MEGA MODELLING OF HIBERNATE

WITH CONCENTRATION ON THE

ASPECT: READING FROM DB

Introduction

- What is Hibernate…
  - … in general?
  - … and why it is used?
  - … and why it is an important example?
  - … and how it is used?
- Presentation of the mega models
- Sources

What is Hibernate?

Does it mean turning your PC off?

No, it means turning your DB with Java really on!

What is Hibernate? – Facts

- Very popular Java-Library
- Object Persistence Framework (OPF) / Object Relational Mapper (ORM)
- Compatible with common Database Management System (DBMS)

→ You use it to store POJOs in your DB.

What does Hibernate offer?

- SQL mapping by XML-files or by annotations
- Supports EJB-Entities
- An own query language (HQL) similar to SQL
- Supports native SQL (with hydration) and stored procedures
- Reengineering possibilities to generate classes from SQL schemas
- Implements an own DBMS for testing (HSQL-DB-Server)

What is Hibernate? – Why?

- + Easier to use
- + Abstracts from the DBMS as additional layer
- + Increases security
- + Better maintainability
- ➨ Increased effort
  - ➨ Separating code and configuration
  - ➨ for putting the parts together (e.g. implementing boiler plates)
- ➨ Decreased performance for complex data

Why a mega model?

- used in more than 10.000 Java-projects
- 3.000 downloads per day

- Other OPF/ORMs could be better modeled
- This would be good to compare them
  - in their behavior and implementation
  - with the result of advantages and disadvantages

What is Hibernate – How?

```
// Take a POJO you would store in database
public class Department {
  private Long             id;          // Additional PK
  private String           name;
  private Set<Employee>    employees;
  private Set<Department> subdepts;

  public Long getId() { ... }
  private void setId(Long id) { ... }
  public String getName() { ... }
  public void setName(String name) { ... }
  public Set<Employee> getEmployees() { ... }
  private void setEmployees(Set<Employee> employees) { ... }
  public Set<Department> getSubdepts() { ... }
  private void setSubdepts(Set<Department> subdepts) { ... }
}
```

Code examples are taken from the 101companies project!

What is Hibernate – How?

```
<!-- Provide a mapping where you describe relations
    and foreign keys -->
<hibernate-mapping>
 <class name="org.softlang.company.Department" table="DEPARTMENT">
  <id name="id" column="ID">
    <generator class="native" />
  </id>
  <property name="name" />
  <set name="employees" cascade="all">
   <key column="DEPT_ID" />
   <one-to-many class="org.softlang.company.Employee" />
  </set>
  <set name="subdepts" cascade="all">
   <key column="DEPT_ID" />
   <one-to-many class="org.softlang.company.Department" />
  </set>
 </class>
</hibernate-mapping>
```

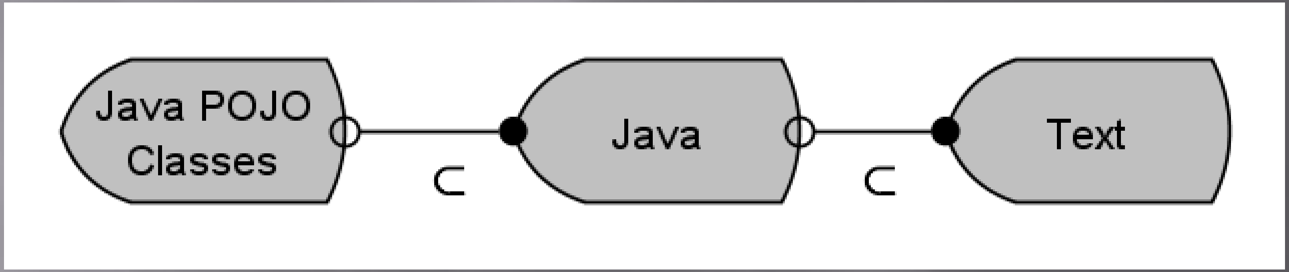
What is Hibernate – How?

```
<!-- Provide a configuration for your database -->
<hibernate-configuration>
  <session-factory>
    <!-- Database connection settings. -->
    <property name="connection.driver_class">org.hsqldb....
    <property name="connection.url">...
    <property name="connection.username">...
    <property name="connection.password">...
    ...
    <!-- This part lists all the mapping files present in the project -->
    <mapping resource="org/softlang/company/Company.hbm.xml" />
    <mapping resource="org/softlang/company/Department.hbm.xml" />
    <mapping resource="org/softlang/company/Employee.hbm.xml" />
    ...
  </session-factory>
</hibernate-configuration>
```

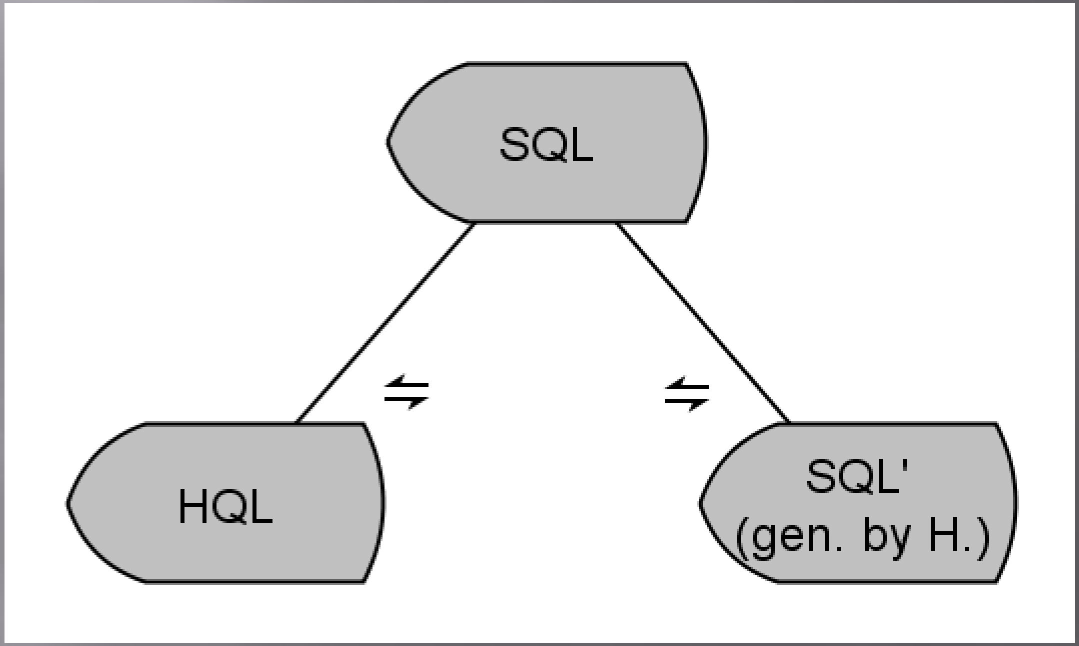
What is Hibernate – How?

```
<!- Load objects with HQL -->
public static void main(String[] args) {
  SessionFactory sessionFactory = new Configuration()
      .configure().buildSessionFactory();
  Session session = sessionFactory
      .getCurrentSession();
  session.beginTransaction();
  List<?> result = this.session.createQuery(
           "from Company where name = 'meganalysis'"
      ).list();
  Company company = (Company) result.getNext();
  Cut.cut(company);
  session.save(company);
  session.getTransaction().commit();
}
```

