



2023 INTERNET2  
**TECHNOLOGY**  
exchange

Passwordless Authentication

Peter Balcirak, CESNET & Masaryk University

# Authentication

- Validation process of proclaimed identity
- Factors
  - Something you **know**
  - Something you **have**
  - Something you **are**

# Passwords

- Easy to use
- Easy to implement
- Relatively secure on server side
- Not secure on user side

# Passwords - Easy to break

Email\*  
422570@muni.cz

---

Password \*  
password 

---

Confirm password \*  
password 

---

 Password must be at least 8 characters long. Please **avoid using accented characters**. It might not be supported by all backend components and services.

# Passwords - Easy to break

The image displays two overlapping screenshots of a registration form. The left screenshot shows the 'Email\*' field with '422570@muni.cz', the 'Password \*' field with 'password', and the 'Confirm password \*' field with 'password'. Below these fields is a warning message: 'Password must be at least 8 characters long. Please avoid using accented characters. It might not be supported by all backend components and services.' The right screenshot shows the same 'Email\*' field, but the 'Password \*' field contains '01234567' and has a visibility icon. The 'Confirm password \*' field also contains '01234567' and has a visibility icon. Below these fields is a warning message: 'Password must be at least 8 characters long. Please avoid using accented characters. It might not be supported by all backend components and services.'

# Passwords - Easy to break

The image displays three overlapping registration forms, each with a different password choice and associated warning message.

- Form 1 (Left):** Email: 422570@muni.cz; Password: password; Confirm password: password. Warning: *Password must be at least 8 characters long. Please avoid using **accented characters**. It might not be supported by all backend components and services.*
- Form 2 (Middle):** Email: 422570@muni.cz; Password: 01234567; Confirm password: 01234567. Warning: *Password must be at least 8 characters long. Please avoid using **accented characters**. It might not be supported by all backend components and services.*
- Form 3 (Right):** Email: 422570@muni.cz; Password: AliceBob; Confirm password: AliceBob. Warning: *Password must be at least 8 characters long. Please **avoid using accented characters**. It might not be supported by all backend components and services.*

# Passwords - Easy to forget

Email\*

422570@muni.cz

Password \*

1kuX3mdk0Lxr



Confirm password \*

1kuX3mdk0Lxr



 Password must:

- contain only printable (non-accented) characters
- be at least 12 characters long
- consist of at least 3 of 4 character groups:
  - lower-case letters
  - upper-case letters
  - digits
  - special characters

# Passwords - Easy to forget

UČO

Primary password

Remember me

LOG IN

> [I have trouble logging in](#)

# Passwords - Easy to forget

---

MUNI Unified  
Login

Wrong UČO or password

UČO

Primary password

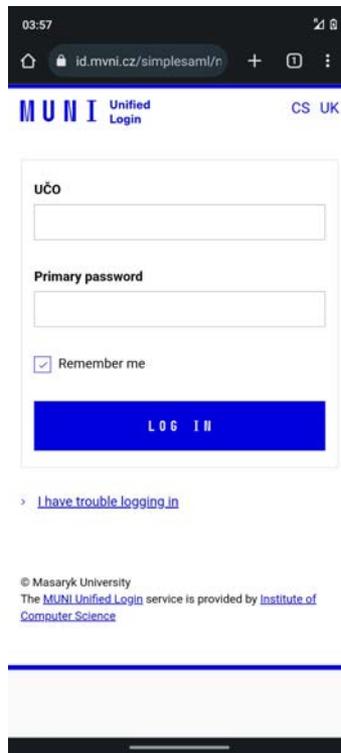
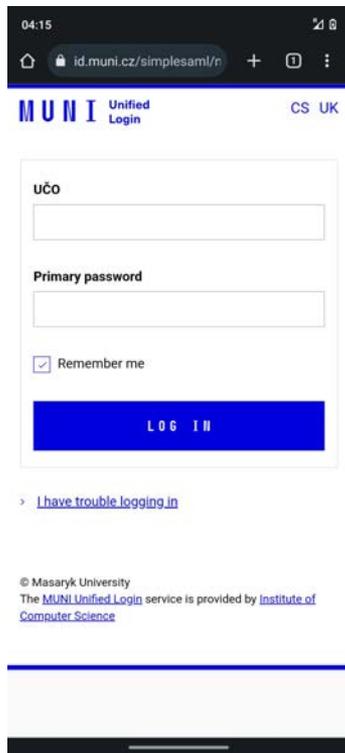
Remember me

LOG IN

> [I have trouble logging in](#)

© Masaryk University  
The MUNI Unified Login service is provided by [Institute of Computer Science](#)

# Passwords - Phishing



# Passwords - Phishing



Učo



Učo

## What can we do?

- Increase password requirements
- Implement weak pass check
- Force to use password managers

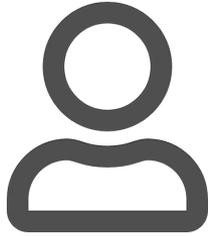
## What can we do?

- Increase password requirements
- Implement weak pass check
- Force to use password managers
- Enable Multi-Factor Authentication (MFA)

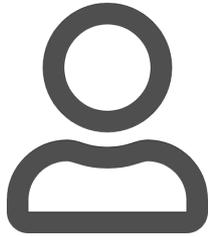
# MFA

- Combines more authentication factors (at least 2)
- Usually something you know + something you have
- Improves security
- Should not be difficult for users

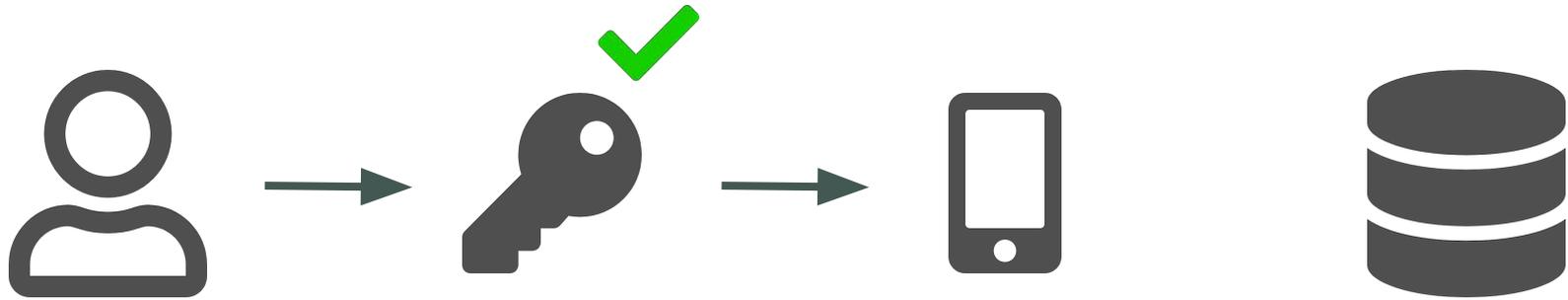
# MFA - User flow



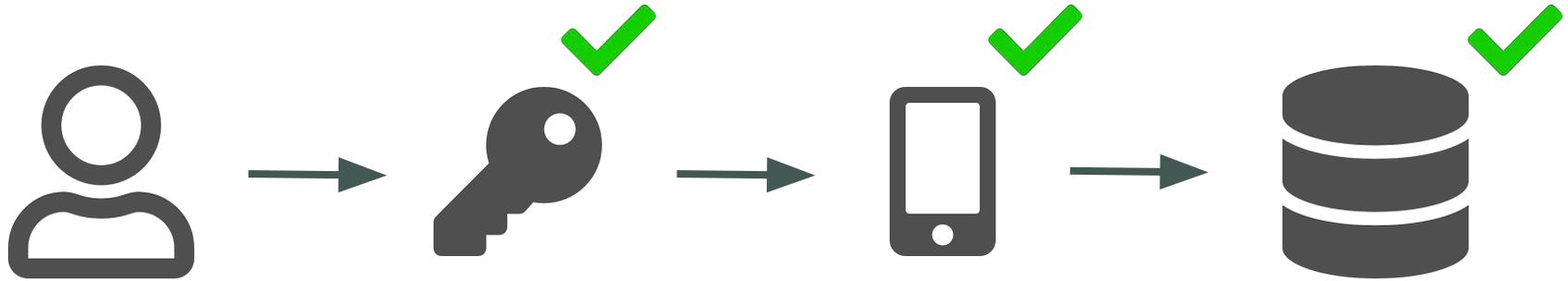
# MFA - User flow



# MFA - User flow



# MFA - User flow



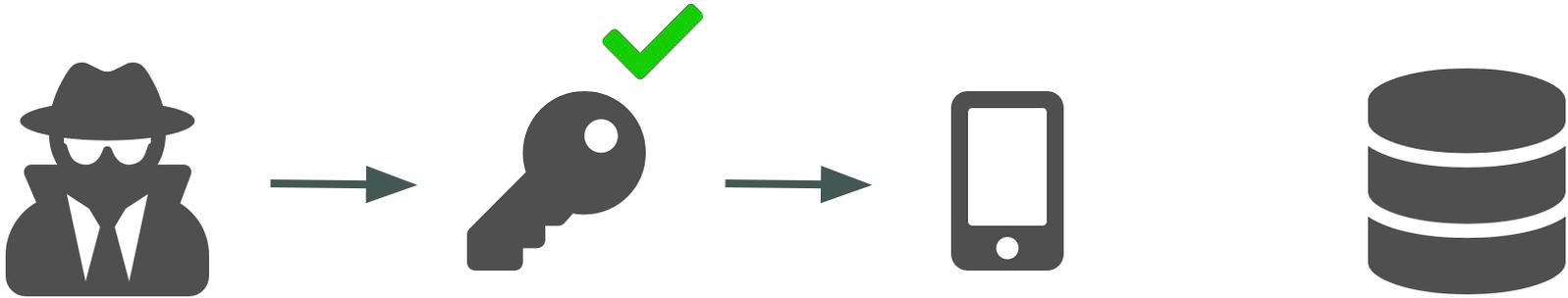
# MFA - Attacker flow



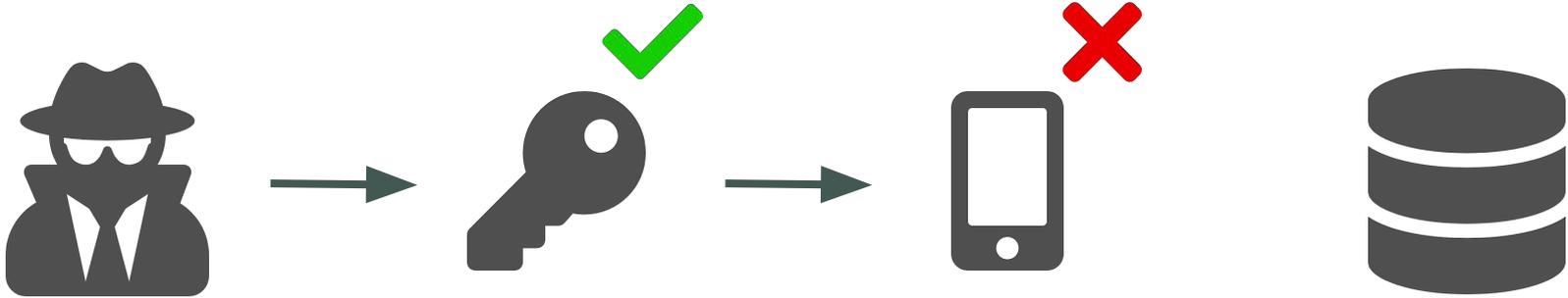
# MFA - Attacker flow



## MFA - Attacker flow



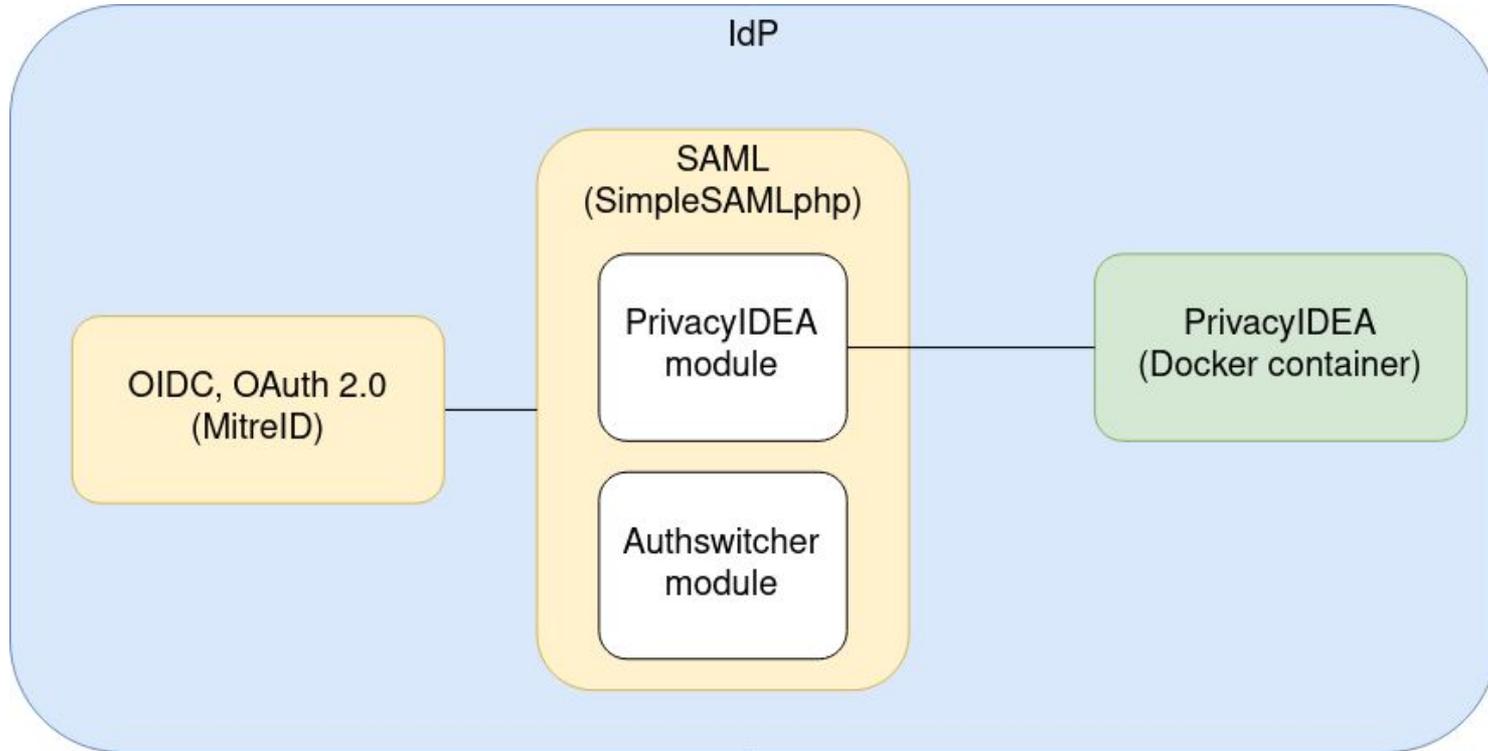
# MFA - Attacker flow



## MFA - Our solution

- Built on top of the PrivacyIDEA
  - Open-source project
  - Easy to integrate and extend
  - Large community
  - UI for token management
  - API for authentication
- Solution follows the REFEDS MFA Profile
- Currently supports TOTP and WebAuthn

# MFA - Architecture



# MFA - PrivacyIDEA SimpleSAMLphp module



# MFA - Authswitcher SimpleSAMLphp module



# MFA - Token management

The screenshot displays the MFA Token Management interface. At the top, there is a navigation bar with a logo 'M', a refresh icon, and a user ID '422570'. Below this, a blue bar contains the text 'All tokens'. Underneath, there is a link 'Enroll Token'. The main content area shows 'total tokens: 2' and a table of tokens. The table has columns for serial, type, active, description, failcounter, and rollout state. Two tokens are listed: one with serial 'TOTP04107F21' and another with serial 'TOTP10d45c7124f4fe53656f8754b298e3d1'. Both are of type 'totp' and are active. The first is in an 'enrolled' state, and the second is 'imported'. Both have a failcounter of 0.

serial	type	active	description	failcounter	rollout state
TOTP04107F21	totp	active		0	enrolled
TOTP10d45c7124f4fe53656f8754b298e3d1	totp	active	imported	0	

privacyIDEA 3.8.1

# MFA - Authentication page

---



## Multi-factor authentication

### One time code

Enter a verification code from authenticator app or a recovery code.

