

Web Authentication

User Journeys

Simple Secure Authentication

Steven Soneff - sso@google.com

Dec 2018

Original Authors:

UX: tringuye@

PM: cbrand@

WebAuthn enables user journeys that are

Simple – intuitive and easy for user

Secure – resistant to phishing, re-use, etc.

Authentication has two core user journeys

01

Bootstrap

User authenticates to a service for the first time

02

Re-Authentication

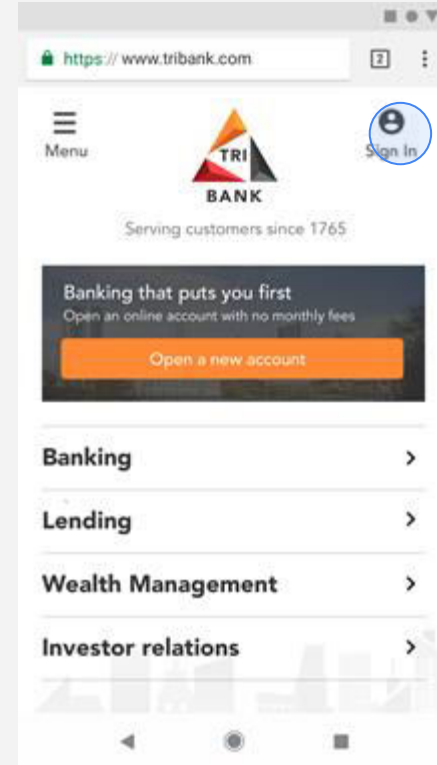
User does a repeat authentication to a service

...The next slides will walk through these user journeys as a user might encounter them on the web

1. Registering built-in authenticator for reAuth (mobile web)



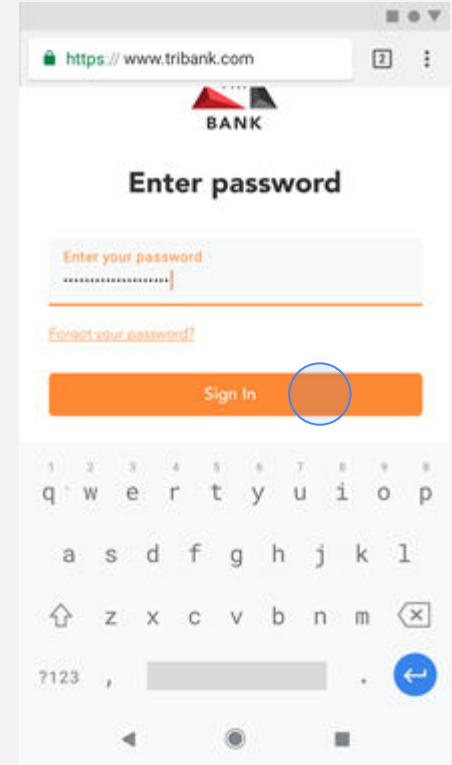
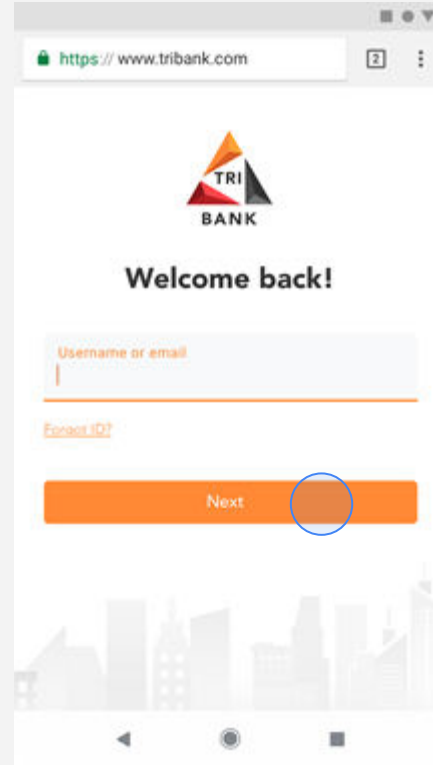
Elisa opens launches her mobile browser, Chrome, and goes to Tri-Bank



1. Registering built-in authenticator for reAuth (mobile web)



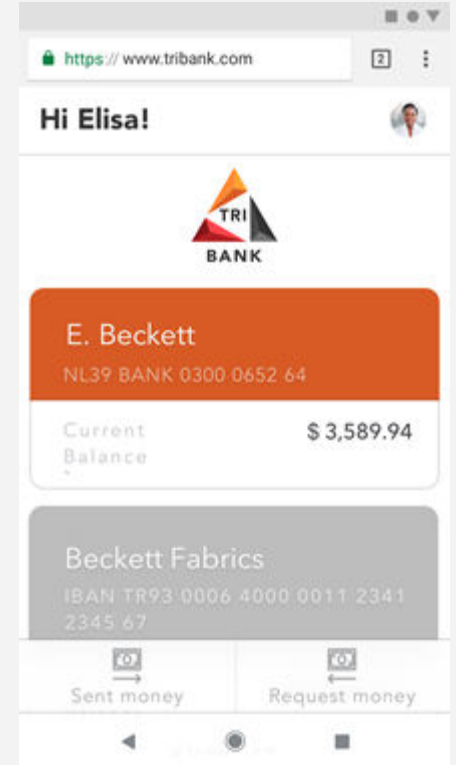
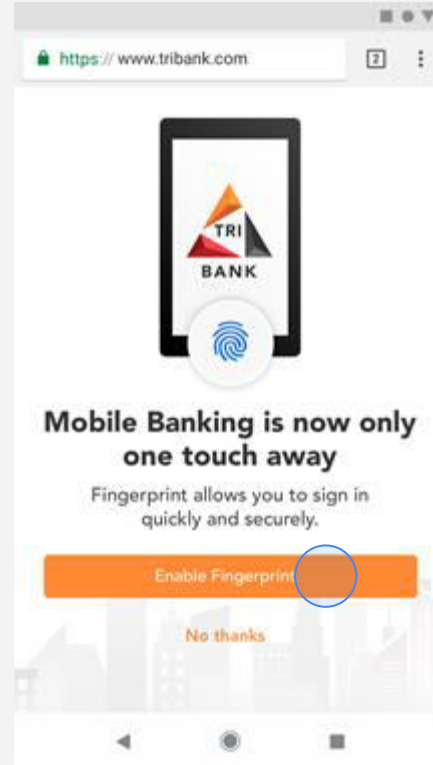
She signs in with her **username** and **password** (+potentially other factors)