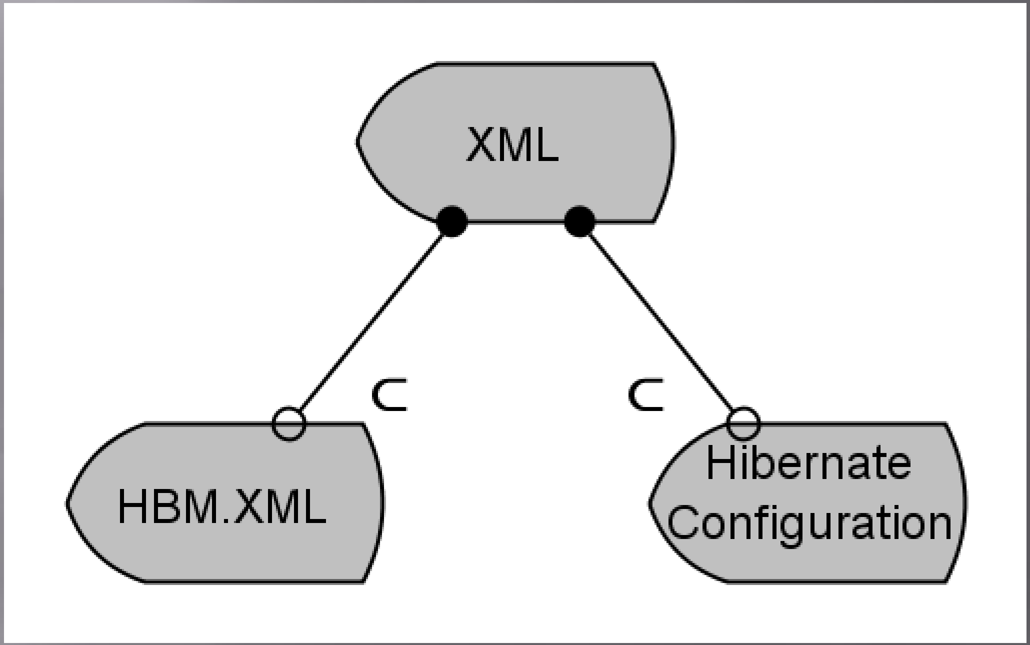
Mega model – Artifacts



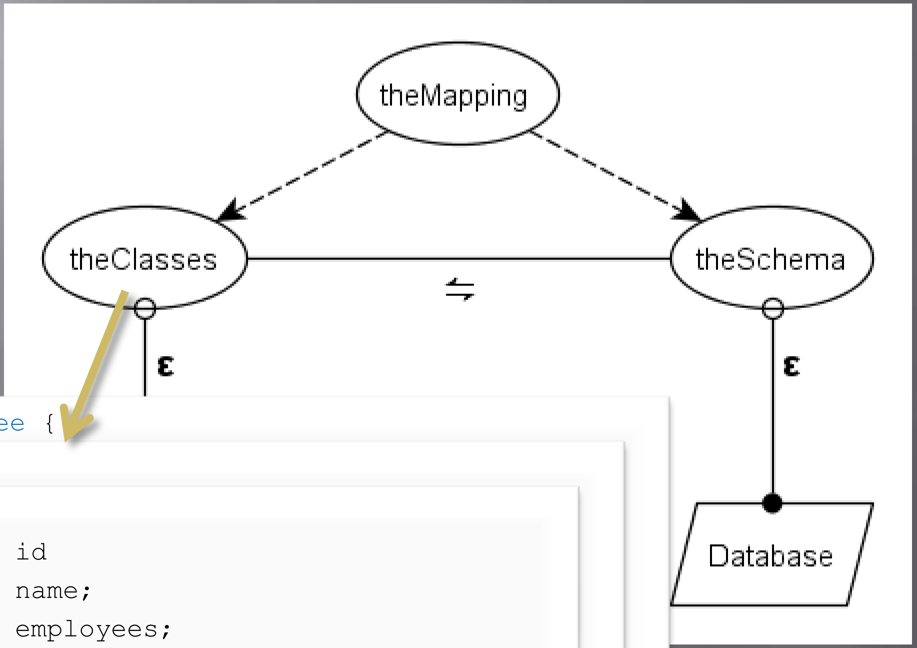
Mega model – Artifacts



Mega model – Artifacts



Mega model – O/R-Triangle

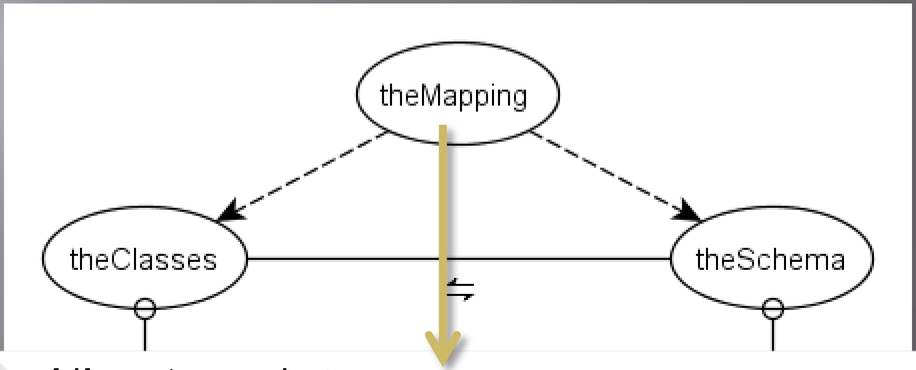


```
public class Employee {
```

```
public class Company {
```

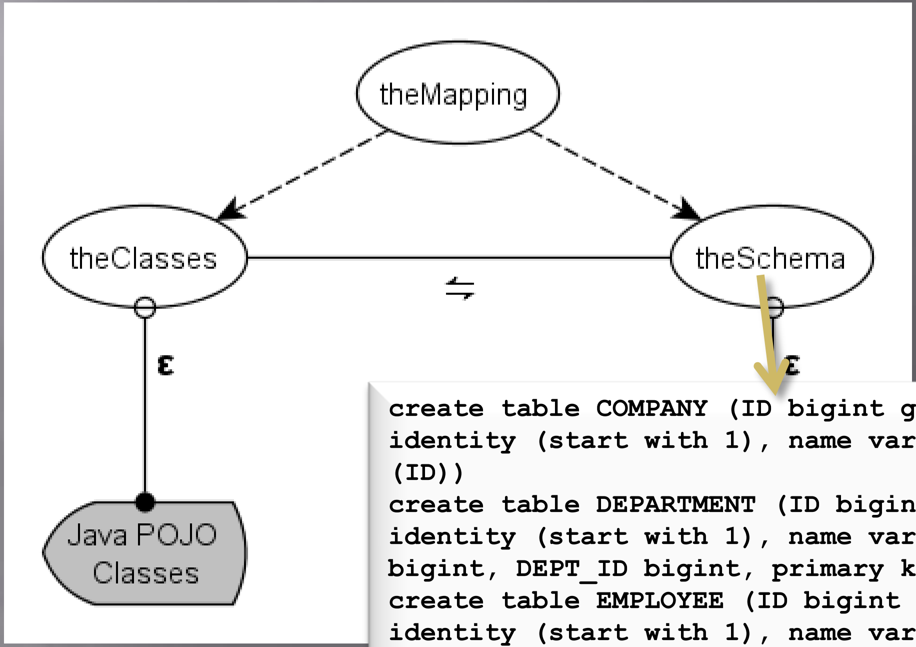
```
public class Department {
  private Long             id
  private String           name;
  private Set<Employee>    employees;
  private Set<Department> subdepts;

  public Long getId() { ... }
  private void setId(Long id) { ... }
  public String getName() { ... }
  public void setName(String name) { ... }
  public Set<Employee> getEmployees() { ... }
  private void setEmployees(Set<Employee> employees) { ... }
  public Set<Department> getSubdepts() { ... }
  private void setSubdepts(Set<Department> subdepts) { ... }
}
```

