Web Bluetooth Origin Trial Results

The Web Bluetooth Origin Trial started in Chrome 53 and is scheduled to run through Chrome 55. (Intent to Experiment)

- Sept 7, 2016: Chrome 53 Stable.
- ~Nov 17: Chrome 56 Branch to Dev.
- ~Jan 31, 2017: Chrome 56 Stable

As of 2016-10-11, about 29 distinct sites have renewed their tokens. Star users include:

- https://kinomap.tv
- https://espruino.github.io/EspruinoWebIDE/
- http://cf.physical-web.org

Usage was self-reported as approximately:

- 1/5 commercial
- 1/5 personal
- 3/5 experimental

Developers rated the API about 3.5/5 for ease-of-use. They requested the following new features:

- Reading advertising data: https://crbug.com/654897 (v2)
    - RSSI (v2 within advertisements; WebBluetoothCG/web-bluetooth#311 for connections)
- Store permission to access devices: https://crbug.com/638721 (v2)
- Customizable device selection dialog: No current plans
- Possibility to scan for all nearby Bluetooth 4.0 devices, without specifying filters: https://crbug.com/654918 (v2)
- More debugging information to see what is/isn't working: https://crbug.com/651282 (in progress)
- Transmitting advertisement packets, specifically to interact with a smart mesh: https://github.com/WebBluetoothCG/web-bluetooth/milestone/4
- iOS and Windows support: https://crbug.com/655222 https://crbug.com/507419 (v2)

API Ergonomics:

- User gestures interact badly with complex event-handling code: https://crbug.com/404161 (spec change in progress)

Details have been drawn from the responses to the token-renewal form at https://docs.google.com/a/google.com/forms/d/1b2WxZDA4SmQdIJxz-veSptu5KxEFNfo6dZn9BTIc2KA/edit.

Questions from the Intent to Experiment:

- Have we selected a useful subset of Bluetooth functionality?
    - Some example sites that are using our subset successfully include https://kinomap.tv, https://espruino.github.io/EspruinoWebIDE/, and http://cf.physical-web.org.
- Is the API surface convenient for developers?
    - They rated it 3.5/5 for ease-of-use, and didn't suggest any particular improvements to the shape.
- What wrappers do people build to improve the API? Can we provide that interface directly?
    - Sites appear to be using the browser API directly.
- Do we need to implement the whole spec to ship, or can we usefully omit things like getDescriptor?
    - We've had requests that we should save permission across reloads and expose advertising data from paired devices, but enough folks are successfully using the API anyway that we think we should ship our current subset.
- What types of websites / device manufacturers are interested in the API?
    - It appears to be a wide variety: fitness, microcontroller development, beacon configuration, quadcopter control, etc.