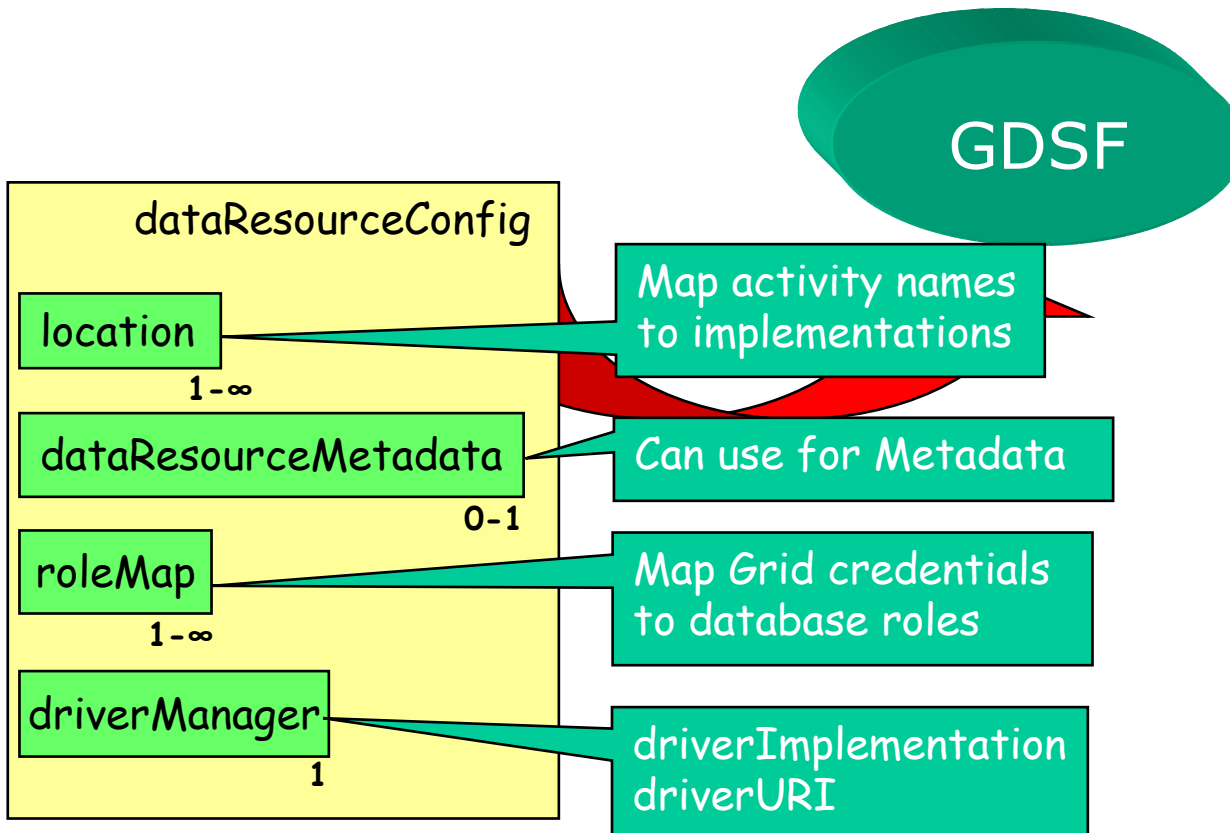


Data Resource Configuration

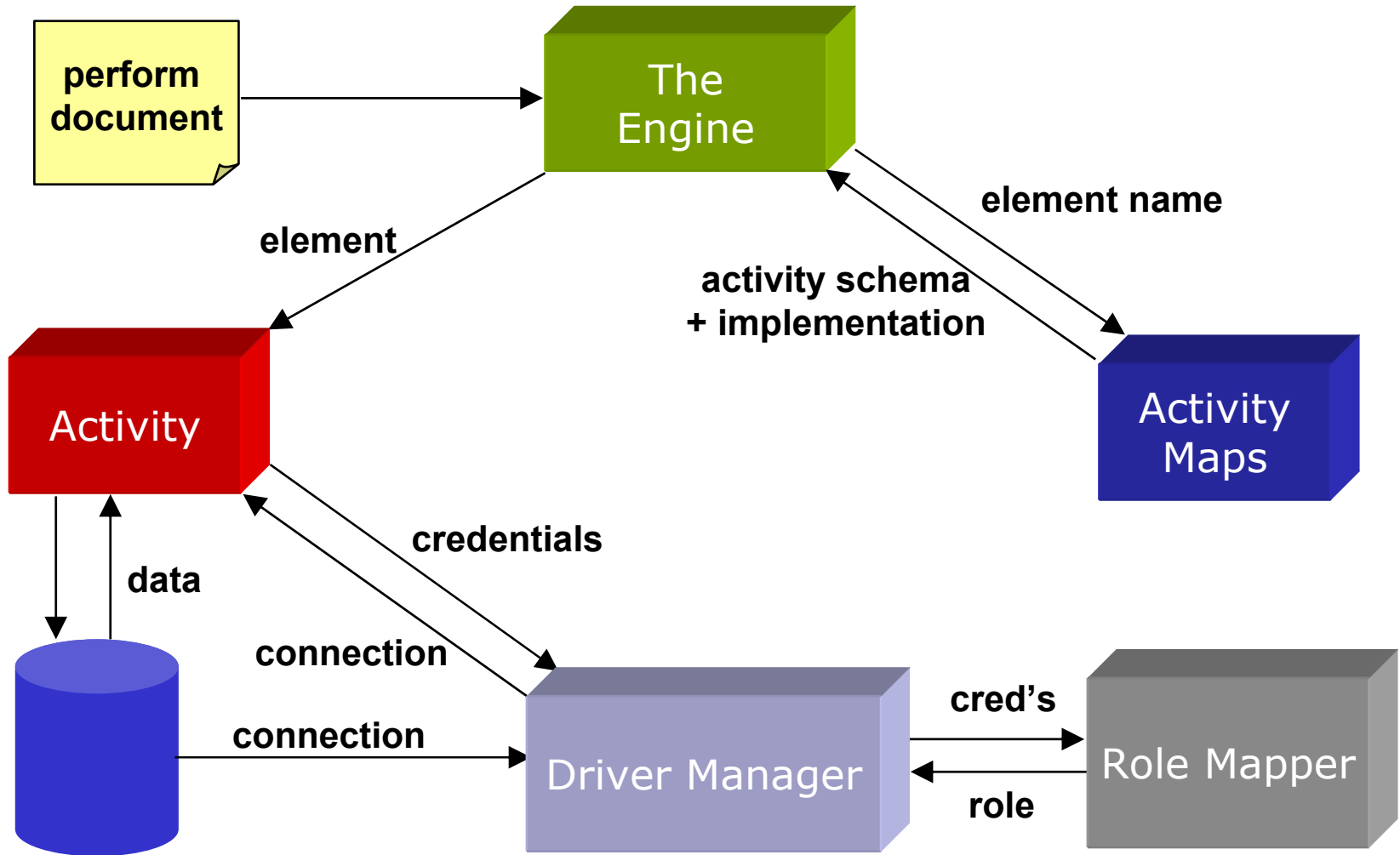
Andrew Borley
borley@uk.ibm.com

- ▶ Data Resource Configuration
- ▶ How a GDS Works
- ▶ Elements of Configuration
- ▶ Configuring Your Database

