|  |  |
| --- | --- |
| Handle系统概述 |  |

英文原文：[https://www.rfc-editor.org/rfc/pdfrfc/rfc3650.txt.pdf](https://www.rfc-editor.org/rfc/pdfrfc/rfc3652.txt.pdf)

版权声明

版权所有(C)互联网协会(2003)。 保留所有权利。

IESG说明

IETF和IRTF中的几个小组讨论了Handle系统及其与现有标识符系统的关系。IESG希望指出，这些讨论并没有导致IETF在所描述的Handle系统上达成一致，也没有导致它如何适合用于标识符的IETF体系结构。虽然已经讨论过将Handle作为URI的一种形式，特别是作为URN，但是这些文档描述了名称空间和标识符在互联网上如何工作的另一种观点，并包含了可能与IETF共识视图不匹配的现有系统的特征。

摘要

本文档提供了Handle系统的名称空间和服务体系结构的概述，以及它与其他互联网服务(如DNS、LDAP/X.500)的关系。Handle系统是一种通用的全局名称服务，它允许在互联网等网络上进行安全的名称解析和管理。Handle系统管理Handle，Handle是数字对象和其他互联网资源的惟一名称。

目录

* 1. 介绍。。。。。。。。。。。。。。。。。。。。。。。。。2
  2. 动机。。。。。。。。。。。。。。。。。。。。。。。。。。6
  3. Handle名称空间中。。。。。。。。。。。。。。。。。。。。7
  4. Handle系统架构……………8
  5. Handle系统安全。。。。。。。。。。。。。。。。。。。。11
  6. Handle系统和其他互联网服务……12个

6.1。域名服务(DNS)。。。。。。。。。。。。。。。。13

6.2。目录服务(X.500 / LDAP)。。。。。。。。。。。。。13

6.3。统一资源标识符(URI)/统一资源名称(URN)………………………14

* 1. 安全方面的考虑。。。。。。。。。。。。。。。。。15   
     7.1。一般安全惯例…………15

7.2。隐私保护。。。。。。。。。。。。。。。。。。。16

7.3。缓存和代理服务器……16

7.4。镜像。。。。。。。。。。。。。。。。。。。。。。。。17

7.5。拒绝服务(DoS)。。。。。。。。。。。。。。。。。17

* 1. Handle系统的历史………18
  2. 感谢。。。。。。。。。。。。。。。。。。。。18
  3. 参考文献和参考书目…………19
  4. 作者的地址。。。。。。。。。。。。。。。。。。。20.
  5. 完整的版权声明。。。。。。。。。。。。。。。。21

# 介绍

本文档提供了Handle系统的概述，Handle系统是一种分布式信息系统，旨在为诸如互联网之类的网络提供高效、可扩展和安全的全局名称服务。Handle系统包括开放协议、名称空间和协议的参考实现。该协议使分布式计算机系统能够存储数字资源的名称或Handle，并将这些Handle解析为定位、访问和以其他方式使用资源所需的信息。可以根据需要更改这些关联值，以反映标识的资源的当前状态，而无需更改Handle。这允许条目的名称在位置和其他当前状态信息更改期间保持不变。每个Handle可以有自己的管理员，可以在分布式环境中进行管理。Handle系统支持安全Handle解析。按客户机请求提供数据机密性、数据完整性和不可抵赖性等安全服务。

Handle系统提供联合名称服务，允许任何现有的本地名称空间通过获得惟一的Handle系统命名权威（naming authority）来加入全局Handle名称空间。本地名称及其值绑定在加入Handle系统后保持不变。可以通过一个遵照Handle系统协议的接口服务处理对本地名称空间的任何Handle请求。与惟一的命名权威相结合，任何本地名称在全局Handle名称空间下都是惟一的。

现在有几种服务用于为互联网资源提供名称服务。其中，使用最广泛的是域名系统(DNS)[2,3]。DNS的设计目的是“提供一种命名资源的机制，使名称可以映射到IP地址，并在不同的主机、网络、协议族、互联网和管理组织中使用”。互联网的发展对DNS的各种扩展提出了要求。也有人尝试将DNS用作通用资源命名系统。但是，由于DNS在基本网络路由中的重要性，在实现任何DNS扩展或重载用于通用资源命名的DNS时都要非常谨慎。反对将DNS用作通用命名服务的另一个因素是DNS管理模型。DNS名称通常由网络管理员在DNS区域级别进行管理。没有为每个名称提供管理结构，除了网络管理员之外，任何人都不能创建或管理DNS名称。这适用于域名管理，但不适用于通用资源命名。

Handle系统从一开始就被设计成一种通用的命名服务。它的目的是容纳大量的实体，并允许在公共互联网上进行分布式管理。Handle系统数据模型允许在与给定Handle关联的每个数据值的级别上定义访问控制。每个Handle可以进一步定义独立于网络或主机管理员的自己的管理员集。