© Masaryk University

The [MUNI Unified Login](#) service is provided by [Institute of Computer Science](#)

# MFA - Services management (end user)

- Profile
- Linked accounts
- Services
- Groups
- Privacy
- Authentication**
- Change primary password

## Multi-factor authentication

Save settings

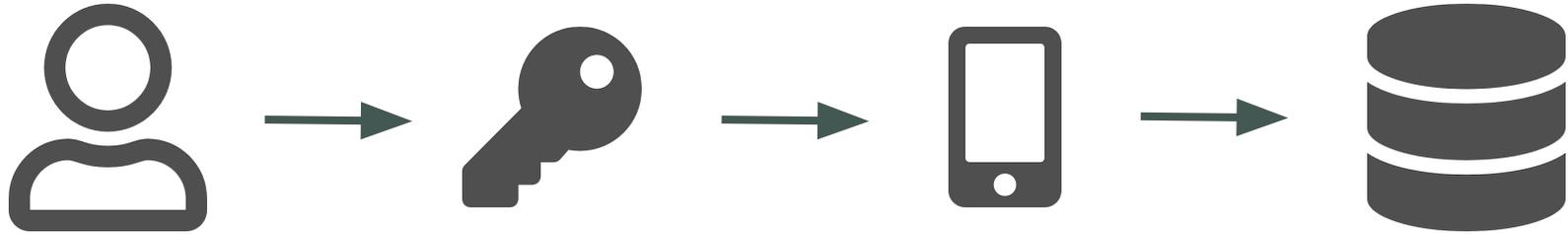
Manage my MFA tokens

- Turn on multi-factor authentication for all services ^
- Information Systems v
- Other
- Web, Webhosting and Marketing v
- Teaching and Learning v
- IT supporting services v
- Cloud and High Performance Computing ^
  - Grafana Dashboard for Sensitive Cloud
  - C4E OpenStack cloud
  - CSIRT-MU Rancher K8s
  - CESNET e-Infrastructure
  - Sensitive Cloud MU
  - CERIT Rancher
- Data management and storage v

# MFA - Summary

- Is it secure?
- Is it user friendly?
- Is it necessary?

# MFA - User flow



# Passwordless - User flow



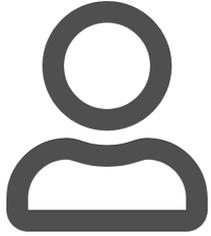
# Passwordless

- Based on something user has
- Trade-off between security and user experience?
- Eliminates the most common vector of attacks

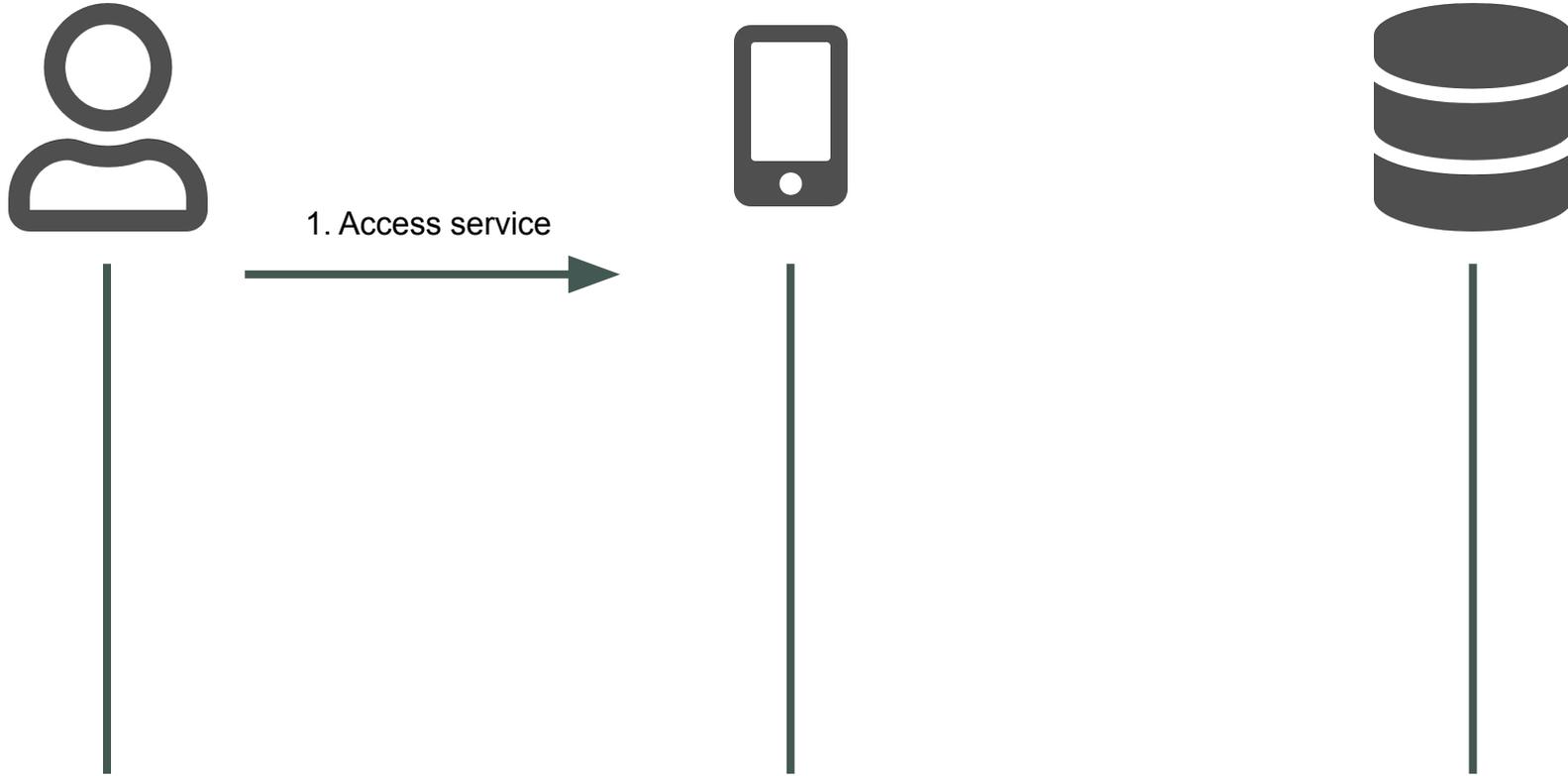
# Passkeys

- Created by W3C and FIDO alliance
- Built on top of the WebAuthn standard
- Can be synchronised between devices
- Does not necessarily require username
- Highly phishing resistant

