MediGRID

ZIH
Zentrum für Informationsdienste
und Hochleistungsrechnen

D-Grid Security Workshop
Göttingen, 28th March 2007

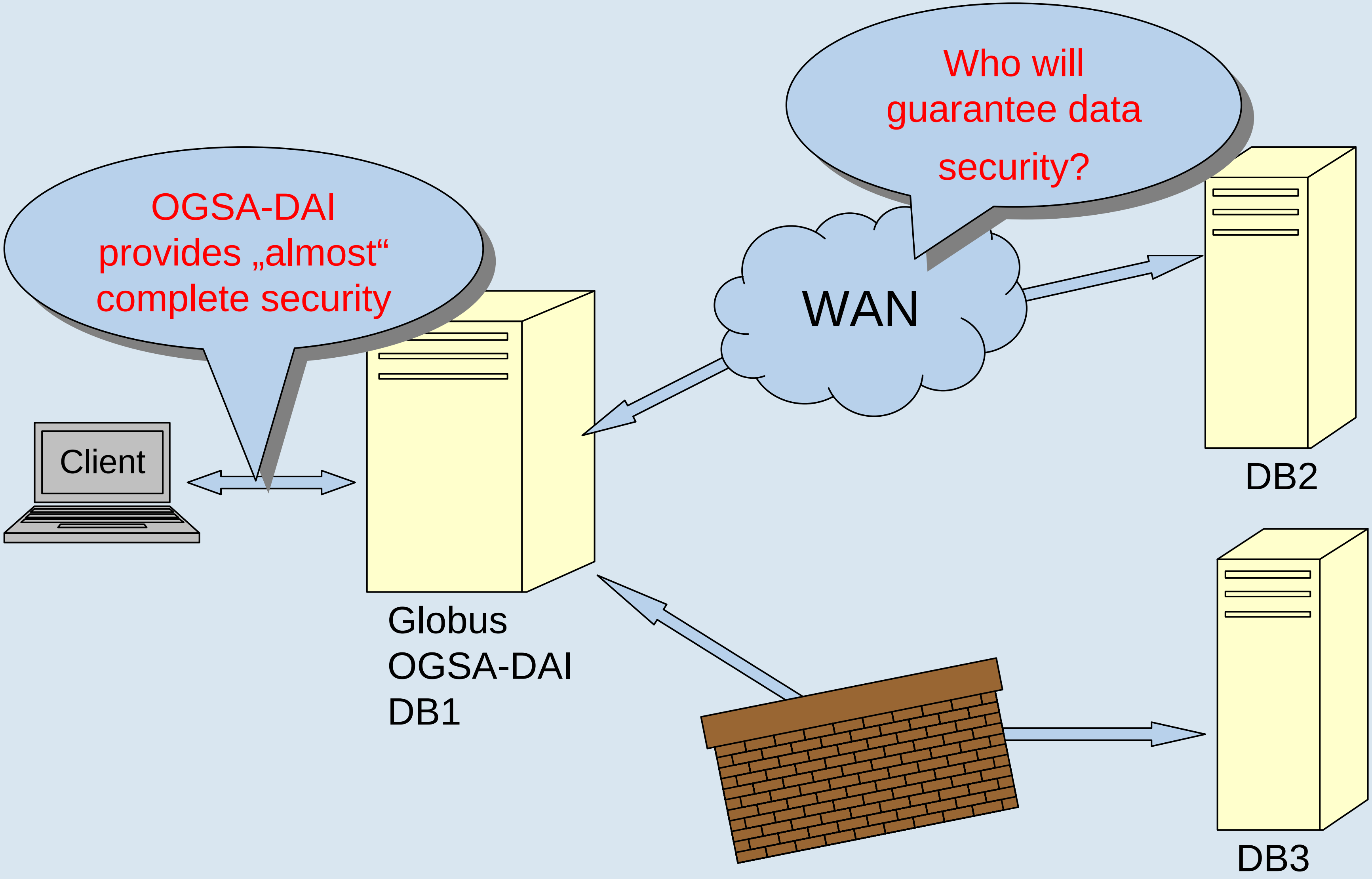
Security and Authorizations in OGSA-DAI & SRB

Samatha Kottha

Content

- Security & Authorizations in OGSA-DAI
  - Features
  - Holes
- Security & Authorizations in SRB
  - Features
  - Holes

