

Data Analysis using Java Mobile Agents

Mark Dönszelmann, Information, Process and Technology Group, IT, CERN
ATLAS Software Workshop
Analysis Tools Meeting, 20 May, 1999



Contents

- **Requirements and Problems**
- **Mobile Agent Paradigm**
- **Java to implement Mobile Agents**
- **Data Analysis System using Mobile Agents**
- **Problems to integrate this with FORTRAN / C++**



User Requirements

■ Access to data

- $200 \text{ Ph} * 1 \text{ Pbyte} * 1\% = 2 \text{ Pbytes of Data}$
- $200 \text{ Ph} * 25 \text{ Mbyte Jobs} * 2000 \text{ Machines} = 10 \text{ Tbytes of Data}$
- \implies move jobs rather than data

■ Access to processing power

- $130.000 \text{ SPECint95} / 100 \text{ (per machine)} = 1300 \text{ Machines}$
- \implies run on multiple cpus and/or parallelize the jobs

■ Access to/storage of intermediate results

- Jobs run for weeks, days or hours
- Machines may go down in that time
- \implies restartable (through-startable) analysis jobs

■ Easy for the physicists

- Concentrate on implementing his algorithm
- \implies not be bothered by system aspects

System Requirements

■ Move Jobs rather than Data

- Ship the code (securely)
- Ship the state objects

■ Run on multiple CPUs and/or Parallelize Jobs

- Platform independent
- Merging of the results

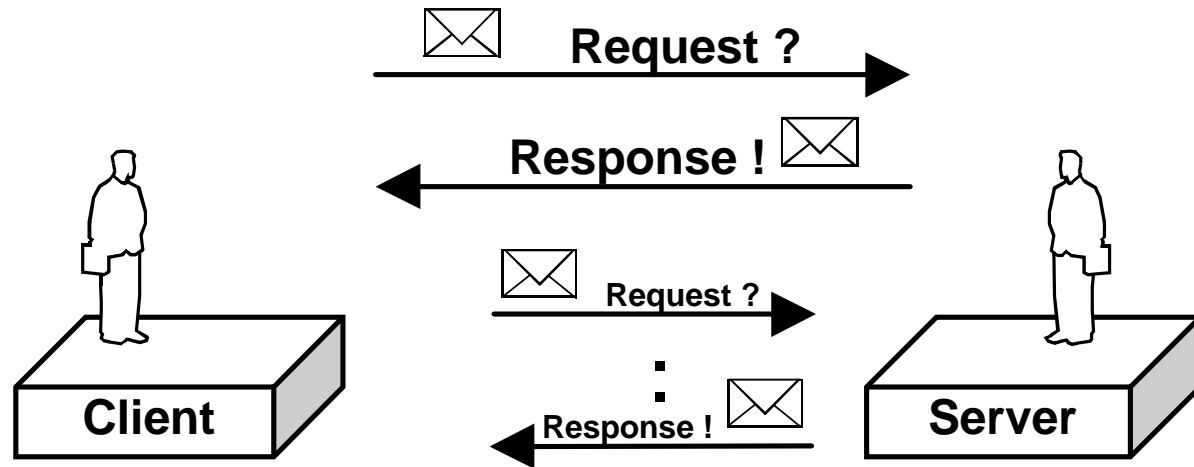
