UCognito: Private Browsing without Tears

Meng Xu, Yeongjin Yang, Xinyu Xing, Taesoo Kim, Wenke Lee

Georgia Institute of Technology

Private Browsing Mode

Private Browsing

Incognito Mode
Guest Mode

InPrivate

Private Window

Private Browsing Mode

Private Browsing

Incognito Mode
Guest Mode

InPrivate

Private Window

Questions:

- Same ?
- Expected ?
- Implemented ?
- Private ?

Private Browsing Mode

Private Browsing

Incognito Mode
Guest Mode

InPrivate

Private Window

Questions:

- Same ?
- Expected ?
- Implemented ?
- Private ?

Problem:
Different Definitions of Private Browsing

Use of persistent data in private browsing mode	**Firefox**	**Chrome Incognito**	**Opera**	**Safari**	**IE**
Download entries	✘	✔	✔	✔	✘
SSL self-signed certificate	✔	✘	✘	✔	✘
Add-on enabled by default	✔	✘	✘	✔	✔

Problem:
Different Definitions of Private Browsing

Category	Use		Store	
	Incognito	Guest	Incognito	Guest
Browsing history	✔	✘	✘	✘
Cookies	✘	✘	✘	✘
Cache	✘	✘	✘	✘
Local storage	✘	✘	✘	✘
Flash storage	✘	✘	✘	✘
Download entries	✔	✘	✘	✘
Autofills	✔	✘	✘	✘
Bookmarks	✔	✘	✔	✘
Per-site zoom	✔	✘	✘	✘
Per-site permission	✔	✘	✘	✘
SSL self-signed cert	✘	✘	✘	✘
SSL client cert	✔	✔	✔	✔
Add-on storage	✔	✘	✔	✘

Private Browsing Mode

Private Browsing

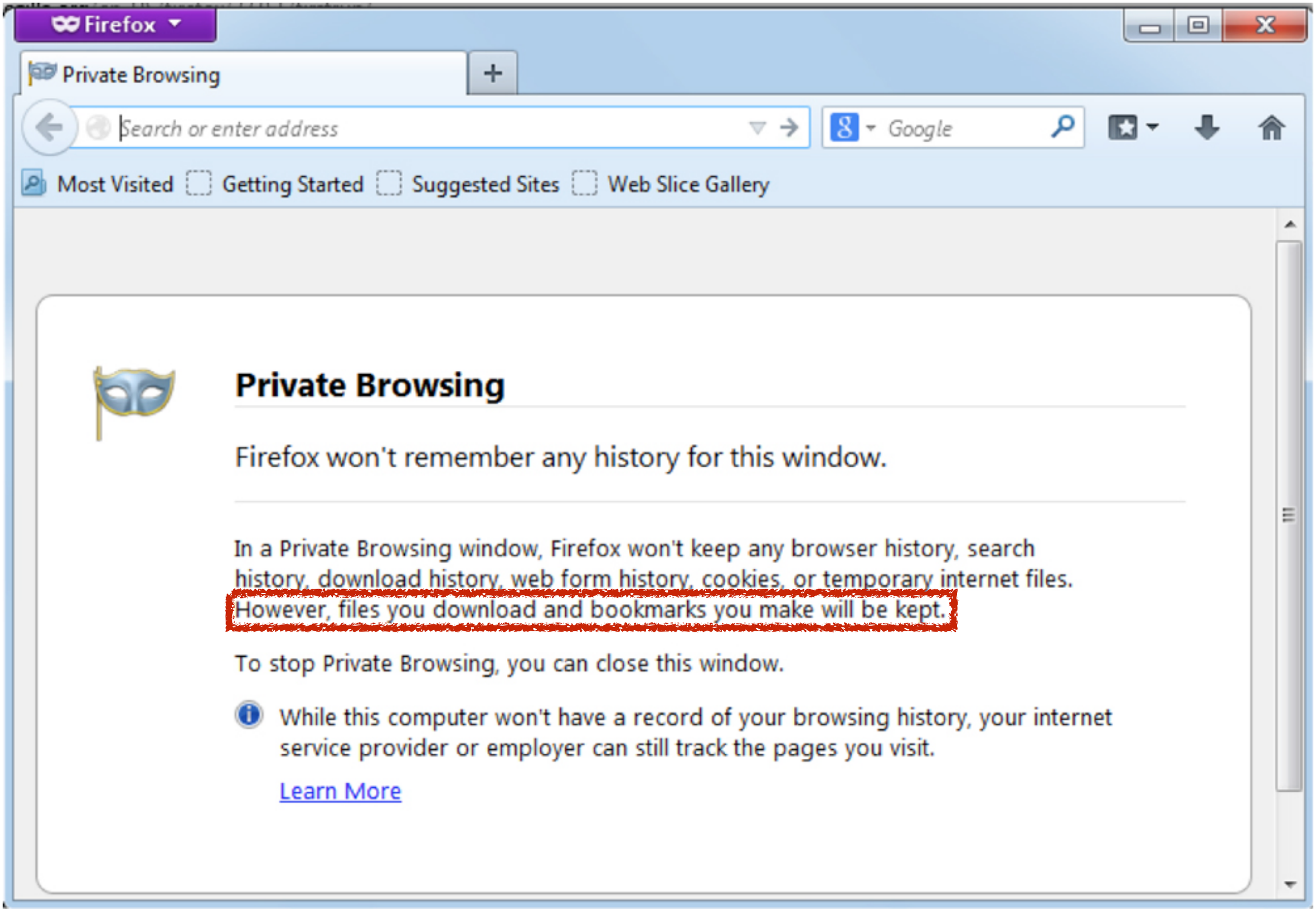
Incognito Mode
Guest Mode

InPrivate

Private Window

Questions:

- Same ?
- Expected ?
- Implemented ?
- Private ?


Firefox
Private Browsing
Search or enter address
Google
Most Visited
Getting Started
Suggested Sites
Web Slice Gallery
Private Browsing
Firefox won't remember any history for this window.
In a Private Browsing window, Firefox won't keep any browser history, search history, download history, web form history, cookies, or temporary internet files. However, files you download and bookmarks you make will be kept.
To stop Private Browsing, you can close this window.
While this computer won't have a record of your browsing history, your internet service provider or employer can still track the pages you visit.
Learn More

Private Browsing Mode

Private Browsing

Incognito Mode
Guest Mode

InPrivate

Private Window

Questions:

- Same ?
- Expected ?
- Implemented ?
- Private ?

