Navigating Murky Waters:
Automated Browser Feature Testing for Uncovering Tracking Vectors

Mir Masood Ali, Binoy Chitale, Mohammad Ghasemisharif,
Chris Kanich, Nick Nikiforakis, Jason Polakis

UIC   Stony Brook University

NDSS SYMPOSIUM

Browsers are complex

- Wide range of APIs and features
- New web standards:
  - HTTP/1 → HTTP/2 → HTTP/3
- Progressive Web Apps:
  - Service Workers
- Google’s Privacy Sandbox:
  - Trust Tokens, FLEDGE API, etc.

Tracking is ubiquitous

- Used across online services
- Crucial for:
  - Analytics
  - Personalization
  - Authentication
  - Advertising

Re-identifying Users

- Cookies
  - First-party (1P) and Third-party (3P)
- Storage
  - Local Storage, indexedDB, HTTP Cache
- Evercookies
  - Flash cookies[1, 2], HSTS policies[3], Favicon cache[4]

[1] Kamkar, 2010. Available: http://samy.pl/evercookie/
[2] Englehardt and Narayanan, CCS ’16, Available: https://doi.org/10.1145/2976749.2978313
[3] Syverson and Traudt, FOCI 18. Available: https://www.usenix.org/conference/foci18/presentation/syverson
[4] Solomos et al., NDSS 2021. Available: https://dx.doi.org/10.14722/ndss.2021.24202

Mechanisms need Evaluation

- Browsers continue to develop new mechanisms
- Client-side implementations include:
  - Caching
  - Storage
  - Access-control
  - Policy
- Incomplete testing can open up new tracking vectors
  - Testing is tricky[1]
  - Multiple teams, massive code-bases
  - Do not cover all contexts
  - Security ✅; Privacy ❓

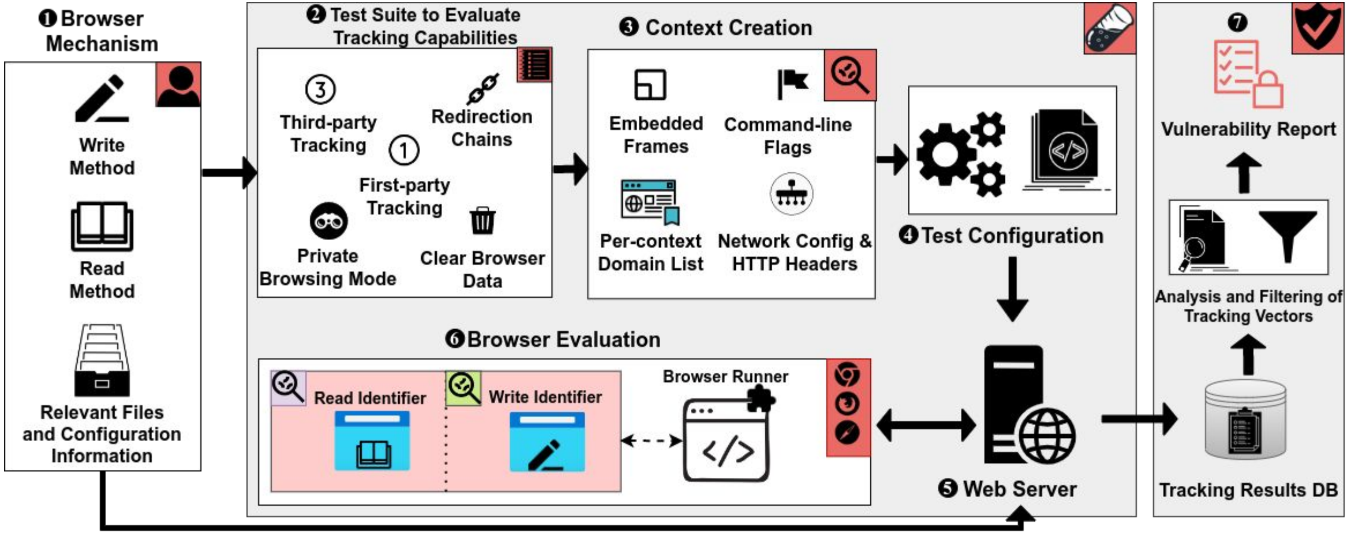
[1] Luo et al., CCS 2017. Available: https://doi.org/10.1145/3133956.3133987

Any mechanism that stores some form of data in the browser or affects client-side policies is a potential tracking vector.

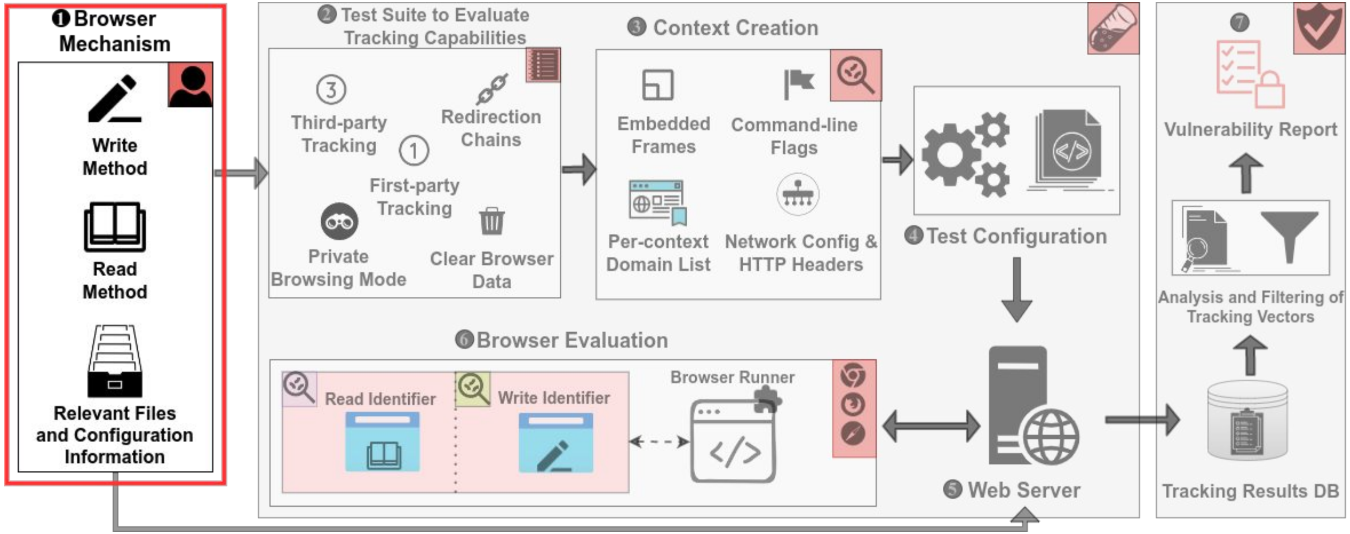
CanITrack

- Framework:
  - Streamlines testing of browser features
  - Assesses misuse for tracking
  - Systematic and comprehensive
- Against pertinent capabilities:
  - 1P & 3P Contexts
  - Persistence across sessions
  - Private Mode
  - Clearing Browser Data
- Based on a **write()** and a **read()** method

