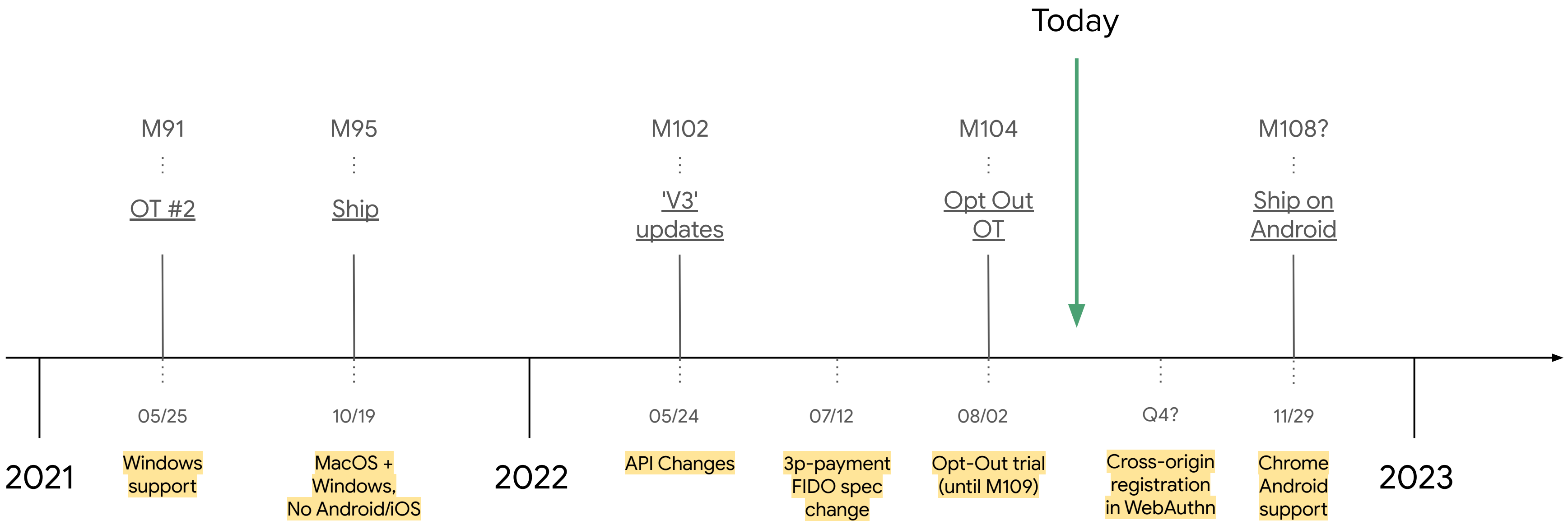
WPWG @ TPAC
SPC Implementation Update (Chrome)

smcgruer@google.com

2022/09/02

Timeline



Today

2021

M91
OT #2
05/25
Windows support

M95
Ship
10/19
MacOS + Windows, No Android/iOS

2022

M102
'V3' updates
05/24
API Changes

07/12
3p-payment FIDO spec change

M104
Opt Out OT
08/02
Opt-Out trial (until M109)

Q4?
Cross-origin registration in WebAuthn

M108?
Ship on Android
11/29
Chrome Android support

2023

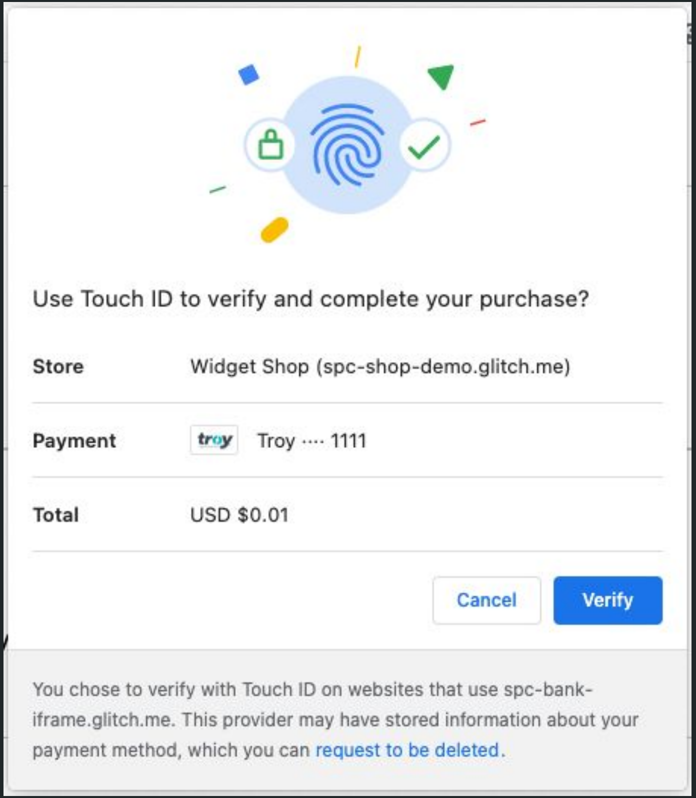
Recent Changes

- User activation requirement for both registration* and authentication
  - Registration was a change to the SPC spec, authentication is from finally fixing a bug!
  - PaymentRequest spec has required UA for show() 'forever' (since early 2018).
  - Both landed in M102 (released May 2022).
- 'thirdPartyPayment' extension landed in FIDO CTAP2 spec
  - No immediate impact, but hoping that it is a base for platform APIs to follow.
  - Still work needed to figure out the story for remote authenticators.
- Renamed rp → rpId in CollectedClientAdditionalPaymentData
  - Only affects the **output** cryptogram and validation of it (e.g., by the Relying Party server-side).
  - Breaking change, so deprecation plan:
    - Chrome M107 → has **both** in the cryptogram (please switch to using rpId if available)
    - Chrome M110 → remove 'rp' entirely (3 month period - too short?)

* When registering in a cross-origin iframe

Recent Changes - Opt Out

- In origin trial M104 ➔ M109 (inclusive).
- Shown on both the transaction dialog and no matching credentials dialog, to preserve user privacy.
- When user clicks the 'link':
  - show() promise rejected with AbortError.
  - Caller handles it and presents an Opt-Out flow to the user.
  - Caller **cannot** immediately open a pop-up - no user activation!
- Possible change: return a successful 'opt-out' return value rather than AbortError.
  - Easier to detect, less ambiguous.



Use Touch ID to verify and complete your purchase?

Store	Widget Shop (spc-shop-demo.glitch.me)
Payment	troy Troy ···· 1111
Total	USD $0.01

You chose to verify with Touch ID on websites that use spc-bank-iframe.glitch.me. This provider may have stored information about your payment method, which you can request to be deleted.

Upcoming Changes - WebAuthn/FIDO integration

- Currently: SPC 'overrides' WebAuthn to allow credential creation in a cross-origin iframe (when payment extension specified).

- Plan: move this logic into WebAuthn proper.
    - Will require both a permission policy **and** user activation, as anti-tracking mitigations.

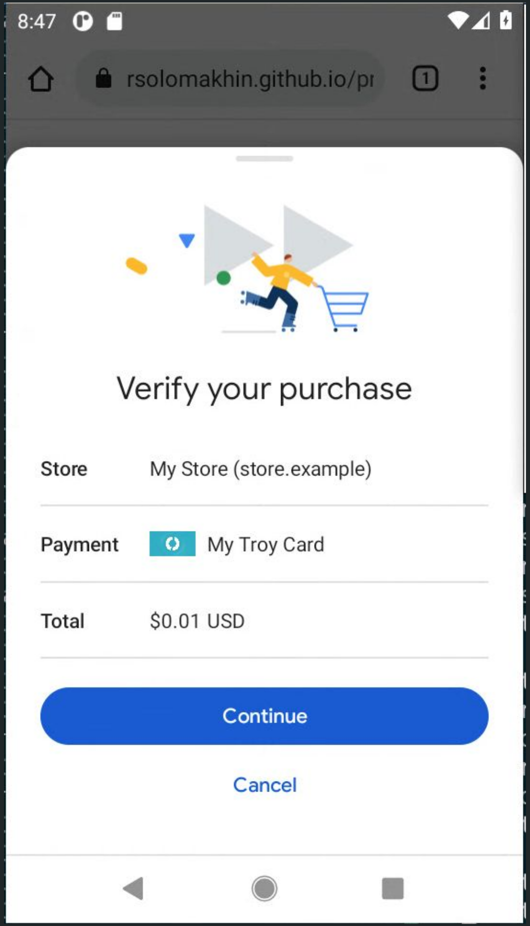
- Status: reopened discussion in WebAuthn issue, no response yet.
    - To be discussed at WPWG/WebAuthn meeting?

Upcoming Changes - SPC on Chrome Android

- Currently: In development.
  - Updated with new SPC features (payeeName, iconMustBeShown)
  - Changed spec + impl to allow residentKey = 'preferred'

- Plan: Finish up development work and ship it!
  - Add (experimental) opt-out support
  - Integrate with credential store APIs...

