UPPSALA
UNIVERSITET

GRID Stream Database Management for Scientific Applications

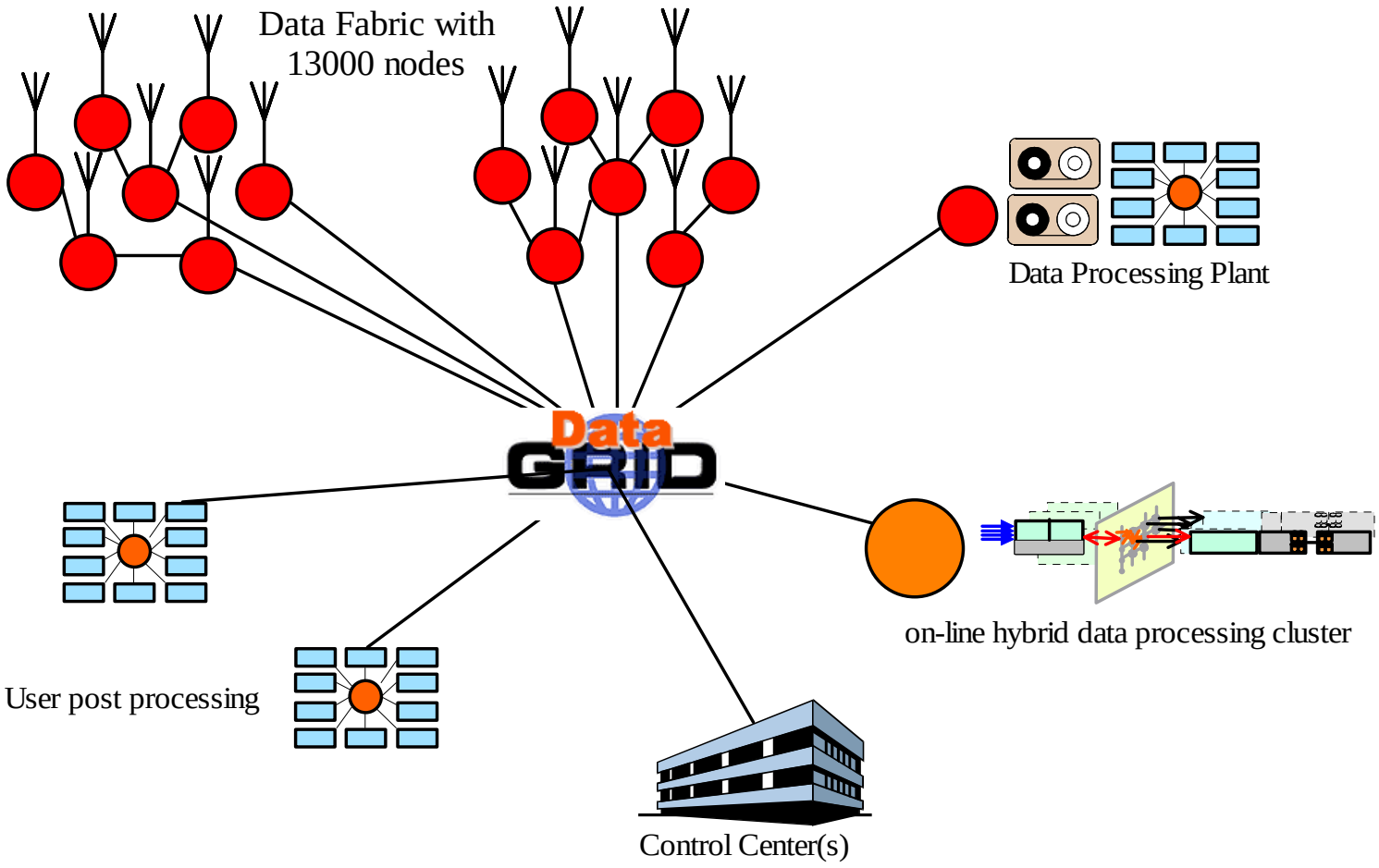
Milena Ivanova (Koparanova)
and Tore Risch

