ALIBABA CLOUD

阿里云

应用身份服务 IDaaS

身份安全管理导论

文档版本：20230215

阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击**设置**> **网络**> **设置网络类型**。
粗体	表示按键、菜单、页面名称等UI元素。	在**结果确认**页面，单击**确定**。
`Courier字体`	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.概论

1.1. 单点登录 和 身份联邦

什么是「身份联邦」？

「身份联邦」并不是一个新词，我们几乎天天都会在使用「身份联邦」概念下的应用场景。但在身份安全领域，即便是在行业内，很多人对「单点登录」的认知都是不准确的，对「身份联邦」甚至没有任何认知，或和「单点登录」的概念混淆不清。而在行业外，可想而知这两个词汇可能带来的困扰。

对于最终使用的用户而言，这两者实现提供的使用体验很相近。实现任意一种，普通用户都只需要登录一次，即可访问所有的应用服务。（当然，对于开发人员和管理人员来讲，身份联邦比单点登录高效地多，也复杂地多。）所以从需求侧，往往对这两个词的混淆不求甚解。

不仅如此，几乎所有新一代的 IAM 统一身份认证平台都会在某种程度上「同时」实现这两个概念，而又不加说明，这使得两个概念的边界更加模糊。为了让阿里云 IDaaS 的客户对这两个概念有所区分，增强对 IAM 产品的认知，本文会梳理两个概念的区别和优劣。

我们从单点登录开始说起。

单点登录 Single Sign On（SSO）

单点登录不是一个技术实现方式的定义，而是代表着一种特定用户认证体验的观念。一般而言，单点登录指的是：

> *用户只需要认证一次，即可用一个身份访问所有在当前可信环境中的所有应用。*

只要实现了这种效果，就可以定义为实现了「单点登录」。企业常见的实现方式多种多样，但往往并没有过多考虑安全性和扩展性，远远低估了一套完整统一身份认证系统的复杂度，在实现后也往往需要重新设计或进行外部采购。实现方式包括使用 cookie、jsonp、页面重定向等等，也包括了一些单点登录标准协议例如 SAML、OIDC 等。（这里会触及到一点「身份联邦」的概念，但远不完整。）

有一些人会把「用户访问所有的应用都共享一套用户名+密码」作为单点登录的另一种形式，用户每访问不同应用都需要单独输入一次统一共享用户名+密码，我们认为这种实现方式是「假单点登录」（虽然在一些场景中我们必须通过「密码代填」来实现 SSO），在这里不多讨论，免得让概念更加错综复杂。

往往企业实现单点登录后，各系统身份仍然相对隔离没有打通，很可能应用系统自己的账密登录入口也仍然保留着，各系统仍然单独维护着独立的账号体系。这种情况的问题如下：

1. 由于一套账号即可通用所有应用，需要依赖每个应用独立维护多因素认证，以达到最低安全要求，避免单点爆破，消耗研发资源。
2. 员工离职后，需要手动为其从每个应用系统中删除/禁用信息。如有遗忘：
   i. 可能会造成长期信息泄露，造成潜在安全隐患。
   ii. 可能会导致企业长期为离职员工维护企业软件 license，造成额外开支。

于此需求，「身份联邦」与「单点登录」的区别浮现出来。

身份联邦 Identity Federation

「身份联邦」是实现「单点登录」的一种标准方式，但「单点登录」远不是「身份联邦」的唯一目的。只要实现了「身份联邦」，即必然已经实现了「单点登录」，而反之不成立。

「身份联邦」的目的是用标准协议来打通不同安全域之间的用户身份，在跨域、跨产品、跨公司的场景中实现身份信息共享，包含了一系列认证、授权、身份治理、跨域身份同步、统一 license 管理、跨域字段转换的策略、协议和最佳实现。使用实现了「身份联邦」的 IAM 服务，将可以在企业 IT 架构中将身份与权限管理层完整抽离出来，并统一到一个安全平台进行管理。「身份联邦」是一个涵盖面非常广（且仍然在快速完善中）的词。

在「身份联邦」的实现中，浮现出了一个身份和权限访问的中央处理机制，负责统筹所有应用的访问服务。各应用系统不再单独维护独立的账号体系，全部转化为统一的登录入口，登录后将用户分发到希望访问的目标应用中去。

这个处理身份和权限管理的核心即为 IDP Identity Provider，在服务中处于身份提供方。其他的业务服务为 SP Service Provider，服务提供方。

身份提供方 IDP Identity Provider

这样一来，每个单独的业务应用将完全无需关注安全的「身份治理」，包括二次认证、密码强度、定期改密、风险识别等等能力，即可通过统一在 IDP 上增加安全机制，实现不需要应用任何改造，即可满足任意场景的访问安全需要。

除此外，IDP 可以统一管理所有用户的隐私设置以及信息共享，很多国家对各自公民的隐私信息都有立法保护，著名的有欧洲的 GDPR 以及加州的 CCPA。出于对合规的诉求，外国企业（特别是面向顾客提供服务的零售、娱乐等行业）往往对于一个统一登录入口、统一管理所有应用访问安全性有刚性需要。国内虽然立法上相对落后，但明确正在加速完善网络安全、隐私安全的保障，很快由安全合规推进的 IDP 需求会迎来爆发期。

由于「身份联邦」强调的互操作性（Interoperability，即系统之间按照标准协议对接而获得的一种解耦性），在「身份联邦」体系中实现的「单点登录」不能再依赖于 cookie，不可限制在特定安全域内，需要统一使用 SAML、OIDC、CAS 等标准协议实现跨域（或非跨域）单点登录、单点登出的能力。由此一来，按照标准协议实施的应用，有能力与任何支持 SAML、OIDC、CAS 的 IDP 进行配置集成，实现依赖于任意 IDP 身份源的单点登录。

同步 Provisioning

单点登录的实现是不需要依赖于同步的，但身份联邦涉及到多套身份体系之间的关联、映射和集中管理，需要同步来作为 IDP 对外连接的关键触角。

企业的 IDP 往往只有一个（或有限的、架构需要的几个），但身份系统在绝大部分关键系统中都存在。如果希望实现在 IDP 中对账号的「一处修改、处处生效」的话，同步机制必不可少。

一个很常见的场景，企业现有的 IAM 作为身份权限管理核心使用，但针对用户的增删改查等操作，仍然需要同步到 AD 中，因为仍然会有应用依赖于 AD 作为他们的 IDP。

IDP 需要具备类似 LDAP、SCIM 这样的标准身份提供、身份交换的协议，与其他的 IDP、SP 在身份层面打通，实现身份联邦治理。

权限管理 Permission Management

有一些 IDP 会延展现有能力，提供统一的权限系统管理能力。这是 FID（Federated Identity Management 联邦身份管理）的自然延伸。当企业中的核心 IDP 包揽了所有应用的身份对接后，员工的生命周期往往在 IDP 中进行集中管理，设置员工权限往往是下一个步骤。

员工能够访问哪些应用，能访问应用中的哪些菜单、按钮甚至数据，都可以算在身份联邦体系中 IDP 可以提供的权限集中管理能力范围内。

社交账号登录 Social Login

「身份联邦」其实很常见，我们天天使用的扫码登录即是一例。

在网络中，支付宝、淘宝、微信等（海外则包括 Facebook、Google 等）存有用户身份信息的平台（IDP 身份提供方），通过 OAuth 协议，将自己的平台用户信息开放给自由注册的第三方调用，并提供统一认证机制（扫码、或者账密登录），统一返回结果给应用（即 SP 服务提供方）。

在 AWS、Google 中都有「联合身份」（Federate Identity）的概念，可以这么理解：「联合身份」是一个动作片段，最终实现的结果是「身份联邦」。

总结 Conclusion

「单点登录」只牵扯到「认证」这一部分的技术实现，可见单点登录的实现是「身份联邦」整体能力的一小块关键部分。「单点登录」强调的是用户与客户端之间的互动，而「身份联邦」则普遍需要包含用户对用户、用户对服务、以及服务之间的互动。

由此我们可以想象，在公有云上，「身份联邦」将会依托于各层面的身份和权限的协议和标准，组成一个网状结构。网上有两种节点，IDP 和 SP（Service Provider，即业务应用）。任意 IDP 和 IDP，IDP 和 SP 之间，可以按照企业客户需求，通过标准协议、标准定义接口，组成新的网络连接，实现企业特有的「身份联邦」体系，并以此实现集中的身份权限管理，极大程度上优化企业内部 IT 架构，降低管理和使用成本，提高企业运转效率。

阿里云 IDaaS 致力于为客户提供完整的身份管理解决方案，详情参见：https://www.aliyun.com/product/idaas

1.2. 身份安全的重要性

身份安全面临的威胁

数据已经成为国家和企业发展的重要资源，随着云化、移动化的进程加快，网络边界变得越来越模糊，在互联网访问网站，在政府或者企业进行办公，我们都需要有一个身份才能访问对应的系统和资源。日益成熟的数据挖掘技术，廉价的存储手段使得个人信息更加透明，充满高利用价值和商业利益的个人身份信息成为各方争夺的资源，一旦这些用户身份信息被泄露或者进行恶意使用，将给政府，企业和个人带来不可挽回的巨大损失。

近年来国内外经常发生各类信息和数据泄露的事件都跟身份安全离不开关系，例如雅虎数据泄密事件、乌克兰电网受攻击事件，万豪酒店数据泄露事件，甚至也不乏因为身份管理不当导致离职员工恶意删除公司资料的案例等。从个人的隐私到企业的商业秘密，甚至是政府国家的核心机密，都出现了不同程度的信息安全问题。

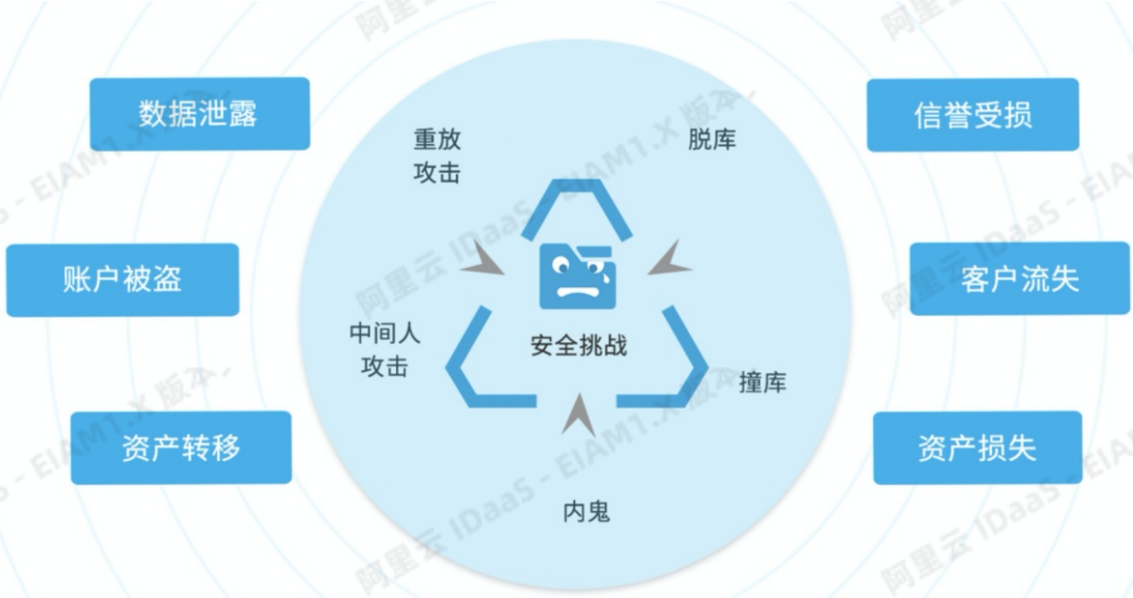
数据泄露简介

当发生敏感的受保护或机密数据可能被未经授权的个人查看，偷窃或使用的事件时，即可定义为数据泄露。数据泄露往往涉及支付卡信息，个人健康信息，个人身份信息，商业秘密或知识产权。

在数据泄露方面，81%的数据泄露涉及撞库或弱口令，都与账号密码被盗用有关。使用传统的账户密码认证方式，很容易被破解和泄露，大部分人习惯使用相同的账户和密码登录不同的系统，绝大部分的人并没有养成定期修改密码的习惯，不但给登录的应用系统带来风险，也给其它应用系统带来巨大的安全隐患。

身份安全给企业带来的挑战

据统计其中89%的数据泄露都是由经济利益或商业间谍驱动的，对于个人，用户信息泄露，则用户账号面临被盗风险，个人隐私及财产安全难以保障；对于企业，数据泄露导致其在公众中的威望和信任度下降，会直接改变客户原有选择倾向，使企业失去一大批已有的潜在的客户。也可以说，在数据信息的作用和地位日渐重要的今天，数据信息的安全关乎企业声誉，公众信任感，经济利益，生死存亡，企业数据信息的安全程度将会影响企业的外部竞争力。



从全球来看，数据泄露的威胁大部分来自于外部恶意攻击，而内部人员的威胁也在上升。从企业角度考虑，确保正确的人在正确的时间，因为正确的原因访问了正确的资源是非常重要的。比如，客服人员可以即时查询客户信息，运营人员可以查询负责产品的运营数据，研发人员可以查看有权限访问的代码。IT部门需要保持对系统访问的严密控制，从而保证只有经过授权的人才能访问企业资源。但是面对庞大的员工、不同应用的账户体系，各应用的访问权限控制，IT部门不但需要花费大量的人力用于维护以上数据，还很容易导致误操作，导致数据泄露的风险。

现在越来越多的外部人员需要访问企业敏感IT资源，如客户、供应商和合作伙伴，那些对外公布的资源通常以Web形式进行发布，外部人员可以很容易的通过浏览器进行访问。但是越来越多的数据泄露事件来源于第三方未经授权的访问，如何保证对外发布数据访问的安全性是越来越大的挑战。

身份认证技术

如何保证以数字身份进行操作的操作者就是这个数字身份合法拥有者，也就是说，保证操作者的物理身份与数字身份相对应，身份认证技术就是为了解决这个问题应运而生的。

身份认证技术是网络通信双方进行真实身份鉴别，也可以说是网络信息安全的看门人。身份认证的目的是鉴别网络使用者的身份是否合法真实，再决定是否给与访问网络资源的授权。对于身份认证不能通过的使用者，网络系统就会阻止其对相关网络的访问，身份认证技术是计算机网络中鉴别使用者身份的有效手段。

随着信息化过程的不断推进以及计算机网络技术的发展，人们的日常生活不断网络变化，资产不断数字化，身份认证逐渐成为保障用户信息安全的基本手段。身份认证安全是任何一个互联网产品的生命线， 身份认证安全依托于网络安全产品和策略的升级。

在真实世界，对用户的身份认证基本方法可以分为以下三种，也可将这几种认证方式相结合。

i. 根据你所知道的信息来证明你的身份，比如账号密码，手势密码，PIN码等
ii. 根据你拥有的东西来证明你的身份，比如移动PKI体系认证，USB key， 智能卡，硬件动态令牌，手机等
iii. 根据独一无二的生物特征来证明你的身份，比如人脸，声纹，指纹，掌纹，虹膜，静脉等

在信息复杂的网络时代，安全与便捷似乎一直是一个矛盾的话题，但在安全的基础上如何便捷的实现身份认证是需要重点关注的。身份认证从单因子认证到多因子认证，从静态认证到动态认证，安全级别逐渐提升。作为防护网络资产的第一道关口，身份认证将起着越来越重要的作用。

1.3. 使用 IDaaS， 你需要做什么准备？

在使用 IDaaS 前，我们首先要搞明白的 8 个问题

一、云上？本地？

为什么要将部署方式作为首要问题，因为这个是在使用 IDaaS 的功能以及服务上是有本质区别的。

云上：也就是 SaaS 服务，方便快捷，开箱即用，不需要运维，解放劳动力。但前提条件就是，需要网络的通畅。大多数企业都存在内网环境，这就意味着我们需要将网络打通，让云上的 IDaaS 服务能够无障碍的和各个应用进行通讯，数据交互。

好处：即买即用，方便快捷，产品服务会不断迭代提升，越来越好。

本地：IDaaS 也可以本地化部署，这样就避免了内网应用通讯问题，但缺点就在于需要维护服务器，并且升级更新较为繁琐。

如果企业对网络没有特殊的要求，作者推荐使用云上 IDaaS ，毕竟，现在很多云产品也能帮助企业更好迁移上云。

二、数据源及数据流向如何？

在使用 IDaaS 之前，我们需要规划整体的数据同步流向。

- 谁是数据源，数据产生者（上游是谁）？
- 下游该同步给哪些业务系统（下游是谁）？

作为数据源，我们要确保唯一，做到一处修改，处处生效，这不仅有利于公司的整体身份中台建设规划，解决身份信息孤岛问题，同时也为后续更多新上线的业务系统，打下身份数据的基础。

