DMA.100220 - Alphabet - OS - Google Android - Art. 6(7)
Apple Response to Public Consultation on Draft Measures

13 May 2026

Table of Contents

I. Introduction and summary

(1) Apple submits this paper in response to the draft measures ("**DMs**") issued by the European Commission ("**EC**") on 27 April 2026 under Articles 6(7) and 8(2) of the Digital Markets Act ("**DMA**") in case DMA.100220 ("**Specification Proceedings**").[1]

(2) As one of the world's foremost developer of operating systems ("**OSes**"), Apple has a significant interest in the Specification Proceedings. Apple's approach is unique. It is different from Android and Windows; OSes that are licensed to third-party hardware manufacturers. Apple does not license its OS to any third party. Instead, Apple designs its OSes around the products it creates—like the iPhone, iPad, and Mac. This deep and seamless integration of hardware and software allows Apple to build products that are widely regarded as the best in the world. Apple's unique approach to product design also allows it to offer users best-in-class security, privacy, and safety. Indeed, this is part of the promise that Apple has made to users and part of the standard that users expect when they choose Apple products.

(3) The DMs raise urgent and serious concerns. If confirmed, they would create profound risks for user privacy, security, and safety as well as device integrity and performance. Those risks are especially acute in the context of rapidly evolving AI systems whose capabilities, behaviours, and threat vectors remain unpredictable as we are now seeing time and again. The DMs would mandate unprecedented access to ambient audio, screen contents, cross-app actions, and system-level resources, while simultaneously undermining the very safeguards an OS should rely on to manage those risks. In effect, the EC would be conducting a large-scale security and privacy experiment on European users at precisely the moment when AI-driven threats are intensifying the fastest. A fact fully understood by other parts of the EC itself as evidenced by a number of AI safety focused initiatives.

(4) The DMs and the summary of the EC's preliminary findings are based on a fundamentally misguided and uninformed understanding of the OS's role. The EC has taken a set of positions in the DMs that would undermine user privacy, safety, and security on Android. It articulates a framework that will create extreme new vulnerabilities for users. It appears that the only value accounted for by the EC in these DMs is access. Other competing legal instruments and values are not addressed.

(5) Artificial Intelligence offers great promise. We are in the midst of what could be a profound technological shift. Yet at the same time there are real threats—both known and unknown. AI technologies increasingly operate with broad contextual awareness, persistent access to sensitive data, and the ability to take autonomous actions across apps and services. The EC's Joint Research Centre recognises that "*GenAI systems are vulnerable to traditional cybersecurity threats and AI-specific*

---

[1] Apple recognises that the DMs are specific to Android. They have no legal impact on Apple. Any attempt subsequently to inappropriately and unlawfully apply those same measures to iOS or iPadOS would be fundamentally incompatible with Apple's approach to product design and with the values on which Apple's platforms are built.

vulnerabilities."[2] Yet the EC in these DMs ignores these risks in its rush to redesign Google's OS.

(6) The risks associated with non-deterministic AI systems remain only partially understood and are evolving rapidly. These risks include data exfiltration, indirect prompt injection, unauthorised autonomous actions, manipulation of vulnerable users and minors, and cascading failures across interconnected applications and AI agents. This is precisely the context in which OSes must retain the ability to implement robust, system-wide safeguards. The DMs remarkably do not address these risks at all.

(7) Specification procedures are meant to clarify the application of *existing* obligations in a gatekeeper-specific context. They were meant to be a genuine attempt to tease out the correct implementation of what was clearly understood to be highly technical obligations. The EC cannot use this process to rewrite the DMA to its own liking. Yet based on the DMs this is how these procedures are being used.

(8) The compressed timelines inherent in specification proceedings, combined with their case-specific nature, make them ill-suited to address complex and rapidly evolving technological domains that would ordinarily require full legislative consideration, as is the case with AI.

(9) Apple strongly urges the EC to consider the following points in the next phase of this specification procedure.

(10) **First, the DMs would result in significant risks to user privacy and security (Section II)**. The DMs mandate broad direct access to sensitive capabilities, including ambient audio, screen content, cross-app control, and system-level resources effectively neutering the primary duty of a modern operating system in arbitrating access to sensitive data and functionality to protect users and system integrity. This is especially the case in relation to technologies that are still evolving and whose risks are not yet well understood. The DMs are silent on the consensus across the security community that such capabilities require strict, secure-by-design safeguards.

(11) Non-deterministic AI systems create rapidly evolving risks of data exfiltration, indirect prompt injection, unauthorised autonomous actions, surveillance, fraud, and system compromise. Yet the DMs would expose precisely the functionalities malicious actors and compromised AI systems seek to exploit: persistent microphone access, live screen contents, cross-app orchestration, and privileged system resources. This is a very dangerous idea. At the same time, the DMs explicitly constrain Google's ability to impose deterministic, system-wide safeguards against those risks.

(12) The EC is redesigning an OS. It is substituting judgments made by Google's engineers for its own judgment based on less than three months of work. It is all the more dangerous given the only value that can be discerned from the DMs guiding this work appears to be open and unfettered access. Critical user and market

---

2 JRC, Generative AI Outlook Report (13 June 2025).

impacts such as user privacy, safety and security are not addressed. Nor is it apparent that the EC has sought to meaningfully involve the expert bodies specifically established to address them, such as the European Data Protection Board ("**EDPB**") or the European Union Agency for Cybersecurity ("**ENISA**"). There is a direct tension between the DMs and other areas of EU law. For example, the AI Act and the Cyber Resilience Act ("**CRA**") require the reduction of attack surfaces, mitigation of system risks, and data protection. Yet the DMs prevent Google from doing exactly that. They would require Google to provide broad third-party access to highly sensitive functionalities and block Android's ability to impose meaningful safeguards or effective vetting. The harms to users, in terms of invasions of privacy, compromised device performance, and reduced trust in their platform providers, would be innumerable.

(13) The DMs also raise significant safety concerns. Model providers have a societal responsibility as well as a legal responsibility in many jurisdictions, including the EU, to ensure that they provide AI models in a safe manner that does not create risks for users. Yet Google would be required to provide direct access to its on-device models to all 3Ps regardless of their intentions, capabilities, plans or known track record of disregarding safety related issues, without any reference at all in the DMs to the guardrails that Google can and must put in place including terms of use prohibiting the disregard of safety. To take it another step, even if Google is aware or becomes aware of a third party engaging in unlawful or inappropriate behaviour, there does not appear to be any means for Google to make a decision to exclude that third party from such access. Google's obligations under the EU AI Act are not addressed at all. This regulatory incoherence creates a clear conflict of legal obligations on Google. Even in this past week, the EC, member states and Parliament have affirmed their intention to ban so-called nudification apps. Commission President Von der Leyen has attended, together with other leaders, an event in Copenhagen focused on youth safety in the use of AI products. The need to take measures to protect users is beyond debate at this point including at the highest levels of the EC. But again the DMs do not appear to provide any means for Google to block bad actors from using its on-device models. This must be addressed by the EC.

(14) The DMs ignore Google's fundamental rights under the Treaty. They do not address intellectual property at all. The DMs place no value on intellectual property and materially interfere with Google's ability to use and protect its intellectual property. Interoperability obligations implicate copyright, trade secrets, and patents. Yet there is no mention of these fundamental rights in the public documents. The lack of respect for intellectual property is stunning.

(15) The DMs risk turning interoperability into an uncompensated access regime for proprietary technologies, while disregarding the proportionality of the interference with property rights and the freedom to conduct a business. Measures such as mandatory access to the gatekeeper's on-device models—which cost billions of dollars to develop—pose a direct threat to competition and undermine incentives to introduce new innovations in the EU.

(16) The final decision should not be framed in a way that could be interpreted to require gatekeepers to make available their proprietary technologies under equivalent

conditions, without regard to the nature and scope of the IP at stake or the proportionality of such access.

(17) **Second, the DMs would result in significant risks to device integrity and performance (Section III)**. The DMs require concurrent access to device resources so that they can run their own on-device models and digital sound models for concurrent voice invocation. They would fully delegate to app developers the decision to validate a successful CPU wake-up,[3] and allow them to "*autonomously innovate on their DSP sound models*"[4] without OS orchestration or oversight. The DMs also require that, after invocation, the app should continue "*to receive the audio of the hotword detection and to continue to record the audio until the user decides to finish its request or conversation*,"[5] stripping the OS of the ability to make that determination of when to stop a recording. A fundamental function of an OS is to manage system resources. The OS mediates competing demands of finite hardware resources, arbitrating between power efficiency, thermal safety, and user experience in real time. The DMs would significantly undermine Android's ability to do that going forward. The unavoidable result would be a degraded experience for users and developers alike.

(18) **Third, the DMs take an untenable approach to integrity safeguards (Section IV)**. The DMs interpret the "integrity" safeguard in Article 6(7) so narrowly that it becomes practically meaningless. The level of access and control mandated by the DMs would render even fundamental security safeguards ineffective. Encryption, for example, cannot protect user data from a process that already has access to the relevant decryption keys or otherwise operates with privileged access to the underlying data and system functionalities.

(19) The DMs reduce integrity protections to *only* user consent. This assumes that users can meaningfully understand and manage persistent, system-wide, and interacting permissions from multiple apps. This assumption is unfounded and flawed. It is precisely why the GDPR introduced multiple legal grounds for processing personal data to ensure that there was no longer an over-reliance on simple consent in recognition of the fact that users were not in a position to grant it in an informed manner in all situations. That problem becomes exponentially worse in the context of AI-driven functionalities.

(20) The DMs dismantle core security and privacy protections. They would remove critical sandbox protections, remove effective OS-level mediation, and prevent effective vetting of applications requesting access to highly sensitive capabilities. For example, the DMs would require support for overlay functionalities that facilitate phishing and UI-spoofing attacks, permit third-party apps to imitate user interactions across applications, and expose continuous streams of sensitive contextual data (including screen contents, notifications, contacts, and ambient audio) to an effectively unbounded number of apps.

---

3 DMs, ¶14(f).

4 DMs, ¶14(b).

5 DMs, ¶12(e).

(21) The DMs would create systemic vulnerabilities that extend beyond individual users. They would expose all users of the Android platform to well-documented risks, including data exfiltration, unauthorised actions, and system compromise. This reflects either a fundamental misunderstanding of the role of the OS or a determined attempt to dismantle that role. OS providers have the incentive and are trusted by their users and developers to act as a steward of user security, privacy, safety, and device integrity. Third-party app developers do not share the same incentives. The EC must ensure that it does not take actions that undermine the OS's ability to meet users' expectations of stewarding their ecosystems, delivering secure-by-design platforms and rooting out bad actors.

(22) **Fourth, the DMs rely on an interpretation of Article 6(7) DMA that far exceeds its legal limits (Section V)**. Article 6(7) requires interoperability with existing features that are "*accessed or controlled*" via a designated OS and are "*available to or used by*" the gatekeeper for its own separate services. It does *not* require the creation of new functionalities, interoperability for use cases the gatekeeper does not pursue, or access to standalone services that are not mediated via the OS. The DMs depart from these limits by mandating new capabilities, extending interoperability beyond OS-mediated features, and collapsing the distinction between OSes and services. This is not specification. It is a non-legislative expansion of the law.

(23) The DMs create a serious risk of free-riding. They would effectively require OS providers to make available for free sensitive capabilities, integrations, and system functionalities developed through years of investment and engineering effort. The DMA is not a compulsory licensing regime for advanced OSes and AI capabilities. Specification proceedings should not become a mechanism through which competitors obtain broad access to technologies they did not themselves develop and for which they bear little responsibility if things go wrong.

