

Распределенные вычислительные системы

Лекция №8: Долговременное хранение распределенных объектов

Алексей В. Бурдаков, к.т.н.
burdakov@usa.net

План лекций

№	Дата	Тема
1	04.09	Вводная лекция
2	11.09	Эволюция распределенных технологий
3	18.09	Принципы ПО среднего слоя
4	25.09	Стандарты OMG, CORBA и ORB
5	02.10	Пример приложения на CORBA
6	09.10	COM, Java/RMI
7	16.10	Определение местонахождения
	23.10	Перенос
8	30.10	Долговременное хранение + <u>Рейтинг по Л.1-7</u>
9	06.11	Лекция

План лекций (продолжение)

№	Дата	Тема
11	13.11	Лекция
12	20.11	Лекция
13	27.11	Лекция
14	04.12	Лекция
15	11.12	Лекция
16	18.12	Лекция
17	25.12	Зачет

План лекции

- Концепция долговременного хранения объектов
- Применение хранилищ данных
 - Файлы
 - РСУБД
 - ОСУБД

Объекты с внутренним состоянием и без него

- Объекты в распределенной вычислительной системе могут быть:
 - С внутренним состоянием (statefull)
 - Атрибуты в определении интерфейса – достаточное, но не необходимое
 - Внутренние переменные экземпляра
 - Без внутреннего состояния (stateless)
 - Долговечными (persistent) – продолжающие существовать и после завершения процесса
 - Использование энергонезависимой памяти
 - Временными (transient)
 - Хранение в энергозависимой памяти
- Пример: служба именования
 - statefull
 - persistent

Долговременное хранение (persistence)

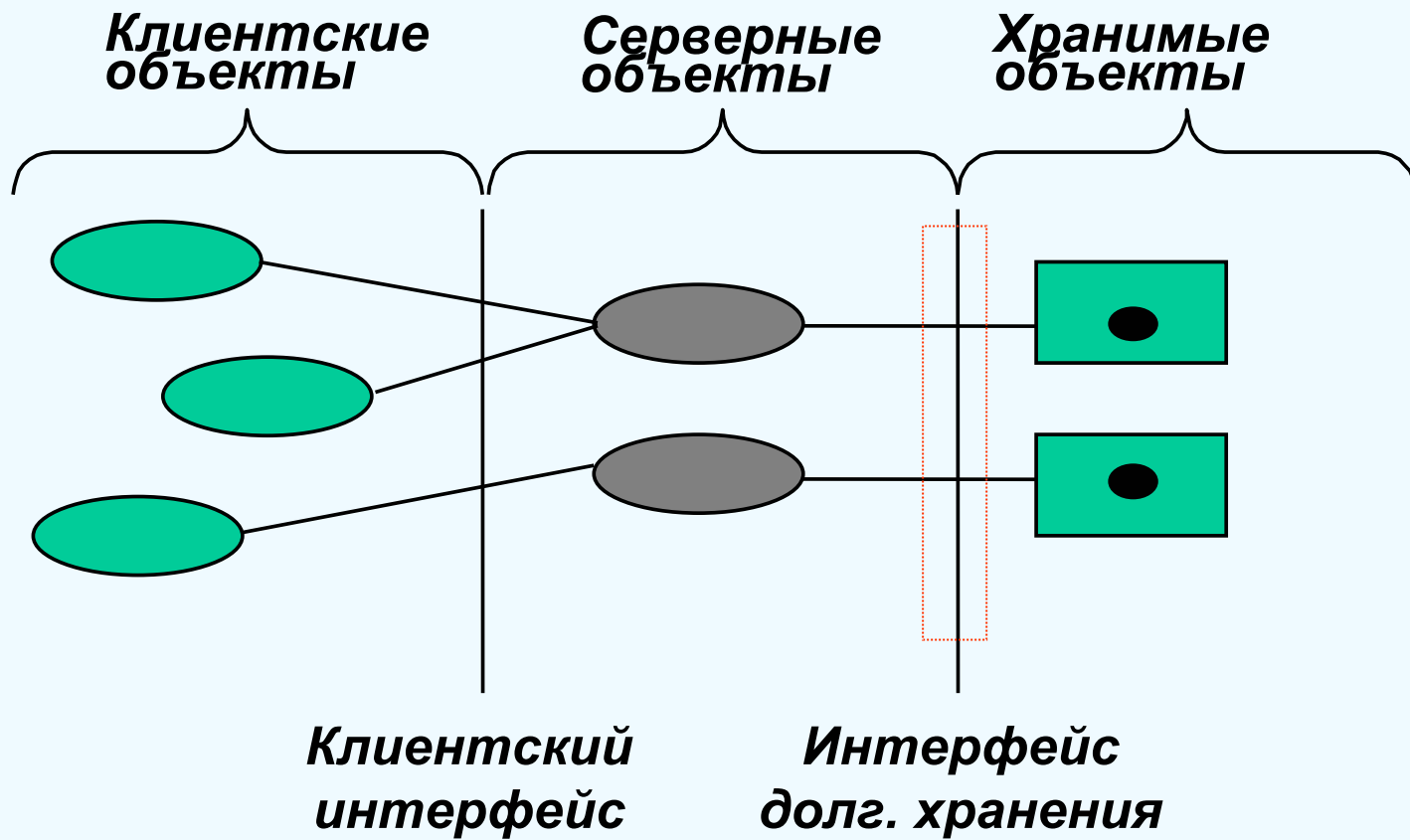
- Долговременное хранение - способность объекта переживать процесс, в котором он находится
- Состояние должно запоминаться на промежуток деактивации и активации

Как реализовать долговременное хранение?