Implementation is `mimicking` and ad-hoc

```
1  // @netwerk/cookie/nsCookieService.cpp
2  DBState *mDBState;
3  nsRefPtr<DBState> mDefaultDBState; // DB for normal mode
4  nsRefPtr<DBState> mPrivateDBState; // DB for private mode
5
6  // invoked when initializing session
7  void nsCookieService::InitDBStates() {
8    ...
9    mDefaultDBState = new DBState(); // DB for normal mode
10   mPrivateDBState = new DBState(); // DB for private mode
11   // default: normal mode
12   mDBState = mDefaultDBState;
13   ...
14 }
15
16 // invoked when storing cookies
17 void nsCookieService::SetCookieStringInternal() {
18   ...
19   // decide which cookie DB to use, depending on the mode
20   mDBState = aIsPrivate ? mPrivateDBState : mDefaultDBState;
21   ...
22 }
```

Implementation is `mimicking` and ad-hoc

```
1  // @netwerk/cookie/nsCookieService.cpp
2  DBState *mDBState;
3  nsRefPtr<DBState> mDefaultDBState; // DB for normal mode
4  nsRefPtr<DBState> mPrivateDBState; // DB for private mode
5
6  // invoked when initializing session
7  void nsCookieService::InitDBStates() {
8    ...
9    mDefaultDBState = new DBState(); // DB for normal mode
10   mPrivateDBState = new DBState(); // DB for private mode
11   // default: normal mode
12   mDBState = mDefaultDBState;
13   ...
14 }
15
16 // invoked when storing cookies
17 void nsCookieService::SetCookieStringInternal() {
18   ...
19   // decide which cookie DB to use, depending on the mode
20   mDBState = aIsPrivate ? mPrivateDBState : mDefaultDBState;
21   ...
22 }
```

Problem:
Code complexity grows exponentially

- How many duplications ?
  - cookie, history, cache, download entries, autofills, bookmarks, flash storage …
  - per-site permission, per-site zoom level, SSL certs …
  - html5 local storage, indexDB …

Problem:
Code complexity grows exponentially

- How many duplications ?
    - cookie, history, cache, download entries, autofills, bookmarks, flash storage …
    - per-site permission, per-site zoom level, SSL certs …
    - html5 local storage, indexDB …

X 3 !!!

Normal mode
Incognito mode
Guest mode

Problem:
Lack of elegant support for add-ons

```
1 // 1. Detecting private browsing mode @MDN
2 Components.utils.import(
3     "resource://gre/modules/PrivateBrowsingUtils.jsm");
4 if (!PrivateBrowsingUtils.isWindowPrivate(window)) {
5   ...
6 }
7
8 // 2. Detecting mode changes @MDN
9 function pbObserver() { /* clear private data */ }
10 var os = Components.classes["@mozilla.org/observer-service;1"]
11            .getService(Components.interfaces.nsIObserverService);
12 os.addObserver(pbObserver, "last-pb-context-exited", false);
```

Warning: Google Chrome cannot prevent extensions from recording your browsing history. To disable this extension in incognito mode, unselect this option.

Private Browsing Mode

Private Browsing

Incognito Mode
Guest Mode

InPrivate

Private Window

Questions:

- Same ?
- Expected ?
- Implemented ?
- Private ?

New Tab

Translate

You've gone incognito. Pages you view in this window won't appear in your browser history or search history, and they won't leave other traces, like cookies, on your computer after you close **all** open incognito windows. Any files you download or bookmarks you create will be preserved, however.

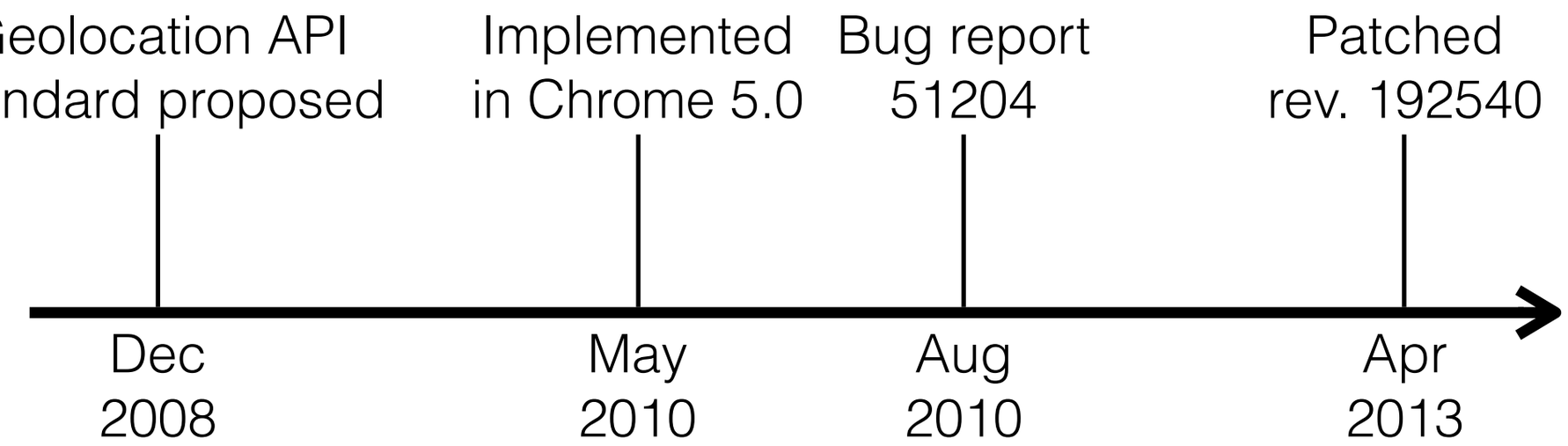
Going incognito doesn't affect the behavior of other people, servers, or software. Be wary of:

- Websites that collect or share information about you
- Internet service providers or employers that track the pages you visit
- Malicious software that tracks your keystrokes in exchange for free smileys
- Surveillance by secret agents
- People standing behind you

Learn more about incognito browsing.

Because Google Chrome does not control how extensions handle your personal data, all extensions have been disabled for incognito windows. You can reenable them individually in the extensions manager.

Per-site permission reveals browsing history



PNaCl cache reveals browsing history

- PNaCl translation cache reveals whether you have previously visited a website.

