Platform Interoperability Report

September 2026

Contents

Apple's commitment to platform interoperability 3

The evolution of Apple's platforms 3

Fostering developer innovation through platform interoperability 4

Privacy, security, and integrity by design 4

Developer engagement 5

Platform interoperability enhancements announced at WWDC26 6

Enhanced Apple Intelligence and other AI technologies and capabilities 6

Deeper system integrations and discoverability 8

Refined design and UI frameworks 11

Tools and resources for games 12

Investing in protocols and standards 12

Enhanced app development productivity 13

Apple's support for platform interoperability in the UK 15

Engagement with the UK developer community 15

Showcasing UK developer success 15

Legal reporting 15

Apple’s commitment to platform interoperability

At Apple, we empower developers to build great experiences for our users. We have long invested in system capabilities and developer frameworks, tools, and technologies that enable richer integration across apps, services, and devices. Our commitment to fostering innovation and platform interoperability guides these investments, along with preserving the privacy, security, safety, performance, and integrity of the Apple ecosystem.

The evolution of Apple’s platforms

Apple’s operating systems are the foundational software layers that manage hardware and software, and they provide the platforms on which applications operate. They enable core functions such as security, networking, identity and authentication, user interfaces, data management, and communications between applications, services, and hardware components.

Modern operating systems support complex ecosystems of applications, cloud services, connected devices, artificial intelligence capabilities, payment technologies, accessibility features, and developer frameworks. They run on devices that users depend on throughout their daily lives. In a networked world, they face constant threats of attack. Operating systems must balance an increasing range of objectives, including functionality, privacy, security, safety, ease of use, design, reliability, integrity, and performance.

Apple has a range of operating systems: iOS, iPadOS, macOS, watchOS, tvOS, and visionOS. Each year, Apple invests in advancing these platforms through new user features, developer tools and technologies, privacy and security protections, accessibility enhancements, and performance improvements. These investments enable devices to support increasingly sophisticated user experiences and provide a robust foundation for developers to build innovative apps while protecting users’ privacy, security, and safety.

Apple’s approach to interoperability must also protect the IP at the core of its platforms. Apple invests billions of dollars in research and development to build the technologies that underpin the Apple experience on its platforms and drive its competitive edge. These investments are protected by an extensive portfolio of Apple IP, including trade secrets, patents, and copyrights. Providing interoperability involves granting access to Apple’s IP-protected technologies, which can affect Apple’s innovation incentives and ability to differentiate its offerings. Apple therefore aims to give third parties the necessary access to develop their own innovations on Apple’s platforms while protecting its own proprietary efforts and differentiated offerings. Because Apple innovates with knowledge of how its products and services work together and protect users, developing interoperable solutions often involves a second step of investment specifically to facilitate third-party use.

Fostering developer innovation through platform interoperability

Apple's approach to platform interoperability is grounded in a belief that users benefit most from a vibrant, trusted, and innovative ecosystem of apps, services, and connected devices. It helps users build confidence in the ecosystem and in turn enhances developers' ability to invest in building apps.

Apple has enabled developers of all sizes to reach users across Apple's platforms, helping to create an economy that now supports millions of jobs globally and has facilitated over $1.4 trillion in commerce for developers.

The App Store hosts millions of apps across a broad range of categories, including productivity, education, health, entertainment, financial services, communications, and AI services. Developers can build experiences that interoperate with iPhone, iPad, Mac, Apple Watch, Apple TV, and Apple Vision Pro.

Apple provides developers with over 250,000 application programming interfaces (**APIs**). It also offers development tools, testing environments, analytics tools, technical support resources, and extensive documentation. To support these integrations, Apple emphasizes stable APIs, clear implementation standards, and long-term platform support so that integrations remain reliable and consistent across software releases and device generations. This approach has enabled developers to create rich and differentiated user experiences using platform capabilities such as Widgets, Live Activities, Dynamic Island, Shortcuts, HealthKit integrations, and Face ID and Touch ID authentication frameworks.

