



# Sedna: Advanced Databases

---

Group 20

Ankita B. | Ankit S. | Terry P. | Siddharth G.



# History

---



# Examples Of XML Documents

```
<invoice>
  <number>421</number>
  <date>2008-05-24</date>
  <items>
    <item>
      <description>Wool Paddock Shet Ret Double Bound Yellow 4'0"</description>
      <quantity>1</quantity>
      <unitPrice>105.00</unitPrice>
    </item>
    <item>
      <description>Wool Race Roller and Breastplate Red Double</description>
      <quantity>1</quantity>
      <unitPrice>75.00</unitPrice>
    </item>
    <item>
      <description>Paddock Jacket Red Size Medium Inc Embroidery</description>
      <quantity>2</quantity>
      <unitPrice>67.50</unitPrice>
    </item>
  </items>
</invoice>
```

# Why Sedna?

---

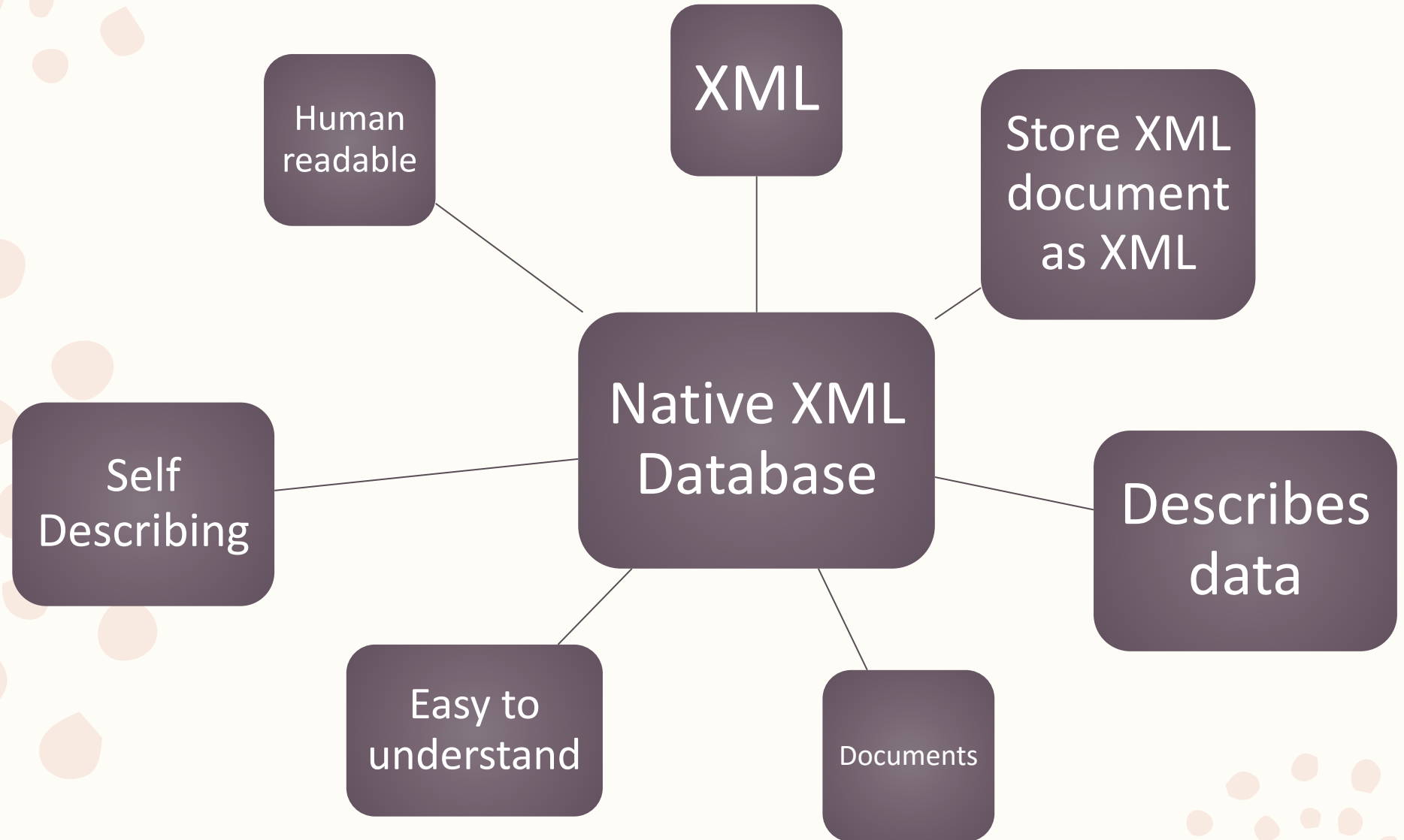
*Before*

- **Non native** strategies - mapping an XML data model onto relational or object-oriented model

*After*

- **Native XML** – XML files are fundamental unit of storage

# Native XML Database



# Native XML Databases



Native XML Database System

ORACLE®

BERKELEY DB



# XML Model vs. Relational Model

Computer Table

<i>Id</i>	<i>Speed</i>	<i>RAM</i>	<i>HD</i>
101	800Mhz	256MB	40GB
102	933Mhz	512MB	40GB

```
<Table>
  <Computer Id='101'>
    <Speed>800Mhz</Speed>
    <RAM>256MB</RAM>
    <HD>40GB</HD>
  </Computer>
  <Computer Id='102'>
    <Speed>933Mhz</Speed>
    <RAM>512MB</RAM>
    <HD>40GB</HD>
  </Computer>
</Table>
```

# Challenges Sedna Addresses



---

Improved schema-based clustering  
storage strategy

Novel memory management  
technique

Not having to evaluate the special join  
operation



# Sedna Features

---

- Open Source
- Native XML Database
- Based on the XQuery language and the XQuery/XPath data model
- XUpdate language
- SQL connection from XQuery
- Full-text search indices
- Support for ACID transactions
- Support for fine-grained XML triggers

# Sedna Features

---

- Built-in Database Connection Pooling manager.
- Persistent storage
- Incremental hot backup
- Supports Unicode
- Zero dependencies
- Supports Binary BLOBS as well as Java Object storage.
- Database security

# Why Move From Relational to Sedna

- Flexibility
- Scalability and Interoperability
- Performance increase
- No capacity limits for data type
- Searches: Structural and Semantical



