Bundesministerium
für Bildung
und Forschung

Data Management within the D-Grid HEP Community Project

Michael Ernst
DESY

HEPCG Workshop April 2006

Credits

- DESY Hamburg: M. E., P. Fuhrmann, M. Radicke
- FZK: D. Ressmann
- NIC / DESY Zeuthen: D. Melkumyan, P. Wegner, D. Pleiter
- Uni Dortmund: R. Yahyapour, L. Schley, S. Freitag
  C. Wissing
- Uni Freiburg: K. Desch, P. Wienemann,
  T. Potjans
- ZIB Berlin: H. Stueben, S. Wollny

Work Package 1: Data Management

1. **Development and Distribution of a *Scalable Storage Element* for the specific requirements of the High Energy Physics Community based on Standard Grid Interfaces**

   (DESY, Unis Dortmund und Freiburg, FZK)

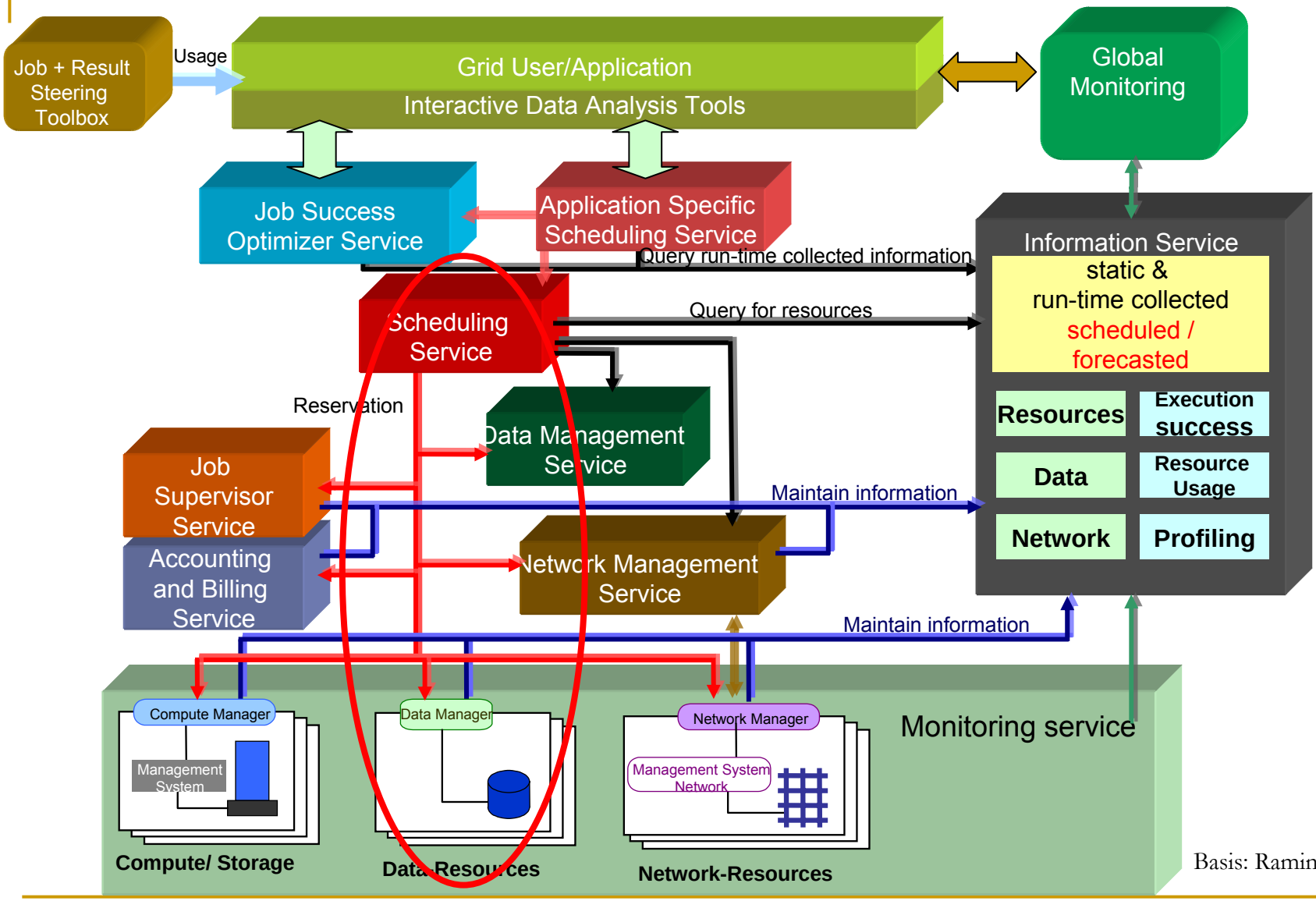
2. ***Optimization of Job Scheduling in Data Intensive Applications***

