Surveillance capitalism

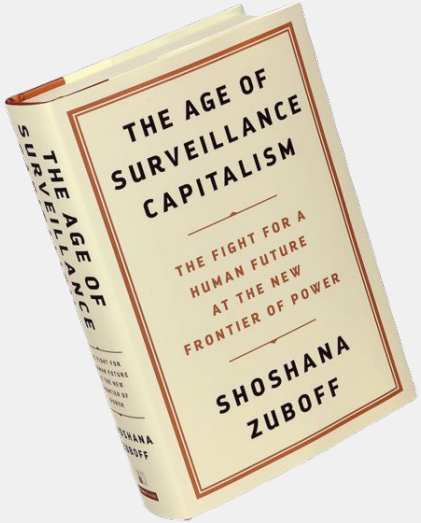
The battle for privacy in the age of information

What is surveillance capitalism?

"Surveillance capitalism is a new form of information capitalism [which] aims to predict and modify human behavior as a means to produce revenue and market control"[1]

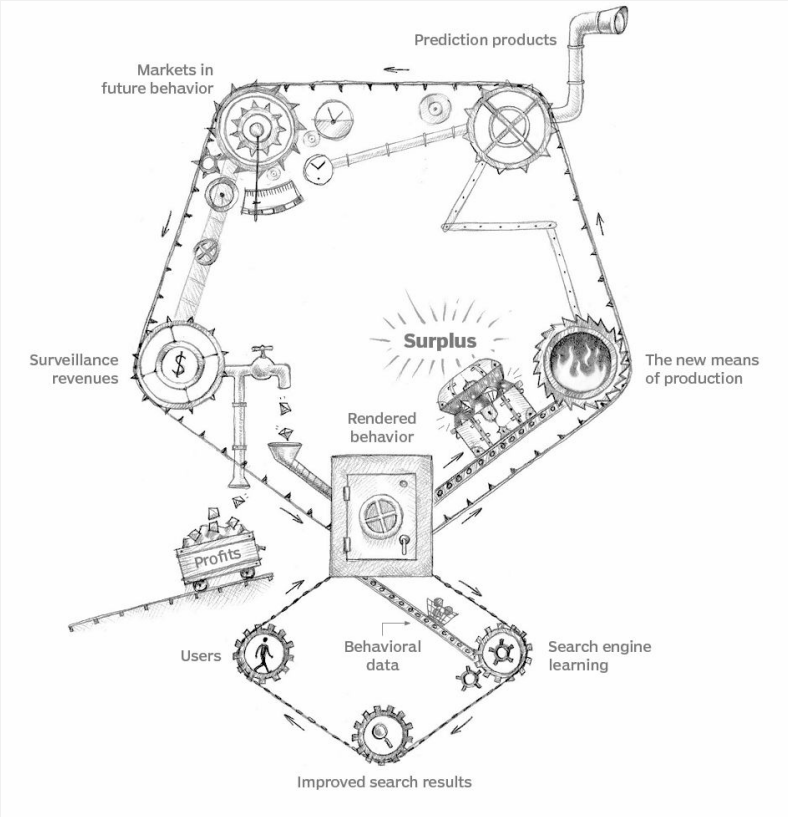
- Shoshana Zuboff

- Rejecting the term "Big Data", since it has no clear definition
- It's not a technology or a technology side effect
- It originates in the social and it's intentional



[1] Zuboff, Shoshana. Big other: Surveillance Capitalism and the Prospects of an Information. Civilization Journal of Information Technology 30:1, 75-89 2015.

High level overview



The surveillance capitalism machine [2]

1. **Data collection**
   Companies collect from users interacting with their products. This is data that doesn't have a direct relevance to an individual's use of a digital device or service but is a byproduct of their use.

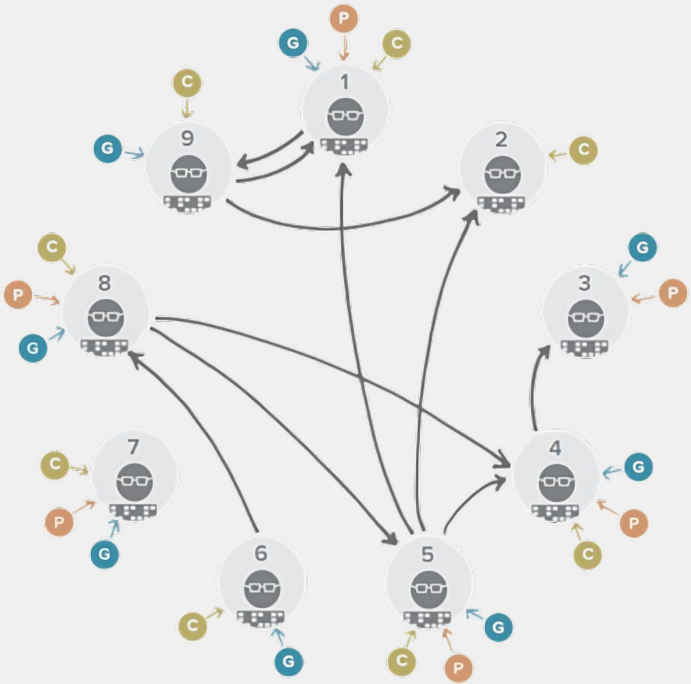
2. **Data manipulation**
   The data is fed into algorithms that try to predict or influence user actions.