# Sedna Architecture

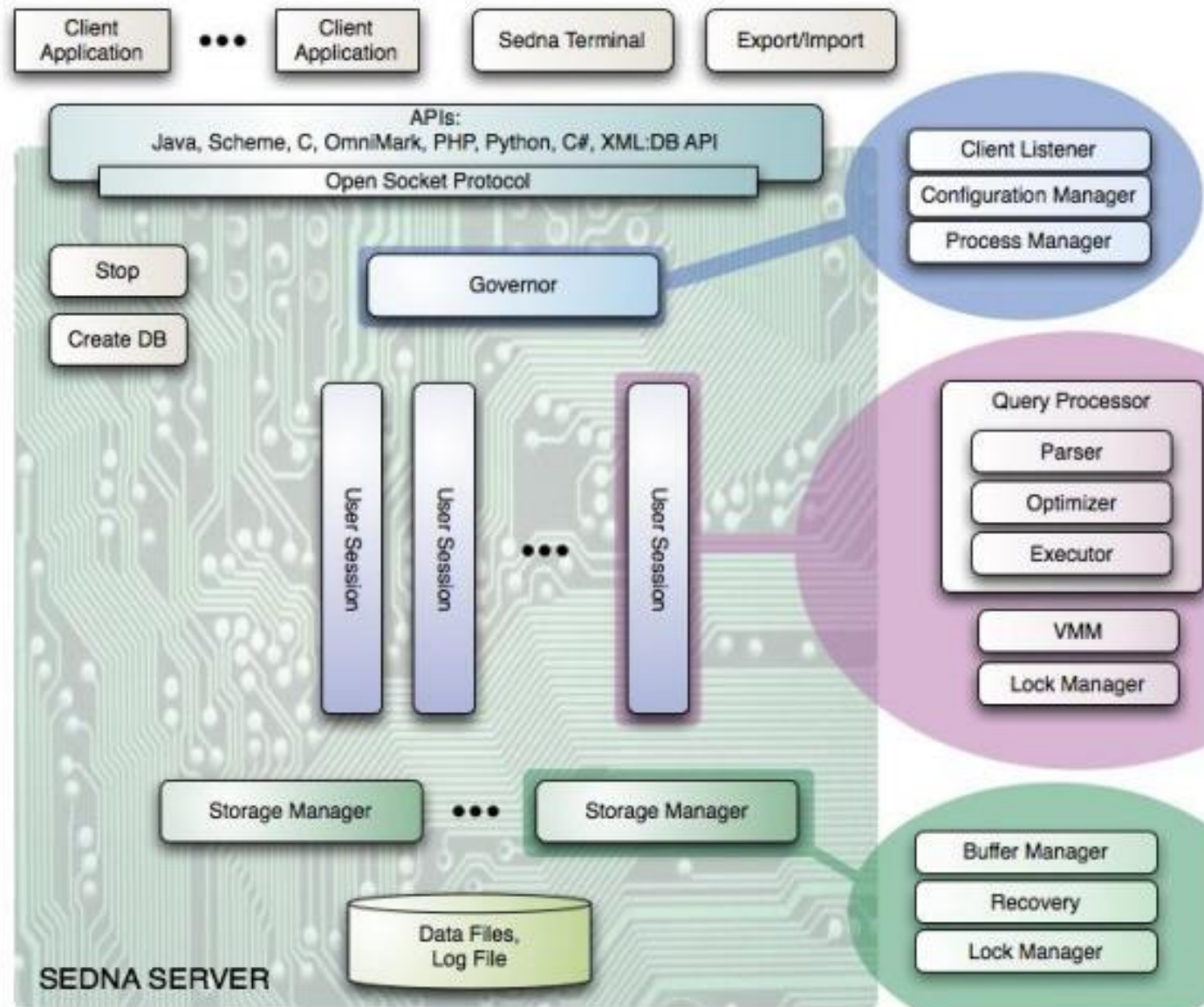
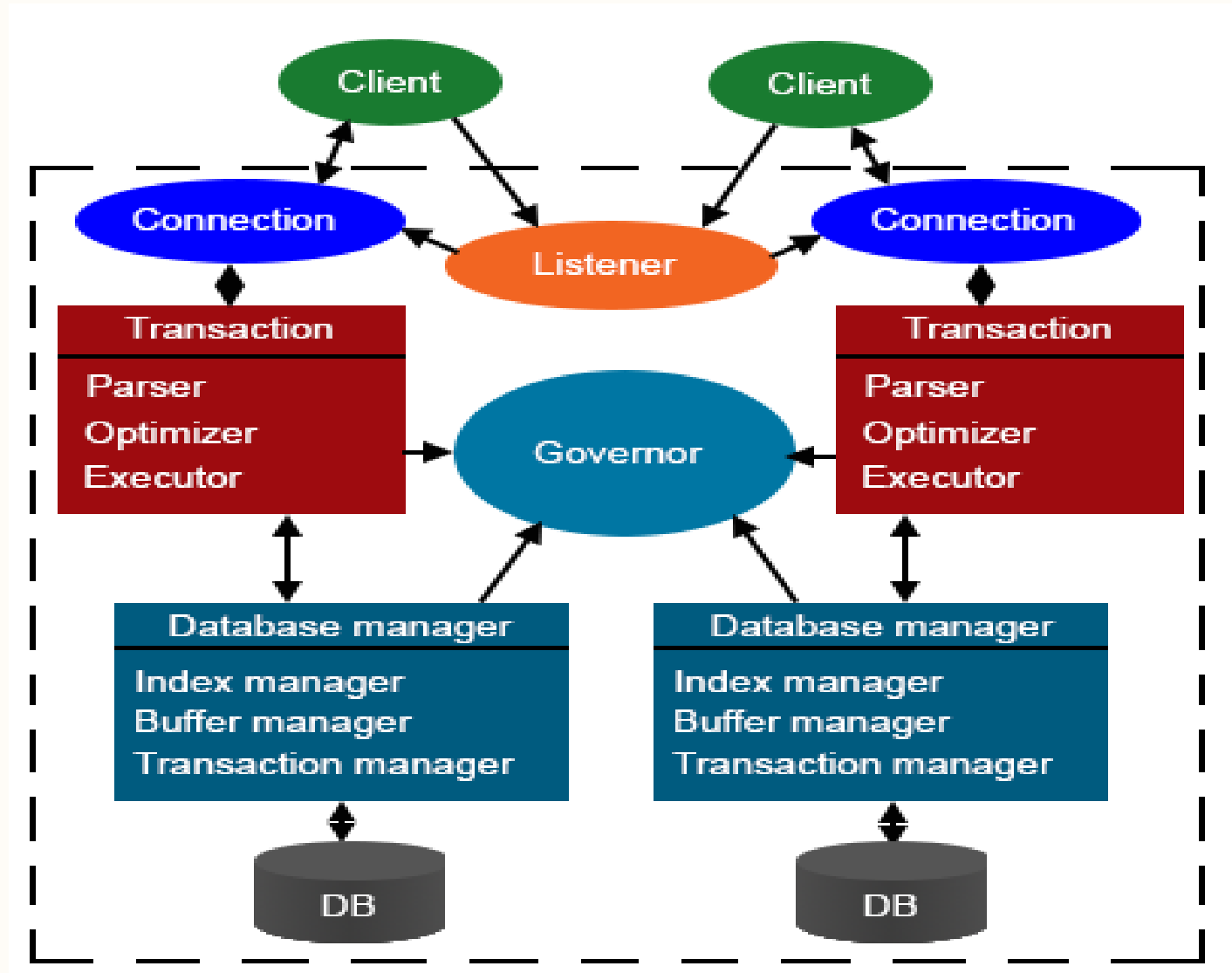


Figure 1: SEDNA Architecture Overview

# Sedna Architecture



# Sedna Native XML Database Client/Server Protocol

---

Message-based protocol for communication with clients through the TCP/IP sockets.

