
ACHIEVO SZ

路演天下 架构设计说明书

2016 年 4 月 20 日发布
2016 年 4 月 20 日实施

起草：胡登军

审批：

批准：

版权声明和保密须知

本文件中出现的任何文字叙述、文档格式、插图、照片、方法、过程等内容，除有特别注明，版权均属 ACHIEVO 所有，受到有关产权及版权法保护。任何单位和个人未经 ACHIEVO 的书面授权许可，不得复制或引用本文件的任何片断，无论通过电子形式或非电子形式。

Copyright © 2012

ACHIEVO 版权所有

修订记录

版本	修订人	修订日期	修订内容
V1.0	胡登军	2016/4/20	创建文档
V1.1	胡登军	2016/4/28	修复错误描述
V1.2	胡登军	2016/4/29	调整物理架构，互动功能单独部署

目 录

1.1 编写目的.....	4
1.2 项目背景.....	4
1.3 定义.....	4
1.4 参考资料.....	5
2. 总体设计	6
2.1 需求规定.....	6
2.2 运行环境.....	8
2.3 基本设计概念及处理流程	13
2.4 需求与程序关系矩阵.....	26
2.5 人工处理过程	26
2.6 尚未解决的问题	26
3. 接口设计	27
3.1 用户接口.....	27
3.2 外部接口.....	27
3.3 内部接口.....	28
4. 运行设计	29
4.1 运行模块组合	29
4.2 运行控制.....	30
4.3 运行时间.....	30
5. 系统数据结构设计	31
4.1 逻辑结构设计要点	31
4.2 物理结构设计要求	31
4.3 数据结构与程序关系.....	31
6. 系统出错处理设计	32
6.1 出错信息.....	32
6.2 补救措施.....	32
6.3 系统维护设计	32

1. 引言

1.1 编写目的

为更好的规划、设计路演天下系统，为本系统所有相关人员理解系统的架构特性和系统性能，为项目经理把控项目、制定工作计划提供支持，为开发人员提供技术方向，为测试人员提供黑盒、白盒测试依据，故编写引文档。

本文档为系统架构设计文档，文档内容及格式符合 GB-8567 国家标准的相关要求。为系统开发提供技术上的提导，主要读者为项目负责人、系统设计人员、系统开发人员、系统测试人员以及证券信息各业务部门、运维部门、二次开发人员。

1.2 项目背景

项目/系统名称：路演天下系统

项目发起人：深圳市证券信息有限公司

项目实施团队：深圳市大展信息技术有限公司

系统用户：证券信息公司用户、路演企业、互联网用户

全景网是中国资本市场信息服务领域的龙头企业。公司构建起以数据库为基础、以信息披露为核心、集网站、电视、广播、杂志、新媒体终端为一体的多层次资本市场跨媒体信息传播体系，形成了巨潮资讯网、全景网、新财富杂志、财富天下数字电视频道及系列财经电视节目、深证系列指数、巨潮系列指数等财经知名品牌。

全景网旗下的路演频道通过互联网方式，为广大投融资方提供了一个演示产品、推介理念的路演平台，促进了投资者与上市公司之间的沟通和交流。在新的时代下，为响应“大众创业，万众创新”国家战略号召，支持多层次资本市场建设，促进资本市场的繁荣与发展，提高公司治理透明度，建立顺畅的投资者关系管理通道，现根据全景网业务发展需要，对原有的路演业务和投资者关系互动业务进行整合与提升，改版路演频道，以全新的“路演天下”品牌，服务资本市场的广大企业和机构投资者。

1.3 定义

缩写、术语	解 释
-------	-----

路演	<p>路演（包含但不限于证券领域）是指在公共场所进行演说、演示产品、推介理念，及向他人推广自己的公司、团体、产品、想法的一种方式。</p> <p>路演（Roadshow）最初是国际上广泛采用的证券发行推广方式，指证券发行商通过投资银行家或者支付承诺商的帮助，在初级市场上发行证券前针对机构投资者进行的推介活动。是在投资、融资双方充分交流的条件下促进股票成功发行的重要推介、宣传手段，促进投资者与股票发行人之间的沟通和交流，以保证股票的顺利发行，并有助于提高股票潜在的价值。</p>
奥点云	<p>奥点云隶属于杭州奥点科技有限公司，公司成立于 2006 年，拥有超过 10 年的互联网音视频开发经验。专注为网络媒体电台、电视台、在线教育、游戏直播、社交娱乐等新媒体行业提供专业的云媒体云服务。</p>
OSS	<p>OSS（Open Storage Service）开放云存储服务，是阿里云对外的提供的海量，安全和高可靠的云存储服务。</p>
SLB	<p>SLB（Server Load Balancer）是阿里云对多台云服务器进行流量分发的负载均衡服务。SLB 可以通过流量分发扩展应用系统对外的服务能力，通过消除单点故障提升应用系统的可用性。</p>
ECS	<p>阿里云提供的云服务器（Elastic Compute Service，简称 ECS）是一种简单高效、处理能力可弹性伸缩的计算服务，帮助您快速构建更稳定、安全的应用，提升运维效率，降低 IT 成本，使您更专注于核心业务创新。</p>
...	

1.4 参考资料

1. 路演天下软件需求规格说明书；
2. 路演天下交互原型设计文档；
3. GB 8566 《计算机软件产品开发规范》
4. GB 8567 《计算机软件文档编制规范》
5. GB/T 11457 《软件工程术语》

2. 总体设计

2.1 需求规定

2.1.1 路演互动：

在原有 A 股上市公司路演的基础之上，增加香港/海外上市、新三板企业、创业企业、券商机构、公募基金、私募基金等企业主体，扩大平台的服务主体范围。标准化路演厅和路演活动页面，整合路演和交流业务，增强路演过程中的互动感。

2.1.2 路演直播：

实现客户从注册、实名认证到申请路演、路演直播的全流程自主管理。强化对路演直播的控制，对路演过程中的视频、文档及互动交流能灵活配置。

2.1.3 资讯：

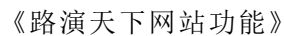
与全景网资讯频道对接，实现资讯信息的快速发布展示，可按路演公司进行资讯的精准投放。

