OMA3

[IWPS Request for Proposals]

1. Introduction

This Request for Proposals (RFP) is released by OMA3's Portaling and Mapping Working Group (PMWG[1]). The purpose of this RFP is to solicit specification proposals for the Inter World Portaling System (IWPS). It is also a call for non-members with expertise in this area to join OMA3 and help build IWPS.

IWPS was introduced to the public earlier this year when OMA3 released the IWPS position paper. IWPS is the 3D hyperlink for the metaverse and allows avatars and their items to traverse seamlessly among virtual worlds using a teleportation system. Since the release of the position paper the PMWG has been compiling IWPS requirements to serve as a measuring stick for evaluating IWPS specification proposals. These requirements fall into different categories such as user experience and governance. These requirements form the foundation upon which IWPS aims to build a truly interconnected and user-friendly metaverse.

---

[1] The charter of PMWG and other OMA3 working groups can be found on OMA3's website.

2. Purpose

The purpose of this Request for Proposals (RFP) is to outline the requirements and criteria for proposed Inter-World Portal System (IWPS) specifications. IWPS specifications will serve as a guide for developers, stakeholders, and team members involved in the development, maintenance, and governance of IWPS. As importantly, this document is the vehicle to solicit contributions from the diverse OMA3 community of Creators, Participants and Sponsors.

3. Scope

The Inter-World Portal System (IWPS) aims to revolutionize the way users interact with multiple virtual and digital platforms by facilitating seamless transitions—or "teleportations"—between them. This ambitious system seeks to provide a streamlined experience for users while addressing a myriad of requirements that range from performance and reliability to governance and security. This RFP starts with a high-level "use cases" description of how IWPS will be used. The RFP then outlines high-level threats for circumventing the system. The RFP finally details a comprehensive range of functional and non-functional requirements that IWPS needs to fulfill. The requirements cover latency standards, throughput features, governance models, and permissions policies. They span across performance, reliability, governance, usability, security, and interoperability. Also included are specific requirements related to identity services, payment systems, and asset transfer systems. However, this document does not delve into the architectural or design aspects of the system, as these will come from proposals submitted in response to this RFP. This RFP serves to offer a clear acceptance criteria that can be used to judge such specification proposals.

4. Use Cases

The definition of use cases is an important and valuable step when capturing requirements. Use cases capture the main objectives of a system. A use case driven specification development process allows one to detect and analyze top-level building blocks of a system in order to identify the main actors and determine how such actors are going to interact with the system. An actor specifies a role played by an entity external to the system, that interacts with the system.

A use case can be described as a list of actions which defines the interactions between an actor and the system in order to achieve a specific individual goal or execute a specific activity. A use case may include events, which describe an occurrence or a sudden change in the system's status. Actors can be a human or an external system.

When creating a use case document, use cases and activities are described and analyzed for the following reasons:

- ▲ To derive high-level technical and non-technical requirements.
- ▲ To understand dependencies from external stakeholders and from the external environment.
- ▲ To scope the technical architecture.
- ▲ To describe the main activities.

There are two main components of a use case definition- the system and actors.

System

The System includes all aspects described in a specification or the infrastructure that provides the desired functionality. It may include one or more blockchains, including validators and other servers, and smart contracts. It could also include centralized servers run by OMA3.

Actors

In the context of a UML (Unified Modeling Language) Use Case diagram, an "actor" represents an external entity that interacts with the system being modeled. Actors can be human users, external hardware, or other systems. They are not part of the system itself, but they do interact with it, typically by triggering use cases.

For example, in a banking application:

The "Customer" could be an actor who can perform actions like "Check Balance" or "Transfer Money." An "ATM" could be another actor that interacts with the banking system to "Dispense Cash."

Actors help define the boundary of a system by illustrating how external elements interact with it, making them crucial for understanding and specifying requirements during the development process.