3. **Behavioural markets**
   Companies sell these products in the “data-market”.

[2] Zuboff, Shoshana. “The Age of Surveillance Capitalism the Fight for a Human Future at the New Frontier of Power”. PublicAffairs, 2020.

We will look into how tech companies can collect data about their customers outside their platforms

How companies collect data



Data brokers and their source[3]

1. Government data
2. Publicly available data
3. Commercial source
   a. Data from retailers
   b. Registration to websites
   c. Browsing history
   d. Analytics services
   e. Free services

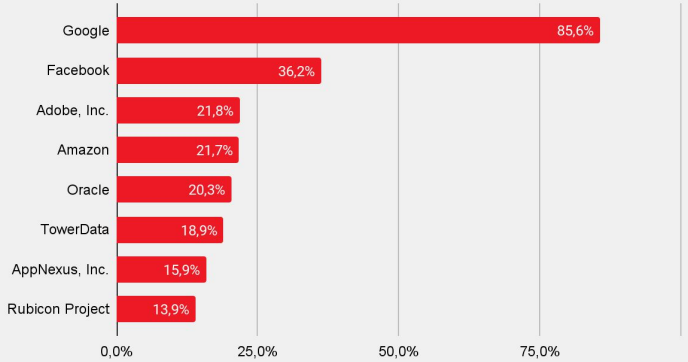
[3] Ramirez, E., Brill, J., Ohlhausen, M.K., D. Wright, J., McSweeny, T. (2014). “Data Brokers A Call for Transparency and Accountability”, Federal Trade Commission report.

Tracking in SDKs

Companies offer to developers libraries to enhance user experience. For example:

- Analytics
- Social network integration
- Crash reporting

In exchange, they collect data about the users using the products.



Trackers in the top 50k most visited websites[4]

"We find that 60% of the apps monitored by Haystack connect to at least one ATS domain and 20% of the apps use at least 5 ATS"[5]

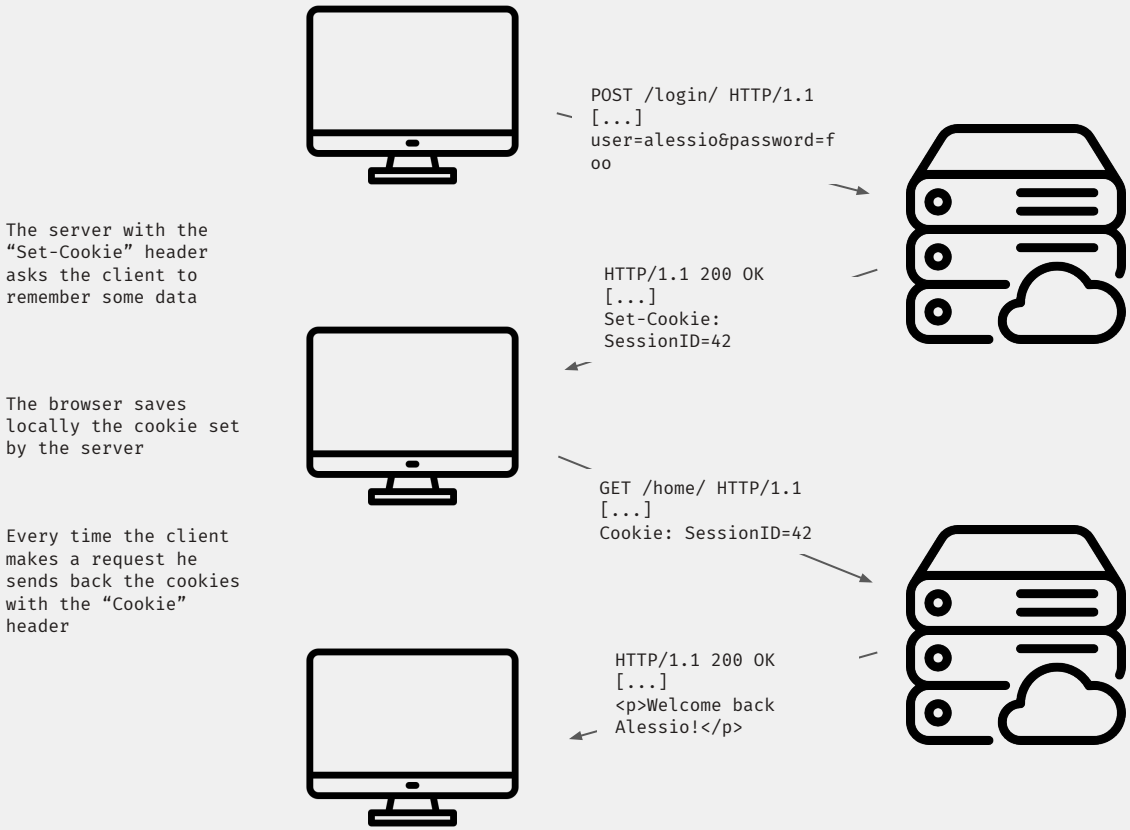
[4] https://web.archive.org/web/20230623120039/https://spreadprivacy.com/duckduckgo-tracker-radar/
[5] Vallina-Rodriguez N, Sundaresan S, Razaghpanah A, Nithyanand R, Allman M, Kreibich C, Gill P. Tracking the trackers: Towards understanding the mobile advertising and tracking ecosystem.

