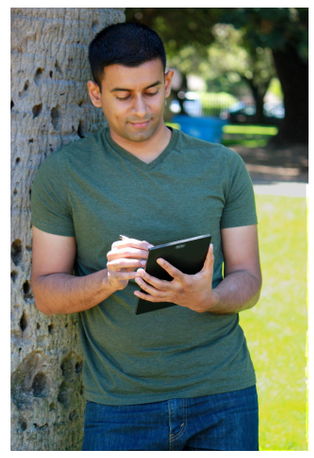
Battery Status Not Included: Assessing Privacy in Web Standards

Lukasz Olejnik
Independent Researcher
lukaszolejnik.com

Steven Englehardt
Princeton University
senglehardt.com

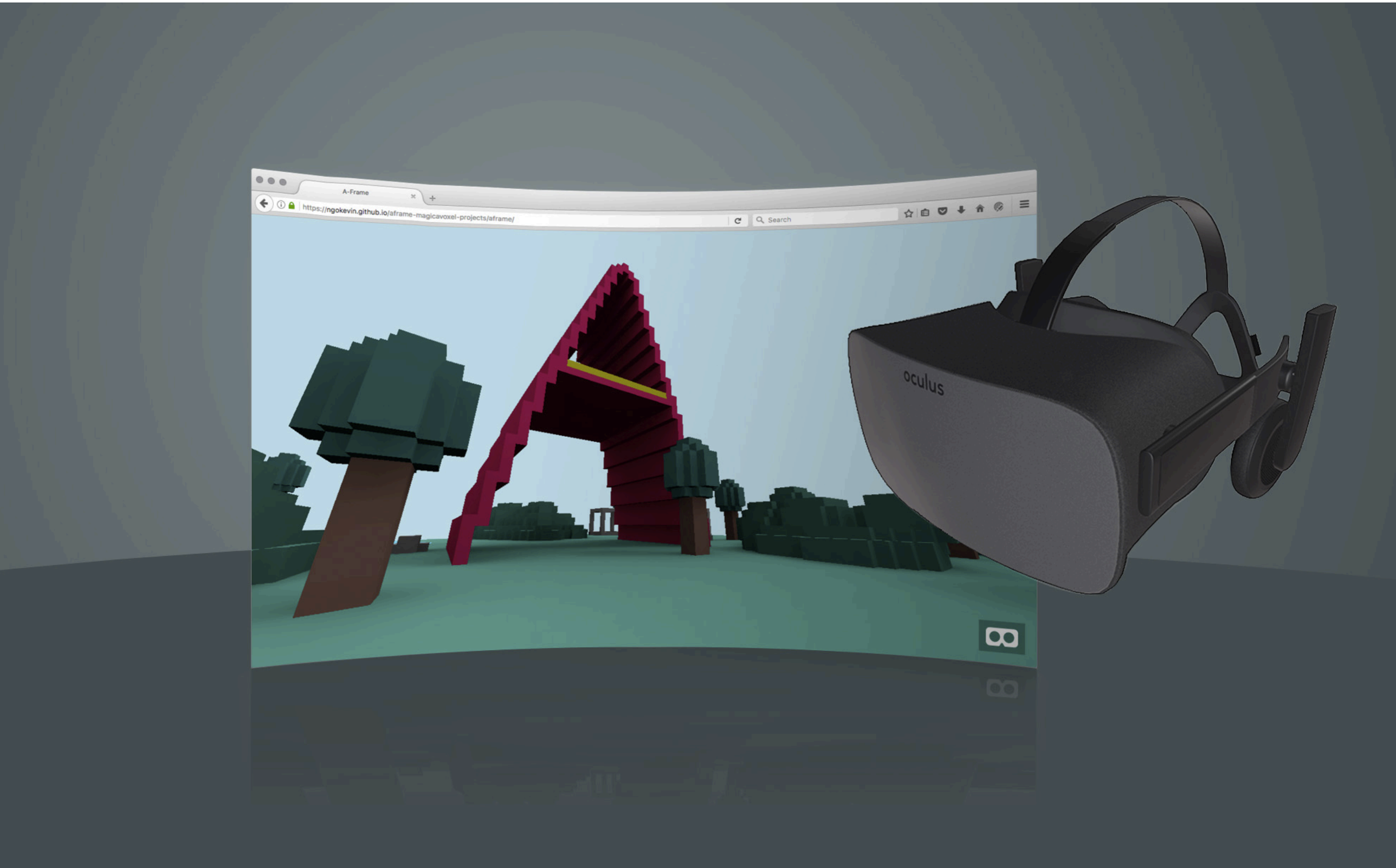


Arvind Narayanan
Princeton University
randomwalker.info

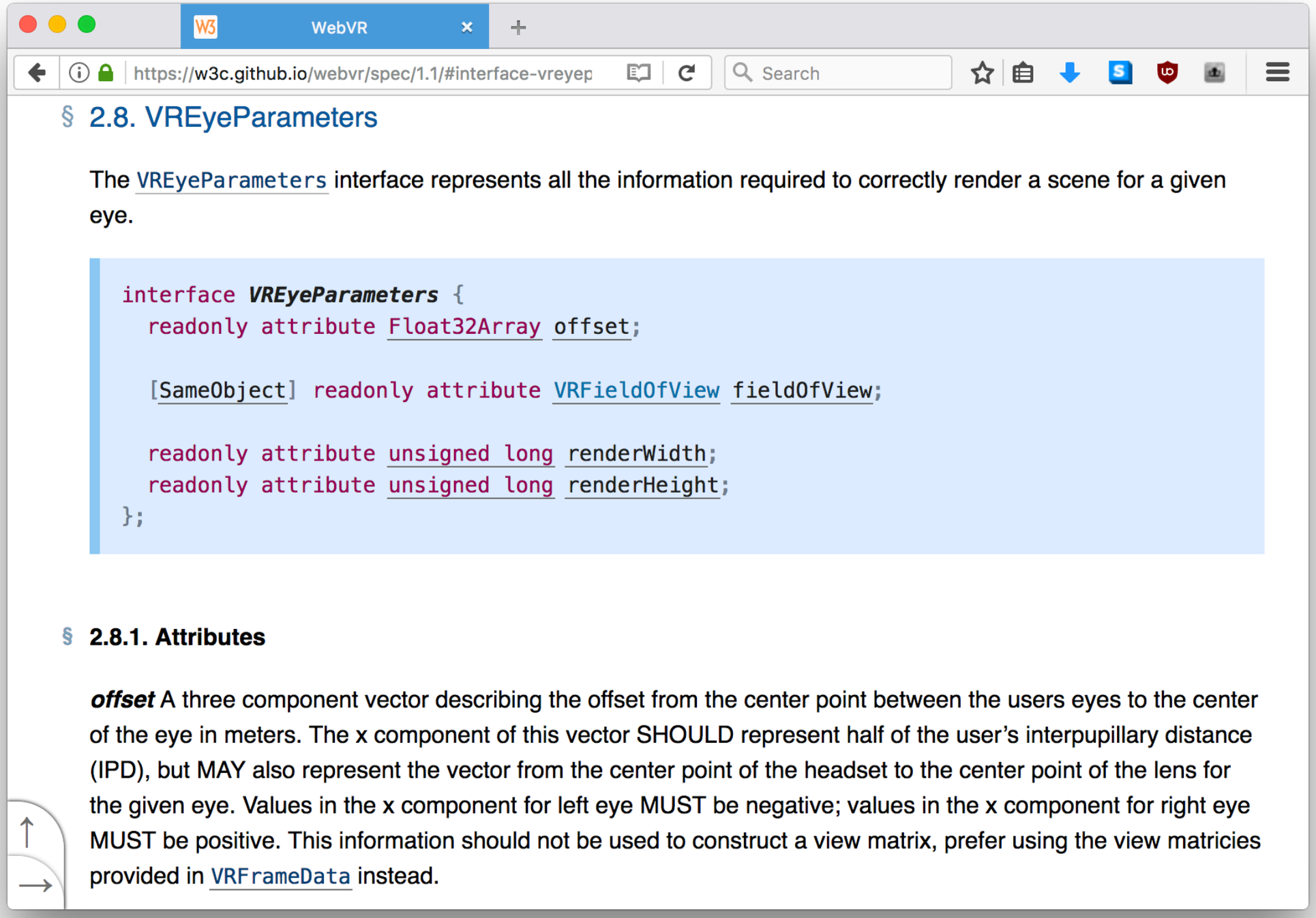
New Web features lead to privacy concerns

