The Printer Working Group

October 17, 2025
Candidate Standard 5100.23-2025

IPP OAuth Extensions v1.0 (OAUTH)

Status: Approved

Abstract: This document defines IPP extensions, best practices, and implementation guidance for using OAuth 2.0 and OpenID Connect with IPP.

This is a PWG Candidate Standard. For a definition of a "PWG Candidate Standard", see:

https://ftp.pwg.org/pub/pwg/general/process/pwg-process-4.pdf

This document is available electronically at:

https://ftp.pwg.org/pub/pwg/candidates/cs-ippoauth10-20251017-5100.23.docx
https://ftp.pwg.org/pub/pwg/candidates/cs-ippoauth10-20251017-5100.23.pdf

About the Industry Standards and Technology Organization

The Industry Standards and Technology Organization (ISTO) was established in January 1999 as a global, not-for-profit corporation [501(c)(6)] designed to accelerate and extend standards development and adoption activities for technology industry consortia, trade associations and other organizations.

For additional information regarding the ISTO and its industry programs visit:

https://www.isto.org/

About the ISTO Printer Working Group

The Printer Working Group (PWG) is a Program of the ISTO with member organizations including printer manufacturers, print server developers, operating system providers, network operating system providers, network connectivity vendors, and print management application developers. The PWG is chartered to make printers and the applications and operating systems supporting them work together better. All references to the PWG in this document implicitly mean “The Printer Working Group, a Program of the ISTO.”

To meet this objective, the PWG documents the results of their work as open standards that define print related protocols, interfaces, procedures, and conventions. A PWG standard is a stable, well understood, and technically competent specification that is widely used with multiple independent and interoperable implementations. Printer manufacturers and vendors of printer related software benefit from the interoperability provided by voluntary conformance to these standards.

For additional information regarding the Printer Working Group visit:

https://www.pwg.org

Contact information:

The Printer Working Group
c/o The Industry Standards and Technology Organization
445 Hoes Lane
Piscataway, NJ 08854
USA

Table of Contents

List of Figures

List of Tables

1. Introduction

The OAuth 2.0 Authorization Framework [RFC6749] and OpenID Connect [OPENID] are used to provide identification, authorization, and access control for a wide variety of Internet applications, including printing. This document defines the IPP extensions, best practices, and implementation guidance for using OAuth 2.0 and OpenID Connect with the Internet Printing Protocol.

OAuth 2.0 provides various optional components needed for printing while OpenID Connect makes most of these components mandatory. Necessary and recommended components are listed with guidance on interoperability between IPP Clients, Printers, and Proxies and OAuth 2.0 and OpenID Connect authorization servers.

OAuth also requires the use of TLS [RFC8446] and X.509 certificates that have been signed by a Trusted Root/Certificate Authority (CA). These requirements, along with implementation recommendations, are described in the security considerations (section 10).

2. Terminology

2.1 Conformance Terminology

Capitalized terms, such as MUST, MUST NOT, RECOMMENDED, REQUIRED, SHOULD, SHOULD NOT, MAY, and OPTIONAL, have special meaning relating to conformance as defined in Key words for use in RFCs to Indicate Requirement Levels [BCP14]. This specification defines the following additional capitalized conformance terms:

CONDITIONALLY REQUIRED: A MUST conformance requirement that applies only when a specified condition is true.

DEPRECATED: A SHOULD NOT conformance requirement for previously defined and approved protocol elements that are planned to be removed from use.

OBSOLETE: A MUST NOT conformance requirement for previously defined and approved protocol elements that have been removed from use.

2.2 Printing Terminology

The following printing terms are used in this document:

Administrator: An End User who is also authorized to manage all aspects of an Output Device or Printer, including creating the printer instances and controlling the authorization of other End Users and Operators [RFC2567].

Authenticated User: The same as "authenticated user" defined in [STD92] RFC 8011 Section 9.3.

Document: An object created and managed by a Printer that contains the description, processing, and status information. A Document object may have attached data and is bound to a single Job [PWG5100.5].

End User: A person or software process that is authorized to perform basic printing functions, including finding/locating a printer, creating a local instance of a printer, viewing printer status, viewing printer capabilities, submitting a print job, viewing print job status, and altering the attributes of a print job [RFC2567].

i^{th}: Referring to a specific IPP '1setOf' value - the first value, the second value, and so forth.

Job: An object created and managed by a Printer that contains description, processing, and status information. The Job also contains zero or more Document objects.

Logical Device: A print server, software service, or gateway that processes jobs and either forwards or stores the processed job or uses one or more Physical Devices to render output [STD92].

Operator: An End User that also has special rights on the Output Device or Printer. The Operator typically monitors the status of the Printer and manages and controls the Jobs at the Output Device [RFC2567]. The Operator is allowed to query and control the Printer, Jobs, and Documents based on site policy.

Output Device: A single Logical or Physical Device [STD92].

Physical Device: A hardware implementation of a endpoint device, e.g., a marking engine, a fax modem, etc. [STD92]

2.3 Protocol Role Terminology

The following protocol roles are defined to specify unambiguous conformance requirements:

Client: Initiator of outgoing connections and sender of outgoing operation requests (Hypertext Transfer Protocol -- HTTP/1.1 [STD99] User Agent) [STD92].

Infrastructure Printer: A Printer that represents a Logical Device associated with both a Client and Proxy [PWG5100.18]. For Cloud-based implementations, the Infrastructure Printer corresponds to a Cloud Imaging Service [PWG5109.1].

Infrastructure System: A System that represents an entire Imaging System and accepts incoming requests and connections from both Clients and Proxies and contains zero or more Infrastructure Printers [PWG5100.18]. For Cloud-based implementations, the Infrastructure System corresponds to a Cloud Imaging System [PWG5109.1].

Printer: Listener for incoming connections and receiver of incoming operation requests (Hypertext Transfer Protocol -- HTTP/1.1 [STD99] Server) that represents one or more Physical Devices or a Logical Device [STD92].

Proxy: A Client that sends configuration and status information to and retrieves and manages Jobs and Documents from an Infrastructure Printer [PWG5100.18] on behalf of one or more Output Devices and also communicates internally with an Infrastructure System to register the local System and get back Infrastructure Printer URIs.