传统的URL(统一资源定位器)[4]允许将某些互联网资源命名为DNS名称和本地名称的组合。本地名称可以是本地文件路径，也可以是对某些本地服务的引用(例如，cgi-bin脚本)。DNS名称和本地名称的组合为命名和管理单个互联网资源提供了一个灵活的管理模型。然而，URL实践也有一些关键的限制。大多数URL模式(例如，http)都是为解析而定义的。任何URL管理都必须在本地主机上完成，或者通过NFS等其他网络服务完成。使用URL作为名称通常将互联网资源与其当前网络位置联系起来。例如，当文件路径是URL的一部分时，URL将被绑定到它的本地文件路径。当资源出于某种原因从一个位置移动到另一个位置时，URL会破坏。当位置改变的原因是资产所有权的改变时这个工作尤其困难，因为所有权通常反映在域名上。

Handle系统旨在克服这些限制并增加重要的功能。具体而言，Handle系统的设计目标如下:

* 惟一性:Handle系统中的每个Handle都是全局惟一的。
* 持久性:Handle可以用作互联网资源的持久标识符。Handle不必派生自它所命名的实体。为了方便起见，Handle中可以包含现有的名称，甚至可以包含助记符，但是Handle和它所命名的实体之间的惟一操作连接是在Handle系统中维护的。当然，这并不保证持久性，持久性是管理注意的一个功能。但它确实允许相同的名称在位置、所有权和其他州条件的变化期间保持不变。例如，当命名资源从一个位置移动到另一个位置时，可以通过更新Handle系统中的值来反映新的位置，从而保持Handle的有效性。
* 多个实例:一个Handle可以引用一个资源的多个实例，这些实例位于网络中的不同位置，并且可能会发生变化。应用程序可以利用这一点来提高性能和可靠性。例如，网络服务可以用一个Handle为其服务定义多个入口点，以便分配服务负载。
* 多个属性:一个Handle可以引用一个资源的多个属性，包括相关的服务，这些服务可以通过任何方法在不同且可能改变的网络位置上使用。因此，Handle可以用作与已标识的资源相关联的不断发展的服务世界的持久入口点。
* 可扩展的命名空间:现有的本地命名空间可以通过获得唯一的Handle命名权威来加入Handle命名空间。这允许将局部名称空间引入全局上下文中，同时避免与现有名称空间发生冲突。使用命名权威还允许将服务(包括解析和管理)委托给本地Handle服务。
* 国际支持:Handle命名空间基于Unicode 3.0[17]，它包含了目前世界上使用的大多数字符。这允许在任何本机环境中使用Handle。Handle协议强制使用UTF-8[5]作为Handle的编码。
* 分布式服务模型:Handle系统定义了一个分层的服务模型，这样任何本地Handle名称空间都可以由相应的本地Handle服务、全局服务或两者同时提供服务。全局服务(称为全局Handle注册表GHR)可用于将任何Handle服务请求分派给负责的本地Handle服务。分布式服务模型允许将任何给定的服务复制到多个服务站点，每个服务站点可以进一步将其服务分布到单个服务器集群中。(注意，这里的本地只涉及名称空间和管理问题。一个本地Handle服务实际上可以有许多分布在互联网上的服务站点。)
* 安全名称服务:Handle系统允许在公共互联网上进行安全名称解析和管理。Handle系统协议定义了客户机和服务器身份验证以及服务授权的标准机制。它还提供了确保数据完整性和机密性的安全选项。
* 分布式管理服务:每个Handle可以定义自己的管理员或管理员组。每个Handle的所有权根据其管理员或管理员组定义。这与Handle系统身份验证协议相结合，允许在任何网络位置上的任何Handle由其管理员在公共网络上安全地管理。
* 高效解析服务:Handle协议被设计成允许高效的名称解析性能。为了避免解析受到计算成本高的管理服务的影响，Handle名称解析和管理的各自服务接口(即服务器进程和通信端口)可以由任何Handle服务定义。

本文档提供了Handle名称空间和服务体系结构的概述。它还将Handle系统与其他现有的互联网服务、协议和规范(例如，DNS[2,3]、url[4]、X.500/LDAP[6,7,8]和URN[9,10])进行比较。Handle系统数据和服务模型及其通信协议的详细信息在单独的文档中指定。他们可以在Handle系统网站http://www.handle.net找到。

# 动机

因为在互联网社区中有许多与名称相关的项目，所以有必要准确地定义我们认为Handle系统适合的位置。不幸的是，这一点特别困难，因为其他的主要命名方案要么采用抽象服务方法(例如URI/URN)，要么采用名称解析方法，而没有一个独立的框架来可靠地分布式管理底层数据库(例如DNS)。这使得对Handle系统进行分类变得困难。

