12.–15.09.2010
in Nürnberg

Herbstcampus

Wissenstransfer
par excellence

Keep Persistence Simple, Stupid

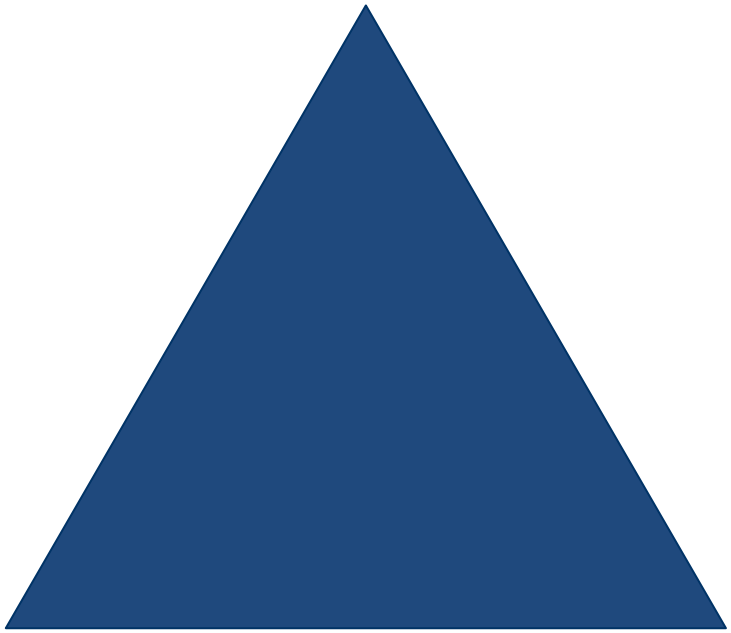
A possible future for Java Persistence

Robert Bräutigam

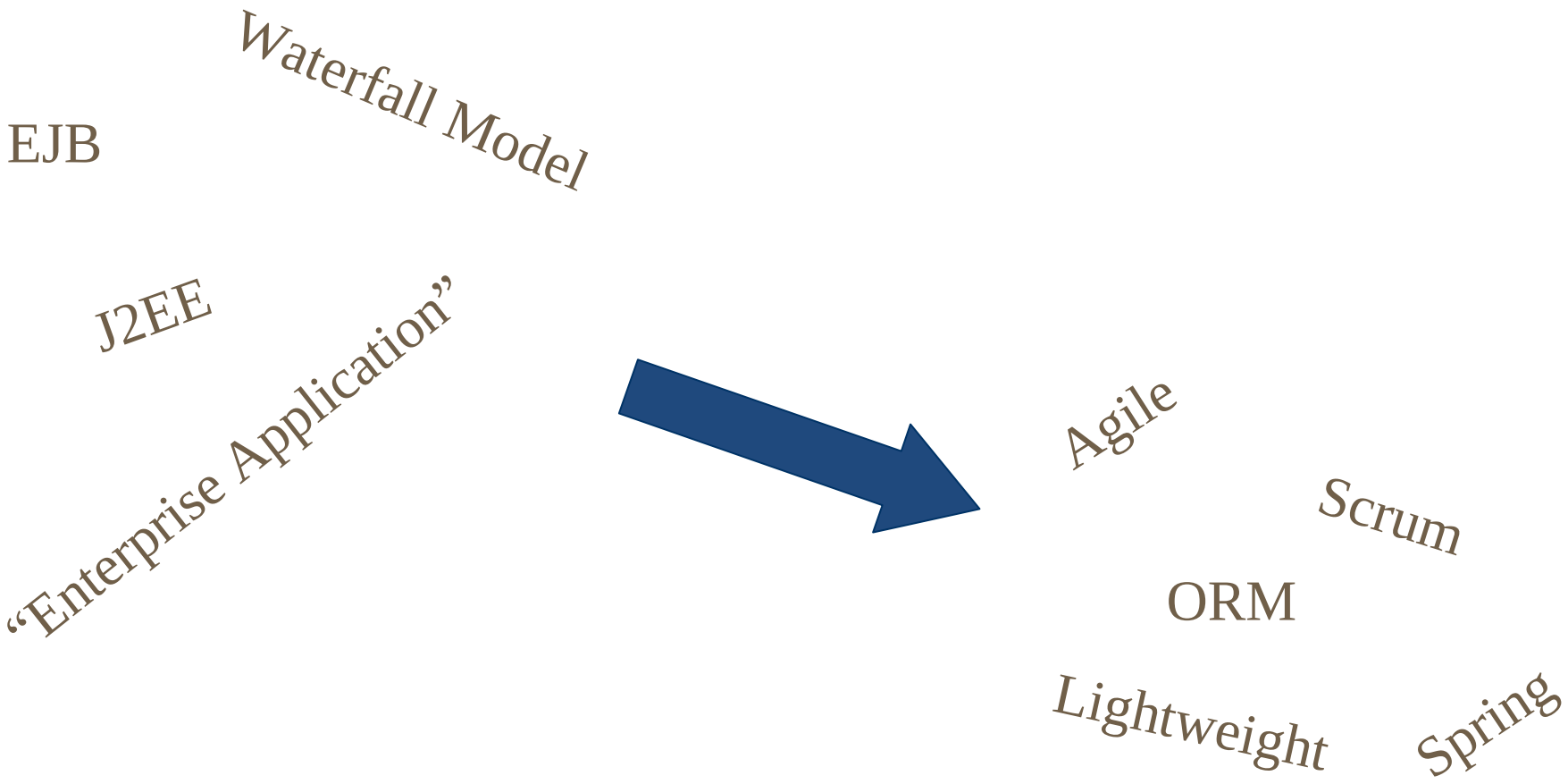
adidas AG

The KISS principle.

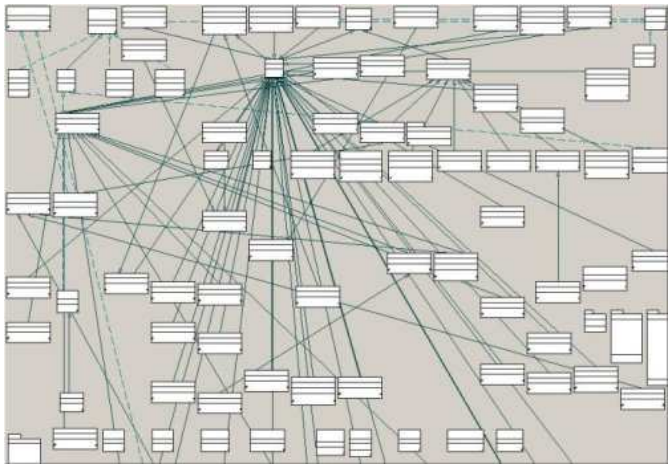
Complexity – Cost – Functions



Don’t take my word for it? → Trends



We’re not there! Some current issues.



ORM

Introducing BeanKeeper

Getting Started

```
...
   <dependency>
        <groupId>hu.netmind.beankeeper</groupId>
        <artifactId>beankeeper</artifactId>
        <version>2.6.3</version>
   </dependency>
...
```

```
// Allocate the "Store" object
Store store = new Store("org.postgresql.Driver",
   "jdbc:postgresql://localhost:5432/test");
// Create the book
Book book = new Book("Snow Crash","ISBN");
book.getAuthors().add(new Author("Neal Stephenson"));
// Save your first object
store.save(book);
```

Domain Model Object example

```
public class Book
{
    private String title;
    private String isbn;
    private List<Author> authors;

    public Book()
    {
    }

    public Book(String title, String isbn)
    {
        this.title=title;
        this.isbn=isbn;
        authors = new ArrayList();
    }

    // Normal setter/getters
    ...
}
```

