Our Experiences and Analysis of the POOL Storage Manager for Relational Back-ends

By:

S.S.Tomar
Computer Center
CAT, Indore, India

On Behalf of
CAT-POOL TEAM

Topics Of Discussion

- Introduction
- The POOL Storage Manager
- The two “proof-of-concept” tests
- RelationalStorageSvc prototype details
- Storage Manager – Our Feedback from the implementation for Relational back ends
- Summary.

Thank You

Introduction

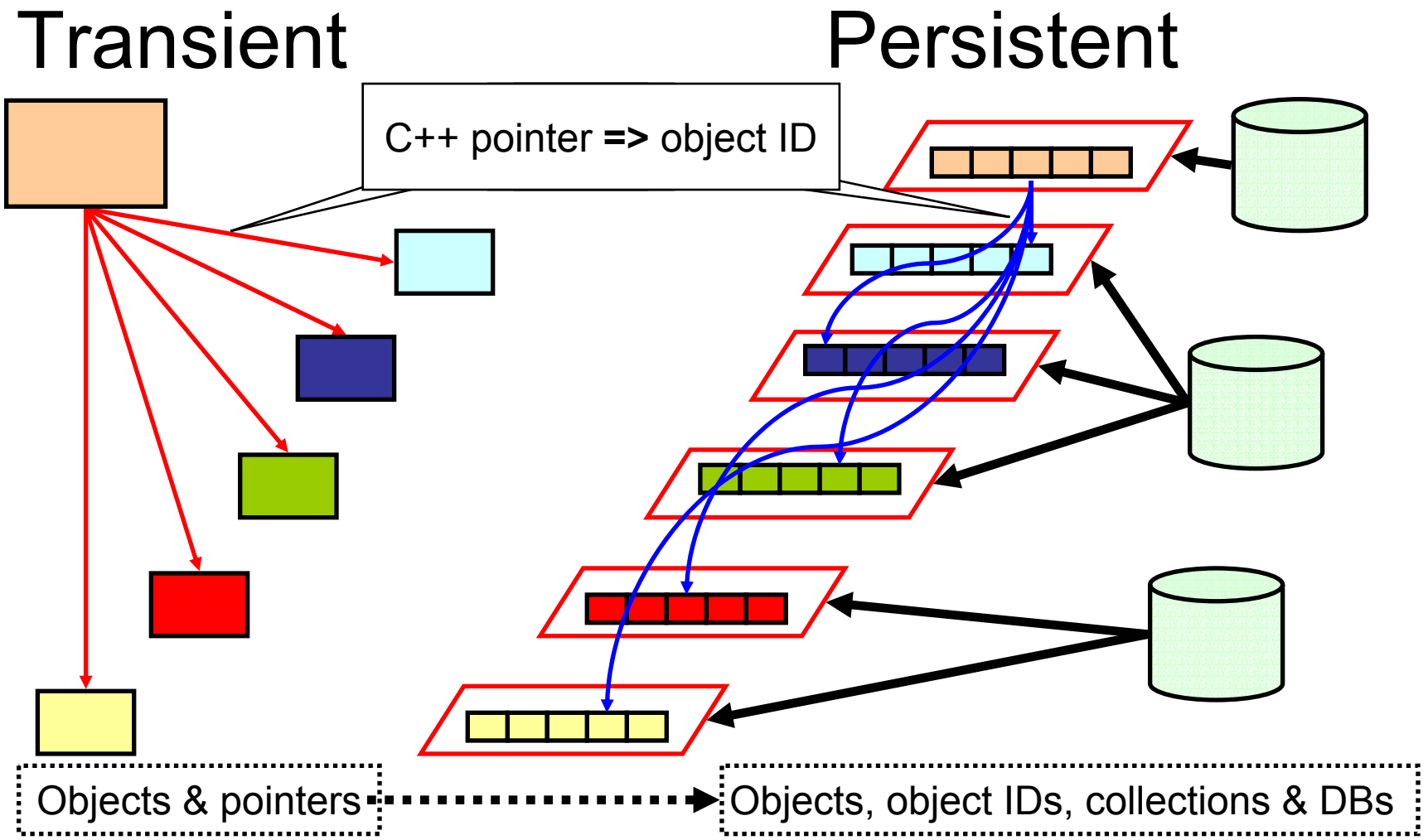
- POOL project.
  - Objective -Persistency of Objects for the LHC.
  - The Basic Concept.
- POOL, File Catalog.
  - Provides unique ID to a Database/File in POOL.
  - PFN, FID & LFN
- POOL, Storage Manager
  - Provides Interfaces for diverse Storage Technologies.

(Cont…)

- Storage option in current version of the POOL.
  - ROOT (by using RootStorageSvc component)
- RelationalStorageSvc prototype
  - Storage of Objects in Relational Backends.
  - Proof-of-concept for the POOL Storage Manager.
  - CAT, India's contribution to the POOL project.

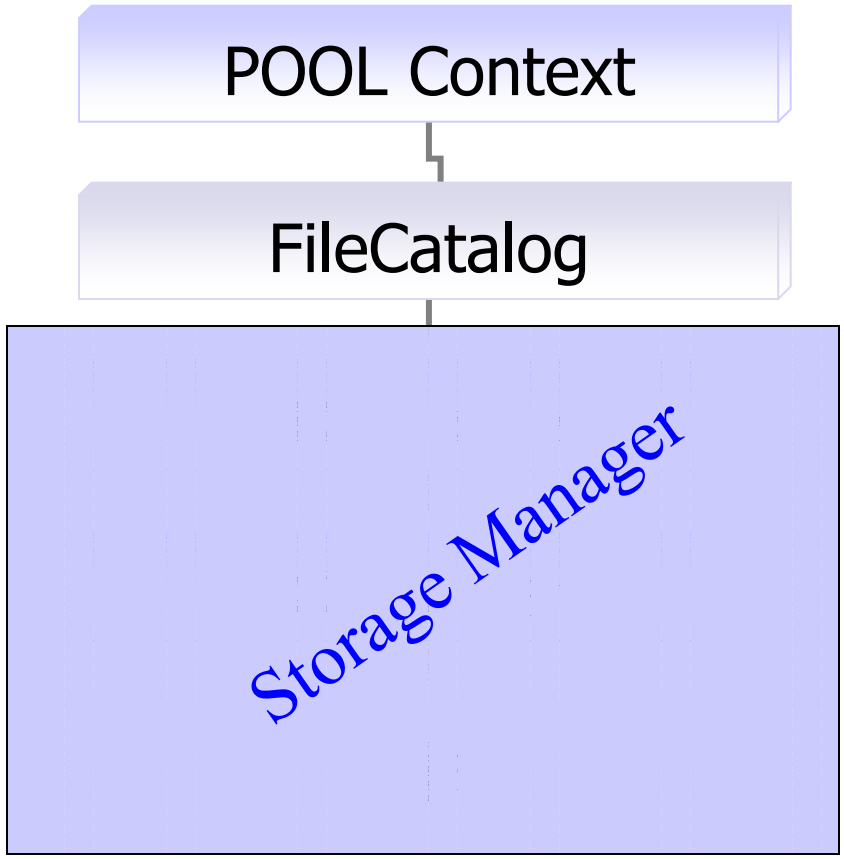
Back…

Persistency – What is it?



