

开发人员和身份服务

— 通过身份中心处理身份数据

Oracle 白皮书
2008 年 9 月

开发人员和身份服务

一 通过身份中心处理身份数据

执行概要	3
引言	3
围绕身份数据开发 — 一项艰巨的挑战	4
身份存储.....	4
身份协议.....	4
身份模式.....	5
身份数据安全.....	5
身份隐私和保护	6
身份中心 — 缺少的环节	7
部署提取	7
安全特性.....	8
通过 ORACLE Virtual Directory 实现身份中心	9
通过身份治理框架声明身份要求	10
结论	12

开发人员和身份服务

— 通过身份中心处理身份数据

执行概要

面向服务的安全性 (SOS) 与整个 Oracle 融合中间件平台以应用为中心的整体方法相一致 — 其目标是提供一个全面的、基于标准的、开发人员友好的平台。通过利用和共享许多共同的身份服务，SOS 使得开发人员可以将精力用在最值得投入的工作上 — 应用程序逻辑本身。但是，要真正摆脱传统的、基于孤岛的处理身份管理的方法，构建识别身份的应用程序的开发人员必须挖掘这些服务的优势，并了解在其应用程序设计中使用这些服务的时机和使用方式。

这是“开发人员和身份服务”系列白皮书中的第一个，每个白皮书旨在重点介绍身份服务这一特殊的领域（从开发人员的角度）。本文的创作基础是以前在一篇题为“*面向服务的安全性 — 以应用为中心 看待身份管理 (Service-Oriented Security – An Application-Centric Look at Identity Management)*”的白皮书中定义的身份服务和身份外化的概念，可登录下列站点查阅：

http://www.oracle.com/technology/products/id_mgmt/pdf/serv_oriented_sec.pdf

在本白皮书，我们将仔细研究有关身份数据的当前访问模式的问题，并讨论一个理想的 SOS 身份中心 (Identity Hub) 应为开发人员提供什么？

引言

了解每种类型的身份服务和及其使用情况和实现本身一样至关重要。对于开发人员而言，在所有与身份相关的任务中，能够访问身份数据也许是任何识别身份的应用程序最基础的部分。然而，这并不是无足轻重的任务。诸多因素以及身份数据的性质给开发人员带来众多的挑战。

“但是，要真正摆脱传统的、基于孤岛的
处理身份管理的方法，构建识别身份的应用程
序的开发人员必须挖掘[身份服务]的优势，并
了解在其应用程序设计中使用这些服务的时机
和使用方式”。

“对于开发人员而言，在所有与身份相关的任务中，能够访问身份数据也许是任何识别身份的应用程序最基础的部分。”

一般来说，身份数据的存储、分发和使用方式与部署有关。开发人员必须设法考虑这些变化，决不要对客户环境强加不必要的特定应用限制。身份数据也是敏感数据。当应用程序能够访问身份来源以及能够处理并管理身份数据时，安全即成为问题。身份数据的敏感性还带来一系列公司现在必须遵守的合规性和隐私法规。

围绕身份数据开发 — 一项艰巨的挑战

应用程序如何使用身份数据是开发识别身份的应用程序最具有挑战性的一个方面。身份数据不同于常规的应用数据，因为有许多政策、隐私、安全和部署事项都必须予以考虑。

身份存储

传统应用程序在运行时通常依赖应用程序本身的存储作为主身份数据提供方。采用常规方法分发身份数据来源时，同步就成为此方法保持本地存储最新的一个关键因素。减少每一个应用程序的同步需求，并将身份数据外化为可跨应用程序共享的集中身份信息库，这样才能简化该问题。集中存储始终存在同步的需求。实际上，并非一切都可实现集中，开发人员通常需要处理多个身份信息库。例如，人力资源管理系统 (HRMS) 不应该将一名员工的工资或社会保险编号同步到任何外部资源，即使 HRMS 外部的其他应用可能需要该信息也不可以。

