Open Mobile Hub Project FAQ

[PUBLIC RELEASE]

First Draft - April 10, 2024
Final Draft - April 18, 2024
V1.0 Public Release - TBD

Table of Contents

Open Mobile Hub Technical Architecture

What Exactly Is the Framework?

Open Mobile Hub (OMH) acts as an abstraction layer or unified interface that sits above the Software Development Kits (SDKs) from multiple service providers. This layer is designed to standardize and simplify the way applications interact with underlying services, regardless of the service provider. The primary technical objectives and benefits of OMH include:

Open source: As an open source project, OMH offers several benefits associated with open source software, including freedom and flexibility in usage, high quality, security, compliance, and access to support and innovation.

Unified interface: Open Mobile Hub provides a single, standardized API interface for application developers. This means that instead of having to learn and integrate multiple APIs for similar services from different providers, developers can code against one OMH interface. This drastically reduces the learning curve and integration efforts.

Provider agnosticism: By abstracting the specifics of each service provider's SDK, OMH enables applications to be less dependent on any single provider's technology. This agnosticism allows for easier swapping of service providers with minimal to no changes in the application codebase, facilitating better negotiation leverage, cost efficiencies, and resilience against service discontinuation or degradation.

Rapid integration and time to market: Since OMH standardizes access to services across providers, developers can integrate new services more quickly and bring their products to market faster. This is particularly beneficial in a competitive landscape where speed to market is often a critical success factor.

Innovation through service composition: Open Mobile Hub's unified interface encourages innovation by making it easier to combine services from different providers. Developers can leverage the best-of-breed services, mixing and matching functionalities as needed to create unique and powerful application features.

Cross-platform support and filing service gaps: Open Mobile Hub can help address the fragmentation in the Android ecosystem, particularly in regions or devices where Google Play Services are not available. By offering a common layer through which users can access services, OMH can facilitate the provision of alternative services that fill these gaps, ensuring a more uniform application experience across devices and regions.

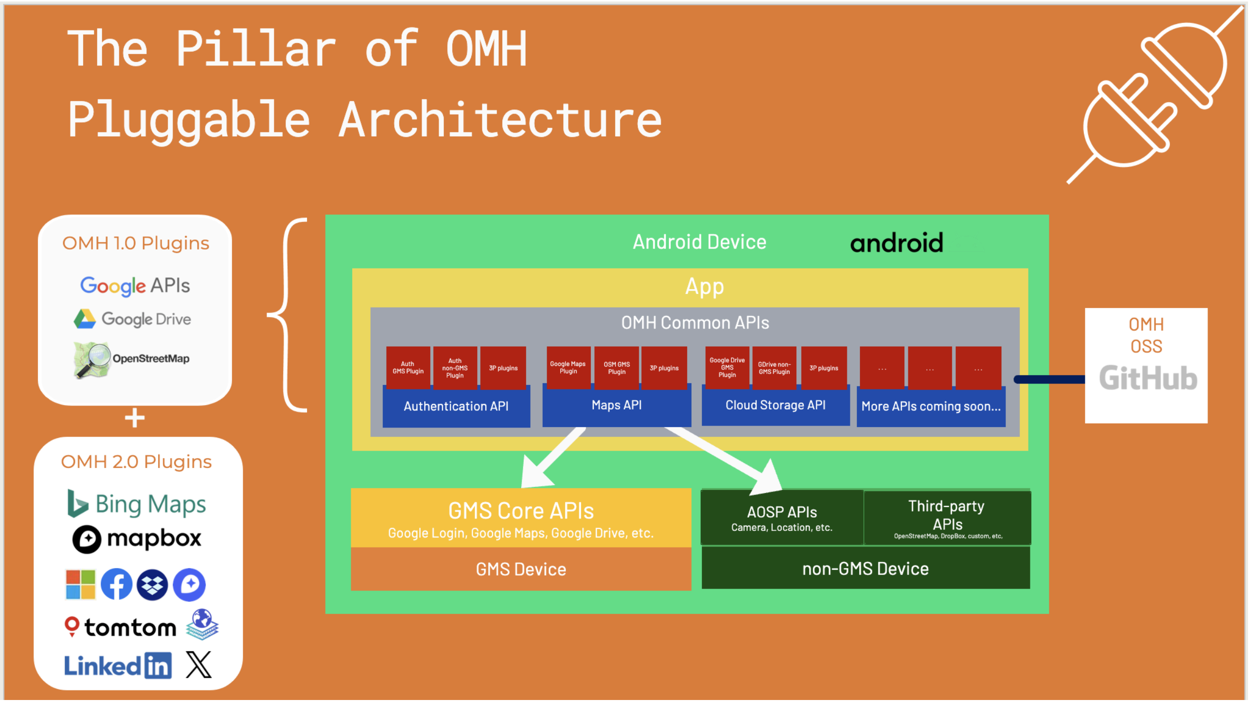
Free of charge: Open Mobile Hub is entirely open sourced and free of charge for users. There is no additional financial burden for app developers to adopt OMH services.

Technically, implementing OMH involves creating a layer that can translate generic calls to this unified interface into the specific calls that each underlying service provider's SDK expects. This requires a deep understanding of the functionalities and idiosyncrasies of each service provider's SDK, as well as the ability to maintain this abstraction layer as services evolve. Service providers, on their part, need to adhere to the specifications of OMH to ensure their services can be seamlessly accessed through this middleware.

In essence, OMH represents a significant architectural design choice aimed at reducing complexity, fostering innovation, and ensuring flexibility in the rapidly evolving landscape of mobile and web services.

What Is the Framework's Architecture?

Open Mobile Hub is designed with a pluggable architecture in mind based on open interfaces available to service providers to surface their services with the goal of providing freedom of choice to app developers while providing interoperability.


The Pillar of OMH
Pluggable Architecture

What Building Blocks Are in the Framework?

The building blocks of the OMH framework are primarily its APIs, which serve as the abstraction layer enabling uniform access to various service provider technologies. These APIs are designed to encapsulate common functionalities that apps might need, regardless of the underlying service provider. Here's a breakdown of these core building blocks within OMH:

1. **OMH Maps API:** This API abstracts the functionality of various mapping services, allowing developers to integrate mapping features into their apps without being tied to a specific mapping service. It supports plugins or implementations for popular services such as Google Maps, Azure Maps, Mapbox, OpenStreetMap, and Apple Maps. This means developers can switch between these mapping services with minimal changes to their code.
2. **OMH Auth API:** Authentication is a critical component of modern apps, and the OMH Auth API provides a unified way to implement authentication mechanisms. It supports various authentication providers, including Google Account, Microsoft Account, Facebook, and Dropbox. This flexibility ensures that developers can offer users multiple authentication options without having to integrate each provider's SDK separately.
3. **OMH Cloud Storage API:** For apps that require cloud storage capabilities, this API offers a unified interface to access different cloud storage services such as Google Drive, Dropbox, and OneDrive. This allows for easy storage and retrieval of app data across different cloud platforms, enabling developers to choose the best service based on their needs or user preferences.

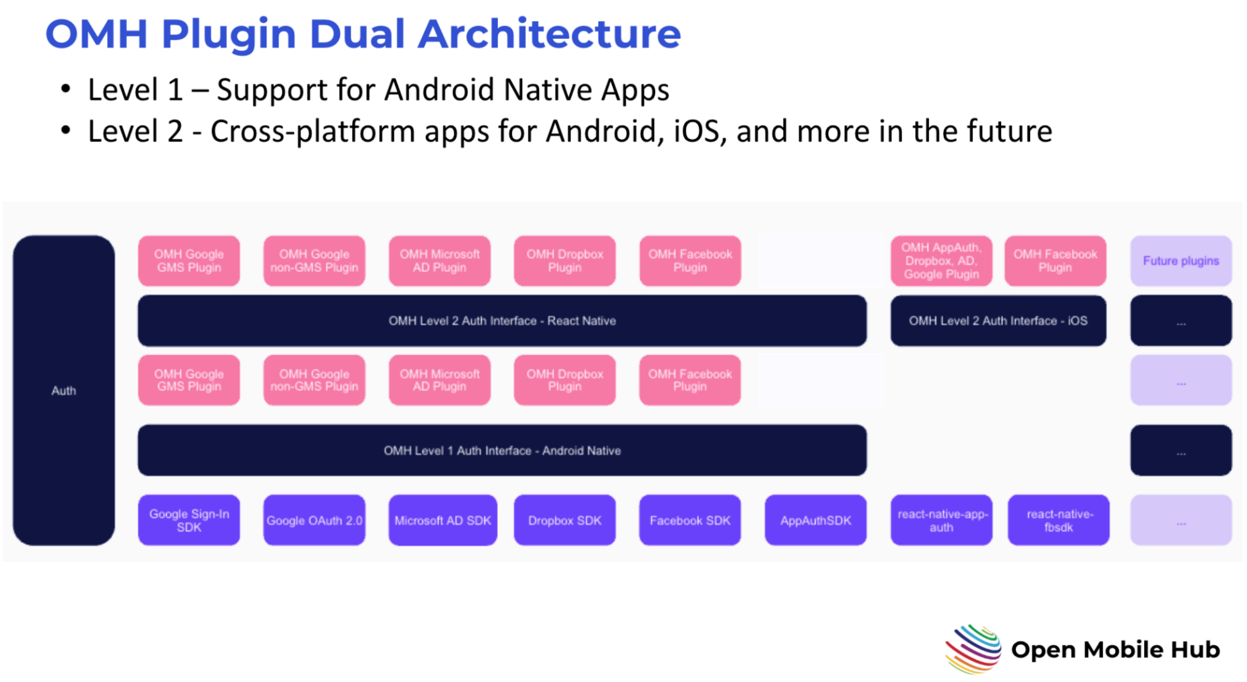
Moreover, OMH is expanding its suite of APIs to include more interfaces that cover other essential functionalities such as messaging, in-app purchases, and app/device integrity checks. This expansion is in line with OMH's goal to provide a comprehensive abstraction layer that caters to a wide range of app development needs.

The architectural foundation of OMH, as illustrated in "The Pillar of OMH - Pluggable Architecture" diagram, emphasizes its pluggable nature. This design allows OMH to serve both Google Mobile Services (GMS) and non-GMS devices seamlessly during runtime, making this process completely transparent to developers. This capability is crucial for ensuring app compatibility and functionality across a diverse Android ecosystem, including regions where GMS is not available. **See further details in the "What Is the Framework's Architecture?" section above.**

Additionally, the "Dual plugin architecture" diagram highlights OMH's support for cross-platform development, specifically for React Native, covering both Android and iOS platforms. This approach not only simplifies the development process for apps targeting multiple platforms but also lays the groundwork for future support of emerging platforms such as HarmonyOS.