1. Registering built-in authenticator for reAuth (mobile web)



Tri-Bank shows a **promo** asking Elisa if she wants to **opt-in to use Fingerprint to sign-in.**

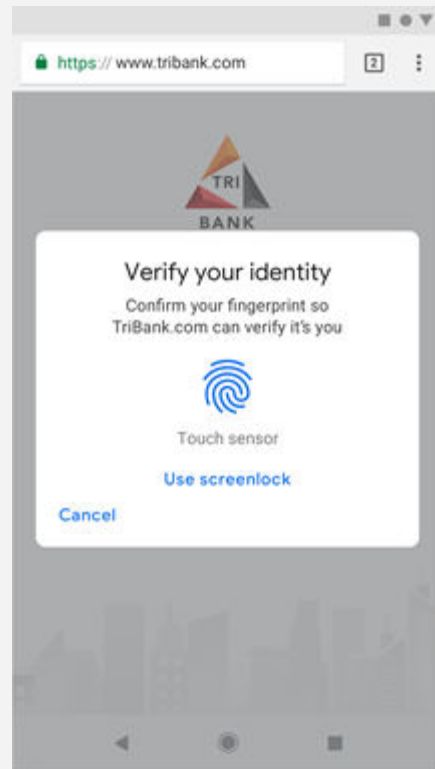
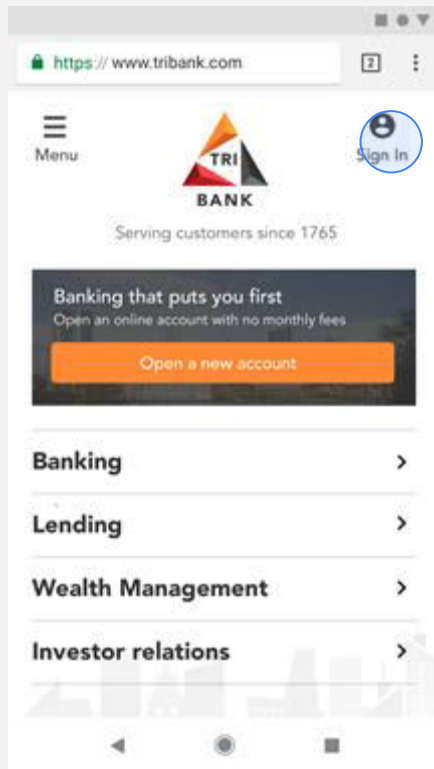


Elisa comes back to Tri-Bank in another session

2a. Using built-in authenticator for reAuth (mobile web)



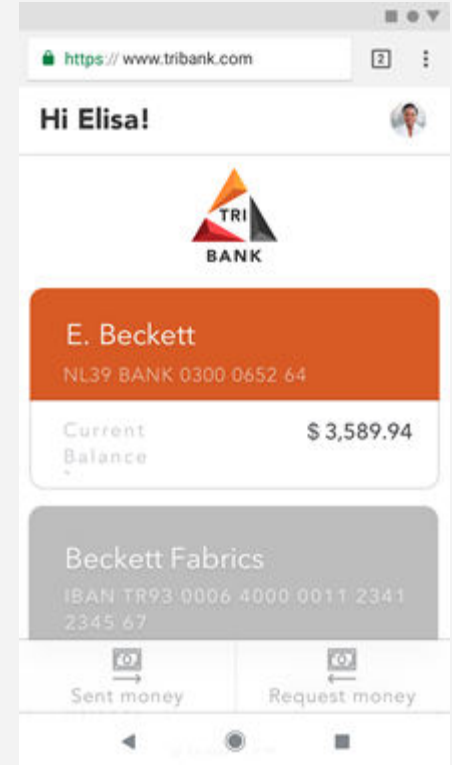
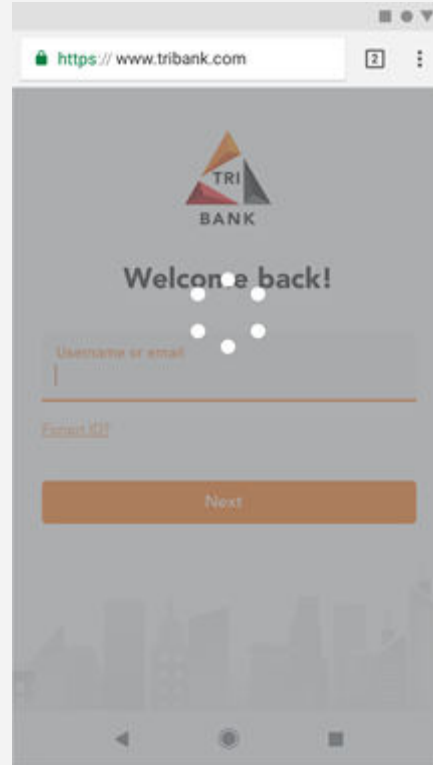
The next time **Elisa** opens Tri-Bank on **mobile browser**, she gets a **fingerprint dialog**