In this RFP, the system of interest is the Inter World Portaling System ("IWPS" or the "System"). It enables an Avatar (and its Items) to teleport from one virtual world to another within the metaverse. The main actors are as follows:

- User- The User is an entity (most likely a human, but it could be machine-based) that operates in real life and operates one or more Avatars in one or more Metaverse Platforms.
- Asset- An avatar or an item, owned by the User or a Platform.
- Avatar- A graphical representation of the User in a Platform. If the Platform is a first-person Platform, the Avatar is the object the Platform uses to identify the User as an entity in the Platform world.
- Item- A thing belonging to the Avatar on the Originating Platform, which the Avatar may bring with it to the Destination Platform.
- Platform- The software that creates a virtual world that users can explore. All the Platforms available to a User defines the Metaverse.
- Originating Platform- the Platform from which the Avatar starts.
- Destination Platform- the Platform the Avatar goes to after the portaling process.
- Destination Location- address of the location in the Destination Platform
- Service- A software that refers to an action of helping or doing work for an individual platform or many platforms.
- Portal UI- A Service that provides a user interface element within a Platform that allows the initiation of the teleportation of an Asset to a Destination Location in the Metaverse.
- Identity Service- A Service that allows a User to log in to a Platform, either manually or automatically.
- Asset Transfer System- A Service that transfers Assets (e.g.- Avatars and Items) between Platforms, and possibly the infrastructure these Platforms are built on (e.g.- blockchains).
- Inter-Platform Messaging System (IPMS)- A Service that sends messages between Platforms.
- Payment System- A Service that allows Users to pay for teleportation fees, gas fees, and Platform subscription fees.
- Search Engine- A Service that allows Actors to list or discover Platforms in the System.
- Data Collector- A Service that collects data (such as portal transactions).
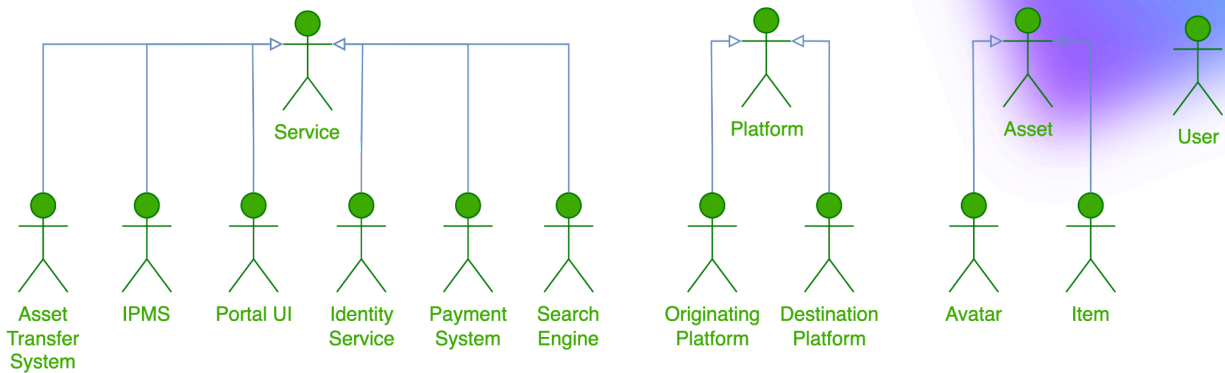
Fig 1. Main actors