“权威性”往往是这种应用数据和集中数据分离的原因。如果将某个特定应用看作某个特定身份属性的唯一来源，则应将该应用视为此属性的“权威”。因此，HRMS 对员工工资或社会保险编号具有权威性。最后，为了创建应用所需的单个身份的完整表述，开发人员需要通过有点复杂的流程来使不同存储中的身份记录相关联，因为组成一个身份的不同属性跨多个身份存储分布。

“实际上，并非一切都可实现集中，开发人员通常需要处理多个身份信息库”。

身份协议

采用 RDBMS 或 LDAP 形式的身份信息库在提供身份数据方面发挥了核心作用。不同类型的信息库对应不同类型的协议。

目前开发人员必须装备自己来处理不同类型协议的各个方面，包括存储不同类型的连接信息（如数据库连接字符串或 LDAP URL）以及了解如何使用各种协议本身（如 JDBC 和 JNDI）来检索信息。随着联合协议和基于 Web 的协议（如 SAML、WS-*、SOAP 等）的出现，这些协议的性质和用途变得越来越多样和复杂。

身份模式

正确设置访问后端数据的协议后，开发人员现在必须浏览不同种类的身份模式以请求或提取数据。可以通过企业 RDBMS 的某一特定表的某一特定列检索用户的电子邮件。如果一个企业针对身份数据使用集中 LDAP，则开发人员可以使用 JNDI 查询，该查询将从用户的 LDAP 可识别名称中检索“邮件”属性。LDAP 提供了众所周知的“对象类”，这些类为公共对象（如用户、组和组织等）定义了一组标准的 LDAP 模式。然而，在使用自定义 LDAP 模式的情况下，客户端应用程序可能需要做出相应的反应。在使用 SAML 的情况下，通过用户的 SAML 断言中的“属性断言”可提供一组用户配置属性（包括电子邮件）。同样，包含电子邮件值的实际属性及该值的表示方式可能会因使用的 SAML 配置文件的类型而有所不同，而该配置文件又取决于 SAML 身份提供方和 SAML 服务提供方的握手配置方式。如果应用程序需要写回信息以更新任何身份信息时，则了解身份模式的重要性变得更为关键。

“归根结底，数据的存储方式和表示方式取决于客户部署环境 — 即使借助强大的互操作标准”。

经仔细研究发现，信息库、协议和模式方面出现的问题大多与部署时间相关。归根结底，数据的存储方式和表示方式取决于客户部署环境 — 即使借助强大的互操作标准。当面临各种身份信息库、协议和身份模式选择时，开发人员通常不得不采用最简便的方法来获得更简单、更易管理的解决方案 — 也许通过仅限于单个私有存储，该存储可通过单个协议（其中的身份数据采用众所周知、甚至专有的模式表示）使用单个 API 访问。至此，我们兜了一圈又回到了我们极力设法避免的特定应用孤岛方法。

身份数据安全

访问控制

一旦开发人员解决了访问身份数据的问题，则下一个难题是如何控制数据访问。在信息库级别始终有某种形式的内部安全机制 — 即在存储机制本身内部实现的安全机制，用以控制数据访问。

然而，要定义内部访问策略，内部存储必须识别正在访问它的实体 — 无论是用户、应用程序，或者是二者的组合。举 LDAP 为例，要通过 LDAP 访问控制来定义某个应用程序的访问控制，则该应用程序必须是一个采用可识别名称形式的可识别实体，该实体可通过 LDAP 绑定和查询。换言之，该应用程序的足迹必须存在。这同样适用于 RDBMS，其中数据库用户是在可在数据库内部定义的访问控制之前定义的。

