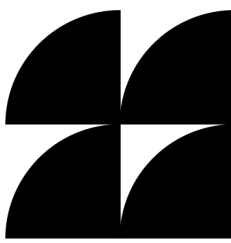
Knight Georgetown Institute

JULY 1, 2025

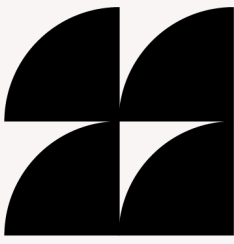
The Technical Feasibility of Divesting Google Chrome

Eric Rescorla
Alissa Cooper
Knight-Georgetown Institute



About the Knight-Georgetown Institute

The Knight-Georgetown Institute (KGI) is dedicated to connecting independent research with technology policy and design. KGI serves as a central hub for the growing network of scholarship that seeks to shape how technology is used to produce, disseminate, and access information. KGI is designed to provide practical resources that policymakers, journalists, and private and public sector leaders can use to tackle information and technology issues in real time. Georgetown University and the Knight Foundation came together to launch the institute in 2024. Learn more about KGI at https://kgi.georgetown.edu.



Executive Summary

In the *US v. Google* search antitrust litigation, the U.S. Department of Justice and State Attorneys General have proposed that Google divest its Chrome web browser as a remedy for Google's illegal monopoly in online search. This report provides a comprehensive technical assessment of whether such a divestiture is feasible, examining whether a non-Google entity ("ChromeCo") could successfully operate Chrome ("NewChrome") together with its underlying open-source code base, Chromium, to serve the existing four billion Chrome users with a browser that is competitive with today's other major browsers (Microsoft Edge, Apple Safari, and Mozilla Firefox). The report concludes that:

The divestiture of Chrome is technically feasible. Under reasonable assumptions, the engineering required to deliver a competitive NewChrome browser is achievable on a limited timeline, any necessary support from Google can be compelled by court order, and ChromeCo could reasonably expect to be able to retain or recruit the personnel needed to maintain and improve the browser.

A divested Chrome can operate effectively without relying on Google's proprietary technology. Existing Chromium-based browsers such as Microsoft Edge and Brave already offer competitive alternatives without access to Google's internal services or infrastructure. ChromeCo would need to replace or replicate proprietary components (related to audio and video) and numerous proprietary services. Each of these functions has either an available licensing model, proven alternatives in the market, or a viable engineering path forward. In cases such as Safe Browsing and data syncing, transitional support from Google (possibly by court order) could ensure continuity of operation.

A non-Google entity can maintain and develop Chromium independently. The open source Chromium code base is already accessible to the public, and organizations such as Mozilla and Apple have long maintained competitive browser engines without access to Google's internal tools or engineering teams. While Google currently contributes the majority of the code to Chromium, other vendors and independent projects demonstrate that browsers can thrive with smaller teams and more collaborative governance models. ChromeCo would need to recruit or retain browser engineering talent, but the relevant expertise exists both inside and outside Google.

A smooth transition from Google is possible, particularly if the court requires Google to assist. Software update mechanisms, data import from Google accounts, and continued operation of Safe Browsing and other services could all be managed through a combination of technical solutions and court-ordered cooperation from Google. The court would need to ensure that Google provides the relevant technical and organizational documentation and software source code to bootstrap ChromeCo's operations. These steps are within the norms of large-scale software transitions and would allow Chrome's four billion users to move to NewChrome without significant disruption.

A successful divestiture depends on the court imposing clear guardrails that prevent Google from undermining the remedy and that promote browser competition. Strong line-of-business

restrictions prohibiting Google from reentering the browser market for at least 5-10 years are needed to ensure that ChromeCo can build the technical and business foundations to succeed. To help preserve a competitive browser ecosystem, the court should also require that ChromeCo maintain Chromium under an open source model with new governance. Google must relinquish unilateral control of Chromium updates. These provisions are essential to ensure that the remedy delivers meaningful structural change and protects the public interest.

The browser market already demonstrates the feasibility of independent browser operation. Microsoft Edge, Brave, Opera, and other Chromium-based browsers successfully compete using combinations of open-source Chromium code, independent proprietary services, and third-party infrastructure. Mozilla Firefox and Apple Safari operate entirely independently with different browser engines and have driven major web innovations despite significantly fewer resources than Google.

In short, divesting Chrome from Google is a technically achievable remedy. Provided that the divestiture is structured with appropriate court-ordered cooperation and sufficient transition time, it is technically feasible for ChromeCo to deliver a competitive browser at global scale.

Table of Contents

I. Introduction

In the *US v. Google* search antitrust litigation, the U.S. Department of Justice and State Attorneys General have proposed a wide-ranging set of remedies to address Google's illegal monopoly in online search.[1] Perhaps none of these has attracted as much attention as the plaintiffs' proposal for Google to divest its web browser, Chrome. During the three-week remedies trial that took place in April and May 2025, the question of whether such a divestiture would be technically feasible arose repeatedly. The confines of trial testimony and evidence provide a limited window for analyzing this question, but not a complete treatment of the issue.

To close that gap, this report provides an in-depth assessment of whether it is technically feasible for Google to divest Chrome and its underlying open source software project, Chromium, to a non-Google entity (hereafter referred to as "ChromeCo") such that Chrome's existing four billion users can continue to make use of a browser (hereafter referred to as "NewChrome") competitive with today's other major browsers (Microsoft Edge, Apple Safari, and Mozilla Firefox). In this context, "feasibility" means:

- The software engineering required to deliver the competitive NewChrome browser is possible on a reasonable time frame post-divestiture;
- The technical assistance that ChromeCo would need from Google to execute the transition successfully can reasonably be ordered by the court; and
- ChromeCo can reasonably expect to be able to retain or recruit the personnel necessary to maintain NewChrome.

This report explicitly does not address the question of whether ChromeCo can duplicate and maintain Chrome's entire current feature set and corresponding services infrastructure resulting from choices Google has made according to its own business interests. For ChromeCo, the objective of acquiring Chrome would most likely not be to continue to provide exactly the product that Google would have provided in the market absent the divestiture, but rather to make NewChrome successful without the need to conform to Google's broader business strategy. A successful divestiture requires ChromeCo to smoothly transition Chrome's existing four billion users during the period of Chrome's ownership transfer, and to continue to offer a competitive browser into the future, but that browser need not behave identically to Chrome as it currently exists.

The report concludes that the divestiture of Chrome is technically feasible. A successful divestiture would require: (1) ChromeCo to replace the Google-proprietary technology currently in Chrome; (2) ChromeCo to maintain or acquire the personnel, collaboration, and internal software tooling necessary to support a competitive browser; (3) Chromium's governance to be established independent of Google; and (4) Google to assist in the transition of key services to ChromeCo, such as software updates and data syncing. In most of these cases, other browsers on the market today demonstrate how these requirements can be met by independent entities; in the rest of the cases, it would be either unnecessary or counterproductive for ChromeCo to copy Google's approach.

[1] United States et al., "Plaintiffs' Revised Proposed Final Judgment."

This report focuses on technical feasibility, as the business case for a divested Chrome has been made elsewhere.[2] The report assumes that if the divestiture is ordered by the court, Google would be prevented from introducing its own browser into the market for a period of 5-10 years at a minimum. Considering the investment that would be required from both Google and ChromeCo to transition Chrome to its new owner, and the expectation that Chromium would remain open source (and therefore relatively straightforward for Google to use to create a new browser, if it were allowed to do so), this is an important assumption when analyzing technical feasibility.

The report is based on extensive knowledge of web browsers and web technologies as well as analysis of source code in Chrome, Brave, Firefox, and other browsers.[3] While this analysis has been conducted in the context of the *US v. Google* search case, it is equally relevant in other jurisdictions or cases where the divestiture of Chrome may be considered as a result of competition regulation or enforcement.

The report is organized as follows: Section II provides technical background about web browsers; Section III provides the feasibility assessment; Section IV discusses transition planning; Section V summarizes the preconditions for a successful divestiture; Section VI summarizes what the court's order needs to include; and Section VII concludes.

II. Web Browser Technical Background

A. Browser Engines

A modern web browser is a complicated piece of software. Chromium, the open source project on which Chrome is based, contains over 35 million lines of code, and it has been developed over a span of more than 15 years.[4] Other browsers, such as Firefox and Safari, are of similar size and complexity.

Browsers typically have two components:

- a **browser engine**, the core software component inside the browser. The browser engine transforms web source code (HTML, CSS, JavaScript) into web pages or web applications (web apps) that users see and interact with.

- a branded **user interface (UI)**, which is responsible for user-facing functionality. The browser UI gives the browser its look and feel, layout, navigation bar, and settings. The default search

[2] Cooper, "The True Cost of Browser Innovation"; Montoya, "It's Time to Imagine Chrome Without Google."
[3] One of the report's authors, Eric Rescorla, spent over 10 years working on the Firefox browser, including five years as Firefox's Chief Technology Officer. Eric has also contributed extensively to the development of the web platform and web standards, and he has written software code that is included in Chrome.
[4] Chrome itself was launched in 2008, but the Blink browser engine is a descendant of an older engine, KHTML, which was first developed 10 years prior.

engine, password manager, history syncing, and translation are all surfaced to the user via the UI.

The bulk of a browser's code resides in the browser engine, which is responsible for key functionality that consumers use to navigate the internet: HTML rendering, running JavaScript code provided by web sites, networking, encryption of traffic, management of cookies, real-time audio and video processing, and much more. The browser engine is sometimes referred to as the "back end" while the UI is referred to as the "front end."

Most of the cost of creating a browser comes from building and maintaining the browser engine. However, most of what the engine does is invisible to the user, or depends on how the website that the user is visiting is designed. As a result, there are limited opportunities for competing browsers to differentiate in the engine itself, although browsers do compete on performance, security, and support for new web features. Many of the marquee features in each browser are front-end features.

As of this writing, there are three major browser engines:[5]

1. **Blink**, which is used in Chrome and primarily developed by Google.
2. **WebKit**, which is used in Safari and primarily developed by Apple.
3. **Gecko**, which is used in Firefox and primarily developed by Mozilla.

All commercial browsers also include a JavaScript virtual machine (VM), a software environment that can execute the same JavaScript code across different operating systems. For historical and organizational reasons, the VM is often a separate software project from the browser engine itself, but as a practical matter, each major engine only works with a single VM: Blink with V8, Gecko with SpiderMonkey, and WebKit with JavaScriptCore.

Together, the browser engine and front end comprise the browser. Many browsers have more than one front end: for example, Firefox on Android and Firefox on desktop use different front ends. Furthermore, some engines are used by multiple third-party browser vendors, and some browsers use different engines on different platforms. For example, many non-Google browsers are built on Blink, most notably Microsoft Edge. In addition, outside of the European Union, Apple requires that all browsers on iPhone and iPad use the built-in WebKit browser engine.[6] As a result of this policy, the iOS version of Firefox and Chrome both use WebKit, rather than Gecko and Blink, respectively.[7]

[5] All three of these engines are released under open source licenses.
[6] Apple, "Using alternative browser engines in the European Union."
[7] As of this writing, neither Mozilla nor Google has provided a non-WebKit browser on iOS, even in Europe.

B. Chrome, Chromium, and Chromium-Based Browsers

On most platforms (except iOS), Chrome combines the Blink browser engine, the V8 JavaScript VM,[8] a number of other open source components, and a front end into a single open source project that is collectively called Chromium.[9] Chromium is fully open source. Any individual or company can download and build its own copy of Chromium, which is itself a fully functional web browser. Google leads the development of Chromium, but others also contribute, including Microsoft, Intel, and Samsung. Google ultimately retains control of the governance of the project and determines what changes to accept or reject.[10]

Chrome is built on top of Chromium. Google takes the open source Chromium and incorporates two kinds of additions to create Chrome:

- **Proprietary components:** These include the Google-owned Widevine digital rights management (DRM) module,[11] which supports playing DRM-protected media from common commercial streaming services, and software components for encoding and decoding audio and video.