Cookies

Cookies are used to make the HTTP protocol stateful.

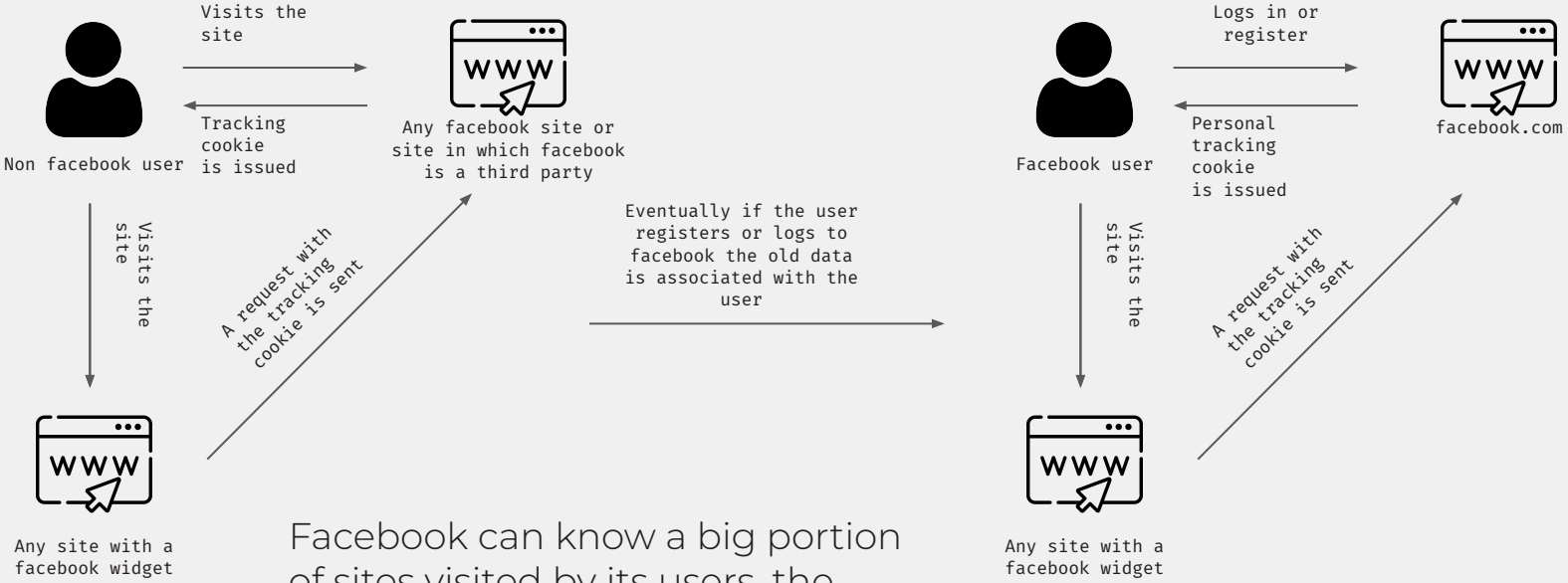
A server with an HTTP response can ask the client to save locally a piece of information (cookie).

Every time a client makes a request it will attach his cookies.



[6] https://datatracker.ietf.org/doc/html/rfc2109

Cross site tracking



Facebook can know a big portion of sites visited by its users, the same techniques are used also by other companies like Google.

[7] Acar, Güneş, et al. "Facebook tracking through social plug-ins." Technical report prepared for the Belgian Privacy Commission (2015): 1-24.
[8] https://web.archive.org/web/20230314181746/https://developers.facebook.com/docs/plugins/like-button

Browser Fingerprinting

Cookies are not the only way to track users across sites.