2.1.4 一城一站：

实现按城市管理和展示区域内的上市公司和路演厅，突出地方形象。在建设初期，遵循“先易后难，先简后繁”的原则，分阶段来实现目标。

2.1.5 路演互动：

为认证路演企业开设展示、销售产品的通道，与路演企业联合举办回馈投资者的活动，实现与聚上汇电商平台的数据共享及对接。





《路演天下网站后台功能》

2.2 运行环境

名称	说明	备注
浏览器	IE8 及 IE8 以上浏览器	
HTTP 负载均衡服务器(必须)	自有主机托管阿里云/电信 IDC 机房 操作系统：Red Hat 6.3 64bit 负载均衡软件：Nginx 1.9.12 + Keepalived 反向代理：HTTP	方案一
	阿里云 SLB	方案二（推荐）

官网应用服务器 (必须,多节点集群)	自有主机托管阿里云/电信 IDC 机房 操作系统： Red Hat 6.3 64bit 应用服务器： Apache Tomcat 7.0.42 JAVA 运行时环境：jdk1.7	方案一
	云服务器：阿里云 ECS 操作系统： CentOS 64bit 应用服务器： Apache Tomcat 7.0.42 JAVA 运行时环境：jdk1.7	方案二（推荐）
管理后台应用服务器(必须,多节点集群)	自有主机托管阿里云/电信 IDC 机房 操作系统： Red Hat 6.3 64bit 应用服务器： Apache Tomcat 7.0.42 JAVA 运行时环境：jdk1.7	方案一
	云服务器：阿里云 ECS 操作系统： CentOS 64bit 应用服务器： Apache Tomcat 7.0.42 JAVA 运行时环境：jdk1.7	方案二（推荐）
互动交流应用服务器(必须,多节点集群)	自有主机托管阿里云/电信 IDC 机房 操作系统： Red Hat 6.3 64bit 应用服务器： Apache Tomcat 7.0.42	方案一

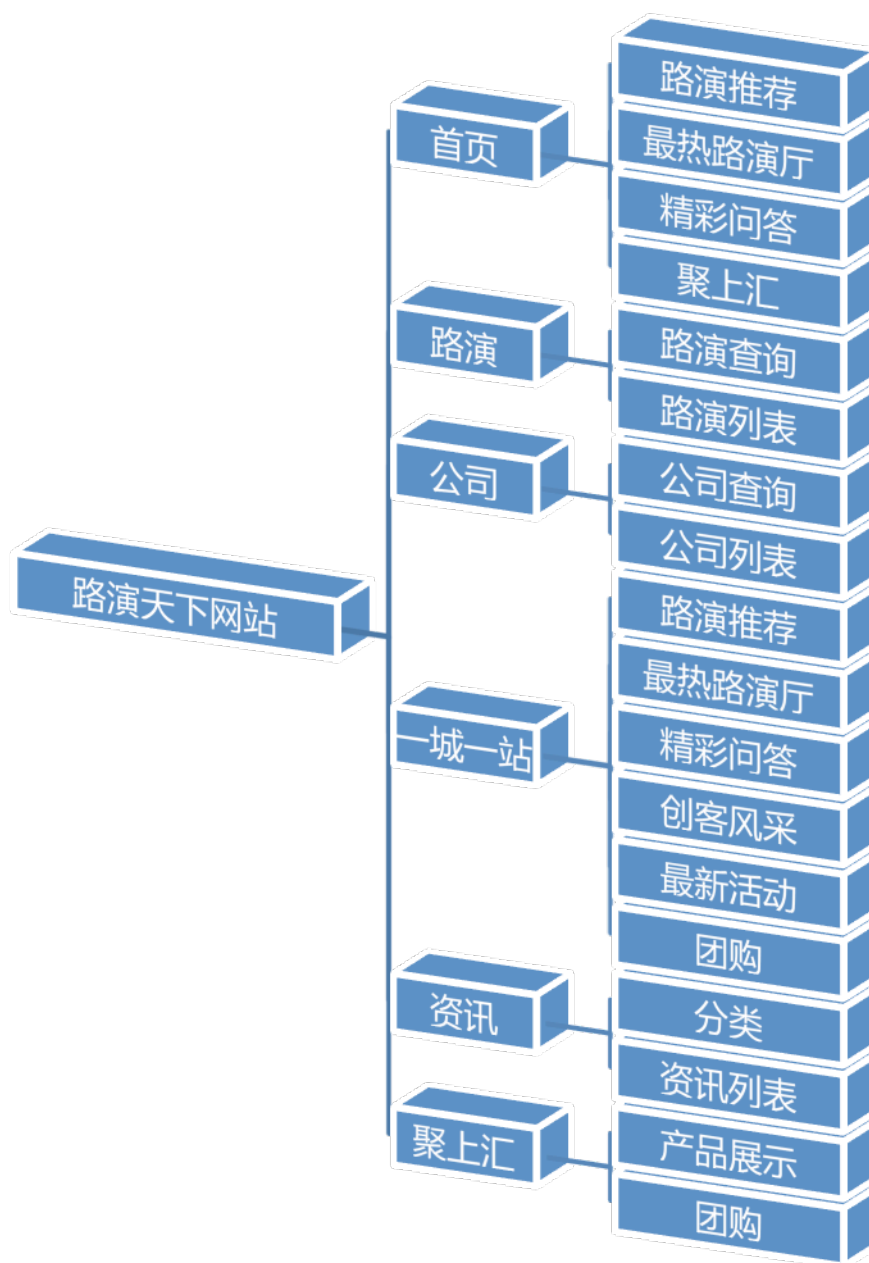
	JAVA 运行时环境：jdk1.7	
	云服务器：阿里云 ECS	方案二（推荐）
	操作系统：CentOS 64bit	
	应用服务器：Apache Tomcat 7.0.42	
	JAVA 运行时环境：jdk1.7	
对外接口应用服务器(非必须)	自有主机托管阿里云/电信 IDC 机房	方案一
	操作系统：Red Hat 6.3 64bit	
	应用服务器：Apache Tomcat 7.0.42	
	JAVA 运行时环境：jdk1.7	
	云服务器：阿里云 ECS	方案二（推荐）
	操作系统：CentOS 64bit	
	应用服务器：Apache Tomcat 7.0.42	
	JAVA 运行时环境：jdk1.7	
静态资源服务器(非必须)	自有主机托管阿里云/电信 IDC 机房	存放静态化 HTML 页面、页面资源文件
	操作系统：Red Hat 6.3 64bit	
	HTTP 服务器：Nginx1.9.12 + Keepalived	
	静态资源服务器容灾：主从备份	
对象服务器(必)	自有主机托管阿里云/电信 IDC 机房	方案一