2a. Using built-in authenticator for reAuth (mobile web)



Using only her fingerprint,
she's able to sign-in
without using her
username + password on
mobile web



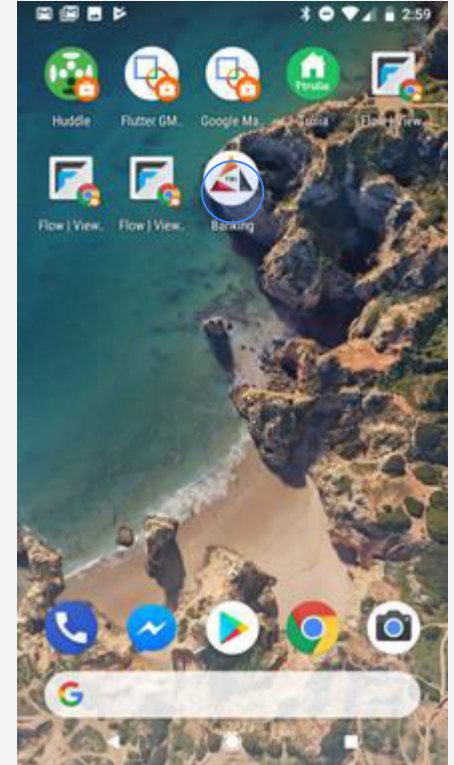
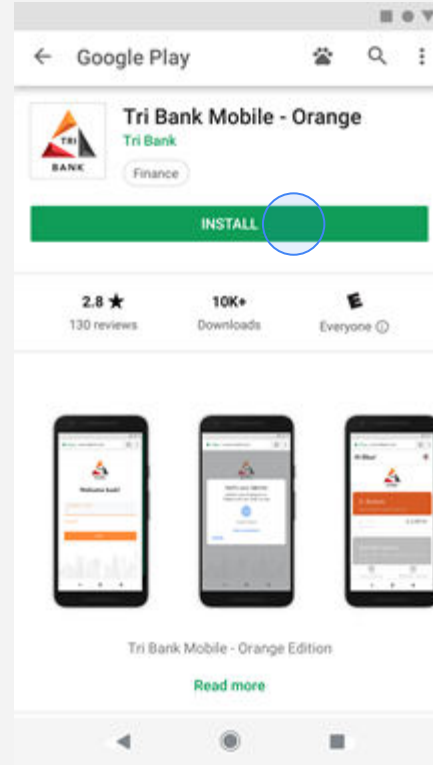
Using built-in authenticator for reAuth (native mobile app)

Elisa downloads Tri bank from the Play Store, she **launches the app for the first time** to sign in to check her funds

2b. Using built-in authenticator for reAuth
(native mobile app)



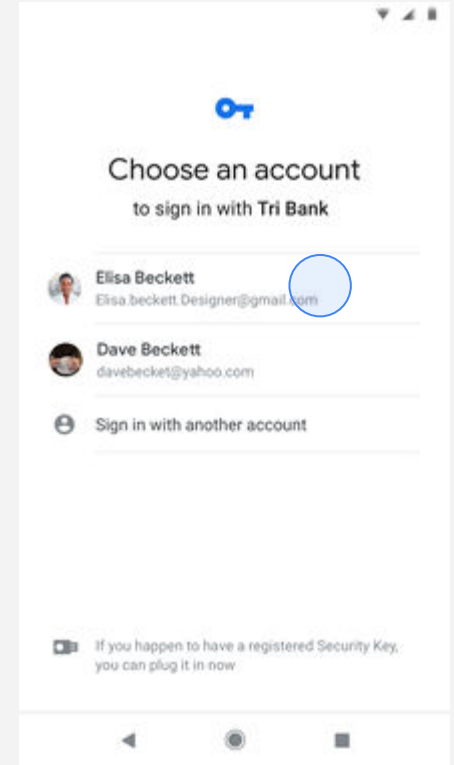
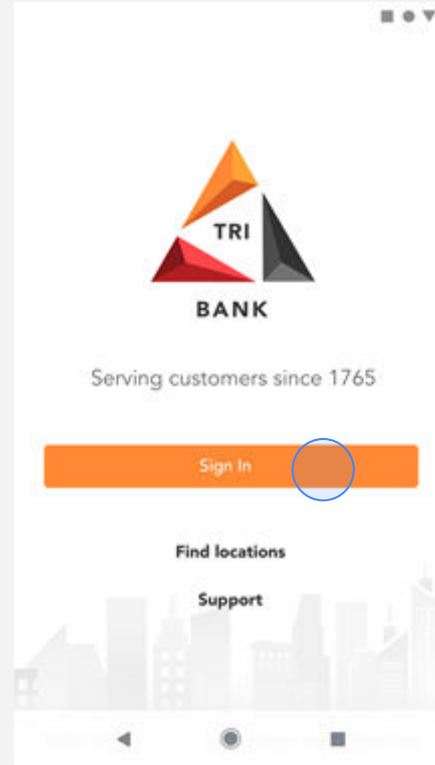
**She installs Tri Bank from
Google Play Store and
opens the app**



2b. Using built-in authenticator for reAuth (native mobile app)