A browser fingerprint is a set of information related to a user's device from the hardware to the operating system, to the browser and its configuration, and they can be used to uniquely identify users across websites.

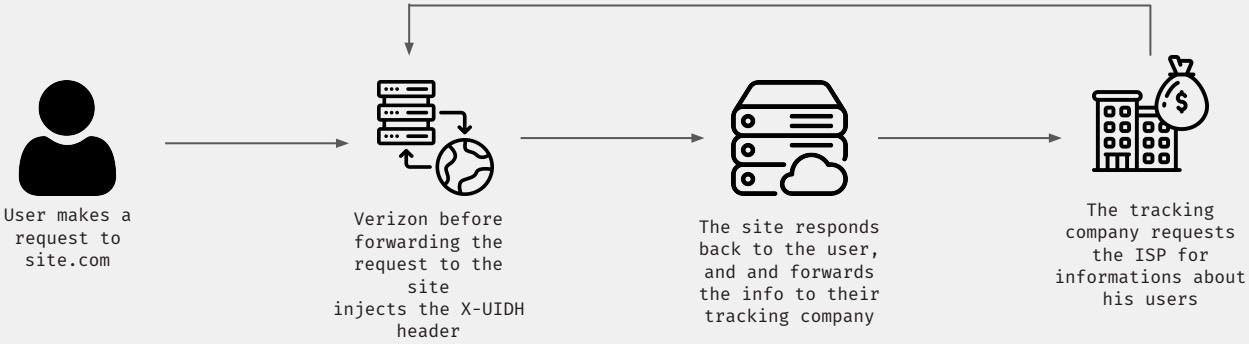
	Panopticlick (2010)	AmIUnique (2016)		Hiding in the Crowd (2018)	
	Desktop	Desktop	Mobile	Desktop	Mobile
Number of fingerprints	470,161	105,829	13,105	1,816,776	251,166
Unique fingerprints	94.2%	89.4%	81%	35.7%	18.5%

Attribute	Source
User agent	HTTP header
Accept	HTTP header
Content encoding	HTTP header
Content language	HTTP header
List of plugins	JavaScript
Cookies enabled	JavaScript
Use of local/session storage	JavaScript
Timezone	JavaScript
Screen resolution and color depth	JavaScript
List of fonts	Flash or JS
List of HTTP headers	HTTP headers
Platform	JavaScript
Do Not Track	JavaScript
Canvas	JavaScript
WebGL Vendor	JavaScript
WebGL Renderer	JavaScript
Use of an ad blocker	JavaScript

[9] Pierre Laperdrix, Nataliia Bielova, Benoit Baudry, and Gildas Avoine. 2020. Browser Fingerprinting: A Survey. ACM Trans. Web 14, 2, Article 8 (May 2020). https://doi.org/10.1145/3386040

Unique Identifier Header

Unique identifier headers (also called “Super Cookies”) are another way to track the browsing behaviour of users, in contrast to classical tracking techniques this is totally transparent to the user since is a modification applied by the ISP after the packet has left the user home network.

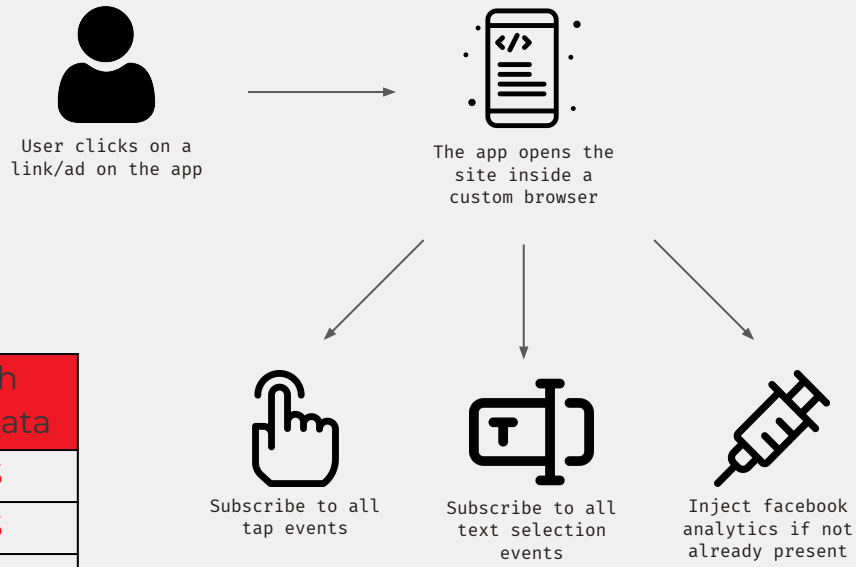


[10] https://web.archive.org/web/20230330035239/https://www.eff.org/it/deeplinks/2014/11/verizon-x-uidh
[11] https://web.archive.org/web/20230329132218/https://www.theverge.com/2016/3/7/11173010/verizon-supercookie-fine-1-3-million-fcc

In-App Browser JS hijacking

Companies can use In-App browser to track users outside their apps.