In summary, the building blocks of the OMH framework—its APIs—represent a strategic effort to standardize app development processes, reduce dependency on specific service providers, and ensure a broad compatibility range across devices and operating systems. This architecture empowers developers to innovate and adapt quickly to changing market demands and technological advancements.

OMH Dual Architecture



OMH Plugin Dual Architecture

- Level 1 – Support for Android Native Apps
- Level 2 - Cross-platform apps for Android, iOS, and more in the future

Auth

OMH Google GMS Plugin
OMH Google non-GMS Plugin
OMH Microsoft AD Plugin
OMH Dropbox Plugin
OMH Facebook Plugin
OMH AppAuth, Dropbox, AD, Google Plugin
OMH Facebook Plugin
Future plugins

OMH Level 2 Auth Interface - React Native
OMH Level 2 Auth Interface - iOS
...

OMH Google GMS Plugin
OMH Google non-GMS Plugin
OMH Microsoft AD Plugin
OMH Dropbox Plugin
OMH Facebook Plugin
...

OMH Level 1 Auth Interface - Android Native
...

Google Sign-In SDK
Google OAuth 2.0
Microsoft AD SDK
Dropbox SDK
Facebook SDK
AppAuthSDK
react-native-app-auth
react-native-fbsdk
...

Open Mobile Hub

What Is the Interface Between Non-Google Mobile Services and Open Mobile Hub?

The interface between non-GMS environments and OMH is designed to be intuitive for developers, particularly those already familiar with GMS interfaces. Essentially, OMH acts as an abstraction layer that integrates various service providers, including GMS. For Google Maps, for instance, OMH utilizes the Google Maps Android SDK under the hood.

This means that OMH's Level 1 Android-native interfaces draw inspiration from the GMS interface, allowing developers who are accustomed to working with GMS to easily transition to using OMH without needing to learn a completely new SDK. This approach not only facilitates a smoother adoption process for developers but also incorporates best practices and design principles from a range of services, thereby offering a versatile and efficient development environment.

Level 1: Native Apps

For native applications, OMH provides an interface that GMS-based SDKs primarily inspire. This means that developers who are familiar with creating apps for Android using GMS will find the

OMH interface for native development similar, allowing for a smoother transition to development for non-GMS devices. The goal is to offer a set of APIs and services that are intuitive for developers coming from an Android background, thereby reducing the learning curve and accelerating the development process.

Level 2: Cross-Platform Apps
The second level of OMH is focused on cross-platform applications, initially supporting frameworks such as React Native. This choice is strategic, considering the widespread use and community support for React Native. The OMH interface for cross-platform development adopts idiomatic React Native interfaces, ensuring that developers can leverage their existing knowledge and skills to create apps for non-GMS devices without needing to learn entirely new paradigms or languages. This approach not only facilitates easier and faster development but also ensures that apps can maintain high performance and a native look and feel across different platforms.

Future Expansion
Looking ahead, OMH plans to extend its support to other popular cross-platform frameworks such as Flutter and NativeScript. This expansion will further enhance the versatility and reach of non-GMS devices, allowing developers to choose the framework that best suits their needs while still benefiting from the unified OMH interface.

The Goal of Open Mobile Hub Interface
The overarching goal of the OMH interface is to unify and simplify the development process across different platforms and frameworks. By providing a consistent set of APIs and services that cater to both native and cross-platform app development, OMH aims to reduce fragmentation and enable developers to create versatile, high-quality applications more efficiently. For non-GMS ecosystems, incorporating the OMH interface means that it can offer a robust and developer-friendly platform that supports a wide range of applications, from native to cross-platform, thereby enhancing the ecosystem and user experience of these ecosystems.

What Is the Interface Between iOS and OMH?

The interface between iOS and the OMH is tailored to address the unique ecosystem of iOS, which is markedly less fragmented than Android, especially regarding the native development environment. Unlike Android, which has a diverse range of devices and operating system versions, iOS operates within a more controlled and uniform ecosystem. This difference significantly influences the approach OMH takes in interfacing with iOS, particularly emphasizing the importance of cross-platform development compatibility over native integration.

Cross-Platform Focus (Level 2)

Given the streamlined nature of iOS's native development environment, OMH's interface with iOS is primarily concentrated at the cross-platform level, or Level 2. This focus is strategic, aiming to empower developers to efficiently deploy applications across multiple platforms, including iOS, without the need for platform-specific code. By leveraging OMH's unified interface, developers can create applications using cross-platform frameworks that can run on iOS, alongside other platforms, with minimal adjustments. This approach significantly reduces development time and resources, enabling a broader reach across different user bases with less effort.

Implications for Native iOS Development

While OMH's interface offers substantial benefits for cross-platform development, it's important to acknowledge that native iOS developers might not experience the same unified interface and level of integration as their cross-platform counterparts. The absence of a Level 1 common interface for native development on iOS means that developers working directly with Swift or Objective-C won't have access to OMH's unified APIs and services, which are designed to bridge native development gaps across different operating systems.

Strategic Focus

The decision to concentrate OMH's efforts on the cross-platform level stems from a pragmatic assessment of resource allocation and the desire to maximize impact. By focusing on enhancing the cross-platform development experience, OMH aims to address the needs of a significant portion of the developer community looking to build versatile and scalable applications. This strategy acknowledges the current trends in app development, where cross-platform solutions are increasingly favored for their efficiency and broader market reach.

Conclusion

In summary, the interface between iOS and OMH is strategically designed to enhance the cross-platform app development experience, recognizing the unique characteristics of the iOS ecosystem. While this approach offers significant advantages for developers aiming to deploy applications across multiple platforms, it does mean that native iOS app developers will not benefit from a unified OMH interface at the native level. Nonetheless, OMH's focus on cross-platform compatibility represents a forward-thinking approach to app development, aiming to streamline processes and reduce fragmentation across the mobile app landscape.

Open Mobile Hub Success Criteria

Benefits for App Developers

Improved Productivity and Reduced Maintenance

Open Mobile Hub simplifies the app development landscape by providing a unified interface that diminishes the need to maintain multiple app codebases. Traditionally, developers grapple with numerous SDKs that, while offering the same functionality, demand adaptation to their proprietary interfaces. By consolidating these into a single, universal framework, OMH drastically reduces the number of codebases a developer must manage. Moreover, testing is streamlined, as a single test suite that was originally developed for one provider becomes applicable across all integrated providers. This not only reduces the effort and resources required for testing but also speeds up the time to market, enabling quicker responses to market changes and opportunities.

Expanded Choices and Enhanced Innovation

Open Mobile Hub introduces greater flexibility and choice for developers. With the ability to easily swap components, developers can experiment and optimize their apps without extensive overhauls—enhancing adaptability and encouraging innovation. This openness fosters a competitive environment that leverages the best technologies, irrespective of their origin. As competition increases, so does the drive for innovation, resulting in higher-quality products that stand out in the marketplace. The result is a higher caliber of apps, improved user satisfaction, and potentially, a more robust revenue stream.

Cost Efficiency

The adoption of OMH significantly reduces development and ongoing support costs. The streamlined approach minimizes the need for different interfaces and SDKs specific to certain regions or devices, which traditionally inflate costs, especially for small companies or startups. This cost reduction makes it feasible for smaller players to compete on a larger scale and invest more in product development rather than maintenance.

Access to Broader Markets

Open Mobile Hub’s unified approach breaks down barriers to entry into diverse device markets. Developers can now deploy their apps across a wide array of devices without the need for rewriting or specific adaptations for regional SDKs. This is particularly advantageous for developers aiming to expand internationally, where regions may have different device preferences and technological standards. Open Mobile Hub’s flexibility and wide support mean developers can reach more users across the globe, all while exerting less effort on localization and compliance, thus opening up new revenue streams and user bases.

In conclusion, OMH offers substantial benefits for app developers by reducing maintenance burdens, providing greater flexibility, lowering costs, and facilitating access to a broader market. These advantages make OMH an invaluable tool for developers looking to enhance their operational efficiency, expand their market reach, and deliver superior products to a diverse user base.

How To Measure the Success of OMH?

The first aspect of measuring success in an open source SDK framework such as OMH is the level of developers' participation and influence in the OMH-related open source community. One way of measuring this is by analyzing the number of OMH integrations done by mobile app developers. The organization's reputation within open source communities is another key metric. A good reputation can be an indicator of the organization's commitment to open source principles, such as transparency, collaboration, and meritocracy. Additionally, the ability to attract and retain talented developers is a testament to the quality of an open source project.

Another success criterion for the OMH project is related to developers' contributions, which can be assessed through such factors as the number of active contributors, the frequency of updates, and the responsiveness to issues that the community raises. License compliance is also crucial in the open source world. The OMH project must ensure it manages open source licenses properly to avoid legal issues and maintain good relations with the open source community.

The value creation of the open source ecosystem is a broader measure of success. One can measure value creation by the acceleration of innovation at scale, as shown by the number of apps that the OMH open source framework enables, helping them solve UX consistency issues on different mobile ecosystems. The specific goals and metrics for success may vary depending on the unique strategy.

Who Is the Target Customer of OMH?

Open Mobile Hub's main target customers are mobile app developers, so they can integrate their apps with the OMH framework and easily support multiple mobile ecosystems with one single codebase, including support for Google Android, Apple iOS, and HarmonyOS Next. A unified interface in OMH provides a consistent way of interacting with various services. This significantly reduces the learning curve for developers, allowing them to understand and start using the framework more quickly. It also ensures predictable software behavior across different services, leading to fewer errors and more efficient development.

Additionally, OMH makes it easy to swap service providers, which gives developers the flexibility to choose the services that best meet their needs. This can lead to cost savings, improved performance, and better alignment with project requirements. It also reduces the risk of vendor lock-in, giving developers the freedom to change service providers as their needs evolve.

Open Mobile Hub comes with extensive documentation, pre-built libraries, and plugins for integration with various services. This can significantly speed up the integration process, allowing developers to focus more on creating unique value for their projects rather than on the nuts and bolts of integration. Further, a community of developers who are passionate about the technology and are constantly pushing the boundaries of what is possible has built OMH. This can expose developers to new ideas and approaches, helping them to stay current with the latest trends and technologies.