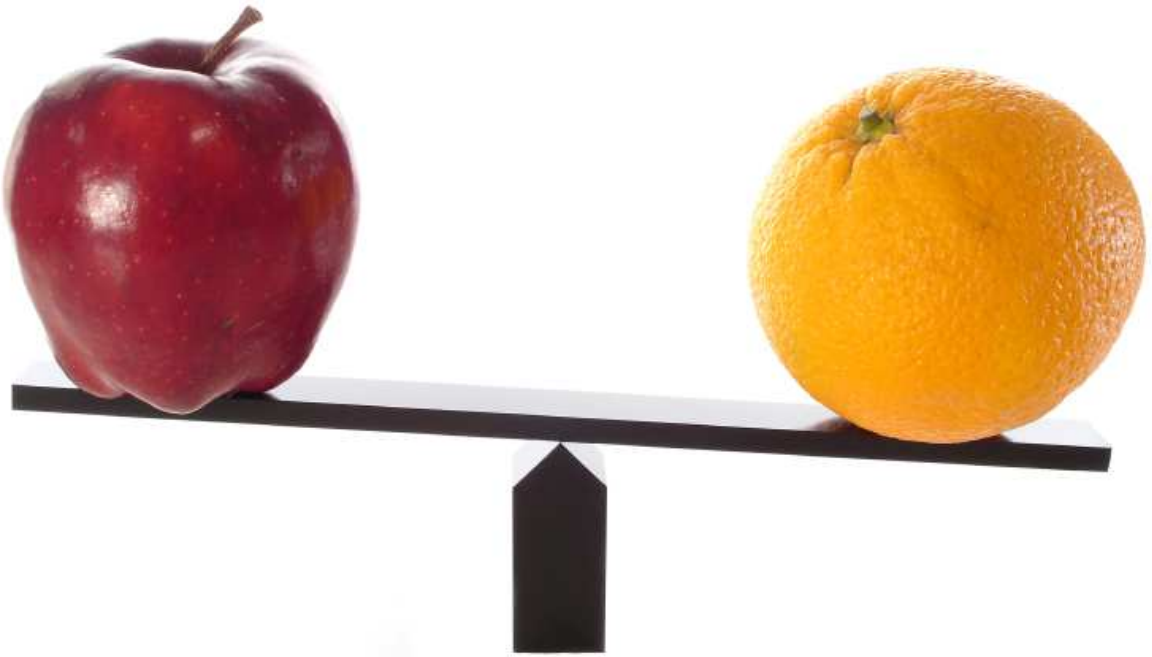
Store API

```
// Saving (creating or updating) an object in the database
public void save(Object obj);

// Removing an object from database
public void remove(Object obj);

// Getting objects from the store
public List find(String statement, Object... parameters);


// Getting the lock tracker service
public LockTracker getLockTracker();

// Getting the transaction tracker
public TransactionTracker getTransactionTracker();

…
```

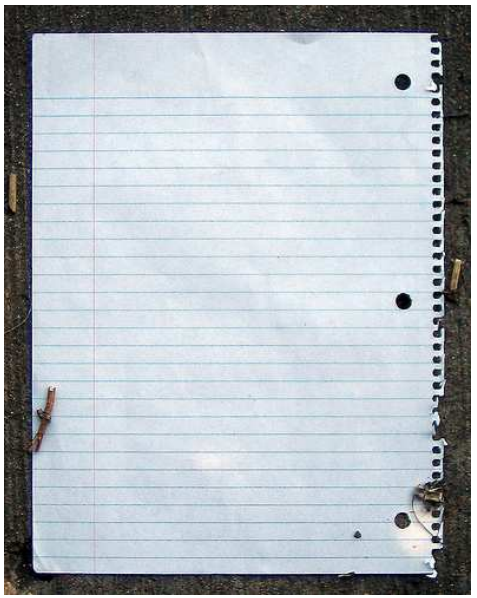
Some possible listing constructs

```
List<Book> books = (List<Book>) store.find(“find book”);

find book where title='Snow Crash' order by title asc

find author where book.authors contains author and book.title='Snow Crash'

find book where book.mainauthor.name = 'Neil Stephenson'

find object

find serializable

find permission where permission.user.name='Joe' at '2010-06-20'

view book.title, book.mainauthor.name where book.publishdate < '2010-06-20'
```

