16 Oktober 2025 BaselOne

Welcome01

Passwordless Authentication with Webauthn

A Hands-On Developer’s Guide

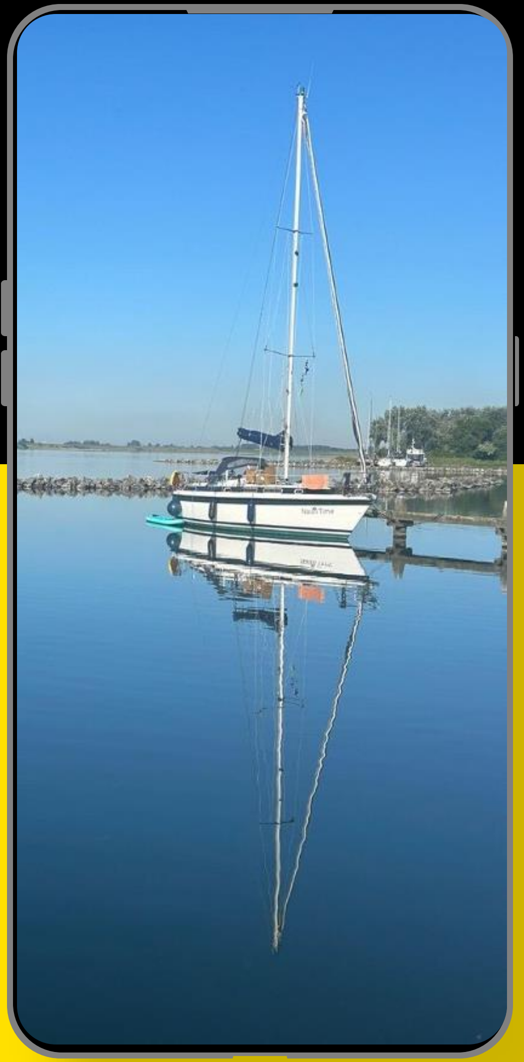
Ralph Hees

Ralph Hees

Living in The Netherlands

Consultant & IT Architect

Software, Cloud, CICD, Enterprise



Content

The problem with passwords

Weaknesses in Traditional MFA

Why Passkeys are better?

How does it work?

Demo’s:

Passwordless login on Kubernetes

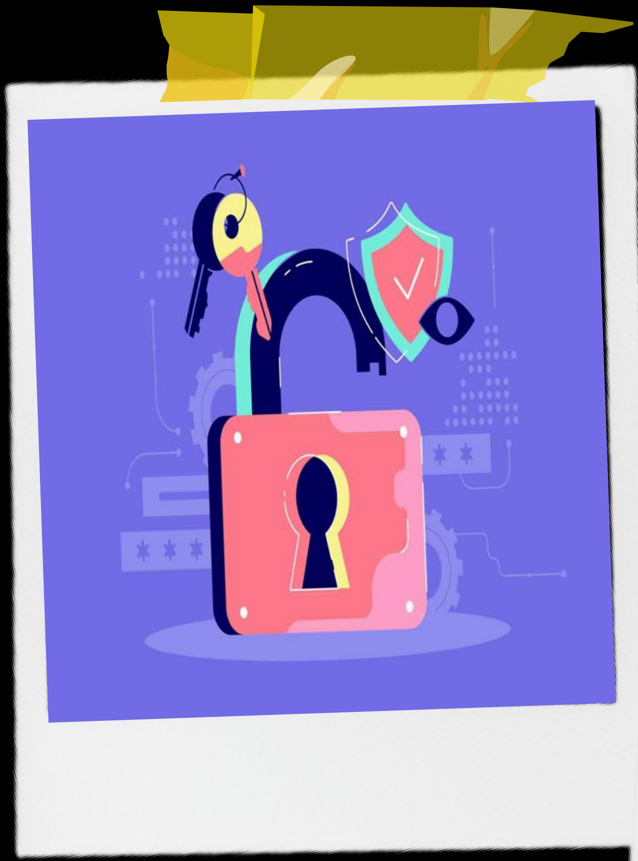
Spring-boot and Keycloak

Plain spring-boot passwordless login

The problem with passwords

- Easy to forget, reuse, and steal
- Costly to reset (helpdesk burden)
- Even with MFA, users are still phished

→ We need something simpler and more secure



Weaknesses in Traditional MFA

Code sent by SMS

Commonly used, but highly insecure.