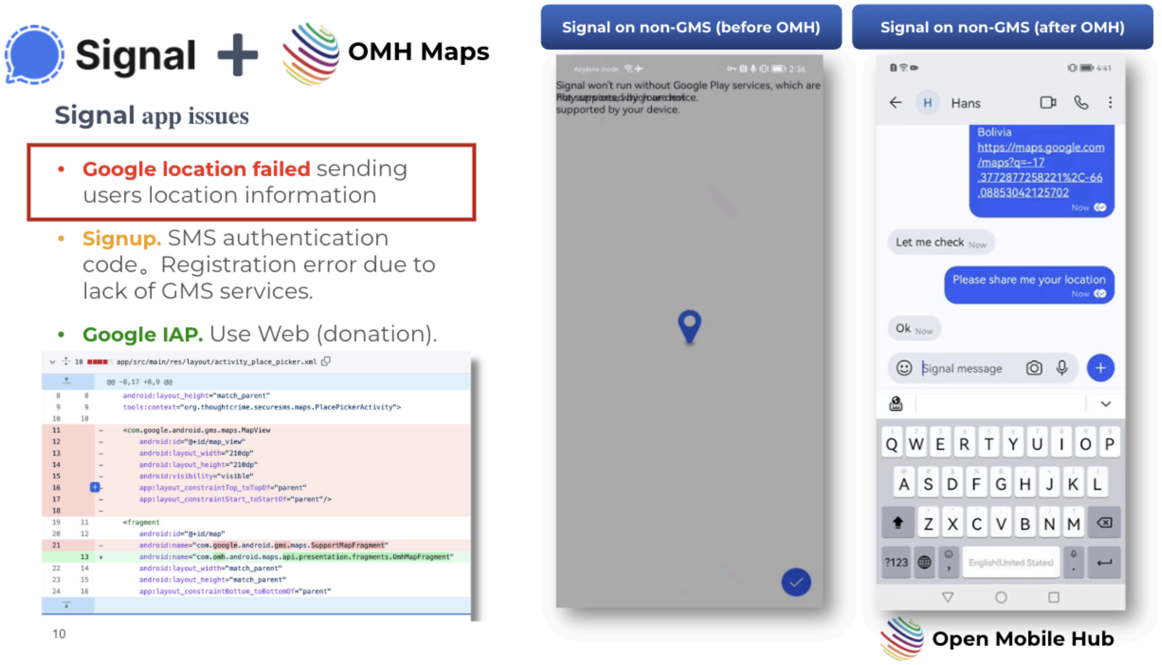
These are the major benefits of the OMH framework, and below are two mobile apps that are integrated with OMH for the same benefits: Signal Messaging and Wordpress Mobile App.

Signal Integration

Signal is a secure, free, and open source messaging application that uses end-to-end encryption to securely send and receive all kinds of communications with other Signal users. All communications on Signal, including one-to-one messages, group messages, file transfers, photos, voice calls, and video calls, are end-to-end encrypted. This means that not even the owners of Signal can monitor them; only the people in the conversation can see them. Signal is completely open source. The source code for the project's client apps and server software is available on GitHub.

Signal offers self-destructing (disappearing) messages that will automatically be removed after a specified period. Signal is available for Android, iPhone, iPad, Windows, Mac, and Linux. The user experience of Signal is just like WhatsApp, Facebook Messenger, and other popular chat apps. It's a messaging app with features such as one-to-one messages, groups, stickers, photos, file transfers, voice calls, and video calls.

Open Mobile Hub integration with the Signal app solves the Google Maps location problem regarding running on non-GMS devices. The use of OMH provides a single SDK framework that allows Signal app users to share location information with others, regardless of whether the app is running on a GMS or non-GMS device.



Signal + OMH Maps

Signal app issues

- **Google location failed** sending users location information
- **Signup.** SMS authentication code。Registration error due to lack of GMS services.
- **Google IAP.** Use Web (donation).

```
app/src/main/res/layout/activity_place_picker.xml
@@ -8,17 +8,9 @@
        android:layout_height="match_parent"
        tools:context="org.thoughtcrime.securesms.maps.PlacePickerActivity">

-       <com.google.android.gms.maps.MapView
-           android:id="@+id/map_view"
-           android:layout_width="210dp"
-           android:layout_height="210dp"
-           android:visibility="visible"
-           app:layout_constraintTop_toTopOf="parent"
-           app:layout_constraintStart_toStartOf="parent"/>
-
        <fragment
            android:id="@+id/map"
-           android:name="com.google.android.gms.maps.SupportMapFragment"
+           android:name="com.omh.android.maps.api.presentation.fragments.OmhMapFragment"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            app:layout_constraintBottom_toBottomOf="parent"
```

10

Signal on non-GMS (before OMH)

Signal won't run without Google Play services, which are supported by your device.

Signal on non-GMS (after OMH)

Hans

Bolivia https://maps.google.com/maps?q=-17.3772877258221%2C-66.08853042125702 Now

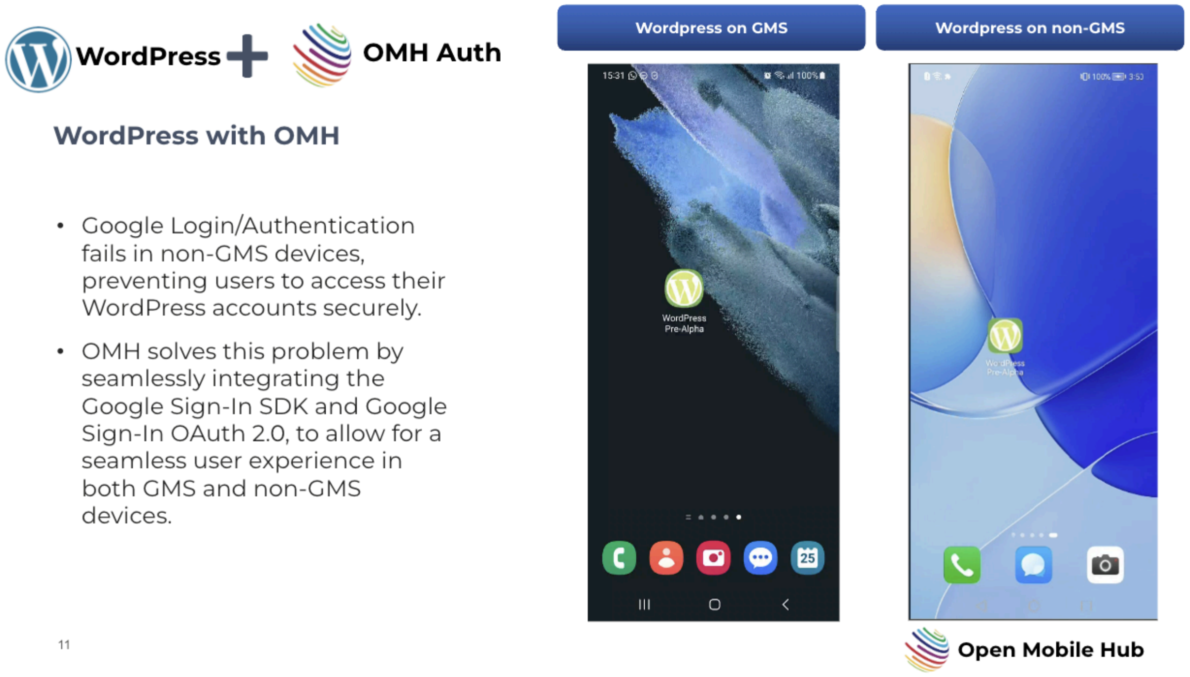
Let me check Now

Please share me your location Now

Ok Now

WordPress Integration

The WordPress Mobile App, developed and maintained by Automattic, allows users to access and work on WordPress sites from iOS and Android phones. It is a website builder and a blog maker that lets them pick the right look and feel from a wide selection of WordPress themes, and then customize it with photos, colors, and fonts to tailor it to their preferences. The app is available for free.



WordPress + OMH Auth

WordPress with OMH

- Google Login/Authentication fails in non-GMS devices, preventing users to access their WordPress accounts securely.
- OMH solves this problem by seamlessly integrating the Google Sign-In SDK and Google Sign-In OAuth 2.0, to allow for a seamless user experience in both GMS and non-GMS devices.

Wordpress on GMS

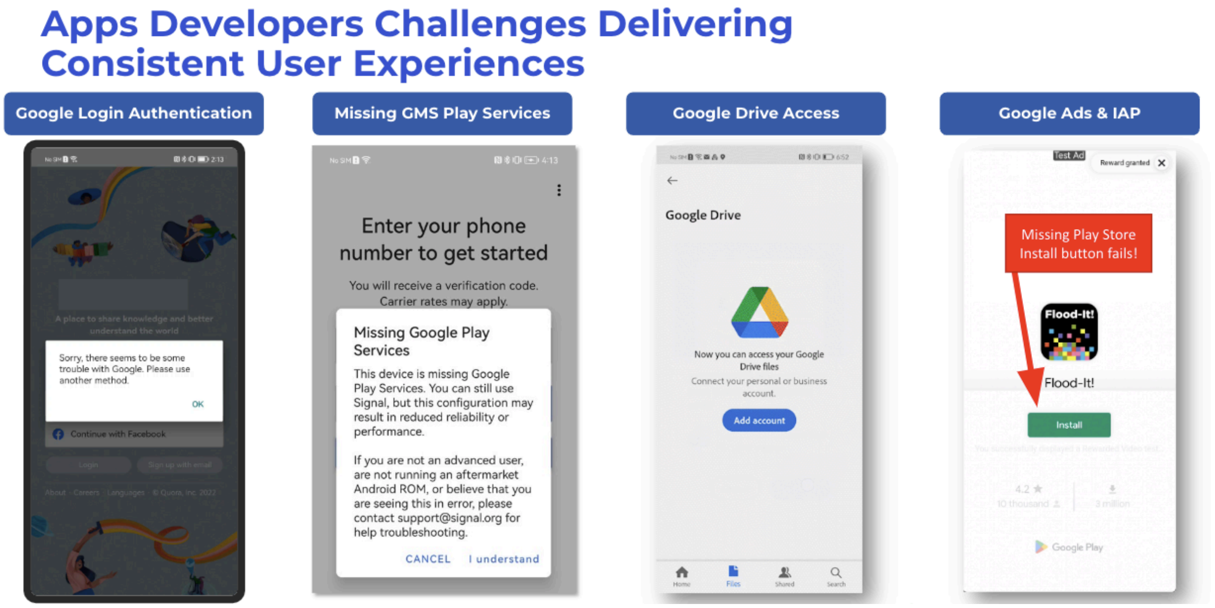
Wordpress on non-GMS

11

Open Mobile Hub

The main benefit of the app is that users no longer require access to a computer to create and publish content. Plus, they might have other useful mobile services such as email apps and photo editing software active. Therefore, they can easily switch between these platforms, which improves efficiency and productivity.

