ManageEngine

FIDO2 authentication

with **ADSelfService Plus**

www.manageengine.com

FIDO2 authentication with ADSelfService Plus

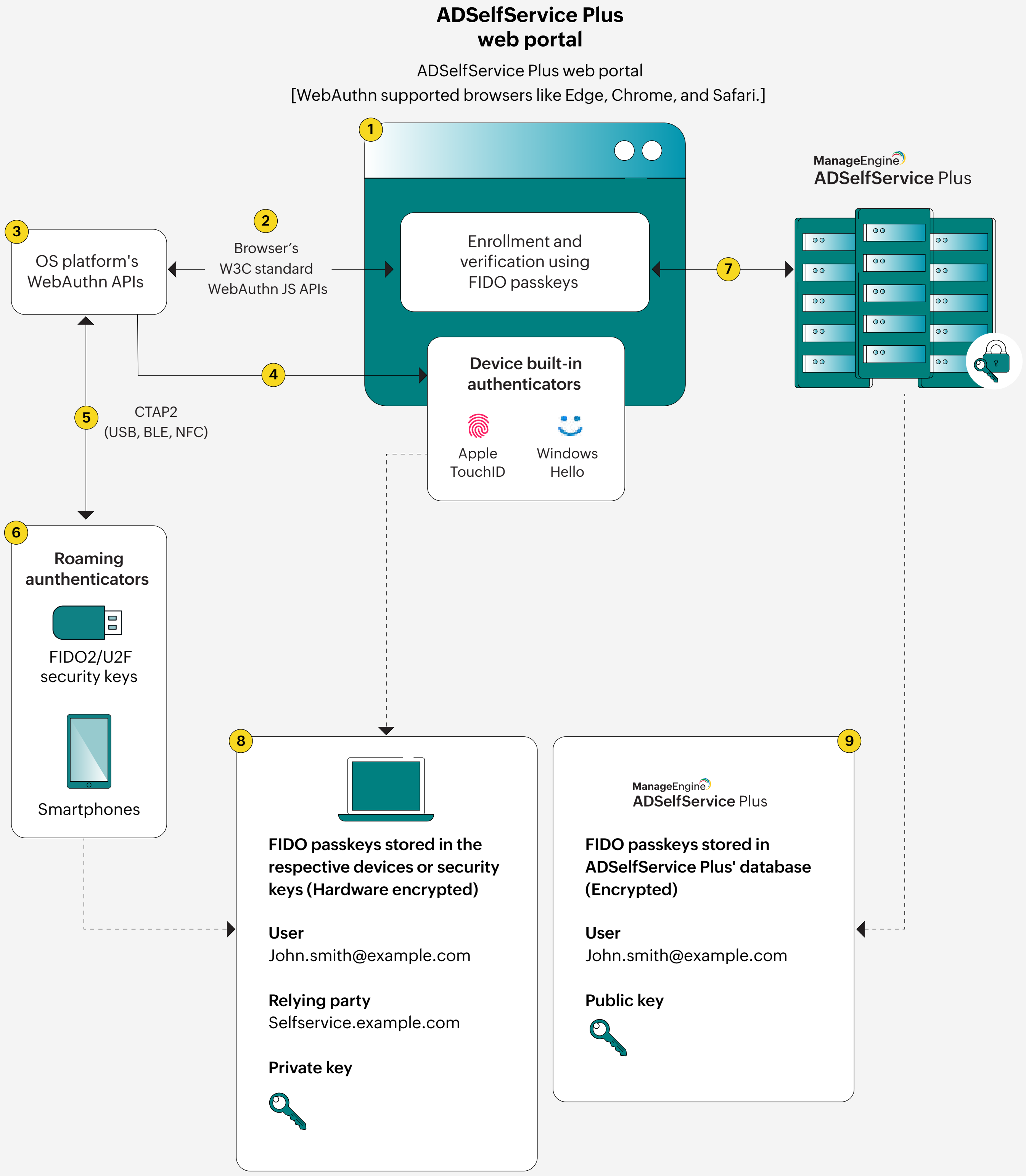
The FIDO Passkeys authenticator in ADSelfService Plus implements FIDO2 authentication via WebAuthn API and public key cryptography to provide passwordless, phishing-resistant MFA. Using FIDO Passkeys, you can secure end-user access to cloud applications, OWA, and self-service actions performed using ADSelfService Plus' web portal.