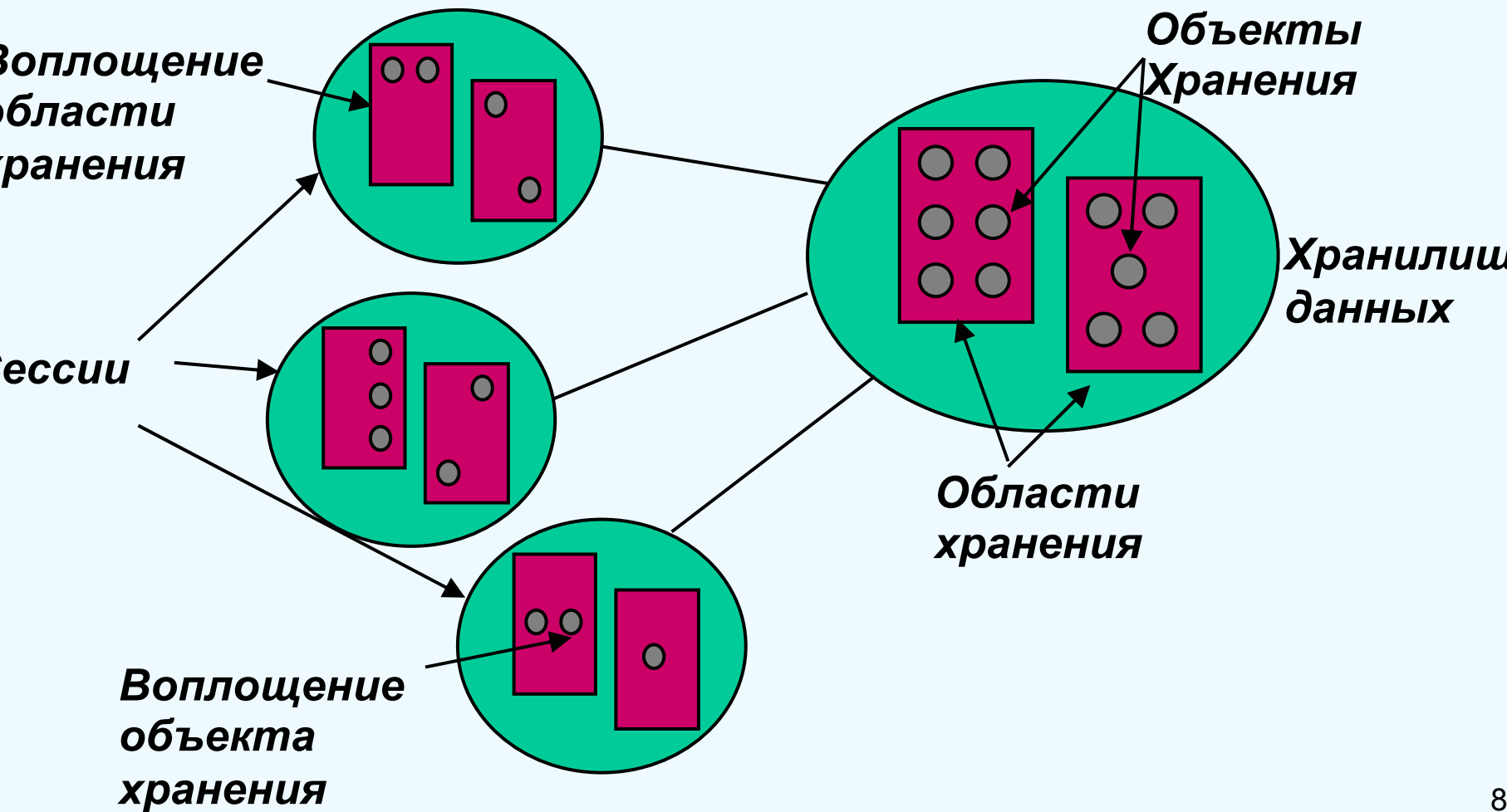
- Сохранить объект на постоянном носителе до деактивации
- Восстановить объект с постоянного носителя после активации
- Постоянным может быть:
 - Файловая система
 - Реляционная СУБД
 - ООСУБД
 - Флэш-память

Прозрачность долговременного хранения

- Долговременное хранение должно быть прозрачным как для пользователей, так и для проектировщиков клиентских объектов



Концепции долговременного хранения



Основные понятия

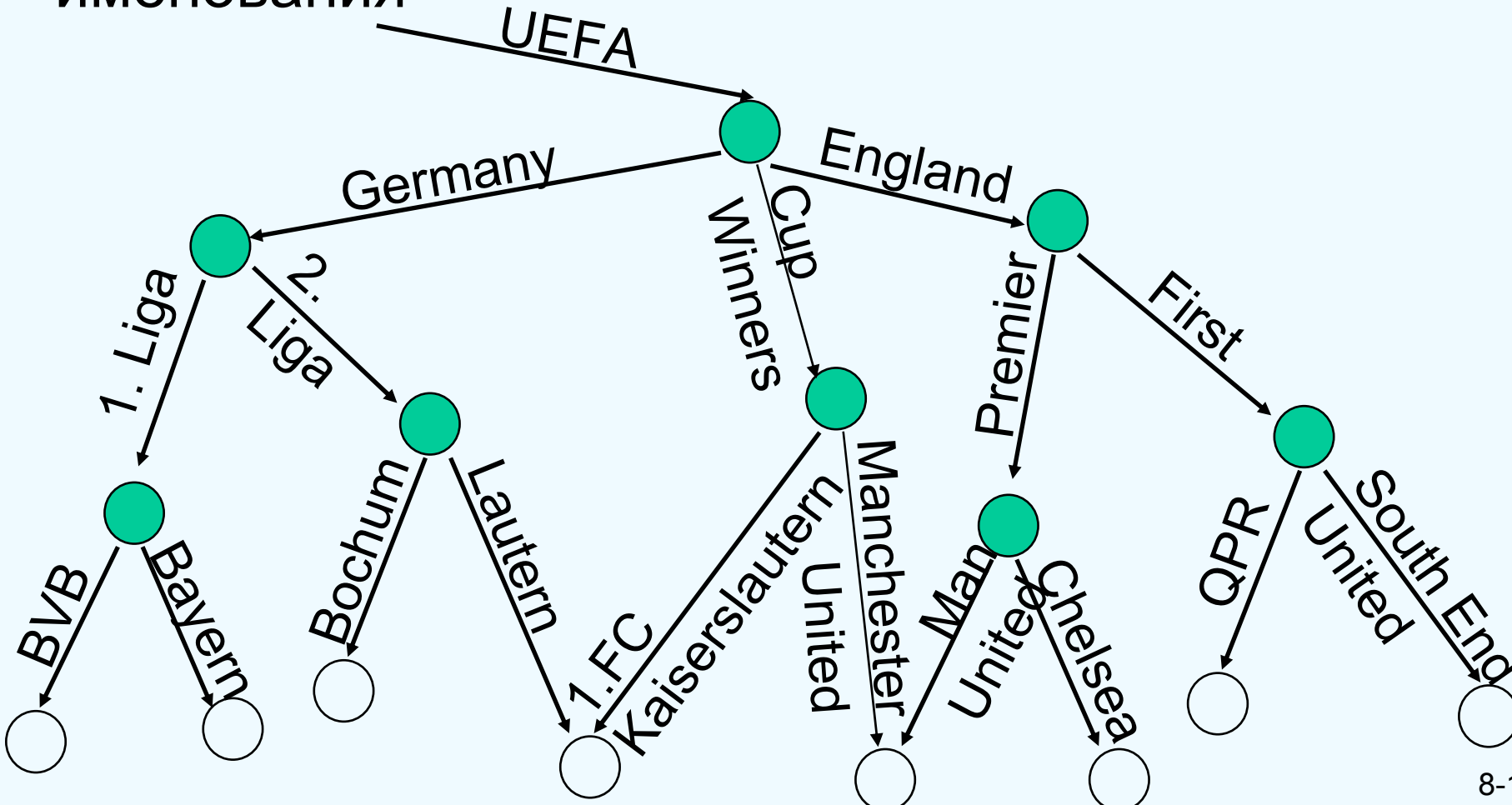
- *Хранилище данных (datastore)* - абстрактное понятие, охватывающее различные технологии хранения - от файлов до ООБД, пример: БД
- *Область хранения (storage home)* - контейнер, содержащий долговременные представления хранимых объектов, пример: таблица
- *Объект хранения (storage object)* - представление состояния объекта-сервера в области хранения, пример: строка
- *Тип хранения (storage type)* - определяет интерфейс объекта хранения