New Web features lead to privacy concerns

New Web features lead to privacy concerns



New Web features lead to privacy concerns



New Web features lead to privacy concerns

WebVR

https://w3c.github.io/webvr/spec/1.1/#interface-vreyep

Search

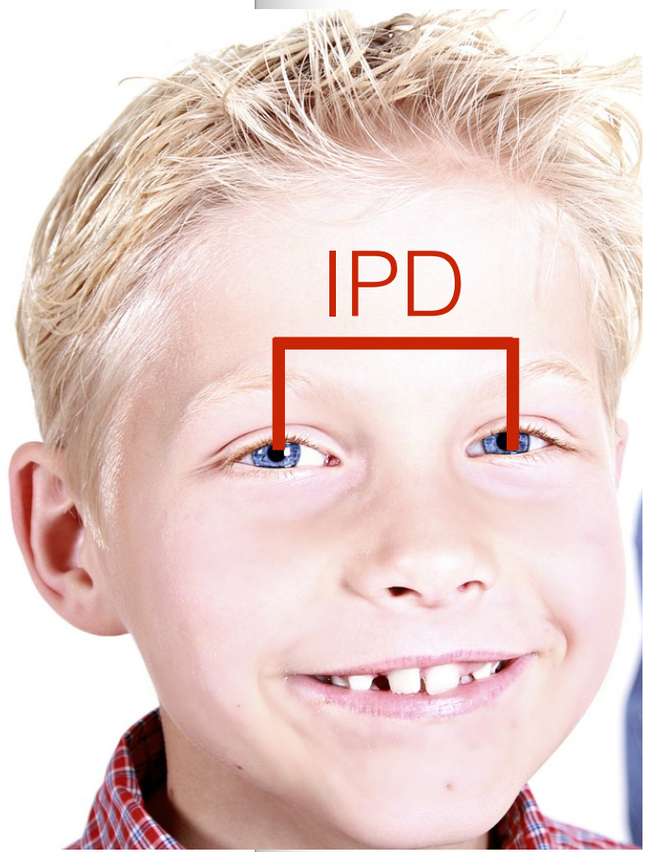
§ 2.8. VREyeParameters

The `VREyeParameters` interface represents all the information required to correctly render a scene f
eye.

