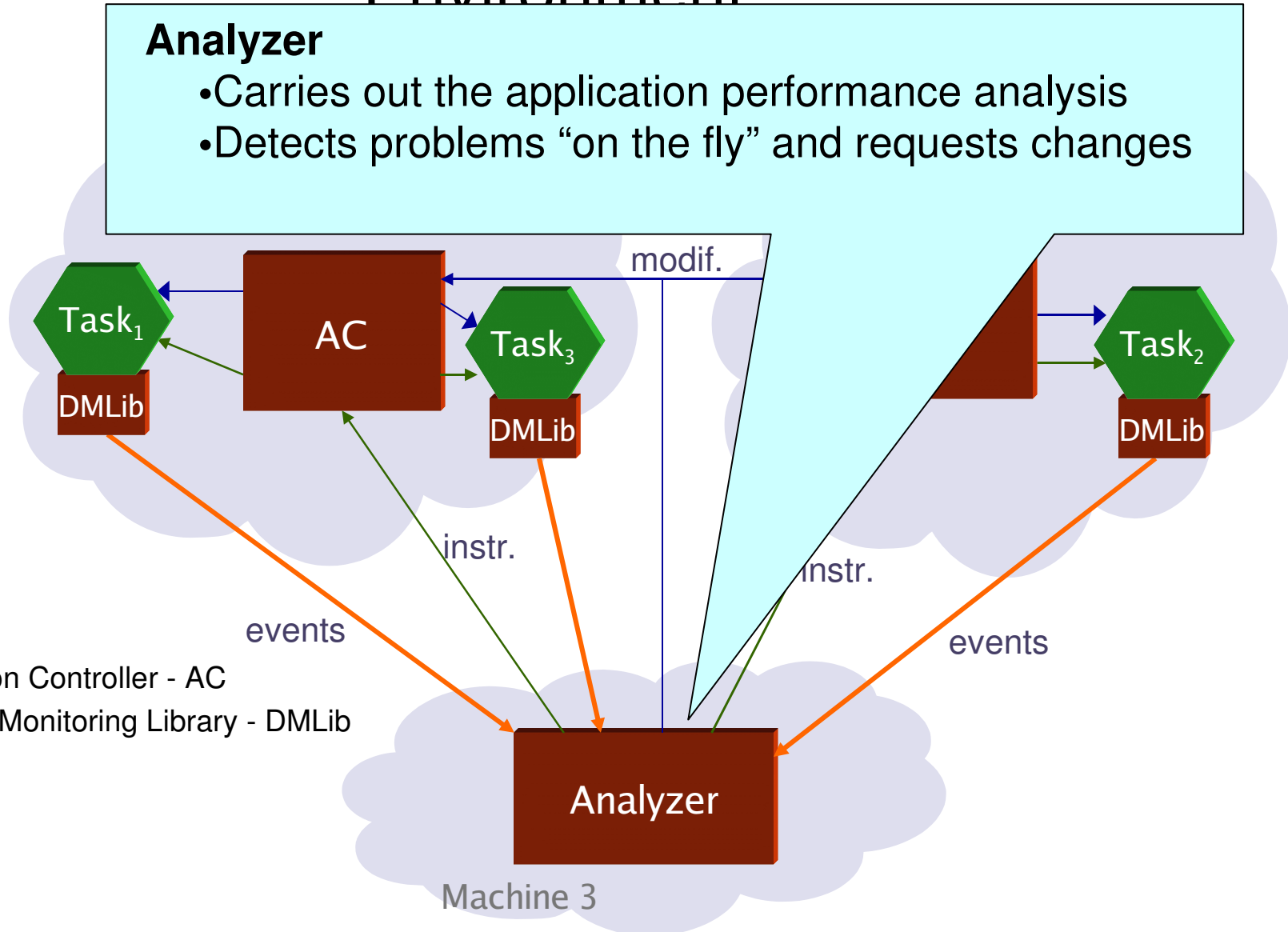# MATE: Monitoring, Analysis and Tuning Environment

- Application Controller - AC
- Dynamic Monitoring Library - DMLib
- Analyzer

MATE: Monitoring, Analysis and Tuning Environment

Analyzer

- Carries out the application performance analysis
- Detects problems "on the fly" and requests changes

Task₁

AC

Task₃

DMLib

DMLib

modif.

