
# 安当统一身份认证平台白皮书

公司名称：上海安当技术有限公司

公司地址：上海市松江区泗泾镇赵非公路 51 号 18 幢 1 层 116（总部）

邮政编码：201601

联系电话：15830600995

©版权所有 2022 上海安当技术有限公司，保留所有权利　2022 年 12 月

未经上海安当技术有限公司书面许可，本白皮书任何部分的内容不得被复制或抄袭用于任何目的。

本白皮书并不暗示上海安当技术有限公司提供的任何产品或服务或其功能与本白皮书描述的内容完全一致。

本白皮书的内容在未经通知的情形下可能会发生改变，敬请留意。

目录

# 1 概述

## 1.1 背景

随着云计算的飞速发展，越来越多的云应用、云服务充斥在日常的工作当中。人们在享受信息化带来的便捷的同时，也遭受着应用系统反复登录，工作入口来回切换，记忆多套账号密码等诸多烦恼。

企业的发展越来越快，企业内部需要使用到的企业级应用也越来越多，每个应用都需要账号密码，是带给每个员工最直接的困扰。为了安全起见，员工普遍采用两种方法来确保密码的安全：一使用长密码，密码中要求包含数字、字母和符号；二不能将同一套用户名和密码应用于多个应用系统或者网站。理论上这些方法是可行的，但实际操作起来却非常困难，每过一段时间账号密码还需重置一次，更加剧了这种繁琐。

另外，企业上云使得企业的 IT 管理员也在不断应对企业内部各种不同的关键性变革，面临十分沉重的压力。一方面要在混合 IT 环境下管理数量众多的应用系统，另一方面还要为企业员工用户提供良好的使用体验。应用系统数目的不断增多，其所带来的访问权限管理的威胁同样与日俱增。

上海安当技术有限公司，作为数据安全领域的创新解决方案供应商，提供了具备用户身份管理和企业内部管理的 ASP 产品，方便企业或集成商在

身份认证方案上快速集成安全的多因素认证和统一身份认证的解决方案，以及满足等保 2.0 的检测要求。

## 1.2 产品介绍

安当 ASP（Authentication Service Platform）身份认证管理平台为企业用户提供集中式的身份、权限、应用管理服务，产品支持云部署或者本地私有化部署。主要功能包括：

- MFA 多因素身份认证:支持 FIDO2、OTP、USBkey 等多种验证方式，可以根据客户不同业务场景及安全需要选择适合的二次验证方式
- SSO 单点登录:企业人员可以通过一套用户名和密码即可访问所有的工作相关应用系统。
- 账号全生命周期管理:实现人员职位流转后其身份信息在应用系统中的自动化流转。
- 安全审计:实时记录企业信息的安全状况，辨别异常访问和潜在威胁
- RADIUS 服务器:支持 RADIUS 普通登录和 OTP 登录
- 组织架构管理，人员管理：支持自由的部门管理，人员管理。支持 RBAC 模式的权限管理
- 应用管理:用户自由创建应用及用户业务系统

## 1.3 行业规范

安当 ASP 在云上为企业和开发者提供专业的身份认证和授权服务，遵循了如下几个行业标准:

- OAuth 2.0: 一种授权标准，允许用户在一个站点向其他站点授予对其资源的有限访问权限，而无需获得其凭证（通常是账号密码）。
- Open ID Connect: 这是 OAuth 2.0 的一个超集，他在 OAuth 2.0 之上提供了更多用户信息和获取权限和标准，比如他定义了用户的头像为 picture。
- JSON Web Tokens: 一种开放标准，主要用来安全的传输信息，他的格式非常紧凑和独立，解析之后是一种 JSON 格式。
- LDAP: 你可以把轻量目录访问协议（Lightweight Directory Access Protocol，简称 LDAP）理解为一个树型的用来存储用户和组织信息的数据库，常被用来做单点登录（ SSO ）和企业员工信息管理。

# 2 解决方案

ASP 针对不同的应用场景提供了不同的集成方案，可以根据客户需求合理的选择哪种方案进行集成。本节我们针对几种常见的集成常见进行说明:

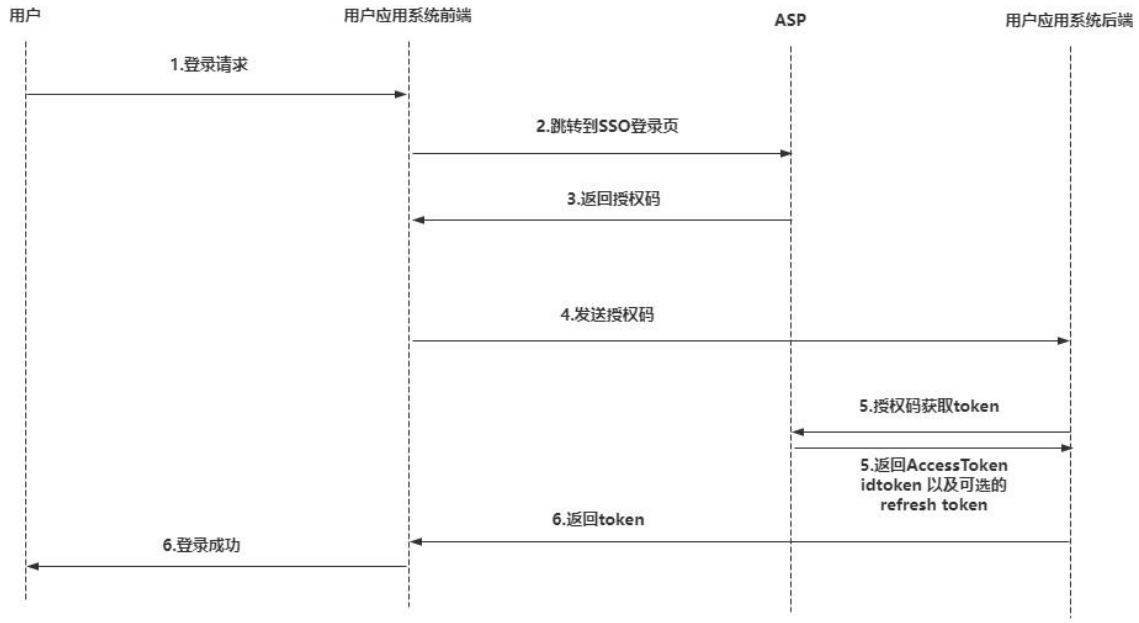
## 2.1 有用户源场景

有用户源是指用户系统具备完善的前端和后端。对于此类型的应用，我们建议使用授权码的方式，前端进行修改集成获取授权码，后端进行授权码换取 token 操作来完成集成。

授权码模式适合应用具备后端服务器的场景。授权码模式要求应用必须能够安全存储密钥，用于后续使用授权码换 Access Token。授权码模式需

要通过浏览器与终端用户交互完成认证授权，然后通过浏览器重定向将授权码发送到后端服务，之后进行授权码换 Token 以及 Token 换用户信息。

对于客户的系统具备后端服务器的场景，在集成 ASP 的时候可以采用下方模式进行集成:




说明:使用次方案大致分为如下几步:

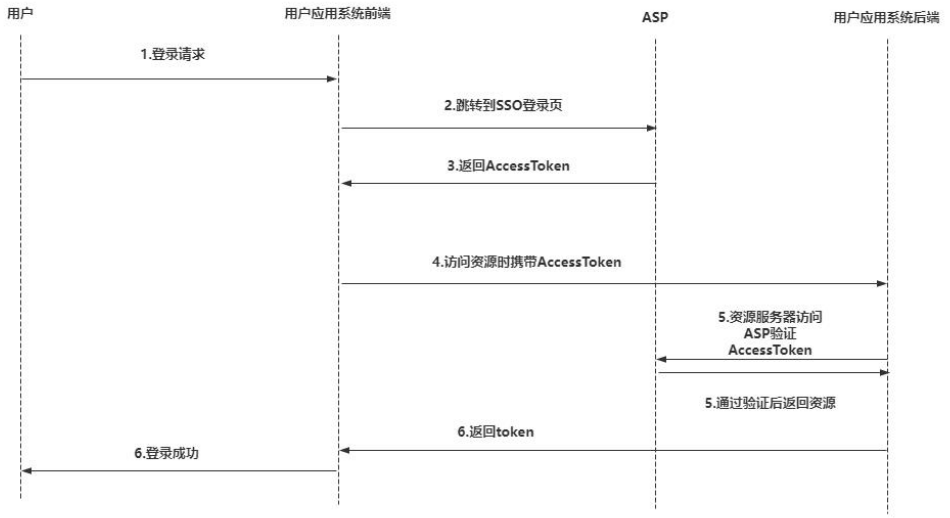
1. 用户发起登录请求；
2. 跳转至 SSO 登录页；
3. ASP 返回授权码；
4. 前端收到授权码并将授权码发送到后端；
5. 后端调用授权码换取 token 的 api 获取 AccessToken idToken 以及可选的 Refresh Token，后端通过解析 IdToken 获取用户信息并在用户表中查找用户对应关系，如果没有用户则创建用户和用户对应关系，然后生成自己的 Token 或者使用换取的 AccessToken，将 token 返回给前端
6. 前端存储 token 到本地，之后使用此 token 进行资源调用。

## 2.2　无用户源场景

无用户源是指的用户在登录自己系统时后端不会做验证，前端只需要进行校验。对于此类型的系统我们建议使用隐藏式模式进行集成，后端无需存储安全密钥，只需要在前端进行部分修改即可实现。

隐式模式适合不能安全存储密钥的场景（例如前端浏览器）。在隐式模式中，应用不需要使用 code 换 token，无需请求 /token 端点，AccessToken 和 IdToken 会直接从认证端点返回。

对于有自己的系统但是没有具备后端身份验证服务的客户在使用 ASP 进行单点登录时，可以采用下方描述模式进行集成:




说明：使用此方案大致分为 6 步:

1. 拼接登录授权链接并引导用户点击
2. 跳转 ASP 进行认证
3. 在前端直接获取 AccessToken 并存储以便于后续使用

4. 访问资源时携带 AccessToken

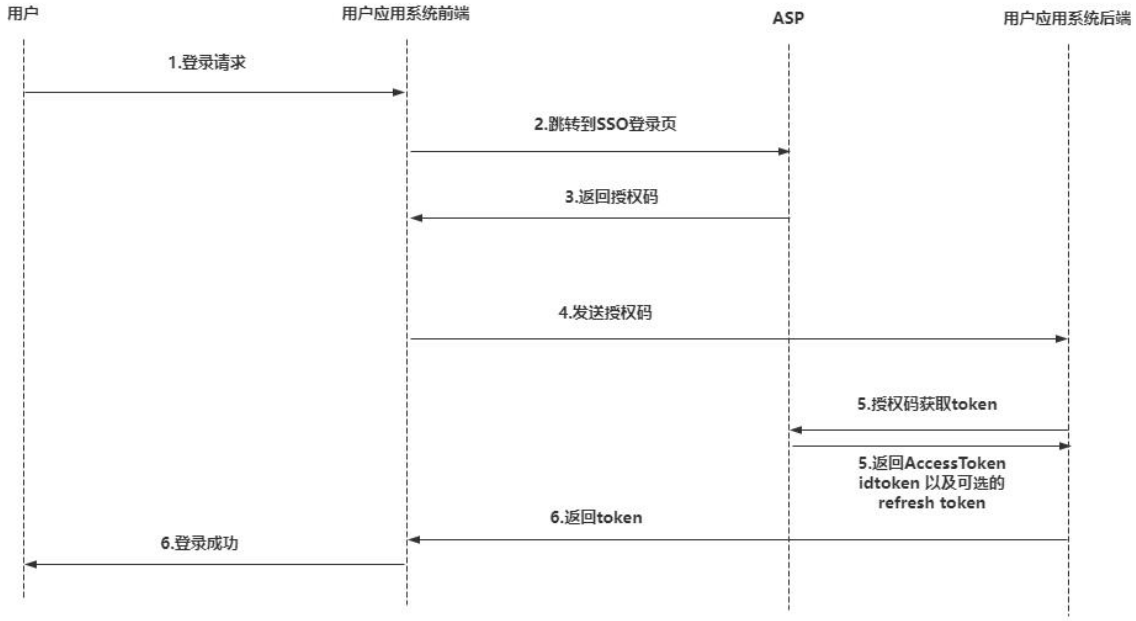
5. 资源服务器与 ASP 验证成功返回资源

用户获得资源继续其他操作

## 2.3 第三方应用场景

对于客户使用的其他应用，支持标准协议对接的第三方应用如(Git)，可以通过 ASP 提供的对接模式来进行对接 ASP。

对于使用第三方应用的用户在集成 ASP 的时候可以采用下方模式进行集成:




1. 用户发起登录请求；

2. 跳转至 SSO 登录页；

3. ASP 返回授权码；

4. 前端收到授权码并将授权码发送到后端；

5. 后端调用授权码换取 token 的 api 获取 AccessToken idToken 以及可选的 Refresh Token，后端通过解析 IdToken 获取用户信息并在用户表

中查找用户对应关系，如果没有用户则创建用户和用户对应关系，然后生成自己的 Token 或者使用换取的 AccessToken，将 token 返回给前端；

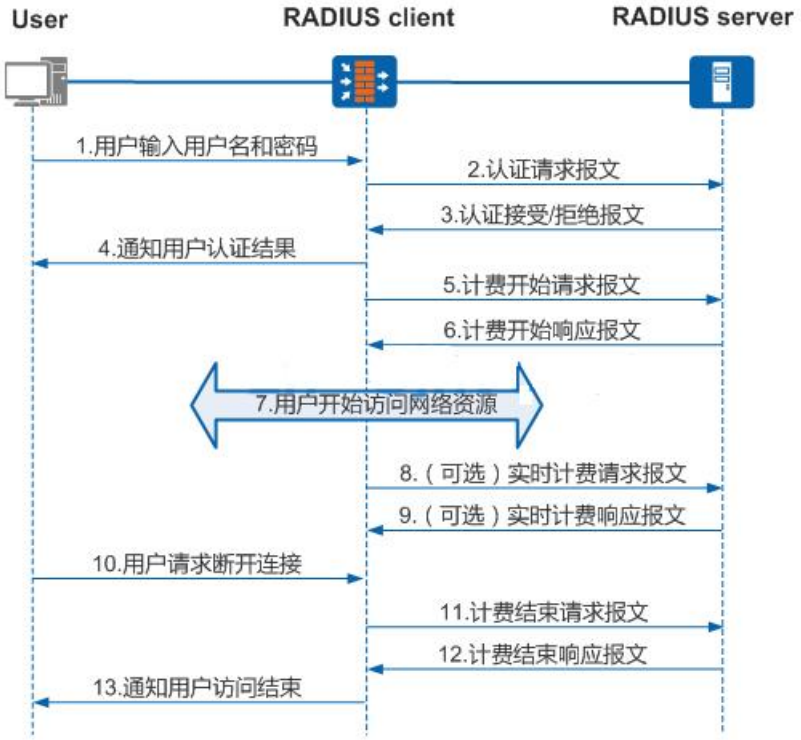
6. 前端存储 token 到本地，之后使用此 token 进行资源调用。

## 2.4 Radius 认证场景

一般的用户为了加强登录安全，通常会采用第三方 Radius 双因素/双因子（2FA）身份认证，以此做到等保合规和安全加固；之所以选择 Radius 认证，是因为 Radius 具有非常广泛的兼容性以及独有的优势。

安当 ASP 提供了 RADIUS 服务器，用户可以选择开启服务器来进行 RADIUS 服务器认证和 RADIUS OTP 认证。

认证原理如下：




认证原理

# 3 功能说明

## 3.1 应用管理

用户作为开发者进入 ASP 开发平台，可以创建任意的应用数量来添加自己企业的应用系统。主要功能包括:

- 创建应用: 创建一个应用获取 APPID 和密钥用于前端集成。
- 编辑应用: 编辑应用可以查看应用基本信息，应用配置，设置访问授权等。
- 删除应用: 删除已添加的应用进行删除操作。

## 3.2 身份源管理

开发者可以根据自己系统的身份来源来进行配置，创建一个用户源，包括 OIDC、LDAP 等。

OIDC:OIDC 是一个基于 OAuth2 协议的身份认证标准协议。

LDAP:LDAP 是一个开放的，中立的，工业标准的应用协议。

## 3.3 权限管理

开发者可以根据自己系统的权限进行资源的分配，角色权限的分配，给与角色授权等操作。用户身份认证成功之后，通过授权来确定：

- 用户能够使用的命令
- 用户能够访问的资源
- 用户能够获取的信息

## 3.4 组织机构

开发者可以使用此功能进行用户管理，部门管理，其中包括创建用户、编辑用户、禁用用户、创建部门、创建子部门等。

## 3.5 安全管理

开发者可以在通用安全项进行登录注册控制的设置，可以设置添加多因素认证，可以配置是否开启 radius 服务器认证等。

### 3.5.1 多因素认证

多因素认证（Multi Factor Authentication，简称 MFA）是一种非常简单的安全实践方法，能够在用户名称和密码之外再额外增加一层保护。启用 MFA 后，用户登录时，除了需要提供用户名和密码外（第一次身份验证），还需要进行第二次身份验证，多因素认证结合起来将为你的帐号和资源提供更高的安全保护。

目前安当 ASP 提供了如下几种方式:


#### 3.5.1.1 支持 FIDO

**FIDO：**FIDO 既指 FIDO2，FIDO2 提供基于 WebAuthn 标准的 FIDO2 线上快速身份验证客户端实现，为应用及浏览器提供安卓 Java API，支持使用 USB、NFC、蓝牙漫游认证器，以及指纹和 3D 面容的平台认证器，进行安全认证。

**产品规格**

| | |
|---|---|
| 尺寸 | 44×41×3mm |
| 重量 | 2.7g |
| 接口类型 | USB |
| 按钮 | 触摸类型 |
| 指示器 | 绿色LED灯 |
| 通信协议 | HID |
| 安全算法 | ECDSA, SHA256 |
| 工作电压 | 5.0V |
| 额定电流 | 35mA |
| 功率 | 0.17W |
| 工作温度 | -10℃~60℃ |
| 贮存温度 | -20℃~71℃ |
| 内存数据保留 | 至少10年 |

3.5.1.2　支持 OTP

**OTP**：一次性密码（One Time Password，简称 OTP），又称“一次性口令”，是指只能使用一次的密码。一次性密码是根据专门算法、每隔 60 秒生成一个不可预测的随机数字组合，iKEY 一次性密码已在金融、电信、网游等领域被广泛应用，有效地保护了用户的安全。

**OTP（国密）**:支持国密算法的一次性口令。

安当OTP令牌:
1.支持多种手机系统，Android、iOS;
2.支持产生基于时间，验证服务端动态口令;
3.支持6位动态口令，支持产生30秒时间间接动态口令;
4.支持通用算法，支持国密算法;
5.完全兼容谷歌，微软腾讯等第三方身份验证器。

**技术参数**

| | |
|---|---|
| 标准 | 符合OATH相关标准 |
| 内置安全算法 | 符合OATH组织的标准TOTP算法 |
| 密钥类型 | base32、hex编码 |
| 口令长度 | 6位 |
| 口令周期 | 30s |
| 算法支持 | SHA1、SHA256、SHA512、SHA224、SHA384、SM3 |
| 认证模式 | 时间型动态口令认证、验证服务端 |

#### 3.5.1.3　支持硬 Key

USB Key，采用国产高性能智能卡芯片，内置 SSF33、SM1、SM2、SM3、SM4 等国产算法，支持高速数据国密算法加解密，提供 CSP 以及 PKCS11 和国密接口，完全符合国家密码管理局关于“密钥不落地”的技术规范要求

**技术参数**

| | |
|---|---|
| USB接口 | USB2.0 |
| 保存温度 | -10~70℃ |
| 工作温度 | 0~70℃ |
| 保存湿度 | 5%~95% |
| UKEY空间 | 256KB |
| 算法性能 | SM1：7.2Mbps<br>SM2：签名 83.33c/s，验签 50c/s<br>SM3：5.44Mbps<br>SM4：6.96Mbps |
| 加密算法 | 支持 SM1/SM2/SM3/SM4 |
| 支持平台 | Windows XP 及以上系列，Linux 及基于 Linux 的国产操作系统 |

**UKEY KeyId**：UKEY keyId 指的是 UKEY 唯一序列号，通过绑定 UKEY 唯一 ID 进行身份认证。

**UKEY PUB**：通过 UKEY 内部存储的用户密钥对来进行签名验签操作达到用户身份认证。

**UKEY 证书**：支持 UKEY 内存放证书认证。

## 3.6　日志管理

安全性和合规性对一个企业或者组织的运作至关重要，是企业和组织运转的基石。想要实现安全与合规并不简单，ASP 提供操作审计功能以帮助企业和组织在业务不断增长的同时保持安全与合规。

