Bridges to Self: Silent Web-to-App Tracking on Mobile via Localhost

(aka Local Mess)

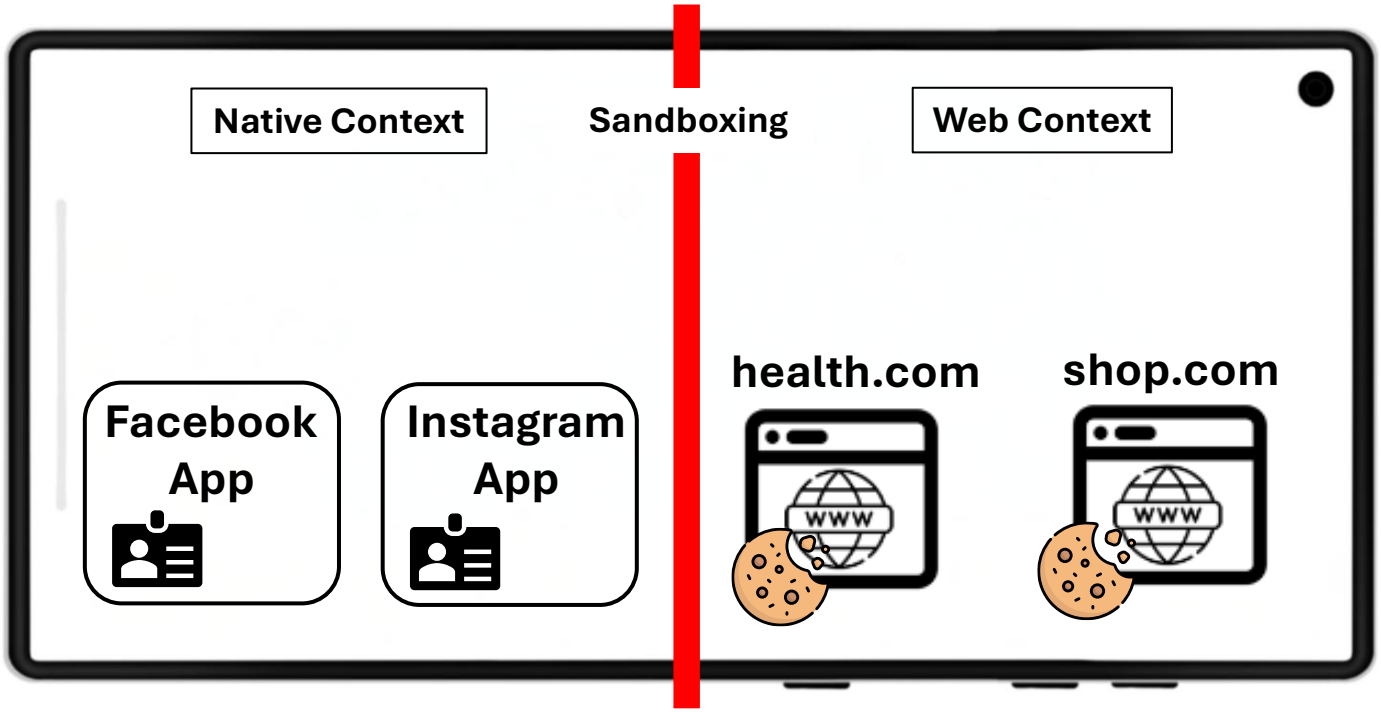
Tim Vlummens, Aniketh Girish, Nipuna Weerasekara, Frederik Zuiderveen Borgesius, Gunes Acar, Narseo Vallina-Rodriguez

🏆 ***CNIL-Inria Privacy Award 2026***
🏆 ***USENIX Security'26 Internet Defense Prize 2026***
🏆 ***USENIX Security'26 Distinguished Paper Award***

Radboud Universiteit · institute imdea networks · KU LEUVEN · COSIC

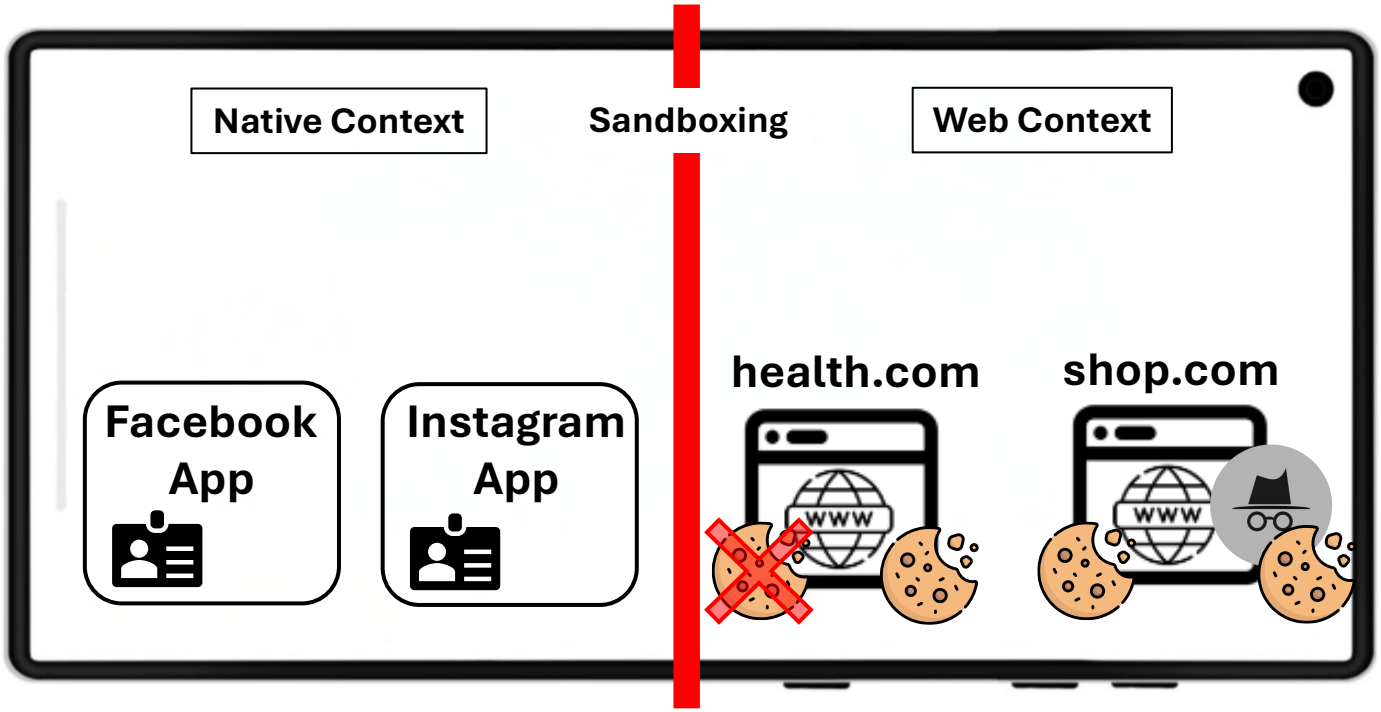
The Problem: Local Mess

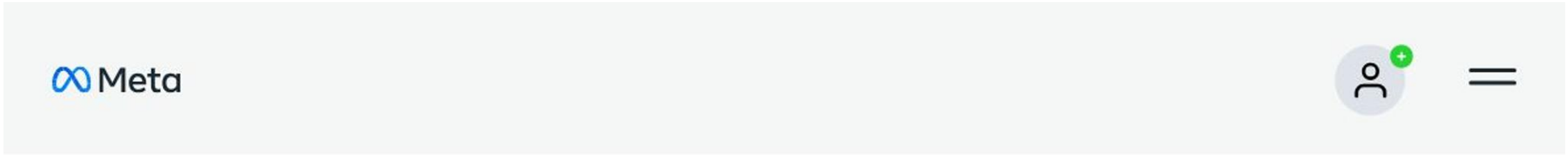
- Web users tracked by cookies, not linked to identity
  - Enable incognito mode, clear cookies, work profiles
- Apps know identity, but not online activity



The Problem: Local Mess

- Tracked on web by cookies, not linked to identity
  - Enable incognito mode, clear cookies, work profiles
- Apps know identity, but not online activity





What is a pixel?

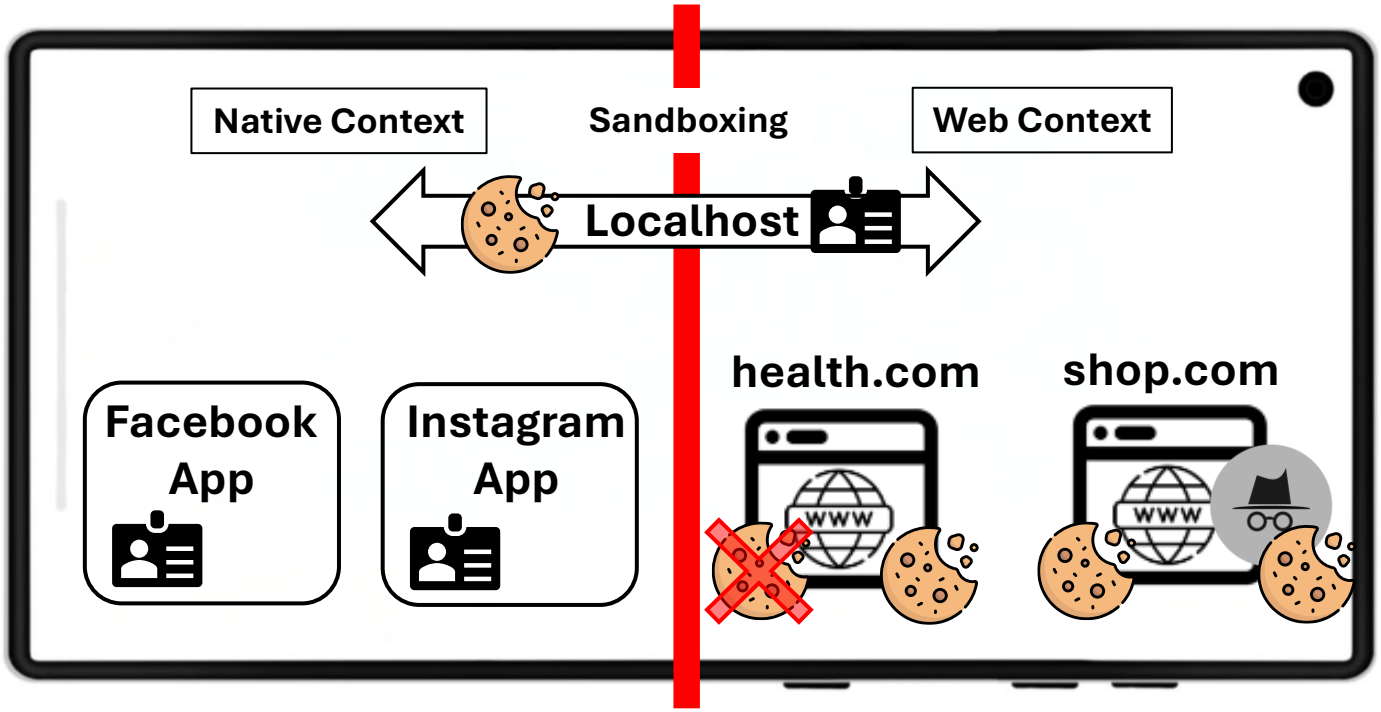
Meta Pixel is a piece of JavaScript code that you add to your website to track website activity and optimize your advertising performance. Millions of businesses globally use Meta Pixel — it's the foundation for data-driven marketing that delivers measurable results.