CanITrack: Overview



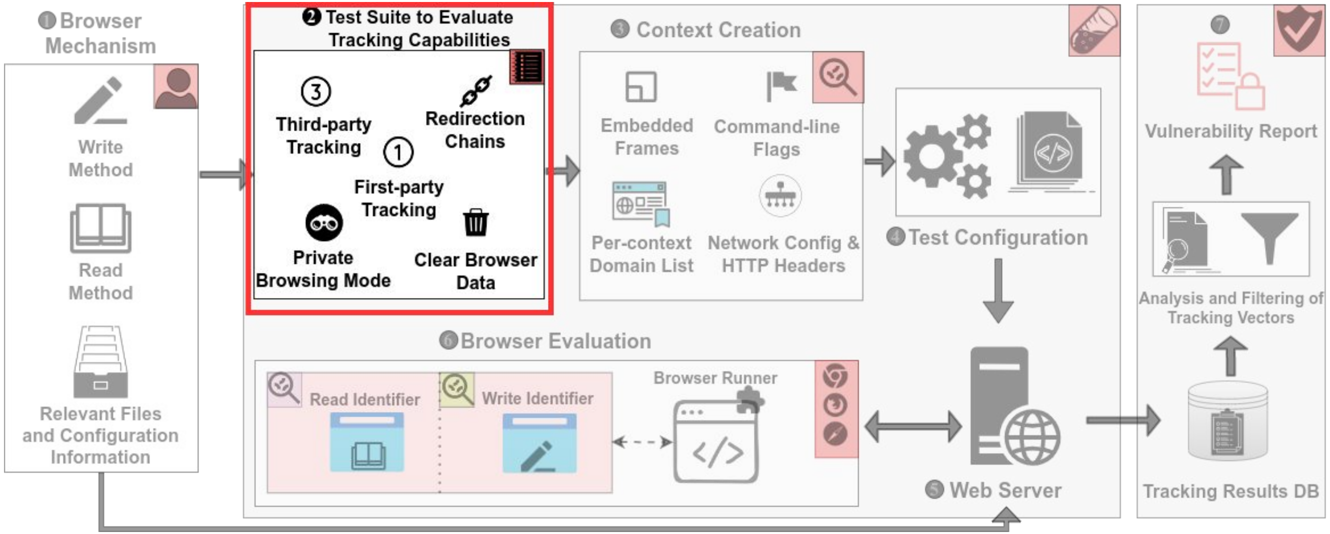
CanITrack: Browser Mechanism



1

- `write()`: set a value using the mechanism
- `read()`: read back the value from the mechanism
- Additional info. about the mechanism; which tests?

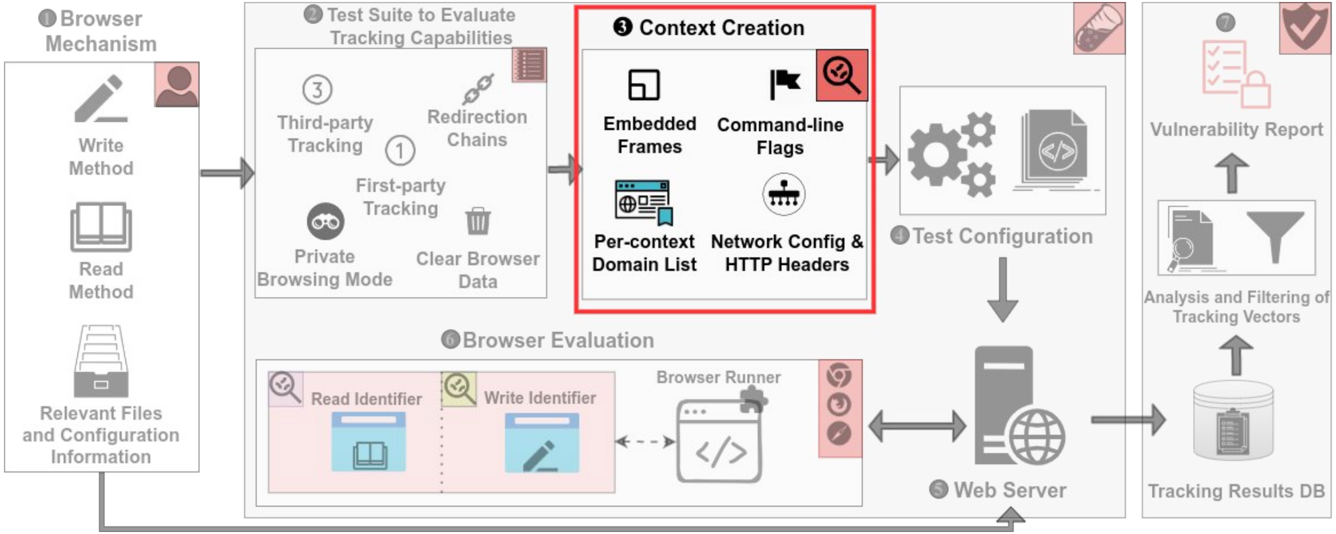
CanITrack: Test Suite



2

- Contexts: First-party (1P) and Third-Party (3P)
- Modes: Regular Browsing <--> Private Browsing
- Clearing Data: Persists despite deletion?

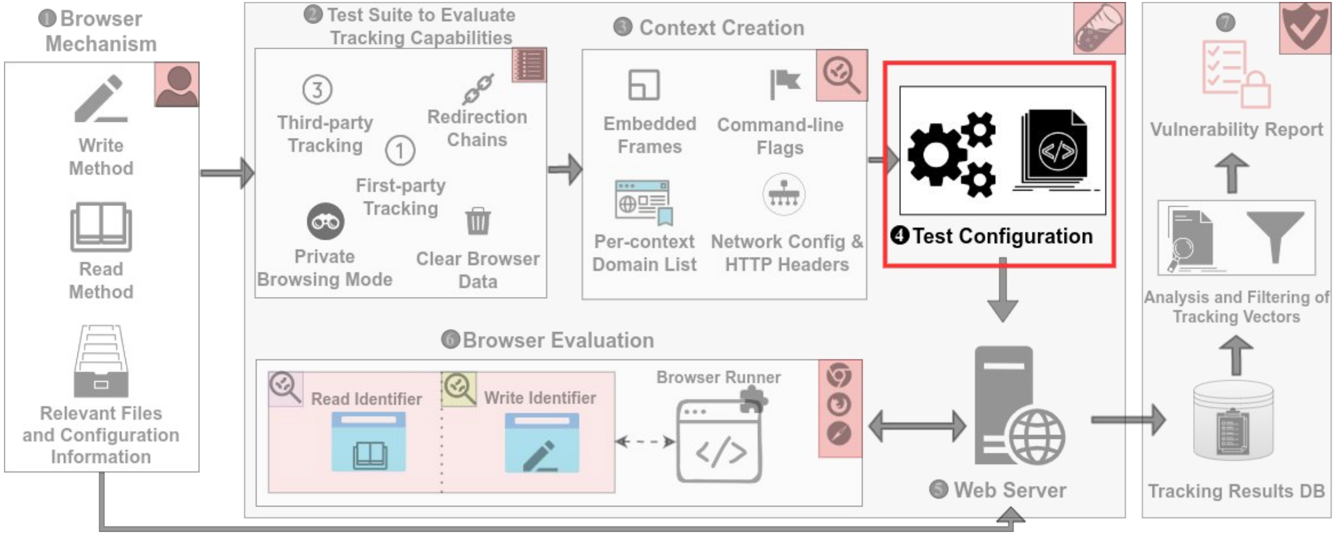
CanITrack: Context Creation



3

- Contexts: Different domains for iframes
- Command-line Flags: Experimental mechanisms
- N/w Config: Handle CORS, Auth, Cache-Control, Request Paths, etc.

