



单军军

前端开发工程师

## 教育经历

男 / 1995.09

郑州大学 · 计算机科学与技术

本科 / 2017年毕业

## 专业技能

HTML

CSS

Less

Sass

JavaScript

TypeScript

Vue.js

React

AngularJS

RabbitMQ

Nginx

MySQL

Redis

## 产品经验

2019-05 ~ 2019-07

[Link](#)

### Soil-VMware纳管

产品市场调研与反馈总结 产品设计 前端架构选型与设计 后端服务架构选型与设计 前端研发 后端服务研发 CI/CD构建

该产品其最初目标是为 **企业用户** 基于 **OpenStack** 架构纳管第三方云计算产品，实现操作入口、用户管理、使用习惯的整合，通过开源架构管理所有云资源。从而实现更好的服务、效率和管理的一体化运营。

该产品采用了 **热插拔** 的架构设计，既可以整合到OpenStack整个云产品中进行发布，同时也可以作为独立应用进行单独发布。

该产品前端服务实现了 **Horizon** 功能的 **完全插件化开发**，实现原理是Horizon允许添加dashboards(仪表盘)，panels(面板)和panels group(面板组)，而无需修改默认设置。

该产品实现了 **soildient(API的外层需要一重转化机制)**，通过soilclient让soil产品一般的用户和管理员能够通过执行简易的 **Soil Commands** 来管理和使用Soil。

该产品后端服务借鉴了OpenStack **Nova模块** 的 **API**、**DB**、**WSGI服务**、**RPC调用**，纳管VMware采用了 **pymomi** 这款SDK来实现了VMware **虚拟机**、**存储** 以及 **网络** 等资源的管理。

该产品代码管理工具使用的是 **gitlab**，rpm打包工具(**SPEC** 配置规则文件的编写)使用的是 **Jenkins**，CI/CD(持续集成、持续部署)工具成功的构建并且与内部 **LDAP** 打通，缩短了 **40%** 左右的开发周期。

该产品 **Version 1.0.0** 版本的设计、研发到发布时间节点总共没有超过2个月，**POC** 服务了 **10家大型企业**，**交付了 3家企业**，产品涉及了包括 **金融**、**能源**、**制造业** 等领域，简化了 **30%** 企业用户基于OpenStack云平台管理其他云平台的操作。为公司带来了10W的实际营收以及100W的预计营收。

2019-10 ~ 2019-11

[Link](#)

### Hamal-V2V迁移

产品市场调研与反馈总结 产品设计 前端架构选型与设计 缓存架构选型与设计 后端服务架构选型与设计 前端研发 缓存研发 后端服务研发 CI/CD构建 应用标准镜像制作

该产品其目标是为了 **企业用户** 更加方便地整合云平台资源，基于 **Red Hat virt-v2v** 架构迁移 **第三方云计算产品** 到企业 **特定的云计算产品** 中，实现了 **操作入口**、**用户管理**、**使用习惯** 的整合，通过开源架构平滑无感迁移所有第三方云计算资源。从而实现更好的服务、效率和管理的一体化运营。

该产品实现了以 **微服务应用** 的架构进行实现，即可以将做好的镜像直接进行发布，同时也可以将迁移应用的rpm包部署到特定的系统(目前是Centos 7)上。

该产品前端架构采用了 **React**、**ReactDOM**、**React Router**、**Redux**、**DvaJS**、HTTP请求采用了 **Axios**，UI采用了 **Ant Design**。

该产品Web服务架构采用了 **NGINX** 高性能的HTTP和反向代理web服务器,同时也提供了IM API/POP3/SMTP服务。

该产品缓存架构采用了 **Redis** 一款开源的使用ANSI C语言编写、支持网络、可基于内存亦可持久化的日志型、Key-Value数据库，并提供多种语言的API。

该产品后端服务借鉴了OpenStack **Nova模块** 的 **API**、**DB**、**WSGI服务**、**RPC调用**，纳管VMware采用了 **pymomi** 这款SDK来实现了VMware **虚拟机**、**存储** 以及 **网络** 等资源的管理。

该产品代码管理工具使用的是 **gitlab**，rpm打包工具(**SPEC** 配置规则文件的编写)使用的是 **Jenkins**，CI/CD(持续集成、持续部署)工具成功的构建并且与内部 **LDAP** 打通极大程度缩短了开发周期。

该产品目前的标准镜像格式是 **QCOW2** (压缩)和 **RAW** (未压缩)两种格式，镜像系统是标准的 **Centos 7** 系统。

该产品 **Version 1.0.0** 版本的设计、研发到发布时间节点总共没有超过1个月，**POC** 服务了 **5家大型企业**，**交付了 2家企业**，产品涉及了包括 **金融**、**能源** 领域，简化了40%企业用户迁移第三方云计算资源到特定云平台的操作。为公司带来了5W的实际营收以及20W的预计营收。