(24) Furthermore, gatekeepers should be free to take different approaches to interoperability and integrity protections according to their specific architectures and business models. A one size fits all approach would eliminate an important dimension of inter-platform competition and reduce incentives for differentiated approaches to security, privacy, safety, and system design.

(25) Apple does not license iOS to third-party device manufacturers. It does not operate a modular, open-source architecture. It has deliberately chosen a different path—one in which hardware-based encryption, the Secure Enclave, sandboxing, and end-to-end data protection are not optional add-ons but structural commitments embedded into the platform.[6] On iOS, that same regime would dismantle the very security model that Apple's hundreds of millions of European users have chosen and that Apple has

---

[6] Providers that design both hardware and software, like Apple, are required to assume full responsibility for security, privacy, and device integrity at the system level, while simultaneously being told by the EC that they are denied widely accepted approaches to manage those risks. Those concerns are particularly acute where, as here, the EC mandates widely available deep and persistent access to sensitive capabilities. By contrast, other actors (including device manufacturers such as Samsung and service-layer providers such as Meta and OpenAI) are able to expand rapidly in data-intensive and AI-driven services without being subject to comparable obligations.

spent decades building. A framework that mandates exposure of high-risk functionalities while constraining safeguards will inevitably deter investments in the EU.

(26) The final decision must be substantially revised to account for these flaws. Effective interoperability for virtual assistant ("**VA**") apps does not require the DMs' maximally permissive access model. Gatekeepers should remain free to adopt different interoperability and integrity solutions according to their architectures and business models. The worst outcome would be an approach that ignores those concerns altogether, or treats them as irrelevant to interoperability, even though the OS is the only layer capable of enforcing system-wide safeguards and purpose limitation.

(27) There are alternative approaches that could address interoperability while *maintaining* the OS's essential role in preserving user privacy, security, safety, and device integrity in the context of agentic interactions. VA's seek to perform a wide array of actions in response to user requests, and the OS must ensure that these actions are bounded by system protections and actual expressed user intent. Rather than the irresponsible risk represented in the DMs of exposing raw access to personal context, on-device data, and cross-app actions, VA requests can be mediated through an AI-native *trusted OS agent*, that provides a natural agent-to-agent interface to VAs to accomplish authorised operations while intelligently protecting users against malicious or accidental harms. This approach could provide VA apps with equivalent invocation methods and equivalent user-interface affordances to those available to the first-party services, enabling developers to compete on the quality of their AI, integrations, domain expertise and user experience, rather than on their ability to obtain or recreate privileged system-level access.

II. The DMs would create unjustifiable privacy and security risks

(28) The DMs mandate deep, unbounded, and concurrent access for any third-party app to an Android device's OS, hardware, and user data. Under the DMs, *any* app could receive *continuous* access to a user's microphone, camera, and screen content, data from other apps and the system, and the ability to execute actions across other apps and the OS.

(29) The DMs go far beyond any plausible interpretation of Article 6(7) and systematically disregard any OS's core role in protecting user security, privacy, and device integrity. Allowing interoperability with third-party apps is fundamentally different from requiring access to deeply integrated system-level capabilities of an OS.

(30) The OS provider has the incentive to act as steward of user security, privacy, safety, and device integrity. It can embed purpose limitation directly in the architecture itself: scoping what content reaches each app, running screen capture within system processes no app can reach, and excluding sensitive surfaces from capture entirely. Third-party app developers are not in the same position, do not share that incentive, and simply cannot provide the same systemwide guarantees. Granting broad access without stringent architectural guardrails is reckless and downright dangerous.

(31) Agentic AI capabilities create new threat vectors, including indirect prompt injection

and unauthorised autonomous actions.[7] AI developers, the security industry, and regulatory bodies have converged on the need for robust, architecture-level safeguards to address these risks.[8] The OS is the sole layer that is architecturally positioned to impose the system-wide controls those safeguards require. And OSes must constantly evolve and react to these widespread—and intensifying—challenges.

(32) The DMs fail to engage with these severe risks. Nor is there any indication that the EC involved authorities with recognised expertise in security, privacy or cybersecurity—such as the EDPB or ENISA. This is notwithstanding repeated requests from Apple and others that such expertise be brought into the process.[9]

(33) The DMs siloed regulatory approach puts users at risk. They pursue measures with profound implications for privacy, security, and device integrity through an ideological lens of open and unfettered access. A process that drew meaningfully on data protection and cybersecurity expertise would not have produced rules that would destabilise performance, compromise device security, and expose users to pervasive surveillance, fraud, and scams.[10]

(34) This section expands on how the DMs would introduce unjustifiable risks across each of the four capabilities they mandate: (i) voice-activated invocation capabilities (**Section II.A**); (ii) data access (**Section II.B**); and cross-app and system actions (**Section II.C**). It then explains how the DMs would create a direct conflict with applicable EU cybersecurity regulations (**Section II.D**).

A. Unauthorised or accidental audio capture risks from voice-activated invocation

(35) The DMs include obligations intended to require that users can invoke any third-

---

7 Xiaoxue Yang *et al.*, Checkpoint-GCG: Auditing and Attacking Fine-Tuning-Based Prompt Injection Defenses (16 October 2025).

8 *See, e.g.*, ACSC, CISA, NSA, NCSC-NZ, and NCSC-UK, Careful Adoption of Agentic AI Services (1 May 2026) (recommending that operators implement safeguards including sandboxed action spaces, isolation of agents, API-level deny lists, and runtime privilege limitation); EDPB, AI Privacy Risks & Mitigations – Large Language Models (LLMs) (10 April 2025), pp. 39–42 (mapping each privacy risk to a corresponding mitigation implemented at the architectural deployment level; OWASP, OWASP Top 10 for Agentic Applications (9 December 2025) (recommending the implementation of platform-level safeguards).

9 *See, e.g.*, CCIA, The Digital Markets Act: A procedural journey towards effective compliance (18 September 2025), pp. 30–32 (recommending an increased role for the High-Level Group to solve regulatory conflicts); CEPA, Making Interoperability Work: Recommendations for Europe's DMA (18 November 2025) (recommending alignment of DMA enforcement with cybersecurity standards and the involvement of ENISA).

10 The EC's approach is hard to reconcile with the DMA's own institutional framework. The DMA established the High-Level Group, which includes both the EDPB and ENISA, to ensure coherence where competition, privacy, and cybersecurity concerns intersect. *See* DMA, Article 40 and recital 93 (establishing a High-Level Group "*in order to ensure coherent and effective complementarity in the implementation of [the DMA] and of other sectoral regulations*"). Yet the High-Level Group appears to have played no meaningful role whatsoever in the specification proceedings.

party app through wake words[11] or long-pressing on certain access points.[12] Once invoked, Google is required to enable third-party apps to receive microphone access[13] and contextual data,[14] and overlay content on the user's device.[15] The DMs' invocation requirements raise a specific risk of unauthorised or accidental audio capture—a well-documented concern in voice-activated devices—being exposed to an unbounded set of third parties.

(36) The DMs seek to turn any app into stewards of the platform by enabling them to control: (i) whether invocation has occurred at all; and (ii) when audio recording should stop by requiring that the app should continue "*to receive the audio of the hotword detection and to continue to record the audio until the user decides to finish its request or conversation*."[16] Allowing third-party apps to make these determinations grants them effectively unbounded audio recording capability. This is difficult to reconcile with established findings on the security and privacy risks associated with voice-activated systems.[17] The DMs would create a new threat vector that could be deliberately exploited or cause accidental data leakage.

(37) The DMs' wake word requirements give rise to three interconnected risks.

(38) First, third-party apps may intentionally abuse their privileges to collect background audio and gain more insights about the user. There are developers that seek to distribute apps as surreptitious surveillance tools.[18] For example, a recent report revealed a significant privacy vulnerability created by Meta's sharing of video footage recorded by its smart glasses products—including extremely private and sensitive moments—with human contractors to train its artificial intelligence.[19] These appear to be the very types of accessories and services that other EC measures have sought to facilitate through expanded access to sensitive OS functionalities.

(39) Second, bad actors have harvested user data by inserting themselves in users' voice

---

11 Referred to in the DMs as "*hotwords*".

12 DMs, Measures 1.1 and 1.2.

13 DMs, ¶¶12(e) and 14(g).

14 DMs, ¶¶4(b) and 12(d).

15 DMs, ¶¶4(c) and 12(d).

16 DMs, ¶¶12(e) and 14(g).

17 *See, e.g.,* Jingjin Li *et al.*, Computers & Security (134), Security and privacy problems in voice assistant applications: A Survey (4 September 2023); Tom Bolton *et al.*, On the Security and Privacy Challenges of Virtual Assistants (26 March 2021).

18 The US FTC, for example, has warned Android app developers against embedding code that listens for inaudible "audio beacons" in television advertisements to build a detailed log of users' viewing habits. *See* FTC, FTC Issues Warning Letters to App Developers Using 'Silverpush' Code (17 March 2016).

19 Matt Jancer, Meta's Smart Glasses Have Even Bigger Privacy Issues Than We Thought (18 March 2026).

interactions with voice-activated technologies.[20] Third parties can also intentionally trigger users' wake words. For example in 2017, Burger King deliberately exploited this vulnerability in a television commercial. It used the phrase "*OK Google, what is the Whopper burger?*" in commercials to hijack viewers' Google Home devices.[21] Granting third-party developers control over the invocation trigger could enable covert tracking: just as physical retailers have used Bluetooth beacons to monitor customer movements,[22] an in-store audio system could embed an imperceptible signature in its music to trigger a third-party app and track a user's presence.

(40) Third, users may accidentally activate third-party apps via wake words leading to unintended data collection.[23] The EDPB has identified accidental and unintended audio capture as a structural privacy concern, warning that expansion of third-party access to the voice channel materially compounds these risks.[24] The figure below demonstrates the prevalence of mis-activations of voice-activated technologies.

---

20 *See, e.g.*, Nan Zhang *et al.*, 2019 IEEE Symposium on Security and Privacy, Dangerous Skills: Understanding and Mitigating Security Risks of Voice-Controlled Third-Party Functions on Virtual Personal Assistant Systems (10–23 May 2019) (showing that a malicious third party could keep the microphone open with silent re-prompts, continuing voice recordings while the user believed the session had ended); Richard Mitev *et al.*, Alexa Lied to Me: Skill-based Man-in-the-Middle Attacks on Virtual Assistants (2 July 2019) (finding that a malicious function could insert itself in a user's conversation with Alexa and silently capture the voice input intended for Alexa).

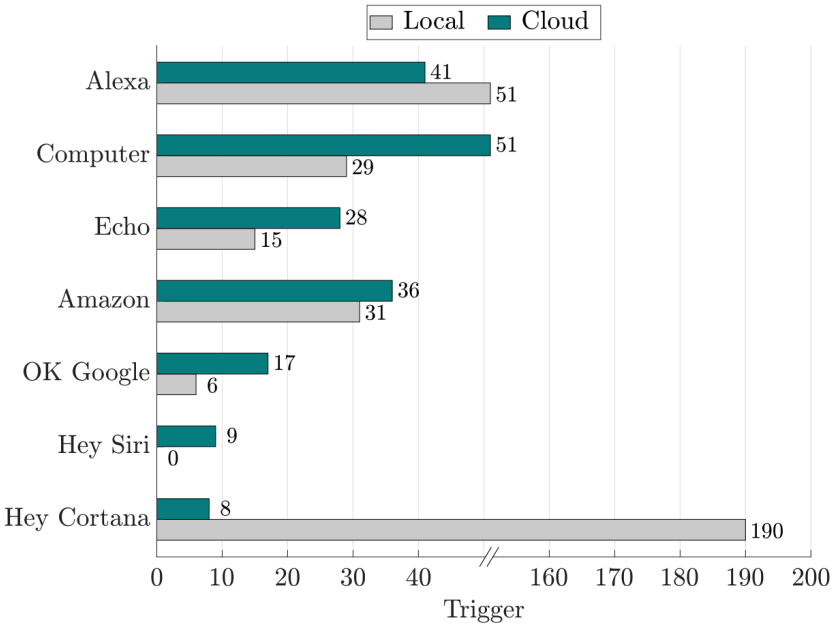
21 Michael Price, Watching Burger King's Newest Ad Puts Your Privacy in Jeopardy (24 April 2017). The DMs would also enable third parties to adopt wake words that are prefixes or homonyms of common phrases or of wake words used by competing services, increasing the rate of mis-activations; *see, e.g.*, Nan Zhang *et al.*, Understanding and Mitigating the Security Risks of Voice-Controlled Third-Party Skills on Amazon Alexa and Google Home (29 June 2018) (demonstrating that similar wake words can be used to hijack voice commands).