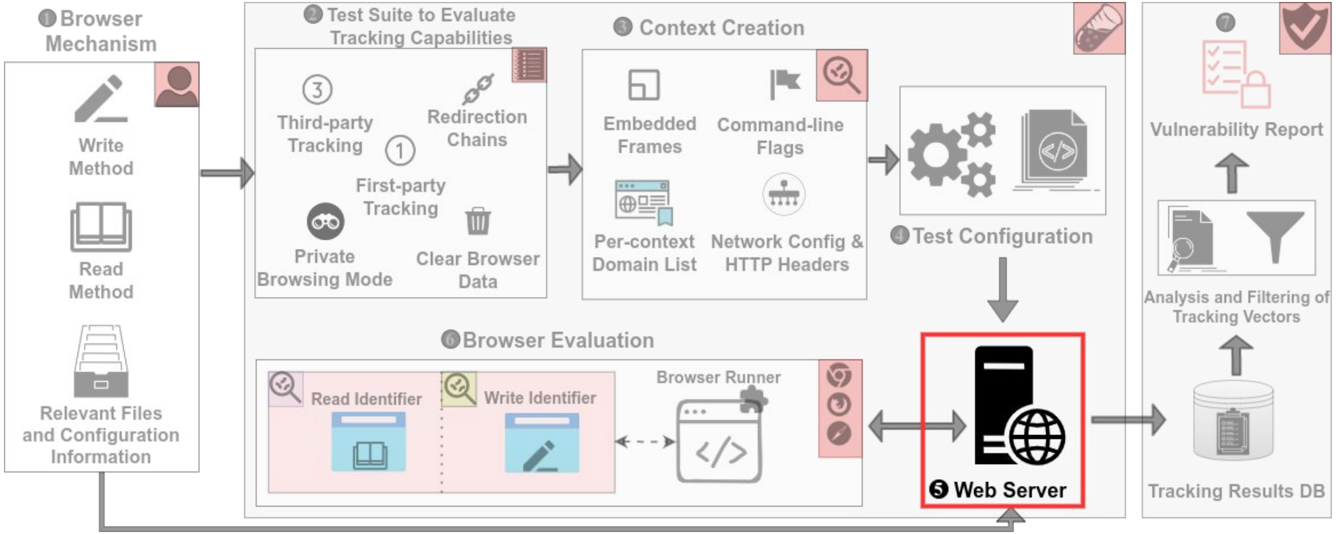
CanITrack: Test Config



4

Combine Mechanism Config from Step 1, Test Suite from Step 2, and Context from Step 3.

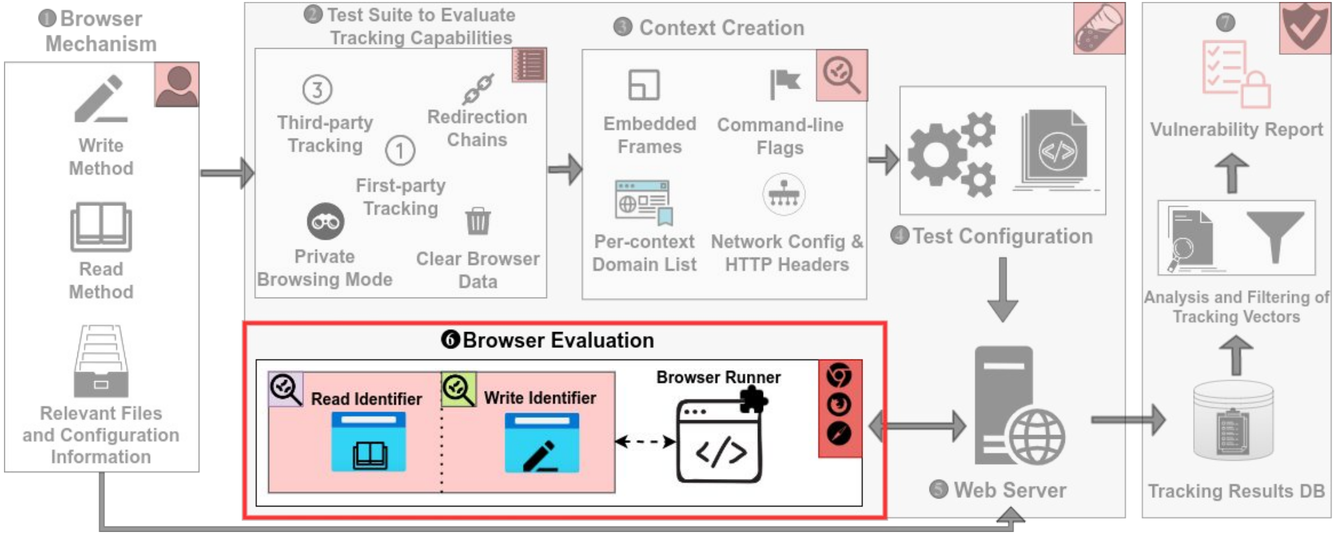
CanITrack: Web Server



5

- Host domains
- Handle requests
- Capture test reports

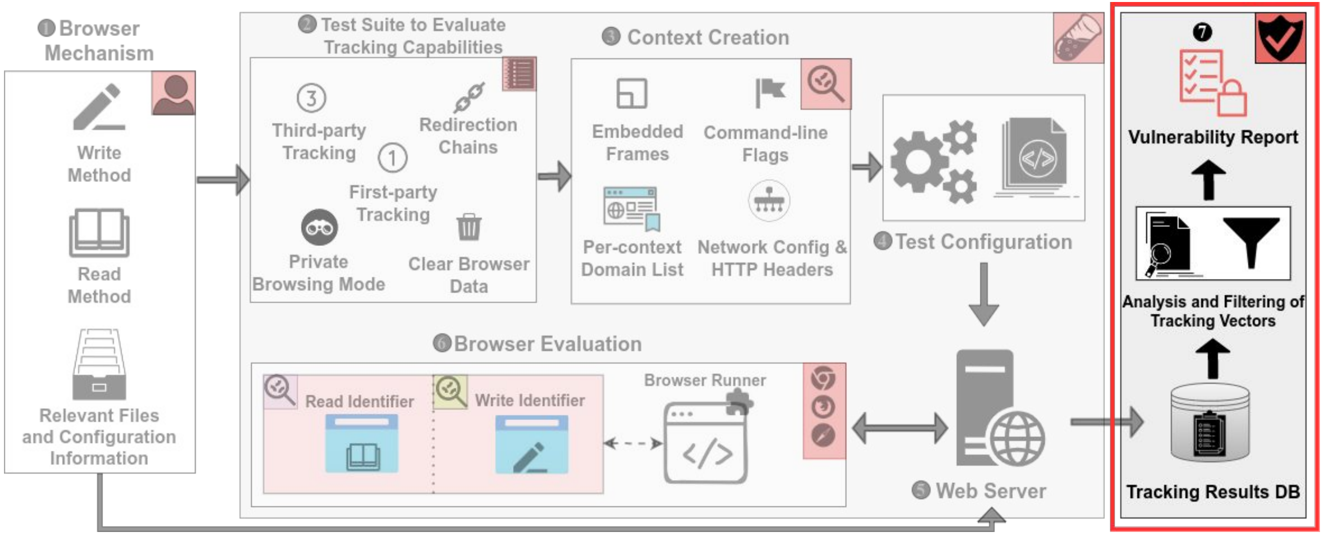
CanITrack: Browser Evaluation



6

- Script opens fresh browser instances
- Visits domains for specific tests
- Domains run **write()** and **read()** methods

