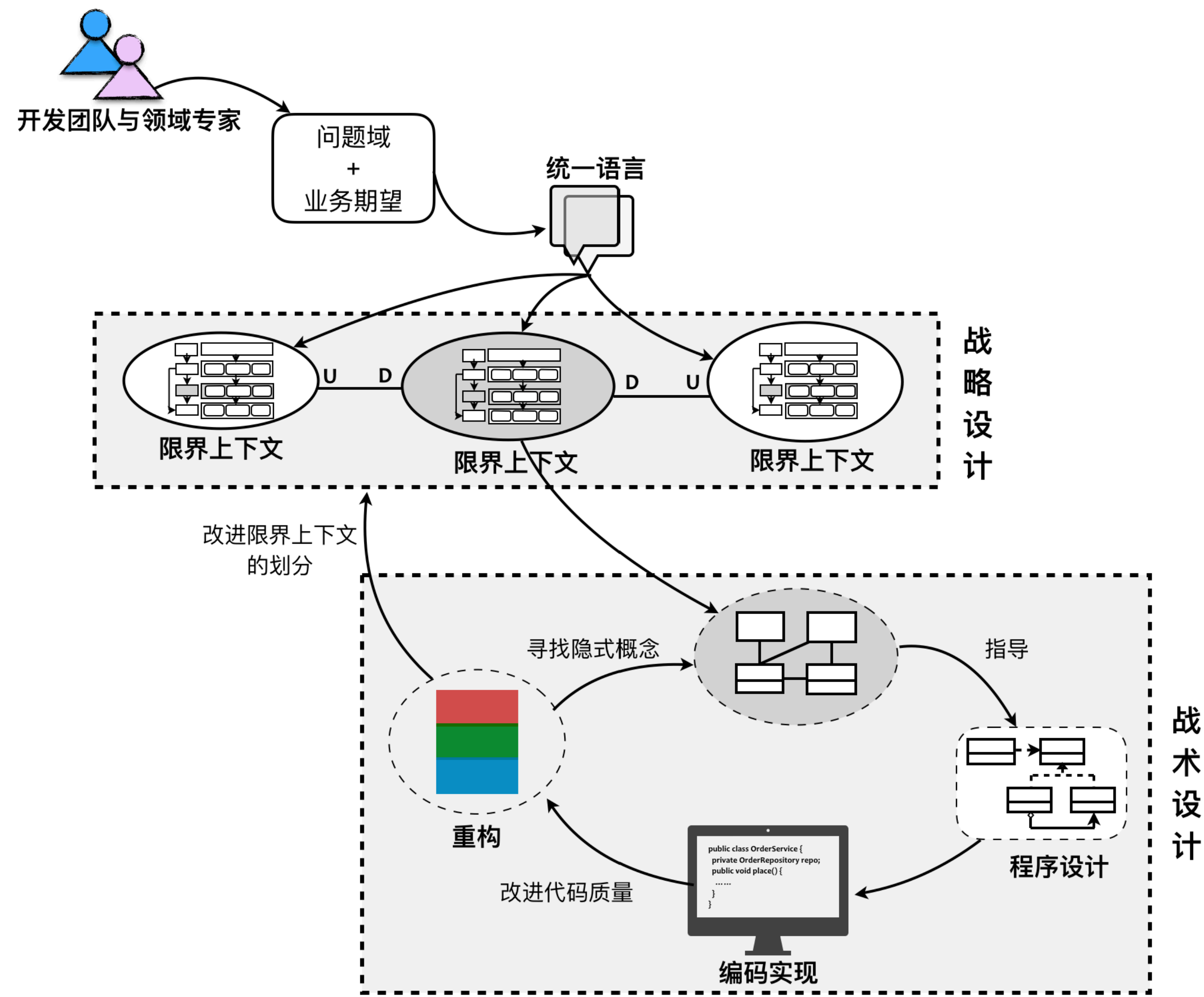
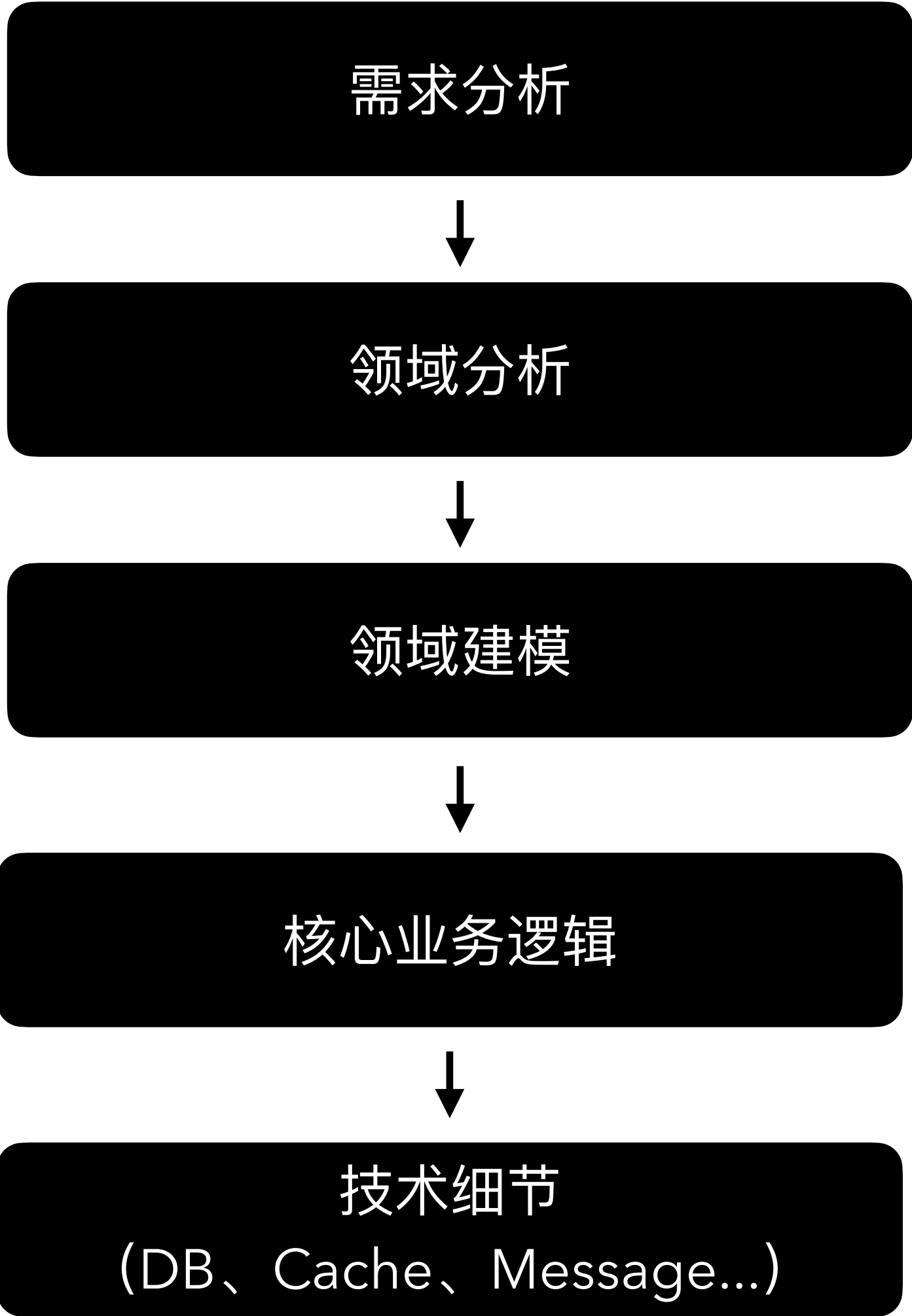


DDD-领域驱动设计

CODINGAPI

Lorne 2020-05-16

什么是领域驱动?