Handle系统跨越边界。如果将其视为名称解析系统，则可以将其与DNS进行比较。如果用于实现URI/URN名称空间，它可以与任何URI/URN方案一起使用。如果用于分布式信息更新和管理，则可以将其视为分布式数据库系统的简化版本。

最好将Handle系统视为具有特定协议的名称-属性绑定服务，用于安全创建、更新、维护和访问分布式数据库。它的设计目的是在公共互联网等网络上实现安全信息和资源共享。Handle系统的应用可以包括用于数字出版物的元数据服务、用于虚拟身份的身份管理服务，或者需要解析和/或管理全局唯一标识符的任何其他应用。

基于探索的精神，Handle系统被设计为具有高性能的名称解析，并推动分布式访问控制和管理的边界。与大多数传统系统(甚至是分布式系统)的设计目标是拥有相对较少的具有广泛授权的管理员不同，Handle系统允许极其细粒度的管理控制。它有一个独特的自包含的管理框架，该框架将每个Handle的所有权与系统管理员分离，并允许为每个Handle值定义访问控制。

应该指出，与所有实际系统一样，Handle系统是一些技术和实际问题之间的折衷。在IETF中，对于Handle系统与其他现有互联网名称服务的关系，也有不同的看法。我们编写这个RFC的目标是向更广泛的社区展示概念、方法、特定的决策、权衡和结果。

# Handle名称空间

每个Handle由两部分组成:它的命名权威(也称为前缀)和命名权威下唯一的本地名(也称为后缀):

<Handle>::= < Handle命名权威> "/" < Handle本地名>

命名权威和本地名称由ASCII字符“/”分隔。命名权威下的本地名称集合定义该命名权威的本地Handle命名空间。任何本地名称在其本地名称空间下都必须是唯一的。命名权威和该机构下的本地名称的唯一性确保了在Handle系统上下文中任何Handle都是全局惟一的。

例如，“10.1045/january99-bearman”是发表在D-Lib杂志[12]上的一篇文章的Handle。它的命名权威是“10.1045”，它的本地名字是“january99-bearman”。Handle名称空间可以被认为是许多本地名称空间的超集，每个本地名称空间在Handle系统下都有唯一的命名权威。命名权威标识了关联Handle创建时的管理单元，尽管不一定要继续管理。保证每个命名权威在Handle系统中是全局惟一的。任何现有的本地名称空间都可以通过获得唯一的命名权威来加入全局Handle名称空间，这样名称空间下的任何本地名称都可以作为命名权威和上面所示的本地名称的组合全局引用。

Handle系统下的命名权威以类似于树结构的分层方式定义。树的每个节点和叶子都有一个对应于命名权威段的标签。父节点通知父节点（译者：应该指爷节点）其子节点的命名权威。与DNS不同，Handle命名权威是由左向右构造的，将树的根到表示命名权威的节点之间的标签连接起来。每个标签被ASCII字符“.” (0x2E)分隔。例如，国会图书馆（“loc”）的国家数字图书馆项目(“ndlp”)的命名权威被定义为“loc.ndlp”

每个命名权威下面可能注册了许多子命名权威。**任何子命名权威只能在其父命名权威已注册后由其父命名权威注册**。但是，父命名权威和子命名权威表示的名称空间之间没有内在的管理关系。父名称空间及其子名称空间可以由不同的Handle服务提供，它们可以共享也可以不共享任何管理特权。

Handle可以由ISO/IEC 10646通用字符集(UCS-2)中的任何可打印字符组成，该字符集是由Unicode v3.0[17]定义的。UCS-2字符集包含了当今所有主要语言中使用的大多数字符。为了允许与大多数现有系统兼容，并防止不同编码之间的歧义，Handle系统协议要求将UTF-8作为Handle使用的唯一编码。UTF-8编码保留任何ASCII编码的名称，以便在不引起命名冲突的情况下最大限度地与现有系统兼容。[13]讨论了全局名称空间的一些编码问题和UTF-8编码的选择。

默认情况下，Handle是区分大小写的。但是，任何单独的Handle服务都可以定义其名称空间，使该名称空间下的任何Handle中的ASCII字符不区分大小写。

# Handle系统架构

Handle系统定义了一个层次服务模型。顶层包含一个Handle服务，称为全局Handle注册表(GHR)。较低层由所有其他Handle服务组成，通常称为本地Handle服务(LHS)。

全局Handle注册表可用于管理任何Handle名称空间。它在Handle服务中是惟一的，因为它提供用于管理命名权威的服务，所有这些权威都作为Handle进行管理。命名权威Handle提供客户机可以用来访问和利用本地Handle服务的信息，用于命名权威下的Handle。

