OFFICE OF

INFORMATION

AND TECHNOLOGY

# Identity and Access Management (IAM) Enterprise Design Pattern

OAuth 2.0 Authorization Code Grant

August 2018 | Demand Management Division

Table of Contents

1 Context 2

2 Challenge 2

3 Guidance 3

3.1 Authorization Code Grant 4

4 Application of Practices 6

4.1 OAuth 2.0 for Mobile Application Making API Call 6

4.1.1 Purpose 6

4.1.2 Assumptions 7

4.1.3 Use Case Description 7

4.2 Key Practices 8

5 Impact 9

Appendix: References 9

[Table 1: Change Matrix i](#_Toc526340756)

[Table 3: Key Practices IAM OAuth 2.0 Authorization Code Grant EDP 8](#_Toc526340757)

[Figure 1 - Overview of IAM Progression 3](#_Toc526340763)

[Figure 2 - Authorization Code Grant with PKCE Overview 5](#_Toc526340764)

Table 1: Change Matrix

| Version | Date | Description of Updates |
| --- | --- | --- |
| 1.0 |  | IAM EDP OAuth 2.0 Authorization Code Grant Segment document approved |

### Context

The Department of Veterans Affairs (VA) has a unified enterprise Identity and Access Management (IAM) Program that coordinates secure access to VA resources for both internal and external users. IAM services are guided by the Office of Management and Budget (OMB) M 11-11, the Federal Information Processing Standard (FIPS) 200, the National Institute of Standards and Technology (NIST) guidelines (800-63 and 800-53 per Appendix D), and the Federal Identity, Credential, and Access Management (FICAM) initiative.

VA has two general populations of users who require access: (1) internal users include employees, contractors, trainees and volunteers; and (2) external users, comprised of Veterans, beneficiaries, and health partners, including employees and contractors from other Government agencies. All require varying levels of access to interact with VA services.

### Challenge

The use of the OAuth 2.0 Authorization Framework (OAuth)[[1]](#footnote-1) is expanding at VA, along with the increased use of application programming interfaces (APIs). The improper implementation of OAuth has led to significant unauthorized access risks for services hosted by major commercial organizations.[[2]](#footnote-2) VA requires enterprise guidance on the design of OAuth to provide consistent security and limit risks. The area addressed in this document is the part of the IAM progression that is highlighted in red in Figure 1.

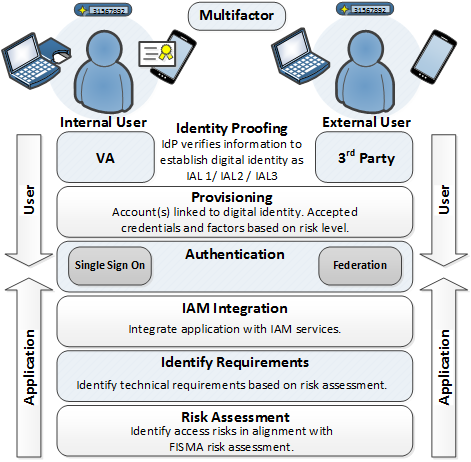
[[3]](#footnote-3)

Figure 1 - Overview of IAM Progression

### Guidance

As developers have migrated from WS-\*[[4]](#footnote-4) for building web services to APIs, OAuth adoption has steadily grown. OAuth is a delegated authorization framework, described under the Internet Engineering Task Force (IETF) Request for Comment (RFC) 6749.[[5]](#footnote-5) OAuth provides a convenient method for integrating with APIs to provide limited access to resources, without exposing user credentials. The user can select any approved credential service provider (CSP) to authenticate and gain access to the desired resource, without exposing the credentials to the resource provider. The CSP only passes a token to the resource provider.

#### Authorization Code Grant

Grants or “flows” are different methods of implementing the OAuth standard.[[6]](#footnote-6) The Authorization Code Grant, the most commonly used OAuth grant type, uses an authorization server as an intermediary between the client and resource owner.[[7]](#footnote-7) The use of the code provides additional protection over the Implicit Grant[[8]](#footnote-8) as it prevents the attacker from intercepting the access token. The authorization code is exposed to intercept, while the access token is sent over a secure channel. Client registration is required, which is outside the scope of OAuth. When registering a client, the developer is expected to register the client type (confidential or public), identify the redirection Uniform Resource Identifier (URI), and include any information required by the authorization server.[[9]](#footnote-9)

As interception of the authorization code is possible and exploits have now been identified, Proof Key for Code Exchange (PKCE)[[10]](#footnote-10) was defined in IETF RFC 7636[[11]](#footnote-11) for protection. In PKCE, a cryptographically random key is created that is called the code\_verifier. A transformation method (t\_m) is applied to the code\_verifier to create a code\_challenge. This can be thought of as the following: code\_challenge x t\_m = code verifier.