■ Make Jobs re-startable

- Checkpointing of the job (saving of state)
- Rebuilding (reloading) of the job

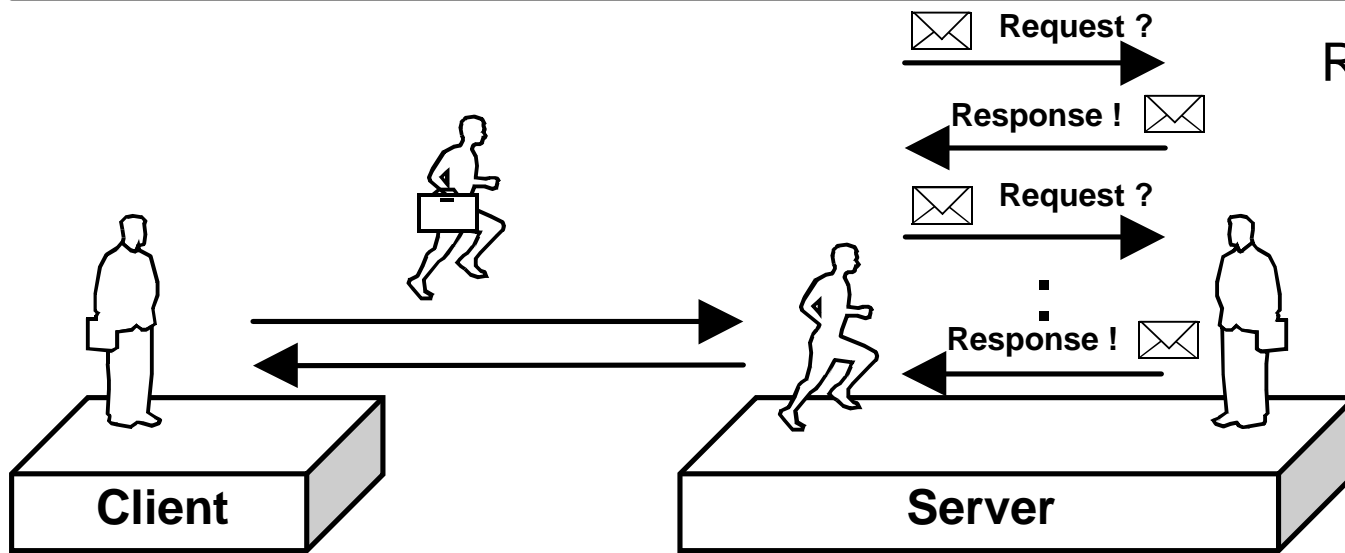


Mobile Agent Paradigm

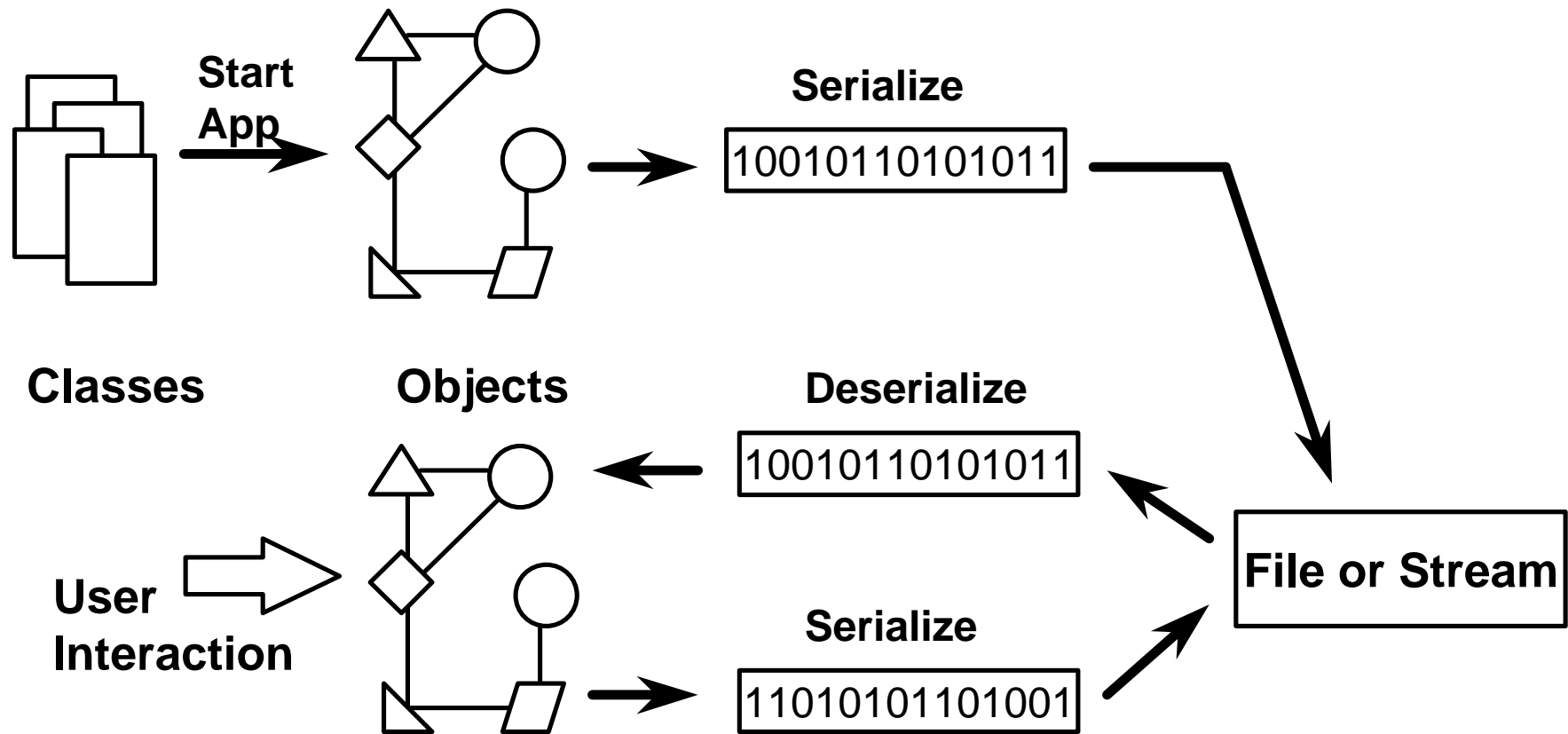
Remote Procedure Calls



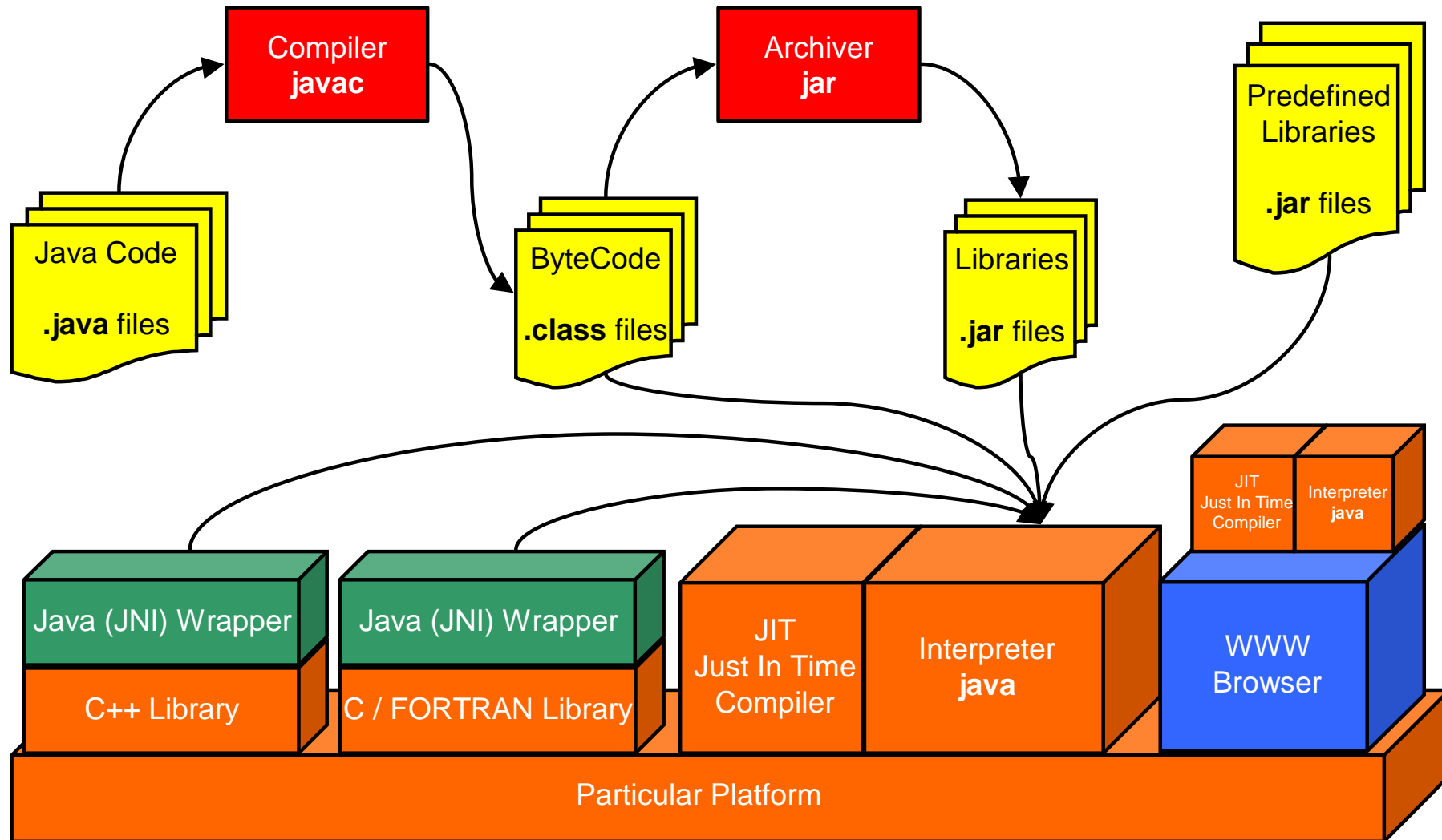
Remote Programming



Serialization and Deserialization



The Java System



Java to implement Mobile Agents

■ Platform independence

- runs on any type of CPU/OS

■ Secure Execution

- no need for special checking of the job

■ Object Serialization

- save and recover state of agent and or job

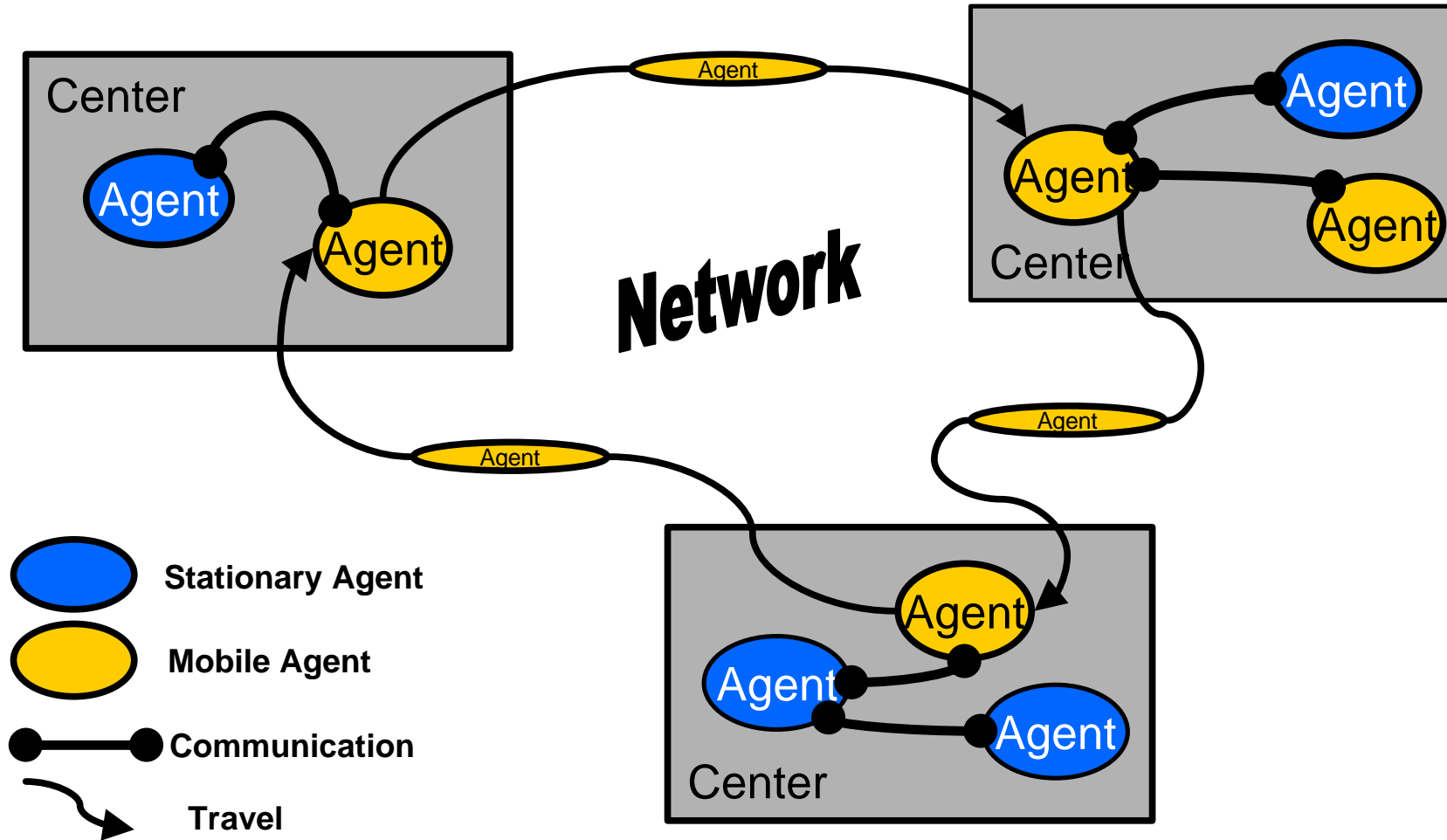
■ Reflection (runtime-type info)

