Using Passkeys for Domino Web Authentication

Passkeys are being heralded as the future for web authentication, supported by Domino and modern web browsers.

But what are they and why should you consider using them?

Paul Harrison

Version 1.0.0, 17th October 2024

About Paul Harrison

- Developer at FoCul, focusing on Front-end development with Angular
- Over 20 years experience with HCL Notes/Domino - everything from support and administration, to infrastructure, migrations and development

HCLSoftware
HCL Ambassador
2024

openntf
Webinar Presenter
2023

Email: paul.harrison@focul.net

TwitterX: @PaulHarrison

Introduction

Agenda

- Passwords vs. Passkeys
- Passkey Client Demo
- Domino Passkey Setup
- How Passkeys Work
- Passkey Eco-Systems
- Virtual Authenticators for Testing & Troubleshooting
- Summary

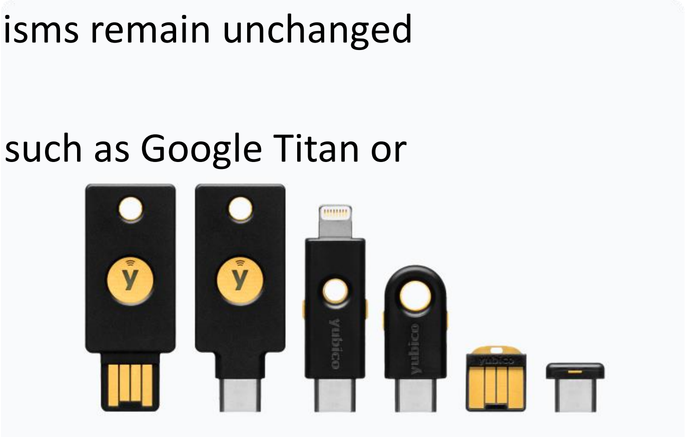
Passwords vs. Passkeys

What’s Wrong With Passwords?

- Often simple guessable passwords - password, passw0rd, 123456 etc.
- Very often shared between multiple accounts and services
- Subject to Phishing and MITM attacks
- Passwords saved on webservers - potential risk of data breach
- Susceptible to dictionary and brute force attacks
- Complex or rule-based passwords almost discourage passwords changing because they are often difficult to create, remember and use
- Multi-factor authentication (MFA) is a significant improvement, but is still vulnerable to Phishing and MITM attacks, plus also mailbox and SIM swapping attacks

Why Are Passkeys Better?

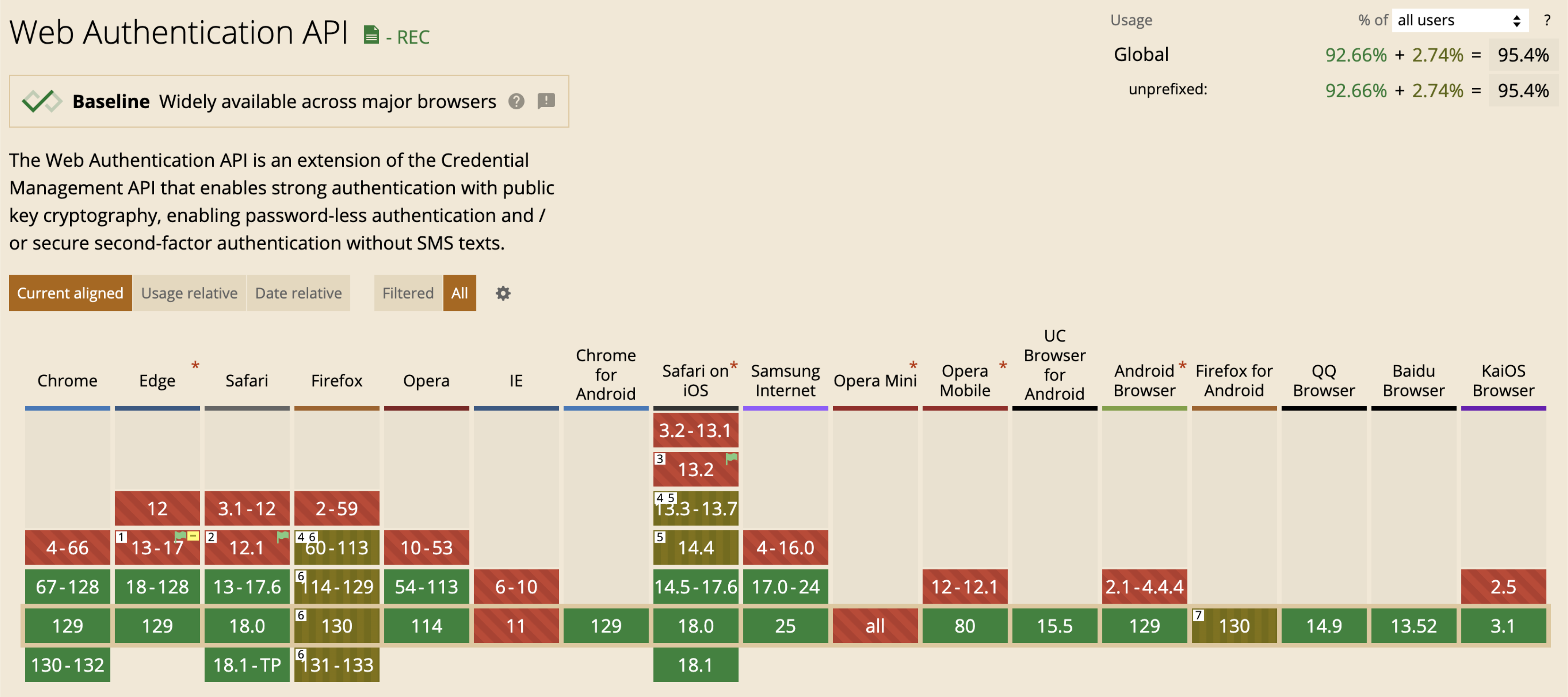
- Standards based – The FIDO2 specification comprise of the World Wide Web Consortium’s (W3C) Web Authentication (**WebAuthn**) specification, and FIDO Alliance’s corresponding Client-to-Authenticator Protocol (**CTAP**), which together provides for a strong authentication experience
- Uses Public Key Cryptography to create a unique asymmetric key pair for each account/website on a device. These keys work to verify a user's identity and grant access to a website:
  - Private key - securely stored on the user's device, in their credentials or password manager, and is used to sign website generated challenges, when attempting to access an account
  - Public key – shared with, and stored on the website during registration, and is used to verify a client's signed challenge
- Improved usability and more streamlined experience than MFA, using the same simple action that users are very familiar with, and use multiple multiple times each day (biometrics/PIN)
- Passkeys only validate your identity, the underlying authentication and authorisation mechanisms remain unchanged (Cookies, JWT tokens etc.)
- Authenticators can be either standalone physical devices (e.g. USB/NFC/Bluetooth FIDO keys such as Google Titan or Yubico keys), or synced between a user's devices
- Passwords may still be required for initial Passkey registration and for Passkey recovery



Security Benefits

- Passkeys are specific to users/website by design – prevents Phishing via fake websites
- One-time, time-bound challenges are exchanged during registration and login – preventing MITM attacks
- No passwords are stored on the website – reduces impact of data breaches

Browser Availability



Web Authentication API - REC

Baseline Widely available across major browsers

Usage % of all users ?

