Network Working Group
Internet-Draft
Intended status: Informational
Expires: 22 August 2025

P. M. Hallam-Baker
ThresholdSecrets.com

18 February 2025

DNS Account Handles, A Whitepaper
draft-hallambaker-any-01

Abstract

A proposal for use of DNS names to support universal account identifiers ’handles’ is described. Once registered, a handle may be used for authentication to any network service, to initiate communication with the holder or as the basis for IoT device management.

This document is a whitepaper proposing the general approach. A strawman prototype supporting single account Web login across multiple sites, onboarding of IoT devices and end to end secure messaging, file-drop, voice and video and has been built using existing and proposed IETF work.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 22 August 2025.

Copyright Notice

Table of Contents

1. Introduction

The naming of devices, hosts and services has been a core concern in network protocol design from the earliest days of computer networking. In the Internet architecture, the primary means by which services and devices are identified to users is by DNS name.

While DNS has proved remarkably successful within its design scope, attempts to provide users with universally recognized identifiers have proved less successful. Most Web sites require users to create a separate account with a username and password for authenticated access.

Being required to remember they have created an account with a site at all represents an unacceptable imposition on the user, let alone remembering the username. The notion that users should be expected to create and remember a strong, unique password for every site they might with to interact with is a colossal absurdity no matter how many times 'experts' intone their sage security wisdom.

OAUTH2 allows a user with an account held at one service to access resources at another and OpenID connect purports to provide a form of portable account login across the Web. But in practice this only works for users with accounts at a small and shrinking number of providers and introduces serious business concerns for relying parties facing competition from the monopolist providers.

The ATprotocol [atproto], proposed by BlueSky allows users to use any DNS name they control as an account identifier.

This whitepaper shows how the ATprotocol scheme may be used as a general-purpose Web account login scheme and proposes modifications to the approach to make the use of DNS names as authentication handles entirely independent of the services that rely on them. The use of DNS handles is then extended to support all the personal identity services an Internet user might require including initiating communications with other users and provisioning IoT devices with network names and credentials.

1.1. Handle Types

Naming is a central concern in network architecture because different forms of identifiers offer different properties. The use of DNS names as identifiers emerged from the use of the 'hosts.txt' file to provide stable, user-facing names in place of the IP addresses used at the network layer.

Three types of handle are defined:

Direct Trust Fingerprints [@mbqc-7oha-rnba-frdl-r4gi-yqha-dl36]

: An immutable, globally unique identifier derived from a public key that serves as a root of trust for validating assertions related to the thing identified.

DNS Name Handles [@alice.example.com]

: A public DNS name that serves as a unique user-facing identifier. For example, Alice might register the name example.com and issue herself the handle @alice.example.com. As with all DNS names, DNS name handles are mutable, particularly when the holder of the handle does not control the underlying DNS names.

  To allow DNS Name Handles to be distinguished from Personal Names, a DNS Name Handle MUST have at least two labels.

Personal Names [@doctor]

: A private name that only has interpretation for the individual user. For example, Alice might user the personal name @doctor to refer to her personal physician and @bank to refer to her banker, changing the binding of these names as necessary.

  Personal Names typically have a single label but personal names MAY be defined with two or more labels as a means of overriding the public interpretation of a DNS Name Handle.

This approach does not guarantee that the interpretation of a DNS Name handle will be immutable but does ensure that it will only change when the controller of the underlying DNS registration intends.

Similarly, the binding of a personal name is entirely under the control of the user defining it. Thus if Bob exchanges contact details with Alice when she is using the handle @alice.example.com, he can assign Alice the personal name @alice to ensure that whenever he interacts through that handle, he will be interacting with the Alice he expects.

1.2. Personas

DNS Handles provide a means of establishing a common persona across diverse Internet resources. Users seeking to partition their Internet use MAY establish multiple independent personas with separate DNS handles.

Likewise, users MAY link multiple DNS handles to a single persona.

1.3. DNS Handle Provider

The infrastructure described in this document anticipates the provision of a suite of Internet service to permit use of the handle in different contexts:

DNS Authoritative Service To publish records corresponding to the handle under management.

OAUTH2 Authentication Service To manage authentication requests under the handle.

Device Provisioning Service To relay provisioning requests for network names and credentials to the relevant DNS and CA services.

Contact Catalog Manager To manage the user’s contact catalog and synchronize it across their devices.

Presence Service To broker device to device connections when establishing synchronous communication (chat, voice, video)

While the services proposed as the means of delivering the last three of these services are based on the Mathematical Mesh platform, these services could be provided by any platform(s) offering similar functionality.

