

WHITE PAPER

Application authentication In Helse Vest

Version 1.0

English Edition



Innhold

Version.....	2
Summary	3
Introduction.....	3
Target audience	3
Authentication -/ Identity provider in Helse Vest.....	4
Required technical standards we support.....	5
Access	5
Authentication.....	5
Provisioning	5
Standards that are not supported or desired.....	5
Authentication Transition Phase.....	6
Overall requirements for applications.....	7
Implementation guidance	8
Concepts and explanation	9

Version

This document is version 1.0. The document will be revised annually by the IAM team of Helse Vest IKT.

Summary

Helse Vest wants to deliver effective identity and access management capabilities, security and reduce security risk.

We want the applications to use OAuth 2.0, OpenID Connect, JWTs and SCIM 2.0 to avoid vendor lock-in and balance security, privacy, usability, and scale when building and deploying applications and services.

Introduction

This White Paper provides a technical overview and some insights into the technologies that Helse Vest wants applications or service solutions to support and use. The purpose is to help purchasers, IT managers, vendors, and developers to understand the technical authentication demands of Helse Vest. This White Paper does not describe how applications should implement authentication, but what authentication standards we support, and guidance for how they can be implemented to fulfill our demands.

We focus on modern authentication in mobile, web and desktop applications.

This document will be reviewed yearly by the IAM team in Helse Vest.

Target audience

If you want an application to be implemented in Helse Vest, there are some rules to follow. If you're in the process of buying an application, you're a vendor or a team/person developing mobile, web or desktop application that will be integrated or installed in Helse Vest, local on-prem environment or in the public Cloud, you should read this document.

It gives you an overview of the demands and requirements your application should support when dealing with authentication.

This document describes the requirements and guidance to how FIDO2 authentication can be implemented in an application.

Authentication-/ Identity provider in Helse Vest

Microsoft Entra ID is Helse Vest's Identity provider. Applications running in Helse Vest's infrastructure must authenticate against Microsoft Entra ID. FIDO2(Fast Identity Online 2.0) is our preferred security solution regarding authentication. FIDO2 communicates with Microsoft Entra ID.

Application must make use of our preferred standard and protocols when authenticating, [see chapter](#). The application must handle the authorization in the application after a valid authentication is done against Microsoft Entra ID.

Helse Vest's vision is to become passwordless. Additionally, in accordance with information security standards, we mandate the use of two-factor authentication as a minimum requirement for accessing health and personal information, as well as for logging into mobile devices. In future, this will also include a PC or other device connected to Helse Vest's infrastructure.

