

CASE SUMMARY
CASE DMA.100220 - ALPHABET - OS - GOOGLE ANDROID - ART. 6(7)
FEATURES FOR AI OR AI-RELATED SERVICES

Background

On 5 September 2023, the Commission designated Alphabet as a gatekeeper under the Digital Markets Act (DMA) for a number of its services, including its operating system Google Android. Google Android is defined as a system software that controls the basic functions of the hardware or software and enables software applications to run on it. Google Android includes the Android Open Source Project, which provides the foundational framework for Google Android, as well as Alphabet's middleware, insofar as it contributes to controlling the basic functions of Google Android tablets and smartphones and contributes to enabling software applications to run on them. This designation made Google Android subject to the obligations set out in the DMA.

Under Article 6(7) of the DMA, Alphabet must, free of charge, provide third parties with effective interoperability with, and access for the purposes of interoperability to, hardware and software features accessed or controlled via its operating system, Google Android. Access is also to be provided to features which are available to, or used by Alphabet when providing services provided together with, or in support of Google Android, regardless of whether those features are part of the operating system.

On 27 January 2026, the Commission opened proceedings pursuant to Article 20(1) of the DMA with a view to specify the measures Alphabet must implement to effectively comply with its interoperability obligations with respect to features which are relevant for third-party providers of artificial intelligence (AI) services on mobile devices (i.e., smartphones and tablets).

On 27 April 2026, the Commission adopted its preliminary findings setting out the proposed measures, which are subject to this public consultation.

Summary of the Commission's preliminary findings

The smart mobile device market is at a technological inflection point, with AI services becoming central access points on mobile devices. In parallel, there is a clear consumer demand for AI-powered devices supported by an industry-wide push towards AI.

AI based services such as AI assistants are gaining user traction on mobile devices as they allow users to navigate their mobile habits with increasing ease of use. Through access to the key components of the AI value chain including the infrastructure (Google cloud), foundational models (Gemini family of LLM models), platform integration (web of integrated services such as messaging, Google docs, Google Search, Google Maps etc.) and distribution (through its various services and Google Android), Alphabet is uniquely placed to leverage its strength into this new area.

With around 60% of mobile users in Europe owning a Google Android smart mobile device, mobile devices powered by Google Android represent a crucial gateway for providers of stand-alone AI services to reach end users.

Interoperability with Google Android is therefore critical for third-party AI service providers to compete with Alphabet's AI-based services, such as Gemini. For example, any AI service provider that wants to offer its services to users of smart mobile devices powered by Google

Android needs its services to be able to interact with the device hardware, have access to invocation mechanisms such as voice-based hotword activation, the user's context and data, integration with other apps, and underlying system services. Such access would ensure that alternative AI services can provide at least comparable performance, responsiveness, visibility, functionality, and utility to end users with Alphabet's AI services.

The preliminary findings take a preliminary view that Alphabet, as a designated gatekeeper that controls Google Android, needs to ensure that that Google Android fully complies with the DMA obligations on all devices on which it is implemented.

Given the complexity of the Alphabet ecosystem, the Commission decided, through these proceedings, to provide Alphabet with guidance as to how to comply with its obligations under the DMA when it comes to a defined set of features pertaining to the invocation of services, access to contextual information, ability to execute actions on apps and the operating system and ability to access device resources.

The draft measures set out in the Annex do not impact the ability of OEMs to continue differentiating their smart mobile devices in various ways, such as through pre-installing certain apps, setting default services, offering innovative complementary software and user interface layouts, as well as on the basis of hardware features (such as camera, battery, memory space, etc.) or providing attractive experiences for users and maintaining branding.

Furthermore, the draft measures clarify, in line with Article 6(7) of the DMA, how Alphabet may take strictly necessary and proportionate measures to ensure that interoperability does not compromise the integrity of the operating system, hardware and software features, within pre-defined timelines.

Interested third parties are now consulted on these measures, and in particular on their effectiveness, completeness, the integration effort concerning the measures, and implementation timelines. These measures are preliminary and might be adjusted following the feedback gathered from third parties and Alphabet, as well as further investigative steps. By 27 July 2026, the Commission will adopt its final decision in these proceedings setting out the binding measures Alphabet will need to implement.

Summary of the features and the draft measures

The below constitutes a high-level summary of the features and the draft measures. For more complete information about the draft measures, please refer to the Annex.

- I. **Features for invocation**, or the ability for an end user to initiate their interaction with an AI service through a variety of access points.
 - (1) *Contextual invocation through long-press home button/navigation handle*
 - (a) Long pressing the home button or navigation handle is a crucial access point, available on all Android devices. Placed at the bottom of the screen, it is available on all Android phones. This access point allows AI services to be invoked by users at any time of using their device. Invocation through that access point enables an app to overlay on top of the screen and offer a service based on contextual data, such as screen content.
 - (b) That access point is however not equally available to third party developers as it is to Alphabet's services as it is often mapped to Alphabet's Circle to Search.
 - (c) Under the proposed measures, Alphabet will have to provide effective interoperability with all functionalities of contextual invocation through providing access to the LPH/LPNH access point, including, but not limited to,

ability to be invoked through the access point, ability to overlay on top of the device's screen, and ability to receive contextual data. Moreover, users must be able to customize other access points to apps of their choice, with the exception of the access point reserved for the default assistant role.

(2) *Always-on hotword detection*

- (a) The always-on hotword detection feature allows apps to be invoked through user uttering a hotword or wake word. Hotwords allow users to invoke and interact with their favourite assistant hands-free. The feature is implemented in a power-saving manner that allows always-on hotword detection to run continuously, even when the phone screen is turned off and the phone is in battery saver mode.
- (b) Currently, the agent-controlled interactions feature on Google Android is available to and used by Alphabet for Gemini and Google Assistant. Users can wake the assistant by speaking the hotword "Hey Google" and then continue to make requests to the assistant. This feature is not available to user-installed third-party apps.
- (c) Under the proposed measures, Alphabet will have to provide effective interoperability with all functionalities of the always-on hotword detection feature, through two different implementations. This includes allowing third-party user-installed apps to register a custom, developer-defined hotword in Google Android via a digital signal processing (DSP) chip, specialised for executing efficiently tasks such as hotword detection. Aside from developers implementing a developer-defined hotword for their app, Alphabet has to allow third-party apps to allow their users to create a custom hotword for the developer's app. Alphabet will have to ensure that the additional hotwords will be supported concurrently to the existing Google hotword, e.g. "Hey Google", within the technical limits of the device.

II. Features for context, or the ability for an AI service to gather inputs and data to then understand and anticipate end users' needs and provide helpful suggestions across apps.

(3) *Centralised access to apps' data stored on-device*

- (a) App data, and in particular app data that is exclusively stored on-device, is an essential input for AI services to understand the user and provide personalised services. Access to such data in a centralised manner, as opposed to on an app-by-app basis, removes the need for AI services having to engage in one-on-one integration with every app developer.
- (b) Centralised access to apps' data stored on-device is currently enabled via an AppSearch role-specific permission, linked to the default assistant role. The access to this permission is currently limited to the default assistant and cannot be concurrently obtained by all the assistants that may be installed on a device. In addition, third-party apps do not enjoy equal access to the data donated by Alphabet's first-party apps, as access to that data is granted to Alphabet's services only.
- (c) Under the proposed measures, Alphabet will have to provide effective interoperability with all functionalities of the centralised access to apps' data stored on-device feature, which are available to Alphabet's services and hardware. This includes enabling concurrent access to data that is stored

centrally on-device and that is shared on an opt-in role-based basis, as well as to data that is stored by Alphabet's apps and that is shared with Alphabet services.

(4) *Proactive suggestions*

- (a) Proactive suggestions surface relevant information and recommend helpful actions across different apps. These suggestions are based on the user's data collected from apps, recently viewed content, and the user's context in general. For instance, Pixel's Magic Cue can surface the relevant flight number when calling an airline or suggest helpful actions such as "View calendar" when a friend sends an invitation to dinner via a message.
- (b) Currently, the main apps which can participate in proactive suggestions, either by donating input to be surfaced in other apps, or by being themselves the apps that surface a proactive suggestion, are Alphabet's first-party apps and services, including Gemini.
- (c) Under the proposed measures, Alphabet will have to provide effective interoperability with all the functionalities of the proactive suggestions feature, including, but not limited to, the ability to be part of inputs and the ability to be part of outputs for proactive suggestions.
- (d) For example, third-party messaging apps will be able to surface the relevant flight number when the user is asked about an upcoming flight, even if the flight number is stored in a (participating) third-party email app and not Gmail. In addition, users will be able to receive proactive suggestions to use third-party AI services for a specific prompt which results from a text message received from a friend, like searching for a restaurant.

(5) *Context-aware intelligence*

- (a) Context-aware intelligence offers users their own version of proactive suggestions and other "intelligent experiences", that is personalized to the user's context and is continuously present to enhance the user's device experience. This feature is comprised by multiple functionalities which includes both inputs from a device sensor but also the ability to process these inputs to enable proactive actions. This functionality may be leveraged to for example provide live translation or recognizing songs by using microphone data. These renders the use of AI services effortless for users and give a sense of true assistance and deep integration with what the user is doing, at any time.
- (b) The proactive suggestions feature and the context-aware intelligence feature are two sides of the same coin, and enable interoperability at high and low level, respectively. On the one hand, the proactive suggestions feature covers access for third parties to existing proactive suggestions solutions provided by Alphabet (such as Magic Cue), by letting third parties be part of what is suggested to the user, through contributing data and surfaces. On the other hand, the context-aware intelligence feature covers access for third parties to functionalities necessary to offer their own alternative proactive suggestions solutions (for instance, an alternative to Magic Cue) by enabling third parties to access the same resources as Alphabet (e.g. data and surfaces contributed by others).

- (c) The ability to offer proactive suggestions and intelligence experiences depends on several functionalities, such as:
 - inputs from the user’s physical context (e.g. microphone), digital context (e.g. on-screen content), and cross-app data, and this continuously and in the background;
 - processing resources such as machine-learning models and environments to treat data securely and efficiently; and
 - display surfaces such as screen overlays, keyboard “chips”, and surfaces inside, on top of, and outside apps.
- (d) Currently, context-aware intelligence is primarily implemented through Alphabet’s Android System Intelligence component. Alphabet only makes the functionalities accessible by OEMs and Alphabet itself.
- (e) Under the proposed measures, Alphabet will have to provide effective interoperability with all functionalities of the context-aware intelligence feature, including, but not limited to, access to the aforementioned inputs, resources, and outputs.
- (f) For example, third-party apps will be able to offer proactive suggestions and intelligent experiences similar to Now Playing (showing the name of songs being played nearby, using ambient microphone data), Live Translate (overlaying a real-time translation of what is heard or seen on the screen) or Magic Cue (explained above in Section (4) on proactive suggestions).