Основные понятия

- *Воплощение объекта хранения (storage object incarnation)* - представление объекта хранения в процессах, является объектом языка программирования
- *Воплощение области хранения (storage home incarnation)* - представление экземпляра области хранения на языке программирования
- *Ключ* - атрибут, однозначно идентифицирующий объект хранения в пределах области хранения
- *Сессия* - логическая связь между хранилищем данных и процессом, в котором выполняется объект-сервер

Пример

- Долговременное хранение объектов сервиса именования



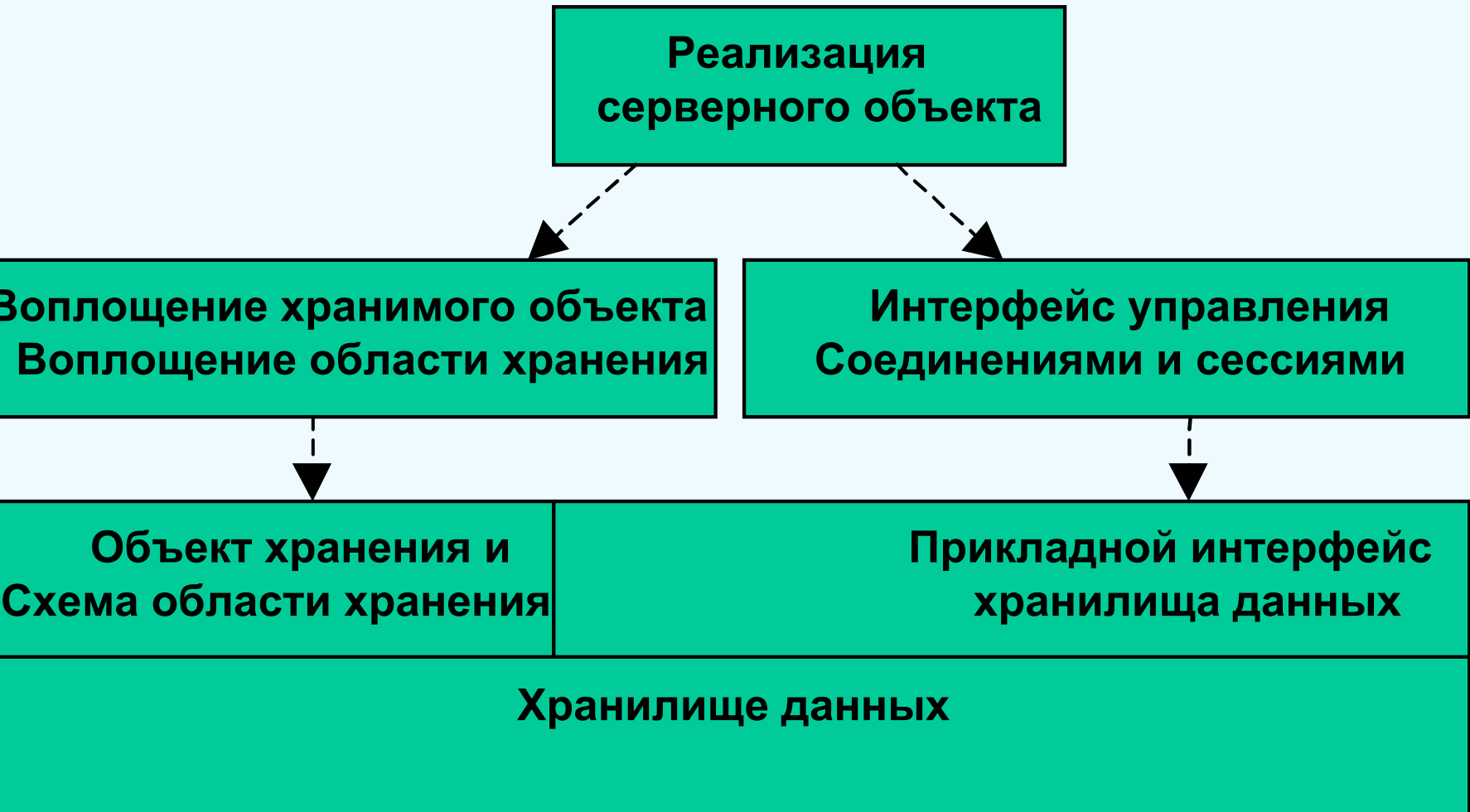
PSSDL - Язык определения службы пост. хранения

- *PSSDL (Язык определения состояний службы
долговременного хранения)* - строгое
супермножество языка CORBA/IDL
- Дополнительно обеспечивает определение:
 - объектов хранения
 - областей хранения
 - ключей

