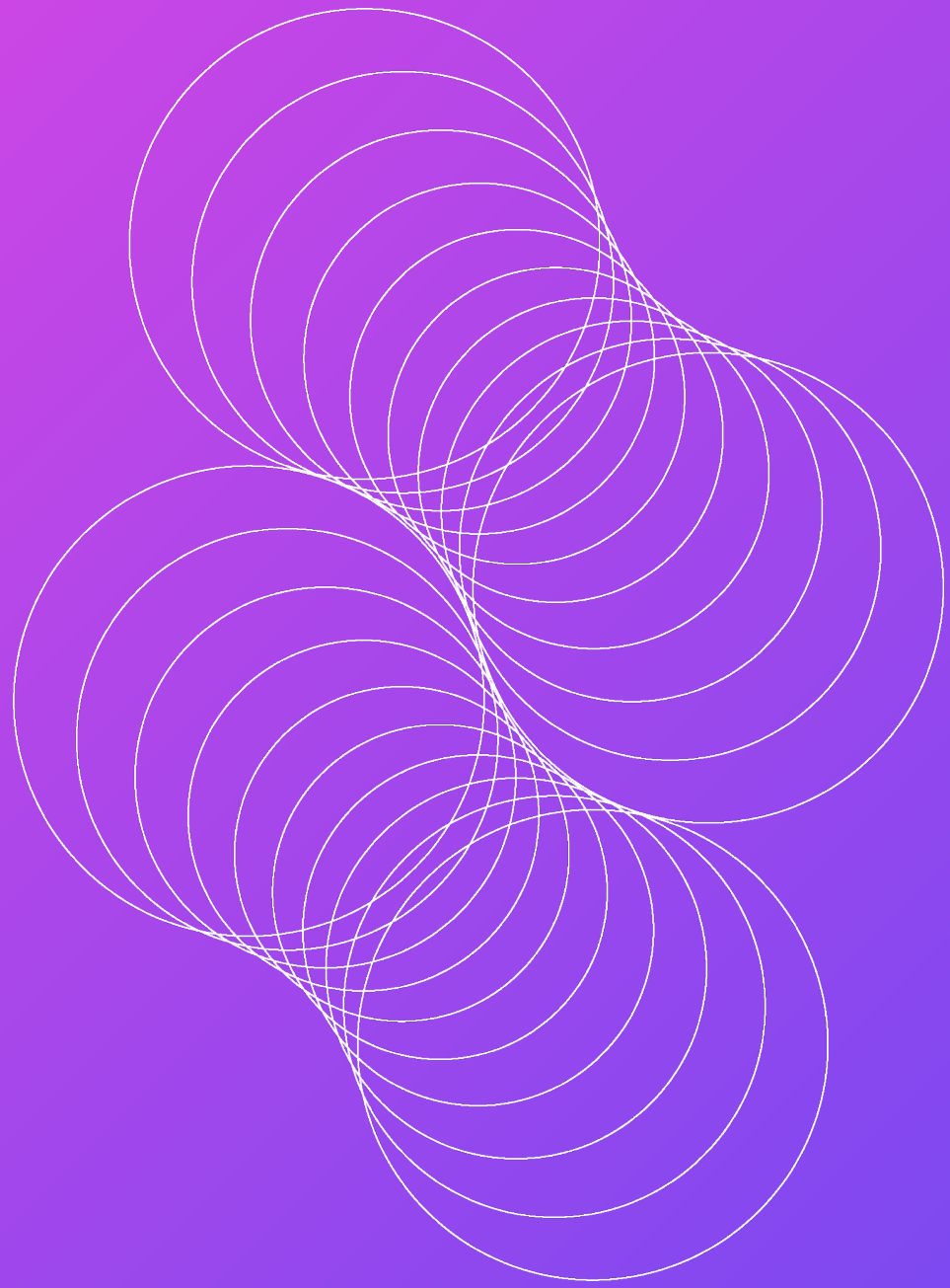
Auth0 by Okta



A Passwordless Future!

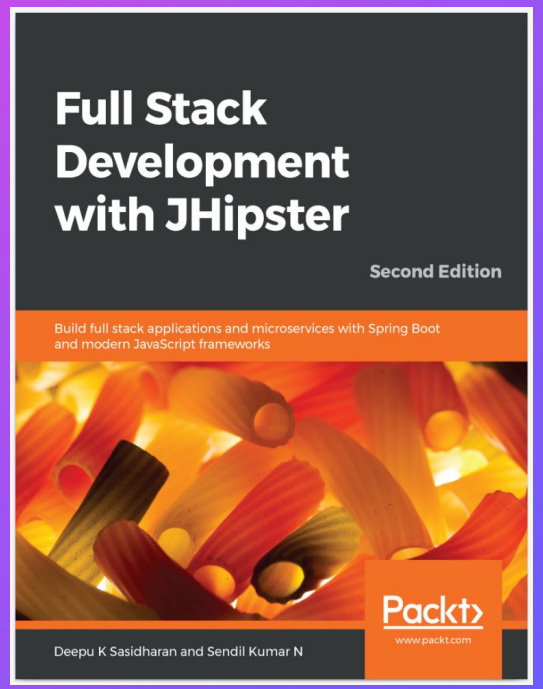
Passkeys and WebAuthn for Java developers

Deepu K Sasidharan

Hi, I'm Deepu K Sasidharan

- JHipster co-chair
- Java Champion
- Creator of KDash, JDL Studio, JWT UI
- OSS aficionado, polyglot dev, author, speaker
- Developer Advocate @ Okta