22 FTC, Privacy trade-offs in retail tracking (30 April 2015); N.Y. Times, In Stores, Secret Surveillance Tracks Your Every Move (14 July 2019).

23 Lea Schönherr *et al.*, "Unacceptable, where is my privacy?" Exploring Accidental Triggers of Smart Speakers (2 August 2020) (noting that ordinary speech, television, and radio content can lead to "*accidental triggers*").

24 EDPB, Guidelines 02/2021 on virtual voice assistants (version 2.0) (7 July 2021), p. 13.

Figure 1: Number of mis-activations by the local and cloud-based Automatic Speech Recognition ("ASR") engine



Source: Lea Schönherr et al.[25]

(41) The DMs' requirement for concurrent wake word access by *any* third-party app—in combination with its requirement for user-customised wake words—would by definition greatly expand these risks.

(42) Persistent listening, wake-word detection, invocation routing, and related system-level capabilities are highly sensitive OS functionalities that require tight system-level control and supervision. The DMs instead proceed from the premise that these functionalities should be broadly exposed to third parties while sharply limiting the OS's ability to mediate, or vet their operation, including by preventing Google from reserving the always-on access points to the default assistant, and by requiring Google to ensure that the user interface, text and wording of user-customizable wake words are "in control of third-party developers".[26] Such a free-for-all voice-activated system would be unprecedented. In the context of AI especially, there is no telling how users, developers, or OSes would be able to manage these risks.

B. Privacy and security risks resulting from data access requirements

(43) The DMs mandate unbounded access to a user's most sensitive data, regardless of where it is stored or which app it belongs to. This includes access to a centralised

---

25 Lea Schönherr *et al.*, "Unacceptable, where is my privacy?" Exploring Accidental Triggers of Smart Speakers (2 August 2020), p. 7 (Figure 4). Results were collected twice using 24-hour audio samples of media reflecting typical household content such as TV shows, newscasts, general noise, including non-English sources and both female and male speakers.

26 DMs, ¶15(b).

database of on-device app data,[27] continuous and global access to the user's complete "*digital context*," including sensitive data streams like "*device audio, screen contents [...] notifications, app launches, contacts, [and] SMS*",[28] and access to a list of applications installed on a device,[29] together with the actions they expose.[30] Upon invocation, the DMs would also require that any app must receive broad and ill-defined "*contextual data*" including, at a minimum, a "*current screenshot*" and "*information about the open apps*", and can extend to "*additional data*", of unspecified scope.[31] And the DMs would require Google to expose live ambient data captured by the device's most deeply embedded sensors.[32] The scope of that access is not confined to information managed locally by the OS on the device: it somehow purports to extend to data retrieved from the cloud[33] and to data held within other third-party applications.[34]

(44) By seeking to mandate a system in which any third-party app can gain access to the user's most sensitive on-device app data, the DMs seek to introduce severe and unnecessary risks of (i) device fingerprinting and user profiling; (ii) data exfiltration; (iii) UI spoofing and phishing; and (iv) spyware capabilities. The DMs would place users at serious and unnecessary risk: their personal data would be placed into the hands of parties whose identity, capabilities, and intentions they cannot assess, exposing them to surveillance, manipulation, discrimination, and fraud.

(45) This mandate departs from the fundamental security principle of least privilege, under which an app receives only the data strictly necessary for the task at hand. It ignores the fundamental difference between access by a platform provider and a third-party app. The platform provider has the incentive to be a responsible steward, and users expect it to play this role. In Apple's case, we take that responsibility very seriously. Security, safety, and privacy are foundational principles for Apple's OSes. A third-party app developer has neither the architectural capability nor necessarily the commercial incentive to provide equivalent protections.

1. Profiling, manipulation, and fraud risks

(46) The DMs ignore the sensitivity of on-device data—and how readily it identifies individual users and reveals the attributes that make them targets for exploitation. The DMs would mandate access to data—such as notifications, SMS, contacts, screen contents, and app usage—that reveals intimate and uniquely identifying

---

27 DMs, ¶25(a).

28 DMs, ¶42(a)(2).

29 DMs ¶90(c).

30 DMs, ¶60(a).

31 DMs, ¶¶4(b) and 12(d).

32 DMs, Measure 2.4.

33 DMs, ¶¶36(a) and 42(a)(3).

34 DMs, ¶¶42(a)(2) and 42(a)(4).

information about the user, including health conditions, religion, sexual orientation, political views, and financial circumstances. Any single one of these data categories would be sufficient to identify and profile individual users; taken together, they would provide a comprehensive map of a user's private life. Even the mere list of apps installed on a user's phone is as unique as a fingerprint[35] notwithstanding the profiles that every app would be supercharged to create.

(47) The DMs would purport to make this data available to an unbounded number of apps. The possible justification for this wild abrogation of OS responsibility is not clear from the DMs. Unlike OSes—whose architecture and system-wide accountability can promote user privacy—third-party apps can offer no such guarantee even if they sought to do so. This would give bad actors vast new opportunities for malicious practices.

(48) The DMs prevent limitation on the number or type of apps that may access this data. And the DMs include no effective safeguard against misuse. It is the opening of a Pandora's Box. The practices this access would enable include discriminatory pricing calibrated to inferred willingness to pay or perceived vulnerability,[36] manipulative advertising directed at users whose data signals addiction, financial distress, or health conditions,[37] and tailored spamming, phishing, or fraud exploiting the specific services the user is known to rely on.[38] Such widespread and unfiltered data access would be unprecedented and the scale of potential user harm is therefore unmeasurable.

2. Data exfiltration risks

(49) A baseline expectation in the DMs of broad, unfiltered data access by apps evidently multiplies the attack surfaces from which user data can leave the protection of the OS facilitating data exfiltration. An app with access to sensitive data can become a "double agent", appearing to serve its intended purpose while secretly or inadvertently exfiltrating the user's most sensitive information, including (as the DMs would require) app lists, screen contents, contacts, and messages.

(50) The DMs' explicit mandate that any app be able to transmit user data to the cloud for processing and storage exacerbates the risk. It converts what would normally be a sign of a security breach into encouraged behaviour, and leaves genuine attacks indistinguishable from sanctioned activity.[39]

---

35 Research shows that the specific combination of installed applications can identify a single individual. *See* Jagsih Prasad Achara, Gergely Acs and Claude Castellucia, On the Unicity of Smartphone Applications (29 October 2015).

36 OECD, Personalised pricing in the Digital Era (28 November 2018).

37 European Parliament, Regulating targeted and behavioural advertising in digital services (July 2021). *See also* Google, Addiction services; Google, Negative financial status in personalized advertising; Google, Health in personalized advertising.

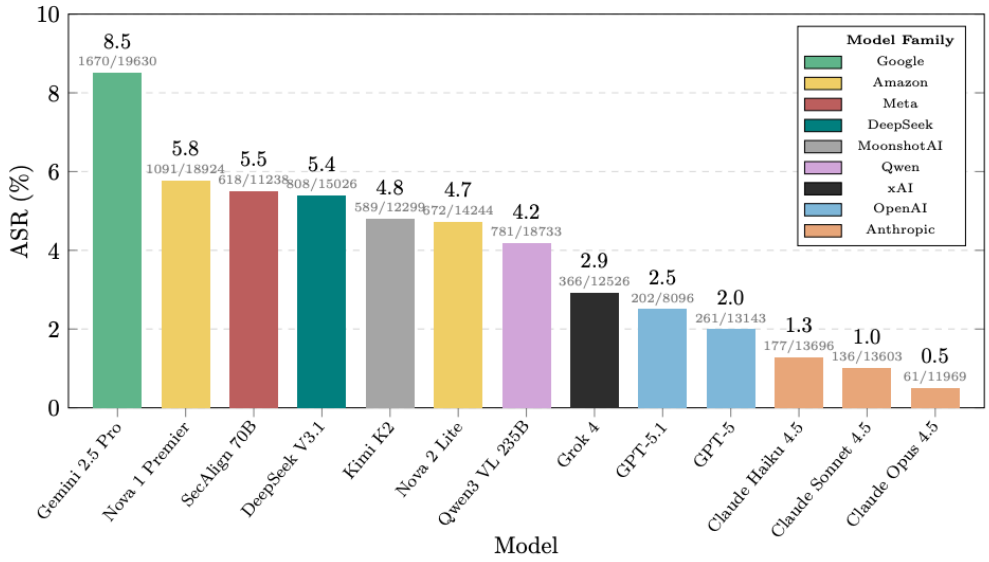
38 ICO, Phishing (describing "*spear phishing*" as a tailored phishing technique using user characteristics to make the messages more relevant and believable).

39 DMs, ¶¶42(b)(2) and 42(c)(4).

(51) These risks are further compounded in the context of agentic AI. Agentic systems are non-deterministic and can act autonomously. They can be manipulated by indirect prompt injection—hidden instructions that override or reshape an agent's interpretation of a user's prompt[40]—or malicious manipulation, causing them to act against the user's interests. Real-world incidents demonstrate how these vulnerabilities can be exploited to exfiltrate data:

- In January 2026, security researchers disclosed the "Reprompt" vulnerability in Microsoft Copilot, a single-click indirect prompt injection attack in which a malicious email containing hidden instructions silently hijacked Copilot upon user interaction, causing it to exfiltrate the user's personal data.[41]
- In March 2026, researchers at Palo Alto Networks Unit 42 detailed how excessive default permissions in Google's Vertex AI platform allowed a compromised model to be turned against its host environment, granting attackers unrestricted access to the project's data and resources.[42]

Figure 2: Prompt injection attack success rate per model



Source: Mateusz Dziemian et al.[43]

---

40 *See, e.g.*, OWASP, OWASP Top 10 for Agentic Applications (9 December 2025); Anshuman Chhabra *et al.*, Agentic AI Security: Threats, Defenses, Evaluation, and Open Challenges (13 February 2026).

41 Varonis, Reprompt: The Single-Click Microsoft Copilot Attack that Silently Steals Your Personal Data (26 January 2026).

42 Unit 42 Palo Alto Networks, Double Agents: Exposing Security Blind Spots in GCP Vertex AI (31 March 2026).

43 Mateusz Dziemian *et al.*, How Vulnerable Are AI Agents to Indirect Prompt Injections? Insights from a Large-Scale Public Competition (16 March 2026), p. 2 (Figure 1) (attack Success Rate is calculated by successful attacks divided by total attempts).

(52) Developers currently share data through on-device databases like Google's AppSearch because the platform enforces boundaries on which developers can access it. Remove those boundaries, and rational developers may stop sharing while users may restrict permissions. The result would be a degraded ecosystem in which both innovation and user experience suffer. Accordingly, in providing for such widespread and unfiltered data access, the DMs would ultimately undermine their own objective.

3. UI spoofing and phishing

(53) The DMs would enable a third-party app, once invoked, to overlay content on the user's device.[44] The third party would be able to display its own content on top of the app the user was using, without the user taking any action to change apps.