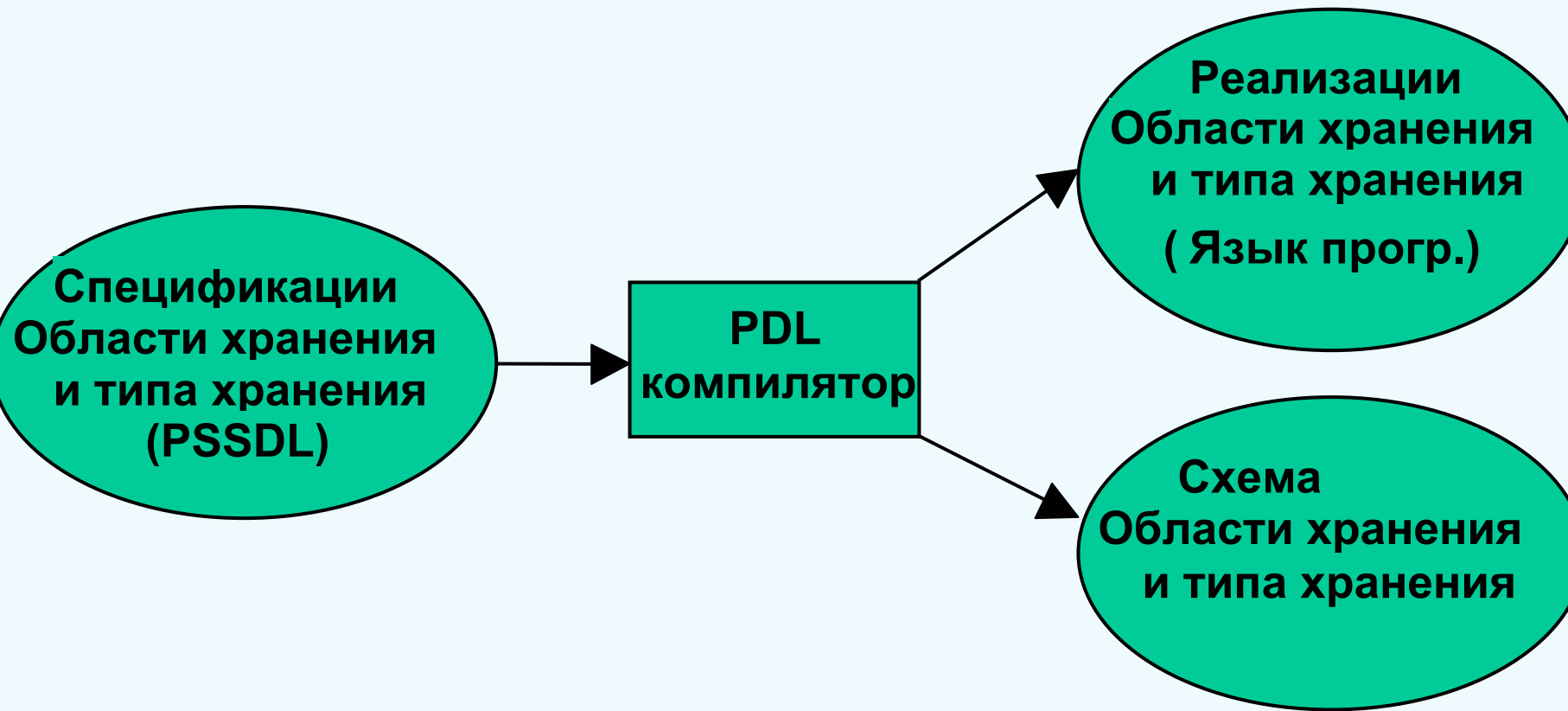
Определение типов хранения

```
// file NamingContextState.pssdl
#include <CosNaming.idl>
storagetypespec NamingContextState{
    Object resolve(in CosNaming::NameComponent n)
        raises(CosNaming::NotFound, CosNaming::CannotProceed,
            CosNaming::InvalidName);
    void bind(in CosNaming::NameComponent n, in Object obj)
        raises(CosNaming::NotFound, CosNaming::CannotProceed,
            CosNaming::InvalidName, CosNaming::AlreadyBound);
    void unbind(in CosNaming::NameComponent n, in Object obj)
        raises(CosNaming::NotFound, CosNaming::CannotProceed,
            CosNaming::InvalidName);
    void rebind(in CosNaming::NameComponent n, in Object obj)
        raises(CosNaming::NotFound, CosNaming::CannotProceed,
            CosNaming::InvalidName); };
Storagehomespec NamingContextStateHome of NamingContextState {
    ref <NamingContextState> create();
};
```

Архитектура сервиса постоянного хранения



Компилятор с языка PDL



Применение хранилищ данных для организации постоянного хранения

Технологии хранения

- Файлы
 - Экстернализация CORBA
 - Сериализация Java
 - Структурированное хранение COM
- Реляционные СУБД
 - Объектно-реляционное отображение
 - JDBC/ODBC
- Объектные БД
 - Versant
 - O2

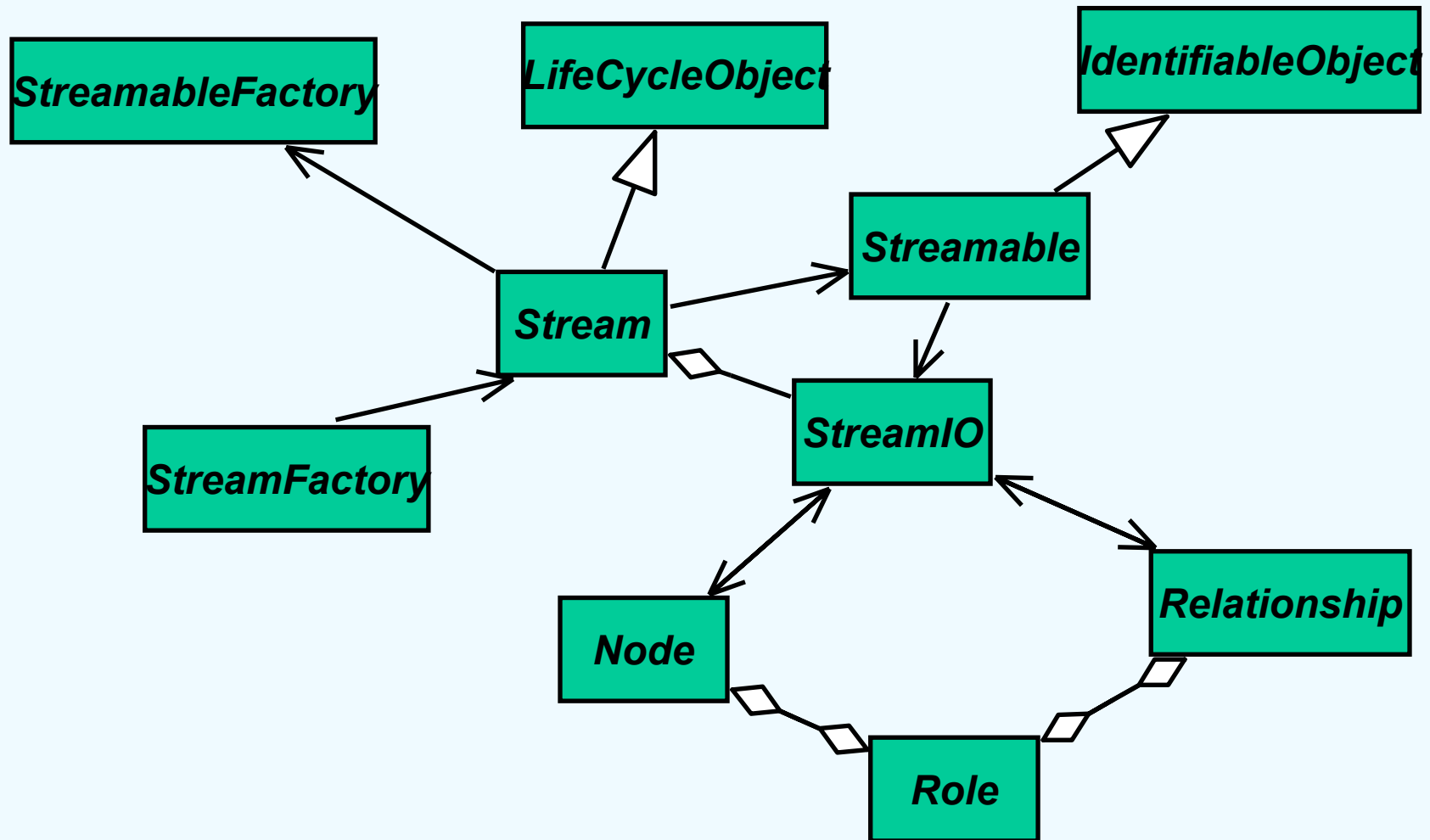
Экстернализация в CORBA

- Технология, позволяющая:
 - Записывать композитные объекты в бинарного потока
 - Считывать композитные объекты из бинарного потока
- Бинарный поток может быть сохранен в или считан с файловой системы
- Поддерживается реализациями CORBA (служба экстернализации)

