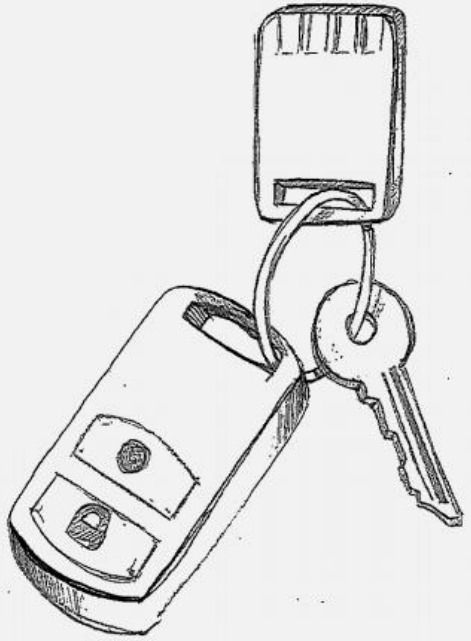
Google U2F (Gnubby) Documents - Snapshot prior to joining FIDO



Google™

U2F - Universal 2nd Factor

Web Keychain Device for users

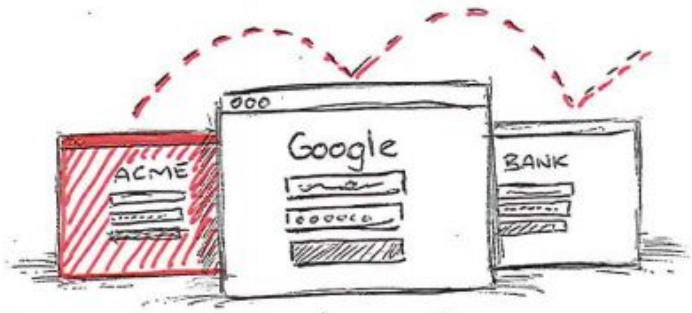
open standard strong authentication for the web

What we'll cover

- **U2F Overview**
  - Problem being solved
  - Value to the end user
  - Value to the Service Provider (RP)
  - Value to the device vendor, integration vendor
- **How U2F works**
  - Protocol design considerations
  - Integration into browser
  - More use cases
  - Current Status
- **The larger view: FIDO Alliance**
  - Device Centric Auth
  - FIDO offerings as a complementary whole

