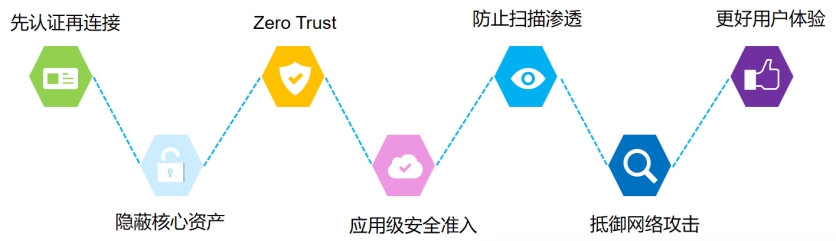
技术架构内容要求

## 业务逻辑架构，请结合实际平台或产品进行分析；

软件定义边界（Software Defined Perimeter，SDP）是国际云安全联盟（CSA）为阻止对应用基础设施的网络攻击，增强用户网络安全，于2013年提出的一个全面的安全解决办法，并在2017年CSA Summit（云安全联盟论坛）上提出了全新的SDP-New Software Defined Perimeter（软件定义边界）概念。其核心思想是通过SDP架构隐藏核心网络资产与设施，使之不直接暴露在互联网下，使得网络资产与设施免受外来安全威胁。

通过使用新的SDP安全模型，融合设备身份验证、基于身份的访问和动态配置连接等安全组件，阻止所有类型的网络攻击，包括DDos、中间人攻击、服务器查询（OWASP十大威胁）和高级持续性威胁（ATP）。

云计算、BYOD（移动办公）等新的IT技术影响，使得企业的网络边界随之变化。紧随而来的是因新技术导致的各类安全问题，传统网络架构是通过安全设备（如防火墙）设置好规则墙，通过之后才能访问内部应用。但是，防火墙规则局限性大，灵活性不足，无法满足随时随地异地办公的动态需求。而且传统VPN技术的应用中，只要被授权允许进内网的用户，VPN用户便对企业网络上的资源拥有过于宽泛的访问权限，令敏感资源及信息可能暴露在VPN用户和攻击者面前。问题二是传统的TCP/IP网络中，是先连接后验证，那么不可避免的将网络资产与设施暴露在公网下。SDP采用的是先认证、再准入原则，动态、按需、最小化实现网络应用安全连接与访问。



SDP（Software Defined Perimeter）软件定义边界的英文缩写，是一种全新的网络边界概念。由CSA国际云安全联盟定义，核心思想是通过SDP架构可以隐藏核心网络资产与设施使之不暴露在internet，极大程度免受各种攻击行为与安全威胁。SDP弥补了TCP/IP架构的缺陷，避免重要网络资产直接暴露在internet ,SDP可以将应用直接部署在互联网上就可以像内网应用一样高效且安全访问，实现关键资产隐藏、零信任，规避网络安全风险，在架构上实现彻底安全防护。

SDP主要包含三部分：SDP控制器（Controller）、SDP网关（Gateway）和SDP客户端（Client）。

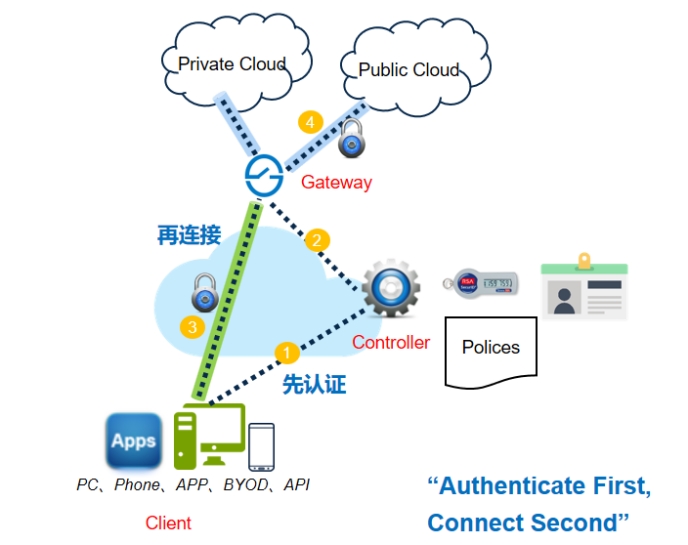


图 SDP 架构图

**1安全控制器（Controller）**

作为SDP核心组件之一，SDP安全集中控制器（controller）实现零信任安全准入控制，基于零信任安全模型，不再以网络为中心实现准入控制，而是以用户权限为中心实现应用级的安全准入，通过形成按需、动态的权限矩阵，结合IAM认证技术对用户身份进行管理，同时可对接第三方IAM平台实现用户身份的多因素认证。在用户认证基础上，对访问资源范围进行策略预设立，有效控制访问资源列表范围。

