



# 架构设计

张彦

修订于 2016-9-20



# 什么是架构设计

- 定义很多，没有一个权威的说法
- 两种理解
  - 针对应用主题，进行系统建模，实现，部署的技术和行为指导
  - 在经验的基础上提炼出普遍有效的技术框架和规范
- 架构设计不是一种方法论，而是一种实践性的工作

# 与系统设计的不同

- 系统设计是架构设计的一部分工作
- 系统设计指导程序员的开发
- 架构设计的范围更广，对需求建模，软件开发，系统测试，运维部署都具有指导性

# 如何进行架构设计

- 指导原则
- 方法论
- 工具



# 架构设计指导原则

- 分而治之
  - 横向切分
  - 纵向切分
- 抓关键
  - 关键需求
  - 关键技术



# 横向切分

- 按软件过程的生命周期进行切分
  - 需求分析
  - 系统设计
  - 测试验收
  - 部署实施
- 按功能进行切分
  - 功能性需求
  - 非功能性需求
    - 性能
    - 扩展性
    - 移植性
    - . . . . .



# 纵向切分

- 建立软件系统的依赖层次
- 避免双向依赖和跨层次依赖
- 高内聚，低耦合



# 架构设计方法论

- 概念视图
- 物理视图
- 网络视图
- 数据视图
- 设计视图
- 运行视图





# 概念视图

- 描述整个软件系统及数据流
- 侧重于主题
- 面向目标用户



# 物理视图

- 描述软件系统物理环境
- 指导部署和运维



# 网络视图

- 描述进程间如何交互
- 私有协议
  - 协议栈
  - 连接方式
  - 编码字符集
  - 同步与异步
  - 超时与异常
  - 接口规范
- 消息中间件
- 数据库



# 数据视图

- 实体关系模型
  - 识别实体
  - 数据库设计范式
- 持久化框架
- 缓存的应用
- Nosql数据库



# 设计视图

- 描述系统实现
- 面向对象的设计
  - 单一职责原则：类仅有一个职责
  - 开闭原则：对扩展开放，对修改关闭
  - 里氏替换原则：能使用父类的地方也可以使用子类
  - 依赖倒置原则：依赖接口而不是具体的类
  - 接口分隔原则：设计多个专门的接口好于设计一个大而全的接口
- 活动图、状态图和序列图
- 设计模式

# 运行视图

- 描述系统的运行状态
  - 交互与顺序
    - 模块之间
    - 进程之间
  - 单线程与多线程
  - 状态变迁与场景恢复
    - 系统崩溃后恢复到崩溃时的场景



# 架构设计工具

- 一图抵万言
- UML统一建模语言
- 架构设计模板



# 架构设计实践







THE END

谢谢观看