**APP业务描述**

一 基本需求和商务模式

1：基本需求

* APP支持WINDOWS/MAC/IOS/ANDRIOD 4 个平台；
* APP既能作为加速之用（SAAS加速类），也能作为虚拟组网用；加速和虚拟组网的区别在于虚拟组网需要和网络内机器互联互通，不仅自己机器能访问其他机器，其他机器也需要能访问本机，所以在IP的规划上有要求。
* APP不仅仅是解决移动办公问题，对于小的分支，也可以用APP替代盒子。
* 需要考虑和钉钉做集成，比如作为钉钉的一个插件提供服务。
* 未来也许会做针对2C的APP，技术上应该保持一致。

2：商务模式

技术上支持2种计费模式：按流量和按账号，按账号需要考虑唯一账号登录问题，而且单一账号需要考虑月流量上限控制。

商务上更复杂的考虑：

* 是否需要考虑国内/国外加速的不同计费模式或不同费率？
* 是否需要考虑一个企业的有几个账号是高级帐号，几个账号是普通账号，高级帐号类似走专线。

二 业务描述

1：BOSS商务/运维视角

* 可以为客户增加/删除APP支持
* 如果增加APP支持，需要选择计费模式
* 如果是按账号计费模式，需要输入账号数量和每月最高流量限制；
* 如果之前已用账号计费模式而修改账号数量，需要检查是增加还是减少账号，如果是减少账号，需要把客户的账号全部设置为不激活状态（比如原来是100个账号，现在改成10个账号，如果不设置为不激活，则留后门。有更好做法？），同样，如果之前是按流量计费，修改成按账号计费，也需要自动把客户所有的账号设置为不启用状态。
* 如果客户有多个组网/加速方式，需要选择APP要支持的组网/加速方式（这个是客户决定还是我们决定？）
* 我们之前的隧道，是预先设置好的，APP的隧道需要预先设置好吗？建议采用动态隧道方式。APP隧道的最大带宽不做限制，可以设置默认值为50M之类，足够一个用户使用。(但这个会对TWS的分配产生影响，TWS需要预留带宽，对于APP类的，如何预留？）
* 对于组网应用，需要客户划分出一段IP给APP，在BOSS里输入，用户登录时分配给APP对应的IP。

2：BOSS 客户视角

* 客户管理员可以设置账号/密码，并决定账号是否启用，如果是按账号计费，则启用的账号必须在最大账号数之内。
* 管理员可以查询账号的使用情况（登录时间、流量，但是否需要显示访问的内容，比如上网URL? 有点泄漏隐私)
* 账号不能删除，因为流量和账号是挂钩的，一旦删除，流量就成了无源了。
* 如何有效控制账号数，比如一个客户有10个账号数，他先给10个人开了账号，等这10人把一个月的流量全部花光了，再把这10个账号关闭，启用另外10个账号继续用。（一种方式是账号由我们产生，密码由客户设置）

3：账号

账号必须全平台唯一，通过登录的账号，我们能找到账号对应的客户。所以账号可以是EMAIL或手机号。

如果账号是客户维护，需要考虑和第三方认证打通，比如钉钉，或者其他认证系统如RADIUS等等

4：登录

* APP启动，用户需要先登录。
* 登录的URL类似：/appcmd/userauth?username=xxxx&passwd=xxxxx&forcelogin=0，其中passwd的值为密码的hex-md5值，forecelogin标识是否强制登录。
* 服务端收到后，检查用户名密码，如果失败，返回错误给用户；
* 如果成功，检查当月流量是否超标，如果超标，返回错误给用户；
* 如果没超标，检查这个用户ID是否已经登录，如果已经登录，而且不是强制登录，则返回错误给用户，同时提醒用户是否强制登录。
* 如果已经登录，则用户需要强制登录，则产生一个SESSIONID,在内存（REDIS里）维护USERID/SESSIONID信息。返回成功给用户，同时返回用户组网信息。
* 登录成功后，获得组网/加速配置，APP需要设置路由，把流量引导过来；
* APP和后端的交互，建议采用HTTPS（而且IOS里是强制要求HTTPS，HTTP需要特殊批准过）

5：KEEPALIVE和流量报告

* 登录成功后，APP需要定时KEEPALIVE，建议30秒一次。
* KEEPALIVE格式为：/appcmd/userkeepalive?userid=xxx&sessionid=xxxx&traffic=xxxx，其中traffic是APP累计的。
* 服务端收到后，检查userid/sessionid，如果存在，记录traffic并成功返回，如果不存在，返回错误，APP提示用户被动退出。
* 如果服务端超过一定时间没有收到KEEPALIVE(比如2分钟），则认为用户异常退出，把用户状态设置为异常退出状态。
* 如果后台配置有变化，也可以通过KEEPALIVE的RESPONSE获得更新的配置。

6：SESSION报告

* 每个SESSION结束时汇报。汇报内容同现在STEP里的SESSION汇报。

7：选路

* 同现在的STEP

8: 退出

* APP退出时，必须向后台发送退出请求，成功返回或超时后才彻底退出（超时可设置为10秒），APP同时设置清除路由设置。
* 退出请求：/appcmd/logoff?userid=xxx&sessionid=xxx&traffic=xxxx
* 服务端收到后，记录最终的流量，并在内存里删除用户和SESSION的对应信息。

9： APP的故障报告

TBD

三：APP基础架构

* 4个平台，都用TUN作为虚拟网卡设备，获得要加速的流量；
* 登录后，根据获得的组网/加速信息，设置对应的路由，把流量导到TUN设备上；
* UI作为配置和控制之用，核心的组网和加速模块基于开源tun2socks改造，把tun的流量转变成TCP/UDP的SESSION进行加速/组网处理。
* WINDOWS上，核心模块作为WINDOWS的SERVICE，由UI进程控制和管理；2个进程间通过SOCKET进行通讯。
* 安卓上，核心模块作为独立进程，由UI进程控制和管理，2个进程间通过SOCKET进行通讯。
* MAC/IOS上，核心模块作为库合并进UI进程，通过API进行信息交互。

四：自动升级

* 由于IOS一旦上架，是有APP STORE自动推送给用户升级的，所以APP必须做到向上兼容，就是说链接的STEP可能是老版本的，APP必须支持多个版本的协议，和STEP建立隧道时，首先需要协商版本号。
* 安卓/WINDOWS/MAC需要在APP上支持自动升级功能。

五：APP监控和问题排查

TBD

六：其他

* 移动APP应该能获得用户所在的地址（经纬度）