

Objektorientierte Datenbanken – Back to the future –

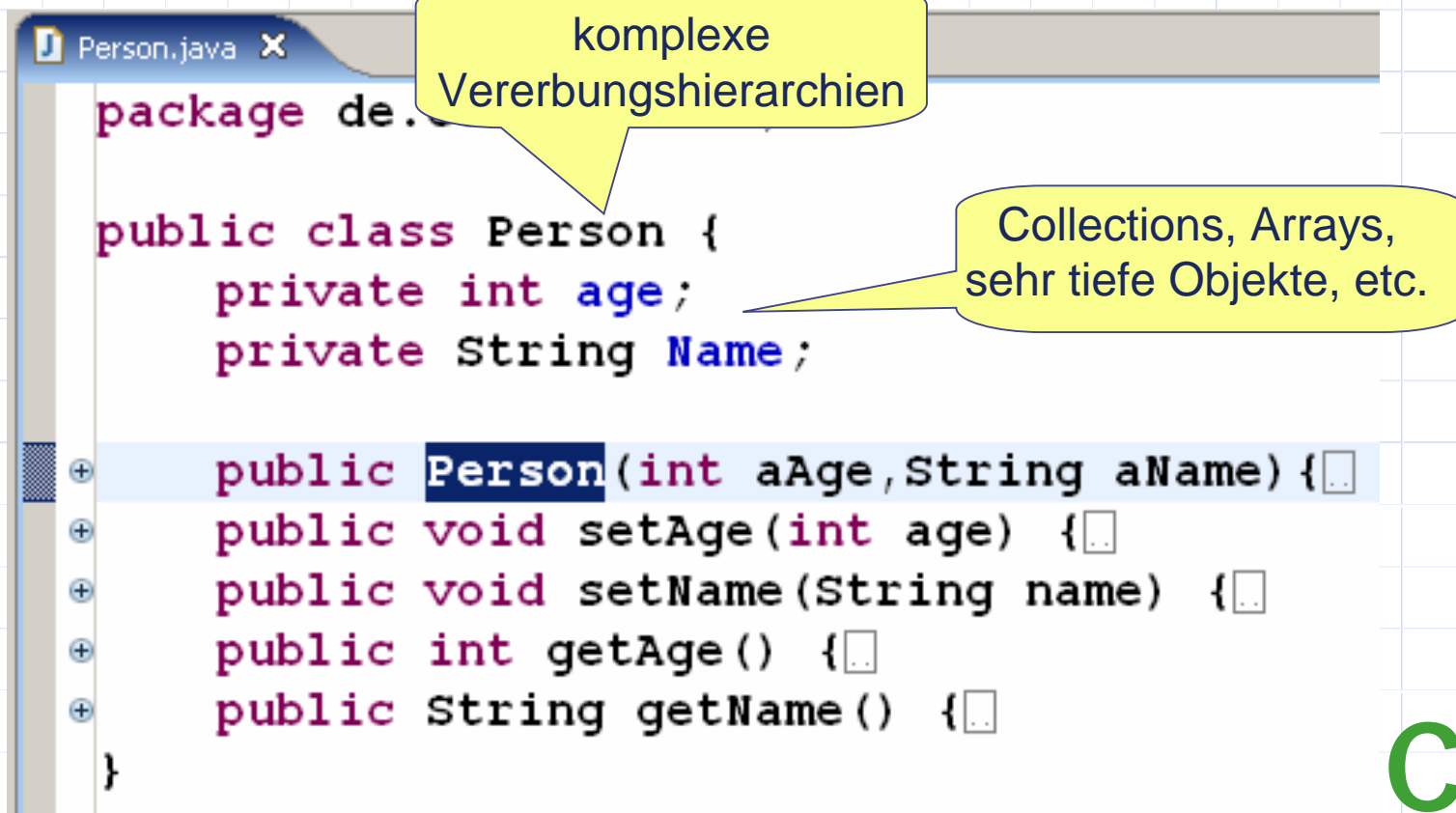
Stefan Edlich

Eine (zukunfts)kritische
Bestandsaufnahme am Beispiel von db4o

05.12.06

Kann das gehen?