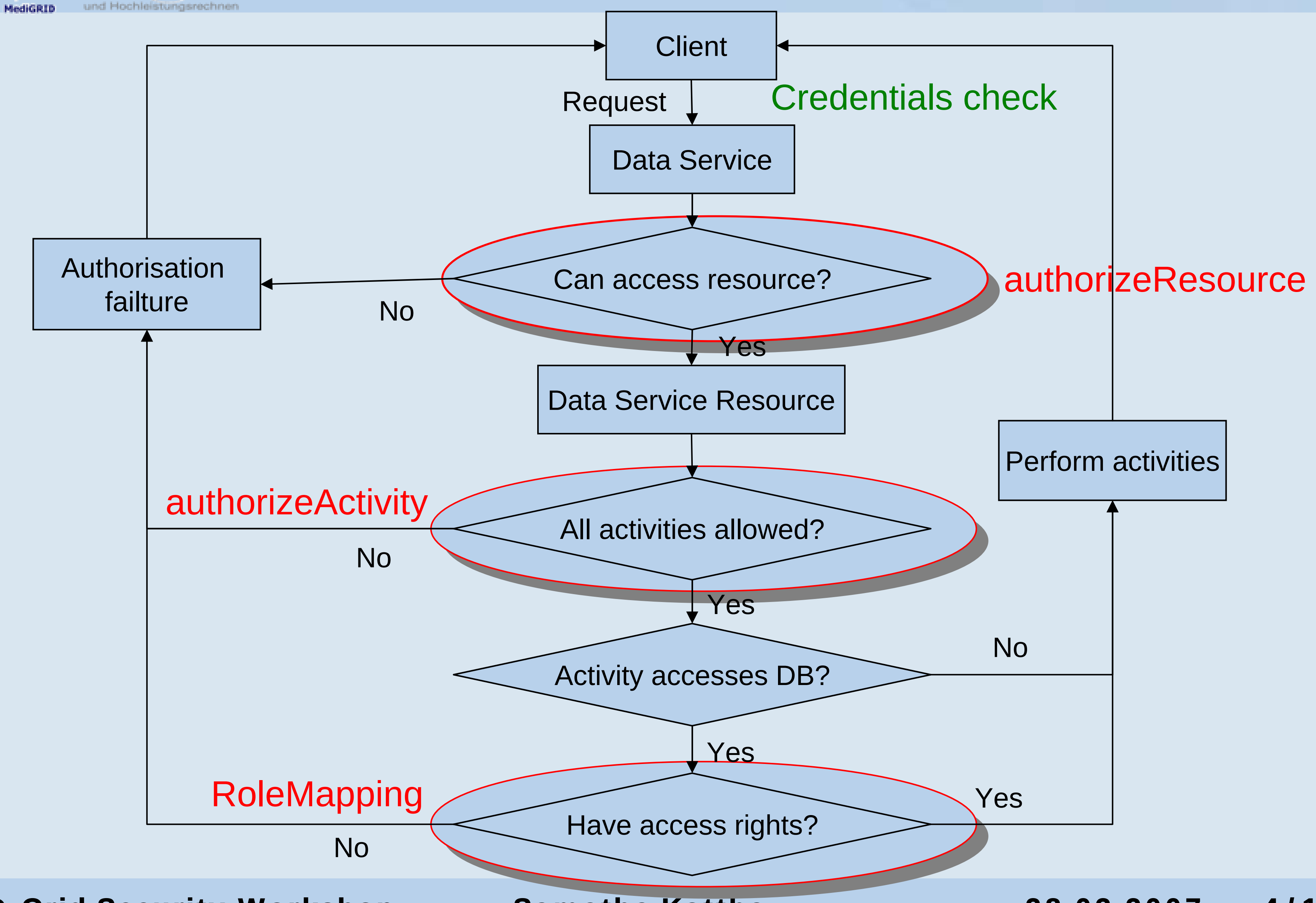
OGSA-DAI Landscape



OGSA-DAI provides „almost“ complete security

Who will guarantee data security?