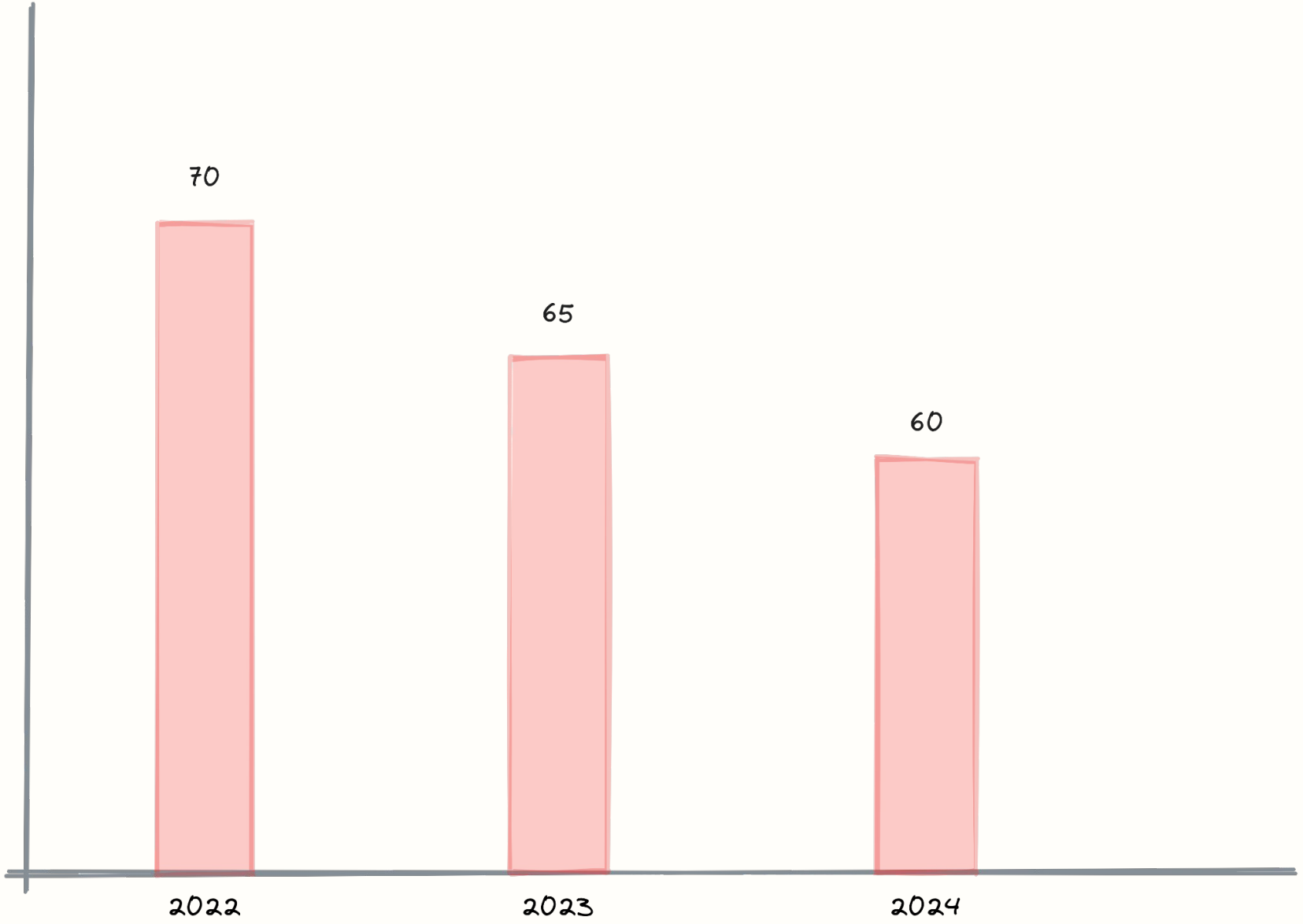
@deepu105@mastodon.social
deepu.tech
@deepu105
deepu05



Why passwordless?

