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

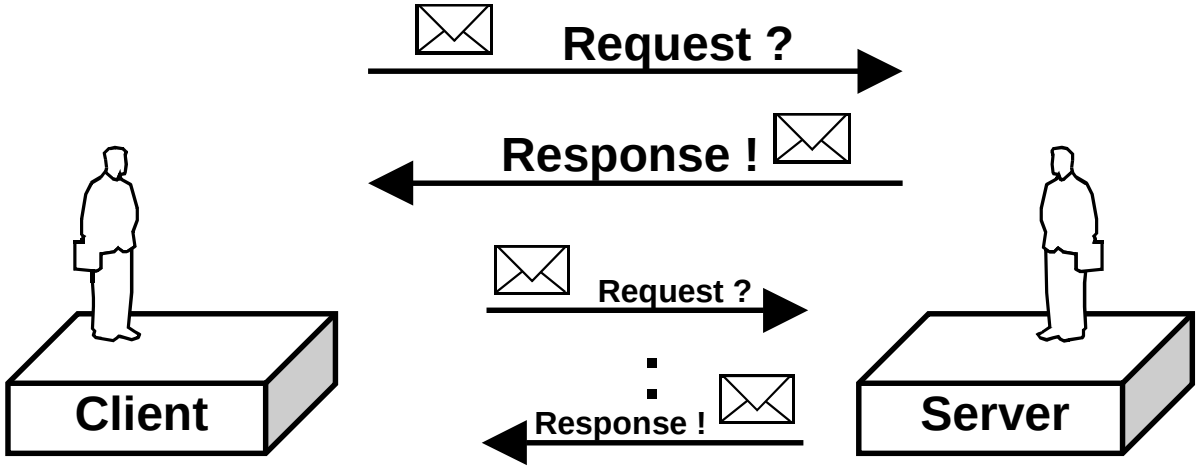
- **Access to data**
  - 200 Ph * 1 Pbyte * 1% = 2 Pbytes of Data
  - 200 Ph * 25 Mbyte Jobs * 2000 Machines = 10 Tbytes of Data
  - ==> move jobs rather than data
- **Access to processing power**
  - 130.000 SPECint95 / 100 (per machine) = 1300 Machines
  - ==> run on multiple cpus and/or parallelize the jobs
- **Access to/storage of intermediate results**
  - Jobs run for weeks, days or hours
  - Machines may go down in that time
  - ==> restartable (through-startable) analysis jobs
- **Easy for the physicists**
  - Concentrate on implementing his algorithm
  - ==> not be bothered by system aspects

System Requirements

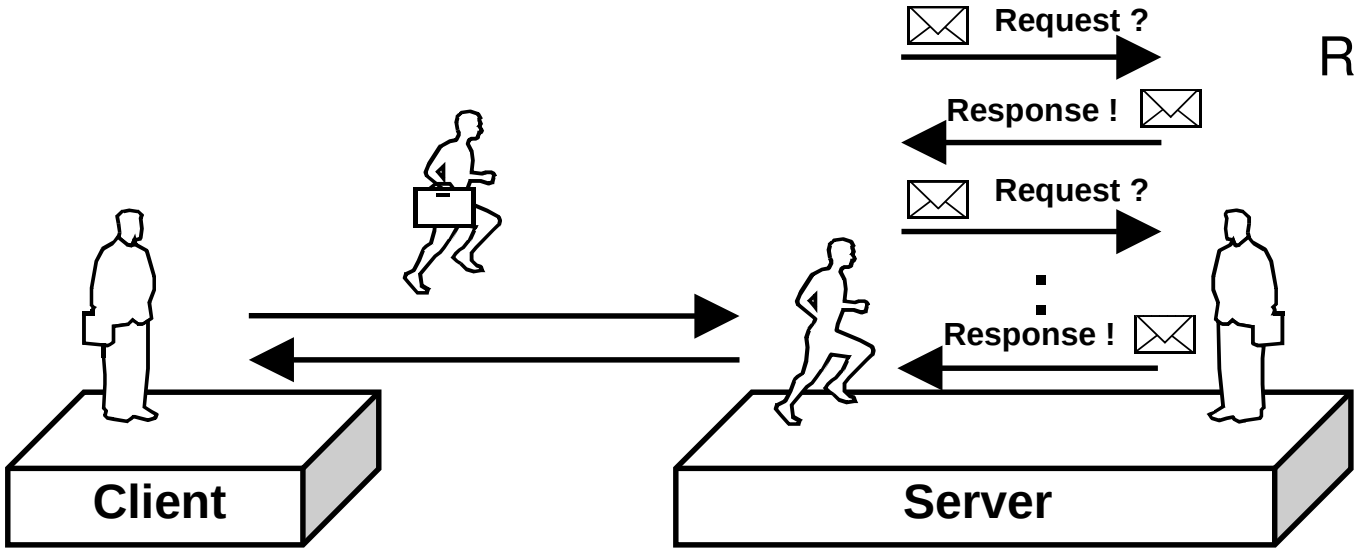
- **Move Jobs rather than Data**
  - Ship the code (securely)
  - Ship the state objects
- **Run on multiple CPUs and/or Parallelize Jobs**
  - Platform independent
  - Merging of the results
- **Make Jobs re-startable**
  - Checkpointing of the job (saving of state)
  - Rebuilding (reloading) of the job