- **Proprietary services**: These include Google's identity management system (known as Gaia), Safe Browsing, software updating, crash reporting, translation, and other services. In many cases, the actual code to use these services is embedded in Chromium, so that the interface to the services is not proprietary. However, the services may require each browser instance to provide authentication information (known as "API keys") that gives the browser access to these services. Google embeds this information directly into Chrome. Third-party Chromium-based browsers that want to use these services would need to make arrangements with Google to obtain their own API keys.

Chrome is the most widely used of the Chromium-based browsers.[12] The vast majority of third-party browsers are based on Chromium (using not just the Blink engine, but the whole software project). These include Microsoft Edge, Brave, Amazon Silk, Opera, Arc, Samsung Internet, UC Browser, Vivaldi, Ecosia, and others.[13]

In order to differentiate, each vendor of a Chromium-based browser adds its own code. Because Google regularly changes the Chromium source code, these vendors must regularly update to the latest version of the Chromium source code, both in order to get new features and to ensure that they

[8] The remainder of this document follows conventional industry practice and references "Blink" or "Chromium" as appropriate rather than "Blink/V8".
[9] Chromium, "The Chromium Projects." Chromium consists of a number of source code repositories that are combined during the process of building the executable version of the browser. By contrast, other browsers such as Firefox use what is known as a "monorepo" in which all the source code lives in the same place. This is an important detail for engineers but not relevant to this report's analysis.
[10] Chromium, "Contributing to Chromium."
[11] Widevine, "Widevine."
[12] CloudFlare Data Insights Team, "Browser Market Share Report for 2024 Q1."
[13] There are over two dozen Chromium-based browsers on the market today. Google, "Defendant's Proposed Findings of Fact," 50.

receive any security fixes made to Chromium. This limits the extent of the differences in functionality that a vendor can maintain in its own Chromium-based browser because any changes the vendor makes to the core of Chromium must be reconciled with any conflicting changes Google makes. As a result, Chromium-based browsers largely have similar behavior to Chromium, with the differences typically limited to adding features to the front end, disabling features, or replacing Google's proprietary services with alternatives.

III. Feasibility Assessment

Although the technical feasibility of the Chrome divestiture was discussed at numerous points during the remedies trial, that discussion was not grounded in a precise definition of what it means for the divestiture to be feasible. At least four different versions of the feasibility question were implied at various points during the trial:

1. Is it feasible for a non-Google entity to maintain a browser which is substantively the same as the current Google-provided Chrome, including Google's proprietary technology?
2. Is it feasible for a non-Google entity to maintain a competitive browser based on Chromium, even if that browser is not precisely the same as the current Google-provided Chrome?
3. Is it feasible for a non-Google entity to maintain a browser which is functionally on par with the current Google-provided Chrome, without access to Google's proprietary technology?
4. Is it feasible for a non-Google entity to maintain a competitive browser based on Chromium without Google continuing to invest in Chromium?

As noted in Section I, the first question is irrelevant to the divestiture feasibility assessment, because prospective buyers are not likely to be interested in purchasing Chrome only to continue to operate it in exactly the manner that Google would have.

The answer to the second question is plainly yes, and the proof lies in the browser market. There are many successful Chromium-based browsers, including Edge and Brave.

That leaves the final two questions. As noted in Section I, feasibility must be tested against whether ChromeCo can offer a browser that is competitive with the four major browsers (Chrome, Edge, Safari, and Firefox) competing today. The analysis in this section centers on this feasibility question, and the related market reality that Google's investments in Chromium will very likely diminish post-divestiture.

This analysis assumes that if the court orders a divestiture, the order would also take the necessary step of preventing Google from offering its own browser in the market for 5-10 years or longer if clear benchmarks of search market competitiveness have not been reached. The time window is an important factor that will shape ChromeCo's incentives to purchase, maintain, and build NewChrome and Chromium. The time window needs to allow for the technical transition and for ChromeCo to build up its new business to the level necessary to maintain Chrome. With Google's strong market position across Android, ads, YouTube, cloud, web services, and artificial intelligence (AI), it will have an

incentive to reenter the browser market at its first opportunity. Assuming that Chromium remains open source, it would be relatively straightforward for Google to build its own browser in the future and position it to become an immediate strong competitor to NewChrome. ChromeCo will need sufficient time to transition the Chrome user base and build a business from its new product while the search market sees increased competition over the term of the remedy.

This section proceeds in four parts that cover: (1) operating Chrome without Google's proprietary technology; (2) operating Chrome if Google significantly reduces investments in Chromium; (3) transitioning Chromium's governance under new ownership; and (4) Google's other Chromium-dependent products, ChromeOS and WebView.

A. Replacing Google's Proprietary Technology

The Google-proprietary technology present in Chrome falls into two categories: proprietary components and proprietary services.

1. Proprietary Components

While Chrome currently includes a number of Google's proprietary software components, ample evidence from existing non-Google browsers demonstrates that ChromeCo could incorporate these or equivalent components without issue.

The most notable Chrome-specific component is the Widevine content decryption module which supports playing DRM-protected media from common video streaming services such as Netflix or YouTube. Streaming video is a very common browsing activity, so this is an essential capability for a browser. Although Widevine is owned by Google, Google provides the Widevine component to other browsers on a no-fee licensing basis, thus allowing users of those browsers to view DRM-protected content on YouTube and other services.[14] Many other browsers currently use Widevine, including Firefox and Brave, as well as non-browser devices such as Roku and Sony Playstation. ChromeCo would need to obtain a Widevine license from Google in order to use this component, and the potential divestiture order should ensure that this is possible.

The second major class of Chrome-specific components is patent-encumbered software that encodes and decodes audio and video (known as "codecs"). In general, source code for these codecs is widely available, and at least some of them are included with Chromium, but they are disabled by default. Enabling these components requires the vendor of any Chromium-based browser to obtain appropriate patent licenses. ChromeCo would need to either pay the license fees to the patent owners (as many other hardware and software vendors already do) or make other arrangements. Mozilla, for example, does not pay patent license fees for codecs, but still provides many patent-encumbered codecs in Firefox by using components provided by the underlying operating system or by third parties.[15]

[14] See Widevine, "Widevine."
[15] Mozilla, "Why is there an OpenH264 plugin in Firefox?".

The final major class of Chrome-specific components is those which are designed to interact with Google proprietary services. These are discussed together with Google's proprietary services in the next section.

2. Proprietary Services

While replacing Google's proprietary components would be relatively straightforward for ChromeCo, replacing Google's proprietary services would be more complex. In the case of some services, other independent browsers have already demonstrated approaches that ChromeCo could potentially adopt with relatively low effort. But for other services, ChromeCo would likely need greater engineering investment, assistance from Google (possibly by court order), or both to ensure NewChrome remains competitive.

As discussed in trial testimony, Chrome depends on a large number of Google-proprietary services made available via application programming interfaces ("APIs").[16] Many of these services only exist to support Chrome. So, it is unrealistic to expect Google to maintain them after the divestiture, and in many cases ChromeCo would need to find a way of fulfilling the same functions. At a high level, these services can be divided into three buckets:

- Core Services – required for any competitive browser
- Operational Services – necessary for operating a browser at Chrome's current scale
- Ancillary Services – Google-specific or optional services

The remainder of this section examines a number of high-profile services. This assessment does not include a discussion of every service in use today by Chrome. As noted in Section I, complete duplication of the existing services infrastructure should not be the bar to assess feasibility. Rather, it is important to understand these categories in order to identify the most critical services and to assess the technical challenges of replacing or transitioning those critical services. Each category is

[16] According to testimony from Plaintiffs' witness and Harvard Professor James Mickens, Google maintains many private APIs enabling Chrome to interoperate with Google's backend infrastructure:

> *"Q.…Google has back-end server infrastructure; correct? A. Yes. Q. And today, that back-end server infrastructure is used by Chrome? A. Yes. Q. And today, that back-end infrastructure is providing functionality for Chrome? A. Some of the functionality, yes…Q. And Google Chrome currently calls on APIs that are not publicly available; correct? A. Yes. Q. And when it comes to Google Chrome's private API calls, you have not individually studied those APIs; correct? A. I've looked at some of them. Q. But you mentioned that there were hundreds; correct? A. Yes."* Mickens and Maier, "Minute Entry for proceedings held before Judge Amit P. Mehta."

Mickens' testimony also referenced how Google also uses various public APIs to support the proprietary services accompanying Chrome:

> *"Q. For the purpose of your report, you relied on a public-facing Google document that identifies API keys; correct? A. Yes, I recall that document. Q. And that document identified 11 APIs. That was cloud search API, geolocation API, Google Drive API, safe browsing API, time zone API, admin SDK . . . cloud translation API, geocoding API, Google assistant API, Google calendar API, and nearby messages API. Does that sound familiar? A. Yes . . . . Q. You agree that those are the main public ones that you are aware of; correct? A. That's correct."* Mickens and Maier, "Minute Entry for proceedings held before Judge Amit P. Mehta."

discussed in turn below. A longer list of services that should be considered Ancillary Services is included in the Appendix.

Chrome services require functionality that is embedded in the browser (known as the "client") and the service provider's infrastructure (the "server"). The communication between the client-side code and the server-side code may use a communications protocol, which also may be proprietary to Google.

For Google's proprietary services discussed below, in general the client side of the service is included directly in the Chromium software. The server-side code is not public and can only be accessed via an API. Therefore, if ChromeCo wishes to retain the functionality provided by a Google service without support from Google, there are three primary options.

The most obvious option is for Google to transfer the existing service–including the code that powers it–to ChromeCo. During the trial, Google argued that this was impractical because the code was tightly integrated with the rest of Google's proprietary code and infrastructure and that it would be very difficult to decouple. The code for these services is not publicly available, so independent analysis of this question is not possible. However, Google's IT infrastructure is well-known in the industry for being tightly integrated, and so it seems likely that this type of transfer would be impractical, at least in the case of some services.

The second option is for ChromeCo to retain the existing Chromium client code and protocol, implement the same APIs that Google implemented but host them outside of Google's private infrastructure, and direct NewChrome to access those APIs at their new location. This option would be facilitated by the court requiring Google to share the code for its existing services with ChromeCo, allowing ChromeCo to reuse the generic, non-Google-specific pieces (for example, the API implementations themselves) while replacing the Google-specific pieces that depend on Google's infrastructure. In addition, Google could be required to share documentation and specifications for the APIs provided by these services.

Finally, ChromeCo could replace the client-side code in Chromium with its own code that speaks a different protocol from Google's protocol and connects to non-Google server infrastructure. This option could be facilitated by requiring Google to provide documentation about the existing implementation to ChromeCo.

Existing vendors of Chromium-based browsers have chosen from both of the latter two options depending on the situation.

a. Core Services

Core Services include services that users would expect in any competitive browser. As a general guideline, if a service is noticeable to the user and is currently present in some form in all four major browsers, it can be considered a Core Service.

This section analyzes the feasibility of ChromeCo offering five Core Services in detail: Safe Browsing, sync, translation, Chrome Enterprise, and the Chrome Web Store. The final section touches briefly on additional Core Services: captive portal, push notifications, geolocation, basic spell check, form fill, and password manager.

Safe Browsing

Safe Browsing is a reputation service in which Google compiles a list of websites, web pages, and downloadable files that it deems dangerous from a security or safety perspective.[17] Browsers can use the Safe Browsing service to determine whether the user is about to do something dangerous and display a warning and/or block the activity. The list of unsafe sites is updated by Google every 30 minutes.[18] Safe Browsing is used by Chrome as well as by Safari, Firefox, and Brave. Microsoft Edge uses a similar service called Microsoft Defender SmartScreen.[19] Google does not charge a fee for the use of Safe Browsing.

It is unclear to what extent Google would continue to support Safe Browsing if Google no longer owns Chrome. Chrome is not the only Google product that makes use of Safe Browsing: Google Search, Android, Gmail, and Google Ads also use Safe Browsing, so discontinuing Safe Browsing would be very disruptive to Google. However, Chrome likely represents a larger fraction of the marginal cost of offering Safe Browsing as compared to any other Google product or competing browser.

Safe Browsing incurs two main categories of costs:

1. The **fixed costs** of determining which sites, pages, and files are dangerous, and creating a database of those. These costs must be incurred to operate Safe Browsing, regardless of how many clients actually use it.
2. The **marginal costs** of providing Safe Browsing information to clients, which increase as the number of clients increases.