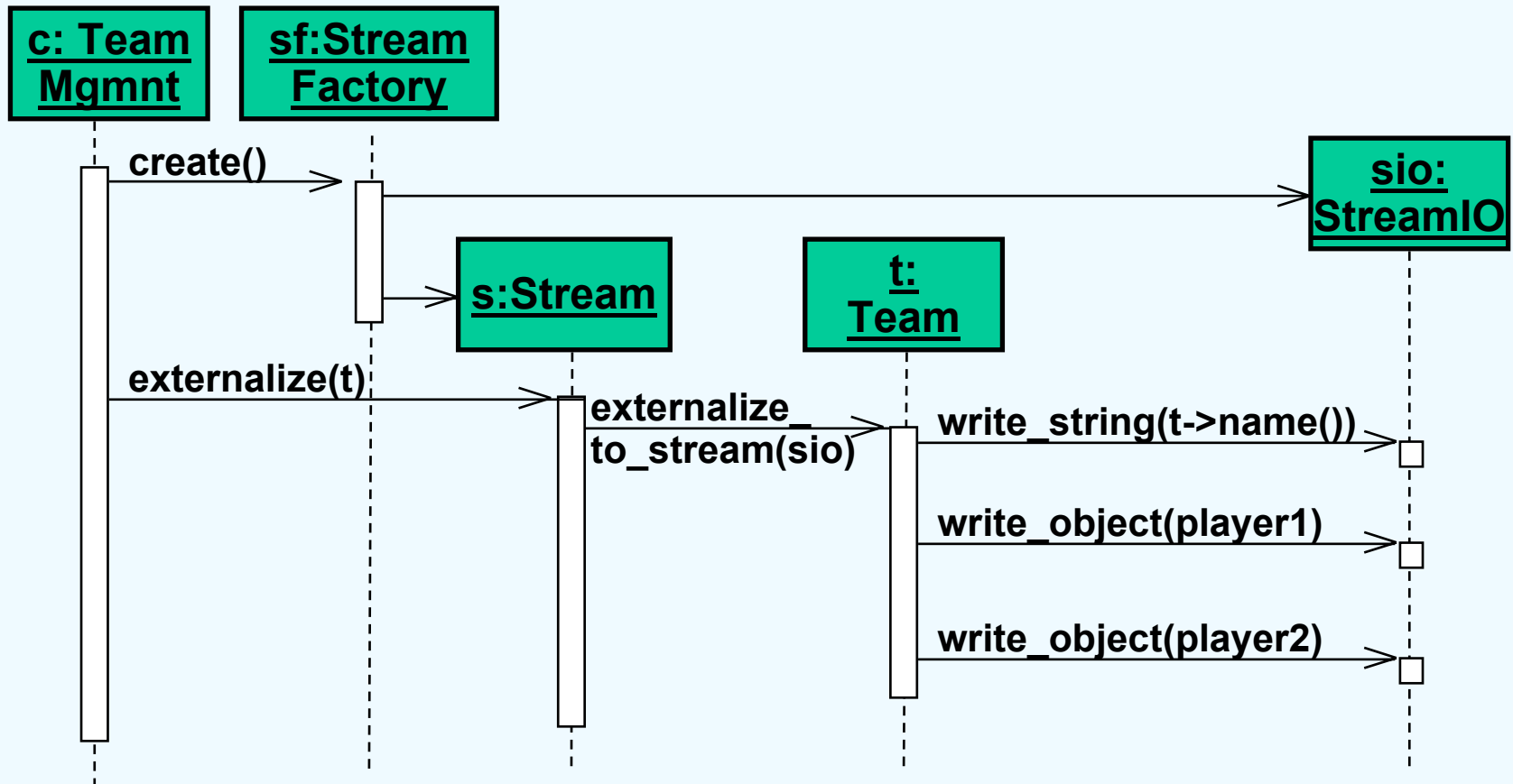
Служба экстернализации в CORBA

- Спецификация принята в 1994 г.
- Для обеспечения интероперабельности различных реализаций спецификация стандартизировала:
 - Формат хранения атомарных типов данных
 - Способ представления сконструированных данных

