



Heterogeneous Replication with *Octopus*

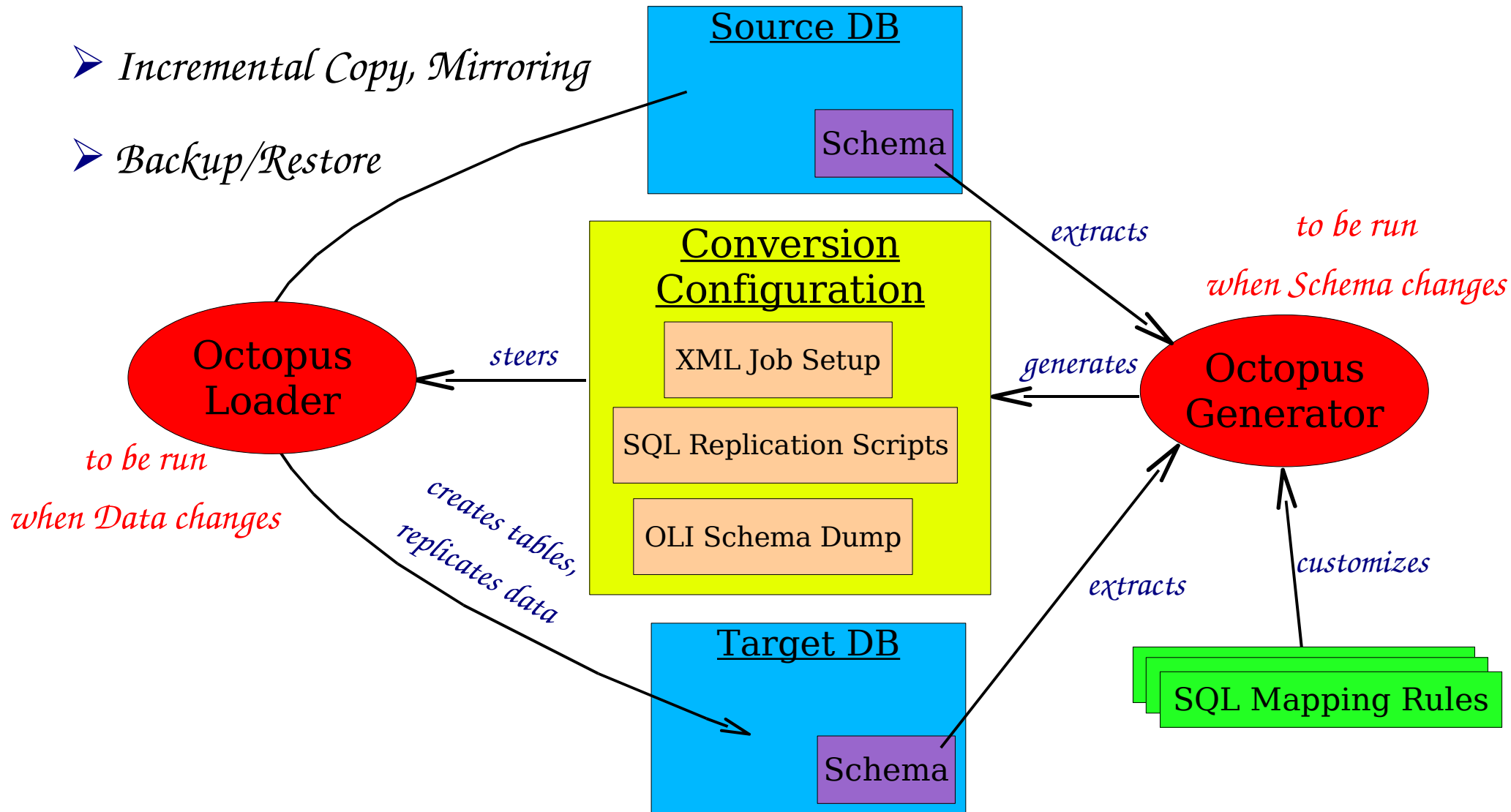


- *Octopus* is a Java-based Extraction, Transformation, and Loading tool. It may connect to any *JDBC* data sources and perform transformations defined in an XML file.
- *Octopus* supports MSSQL, Oracle, DB2, QED, JDBC-ODBC, Excel, Access, MySQL, PostgreSQL, McKoi, Hypersonic, Informix, Paradox, Sybase, CSV-files, XML-files, CSV-files, MS-SQL, XML and i18n. Other databases can be easily included.
- Enhydra *Octopus* is provided by **ObjectWeb** Consortium, Open Source Middleware released under Apache/GPL, it has active user base and responsive developers.

Octopus Architecture

➤ Modes:

- Full Copy
- Incremental Copy, Mirroring
- Backup/Restore





Octopus Setup



- Ant build.xml file
- Run description build.properties file
- Customised mapping map.properties file
(designed to satisfy special Atlas requirements)

```
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
```



Octopus Customization

```
<LoaderGeneratorTask
  sourceType="Oracle"
  sourceDriverName="oracle"
  sourceDataBase="${Src.Db}"
  sourceUser="${Src.User}"
  sourcePassword="${Src.Passwd}"
  targetType="MySQL"
  targetDriverName="mm"
  targetDataBase="${Dest.Db}"
  targetUser="${Dest.User}"
  targetPassword="${Dest.Passwd}"
  valueMode="copy"
  generatorOutput="../run"
  domlUrl=""
  generateDoml="true"
  packageName="org.webdocwf.util.loader"
  generateDropTableStmt="true"
  generateDropIntegrityStmt="false"
  generateCreateTableStmt="true"
  generateCreatePKStmt="true"
  generateCreateFKStmt="true"
  generateCreateIndexStmt="true"
  generateXml="true"
  fullMode="true"
  logMode="${Log.Mode}"
  schema="${Src.Schema}"
  includeTableList="${Tables}"
  logDirName="../run"
  octopusHome="${Octopus.Home}"/>
```

```
<LoaderTask
  mode="${Log.Mode}"
  logDir="../run"
  restartIndicator="false"
  onErrorContinue="true"
  loadJob="../run/LoaderJob.olj"/>
```