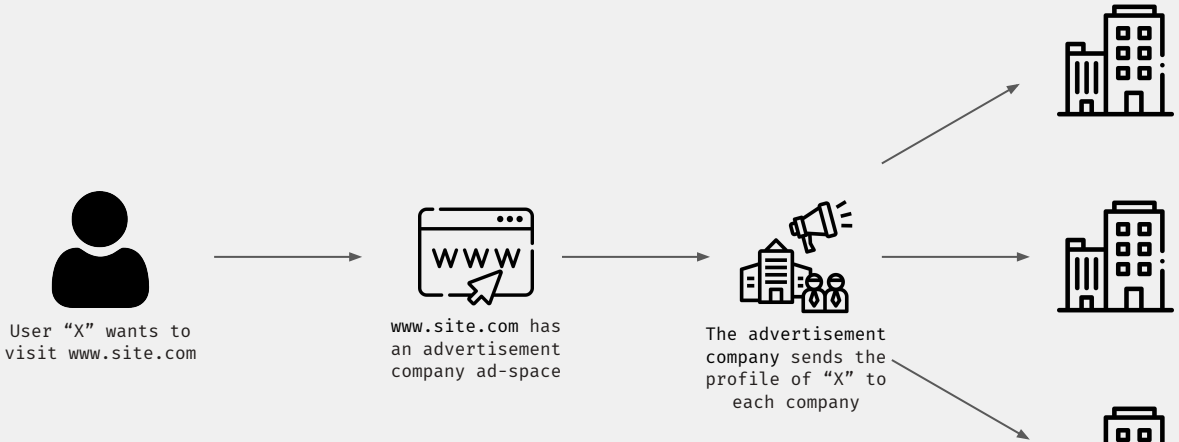
They inject custom javascript code inside each site their users open from their apps.



App	Option to open in default browser	Modify page	Fetch metadata
TikTok	NO	YES	YES
Instagram	YES	YES	YES
FB Messenger	YES	YES	YES
Facebook	YES	YES	YES
Amazon	YES	NO	YES

[12] https://web.archive.org/web/20230703192941/https://krausefx.com/blog/ios-privacy-instagram-and-facebook-can-track-anything-you-do-on-any-website-in-their-in-app-browser
[13] https://web.archive.org/web/20230703192955/https://krausefx.com/blog/announcing-inappbrowsercom-see-what-javascript-commands-get-executed-in-an-in-app-browser

Real Time Bidding



Every time an ad is going to be shown to a user an auction is created where multiple companies can review the profile of the user an decide how much to bid to show their ad to him