ASP 提供的日志审计功能分为两个方面：

- 管理员行为的操作日志：你可以得到所有管理员用户通过身份管理平台的行为
- 用户行为的日志：可以清晰得还原出用户在平台中的行为，以支持企业的合规管理，同时可用作事件发生后的回溯和定责。

# 4 使用说明

## 4.1 单点登录

使用安当 ASP 进行进行单点登录主要包括以下步骤:

**1. 注册安当 ASP 开放平台开发者**

安当提供了 ASP 开发平台，用户可以通过注册 ASP 开放平台成为一名开发者，注册成为开发者之后即可登录使用功能。

**2. 创建应用**

登录开发者后台后即可使用平台功能，每个开发者对应一个企业，每个应用对应一个企业应用系统，开发者可以创建任意数量的应用进行集成自己应用实现想要的功能。

**3. 前后端集成**

开发者创建应用后会获得 APPID 和 Secret，开发者可以通过提供的集成示例进行前后端集成。

**4. 用户登录**

集成完成后，用户访问自己的业务系统登录页即可跳转至 SSO 登录

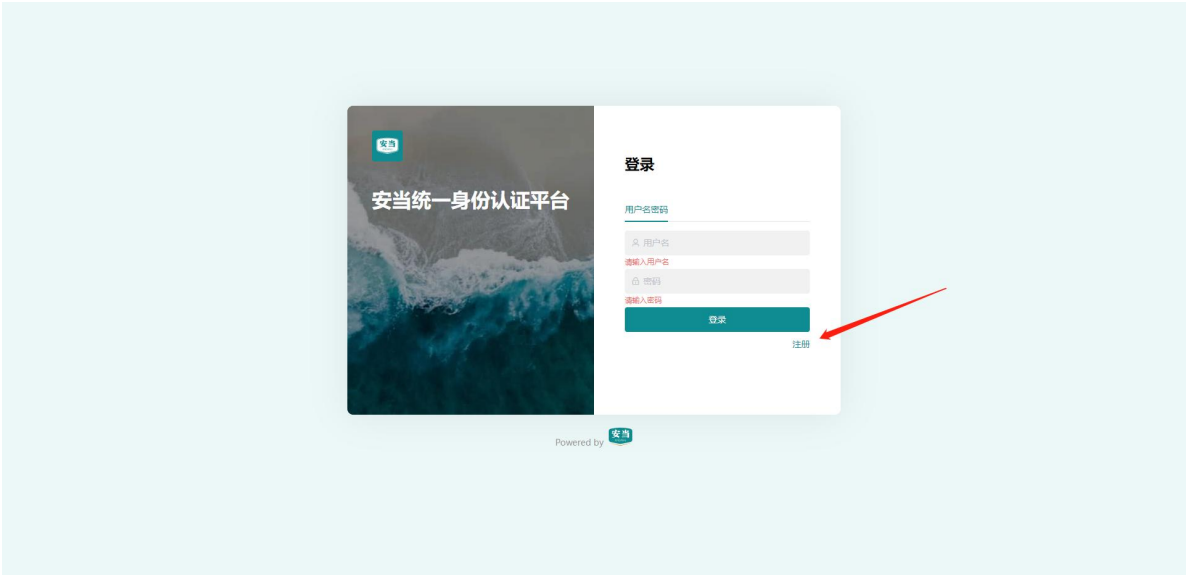
页，用户登录后 token 会被 ASP 所保留确保下一次登录系统或登录绑定的其他应用直接可以登录，从而完成 SSO 登录认证。

## 4.2 多因素认证

使用安当 ASP 进行进行多因素认证主要包括以下步骤:

1. **注册安当 ASP 开放平台开发者**

安当提供了 ASP 开发平台，用户可以通过注册 ASP 开放平台成为一名开发者，注册成为开发者之后即可登录使用此功能。

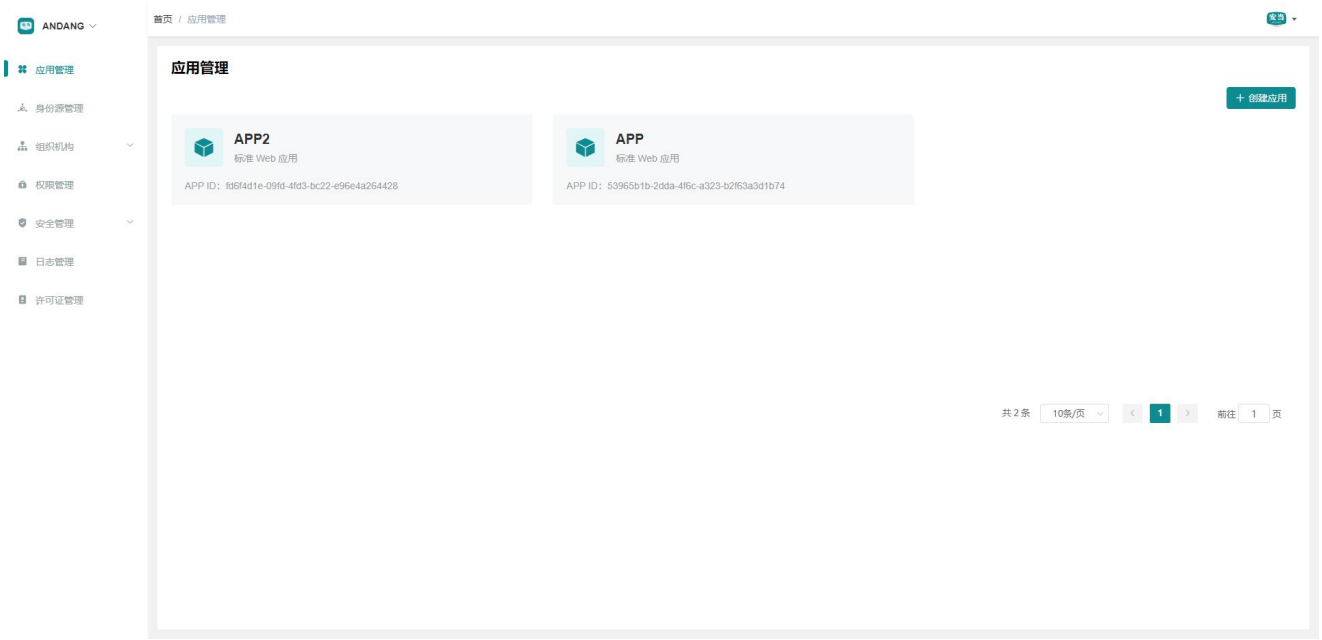



2. **创建应用**

登录开发者后台后即可使用平台功能，每个开发者对应一个企业

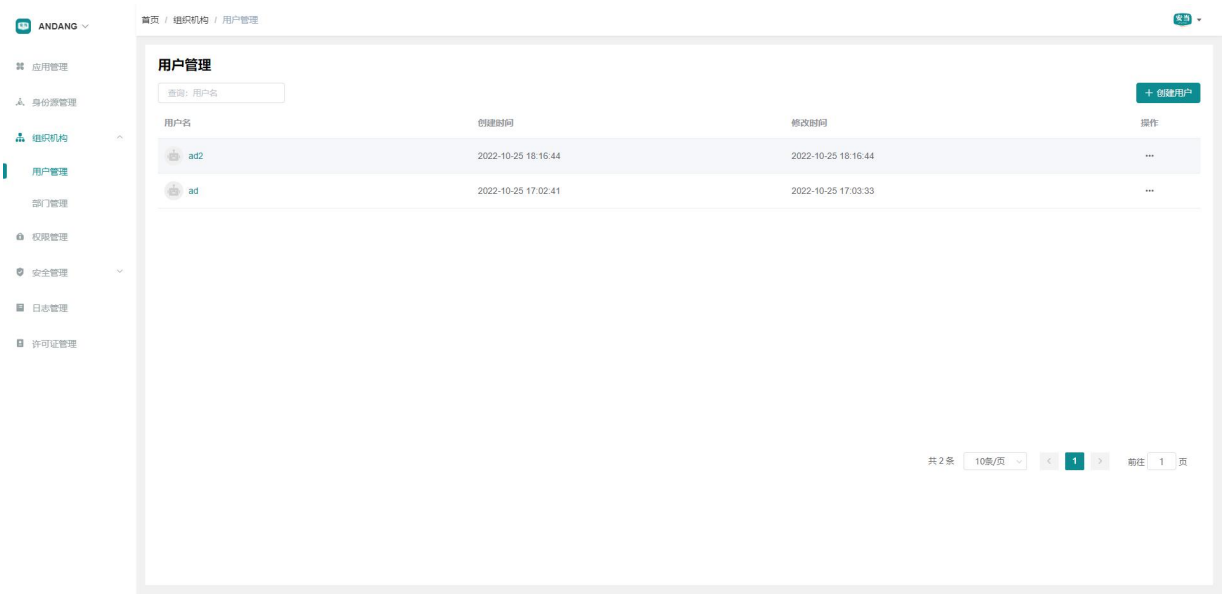

### 3. 创建应用

每个应用对应一个企业应用系统，开发者可以创建任意数量的应用进行集成自己应用实现想要的功能。