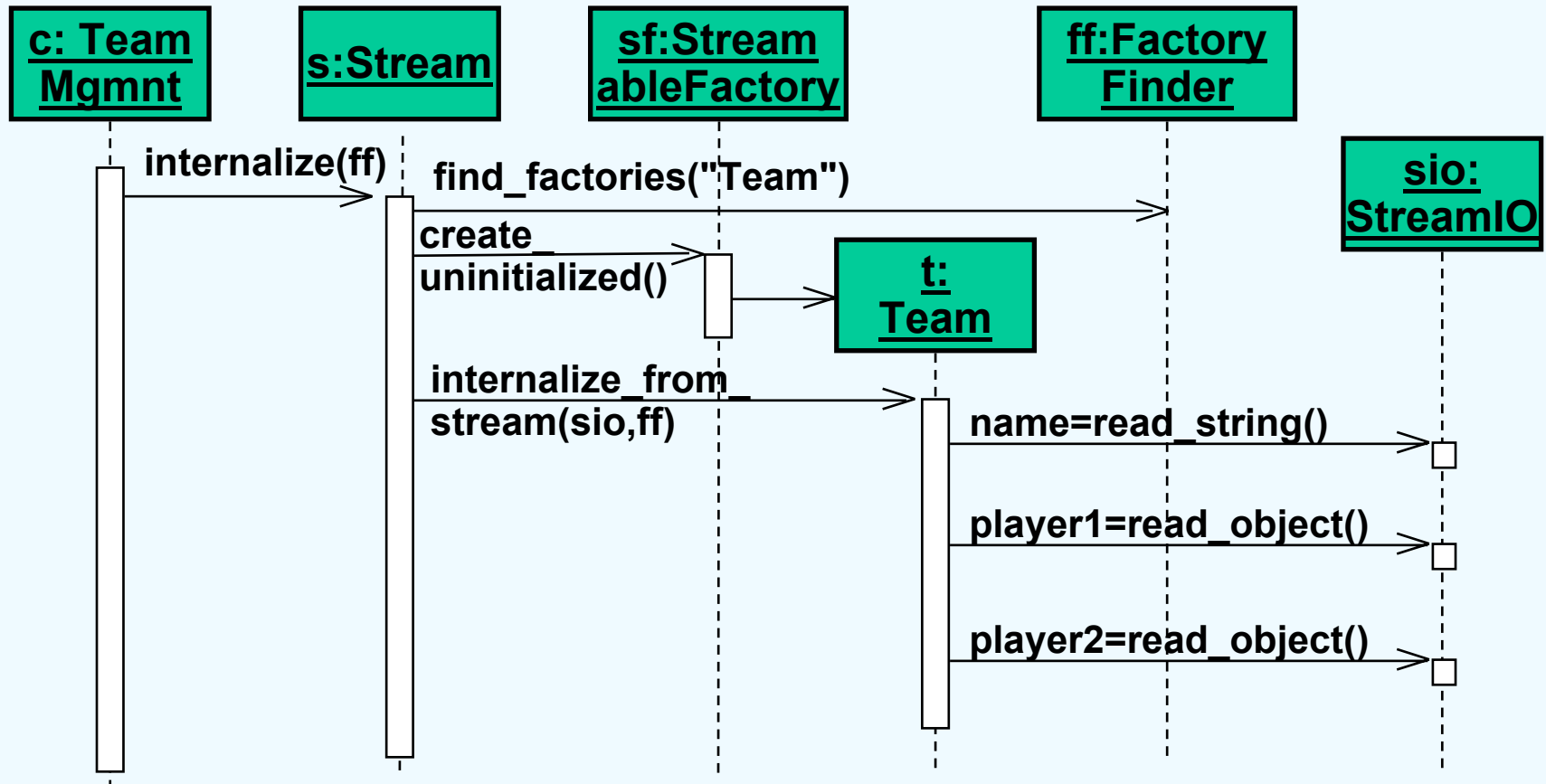
Интерфейсы экстернализации



Сценарий экстернализации



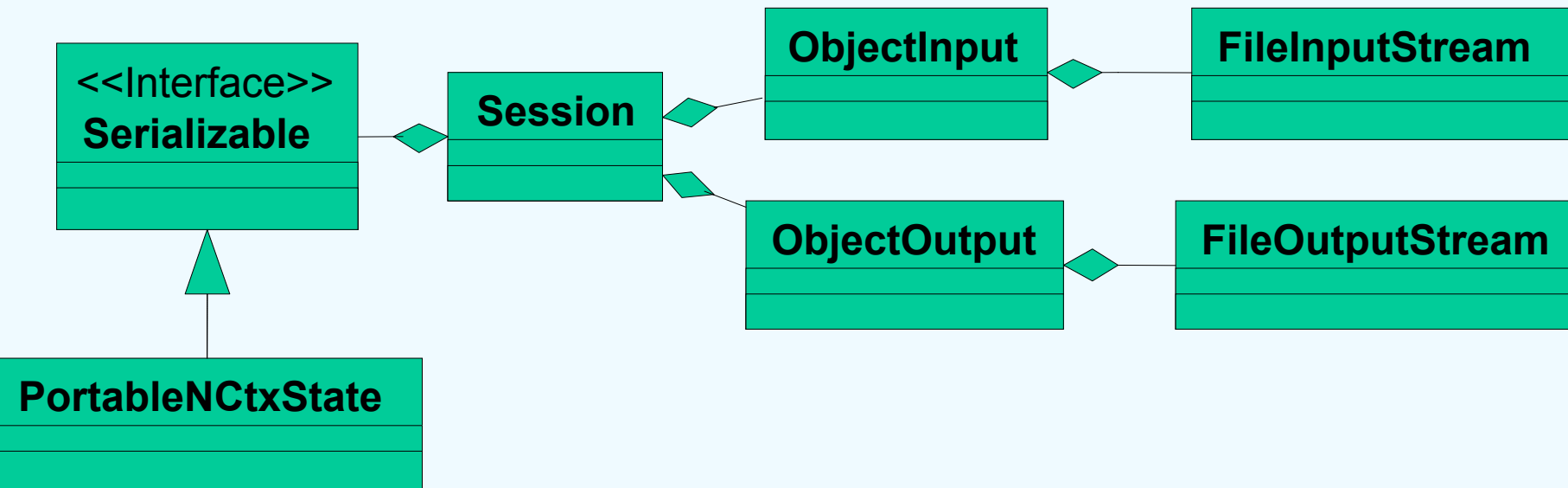
Сценарий интернализации



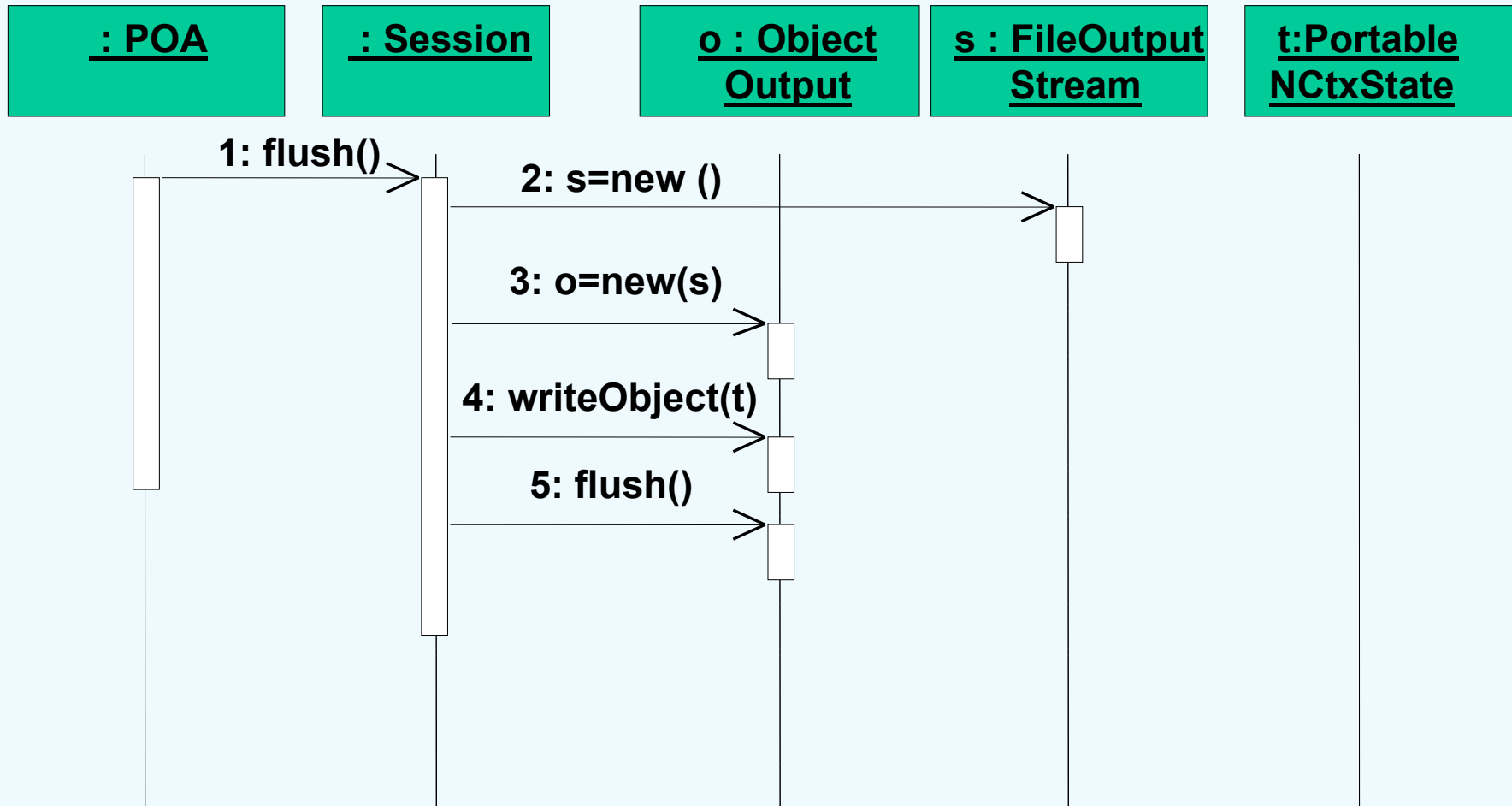
Сериализация в Java

- Перевод объектов Java в поток
- Отслеживает все объекты на которые есть ссылки
- Поток может быть записан в файл
- Объекты должны реализовывать интерфейс `Serializable`
- Атрибуты, которые нет необходимости хранить долговременно, могут быть продекларированы в определении класса как *transient*