What Are the Problems That the Framework Can Solve for Developers?



Apps Developers Challenges Delivering Consistent User Experiences

Sorry, there seems to be some trouble with Google. Please use another method.

Enter your phone number to get started

You will receive a verification code. Carrier rates may apply.

Missing Google Play Services

This device is missing Google Play Services. You can still use Signal, but this configuration may result in reduced reliability or performance.

If you are not an advanced user, are not running an aftermarket Android ROM, or believe that you are seeing this in error, please contact support@signal.org for help troubleshooting.

Google Drive

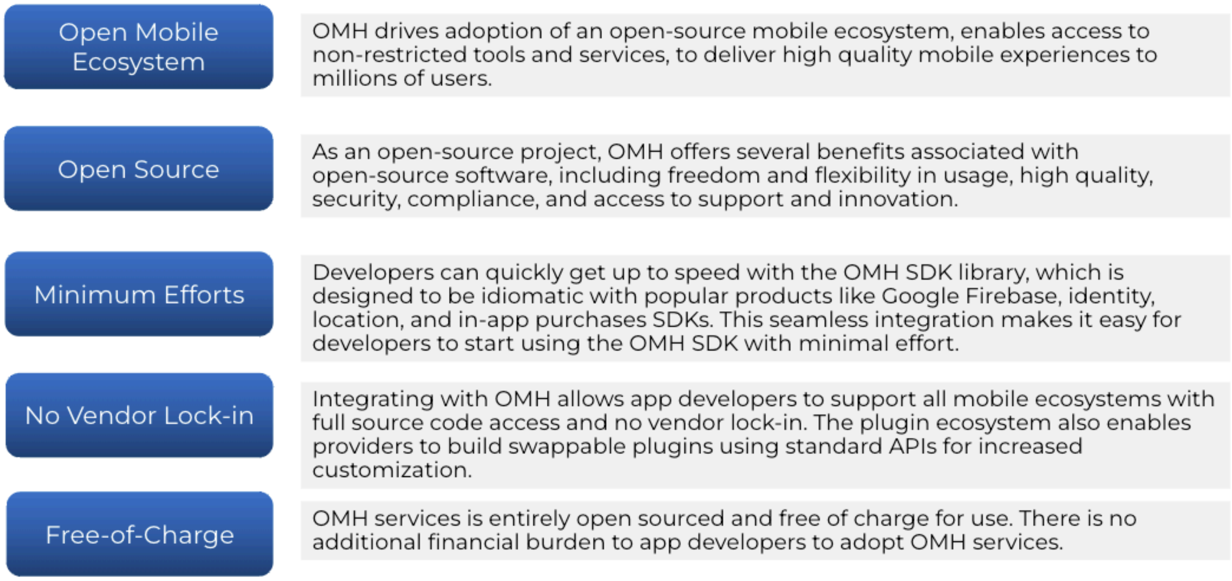
Now you can access your Google Drive files

Connect your personal or business account.

The OMH framework can provide a consistent UX to support multiple ecosystems. By allowing app developers to use their existing knowledge and tools to target non-GMS in addition to Android and iOS, OMH can significantly lower the barrier to entry for app development on its platform. This approach not only accelerates the adoption of OMH among developers but also enriches the platform with a diverse range of apps faster than if developers were required to learn an entirely new stack.

While there are a lot of mobile app frameworks available today that support both Android and iOS platforms, e.g., React Native, Flutter, and NativeScript, there are very limited options available that can also support non-GMS ecosystems. Open Mobile Hub is the only open source SDK framework that can support all three mobile ecosystems (GMS and non-GMS).

What Are the Main Benefits for Developers?



Open Mobile Ecosystem	OMH drives adoption of an open-source mobile ecosystem, enables access to non-restricted tools and services, to deliver high quality mobile experiences to millions of users.
Open Source	As an open-source project, OMH offers several benefits associated with open-source software, including freedom and flexibility in usage, high quality, security, compliance, and access to support and innovation.
Minimum Efforts	Developers can quickly get up to speed with the OMH SDK library, which is designed to be idiomatic with popular products like Google Firebase, identity, location, and in-app purchases SDKs. This seamless integration makes it easy for developers to start using the OMH SDK with minimal effort.
No Vendor Lock-in	Integrating with OMH allows app developers to support all mobile ecosystems with full source code access and no vendor lock-in. The plugin ecosystem also enables providers to build swappable plugins using standard APIs for increased customization.
Free-of-Charge	OMH services is entirely open sourced and free of charge for use. There is no additional financial burden to app developers to adopt OMH services.

Additionally, here are the primary technical objectives and benefits of OMH:

Unified interface: Open Mobile Hub provides a single, standardized API interface for application developers. This means that instead of having to learn and integrate multiple APIs for similar services from different providers, developers can code against one OMH interface. This drastically reduces the learning curve and integration efforts.

Provider agnosticism: By abstracting the specifics of each service provider’s SDK, OMH enables applications to be less dependent on any single provider’s technology. This agnosticism allows for easier swapping of service providers with minimal to no changes in the application codebase, facilitating better negotiation leverage, cost efficiencies, and resilience against service discontinuation or degradation.

Rapid integration and time to market: Since OMH standardizes access to services across providers, developers can integrate new services more quickly and bring their products to market faster. This is particularly beneficial in a competitive landscape where speed to market can be a critical success factor.

Innovation through service composition: Open Mobile Hub’s unified interface encourages innovation by making it easier to combine services from different providers. Developers can leverage the best-of-breed services, mixing and matching functionalities as needed to create unique and powerful application features.

Cross-platform support and filing service gaps: Open Mobile Hub can help address the fragmentation in the Android ecosystem, particularly in regions or devices where Google Play Services are not available. By offering a common layer through which services can be accessed, OMH can facilitate the provision of alternative services that fill these gaps, ensuring a more uniform application experience across devices and regions.

Why Would They Care About the Benefits?

1. **Code reusability**: Cross-platform frameworks such as OMH allow developers to write code once in a single language and then compile it to run natively on multiple platforms. This means that instead of having to write separate codebases in different languages, developers can reuse much of the same code across different platforms. This can significantly reduce the amount of time and effort required for development.

2. **Cost efficiency**: Developing separate apps for each platform can be expensive, as it requires separate teams with different skill sets. With OMH cross-platform development, app developers can leverage a single framework to develop for all platforms, which can lead to lower development costs. Additionally, maintaining and updating a single codebase can be more cost-effective than maintaining multiple codebases.

3. **Uniformity and consistency**: Open Mobile Hub cross-platform frameworks provide a consistent set of UI components that look and behave the same way across different platforms. This ensures that the app provides a consistent user experience, regardless of the platform it’s running on. It also makes it easier to implement and maintain the app’s UI, as developers only need to do it once.

4. **Faster time to market**: Since developers are developing for all platforms simultaneously using OMH, they can release their apps on all platforms at the same time. This can give developers a competitive advantage, as it allows them to reach users on all platforms more quickly.

5. **Easier updates**: When developers need to update their app or fix bugs, they only need to do it in one place. Once they update the codebase, they can compile and deploy the updates to all platforms. This is much easier and faster than having to make the same updates in multiple codebases.

6. **Wider reach**: Since OMH-enabled cross-platform apps can run on any platform on the top three mobile ecosystems, namely Google Android, Apple iOS, and HarmonyOS Next, they can potentially reach a larger audience than just supporting one or two of the ecosystems. This can lead to increased user engagement and higher revenue.

Do Developers Have Any Alternatives?

While there are a lot of mobile app frameworks available today that support both Android and iOS platforms, e.g., React Native, Flutter, and NativeScript, there are limited options available that can also support non-GMS ecosystems. Open Mobile Hub is the only open source SDK framework that will support all three mobile ecosystems. Here are some alternative frameworks that can currently support both Android and iOS:

1. **React Native**: React Native is a cross-platform framework developed by Facebook. It allows developers to write code in JavaScript and render it using native components. This means that instead of having to write separate codebases in different languages (such as Java for Android and Swift for iOS), developers can reuse much of the same code across different platforms. This can significantly reduce the amount of time and effort required for development.

2. **Flutter**: Flutter is Google's SDK for crafting high-quality, native interfaces on iOS and Android. It allows developers to write code once in a single language (Dart) and then compile it to run natively on multiple platforms. Flutter is designed as an extensible layered system. It exists as a series of independent libraries that each depend on the underlying layer. No layer has privileged access to the layer below, and every part of the framework level is designed to be optional and replaceable. Flutter code compiles to ARM or Intel machine code as well as JavaScript for fast performance on any device.

3. **Ionic**: Ionic is an open source mobile UI toolkit for building performant, high-quality mobile apps using web technologies—HTML, CSS, and JavaScript—with integrations for popular frameworks such as Angular, React, and Vue[1]. Ionic is very useful to web developers as they can turn their web pages into a web app, with native app functionalities easily using Ionic.

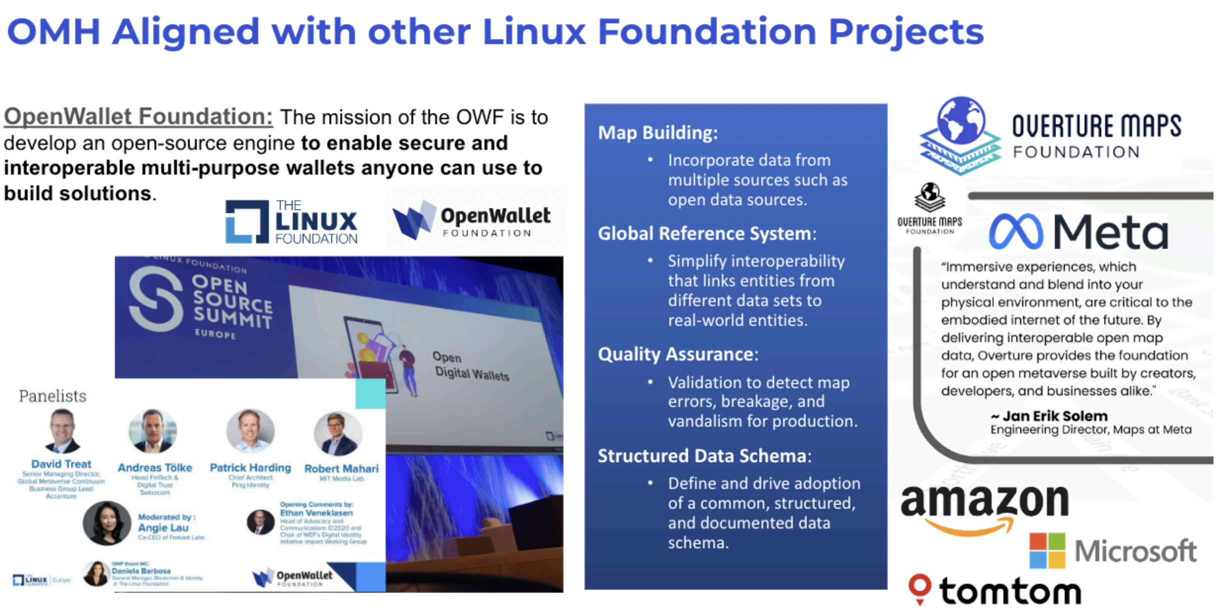
4. **Apache Cordova**: Apache Cordova, formerly known as PhoneGap, is a mobile application development framework created by Nitobi. Adobe Systems purchased Nitobi in 2011, rebranded it as PhoneGap, and later released an open source version of the software called Apache Cordova. Apache Cordova enables software programmers to build hybrid web applications for mobile devices using CSS3, HTML5, and JavaScript instead of relying on platform-specific APIs such as those in Android, iOS, or Windows Phone[2]. It enables the wrapping up of CSS, HTML, and JavaScript code, depending on the device's platform.