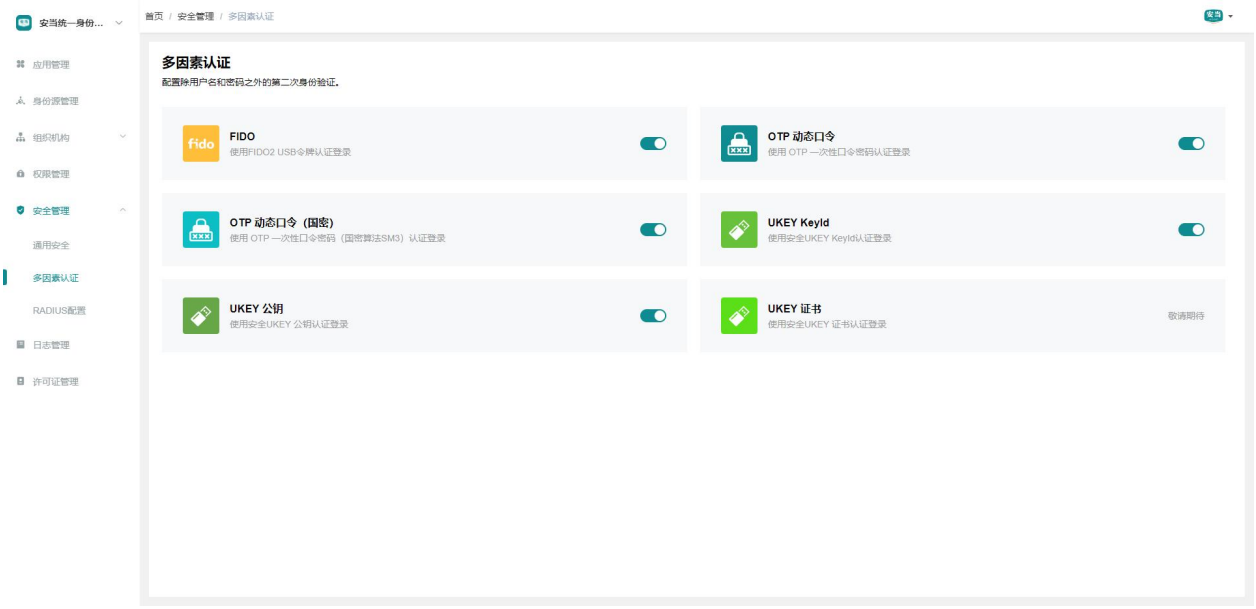

### 4. 添加用户

开发者可以在应用管理为企业添加用户




## 5. 选择开启 MFA 方式

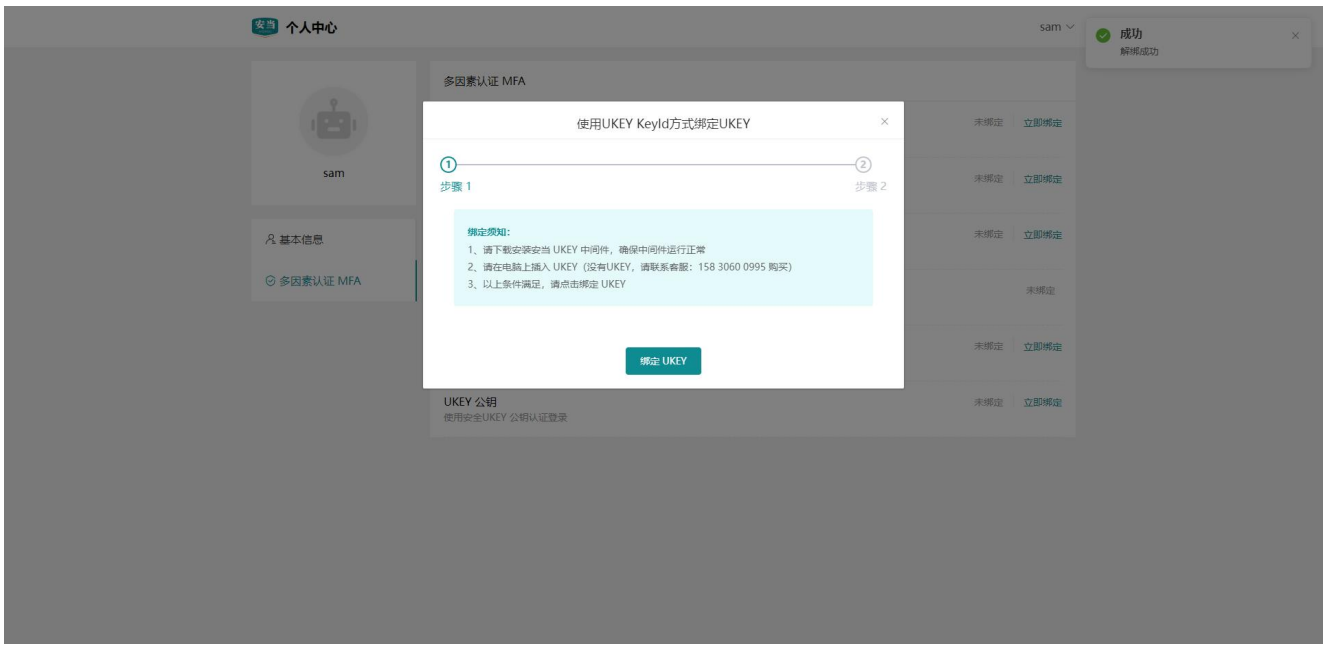
根据需求可以选择开启 MFA 认证方式




## 6. 用户绑定

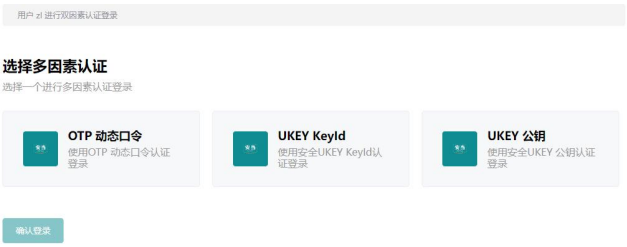
集成完成后，用户访问自己的业务系统登录页即可跳转至 SSO 登录页，如果开启了多因素认证则会提示用户绑定令牌，用户登录时即可在

