Doc Translate Viewer 系统架构设计文档

# 一、系统总体架构

Doc Translate Viewer 系统采用前后端分离架构，整体系统分为三层：表示层、业务逻辑层和数据访问层。  
系统组件包括：Web 前端、翻译服务 API、文档处理模块、存储服务、用户管理模块等。

系统架构逻辑图如下：

（此处可插入架构图，当前为文字描述）

用户通过 Web 前端上传文档 -> 后端 API 接收请求 -> 文档预处理模块解析内容 -> 翻译服务调用第三方 API -> 翻译结果存储至数据库或临时文件 -> 前端展示翻译内容。

# 二、模块设计

## 2.1 用户界面模块

基于 React/Vue 实现，负责用户登录、文档上传、翻译触发、结果查看等操作。通过 REST API 与后端交互，展示翻译状态与内容。

## 2.2 文档处理模块

支持 DOCX、PDF、TXT 等文档格式解析，提取文本与段落结构。对提取后的内容进行预处理，如清理特殊字符、语言识别等。

## 2.3 翻译服务模块

封装主流翻译引擎 API（如 Google Translate、DeepL），统一接口调用。根据语言类型与段落内容异步发起翻译任务，支持批量翻译与缓存。

## 2.4 存储模块

使用数据库或文件存储方式保存用户文档、翻译任务状态、翻译结果。支持历史记录管理、用户数据隔离。

## 2.5 用户与权限模块

支持用户注册、登录、权限校验，管理用户文档与任务访问权限。

# 三、系统部署架构

前端部署于 CDN 或静态服务器，后端服务可基于 Flask/Django 等框架运行于应用服务器，数据库使用 MySQL/PostgreSQL，文件存储支持本地或云存储（如阿里云 OSS、Amazon S3）。  
可使用 Docker 容器化部署，结合 Nginx 做统一入口与负载均衡。