- Status: Available behind developer flag as of M107 (in Canary now).
  - M108 launch? **TBD!**

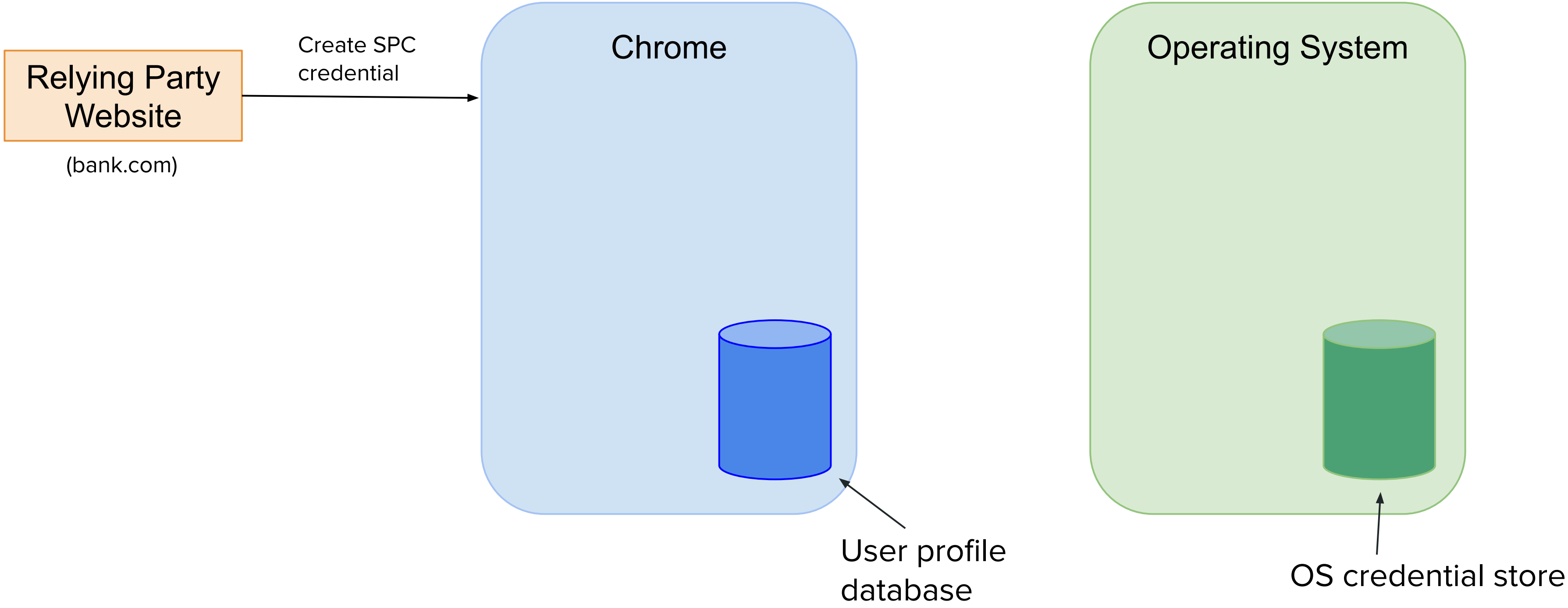


Verify your purchase

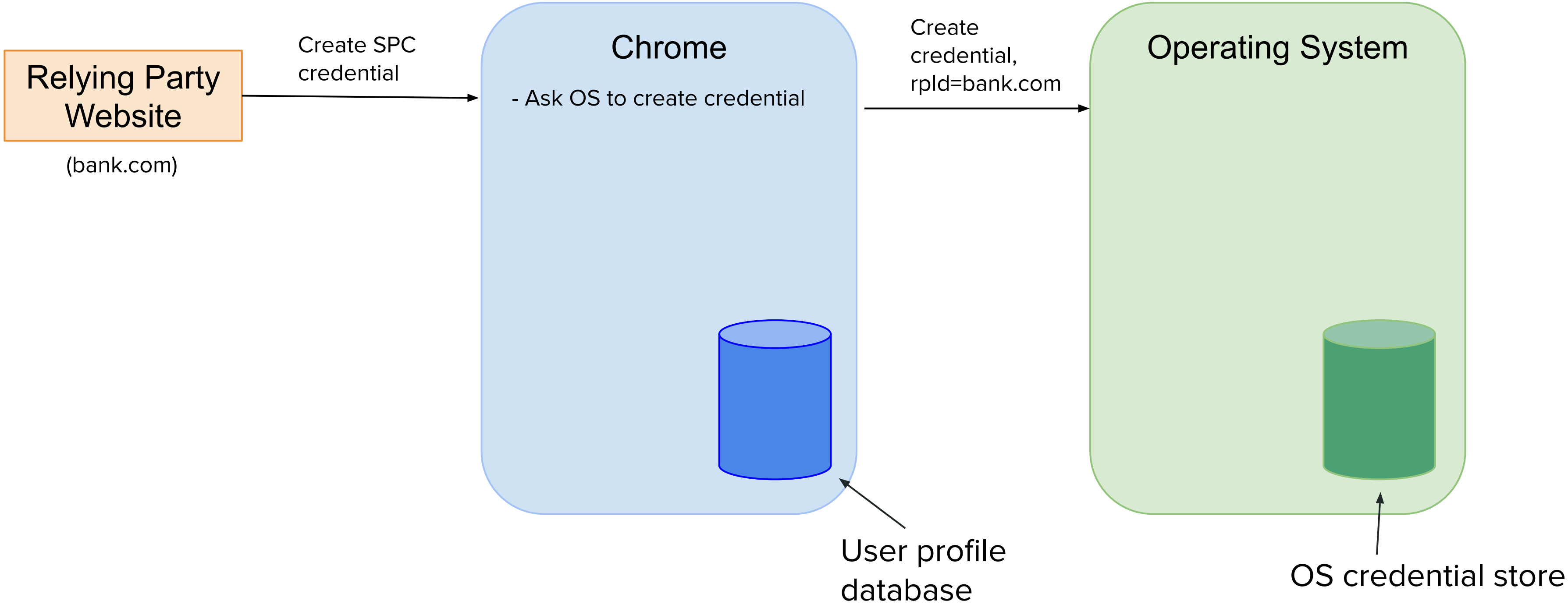
Store	My Store (store.example)
Payment	My Troy Card
Total	$0.01 USD

SPC Android - what took so long?

SPC - the 'user profile hack' (creation)



SPC - the 'user profile hack' (creation)



Relying Party Website

(bank.com)

