

# 软件体系结构概念

---

清华大学软件学院 刘强



# 软件的复杂性

问题：当系统的规模和复杂度不断增大的时候，构造整个系统的关键是什么？



# 软件的复杂性

---

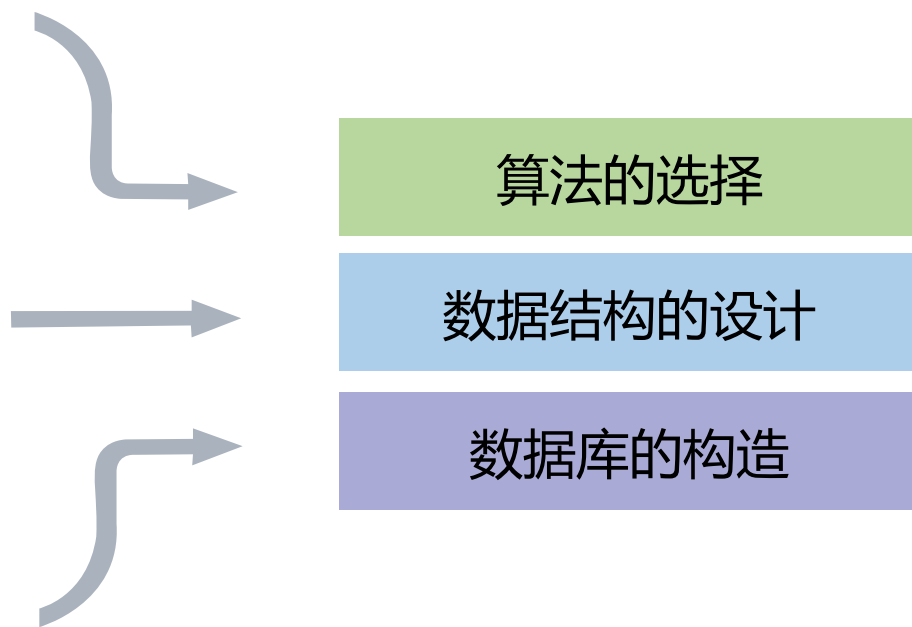
举例：要求编写一个程序，该程序读入一个文本文件，统计在该文本文件中每个英文单词出现的频率，并输出单词频率最高的100个单词。