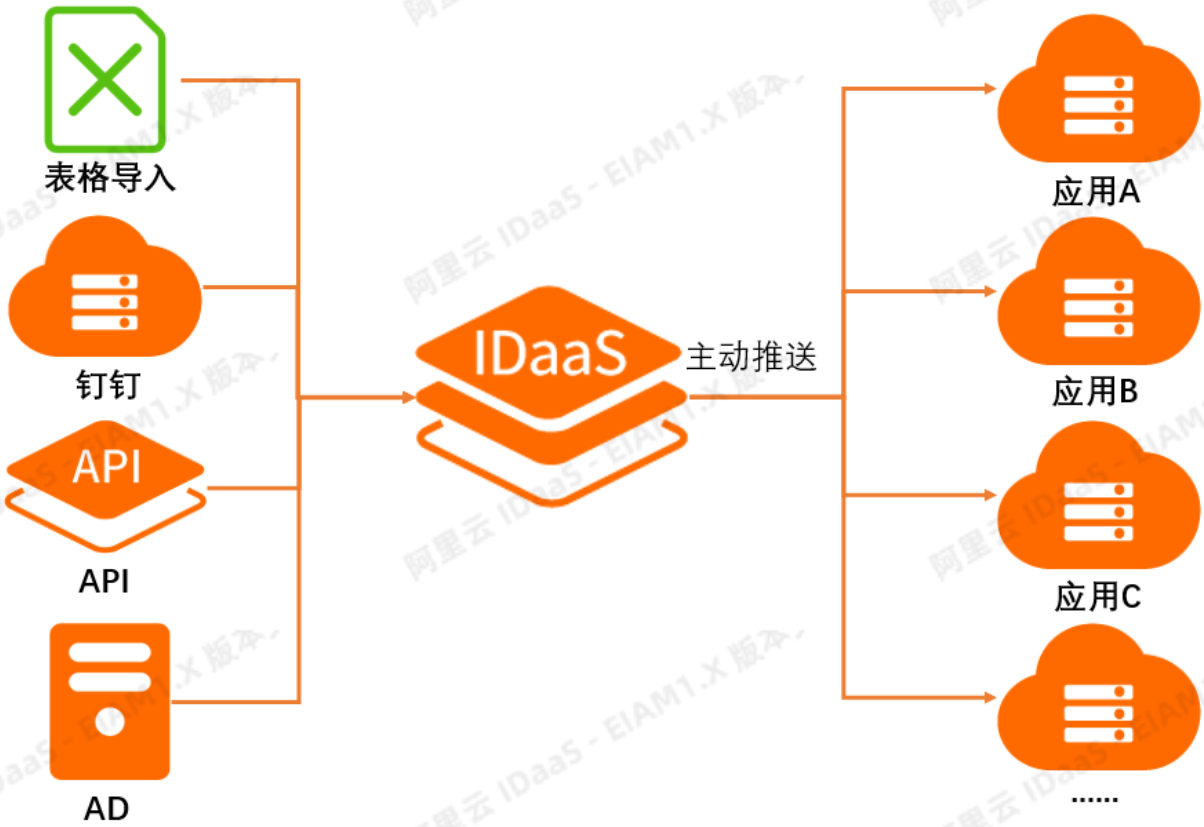
经典数据同步场景

1）IDaaS 作为数据源（ IDaaS -> SP）

IDaaS 作为数据源是，当进行数据的增、删、改等操作都会实时同步至业务系统。通常情况下，为保障数据的一致性，避免数据混乱，业务系统和 IDaaS 只做单项同步对接。

2）第三方业务系统作为主数据源 （SP -> IDaaS -> SP）

IDaaS 同时也支持钉钉、AD、OA、HR 等业务系统作为数据源进行数据同步。通过和 IDaaS 的打通，集中收集数据，然后再推送给业务系统。做到一处修改，处处生效。



三、是否支持多种 SSO 协议？

在这里，我推荐几种不同类型的 SSO 协议。选择正确的协议，会让应用对接 IDaaS 很加简单高效，同时，也考验 IDaaS 的 SSO 协议支持力度。用户在这里需要对不同的产品精心挑选。

我们可以从以下几个角度考虑如何选择 SSO 协议对接。

1）IDaaS 预集成

在 IDaaS 里，会提前预先集成了一些常用的 SaaS 应用，比如：钉钉、阿里云 RAM 、阿里邮箱等

2）B/S or C/S？

通常业务系统的网络结构模式主要有两种：B/S 结构（浏览器/服务器模式）C/S 结构（客户端/服务器模式）

B/S 结构：我们通常推荐使用：SAML、OAuth2.0、JWT、CAS 应用模板对接

3）IDaaS 标准协议

IDaaS目前主要提供四种 SSO 协议：SAML、JWT、OAuth2.0、CAS

如果您的业务系统支持上述其中的某个协议，仅需在应用模板中进行配置，然后业务系统进行少量的开发工作，即可实现 SSO 单点登录。

如果以上协议都不支持，在这里，我们将主推 JWT 模式，开发简单，容易实现。

4）代填模板

由于某些特殊情况下，业务系统不支持 IDaaS 提供的 4 种协议，同时又面临改造困难的问题。比如：系统建设时间长，联系供应商困难，或者供应商收费高昂无法进行适配改造等。

针对这种情况，可以使用表单代填来实现业务系统的单点登录，都是通过模拟用户输入账号密码方式实现。

通过下表，您可以进一步选择模板

对接协议（应用模板）	B/S	C/S	SP是否需要开发	开发/对接难度
JWT	✔	×	✔	简单
SAML	✔	×	✔	中等
OAuth	✔	×	✔	中等
CAS	✔	×	✔	简单
C/S（程序）	×	✔	✔	简单
C/S（浏览器）（指令代填）	✔	✔	×	简单
表单代填	✔	×	×	简单

四、权限怎么管理？

让我们完成了应用的 SSO 单点登录对接，完成了数据同步。此时，我们就要开始确定权限该如何分配。

这里，我们不需要明确到各个部门/岗位，到底该分配什么样权限，而是要明确，我们所需要的授权方式，授权模型。

在现有的 IDaaS 产品中，授权能力有有所区别。

比如，有的 IDaaS 仅支持应用的访问控制权限，不支持二级菜单、按钮权限；有的 IDaaS 不支持按部门授权等等。

这就需要我们整理所需的授权颗粒度，授权需求。来对各类 IDaaS 产品进行对比，挑选。

五、使用成本如何？

目前，市场上的 IDaaS 收费模式也可分为两大块

（1）云上 IDaaS：购买 license （使用人数）。按月、年计费购买

（2）本地部署 IDaaS：按产品标准费用，或是招投标项目形式进行采购。通常，本地化的 IDaaS 价格会高于云上 IDaaS。

（3）此外，如果需要第三方系统的对接 IDaaS 改造，也需要相应的费用。

六、可扩展性如何？

可扩展性主要指，IDaaS 所提供的接口是否够全面？未来有新的需求是否能灵活的进行功能的改造和扩展。这里也要提下问题1。

通常，云上的 IDaaS 服务可定制修改的能力较弱，依赖产品整体的迭代、更新、发布，周期较长，对急需相应的需求不适用。当然，我们也可以考虑云上的专属版本。

本地化部署的 IDaaS 可扩展性会好很多，大部分可以按客户的需求进行定制化改造。

七、性能及易用性如何？

性能方面：云上和本地最大的区别在于，云上是弹性计算，服务器会随着需求自动扩展。本地部署的需要提前就分配好服务器资源，做好负载均衡，高可用等

易用性方面：主要是指，管理员操作使用，上手难易度，以及进行对接，授权等工作的操作便宜性，这个就需要实际体验试用下来的更加直接。

八、安全策略如何设定？

安全策略主要指：

（1）密码规则策略

（2）数据备份策略

（3）是否需要二次认证

（4）是否需要多因素认证（指纹、手势、人脸、实人认证等）

在选择 IDaaS 产品的时候，也需要考虑这方面内容

1.4. CIAM 顾客身份权限管理

当我们讨论 IAM 的时候，大多数人想到的都是近二十年来不断发展的、针对企业内部员工、合作伙伴、渠道方、临时人员等提供统一身份和权限管理能力的内部产品。然而，随着「应用」的边界拓展到我们生活的方方面面，统一管理身份的需求也不断拓展边界，以企业为核心，由内而外，开始容纳企业外部的海量客户。

在这 20 年中，IAM（Identity and Access Management）的概念和场景都是统一的，但最近 3 至 5 年，由于终端用户消费市场的快速发展，孕育出了一套针对 C 端用户的专属用户和身份管理解决方案，称为 CIAM（Customer Identity and Access Management，顾客身份权限管理），以区别于较为传统的 EIAM（Employee Identity and Access Management，员工身份权限管理）。

中国的 C 端市场有着全世界绝无仅有的旺盛生命力，针对不同的、相互隔绝的细分市场，产生了层出不穷的模式创新，不断地深入地为用户提供方便、增加留存和黏性。中国用户的独特体验预期，也给外企入华带来了巨大的挑战。在白热化的发展中，企业的支配权越来越低，用户的选择权越来越大。用户体验不再是一个锦上添花的能力，而变成了差异化竞争的利器。

企业内的员工，可以通过培训和熟悉，来接受一款体验不那么优秀、但可以有效解决业务问题的 IAM 产品。但最终 C 端顾客永远是有选择的，预期未得到满足的结果，就是顾客流失。

2015 年开始，开始有咨询公司将 CIAM 作为一个独立的、拥有其独特要求的产品来看待。完全不同于以业务效率为核心的 EIAM 产品，CIAM 的目标是协助企业完成信息化转型，在所有对外服务中统一用户的身份，在以体验为核心的用户争夺战中，为终端用户提供完整的身份自助服务、为不同平台用户提供统一而流畅的使用、注册体验，为企业 IT 架构提供清晰的核心顾客身份管理系统，以此来提高用户的留存、黏性，提高顾客画像分析准确性和营销活动的有效性，进一步创造价值，在行业竞争中取得先机。

> *The unique requirements of customer identity, especially scale, performance, usability, and support for seamless multichannel interactions, have necessitated the development of CIAM as its own market segment with competitive offerings distinct from traditional solutions for employee IAM.*
>
> *顾客身份管理的独特要求，特别是关乎用户规模、性能、可用性以及跨终端体验的流畅支持等问题，使得 CIAM 作为拥有与传统 EIAM 迥然不同的特殊需求的独立市场。*

> —— Forrester

中国市场的终端用户需求非常独特，其使用习惯、沟通方式、对便捷性、服务可用性、安全性的要求都与国际上迥乎不同。在竞争中，能够用最本土化的方式，让用户稳定、安全使用所提供服务的企业，将会有巨大的竞争优势。

在企业内部，CIAM 的需求从最迫切的市场部门，一直延伸到安全部门、技术部门、法务部门、客户服务部门等，不同部门对 CIAM 的关注点也不全相同，在选取一个合适的 CIAM 方案的时候，需要综合内外的需求，选择一个有充足处理海量用户经验、了解 IAM 领域的最佳安全和功能实践、并能跟随行业发展而随时提供最贴合用户需求的服务提供商。选择错误的 CIAM 的后果，轻则影响到用户体验，严重的会显著损害品牌形象和企业营收。

阿里云 IDaaS 为企业客户提供基于私有环境和公共云的 CIAM 解决方案。欢迎来咨询我们获取 CIAM 产品白皮书。详情参考：

https://www.aliyun.com/product/idaas

。

1.5. 如何评估对身份系统的性能要求

随着企业对客业务的不断增长，中国的 ToC 业务也在不断变得复杂，身份系统过硬的性能表现成为了平台刚性要求。不同业务对于身份体系的要求也不尽相同，企业提出的性能指标也很容易不契合实际的需要，根据业务种类的不同，可能造成针对营收、口碑、品牌的损失。

下面提供一些参考，供企业评估性能要求时使用。

首先，需要建立对性能要求的最基本基准，企业需要从业务角度出发，判断出对身份系统的要求的各项平均值和上限。有了基准，才拥有了可衡量是否达标的标准。

建立性能基准是一项很容易失之毫厘谬以千里的工作，需要精细地控制变量。

第一 判断最常用身份功能

作为身份体系，往往最常用的功能有两部分：

1. 注册、认证体系
2. 获取身份信息

从这两部分中，还可以细化出很多不同的侧重点。举例而言，某个业务应用的用户有 80%+ 的认证频次是使用支付宝登录的。测试认证时则可以额外针对该点进行有效且充分的压力测试。

判断最常用的功能可以有助于梳理清楚业务系统对身份体系的最严重依赖点，为专项性能测试提供了逻辑，并为系统优化也指明了方向。

有一些频繁调用的请求是可以通过巧妙的接口设计来规避掉的，比如秒杀活动时对身份体系接口的依赖，设计逻辑时需要额外小心，尽量规避掉不必要的高频调用。

第二 判断最复杂的关键逻辑

这里的复杂指的不是用户使用复杂，而是可能简单的操作背后，是否有着复杂的逻辑判断和频繁的数据库交互。阿里云 IDaaS 会使用尽量多的技术手段规避掉频繁的复杂业务查询或过量的数据库访问，但仍然不可避免会有请求有更高的复杂度。

判断出最复杂的关键逻辑，往往可以准确定位到身份系统的瓶颈。结合最常用功能，如果二者有交叉，代表功能既常用又复杂，那么交叉范围内的能力就需要额外的性能保障。

第三 评估业务场景的频次和周期

根据企业所拥有的对客场景和集成方式的不同，用户使用 IDaaS 身份功能的频率也会有非常大的变化。举例而言，证券企业对客服务可能高峰集中在早上 9点30 和下午 1 点，每周一至周五往往非常平均；零售企业的峰值可能在每月的促销秒杀活动以及一年一次的双十一凌晨秒杀期；旅游文旅项目的用户访问频次往往较低，每月一次、每年一次甚至数年一次；在线教育的访问往往一天需要数次打卡，打卡时间相对集中。

企业需要评估一年内、一个季度内、一个月内、一周内、一天内的访问规律，并以此来判断对 IDaaS 对应功能的频次要求。

第四 判断峰值和均值

在建立了如上三点认知后，就可以着手判断请求的峰值和均值。举例而言，假设一次抢票活动，每次进入抢票页都需要来 IDaaS 获取一次身份信息。活动为时约 10 分钟，预计共有 10W 人参与，在前 10 秒会有 10% 的用户参与抢票。那么，计算可知前 10 秒内平均每秒会有 1K 用户访问系统。我们可以以此为假设峰值，来对系统的性能需要进行基本判断。

当没有活动时，系统往往会出于一个相对平缓的、周期性的波动范围中，可以以此作为系统均值。

总结

性能的评估、测试和上线稳定性的确认，都只能在一定程度上、在波动范围不异常剧烈的情况下，尽可能地确保环境受到冲击下的稳定性和可靠性。针对企业明确预知的大流量情况，还需要额外的评估、定方案、压测总结和优化的过程，才能确保重大事件下性能的稳定和可靠。阿里云 IDaaS 依托于阿里云平台的优异性能表现，可以很好地处理应对多样化的性能需要，详情请咨询产品团队。