什么是数据驱动?

需求分析



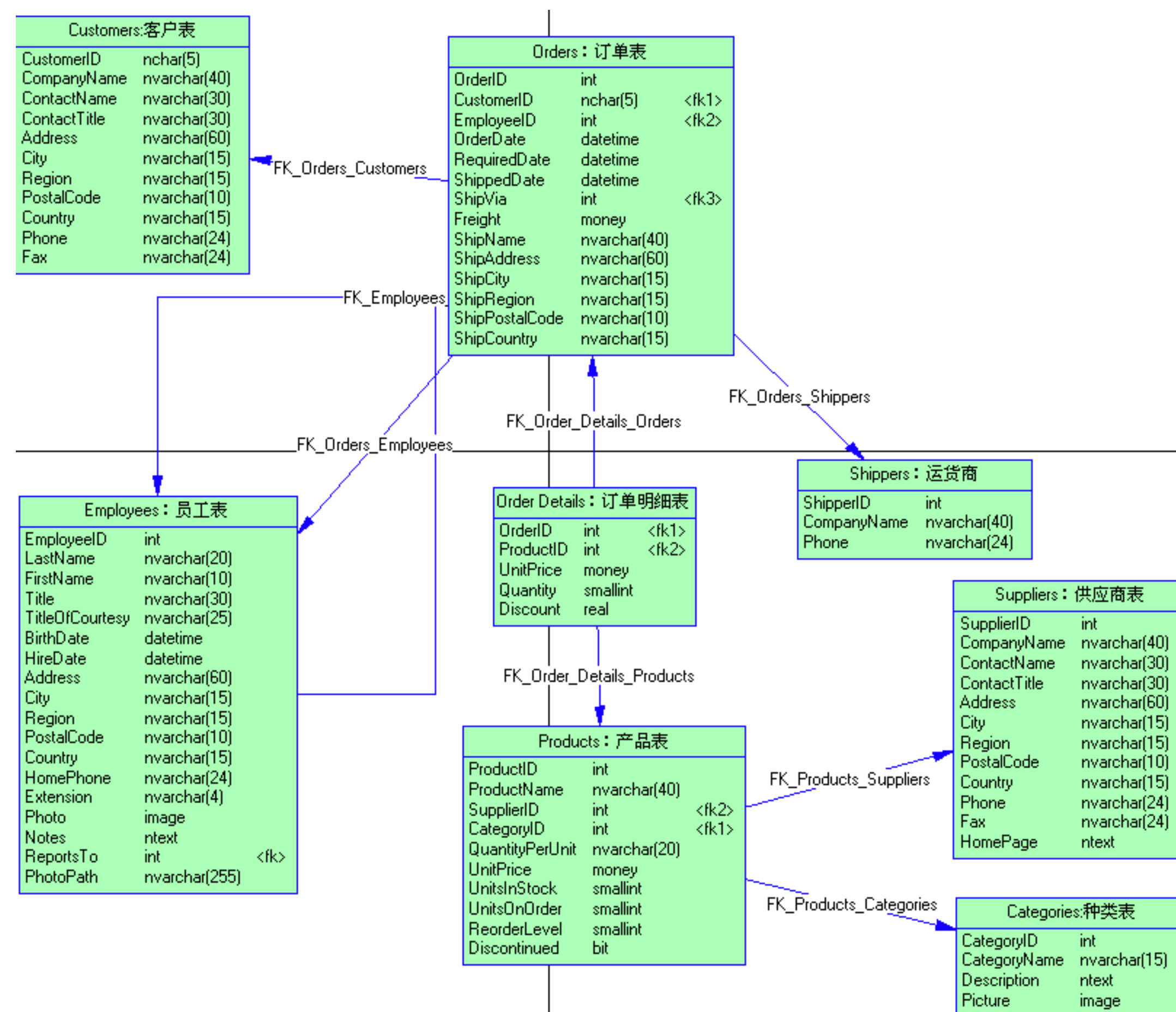
数据建模 (ER图)



建库建表 (写DAO)

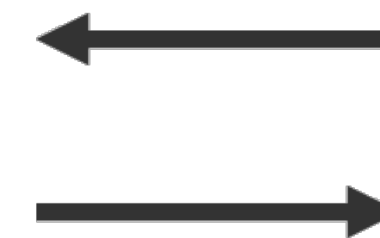


编写业务逻辑



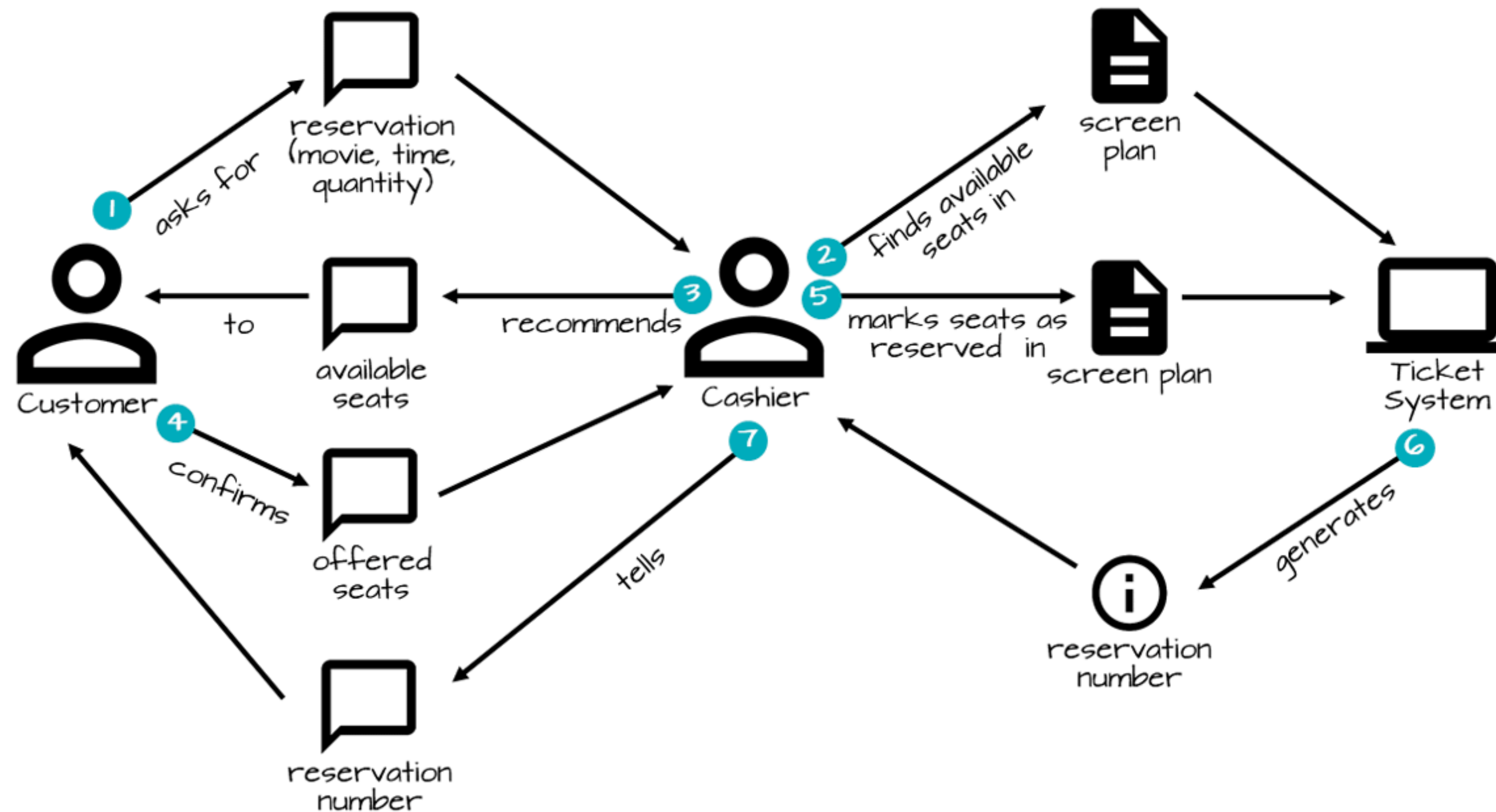
Domain StoryTelling

- Pictograms and Vocabulary
 - Actors
 - WorkObjects
 - Activities
 - Annotations



