PANOPTYKON FOUNDATION

BRIEFING NOTE

Service-to-client interoperability as a consumer protection measure

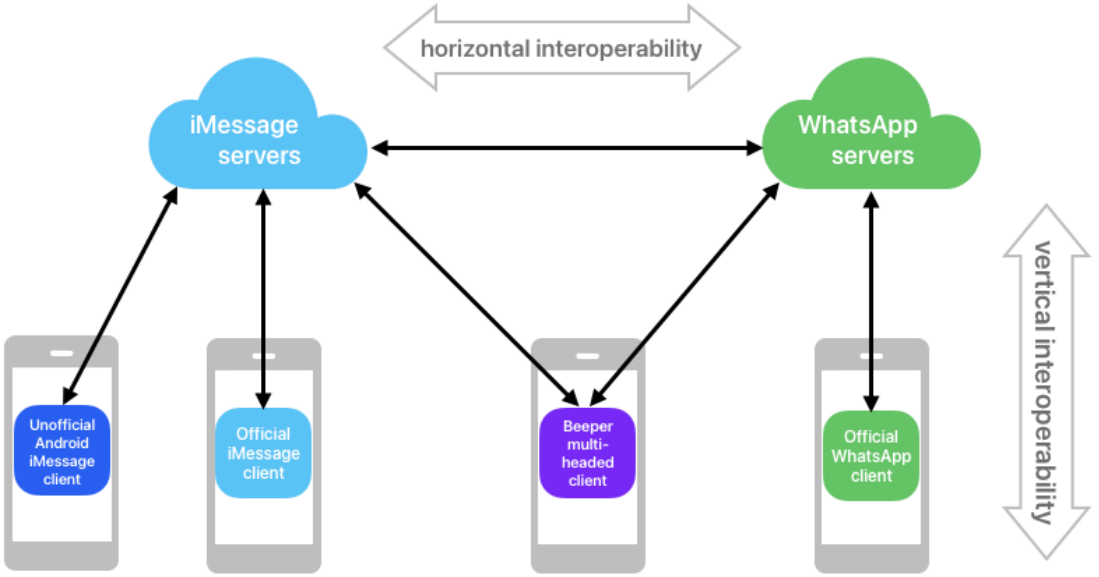
October 2025

In this briefing:

Panoptykon Foundation

A Polish civil society organisation, founded in 2009, with a mission to protect human rights and agency in the context of growing surveillance and intrusive information technologies. Panoptykon works at the national and European level. Through our research, litigation, and strategic communication, we expose risks related to commercial and public surveillance. We keep track of new legislation that regulates technology, develop human-centric solutions, advocate for them, and test them in practice. Learn more at https://en.panoptykon.org/

Author: Vid Logar
Editor: Katarzyna Szymielewicz



Natural state of the internet

It is important to understand that the solution we advocate for in this brief is not an experiment. On the contrary: it rests on tried, tested, and practical solutions that make a real difference for consumers. In other words, what we propose is not a novelty but a return to the internet as it once used to be: more open.

Interoperability was a natural state of the internet before services like then-Twitter and Reddit decided to close down (or charge unfair prices for) APIs that allowed third-party clients to exist, killing valuable products that made those services more enjoyable for users. Moreover, service-to-client interoperability is *already* present on all services that would fall under the proposed obligation. But it usually is restricted to the client *owned by the service provider*.

The mandate we advocate for would require merely **enabling third-party access to interfaces that already exist but are not publicly documented**, because service providers chose to restrict access for themselves.

PANOPTYKON FOUNDATION

Definitions

Service-to-client interoperability (or **service-side client interoperability** in more precise technical terms) refers to a type of interoperability where the **client** application[1] (the front end) – that the user interacts with directly – is able to effectively communicate, exchange information, and function with the **service**[2] application[3] (the back end), using the requisite interface it exposes. In other words, it is the service that enables interoperability with (third-party) clients which are then free to interoperate with it if they so choose[4]. One example would be a user's preferred email client being able to interoperate with an email server (which forms part of the email interpersonal communication service). In simpler, less-technical terms, this may also be called **third-party client interoperability**.

Services may engage in many types of interoperability. **Horizontal (service-to-service) interoperability** allows different systems at the same level of a value chain or stack to communicate and exchange data, such as a user with an identity on one email service (server) sending a message to a user on another email service (server). This is a more complex type of interoperability (with many tradeoffs) and is what Article 7 DMA imposes on designated number-independent interpersonal communication services.