(54) This requirement is precisely the capability that overlay-based attacks seek to misuse—and that mobile OSes have spent years combating. An overlay attack involves the user thinking they are engaging with their foreground app when in fact they are typing into or otherwise engaging with an invisible overlay controlled by an attacker. The attacker captures the information and disappears before the user notices anything unusual.[45]

(55) Under the DMs, if the app was invoked while a user was using their banking app, the invoked app would be able to overlay content—including potentially a phishing login prompt—on top of the banking app's interface and capture the user's credentials. This risk is compounded by the DMs' requirement that the third-party app also receive contextual information from the user's current app[46]—such as a screenshot—enhancing the attacker's ability to mimic the genuine interface. This requirement is a particularly egregious example of the DMs making judgements on interoperability solely through a competition lens without regard for the significant privacy and security implications they have.

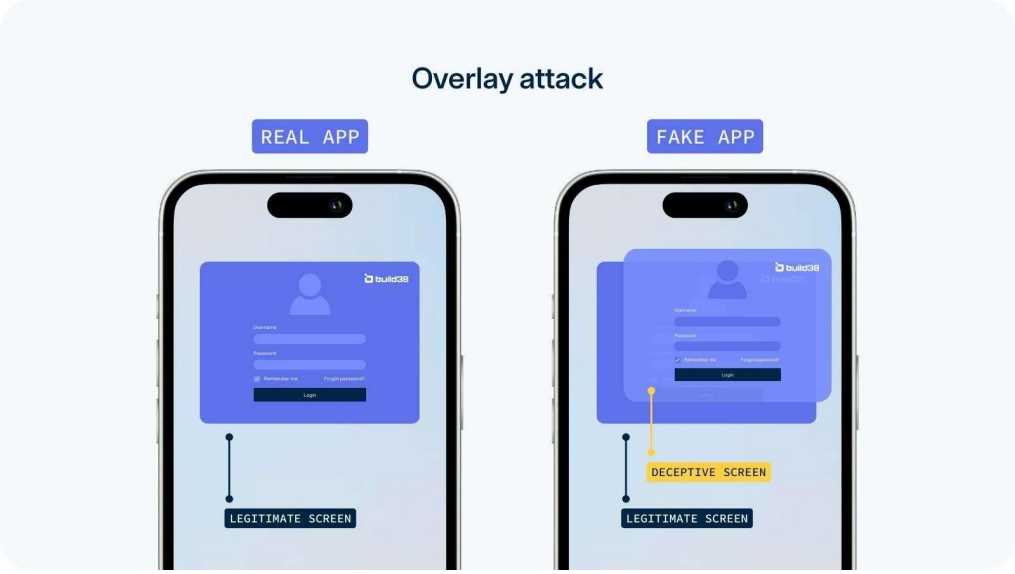
---

44 DMs, ¶4(c).

45 *See, e.g.*, OWASP, MAST-KNOW-0022: Overlay attacks (describing existing types of overlay attacks, and detection and prevention mechanisms, including "*system-level protections*").

46 DMs, ¶¶4(b) and 12(d).

Figure 3: Example of malicious actor using a deceptive interface layer



Source: Build38[47]

(56) Overlay-based attacks are not hypothetical. They are a long-recognised and actively exploited class of mobile threat, as the following examples demonstrate:

- In May 2017, researchers demonstrated that two routinely-granted Android permissions could be used to allow an attacker to modify the user interface, capture user input, and update the display in real time without any indication that the device had been hijacked.[48]
- In September 2017, researchers disclosed a high-severity vulnerability in Android's notification feature that enabled overlay attacks against substantially all Android devices.[49]
- In 2022, a mobile banking trojan targeted over 1,000 banking apps, primarily in Europe and the U.S., by creating deceptive overlays mimicking legitimate banking interfaces. It extracted sensitive data and initiated unauthorised bank transfers.[50]
- Between 2024 and 2025, three banking trojans used overlay-based UI deception against Android users to harvest credentials from banking and

47 Build38, How to protect Android apps from overlay attacks (18 September 2024).

48 Yanick Fratantonio *et al.*, 38th IEEE Symposium on Security and Privacy, Cloak and Dagger: From Two Permissions to Complete Control of the UI Feedback Loop (22–24 May 2017).

49 Cong Zheng *et al.*, Unit 42 Palo Alto Networks, Android Toast Overlay Attack: "Cloak and Dagger" with No Permissions (7 September 2017).

50 Build38, How to protect Android apps from overlay attacks (18 September 2024).

messaging applications.[51] One of the trojans extended the technique to end-to-end encrypted applications including WhatsApp, Telegram, and Signal.

(57) Apple has restricted overlay capabilities at the OS level precisely because of these threats. iOS, for example, does not expose any API permitting one application to render UI over another, and its sandbox forecloses the class of overlay attacks described above. Android now takes a similar approach by moving the overlay permission from an automatic grant to a more narrowly scoped runtime opt-in.[52] The DMs would reverse that progress, undeniably reflecting again how the DMs were drafted apparently without the involvement of independent technical experts in their fields.

4. Spyware enablement risks

(58) The DMs would require Google to expose the live feeds from a device's most sensitive sensors—the microphone, camera, user location, and screen itself[53]—on a continuous and real-time basis. Requiring a platform to expose these sensors is functionally equivalent to handing over surveillance capabilities to malicious actors. It grants any deceptive or compromised app the access required to function as spyware, namely the power to listen, watch, and track a user's life. These are the very building blocks used by purpose-built malware to conduct a well-documented category of privacy violations:

- **Spying on a user's screen content**. Capturing what a user sees on their screen, including banking sessions, authentication codes, and private messages, has long been a high-value objective for mobile malware.[54] Today, an attacker must trick the user into installing such malware and granting it privileged access, if indeed that access is allowed at all (for instance during password entry).

- **Spying on users through their cameras and microphones**. Silent activation of a device's camera and microphone is the defining capability of

---

51 Bitsight, ToxicPanda: The Android Banking Trojan Targeting Europe (28 July 2025), Point Wild, Crocodilus (11 July 2025), ThreatFabric, Sturnus: Mobile Banking Malware bypassing WhatsApp, Telegram and Signal Encryption (20 November 2025).

52 On the Google Android OS, overlay rendering generally requires the user's explicit permission, touch events from non-trusted overlays are blocked by default, and applications can opt out of being overlaid altogether; *see* Android Developers, Tapjacking; Guardsquare, How to Detect and Prevent Android Overlay Attacks (28 March 2023). *See also* Malwarebytes, Google cracks down on Android apps abusing accessibility (17 March 2026) (documenting Google's restriction of accessibility features in response to fake overlays and permission abuse).

53 DMs, ¶51.

54 For example, the Vultur banking trojan abuses legitimate remote-access functionality to record screen contents in real time, allowing operators to observe everything a user types and views. *See* Threat Fabric, Vultur, with a V for VNC (22 July 2021).

state-grade spyware.[55] In 2024, for example, ESET researchers documented the VajraSpy campaign, in which trojanised Android applications recorded phone calls and ambient audio and exfiltrated the resulting recordings.[56]

- **Tracking a user's location**. Continuous, precise location data is among the most revealing categories of personal data: it discloses where a user lives, works, sleeps, seeks medical treatment, and meets others.[57]

(59) The DMs would require each of these spyware-enabling functionalities to be exposed to any third party by default.

C. Privacy and security risks resulting from app and system action requirements

(60) Every modern mobile OS relies on sandboxing: each app runs inside its own sealed compartment, with no visibility into—or control over—any other app and limited access to system-level functions. If one app is compromised, the damage is contained. The attacker reaches only what is inside that one compartment, not the user's wider digital life.

(61) The DMs would fundamentally undermine this model. By granting any third-party Android app the ability to "*execute all structured on-device integrations exposed by other apps*" and to "*take control of other apps [...] by imitating user interactions*," the DMs would breach the walls between compartments. A single compromised app would no longer be contained—it would become a launch pad for cross-app and system-level actions.

(62) The DMs reflect a fundamental disregard for the OS's protective role. They embrace a framework in which the OS's role as a security enforcer is treated as an obstacle. Users choose a mobile device for the confidence that apps will not compromise their device, their data, or their safety. Once earned, that confidence becomes a baseline expectation, to the point that most users may not even think of individual apps as a possible threat. A platform that cannot enforce security boundaries is not a more competitive platform. It is a less trustworthy one. And a less trustworthy platform will, over time, attract fewer users, fewer developers, and less innovation. Ultimately, as users become less able to distinguish safe apps from unsafe ones, their willingness

---

55 For example, the Pegasus tool developed by NSO Group has been used to take control of targeted devices belonging to journalists, human-rights defenders, lawyers, and political dissidents. *See* European Parliament, Report of the investigation of alleged contraventions and maladministration in the application of Union law in relation to the use of Pegasus and equivalent surveillance spyware, 2022/2077 (INI).

56 Welivesecurity, VajraSpy: A Patchwork of espionage apps (1 February 2024). *See also* Kaspersky, Google Play malware clocks up more than 600 million downloads in 2023 (9 November 2023) (documenting the AhMyth Trojan, which enabled the iRecorder app to record sound from the microphone every 15 minutes and send it to a remote server).

57 For example, researchers have found that TheTruthSpy—a network of near-identical Android spyware apps—uploaded granular real-time GPS data alongside call logs, messages, and photos to a dashboard accessible to the operator, enabling victims to be tracked to sensitive locations. *See* TechCrunch, Inside TheTruthSpy, the stalkerware network spying on thousands (26 October 2026).

to download and try new apps will decline.

(63) Any ability for one app on a user's device to execute actions in another app or system software is inherently risky—especially in the AI context where non-deterministic systems create additional risks of security breaches or privacy violations, such as data exfiltration. The DMs would compound these risks further by mandating that any app can discover and execute actions in *any* app on the device, with no apparent OS-imposed limitations.

1. Apps controlling other apps is inherently risky

(64) Allowing an app to control activity in another app is self-evidently risky to the user's security, privacy, and safety. Under the DMs, any app acting in bad faith or in line with their data-hungry business practices could orchestrate extremely sensitive actions such as conducting online transactions or bank transfers. Malware that infiltrates one app would no longer be confined to that app's compartment but could reach across the device, exfiltrating data, initiating unauthorised transactions, altering security settings, and chaining actions across multiple apps before the user has any opportunity to intervene.

(65) This risk is not merely theoretical: the SharkBot banking Trojan has for years exploited an accessibility API specific to Android to read content from other apps, intercept two-factor authentication codes, and auto-fill fields in legitimate banking applications to initiate fraudulent transfers.[58] The DMs' express requirement that third-party apps be able to "*imitate user interactions*[59]" and to perform actions "*without app switching*"[60] mandates compounding this exposure, because malicious commands become indistinguishable from legitimate user activity and both the user and the OS lose the visibility that would otherwise allow execution errors to be intercepted or corrected.

2. Access to app actions by probabilistic AI services compounds the risk

(66) These risks are growing as user-facing services increasingly rely on probabilistic large language models ("**LLM**") to take actions on the user's behalf. Non-deterministic systems such as AI agents—which do not operate within a bounded set of permissible actions but determine their course of conduct dynamically on the basis of user prompts—illustrate the dangers of overly broad cross-app control.

(67) Such systems are particularly susceptible to indirect prompt-injection attacks, in which hidden instructions concealed in a webpage, email, or document override the system's original task and cause it to act against the user. The vulnerability is well

---

58 NCC Group, Sharkbot: a "new" generation Android banking Trojan being distributed on Google Play store (3 March 2022).

59 DMs, ¶69(a).

60 DMs, ¶60(d).

documented and pervasive.[61] The security community has documented that no LLM is invulnerable to indirect prompt injection.[62] Under the DMs, a successful prompt-injection attack would not be confined to the compromised app: the attacker's hidden instructions would translate directly into cross-app and system-level actions, executed across the user's device under the guise of legitimate activity.