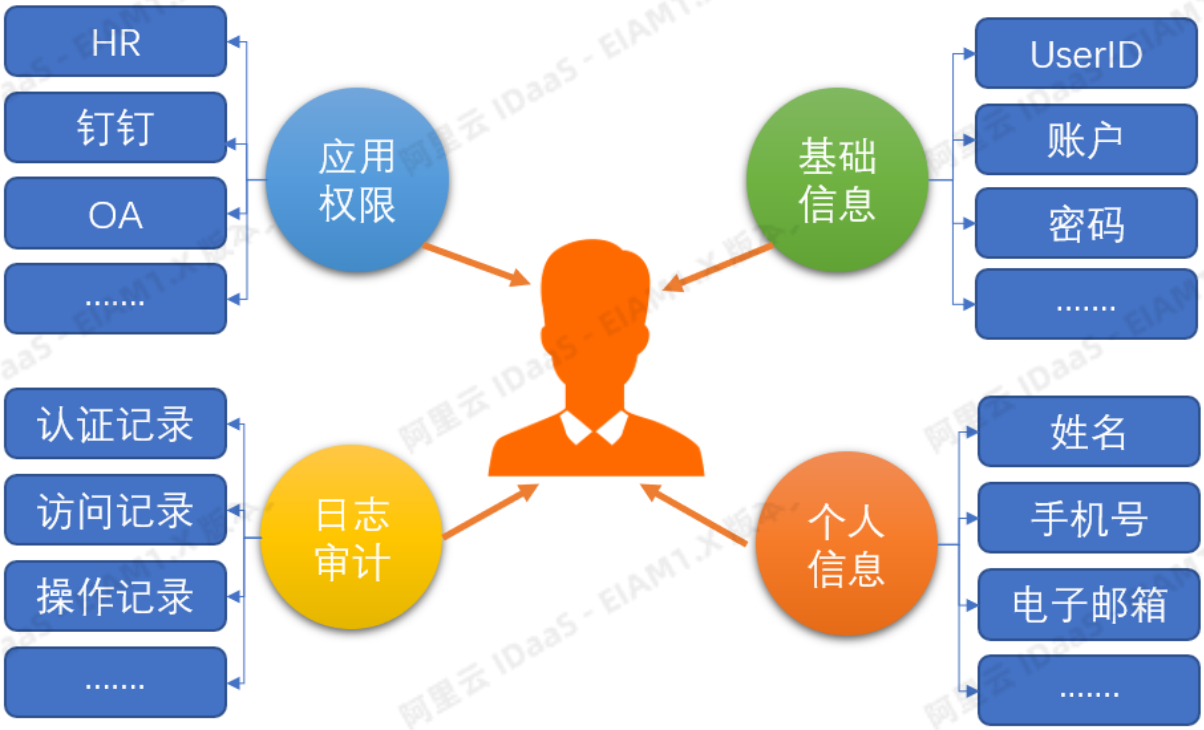
2.账户

2.1. 账户全生命周期管理

什么是账户全生命周期管理

简单来说，就是从账户的产生到消亡的整个过程管理，从业务角度来说就是员工入职起新建账户到该员工离职时的账户归档的全过程管理。

账户有哪些关键属性



通常，一个完整的账户都会有以下四种属性：

1）基础信息：是一个账户最基本不可或缺的信息，主要包括：ID（唯一标志）、账户名称、密码等。

2）个人信息：用来完善每个账户的属性，主要包括：用户姓名、手机号、邮箱账号、所属部门、工号等。

3）应用权限：用来记录每个账号的权限资源，确保账号访问正确的资源。

4）日志审计：用来记录每个账号的日志记录，主要包括：认证记录、资源访问记录、登入/登出记录、操作记录等。

账户全生命周期管理主要内容

一个账号的生命周期主要包括一下几个方面：

1）员工入职

2）创建员工账户

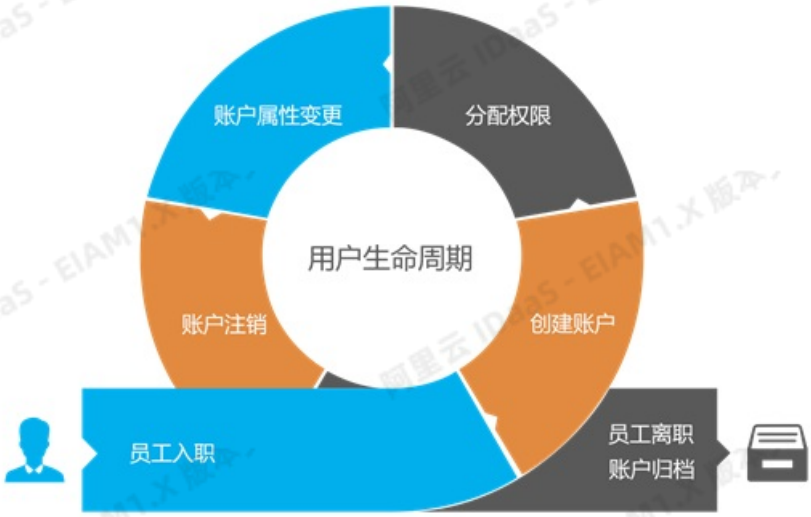
3）账户权限分配

4）账户属性变更（员工转岗）

5）账户注销（账号停用或启用）

6）员工离职

7）账户归档，权限回收



企业通常面临账户管理的难题

随着公司的发展，企业内部应用和人员数量会不断增加。人员的入职，离职人员组织架构频繁调整（转岗），同时企业内部人员角色（正式员工/临时工、渠道/合作伙伴等）愈加复杂，每个应用的管理员手动开关账号的工作量飙升。同时，因手动管理人员账号，经常出现人员已离职但账户未关闭的高危情况。

就像图中的场景

1）入职员工账户未第一时间开通

2）公司组织机构调整，但依旧有业务系统还未调整完毕

3）离职员工账户权限未第一时间将权限收回

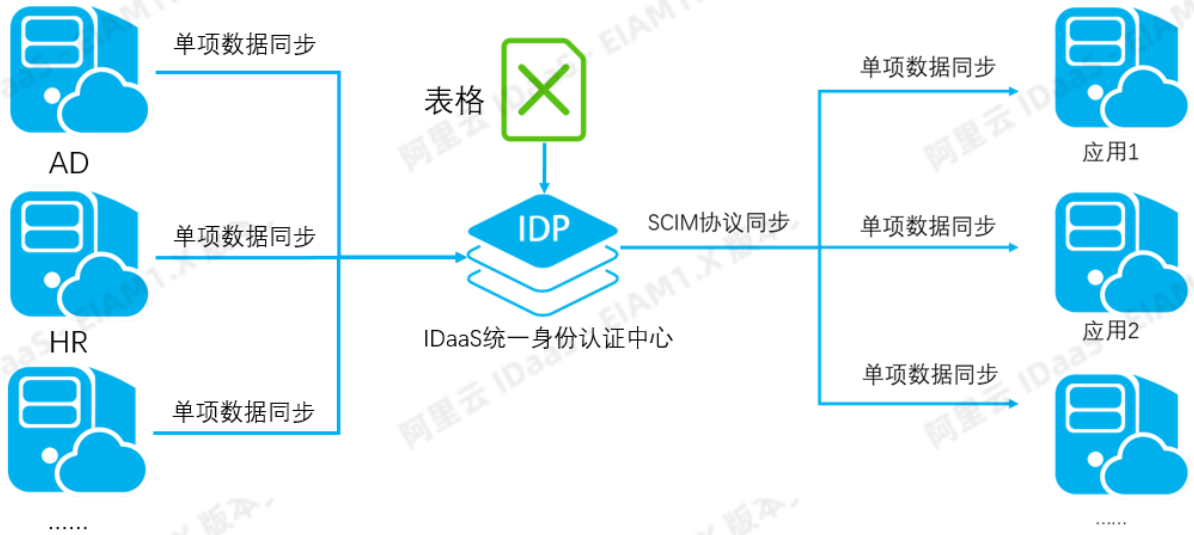
不仅增加了各个业务系统管理员的工作量，重复工作；同时还存在账号管理不到位，同步信息滞后，影响正常业务；若员工离职账号权限未及时收回，甚至还会导致公司数据泄露等重大风险。

所以，账户全生命周期管理的需求，随着企业的发展日益凸显。

近10年来，各大企业也纷纷寻找合适的系统供应商来解决账户全生命周期的管理需求。譬如：企业OA系统、HR系统、AD等来实现账户管理。但是又面临着身份孤岛的新问题，各系统身份信息独立，信息同步困难，建设难度大，权限不集中等。

IDaaS账号全生命周期管理

IDaaS为解决企业账户管理的难的痛点，深入了解账户特征，基于业务场景。研究梳理出一套完整的账户全生命周期管理办法。IDaaS管理账号的核心思想：统一集中身份，一处修改，处处生效



2.2. Active Directory

什么是 Active Directory

活动目录（Active Directory），简称 AD，是微软Windows Server中，负责架构中大型网路环境的集中式目录管理服务（Directory Services），Windows 2000 Server开始内建于Windows Server产品中，它处理了在组织中的网路物件，物件可以是**计算机，用户，群组，组织单元（OU）**等等，只要是在Active Directory结构定义档（schema）中定义的物件，就可以储存在Active Directory资料档中，并利用Active Directory Service Interface来存取。

AD 有什么功能？

1.用户服务

管理用户的域账号、用户信息、企业通信录（与电子邮箱系统集成）、用户组管理、用户身份认证、用户授权管理、按需实施组管理策略等。这里不单单指某些线上的应用更多的是指真实的计算机，服务器等。

2.计算机管理

管理服务器及客户端计算机账户、所有服务器及客户端计算机加入域管理并按需实施组策略。

3.资源管理

管理打印机、文件共享服务、网络资源等实施组策略。

4.应用系统的支持

对于电子邮件（Exchange）、在线及时通讯（Lync）、企业信息管理（SharePoint）、微软CRM,ERP等业务系统提供数据认证（身份认证、数据集成、组织规则等）。这里不单是微软产品的集成，其它的业务系统根据公用接口的方式一样可以嵌入进来。

5.客户端桌面管理

系统管理员可以集中的配置各种桌面配置策略，如：用户适用域中资源权限限制、界面功能的限制、应用程序执行特征的限制、网络连接限制、安全配置限制等。

Active Directory域架构常用对象

1.域（Domain）.

简单理解为：A公司总部

域（Domain）是AD的根，是AD的管理单位。域中包含着大量的域对象，如：组织单位（Organizational Unit），组（Group），用户（User），计算机（Computer），联系人（Contact），打印机，安全策略等。

2.组织单位（Organization Unit）.

简单理解为：A公司的分公司

组织单位简称为OU是一个容器对象，可以把域中的对象组织成逻辑组，帮助网络管理员简化管理组。组织单位可以包含下列类型的对象：用户，计算机，工作组，打印机，安全策略，其他组织单位等。可以在组织单位基础上部署组策略，统一管理组织单位中的域对象。

3.群组（Group）.

简单理解为：某分公司里的某事业部

群组是一批具有相同管理任务的用户账户，计算机账户或者其他域对象的一个集合。例如公司的开发组，产品组，运维组等等。

群组类型分为两类：

安全组：用来设置有安全权限相关任务的用户或者计算机账户的集合。比如：Tiger组都可以登录并访问某ftp地址，并拿到某个文件。

通信组：用于用户之间通信的组，适用通信组可以向一组用户发送电子邮件。比如：我要向团队内10为成员都发送同一封邮件这里就要抄送9次，而使用组的话我直接可以发送给@Tiger，所有Tiger组内的成员都会收到邮件。

4.用户（User）.

简单理解为：某个工作人员

AD中域用户是最小的管理单位，域用户最容易管理又最难管理，如果赋予域用户的权限过大，将带来安全隐患，如果权限过小域用户无法正常工作。

域用户的类型，域中常见用户类型分为：

普通域用户：创建的域用户默认就添加到"Domain Users"中。

域管理员：普通域用户添加进"Domain Admins"中，其权限升为域管理员。

企业管理员：普通域管理员添加进"Enterprise Admins"后，其权限提升为企业管理员，企业管理员具有最高权限。

3.应用

3.1. SAML

SAML简介

SAML全称是安全断言标记语言（Security Assertion Markup Language）是一个基于XML的开源标准数据格式。用于在不同的安全域之间交换认证和数据授权。在SAML标准定义了身份提供者（IDP）和服务提供者（SP），这两者构成了前面所说的不同的安全域。 SAML是OASIS组织安全服务技术委员会（Security Services Technical Committee）的产品。

SAML解决的最重要的需求是Web端应用的单点登录（SSO）。

SAML协议工作流程

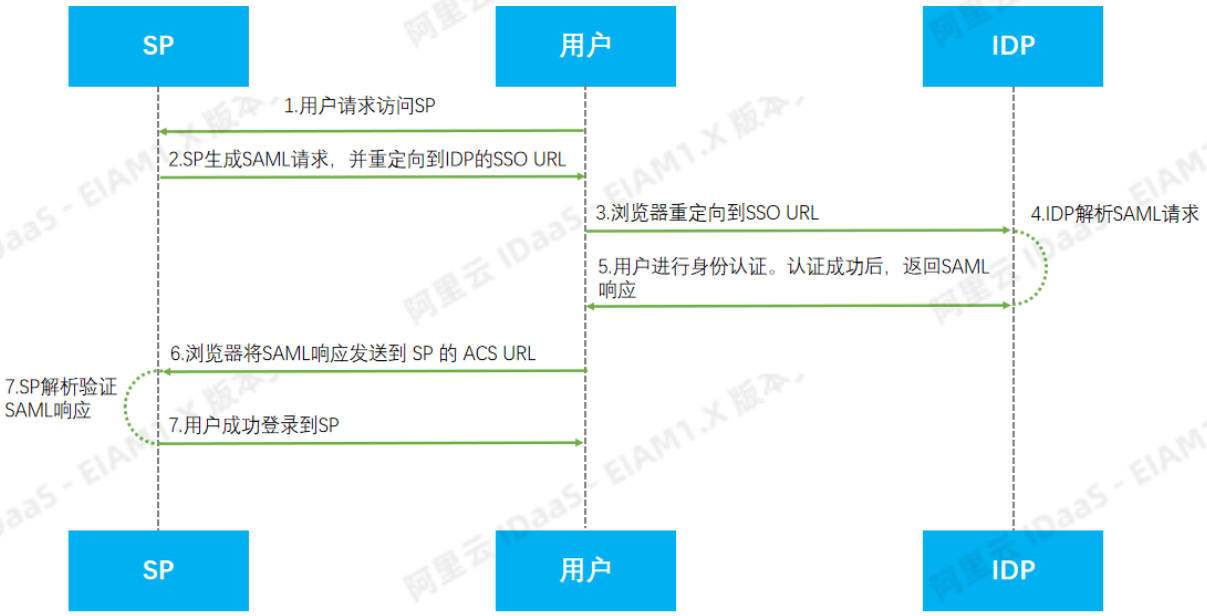
SAML 协议主要有三个角色：

SP（Service Provider）：向用户提供服务的web 端应用。

IDP（Identity Provide）：向SP提供用户身份信息

用户：通过登录IDP获取身份断言，并向SP返回身份断言来使用SP提供的服务。

下图是 SAML 协议的工作流程



1. 用户请求访问 Web 应用系统。
2. Web 应用系统生成一个 SAML 身份验证请求。
3. Web 应用系统将重定向网址发送到用户的浏览器。重定向网址包含应向SSO 服务提交的编码 SAML 身份验证请求。
4. IDP 对 SAML 请求进行解码。
5. IDP对用户进行身份验证。认证成功后，IDP生成一个 SAML 响应，其中包含经过验证的用户的用户名。然后将SAML 响应编码并返回到用户的浏览器。

6. 浏览器将 SAML 响应转发到 Web 应用系统 ACS URL。
7. Web 应用系统使用 IDP 的公钥验证 SAML 响应。如果成功验证该响应，ACS 则会将用户重定向到目标网址。
8. 用户将重定向到目标网址并登录到 Web 应用系统。

从上述流程我们可以看出：在 SAML 协议中，IDP 和 SP 不需要直接进行通讯，只要用户浏览器可以访问到 IDP 和 SP 即可。也就是说 SAML 协议在混合云环境下也可以正常进行使用，只要用户浏览器可以访问到公有云的 IDP 和内网的应用就可以使用 SAML 协议集成应用的单点登录。

3.2. OAuth2

OAuth（Open Authorization，开放授权）是一个开放标准的授权协议，允许用户授权第三方应用访问他们存储在资源服务上受保护的信息，而不需要将用户名和密码提供给第三方应用，解耦了认证和授权。OAuth作为一种国际标准，目前传播广泛并被持续采用。OAuth2.0是OAuth协议的延续版本，更加安全，更易于实现，但不向后兼容OAuth1.0，即完全废止了OAuth1.0。

场景举例

让朋友去自己家取东西，如果我直接给他门禁系统的密码，朋友就拥有了与我同样的权限，这样不太合适。如果我想取消朋友进入我家的权限，需要更改密码才可以，操作麻烦。

能不能有种方法，朋友可以进入我家，但是又不知道我家门禁的密码。朋友如果多次进入我家，每次都需要经过我的授权，我不授权的就进不去，我授权后才能进去。

我们设计如下方案：

第一步，在门禁系统的密码输入器下面，增加一个按钮，比如叫做"获取授权"。朋友需要首先按这个按钮，去申请授权。

第二步，朋友按下按钮以后，我的手机就会弹出提示：有人正在请求授权，并显示朋友的姓名。我确认情况属实，点击同意授权。

第三步，门禁系统得到我的确认后，向朋友发送一个进入房屋的令牌（access token），令牌是类似密码的一串数字，只在短期内有效。

第四步，朋友向门禁系统输入令牌，进入房屋。

OAuth2介绍

如果互联网应用需要实现上面的授权场景，肯定有许多的实现方式。而OAuth2就是实现上述目标的一种规范，也就是具体实现的指导方案。OAuth（Open Authorization，开放授权）是为用户资源的授权定义了一个安全、开放及简单的标准，第三方无需知道用户的账号及密码，就可获取到用户的授权信息。

- OAuth2是用于REST/APIs的代理授权框架（delegated authorization framework）
- OAuth2是基于令牌Token的授权,在无需暴露用户密码的情况下,让应用获取对用户数据有限访问权限
- OAuth2解耦认证和授权

名词解释

- **Third-party application**：第三方应用，又称"客户端"（client），如上面场景中的门禁系统。
- **Resource Owner**：资源拥有者，也就是用户。
- **Http Service**：服务提供商，也就是持有Resource Server的存在方。可以理解为类似微信，钉钉这样具备用户信息的服务者。

- **Authorization server**：认证服务器，即服务提供商专门用来处理认证的服务器。
- **Resource server**：资源服务器，即服务提供商存放用户生成的资源的服务器。与认证服务器是不同的逻辑节点，但是在物理上，双方是可以在一起的
- **User Agent**：用户代理，一般就是指的浏览器。
- **客户端凭证**：Client Id和密码用于认证用户。
- **访问令牌**：授权服务提供者在接收到用户请求后,颁发的访问令牌。
- **刷新令牌（Refresh Token）**：用于获取一个新的令牌。由于令牌的有效期比较短，一旦失效，用户需要再获取令牌的流程是比较繁琐的。为了提升用户体验，可以使用reflesh_token来获取新的令牌。

互联网场景举例

OAuth在"客户端"与"服务提供商"之间，设置了一个授权层（authorization layer）。"客户端"不能直接登录"服务提供商"，只能登录授权层，以此将用户与客户端区分开来。"客户端"登录授权层所用的令牌（token），与用户的密码不同。用户可以在登录的时候，指定授权层令牌的权限范围和有效期。