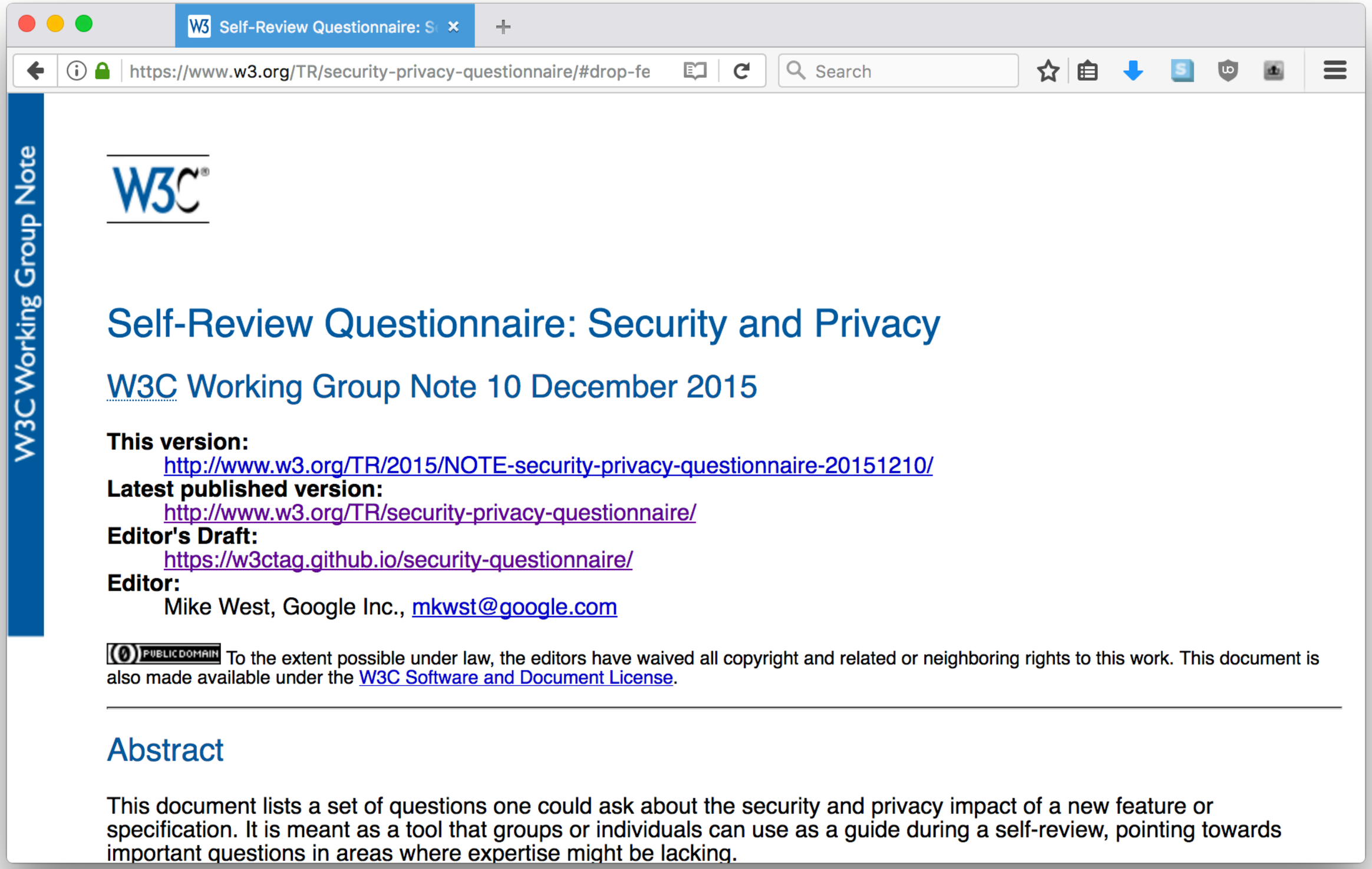
```
interface VREyeParameters {
  readonly attribute Float32Array offset;

  [SameObject] readonly attribute VRFieldOfView fieldOfView;

  readonly attribute unsigned long renderWidth;
  readonly attribute unsigned long renderHeight;
};
```

§ 2.8.1. Attributes

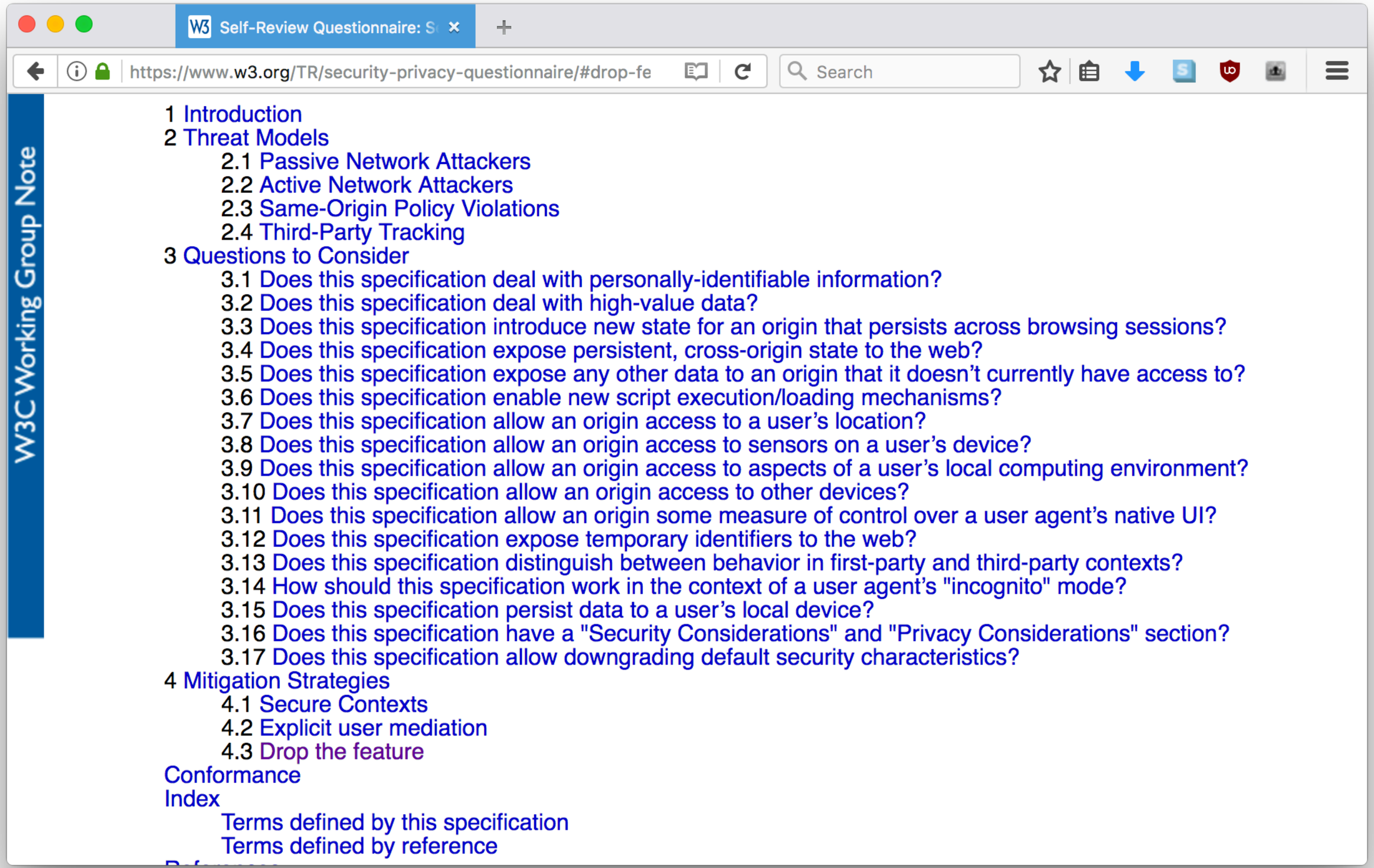
offset A three component vector describing the offset from the center point between the users eyes of the eye in meters. The x component of this vector SHOULD represent half of the user's interpupillary distance (IPD), but MAY also represent the vector from the center point of the headset to the center point of the lens for the given eye. Values in the x component for left eye MUST be negative; values in the x component for right eye MUST be positive. This information should not be used to construct a view matrix, prefer using the view matricies provided in `VRFrameData` instead.