Global	92.66% + 2.74% =	95.4%
unprefixed:	92.66% + 2.74% =	95.4%

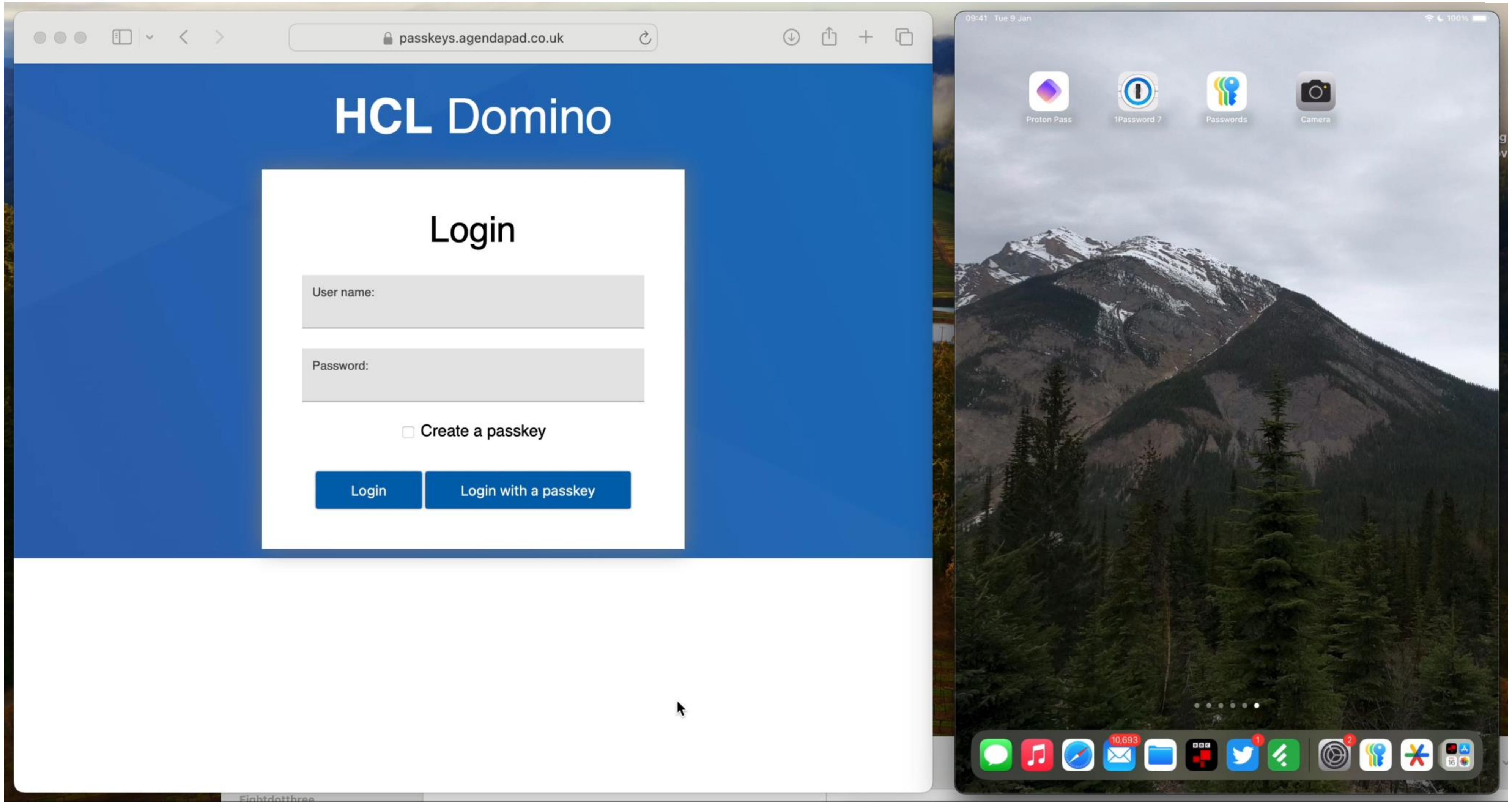
The Web Authentication API is an extension of the Credential Management API that enables strong authentication with public key cryptography, enabling password-less authentication and / or secure second-factor authentication without SMS texts.

Current aligned | Usage relative | Date relative | Filtered | All

Chrome	Edge *	Safari	Firefox	Opera	IE	Chrome for Android	Safari on iOS *	Samsung Internet	Opera Mini *	Opera Mobile *	UC Browser for Android	Android Browser *	Firefox for Android	QQ Browser	Baidu Browser	KaiOS Browser
							3.2-13.1									
							[3] 13.2									
	12	3.1-12	2-59				[4] [5] 13.3-13.7									
4-66	[1] 13-17	[2] 12.1	[4] [6] 60-113	10-53			[5] 14.4	4-16.0								
67-128	18-128	13-17.6	[6] 114-129	54-113	6-10		14.5-17.6	17.0-24		12-12.1		2.1-4.4.4				2.5
129	129	18.0	[6] 130	114	11	129	18.0	25	all	80	15.5	129	[7] 130	14.9	13.52	3.1
130-132		18.1-TP	[6] 131-133				18.1									

Attribution: https://caniuse.com/webauthn

Passkey Client Demo



passkeys.agendapad.co.uk

HCL Domino

Login

User name:

Password:

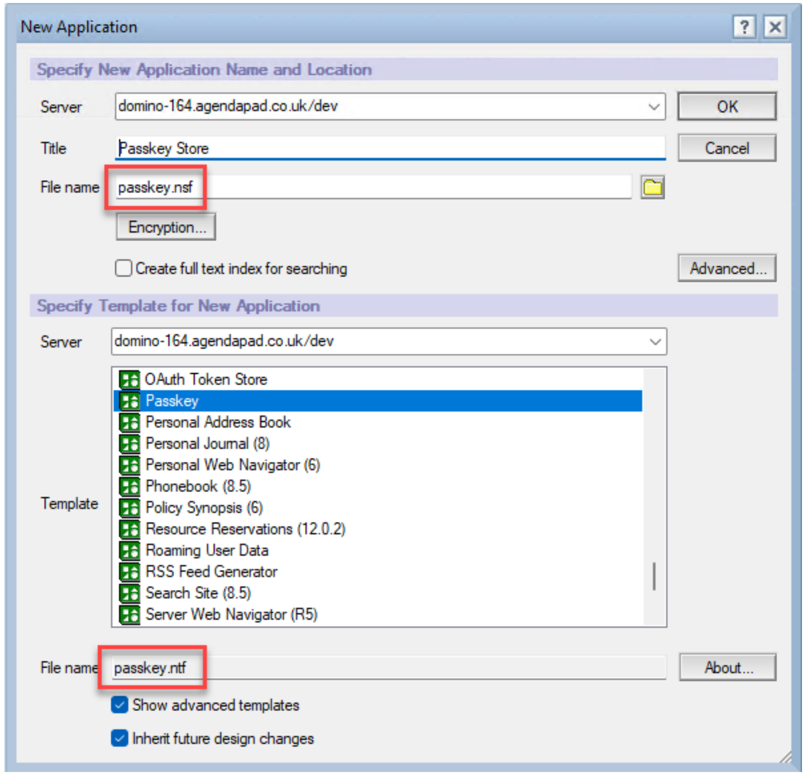
□ Create a passkey

Domino Passkey Setup

Domino Passkey Setup Demo

Domino Passkey Setup (1/4)

- Create Passkey database
  - Must reside in root of the Domino Data directory
  - Filename must be “passkey.nsf” (all lowercase).
    Optionally the filename and location can be changed by a notes.ini option
  - Replicate to cluster partners
  - Ensure strong ACLs (particularly write access)



New Application

Specify New Application Name and Location

Server: domino-164.agendapad.co.uk/dev

Title: Passkey Store

File name: passkey.nsf

☐ Create full text index for searching

Specify Template for New Application

Server: domino-164.agendapad.co.uk/dev

Template:
- OAuth Token Store
- Passkey
- Personal Address Book
- Personal Journal (8)
- Personal Web Navigator (6)
- Phonebook (8.5)
- Policy Synopsis (6)
- Resource Reservations (12.0.2)
- Roaming User Data
- RSS Feed Generator
- Search Site (8.5)
- Server Web Navigator (R5)