本地Handle服务由在某些命名权威下对Handle负有管理责任的组织承载。本地Handle服务可以负责任意数量的本地Handle名称空间，每个名称空间由唯一的命名权威标识。本地Handle服务及其负责的本地Handle名称空间集必须在全局Handle注册表注册。

Handle系统的一个重要方面是它的分布式体系结构。整个Handle系统由许多单独的Handle服务组成。每个服务都可能包含一个或多个服务站点。就Handle解析而言，每个服务站点是服务中所有其他站点的完全复制。每个服务站点可能由一个或多个Handle服务器组成。指向给定服务站点的所有Handle，以及因此而产生的所有Handle请求都将均匀地分布在这些Handle服务器上。Handle系统作为一个整体可以由任意数量的Handle服务组成。对于Handle服务的数量或组成每个服务的站点的数量没有设计限制，对于组成每个站点的服务器的数量也没有任何限制。任何服务站点之间的复制并不要求每个站点包含相同数量的服务器。换句话说，虽然每个站点都有相同的一组Handle，但是每个站点可以跨不同数量的服务器分配这组Handle。这种分布式方法旨在帮助可伸缩性，适应任何大规模的操作，并减轻单点故障的问题。

图3.1演示了一个潜在的Handle服务，它由两个服务站点组成:一个位于美国东海岸，另一个位于美国西海岸。东海岸服务站点由四台服务器计算机组成。部署了更强大的计算机的西海岸服务站点认为两台服务器就足够了。任何Handle服务的服务站点数量，以及任何服务站点使用的服务器数量，都可以根据服务需求动态地添加或删除。

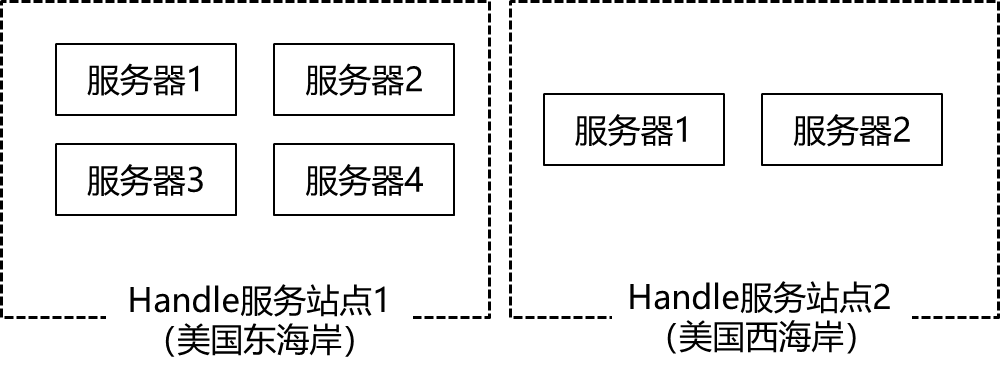


图3.1:Handle使用两个服务站点配置的服务

每个Handle服务在Handle系统下管理一个不同的子名称空间。不同Handle服务下的名称空间可能不会重叠。子命名空间通常由许多命名权威下的Handle组成。Handle服务被称为这些命名权威的“home”服务，并且是唯一为这些命名权威下的Handle提供解析和管理服务的服务。在解析Handle之前，客户机必须在请求中确定Handle的“home”服务。每个Handle的“home”服务是其命名权威的“home”服务，并在全局Handle注册表中注册。客户机可以通过查询全局Handle注册表中的命名权威Handle来找到每个Handle的“home”服务。

全局Handle注册表维护命名权威Handle。每个命名权威Handle维护描述命名权威的“home”服务的服务信息。服务信息列出了给定的Handle服务的服务站点，以及每个站点中每个Handle服务器的接口。要查找任何Handle的“home”服务，客户机可以查询全局Handle注册表，以查找与相应的命名权威Handle关联的服务信息。服务信息为客户机与“home”服务通信提供了必要的信息。

图3.2显示了典型Handle解析过程的一个示例。在本例中，“home”服务是一个本地Handle服务。客户机正在尝试解析Handle“10.1045/july95-arms”，并且必须从全局Handle注册表中找到它的“home”服务。“home”服务可以通过向全局Handle注册表发送一个查询来找到“10.1045”的命名权威Handle。全局Handle注册表返回负责命名权威“10.1045”下Handle的本地Handle服务的服务信息。服务信息允许客户机与本地Handle服务通信，以解析Handle“10.1045/july95-arms”。