<https://domainstorytelling.org/#dst-ddd>

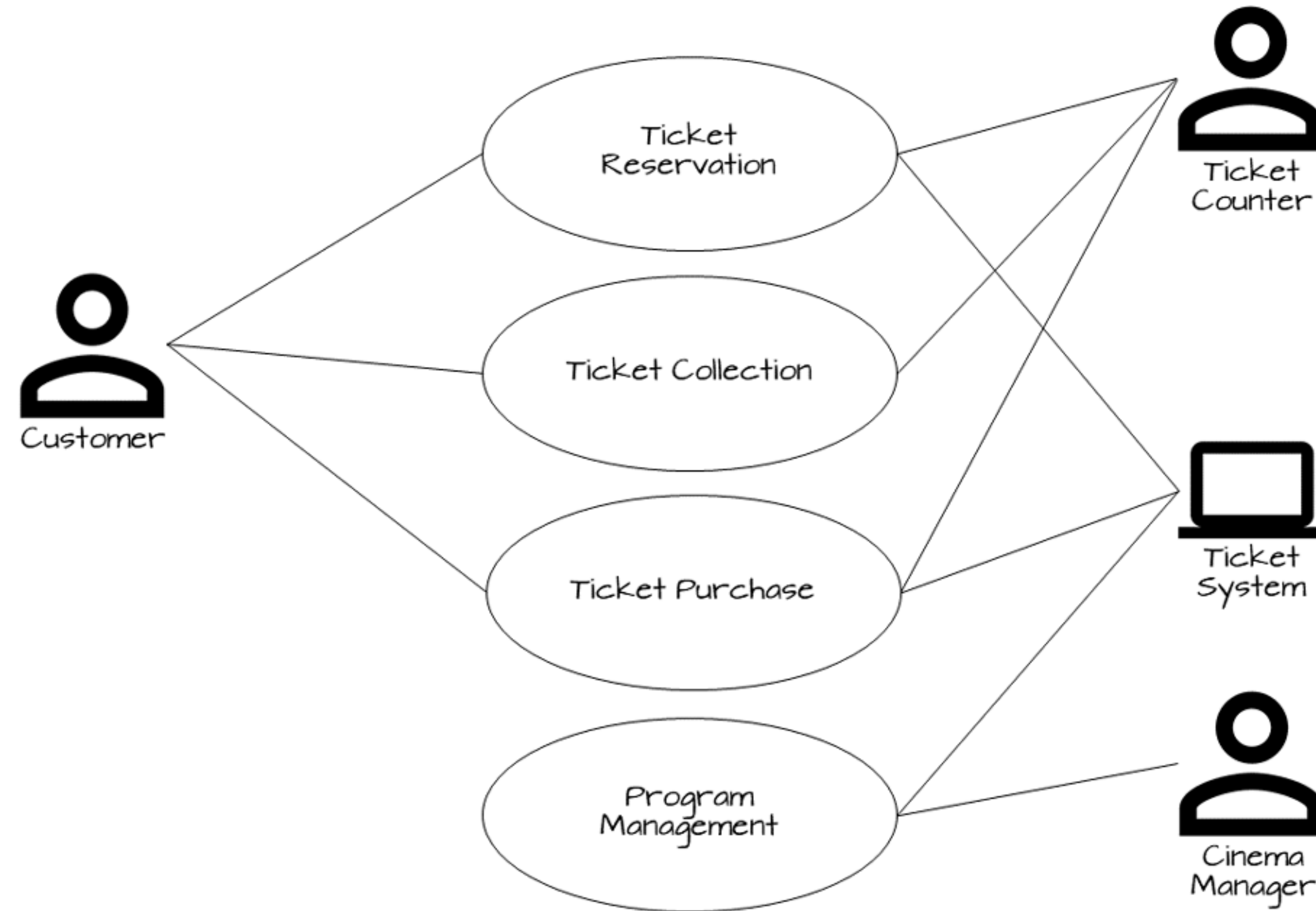
An Domain Story Example



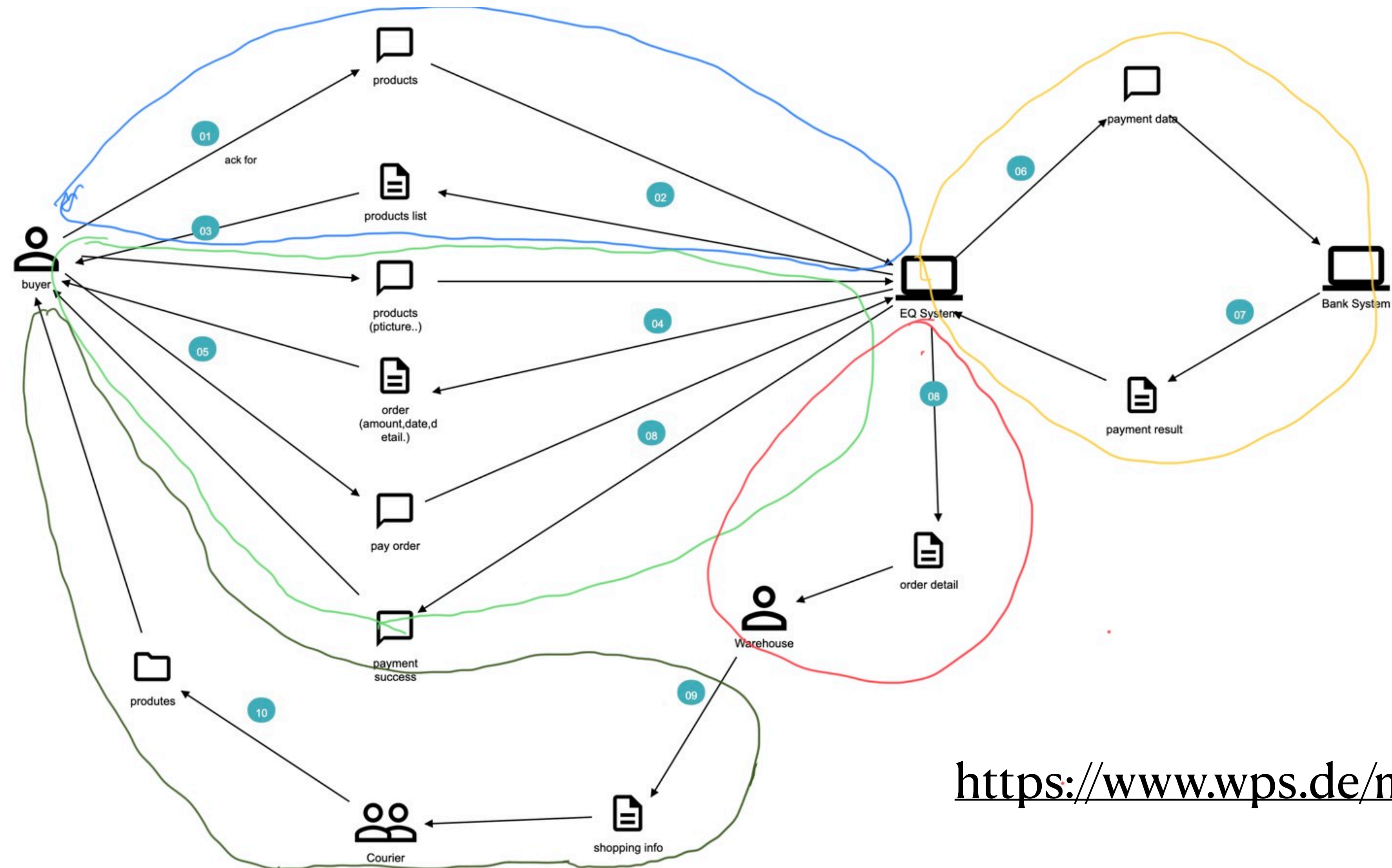
<https://www.wps.de/modeler/>

Use Case Analysis

- Actors
 - Customer
 - Ticket Counter
 - Ticket System
 - Cinema Manager
- Use Case
 - Ticket Reservation
 - Ticket Collection
 - Ticket Purchase
 - Program Management

