

---

# Workflow Management Systems

---

Marco Slot ([marco@few.vu.nl](mailto:marco@few.vu.nl))

Paul van Zoolingen ([pfvzooli@few.vu.nl](mailto:pfvzooli@few.vu.nl))

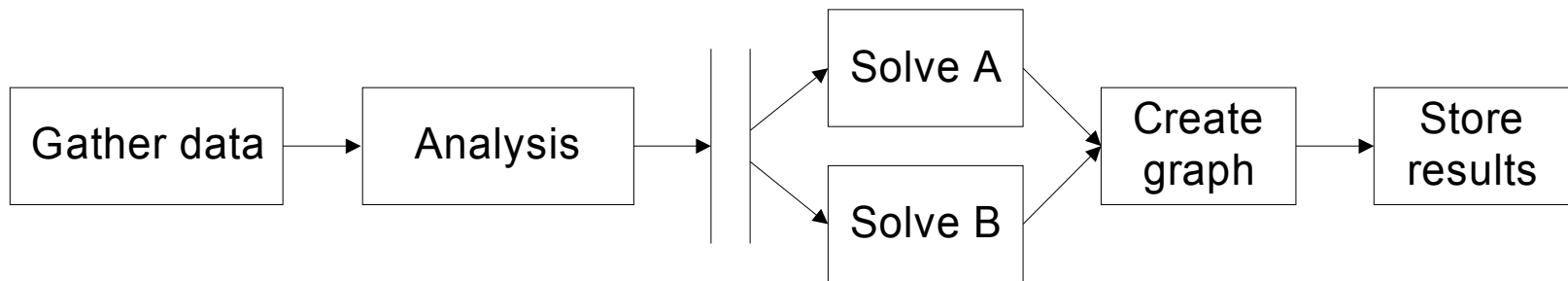
---

# Outline

- **What is a Workflow (management system)?**
  - Workflow design
  - Scheduling
  - Execution
  - Monitoring
  - Fault tolerance
  - Case study: Triana
-

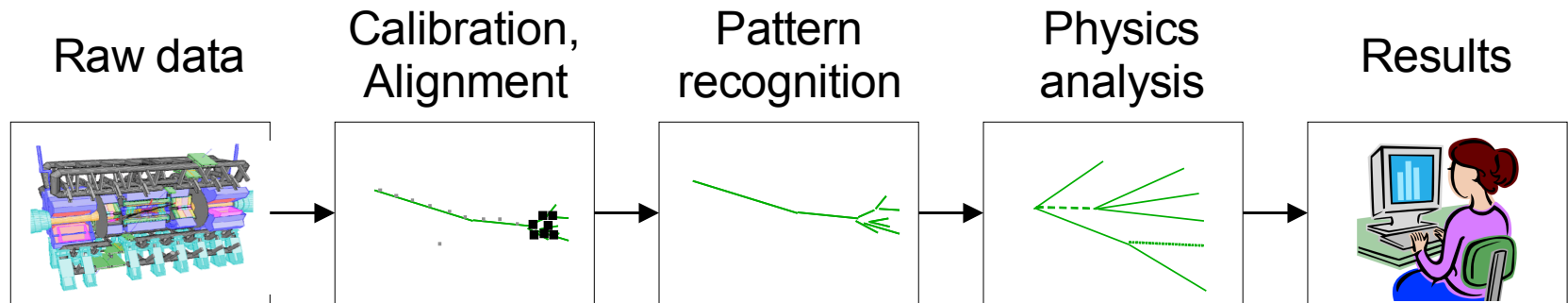
# What is a Workflow?

Link reusable components and data-filters together to achieve data-flow parallelism.



# What is a Workflow?

Complex applications may consist of several different 'independent' steps. Separate these steps and create a workflow out of them.