➤ Overall customization: *build.xml*

➤ Detailed customization via
generated files:

➤ *XML Job Setup*

➤ *SQL Replication Scripts*

*Customization is simple as long as it can
be expressed via generic XML description
and SQL commands. More advanced
filters can be written as plugins.*



Octopus Enhancements in Atlas



- Support for SQLite: LCG doesn't provide JDBC driver for SQLite, out-of-box Octopus doesn't support SQLite. => SQLite JDBC Driver has been written and included in Octopus.
- Support for non-standard SQL types mapping: Databases based on LCG tools are often incorrect and/or inconsistent. => Octopus can be forced to accept incorrect/inconsistent mapping.
- Support for database configurations not supported in out-of-box Octopus: Octopus doesn't understand Oracle Schema, Synonyms,... => All needed configurations have been added.
- Bug-fixes: included in the Octopus distribution.



Octopus Use in Atlas



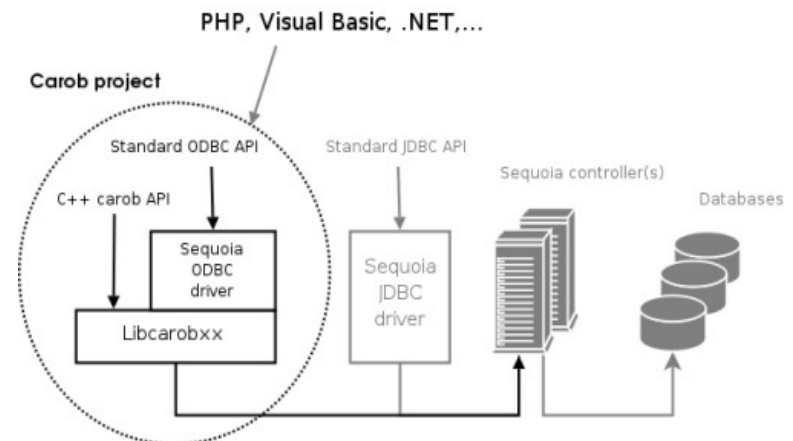
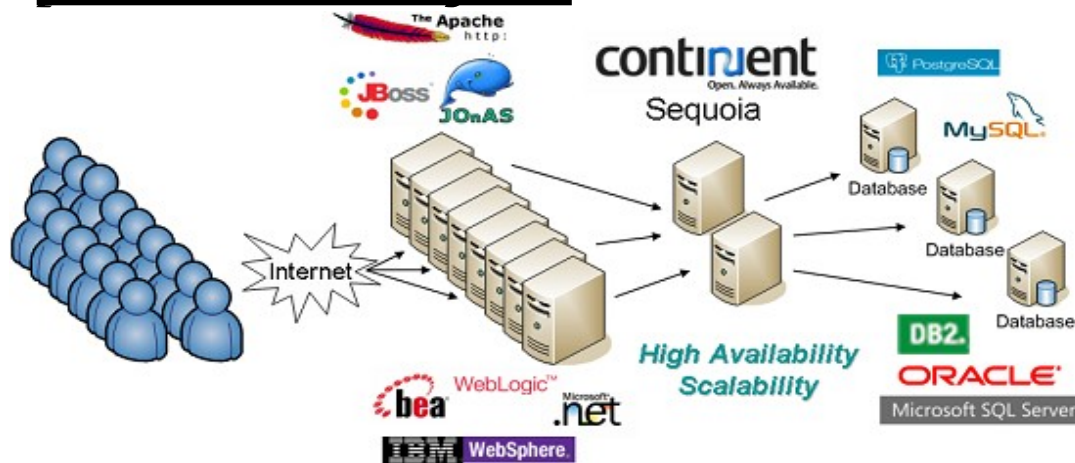
- Geometry DB: replicated databases in Atlas distribution
 - Oracle -> MySQL
 - Oracle -> SQLite
- Tag DB: all Rome tags replicated from CERN to BNL
 - Oracle -> MySQL
 - MySQL -> Oracle
 - MySQL -> MySQL



Sequoia & Carob



- *Octopus* is part of a wider *Sequoia* project: <http://sequoia.continuent.org>
- *Sequoia* is a transparent middleware solution for offering clustering, load balancing and failover services for any database. *Sequoia* is the continuation of the C-JDBC project. The database is distributed and replicated among several nodes and *Sequoia* balances the queries among these nodes. *Sequoia* handles node failures and provides support for checkpointing and hot recovery.
- *Carob* provides *Sequoia* access to ODBC, C and C++ users.
- JDBC + Sequoia/Carob + Hibernate provide now all functionality which LCG may provide in several years.





Documentation



➤ *Documentation:*

➤ *Home: <http://octopus.objectweb.org>*

➤ *Atlas Wiki: <https://uimon.cern.ch/twiki/bin/view/Atlas/DatabaseReplication>*

➤ *Distribution:*

➤ *[/afs/cern.ch/user/h/hrivnac/public/OctopusReplicator](afs/cern.ch/user/h/hrivnac/public/OctopusReplicator)*