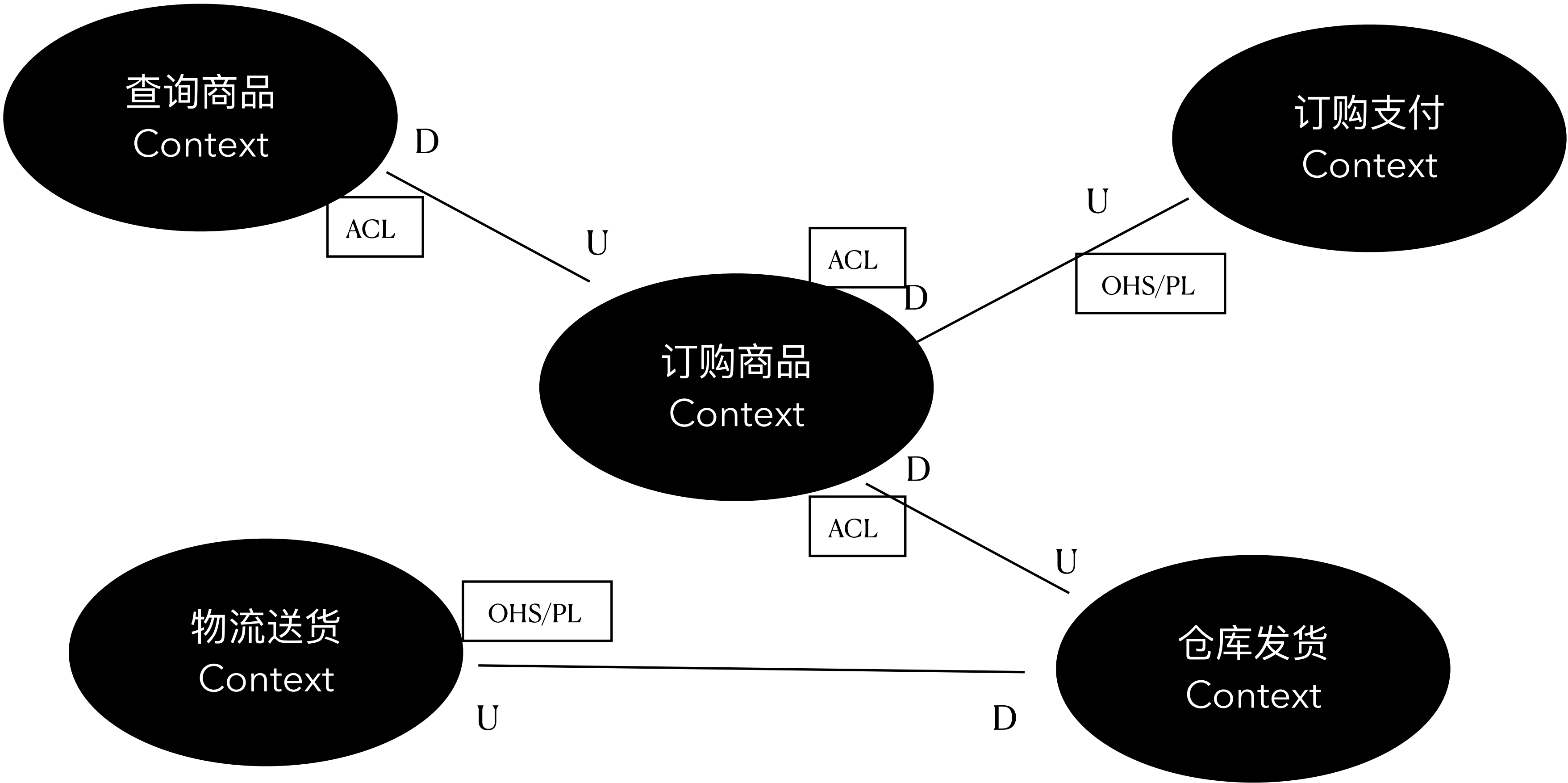


Domain Analysis



<https://www.wps.de/modeler/>

Contexts Map



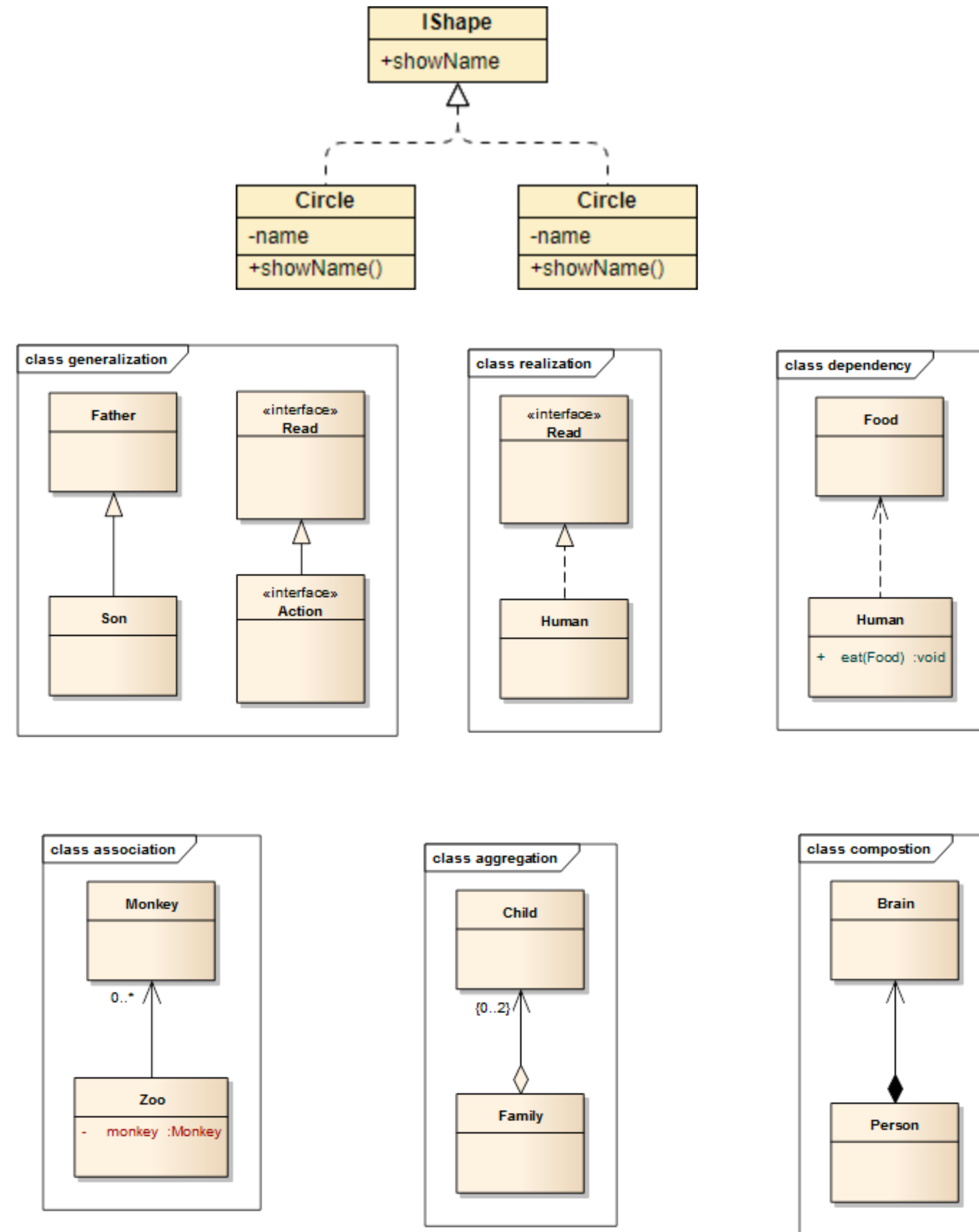
Domain Design

- DDD(Domain Driver Designer)
 - Bounded Context
 - Aggregate
 - Entities
 - Value Object
 - Services
 - DomainEvents
 - Factories
 - Repositories