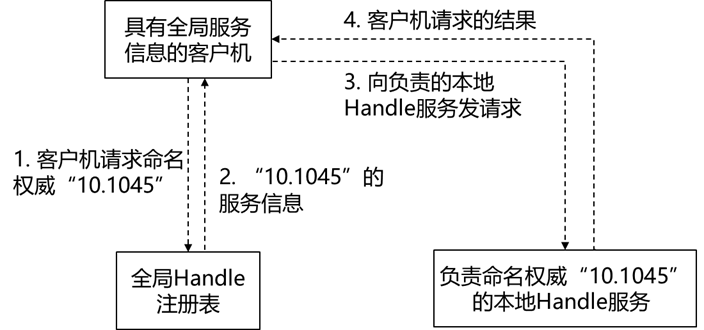


图3.2:从全局开始的Handle解析

为了提高解析性能，任何客户机都可以选择缓存从全局Handle注册表返回的服务信息，并将其用于后续查询。单独的Handle缓存服务器(独立的或作为通用缓存机制的一部分)也可以用于在本地社区中提供共享缓存。给定一个缓存的解析结果，可以在本地回答相同Handle的后续查询，而不需要联系任何Handle服务。给定缓存的服务信息，客户机可以直接将其请求发送到正确的本地Handle服务，而无需与全局Handle注册表联系。

# Handle系统安全

Handle系统通过公共互联网等网络提供Handle解析和管理服务。每个Handle可以分配一组值。客户机使用Handle解析服务将任何Handle解析为其值集。每个值都有一个数据类型和一个惟一的值索引。客户机可以根据数据类型或值索引查询特定的Handle值。

Handle管理服务响应来自客户机管理Handle的请求。这些请求包括添加Handle、删除Handle或更新它们的值。它还通过命名权威Handle管理命名权威。每个Handle可以有自己的管理员，每个管理员可以被授予一组特定的权限。Handle系统身份验证协议在完成任何管理请求之前对Handle管理员进行身份验证。

Handle系统提供诸如客户机和服务器身份验证、数据机密性和完整性以及不可抵赖性等安全服务。默认情况下，Handle解析不需要任何客户机身份验证。但是，对于分配给任何Handle(由其管理员)的机密数据的解析请求，以及任何管理请求(例如，添加或删除Handle值)，都需要客户机进行身份验证，以便进行适当的授权。在授权过程中，服务器将决定客户机是否具有访问这些机密Handle值的权限，或者是否具有添加或更新Handle和Handle值的权限。当需要身份验证时，Handle服务器将在执行客户机请求之前向请求客户机发出挑战。为了满足身份验证要求，客户机必须发回正确的响应，并将自己标识为合格的管理员。Handle服务器只有在客户机身份验证成功后才会响应初始请求。Handle客户机可以选择使用密钥或公钥密码进行身份验证。Handle系统认证也可以通过第三方认证服务进行。为了确保数据完整性，客户机可以从任何Handle服务器请求数字签名的响应。它们还可以与Handle服务器建立安全的通信会话，以便使用会话密钥对任何交换的信息进行加密(用于数据保密性)。Handle服务器还可以通过在将Handle数据发送给客户机之前对其进行加密来提供机密性。

Handle系统为客户机和服务器之间的安全信息交换提供服务选项。当然，这并不能保证Handle值的真实性。管理员分配给任何Handle的不正确的值很可能误导客户机。另一方面，Handle值可能包含对其他Handle值的引用，以提供额外的凭据。例如，Handle值R(例如，一个声明)可能包含对其他Handle值的引用，而该Handle值包含（来自一个可信源）对R的数字签名。信任签名的客户机就可以信任Handle值R。

# Handle系统和其他互联网服务

有许多现有的和提议的互联网标识服务或规范，通过设计或意图，涵盖了为Handle系统提议的一些功能。本节简要回顾它们与Handle系统的关系。

## 6.1 域名服务(DNS)

域名服务(DNS)的最初设计主要用于将域名映射到IP地址以实现网络路由。RFC 1034[2]和RFC 1035[3]提供了其设计和实现的详细说明。互联网的发展增加了对DNS的各种扩展的需求，甚至可能将其用作通用资源命名系统。然而，任何这样的使用都有可能减慢网络地址转换和/或影响其在网络路由中的有效性。当大量数据与任何特定的DNS名称关联时，DNS实现通常不能很好地扩展。因此，通常认为将DNS用作通用命名服务是不合适的。

反对将DNS用作通用命名服务的另一个因素是DNS管理模型。DNS名称通常由网络管理员在DNS区域级别进行管理。没有为每个名称提供管理结构。除了网络管理员之外，没有为任何人提供创建或管理DNS名称的工具。这适用于域名管理，但不适用于通用名称管理。

Handle系统在分布式管理和服务模式上与DNS不同，在安全特性上也有所不同。Handle系统协议包括安全选项，以确保数据传输期间的机密性和完整性。每个Handle可以有自己的管理员，独立于服务器管理员。Handle系统协议允许任何Handle管理员在公共网络上安全地管理他或她的Handle。此外，Handle系统服务模型允许任何服务站点在服务器集群中动态配置服务分发，以适应增加的服务请求。这也允许不太强大的计算机一起使用来支持任意数量的Handle。