须)	操作系统： Red Hat 6.3 64bit 支持 FTP, NFS 协议	
	阿里云 OSS 服务	方案二（推荐）
缓存服务器（必须）	自有主机托管阿里云/电信 IDC 机房 操作系统： Red Hat 6.3 64bit 缓存服务器软件： Redis 3.0.7	方案一
	阿里云 Redis 服务 自带主从双节点、硬件自测和恢复	方案二（推荐）
数据库服务器（必须）	自有主机托管阿里云/电信 IDC 机房 操作系统： Red Hat 6.3 64bit 数据库软件： MySQL 5.6 容灾： Master/Slave	方案一（推荐） 与应用服务器在同一数据中心 ,通过内网直接通讯
	阿里云 RDS 服务 数据库服务： MySQL 5.6	方案二
多媒本服务器（必须）	奥点云	
UCenter 单点登录服务器（必须）	自有主机托管阿里云/电信 IDC 机房 操作系统： Red Hat 6.3 64bit 应用服务器： Apache HTTP Server	方案一

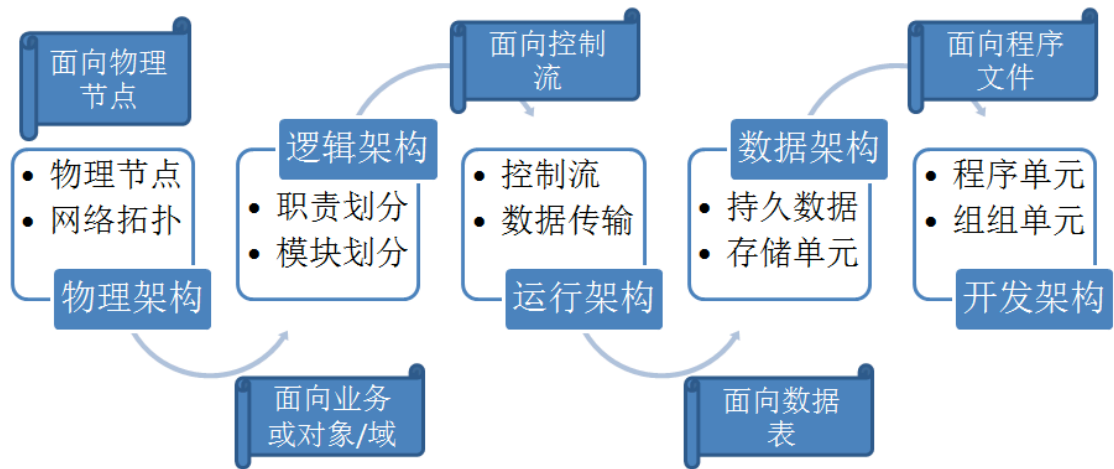
	阿里云 ECS 操作系统：CentOS 64bit 应用服务器：Apache HTTP Server	方案二（推荐）
--	---	---------