(6) *Ambient data*

- (a) Ambient data is the continuous stream of real-time inputs/outputs from a device’s core sensors (e.g., microphone, camera, screen, speakers). It provides the foundation that enables AI services to deliver personalised, context-aware experiences such as sound detection, live screen guidance, or object recognition by processing raw sensor data seamlessly.
- (b) Currently, Alphabet’s first-party services benefit from system-privileged access to the ambient data. For certain services this encompasses continuous access to camera, microphone, screen and speakers of the device, with reduced consent processes and privacy indicators. In contrast, third-party AI services need to rely on restricted public APIs that are developed and controlled by Alphabet itself. This means that in practice third parties face fragmented access (runtime consent per use and privacy indicators) as well as higher latency, battery drain and possibly lower data quality and lastly differentiated basis for innovating on top of ambient data.
- (c) Under the proposed measures, Alphabet will have to provide effective interoperability with all functionalities for access to ambient data, such as ensuring that third-party AI services have access to real-time sensor inputs including microphone, camera, screen, and speakers on equal terms as Alphabet’s own services. This includes equivalent data quality, latency, background processing, and consent flows.
- (d) For example, third-party apps would be able to leverage the end user’s contextual and ambient data to offer novel ways of personalised assistance, such

as live camera guidance, or power other services, such as agent-controlled actions on the user's screen.

III. **Features for actions on apps and the OS**, or the ability for an AI service to execute tasks on user's behalf.

(7) *Structured on-device integration*

- (a) Structured on-device integration enables an AI service to take actions inside other apps on the device in response to user request.
- (b) Alphabet's services have access to the structured on-device integration feature on Google Android, as currently implemented via OS-level integration channels such as App Actions and App Functions. User-installed third-party apps cannot effectively access these channels. They face difficulties discovering App Actions. Furthermore, App Functions can only be accessed by apps which have been granted privileged permissions.
- (c) Under the proposed measures, Alphabet will have to provide effective interoperability with all functionalities of the structured on-device integration feature which are available to Alphabet's own services and hardware. This includes access for third-party apps to discover and execute App Actions and App Functions of other installed apps. Developers will be able to deeply integrate with apps on the Android device, as Google Gemini and Google Assistant do today.
- (d) As a result of interoperability with structured on-device integration, apps would be able to execute user requests such as finding a recipe from a message received by the user and adding the ingredients to the user's shopping list.

(8) *Screen automation*

- (a) Agentic AI assistants may take control of other apps to automate multi-step tasks on behalf of the user. These services can imitate user behaviour and observe the controlled app's screen to automate the user interface and to perform most tasks a user would be able to do. An agentic AI assistant can run the controlled app in a separate virtual window, allowing the assistant to complete the task in the background, while the user can do something else.
- (b) Currently, the screen automation feature on Google Android, as currently implemented via the Computer Control API, is available to and used by Alphabet for Gemini's screen automation of Android apps. This feature is not available to user-installed third-party apps.
- (c) Under the proposed measures, Alphabet will have to provide effective interoperability with all functionalities of the screen automation feature.
- (d) If implemented, third-party developers will be able to offer multi-step experiences such as automating hotel booking apps to compare prices and recommend a set of hotels, or automatically placing online orders based on chat messages with friends.

(9) *Integration with first-party services*

- (a) Google Android devices come with a bundle of Alphabet apps and services that are seamlessly integrated with the OS and with each other. These include

Gemini assistant, Gmail, Google Calendar, Google Drive, Google Photos, Google Maps, and YouTube.

- (b) Currently, third-party AI assistants on Google Android do not have the same level of integration with Alphabet's apps and services that is available to Alphabet's AI services, such as Gemini.
- (c) Under the proposed measures, Alphabet will have to provide effective interoperability with the features that are used to integrate Alphabet's AI services with other Alphabet apps and services that are deeply integrated with Google Android. This includes access to the user's personal context, subject to user consent.
- (d) For example, third-party AI assistants will be able to help the user to schedule events in Google Calendar or find relevant photos in Google Photos. Third-party AI assistants will also be able to answer questions about YouTube videos or helping with navigation in Google Maps.

(10) *System integration*

- (a) AI services and assistants may want to interact and integrate with the operating system on behalf of the user. The user may request the assistant amongst other things to change a specific setting, to interact with playing media, or to open an app.
- (b) The system integration feature on Google Android, as currently implemented via a range of APIs, is available to and used by Alphabet for Gemini. Gemini can interact and integrate with the operating system. However, user-installed third-party apps are limited to a reduced set of system integrations.
- (c) Under the proposed measures, Alphabet will have to provide effective interoperability with the features that are used to integrate with the operating system. This includes allowing access to the same integration points that Gemini can use today, such as controlling music, changing settings, or opening apps on the device.
- (d) If implemented, a third-party assistant will be able to directly control common system settings based on a user request, such as turning off Bluetooth off or pausing the playing music.

IV. **Features for access to resources**, or the ability for AI service to use sufficient hardware and software resources to execute their tasks.

(11) *System-level on-device models (ODMs)*

- (a) On-device models are machine learning models, including large language models, that run directly on the device hardware. Compared to cloud-based models, they offer lower latency, offline availability and enhanced privacy protection. System-level ODMs are part of Google Android (including middleware) and include the Gemini Nano ODMs on many Google Android devices.
- (b) As ODMs are an evolving area, there is significant uncertainty as to how Alphabet will make system-wide ODMs accessible in the future, including which system-wide models will be made available and under what conditions

(e.g., applying quotas restricting the usage of the ODM to a certain amount of time). Alphabet currently uses private APIs to access certain Gemini Nano ODM functionality.

- (c) Under the proposed measures, Alphabet will have to grant access to all system-level ODMs and their functionalities. Third parties must have access to these ODMs under equal conditions as Alphabet's own services, including with respect to conditions and customizability. Third parties must be allowed to use the system-level ODMs for any use case.
- (d) If implemented, user-installed apps will, for example, be able to use Gemini Nano ODM functionality for offering services such as summarisation, proofreading, or speech recognition.

(12) *On-device model (ODM) implementation*

- (a) Third-party developers seek to develop and offer ODMs (and software to implement such ODMs) and use them on Google Android devices. Such third-party ODMs can enable innovative functionality or efficiency gains. These ODMs may be used by a developer's own app or other developers' apps. These ODMs require access to hardware resources to function properly.
- (b) Alphabet currently uses certain APIs not accessible to third parties. Alphabet's AICore currently enjoys preferential access to RAM. More generally, AICore and the Gemini Nano ODMs currently provide functionality both for Alphabet's first-party apps and system functionality (such as Alphabet System Intelligence). This may result in preferential access for Alphabet's first-party apps as they can rely on ODM functionality enjoying preferential access to hardware resources to execute system functionality.
- (c) Under the proposed measures, Alphabet will have to grant third parties the same ability to implement and use ODMs as it grants its own models. This includes applying equal conditions for access to hardware resources and background execution and the ability to make third-party ODMs centrally available to other apps. This can be achieved by allowing the user to grant such preferential access to a third-party ODM while removing it from Alphabet's ODM to ensure the availability of sufficient hardware resources for third-party ODMs.
- (d) If implemented, third parties offering ODMs for improved translation will, for instance, be able to offer their services to users without facing different restrictions in terms of memory access or needing to be always in the foreground to function.

(13) *Background execution*

- (a) Background execution allows an app to execute actions in a timely manner in the background, including when the user is not actively looking at the app or is not using the device. Background execution is particularly useful for AI-related apps, which in many cases are designed to help the user while performing other tasks in other apps, or even to automate tasks while the phone's screen is off.
- (b) Currently, background execution for third-party apps on Google Android is limited in several ways. While limitations are necessary to manage power consumption, such limitations are not always applied fairly. On the one hand, Alphabet and OEMs can exempt their own apps and services from any limitation; on the other hand, they can apply further restrictions to specific third-

party apps, affecting the user experience and developers' ability to provide a reliable service. In either case, the user is not always allowed to modify such exemptions and restrictions or can be required to follow complex and high-friction procedures to do so.

- (c) Under the proposed measures, Alphabet will have to ensure that user-downloaded apps have the same access to background execution as Alphabet's apps. In particular, Alphabet will have to define transparent, objective, precise and non-discriminatory rules that also apply to pre-installed apps. Alphabet will also have to ensure that users can grant any app access to the same level of background execution that Alphabet requires OEMs to grant to its own apps.
- (d) If implemented, third-party AI services will, for instance, be able to execute user's requests reliably, even if the user turns off the screen for some time or switches to another app while waiting for the AI service to process its request.

V. General measures for all features

Under the proposed measures, Alphabet will have to implement a number of general measures for all the features listed above. For example:

- (a) Alphabet will have to ensure that interoperability is offered free of charge, via complete, accurate and well-documented frameworks and APIs on all Google Android devices, including on devices supplied by OEMs.
- (b) Alphabet will have to ensure that the features listed above will be accessible to all apps, including user-installed apps and pre-installed apps without system privileges.
- (c) Alphabet will have to ensure that the interoperability solutions for third parties will have to be equally effective to those available to Alphabet and must not require more cumbersome system settings or additional user friction.
- (d) Alphabet will have to ensure that access to features is not subject to the app having a default role, including the default assistant role.
- (e) Any integrity measure taken by Alphabet will have to be duly justified and based on transparent, objective, precise, and non-discriminatory conditions that also apply to Alphabet's services and hardware.
- (f) Alphabet will not be prevented from requiring user consent to access the features, provided that the requirement is based on transparent, objective, precise, and non-discriminatory conditions that also apply to Alphabet's services and hardware.
- (g) Alphabet will have to make available to third parties any new functionalities of the listed features once they become available to Alphabet.
- (h) Finally, Alphabet will have to provide third parties with the required technical assistance and report how it implemented the measures.

Annex - Measures

1. MEASURES ON FEATURES FOR INVOCATION

1.1. Measures for Long-press home/Long-press navigation handle contextual invocation

- (1) Alphabet shall provide effective interoperability with the *Long-press home (LPH)/Long-press navigation handle (LPNH) contextual invocation* feature.
- (2) The LPH/LPNH contextual invocation feature is described in Section 8.2.2 of the Preliminary Findings. The feature enables Alphabet to use Google Android access points such as LPH and LPNH. These access points enable Alphabet's services and hardware to be invoked by users at any time while using the device. Invocation through the LPH and LPNH functionalities enables Google Search to overlay on top of the screen and offer a service based on contextual data, such as screen content. Alphabet advertises the implementation of this feature as Circle to Search, allowing users to circle parts of their screen to search for the content.
- (3) Alphabet shall implement an interoperability solution that provides third parties ⁽¹⁾ with access to the same Google Android feature (as described in the preceding paragraph) and all its functionalities, as available to Alphabet's services, including to Google Search via Circle to Search, and in a way that is equally effective as the solution available to Alphabet services.
- (4) These functionalities are:
 - (a) Ability to be invoked system-wide through the LPH/LPNH access points and any other access point.
 - (b) Ability to receive, upon invocation, to receive contextual data, including but not limited to a current screenshot, information about the open apps, information provided by open apps, such as a URL, picture, or additional data.
 - (c) Upon invocation, the ability to overlay content of any current activities on the user's device.
- (5) Alphabet shall grant third parties access to additional functionalities available to Alphabet's services if necessary to enable effective interoperability with the feature referred to in paragraph (2).
- (6) To provide third parties with an interoperability solution for the LPH/LPNH contextual invocation feature and functionalities referred to in paragraphs (2) and (4) that is equally effective as that available to any of Alphabet's own services or hardware, Alphabet shall implement the following measures:
 - (a) Alphabet shall make any APIs relied upon by Google's Circle to Search (including Contextual Search APIs, as well as any other API relied upon by Alphabet's services, such as Google Search, to be invoked through the

⁽¹⁾ The terms "third parties" and "third-party apps" across this Annex refer to all apps, including user-installed apps, pre-installed apps without system privileges, and apps of original equipment manufacturers (OEMs).

LPH/LPNH access point or Circle to Search access point), public and accessible to all apps, regardless of whether pre-installed or user-installed.