Privacy, security, and integrity by design

Apple develops APIs, frameworks, and platform access models with privacy, security, and safety built in from the beginning. It does so by applying safeguards designed to protect users and preserve ecosystem integrity, such as:

- **Security safeguards**, including runtime protections, code signing requirements, sandbox enforcement, entitlement controls, encryption, Secure Enclave protections, risk evaluations, unlock authentication, and fraud and malware prevention mechanisms.
- **Privacy safeguards**, including data minimization principles, transparency requirements, on-device processing, granular user consent flows, and limitations on unnecessary background access to sensitive data.
- **Integrity and performance safeguards**, including memory limits, Low Power Mode, process prioritization, and runtime limits.

In certain cases, enabling interoperability for third-party apps or hardware is not possible or appropriate while maintaining essential platform safeguards, even where there is developer appetite for access. Interoperability may create disproportionate privacy, security, safety, or integrity risks for users by requiring access to sensitive system capabilities in ways that undermine or avoid system protections. Apple looks for safe alternatives where possible. And finally, as discussed above, IP protection and disclosure of trade secrets are additional factors when considering if and how to

provide interoperability support. Apple's IP rights reflect and secure the innovations that its technologies and features are built on. Apple may decline to provide or limit interoperability that would give access to Apple's proprietary, IP-protected innovations, and may charge for access to Apple tools and technologies.

Developer engagement

Apple maintains extensive ongoing developer engagement and feedback processes designed to ensure that interoperability frameworks and APIs continue to evolve in response to real-world developer needs.

The annual Worldwide Developers Conference is the key moment when Apple showcases its biggest platform updates to a global audience of developers. It also serves as an important mechanism for engaging directly with the developer community. It includes hundreds of technical sessions, hands-on labs, workshops, and one-to-one developer meetings covering APIs, interoperability features, security protections, accessibility, design, and performance optimization. At WWDC26, Apple hosted 107 technical sessions, 19 group labs, 815 one-to-one online lab appointments, and 57 Q&A forum sessions.

Apple also engages extensively with developers year-round through:

- Apple Developer Forums.
- Feedback Assistant and bug reporting systems.
- Apple Developer Technical Support.
- Worldwide Developer Relations and community engagement.
- Developer labs and consultation channels.
- Developer betas and preview releases that allow developers to test and provide feedback on interoperability enhancements ahead of public launch.

Enriching people's lives with amazing experiences is a purpose Apple shares with developers. Apple's commitment extends to hearing from developers about interoperability that preserves the user experience and protects user privacy and security at every step. To support this, Apple provides a dedicated form for Apple Developer Program members to request additional platform interoperability with iOS and iPadOS.[1]

[1] This process does not apply in the European Union, where Apple is required to have a separate process under the Digital Markets Act.

Platform interoperability enhancements announced at WWDC26

At WWDC26, Apple announced 38 new frameworks and over 38,000 new symbols[2] across its platforms. These are available to developers on iOS 27, iPadOS 27, macOS 27 Golden Gate, watchOS 27, tvOS 27, and visionOS 27. They can be summarized into the following themes:

- **Enhanced Apple Intelligence and other AI technologies and capabilities**, enabling developers to bring even more intelligent experiences to their users while putting user privacy first.[3]
- **Deeper system integrations and discoverability**, allowing developers to make their apps' content, data, and actions available to system functions like Siri AI, Spotlight search, and Shortcuts.
- **Refined design and UI frameworks**, including improvements to SwiftUI and Liquid Glass.
- **Additional tools and resources for game developers** that help them bring high-quality games and spatial experiences to Apple platforms.
- **Continued investment in protocols and standards**, helping developers build experiences that operate seamlessly across apps, platforms, and ecosystems.
- **Development tools** that enhance developer productivity through agentic coding.

Enhanced Apple Intelligence and other AI technologies and capabilities

Foundation Models framework

Building on the Foundation Models framework introduced at WWDC25—which gave developers access to the on-device large language model (**LLM**) at the core of Apple Intelligence—Apple announced several enhancements this year:

- **Multimodal prompting**. The Foundation Models framework now supports multimodal prompts, enabling developers to analyze and reason about images alongside text. Combined with on-device Vision framework tools, which include new capabilities in image understanding (such as "tap to segment" specific objects or prompting to caption images), developers can classify images, extract information, summarize documents, and generate accessibility descriptions.
- **Extending the Foundation Models framework to any model**. Developers can now bring their model of choice—whether local or server-based—using the new LanguageModel protocol. It gives developers a consistent way to integrate and leverage any AI model that supports the protocol across their apps, enabling their

[2] Symbols are callable software elements exposed by a software library or framework. They are individually named entry points within the APIs that group them together.