CanITrack: Report



7

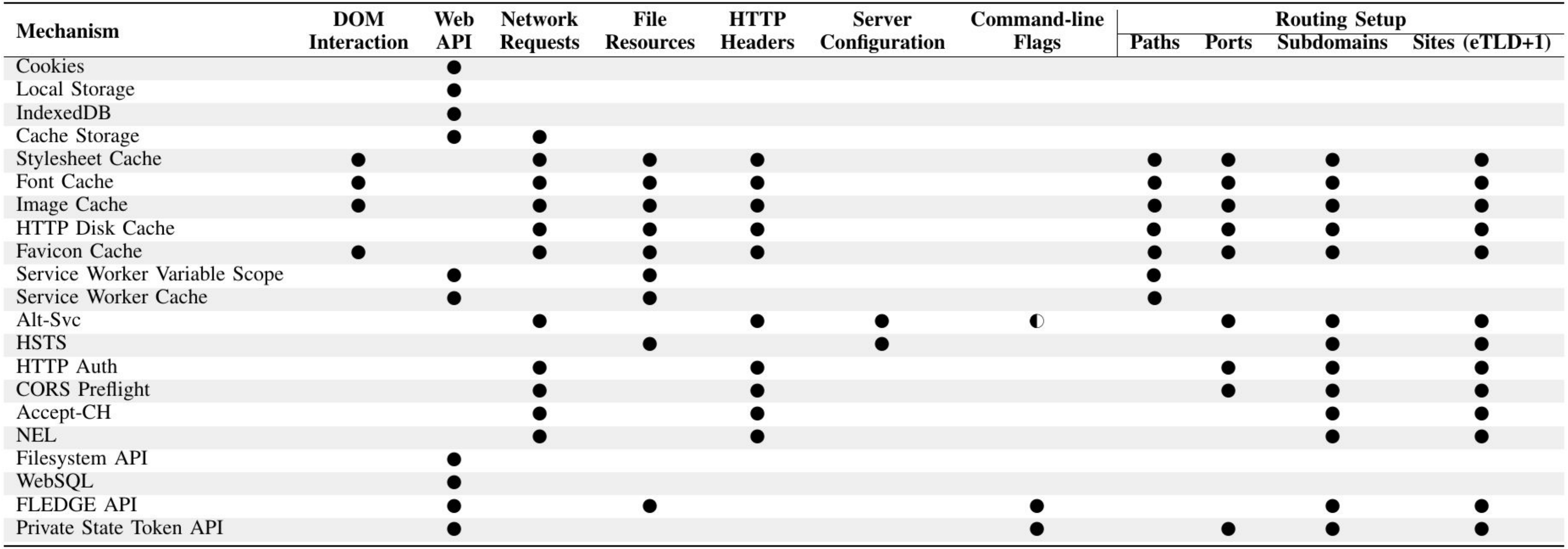
- Generates report
- Lists contexts under which tracking value persists

CanITrack: Plug & Play

- Mechanisms
  - Need **write()** & **read()** methods + relevant configuration
  - Support for new & existing mechanisms
- Tests
  - Based on known tracking vulnerabilities
  - Can be extended to add new tests

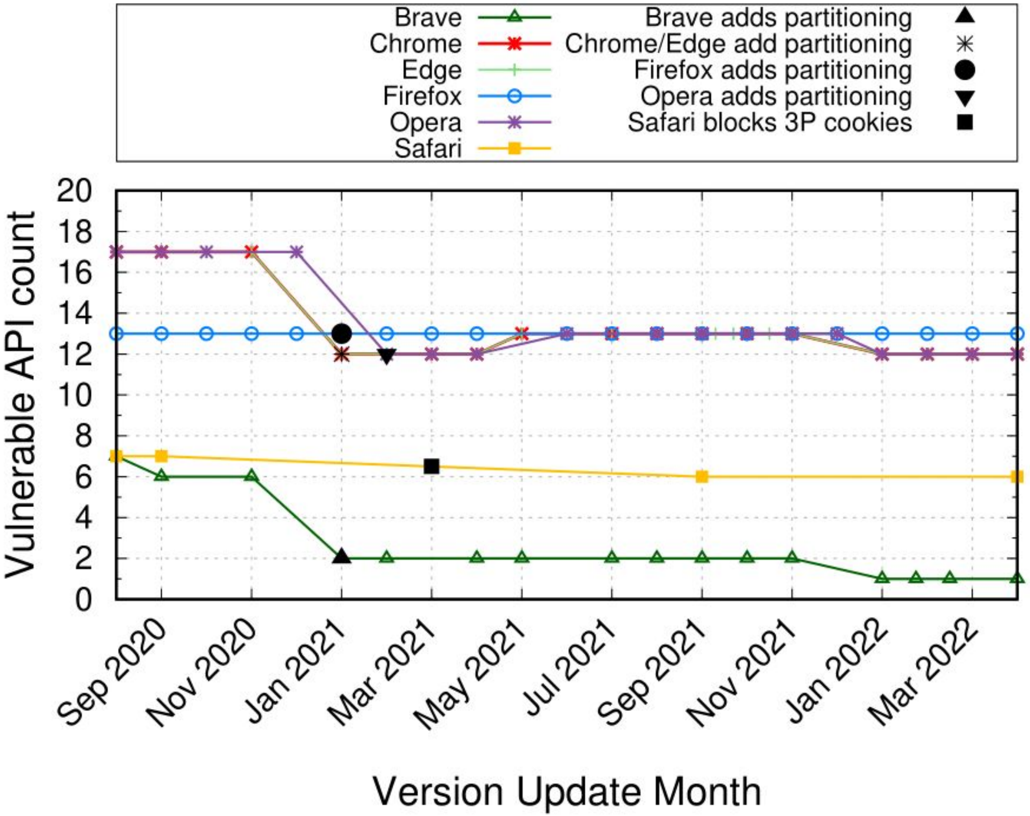
Evaluating Browser Mechanisms

- **21** browser mechanisms; **7** browsers; **2** years
- **4** new mechanisms:
  - Private State Tokens (formerly Trust Tokens)
  - FLEDGE API
  - CORS Preflight Cache
  - Client Hint Headers



