SQL Replication

- Replication Strategies
- Evaluated Technologies
- Enhydra Octopus:
  - Architecture
  - Setup
  - Use:
    - GeometryDB: Oracle -> MySQL
    - ConditionsDB: MySQL -> MySQL

Replication Strategies

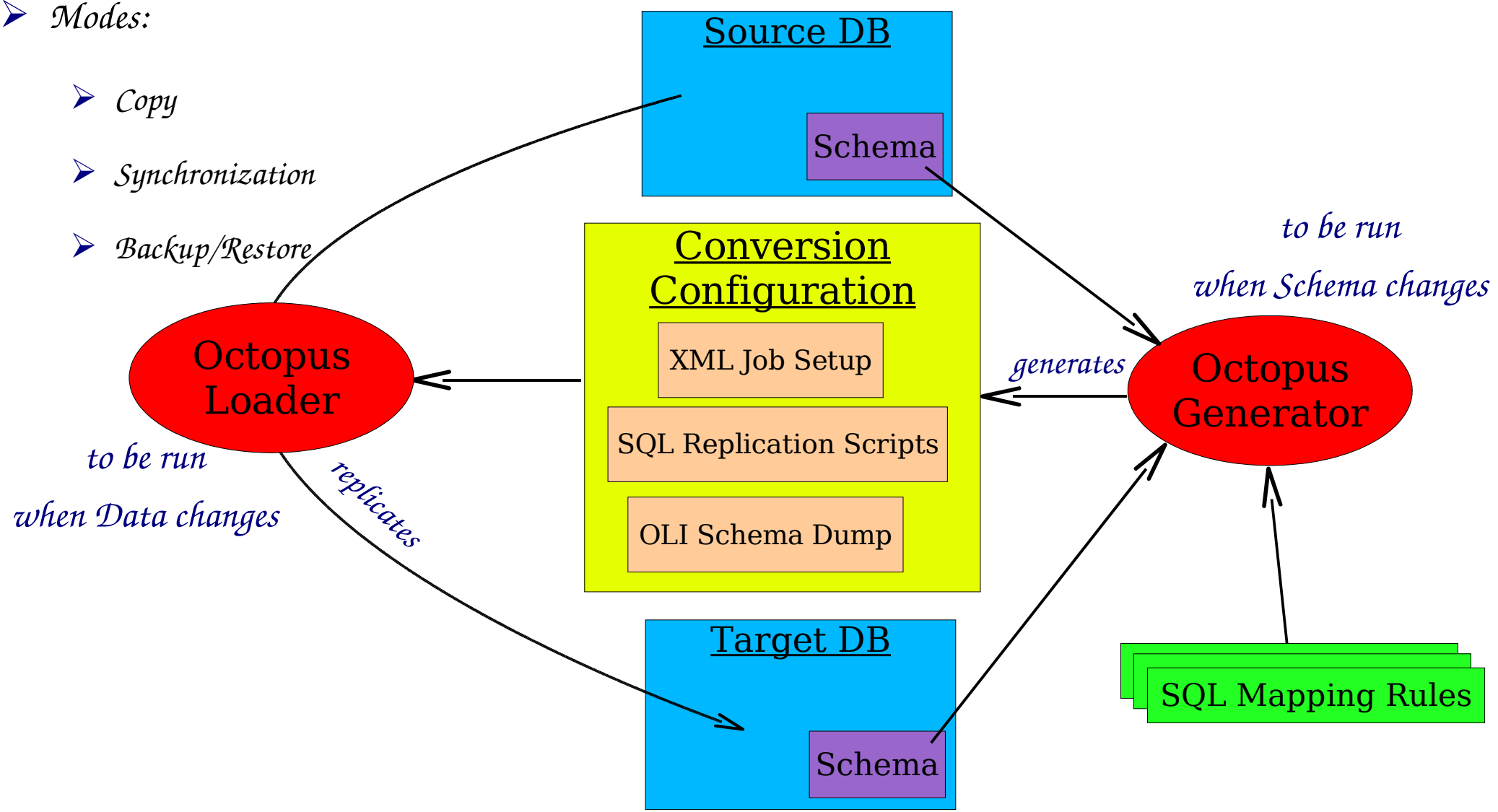
- Replication ways:
  - Native (MySQL, Oracle,...) Tools
  - Generic Convertors (like Octopus)
  - Actual Applications
- Replication Use Cases:
  - Replication within technology (MySQL->MySQL,...):
    - Native Tools are fast and sure
    - Generic Convertors provide unified API
  - Replication between technologies (Oracle->MySQL,...):
    - If same Schema in both technologies: Generic Tools do well
    - If almost same Schema in both technologies: Generic Tools can be configured with customised mapping (using configuration files or special plugins)
    - If different Schema in both technologies: Actual Applications should be used