- (b) Alphabet shall ensure that the user can customise the LPH/LPHN access point and any other access point, subject to the possibility of OEMs to reserve one access point ⁽²⁾ other than the LPH/LPHN to the default assistant role.
 - (c) Alphabet shall not reserve, including through technical or contractual means, any access point on Google Android mobile devices for its services (subject to paragraph (6)(b)).
 - (d) Alphabet shall ensure that the user can select which app is invoked through any such access point. The mechanism allowing the user to exercise this choice must be non-discriminatory, including in terms of user interface. Alphabet shall ensure that when a third-party app gets invoked through such an access point, it can receive contextual data. Alphabet shall provide effective interoperability with any future updates, including new functionalities, of the LPH/LPNH contextual invocation feature insofar as they are available to Alphabet's own services or hardware.
- (7) Alphabet shall implement the measures above in compliance with the measures for all features in Section 5 of this Annex.
 - (8) Alphabet shall implement the measures for the LPH/LPNH contextual invocation feature by 1 January 2027.

1.2. Measures for always-on hotword detection

- (9) Alphabet shall provide effective interoperability with the *always-on hotword detection (AOHD)* feature including through *third-party sound models* and *user customised hotword*.
- (10) The AOHD feature is described in Section 8.3.2 of the Preliminary Findings. The feature enables Alphabet services and hardware to be invoked through the user uttering a hotword or wake word. Hotwords allow users to invoke and interact, hands free, with their favourite assistant. The feature is implemented in a power-saving manner that allows always-on hotword detection to run continuously, even when the phone screen is turned off and when the phone is in battery saver mode. The feature can be implemented based on specific hardware, i.e. a digital signal processor (DSP) which depends on the characteristics of the device or through a user customised hotword.
- (11) Alphabet shall implement an interoperability solution that provides third parties with access to the AOHD Google Android feature (as described in the preceding paragraph) and all its functionalities, as available to Alphabet's services, and in a way that is equally effective as the solution available to Alphabet services.
- (12) The functionalities are:
 - (a) Sound model enrolment. The ability to enrol a sound model in the Android operating system to persistently store and retrieve the sound model upon request. An enrolled sound model can be loaded on the DSP when the app starts always-on hotword recognition.

⁽²⁾ For access point which can be invoked in different ways, such as long press, single tap and double tap, each such way of invocation constitutes a different access point.

- (b) First-stage detection on the DSP. The ability to load and execute its enrolled sound model on the DSP, and to ensure that the DSP performs the first stage hotword detection on the app's behalf.
 - (c) Second-stage validation. The ability to run a second stage validation after the DSP sound model has recognised the hotword in the first-stage detection, to confirm that the hotword was detected correctly. The second stage runs in a secure and isolated process, disallowing to extract audio data before the hotword was validated.
 - (d) Invocation of the app. The ability of the app to be invoked (once the second stage validation confirms the hotword detection) and to handle the request of the user. This includes showing a user interface that overlays any previously shown user interface on the device. In addition, the app receives contextual data on invocation.
 - (e) Continued access to record audio. The ability, after invocation, 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.
- (13) To provide third parties with an interoperability solution for the AOHD feature referred to in paragraph (10) that is equally effective as that available to any of Alphabet's services, Alphabet shall implement the measures in paragraphs (14), (15) and (16) below.
- (14) To that end, Alphabet shall implement the following measures to enable always-on hotword detection *using third-party sound models* supporting hotwords defined by the third-party developer.
- (a) Alphabet shall prepare and provide developers with a detailed documentation which is required on how to create a first stage DSP sound model compatible with Alphabet's DSP library. As part of this documentation, Alphabet shall specify the TFLM operators that the DSP library on Google Android supports, which input the model receives and how the model must structure its outputs. Alphabet shall enable third-party developers to implement custom first stage DSP sound models solely based on the documentation. Alphabet may provide sample implementations to aid developers.
 - (b) Alphabet shall not hinder third-party developers to autonomously innovate on their DSP sound models by building a custom model training pipeline that can be used create these sound models. Alphabet shall not hinder developers to implement a specific model. This does not preclude that third-party developers may need to meet certain implementation criteria that are necessary to preserve integrity, e.g. that their model must use TFLM operators that are supported by the DSP library.
 - (c) Alphabet shall create an API or open existing APIs to allow third-party developers to enrol these first stage sound models for always-on hotword detection within Google Android.
 - (d) Alphabet shall create an API or open existing APIs to allow third-party developers to load their DSP sound models onto the Google Android mobile device DSP and to run them using Alphabet's DSP library, to facilitate always-on hotword detection for the third-party app.

- (e) Alphabet shall enable third-party developers to test their custom DSP sound models on their Android mobile devices, without any prior agreements and prior validation from any other party.
 - (f) Alphabet shall enable third-party developers to implement a second stage hotword detection. Equally to Alphabet's implementation, the second stage must receive an audio snippet to validate the detected hotword. Alphabet shall enable the second stage hotword detection to run in a secure and isolated process, disallowing to extract audio data before the hotword was validated.
 - (g) Alphabet shall enable third-party developers to invoke their app after the app validated the hotword in the second stage. Alphabet shall ensure that the third-party app also receives the audio snippet of the detected hotword and that the app has continued access to the microphone.
 - (h) Alphabet shall make available to third-party developers testing tools, such as Alphabet's open source hotword stress test tool,³ that allow third-party developers to test always-on hotword detection on the third-party's Google Android mobile devices, enabling third parties to verify that their sound models correctly recognise the hotword phrase. Alphabet shall continue to support this tool.
- (15) In addition, Alphabet shall implement the following measures enabling always-on hotword detection *using user-customised hotwords*.
- (a) Alphabet shall make available to third-party developers a functionality that allows their users to create a custom hotword without leaving the developer's app.
 - (b) Alphabet shall ensure that the procedure to set up the user-customisable hotword must be frictionless and may not be cumbersome. In addition, the user interface, text, and wording must be in control of third-party developers.
 - (c) Alphabet shall allow a third-party app to suggest a custom hotword phrase, allowing the app to use a branded hotword.
 - (d) Alphabet shall ensure that two custom sound models can be created based on the user's voice recordings: one for the first stage hotword detection on the DSP and one for the second stage hotword validation. Alphabet shall ensure that the first stage sound model can be loaded onto the DSP and that the model will run using Alphabet's DSP library. Alphabet shall ensure that Google Android will run the second stage model in an isolated process to validate the hotword detection.
 - (e) Once the hotword is detected at both stages, Alphabet shall ensure that Google Android activates the third-party app. In addition, Alphabet shall ensure that the app receives the audio snippet of the hotword detection and that the app has continued access to the microphone.
- (16) With respect to both implementations of third-party always-on hotword detection, that is (15) and (16), Alphabet shall implement the following measures.

(³) See OpenHST: Open sourced Hotword Stress Test tool, available at <https://android.googlesource.com/platform/tools/test/openhst/>.

- (a) Alphabet shall ensure that always-on hotword detection from multiple third-party apps and first-party apps are able to run concurrently. Therefore, Alphabet shall ensure that multiple first stage DSP sound models can run at the same time. This means that a third-party hotword may run alongside the existing Google hotwords. Concurrent always-on hotword detection may be limited by the capabilities of the DSP on the Google Android mobile device.
 - (b) Alphabet shall enable third-party developers to use always-on hotword detection while the Google Android mobile device is in battery saver mode, as available to Alphabet's always-on hotword detection.
 - (c) Alphabet shall ensure that, upon invocation, the app receives access to all data and context available to Gemini, Google Assistant or any other Alphabet services.
 - (d) Alphabet shall ensure that users have control over which apps use always-on hotword detection, such as through user consent to enable a hotword. Alphabet shall implement a centralised settings page must allow the user to enable or disable hotword detection for any app. The mechanism allowing the user to exercise this choice should be non-discriminatory, including in terms of user interface. In particular, the setting page must include both third-party apps and Alphabet's apps.
 - (e) Alphabet shall not reserve the always-on hotword access points to the default assistant role.
- (17) Alphabet shall grant third parties access to additional functionalities available to Alphabet's services if necessary to enable effective interoperability with the always-on hotword detection feature referred to in paragraph (10).
 - (18) Alphabet shall provide effective interoperability with any future updates or implementations, including new functionalities, of the always-on hotword detection feature insofar and as soon as they are available to Alphabet's own services or hardware.
 - (19) Alphabet shall implement the measures above in compliance with the measures for all features in Section 5 of this Annex.
 - (20) Alphabet shall implement the measures with respect to third-party always-on hotword detection using third-party sound models for the always-on hotword detection feature, as defined in paragraphs (14) and (16) by 1 January 2027 at the latest. Alphabet shall implement the measures with respect to third-party always-on hotword detection using user-customised hotwords for the always-on hotword detection feature as defined in paragraphs (15) and (16) by 1 July 2027.

2. MEASURES ON FEATURES FOR CONTEXT

2.1. Measures for centralised access to apps' data stored on-device

- (21) Alphabet shall provide effective interoperability with the *centralised access to apps' data stored on-device* feature.
- (22) The centralised access to apps' data stored on-device feature is described in Section 9.2.2 of the Preliminary Findings. The feature enables access to apps' data that is stored on-device in a centralised manner through on-device databases or data sharing platforms, thereby allowing efficient cross-app data access, search and retrieval. For instance, AppSearch currently enables centralised access to apps' data

- stored on-device on Google Android, but only for the one app holding the default assistant role.
- (23) Alphabet shall implement an interoperability solution that provides third parties with access to the same Google Android feature (as described in the preceding paragraph) and all its functionalities, as used by and available to Alphabet's services, and in a way that is equally effective as the solution available to Alphabet services.
 - (24) Alphabet shall provide effective interoperability with all functionalities of the centralised access to apps' data stored on-device feature which are available to Alphabet's services and hardware.
 - (25) To provide third parties with an interoperability solution for the centralised access to apps' data stored on-device feature that is equally effective as that available to Alphabet's services, Alphabet shall implement the following measures:
 - (a) Alphabet shall ensure access by third parties, including third-party apps, to data that is stored centrally on-device and that is shared on an opt-in role-based basis. ⁽⁴⁾
 - (b) Alphabet shall ensure that data that is stored by Alphabet's apps and that is shared with Alphabet's services is equally accessible by third parties.
 - (c) Alphabet shall ensure that access to data covered in paragraph (25), letters (a) and (b) is possible on a concurrent basis and in a manner that ensures equal centralised access to apps' data stored on-device as is available to Alphabet's services.
 - (26) Alphabet shall provide effective interoperability with any future updates or implementations, including new functionalities, of the centralised access to apps' data stored on-device feature insofar and as soon as they are available to Alphabet's own services or hardware.
 - (27) Alphabet shall implement the measures above in compliance with the measures for all features in Section 5 of this Annex.
 - (28) Alphabet shall implement the measures for the centralised access to apps' data stored on-device feature in the release of Android 17 QPR2 and in any case, by 1 January 2027 at the latest.