"客户端"登录授权层以后，"服务提供商"根据令牌的权限范围和有效期，向"客户端"开放用户储存的资料。

【小明】授权【在线打印app】到【QQ空间】访问【小明的指定相册】，以完成打印工作。

关键概念	解释	举例
资源拥有者	受保护资源的拥有者，有权决定【谁】可以访问受保护资源	小明
客户端	需要获得资源拥有者授权，访问受保护资源的第三方应用	在线打印APP
认证服务器	资源拥有者可通过认证服务器下放权限令牌	QQ空间
资源服务器	存储受保护资源的服务器，根据权限令牌返回第三方应用请求的数据	小明的指定相册，有时候认证服务器和资源服务器是一台
客户端凭证	授权服务器给客户端颁发的身份凭证（Client ID、Client Secret）	QQ空间得认为在线打印app是合法的
访问令牌	授权服务器验证了用户身份后，颁发的代表权限范围的访问令牌	小明授权在线打印app只能看指定相册，不能看其他相册和日志等

OAuth应用场景

开放系统间授权：社交联合登录、开放API平台

比如：APP或者网页接入一些第三方应用时，时常会需要用户登录另一个合作平台，比如钉钉，QQ，微博的授权登录,这些平台都提供了基于OAuth2的机制，你可以用这些平台的账号登录而无需在其他网站上进行注册，并授权此网站获取其账号信息，包括用户名、头像等。

现代微服务安全：单页浏览器App（HTML5/JS/无状态）、无线原生App

比如：app登录请求后台接口，为了安全认证，所有请求都带token信息，如果登录验证、请求后台数据。

企业内部应用认证授权（IAM/SSO），前后端分离单页面应用（spa）

比如：有个应用是别人开发的，你需要将系统进行整合或者数据对接，此时需要单点登录；前后端分离框架，前端请求后台数据，需要进行oauth2安全认证。

令牌和密码

令牌（token）与密码（password）的作用是一样的，都可以进入系统，但是有几点差异。

（1）令牌是短期的，到期会自动失效，用户自己无法修改。密码一般长期有效，用户不修改，就不会发生变化。

（2）令牌可以被资源持有者撤销，会立即失效。以上例而言，我可以随时取消朋友的令牌，密码一般不允许被他人撤销。

（3）令牌有权限范围（scope），比如使用令牌只允许进入客厅，不允许进入卧室。对于网络服务来说，只读令牌比读写令牌更安全，而密码一般是完整权限。

上面这些设计，保证了令牌既可以让第三方应用获得权限，同时又随时可控，不会危及系统安全。这就是 OAuth 2.0 的优点。

但是只要知道了令牌，就能进入系统。系统一般不会再次确认身份，所以**令牌必须保密，泄漏令牌与泄漏密码的后果是一样的。** 这也是为什么令牌的有效期，一般都设置得很短的原因。

OAuth2与session、cookie机制比对

1、与session机制类似，OAuth2只是变成了token，但是session有其局限性，特别是API对接

2、还有一些终端默认是不会带cookie的，比如Android

3、OAuth2可以返回refresh_toke，让客户端在一定时间内刷新token，特别适合app一定时间内无需重复登录。

OAuth2缺点

- 协议框架太宽泛,造成各种实现的兼容性和相互操作性差
- 和OAuth1.0不兼容
- OAuth2.0不是一个认证协议（是授权协议）,OAuth2.0本身并不能告诉你任何用户信息。

OAuth四种授权模式

客户端必须得到用户的授权（authorization grant），才能获得令牌（access token）。OAuth 2.0 规定了四种获得令牌的流程，可以选择实际情况选择最适合的一种，向第三方应用颁发令牌。

- 授权码模式（authorization-code）：正宗方式，支持refresh_token
- 简化模式（implicit）：为Web浏览器应用设计，不支持refresh_token
- 密码模式（password）：为遗留系统设计，支持refresh_token
- 客户端模式（client credentials）为后台api服务设计，不支持refresh_token

不管哪一种授权方式，第三方应用申请令牌之前，都必须先到系统备案，说明自己的身份，然后会拿到两个身份识别码：客户端 ID（client ID 用来标识第三方应用）和客户端密钥（client secret 用来进行安全加密）。这是为了防止令牌被滥用，没有备案过的第三方应用，是不会拿到令牌的。

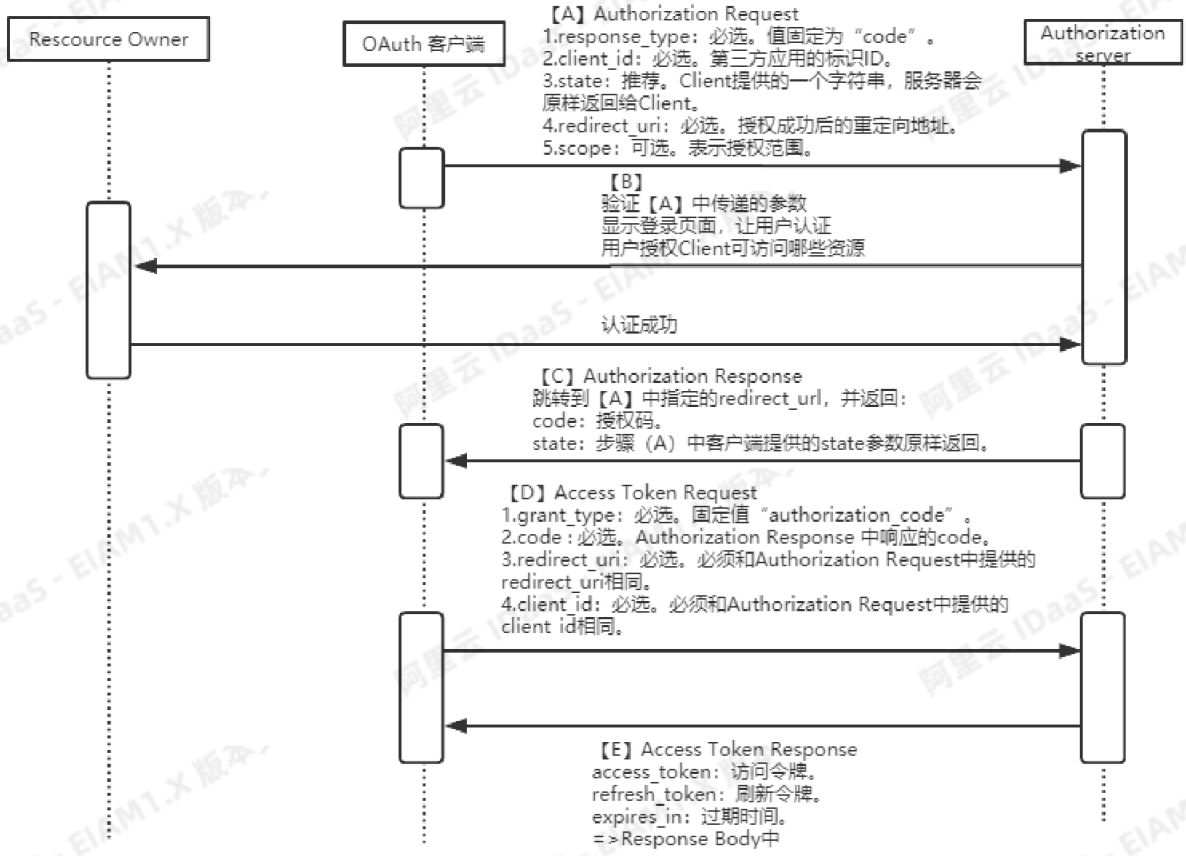
授权码模式

这是OAuth2最常用的一种授权许可类型，要求Client具有可公开访问的Server服务器来接受Authorization Code

- 这种模式算是正宗的oauth2的授权模式
- 设计了auth code，通过这个code再获取token
- 支持refresh token

优点	缺点	备注

Access token通过服务器之间进行交换，比较安全	请求次数比较多	推荐该模式



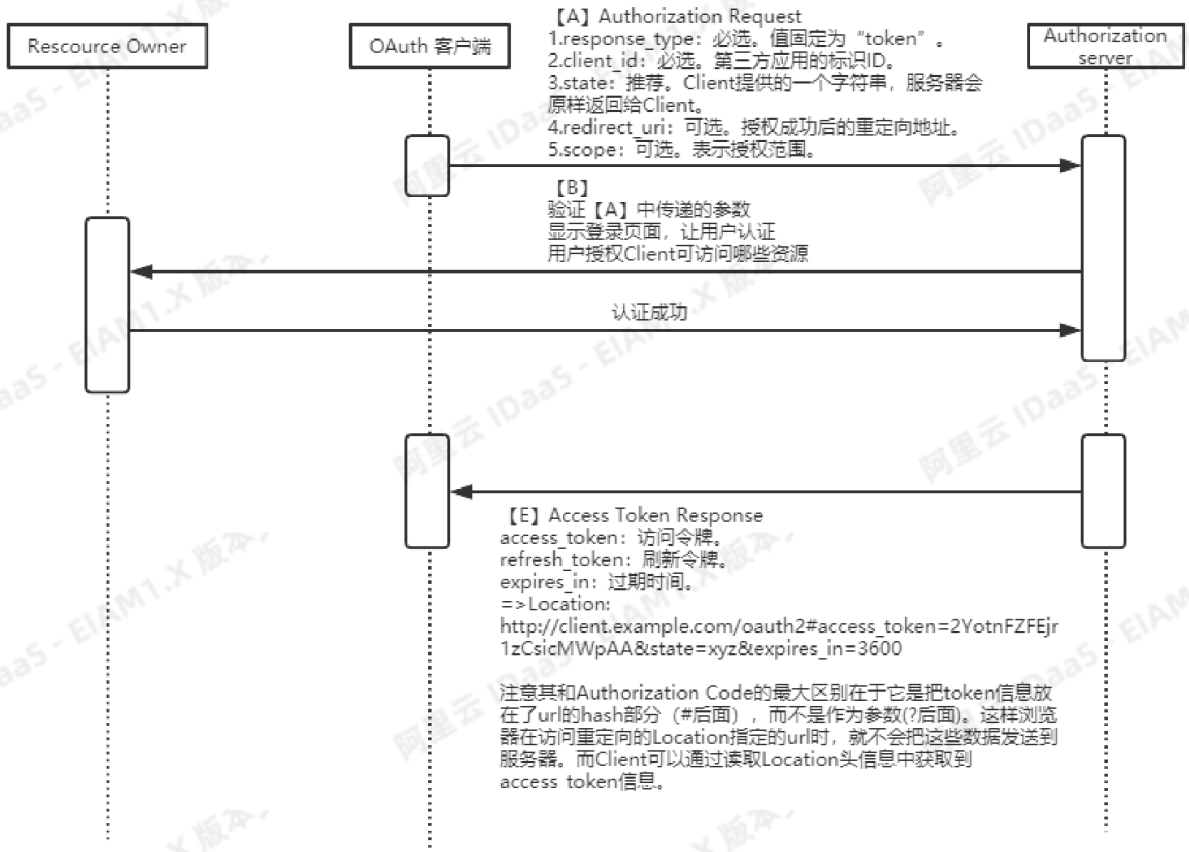
1. 用户登录应用系统，请求跳转到认证服务器，并302返回登录认证页面；
2. 用户输入账户+密码进行认证，认证服务器认证通过返回code给应用系统；
3. 应用系统携带code向认证服务器换取访问令牌，认证服务器验证Client ID，code等信息，给应用系统发送访问令牌；
4. 应用系统携带访问令牌查询用户登录信息，认证服务器返回用户信息，如用户名；
5. 应用系统验证用户名正确，创建会话，并跳转到redirect url。

简化模式

省略掉了颁发授权码（Authorization Code）给客户端的过程，直接返回访问令牌和可选的刷新令牌。其适用于没有Server服务器来接受处理Authorization Code的第三方应用

优点	缺点	备注

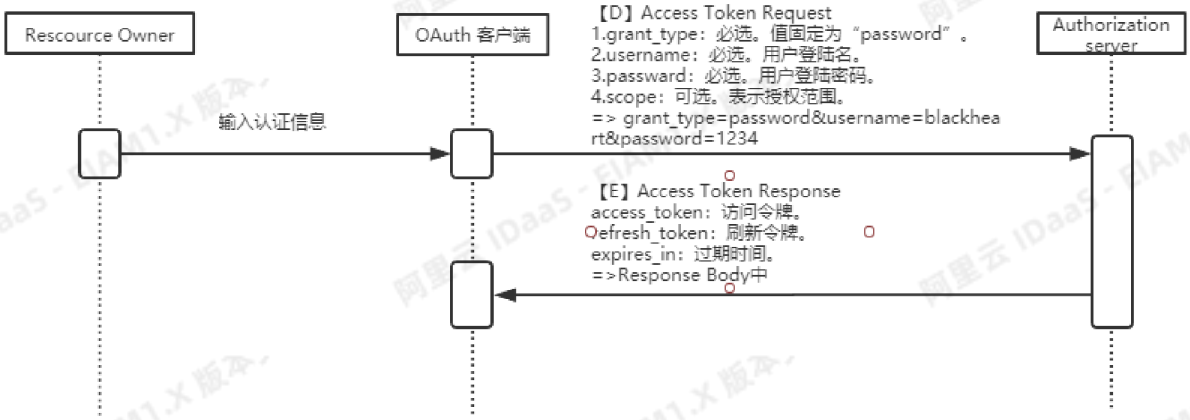
请求次数比较少，简单	1.没有获取code的过程， Access token直接从授权服务器返回给client客户端，令牌容易因为被拦截窃听而泄露 2.无法存储refresh token，不支持刷新令牌： 要么access token有效性给很长 ，要么access token过期后，让用户重新认证	适用于公开的浏览器单页应用



密码模式

这种模式再一步简化，和Authorzation Code类型下重要的区分就是省略了Authorization Request和Authorization Response。而是Client直接使用Resource owner提供的username和password来直接请求access_token（直接发起Access Token Request然后返回Access Token Response信息）。这种模式一般适用于Resource server高度信任第三方Client的情况下

优点	缺点	备注
请求次数比较少，简单	1.Client会获得用户的登录信息，除非是非常信任的应用，否则可能导致登录信息泄露。 2.没有多因子认证这样的机制	1. 可以用来做遗留项目升级为oauth2的适配方案 2. Client是自家应用的场景

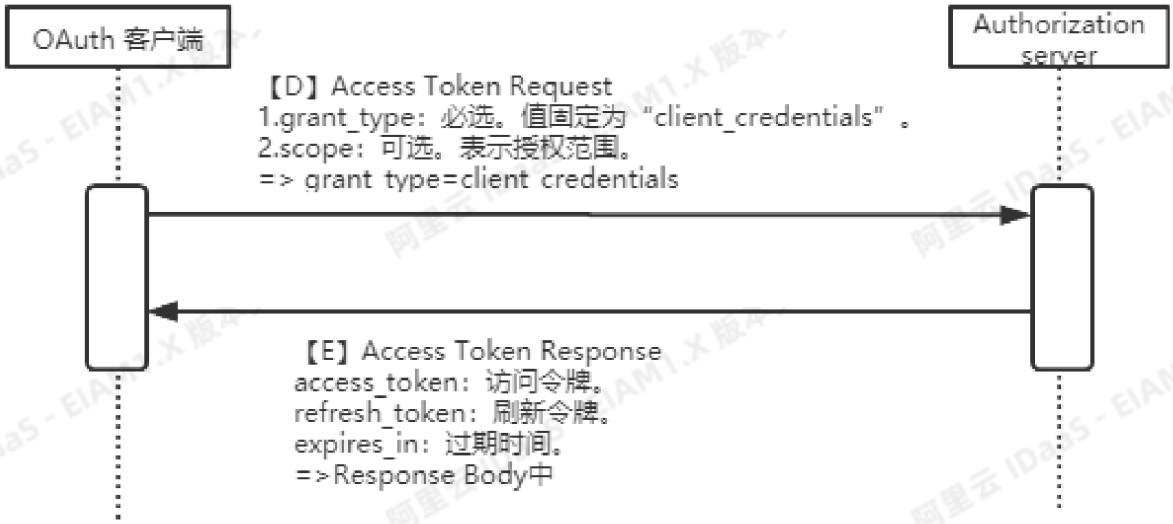


客户端模式

Client直接以自己的名义而不是Resource owner的名义去要求访问Resource server的一些受保护资源

- 适用于服务器间通信场景，直接根据client的ID和密钥即可获取token，无需用户参与
- 这种模式比较合适消费api的后端服务，比如拉取一组用户信息等
- 不支持refresh token

Refresh token的初衷主要是为了用户体验不想用户重复输入账号密码来换取新token，因而设计了refresh token用于换取新token，客户端模式由于没有用户参与，而且也不需要用户账号密码，仅仅根据自己的id和密钥就可以换取新token，因而没必要refresh token.