- Vulnerable to SIM swapping
- Phishable
- Dependent on mobile service

SMS

Email

TOTP

Push

Weaknesses in Traditional MFA

Code sent by E-mail

- **Email compromise**
- **Phishable**
- **Slow and inconsistent**
- **Weak binding -** security is only as good as your email password

SMS

Email

TOTP

Push

Weaknesses in Traditional MFA

Time based token in apps

- Still phishable
- Manual entry = friction
- Human error

SMS

Email

TOTP

Push

Weaknesses in Traditional MFA

Push notification approval

- Push fatigue
- Real-time phishing
- Device-dependent

SMS

Email

TOTP

Push

The solution

Passwordless with WebAuthn

- Replaces passwords with cryptographic keys
- Authenticates via biometrics or device PIN
- Private key never leaves the device
- Tied to legitimate website domain → Phishing-resistant

Cryptographic keys

Biometrics or device PIN

Private key protection

Domain tied

THE SOLUTION

Why Passkeys Are Better

- No shared secrets → Nothing to steal or reuse
- Inherently multi-factor (device + biometric)
- Seamless user experience (one tap login)
- Works across all major platforms & browsers

Cryptographic keys

Device & Biometrics

Private key protection

All platforms

Security Comparison

Password + SMS

Push MFA

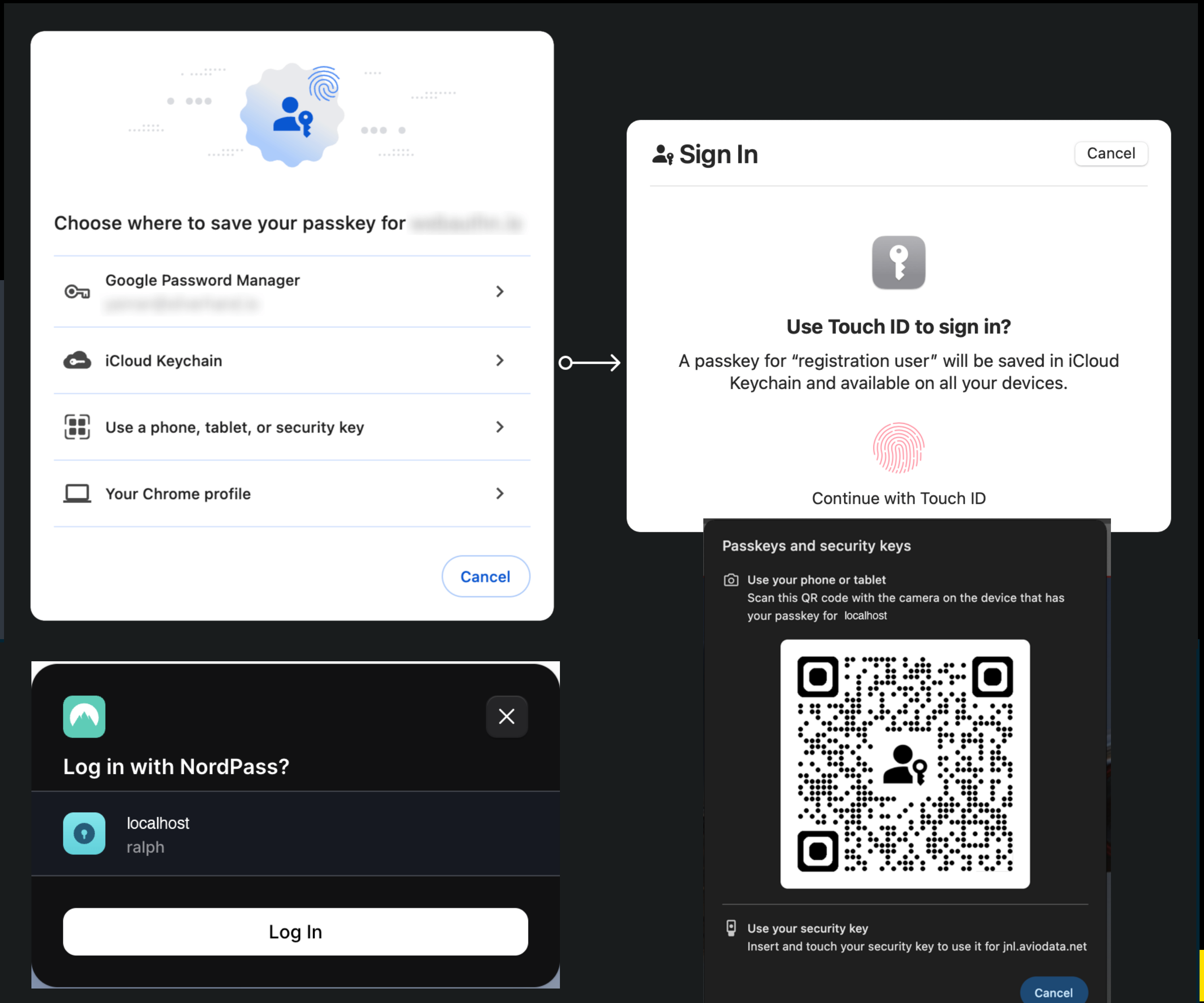
Password + TOTP

Passkey

Passkey

Phishing-resistant, Seamless, Secure

How does it look like for a user?



Choose where to save your passkey for [redacted]

- Google Password Manager [redacted]
- iCloud Keychain
- Use a phone, tablet, or security key
- Your Chrome profile

Cancel

Sign In Cancel

Use Touch ID to sign in?

A passkey for "registration user" will be saved in iCloud Keychain and available on all your devices.

Continue with Touch ID

Passkeys and security keys

Use your phone or tablet
Scan this QR code with the camera on the device that has your passkey for localhost

Use your security key
Insert and touch your security key to use it for jnl.aviodata.net

Cancel

Log in with NordPass?

localhost
ralph

Log In

Register a passkey

Create a certificate based on the server settings.

Sign in using passkey

Biometrics check using face ID, Touch ID, FIDO2 U2F-key or login on another device.

Webauthn in depth?