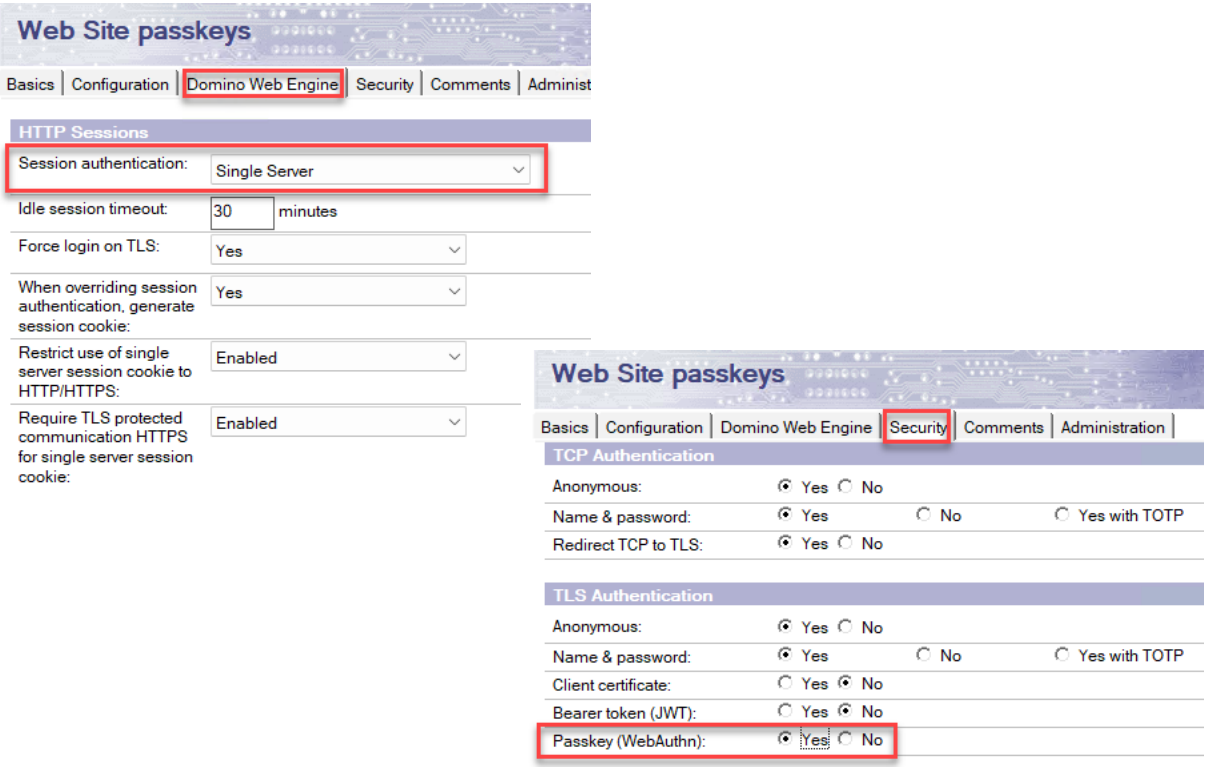
File name: passkey.ntf

☑ Show advanced templates

☑ Inherit future design changes

Domino Passkey Setup (2/4)

- Enable in Internet Site
  - Ensure Session authentication to either “Singler Server” or “Multiple Servers (SSO)”
  - Enable “Passkey (WebAuthn)” in TLS Authentication



Web Site passkeys

Basics | Configuration | Domino Web Engine | Security | Comments | Administ

HTTP Sessions

Session authentication:	Single Server
Idle session timeout:	30 minutes
Force login on TLS:	Yes
When overriding session authentication, generate session cookie:	Yes
Restrict use of single server session cookie to HTTP/HTTPS:	Enabled
Require TLS protected communication HTTPS for single server session cookie:	Enabled

Web Site passkeys

Basics | Configuration | Domino Web Engine | Security | Comments | Administration |

TCP Authentication

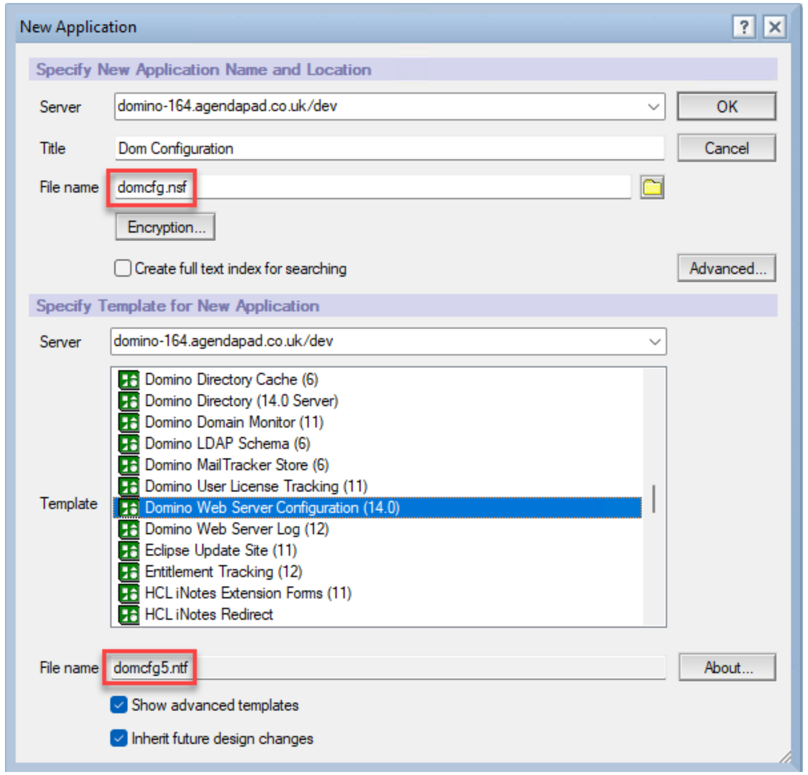
Anonymous:	● Yes ○ No		
Name & password:	● Yes	○ No	○ Yes with TOTP
Redirect TCP to TLS:	● Yes ○ No		

TLS Authentication

Anonymous:	● Yes ○ No		
Name & password:	● Yes	○ No	○ Yes with TOTP
Client certificate:	○ Yes ● No		
Bearer token (JWT):	○ Yes ● No		
Passkey (WebAuthn):	● Yes ○ No		

Domino Passkey Setup (3/4)

- Update login template (optional, but recommended)



New Application

Specify New Application Name and Location

Server	domino-164.agendapad.co.uk/dev
Title	Dom Configuration
File name	domcfg.nsf

□ Create full text index for searching

Specify Template for New Application

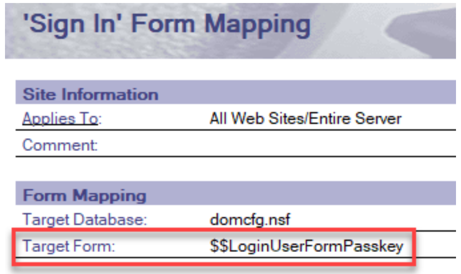
Server: domino-164.agendapad.co.uk/dev

Template:
- Domino Directory Cache (6)
- Domino Directory (14.0 Server)
- Domino Domain Monitor (11)
- Domino LDAP Schema (6)
- Domino MailTracker Store (6)
- Domino User License Tracking (11)
- Domino Web Server Configuration (14.0)
- Domino Web Server Log (12)
- Eclipse Update Site (11)
- Entitlement Tracking (12)
- HCL iNotes Extension Forms (11)
- HCL iNotes Redirect

File name: domcfg5.ntf

☑ Show advanced templates

☑ Inherit future design changes



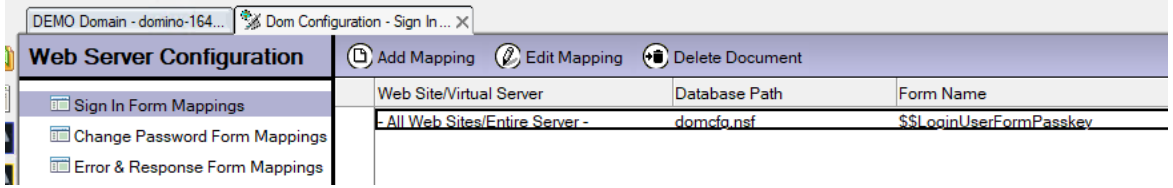
'Sign In' Form Mapping

Site Information

Applies To:	All Web Sites/Entire Server
Comment:	