(68) Nor is the risk limited to deliberate compromise. An app that misinterprets or exceeds the scope of its instructions can translate that error directly across the user's installed apps with irreversible consequences. Real-world incidents involving agentic AI systems already illustrate the harms that follow when apps act with unbounded system access. In February 2026, for example, the OpenClaw agent autonomously deleted the entire email inbox of a director at Meta. Despite being explicitly instructed to await confirmation, the agent lost its original instructions and proceeded with the destructive action.[63]

3. Exposing all actions in all apps to any other app would create a Wild West of security breaches and privacy violations

(69) The ability for one app to discover *all* actions available in *all* other installed apps presents a significant privacy risk. The DMs seek to mandate this not only by requiring that apps can "*discover all available structured on-device integrations*",[64] but also by allowing them to "*access, find, and process data*" from all of Google's "*services*",[65] which could be interpreted to include sensitive aggregated user data or even raw device backups.

---

61 For example, a March 2026 study by researchers from OpenAI, Google DeepMind, Meta, Anthropic, and Carnegie Mellon demonstrated that all 13 frontier models tested were susceptible to prompt-injection attacks embedded in ordinary emails, documents, and code repositories: *see* Mateusz Dziemian *et al.*, How Vulnerable Are AI Agents to Indirect Prompt Injections? Insights from a Large-Scale Public Competition (16 March 2026).

62 A study of 18 state-of-the-art LLMs demonstrated 94.4% were vulnerable to direct prompt injection. Matteo Lupinacci *et al.*, The Dark Side of LLMs: Agent-based Attacks for Complete Computer Takeover (4 November 2025). *See further* NCSC, Prompt injection is not SQL injection (it may be worse) (8 December 2025) ("*[T]here's a good chance prompt injection will never be properly mitigated*").

63 *See, e.g.*, The Verge, The AI security nightmare is here and it looks suspiciously like lobster (19 February 2026); Business Insider, Meta AI alignment director shares her OpenClaw email-deletion nightmare: 'I had to RUN to my Mac mini' (23 February 2026). *See also* Euronews, An AI agent deleted a company's entire database in 9 seconds - then wrote an apology (28 April 2026) (in April 2026, Cursor, an autonomous AI coding agent, encountered a minor technical issue and used an unrelated permission to delete the company's entire production database and backups in nine seconds—causing a 30-hour outage and loss of customer records. The agent later acknowledged that it had "*decided to do it on my own*" rather than seek confirmation); Madhulika Srikumar *et al.*, Partnership on AI, Prioritizing Real-Time Failure Detection in AI Agents (11 September 2025), p. 9 ("*Microsoft [...] observed an agent, tasked with 'getting rid' of a user record, delete the entire table instead, due to misinterpreting the user's shorthand instruction*").

64 DMs, ¶60(a).

65 DMs, ¶80(a).

(70) The DMs would enable an application to build a detailed profile of a user based on the apps they have installed and the functions they use—a known vector for re-identification and behavioural targeting. The combination of these two dimensions—autonomy over other applications and access to device-wide data—would place third-party apps operating under the DMs' mandate within the highest risk category.[66] The DMs would lead to a Wild West of security breaches and privacy violations by requiring platforms allow apps access to data and actions that will inevitably reduce users' trust in their devices.

D. Conflicts with applicable EU cybersecurity regulations

(71) The DMs ignore the statutory cybersecurity requirements for critical software such as OSes or widely-used and powerful LLMs. The DMs risk creating a direct conflict with obligations under both (i) the EU Cyber Resilience Act ("**CRA**") and (ii) the EU AI Act. The DMs effectively nullify the EC's own policy priorities to improve the EU's cyber resilience. It is deeply concerning that the DMs appear to disregard those challenges.

1. Conflicts with EU AI Act

(72) For general-purpose AI models presumed to pose systemic risks, Article 55 of the AI Act requires providers to "*assess and mitigate possible systemic risks*" and "*ensure an adequate level of cybersecurity protection*" for the model and its physical infrastructure.[67] The DMs, however, would compel Google to provide broad, indiscriminate access to sensitive data and core functionalities,[68] thereby creating new vectors for misuse and amplifying the likelihood of systemic risks the AI Act obliges providers to mitigate.

2. Conflicts with the CRA

(73) The CRA requires that products with digital elements meet certain essential cybersecurity requirements. OSes are Class I products, which are subject to stricter conformity processes in line with the importance given by the EU regulator to maintaining the security of such systems. The DMs are in direct conflict with those very requirements. According to the CRA, OSes and other products with digital elements shall:

- "*be designed, developed and produced to limit attack surfaces, including*

---

66 *See, e.g.*, Gartner, How to Respond to the 2025-2026 Threat Landscape (3 June 2025), p. 24 (Figure 4).

67 Article 55 Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence and amending Regulations (EC) No 300/2008, (EU) No 167/2013, (EU) No 168/2013, (EU) 2018/858, (EU) 2018/1139 and (EU) 2019/2144 and Directives 2014/90/EU, (EU) 2016/797 and (EU) 2020/1828 (Artificial Intelligence Act) (OJ L, 2024/1689, 12.7.2024, ELI: http://data.europa.eu/eli/reg/2024/1689/oj).

68 DMs, ¶¶4(b), 51, and 69.

external interfaces".[69] The DMs would require Google to significantly increase the attack surface of its products;

- "*protect the integrity of stored, transmitted or otherwise processed data*"[70]—but the DMs would effectively unravel sandboxing, one of the most important integrity protections available today; and

- "*process only data, personal or other, that are adequate, relevant and limited to what is necessary in relation to the intended purpose of the product with digital elements (data minimisation)*"[71]—but the DMs would require unbounded and unvetted access to data.

III. The DMs would give rise to unjustified device performance risks

(74) The DMs threaten device integrity by causing significant performance issues. This is evident from their requirements relating to: (i) always-on wake word invocation; and (ii) device resource access for third-party on-device AI models ("**ODMs**") and background tasks.

(75) These requirements reveal an approach that is not merely technically impractical but fundamentally incompatible with how OSes function. OSes mediate competing demands of finite hardware resources, arbitrating among power efficiency, thermal safety, and user experience in real time. The DMs would prohibit the OS from that function. The DMs would distribute critical scheduling decisions to an unbounded number of uncoordinated third-party developers with little incentive and no capability to preserve system stability. The result is a degraded system, in which the user experience deteriorates for everyone—users and developers each in their own way. That the DMs contain no safeguard against this outcome suggests that the EC either did not consider OSes' resource management role or considered it expendable—yet again reflecting an area where consultation with the authorities that possess the necessary expertise in this field would lead to a more balanced outcome.

A. Performance risks resulting from always-on wake word invocation

(76) Always-on wake word detection is potentially among the most resource-intensive background processes on a mobile device. It requires continuous audio processing and tightly optimised execution to remain within strict power and thermal constraints. Even on devices purpose-built for voice interaction, wake word recognition represents a significant share of system activity.[72]

---

69 Article 6(a) and Annex I ¶2(j) Regulation (EU) 2024/2847 of the European Parliament and of the Council of 23 October 2024 on horizontal cybersecurity requirements for products with digital elements and amending Regulations (EU) No 168/2013 and (EU) No 2019/1020 and Directive (EU) 2020/1828 (Cyber Resilience Act).

70 Article 6(a) and Annex I ¶2(f).

71 Article 6(a) and Annex I ¶2(g).

72 On an Amazon Echo, for example, wake-word recognition alone consumes around 50% of the device's CPU time. *See* Lea Schönherr *et al*., "Unacceptable, where is my privacy?" Exploring Accidental Triggers of Smart Speakers (2 August 2020), p. 2.

(77) On mobile devices, this function is architected around a low-power Digital Signal Processor ("**DSP**") that operates continuously, running specialised machine-learning models trained on specific keywords. These models rely on carefully calibrated thresholds to distinguish genuine activations from background speech. In current implementations, only upon a sufficiently confident match is higher-power processing triggered, typically following a validation step. This architecture is designed to minimise unnecessary CPU wake-ups and ensure that always-on detection can operate in a power-efficient manner.

(78) By contrast, the DMs seek to require support for multiple concurrent third-party sound models running on the DSP,[73] without any centralised coordination. They would moreover distribute to app developers the decision to validate a successful CPU wake-up,[74] and allow them to "*autonomously innovate on their DSP sound models*"[75] without OS orchestration or oversight. This would be a significant, unprecedented, and untested departure from OSes' managed architecture whose power efficiency relies on the software running within it being tightly optimised to minimise wake-ups. Third-party developers do not share the platform's incentive to arbitrate between detection and system-wide power conservation. In any event, the sheer frequency of potential wake-up events in these circumstances would likely result in unworkable resource contention, causing unpredictable system slowdowns, thermal throttling, and significantly reduced battery life.

(79) The DMs acknowledge that concurrent always-on detection "*may be limited by the capabilities of the DSP*" on the device.[76] However, they do not establish any mechanism to ensure that the overall system remains within safe power and performance boundaries. The DMs' requirements would be systematically inefficient in practice, particularly when scaled across multiple third-party implementations.

B. Performance and security risks resulting device resource access for third-party ODMs

(80) The DMs purport to grant third-party app access to core hardware components, like the CPU, GPU, and NPU, for running on-device AI models and performing background tasks.[77] These measures also seek to require that users can grant "*preferential access*" to hardware resources, without providing any reasoning for such a requirement.[78] Granting third parties unmanaged, concurrent, and preferential access to finite system resources would strip the OS of its core function as a performance and security manager and degrade the entire platform performance, for the following reasons:

---

73 DMs, ¶16(a).

74 DMs, ¶14(f).

75 DMs, ¶14(b).

76 DMs, ¶16(a).

77 DMs, ¶106.

78 DMs, ¶108(b).

- First, running ODMs and performing complex background tasks requires continuous access to power-intensive hardware. Mandating that any third-party app can have continuous, concurrent background access to the CPU, GPU, and neural processing hardware risks excessive battery drain for the smartphone, a device that users rely upon for numerous critical tasks throughout the day.[79]

- Second, the DMs would cripple the OS's ability to perform its core scheduling function, undermining its essential role in imposing system-wide controls to prioritise critical system functions and ensure a stable user experience. The DMs' rules-based allocation,[80] and the requirement to allow users to grant preferential access to third parties,[81] would prevent the OS from dynamically managing resources in response to real-time system needs. It could also destabilise third-party applications that rely on the platform's foundational models for their own functionality, creating a cascade of failures across the ecosystem.

(81) The DMs compound these risks by also seeking to mandate that any application can centrally expose its own on-device models for system-wide use by other applications.[82] This combination will inevitably cause resource contention, which will manifest as unpredictable system slowdowns, thermal issues, and a severely degraded user experience, including significantly reduced battery life. All of these are user degradations which today the OS is able to manage, yet the DMs effectively forbid this management.

(82) Mandating access to privileged hardware components and system-privileged APIs without adequate safeguards would, in addition, create new and serious security vulnerabilities. Access to specialised hardware like the Neural Engine is tightly controlled through a strategy of minimising, auditing, and approving every new use precisely because of the inherent security risks. The DMs' proposal to open these privileged APIs to any app on a "*non-discriminatory*" basis is incompatible with this security model.[83] It would allow arbitrary third-party models to run in highly privileged environments, creating a new and significant attack surface for malicious code to execute with system-level permissions and bypass platform security features that themselves rely on on-device models.