Task₂

DMLib

instr.

instr.

events

events

- Application Controller - AC
- Dynamic Monitoring Library - DMLib
- Analyzer

Analyzer

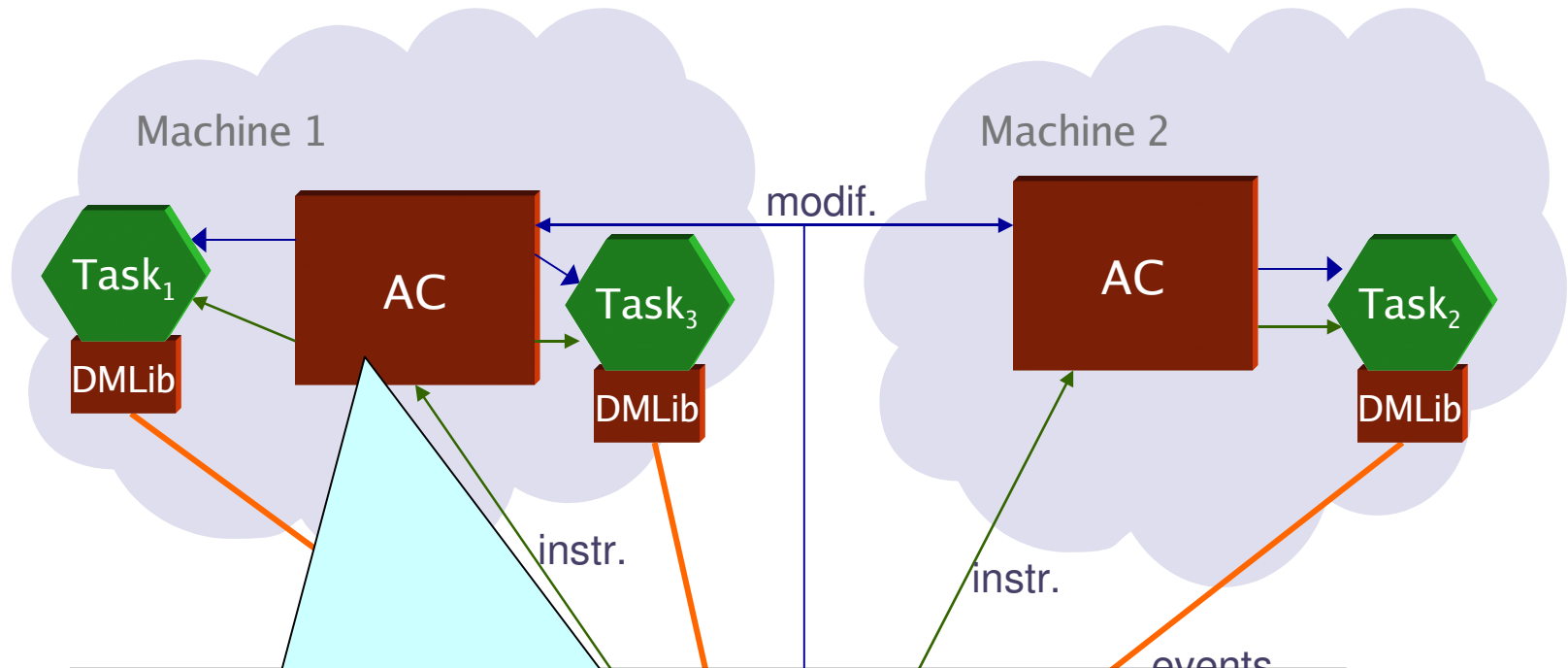Machine 3

MATE: Monitoring, Analysis and Tuning Environment

Machine 1

Task₁

DMLib

AC

Task₃

DMLib

modif.