Form Mapping

Target Database:	domcfg.nsf
Target Form:	$$LoginUserFormPasskey



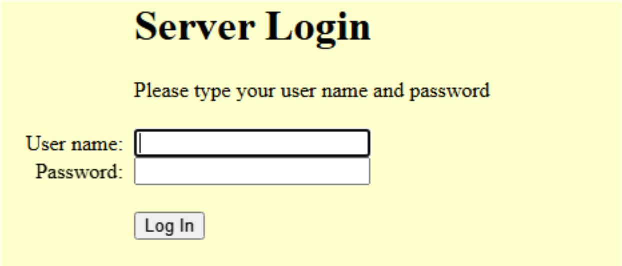
Web Server Configuration

- Sign In Form Mappings
- Change Password Form Mappings
- Error & Response Form Mappings

Web Site/Virtual Server	Database Path	Form Name
- All Web Sites/Entire Server -	domcfg.nsf	$$LoginUserFormPasskey

Domino Passkey Setup (4/4)

Default (Without Passkeys)



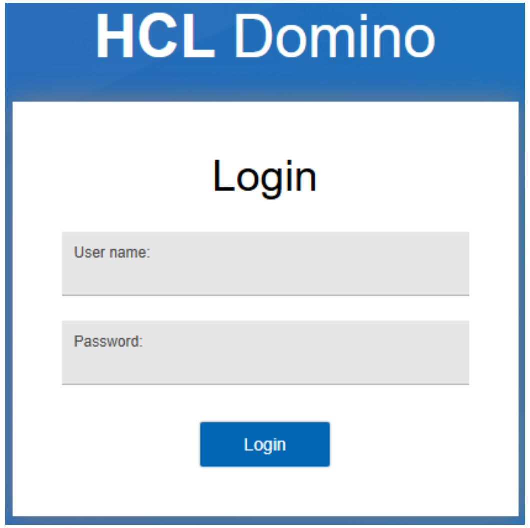
Server Login

Please type your user name and password

User name:

Password:

Log In



HCL Domino

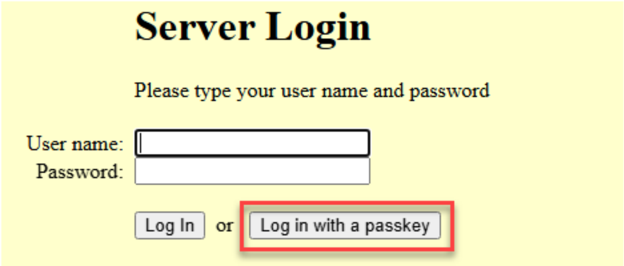
Login

User name:

Password:

Login

With Passkeys



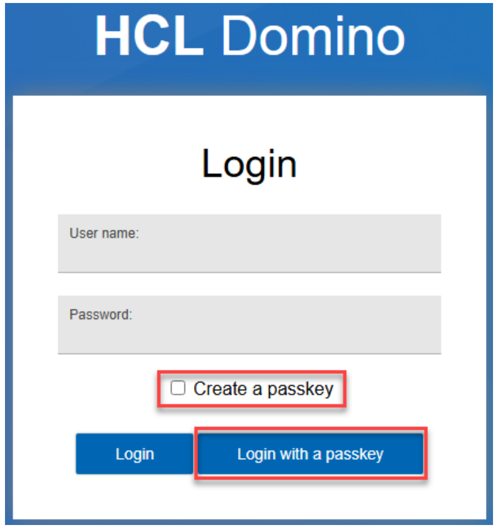
Server Login

Please type your user name and password

User name:

Password:

Log In or Log in with a passkey



HCL Domino

Login

User name:

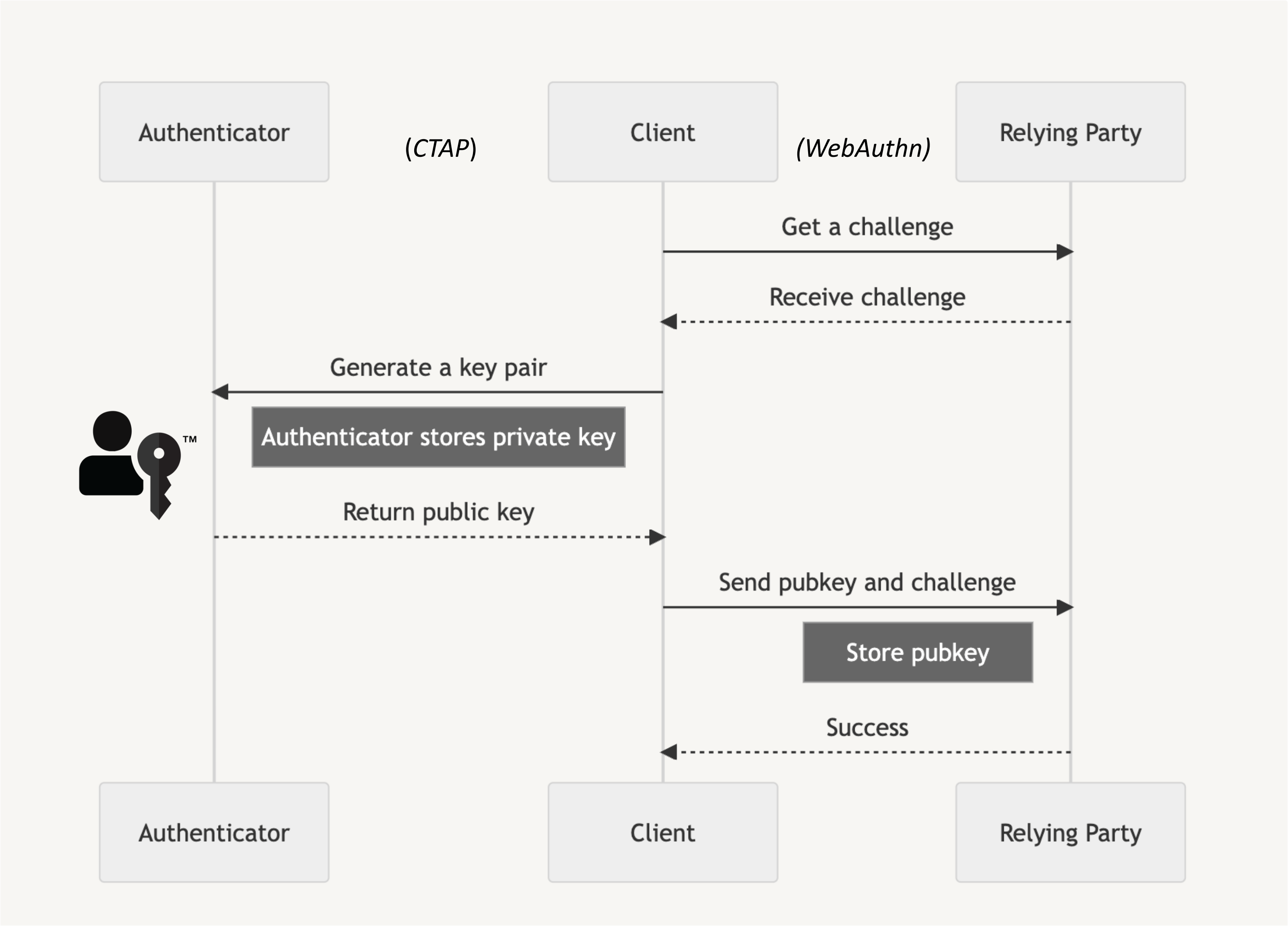
Password:

☐ Create a passkey

Login Login with a passkey

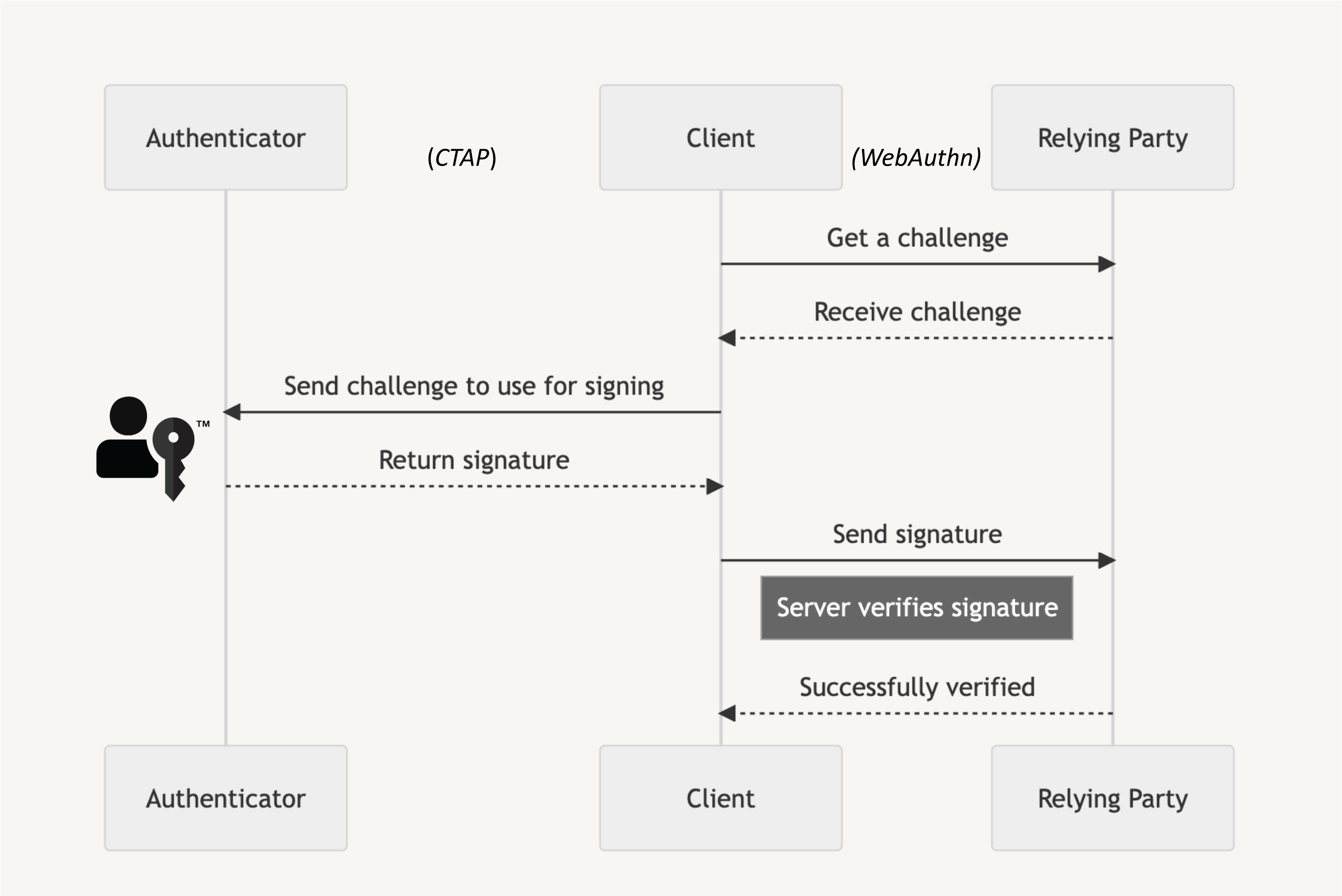
How Passkeys Work

How Passkeys Work – Registration Flow



Attribution: https://files.smashing.media/articles/passkeys-explainer-future-password-less-authentication/5-decision-tree-workflow.png

How Passkeys Work – Login Flow



Attribution: https://files.smashing.media/articles/passkeys-explainer-future-password-less-authentication/6-decision-tree-workflow.png

Passkey Eco-Systems

Passkey Eco-Systems

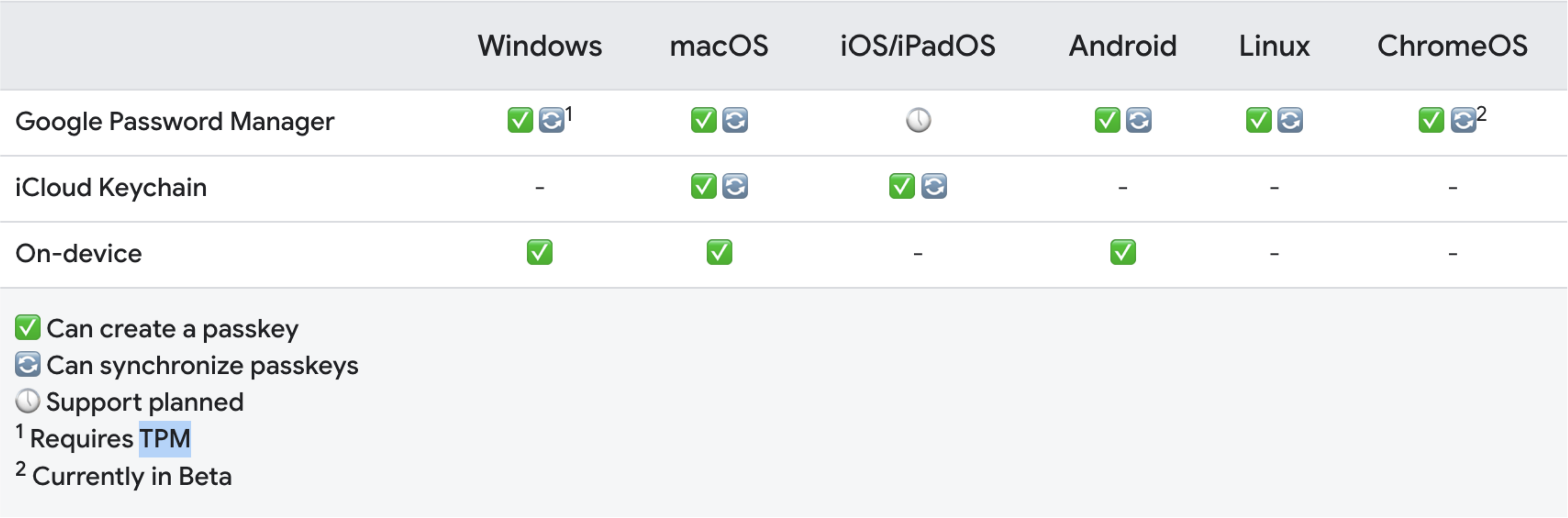
- Synchronises Passkeys between trusted devices and adds redundancy
- Apple (Password App/iCloud Keychain) – Mac, iPhone, iPad
- Google (GPM) – Android Phone/Tablet, Chromebooks (currently beta) and Linux, plus Chrome Browser on multiple platforms (Windows/Mac), with planned support for iOS/iPadOS
- Microsoft – Windows devices only
- Third-Party

Apple Eco-System

- Syncs Safari via iCloud Keychain between all devices that share a common Apple ID:-
  - iPhone
  - iPad
  - Mac
- As of iOS 18, iPadOS 18, macOS Sequoia, Passkeys are managed with the new dedicated Passwords app

Google Eco-System

- Syncs Chrome via Google Password Manager (GPM)
- Support planned for iOS/iPadOS (currently uses iCloud Keychain)



	Windows	macOS	iOS/iPadOS	Android	Linux	ChromeOS
Google Password Manager	✅ 🔄[1]	✅ 🔄	🕒	✅ 🔄	✅ 🔄	✅ 🔄[2]
iCloud Keychain	-	✅ 🔄	✅ 🔄	-	-	-
On-device	✅	✅	-	✅	-	-

✅ Can create a passkey
🔄 Can synchronize passkeys
🕒 Support planned
[1] Requires TPM
[2] Currently in Beta

Attribution: https://developer.chrome.com/blog/passkeys-gpm-desktop

Microsoft Eco-System

- Uses Edge via Windows Hello
- Windows 10 is very limited, no management or synchronization
- Windows 11 has better support, but still evolving - planned improvements recently announced include:
    - A plug-in model for third-party passkey providers
    - Enhanced native UX for passkeys
    - A Microsoft synced passkey provider

Third Party Eco-Systems

- Cross-device and eco-system agnostic
- Popular examples include (all have free lifetime or limited time trials available):-
  - 1Password (cloud)
  - Proton Pass (cloud)
  - NordPass (cloud)
  - Bitwarden (cloud or self-host)
- Native apps and browser extensions
- Business/enterprise versions available
- Can be problematic to setup on devices due to differing device security constraints