   (Uni-Dortmund (CEI und Physik), DESY – Synergy with C3 Grid)

3. **Development of an *Extensible Metadata Catalog for Semantic Data Access***

   (DESY, ZIB, HUB, NIC – Interest shown by AstroGrid)

Architecture and Services of the HEP Community Grid



Basis: Ramin Yahyapour

Optimized Job Scheduling – Project Goals

Current situation:

- Allocation of jobs vs. sites incorporates little information about data availability
  - E.g., files may be on tertiary storage → staging needed before access (time consuming)
- Storage element does not provide advanced planning features

- More precise planning and prediction of file availability prior to job allocation to compute elements
- Better integration of local job and data scheduling to improve response times and throughput

Requirements

- Co-scheduling of jobs and data
- Job types that can be distinguished
  - *CPU intensive algorithms use small input data sets to produce large sets of simulation data:*
    - Due to the high computational resource consumption the resulting data sets are valuable
    - Resulting data sets have to be distributed → coherence?
  - *Reconstruction / Reprocessing:*
    - Access to the simulation data is needed
    - Use of CPU intensive algorithms
  - *Analyze jobs:*
    - Quantity and local occurrence are not predictable
    - Use of data sets from experiments and/or simulations
    - Variable in CPU utilization and resulting data sets

Development

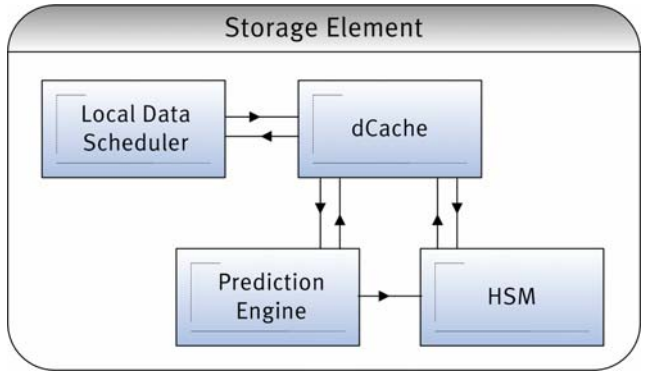
- Work will be based on Components used in LCG
  - Initially, LCG 2.X software
  - As soon as sensible, switch to web service based gLite (target middleware)
- Extensions to existing dCache system
- **Desirable:** cooperation, discussion and maybe integration of our (future) research and corresponding results into LCG

Extended Scheduling System

- System makes use of the local schedulers in the extended CEs and SEs
  - Add SE interface to communicate about future data/storage requests
  - Add CE interface to communicate about future job executions
- If replication has to be accomplished a corresponding replication scheduler will be invoked.
- **Desirable**: Possible addition to current WMS development

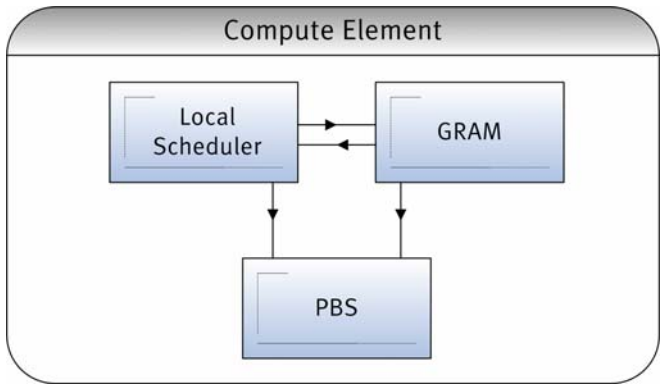
Extension of SE

- Due to the use of HSM, files appear “available” on a local file system → essentially located on tertiary storage
  - Restaging on demand can take a long period of time
    →ready to run jobs have to stay idle until the recovery of prerequisite data
  - Restaging time is difficult to predict
- Ext. of the SE (dCache & HSM) by a prediction engine
  - Estimation of restoration times: When will a file be accessible?
- Ext. of the SE by a local data scheduler (LDS) to a schedulable resource
  - LDS has to deal with space reservation (dCache support)
  - LDS communicates with dCache: When is a file accessible?
  - dCache and LDS need agreement mechanisms