The password problem

Verizon Data Breach Investigation Report



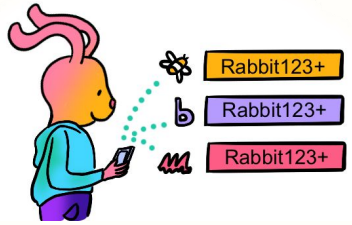
The human problem

Knowledge-based

Phishing

Remote replay

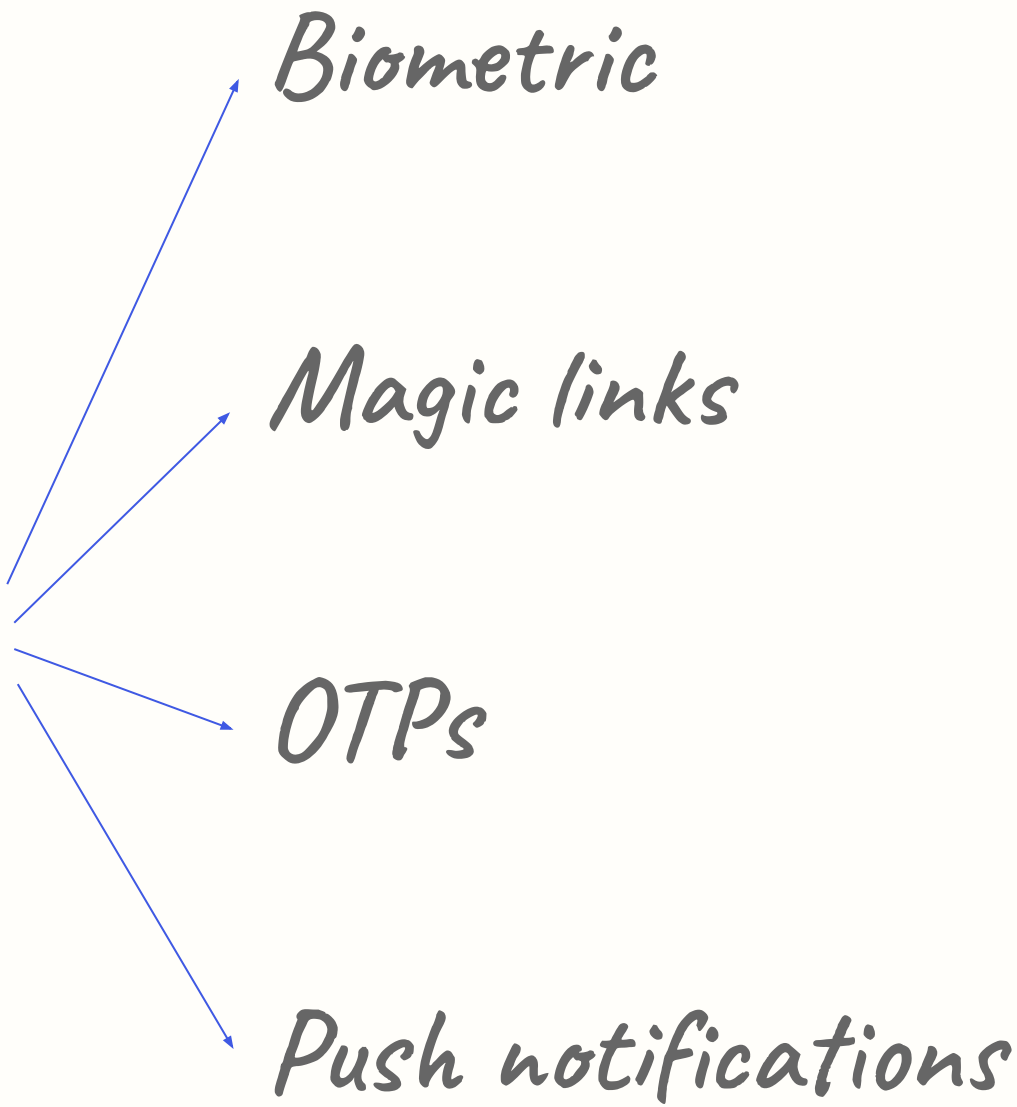
Data breach



Reuse & share

Password management

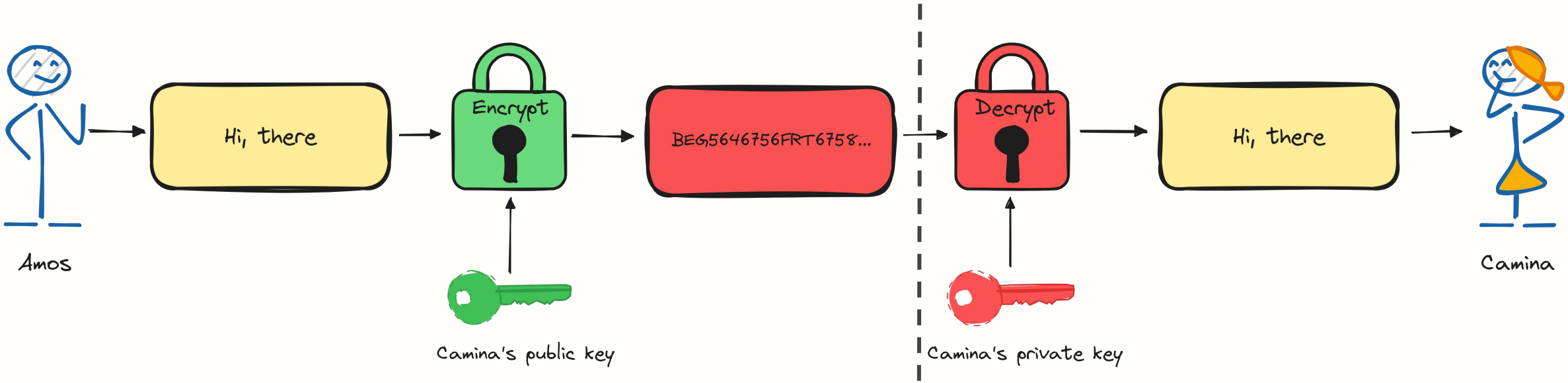
Passwordless

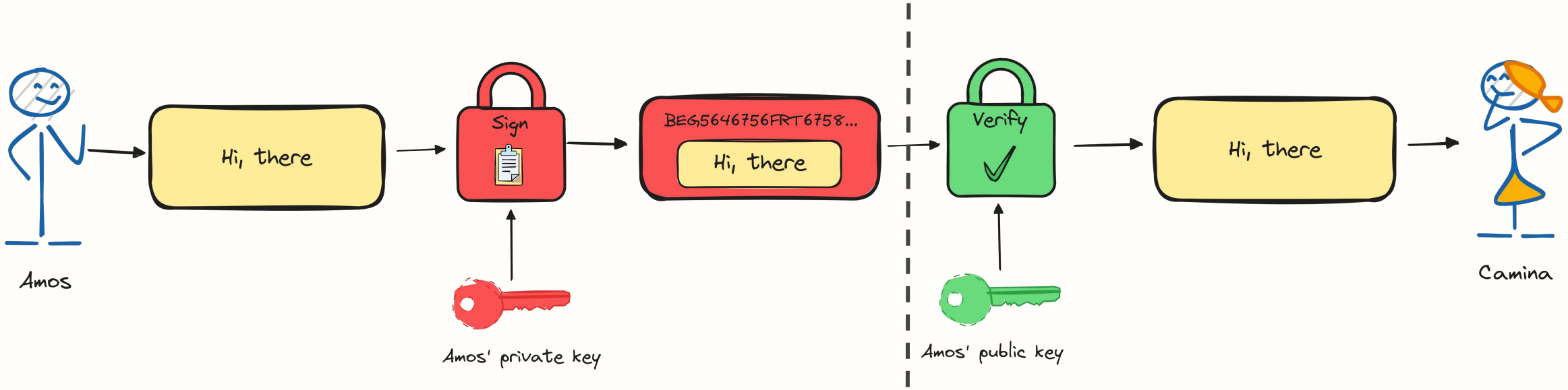


Passwordless future == Passkeys

Public-key cryptography

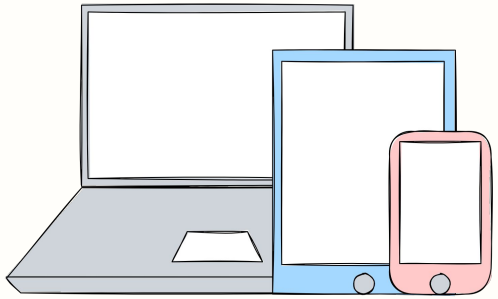
Pair of mathematically linked keys


Amos
Hi, there
Encrypt
BEG5646756FRT6758...
Camina's public key
Decrypt
Camina's private key
Hi, there
Camina


Amos
Hi, there
Sign
Amos' private key
BEG5646756FRT6758...
Hi, there
Verify
Amos' public key
Hi, there
Camina

Authenticator

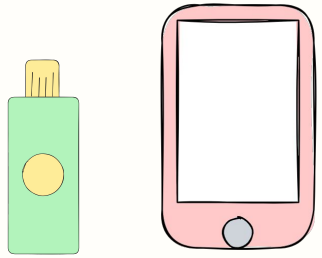
Can create and store public-private key pairs



Platform authenticators

Built into the device

- TouchID
- FaceID
- Smartphone authenticators
- Windows Hello



Roaming authenticators

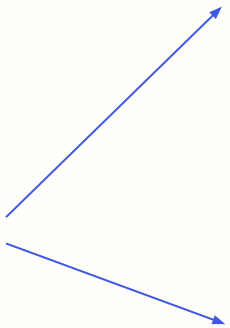
Removable device via USB, NFC, Bluetooth

- Yubikey
- Google Titan
- Smartphones

FIDO == Authentication standard

Based on public key cryptography.

FIDO2



WebAuthentication (WebAuthn)

Client to Authenticator Protocol (CTAP)