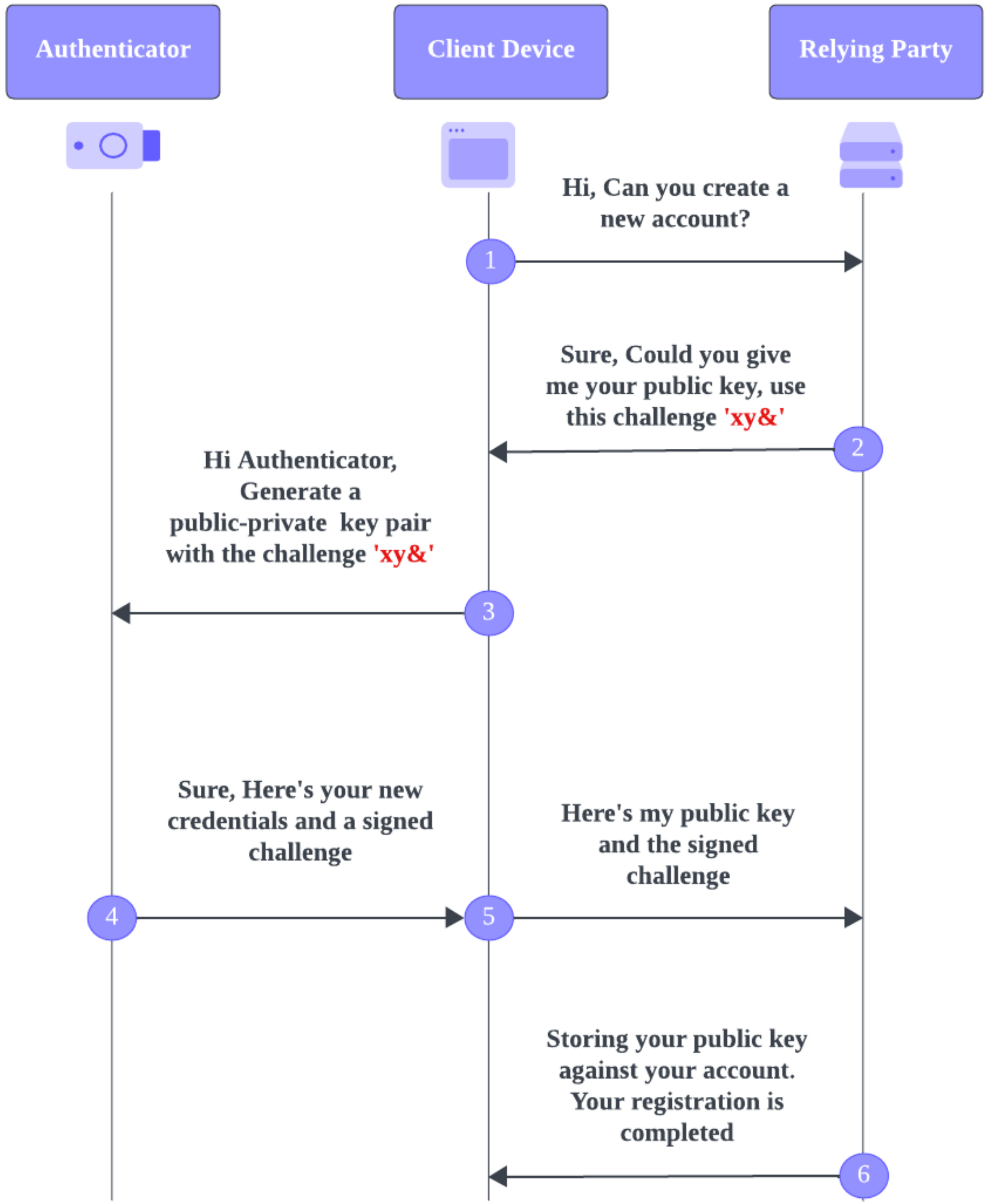
navigator.credentials.register

Request public key creation

navigator.credentials.get

Request public key

Demo time



1 Hi, Can you create a new account?

2 Sure, Could you give me your public key, use this challenge 'xy&'

3 Hi Authenticator, Generate a public-private key pair with the challenge 'xy&'

4 Sure, Here's your new credentials and a signed challenge

5 Here's my public key and the signed challenge

6 Storing your public key against your account. Your registration is completed

Webauthn in depth?