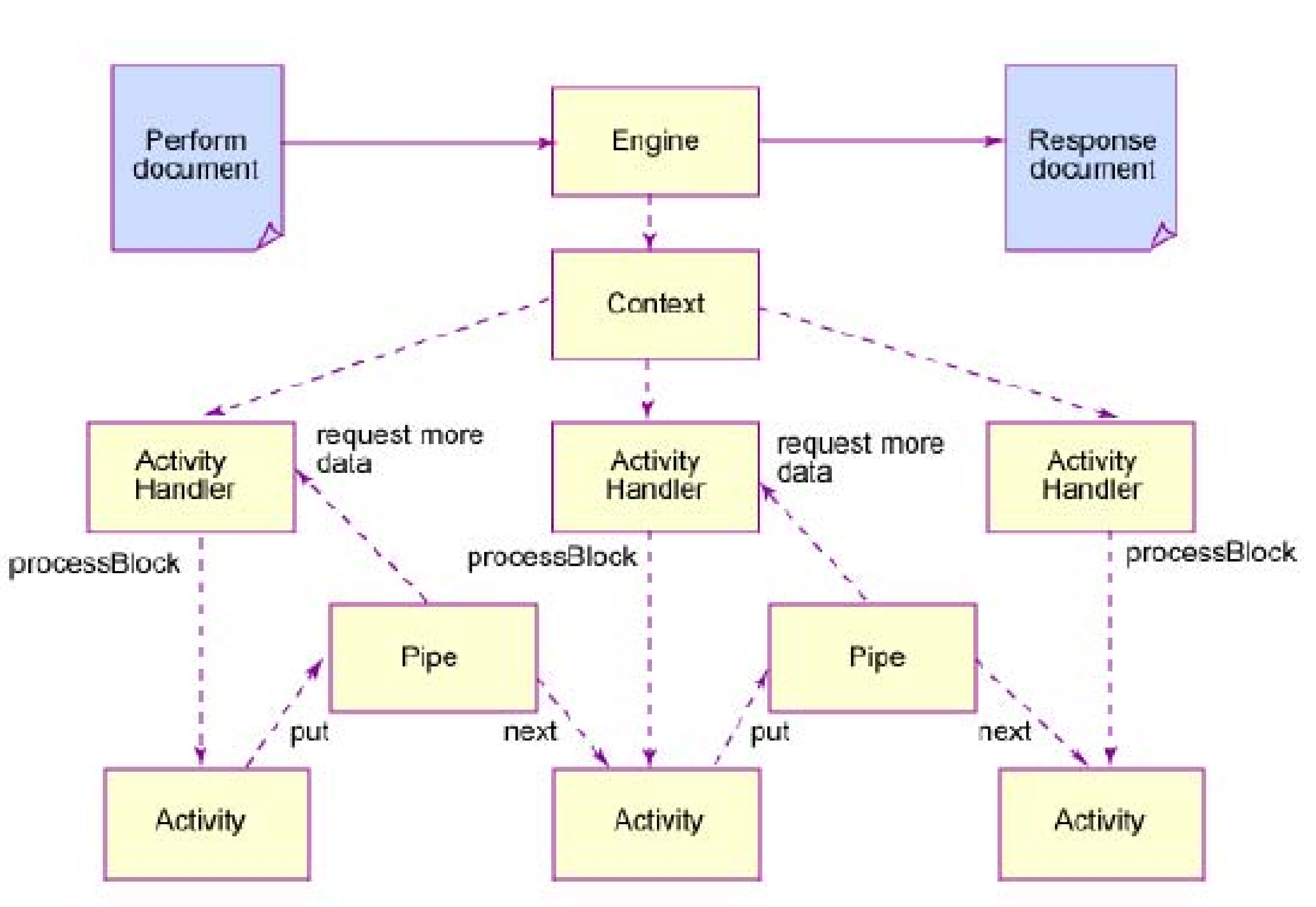
Authentication & Authorizations



Role Mapping

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- (c) International Business Machines Co           n, 2002 - 2005.-->
<!-- (c) University of Edinburgh, 2002 - 2                          -->
<!-- See OGSA-DAI-Licence.txt for lic                                -->

<DatabaseRoles xmlns="http:/
               xmlns:xsi="http://
               xsi:schemaLocatio                                  nig      file:///opt/globus-
4.0.2/share/schema/ogsadai/xsd/

<Database name="jdbc:mysql://tini.zib.                  IMAP?jdbcCompliantTruncation=false">

<User dn="" userid="dummy" password="dur   ny" />
<User dn="/O=GermanGrid/OU=TUD/CN=Samatha Kottha" userid="kottha" password="unknown" />