System: Listener for incoming IPP session requests and receiver of incoming IPP operation requests (Hypertext Transfer Protocol -- HTTP/1.1 [STD99] Server) that exposes an IPP System object and implements a System Service [PWG5100.22].

2.4 Other Terminology

The following other terms are used in this document:

Allow List: A list of discrete entities, such as hosts or applications that are known to be benign and are approved for use within an organization and/or information system [NIST-SP800-167].

Authorization Server (AUTHZ): The server issuing access tokens to the Client after successfully authenticating the Resource Owner and obtaining authorization [RFC6749].

Certificate Authority (CA): The issuer of an X.509 certificate used for authenticating TLS connections during the initial handshake [RFC5280].

Resource Owner: An entity capable of granting access to a protected resource (Printer/System). When the Resource Owner is a person, it is referred to as an End User [RFC6749].

Transport Layer Security (TLS): A communication protocol that protects data in transit from eavesdropping, tampering, and message forgery [RFC8446].

Trust Anchor: A public or symmetric key that is trusted because it is directly built into hardware or software, or securely provisioned via out-of-band means, rather than because it is vouched for by another trusted entity (e.g. in a public key certificate). A trust anchor may have name or policy constraints limiting its scope [NIST-SP800-63-3].

Trust On First Use (TOFU): In a protocol, TOFU calls for accepting and storing a public key or credential associated with an asserted identity, without authenticating that assertion. Subsequent communication that is authenticated using the cached key or credential is secure against an MiTM attack, if such an attack did not succeed during the vulnerable initial communication [RFC7435].

2.5 Acronyms and Organizations

IANA: Internet Assigned Numbers Authority, https://www.iana.org/

IETF: Internet Engineering Task Force, https://www.ietf.org/

ISO: International Organization for Standardization, https://www.iso.org/

PWG: The ISTO Printer Working Group, https://www.pwg.org/

3. Requirements

3.1 Rationale

Given the following existing specifications:

1. Internet Printing Protocol/1.1 [STD92]
2. OpenID Connect [OPENID];
3. The OAuth 2.0 Authorization Framework [RFC6749];
4. The OAuth 2.0 Authorization Framework: Bearer Token Usage [RFC6750];
5. JSON Web Token (JWT) [RFC7519];
6. OAuth 2.0 Dynamic Client Registration Protocol [RFC7591];
7. Proof Key for Code Exchange by OAuth Public Clients [RFC7636];
8. OAuth 2.0 for Native Apps [RFC8252];
9. OAuth 2.0 Authorization Server Metadata [RFC8414];
10. OAuth 2.0 Device Authorization Grant [RFC8628];
11. OAuth 2.0 Token Exchange [RFC8693];
12. JSON Web Token Best Current Practices [RFC8725]; and
13. JSON Web Token (JWT) Profile for OAuth 2.0 Access Tokens [RFC9068].

And given the need for a flexible, single sign-on authorization mechanism for IPP, the IPP OAuth Extensions v1.0 should:

1. Define IPP attributes and values to support the use of OAuth and OpenID Connect; and
2. Define best practices for the use of OAuth with IPP.

3.2 Use Cases

3.2.1 Print with OAuth Authorization

Alex wants to print on a printer that requires authentication against a network account. They open the print dialog in an application, choose print options, and submit a print job. Their client device queries the printer, discovers it requires authentication against a permitted Authorization Server, and displays the corresponding authorization web page. After Alex

completes the authorization page, the Authorization Server returns a token to the client device, allowing it to obtain an access token for the printer and submit the job for printing.

3.2.2 Print with Cached OAuth Authorization

Sam wants to print several documents to a printer that requires authentication against a network account. The first document prints as in section 3.2.1. For the remaining documents, the client device re-uses the access token from the first print job for subsequent submissions.

3.2.3 Print to Multiple Domains

Pat often uses a personal laptop to prepare and print company reports and other documents. When printing to a home printer, Pat authorizes print jobs using their home account. When printing from home or in the office to a work printer, Pat uses their work account to authorize printing. The client software on Pat's laptop authorizes and caches any access tokens separately for each printer and domain.

3.3 Exceptions

The following subsections define exceptions in addition to those defined in the Internet Printing Protocol/1.1 [STD92].

3.3.1 Unknown Authorization Server

Alex submits a print job for a printer that specifies an unknown Authorization Server. The client device can connect to the Authorization Server to validate the Authorization Server's X.509 certificate and then, if valid, prompt Alex to allow the Authorization Server. Alternately, the client device can prompt Alex to explain that the Authorization Server is unknown to the client device and to contact their Administrator to investigate and act accordingly.

3.3.2 Known But Disallowed Authorization Server

Sam submits a print job for a printer that specifies an Authorization Server that is not an allowed server. The client device displays an error message telling Sam that they cannot use the printer.

3.3.3 Authorization Failed

Alex attempts to authorize access to a printer using the Authorization Server web page. When authorization fails, the client device displays an error message and does not access the printer.

3.3.4 Access Token Expired

Sam is printing their 100th document. When the client device submits this job it gets an error indicating that the cached access token has expired. Their client device then attempts to

refresh the access token or, if necessary, display the authorization web page in order to re-authorize submission of the print job.

3.3.5 Failed X.509 Certificate Validation

Jesse is printing to a network printer. The client device negotiates a TLS connection to the printer and validates the printer's X.509 certificate. If validation fails, the client device displays an error message and stops communicating with the printer.

3.4 Out of Scope

The following are considered out of scope for this white paper:

1. Extensions to the OAuth or OpenID Connect protocols.

3.5 Design Requirements

The design requirements for this white paper are:

1. Define a minimum profile of OAuth RFCs and features needed to support interoperable printing;
2. Define a minimum profile of OpenID Connect specifications and features needed to support interoperable printing;
3. Define IPP attributes and values needed to support OAuth and OpenID Connect;
4. Define best practices for Clients and Proxies to discover and use OAuth and OpenID Connect;
5. Define best practices for Printers and Systems to advertise, configure, and use OAuth and OpenID Connect;
6. Define internationalization, security, and privacy considerations; and
7. Define sections to register all attributes and values with IANA.

4. Model

4.1 General Trust Relationships

Figure 1 shows the relationships between the common IPP and OAuth actors. Trust is established first through negotiation of a secure TLS [RFC8446] connection to the Authorization Server, Printer, and/or System. OAuth requests are used to obtain and/or query metadata from (introspect) access tokens, while IPP requests provide these tokens in the HTTP "Authorization" request header, which are then validated by the recipient.