## 6.2目录服务(X.500 / LDAP)

X.500[6]是由ISO和ITU定义的OSI目录标准。它的设计目的是“提供一个白页服务，返回电话号码或X.400 O/R地址的人”，并“主要涉及提供名称服务器服务的开放系统互连(OSI)应用程序”[7]。X.500使用一组协议定义层次数据和信息模型，以允许全局名称查找和搜索。然而，事实证明，该协议很难实现，而且很难将“客户机访问集成到现有产品中”[14]。LDAP(轻量级目录访问协议)[8]通过使协议更简单、更容易实现，克服了许多这些困难。但是，仍然存在一些问题，即随着LDAP从本地目录访问协议(LDAP v2)发展到分布式服务协议(LDAP v3)，它面临许多在其原始设计中没有解决的问题，从而导致新的复杂性。

名称解析服务(如Handle系统)和目录服务(如LDAP)之间的根本区别在于搜索功能。能够搜索目录服务的附加功能必然会增加复杂性，从而影响其效率。一个纯名称服务，例如Handle系统，可以只围绕对已知条目的有效解析进行设计，而不根据不完整的标准发现未知条目所需的函数和数据结构。

目录服务，如LDAP或whois++[15,16]，可以与Handle系统一起使用，以提供反向查找服务。例如，现有的公司目录服务可以为这两个服务提供接口。Handle系统接口将提供高效的名称解析服务。目录服务接口将提供扩展的搜索功能。在LDAP服务引用中也可以使用Handle。例如，可以将LDAP服务作为Handle引用。这样做将使引用持续化，与位置更改无关。

## 6.3统一资源标识符(URI)/统一资源名称(URN)

统一资源标识符(URI)[23]定义了一种统一的、可扩展的命名机制，用于在web应用程序中标识互联网资源。统一资源名称(URN)[11]是URI的子集，它为URI下的持久名称空间定义了名称空间注册机制。URI/URN表示web应用程序中使用的大部分互联网名称服务。本节讨论Handle系统与URI/URN的关系，以及应用程序如何在URI/URN上下文中使用Handle系统。

Handle系统为互联网提供了一个通用的名称服务。与DNS或X.500目录服务一样，Handle系统在任何URI/URN名称空间之外定义其名称空间。Handle可以直接转录和解析，不需要任何URI/URN模式作为前缀。例如，一个库应用程序可以将Handle“10.1045/july95-arms”直接解析为它的一组Handle值。在这种情况下不需要URI/URN模式。

Handle系统可用于需要持久名称服务的应用程序。Handle系统提供了必要的机制，允许将持久名称注册为Handle。可以定义特定的命名权威来承载那些设计为持久的Handle。但是，Handle的持久性更多地取决于管理策略，而不是技术本身。这些策略超出了Handle系统服务的范围，如本文档集所述。

另一方面，Handle系统也可以用于不需要持久名称的应用程序。这样的Handle可能只有很短的生命周期，它们也可能用于在不同的时间标识不同的对象。

可以使用Handle系统作为底层名称服务来开发不同的web应用程序。每个应用程序都可以定义自己的URI/URN名称空间以满足其应用程序的需要。例如，应用程序FOO可能注册了一个URI命名空间“FOO:”来标识web上的任何FOO服务。同时，应用程序栏可以注册URN名称空间“URN:BAR”来标识任何需要持久名称的BAR对象。FOO和BAR应用程序都可以使用Handle(在各自的命名权威下)来命名和解析服务和/或对象。这类似于DNS，在不同的应用程序中定义了不同的URI模式(例如，“telnet”、“ftp”、“mailto”等)，所有这些都使用DNS服务。

IETF和IRTF讨论了URI相关工作领域中的Handle系统。对于Handle系统是否适合特定的URI或URN名称空间，有不同的看法。还需要考虑Handle系统与互联网上其他现有名称服务的关系。这种讨论超出了本文件的范围。

# 安全注意事项

本节旨在告知人们Handle系统的安全限制，以及应用程序开发人员、服务提供者和Handle系统客户机应该采取的预防措施。Handle系统协议[21]及其数据和服务模型[22]的具体安全注意事项在单独的文档中讨论。

## 7.1一般安全实践

Handle系统的安全性在事务的每一步都取决于客户机和服务器主机的安全性。它假设客户机主机没有被篡改，并且客户机软件将可靠地将接收到的数据传递给客户机。任何Handle服务的客户机还必须假设所涉及的任何Handle服务器都没有被破坏。信任全局Handle注册表将正确地将客户机请求定向到负责任的本地Handle服务。信任本地Handle服务就是相信本地Handle服务将正确地返回由其管理员分配给该Handle的数据。本地Handle服务通常支持一组命名权威。因此，信任本地Handle服务就意味着信任那些命名权威。