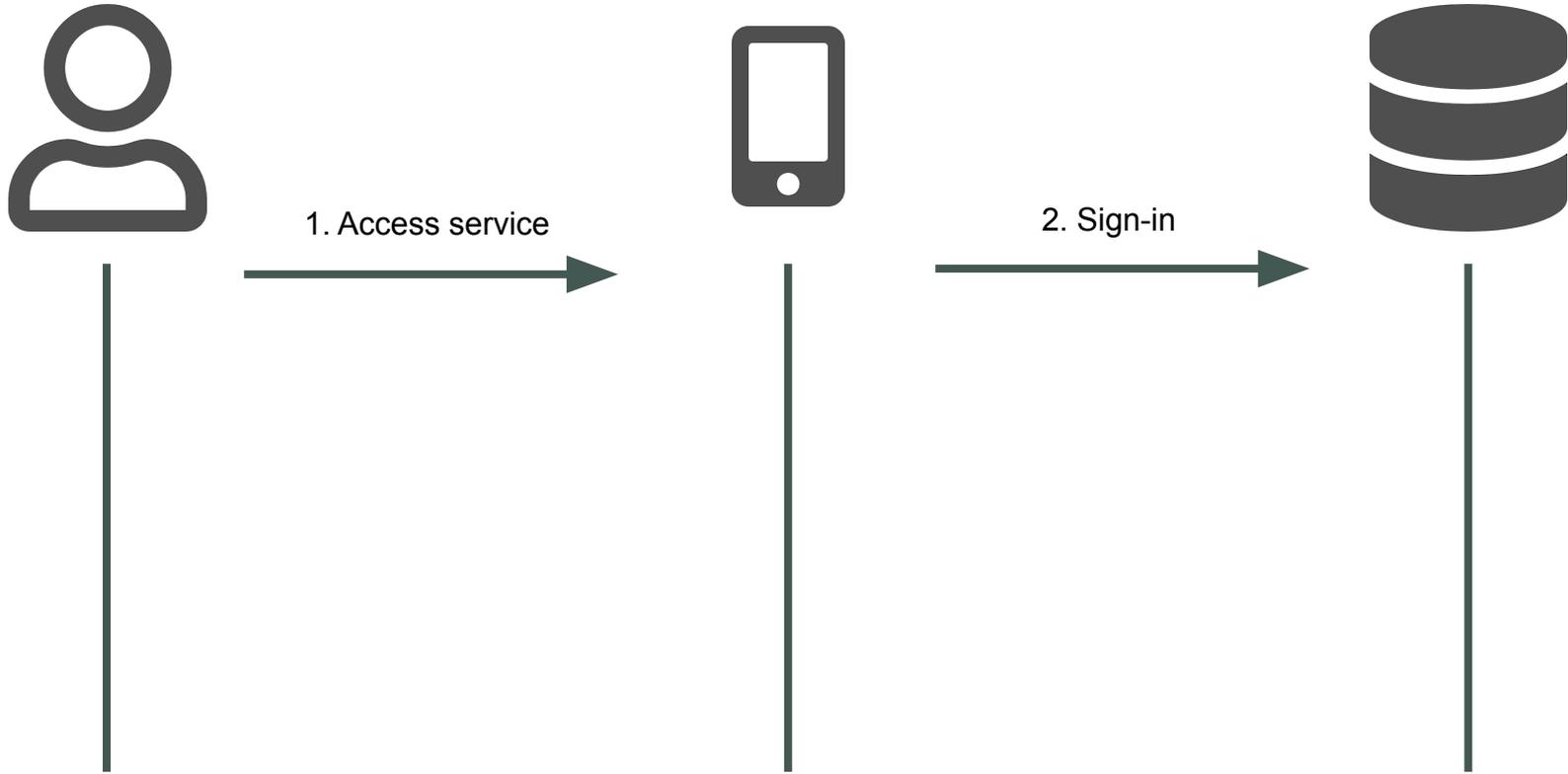
# Passkeys - Registering a new passkey



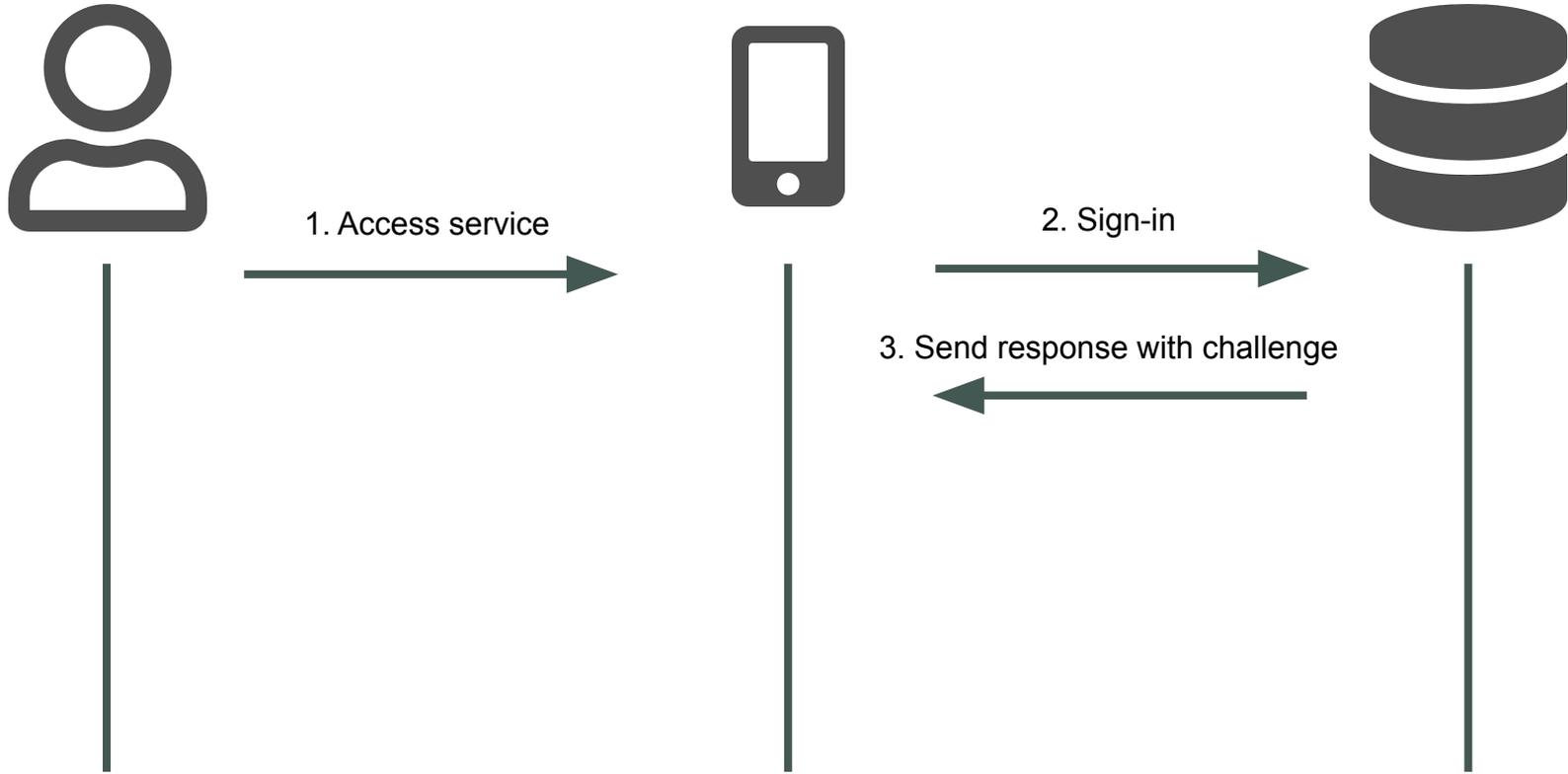
# Passkeys - Registering a new passkey



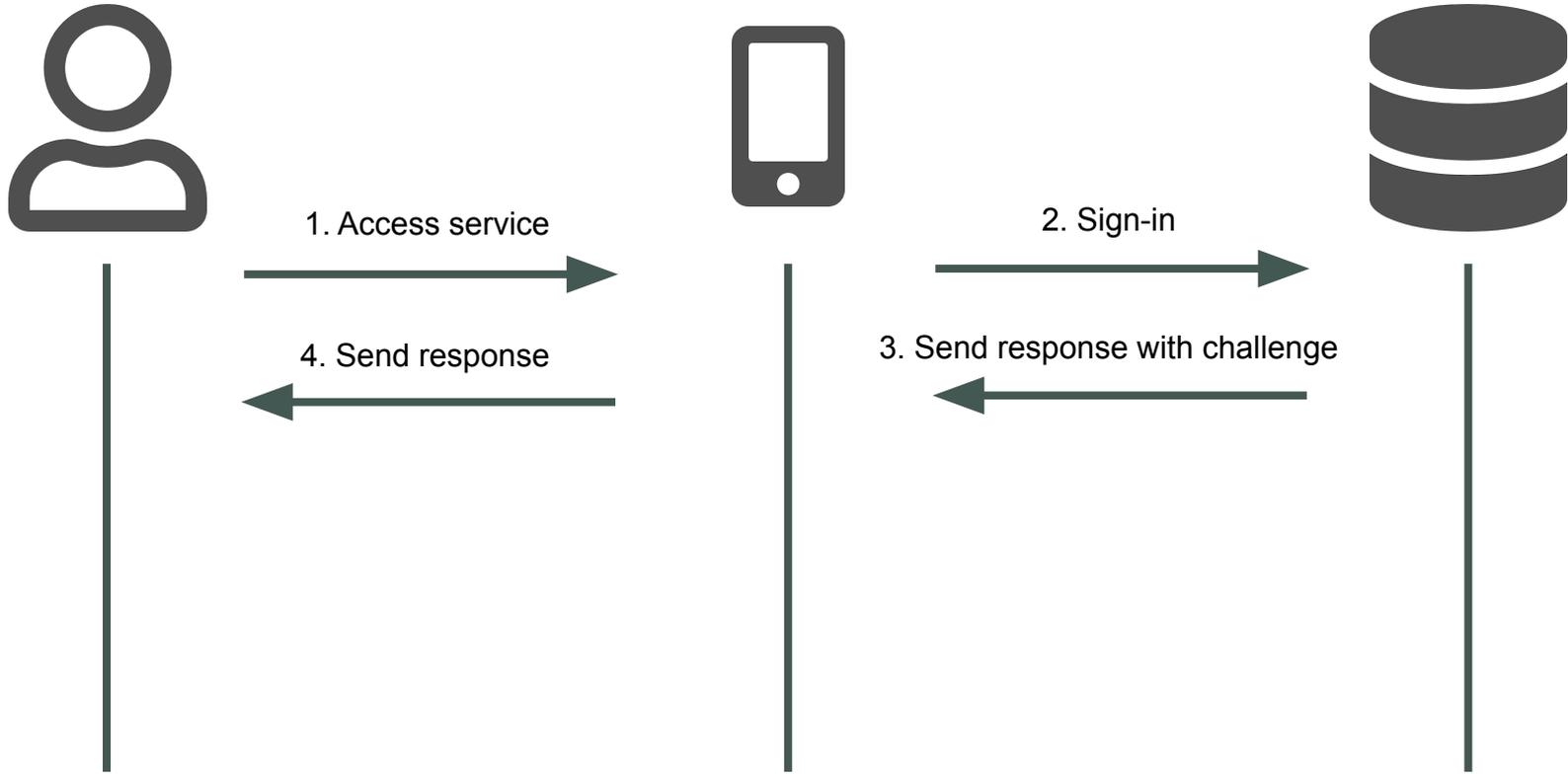
# Passkeys - Registering a new passkey



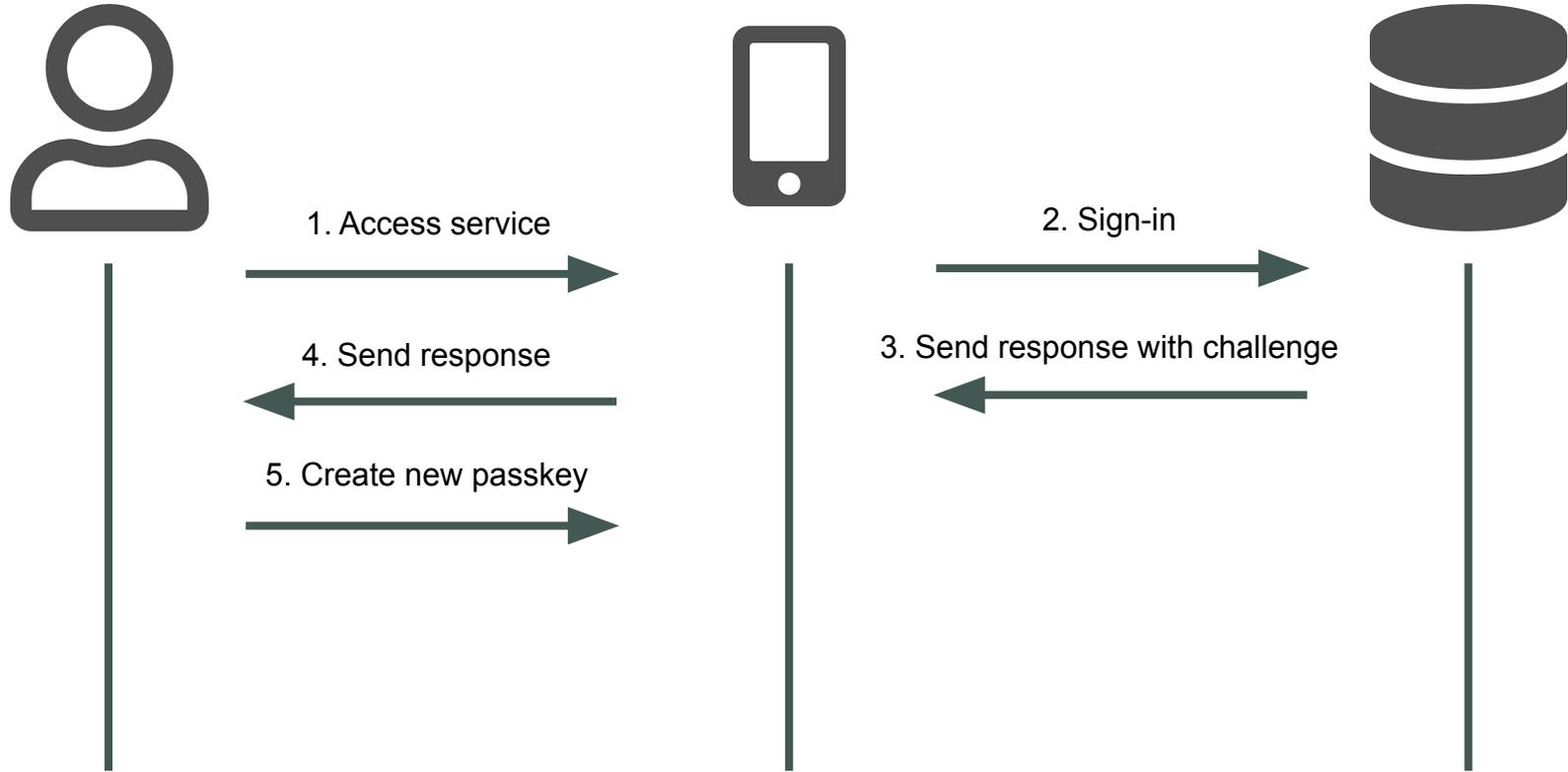
# Passkeys - Registering a new passkey



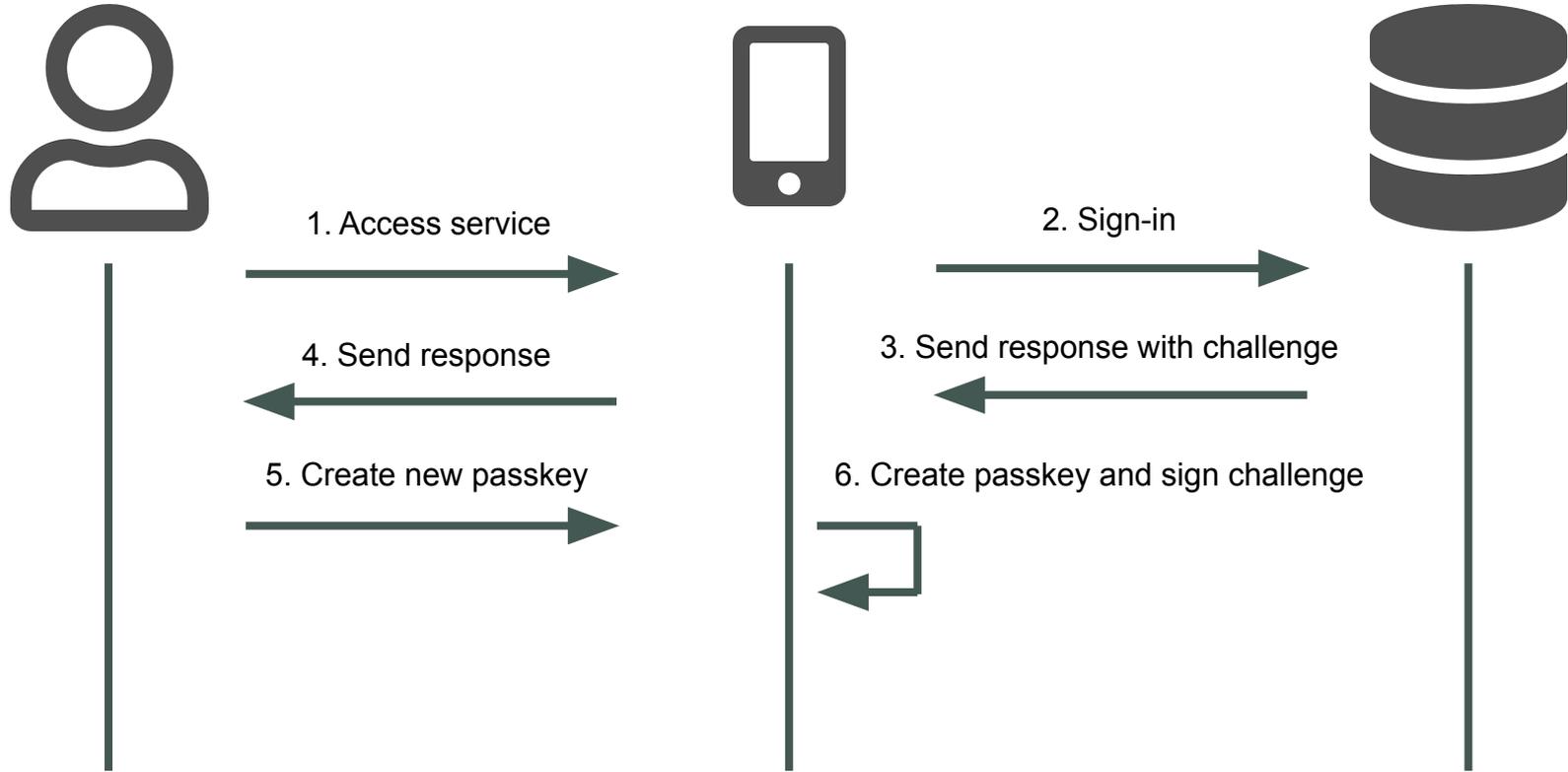