2.3 基本设计概念及处理流程

2.3.1 系统组成

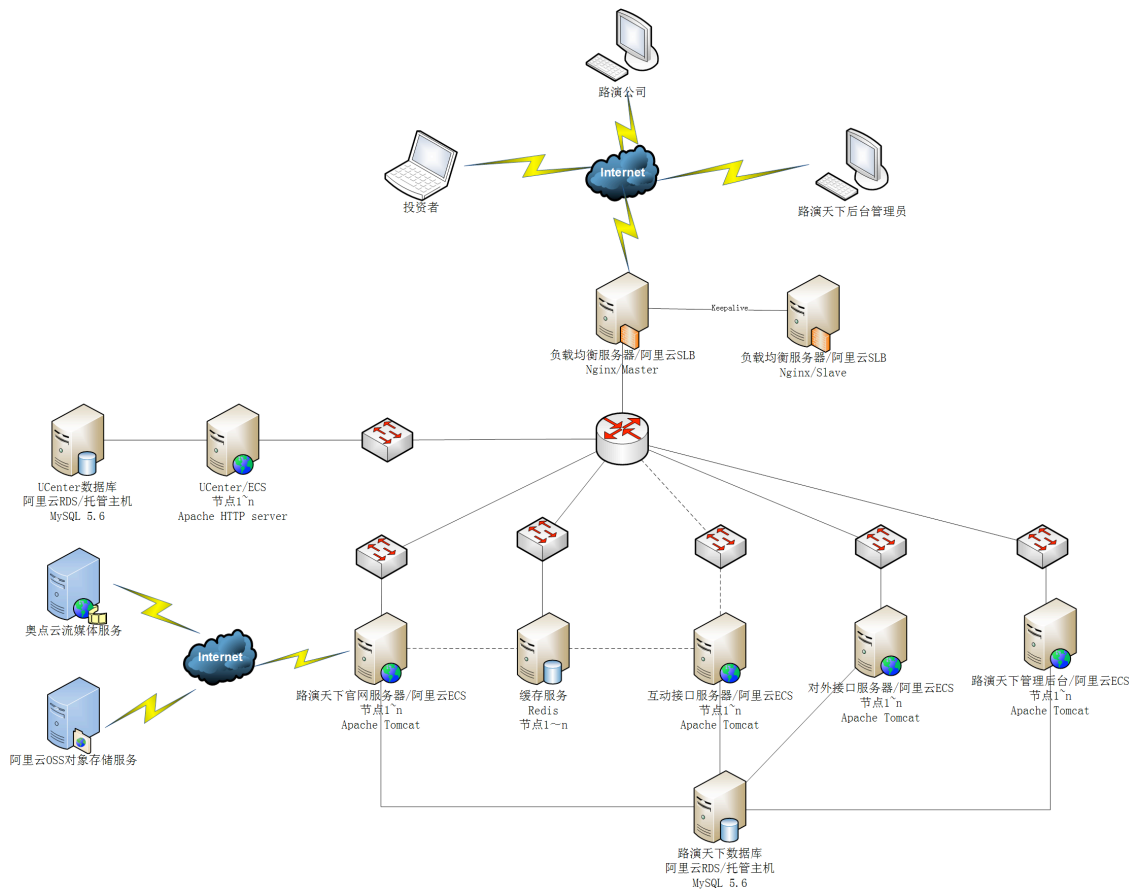


2.3.2 架构描述



名称	说明	备注
物理架构	描述系统的物理节点，系统整体的部署结构、网络拓扑，展现个节点之后间相互关系；	
逻辑架构	描述系统的逻辑分层，说明各个层级的作用与依赖，构建完整的业务系统体系；	
运行架构	描述数据传输控制模式，以及数据流通过程中的转换机制；	
数据架构	描述数据持久机制、存储方案、备份方案；	
开发架构	描述系统程序组织结构、文件结构、命名规则；	

2.3.3 物理架构



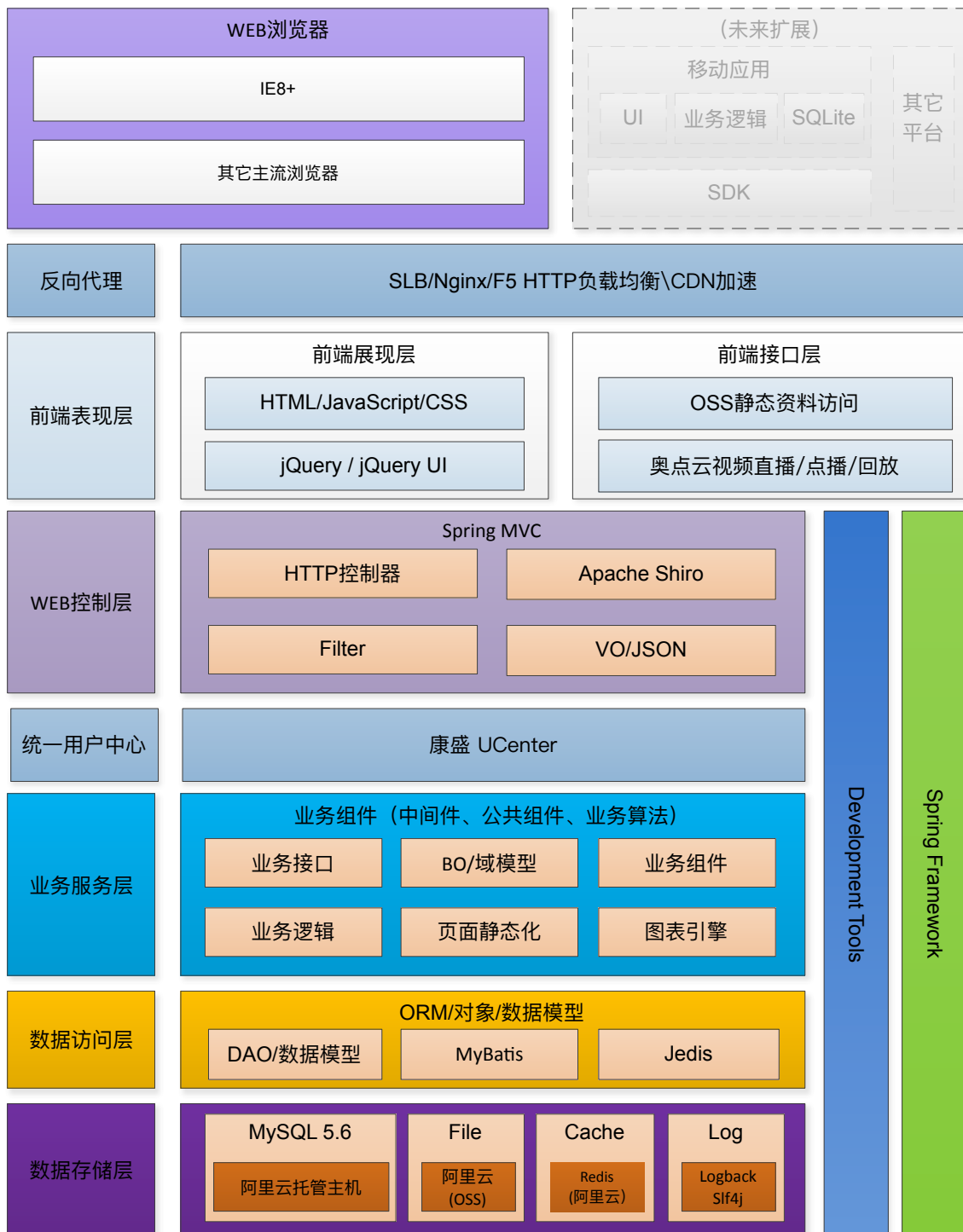
名称	说明	备注
负载均衡服务器	直接面向终端用户，分流 HTTP 请求，分发请求到不同的业务服务器节点，负载均衡服务负责对 WEB 请求的分发，可选方案为软件负载 Nginx 或 F5 硬件负载，建议采用阿里云 SLB 方案。	
路演天下官网	服务以 WEB 应用进行部署，应用服务器为 Apache Tomcat，服务器建议选择同一数据中心的阿里云 ECS 服务，2 个 ECS 节点以上可以开启 SLB 负载均	