The W3C has a self-review questionnaire



The W3C has a self-review questionnaire



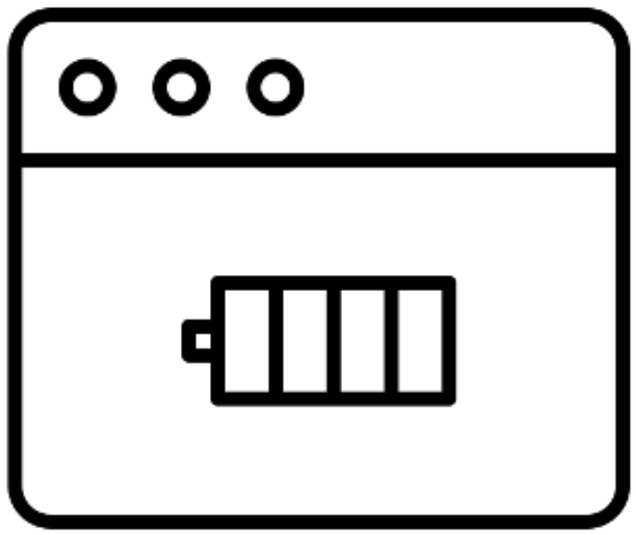
W3C Privacy Interest Group (PING) offers guidance and reviews

The mission…is to improve the support of privacy in Web standards by:

1. Monitoring ongoing privacy issues that affect the Web
2. Investigating potential areas for new privacy work
3. Providing guidelines and advice for addressing privacy in standards development.

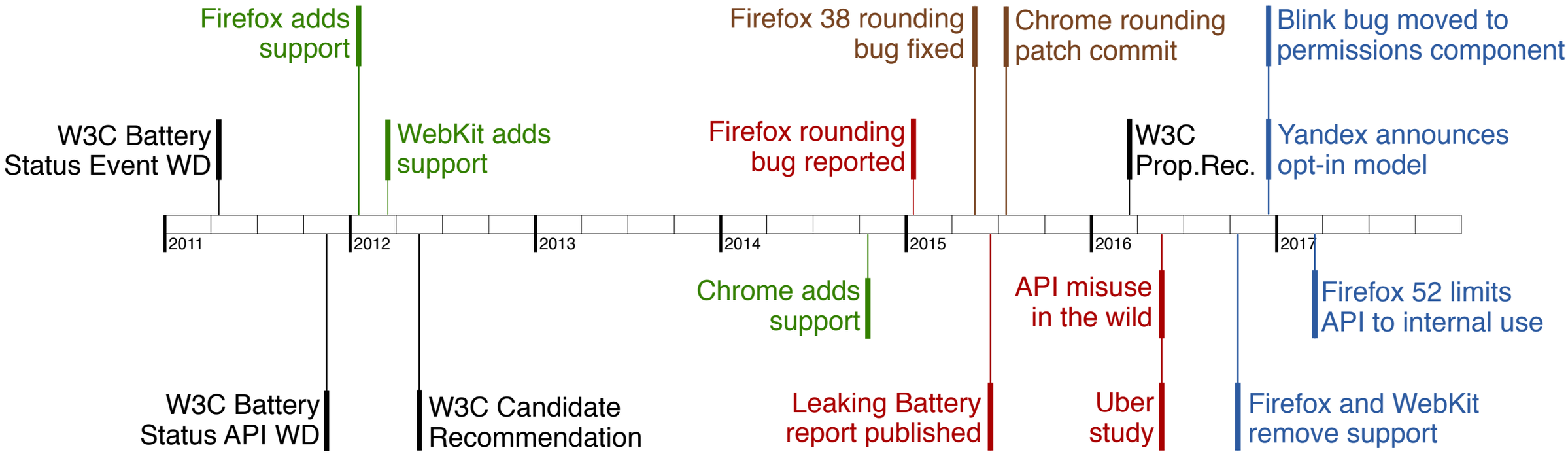
https://www.w3.org/2011/07/privacy-ig-charter