Considering the usage of Safe Browsing by browsers (Chrome and competitors), the marginal costs of operating Safe Browsing likely scale roughly with the size of the browser user base. Each browser needs to use Google's API to obtain the information needed to determine if the sites the user visits are safe. Because Chrome is the most widely used browser, it likely incurs a larger marginal cost than its competitors.

If Chrome is divested, there are four potential scenarios for how Google could handle Safe Browsing:

1. Google stops operating Safe Browsing entirely, thus avoiding all of the above costs, but also rendering its own non-Chrome products more vulnerable.
2. Google operates Safe Browsing only for Google products, thus avoiding the marginal costs of offering it to competing browsers.

[17] Google, "Making the world's information safely accessible."
[18] Google, "Enhanced Safe Browsing on Chrome."
[19] Microsoft, "Microsoft Defender SmartScreen."

3. Google operates Safe Browsing for Google's own products, and makes the database available for others to download, but does not operate a public Safe Browsing service.[20] This avoids the marginal costs of serving other browsers, including NewChrome.
4. Google continues to operate Safe Browsing as-is, potentially charging browser vendors for access, perhaps as the result of a requirement in the potential divestiture order.

The first two of these would require ChromeCo to identify or build a replacement for Safe Browsing or operate without it, while the last two present viable paths for ChromeCo to offer an equivalent service to what Google offers today, with some support from Google.

In the first and second scenarios, ChromeCo would need to develop its own mechanism for providing equivalent functionality. From a technical perspective, this would require a relatively small change to the client-side code in the browser, as the implementation of a Safe Browsing-type protocol is not very complicated. The challenging part of Safe Browsing is building and maintaining the reputation database of websites. This would require a substantial investment and different domain of expertise from browser software engineering. Major search providers are able to exploit their existing web crawl and index to build this reputation database, but this information will not necessarily be available to ChromeCo.[21] Potential Chrome buyers that are otherwise capable of running general web services at Google's scale may or may not have the capacity or interest in building such a service from scratch.

One possibility would be for ChromeCo to attempt to work with Microsoft SmartScreen, if Microsoft is willing to offer SmartScreen as a service for other web browsers. There are a number of other web-address-based reputation systems that exist today, but none of them are as comprehensive as Safe Browsing. If ChromeCo were to adopt one of those, it would reflect a significant security regression from what Safe Browsing offers today.

Because Safe Browsing provides protection for Google's web properties and for Android, it seems unlikely that Google would opt to stop operating Safe Browsing entirely, though it might decide to no longer make Safe Browsing publicly available, as in scenario 2. Given that Google debuted Safe Browsing 20 years ago (before launching Chrome), and that it has been offering it as a free service to competitors for much of that time, the court might find it reasonable to compel Google to continue offering the service publicly in some form to support a successful divestiture.

During the remedies trial, Google emphasized the role of an additional service, Enhanced Safe Browsing, that is only available in Chrome and not in other browsers.[22] The Enhanced Safe Browsing

[20] Ensuring that users receive timely updates is critical to Safe Browsing's effectiveness, so third-party vendors would need to have a mechanism to receive rapid updates.

[21] Microsoft, "Microsoft Defender SmartScreen"; Yandex, "Yandex Browser Protect."

[22] During her testimony, Defendant's witness and senior Google executive Parisa Tabriz reinforced the idea that Enhanced Safe Browsing is a key security feature exclusively available on Chrome:

> *"Q. . . . What is enhanced protection safe browsing? A. So enhanced protection is what we see as more advanced security protection. It uses AI and can help detect active attacks in real time. I have a security background, and attackers are constantly evolving how they do fishing and malware. And by a user opting in to give some additional telemetry and data from their Chrome client to safe browsing servers, we're able to detect more sophisticated attacks in real time. And it's something the user opts into. This is showing you in settings what happens when it's on and some things to keep in mind. Q. And is this an accurate description of how enhanced protection works for*

service conducts real-time analysis of the website URL and website content that the user visits, and shares the site visit information across Google's other products (such as Gmail and Google Docs). This allows users to be warned immediately about unsafe phishing or scam sites that may materialize during Safe Browsing's 30-minute update window. However, unlike regular Safe Browsing, Enhanced Safe Browsing requires the user to share website URLs and page content directly with Google, creating increased privacy risk. Enhanced Safe Browsing is off by default for consumer users of Chrome, although many of them have chosen to turn it on.[23]

Given that Enhanced Safe Browsing is off by default, is available only in Chrome, and entails significantly different privacy/security tradeoffs compared to Safe Browsing, Enhanced Safe Browsing should not be considered to be a Core Service from the perspective of the divestiture feasibility assessment. It is more properly understood as an Ancillary Service that could be removed from NewChrome and that ChromeCo may or may not wish to replicate.

Sync

All major browsers can synchronize ("sync") user data between browser instances, allowing users to more seamlessly use the same browser across multiple devices. Examples of user data include browsing history, bookmarks, and passwords.

Client-side sync code is built into Chromium, but actual syncing depends on services operated by Google, specifically its Gaia authentication system (tied into Google accounts) and the Chrome sync service.[24] It is possible that Google might continue to want to provide some kind of sync service in order to enhance the Android experience post-divestiture (for example, having passwords entered in NewChrome work in Android apps) or for data collection purposes.[25] However, Google might also decide not to operate a sync server, or ChromeCo might decide that it does not wish to use Google sync. Both Microsoft and Brave have built their own sync services independent of Google.[26]

Chrome users? A. Yes. Q. Is the enhanced safe browsing protection only available in Chrome? A. Yes." Tabriz and Maier, "Minute Entry for proceedings held before Judge Amit P. Mehta."

Defendant's witness and senior Google executive Heather Adkins echoed this point:

"Ms. Adkins, I want to ask a quick question or two about Google's safe browsing . . . does Chrome's enhanced safe browsing promote Google's ability to keep users secure? A. It does. It's actually one of the very first things that we built as a security kind of focused product…And is that enhanced version only available in Chrome? A. Yes." Adkins and Maier, "Minute Entry for proceedings held before Judge Amit P. Mehta."

Finally, during questioning of James Mickens, Google's counsel pressed Mickens on Google's support for the feature:

"Q. Is the enhanced safe browsing protection only available in Chrome? A. Yes. Q. How many Chrome users have opted into enhanced safe browsing? A. As of this year, over a billion users have opted into enhanced safe browsing. Q. Is enhanced safe browsing developed by the Chrome team? A. So the Chrome team works with the safe browsing team, which is a different team at Google.…Q. Does enhanced safe browsing rely on Google shared infrastructure? A. Yes." Mickens and Maier, "Minute Entry for proceedings held before Judge Amit P. Mehta."

[23] As part of her testimony, Google executive Parisa Tabriz disclosed that over one billion users have chosen to use Enhanced Safe Browsing. Tabriz and Maier, "Minute Entry for proceedings held before Judge Amit P. Mehta."

[24] Google, "Sign in and sync in Chrome."

[25] Google used data collected from Chrome sync to help develop its Privacy Sandbox feature, which is intended to provide quality advertisement targeting using less data collection. See Google, "Protecting privacy online."

[26] Brave, "How do I set up Sync?"; Microsoft, "Sign in to sync Microsoft Edge across devices."

The most straightforward approach would be to operate a new sync service that is compatible with the code in Chromium. Brave has already developed its own open source implementation of Chromium's sync protocol,[27] which would likely be a good starting point for ChromeCo. Brave's design does not depend on Gaia or any other account system, though ChromeCo might make a different design choice. Edge's sync system is tied to Microsoft's account system.[28]

While it would be easiest to get started by simply taking the Chromium sync protocol as-is, ChromeCo might also wish to evolve the protocol, which would be easier post-divestiture when ChromeCo is not burdened by Google's broader choices about Chromium's structure. For example, Mozilla's sync protocol is encrypted by default, whereas Chrome's design allows Google to read all the synced data by default.[29] According to Google's public disclosures, Google uses this data to personalize search and other Google services.[30] If ChromeCo does not have the incentive to use sync data for other purposes, ChromeCo might decide to make different choices in this area, or, as Brave has done, decide to decouple the sync service from user accounts.

Translation

Chrome has extensive translation features, including automatically translating web pages and translating captions from videos, podcasts, video calls, etc. These features operate by sending the content to be translated to Google, where it is processed, and the results returned. In general, any browser-based translation system for web pages has two parts:

1. Identifying the text to be translated and replacing it with the translated version inside the page.
2. The AI model which performs the text translation.

In the case of Chrome, Chromium internally detects when a page might need to be translated and if so downloads a translation script written in JavaScript from Google's servers. This script reads the page and then is responsible for sending the fragments of the page text to be translated to Google using a proprietary API. The API takes those fragments in the source language and returns the corresponding translations in the destination language. The downloaded script then reinserts the fragments into the page.

Both Brave and Edge offer their own translation systems. Brave downloads a script from Brave's servers rather than Google's servers. That script can then talk to Brave's translation service, which uses technology provided by Lingvanex.[31] A similar strategy should be usable with other commercially available translation APIs, such as that offered by Microsoft[32] or even the public Google translation API.[33]

[27] Brave, "Brave Sync Server v2."
[28] Microsoft, "Sign in to sync Microsoft Edge across devices."
[29] Mozilla, "Sync Firefox data."
[30] Google, "Find & control your Web & App Activity."
[31] Lingvanex, "Advanced Learning Technologies."
[32] Microsoft, "Pricing."
[33] Google, "Cloud Translation API."

Firefox uses a different strategy which performs all translation locally on the user's device. The overall strategy is similar to that described above except that instead of invoking an external API the translation system queries a local AI model. This approach has a different set of tradeoffs between privacy and accuracy because the content never leaves the user's device, but the AI models used are limited to those that can be run locally.

Enterprise Management

Enterprise administrators often want to monitor and control the deployment and behavior of software installed on employees' devices. Google provides a version of Chrome ("Chrome Enterprise")[34] intended for this setting.[35] Chrome Enterprise deployments are enrolled in a management system which allows system administrators to remotely configure Chrome (e.g., to restrict the set of sites an employee can visit or which extensions they can install) and to monitor system and user behavior (e.g., for data loss prevention). This administration is performed via a web console tied into the enterprise's Google account. Google offers both a basic free ("Core") version of Chrome Enterprise and a paid ("Premium") version.

Enterprise management is a common browser feature and is provided in some form by Edge,[36] Safari,[37] and Firefox.[38] Unlike Google, Microsoft and Apple have integrated management functionality with the remote management capabilities they offer to enterprises that use their operating systems, which support both browser and non-browser applications. Mozilla also uses those operating system mechanisms to enable enterprise remote management of Firefox.

ChromeCo could opt to provide a full-featured remote management system as Chrome does, or a more stripped down system that depends on existing operating system mechanisms as Firefox does. Which of these options is most attractive would depend on ChromeCo's business objectives and in particular whether they wish to emphasize consumer or enterprise users.

If ChromeCo chose to provide a full-featured remote management system, it could opt to write the code for this itself (as Microsoft effectively did) or to reuse Google's existing console. This is a standard "build versus buy" analysis and would depend on the degree to which Google's management console relies on Google's proprietary infrastructure.[39]

[34] Google, "The future of endpoint security."
[35] This report does not analyze how much additional code this version of Chrome has. In any case, Google could be required to provide any proprietary changes to ChromeCo as part of a potential divestiture order.
[36] Microsoft, "Microsoft Edge for Business."
[37] Apple, "Work."
[38] Mozilla, "Firefox for Enterprise Support."
[39] It may also be the case that the management console depends on the Google Cloud Platform. This would not preclude ChromeCo from taking over this functionality, though ChromeCo might eventually wish to transition to a different cloud provider.

Chrome Web Store

All major desktop browsers (and some mobile browsers) support an “extensions” mechanism that allows for third parties to write components that run inside the browser.[40] Examples of these extensions include ad blockers, password managers, shopping assistants, and grammar checkers. The Chrome extension mechanism is built into Chromium and available to other Chromium-based browsers, but Google, like other browser vendors, operates a web store (the “Chrome Web Store”) that allows users to shop for and download extensions.[41] As part of operating the Chrome Web Store, Google reviews extensions for compliance with Google’s policies and automatically updates installed extensions.[42]

ChromeCo would most likely need to provide its own web store. From a technical perspective, the web store site is relatively straightforward to develop. As with Enterprise Management, ChromeCo would face a build-versus-buy decision about whether to write a new site itself (as Microsoft did) or try to adopt Google’s existing site. Given the comparative simplicity of the web store, building its own store is more likely to be attractive.

