Managed by

European Research Consortium for Informatics and Mathematics ERCIM

Scalable Peer-Group Services in Grids

Vladimir Vlassov (KTH), Cosmin Arad (SICS), Seif Haridi (SICS)
vladv@kth.se, cosmin@sics.se, seif@sics.se

Bridging Global Computing with Grid (BIGG), Nov 2006

Information Society Technologies

Outline

Grid and Distributed Systems Research at KTH, SICS, Stockholm, Sweden

- Democratization of the Grid
- Self-Management for Large-Scale Distributed Systems
- GODS: Global Observatory for Distributed Systems

Scalable Peer-Group Services in Grids

- Use of P2P to improve scalability, availability and performance of data services – Peer Group (Data) Service
- Use cases (examples)
  - A P2P (overlay) replica management service
  - A P2P distributed file system (KESO)
  - A P2P resource discovery
  - A P2P distributed back-up storage (MyriadStore)
  - A P2P content delivery network (DOH)

Democratization of the Grid

EU STREP Grid4All: Self-* Grid: Dynamic Virtual Organizations for Schools, Families, and All

- Democratization of the Grid for ordinary people, small organizations, SMEs, families and schools
  - IT-inexperienced users
  - Dynamicity: highly dynamic Grids for highly dynamic VOs
  - Focus on a dynamic and semi-open infrastructure where resources are provided mainly by the community itself but also, upon need by commercial utility-computing centers
- Incorporate P2P techniques and semantics driven approaches in Grid architecture
  - to provide self-management, scalability, dynamicity and heterogeneity support
- Self-management
  - Component-based, loopback control
  - P2P techniques to manage collections of …

Self-Management for Distributed Systems

Motivation for self-management

- High complexity, large-scale
- High dynamicity
- A combination of the above

STREP SELFMAN: Self Management for Large-Scale Distributed Systems based on Structured Overlay Networks and Components

- Goal: a service architecture that is a framework for building large-scale self-managing distributed applications.
- Objectives for the self-management abilities:
  - Self configuration: reconfigure itself during execution;
  - Self healing: continued execution (service, SLA) under failures;
  - Self tuning: load balancing and overload management;
  - Self protection:
- Feedback loops throughout the system
  - the detection of an anomaly
  - the calculation of a correction
  - the application of the correction
- System behavior should converge

GODS: Global Observatory for Distributed Systems

Why GODS?

To set the bar for distributed applications development:

- Deployment in "real-world" testbed
- Tracing/Debugging algorithms
- Performance tuning
- Quality Assurance

Uses for Grids

- to study dynamicity / scalability / availability in Grids
- to evaluate scalable group services

The Cathedral and the Bazaar, by Eric S. Raymond:

- "Every good work of software starts by scratching a developer's personal itch"
- "To solve an interesting problem, start by finding a problem that is interesting to you"
- "Any tool should be useful in the expected way, but a truly great tool lends itself to uses you never expected"

Uses of GODS

Expected uses

- Deployment and evaluation of large-scale distributed systems
- WAN emulation with ModelNet
  - Topology
  - Link latency
  - Link bandwidth
  - Link packet loss
- Control and Monitorization
- Collecting and aggregating statistics

Unexpected uses

- Emulation of node arrivals, leaves and failures (churn)
- Emulation of network partitioning
- Dynamic change of link properties
- Bandwidth consumption measurements
- Statistical models, node groups
- Automated experiments

Uses of GODS (cont'd)

