The Corporation for National Research Initiatives

Handles and DNS Persistence

IDCC/W3C Workshop on Domain Name Persistence

Sean Reilly
sreilly@cnri.reston.va.us

Hi, I'm Sean Reilly. I am a developer for CNRI, working on multiple projects including the Handle System (at handle.net) which is a distributed, scalable and secure identifier resolution system.

Introduction to Handles

Handles are persistent identifiers for Digital Objects. They were conceived by Bob Kahn, who along with Vint Cerf are considered the “fathers of the internet” for creating TCP/IP. Handles are a primary component of what he calls the Digital Object Architecture. The Digital Object Architecture is basically a system for managing digital information as distinct, clearly identified units in order to index, find and interact with that information. The Handle System provides the identifier service for the architecture. The other two components are the DO Repository, which provides access to digital objects, and the DO Registry which provides an index and searching or discovery mechanism.

Handle System

- Protocol
- Software
- Name Service

The Handle System is comprised of:
- a protocol with an open specification (RFC 3652) that is secure, scalable, and optimized for a relatively flat namespace
- software: open source client, server, resolver, and admin tools
- name service (thousands of servers worldwide, containing billions of identifiers) with root service managed by CNRI and partners in Germany, China, the US, and elsewhere.

4263537/4086

4263537/4086

4263537/4086

4263537/4086

The Corporation for National Research Initiatives

hdl:4263537/4086

The Corporation for National Research Initiatives

http://hdl.handle.net/4263537/4086

resolveHandle("4263537/4086")

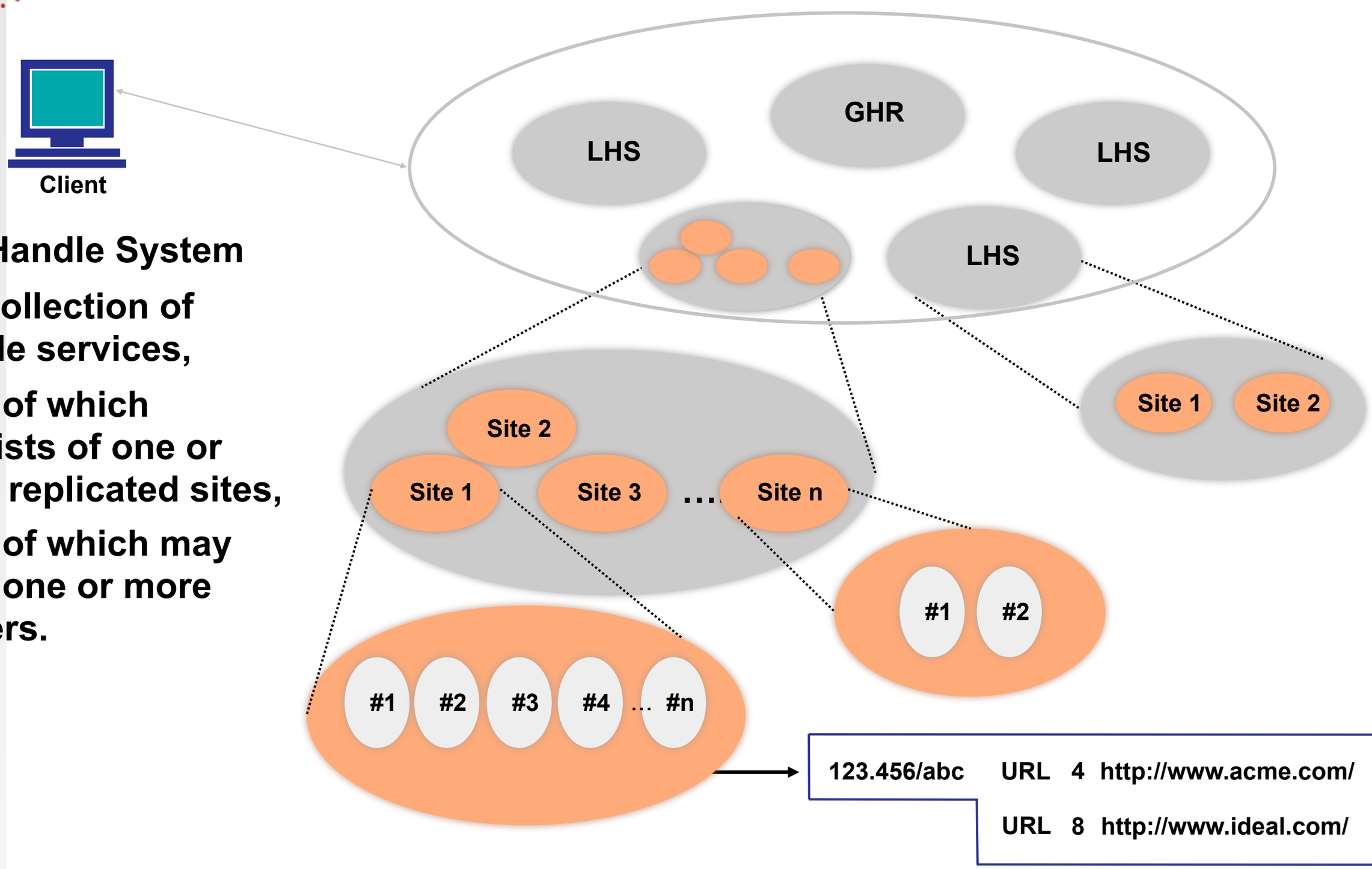
Handle Service Structure

The Handle System

is a collection of handle services,

each of which consists of one or more replicated sites,

each of which may have one or more servers.