WebAuthn == W3C standard

WebAuthn is the standard that allows for passkeys implementation

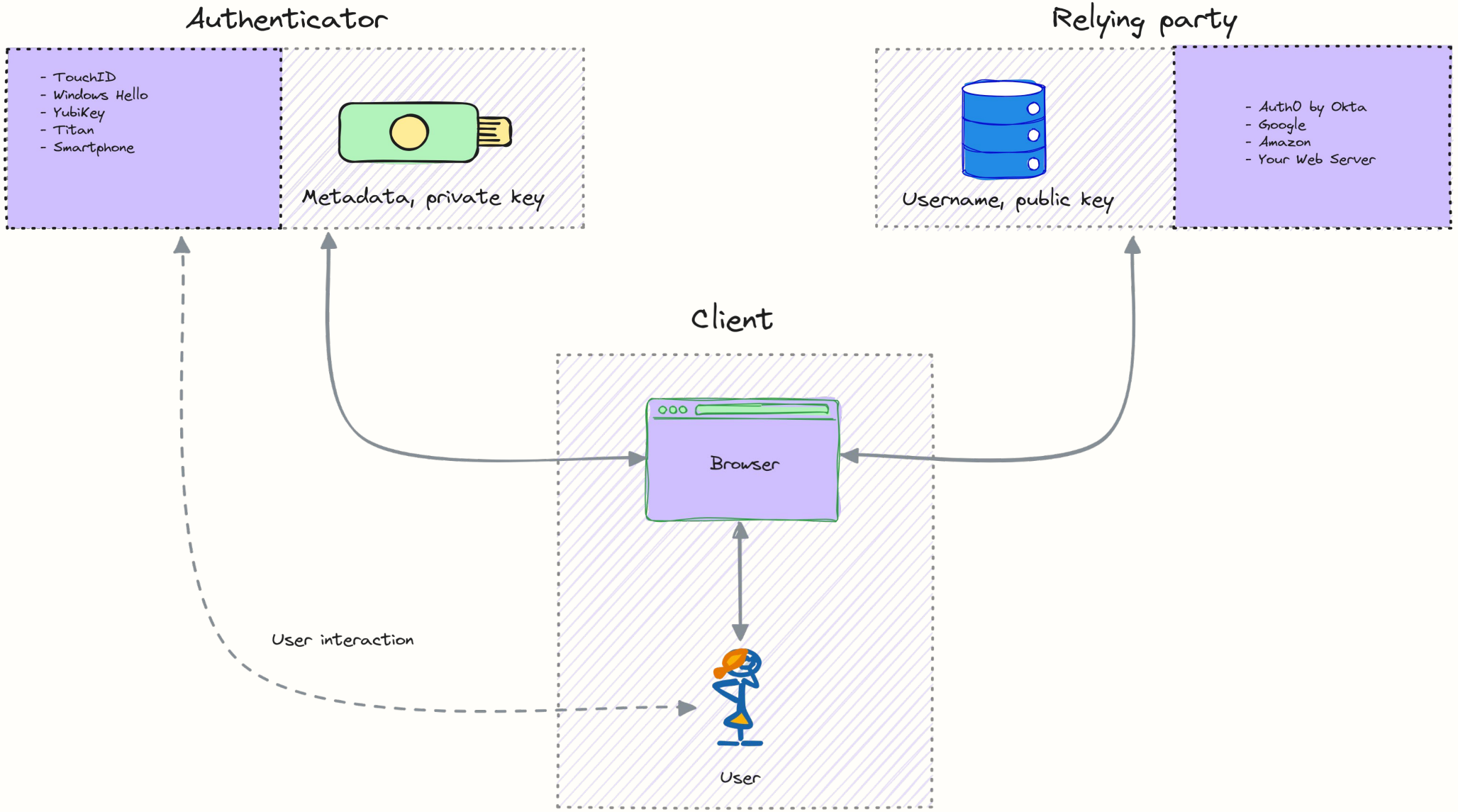
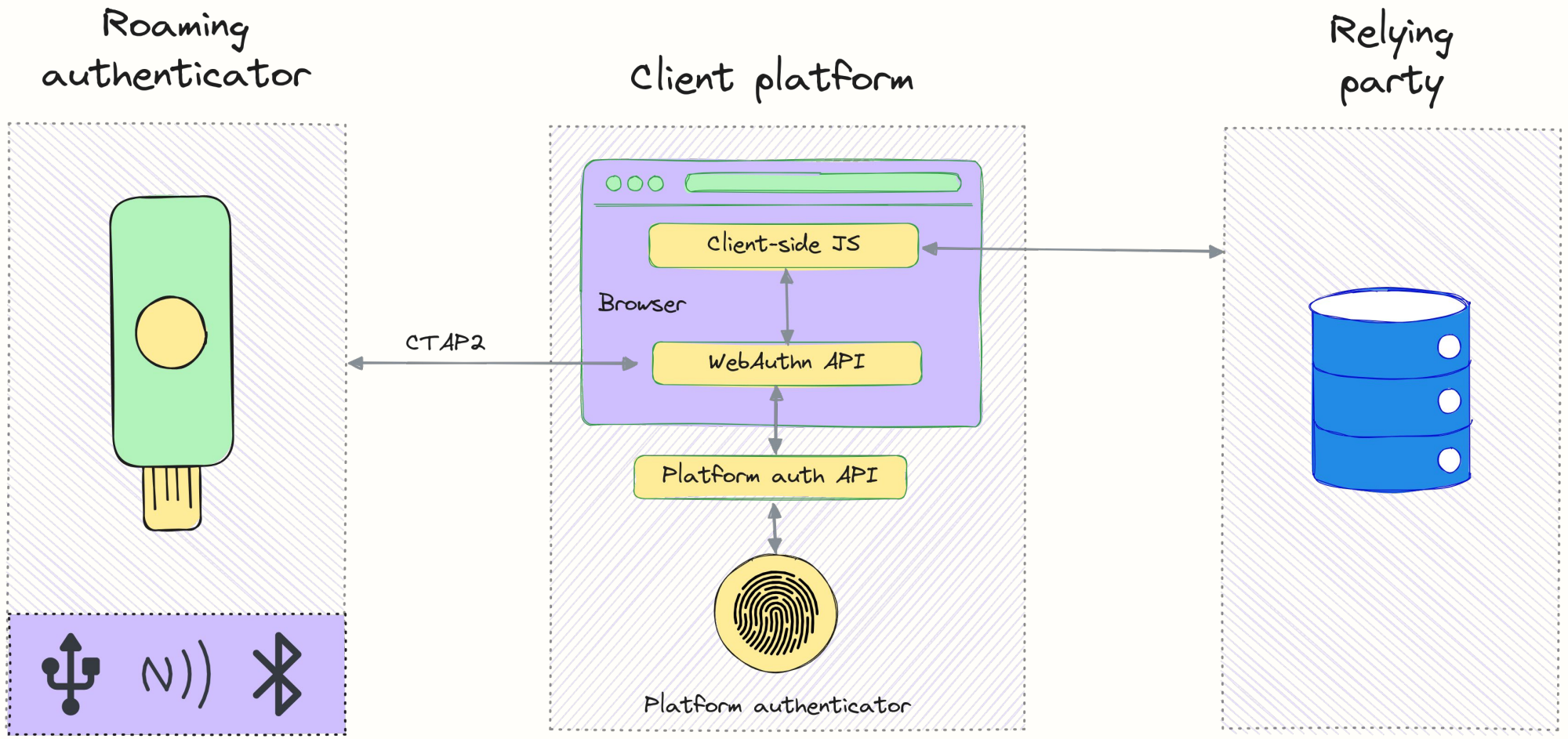


Illustration based on https://webauthn.me/introduction

Client to Authenticator Protocol

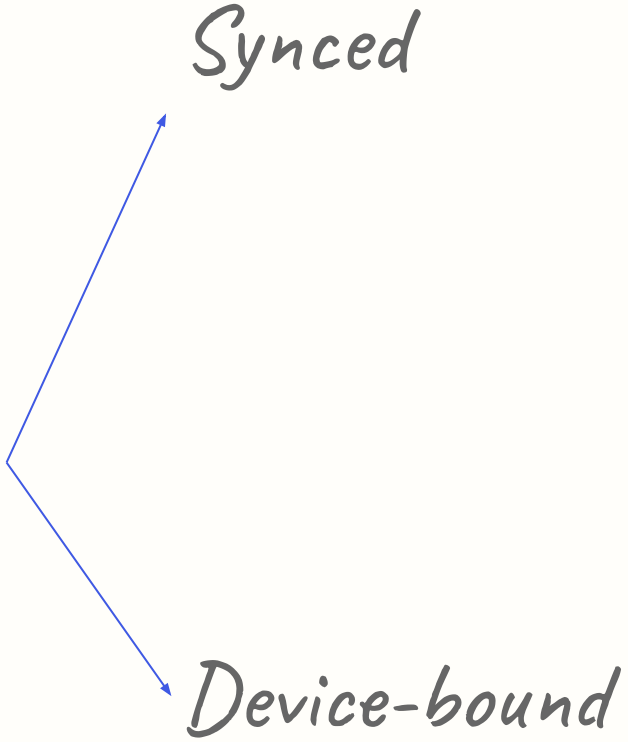
Communicate with authenticators over USB, NFC, and Bluetooth