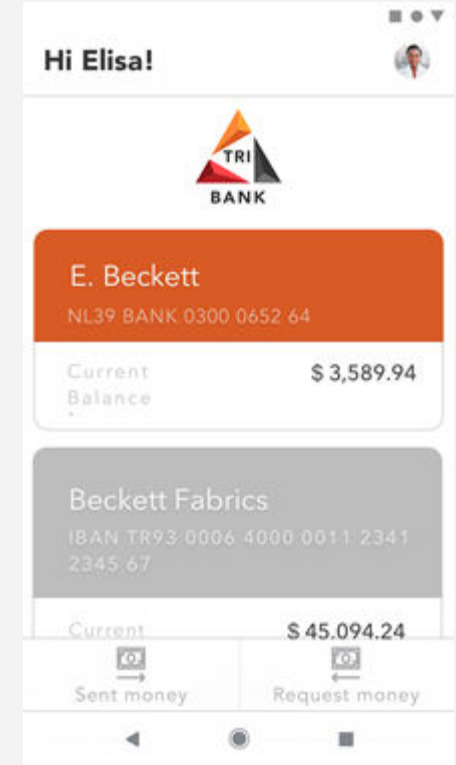
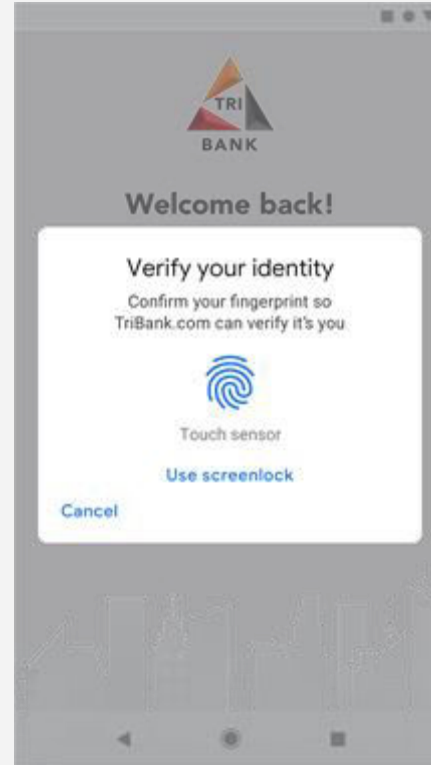
**Elisa chooses sign in
and also chooses an
account.**



2b. Using built-in authenticator for reAuth (native mobile app)



