# 用例图

## 简单介绍

由****参与者（Actor）、用例（Use Case）**** 以及****它们之间的关系****构成的用于描述系统功能的动态视图称为用例图。用例图的构成元素又四个：****参与者（角色）****、****用例、系统边界、元素之间的关系****

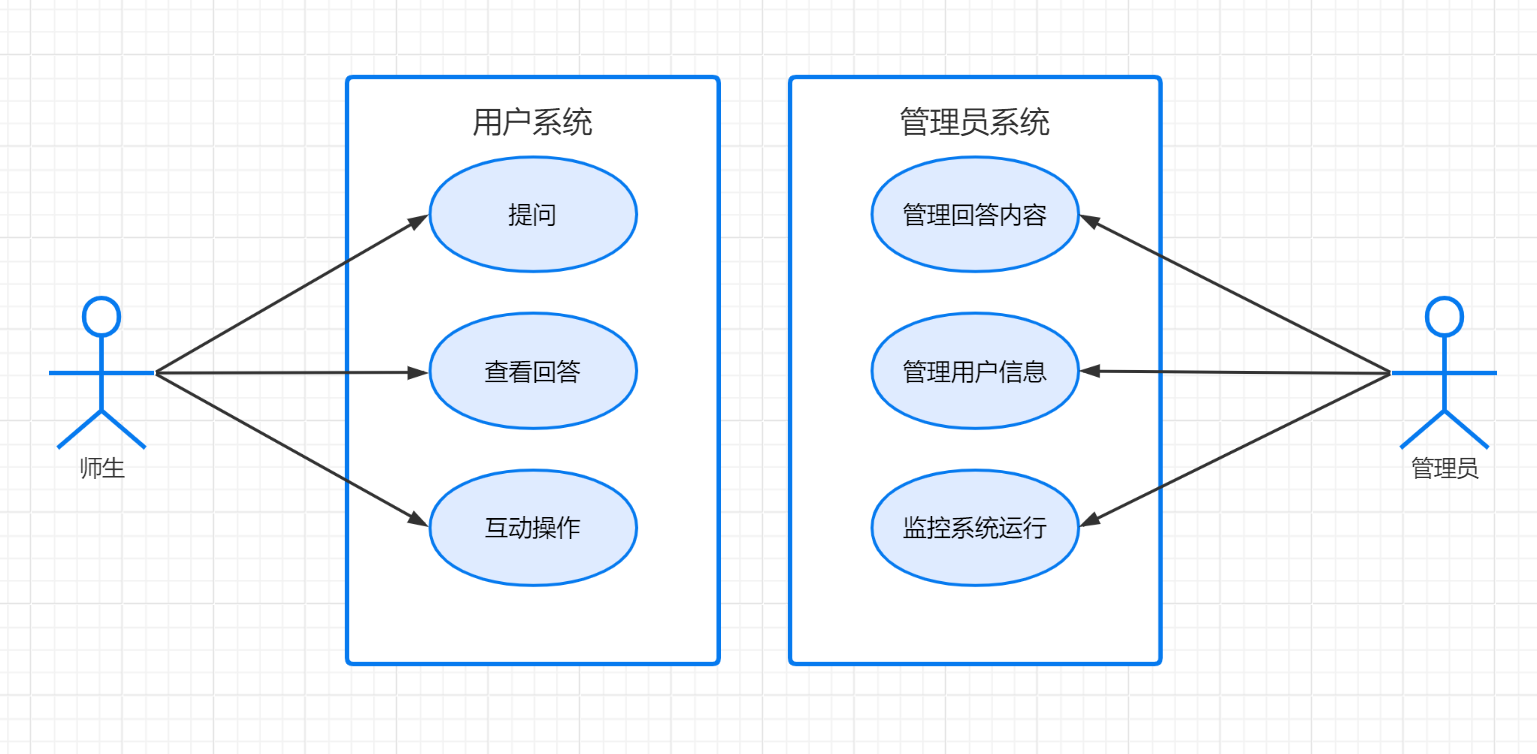
用例图中的关系有四种：****关联、泛化、包含、扩展****

| **关系类型** | **说明** | **表示符号** |
| --- | --- | --- |
| 关联 | 参与者与用例之间的关系 | IMG_256 |
| 泛化 | 参与者之间或用例之间的关系 | IMG_257 |
| 包含 | 用例之间的关系 | IMG_258 |
| 扩展 | 用例之间的关系 | IMG_259 |

