###### 需求提要

目前的统计SDK大多是针对游戏后台开发了各个语言版本的接口（c++/php/java/Node.js），通过tcp协议传输将游戏数据发送到统计后台。但针对游戏前端（即客户端）的统计接口只开发了new-misc系统，且只支持自定义数据。在这种情况下，要统计基础统计项数据（登录、付费等）就只能将这些数据先传至后台，由其调用统计接口，如果后台的技术栈并不在支持的语言列表内（例如Lua），数据的接入会是一项现成环境难以实现的任务，针对目前偏重手游的开发情况，需要考虑Android/iOS的技术支持，我们考虑拓展既有的new-misc系统，使之支持基础统计项，为游戏方的接入提供多一种选择，支持直接的http协议调用，进而支持Android/iOS SDK封装包的数据接入方式。

###### 设计宗旨

一、new-misc系统本身支持自定义数据接入，在此基础上实现基础统计项的接入，需要对两类接口做好区分

二、考虑到之后基础统计项可能还会有扩展，建议在设计之初就要考虑做好对扩展具备较好的兼容性

三、目前的统计平台正在开发基于Hadoop生态技术栈的新系统TMS，因而在数据格式、数据传输等问题上需要考虑能同时兼容既有的两套系统，并要考虑后期能较方便地进行切换和分离。

四、安全性、鲁棒性问题：由于数据是走http协议，公网传输，如果报文显式传输，数据的安全性确实难以保证，容易发生盗取、篡改、捏造等安全性问题。另外http报文传输对于接口格式也没有硬性格式限定，需要在new-misc服务器端做好数据检查，对于发生数据丢包等情况需要做好应对。

###### 接口概览

构造器参数

与后台SDK先构造再调用的方式不同，new-misc需要同时接收来自多个游戏的HTTP格式报文，不存在构造器的概念，因而在每个接口的调用时都需要加入gzsp的信息

具体参数包括：

int game\_id, //游戏ID,由数据分析平台统一分配

int16\_t zone\_id = -1, //区ID，缺省值-1表示全区或不分区

int16\_t svr\_id = -1, //服ID，缺省值-1标识全服或不分服

int16\_t site\_id = -1, //平台ID，缺省值为-1时代表全平台，如果这款游戏将来或者现

隐性参数

在调用后台统计SDK的情况下，某些参数不需要显式指定，但这种情况对于http调用可能未必适用，例如如下参数需要注意：

string hip //落数据的IP地址

long ts //数据事件发生的时间戳

数据支持接口及参数

**（1）****登录游戏online服务器(login\_online)**

std::string acct\_id, //用户账户(米米号)

std::string player\_id,//用户角色标识(单角色游戏可赋空值，多角色游戏可填写角色id)

std::string race, //用户选择的职业(对无职业游戏，该字段可赋空值)

bool isvip, //是否VIP用户（true:是 false：否）

int lv,//用户当前等级

uint32\_t cli\_ip,//用户的IP地址, 无法获取时取值0

std::string ads\_id, //用户是从哪个广告渠道跳转过来的

std::string zone = "",//用户登陆区服（不区分区服的，填空值）

std::string browser = "", //页游填用户使用的浏览器类型（如firefox, IE）

std::string device = "",//页游填写空值

std::string os = "", //用户浏览器的flash版本。

std::string resolution = "", //用户屏幕的分辨率

std::string network = "", //用户使用的网络

std::string isp = ""//用户网络的服务提供商

**（2）****创建游戏中角色（reg\_role）**

std::string acct\_id, //用户账户(米米号)

std::string player\_id, //用户角色标识(单角色游戏可赋空值，多角色游戏可填角色id)

std::string race, //用户选择的职业(对无职业游戏，该字段可赋空值)

uint32\_t cli\_ip,//用户的IP地址, 无法获取时取值0

std::string ads\_id, //用户是从哪个广告渠道跳转过来的

std::string browser = "",//用户使用的浏览器

std::string device = "",//页游，填写空值

std::string os = "",//用户浏览器的flash版本。

std::string resolution = "",//用户屏幕的分辨率

std::string network = "", //用户使用的网络

std::string isp = ""//用户网络的服务提供商

**（3）****退出游戏online服务器(logout)**

std::string acct\_id, //用户账户(米米号)

bool isvip, //是否VIP用户(true:是 false:否)

int lv, //用户退出时的等级

int oltime //本次用户总共的在线时长

**（4）****统计当前在线人数(online\_count)**

int cnt, //当前在线人数

std::string zone="" //1、默认值为空是统计总在线人数；2填写“电信”或“网通”，则分别统计电信或网通的在线人数。

**（5）****用户升级(level\_up)**

std::string acct\_id, //用户帐号(米米号)

std::string race, //用户职业(如果不需要区分职业，赋值为空)

int lv //升之后的等级

**（6）****付费（pay）**

std::string acct\_id,//用户账户(米米号)

bool isvip, //是否VIP用户(true:是，false:否)

float pay\_amount, //付费额度，见详细说明

CurrencyType currency, //货币类型，见详细说明

PayReason pay\_reason,//支付类型，见详细说明

std::string outcome,//支付产生的物品，见详细说明

int outcnt,//获得数量，见详细说明

std::string pay\_channel = "米币帐户"//支付渠道号，见详细说明

**（7）****免费获得游戏币(obtain\_golds)**

std::string acct\_id,//用户账户(米米号)

int amt//获得金币的数量

**（8）****使用游戏币购买道具(buy\_item)**

std::string acct\_id,//用户账户(米米号)

bool isvip,//是否VIP用户(true:是 false：否)

int lv,//用户购买道具时的等级

float pay\_amount,//支付的金币数量

std::string outcome,//购买的道具

int outcnt//购买的道具数量

**（9）****消耗游戏币（use\_golds）**

std::string acct\_id, //用户帐号(米米号)

bool isvip, //是否VIP用户(true:是 false:否)

std::string reason, //原因（开启新功能、跳过关卡等）

int amt, //支付的金币数量

int lv//用户等级

**（10）****接收任务（accept\_task）**

TaskType type, //任务类型，枚举类型，如新手任务：StatLogger::task\_newbie

std::string acct\_id, //用户(米米号)

std::string task\_name, //任务名称

int lv//接收任务时的等级

**（11）****完成任务（finish\_task）**

TaskType type, //任务类型，枚举类型，如新手任务：StatLogger::task\_newbie

std::string acct\_id, //用户(米米号)

std::string task\_name, //任务名称

int lv//完成任务时的等级

**（12）****放弃任务（abort\_task）**

TaskType type, //任务类型，枚举类型，如新手任务：StatLogger::task\_newbie

std::string acct\_id, //用户(米米号)

std::string task\_name, //任务名称

int lv//放弃任务时的等级

**（13）****退订VIP服务（unsubscribe）**

std::string acct\_id, //用户(米米号)

UnsubscribeChannel uc//渠道，目前只有短信和米币两个渠道(StatLogger::uc\_duanxin和StatLogger::uc\_mibi)

**（14）****销户VIP（cancel\_acct)**

std::string acct\_id, //用户(米米号)

string channel//销户渠道，如支付宝、财付通等。Channel是字符串，由调用方和平台共同商定