It is anticipated that a range of providers will offer services under a variety of business models including free at point of delivery and subscription based. For the purposes of this document, there are two important differences:

Personal Registration The DNS name is registered by the user of the

handle.

At-will Registration The DNS name is registered under a DNS name held by the service provider who allows the user temporary use of the handle that can in principle be revoked at any time.

At-Will registrations subordinate the user's autonomy to that of the service provider but allow service providers to offer precisely the same functionality as personal registrations without charging a registration fee.

Personal registration offers the name holder the ability to change service providers at any time without switching costs.

1.4. @nywhere Sign In

Use of DNS handles allow social media platforms to offer users a common identity across independent platforms. Alice can identity herself as @alice.example.com at a microblogging site, a photo sharing site and commenting on Web forums and other user's personal blogs. Other users can recognize these contributions as all being from the same person.

Interoperability is achieved through the use of a common identity, it is not necessary for the platforms to be members of a common information exchange federation or run any common protocol except for those used to translate DNS handles into direct trust fingerprints (DNS) and to perform authentication (OAUTH2).

Once it is established that a user will use a DNS handle as a common identity for authentication purposes, it is natural for them to expect use of the same handle for other purposes. If Bob knows that Alice has the DNS handle @alice.example.com, it is natural for him to expect to find a personal site about Alice at https://alice.example.com/ and to be able to use the same identifier to contact Alice through email and other modes of communication.

1.4.1. Authorization Policy

A common identification infrastructure facilitates specification of authorization policy. If Alice has blocked Mallet from reading her posts on a microblogging site, it is likely she would want to block Mallet from commenting on her personal blog as well and she is almost certainly not going to be willing to accept emails or other messaging communications from him.

1.4.2. Subscription Services

A common identification infrastructure facilitates implementation of traditional paid-content models and permits the creation of new ones. For example, Alice might subscribe to a curated feed of international news with links to articles that would normally require a separate subscription fee.

1.4.3. Privacy protection

Traditional approaches to privacy protection have focused on making it difficult for third parties to aggregate information across sites with mixed success. While each individual DNS handle is a globally unique identifier whose very purpose is to link activity across different sites, there is no limit to the number of DNS handles a user can use and with appropriate client support, ’disposable’ handles may be registered as needed.

1.5. @nyone Communications

When the Internet first emerged as a public infrastructure in the mid 1990s, person-to-person communication was limited to emails and a rudimentary form of chat. Today, users communicate through myriad platforms offering diverse combinations of synchronous and asynchronous messaging, text, image, audio and video, almost every platform except for the venerable SMTP is a walled garden requiring a separate account and applications.

DNS handles provide a basis for establishing person-to-person and person-to-group communications by any modality using a common identifier controlled by the user.

1.5.1. Authorization Policy

Abuse has become a severe problem in both email and telephone communications. A common identity allows users to adopt a ’default deny’ approach to accepting inbound communications from unknown users, with optional exceptions for specific modes (e.g. contact request) and for those on chosen shared lists.

For example, Alice might choose to only allow voice or video communication requests from people she has exchanged contact information but allow anyone who is on a list of family members, her golf club or her business club to send her email messages.

1.5.2. End-to-End Trust

Support for end-to-end encryption has been widely regarded as an essential feature for new personal communications services for over a decade. But while applying encryption to the communication channel is straightforward, almost none provide end-to-end trust and those that do typically support it in a form no real user could possibly be expected to make use of.

All that is required to intercept end-to-end encrypted communications is the ability to perform an active on path attack and to convince the parties to use keys of the attacker's choosing. Almost every provider of such services has the ability to perform such an attack and by extension, so does any party that is able to coerce the service provider.

Use of a common identifier bound to a Direct Trust Fingerprint allows the user to take full control of their communications security. Once Alice and Bob have exchanged contact information, their communications will be authenticated and encrypted under the immutable Direct Trust Fingerprint stored in the contacts catalogs under their personal control.

1.6. @nything Device Configuration

When the Internet was originally built, computers were large, expensive devices, typically shared among a large number of users. Today, a typical home contains tens or hundreds of computers, an increasing proportion of which require local network or Internet connectivity for optimal functionality.

Provisioning consumer devices in the 'Internet of Things' with the names and credentials required for network functionality has been regarded as a hard problem for many years. Once it is decided that each Internet user is going to have their own DNS name, the solution to these provisioning difficulties becomes obvious: The devices will use the DNS for naming and PKIX credentials binding public keys to those DNS names to establish secure communications.