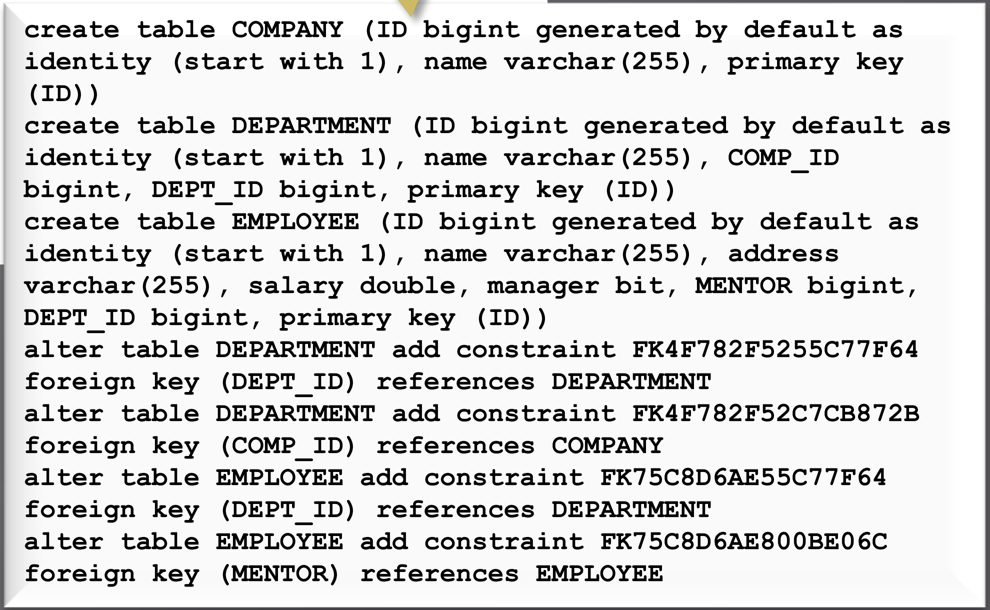
Mega model – O/R-Triangle



```
<hibernate-mapping>
 <class name="org.softlang.company.Department"
      table="DEPARTMENT">
  <id name="id" column="ID">
    <generator class="native" />
  </id>
  <property name="name" />
  <set name="employees" cascade="all">
   <key column="DEPT_ID" />
   <one-to-many class="org.softlang.company.Employee" />
  </set>
  <set name="subdepts" cascade="all">
   <key column="DEPT_ID" />
   <one-to-many class="org.softlang.company.Department" />
  </set>
 </class>
</hibernate-mapping>
```

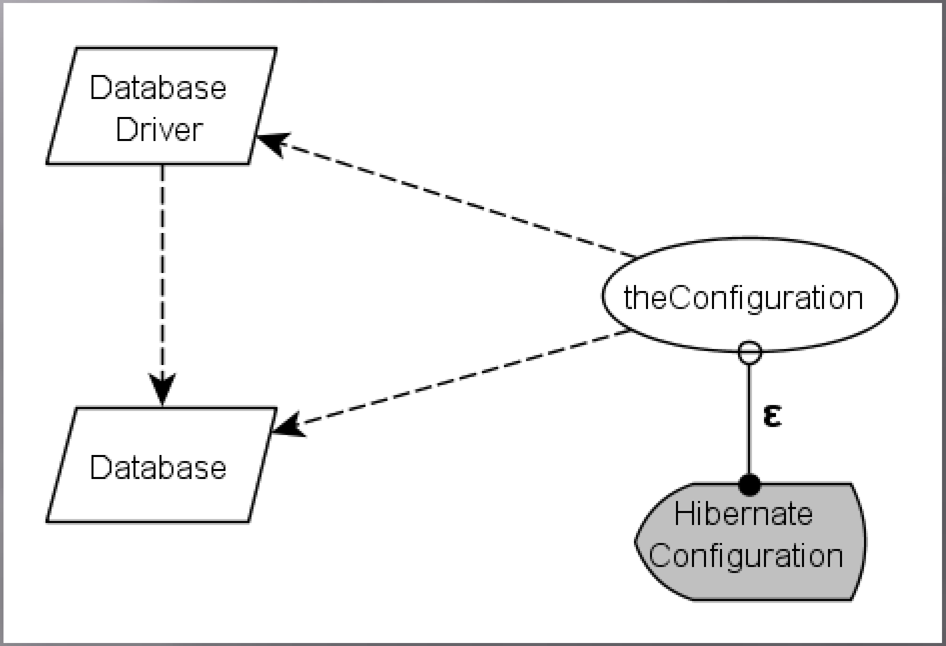
Mega model – O/R-Triangle



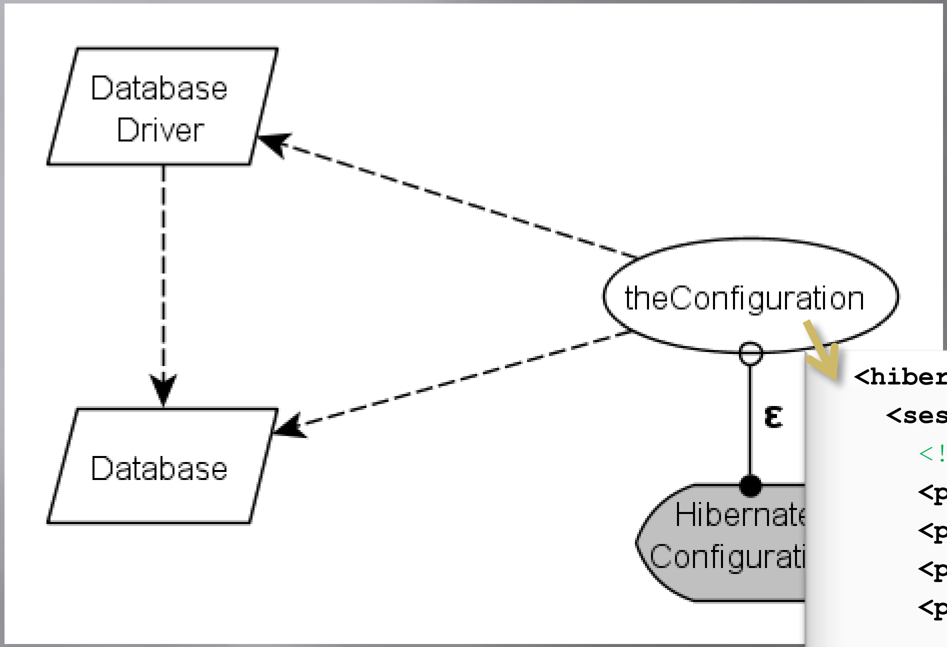


```
create table COMPANY (ID bigint generated by default as
identity (start with 1), name varchar(255), primary key
(ID))
create table DEPARTMENT (ID bigint generated by default as
identity (start with 1), name varchar(255), COMP_ID
bigint, DEPT_ID bigint, primary key (ID))
create table EMPLOYEE (ID bigint generated by default as
identity (start with 1), name varchar(255), address
varchar(255), salary double, manager bit, MENTOR bigint,
DEPT_ID bigint, primary key (ID))
alter table DEPARTMENT add constraint FK4F782F5255C77F64
foreign key (DEPT_ID) references DEPARTMENT
alter table DEPARTMENT add constraint FK4F782F52C7CB872B
foreign key (COMP_ID) references COMPANY
alter table EMPLOYEE add constraint FK75C8D6AE55C77F64
foreign key (DEPT_ID) references DEPARTMENT
alter table EMPLOYEE add constraint FK75C8D6AE800BE06C
foreign key (MENTOR) references EMPLOYEE
```