GDSF Configuration



How A GDS Works



▶ Database Driver

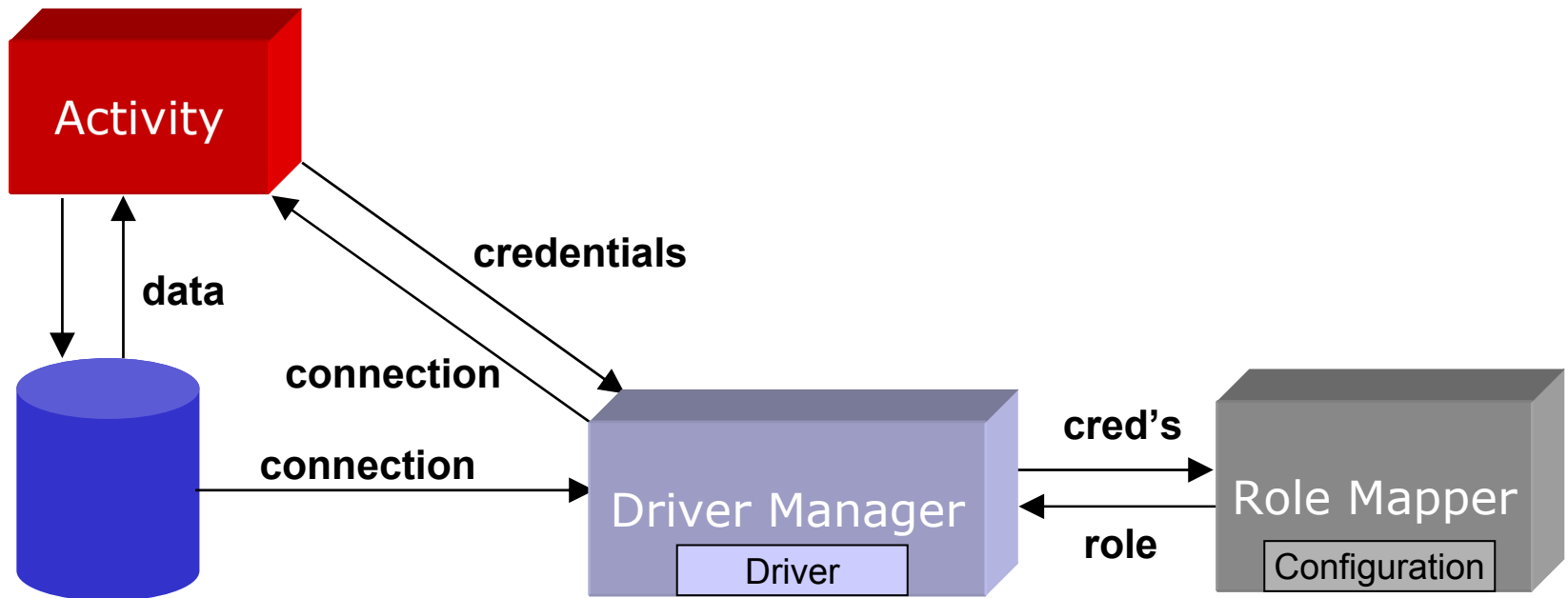
- Name of the driver class
- Is the JAR file in the library path of the container?

▶ Database specifics

- Relational or XML database?
- Metadata you'd like to publish
- Vendor and version of your database

▶ Authorisation

- Grid Credentials: How do distinguished names map to database roles?



- ▶ The Driver Manager implements a wrapper around the JDBC or XMLDB driver
- ▶ OGSA-DAI provides simple driver managers for
 - JDBC drivers:
`SimpleJDBCDataResourceImplementation`
 - XMLDB drivers:
`XMLDBDataResourceImplementation`

- ▶ The *driver* element specifies the driver class and the connection URI

```
<driver>
  <driverImplementation>
    org.apache.xindice.client.xmldb.DatabaseImpl
  </driverImplementation>
  <driverURI>
    xmldb:xindice://localhost:4080/db/littleblackbook
  </driverURI>
</driver>
```


- ▶ Maps Grid credentials to database roles
- ▶ OGSA-DAI provides a simple file role mapper:
 - Maps distinguished name to database roles
 - Mappings are saved in a configuration file
 - User names and passwords are in plain text

```
<roleMap name="Name"  
  implementation="uk.org.ogsadai.common.  
                rolemap.SimpleFileRoleMapper"  
  configuration="path/ExampleDatabaseRoles.xml"/>
```

- ▶ Rolemap files can be shared among GDSFs
- ▶ One **Database** entry per data resource
- ▶ The driver URI identifies the database

```
<DatabaseRoles>
```

```
  <Database
```

```
    name="jdbc:mysql://localhost:3306/ogsadai">
```

```
    <User dn="No Certificate Provided"
```

```
      userid="ogsadai" password="ogsadai" />
```

```
  </Database>
```

```
</DatabaseRoles
```