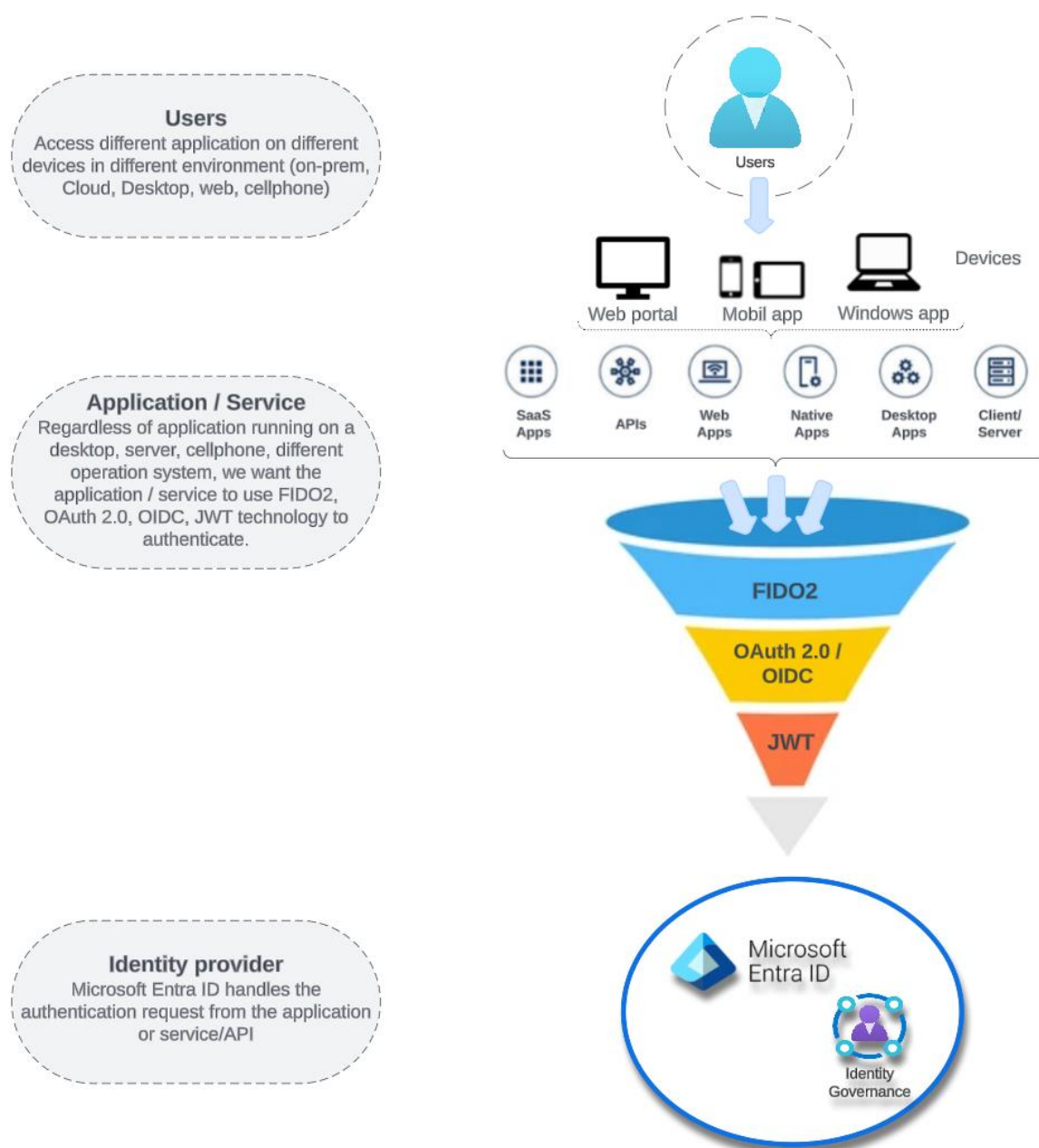


Figure 1 - Simplified concept sketch of authentication

Required technical standards we support

The following technical standards we support in Helse Vest is:

Access

- JSON Web Tokens (JWT)
- OAuth 2.0 / OpenID Connect

Authentication

- FIDO2 – Fast Identity Online 2

Provisioning

- System for cross-domain identity Management (SCIM 2.0)

Standards that are not supported

- Authentication to Local Active Directory
 - Kerberos
 - NTLM
 - LDAP
- Authentication against local user database implemented in the application.

INFO! *For applications that have this implemented, Helse Vest IKT wishes that in the future, our supported standards are implemented.*

Authentication Transition Phase

Helse Vest is planning a digital transition in authentication. This means that we are standardizing on technical authentication standards, as [referenced this chapter](#). During this transition phase, we want applications to be able to support their existing authentication solution as well as the new FIDO2 security key authentication. Therefore, we do not want to remove the possibility of existing authentication but rather offer the new FIDO2 in addition during this transition.

During this transition some employees will become password less, while others will still need to use passwords due to varying implementation speeds among employees and application.

Logon scenario:

1. User starts the application.
2. They can choose between logging in the usual way as they do today or selecting a different identity provider (in our case, Microsoft Entra ID).
3. Regular login will remain as before, while for the new solution, users will have to choose the account used by Helse Vest for logging in to Microsoft Entra ID and select the security key.