Message Structure – First 4 – Instruction; Next 4 –Length; 'Length ' Bytes -Body

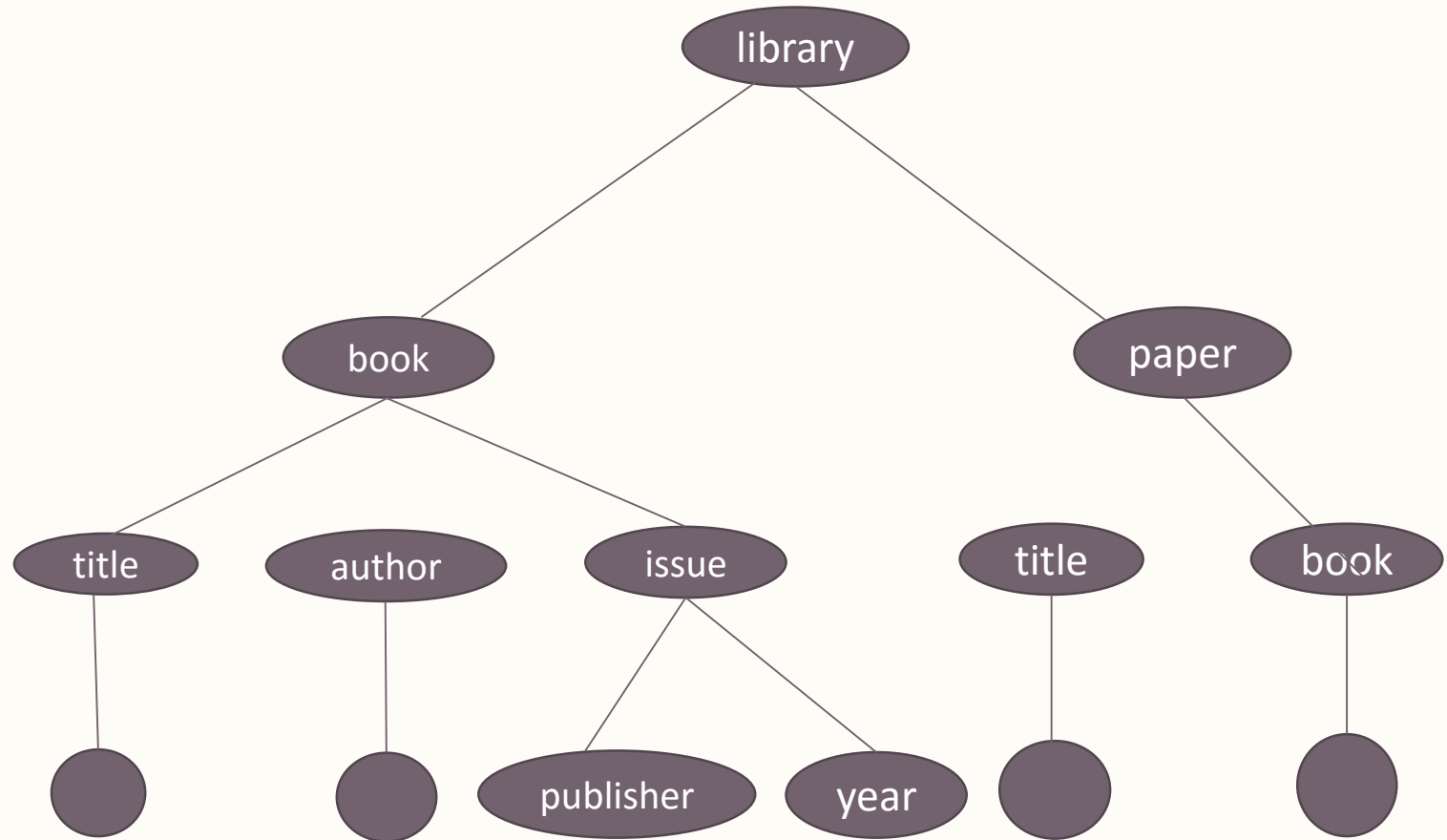
To begin a session – start up message

3 types of queries – **query** , **update** , **bulk**

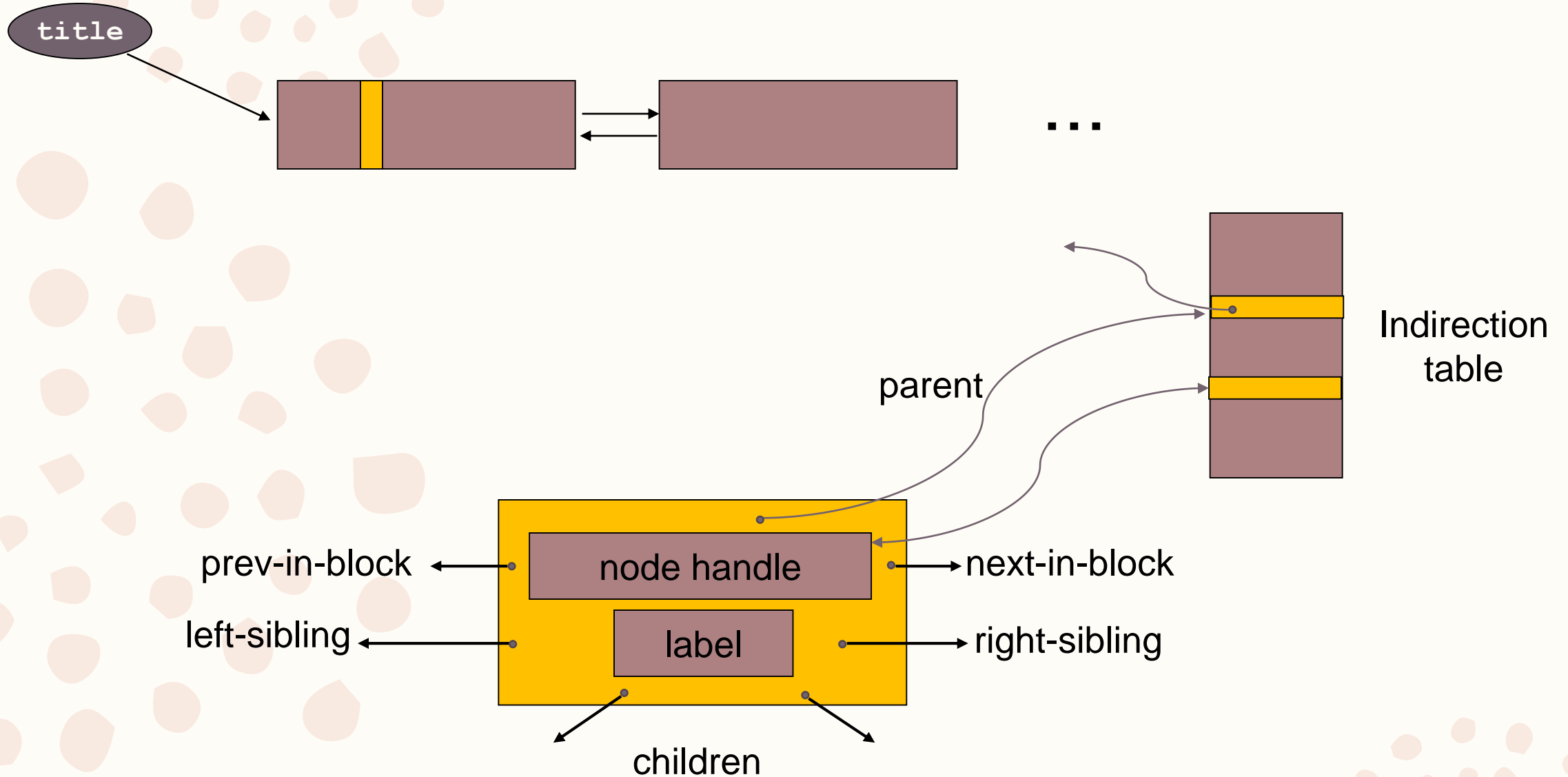
Termination initiated by the client

# Descriptive Schema (Data Guide)

```
<library>
<book>
  <title>Foundation on databases</title>
  <author>Abiteboul</author>
  <author>Hull</author>
  <author>Vianu</author>
</book>
. . .
<book>
  <title>An Introduction to Database
  Systems</title>
  <author>Date</author>
  <issue>
    <publisher>Addison-Wesley</publisher>
    <year>2004</year>
  </issue>
</book>
<paper>
  <title>A Relational Model for Large Shared
  Data Banks</title>
  <author>Codd</author>
</paper>
. . .
<paper>
  <title>The Complexity of Relational Query
  Languages</title>
  <author>Codd</author>
</paper>
</library>
```