Extension of CE

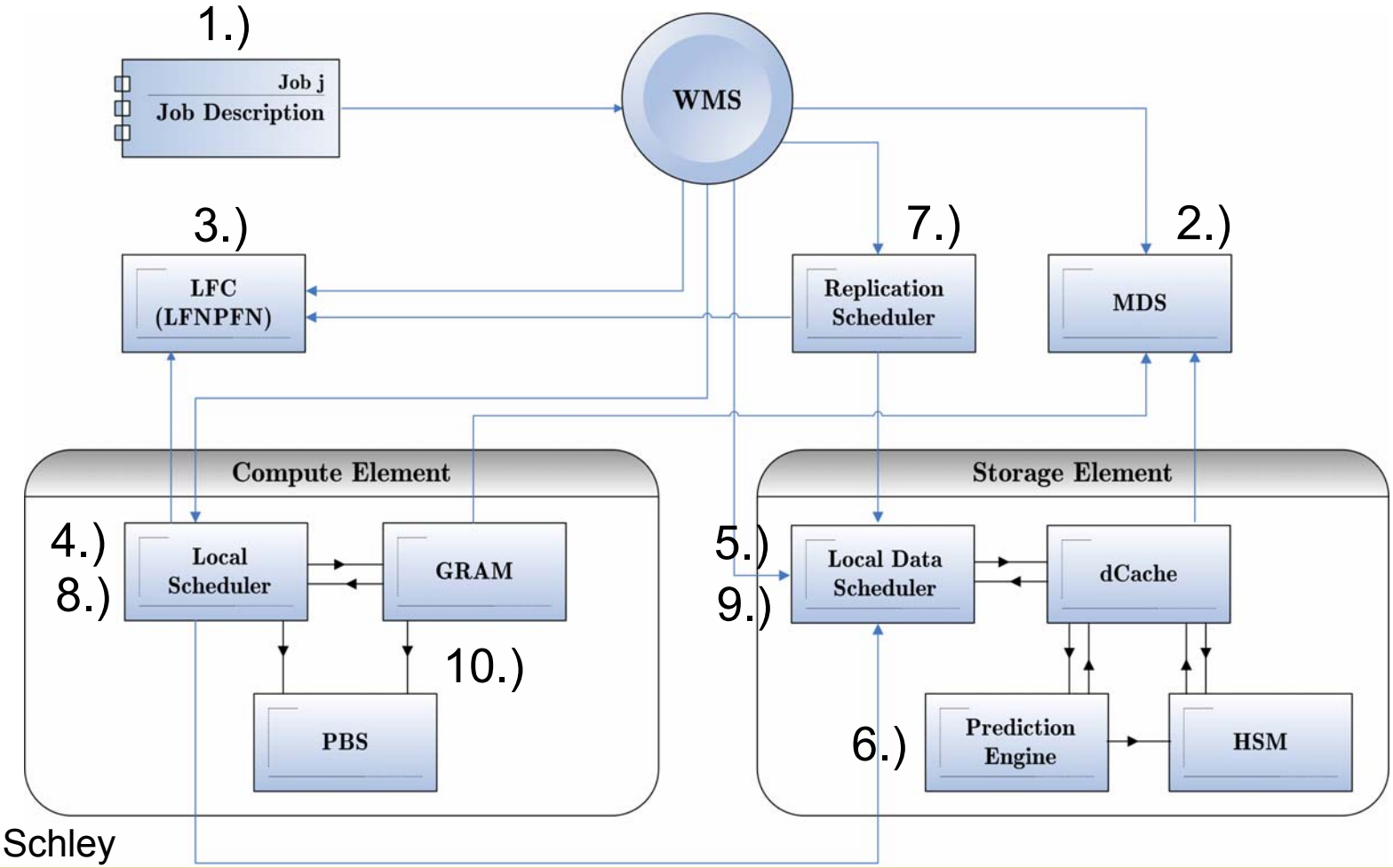
- Extension of the CE by a local scheduler to a schedulable resource
- Functions of the local scheduler:
  - Query of the dynamic state: When can the CE execute the next job?
  - Agreement mechanism for the execution of jobs
  - Interaction with SE: Is the required data finally available on the SE?
- Architecture is based on the current CE
- Torque controlled by GRAM