[3] Apple Intelligence features are available only in supported regions.

apps to move quickly between Apple and third-party models. This is made possible by a new public protocol: any model provider can create a Swift package that conforms to the LanguageModel protocol, so developers can pick the model they want for their app. Developers call them all the same way and can easily swap one for another without changing their session code. A model built on the protocol still gives access to wider framework features such as building agentic app experiences with Dynamic Profiles (see below).

- **Server-side intelligence with Private Cloud Compute**. The Foundation Models framework now provides access to a more powerful server-based model through Private Cloud Compute with just a single line of code. The model supports advanced reasoning, richer context handling, and agentic tool use—all while preserving user privacy—and is available at no cloud API cost to eligible developers in the App Store Small Business Program with fewer than two million first-time App Store downloads.[4]

- **Dynamic Profiles**. The Dynamic Profiles API enables developers to adjust models, tools, and instructions dynamically within a single session, allowing apps to adapt to changing context and support more sophisticated, multi-step user workflows.

- **New ways to access the Foundation Models framework**. Developers can now access Foundation Models in new ways on macOS through the `fm` command-line tool and a new Python SDK, making it easier to incorporate Apple's models into automation, machine learning, and data science workflows.

- **The Foundation Models framework is going open source**. Developers can start using the Foundation Models framework on other platforms such as Linux Swift on Server. This will make it the preferred Swift API for any LLM on any platform.

Core AI framework

The Core AI framework allows developers to build, run, and deploy AI models in their apps by providing greater control over model specialization, inference performance, and caching to reduce load times. For apps that download models or need to load them quickly, Core AI provides APIs to control when specialization occurs, how caches are managed, and how much storage is used for specialized assets.

Building on the foundation established by Core ML, Core AI gives developers greater flexibility in how they deploy and optimize any AI model they choose on Apple platforms. By exposing more of the model execution lifecycle and enabling developers to take advantage of the CPU, GPU, and Neural Engine through a common framework, Core AI makes it easier to bring existing AI models and workflows to Apple devices while delivering high-performance on-device experiences.

At the forefront of Core AI's ecosystem is the coreai-models repository, which includes a Swift package for running LLMs in developers' apps. It is an open-source repository of models, including cutting-edge LLMs. It also includes reusable components and agent skills that developers can install into their coding assistant to leverage Core AI best practices from day one.

---

[4] For full eligibility requirements, see https://developer.apple.com/private-cloud-compute/.

Image Playground API

Image Playground has been reimagined with a powerful new generative model running on Private Cloud Compute, letting developers create high-quality images in virtually any style, including photorealistic—right in their own apps. Users can start from a photo or a description, then refine their images using natural language or touch, and choose from landscape or portrait dimensions to use their creations in even more places. All of these new capabilities are available to developers through the Image Playground API. These capabilities build on enhancements announced last year—moving from limited to unlimited styles, improving image quality, and simplification through natural language prompting.

Music Understanding framework

Developers now have access to on-device musical intelligence that enables apps to analyze audio across six dimensions: key, rhythm, structure, pace, instrument activity, and loudness. The Music Understanding framework handles the underlying signal processing and model inference, eliminating the need for expertise in signal processing or machine learning. This enables developers to incorporate advanced music analysis capabilities into their apps without building and maintaining their own audio intelligence pipelines.

Evaluations framework

Building app features with generative AI poses new testing challenges, because the same input can produce different outputs. With software using generative AI, developers can't rely on functional consistency to verify behavior. Users expect intelligent features in developers' apps, like any feature, to be safe, trustworthy, and reliable. Shipping a feature with unpredictable behavior can have adverse consequences for an app's reputation.

Now, developers can use the Evaluations framework to systematically evaluate their app's intelligence-powered features. Developers can compare prompt strategies, track quality over time, and catch regressions before they ship.