Mega model –Configuration

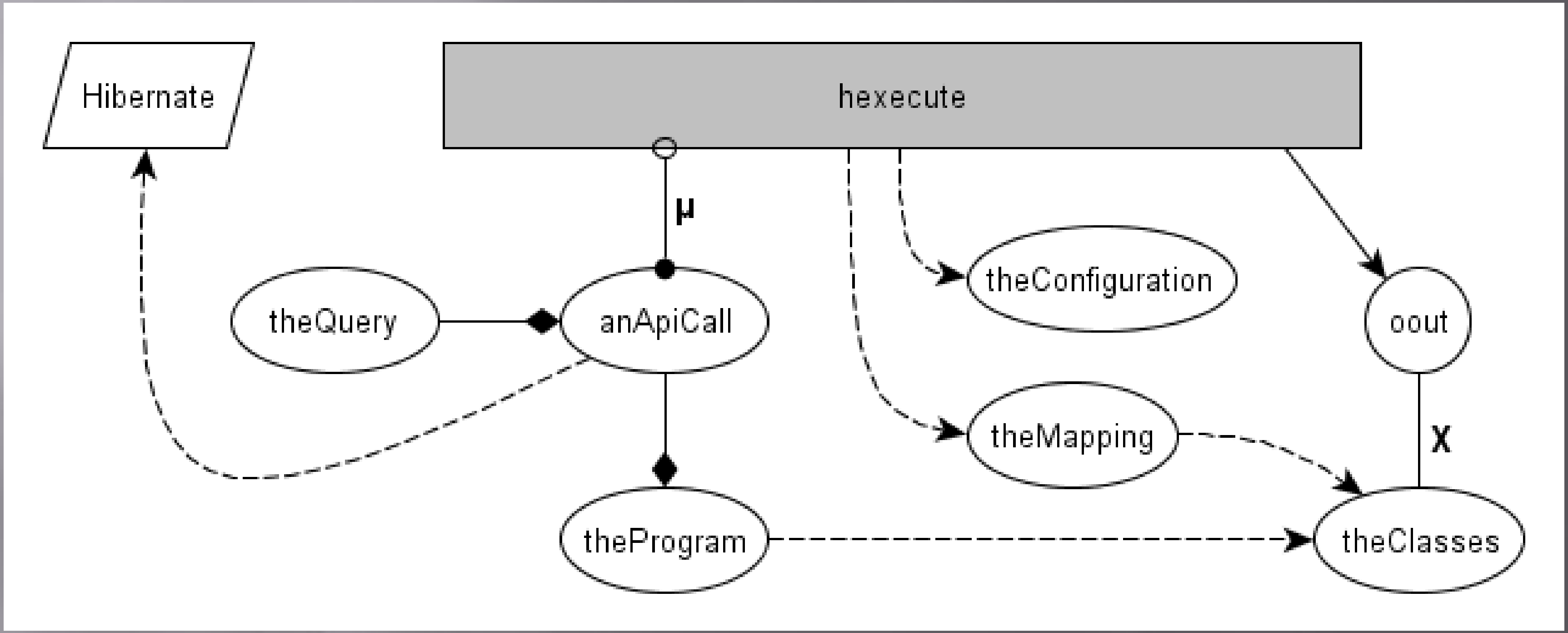


Mega model –Configuration



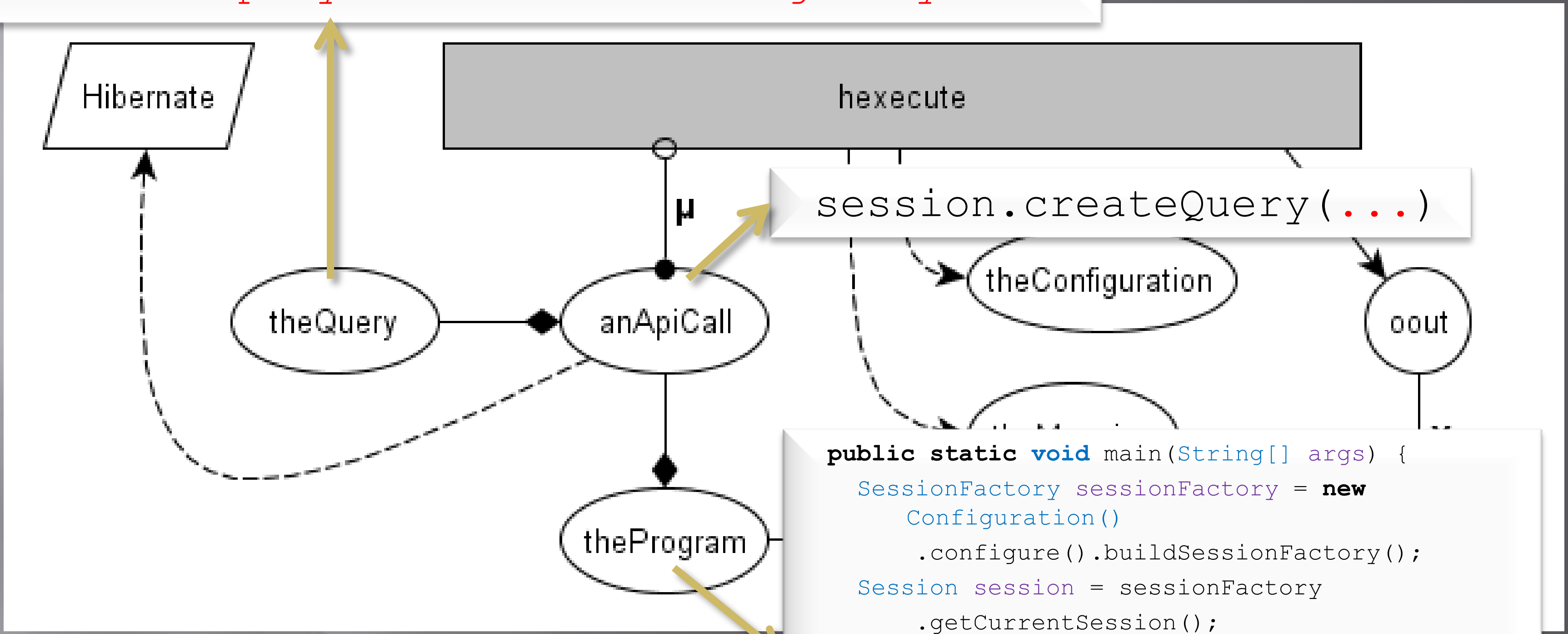
```
<hibernate-configuration>
  <session-factory>
    <!-- Database connection settings. -->
    <property name="connection.driver_class">org.hsqldb....
    <property name="connection.url">...
    <property name="connection.username">...
    <property name="connection.password">...
      ...
      <!-- This part lists all the mapping files present in
      the project -->
    <mapping
      resource="org/softlang/company/Company.hbm.xml" />
    <mapping
      resource="org/softlang/company/Department.hbm.xml" />
    <mapping
      resource="org/softlang/company/Employee.hbm.xml" />
    ...
  </session-factory>
</hibernate-configuration>
```