IT Department, Uppsala University, Sweden

Outline

- Motivation
- Stream Data Management
- Computational GRIDs
- GSDM Distributed Architecture
- Project status
- Next steps
- GSDM Demo

Motivation: LOFAR/LOIS Application



Motivation: LOFAR/LOIS Application

- Geographically distributed set of electromagnetic sensors and emitters
- Data products: Raw data streams (beams)
- Very high data volume and rate
- Complex numerical data
- Continuous queries: filtering, reduction, combining of data streams
- User-defined computational functions

➢ Problem:

High–performance processing of distributed continuous queries over many data streams. Utilizing GRID infrastructure for achieving high-performance

How stream data are different?

Streams peculiarities:

- Infinite
- Representation of substreams of limited size: *windows*
- *Continuous Queries (CQs)*
- Immediate processing of elements, followed by archiving or deletion
- Order preserving and non blocking processing
- Stream specific operations, e.g., windows join, moving average
- DBMS techniques for streams: approximate query answering, adaptivity, data reduction, multi-query optimization

GRID Computing

- Heterogeneous *sets of clusters* of computers
- For applications with much *need for CPU cycles*
- Toolkits (e.g. GLOBUS, CONDOR-G) provide *transparent* access to GRID clusters
- *Batch* processing
  (upload – compute – download)
- High data volume *bottleneck*

GSDM as a Computational GRIDs Application

High data flow rate and large data volumes require high performance

- Parallelism and distribution

Varying computer resource needs because of varying number of queries and streams, varying stream rates

