Safe Family - 英特尔（中国）有限公司

时间：2016-03 至 2017-03
描述：跨平台家庭设备保护系统，支持 Android、iOS 和 Windows 平台。

- 功能包括：应用程序可用性控制；网页网址访问控制（仅限 Android 和 Windows 平台）；设备可使用时段控制（仅限 Android 和 iOS 平台）；电子地图围栏设定（仅限 Android 和 iOS 平台）；新装应用程序、访问网页网址、设备使用时间的请求即时通知和即时回应控制；设备所在地址变动的及时上报和设备进出电子地图围栏的及时通知；活动历史记录和查阅；设备位置及时跟踪。
- 账户系统为每个家庭一个账户，账户内分家长和孩子两种角色的家庭成员，每个家庭的家庭成员有数量限制，每位家长和各个孩子都可各自拥有多个设备，每个家长的设备可同时用于控制自家的孩子的所有设备，每个孩子的同种平台的设备共享相同的应用程序可用性控制、网页网址访问控制、电子地图围栏进出通知，每个孩子所有设备共享相同的设备可使用时段控制。
- 其中 Android 应用程序支持根据应用分类配置可用性控制，iOS 应用程序支持根据年龄阶段配置可用性控制，所有平台应用程序支持单独配置可用性控制，以及即时授权短时段内或长时间应用程序可用性。设备可使用时段分工作日及非工作日配置，可配置每天哪些时段可以使用设备。电子地图围栏设定在打开的地图中可选取围栏中心及方圆半径配置参数，可配置不限数量个电子地图围栏。
- 更多项目详情请查阅项目官方网站及各个平台的客户端说明（见后面链接）。

工作内容：后台（命名为 CloudServices）主导开发。

- 整个后台分为多个不同功能的服务运行在不同的机器（或 AWS EC2 实例）上，不同的服务位于反向代理/负载均衡的后面（本地开发及测试环境使用 haproxy 作为反向代理及服务请求转发路由作为替代），每种服务（除了任务调度 Scheduler 服务之外）运行多个 Python/Tornado 服务器示例并在其前面放置 nginx 作为负载均衡（因 Python 多线程/多进程问题）。各个后台服务的 Python/Tornado 进程实例通过 Supervisor 管理。
- 后台服务包括：用于认证和授权的 Auth 服务，用于同步 iOS 设备控制规则变动命令的 MDM 服务，用于和各个客户端界面操作相关的 Family 服务，用于同步各种控制规则及收集客户端数据的 Context 服务，用于定时发起 iOS 设备控制规则变动计算的 Scheduler 服务，用于计算 iOS 设备控制规则变动的 ActionTrigger 服务，以及用于生成具有国际化多语言支持的邮件内容的邮件模板服务。其中 Auth 服务和 MDM 服务一般成对部署在同一台机器（或 AWS EC2 实例）上，用于生成具有国际化多语言支持的邮件内容的邮件模板服务由官网同时提供（兼职）。
- 后台和各个平台客户端之间主要使用 HTTP 协议进行通讯，使用 JSON 格式作为数据交换格式，并使用 JSON Schema 校验传输数据的合法性。各个平台客户端的消息通知集成自其他开发团队的统一消息管理平台 CSP 负责，后台将即时消息发送到 CSP 平台后 CSP 平台放入消息队列并按照设备的不同平台经过 GCM/FCM（Android 平台）、APNS(iOS 平台）及客户端定时向服务器端轮询（Windows 平台）投递消息到目标设备。各种控制规则的同步由消息通知触发并通过 HTTP 协议同步（iOS 不需要，iOS 通过 Apple 的 MDM protocol 同步）。

- 后台使用 OAuth 2 作为授权机制，并通过集成自其他开发团队的统一账户管理平台的账户系统作为认证机制，本身并不存储账户认证凭证，通过电子邮件地址关联统一账户管理平台的账户系统。家庭成员通过各平台客户端登录使用系统，家长使用统一账户管理平台的账户系统认证，而孩子则使用由家长分配的不重复的多个字符串组成的密码认证。