Mega model – Executing HQL-Select-Queries



Mega model – Executing HQL-Select-Queries

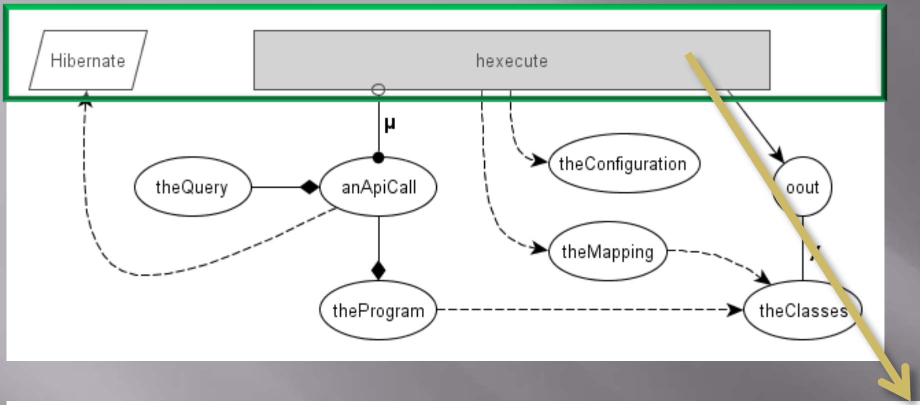
```
"from Company where name = 'meganalysis'"
```

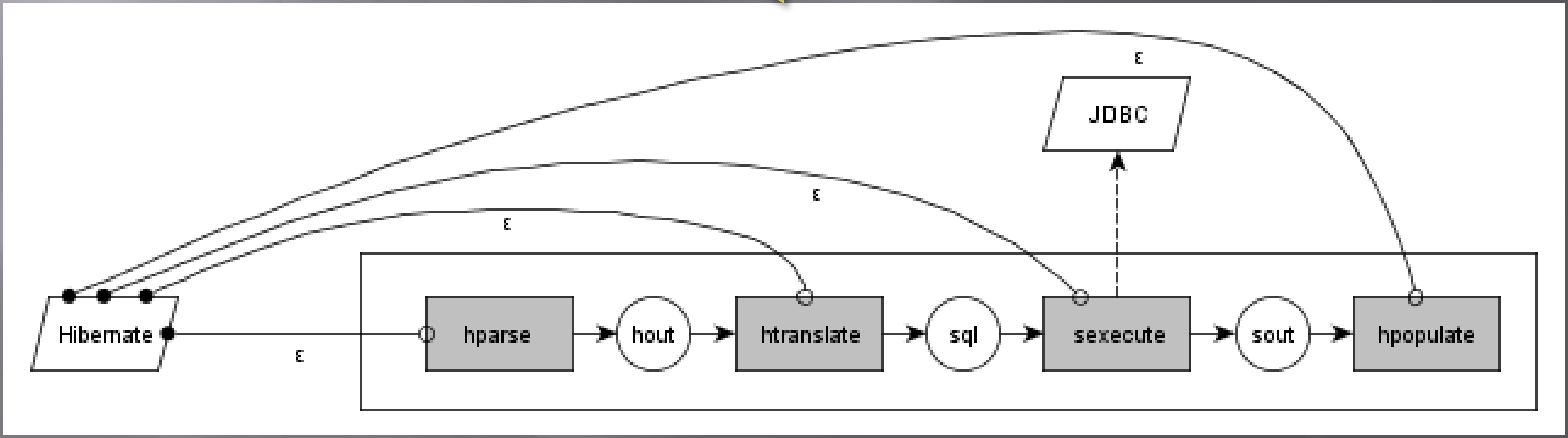


```
session.createQuery(...)
```

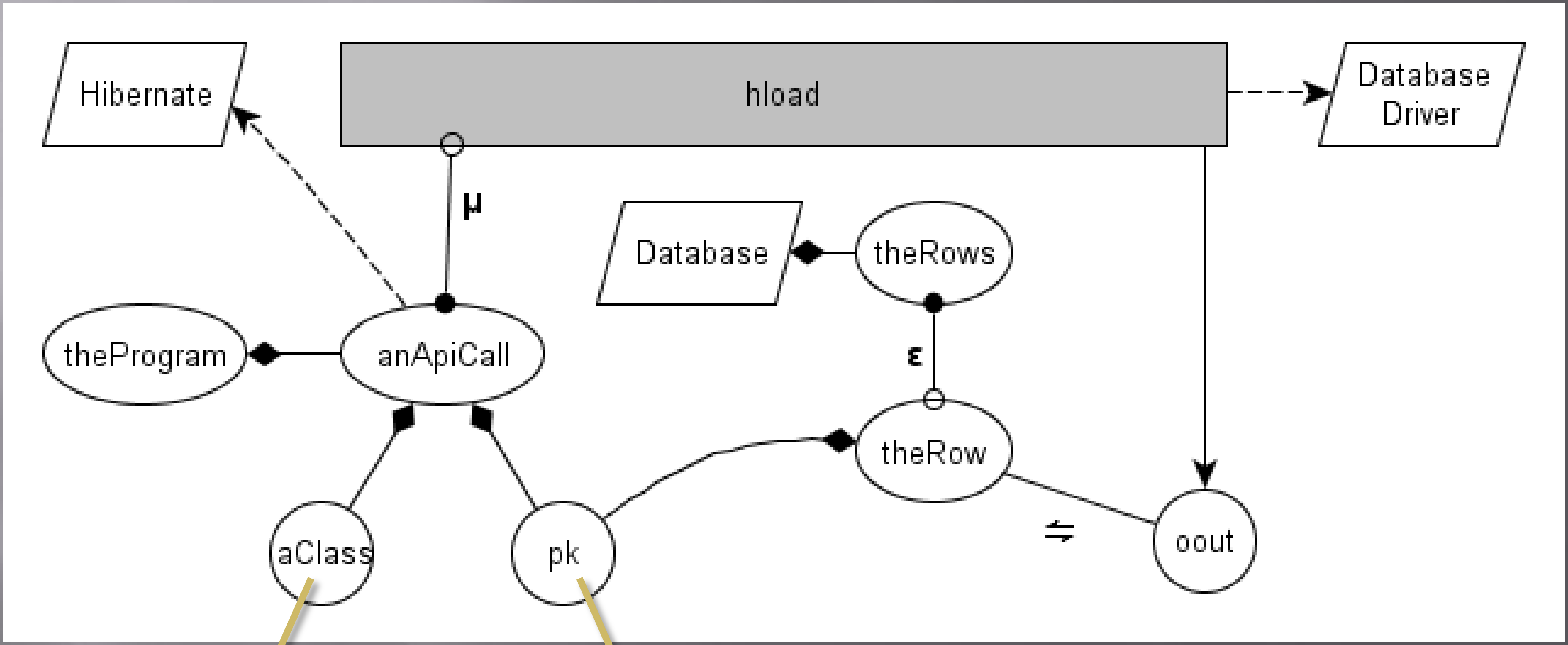
```
public static void main(String[] args) {
  SessionFactory sessionFactory = new
      Configuration()
       .configure().buildSessionFactory();
  Session session = sessionFactory
       .getCurrentSession();
  session.beginTransaction();
  List<?> result = this.session.createQuery(
            "from Company where name =
      'meganalysis'"
       ).list();
  Company company = (Company) result.getNext();
}
```