Roaming authenticator
Client platform
Relying party
CTAP2
Client-side JS
Browser
WebAuthn API
Platform auth API
Platform authenticator

Passkeys == Discoverable passwordless FIDO credentials

It uses asymmetric public key cryptography

Passkeys



Synced

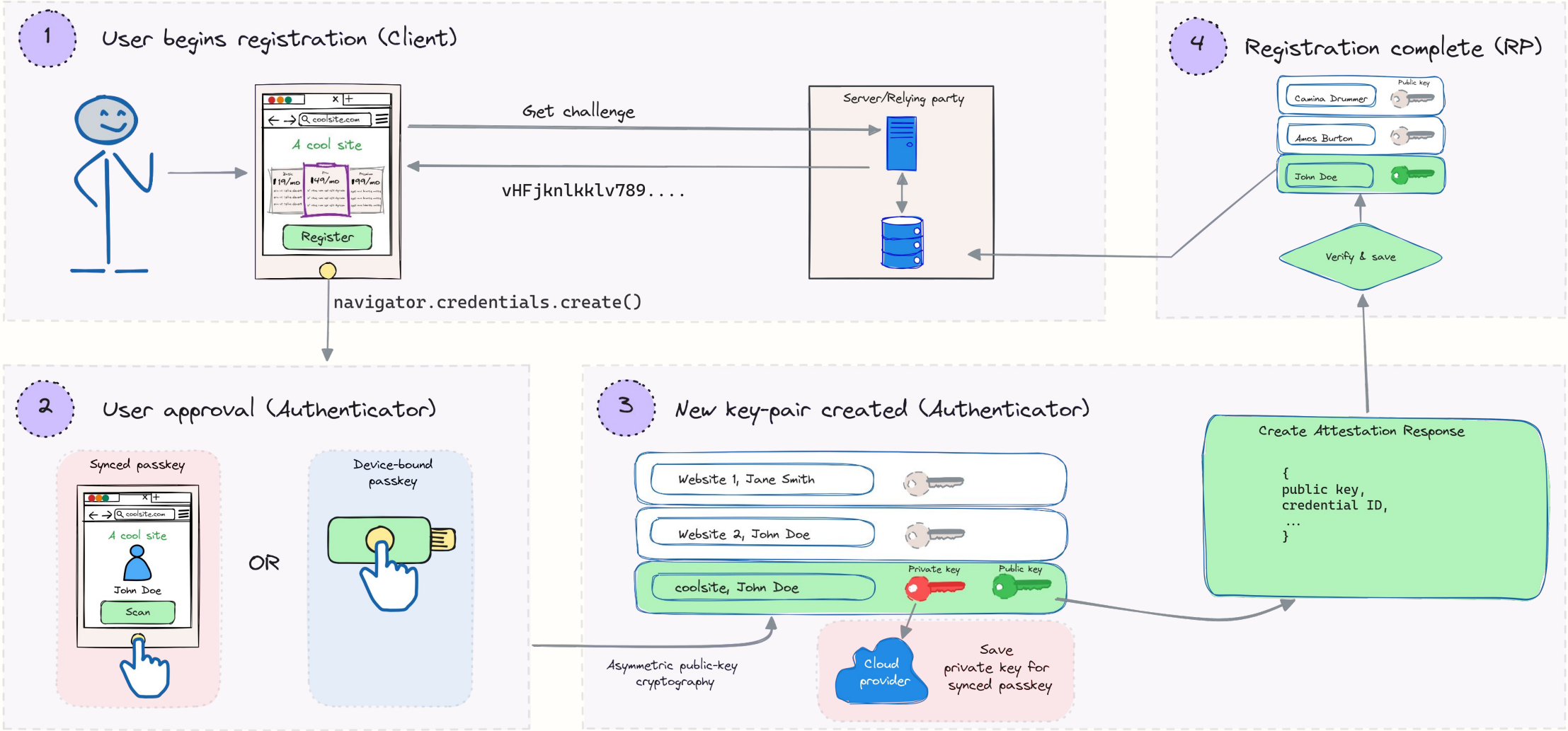
- Private key synced between devices in same ecosystem and backed up to cloud
- Better usability
- One time enrollment
- Can be restored on device loss or on new device
- Less secure than device-bound passkeys