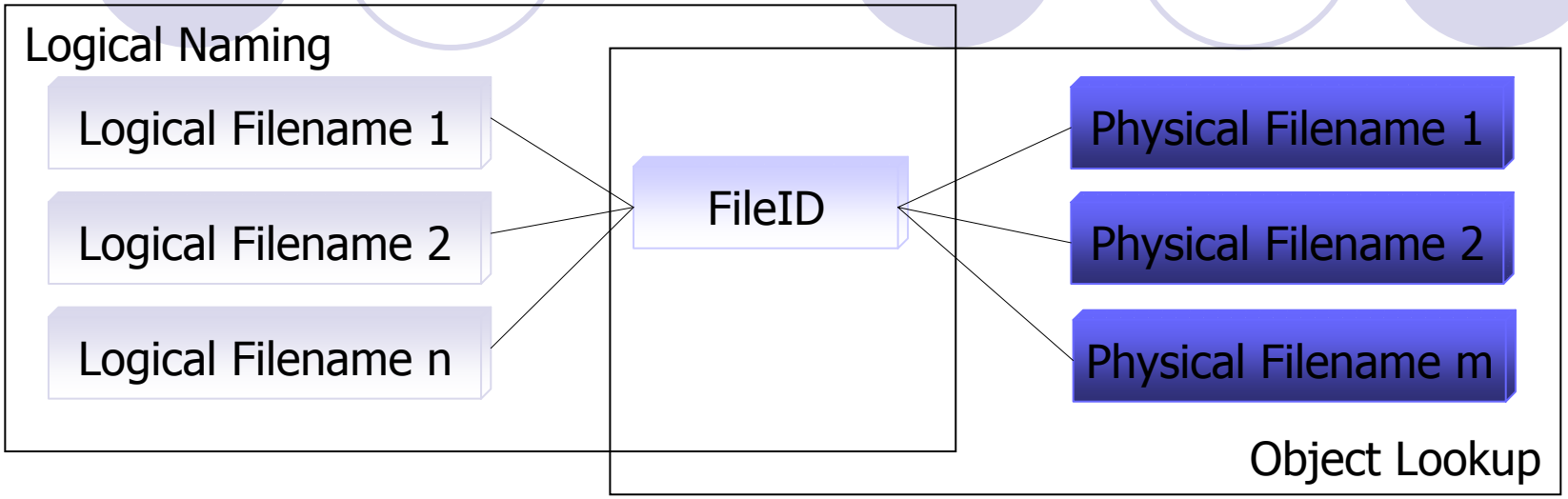
The POOL – Basic Concept

- A application may access databases (eg ROOT files) from a set of catalogs
- Each database has containers of one specific technology (eg ROOT trees)
- Smart Pointers are used
  - to transparently load objects into a client side cache
  - define object associations across file or technology boundaries



Back…

POOL File Catalog



- POOL uses GUID implementation for FileID
  - unique and immutable identifier for a file (generated at create time)
  - allows to produce sets of file with internal references without requiring a central ID allocation service
  - catalog fragments created independently can later be merged without modification to data files.
- Object lookup is based only on right side box!
  - Logical filenames are supported but not required