5. **jQuery Mobile**: jQuery Mobile is a touch-optimized web framework, specifically a JavaScript library, developed by the jQuery project team. The development focuses on creating a framework compatible with many smartphones and tablet computers, which is necessary for the growing but heterogeneous tablet and smartphone market. jQuery

Mobile is an HTML5-based user interface system designed to make responsive websites and apps that are accessible on all smartphone, tablet, and desktop devices.

What Can the Linux Foundation Europe Do to Help the Mobile Ecosystem Through the OMH Project?

Open Mobile Hub is a Linux Foundation Europe hosted project. Linux Foundation Europe supports the project with development infrastructure, marketing, promotion, research, legal, governance, and business development. Linux Foundation Europe has no development or coding resources and relies on the project team and open source community to contribute actual source code to support the community development of these projects.



OMH Aligned with other Linux Foundation Projects

OpenWallet Foundation: The mission of the OWF is to develop an open-source engine **to enable secure and interoperable multi-purpose wallets anyone can use to build solutions**.

Panelists

David Treat, Senior Managing Director, Global Metaverse Continuum Business Group Lead, Accenture

Andreas Tölke, Head FinTech & Digital Trust, Swisscom

Patrick Harding, Chief Architect, Ping Identity

Robert Mahari, MIT Media Lab

Moderated by : Angie Lau, Co-CEO of Forkast Labs

Opening Comments by: Ethan Veneklasen, Head of Advocacy and Communications ID2020 and Chair of WEF's Digital Identity Initiative Impact Working Group

OWF Event MC: Daniela Barbosa, General Manager, Blockchain & Identity @ The Linux Foundation

Map Building:

- Incorporate data from multiple sources such as open data sources.

Global Reference System:

- Simplify interoperability that links entities from different data sets to real-world entities.

Quality Assurance:

- Validation to detect map errors, breakage, and vandalism for production.

Structured Data Schema:

- Define and drive adoption of a common, structured, and documented data schema.

"Immersive experiences, which understand and blend into your physical environment, are critical to the embodied internet of the future. By delivering interoperable open map data, Overture provides the foundation for an open metaverse built by creators, developers, and businesses alike."

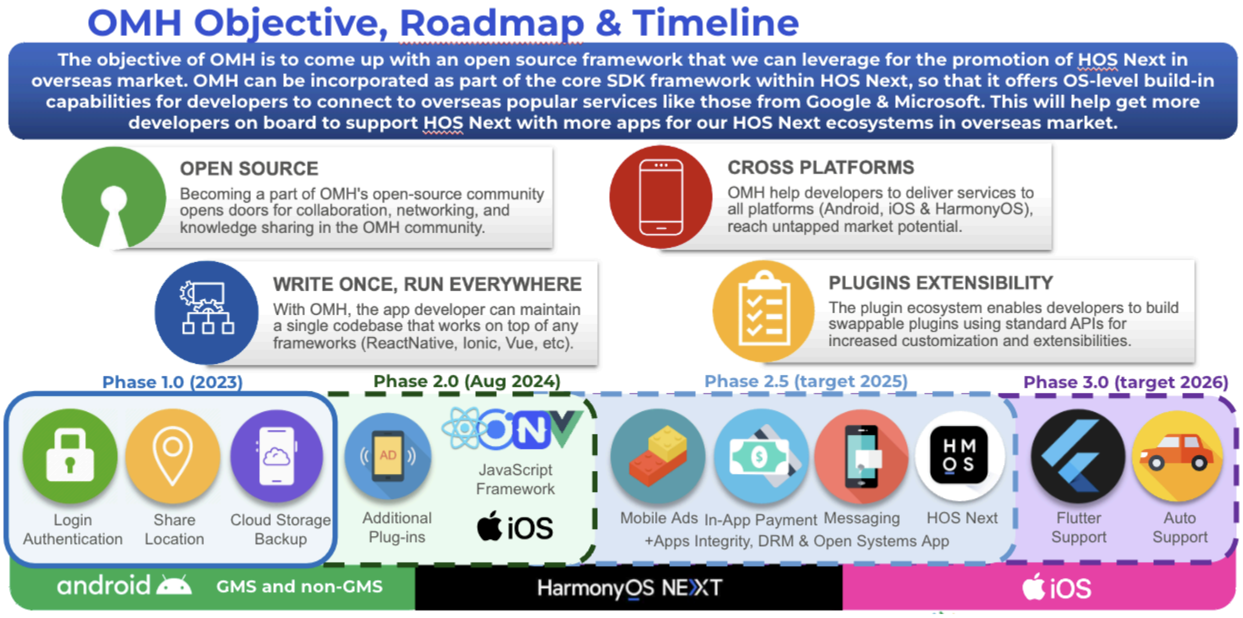
~ Jan Erik Solem
Engineering Director, Maps at Meta

Building up a contributor community and gathering support from other companies around Open Mobile Hub is most important for further adoption, , which is especially true for top companies such as Google, Microsoft, and Amazon. The LF uses a success formula it used in many of its open source projects such as the Open Wallet Foundation, which eventually got sponsorship support from Google and Microsoft (reportedly Apple also expressed interest) after spending 1.5 to two years building the foundation of the community development.
In addition to the Open Wallet Foundation, we are also discussing another Linux Foundation-hosted project, the Overture Maps Foundation. It offers open data for global maps data, which we could use to offer additional maps services using this map data for integration with their apps. Its mission is to build reliable, easy-to-use, and interoperable open map data. The foundation believes in harnessing the collective intelligence and resources of communities, organizations, and governments worldwide and envisions a world where the best maps are not

proprietary assets but shared, open resources that the very people who rely on them build and enrich.

Open Mobile Hub 1.0-2.0 Project Progress

Open Mobile Hub Roadmap and Timeline Overview



OMH Objective, Roadmap & Timeline

The objective of OMH is to come up with an open source framework that we can leverage for the promotion of HOS Next in overseas market. OMH can be incorporated as part of the core SDK framework within HOS Next, so that it offers OS-level build-in capabilities for developers to connect to overseas popular services like those from Google & Microsoft. This will help get more developers on board to support HOS Next with more apps for our HOS Next ecosystems in overseas market.

OPEN SOURCE

Becoming a part of OMH's open-source community opens doors for collaboration, networking, and knowledge sharing in the OMH community.

CROSS PLATFORMS

OMH help developers to deliver services to all platforms (Android, iOS & HarmonyOS), reach untapped market potential.

WRITE ONCE, RUN EVERYWHERE

With OMH, the app developer can maintain a single codebase that works on top of any frameworks (ReactNative, Ionic, Vue, etc).

PLUGINS EXTENSIBILITY

The plugin ecosystem enables developers to build swappable plugins using standard APIs for increased customization and extensibilities.

Open Mobile Hub 1.0 Features and Release

The first release of the OMH SDK (year 2023) introduces support for three client libraries and a Gradle plugin. As we continue to develop OMH, future releases will bring even more client libraries, expanding the range of services available:

1. **Login and Authentication SDK**
   OMH Auth is a client library that makes it easy to integrate auth providers on all types of devices running different OS platforms. It eliminates the need for separate codebases for different app builds.

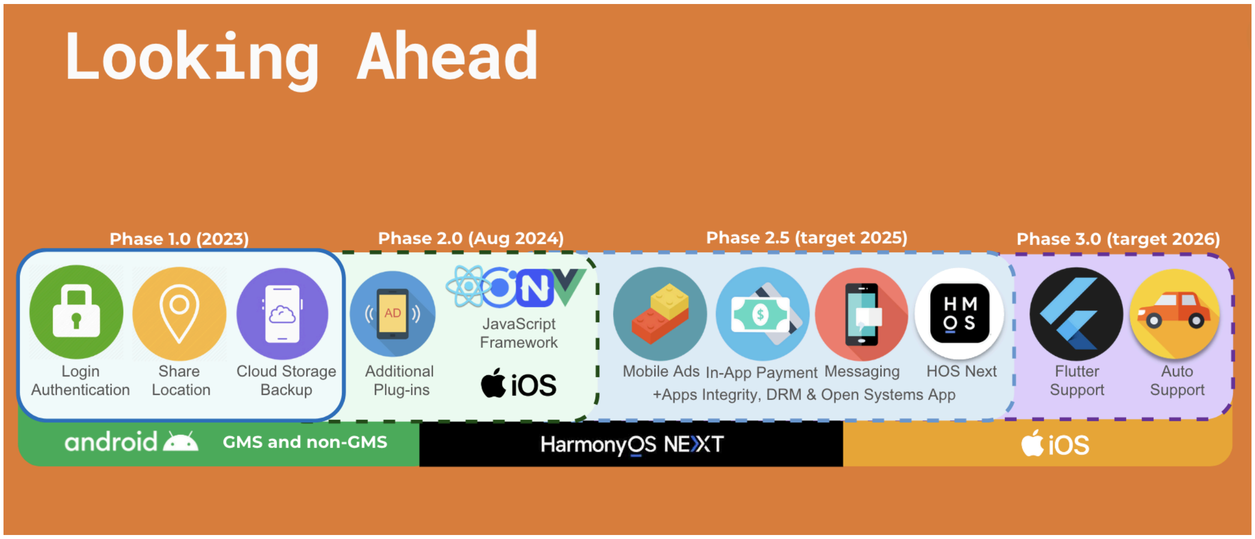
2. **Maps & Location SDK**
   OMH Maps is a client library that makes it easy to integrate maps on all types of devices running different OS platforms. It eliminates the need for separate codebases for different app builds.