---

# Workflow Management System?

From the programmer perspective:

When building a Workflow Application you run into:

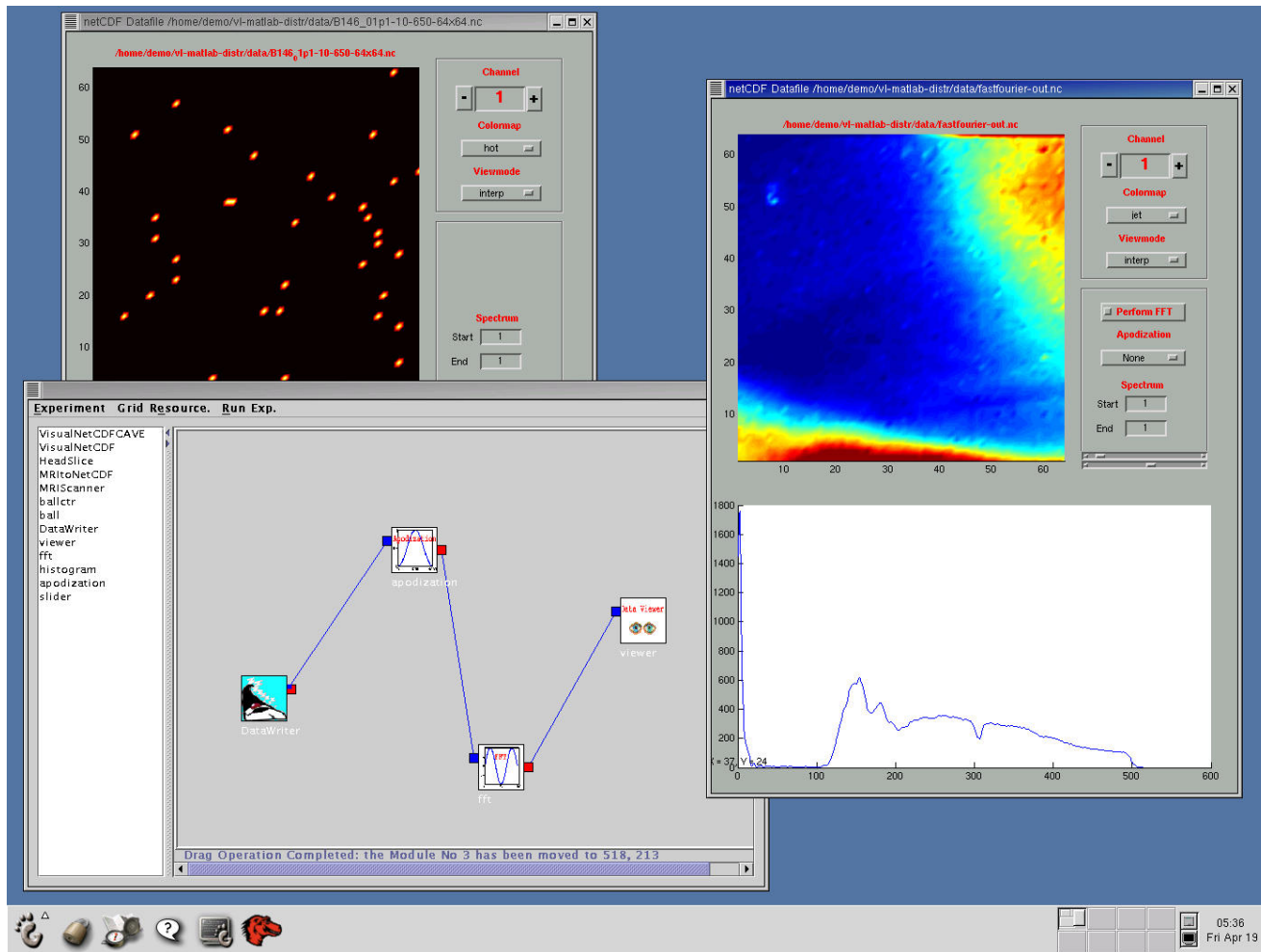
- Complex scheduling
- Complex monitoring
- Complex fault tolerance
- etc.

Workflow Management Systems help you develop and manage the application.

---

# Workflow Management System?

From the user (e-scientist) perspective:

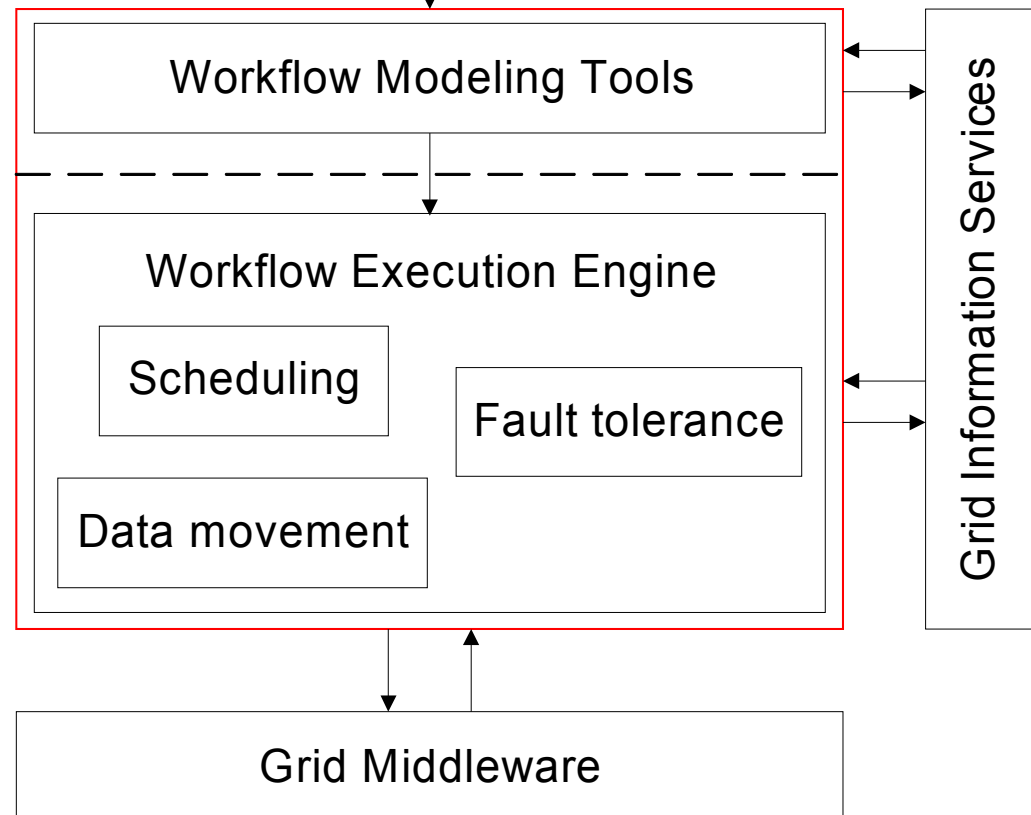


Virtual Lab Amsterdam (now VL-e)

# Workflow Management System?



General Architecture:



---

# Outline

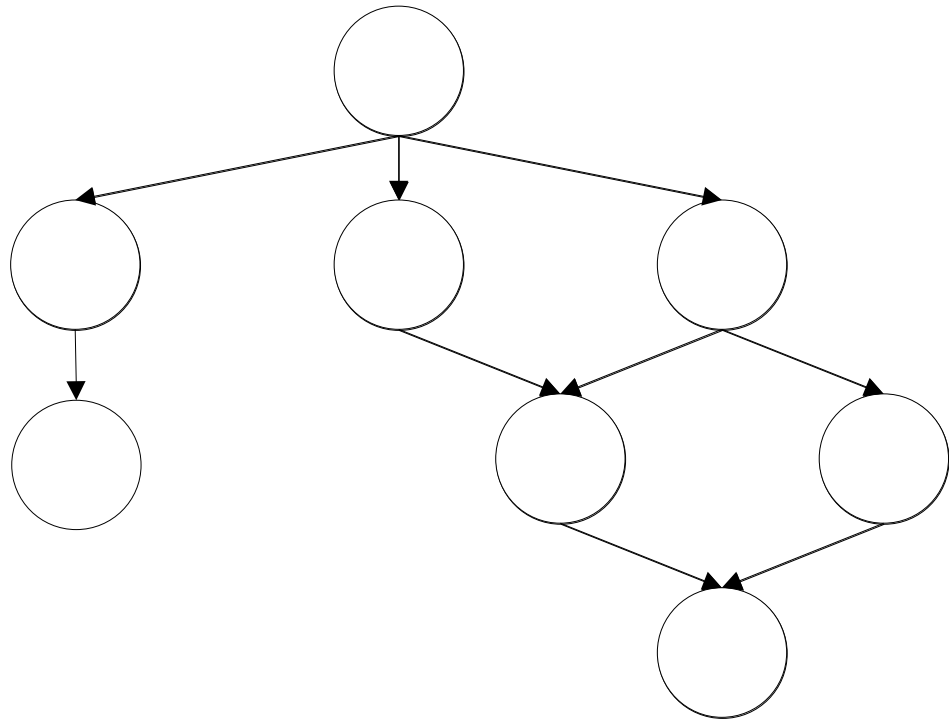
- What is a Workflow (management system)?
  - **Workflow modeling**
  - Scheduling
  - Execution
  - Monitoring
  - Fault tolerance
  - Case study: Triana
-