- iOS 平台应用控制及设备可使用时段控制通过 Apple 的 MDM protocol 实现。而 Android 平台应用控制及设备可使用时段控制在 Android 客户端实现。后台通过任务调度服务（Scheduler）定时计算家庭成员中孩子的 iOS 平台设备控制规则的变动并将需要应用的命令投递到后台的 MDM 服务命令队列中，后台的 MDM 服务定时检查命令队列并将控制规则变动的命令按照 Apple 的 MDM protocol 最终发送到目标设备。

- 后台集成了其他的第三方服务，包括：用于验证家庭账户的统一账户管理平台（见上面描述）；用于获取 iOS 平台的应用程序元数据的 iTunes 服务；用于获取 Android 平台的应用程序（及部分无法通过 iTunes 服务获取得到的 iOS 应用程序）元数据的服务提供者；用于消息通知的 CSP 平台（见上面描述）；Google Geocoding API 服务（后来停用，改为在客户端获取并直接上报给后台服务器端）；用于集成 Apple 的 MDM protocol 的 Apple APNS 服务；用于向新注册账户及新添加孩子发送邮件的 SMTP 服务。

- 后台数据存储统一使用 Cassandra 集群，所有服务连接到同一个 Cassandra 集群，每个服务都有其独立的一个或多个 Cassandra Keyspace 用于存储数据。使用 Datastax 官方提供的 Cassandra Python 客户端，并添加对 Tornado 的异步的支持，使用 CQL 操作数据库。各个平台的应用程序元数据的存储带有 TTL 属性，以此作为缓存机制实现，使得应用程序元数据能得到及时的更新。

- 后台代码的组织使用标准的 Python 包的方式，由多个 Python 包组成，每个服务都有对应的 Python 包，通用的代码由其中一个 Python 包提供，封装了对数据库的基本操作、服务之间接口的统一调用基础、为每个服务提供的通用监控（Health）接口、以及每个服务的统一命令行基础。Python 代码中多次使用 Python 包的 entry points 以及 OpenStack 的 stevedore 实现插件机制。

- 后台代码使用大量的单元测试和静态代码分析工具等保证代码质量。每次新的代码改动必须保证单元测试覆盖率达到 80% 以上，否则不予通过，而静态代码分析工具使用 flake8 及部分调整的 pylint ，另外还使用代码复杂度 Radon 等用于保证代码的复杂性和可维护性。这些都在集成测试中强制执行，配合代码审核系统 Gerrit 防止未经确认的改动直接提交到代码库中。

- 后台还提供一个周期性数据收集并发送收集结果的脚本，每周定时连接数据库，收集当时所有账户信息（包括家庭成员角色等）、所有设备信息（包括设备平台、设备注册时间及最后在线时间等），最后将收集结果通过邮件附件方式放送给相应的人员。

- 后台每台机器（或 AWS EC2 实例）上部署了 Logstash，DevOps 使用 ELK 技术栈收集管理日志，然后通过 Kibana 来查阅日志，同时也配置了 AWS 的 CloudWatch 来查阅日志。

- 后台使用外部的监控服务从全球各个地区定时监控各个服务的健康（通过各个服务暴露出来的 Health 接口）状态。

- 集成测试环境在计算中心的 TeamCity 主机及 Agent 机器中进行。QA 工程师的测试环境部署在 AWS 的多个 EC2 示例上。预发布（Staging）环境和生产环境一开始也是部署在 AWS 的 EC2 示例中，后期迁移到英特尔自己的计算中心。

- 后期为 Verizon 加入了第三方合作的集成。

技术栈：

- 服务器后端：Python、Tornado、Cassandra、nginx、Supervisor、Ubuntu Server 等。

链接：

- Safe Family: http://family.mcafee.com/