Machine 2

AC

Task₂

DMLib

instr.

instr.

events

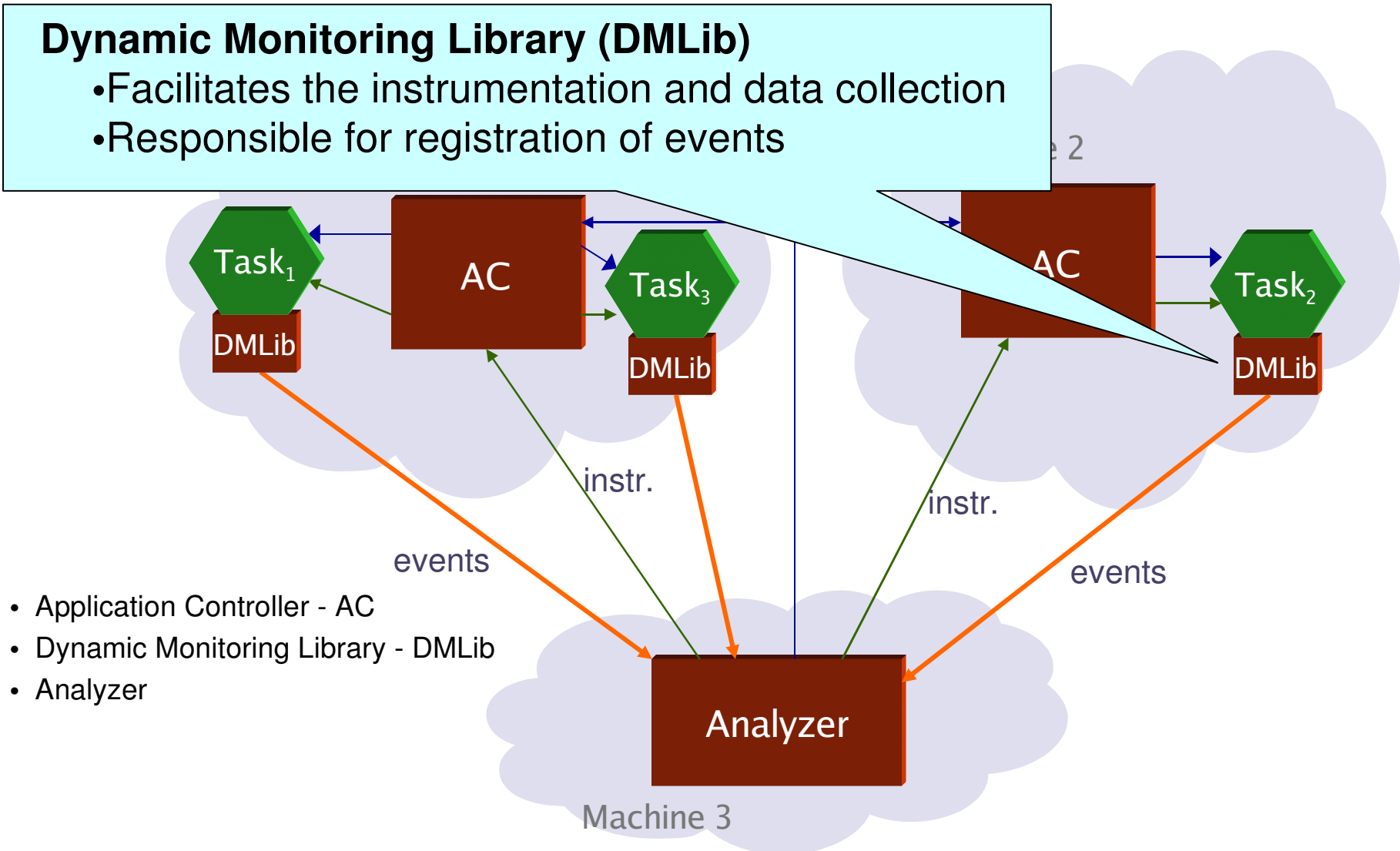Machine 3

- Application Co...
- Dynamic Mon...
- Analyzer

Application Controller (AC)
- Controls the execution of the application
- Has a Monitor module to manage instrumentation via DynInst and gather execution information
- Has a Tuner module to perform tuning via DynInst

