

Replica Optimisation

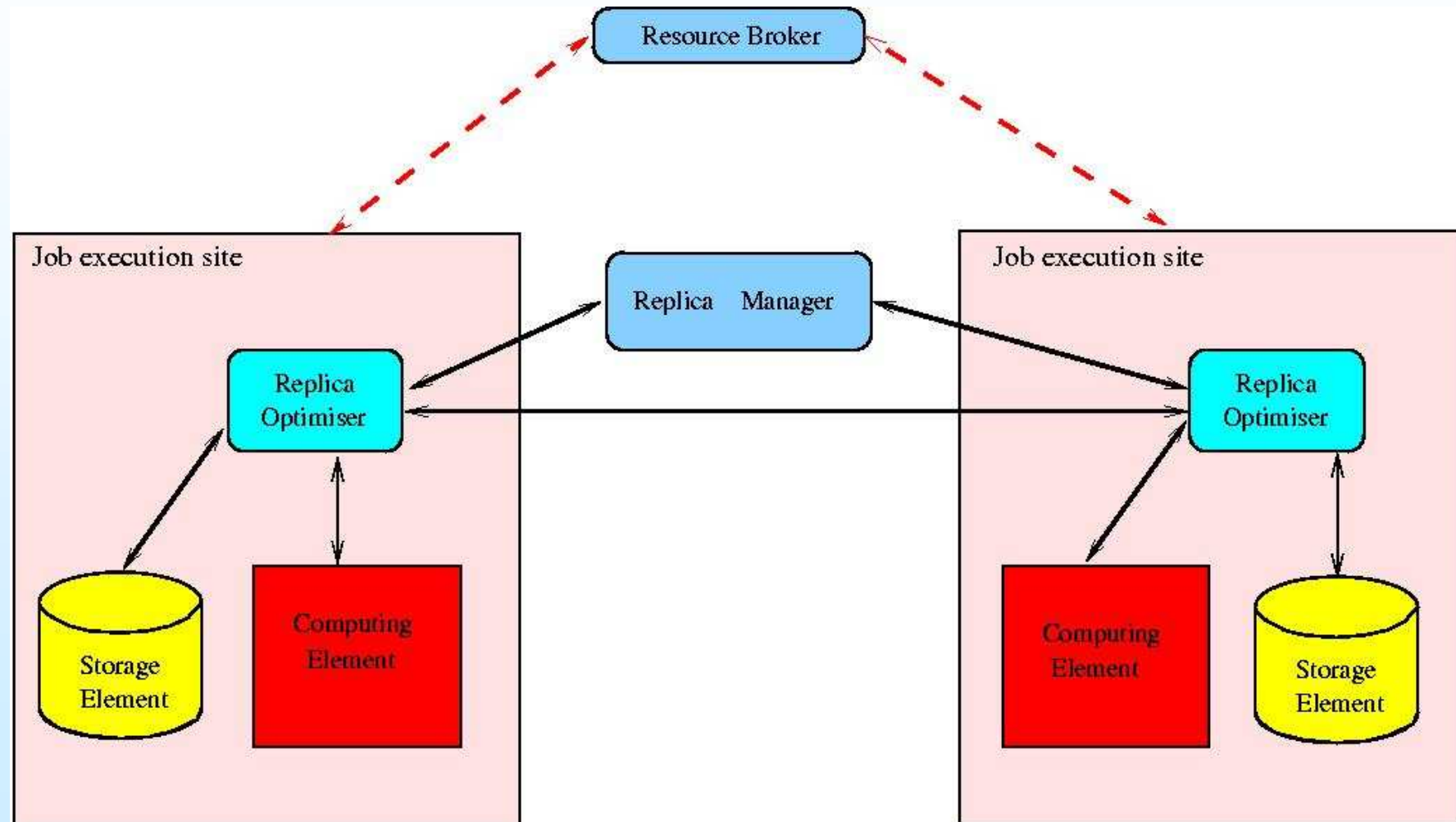
OptorSim vs. Optor

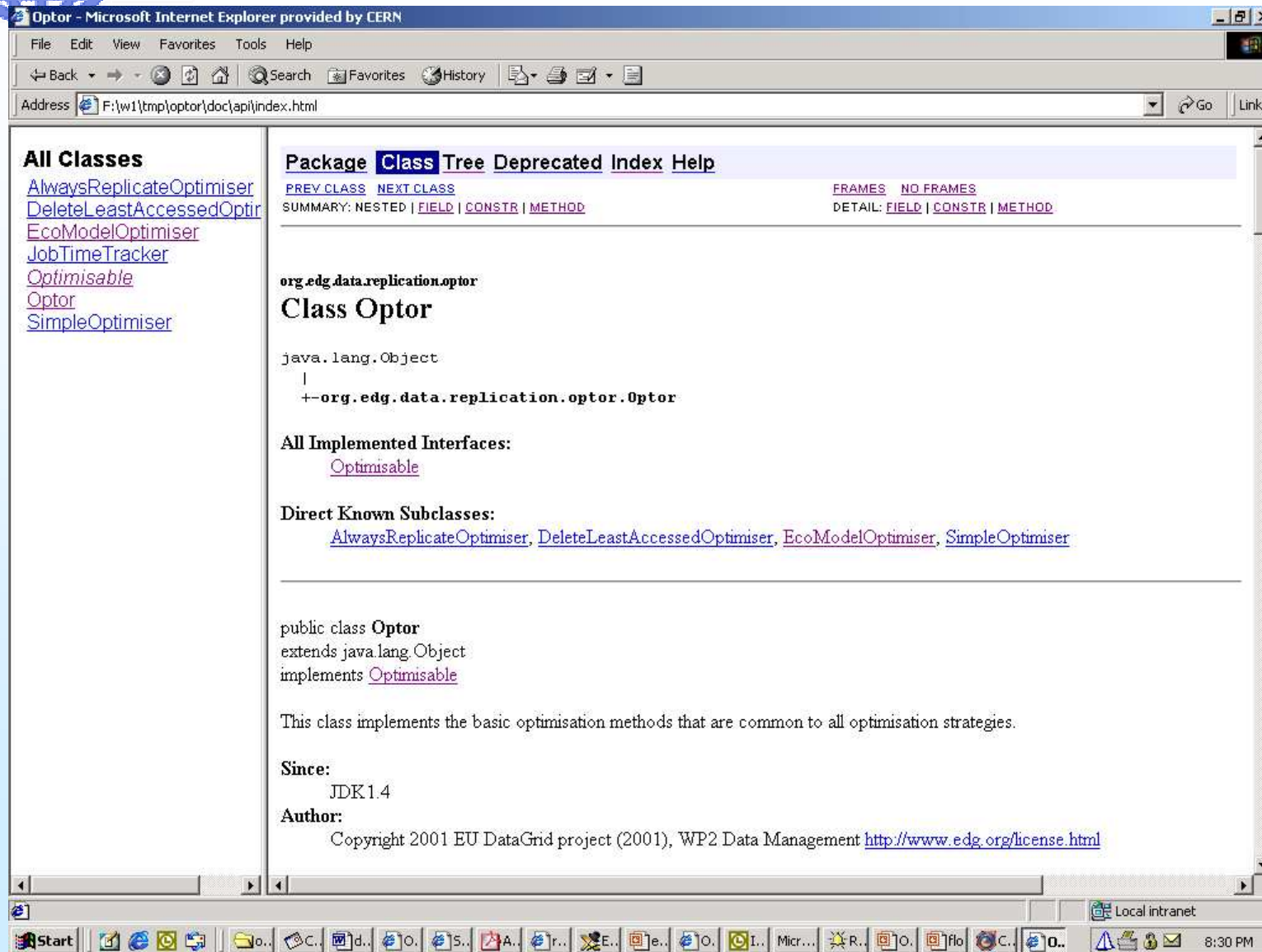
W. H. Bell, D. G. Cameron, R. Carvajal,
A. P. Millar, K. Stockinger, F. Zini

Optimisation of data access and replication based on an **economic model**

- View files as digital assets which can be bought and sold for profit
- Let nodes on the grid interact according to an economic model and thus optimise the system implicitly rather than explicitly







Optor - Microsoft Internet Explorer provided by CERN

File Edit View Favorites Tools Help

Address [F:/w1/tmp/optor/doc/api/index.html](file:///F:/w1/tmp/optor/doc/api/index.html)

All Classes

- [AlwaysReplicateOptimiser](#)
- [DeleteLeastAccessedOptir](#)
- [EcoModelOptimiser](#)
- [JobTimeTracker](#)
- [Optimisable](#)
- [Optor](#)
- [SimpleOptimiser](#)

Package **Class** **Tree** **Deprecated** **Index** **Help**

[PREV CLASS](#) [NEXT CLASS](#) [FRAMES](#) [NO FRAMES](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#) [DETAIL: FIELD](#) | [CONSTR](#) | [METHOD](#)

org.edg.data.replication.optor

Class Optor

java.lang.Object

```

|
+--org.edg.data.replication.optor.Optor
    
```

All Implemented Interfaces:

- [Optimisable](#)

Direct Known Subclasses:

- [AlwaysReplicateOptimiser](#), [DeleteLeastAccessedOptimiser](#), [EcoModelOptimiser](#), [SimpleOptimiser](#)

```

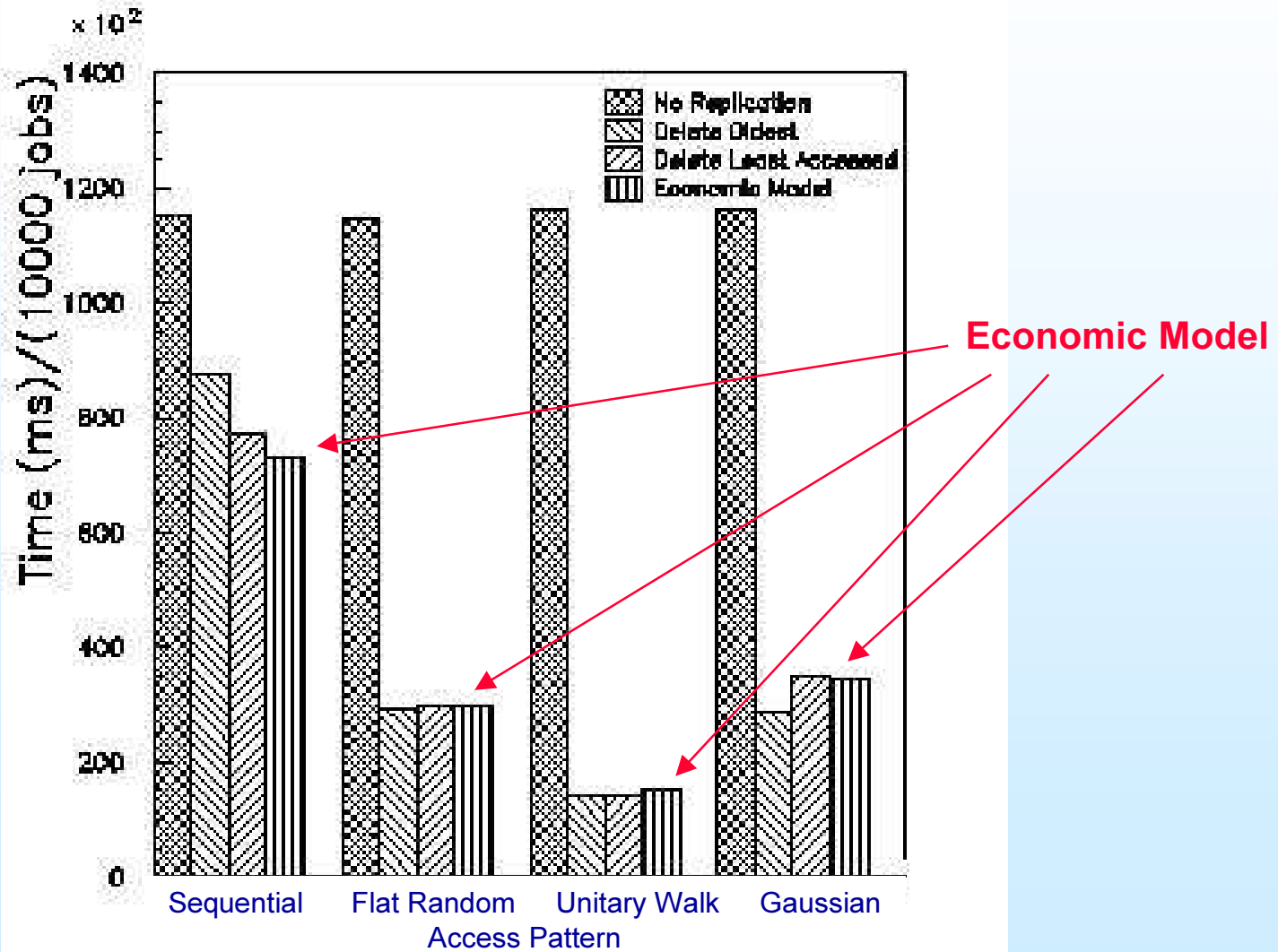
public class Optor
extends java.lang.Object
implements Optimisable
    
```

This class implements the basic optimisation methods that are common to all optimisation strategies.