Mega model – Hibernate Internals





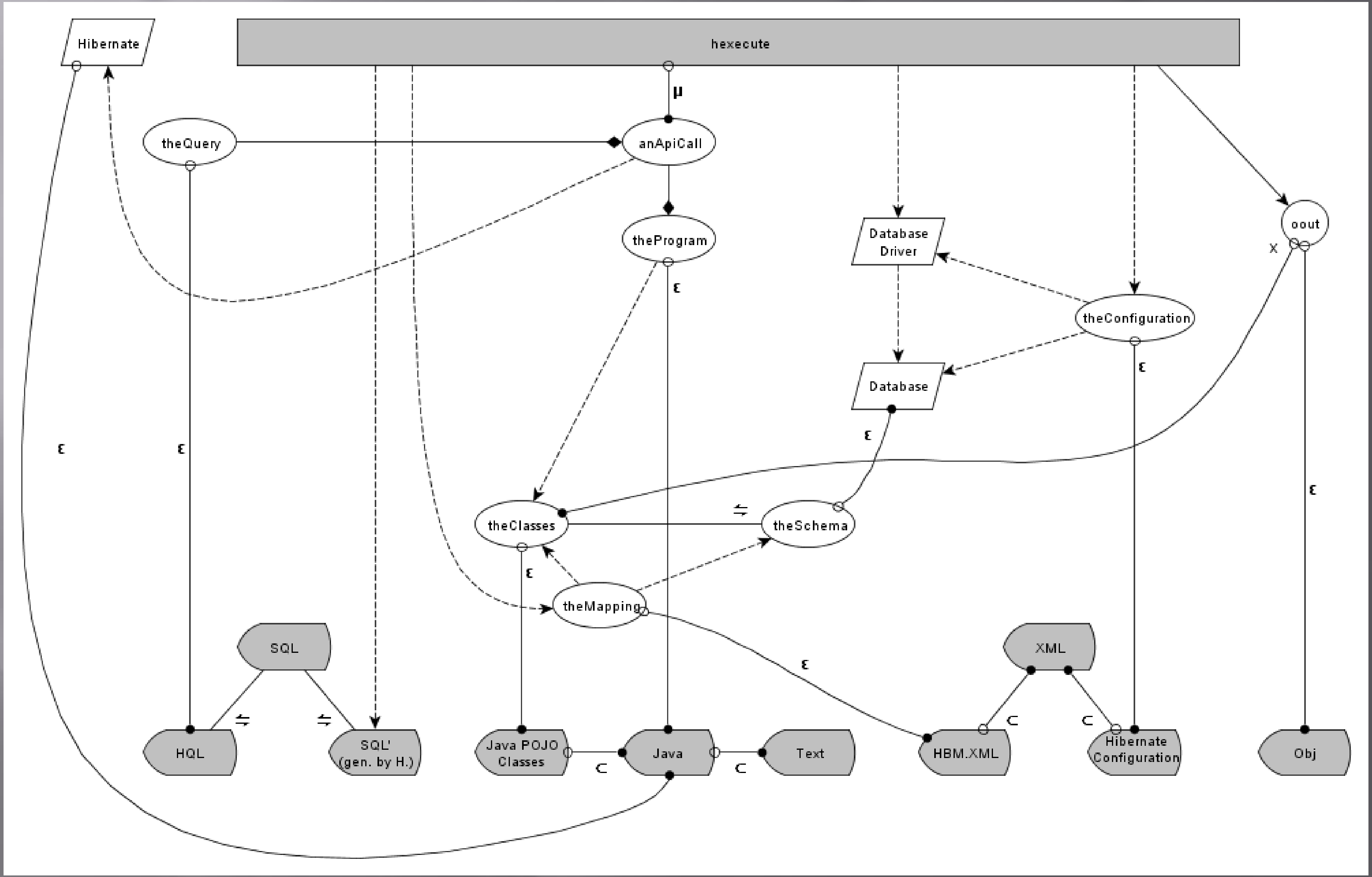
Mega model – Loading by Primary Key



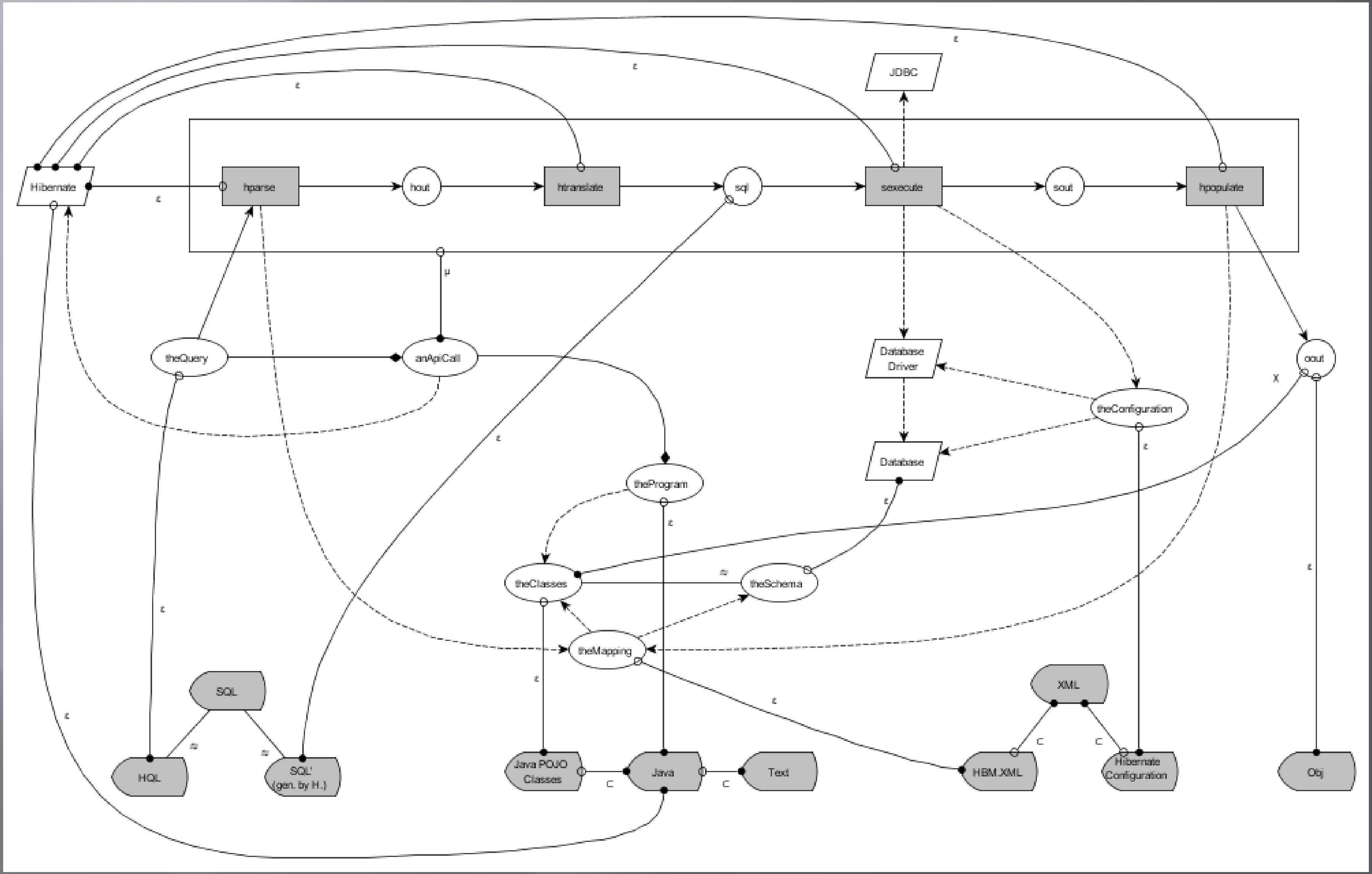
```
Company.class
```

```
42
```

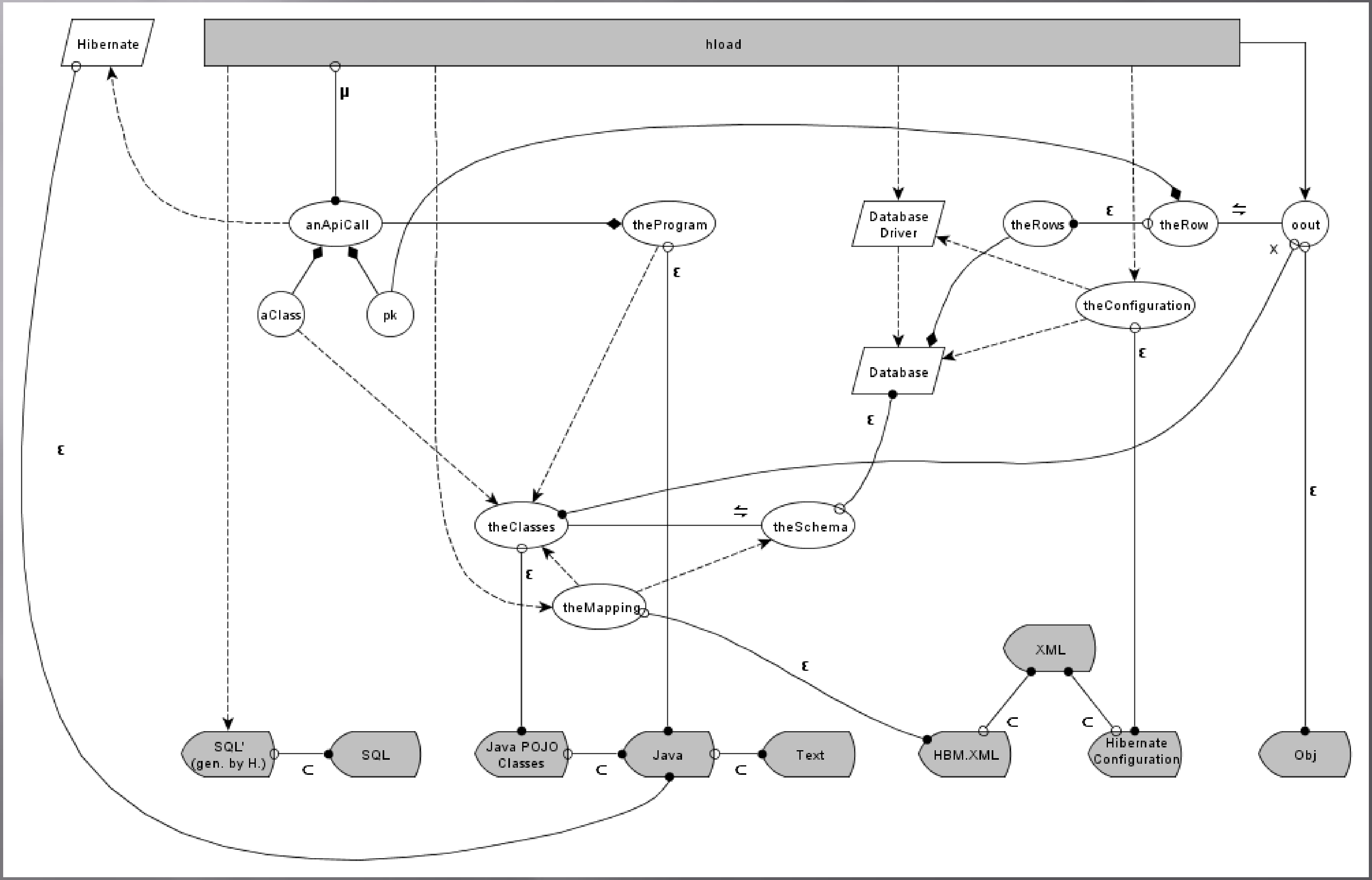
Mega model – Putting all together



Mega model – Putting all together