Device-bound

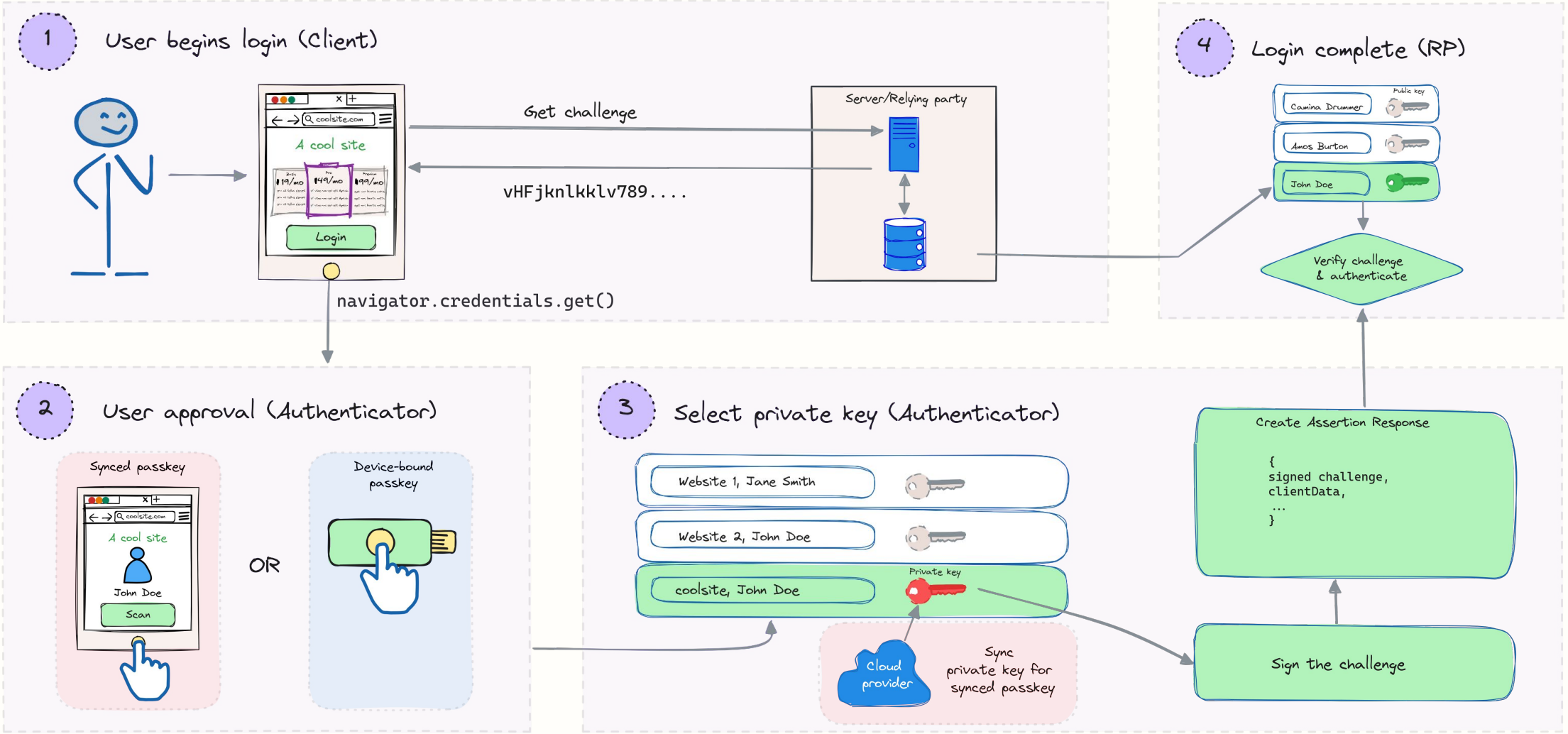
- Private key stored only on the device
- Not as convenient as synced passkeys
- Each device needs enrollment
- No recovery or backups
- Most secure option

How does the magic happen?

Registration flow



Authentication flow



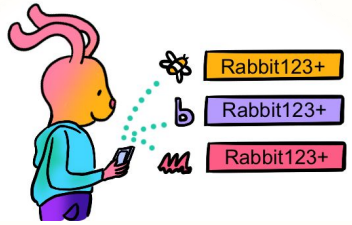
Why passkeys?