2.2. Measures for proactive suggestions

- (29) Alphabet shall provide effective interoperability with the *proactive suggestions* feature.
- (30) The proactive suggestions feature is described in Section 9.3.2 of the Preliminary Findings. The proactive suggestions feature surfaces relevant information and recommends helpful actions across different apps. These suggestions are based on the user's data collected from apps, recently viewed content, and the user's context in general. For instance, Pixel's Magic Cue can surface the relevant flight number when calling an airline or suggest helpful actions such as "View calendar" when a friend sends an invitation to dinner via a message. Currently, apps which can

⁽⁴⁾ With AppSearch, this may be enabled by allowing all third-party apps to access the data stored with READ_ASSISTANT_APP_SEARCH_DATA visibility. However, the measure applies regardless of the technical implementation of the centralised access to apps' data stored on-device feature.

participate in proactive suggestions, either by donating input to be surfaced in other apps, or by being themselves the apps that surface a proactive suggestion, are mainly Alphabet's first-party apps and services, including Gemini.

- (31) Alphabet shall implement an interoperability solution that provides third parties with access to the same Google Android feature (as described in the preceding paragraph) and all its functionalities, as available to Alphabet's services and in a way that is equally effective as the solution available to Alphabet's services.
- (32) These functionalities are:
 - (a) The ability to be part of inputs for proactive suggestions:
 - (1) The ability to donate app data, including structured data. For example, an email app can donate its emails, which are later used to retrieve a flight number to suggest.
 - (2) The ability to donate actions to be used in generating proactive suggestions. For example, a calendar app can donate the action "*view calendar*", which is later suggested when the user receives an event invitation.
 - (3) The ability to provide screen context, including by having their screen content captured. For example, the chat app's on-screen chat messages asking about an upcoming event can be used to suggest the location of that event.
 - (4) The ability to donate additional data while proactive suggestions are being generated. For example, a phone app can donate the currently dialled phone number (even if not shown on screen), which is used to determine that the user is calling an airline and therefore suggest the flight number.
 - (b) The ability to be part of outputs for proactive suggestions:
 - (1) The ability to surface (i.e. display) proactive suggestions, on any surface inside apps, outside apps, and on top of apps (e.g., context menu, overlay, notifications) available to Alphabet's first-party apps. For example, a phone app can show the flight number that is suggested (e.g. being relevant for the ongoing call).
 - (2) The ability to be the egress action of proactive suggestions, via any mechanism (e.g. through an overlay or by being directed to the app) available to Alphabet's first-party apps. For example, Magic Cue can suggest "search restaurants in Gemini", prompting the user to open Gemini. In this case, Gemini is the egress action. As another example, a calendar app can be opened or presented in an overlay as part of a suggestion to "View calendar".
 - (3) The ability to receive the egress data (i.e. output) of a proactive suggestion. For example, a chat app can receive the suggested text to copy it into a chat message.
- (33) Alphabet shall grant third parties access to additional functionalities available to Alphabet's services if necessary to enable effective interoperability with the proactive suggestions features referred to in paragraph (30).
- (34) To provide third parties with an interoperability solution for the proactive suggestions feature referred to in paragraph (30) that is equally effective as that

available to any of Alphabet's services, Alphabet shall implement the following measures:

- (a) Alphabet shall allow third-party apps to donate any type of app data or additional data (both in terms of the type of raw data or the type of insight derived from it) and any type of actions, in an equally effective manner (e.g. via equally effective donation channels) as Alphabet's apps.
- (b) Alphabet shall allow third-party apps to consent to or withdraw from donating app data, actions, screen context, and additional data for proactive suggestions and to becoming surfaces for proactive suggestions, in an equally effective and granular manner as Alphabet's apps.
- (c) Alphabet shall ensure that, when proactive suggestions are surfaced inside apps, the surfacing app is able to control the positioning of the proactive suggestion in relation to the surfacing app's own user interface.
- (d) Alphabet shall allow third-party apps to combine being the egress action and receiving the egress data for proactive suggestions. For example, a suggestion could result in opening an assistant in an overlay with a prepared prompt.
- (e) Alphabet shall allow that, when third-party apps are part of outputs in any of the ways described in Section 9.3.3.1 of the Preliminary Findings, they must be able to receive data/input from Alphabet's first-party apps, in an equivalent manner as Alphabet's first-party apps can.
- (f) Alphabet shall allow end users, for the purpose of receiving proactive suggestions, to consent with an equivalent level of granularity for the types of proactive suggestions experiences, as allowed with respect to the proactive suggestions experiences enjoyed by Alphabet's first-party apps.
- (g) Alphabet shall allow users, for the purpose of receiving proactive suggestions, to consent with an equivalent level of granularity and via an equivalent user interfaces and user journeys for the usage of data from specific apps and surfaces and on specific apps and surfaces, as allowed for Alphabet's first-party apps.
- (h) Alphabet shall allow users, whenever third-party apps are part of inputs or outputs of proactive suggestions, to interact with the proactive suggestions in an equivalent manner and via an equivalent user interfaces and user journeys as users can interact with the proactive suggestions where Alphabet's first-party apps are part of inputs or outputs.
- (i) Alphabet shall ensure that, when proactive suggestions are generated, third-party apps, whether they are part of inputs or outputs, are treated non-discriminatorily, in particular by intelligence components, and benefit from an equivalent experience of proactive suggestions as Alphabet's first-party apps.
- (j) Alphabet shall ensure that third-party apps are able to surface app data or actions from Alphabet's first-party apps, in an equivalent manner as Alphabet's first-party apps can surface suggestions based on app data or actions from other Alphabet first-party apps. This includes the ability to combine the screen context and additional data from the third-party app with the app data or action from the Alphabet first-party app.
- (k) Alphabet shall ensure that Alphabet's first-party apps can surface suggestions based on app data or actions from third-party apps, in an equivalent manner

as Alphabet's first-party apps can surface suggestions based on app data or actions from other Alphabet first-party apps. This includes the ability to combine the screen context and additional data from the Alphabet first-party app with the app data or action from the third-party app.

- (l) Alphabet shall allow multiple third-party apps must be able to be part of inputs and outputs, at any time, regardless of the number of other apps that are already part of inputs and outputs, simultaneously and concurrently, in an equivalent and non-discriminatory manner as Alphabet's first-party apps.
- (35) Alphabet may implement technical measures such that proactive suggestions are provided in line with principles of process isolation, encryption, anonymous and aggregated telemetry, and user control (e.g. requiring user interaction before data egress to a surfacing app), insofar as they are based on transparent, objective, precise, and non-discriminatory conditions that also apply to Alphabet's services and hardware.
- (36) Alphabet shall provide effective interoperability with any future updates or implementations, including new functionalities, of the proactive suggestions feature insofar and as soon as they are available to Alphabet's own services. For the avoidance of doubt, such future updates of the proactive suggestions feature include any updates concerning the:
 - (a) Ability to donate or surface any type of input needed for generating proactive suggestions, irrespective of whether such input is stored on a cloud service.
 - (b) Advanced assistant capabilities allowing assistants to surface, read, receive, and/or interact with the proactive suggestion.
 - (c) Ability to display multiple proactive suggestions concurrently in one surface.
 - (d) Ability for the user to interact with the proactive suggestion (e.g., provide feedback, edit the suggestion).
- (37) Alphabet shall implement the measures above in compliance with the measures for all features in Section 5 of this Annex.
- (38) Alphabet shall implement the measures for the proactive suggestions feature in the release of Android 17 QPR2 and, in any case by 1 January 2027 at the latest.

2.3. Measures for context-aware intelligence

- (39) Alphabet shall provide effective interoperability with the *context-aware intelligence* feature.
- (40) The context-aware intelligence feature is described in Section 9.4.2 of the Preliminary Findings. Context-aware intelligence allows to provide third-party versions of proactive suggestions and "intelligent experiences", i.e. experiences that are personalised to the user's context and can be continuously active to enhance the user experience. This relies, for example, on accessing inputs from device sensors and processing those inputs to enable proactive actions, such as live translation or song recognition. Currently, most context-aware intelligence experiences are implemented through Alphabet's Android System Intelligence component.
- (41) Alphabet shall implement an interoperability solution that provides third parties with access to the same Google Android feature (as described in the preceding

paragraph) and all its functionalities, as available to Alphabet's services and in a way that is equally effective as the solution available to Alphabet's services.

(42) These functionalities are:

(a) Access to inputs:

- (1) Access to the user's physical context, as used by or available to Alphabet's services. This includes ambient data, i.e. data from device sensors, which includes but is not limited to, data from the camera, microphone, accelerometer, proximity sensor, GPS, and any other location signals.
- (2) Access to the user's digital context, as used by or available to Alphabet's services. This includes, but is not limited to, device audio, screen contents and context, screenshots, data of the foreground app, notifications, app launches, contacts, shortcuts, SMS, and app package details.
- (3) The ability to retrieve data from an app, service, or the OS to the proactive suggestion or intelligent experience, which may include retrieving the data from the cloud.
- (4) Global access to apps' data and actions stored on-device, including data from Alphabet apps, OEM apps, and third-party apps, as donated by apps, including but not limited to, through an on-device database or data sharing platforms allowing for centralised data access and sharing, as used by or available to Alphabet's services.

(b) Access to processing resources:

- (1) Ability to use an on-device model to process the input data and generate the outputs, both those already present on the device and those selected and brought by the service, with the same level of model availability (in particular if already present), hardware access and background execution (in particular selected and brought by the service) as Alphabet's services.
- (2) Ability to use cloud-based models to process the input data and generate the outputs, and to use communication channels to receive and send such data.
- (3) Ability to access and write to dedicated and shared storage and databases.
- (4) Ability to access low-power processors and the runtime environments thereof, allowing to process data, such as ambient data, efficiently.
- (5) Ability to use protected execution environments allowing to process particularly sensitive data, such as ambient data, in isolation.

(c) Access to output channels and display surfaces:

- (1) The ability to discover the channels and surfaces that are available for proactive suggestions or intelligent experiences and be aware of newly available channels and surfaces.
- (2) The ability to decide on the most relevant channel or surface, including through accessing the properties of channels and surfaces (e.g. their size or location).

- (3) The ability to present proactive suggestions or intelligent experiences on a surface visible or audible to the user in the UI, mediated by the OS. Such surfaces can take many forms. For instance, such surfaces may appear inside Alphabet's apps and third parties' apps, as part of system interfaces (e.g. lock screen or notifications), as "chips" or small insets in keyboards, as a screen overlay, etc.
 - (4) The ability to transmit data from the proactive suggestion or intelligent experience to an app, service, or the OS, which may include transmitting the data to the cloud.
 - (5) The ability to trigger integrations and interactions with apps and the OS from the proactive suggestion or intelligent experience, such as redirecting to another app, opening an app in an overlay, or deep linking to parts of the OS such as settings.
- (43) Alphabet shall grant third parties access to additional functionalities available to Alphabet's services if necessary to enable effective interoperability with the context-aware intelligence feature referred to in paragraph (40).
- (44) To provide third parties with an interoperability solution for the context-aware intelligence feature referred to in paragraph (40) that is equally effective as that available to any of Alphabet's services, Alphabet shall implement the following measures:
- (a) Alphabet shall offer access to device sensors, outputs, and data as specified under paragraph (42)(a) of the Annex on a continuous basis and in the background, as are used by or available to Alphabet's services.
 - (b) Alphabet shall grant access to device sensors, outputs, and data as specified under paragraph (42)(a) of the Annex with the same degree of granularity, frequency, and recency, as are used by or available to Alphabet's services.
 - (c) Alphabet shall allow third parties to design and tailor the surfacing experience, when accessing display surfaces as specified under paragraph (42)(c) of the Annex, for instance in terms of including the branding, icon, or link to the apps the surfaced information has been sourced from.
 - (d) Alphabet shall offer access to inputs as specified under paragraph (42)(a) of the Annex, to processing resources as specified under paragraph (42)(b) of the Annex, and to output channels and display surfaces as specified under paragraph (42)(c) of the Annex on a non-discriminatory basis between Alphabet's and third-party services, in particular when access cannot be provided concurrently.
 - (e) Alphabet may implement technical measures requiring that, when accessing inputs as specified under paragraph (42)(a) of the Annex, processing resources as specified under paragraph (42)(b) of the Annex, and output channels and display surfaces as specified under paragraph (42)(c) of the Annex, third parties must do so in line with principles of process isolation, encryption, anonymous and aggregated telemetry, and user control, insofar as this is based on transparent, objective, precise, and non-discriminatory conditions that also apply to Alphabet's services and hardware.
- (45) Alphabet shall provide effective interoperability with any future updates, including new functionalities, of the context-aware intelligence feature insofar as they are

available to Alphabet's services. For the avoidance of doubt, such future updates of the context-aware intelligence feature include any updates concerning:

- (a) Additional inputs, processing resources, or output channels and display surfaces.
 - (b) Further integration with cloud-based AI services and AI assistants that contribute data to and retrieve data from the user's context.
- (46) Alphabet shall implement the measures above in compliance with the measures for all features in Section 5 of this Annex.
- (47) Alphabet shall implement the measures for the context-aware intelligence feature in the release of Android 17 QPR2 and, in any case by 1 January 2027 at the latest.