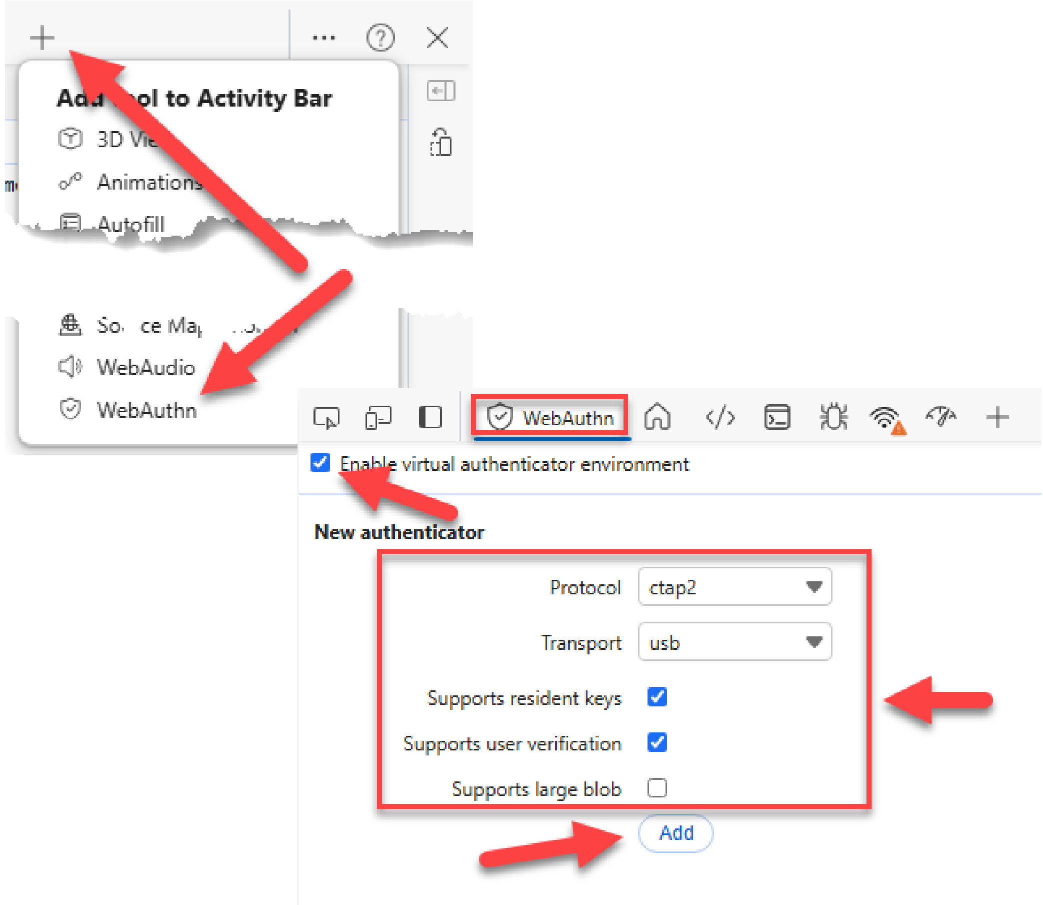
Virtual Authenticators
for
Testing & Troubleshooting

Virtual Authenticators

- WebAuthn specification includes an embedded Browser API specification for Virtual Authenticators - however only Chrome/Edge have currently implemented it
- This embedded API is also accessible via the Chrome Debug Protocol built into Chrome/Edge and is therefore accessible from E2E tools such as Cypress
- Chrome/Edge WebAuthn DevTools together with third-party Demo WebAuthn specification sites (for example webauthn.io) provide a great insight into the inner workings of Passkeys, and can aid testing and troubleshooting

Virtual Authenticators Demo

Virtual Authenticators - Setup

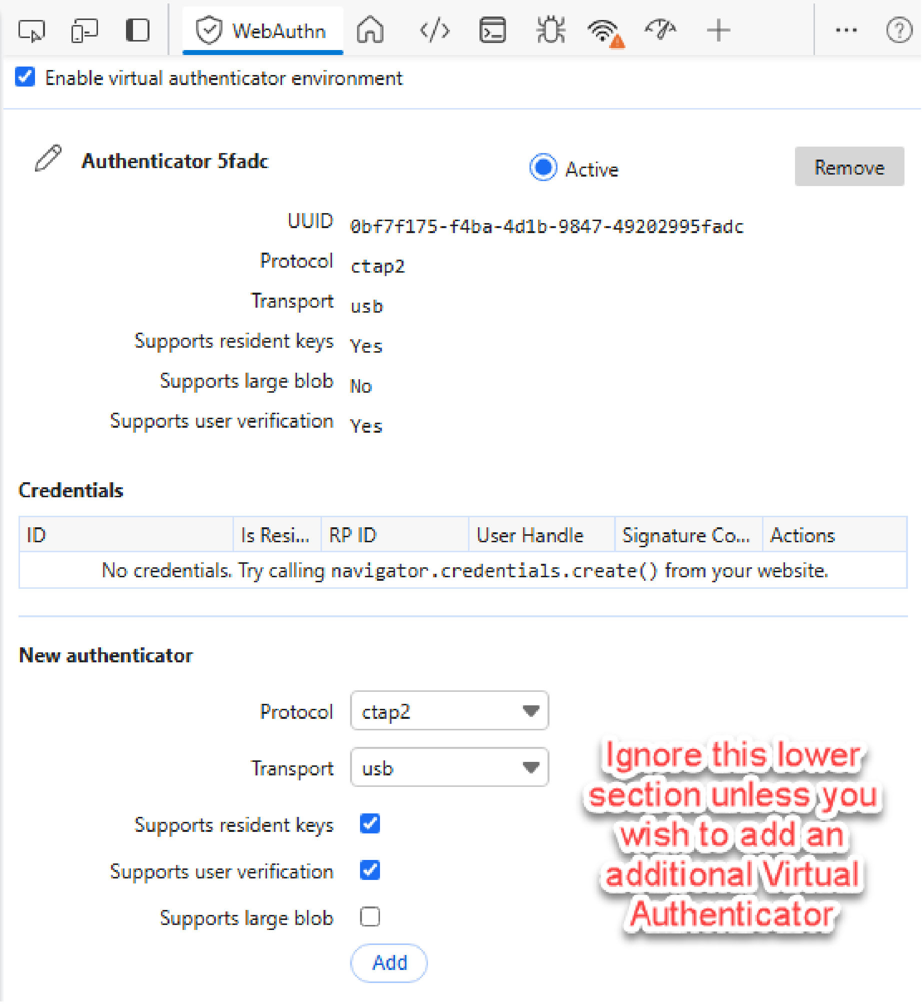


☑ Enable virtual authenticator environment

New authenticator

Protocol	ctap2
Transport	usb
Supports resident keys	☑
Supports user verification	☑
Supports large blob	□

Add



☑ Enable virtual authenticator environment

Authenticator 5fadc ● Active Remove

UUID	0bf7f175-f4ba-4d1b-9847-49202995fadc
Protocol	ctap2
Transport	usb
Supports resident keys	Yes
Supports large blob	No
Supports user verification	Yes

Credentials

ID	Is Resi...	RP ID	User Handle	Signature Co...	Actions
No credentials. Try calling navigator.credentials.create() from your website.					