“然而，要定义内部访问策略，内部存储必须识别正在访问它的实体 — 无论是用户、应用程序，或者是二者的组合”。

随着应用程序反复来回，这会给身份存储管理员带来麻烦和潜在的安全漏洞，他们通常不欢迎这些类型的应用程序特定的安全要求。在某些情况下，需要同时基于应用程序上下文和用户上下文来确定安全机制，诸如 LDAP 之类的协议根本不具备处理安全事务的能力。例如，一个薪金发放应用程序和 HR 应用程序具有登录用户数据的不同访问权，不同的应用程序据此来获取相应的用户属性。如果没有必要的后端支持，开发人员将很难解决这个棘手的问题。

委托

委托是开发人员遇到的另一个棘手的情况。例如，经理可以代表休假的直接下属执行某些任务。要精确地捕捉合适的委托及这些任务所需的访问控制往往很困难。从只读角度看，授予经理的查看身份数据的访问权应该与其直接下属相同或者该经理至少有足够的权限代表其直接下属执行操作 — 但仅此而已。此外，对身份数据或所做记录进行的任何操作或更改都应记录为此经理的行为。必须认真地控制和审查访问。同样，要在应用程序端实现这些访问控制对开发人员是很困难的。多个促进这些访问控制和策略的条件可能驻留在身份层 — 如经理/直接下属关系等。

“难以定义合适的访问级别，主要是由于无法定义合适的上下文”。

难以定义合适的访问级别，主要是由于无法定义合适的上下文。今天的访问控制可以处理一维上下文，如对某个特定用户或应用程序的访问。但是，更为复杂的安全要求正在迫使访问控制处理多维上下文 — 以了解完整的上下文，包括用户、应用程序、目标、协议、用途等的结合，以评估合适的访问级别。

身份隐私和保护

由于新的法规和合规性要求不断出台，数据保护已不只是保护数据源那么简单。现在，在处理应用程序本身使用和管理身份数据的方式方面，开发人员面临着许多新出现的要求。

身份数据生命周期

欧盟 Directive on Data Protection 等法规均规定，对不再需要或闲置一段时间的信息必须处置和/或归档。对于开发人员而言，这可能会影响某些信息在应用程序中的保留时间。

最少信息方法

请考虑此情况，某个应用程序需要知道其最终用户是否是成人。一种选择是提取该用户的出生日期，基于当前的日期，开发人员可以编写代码来计算此人是否是成年人。但这违反了*最少知识原则*，因为知道一个人的出生日期比知道他或她是否是成人更加敏感。另一个例子是核实某人社会保险编号的后 4 位数。同样，支持这一切的理念（即使从最终用户的角度看）是校验员（无论是真实的人还是软件应用程序）不必知道完整的社会保险编号就可确认此最终用户的身份。对于开发人员而言，最简单的实现方法是提取整个社会保险编号，并检查其后 4 位数是否正确。同样，这会将超过所需的信息暴露给应用程序。

“由于新的法规和合规性要求不断出台，数据保护已不只是保护数据源那么简单”。

可使用最少信息方法解决上述情况，通过仅获取必要的信息来尽量减少开发人员和应用程序所需的信息量。只需告诉应用程序用户是否是成人（例如派生属性或声明），或让提供方接受社会保险编号的后 4 位并返回真或假（如验证服务）。

要解决这些类型的隐私问题在今天看来是极为困难的，因为大多数情况下身份提供方是内部身份存储，本身不具备必要的能力来支持这些情况。

身份中心 — 缺少的环节

身份中心充当企业和联盟情景下的应用程序和各种身份属性权威来源间的经纪人 — 提供身份数据给应用程序。在讨论上述这些用例中，我们已经确定一些关键要求，以便身份中心能够取得成功。