2.4. Measures for access to ambient data

- (48) Alphabet shall provide effective interoperability with the *access to ambient data* feature.
- (49) The access to ambient data feature is described in Section 9.5.2 of the Preliminary Findings. Access to ambient data provides access to the continuous stream of real-time inputs/outputs from a device's core sensors for example microphone, camera, screen, speakers. The access enables AI services to deliver personalised, context-aware experiences such as sound detection, live screen guidance, or object recognition by processing raw sensor data.
- (50) Alphabet shall implement an interoperability solution that provides third parties with access to the same Google Android feature (as described in the preceding paragraph) and all its functionalities, as available to Alphabet's services and in a way that is equally effective as the solution available to Alphabet's services.
- (51) These functionalities are the ability to continuously collect and process the same real-time inputs as well as outputs from a device's core sensors, as are available to Alphabet's services including in particular:
- (a) the microphone input,
 - (b) speaker output (system audio),
 - (c) camera,
 - (d) screen,
 - (e) location data,
 - (f) environmental sensors such as the accelerometer or proximity sensor.
- (52) Alphabet shall grant third parties access to additional functionalities available to Alphabet's services if necessary to enable effective interoperability with the access to ambient data feature as referred to in paragraph (49).
- (53) To provide third parties with an interoperability solution for the access to ambient data referred to in paragraph (49) that is equally effective as that available to any of Alphabet's services, the following measures apply.
- (a) Alphabet shall provide access to the same data under equally effective conditions as available to Alphabet's services, including in terms of quality, frequency, latency, and availability in the background.
 - (b) Alphabet shall provide access under equivalent consent flows as for Alphabet's services, including in terms of frequency.

- (c) Alphabet shall allow third-party apps to access ambient data related to Alphabet's applications (such as screen content), whenever such functionality is enabled for Alphabet's services, such as Gemini.
 - (d) Alphabet may implement technical measures requiring that, when accessing ambient data, third parties must do so in line with principles of process isolation, encryption, anonymous and aggregated telemetry, and user control, insofar as this is based on transparent, objective, precise, and non-discriminatory conditions which also apply to Alphabet's services and hardware.
- (54) Alphabet shall also provide effective interoperability with any future updates, including new functionalities, of the access to ambient data insofar as they are available to Alphabet's own services or hardware.
 - (55) Alphabet shall implement the measures above in compliance with the measures for all features in Section 5 of this Annex.
 - (56) Alphabet shall implement the measures for access to ambient data feature by 1 January 2027.

3. MEASURES ON FEATURES FOR ACTIONS ON APPS AND THE OS

3.1. Measures for structured on-device integration

- (57) Alphabet shall provide effective interoperability with the *structured on-device integration* feature.
- (58) The structured on-device integration feature is described in Section 10.2.2 of the Preliminary Findings. This feature enables an AI service to take actions inside other apps on the device in response to a user's request and is currently implemented through on-device integration channels, such as App Actions and App Functions.
- (59) Alphabet shall implement an interoperability solution that provides third parties with access to the same Google Android feature (as described in the preceding paragraph) and all its functionalities, as available to Alphabet's services and in a way that is equally effective as the solution available to Alphabet's services.
- (60) These functionalities are:
 - (a) The ability for apps to discover all available structured on-device integrations, including metadata necessary to determine which actions to trigger.
 - (b) The ability for apps to execute structured on-device integrations.
 - (c) The ability to let the user be aware of and observe that a structured on-device integration is being executed.
 - (d) The ability to execute the structured on-device integration within the app, i.e. without app switching.
- (61) Alphabet shall grant third parties access to additional functionalities available to Alphabet's services, if necessary to enable effective interoperability with the structured on-device integration feature referred to in paragraph (58).
- (62) The interoperability solution for the structured on-device integration feature referred to in paragraph (58) must be equally effective as that available to any of

Alphabet's own services. To that end, Alphabet shall implement the following measures:

- (a) Alphabet shall make available all channels of structured on-device integration to third parties, including App Actions and App Functions, in a manner that is equally effective as that available to any of Alphabet's own services.
 - (b) Alphabet shall enable third-party apps to discover all structured on-device integrations exposed by apps that are installed on the user's device, including a mechanism that ensures third-party apps can retrieve all metadata necessary to determine which actions to trigger.
 - (c) Alphabet shall enable third-party apps to execute all structured on-device integrations exposed by other apps, including apps of other third-party developers, OEM apps, system apps, and Alphabet apps.
 - (d) Alphabet shall make public the documentation necessary for third parties to access, integrate and execute structured on-device integration. This requirement captures documentation regarding how a third party can execute an App Action using the information provided through the new discovery mechanism described in paragraph (62)(a).
- (63) Alphabet shall also provide effective interoperability with any future updates, including new functionalities, of the structured on-device integration feature insofar as they are available to Alphabet's own services or hardware.
- (64) Alphabet shall implement the measures above in compliance with the measures for all features in Section 5 of this Annex.
- (65) Alphabet shall implement the measures for the structured on-device integration feature by 1 January 2027.

3.2. Measures for screen automation

- (66) Alphabet shall provide effective interoperability with the *screen automation* feature.
- (67) The agent-controlled interactions feature is described in Section 10.3.2 of the Preliminary Findings. The feature enables Alphabet's services, including Gemini, to take control of other apps on the Android mobile device to automate multi-step tasks on behalf of the user. Gemini can imitate user behaviour and observe the controlled app's screen to automate the user interface and to perform most tasks a user would be able to do. Gemini will run the controlled app in a separate virtual window, allowing the assistant to complete the task in the background, while the user can do something else.
- (68) Alphabet shall implement an interoperability solution that provides third parties with access to the same Google Android feature (as described in the preceding paragraph) and all its functionalities, as available to Alphabet's services and in a way that is equally effective as the solution available to Alphabet's services.
- (69) These functionalities are:
- (a) Take control over apps. Ability for an app (controlling app) to take over control of other apps (controlled apps) by imitating user interactions, such as taps, clicks, and typing.

- (b) Access to on screen content. Ability for an AI service to access the screen content of the controlled app to understand its UI, allowing the AI service to control the app.
 - (c) Discovery of apps. Ability for an AI service to determine which apps are installed on the device and retrieve metadata that may be relevant to decide which app to automate.
 - (d) Control apps in background. Ability for an AI service to fully execute control of other apps in background. The process of controlling apps is not visible to the user by default and can be fully executed in background.
 - (e) User observation. Ability for the user to surface the controlled app and observe the AI service controlling it. In addition, the user can stop the AI service from controlling or take over control from the AI service.
 - (f) Seamless transition. Ability to transition seamless between background and foreground when the user chooses to observe the controlled app or the AI service, surfaces the app to the user for required input, the transition between background and foreground is seamless.
 - (g) OS indicators. Ability for OS indicators to demonstrate to the user that an AI service is currently controlling another app.
 - (h) Concurrency management. Ability for the Google Android OS to manage concurrent access, including scheduling or preventing multiple accesses. This can include user-facing mechanisms, such as greying out the app icon of the controlled app so that the user cannot use the same app at the same time as the controlling app.
 - (i) Blocking sensitive areas. Ability for controlled apps to block sensitive views from access of the controlling app. This functionality allows apps to protect sensitive user data, critical settings, or purchase decisions from being automated by the controlling app.
- (70) Alphabet shall grant third parties access to additional functionalities available to Alphabet's services if necessary to enable effective interoperability with the screen automation feature referred to in paragraph (67).
- (71) The interoperability solution for the screen automation feature referred to in paragraph (67) must be equally effective as that available to any of Alphabet's own services. To that end, Alphabet shall implement the following measures.
- (a) Alphabet shall enable third-party apps to take control of other apps, including apps of other third-party developers, OEM apps, system apps, and Alphabet's apps. Developers must be allowed to perform interactions with the apps by imitating user behaviour.
 - (b) Alphabet shall enable third-party apps to discover all apps that are installed on the user's device. Alphabet shall ensure that third-party apps can retrieve metadata that is relevant to decide which app to automate, including but not limited to app name, the package name, and the app developer's name.
 - (c) Alphabet shall enable third-party apps to imitate the same user interactions that are available to Alphabet. This includes, but is not limited to, the interactions available to Computer Control API.
 - (d) Alphabet shall enable third-party apps to access the screen content of the controlled app.

- (e) Alphabet shall enable third-party apps to control other apps using imitated user interactions in the background on a virtual display. The controlling app can complete the task, while the user can use their phone otherwise.
 - (f) Alphabet shall enable third-party apps to mirror or otherwise present the ongoing interactions of their app to the user. Alphabet shall enable the user to observe and intervene if the controlling app is misbehaving or not able to achieve the ask.
 - (g) Alphabet shall ensure that the transition between the observation of the controlled app and controlling the app in background should be seamless.
 - (h) Alphabet shall enable multiple apps to have the capability to take control over other apps. Alphabet shall not limit this capability to one single app such as a default app.
- (72) Alphabet shall also provide effective interoperability with any future updates, including new functionalities, of the screen automation feature insofar as they are available to Alphabet's own services or hardware.
- (73) Alphabet shall implement the measures above in compliance with the measures for all features in Section 5 of this Annex.
- (74) Alphabet shall implement the measures for the screen automation feature by 1 January 2027.

3.3. Measures for integration with first-party services

- (75) Alphabet shall provide effective interoperability with the *read and write access* feature *for integration with first-party services*.
- (76) The read and write access features are described in Section 10.4.2 of the Preliminary Findings. The read access feature allows a service to retrieve information from another service, which can be displayed to the user as-is or after processing. The write access feature allows a service to perform operations in another service that modifies the data held in that service and execute actions with it.
- (77) Alphabet shall implement an interoperability solution that provides third parties with access to the same Google Android feature (as described in the preceding paragraph) and all its functionalities, as available to Alphabet's services and in a way that is equally effective as the solution available to Alphabet's services.
- (78) These functionalities are:
- (a) Semantic search. The read access feature can be used to search for and retrieve information in a way that understands, on one hand, the user's intent and, on the other hand, the contextual meaning of terms and concepts in the dataset where the information is searched. For example, a user can ask an AI assistant "When is my trip to Spain?" and the AI assistant will be able to find the calendar event with title "Flight to Barcelona", which contains neither "trip" nor "Spain".
 - (b) Creating and editing user data. The most prominent functionality of the write access feature is the ability to create and edit user data. For example, Gemini assistant can create and edit notes in Google Keep.