* 零信任安全模型
* 用户权限为中心安全准入
* 按需动态权限矩阵
* IAM及多因素认证管理
* 资源访问控制
* 安全策略管理

**2安全网关（Gateway）**

SDP架构所拥有的优势中，网络隐身的作用最为突出。作为实现网络隐身的模块之一，安全网关在控制器模块的预验证、预授权、应用级访问准入、扩展性四点的基础上，为后端应用资源增加了一层隐身防护壳，即所有受安全网关连接控制的后端资源，没有对外暴露的服务或端口，只有通过验证且被授权的用户，才可以接入隐身的网络资源，并且范围被严格控制且实现动态开放与回收。此项能力大大增强了新的网络架构的安全等级，从而提升了针对不良攻击的防御系数。

* 网络资产隐藏
* 单向报文认证（设备接入认证、身份接入认证）
* 动态资源分配
* 加密隧道数据传输
* 动态防火墙

**3SDP客户端（Client）**

SDP客户端主要为实现终端安全防护，支持采用安全终端或SDK嵌入方式，与SDP控制器形成可信的用户身份认证，并对接入终端信息进行检测与核查，同时可实现终端安全基线扫描，帮助用户全面了解终端运行环境的安全性以及实现安全准入控制，保证接入终端的可靠性，从而实现在终端访问应用资源的动作是一个安全的、可溯源的、可管可控的过程。

|  |  |
| --- | --- |
| * 用户信息 * 终端指纹 * 数据证书 * 短信密码等多方式 | * 动态接入 * 可信认证 * 权限控制 * 数据加密 |

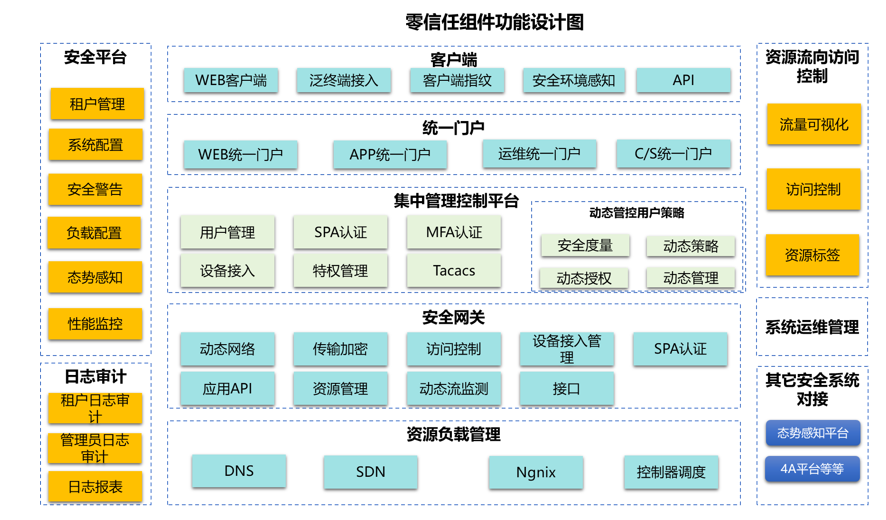
零信任软件平台总体设计以SDP标准技术为底座，以业务安全场景为向导，从“云、管、网、边、端”的安全防护进行设计，具备多云部署、弹性扩展、服务交付的数字化型能力的零信任功能组件，具体如下：

零信任软件平台设计以SDP标准为基础，从标准规范三角架构开展客户端、集中管理控制平台、安全网关进行功能设计，在设计组件的同时充分考虑零信任软件平台安全需求进行全方位设计。整体设计包括零信任软件平台设计和辅助功能组件设计。

### 1.功能架构

#### 1.1功能性设计

依据零信任标准组件和辅助功能组件设计要求，对本次零信任功能进行设计，具体如下图：



图零信任软件平台功能设计图

组件功能设计从软件开发实现的角度着手，综合考虑功能归类、接口耦合、用户应用体验、系统整体性以及可管性等等，设计零信任安全组件功能，包括客户端、统一门户、集中安全管理控制平台（包含动态管控用户策略）、安全网关、资源负载管理、资源流向访问控制、日志审计和系统运维工具。