By contrast, service-to-client and client-to-service interoperability are kinds of **vertical interoperability**, which is considerably simpler to implement and comes with fewer tradeoffs**.** Vertical interoperability enables communication and data exchange between different levels of a value chain or across different layers of a system. These are also other kinds of vertical interoperability that a service may engage in, such as interoperability of a service with (third-party) recommender systems or moderation services.

---

[1] Typically a software application that is installed on an operating system, or a web application run in a web browser and served to the user by the web server (which is often developed and operated separately from the backend server logic).

[2] This therefore only applies to (digital) services which are designed for use with a client and must be functionally severable from it. This encompasses a number of services currently in scope of EU digital regulation, such as social networking services, interpersonal communication services, other user-facing online intermediation services, audiovisual media services and virtual assistants.

[3] Such as an API endpoint, content-delivery network, cloud service or other kind of backend. While typically operated by its developer, the service application may also be hosted by the user (self-hosting) or some other third party. In the vast majority of cases the 'service' corresponds to the 'server' in the client-server model. However in some cases, the 'service' does not have a clear 'server' component, such as in peer-to-peer networks. For example, interpersonal communication services often establish a direct peer-to-peer connection between two clients for video and audio calls for better call quality. There the server may at most act in a coordinating capacity, such as establishing the connection. Nevertheless a 'service' is still present as it assigns an identity to each participant and orchestrates the communication, including ensuring that clients of various participants can interoperate with each other.

[4] This is to be differentiated from **client-to-service interoperability** (or **client-side service interoperability** in more precise technical terms) where the obligation is placed on the client providers/developers, for example on terminal equipment manufacturers to ensure they can interoperate with various public telecommunications networks/services. Both are subtypes of **client-service interoperability**, which is interoperability that happens on the relation between the service and the client.

PANOPTYKON FOUNDATION

Use cases

Examples below show how third-party client interoperability[5] can enhance (personalise) user experience or pave the way for entirely new products and solutions:

- **Third-party recommender systems, moderation services and feed curation**: Users can regain some autonomy over what they see in their feeds if third party clients can apply their own filters, such as better or more aggressive spam and scam filters.
    - Bot Sentinel is tailormade for detecting inauthentic accounts, analyzing networks of accounts and identifying trolls.
    - Block Party is a more advanced toolset to combat harassment than what is allowed by most platforms. Content moderation won't be solved just with more regulation and more innovation is needed for a fairer digital experience for everyone.
    - While fully fledged recommender systems that can compete with those of the service would require data exchange and interoperability beyond that on just the client-service relation, third-party clients can get us part of the way to true algorithmic plurality. Third-party recommender systems and content moderation services would also be able to control the display of certain metadata (e.g. fact-checking labels) or collect signals (e.g. likes/upvotes and dislikes/downvotes) that the service provider may bar them from being able to do otherwise.
    - The Legacy Verified browser extension returns verification tags to users that were verified under Twitter's old verification regime. Return YouTube Dislike, besides doing what its name says that it does, also puts into practice a more powerful idea: crowd-sourced video quality signals independent of the host platform.
    - Community moderators themselves also stand to benefit. For instance Reddit moderators (as well as regular users) often rely on tools like Reddit Toolbox, the AutoModerator bot, mod_mailer and others.
    - On online marketplaces, web extensions like Savino can analyze reviews and combine other signals to give the user quick information on the trustworthiness of a seller. Keepa tracks the price history of an item on Amazon, allowing consumers to figure out whether a sale is actually worth it or if it was preceded by a price hike.
    - DeArrow provides alternative and community-sourced titles and thumbnails on YouTube that are not sensational and accurately reflect the contents of a video.

[5] Examples include modifications of first-party clients, achieved almost exclusively through desktop web browser extensions, which could easily be integrated into third-party clients (including on mobile devices where such modifications are less common).