部署提取

一个良好定义的身份中心应该为不了解实际部署环境的开发人员提供必要的提取。

“一个良好定义的身份中心应该为不了解实际部署环境的开发人员提供必要的提取”。

- 后端不可知 — 与一个或多个身份存储的集成和握手在身份中心自身内部处理并被屏蔽，使开发人员无法看到。此外，身份中心还应能够跨环境中的多个身份存储进行协商，并为身份提供单一的内聚视图。这包括能够聚合跨多个存储分布的身份，并聚集驻留在不同信息库中的不同身份。
- 协议不可知 — 开发人员现在通过单一的编程接口来处理单个虚拟身份中心 — 不再需要开发人员处理多个协议，提升了客户部署环境的可扩展能力和定制能力。
- 模式不可知 — 开发人员现在处理受身份中心支持的单一模式。通过这一增加的身份中心层，应该能够为不同的应用程序和垂直行业定义不同的模式配置文件，根据它们的特定需求进行量身定制。然后，可以在属性级别定义合适的映射，从而使开发人员的工作更简单。

安全特性

增加的身份中心层还应提供额外的安全特性，以更好地保护身份数据。

- 额外的访问控制 — 身份中心可以为内部身份存储中不合适或不可能实现的过滤和访问控制提供额外的过滤和访问控制。关键的要求是使安全机制与信息请求方的上下文相一致，以支持应用程序-用户-上下文和其他委托情况。
- 身份生命周期 — 对身份数据的任何生命周期要求都可以在身份中心内进行定义。现在可以根据身份中心定义的一组集中策略，实施本地身份数据的适当处理。
- 最少信息 — 身份中心可以扩展其模式来提供更复杂的属性映射，以支持这些情况。例如，可以为“是成人”的情况构建布尔响应（谓词），无需返回最终用户的出生日期。另外，“是成人”意味着在一些国家是大于 18 岁，而在另外一些国家是大于 21 岁。这些策略现驻留在身份中心而不是应用逻辑本身中 — 从而支持更轻松的自定义、统一的实施和法规遵守。

“身份中心可以为内部身份存储中不合适或不可能实现的过滤和访问控制提供额外的过滤和访问控制”。

身份中心允许应用程序开发人员将处理身份数据方面的一些复杂性下推到堆栈——远离应用逻辑本身。此外，它为应用程序提供了预先声明其要求的能力——由身份中心来满足这些要求。通过省却提供方的工作，开发人员可以解放出来，将更多精力集中在应用逻辑上。但最重要的是，身份中心提供了相应的工具来处理更为复杂的与更容易实现的自定义、配置和扩展要求相关的身份数据。

通过 ORACLE Virtual Directory 实现身份中心

“为支持 LDAP 的应用程序实现身份中心的一个简单方法是通过 Oracle Virtual Directory”。

理想的身份中心应为用户数据提供一个单一的权威视图，并且通常是分散的环境，其中用户数据分散在多个信息库中，包括各种目录和数据库。为支持 LDAP 的应用程序实现身份中心的一个简单方法是通过 **Oracle Virtual Directory (OVD)**。利用 OVD 的虚拟技术可以统一多个目录，并可以对数据库或其他专用身份数据存储进行 LDAP 访问。OVD 支持适配器连接到大多数目录服务器，包括 Oracle Internet Directory、Microsoft Active Directory、Novell eDirectory、IBM Tivoli Directory Service 和 Sun Java System Directory Server 等。它还支持适配器连接到关系数据库，包括 Oracle、IBM 和 Microsoft 的数据库。