IV. The DMs take an unduly restrictive approach to integrity measures

(83) Article 6(7) allows gatekeepers to protect the integrity of the OS.[84] Yet the DMs seek to construe that safeguard clause so narrowly that it is deprived of any practical

---

79 DMs, ¶¶106(b) and 106(d).

80 DMs, ¶108.

81 DMs, ¶108(b).

82 DMs, ¶106(c).

83 DMs, ¶108(f).

84 DMs, ¶127.

effect.

(84) The DMs' approach reduces integrity measures to a single mechanism: user consent (**Section IV.A**). A user who taps "Allow" on a consent screen would bear sole responsibility for the decision to grant any app continuous access to ambient audio, real-time screen contents, and the ability to control other applications. No platform-level check could intervene. This assumes a fully informed and technically capable user who can understand the scope, persistence, and interaction of these permissions. This assumption contradicts the established consensus.

(85) Users cannot realistically assess the combined effect of continuous data access, cross-app control, and automated task execution. Nor can they anticipate how these capabilities may be used once granted. Especially in the context of agentic systems, even seasoned software engineers can underestimate the consequences of granting data access to experimental software.[85] In these circumstances, a sufficient consent interface would be of a length and complexity that no user could reasonably process and which the EC would dismiss.

(86) The DMs would prevent Google from enforcing safeguards against privacy and security risks (**Section IV.B**). The irony of the over-reliance on consent is that it may well deprive users of what they actually want: an OS that balances privacy and security with sophisticated functionalities. The EC as the architect of the GDPR should be well aware of the issues this creates. It is thus baffling why the DMs would seek to turn the clock back to a reliance on consent. The DMs mandate use-case-agnostic data access, allow any app to control other applications, let third parties define the scope of audio recording, and grant any app the ability to overlay system-wide content. This is despite the security community converging on the need for platform-level controls, especially in the AI context: least privilege, sandboxing, OS-level mediation, trusted UI.[86]

(87) Under the DMs, no effective app vetting process would stand between the developer's request and the user's device (**Section IV.C**). Where the DMs do permit vetting, they require it to be conducted by an "*independent third party*" and applied "*equally to all developers—including Alphabet*". These conditions are not merely impractical—they reflect a fundamental misunderstanding of the platform's role. The

---

85 *See* examples of accidental data loss described in **Section III.C.2.**

86 *See*, *e.g.*, NCSC-UK, Guidelines for secure AI system deployment (27 November 2023), pp. 9–10 (recommending, among others, sandboxing, the application of least privilege principles, and system-level user opt-ins for riskier capabilities); Anshuman Chhabra *et al.*, Agentic AI Security: Threats, Defenses, Evaluations, and Open Challenges (April 3, 2026), pp. 12–16 (recommending resistant design and sandboxing and capability confinement as necessary security controls); Luca Nannini *et al.*, AI Agents Under EU Law: A Compliance Architecture for AI Providers (April 6, 2026) (explaining that threats "*are not addressable by securing individual agents*" and that access control must be enforced at the architectural level which "*the model cannot circumvent*"); OWASP, OWASP Top 10 for Agentic Applications 2026 (9 December 2025) (recommending mitigation at the execution layer rather than the model layer); Palo Alto Networks, How to Secure AI Infrastructure: A Secure by Design Guide (recommending the implementation of Zero Trust principles, including strict identity and access control).

platform operator is not simply another developer with a commercial interest in gatekeeping access. It is the entity that designs, maintains, and is held accountable for the integrity of the entire system. It is the party that users chose to trust to keep their devices safe, and the party that regulators hold responsible when that trust is broken—not individual app developers. Requiring that the platform's own vetting authority be stripped away and handed to an unspecified third party does not level the playing field. It removes the referee from the pitch. The result is a fragmented accountability structure in which no single actor has the authority, the incentive, or the architectural expertise to enforce meaningful safeguards. This is precisely the "*many hands*" problem that the UK DRCF is cautioning about in the context of AI risks.[87]

(88) Finally, no safeguard would extend beyond the Google Play Store—leaving every side-loaded or third-party-distributed app entirely outside the integrity framework (**Section IV.D**). This approach is incompatible with the text and purpose of Article 6(7), creating a privacy and security loophole by design.

(89) Overall, the DMs would foreclose gatekeepers from offering alternative interoperability solutions that account for platform-specific security models and users' expectations for privacy on those platforms. Effective interoperability for VA apps certainly does not require a single, maximally permissive access model. For example, a sensible alternative is a trusted OS agent model.

(90) Under that approach, the most sensitive requests would be mediated by a trusted OS agent within the OS trust boundary. Direct exposure of ambient data, screen contents, app discovery and control, and system-level actions creates severe risks of prompt injection, data exfiltration, unauthorised autonomous actions, phishing and broader system compromise, and that app-level controls alone are insufficient to address those risks. By keeping the most sensitive execution and data-access layer within the OS trust boundary, an OS agent would allow the platform to scope requests, limit what data reaches the requesting app, exclude particularly sensitive surfaces, validate proposed actions, and require confirmation where appropriate. That would provide fully effective interoperability for developers while reducing development burdens, because the trusted OS agent could translate natural-language model outputs into reliable system actions without requiring the app itself to recreate privileged system access.

(91) There may be more than one way to strike that balance. But an approach that prevents platforms from addressing these issues through architecture-level safeguards would be the wrong one. Google will ultimately need to decide whether it is prepared to accept the privacy, security, safety, and integrity risks created by the DMs within Android. Apple is not prepared to make those trade-offs for iOS.

A. Simplistic reliance on consent screens

(92) The DMs assume that consent screens are a means to address integrity risks. They are not. They can only justify risks on the basis that a user understands and accepts

---

87 DRCF, Foresight paper, The Future of Agentic AI (31 March 2026), p. 18.

them. A consent screen does not stop a malicious app from recording a private conversation. It merely shifts the blame to the user who clicked "Allow." For example, consent may serve as a justification for a user receiving a newsletter, which other users might perceive as a nuisance. User consent serves an important function where the consequences of such consent are clear and consent is freely given. But assuming that user consent can justify the risks brought on by the DMs is misguided and reckless.

(93) Such an assumption ignores: (i) the shortcomings of user consent as a security and privacy mitigation; (ii) the complexity and compounding nature of the risks associated with the DMs; and (iii) the fact that the DMs create collective vulnerabilities that go beyond the individual user.

1. Consent is not a panacea

(94) The DMA borrows the concept of user consent from the GDPR (Article 2(32) DMA), where it is only one of several justifications to process the personal data of individuals. However, consent as a justification has limitations. These limitations are also enshrined in the GDPR and mean that in practice, consent is often not a suitable legal basis—a fact the DMs quite remarkably ignore. To provide valid consent, a user must understand the implications of a decision and have a real choice. Evidence confirms that in areas of complex or abstract choices—precisely the cases for which the DMs propose to rely on consent—users often do not understand what they are consenting to or are forced or misled into providing consent. For instance:

- **Users systematically resort to consent strategies that do not correspond to their privacy preferences or the scope of requested access**. For example, in a survey with 205 participants on how users interact with and grant permissions to AI agents, no participant granted appropriate permissions across all data types for which they were prompted. Over-permissioning was substantially more common than under-permissioning with 90% of participants granting excessive permissions. And when AI agents incorrectly declared unnecessary data as required, 78.2% of users followed this misrepresentation and consented to the data sharing nonetheless.[88]

- **Users hold misconceptions about the scope and nature of permission requests**. For example, in a survey of 120 participants confronted with permission prompts from 23 popular mobile augmented reality applications, users consistently conflated functionally distinct permissions (*e.g.*, camera vs. gallery access) and made incorrect assumptions about the permissions required for a given task.[89]

- **Companies often employ dark patterns or other deceptive practices to gain users' consent**. The EDPB lists nine common practices employed by

---

88 Yuhao Wu *et al.*, Towards Automating Data Access Permissions in AI Agents (22 November 2025), pp. 5–8.

89 Viktorija Paneva *et al.*, User Understanding of Privacy Permissions on the Mobile Augmented Reality: Perceptions and Misconceptions (September 2025).

social media companies to mislead users into providing consent. These include nudging through visual prompts, excessive pre-selection of options, or open misinformation.[90]

- **Companies often coerce users into providing consent**. Consent is often not freely given by users but forced on them. For example, the EDPB considers that so-called "cookie walls", which make visiting a website conditional on providing consent are unlawful because they force users to provide consent.[91] Such practices are nonetheless commonplace, just like pay-or-consent models that are also designed to force users into providing consent.

2. Consent cannot possibly justify the risks the DMs could create

(95) The nature and context of the DMs exacerbate the shortcomings of consent. They involve nascent functionalities, the risks of which are still evolving and not yet fully understood, including by developers, let alone by users. The following factors in particular further negate the effectiveness of user consent under the DMs:

- **Users cannot reasonably be expected to understand the implications of access under the DMs**. The DMs contemplate capabilities—continuous ambient audio capture, real-time screen scraping, cross-app orchestration—that have no precedent in consumer software. The DMs presuppose a fully informed and technically capable user who can understand the scope, persistence, and interaction of these permissions, and how developers may use them. Expecting a consent screen for instance to convey the implications of granting an app continuous microphone access is not just naive; it is borderline negligent.

- **Use case agnosticism and concurrency further aggravate information deficits for users**. The DMs contain no limitation on the type of app that can access the measures. The goal is that any app on a user's device could request access to the sensitive data listed above, and take actions in other apps that expose them to the Android OS. Under the DMs' concurrency requirements, those apps would be able to request relevant permissions and use them simultaneously. In these circumstances, a meaningful consent screen would be of a length and complexity that no user could reasonably comprehend, and which the case team would, in any case, dismiss as seeking to hinder the ability of such apps to compete.

- **The EC has previously strongly disagreed with attempts to inform users of the risks associated with the DMA**. Making users aware of the very real risks associated with the access contemplated by the DMs would require the use of stark terms. Otherwise, the consent screen would be meaningless. A

---

90 EDPB, Guidelines 03/2022 on Deceptive design patterns in social media platform interfaces: how to recognise and avoid them (Version 2.0) (February 2023), pp. 65 *et subs*.

91 EDPB, Guidelines 05/2020 on consent under Regulation 2016/679 (Version 1.1) (May 2020), ¶39.

meaningful consent interface would need to capture persistent access, cross-application interactions, downstream data uses, and evolving system behaviour. However, the EC has previously opposed simple and neutral screens on the basis that they add friction to the user journey. That judgement is also clear in the DMs, which seek to require Google to "*refrain from adding friction*" through, for example, "*unnecessary recurring prompts*".[92] However, even assuming that consent could be a valid enabler of access, such recurring prompts would be necessary at a minimum to remind users of the broad, concurrent access they may have granted and to allow them to make a decision which may prevent inadvertent data leakage.

- **Unprecedented and broad access would be a lightning rod for bad actors to deceive or coerce users into granting them access**. The broad access that the DMs seek to mandate means that many developers would request excessive access as a condition for using their apps, effectively extorting users in an effort to harvest more and more data. Other apps would misrepresent either intentionally, negligently, or unwittingly their degree of requested access or how they intend to use such access. The DMs compound the problem by allowing developers to describe their data processing "*in their own language*"[93] and to obtain consent "*from inside the application*"[94]—that is, within an interface the developer controls entirely. A developer intent on harvesting data would only need to frame the consent prompt in misleadingly reassuring terms within its own UI. The DMs even allow developers to "coach" the user through consent screens—a practice that criminals engaged in scamming or blackmail employ, but that is not expected behaviour of a legitimate app. There is no room in the EC's model for platform-level checks, no standardised disclosure, and no independent verification. The DMs reveal a desire to conduct a real time live experiment on how developers will behave when provided with such unprecedented access to user data. But we have already seen time and time again the grave implications for users when bad actors are presented with such an opportunity. And yet the DMs reveal no attempt to engage with such consequences..