Mechanism	DOM Interaction	Web API	Network Requests	File Resources	HTTP Headers	Server Configuration	Command-line Flags	Routing Setup			
								Paths	Ports	Subdomains	Sites (eTLD+1)
Cookies		●									
Local Storage		●									
IndexedDB		●									
Cache Storage		●	●								
Stylesheet Cache	●		●	●	●			●	●	●	●
Font Cache	●		●	●	●			●	●	●	●
Image Cache	●		●	●	●			●	●	●	●
HTTP Disk Cache			●	●	●			●	●	●	●
Favicon Cache	●		●	●	●			●	●	●	●
Service Worker Variable Scope		●		●				●			
Service Worker Cache		●		●				●			
Alt-Svc			●		●	●	◐		●	●	●
HSTS				●		●				●	●
HTTP Auth			●		●				●	●	●
CORS Preflight			●		●				●	●	●
Accept-CH			●		●					●	●
NEL			●		●					●	●
Filesystem API		●									
WebSQL		●									
FLEDGE API		●		●			●			●	●
Private State Token API		●					●		●	●	●

21 Evaluated Browser Mechanisms

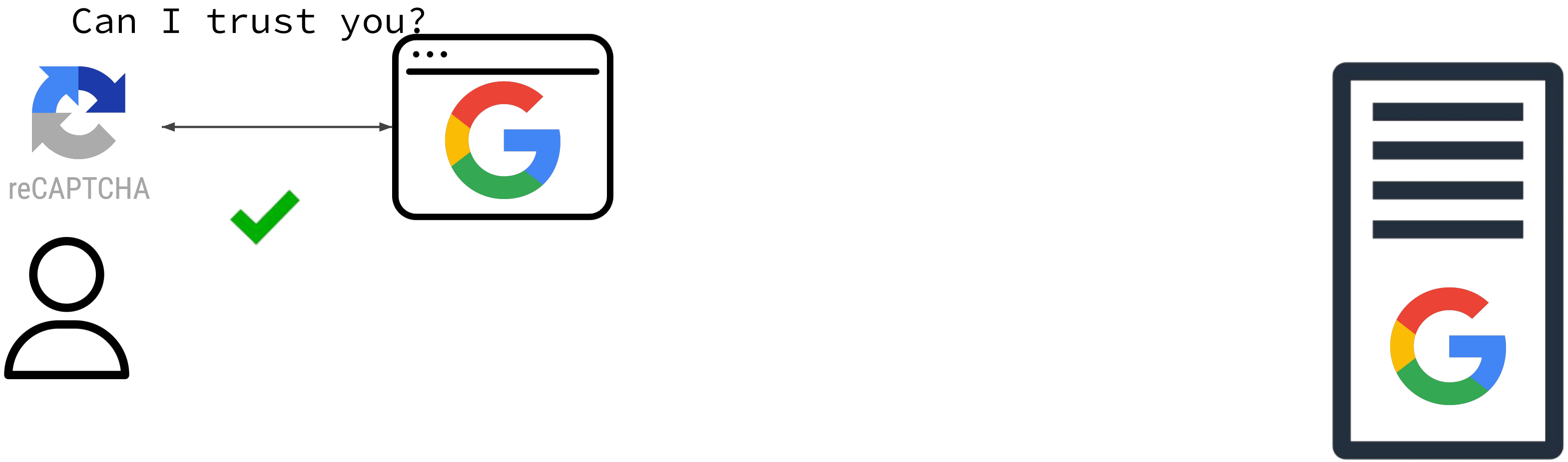


Evaluated Mechanisms as Potential 3P Tracking Vectors

Private State Tokens (formerly Trust Tokens)

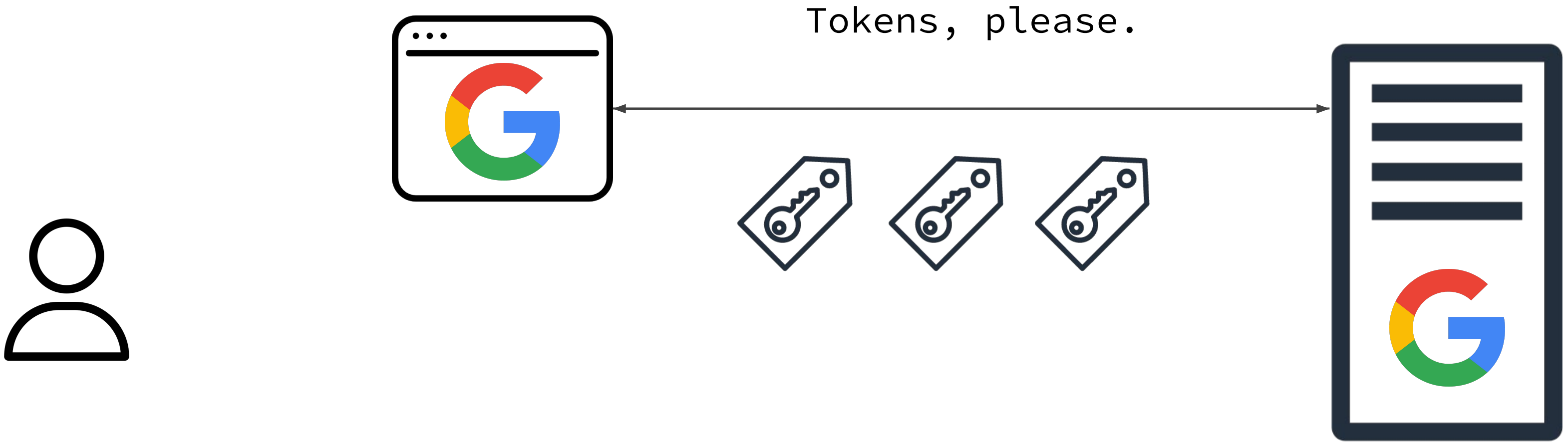
- One of Google’s Privacy Sandbox Proposals
  - Cross-site use cases without 3P cookies
- Communicates “Trust”:
  - Website 1 thinks I’m *trustworthy*
  - Website 2 can learn that Website 1 thinks I’m *trustworthy*