Handle系统的完整性在很大程度上取决于全局服务信息的完整性。无效的全局服务信息可能会误导客户机提供不适当的本地Handle服务。它还允许攻击者伪造服务器签名。全局Handle注册表必须非常小心地保护全局服务信息和用于签名全局服务信息的公钥对。客户机应用程序应该只接受全局Handle注册表中的全局服务信息。他们应该在每次更新时检查其完整性。

出于效率考虑，除非客户机特别要求，否则handle服务器不会为每个服务响应生成或返回数字签名。为了确保数据的完整性，客户机必须显式地要求服务器返回数字签名。为了保护敏感数据不被暴露，客户机可以与服务器建立一个通信会话，并要求服务器使用会话密钥加密任何数据。

## 7.2隐私保护

默认情况下，存储在Handle系统中的大多数Handle数据都是可公开访问的，除非Handle管理员另行指定。Handle管理员在添加包含私有信息的Handle值时必须注意。他们可以选择标记这些只有Handle管理员可读的Handle值，或者将它们存储为加密的Handle值，以便这些值只能在受控的受众中读取。

Handle服务器生成的日志文件是客户机隐私可能受到攻击的另一个脆弱点。Handle服务器的操作员必须小心保护这些信息。

## 7.3缓存和代理服务器

除了性能提升和其他增值服务之外，代理服务器和缓存服务器都表现为中间人，因此很容易受到中间人攻击。重要的是要知道代理和缓存服务器不是任何Handle服务的一部分。他们是Handle系统的客户机。代理和缓存服务器的服务响应不能通过Handle系统协议进行身份验证。客户机与其直接代理/缓存服务器之间的信任必须独立设置，而不考虑处于通信路径中间的代理/缓存服务器的数量。

通过使用代理和缓存服务器，客户机假定服务器将提交它们的请求并转发来自Handle系统的任何响应，而不会错误地处理任何内容。他们还假定服务器将代表他们保护任何敏感信息。

代理和缓存服务器操作人员应该保护这些服务器所在的系统，就像保护包含或传输敏感信息的任何系统一样。特别是，从代理服务器收集的日志信息通常包含高度敏感的个人信息和/或组织信息。这些资料应小心加以保护，并制订和遵守使用这些资料的适当准则。

缓存服务器提供了额外的潜在漏洞，因为缓存的内容是恶意攻击的目标。对缓存的潜在攻击可以为Handle用户显示私有数据，或者在用户认为他们已经从网络中删除后仍然保留的信息。因此，缓存内容应该作为敏感信息进行保护。

## 7.4.镜像

Handle系统客户机应注意镜像站点之间的内容复制可能出现的延迟。对于任何时间敏感的数据，他们应该考虑将请求发送到主服务站点。服务管理员必须仔细选择镜像站点。为了确保数据完整性，每个镜像站点必须遵循相同的安全过程。可以使用软件工具来确保镜像站点之间的数据一致性。

## 7.5拒绝服务(DoS)

与任何公共服务一样，Handle系统也会受到拒绝服务攻击。在当今的技术中，没有通用的解决方案可以保护我们免受此类攻击。可以开发服务器实现来识别这种攻击，并在攻击发生时通知管理员。无状态cookie[19, 20]是一种减轻DoS攻击对执行身份验证、完整性和加密服务的主机的影响的方法。此外，服务器实现需要可升级以利用新的安全技术，包括可用的拒绝服务（anti-DoS）技术。

# Handle系统的历史

Handle系统最初是在CNRI作为整体数字对象架构的一部分构思和开发的。1994年秋天，在戴维·埃利(David Ely)的努力下，CNRI创建了第一个公共实施项目。整个数字对象架构，包括Handle系统，在1995年由Robert Kahn和Robert Wilensky[1]在一篇论文中描述。发展在CNRI继续作为计算机科学技术报告(CSTR)项目的一部分，由美国国防部高级项目局(DARPA)资助，资助号为MDA-972-92-J-1029和MDA-972-99-1-0018。这个早期数字图书馆项目的一个方面是为数字图书馆的底层基础设施开发一个框架，这也是网络计算机科学技术参考图书馆(NCSTRL)[18]和相关活动发展的一个主要因素。

Handle系统的早期使用者包括国会图书馆、国防技术信息中心(DTIC)和国际DOI基金会(IDF)。来自这些组织以及NCSTRL、其他数字图书馆项目和前面提到的相关IETF工作的反馈都对Handle系统的发展做出了贡献。客户机和服务器的当前状态和可用软件可以在http://www.handle.net找到。