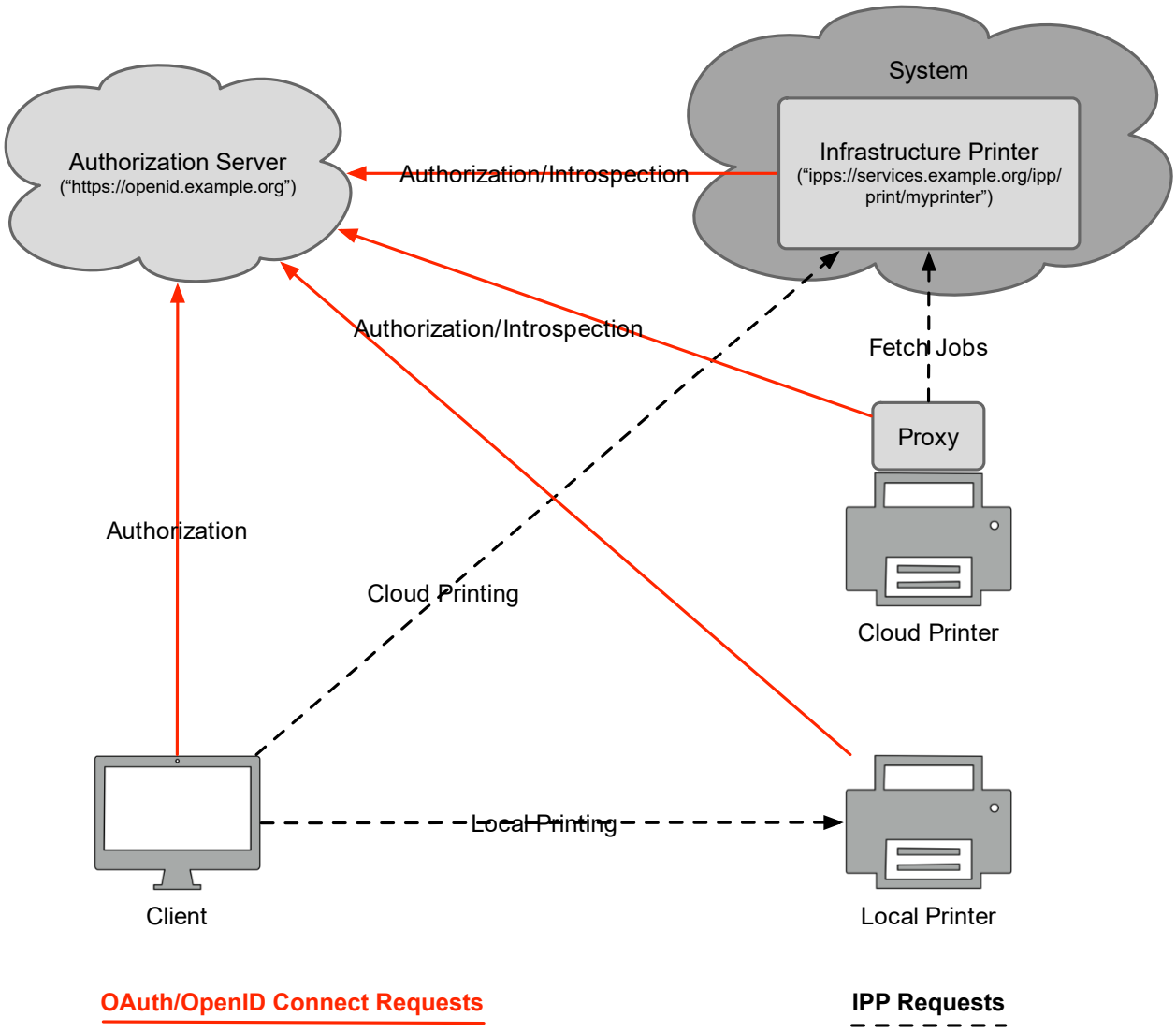

Figure 1 - General Relationships for OAuth-Based Printing

Authorization Servers are configured out-of-band for Printers and Systems, with the corresponding URI reported to Clients in the "oauth-authorization-server-uri" Printer/System Description attribute (section 5.1.2).

Table 1 summarizes how each actor establishes trust with another.

Table 1 - Establishing Trust Relationships

Relationship	Mechanism
Client/Proxy Trusts AUTHZ	Allow List, X.509 Validation
AUTHZ Trusts Client	Device Registration; User Authentication
Printer/System Trusts AUTHZ	Allow List, X.509 Validation
AUTHZ Trusts Printer/System	Device Registration
Client/Proxy Trusts Printer/System	X.509 Validation; URI Validation
Printer/System Trusts Client/Proxy	Access Token Validation
Client/Proxy Trusts AUTHZ and Printer/System Know Each Other	Token Exchange

4.1.1 Trusting Authorization Servers

OAuth and OpenID Connect provide a HTTP-based authorization solution. Authorization Servers are identified by URI such as "https://samples.auth0.com" and by the X.509 certificate received from the server when negotiating a TLS [RFC8446] connection.

The Client, Proxy, Printer, or System MUST only use the specified Authorization Server if its URI is in an Allow List. Similarly, the Client, Proxy, Printer, or System MUST validate that the X.509 certificate [ITU-X509] is signed by a Trust Anchor and has a matching common name or subject alternate name for the hostname in the URI [RFC9325].

4.1.2 Trusting Clients and Proxies

Client and Proxy trust is established through OAuth authorization. Clients and Proxies query the "oauth-authorization-server-uri" Printer/System Description attribute (section 5.1.2) and then obtain an access token from the Authorization Server for the given Printer or System. The access token is supplied in the HTTP Authorization header using the Bearer [RFC6750] scheme. Figure 2 and Figure 3 show the general sequence of requests for establishing trust, submitting a print Job, and monitoring that Job using authorization through a web browser. Other flows are described in section 7.3.