Back…

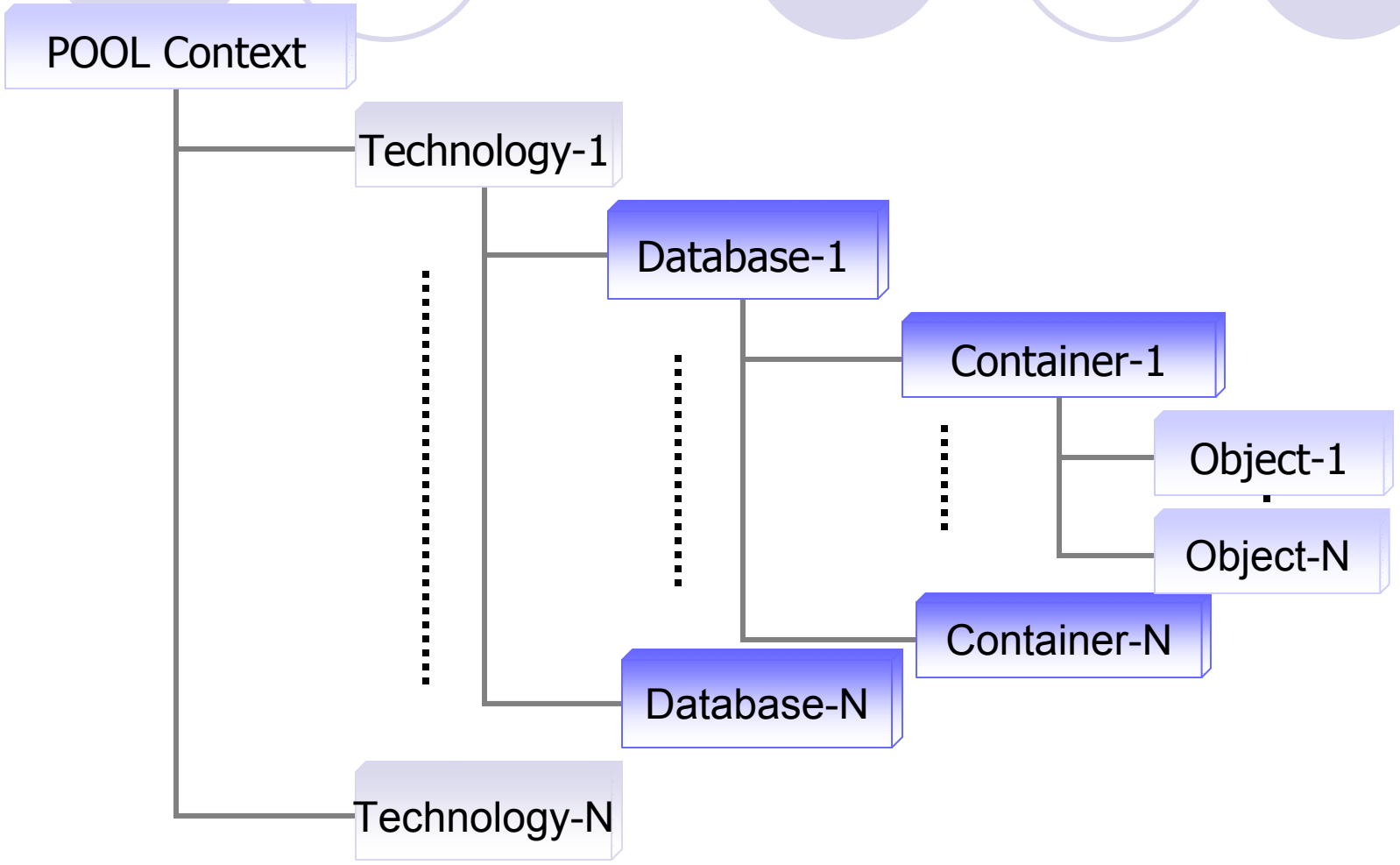
POOL Storage Manager

- Objectives/Goals.
  - Diverse storage Tech. support (Root, RDBMS,…).
  - Manage on-demand use of storage technologies.
  - Rapid adoption of new storage technologies.

- Conceptual Architecture.
  - POOL Storage Hierarchy.
  - Internal Data Structures in a database/file.
  - Token – The persistent object ID.