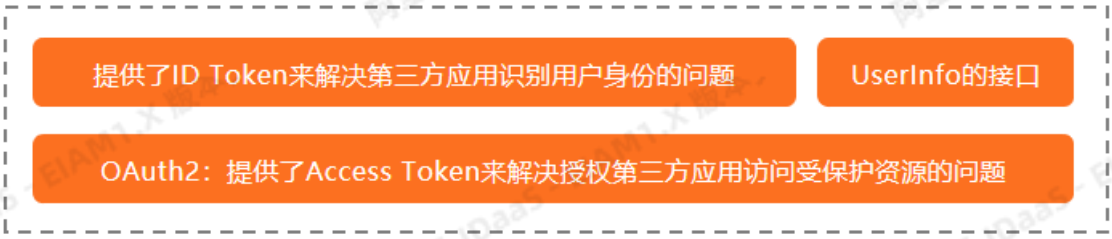


刷新令牌

以便在访问令牌过期失效的时候可以由客户端自动获取新的访问令牌，而不是让用户再次登录授权。

3.3. OIDC

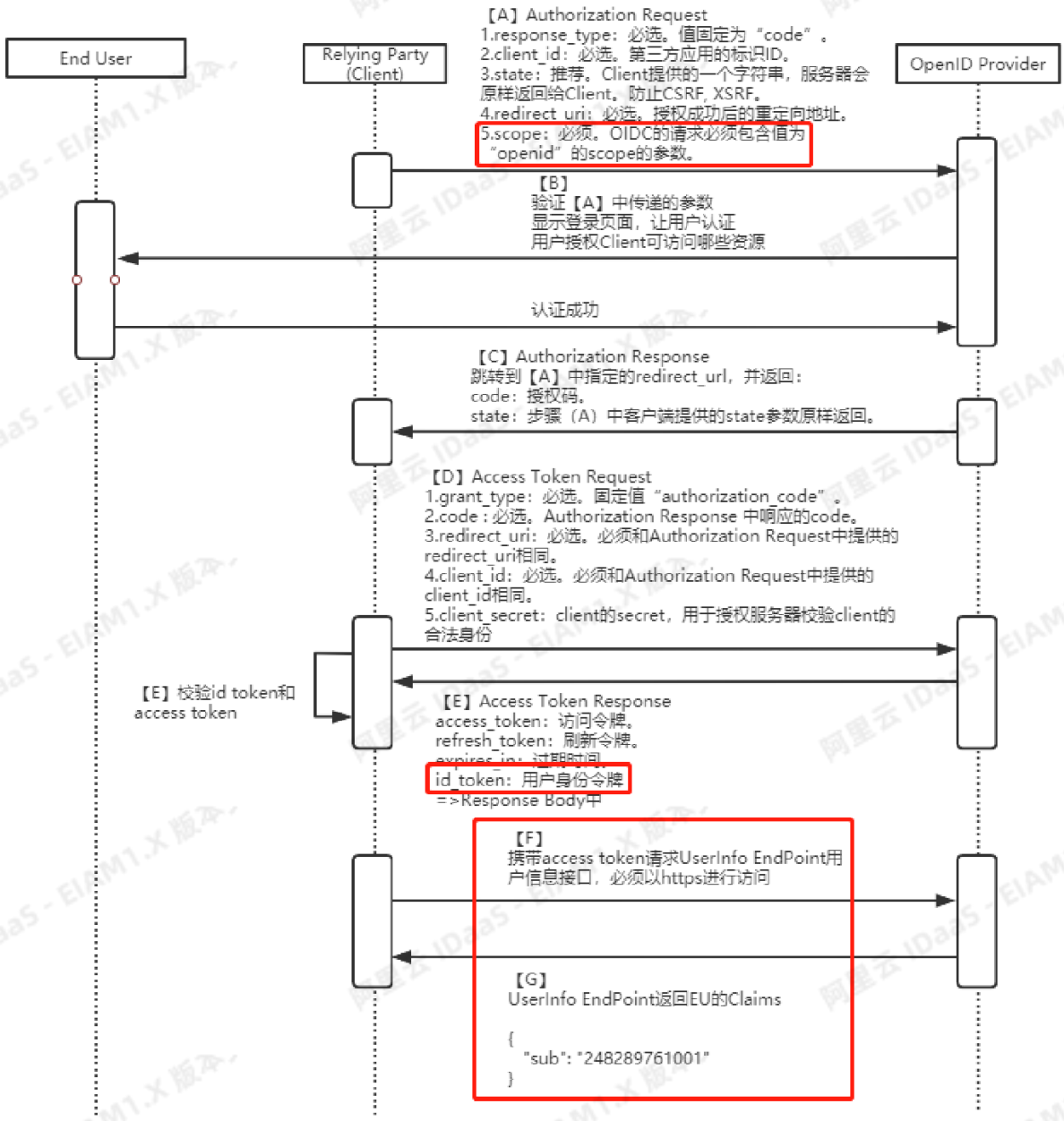
OIDC是OpenID Connect的简称，OIDC=(Identity, Authentication) + OAuth 2.0。它在OAuth2上构建了一个身份层，是一个基于OAuth2协议的身份认证标准协议。OIDC是一个协议族，提供很多的标准协议，包括Core核心协议和一些扩展协议。



OIDC认证流程

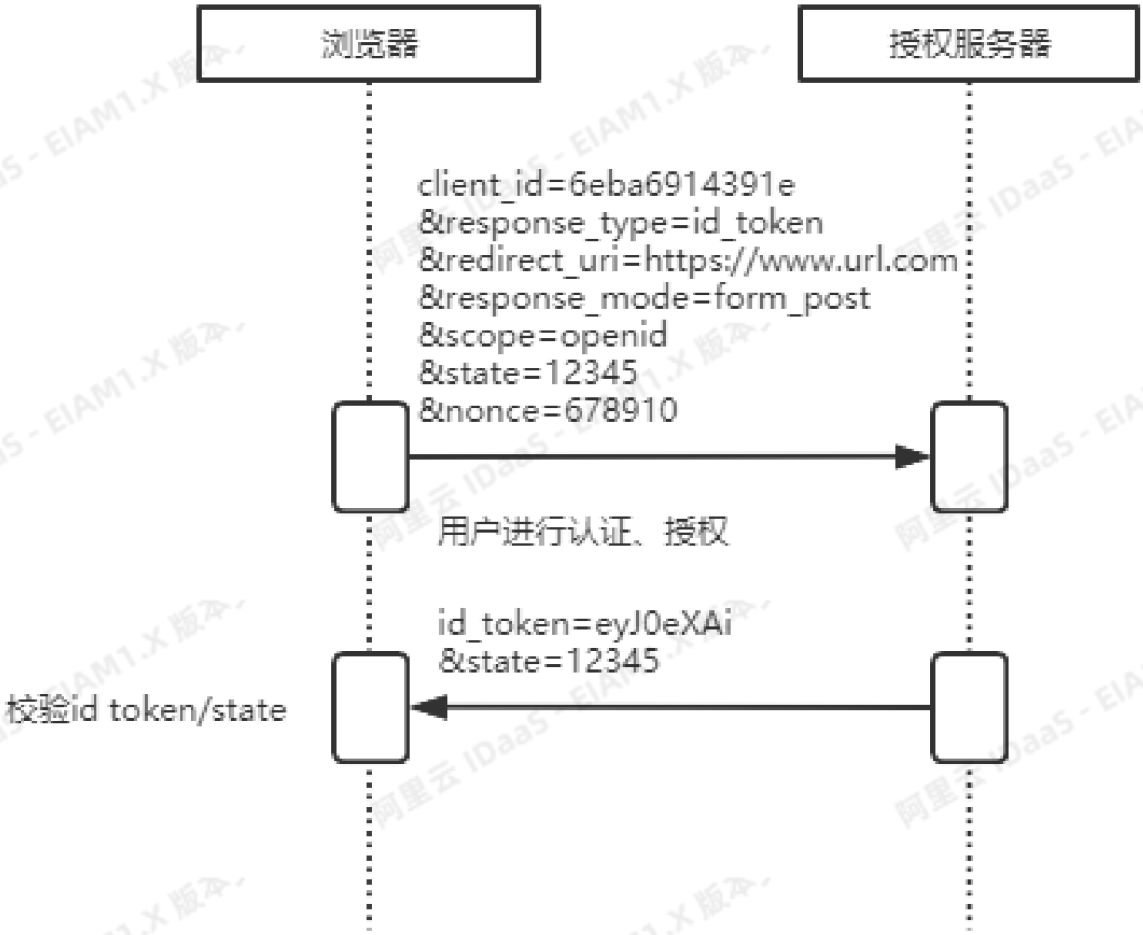
Authorization Code Flow

使用OAuth2的授权码流程来换取Id Token和Access Token

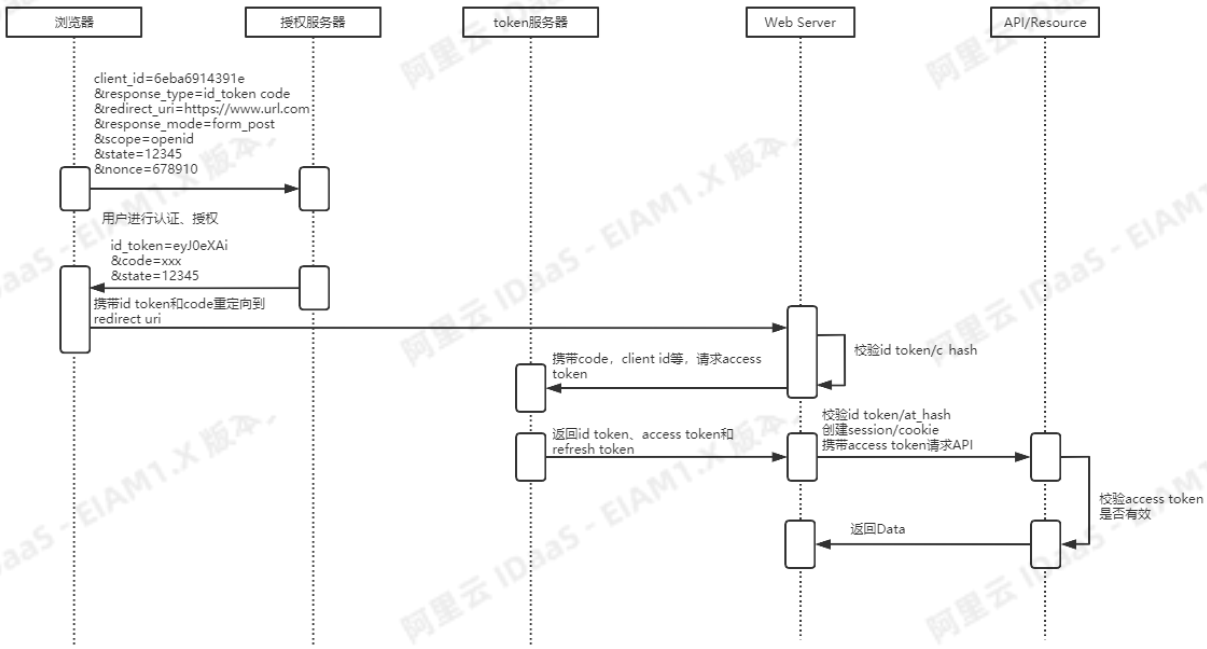


Implicit Flow

使用OAuth2的简化模式来换取Id Token



Hybird Flow

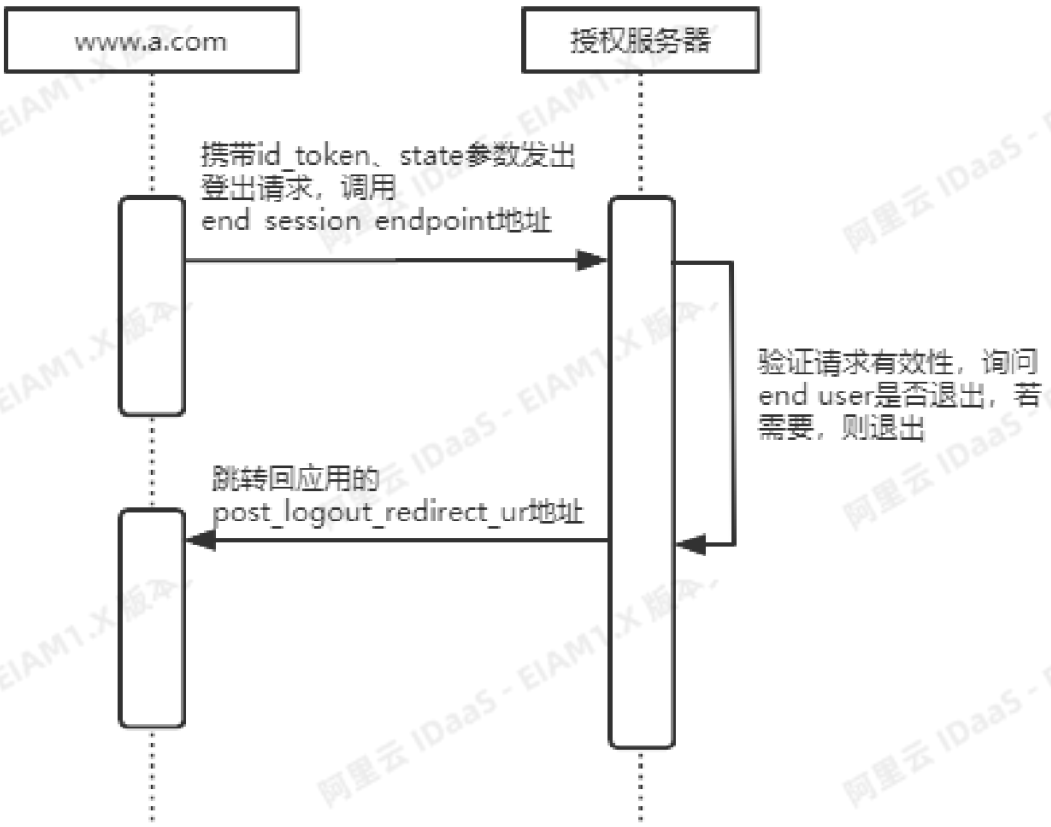


其它

OIDC登出机制

准备：

1.通过Discovery标准协议获得授权服务器的end_session_endpoint地址
2.应用在注册的时候提供登出成功后的跳转地址post_logout_redirect_uri给授权服务器



OIDC 和 OAuth2的区别

会返回用户的身份令牌id_token

有查询用户详细身份的userinfo接口

- Core
- Discovery：查询OIDC服务的接口地址，例如授权地址、获取Token的地址
- Dynamic Registration：客户端获取身份凭证
- Session Management

... ...

3.4. CAS(需要修改)

CAS是什么？

CAS是英文单词CompareAndSwap的缩写，中文意思是：比较并替换。CAS需要有3个操作数：内存地址V，旧的预期值A，即将要更新的目标值B。

CAS算法理解

对CAS的理解，CAS是一种无锁算法，CAS有3个操作数，内存值V，旧的预期值A，要修改的新值B。当且仅当预期值A和内存值V相同时，将内存值V修改为B，否则什么都不做。

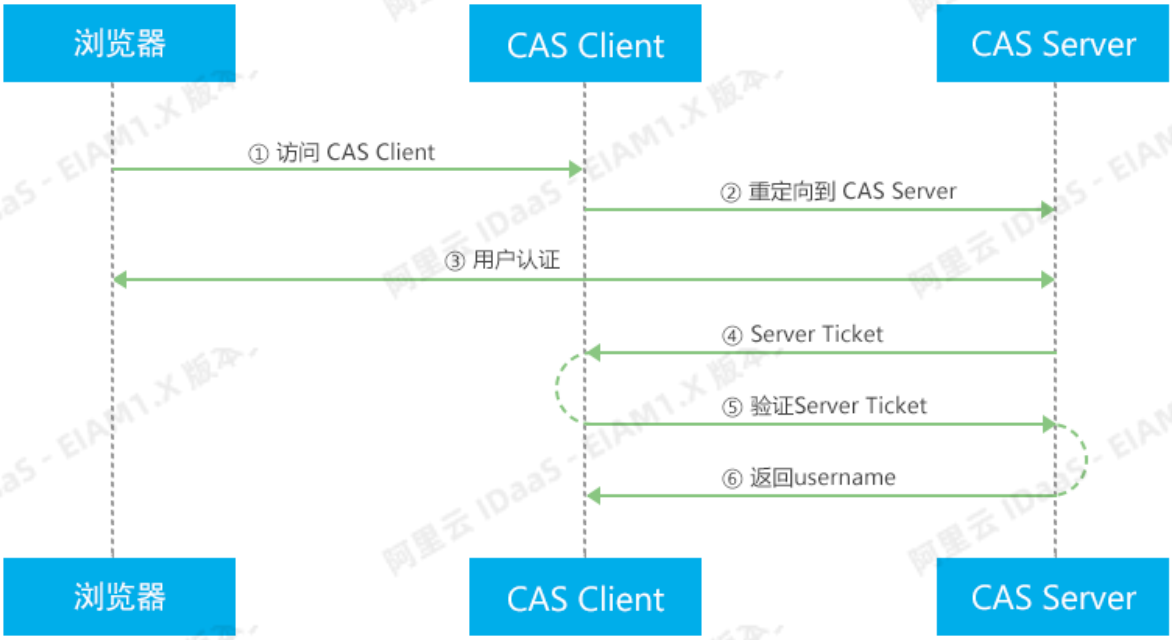
cas实现流程

使用CAS 标准时，首先是由 CAS Client 发起， CAS Client 会重定向到 CAS Server进行登录，由 CAS Server 进行账户校验且多个 CAS Client 之间可以共享登录的 session ，Server 和 Client 是一对多的关系；

原理和协议

从结构上看，CAS 包含两个部分： CAS Server 和 CAS Client 。 CAS Server 需要独立部署，主要负责对用户的认证工作； CAS Client 负责处理对客户端受保护资源的访问请求，需要登录时，重定向到 CAS Server。

下图是标准 CAS 最基本的协议过程：



CAS Client 与受保护的客户端应用部署在一起，以 Filter 方式保护受保护的资源。对于访问受保护资源的每个 Web 请求， CAS Client 会分析该请求的 Http 请求中是否包含 Service Ticket。如果没有，则说明当前用户尚未登录，于是将请求重定向到指定好的 CAS Server 登录地址，并传递 Service （也就是要访问的目的资源地址），以便登录成功过后转回该地址。

用户在上图流程中的第 3 步输入认证信息，如果登录成功，CAS Server 随机产生一个相当长度、唯一、不可伪造的 Service Ticket，并缓存以待将来验证。之后系统自动重定向到 Service 所在地址，并为客户端浏览器设置一个 Ticket Granted Cookie（TGC），CAS Client 在拿到 Service 和新产生的 Ticket 过后，在第 5，6 步中与 CAS Server 进行身份核实，以确保 Service Ticket 的合法性。

在 IDaaS 中， CAS （标准）应用模板实现了标准的 CAS 流程。它充当一个 CAS Server的角色。当 CAS Cient 决定使用IDaaS作为 CAS Server 时。在登录认证时需要使用 IDaaS 系统中公司的主账号，密码进行认证。

CAS的缺点

CAS虽然很高效的解决了原子操作问题，但是CAS仍然存在三大问题。

1. 循环时间长开销很大。
2. 只能保证一个变量的原子操作。
3. ABA问题。

4.认证

4.1. RADIUS 和 WIfi 认证

RADIUS

什么是 RADIUS?