	衡服务。	
路演天下管理后台	与路演天下网站同样，建议独立于路演天下网站进行单独部署，应用服务器为 Apache Tomcat, 可以根据后台管理人员的用户量决定是否采用多节点集群，建议服务器采用同一数据中心的阿里云 ECS 服务。	
路演天下互动交易服务器	服务以 WEB 应用进行部署，应用服务器为 Apache Tomcat，服务器建议选择同一数据中心的阿里云 ECS 服务，2 个 ECS 节点以上可以开启 SLB 负载均衡服务。	
路演天下对外接口服务器（不在本期开发范围，仅表示未来扩展）	服务以 WEB 应用进行部署，应用服务器为 Apache Tomcat，服务器建议选择同一数据中心的阿里云 ECS 服务，2 个 ECS 节点以上可以开启 SLB 负载均衡服务。	
UCenter 服务器	提供集中用户管理服务，并支持路演天下网站、管理后台通过 UCenter 实现单点登录功能，服务以 WEB 应用进行部署，UCenter 是康盛公司的一款产品，开发语言为 PHP，因此应用服务器采用	

	Apache HTTP Server 提供基于 PHP 的 HTTP 服务。	
路演天下应用 数据库	<p>路演天下网站和管理后台的核心数据为，由于本项目是一个互联网产品，因此选择 MySQL 5.6 为本系统的关系型数据库系统。</p> <p>考虑到证券信息公司在阿里云有托管自己的服务器主机，以及网站数据的自主可控性，建议路演天下的应用数据库在托管于阿里云的服务器上自建。</p>	
UCenter数据 库	<p>由于 UCenter 是一个集中用户平台，将来还有可能为其它应用提供服务，因此 UCenter 的数据库单独部署。</p> <p>同样建议 UCenter 的数据库在托管于阿里云的服务器上自建。</p>	
奥点云流媒体 服务	提供视频自播、点播、回复服务，该用应用由奥点云提供，具体服务器配置见奥点云要求。	
对象存储服务	存储路演天下静态业务资源，包括业务过程中产生的图片、文件等。建议采用阿里云 OSS 服务，对于静态资源的访问不需要经过 WEB 应用服务器，节省 WEB 应用服务器的网络 I/O。	
缓存服务器	使用 Redis 作为缓存服务器，用来缓存业务处理过	

程需要暂时存储的数据，以及集群环境下的 HTTP 会话信息的共享，建议采用阿里云提供的 Redis 服务。

2.3.4 逻辑架构



名称	说明	备注
反向代理	利用 Nginx、阿里云 SLB、F5 实现负载均衡，以	

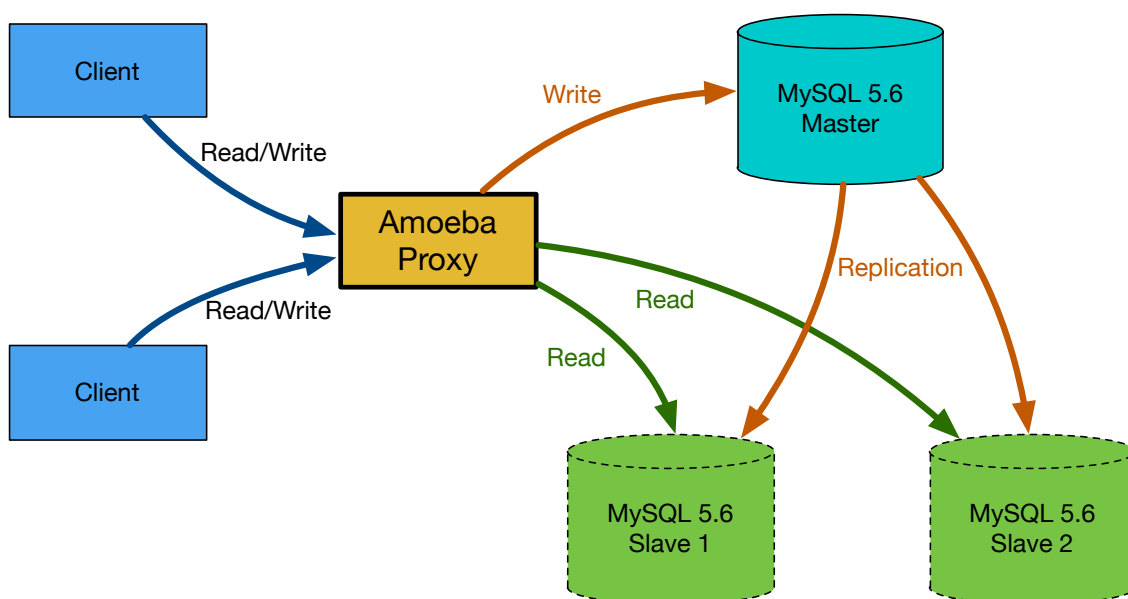
	实现物理节点的横向扩展。	
前端表现层	以前端展示为主，包括用户数据接收、系统数据展示，界面布局、排版、UI 美化，所有用到的框架、组件，如 jQuery、jQuery UI 等；	
前端接口层	静态资源分离，直接从 OSS 上获取，避免占用 WEB 应用服务器的资源； 流媒体资源直接从奥点云获取、推流，前端流媒体播放器。	
WEB 控制层	提交界面数据进行封装，控制转发，调用后台业务服务器，与服务端进行通讯； 接收服务端业务处理结果进行数据封装，控制转发展示界面；	
服务实现层	面向服务/业务的接口定义，数据接口定义，是服务端与 WEB 前端口口的交互点，服务接口的具体实现；处理业务逻辑，界面静态化处理，图表引擎	
数据访问层	访问数据库，业务数据操作的持久；	
数据存储层	存储业务活动发生的业务数据、系统数据；缓存业务临时数据，记录日志。	