## 精选项目经验

Git

Linux

数据结构

算法

联系方式

 CSDN博客  
 github.com/jackdan  
 j.dan92016@gmail.com  
 13027710816

2019-05 ~ 2019-07

[Link](#)

入网安评·南方电网

[入网安评·验证码](#) [入网安评·密码防止暴力破解](#) [入网安评·多Session登录限制](#) [入网安评·新创建第一次登陆的用户强制修改密码](#) [入网安评·RSA不可逆加密用户名与密码信息](#)

该项目实现的功能解决了50%基于OpenStack私有云平台入网安评·安全评测功能。

该项目引入[Captcha](#)去实现了验证码功能，基于[OpenStack](#)私有云计算平台实现的[热插拔](#)验证码功能。

该项目实现了用户输入密码次数超过[指定的次数](#)就等待[指定的时间](#)·密码防止暴力破解，基于[OpenStack](#)私有云计算平台实现的[热插拔](#)密码防止暴力破解功能。

该项目实现了[同一用户只允许在同一PC端登录](#)·多Session登录限制，基于[OpenStack](#)私有云计算平台实现的[热插拔](#)多Session登录限制功能。

该项目实现了基于云计算平台[新创建的用户登陆时强制提醒修改密码](#)，基于[OpenStack](#)私有云计算平台实现的[热插拔](#)新创建第一次登陆的用户强制修改密码。

该项目实现了基于云计算平台[用户登录时用户名和密码私钥加密](#)，到认证时公钥解密，基于[OpenStack](#)私有云计算平台实现的[热插拔](#) RSA不可逆加密用户名与密码信息。

2019-01 ~ 2019-02

[Link](#)

Zabbix监控·人行征信

[Zabbix监控·云计算硬件资源获取](#) [Zabbix监控·云计算虚拟资源获取](#) [Zabbix监控·Ceph OSD资源获取](#)

该项目解决了基于[OpenStack](#)私有云平台监控[内部资源](#) (CPU、Memory、Core、RAM)使用情况以及[Ceph存储健康](#)情况。结合[Shell脚本](#)与[Zabbix](#)将私有云平台的资源监控率提升了[15%](#)。

该项目脚本[hardware-cpu-check.sh](#)实现了基于[OpenStack](#)私有云计算平台获取物理CPU的总量和已使用量，以及对应的[hardware-cpu-check.conf](#) Zabbix配置文件

该项目脚本[hardware-memory-check.sh](#)实现了基于[OpenStack](#)私有云计算平台获取物理Memory的总量和已使用量，以及对应的[hardware-memory-check.conf](#) Zabbix配置文件

该项目脚本[virtual-core-quota-check.sh](#)实现了基于[OpenStack](#)私有云计算平台获取虚拟Core的总量和已使用量，以及对应的[virtual-core-quota-check.conf](#) Zabbix配置文件

该项目脚本[virtual-instance-quota-check.sh](#)实现了基于[OpenStack](#)私有云计算平台获取虚拟Instance的总量和已使用量，以及对应的[virtual-instance-quota-check.conf](#) Zabbix配置文件

该项目脚本[virtual-ram-quota-check.sh](#)实现了基于[OpenStack](#)私有云CORE计算平台获取虚拟RAM的总量和已使用量，以及对应的[virtual-ram-quota-check.conf](#) Zabbix配置文件

该项目脚本[ceph-cluster-capacity-check.sh](#)实现了基于[OpenStack](#)私有云计算平台获取Ceph Cluster的总量和已使用量，以及对应的[ceph-cluster-capacity-check.conf](#) Zabbix配置文件

该项目脚本[ceph-osd-down-num-check.sh](#)实现了基于[OpenStack](#)私有云计算平台获取Ceph OSD Down的数量，以及对应的[ceph-osd-down-num-check.conf](#) Zabbix配置文件

该项目脚本[ceph-osd-stats-check.sh](#)实现了基于[OpenStack](#)私有云计算平台获取Ceph OSD Status的状态描述，以及对应的[ceph-osd-stats-check.conf](#) Zabbix配置文件

2017-01 ~ 2017-06

[Link](#)

React Echarts·数据可视化

[项目设计](#) [界面设计](#) [前端开发](#) [持续升级](#)

该项目解决了React与Echarts结合的问题，以及模块按需加载，浏览加载优化，前后端[完全分离](#)进行开发，提升了[30%](#)的开发效率。

该项目为数据可视化·PC端项目，采用了[React](#)、[ReactDOM](#)、[React Router](#)、[Redux](#)、[MobX](#)、渲染层采用了[Sass](#)，HTTP请求层采用了[Axios](#)、可视化层采用了[Echarts](#)、包构建层采用了[Webpack](#)，按需引入和加载模块，极大程度上优化了代码压缩后的体积，提升了浏览器加载速度，逻辑层采用组件化的开发，项目使用了开放的本地MOCK数据接口。

