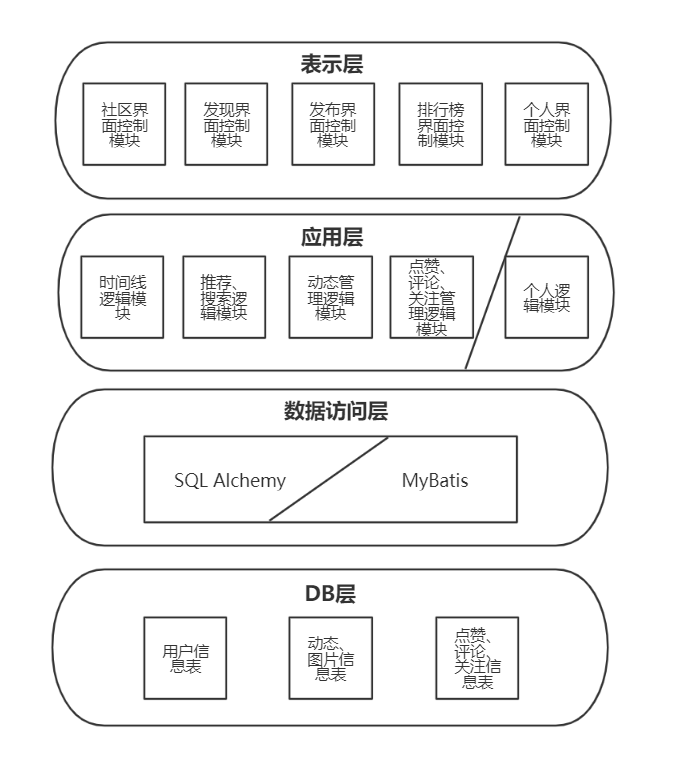
架构设计

软件架构设计

Stack社区的系统构架工作是整个系统设计阶段的重要内容，其中对系统软件体系结构和硬件架构的设计尤为重要，需要结合各类软件技术和硬件服务，围绕用户操作、业务逻辑处理和数据信息存储等重要内容进行设计。本平台的系统软件体系结构以分层模式为主体，将系统分成不同的抽象层次，每一层都为上层提供好用的API，并屏蔽掉下层的细节。这里主要采用四层架构的方式进行实现，详细软件架构设计如图4.1所示。



从图4.1可以看出四层体系软件架构设计方面的内容，按照业务类型、功能类别、关联关系等方式，对系统每个逻辑层涉及的相关功能和服务进行模块设计，每一层操作的具体内容如下：

表示层

表示层利用CSS和JS等技术在设计的界面控制模块中的index.ux文件中完成代码编写，实现用户交互前端界面。交互界面主要包含主Tab中可自由左右滑动的五个控制模块：社区界面、发现界面、发布界面、排行榜界面以及用户个人界面。其中发现界面上方包含搜索栏，个人界面上方包含设置界面入口，各子界面都可通过这些主界面来回切入。

应用层

表示层和应用层的划分是前后端分离架构的体现，通过将前后端进行有效的解耦，系统才能被模块化，从而为之后的分布式框架、微服务框架等打下坚实的基础。因此表示层的Javascript代码除了负责部分界面模块的动态交互，还负责调用应用层在服务器后端提供的接口。后端接收前端发来的以POST表单提交为主体的HTTP请求，利用数据访问层提供的接口进行相应数据处理，并通过一定包装后将其以JSON格式再传回前端。

应用层的主体——服务器后端根据开发人员的特点也存在一定程度的分离。个人逻辑模块部分的API接口主要来自基于Java语言的Spring Boot框架，而其余部分的API接口则来自基于Python语言的Flask框架。

数据访问层

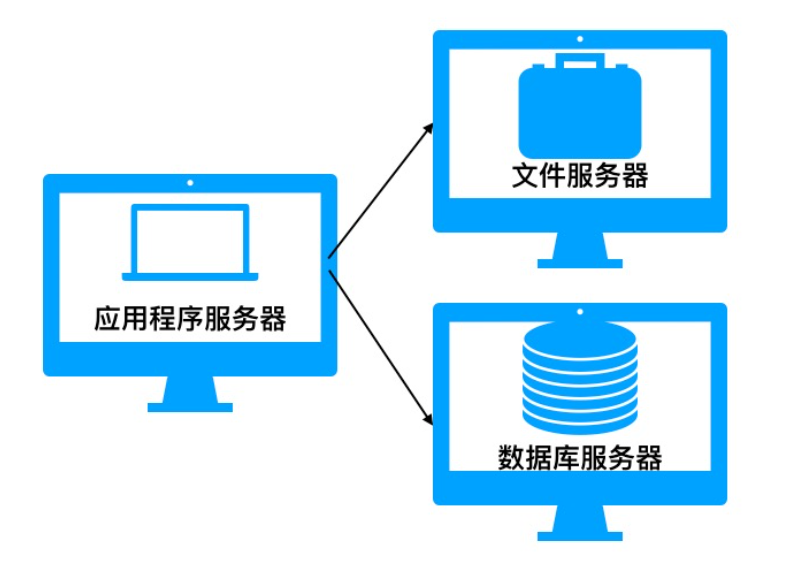
API部分的分离也导致了数据访问层的分离。其中源于Spring Boot框架中的部分API通过ORM Mybatis将面向对象语言程序中的对象自动持久化到关系数据库中，而 Flask中API则主要通过SQL Alchemy完成相应功能。尽管使用的插件不同，但二者都是通过对象关系映射，通过连接远程云关系数据库，通过使用DB层的接口完成一切增删查改的数据操作。

DB层

DB层采用基于华为云的关系数据库MySQL存储数据表，向数据访问层提供数据支撑。具体情况将在数据结构设计中描述。

硬件架构设计

系统硬件架构是从物理部署的角度对系统的总体框架进行分析。而在云服务非常普遍的今天，综合经济成本、服务器性能、可靠性和安全性等方面考虑，将应用程序、文件和数据库分离，分别部署在独立的云服务器上。如图。



由于当前Stack社区处于开发的初级阶段，使用用户较少，因此暂未使用缓存、负载均衡、分布式等技术提升整体性能。

以上的基础式应用程序、文件、数据库分离服务器架构将这三个部分分别部署于不同的云服务器中：

* 应用程序服务器：
  + 基于Java语言的Spring Boot框架部署于阿里云的轻量应用服务器 （Simple Application Server）中；
  + 基于Python语言的Flask框架部署于华为云的弹性云服务器（Elastic Cloud Server）中。
* 文件服务器：部署于腾讯云的对象存储（Cloud Object Storage，COS）中，基于其提供的支持 HTTP/HTTPS 协议访问的分布式存储服务实现社区图片文件的高效存储。
* 数据库服务器：部署于华为云的云数据库（RDS for MySQL）中，利用其高可靠性和高安全性的特点，实现了数据访问层和DB层的解耦合。

将以上基础式应用程序、文件、数据库分离服务器架构映射到具体服务器后架构图如下。其中上传到对象存储在前端实现，将图像映射为唯一URL后通过后端云服务器上传到关系数据库，避免了大文件的二次上传。此外，通过各家云服务器运营商提供的云监控可以实时监测服务器使用情况，可以有效实现数据统计及基于此的服务器调控和流量预测等功能。