2.3.5 运行架构

2.3.5.1 数据流



2.3.6 数据库架构



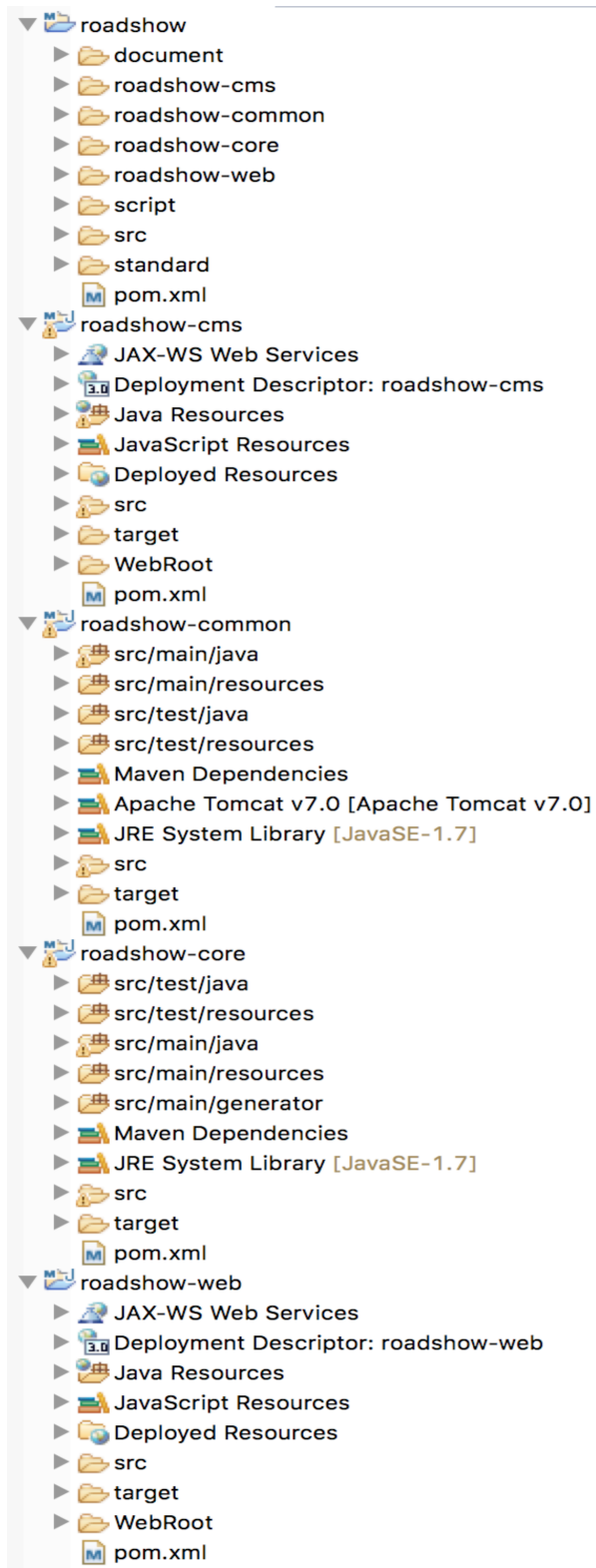
《MySQL 读写分离方案》

由于路演天下是一个互联网项目，用户量将会非常大，数据库要实现读写分离，提升性能，代理使用 Amoeba，配置读写分离策略。

2.3.7 开发架构

项目采用 Maven 进行构建，共分为一个父项目，四个子项目，具体如下：

1. 工程结构：



名称	说明	备注
----	----	----

roadshow	Maven 主项目，pom 类型。	
roadshow-common	Maven 子项目，通用算法、工具，非业务类处理，jar 类型。	
roadshow-core	Maven 子项目，Service、Controller 层父类，通用业务处理，jar 类型。	
roadshow-cms	Maven 子项目，WEB 工程，路演天下管理后台。	
roadshow-im	Maven 子项目，WEB 工程，交流互动服务。	
roadshow-api	Maven 子项目，WEB 工程，对外接口服务。	
roadshow-web	Maven 子项目，WEB 工程，路演天下官网。	

2.3.10 监控日志

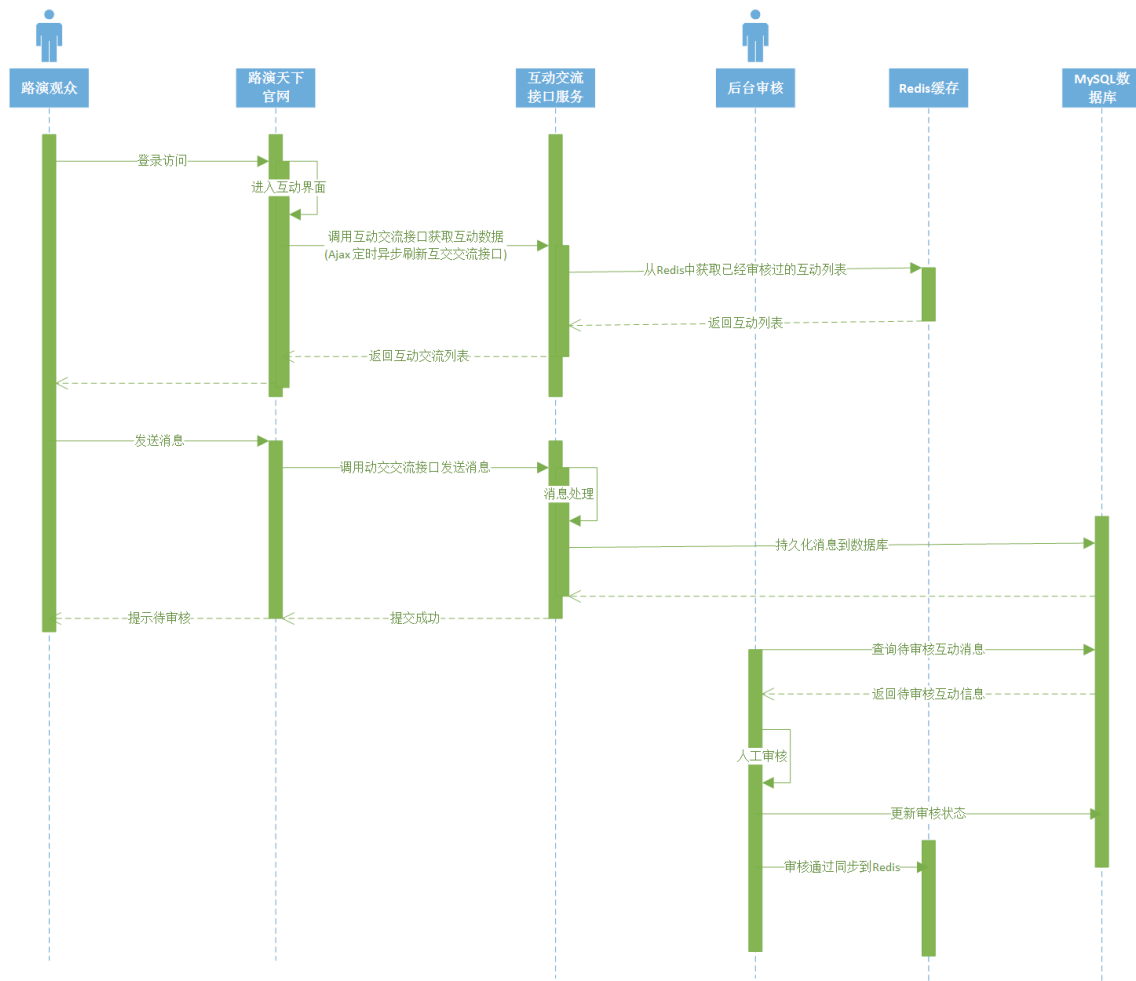
监控系统实际上是一个非常复杂和庞大的工程，通常来说一个组织的所有业务系统应该有一套独立的监控系统来负责，监控系统通常不仅要监控业务系统本身的信息，还要监控业务系统操作系统、硬件环境、网络环境、安装环境等多方面的信息，鉴于股转公司目前没有这方面的监控系统，因此本项目根据实际业务情况，主要以监控系统日志来构建系统的监控方案。