Portal Utilization Use Case

1. A User uses the Identify Service to log into the Originating Platform and starts controlling the Avatar.
2. User uses a Portal UI to initiate the teleportation of an Avatar to a Destination Location. This use case assumes that the destination already is determined by the Portal. The Portal can be a visual "door" in the Platform that the Avatar "walks through". It can be a UI toolbar that allows the User to enter a destination "address". It may or may not offer the User a dialog to confirm the intended destination.
3. Originating Platform triggers IWPS to initiate the teleportation.
4. Portal system launches the Destination Platform and notifies the Destination Platform of the incoming Avatar and the Destination Location.
5. Destination Platform gives portal system instructions.
6. User uses Identity Service to log into the Destination Platform, possibly going through steps to register the User or possibly doing it automatically.
7. Avatar appears in the Destination Platform (with the native representation in the Destination Platform) at the Destination Location specified by the Originating Platform Portal, or other location the Destination Platform specifies that overrides the Destination Location based on the Destination Platform rules.
8. Optional- Portal system may use an Asset Transfer System to transfer Items to the Destination Platform. Such Items must be messaged to the System by the Originating Platform.
9. Optional- Avatar can return to the Originating Platform using a Portal UI at the Destination Location (or overridden location) that has stored the location of the Originating Platform Portal UI.
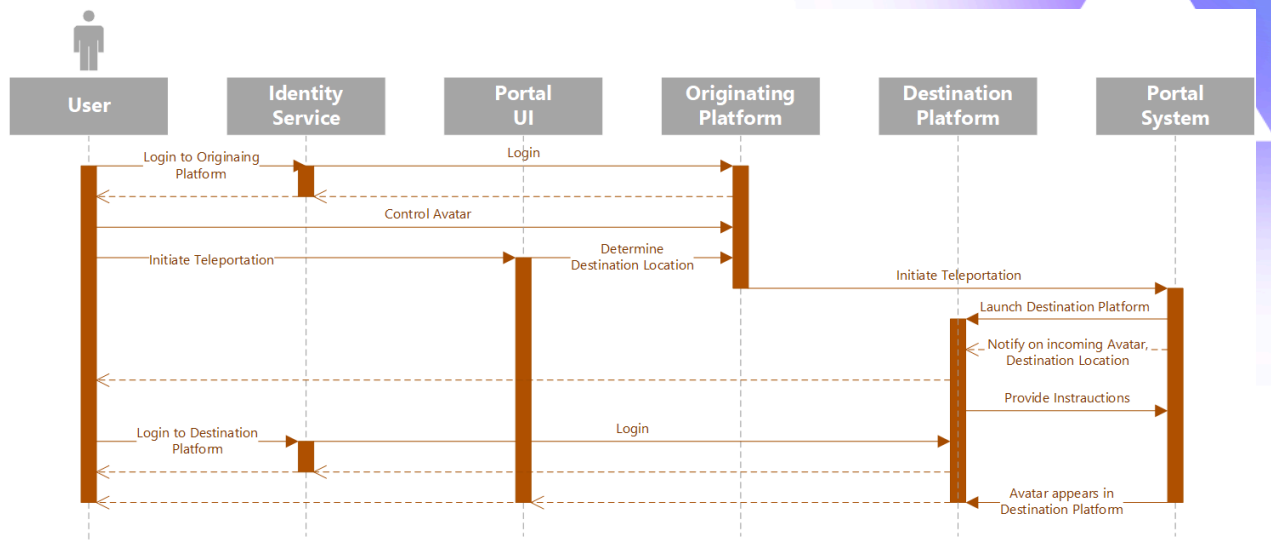
Fig 2. Portal Utilization