Google

Web passwords are broken



REUSED **PHISHED** **KEYLOGGED**

Google

Today's solution: One time codes: SMS or Device

SMS USABILITY

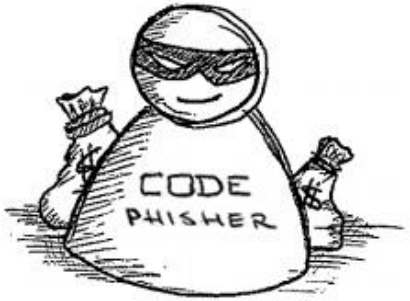
Coverage Issues - Delay - User Cost

DEVICE USABILITY

One Per Site - Expensive - Fragile

USER EXPERIENCE

Users find it hard

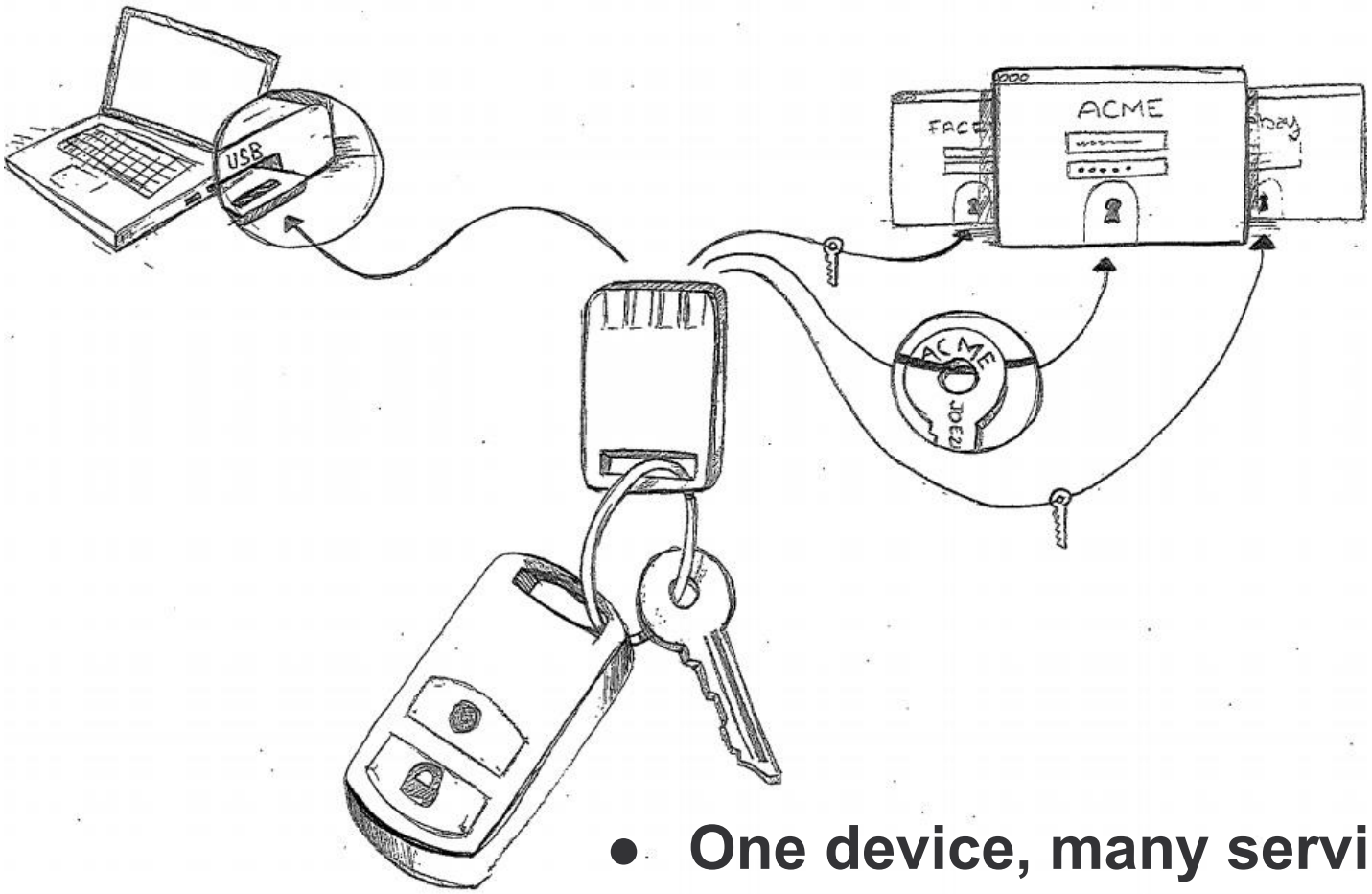


PHISHABLE

German Police re: iTan:
".. we still lose money"

Google

The U2F solution: How it works



- **One device, many services**
- **Easy: Insert and press button**
- **Safe: Un-phishable Security**