- **Accessibility**: Some third-party clients that died with Reddit's closing of its API were focused towards being more accessible for people with disabilities.
- **Multi-headed clients**[6]: Beeper lets users receive messages from several services in a single inbox. Notably missing is iMessage due to Apple's aggressive efforts to keep third-party clients out. On the social media side, OpenVibe combines Bluesky, Mastodon, Threads, Nostr, and even RSS feeds in a single app, removing the need to scroll through those feeds individually on different apps. Through it users can also cross-post to multiple services at once, an otherwise quite laborious task. This can also be useful in a professional or business context, for example for social media managers or public relations officers. Media from different streaming services could be searchable and viewable through a single app, without the user having to worry about what streaming service a certain piece of content is on. Tracking tags, like Apple's AirTag, could be part of multiple networks (such as Google's Find Hub network) and increase the chances that a lost item will be found.
- **Chatbots**: Various apps and IoT devices can benefit from users being able to interact with them through an interpersonal communication service like WhatsApp[7], accessed by that device or application through a light-weight client (needed for performance reasons or to conserve battery). For example a user could text with third-party AI models or IoT devices through a service like WhatsApp.
- **Combatting addiction and dark patterns**: Service providers are well-aware that the client and its user interface are prime real-estate and thus use it to shape the user experience to their ends, notably retaining the user for as long as possible. The aforementioned Beeper client allows users to stay in contact with their friends and family through Instagram direct messages without being tempted by short-form video content (Instagram Reels) that autoplay every time the Instagram app is opened. Furthermore, a third-party client may remove such distracting elements altogether, focus on chronological feeds from accounts that the user actually follows (that can be reviewed quickly), disable autoplay and institute controls and time limits which cannot be easily dismissed with a tap or two.
- **Privacy and security**: Third-party clients can improve user privacy by not collecting telemetry, URL tracking parameters and actually respecting EU data protection law. They may also adopt security features that the native client does not as well as be more transparent by open-sourcing the client's code. With account-based services, the user could be signed into multiple accounts on a single client without the latter snitching on which identities connected. Furthermore, depending on the final shape of the obligation, it could also be used to make the use of a service without that requiring an account. There is already a range of alternative front-ends that allow this. It would also enable various privacy management tools that can run periodic deletions of recent posts and search through past posts so the user can then decide what to delete.

---

[6] Multi-headed clients are clients that interoperate with multiple services. Email clients are a typical example since once can connect many different email services to them and see emails in a unified inbox.

[7] Note however that WhatsApp has recently banned general-purpose chatbots.

- **Escaping ecosystem lock-in**: Certain services (such as iMessage) are only accessible within the ecosystem of that service provider (in the case of iMessage, only from Apple devices). Being able to access those services from outside the ecosystem – for example through clients made for other platforms – would lessen the lock-in effect of that ecosystem.
- **Better quality, more functional and more personalized clients**: Third-party clients may be of a better quality and offer additional functionality and more personalization options (such as theming) not available on the native client. The native macOS (Apple) Music app is widely known as a very bug-ridden experience due to the low level of investment by Apple. Third party clients like Cider thus solve this issue and use Apple Music simply as a media backend, while also allowing the user to maintain access even if they switch to an otherwise unsupported OS like Linux. Tweetdeck, a Twitter client that once allowed users to display multiple feeds simultaneously (from multiple accounts), was initially a third-party client (later acquired by Twitter and transformed into X Pro). PocketTube for YouTube lets users organize subscriptions into several groups for better curation.
    - **User convenience**: Third-party clients may be more tailored towards keyboard use and may employ keyboard shortcuts than the native client. They may also not implement dark mode or other features that the native client doesn't. Third-party social networking service clients (like Tweetbot once did) often enable users to save their place in the feed, sync it across various devices and thus prevent losing one's position in the feed when moving across devices or reopening the (native) app, which often triggers an unintended refresh of the feed.
    - **Translation and language-learning tools**: A third-party social media client may, for instance, auto-translate all content that is not in the user's own language, without having to do so manually by pressing an additional button as is usually the case. They can also allow a user to select a better translation service – such as the German DeepL – other than the one offered by the service itself. FlipChat is a third-party WhatsApp client that lets users chat in any language, as it translates every message in the user's preferred language. Such tools bring different (European) societies closer.
- **Push notifications on other devices**: Push notification services like the Apple Push Notification Service (APNs) and Google Cloud Messaging (GCM) could be forced to allow third-party clients (operating systems) to also receive notifications for a particular device. Thus, for example, if a user is on a run with their cellular-enabled Android wearOS smart watch, and away from their iPhone, they could still receive notifications that were delivered to the iPhone, without the need for a connection between the need for a connection between the two devices.
- **Dynamic service selection layer**: In cases where a task might be done with a varying degree of quality by different services and where that differs from request-to-request, a multiplex layer may try to process a request through several services and in the end choose the one that it thinks did the best. For example, a

user request may be sent to several virtual assistants or AI models and then the returned responses can be compared and the best one presented to the user.

As demonstrated in the example use cases above, third party clients do not necessarily have to be merely reimplementations of native clients. Often they only implement certain functionalities (for instance Instagram DMs without the rest of Instagram). Also, they do not have to be clients in the traditional sense (i.e. used directly by users) as they may be used for automation purposes or controlled by an (AI) agent to act on the user's behalf.

