



## ***Persistence***



## ***What is persistence?***

- ***Persistence is the ability of an object to survive the lifetime of the OS process in which it resides.***
- ***Persistence is relevant for objects with an internal state.***
- ***The state needs to be retained between object deactivation and object activation***



## ***How to achieve persistence?***

- ***Should be transparent to application developer***
- ***Storing object state on persistent storage before de-activation***
- ***Upon activation, load object state from persistent storage***
- ***Persistent storage can be***
  - *File system*
  - *Relational Database*
  - *Object-Database*
  - *Flash-RAM*

© Wolfgang Emmerich, 1997

3



## ***Externalization***

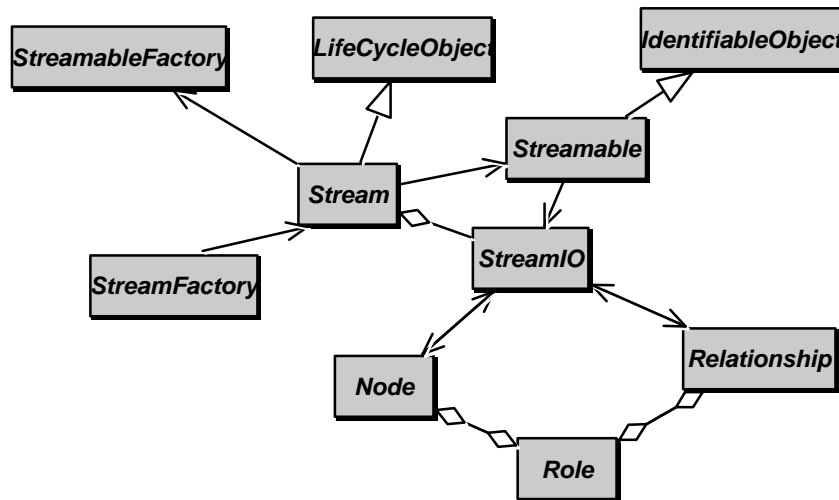
- ***Technique to***
  - *write composite objects into a byte stream*
  - *load composite objects from a byte stream*
- ***Byte stream can then be written to/read from the file system***
- ***Supported by several CORBA products***
- ***Also used to store Java objects***

© Wolfgang Emmerich, 1997

4



## Design of Externalization Interfaces



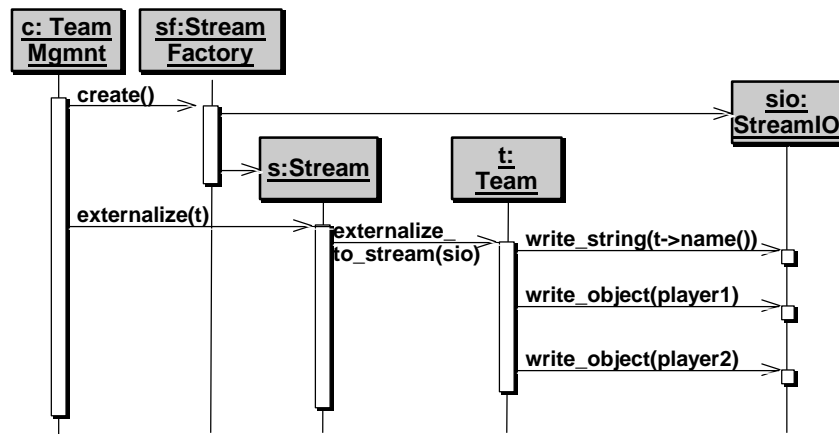
© Wolfgang Emmerich, 1997

5



## Externalization Scenario

### ■ Externalize Team object (with its players):



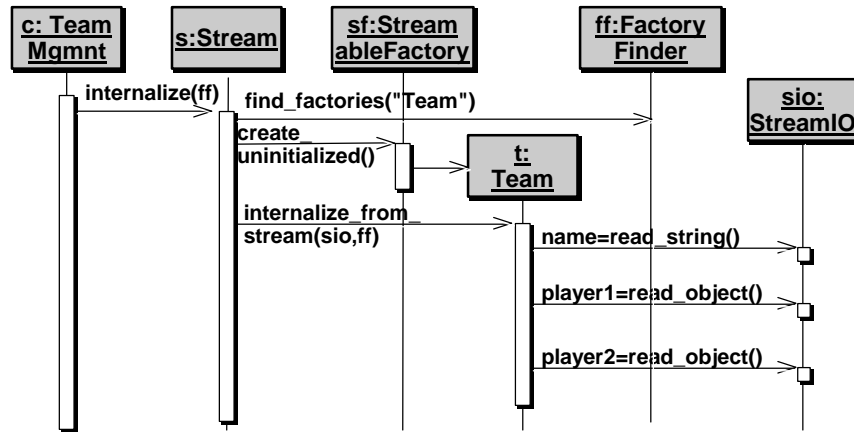
© Wolfgang Emmerich, 1997

6



## Internalization Scenario

### ■ Internalize Team object from a stream:



© Wolfgang Emmerich, 1997

7



## Mapping to DBMSs

- **Need for mapping middleware object model to DBMS data model**
  - *Straight-forward for object-oriented DBMSs*
  - *Clumsy for relational DBMSs*
- **A schema needs to be established in the DBMS**
- **For each object type:**
  - *implement attributes in schema*
  - *implement operations in schema (if possible)*

© Wolfgang Emmerich, 1997

8



## Mapping to RDBMSs

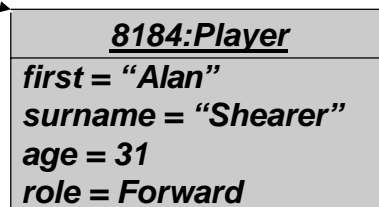
- **Relational database schemas consist of sets of tables**
- **Define a table for each object type**
- **In each table create**
  - **primary key for object identifier**
  - **a column for each attribute of the object**
    - **mapping of middleware atomic types to primitive types supported by RDMBS**
    - **secondary keys for object references**
- **Resolve inheritance statically**

© Wolfgang Emmerich, 1997

9



## RDBMS Mapping Example



Key	Player
1	8987
2	8184

Key	first	surname	age	role
8987	Jürgen	Klinsman	34	4
8184	Alan	Shearer	31	4

© Wolfgang Emmerich, 1997

10



## ODBMS

- **ODBMSs have been standardized by the Object Database Management Group**
  - *Schema definition language (ODL)*
  - *Programming language bindings to*
    - *C++*
    - *Java*
    - *Smalltalk*
  - *Object Query Language (OQL)*
- **Support persistence of OO programming language objects**

© Wolfgang Emmerich, 1997

11



## Mapping to ODBMSs

- **ODL is a superset of OMG/IDL**
- **Programming language bindings of ODBMSs are also supported by CORBA**
- **CORBA portable object adapter supports ODBMSs**
- **ODBMS objects can be**
  - *clients of CORBA objects*
  - *servers for CORBA objects*

© Wolfgang Emmerich, 1997

12



## Assessment

- *Externalization is not transparent for implementors of server objects*
- *Persistence in RDBMSs is*
  - *complicated by impedance mismatch*
  - *simplified by wide availability of RDBMSs*
- *Persistence in ODBMSs is*
  - *simplified by conceptual similarities of*
    - *object models*
    - *programming language bindings*