Private State Tokens: Workflow

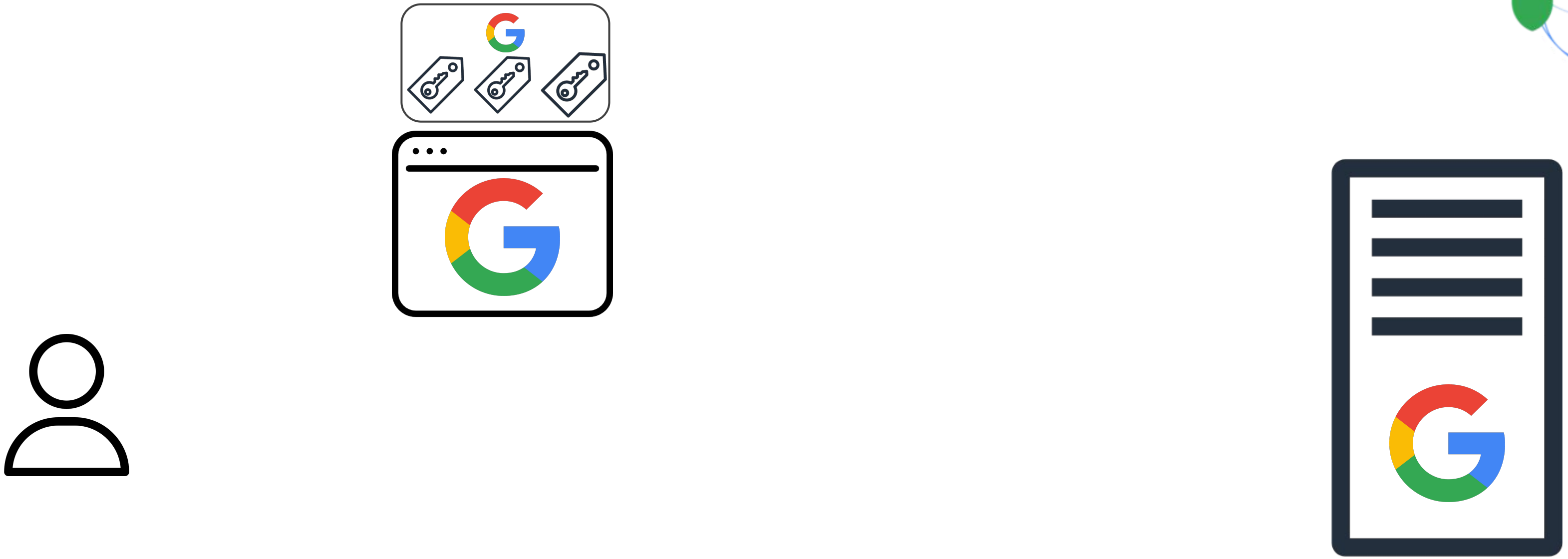


Can I trust you?