- http://gonativeclient.appspot.com/demo/lua (demo)

Problem: Not secure by default

- How many places to instrument ?
  - cookie, history, cache, download entries, autofills, bookmarks, flash storage …

ok, these are common

  - per-site permission, per-site zoom level, SSL certs …

hmm, we can think of these

Problem: Not secure by default

- How many places to instrument ?
  - html5 local storage, indexDB …

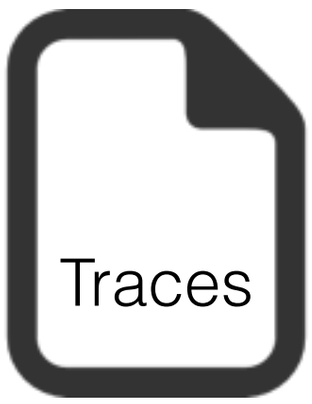
new features are coming in!

- PNaCl, OCSPResponse …

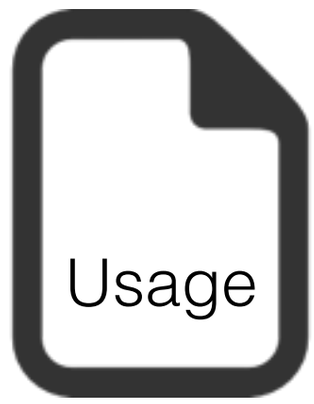
oh I forgot them!

Uverifier: Testing Private Browsing Mode

```
open(<file>, “w”)
……
write(<file>, ……)
……
no delete(<file>)
```




```
open(<file>, “r”)
……
read(<file>, ……)
```



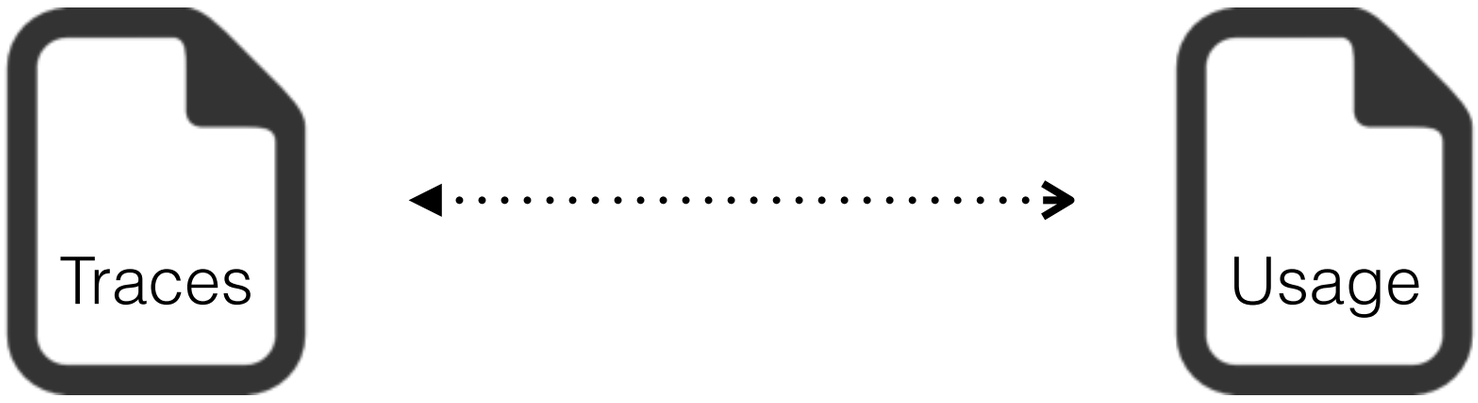
PNaCl cache explanation

Normal mode

```
open(<file>, “w”)
……
write(<file>, ……)
……
no delete(<file>)
```

Private mode

```
open(<file>, “r”)
……
read(<file>, ……)
```



<profile>/PnaclTranslationCache/index
<profile>/PnaclTranslationCache/data_1
<profile>/PnaclTranslationCache/data_2
<profile>/PnaclTranslationCache/data_3

UCognito: Decouple private mode implementation from browser codebase.

Private Browsing

Incognito Mode
Guest Mode

InPrivate

Private Window

UCognito: Decouple private mode implementation from browser codebase.

UCognito Mode

UCognito: Decouple private mode implementation from browser codebase.

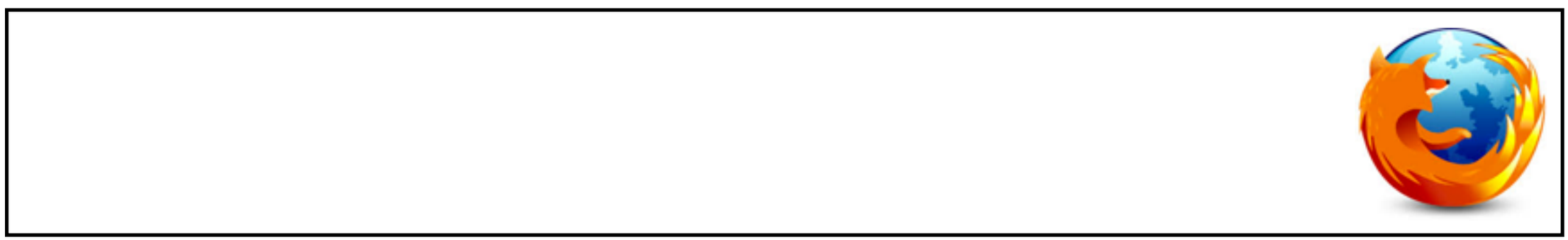
UCognito Mode

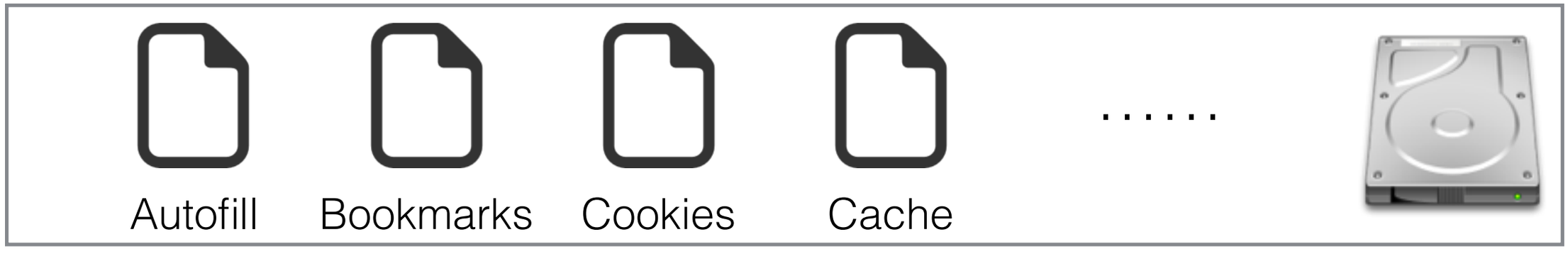
Questions:

- Same ?
- Expected ?
- Implemented ?
- Private ?