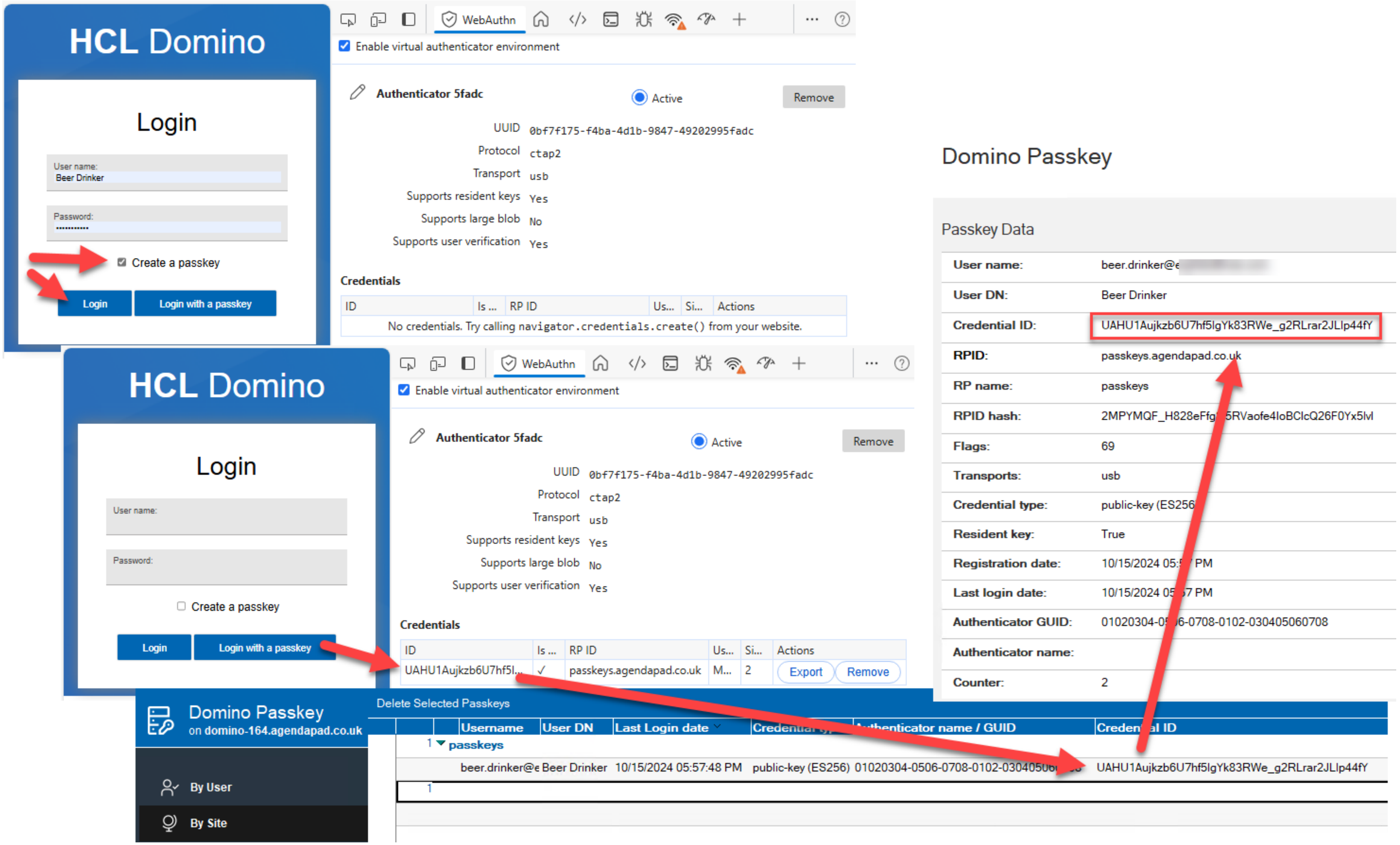
New authenticator

Protocol	ctap2
Transport	usb
Supports resident keys	☑
Supports user verification	☑
Supports large blob	□

Add

Ignore this lower section unless you wish to add an additional Virtual Authenticator

Virtual Authenticators - Domino



HCL Domino

Login

User name: Beer Drinker

Password: ••••••••••••

☑ Create a passkey

Login | Login with a passkey

WebAuthn

☑ Enable virtual authenticator environment

Authenticator 5fadc ● Active Remove

UUID	`0bf7f175-f4ba-4d1b-9847-49202995fadc`
Protocol	`ctap2`
Transport	`usb`
Supports resident keys	Yes
Supports large blob	No
Supports user verification	Yes

Credentials

ID	Is ...	RP ID	Us...	Si...	Actions
No credentials. Try calling `navigator.credentials.create()` from your website.					

HCL Domino

Login

User name:

Password:

☐ Create a passkey

Login | Login with a passkey

WebAuthn

☑ Enable virtual authenticator environment

Authenticator 5fadc ● Active Remove

UUID	`0bf7f175-f4ba-4d1b-9847-49202995fadc`
Protocol	`ctap2`
Transport	`usb`
Supports resident keys	Yes
Supports large blob	No
Supports user verification	Yes

Credentials

ID	Is ...	RP ID	Us...	Si...	Actions
UAHU1Aujkzb6U7hf5l...	✓	passkeys.agendapad.co.uk	M...	2	Export Remove

Domino Passkey

Passkey Data

User name:	beer.drinker@e[redacted]
User DN:	Beer Drinker
Credential ID:	UAHU1Aujkzb6U7hf5lgYk83RWe_g2RLrar2JLlp44fY
RPID:	passkeys.agendapad.co.uk
RP name:	passkeys
RPID hash:	2MPYMQF_H828eFfg[illegible]5RVaofe4loBClcQ26F0Yx5lvl
Flags:	69
Transports:	usb
Credential type:	public-key (ES256)
Resident key:	True
Registration date:	10/15/2024 05:57 PM
Last login date:	10/15/2024 05:57 PM
Authenticator GUID:	01020304-0506-0708-0102-030405060708
Authenticator name:	
Counter:	2