Mobile Agent Paradigm

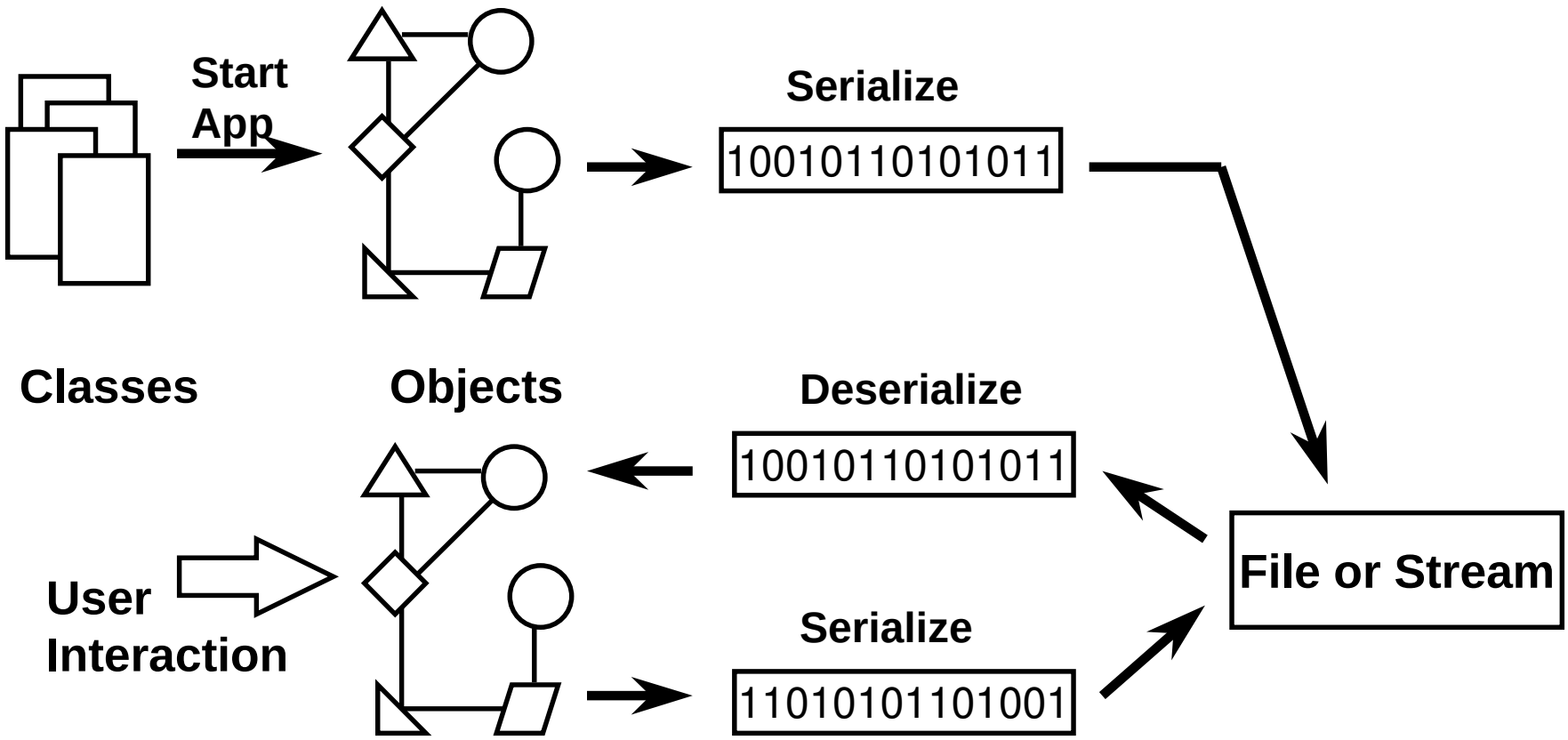
Remote Procedure Calls



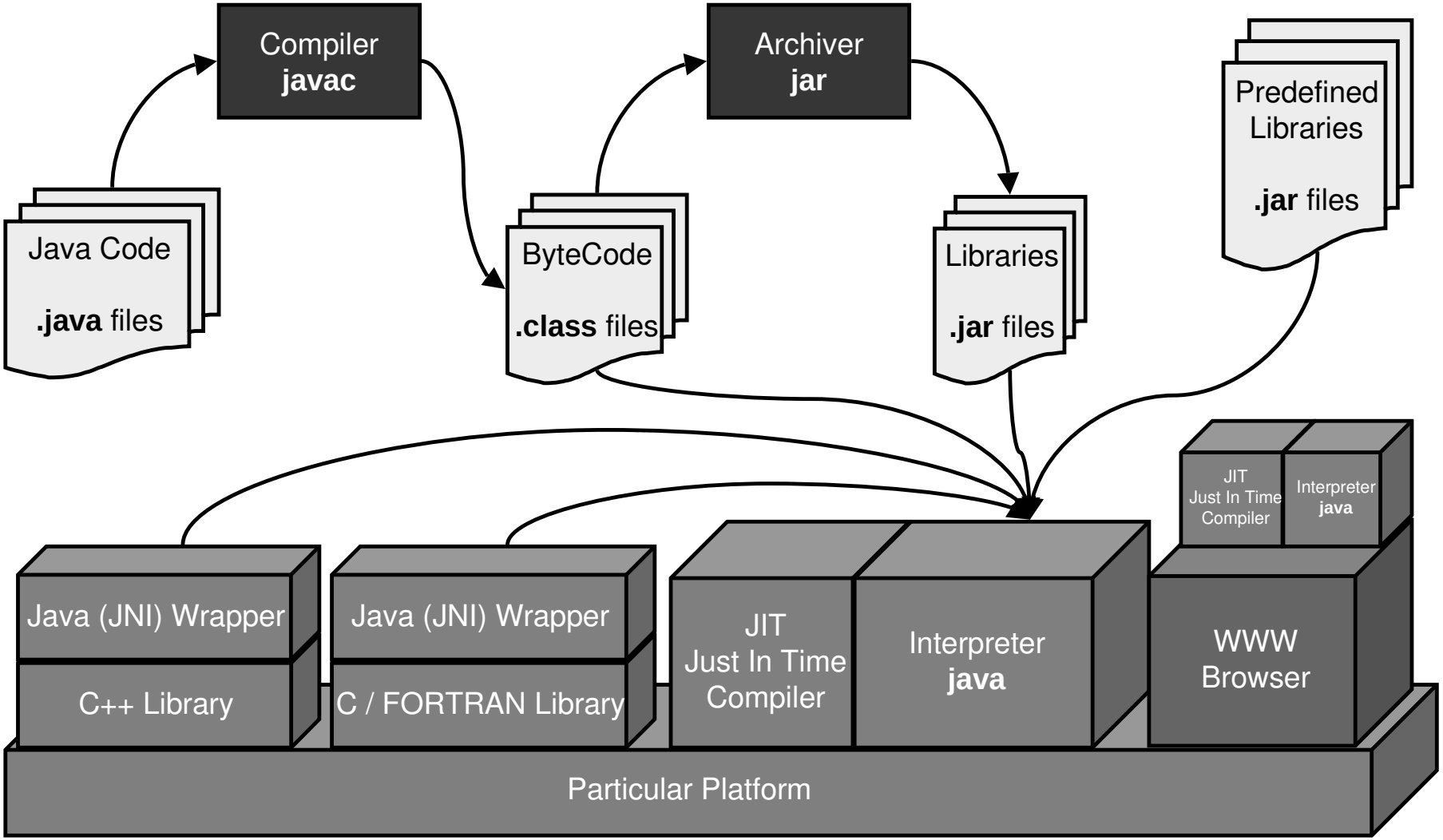
