What is Blink in 2017?

dglazkov@, with insights by others incorporated
2017-04-25

When we forked 4 years ago (happy belated birthday!), we thought of Blink as a rendering engine. The first sentence on the blink home page still says so. Over time, through the process of trial and error, and the evolution of how the Web is seen by developers (from documents to applications), we realized that the project is about something slightly bigger in scope -- the whole Web Platform[1] implementation. Today, Blink is definitely not just the rendering engine.

What the heck is it?

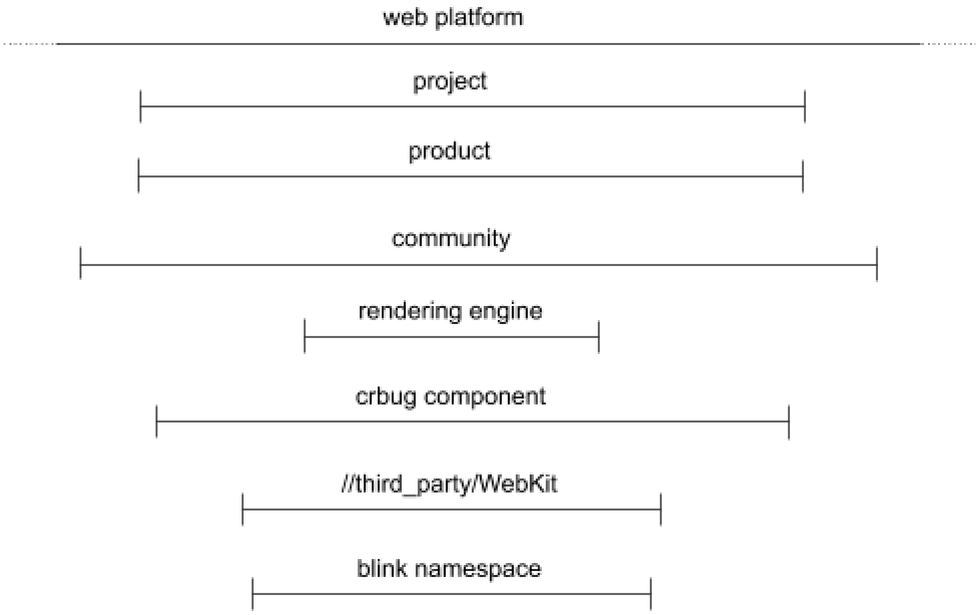
- An **open source project** that develops an implementation of Web Platform. This includes the process of making changes, focused on finding the right balance of forces in a difficult constraint space.
- The **product** that is the output of the project. The actual product is deeply interwoven across the Chromium codebase: large chunks of it are in //third_party/WebKit, as well as in /content/renderer, //media/blink, //content/browser/service_worker, and many other places.
- A **community**: a place for teams, companies and individuals in the common goal of building a better Web Platform -- and not just one particular implementation of it. The mailing list blink-dev is the asynchronous communication channel, and BlinkOn is our twice-yearly synchronous gathering. There’s also an inward-facing "group of people who work on this particular set of source files" aspect to this community, which includes discussions like "why doesn't my IDL file build" and "what changes are happening to WTF".
- A **component in crbug.com**, with sub-components that span bugs/features for //third_party/WebKit, V8[2], and other bits of Web Platform implementation.
- The C++ **namespace** for most code in //third_party/WebKit. The //third_party/WebKit, in itself, is neither the full implementation of Web Platform or just the rendering engine: roughly, it’s most of the rendering engine mixed with other code (not all Web Platform) that mostly runs in the renderer process.

[1] The Web Platform is the union of all technologies that are used to build the Web. It includes all APIs, languages, and concepts that the builders of the Web--the Web developers--use to create content that comprises the whole of the Web. There are multiple implementations of Web Platform and Blink develops just one of them.

[2] There is also a separate V8 bug tracker, and many Blink>JavaScript bugs are proxied there.

- Elsewhere in src/, a general directory **naming convention** for things related to a Web platform implementation (cc/blink, media/blink, etc.). In this particular case, “blink” doesn’t refer to Web Platform implementation per se, just the fact that these directories contain implementations of interfaces in //third_party/WebKit.

All these things are overlapping, but not all congruent (drawing not to scale, proportions -- aspirational, desire to compare unrelated things in one graph -- downright manic):



In summary:

1. **Blink is an open source project that develops an implementation of the Web Platform, and a community around this project**.

2. The relationship between the project and its output is unclear: it doesn’t seem like Blink is just a rendering engine, neither Blink crbug label nor //third_party/WebKit (along with the blink namespace) captures all of project activity.

I am worried because I need a bit more clarity: having the same name for slightly different things is a sign we all have subtly different understanding of what we’re working on.

I propose that we untangle the mess and **clarify the relationship** between Blink-the-project and its output. Clarifying the relationship would involve setting out a principle (a definition of what Blink is), and applying it **consistently**. Rather than changing the name by fiat, any principle we articulate about the definition of "Blink" should be guided primarily by observation of how people use the term in practice.

After searching for potential strategies, I can see two choices:

1. Keep Blink as strictly the name of the project and community, and not explicitly name the artifacts (the code) that the project/community produces. The principle is *Blink is the project and the community, not its artifacts*.
2. Call the project output "Blink" as well. Here, the principle is *Blink is an implementation of the Web Platform, with the project and the community that produces it.*

Both choices will require effort to implement, and the outcomes will look quite different, depending on which route we take.

If we decide to go with the first choice and consistently apply the underlying principle, we'll come up with a different name (or a set of names) for the things we currently call "blink" in code (blink namespace, crbug label, blink as the naming convention, etc.), and then replace everything named "blink" in with selected name(s). In this choice, //third_party/WebKit will move to something other than "blink" and perhaps even break apart into a bunch of smaller bits over time.

The consistent application of the second choice means that there will be a root "blink" directory, and we'll eventually move all code that comprises our Web platform implementation into that directory.

Weirdly, neither one of these choices leads to //third_party/WebKit moving to //blink: in "Blink is the project" case, there shouldn't be any artifacts named blink, and in "Blink is the code" case, //blink should host all of the browser process code alongside the renderer process code (which means that //third_party/WebKit moves to //blink/renderer or something like that).

Both options have pros and cons. The "Blink is the project" seems maddeningly pedantic: I can totally see peeps getting stumped by how is that there's no mention of Blink or a "blink" directory anywhere in the code base. The "Blink is the code" is ultimately unattainable: we are unlikely to ever neatly separate all of the Web Platform implementation bits, and thus, there will always be "is it Web Platform or is it Blink (and does my work need to go through the intent process)" tension that we'll need to compensate for.

The thing is that now, we *have to* choose :) The venerable //third_party/WebKit dir has served us well and is a proud reminder of our heritage, but it's time for it to go. Let's figure out where.

Discussion

Add your thoughts here.

Blink is the project and the community, not its artifacts

Pros:

- ?

Cons:

- ?

Blink is an implementation of the Web Platform, with the project and the community that produces it

Pros:

- ?

Cons:

- ?