Create SPC credential

Chrome

- Ask OS to create credential

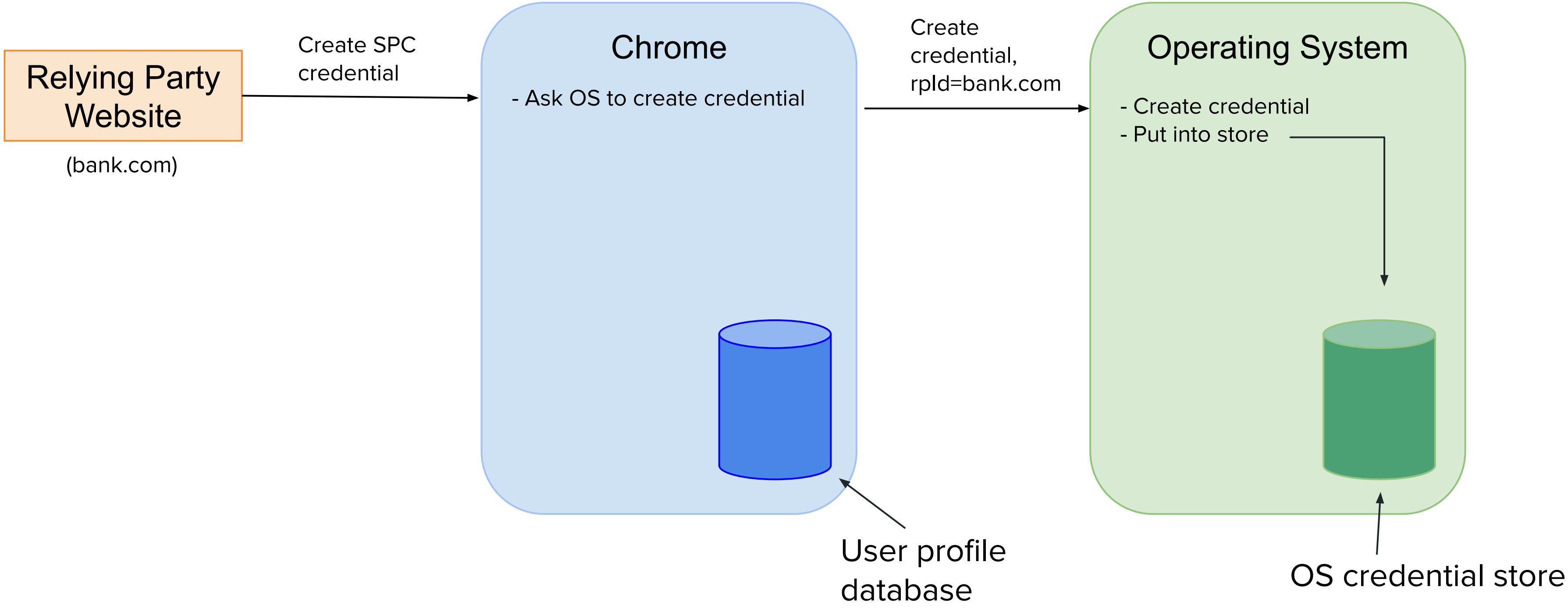
Create credential, rpId=bank.com

Operating System

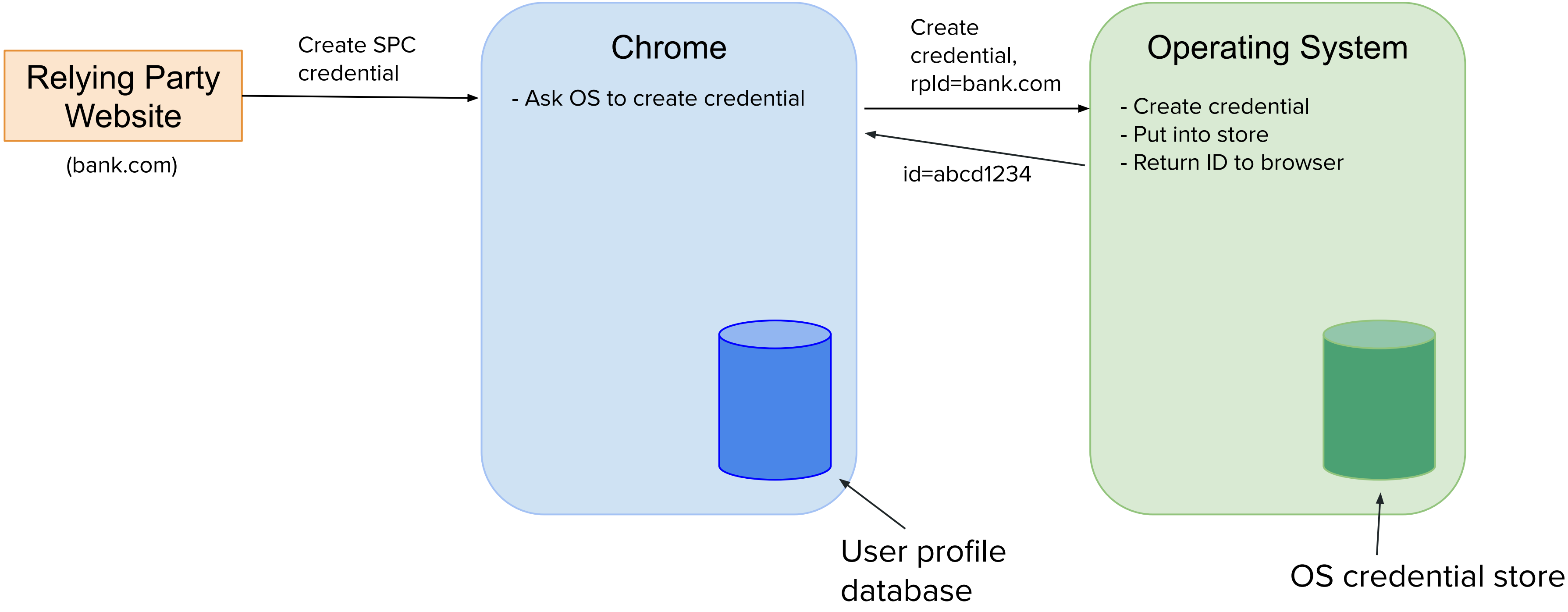
User profile database

OS credential store

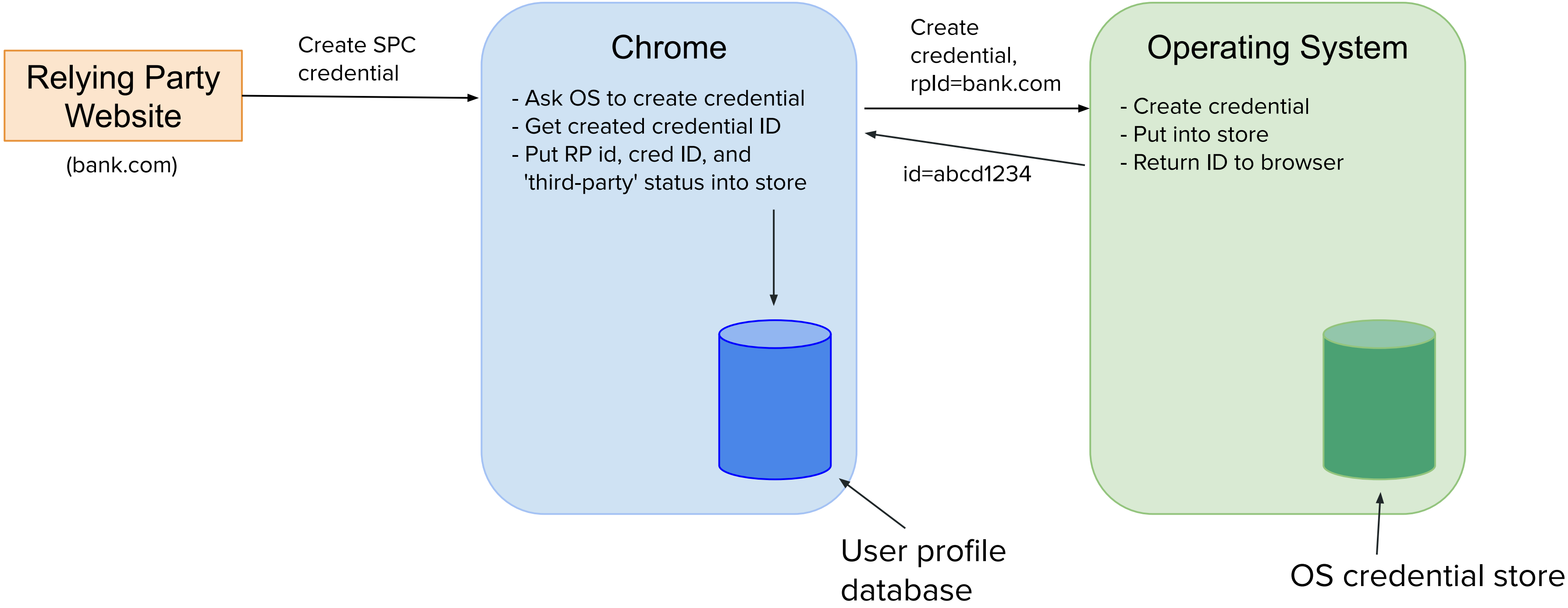
SPC - the 'user profile hack' (creation)



SPC - the 'user profile hack' (creation)



SPC - the 'user profile hack' (creation)



Relying Party Website

(bank.com)

Create SPC credential

Chrome

- Ask OS to create credential
- Get created credential ID
- Put RP id, cred ID, and 'third-party' status into store

Create credential, rpId=bank.com

id=abcd1234

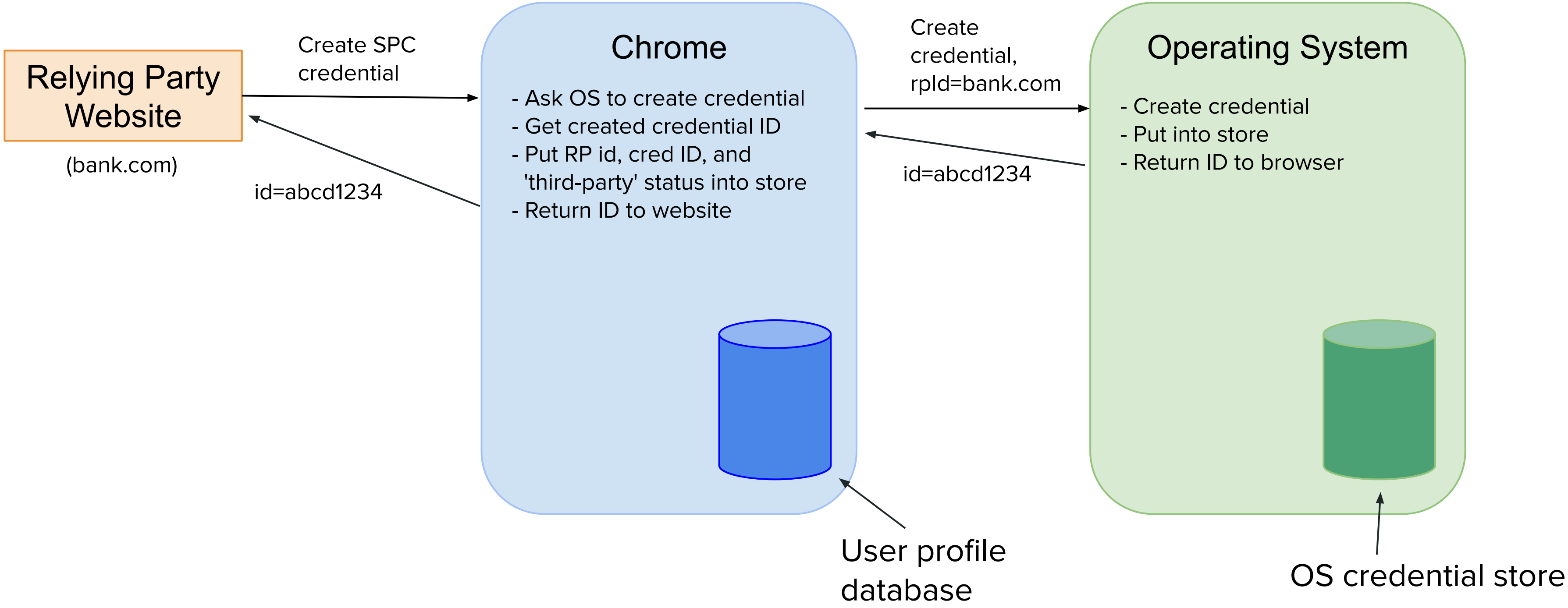
Operating System

- Create credential
- Put into store
- Return ID to browser

User profile database

OS credential store

SPC - the 'user profile hack' (creation)