[14] https://web.archive.org/web/20230216172527/https://www.facebook.com/business/help/430291176997542
[15] https://web.archive.org/web/20230531205603/https://developers.google.com/authorized-buyers/rtb/start

```
// The Google ID for the user. This field is the unpadded web-safe base64
// encoded version of a binary cookie id. See the "Base 64 Encoding with URL
// and Filename Safe Alphabet" section in RFC 3548 for encoding details. Thi
// field may be the same as the Google ID returned by the cookie matching
// service. Not set if there is one or more user_data_treatment value.
optional string google_user_id = 21;
```

```
message Device {
  // The type of device on which the ad will be shown.
  enum DeviceType {
    UNKNOWN_DEVICE = 0;
    HIGHEND_PHONE = 1;
    TABLET = 2;

    // Desktop or laptop devices.
    PERSONAL_COMPUTER = 3;

    // Both connected TVs (that is, smart TVs) and connected devices
    // (such as Roku and Apple TV).
    CONNECTED_TV = 4;

    GAME_CONSOLE = 5;

    SET_TOP_BOX = 6;
  }
  optional DeviceType device_type = 1 [default = UNKNOWN_DEVICE];

  // The platform of the device. Examples: android, iphone, palm
  optional string platform = 2 [default = ""];

  // The brand of the device, for example, Nokia, Samsung.
  optional string brand = 3 [default = ""];

  // The model of the device, for example, N70, Galaxy.
  optional string model = 4 [default = ""];

  // Contains the OS version of the platform. For instance, for Android 2,
  // major=2, minor=0. For iPhone 3.3.1, major=3 and minor=3.
  message OsVersion {
    optional int32 major = 1 [default = -1];
    optional int32 minor = 2 [default = -1];
    optional int32 micro = 3 [default = -1];
  }
  // The OS version; for example, 2 for Android 2.1, or 3.3 for iOS 3.3.1.
  optional OsVersion os_version = 5;
```

```
// The identifier of the mobile app when this ad query comes from a mobile
// app, or from a mobile web page contained inside an app. If the app was
// downloaded from the Apple iTunes app store, then this is the app-store
// id, for example, 343200656. For Android devices, this is the fully
// qualified package name, for example, com.rovio.angrybirds. For Windows
// devices it's the App ID, for example,
// f15abcde-f6gh-47i0-j3k8-37193817mn3o. For SDK-less requests (mostly from
// connected TVs), the app ID provided by the publisher directly in the
// request.
optional string app_id = 6;
```

```
message UserAgent {
  // Identifies a device's browser or similar software component, and the
  // user agent's execution platform or operating system.
  message BrandVersion {
    // A brand identifier, for example, "Chrome" or "Windows". The value may
    // be sourced from the User-Agent Client Hints headers, representing
    // either the user agent brand (from the Sec-CH-UA-Full-Version header)
    // or the platform brand (from the Sec-CH-UA-Platform header).
    optional string brand = 1;
    // A sequence of version components, in descending hierarchical order
    // (major, minor, micro, ...).
    repeated string version = 2;
  }

  // Each BrandVersion object identifies a browser or similar software
  // component. Exchanges should send brands and versions derived from
  // the Sec-CH-UA-Full-Version-List header.
  repeated BrandVersion browsers = 8;
  // Identifies the user agent's execution platform / OS. Exchanges should
  // send a brand derived from the Sec-CH-UA-Platform header, and version
  // derived from the Sec-CH-UAPlatform-Version header.
  optional BrandVersion platform = 2;
  // True if the agent prefers a "mobile" version of the content if
  // available, meaning optimized for small screens or touch input. False
  // if the agent prefers the "desktop" or "full" content. Exchanges should
  // derive this value from the Sec-CH-UAMobile header.
  optional bool mobile = 3;
  // Device's major binary architecture, for example, "x86" or "arm".
  // Exchanges should retrieve this value from the Sec-CH-UA-Arch header.
  optional string architecture = 4;
  // Device's bitness, for example, "64" for 64-bit architecture. Exchanges
  // should retrieve this value from the Sec-CH-UA-Bitness header.
  optional string bitness = 9;
  // Device model. Exchanges should retrieve this value from the
  // Sec-CH-UAModel header.
  optional string model = 5;
}
```

```
// The user's approximate geographic location. All location information is
// IP geolocation-derived. The lat/lon fields may be a reference position
// (for example, centroid) for the IP geolocation-derived location that's also
// carried by the other fields (for example, a city), and accuracy will be the
// radius of a circle with the approximate area of that location. Location and
// its accuracy will be fuzzified as necessary to protect user privacy. See
// Geotargeting Guide:
// https://developers.google.com/authorized-buyers/rtb/geotargeting
message Geo {
  // Latitude from -90.0 to +90.0, where negative is south.
  optional double lat = 1;

  // Longitude from -180.0 to +180.0, where negative is west.
  optional double lon = 2;

  // Country using ISO-3166-1 Alpha-3.
  optional string country = 3;

  // Region code using ISO-3166-2; 2-letter state code if USA.
  optional string region = 4;

  // Google metro code; similar to but not exactly Nielsen DMAs.
  optional string metro = 6;

  // City using United Nations Code for Trade & Transport Locations.
  // (https://www.unece.org/cefact/locode/service/location.htm).
  optional string city = 7;

  // Zip/postal code.
  optional string zip = 8;

  // Estimated location accuracy in meters.
  optional int32 accuracy = 11;

  // Local time as the number +/- of minutes from UTC.
  optional int32 utcoffset = 10;
}
optional Geo geo = 62;
```

```
// The Google ID for the user. This field is the unpadded web-safe base64
// encoded version of a binary cookie id. See the "Base 64 Encoding with URL
// and Filename Safe Alphabet" section in RFC 3548 for encoding details. Thi
// field may be the same as the Google ID returned by the cookie matching
// service. Not set if there is one or more user_data_treatment value.
optional string google_user_id = 21;
```

```
message Device {
  // The type of device on which the ad will be shown.
  enum DeviceType {
    UNKNOWN_DEVICE = 0;
    HIGHEND_PHONE = 1;
    TABLET = 2;

    // Desktop or laptop devices.
    PERSONAL_COMPUTER = 3;

    // Both connected TVs (that is, smart TVs) and connected devices
    // (such as Roku and Apple TV).
    CONNECTED_TV = 4;

    GAME_CONSOLE = 5;

    SET_TOP_BOX = 6;
  }
  optional DeviceType device_type = 1 [defau

  // The platform of the device. Examples: a
  optional string platform = 2 [default = ""

  // The brand of the device, for example, N
  optional string brand = 3 [default = ""];

  // The model of the device, for example, N
  optional string model = 4 [default = ""];

  // Contains the OS version of the platform. For instance, for Android 2,
  // major=2, minor=0. For iPhone 3.3.1, major=3 and minor=3.
  message OsVersion {
    optional int32 major = 1 [default = -1];
    optional int32 minor = 2 [default = -1];
    optional int32 micro = 3 [default = -1];
  }
  // The OS version; for example, 2 for Android 2.1, or 3.3 for iOS 3.3.1.
  optional OsVersion os_version = 5;
```

```
// The identifier of the mobile app when this ad query comes from a mobile
// app, or from a mobile web page contained inside an app. If the app was
// downloaded from the Apple iTunes app store, then this is the app-store
// id, for example, 343200656. For Android devices, this is the fully
// qualified package name, for example, com.rovio.angrybirds. For Windows
// devices it's the App ID, for example,
// f15abcde-f6gh-47i0-j3k8-37193817mn3o. For SDK-less requests (mostly from
// connected TVs), the app ID provided by the publisher directly in the
// request.
optional string app_id = 6;
```

```
// The user's approximate geographic location. All location information is
// IP geolocation-derived. The lat/lon fields may be a reference position
// (for example, centroid) for the IP geolocation-derived location that's also
// carried by the other fields (for example, a city), and accuracy will be the
// radius of a circle with the approximate area of that location. Location and
// its accuracy will be fuzzified as necessary to protect user privacy. See
// Geotargeting Guide:
// https://developers.google.com/authorized-buyers/rtb/geotargeting
message Geo {
  // Latitude from -90.0 to +90.0, where negative is south.
  optional double lat = 1;

  // Longitude from -180.0 to +180.0, where negative is west.
  optional double lon = 2;

  // Country using ISO-3166-1 Alpha-3.
  optional string country = 3;

  // Region code using ISO-3166-2; 2-letter state code if USA.
  optional string region = 4;

  // Google metro code; similar to but not exactly Nielsen DMAs.
  optional string metro = 6;
```