The framework evaluates developers' intelligence-powered features against the metrics they define, from simple pass or fail checks to detailed scoring with model-as-judge patterns. It aggregates the results into summaries that show developers which approach performs best and where individual responses fall short. The framework works with any model available through the Foundation Models framework, including on-device models, Private Cloud Compute, as well as a variety of frontier server models. Additionally, the Foundation Models Instrument records every model request, response, tool call, instruction set, and latency during an interaction, enabling developers to inspect the full execution trace to understand how prompts, tools, and instructions change over time and where any failures occurred. It also provides performance metrics to optimize reliability and responsiveness.

Deeper system integrations and discoverability

App Intents framework

Apple has made its platforms more seamless than ever by enriching the ways in which users can interact with app content. Someone might use Siri AI while driving to perform actions using developers' apps, like sending a message or creating a note. When

Spotlight displays an app's content in search results, the person can tap that result to launch the app to display the content there. Widgets can display apps' content from the Home Screen, Lock Screen, and other places, and even let people interact with that content. All of these features rely on developers' apps supporting the App Intents framework, which provides the infrastructure to make their content available from outside their app.

Apple expanded App Intents to make app capabilities more accessible across system-wide experiences like Siri AI, Shortcuts, and Spotlight at WWDC25. At WWDC26, Apple announced enhanced Apple Intelligence and Siri AI technologies to make these integrations more intelligent, more contextual, and significantly easier to adopt and scale.

- **App Intents and App Entity schemas**. Developers can now build App Intents for common actions, data types, and enumerations, such as those used by music apps (e.g., play/pause actions) and email apps (e.g., drafting a new email) using predefined schemas, rather than creating custom App Intents from scratch. This information enhances the system's understanding of an app's content and helps facilitate interactions with Siri AI, Apple Intelligence, and other system features.

- **Updates to making app entities available in Spotlight**. Developers can now integrate more deeply with Spotlight by displaying results in their app's own search interface when queries return large result sets, and by giving Spotlight greater visibility into app content through enhanced indexing controls. This improves the discoverability of app content in Spotlight and Apple Intelligence experiences.

- **View Annotations API**. The new View Annotations API adds on-screen awareness by letting developers map their apps' views to entities so users can reference and act on what's right in front of them conversationally.

- **Performing App Intent actions**. Developers can now extend the behavior of their App Intents by adding support for additional protocols. These include LongRunningIntent for actions that need extra time to run, CancellableIntent to support system- or user-initiated requests to cancel an in-progress action, and UndoableIntent for actions that affect the app's interface or content in a way people might want to reverse.

- **App Intents Testing framework**. The new App Intents Testing framework lets developers validate their integrations with Siri AI, Shortcuts, and Spotlight through real system pathways, without UI automation, so they can catch issues early and ship with confidence.

- **Making indexed app content available to Foundation Models**. Developers can now ensure the Foundation Models framework takes their app's content into consideration by making their data available to the model using the Spotlight search tool. This ensures that the model can answer specific questions about an app's content and helps users find the information they need via Spotlight across their apps.

Media Intents framework

Developers can use the Media Intents framework to receive and resolve Siri AI's audio search and playback requests, regardless of how a user phrases them. Users can make both specific requests, such as asking Siri AI to play a particular song or artist, and vague requests, such as asking Siri AI to play energetic music. The system delivers the request to the app using the App Intents framework, and the developer can use App Intents to return matching results to Siri AI.

Now Playing framework

Apple has introduced the Now Playing framework, which makes it easy for developers to surface their app's media content across the system now-playing experience on all Apple platforms, including the Lock Screen, Control Center, Apple Watch, and CarPlay. Developers describe their media content and supported commands using an Observable model, and the framework updates the system automatically whenever the model changes. The framework supports two kinds of sessions:

- Local sessions publish playback happening on the current device. The framework observes the developer's model and publishes changes to the system as they happen.

- Remote sessions publish playback happening on external devices, such as speakers, streaming sticks, or smart TVs. Developers can start these from their app, or from their server using push notifications when the app isn't running.

Suggested Actions framework

The Suggested Actions framework allows developers to display suggested actions privately and securely next to a message in their messaging app based on content, participants, subjects, and dates in a message (e.g., creating an event in Calendar, adding a task to Reminders, or opening a location in Maps).