- figure out how to merge

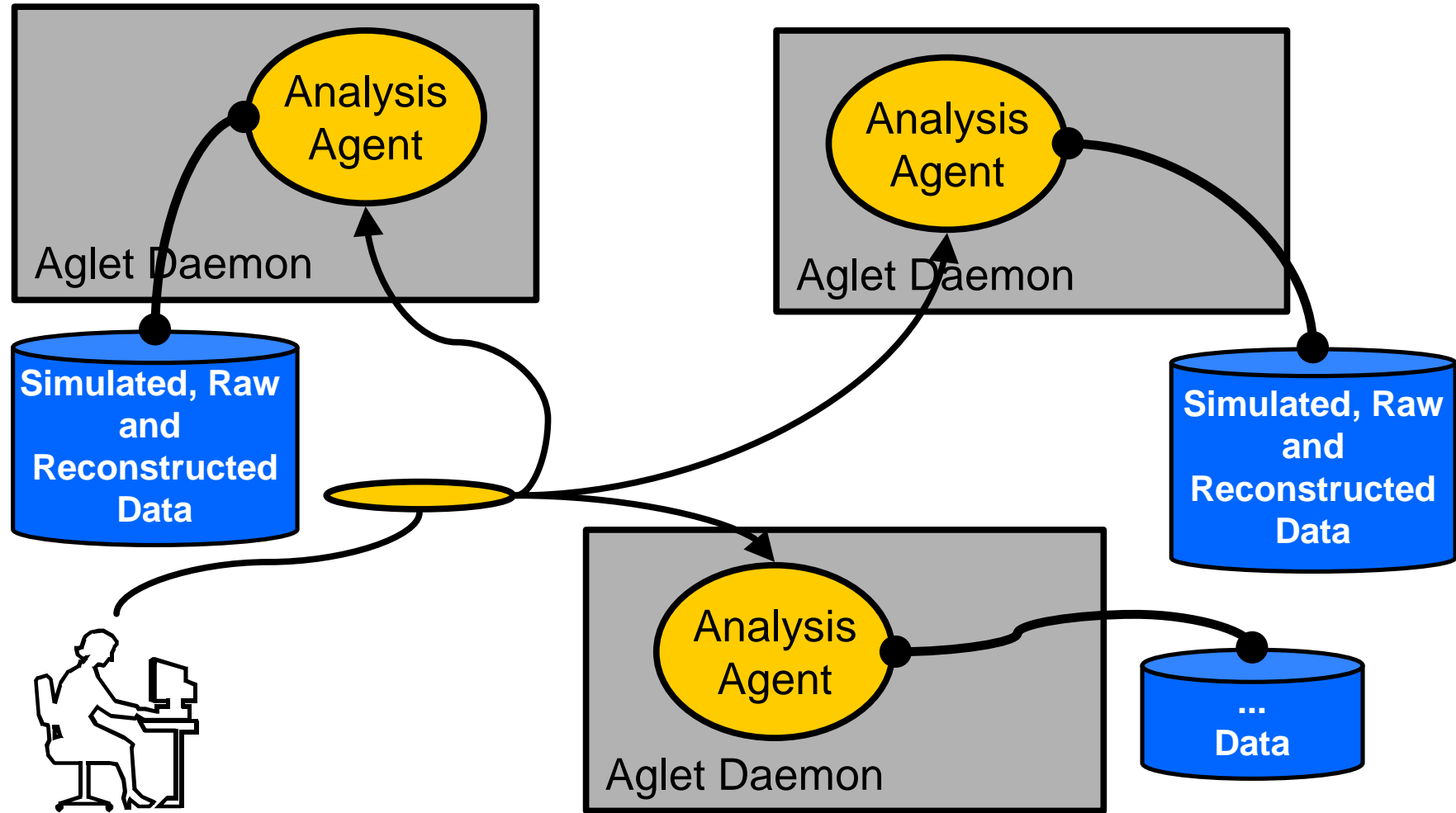
■ Example implementations:

- Aglets (IBM), Voyager (ObjectSpace), ...