Central Scheduling System

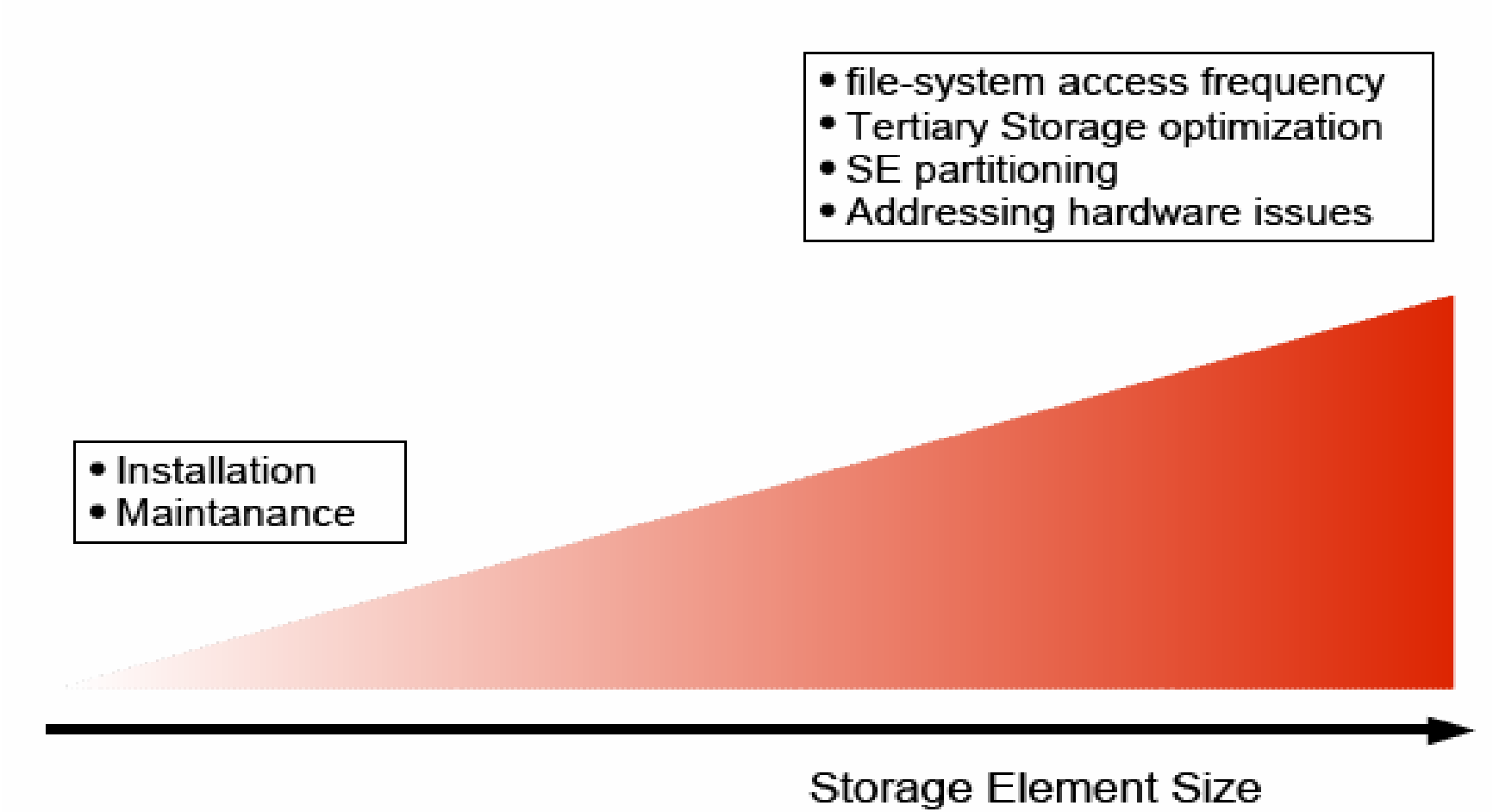
- Receives jobs from the user (JDL)
- Interacts with the Monitoring and Discovery Service (MDS) to find resources
  - dependent on the submitted JDL
- Interacts with a File Catalog (LFC) to translate LFNs, PFNs, and GUIDs
  - Where are the files located?
- System makes use of the local schedulers in the extended CEs and SEs
  - Which is the earliest start time for a job?
  - Has file replication to take place?
  - Are the files during execution time immediately accessible?
- If replication has to be accomplished a replication scheduler will be invoked