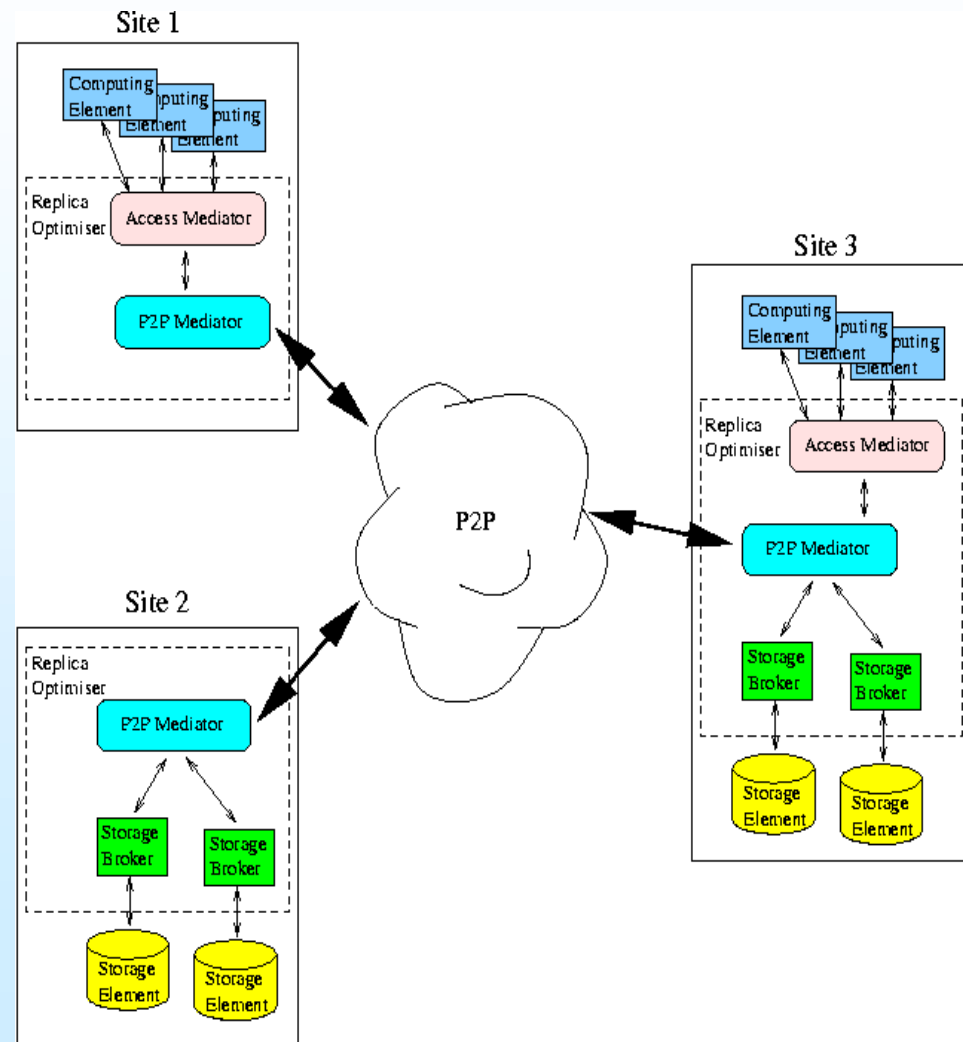
Since: JDK1.4

Author: Copyright 2001 EU DataGrid project (2001), WP2 Data Management <http://www.edg.org/license.html>