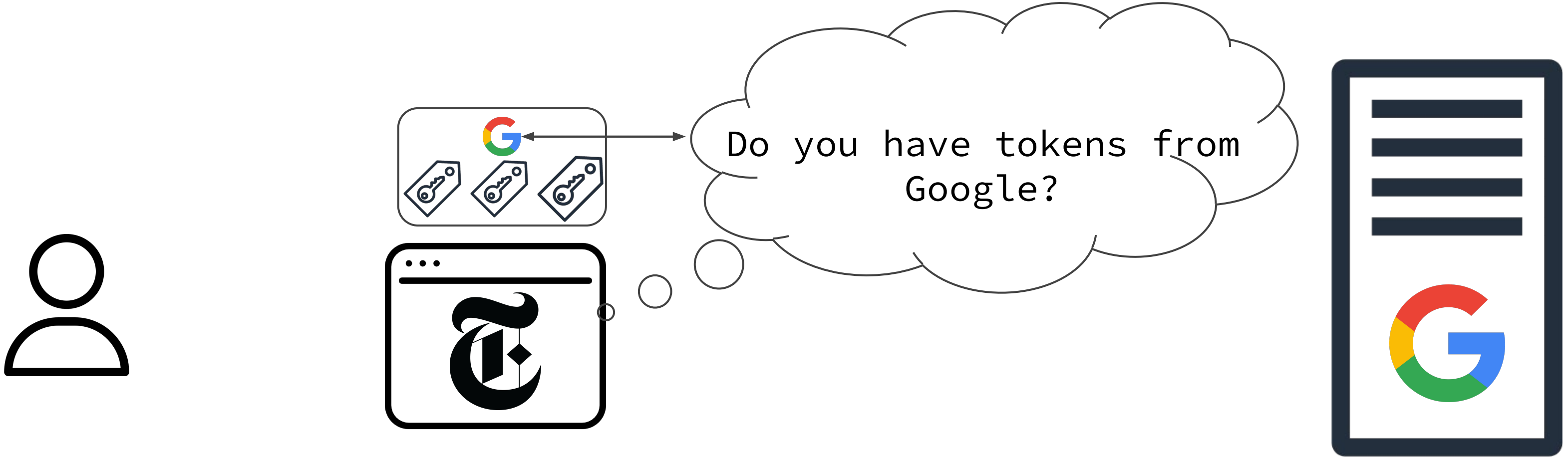
Private State Tokens: Workflow



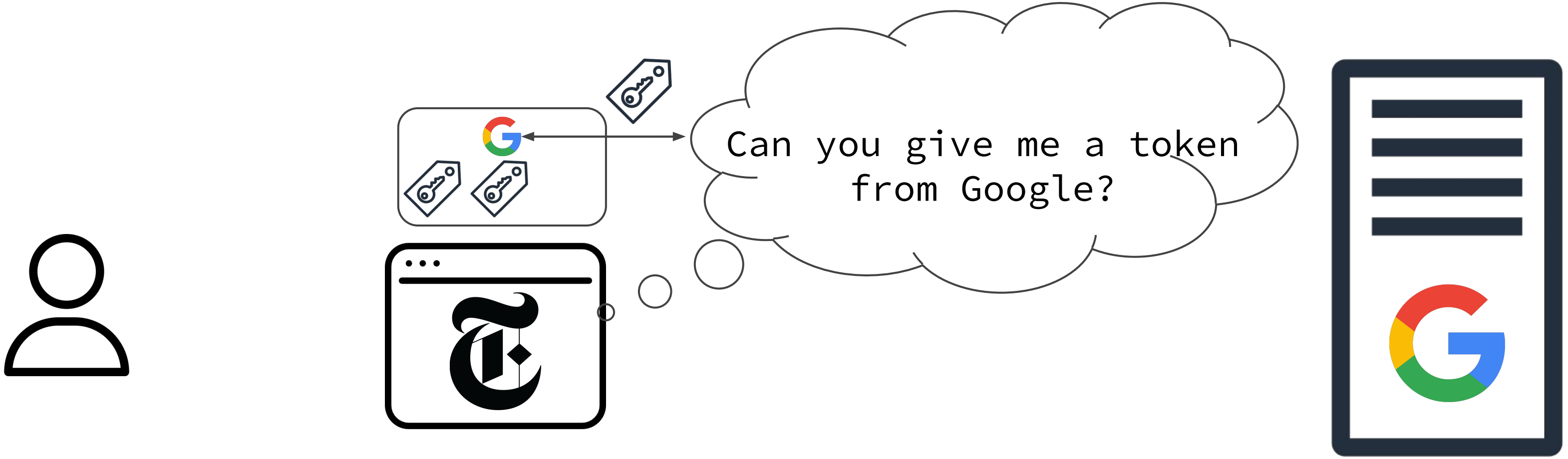
Private State Tokens: Workflow



Private State Tokens: Workflow

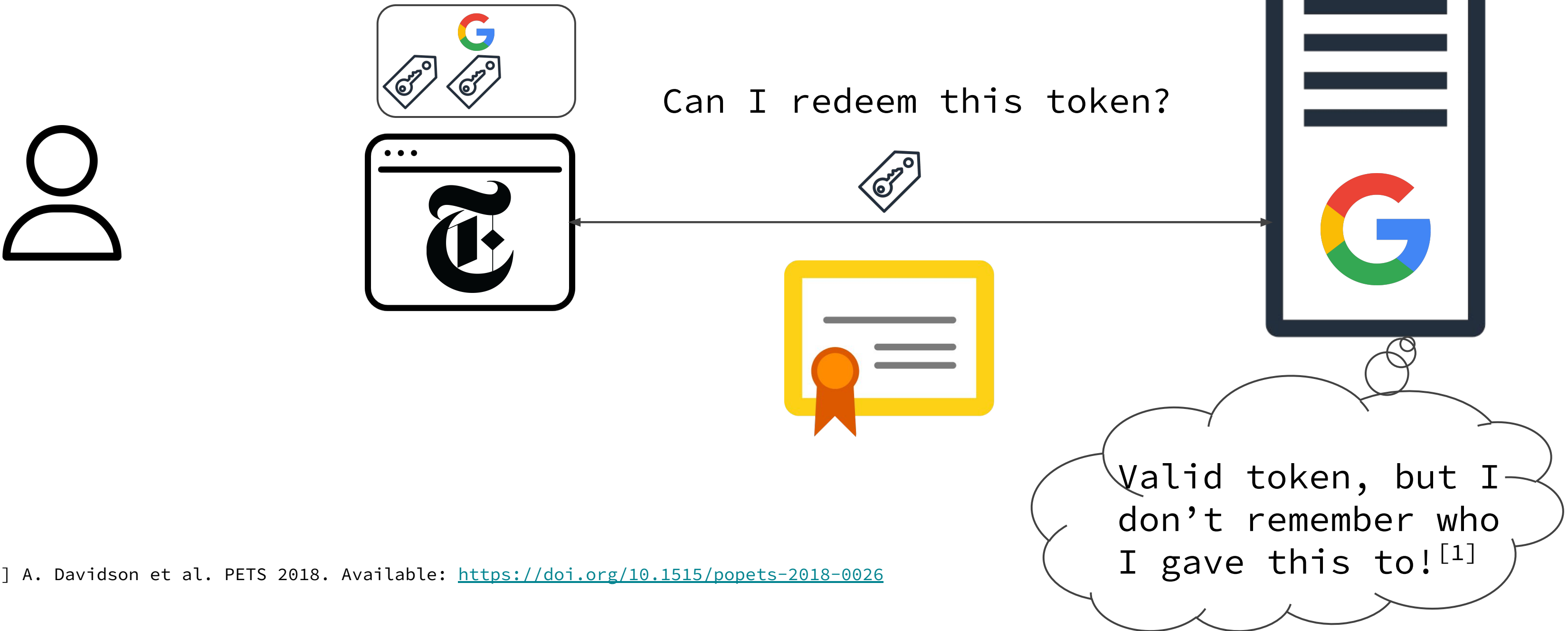


Private State Tokens: Workflow



Can you give me a token from Google?

Private State Tokens: Workflow

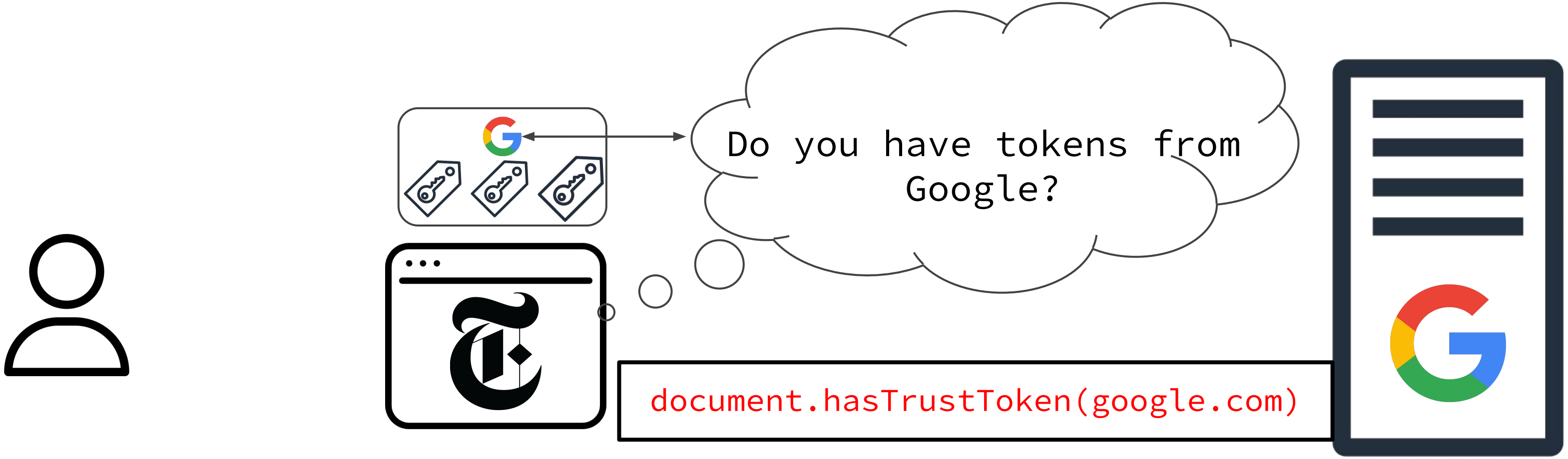


Can I redeem this token?

Valid token, but I don't remember who I gave this to![1]

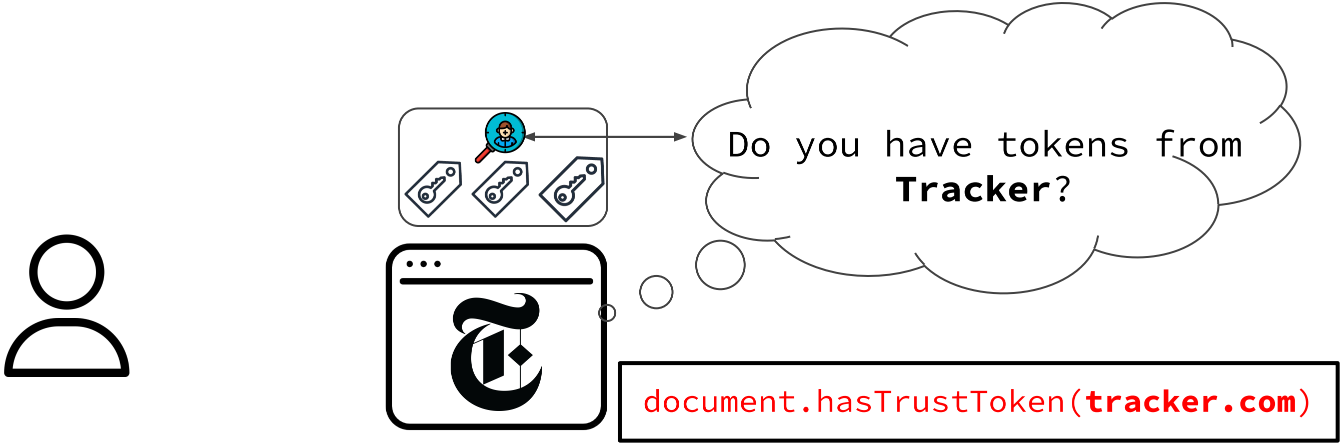
[1] A. Davidson et al. PETS 2018. Available: https://doi.org/10.1515/popets-2018-0026