The code\_challenge and t\_m are sent to the authorization server in the initial request. Later, the client sends the code\_verifier with the authorization code. The authorization server checks to see if the code\_challenge x t\_m from the first step = code\_verifier was received later. This prevents a Man in the Middle (MITM) attack from using an intercepted authorization code. The code\_verifier must be unique for each request. As described in the RFC authorization code, MITM attacks have been observed in the wild. As PKCE has become a common protection, the Authorization Code Grant is shown with PKCE applied, as shown highlighted in green in Figure 2.

[[12]](#footnote-12)

Figure 2 - Authorization Code Grant with PKCE Overview

1. The client application initiates the flow when authorization is required to access a resource. The client creates and records a secret (code\_verifier) and derives a transformed version of that (code\_challenge), and a transformation method (t\_m).
2. The initial authorization request contains the client identifier, requested scope, local state, code\_challenge, t\_m, and the redirection URI, to which the authorization server will send the user-agent. Note that the redirection URI must have been registered with the authorization server in advance, using the client id/secret. The client redirects the resource owner’s user agent to the authorization server with the authorization request. The user is required to authenticate to the authorization server. The authorization server may authenticate the user directly or by using a service (both require VA approval). Upon being presented with an authenticated user, the authorization service must ensure that requested scopes are authorized by the resource owner by presenting a list of requested scopes and the option to approve or deny the request. The scope(s) are defined by the API or web application. If the user approves, the authorization server will include the requested scope(s) when the access token is generated (the source of data for step 5).
3. The authorization server records the code\_challenge and the t\_m, then redirects the user-agent back to the client, using the redirection URI, provided along with the authorization code.
4. The client requests an access token using the authorization code, the code\_verifier, and redirection URI.
5. The authorization server transforms the code\_challenge using t\_m, and verifies that the resulting code\_verifier matches the code\_verifier sent by the client. The authorization code and redirection URI are also verified. The authorization server responds with an access token, and optionally, a refresh token.
6. The client application presents the access token to the resource owner, who determines which resources are accessed.

### Application of Practices

The following use case relates to the application of the described risk management principles to solution development.

#### OAuth 2.0 for Mobile Application Making API Call

### Purpose

The Veterans Health Administration (VHA) would like to release a mobile application that uses OAuth to allow Veterans to access their electronic health records (EHRs). The system owner wants to determine the technical requirements for IAM integration for the API.

### Assumptions

* The API is accessing sensitive information, including protected health information (PHI).
* A native mobile application is a possible choice.
* Any external credentials that are used are approved for use by VA.

### Use Case Description

* The system owner begins the intake process for the Veteran-focused Integration Process (VIP)[[13]](#footnote-13) by submitting the business case for a new solution as a VIP Request (VIPR).[[14]](#footnote-14)
* The solution is targeting mobile devices and use of VA API services to access EHRs for Veterans, using their own credentials.
* OAuth 2.0 is identified as the best method to authorize access to the VA data.
* A risk assessment is performed due to access to PHI. The system owner decides that the OAuth 2.0 Client Authorization Code Grant with PKCE provides the appropriate level of security for the purpose.
* The application owner contacts VA IAM to register their application with the VA-approved authorization server. The OAuth configuration is updated with the provided information.
* Authentication is provided by the approved CSP and authorization is provided by VA IAM authorization services.

#### Key Practices

The following table highlights key practices identified in this EDP.

Table 3: Key Practices IAM OAuth 2.0 Authorization Code Grant EDP

| **Category** | **Area** | **Description** |
| --- | --- | --- |
| **Identity and Access Management** | Delegated Authorization | OAuth 2.0 Client Authentication   * Confidential applications - Authorization header using Hypertext Transfer Protocol (HTTP) - basic authentication is required, where the username is the application’s client\_id and the password is the application’s client\_secret. |
| **Identity and Access Management** | Delegated Authorization | OAuth 2.0 Grant Types   * The Authorization Code Grant must only be used with applications that can protect a secret and must be used in conjunction with PKCE. |
| **Identity and Access Management** | Delegated Authorization | OAuth 2.0 Client Secret   * Authorization Code Grant must use installation-specific client secrets with at least 256 bits of entropy. |
| **Identity and Access Management** | Delegated Authorization | OAuth 2.0 Token Expiration  The listed time (in seconds) must be the maximum expiration time for the specified resource type:   * Authorization Code: 60s |
| **Identity and Access Management** | Delegated Authorization | OAuth 2.0 Token Integrity   * A refresh token must be bound to the same client\_id and must contain the same, or a subset of, the set of claims authorized for the access token with which it is associated. |

### Impact

If risk management is not used to define technical requirements for IAM components of VA solutions, inadequate technical protections for sensitive data may contribute to unauthorized access or data breach.

