Secure Data Transfer and Replication Mechanisms in Grid Environments

Konrad Karczewski, Lukasz Kuczynski and Roman Wyrzykowski

Institute of Computer and Information Sciences,
Czestochowa University of Technology, Poland
email: [xeno,neron,roman]@icis.pcz.pl
www: http://icis.pcz.pl

Overview

In this presentation we would like to:

- present tasks of the modern Data Management System
- propose new solutions for data safety and security
- show a concept of modular system architecture designed with security in mind

Data Management

Data Management service:

- maintains
- discovers
- stores
- validates
- transfers
- instantiates

the data for the users' applications transparently

Features of Data Management System

- transparent access
- reliability
- security and safety of transferred and stored data
- access control
- possibility of transparent data compression
- access optimization

Transparent Data Access

Data Transfer System should be implemented as a layer between client application and Data Management System. Such an approach allows to:

- hide real data access mechanisms
- add new solutions without need for rewriting end–user application
- implementation of “intelligent” data access functionality

Data access optimization (1)

Modern Data Management System should provide data access optimization mechanisms. The main task of this subsystem is choosing the most suitable data location to use taking into account multiple factors, such as:

- available resources in the storage elements
- network properties:
  - bandwidth
  - topology
  - current throughput
- user’s access permissions

Data access optimization (2)

Additionally to improve efficiency and minimize data access time, data partitioning mechanism (splitting into smaller parts) could be used. In such case Data Broker would:

- decide on partitioning of the file
- search optimal locations for the parts
  - every part would be replicated in several locations to minimize chance of losing the data in case of a failure

Such a solution, besides improving the system fault–tolerance, would as well improve data access time by copying several parts in parallel from different locations.

Even the best Data Management System won’t be able to function properly when the data itself are unavailable

Storage Element or network failure could result in such an effect

Reliability (1)

Reliability is one of the most important aspects of the Data Management System. The following basic functionalities are required:

- improving fault–tolerance of the system by providing Data Replication mechanism
- automation and control by the Data Broker assures maximum transparency

Reliability (2)

Increase of reliability level could be accomplished by elimination of the single–point–of–failure. The basic way of achieving this is implementation of a distributed Data Broker Service. This implies:

- automated control takeover in case of failure
- metadata replication and synchronization
- distributed metadata server
- heartbeat mechanism

Security and Safety of Data (1)

To provide required level of security, the Data Management System must include:

- user authentication and authorization (e.g. GSI based)
- data encryption possibility
- permissions delegation (single–sign–on)

Security and Safety of Data(2)

To improve data safety the following mechanisms could be implemented:

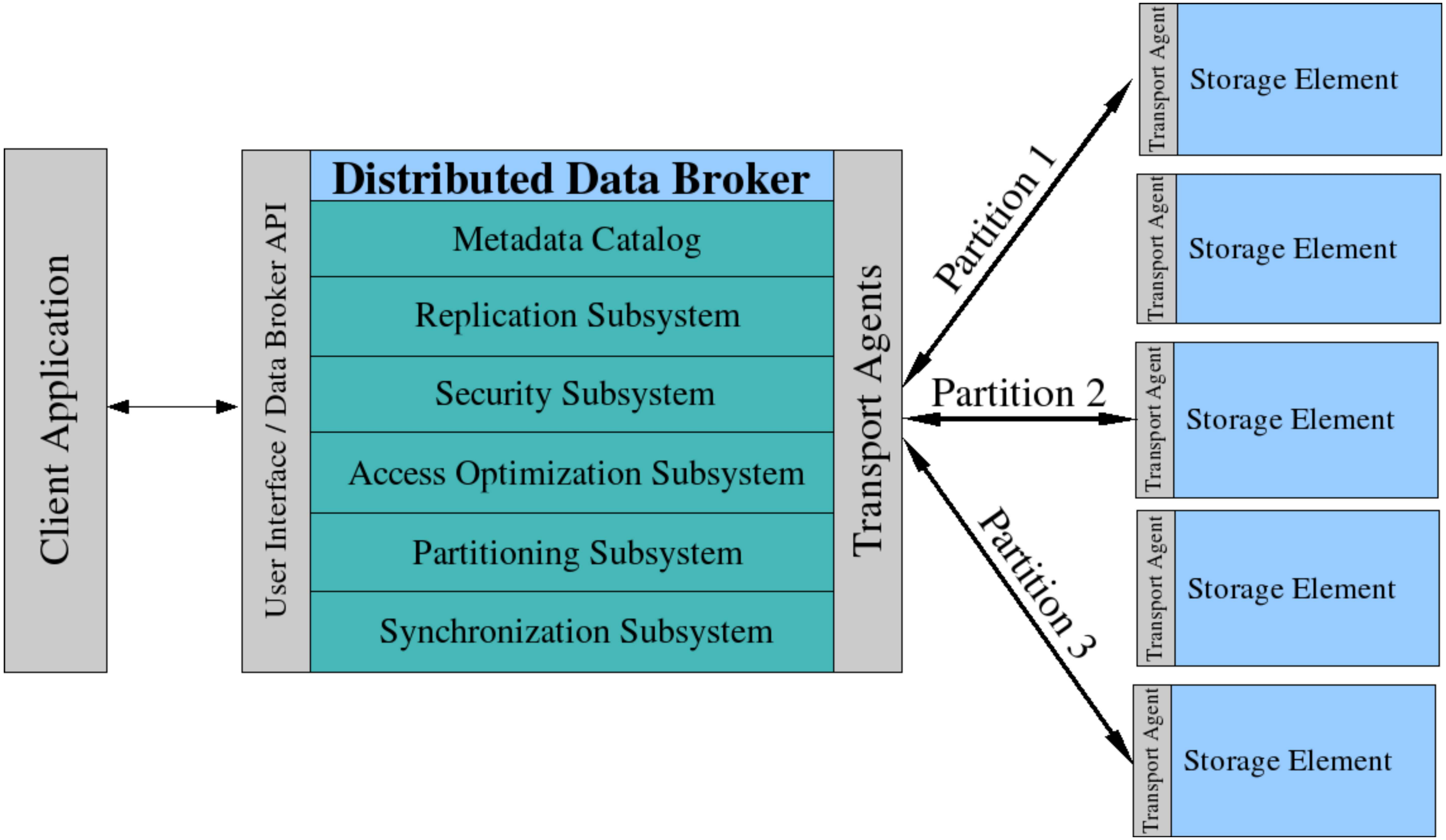
- Access Control Lists embedded in the metadata
- Data partitioning (only a part of data is available on each storage element)
- dataset name transformation (e.g. md5)

Access Control Lists

Access Control Lists allow to:

- manage access to resources
- constrain users' rights
  - access rights to data
  - access rights to metadata
  - manage visibility of data

System Architecture



Data Partitioning

Data Partitioning enables:

- increased data security:
  - no Storage Element holds complete information
  - repartitioning information available only to Data Broker
- increased data safety
  - parts stored on different Storage Elements
  - decreased chance to lose entire data
- increased performance
- equal load distribution amongst Storage Elements

Data Partitioning Mechanism (1)