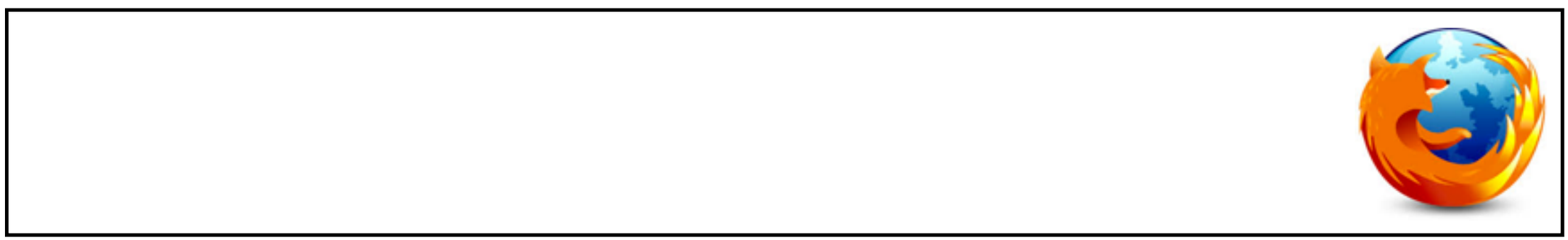
UCognito Architecture

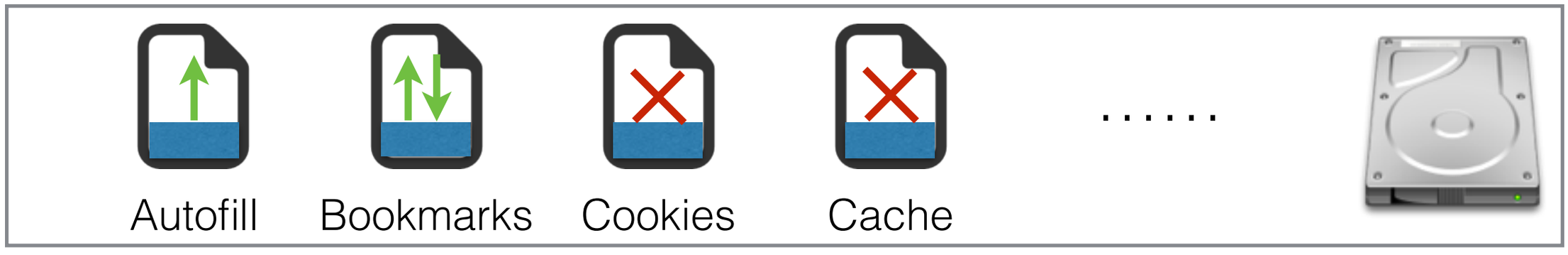
Step 0: Specify Policies



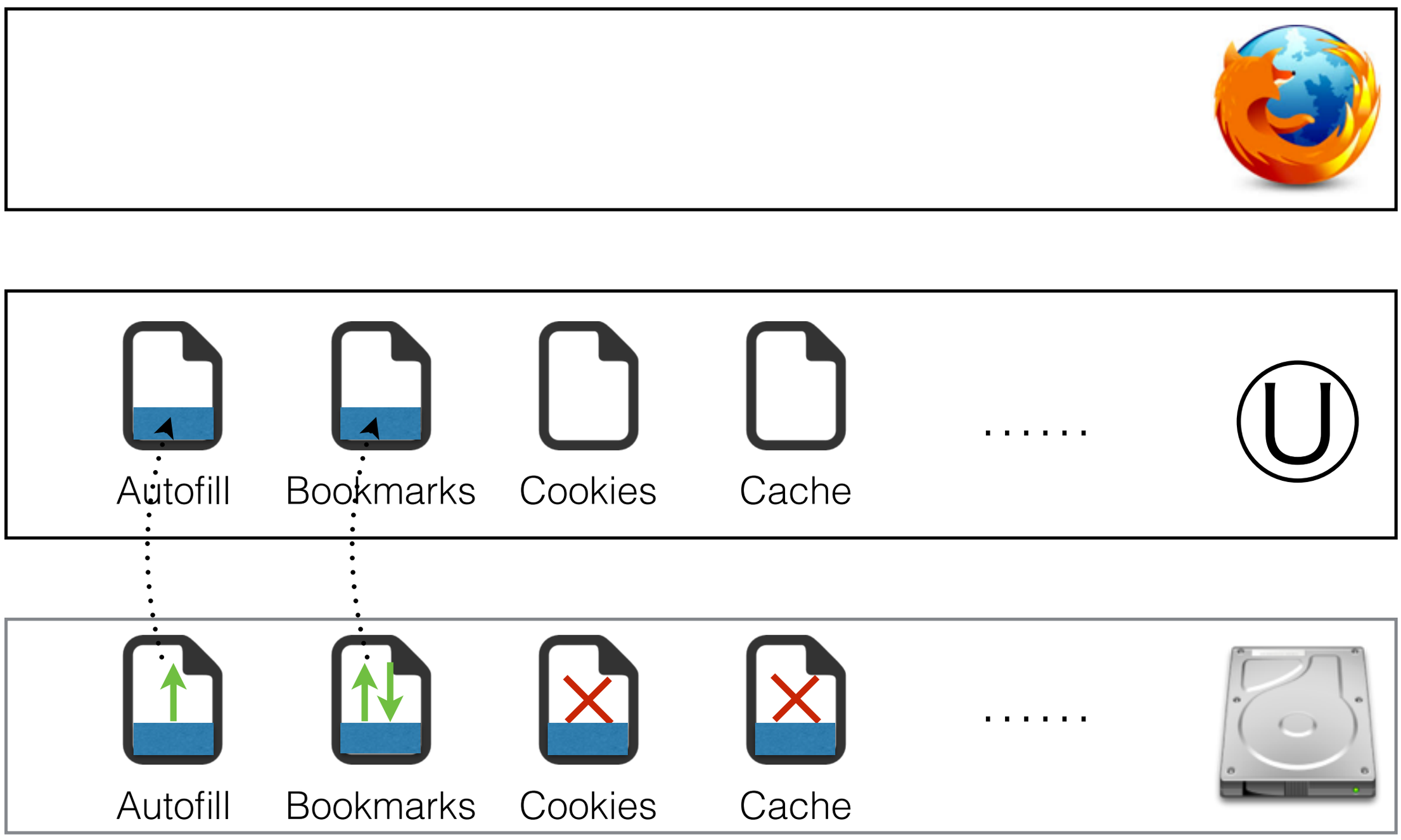


Step 0: Specify Policies

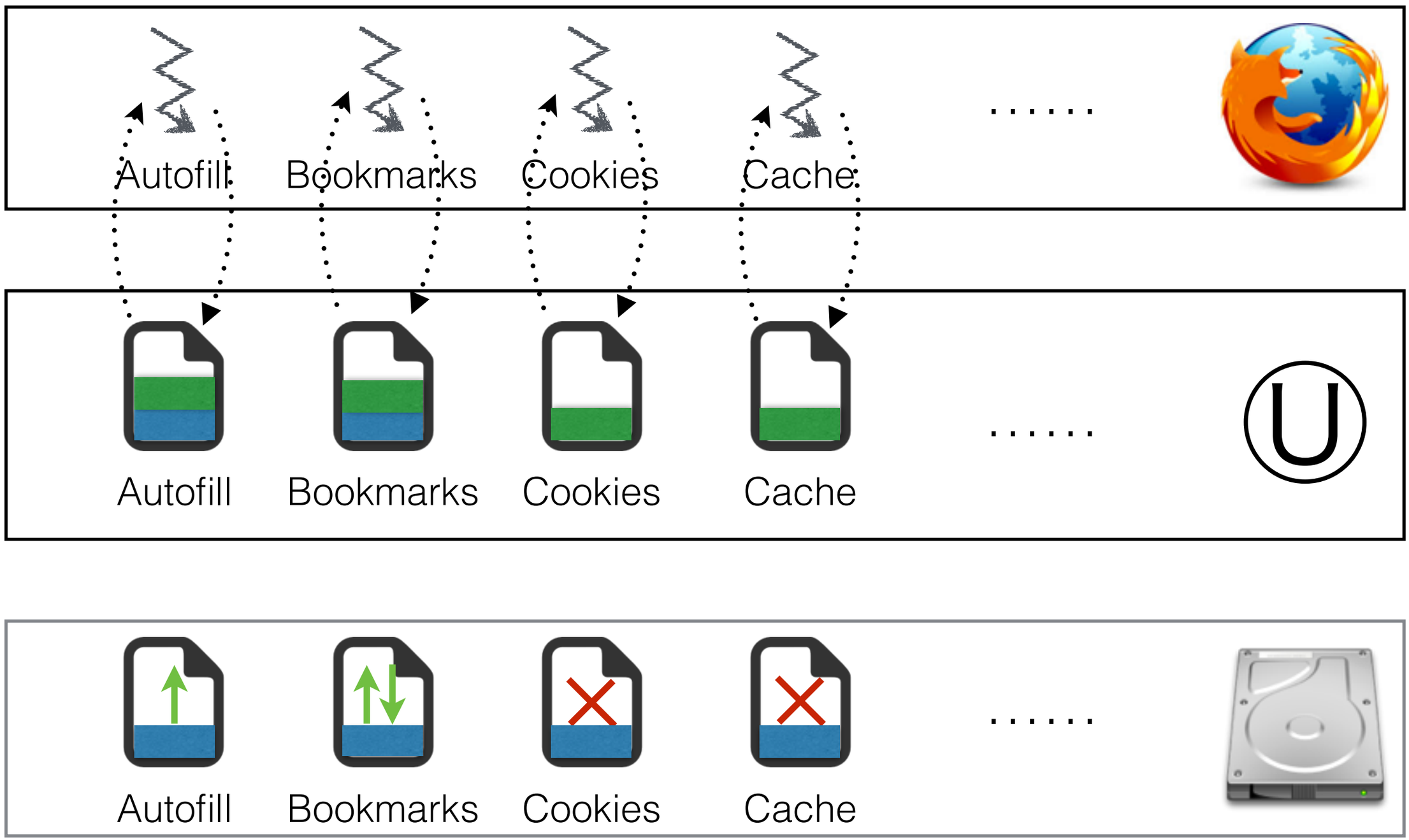




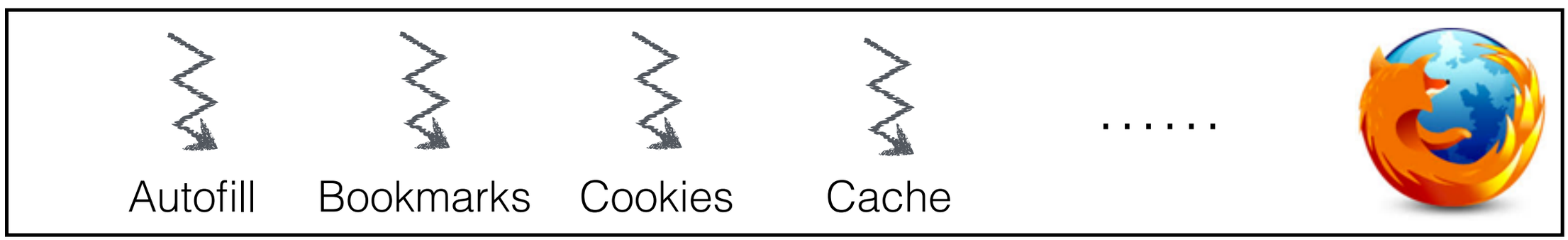
Step 1: Starting UCognito

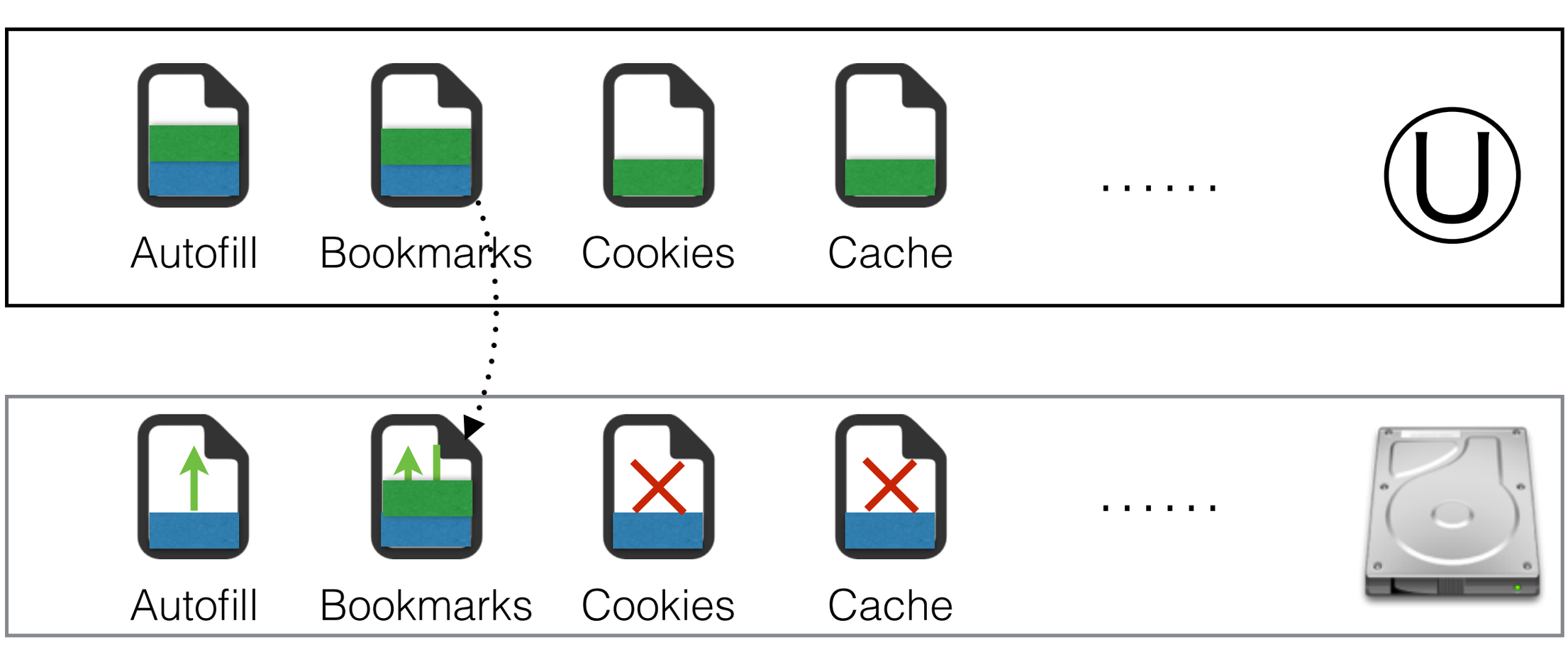


Step 2: Browsing



Step 3: Cleaning





UCognito Sandbox

- Goal: redirecting all path to a contained location
  - e.g., /home/user/profile/* → /tmp/<pid>/home/user/profile/*
- Implementation: *seccomp-bpf*
  - Leverage MBox, a lightweight sandboxing for non-root users