Remote Programming



Serialization and Deserialization



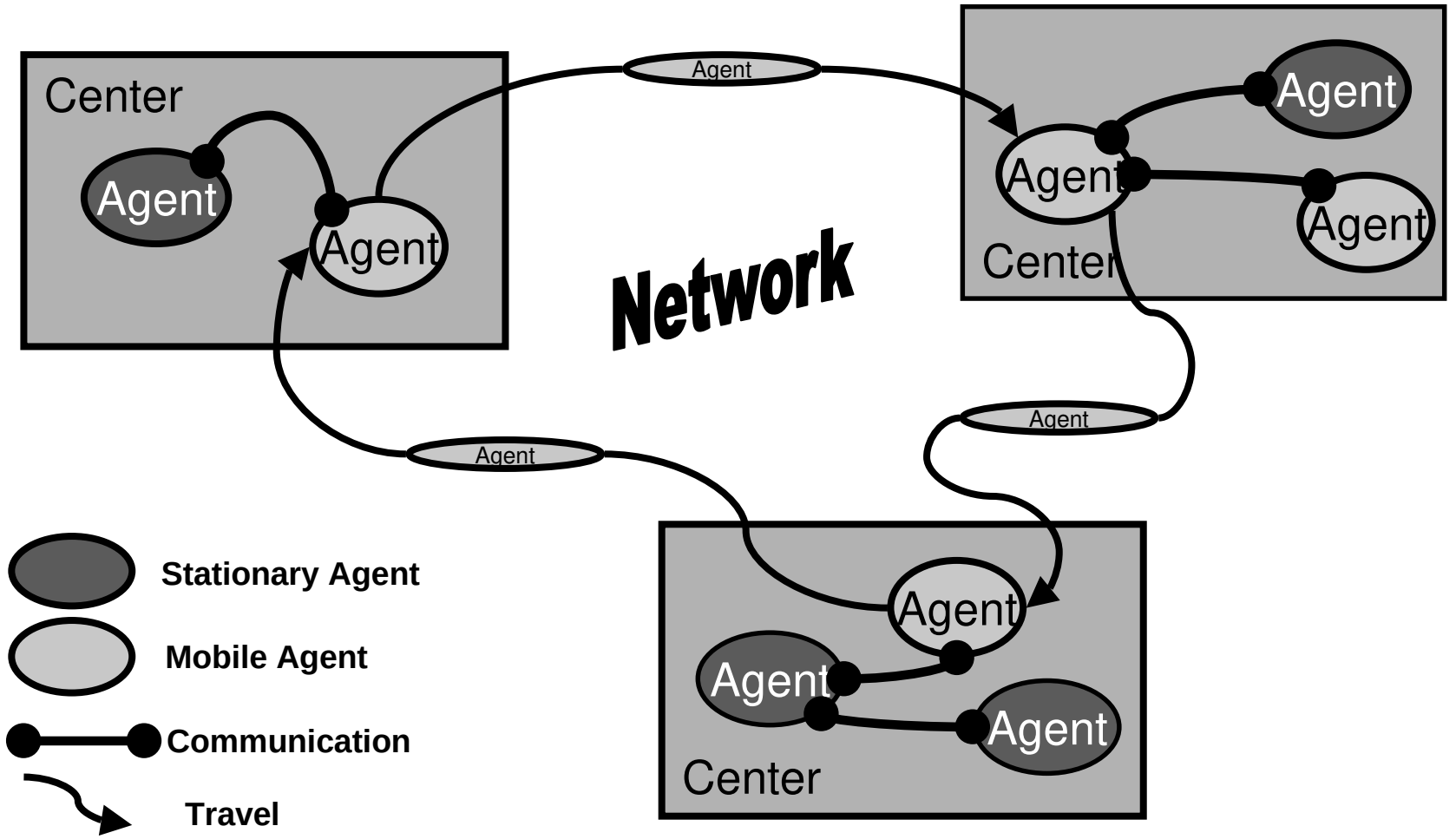
The Java System