# Data Structure – Node Descriptor





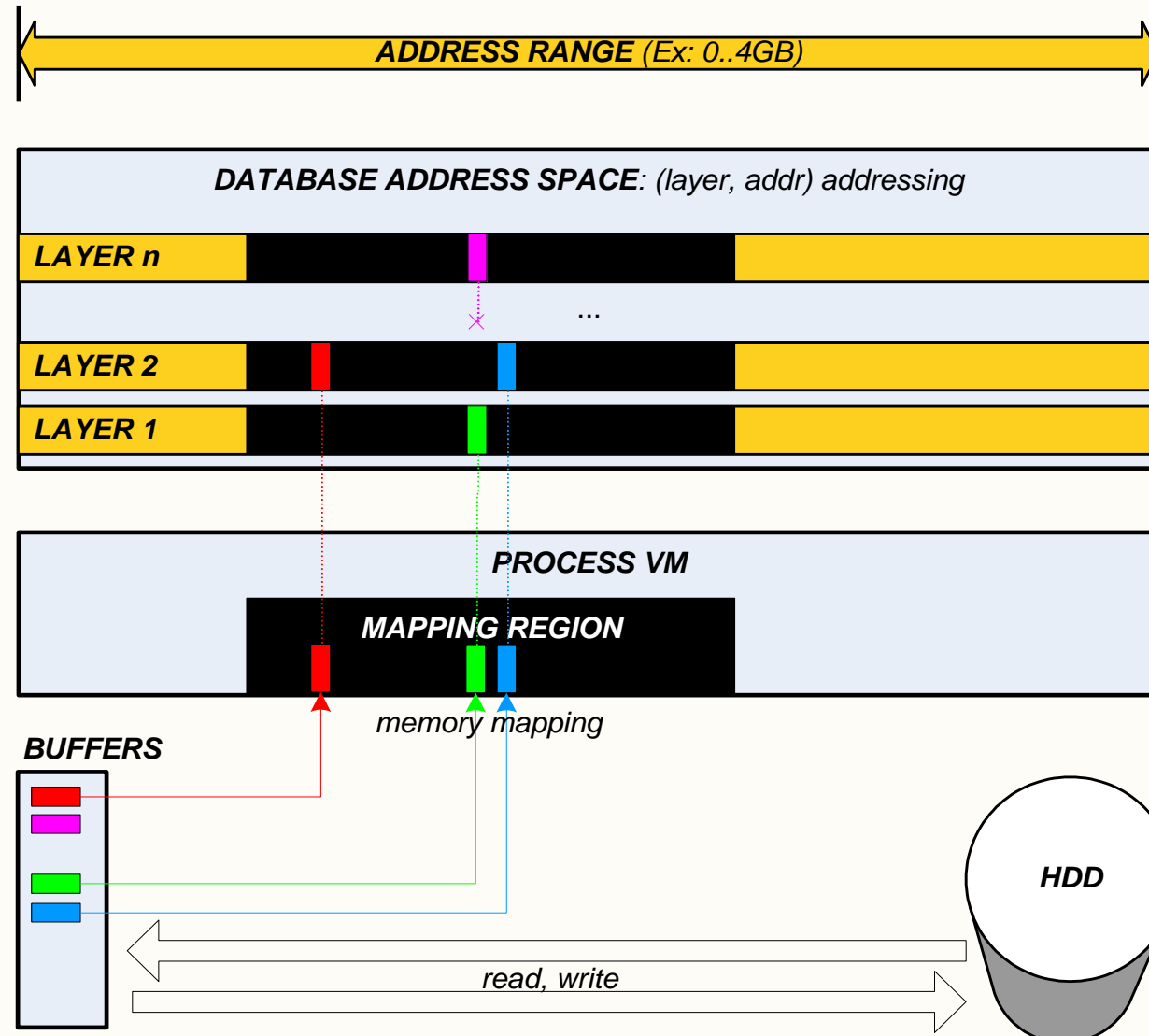
# Memory Management

Database pointers need conversion from DAS(Database Address Space) to VAS(Virtual Address Space)

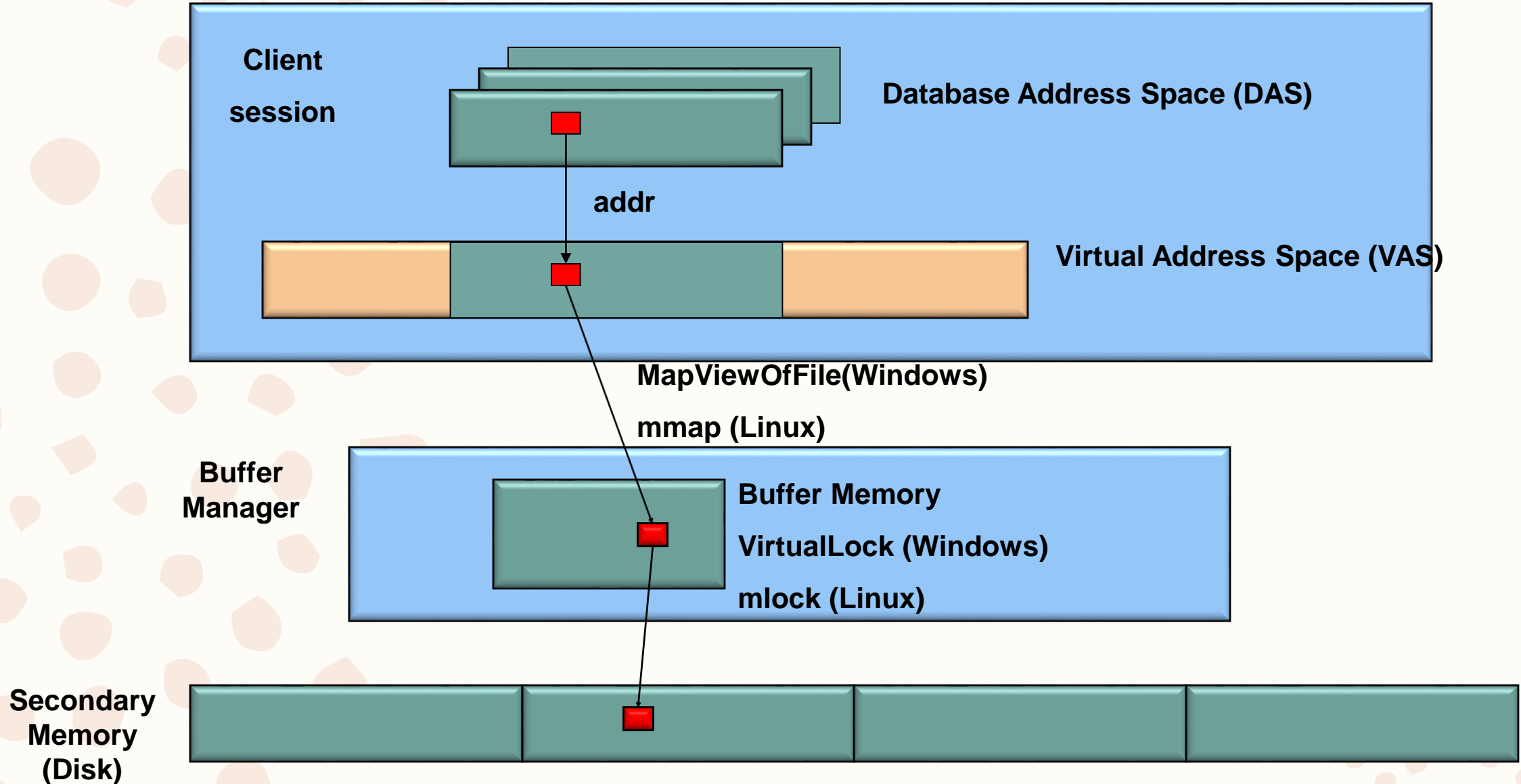
Pointer swizzling creates and overhead cost

SEDNA eliminates the need of swizzling

# Memory Management



# Sedna Memory Management



# Advantages of Sedna Memory Management

---

- 64-bit virtual address space
- Pointer dereferencing is comparable to conventional pointers
- Avoids costly pointer swizzling





Java™



python

# Supported API Packages

Language:	Name:	Type:	Support:	Author:
Java	Sedna XQJ API	Driver	+	Charles Foster
Java	Sedna XML:DB API	Driver	+	Charles Foster
Python	Python API	Driver	+	Sedna Team
-	mod_sedna v.1.5	Apache HTTP Server Module	+	Sedna Team
Java	SednaAdmin	Admin GUI, Web Admin GUI	+	Flávio R. C. Sousa
Python	zif.sedna	Driver	+	Jim Washington
C#	.NET API (sources/svn)	Driver		Mindtouch, John Wood
PHP	PHP API	Driver	+	Sedna Team
Scheme	Sedna API for Chicken Scheme	Chicken Egg	+	Felix Winkelmann
Python	SednaPy (DBAPI 2.0)	Driver		Steve Howe
-	SDBAdmin (for Windows)	Admin GUI		Tony Scott
Delphi	Delphi API for Sedna	Driver	+	Alexander Kardailsky
Ruby	Ruby API for Sedna	Driver		Rolf Timmermans
Perl	Perl API for Sedna	Driver	+	Daniel Ruoso
Catalyst	Catalyst Model for Sedna	Adapter	+	Daniel Ruoso
Component Pascal	BlackBox Subsystem for Sedna	Driver	+	Eugene Temirgaleev
Haskell	Haskell Bindings	Driver		Eric Jones



# JAVA API

## Session

```
SednaConnection getConnection(String url,  
                               String DBName,  
                               String user,  
                               String password)  
  
throws DriverException
```

## Transaction

```
public void begin() throws DriverException  
public void commit() throws DriverException  
public void rollback() throws DriverException
```

## Statement

```
public SednaStatement createStatement()  
    throws DriverException  
public void loadDocument(InputStream in,  
                          String doc_name)  
    throws DriverException, IOException  
public void loadDocument(InputStream in,  
                          String doc_name,  
                          String col_name)  
    throws DriverException, IOException  
public boolean execute(String queryText)  
    throws DriverException
```

