

# PYTHON 自动化开发S10

DAY 15

ALEX |SIMPLE IS GOOD

# 本节内容

浅谈ITIL

架构讲解

CMDB开发

接口认证

# 浅谈ITIL

- 1、事件管理（Incident Management）
- 2、问题管理（Problem Management）
- 3、配置管理（Configuration Management）
- 4、变更管理（Change Management）
- 5、发布管理（Release Management）

事件管理的目标是在不影响业务的情况下，尽可能快速的恢复服务，从而保证最佳的效率和服务的可持续性。事件管理流程的建立包括事件分类，确定事件的优先级和建立事件的升级机制。

问题管理是调查基础设施和所有可用信息，包括事件数据库，来确定引起事件发生的真正潜在原因，一起提供的服务中可能存在的故障。

配置管理的目标是：定义和控制服务与基础设施的部件，并保持准确的配置信息。

变更管理的目标是：以受控的方式，确保所有变更得到评估、批准、实施和评审。

发布管理的目标是：在实际运行环境的发布中，交付、分发并跟踪一个或多个变更。

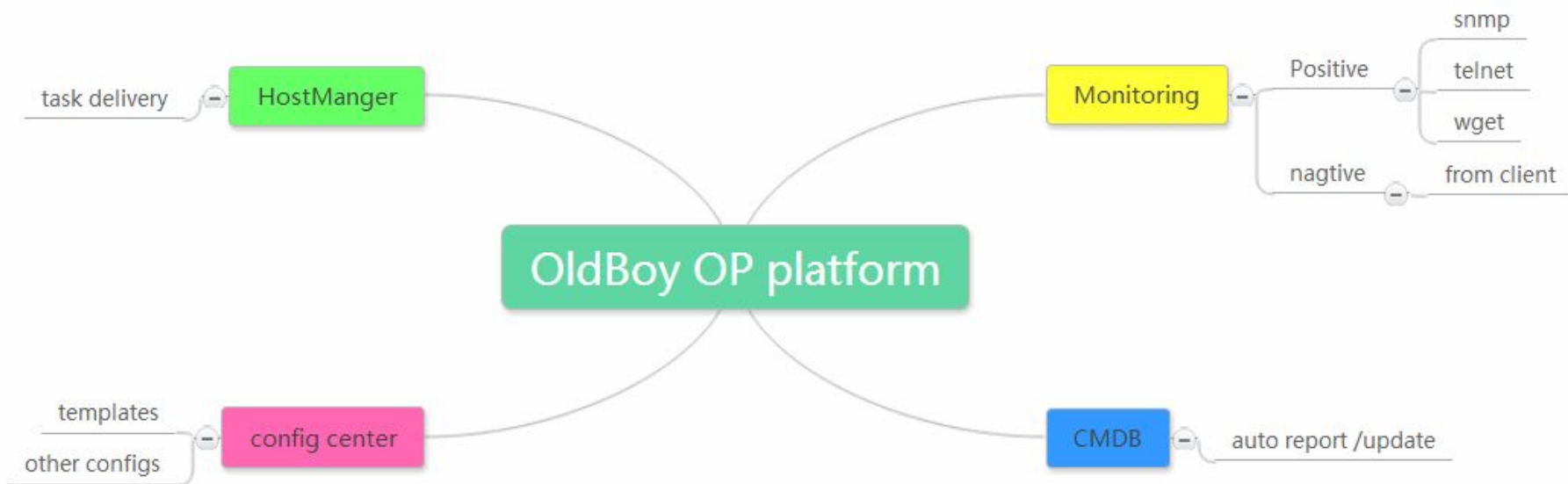
# 浅谈ITIL

ITIL即IT基础架构库(Information Technology Infrastructure Library, ITIL , 信息技术基础架构库)由英国政府部门CCTA(Central Computing and Telecommunications Agency)在20世纪80年代末制订，现由英国商务部OGC(Office of Government Commerce)负责管理，主要适用于IT服务管理 ( ITSM ) 。ITIL为企业的IT服务管理实践提供了一个客观、严谨、可量化的标准和规范。