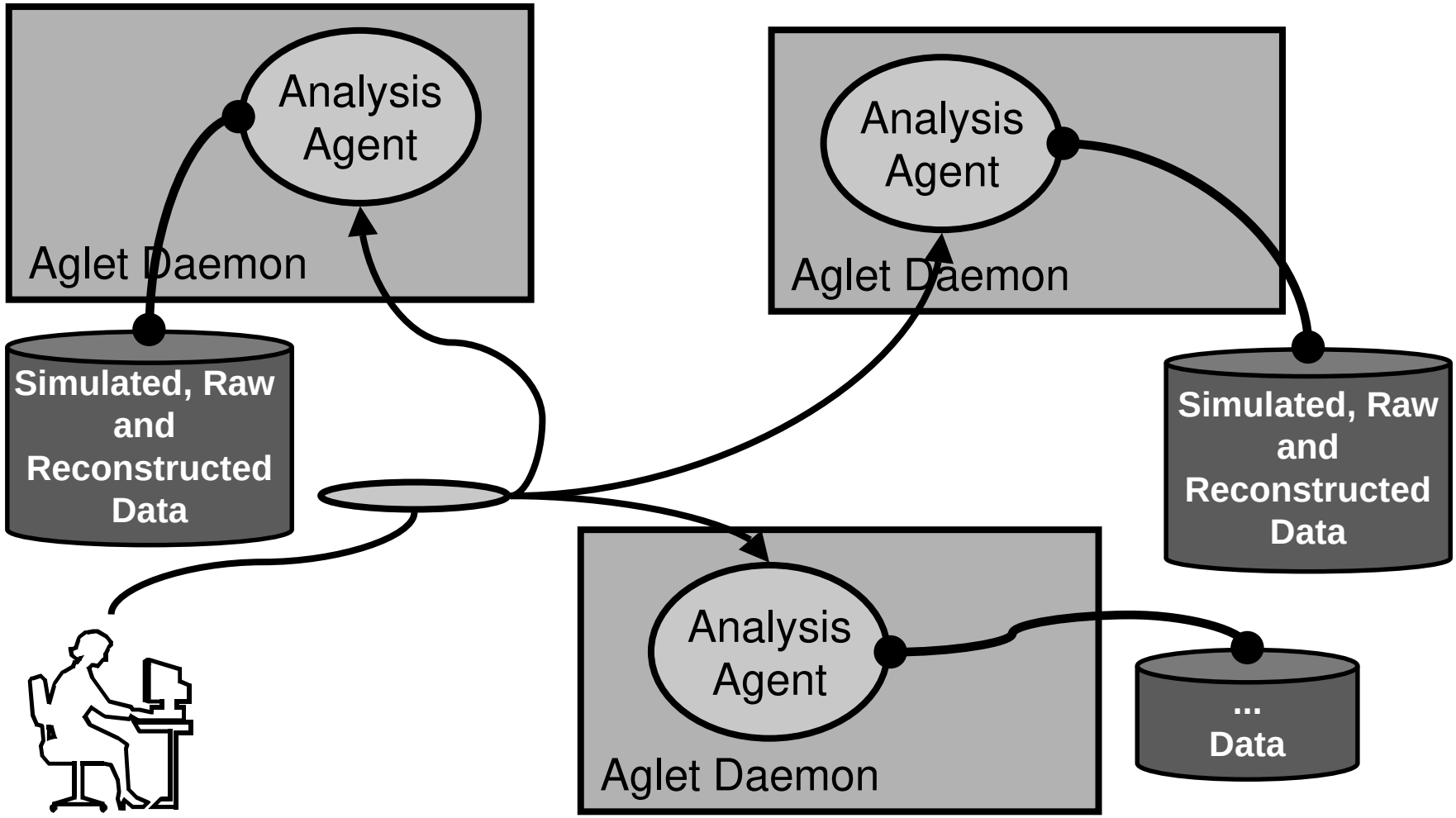
Java to implement Mobile Agents

- **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), ...