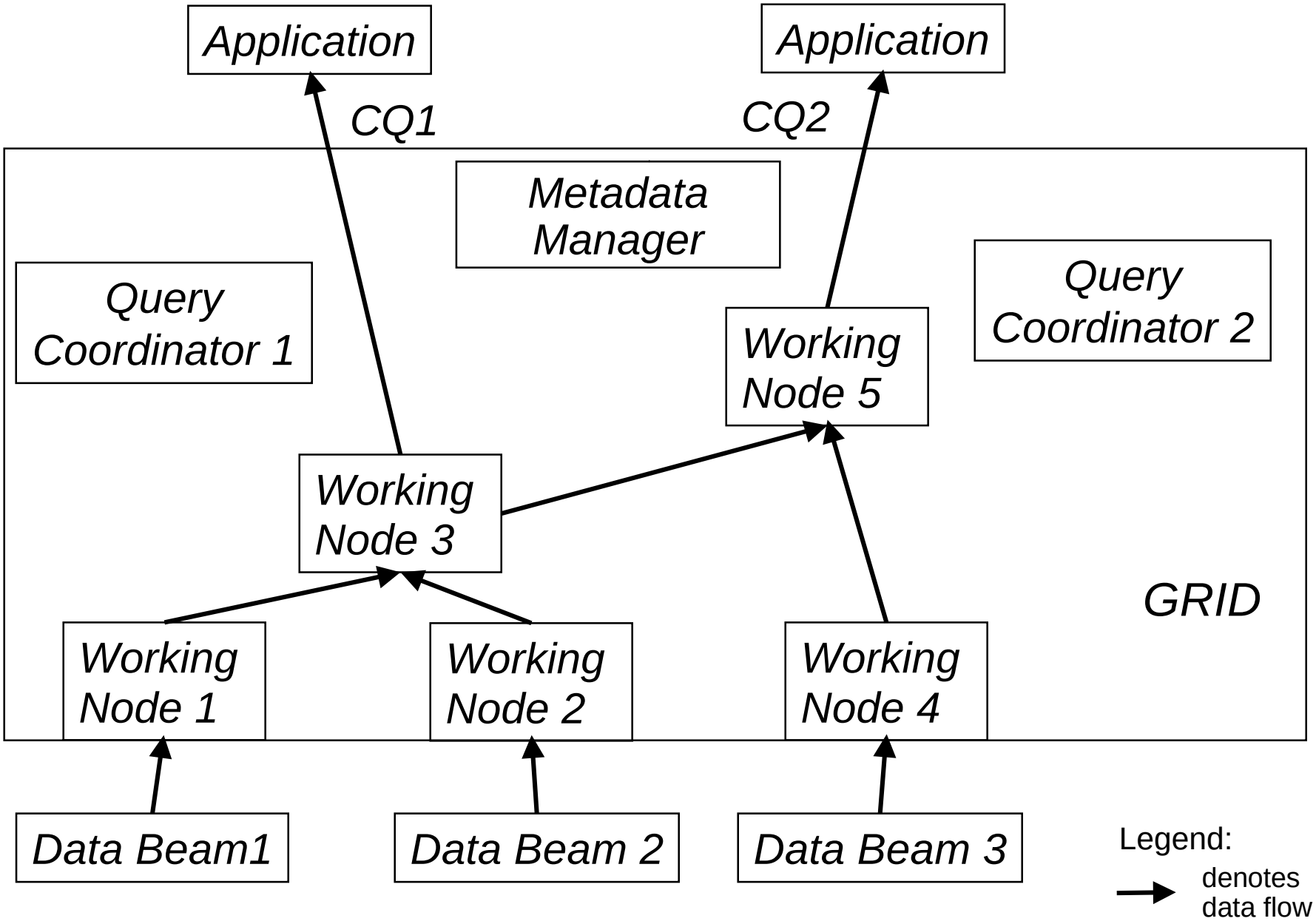
- Dynamic resource allocation

↳Computational GRID infrastructure

GSDM: GRID Stream Database Manager for Scientific Data

- Approach
  - Distributed and parallel
  - Stream-oriented
  - Main-memory Object-Relational DBMS
  - Utilization of GRID infrastructure for achieving high-performance

GSDM Scenario



GSDM Distributed Architecture

Four types of nodes

- Metadata manager
- Coordinator server
- Working node
- Application

Metadata and Coordinator Server

Functions:

- store metadata about the system
- decompose, install and activate CQs
- assign queries to working nodes
- start and kill working nodes
- coordinate processing at working nodes