构造上述软件的重点是设计哪些部分



# 软件的复杂性

举例：要求编写一个程序，该程序读入一个文本文件，统计在该文本文件中每个英文单词出现的频率，并输出单词频率最高的100个单词。



# 软件的复杂性

---



举例：Web信息检索是对发布在Web信息资源进行搜集、整理和组织，形成一个信息资源索引库，并通过检索界面将最符合用户要求的网站或网页提供给用户。



# 软件的复杂性

---

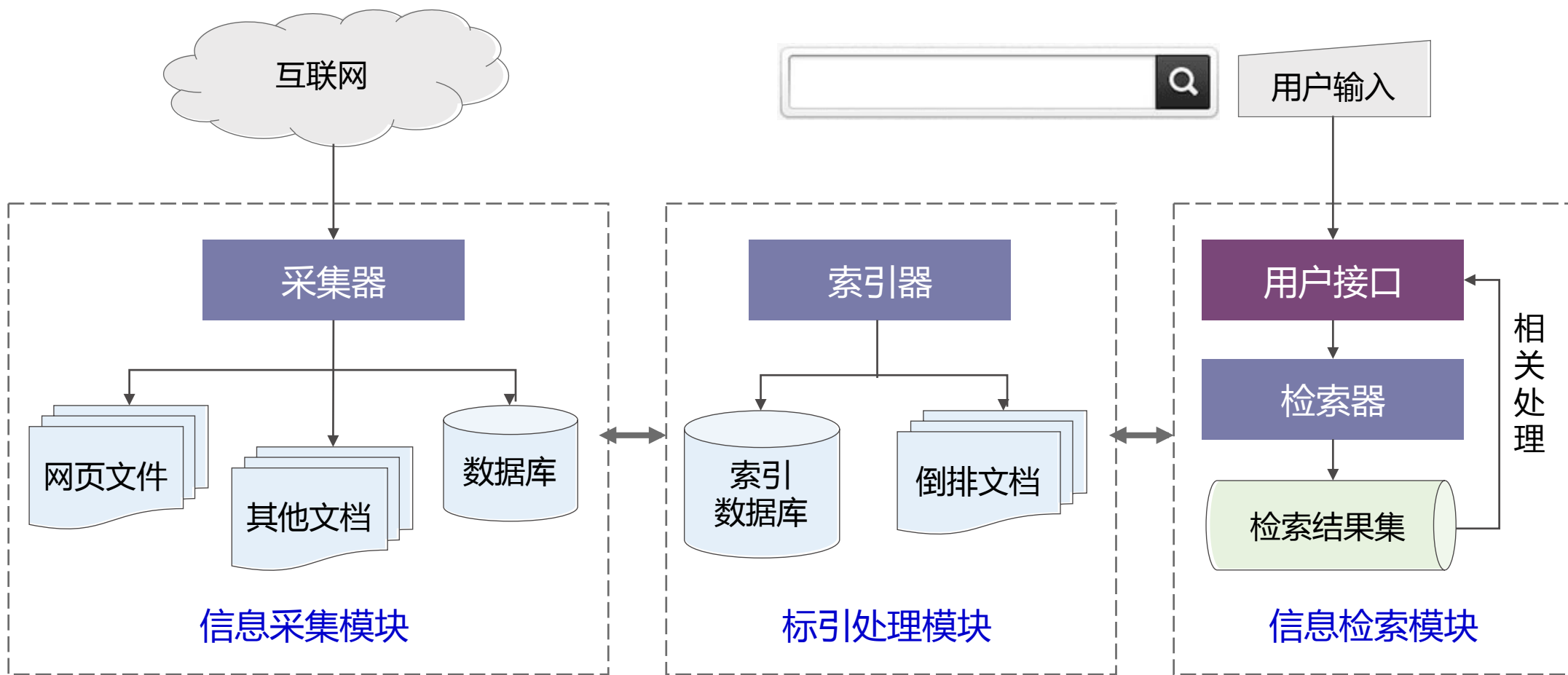


举例：Web信息检索是对发布在Web信息资源进行搜集、整理和组织，形成一个信息资源索引库，并通过检索界面将最符合用户要求的网站或网页提供给用户。

Web信息检索的处理过程：



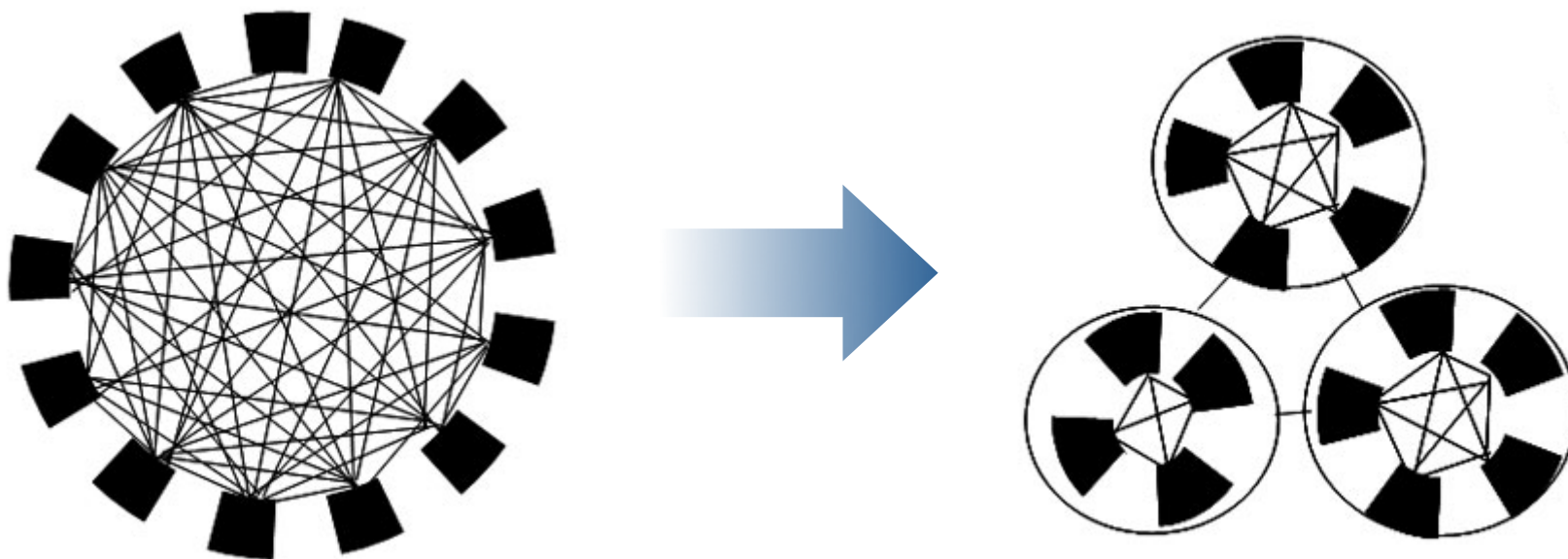
# 软件的复杂性



# 处理复杂性



随着软件系统的规模和复杂性不断增加，对系统的全局结构设计和规划变得比算法的选择和数据结构的设计明显重要得多。





# BOEING 787-8



- 1 Radome housing "smart" radar
- 2 Forward pressure bulkhead - composite
- 3 Two-crew cockpit featuring five 228 x 305mm Rockwell Collins LCDs, electronic flight instrument system and flight bag with dual head-up display system
- 4 Avionics bay
- 5 Forward retracting, fully steerable twin-wheel nose landing gear - Messier-Dowty
- 6 Lavatory
- 7 Cabin crew jump-seat
- 8 Flight crew access ladder
- 9 Flight crew rest area
- 10 Fuselage - single-piece composite fibre-glass carbon laminate
- 25 Aileron - fibre-glass
- 26 Main-wing box - carbon laminate
- 27 Outboard hinged panels - carbon laminate
- 28 Outboard flap - carbon laminate