Thus, if Alice buys a coffee pot and gives it the name 'coffee', the relevant configurations should be made to the local and public DNS and a WebPKI certificate acquired to access the device as https://coffee.alice.example.com/

All the component protocols required to support this configuration after a device has been onboarded to the user’s control exist already. Multiple options exist for supporting onboarding of devices. The only thing lacking is a service to accept network naming and credentialling requests from devices and forward the relevant configuration updates to the services selected by the user.

2. Definitions

This section presents the related specifications and standard, the terms that are used as terms of art within the documents and the terms used as requirements language.

2.1. Requirements Language

This is an informational document and does not contain any normative language.

2.2. Defined Terms

ACME

At-Will Registration

Certificate Authority

Cross Certificate

DID

Direct Trust Fingerprints

DNS Authoritative

DNS Name Handle

DNS Primary

Handle

Intermediate Certificate

OAuth Client

OAuth Resource Server

Personal Name

Personal Registration

Private Certificate Root

WebPKI

2.3. Related Specifications

DNS handles represent a 'glue' technology that mostly serve to connect existing technologies and infrastructures. According, the number of related specifications is unusually large.

2.3.1. @nywhere

The following documents are relevant to the use of DNS handles to provide a ubiquitous login infrastructure:

The OAuth 2.1 Authorization Framework [draft-ietf-oauth-v2-1] Describes the OAuth authentication framework.

Proof Key for Code Exchange [RFC7636] Describes the PKCE extension used to authenticate OAUTH requests.

OAuth Client ID Metadata Document [draft-parecki-oauth-client-id-metadata-document] A JSON document describing an OAUTH 'client'.

OAuth 2.0 Protected Resource Metadata [draft-ietf-oauth-resource-metadata] A JSON document describing an OAUTH resource.

Decentralized Identifiers (DIDs) v1.0 [did] Describes the reservation of a subdivision of the URI space for user identifiers.

DID PLC Method [plc] Describes the PLC identifier formed as the digest of a public signature key used to authenticated ATprotocol interactions.

Anywhere Sign-In [draft-hallambaker-anywhere] Describes the profile of the above documents used to implement @nywhere sign in and proposed extensions and improvements.

2.3.2. @nything

The following documents are relevant to the provisioning of network names and credentials:

Automatic Certificate Management Environment (ACME) [RFC8555] The issue of TLS certificates to devices

Dynamic Updates in the Domain Name System [RFC2136] Publication of DNS records to local and public DNS.

Mathematical Mesh 3.0 Part IV. Schema Reference [draft-hallambaker-mesh-schema].

The assertion format used to describe Mesh devices.

Anything Service [draft-hallambaker-anything] Describes a service that receives device requests for provisioning of network names and credentials and performs the necessary actions to satisfy authorized requests.

2.3.3. @nyone, @nytime

The following documents describe one approach top use of DNS handles for person-to-person and person-to-group communication:

JSContact: A JSON Representation of Contact Data [RFC9553]. Describe s the format used for contact data interchange.

JSCalendar: A JSON Representation of Calendar Data [RFC8984]. Descri bes the format used for calendar data interchange

Mathematical Mesh 3.0 Part II: Uniform Data Fingerprint [draft-hallambaker-mesh-udf]. Describes the UDF format used to express fingerprints of public keys and other data.

2.4. Implementation Status

Reference code under the MIT Open Source license has been developed to demonstrate all the features described in this document.

3. Architecture

To access an Internet service using a DNS handle or a personal name, an application requires three pieces of information:

* The DNS prefix for the service protocol (e.g. _http._tcp).

* The DNS name of the service.

* The Direct Trust Fingerprint of the user account.

The DNS name of the service and Direct Trust Fingerprint are obtained from a DNS handle or personal name through the Handle resolution process.

3.1. Handle Resolution

Handle resolution is performed in the following order with resolution by means of a Personal Contact Assertion having the highest precedence:

Personal Contact Assertion

DNS TXT

HTTP Well-Known

3.1.1. Personal Contact Assertion

Users MAY use the platform and technology of their choice for resolution of personal names. It is highly desirable but not essential for contacts to be synchronized across devices so that the user enjoys a consistent experience. But even here, personal requirements may come first. A developer might choose to use a separate personal name resolution platform for test purposes.

Mesh Contact Catalogs provide one means of expressing personal name bindings and synchronizing them between devices. Entries in the catalog contain the following information

A set of local names to which the contact is bound

A dictionary mapping DNS protocol identifiers (e.g. ’SMPT’) to addresses

Contact credentials (e.g. PKIX certificates)

[Optional] Source material(s) from which the contact data was obtained

[Optional] Update information

Personal names MAY be bound to any protocol including those that do not use DNS name handles. This allows an application to allow Bob to send an email to Alice using the name @alice and select the SMTP transport and address alice@example.com if that is the most appropriate protocol Alice allows Bob to use.