Back…

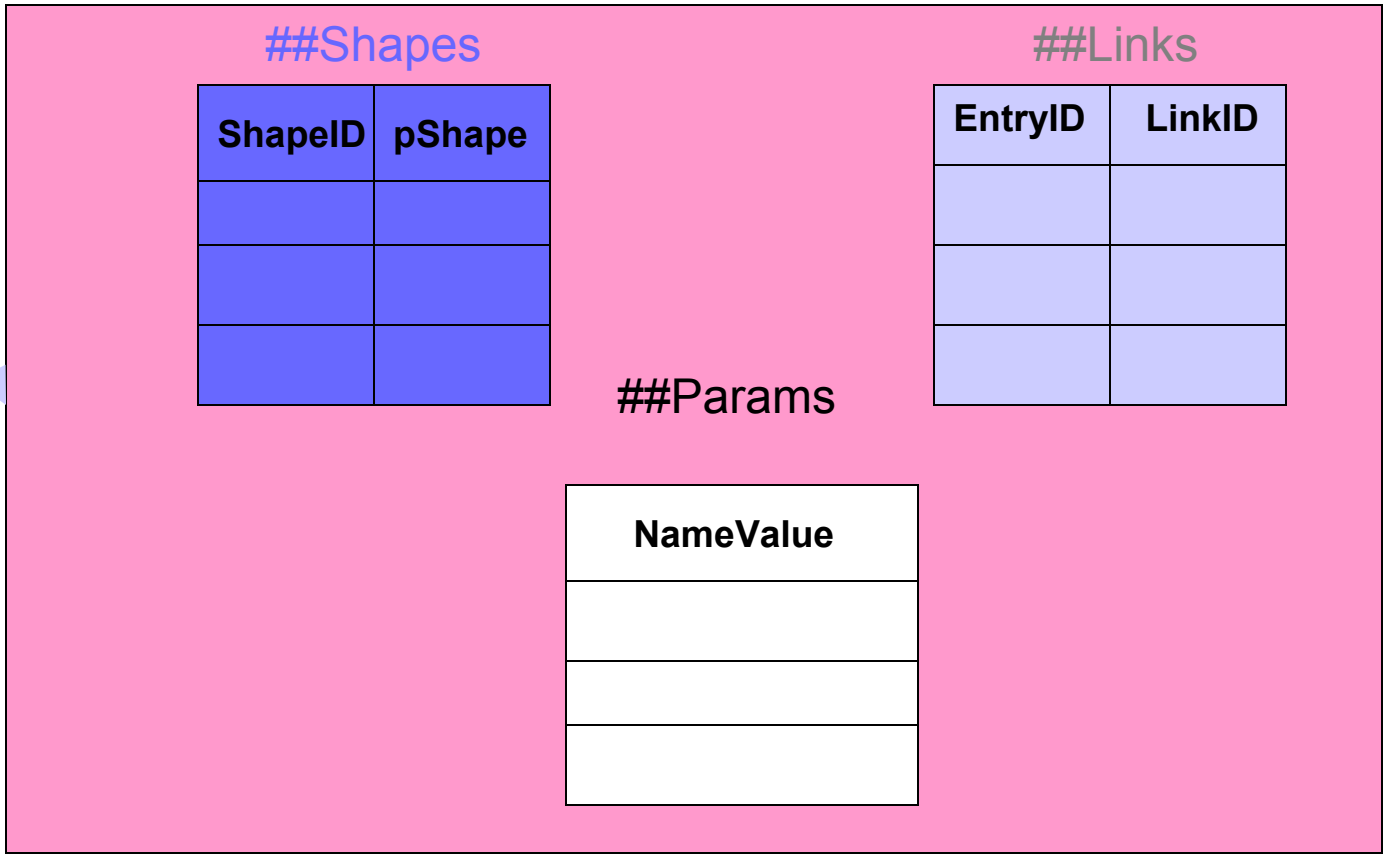
POOL Storage Hierarchy



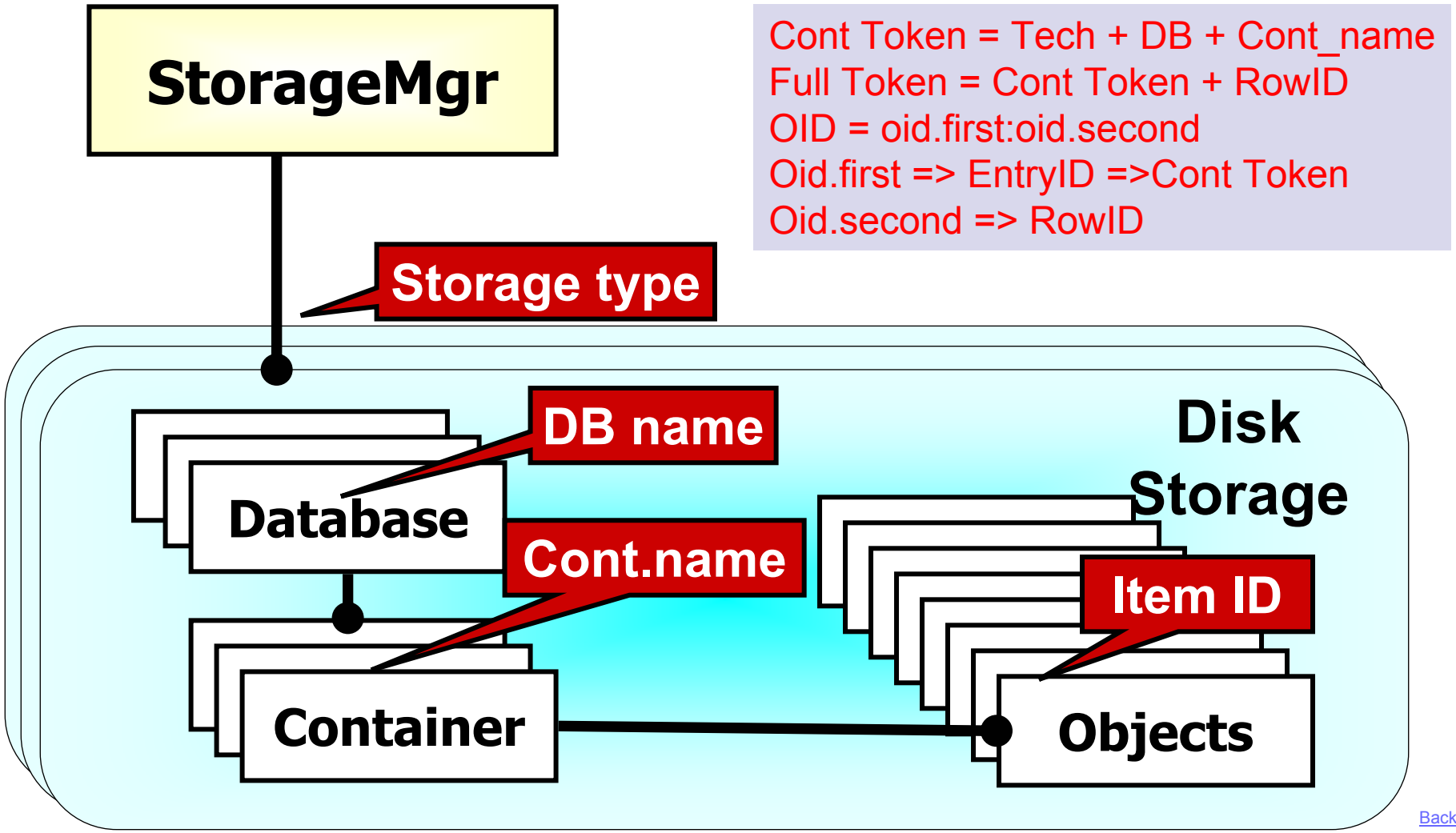
Back…

Internal Data Structures in a Database

DataBase Context



Token – The Persistent Object ID



Cont Token = Tech + DB + Cont_name
Full Token = Cont Token + RowID
OID = oid.first:oid.second
Oid.first => EntryID =>Cont Token
Oid.second => RowID

Back…

The Two “Proof-of-concept” tests

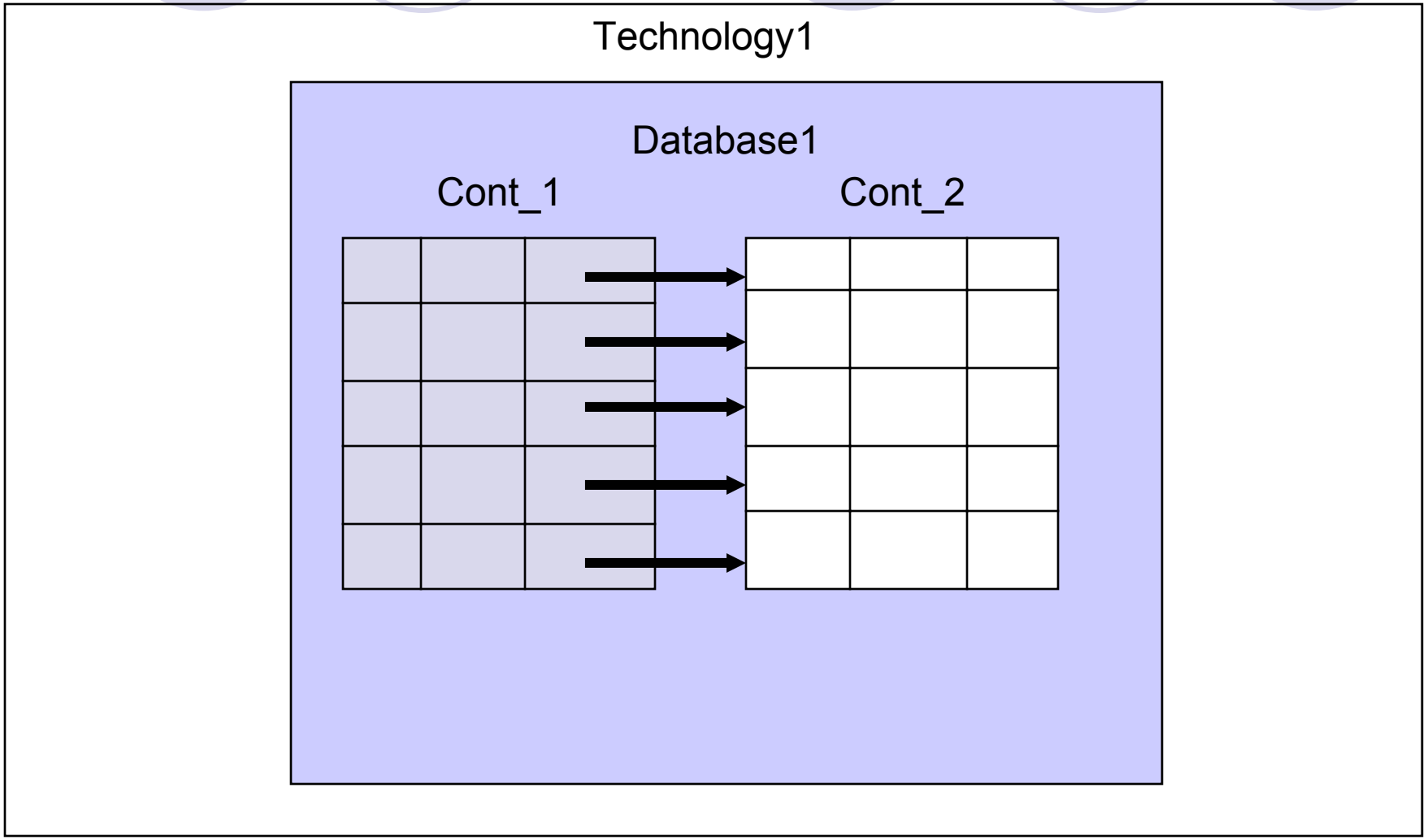
- Test1:
  - Navigation, storage and retrieval of objects with primitive data members.
- Test2:
  - Navigation, storage and retrieval of objects with referenced data members.