## Result

```
public SednaSerializedResult getSerializedResult()  
public String next() throws DriverException
```

# Connection Drivers

JAVA

```
sedna.java x
1 import ru.ispras.sedna.driver.*;
2 class Client {
3     public static void main(String args[]) {
4
5         SednaConnection con = null;
6         try {
7             /* Get a connection */
8             con = DatabaseManager.getConnection("localhost",
9                                               "testdb",
10                                              "SYSTEM",
11                                              "MANAGER");
12
13             /* Begin a new transaction */
14             con.begin();
15
16             /* Create statement */
17             SednaStatement st = con.createStatement();
18
19             /* Load XML into the database */
20             System.out.println("Loading data ...");
21             boolean res;
22             res = st.execute("LOAD 'C:/region.xml' 'region'");
```

C++

```
Sedna x * Sedna x Sedna.c
1 struct SednaConnection conn = SEDNA_CONNECTION_INITIALIZER;
2
3 int SEconnect(SednaConnection* conn,
4              const char* url,
5              const char* db_name,
6              const char* login,
7              const char* password);
8
9 int SEclose(SednaConnection* conn);
```

# DB GUI Interface

Support for Sedna 3.x.x

Validation XML (DTD, XSD)

Manage all aspects of databases

Show metadata (DataGuide)

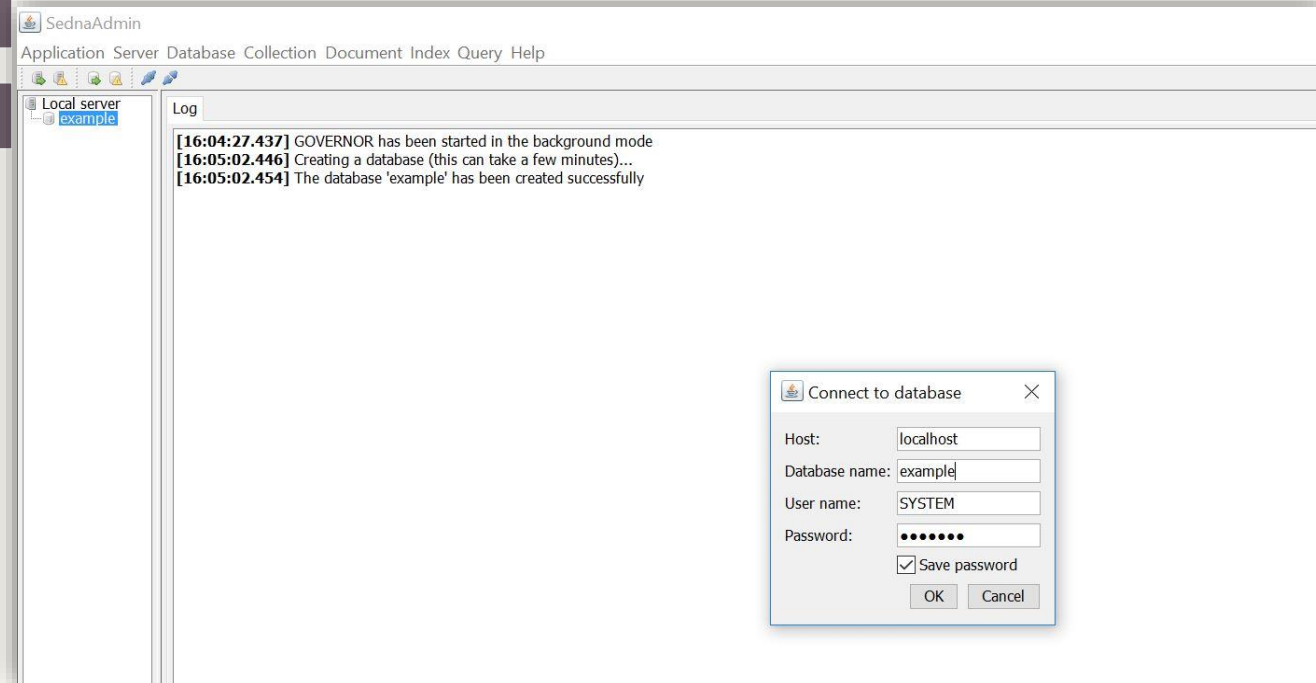
Execute XQuery and update extensions

Platform Independent (Java)