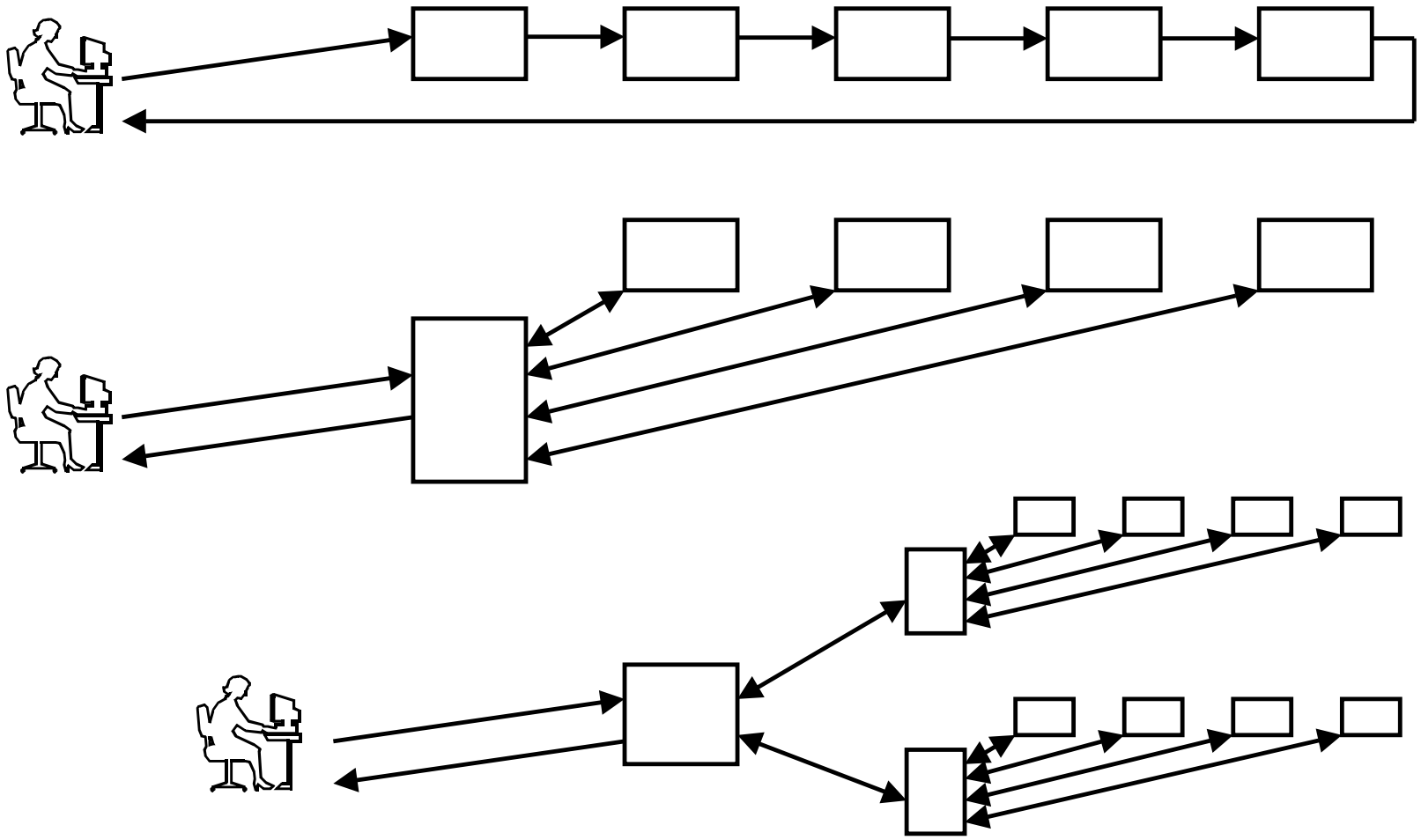
Mobile Agents and Stationary Agents