Relying Party Website

(bank.com)

Create SPC credential

id=abcd1234

Chrome

- Ask OS to create credential
- Get created credential ID
- Put RP id, cred ID, and 'third-party' status into store
- Return ID to website

Create credential, rpId=bank.com

id=abcd1234

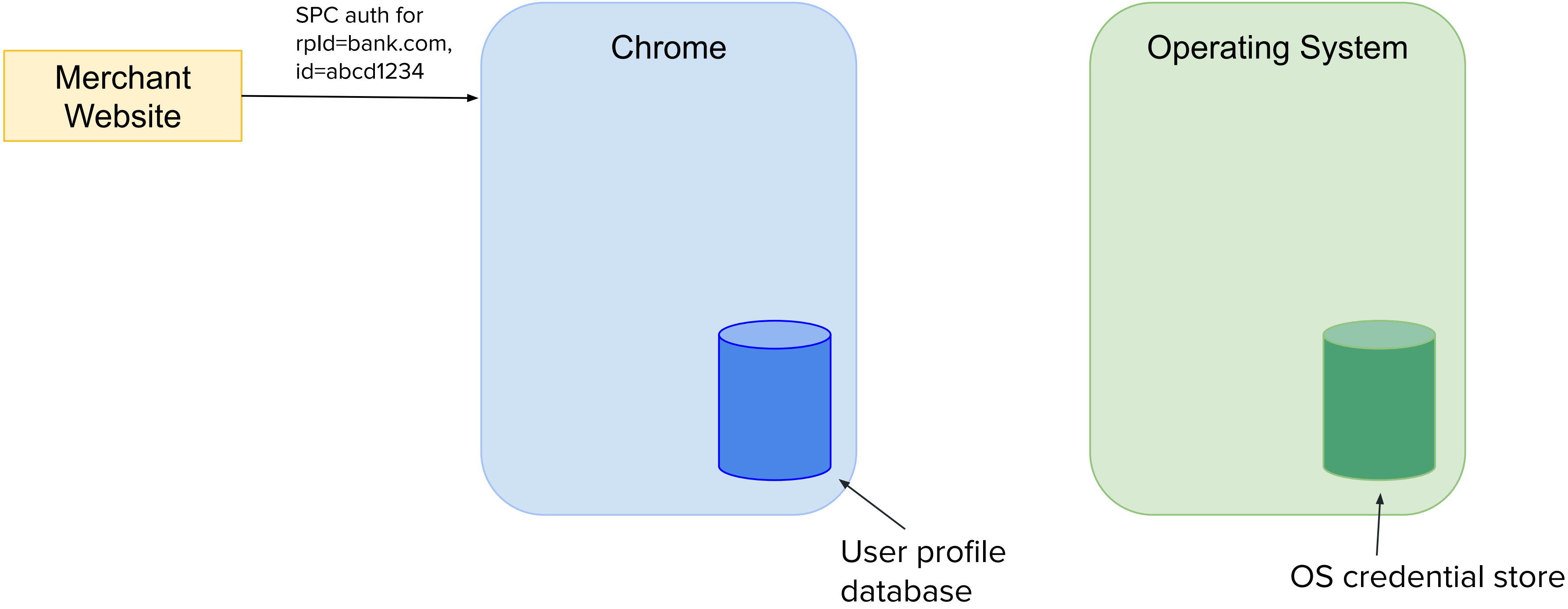
Operating System

- Create credential
- Put into store
- Return ID to browser

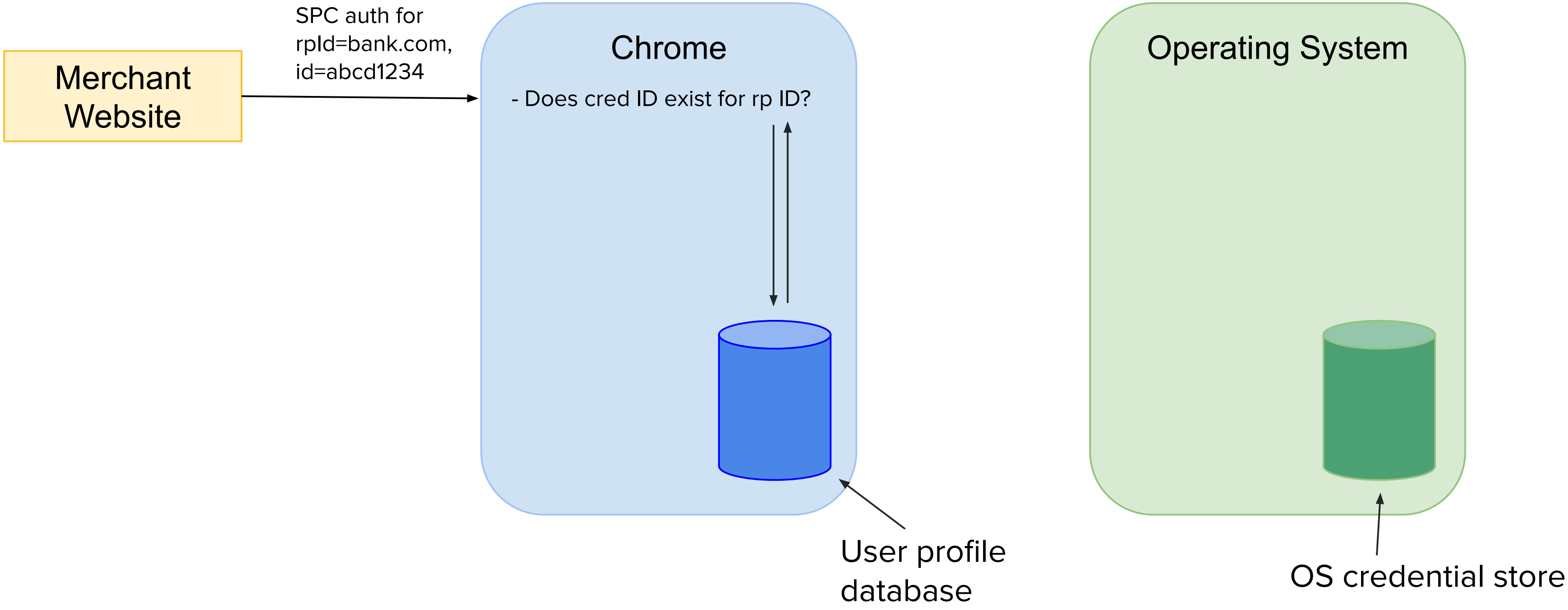
User profile database

OS credential store

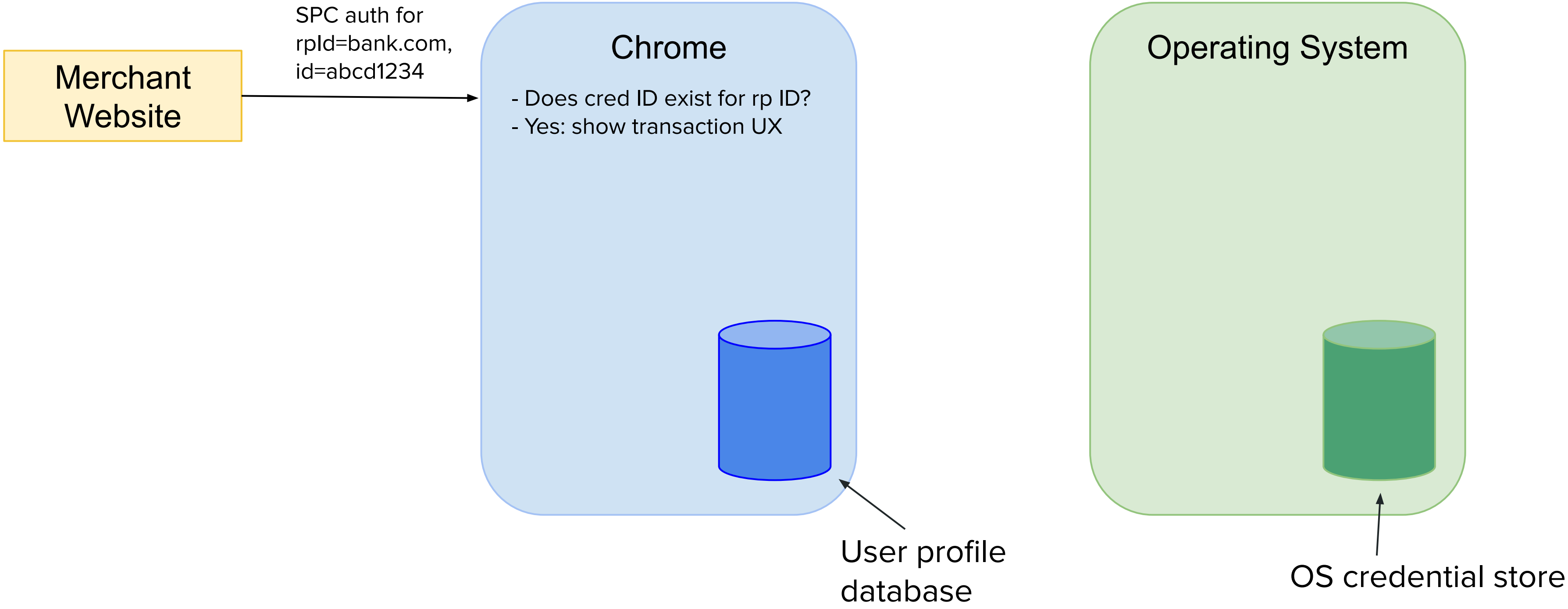
SPC - the 'user profile hack' (authentication)