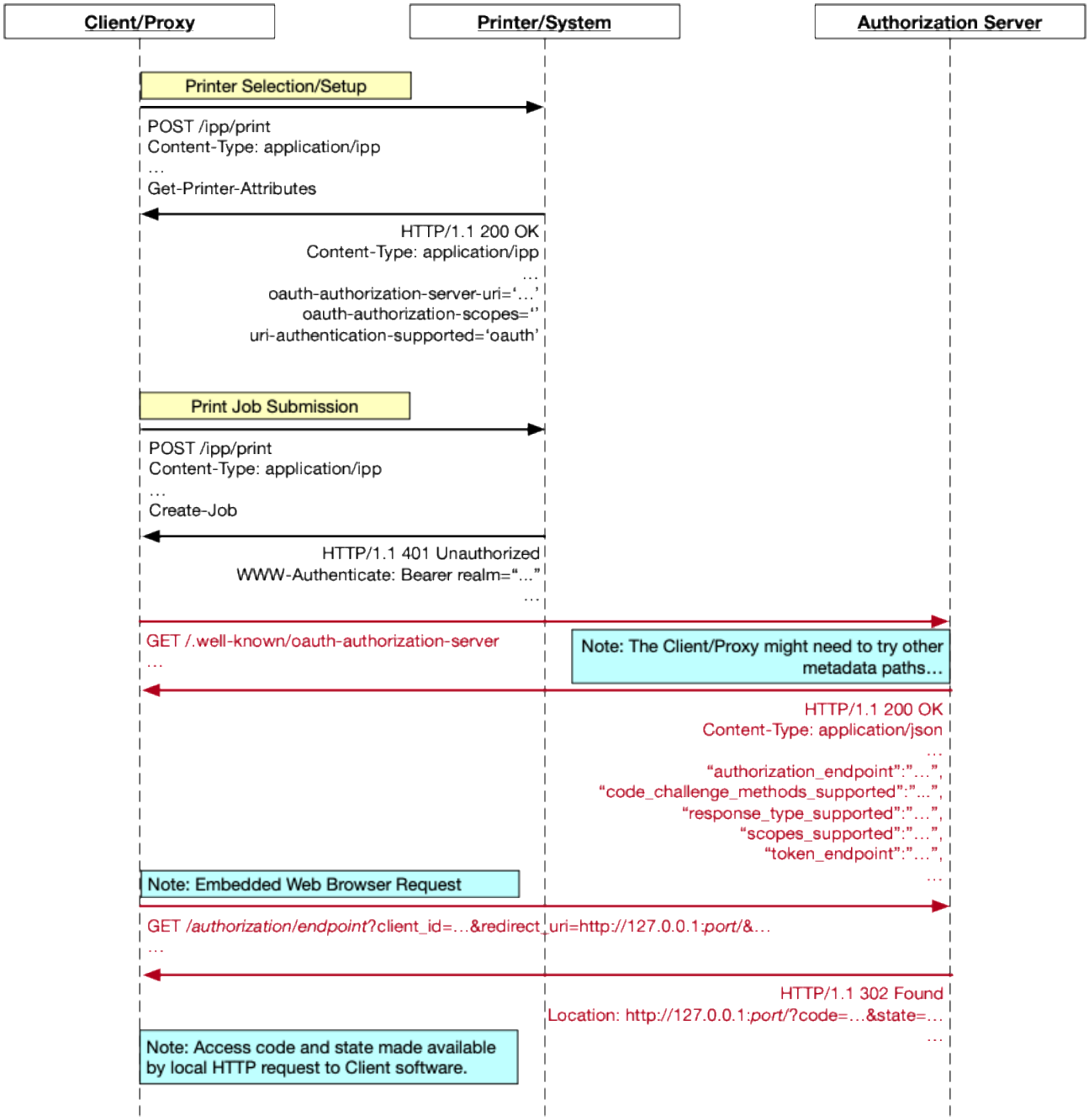

Figure 2 - Establishing Client/Proxy Trust

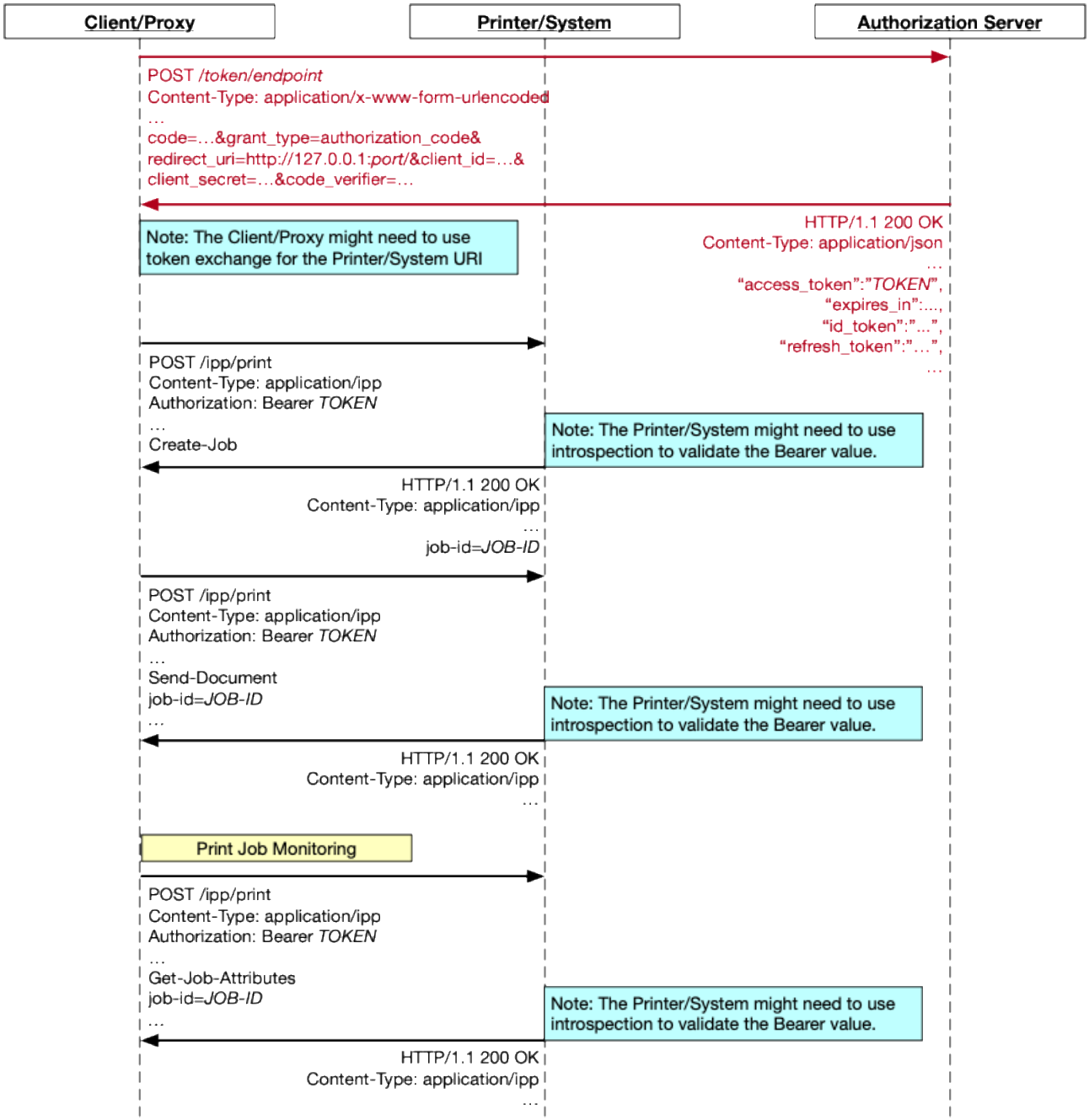

Figure 3 - Obtaining Credentials and Printing

4.1.3 Trusting Printers and Systems

IPP Printers and Systems are identified using "ipps" URIs such as "ipps://pr123456789abc.local/ipp/print" and "ipps://system.example.com/ipp/system". They are validated using the X.509 certificate [ITU-X509] received from the Printer or System when negotiating a TLS [RFC8446] connection.

The Client or Proxy MUST validate that the X.509 certificate is signed by a Trust Anchor and has a matching common name or subject alternate name for the hostname in the Printer or System URI [RFC9325]. Because the Client or Proxy cannot know whether OAuth is in use prior to sending its first Get-Printer-Attributes request, the Client or Proxy might need to re-evaluate the X.509 certificate after getting the response since Trust On First Use (TOFU) might otherwise be an acceptable Printer or System trust validation policy for the Client.

4.2 Authorization Server URI and Metadata

The OAuth Authorization Server URI is provided by the "oauth-authorization-server-uri" Printer/System Description attribute (section 5.1.2) and provides the base URL for the Authorization Server. For example, an Authorization Server might use "https://example.com" for a common authorization endpoint or "https://example.com/tenant-id/oauth" for an authorization endpoint for a specific organization or customer.

Metadata is critical for both endpoint discovery and supported values. Clients, Printers, Proxies, and Systems MUST support both OAuth 2.0 Authorization Server Metadata ("/.well-known/oauth-authorization-server") [RFC8414] and OpenID Connect Discovery metadata ("/.well-known/openid-configuration") [OPENID] to provide maximum compatibility and interoperability. Once obtained, the metadata can be inspected to confirm that all required endpoints and values are supported.

When hosting an embedded web server that also requires OAuth authorization, Printers and Systems SHOULD provide OAuth 2.0 Protected Resource Metadata [RFC9728] to allow HTTP/HTTPS clients to discover the OAuth Authorization Server URI and associated values that are needed.

4.3 Scoped Access Tokens

OAuth offers different methods of limiting the scope of access tokens:

> *Scopes*: OAuth 2.0 [RFC6749] defines a list of named scopes that can be requested during authorization which are listed in the "oauth-authorization-scope" Printer/System Description attribute (section 5.1.1); and
>
> *Token Exchange*: OAuth 2.0 Token Exchange [RFC8693] requests an access token that is restricted to the specified resource (Printer or System) URI.