In addition to the site itself, ChromeCo would need to operate a review program for submitted extensions. Because extensions can be powerful tools with the ability to access or modify sensitive user data, such a review program is an essential part of safeguarding the security of the extension ecosystem and ensuring that users can safely download and install extensions. A web store program is a mix of policies governing extensions, practices for reviewing extensions for compliance with those policies, and software tools to assist in review of extensions. As with the web store site, ChromeCo could develop these independently, but would benefit from being able to start fromGoogle’s existing review processes even if it were unable to directly adopt the tooling. A potential divestiture order could require Google to share information and code with ChromeCo in order to accelerate this transition, including Google’s existing records about which extensions and extension vendors have been approved, rejected, and/or blocked.

Other Core Services

Chrome also contains a number of other core services that ChromeCo would most likely wish to provide. These are by and large straightforward to implement:

- The *captive portal service* assists in determining whether the user is behind a "captive portal" that requires the user to log in before providing network access, as is common in hotels, coffee shops, and other locations. This feature depends on a server operated by Google. Other browsers have similar features and so it would not be difficult for ChromeCo to implement this using a new server.

[40] Brave, “Browser Extensions”; Microsoft, “Edge Add-ons”; Mozilla, “Firefox Browser Add-ons.”
[41] Google, “Welcome to the Chrome Web Store.”
[42] While many of these policies are non-controversial, Google has also used its control of Chrome to restrict the use of APIs which are useful for advanced ad and content blocking extensions. For example, uBlock Origin is no longer available on Chrome because Google has forbidden the use of one of the APIs on which it depends. See uBlock Origin, “uBlock Origin.”

- The *push notification service* allows websites to send notifications to users even when the user is away from the site. This is implemented in accordance with public WebPush specifications which require the browser vendor to provide a server that forwards messages from sites to users.[43] Both iOS and Android provide push notification services that can process notifications regardless of which browser is used, but ChromeCo would need to operate its own push notifications server for desktop users.

- The *geolocation service* allows websites to ask the browser for the location of the user's device via the Web Geolocation API.[44] This is useful for location-dependent services like travel or weather sites. On mobile devices, the browser can query the operating system location system, but on desktop devices the browser needs to use a geolocation service based on the user's internet protocol (IP) address. Google uses its own geolocation service for this purpose and also offers it as a commercial service.[45] ChromeCo might choose to use Google's service or work with another geolocation provider.

In each of these cases, ChromeCo could potentially use Google's existing services while it transitioned to a new service. Other core services noted in trial testimony include basic spellcheck, form fill, and password manager. All of these are included in Chromium and do not require server support from Google.

b. Operational Services

Modern browsers depend on a number of Operational Services to keep them functional and up-to-date. These services are common to any large piece of end-user software (e.g., a spreadsheet or a messaging app). They are especially critical for web browsers because browsers face an ever-evolving array of security threats, attacks, and usage patterns originating from the open web. The three services discussed below are automatic updates, telemetry, and crash reporting.

Automatic Update Service

Automatic software updates are an essential feature for modern browsers. On mobile operating systems, software updates are provided by the vendor via an app store, but on desktop the browser vendor needs to provide its own update mechanism. Chrome's automatic update service is built on a system called "Omaha," which has both a server component (to serve the updated versions) and a client component (to download and install the updates).[46] Omaha is also used to provide updated configuration information to the browser, which can enable new features or inform the browser about compromised security certificates. Omaha also provides a reliable source of time even if the computer's clock is wrong.[47] The current Omaha client is open source.[48] Google would be unlikely to

[43] Beverloo et al., "Push API"; Thomson and Damaggio, "Generic Event Delivery Using HTTP Push."
[44] W3C, "Geolocation."
[45] Google, "Geolocation request and response."
[46] Omaha Consulting, "Google Omaha Tutorial."
[47] Google, "Manage your location settings in Chrome."
[48] Omaha Consulting, "Google Update for Windows."

continue to run the Omaha server for ChromeCo's releases post-divestiture, so ChromeCo would need to make other arrangements.

Fortunately, both Edge and Brave already update via Omaha, so there is existing precedent in the industry for ChromeCo to follow. Brave uses a server licensed by Omaha Consulting which offers Omaha integration for Chromium-based clients.[49] Omaha Consulting has its own server code which it will run for customers or will license to customers to run in their own infrastructure. The list price for a self-hosted installation is 13,000 Euros with 7,000 Euros/year in maintenance.[50] Brave currently operates such a self-hosted instance.[51]

Alternatively, ChromeCo could independently develop and operate a server which speaks the Omaha protocol, potentially based on the existing open source server.[52] The protocol is fairly straightforward, so this should not be prohibitively expensive.[53] Although rapidly updating the user base of a product the size of Chrome involves considerable resources in terms of computing power and network capacity, this is the type of task for which modern hyperscaling platforms such as content distribution networks (CDNs) are optimized, and support for such updates is available on a commodity basis.

Telemetry Service

Chrome, like most other browsers, includes a telemetry service that Chrome calls User Metrics Analysis (UMA).[54] UMA allows the browser to collect a large variety of metrics about user behavior (for example, page views, loading speeds, which features are used frequently, etc.) and report them to Google for analysis. The client-side code for this feature is already in Chromium, but the server-side is a Google proprietary service. The API used by Chromium to report this information is reasonably straightforward and can be determined by reading the Chromium source code that sends the reports.

Assuming that ChromeCo wishes to continue to collect this kind of data, it would have two main options:

1. Implement a new service that accepts Google's protocol and send the data to that new service.
2. Replace the current metrics uploader in Chrome with client-side code that implements the protocol for another analytics platform (as Brave does).

Both of these approaches would allow ChromeCo to retain the code in Chromium that collects each metric at the appropriate time, which represents much of the client-side engineering effort.

---

[49] Omaha Consulting, "We solve your automatic update needs."
[50] Omaha Consulting, "Pricing."
[51] Herrmann, "Brave omaha."
[52] The existing open source server implements an older version of the Omaha protocol and would need to be updated to support the newest protocol. This should be technically straightforward. It would also be possible to design or adopt a different update system, such as the one in Firefox, but that approach is likely less convenient. See Omaha Consulting," Google Omaha ("Chromium Update") Server."
[53] Chromium, "Omaha Protocol 3.1"; Chromium, "Omaha Protocol 4.0."
[54] Google, "User Metrics processes data to improve Chrome services."

As Google's server-side UMA code is not public, ChromeCo would need to provide its own server or use an existing third-party metrics platform. The server-side engineering effort largely consists of receiving and parsing the uploaded metrics, detecting and rejecting erroneous data, loading the metrics into a database, and computing bulk metrics. Maintaining this kind of metrics pipeline is well within the capabilities of a sophisticated engineering organization. For instance, Mozilla operates such a pipeline for Firefox. ChromeCo could also potentially outsource the analytics system to one of many existing analytics vendors, though most of these vendors' systems do not currently operate at the scale that Chrome does.

Analysis of telemetry data often depends on comparing newly received data to historical data. It may be beneficial for the court to require that Google share historical telemetry data with ChromeCo.

Crash Reporting

Despite the best efforts of software vendors, users frequently experience crashes. Chrome, like many other browsers, can be configured to report details of crashes so that Google can try to diagnose and fix them. Fortunately, the Crashpad library that Chrome uses is widely used for open source crash reporting, and implements a straightforward protocol.[55]

ChromeCo would have several options for using Crashpad-based crash reporting. There are a number of open source servers which will accept CrashPad reports, so ChromeCo could simply run one of these on its own infrastructure.[56] Alternatively, ChromeCo could use an existing commercially hosted service such as Backtrace.[57]

c. Ancillary Services

In addition to the above services, Chrome offers a variety of services that make use of Google-specific APIs and are not typically included in other major browsers.

For example, Price Tracking was cited at trial by Google's expert witness as a service that depends on Google's infrastructure and that would no longer work if NewChrome's ability to communicate with Google servers were blocked.[58] This service is off by default in Chrome today. Similar to Enhanced Safe Browsing, users can opt into this service in exchange for disclosing their browsing activity directly to Google, thereby accepting additional privacy risks, and making that data available to improve Google search.[59] Price Tracking is similar to other coupon-clipping web services that provide consumers with price alerts or discounts in exchange for tracking their browsing activity.[60]

ChromeCo could find alternative price tracking services if it wanted to keep this functionality. However, as it is off by default in Chrome and may be providing benefits to Google (in the form of user data) in

[55] Chromium, "Crashpad Overview Design," Upload to collection server.
[56] Electron, "Minimum breakpad crash reports collecting server"; Mozilla, "Socorro."
[57] Backtrace, "Error and Crash Reporting for Games & Mobile Apps."
[58] Nieh, "Minute Entry for proceedings held before Judge Amit P. Mehta."
[59] Google, "Shopping insights & price tracking in Chrome."
[60] Honey, "We search for coupons at 30,000+ sites to help you save money." Amazon Assistant, now defunct, offered a similar service. See Mellon, "Amazon is killing one of its most underrated shopping features."

which ChromeCo has no interest, ChromeCo may choose not to replicate this service. Disabling it would require relatively minor changes in the Chromium client code.

A number of other Google-specific Chrome services fall into this Ancillary Services category, for example:

- Enhanced spell check – uses Google's servers for spell checking[61]
- Gemini in Chrome – uses Google's Gemini AI models to assist with web browsing[62]
- Google Lens – image search built into Chrome and based on Google Search[63]

A future ChromeCo might wish to retain some version of these features. Edge and Firefox have their own AI integration features, for comparison.[64] Or ChromeCo might wish to offer a different feature set entirely, as newer browsers such as Arc, Dia, and Fellou have done.[65] In any case, there is wide variation in the availability of these Ancillary Services across the major browsers. As a result, none of them are specifically necessary in order to have a competitive browser.

The Appendix lists other Google-specific services present in Chrome that were mentioned during the trial and should be considered Ancillary Services.

B. Feasibility of Chromium-Based Browsers without Google

As explained in Section II, there are currently three companies that build and maintain browser engines and use those engines as the basis of their browser offerings: Google, Apple, and Mozilla. The fact that Apple and Mozilla both offer competing engines and browsers, with vastly different business models, demonstrates that non-Google entities can indeed maintain competitive browser engines.

Given its market position, Google has been able to invest the most in its browser and engine, and WebKit and Gecko have benefited by incorporating technologies originally developed for Blink, including graphics libraries and rendering engines,[66] real-time audio and video code,[67] and more. But Chrome has also benefited from the other companies' innovations, including the WebAssembly format used for high-performance applications[68] and the Rust programming language,[69] both of which were developed at Mozilla. Perhaps more importantly, the three companies have prioritized different aspects of browser technology in developing their engines, reflecting their own business priorities and enabled by the fact that each company has its own engine. Google has prioritized new capabilities designed to

[61] Google, "Turn Chrome spell check on and off."
[62] Google, "Meet Gemini in Chrome."
[63] Google, "Search with Google Lens in Chrome."
[64] Brave, "Leo, Brave's in-browser AI assistant, now incorporates real-time Brave Search results for even better answers"; Mozilla, "Access AI Chatbots in Firefox."
[65] Arc, "Ready to let go of the old internet?"; Dia, "Dia"; Fellou, "Fellou."
[66] Skia, "Welcome to Skia."
[67] WebRTC, "Real-time communication for the web."
[68] Eng et al., "WebAssembly and WebGPT enhances for faster Web AI, part 1."
[69] Rust, "Rust"; Tung, "Programming Languages."

bring the browser environment into parity with mobile development,[70] which has not been a priority for Apple or Mozilla.[71] Both Apple and Mozilla have prioritized privacy features that Blink lacks.[72]

Plainly, then, it is possible for non-Google entities to develop and maintain a modern, competitive browser engine. The question then becomes whether there is something unique about Chromium that would make it difficult for an entity other than Google to maintain it. The primary issues that Google raised on this topic during the trial were:

- **Personnel.** Google has contributed most of the personnel to Chromium development.[73] ChromeCo would need the requisite personnel to do the work going forward.