Storing the data

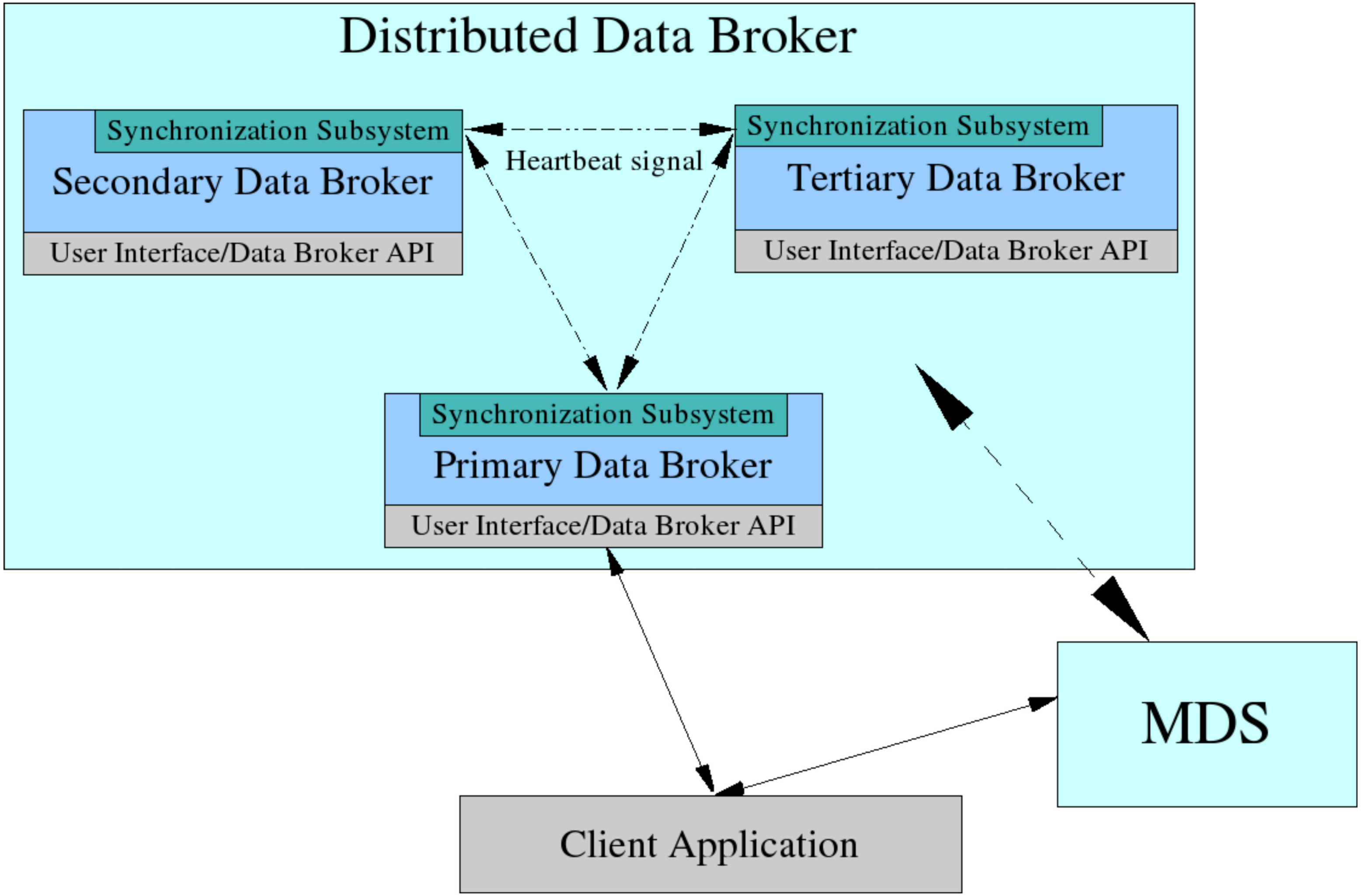
- client authenticates and authorizes in the GSI subsystem
- new dataset is registered in the Data Broker
- Data Broker:
  - decides on partitioning of the dataset
  - searches for optimal locations
  - encrypts and splits data
  - digests dataset name
  - stores parts on the Storage Elements
  - updates Metadata Server information

Data Partitioning Mechanism (2)

Data retrieval

- client
  - authenticates and authorizes in the GSI subsystem
  - requests data retrieval using dataset name
- Data Broker:
  - digests dataset name and checks users’ rights in the Metadata Server
  - retrieves repartitioning information
  - chooses optimal Storage Elements for data retrieval
  - retrieves parts of the dataset and reunites the data
  - decrypts dataset and tranfers it to the requested location

Distributed Broker Architecture


Distributed Data Broker
Synchronization Subsystem
Secondary Data Broker
User Interface/Data Broker API
Heartbeat signal
Synchronization Subsystem
Tertiary Data Broker
User Interface/Data Broker API
Synchronization Subsystem
Primary Data Broker
User Interface/Data Broker API
MDS
Client Application

Broker Operation Schema

Normal operation:

- client queries MDS for the Data Broker
- MDS returns current Primary Data Broker address
- client communicates with the Data Broker

Broker / network failure:

- Synchronization Subsystem detects failure
- Secondary Data Broker takes over Primary Broker functions
- the new Primary Data Broker updates MDS information

Summary

The proposed solution will allow to:

- use common storage elements for sensitive data
- hide presence of a dataset on the storage element to unauthorised users
- eliminate single–point–of–failure

It will help to:

- secure data from unauthorized access
- balance the load of storage elements
- optimize access time