Private State Tokens: The Problem


Do you have tokens from Google?
document.hasTrustToken(google.com)

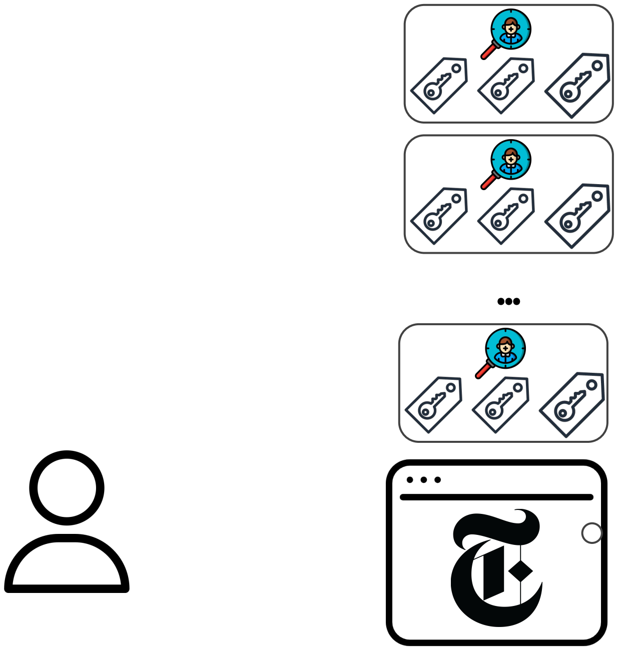
- No redemption → no requests to Google -> Google learns nothing
- NYTimes learned that Google gave me a token

Private State Tokens: The Problem



Do you have tokens from **Tracker**?

```
document.hasTrustToken(tracker.com)
```



Do they have tokens from
Tracker1?
Tracker2?
…
Tracker32?

```
document.hasTrustToken(tracker1.com)
document.hasTrustToken(tracker2.com)
…
document.hasTrustToken(tracker32.com)
```

True = **1**
False = **0**
…
True = **1**

User ID = 10100…001

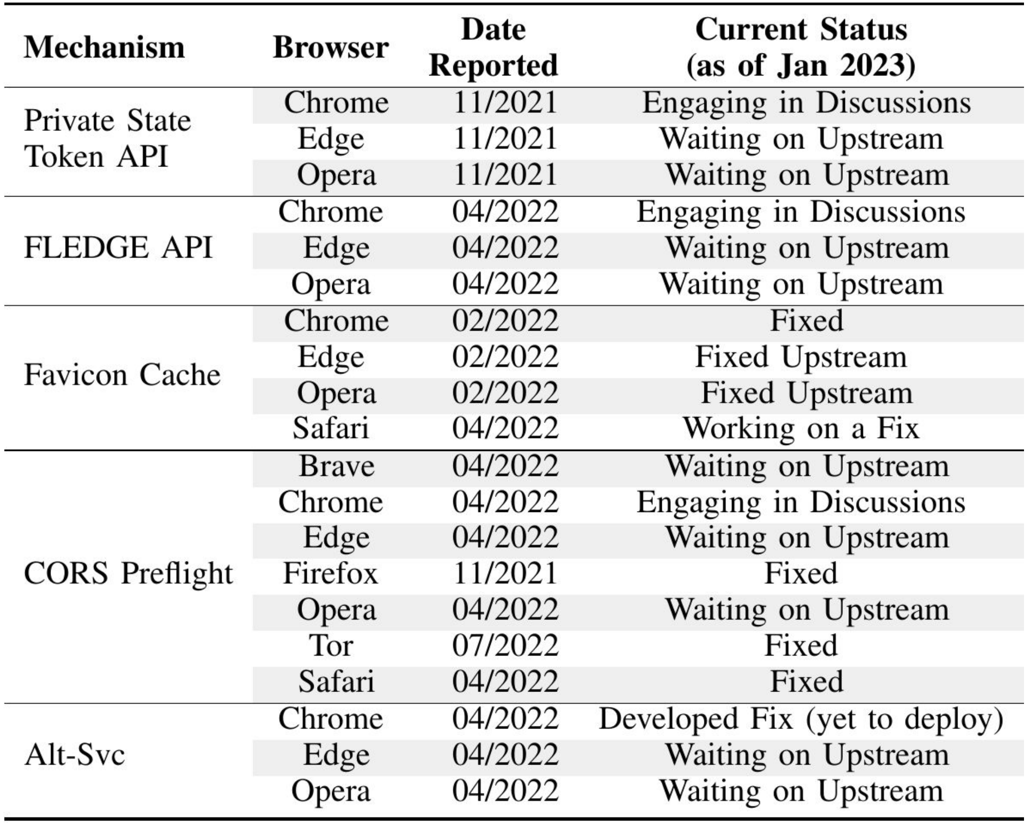
Private State Tokens: CanITrack

- write()
    - Issue tokens
- read()
    - Do you have tokens?
- CanITrack Report:
    - Tracking in Private Mode? **No.**
    - 3P Tracking? **Yes.**
    - Expand ID w/ Redirections? **Yes.**

```
write (uniqueID, domainList, i) {
    if(uniqueID[i] == '1') {
        fetch(`https://${domainList[i]}/tokens`, {
                method: "POST",
                trustToken: {type: "token-request"}}}
```

```
read (domainList, i) {
    return await
document.hasTrustToken(`https://${domainList[i]}`);}
```

Disclosures



Mechanism	Browser	Date Reported	Current Status (as of Jan 2023)
Private State Token API	Chrome	11/2021	Engaging in Discussions
	Edge	11/2021	Waiting on Upstream
	Opera	11/2021	Waiting on Upstream
FLEDGE API	Chrome	04/2022	Engaging in Discussions
	Edge	04/2022	Waiting on Upstream
	Opera	04/2022	Waiting on Upstream
Favicon Cache	Chrome	02/2022	Fixed
	Edge	02/2022	Fixed Upstream
	Opera	02/2022	Fixed Upstream
	Safari	04/2022	Working on a Fix
CORS Preflight	Brave	04/2022	Waiting on Upstream
	Chrome	04/2022	Engaging in Discussions
	Edge	04/2022	Waiting on Upstream
	Firefox	11/2021	Fixed
	Opera	04/2022	Waiting on Upstream
	Tor	07/2022	Fixed
	Safari	04/2022	Fixed
Alt-Svc	Chrome	04/2022	Developed Fix (yet to deploy)
	Edge	04/2022	Waiting on Upstream
	Opera	04/2022	Waiting on Upstream

Conclusion

- Framework to streamline testing:
  - Open source; **21** browser mechanisms
  - Support browser vendors & researchers
  - New mechanisms can be added in as little as 30 minutes (depending on mechanism complexity)
- **4** new tracking vectors
  - **2** from Google's Privacy Sandbox
- Reported vulnerabilities
  - **20** disclosure reports across **7** browser vendors

Code: https://github.com/masood/canitrack

Email: mali92@uic.edu