- 29 Flaperon - carbon laminate
- 30 Inboard spoiler (winglet) - carbon laminate
- 31 Four wheel main landing gear - Messier-Dowty incorporating Goodrich or Messier-Bugatti electric brakes
- 32 Main landing gear support beam
- 33 Inboard flap - fibre-glass epoxy
- 34 Centre wing box - CFRP
- 35 Centrally located photochromatic passenger windows - 483 x 270mm
- 36 Mid-fuselage lavatories

Main instrument panel, shroud and centre console



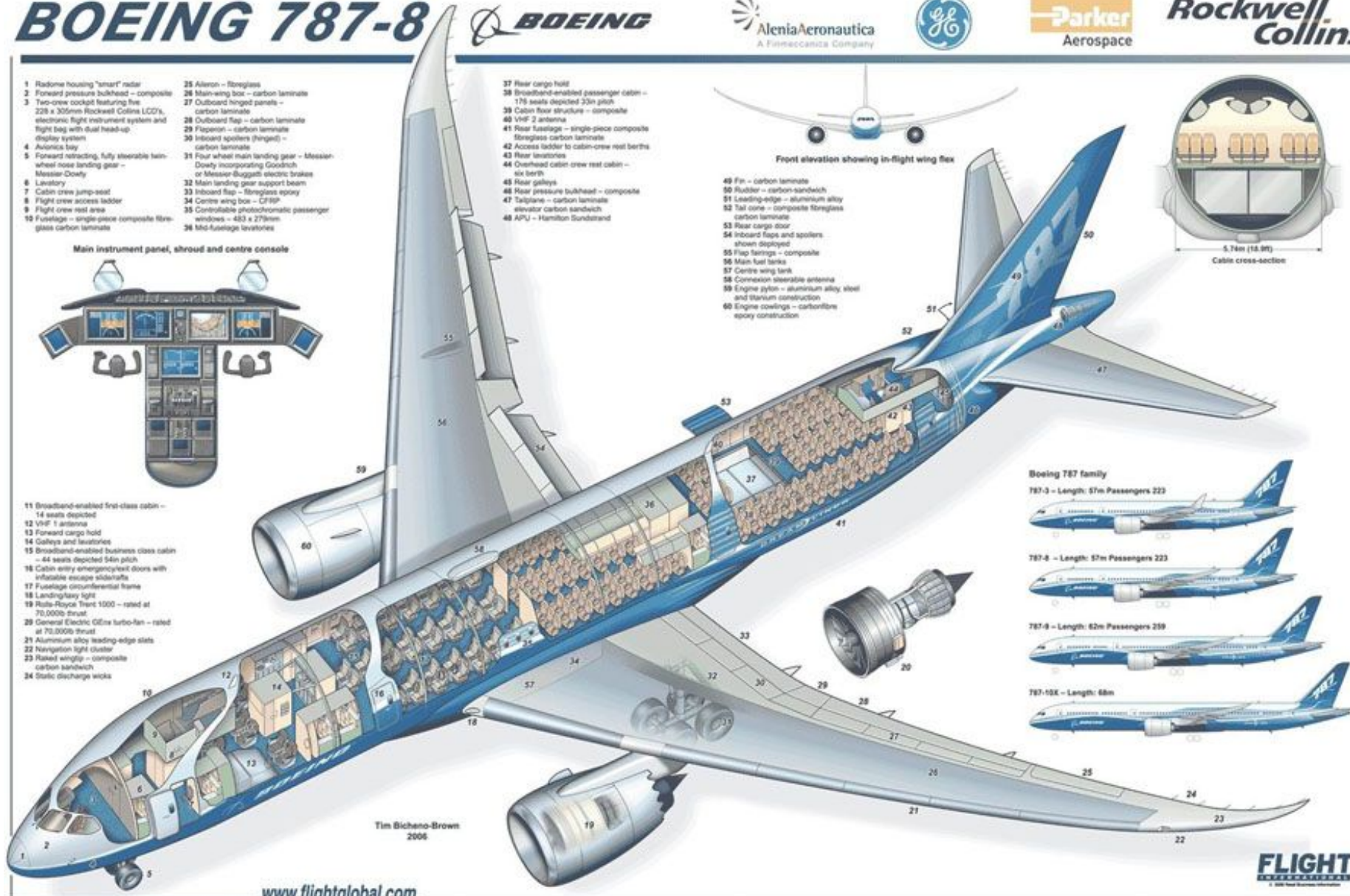
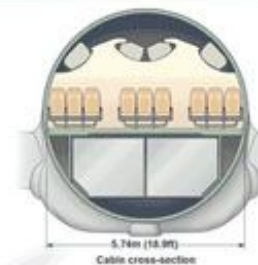
- 11 Broadband-enabled first-class cabin - 14 seats depicted
- 12 VHF 1 antenna
- 13 Forward cargo hold
- 14 Galleys and lavatories
- 15 Broadband-enabled business class cabin - 44 seats depicted 54in pitch
- 16 Cabin entry emergency/exit doors with inflatable escape slides
- 17 Fuselage circumferential frame
- 18 Landing/lift light
- 19 Rolls-Royce Trent 1000 - rated at 70,000lb thrust
- 20 General Electric GE90 turbo-fan - rated at 70,000lb thrust
- 21 Aluminium alloy leading-edge slats
- 22 Navigation light cluster
- 23 Raked wingtip - composite carbon sandwich
- 24 Static discharge wicks