Easy to install and configure

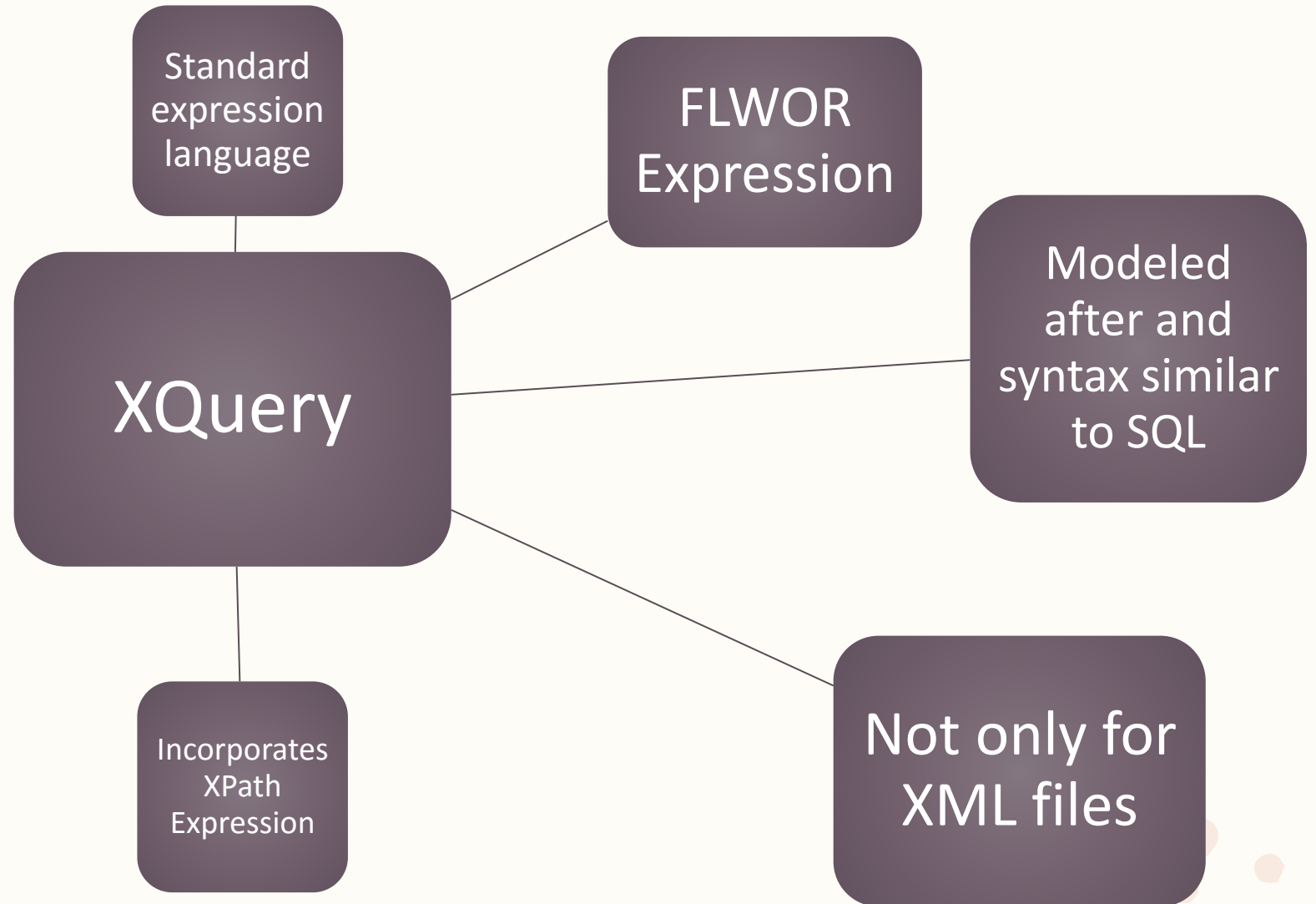
## Features

## SednaAdmin Interface





# XQuery



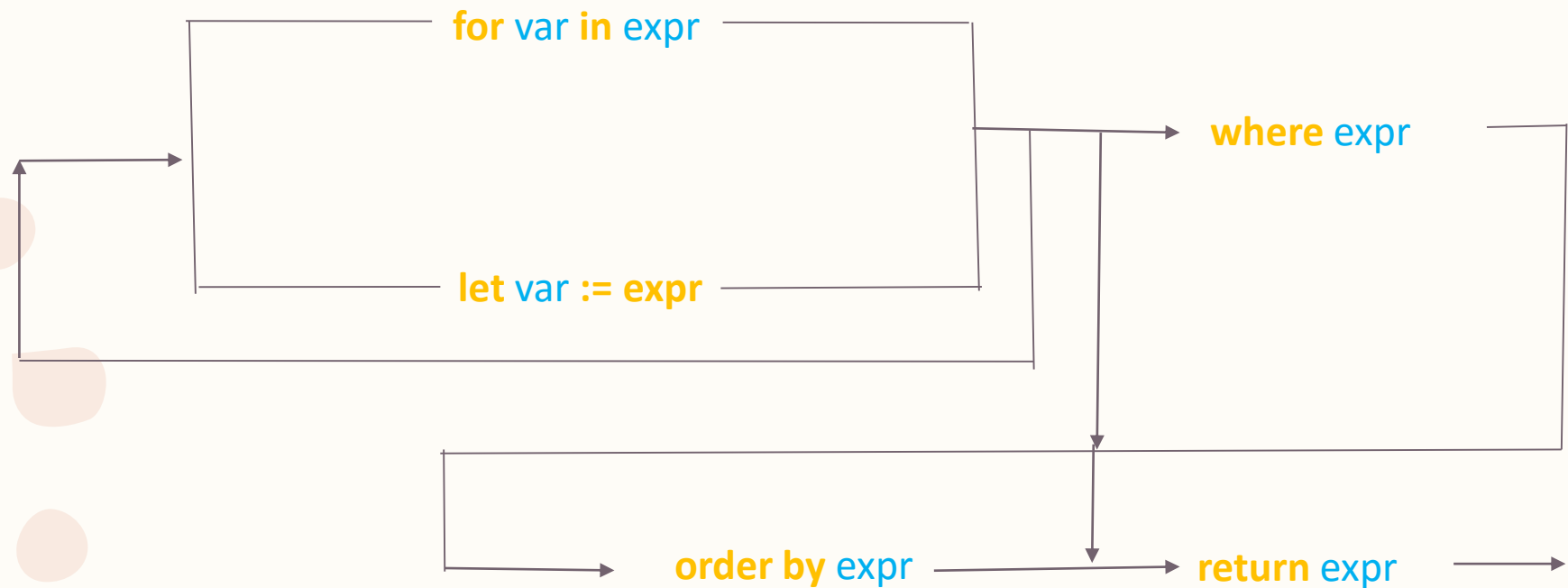
# XQuery Functions

---

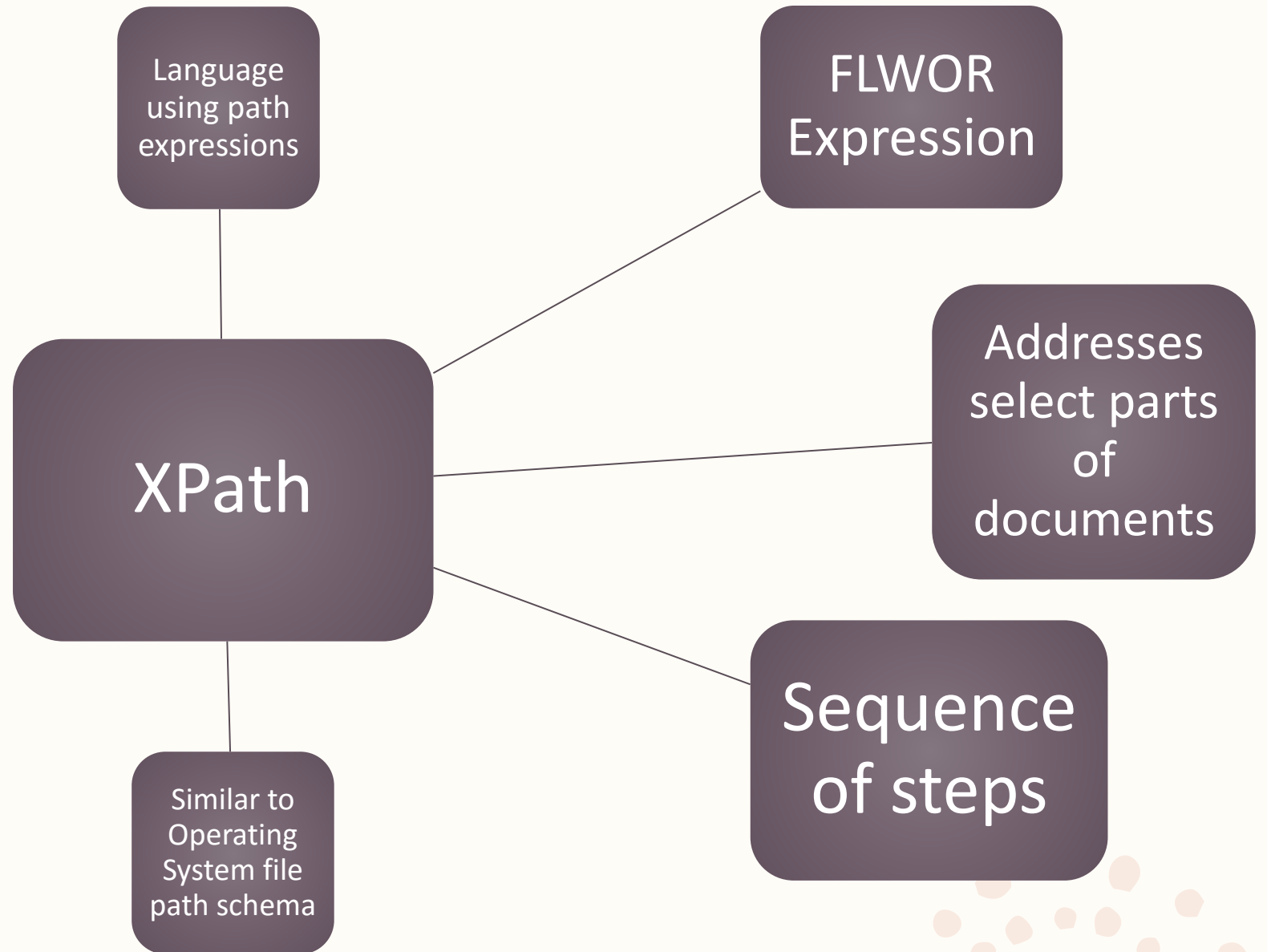
- Built-in functions
  - Xpath/XQuery function library, e.g., document( )
  - Aggregation functions, e.g., avg, sum, count, max, min
- User-defined functions
  - Ex.: xq. Functions.xq

```
declare function prefix:function_name($parameter as datatype?...)
as returnDatatype?
{
  function body...
};
```

# XQuery FLWOR Expression



# XPath?



# Operating Systems

# XPath

`/` = the root directory

`/users/dave/foo` = the (one) file named `foo` in `dave` in `users`

`foo` = the (one) file named `foo` in the current directory

`.` = the current directory

`..` = the parent directory

`/users/dave/*` = all the files in `/users/dave`

`/library` = the root element (if named `library`)

`/library/book/chapter/section` = *every* `section` element in a `chapter` in every `book` in the `library`

`section` = *every* `section` element that is a child of the current element

`.` = the current element

`..` = parent of the current element

`/library/book/chapter/*` = all the elements in `/library/book/chapter`

# XQuery Triggers

---

- Specific to XML data
- Native based
- XML data hierarchy dependent
- Similar purposes as relational database triggers: integrity constraints, event-based applications, statistics gathering, monitoring specific data changes...