# Passkeys - Registering a new passkey



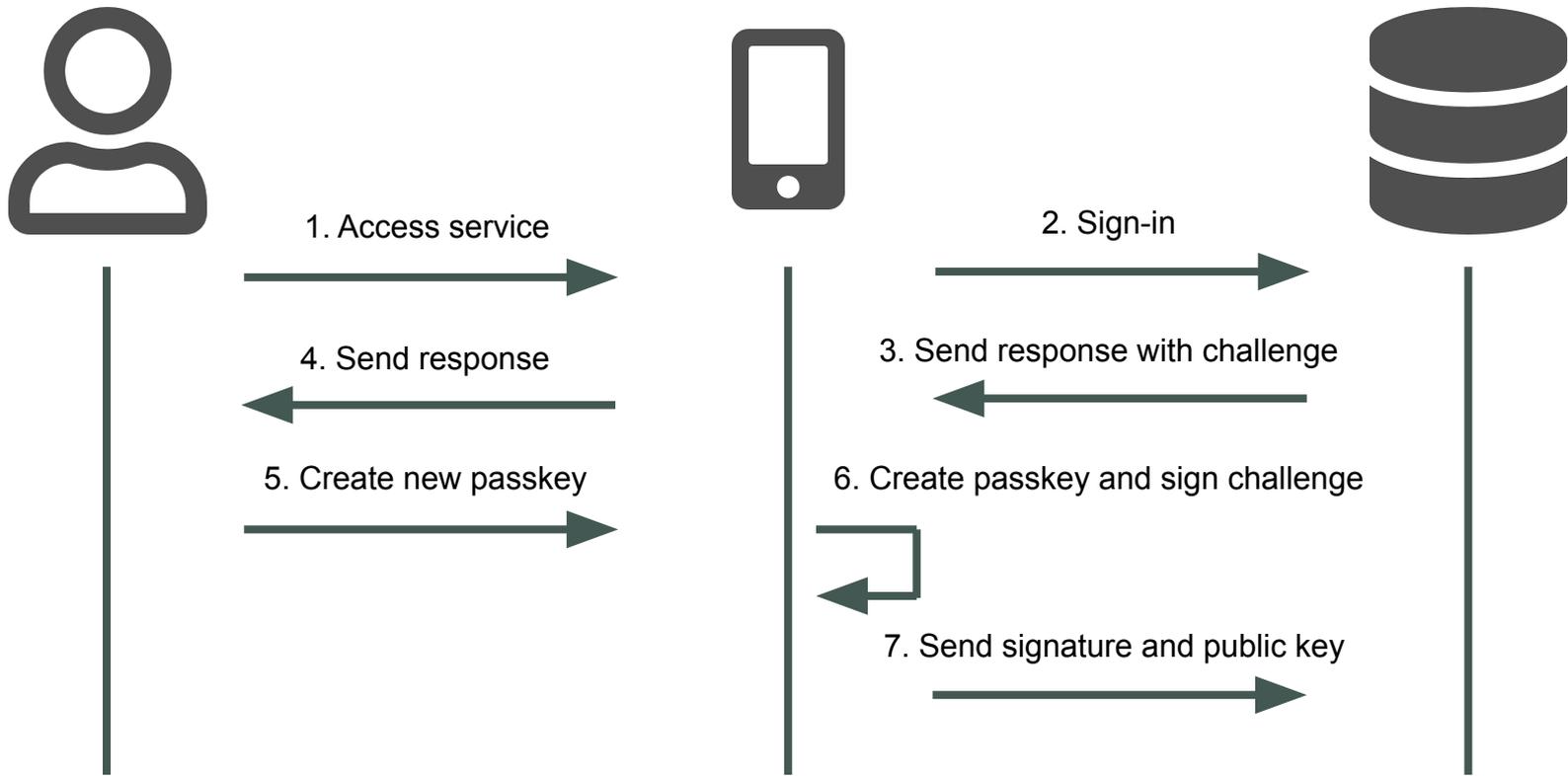
# Passkeys - Registering a new passkey



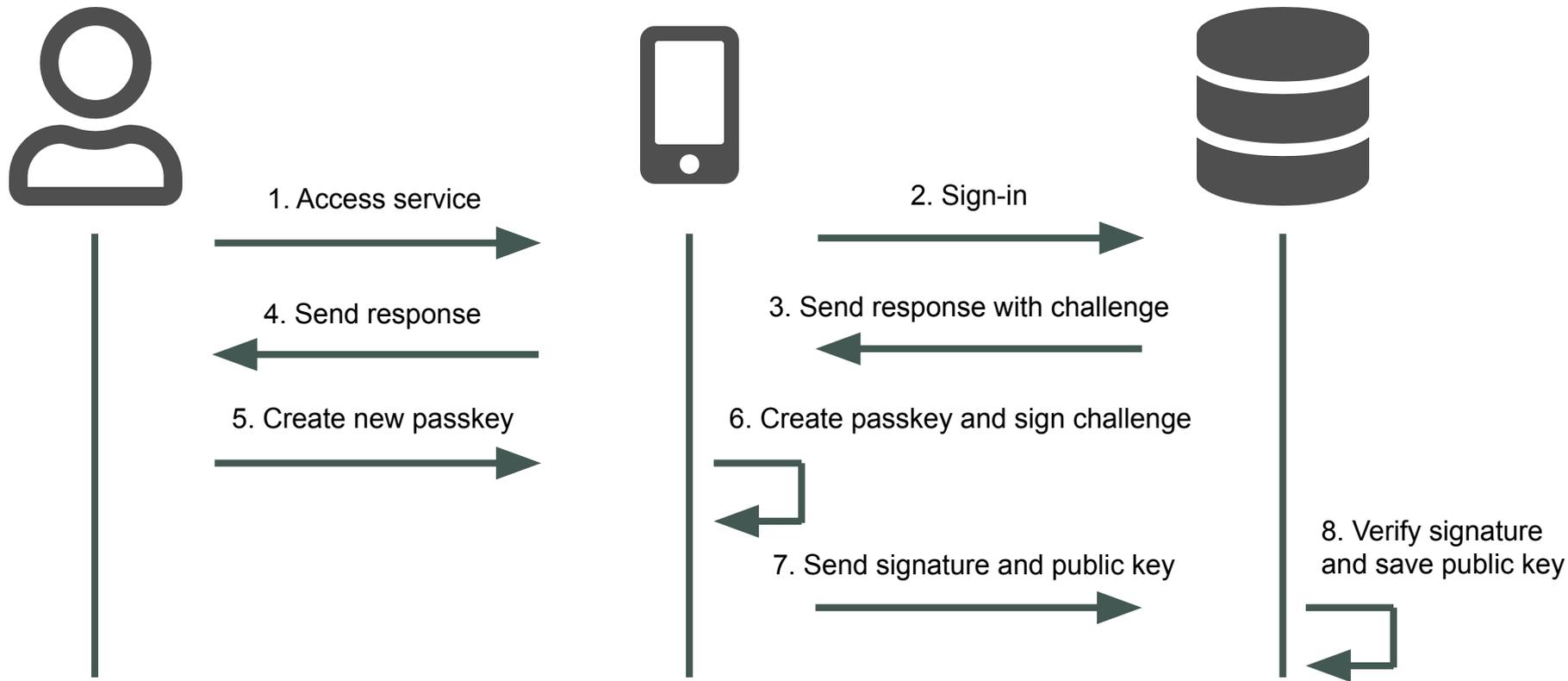
# Passkeys - Registering a new passkey



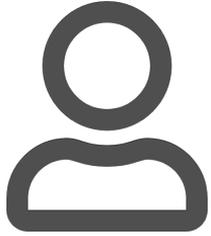
# Passkeys - Registering a new passkey



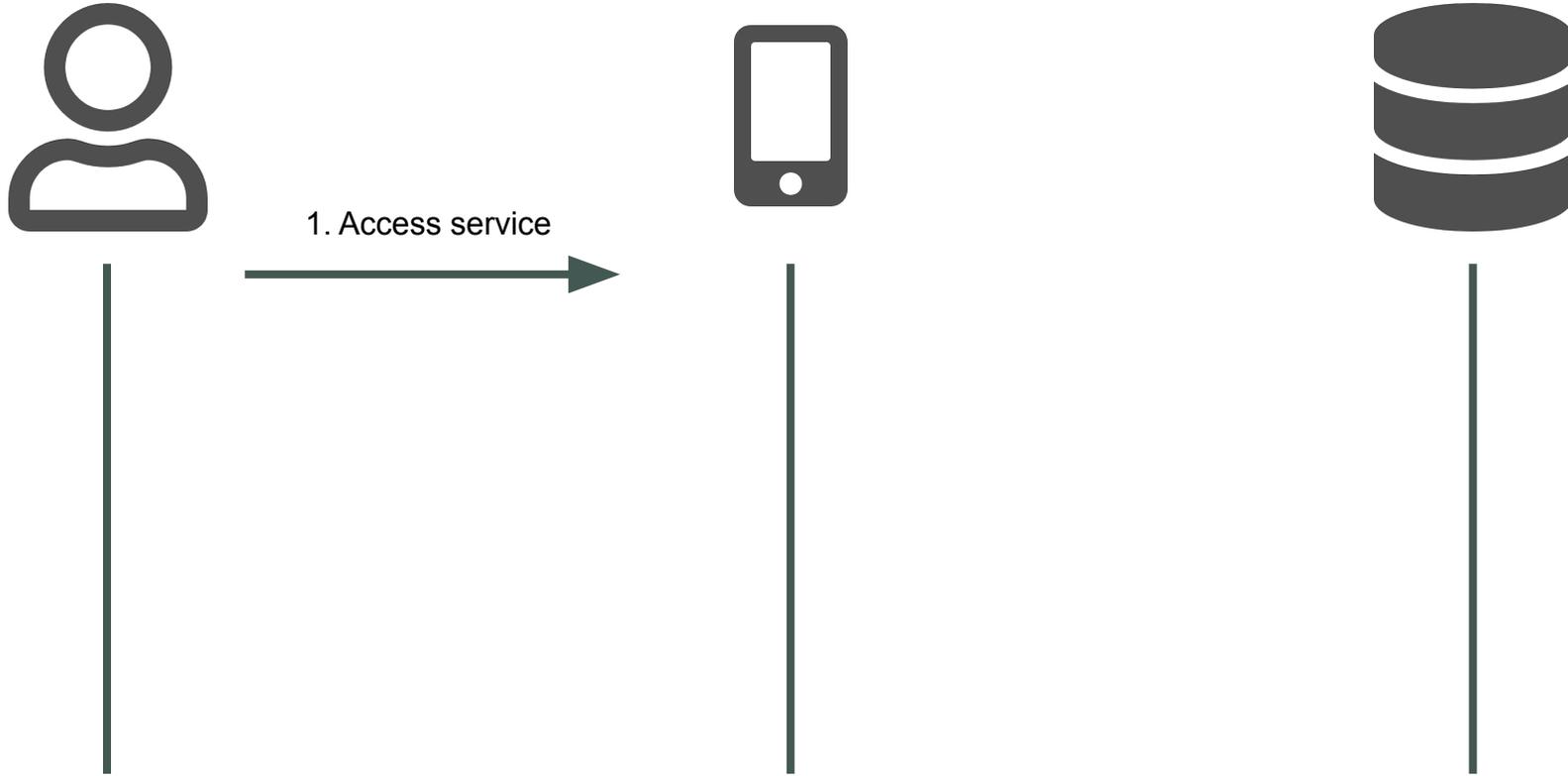
# Passkeys - Registering a new passkey



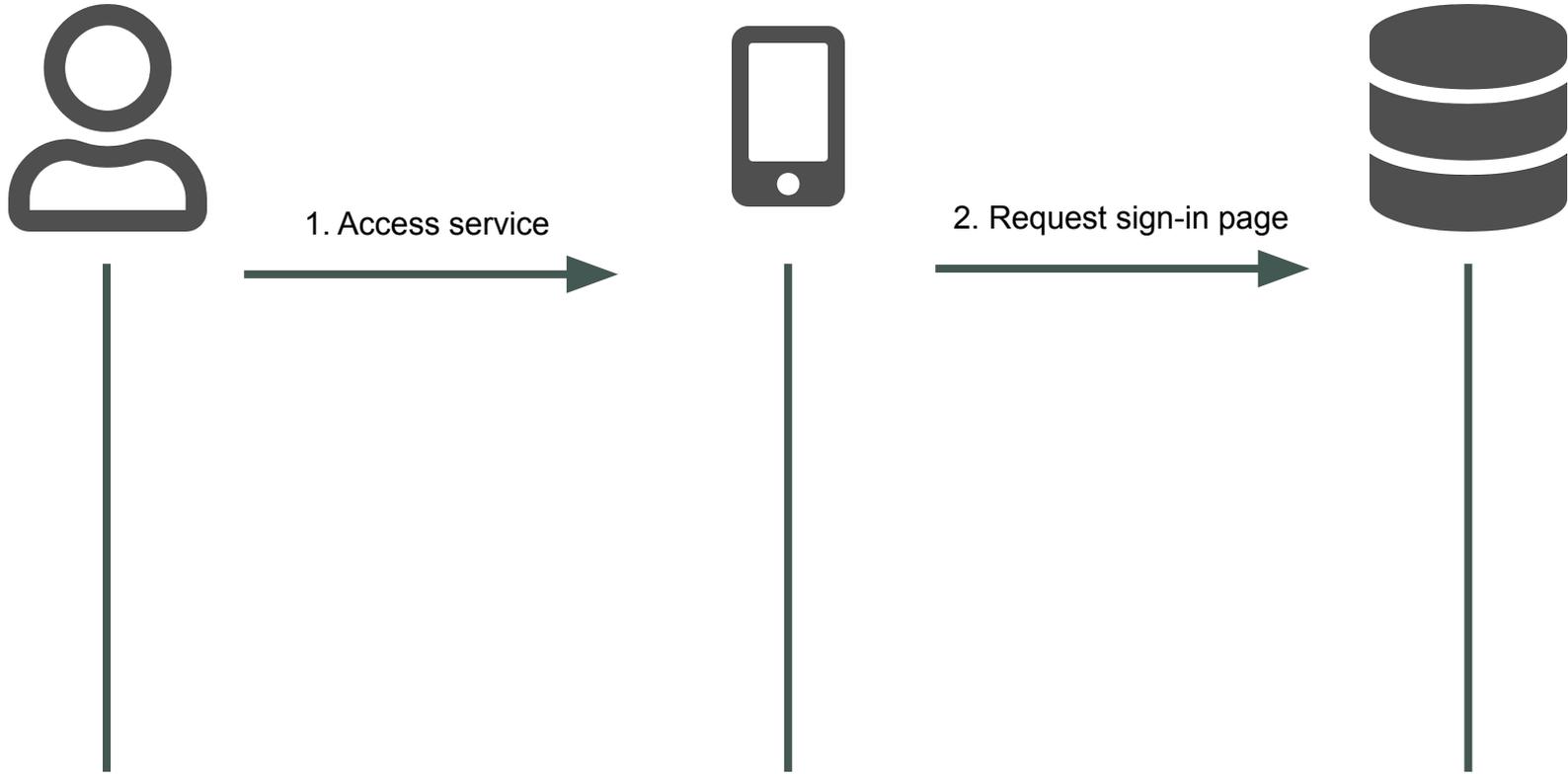