Knowledge-based

Phishing

Remote replay

Data breach

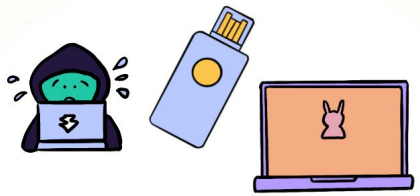


Reuse & share

Password management

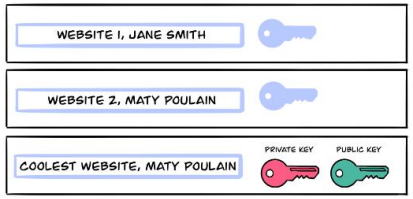
Discoverable

Phishing resistant

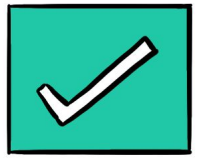


Remote attack resistant

Breach resistant

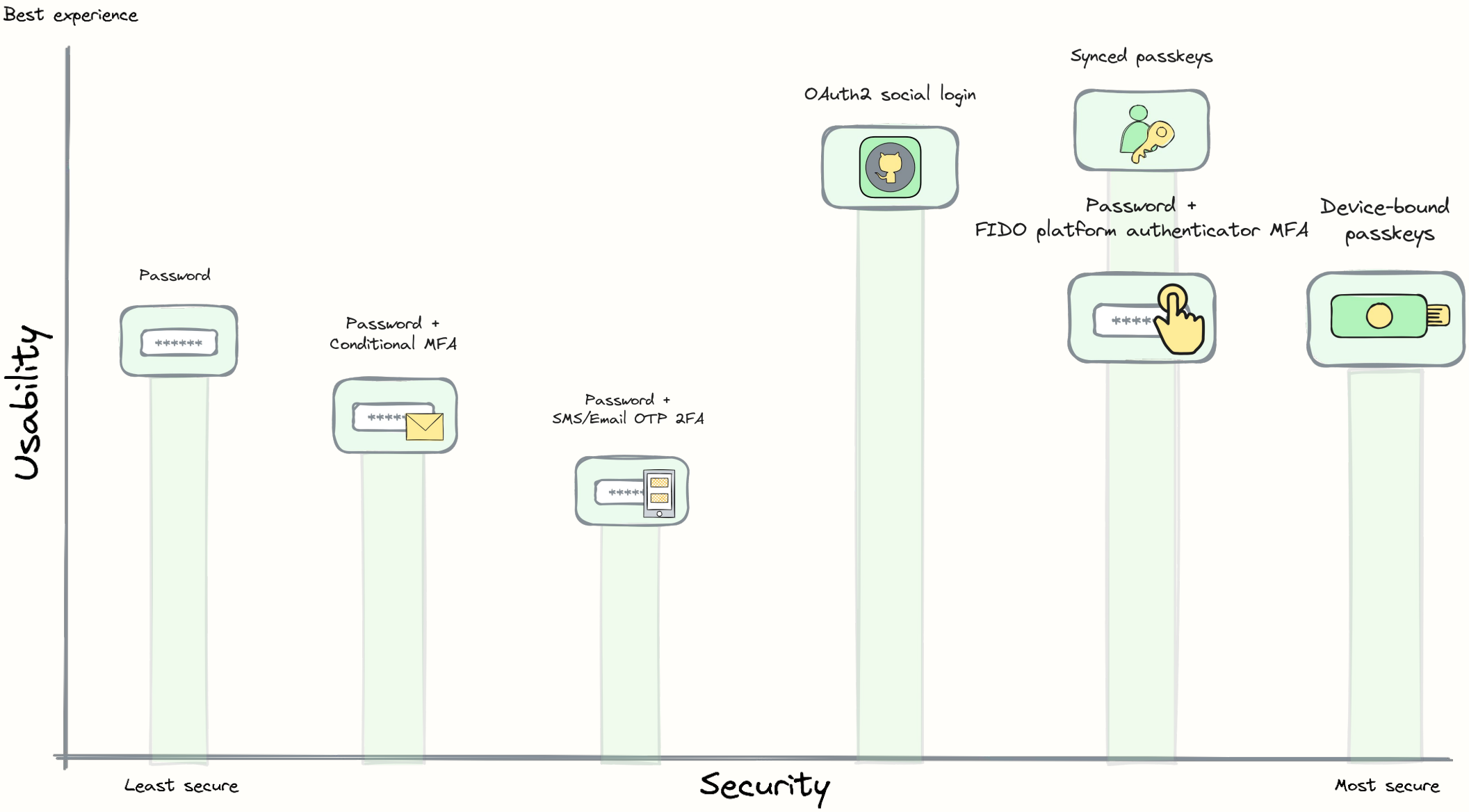


Not reusable & shareable*



Easier to maintain

Passkeys security and usability spectrum



Challenges

- OS/Browser support
- Cloud vendor reliance
- Enterprise use cases
- Reset & recovery

Passkeys in Spring Boot web app using an IdP

Auth0 by Okta as IdP

```
# Create a Spring Boot web app
$ curl -G https://start.spring.io/starter.tgz \
     -d dependencies=web,okta -d baseDir=passkey-demo | tar -xzvf -

# Add controller for @GetMapping("/")

# Create an Auth0 account and configure tenant to enable passkeys
# Login to the tenant
$ auth0 login

# Create an Auth0 app
$ auth0 apps create \
  --name "Spring Boot Passkeys" \
  --description "Spring Boot Example" \
  --type regular \
  --callbacks http://localhost:8080/login/oauth2/code/okta \
  --logout-urls http://localhost:8080 \
  --reveal-secrets


# Update OIDC credentials
# Start the app
$ ./gradlew bootRun
```

a0.to/spring-passkey

Passkeys in Java apps

WebAuthn4j

- FIDO2 conformant
- Supports attestation validation
- Supports all attestation formats
- Suitable for relying party server implementation
- Supports passkeys
- Used by Keycloak
- Has Spring Security support
- Kotlin friendly

java-webauthn-server

- Not 100% FIDO2 conformant
- Supports attestation validation
- All attestation formats not supported
- Suitable for relying party server implementation
- Supports passkeys
- From Yubico

Passkeys with Spring Security and WebAuthn4j

Spring Boot web app as a relying party server using WebAuthn4j

WebAuthn4J Spring Security

```
# Clone the repo
$ git clone https://github.com/deepu105/webauthn4j-spring-boot-passkeys-demo

# Start the app
$ ./gradlew bootRun
```

a0.to/spring-webauthn

Passkeys with Spring Security

spring-security-webauthn

- Provides default registration and login pages
- Will become a Spring Security core option
- Based on WebAuthn4j
- At experimental stage now
- Expected in Spring Security 6.4 (November, hopefully)

How does it differ from WebAuthn MFA?

Passkeys

- Implemented using WebAuthn and FIDO2
- Can be synced or device-bound
- Discoverable credentials (Resident keys)
- Can be used for account registration as first factor
- Enrollment required only once for synced passkeys

WebAuthn MFA

- Implemented using WebAuthn and FIDO2
- Only device-bound
- Non-Discoverable credentials
- Can only be second factor after account registration with password
- Enrollment required on each device

Resources

https://learnpasskeys.io

https://webauthn.me/passkeys

https://passkeys.dev

https://passkey.org

https://fidoalliance.org/passkeys

Auth0 by Okta

Make login our problem. Not yours.

How we can help:

Authentication

Authorization

Security

Single Sign-On | Adaptive Multi-Factor Authentication | Universal Login | Passwordless | Bot Detection & Prevention | Security Center | Breached Password Detection | Brute Force Protection | FGA

SCAN ME

a0.to/plg_signup

Try Free Today:

Free Plan (forever) $0
Up to 7,500 monthly active users. Unlimited user logins. Includes passkeys support*.
No credit card required.

Special Plans for Startups & Nonprofits

Plans for Everyone

B2C: your users are consumers

B2B: your users are businesses or a mix of businesses and consumers

Enterprise: Best for production applications that need to scale - **Contact Us**

Thank You

Subscribe to our newsletter

a0.to/nl-signup/java

SCAN ME

Try our free Spring Boot microservices workshop

a0.to/spring-boot

SCAN ME