- **Collaboration.** The Chrome team collaborates extensively with other parts of Google. Google points to advantages this creates that would be lost if Chrome and Google were to be separate companies.[74] ChromeCo's engineers would operate without internal collaboration with other parts of Google.

- **Dependence on Google-internal tools.** Google invests heavily in software engineering tooling, which is used by the Chrome team to develop Chrome.[75] ChromeCo's engineers would not have access to some of this tooling.

Each of these concerns are discussed in detail below.

---

[70] Google, "New capabilities status."

[71] Mozilla, "Mozilla's vision for the evolution of the Web."

[72] For example, both Firefox and Safari have extensive anti-tracking features that Chrome lacks. Firefox also has a feature called "multi-account containers" that allows the user to separate out their activity on different websites for increased privacy. These features all depend on deep integration with the browser engine.

[73] Defendant's witness and Columbia University Professor Jason Nieh testified during the remedies trial that Google has written the vast majority of Chromium's code:

> *"Google contributes over 90 percent of the code commits to Chromium. In fact, I think last year, it was 94 percent of the commits. So it's doing the vast majority of work to make Chromium possible. And if then Chrome is divested and say you're divesting the Chrome team too, then I don't know who's left to make all these contributions to Chromium at Google."* Nieh, "Minute Entry for proceedings held before Judge Amit P. Mehta."

[74] In her testimony, Google executive Parisa Tabriz emphasized the benefits of collaboration over development of Chromium happening within a single company:

> *"Chromium, people can build browsers on Chromium, but we have had 17 years of innovation and building interdependencies because we work as one company. We can all get in a room, we can work from shared co[de]bases and continue to deliver features and innovations in close collaboration. . . . I think people can build Chromium-based browsers. When it comes to Google Chrome, it, again, is 17 years of working closely with Google. And we benefit from efficiencies from innovation, from security and infrastructure that comes with Google . . . . And I think Chrome is the best browser in the world because we get to work at, you know, one, if not the most innovat[ive] companies in the world."* Tabriz and Maier, "Minute Entry for proceedings held before Judge Amit P. Mehta."

[75] Google's counsel questioned Plaintiffs' expert witness James Mickens about the Chrome team's use of Google's proprietary developer tools:

> *"Q. And today, the Chrome team uses Google's shared developer tools to build the code of the Chrome browser; correct? A. Those are some of the tools they use, yes."* Mickens et al., "Minute Entry for proceedings held before Judge Amit P. Mehta."

1. Personnel

Labor concerns are an important factor in any divestiture or acquisition. In the case of the potential Chrome divestiture, there are a number of reasons why ChromeCo would likely prioritize having as much of the existing Chrome team as possible join ChromeCo. Staffing up a large software organization is challenging under any circumstances, so retaining a significant portion of the Chrome team would allow ChromeCo to transition the product more quickly. The talent pool of software engineers with deep browser expertise is also relatively small compared to the overall software industry.[76] Many software engineers have moved from working on one browser engine to another successfully–indicating that a talent pool certainly exists outside of Google, and that specific knowledge of the Chromium code base matters less than knowledge of the general engineering principles underlying browser technology. But Google presently has the largest swath of that talent pool, making Chrome engineers an important resource for ChromeCo.

Retaining key members of the existing Chrome team is especially important early in the transition process because those staff members represent a valuable source of institutional knowledge. Although in principle a strong web engineer will be able to understand the Chrome code base on their own, access to existing staff can significantly accelerate the process of coming up to speed. Google should be expected to provide relevant organizational charts, product plans, engineering and architectural documents, and any other artifacts that ChromeCo would need for new engineers to develop expertise with Chromium.

As a practical matter, this means ChromeCo would need to make attractive offers, both in terms of compensation and by offering a compelling sense of mission for those interested in contributing to an endeavor broader than a single software product: the web's continued development. The right organization might even find it easier to hire than Google does. Google's hiring process is notoriously challenging for experienced staff,[77] and ChromeCo may be able to articulate a vision for the web more compelling than Google's. Mission alignment is one reason why Mozilla has been able to maintain a competitive browser and contribute substantial innovation to the web ecosystem while paying its engineers significantly less than Google.[78]

---

[76] Defendant's witness Parisa Tabriz testified that around 1,000 Google engineers contribute to Chromium:

> *"Q. How many members of the Chrome team work on Chromium? So most people who are working directly on Chromium are engineers, and I would say, roughly, my team is a thousand engineers and the rest are designers, product managers, and so at least a thousand people are contributing to Chromium from within my team."* Tabriz and Maier, "Minute Entry for proceedings held before Judge Amit P. Mehta."

By comparison, Mozilla is estimated to employ roughly 500 staff (including non-engineers) to maintain Firefox. See Cooper, "The True Cost of Browser Innovation." There are over 1.5 million software developers employed across the US, according to Department of Labor statistics. See United States Bureau of Labor Statistics, "Occupational Wages and Employment, May 2022."

[77] See, e.g., Lakshmi, "These are the hardest companies to interview for, according to Glassdoor."

[78] For comparative salaries, see Cooper, "The True Cost of Browser Innovation."

2. Collaboration

Collaboration with a larger parent organization is not necessary to produce web innovations. While such collaboration within Google has been beneficial, it has also been harmful for competition in some instances.

Chrome and Chromium development have benefitted from close collaboration with engineers from other Google teams including Android, security, cloud, and video. Examples of this collaboration over the past 15 years include: Safe Browsing (discussed above); WebRTC, which is the basis for real-time audio and video calling on the web; and the QUIC networking protocol, which is the basis for the most recently developed mechanism for transporting data on the web. While these are real contributions to the technological base of the web, innovation on the web platform is by no means unique to Google. During the same time period, Mozilla developed WebAssembly,[79] the Rust programming language (which is now used inside Chromium as well as across the web), and Let's Encrypt, a certificate authority which is now responsible for issuing more than half of the certificates used by servers to secure their connections on the web.[80]

This is not to compare the amount of technological development between Mozilla and Google—indeed, it would be shocking if Google did not develop more web technology given its vastly greater resources—but merely to illustrate that collaboration with a larger parent organization is not necessary to continue to innovate on the web. These examples show that there is precedent for ChromeCo to be able to drive web innovations whether or not it or its parent company also offers products in adjacent markets.

Furthermore, many key web innovations have resulted from cross-industry collaboration, not merely collaboration within Google. While each of the examples cited above derived to some extent from Google internal work, the resulting technology development ultimately became a collaboration with other industry participants, with both WebRTC and QUIC becoming industry standards, and improving dramatically in the process.[81] This dynamic continues to be on display with the development of the so-called "agentic web," with large and small AI, cloud, and browser companies collaborating to develop and deploy technologies that facilitate AI agents' use of the web.[82] ChromeCo would benefit from the web's culture of cross-industry collaboration, and hopefully contribute to it.

Google has also leveraged internal collaboration to exclude competitors. Strong internal collaboration allows Google to design technologies that meet its own product needs and disadvantage rivals. To pick one well-known example, the early versions of WebRTC deployed by Google included a specific mechanism for handling multiple media streams (e.g., for a video conference with more than two participants), that was different from what was eventually standardized.[83] This produced an incompatibility with Firefox—which followed the standard—and was in part responsible for a

[79] In fact, WebAssembly replaced a Google proprietary system known as "native client."
[80] Web3Techs, "Usage statistics and market share of Let's Encrypt as SSL certificate authority on websites."
[81] Iyengar and Thomson, "RFC 9000"; Jennings et al., "WebRTC."
[82] See, e.g., contributions to the Agent2Agent (A2A) Protocol and Model Context Protocol (MCP) specifications. Google, "A2A"; Model Context Protocol, "modelcontextprotocol"; Surapaneni et al., "Google Cloud donates A2A to Linux Foundation."
[83] Boström, "Intent to Deprecate."

years-long period where Firefox did not work with either Google Meet or Microsoft Teams. This situation was directly facilitated by Google controlling both the biggest browser and its own conferencing site (Google Meet), and being able to collaborate between those teams.

3. Google-Internal Tools

During the trial, questions were raised about whether a Chrome buyer could maintain the browser without Google's internal software development tools. There are two aspects of this concern: access to Google's internal code base ("Google 3"), and access to Google's proprietary internal development tools.

ChromeCo's lack of access to Google 3 does not present any real concern. It is true that Google has an extensive common shared code base that is used for internal Google tools, but all of the Chromium code base is open source.

Google has invested very heavily in its own internal development toolchain, which is in many respects superior to what is publicly available. Google is not the only organization which has done so. For instance, Meta has also invested heavily in this area. However, smaller organizations are unlikely to have comparable tooling.

While ChromeCo would not be able to take advantage of Google's specific tools, there are now a large number of advanced commercial and open source development tools available—in many cases modeled on Google's contributions in this area—and ChromeCo should be expected to take advantage of this kind of commodity tooling.[84] Moreover, while convenient, it is clear that Google's specific tools are not necessary in order to develop a competitive browser because Apple and Mozilla already do so. Mozilla only makes use of publicly available tools (including some that Mozilla developed and open sourced).

C. Chromium Governance

The Plaintiffs' proposed remedies include both Chrome and Chromium in the proposed divestiture.[85] While Google is currently the main contributor to and leader of the Chromium project, the company is likely to significantly reduce its contributions to Chromium once it no longer controls Chrome. There are a number of potential future governance structures for Chromium, and the potential divestiture order should take care to allow ChromeCo and the web community the flexibility needed to identify the best one. The possibilities include:

- ChromeCo takes over Chromium and sets the direction similar to how Google does now.

[84] For instance, Cursor.ai and GitHub CoPilot for AI-assistance, VSCode for cloud development, and Mozilla's taskcluster for continuous integration. See Cursor, "The AI Code Editor"; GitHub, "GitHub Copilot"; Mozilla, "Taskcluster"; Visual Studio Code, "Your code editor."
[85] United States et al., "Plaintiffs' Revised Proposed Final Judgment," 12-13.

- Chromium transitions to a governance model that seeks to balance the needs of the various stakeholders, similar to how the popular Linux operating system[86] or the Python programming language[87] operate today.[88]
- The Chromium project splits ("forks") into multiple independent projects operated by different entities, with each borrowing from one another.

Under any of these scenarios, it would be critical for the court to include public interest conditions in its order to ensure that Chromium remains open source,[89] that Chromium's governance structure can be redesigned in light of the ownership transfer, and that Google is prevented from having sole decisional authority over Chromium updates.[90] Without these conditions, ChromeCo could choose not to release any of its improvements to Chromium or release them under more restrictive conditions while simultaneously discouraging contributions from others. This could cause inadvertent harm to the browser industry resulting from the search remedy. Ideally the court would condition the sale of Chrome on the buyer's commitment to maintain Chromium as open source with the full project, including the buyer's own future contributions, offered under a permissive license for the duration of the remedy term.

As noted in Section II, the V8 JavaScript VM is technically separate from Chromium/Blink. V8 is used in other products that need a JavaScript implementation, such as the cross-platform Node.js JavaScript development environment[91] and the Electron application development framework.[92] Given its broader use, V8 need not have exactly the same governance model as Chromium, though Chromium would likely remain the most important user of V8. As with Chromium, the court would need to ensure that V8 remains open source and that Google is prevented from having sole decisional authority over V8 updates. Providing flexibility for ChromeCo to work with other interested stakeholders to determine the best governance structure for V8 will be key.

D. ChromeOS and WebView

Chromium governance changes could potentially impact all developers who are downstream of Chromium (i.e., all developers whose products rely on Chromium code). In addition to Chrome, Google maintains two other such products: ChromeOS and WebView. The potential divestiture order needs to make clear whether these products are included in the divestiture. The implications for each product are discussed below.