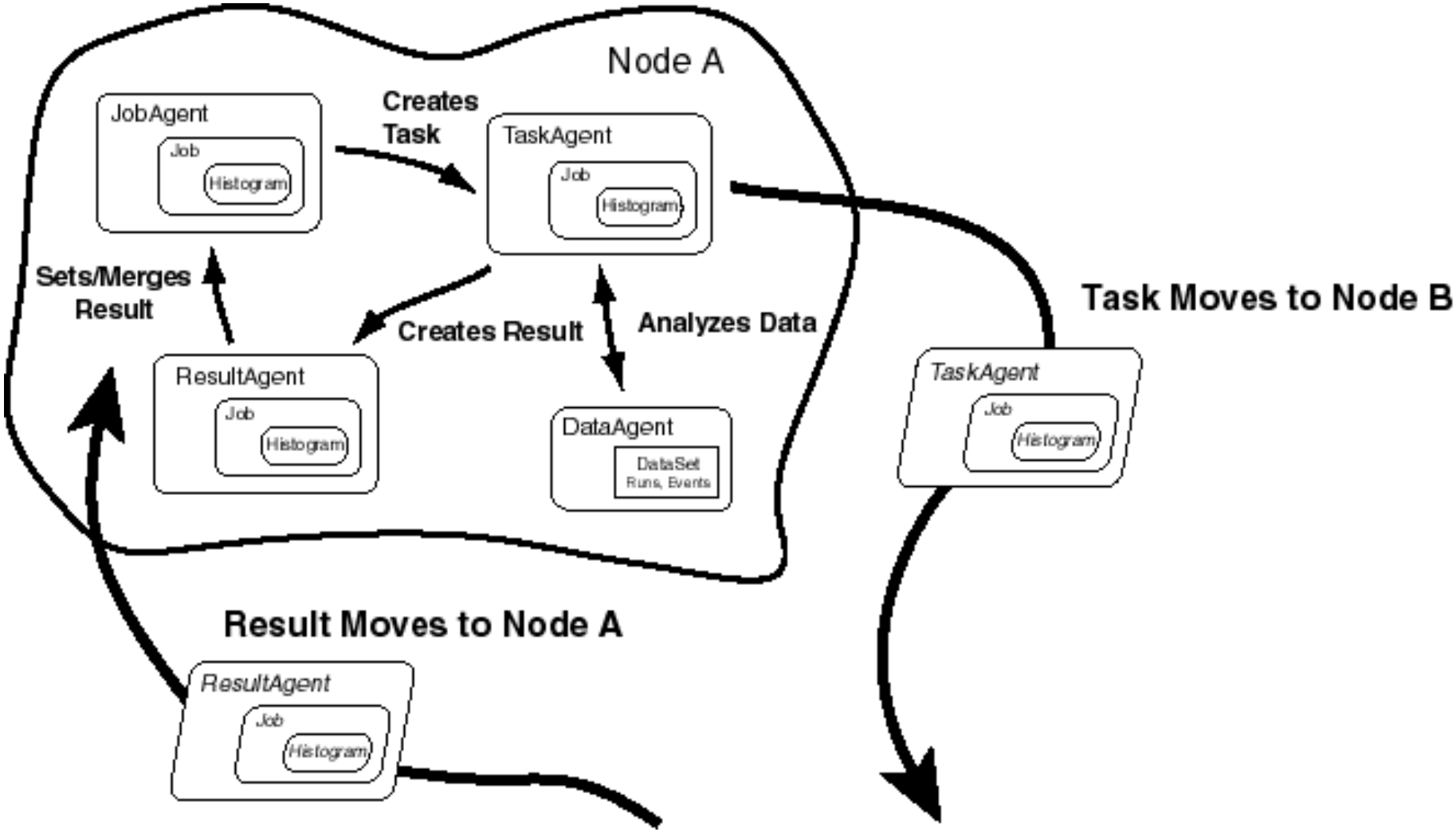
Data Analysis System using Mobile Agents