(Cont…)

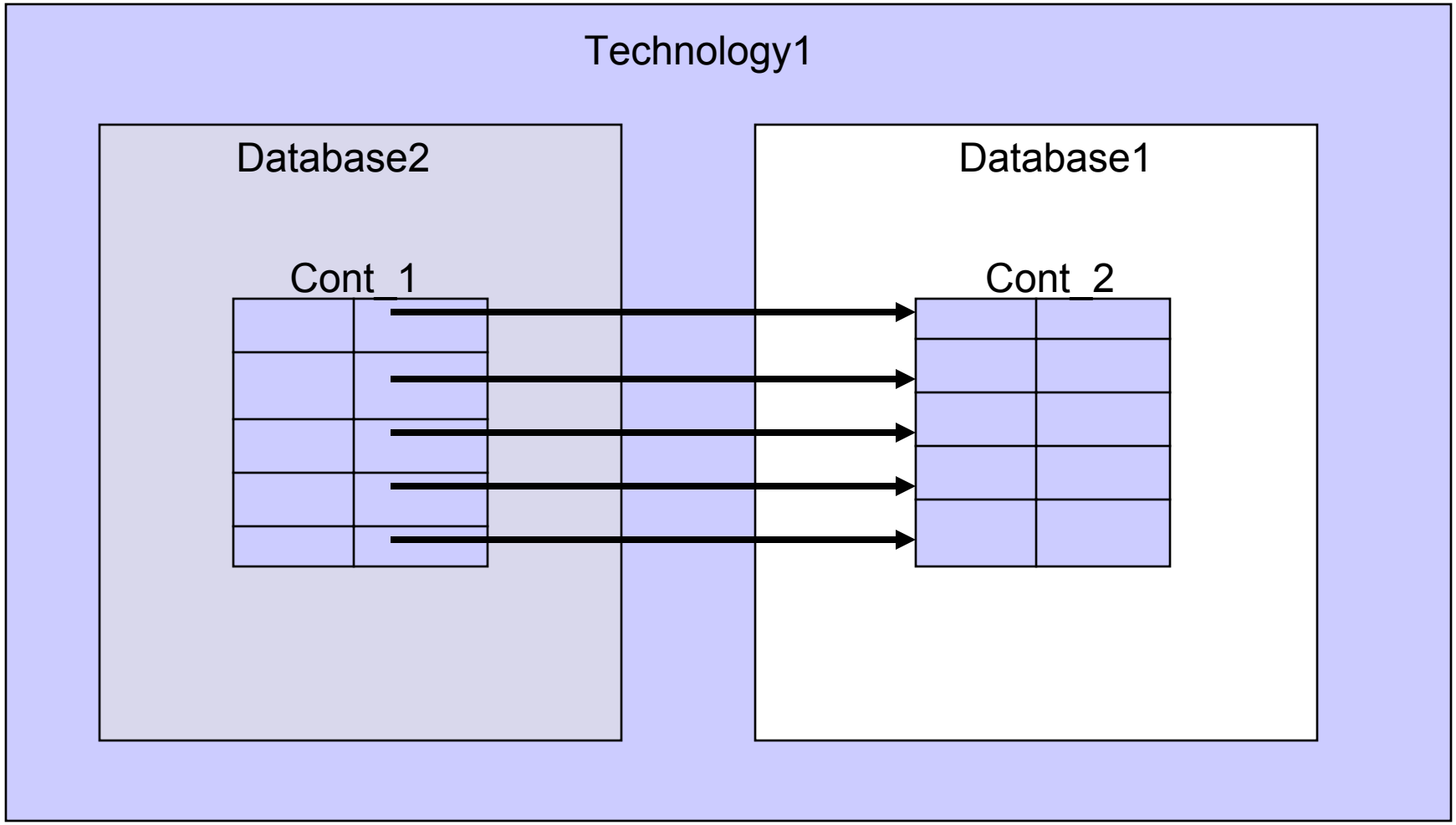
- Test2 : Various Cases:
  - Case1: Same Technology Same Database Reference Test: Objects in one Technology referencing objects in same technology and same database.
  - Case2: Same Technology Different Database Reference Test: Objects in one Technology referencing objects in same technology but in different database.
  - Case3: Cross Technology Reference Test: Objects in one Technology (RDBMS) referencing objects in Different Technology (ROOT).