***Note!** The following must not be treated as a “solution requirements specification”, but an example of an authentication with old and new authentication method.*

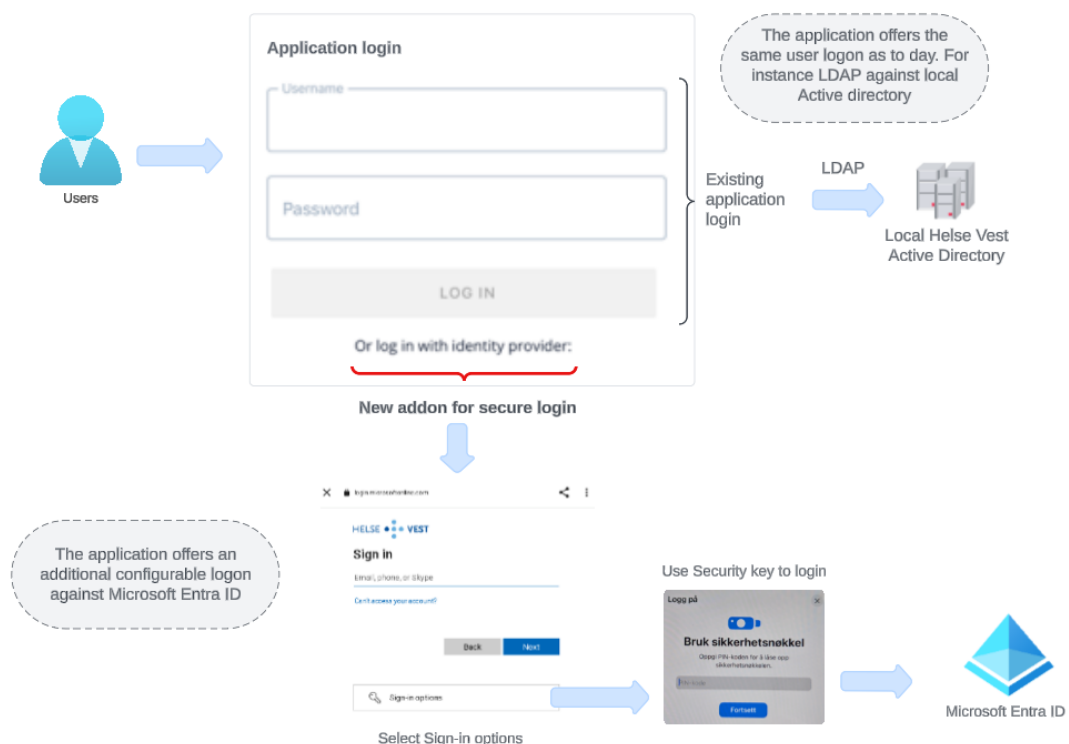


Figure 2 Example of authentication in an application with FIDO2 and current login.

Overall requirements for applications

Requirement	Details
New applications must support authentication against <u>Microsoft Entra ID</u> . This requirement applies regardless of the platform on which the application runs	Microsoft Entra ID is Helse Vest's identity provider and authentication service. The application must be registered in Entra ID and could use the Microsoft Authentication Library (MSAL) for handling authentication, or another Microsoft-supported library, choose one with a certified OpenID Connect implementation .
New applications must support the <u>OpenID Connect (OIDC)</u> protocol.	OpenID Connect is a standard that ensures secure and flexible identity management across applications.
New applications must support authentication through <u>FIDO2</u> .	FIDO2 is a phishing-resistant MFA (Multi-Factor Authentication) authentication based on cryptographic keys and is a standard that provides secure, user-friendly, and platform-independent authentication.
All applications should <u>facilitate Single Sign-On (SSO)</u> from device login.	Applications configured with OpenID Connect will be able to implement SSO from device login relatively easily. Applications installed on the device can facilitate SSO through Entra Connect.
All applications should use <u>PKCE (Proof Key for Code Exchange)</u> to enhance authentication security	PKCE helps to prevent attacks where a malicious actor can intercept the authorization code and use it to obtain an access token.
Existing applications that authenticate with username and password against on-premises Active Directory should also support FIDO2.	On-premises Active Directory now functions as a secondary authentication service. Entra ID supports older authentication technologies so that employees can log in with FIDO2, but the application must allow both password/username and FIDO2 until everyone has obtained FIDO2. This assumes that password authentication goes against on-premises Active Directory.