RADIUS（Remote Authentication Dial-In User Server，远程认证拨号用户服务）是一种分布式的、C/S架构的信息交互协议，能包含网络不受未授权访问的干扰，常应用在既要求较高安全性、又允许远程用户访问的各种网络环境中。

协议定义了基于UDP（User Datagram Protocol）的RADIUS报文格式及其传输机制，并规定UDP端口1812、1813分别作为认证、计费端口。

如果是思科设备：认证和授权端口为UDP1645，计费端口1646.

RADIUS最初仅是针对拨号用户的AAA协议，后来随着用户接入方式的多样化发展，RADIUS也适应多种用户接入方式，如以太网接入等。它通过认证授权来提供接入服务，通过计费来收集、记录用户对网络资源的使用。

Radius的架构

客户端/服务器模式。

RADIUS客户端：一般位于网络接入服务器NAS（Network Access Server）上，可以遍布整个网络，负责传输用户信息到指定的RADIUS服务器，然后根据从服务器返回的信息进行相应处理（如接受/拒绝用户接入）。

设备作为RADIUS协议的客户端，实现以下功能：

支持标准RADIUS协议及扩充属性，包括RFC（Request For Comments）2865、RFC2866。

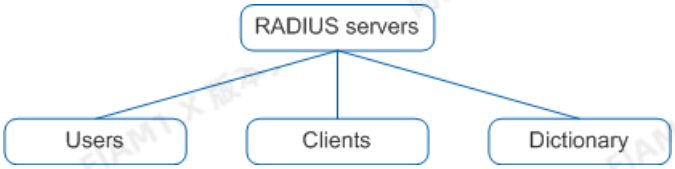
支持华为扩展的私有属性。

对RADIUS服务器状态的主动探测功能。

计费结束报文的本地缓存重传功能。

RADIUS服务器的自动切换功能。

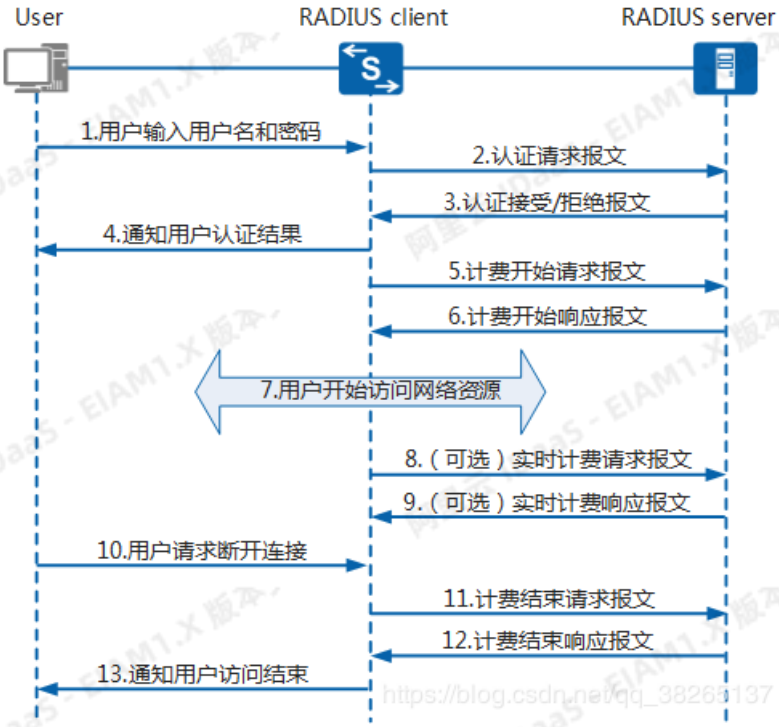
RADIUS服务器：一般运行在中心计算机或工作站上，维护相关的用户认证和网络服务访问信息，负责接收用户连接请求并认证用户，然后给客户端返回所有需要的信息（如接受/拒绝认证请求）。RADIUS服务器通常要维护三个数据库。



- Users：用于存储用户信息（如用户名、口令以及使用的协议、IP地址等配置信息）。
- Clients：用于存储RADIUS客户端的信息（如接入设备的共享密钥、IP地址等）。
- Dictionary：用于存储RADIUS协议中的属性和属性值含义的信息。

RADIUS工作原理

RADIUS认证、授权和计费：



RADISU报文类型：

目前RADIUS定义了十六种报文类型。

RADIUS认证报文：

报文名称	说明
Access-Request	认证请求报文，是RADIUS报文交互过程中的第一个报文，用来携带用户的认证信息（例如：用户名、密码等）。认证请求报文由RADIUS客户端发送给RADIUS服务器，RADIUS服务器根据该报文中携带的用户信息判断是否允许接入。
Access-Accept	认证接受报文，是服务器对客户端发送的Access-Request报文的接受响应报文。如果Access-Request报文中的所有属性都可以接受（即认证通过），则发送该类型报文。客户端收到此报文后，认证用户才能认证通过并被赋予相应的权限。
Access-Reject	认证拒绝报文，是服务器对客户端的Access-Request报文的拒绝响应报文。如果Access-Request报文中的任何一个属性不可接受（即认证失败），则RADIUS服务器返回Access-Reject报文，用户认证失败。
Access-Challenge	认证挑战报文。EAP认证时，RADIUS服务器接收到Access-Request报文中携带的用户名信息后，会随机生成一个MD5挑战字，同时将此挑战字通过Access-Challenge报文发送给客户端。客户端使用该挑战字对用户密码进行加密处理后，将新的用户密码信息通过Access-Request报文发送给RADIUS服务器。RADIUS服务器将收到的已加密的密码信息和本地经过加密运算后的密码信息进行对比，如果相同，则该用户为合法用户。

RADIUS计费报文：

报文名称	说明
Accounting-Request(Start)	计费开始请求报文。如果客户端使用RADIUS模式进行计费，客户端会在用户开始访问网络资源时，向服务器发送计费开始请求报文。
Accounting-Response(Start)	计费开始响应报文。服务器接收并成功记录计费开始请求报文后，需要回应一个计费开始响应报文。
Accounting-Request(Interim-update)	实时计费请求报文。为避免计费服务器无法收到计费停止请求报文而继续对该用户计费，可以在客户端上配置实时计费功能。客户端定时向服务器发送实时计费报文，减少计费误差。
Accounting-Response(Interim-update)	实时计费响应报文。服务器接收并成功记录实时计费请求报文后，需要回应一个实时计费响应报文。
Accounting-Request(Stop)	计费结束请求报文。当用户断开连接时（连接也可以由接入服务器断开），客户端向服务器发送计费结束请求报文，其中包括用户上网所使用的网络资源的统计信息（上网时长、进/出的字节数等），请求服务器停止计费。
Accounting-Response(Stop)	计费结束响应报文。服务器接收计费停止请求报文后，需要回应一个计费停止响应报文。

RADIUS授权报文：

报文名称	说明
CoA-Request	动态授权请求报文。当管理员需要更改某个在线用户的权限时（例如，管理员不希望用户访问某个网站），可以通过服务器发送一个动态授权请求报文给客户端，使客户端修改在线用户的权限。
CoA-ACK	动态授权请求接受报文。如果客户端成功更改了用户的权限，则客户端回应动态授权请求接受报文给服务器。
CoA-NAK	动态授权请求拒绝报文。如果客户端未成功更改用户的权限，则客户端回应动态授权请求拒绝报文给服务器。
DM-Request	用户离线请求报文。当管理员需要让某个在线的用户下线时，可以通过服务器发送一个用户离线请求报文给客户端，使客户端终结用户的连接。
DM-ACK	用户离线请求接受报文。如果客户端已经切断了用户的连接，则客户端回应用户离线请求接受报文给服务器。
DM-NAK	用户离线请求拒绝报文。如果客户端无法切断用户的连接，则客户端回应用户离线请求拒绝报文给服务器。

WIFI 认证

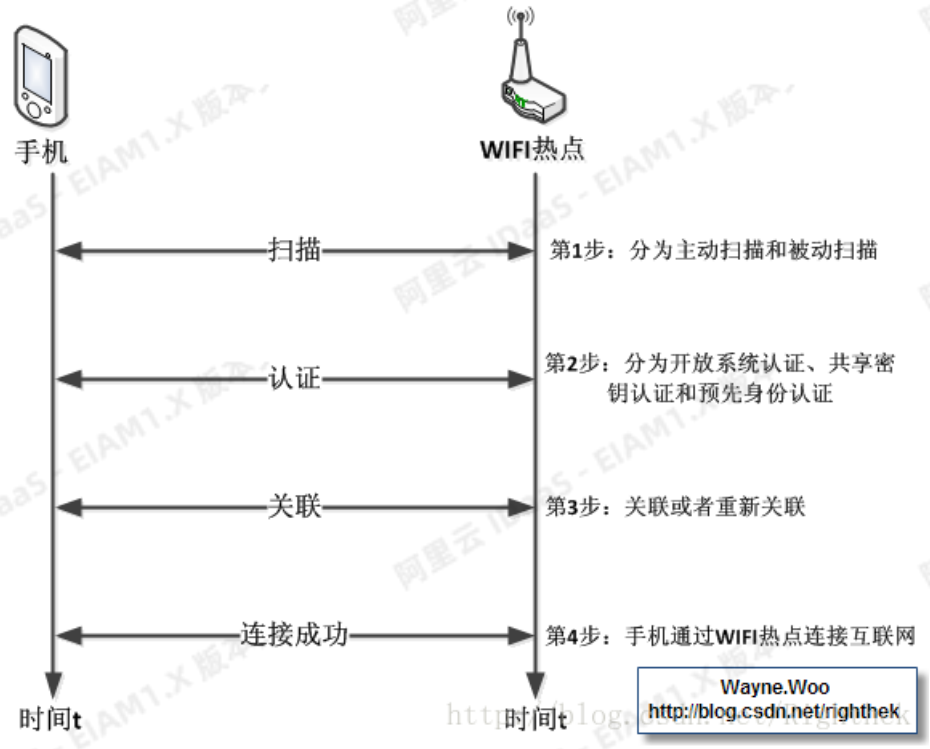
WIFI 认证三大流程

通常对于无线用户使用来说，在使用WIFI功能时，经常性的操作是打开手机上的WIFI设备，搜索到心目中的热点，输入密码，联网成功，各种低头上网。

这个看似简单的过程，背后却是隐藏着大量的无线通信技术。用几个专业术语来表示这个过程，分别是：

- **扫描（Scanning）**
- **认证（Authentication）**
- **关联（Association）**

下面用一张图来表示这个过程。



1）扫描

扫描又分为两种类型

主动扫描

即我们的手机（工作站STA）以主动的方式，在每个信道上发出Probe Request帧，请求某个特定无线网络予以回应。主动扫描是主动寻找网络，而不是静候无线网络声明本身的存在。使用主动扫描的工作站将会以如下的程序扫描信道表所列的频道：

（1）跳至某个信道，然后等候来帧指示（indicationof an incoming frame）或者等到ProbeDelay定时器超时。如果在这个信道收到帧，就证明该信道有用户在使用，因此可以加以探测。而ProbeDelay定时器可用来防止某个闲置信道让整个过程停止，因为工作站不会一直等待帧的到来。

（2）利用基本的DCF访问过程取得媒介使用权，然后送出一个Probe Request帧。

（3）至少等候一段最短的信道时间（即MinChannelTime）。

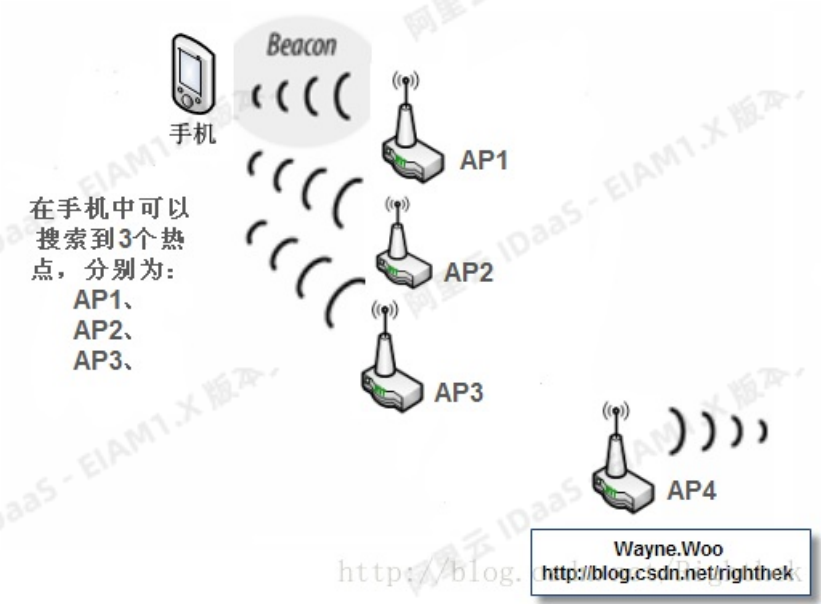
a.如果媒介并不忙碌，表示没有网络存在，因此可以跳至下个信道。

b.如果在MinChannelTime这段期间媒介非常忙碌，就继续等候一段时间，直到最长的信道时间（即MaxChannelTime）超时，然后处理任何的Probe Response帧。

被动扫描

现在大部分移动电子产品都是采用被动扫描（passive scanning）的方式，原因是扫描过程中不需要传送任何信号，可以省电。在被动扫描中，工作站会在信道列表（channel list）所列的各个信道之间不断切换，并静候Beacon帧的到来。所收到的任何帧都会被暂存起来，以便取出传送这些帧的BSS 的相关数据。

在被动扫描的过程中，工作站会在信道间不断切换，并且会记录来自所收到的任何Beacon的信息。Beacon在设计上是为了让工作站知道加入某个基本服务集（Basic Service Set，简称 BSS）所需要的参数以便进行通信。在下图中，通过监听来自前三个接入点的Beacon帧，移动式工作站以被动扫描找出该区所有BSS。如果该工作站并未收到来自第四个接入点的Beacon，就会汇报目前只发现的三个BSS。



2）认证

无线网络所使用的媒介是经过特殊编码与调制过的无线电波链路。它是众所周知的开放标准，只要对无线传输有所研究和了解的人，都可以在无线覆盖范围内发送和接收信号。因此对数据的拦劫和非法灌注是多么的简单！

由于无线网络的最大缺陷就是安全性，进行身份认证是必须的，同时认证的连接工作也必须予以加密，以防未经授权的使用者访问。

早期的IEEE802.11定义了两种认证方式：

（1）开放系统认证（OpenSystem authentication）；

（2）共享密钥认证（SharedKey authentication）。

开放系统认证是IEEE802.11默认的认证方式，实质上并没有做认证。连接无线网络时，基站并没有验证工作站的真实身份。认证过程由以下两个步骤组成：第一，工作站发送身份声明和认证请求；第二，基站应答认证结果，如果返回的结果是“successful”，表示两者已相互认证成功。

共享密钥认证依赖于WEP（Wired Equivalent Privacy，有线等效加密）机制，而上文已经提到WEP渐渐被淘汰，我们就不讲解了，接下来侧重讲解与我们家庭无线网络息息相关的WPA-PSK/WPA2-PSK认证机制。当我们进入无线路由器设置界面，打开无线安全设置，就会看到以下信息，如图1。WPA（Wi-Fi Protected Access）是WIFI联盟制定的安全性标准，WPA2是第二个版本。PSK（PreShared Key）叫做预共享密钥。WPA-PSK/WPA2-PSK主要是针对个人或家庭网络等，对安全性要求不是很高的用户。而WPA /WPA2是针对企业的，对安全性要求很高的用户，在WPA/WPA2选项中，大家可以看到它比WPA-PSK/WPA2-PSK多了一个Radius服务器，这个就是认证服务器。而对我们家庭网络来说，适合选择WPA-PSK/WPA2-PSK选项，因为我们不需要认证服务器。现有我们来详细分析WPA-PSK/WPA2-PSK认证机制。