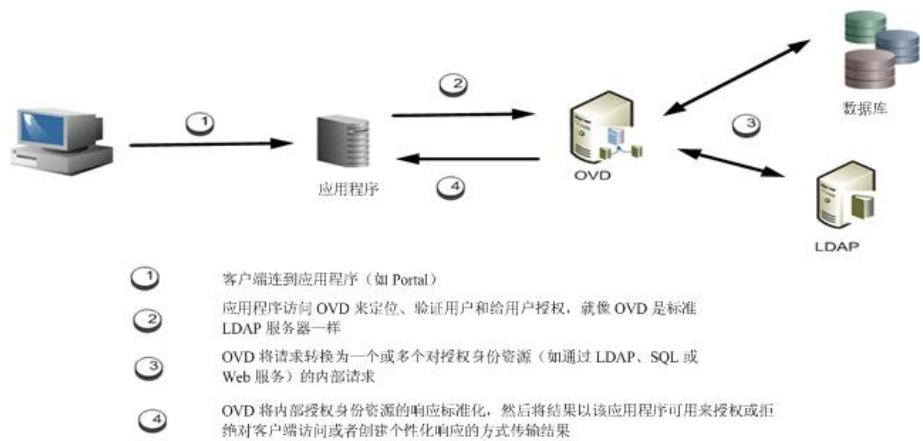
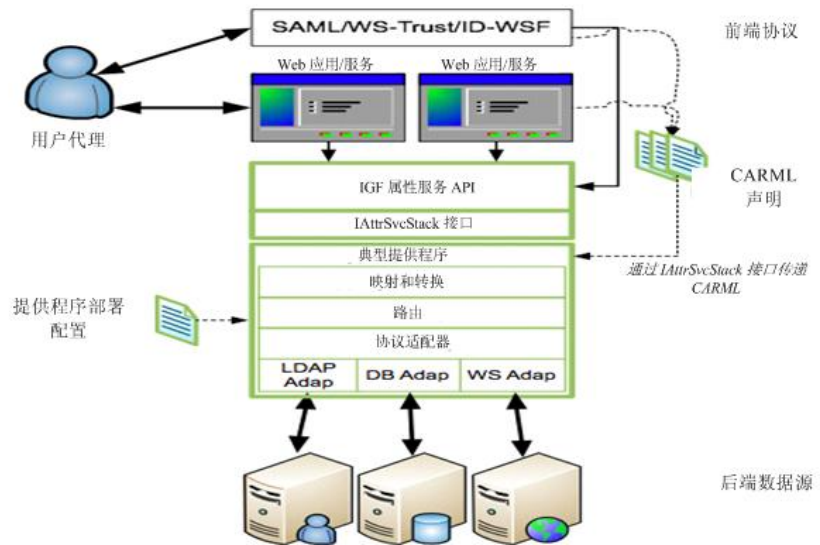


图 1. 典型的 OVD 部署

通过 OVD，开发人员可以省却统一身份数据的工作和后勤服务。应用程序开发人员现在可以通过 LDAPv3 接口（如，使用 JNDI）与此虚拟身份中心实现互动，并可获得个人身份记录的实时整合视图。

通过身份治理框架声明身份要求

身份治理框架 (IGF) 通过为“应用-身份中心”握手定义适当的开放标准来为身份中心解决方案提供多个关键组件。对于应用程序而言，IGF 的**客户端属性要求标记语言 (CARML)** 为设计者和开发人员提供了一种声明方式来传达他们的“身份要求”，而无需在应用程序本身内部建立该逻辑。采用 CARML 的一个重要目标是，支持对“身份数据”隐私限制的表达（用 WS-Policy 形式）。这提供了语言来定义安全要求，这些要求可根据特定应用要求量身定制。



“属性服务 API（亦称 Aris ID API）支持外化的身份服务，因为有了它，开发人员不必做出最好由部署管理人员和身份基础架构管理人员做出的决定”。

图 2. 高级别 IGF 体系结构

采用 OVD 方法，可以为当今支持 LDAP 的应用程序提供解决方案。但是，OVD 本身并不提供一个完全协议不可知的解决方案。Oracle 一直在开放源码方面居于领先地位，其 Aris 项目旨在开发一个允许开发人员使用外化身份数据的 API。有关详情请登录 www.openLiberty.org。

属性服务 API（亦称 **Aris ID API**）支持外化的身份服务，因为有了它，开发人员不必做出最好由部署管理人员和身份基础架构管理人员做出的决定。

API 实现 IGF（身份治理框架）标准，并支持非耦合开发，这样开发人员能够在 IDE 环境中全面测试身份服务的使用情况，再不会因不得不在开发时间部署复杂的基础架构而受到妨碍。此外，通过 Aris API 开发应用程序时，应用程序随时可充分使用外化数据，不论这些数据是位于使用 SAML、WS-* 的联合环境，还是在使用目录或数据库的内部企业服务中。