Implementation guidance




A prerequisite for any service or application seeking authentication through Microsoft Entra ID is to register access by first registering the application in Microsoft Entra ID. This is done by contacting “Team Skytjenester”. Rules and process for this are owned by this group.

Helse Vest has developed code examples in .NET demonstrating how to achieve modern authentication for:

- Web applications
- Mobile applications
- Desktop/Windows applications

The code examples are published and available on Helse Vest's GitHub here.

[GitHub - HelseVestIKT/HVI-UTV-EntraEksempler: Eksempler for implementasjon av AD/Entra-pålogging for Web, Flutter, og Desktop.](#)

 angular	angular SPA eksempel
 desktop/EntraDemo	Desktop eksempel
 flutter/hviktentra	angular SPA eksempel

We have divided the examples into three, reflecting different types of applications.

We remind you that these are examples, and other tools and components can be used to achieve the same.

Important! Each individual supplier/team, or developer, is responsible for the security of the application. The examples referenced do not specify the applicable security level. In addition to strong authentication, the application must have good access control and authorization control.

Concepts and explanation

- **OAuth 2.0:** Designed to delegate access to applications and services and to protect APIs. It enables any type of application (i.e., OAuth 2.0 client), such as a native application, a JavaScript-based application running in a web browser, an internal or external service or a web server application to prove that they are authorized to access services. There may be a user present or the client could be acting on its own behalf. OAuth 2.0 is a first-class identity protocol, and it underpins many other modern identity protocols.
- **OpenID Connect:** SAML 2.0 has long been the protocol of choice for cross-domain authentication; however, implementations have lacked support for easy establishment of trust across systems and support for mobile and API use cases. SAML also uses complex XML-based assertions that can be challenging to parse and validate. OIDC builds on OAuth 2.0, but the two don't solve the same use cases. Compared with OAuth 2.0, as described above, OIDC enables cross-domain authentication and it's the modern equivalent to SAML 2.0, which improves on the shortcomings of SAML.
- **SCIM 2.0:** OIDC and OAuth 2.0 are access control mechanisms for applications. SCIM 2.0 on the other hand enables the handling of automated identity life cycle management. The SCIM 2.0 standard is built to help onboard, update and decommission identities in target applications that's deployed on-premises and in the cloud.
- **JWT:** A JavaScript Object Notation (JSON) Web Token is now the de facto credential used when accessing APIs and services. Applications acquire JWTs with the help of an OAuth 2.0 flow and APIs and services validate and authorize access with the help of the JWT. JWTs are signed and, potentially, encrypted JSON documents that include claims that can be used as a basis for authorization decisions.
- **SaaS apps:** Software as a service (SaaS) allows users to connect to and use cloud-based apps over the Internet.
- **APIs:** API stands for application programming interface, which is a set of definitions and protocols for building and integrating application software
- **Single Page Apps:** A single page application is a website or web application that dynamically rewrites a current web page with new data from the web server, instead of the default method of a web browser loading entire new pages
- **Web Apps:** is an application program that is stored on a remote server and delivered over the internet through a browser interface
- **Native Apps:** The term native app suggests an app you can download and install on your device. A native mobile app is developed specifically for a mobile device
- **Desktop Apps:** Desktop applications are software programs run locally on computer devices. They aren't accessible from a browser, like web-based apps, and require deployment on a personal computer or laptop
- **Client / Server:** Client-server denotes a relationship between cooperating programs in an application, composed of clients initiating requests for services and servers providing that function or service.