# 感谢

这项工作源自Handle系统实现的早期版本。设计思想基于Handle系统开发团队中的讨论，包括David Ely、Charles Orth、Allison Yu、Sean Reilly、Jane Euler、Catherine Rey、Stephanie Nguyen、Jason Petrone和Helen She。我们感谢他们对这项工作的贡献。

作者还感谢Russ Housley (housley@vigilsec.com)、Ted Hardie (hardie@qualcomm.com)和Mark Baugher (mbaugher@cisco.com)的广泛评论，以及来自IETF/IRTF社区其他成员的建议。

# 引用和参考书目

* 1. 卡恩，R.和R.威伦斯基，“分布式数字对象服务的框架”，D-Lib杂志，1995。
  2. Mockapetris, P。，“域名-概念和设施”，std13, RFC 1034, 1987年11月。
  3. Mockapetris, P。，“域名-实施和规范”，std13, RFC 1035, 1987年11月。
  4. berners - lee, T。， Masinter, L.和M. McCahill，“统一资源定位器(URL)”， RFC 1738, 1994年12月。
  5. Yergeau F。，“UTF-8, Unicode及iso10646的转换格式”，RFC 2044, 1996年10月。
  6. 国际电信联盟第X.500号出版物，“目录:概念、模型和服务概览”，1993年。
  7. 查德威克，“瞭解X.500 -目录”，查普曼与霍尔ISBN: 0-412-43020-7。
  8. Wahl, M。， Howes, T.和S. Kille，“轻量级目录访问协议(v3)”， RFC 2251, 1997年12月。
  9. “统一资源名称的功能要求”，RFC 1737, 1994年12月。
  10. Sollins, K。“统一资源名称解析的体系结构原理”，RFC 2276, 1998年1月。
  11. IETF统一资源名称工作组，1998年4月。
  12. D-Lib杂志,http://www.dlib.org
  13. 孙树松，“Handle系统的国际化-一个持续的全球名称服务”，第十二次国际统一码会议，1998年4月。
  14. D.古德曼，C.罗宾斯，《了解LDAP和X.500》，1997年8月。
  15. 多伊奇P。,Schoultz R。，“WHOIS++服务的体系结构”，RFC 1835, 1995年8月。
  16. Weider C。，“Whois++索引服务的架构”，RFC 1913, 1996年2月。
  17. 统一码联盟，“统一码标准，v3.0版本”，Addison-Wesley Pub Co;ISBN: 0201616335。
  18. 网络计算机科学技术报告图书馆(NCSTRL)， http://www.ncstrl.org/
  19. 辛普逊，“资讯系统:会话密钥管理协议”，台湾科技大学出版社，1999年3月。
  20. Harkins, D.和D. Carrel， "The 互联网 Key Exchange (IKE)"， RFC 2409, 1998年11月。
  21. 太阳,S。，“Handle系统名称空间和服务定义”，RFC 3651, 2003年11月。
  22. 太阳,S。赖利,S。，“Handle系统协议(ver 2.1)规范”，RFC 3652, 2003年11月。
  23. berners - lee, T。， "统一资源标识符(URI):通用语法"，RFC 2396, 1998年8月。

# 作者的地址

山姆x太阳

国家研究计划公司(CNRI)

1895年，普雷斯顿怀特医生，100号套房

雷斯顿,弗吉尼亚州20191

电话:703-262-5316

电子邮件:ssun@cnri.reston.va.us

拉里Lannom

国家研究计划公司(CNRI)

1895年，普雷斯顿怀特医生，100号套房

雷斯顿,弗吉尼亚州20191

电话:703-620-8990

电子邮件:llannom@cnri.reston.va.us

布莱恩·伯施

国家研究计划公司(CNRI)

1895年，普雷斯顿怀特医生，100号套房

雷斯顿,弗吉尼亚州20191

电话:703-262-5316

电子邮件:bboesch@cnri.reston.va.us

# 完整的版权声明

版权所有(C)互联网协会(2003)。 保留所有权利。

这个文档和翻译的可能被复制和家具,和衍生著作,评论或者解释或者协助其实现可能准备,复制,发布和分发,在全部或部分,没有任何形式的限制,前提是上面的版权声明和本段包括在所有此类副本和衍生著作。然而,这个文档本身可能不会以任何方式修改,如通过删除版权通知或引用互联网协会或其他网络组织,除所需的目的为版权开发互联网标准在这种情况下,程序中定义的过程必须遵循网络标准,或根据需要翻译成除英语之外的其他语言。

上述所授予的有限权限是永久的，互联网协会及其继任者或受让人不会撤销上述权限。

这个文档,此处包含的信息是“是”的基础上提供的互联网协会和互联网工程任务组并不保证,明示或暗示,包括但不限于任何保证,使用文中的信息不会侵犯任何权利或任何隐含保证适销性或健身为特定目的。

译者：王伟兵