The Battery Status API

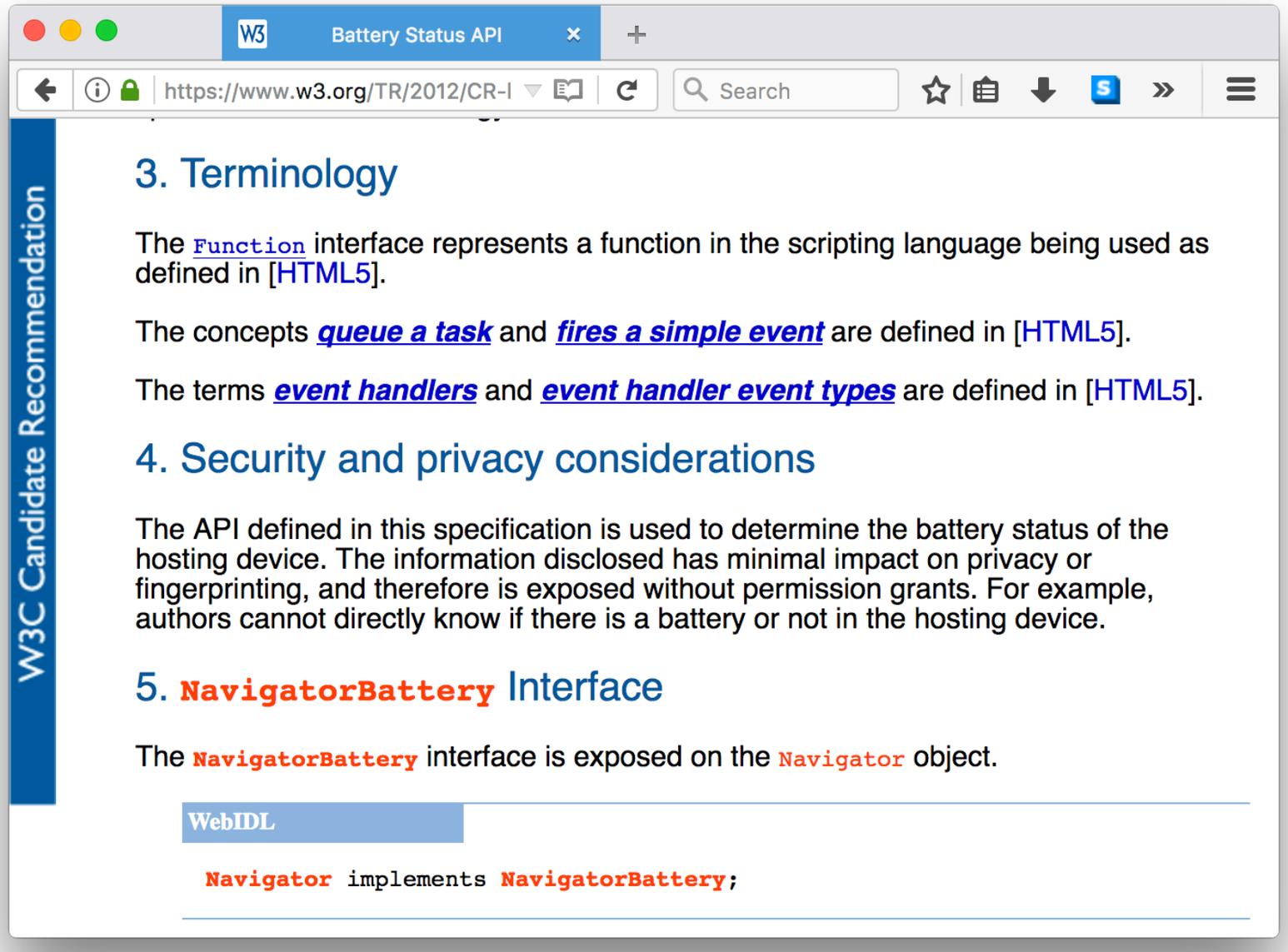


- charge **`level`**
  - value between 0 and 1
  - e.g 0.43 if the battery at 43%
- **`charging`** status
  - boolean indicator
- time to charge or discharge
  - **`dischargingTime`**
  - **`chargingTime`**
  - time in seconds