- 37 Rear cargo hold
- 38 Broadband-enabled passenger cabin - 176 seats depicted 33in pitch
- 39 Cabin floor structure - composite
- 40 VHF 2 antenna
- 41 Rear fuselage - single-piece composite fibre-glass carbon laminate
- 42 Access ladder to cabin-crew rest berths
- 43 Rear lavatories
- 44 Overhead cabin crew rest cabin - six berths
- 45 Rear galleys
- 46 Rear pressure bulkhead - composite
- 47 Tailplane - carbon laminate
- 48 elevator carbon sandwich
- 49 APU - Hamilton Sundstrand



Front elevation showing in-flight wing flex

- 49 Fin - carbon laminate
- 50 Rudder - carbon sandwich
- 51 Leading-edge - aluminium alloy
- 52 Tail cone - composite fibre-glass carbon laminate
- 53 Rear cargo door
- 54 Inboard flaps and spoilers shown deployed
- 55 Flap fairings - composite
- 56 Main fuel tanks
- 57 Centre wing tank
- 58 Connection steerable antenna
- 59 Engine pylon - aluminium alloy, steel and titanium construction
- 60 Engine cowling - carbonfibre epoxy construction



Tim Bicheno-Brown  
2006

[www.flightglobal.com](http://www.flightglobal.com)