Elisa now is asked to
authenticate **with the
fingerprint dialog**

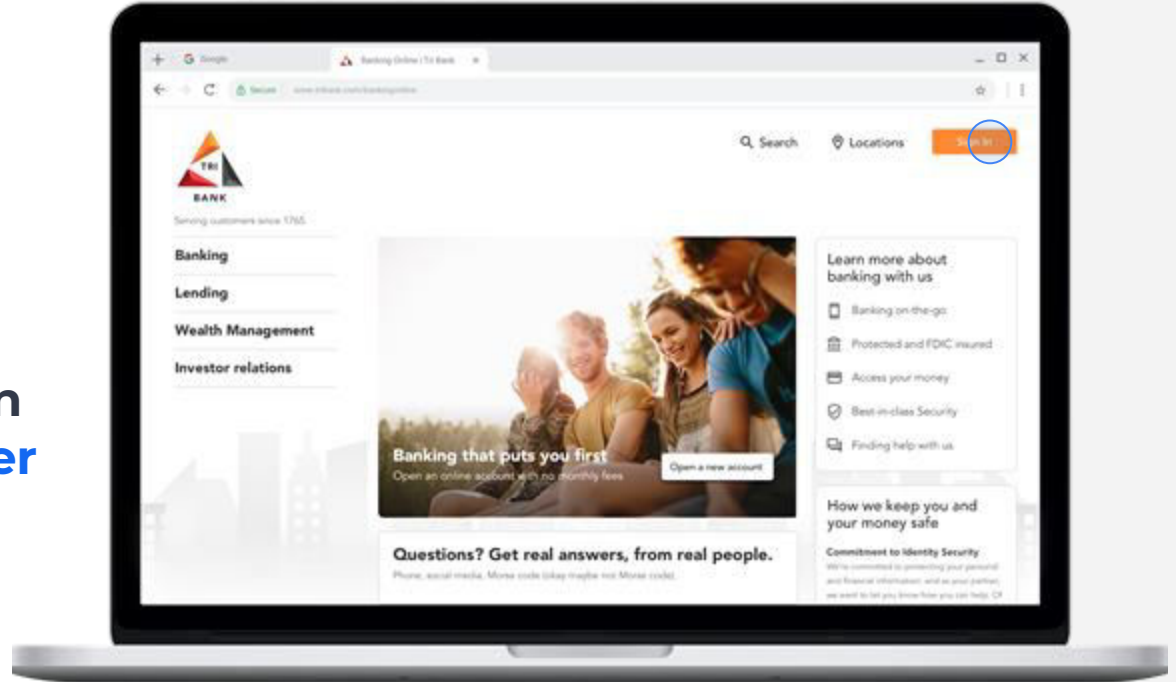


Elisa wants to sign in
to her bank on her
desktop computer

3i. Cross-Platform bootstrap



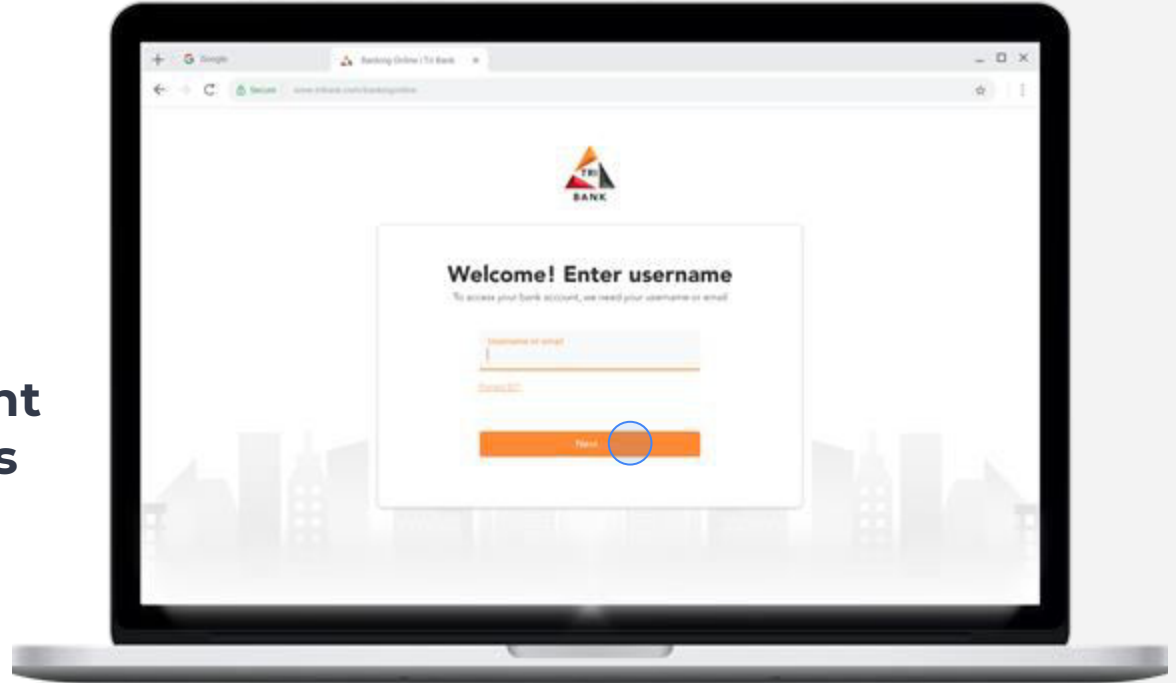
Elisa chooses to sign-in
on her **desktop browser**



3i. Cross-Platform bootstrap



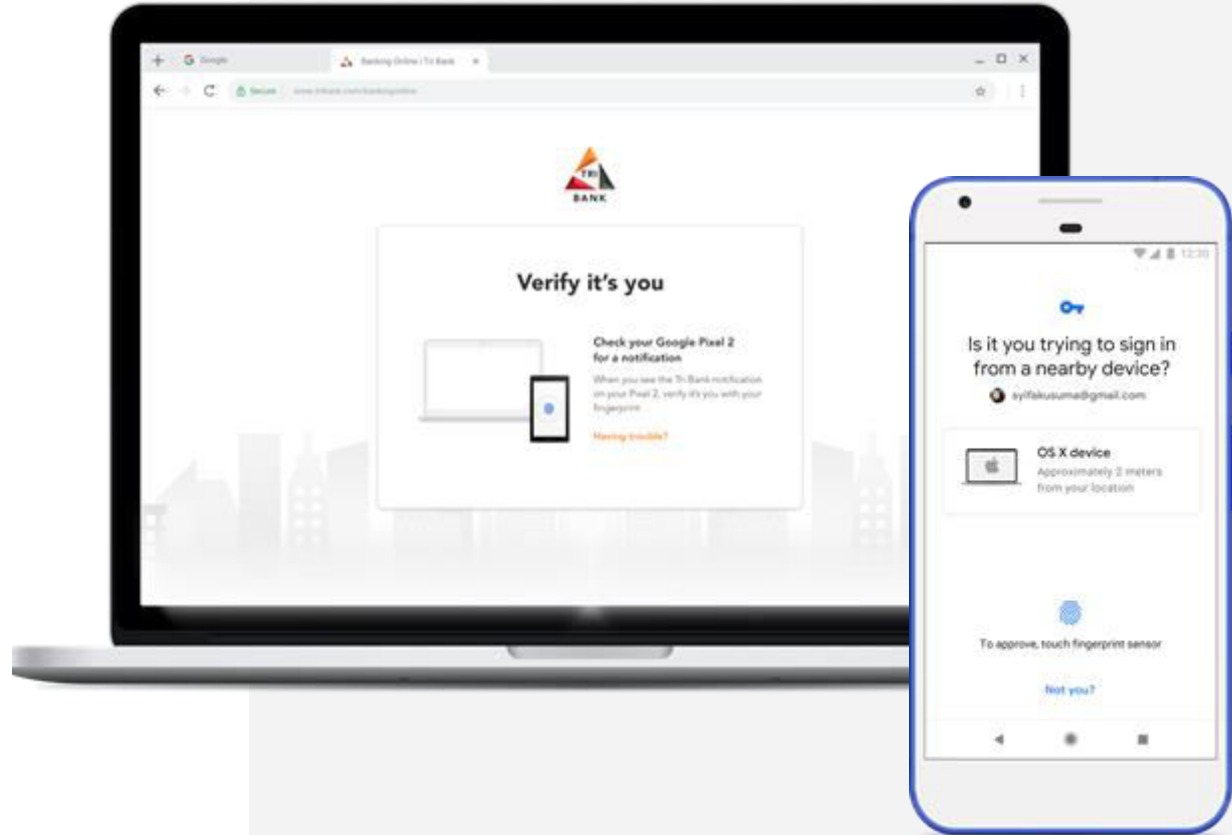
Elisa enters her account **username** and chooses to proceed 'next'