通常来说，系统日志分为运行日志和业务日志：

1. 运行日志通常是指系统在运行过程中记录的信息，通常以平面文件或者日志服务器的形式存储运行日志，本系统运行日志主要采用 Logback 和 Slf4j 这两个开源框架进行记录，运行日志的存储载体为平面文件；日志文件以每天产生一个新文件的方式避免日志文件过大；运行日志的记录分为 4 个级别，分别是 ERROR 级别、WARNING 级别、INFO 级别、DEBUG 级别，每个级别的日志分别输出到各自的日志文件中，应用程序在生产环境可以根据需要来设置输出哪一个级别的日志信息，其中 DEBUG 主要用在开发、测试环境，而 ERROR 日志输出最少。
2. 业务日志通常是指具体的操作时，记录操作痕迹，业务日志最终记录到当前业务系统的数据库表中，一般来说记录的业务日志最终能完整描述业务操作的情形，例如：某人在某时以某种方式成功或失败的操作了某个模块下的某个业务功能；业务日志的记录通常由程序代码来完成，主要有两种记录方式，第一种是基于事务级的记录；另一种是基于数据库表的记录；本系统中，业务日志的记录会结合这两种记录方式，原则上基于事务进行记录，部分业务细化到基于数据库表的记录；业务日志的记录可以分别记录针对数据库的增加、修改、删除来记录，同时针对部分敏感数据的查询也会记录业务操作日志。
3. 日志记录的控制
 - A. 运行日志通过修改 Logback 的配置文件来控制日志输出级别，修改后需要重启应用服务器；在开发过程中，在需要记录日志的代码块中，判断日志的级别之后输出相关的日志。
 - B. 业务日志通过系统配置参数在线修改日志输出的业务操作类型，修改之立即生效。
4. 日志数据查看
 - A. 运行日志的查看通常需要将应用服务器上生成的日志文件下载到本地，利用文本工具打开查看；
 - B. 业务日志的查看通过在线的日志管理模块查看、检索日志信息。

2.3.11 文字互动

根据项目的特点，文字互动这一部分将会是本系统性能的一个瓶颈，为了提高性能，文字互动的功能抽取出来形成一个独立的服务，并对互动服务进行多节点负载均衡，从而提高性能。



[illegible]

无

无

3. 接口设计

3.1 用户接口

终端用户直接访问静态页面，服务器直接返回 html 页面和动态数据转发 Web API 服务器，动态数据采用 JSON 数据格式返回到前端页面中，通过 AJAX 解析后展示在页面上。

3.2 外部接口

3.2.1 奥点云流媒体服务接口

本系统有关视频直播、点播、回放、PPT 播放的功能均调用奥点云提供的接口，奥点云提供视频推流、拉流服务，视频文件存储服务，PPT 上传、播放服务、浏览器控制本地摄像头功能。

3.2.2 短信平台接口

本系统有关短信验证码、短信通知均统一调用短信平台的接口来完成。

3.2.3 UCenter 统一用户中心接口

本系统用户统一使用 UCenter 进行集中管理，并使用 UCenter 实现单点登录功能。

3.2.4 聚上汇电商平台接口

本系统有关电商部分的商品展示、团购、订单查询，统一调用聚上汇电商平台提供的接口来实现，本系统不存储任何与电商平台相关的数据，仅做展示。

3.2.5 深交所互动易问答接口

本系统有关投资人通深交所互动易平台提交的问答数据需要能够在路演天下网站中展示，这一部分问答数据需要调用深交所互动易平台提供的接口进行获取。

3.2.6 第三方登录接口

本系统提供微信、QQ、新浪微博 3 种第三方用户授权登录接口实现快速登录，第三方登录接口统一使用 OAuth 2.0 标准。

3.3 内部接口

3.3.1 业务访问接口

本系统是一个 WEB 系统，所有业务访问均通过 HTTP(s)协议进行，未来对外提供的业务接口也基本上为 HTTP 接口，不提供 WebService、TCP 接口。

3.3.2 数据访问接口

系统自身对于数据访问时，使用统一的数据库账号，采用 JDBC Thin 模式进行通讯，应用层采用 MyBatis 进行 ORM 映射，所有分页数据均基于数据库层进行处理。