It works by sharing data about user actions on your website — from page views to purchases — and sends this information to Meta to improve ad delivery and measurement.

https://www.facebook.com/business/tools/meta-pixel

The Problem: Local Mess

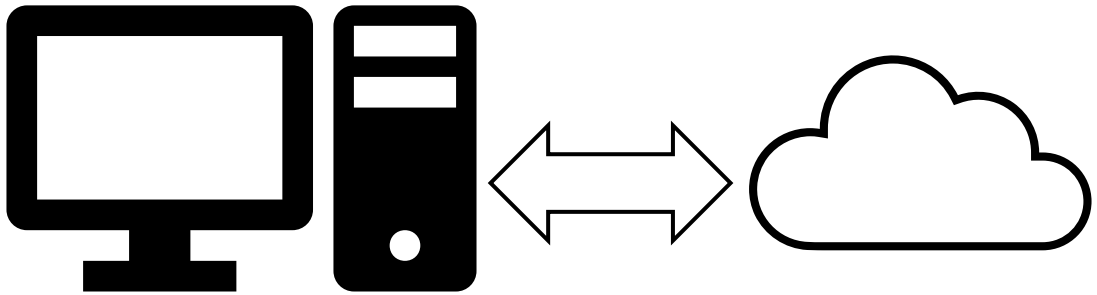
- Method bridges web to app identities
  - Links online activity to user identities
- Used by Meta and Yandex



Localhost: 127.0.0.1

Remote endpoint

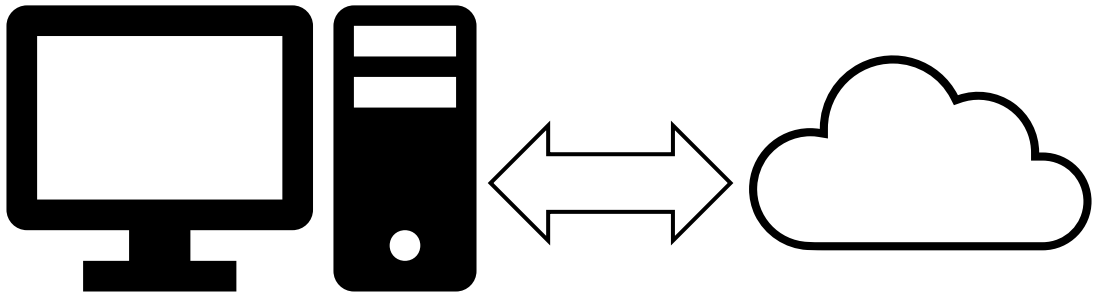
- Connecting to a remote server
  - URL resolves to server