- Komplexe Klassen**P** & Abfragesprachen **P**



The screenshot shows a Java IDE window titled 'Person.java'. The code defines a 'Person' class with private fields 'age' and 'Name', and public methods for setting and getting these values. A yellow callout bubble points to the class name 'Person' with the text 'komplexe Vererbungshierarchien'. Another yellow callout bubble points to the field declarations with the text 'Collections, Arrays, sehr tiefe Objekte, etc.'. A large green letter 'C' is positioned in the bottom right corner of the code area.

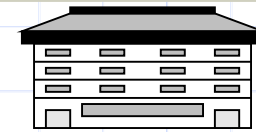
```
package de...  
  
public class Person {  
    private int age;  
    private String Name;  
  
    public Person(int aAge, String aName) {..  
    public void setAge(int age) {..  
    public void setName(String name) {..  
    public int getAge() {..  
    public String getName() {..  
}
```


db4o Geschäftsmodell

Nach 10 Jahren
gründet sich 2004
wieder eine ODBMS
Company!?

mysql Modell

virtual Company



Dual GPL
Lizenzmodell

Zero Administration

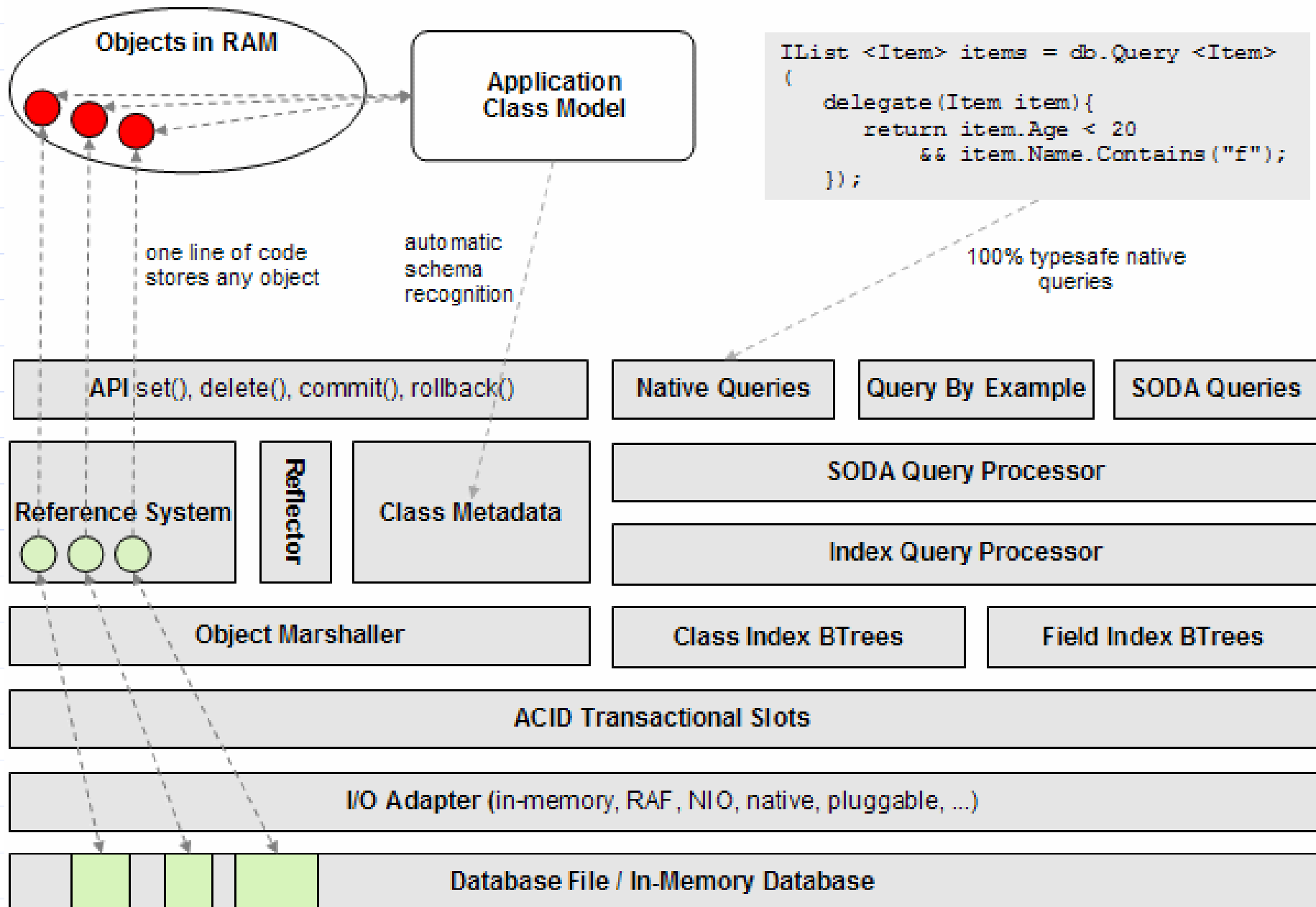
(unclustered)
Performance

Low
Footprint

no Consulting
Product

Embedded
Boom

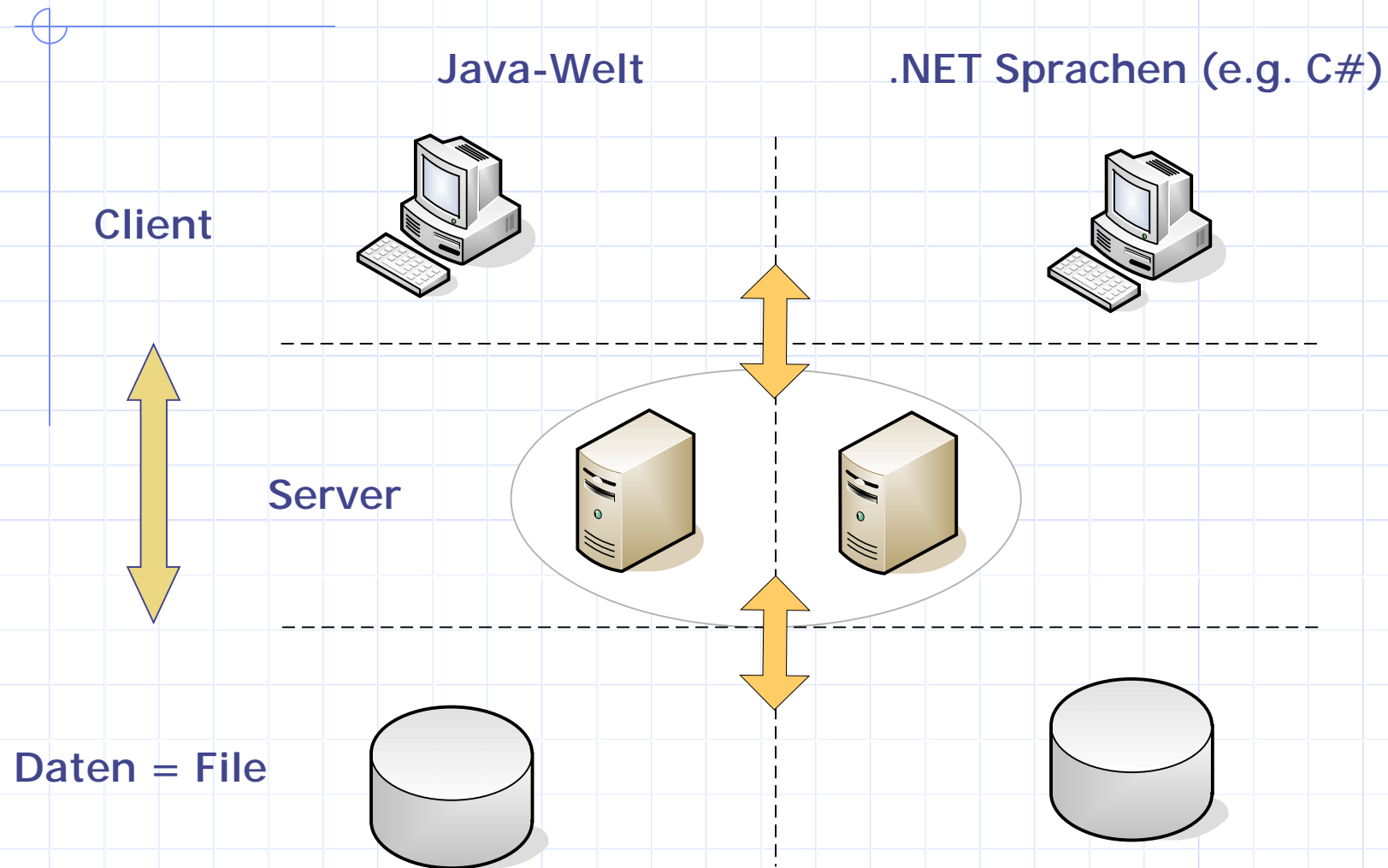
db4o Architektur



Was zeichnet db4o aus?