How do Handles encourage and enable persistence?

Numeric Prefixes

top level namespaces/prefixes are all non-semantic numbers

Permanent Prefix Allocation

Prefix allocation is permanent. Once you are given a prefix, it will never be put back into the pool of available prefixes and given to someone else.

Limited Delegation

Single level of namespace delegation which tends to discourage embedding hierarchy in names. Hierarchies are often based on real-life structures such as businesses, governments, countries, classification systems. These things change and make for bad namespace components.

The Handle System protocol and software actually does support hierarchical namespaces, but it is currently limited by policy.

Secure Resolution Service

http://something ≠ https://something

Secure resolution. Resolution security is important, but more important for persistence is that the security is not embedded in the identifier itself. This is unlike HTTP which requires you to change the identifier from http to https if you wish for it to be accessed securely. DNSSEC now allows for secure resolution of some DNS names, similar to what Handles have had for more than a decade, but it needs to be incorporated in protocols and software in order to solve the http vs https problem.

How are Handles Resolved?

- Handle-aware Clients
- Browser Plugin
- HTTP proxy: http://hdl.handle.net/4263537/4086

Whoa, Hold on There

Ultra-Persistent Handles depend upon DNS?

Not exactly...

Handle System Doesn't Use DNS

the system itself doesn't depend upon DNS in any way. hdl.handle.net is an interface to access the system and resolve handles from standard web browsers.

However, the handle proxy services are the most visible point of access.

One Domain Gets All the Attention

hdl.handle.net is a single domain: we can extend a lot of effort into making sure that it is always available. Kindof like how all Google services (for each country) go through one google.com domain name. Our effort is focused on providing 100% uptime for that one domain.

Exceptional Domain Names

http://geographiclocations.org/?lat=x&lon=y
→ Google Maps
→ MapQuest
→ Car Navigator

http://hdl.handle.net/12345/abcde
→ Handle Resolver
→ Get Metadata

hdl.handle.net is a commonly known special domain. The opinion of many members of the IETF URI mailing list (including W3C representatives) was that URIs that don't map cleanly to HTTP (such as geographic location references) should use a "special domain" that would trigger special handling by clients that recognize that domain. This way, references to geographic locations would invoke whatever service was appropriate, depending upon the client's preferences and capabilities.

This lets domain names exist independently of the DNS system. This is already happening with browser add-ons like the CNRI Handle Resolver extension for FireFox. Other people are also doing this, for example price comparison shopping browser plugins that key off of Amazon.com and other URLs.

DNS Persistence Issues

Domain names are only part of the picture

If you have a domain name, you're only halfway towards getting what you need. To do anything useful you also need a protocol, a port number, probably a path to a file on that machine.

Depending upon the protocol you may need a third party to verify the identity of the host using a certificate authority. The case sensitivity of the file system on the server may play a factor. When using a domain in an identifier there are many pieces, not just the domain name itself, that can change.

Should an identifier be atomic?

With Handle and DOI, you have a complete, atomic identifier. How you resolve it or reference it can change, but the identifier itself never needs to be parsed or understood by applications.

Should an identifier actually exist?

http://www.w3.org/Provider/Style/URI.html
vs
http://www.w3.org/Provider/Style/URI.html?new_url_same_resource
vs
URI.Provider.Style.w3.org

One benefit (and some might say a hassle) of handles is that you actually *create* a Handle. It exists in a database and can be changed to reference different URLs, have metadata hung off of it, etc. With a DNS-based HTTP URL, it “exists” if you can HTTP GET something from it (and sometimes even if you can’t). What if a DNS name (as opposed to a full URL) could be used to refer to specific resources, instead of just the host part of a URL? Then you could create and manage DNS-based identifiers in a more formal way.

Hierarchy is Harmful

Any domain is dependent on all of your parent domains continuing to exist, not changing, and referring to your subdomain. Each additional domain name segment is a potential liability. Especially when those domains can contain a company, department, country, or location name. RFC 1123 says domain name segments cannot be all numeric, so making “pure” persistent domains is not as simple as allowing numeric top- (or even second-) level domains.

Any persistent DNS-based naming system should try to compress or avoid hierarchies

Top-Level Domain for Persistence?

Helpful, but not a complete solution

A dedicated top-level domain with additional persistence-enhancing policies would help with the hierarchy issue. But a reliable hostname is only part of what is needed to make a good URL-based identifier.

Workshop Question

What are the negative consequences of domain name impermanence?

- Not much!
- Domains already do their job really well. There are other better methods for providing permanence:
  - Special Domains
  - URI Schemes

There is much more to a persistent identifier than the domain name, so making the DNS part more permanent would only be a partial solution.

Workshop Question

What action can be taken to forestall or ameliorate those consequences?

- A top-level domain intended for persistent namespaces would be an improvement if done properly, but it's not a silver bullet

There is much more to a persistent identifier than the domain name, so making the DNS part more permanent would only be a partial solution.

The Corporation for National Research Initiatives

Handles and DNS Persistence

IDCC/W3C Workshop on Domain Name Persistence

Sean Reilly
sreilly@cnri.reston.va.us