Localhost: 127.0.0.1

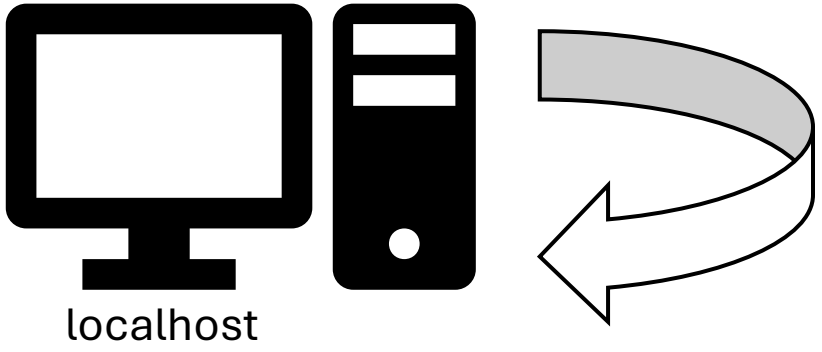
Remote endpoint

- Connecting to a remote server
  - URL resolves to server



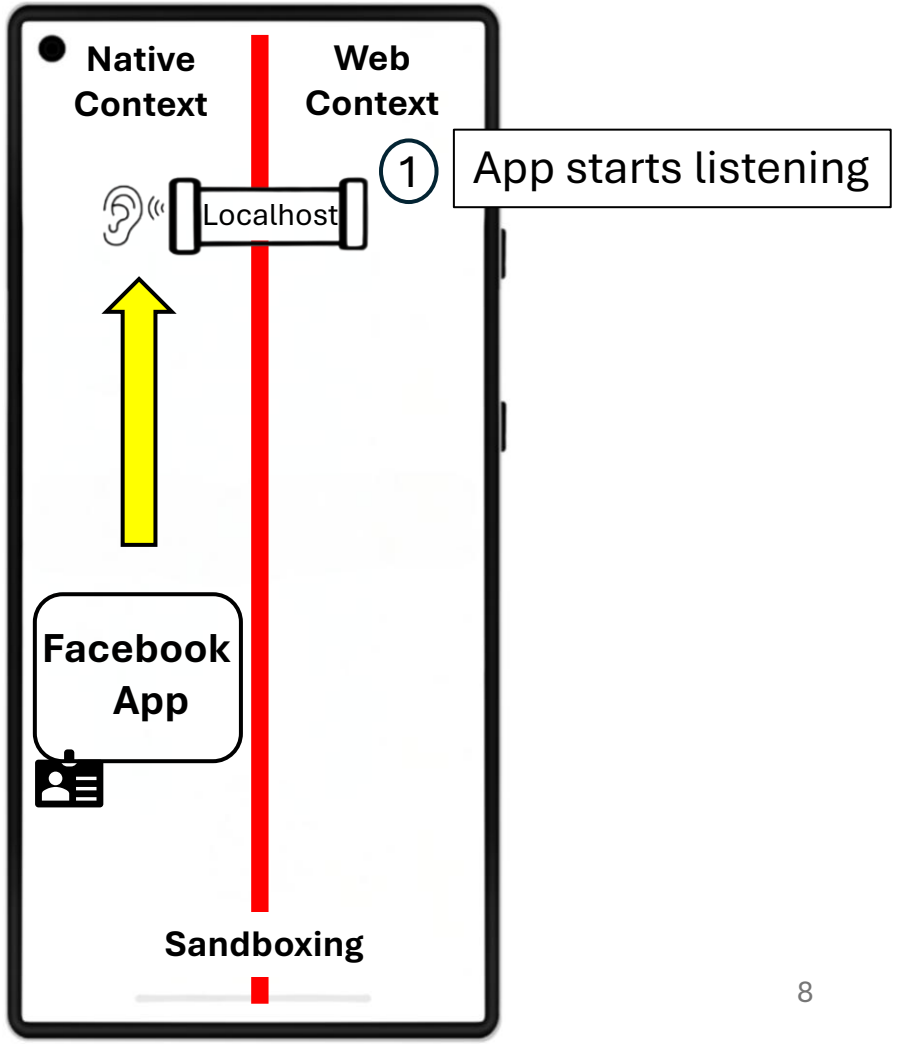
Localhost

- Connecting to local server on own device
- Allows Inter Process Communications



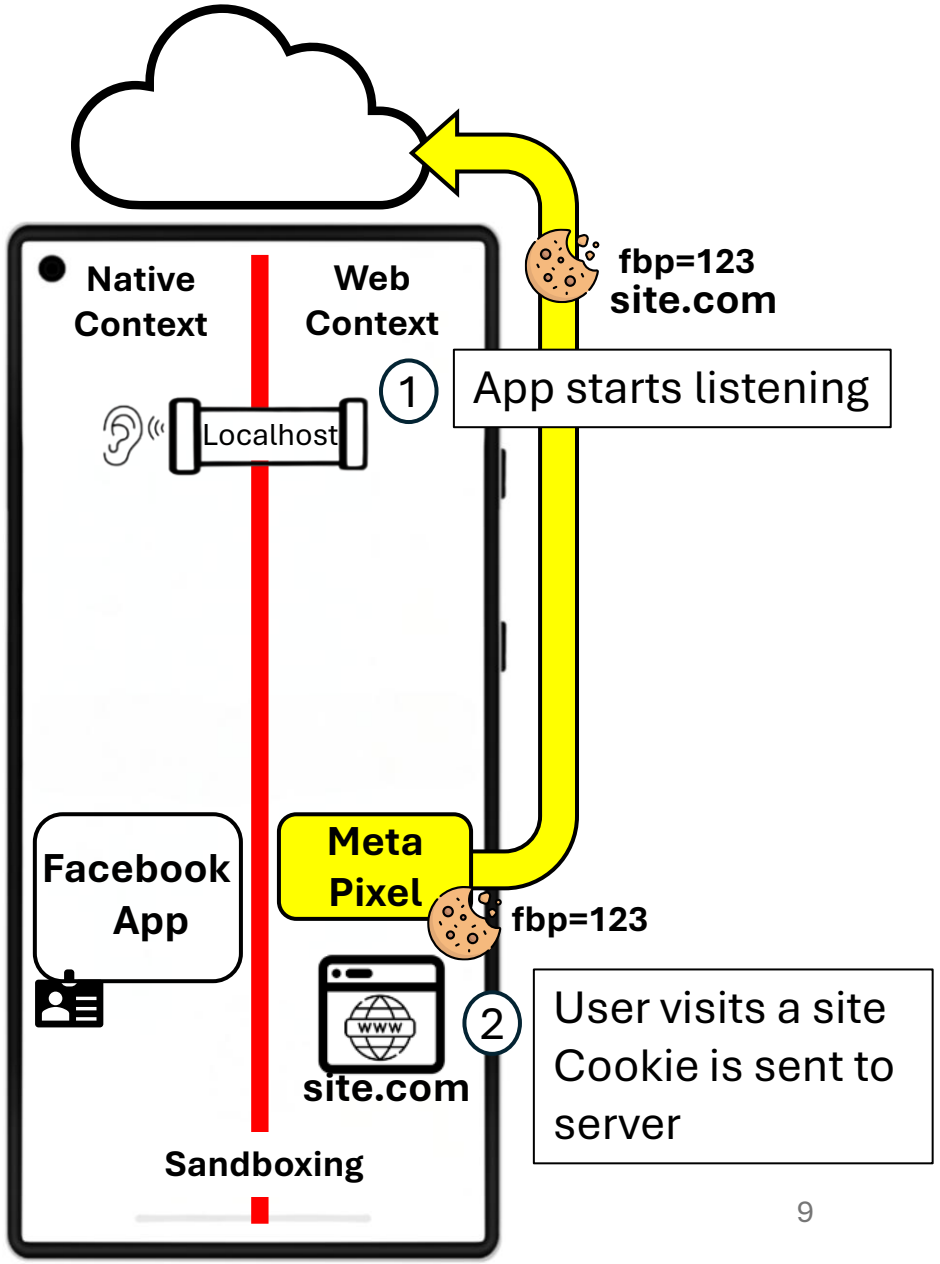
Localhost Channel: Meta

1. App starts listening to network requests to localhost



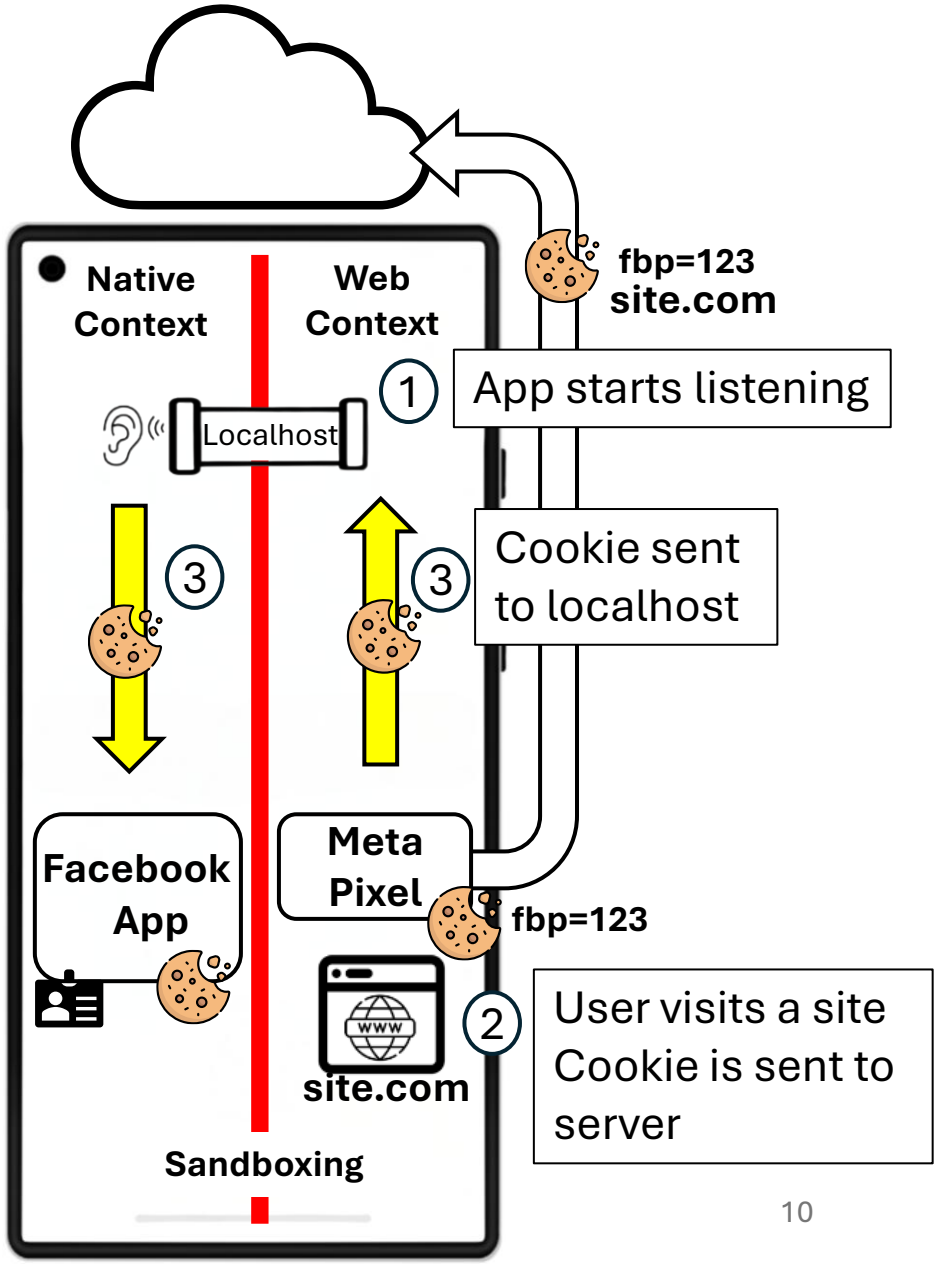
Localhost Channel: Meta

2. When user visits a site, cookie is generated



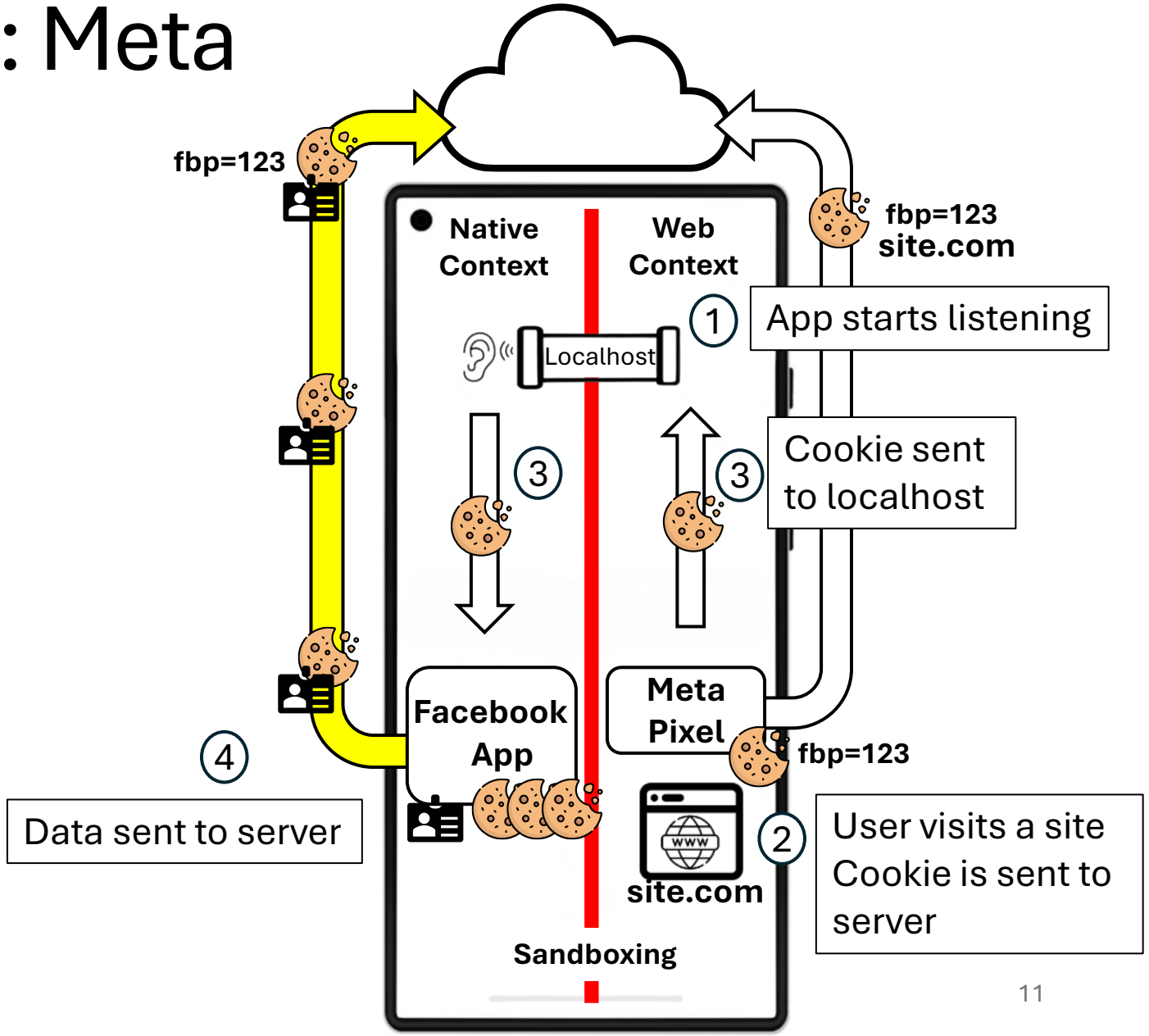
Localhost Channel: Meta

3. Cookie is sent to app
   via localhost



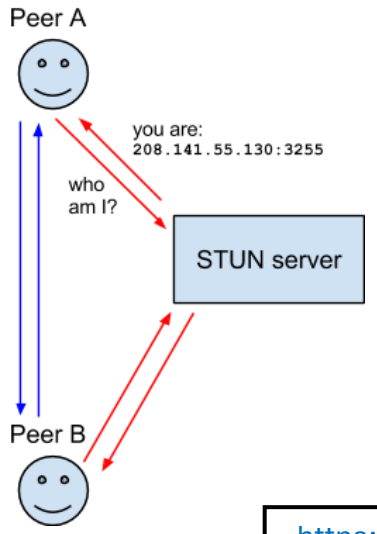
Localhost Channel: Meta

4. Cookies and identities sent to remote servers



WebRTC

- Peer-to-peer communication for multimedia such as video calls
- STUN / TURN: Protocols to establish the connection
- SDP: Data format to describe connection and media support
- SDP munging: Modification of SDP string




```
    suppression with any endpoint that does not offer it, making silence
    suppression support bilateral even with non-JSEP endpoints.

5.4.  Modifying an Offer or Answer

    The SDP returned from createOffer or createAnswer MUST NOT be changed
    before passing it to setLocalDescription.  If precise control over the
    SDP is needed, the aforementioned createOffer/createAnswer options or
    RtpTransceiver APIs MUST be used.

    After calling setLocalDescription with an offer or answer, the
    application MAY modify the SDP to reduce its capabilities before
```

https://datatracker.ietf.org/doc/html/rfc8829#name-modifying-an-offer-or-answe

```
}, p);
m.onnegotiationneeded = function(b) {
    return m.createOffer().then(function(b) {
        b.sdp = b.sdp.replace(/^a=ice-ufrag.*$/m, "a=ice-ufrag:aid+" + h),
        b.sdp = b.sdp.replace(/^a=ice-pwd.*$/m, "a=ice-pwd:OEKutPgoHVk/99FfqPOf444w"),
        m.setLocalDescription(b),
```

"fb+1+1744300[redacted]+202166819879[redacted]"

b = {sdp: 'v=0\r\no=- 296[redacted]6214332281235 2 IN IP4 127.0.0.1\r\ns=-\r\nt=0\r\na

fb_pixel_stun.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

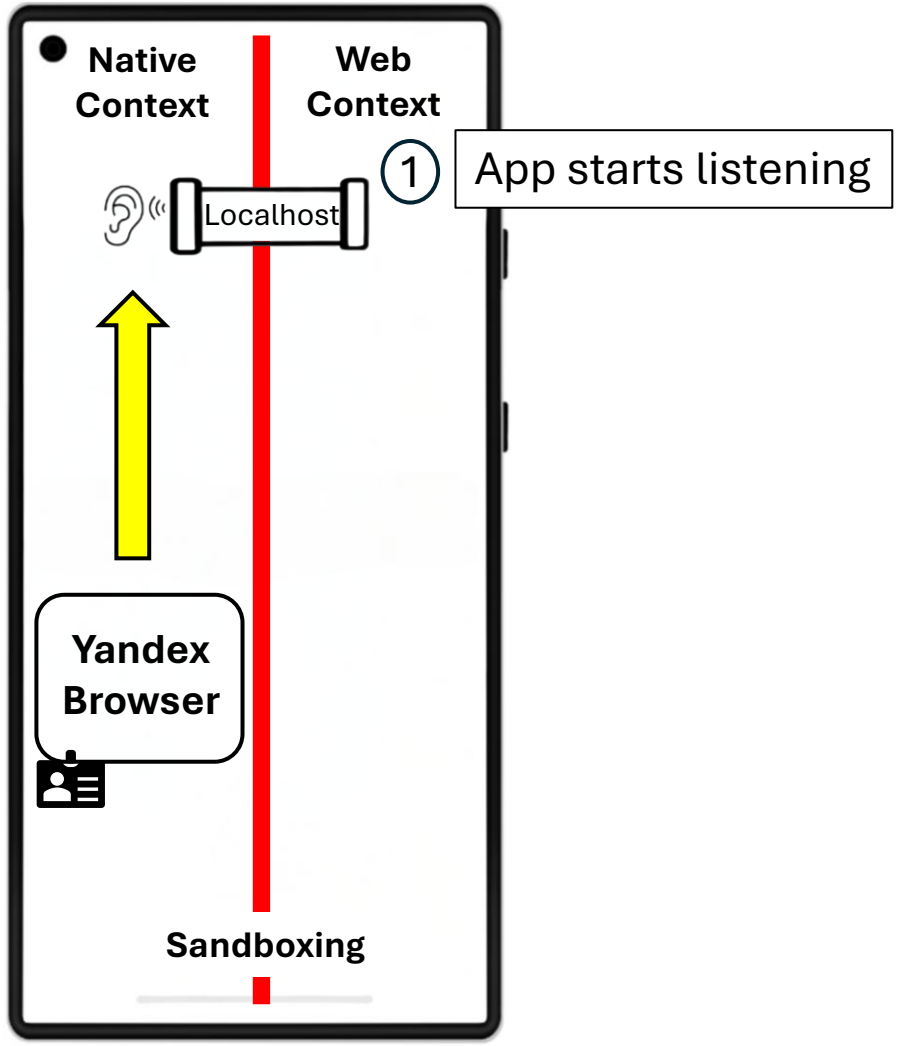
No.	Time	Source	Destination	dst port	Protocol	Info
1	0.00000…	localhost	localhost	12580	STUN	Binding Request user: fKVhbscsMWDGAnBg:aid+fb+1+17392340[redacted]+1032466223733[redacted]