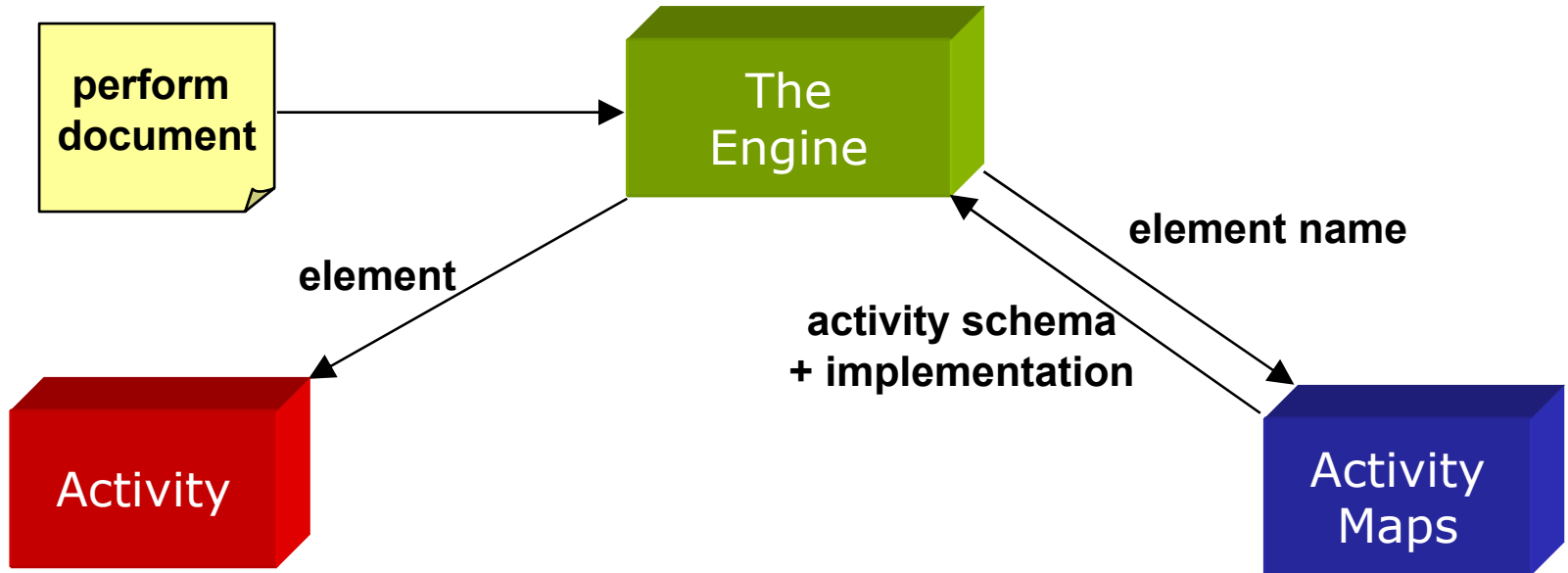
- ▶ Can provide static and dynamic meta data
- ▶ Static metadata is contained in the data resource config

```
<productInfo>  
  <productName>MySQL</productName>  
  <productVersion>4</productVersion> ...  
</productInfo>
```

- ▶ Dynamic metadata is obtained from callback functions at time of the request

```
<relationalMetaData>  
  <databaseSchema callback="uk.org.ogsadai.porttype.gds.  
    dataresource.MySQLMetaDataExtractor"/>  
</relationalMetaData>
```

- ▶ The configuration file provides extension points for new meta data elements
- ▶ Elements within the **dataResourceMetaData** tag will be published automatically
- ▶ Static metadata
 - Just add a new subelement to the **dataResourceMetaData** element
 - Can be of arbitrary (XML) format
- ▶ Dynamic metadata
 - Add a new callback element and provide the implementation



- ▶ Maps from an element in the perform document to an implementation
- ▶ Specifies a schema type against which the element must validate

```
<activityMap name="xslTransform"  
  implementation="uk.org.ogsadai....XSLTransformActivity"  
  schemaFileName="xsl_transform.xsd" />
```

- ▶ Main categories:
 - Statement
 - Delivery
 - Transform
- ▶ Delivery and transform activities do not depend on the data resource

- ▶ Statement activities work only with certain types of data resources
 - XPathQueryActivity
 - SQLQueryStatementActivity
 - SQLUpdateStatementActivity
- ▶ Your database may need a specific implementation for some activities, e.g.
`uk.org.ogsadai.oracle.SQLQueryStatementActivity`
The activity schema remains the same.

- ▶ Wraps a set of activityMap elements
- ▶ The base attribute points to a common location where the XML schemas of all enclosed activities can be found

```
<location base="http://localhost:8080/ogsa/  
            schema/ogsadai/xsd/activities/">  
  <activityMap name="sqlQueryStatement" ... />  
</location>
```

▶ DriverManager:

- Type of driver? JDBC or XMLDB

▶ Driver:

- Driver class e.g. `org.gjt.mm.mysql.Driver`
- connection URI e.g. `jdbc:mysql://localhost:3306/ogsadai`

▶ RoleMap:

- Does the configuration attribute point to the correct rolemap file?
- Are database URIs in the rolemap file and in the `dataResourceConfig` file identical?

▶ MetaData:

- Configure static metadata as desired
- Set up callback classes for XML or relational databases