Clients, Printers, Proxies, and Systems MUST support OAuth 2.0 scopes and Token Exchange. Printers and Systems MUST validate access tokens, either using OAuth 2.0 Token Introspection [RFC7662] or JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants [RFC7523] and JSON Web Token (JWT) Profile for OAuth 2.0 Access Tokens [RFC9068]. When using Token Exchange, Printers and Systems MUST validate that the access token resource URI matches one of the supported Printer or System URIs, respectively.

4.4 Token Reuse/Caching

OAuth 2.0 access tokens can be cached and reused according to [RFC6749]. When Token Exchange [RFC8693] is used, both the End User's access token and the Printer/System access token are cached separately.

4.5 Authorization Server Allow Lists

Allow Lists are used to limit potentially malicious Authorization Servers from being used and/or to protect against "phishing" attacks. An implementation provides an Allow List that is configured by an Administrator, either directly through local Client actions/files or through a trusted network service outside the scope of IPP.

5. New IPP Attributes

5.1 Printer/System Description Attributes

5.1.1 oauth-authorization-scope (1setOf name(MAX) | no-value)

This REQUIRED attribute provides an ordered list of OAuth 2.0 scopes that have been configured to be used in an authorization request for this Printer or System. If the attribute lists more than one scope name, the first name provides the least access, e.g., the "End User" role in IPP, while the last name provides the most access, e.g., the "Administrator" role in IPP. The scope name 'openid' has special meaning for OpenID Connect [OPENID].

Clients SHOULD provide the full list of scopes in the initial OAuth 2.0 authorization request and only prune the list if the Authorization Server returns the "invalid_scope" error.

5.1.2 oauth-authorization-server-uri (uri | no-value)

This REQUIRED attribute provides the base URL of the OAuth/OpenID Authorization Server configured for use with the Printer or System. The Printer or System MUST validate that the Authorization Server supports OAuth 2.0 Authorization Server Metadata [RFC8414] and/or OpenID Connect Discovery metadata [OPENID].

6. New Values for Existing Attributes

6.1 uri-authentication-supported (1setOf type2 keyword)

This specification defines a new keyword, 'oauth', to indicate that a Printer is configured to use the OAuth 2.0 Authentication Framework [RFC6749] or OpenID Connect [OPENID] for the corresponding "printer-uri-supported" [STD92] value.

6.2 xri-authentication-supported (1setOf type2 keyword)

This specification defines a new keyword, 'oauth', to indicate that a Printer or System is configured to use the OAuth 2.0 Authentication Framework [RFC6749] or OpenID Connect [OPENID] for the specified "printer-xri-supported" [STD92] or "system-xri-supported" [PWG5100.22] value.

7. Implementation Guidance

The following subsections provide non-normative guidance in implementing OAuth 2.0 and OpenID support based on prototyping experience.