# XQuery Triggers

```
CREATE TRIGGER "trigger-name"  
(BEFORE | AFTER) (INSERT | DELETE | REPLACE)  
ON <XPath-expression> (,<XPath-expression>)*  
( FOR EACH NODE | FOR EACH STATEMENT )  
DO  
{  
    (<XUpdate-expression($NEW, $OLD, $WHERE)>;)*  
    <XQuery-expression($NEW, $OLD, $WHERE)>  
}
```

```
DROP TRIGGER "trigger_name"
```

# XQuery Trigger Example

```
CREATE TRIGGER "tr1"  
BEFORE INSERT  
ON doc("auction")/site//person  
FOR EACH NODE  
DO  
{  
  if($NEW/age < 14)  
  then  
    <person>{attribute id {$NEW/@id}}  
      {$NEW/*}  
      <age-group>young</age-group>  
    </person>  
  else  
    <person>{attribute id {$NEW/@id}}  
      {$NEW/*}  
      <age-group>adult</age-group>  
    </person>; }  
}
```

*Node-level before-trigger*



# Transactions & Recovery



## ➤ ACID Transactions

- Atomicity – rollback procedure
- Consistency – by design
- Isolation – S2PL and snapshots
- Durability - two level recovery scheme

## ➤ Multiversioning – Concurrency Control

## ➤ Logging and Recovery



# Multi-versioning Scheme

## ➤ Page-Level Versioning

- Snapshot-based schema with data elements as pages.
- Transaction transparent solution
- All the logic is encapsulated in the storage manager.
- No worry of garbage collection.

## ➤ Read-Only Transactions

- Support faster execution for Read Only transactions (Queries)
- Isolation not needed: Non-Blocking processing or non-S2PL

# Logging & Recovery Mechanisms

## ➤ Normal Processing

- Logging of all main operations such as insert node, create index, etc.
- Transaction-consistent snapshot at checkpoints – **Persistent snapshot**

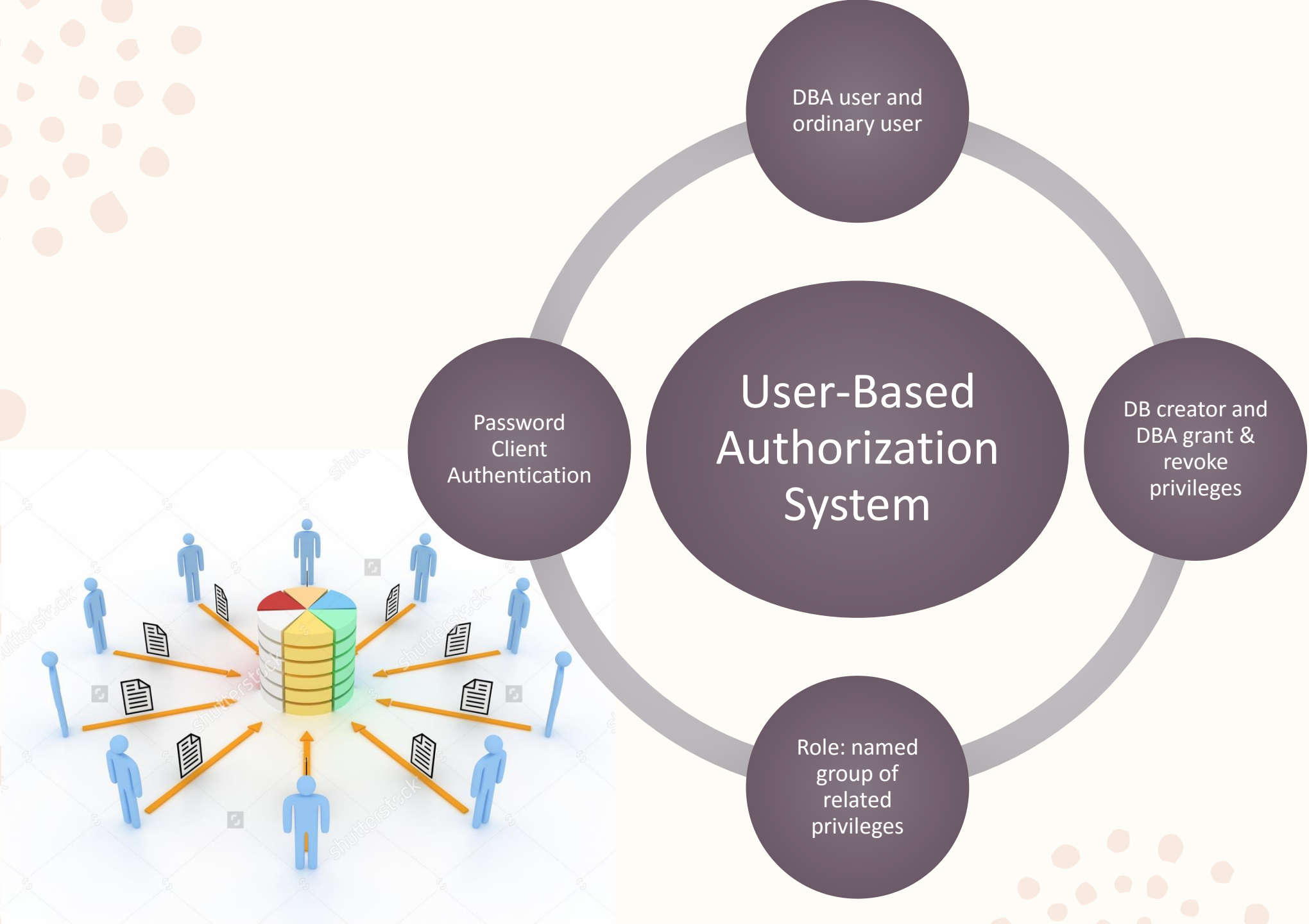
## ➤ Rollback Processing

- Undo the operations using the data from the already created log.

## ➤ Two-Level Recovery Process

- Restoring transaction-consistent state of DB using the persistent snapshot.
- Redo the necessary operations using the log generated





# Privileges

DELETE

RENAME  
RETRIEVE-  
METADATA

CREATE-  
USER

CREATE  
DOCUMENT

CREATE  
COLLECTION

CREATE  
INDEX

INSERT

CREATE FT  
INDEX

QUERY

DROP

LOAD

LOAD  
MODULE

CREATE  
TRIGGER

# Sedna and its competitors

## Sedna vs. X-Hive

- ▶ 100 MB XMark Benchmark
- ▶ AMD Athlon 64 2.00 GHz, 1 GB of RAM.
- ▶ Timeout: 2000

	X-Hive	Sedna
XPath	1.2	0.8
XPath, pos, trans	4.0	1.7
Complex XPath	6.8	2.2
Id comparison	3.7	2.3
XPath, count	3.0	0.4
FLWR	4.6	0.5
FLWR, count	16.1	0.8
Join(1,2)	*	1046
Join(1,2,3)	*	1350
Group by	34.8	81
Semijoin	*	1664
Complex semijoin	*	373
Struct. XPath + trans	3.3	1.3
Contains substring	10.4	8.4
Long XPath	1.8	0.1
Nested Long XPath	2.3	0.7
Empty	3.1	2.1
Function Calls	2.6	1.0
Sorting	24.3	3.5
Trans(nested XPaths)	3.3	2.5