# Workflow Structure

Starting point: Directed (Acyclic) Graph

Node could be:

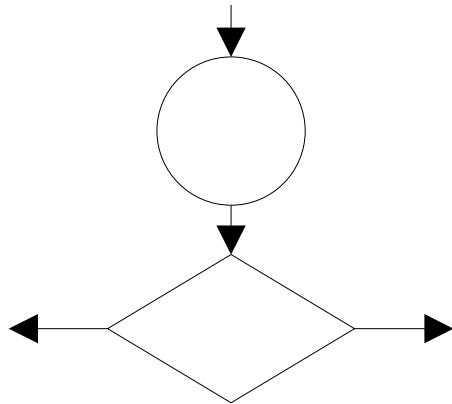
- Stage in
- Job
- Stage out
- Collect data
- Sub-workflow
- ...



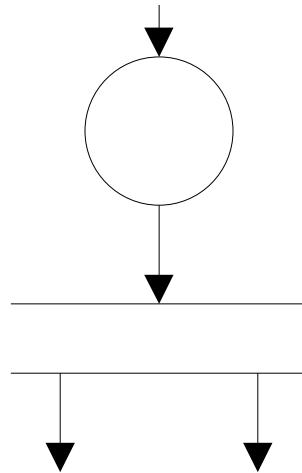