3. **Storage SDK**
   OMH Storage Client Library allows users to support file management and app backup services seamlessly, regardless of the OS platform, on all types of devices.

4. **Core Gradle Plugin**
   OMH Core is an essential Gradle plugin designed to streamline the configuration and setup of OMH Client Libraries within developers' projects. With this plugin, developers can effortlessly incorporate the necessary dependencies and activate custom-build variants, facilitating the compilation of various builds that leverage the defined providers.

Open Mobile Hub 2.0 Features and Release

Open Mobile Hub 2.0 is also compatible with the React Native framework, which will form the foundation for the OMH project to support multiple mobile ecosystems, including Google Android, Apple iOS, and HarmonyOS Next.



Looking Ahead

Phase 1.0 (2023)
Login Authentication
Share Location
Cloud Storage Backup

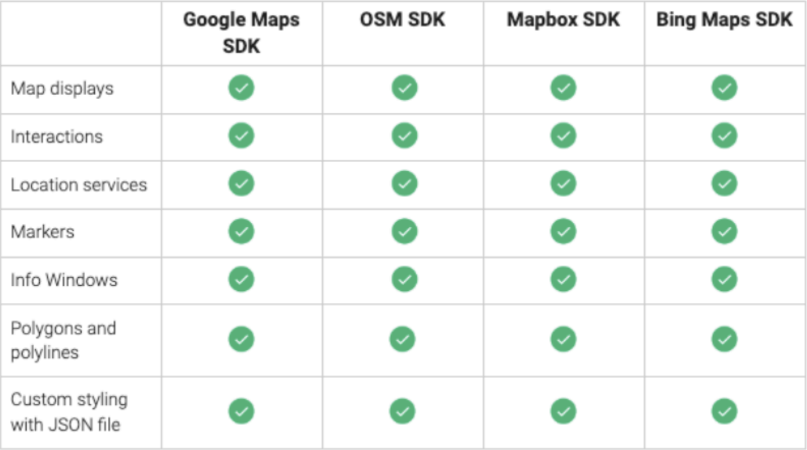
Phase 2.0 (Aug 2024)
Additional Plug-ins
JavaScript Framework
iOS

Phase 2.5 (target 2025)
Mobile Ads
In-App Payment
Messaging
HOS Next
+Apps Integrity, DRM & Open Systems App

Phase 3.0 (target 2026)
Flutter Support
Auto Support

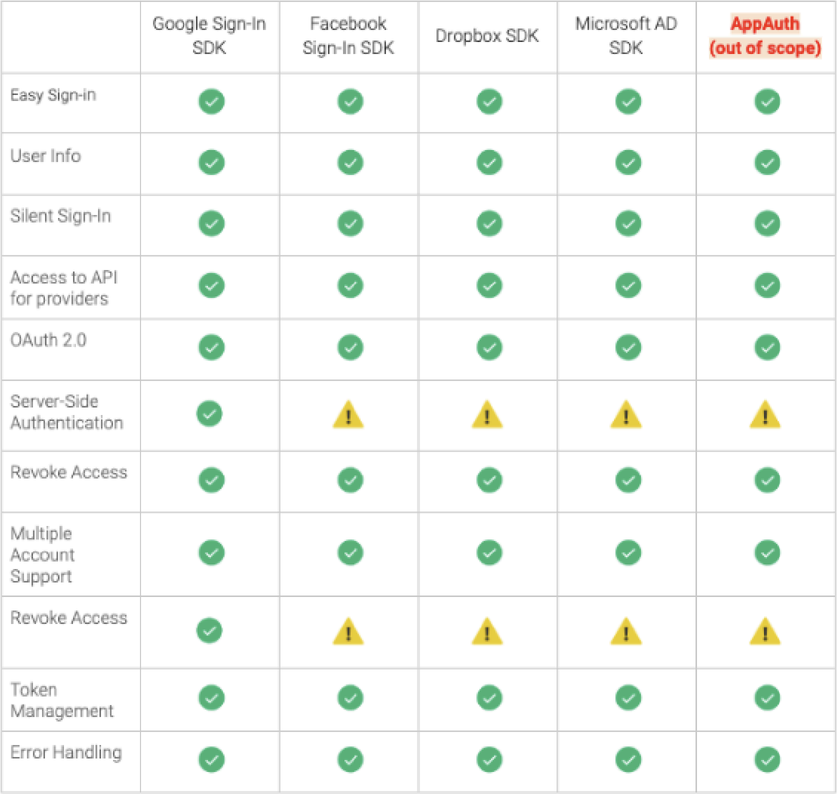
android GMS and non-GMS
HarmonyOS NEXT
iOS

Open Mobile Hub 2.0 Maps Features



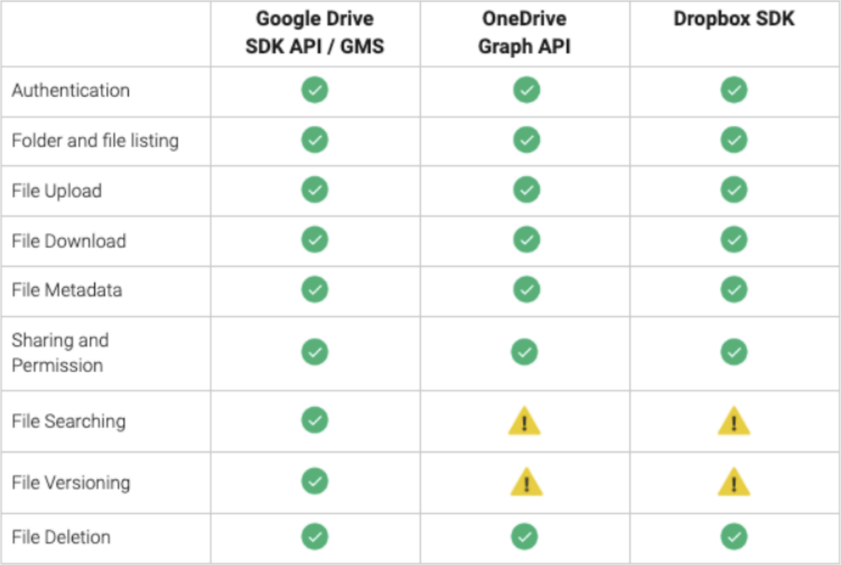
	Google Maps SDK	OSM SDK	Mapbox SDK	Bing Maps SDK
Map displays	✓	✓	✓	✓
Interactions	✓	✓	✓	✓
Location services	✓	✓	✓	✓
Markers	✓	✓	✓	✓
Info Windows	✓	✓	✓	✓
Polygons and polylines	✓	✓	✓	✓
Custom styling with JSON file	✓	✓	✓	✓

Open Mobile Hub 2.0 Auth Features



	Google Sign-In SDK	Facebook Sign-In SDK	Dropbox SDK	Microsoft AD SDK	AppAuth (out of scope)
Easy Sign-in	✓	✓	✓	✓	✓
User Info	✓	✓	✓	✓	✓
Silent Sign-In	✓	✓	✓	✓	✓
Access to API for providers	✓	✓	✓	✓	✓
OAuth 2.0	✓	✓	✓	✓	✓
Server-Side Authentication	✓	⚠	⚠	⚠	⚠
Revoke Access	✓	✓	✓	✓	✓
Multiple Account Support	✓	✓	✓	✓	✓
Revoke Access	✓	⚠	⚠	⚠	⚠
Token Management	✓	✓	✓	✓	✓
Error Handling	✓	✓	✓	✓	✓

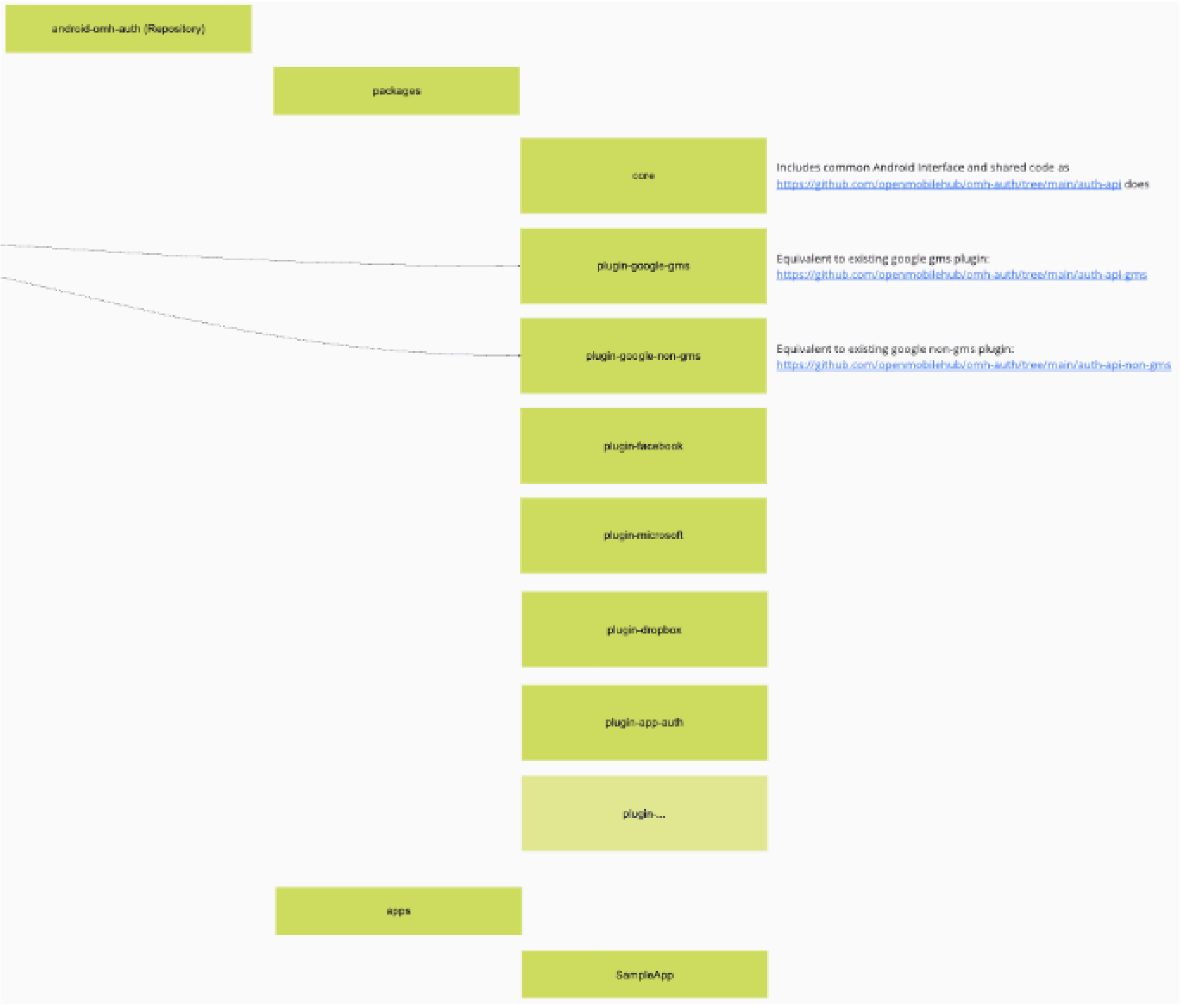
Open Mobile Hub 2.0 Cloud Storage Features