### Appendix: References

* DEA User Stories: <https://vaww.portal2.va.gov/sites/asd/TechStrat/IPTS/SitePages/Home.aspx>
* FISMA User Stories: <https://vaww.portal2.va.gov/sites/asd/AERB/FISMASecurityCompliance/SitePages/Home.aspx>
* TRM: <http://trm.oit.va.gov/>
* NIST 800-63-3: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63-3.pdf>
* OAuth 2.0 Framework: <https://tools.ietf.org/html/rfc6749#section-1.2>
* OAuth 2.0 Threat Model: <https://tools.ietf.org/html/rfc6819#page-16>
* OAuth 2.0 PKCE: <https://tools.ietf.org/html/rfc7636>
* SMART Application Authorization Guide: <http://docs.smarthealthit.org/authorization/>
* VA 6500.3: <http://vaww.va.gov/vapubs/viewPublication.asp?Pub_ID=733&FType=2>
* VA 6510 (under revision): <http://vaww.va.gov/vapubs/viewPublication.asp?Pub_ID=823&FType=2>

**Disclaimer:** This document serves both internal and external customers. Links displayed throughout this document may not be viewable to all users outside the VA domain. This document may also include links to websites outside VA control and jurisdiction. VA is not responsible for the privacy practices or the content of non-VA websites. We encourage you to review the privacy policy or terms and conditions of those sites to fully understand what information is collected and how it is used.

**Statement of Endorsement:** Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, and must not be used for advertising or product endorsement purposes.

1. In this document, OAuth refers to the OAuth 2.0 Authorization Framework, developed in 2012, as the open-standard authorization protocol that describes how unrelated servers and services can safely allow authenticated access to assets, without sharing the initial, related, single logon credential. It is used as a secure, third-party, user-agent, delegated authorization (Source: https://www.csoonline.com/article/3216404/authentication/what-is-oauth-how-the-open-authorization-framework-works.html). [↑](#footnote-ref-1)
2. Source: https://www.csoonline.com/article/3194727/security/google-docs-phishing-attack-underscores-oauth-security-risks.html. [↑](#footnote-ref-2)
3. Figure 1 was created by the VA Office of Information and Technology (OIT) Architecture and Engineering Service (AES) Enterprise Design Pattern (EDP) Team from information obtained from VA OIT IAM Subject Matter Experts (SMEs) and the National Institute of Standards and Technology (NIST) Special Publication 800-63A at https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63-3.pdf. [↑](#footnote-ref-3)
4. WS-\*is a prefix used to indicate specifications associated with web services and there exist many WS\* standards including WS-Addressing, WS-Discovery, WS-Federation, WS-Policy, WS-Security, and WS-Trust. For additional information, refer to https://www.dotnettricks.com/learn/webservice/understanding-ws-star-standards-and-specifications. [↑](#footnote-ref-4)
5. Reference the IETF at https://www.ietf.org/rfc/rfc6749.txt. [↑](#footnote-ref-5)
6. For more information on grants, see the OAuth Security Primer Enterprise Design Pattern [↑](#footnote-ref-6)
7. Reference the Authorization Code Grant at IETF at https://tools.ietf.org/html/rfc6749#section-1.3.1. [↑](#footnote-ref-7)
8. Reference the Implicit Code Grant at IETF at https://tools.ietf.org/html/rfc6749?#section-1.3.2. [↑](#footnote-ref-8)
9. The Trusted Exchange Framework and Common Agreement (TEFCA) recommends OAuth 2.0 Dynamic Client Registration Protocol. Refer to https://tools.ietf.org/html/draft-ietf-oauth-dyn-reg). [↑](#footnote-ref-9)
10. PKCE is pronounced "pixy.” [↑](#footnote-ref-10)
11. Reference the IETF RFC 7636 at https://tools.ietf.org/html/rfc7636. [↑](#footnote-ref-11)
12. Figure 2 was created by the VA Office of Information and Technology (OIT) Architecture and Engineering Service (AES) Enterprise Design Pattern (EDP) Team from information obtained from VA OIT IAM Subject Matter Experts (SMEs).and the Internet Engineering Task Force (IETF) Request for Comments 6749 at https://tools.ietf.org/html/rfc6749. [↑](#footnote-ref-12)
13. Source: The Veteran-focused Integration Process (VIP) Guide 3.1, April 2018, at https://www.voa.va.gov/DocumentView.aspx?DocumentID=4371. [↑](#footnote-ref-13)
14. Reference the VIPR Request Portal at https://vaww.vashare.oit.va.gov/sites/dmo/VIPR/SitePages/VIPR%20Home%20Page.aspx. [↑](#footnote-ref-14)