对于敏感数据在存储时，进行加密保存，对于重要业务数据的操作自动记录日志，便于数据审计。

为了保证数据安全，禁止直接对外接口基于数据库层面的接口，如需对外提供接口，要通过程序来提供。

3.3.3 分层调用接口

系统的程序结构方面，严格分层，层与层之间解耦，层与层之间的通讯原则上必须通过接口来实现，不允许在上一层直接调用下一层的实现类。

接口业务动作（方法）的命名按照越往上越面向业务、面向用户，越往下越面向数据进行命名。

各层之间的调用尽可能的不出现“new”的方式来实例化，原则上采用 Spring 的 IoC 注入技术，以注解和配置相结合的方式实现 Spring 的管理。

3.3.4 子系统接口

1. 互动交流服务接口：

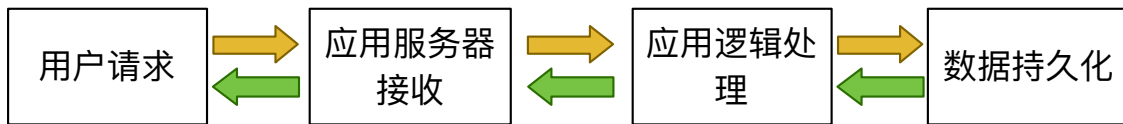
路演天下官网 HTML 页面只接调用互动交流服务提供的 HTTP 接口，接口功能包括查询已审核的互动信息、发布互动消息；

2. 对外服务接口：

本阶段无对外接口提供，但为了将来的扩展，提供对外接口接入点供后续二次扩展。

4. 运行设计

4.1 运行模块组合



名称	说明	备注
用户请求	<p>前端用户以浏览器访问系统，通过 HTML 页面完成用户操作，业务数据转换成 JSON 传输给后台。</p> <p>后台处理好业务请求之后，将响应数据封装成 JSON 返回到 HTML 页面进行展示。</p>	
应用服务器接收	<p>应用服务器提供 HTTP 服务，接收用户请求，当用户请求到达应用服务器时，由应用服务器控制路由选择，调用具体的业务处理逻辑。</p> <p>业务处理逻辑完成并返回消息之后，再控制响应路由的选择，导向处理结果页面。</p>	

应用逻辑处理	原子业务为最小逻辑单元 ,由多个原子业务单元组合形成一个业务领域的完整业务处理逻辑 ,业务处理逻辑以多种形式存在 ,包括数据计算、外部接口调用、数据转发等。	
数据持久化	对业务逻辑处理的原数据和结果数据进行持久化存储 ,持久化存储的形式包括关系型数据库、平面格式文件、非关系型数据库的存储和数据提取。	

4.2 运行控制

系统运行期间，不允许人为控制各逻辑层的流程，每一层之间的跳转必须由程序根据上一节点的处理结果参数进行控制，系统的数据源头来自提交业务提交的数据，其它数据均为中间数据。

4.3 运行时间

1. 服务端确保 7×24 小时稳定可靠运行。提供系统崩溃时的快速恢复机制，确保系统出现故障时能快速恢复系统正常运行；
2. 操作系统的稳定性；
3. 数据连接的稳定性
4. 数据层采用连接池技术，并使用连接池提高系统性能，对每个连接的申请和归还跟踪监视，严格保持连接数量在系统性能的允许范围之内。

5. 系统数据结构设计

4.1 逻辑结构设计要点

本系统从以下两方面考虑设计：

1. 动态变化：系统栏目及下级栏目为动态的，且从扩展的角度来考虑，把栏目增加，文档需要的内容设计为模型原形，可以十分方便的动态增加栏目和下级栏目，更可以随时修改文档的输入内容和字段。
2. 存储高效及结构简明：栏目和文档数据将被持久化为静态文件，访问和读写频率不是太频繁，非常适合静态化，数据访问时效性也有延迟，故，设计时主要考虑存储的高效性。当数据被加载后，更快速的访问到相应的数据及减少对数据库的访问，非常有利于系统的用户体验。
3. 数据字典：系统具体数据字典功能，对下拉框的选项，以及可以采用字典维护的业务数据统一纳入数据字典管理。
4. 业务功能抽象：业务功能的原子化，以最少功能进行编码设计，组合成一个完整的业务模块；通用功能、相似逻辑抽取为方法、类，减少重复代码的存在。
5. 允许适当数据冗余：为了提升系统的整体性能，对于部门关联性不是很强的数据，允许在数据库中存在多份。

4.2 物理结构设计要求

根据用户非功能需求约束，本系统物理结构设计满足以下要点：

1. 应用之间独立部署；
2. 静态资源共享；
3. 系统支持集群环境运行。

4.3 数据结构与程序关系

略。

6. 系统出错处理设计

6.1 出错信息

故障现象	用户展示	处理机制
页面不存在	404 错误	发送更新文件消息到消息服务器，触发该页面更新。
服务错误	提示服务错误，并重试	记录服务错误日志，发送消息到管理员。
数据库宕机	前端无影响	切换备库，发送消息重起服务器。
网络中断	提示服务中断	多站点服务，数据分发。

6.2 补救措施

说明故障出现后可能采取的变通措施，包括：

1. 后备技术，由于服务器的元件在使用过程中有机会出现故障，建议在核心业务节点均备用一台，随时可以切换到备用节点；
2. 降效技术，由于系统的运行受硬件环境、用户环境的影响，而对性能、响应产生变化，当用户急增导致系统无法承受时，可以考虑优化业务办理流程，分时段操作系统，以及增加硬件投入来应急；
3. 恢复与安装，系统恢复主要分为两个层面，第一数据层面，在运维过程中，应制定数据备份方案与计划，并按计划执行数据备份，当系统数据库出现故障时，可以能过数据库厂商推荐的恢复方案以最近备份数据为蓝本进行恢复； 第二程序层面，通常要求在升级新系统之前，全量备份旧系统，如果新系统安装失败，应从旧系统还完程序。

6.3 系统维护设计

系统维护是一个软件系统生命周期中重要的一环，本系统也没例外：

1. 运行维护，通常包括制定运维方案与计划，主要包括系统监控、备份、审计、优化等；

2. 升级维护，通常涉及到功能变化，一般都需要修改源程序，一般这类的升级都会影响到业务操作，因此每次升级程序，应该得到组织的审核和风险评估，升级之前全量备份旧系统，升级完成之后投入使用之前，应该做好健康性检查，并更新相关文档。