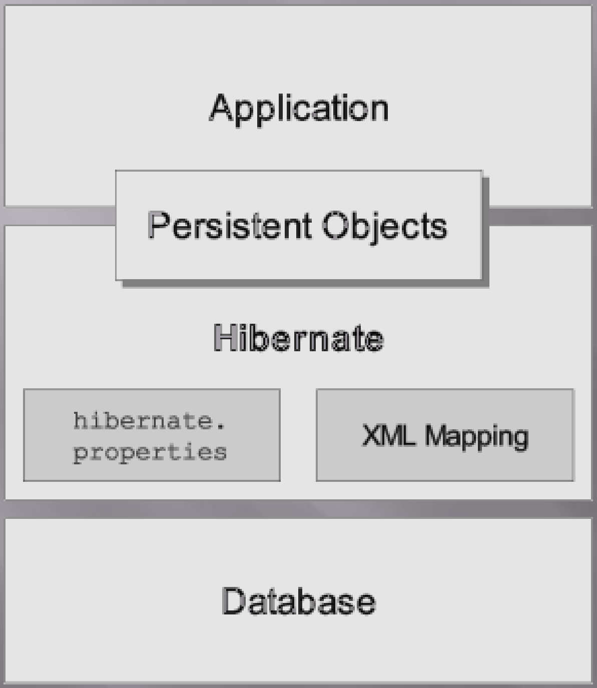


Mega model – Putting all together

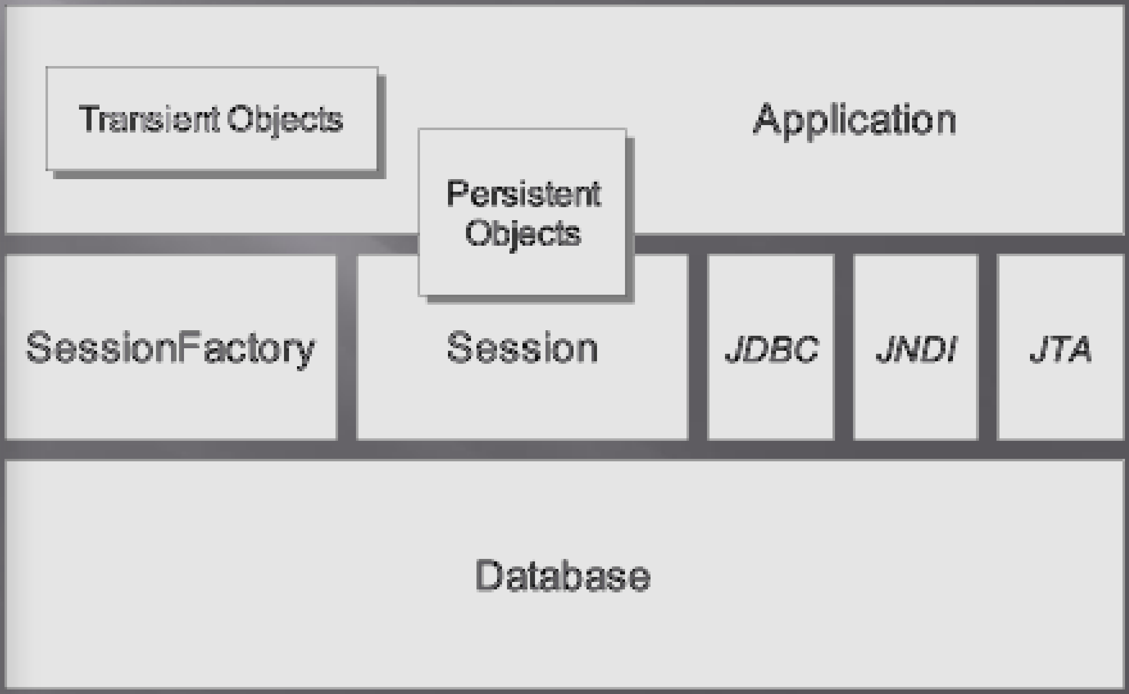


How is the Hibernate architecture really looking?

modeled



in real (minimum!)



How much do the mm cover?

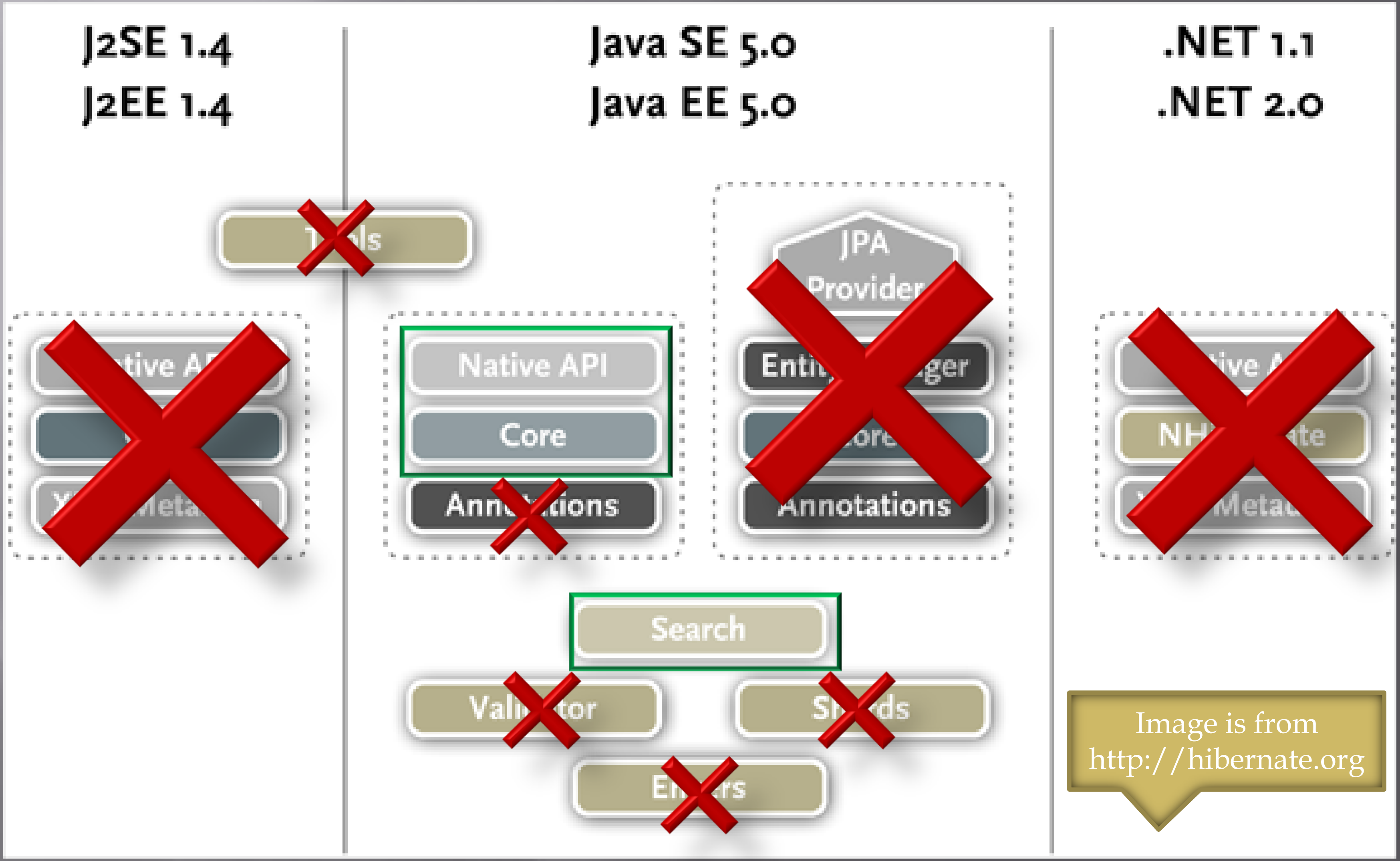


Image is from http://hibernate.org

Sources

- http://101companies.org/index.php/101implementation:hibernate
- http://de.wikipedia.org/wiki/Hibernate (Framework)
- http://www.hibernate.org/
- http://docs.jboss.org/hibernate/core/3.6/reference/en-US/html/architecture.html#architecture-overview
- Further sources: see the paper belonging to this presentation

THANK YOU FOR YOUR KIND ATTENTION!