# Workflow Structure

## Additional control structures

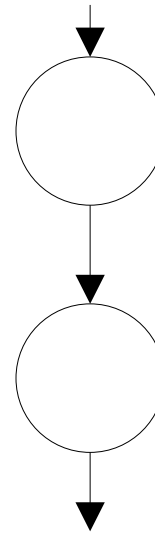
Choice



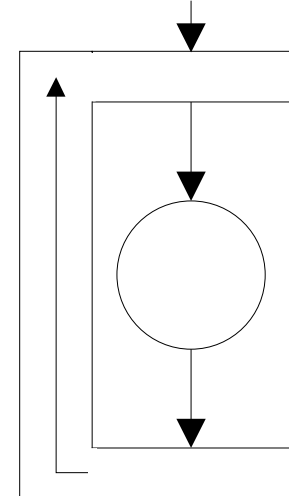
Parallel



Sequence

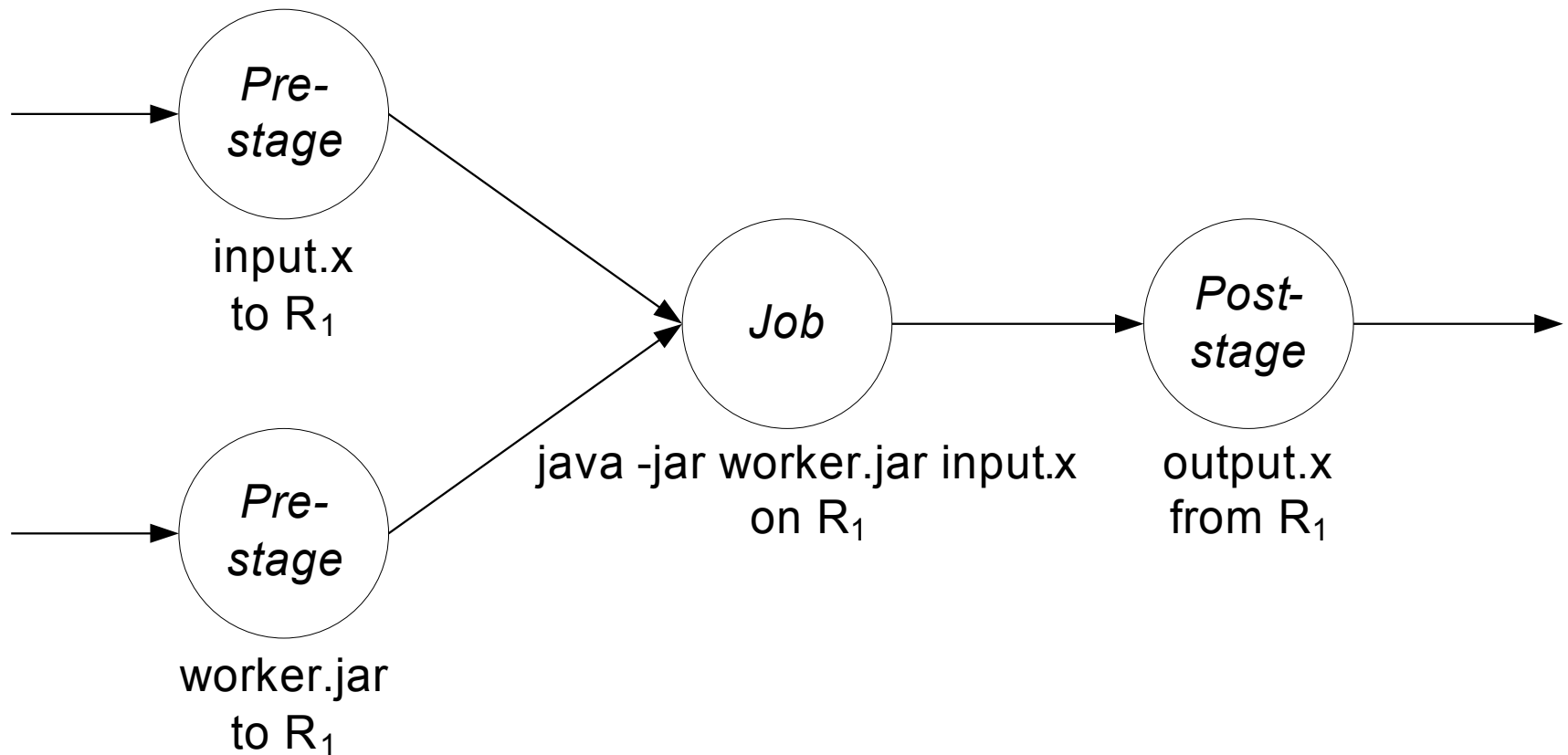


Loop



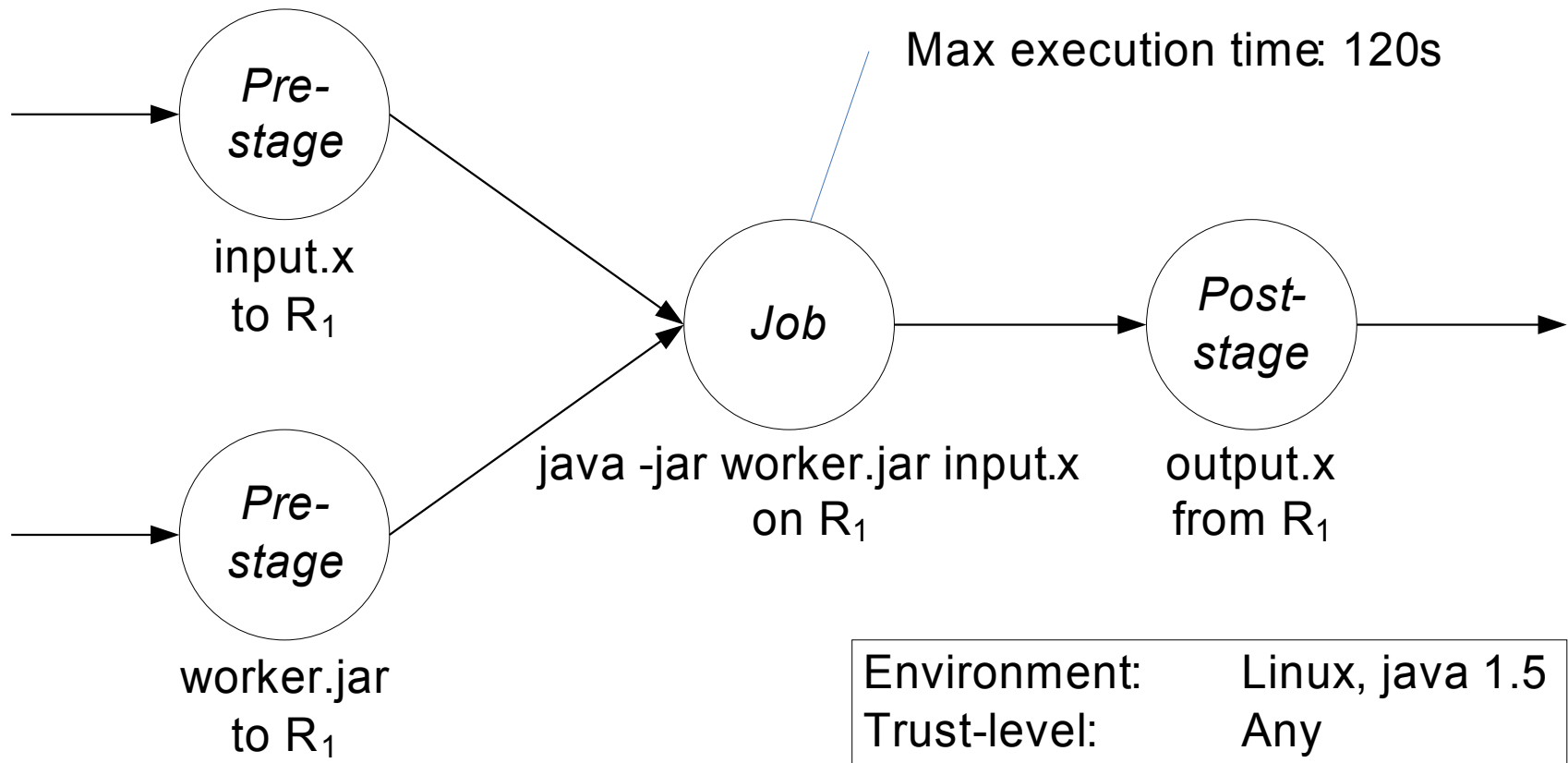
# Workflow Modeling

Example:



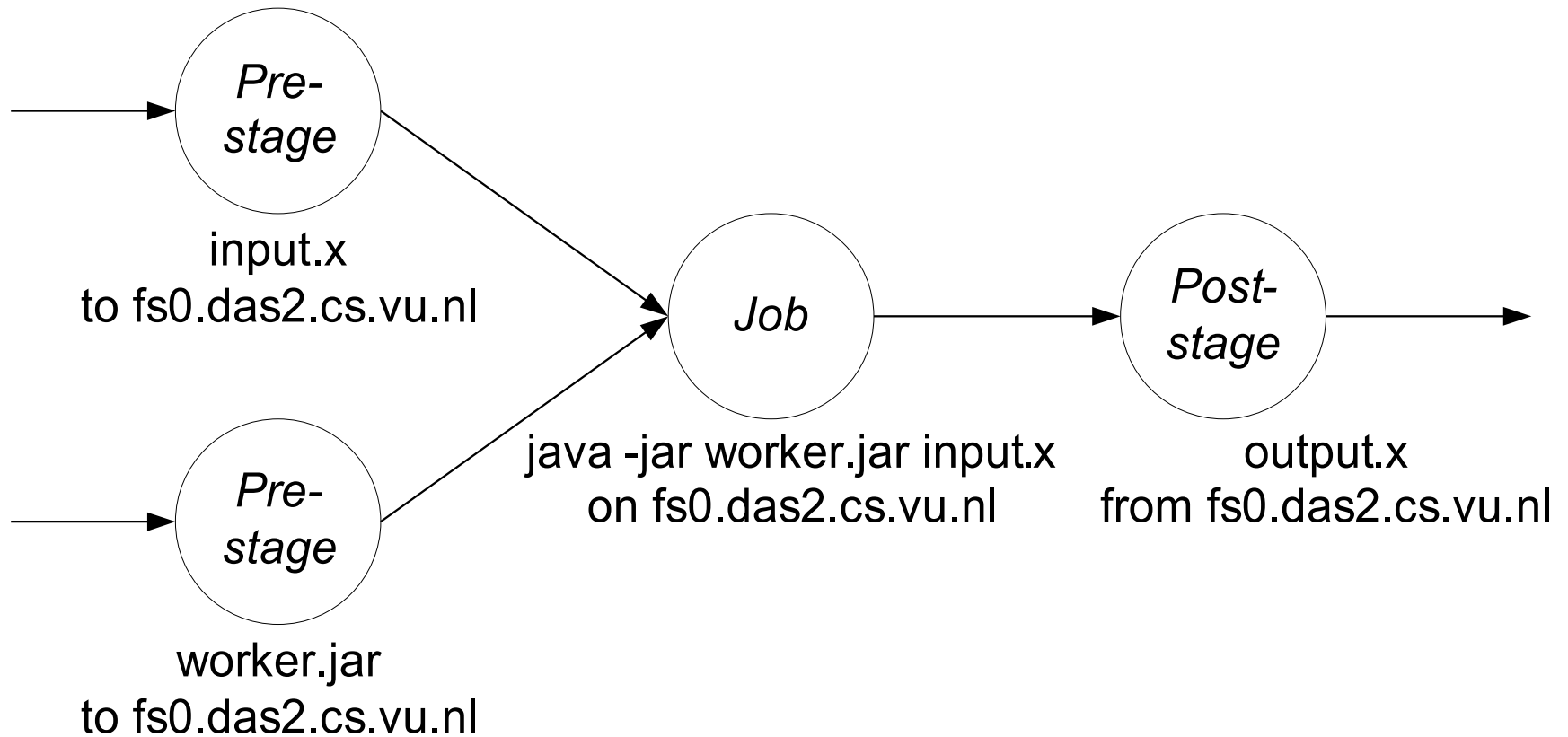
# Workflow Modeling

Add resource requirements and QoS constraints:



# Workflow Modeling

Specify specific resources:



---

# Workflow Modeling Tools

Workflow Modeling Tools let you describe:

- Tasks
- Data
- Dependencies
- Optional: QoS constraints
- Optional: Resources

Method:

- Graphical tools
  - Languages
-

# Graphical Modeling Tools

Assemble a workflow using a graphical interface.

The screenshot displays the UNICORE Client software interface, which is used for assembling workflows. The window title is "UNICORE Client" and it features a menu bar with "File", "Job Preparation", "Job Monitoring", "Settings", "Extensions", and "Help".

The interface is divided into several panels:

- Left Panel (Job Preparation):** Shows a tree view of job preparation tasks. The "Solve" task is selected and highlighted in yellow. Below it are "Postprocess", "Download...", "Upload\_Data", "Preprocess", "Solve\_FO1", and "Solve\_PP".
- Bottom-Left Panel (GROW):** Shows a tree view of GROW (Grid Resource Object Wrapper) resources. The "GROW" folder is expanded, showing "GrowUnicoreCCS <NJS>" and "ZAM461". Under "GrowUnicoreCCS <NJS>", there is a "Solve" task and its sub-tasks: "Postprocess", "Download\_Data", "Upload\_Data", "Preprocess", "Solve\_FO1", "Solve\_PP", and "New\_Script5".
- Top-Right Panel (Dependencies):** Shows the "Dependencies" tab for the selected "Solve" task. It lists "UNICORE Site" (GATEWAY, GROW, ZAM461, ZAM461 Test, abby, zam285, zam461) and "Virtual Site" (GrowUnicoreCCS <NJS>, GrowUnicoreNoBatch <NJS>).