  - Place hooks on 50 system calls that deals with file paths, e.g., *open, creat, unlink*, stat etc

UCognito Policy System

- Goal: control trace storage and trace usage on a per-file granularity

- Design:

- **CLEAN**: disallow loading of any traces, run browser at its pristine stage

- **COPY**: allow use of existing traces, carrying existing information to the sandbox

- **WRITE**: allow storing of new traces, committing data in sandbox back to file system

Default Policy

```
1 # exclude all files in home directory
2 [clean]
3 ~/
```

Whitelist principle:

By default, nothing is allowed
to be stored or used
unless specified in a policy

Category	Use	Store
Browsing history	✘	✘
Cookies	✘	✘
Cache	✘	✘
Local storage	✘	✘
Flash storage	✘	✘
Download entries	✘	✘
Autofills	✘	✘
Bookmarks	✘	✘
Per-site zoom	✘	✘
Per-site permission	✘	✘
SSL self-signed cert	✘	✘
SSL client cert	✘	✘
Add-on storage	✘	✘
(All others)	✘	✘

Chrome Guest Mode

```
# exclude all files in home directory
[clean]
~/

# Use: SSL client certificates
[copy]
~/.pki/nssdb/cert9.db

# write-back client certificates
[write]
~/.pki/nssdb/cert9.db
```

Category	Use	Store
Browsing history	✘	✘
Cookies	✘	✘
Cache	✘	✘
Local storage	✘	✘
Flash storage	✘	✘
Download entries	✘	✘
Autofills	✘	✘
Bookmarks	✘	✘
Per-site zoom	✘	✘
Per-site permission	✘	✘
SSL self-signed cert	✘	✘
SSL client cert	✔	✔
Add-on storage	✘	✘
(All others)	✘	✘