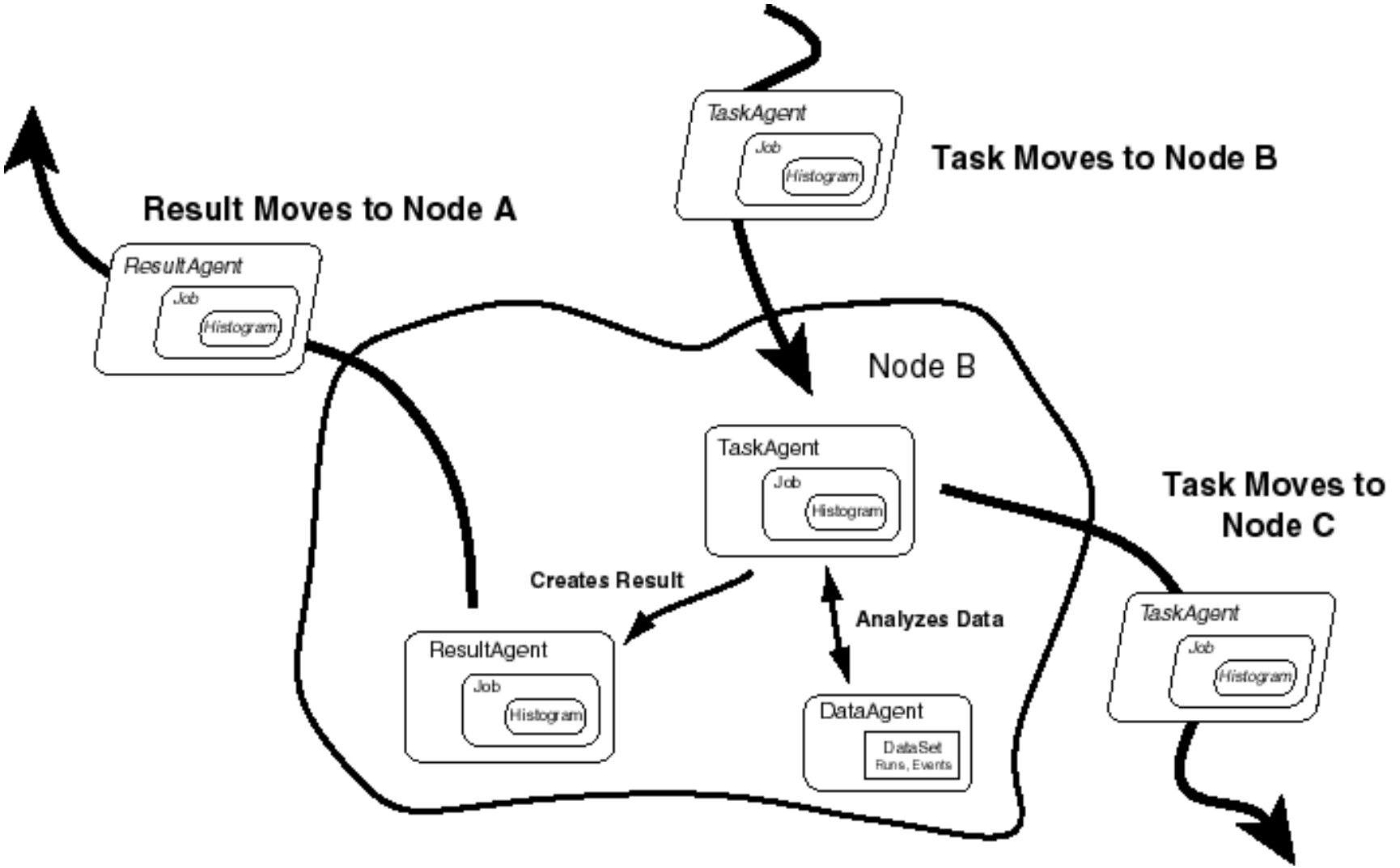
Topologies



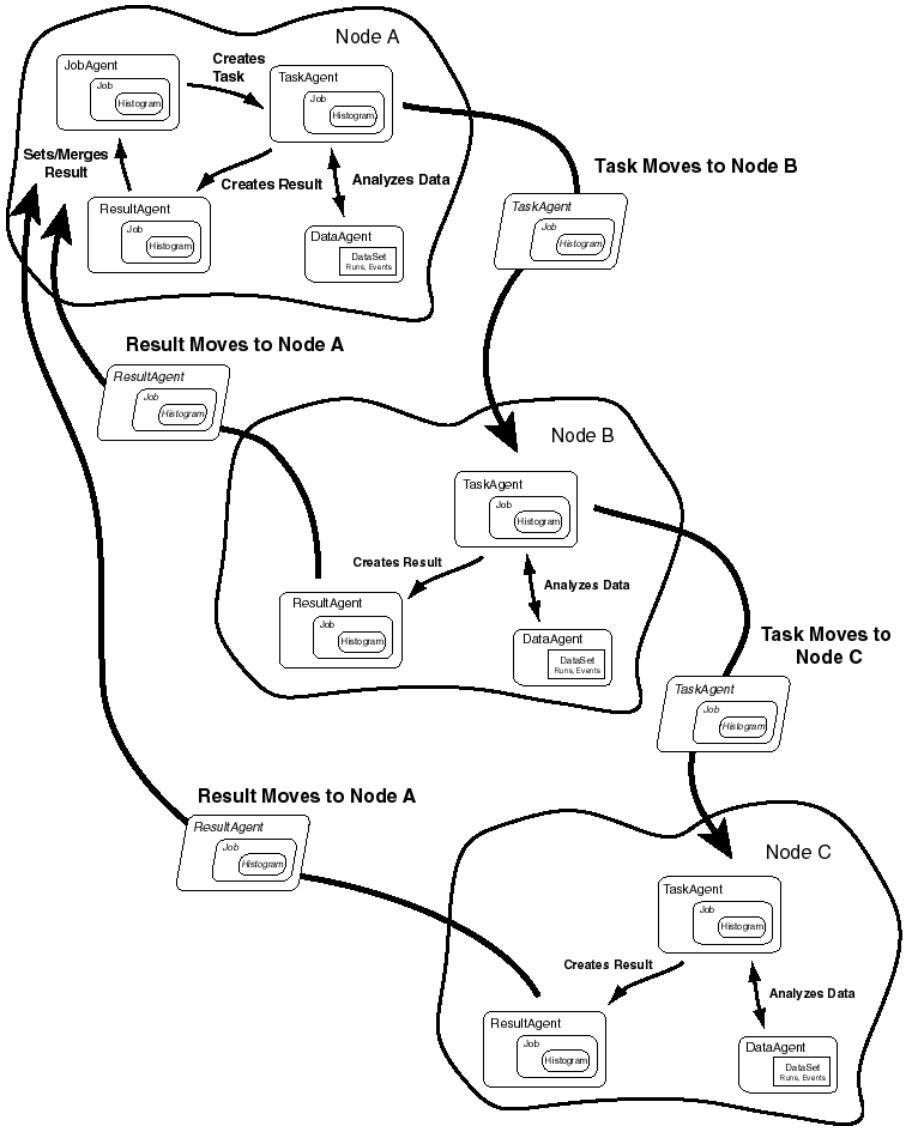
Submit, Run and move on...



… and Run and Move on…



… and finish.


Node A
JobAgent
Job
Histogram
Creates Task
TaskAgent
Job
Histogram
Sets/Merges Result
Creates Result
Analyzes Data
ResultAgent
Job
Histogram
DataAgent
DataSet
Runs, Events
Task Moves to Node B
TaskAgent
Job
Histogram
Result Moves to Node A
ResultAgent
Job
Histogram
Node B
TaskAgent
Job
Histogram
Creates Result
Analyzes Data
ResultAgent
Job
Histogram
DataAgent
DataSet
Runs, Events
Task Moves to Node C
TaskAgent
Job
Histogram
Result Moves to Node A
ResultAgent
Job
Histogram
Node C
TaskAgent
Job
Histogram
Creates Result
Analyzes Data
ResultAgent
Job
Histogram
DataAgent
DataSet
Runs, Events

CERN School of Computing

- **35 machines, each with 150 Mbyte data locally stored.**
- **Run a analysis job:**
  - creating a histogram (total jobsize 0.5 Mbyte)
  - analyzing and accessing all data 35 * 150 Mbyte ~ 5 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 Gbyte / 1Mbyte/s = 5000 seconds
  - Job transfer: 0.5 Mbyte * 35 / 1Mbyte/s = 17.5 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++