基于WebService构建系统管理平台

通过WebService方式提供系统管理功能。

注册：

注册包括**主机注册**，**服务注册**和**设备注册**。

主机注册：对应每台pc，或者独立拥有IP地址的单元，拥有唯一的名字；

服务注册：服务运行与主机，服务注册总是需要绑定到主机注册上。服务可以拥有自己的入口，如xmpp jid，rtsp url之类；

设备注册：设备绑定与主机。包含设备厂商，型号，版本号，串码之类；

逻辑单元注册：如班级，年级，学校之类；

典型的注册需要周期发送心跳，用以声明自己的“在线状态”；逻辑单元不需要，总是在线。

主机注册：

token = regHost(

hostname， // 主机唯一名字

showname, // 方便显示的名字

ip[], // ip 地址,可能多个

);

服务注册：

token = regService(

hostname, // 绑定的主机名字，一般通过环境变量，或注册表得到

name, // 在hostname下，唯一的名字

showname, // 方便显示的名字

type, // 服务类型，如 eval，record service, …

urls[], // 服务入口，如 xmpp:xxxx, rtsp://….，http://….

);

设备注册：

token = regDevice(

hostname,

name, // 在 hostname下，唯一的名字

showname， // 方便显示的名字

catalog, // 设备类型，如ptz，encoder，….

vender, // 生产厂商

model， // 型号

version， // 版本号

serial， // 序号

);

逻辑单元的注册：

token = regLogic(

name, // 全局唯一名字

showname， // 方便显示的名字

parent, // 父节点，必定是逻辑单元类型

children[], // 子单元名字，可能是主机，设备，服务，或逻辑单元

);

心跳：

heartBeat(

token // 注册返回

next, // 时间，下次心跳将在该时间之前发出，

// 如果ws在此时间没有收到心跳，则可以认为该

注册失效

);

注销：

unReg(

token // 注册返回

);

用户管理：

用户的创建，修改，删除，分配权限，角色等。

查询：

支持根据主机名字，服务类型，设备类型，是否在线等多种条件的查询。

查询得到的结果不能作为准确在线的依据，必须通过对被查询单元的进一步检查才行。