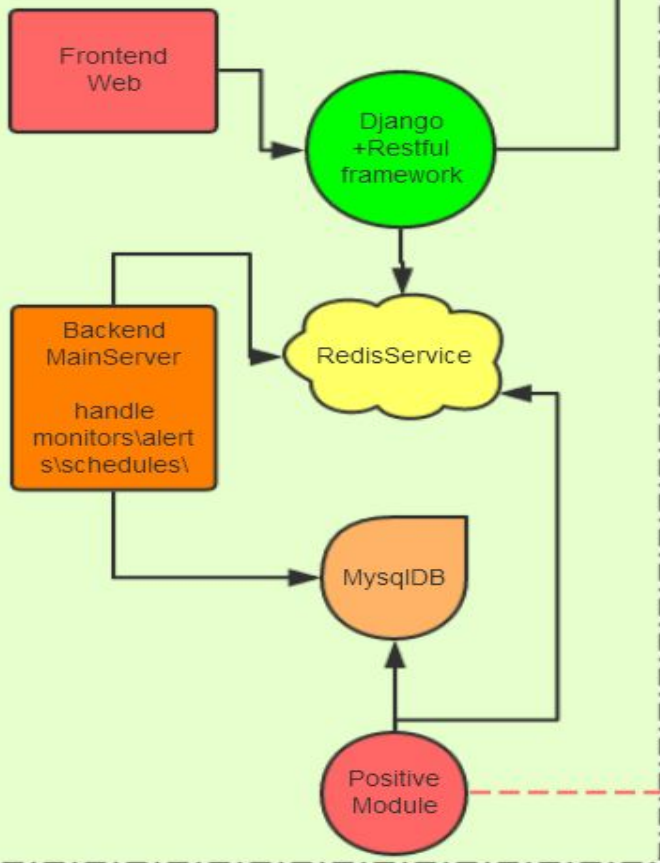
# 毕业项目功能介绍

开发CMDB资产数据库，实现资产数据的自动收集、更新  
实现可分布式的C/S架构监控，可自定义监控模板、监控插件  
实现基本主机管理  
接口如何安全认证？  
如何单机支持5000+并发？  
可实现任务批量下发、批量命令、文件分发  
WEB界面展示、监控画图等  
监控画图的历史数据如何最小化存储又不失真？

# OP运维平台功能概况



## Server Side



Nginx+Uwsgi  
=HttpApi

Get Configuration  
Get Tasks  
Download files  
Report Result

Client side  
SocketServer  
urllib2  
run tasks  
monitor services

## 老男孩教育PY开发第 6 期 毕业项目

项目要求：

1. 可实现分布式、可插拔
2. 采用生产者消费者模型
3. 接口符合restful api标准
4. 单机可处理5000+监控客户端
5. web展示要简洁、易用、美观

Postive monitor  
right away tasks

自动化运维路线图

AutoBench系统

- 硬件测试标准、工具、方法
- 软件测试
- 业务性能测试

运维协同开发环境

- gitlab代码仓库的建立
- 技术WIKI建立与分享

监控平台 (AutoWatch)

- 系统级监控
  - 系统资源监控
- 业务级监控
  - APM性能监控
  - 可用性监控

代码发布系统

- 公司svn/git仓库标准建立
- 产品业务代码的后台界面化上线发布
- 审计需求

日志采集系统

- 服务器错误及异常日志
- CDN运行日志
- 网络设备运行日志

数据分析系统

- 系统运行分析
- CDN运行状况分析

运维移动化

- 业务运营状况查看
- 故障、报警终端处理

自动化安装(AutoInstall)

- 目标: 界面化的批量部署安装
- 安装操作系统的模板定制

配置管理系统

- 目标: 服务器配置管理后台化操作
- Puppet & SaltStack 二次开发

CMDB运维资产管理系统

- 硬件基础信息
- 业务使用方信息
- 资产变更记录
- API接口

基础系统平台

- 自建CDN系统
  - 视频
  - 网页、图片
  - 目标
    - 商业CDN与自建CDN无缝调度
    - 速度、服务质量提升
- 存储系统
  - DB & NoSQL
  - 文件、图片
- 负载均衡系统
  - LVS/Haproxy/Nginx
  - 商业NetScaler负载均衡的管理
- 虚拟化平台
  - OpenStack & Docker 虚拟化平台开发



# CMDB

## 功能实现：

- 存储所有IT资产信息
- 数据可手动添加
- 硬件信息可自动收集
- 硬件信息可自动变更
- 可对其它系统灵活开放API
- API接口安全认证

# 立业之本：定义表结构

各种硬件都能存

资产变更有纪录

资产ID永不变

资产要有状态机

# 重中之重：接口设计好

可对内外灵活开放接口

接口定义要标准化

一定要提供排错依据

数据返回要标准

要能增删改查

所有异常要抓住

接口安全要注意

# 需求文档&架构图

# RESTFUL

## Representational State Transfer

资源

表现

状态

## Stateless

- Server contains no client state
- Each request contains enough context to process the message
  - Self-descriptive messages
- Any session state is held on the client

# 资源操作 示例

GET /tickets/12/messages- Retrieves list of messages for ticket #12

GET /tickets/12/messages/5- Retrieves message #5 for ticket #12

POST /tickets/12/messages- Creates a new message in ticket #12

PUT /tickets/12/messages/5- Updates message #5 for ticket #12

PATCH /tickets/12/messages/5- Partially updates message #5 for ticket #12

DELETE /tickets/12/messages/5- Deletes message #5 for ticket #12

<http://www.weiguda.com/blog/18/> rest framework快速入门

[http://www.ruanyifeng.com/blog/2014/05/restful\\_api.html](http://www.ruanyifeng.com/blog/2014/05/restful_api.html)  
restful 快速实践