参考资料：[用例图详解-CSDN博客](https://blog.csdn.net/cold___play/article/details/100824261)

<https://blog.csdn.net/cold___play/article/details/100824261>

## 简易用例图



1. ****师生****与****提问****、****查看回答****、****互动操作****之间存在关联关系，表明师生可以发起这些用例。
2. ****管理员****与****管理回答内容****、****管理用户信息****、****监控系统运行****之间存在关联关系，表明管理员能够执行这些用例。

# 类图

## 简单介绍

****类图****是面向对象系统建模中最常用和最重要的图，是定义其它图的基础。类图主要是用来显示系统中的类、接口以及它们之间的静态结构和关系的一种静态模型。

表示类的关系总共有6种，这6种又可以分为3类。

第一类，泛化。泛化表达了is a的关系模型，当A以某种形式是一个B的时候，就是这种关系，包含了有2种关系：继承、实现。

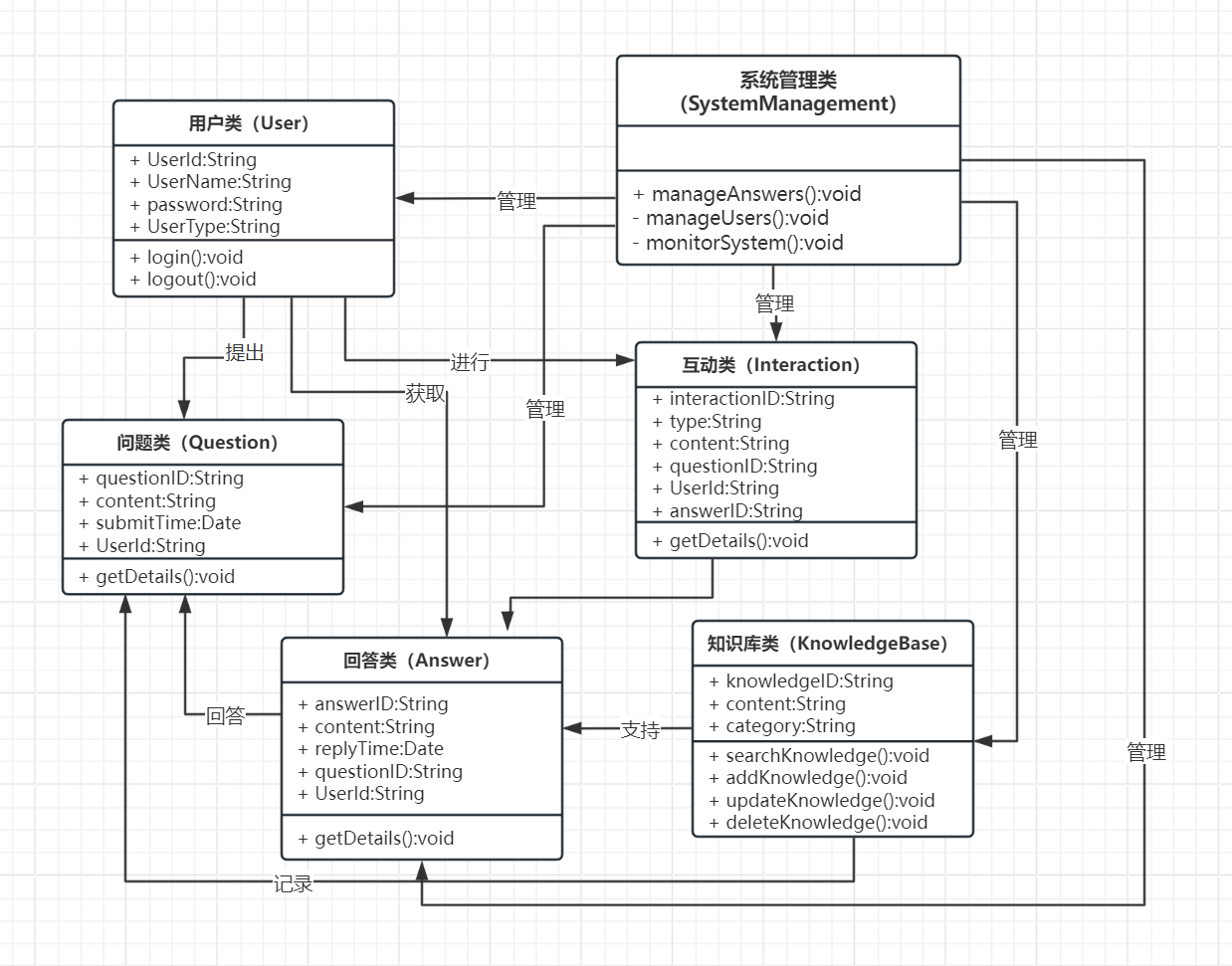
第二类，关联。关联表达了has a的关系模型，当A拥有一个B的时候，就是这种关系，包含了有3种关系：聚合、组合、关联。