Google

Simple for Users

1

Userid & Password

2

Insert, Press button

3

Successful Sign in

Google

User self-registration

1 **Userid & Password**

2 **Insert, Press Button**

3 **Backup Options**

4 **Registration Done**

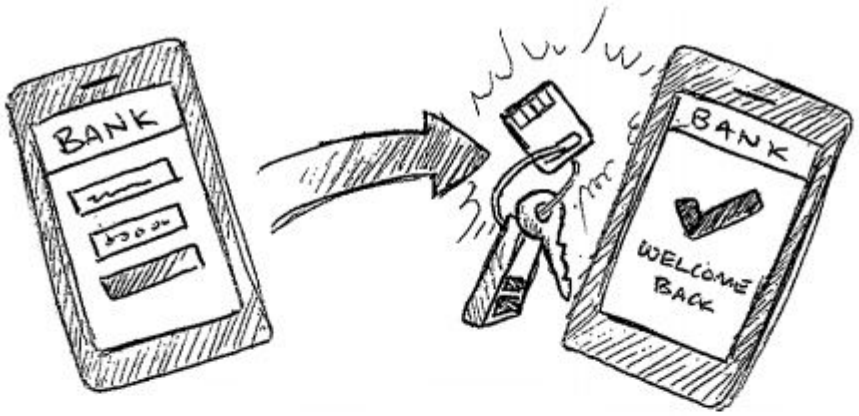
Google

Usage on Mobiles

Tomorrow

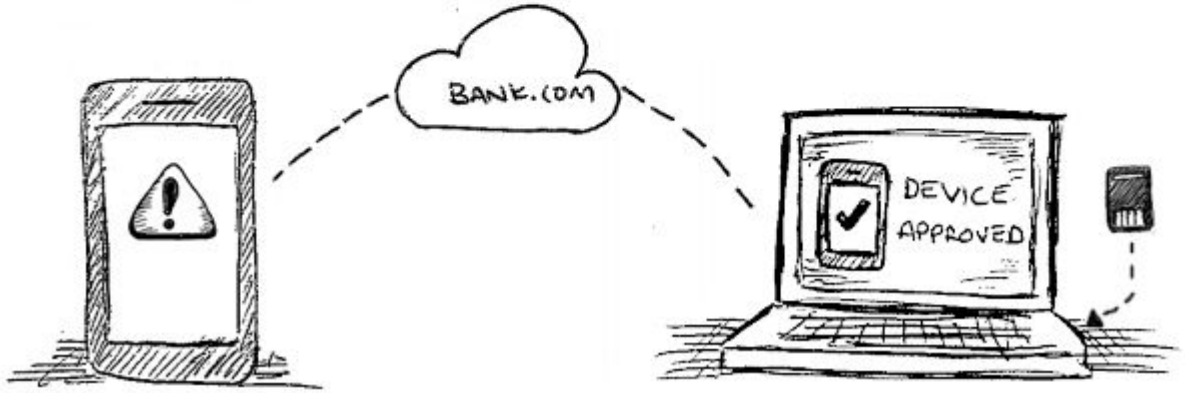
Tap your NFC-aware device to your NFC-enabled phone

(modulo, choosing the right app isolation tradeoff ...)



Today

Use your computer to bless your mobile (one time action)



Google

Like a real key: Small, Reliable, Secure

No Batteries

Vigorously Tested

Secure Element Guarantees

Google

U2F Protocol

Core idea: Standard public key cryptography:

- User's device mints new key pair, gives public key to server
- Server asks user's device to sign data to verify the user.
- **One device, many services, "bring your own device" enabled**

Lots of refinement for this to be consumer facing:

- **Privacy**: Site Specific Keys, No unique ID per device
- **Security:** No phishing, man-in-the-middles
- **Trust:** Verify who made the device
- **Pragmatics**: Affordable today, ride hardware cost curve down
- **Speed for user:** Fast crypto in device (Elliptic Curve)
- **Feature Growth:** Server<->device encrypted comm.; future trusted display

Think "Smartcard re-designed for modern consumer web"

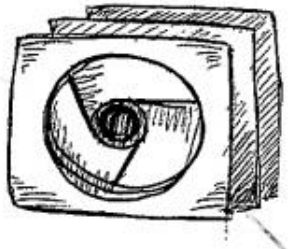
Under the hood

- **Device core:** Secure element accessed over USB or NFC
- **Driverless USB on Win, Mac, Linux, Chrome OS**
  - Just plug in and use
- **Direct Access from Browser:**
  - **No client middleware to install**
  - Simple Javascript API: **'Create Key Pair'** and **'Sign'**
  - Not just tied to login! Use anytime you want to strongly verify user.
- **Same API on Android**
  - Just integrate with your app
- **UI seen by user completely under server control**
- **Server side integrates easily with existing auth services**

Google

Open Ecosystem: Virtuous Cycle

CONSULTING



BROWSER SUPPORT

OPEN

joining

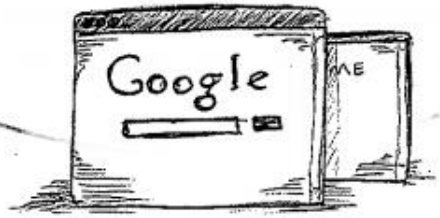
FIDO ALLIANCE

creating

U2F WORKING GROUP

MOBILE SUPPORT

MANY DEVICES & FORM FACTORS



ADOPTION

Google

U2F - Universal Second Factor: In a nutshell

- **User carries a strong auth. device, works across services:**
  - Small USB/NFC dongle with secure element
  - Works out of box, no software install
  - Mental model "Like a key on your chain, a card in your wallet"

- **For the user: Easy Secure Login**
  - One device, Many services
  - Simple UX - Insert and press button or tap, no software install
  - **Passwords can be made simple -- 4 digit pins like ATM?**
  - Very rugged and reliable, like a real physical key