- (c) Sending data to other users. The write access feature can be used to allow the user send or share content with other users. For example, Gemini assistant can send emails and SMS/RCS messages on behalf of the user.
 - (d) Changing state and settings. The read and write access can be used to control another service and to change its settings. For example, users can invoke Gemini assistant when watching a YouTube video to pause or play the video. They can also invoke Gemini assistant while navigating in Google Maps to retrieve information about the estimated time of arrival without looking at the screen (read access) or modify the final destination (write access).
 - (e) Interactive access. Both read access and write access can happen through an interactive mechanism, such as interactive interfaces. For example, the user can ask Gemini assistant to find Italian restaurants nearby. Gemini assistant will then use the read access feature to retrieve the information from the Google Maps online services and provide the user with an “interactive card” that provides an interactive map and list of restaurants.
- (79) Alphabet shall grant third parties access to additional functionalities available to Alphabet’s services if necessary to enable effective interoperability with the read and write access features referred to in paragraph (76).
- (80) To provide third parties with an interoperability solution for the read and write access features referred to in paragraph (76) that is equally effective as that available to any of Alphabet’s own AI assistants, Alphabet shall implement the following measures:
- (a) Read access: 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.
 - (b) Write access: Alphabet shall allow third parties to perform operations that modify the data held in services that are provided together with, or in support of, Google Android and execute actions within such services – in a way that is equally effective to what is available to Alphabet’s services, such as Gemini assistant.
 - (c) For all measures above, Alphabet may require user consent, compatibly with the measures specified in Section 5 of this Annex, and particularly in Section 5.2.
 - (d) In cases where it is not technically feasible to provide access immediately, Alphabet shall provide access without undue delay, following a transparent and non-discriminatory process that includes clear deadlines for Alphabet and third parties.
- (81) All the measures above should apply regardless of whether the relevant features are part of Google Android, and in particular regardless of whether they rely on server-based integrations.
- (82) Alphabet shall provide effective interoperability with any future updates, including new functionalities, of the read and write access features insofar as they are available to Alphabet’s own AI assistant services.
- (83) Alphabet shall implement the measures above in compliance with the measures for all features in Section 5 of this Annex.

- (84) Alphabet shall implement the measures for the read access functionality by 1 January 2027, and the measures for the write functionality by 1 June 2027.

3.4. Measures for system integration

- (85) Alphabet shall provide effective interoperability with the *system integration* feature.
- (86) The system integration feature is described in Section 10.5.2 of the Preliminary Findings. The feature enables Alphabet's services, including Gemini, to interact and integrate with the operating system on behalf of the user. The user may request the assistant amongst other things to change a specific setting, control media playback, or open an app.
- (87) Alphabet shall implement an interoperability solution that provides third parties with access to the same Google Android feature (as described in the preceding paragraph) and all its functionalities, as available to Alphabet's services and in a way that is equally effective as the solution available to Alphabet's services.
- (88) These functionalities are:
- (a) Ability to interact with OS settings. These settings include turning on/off Bluetooth, turning on/off battery saver mode, turning on/off do not disturb, reading the current battery level.
 - (b) Ability to interact with ongoing activities on the Google Android mobile device. These activities include controlling playing media and volume.
 - (c) Ability to interact with Google Android OS-level functionalities. These functionalities include restarting the device, turning the device off, taking screenshots, and integrate with notifications.
 - (d) Ability to integrate with common functionalities of Google Android. These functionalities include setting timers, managing alarms, integrate with text messaging functionalities.
- (89) Alphabet shall grant third parties access to additional functionalities available to Alphabet's services if necessary to enable effective interoperability with the system integration feature referred to in paragraph (86).
- (90) Alphabet shall implement an interoperability solution that provides third parties with access to the same Google Android feature as available to Alphabet (as described in paragraph (86) in a way that is equally effective as the solution available to Alphabet. To that end, Alphabet shall implement the following measures:
- (a) Alphabet shall implement interoperability for system integration through App Functions that contain all functionalities for system integration.
 - (b) Alphabet shall not restrict third-party apps to access the App Functions for the system integration feature.
 - (c) Alphabet shall enable third-party apps to discover all apps that are installed on the user's device, to understand user queries that demand to open specific apps. Alphabet shall make available metadata on installed apps to third-party apps, including but not limited to the app name, the package name, and the app developer's name.
 - (d) Alphabet shall ensure that third-party apps have the same capabilities for system integration that are available to Alphabet's services. Alphabet shall

ensure that third-party apps are not subject to access restrictions that require preinstallation, privileged permissions, or any other OEM controlled access restriction.

- (e) Alphabet shall ensure that all system integrations, including the changing of settings, must be directly done inside the third-party app, rather than requiring switching the user to the settings app or another app. This applies to all capabilities that are available to Alphabet's services for direct change.
- (91) Alphabet shall provide effective interoperability with any future updates, including new functionalities, of the system integration feature insofar as they are available to Alphabet's own services or hardware.
- (92) Alphabet shall implement the measures above in compliance with the measures for all features in Section 5 of this Annex.
- (93) Alphabet shall implement the measures for the system integration feature by 1 January 2027 at the latest.

4. MEASURES ON FEATURES FOR ACCESS TO RESOURCES

4.1. Measures for system-level on-device models

- (94) Alphabet shall provide effective interoperability with the *system-level on-device models (ODM)* feature.
- (95) The system-level ODM feature is described in Section 11.2.2 of the Preliminary Findings. The feature allows Alphabet to discover, access, use and customise all on-devices models that are part of Google Android (including Google Android middleware) and that are made accessible to Alphabet's services outside the operating system.
- (96) Alphabet shall implement an interoperability solution that provides third parties with access to the same Google Android feature (as described in the preceding paragraph) and all its functionalities, as available to Alphabet's services and in a way that is equally effective as the solution available to Alphabet's services.
- (97) These functionalities are:
 - (a) Discovery of all system-level ODMs. The ability to discover system system-level ODMs.
 - (b) Access to all system-level ODMs. The ability to all access and use system-level ODMs and all their functionalities, including specialized APIs for specific tasks.
 - (c) Ability to customise functionality. The ability to customise the system-level ODMs, its functionalities, and output, including by low rank adaption.
- (98) Alphabet shall grant third parties access to additional functionalities available to Alphabet's services if necessary to enable effective interoperability with the system-level ODM feature referred to in paragraph (95).
- (99) To provide third parties with an interoperability solution for the system-level ODM feature referred to in paragraph (95) hat is equally effective as that available to any of Alphabet's services, Alphabet shall implement the following measures:
 - (a) Access subject to the same restrictions. Alphabet shall allow third-party services or hardware to use the system-level ODMs without restrictions

which do not equally apply to Alphabet's services and hardware using the system-level ODM feature.

- (b) Ability to use functionality for all use cases. Alphabet shall allow third-party services or hardware to use the system-level ODMs for all use cases, including those that Alphabet supports at the system level.
 - (c) Access conditions. Alphabet shall ensure access to the system-level ODM feature for third-party services and hardware according to transparent, objective, precise, and non-discriminatory rules that also apply to Alphabet's services, including for use cases that Alphabet does not offer.
- (100) Alphabet shall provide effective interoperability with any future updates, including new functionalities, of the system-level ODM feature insofar as they are available to Alphabet's own services or hardware. For the avoidance of doubt, such future updates of the system-level ODM feature include any updates concerning the:
- (a) The inclusion of new system-level on-device models, such as those related to image or video creation.
 - (b) Updates to existing system-level ODMs such as the Gemini Nano on-device models.
- (101) Alphabet should implement the measures above in compliance with the measures for all features in Section 5 of this Annex.
- (102) Alphabet should implement the measures for the system-level ODM feature by 1 January 2027.

4.2. Measures for on-device model implementation

- (103) Alphabet shall provide effective interoperability with the *on-device model (ODM) implementation* feature.
- (104) The ODM implementation feature is described in Section 11.3.2 of the Preliminary Findings. The feature allows Alphabet to install, run and use on-device models and software to implement such models on Google Android mobile devices. The feature relies on access to hardware resources such as the CPU, GPU, NPU and RAM (including RAM residency), background execution capabilities, and the ability to make functionality of Alphabet's ODMs available to all or some apps running on the device to enable broad adoption.
- (105) Alphabet shall implement an interoperability solution that provides third parties with access to the same Google Android feature (as described in the preceding paragraph) and all its functionalities, as available to Alphabet's services and in a way that is equally effective as the solution available to Alphabet's services, including its Gemini Nano models, machine learning models, and AI Core, as well as any future Alphabet ODMs or software to implement such models.
- (106) These functionalities are:
- (a) Operation of on-device models. The ability to install, run and use on-device models and software to implement such models.
 - (b) Hardware resources. The ability to timely and reliably access to hardware resources that are necessary to run on-device models or software to implement such models. This includes access to specialised chips to run on-device model tasks, such as the TPU/APU/NPU, access to the Graphics

Processing Unit (GPU), the CPU, and access to RAM (incl. RAM residency) and other memory.

- (c) Centralised availability. The ability for an app to centrally expose on-device models and software that implements such models to another app, and the ability for the other app to use such models and software. This includes the ability to exchange information between the service implementing the on-device model and the service making use of the on-device model.
 - (d) Background execution. The ability to provide and use on-device models or software to implement such models in the background, i.e. when the app hosting the model and the app using the model are not in the foreground.
- (107) Alphabet shall grant third parties access to additional functionalities available to Alphabet's services if necessary to enable effective interoperability with the ODM implementation feature described in paragraph (104).
- (108) To provide third parties with an interoperability solution for the on-device model implementation feature that is equally effective as that available to Alphabet's services, Alphabet should implement the following measures:
- (a) Alphabet shall allocate hardware resources to its own on-device models and software to implement such models and third-party on-device models and software to implement such models according to transparent, objective, precise, and non-discriminatory rules. Alphabet may do this by, for example, implementing an appropriate system-level scheduler that comply with these requirements.
 - (b) Regardless of the rules referred to under paragraph (108)(a), if Alphabet's own on-device models and software to implement such models are granted preferential access to hardware resources, it shall allow the user to remove such preferential access from Alphabet's on-device models and software to implement such models and grant it to third-party on-device models and software to implement such models. Such preferential access may consist in Alphabet's own on-device models or software to implement such models being granted access to memory resources which are not equally accessible by third-party on-device models or software to implement such models.
 - (c) Alphabet shall apply any restrictions, time windows, and resource limitations (e.g. on CPU, GPU or NPU/APU/NPU execution) on background execution according to transparent, objective, precise, and non-discriminatory rules that also apply to Alphabet's on-device models and software to implement such models, including for use cases that Alphabet does not offer.
 - (d) Any limitation or choice on the background execution capabilities of, or access to hardware resources for third-party on-device models or software to implement such models as a result of a user action shall only be permissible if the user can take the same action with the same limiting effect regarding Alphabet's on-device models and software to implement such models.
 - (e) If Alphabet allows OEMs to customise or otherwise changes the level of access to hardware resources or background execution for on-device models or software to implement such models, Alphabet shall ensure that such customisation complies with all the measures specified in this Annex. In particular, any such customisations or changes granted to Alphabet's on-device models or software to implement such models must also be granted to third-party on-device models or software to implement such models.