## Boeing 787 family

787-3 - Length: 57m Passengers 223



787-8 - Length: 57m Passengers 223



787-9 - Length: 62m Passengers 259



787-10X - Length: 68m



**FLIGHT**  
INTERNATIONAL







# 起源于建筑学的“体系结构”

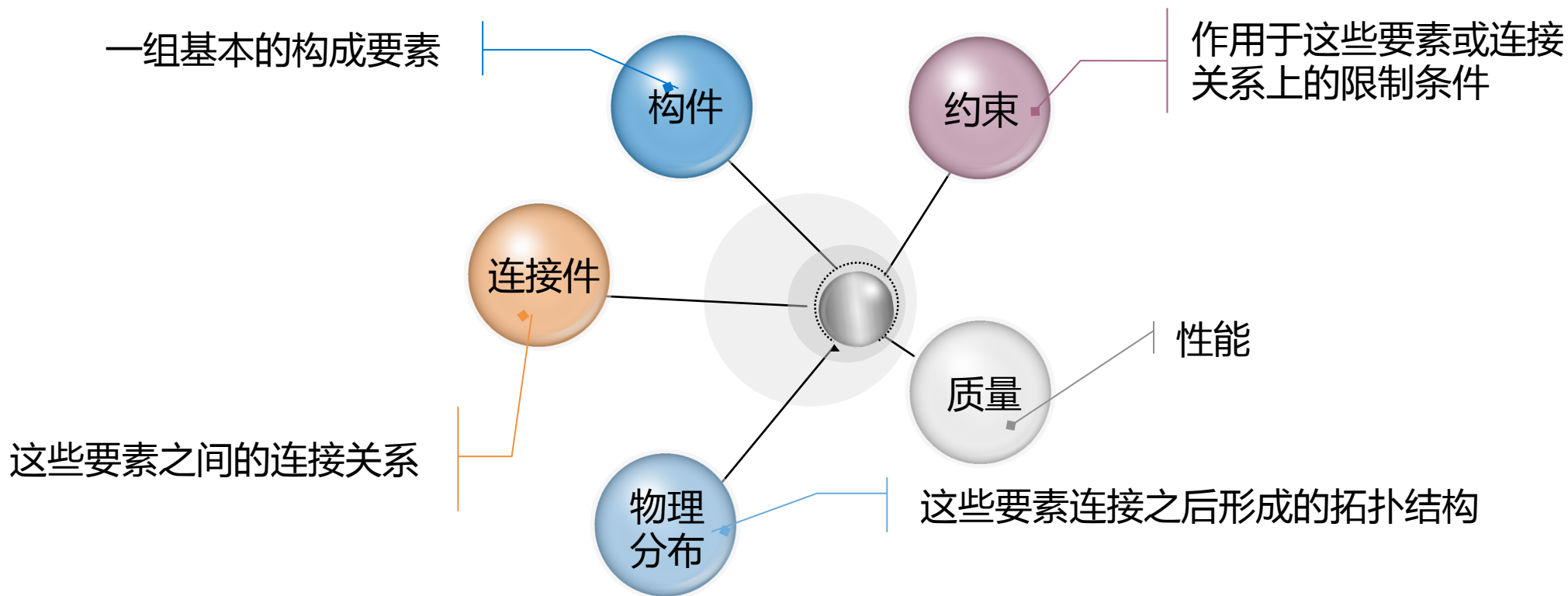


## 建筑设计原则：坚固、适用、赏心悦目

- 有哪些基本的建筑单元？
- 如何将这些基本单元进行组合形成整体建筑？
- 建筑单元怎样搭配才合理？
- 不同类型的建筑有什么典型的结构？
- 如何快速节省地进行建造施工？
- 怎样对建造完成的建筑进行适当的修改？
- 如何保证单元的更改不会影响整栋建筑的质量？