2	0.00239…	localhost	localhost	12580	STUN	Binding Request user: fKVhbscsMWDGAnBg:aid+fb+1+17392340[redacted]+1032466223733
3	0.05606…	localhost	localhost	12581	STUN	Binding Request user: fKVhbscsMWDGAnBg:aid+fb+1+17392340[redacted]+1032466223733
4	0.05618…	localhost	localhost	12581	STUN	Binding Request user: fKVhbscsMWDGAnBg:aid+fb+1+17392340[redacted]+1032466223733
5	0.11832…	localhost	localhost	12582	STUN	Binding Request user: fKVhbscsMWDGAnBg:aid+fb+1+17392340[redacted]+1032466223733
6	0.11839…	localhost	localhost	12582	STUN	Binding Request user: fKVhbscsMWDGAnBg:aid+fb+1+17392340[redacted]+1032466223733
7	0.18088…	localhost	localhost	12583	STUN	Binding Request user: fKVhbscsMWDGAnBg:aid+fb+1+17392340[redacted]+1032466223733
8	0.18099…	localhost	localhost	12583	STUN	Binding Request user: fKVhbscsMWDGAnBg:aid+fb+1+17392340[redacted]+1032466223733
9	0.24332…	localhost	localhost	12584	STUN	Binding Request user: fKVhbscsMWDGAnBg:aid+fb+1+17392340[redacted]+1032466223733
10	0.24340…	localhost	localhost	12584	STUN	Binding Request user: fKVhbscsMWDGAnBg:aid+fb+1+17392340[redacted]+1032466223733
11	0.30591…	localhost	localhost	12585	STUN	Binding Request user: fKVhbscsMWDGAnBg:aid+fb+1+17392340[redacted]+1032466223733
12	0.30603…	localhost	localhost	12585	STUN	Binding Request user: fKVhbscsMWDGAnBg:aid+fb+1+17392340[redacted]+1032466223733
13	0.36857…	localhost	localhost	12580	STUN	Binding Request user: fKVhbscsMWDGAnBg:aid+fb+1+17392340[redacted]+1032466223733

```
.... ...0 ...0 .... = Message Class: 0x00 Request (0)
..00 000. 000. 0001 = Message Method: 0x0001 Binding (0x001)
..0. .... .... .... = Message Method Assignment: IETF Review (0x0)
Message Length: 124
Message Cookie: 2112a442
Message Transaction ID: 2b6f6c2f4d73376859366239
[STUN Network Version: RFC-5389/8489 (3)]
Attributes
  USERNAME: fKVhbscsMWDGAnBg:aid+fb+1+17392340[redacted]+1032466223733[redacted]
    Attribute Type: USERNAME
      0... .... .... .... = Attribute Type Comprehension: Required (0x0)
      .0.. .... .... .... = Attribute Type Assignment: IETF Review (0x0)
    Attribute Length: 57
    Username: fKVhbscsMWDGAnBg:aid+fb+1+17392340[redacted]+1032466223733[redacted]
    Padding: 3
  GOOG-NETWORK-INFO
    Attribute Type: GOOG-NETWORK-INFO
      1... .... .... .... = Attribute Type Comprehension: Optional (0x1)
      .1.. .... .... .... = Attribute Type Assignment: Designated Expert (0x1)
    Attribute Length: 4
```