Aris 是第一个开发人员 API，用于实现 IGF 的 CARML 和隐私限制标准。通过在 CARML 的声明要求中定义安全机制，解除耦合允许属性服务 API 提供一个更简单的界面和使用模式。开发人员不再需要考虑实现方面的差异——避免了复杂的协议处理代码和配置要求（如 JNDI）。另外，借助图片中的 CARML，可采用后端不可知的方式定义额外的访问控制和隐私限制，以便基于特定应用的要求进一步保护身份数据。

“借助图片中的 CARML，可采用后端不可知的方式定义额外的访问控制和隐私限制，以便基于特定应用的要求进一步保护身份数据”。

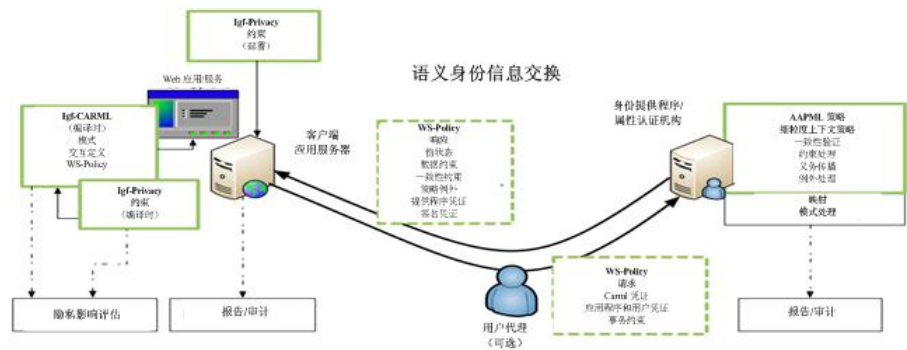


图 3. IGF 中的数据交换

通过 Aris API 支持的应用程序将有机会基于开源标准接口选择多个属性服务“提供方”技术。例如，Oracle 客户可以使用嵌入的 Oracle Virtual Directory 技术或者使用 Oracle 竞争对手或开源社区提供的其他实现技术进行部署。（注：为实现此目的，Oracle 目前正与 Higgins 社区合作以适配 Higgins IdAS）。

有关身份治理框架的更多信息，请访问：

http://www.projectliberty.org/liberty/resource_center/specifications/igf_1_0_specs

结论

确定身份数据的来源是任何识别身份的应用程序设计的一个重要部分。通过合适的身份中心，开发人员可以从应用程序和后端身份存储间的传统集成所固有的许多麻烦中解脱出来。由于围绕身份数据提供方的问题大多与部署时间有关，因此解除耦合可将身份中心置于合适的位置，以解决这些不再需要开发人员插手的问题。

通过允许应用程序以声明方式向身份中心公开要求，一个新的范例从身份中心显露出来 — 由提供方满足这些要求。这样一来，身份中心能够提供模式映射、访问控制和隐私支持方面的特性。而采用传统方法则很难在应用程序级别实现这些特性。最终的结果是，开发的应用程序可以采用更有效、更简洁和更安全的方式来处理和操作身份数据。

有关 **Oracle** 安全技术的详细信息，
请访问 <http://www.oracle.com/security>

甲骨文（中国）软件系统有限公司

北京总部

地址：北京市朝阳区建国门外大街1号，国贸大厦2座2208室
邮编：100004
电话：(86.10) 6535-6688
传真：(86.10) 6505-7505

北京上地6号办公室

地址：北京市海淀区上地信息产业基地，上地西路8号，
上地六号大厦D座702室
邮编：100085
电话：(86.10) 8278-7300
传真：(86.10) 8278-7373

上海分公司

地址：上海市卢湾区湖滨路222号，企业天地商业中心1号楼16层
邮编：200021
电话：(86.21) 2302-3000
传真：(86.21) 6340-6055