- (f) Alphabet shall allow third-party apps to use the same system-privileged APIs that Alphabet's AICore uses, or APIs that provide equivalent functionalities.
 - (g) If Alphabet allows any third party, including OEMs, to provide on-device models via AICore, it shall allow other third parties to also provide on-device models via AICore.
- (109) Alphabet should provide effective interoperability with any future updates, including new functionalities, of the on-device model feature insofar as they are available to Alphabet's own services or hardware.
- (110) Alphabet shall implement the measures above in compliance with the measures for all features in Section 5 of this Annex.
- (111) Alphabet shall implement or ensure the implementation of the measures for the ODM implementation feature by 1 January 2027.

4.3. Measures for background execution

- (112) Alphabet shall provide effective interoperability with the *background execution* feature.
- (113) The background execution feature is described in Section 11.4.2 of the Preliminary Findings. The feature allows to execute tasks in a timely manner, including when the app or service is not in the foreground. This ability relies on the OS providing adequate interfaces to start and resume tasks, and allocating sufficient system resources (including CPU, GPU, NPU, and RAM) to the app or service.
- (114) Alphabet shall implement an interoperability solution that provides third parties with access to the same Google Android feature (as described in the preceding paragraph) and all its functionalities, as available to Alphabet's services and in a way that is equally effective as the solution available to Alphabet's services.
- (115) These functionalities are:
- (a) Access to hardware resources. This consists in the ability to access hardware resources – including CPU, GPU, NPU, and RAM – to execute tasks. Access to the same background execution capabilities entails, in particular, being exposed and subject to the same limitations of these capabilities as a result of a user action. This includes the effect of an end user terminating an Android app in the app switching menu (“force-quitting” or “force-killing”).
 - (b) Requesting resource allocation. This consists in the ability to indicate to the OS that the service requests allocation of a certain amount of resources, e.g. for a certain period of time or while a certain process is active. This can be done through pre-defined OS interfaces (APIs) that guarantee a certain degree of access to resources, typically under certain conditions that are tied to specific use cases. For example, such an API could guarantee that an AI service can synchronise data in the background for at least 10 seconds without showing a visible notification for the user, or for longer if a notification is shown. It could also guarantee that a connected device (e.g. earbuds) maintains a persistent connection with the Google Android mobile device that allows the user to invoke an app (e.g. an AI assistant) using a hotword detected on the connected device. While such interfaces can be generally useful for specific use cases, they typically do not enable every use case, for which a more generic way of requesting resource allocation is needed.

- (c) Starting or resuming a task while the app is in the background. This can depend on external conditions (e.g. an instruction received by another app; time-based conditions; a user approaching a certain location, such as their home; the user taking a picture) or internal conditions (e.g. the app concluding a task, such as the summarisation of a long audio recording). The task can be executed by the app itself (e.g. an AI assistant displaying a reminder notification at a given time) or by another app (e.g. an AI assistant sending a message via another messaging app at a given time).
 - (d) Device configurations. This consists in the ability of an app or service to be treated preferably by the OS, directly or indirectly, in terms of access to resources because of configuration settings. This includes configuration settings that allow an app to be persistently stored in memory or that allow an app to be exempted from power management mechanisms. These settings can be defined by Alphabet and/or the OEM but can also be configurable by the end user. This functionality therefore includes the ability for the user to make choices on the level of background execution to a service, including regarding time, place, and cadence, and the presentation and effect of any prompt or user interface for that purpose.
- (116) Alphabet shall grant third parties access to additional functionalities available to Alphabet's services if necessary to enable effective interoperability with the background execution feature referred to in paragraph (113).
- (117) To provide third parties with an interoperability solution for the background execution feature referred to in paragraph (113) that is equally effective as that available to any of Alphabet's own AI assistants, Alphabet shall implement the following measures:
- (a) Alphabet shall ensure that third-party apps have the same access to background execution as Alphabet's services (including Alphabet's apps) have access to, directly or indirectly. To this end, Alphabet should define transparent, objective, precise and non-discriminatory rules that also apply to Alphabet's services, including for use cases that Alphabet does not offer. The same rules should apply to non-Alphabet services and Alphabet services, regardless of whether the apps providing the relevant services are pre-installed. Such rules may not treat any specific service differently, for example by restricting or exempting a specific service, unless the user has provided consent for the specific service (e.g. by exempting the corresponding app).
 - (b) Alphabet shall ensure that third parties can offer always-on hotword detection on connected devices to invoke an app or perform other operations on the Google Android mobile device, using the same access to background execution that is available to Alphabet for the same purpose. Alphabet should publish developer documentation explaining how third parties can use the relevant Google Android APIs to implement this use case.
 - (c) If Alphabet allows OEMs to customise background execution rules, Alphabet shall ensure that such rules comply with all the measures specified in this Annex. In particular, such rules should be transparent, objective, precise and non-discriminatory rules that also apply to pre-installed services.
 - (d) Alphabet shall ensure that third-party apps may prompt the user to grant it access to the same level of background execution that other services –

including Alphabet's services – have access to. In particular, granting this access should provide the app the same level of background execution that Alphabet requires OEMs to grant to its own apps (also with respect to proprietary OEM restrictions) or similar requirements. The user should be able to grant access with a single in-app step, without having to navigate to settings or back to the app.

- (e) If Alphabet allows any limitation or choice on the background execution capabilities of user-downloaded apps as a result of a user action, Alphabet shall ensure that the user can take the same action with the same limiting effect regarding other apps, including pre-installed services. This includes the action of a user terminating an app in the app switching menu (“force-quitting”).
- (118) Alphabet shall provide effective interoperability with any future updates, including new functionalities, of the background execution feature insofar as they are available to Alphabet's own services or hardware.
- (119) Alphabet shall implement the measures above in compliance with the measures for all features in Section 5 of this Annex.
- (120) Alphabet shall implement the measures in paragraph (117)(d) on user prompts for the background execution feature by 1 January 2027. Alphabet should implement the rest of the measures laid down in paragraph (117) by 1 June 2027.

5. MEASURES FOR ALL FEATURES

- (121) The following measures shall apply in relation to all features referred to in Sections 1-4 of this Annex.

5.1. Implementation across the Google Android ecosystem

- (122) Alphabet shall ensure, including by technical and contractual means, that the measures in this Annex (including in Section 5) are complied with on all Google Android mobile devices,⁽⁵⁾ including on devices supplied by third-party OEMs. Specifically, Alphabet shall ensure that, for each feature in this Annex, the measures in this Annex are implemented on all Google Android mobile devices that meet the following conditions:
 - (a) The relevant feature exists on the device, and
 - (b) The device still receives Android AOSP, Google Play system, or Google system services updates.
- (123) Alphabet shall implement all interoperability solutions in a way that avoids ecosystem fragmentation, as Alphabet does for other Android SDK APIs. In particular, Alphabet shall ensure that third-party apps can use the interoperability solutions consistently via the same APIs across Google Android versions, devices, and form factors. This requirement applies regardless of how the solution is implemented on the specific Google Android mobile device or Android version. Alphabet may achieve this, for example, by defining Android SDK APIs and using

⁽⁵⁾ Google Android mobile device in this Annex includes all mobile devices that Alphabet certified as a certified Android device or a Google Play supported device.

Android support libraries, such as Android Jetpack, to define a consistent API across different Android versions and potentially different implementations.

- (124) If Alphabet allows OEMs to apply any customisations that affect the features in Sections 1-4 of this Annex or their implementation, Alphabet shall ensure that the custom implementation complies with all the measures specified in this Annex.

5.2. User consent

- (125) Alphabet shall not be prevented from implementing technical measures (e.g. system-level prompts) to ensure that features are accessible only with user consent.
- (126) Any requirements for feature access, such as user consent and the use of privacy indicators, shall be based on transparent, objective, precise, and non-discriminatory conditions that also apply to Alphabet's services and hardware. In particular, such requirements shall apply equally to user-installed apps and pre-installed apps. They should also comply with the other measures in this Annex, particularly the measures in Section 5.5 on equal effectiveness.

5.3. Integrity

- (127) Alphabet may take strictly necessary and proportionate measures to ensure that interoperability does not compromise the integrity of the operating system, hardware and software features.
- (128) Any integrity measure shall be duly justified and based on transparent, objective, precise, and non-discriminatory conditions that also apply to Alphabet's services and hardware. Alphabet can only impose conditions and take integrity measures that reflect a genuine integrity risk and do so in a consistent and systemic manner. Similarly, Alphabet should only apply such conditions with which compliance is capable of being independently verified and not exclusively within the gatekeeper's control. An integrity measure cannot be considered strictly necessary and proportionate if it seeks to achieve a higher integrity standard than the one that Alphabet requires or accepts in relation to its own services or hardware. Alphabet shall not impose a higher level of integrity on third parties than it applies to itself.
- (129) Any integrity measure shall be implemented in a way that does not undermine effective compliance with these measures and Alphabet's obligations with Regulation 2022/1925 including for instance by subverting end users' or third parties' autonomy, decision-making, or free choice via the structure, design, function or manner of operation of a user interface or apart thereof.
- (130) Alphabet shall ensure that the following conditions apply to any integrity measures:
- (a) The measures must be consistent with the approach that Alphabet already applies in similar or comparable cases.
 - (b) The measures shall apply equally to the Alphabet's services and to third parties' services. Moreover, measures must apply equally to pre-installed and user-installed apps.
 - (c) The measures shall be based on objective and verifiable evidence showing the existence and magnitude of the integrity risk and showing that the measures will be effective in reducing such risk. Alphabet shall retain such evidence.

- (d) The measures that concern informed consent and agency for the user, such as prompts, shall be supported by the necessary technical measures to appropriately inform the user or ask their consent.
 - (e) The measures shall not impose limits on the purpose, beneficiaries, apps, technologies used or use cases for feature access.
 - (f) The measures must not impose any commercial or financial requirements on the beneficiaries.
 - (g) The measures must not impose functional requirements beyond those strictly necessary to preserve integrity. In particular, they must not assume or require that a certain technology is used.
 - (h) The measures must be enacted only for as long as is necessary, and be adapted to technological evolution, including changes to the operating system.
 - (i) The measures shall be grounded in established industry standards and practices.
- (131) Alphabet shall inform the Commission in writing of any integrity measure that Alphabet intends to take, providing a justification of their strict necessity and proportionality at least 4 weeks in advance of their implementation, or without undue delay in case of urgency, unless the measure meets each of the following cumulative conditions:
- (a) it is not user-facing,
 - (b) it is exclusively of a technical nature,
 - (c) it is implemented for Alphabet and third parties in precisely the same way, and
 - (d) Alphabet has determined that the change will have no or only insignificant impact of any nature on third parties, including technical or commercial impact. Alphabet shall retain written documentation on how any such determination was made.

5.4. Eligibility of beneficiaries, applications and use cases

- (132) Alphabet shall make available the interoperability solutions and measures implemented in compliance with these measures to all providers of services and providers of hardware without undue delay, to the extent they indicate, including through the use of APIs, an interest in making use of any or all of the features listed in Sections 1 to 4 of this Annex, including for use cases that Alphabet does not offer.
- (133) Alphabet shall make the features in Sections 1-4 of this Annex, and interoperability solutions to implement these features pursuant to the measures in this Annex, accessible to all apps, including user-installed apps and pre-installed apps without system privileges. ⁽⁶⁾

⁽⁶⁾ User-installed apps are apps that are only installed following a decision by the user. Apps that are pre-installed on the Google Android mobile device without the user making such decision (pre-installed apps) are not user-installed apps.