1、WPA-PSK

WPA-PSK（Wi-Fi ProtectedAccess，Wi-Fi保护访问）是WiFi联盟推出的标准，它是为兼容原有的WEP硬件产品，所以它采用的模式是：

WPA-PSK = PSK + TKIP + MIC

PSK：PreShared Key，预共享密钥。

TKIP：Temporal KeyIntegrity Protocol，临时密钥完整性协议。

MIC：Message IntegrityCode，消息完整性校验码。

TKIP的开发目的是为了提高原有的基于WEP硬件的安全性，因此它和WEP一样都是采RC4加密算法，同时保留了WEP的基本结构和操作方式。TKIP是一种过渡的加密协议，现已被证明安全性不高。因此，我们就不去详细的讲解了。

2、WPA2-PSK

WPA2是在802.11i颁布之后，WIFI联盟随即推出的最新无线安全标准，它遵循802.11i标准，以下是它采用的模式：

WPA2-PSK= PSK + AES + CCMP

PSK：PreShared Key，预共享密钥，它是一种802.11身份验证方式，以预先设定好的静态密钥进行身份验证，此密钥必须手动进行传递，即是我们的手机连接WIFI热点时需要输入的密码。

AES：Advanced EncryptionStandard，高级加密标准。AES是美国NIST制定的替代DES的分组加密算法。AES具有优秀的密钥扩展方案，灵活的密钥生成算法。算法对内存要求极低，即使在限制较大的环境中也能获得很好的性能。分组和密钥被设计成可以在三种长度中自由选择的形式，AES具有128、192、256位的密钥。802.11规定CCMP中的AES使用的是128位密钥，它的加密块大小也是128位。

CCMP：Counter modewith Cipher-block chaining Message authentication code Protocol，计数器模式及密码块链消息认证码协议。它是基于高级加密标准（AES）的CCM（CTR with CBC-MAC）模式。CCM是一种通用的模式，它可以使用在任何成块的加密算法中。CCM模式使用CTR（Counter Mode）提供数据保密，并采用密码块链信息认证码（Cipher-Block Chaining with Message Authentication Code，CBC-MAC）来提供数据认证和完整性服务。

3）关联

工作站与基站进行关联，以便获得网络的完全访问权。关联属于一种记录（record keeping）过程，它让分布式系统（Distribution System）得以记录每个移动式工作站的位置，以便将传送给移动式工作站的帧，转送给正确的基站。形成关联之后，基站必须为该移动式工作站在网络上注册，如此一来，发送给该移动式工作站的帧，才会转送至其所属基站。其中一种注册方式是送出一个ARP信号，让该工作站的MAC地址得以跟与基站连接的交换端口形成关联。

关联只限于基础型（Infrastructure）网络，在逻辑上等同于在有线网络中插入网线。一旦完成此过程，无线工作站就可以通过分布式系统连接互联网，而其他人也可以经由分布式系统予以回应。IEEE802.11在规格中公开禁止工作站同时与一个以上的基站形成连接。

和认证一样，关联过程是由移动式工作站发起的。在此并不需要用到顺序编号，因为关联程序只牵涉到三个步骤。其中所用到的两个帧，被归类为Association管理帧。和单点传播（Unicast）管理帧一样，关联程序的步骤是由一个连接帧及必要的链路层回应所组成：

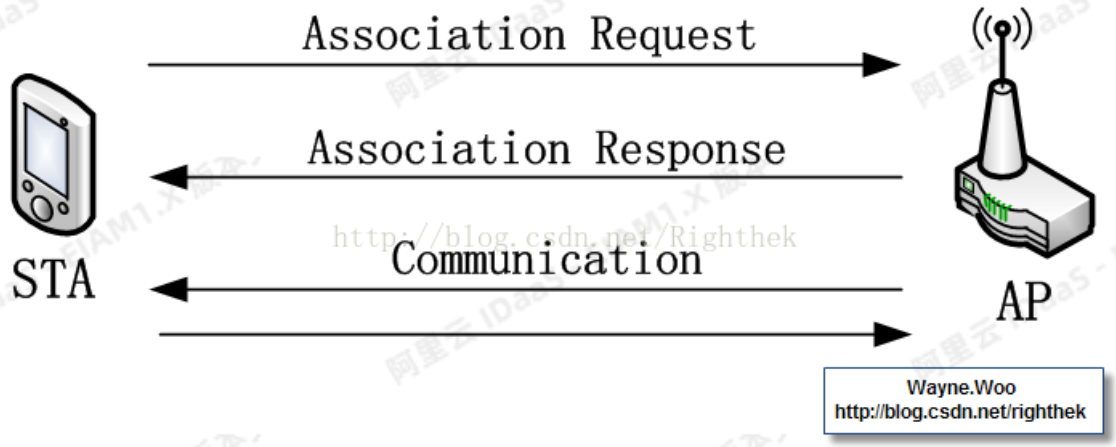
1、一旦移动式工作站与基站完成认证，便可送出关联请求（Association Request）帧。尚未经过身份认证的工作站，会在基站的答复中收到一个解除关联（Deauthenticaton）帧。

2、基站随后会对关联请求进行处理。IEEE802.11标准并未规范如何判断是否请允许关联；这因基站的实现而异。较常见的方式是考虑帧暂存所需要的空间大小。以Association Request帧中的聆听间隔（Listen Interval）字段来推算，大致上可以粗略推算出。

a. 一旦关联请求获准，基站就会以代表成功的状态代码0及关联识别码（Association ID，简称 AID）来回应。AID本身是数值形式的识别码，在逻辑上则是用来辨识暂存帧所要传递的行动式工作站。

b. 关联请求如果失败，就只会返回状态码，并且中止整个过程。

3、基站开始为移动式工作站处理帧。在常见的产品中，所使用的分布式系统媒介通常是Ethernet。当基站所收到的帧目的地为与之关联的移动式工作站时，就会将该帧从 Ethernet桥接至无线媒介，如果该移动式工作站处于省电（Power-Saving）状态，则为之暂存帧。在共享式Ethernet中，该帧会被送至所有基站，不过只有正确的基站会进行桥接处理。在交换式Ethernet 里，该工作站的MAC地址得以跟某个特定的交换端口（Switch Port）形成关联。当然，该交换端口必须连接到当前为该工作站提供服务的基站。



4.2. 多因素认证

概述

多因素验证指的是用户需要通过多种认证机制的校验之后，才可以获取到授权。通常情况下，我们使用用户名+密码作为身份验证方式。但这是不够安全的，攻击者一旦获取到密码，就可以冒用用户的身份进行攻击。

多因素认证在账户名+密码的验证方式的基础上，多加了一重或者多重认证方式以增强应用程序和服务的安全性。

认证因素

认证因素可以分为三大类：知识因素、拥有因素和内在因素。

知识因素

知识因素指的是用户的获取授权的时候，需要先回答用户设置的安全性问题。安全性问题通常是仅有用户本人知道正确答案的，所以可以在一定程度上保证安全性。但由于安全性问题的答案与密码类似，是比较固定的一串字符串，攻击者可以通过猜测或者暴力破解来获得问题的正确答案。

拥有因素

拥有因素指的是用户独自拥有的东西。拥有因素是一个比较宽泛的认证因素，我们还可以继续细分为以下几类：

硬件设备

硬件设备指的是使用特定的硬件设备进行用户身份的验证。如网上银行的 U盾、U key 证书等。相对知识因素来讲，由于使用到了用户拥有的设备，更加的安全。缺点是需要多携带一个硬件设备，并且在设备遗失或被盗时，会影响用户的正常使用，有可能还会造成安全问题。

一次性密码(OTP)

一次性密码（OTP）指的是通过短信、邮件发送给用户的密码。通过向用户拥有的设备（通常是手机）发送的一次性密码来验证用户身份。一次性密码不需要使用额外的硬件设备，只需要携带手机即可，更加的便捷。针对手机的攻击会对 OTP 验证的安全性造成一定的影响，比如攻击者通过克隆 SIM 卡来获取用户的OTP码。

应用程序生成的密码（TOTP）

应用程序生成的密码，主要是通过TOTP算法（Time-based One-Time Password），根据用户和服务器共有的密钥和当前的时间计算出数字码。通过对比服务器的数据码和用户的数字码，来允许或拒绝用户的访问。是安全性比较高的认证因素。

内在因素

内在因素指的是个人的生物因素，如指纹、声纹、人脸等。个人的生物因素相对来说最难以仿冒和破解，因此也相对最安全。缺点是为了确保安全性和准确性，需要对生物识别的硬件设备进行投资。

IDaaS的多因素认证

IDaaS主要是以OTP和TOTP作为二次认证的验证因素，用户可以自主选择向手机发送OTP码或者使用移动端的TOTP码进行二次认证。同时您也可以选择开通人脸识别服务，使用人脸作为二次认证的安全验证因素。

4.3. OTP 动态口令

为什么要用动态口令？

在对外网开放的后台管理系统中，使用静态口令进行身份验证可能会存在如下问题：

（1） 为了便于记忆，用户多选择有特征作为密码，所有静态口令相比动态口令而言，容易被猜测和破解；

（2） 黑客可以从网上或电话线上截获静态密码，如果是非加密方式传输，用户认证信息可被轻易获取；

（3） 内部工作人员可通过合法授权取得用户密码而非法使用；

静态口令根本上不能确定用户的身份，其结果是，个人可以轻松地伪造一个假身份或者盗用一个已有使用者的身份，给企业造成巨大的经济和声誉损失。本文主要介绍并实现了一种动态口令（OTP）的实现方式。

什么是动态口令？

动态口令（OTP，One-Time Password）又称一次性密码，是使用密码技术实现的在客户端和服务器之间通过共享秘密的一种认证技术，是一种强认证技术，是增强目前静态口令认证的一种非常方便技术手段，是一种重要的双因素认证技术，动态口令认证技术包括客户端用于生成口令产生器的，动态令牌，是一个硬件设备，和用于管理令牌及口令认证的后台动态口令认证系统组成。

一次性密码的工作方式

在基于OTP的身份验证方法中，用户的OTP应用程序和身份验证服务器依赖共享机密。使用哈希消息认证码（HMAC）算法和移动因子（例如基于时间的信息（TOTP）或事件计数器（HOTP））来生成一次性密码的值。OTP值具有分钟或第二个时间戳，以提高安全性。一次性密码可以通过多种渠道传递给用户，包括基于SMS的文本消息，电子邮件或端点上的专用应用程序。

长期以来，安全专业人员一直担心SMS消息欺骗和中间人（MITM）攻击可用来破坏依赖一次性密码的2FA系统。但是，美国国家标准技术研究院（NIST）宣布了计划弃用2FA和一次性密码使用SMS的计划，因为该方法容易受到各种攻击的攻击，这些攻击可能会破坏这些密码和代码。因此，考虑一次性部署密码的企业应该探索SMS之外的其他交付方式。

一次性密码的好处

一次性密码避免了IT管理员和安全经理面临密码安全性的常见陷阱。他们不必担心组合规则，已知的错误和弱密码，共享凭据或在多个账户和系统上重复使用相同的密码。一次性密码的另一个优点是，它们会在几分钟内失效，这可以防止攻击者获取密码并再次使用它们。

OTP 的三种形式

OTP 从技术来分有三种形式， 时间同步、事件同步、挑战/应答。

（1） 时间同步

原理是基于动态令牌和动态口令验证服务器的时间比对，基于时间同步的令牌，一般每60秒产生一个新口令，要求服务器能够十分精确的保持正确的时钟，同时对其令牌的晶振频率有严格的要求，这种技术对应的终端是硬件令牌。

（2）事件同步

基于事件同步的令牌，其原理是通过某一特定的事件次序及相同的种子值作为输入，通过HASH算法中运算出一致的密码。

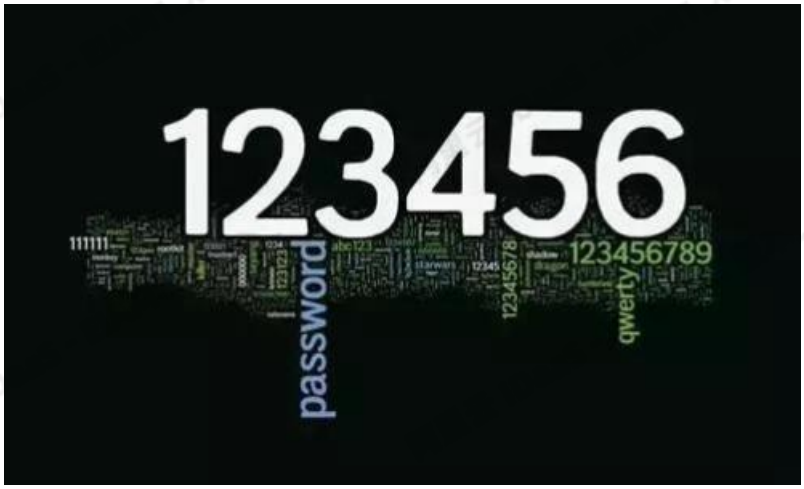
（3）挑战/应答

常用于的网上业务，在网站/应答上输入服务端下发的挑战码，动态令牌输入该挑战码，通过内置的算法上生成一个6/8位的随机数字，口令一次有效，这种技术目前应用最为普遍，包括刮刮卡、短信密码、动态令牌也有挑战/应答形式。

4.4. 弱密码的风险（需要修改）

弱密码对于依靠云服务的企业来说是一种常见的威胁。云服务在过去几年如雨后春笋般崛起，并被大量的个人和公司广泛使用。然而，大量的云服务和应用也带来了许多需要记住的，用以连接和使用这些云服务的密码。

有这么多可以通过某种凭证，例如一个密码、一个PKI密钥或别的什么方式来访问的云服务，自然也让攻击者有了很多的机会来获取云服务的访问。在大多数情况下，只要提供正确的密码就可以从世界任何地方，通过互联网来访问云服务。这就是为什么他们是单点故障；弱的云密码可以被黑客轻易取得来获得对云服务的访问。



要防范弱密码的问题，我们在设置或更改密码时使用最佳的密码安全措施是非常重要的，这包括：

初始密码：

如果密码是由第三方设定为一个初始的默认值，请重置它，这样它就不会被存储在历史或缓存的某处，导致整体安全性降低。

共享密码：

设定共享密码时，请选择一个没有在其他任何地方使用的密码。如果你在另一个服务也使用相同的密码，攻击者可以同时获得两个云服务的访问。

密码有效时间：

假定攻击者已经破解了密码，并可以访问云服务，那么每90天修改一次密码就非常关键。这种做法有助于防止攻击者进一步取得认证并窃取更多的敏感信息。

密码最短长度：

密码长度应至少8位，虽然我们通常建议更长的密码。为了安全起见，造一个句子来作为你的密码。

密码强度：

密码应该同时使用小写和大写字母，数字和特殊字符。这确保攻击者在暴力破解密码时必须通过更多数量的组合才能成功。

密码历史：

保存并使用密码的历史版本，这让系统能够比较当前密码与历史密码并确定有些密码是否会过于相似。如果过于相似的话，应该拒绝本次密码更改。

5.授权

5.1. 授权概览

什么是授权 (Authorization)?