---

[86] Linux Kernel Organization, "A guide to the Kernel Development Process."
[87] Python Software Foundation, "PEP 13 - Python Language Governance."
[88] The Linux Foundation has also recently formed a "Supporters of Chromium-Based Browsers" initiative. See Linux Foundation, "Supporters of Chromium-Based Browsers."
[89] The current Chrome code is already open source licensed and publicly available with copies in many locations, so there is nothing ChromeCo could do to reverse that. However, the current Chromium software license does not require the owner of Chromium to make its own future contributions open source, nor does it prevent the owner from releasing its own contributions under a more restrictive license.
[90] For more on the public interest conditions, see Cooper, "The True Cost of Browser Innovation."
[91] node, "Run JavaScript Everywhere."
[92] Electron, "Build cross-platform desktop apps with JavaScript, HTML, and CSS."

1. ChromeOS

ChromeOS is an end-user operating system which powers Google's Chromebook devices. Like Android, ChromeOS sits on top of a stripped down version of the Linux operating system (the "kernel") but the user interface is currently provided via Chrome. Effectively, ChromeOS apps are enhanced web apps. The non-Chrome pieces of ChromeOS are found in the ChromiumOS project, operated by Google.[93]

A potential divestiture order may or may not include ChromeOS and ChromiumOS. If they are included, the feasibility assessment is straightforward and aligns with the considerations discussed throughout this report. ChromeCo would own and maintain the code bases for ChromeOS and Chromium OS, allowing for ease of management. Chromebook vendors would license ChromeOS from ChromeCo rather than from Google. Developers of ChromiumOS derivative products (of which there are few) would experience a new governance structure for ChromeOS as chosen by ChromeCo.[94]

The scenario where ChromeOS and ChromiumOS are not included in the divestiture is more complicated. As a developer of a product downstream of Chromium, Google would have to contend with changes made to Chromium under Chromium's new governance structure. The difficulty of the managerial task for Google would turn on the extent to which ChromeOS is dependent on internal implementation choices in Chromium. Similar to how Microsoft, Brave, and other developers of Chromium-based browsers must contend with regular updates to Chromium, so too would Google for ChromeOS.

Having strict court-ordered line-of-business restrictions and vigilant monitoring of those restrictions would be critical under this scenario, because Google would be allowed to maintain what is effectively an OS built out of a browser. The potential divestiture order would need to prevent Google from offering or evolving ChromeOS as a stand-alone browser, so as to keep Google out of the browser market for the duration of the remedy period. Google would also likely want to arrange to have a third-party browser, whether NewChrome or another browser, preinstalled with ChromeOS by default. The remedy order's contractual prohibitions on Google would need to allow for this, as those proposed by the Plaintiffs do.[95] In this scenario, Google would have an incentive to continue to contribute to Chromium, especially if it included NewChrome as its default browser.

2. WebView

Developers of both desktop and mobile apps now commonly embed a browser engine inside their applications. This engine is used to display web pages and to implement the application user interface, allowing the developer to write the interface once and have it work on multiple platforms or as a web-only version of the application. On desktop systems, the most common practice is to use a full application development framework that bundles in a web engine, such as Electron, which bundles

[93] Google, "chromium/chromiumos."
[94] See, e.g., FydeOS, "FydeOS"; openFyde, "openFyde."
[95] United States et al., "Plaintiffs' Revised Proposed Final Judgment," 7-12.

in Chromium.[96] Electron is widely used to develop cross-platform desktop apps, including Slack, Discord, Figma, and many more.

On mobile operating systems, the standard practice is to use a WebView, which is effectively an encapsulated web engine provided by the operating system. On Android, Google offers a WebView based on Chromium/Blink.[97] Developers have the option of providing their own web engine, but the more common practice is to use the Android-provided WebView. This spares the developer from having to keep the web engine up-to-date and it reduces the download size of the developer's app.

As with ChromeOS, the court's order may include WebView in the divestiture. Unlike ChromeOS, which is largely a discrete product offering, WebView is an essential Android-based tool that helps Android developers build full-featured apps. If Android were no longer bundled with a built-in WebView, the result would be extremely disruptive to developers who depend on WebView. There are two main scenarios in the event that WebView would be included in the Chrome divestiture:

1. The court would allow Google to continue including its own WebView component in Android.
2. The court would forbid Google from developing its own WebView component but permit Google to bundle one or more third-party WebViews with Android.

In the first scenario, Google would again be a downstream developer of Chromium code maintained by a third party, as in the ChromeOS scenario above (and similar to the case where WebView is not divested). Experience with Electron is instructive here. Electron is independent of Google, having been created by GitHub and later moved to be governed by the OpenJS Foundation.[98] Electron's broad popularity suggests that it is practical for an entity to maintain downstream software that is tightly integrated with Chromium.

The second scenario depends on a third party continuing to maintain a WebView component. Presumably in this scenario, Google would contract with ChromeCo or another third party to provide the WebView.

IV. Transition Planning

Given the scale of Chrome's current deployment and the complexity of browser software, any Chrome divestiture will require a significant transition period where Google and ChromeCo cooperate to deliver Chrome. This section discusses specific considerations for the transition.

The first two sections below discuss aspects that are critical to the use of the browser and need to be transitioned rapidly to ChromeCo: (1) software distribution and update and (2) user data migration. The

[96] Electron, "Build cross-platform desktop apps with JavaScript, HTML, and CSS."
[97] Google, "Android System WebView."
[98] OpenJS Foundation, "Electron joins the OpenJS Foundation."

final section examines the remainder of Google-specific services utilized by Chrome. If ChromeOS were to be divested together with Chrome, these same issues would need to be addressed for ChromeOS and Chromebook users.

A. Software Distribution and Update

The first essential part of the transition is to migrate users from Google's version of Chrome to the one maintained by ChromeCo. This needs to happen even if those versions are otherwise nearly identical. There are two issues which must be addressed:

1. New users who are trying to download Chrome must be redirected to a page where they can get NewChrome.
2. Existing users must be updated to NewChrome.

New users are comparatively easy to handle: Google can configure their existing Chrome download page to automatically redirect the user to ChromeCo's page for NewChrome.

Existing users are more complicated because they automatically update via Omaha. There are multiple possible approaches for handling existing users. It may be possible to redirect the update from inside Omaha or Google may need to serve some code provided to it by ChromeCo, which then effectuates the update. Once the user has updated once, future updates can come from ChromeCo's servers. Google will also need to cooperate with ChromeCo to ensure that the proper signing keys are used to digitally sign ChromeCo's software releases so that they can be seamlessly installed on existing machines.[99]

Both of these transition measures will need to be in place for an extended period of time, likely years. Automatic updates must continue until the vast majority of the Chrome user base has transitioned.[100] This is not merely an issue of complete transition but also user security, as otherwise some users will have out-of-date versions which have disclosed vulnerabilities. The redirect for new users needs to remain in place for as long as the court can reasonably order it, in order to handle any links on the internet to the old Chrome download page (which could continue to exist for decades).

B. User Data

User data to be migrated from Google to ChromeCo falls into two categories: data stored locally on the user's device, and sync data, some of which may be stored on Google's servers as well as on the user's device.

[99] This applies to desktop updates. As noted above, mobile application updates are handled by the app store. For iOS, see Apple, "Overview of app transfer." For Android, see Google, "Transfer apps to a different developer account."
[100] Typically a large portion of a browser user base can be updated in a matter of days. A smaller portion will take slightly longer (weeks) for a variety of reasons, such as internet connectivity issues that limit the user's ability to receive updates. And a longer tail of users can take months or years. Even for this long tail, software makers want to maintain the ability to update these users so as to minimize the number of vulnerable browsers in use and limit browser-based security risks.

1. Local Data

As a technical matter, on most platforms, any browser can read any other browser's data stored locally on the user's device. Conventionally, when browsers are first installed they may ask to import data from other browsers, but they will not do so automatically.[101] When browsers are updated, they simply read the data associated with their previous version. Depending on whether NewChrome is viewed as a new product or just the continuation of Chrome, either option may be defensible. It seems likely that NewChrome would opt for automatic import unless the terms of the potential divestiture order require ChromeCo to obtain user consent.

2. Synced Data

The situation with synced data is more complicated. As described above, Chrome's sync infrastructure is tied to Google accounts (the Gaia system). It seems likely that ChromeCo will wish to migrate users into their own infrastructure. This may require the users to make an account on ChromeCo's account system, as most browsers tie sync data to an account, as with Firefox and Edge. Alternatively, ChromeCo could use Google accounts to authenticate to its own infrastructure, as many websites do.

There are two possible ways to effectuate the transfer of synced data:

- The client syncs and then updates its data to the new server.
- Google's servers send the data to ChromeCo's servers directly.

ChromeCo can effectuate the first option without any assistance from Google. Note that because the same credentials are used for Chrome sync and the user's Google account, NewChrome could in principle retrieve all of the user's data, not just the Chrome sync data. This likely would not match user expectations, and as such it may be necessary to have some technical or administrative controls to prevent this.

In the second option, Google would need to provide ChromeCo with an API or protocol that ChromeCo can use to transfer the data from Google to ChromeCo. Given the ubiquity of server-to-server data transfers on the internet, setting this up would likely be reasonably straightforward. The court might also consider requiring Google to operate the sync services for an interim period while ChromeCo arranges to transition sync to its own service.

3. Operational Data

To understand its user base, Google likely follows the common browser vendor practice of maintaining information about each browser client, including information about the operating system, approximate geographic location, preferred language, and other client-specific settings. This information can help to inform decisions about which features should be supported or deprecated in the browser over time, especially when it comes to localizing the browser in different languages and customizing it for

[101] See, e.g., Microsoft, "What's imported to Microsoft Edge"; Mozilla, "Import data from another browser."

different operating system platforms. Requiring Google to share this information with ChromeCo would give ChromeCo necessary user and market information to assist with this kind of decisionmaking.

C. Services Transition

While the software update/distribution and user data transitions have to happen on a relatively short timeline, over time ChromeCo must also transition the other services it uses, at least for those services that Google does not continue to operate. This section examines the transition strategy for other services.

1. Core Services

Because Core Services are necessary for a modern browser, they need to function seamlessly through the transition to ChromeCo. Unlike software updating and user data syncing, Safe Browsing and translation are not user-specific or browser-specific services, so Google can continue to operate them through a transitional period, just as Google already provides Safe Browsing for non-Chrome browsers. Given this, it seems desirable for the potential divestiture order to require Google to provide these services to ChromeCo throughout the transition period while ChromeCo establishes alternatives. This period should be designed to allow ChromeCo to make a smooth transition while not requiring Google to operate these services indefinitely (unless it wants to), probably on the order of four years (assuming some delays resulting from Google being uncooperative).

Enterprise Management and the Chrome Web store present a more complicated case due to their integration with Google's accounts infrastructure and potentially other infrastructure. Depending on how much time ChromeCo has prior to assuming control and on ChromeCo's chosen implementation strategy, it may be possible to execute a complete transition of these services prior to the change of control. Depending on the timeframe, however, it may be prudent for the court to order Google to operate these services for a well-defined and limited time period in order to ensure a smooth transition for enterprises, extension developers, and end users. In the case of the management console, it may be beneficial to require Google to permit existing Google customers to continue to use Google's version of the enterprise management system for an orderly sunset period past the ownership transition, allowing those customers to make alternative arrangements if they so choose.

2. Operational Services

Unlike software updates, it is possible to operate a browser for a short period of time without crash reporting and telemetry. ChromeCo would need to be relatively cautious about the changes it makes to NewChrome during this period, since the major purpose of these functions is to detect failures introduced by new code. Moreover, it is not practical for Google to operate these services for ChromeCo. For these reasons, ChromeCo would need to develop these services itself, potentially with technical support from Google that the court should require Google to provide. Ideally, the potential divestiture order would provide ChromeCo with sufficient time to deploy some version of these services prior to the transfer of control.

3. Ancillary Services

By definition, the Ancillary Services are not essential to offering a competitive browser. In many cases it seems likely that ChromeCo would want to disable them and replace them with services of its own choosing. As such, Google does not need to be required to provide them, even on an interim basis.

V. Summary of Preconditions for Success

While this report concludes that it is technically feasible for a potential ChromeCo to make a competitive version of Chrome, successfully doing so depends on a number of factors that are discussed throughout the report and summarized here for convenience.