客户端：实现用户登录管理、统一应用入口，管控接入端的安全，包括环境安全、身份安全、操作安全等。零信任客户端支持Windows/MAC、Android、Linux及SDK等形态，包括WEB客户端、泛终端接入、客户端指纹、安全环境感知、API等主要功能，同时还提供其它辅助功能，如全局路由模式、SDK内接口、多协议访问、资源访问控制，环境感知模块基于动态的策略，对终端用户进行权限管控，资源访问控制等。

统一门户：打造应用统一入口，支持WEB统一门户、APP统一门户、运维统一门户以及C/S应用统一门户。通过统一门户实现应用安全、集中、便捷访问。

集中安全管理控制平台：实现IAM可信认证以及基于应用级策略的动态权限管控下发。集中安全管理平台控制器通过客户端采集接口对用户身份信息进行采集，并与核心IAM统一认证模块进行单包认证；IAM对用户身份信息及权限与身份权限库或第三方认证平台进行匹配，通过后对网关及客户端进行策略联动下发，并严格控制策略，功能包括用户管理、MFA认证、SPA隐身、设备接入、特权管理、第三方接入等等。

平台架构：平台架构作为零信任SDP系统接入配置的可视化介质，采用VUE框架开发，通过中文可视化界面，树状目录，与管理员形成良好交互。实现集中界面管理，增加运维效率及成本。通过建立零信任安全管控平台，实现对各个组件可视化统一管理能力，增加运维效率；调度各功能模块，实现安全接入、身份管理、终端设备管理、暴露面防护、资源流向管控、网络设备管理等，形成零信任东西向及南北向统一管控；当在小规模部署场景时，平台架构可与控制器集成为SDP统一管理平台。

安全网关：实现接入认证及连接管理，实现业务系统网络隐身，并基于动态安全策略实时管控合法用户的访问。安全网关通过接收控制器策略指令，对用户客户端身份进行信息采集，并对用户身份及权限进行鉴权匹配，进行动态开放加密隧道访问以及动态策略回收管理，并对用户流量进行安全检测；同时在应用侧对资产的隐身防护，减少互联网暴露面。

日志审计：零信任SDP系统日志审计功能，可以收集各个SDP网关以及所有SDP客户端发送过来的日志信息，并对这些日志信息进行大数据智能统计分析，以满足企业运维及安全审计需求。通过日志审计系统，企业管理员随时了解整个IT系统的运行情况，及时发现系统异常事件；另一方面，通过事后分析和丰富的报表系统，管理员可以方便高效地对信息系统进行有针对性的[安全审计](https://baike.baidu.com/item/%E5%AE%89%E5%85%A8%E5%AE%A1%E8%AE%A1/5390213" \t "_blank)。

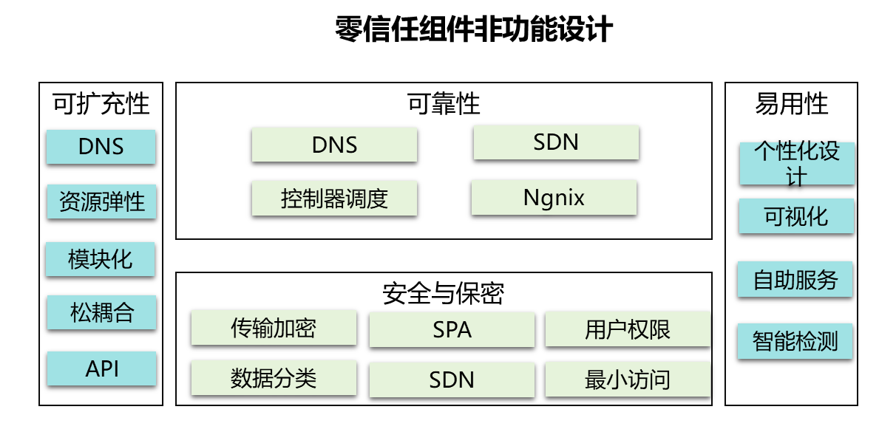
资源流向访问控制：资源流向访问控制系统（微隔离），通过在宿主机上部署对应的微隔离agent，通过agent采集相关主机信息，并通过集中管控平台对agent批量化控制实现策略下发、策略采集、策略同步、拓扑呈现等，通过调用系统iptables实现对主机服务器大批量调度安全策略管理。

动态管控用户策略：基于IAM可信认证对用户进行动态权限管控，结合用户身份、环境感知、访问操作内容等，多维一体化感知动态控制接入权限，接入用户授权等。

系统运维工具：系统运维工具主要为管理员提供相关便捷操作手段，包括IP反查、ICP备案、Dig工具、ping工具、恶意域名及IPv6查询等，实现网络质量、业务服务等的综合检测，提升系统运维故障的排查效率。