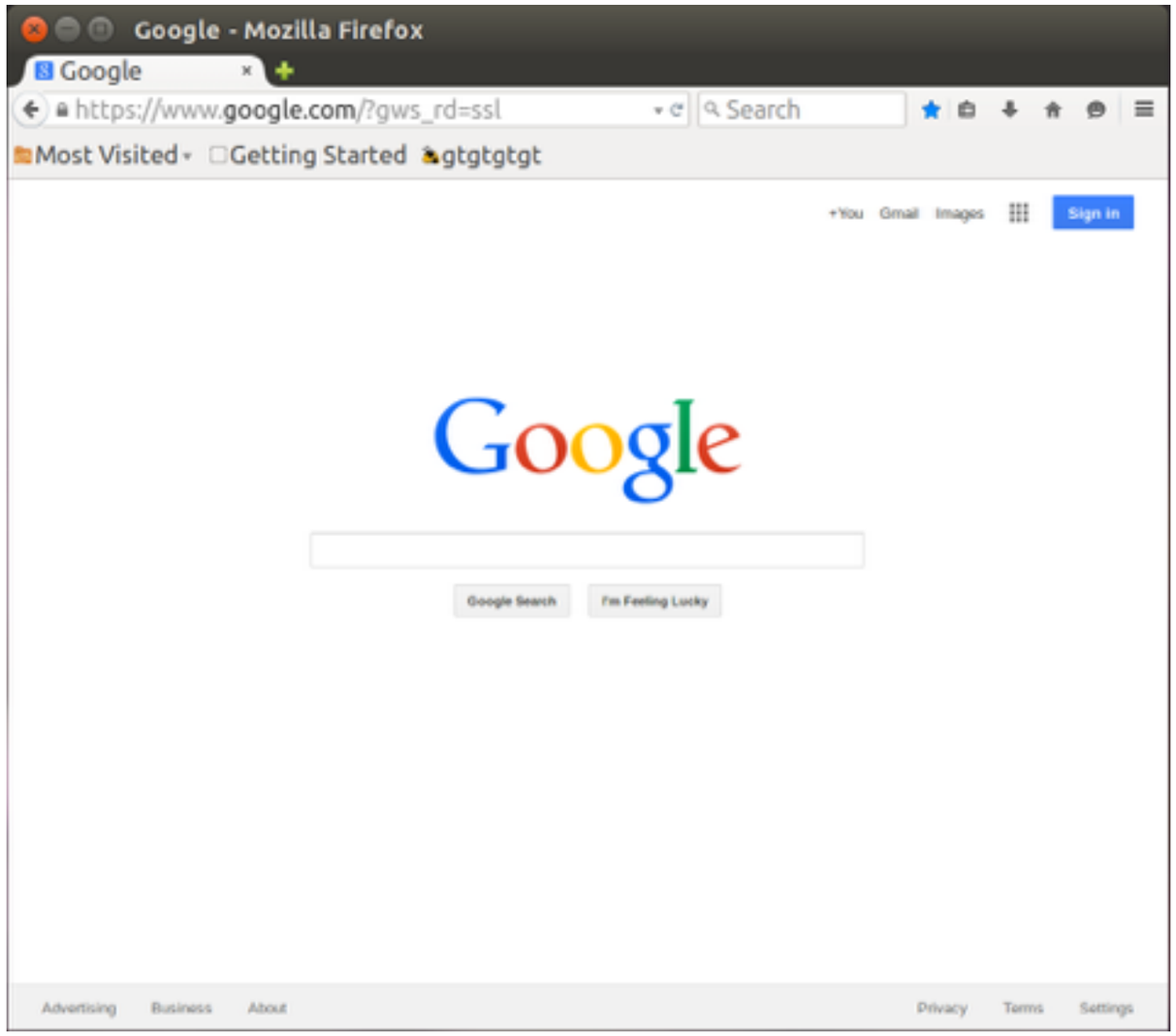
Chrome Incognito Mode

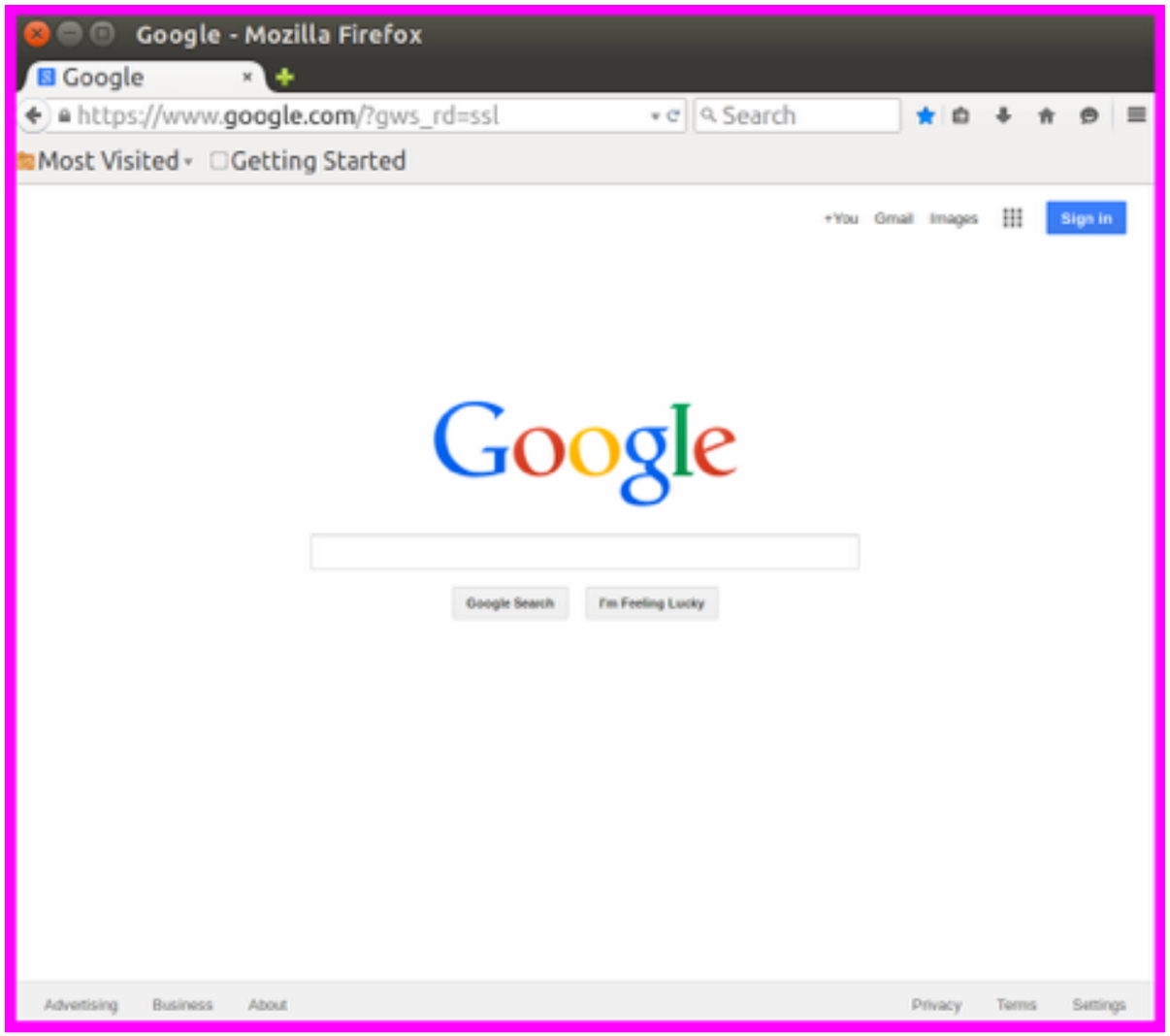
```
# copy section: copying files from the user profiles
[copy]
# Use: browsing history
~/.config/google-chrome/Default/History
~/.config/google-chrome/Default/History-journal
~/.config/google-chrome/Default/Visited Links
~/.config/google-chrome/Default/Favicons
~/.config/google-chrome/Default/Favicons-journal
~/.config/google-chrome/Default/Top Sites
~/.config/google-chrome/Default/Top Sites-journal

# Use: autofill data
~/.config/google-chrome/Default/Login Data
~/.config/google-chrome/Default/Login Data-journal
~/.config/google-chrome/Default/Web Data
~/.config/google-chrome/Default/Web Data-journal

# Use: per-site preferences
~/.config/google-chrome/Default/Preferences
~/.config/google-chrome/Default/Secure Preferences

# Use: SSL certificates
~/.config/google-chrome/Default/TransportSecurity
~/.config/google-chrome/Default/Origin Bound Certs
~/.config/google-chrome/Default/Origin Bound Certs-journal

# Use: SSL client certificates
~/.pki/nssdb/cert9.db

# Use: bookmarks
~/.config/google-chrome/Default/Bookmarks

# Use: extension storage
~/.config/google-chrome/Default/Local Extension Settings/

# clean section: exclude files & sub-directories
[clean]
# exclude all other files in the home directory
~/

# write section: write-back data to the user profile
[write]
# write-back bookmarks
~/.config/google-chrome/Default/Bookmarks
# write-back client certificates
~/.pki/nssdb/cert9.db
# write-back extension storages
~/.config/google-chrome/Default/Local Extension Settings/
```