Personnel

While ChromeCo will have the existing Chromium code as a starting point, it will need to continue developing Chrome so as not to fall behind other browsers. Developing a browser entails a significant level of specialized engineering expertise and hiring personnel with this expertise is necessary for a successful browser product. Although that expertise exists outside Google, the majority of Chromium engineering talent is inside Google, and so ChromeCo will need to prioritize retaining members of the existing Chrome team. Google should be required to share the necessary organizational and technical documentation to develop its strategy for retaining key employees and to smooth the onboarding of new ChromeCo team members.

Cooperation by Google

A clean transition from Google to ChromeCo depends upon cooperation by Google in a number of areas, including technical documentation, data transfers, and operating services during the transitional period. This dependency also represents an opportunity for Google to prevent a successful transition by providing minimal or superficial cooperation or by delaying cooperation. Any divestiture order would need to provide mechanisms to ensure and oversee Google’s full cooperation.

Functional open source governance of Chromium

Google’s decision to make Chromium open source has allowed for a large ecosystem of independent Chromium-based browsers. To avoid causing inadvertent harm to the browser ecosystem and browser competition, Chromium needs to be maintained as a publicly accessible open source project, including ChromeCo's own future contributions. ChromeCo should retain the flexibility to collaborate with other interested parties to determine the best governance structures for Chromium and V8–structures where Google no longer has decisional authority about updates to the software projects.

Monetization strategy

While this report has not attempted to estimate the costs of maintaining a browser, they are known to be significant. Any successful ChromeCo will need a monetization strategy that is sufficient to fund that development.[102]

VI. Summary of Court Order Requirements

This report has identified numerous requirements that would be beneficial or essential for the court to include in its divestiture order to facilitate the development and success of NewChrome. These are summarized below for ease of reference.

Line-of-business restrictions preventing Google from reentering the browser market for at least 5-10 years, or until appropriate search competition benchmarks have been reached

- A change of ownership for a software product of this size and complexity will require time for the new owner to fully operationalize and monetize.
- These restrictions would require vigilant monitoring, particularly if Google is allowed to retain ChromeOS, to ensure that Google does not evolve this offering into a stand-alone browser.

Public interest conditions to support a thriving browser ecosystem

- Chromium should be maintained as publicly accessible open source, including ChromeCo's own future contributions to the project.
- Chromium's governance structure should be allowed to be redesigned in light of the ownership transfer.
- Google should be prevented from having sole decisional authority over future Chromium updates.
- All of the above conditions should apply to V8 as well.

Transitional support and services

- Google should continue to support each Core Service through a limited transition period, which may vary depending on the service.
- Google should offer SafeBrowsing as a publicly available service, either on an interim basis until ChromeCo develops its own offering, or for the remedy term.
- Google should continue to operate its Chrome Enterprise management console on behalf of ChromeCo during the transition, or permit existing Google customers to continue to use Google's version of the enterprise management system for an orderly sunset period.
- Google should redirect visitors to its Chrome download page to ChromeCo, and provide the technical support necessary to transition users to ChromeCo for software updates, for as long as the court can reasonably order it.

[102] For more on this topic see Cooper, "The True Cost of Browser Innovation".

Provision of data, code, and other artifacts to ChromeCo

- ChromeCo should be able to obtain licenses for proprietary components that Google provides to other vendors, including Widevine.
- Google should share proprietary server-side code, documentation, and API specifications for Core and Operational Services to facilitate ChromeCo building its own versions of these services.
- Google should share product data needed for ChromeCo to understand the browser's operations and user base, including historical telemetry data and current client operational data.
- Google should share organizational charts, product plans, engineering and architecture documents, and other relevant artifacts necessary to develop employee retention plans and train new ChromeCo staff.

VII. Conclusion

This report has examined whether it is technically feasible for Google to divest Chrome to a non-Google entity (ChromeCo) such that Chrome's existing four billion users can continue to make use of a competitive browser (NewChrome). As defined in the introduction, "feasibility" in this context means: (1) the software engineering required to deliver NewChrome is possible on a reasonable timeframe post-divestiture; (2) the technical assistance needed from Google can reasonably be ordered by the court; and (3) ChromeCo can reasonably expect to retain or recruit the personnel necessary to maintain NewChrome. This is the appropriate standard by which the court should evaluate technical feasibility—not whether ChromeCo can precisely replicate Google's approach, but whether it can operate a competitive browser for a global user base in line with its own business interests.

The assessment demonstrates that the divestiture of Chrome is technically feasible across all three dimensions.

First, ChromeCo could feasibly replace all necessary proprietary components and services currently provided by Google. ChromeCo could license or substitute alternatives for proprietary components such as Widevine and codecs, as many existing Chromium-based browsers already do. For proprietary services, such as sync and translation, other browsers like Edge, Brave, and Firefox offer evidence that non-Google providers can build and operate comparable systems. In some cases, such as Safe Browsing, it would be prudent for the court to require Google to provide continued access or transitional support.

It is similarly feasible for ChromeCo to replicate Google's Operational Services. This is a complex engineering task, requiring careful planning, time, and engineering resources. The expertise to do so, however, is not at all novel. Any serious potential Chrome buyer can be expected to be capable of

building and maintaining these Operational Services. Whether ChromeCo chooses to replicate Google-specific Ancillary Services currently included in Chrome can be left to ChromeCo's discretion.

Second, Chromium-based browsers can continue to operate competitively without Google's ongoing investment in Chromium. While Google currently contributes the majority of the code to Chromium, other vendors and independent projects demonstrate that browsers can thrive with smaller teams. ChromeCo would need to recruit or retain browser engineering talent, but the relevant expertise exists both inside and outside Google. Similarly, while ChromeCo would no longer benefit from Google's internal tools or organizational synergies, it could rely on widely available commercial and open-source development tools and contribute to the existing thriving culture of cross-industry web collaboration.

Finally, transition planning is a critical element of feasibility, and the report identifies clear, actionable steps for ensuring a smooth transition. ChromeCo would need to manage software updates, user data migration, and the orderly replacement of services, with limited and time-bounded support from Google. Google would need to share documentation, code, and data to allow ChromeCo to bootstrap its operations and train new personnel. Importantly, these steps do not require novel technical breakthroughs—rather, they mirror the practices of existing browser vendors and established norms for large-scale software migrations.

Finally, the court has an important role to play in establishing guardrails that prevent Google from undermining the remedy while promoting browser ecosystem competition. Line-of-business restrictions and vigilant monitoring can ensure that Google stays out of the browser market long enough for ChromeCo to establish itself. Public interest conditions on the buyer would preserve the benefits of Chromium and V8 as shared digital infrastructure.

In sum, the divestiture of Chrome is technically feasible under the defined standard. With support from a court-ordered transition framework and access to existing technologies and expertise, ChromeCo can deliver a competitive browser on a reasonable timeline.

Appendix: Ancillary Services

This section provides a brief description of some additional services not covered in Section III(B), partly drawn from the services that expert witnesses discussed during the remedies trial. This list does not include services that no longer appear to be active in Chrome as of this writing. These should all be considered Ancillary Services.

Service Name	**Description**
Enhanced Spell Check[103]	By default, basic spell checking uses a local dictionary. Enhanced spell check sends text to Google servers for spelling check.
Gemini in Chrome[104]	Allows the use of AI models to analyze web content as the user browses. This is likely to be a prime area of innovation for browsers.
Google Lens[105]	Provides image search built into Chrome and based on Google search.[106]
Web speech API[107]	Provides speech recognition in the browser, using a server provided by Google. Supported in most browsers but not Firefox.
Google Pay[108]	Google-integrated payments.
Live Captioning[109]	Provides live captioning for audio. Runs locally on the user's device.
Live Translate[110]	Provides translation of live captions using a server provided by Google.

[103] Google, "Turn Chrome spell check on and off."
[104] Google, "Meet Gemini in Chrome."
[105] Google, "Search with Google Lens in Chrome."
[106] In trial testimony, Google expert witness Jason Nieh referred to a "translate images" feature. This appears to be part of Google Lens. Nieh, "Minute Entry for proceedings held before Judge Amit P. Mehta."
[107] Mozilla, "Web Speech API."
[108] Google, "Pay on an app or website."
[109] Google, "Manage captions and translations in Chrome."
[110] Ibid.

Bibliography

Adkins, Heather, and Gloria Maier. “Minute Entry for proceedings held before Judge Amit P. Mehta: Evidentiary Hearing held on 4/29/2025.” United States District Court for the District of Columbia, April 29, 2025. https://www.courtlistener.com/docket/18552824/united-states-of-america-v-google-llc/?page=10#minute-entry-423864913.

Apple. “Overview of app transfer.” Accessed June 20, 2025. https://developer.apple.com/help/app-store-connect/transfer-an-app/overview-of-app-transfer.

———. “Using alternative browser engines in the European Union.” Accessed June 18, 2025. https://developer.apple.com/support/alternative-browser-engines/.

———. “Work. Like never before.” Accessed June 23, 2025. https://www.apple.com/business/enterprise/#device-management.

Arc. “Ready to let go of the old internet?”. Accessed June 29, 2025. https://arc.net/download.

Backtrace. “Error and Crash Reporting for Games & Mobile Apps.” Accessed June 18, 2025. https://backtrace.io/.

Beverloo, Peter, Martin Thomson, Marcos Caceres, et al. “Push API.” W3C, September 3, 2024. https://www.w3.org/TR/push-api/.

Boström, Henrik. “Intent to Deprecate: WebRTC Plan B SDP semantics.” December 22, 2020. https://groups.google.com/a/chromium.org/g/blink-dev/c/IY2amligFFs.

Brave. “Brave Sync Server v2.” Last modified June 2025. https://github.com/brave/go-sync.

———. “Browser Extensions.” Accessed June 20, 2025. https://brave.com/learn/category/browser-extensions/.

———. “How do I set up Sync?”. Accessed June 18, 2025. https://support.brave.com/hc/en-us/articles/360021218111-How-do-I-set-up-Sync.

———. “Leo, Brave’s in-browser AI assistant, now incorporates real-time Brave Search results for even better answers.” June 13, 2024. https://brave.com/blog/leo-real-time-results/.

Chromium. “Contributing to Chromium.” Accessed June 18, 2025. https://chromium.googlesource.com/chromium/src/+/main/docs/contributing.md.

———. “Crashpad Overview Design.” Accessed June 18, 2025. https://chromium.googlesource.com/crashpad/crashpad/+/HEAD/doc/overview_design.md#Upload-to-collection-server.

———. “Omaha Protocol 3.1.” Accessed June 20, 2025. https://chromium.googlesource.com/chromium/src/+/main/docs/updater/protocol_3_1.md.

———. “Omaha Protocol 4.0.” Accessed June 20, 2025. https://chromium.googlesource.com/chromium/src/+/refs/tags/135.0.7020.1/docs/updater/protocol_4.md.

———. “The Chromium Projects.” Accessed June 18, 2025. https://www.chromium.org/chromium-projects/.

CloudFlare Data Insights Team. “Browser Market Share Report for 2024 Q1.” May 3, 2024. https://radar.cloudflare.com/reports/browser-market-share-2024-q1.

Cooper, Alissa. “The True Cost of Browser Innovation.” *Tech Policy Press*, February 24, 2025. https://www.techpolicy.press/the-true-cost-of-browser-innovation-why-chromes-divestiture-wouldnt-end-the-open-web/.

Cursor. “The AI Code Editor.” Accessed June 18, 2025. https://www.cursor.com/en.

Dia. “Dia.” Accessed June 29, 2025. https://www.diabrowser.com/.

Electron. “Build cross-platform desktop apps with JavaScript, HTML, and CSS.” Accessed June 18, 2025. https://www.electronjs.org/.

———. “Minimum breakpad crash reports collecting server.” Last modified December 5, 2022. https://github.com/electron/mini-breakpad-server.

Eng, Austin, Deepti Gandluri, and Francois Beaufort. “WebAssembly and WebGPT enhances for faster Web AI, part 1.” Last modified May 16, 2024. https://developer.chrome.com/blog/io24-webassembly-webgpu-1.

Herrmann, Michael. “Brave omaha.” Last modified September 21, 2023. https://github.com/brave/brave-browser/wiki/Brave-omaha.