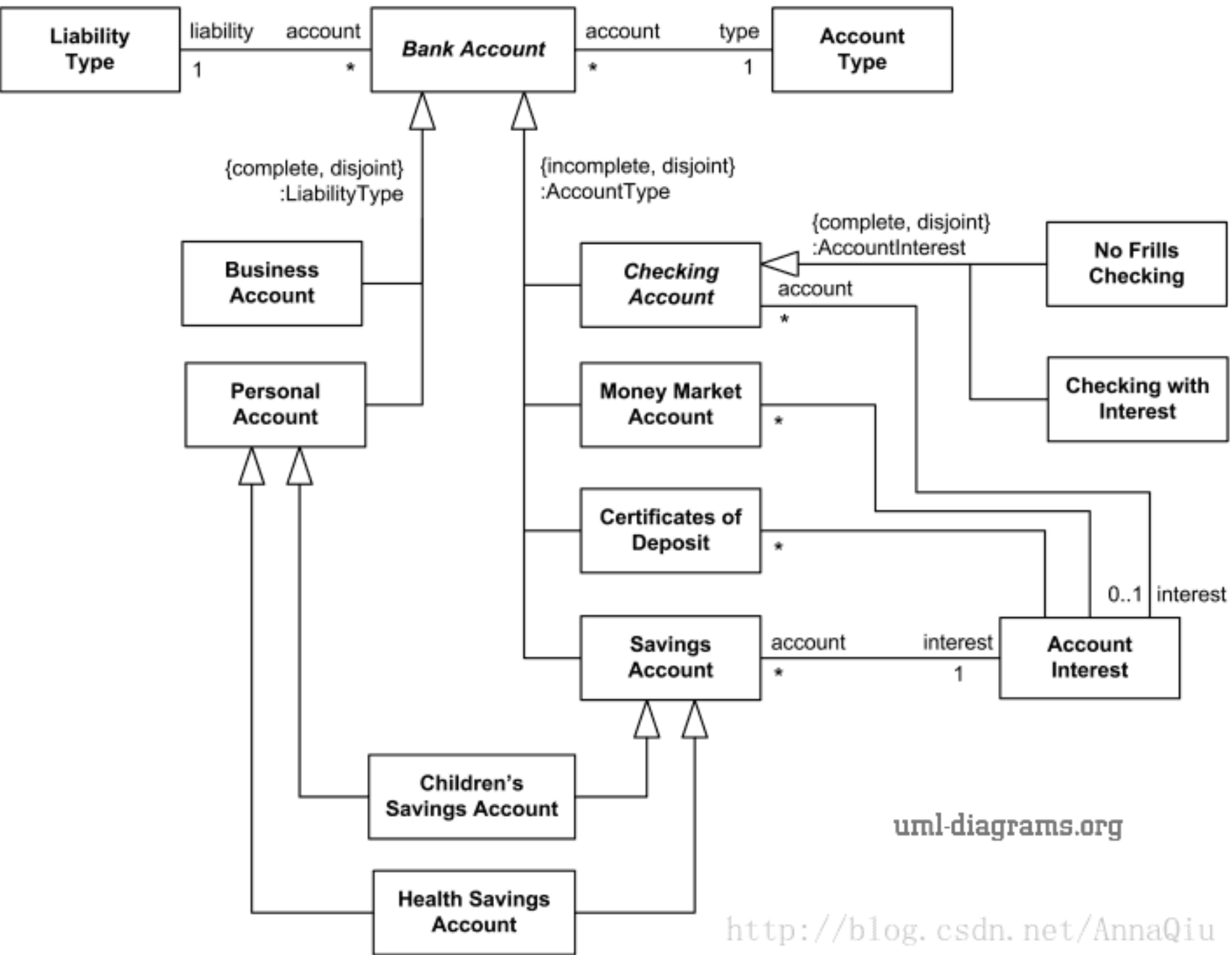
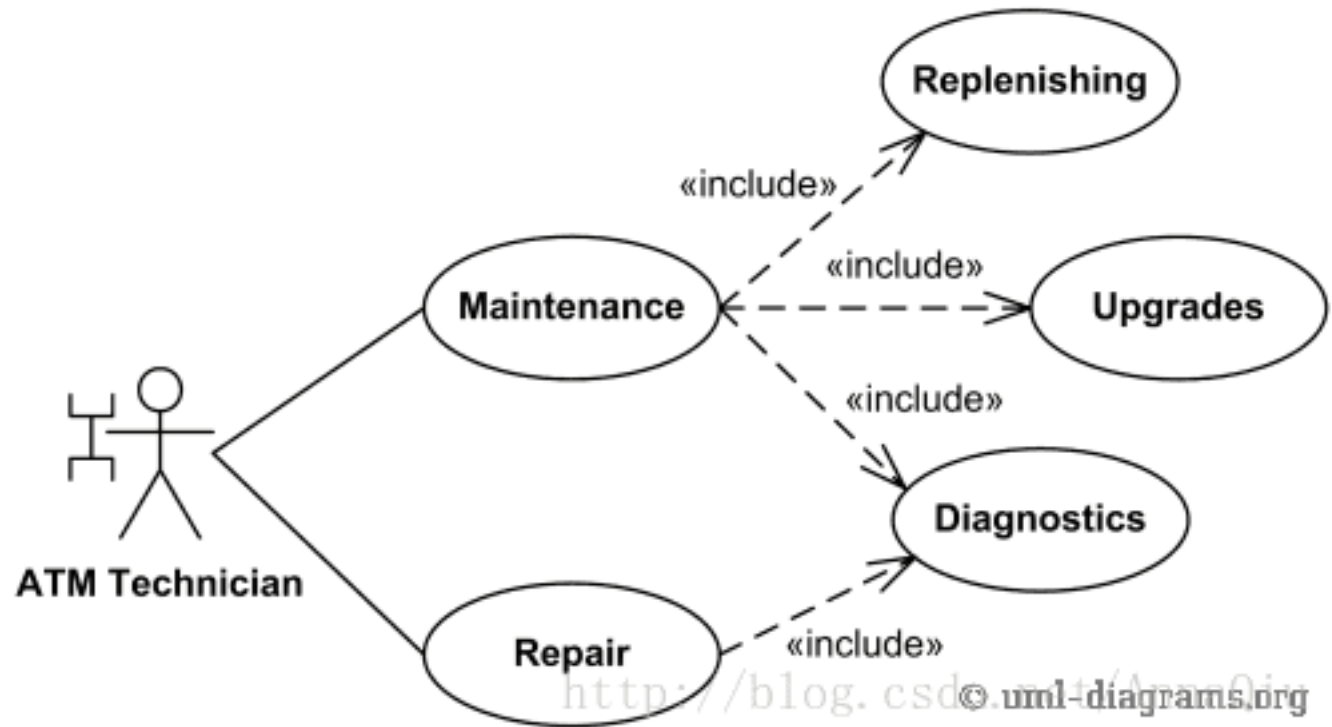
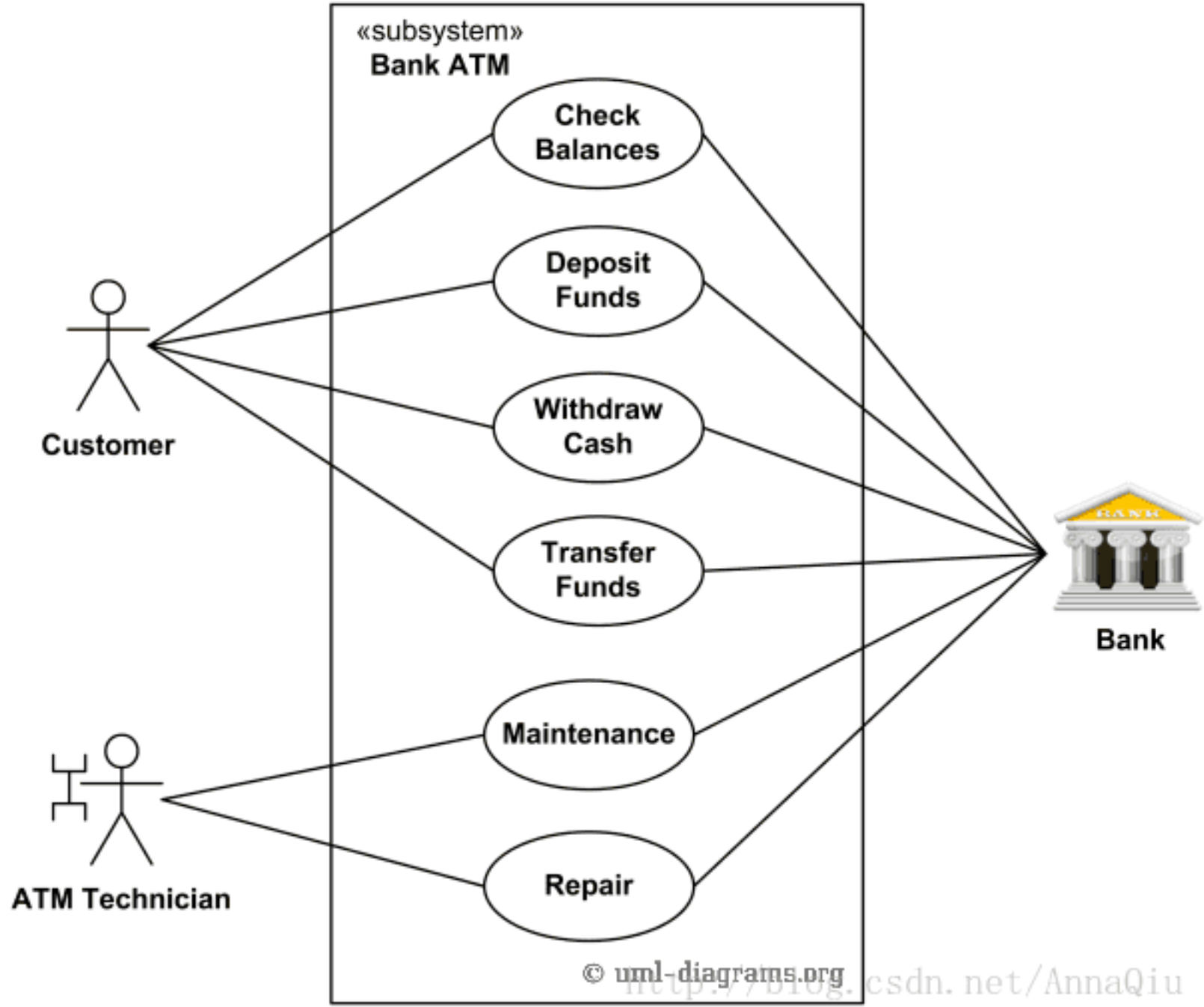