More Unexpected Uses

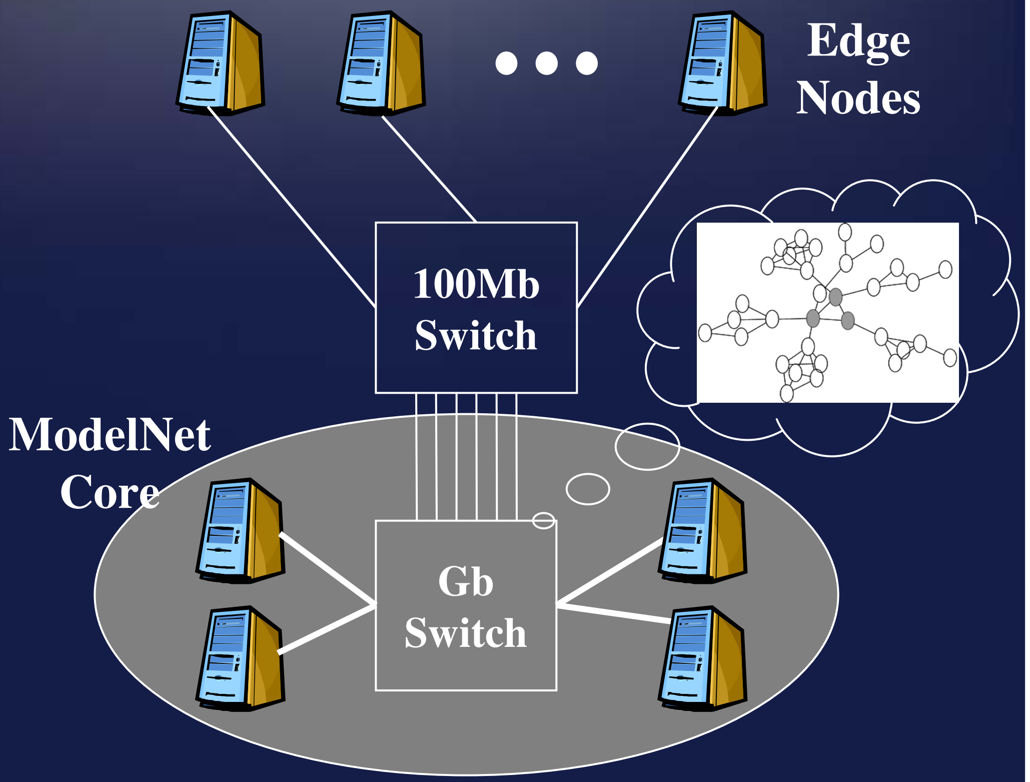
- System performance tuning
- Collection of events for visualization of execution
- Experiment recording + step-by-step and backwards replay for debugging
- Total-ordering of events
- Regression test suite for QA
- Benchmarking similar systems