- (134) Alphabet shall not impose any restrictions as to the beneficiaries; any verification or registration process shall, at a minimum, be strictly necessary and proportionate for the specific feature, apply only on the Google Play Store, be conducted by neutral and independent third parties, be based on transparent and non-discriminatory conditions, apply equally to all developers – including Alphabet – and irrespective of pre-installation, and have an independent dispute resolution mechanism.
- (135) Alphabet shall not impose any restrictions on the type or use case of the software application that can access or make use of the features listed in Sections 1 to 4 of this Annex.
- (136) Alphabet shall not impose any undue restrictions, including by requiring third parties to use other Alphabet products or services unless required for the functioning of the feature, or by requesting third parties to make choices in situations where such choice is not justified (for example, choosing between using the interoperability solution and continuing to use the same application ID), require third parties to use other Alphabet products or services in order to access and use the feature unless required for the functioning of the feature, or prevent third parties from benefitting from access to other features, including using the feature in combination with other features within the scope of Article 6(7) of Regulation 2022/1925.
- (137) Alphabet shall not undermine effective interoperability with the features listed in Sections 1 to 4 of this Annex by behaviour of a technical, commercial, contractual or any other nature. In particular, Alphabet shall:
- (a) enable third parties to make use of the interoperability solution in their existing apps via an automatic update of such apps, without being in any way disadvantaged for doing so, such as by having to reacquire or migrate existing users to a separate app; and
 - (b) not degrade, remove, disable, or otherwise make ineffective the interoperability solution, or prevent or impede updates, including security updates, for the end user as long as the end user is eligible to benefit from the functionalities allowed by these interoperability solutions; before taking any such measure in this respect, Alphabet shall notify the end user explaining how the measures Alphabet intends to take will affect the interoperability solution relied on by third-party services on the end user's Google Android mobile device.

5.5. Equal effectiveness

- (138) Alphabet shall ensure that any interoperability solution implemented for the features listed in Sections 1 to 4 of this Annex is equally effective to the solution available to Alphabet's services and hardware (including Alphabet's connected physical devices). Alphabet shall apply such equal effectiveness across all dimensions, including, but not limited to, the end user journey, ease of use for end users, device and software setup, data transmission speed, and energy consumption.
- (139) Alphabet shall make all interoperability solutions accessible in a manner that is technically sound, stable, and workable in practice for third parties without unnecessary hurdles.
- (140) Alphabet shall design all interoperability solutions enabling the concurrent use of the features by all apps in a non-discriminatory way. Alphabet shall not subject

access to features to the app holding a default role, including the default assistant role.

- (141) To ensure access is equally effective to the solution available to Alphabet's own services and hardware in terms of end user journey, Alphabet shall not decrease the ease, convenience and speed of using third-party services and hardware from the end user perspective. In particular, Alphabet shall refrain from adding friction by:
- (a) Offering choices to, or requesting permission from, the end user in a non-neutral or leading manner, including by using design patterns, dark patterns, or misrepresenting or exaggerating any risks of using the third-party connected physical device or granting a permission.
 - (b) Preventing the third party from explaining to end users in their own language the relevance of any system prompts shown, immediately before the prompt is shown or within the prompt.
 - (c) Showing unnecessary recurring prompts or notifications that the end user cannot easily and permanently disable in the same prompt or notification.
 - (d) Preventing the third party from triggering a permission prompt again in the future, unless the end user has so decided.
 - (e) Preventing third parties from obtaining user consent for any required permissions (i) from inside the application itself or with a simple system prompt or (ii) by indicating the user to follow a link directly pointing to and highlighting the relevant item in the system settings (so-called "deep-linking").
 - (f) Requiring end users to process multiple successive permission prompts that could be presented in a single prompt.
- (142) In particular, Alphabet shall apply the same end user journey and ease of use for end users in order to obtain the necessary consent for its own and third-party services and hardware, including:
- (a) showing the same user consent prompts (in terms of, inter alia, number, content, format and design) as Alphabet,
 - (b) showing the same information screens (in terms of, inter alia, number, content, format and design) as shown to users by Alphabet,
 - (c) limit the necessary user engagement to the same level as adopted by Alphabet, including the number of prompts and information screens.
- (143) Alphabet shall base any quantitative limitations (e.g., quotas and rate limits) on third-party access to a feature on transparent, objective, precise, and non-discriminatory conditions that also apply to Alphabet's services and hardware, including for use cases that Alphabet does not offer.
- (144) Alphabet shall enable third-party apps to centrally expose and access their functionalities to other apps, in a way that is equally effective to what is available to Alphabet's services and hardware. ⁽⁷⁾ Alphabet shall also enable third-party apps to dynamically provide and retrieve assets – including executable modules – at runtime, including from other apps, in a way that is equally effective to what is

⁽⁷⁾ For example, Google Play Services APIs are exposed centrally to other apps.

available to Alphabet's services and hardware. ⁽⁸⁾ If the app providing the assets is not installed when another app requests those assets, Alphabet shall enable the user to install the app via a low-friction in-app mechanism, in a way that is equally effective to what is available to Alphabet's services and hardware. ⁽⁹⁾

5.6. Free of charge

- (145) Alphabet shall provide the interoperability solutions and measures implemented to allow for effective interoperability with the features listed in this Annex free of charge, irrespective of their beneficiary, app, service, product, and use case. Alphabet shall also not charge any fees indirectly for any of the measures set out in this Annex.

5.7. Documentation and APIs

- (146) Alphabet shall make available complete documentation of all interoperability solutions it will be making available in the context of the implementation of the measures set out in this Annex. This includes Alphabet making available complete, accurate, and well-documented APIs to the extent that access to such APIs is relevant for the implementation of the measures set out in this Annex.
- (147) Alphabet shall maintain this documentation and these APIs in line with Alphabet's internal processes and policies, so that effective access for the purpose of ensuring interoperability is guaranteed over time.

5.8. Assistance and testing

- (148) Alphabet shall provide reasonable technical assistance, free of charge, to third parties to implement and achieve effective interoperability with the features in this Annex.
- (149) Alphabet shall ensure that all interoperability solutions implemented to address the measures specified in this Annex are subject to Alphabet's usual practices, including beta testing.

5.9. Future updates and new functionalities

- (150) Should Alphabet make changes to a feature listed in in this Annex, including adding new functionalities to the feature or making updates, Alphabet shall:
- (a) Develop such new or updated features or functionalities in a way that they are interoperable with third-party services or hardware.
 - (b) Include the interoperability solutions at an appropriate time in the beta version of the new or updated features or functionalities.
 - (c) Make accessible the updated interoperability solution on a Google Android mobile device and documentation for the relevant feature no later than at the time the new or updated feature becomes accessible to any Alphabet service or hardware on the same Google Android mobile device.

⁽⁸⁾ For example, Gemini Nano is downloaded via AI Core when the user takes an action that triggers the download, e.g. by using a feature in Alphabet's apps that requires Gemini Nano.

⁽⁹⁾ For example, in Google Play, Alphabet implements the Play Auto Install and Google Play Inline Installs mechanisms.

- (151) Alphabet shall maintain the interoperability solution over time such that the solution and its documentation continue being available, functional, usable, and effective for all developers without interruption. If, in exceptional circumstances, Alphabet wishes to deprecate an interoperability solution or parts of it, Alphabet shall submit a reasoned request in accordance with the procedure described in paragraph (158) of this Annex.

5.10. Reporting

- (152) Alphabet shall communicate to the Commission shortly after the notification of the decision pursuant to Article 8(2) of Regulation (EU) 2022/1925 in the present proceedings, and in any event within two months of the date of notification of such decision, all measures that it intends to take to comply with the decision in sufficient detail to enable the Commission to assess whether the measures are *prima facie* suitable to address the requirements of the decision (“Initial Implementation Report”). The Initial Implementation Report shall, for each specified measure, include, in particular, the following information:
- (a) a detailed description of the interoperability solutions that Alphabet intends to make available;
 - (b) a description of how these solutions address all of the measures in the decision and how it will provide third parties with effective interoperability under equal conditions to those available to Alphabet’s services;
 - (c) if applicable, a description of any integrity measures that Alphabet intends to apply; and
 - (d) a detailed timeline for the design, development, implementation and release of each of the effective interoperability solutions.
- (153) Following the Initial Implementation Report and until the expiry of the implementation deadlines for all features, Alphabet shall provide, on a monthly basis, a report (“Monthly Implementation Report”) providing, in detail, for each specified measure, the following information:
- (a) the envisaged interoperability solutions that Alphabet intends to make available to address each of the specified measures;
 - (b) status on the design, development, implementation and release of each of the interoperability solutions, including whether progress is on schedule and, if not, why not;
 - (c) if necessary, an updated detailed timeline for the design, development, implementation and release of each of the interoperability solutions, including, for instance, beta testing;
 - (d) any potential issue that Alphabet may have encountered during the design, development, implementation or release of each of the envisaged interoperability solutions;
 - (e) a description (including, where necessary, the relevant supporting documents) of Alphabet’s engagement with third parties for the purposes, and in the context of, the design, development, implementation and release of each of the interoperability solutions. This shall include:
 - (1) a description of the issues raised by third-party developers that relate to the features and an assessment of whether such issues will require a

- modification of the interoperability solution, including any relevant rules (e.g. rules on background execution); ⁽¹⁰⁾
- (2) any information and feedback provided to Alphabet by third parties regarding the betas of Alphabet's interoperability solutions;
 - (3) all interoperability requests made by third parties with regards to all features referred to in Sections 1-4 of this Annex.
- (f) any relevant changes that Alphabet has made or intends to make to policies, requirements and agreements, including the CDD, the GMS requirements, placement agreements, etc.;
 - (g) a description of how OEMs have applied customisations, if any, to the interoperability solutions, including any relevant rules (e.g. rules on background execution).
- (154) At least 10 working days prior to any public release in relation to a feature or functionality within the scope of these Proceedings, be it beta testing or the final solution, Alphabet shall communicate its plans to the Commission ("Pre-release Feature Implementation Report").
 - (155) Upon expiry of the implementation deadline for each feature, Alphabet shall provide a report ("Final Feature Implementation Report") to the Commission explaining all the measures that it has taken to comply with the final decision and confirming the date of its public release. Under this obligation, Alphabet shall describe the interoperability solution made available to third parties, including all technical details and potential APIs, as well as any potential integrity measures.
 - (156) For two years after expiry of the implementation deadlines for all features, Alphabet should provide to the Commission, every six months, a report ("Semi-annual Implementation Report") containing the same information as the Monthly Implementation Report in relation to any change that Alphabet has made or intends to make to the interoperability solutions, OEM customisation, and engagement with third parties.
 - (157) Alphabet shall provide the Commission with a non-confidential version of each of the abovementioned reports (Initial Implementation Report, Monthly Implementation Report, Pre-release Feature Implementation Report, Final Feature Implementation Report and Semi-annual Implementation Report) for publication when the report is due.

5.11. Waiver

- (158) The Commission may, in response to a reasoned request from Alphabet showing good cause, modify or substitute one or more of the measures or a part of them. To ensure the effectiveness and timely implementation of the measures, it is important that the request does not have the effect of suspending the implementation of the measures and, in particular, of suspending the expiry of any time period in which the measure has to be complied with.

⁽¹⁰⁾ Modifications of the rules may also include changes to existing APIs and the introduction of new APIs that relate to background execution.