Planned Architecture

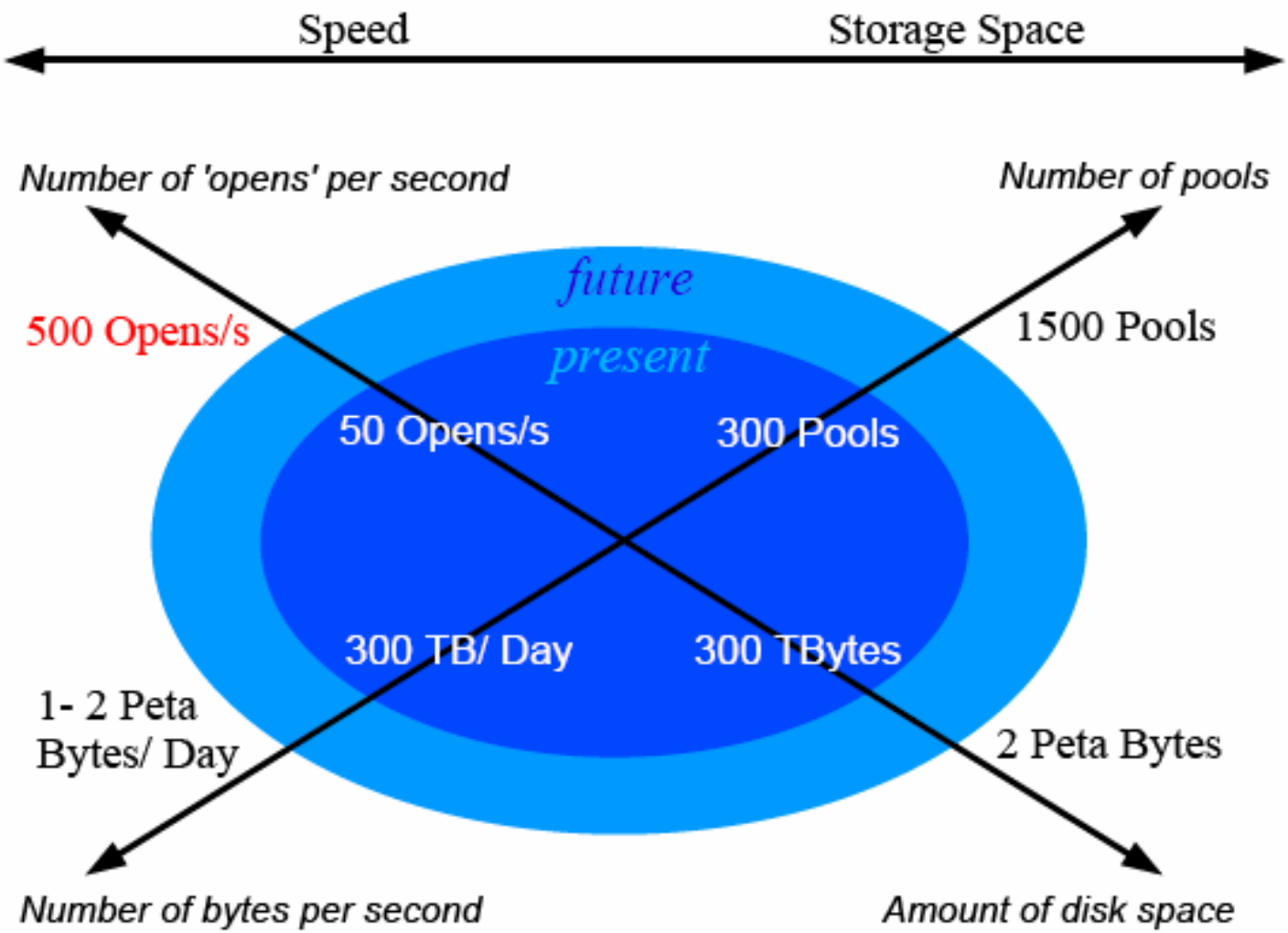


L. Schley

Issues related to SE scaling



SE scaling in the near future



P. Fuhrmann

Co-scheduling of jobs and data

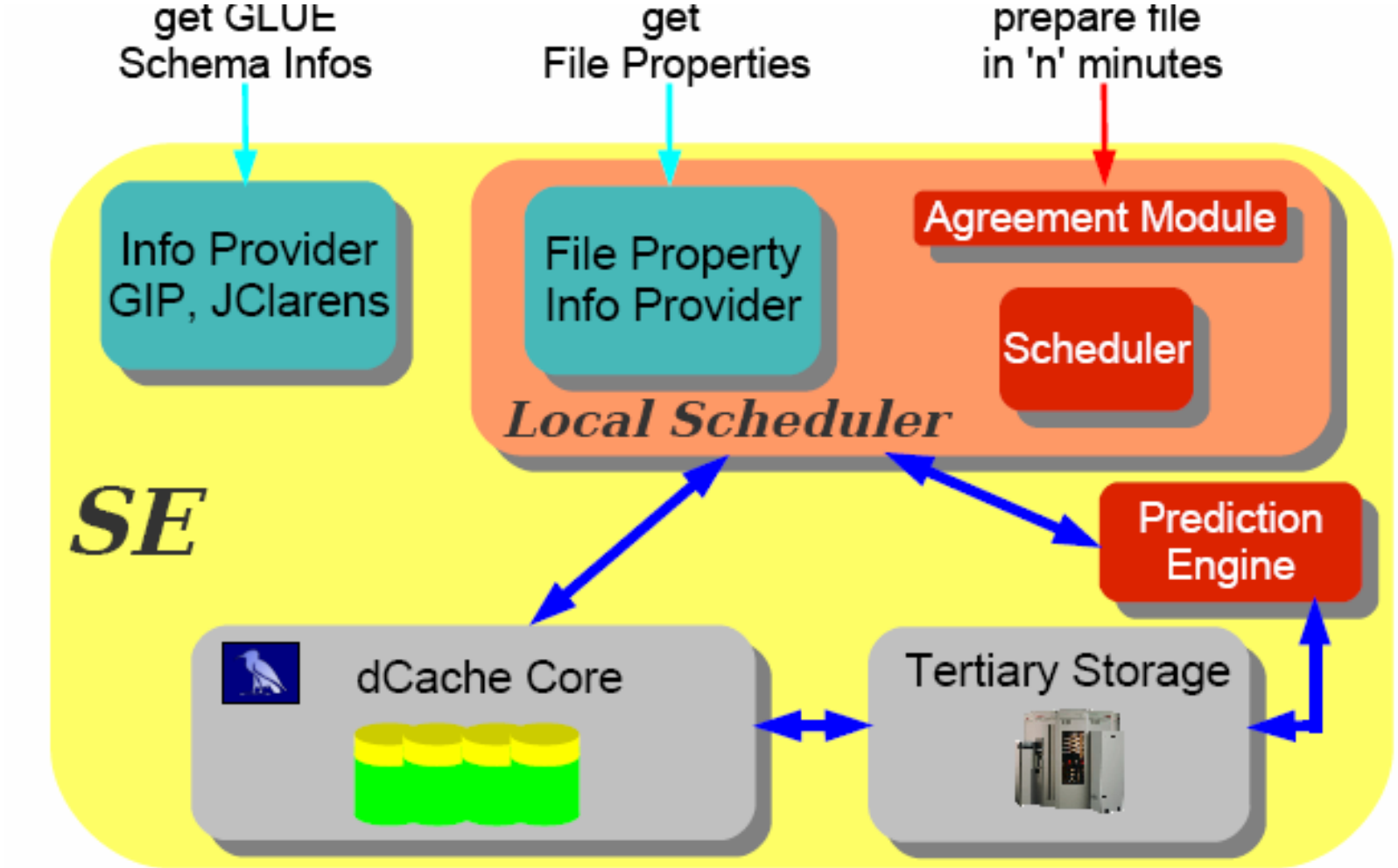
- More precise information provided by SE's will help the Resource Broker to improve matchmaking
  - RB is querying SE for specific file metadata using File Property Information Provider