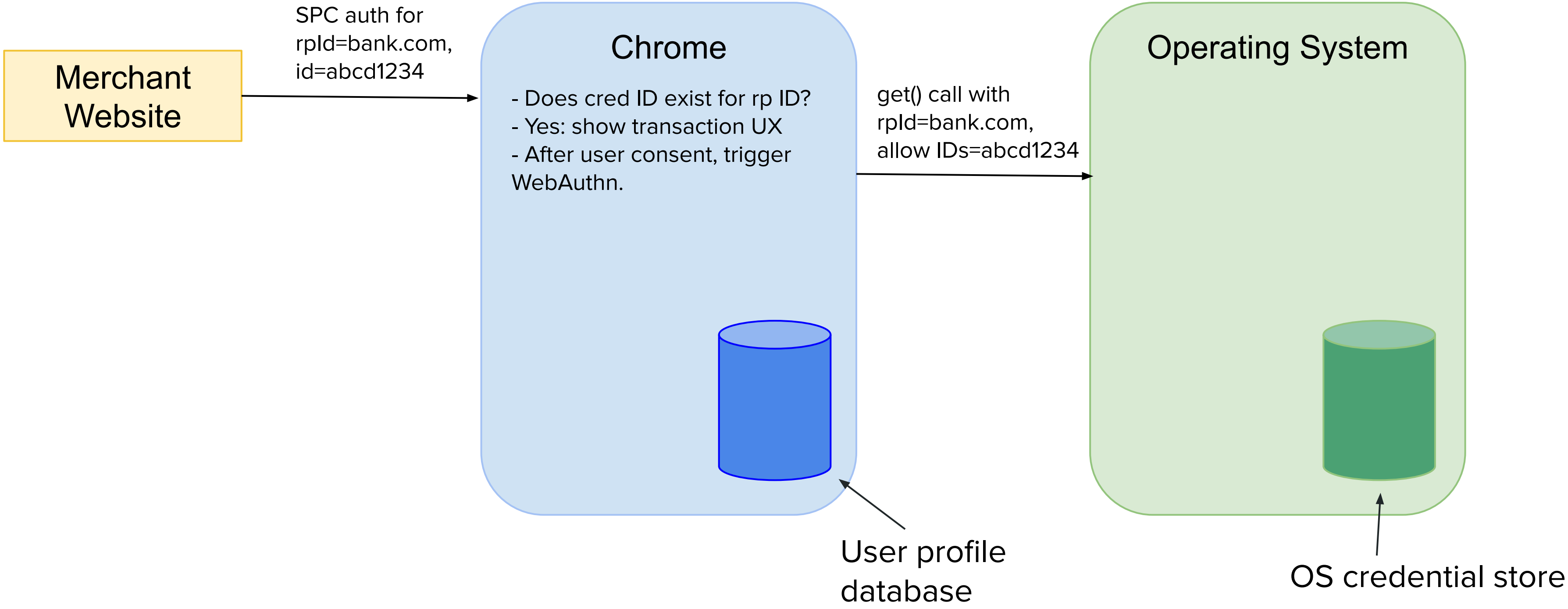
SPC - the 'user profile hack' (authentication)



SPC - the 'user profile hack' (authentication)



SPC - the 'user profile hack' (authentication)



Merchant Website

SPC auth for rpId=bank.com, id=abcd1234

Chrome

- Does cred ID exist for rp ID?
- Yes: show transaction UX
- After user consent, trigger WebAuthn.

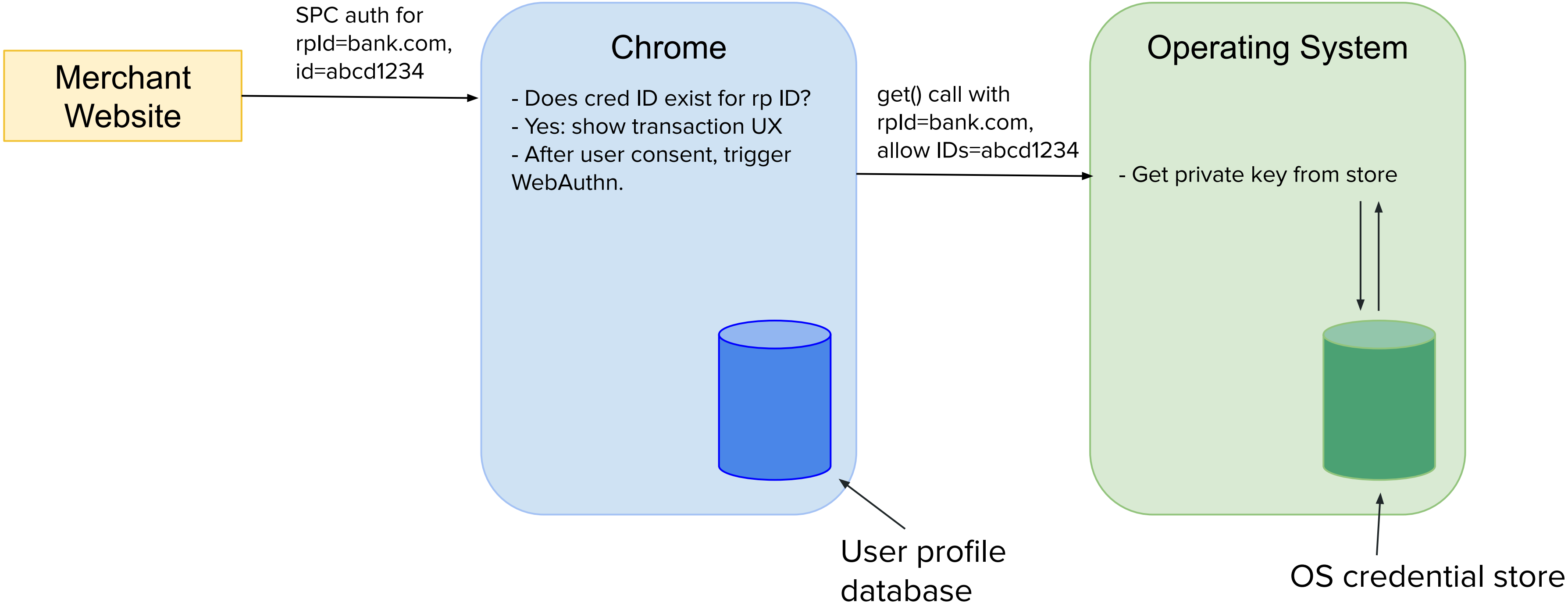
get() call with rpId=bank.com, allow IDs=abcd1234

Operating System

User profile database

OS credential store

SPC - the 'user profile hack' (authentication)



Merchant Website

SPC auth for rpId=bank.com, id=abcd1234

Chrome

- Does cred ID exist for rp ID?
- Yes: show transaction UX
- After user consent, trigger WebAuthn.