第三类，依赖。依赖表达了use a的关系模型，当A使用了一个B的时候，就是这种关系，包含了1种关系：依赖。

原文参考：[UML详解（1）：类的关系与如何绘制类图\_怎么画uml类图-CSDN博客](https://blog.csdn.net/realmagicboy/article/details/90289062)

<https://blog.csdn.net/realmagicboy/article/details/90289062>

## 类图



1. 用户类（User）

2. 问题类（Question）

3. 回答类（Answer）

4. 互动类（Interaction）

5. 知识库类（KnowledgeBase）

6. 系统管理类（SystemManagement）

**类之间关系：**

1. ****用户类与其他类****
   * User 与 Question：一个 User 可以提交多个 Question，存在一对多的关联。
   * User 与 Answer：一个 User 可以给出多个 Answer，存在一对多的关联。
   * User 与 Interaction：一个 User 可以发起多个 Interaction，存在一对多的关联。
   * User 与 SystemManagement：管理员 User 可以使用 SystemManagement 类的方法，存在关联关系。
2. ****问题类与回答类****
   * Question 与 Answer：一个 Question 可以有多个 Answer，存在一对多的关联。
3. ****问题类、回答类与互动类****
   * Question 与 Interaction：一个 Question 可以有多个 Interaction，存在一对多的关联。
   * Answer 与 Interaction：一个 Answer 可以有多个 Interaction，存在一对多的关联。
4. ****知识库类与问题类、回答类****
   * KnowledgeBase 与 Question、Answer：KnowledgeBase 为 Question 的解答和 Answer 的生成提供知识支持，存在关联关系。