7.1 Allow Lists

For both security and privacy reasons, Clients, Printers, Proxies, and Systems normally will default to an empty OAuth Authorization Server Allow List. The configuration interface provided to Administrators can easily offer a list of choices in addition to a manual configuration interface, such as:

- Public OAuth/OpenID Connect Authorization Servers that can be used by anyone;
- Public IPP "cloud printing" solutions that have an IPP System and associated OAuth Authorization Server; and
- A private "managed printing" solution that can be discovered using DNS-SD SRV queries of the "_ipps-system._tcp" service type.

If manual entry of OAuth Authorization Server URIs is supported, the capabilities of the server need to be validated using its metadata so that the Client, Printer, Proxy, or System is not configured to use an incompatible server.

7.2 Metadata

OAuth uses OAuth 2.0 Authorization Server Metadata [RFC8414] located at "/.well-known/oauth-authorization-server" while OpenID Connect [OPENID] provides its metadata document at "/.well-known/openid-configuration". Both are simple JSON objects with compatible names and values.

Clients, Printers, Proxies, and Systems need to try both well-known paths, first by appending the paths to the supplied Authorization Server URI and then by replacing the path components of the URI. For example, if the Authorization Server URI is "https://oauth.example.com/tenant/42", then the requestor could try the following metadata URLs:

- https://oauth.example.com/tenant/42/.well-known/oauth-authorization-server
- https://oauth.example.com/tenant/42/.well-known/openid-configuration
- https://oauth.example.com/.well-known/oauth-authorization-server
- https://oauth.example.com/.well-known/openid-configuration

7.3 Authorization Flows

Clients, Printers, Proxies, and Systems can use any OAuth authorization flow supported by the Authorization Server.

OAuth 2.0 Proof Key for Code Exchange [RFC7636] and OpenID Connect [OPENID] use two different methods for securing the interactive ("native") authorization flow. If the "code_challenge_methods_supported" metadata contains the value "S256" (SHA-256 hash), then the Authorization Server supports the PKCE "code_challenge" and "code_verifier" parameters. Similarly, if the "scopes_supported" metadata contains the value "openid", then the Authorization Server supports the OpenID Connect "nonce" parameter.

Both OAuth 2.0 [RFC6749] and OpenID Connect can also support the OAuth 2.0 Device Authorization Grant [RFC8628] authorization flow. If the "device_authorization_endpoint" metadata is present with a "https:" URL value, then the Authorization Server supports device authorization grants.

Finally, Clients, Printers, Proxies, and Systems can choose between OAuth 2.0 authorization [RFC6749], token exchange [RFC8693], and JWT [RFC7523] codes when obtaining access tokens from the Authorization Server's token endpoint. If the "grant_types_supported" metadata contains the value "urn:ietf:params:oauth:grant-type:token-exchange", then the Authorization Server supports token exchange. The "resource" parameter that is passed to the token endpoint for token exchange is the "https:" version of the Printer or System URI [RFC7472].

If the "grant_types_supported" metadata contains the value "urn:ietf:params:oauth:grant-type:jwt-bearer", then the Authorization Server supports JWT access tokens.

7.4 Authorized User Names

The IPP Authenticated Username is copied to the "job-originating-user-name" [STD92] and "job-originating-user-uri" [PWG5100.13] Job Status attributes when processing a Job Creation request. The JWT "sub" [RFC7519], "name" [OPENID], and "preferred_username" [OPENID] claims are good choices for the value of the "job-originating-user-name" attribute. Similarly, the "email" [OPENID] or "phone_number" [OPENID] claims can be used to construct a value for the "job-originating-user-uri" attribute.

Note: The "job-originating-user-xxx" attributes are purely informative and allow recipients to informally determine the owner of a particular Job in a Get-Jobs or Get-Job-Attributes [STD92] response, but these attributes are not used for authorization or access control. Instead, a Printer implementation privately stores any Authenticated User credentials in the Job object, and uses those privately stored credentials for subsequent authorization and/or access control, subject to the Printer and/or site security policies.

7.5 Client Registration, "client_id", and "client_secret"

The OAuth 2.0 Dynamic Client Registration Protocol [RFC7591] is used by both OAuth and OpenID Connect [OPENID] Authorization Servers to register an OAuth client and obtain the "client_id" and "client_secret" values that are used during authorization. If the "registration_endpoint" metadata is present with a "https:" URL value, then the Authorization Server supports dynamic client registration.

In the absence of dynamic client registration support, pre-registered values for public Authorization Servers can be used safely so long as PKCE [RFC7636] and/or OpenID Connect "nonce" values are supported by the authorization flow. Finally, manually-obtained "client_id" and "client_secret" values can be used so long as they are protected from disclosure in transit and at rest.

8. Conformance Requirements

In order for a Client, Proxy, Printer, or Server to claim conformance to this specification, a Client, Proxy, Printer, or Server supports:

1. The required attributes and values defined in section 5;
2. The additional values defined in section 6;
3. The internationalization requirements defined in section 9; and
4. The security and privacy requirements defined in section 10.

9. Internationalization Considerations

For interoperability and basic support for multiple languages, conforming implementations MUST support:

1. The Universal Character Set (UCS) Transformation Format -- 8 bit (UTF-8) [STD63] encoding of Unicode [UNICODE] [ISO10646]; and
2. The Unicode Format for Network Interchange [RFC5198] which requires transmission of well-formed UTF-8 strings and recommends transmission of normalized UTF-8 strings in Normalization Form C (NFC) [UAX15].

Unicode NFC is defined as the result of performing Canonical Decomposition (into base characters and combining marks) followed by Canonical Composition (into canonical composed characters wherever Unicode has assigned them).

WARNING – Performing normalization on UTF-8 strings received from Clients and subsequently storing the results (e.g., in Job objects) could cause false negatives in Client searches and failed access (e.g., to Printers with percent-encoded UTF-8 URIs now 'hidden').