广州分公司

地址：广州市天河区北路233号，中信广场53楼5301&5308室
邮编：510613
电话：(86.20) 8513-2000
传真：(86.20) 3877-1026

成都分公司

地址：成都市人民南路二段18号，四川川信大厦20层A&D座
邮编：610016
电话：(86.28) 8619-7200
传真：(86.28) 8619-9573

大连分公司

地址：大连软件园东路23号，大连软件园国际信息中心2号楼
五层502号A区
邮编：116023
电话：(86.411) 8465-6000
传真：(86.411) 8465-6499

济南分公司

地址：济南市泺源大街150号，中信广场11层1113单元
邮编：250011
电话：(86.531) 8518-1122
传真：(86.531) 8518-1133

甲骨文软件研究开发中心（北京）有限公司

地址：北京市海淀区中关村软件园孵化器2号楼A座一层
邮编：100094
电话：(86.10) 8278-6000
传真：(86.10) 8282-6455

甲骨文研究开发中心（深圳）有限公司

地址：深圳市南山区高新南一道飞亚达大厦16层
邮编：518057
电话：(86.755) 8396-5000
传真：(86.755) 8601-3837

沈阳分公司

地址：沈阳市沈河区青年大街219号，华新国际大厦17层D单元
邮编：110016
电话：(86.24) 2396 1175
传真：(86.24) 2396 1033

南京分公司

地址：南京市玄武区洪武北路55号，置地广场19层1911室
邮编：210028
电话：(86.25) 8476-5228
传真：(86.25) 8476-5226

杭州分公司

地址：杭州市西湖区杭大路15号，嘉华国际商务中心702室
邮编：310007
电话：(86.571) 8717-5300
传真：(86.571) 8717-5299

西安分公司

地址：西安市高新区科技二路72号，零壹广场主楼1401室
邮编：710075
电话：(86.29) 8833-9800
传真：(86.29) 8833-9829

福州分公司

地址：福州市五四路158号，环球广场1601室
邮编：350003
电话：(86.591) 8801-0338
传真：(86.591) 8801-0330

重庆分公司

地址：重庆市渝中区邹容路68号，大都会商厦1611室
邮编：400010
电话：(86.23) 6370-8898
传真：(86.23) 6370-8700

深圳分公司

地址：深圳市南山区高新南一道飞亚达大厦16层
邮编：518057
电话：(86.755) 8396-5000
传真：(86.755) 8601-3837

甲骨文亚洲研发中心（上海）

地址：上海市杨浦区淞沪路290号，创智天地10号楼512-516单元
邮编：200433
电话：86-21-6095 2500
传真：86-21-6095 2555



开发人员和身份服务 — 通过身份中心处理身份数据

2008 年 9 月

作者: Stephen Lee

合作作者: Phil Hunt、Nishant Kaushik

公司网址: <http://www.oracle.com> (英文)

中文网址: <http://www.oracle.com/cn> (简体中文)

销售中心: 800-810-0161

售后服务热线: 800-810-0366

培训服务热线: 800-810-9931

欢迎访问:

<http://www.oracle.com> (英文)

<http://www.oracle.com/cn> (简体中文)

版权©2008归Oracle公司所有。未经允许，不得以任何形式和手段复制和使用。

本文的宗旨只是提供相关信息，其内容如有变动，恕不另行通知。Oracle公司对本文内容的准确性不提供任何保证，也不做任何口头或法律形式的其他保证或条件，包括关于适销性或符合特定用途的所有默示保证和条件。本公司特别声明对本文档不承担任何义务，而且本文档也不能构成任何直接或间接的合同责任。未经Oracle公司事先书面许可，严禁将此文档为了任何目的，以任何形式或手段(无论是电子的还是机械的)进行复制或传播。

Oracle是Oracle公司和/或其分公司的注册商标。其他名字均可能是各相应公司的商标。