# 状态图

## 简单介绍

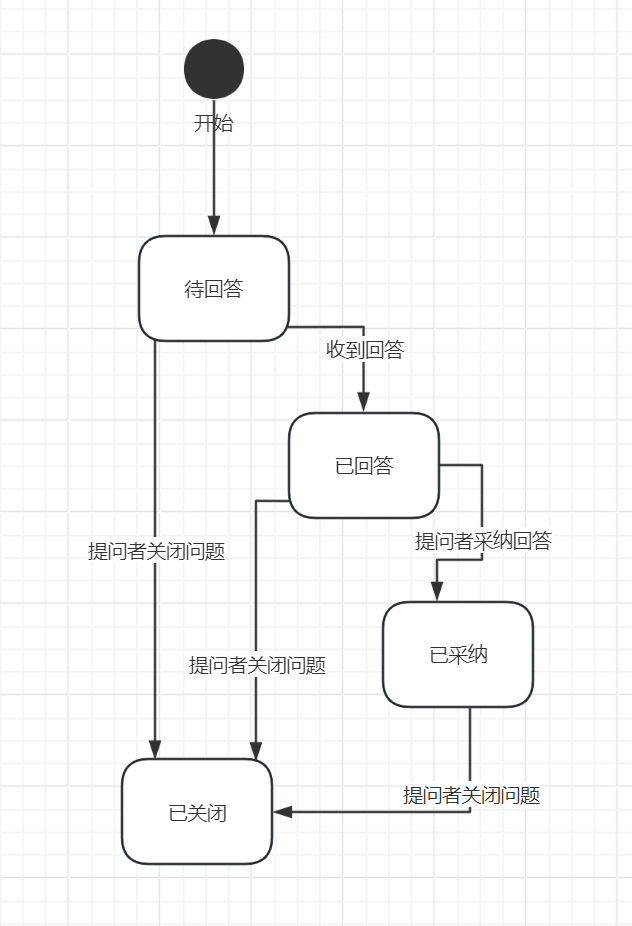
****状态图是一种行为图****。描述一个特定对象的所有可能的状态以及引起状态转换的事件。****状态机由状态、转移、事件、活动和动作五部分组成。****

|  |  |
| --- | --- |
| 状态机 | 注释 |
| 状态 | 状态指的是对象在其生命周期中的一种状况，处于某个特定状态中的对象必然会满足某些条件、执行某些动作或者是等待某些事件。一个状态的生命周期是一个有限的时间阶段。 |
| 转移 | 转移指的是两个不同状态之间的一种关系，表明对象在第一个状态中执行一定的动作，并且在满足某个特定条件下由某个事件触发进入第二个状态。 |
| 事件 | 事件指的是发生在时间和空间上的对状态机来讲有意义的那些事情。事件通常会引起状态的变迁，促使状态机从一种状态切换到另一种状态，如信号、对象额度创建和销毁等。 |
| 活动 | 活动指的是状态机中进行的非原子操作。 |
| 动作 | |  | | --- | | 动作指的是状态机中可以执行的哪些原子操作。所谓原子操作，指的是他们在运行的过程中不能被其他消息中断，必须一直执行下去，以至最终导致状态的变更或者返回一个值。 | |

参考资料：[面向对象软件系统中的状态图详解与应用实例-CSDN博客](https://blog.csdn.net/zhaxun/article/details/124201040)

<https://blog.csdn.net/zhaxun/article/details/124201040>

## 问题状态图



****解释****：

* 问题初始状态为 “待回答”，表示刚被用户提出，等待回答。
* 当收到回答时，问题状态变为 “已回答”。
* 提问者可以选择关闭问题，无论问题是否已有回答，状态都会变为 “已关闭”。
* 在 “已回答” 状态下，若提问者采纳了某个回答，问题状态变为 “已采纳”，最终都会走向 “已关闭” 状态结束流程。

# 顺序图（时序图）

## 简单介绍

顺序图有四部分：****参与者****（Actor)或对象（Object）、****生命线****（Lift Line) 、****激活期**** 和 ****消息****