Improvements: Information Service

- Information Provider (SE-specific)
  - dCache gets grid-enabled by publishing its status according to the GLUE schema
  - already deployed in LCG and OSG installations
- File Property Infoprovider (file-specific)
  - Extended Information Service, WS-based
  - File is cached/ on tape
  - time to get file ready for transfer

→ important prerequisite for co-scheduling

Extended SE overview



P. Fuhrmann

ILDG Initiative

- The **International Lattice DataGrid** was proposed 2001.
- Aim:
  - Longterm storage and global sharing of gauge configurations within a Datagrid
  - **Make more efficient use of expensive data**
- **Participants:** Australia, France, Germany, Italy, Japan, UK, USA
- **Working groups:**
  - Metadata working group
  - Middleware working group

- http://www.lqcd.org/ildg

Requirements

- **Sharing gauge configurations requires**
  - Semantic access to worldwide distributed data
  - Standardised **metadata**
- XML documents which conform to a **XML schema**
- **Extensible** schema required
  - Standards on **binary file format**
  - Definition of common **middleware interfaces**
- ILDG is planned to be a **grid-of-grids**

Requirements

- Load/store/query of XML documents which conform to **extensible schema**
- Access via **web service** front-end
- Standard **relational database** as back-end
- Usable for other research communities

Strategy

- Issues to be addressed:
  - XML-Java binding
  - Java object content persistence

MDC Front-End

- **Web services** to query and download documents standardized by ILDG

  (Additional services for LatFor DataGrid)
- Read access: open
- Write access: **GSI** based authentication
- Used software:
  - Tomcat5 + Axis
  - gLite trustmanager

Evaluation

- Chosen solution meets requirements
  - Support for extensible schema
  - But: XML schema specification not yet fully supported
- Usable for other research communities
- Flexible front-end
- SQL servers provide standard, well-supported back-end technology
- Fast queries for simple elements
- Performance issues: materialization of XML IDs expensive:
  - Loading requires O(0.02) seconds per XML ID
  - Storing requires O(0.04) seconds per XML ID

Authorization / Access Control

- MDC stores **for each ensemble** permissions for
  - modifying metadata
  - modifying data files (configurations)
  - downloading data files
- Read or write permissions are assigned to groups
- Project (=owner) administrators can
  - create and modify groups
  - modify access permissions
- ACL will be forwarded to file catalogue
  - Use ACL feature of LFC

LatFor Datagrid

- **Regional grid for groups in Italy, France, Germany**
  - Resource requirements:
  - O(100.000) configurations, O(10-100) TBytes
- **Infrastructure:**
- Information services
  - Metadata catalogue (DESY Z)
  - VOMS server (VO “ildg”) (DESY HH)
  - BDII, LFC file catalogue (DESY HH)
- Storage elements
  - Using **dCache** for SEs
  - SE at DESY (HH/Z), ZIB (Berlin), ZAM (Jülich)
  - Includes all sites with HPC for LQCD

User Client Software

- **LCG-2** based **Grid User Interface**
  - Compiled for several Linux flavours
  - Globus-2.4, LCG client data management tools
- Other client software Java/Perl based
- **RPM-based** installation mechanism for all client software
  - Installation in user space
  - No root rights required
  - Same installation mechanism for different Linux flavours
- GUI installations in: Germany, UK, France, Japan, Spain, Italy, Cyprus

Summary (1/2)

- WP1/ Data Management within HEP Community Project on track
  - Milestones reached so far
  - Some areas ahead of plan
  - dCache Scalable Storage Element as it currently exists is already used world-wide and an important component in LHC Computing
  - Have met the INFN/EGEE Workload Management System Developers to investigate collaboration potential and find a non-intrusive way to integrate the extended scheduling mechanism

Summary (2/2)

- WP1/ Data Management within HEP Community Project on track
  - Goals as outlined in project plan will – according to current understanding - be met
  - No changes to project plan
    - Concepts and Strategies worked out and are well aligned with expectations expressed by community
  - Components developed are expected to be used as part of the DGI infrastructure and the international LHC computing environment