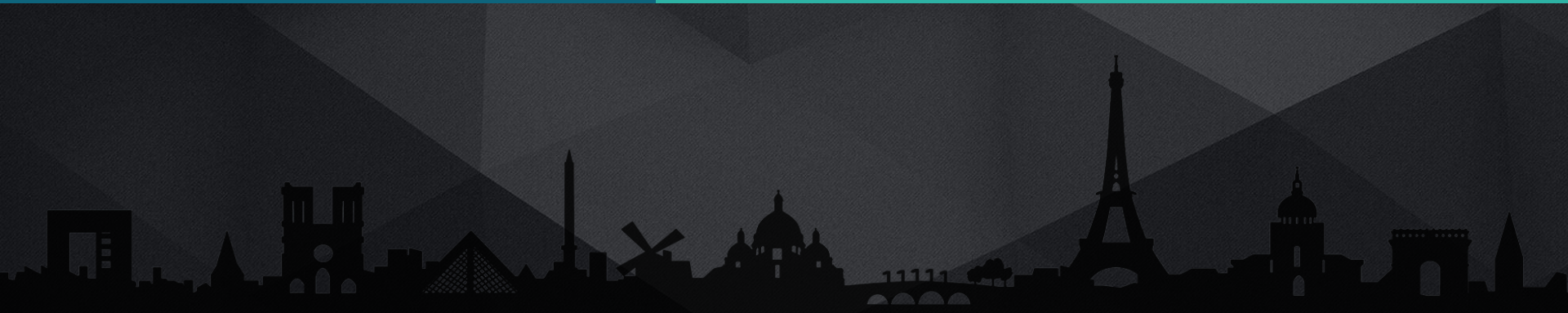


图书管理系统的体系结构设计



该系统的体系结构设计步骤：

一、构造系统的物理模型

二、设计子系统

三、非功能需求的设计

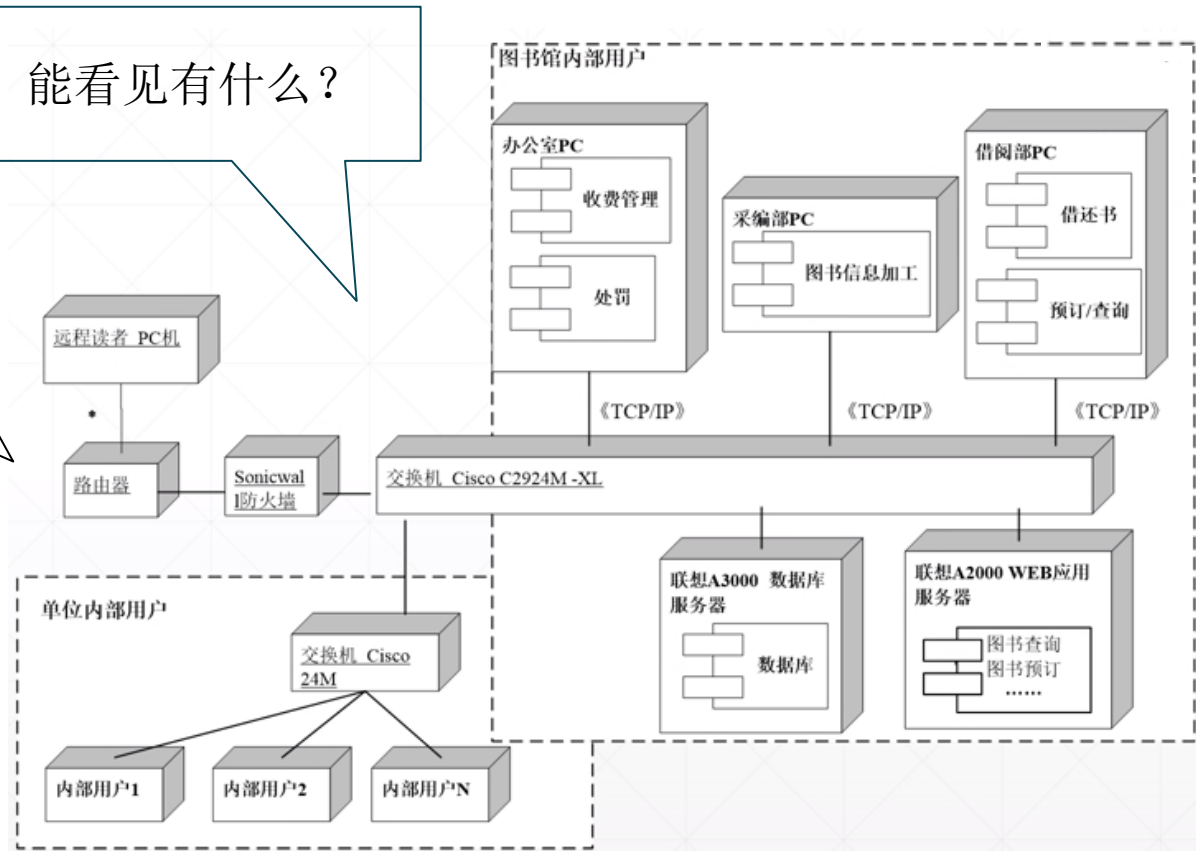
一、构造系统的物理模型

- 1、用UML的部署图描述系统的物理架构**
- 2、将需求分析获取的系统功能分配到这些节点上**
- 3、在配置图上显示计算机解答的拓扑结构、通信路径、硬件配置、节点上的运行的系统软件应用
软件配置等**