****1、参与者（Actor) / 对象（Object）****

****参与者与对象按照一定顺序从左到右排列。UML中并没有对排列顺序作出任何要求，但是经过长期的积累得出以下经验：****

****活动的起始点放在最左侧。****

****交互频繁的参与者或对象靠拢。****

****2、生命线****

****每个参与者或对象都有生命线，生命线随着参与者的创建而产生，销毁而小时。当创建了参与者后，UML图上会顶部和底部各显示参与者，并且通过垂直虚线相连，这条垂直虚线就是生命线。****

****3、激活期****

****激活期是指对象在整个活动中获得了焦点，需要一段时间来执行某个动作，在UML图中是一个空心的矩形表示。****

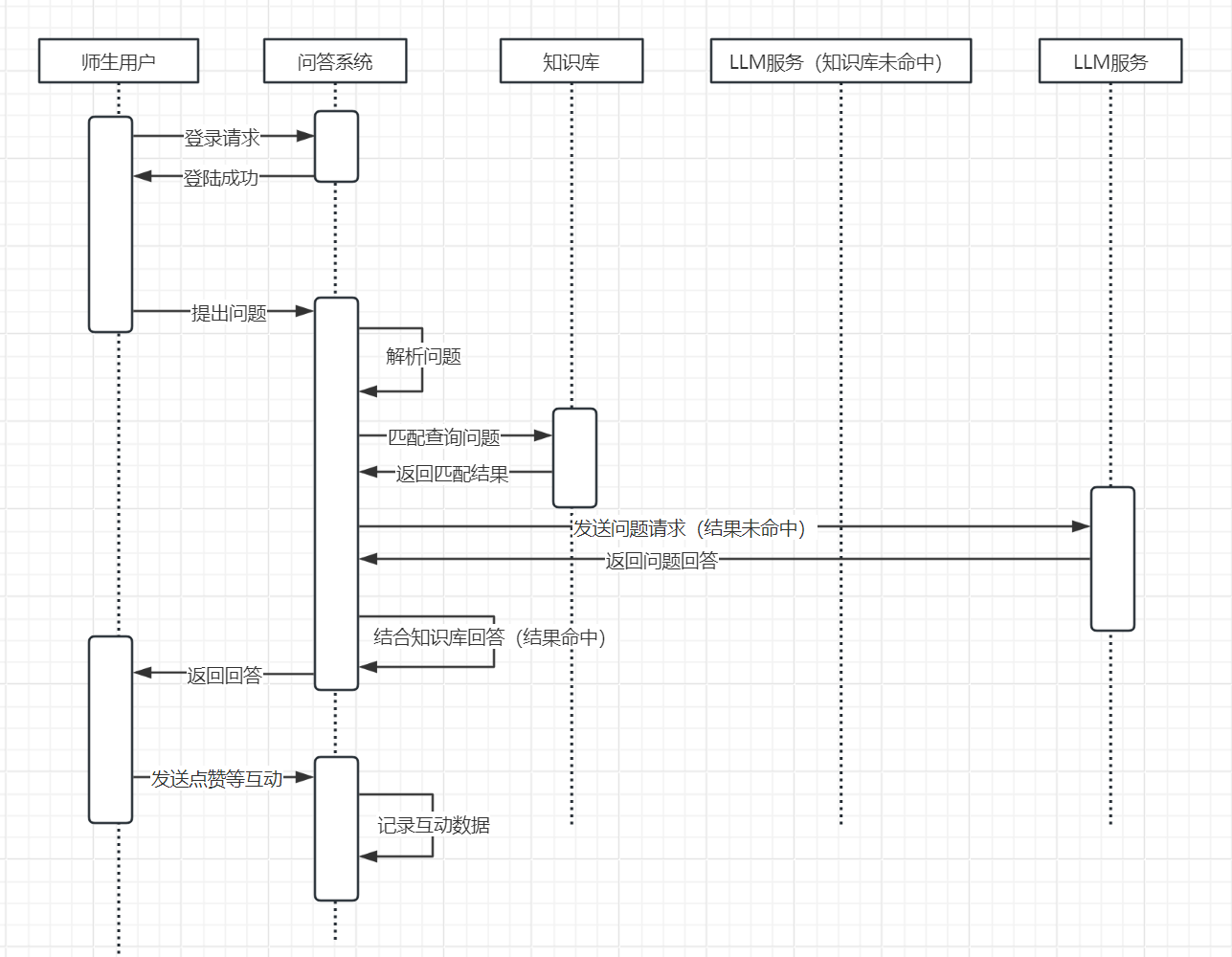
****4、消息****

****对象之间的交互是通过相互发消息来实现的。一个对象通过发送消息请求另一个对象做事件。****

****参考资料：****[UML顺序图详解：动态交互与场景流程-CSDN博客](https://blog.csdn.net/zhaxun/article/details/124267909)

<https://blog.csdn.net/zhaxun/article/details/124267909>