Similarly, personal contact assertions MAY be used to provide supplemental means of contacting a user known through their DNS handle.

3.1.2. DNS TXT, SRV

The primary means for resolving DNS Handles is through the DNS. Publication and resolution are performed according to the mechanisms described in [RFC6763], [RFC8552] and [RFC8553].

The ATprotocol specifies a single TXT record that binds a DNS handle to a DID. For example, the following record binds the handle @bsky.app to the DID did:plc:z72i7hdynmk6r22z27h6tvur:

```
atproto.bsky.app TXT did=did:plc:z72i7hdynmk6r22z27h6tvur
```

The ATprotocol resolution protocol defines a resolution mechanism that allows the DID to be resolved to determine the corresponding ATprotocol resource server from which the OAuth2 authentication service MAY be determined. In the case that we are using the DNS handle as a generic authentication mechanism, it is more convenient to be able to obtain the OAUTH service address directly from the DNS by means of an SRV record

```
_oauth._tcp.bsky.app SRV 1 1 443 bsky.social
```

The authorization service record MAY optionally publish a TXT record to specify the location of the authorization server metadata:

```
_oauth._tcp.bsky.social TXT \

  "meta=https://bsky.social/.well-known/oauth-authorization-server"
```

Since the TXT record merely specifies the default URI for locating the metadata for the OAUTH service, it does not add value in this particular case. But in the case that the DNS zone is authenticated by DNSSEC, the record might contain an additional entry to authenticate the metadata document by means of a fingerprint of the document and/or a self-authenticating resolution mechanism such as an EARL [draft-hallambaker-mesh-udf].

While manual management of such DNS records is likely to prove impractical, a mechanism for automation of the zone entries is described in this document.

The same approach is used to publish the User Profile Fingerprint and services used to support communication by Mesh messaging and presence.

In the case that the number of services and protocols supported becomes large, it is likely to prove useful to specify a 'directory' record to specify the set of protocols that a handle supports.

3.1.3. HTTP Well-Known

While the DNS is the natural mechanism to use for resolution of DNS names, some DNS users lack the ability to edit DNS entries directly but do have the ability to place content on a Web service published at the domain. In this case, the HTTP .well-known mechanism MAY be used as a substitute for DNS records.

For example, the handle bsky.app would be resolved by an GET request to https://bsky.app/.well-known/atproto-did as described in the ATprotocol specifications.

4. @nywhere Application Authentication

One of the challenges in describing the use of the OAuth service as an authentication service is that it was originally presented as an 'authorization' service and the nomenclature applied in the documentation is confusing in the extreme.

The use case we consider here is one in which user Alice is attempting to authenticate to a Web site 'example.net' by means of her DNS handle which is bound to an account managed by her authentication service provider, example.com. In OAuth2 terminology, the Web site Alice is attempting to access is the OAuth 'client' and the authentication service provider is an 'authorization service'.

The ATprotocol is essentially a profile of OAuth2 with extensions currently under development plus the DNS handle resolution convention and PLC DID scheme.

In order to accept @nywhere authentication, the OAuth client generates an authentication key (and optionally an encryption key) and publishes this and other information as its client metadata at a URI which is also used as the client identifier.

To authenticate to the Web Site example.net, Alice provides her DNS handle 'alice.example.com'. The site then performs the first stage of the authentication process as follows as an OAuth2 client:

Resolves the DNS handle to a DID by means of either a DNS TXT record or a HTTPS .well-known entry.

Resolves the DID using the PLC protocol to obtain the list of protocols supported. This specifies an ATprotocol resource server.

Fetches the metadata for the resource server, this specifies the authorization service.

Fetches the metadata for the authorization service.

Generates a Proof Key for Code Exchange (PKCE) token

Submits a Pushed Authorization Request to the Authorization service containing the PKCE verifier

Receives back the HTTP redirect to return to the user.

Returns the HTTP redirect to the user.

At this point, the user is redirected to the authentication service and is asked to perform whatever additional steps are required for authentication.

In most cases, the user will have already authenticated to the Authorization service allowing this step to be skipped. This is important because while the currently common practice of each Web site requiring users to respond to a second factor authentication challenge is obnoxious and absurd, allowing users the option of protecting their accounts with a single challenge they are only required to respond to once in their day is much more likely to gain user acceptance.