get() call with rpId=bank.com, allow IDs=abcd1234

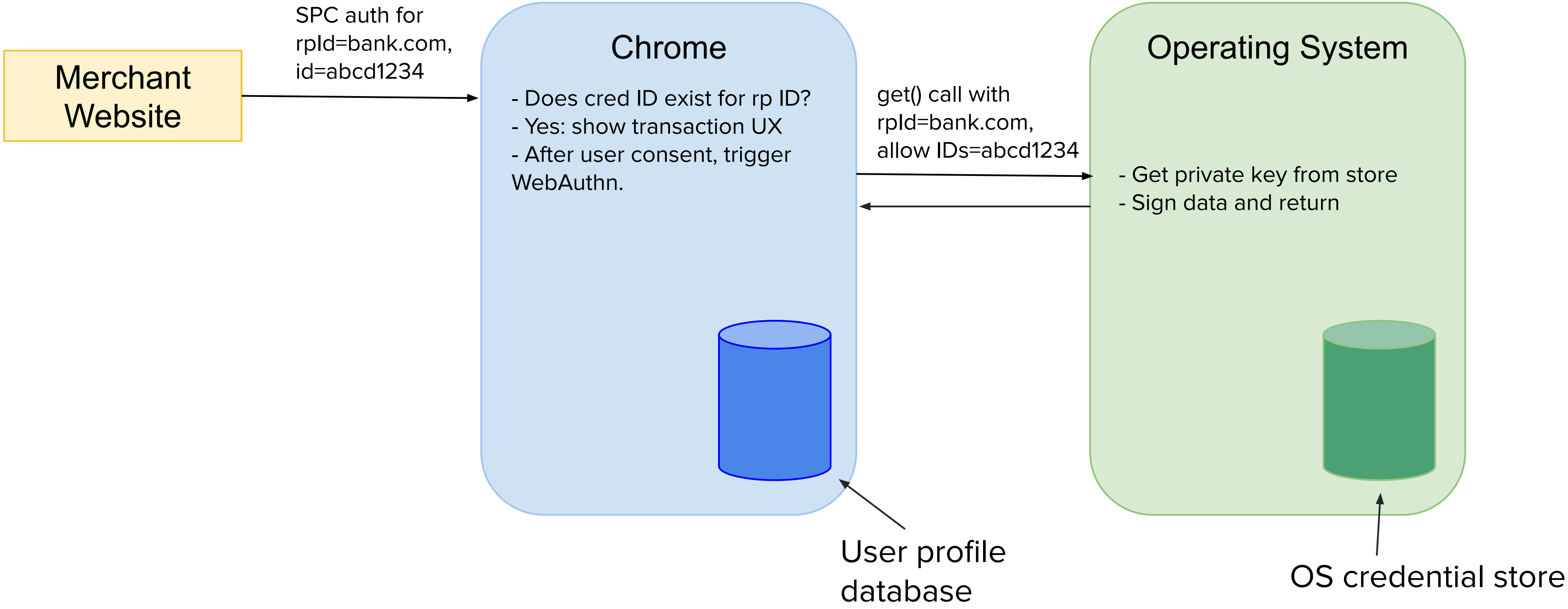
Operating System

- Get private key from store

User profile database

OS credential store

SPC - the 'user profile hack' (authentication)



Merchant Website

SPC auth for rpId=bank.com, id=abcd1234

Chrome

- Does cred ID exist for rp ID?
- Yes: show transaction UX
- After user consent, trigger WebAuthn.

get() call with rpId=bank.com, allow IDs=abcd1234

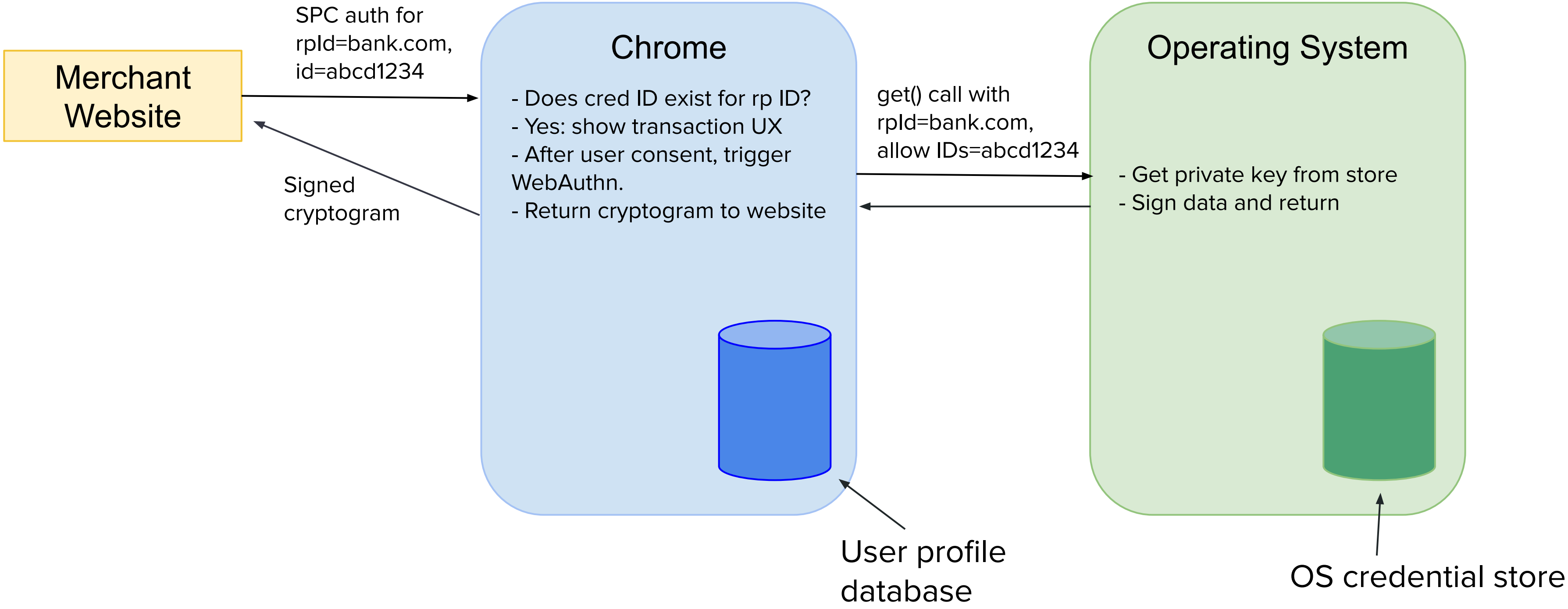
Operating System

- Get private key from store
- Sign data and return

User profile database

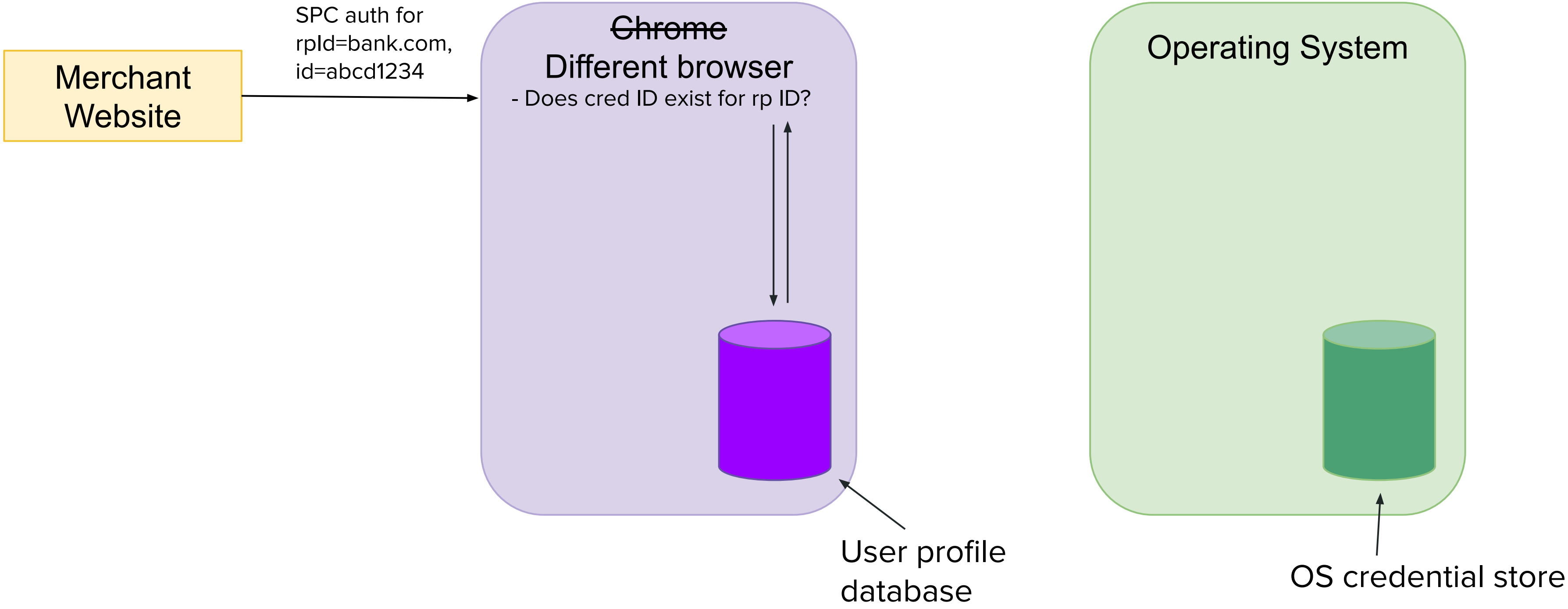
OS credential store

SPC - the 'user profile hack' (authentication)