MATE: Monitoring, Analysis and Tuning Environment

Dynamic Monitoring Library (DMLib)
- Facilitates the instrumentation and data collection
- Responsible for registration of events

Task$_1$

AC

Task$_3$

DMLib

DMLib

AC

Task$_2$

DMLib

instr.

instr.

events

events

- Application Controller - AC
- Dynamic Monitoring Library - DMLib
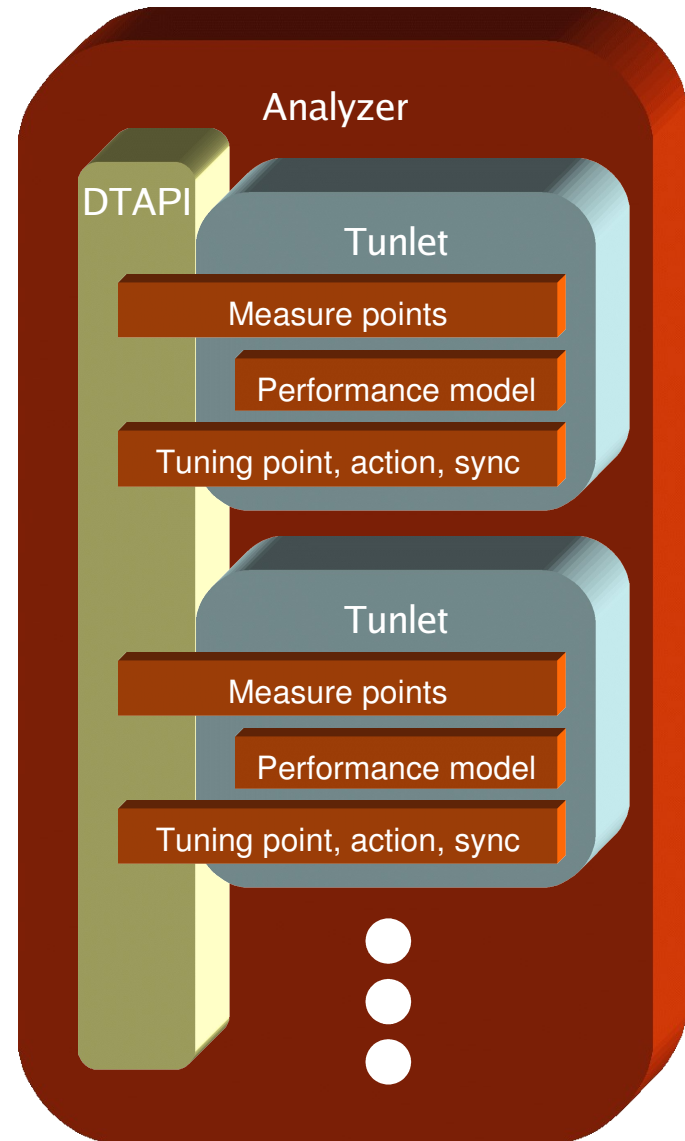- Analyzer

Analyzer

Machine 3

MATE: Monitoring, Analysis and Tuning Environment

– **Automatic Performance Analysis on the fly**

- Find bottlenecks among events applying performance model
- Find solutions that overcome bottlenecks

– Analyzer is provided with an application knowledge about performance problems

– Information related to one problem is called a tuning technique

– A tuning technique describes a complete performance optimization scenario

MATE: Monitoring, Analysis and Tuning Environment

- Each tuning technique is implemented in MATE as a **"tunlet"**, a C/C++ library dynamically loaded to the Analyzer process.

  – **measure points** – *what* events are needed

  – **performance model** – *how* to determine the behavior and possible solutions

  – **tuning actions/points/synchronization** - *what* to change, *where, when*

University Autonoma of Barcelona (**UAB**)