SSO 登录页进行多因素认证方式的绑定。




## 7. 用户登录

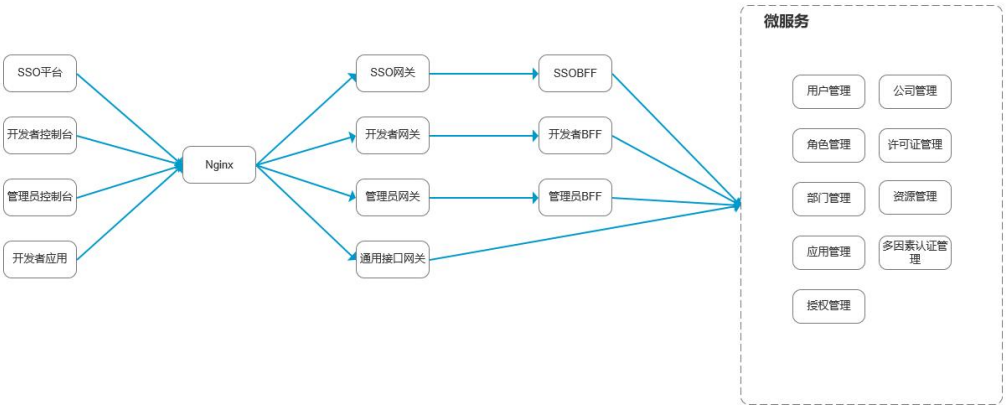
绑定完多因素认证令牌后，在登录的时候即可使用令牌进行多因素身份验证。




## 4.3 部署方式

### 4.3.1 私有化部署

ASP 提供了便携的部署方式，只需要简单的几步即可实现私有化部署。




项目架构图

### 4.3.2 公有云使用

对于想使用公有云的用户，ASP 提供了开放的接口可进行调用，用户只需要在前端进行简单的修改后端进行调用接口即可实现公有云的使用。

# 5 服务端集成示例

后端集成使用 API 方式集成，更加便捷可靠。

## 5.1 换取 token 身份验证方式

针对不同场景，换取 token 时需要提供不同的身份验证方式，防止非法伪造请求：

- client_secret_post: 在请求体 Body 中传递应用密钥 client_secret。

- client_secret_basic:在 authorization 请求头中传递 Basic encodedString，其中 encodedString 是 client_id:client_secret 的 Base64 编码值。
- none: 不需要提供认证方式，一般适用于直接在前端单页应用这类无法安全存储应用密钥 client_secret 的场景。

## 5.2 不同授权模式获取 token 的示例

下面是不同模式下获取 id_token 和 access_token 的示例（以 client_secret_post 方式为例）。

### 5.2.1授权码模式

```
curl --request POST 'https://<服务域名>/api/v1/sso/oidc/token' \
  --header 'Content-Type: application/x-www-form-urlencoded' \
  --data client_id=xxxx \
  --data client_secret=xxxx \
  --data grant_type=authorization_code \
  --data redirect_uri=https://baidu.com' \
  --data code=NBS4GK4rZli1jtWcVy7yqd8AT3PxyENhu8XbT3oBMko
```

### 5.2.2授权码+PEC 模式

```
curl --request POST \
  --url https://<服务域名>/api/v1/sso/oidc/token \
  --header 'Content-Type: application/x-www-form-urlencoded' \
  --data client_id=xxxx \
  --data client_secret=xxxx \
  --data grant_type=authorization_code \
  --data redirect_uri=https://baidu.com \
  --data code=NBS4GK4rZli1jtWcVy7yqd8AT3PxyENhu8XbT3oBMko \
  --data code_verifier=dBjftJeZ4CVP-mB92K27uhbUJU1p1r_wW1gFWFOEjXk
```

### 5.2.3 Client Credentials 模式（编程访问模式）

```
curl --request POST \
  --url https://<服务域名>/api/v1/sso/oidc/token \
  --header 'Content-Type: application/x-www-form-urlencoded' \
  --data client_id=xxxx \
  --data client_secret=xxxx \
  --data grant_type=client_credentials
```

### 5.2.4 刷新 token

```
curl --request POST \
  --url https://<服务域名>/api/v1/sso/oidc/token \
  --header 'Content-Type: application/x-www-form-urlencoded' \
  --data client_id=xxxx \
  --data client_secret=xxxx \
  --data grant_type=refresh_token\
  --data refresh_token=xxxx
```

# 6 前端集成示例

```
<script >
  import ADukey from "/js/sdk.js";  // 引入 asp 的 sdk

  // 初始化-配置跳转 sso 的信息
  ADukey.config = {
    response_type: "nocode",  // OIDC 授权方式（授权码模式：code，隐藏式模式：nocode）
    client_id: "011f2fa2-4bb2-462a-99c1-0383fe0b1e64",  // 应用 ID
    redirect_uri: "http://192.168.0.121:8084" // 发起登录的 URL（系统的 url）
  };
  ADukey.init().then(res => {
    console.log("res", res);
    location.href =  ADukey.config.redirect_uri
  })
    .catch(err => {
      console.log("err", err);
      reject(err);
```

```
    });
</script>
```

# 7 性能设计

系统性能，由于单设备性能局限，对于更高的性能要求，可以通过集群的方式来满足。考虑到集群中主机数量到达一定程度后会有 80%的衰减，通过负载均衡、LB 等手段，可以满足项目的性能要求。

针对单设备性能指标如下：

1) 2FA 平台稳定性 MTBF(平均无故障时间间隔)在服服务器稳定情况下>20000 小时；
2) 平台接入设备数无限制；
3) 支持最大服务并发访问量不小于 5000 个，在系统达到最大峰值时表现稳定；
4) 支持最大处理登录请求（TPS）≥1500/秒；
5) 认证请求 RT<100ms；