How FIDO2 authentication works

FIDO2 authentication employs public key cryptography, where each passkey or FIDO credential is represented by a combination of a public and private key. The private key is stored securely using hardware encryption on the user's device, which may be a computer or mobile device, or a security key. The public key is stored securely by ADSelfService Plus corresponding to the user and their respective enrolled device.

During authentication, when the user successfully verifies their identity on their enrolled device using a PIN, an OTP, or biometric information, assertion data is generated by the device and communicated to ADSelfService Plus. This assertion data includes a digital signature encrypted with the user's private key. ADSelfService Plus then verifies the assertion data using the public key stored in its database corresponding to that particular user, and proceeds with granting or revoking access.

Architectural flow of FIDO2 authentication



ADSelfService Plus web portal

ADSelfService Plus web portal
[WebAuthn supported browsers like Edge, Chrome, and Safari.]

1

Enrollment and verification using FIDO passkeys

Device built-in authenticators

Apple TouchID

Windows Hello

2 Browser's W3C standard WebAuthn JS APIs

3 OS platform's WebAuthn APIs

4

5 CTAP2 (USB, BLE, NFC)

6 **Roaming aunthenticators**

FIDO2/U2F security keys

Smartphones

7

ManageEngine ADSelfService Plus

8 **FIDO passkeys stored in the respective devices or security keys (Hardware encrypted)**

User
John.smith@example.com

Relying party
Selfservice.example.com

Private key

9 ManageEngine ADSelfService Plus

FIDO passkeys stored in ADSelfService Plus' database (Encrypted)

User
John.smith@example.com

Public key

Below is the architectural flow for end-user FIDO2 enrollment and authentication using ADSelfService Plus. Users first need to enroll in the FIDO Passkeys authenticator to authenticate using FIDO2 MFA.

1. ADSelfService Plus server initiates the FIDO2 enrollment and authentication processes via the ADSelfService Plus web portal.

2. The ADSelfService Plus web portal triggers the WebAuthn API of the end user's web browser by requesting the creation of a FIDO passkey for the user with ADSelfService Plus' relying party ID, for instance, *selfservice.example.com.*

> *The **WebAuthn API** includes JavaScript APIs that act as an interface between a user's device-native authenticators or security keys and the ADSelfService Plus server.*

3. The browser's WebAuthn API interacts with the device's WebAuthn API, also known as the platform WebAuthn API, to initiate verification.

4. For device-native or platform authenticators, like Windows Hello, Apple TouchID, and Android Biometrics, the platform WebAuthn API interacts directly with the authenticator and initiates verification.

5. For roaming authenticators, like FIDO2- and U2F-compliant security keys such as Precision's InnaIT^{Key}, YubiKey, and Google Titan, the platform WebAuthn API interacts with the roaming device via the CTAP2 protocol.

> *The **CTAP2 protocol** standardizes methods to communicate with roaming authenticators by establishing a confidential and mutually-authenticated data transport channel, such as Bluetooth, NFC, or USB.*

6. User identity verification happens on the respective device using authenticators such as biometric information, a PIN, or an OTP.

7. After successful identity verification, the response is sent back to the WebAuthn API and back to the ADSelfService Plus server.

8. In case of end-user enrollment, the FIDO passkey, i.e., the private key is created on the user's device, and the enrollment data, i.e., the public key and other necessary data, is sent back to ADSelfService Plus.

9. In case of identity verification, the assertion data is sent back to ADSelfService Plus where the signature is verified using the respective passkey's public key.

ManageEngine

ManageEngine is the enterprise IT division of Zoho Corporation. More than 30,000 organizations from different verticals, industries, and sizes use ManageEngine to seamlessly manage their IT management needs. ManageEngine serves as a reliable, long-term partner that has been in the IT management arena since 2002 and will be in that arena as long as IT remains relevant to you and your business.