Media Device framework

The new Media Device framework enables developers to make compatible playback hardware available system-wide, allowing users to discover and stream to those devices directly from any app using the standard media output picker used for AirPlay and Bluetooth devices.

Developers, by building a MediaDevice Extension, can handle device discovery, connection, and playback on behalf of the system—a model that gives developers flexibility within the media-sharing lifecycle and allows universal and seamless integration with apps that have integrated the standard media output picker. Media apps observe route changes and control playback while the developer's extension handles the hardware-specific communication.

Developers that adopt the RealtimeSampleHandling protocol can additionally support real-time audio and video streaming, including screen mirroring.

Shortcuts enhancements

Shortcuts lets people combine everyday app actions and surface them across the system. At WWDC26, Apple announced updates to Shortcuts to further integrate third-

party app actions and content across these system capabilities. This update introduces three primary features:

- Streamlined in-editor automations featuring three new trigger types—screenshot creation, external keyboard connection, and notification receipts—which allow developers to design concise, descriptive notifications that support fine-grained, keyword-filtered triggers.

- The Use Model action is upgraded with web-retrieval capabilities and access to the latest Apple Intelligence models, supported by a new model transcript inspector that allows developers to evaluate the exact App Intent entity properties passed to the system's model.

- A new native Storage capability (supporting get, set, and global storage values synced via iCloud) enables shortcuts to persist data between runs, enabling developers to implement stable, device-consistent identifiers to ensure that custom App Intent entities are correctly recognized across all of a user's synced devices.

Refined design and UI frameworks

Additions to SwiftUI

SwiftUI is designed to handle the complexities of layout, animation, and platform integration so developers can focus on what makes their app unique. This year, SwiftUI makes apps more responsive without requiring developers to change their code through more efficient state initialization and faster layout rendering. Developers can now write less custom code using new reorderable containers, and the new Spatial Preview framework allows 3D models from Mac apps to be viewed spatially when streamed to Apple Vision Pro.

Swift unifies full-stack development, providing the speed, expressiveness, and memory safety developers need to build everything from full-featured mobile apps and embedded firmware to platforms beyond the Apple ecosystem. It prioritizes usability and an intuitive syntax, allowing developers to iterate easily on their ideas while the compiler catches mistakes along the way. Building on this foundation, Swift 6.4 is designed to make everyday tasks feel effortless. It streamlines the daily developer experience by introducing targeted warning suppression, simplified availability attributes like `anyAppleOS`, and improved compiler diagnostics.

Liquid Glass enhancements

Liquid Glass provides a unified visual design that makes apps more expressive and instantly familiar across Apple platforms. This year, the design continues to evolve, with improvements to legibility, customizability, and consistency. A new transparency slider in Settings allows users to personalize their visual experience by adjusting the appearance to their preference, while the design also seamlessly adapts to a variety of accessibility settings users may choose. Apps that have already adopted Liquid Glass customizations gain these improvements automatically. These updates culminate in an even more focused and approachable experience across apps and platforms.

Tools and resources for games

Managed Background Assets

Managed Background Assets reduce game install sizes by introducing intelligent, localized delivery. The system now automatically identifies a player's preferred language and only downloads the specific asset packs required for that language, falling back to the closest match if necessary. Additionally, a new Steam Asset Converter streamlines the process of adapting PC games for iOS, iPadOS, macOS, tvOS, and visionOS.

Game Porting Toolkit 4

Game Porting Toolkit 4 accelerates the development of Mac games by introducing open-source skills for use with agents, equipping them with Apple-specific best practices for Metal development and dramatically cutting the time it takes to bring games to Apple platforms.

Official Unity plug-ins

Official Unity plug-ins bring native Apple integration to one of the most popular game engines. Developers now have access to Apple-provided plug-ins for StoreKit and Background Assets, allowing them to natively implement In-App Purchases across all Apple platforms within their existing Unity workflow.

Reality Composer Pro 3

Reality Composer Pro 3 allows developers to build complete spatial experiences in a single tool. With new Live Previews combined with Mac Virtual Display, developers can now see the results of their edits immediately as they make them.

Investing in protocols and standards

Interoperability also depends on open standards and protocols. Apple continues to invest in standards-based technologies that help developers build experiences that operate across ecosystems.