After the user completes the OAuth2 interaction, they are redirected to the Web site from which the authentication request originated and the OAuth2 client completes the authentication process as follows:

Submits the 'Authorization Request' authenticated by the PKCE nonce to the authorization service to obtain an initial token

Performs a 'code' query against the Authorization server to obtain a set of access tokens bound to a specific account DID. The client MUST verify that this DID is the same as the DID for the account for which authentication was requested.

5. Usability Improvements

The user experience would be improved through definition of a microformat convention allowing the DNS handle to be filled automatically. This would ideally provide an unambiguous signal to the Web client that a DNS handle is being requested for use with the @nywhere protocol so that it can substitute the OAuth2 authentication dance involving a separate authorization server with a direct public key-based authentication.

For example, we might stipulate that a HTML form requesting the user enter their DNS handle have the ’dns-handle’ autocomplete attribute:

```
<div>

<label for="anything">@</label>

<input type="text" id="anything" autocomplete="dns-handle" />

</div>
```

6. Second Factor and Public Key Authentication

As previously mentioned, the ability to access multiple Web sites through a single account and a credential authenticated by a single authentication service makes use of multiple factor authentication much less burdensome.

Besides the advantage that the user is only required to authenticate once per day rather than for each site used, the second factor scheme does not need to be limited those available to the third party sites. In practice, this limits the second factor authentication to email and SMS challenge/response schemes which offer marginal security advantage at best.

An authentication service chosen by the user can make use of a second factor authentication selected by the user. For example, responding to a biometric challenge on a personal mobile device. In this case, we can achieve two factor authentication without requiring the user to enter a password at all, the factors being something carried (the mobile device) and a biometric. Should a passcode be required to unlock the mobile device, it would provide a third factor (something known).

The Mathematical Mesh protocols include the ’confirmation protocol’, an advance on second factor authentication in which the user is told which action is being requested (e.g. enable @nywhere login for their browser) and asked to accept or reject the request.

In the ideal case, Web sites involving a high degree of risk such as access to a brokerage account would offer multiple levels of authentication with viewing account balances placing a minimal burden on the user and placing high value stock trades requiring individual confirmation using a trusted and trustworthy device.

7. @nyone Personal Communications

Users interacting with each other through social media are likely to want to interact in other ways, both online and offline. JSContact provides a JSON based format for expressing postal, geographic, telephone and Internet addresses associated with a persona.

While there are many contact formats in use on the Internet, including the venerable vCard format, JSContact is preferred because it offers a superset of the capabilities of all the formats in current widespread use and has much better provision for incorporating cryptographic credentials to enable end-to-end secure communications.

For example, Alice’s JSContact might be:

```
{
  "version": "1.0",
  "kind": "individual",
  "language": "en",
  "speakToAs": {
    "@type": "SpeakToAs",
    "grammaticalGender": "feminine",
    "pronouns": {
      "p1": {
        "@type": "Pronouns",
        "pronouns": "she/her"}}},
  "titles": {
    "t1": {
      "@type": "Title",
      "name": "Research Scientist",
      "kind": "title"}},
```

```
    "emails": {
      "e1": {
        "@type": "EmailAddress",
        "address": "jqpublic@xyz.example.com",
        "contexts": {
          "work": true}}},
    "@type": "Card",
    "created": "2025-02-18T18:35:30Z",
    "updated": "2025-02-18T18:35:30Z"}
```

The DNS prefix for @nytime is "_contact". The following attributes are supported:

uri URI linking to a JSContact record relating to the persona.

7.1. Cryptographic credentials

While JSContact provides means for incorporating any credential with an IANA content type assignment in a contact, the means by which the user states that particular keys are to be used for particular purposes is likely to require extensions to the underlying format.

For example, if Alice is a developer, she is likely to use OpenPGP keys for signing emails and source code repository commits.

As with any cryptographic credential, it is important that relying parties assure themselves that the credentials they are using are the credentials of the counterparty they expect. The mere fact that a JSContact record Bob has obtained from @alice.example.com contains a public encryption key for Alice is not absolute proof that the key belongs to the person Bob knows as ’Alice’:

* Alice might not be the holder of the @alice.example.com handle.
* The TXT record published by the authoritative DNS server for _contact.alice.example.com might have been compromised.
* The TXT record returned by Bob’s DNS resolver might have been corrupted.

* The JSContact record published at the specified location might have been corrupted.

* Alice might have compromised the security of her cryptographic application.

The use of credentials issued by a Certificate Authority provides an answer to most of these objections but introduces a Trusted Third Party as a potential point of failure.