广义上的授权：

是上级将完成某项工作所必须的权力授给部属人员；是领导者通过为员工和下属提供更多的自主权，以达到组织目标的过程。

信息系统中的授权：

是管理员将某些资源的访问、管理、操作等权限赋予用户，达到管理和使用的目的。譬如主机的访问使用权限，某项功能菜单的使用权限亦或是某个数据的读写权限。

本文将对信息系统中的授权进行着重讲解

授权的意义

授权管理是所有业务系统不可缺少的一部分！

企业角度：

1）贴合管理制度：随着公司的建设发展过程中，组织或岗位的职责越来越清晰，边界越来越明确，所以授权成为公司管理的必要手段。不同的岗位，职责需要进行不同的授权

2）保障数据安全：数据是企业的核心资产，价值不可估量。授权能让公司的数据安全得到保障，不同的权限能看到及操作不同的数据，防止用户在误操作、人为破坏、数据泄露等

3）提升工作效率：好的授权，权限管理，会让员工的工作效率得到提升。让系统更加易容，让和业务操作变得更加简单，数据获取分析变得更加方便快捷。

系统建设角度

1）安全性：保护信息系统的操作使用安全，数据安全，防止泄露，违规操作。

2）商业性：提升产品的市场竞争价值，好的授权功能，能容易获得用户青睐。

3）易用性：让系统使用更加便捷，提升用户体验，对管理员更为友好。

授权的分类

三级授权

我们将整体授权类型划分为三级

（1）一级权限：访问权限

（2）二级权限：菜单、按钮权限

（3）三级权限：数据权限

依据不同等级的授权，来控制授权的最小的颗粒度。

访问控制

用户系统登录访问控制，身份认证

菜单、按钮控制

业务系统的一级、二级、三级功能菜单的授权控制，以及操作按钮等使用控制

数据控制

数据增、删、改、查及其他操作的授权控制

常用的授权模型

常用的授权模型主要有 3 种：

ACL	访问控制列表
RBAC	基于角色的权限控制
ABAC	基于属性的权限控制

ACL（Access Control List），在ACL中，包含用户（User）、资源（Resource）、资源操作（Operation）三个关键要素。

通过将资源以及资源操作授权给用户而使用户获取对资源进行操作的权限。

RBAC（Role-Based Access Control ），是把用户按角色进行归类，通过用户的角色来确定用户能否针对某项资源进行某项操作。RBAC相对于ACL最大的优势就是它简化了用户与权限的管理，通过对用户进行分类，使得角色与权限关联起来，而用户与权限变成了间接关联。

ABAC（Attribute Base Access Control） 基于属性的权限控制

不同于常见的将用户通过某种方式关联到权限的方式，ABAC则是通过动态计算一个或一组属性来是否满足某种条件来进行授权判断（可以编写简单的逻辑）。属性通常来说分为四类：用户属性（如用户年龄），环境属性（如当前时间），操作属性（如读取）和对象属性，所以理论上能够实现非常灵活的权限控制，几乎能满足所有类型的需求。

5.2. 三级权限管理

基于RBAC模型的权限管理， 可以分为三级

（1）一级权限：应用访问权限，也就是用户可以访问哪些应用

（2）二级权限：菜单访问权限，用户可以访问一个应用中的哪些菜单和按钮的权限

（3）三级权限：数据访问权限，用户可以访问某个菜单下的哪些数据的权限

场景举例

一级权限

1. 给张三赋予“人力资源经理”角色，可以访问OA办公系统的权限。张三可以登录OA系统，访问OA中的所有页面，具有“查询员工”、“添加员工”、“修改员工”和“删除员工”权限。
2. 李四不是人力的同学，没有权限登录OA系统。

二级权限

1. 给张三赋予“人力资源专员”角色，可以访问OA办公系统的权限，但是不能看到OA所有的页面，比如只能看到添加员工的页面，具有“添加员工”的权限。

三级权限

1. 因为张三是北京分公司的“人力资源经理”，所以他能够也只能够管理北京分公司员工和北京分公司下属的子公司（海淀子公司、朝阳子公司、东城区子公司等）的员工。
2. 因为李四是海淀子公司的“人力资源经理”，所以他能够也只能够管理海淀子公司的员工。

5.3. RBAC

RBAC是什么？

RBAC 是基于角色的访问控制（Role-based access control ）在 RBAC 中，权限与角色相关联，用户通过成为适当角色的成员而得到这些角色的权限。这就极大地简化了权限的管理。这样管理都是层级相互依赖的，权限赋予给角色，而把角色又赋予用户，这样的权限设计很清楚，管理起来很方便。

RBAC 认为授权实际上是Who、What 、How 三元组之间的关系，也就是 Who 对 what 进行 How 的操作，也就是“主体”对“客体”的操作。

Who：是权限的拥有者或主体（如：User，Role）。

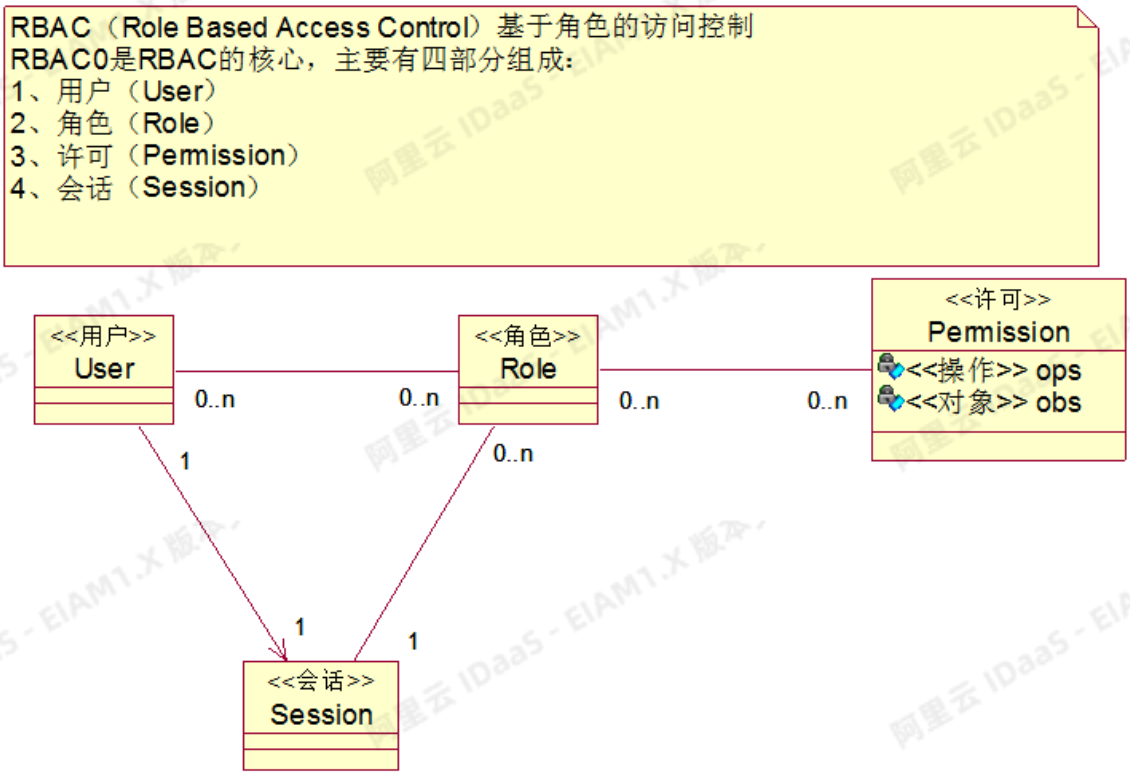
What：是操作或对象（operation，object）。

How：具体的权限（Privilege，正向授权与负向授权）。

RBAC 的四个模型

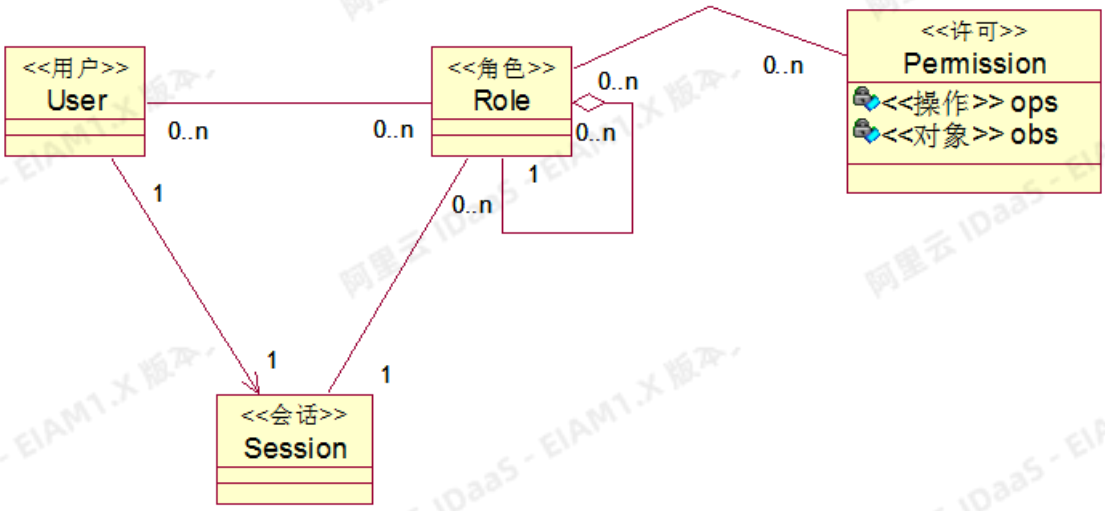
- RBAC0：是RBAC的核心思想。
- RBAC1：是把RBAC的角色分层模型。
- RBAC2：增加了RBAC的约束模型。
- RBAC3：其实是RBAC2 + RBAC1。

RBAC0 基本模型



RBAC1 基于角色的分层模型

RBAC（Role Based Access Control）基于角色的访问控制
RBAC1是对RBAC0进行了扩展，是RBAC的角色分层模型，RBAC1引入了角色继承概念，有了继承就有了上下级的包含关系

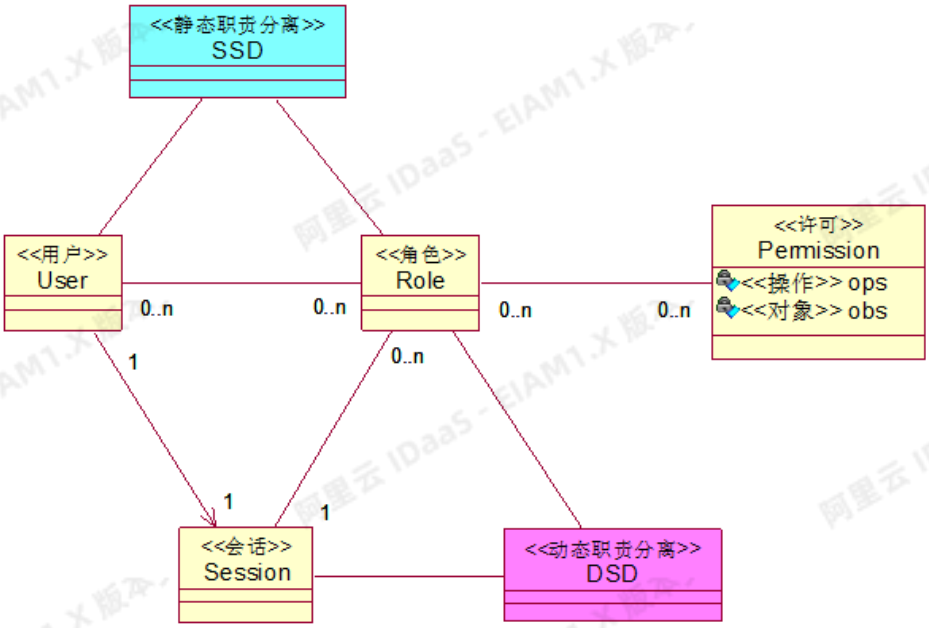


RBAC2 约束模型

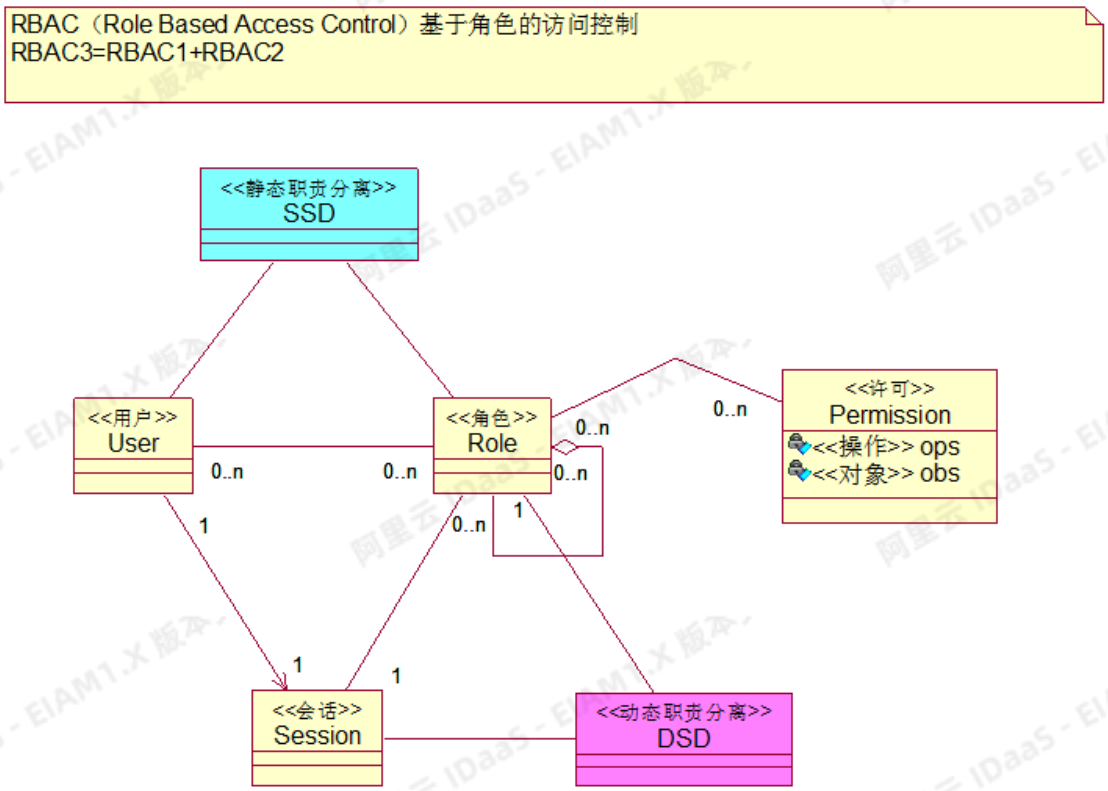
RBAC（Role Based Access Control）基于角色的访问控制
RBAC2是基于RBAC0扩展的，主要引入了SSD（静态职责分离）和DSD（动态职责分离）

SSD主要应用在用户和角色之间（授权阶段），主要约束：
1、互斥角色，同一个用户不能授予互斥关系的角色，如：不能同时授予会计和出纳的角色
2、基数约束，一个用户拥有的角色是有限的，一个角色拥有的许可是有限的
3、先决条件约束，用户想得到高级权利，必须先拥有低级权利

DSD会话和角色之间的约束，主要动态决定怎么样计划角色，如：一个用户拥有5个角色，只激活2个



RBAC3 就是RBAC1+RBAC2



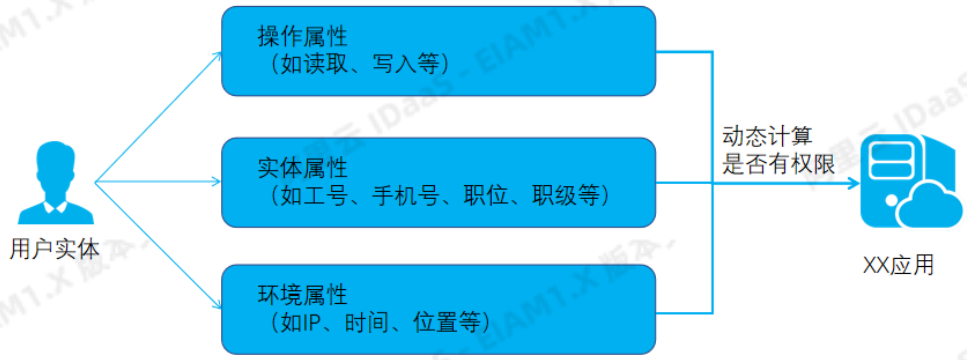
5.4. 基于属性的访问控制（ABAC）

什么是基于属性的访问控制模型

基于属性的访问控制（Attribute-Based Access Control，下文简称ABAC）是一种灵活的授权模型。是通过实体的属性、操作类型、相关的环境来控制是否有对操作对象的权限。

- 例如：P5（职级）的同学有OA系统的权限。
  - 上述是一个简单的ABAC的例子，就是通过实体的职级这一属性来控制是否有OA系统的权限。
- 再比如：P5（职级）的研发（职位）同学有公司Gitlab的权限。
  - 上述例子是通过一组实体的属性（职级和职位）来控制对操作对象的权限。
- 再比如：P5（职级）的研发（职位）同学在公司内网（环境）可以查看和下载（操作）代码。
  - 上述例子显然比之前两个更加复杂，除了判断实体的属性（职级和职位），还判断了当前的环境属性和操作属性。

我们可以ABAC的访问控制模型用下面这张图表现出来：



ABAC的使用场景

ABAC授权模型理论上能够实现非常灵活的权限控制，几乎能满足所有类型的需求。从使用场景来说比较适用于用户数量多并且授权比较复杂的场景。简单的场景也是可以使用ABAC的，但是使用基础的ACL或者RBAC也能满足需求。

场景一：

还是拿上面的例子来说：P5（职级）的研发（职位）同学在公司内网（环境）可以查看和下载（操作）代码。在需要根据环境属性和操作属性来动态计算权限的时候，使用其他的授权模型可能不太能满足需求。这个时候就需要使用ABAC授权模型。

场景二：

ABAC也适用于公司成员（角色）快速变化的场景，由于ABAC 是通过用户的属性来授权的。在新建用户/修改用户属性时会自动更改用户的权限，无需管理员手动更改账户角色。在属性的组合比较多，需要更细粒度地划分角色的情况下，RBAC需要建立大量的角色，ABAC授权模型会更加灵活。

ABAC对比RBAC

ABAC的优势：

- 对于大型组织，基于RBAC的控制模型需要维护大量的角色和授权关系，相比而言，ABAC更加灵活；
- 新增资源时，ABAC仅需要维护较少的资源，而RBAC需要维护所有相关的角色，ABAC可扩展性更强、更方便。
- ABAC支持带有动态参数的授权规则，RBAC只能基于静态的参数进行判断。
- ABAC权限控制的粒度比RBAC更细。

RBAC的优势：

- 对于中小型组织，维护角色和授权关系的工作量不大，反而定制各种策略相对麻烦，更容易接受RBAC授权模型。

IDaaS中的ABAC

IDaaS中的授权模型主要是基于用户属性（扩展字段）的授权。管理员可以在IDaaS中创建各个扩展字段，并且为账户分配这些字段的字段值。然后在分类管理中，根据扩展字段以及字段值创建分类，基于分类实现应用和权限资源的授权管理。

例如：根据用户的某一个属性动态分配权限。

在IDaaS分类管理中，可以根据某一个扩展字段和字段值创建一个分类，可以实现根据分类授权应用的功能。