The development and adoption of the API

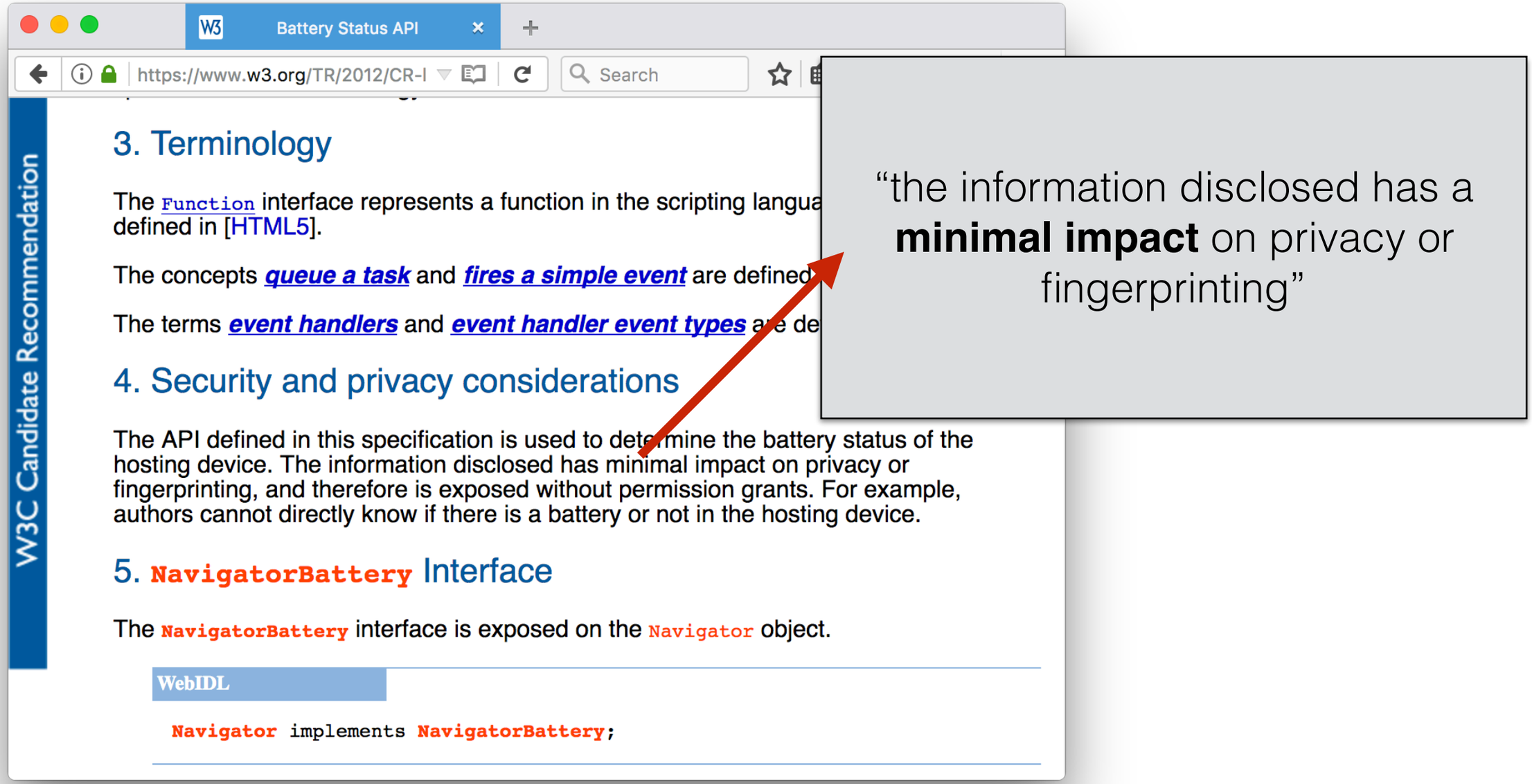


Mid 2012: Candidate Recommendation adds security and privacy considerations



https://www.w3.org/TR/2012/CR-battery-status-20120508/

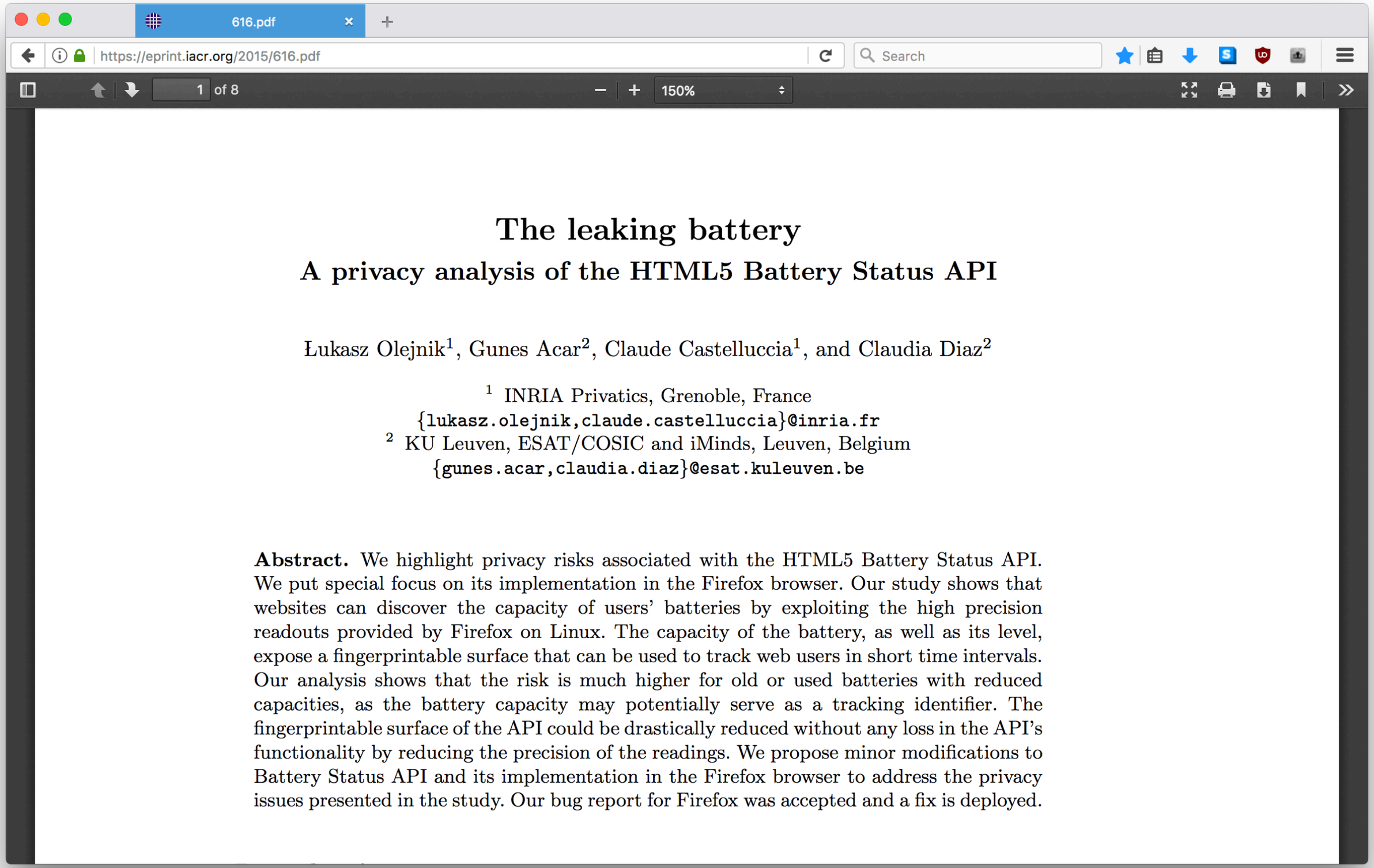
Mid 2012: Candidate Recommendation adds security and privacy considerations