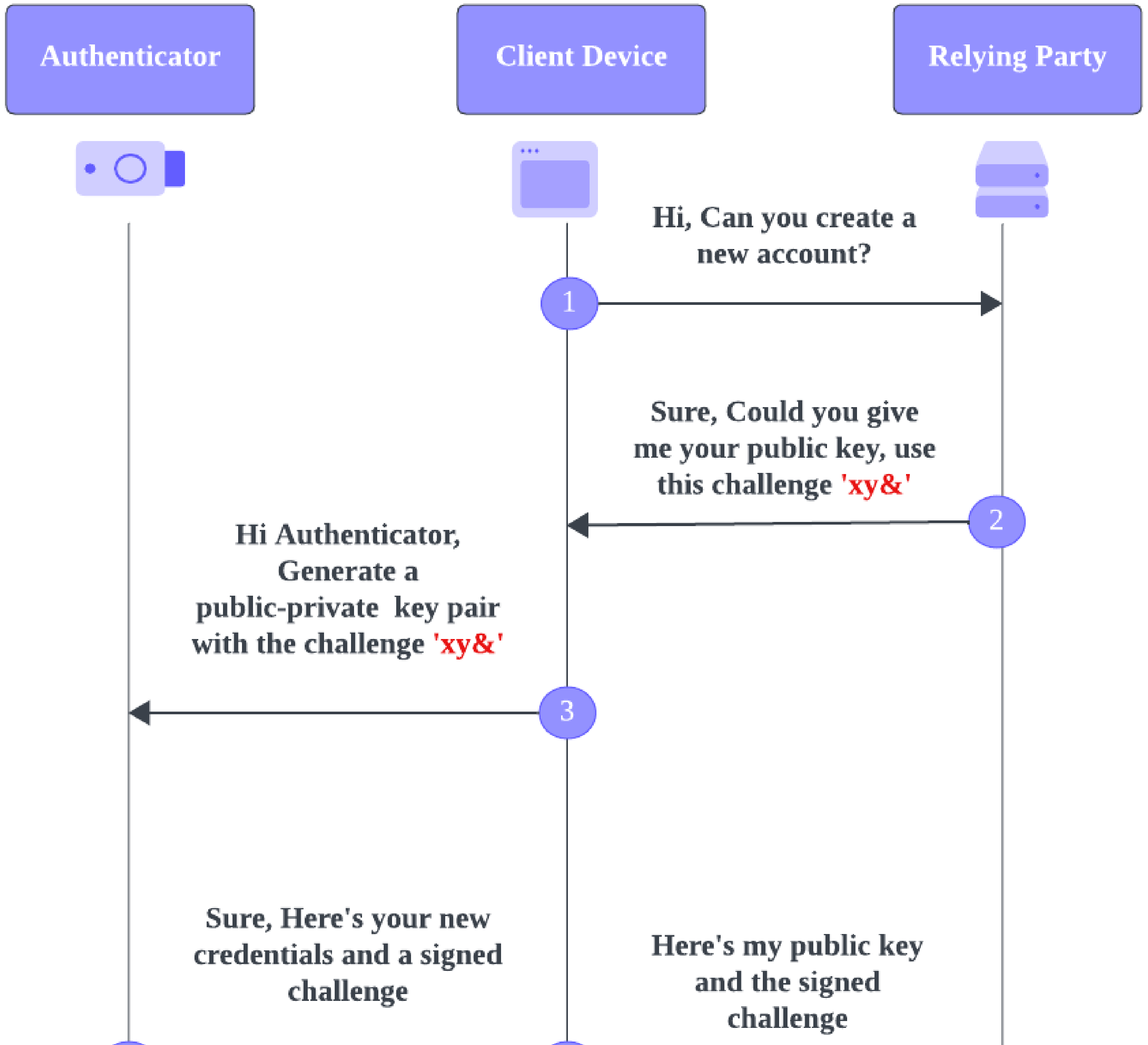
***navigator*.credentials.register**

Request public key creation

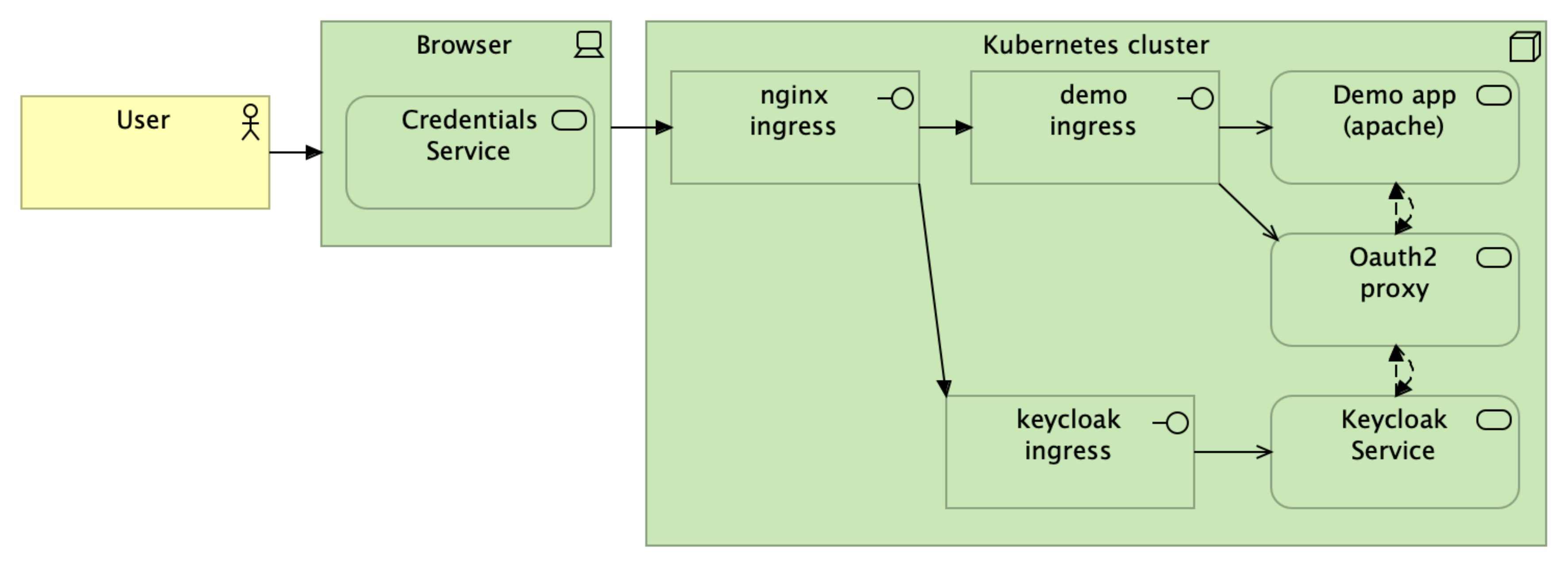
***navigator*.credentials.get**

Request public key

Demo time


Authenticator
Client Device
Relying Party
1
Hi, Can you create a new account?
2
Sure, Could you give me your public key, use this challenge 'xy&'
3
Hi Authenticator, Generate a public-private key pair with the challenge 'xy&'
Sure, Here's your new credentials and a signed challenge
Here's my public key and the signed challenge

Passwordless login on Kubernetes



Apache server

Application requires authentication
Ensure authentication before access.

