**本次作业：**

**请列举出一个现实中的产品/系统/服务/技术/..., 并说明其包含了/使用了哪些软件架构，这样设计的好处和/或坏处是什么？**

**Vue.js**

Vue.js是一种基于Model-view-viewmodel模式的前端JavaScript框架，用于构建用户界面和单页应用。

**MVVM**

Model-view-viewmodel（简称为MVVM）是一种软件体系结构模式，由MVC模式发展而来。MVVM的组成部分有：Model，View, Viewmodel和 Binder。

Model是指表示真实状态内容的域模型（面向对象的方法），还是表示内容的数据访问层（以数据为中心的方法）。

View是用户在屏幕上看到的内容的结构，布局和外观。它显示模型的表示形式，并接收用户与视图的交互（鼠标单击，键盘输入等），并通过数据绑定（属性）将这些操作的处理转发给Viewmodel。

Viewmodel是视图对公共属性和命令的抽象。MVVM具有绑定器，没有MVC模式的控制器，该绑定器可自动执行视图及其在视图模型中的绑定属性之间的通信。Viewmodel可理解为模型中数据的状态。

Binder使开发人员不必编写模板逻辑来同步视图模型和视图。

**好处：**

当JavaScript框架开始支持异步编程技术时，不再需要Web应用程序请求完整的Web页面，虽然网站和应用程序可以通过部分更新视图来更快地响应，但是与此同时造成了一定程度的重复。而MVVM相对于MVC引入了viewmodel及其数据绑定。

MVVM可用于构建具有更快速响应的用户交互和反馈的客户端应用程序，同时避免在整个体系结构中进行昂贵的代码重复和工作。对于Web应用程序，MVVM的设计使我们能够编写能够对用户交互立即做出响应的软件，并允许用户自由地从一项任务转移到下一项任务。这种职责的合并对应用程序的视图具有深刻的影响：当视图模型中的数据发生更改时，绑定到该视图的任何视图都会自动更新。数据绑定器可公开数据，保证当数据发生更改时，都能在视图中反映。

比如说，在Vue示例中，当v-modal的绑定值更改时，页面将自动呈现更新的相应值。我们只需要声明视图模板，并绑定相应的数据模型，而不需要过分注意视图层，这大大提高了前端开发的效率。

**坏处：**

各种MVVM组件和数据绑定之间的信息交流（通信）会增加操作的复杂度。然后是视图和视图模型的代码可重用性很差。

在嵌套视图和复杂的UI中管理视图模型及其状态很困难。

MVVM对于初学者来说不太友好。

对于简单的UI，MVVM可能会过于庞大。在这种情况下，很难预先设计ViewModel以获得适当的通用性。