# 软件体系结构概念

软件体系结构（Software Architecture）包括构成系统的设计元素的描述、设计元素之间的交互、设计元素的组合模式以及在这些模式中的约束。

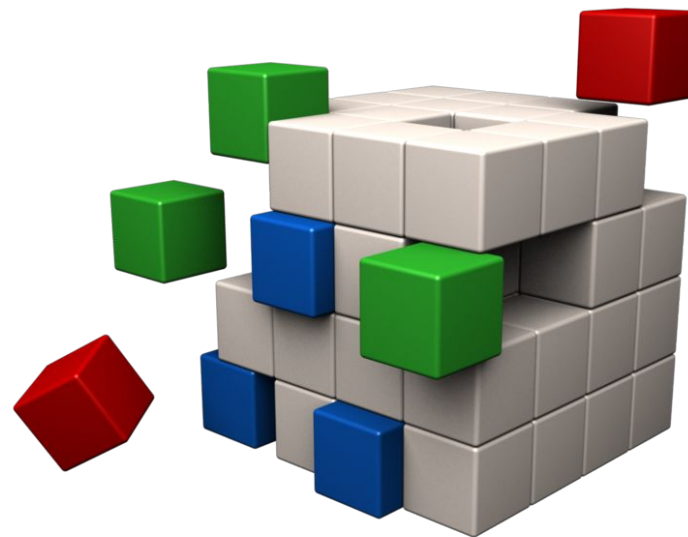


# 软件体系结构概念



软件体系结构 = 构件 + 连接件 + 约束

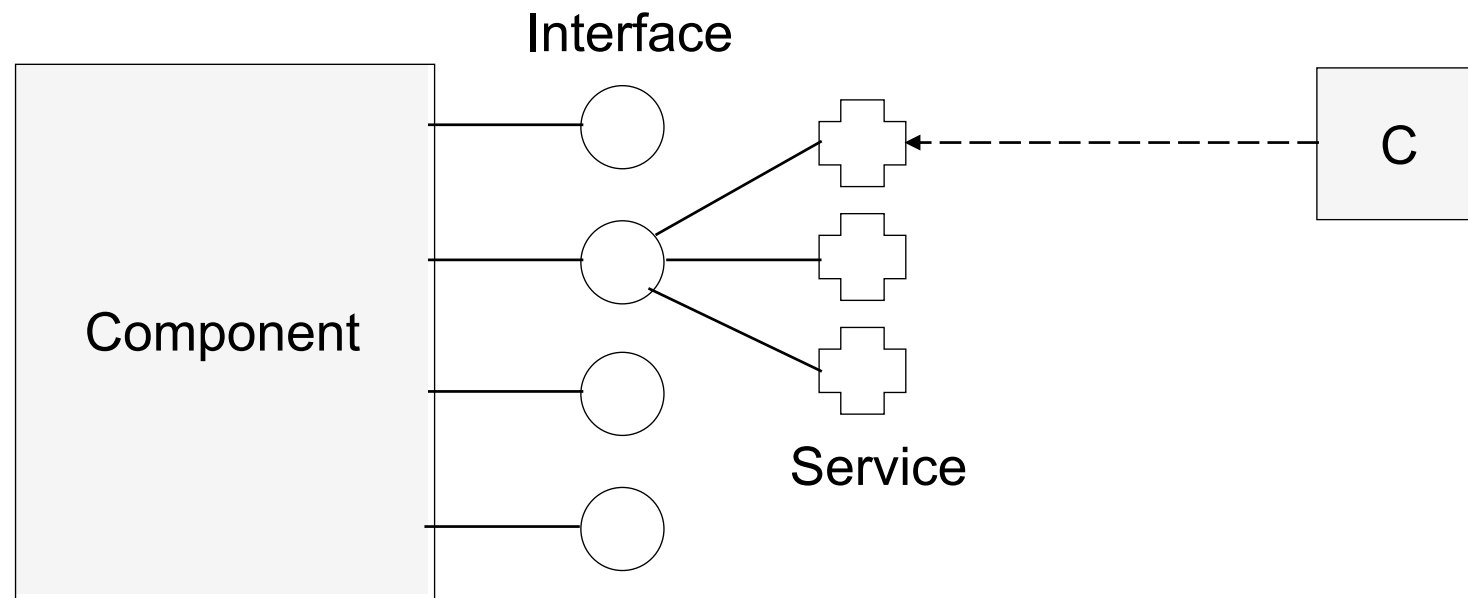
- 如何将复杂的软件系统划分成模块
- 如何规范模块的构成
- 如何将这些模块组织成为完整的系统
- 如何保证系统的质量要求



# 软件体系结构概念

**构件**是具有某种功能的可复用的软件结构单元，表示系统中主要的计算元素和数据存储。

程序函数、模块  
对象、类  
文件  
相关功能的集合  
.....



