



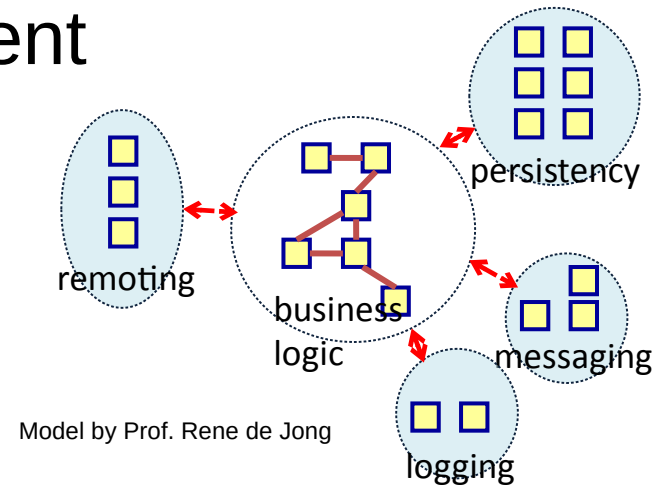
CS544 EA

Hibernate

Introduction to Hibernate

DDD with RDB

- **Domain Driven Development** (DDD) is a big pattern in Enterprise Application development
 - Requires us to design an OO Model
 - Will likely be persisted in an RDB

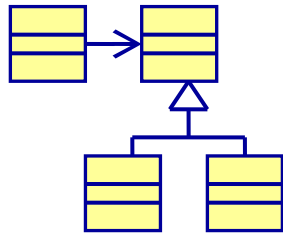


- Although NoSQL databases are on the rise, **Relational Databases** (RDBs) are still dominant

OO / RDB Mismatch

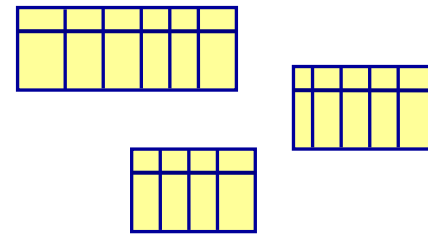
- OO and Relational can both represent data, but
 - They **don't completely match** in several areas