Implementations of this specification SHOULD conform to the following standards on processing of human-readable Unicode text strings, see:

Unicode Bidirectional Algorithm [UAX9] – left-to-right, right-to-left, and vertical

Unicode Line Breaking Algorithm [UAX14] – character classes and wrapping

Unicode Normalization Forms [UAX15] – especially NFC for [RFC5198]

Unicode Text Segmentation [UAX29] – grapheme clusters, words, sentences

Unicode Identifier and Pattern Syntax [UAX31] – identifier use and normalization

Unicode Collation Algorithm [UTS10] – sorting

Unicode Locale Data Markup Language [UTS35] – locale databases

Implementations of this specification are advised to also review the following informational documents on processing of human-readable Unicode text strings:

Unicode Character Encoding Model [UTR17] – multi-layer character model

Unicode Character Property Model [UTR23] – character properties

Unicode Conformance Model [UTR33] – Unicode conformance basis

10. Security and Privacy Considerations

In addition to the security and privacy considerations defined in the Internet Printing Protocol/1.1 [STD92], IPP Shared Infrastructure Extensions v1.0 (INFRA) [PWG5100.18], IPP System Service v1.0 (SYSTEM) [PWG5100.22], Best Current Practice for OAuth 2.0 Security [BCP240], and OpenID Security Best Practices [OPENID-BP], the following subsections define security considerations for using OAuth and OpenID Connect with IPP.

10.1 Unicode Security

Implementations of this specification SHOULD conform to the following standard on processing of human-readable Unicode text strings, see:

Unicode Security Mechanisms [UTS39] – detecting and avoiding security attacks

Implementations of this specification are advised to also review the following informational document on processing of human-readable Unicode text strings:

Unicode Security FAQ [UNISECFAQ] – common Unicode security issues

10.2 Protection of OAuth Tokens

OAuth tokens MUST be treated as data to be protected in transit and at rest with appropriate local access controls to protect against misuse because they provide access to protected resources such as Printers and Systems.

10.3 Protection of Data in Transit, X.509 Certificate Validation

Clients and Proxies MUST negotiate secure connections to Printers and Systems using TLS [RFC8446] [BCP195]. Similarly, Clients, Printers, Proxies, and Systems MUST negotiate secure connections to Authorization Servers.

X.509 certificates [ITU-X509] exchanged during TLS negotiation MUST use strict validation (section 4.1), including validation of the hostname to the certificate's common name and/or subject alternate name fields and validation of the Trust Anchor. So-called "self-signed" certificates MUST NOT be allowed since there is no Trust Anchor to validate.

Internet-accessible services SHOULD use X.509 certificates provided by public CAs. Non-Internet services SHOULD use X.509 certificates generated by local trusted certificate services such as an enterprise certificate server or an IoT ACME service [IOT-ACME].

10.4 OAuth Authorization Server Validation

Clients, Printers, Proxies, and Systems MUST limit use of OAuth Authorization Servers to those whose URL is present in and/or matches an Allow List configured by the Administrator and whose X.509 certificate is signed by a Trusted Root certificate.

11. IANA Considerations

11.1 Attribute Registrations

The attributes defined in this specification will be published by IANA according to the procedures in the Internet Printing Protocol/1.1 [STD92] in the following location:

https://www.iana.org/assignments/ipp-registrations

The registry entries will contain the following information:

```
Printer Description attributes:                          Reference
-------------------------------                          ---------
oauth-authorization-scope (1setOf name(MAX) | no-value)  [PWG5100.23]
oauth-authorization-server-uri (uri | no-value)          [PWG5100.23]

System Description attributes:                           Reference
------------------------------                           ---------
oauth-authorization-scope (1setOf name(MAX) | no-value)  [PWG5100.23]
oauth-authorization-server-uri (uri | no-value)          [PWG5100.23]
```

11.2 Type2 keyword Registrations

The keyword values defined in this specification will be published by IANA according to the procedures in the Internet Printing Protocol/1.1 [STD92] in the following location:

https://www.iana.org/assignments/ipp-registrations

The registry entries will contain the following information:

```
Attributes (attribute syntax)
  Keyword Attribute Value                                Reference
  -----------------------                                ---------
uri-authentication-supported (1setOf type2 keyword)      [STD92]
  oauth                                                  [PWG5100.23]
xri-authentication-supported (1setOf type2 keyword)      [RFC3380]
  oauth                                                  [PWG5100.23]
```


12. References

12.1 Normative References

[BCP14] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119/BCP 14, March 1997, https://datatracker.ietf.org/doc/html/rfc2119

[BCP195] Y. Sheffer, P. Saint-Andre, T. Fossati, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195/RFC 9325, November 2022, https://datatracker.ietf.org/doc/html/rfc9325

[BCP240] T. Lodderstedt, J. Bradley, A. Labunets, D. Fett, "Best Current Practice for OAuth 2.0 Security", RFC 9700/BCP 240, January 2025, https://datatracker.ietf.org/doc/html/rfc9700

[ISO10646] "Information technology -- Universal Coded Character Set (UCS)", ISO/IEC 10646:2011

[ITU-X509] "Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks", Recommendation X.509, October 2019, https://www.itu.int/rec/T-REC-X.509-201910-I/en

[NIST-SP800-63-3] P. Grassi, M. Garcia, J. Fenton, "Digital Identity Guidelines", NIST SP 800-63-3, June 2017, https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63-3.pdf

[NIST-SP800-167] A. Sedgewick, M. Souppaya, K. Scarfone, "Guide to Application Whitelisting", NIST SP 800-167, October 2015, https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-167.pdf

[OPENID] "Welcome to OpenID Connect", https://openid.net/connect/

[PWG5100.5] M. Sweet, "IPP Document Object v1.2 (DOCOBJECT)", PWG 5100.5-2024, May 2024, https://ftp.pwg.org/pub/pwg/candidates/cs-ippdocobject12-20240517.pdf

[PWG5100.18] M. Sweet, I. McDonald, “IPP Shared Infrastructure Extensions v1.1 (INFRA)”, PWG 5100.18-2025, March 2025, https://ftp.pwg.org/pub/pwg/candidates/cs-ippinfra11-20250331-5100.18.pdf

[PWG5100.22] M. Sweet, I. McDonald, “IPP System Service v1.1 (SYSTEM)”, PWG 5100.22-2025, February 2025, https://ftp.pwg.org/pub/pwg/candidates/cs-ippsystem11-20250227-5100.22.pdf

[PWG5109.1] R. Nevo, W. Wagner, “Cloud Imaging Requirements and Model (IMAGINGMODEL)”, PWG 5109.1-2015, June 2015, https://ftp.pwg.org/pub/pwg/candidates/cs-cloudimagingmodel10-20150619-5109.1.pdf