Bonus Features

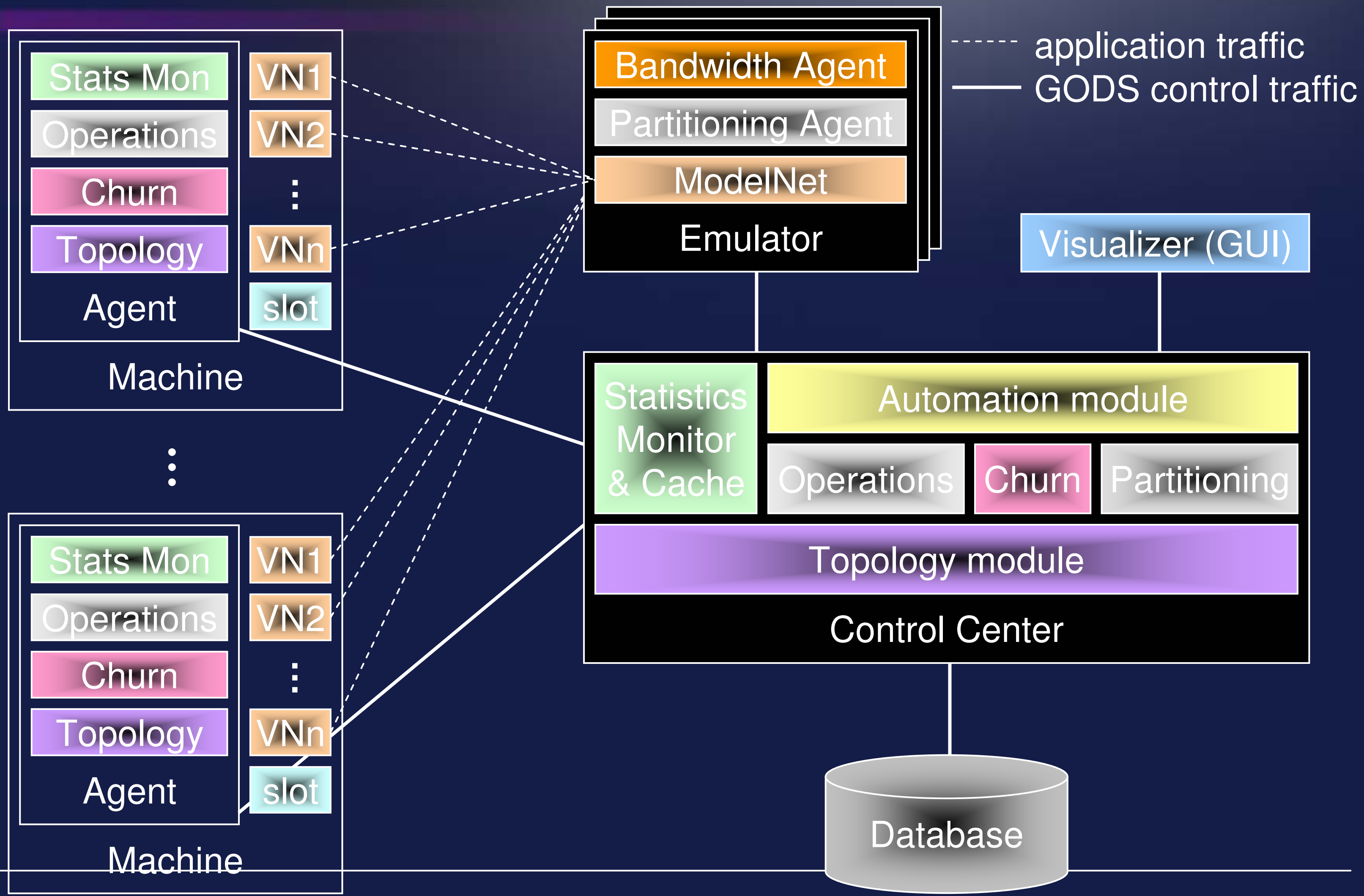
- Evaluating real application code
  - Fix defects only once
  - Account for various overhead
- Unmodified Java source code
  - Smooth adoption by other users
  - Avoid potential bugs
- Global knowledge about system state
  - Compiled global statistics
- Reproducible experiments

An Overview of ModelNet

- Several virtual nodes (app. instances) run as processes on each (edge) machine
- Virtual node bound to one IP alias
- VN traffic is routed to the core
- The core implements WAN emulation
  - Load virtual topology description
  - Each packet is virtually routed through the topology and delayed accordingly
  - The core can scale with more machines



GODS Architecture



Application Interface

XML descriptor for:

- Callable operations (e.g. lookup, broadcast)
- Event notifications (e.g. send/receive msg)
- Watched variables (pushed stats)
- Readable variables (pulled stats)

For Java apps:

- JVM instrumentation by JVMTI and JMX

Explicit interface (library) for other languages

GORDS Summary

Offers

- Evaluation of large-scale dynamic DS (P2P, Grids)
- System deployment and management
- Real-world WAN emulation
- Churn emulation
- Network partitioning emulation
- BW consumption measurement
- Global knowledge about the system
- Debugging/tracing of dist. algorithms
- System performance tuning, evaluation

Limitations

- Bound on accuracy of BW consumption measurement (lower bound on correlation between events and packets timestamps)
- Does not emulate NAT environments
- Lengthy experiments:
  - 2000 nodes + all pairs pings
  - 1 ping/sec -> 46 days ☹
  - 50 pings/sec -> 1 day 😐
  - 500 pings/sec -> 2 hours ☺

Scalable Peer-Group Grid Services

Scalable Grid services as group services (overlay services) deployed in a P2P network of containers

- A group service is provided by a peer group rather than by a single peer
  - Self-organizing, self-managing
- Higher throughput; higher availability
- Can be deployed on an overlay network – a P2P system of containers
  - On structured (name-based routing) overlays with DHT functionality, i.e. decentralized lookup (index) service within a VO
  - On unstructured (flooding) overlays, i.e. decentralized Grid resource information service across orgs or within a VO
- A service is available while at least one peer is up and running
- Client-service binding: one to any
  - Dynamic rebinding (allocation)