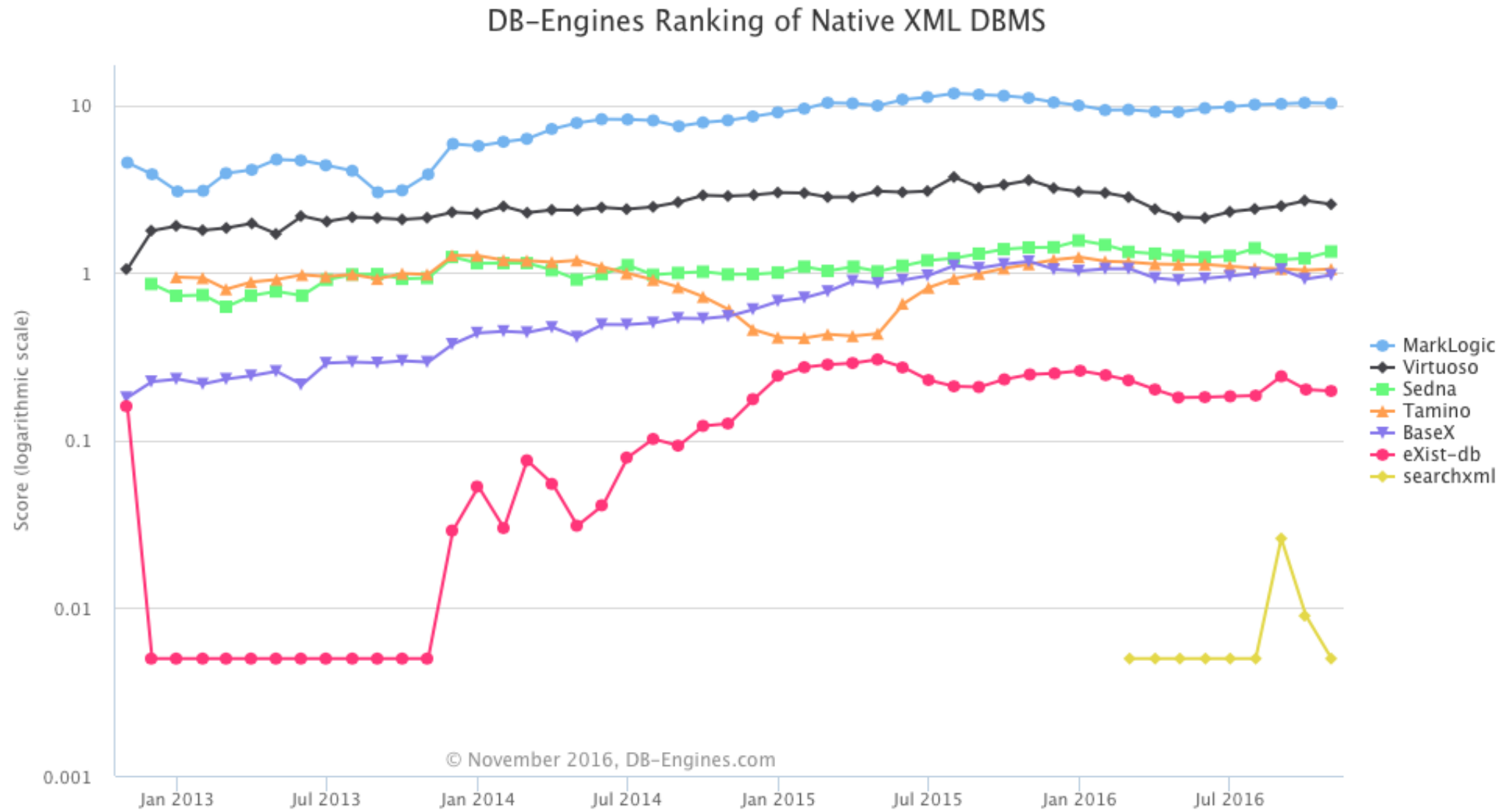
# Sedna and its competitors

## Sedna vs. Berkeley XML DB

- ▶ 12MB XMark benchmark
- ▶ AMD Athlon 64 2.00 GHz, 1 GB of RAM.
- ▶ Timeout: 2000

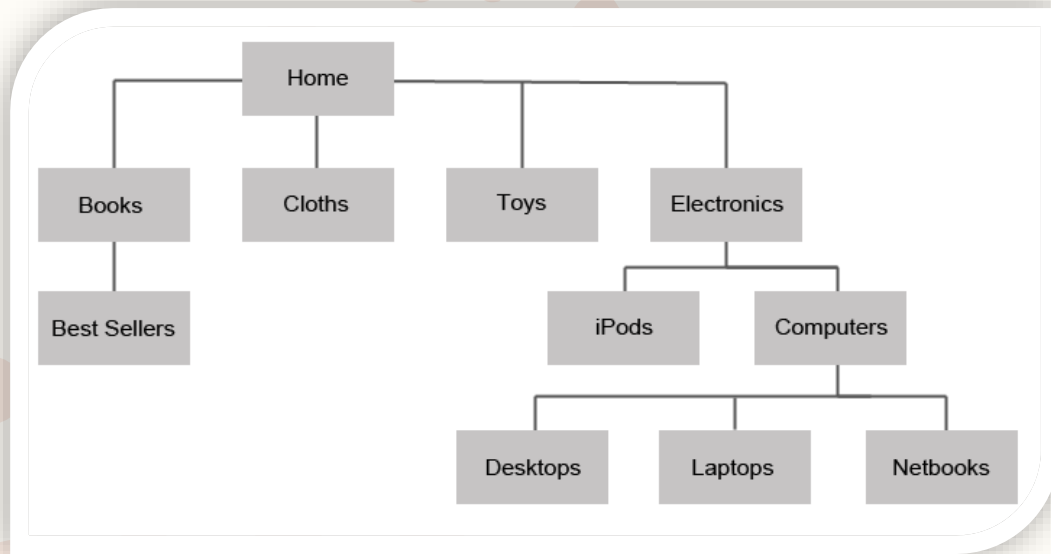
	<b>BDB node</b>	<b>Sedna</b>
<b>XPath</b>	0.172	0.109
<b>XPath, pos, trans</b>	0.421	0.188
<b>Complex XPath</b>	0.625	0.141
<b>Id comparison</b>	0.969	0.250
<b>XPath, count</b>	0.188	0.094
<b>FLWR</b>	1.297	0.109
<b>FLWR, count</b>	7.016	0.172
<b>Join(1,2)</b>	263.219	11.109
<b>Join(1,2,3)</b>	428.453	14.125
<b>Group by</b>	42.250	2.219
<b>Semijoin</b>	281.781	34.625
<b>Complex semijoin</b>	81.453	10.969
<b>Struct. XPath, trans</b>	0.109	0.454
<b>Contains substring</b>	3.797	2.485
<b>Long XPath</b>	0.219	0.047
<b>Nested Long XPath</b>	0.234	0.156
<b>Empty</b>	0.312	0.125
<b>Function Calls</b>	*	0.062
<b>Sorting</b>	*	0.43
<b>Trans(nested XPathes)</b>	1.016	0.156

# XML Databases - Comparison





# Where To Use Sedna?



# Sedna Users



**UNIVERSITY OF  
ALBERTA**



**stilo**



**WikiXMLDB**

# How big is it today?

<https://sourceforge.net/projects/sedna/>

[See All Activities >](#)

## User Reviews

[Write a Review >](#)

Filter:

All

### lagivan



We've been using Sedna XML:DB on Production for about a year as a part of a big product management system based on the xml data model. We're quite happy with this solution although we'd still like to see some more features available (like importing XQuery modules from the local file system resource or from a jar).

Posted 12/01/2011

### externalreality



Few, realize how powerful and flexible an XML database actually is and Sedna is a strong offering in the category. Its free, fast, and flexible. Many man hours went into this XML and it shows. Good thought went into this document store and it certainly shows aswell.

Posted 09/15/2011

### mozinsur






Very cool Native-XML Database with triggers.

Posted 08/31/2009

# Why Sedna Failed to Compete?

7 systems in ranking, November 2016

Rank			DBMS	Database Model	Score		
Nov 2016	Oct 2016	Nov 2015			Nov 2016	Oct 2016	Nov 2015
1.	1.	1.	MarkLogic	Multi-model 	10.22	-0.08	-0.78
2.	2.	2.	Virtuoso	Multi-model 	2.56	-0.13	-1.00
3.	3.	3.	Sedna	Native XML DBMS	1.34	+0.12	-0.07
4.	4.	↑ 5.	Tamino	Native XML DBMS	1.05	+0.02	-0.07
5.	5.	↓ 4.	BaseX	Native XML DBMS	0.97	+0.05	-0.20
6.	6.	6.	eXist-db	Native XML DBMS	0.20	-0.00	-0.05
7.	7.		searchxml	Multi-model 	0.00	-0.01	

# Why Sedna Failed to Compete?

---

- General weakness of XML
- Switching from RDBMS is expensive
- Frozen further development
- Lack of support of Xquery 3.0
- Doesn't support XML 1.1
- Scalability is supported but decreases the performance

# Competitors

---

## Supported APIs

Name ⇄	XQJ ⇄	XML:DB ⇄	RESTful ⇄	RESTXQ ⇄	WebDAV ⇄
BaseX	Yes	Yes	Yes	Yes	Yes
eXist	Yes	Yes	Yes	Yes	Yes
MarkLogic Server	Yes	No	Yes	Yes	Yes
Qizx	No	No	Yes	No	No
Sedna	Yes	Yes	No	No	No

# References

---

<http://elib.mi.sanu.ac.rs/files/journals/kjm/30/kjom3013.pdf>

<https://cs.brown.edu/~akalinin/papers/sedna-sigmod.pdf>

<https://www.sedna.org>





Thank  
you