- Right Panel (Task Dependencies):** Shows a graphical dependency graph. The graph starts with "Upload\_Data" at the top, which points to "Preprocess". "Preprocess" then branches into "Solve\_FO1" and "Solve\_PP". Both "Solve\_FO1" and "Solve\_PP" point to "Postprocess", which finally points to "Download\_Data" at the bottom. A button labeled "Arrange Graph" is located at the bottom of this panel.

The status bar at the bottom of the window shows the user "rebecca breu" and the message "Solve [15:43:28 05/31/2006] not saved yet." The system tray on the right indicates "14.40Mb / 15.42Mb".

---

# Workflow Description Languages

Define your workflow using a (XML based) language

Languages:

- BPEL
  - WSFL
  - XLANG
  - YAWL
  - DAG
  - ...
-

# Business Process Execution Language

```
<sequence>  
  <invoke ... />  
  <flow>  
    <invoke ... />  
    <invoke ... />  
  </flow>  
  <switch>  
    <case condition="xpath_exp">  
      <invoke ... />  
    </case>  
    <case condition="other_xpath_exp">  
      <invoke ... />  
    </case>  
  </switch>  
</sequence>
```

---

# Outline

- What is a Workflow (management system)?
  - Workflow modeling
  - **Scheduling**
  - Execution
  - Monitoring
  - Fault tolerance
  - Case study: Triana
-

---

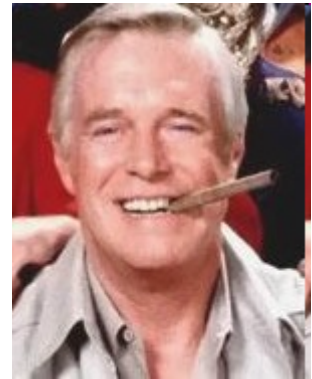
# Scheduling Workflows

We need to schedule different components with different requirements on different resources efficiently

We need a plan!