该项目实现了完全的前后端分离，项目不断地[维护](#)与[重构](#)，[优化用户体验](#)及[性能](#)。

2017-07 ~ 2018-03

[Link](#)

Smart App·智慧城市

[项目设计](#) [界面设计](#) [前端开发](#) [持续优化](#)

该项目解决了移动端原生层构建的问题，用Vue.js开发的页面作为业务页面承载的模式，前后端完全分离进行开发。

该项目为移动端项目，采用了[Vue.js](#)、[Vue Router](#)、[Vuex](#)、[Vue Loader](#)、[Iconfont](#)、[rem自适应](#)，渲染层采用了[CSS](#)，HTTP请求层采用的是[Vue Resource](#)，国际化层采用

了 [Vue 118n](#)，包构建层采用了 [Webpack](#)，移动原生应用层采用了 [React Native](#)，按需引入和加载模块，极大程度上优化了代码压缩后的体积，提升了浏览器加载速度，逻辑层采用组件化的开发，项目使用了开放的本地[MOCK](#)数据接口。

该项目实现了完全的前后端分离，项目不断地[维护](#)与[重构](#)，[优化用户体验](#)及[性能](#)。

---

2017-06 ~ 2017-08

[Link](#)

#### Demo Kugou·酷狗音乐Demo

[项目设计](#) [界面设计](#) [前端开发](#) [持续优化](#)

该项目解决了音乐推荐、音乐搜索以及音乐播放的问题。

该项目为移动端产品，采用了 [Vue.js](#)、[Vue Router](#)、[Vuex](#)，渲染层采用了 [CSS](#)，HTTP请求层采用了 [Vue Resource](#)，UI层采用了 [Mint UI](#)，视图层增加了 [Iconfont](#)、[rem自适应](#)等技术来丰富展示前端界面，HTTP请求层采用的是 [Vue Resource](#)，国际化层采用了 [Vue 118n](#)，包构建层采用了 [Webpack](#)，音乐接口层采用了 [酷狗音乐提供的音乐接口](#)，按需引入和加载模块，极大程度上优化了代码压缩后的体积，提升了浏览器加载速度，逻辑层采用组件化的开发。

该项目实现了完全的前后端分离，项目不断地[维护](#)与[重构](#)，[优化用户体验](#)及[性能](#)。

---

2017-12 ~ 2018-03

[Link](#)

#### CNode Blog·CNode博客

[项目设计](#) [界面设计](#) [前端开发](#) [持续更新、迭代及优化](#)

该项目实现了博客的(各个板块)列表展现，博客详情、评论的查看，新手入门页面，API页面，关于页面以及视频播放页面博客常用的功能。

该项目为响应式多端产品，采用了 [React](#)、[ReactDOM](#)、[React Router](#)、[React Router Dom](#)，渲染层采用了 [CSS](#)，HTTP请求层采用了 [Axios](#)，视频层采用了 [Vdeo.js](#)，包构建层采用了 [Webpack](#)，接口数据采用了 [CNode社区提供的开发接口](#)，按需引入和加载模块，极大程度上优化了代码压缩后的体积，提升了浏览器加载速度，逻辑层采用组件化的开发。

该项目实现了完全的前后端分离，项目不断地[维护](#)与[重构](#)，[优化用户体验](#)及[性能](#)。

---

2018-03 ~ 2018-05

[Link](#)

#### Music App·移动音乐App

[项目设计](#) [界面设计](#) [持续更新、迭代及优化](#)

该项目实现了音乐排行、音乐轮播、音乐搜索以及音乐播放等功能。

该项目是基于微信公众号的移动端Web应用，采用了 [Vue.js](#)、[Vue Router](#)、[Vue Loader](#)，渲染层采用了 [CSS](#)、[Sass](#)，HTTP请求层采用了 [Axios](#)、[Vue Resource](#)，轮播图层采用了 [Vue Swiper](#)，包构建层采用了 [Webpack](#)，视图层采用了 [Flex弹性布局](#)，按需引入和加载模块，极大程度上分化了代码压缩后的体积，提升了浏览器加载速度，逻辑层采用组件化的开发。接口实现基于[qq音乐](#)的音乐接口以及本地开放的[MOCK](#)数据接口。

该项目实现了数据完全的前后端分离。项目不断的维护与重构，优化用户体验及性能，极力去争取找出最规范化的音乐移动端UI设计。

---

### 其他项目经验

---

2017-04 ~ 2017-05

[Link](#)