Fellou. “Fellou: The World’s First Agentic Browser.” Accessed June 29, 2025. https://fellou.ai/.

FydeOS. “FydeOS.” Accessed June 18, 2025. https://fydeos.io/.

GitHub. “GitHub Copilot.” Accessed June 18, 2025. https://github.com/features/copilot.

Google. “A2A.” Last modified June 16, 2025. https://github.com/google-a2a/A2A.

———. “Android System WebView.” Last modified June 17, 2025. https://play.google.com/store/apps/details?id=com.google.android.webview&hl=en_US.

———. “chromium/chromiumos.” Last modified May 2024. https://chromium.googlesource.com/chromiumos/.

———. “Cloud Translation API.” Last modified April 30, 2025. https://cloud.google.com/translate/docs/reference/rest.

———. “Defendant’s Proposed Findings of Fact.” United States District Court for the District of Columbia, May 29, 2025. https://storage.courtlistener.com/recap/gov.uscourts.dcd.223205/gov.uscourts.dcd.223205.1373.0.pdf.

———. “Enhanced Safe Browsing on Chrome.” October 6, 2020. https://youtu.be/w8uNzQqsTrU.

———. “Find & control your Web & App Activity.” Accessed June 18, 2025. https://support.google.com/websearch/answer/54068.

———. “Geolocation request and response.” Accessed June 29, 2025. https://developers.google.com/maps/documentation/geolocation/requests-geolocation/.

———.“Shopping insights & price tracking in Chrome.” Accessed June 29, 2025. https://support.google.com/chrome/answer/11625545.

———. “Making the world’s information safely accessible.” Accessed June 18, 2025. https://safebrowsing.google.com/.

———. “Manage captions and translations in Chrome.” Accessed June 29, 2025. https://support.google.com/chrome/answer/10538231.

———. “Manage your location settings in Chrome.” Accessed June 29, 2025. https://support.google.com/chrome/answer/142065.

———. “Meet Gemini in Chrome.” Accessed June 18, 2025. https://gemini.google/overview/gemini-in-chrome/.
———. “New capabilities status.” Last modified November 12, 2018. https://developer.chrome.com/docs/capabilities/status.
———. “Pay on an app or website.” Accessed June 29, 2025. https://support.google.com/googlepay/answer/7644068.
———. “Protecting privacy online.” Accessed June 23, 2025. https://github.com/WICG/floc#qualifying-users-for-whom-a-cohort-will-be-logged-with-their-sync-data.
———. “Search with Google Lens in Chrome.” Accessed June 18, 2025. https://support.google.com/chrome/answer/15086890.
———. “Sign in and sync in Chrome.” Accessed June 18, 2025. https://support.google.com/chrome/answer/185277.
———. “The future of endpoint security.” Accessed June 23, 2025. https://chromeenterprise.google/products/chrome-enterprise-premium/.
———. “Transfer apps to a different developer account.” Accessed June 20, 2025. https://support.google.com/googleplay/android-developer/answer/6230247.
———. “Turn Chrome spell check on and off.” Accessed June 18, 2025. https://support.google.com/chrome/answer/12027911.
———. “User Metrics processes data to improve Chrome services.” Accessed June 18, 2025. https://support.google.com/chrome/a/answer/13617304.
———. “Welcome to the Chrome Web Store.” Accessed June 20, 2025. https://chromewebstore.google.com/?utm_source=ext_app_menu.
Honey. “We search for coupons at 30,000+ sites to help you save money.” Accessed June 29, 2025. https://www.joinhoney.com/.
Iyengar, J., and M. Thomson. “RTC 9000 - QUIC: A UDP-Based Multiplexed and Secure Transport.” Internet Engineering Task Force, May 2021. https://www.rfc-editor.org/rfc/rfc9000.html.
Jennings, Cullen, Florent Castelli, Henrik Boström, et al. “WebRTC: Real-Time Communication in Browsers.” W3C, March 13, 2025. https://www.w3.org/TR/webrtc/.
Lingvanex. “Advanced Learning Technologies.” Accessed June 23, 2025. https://lingvanex.com/.
Linux Foundation. “Introducing the Open Governance Network Model.” October 15, 2020. https://www.linuxfoundation.org/blog/blog/introducing-the-open-governance-network-model.
———. “Supporters of Chromium-Based Browsers.” Accessed June 18, 2025. https://www.linuxfoundation.org/supporters-of-chromium-based-browsers.
Linux Kernel Organization. “A guide to the Kernel Development Process.” Accessed June 29, 2025. https://docs.kernel.org/process/development-process.html.
Mellon, Rory. “Amazon is killing one of its most underrated shopping features.” *tom’s guide*. Last modified March 24, 2023. https://www.tomsguide.com/news/amazon-is-killing-one-of-its-most-underrated-features-this-sucks.
Mickens, James, and Gloria Maier. “Minute Entry for proceedings held before Judge Amit P. Mehta: Evidentiary Hearing held on 4/25/2025.” United States District Court for the District of Columbia,

April 25, 2025. https://www.courtlistener.com/docket/18552824/united-states-of-america-v-google-llc/?page=10#minute-entry-423564906.

Mickens, James, Gloria Maier, and Amit Mehta. "Minute Entry for proceedings held before Judge Amit P. Mehta: Evidentiary Hearing held on 4/25/2025." United States District Court for the District of Columbia, April 25, 2025. https://www.courtlistener.com/docket/18552824/united-states-of-america-v-google-llc/?page=10#minute-entry-423564906.

Microsoft. "Edge Add-ons." Accessed June 20, 2025. https://microsoftedge.microsoft.com/addons/Microsoft-Edge-Extensions-Home.

———. "Microsoft Defender SmartScreen." April 15, 2025. https://learn.microsoft.com/en-us/windows/security/operating-system-security/virus-and-threat-protection/microsoft-defender-smartscreen/.

———. "Microsoft Edge for Business." Accessed June 23, 2025. https://www.microsoft.com/en-us/edge/business/?form=MA13FJ.

———. "Sign in to sync Microsoft Edge across devices." Accessed June 18, 2025. https://support.microsoft.com/en-us/microsoft-edge/sign-in-to-sync-microsoft-edge-across-devices-e6ffa79b-ed52-aa32-47e2-5d5597fe4674.

———. "Pricing." Accessed June 20, 2025. https://azure.microsoft.com/en-us/products/ai-services/ai-translator#Pricing.

———. "What's imported to Microsoft Edge." Accessed June 18, 2025. https://support.microsoft.com/en-us/microsoft-edge/what-s-imported-to-microsoft-edge-ab7d9fa1-4586-23ce-8116-e46f44987ac2.

Model Context Protocol. "modelcontextprotocol." Last modified June 18, 2025. https://github.com/modelcontextprotocol/modelcontextprotocol/tree/main.

Montoya, Karina. "It's Time to Imagine Chrome Without Google." *ProMarket,* June 12, 2025. https://www.promarket.org/2025/06/12/its-time-to-imagine-chrome-without-google/.

Mozilla. "Access AI Chatbots in Firefox." Last modified June 2025. https://support.mozilla.org/en-US/kb/ai-chatbot.

———. "Firefox Browser Add-ons." Accessed June 20, 2025. https://addons.mozilla.org/en-US/firefox/.

———. "Firefox for Enterprise Support." Accessed June 23, 2025. https://support.mozilla.org/en-US/products/firefox-enterprise.

———. "Import data from another browser." Last modified May 2025. https://support.mozilla.org/en-US/kb/import-data-another-browser.

———. "Mozilla's vision for the evolution of the Web." March 23, 2022. https://www.mozilla.org/en-US/about/webvision/full/.

———. "Socorro." Last modified May 2025. https://github.com/mozilla-services/socorro.

———. "Sync Firefox data." Last modified May 6, 2025. https://support.mozilla.org/en-US/kb/sync.

———. "Taskcluster." Accessed June 18, 2025. https://taskcluster.net/.

———. "Web Speech API." Last modified April 4, 2025. https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API.

———. “Why is there an OpenH264 plugin in Firefox?”. Last modified April 12, 2025. https://support.mozilla.org/en-US/kb/open-h264-plugin-firefox.

Nieh, Jason. “Minute Entry for proceedings held before Judge Amit P. Mehta: Evidentiary Hearing held on 4/30/2025.” United States District Court for the District of Columbia, April 30, 2025. https://www.courtlistener.com/docket/18552824/united-states-of-america-v-google-llc/?page=10#minute-entry-423999081.

node. “Run JavaScript Everywhere.” Accessed June 29, 2025. https://nodejs.org/en.

Omaha Consulting. “Google Omaha (“Chromium Update”) Server.” Last modified 2024. https://github.com/omaha-consulting/omaha-server.

———. “Google Omaha Tutorial.” Accessed June 18, 2025. https://omaha-consulting.com/google-omaha-tutorial-chrome-updater.

———. “Google Update for Windows.” Last modified 2024. https://github.com/google/omaha.

———. “Pricing.” Accessed June 18, 2025. https://omaha-consulting.com/pricing.

———. “We solve your automatic update needs.” Accessed June 28, 2025. https://omaha-consulting.com/.

openFyde. “openFyde.” Accessed June 18, 2025. https://github.com/openFyde.

OpenJS Foundation. “Electron joins the OpenJS Foundation.” December 11, 2019. https://openjsf.org/blog/electron-joins-the-openjs-foundation.

Python Software Foundation. “PEP 13 - Python Language Governance.” December 16, 2018. https://peps.python.org/pep-0013/.

Rust. “Rust.” Accessed June 23, 2025. https://www.rust-lang.org/.

Skia. “Welcome to Skia: The 2D Graphics Library.” Accessed June 18, 2025. https://skia.org/.

Surapaneni, Rao, Todd Segal, and Michael Vakoc. “Google Cloud donates A2A to Linux Foundation.” Google, June 23, 2025. https://developers.googleblog.com/en/google-cloud-donates-a2a-to-linux-foundation/.

Tabriz, Parisa, and Gloria Maier. “Minute Entry for proceedings held before Judge Amit P. Mehta: Evidentiary Hearing held on 4/25/2025.” United States District Court for the District of Columbia, April 25, 2025. https://www.courtlistener.com/docket/18552824/united-states-of-america-v-google-llc/?page=10#minute-entry-423564906.

Thomson, M., and E. Damaggio. “Generic Event Delivery Using HTTP Push.” Internet Engineering Task Force, December 2016. https://www.rfc-editor.org/rfc/rfc8030.html.

Tung, Liam. “Programming Languages: Rust is coming to Chromium and here’s why.” *ZDNet*, January 13, 2023. https://www.zdnet.com/article/programming-languages-rust-is-coming-to-chromium-and-heres-why/.

uBlock Origin. “uBlock Origin.” September 24, 2024. https://ublockorigin.com/.

United States, Colorado, Connecticut, et al. “Plaintiffs’ Revised Proposed Final Judgment.” March 7, 2025. https://coag.gov/app/uploads/2025/03/2025.03.07-Plaintiffs-Revised-PFJ.pdf.

United States Bureau of Labor Statistics. “Occupational Wages and Employment, May 2022.” Last modified April 25, 2023. https://www.bls.gov/oes/2022/may/oes151252.htm.

Varanasi, Lakshmi. “These are the hardest companies to interview for, according to Glassdoor.” *Business Insider*, April 26, 2025. https://finance.yahoo.com/news/hardest-companies-interview-according-glassdoor-000957968.html.

Visual Studio Code. “Your code editor. Redefined with AI.” Accessed June 18, 2025. https://code.visualstudio.com/.

W3C. “Geolocation.” WC3 Editor’s Draft, April 11, 2025. https://w3c.github.io/geolocation/#geolocation_interface.

Web3Techs. “Usage statistics and market share of Let’s Encrypt as SSL certificate authority on websites.” Last modified June 18, 2025. https://w3techs.com/technologies/details/sc-letsencrypt.

WebRTC. “Real-time communication for the web.” Accessed June 18, 2025. https://webrtc.org/.

Widevine. “Widevine.” Accessed June 18, 2025. https://www.widevine.com/.

Yandex. “Yandex Browser Protect.” Accessed June 20, 2025. https://browser.yandex.com/security/pages.