Problems:

- How to create a planning?
- How to take decisions?
- Scalability



---

# Scheduling strategies

- Performance-driven
    - Create fastest schedule
  - Market-driven
    - Create cheapest schedule
    - Electronic market place between WM-systems
  - Trust-driven
    - Use resources from VO's you trust most
-

---

# Performance Estimation

To find a good performance-driven schedule we have to estimate how long our tasks will run on specific resources.

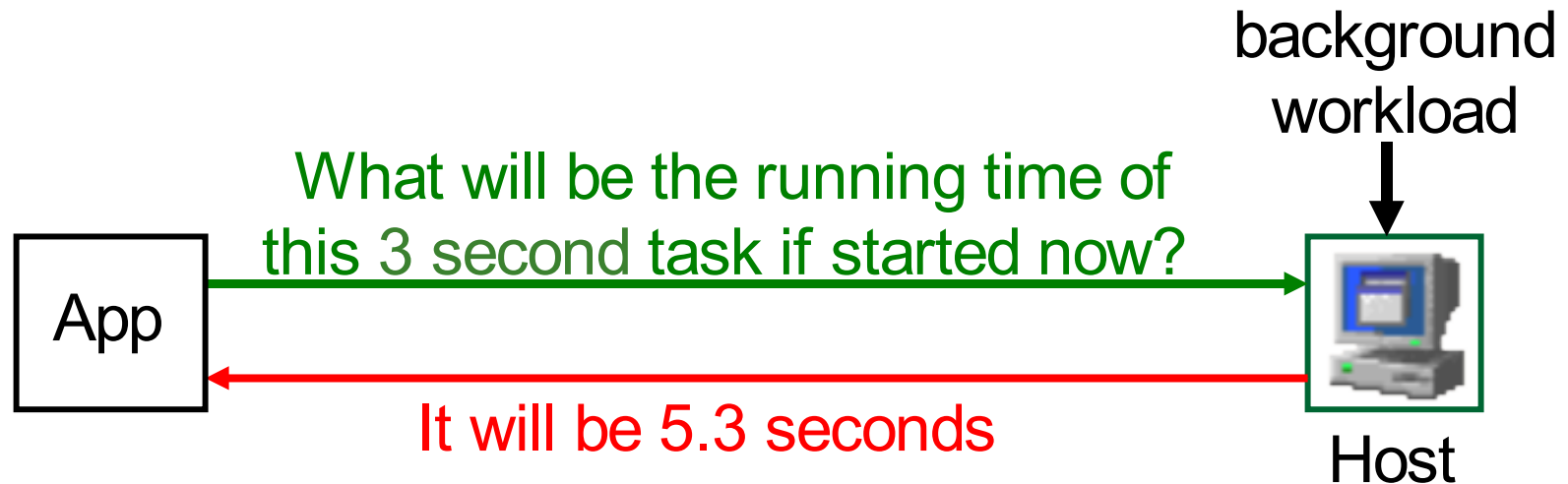
Initial estimation from:

- User / programmer
  - Problem size (computational complexity)
  - Previous runs
  - Simulation
-

# Scheduling time

Creating the schedule:

- Full ahead
- Prediction-based
- Just in-time



# Scheduling architectures

Scalability issue: What happens with massive workflows?

<b>Centralized</b>	<b>One scheduler</b> Good schedule, all information available. Poor scalability	most
<b>Decentralized</b>	<b>Multiple schedulers for sub-workflows</b> Poor schedule, partial information. Good scalability	Triana
<b>Hierarchical</b>	<b>Hierarchy of schedulers</b> Reasonable schedule, different policy per sub-workflow Reasonable scalability	GridFlow Gridbus

---

# Example: GridFlow

One component in the Workflow is a sub-workflow which uses resources from one particular VO ('local grid').

## **Simulation-based planning scheme:**

Pick 'best' local grids and schedule sub-workflows on each of them. Analyze performance results.

## **Hierarchical scheduling architecture:**

Send mobile agent to remote site to manage local scheduling.

---

---

# Outline

- What is a Workflow (management system)?
  - Workflow modeling
  - Scheduling
  - **Execution**
  - Monitoring
  - Fault tolerance
  - Case study: Triana
-