#### 1.2非功能性设计

非功能性设计主要从可靠性、可扩充性、安全与保密、易用性、安全防护维度考虑，具体设计如下：



图零信任软件平台非功能设计

可靠性：零信任软件平台研发中可靠性能力决定该系统是否能被广泛应用，为更好的实现该软件的应用目标，在可靠性方面进行全方位设计，包括采用DNS域名对控制器、网关等资源进行域名负载，确保任何一台资源故障不影响零信任应用；采用SDN技术进行网关和控制器设计，实现各组件运行过程中SDN技术对各组件进行自动巡检、自动故障修复，保障各组件正常运行；采用控制器调度技术，对高用户量的控制器、网关进行访问调度，降低各组件运行过程中的压力；采用Ngnix应用负载均衡，保护零信任支持的业务系统应用负载均衡，保护业务高可用性，整体形成7×24小时不间断运行的运营，具有高可靠性设计。

可扩充性：弹性扩展，采用DNS集群使所有节点为全活模式，并且针对任何故障节点，可自动下线；在资源压力时通过弹性扩展；基于模块化设计可以根据需求进行横向扩展和功能增加。此外，系统各个组件基于云部署模式，利用SDN技术实现控制与转发分离，可以动态扩展扩充。

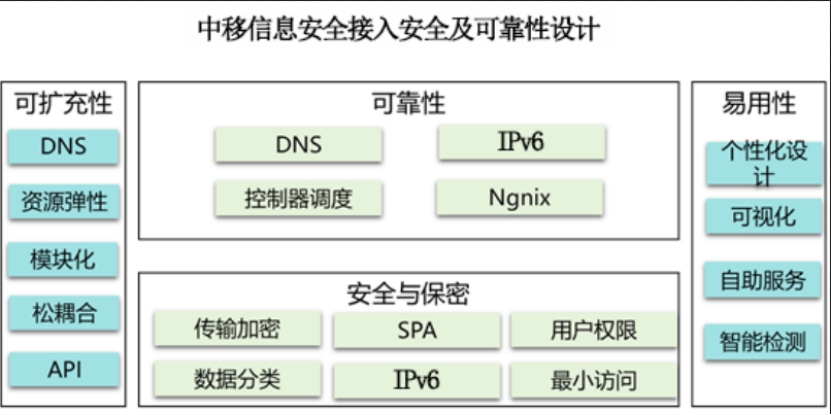
安全与保密：系统结合零信任SPA单包认证能力，结合内部东西向流的访问控制，实现网外－网内一体化安全控制。同时系统对用户及管理员设置多级别、层次的权限管理控制，用户只能查看并访问权限内的应用资源。

易用性：系统平台针对用户及管理员使用适应及易用性进行针对性优化，客户端统一protal访问入口，管理平台采用集中化管理，增强用户使用感、增加运维效率集，另外，针对用户提供自动服务，实现自动管理、自动检测、智能排障、自我管理和自动修复等，方便各类用户使用零信任系统。

安全防护：基于隐身网络，对应用资产进行隐身防护，减少互联网暴露面。

### 2.非功能性架构

SDP平台安全及可靠性设计主要从可靠性、可扩充性、安全与保密、易用性、安全防护维度考虑，具体设计如下：



可靠性：零信任SDP中可靠性能力决定该系统是否能被广泛应用，为更好的实现该软件的应用目标，在可靠性方面进行全方位设计，包括采用DNS域名对控制器、网关等资源进行域名负载，确保任何一台资源故障不影响零信任应用；采用SDN技术进行网关和控制器设计，实现各组件运行过程中SDN技术对各组件进行自动巡检、自动故障修复，保障各组件正常运行；采用控制器调度技术，对高用户量的控制器、网关进行访问调度，降低各组件运行过程中的压力；采用Ngnix应用负载均衡，保护零信任支持的业务系统应用负载均衡，保护业务高可用性，整体形成7×24小时不间断运行的运营，具有高可靠性设计。

可扩充性：弹性扩展，采用DNS集群使所有节点为全活模式，并且针对任何故障节点，可自动下线；在资源压力时通过弹性扩展；基于模块化设计可以根据需求进行横向扩展和功能增加。此外，系统各个组件基于云部署模式，利用SDN技术实现控制与转发分离，可以动态扩展扩充。

安全与保密：系统结合零信任SPA单包认证能力，结合内部东西向流的访问控制，实现网外－网内一体化安全控制。同时系统对用户及管理员设置多级别、层次的权限管理控制，用户只能查看并访问权限内的应用资源。