Local intranet 8:30 PM

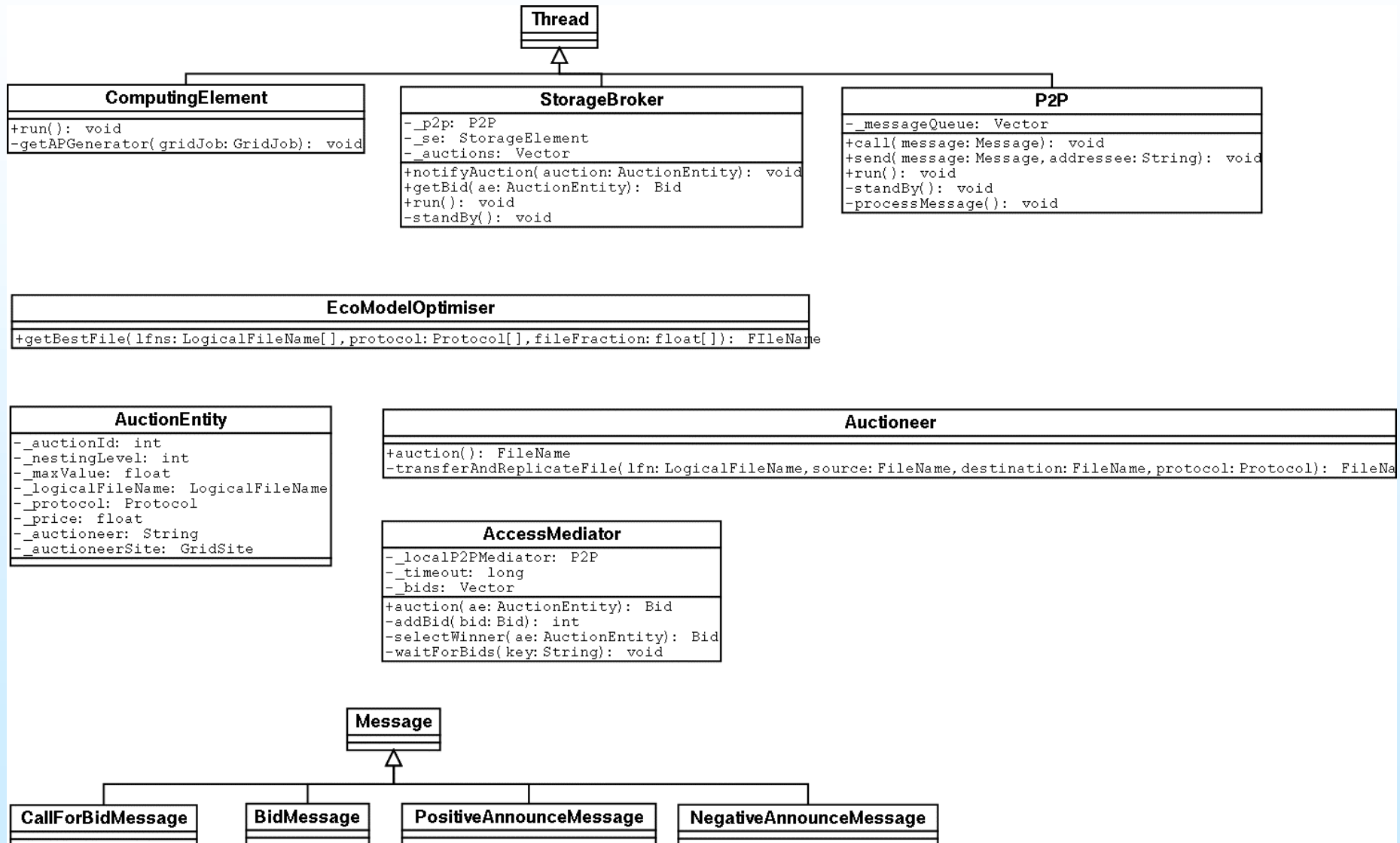


- **Access Mediator (AM)** - contacts replica optimisers to locate the cheapest copies of files and makes them locally available
- **Storage Broker (SB)** - manages files stored in storage element, trying to maximise profit for the finite amount of storage space available
- **P2P Mediator (P2PM)** - establishes and maintains P2P communication between grid sites





OptorSim: Auction ClassDiagram





Methods on SE:

- `addFile()`
- `reserveSpace()`
- `addPreReservedFile()`
- `removeFile()`

- `pinFile()` `unpinFile()`
- `updateAccessHistory()`

Optimisation methods for replica management:

- `getLeastAccessedDeletableFile()`
- `getLeastValuableDeletableFile()`
- `getOldestDeletableFile()`



Demo: Replica Selection with Optor based on Network Bandwidth

```
ReptorServer rs = (ReptorServer) Naming.lookup(serverURL+"/ReptorServer");
```

```
// register first replica of lfn  
rs.registerEntry(lfn[0], fn[0]);
```

```
...
```

```
// produce replicas of lfn
```

```
FileHandler fileHandler[] = new GridFTPFileHandler[5];
```

```
fileHandler[0] = new GridFTPFileHandler();
```

```
fileHandler[0].setSource(fn[0]);
```

```
fileHandler[0].setDestination(fn[1]);
```

```
rs.replicateFile(lfn[0], fn[0], fn[1], fileHandler[0]);
```

```
...
```

```
// select best file
```

```
String destinationSE = "testbed008.cern.ch";
```

```
FileName[] bestFn = rs.getBestFile(lfn, destinationSE, fileHandler, fileFraction);
```