DNSSEC provides an answer to some but not all of these objections but also introduces the operator of the DNS root and the domain(s) in which the handle is registered as Trusted Third Parties.

There are no perfect solutions to the problem of establishing Internet trust. But fortunately, most Internet uses do not require perfection. Experience of deployment of SSH has shown that 'Trust After First Use' is sufficient for most purposes provided that users store the credentials acquired in the initial contact exchange and alert the user when inconsistencies occur.

7.2. Encrypted Authenticated Resource Locators

Encrypted Authenticated Resource Locators (EARLs) are a type of Uniform Data Fingerprint (UDF) that allow a single, compact identifier to be used to locate, decrypt and authenticate a binary data object.

For example, the EARL for Alice's JSContact data shown above published on the HTTP server example.com is:

udf://example.com/application/jscontact+json:EGBP-N7ZI-V5NS-H5P3-VXCI-3PYJ-AOLA

The components of the URI are:

Scheme 'udf'

Site 'example.com'

IANA Media Type (optional, defaults to application/mmm-udf)

Key Seed 'EGBP-N7ZI-V5NS-H5P3-VXCI-3PYJ-AOLA'

The Key Seed is computed from the linked content by computing the SHAKE256 digest of the content plaintext, prefixing it with the byte 0x21, truncating the result to the desired degree of precision and rendering the result in Base32 with groups of four digits separated by hyphens.

The SHA-3-512 UDF fingerprint is then taken of the ASCII representation of the key seed to form the locator path. The locator path has precisely twice the number of bits precision as the key seed. Combining the site specified in the EARL with the well-known service specifier 'mmm-udf' provides the URL from which the linked ciphertext is published:

https://example.com/.well-known/mmm-udf/KBPB-MRBN-NFR2-AXYO-DWXU-MABV-BU2N-VGDP-NZQL-UYUY-VEHS-E2U6-2FWO-QCH

The content ciphertext is obtained by encrypting the plaintext content using AES-GCM using a key and initialization vector derived from the key seed using SHAKE256 with 352 bits of output and appending the 16-byte tag value to the result.

Key:

69 9B C9 22 A9 58 23 F5 09 23 26 DA 88 3C 7E 0C

31 C0 73 F7 FB 9F 09 94 A9 3A 84 FA C7 C1 D6 30

IV:

A5 CA F7 5A 27 B1 A7 FA 61 11 D1 2B

To resolve the EARL, the client fetches the ciphertext, decrypts it and verifies that the digest of the recovered plaintext matches the original key seed

Note that any ciphertext that decrypts to a plaintext matching the digest specification is acceptable.

7.3. Updating contacts

When users update their contact information, relying parties require some means of

* Detecting that the contact has updated

* Fetching the purported updated contact data

* Verifying that the purported updated contact data is from the same source as the original.

* Verifying that the purported contact data was published after the original.

While communication clients MAY simply poll the contact’s DNS Handle at regular intervals for updates, this approach might become unacceptably burdensome if large numbers of clients are constantly updating contacts.

A solution to this involving a JSON assertion structure, an append only Merkle log and a structured Bloom Filter has been developed.

8. @nytime Calendar

The use of JSContact records to publish contact information associated with a persona suggests the use of JSCalendar records to publish relevant events.

For example, a standards body holding three meetings a year at diverse locations around the world might publish a schedule of upcoming meetings.

The DNS prefix for @nytime is "_calendar". The following attributes are supported:

future URI linking to a JSCalendar record publishing future events related to the persona.

past URI linking to a JSCalendar record publishing past events related to the persona

9. @nyplace Personal Place

DNS Handles provide a means by which a user can visit diverse Web resources including social media using a common persona. @nyplace allows users to link a personal Web presence, their ’personal place’ to their DNS handle. The information provided and modes of social interaction afforded may be as sparse or expansive as the user chooses.

The DNS prefix for @nyplace is "_place". The following attribute is supported:

uri URI linking to the personal place associated with the handle.

10. @nything Device Configuration

As it stands today, the Internet of Things has fallen far short of its original promise. While being able to connect to a device deployed in the home or office through an Internet enabled mobile is useful, the functionality typically achieved is limited to that of a remote control that will fail if the cloud service supporting it becomes unavailable for any reason.

For the Internet of Things to deliver real value, devices must do more than communicate with the user, they must communicate with each other and do so securely and directly without the introduction of artificial intermediaries and for this to be possible, the thing that is on the Internet name requires an Internet DNS name and the necessary WebPKI credentials to communicate directly rather than through a vendor controlled portal.