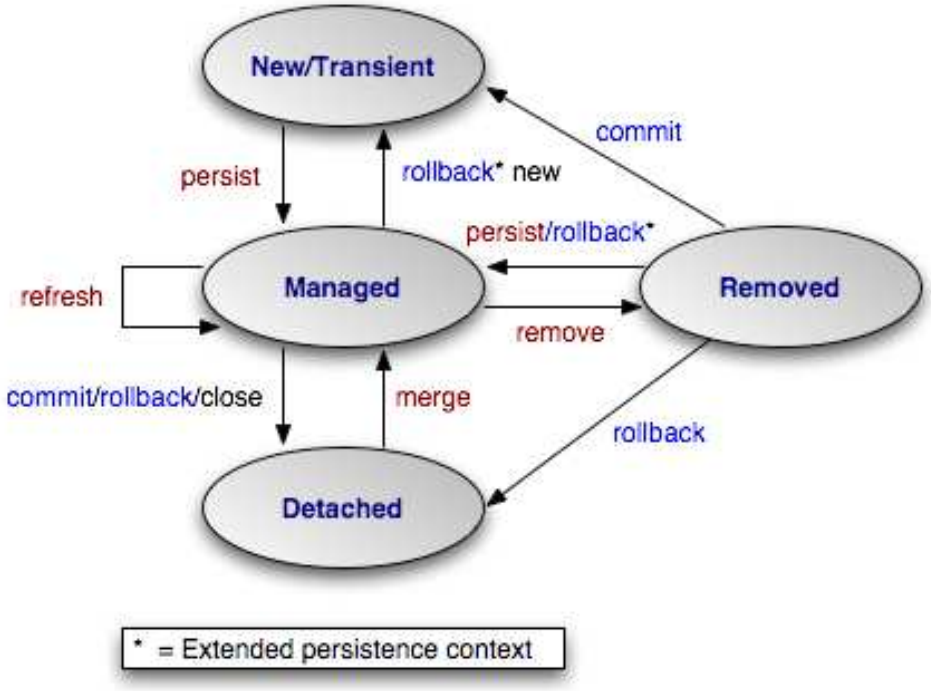
A biased comparison
BeanKeeper vs. JPA 2.0

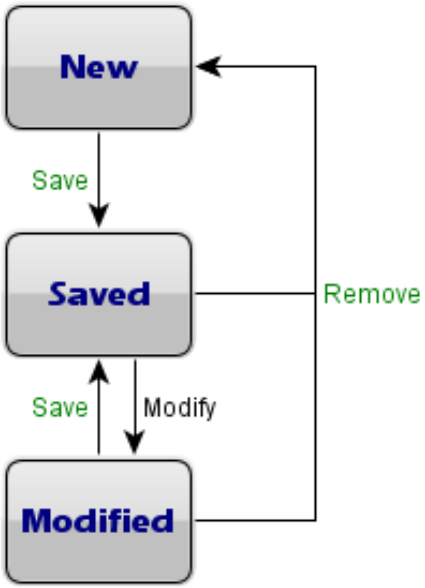


Learning Curve



Object Lifecycle





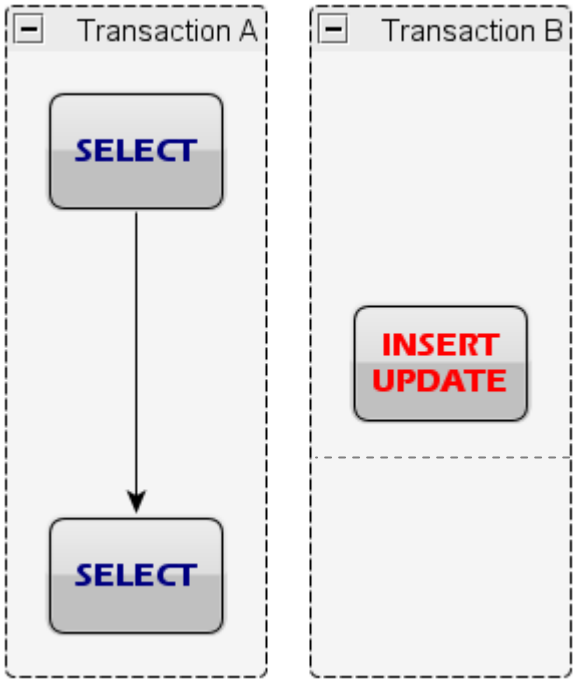
Transparent result list / container paging

```
// Getting millions of books
List<Book> books = (List<Book>) store.find(“find book”);
// Process each without any further configuration
for ( Book book : books )
  process(book);
```

```
// Getting a service which has millions of events
Service service = store.findSingle(“find service where name = ‘Test‘);
// Process each event without any further configuration
for ( Event event : service.getEvents() )
  process(event);
```

Low cost transaction isolation

1. Serializable
   - No dirty reads
   - Repeatable result list
   - No phantom reads
2. Repeatable Read
   - No dirty reads
   - Repeatable entities
3. Read Committed
   - No dirty reads
4. Read Uncommitted



Additional goodies

- list ordering
- schema evolution
- polymorphic pessimistic locking
- “is null” operator
- reserved word usage
- …

http://beankeeper.netmind.hu

12.–15.09.2010
in Nürnberg

Herbstcampus

Wissenstransfer
par excellence

Vielen Dank!

Robert Bräutigam

adidas AG