- **For the web site: Open Strong Security**
  - **Open:** Not proprietary, multiple vendors, **no central service required**
  - **Self provisioned:** No pre-seeding req, "Bring your own token" possible
  - **Strong Security:** Non-Phishable, Blocks most practical MITMs
  - **Strong Privacy:** One site cannot use credential given to another

Other usage models beyond "One key you carry"

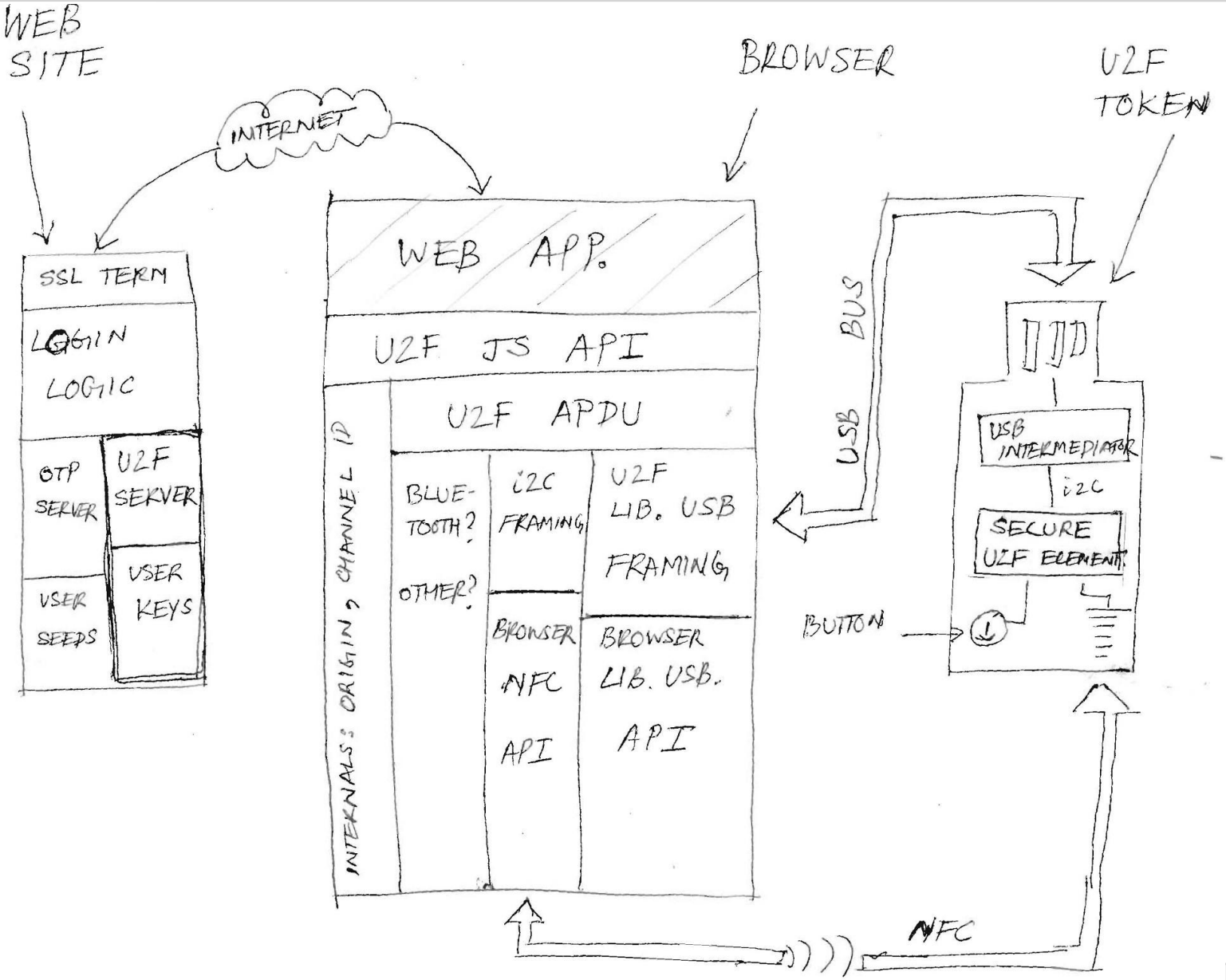
1. **Token plugged permanently into home machine**
   - husband and wife share
   - husband for paypal and google, wife for schwab and amazon
2. **One token plugged into home, one token plugged into work**
   - User provisions both for paypal, can pay from either place
3. **One token plugged into home, one token to carry**
   - Convenience, home computer always ready to go
4. **One (tiny) token plugged permanently into work laptop**
   - Laptop becomes the 2nd factor (maybe built into next-gen laptops?)
5. **Husband/wife, separate tokens, plugged in simulatenously**
   - Each activates own key, protocol has no problem with multiple keys
6. **One account, multiple users, each with own token**
   - Small business users share an account with strong auth
7. **Account lockdown to a single device**
   - Only one token, plugged into office machine
8. **Same token for work account and personal account**
   - Work (= enterprise) leverages user's "bring your own token"
9. **Different token for work account and personal account**
   - If enterprise doesn't like self-provision, so ships pre-provisioned token