# Passkeys - Sign in with a passkey



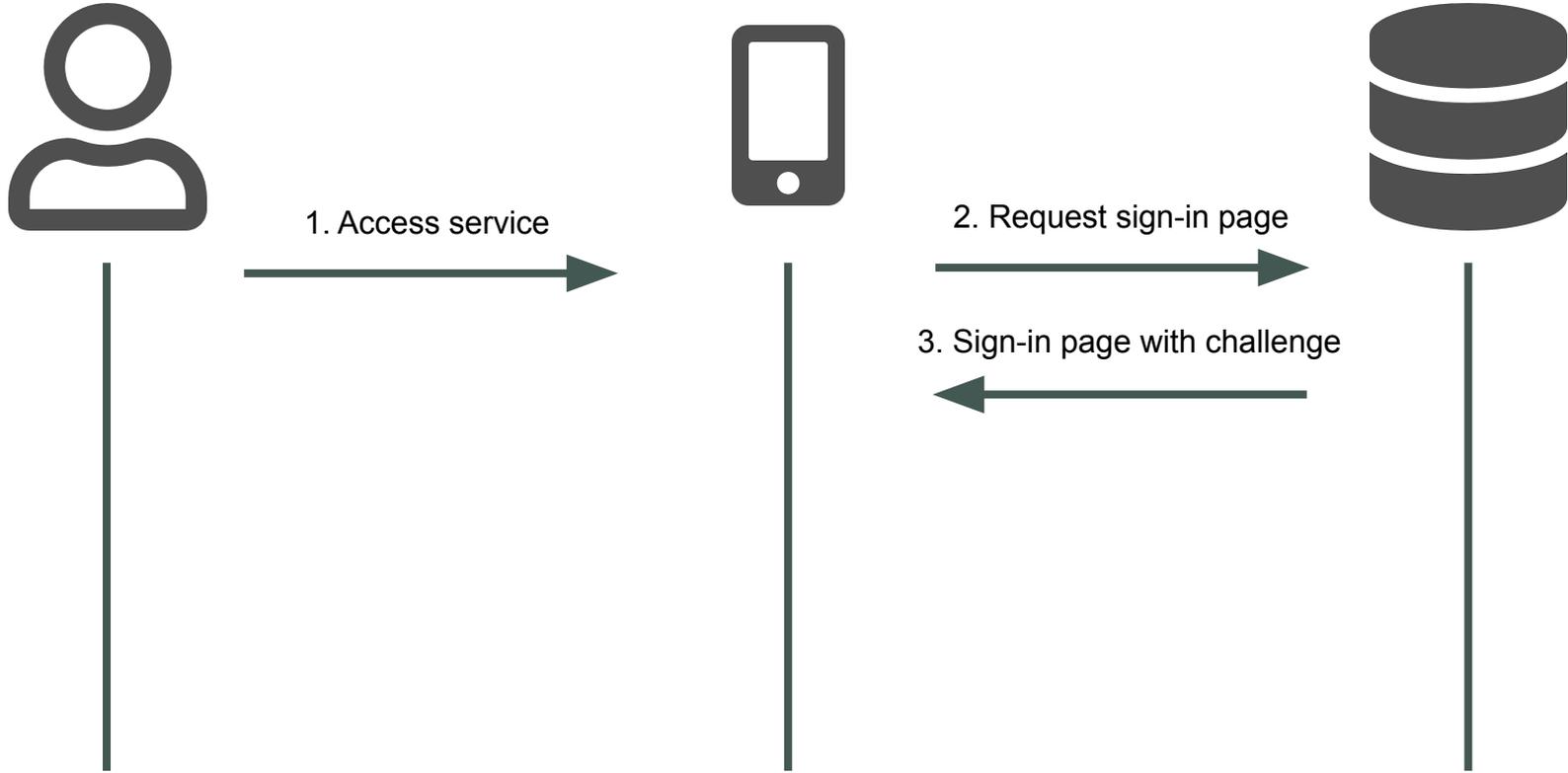
# Passkeys - Sign in with a passkey



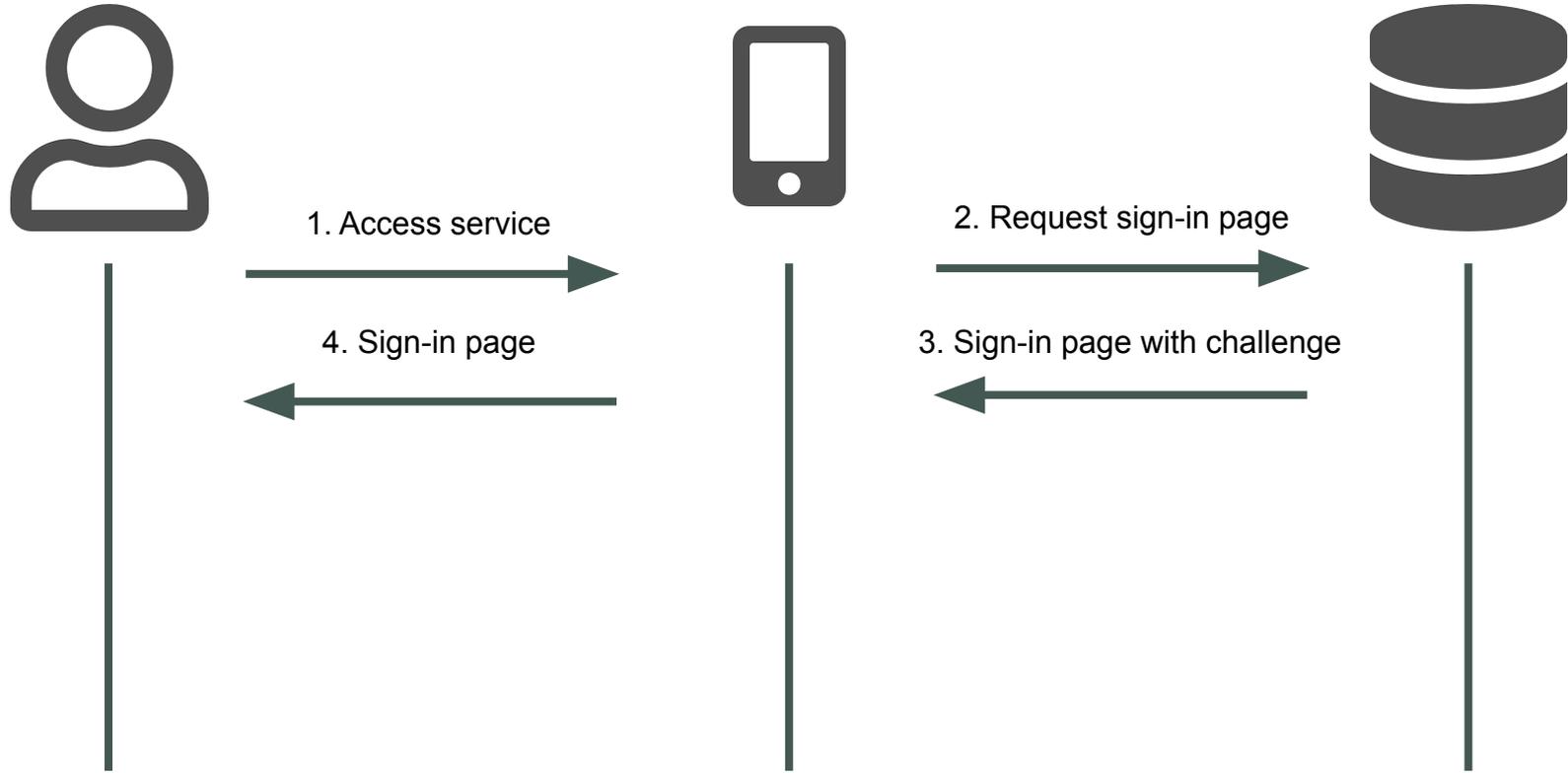
# Passkeys - Sign in with a passkey



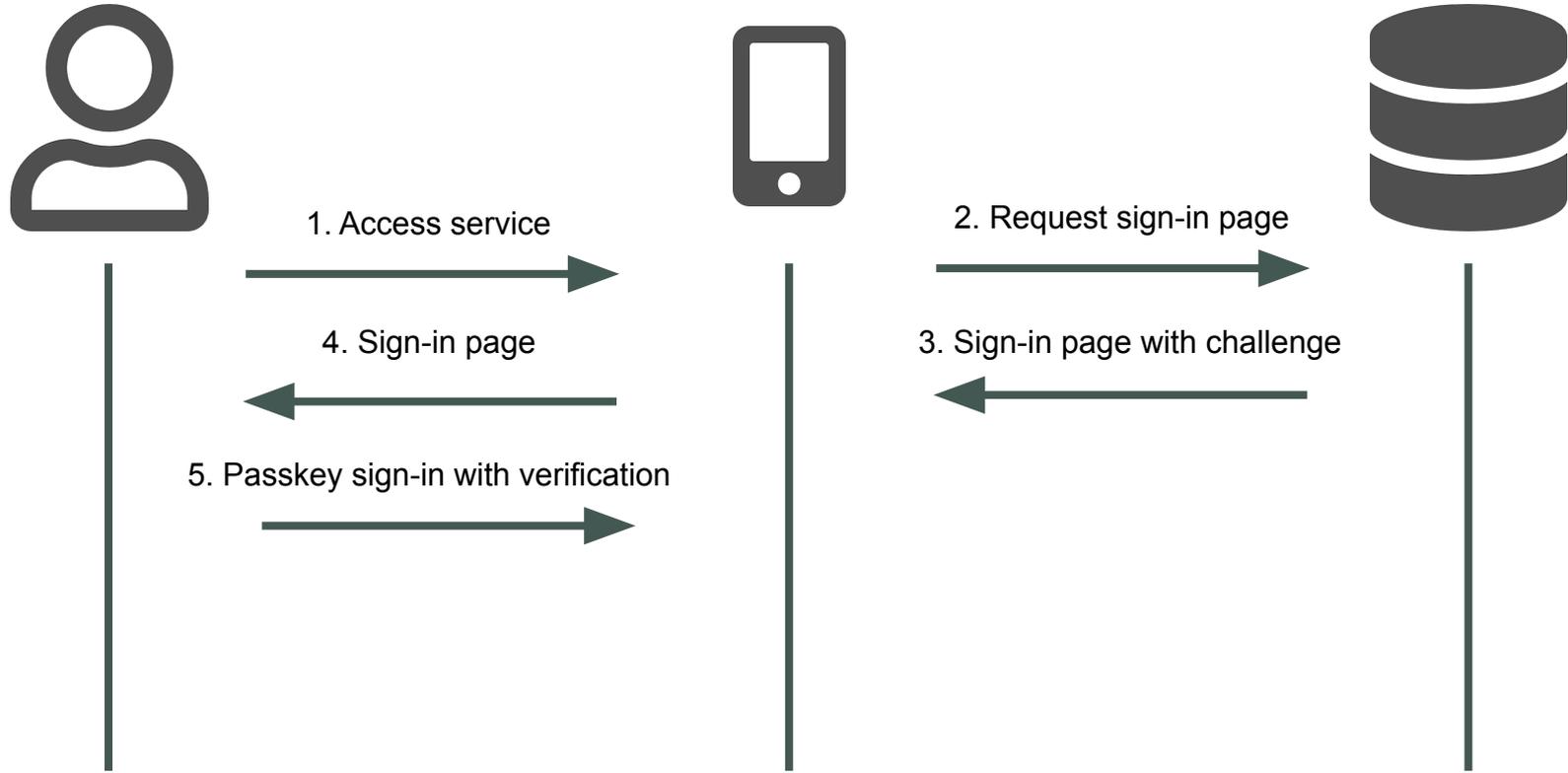
# Passkeys - Sign in with a passkey



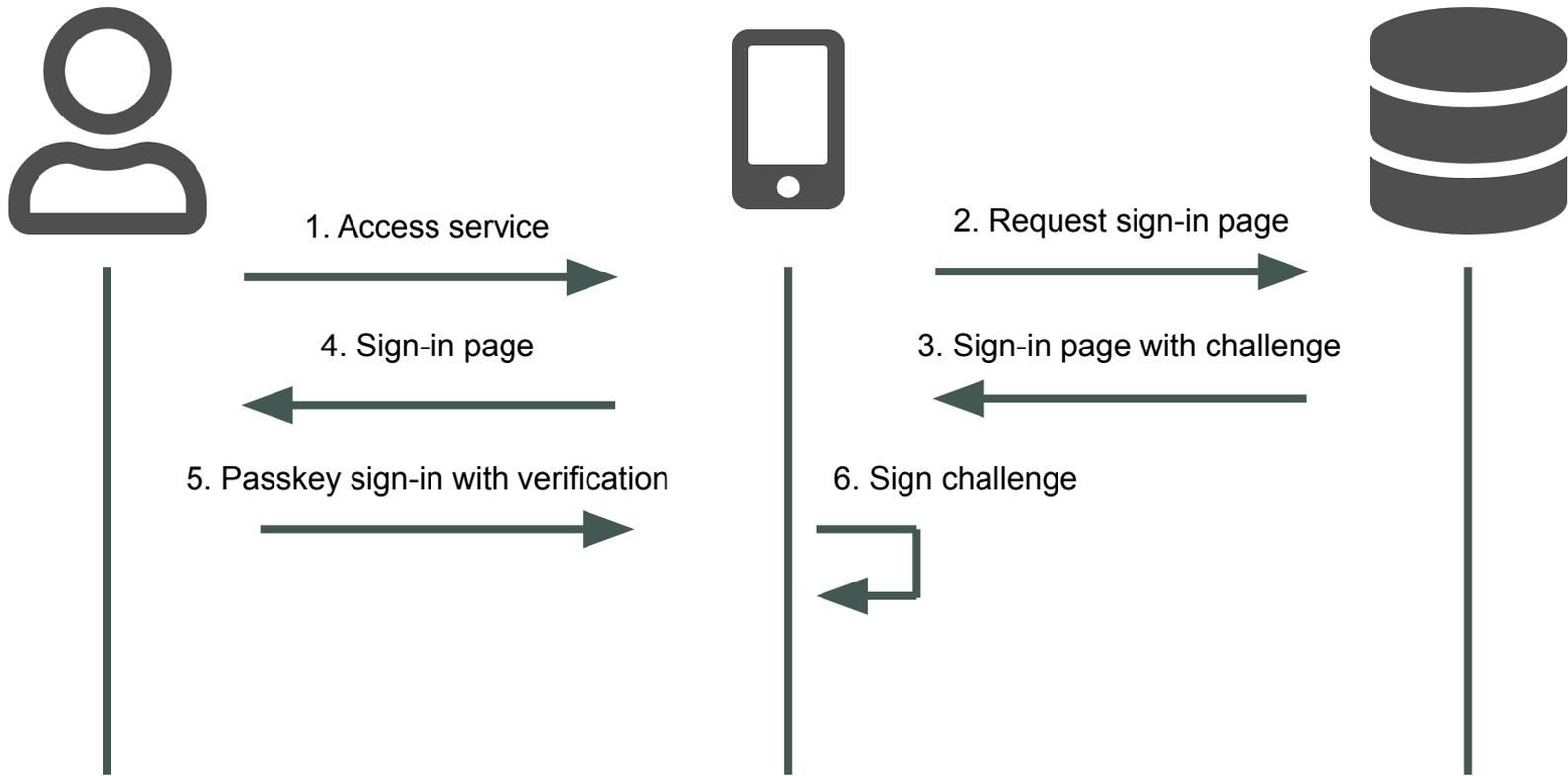
# Passkeys - Sign in with a passkey



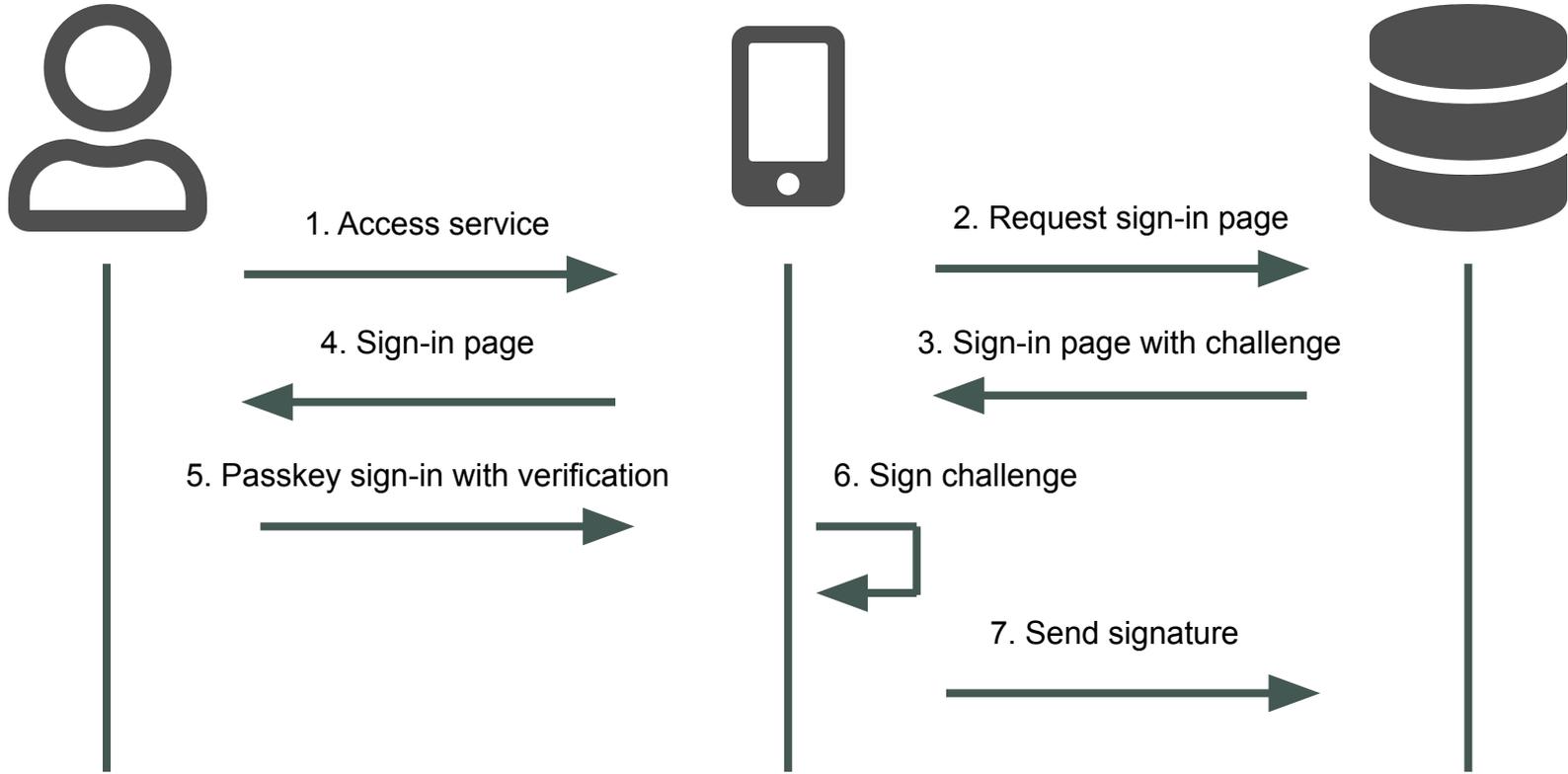