**连接**是构件间建立和维护行为关联与信息传递的途径。



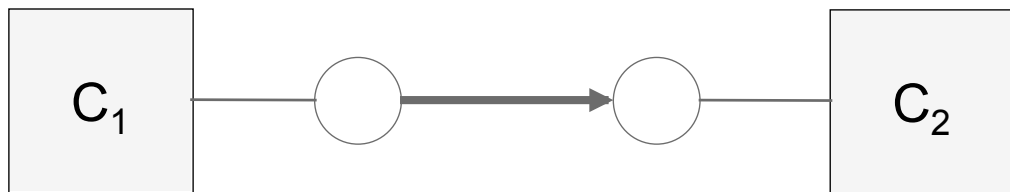
机制

过程调用、中断、I/O、事件、进程、线程、共享、同步、并发、消息、远程调用、动态连接、API 等等

协议

对过程调用来说：参数的个数和类型、参数排列次序  
对消息传送来说：消息的格式

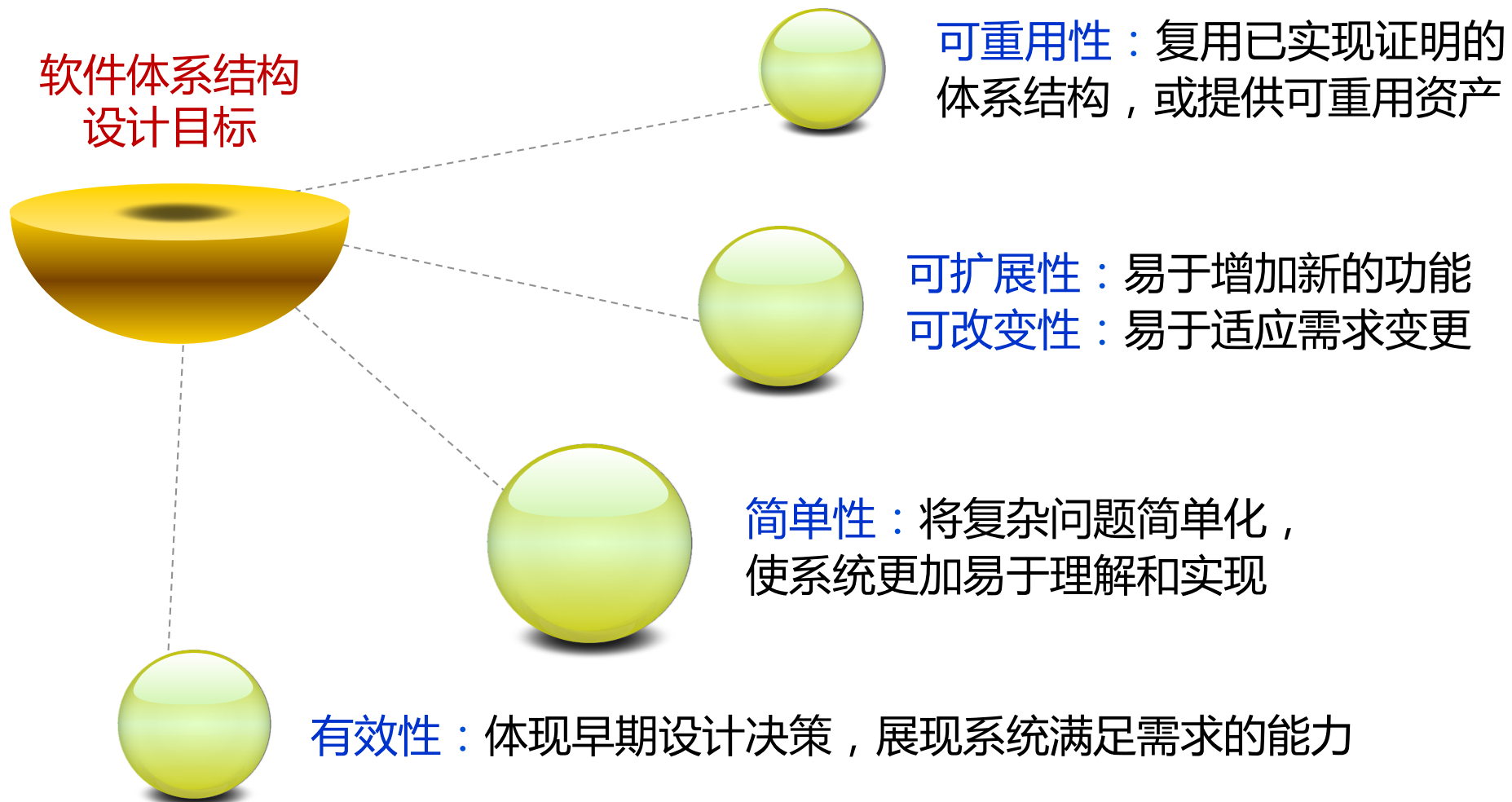
**连接件**表示构件之间的交互并实现构件之间的连接。



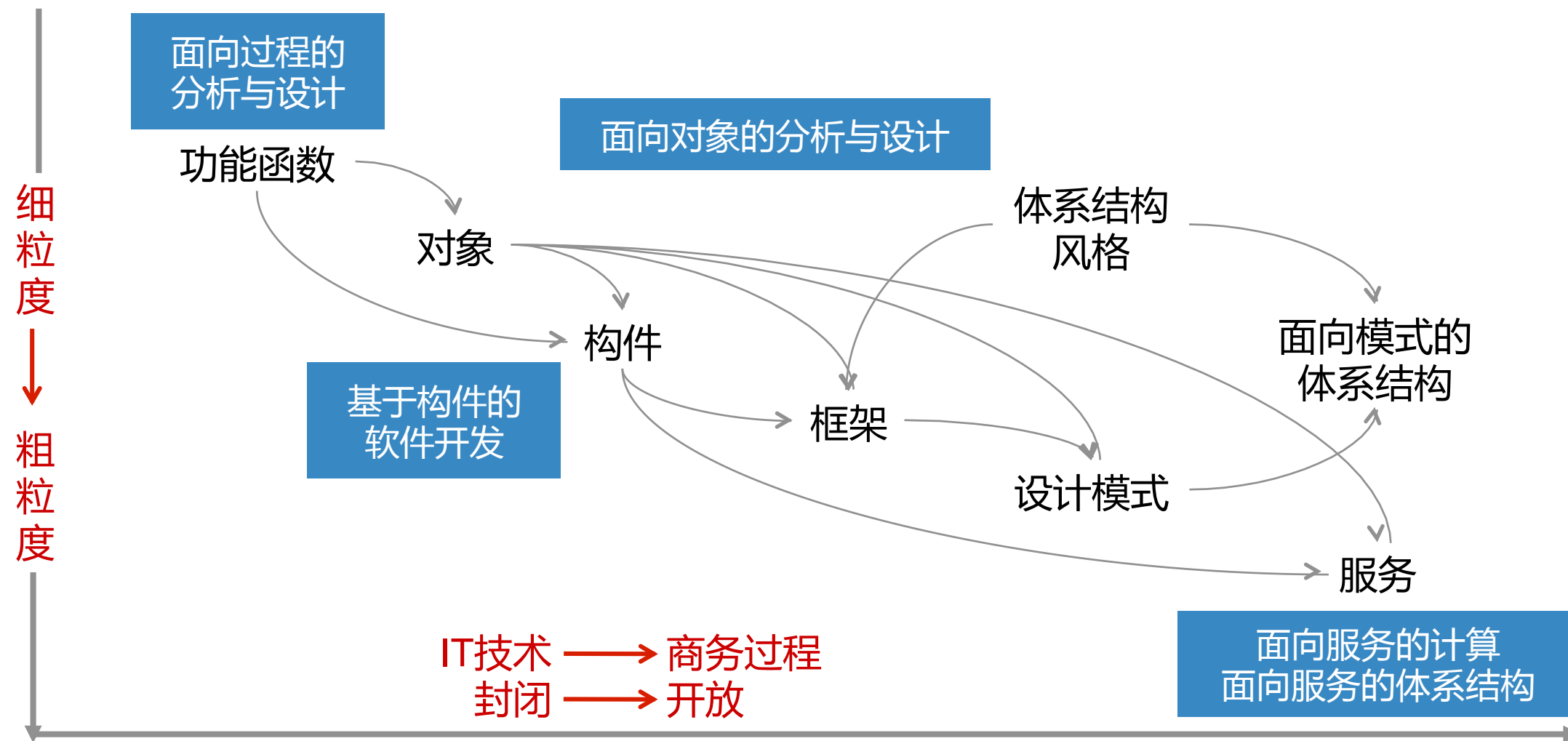
- 管道 ( pipe )
  - 过程调用 ( procedure call )
  - 事件广播 ( event broadcast )
  - 客户机-服务器 ( client-server )
  - 数据库连接 ( SQL )
- 
- 一般构件是软件功能设计和实现的承载体
  - 连接件是负责完成构件之间信息交换和行为联系的专用构件



# 软件体系结构的目标



# 软件体系结构的发展



# 风格、模式和框架

---



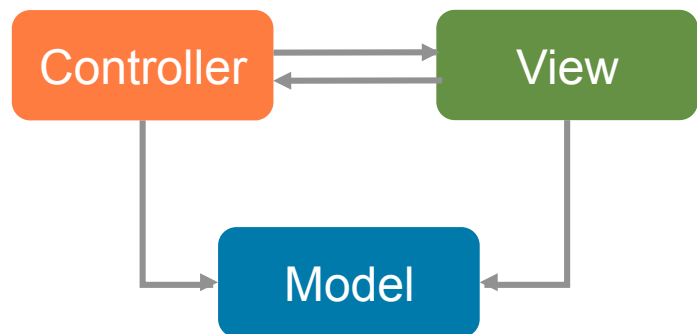