"the information disclosed has a **minimal impact** on privacy or fingerprinting"

https://www.w3.org/TR/2012/CR-battery-status-20120508/

New research exposes multiple privacy vulnerabilities



616.pdf

https://eprint.iacr.org/2015/616.pdf

Search

1 of 8

150%

The leaking battery

A privacy analysis of the HTML5 Battery Status API

Łukasz Olejnik[1], Gunes Acar[2], Claude Castelluccia[1], and Claudia Diaz[2]

[1] INRIA Privatics, Grenoble, France
{lukasz.olejnik,claude.castelluccia}@inria.fr
[2] KU Leuven, ESAT/COSIC and iMinds, Leuven, Belgium
{gunes.acar,claudia.diaz}@esat.kuleuven.be

Abstract. We highlight privacy risks associated with the HTML5 Battery Status API. We put special focus on its implementation in the Firefox browser. Our study shows that websites can discover the capacity of users' batteries by exploiting the high precision readouts provided by Firefox on Linux. The capacity of the battery, as well as its level, expose a fingerprintable surface that can be used to track web users in short time intervals. Our analysis shows that the risk is much higher for old or used batteries with reduced capacities, as the battery capacity may potentially serve as a tracking identifier. The fingerprintable surface of the API could be drastically reduced without any loss in the API's functionality by reducing the precision of the readings. We propose minor modifications to Battery Status API and its implementation in the Firefox browser to address the privacy issues presented in the study. Our bug report for Firefox was accepted and a fix is deployed.

New research exposes multiple privacy vulnerabilities

ccs16_online_tracking.pdf

https://senglehardt.com/papers/ccs16_online_tracking.pdf

Search

1 of 20

Automatic Zoom

Online Tracking: A 1-million-site Measurement and Analysis

Steven Englehardt
Princeton University
ste@cs.princeton.edu

Arvind Narayanan
Princeton University
arvindn@cs.princeton.edu

This is an extended version of our paper that appeared at ACM CCS 2016.

ABSTRACT

We present the largest and most detailed measurement of online tracking conducted to date, based on a crawl of the top 1 million websites. We make 15 types of measurements on each site, including stateful (cookie-based) and stateless (fingerprinting-based) tracking, the effect of browser privacy tools, and the exchange of tracking data between different sites ("cookie syncing"). Our findings include multiple sophisticated fingerprinting techniques never before measured in the wild.

This measurement is made possible by our open-source web privacy measurement tool, OpenWPM[1], which uses an automated version of a full-fledged consumer browser. It supports parallelism for speed and scale, automatic recovery from failures of the underlying browser, and comprehensive browser instrumentation. We demonstrate our platform's strength in enabling researchers to rapidly detect, quantify, and characterize emerging online tracking behaviors

to resort to a stripped-down browser [31] (a limitation we explore in detail in Section 3.3). (2) We provide comprehensive instrumentation by expanding on the rich browser extension instrumentation of FourthParty [33], without requiring the researcher to write their own automation code. (3) We reduce duplication of work by providing a modular architecture to enable code re-use between studies.

Solving these problems is hard because the web is not designed for automation or instrumentation. Selenium,[2] the main tool for automated browsing through a full-fledged browser, is intended for developers to test their *own* websites. As a result it performs poorly on websites not controlled by the user and breaks frequently if used for large-scale measurements. Browsers themselves tend to suffer memory leaks over long sessions. In addition, *instrumenting* the browser to collect a variety of data for later analysis presents formidable challenges. For full coverage, we've found it necessary to have three separate measurement points:

New research exposes multiple privacy vulnerabilities

1. The Battery API could be used as a short-term identifier
2. Firefox was exposing high-precision values for charge level, which allowed battery capacity to be recovered
3. Scripts were abusing the API to track users in the wild

The specification was updated to address privacy vulnerabilities

1. Should avoid high precision readouts
2. Should inform the user when and who is using the API
3. May ask the user for permission
4. May obfuscate or expose fake values

Early 2017: Several vendors remove or restrict support, citing privacy and lack of use

Restricted to non-web content

Removed from source code

Open bug (filed under permissions?)

Opt-in, otherwise dummy values

Our data supports Mozilla's decision

We measured usage on the top 50,000 sites

33 third-parties on 815 sites use the API

- 16 used it for tracking
  - Mostly fingerprinting
- 8 used it for benign purposes
  - Mostly performance measurement
- 9 unclassified

How can we improve the privacy review process?

The specification process should include a privacy review of implementations

Specification requires two implementations to progress

The specification process should include a privacy review of implementations

Specification requires two implementations to progress

Why not require a privacy review of these specifications?

The specification process should include a privacy review of implementations

Specification requires two implementations to progress

Why not require a privacy review of these specifications?

—> Similar precision issues found during privacy review of Ambient Light Sensors API, which included implementation auditing.