Attribute Type (stun.attribute), 64 bytes | Packets: 679 · Displayed: 679 (100.0%) | Profile: Default

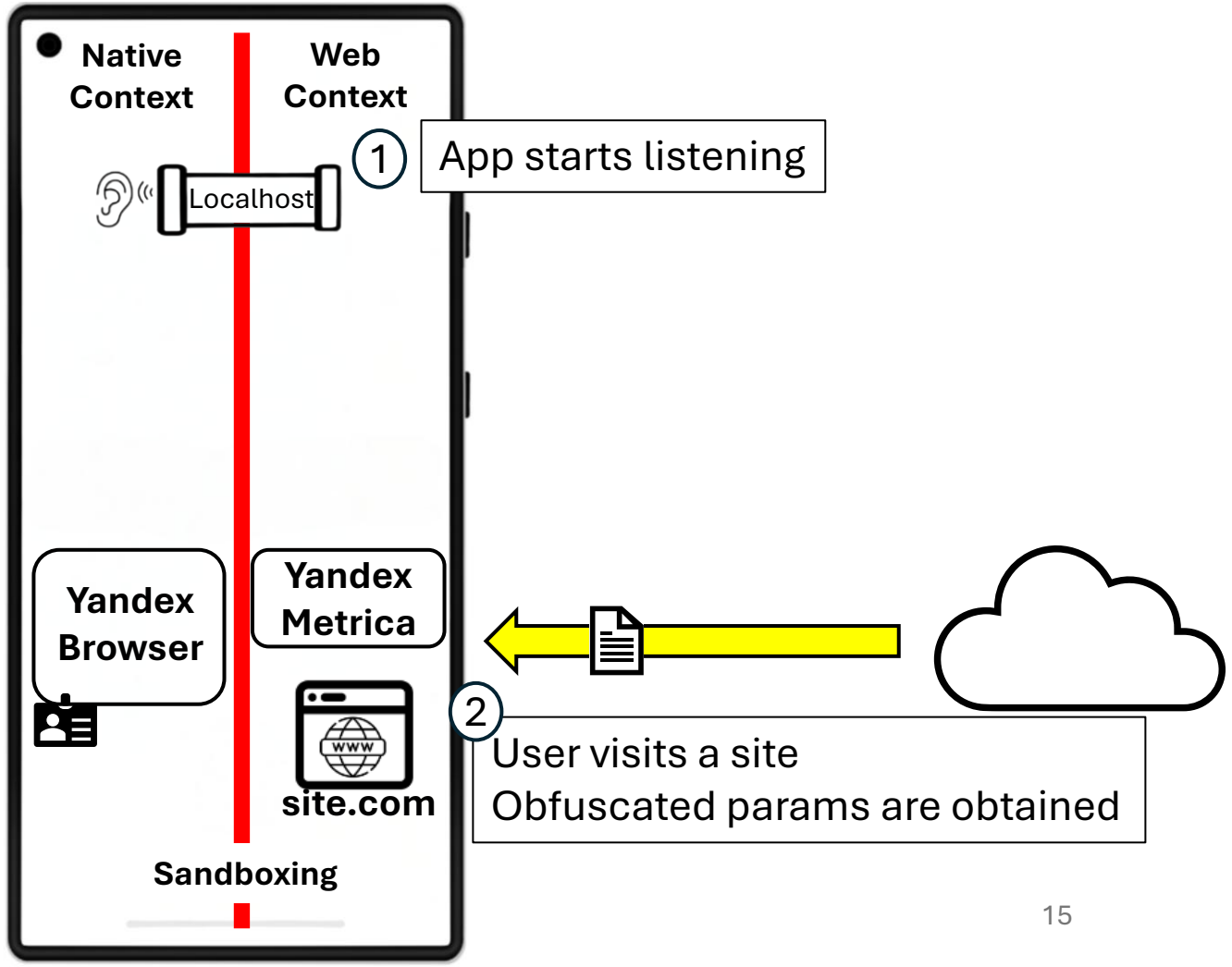
Localhost Channel: Yandex

1. App starts listening to network requests sent to localhost



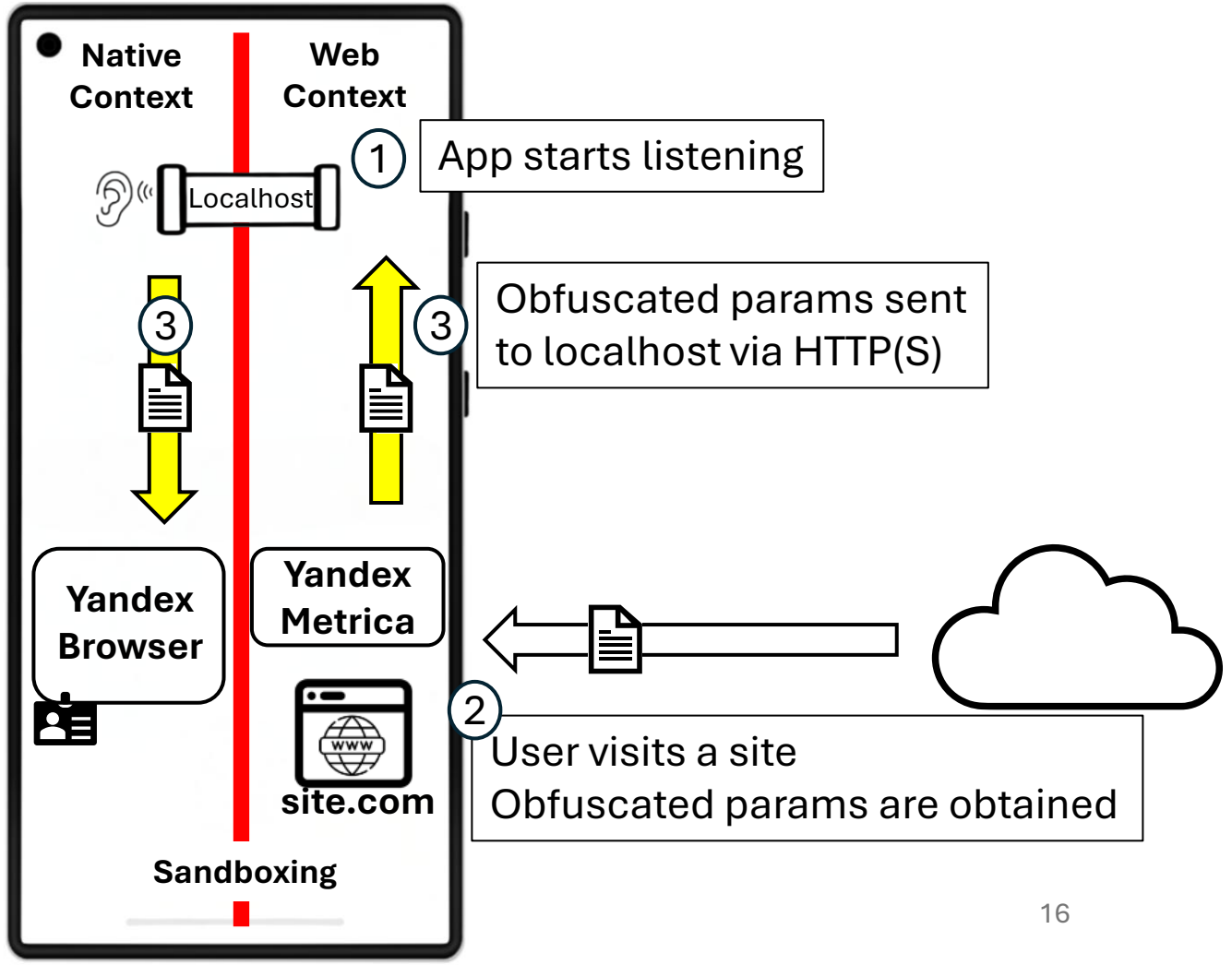
Localhost Channel: Yandex

2. When user visits a site, Yandex Metrica script obtains obfuscated params from the server



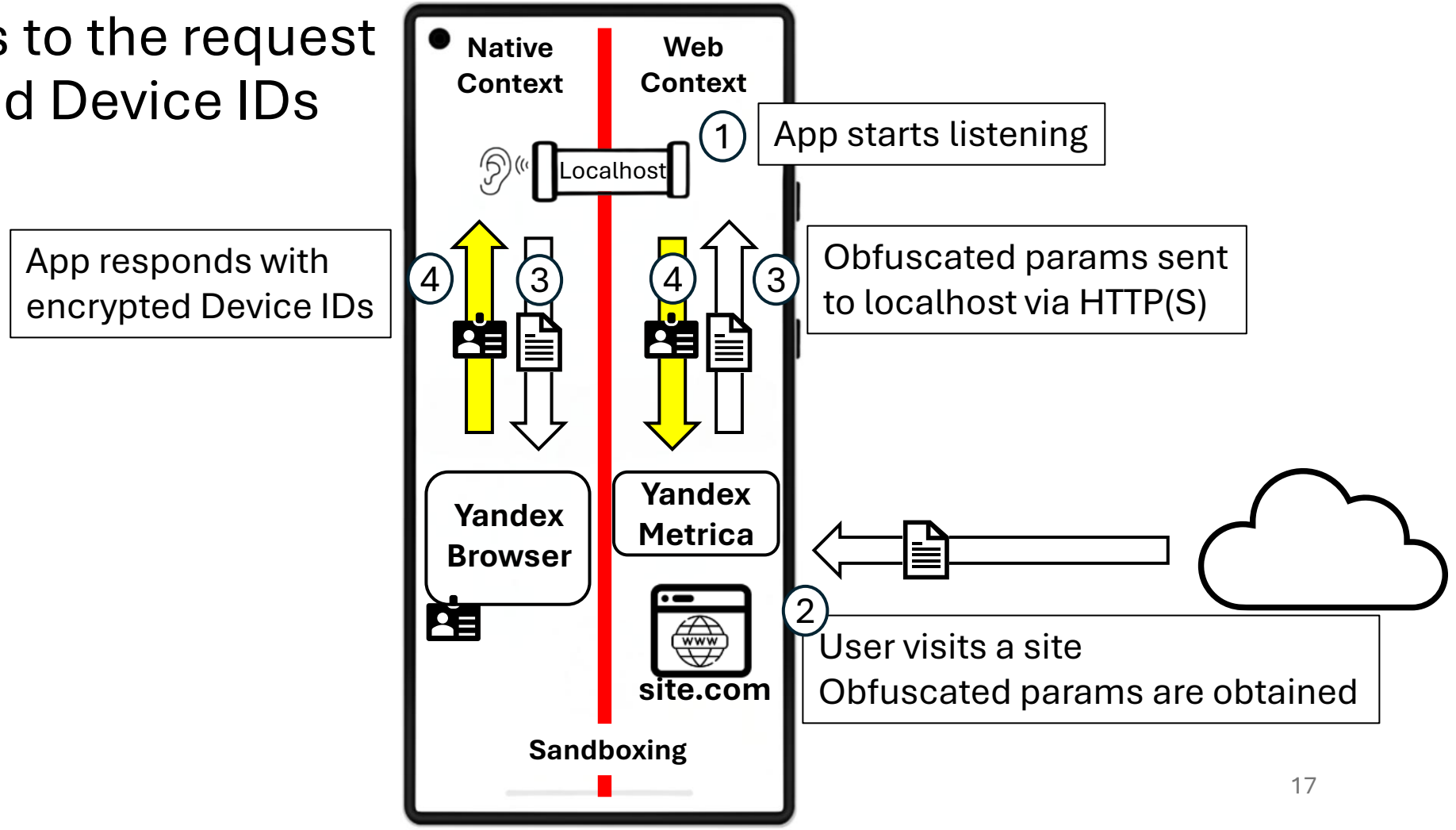
Localhost Channel: Yandex

3. Script sends obtained obfuscated params to localhost via HTTP(S)



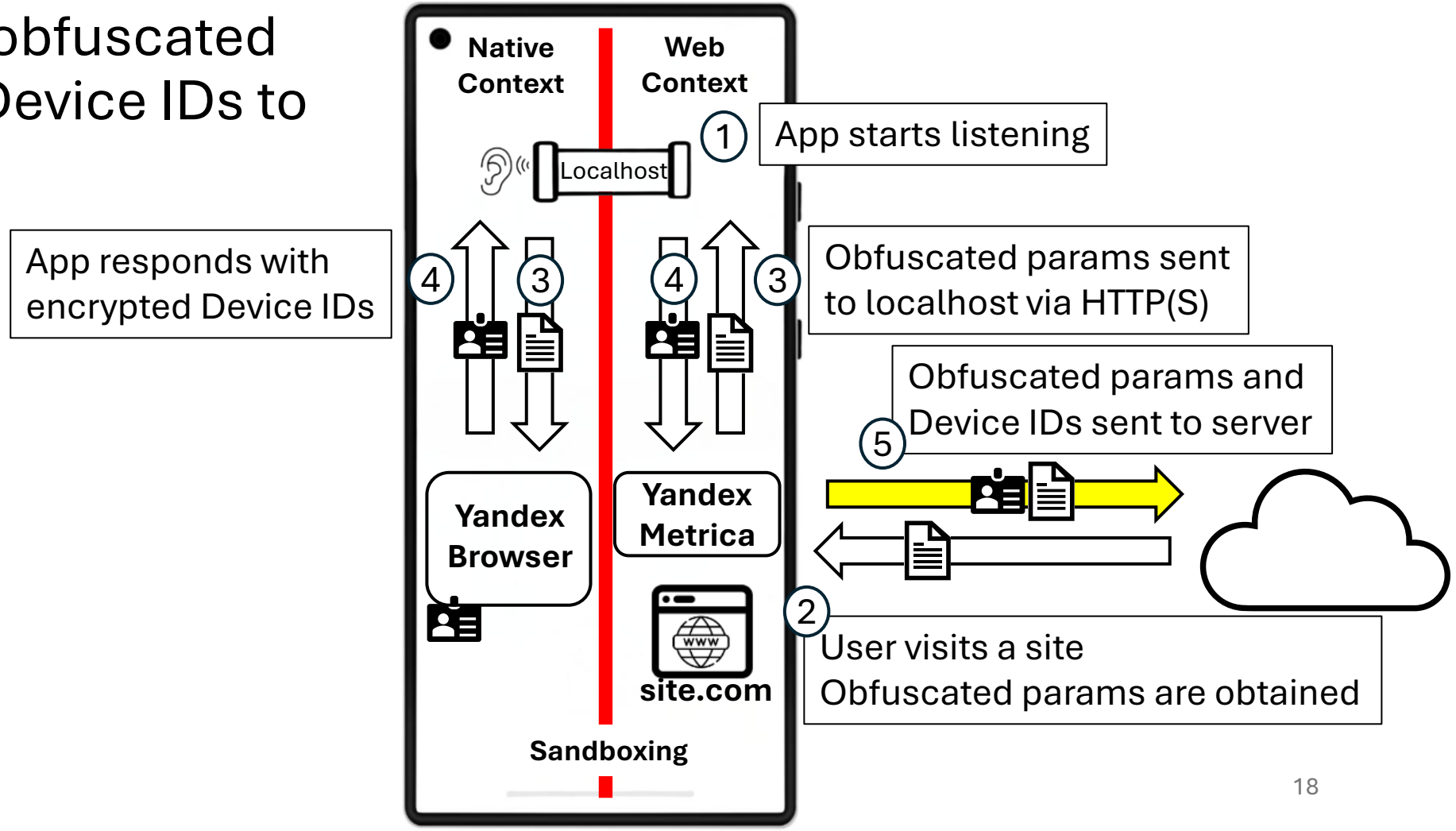
Localhost Channel: Yandex

4. App responds to the request with encrypted Device IDs



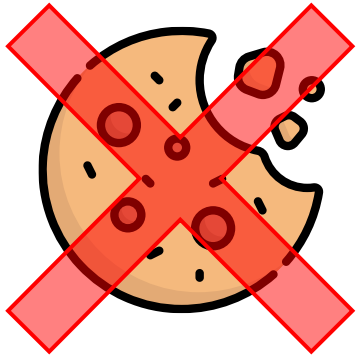
Localhost Channel: Yandex

5. Script sends obfuscated params and Device IDs to the server



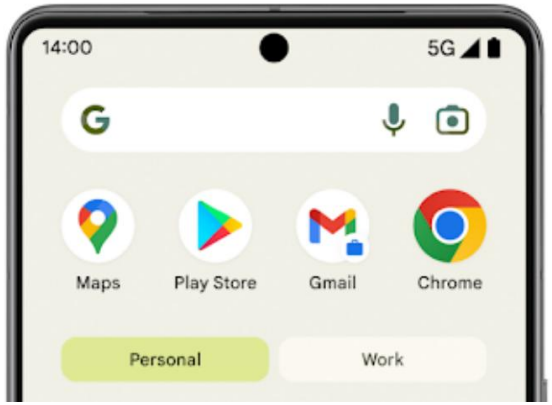
Bypassed Privacy Protections

- Clearing cookies
- Incognito mode
- VPN
- (Android) Work profiles









Prevalence

- **Meta Pixel** embedded on **over 5.8 million websites**
- **Yandex Metrica** present on **close to 3 million websites**
- Lower bounds

https://builtwith.com

Prevalence

- **Meta Pixel** embedded on **over 5.8 million websites**
- **Yandex Metrica** present on **close to 3 million websites**
- Lower bounds
- Only targeted Android phones
  - Method is technically possible on iOS

https://builtwith.com

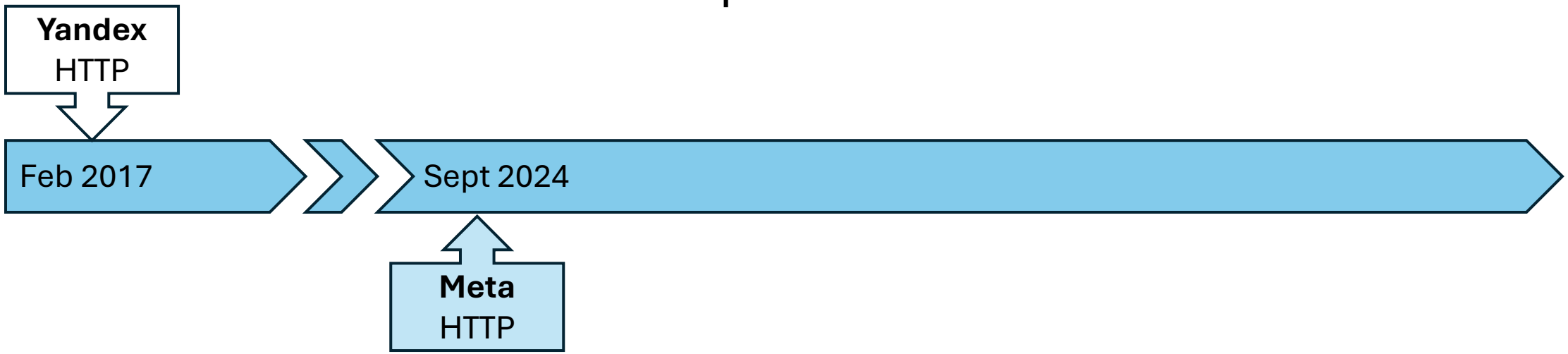
Historical Evolution

- Using historical data (HTTP Archive data and APK archives)
- Yandex remained undetected for **8 years** 🤯
  - App listening only starts 3 days after installation
  - Collusion of web scripts and mobile apps



Historical Evolution

- Using historical data (HTTP Archive data and APK archives)
- Yandex remained undetected for **8 years** 🤯
  - App listening only starts 3 days after installation
  - Collusion of web scripts and mobile apps
- Meta switched between multiple different methods



Facebook SDK config file making call to localhost

13

Since 5th September, our internal JS error tracking has been flagging failed fetch requests to localhost:12387. No changes have been made on our side, and the existing Facebook tracking pixel we use loads via Google Tag Manager.

Specifi
https://
call to l

(functic

Status	Unresolved
Posted	over a year ago

Why does my Pixel Javascript access http://localhost when in an embedded web view on Android

7

We have a meta tracking pixel on a client website, embedded and tracking correctly using google tag manager. We also show some of the content from the site embedded in a web view in their Android app. We've noticed tha the Webview is throwing an error as the Pixel javascript attempts to access http://localhost. Android, by default since Android 9 disallows cleartext http connections unless explicitly enabled. We don't wish to enable this as it could be insecure. I have two questions 1. Anyone know why it accesses http://localhost ? 2. Why does it do it over an http not https? 3. could this be caused by a misconfiguration?

Status	Unresolved
Posted	over a year ago
Last Answered	over a year ago
Type	Unclassified

script retrieved from
able&domain= &hme

hange the version an

way I can disable this

No word from Facebook, raised a ticket but have just got arbitrary stuff back so far. Will update if I get to the bottom of it

September 24 at 12:19 PM

https://developers.facebook.com/community/threads/317050484803752/

4

This looks to be resolved now if you want to check.