- 600k und wenig Memory Requirements
- ACID (Read Committed)
 - keine feineren Levels
- Speichert einfach so jedes Objekt
 - Kein Pre-Compiler, keine Inheritance, kein Interface
 - Beliebige Klassen mit beliebig vielen und tiefen Subobjekten und Collections
 - => anderes Denken
 - Welche Bedeutung hat ein delete(o*) ?
 - Muss ich mich um Ids kümmern?

Cross-Platform



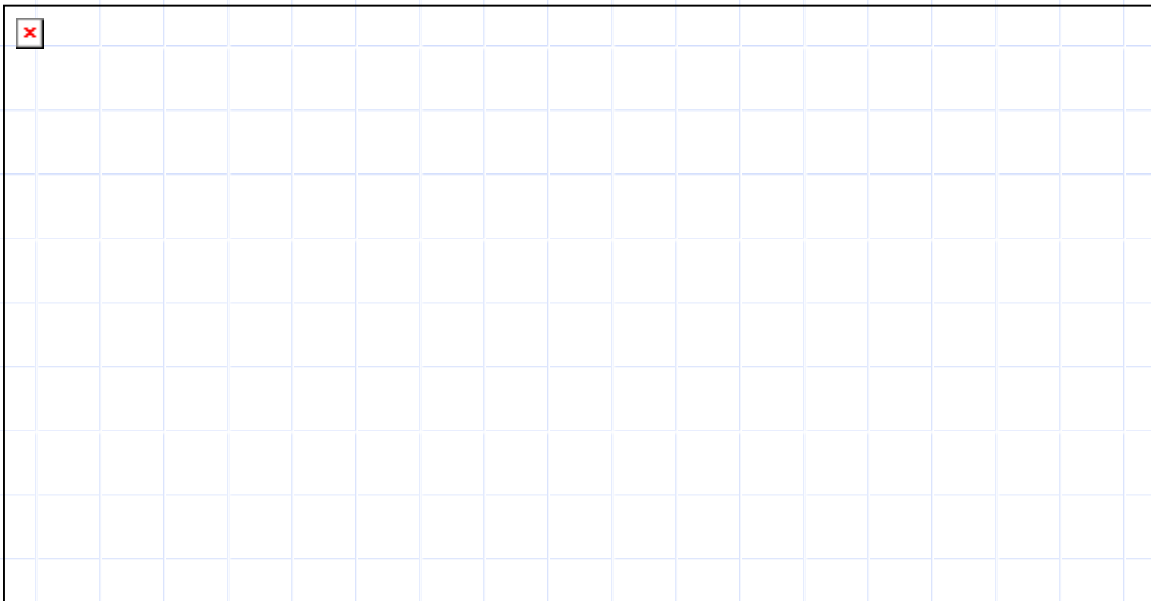
Client-Server & Replikation

```
Object Container db = Db4o.openFile("C:/portal.db");
```



```
Object Container db = Db4o.openServer("C:/server.db", 8732);
```

```
Object Container db = Db4o.openClient("localhost", 8732, "user1", "pwd");
```



beliebige
Objekte

db4o Schema Evolution

- delete Field [transparent (ASD)
- add Field [transparent (ASD)
- rename Field [eine Zeile

```
Db4o.configure().objectClass("MyClass.class").  
    objectField("oldField").rename("newField");
```

- change Field Type [transparent (ASD)
- rename Class [eine Zeile

```
Db4o.configure().objectClass("MyPackage.MyClass").  
    rename("NewPackage.NewClass")
```

[Defragment

Contact

edlich@gmail.com

<http://edlich.de>

23.11.06