	Google Drive SDK API / GMS	**OneDrive Graph API**	**Dropbox SDK**
Authentication	✓	✓	✓
Folder and file listing	✓	✓	✓
File Upload	✓	✓	✓
File Download	✓	✓	✓
File Metadata	✓	✓	✓
Sharing and Permission	✓	✓	✓
File Searching	✓	⚠	⚠
File Versioning	✓	⚠	⚠
File Deletion	✓	✓	✓

Open Mobile Hub 2.0 Kotlin/Java Native Repo Structure

- Mono-repo structure
- Easy to manage. Reduced complexity over managing multiple repos
- Independent modules to support third-party plugins



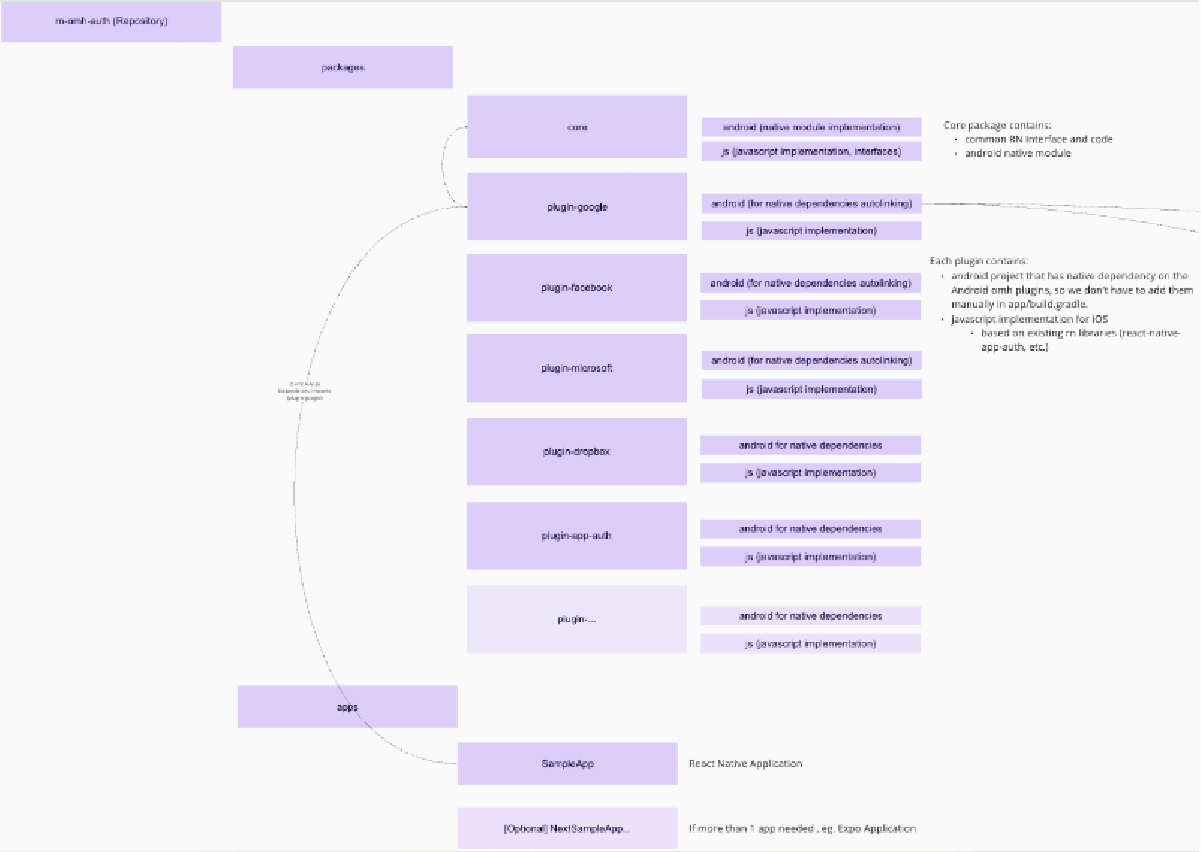
Includes common Android Interface and shared code as https://github.com/openmobilehub/omh-auth/tree/main/auth-api does

Equivalent to existing google gms plugin: https://github.com/openmobilehub/omh-auth/tree/main/auth-api-gms

Equivalent to existing google non-gms plugin: https://github.com/openmobilehub/omh-auth/tree/main/auth-api-non-gms

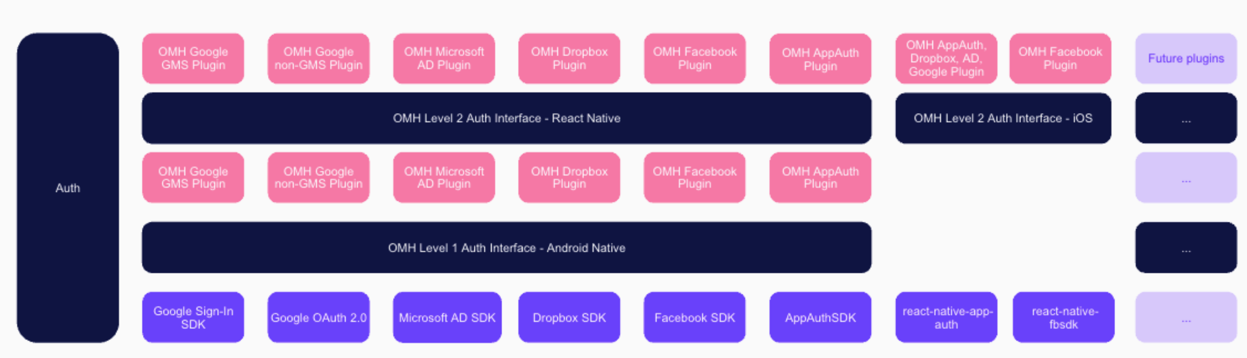
Open Mobile Hub 2.0 React Native Repo Structure

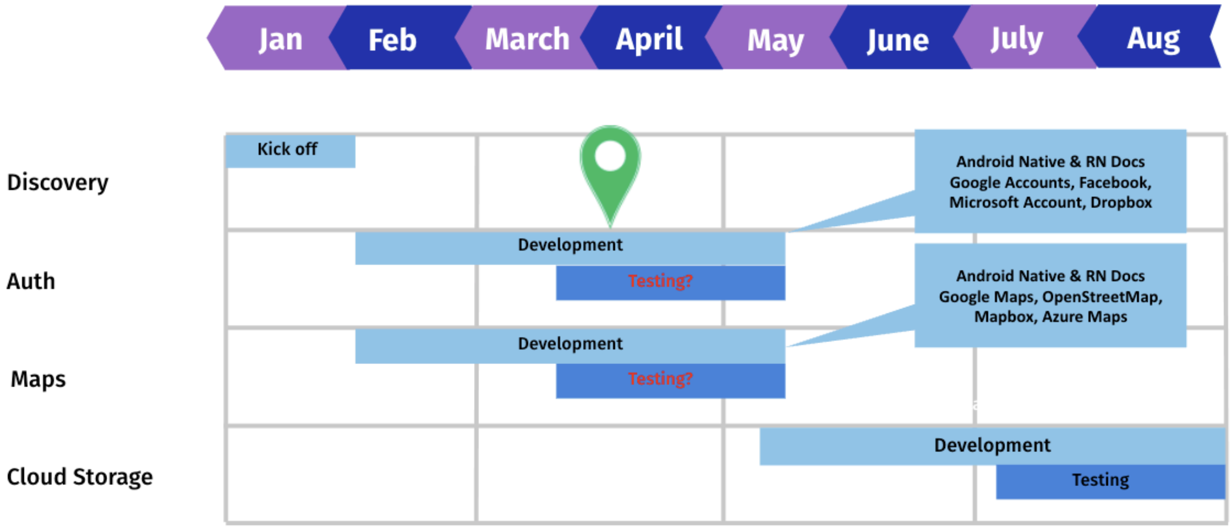
- Mono-repo structure
- Easy to manage. Reduced complexity over managing multiple repos
- Independent modules to support third-party plugins


rn-omh-auth (Repository)
packages
core
android (native module implementation)
js (javascript implementation, interfaces)
Core package contains:
• common RN Interface and code
• android native module
plugin-google
android (for native dependencies autolinking)
js (javascript implementation)
Each plugin contains:
• android project that has native dependency on the Android omh plugins, so we don't have to add them manually in app/build.gradle.
• javascript implementation for iOS
• based on existing rn libraries (react-native-app-auth, etc.)
plugin-facebook
android (for native dependencies autolinking)
js (javascript implementation)
plugin-microsoft
android (for native dependencies autolinking)
js (javascript implementation)
plugin-dropbox
android for native dependencies
js (javascript implementation)
plugin-app-auth
android for native dependencies
js (javascript implementation)
plugin-...
android for native dependencies
js (javascript implementation)
apps
SampleApp
React Native Application
[Optional] NextSampleApp...
If more than 1 app needed , eg. Expo Application

What’s the Status of the Open Mobile Hub 2.0 Development?

The OMH 2.0 development work started in January 2024 and is estimated to be finished around August 2024. It’s currently on schedule. For project OMH 2.0’s roadmap and timeline, see the latest updates below (as of April 2024):





How Can Developers Get Started Using OMH 2.0?

To get started with OMH 2.0, you can begin by exploring and integrating the specific SDKs designed to facilitate app development with OMH. Below are two useful SDKs available on GitHub that can help you harness the capabilities of OMH in your projects:

OMH Authentication SDK for Android: This SDK provides tools for handling authentication processes within your Android applications using OMH standards. You can find the SDK and its documentation here: OMH Authentication SDK. This resource will guide you through setting up authentication features, ensuring that your app can manage user identities and access controls effectively.