Benefits of mandating service-to-client interoperability

Third-party client interoperability is in many cases already a reality, sometimes sanctioned and even encouraged by the service operator, other times resulting from reverse engineering efforts (adversarial interoperability) that the service provider attempts to stamp out with technical, legal, and organizational measures. A legislative mandate would be beneficial in both cases and enable interoperability where it has previously proven too difficult as well as make the development of interoperable solutions much easier.

In certain cases, like with Apple's blocking of third-party iMessage clients, interoperability (at least in a way that is accessible to the regular user) has proven too hard due to the threat of legal liability (such as the Computer Fraud and Abuse Act in the US) and technical countermeasures employed by the service provider. A legal mandate can cut through those barriers by requiring the exposure of an interface for the purpose of interoperability, its documentation[8], set fair use policies and provide legal safe harbor to developers powerless in the face of giant corporations.

Frequent (breaking) changes in the interfaces used for interoperability, either because the service provider does not care about third-party clients or as part of a campaign to (partially) break them, have also led to certain functionalities (or a client as a whole) breaking, leading to a loss of users and deterring future development and investment. The DFA should therefore also require that developers are given advance notice of API changes with enough time to adjust their apps. Where possible, service providers should also use API versioning, a common pattern in API development, which allows older and newer versions of an API to coexist.

Even for existing third-party clients sanctioned by the service provider, regularization of these relationships within a legal framework would achieve more certainty, stability and predictability for developers, without fear that one day the service provider would change tact and capriciously pull the rug under them, ending their business.

These are the preconditions necessary for good market dynamics which would allow for the development of new products and services. A successful third-party client can also serve as a springboard for the development of further products and services. This can contribute to Europe's technological sovereignty, which will not be achieved overnight but must start with a gradual disintermediation and functional breakups of various components that make up services that Europe is dependent upon.

[8] While reverse engineering can be used to figure out how an (often proprietary) undocumented interface works, this is a best-effort endeavor that cannot account for many edge cases or unexpected behavior and is thus likely to lead to a certain level of breakage and poor user experiences.

Third-party clients do not take anything away from the service, other than the immense power that its provider has. Not only can the improved experience draw more users to the service (whilst increasing the competitive pressure on the native client) but the ecosystem built around the service may produce new innovations and additions, later natively adopted by the service. The retweet, a cornerstone of what once was Twitter (with equivalents on many other social networking services), was a bottom-up community innovation of independent client developers[9], rather than a result of top-down product development.

Why the Digital Fairness Act?

European legislators did not include a general interoperability obligation in the DMA, likely due to it being too open-ended of a concept that lacks clarity with regards to actual obligations that it creates. The solution we propose in the context of the DFA is a **precisely defined and targeted interoperability obligation, centered around common design and architecture patterns that make them liable to functional separation and unbundling**.

Third-party client service interoperability can be applied to a broad range of digital services, not just to a closed list (e.g. core platform services regulated in the DMA) or a particular category (e.g. online intermediation services regulated in the DSA). Therefore, taking into account its horizontal nature, the DFA is a well-suited legislative vehicle for introducing this service category-neutral obligation.

Third-party client service interoperability can be applied to a large number of services (not just large online platforms or gatekeepers, as defined by the DSA and the DMA) because of its **low compliance burden**. The obligation may be further narrowed to focus on services that reach a certain threshold of users and may exclude ancillary services, with what is an ancillary and core service judged from the point view of the user.[10]

The mandate could also be tiered based on the size of the service. For example, for large services with 45 million daily active end users (or 10 000 business users), the full list of the above obligations could apply, while smaller services could only be subjected to negative obligations, such as to not unduly interfere with legitimate reverse engineering efforts and the prohibition of unfair use policies for existing interfaces.

Last but not least, it would make perfect sense for the DFA to prohibit unfair (to developers and to users alike) and restrictive use policies that are currently attached to interoperability interfaces. For instance, the Google Photos API User Data and Developer Policy prohibits its use to “create, train, or improve (directly or indirectly) similar or competing products or services”, such as a multi-headed photo management client that draws shared albums from Google Photos, Apple Photos, and other photo-management services. Such policies may also hinder effective data portability to alternative privacy-respecting services such as ente.[11]

---

[9] Paßmann, Johannes. (2019). Less Mutable, More Mobile: The Role of Twitter Apps in the History of the Retweet Button.

[10] See, for instance, a similar test in the European Electronic Communications Code’s definition of an interpersonal communication service in Article 2(5) and Recital 17.

[11] Photo management (and many other types of) services do not qualify as a core platform service under the DMA.