Scalable Group Grid Services (cont'd)

Scalable Grid services as group services (overlay services)

- Dynamicity: peers can join / leave / fail
- Exploit P2P self-organization and self-management
  - Handover on leave and join
- Scalability and (high) availability

Scalable Data Services in Grids Using a P2P Middleware

Multiple access points; high availability and throughput; self-management (comes from P2P)

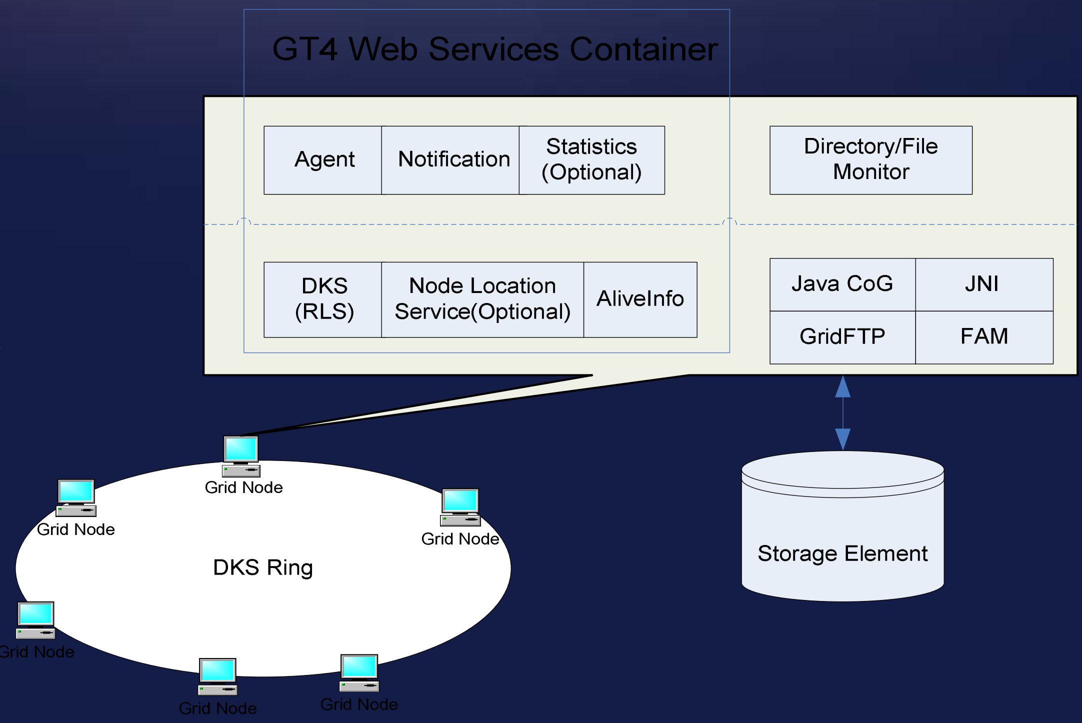
Examples: Scalable Data Services based on the DKS P2P middleware [http://dks.sics.se/]

- A P2P (overlay) replica management service
  - Ant-based mechanism with stigmergy
- A P2P distributed file system (KESO)
- A P2P resource discovery
- A P2P distributed back-up storage (MyriadStore)
- A P2P content delivery network (DOH)