If Alice (@alice.example.com) buys an Internet connected doorbell, she should be able to reach it directly using a standard Web browser configuration through a domain subordinate to her DNS handle or registered DNS name.

10.1. Requirement: Enabling Multiple Users

To change the temperature on my thermostat from my Web browser, I am required to navigate to the vendors Web site, guess that the icon in the top left corner is my profile, select ’my home’, perform an OpenID Connect login and interact with the devices supported by that particular vendor. In theory, my partner could also change the thermostat setting in the same way because I have added her to the account but she never does because it is easier to ask me, even in the case that I am in a meeting the other side of the planet.

While adding other family members to the account by specifying an email address with a link to a Web site with a six-hoop account verification process might appear perfectly reasonable to the designer, no member of my family other than myself has ever been sufficiently motivated to complete it. Thus a device that I bought to reduce effort for myself has substantially increased it.

Use of DNS handles provides an obvious advantage in authorizing additional users and temporary users by simply adding the user’s DNS handles to the list of authorized users. If the property was a short term let, renters could be automatically authorized to control all the IoT devices for the duration of their stay and the authorization automatically cancel when they leave.

10.2. Deployment Modes

Three deployment modes are anticipated:

Local only The @nything service runs in the local network only. This allows configuration of the network gateway (firewall/NAT) and provides continuity of service should the local network lose Internet connectivity but cannot host DNS services or provide reverse HTTP proxy functionality unless the network has a static IP address.

Cloud only The @nything service runs in the cloud only. This allows hosting of DNS services and provision reverse HTTP proxy functionality but not configuration of the network gateway (firewall/NAT). If Internet connectivity is lost, devices already connected will continue to function until their credentials expire but it will not be possible to onboard new devices.

Hybrid The @nything service is split between a local component and a cloud service. This provides all the advantages of local and cloud deployment.

10.3. Onboarding / Offboarding Protocol

The means by which user devices are registered to the user and provisioned with the necessary configuration data to perform their tasks is outside the scope of this document. While the Mathematical Mesh provides one approach to achieving this end, it is not necessary to fix on a single approach. For the purposes of @nything, all that is required to place a device under management is a means of provisioning it with:

* A name for the device from which its public DNS name is to be formed by applying a suffix.

* A public/private keypair unique to the device

* The IP Addresses to be used for local/public access

* An assertion verifiable by the @nything service stating the above.

In addition, the device will of course require some means of communicating with the @nything service during the onboarding process.

10.4. Configured Services

The @nything service receives credentialing and naming requests from devices and makes the relevant configurations as indicated by the device assertion.

Network Configuration

In the general case, IoT devices are deployed with only a network local address. If the device is to be accessed from the public Internet, either a reverse HTTP proxy or some form of NAT traversal will be required.

DNS Configuration

The A, AAAA, SRV and TXT records necessary to support host and service discovery are provisioned to local and public DNS as required. In addition, DNS records required to respond to an ACME DNS challenge are provisioned as required.

Device Credential Provisioning

If the device is to be public facing, it is provisioned with public credentials obtained from a WebPKI CA, (e.g. through ACME). Devices may also be provisioned with certificates under a private CA for purposes such as 802.11x where a long certificate lifetime is desirable.

Storage Configuration

IoT devices frequently require storage. This may be provided through a subscription service or another device belonging to the user.

11. Security Considerations

Privacy Considerations

A frequently voiced objection to the use of a single identity across multiple Web sites is that linking user activity across Web sites makes it easier for parties to link user activity across Web sites.

Since DNS Handles provide a mechanism for linking user activity across Internet properties, client applications MUST ensure that disclosure of the handle to a site is an overt act on behalf of the user and SHOULD provide mechanisms to allow the user to defeat unwanted linkage through creation of additional DNS Handles to separate identities.

For example, a browser offering a 'Private Browsing' mode SHOULD allow the user to specify which identity they are private browsing in, obtaining temporary DNS handles as required.

Switching Costs

DNS Handle Providers could encourage users to invest in an identity established under a DNS name under their control and make unanticipated demands for continued use once the name has been established.

Loss of DNS name

DNS names are effectively rented rather than owned and can be transferred to other parties without the permission of the name holder. This might occur as a result of a fraudulent name transfer, losing a challenge brought under the UDRP or simply forgetting to renew the subscription.

Concentration of Risk

Use of DNS handles potentially represents a concentration of risk in a single identifier should the user lose the use of it.

Impersonation

A social media site could falsely claim that a post was made under a handle that it did not verify so as to allow a defamation action to be brought against the party being impersonated.