易用性：系统平台针对用户及管理员使用适应及易用性进行针对性优化，客户端统一protal访问入口，管理平台采用集中化管理，增强用户使用感、增加运维效率集，另外，针对用户提供自动服务，实现自动管理、自动检测、智能排障、自我管理和自动修复等，方便各类用户使用零信任系统。

安全防护：基于隐身网络，对应用资产进行隐身防护，减少互联网暴露面。

围绕上述安全性需求分析，从如下方面阐述产品安全性设计及其实现方式：

（1）密码算法

与控制器交互使用的是对称加密算法sm4

控制器与客户端会生成一批加密因子，

控制器和客户端会在同一时间片内选择同一个加密因子结合时间，客户端特征计算出一个sm4密钥

双方数据交互使用sm4加密传输数据

客户端与网关交互

\* 握手使用的是sm2非对称加密，并在握手中计算出后续隧道交互数据加密的sm4密钥

\* 握手成功后建立一个通讯隧道，后续隧道的数据交互都使用sm4加密

\* 隧道会定期协商更新sm4密钥

密码算法使用软件方式。

（2）密钥管理

客户端与控制器：

客户端与控制器的密钥是动态生成的，并且有时间片的限制，过期自动废弃

想加强客户端与控制器之间的通讯安全，可定期更新加密因子库，调整加密因子时间片的大小

客户端与网关：

sm2证书放在网关上，部署的时候要控制证书的访问，读写权限，可以定期更行证书

在客户端与网关建立隧道数据传输的sm4密钥都是定期更新的，无其他安全风险

（3）密码协议

客户端访问控制器，客户端和控制器有一组相同的加密因子，为128位的16进制字符串，客户端取当前时间的秒数 如 2019-01-01 12:24:52,

我们取3秒为时间片52-52%3=51

客户端的时间因子是 2019-01-01 12:24:51

客户端的时间因子算出一个hash码 与加密因子的数量取模来选定一个加密因子，客户端使用 时间因子 + 机密因子 + 当前设备的ip地址 取128位hash作为sm4密钥加密鉴权数据。

\* 我们假定控制器在1秒后接到数据2019-01-01 12:24:53，则 53-53%3=51

\* 那么服务端的时间因子有两个：

第一个：2019-01-01 12:24:51

第一个：2019-01-01 12:24:48

\* 服务端使用这两个时间因子计算出来的sm4密钥去解密报文来获得明文报文中包含（时间因子 + 密码的）hash

\* 用户验证通过后控制器通知网关对客户端ip + 用户的数据开放端口6秒

\* 客户端访问网关生成一对sm2的私钥个公钥，网关放置私钥和客户端放置公钥

\* 客户端与网关进行tls握手，握手过程中使用sm2非对称加密传输数据

\* 客户端与网关握手过程中生成一个128位的sm4密钥

\* 握手成功后建立一个通讯隧道，通讯隧道数据交互使用sm4对称加密

\* 客户端与网关的握手要在控制器鉴权完毕后的时间段内完成，超时网关拒绝服务。

（4）角色及其管理策略

本软件使用白名单模式控制，通过本sdp访问的应用必须维护定义。

软件设置了软件组，方便分组管理

用户设置了用户组可以独立设置用户访问应用或应用组的权限

用户组访问应用或应用组的权限：当客户端通过隧道访问应用的时候网关就会根据当前用户的权限放通或者阻断连接。

（5）软件安全

自主设计开发的软件安全性设计和完整性保护措施，所使用服务器网关和控制器运行在centos7 以上；数据库使用的是postgresql9 以上；服务器上的软件使centos标准源上安装， 配置好防火墙策略，禁止sdp系统无关的端口开放。

## 软件部署架构或方案，包括硬件、操作系统、数据库、中间件、私有化部署/SAAS部署、对接协议等内容。

**无**

## 3、开发架构或技术栈介绍，包括微服务、分布式、前端、后端、UIUE、大屏/驾驶舱等；

**无**

## 4、数据库架构，包括数据采集、汇聚、治理、归类、建模、调用等流程；

**无**

## 底层使用的开源能力，比如GIS地图、3D模型、数据治理等；

**无**

## 二次开发流程，在已有平台或产品的基础上怎么实现迭代开发；

**无**

## 二次部署流程，在遇到类似项目时，如何把已有的平台或产品重新部署，以及完成二次开发满足客户需求。

**无**

## 8、是否愿意在已有的平台基础上，依照移动的UI要求进行改造，实现产品风格统一。

**是**

## 产品是否可以进行应用功能级的解耦，形成标准接口，以微服务的方式和移动行业平台实现对接，提供对应的解耦方案。

**是**