WebKit platform enhancements

This year, Apple announced a host of changes aimed at improving the quality of WebKit, including 58 new features and 525 fixes—the largest number of fixes in any recent Safari release. This work was aimed at improving the behavior of existing features, handling more edge cases, fitting together with other features, and aligning with web standards. These changes improve compatibility with the broader web ecosystem and reduce the effort required to support Safari alongside other browsers.

Bluetooth Channel Sounding

As part of Apple's ongoing investment in open connectivity standards, Apple has added support for Channel Sounding—the precise-ranging feature defined in the Bluetooth Core Specification—giving developers a way to find and interact with nearby Bluetooth-only accessories. Through the Core Bluetooth framework, developers can now use Channel Sounding to measure the precise distance between a device and their nearby accessory. Channel Sounding is transparently balanced by the operating system with system-side Bluetooth and Wi-Fi activity, eliminating complexity for both the user and developer.

Apple has also extended support for Nearby Interaction to allow developers to measure both distance and direction with Core Bluetooth using Channel Sounding and fusion between the raw Channel Sounding measurements and camera input. Apple has posted a freely available video to help developers understand and adopt it. This opens up new possibilities for developers to create intuitive, context-aware user experiences.

Enhanced app development productivity

Apple continues to invest in tools and technologies that help developers build, test, maintain, and scale their app enhancements more efficiently. Building on the intelligence capabilities introduced in Xcode 26 at WWDC25, this year's announcements represent a significant step forward in bringing agentic workflows directly into the software development lifecycle. Together, these enhancements help reduce development complexity, lower costs, and enable developers to bring new experiences to market more quickly.

Xcode 27 and agentic coding

Xcode is the best place to build for Apple platforms, and the best place to code with agents. Xcode 27 takes the next big step in agentic coding. With deeper support for the Model Context Protocol (**MCP**) and Agent Client Protocol (**ACP**), Xcode can now connect directly to models and agents from Anthropic, Google, and OpenAI. Conversations with coding agents feature interactive planning, multi-turn Q&A, and a canvas that can render Markdown and display code changes and previews right alongside. Coding agents can also be used by developers to easily localize their apps by adding languages, updating string catalogs, translating strings, and adding language-specific plural variants when needed.

Xcode 27 also gives coding agents the tools to validate their own work, so they can run autonomously for longer, including by writing and running tests, trying ideas in isolation with Playgrounds, checking visual changes with previews, and interacting with the simulator in the new Device Hub. Device Hub brings a developer's simulated and physical devices together in a single workspace, so developers can evaluate new features on simulators, test hardware-dependent features on physical devices, and get finer control over device settings, diagnostics, and pairing through an expanded view.

With plug-ins, developers can extend Xcode with custom skills, bring in the tools they use every day through MCP, and connect any agent compatible with ACP. GitHub and Figma are the first to offer seamless installation between their tools and Xcode.

Xcode Instruments now has a more refined workflow that helps developers find and fix problems that make an app feel slow or unresponsive. The Swift Concurrency instrument shows where background tasks are competing with each other, alongside Time Profiler for spotting what's using too much processing power and System Trace for a closer look at how the app interacts with the system. These enhancements let developers quickly surface the heaviest call stacks and measure the impact of changes to select and deploy a fix.

UIKit modernization

Apple also continues to reduce the effort required to adopt the latest platform capabilities. Developers can use coding agents and modern UIKit technologies to help

modernize existing applications, update layouts for new device form factors, and adopt new navigation, menu, and interface APIs. These capabilities help developers bring existing applications to new devices and experiences more efficiently.

Local AI development with MLX

Apple continues to invest in the infrastructure developers use to build and deploy AI systems. Enhancements to MLX enable developers to build and run fully local agentic AI workflows on Apple silicon, preserving privacy while eliminating cloud costs. New capabilities, including GPU Neural Accelerator support and the MLX-LM Server, make it easier to experiment with, train, fine-tune, and deploy advanced AI models directly on Apple hardware.