Portal Creation Use Case

1. Destination Platform registers itself with the System, including a default Destination Location and persistence support (persistent and/or ephemeral Portals)
2. Optional- System may require Destination Platform to pass an interoperability test in order to be registered.
3. Originating Platform queries System for a list of Destination Platforms.
4. User or Originating Platform (depending on Originating Platform implementation choices) chooses Destination Platform (and optionally Destination Location) and creates the Portal.
5. Optional- User or Originating Platform (depending on Originating Platform implementation) records the desired Destination Location in the Portal.
6. User or Originating Platform decides if the Portal is persistent, depending on Originating Platform implementation.
7. Optional- Originating Platform sends message to Destination Platform using the Inter-Platform Messaging System to set up a return destination Portal with the proper parameters for returning an Avatar from the Destination Platform back to the Originating Platform.
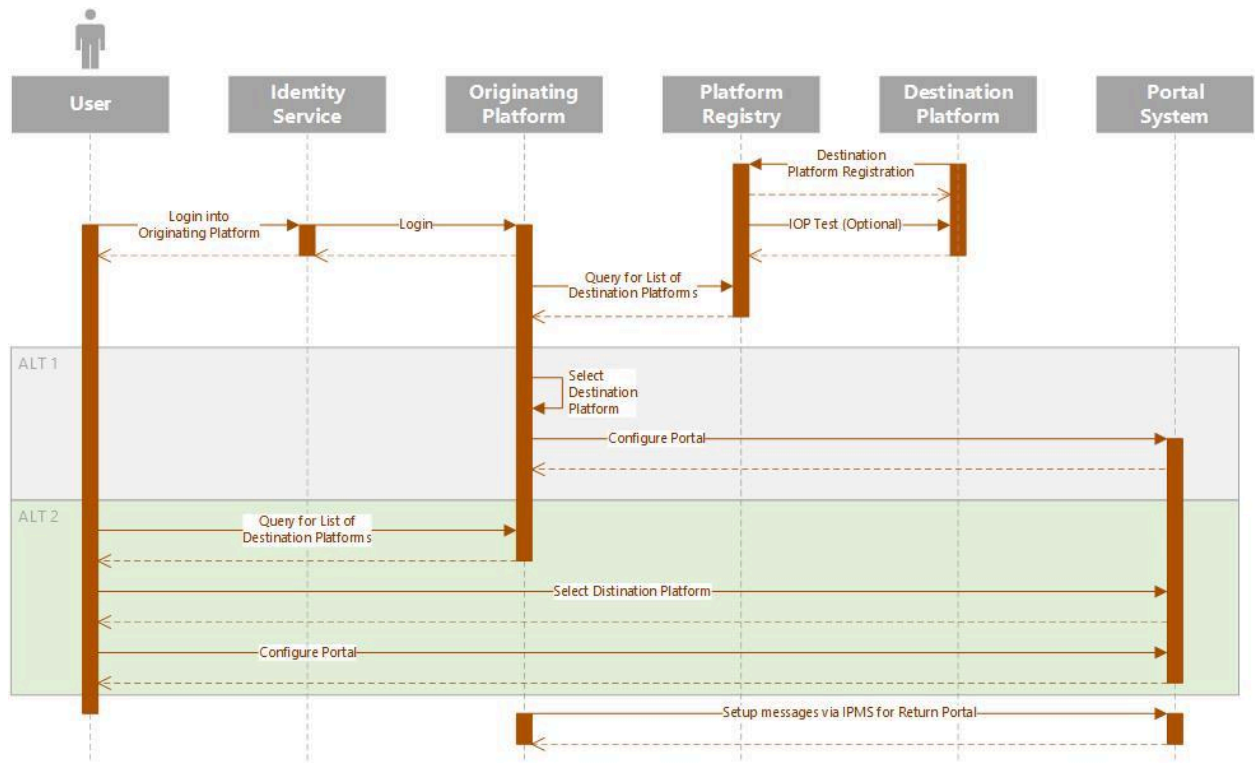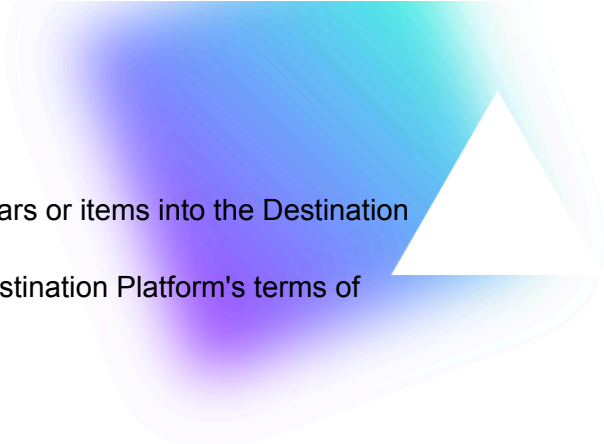
Fig 3. Portal Creation

5. Threat Model

Recognizing and documenting potential threats is crucial for building a robust portaling system, particularly in the complex landscape of the metaverse. The aim of this section is to provide a high level analysis of the diverse ways the system could be compromised or malfunction. This covers deliberate attempts to undermine the system, unintended user errors, or unforeseen edge cases. By preemptively identifying these issues, we can implement targeted strategies to mitigate risks, thus reinforcing both the reliability and credibility of the system. Below is a initial list of these potential weak points:

Portal Utilization

Portal utilization stands as one of the most critical components in the overarching functionality of our system. As the gateway that facilitates the movement of users and digital assets between various platforms—be they games, decentralized finance systems, or social spaces—it's vital that this feature operates smoothly, securely, and in alignment with user expectations. Not only does the integrity of portal utilization directly impact user experience, but it also has far-reaching implications for data security, asset management, and overall system robustness. Therefore, this section aims to outline potential risks, shortcomings, and anomalies that could undermine the effectiveness and safety of portal utilization.