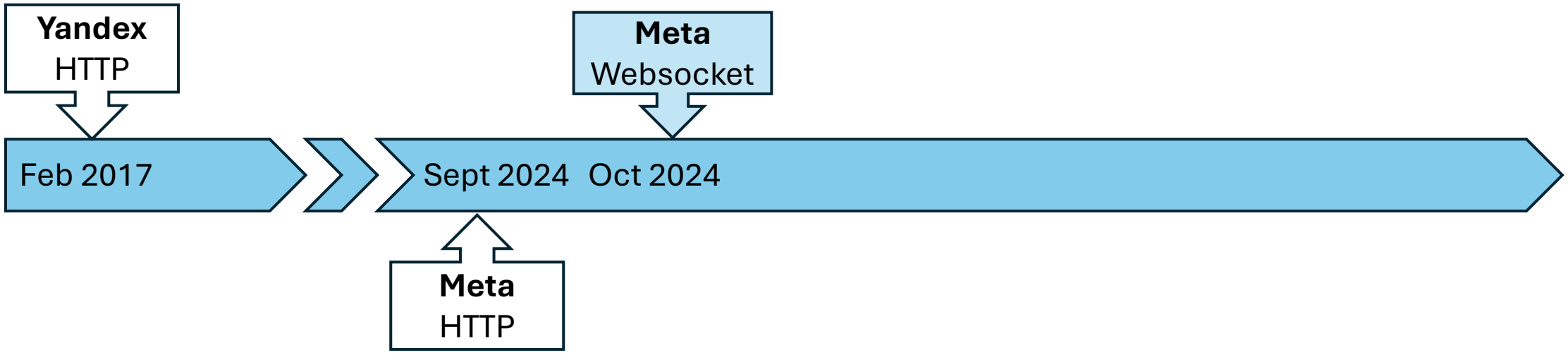
October 10 at 7:48 AM

for me too, thanks!

October 10 at 11:06 AM

This seems to be fixed on my SDK as of 1st October between 19:00 and 20:00 BST. No acknowledgement has come from Meta at all on this though. My support request with them got a generic response and then ignored thereafter.

October 21 at 12:16 PM

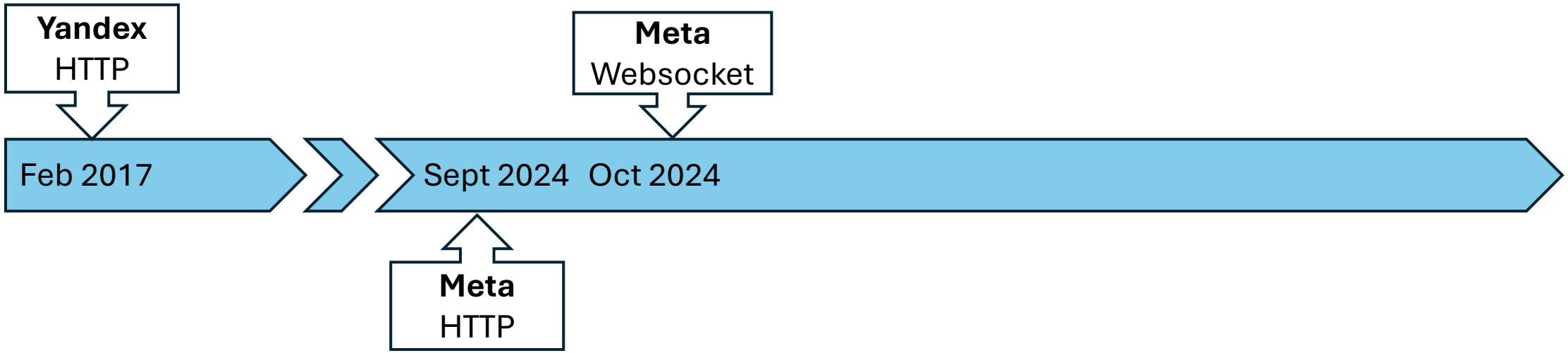


https://developers.facebook.com/community/threads/317050484803752/

Still getting this but now it's a websocket connection

https://connect.facebook.net/signals/config/330423837344621?v=2.9.173&r=stable

October 24 at 4:50 PM

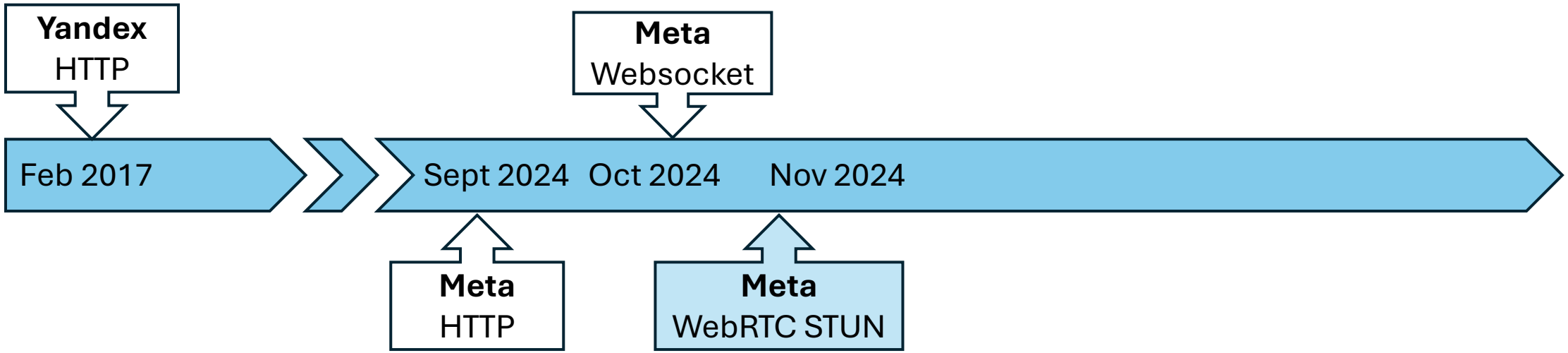


https://developers.facebook.com/community/threads/317050484803752/

Still getting this but now it's a websocket connection

https://connect.facebook.net/signals/config/330423837344621?v=2.9.173&r=stable

October 24 at 4:50 PM



https://developers.facebook.com/community/threads/317050484803752/

PSA: Modification of SDP “ufrag” and “passwd” attributes is going away 1,488 views

Apr 11, 2025, 11:46:02 AM

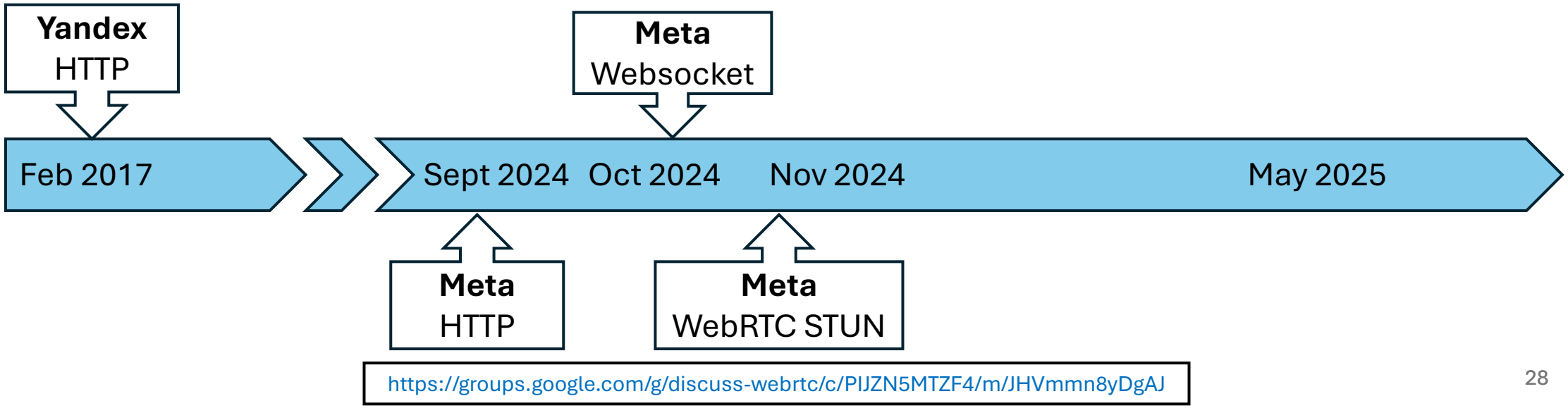
(If you’re not modifying SDP, you can stop reading now)
The standards for WebRTC have always said that the SDP produced by CreateOffer MUST be passed unchanged to SetLocalDescription

https://datatracker.ietf.org/doc/html/rfc8829#name-modifying-an-offer-or-answe
https://w3c.github.io/webrtc-pc/#dom-peerconnection-setlocaldescription step 4.2

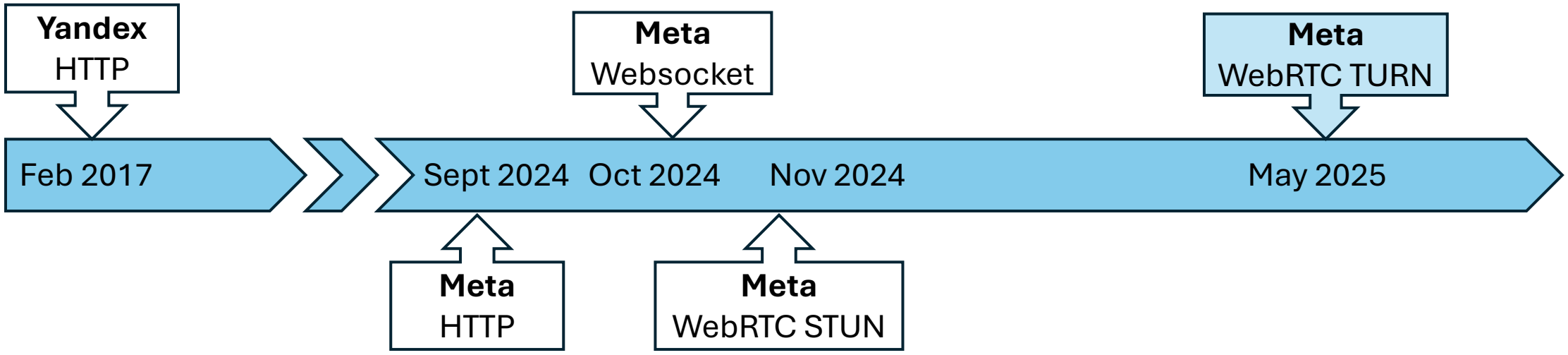
Unfortunately, this has not been policed in Chrome, and people have taken advantage of this in various ways, some of which are obsoleted by later API extensions, and some of which are actively harmful and should not be permitted.

One of the changes we’ve found to be actively harmful is the ability to modify the “ice-ufrag” and “ice-passwd” attributes. We have therefore decided that in the future, calls to SetLocalDescription that have modified these attributes will be rejected, as specified in the standard.
More restrictions are likely to be enabled further on.

The precise rollout dates have not yet been set.