Unified Modeling Language

- 泛化(generalization)
- 实现(realization)
- 依赖(dependency)
- 关联(association)
- 聚合(aggregation)
- 组合(composition)

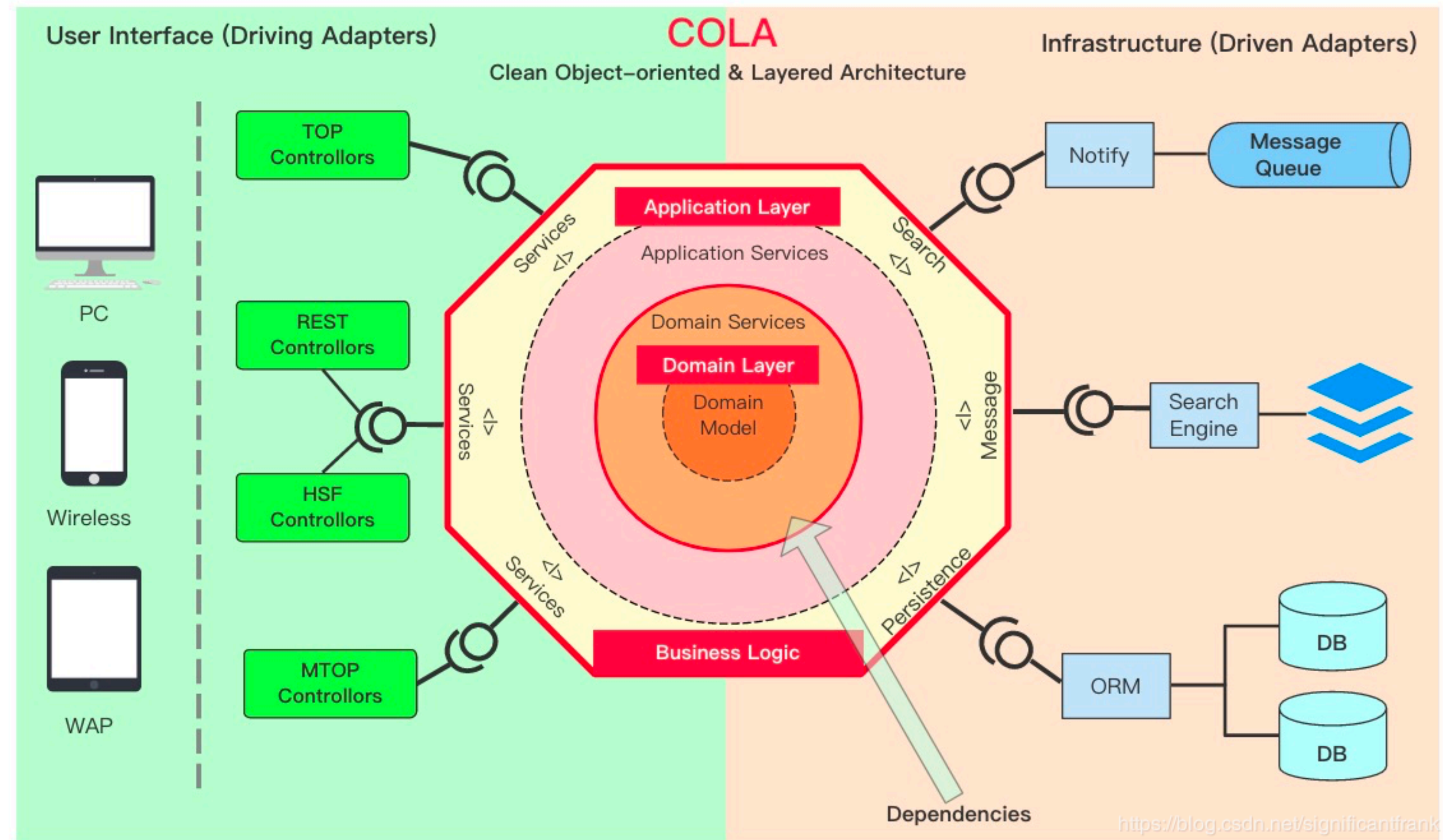


Domain Driver Design



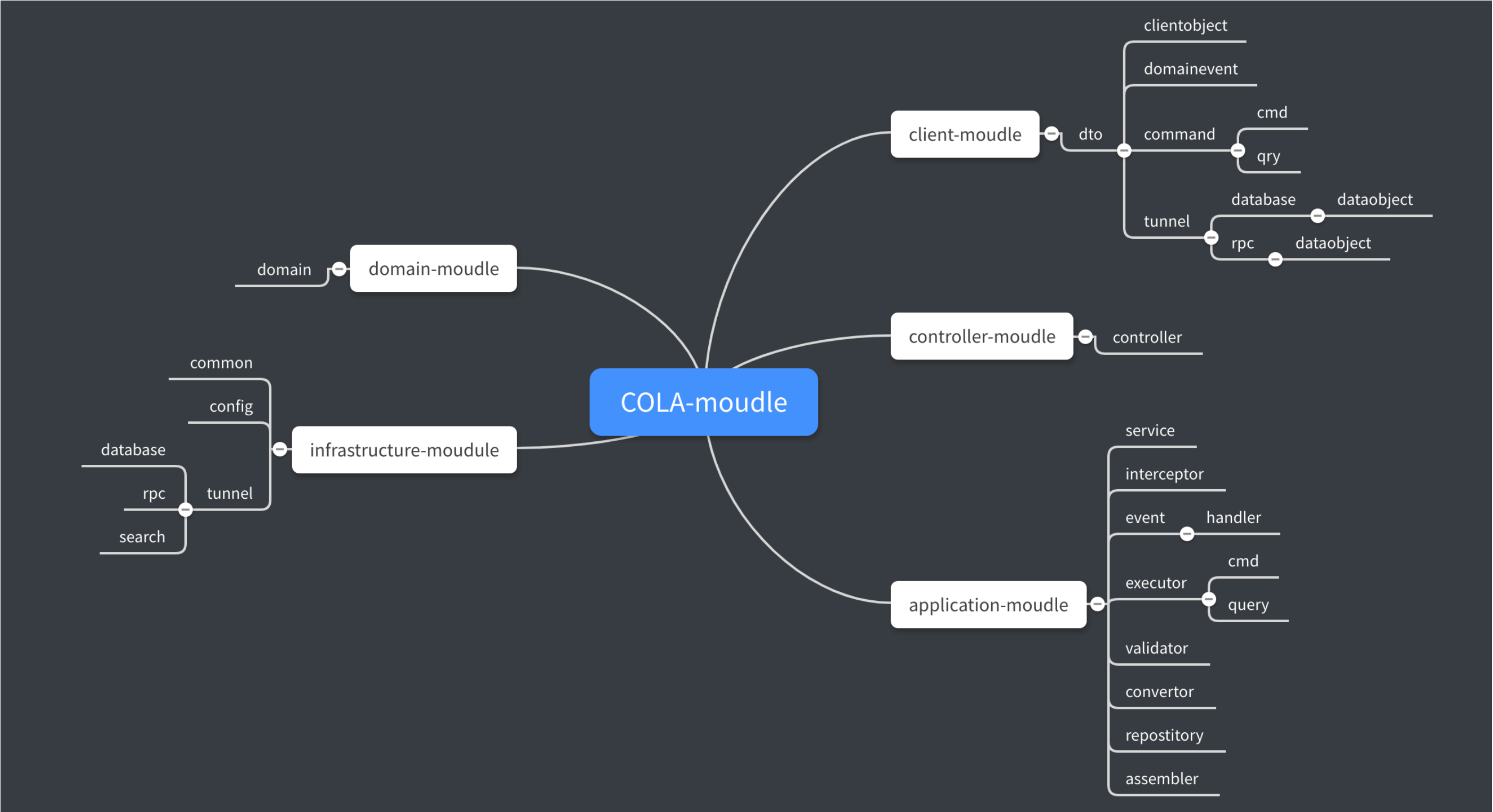
COLA Architecture

- CQRS(Command Query Responsibility Segregation)
- 洋葱架构(Onion Architecture)
- 六边形架构(Alistair Cockburn)