# Passkeys - Sign in with a passkey



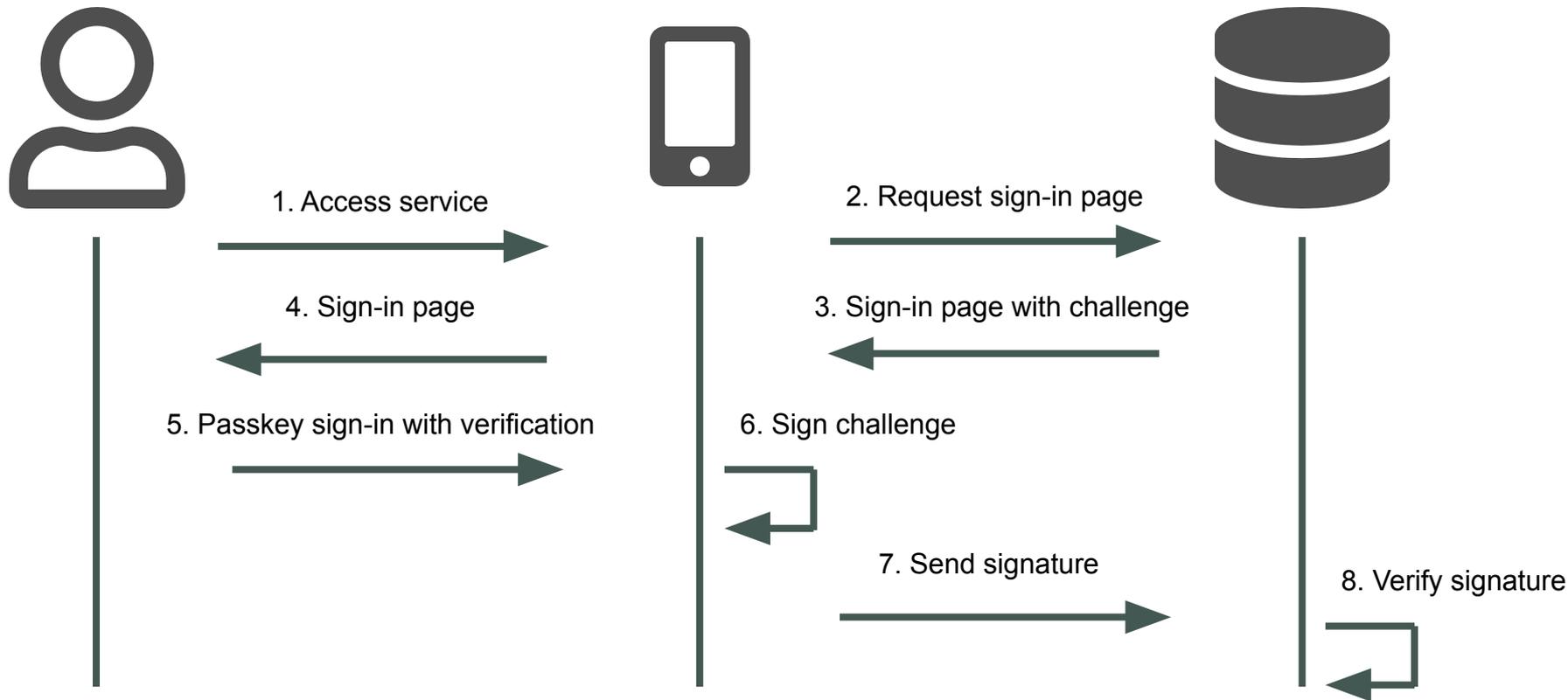
# Passkeys - Sign in with a passkey



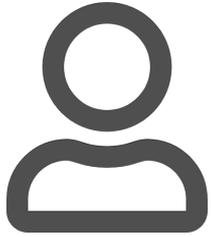
# Passkeys - Sign in with a passkey



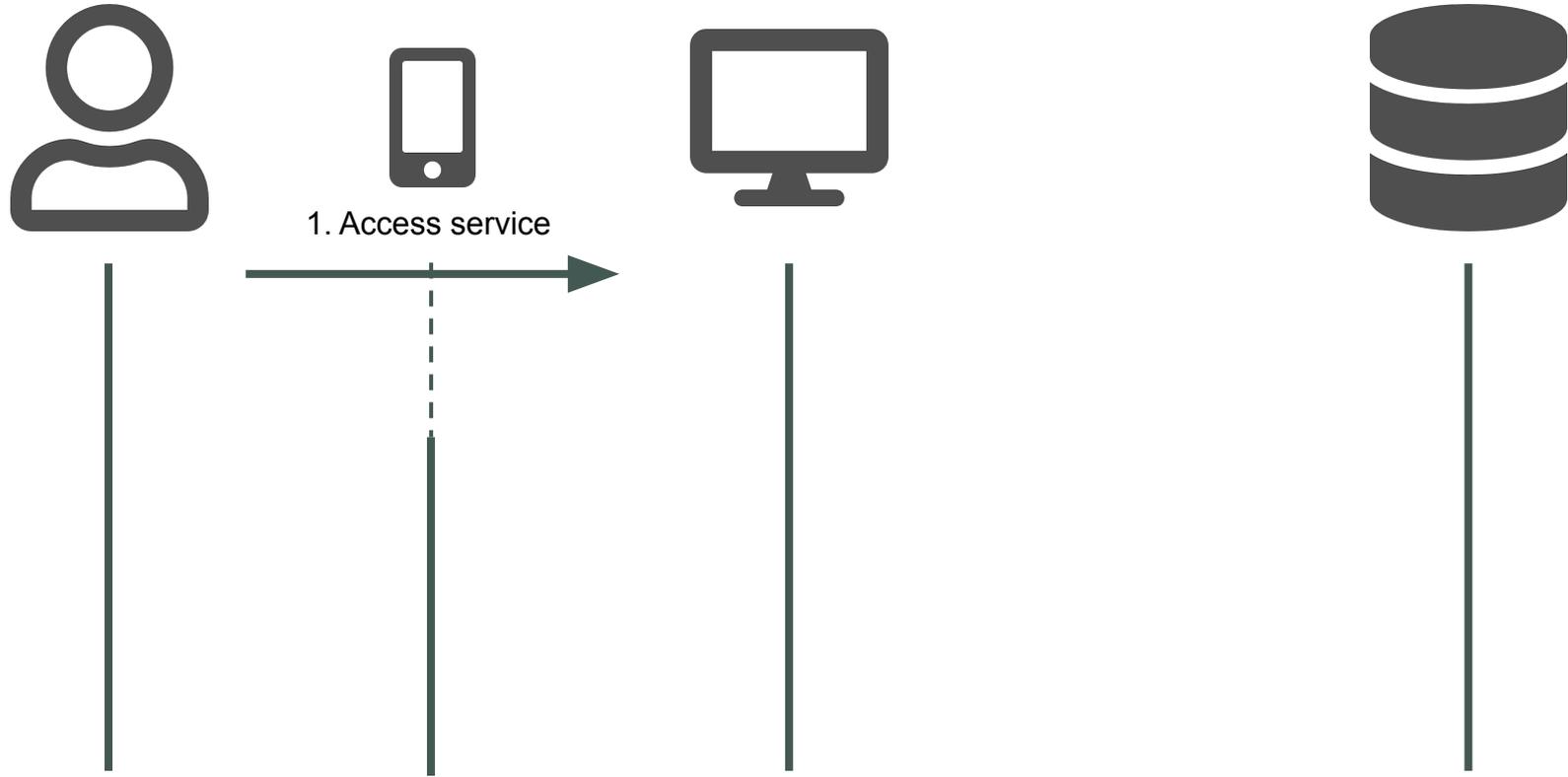
# Passkeys - Sign in with a passkey



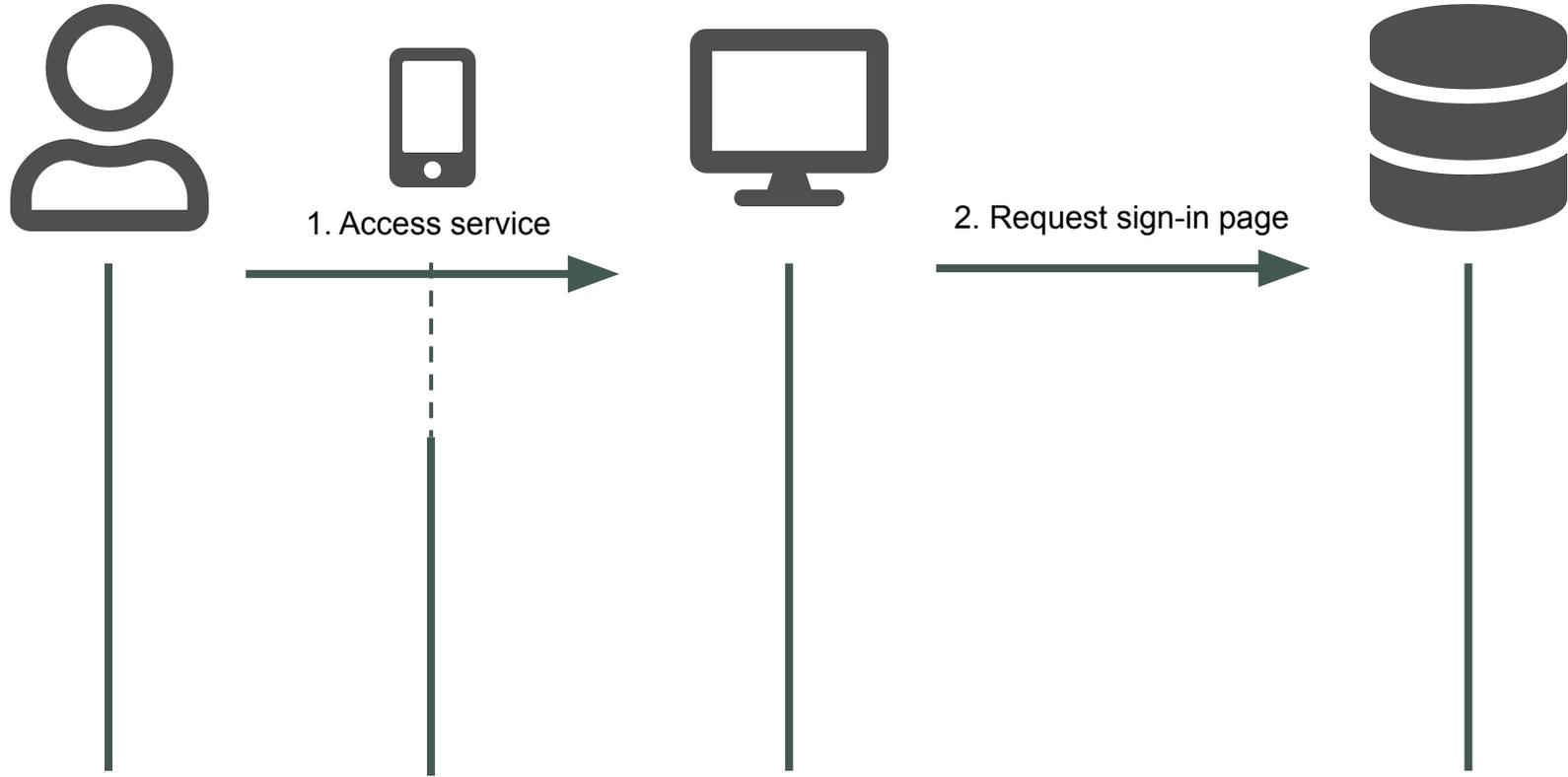
# Passkeys - Sign in with a passkey on other device