API use in the wild should be audited after implementation

Trackers are the early adopters of any new API!

API use in the wild should be audited after implementation

Trackers are the early adopters of any new API!

- Niche adopters will use the API as intended
  - e.g. an online game using WebRTC for multiplayer
- One or two somewhat popular tracking scripts can push the malicious use to thousands of sites.
  - e.g. Canvas being used to fingerprint

API use in the wild should be audited after implementation

It’s not clear that the measurement community will continue to support fingerprinting measurement

API use in the wild should be audited after implementation

It’s not clear that the measurement community will continue to support fingerprinting measurement

Concerns:

1. Lack of novelty in measurement techniques
2. Measurement of each new API is a small contribution
3. Specifications can’t wait for the publication cycle

API use in the wild should be audited after implementation

It's not clear that the measurement community will continue to support fingerprinting measurement

Concerns:

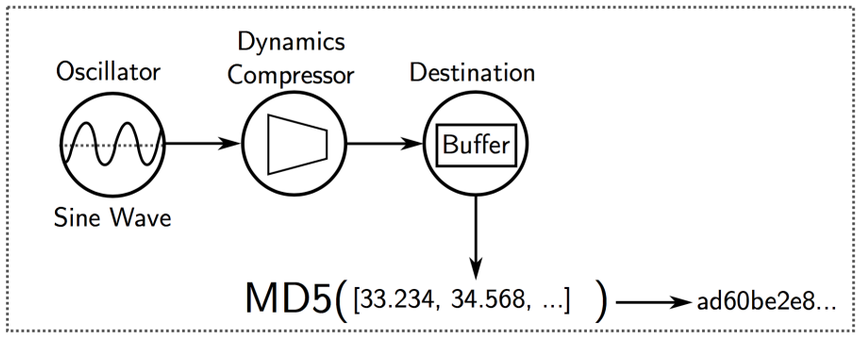
1. Lack of novelty in measurement techniques
2. Measurement of each new API is a small contribution
3. Specifications can't wait for the publication cycle

Suggestions:

1. Measurement through browser telemetry probes?
2. Regular measurement by browser vendors?
3. Public measurements by an NGO — something like archive.org

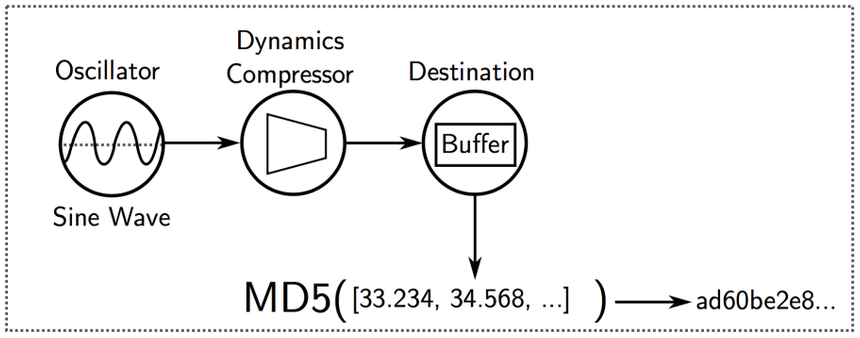
Specification authors should carry out privacy assessments with multiple threat models

An example: fingerprinting with the Audio API



Specification authors should carry out privacy assessments with multiple threat models

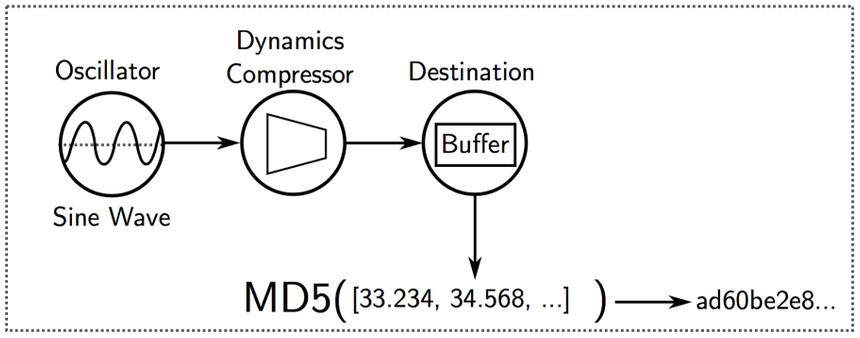
An example: fingerprinting with the Audio API



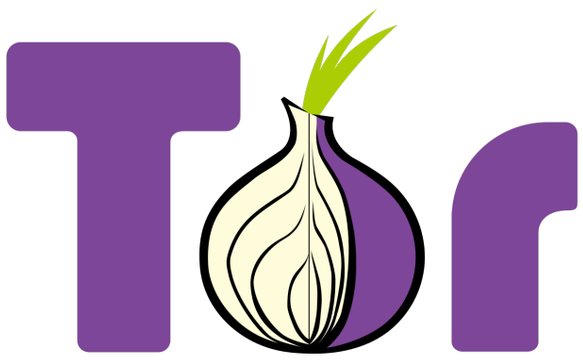
≈ User's OS and browser

Specification authors should carry out privacy assessments with multiple threat models

An example: fingerprinting with the Audio API



≈ User's OS and browser



This is a concern for the Tor Browser!

Thank you!

In summary:

1. Improve incentives for academics to contribute research
2. **Include audits of implementations in reviews**
3. **Audit API use after deployment**
4. **Carry out analysis in multiple threat models**
5. Avoiding over-specification supports innovative solutions
6. Provide guidance for web developers in addition to vendors

Full paper:
https://senglehardt.com/papers/iwpe17_battery_status_case_study.pdf

Image assets from the Noun Project: Browser Battery by Aybige