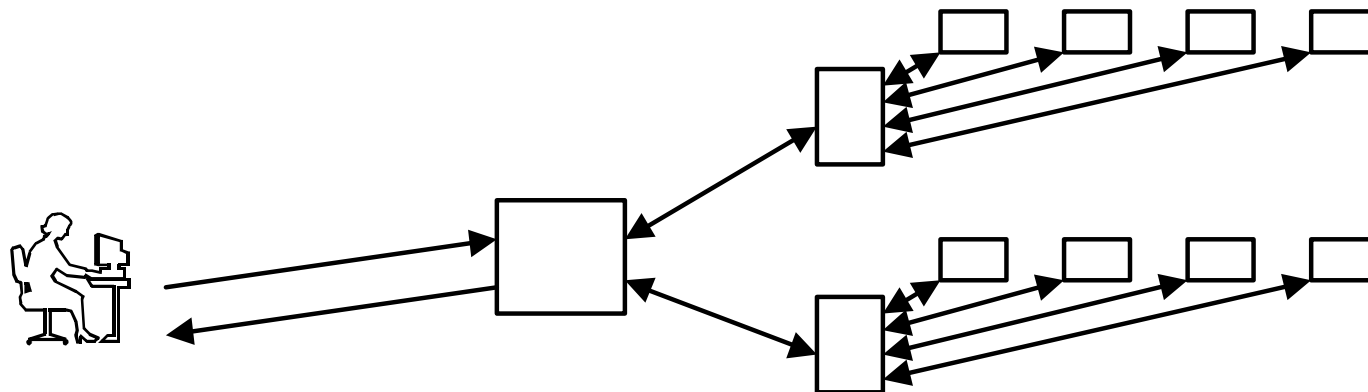
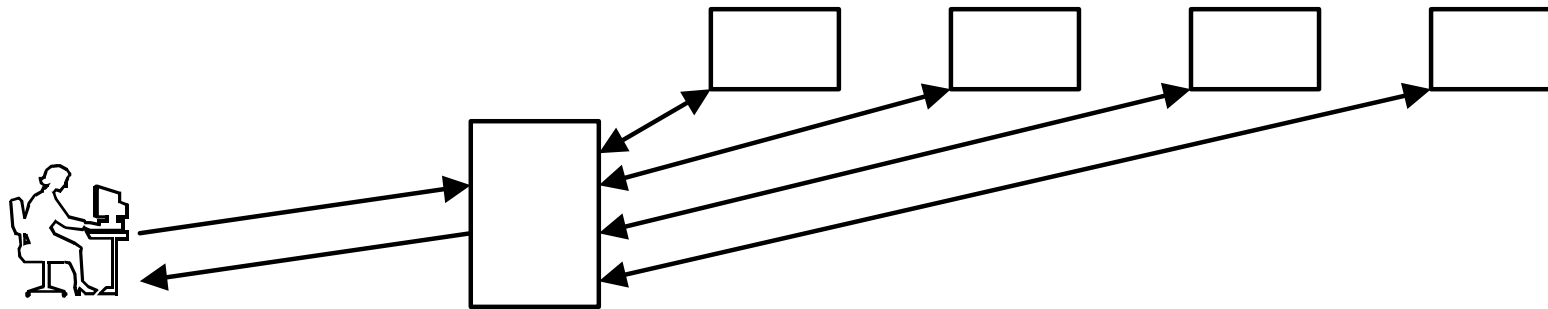
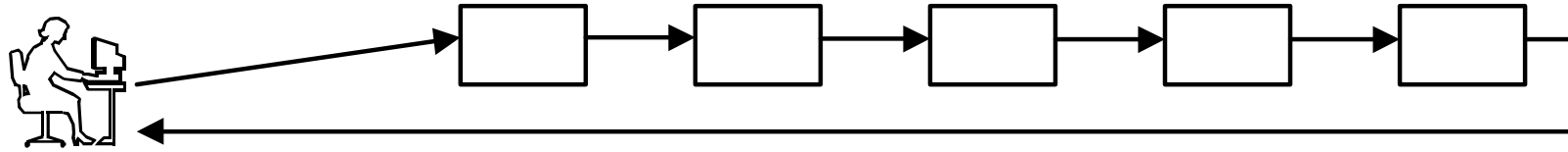
Mobile Agents and Stationary Agents