3i. Cross-Platform bootstrap



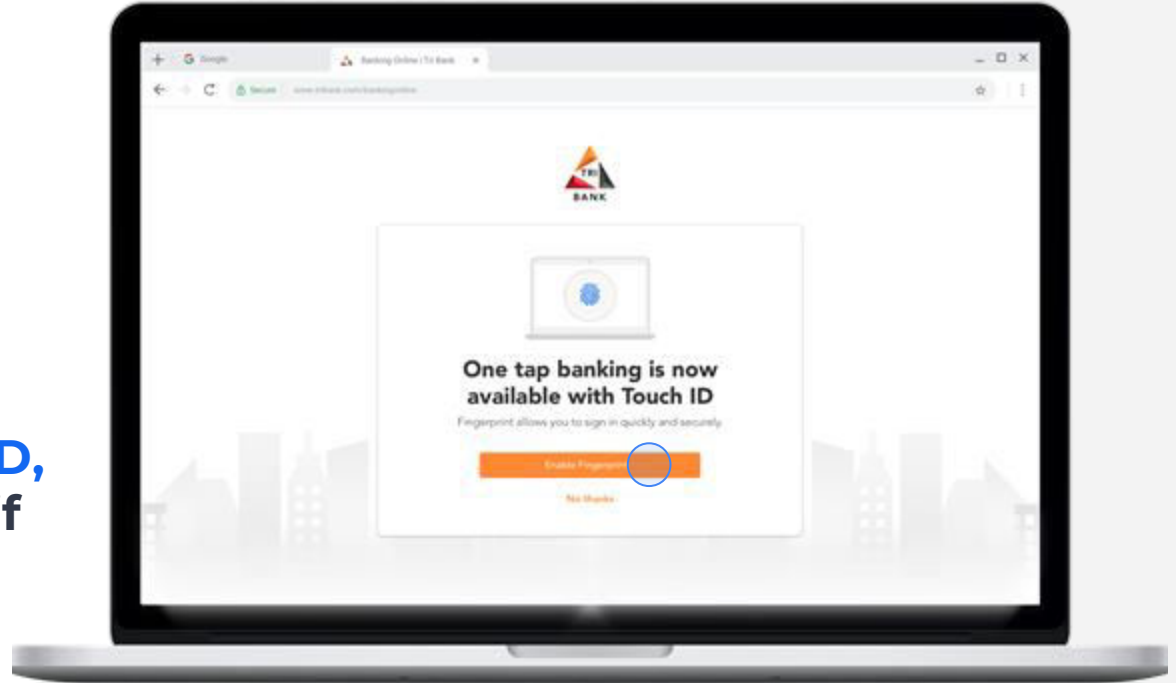
She's asked to **verify the new device using her phone fingerprint** that she's been using to sign-in to Tri-Bank



3ii. Cross-Platform bootstrap



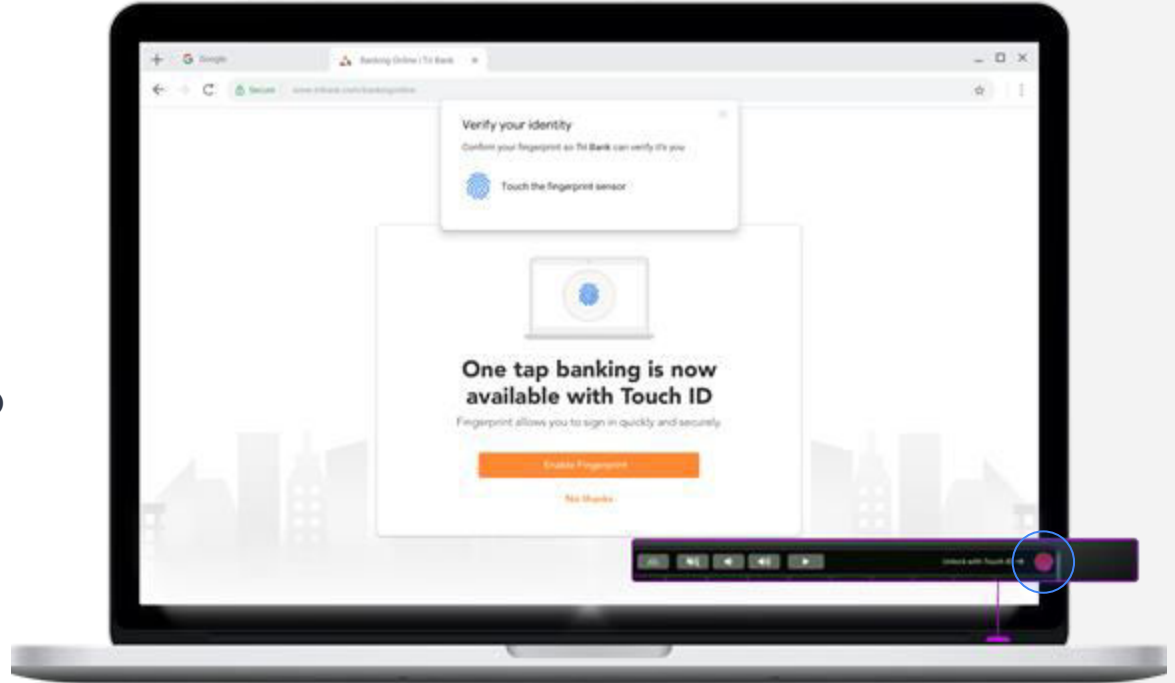
Because Elisa has a **Macbook with Touch ID**, Tri-bank can ask her if she wants to **use local fingerprint on the device**.