[RFC5198] J. Klensin, M. Padlipsky, "Unicode Format for Network Interchange", RFC 5198, March 2008, https://datatracker.ietf.org/doc/html/rfc5198

[RFC5280] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008, https://datatracker.ietf.org/doc/html/rfc5280

[RFC6749] D. Hardt, "The OAuth 2.0 Authorization Framework", RFC 6749, October 2012, https://datatracker.ietf.org/doc/html/rfc6749

[RFC6750] M. Jones, D. Hardt, "The OAuth 2.0 Authorization Framework: Bearer Token Usage", RFC 6750, October 2012, https://datatracker.ietf.org/doc/html/rfc6750

[RFC7523] M. Jones, B. Campbell, C. Mortimore, "JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants", RFC 7523, May 2015, https://datatracker.ietf.org/doc/html/rfc7523

[RFC7591] J. Richer, M. Jones, J. Bradley, M. Machulak, P. Hunt, "OAuth 2.0 Dynamic Client Registration Protocol", RFC 7591, July 2015, https://datatracker.ietf.org/doc/html/rfc7591

[RFC7636] N. Sakimura, J. Bradley, N. Agarwal, "Proof Key for Code Exchange by OAuth Public Clients", RFC 7636, September 2015, https://datatracker.ietf.org/doc/html/rfc7636

[RFC7662] J. Richer, "OAuth 2.0 Token Introspection", RFC 7662, October 2015, https://datatracker.ietf.org/doc/html/rfc7662

[RFC8252] W. Denniss, J. Bradley, "OAuth 2.0 for Native Apps", RFC 8252, October 2017, https://datatracker.ietf.org/doc/html/rfc8252

[RFC8414] M. Jones, N. Sakimura, J. Bradley, "OAuth 2.0 Authorization Server Metadata", RFC 8414, June 2018, https://datatracker.ietf.org/doc/html/rfc8414

[RFC8446] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, August 2018, https://datatracker.ietf.org/doc/html/rfc8446

[RFC8628] W. Denniss, J. Bradley, M. Jones, H. Tschofenig, "OAuth 2.0 Device Authorization Grant", RFC 8628, August 2019, https://datatracker.ietf.org/doc/html/rfc8628

[RFC8693] M. Jones, A. Nadalin, B. Campbell, J. Bradley, C. Mortimore, "OAuth 2.0 Token Exchange", RFC 8693, January 2020, https://datatracker.ietf.org/doc/html/rfc8693

[RFC9068] V. Bertocci, "JSON Web Token (JWT) Profile for OAuth 2.0 Access Tokens", RFC 9068, October 2021, https://datatracker.ietf.org/doc/html/rfc9068

[RFC9728] M.B. Jones, P. Hunt, A. Parecki, "OAuth 2.0 Protected Resource Metadata", RFC 9728, April 2025, https://datatracker.ietf.org/doc/html/rfc9728

[STD63] F. Yergeau, "UTF-8, a transformation format of ISO 10646", RFC 3629/STD 63, November 2003, https://datatracker.ietf.org/doc/html/rfc3629

[STD92] M. Sweet, I. McDonald, "Internet Printing Protocol/1.1", RFC 8010/RFC 8011/STD 92, June 2018, https://datatracker.ietf.org/doc/html/rfc8010, https://datatracker.ietf.org/doc/html/rfc8011

[STD99] R. Fielding, J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 9112/STD 99, June 2014, https://datatracker.ietf.org/doc/html/rfc9112

[UAX9] Unicode Consortium, “Unicode Bidirectional Algorithm”, UAX#9, September 2024, https://www.unicode.org/reports/tr9

[UAX14] Unicode Consortium, “Unicode Line Breaking Algorithm”, UAX#14, September 2024, https://www.unicode.org/reports/tr14

[UAX15] M. Davis, M. Duerst, "Unicode Normalization Forms", Unicode Standard Annex 15, August 2024, https://www.unicode.org/reports/tr15

[UAX29] Unicode Consortium, “Unicode Text Segmentation”, UAX#29, August 2024, https://www.unicode.org/reports/tr29

[UAX31] Unicode Consortium, “Unicode Identifier and Pattern Syntax”, UAX#31, September 2024, https://www.unicode.org/reports/tr31

[UNICODE] Unicode Consortium, "Unicode Standard", Version 16.0.0, September 2024, https://www.unicode.org/versions/Unicode16.0.0/

[UTS10] Unicode Consortium, “Unicode Collation Algorithm”, UTS#10, August 2024, https://www.unicode.org/reports/tr10

[UTS35] Unicode Consortium, “Unicode Locale Data Markup Language”, UTS#35, March 2025, https://www.unicode.org/reports/tr35

[UTS39] Unicode Consortium, “Unicode Security Mechanisms”, UTS#39, September 2024, https://www.unicode.org/reports/tr39

12.2 Informative References

[IOT-ACME] M. Sweet, "ACME-Based Provisioning of IoT Devices", Internet Draft, https://datatracker.ietf.org/doc/draft-sweet-iot-acme/

[OPENID-BP] "OpenID Security Best Practices", http://wiki.openid.net/w/page/12995200/OpenID%20Security%20Best%20Practices

[RFC2567] F.D. Wright, “Design Goals for an Internet Printing Protocol”, RFC 2567, April 1999, https://datatracker.ietf.org/doc/html/rfc2567

[RFC7435] V. Dukhovni, "Opportunistic Security: Some Protection Most of the Time", RFC 7435, October 2014, https://datatracker.ietf.org/doc/html/rfc7435

[UTR17] Unicode Consortium “Unicode Character Encoding Model”, UTR#17, November 2022, https://www.unicode.org/reports/tr17

[UTR23] Unicode Consortium “Unicode Character Property Model”, UTR#23, November 2022, https://www.unicode.org/reports/tr23

[UTR33] Unicode Consortium “Unicode Conformance Model”, UTR#33, May 2025, https://www.unicode.org/reports/tr33

[UNISECFAQ] Unicode Consortium “Unicode Security FAQ”, https://www.unicode.org/faq/security.html

13. Authors

Authors:

Michael Sweet (Lakeside Robotics Corporation)
Piotr Pawliczek (Google)
Smith Kennedy (HP Inc.)

The authors would also like to thank the following individual for his contributions to this document:

Ira McDonald (High North)