Example: Replica Management Framework

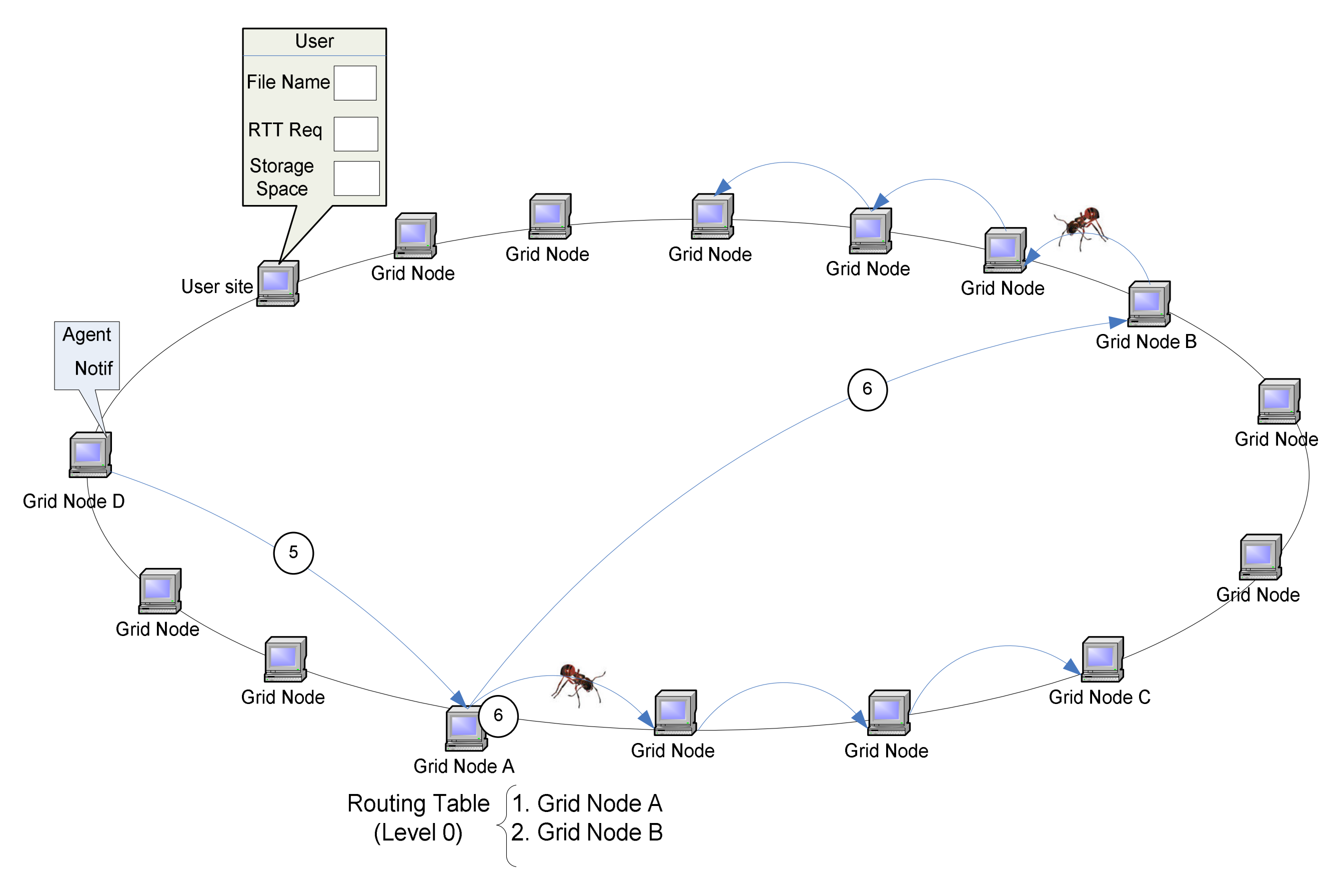
Components:

- Replica Location Service (RLS)
  - Based on DKS' DHT
- Replica Selection (Placement) Component
  - Uses “ants”
- Node Location Service (NLS)
  - using GT4's WS MDS Aggregator Framework
  - Several instances running
- Data Consistency Component
- Data Transfer Component
- Replica Usage Statistics Component



Ant Mechanism for Replica Placement

- Triggered when there is a need to place a new replica
  - neither of existing replicas fulfills QoS requirements
- Ants are sent to find a place(s) for new replica(s)
  - sigmergy