3ii. Cross-Platform bootstrap



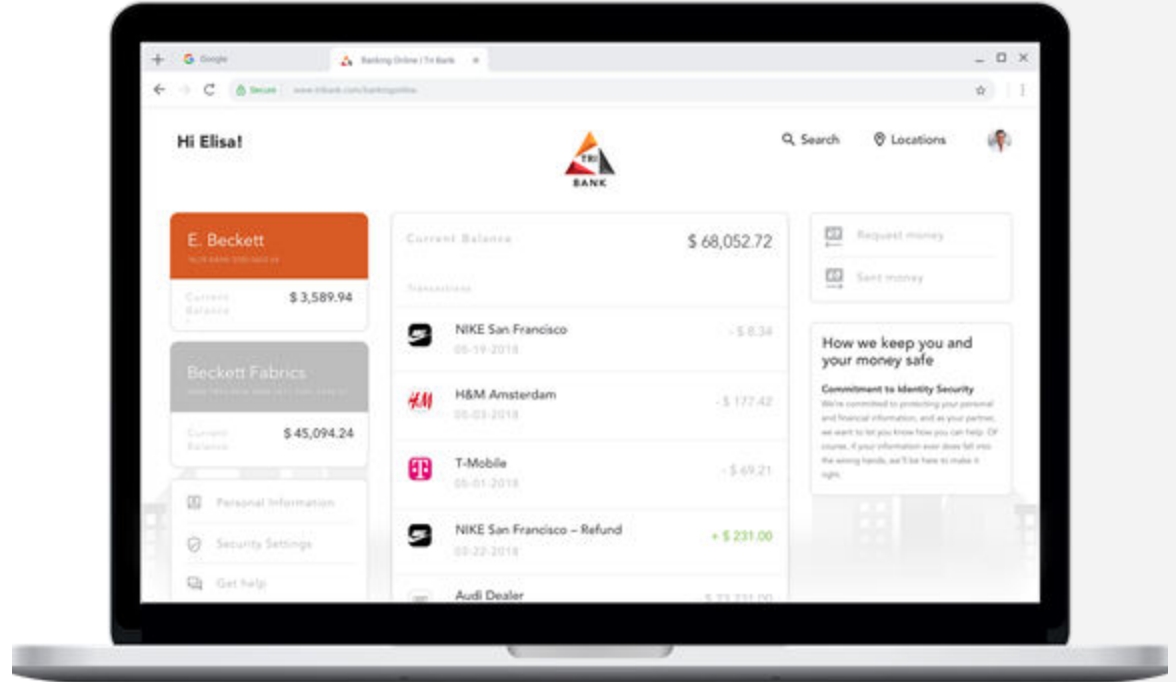
Elisa gets prompted to try using the local fingerprint on the device.