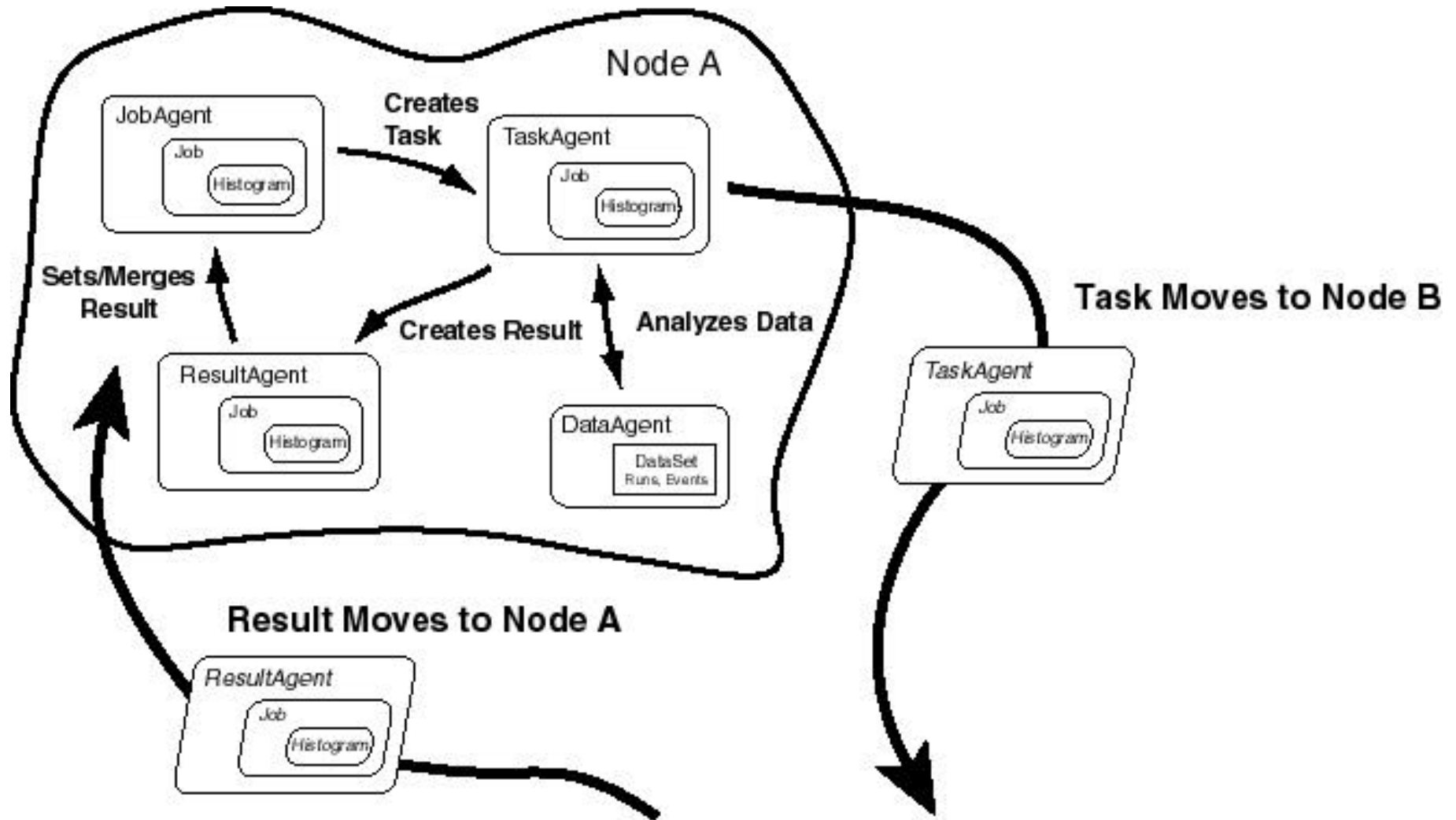
Data Analysis System using Mobile Agents