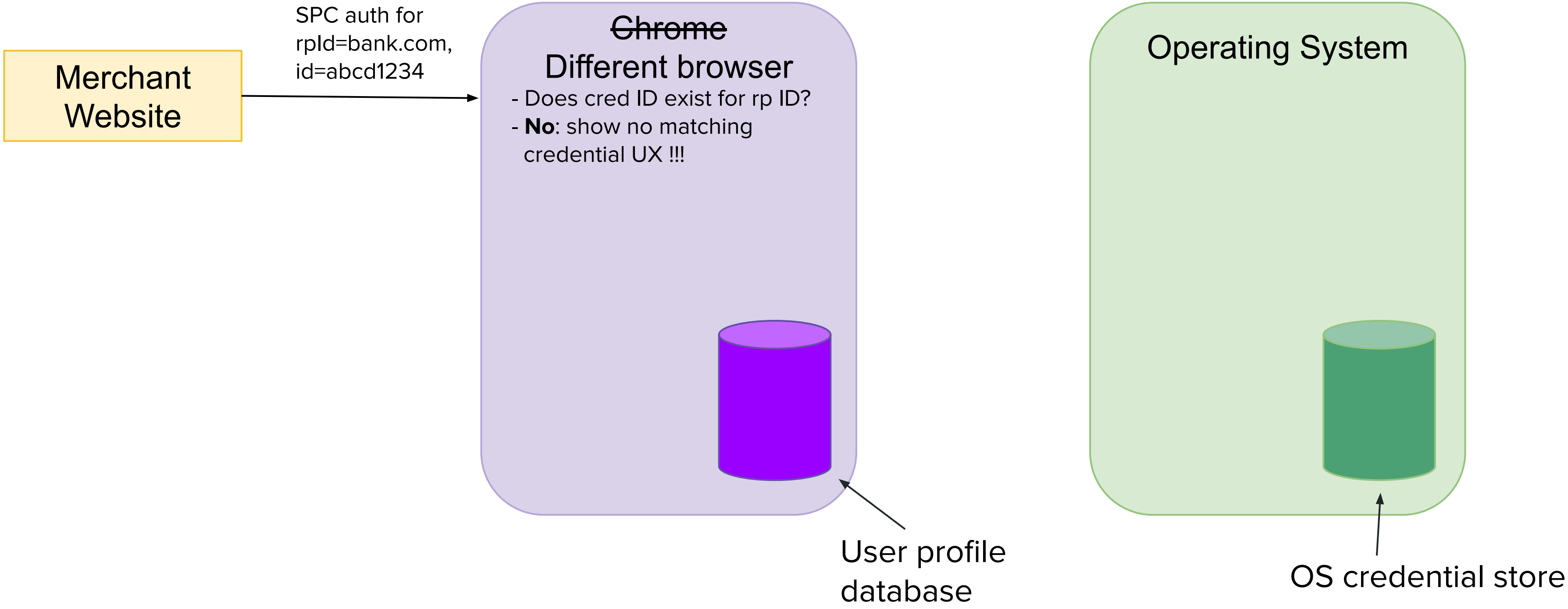


Merchant Website
SPC auth for rpId=bank.com, id=abcd1234
Signed cryptogram
Chrome
- Does cred ID exist for rp ID?
- Yes: show transaction UX
- After user consent, trigger WebAuthn.
- Return cryptogram to website
get() call with rpId=bank.com, allow IDs=abcd1234
Operating System
- Get private key from store
- Sign data and return
User profile database
OS credential store

SPC - the 'user profile hack' (authentication)



SPC - the 'user profile hack' (authentication)



Merchant Website

SPC auth for rpId=bank.com, id=abcd1234

~~Chrome~~
Different browser

- Does cred ID exist for rp ID?
- **No**: show no matching credential UX !!!

User profile database

Operating System

OS credential store

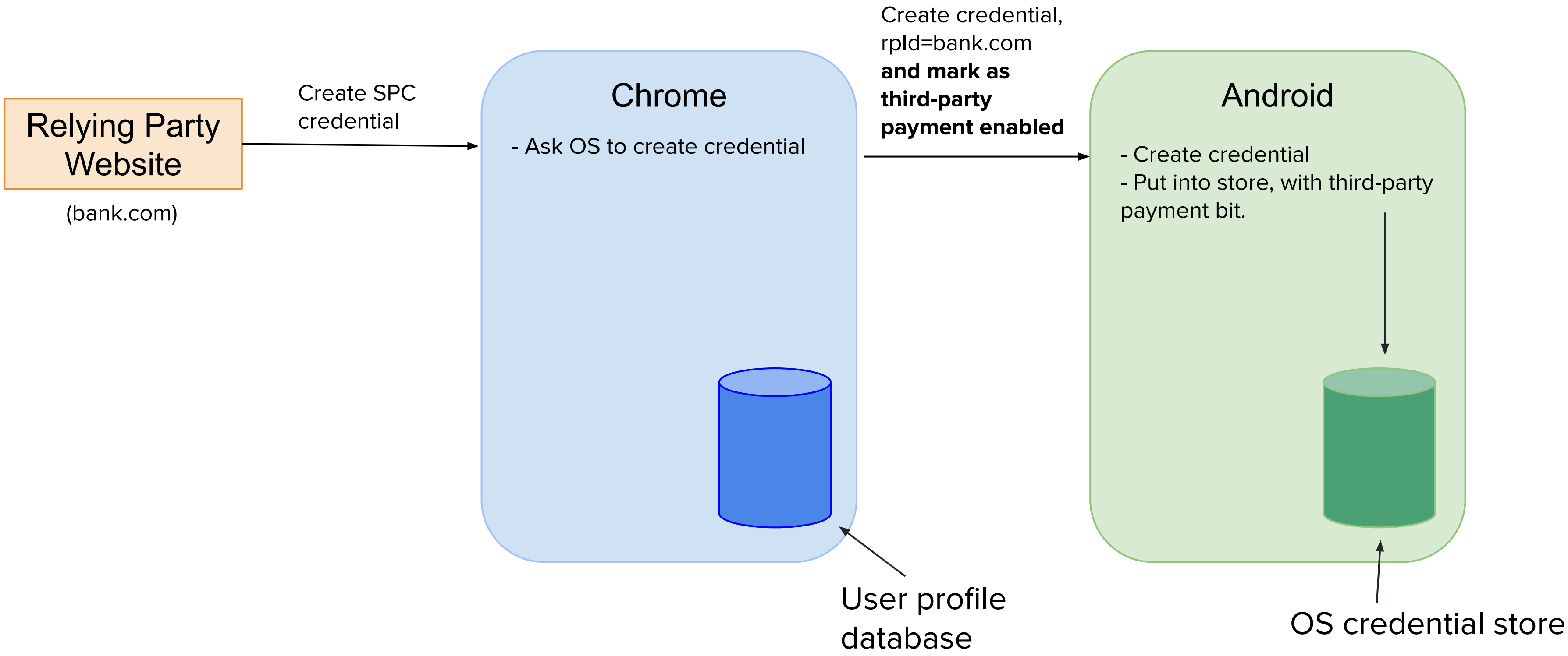
The 'user profile hack' - consequences

- False negatives when trying to use SPC across different browsers (or even user profiles on same browser) - the previous slides.

- Can also have false positive, where a different browser overrides an existing credential (i.e., same user ID for a discoverable credential).
  - User profile database won't see this update - so it thinks the old credential is still valid for SPC!
  - Browser will show the transaction UX, but the WebAuthn attempt after that will fail!!

- Doesn't support the 'first-party payment' use-case for SPC (e.g., RP using SPC on its own origin with a 'vanilla' WebAuthn credential).
  - The user profile database only records SPC credentials (on Chrome, at least).

SPC Android - doing it right



Relying Party Website

(bank.com)

Create SPC credential

Chrome

- Ask OS to create credential

Create credential, rpId=bank.com **and mark as third-party payment enabled**

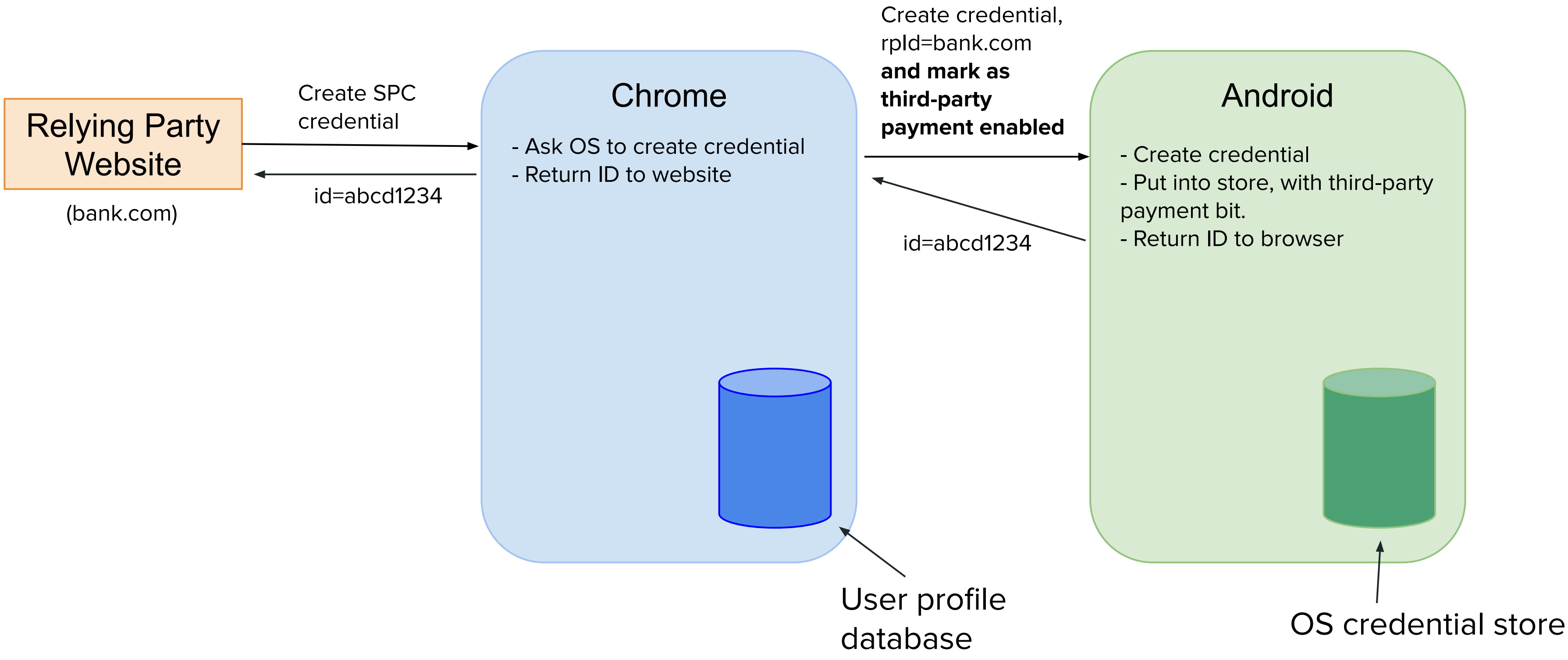
Android

- Create credential
- Put into store, with third-party payment bit.