<https://github.com/alibaba/COLA>

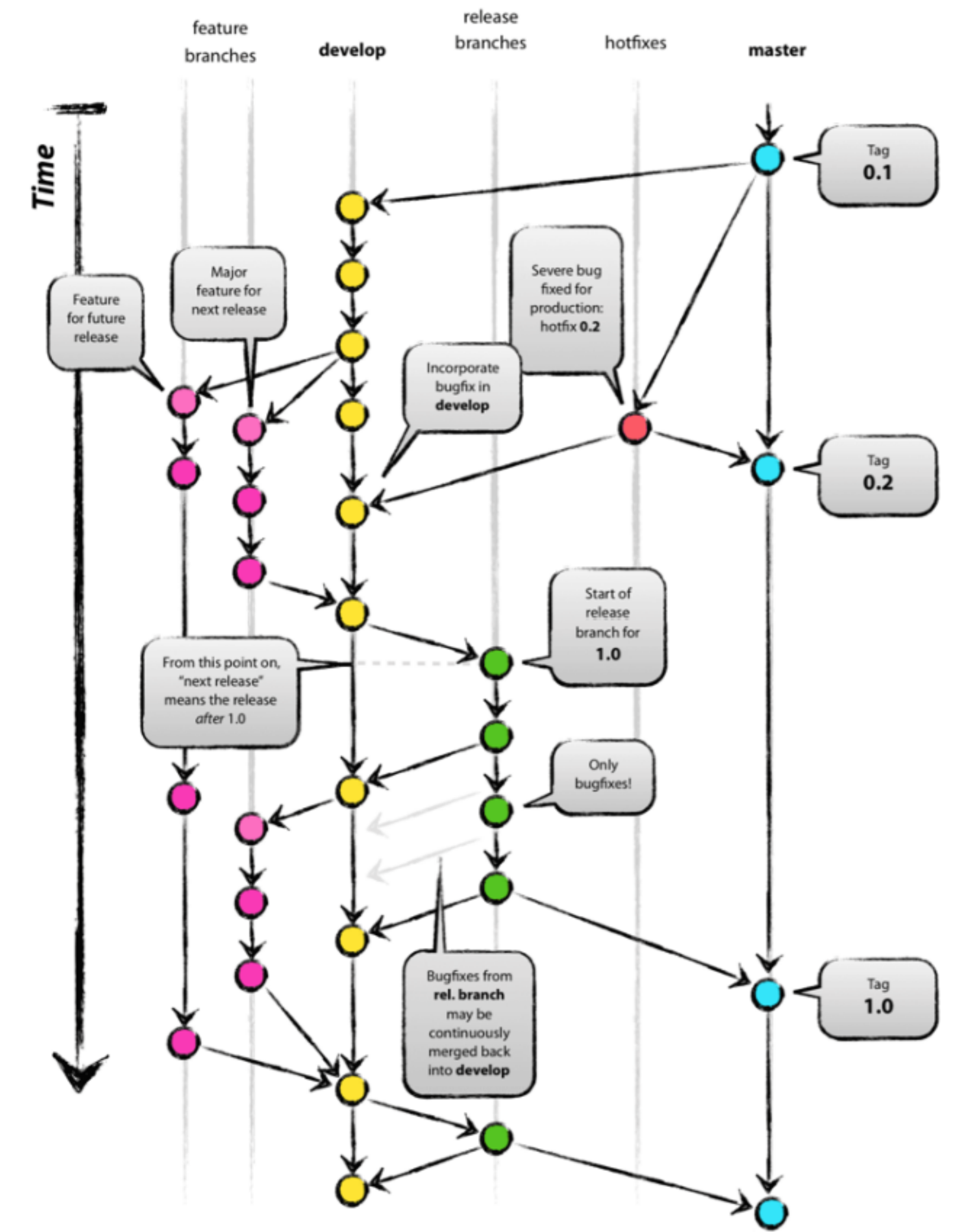
Specification introduction



用途	规范	解释
clientobject	XXXCO	客户视图对象，用于传递数据，等同于 DTO
domainevent	XXXEvent	领域事件消息对象
command	XXXCommand,XX XQuery	CQRS指令对象，区分Command与 Query
tuunel.dataobj ect	XXXDO	数据对象，用于持久化
controller	XXXController	用于接受用户请求数据与相应
service	XXXService XXXServiceImpl	API接口类
interceptor	XXXInterceptor	拦截器，用于处理切面逻辑
event.handler	XXXHandler	事件订阅者，用于处理事件消息
executor	XXXCmd,XXXQry	CQRS命令执行Executor
validator	XXXValidator	校验器，用于校验类
convertor	XXXConvertor	转化器，实现不同层级对象的转换
repostiry	XXXRepostiry	领域资源操作仓库对象，属于应用层
assembler	XXXAssembler	组装器，组装外包服务调用参数
domain.entity	XXXE	领域Entity对象
domain.value-object	XXXV	领域VO对象
domain.service	XXXDomainServic e	领域服务对象
domain.factory	XXXFactory	领域对象工厂
config	XXXConfig	配置类
tunnel	XXXTunnel	数据通道，DAO是常见的通道，也可以是其他通道

GIT Manager

- master
 - 默认分支，也被叫做主分支，保存着稳定版本的代码，每当发布版本以后会设置一个tag用于标记本次版本。
- develop
 - 开发分支，是大家开发维护的直接分支。
- feature branches
 - 功能特征分支，用于分配临时性或功能性任务时指定的分支，通常特征分支会在任务结束以后合并的其他分支并删除该分支。
- release branches
 - 发布分支，该分支保存着的都是发布版本，但是发布版本并不一定是稳定版本，在项目开发流程中这个版本相当于线上测试环境的概念。
- hotfixes
 - 热更新分支，通常是用于解决比较严重的线上bug而建立的分支，在修复完bug以后将再同步给master与develop和release。

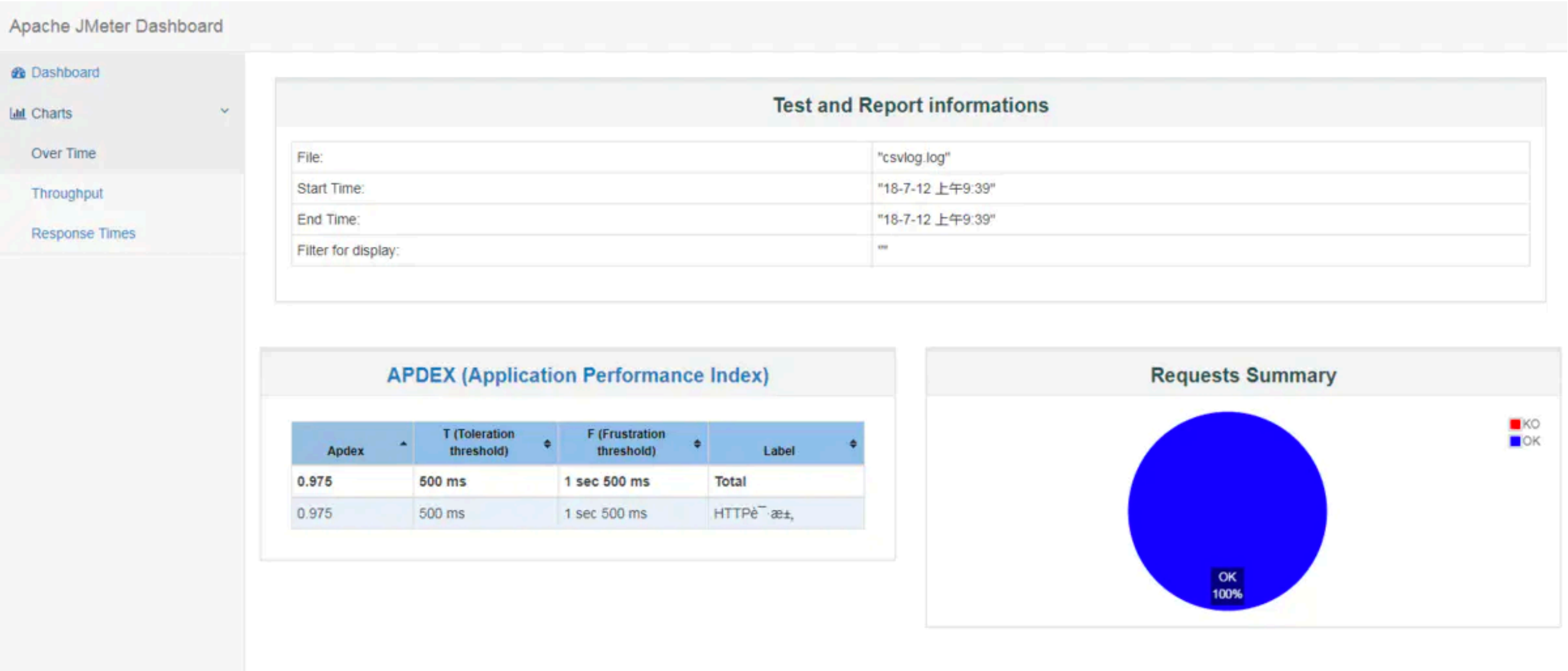


TEST

- Unit Testing
- Performance Testing
- Function Testing