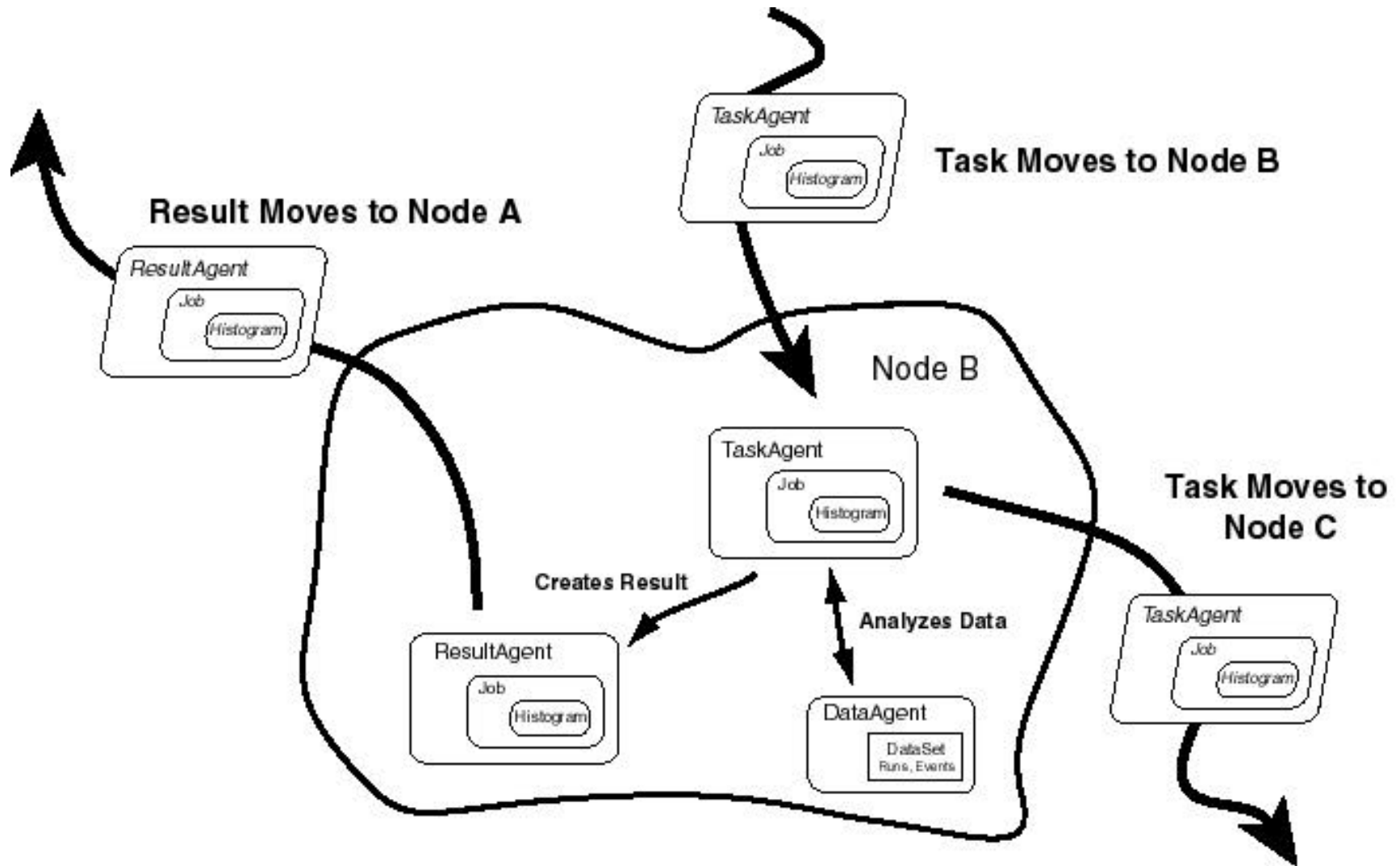
Topologies



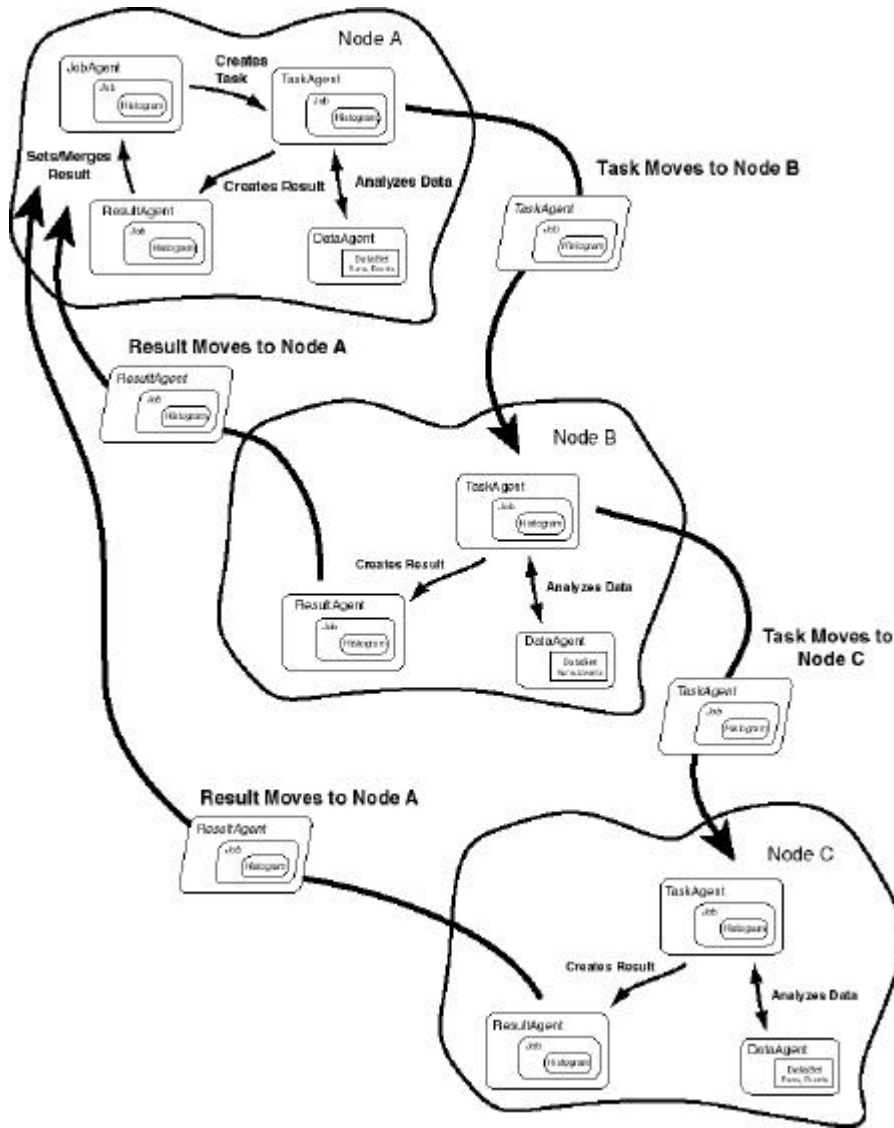
Submit, Run and move on...



... and Run and Move on...



... and finish.



CERN School of Computing

■ **35 machines, each with 150 Mbyte data locally stored.**

■ **Run a analysis job:**

- creating a histogram (total jobsite 0.5 Mbyte)
- analyzing and accessing all data $35 * 150 \text{ Mbyte} \sim 5 \text{ Gbyte}$
- running in serial mode (one machine after the other)
- producing (ongoing) results on the screen of the submitter
- in under 10 minutes
- using virtually no network traffic

■ **Assuming a 10 Mbit network:**

- Data transfer: $5 \text{ Gbyte} / 1 \text{ Mbyte/s} = 5000 \text{ seconds}$
- Job transfer: $0.5 \text{ Mbyte} * 35 / 1 \text{ Mbyte/s} = 17.5 \text{ seconds}$

■ **We could have**

- run jobs in parallel
- merged the histogram automatically



Problems to integrate this with FTN / C++

- Platform independence of executable code
- Capture of state: serialization (rtti may provide that)
- Secure transfer of executable code



Conclusions

■ The prototype of CSC has shown that

- Analysis System using Agents is possible
- reasonably easy implementation in Java
- gives us distributed analysis
- gets around some of the data access problems
 - hides a lot of system detail from the physicist

■ however

- a challenge to make this fail safe
- NOT easy to implement in C++