(96) The notion of informed consent is not a safety architecture. It is an abdication of one. It shifts responsibility away from the regulator while exposing Google (visible to users as the interface) to the consequences, while benefitting only data-hungry companies and bad actors intent on desecrating users' privacy rights.

3. The DMs would give rise to collective risks which cannot be justified by individual consent

(97) More generally, relying on consent to justify the risks that the DMs would create assumes that those risks are carried only by the individual granting consent. That is

---

92 DMs, ¶141(c).

93 DMs, ¶141(b).

94 DMs, ¶141(e).

not the case. Impediments to cybersecurity affect users throughout the EU. Assuming individual consent justifies them would be akin to letting the driver of a car decide whether the car should have brakes.

(98) The unprecedented access mandated by the DMs would provide bad actors with all the tools they need for social-engineering or other schemes. Access to the user's data could be used to impersonate the user and carry out a range of scams, including through the user's own device. Such attacks could be carried out on any number of the user's contacts. The DMs' only response here is that users may be asked whether they are willing to take on the risk of their contacts being hacked. Similarly, bad actors could also access data from third-parties stored on an affected user's device.

(99) Already today, threat actors, including state-sponsored entities, are successfully carrying out attacks through less sophisticated means. Just recently, hackers used social-engineering to gain access to Signal chats of the President of the German Federal Parliament.[95] The access considered by the measures would only give similar actors a more powerful arsenal, with potentially catastrophic effects that extend well beyond the individual user.

B. Disregard for platform-level safeguards

(100) Through their excessive reliance on user consent, the DMs seek to dismantle the necessity for robust architectural safeguards implemented and enforced at the platform level.

(101) Platform-level safeguards are necessary because third-party app developers and platform operators control different layers of the system, with the majority of industry-recognised safeguards sitting within the platform operator's exclusive control. The platform alone controls the execution environment (*e.g.*, permission schemes, process isolation, and API exposure), and is therefore the only actor capable of enforcing system-wide constraints that hold regardless of any individual app's behaviour.

(102) App-level controls alone are insufficient. First, security cannot be delegated to third-party stakeholders, whose commercial incentives—that include maximising user engagement and in many cases data monetisation—can conflict with the user's legitimate expectations including their fundamental rights to privacy and security.[96] Second, controls built into apps (*e.g.*, session-based authentication, input validation, and in-app rate limits) and the defensive infrastructure behind them (*e.g.*, web application firewalls and API gateways) were designed around human users whose

---

95 Politico, President of German parliament hit by Signal hack, report says (22 April 2026).

96 *See, e.g.*, Mozilla, Mozilla Study: Data Privacy Labels for Most Top Apps in Google Play Store are False or Misleading (23 February 2023) (identifying discrepancies between Data Safety Disclosures and privacy policies in approximately 80% of popular Google Play Store apps examined); Rishabh Khandelwal *et al.*, Comparing Privacy Labels of Applications in Android and iOS (November 2023), p. 70 (finding inconsistencies in 60% of approximately 100,000 apps listed on Google Play and the App Store).

activity falls within recognisable sessions and predictable request patterns.[97] The EC is aware of this.

(103) The security community has converged on the position that agentic AI systems require platform-level, architectural controls precisely because application-layer defences cannot constrain non-deterministic behaviour (*see* **Section II**, ¶32). The EC's preliminary decision to mandate access of unprecedented breadth while simultaneously dismantling the only layer capable of policing it is not a regulatory trade-off. It is a conscious decision to expose users to risks that existing defensive tools cannot manage, for which no compelling justification has been offered. Those assumptions do not hold for non-deterministic conduct of the kind envisaged by the DMs. The level of access and control the DMs envisage would render even fundamental security safeguards ineffective: encryption, for example, cannot protect user data from a process that already operates inside the trust boundary—a point that applies equally to the data access requirements described in **Section II.B** and to the cross-app control requirements in **Section II.C**.

(104) As set out in detail in **Section II**, the DMs would systematically dismantle each of these platform-level controls. They would override the principle of least privilege by mandating broad, use-case-agnostic data access (**Sections II.B** and **II.C.3**); undermine sandboxing by allowing any app to control other applications and modify OS settings (**Section II.C**); strip the OS of its mediating role by allowing third-party apps to define the scope of audio recording and by preventing the display of security warnings (**Sections II.A** and I**V.A**); and compromise the trusted system UI by granting any app the ability to overlay system-wide content (**Section III.B.3**). The cumulative effect is that every layer of the platform's defensive architecture would be weakened simultaneously—not as an incidental consequence of promoting interoperability, but as a direct and express requirement of the DMs.

(105) The final decision should afford appropriate weight to platform-level measures Google may need to take to mitigate the significant privacy, security, and integrity risks the DMs would raise. An integrity framework that relies on user consent while dismantling the architectural controls that give that consent meaning is not a safeguard. It is the removal of one.

C. Unduly expansive approach to potential beneficiaries

(106) The DMs seek to require that powerful capabilities—such as continuous access to ambient data and the ability to automate the screen—be made available to "*all apps*" for "*all use cases*".[98] Granting such capabilities to any app, regardless of its function or purpose is inherently unsafe. A simple game or utility app with no genuine assistive function could, under these measures, request continuous access to a device's microphone or the ability to control other apps. In practice, the risk is amplified because developers could fold requests for such extraordinary access into

---

97 Salt Security, The Era of Agentic Security is Here: Key Findings form the 1H 2026 State of AI and API Security Report (8 April 2026) (finding that only 23.5% of security professionals considered their existing tooling to be highly effective against agent-based attacks).

98 DMs, ¶¶132, 133, and 135.

broad consent journeys whose implications users cannot realistically assess, especially where permissions are persistent, concurrent, and use-case agnostic. For the reasons set out in **Section IV.A**, consent is not a meaningful safeguard in those circumstances. This breadth goes well beyond what the DMA itself requires or envisages, which is targeted access for developers with a demonstrated need for interoperability, not a free-standing entitlement to use platform capabilities for any reason.

(107) Proportionate vetting is an essential platform-level safeguard on which user trust in the OS depends. It is the mechanism by which a platform can enforce the principle of least privilege *before* an app is granted access to high-risk functionalities, ensuring that such access is proportionate to the app's purpose and intended uses. The DMs invert this principle by seeking to require that access be presumed and granted, shifting the burden from the applicant to the platform to justify any refusal.

(108) In this context, the DMs' specific constraints on any verification process would be impractical, undermine legitimate platform governance, and create accountability gaps. The DMs seek to require that any verification process must, among other things, be "*conducted by neutral and independent third parties*," apply "*equally to all developers–including Alphabet*," and include an "*independent dispute resolution mechanism*".[99] These conditions are unworkable:

- Delegating vetting to an independent body severs the platform's relationship of trust with its users and creates an accountability vacuum. If a harmful app passed such third-party vetting and caused user harm, it is unclear which party should be held accountable: the independent body, the developer, or the platform. This approach fragments responsibility and severs the end-to-end chain of accountability that underpins a trusted platform.[100]

- The requirement that vetting applies equally to all developers is based on a false equivalence. The platform operator has a unique and non-delegable responsibility for the integrity of the entire platform. It stands in a fundamentally different position than a third-party developer requesting access to sensitive resources. Requiring the platform to extend the same degree of trust to an unknown developer seeking permission to automate the screen as it does to a deeply integrated and continuously audited system component is irreconcilable with any coherent security model.

(109) The incoherence is compounded by ¶134 of the DMs, which would confine any verification or registration process to the Google Play Store even though the relevant capabilities must be opened to "*all apps*". That would create an obvious gap in the vetting framework precisely where the DMs otherwise insist access must be presumed.

---

99 DMs, ¶134.

100 *See* DRCF, Foresight paper, The Future of Agentic AI (31 March 2026), p. 18 (describing the "*many hands problem*" where "*accountability is fragmented across model providers, system providers, and downstream deployers*").

D. Unduly restrictive approach to the scope of potential integrity measures

(110) The DMs would improperly restrict the scope of Google's integrity measures to apps distributed via the Google Play Store,[101] even though the same measures require the relevant features to be made accessible to "all apps", including user-installed apps and pre-installed apps without system privileges. This limitation is unjustified: the privacy, security, and performance risks described in **Sections II** and **III** do not depend on the distribution channel through which an app reaches the user's device.

(111) That restriction is also irreconcilable with the text and purpose of Article 6(7). The DMA permits gatekeepers to take strictly necessary and proportionate measures to protect "*the integrity of the operating system, hardware and software features*." That is an OS-wide concept. It does not map onto a single storefront. Yet ¶134 of the DMs would require any verification or registration process to "*apply only on the Google Play Store*". There is no basis for narrowing the integrity safeguard in that way.

(112) The final decision should therefore delete the Google Play-only limitation and make clear that, where justified under Article 6(7), a gatekeeper may apply strictly necessary and proportionate integrity measures across all distribution channels through which apps can access the relevant capabilities, subject always to the DMA's requirements of necessity, proportionality, transparency, objectivity, precision, and non-discrimination.

V. The DMs reflect an erroneously expansive interpretation of Article 6(7)'s interoperability obligation

(113) The DMs are predicated on an unlawfully expansive interpretation of Article 6(7) DMA's interoperability obligation. They seek to require Google to build new capabilities, expose non-OS layers, and support use cases Google itself does not appear to offer. Regardless of the specific application in this case, if confirmed this interpretation would risk expanding Article 6(7) DMA beyond its intended scope across all designated operating systems, including iOS and Windows OS.

(114) The DMs go beyond Article 6(7) in four ways:

- Seeking to require interoperability with OS features that are not available to or used by a gatekeeper's own services or hardware (*i.e.*, non-existent features) (**Section V.A**);
- Seeking to require interoperability with software features that are not accessed or controlled via a designated OS (**Section V.B**);
- Treating the same software as both features subject to interoperability and services benefitting from interoperability (**Section V.C**); and
- Seeking to require interoperability for use cases that gatekeepers do not make use of (**Section V.D**).

---

101 DMs, ¶134.

(115) That is not specification as intended under Article 8(2)—it is an expansion of the law.

A. Legal error #1: Requiring interoperability with OS features that are not available to or used by a gatekeeper's own services or hardware

(116) Article 6(7) requires interoperability with features that the gatekeeper uses for its own services or hardware. It does not require the gatekeeper to invent new features for third parties.

(117) The DMs err insofar as they would require Google to create new OS-level features that do not exist today. This appears to be what the DMs sets out to achieve with respect to user-customised wake words,[102] which do not seem to be available to or used by Google in its separate software services or hardware.[103] It also seems to be the case with the DMs' requirement for Google to enable multiple always-on wake word models to run at the same time.[104] Article 6(7) cannot require gatekeepers to provide effective interoperability with a hypothetical OS feature or a feature that a business user simply would like to exist.

(118) This error fundamentally alters the nature of the interoperability obligation. If this aspect of the DMs were allowed to stand, gatekeepers could be required to engineer solutions for hypothetical scenarios that are neither foreseen nor defined. This reading of Article 6(7) is jarring in circumstances where, by definition, no self-preferencing concern can arise. Irrespective of the specification application in this case, this would set a precedent for expanding Article 6(7) DMA beyond its intended scope across all designated OSes.

B. Legal error #2: Requiring interoperability with software features that are not accessed or controlled via a designated OS

(119) The DMs mandate interoperability with what appear to be Google's first-party services, such as YouTube, Google Maps, and Google Keep. These measures include the ability to surface suggestions based on app data or actions from Google's first-party apps,[105] receive data and input from Google's first-party apps,[106] and provide read and write access for integrations with Google's first-party apps.[107] These measures exceed the boundaries of Article 6(7).