demo

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
com.example.springboot.demo.pojo.vo	<div></div>	0%	<div></div>	0%	33	33	9	9	18	18	2	2
com.example.springboot.demo.pojo.command	<div></div>	13%	<div></div>	0%	27	35	1	7	9	17	0	3
com.example.springboot.demo.repository.db.domain	<div></div>	10%	<div></div>	0%	23	29	2	5	7	13	0	1
com.example.springboot.demo.event.domainevent	<div></div>	8%	<div></div>	0%	17	20	2	3	6	9	0	1
com.example.springboot.demo.service.impl	<div></div>	0%		n/a	5	5	11	11	5	5	2	2
com.example.springboot.demo.extension	<div></div>	0%		n/a	8	8	12	12	8	8	3	3
com.example.springboot.demo.event.handler	<div></div>	0%		n/a	5	5	6	6	5	5	2	2
com.example.springboot.demo.repository.feign	<div></div>	0%		n/a	4	4	4	4	4	4	2	2
com.example.springboot.demo.controller	<div></div>	0%		n/a	2	2	2	2	2	2	1	1
com.example.springboot.demo	<div></div>	0%		n/a	2	2	3	3	2	2	1	1
com.example.springboot.demo.listener	<div></div>	0%		n/a	2	2	3	3	2	2	1	1
com.example.springboot.demo.converter	<div></div>	82%		n/a	1	2	1	5	1	2	0	1
com.example.springboot.demo.domain.refrigerator	<div></div>	100%	<div></div>	100%	0	9	0	33	0	8	0	2
com.example.springboot.demo.executor	<div></div>	100%		n/a	0	7	0	25	0	7	0	3
Total		852 of 1,192		1%	129	163	56	128	69	102	14	25



Thinks



CODINGAPI