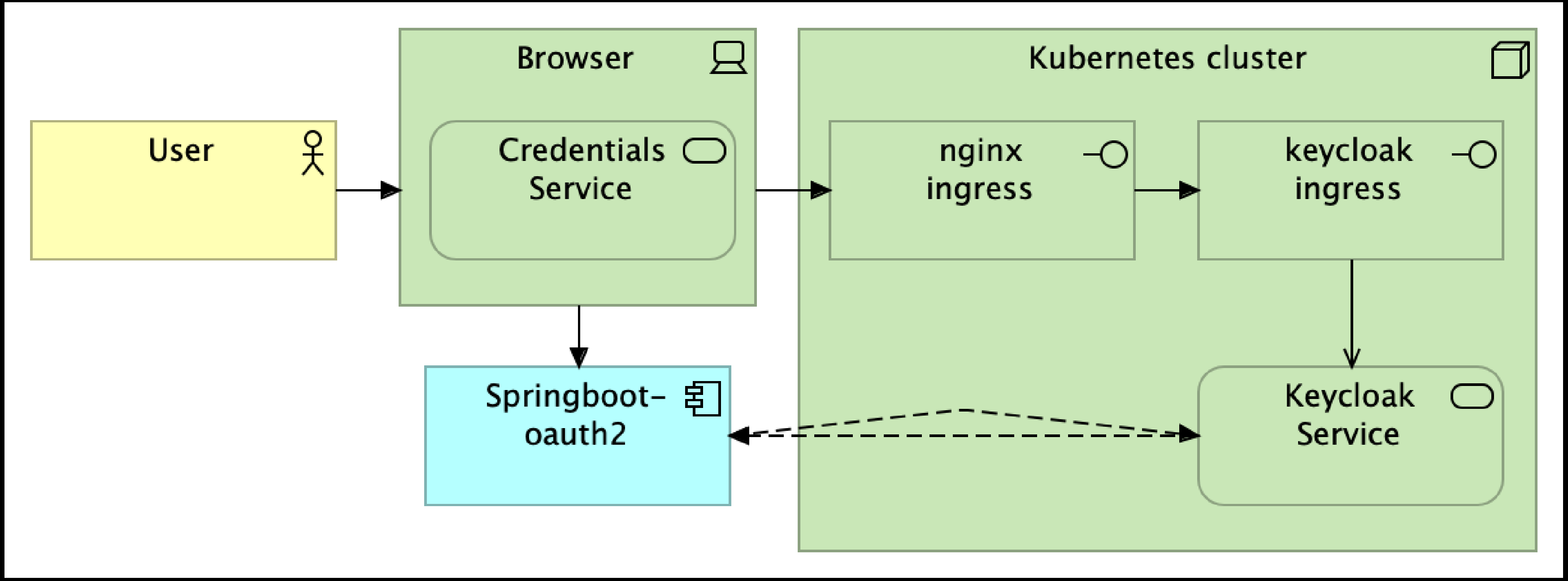
OAuth2-proxy

Protects the ingress to only allow authenticated access.

Keycloak

Identity provider containing two WebAuthn implementations.

Spring-boot and Keycloak



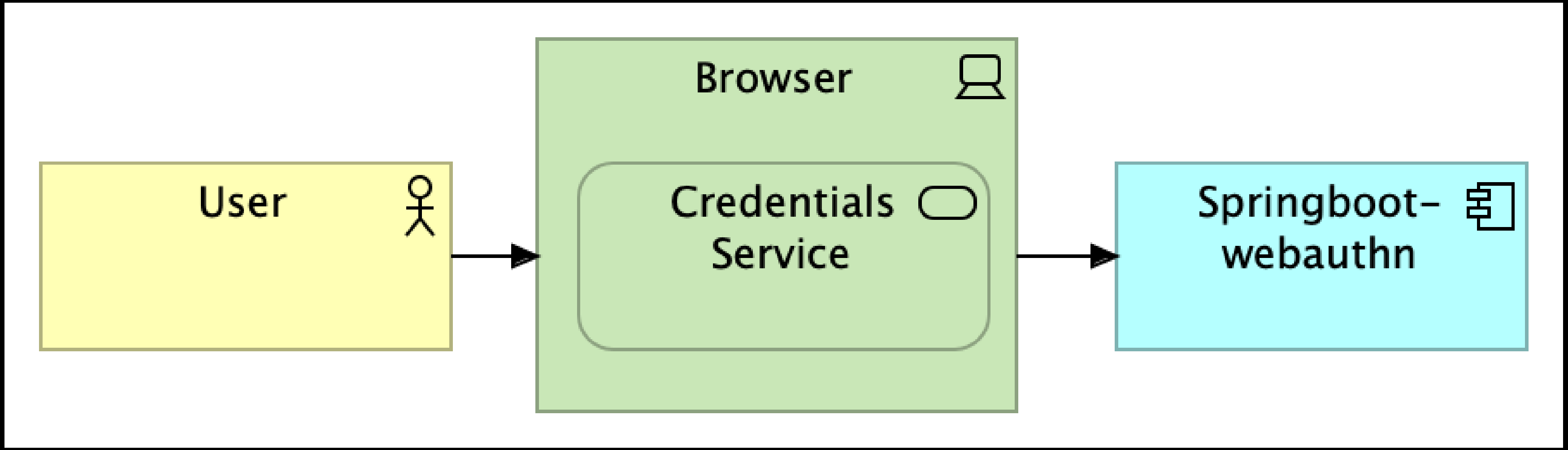
Spring-boot app

Uses oauth2 protocol to login using keycloak.

Keycloak

Identity provider containing two WebAuthn implementations.

Spring-boot passwordless login



Spring-boot app

Uses the spring-webauthn library.

