# 系统设计文档

## 1. 项目之间的依赖关系：

说明：CMS-MODEL是基础项目。所有项目的核心。 其他项目运行都依赖于CMS-MODEL、项目采用MOVEN构建。未来可以配合Jekins做持续集成。留出扩展空间。

## 2. 项目流程图：

分发程序(RSYNC|RUBY)

CMS-PUBLISH

RabbitMQ

CMS-WEB

Tengine(外网)

预览

CMS-SEARCH

Nginx(内网)

手机端：

IOS

H5

CMS-API

Android

说明：

1. 咱们的架构中采用前后端分离。后端服务只提供数据接口、前端托管到Nginx即可。前端使用AngularJs框架。目前比较不错的mvc框架。
2. 手机端使用H5的模式。未来修改不需要重新发版。避免原生客户端的发版问题(针对IOS)。
3. 预览功能通过 Nginx 转发结合静态页资源进行预览。
4. 多个程序之间通信采用异步消息RabbitMq进行通信AMQP协议。无阻塞。无感知生成。缓存使用Redis存储。静态页分发使用NFS/RSYNC

## 3. 项目代码结构图：

说明：代码结构考虑到分层清晰、定义明确。共用性强。结合之前经验。提前预留一些扩展的内容。比如异常管理、日志管理、业务与事务分离。事务内不掺杂任何第三方调用(降低Mysql由于程序中不合理使用 造成连接达到所定阀值。降低第三方出现问题造成整体程序变慢甚至宕机的问题)。第三方调用共享等。

## 4. 项目部署示意图：

Mysql(InnoDB引擎)

RabbitMq

Redis(提供缓存)

ElasticSearch(提供检索)

中间件服务器

后台API

NGINX

后台管理平台

分发服务

服务器A

NGINX

H5

API服务

服务器B

Redis(6379)

Sentinel

分发

说明：分三个服务器 第一个服务器主要是放中间件。第二个服务器主要是内网服务。后台管理平台、分发服务、后台API、Nginx服务。Nginx主要是转发请求、静态页容器，第三个服务器主要存放IOS、Androd的H5服务、APP-API、Nginx做容器和转发

最后一个Redis采用Sentinel做Redis群组管理。暂时单点。未来可以直接添加多个Redis注册到Sentinel即可。如果一台Redis故障可以直接切换另一个。