AND MUCH MORE

```
  // A sequence of version components, in descending hierarchical order
  // (major, minor, micro, ...).
  repeated string version = 2;
}

// Each BrandVersion object identifies a browser or similar software
// component. Exchanges should send brands and versions derived from
// the Sec-CH-UA-Full-Version-List header.
repeated BrandVersion browsers = 8;
// Identifies the user agent's execution platform / OS. Exchanges should
// send a brand derived from the Sec-CH-UA-Platform header, and version
// derived from the Sec-CH-UAPlatform-Version header.
optional BrandVersion platform = 2;
// True if the agent prefers a "mobile" version of the content if
// available, meaning optimized for small screens or touch input. False
// if the agent prefers the "desktop" or "full" content. Exchanges should
// derive this value from the Sec-CH-UAMobile header.
optional bool mobile = 3;
// Device's major binary architecture, for example, "x86" or "arm".
// Exchanges should retrieve this value from the Sec-CH-UA-Arch header.
optional string architecture = 4;
// Device's bitness, for example, "64" for 64-bit architecture. Exchanges
// should retrieve this value from the Sec-CH-UA-Bitness header.
optional string bitness = 9;
// Device model. Exchanges should retrieve this value from the
// Sec-CH-UAModel header.
optional string model = 5;
}
```

```
  optional int32 accuracy = 11;

  // Local time as the number +/- of minutes from UTC.
  optional int32 utcoffset = 10;
}
optional Geo geo = 62;
```

Tracking in the real world

Persona	No Interaction	Interaction
Connected Car	0.364	0.311
Dating	0.519	0.297
Fashion & Style	0.572	0.404
Pets & Animals	0.492	0.373
Religion & Spirituality	0.477	0.231
Smart Home	0.452	0.349
Wine & Beverages	0.418	0.522
Health & Fitness	0.564	0.826
Navigation & Trip Planners	0.533	0.268
Vanilla	0.539	0.232

Table 7: Mean bid values without and with interaction across interest and vanilla personas that were collected close to each other.

IoT devices like Amazon Echo can track users using their voice data.

“Our results indicate that (i) Amazon Echo user interactions are tracked by both Amazon and third-parties, (ii) Amazon uses Amazon Echo interactions data for ad targeting on-platform (e.g., audio ads) and off-platform (e.g., web ads), and (iii) Amazon computed user interests from voice data in a way that was inconsistent with their public statements. In many instances, Amazon and skills did not clearly disclose their data collection practices in their privacy policies”[14]

[16] Iqbal, U., Bahrami, P. N., Trimananda, R., Cui, H., Gamero-Garrido, A., Dubois, D., Choffnes, D., Markopoulou, A., Roesner, F., & Shafiq, Z. (2022). Your Echos are Heard: Tracking, Profiling, and Ad Targeting in the Amazon Smart Speaker Ecosystem (Version 4). arXiv. https://doi.org/10.48550/ARXIV.2204.10920

What have all the case studies in common?