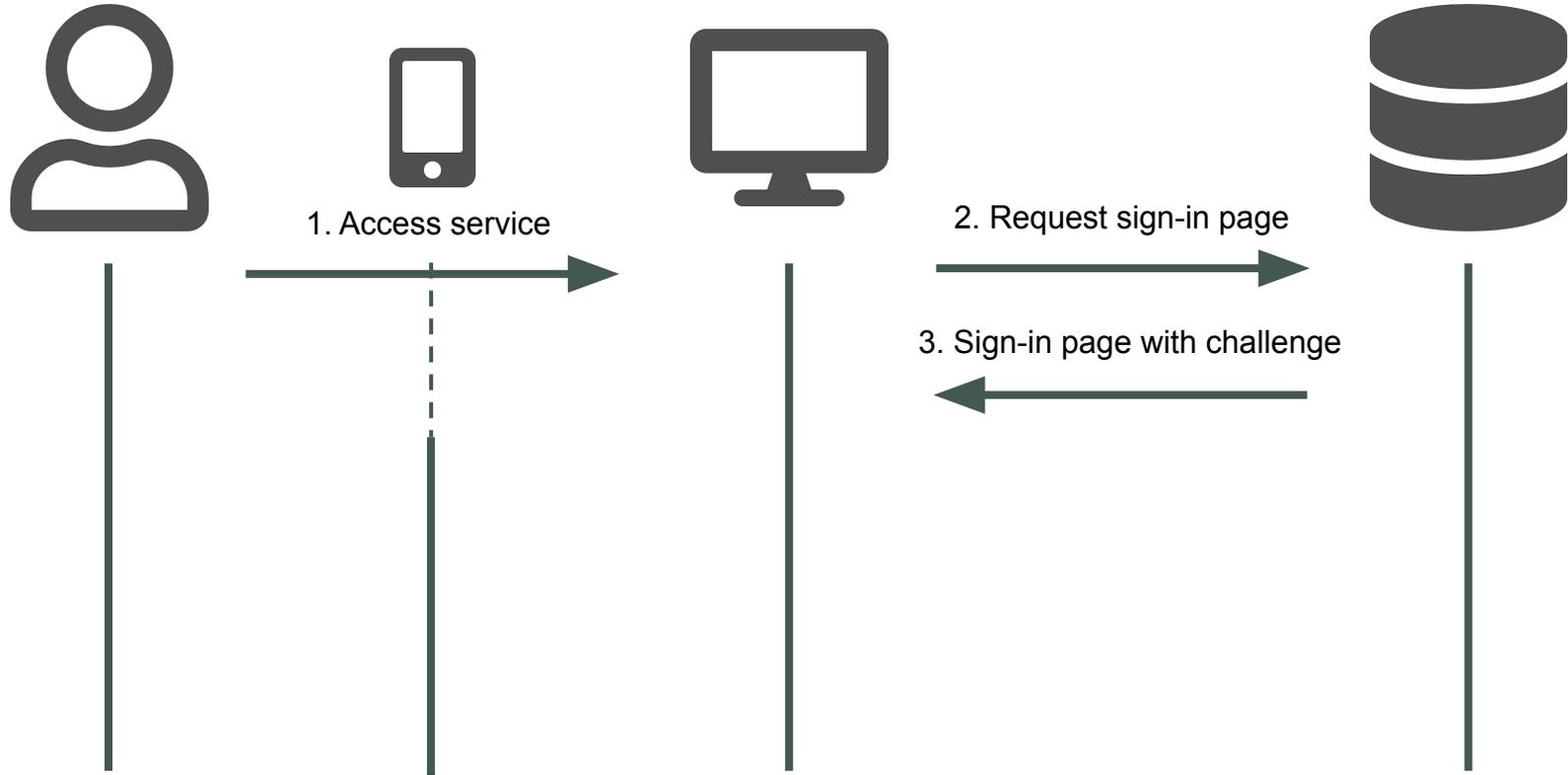
# Passkeys - Sign in with a passkey on other device



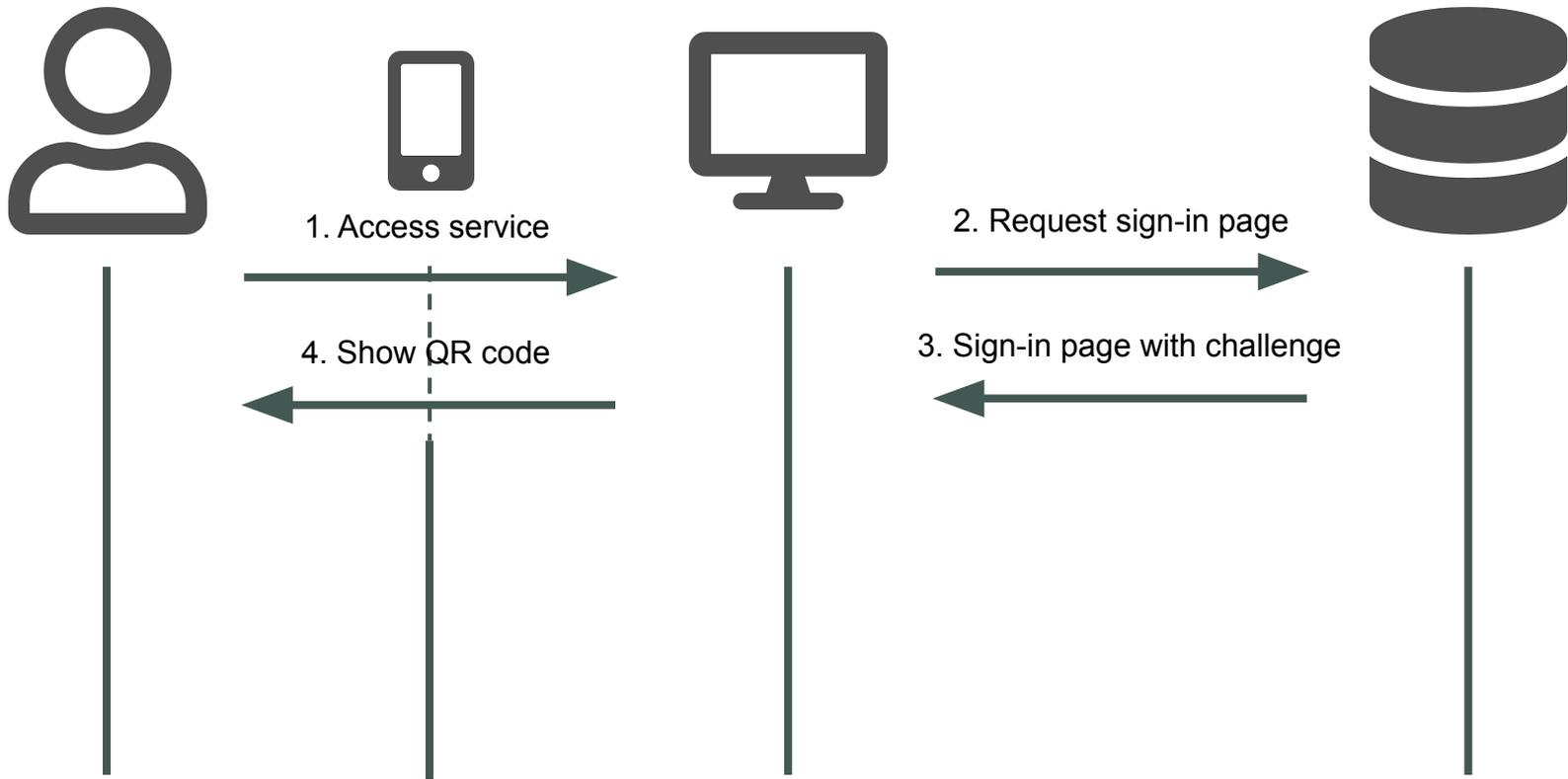
# Passkeys - Sign in with a passkey on other device



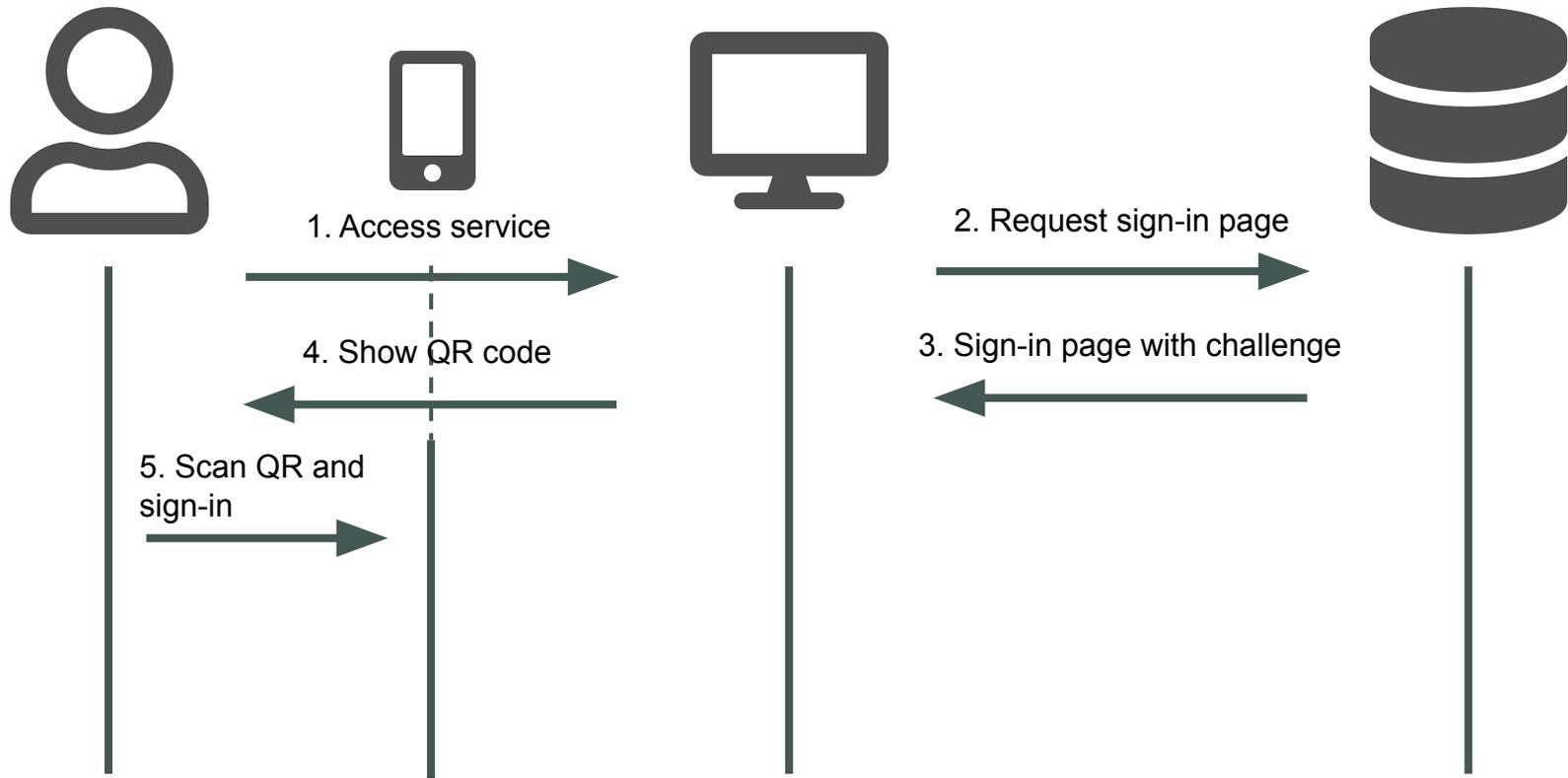
# Passkeys - Sign in with a passkey on other device



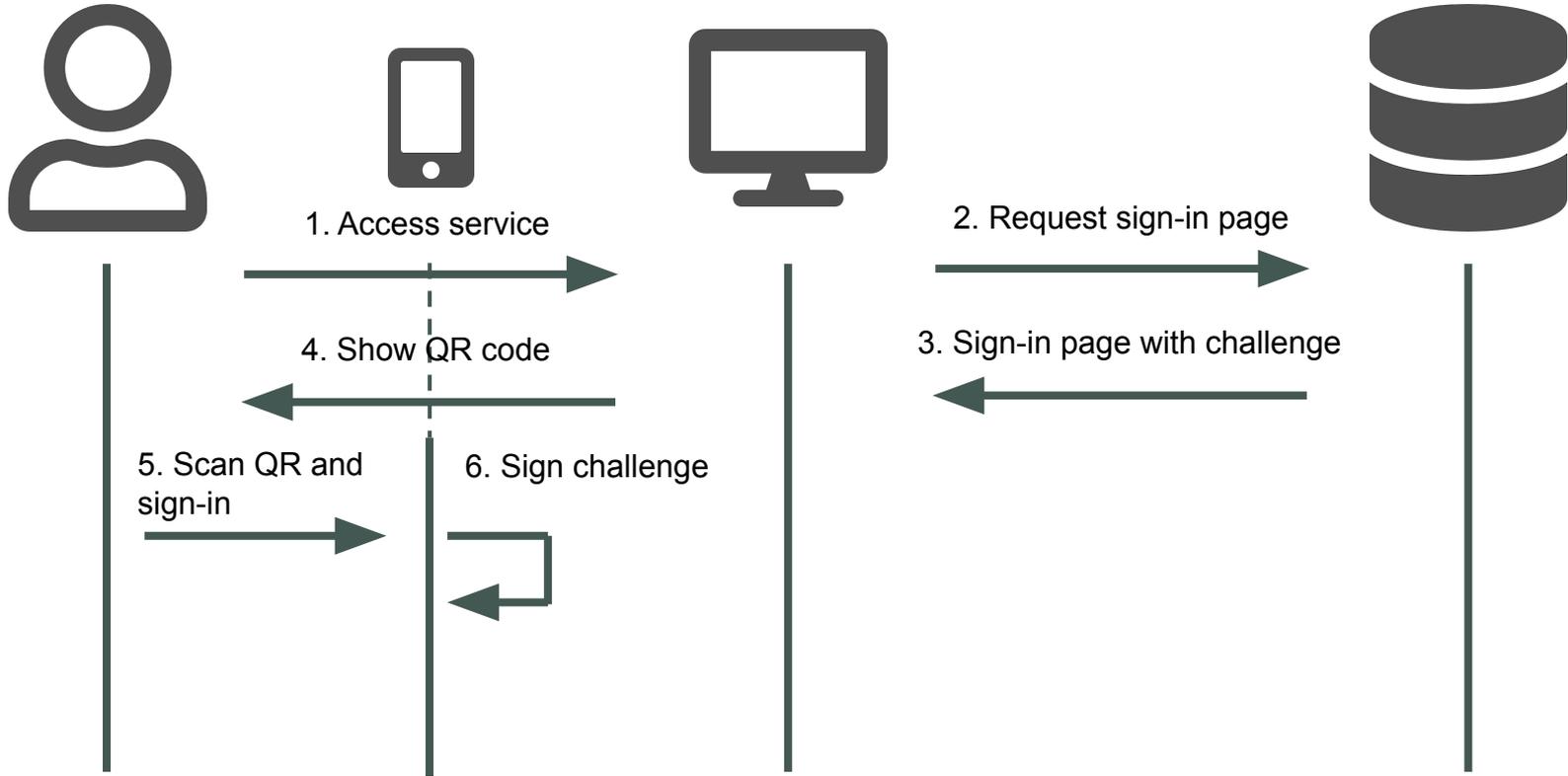
# Passkeys - Sign in with a passkey on other device