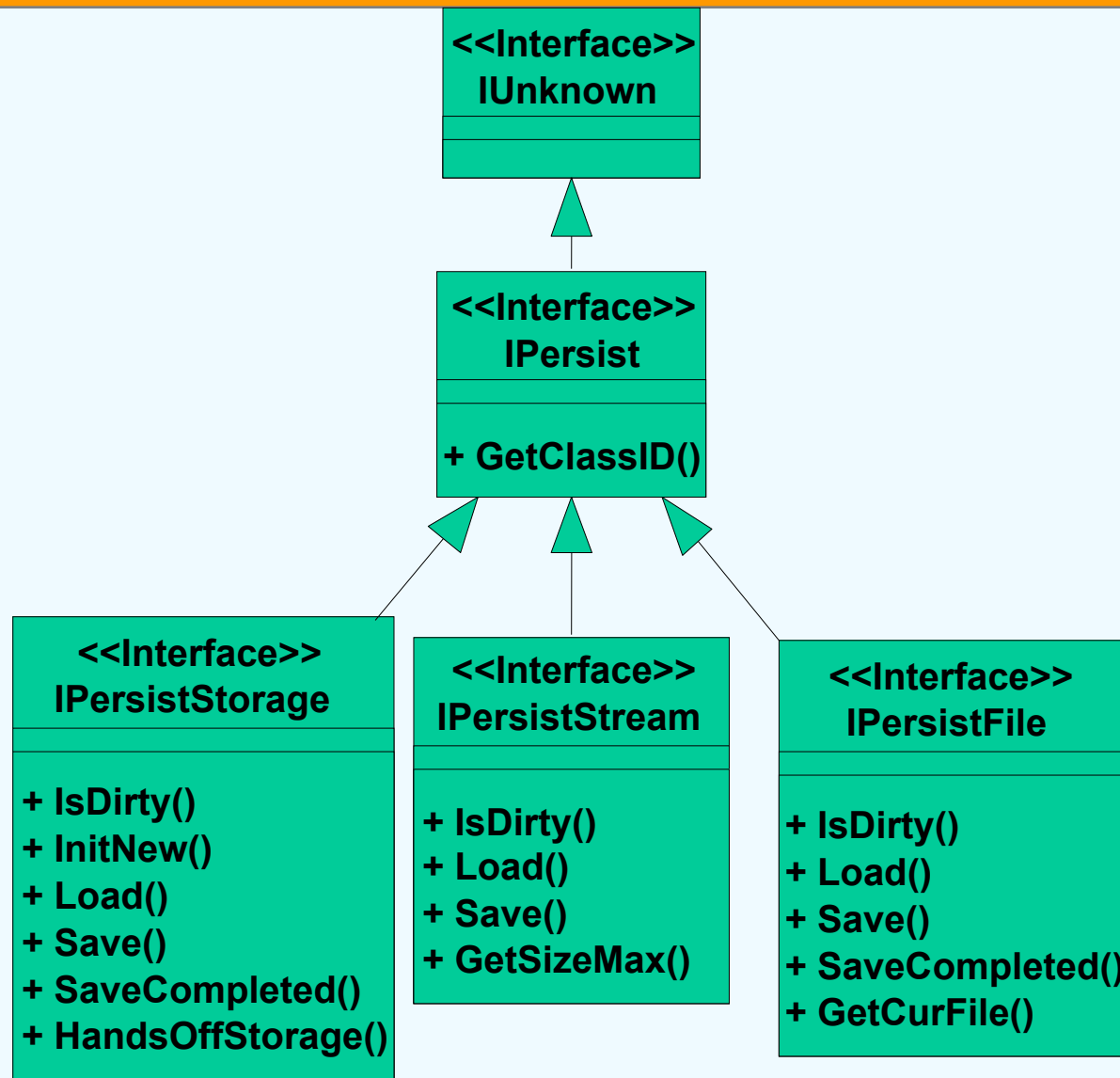
Сериализация в Java: Типы



Сценарий сериализации в Java



Интерфейсы долговременного хранения в COM



Структурированное хранение в COM

- COM также поддерживает структурированное хранение объектов
- COM поддерживает композицию объектов – помещение объектов в объекты – контейнеры (например объект Word содержащий Excel)
- Позволяет применять инкрементную загрузку и изменение

Проблемы, связанные с долговременным хранением в файлах

- Отображение в файлы может быть неэффективным (большие издержки на обработку потоков) для больших композитных объектов
- Файловые системы обеспечивают примитивный и слабый контроль за одновременным доступом (например, нет контроля за взаимными блокировками и т.п.)
- Файловые системы не обеспечивают отказоустойчивость (транзакции)

РСУБД

- Примеры РСУБД:
 - Ingres
 - Oracle
 - Sybase
 - DB2
 - Microsoft SQL Server
 - Microsoft Access
- Основаны на реляционной модели
 - Таблицы
 - Кортежей
 - Неструктурированных (атомарных) атрибутов

Отображение в РСУБД

- Схема РСУБД состоит из набора таблиц
- Определяется таблица для каждого типа
- В каждой таблице создается
 - Первичный ключ для идентификации объекта
 - Колонка для каждого атрибута объекта
 - Отображение атомарных типов ПО среднего слоя на примитивные типы, поддерживаемые РСУБД
 - Вторичные ключи для объектных ссылок
- Статическое разрешение наследования

Отображение в РСУБД: Пример

- Создать схему для данных сервиса именованя:

```
CREATE TABLE NamingCtxts (id TEXT, objref TEXT,  
    CONSTRAINT c1 PRIMARY KEY (id)  
)
```

```
CREATE TABLE Bindings (  
    id TEXT, name TEXT, kind TEXT, objref TEXT,  
    CONSTRAINT c2 PRIMARY KEY(id)  
)
```

```
CREATE TABLE NameBindings (  
    nctx TEXT, bind TEXT,  
    CONSTRAINT c3 FOREIGN KEY(nctx) REFERENCES NamingCtxts(id),  
    CONSTRAINT c4 FOREIGN KEY(bind) REFERENCES Bindings(id)  
)
```