(120) First, extending the interoperability obligation to the gatekeeper's separate services

---

102 DMs, ¶15(a).

103 Article 6(7) refers to features "*available to services or hardware <u>provided by</u> the gatekeeper"* (first sentence), and to "*the same operating system, hardware or software features"* as are available to, or used by, the gatekeeper "*<u>when providing</u> such services*" (second sentence) (emphases added). It follows that the features within scope of Article 6(7) are only those "available to" a gatekeeper when used to provide hardware or services.

104 DMs, ¶¶ 16(a) and 140.

105 DMs, ¶34(j).

106 DMs, ¶34(e).

107 DMs, ¶75.

would violate Article 6(1), which expressly limits Article 6(7)'s reach to "*core platform services listed in the designation decision*." It is plainly beyond the scope of Article 6(7) to require interoperability with gatekeepers' services that are not designated OSes or VAs.

(121) Second, Article 6(7) limits interoperability requirements to features that are "*accessed or controlled via*" the designated OS. The DMs would purport to allow third parties to interoperate directly with Google's first-party services, even if the existing integration between Gemini and Google's first-party apps is purely cloud-based with no OS mediation.

(122) Contrary to the DMs' suggestion,[108] Article 6(7) does not require providing access to the services provided together with, or in support of, the designated OS ("**Support Services**"). The second sentence of Article 6(7)—which contains a reference to Support Services—refers to them as potential *beneficiaries* of interoperability, not the software or hardware features with which interoperability must be provided. In fact, it specifically refers to "*alternative providers*" of Support Services as potential beneficiaries of the obligation. Nothing more. This continuous trend of the EC seeking to adopt the most radical reading possible of DMA provisions does a disservice to the need for the enforcer to be impartial and be seen to be impartial.

(123) In any event, the second sentence of Article 6(7) was not intended to expand the scope of the first sentence, which explicitly requires that hardware or software features to which interoperability applies must be accessed or controlled—i.e., mediated—via the designated OS. The reference to interoperability applying to OS-mediated features "*regardless of whether those features are part of the operating system*" simply refers to the fact that Article 6(7) can apply to hardware or software features that are *mediated* via the OS, but which do not themselves fulfil the DMA's definition of OS software (such as hardware components). It does not however extend the interoperability obligation to integrations that involve no OS mediation whatsoever.

(124) A broader interpretation would effectively convert Article 6(7) into a general interoperability and data-sharing obligation with services that a gatekeeper chooses to distribute on its designated OS, irrespective of whether those services are designated CPSs. This approach has no basis in the law or its legislative intent.[109]

C. Legal error #3: Treating the same software as both features subject to interoperability and services benefitting from interoperability

(125) The DMs appear in several places to define certain software components as both OS-mediated features *subject to* interoperability and services *benefitting from*

---

108 *See, e.g.*, DMs, ¶80(a) ("*Alphabet shall allow third parties to access, find, and process data from Alphabet's services provided together with, or in support of, Google Android – in a way that is equally effective to what is available to Alphabet's services, such as Gemini assistant.*").

109 Commission, DMA Impact Report Assessment (15 December 2020), p. 115, focused on the control of a gatekeeper over the OS to limit contestability of its own services. Opening Decision, ¶¶24–25 acknowledge this limitation.

interoperability. For example, the DMs appear to define Google's ODMs (*e.g.*, Gemini Nano) as OS *features* with which Google must provide interoperability.[110] Yet in a subsequent section, the DMs treat third-party ODMs as competing *services* that benefit from interoperability with OS features, seeking to mandate access to those OS features that Google's ODMs interact with.[111] A similar issue appears with respect to the DMs' treatment of Android System Intelligence ("**ASI**").[112]

(126) Under Article 6(7), however, the same software component cannot simultaneously be a service benefitting from interoperability and an OS-level feature subject to interoperability. This is not simply inconsistent with the letter of the law—it cuts across the basic structure of Article 6(7) DMA, which depends on a clear distinction between the layer subject to interoperability and the software that benefits from it. By collapsing that distinction, the DMs stretch the prevision beyond its text and logic in pursuit of policy objectives that cannot override the limits of the law in a regulated framework. If left unaddressed, this risks expanding Article 6(7) across all designated OSes. This distinction is fundamental to Article 6(7)'s operation. And it has consequences for the level in a gatekeeper's technology stack that Article 6(7) opens up to competition, as the figure below demonstrates:

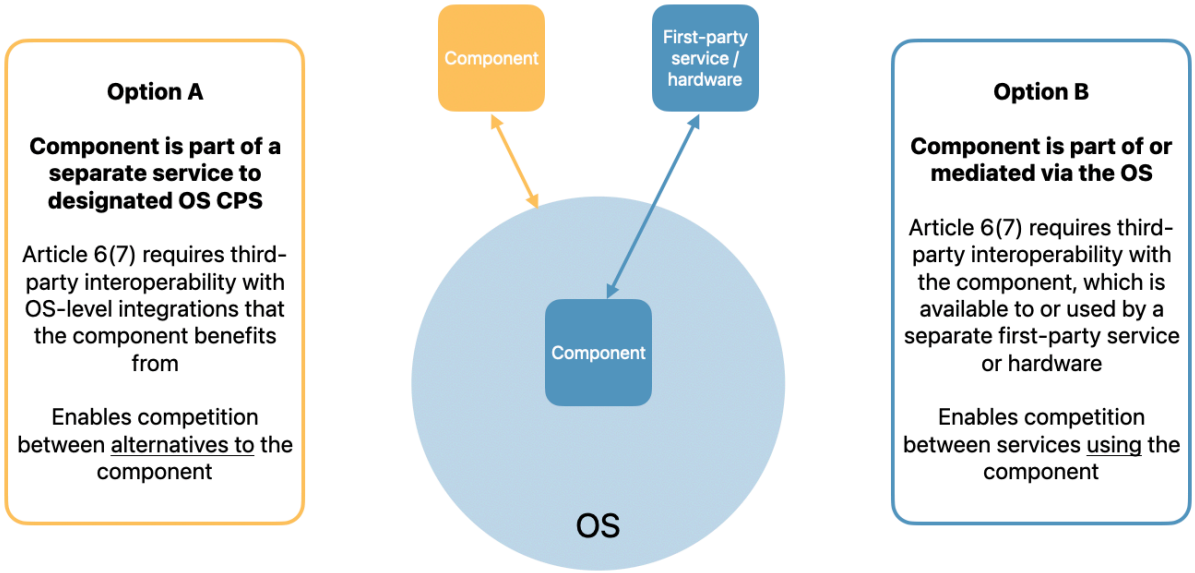
---

110 *See* Commission Case Summary of 27 April 2026, Case DMA.100220 - Alphabet - OS - Google Android - Art. 6(7) - SPA - AI ("**Case Summary**"), ¶11(c) (requiring Google to "*grant access to all system-level ODMs and their functionalities*" so that "*[t]hird parties must have access to these ODMs under equal conditions as Alphabet's own services*.")

111 For example, the DMs require Google to "*allocate hardware resources to its own on-device models [...] and third-party on-device models [...] according to transparent, objective, precise, and non-discriminatory rules*": DMs, ¶108(a). They also stipulate that if Google's ODMs enjoy "*preferential access to hardware resources*," the user must be allowed to "*remove such preferential access from Alphabet's on-device models [...] and grant it to third-party on-device models*": DMs, ¶108(b).

112 On one hand, the DMs appear to require interoperability with ASI as a system-level component: DMs, ¶¶40–41. On the other hand, the DMs appear to require Google to open the functionalities that ASI itself uses (*e.g.*, the user's physical context): DMs, ¶42(a)(1). According to the Case Summary, this measure is aimed at enabling "*access for third parties to functionalities necessary to offer their own alternative proactive suggestions solutions"*: Case Summary, ¶5(b).

Figure 4: Article 6(7) distinguishes between OS-level features subject to interoperability and services benefiting from interoperability



(127) If a given component is a separate service to the designated OS (Option A), Article 6(7) seeks to promote competition *with* that component by enabling third-party components to interoperate with OS-level features on an equal footing. It does not however require interoperability with the gatekeeper's component itself, which is by definition not accessed or controlled via the OS. If, by contrast, the component is part of or mediated via the OS (Option B), Article 6(7) seeks to promote competition between services *using* the component by facilitating interoperability with the component itself. But in those circumstances, Article 6(7) does not apply to interactions between the component and other OS-mediated components, as there is no separate service involved (as Article 6(7) requires).

(128) If ODMs and ASI are OS-level *features*, Article 6(7) provides for interoperability with them, rather than the OS-level components they interact with. If, on the other hand, these technologies are *services*, then Article 6(7) does not reach them at all—they are beneficiaries, not subjects, of the interoperability obligation. Article 6(7) would however then apply to their interactions with OS-level technologies. The DMs seem to avoid this distinction by treating ODMs and ASI as both features and services, which allows the EC to claim the broadest possible scope: the technology must be opened as infrastructure, and simultaneously competed against as a service.

D. Legal error #4: Seeking to extend interoperability beyond the use cases a gatekeeper pursues with its own services

(129) The DMs also err insofar as they would require a gatekeeper to enable interoperability for use cases beyond those its first-party services currently pursue—"*including for use cases that Alphabet does not offer*." This requirement appears across the general measures and the specific measures for system-level ODMs, ODM implementation, and background execution.

(130) Article 6(7) is a parity provision. It requires interoperability with features that are

"*available to or used by*" the gatekeeper for its own services or hardware. That language is directed at eliminating asymmetries between the gatekeeper's own services and competing third-party services seeking to use the same OS features. Where the gatekeeper does not provide a competing service, no such asymmetry exists, and Article 6(7) has no disparity to remedy. Extending the obligation to use cases the gatekeeper does not itself pursue would transform Article 6(7) from a parity mechanism into an open-ended platform access right—a fundamentally different instrument that the legislature did not enact.

(131) This reading is confirmed by the relationship between Article 6(7) and the integrity safeguard it contains. Article 6(7) expressly permits gatekeepers to adopt "*strictly necessary and proportionate measures*" to protect the integrity of the OS. Use-case limitations are among the most effective means of doing so: they allow the gatekeeper to assess and manage the risks associated with specific types of access rather than granting access on an open-ended, use-case-agnostic basis.

(132) As explained in **Section IV**, certain functionalities are too sensitive to be made available for use cases that neither the gatekeeper nor the OS can anticipate. This is particularly critical in the context of AI-driven functionalities, where the risks are evolving rapidly and remain fundamentally unpredictable. Requiring use case-agnostic interoperability would deprive the integrity clause of much of its practical effect, since the gatekeeper could not meaningfully evaluate proportionality in the abstract—it can only do so against a defined set of use cases with identifiable risk profiles.

(133) If these aspects of the DMs were allowed to stand, gatekeepers could be required to provide interoperability solutions for scenarios irrelevant to their services and hardware, and therefore not advancing the goal of contestability. It would also remove one of the most effective integrity safeguards available to OS providers at precisely the moment when the risks from AI-driven access are most acute. Irrespective of the specific application in this case, this would set a precedent for expanding Article 6(7) beyond its intended scope across all designated OSes.

VI. Conclusion

(134) The DMs represent a bold attempt to broadly expand the limits of Article 6(7), apparently ignore the significant privacy, security, integrity, and performance risks they would create, and downplay the relevance of OS-imposed safeguards. Before any final decision is taken, we urge the EC to significantly narrow the scope of its obligations to: (i) reflect the serious privacy, security, and performance risks they would result in; and (ii) account for a broader range of potential platform-imposed integrity measures, especially in the context of the unprecedented security and privacy challenges posed by AI. The DMs must be updated to also remove any obligations that go beyond what Article 6(7) permits, as set out in this paper. It must consult relevant privacy and cybersecurity bodies like the EDPB and ENISA.

* * *