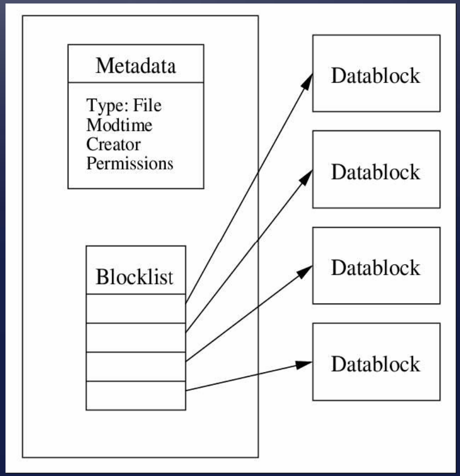
Example: P2P File System

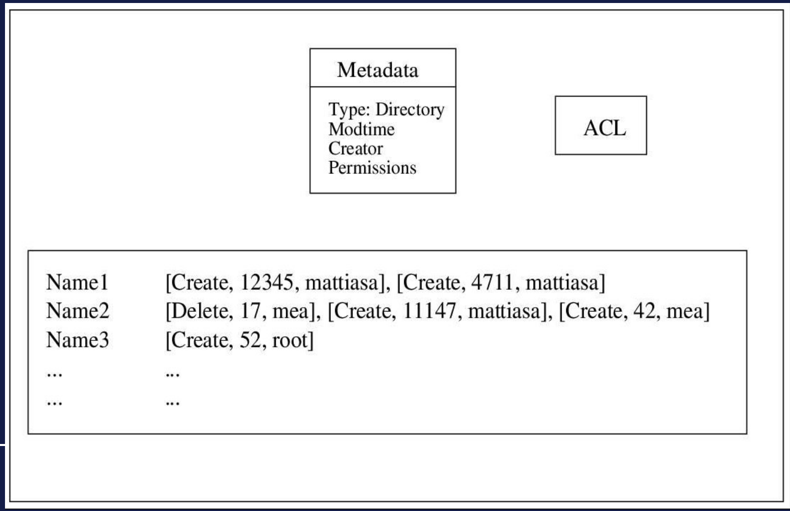
Keso is a distributed P2P file system built using the DKS P2P middleware

- Decentralized, scalable, self-organizing, secure
- Designed for real-world usage
- Can be mounted to a local file system

Organization:

- Files are split into blocks of equal size
- Blocks are referenced from a block list in the inode
- Each block and each inode is stored in DHT using a hash of its content
- Directory acts as a name/inode lookup service
- All versions of files are kept





MyriadStore: A P2P Backup Storage

Basic functionality:

- Backup
- Retrieval
- Browsing of backup data

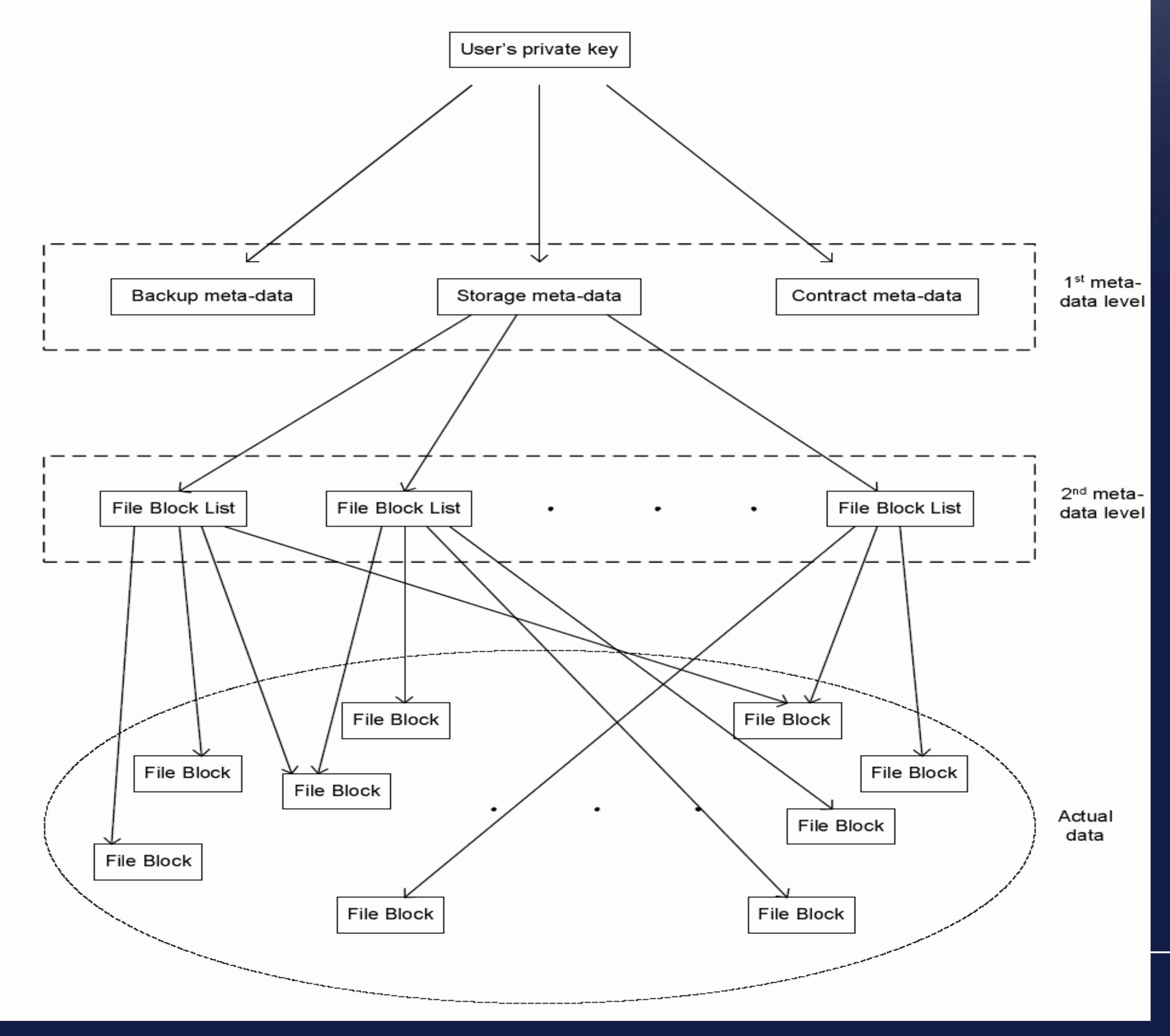
Organization:

- Meta-data is stored in the DHT of DKS
- Actual data are split into blocks and stored directly to the nodes local file systems
- All data items are encrypted before being stored remotely
- All nodes (users) have reputation which is a global numeric value maintained by all other nodes
- A higher reputation provides a longer grace period

Different schemas of trading for storage space

- Nodes exchange equal amount of disk space
- Contracts established between partners

Data in MyriadStore



Dynamic Grids

Management in Dynamic Grids

- A dynamic collection of resources
- Resources can be dynamically added / removed
- Resource can become unavailable
- VO members (both providers and consumers) can dynamically join / leave the VO

Example: VOFS: VO-aware Distributed File System

VOFS is a P2P system that aggregates data objects (files and directories) from different administrative domains in a virtual DFS similar to conventional NFS with standard POSIX file API

- Data objects (files, directories, disk space) are <u>exposed</u> to VOFS
  - Stay on place; logically linked in VOFS
- Spans multiple administrative domains
- Hides heterogeneity of aggregated file systems
- Illusion of an ordinary DFS, e.g. NFS
- Can be mounted to a local file system
  - Standard POSIX file API
  - Applications, existing file clients, e.g. Windows Explorer
- Metadata (index)
  - Centralized or DHT
  - Directory tree – traditional in DFS

VOFS (cont'd)

VOFS Security: based on (similar to) GT4 Community Authorization Service (CAS)

- Single-sign-on; role-based
- Mutual authentication with certificates; credential delegation
- Policy-based authorization
  - VO as a whole
  - User within VO
  - User on the server

Conclusions

Peer-Group Services in Grids

- Deployed on an overlay (P2P) network
- Multi access point
- The service is available while at least one peer can provide a service
- Also allow integration (sharing) of resources

Use of P2P for building group Grid services opens great opportunity for improving availability, throughput and integration