Evaluated Technologies

- SQLTuple/ColMan: good for AttributeList/NTuple or similar mapping
- Groovy-based Replicator: learning tool, simple, easy, basic
- JDBC Importer: a bit different mission, needs a lot of customizations
- Enhydra Octopus: seems to be the best match

Octopus Architecture

- Support for Access, DB2, MSQL, Paradox, CJDBC, Excel, Informix, McKoi, PostgreSQL, Sybase, Csv, Hypersonic, Instantdb, MySQL, Oracle, QED and XML.
- Modes:
  - Copy
  - Synchronization
  - Backup/Restore



Octopus Setup

- Ant build.xml file
- Run description build.properties file
- Customised mapping map.properties file

```
Src.Db=@sundb07.cern.ch:1521:pdb01-1
Src.User=atlasdd
Src.Passwd=bla
Src.Schema=ATLASDD

Dest.Db=atlasdbdev.cern.ch:3306/CollectionTest
Dest.User=CollTester
Dest.Passwd=bla

Log.Mode=full

Tables=ALIN_DATA;ALIN_DATA2TAG
```

```
# Clean everything
ant clean

# Generate replication scripts
ant generate

# Perform the replication
ant load

# Recompile Atlas customizations
ant patch

# Get help
ant -projecthelp
```

```
octopus.mysql.varchar2(4000).type=varchar
octopus.mysql.varchar2(4000).length=255

octopus.mysql.float(126).type=double

octopus.mysql.float(63).type=float

octopus.mysql.number(1).type=tinyint
octopus.mysql.number(1).length=1

octopus.mysql.number(10).type=integer
octopus.mysql.number(10).length=10
```

GeometryDB Replication

- Oracle->MySQL
- Fixed problems:
  - Support for Oracle/varchar2 to MySQL/varchar, etc.: The requirement is to convert Oracle/varchar2 into MySQL/varchar. The problem is that Atlas GeomeryDB contains varchar2(4000), while MySQL allows the longest varchar(255). Standard Octopus doesn't support that (of course). This is now supported by Atlas Patch to Octopus via map.properties file. Note, that those user-supplied rules can be highly unsecure.
  - Support for replication of all tables belonging to an Oracle Schema: This was a missing feature in Octopus (Octopus could replicate either the whole database or just an explicit list of tables). It has been implemented and fed back to Octopus developers.
  - Unique Index on multiple columns: This was a bug of Octopus. It has been fixed.

ConditionsDB Replication

- MySQL->MySQL
- Fixed problems:
  - MySQL tables contain a Primary Index with the name "PRIMARY": While some MySQL commands accept that, it is not generaly supported as MySQL identifier because "PRIMARY" is a MySQL reserved word. Atlas Octopus Patch now creates commands which accept "PRIMARY" as a Primary Index name. Note, that using reserved words as Identifiers is a bug in Database design.

Documentation

- Enhydra Octopus is provided by ObjectWeb Consortium, released under GPL, it has active user base and responsive developers.
- Reusable Atlas-motivated fixes and modifications fed back to Octopus developers to will be included in the distribution.
- Documentation:
  - Home: http://octopus.objectweb.org
  - Wiki: https://uimon.cern.ch/twiki/bin/view/Atlas/DatabaseReplication