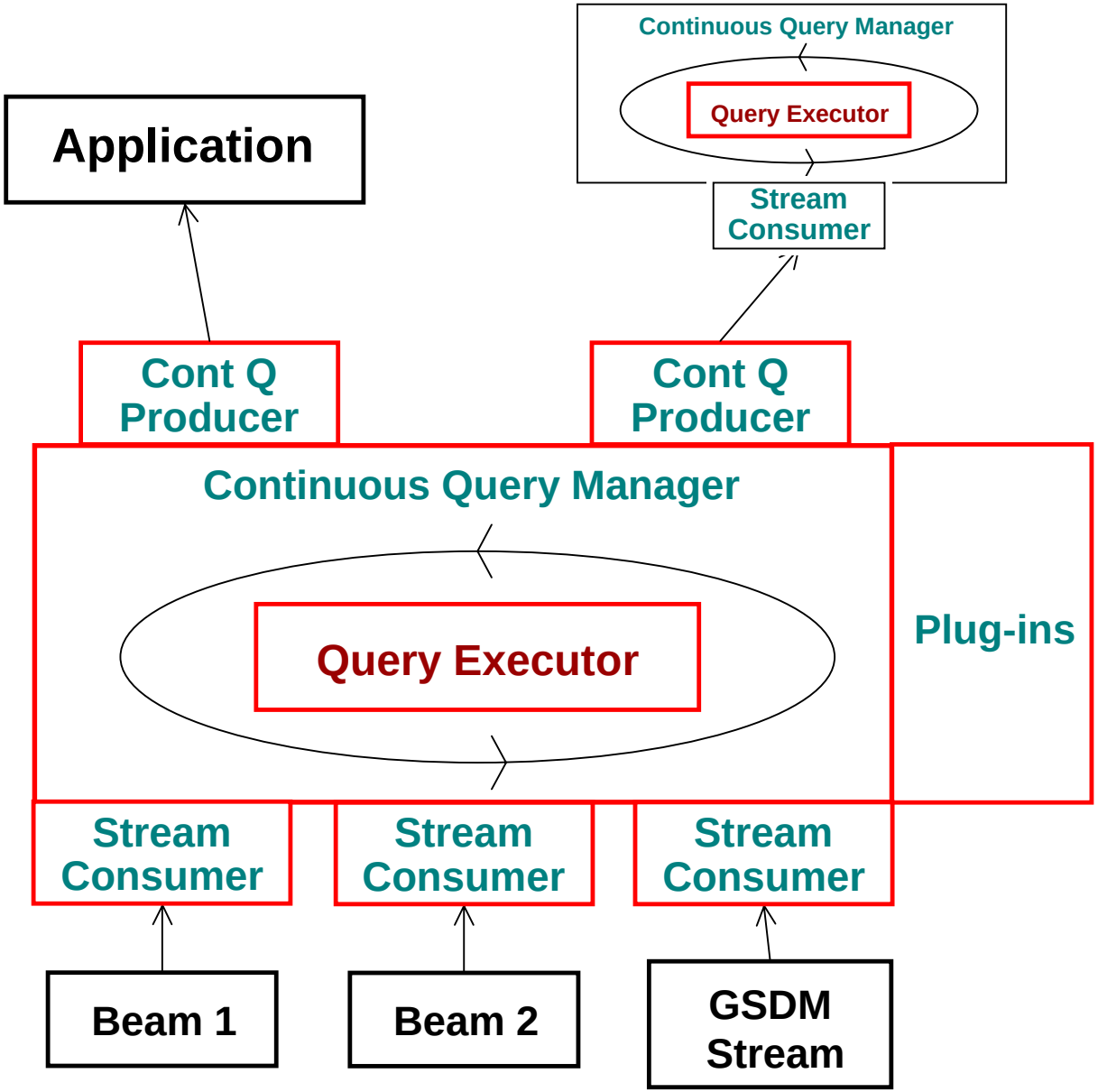
System Metadata

- Type GSDM
  - name(GSDM) -> Charstring
  - working_node(GSDM) -> Boolean
  - coordinator(GSDM) -> Boolean
  - cur_load(GSDM) -> Real
- Type Query
  - qid(Query)->Charstring
  - query_string(Query)-> Charstring
  - producers(Query) -> Bag of GSDM
  - consumers(Query) -> Bag of GSDM
  - installed(Query,GSDM) -> Boolean
  - active (Query,GSDM)-> Boolean

Working Node

- Function: to process continuous queries over streams
- List of active CQ
- CQ Execution loop
  - *Stream Consumer* receives pushed data and incoming commands
  - scheduler picks next CQ ready for execution
  - executes the CQ over current stream windows
  - *Continuous Query Producer* sends streamed result to the consumer (application or working node)