</Database></DatabaseRoles>
```

How it works?


Perform document
Engine
Response document
Context
Activity Handler
request more data
Activity Handler
request more data
Activity Handler
processBlock
processBlock
processBlock
Pipe
Pipe
put
next
put
next
Activity
Activity
Activity

Security Holes in OGSA-DAI

deliverFromFile & deliverToFile – Do not activate

No mechanism to provide authorization control for

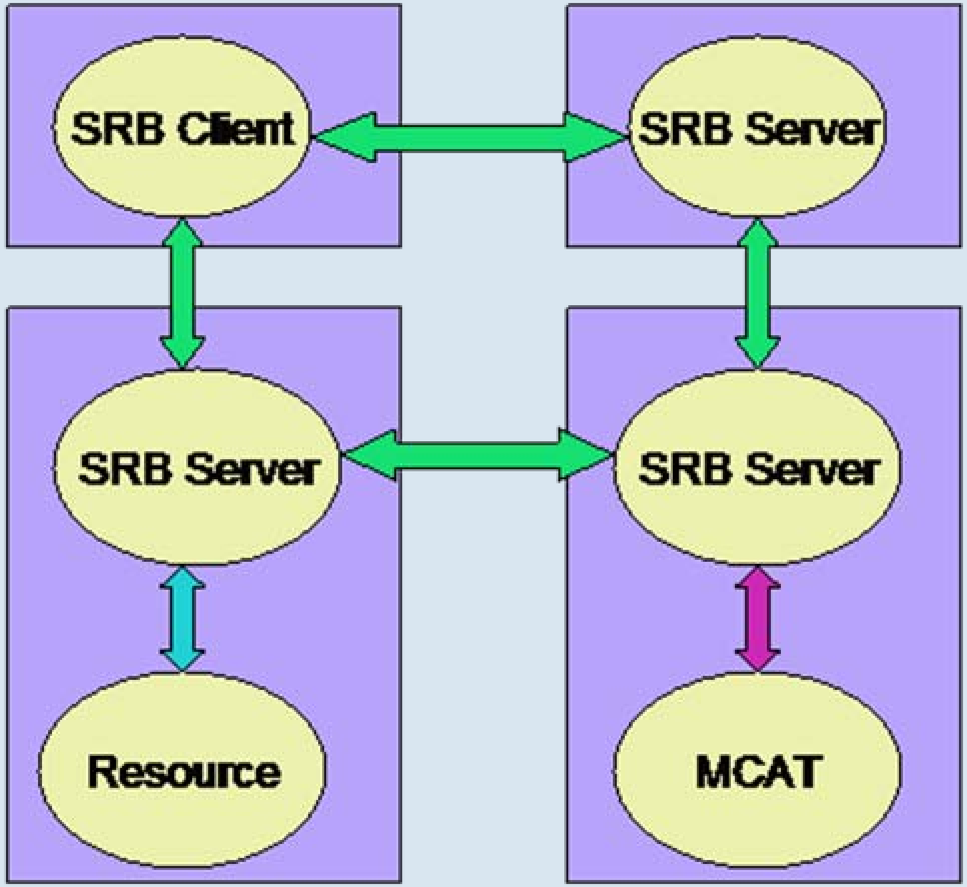
- data transport streams – if someone allowed to access data service resource and is able to guess the name of your data stream could read your data
- data written to the stream servlet

SRB Authentication

Three flavors: SDSC SRB, *i*RODS, and Nirvana SRB

Authentication	SDSC SRB	*i*RODS	Niravana SRB
GSI	☑	☒	☑
Password (Challenge / response protocol)	☑	☑	☑
KERBEROS	☒	☒	☑

SRB Architecture



SRB Processes and Communication Overview

Weakest links:

- ➲ Client
- ➲ DB Server

Security

- SDSC SRB is quite secure
- The code is completely audited by a US Govt. agency – no buffer overflows
- SRB servers execute as a non-privileged user
- SRB MCAT server and DB server are in same LAN – so less vulnerable to WAN attacks

Authorizations

Kind of operating system with in a operating system:

- Users
- Groups
- Domains
- Collections
- Tickets

Data Encryption

- Data is encrypted and/or compressed on the client side and the files are stored in that form.
- The secret key is securely stored in MCAT but in "plain-text".
- These steps are done by Sput.pl and Sget.pl scripts.
- You won’t find these scripts anywhere in SRB website

Security Holes in SRB

- Weak Clients: The passwords could be read from ~/.srb/.MdasAuth file (Encrypt1)
- Vulnerability when Encrypt1 users run Spasswd to change their password
- SRB is as safe as your DBMS – So better choose a commercial DBMS than Open Source
- MdasConfig file contains DBMS access info – so keep it safe and store it on local file system instead of NFS
- Don’t allow anyone to become “globus” user (OGSA-DAI – deliverToFile or deliverFromFile) if you are using GSI

Security Holes in SRB

- GSI authentication between servers: creation of proxy certificate – long vs. short
- One compromised SRB server is enough to compromise the entire zone but threat to another zone is minimal.
- Use –enable-accsctrl configure option – Adds meta data access control otherwise every user can read meta data of every file !!

SRB software itself is quite secure, but ...

Thank you!