一、构造系统的物理模型

部署架构设计是一个反复迭代的过程，通常需要多次查看QoS要求和多次检查先前的设计，需要考虑了服务质量QoS需求的相互关系，平衡取舍相关问题成本以实现最佳解决方案，最终满足项目的业务目标。

能看见有什么？



二、设计子系统

1. 划分子系统

- ① 按功能划分
- ② 按物理布局划分
- ③ 按层次划分

2. 定义子系统之间的关系

- ① 请求-服务关系
- ② 平等关系
- ③ 依赖关系（如果有，要定义接口加以隔离），

3. 定义子系统的接口（信息隐藏、对接口编程）

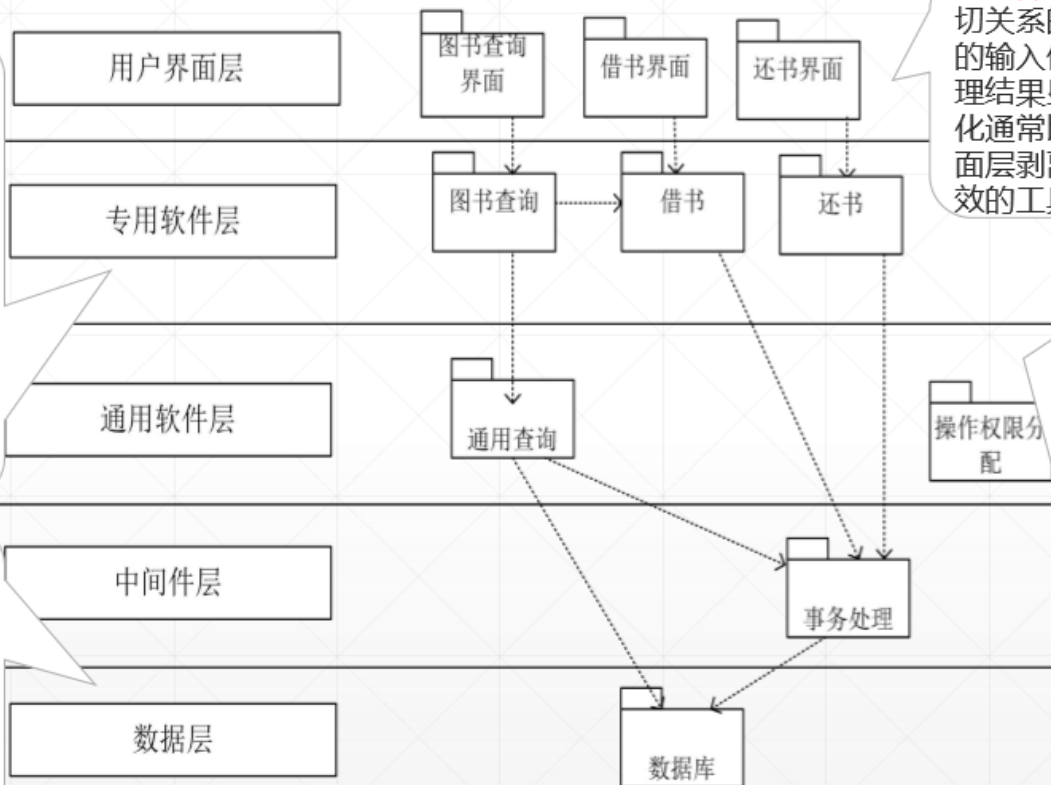
如果发现子系统之间关系太密切，耦合性太强的话，应该怎样做？

二、划分子系统

按照层次划分

专用软件层是每个项目中特殊的应用部分，它们被复用的可能性很小。在开发时可以适当地减小软件元素的粒度，以便分离出更多的可复用构件，减少专用软件层的规模。

数据层主要存放应用系统的数据，通常由数据库管理系统管理，常用的操作有更新、保存、删除、检索等。



用户界面层是与用户应用有密切关系的内容，主要接受用户的输入信息，并且将系统的处理结果显示给用户。这部分变化通常比较大，所以建议将界面层剥离出来，用一些快捷有效的工具实现。

通用软件层是由一些公共构件组成，这类软构件的可复用性很好。在设计应用软件时首先要将软件的特殊部分和通用部分分离，根据通用部分的功能检查现有的构件库。如果有可用的构件，则复用已有的构件会极大地提高软件的开发效率和质量。如果没有可复用的构件，则尽可能设计可复用的构件并且添加到构件库中，以备今后复用。

三、非功能需求的设计

- 系统的可靠性、易用性、安全性、错误监测和故障恢复、可移植性和通用性等等
- 具有共性的非功能需求一般设计在中间层和通用应用层，目的是充分利用已有构件，减少重新开发的工作量。