Object Model



Model by Prof. Rene de Jong

Relational Schema



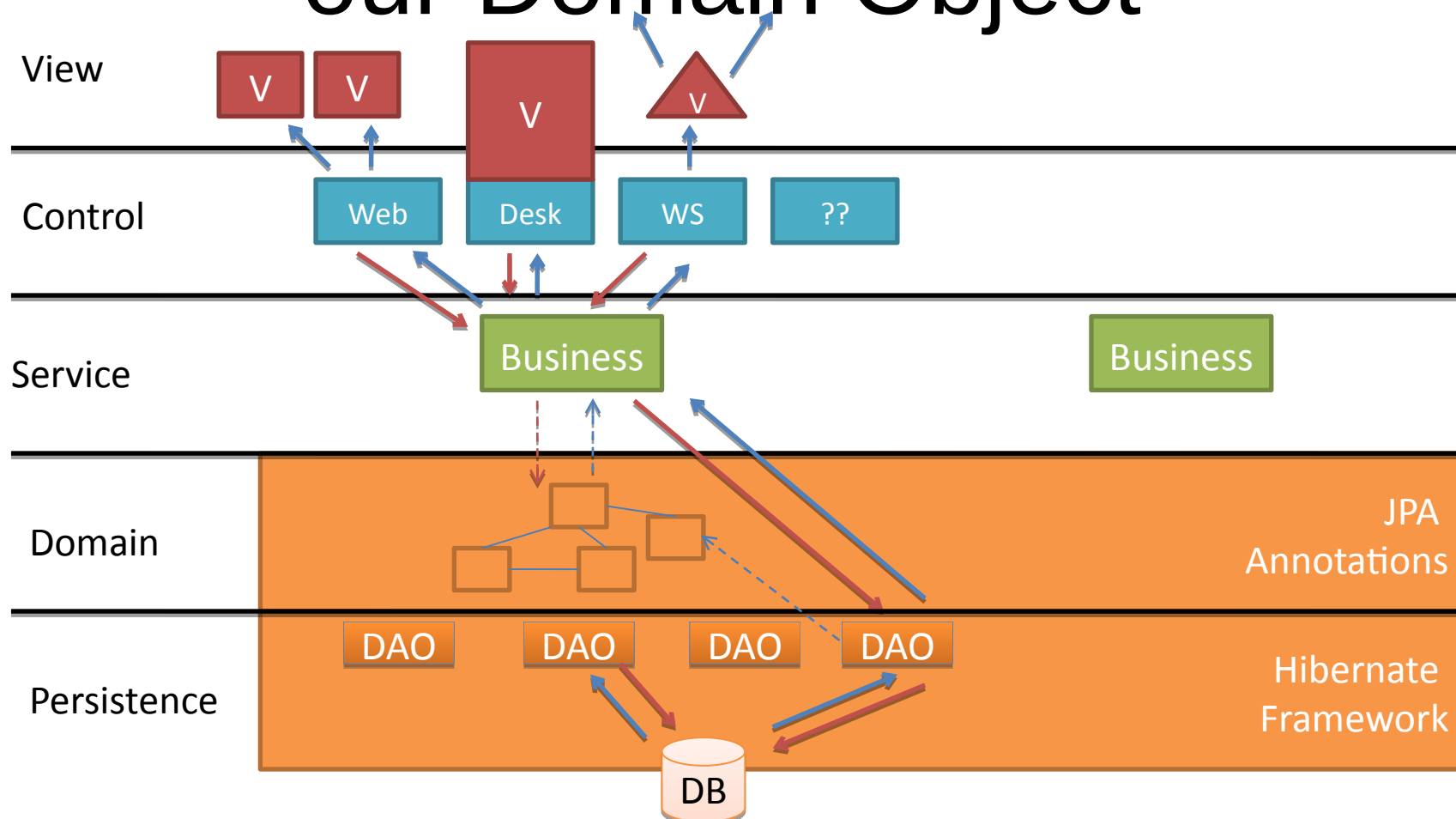
Mismatch Categories

Object Oriented	Relational Database
Objects are instantiations of classes and automatically have (mem address) identity (object1 == object2)	In the relational model the table name and a primary key is used to identity a row in a table
Objects have associations (one-to-one, many-to-one, ...)	Relational model has foreign keys and link tables
OO has inheritance	Relational model has no such thing
Data can be accessed by following object associations	Data can be accessed using queries and joins

Mapping

- With mapping we provide additional (meta-data) information about how to bridge these gaps
 - Usually provided with **annotations** or **XML**
- Allowing us to work with an DDD OO model
 - While hiding the reality of the RDB as much as possible
 - Unfortunately this is a leaky abstraction

Hibernate Manages and Persists our Domain Object



OO Queries + Lazy Loading Objects

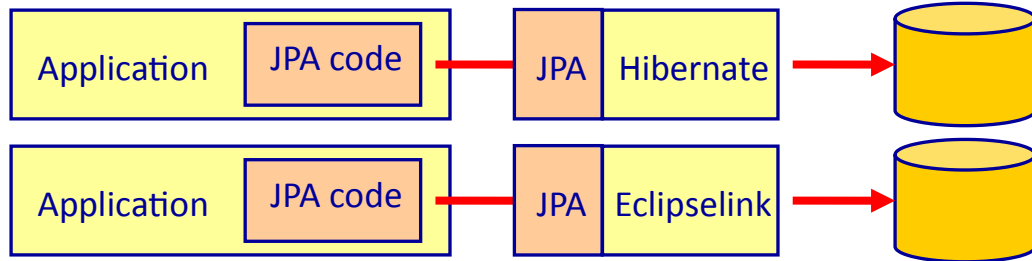
- When you work with Hibernate you write OO queries (JPQL is like SQL but OO)
 - Once you've retrieve an object, you can **follow references** on it to get to other objects
 - Hibernate automatically fetches any missing objects (hiding the fact that they're not in memory yet)
 - By default it only loads related objects as needed (lazy loading)
 - If you change a value, Hibernate **automatically updates** the database for you.

Java Persistence API

- The **Java Persistence API** (JPA) is a standard for **Object Relational Mapping** (ORM)
 - Basically Hibernate's API
 - With some fixes (old API had some problems)

- If you P2I to JPA you can:

- Switch to any Java ORM provider
- Hibernate is the most popular, there are several others



Model by Prof. Rene de Jong