OMH Maps SDK for Android: If your application requires mapping and geographic capabilities, the OMH Maps SDK is ideal. It allows you to integrate map functionalities that are compliant with OMH standards into your Android apps. Check out the SDK and its setup instructions here: OMH Maps SDK. This SDK will help you add interactive maps and location-based services to your applications.

By using these SDKs, you will be able to leverage the standardized features and functionalities of OMH 2.0, enhancing your app development process and product capabilities. Make sure to follow the setup instructions in each repository to properly integrate these tools into your projects.

Are There Any Sample Apps for Open Mobile Hub 2.0?

To find sample apps for OMH 2.0, please check the "Getting Started" section in the GitHub repositories linked above for the OMH Authentication SDK and OMH Maps SDK. These sections often include examples or links to sample applications that can help you get started with integrating OMH into your projects.

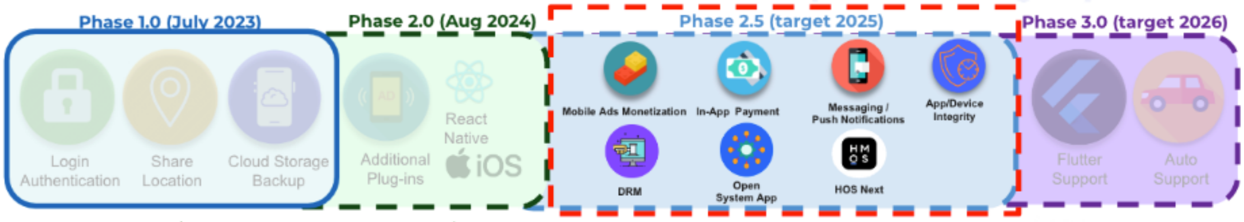
How To Report Bugs/Issues That I Find in Open Mobile Hub 2.0?

To report bugs or issues that you find in OMH 2.0, please raise them through the GitHub Issues section of the relevant OMH repository. This is the most direct way to communicate with the developers and contribute to improvements in the system.

Open Mobile Hub Future Roadmap

Here is the list of preliminary features that we could implement as part of the OMH SDK framework, as well as three future opportunities for OMH, namely the OMH Auto, OMH Health, and OMH CodeGen. These areas are still in the early stages of brainstorming and planning. The details & schedule in the future roadmap need to be finalized by the TSC before implementation.

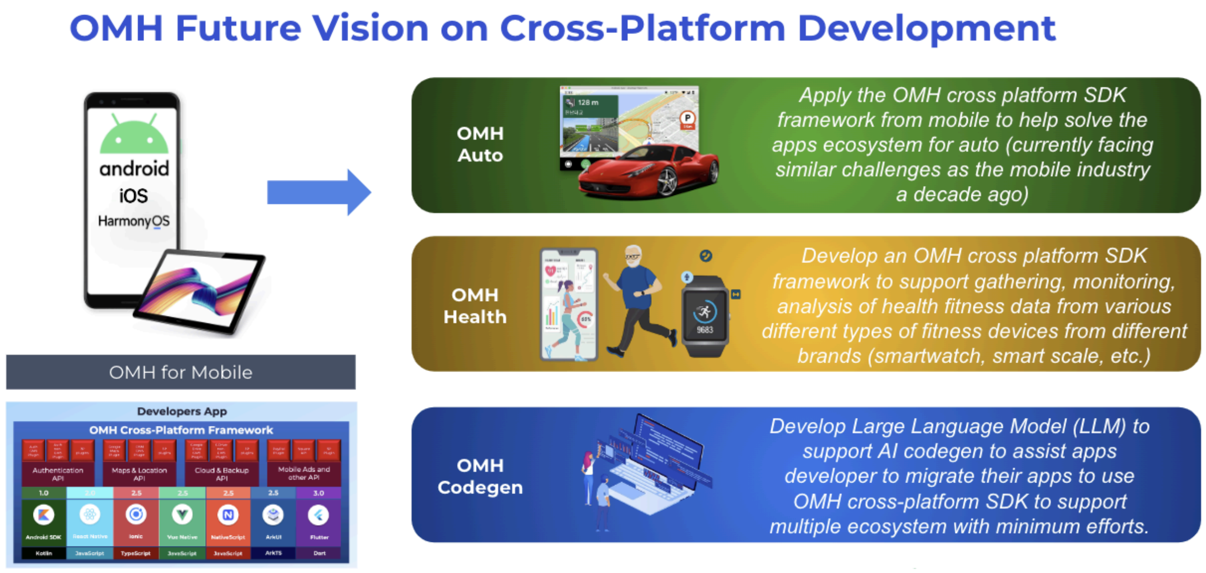
OMH Preliminary Features List



The OMH team needs to conduct comprehensive research based on the OMH future roadmap. Once this research is complete and approved by the TSC, they will need to be validated through a proof of concept (POC). We will then develop a detailed timeline for the implementation of these new features. Here is the list of preliminary features for planning purpose:

1. Expo SDK Framework Support
   - Providing support of the official React Native SDK frameworks with Expo architecture.
2. HarmonyOS Next Cross-Platform Support
   - Extending support to the 3rd mobile operating system HarmonyOS Next, enhancing cross-platform capabilities in addition to iOS and Android.
3. In-App Messaging
   - Integrating messaging features directly within apps.
4. In-App Purchases
   - Enabling seamless in-app purchase functionalities.
5. App/Device Integrity
   - Ensuring robust app and device integrity checks.
6. DRM (Digital Rights Management)
   - Implementing DRM features for content protection.
7. Mobile Ads Monetization SDKs
   - Providing SDKs for mobile ad integration and monetization.
8. Flutter Support
   - Adding Flutter support for cross-platform development using OMH framework.

OMH Future Vision



OMH Future Vision on Cross-Platform Development

OMH Auto

Apply the OMH cross platform SDK framework from mobile to help solve the apps ecosystem for auto (currently facing similar challenges as the mobile industry a decade ago)

OMH Health

Develop an OMH cross platform SDK framework to support gathering, monitoring, analysis of health fitness data from various different types of fitness devices from different brands (smartwatch, smart scale, etc.)

OMH Codegen

Develop Large Language Model (LLM) to support AI codegen to assist apps developer to migrate their apps to use OMH cross-platform SDK to support multiple ecosystem with minimum efforts.

Open Mobile Hub Auto

Apply the OMH cross-platform SDK framework from mobile to help solve the app's ecosystem for auto (currently facing similar challenges to the mobile industry a decade ago). Using an open source SDK framework for the auto app market can offer several benefits:

1. **Improved fragmentation**: Open source SDKs such as OMH help developers build high-performance apps with functional and consistent user interfaces. This could help ease the fragmentation problem that's happening to the auto industry today, similar to what happened to the mobile industry in the early days.

2. **Cross-platform development**: Open source SDKs such as OMH allow for cross-platform mobile application development. This allows auto app developers to use a single codebase to build applications for cars, which can significantly speed up the development process and reduce costs.

3. **Code reusability**: With cross-platform OMH frameworks, app developers can reuse their code across different platforms, i.e., mobile, web, and auto. This means developers don't have to write separate code for each platform, which can save a lot of time and resources.

4. **Faster app development**: Since developers only need to develop features once for all platforms, they can ship new features for auto platforms much faster.

5. **Community support**: Being open source, OMH can offer strong community support to app developers. Developers can benefit from the collective knowledge and contributions of the community, which can include everything from bug fixes to new features.

Open Mobile Hub Health

Develop an OMH cross-platform SDK framework to support the gathering, monitoring, and analysis of health fitness data from various fitness devices from different brands (smartwatch, smart scale, etc.). Here are the key benefits of using OMH to connect smartwatches to different backends for health data consolidation:

1. **Interoperability**: Open source OMH SDK is designed to work with a wide range of devices and platforms. This means it can facilitate the connection of various smartwatch models to different backend systems. For instance, a health app could collect data from an Apple Watch, a Fitbit, and a Samsung Galaxy Watch and then send this data to different backend systems for processing and analysis. This interoperability can greatly enhance the versatility of health apps and allow them to serve a broader user base.

2. **Flexibility and customization**: Open source means that the source code of the OMH SDK is publicly available. This allows developers to modify the code to suit their specific needs. For example, if a health app needs to collect a specific type of data that is not typically supported by the SDK, the developer could modify the SDK to add this functionality. This level of customization can be crucial when dealing with the diverse and complex nature of health data.

3. **Community support**: The open source nature of the OMH SDK means that they are backed by a community of developers who contribute to the software’s development and improvement. This community can be a valuable resource when troubleshooting issues or seeking advice on best practices. Additionally, community members often develop and share plugins or extensions that add new features to the SDK, which can save other developers time and effort.

4. **Data consolidation**: By connecting smartwatches to different backends, health data can be consolidated in one place, making it easier to analyze and interpret. This can be particularly beneficial in the healthcare sector, where patient data is often scattered across various systems. By consolidating this data, healthcare providers can gain a more comprehensive view of a patient’s health, which can lead to more accurate diagnoses and more effective treatment plans.

Open Mobile Hub CodeGen

Users can develop large language models to support AI CodeGen to assist app developers in migrating their mobile apps to use OMH cross-platform SDK to support multiple ecosystems with minimum effort.

AI CodeGen refers to the process of using AI techniques to automatically generate code. This can encompass various aspects of the software development lifecycle, from initial prototyping to optimization and maintenance. AI CodeGen has the potential to significantly accelerate the software development process by automating repetitive tasks, assisting developers in writing correct and efficient code, and enabling rapid prototyping of new ideas. There are several approaches and techniques used in AI code generation:

1. **Natural language processing (NLP)**: NLP techniques are used to interpret human-readable descriptions or requirements and convert them into code. This involves parsing and understanding the natural language input and translating it into the appropriate programming constructs.

2. **Machine-learning (ML)** algorithms: ML algorithms can be trained on large codebases, particularly around the specific use of the SDK framework, to learn patterns and relationships between different components of code. They can be used for tasks such as predicting code completions, suggesting fixes for bugs, or automatically generating code snippets based on context.

3. **Neural code generation**: Recent advancements in deep learning have led to the development of neural architectures specifically designed for code generation tasks. These models are trained on large data sets of code and can generate code snippets or even entire programs based on input specifications.

4. **Code transformation**: AI techniques can also be used to automatically refactor or transform existing code that uses a specific SDK framework (e.g., GMS library from Google) to migrate to an open source SDK framework such as OMH. It can improve readability, performance, and maintainability. This can involve tasks such as renaming variables, extracting common patterns into functions or classes, or optimizing code for specific platforms or architectures.