All the research is based on reverse engineering of products (network analysis, statistical analysis, decompilation/deobfuscation of software) and not based on the analysis of the privacy policy (also most of the time they also found out that the content was false).

So, how can a user make an informed decision?

What can we do about this?

Surveillance capitalism thrives thanks to total lack of transparency, information is buried under pages and pages of privacy policies and most of the time is not explicit in **what** and **how** information is harvested.

What can we do about this? As tech workers

When we develop a product we need to be transparent to our users.

- Use open source and privacy respecting alternatives to popular SDKs
  For example, Plausible Analytics[12] instead of Google Analytics

- Think if collecting data from your users is really useful for making our products better
  Do I need to know everything about the users of my product?

- Be clear on what and how data is collected
  Not buried in a privacy policy that virtually no one reads

- Consider the release of the source code of the product
  Nothing is as transparent as the source code

[17] https://web.archive.org/web/20230716201415/https://plausible.io/

What can we do about this? As users

We should be more mindful on what and how we use products made by companies that are known for selling user data.

- Consider user owned and privacy respecting alternatives
  For example, Mastodon[13] instead of famous social networks

- Consider alternative front ends to non privacy respecting services
  For example, Nitter[14] instead of Twitter or Invidious[15] instead of YouTube

- Use countermeasures to common tracking techniques
  For example, Ad blocking (uMatrix[17]), WebPage containerization (Containerise[18])

Clearly here there is still work to be done by us tech workers to make those solution more accessible to the average user.

[18] https://web.archive.org/web/20230629004333/https://joinmastodon.org/
[19] https://web.archive.org/web/20230630135747/https://nitter.net/about
[20] https://web.archive.org/web/20230627062643/https://invidious.io/
[21] https://web.archive.org/web/20230629194512/https://github.com/gorhill/uMatrix
[22] https://web.archive.org/web/20230630064801/https://github.com/kintesh/containerise/

Why should we care?

Even though professor Zuboff is right about the fact that this is a societal issue, we as tech workers are enabling these behaviours.

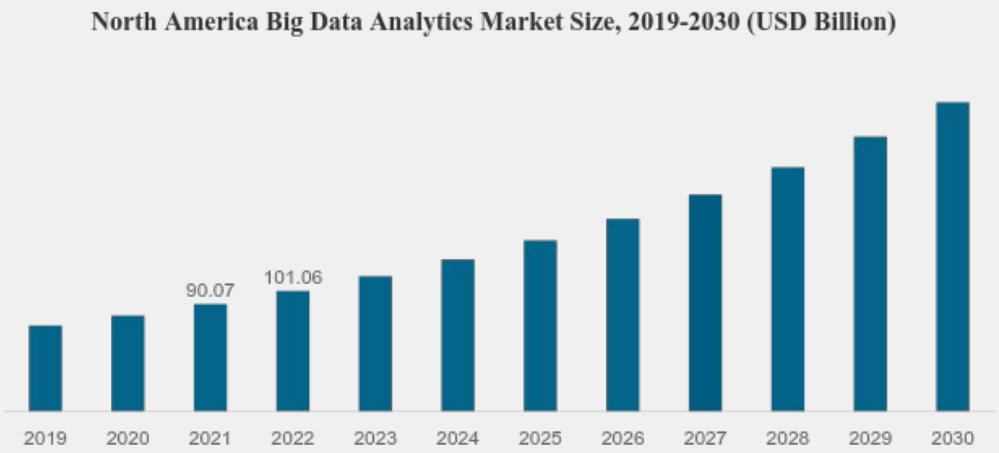
We should be aware of how the data that we collect and analyze with the products we develop can be used to impact the world in negative ways:

- Interfering with democracy[1]
- Unlawful policing[19]
- Exploiting their users data to their advantage[20]

[1] Zuboff, Shoshana. Big other: Surveillance Capitalism and the Prospects of an Information. Civilization Journal of Information Technology 30:1, 75-89 2015.
[23] https://web.archive.org/web/20230629055113/https://www.wired.com/story/fbi-purchase-location-data-wray-senate/
[24] http://web.archive.org/web/20230119194846/https://theconversation.com/could-your-fitbit-data-be-used-to-deny-you-health-insurance-72565

Future prospects

Clearly this phenomenon is not going to stop anytime soon the market was valued at USD 271.83 billion in 2022 and is projected to grow from USD 307.52 billion in 2023 to USD 745.15 billion by 2030.



[25] https://web.archive.org/web/20230512194853/https://www.fortunebusinessinsights.com/big-data-analytics-market-106179

Thanks for the attention!

Questions?