Back…

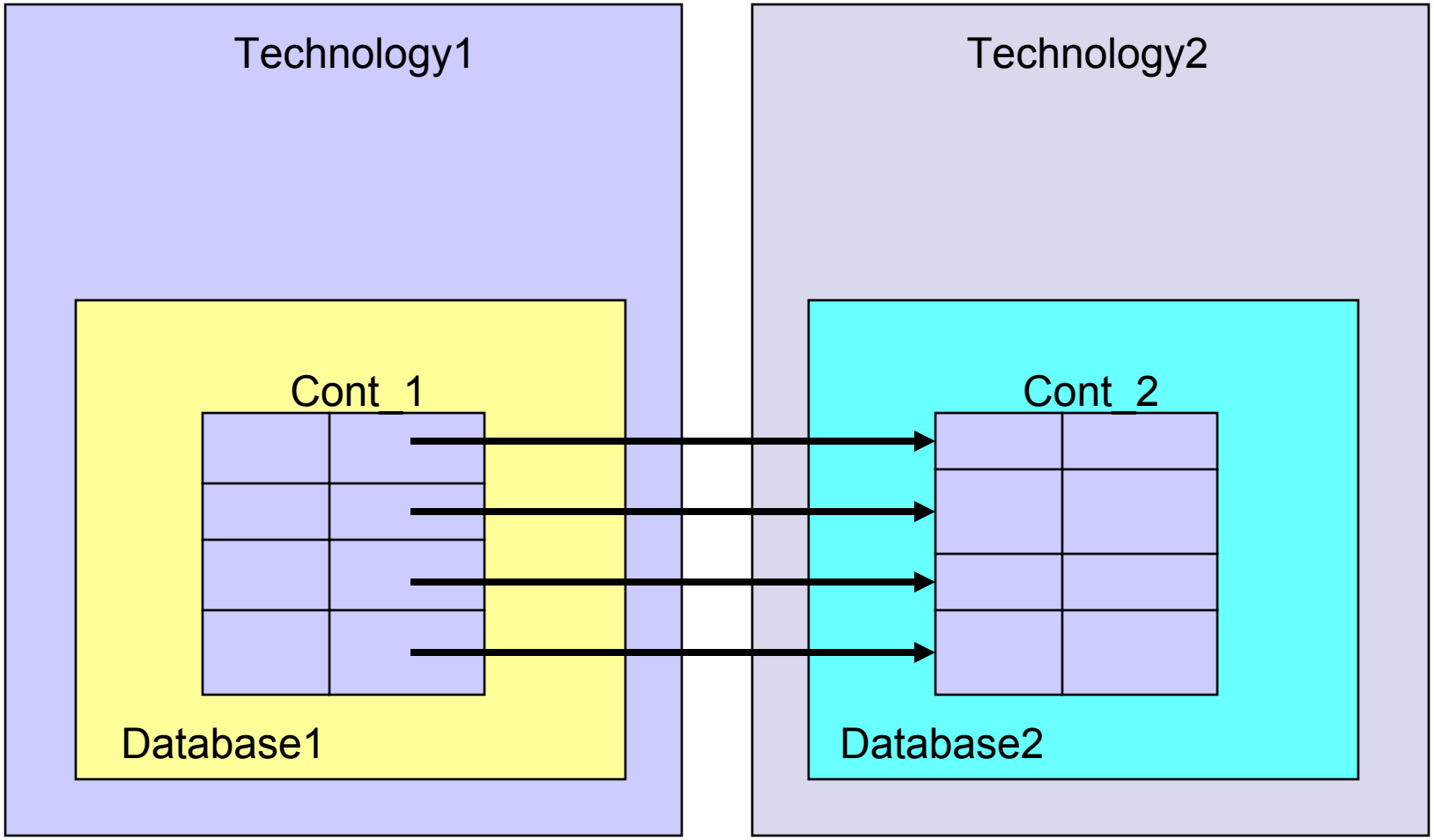
Case1: Same Technology Same Database Reference Test



Case2: Same Technology different Database Reference test



Case3: Cross Technology Reference Test



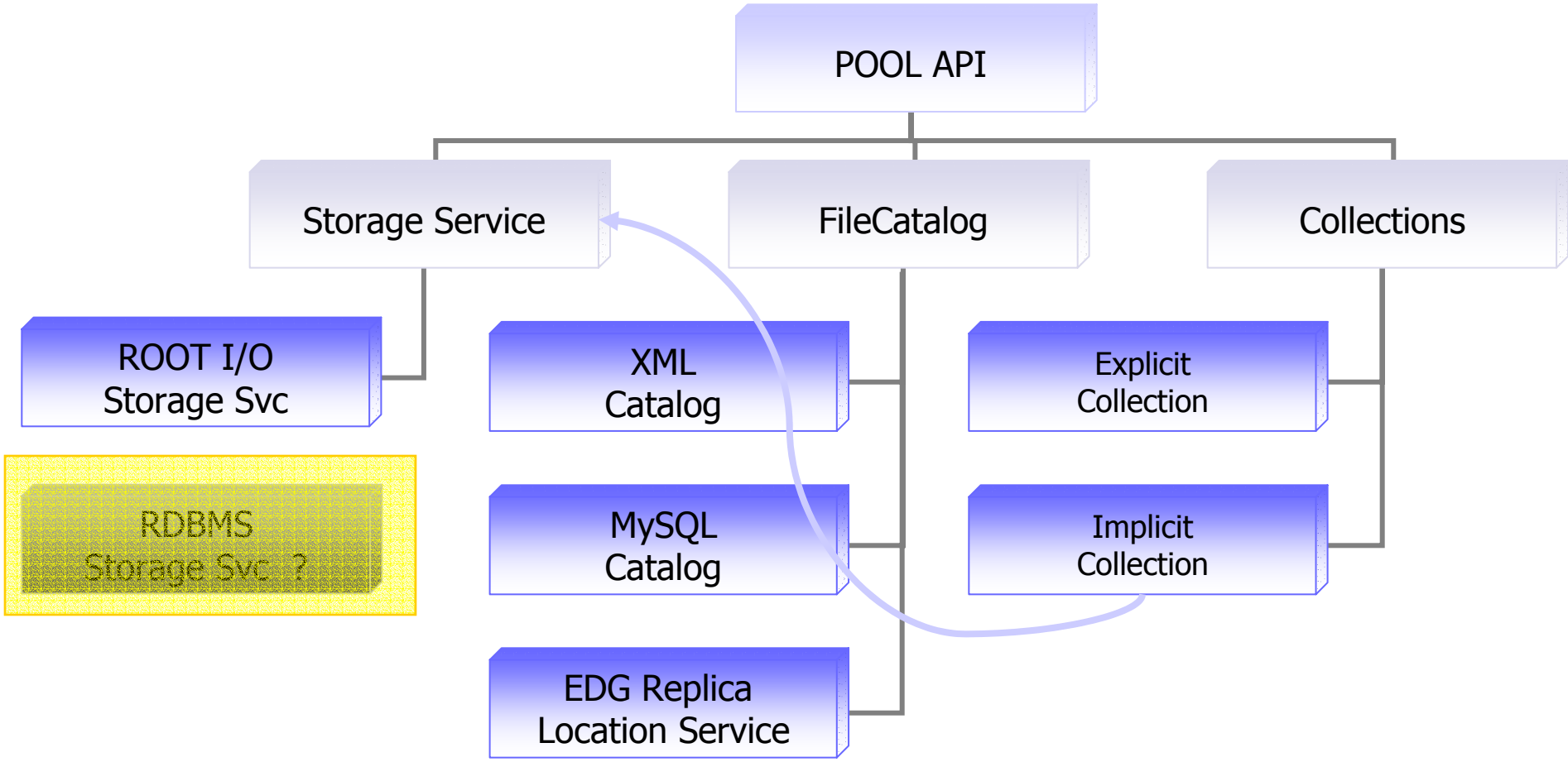
Back…

RelationalStorageSvc prototype

- RSSvc Introduction.
- Objectives/Goals.
- Design Considerations.
- The Design.
- Token Dereferencing.
- Writing/Reading Illustration.