3ii. Cross-Platform bootstrap



She **opts-in** and continues to her account

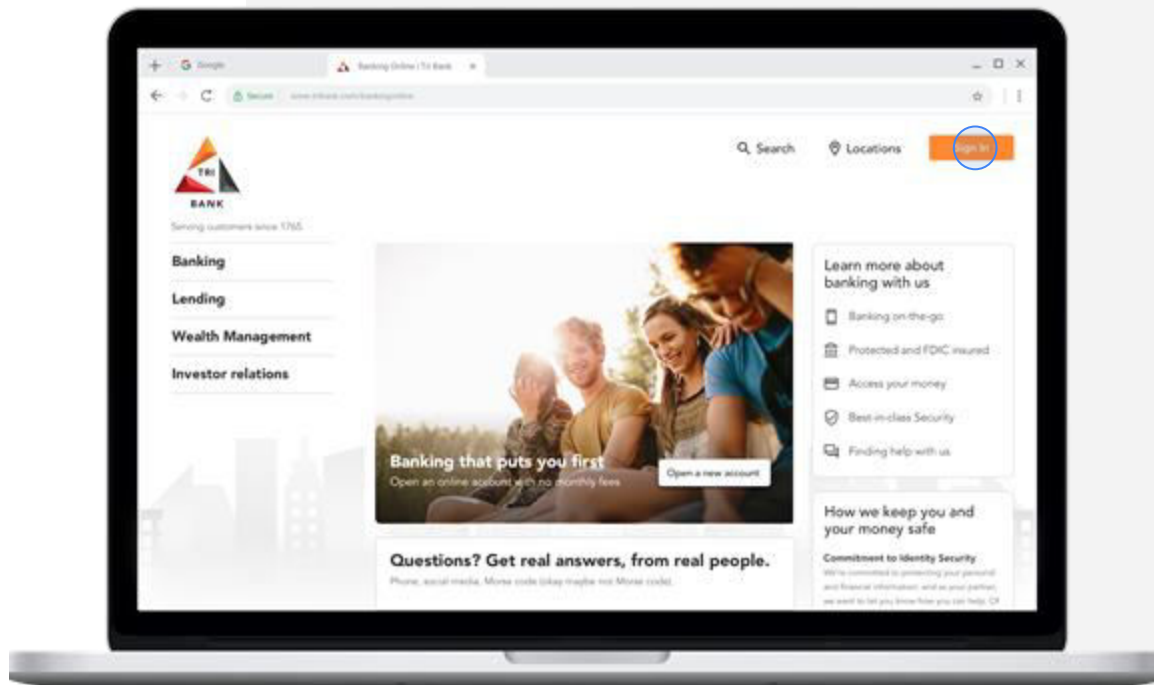


...when Elisa comes back to Tri-Bank on the Macbook Pro

4. Using built-in authenticator for reAuth



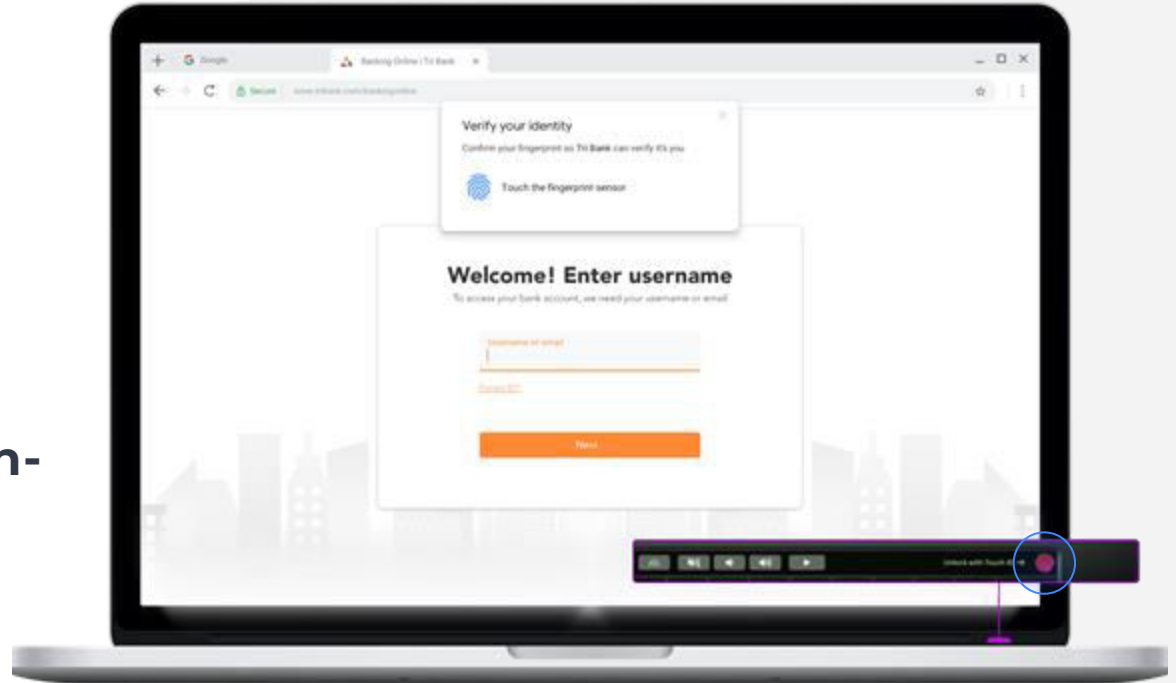
Elisa comes back to sign-in on her desktop browser



4. Using built-in authenticator for reAuth



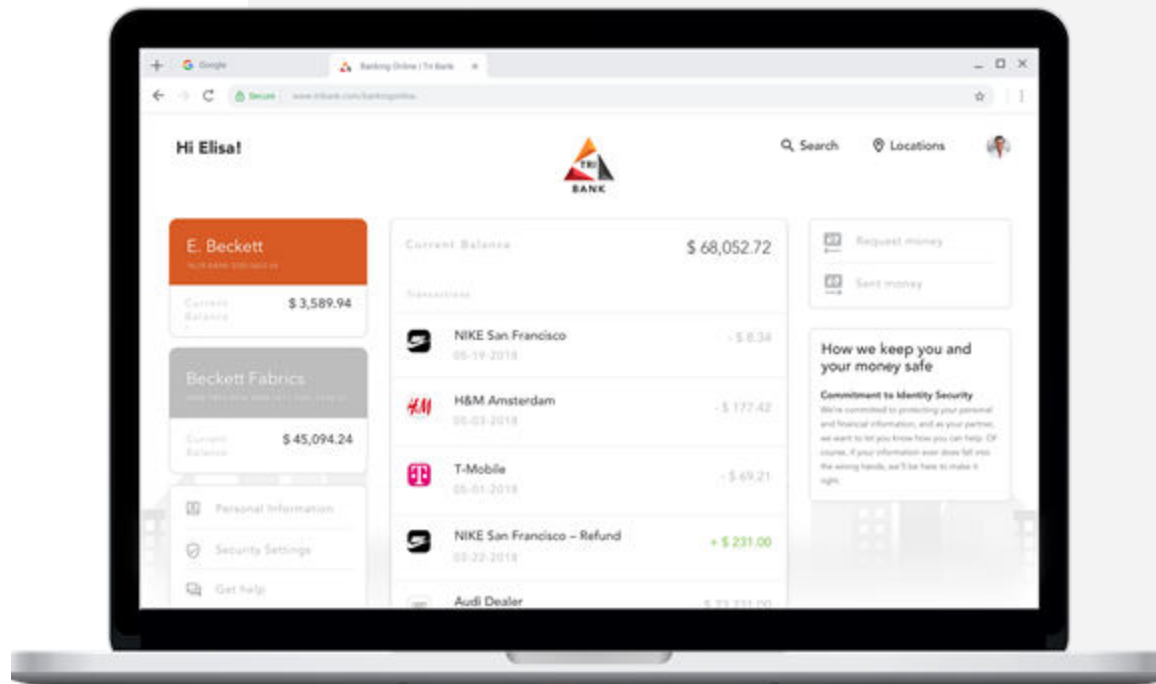
A **fingerprint dialog** appears above the sign-in page and **Elisa touches the sensor**



4. Using built-in authenticator for reAuth



**Elisa's identity is
accepted and she's
signed in!**



Note that we're inheriting the strength of the credentials from the initial bootstrap.

Summary

Simple - avoid typing, avoid passwords, minimal decisions

Secure - passwordless, multi-factor, phishing-resistant

Steven Soneff - sso@google.com
Web Platform Product Manager