Google

Current Status

- **Google "intranet" single sign on is U2F enabled**
  - Our "intranet" is directly on the web, so its just web login
- **Using a Chrome Extension, step towards browser integration**
  - Not using final JS API
  - Integration at "lower" level, but its just eng execution
- **Many thousand devices in daily use**
  - Compliant devices by Yubico, built around NXP "U2F" secure element.
  - Various software milestones since October
  - Now moving into rapid scaling of Beta
  - Intend to fully replace OTP for employees by end of year
  - Will be ~100,000 units in deployment
- **Solving use cases for non-web "legacy" clients**
  - Eg, VPN client via Browser extension
  - We are rolling our own, but fertile ground for ISVs for commercial use
- **Glad to help interested RPs to a proof of concept**
  - You implement server side with FIDO specs, code fragments
  - Compliant token devices already available
  - Get experience on how the end to end experience works

U2F Schematic



Google U2F (formerly Gnubby) - pre-FIDO

Google

Next Section: Larger View: FIDO Alliance

Google The Larger View as Google sees it

- **Core need for web service: Ask user for permission**
  - "You are logging in to create a new session. Please approve"
  - "You are deleting all your email. Please approve"
  - "You are transferring $100,000 to Sam. Please approve"
  - "You are shipping your purchase to a new address. Please approve"
- **Done today with passwords today and maybe OTPs**
  - Difficult for user, insecure (phishable, MITM, not malware resistant)
- **Megatrend: Users moving to varied personal devices**
  - User devices have user-specific local authentication (screen lock etc)
  - PINs today, biometric on horizon -- many different kinds of local auth.
- **Opportunity: Easy and Secure Web authentication**
  - Web service asks user for permission
  - To approve, user does user-specific local auth on their device
  - Web service gets some crypto proof of permission.
  - Passwords no longer required for routine aut

Google A Solution Framework

- User's device has keystore unlocked by user's local auth
  - Each user of device has own keystore space
  - Devices will have different kinds of local auth (PIN, Various biometrics)

- User **registers** device to web service
  - Creates a web service specific key pair
  - Key creation enabled by local auth
  - Hands public key to web service

- Web service **verifies** user permission by
  - Asking for signature matching public key
  - User does local auth, unlocks keystore, signs with private key

- **In gist: User does simple auth gesture on personal device to easily and securely approve a website's request**

- **Very aligned with FIDO alliance's**
  - ***So why not just join them and take U2F under the FIDO banner?***

Google U2F (formerly Gnubby) - pre-FIDO

How the parts of FIDO fit after Google joins

- **Existing FIDO efforts (technical working group)**
  - Larger View, password less, local device auth for sign
- **U2F = Universal 2nd Factor**
  - Critical bridge to future, "classic" 2-factor, incremental change for RP
  - Service (RP) password still present, but can be simple (4 digit PIN?)
- **Fit together as one whole for Service Provider (RP)**

  - **At registration time:**
    Discover user has FIDO passwordless enabled device?
    Register for passwordless experience
    Else offer user FIDO U2F:
    Self-register for simple password + 2 factor experience
  - **At login time:**
    User has FIDO passwordless enabled device + enrollment?
    Give user passwordless "just unlock gesture" experience
    Else user has U2F enrollment?
    Give user simple pwd + "show a 2 factor device" experience

- **Some RPs may want only passwordless, some only U2F**
  - That's no problem: FIDO is all about the right choice for RP and user
  - Note that RP can start offering "other" flavor later seamlessly
  - Same server can talk both passwordless