GSDM Working Node Architecture



GSDM Working Node

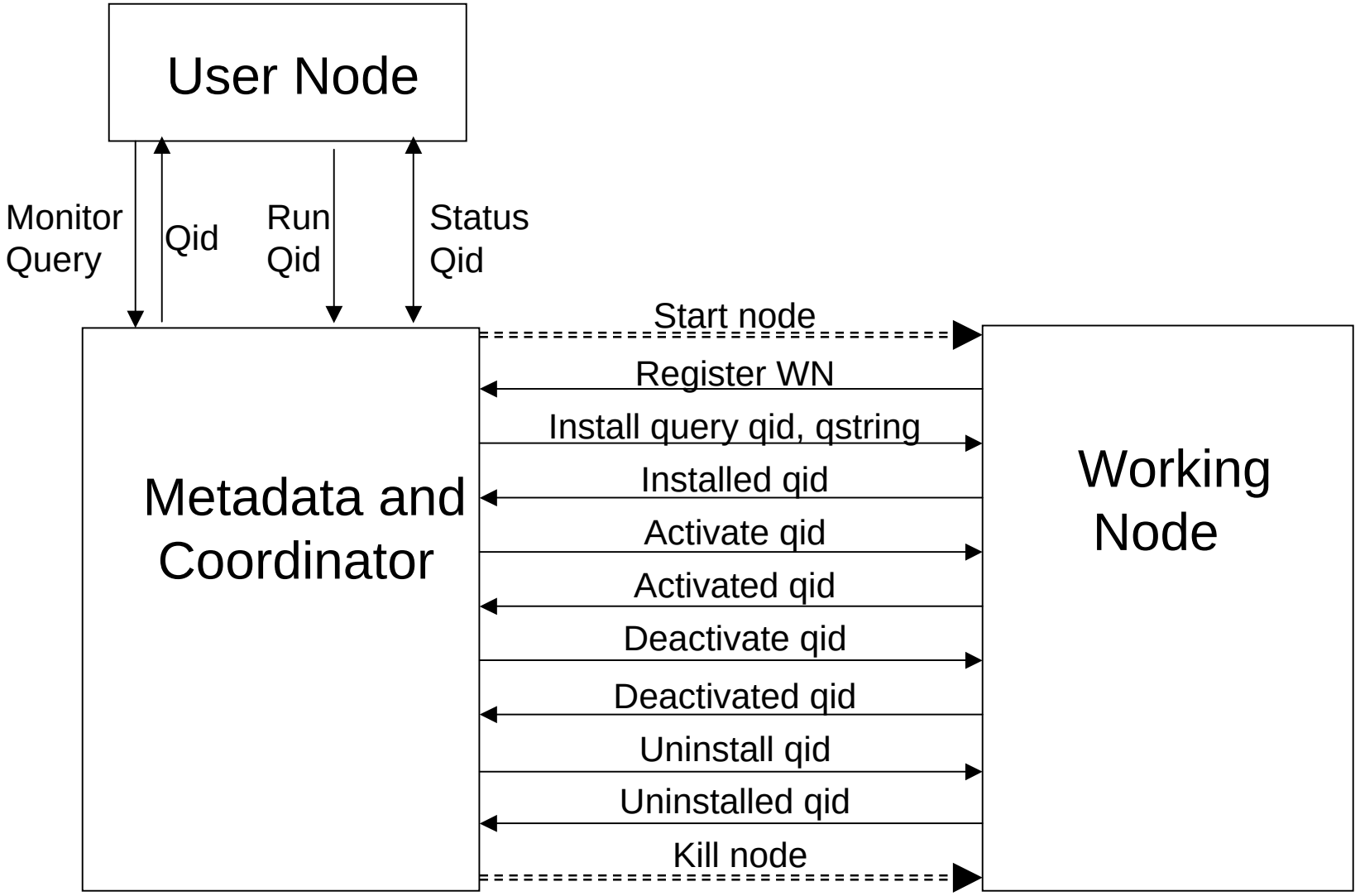
Data sources

GSDM Command Line User Interface

- Commands sent from application
- Executed at coordinator

- monitor(Charstring qstring) -> Charstring qid;
- run(Charstring qid) -> Boolean;
- stop(Charstring qid) -> Boolean;
- status(Charstring qid) -> <Charstring qstring, Charstring prodname>;

GSDM Communication Primitives



Current status

- GSDM prototype architecture
- Initial GSDM prototype implementation
- Formulation of continuous queries with user-defined computational functions from space physics
- UDP receiver of radio on the Internet
- Data beam simulator

Related Work in Stream DBMS

- Relatively low stream rates, centralized architectures
- Relational representations and relational operations modified for stream processing: selection, join, aggregations

Our system

- High rates of LOIS/LOFAR streams
- Distributed architecture
- User-defined functions over non-relational data representations