Own user storage

User storage
Credential storage

The future is passwordless

Stop managing passwords

Reduce risk and costs

Reduce risk and cut operational costs

Join leaders

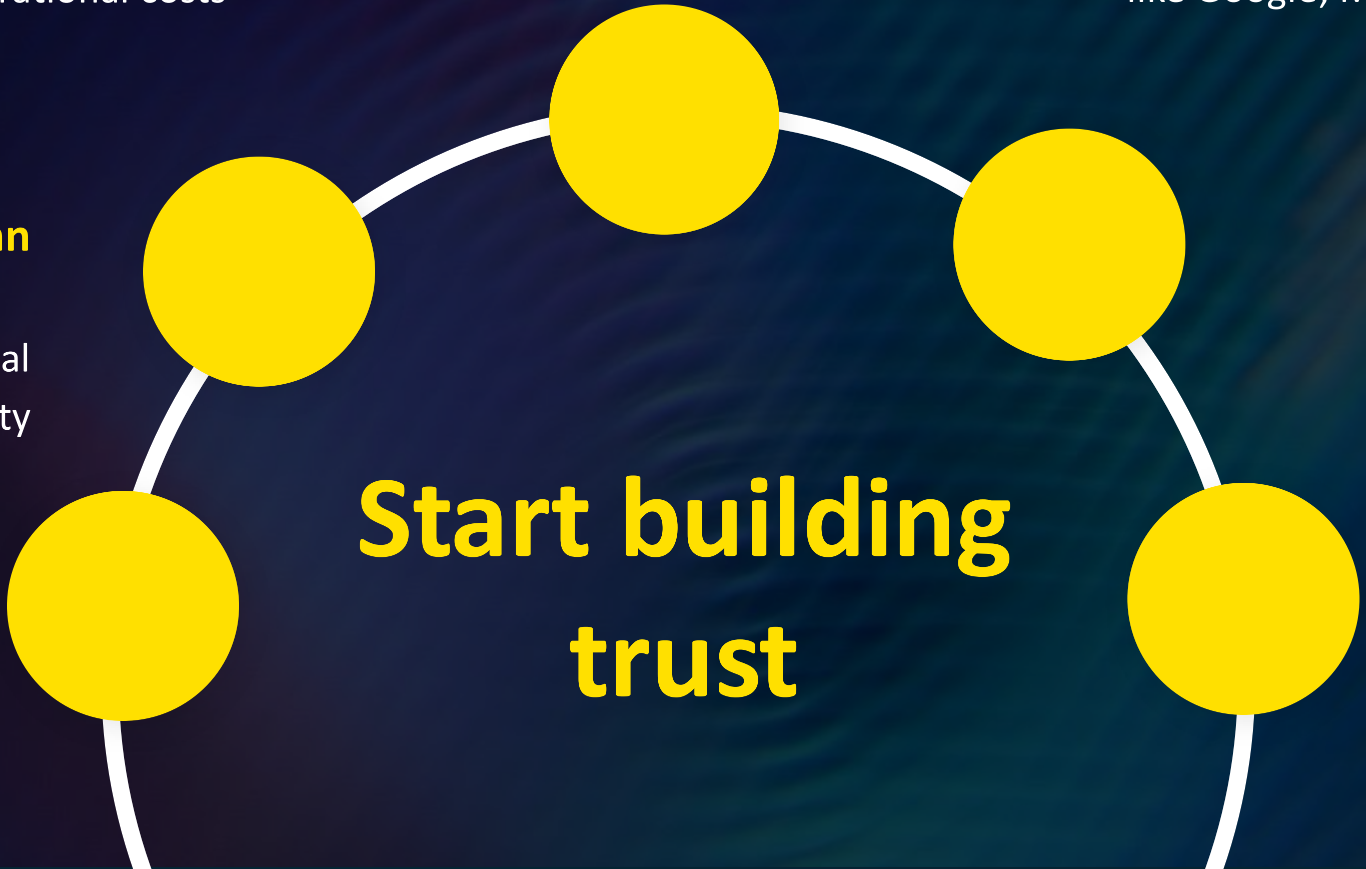
like Google, Microsoft, and GitHub

Adopt WebAuthn

to stay ahead in digital identity security

Improve User Experience

Seamless easy login



Start building trust

baloise

KARAKUN

Kanton Basel-Stadt

ERNI

baseltech

Share feedback

sympany

TuxCare
We Secure Open Source

Code on github

Basel
basel.ch

CSS

swiss made
software

SD:>_
[SwissDevJobs.ch]

JAVA
USER
GROUP
CH

cyon

optravis
company of .msg

BLKB

ICT
Scouts
Wir fördern Zukunft

#BaselOne25 baselone.ch