#### mygirl·爬虫推送信息

[项目设计](#) [架构选型](#) [持续更新、迭代以及优化](#)

该项目实现了指定城市的温度信息、风速信息以及风向信息等天气情况，整理为可自定义的富文本推送到指定人邮箱。

该项目是基于 [Scrapy](#) 框架，爬取Web网站信息将其整理成丰富的富文本页面，在指定的时间推送到指定人的邮箱，完成天气信息提醒到指定人。

该项目会持续更新、迭代以及后期进行分布式的优化。

---

2017-05 ~ 2017-06

[Link](#)

#### mygirl-english·爬虫推送信息

[项目设计](#) [架构选型](#) [持续更新、迭代以及优化](#)

该项目实现了英文短文内容、英文短文视频以及英文短文图片等英文短文情况，整理为可自定义的富文本推送到指定人邮箱。

该项目是基于 **Scrapy** 框架，爬取Web网站信息将其整理成丰富的富文本页面，在指定的时间推送到指定人的邮箱，完成英文短文信息推送到指定人。  
该项目会持续更新、迭代以及后期进行分布式的优化。

2017-06 ~ 2017-07

[Link](#)

mygirl-joke·爬虫推送信息

项目设计 架构选型 持续更新、迭代以及优化

该项目实现了笑话内容、笑话图片等笑话内容情况，整理为可自定义的富文本推送到指定人邮箱。

该项目是基于 **Scrapy** 框架，爬取Web网站信息将其整理成丰富的富文本页面，在指定的时间推送到指定人的邮箱，完成笑话信息推送到指定人。

该项目会持续更新、迭代以及后期进行分布式的优化。

2017-11 ~ 2017-12

[Link](#)

cpu-node·CPU监控

项目设计 界面设计 架构选型 持续更新、迭代以及优化

该项目实现了实时监控CPU的使用情况。

该项目是采用了 **Node.js** 与 **Socket**，视图层采用了 **Chart.js** 进行可视化CPU数据的监控。

该项目会持续更新、迭代以及后期进行分布式的优化。

2018-06 ~ 2018-07

[Link](#)

app-spider·分析视频分析的原链接

项目设计 持续更新、迭代以及优化

该项目实现了获取在即时聊天软件中分享的視頻的原链接地址获取。

该项目是采用了 **Scrapy**、**Request**、**WireShark**，将分析得到的视频原链接整理为JSON数据接口返回。

该项目会持续更新、迭代以及后期进行分布式的优化。

## 职业经历

2018.06 ~ 至今

[软件研发工程师](#)

EasyStack

客户创新部-CIC-软件研发 负责基于**OpenStack**实现的企业私有云计算平台创新性、通用性项目的开发需求、研发实现与交付部署，同时也进行部门产品型工具的调研、架构选型、研发实现与交付部署，此外还负责产品和项目的软件工作分配。

2017.06 ~ 2018.06

[前端开发工程师](#)

郑州拉姆达信息技术有限公司

工程技术部-前端开发工程师 工程技术部-接口负责人 负责项目的前端设计、架构选型以及研发实现，**RESTful API**的设计与实现，并负责项目的研发及维护。

2017.01 ~ 2017-06至今

[前端开发工程师](#)

贵州双龙大数据中心(ddb)

工程技术部-前端开发工程师 负责高度还原UI/UE设计人员设计的设计图，并参与项目的研发及维护。

## 专业技能

▪ 熟练使用 **HTML**、**HTML5**、**CSS2**、**CSS3**、**Less**、**Sass**，能准确还原设计稿。

重视 用户体验。

▪ 娴熟使用 **JavaScript**，**TypeScript**，**Python**。

▪ 熟悉使用 **MMV** 框架 **Vue**、**声明式框架** **React**、**云计算管理平台** **OpenStack**。

▪ 熟悉使用 **Web服务器** **Nginx**、**消息中间件** **RabbitMQ**、**关系型数据库** **MySQL**、**非关系型数据库** **Redis**。

▪ 熟知常用的 **TCP/IP协议**、**UDP/IP协议**、**HTTP协议** 以及 **AMQP协议** 等网络协议。

- 熟练使用 `Git` 进行版本控制和代码托管、`Markdown` 进行文档编写，并以 `Linux`、`VSC ode` 作为日常开发环境进行工作，了解项目开发流程及开发调试工具的使用。
- 了解 `Linux`操作系统常见的命令行操作，例如 `netstat`、`ps` 等，`2` 年以上Linux平台使用经验，熟悉linux下常见应用软件的 `安装`、`配置`、`管理` 及 `简单优化`，如Nginx、memcached等；。

---

感谢您阅读我的简历，希望能与您共事！