Device Onboarding

An attacker might make a malicious device onboarding attempt designed to trick the user into onboarding a device that does not belong to the user.

The onboarding processes supported should ensure that the user is required to provide affirmative confirmation that a specific request is approved. For example, requiring a user to enter a security code printed next to a QR code to approve onboarding and not simply scanning the device alone.

IANA Considerations

This document does not specify any actions for IANA.

Acknowledgements

The work presented in this document builds on work by the DNS, OAuth and other IETF Working Groups over a period of decades.

12. Normative References

[did] Sky, B., "Decentralized Identifiers (DIDs) v1.0", 19 July 2022.

[draft-hallambaker-anything]
"[Reference Not Found!]".

[draft-hallambaker-anywhere]
"[Reference Not Found!]".

[draft-hallambaker-mesh-schema]
Hallam-Baker, P., "Mathematical Mesh 3.0 Part IV: Schema Reference", Work in Progress, Internet-Draft, draft-hallambaker-mesh-schema-13, 14 October 2024, <https://datatracker.ietf.org/doc/html/draft-hallambaker-mesh-schema-13>.

[draft-hallambaker-mesh-udf]
Hallam-Baker, P., "Mathematical Mesh 3.0 Part II: Uniform Data Fingerprint.", Work in Progress, Internet-Draft, draft-hallambaker-mesh-udf-19, 14 October 2024, <https://datatracker.ietf.org/doc/html/draft-hallambaker-mesh-udf-19>.

[draft-ietf-oauth-resource-metadata]
Jones, M. B., Hunt, P., and A. Parecki, "OAuth 2.0 Protected Resource Metadata", Work in Progress, Internet-Draft, draft-ietf-oauth-resource-metadata-13, 15 October 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-oauth-resource-metadata-13>.

[draft-ietf-oauth-v2-1]
Hardt, D., Parecki, A., and T. Lodderstedt, "The OAuth 2.1 Authorization Framework", Work in Progress, Internet-Draft, draft-ietf-oauth-v2-1-12, 15 November 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-oauth-v2-1-12>.

[draft-parecki-oauth-client-id-metadata-document]
Parecki, A. and E. Smith, "OAuth Client ID Metadata Document", Work in Progress, Internet-Draft, draft-parecki-oauth-client-id-metadata-document-02, 9 January 2025, <https://datatracker.ietf.org/doc/html/draft-parecki-oauth-client-id-metadata-document-02>.

[plc] Sky, B., "DID Method PLC", January 2025.

[RFC2136] Vixie, P., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, DOI 10.17487/RFC2136, April 1997, <https://www.rfc-editor.org/rfc/rfc2136>.

[RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013, <https://www.rfc-editor.org/rfc/rfc6763>.

[RFC7636] Sakimura, N., Bradley, J., and N. Agarwal, "Proof Key for Code Exchange by OAuth Public Clients", RFC 7636, DOI 10.17487/RFC7636, September 2015, <https://www.rfc-editor.org/rfc/rfc7636>.

[RFC8552] Crocker, D., "Scoped Interpretation of DNS Resource Records through "Underscored" Naming of Attribute Leaves", BCP 222, RFC 8552, DOI 10.17487/RFC8552, March 2019, <https://www.rfc-editor.org/rfc/rfc8552>.

[RFC8553] Crocker, D., "DNS Attrleaf Changes: Fixing Specifications That Use Underscored Node Names", BCP 222, RFC 8553, DOI 10.17487/RFC8553, March 2019, <https://www.rfc-editor.org/rfc/rfc8553>.

[RFC8555] Barnes, R., Hoffman-Andrews, J., McCarney, D., and J. Kasten, "Automatic Certificate Management Environment (ACME)", RFC 8555, DOI 10.17487/RFC8555, March 2019, <https://www.rfc-editor.org/rfc/rfc8555>.

[RFC8984] Jenkins, N. and R. Stepanek, "JSCalendar: A JSON Representation of Calendar Data", RFC 8984, DOI 10.17487/RFC8984, July 2021, <https://www.rfc-editor.org/rfc/rfc8984>.

[RFC9553] Stepanek, R. and M. Loffredo, "JSContact: A JSON Representation of Contact Data", RFC 9553, DOI 10.17487/RFC9553, May 2024, <https://www.rfc-editor.org/rfc/rfc9553>.

13. Informative References

[atproto] Sky, B., "AT Protocol", January 2025.

Author’s Address

Phillip Hallam-Baker
ThresholdSecrets.com
Email: phill@hallambaker.com