Back…

RSSvc's place in POOL Architecture



RSSvc Introduction

- RelationalStorageSvc has been implemented using a design similar to RootStorageSvc component.
- A plug-in in POOL framework - for persistency of c++ objects in RDBMS.
- RDBMS vendor independent (oracle/mysql/sqlserver/…)

(Cont…)

- Uses ODBC connectivity option.
- Implements POOL StorageSvc interfaces and uses ODBC APIs.
- Prototype development using
  - RedHat Linux 7.3 as OS.
  - Oracle 9i(9.2.0) as RDBMS.
  - Oracle wire protocol driver from DataDirect.
  - POOL ver 1.2 to 1.6.1 for testing.

Back…

RSSvc Design Objectives/Goals

- Store/Retrieve/Navigate simple C++ objects in RDBMS.
- Achieve RDBMS vendor independency (as much as possible).
- Use SEAL dictionary and plugin services
  - for converting transient object to persistent object representation in RDBMS.
  - for implementing the component as a plug-in.

Back…

Design Considerations

- POOL Database = User schema
  Every user has a different context in RDBMS.
- Physical reference to a database to be made using the DSN (data source name) concept of Relational technology.
- A Class is mapped to a table in database.
- A data member of a class is mapped to a column in the database table.
- Member classifiers/qualifiers to be stored along with persistent shape of the class.

Back…

The Design

- File Catalog related Physical file name mapping.
- Storage Manager related Data Structures mapping.
- Primitive data type mapping.
- Reference data type mapping.

Back…

Physical file name mapping

- PFN(in file catalog) = DSN:Username:Password

  E.g PFN = OracleWP2:tomar:catpool

  refers to database on host oradev9.cern.ch with SID=D9,running on port number 1521 accessible to user tomar using password catpool.For establishing connectivity with the database it uses the odbc driver located at /afs/cern.ch/sw/lcg/app/pool/drivers/odbc/lib/ivora19.so

Typical DSN entry in ODBC.ini

```
[OracleWP2]

Driver=//afs/cern.ch/sw/lcg/app/pool/drivers/odbc/lib/ivora19.so
Description=DataDirect 4.20 Oracle Wire Protocol
Hostname=oradev9.cern.ch
LogonID=tomar
Password=******
PortNumber=1521
SID=D9
```

Typical FileCatalog entries

```
<File ID="B2B6AA42-E882-D811-8CDF-00D0B7B86B36">
<physical>
<pfn filetype="Oracle" name="OracleWP2:tomar:catpool"/>
</physical>
<logical>
</logical>
</File>

<File ID="B2992387-E882-D811-81CF-00D0B7B86B36">
<physical>
<pfn filetype="Oracle" name="OracleWP3:tomar2:tomar2"/>
</physical>
<logical/>
</File>
```

Back…

Storage Manager related Data Structures mapping

- ##Shapes(ShapeID, pShape)
  = Table Shapes##
  With columns ShapeID and pShape.

- ##Links(LinkID)
  = Table Links##
  With column LinkID.

- ##Params(NameValue)
  =Table Params##
  With column NameValue

Back…

Primitive data type mapping

- Persistent shape of a class with primitive datatype members is stored as

  datatype@member-classifier

  e.g : int@P;float@O;char@R

  where P-public,R-private,O-protected

- Primitive data types are mapped to RDBMS datatypes as.

Back…

long double	**number**
double	**number**
float	**float**
long	**number**
char	**varchar**
unsigned char	**varchar**
signed char	**varchar**
int	**number**
long int	**number**
short int	**number**
unsigned int	**number**
signed int	**number**
unsigned long int	**number**

signed long int	**number**
signed short int	**number**
unsigned short int	**number**
ulonglong	**number**
unsigned long long int	**number**
longlong	**number**
long long int	**number**
signed long long int	**number**

Back...

Reference Data Type Mapping

- Persistent shape of a class with reference data type member is stored as
  - Datatype@member-classifier
  - Eg.: Reference@P
- The table column stores the string oid.first : oid.second (oid.first indicates the entryID – in local link table- corresponding to the ref's container Token entry, and oid.second indicates the rowID inside the actual ref container) for the referenced member.
- The link table - in local db - contains one column (Link id). Link Id stores the container Token for the referenced object.

Back…

Token Dereferencing

- Full Token = Container Token + RowID
- Container Token = TechID + Database + Container
  = TechID + PFN + Cont_Name
  = “00000900” + DSN:UserID:Passwd + Cont_Name
- DSN = IP, PORT, SID, Driver
- Hence:
  - Full Token = “00000900” +DSN:UserID:Passwd + Cont_Name + RowID
  - Thus any object stored in RDBMS can be located.