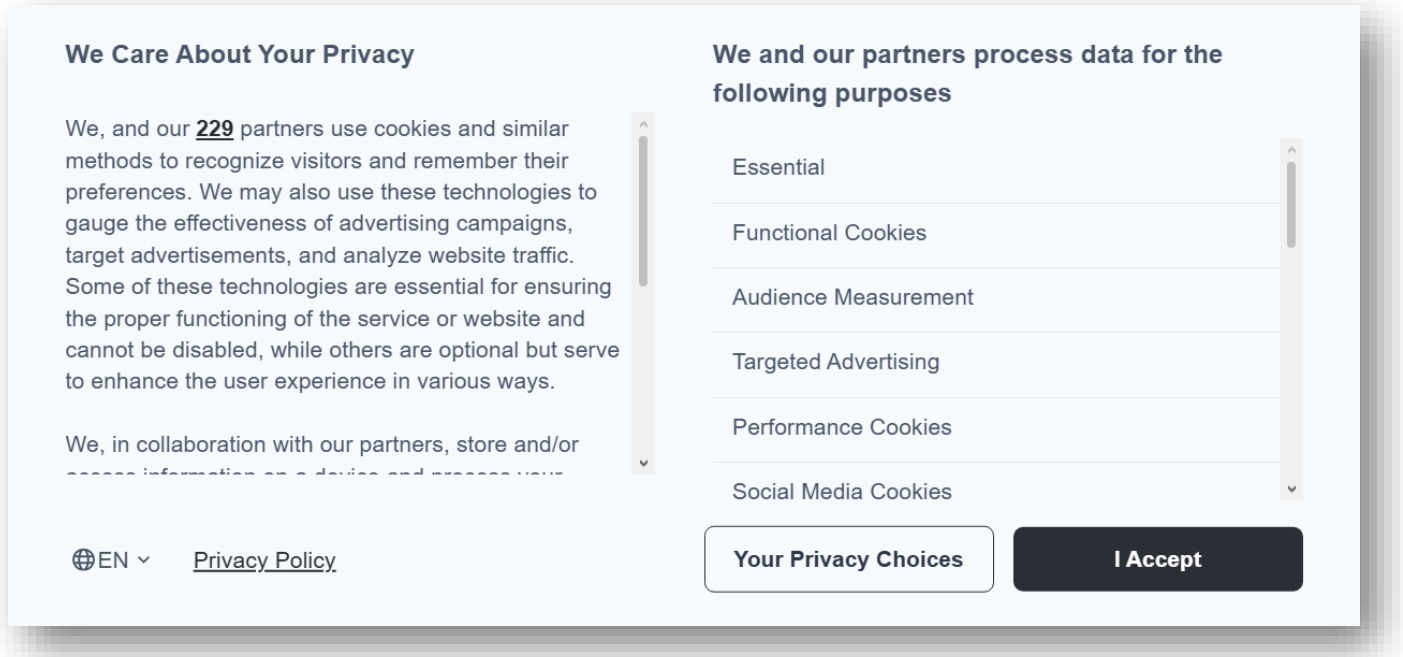


```
config.set("64770[redacted]", "actionID", {
    "portNumber": 12387,
    "ttlInHour": 1,
    "rtcPortNumbers": [12580, 12581, 12582, 12583, 12584, 12585],
    "enableStun": true,
    "enableTurn": true,
    "turnPortNumbers": [12586, 12587, 12588, 12589, 12590, 12591]
});
```



Default Behaviour

- Automated visits on 100k sites from US and EU vantage point
- Meta/Yandex tracking **by default on page load on over 75% of sites**



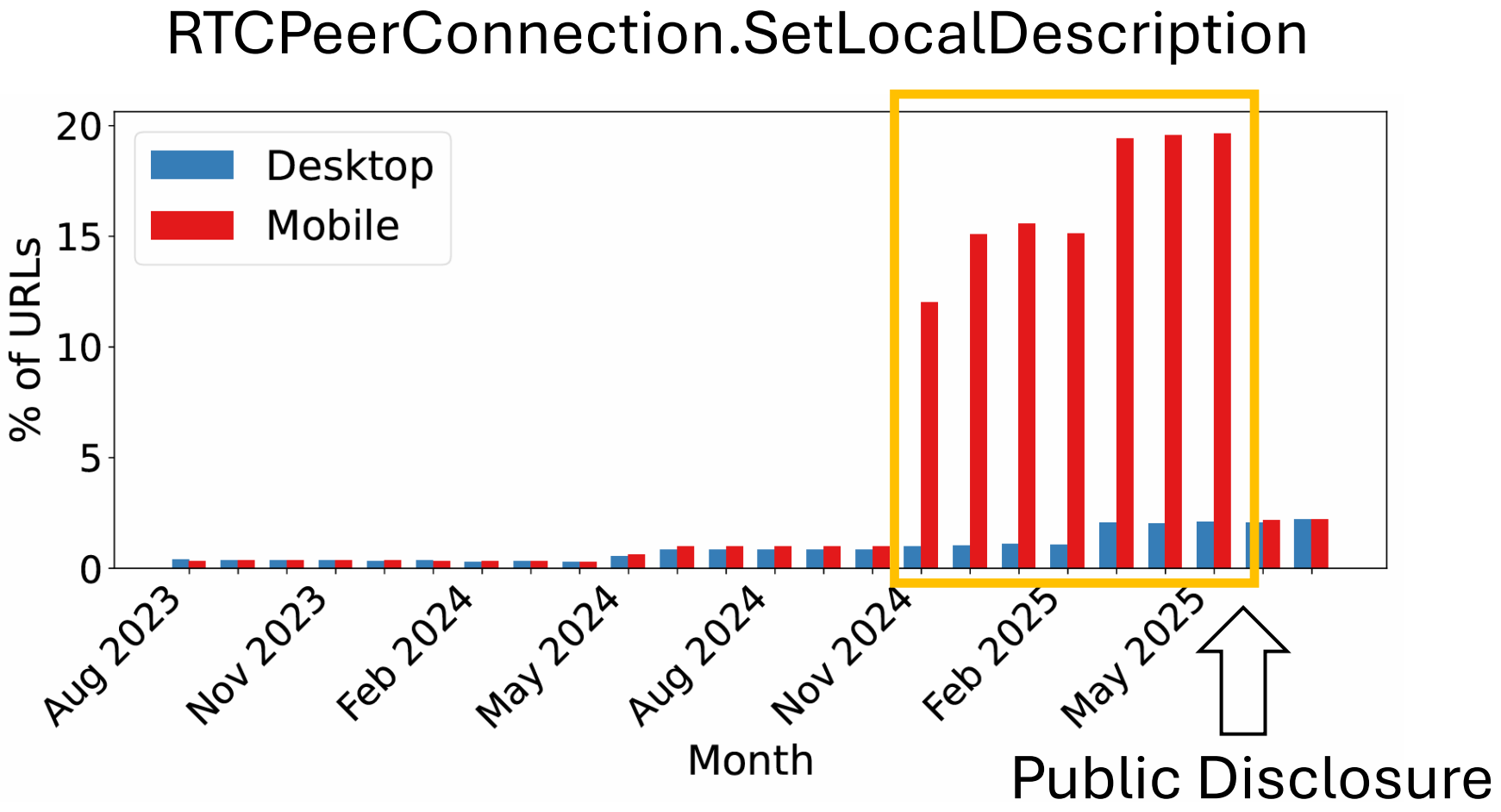
Disclosure Process

- Android and Chrome
  - March 2025
- Other browser vendors
  - Mozilla, Brave, DuckDuckGo, Edge, WebKit (Safari)
  - April-May 2025
- Regulators: European DPAs, US FTC, UK CMA
  - Other consumer protection agencies
  - May 2025

Disclosure Process

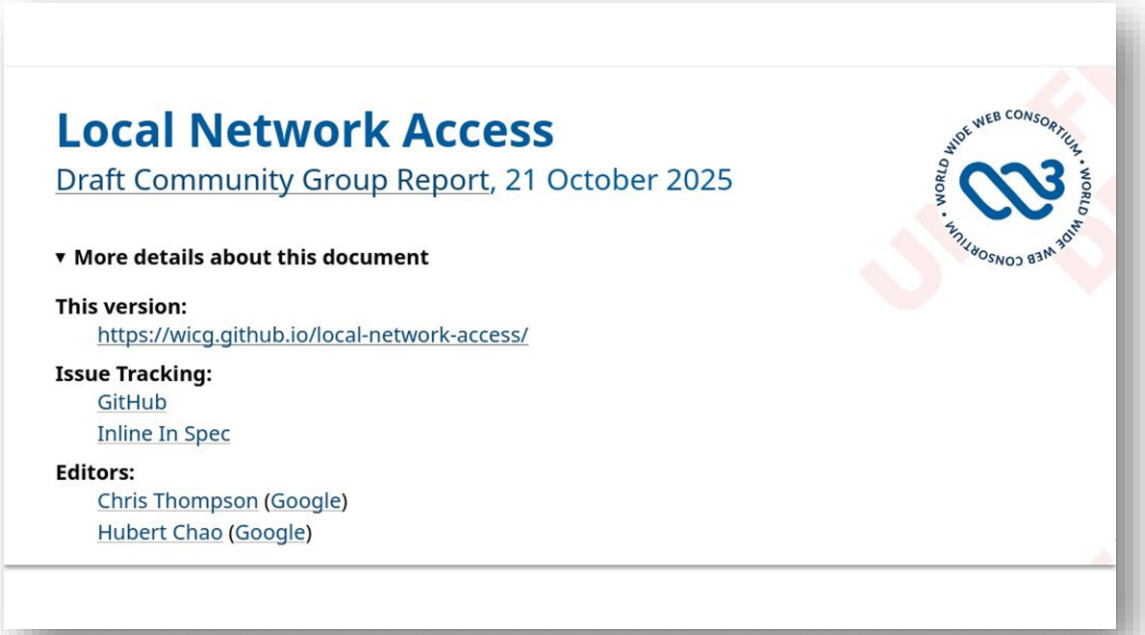
- Public site on June 3rd 2025
  - Concurrent to mitigations deployed by browsers
  - Link: https://localmess.github.io/
- Meta / Yandex stopped on the day of our disclosure

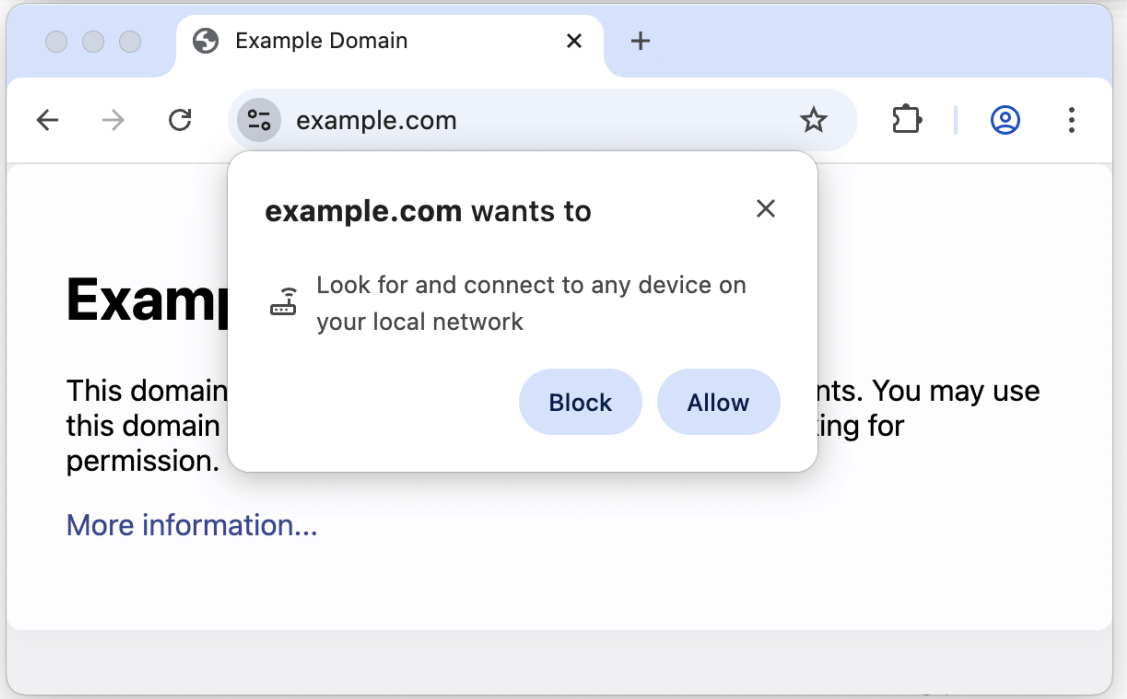
> 📢 **UPDATE**: As of June 3rd 7:45 CEST, Meta/Facebook Pixel script is no longer sending any packets or requests to localhost. The code responsible for sending the _fbp cookie has been almost completely removed. Yandex has also stopped the practice we describe below.