Apple's support for platform interoperability in the UK

Apple's support for platform interoperability has enabled UK developers to build globally successful businesses across sectors including fintech, transport, health, fitness, and creative services, with over 120,000 apps in the App Store. Apple continues to invest in tools, APIs, and developer engagement processes that help UK developers reach users across more than 175 App Store storefronts worldwide while building high-quality app experiences.

Engagement with the UK developer community

Apple maintains an active and ongoing dialogue with the UK developer community through a combination of global and local engagement initiatives.

In addition to WWDC and Apple's year-round developer support programs, Apple regularly hosts in-person developer events at its London offices, as well as online sessions. Over the past year, Apple has welcomed more than 500 local developers to workshops and technical sessions, which focused on topics such as the Foundation Models framework and Apple's latest design technologies. These events provide developers with valuable opportunities to engage directly with Apple engineers, technology evangelists, Worldwide Developer Relations teams, and App Store experts.

Apple also gathers feedback through developer forums, technical support channels, beta programs, labs, and consultation sessions, helping to ensure that platform technologies continue to evolve in response to real-world developer needs. And Apple engages with independent Apple developer communities in the UK, including at the annual iOSDevUK conference in Wales and the SwiftLeeds conference in Leeds.

Showcasing UK developer success

Apple actively promotes and celebrates the work of UK developers through a range of initiatives. These include App Store editorial features, developer spotlights, case studies, and the annual Apple Design Awards, which recognize outstanding innovation, creativity, and technical achievement on Apple platforms.

For example, UK-based developer Hyper Luminal Games received a 2026 Apple Design Award for Pine Hearts, while other UK-based developers were finalists. Through these channels, Apple helps UK-based developers increase their visibility, reach new audiences, and share their innovations with users and the broader developer community.

Legal reporting

To support Apple's commitment to platform interoperability, Apple provides a dedicated form for Apple Developer Program members to request additional interoperability with iOS and iPadOS features. Under commitments agreed with the UK Competition and Markets Authority in April 2026, Apple has agreed to publish on an annual basis certain information demonstrating Apple's ongoing commitment to

platform interoperability for UK-based developers. The relevant reporting period for this report is January 1, 2026, to June 30, 2026.

Apple's active and ongoing engagement with the developer community described above means that developer needs are predominantly identified and addressed directly through Apple's engagement channels, without the need for them to make a formal interoperability request. Accordingly, Apple received only one request from a UK-based developer via the dedicated interoperability request channel during the relevant time period.

Confirmatory statements

Apple confirms that it has maintained its dedicated interoperability request channel, including its internal process where requests are reviewed in the order in which they are received. Apple also confirms that the substantive review criteria it applies to the assessment of eligible requests, which are listed on the form's webpage, remain the most appropriate to use and have not changed.

Transparency metrics

Metric	#
Number of requests received in relevant reporting period	1
Time to Phase 1* decision	26 days
Phase 1 decisions	
Moved to Phase 2†	*0*
Rejected	*1*
Requests incorporated into Apple's feature development cycle	0

** Phase 1 involves Apple making an initial assessment of the eligibility of the developer's request, as described on the webpage.*

† Phase 2 describes Apple's annual software planning process, as described on the webpage. Requests deemed eligible at Phase 1 are moved to Phase 2.

Summary of requested functionalities

The one request Apple received from a UK-based developer in the relevant period concerned system-level event handling and background proactive assistance for Messages, Phone, alarms, Reminders, and Notifications. Apple reviewed this submission and determined that the request did not meet the criteria for an eligible interoperability request. An eligible request must concern access to an operating system capability that is used by Apple's hardware and/or services. The requested access to a user-authorized proactive assistance framework for accessibility apps does not concern an operating system capability that is used by Apple's hardware and/or services.

© 2026 Apple Inc. All rights reserved. Apple and the Apple logo, Apple Watch, Siri, and watchOS are trademarks of Apple Inc., registered in the U.S. and other countries. Apple Intelligence is a trademark of Apple Inc. iCloud is a service mark of Apple Inc., registered in the U.S. and other countries. iOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license. Other product and company names mentioned herein may be trademarks of their respective companies. Product specifications are subject to change without notice. This material is provided for information purposes only; Apple assumes no liability related to its use. Not all features are available everywhere. See www.apple.com/ios/feature-availability for feature availability in your region or language. September 2026.