Таблицы хранилища сервиса именования

NameBindings

nctx	bind	<i>Bindings</i>				<i>NamingCtx</i>	
7	10	id	id	kind	objref	id	objref
8	11	1	UEFA	NCtx	lasd098asjasfd08lasdsfhasdf0	1	lasd098asjasfd08lasdsfhasdf0
8	12	10	Bayern	Team	asldhf2y4395-qFDOhd	2	kljsadpoueurasldkfjasofduqels
4	13	11	Bochum	Team	kasdughf92ry5u-0ASP	3	aslkjdksahdfksgdf9qeijksadhfs
4	14	12	Lautern	Team	kasjsdhf923r5-00OSADFISAG	4	ashkw09we4ty0slajdfhisafsgfs
5	15	13	Kaiserslautern	Team	kasjsdhf923r5-00OSADFISAG	5	saldkjfdasldkjfalsdkfh0828lsdj
5	16	14	Manchester United	Team	lasasaskjhasldjfadfh293ruasld	6	sosdihfw9y4t02-1jwohfkasjdnf
6	17	15	Man United	Team	lasasaskjhasldjfadfh293ruasld	7	lsadkfh924357109ejaojkasbja
6	18	16	Chelsea	Team	ksajdfgw4y250-2=0sdhfkasjgs	8	asldkjfh29y35r-q[ofdhdasldgj2
1	2	17	QPR	Team	aksdhf9245-qSIDHFOASFash		
1	3	18	South End United	Team	33i4y50wswfahsdfjh29345yoa		
1	4	2	Germany	NCtx	kljsadpoueurasldkfjasofduqels		
3	5	3	England	NCtx	aslkjdksahdfksgdf9qeijksadhfs		
3	6	4	Cup Winners	NCtx	ashkw09we4ty0slajdfhisafsgfs		
2	7	5	Premier	NCtx	saldkjfdasldkjfalsdkfh0828lsdj		
2	8	6	First	NCtx	sosdihfw9y4t02-1jwohfkasjdnf		
7	9	7	1. Liga	NCtx	lsadkfh924357109ejaojkasbja		
		8	2. Liga	NCtx	asldkjfh29y35r-q[ofdhdasldgj2		
		9	BVB	Team	ksaldfhqweoiry[pa\sonasglsas		

Запросы

- Разрешение привязки имен

```
SELECT Bindings.objref
FROM Bindings, NameBindings, NamingCtxts
WHERE NamingCtxts.objref="lsadkfh924357109ejaojkasbja" AND
      Bindings.name="BVB" AND
      Bindings.id=NameBindings.bind AND
      NamingCtxts.id=NameBindings.nctx
```

- Получение всех привязок для контекста

```
SELECT Bindings.name_id,
      (SELECT name_id
       FROM Bindings
       WHERE Bindings.id=NameBindings.naming_context)
FROM NameBindings, Bindings
WHERE NameBindings.binding=Bindings.id
```

Реализация запросов в программах

- Внедренный SQL
- ODBC (Open Database Connectivity – Microsoft)
- JDBC (Java Database Connectivity – Sun)

Проблемы, связанные с хранением в РСУБД

- Долговременное хранение в РСУБД
 - Осложняется несовпадением импеданса (impedance mismatch)
 - Просто реализуемо за счет широкого распространения и доступности РСУБД
- Долговременное хранение в ОСУБД
 - Упрощено за счет концептуальной схожести
 - Объектных моделей
 - Связываний языка программирования

Объектные СУБД

- Стандартизацию в области ОСУБД проводит Object Database Management Group – ODMG
- Организована в 1991 г. разработчиками ОСУБД (GemStone, O2, ObjectStore, Objectivity, Poet, Versant)
- В качестве исходной платформы для стандарта на ряду с прочим был использован стандарт CORBA 1.0 для обеспечения:
 - Переносимости приложений между ОСУБД
 - Интероперабельности с архитектурой CORBA

Стандарт ODMG

- ODMG опубликовала 3 основных версии стандарта:
- 1.0 в 1993 – базовый стандарт, связывание с C++ и Smalltalk
- 2.0 в 1997 – дополнено связыванием с Java, изменения в объектной модели, расширения стандарта, определение репозитория метаданных
- 3.0 в 2000 – существенное расширение связывания с языками (организация постоянного хранения)

Стандарт ODMG

- Объектная модель является расширением объектной модели OMG/OMA
- Язык определения схемы – ODL (Object Definition Language) является расширением OMG/IDL
- Связывание с программными языками
 - C++
 - Java
 - Smalltalk
- Язык объектных запросов (OQL)
- Поддержка долговременного хранения для указанных языков программирования
- ODMG участвовал в разработке стандарта PSS

Пример: контекст именования в ODL

```
typedef NameComponent struct {
    string id;
    string kind;
};

typedef struct ListElement {
    NameComponent name;
    Object obj;
};

interface ODLPortableNctxState {
    list<ListElement> m_list;
    extent ODLPortableNctxStateHome;
    Object resolve(in NameComponent n);
    void bind(in NameComponent n, in Object obj);
    void unbind(in NameComponent n, in Object obj);
    void rebind(in NameComponent n, in Object obj);
};
```


Отображение в ОСУБД

- ОСУБД устраняют проблему «несовпадения импеданса» за счет использования объектной модели
- Поддерживают хранение композитных объектов уменьшая время выборки объекта (нет необходимости его «сборки»)
- Поддерживают ассоциативные связи, что позволило использовать более быстрые методы навигации по таким связям
- Поддерживают связи иерархии, упрощающие определение классов/интерфейсов

Сравнение различных методов хранения: ФС, РСУБД, ОСУБД

- Долговременное хранение в файлах
 - Нет затрат на лицензирование
 - Экстернализация не является прозрачной для реализаций объектов
 - Отсутствует контроль за одновременным доступом
 - Не обеспечивает отказоустойчивость
- Долговременное хранение в РСУБД
 - Просто реализуемо за счет широкого распространения и доступности РСУБД
 - Осложняется «несовпадением импеданса»
- Долговременное хранение в ОСУБД
 - Упрощено за счет концептуальной схожести объектных моделей
 - Связываний языка программирования

Основные моменты

- Долговременное хранение реализуется для серверных объектов с внутренним состоянием
- Долговременное хранение прозрачно для разработчиков и проектировщиков клиентских объектов
- Реализация долговременного хранения м.б. реализована с помощью Persistent State Service
- PDL используется для определения структур данных для долговременного хранения

Основные моменты

- Долговременное хранение м.б. реализовано с помощью
 - Файловой системы
 - РСУБД
 - ОСУБД
- Компилятор PDL автоматически генерирует реализацию долговременного хранения
 - Генерирует схему РБД
 - Генерирует операции доступа для сохранения или загрузки состояния

Литература / Internet ИСТОЧНИКИ

- В. Эммерих *Конструирование распределенных объектов*. - М.:Мир. - 2002.
- Vinoksi S. ***CORBA: Integrating Diverse Applications Within Distributed Heterogeneous Environments***, IEEE'96.
- Schmidt D.C. And Vinoski S. ***Object Adapters: Concepts and Technology***, SIGS C++ Report, Vol. 9, No. 11, Nov-Dec 1997.
- Ю.А. Григорьев, А.Д. Плутенко. ***Жизненный цикл проектирования распределенных баз данных***. - Благовещенск. - 1999.
- www.omg.org