Category	Use	Store
Browsing history	✔	✘
Cookies	✘	✘
Cache	✘	✘
Local storage	✘	✘
Flash storage	✘	✘
Download entries	✔	✘
Autofills	✔	✘
Bookmarks	✔	✔
Per-site zoom	✔	✘
Per-site permission	✔	✘
SSL self-signed cert	✘	✘
SSL client cert	✔	✔
Add-on storage	✔	✔
(All others)	✘	✘

UI and UX

$ ucognito -P chrome_incognito.cfg -- google-chrome





Preventing privacy violation cases

- UCognito is able to prevent all the cases in our paper
- UCognito provides natural support to add-ons

Add-on	# Users	Incognito	UCognito
Session Buddy	373409	history, cache, cookies, etc	✘
StayFocusd	600944	Sync Extension Settings	✘
Better History	248112	Extension State	✘
Lazarus Form Recovery	125709	Extension DB	✘

Performance overhead on Javascript benchmarks

	Firefox		Chrome	
Add-on	**Base**	**UCognito**	**Base**	**UCognito**
Karken (ms)	1171.1	1171.2 (0.0%)	1108.6	1115.2 (0.6%)
Sun spider (ms)	158.3	159.8 (0.9%)	173.1	177.4 (2.5%)
Octane (pts)	27164	27013 (-0.6%)	27266	27018 (-0.9%)

- Only hook system calls that deals with file paths
- Not hooking *read, write, send, recv* which are very frequently called in networked applications

Performance overhead on real websites

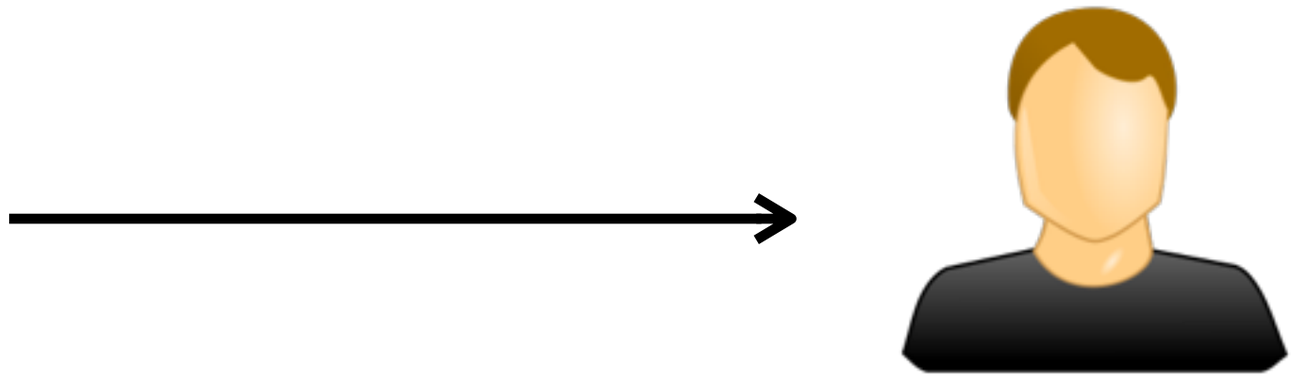
	Firefox		Chrome	
Website (ms)	**Base**	**UCognito**	**Base**	**UCognito**
google.com	277	280 (0.79%)	193	196 (1.55%)
bing.com	208	208 (0.29%)	190	193 (1.58%)
twitter.com	1021	1030 (0.92%)	599	614 (2.50%)
facebook.com	444	447 (0.63%)	256	259 (1.18%)

Policy flexibility

places.sqlite

Category	Use	Store
Browsing history	✔	✘
Cookies	✘	✘
Cache	✘	✘
Local storage	✘	✘
Flash storage	✘	✘
Download entries	✘	✘
Autofills	✔	✘
Bookmarks	✔	✔
Per-site zoom	✔	✘
Per-site permission	✔	✘
SSL self-signed cert	✔	✘
SSL client cert	✔	✔
Add-on storage	✔	✔
(All others)	✘	✘

Discussion:
Customizable / personalized private mode



Specify default
set of policies

Toggle policies
to meet own expectations

Discussion: Portability

- Not solely for browsers, in fact, other applications that are yet to have a private mode available would benefit from this design.




Discussion: Cross-platform

- *seccomp-bpf*: available since Linux Kernel 3.5
- *ptrace*: already available on Mac OS and has substitution on MS Windows




In conclusion, UCognito …

- Provides universal implementation for all browsers
- Caters to with user expectation
- Does not add complexity to browser codebase
- Is secure-by-default

Thank you !

Q & A