- **体系结构风格**：用于描述某一特定应用领域中系统组织的惯用模式，反映了领域中众多系统所共有的结构和语义特性。
- **设计模式**：描述了软件系统设计过程中常见问题的一些解决方案，通常是从大量的成功实践中总结出来的且被广泛公认的实践和知识。
- **软件框架**：软件框架是由开发人员定制的应用系统的骨架，是整个或部分系统的可重用设计，由一组抽象构件和构件实例间的交互方式组成。

# 风格、模式和框架



## 框架和体系结构的关系：

- 体系结构的呈现形式是一个设计规约，而框架则是“半成品”的软件；
- 体系结构的目的是指导软件系统的开发，而框架的目的是设计复用。



# 风格、模式和框架

---



## 框架和设计模式的关系：

- 框架给出的是整个应用的体系结构；而设计模式则给出了单一设计问题的解决方案，且可以在不同的应用程序或者框架中进行应用。
- 举例：一个网络游戏可以基于网易的Pomelo框架开发，这是一个基于Node.js的高性能、分布式游戏服务器框架；在实现某个动画功能时，可能会使用观察者模式实现自动化的通知更新。
- 设计模式的目标是改善代码结构，提高程序的结构质量；框架强调的是设计的重用性和系统的可扩展性，以缩短开发周期，提高开发质量。



# 谢谢大家！

---

## THANKS