## ****用户提问 - 回答流程顺序图****



A)用户登录

用户通过账号密码登录，系统验证后返回会话令牌，用于后续操作鉴权。

B)问题提交与解析

用户输入问题（文本或语音），系统通过自然语言处理（NLP）提取关键词（如 “选课流程”“考试时间”）。

C)知识库优先匹配

系统先在本地知识库中查询是否有预定义的答案。若命中，直接整合答案；若未命中，调用外部 LLM 服务生成回答。

D)回答生成与反馈

无论是知识库还是 LLM 生成的回答，均由系统统一返回给用户。用户可对回答进行点赞、吐槽等互动，数据存入对应数据表。

# 协作图

## 简单介绍

协作图的描述

          协作图显示某组对象，如何为了由一个用例描述的一个系统事件而与另一组对象进行协作的交互图。

协作图的组成部分

协作图强调参与一个交互对象的组织，基本元素包括：活动者（Actor）、对象（Object）、连接（Link）和消息（Message）。

对象：用长方形框表示对象。

连接：使用实线标记两个对象之间的连接。

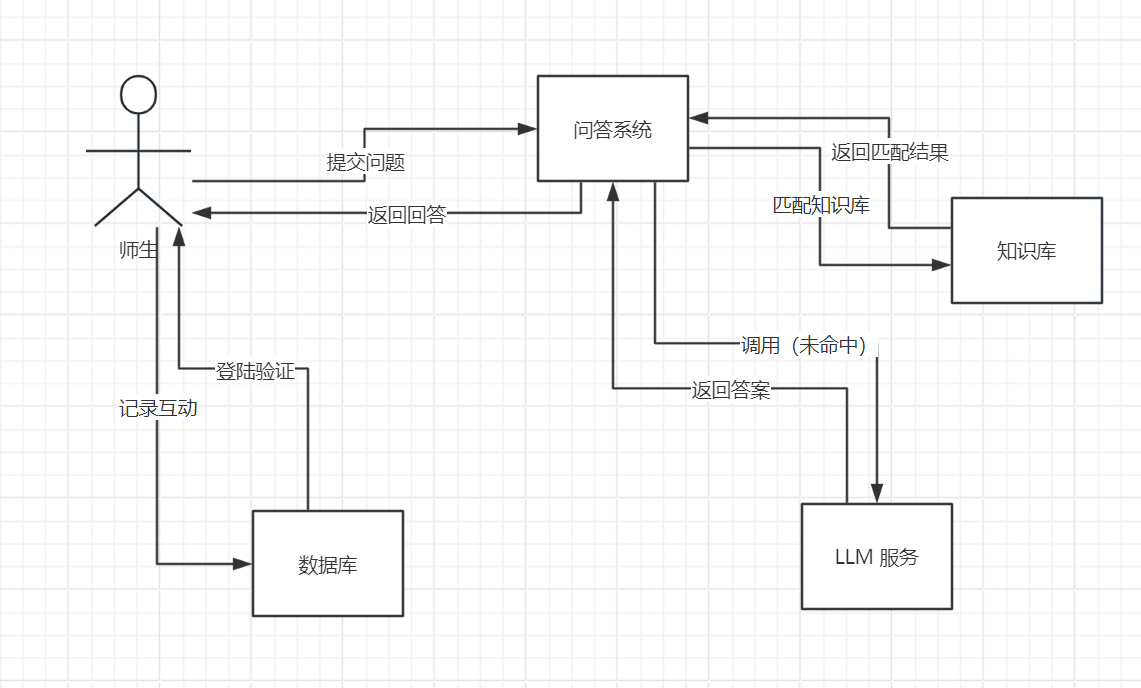
消息：由标记在连接上方的带有标记的箭头表示。

使用协作图可以显示对象角色之间的关系，如为实现某个操作或达到某种结果而在对象间交换的一组消息，如果需要强调时间和序列，最好选择序列图，如果需要强调上下文相关，最好选择协作图。