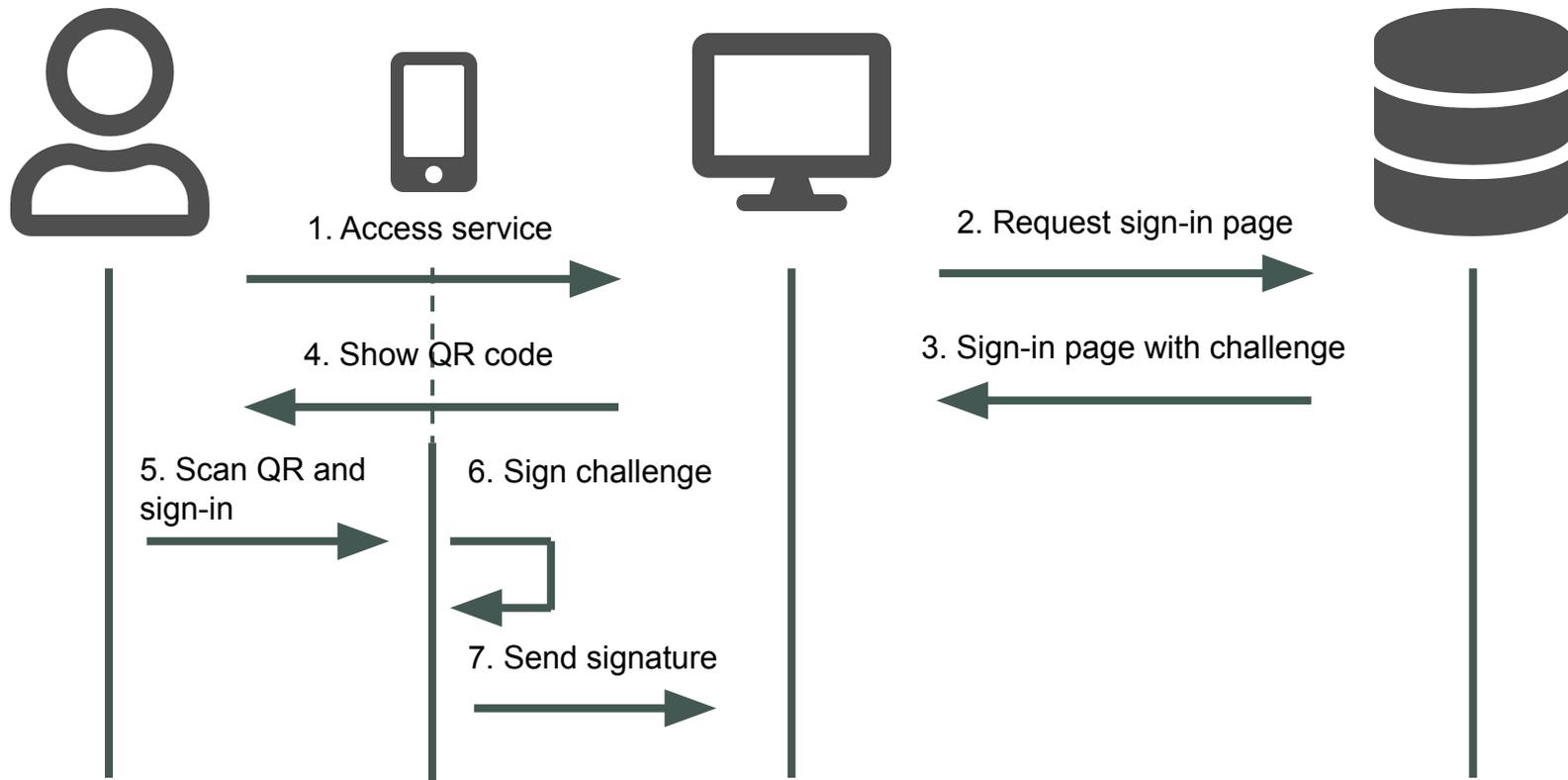
# Passkeys - Sign in with a passkey on other device



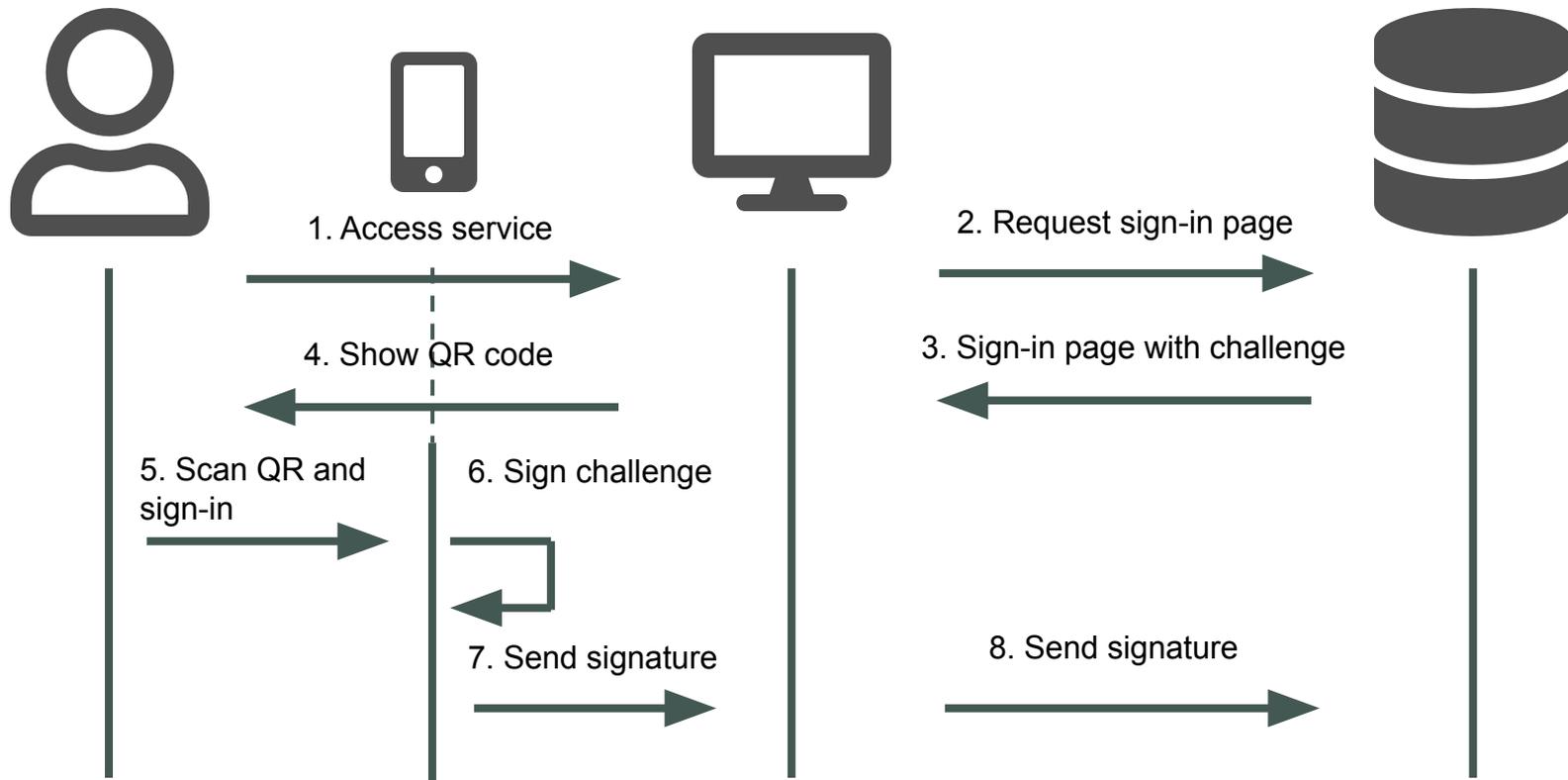
# Passkeys - Sign in with a passkey on other device



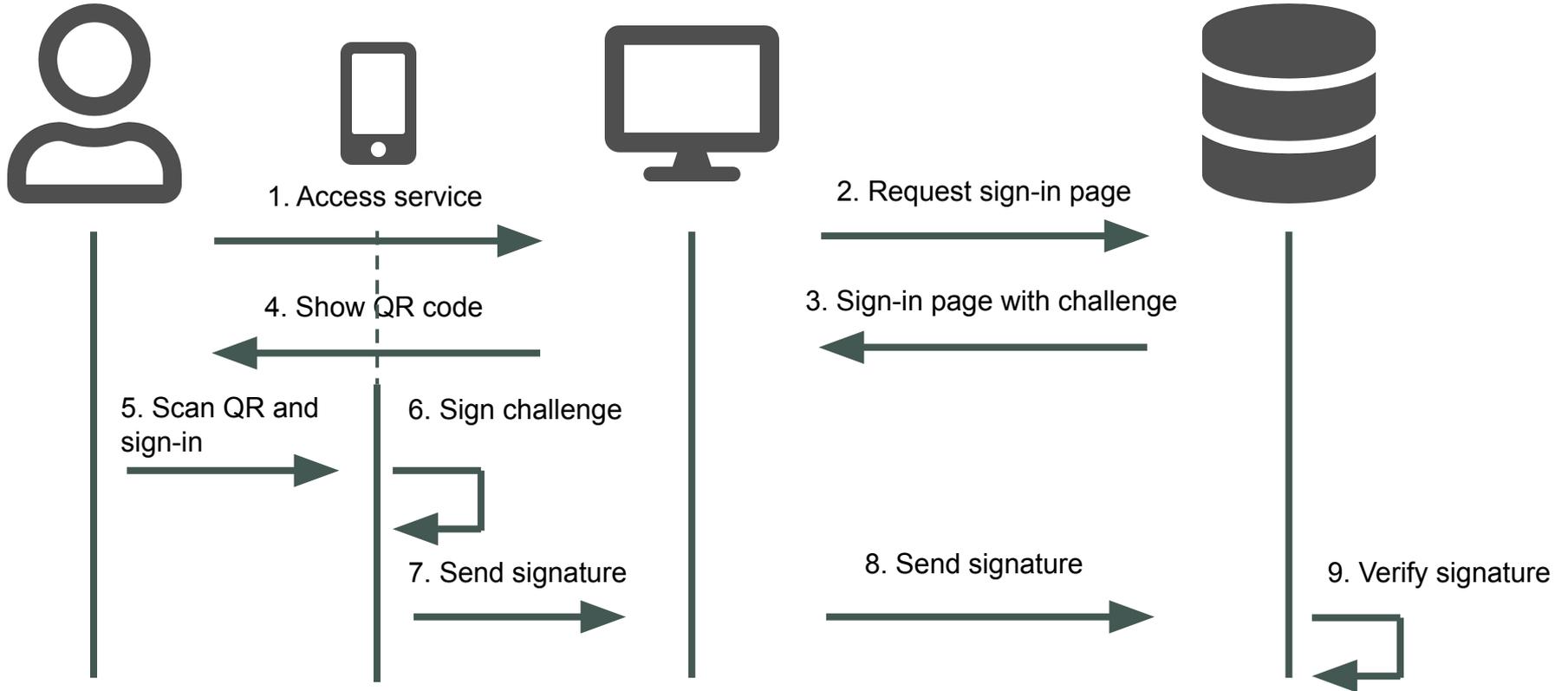
# Passkeys - Sign in with a passkey on other device



# Passkeys - Sign in with a passkey on other device



# Passkeys - Sign in with a passkey on other device



## Passkeys - Situation on the market

- Many commercial vendors **support** passkeys
- SimpleSAMLphp **supports** passkeys
- Shibboleth does **not support** passkeys (in progress)
- Keycloak **supports** passkeys
- PrivacyIDEA does **not fully support** passkeys yet

## Passkeys - Our approach

- Aim to build it on top of our MFA solution
- Need to replace some underlying components first
- Need to solve the UX part before implementation

# Passkeys - How to do the transition?

- Change our authentication gateway
- Need to support both approaches
- Passkeys through a button or autofill
- Need to educate users

## Which multi-factor authentication solution is right for me?

We know that it can be difficult to understand all the options for multi-factor authentication. It can be challenging to decide which method, device or software to use.

However, this guide can help you with that. Come and give it a try!

Start



# Passkeys - How to do the recovery and onboarding?

- What if users lost their passkey?
- How to onboard new users?
- Password as a backup?

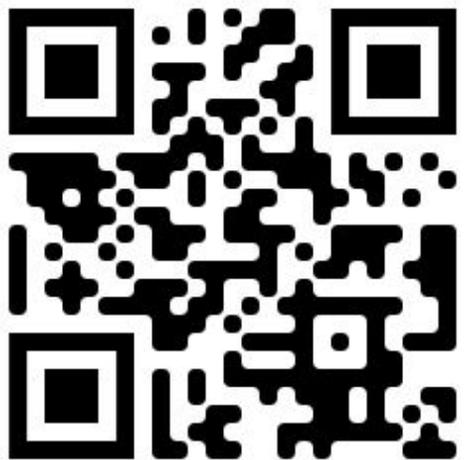
## Passkeys - What if passkeys are not enough?

- What if some community requires another factor?
- We need to know which passkey was used as a first factor
- Doable at one IdP
- Difficult when interconnecting separate AAls

# Passkeys - What if an IdP does not support passkeys?

- IdP does not tell us what factor was used
- Need to handled on the AAI side
- Need to navigate users seamlessly

**If not passwordless, at least use passwords less**



[peter.balcirak@cesnet.cz](mailto:peter.balcirak@cesnet.cz)