C/S体系结构的**通信模式**：

1) 带回复的同步消息通信(synchronous message communication with reply)

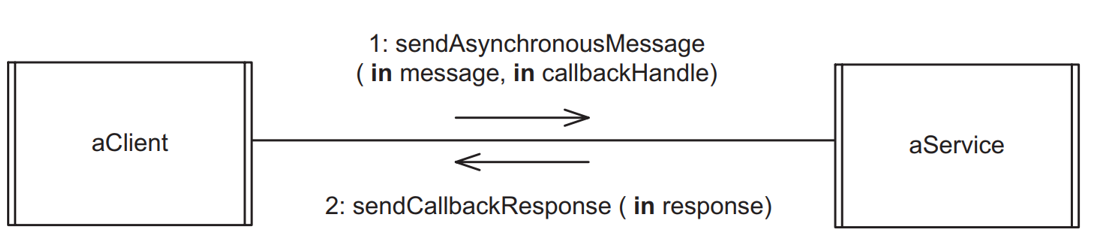
2) 不带回复的同步消息通信(synchronous message communication without reply)

3) 异步消息通信(asynchronous message communication)

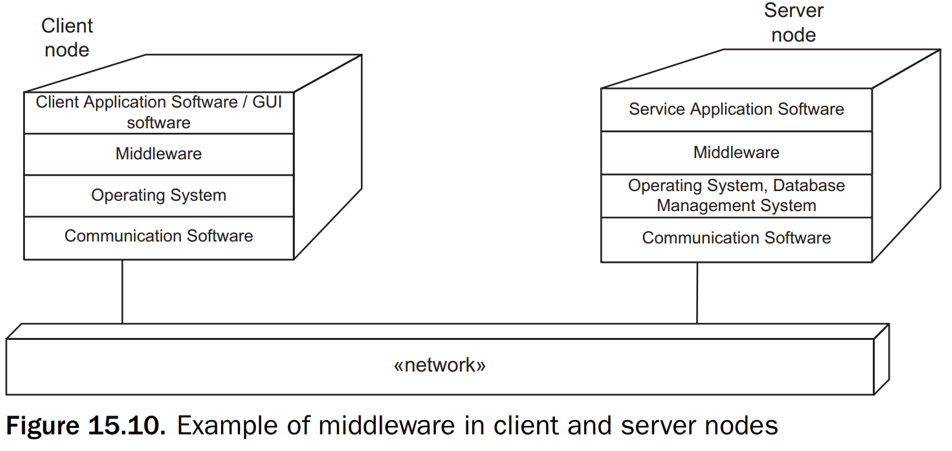
4) 带回调的异步消息通信(asynchronous message communication with callback)

5) 代理者模式(broker pattern)

6) 群组通信模式(group communication pattern)

异步通信中的回调（变种：服务将回调处理转发给另一个构件，并将答复委派给该构件）

**中间件**(middleware)：在软件中处于各种操作系统之上的一层，为各种分布式应用提供一个统一的运行平台

* 远程过程调用(remote process call, RPC)
* Java远程方法调用(remote method invocation, RMI)
* 通用对象请求代理体系结构(CORBA)

**包装器类**(wrapper class)：通过一种粗糙的机制实现与遗留代码的接口，处理客户端对遗留应用请求的服务器类。

若数据直接由实体类管理，则采用数据抽象类处理，很有可能在客户端

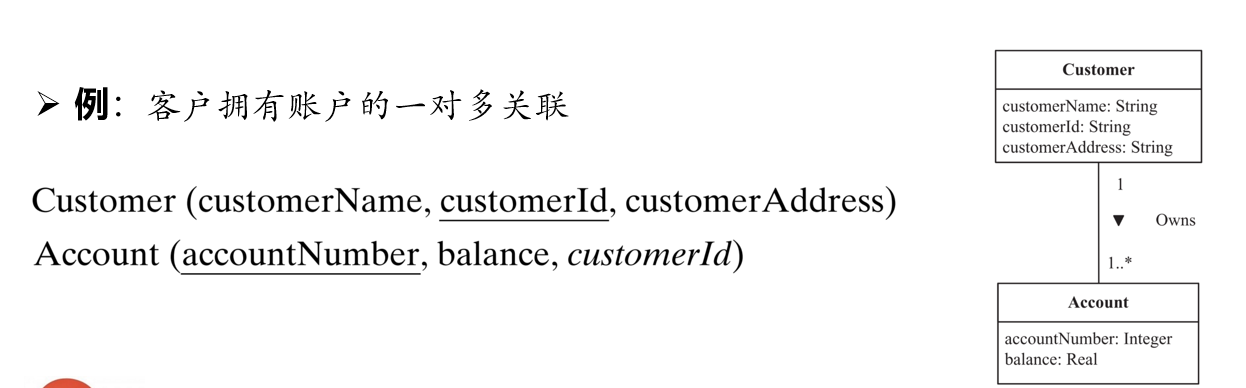
若数据存储于数据库中，则采用数据库包装器类处理，通常在服务器

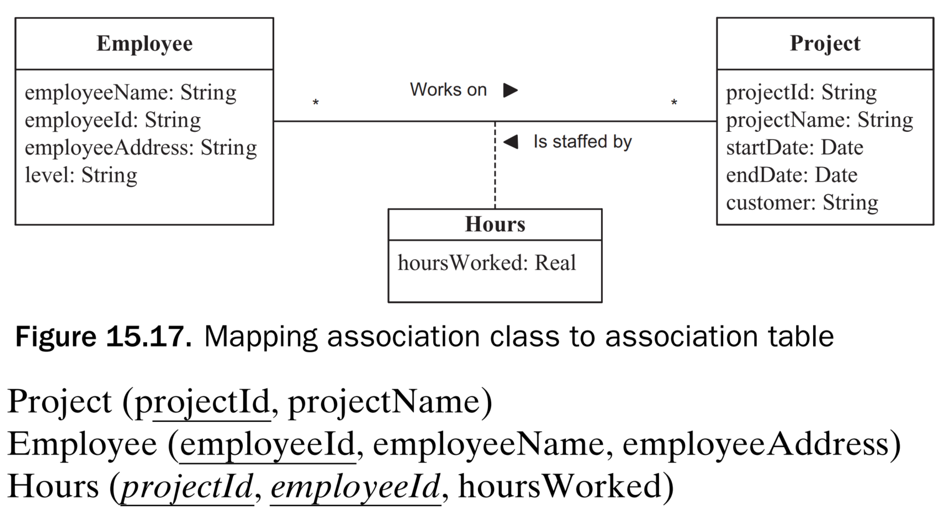
从静态模型到**关系数据库**设计：根据存放数据的实体类设计对应的数据库的关系表

关系数据库(relational database)：由一系列名称唯一的关系表组成。

格式：表名(主键属性, *外键属性*, 其他属性…)

主键(primary key)：关系表中最关键的属性

外键(foreign key)：另一张关系表的主键。形成关系表之间的关联。

关联表(association table)：存放关联类数据的关系表，可以复合外键作为主键。

整体类和部分类生成的关系表中，一对一关系可用外键作主键，一对多可用复合主键。

父类和子类生成的关系表中，可用共享主键。