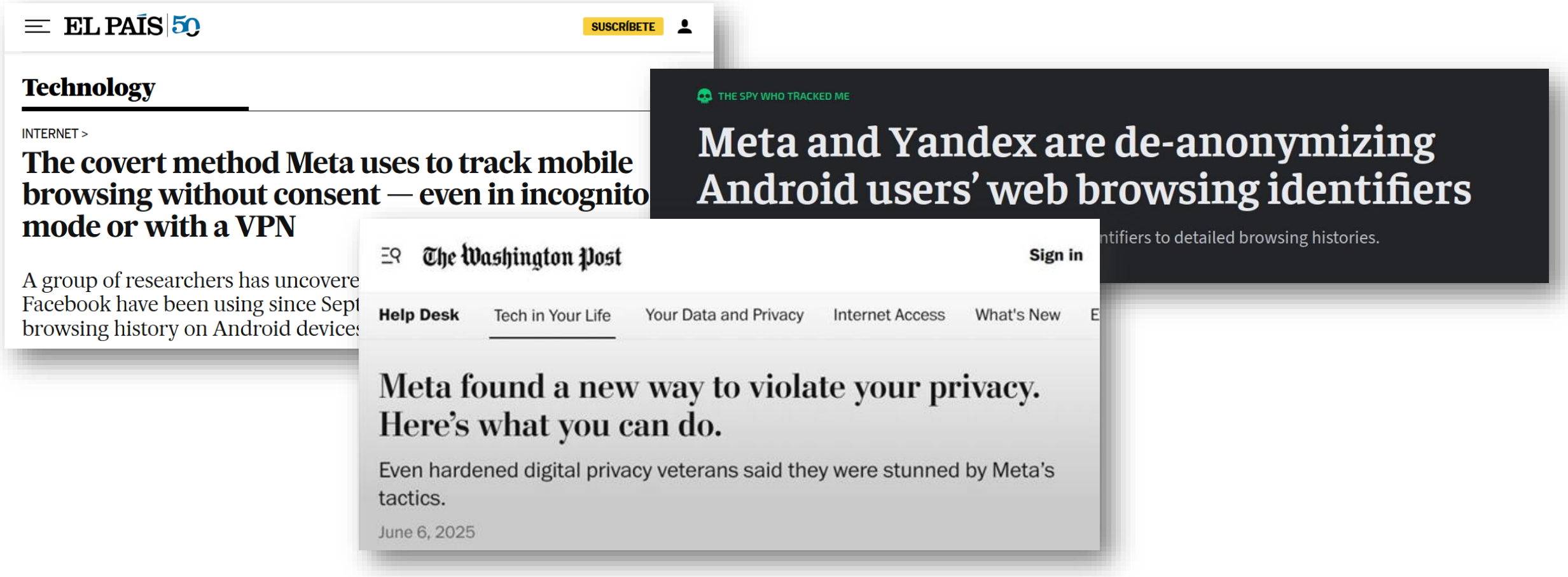

RTCPeerConnection.SetLocalDescription
Desktop
Mobile
% of URLs
0
5
10
15
20
Aug 2023
Nov 2023
Feb 2024
May 2024
Aug 2024
Nov 2024
Feb 2025
May 2025
Month
Public Disclosure

Mitigations

- Local Network Access (**LNA**)
  - Deployed by Chrome, Edge, Firefox, DuckDuckGo, Brave
  - ❌Safari
  - Permission prompts
- Local Network Permissions (**LNP**)
  - Android 17







"The developers [Meta & Yandex] in this report are using capabilities present in many browsers across iOS and Android in unintended ways that blatantly violate our security and privacy principles" — Google Representative

Congress of the United States

Washington, DC 20515

December 9, 2025

Mr. Mark Zuckerberg
Chairman and Chief Executive Officer
Meta Platforms, Inc.
1 Hacker Way
Menlo Park, CA 94025

Dear Mr. Zuckerberg,

We write to express our serious concerns regarding recently published research revealing Meta's use of a technical loophole to deliberately circumvent users' privacy choices and track users on the Android platform. We seek clarity on how Meta will locate and permanently delete data collected through this practice and what changes Meta will implement in its internal decision-making processes to prevent future privacy violations that override users' explicit intentions and expectations.

Unbeknownst to Meta's users, Meta's new tracking method followed them across the web even if the user tried to keep their browsing private. In May 2025, researchers revealed that Meta had been abusing Android features intended exclusively for testing and development to dismantle long-standing security protections built into the mobile operating system.[1] By exploiting these features, Meta enabled its apps, like Facebook and Instagram, to covertly surveil users' behavior on websites embedding the Meta Pixel, which appears on more than 20% of all websites.[2] Each

1. When did Meta begin using localhost snooping to covertly track users?
2. What internal documentation (e.g., design reviews or risk assessments) was created to document the implementation and use of this new tracking method? Does this documentation indicate this was a repurposing of developer-only tools?
3. What consideration, if any, did Meta give to how this tracking method might collect user data not otherwise available?
4. Who approved or authorized the use of this tracking method?
5. How many users were tracked using this method, and how many websites embedding the Meta-Pixel initiated background connections to Meta apps?
6. Did Meta create or modify any user profiles, behavioral segments, or machine learning models using data obtained through this exploit?
7. Did Meta collect any sensitive categories of data through this covert localhost tracking method, such as hospital patient data, FAFSA applicant details, intimate health metrics, or tax-return figures? If so, what categories of sensitive data were involved, and how were they handled?
8. Did Meta produce a "Privacy Review Statement" to assess the privacy risks of this new tracking method as required by Meta's 2019 consent decree with the FTC?[11] Did the statement note any privacy risks and whether new safeguards were needed?

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

JENNIFER VINCENT, WILLIAM BANEY, and DAWN MONTANEZ, on behalf of themselves and all others similarly situated,

Plaintiffs,

v.

META PLATFORMS, INC.; ALPHABET INC.; and GOOGLE LLC,

Defendants.

Case No.

CLASS ACTION

COMPLAINT

JURY TRIAL DEMANDED

CANADA

PROVINCE OF QUÉBEC
DISTRICT OF MONTREAL

No.: 500-06-001400-254

SUPERIOR COURT
(Class Actions)

HUGO CHICOINE-BLAIS, natural person, [redacted]

Applicant

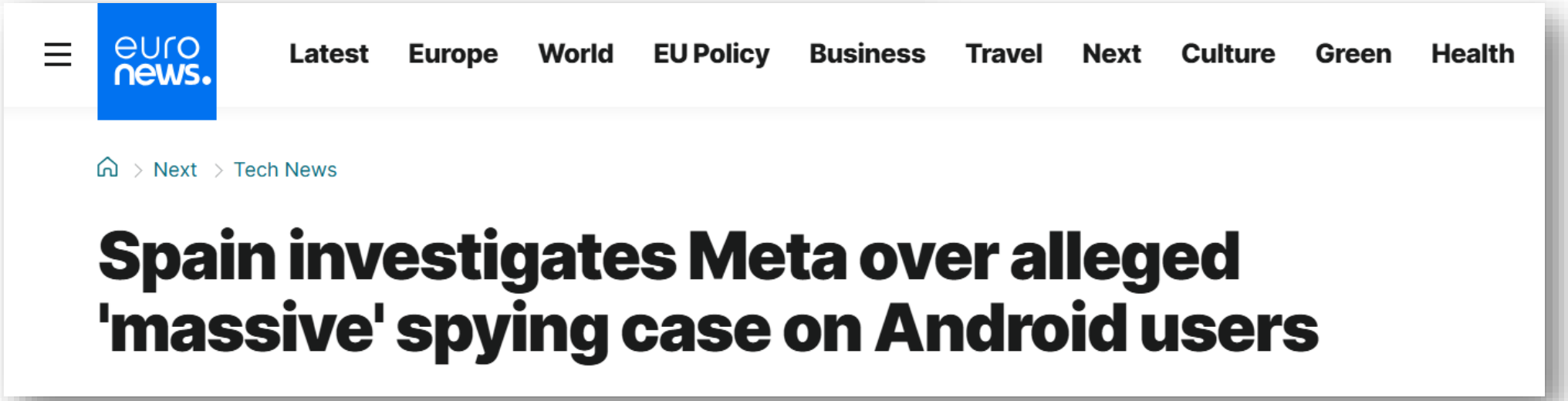
v.

META PLATFORMS, INC., legal person having its registered office at 251 Little Falls Drive, Wilmington, DE, 19808, USA

Defendant

APPLICATION FOR AUTHORIZATION TO INSTITUTE A CLASS ACTION
(Art. 571 C.C.P. and following)


MLex
SEE ALL SECTIONS || ANTITRUST MERGERS & ACQUISITIONS DATA PRIVACY & SECURITY ARTIFICIAL INTELLIGENCE INTELL
Meta sued in Germany over covert tracking, faces €3,000-per-user claim


euronews.
Latest Europe World EU Policy Business Travel Next Culture Green Health
Next > Tech News
Spain investigates Meta over alleged 'massive' spying case on Android users

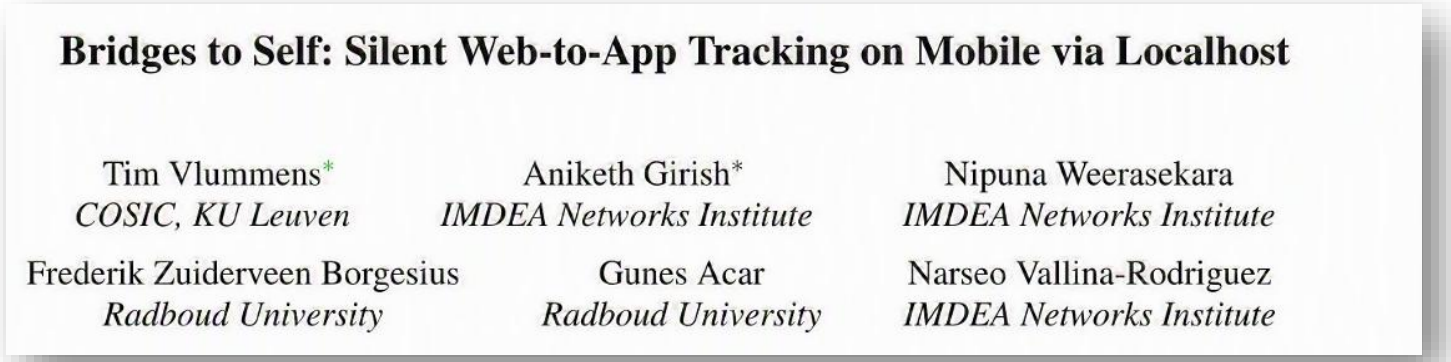
Conclusions

- Meta and Yandex bypassed protections to track web visits
  - Linking web identifiers to stable mobile identities
  - Using network protocols that are hard to monitor and block
  - Immune against Incognito Mode, cookie clears, work profiles
- Web scripts and mobile colluding for tracking
  - Difficult to spot in isolation
- Tracking campaign stopped upon disclosure

Credit: Jorge García Herrero



Paper, code, data: **localmess.github.io**



Bridges to Self: Silent Web-to-App Tracking on Mobile via Localhost

Tim Vlummens* — *COSIC, KU Leuven*
Aniketh Girish* — *IMDEA Networks Institute*
Nipuna Weerasekara — *IMDEA Networks Institute*
Frederik Zuiderveen Borgesius — *Radboud University*
Gunes Acar — *Radboud University*
Narseo Vallina-Rodriguez — *IMDEA Networks Institute*

🏆 ***CNIL-Inria Privacy Award 2026***
🏆 ***USENIX Security'26 Internet Defense Prize 2026***
🏆 ***USENIX Security'26 Distinguished Paper Award***

Radboud Universiteit · institute IMDEA networks · KU LEUVEN · COSIC