Back…

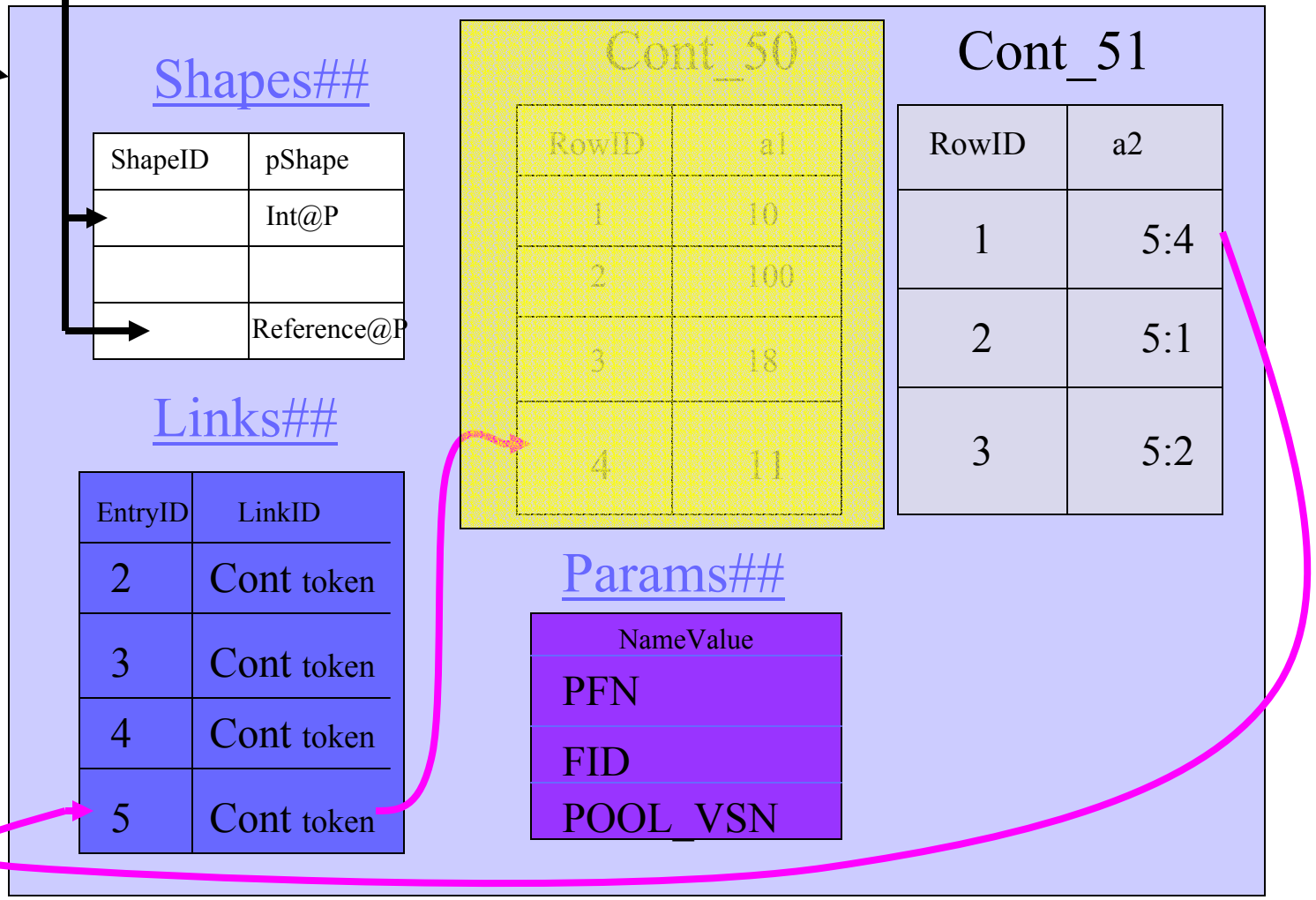
Illustration: Writing/Reading

Cont token=DB+Tech+ContName
DB=RemoteDB, Tech=Relational, Cont=Cont_50
Full token=Rel. + ReDB + Cont_50 + 4(RowID)

Database Context
DSN=OracleWP2

User=tomar
Passwd=**
IP=oradev9
Port=1521
SID=D9

ShapeID for Classes



Back…

Typical Table entries

- Shapes## (on local DB)

Back…

SHAPEID

{ID=4FFF4DBB-FFE4-4F0D-95D7-204737331AD8} {CL=TestClass2} {NCOL=1}
{CNT=0} {COL= {{NAME=TestClass2} {CLASS=TestClass2}
{TYP=21}{OPT=0}{OFF=0}{SIZ=0}{CNT=1}}}

{ID=4FFF4DBB-FFE4-4F0D-95D7
204737331AD7}{CL=TestClass1}{NCOL=1}{CNT=0}
{COL={{NAME=TestClass1}{CLASS=TestClass1}
{TYP=21}{OPT=0}{OFF=0}{SIZ=0}{CNT=1}}}

PSHAPE

---

int@P;int@P;string@P

int@P;int@P;string@P;Reference@P

(Cont…)

Shapes## (on Remote DB)

Back…

SHAPEID

{ID=4FFF4DBB-FFE4-4F0D-95D7-204737331AD8} {CL=TestClass2} {NCOL=1}
{CNT=0} {COL= {{NAME=TestClass2} {CLASS=TestClass2}
{TYP=21}{OPT=0}{OFF=0}{SIZ=0}{CNT=1}}}

PSHAPE

---

int@P;int@P;string@P

int@P;int@P;string@P;Reference@P

(Cont…)

- Links## (on local DB)

Back…

LINKID

[DB=<localDB>][CNT=##Params][CLID=DA8F479C-09BC-49D4-94BC-99D025A23A3B][TECH=00000900][OID=00000002-FFFFFFFF]

[DB=<localDB>][CNT=cont_50][CLID=4FFF4DBB-FFE2-4F0D-95D7-204737331AD8][TECH=00000900][OID=00000003-FFFFFFFF]

[DB=<localDB>][CNT=cont_51][CLID=4FFF4DBB-FFE2-4F0D-95D7-204737331AD7][TECH=00000900][OID=00000004-FFFFFFFF]

[DB=B2992387-E882-D811-81CF-00D0B7B86B36] [CNT=cont_50] [CLID=4FFF4DBB-FFE2-4F0D-95D7-204737331AD8] [TECH=00000900][OID=00000005-00000000]

(Cont…)

- Links## (on remote DB)

Back…

LINKID

[DB=<localDB>][CNT=##Params][CLID=DA8F479C-09BC-49D4-94BC-99D025A23A3B] [TECH=00000900] [OID=00000002-FFFFFFFF]

[DB=<localDB>][CNT=cont_50][CLID=4FFF4DBB-FFE2-4F0D-95D7-204737331AD8] [TECH=00000900] [OID=00000003-FFFFFFFF]

(Cont…)

- Params## (on local DB)

Back…

NAMEVALUE

[NAME=FID][VALUE=B2B6AA42-E882-D811-8CDF-00D0B7B86B36]

[NAME=PFN][VALUE=OracleWP2:tomar:catpool]

[NAME=POOL_VSN][VALUE=1.1]

(Cont…)

- Params## (on remote DB)

Back…

NAMEVALUE

[NAME=FID][VALUE=B2992387-E882-D811-81CF-00D0B7B86B36]

[NAME=PFN][VALUE=OracleWP3:tomar2:tomar2]

[NAME=POOL_VSN][VALUE=1.1]

Storage Manager – Our feedback from implementation for Relational backend

- Either change names of ##Links, ##Shapes,##Params or provide a function for checking supported grammar for table names, especially for ##Links, ##Shapes and ##Params and then changing names.

- Provide function for querying access rights of the tables.

- Provide function for querying transaction capabilities and then force upper layer functions to make use of the database transaction features.

Back…

Summary

- POOL Storage Manager passed all “proof-of-concept” tests for Relational backends.
- We have some feedback on the relational backend specific interface changes in the storage manager, which we will be submitting as a document.

Back…

Thank You

- Questions ???