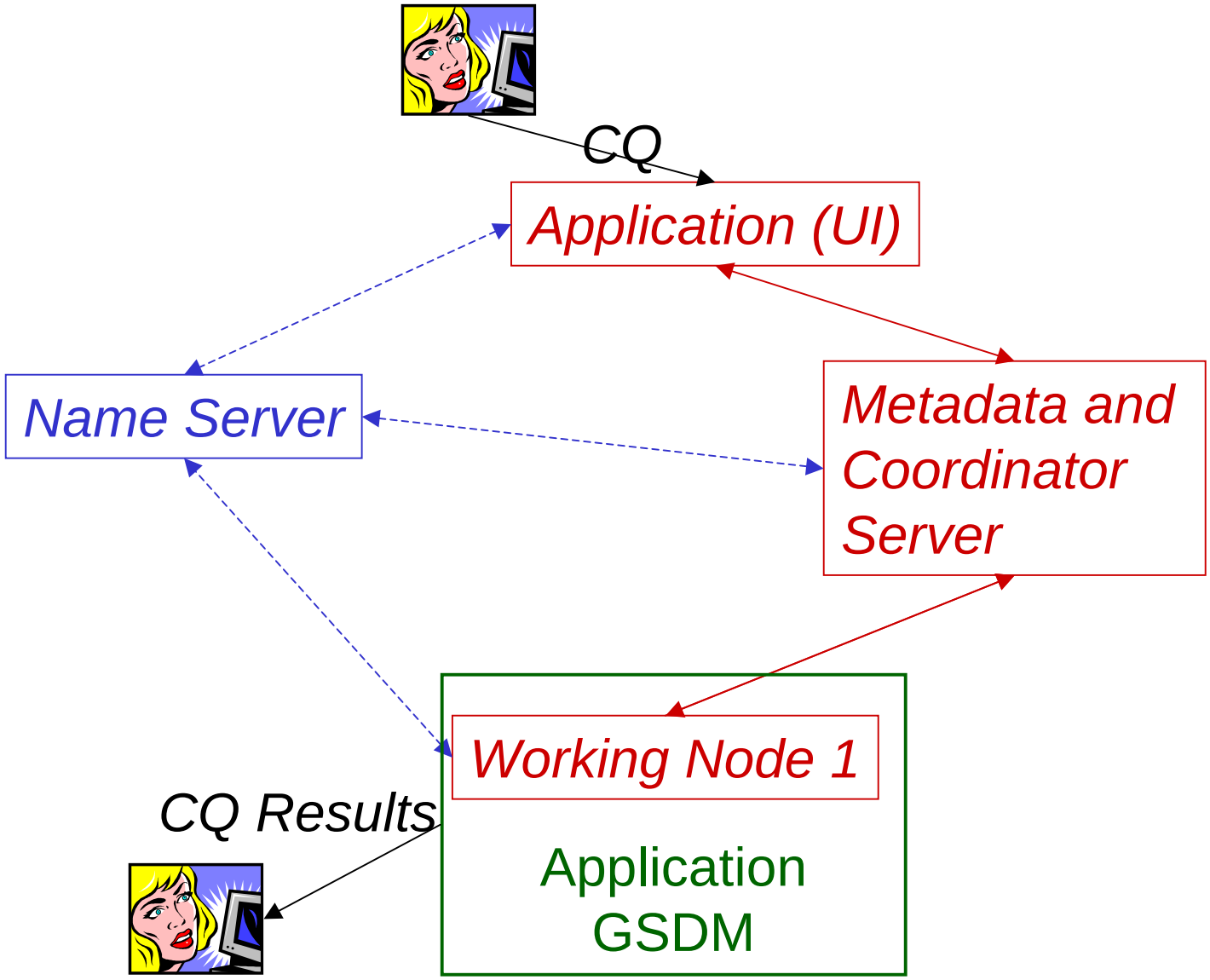
GSDM requirements for GRID

- Interactive job: users can install and stop CQs interactively
- Accessibility & security issues
  - Data delivery directly to the working nodes, e.g. inside of clusters
  - High-performance communication between GSDM servers running on different GRID resources

Next steps

- Evaluation of initial prototype implementation and development
- Optimization of distributed stream database queries
- Utilize GRID infrastructure functionality for resource brokering and management
- New GRID services for streaming data

GSDM Demo Scenario



Demo scenario

- Database schema
  - start_udp_consumer(char s) -> integer;
  - stop_udp_consumer() -> integer;
  - change_udp_scale(real i) -> integer;
  - get_udp_packet() ->
    < vector of complex, vector of complex, vector of complex >;
- Example CQ

```
create function visualized_fft()->boolean
  as select repaint(v,6.0)
  from vector of real v, vector of complex x,
      vector of complex y, vector of complex z
  where v= vect_log_magnitude(fft(windowed_fft(x)))
      and get_udp_packet()= <x,y,z>;
```