六月实习简报

本月主要实习工作：

园丁平台Token托管服务的接口优化，以及和多个第三方应用联调对接。想要接入智慧校园的第三方应用可以通过园丁平台接入，提供给使用智慧校园的教师，学生，家长使用。因为智慧校园依托企业微信，而由于企业微信的受众更多，所以很多应用开发服务商已经和企业微信对接完成。园丁平台充当的角色就是一个“适配器”，第三方应用通过园丁平台认证，授权，我们进行Token托管，如果涉及到第三方应用和企业微信的交互，则通过园丁内部调用企业微信的接口。我所参与的工作主要是，提供多种第三方应用凭证(token)。不同类型的token涉及到不同的权限。大致分为以下两种token，suite\_access\_token。当第三方应用需要注册到智慧校园的应用市场的时候，需要通过园丁平台颁发的suite\_id,suite\_secret和suite\_ticket获取一个suite\_access\_token。其中，suite\_ticket是一个服务端定时推送的随机字符串(每隔10min推送一次)。第三方服务商获取的suite\_access\_token的时效为2h。通过在url携带suite\_access\_token字段，在网关鉴权的时候，会完成两件事，一是校验长度是否合法，二是通过读取缓存判断是否是对应的应用。Suite\_access\_token主要是一个与应用相关的令牌。

还有一种凭证（Corp\_access\_token），是在智慧校园的一个学校安装了这个应用之后会返回给第三方应用开发商一个corp\_access\_token。开发商（第三方应用）可以通过这个token获取智慧校园学校的详细信息（例如，智慧校园学校的组织架构信息）。

实习工作中遇到的问题：

之前主要是完成了接口的开发，但是其实对系统也没有深入了解。开发的时间也很紧。所以在联调的过程中出现了很多的问题，除了需求不明确以外，在实际生产环境中没有考虑到并发的问题。

整个token部分的核心是，两种token的生成。一开始我采用的是随机种子，然后每次生成一个0-63的整型去查一个数组表。随机种子采用的是当前时间戳。但是在联调的时候出现了，两个不同的学校安装一个应用返回的corp\_access\_token完全一致。原因是，两次请求的时间戳（精确到毫秒）是一样的。为了解决这个问题，我考虑了两种方案，第一个是在生成token的时候加锁，保证只有一个请求，不会出现相同的时间戳作为随机种子。第二种是采用twitter开源的snowflake分布式id生成算法。考虑到加锁会影响性能，所以一开始采用了snowflake（雪花算法）生成token。不过最终在跟企业导师的谈论下，采用的是特殊信息和当前时间戳，做hash的方式生成token。

七月正在进行的工作：

企业微信工作台配置，智慧校园的每一个学校在创建的时候可以选定一个应用模板（不同应用的组合）。我们需要提供一个入口，给运营人员使用，帮助他们管理企业微信应用的模板（创建编辑模板），这些模板信息还需要同步到企业微信，同时更新企业微信工作台上的应用模板。对于已经安装了原始模板的学校，还需要为他们切换应用模板。

接触到的技术知识：

Kubernetes（容器编排系统），etcd（服务注册与服务发现，类似Java体系钟的zookeeper但是一致性协议采用的是raft），gRpc（rpc调用框架，使用protobuf作为接口定义语言，底层协议是http2.0）。