参考资料：[UML协作图解析-CSDN博客](https://blog.csdn.net/nangeali/article/details/50879385)

<https://blog.csdn.net/nangeali/article/details/50879385>

## ****用户提问 - 回答协作图****



****用户流程****：  
登录验证 → 提交问题 → 知识库优先匹配 → 未命中则调用 LLM → 返回回答 → 记录互动数据。

# 部署图

## 简单介绍

部署图(deployment diagram，配置图)：是用来显示系统中软件和硬件的物理架构。

部署图用于描述部署软件组件的硬件组件。构件图和部署图密切相关。

（1）可视化系统的硬件拓扑

（2）描述用于部署软件的硬件组件

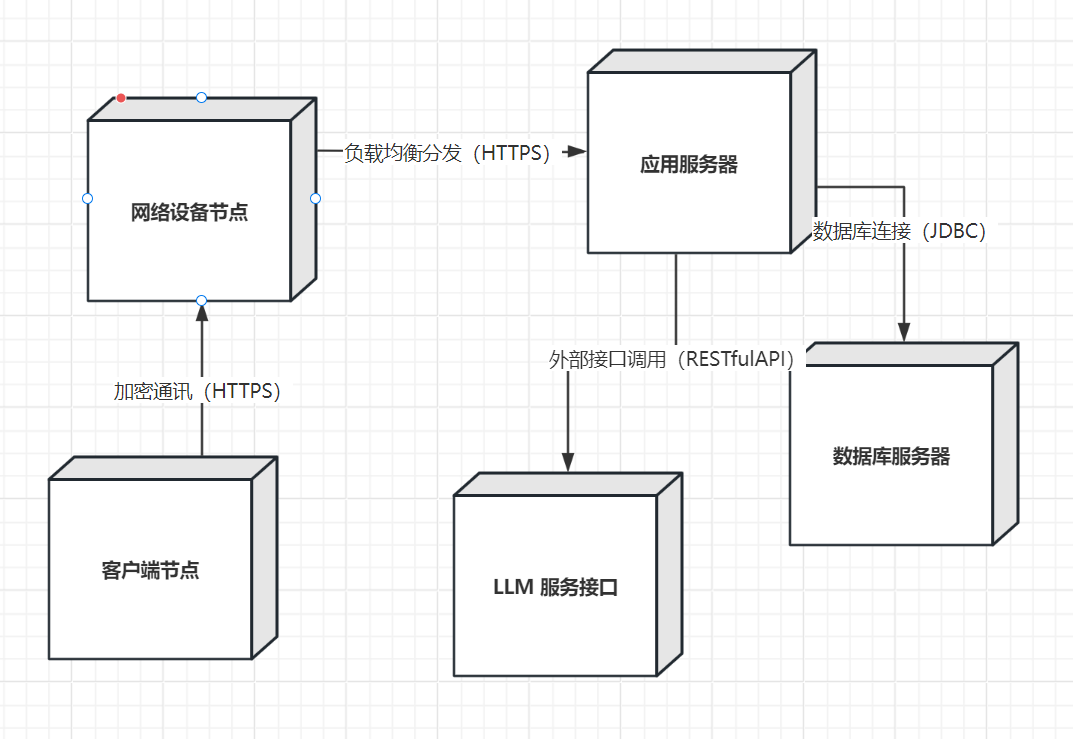
（3）描述运行时处理节点

部署图由节点和关系两部分组成。1、节点：计算资源的通用名称。包括处理器和设备。2、关系：依赖和关联

参考资料：[UML14种图之部署图和构件图最全总结-CSDN博客](https://blog.csdn.net/daima3/article/details/102794897)

<https://blog.csdn.net/daima3/article/details/102794897>

## 简易部署图



含有客户端节点、网络设备节点、服务器节点、外部服务节点