Domino Passkey
on domino-164.agendapad.co.uk

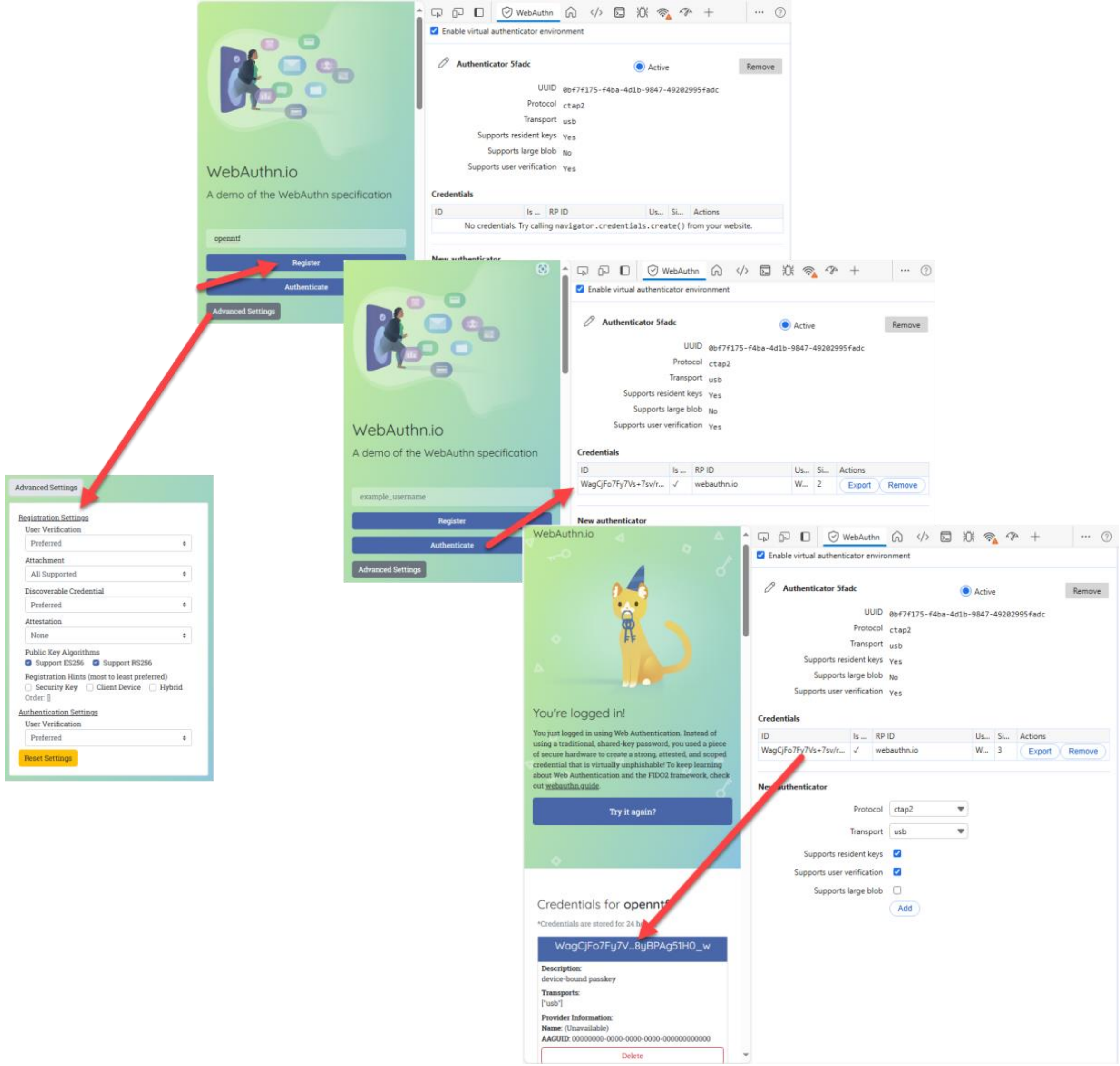
By User

By Site

Delete Selected Passkeys

Username	User DN	Last Login date	Credential type	Authenticator name / GUID	Credential ID
1 ▾ passkeys					
beer.drinker@e	Beer Drinker	10/15/2024 05:57:48 PM	public-key (ES256)	01020304-0506-0708-0102-030405060708	UAHU1Aujkzb6U7hf5lgYk83RWe_g2RLrar2JLlp44fY
1					

Virtual Authenticators - webauthn.io



WebAuthn.io

A demo of the WebAuthn specification

openntf

Enable virtual authenticator environment ☑

Authenticator 5fadc ● Active

UUID	0bf7f175-f4ba-4d1b-9847-49202995fadc
Protocol	ctap2
Transport	usb
Supports resident keys	Yes
Supports large blob	No
Supports user verification	Yes

Credentials

ID	Is ...	RP ID	Us...	Si...	Actions
No credentials. Try calling navigator.credentials.create() from your website.					

Advanced Settings

Registration Settings

User Verification: Preferred

Attachment: All Supported

Discoverable Credential: Preferred

Attestation: None

Public Key Algorithms

☑ Support ES256 ☑ Support RS256

Registration Hints (most to least preferred)

□ Security Key □ Client Device □ Hybrid

Order: []

Authentication Settings

User Verification: Preferred

WebAuthn.io

A demo of the WebAuthn specification

example_username

Credentials

ID	Is ...	RP ID	Us...	Si...	Actions
WagCjFo7Fy7Vs+7sv/r...	✓	webauthn.io	W...	2	Export Remove

New authenticator

WebAuthn.io

You're logged in!

You just logged in using Web Authentication. Instead of using a traditional, shared-key password, you used a piece of secure hardware to create a strong, attested, and scoped credential that is virtually unphishable! To keep learning about Web Authentication and the FIDO2 framework, check out webauthn.guide.

Credentials for opennt

*Credentials are stored for 24 h

WagCjFo7Fy7V...8yBPAg51H0_w

Description: device-bound passkey

Transports: ["usb"]

Provider Information:

Name: (Unavailable)

AAGUID: 00000000-0000-0000-0000-000000000000

Credentials

ID	Is ...	RP ID	Us...	Si...	Actions
WagCjFo7Fy7Vs+7sv/r...	✓	webauthn.io	W...	3	Export Remove

New authenticator

Protocol	ctap2
Transport	usb
Supports resident keys	☑
Supports user verification	☑
Supports large blob	□

Summary

- Despite multiple attempts to improve the security of passwords with MFA, they are still fundamentally flawed and risk attack from multiple vectors
- Passkeys address many of the issues with passwords (Phishing, MITM & data breaches) and currently provide state-of-the-art protection for end-user web authentication
- The end-user Passkey experience is consistent and already familiar
- Within Domino, Passkeys are very easy to setup and enable
- Virtual Authenticators are available for system testing and troubleshooting
- Whilst Passkey eco-systems continue to improve and evolve, there is still more to be done, for example better device support for multiple concurrent Passkey managers, and standardised credential portability import/export
- End-user Passkey take-up will improve significantly as website adoption increases to reach critical mass

Thank You!

Do we have time for any questions?

References

- Passkeys on Windows: Authenticate seamlessly with passkey providers => https://blogs.windows.com/windowsdeveloper/2024/10/08/passkeys-on-windows-authenticate-seamlessly-with-passkey-providers/
- Chrome to sync passkeys on Google Password Manager between desktop and Android => https://developer.chrome.com/blog/passkeys-gpm-desktop
- Passkeys => https://fidoalliance.org/passkeys/
- FIDO Authentication => https://fidoalliance.org/fido2/
- Passkey Central => https://passkeycentral.org

Domino Optional notes.ini Settings (1/2)

- **PASSKEY_DATABASE= test\mypasskeys.nsf** - Change filename and location of passkey database
- **PASSKEY_SERVER_FRIENDLY_NAME=myfriendlyname** - Use to change the friendly name for the RP. This is useful when an administrator wants multiple Internet Site documents to share passkeys and wishes them all to be perceived as a single site or service.
- **PASSKEY_SKIP_LEVEL=N** - Use to skip the first N parts of the effective domain when constructing the RP ID. For example, PASSKEY_SKIP_LEVEL=1 could be used to allow *server01.domino.example.com*, *server02.domino.example.com*, and *server03.domino.example.com* to all use an RP ID of domino.example.com and share passkeys. If these sites were hosted on different Domino servers, then passkey.nsf would need to be replicated between them.
- **PASSKEY_DOMAIN_LEVELS=N** - Can be used to only use the last N parts of the effective domain when constructing the RP ID.
  For example, PASSKEY_DOMAIN_LEVELS=3 could be used to allow *server01.domino.example.com*, *www.mytest.domino.example.com*, and *domino.example.com* to all use an RP ID of domino.example.com and share passkeys. If these sites were hosted on different Domino servers, then passkey.nsf would need to be replicated between them.

Note: PASSKEY_SKIP_LEVEL=N and PASSKEY_DOMAIN_LEVELS=N are mutually exclusive.

Domino Optional notes.ini Settings (2/2)

- **PASSKEY_ALWAYS_UPDATE_LAST_LOGIN=N** - By default, Domino will always update the last logged in time for a passkey credential in passkey.nsf after that credential has been used to authenticate successfully. This may negatively impact performance on a heavily trafficked server. Setting PASSKEY_ALWAYS_UPDATE_LAST_LOGIN=0 in the server's notes.ini will disable this functionality and cause Domino to only update the credential's last logged in time if the authenticator sent a non-zero counter during authentication. This will improve performance, but will result in inconsistent last login times in passkey.nsf.
- **PASSKEY_REQUEST_DIRECT_ATTSTMT=N** - If PASSKEY_REQUEST_DIRECT_ATTSTMT=0 is set in the Domino server's notes.ini, then Domino will request an attestation type of "none". This will cause most authenticators to register an AAGUID of all zeroes, which will correspond to a blank "Authenticator name" field in passkey.nsf.
- **PASSKEY_ALLOW_REPEATED_REGISTRATION=N** - Each user is normally limited to registering one passkey per authenticator for each relying party to prevent confusion. If you wish to remove that restriction, possibly for testing purposes, set PASSKEY_ALLOW_REPEATED_REGISTRATION=1 in the server's notes.ini.
- **PASSKEY_REQUIRE_USER_VERIFICATION=1** Authenticators will always check for user presence before using a passkey. By default, an authenticator will request user verification if it can, but will not require it if it cannot. For example, some older Yubikey devices only have a button to press to signify user presence, but lack a fingerprint reader or PIN to verify the identity of the user with the device. Similarly, some laptops will allow passkey authentication when the laptop lid is closed and the fingerprint reader is not currently active. Administrators can require user verification by setting PASSKEY_REQUIRE_USER_VERIFICATION=1 in the Domino server's notes.ini. This is likely to adversely impact users with FIDO2 devices lacking biometrics that were configured without a PIN, and users with closed and "docked" laptops, but can be used to strictly enforce multi-factor authentication.