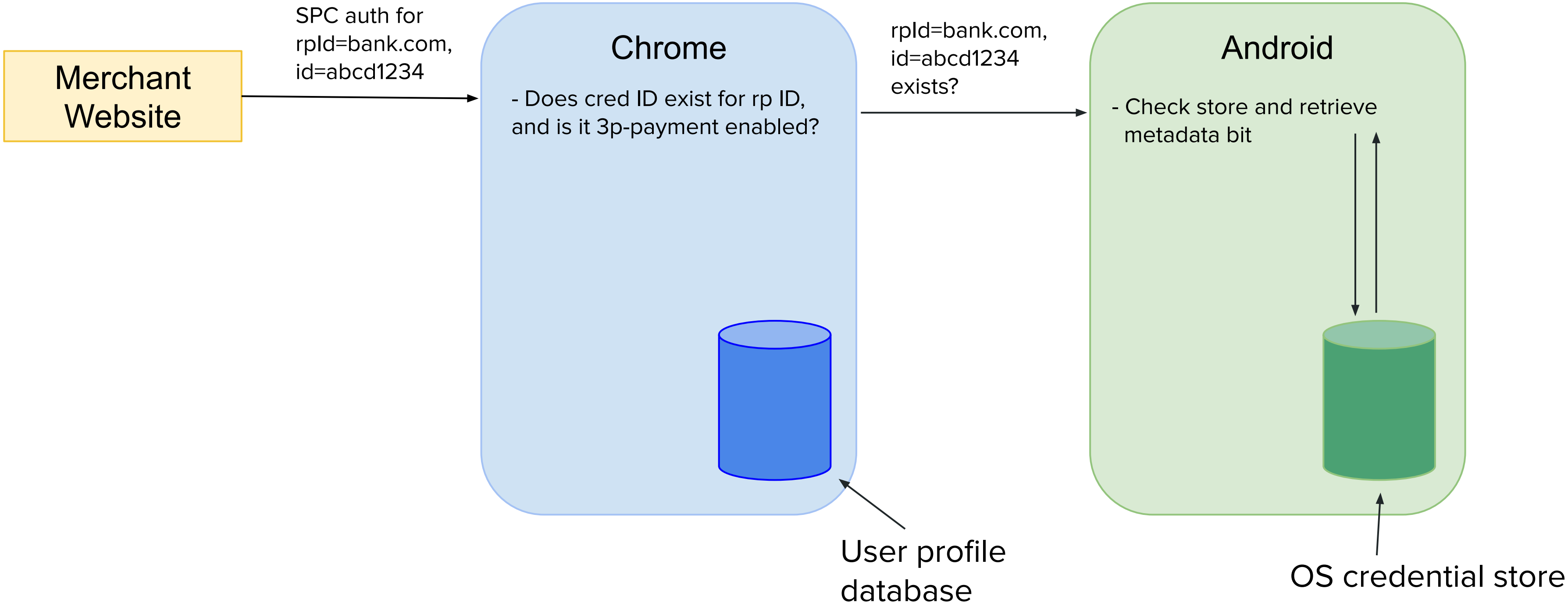
User profile database

OS credential store

SPC Android - doing it right



SPC Android - doing it right



Merchant Website

SPC auth for
rpId=bank.com,
id=abcd1234

Chrome

- Does cred ID exist for rp ID, and is it 3p-payment enabled?

rpId=bank.com,
id=abcd1234
exists?

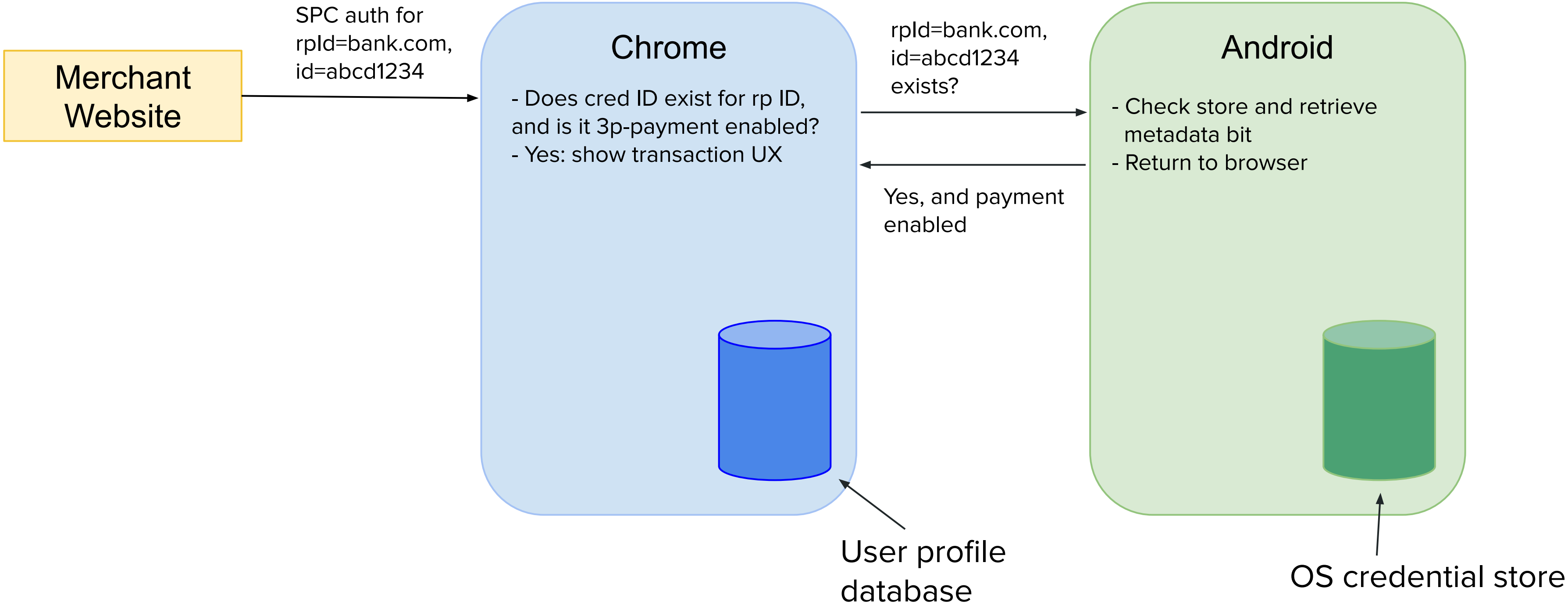
Android

- Check store and retrieve metadata bit

User profile database

OS credential store

SPC Android - doing it right



Merchant Website

SPC auth for rpId=bank.com, id=abcd1234

Chrome

- Does cred ID exist for rp ID, and is it 3p-payment enabled?
- Yes: show transaction UX

rpId=bank.com, id=abcd1234 exists?

Yes, and payment enabled

Android

- Check store and retrieve metadata bit
- Return to browser

User profile database

OS credential store

SPC Android - consequences

- Calling SPC in a first-party context (calling origin == Relying Party) will work for **any** WebAuthn credential on Android.
  - Even if it was created before SPC was even thought of.
  - No need to set 'payment' extension at creation time - only needed for third-party payment.

- Third-party payment status is no longer browser-scoped.
  - Create credential on one browser, use on another!
  - Caveat: Currently SPC is Chrome-only, so not **that** useful...

SPC Android - OS API changes

- Creation: added a 'thirdPartyPayment' bit that can be set by browser
    - Credential store puts this information alongside the stored credential data

- Authentication: built on top of existing listCredentials(rp_id) API:
    - Now returns payment bit alongside existing information
    - Browser retrieves all credential metadata for input RP ID, then for each credential:
        - if ID not in input SPC credentialIDs, skip
        - if caller-origin != rp_id **and** third-party payment bit not set, skip
    - If any credentials left after this - success!

- Android-specific complication: lack of Discoverable Credentials.
    - Turns out not to matter (Android-specific solution :D).

Demo?

The Future - What's Next?

- Ship Opt Out?
  - Question of need.

- Move to 'credential store' APIs on Chrome MacOS
  - Will allow use of 'normal' WebAuthn credentials in first-party scenarios on MacOS (like Android).
  - However, on MacOS Chrome currently uses **its own** credential store, not the platform one.
  - So doesn't solve the cross-browser problem (which applies to WebAuthn too on MacOS).

- Other:
  - Changes to Transaction UX (issuer logo? network logo?)
  - Better experience for users/merchants when no credential exists (somehow!)
  - Support for other platforms - Chrome OS, Linux
  - API ergonomics? (Finally get away from PaymentRequest? ;))

Questions?