1. Misdirected Travel: The system may inadvertently send Users to unintended Destination Locations.
2. Forced Transfers: Users can be transported without their explicit consent or against their will.
3. Third-Party Platform Bypass: A third-party platform might circumvent the Registry system and launch in place of the intended Destination Platform.
4. Unauthorized Platform Registration: Unapproved platforms might be included in the official Registry.
5. System Failures: Avatars may get lost between Platforms if the system crashes or experiences other malfunctions.
6. Avatar Acceptance Loopholes: Even if a Destination Platform rejects an incoming Avatar, Users may find ways to bypass these restrictions.
7. Account Limitations: User doesn't have an account on the Destination Platform.
8. Identity Service Gaps: Destination Platform doesn't support the Identity Service used by the System.
9. Crypto Asset Barriers: Destination Platform requires crypto assets that the User doesn't possess.
10. Travel Mechanics Abuse: Users could use IWPS to undermine the intended mechanics of moving within a Platform, such as visiting Destination Locations that are normally off-limits to the User.

11. Foreign Assets and Avatars: User brings non-native Avatars or items into the Destination Platform (e.g.- laser guns in a Medieval world).
12. Content Policy Violations: Users or Assets breach the Destination Platform's terms of service, including but not limited to:
    ○ Age Restrictions
    ○ Adult Content
    ○ Specific Game Mechanics
13. Asset Security: Adversaries compromise assets stored in User's wallet.
14. Malware Risk: Users may download executable files that install malware on their systems.

Portal Creation

The process of creating portals serves as the initiating step for user interactions within the system. Any errors or shortcomings that occur during this phase can have a cascading effect on subsequent operations, compromising the entire user experience.

15. Misinterpretation of Destination by Platform: Destination Platform misunderstands or incorrectly parses the intended Destination Location.
16. User-Provided Incorrect Destination: User mistakenly inputs the wrong Destination Location.
17. Originating Platform Interpretation Errors: Originating Platform misreads or incorrectly identifies both the Destination Platform and the intended Destination Location.
18. Unregistered Destination Platforms: User adds a Destination Platform that has not been validated or is not listed in the Registry.

Reliability

19. Downtime of Destination Platform: Destination Platform is temporarily unavailable.
20. Registry Unavailability: Registry is down or inaccessible.
21. API Server Outages: API servers that facilitate portaling are down.

Privacy

22. Actors tracking where Users/Avatars go in the metaverse.

6. Requirements

The Requirements section covers a comprehensive range of functional and non-functional requirements that IWPS needs to fulfill. These span across performance, reliability, governance, usability, security, and interoperability. Also included are specific requirements related to identity services, payment systems, and asset transfers. Note that this document does not delve into the architectural or design aspects of the system, as they are covered in separate documents.

Requirements definitions are used to judge specification proposals that describe system implementations. When writing requirement definitions, we use the key words "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", and "MAY" as described in RFC 2119 [1].

SHALL: This word means that the definition is an absolute requirement of the specification.

SHALL NOT means that the definition is an absolute prohibition of the specification.

SHOULD: This word means that there may exist valid reasons in particular circumstances to ignore a particular definition, but the full implications must be understood and carefully weighed before choosing a different course.

SHOULD NOT: This phrase means that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.

MAY: This word means that a definition is truly optional. A specification may choose to fulfill the requirement because a particular marketplace requires it or because the specification feels that it enhances the product while another specification may omit the same item.

1.   Platform Compatibility Requirements

Platform compatibility requirement definitions describe the breadth of platforms and technologies an IWPS specification is expected to support. The metaverse is highly diverse with platforms ranging from 2D to 3D to virtual reality representations. Metaverse platforms run

on mobile devices, desktop operating systems, and browsers. IWPS's goal is to encompass as diverse a technology stack as possible in its goal to connect the metaverse.

1.1.   General Compatibility - IWPS SHALL provide developer hooks that allow integration into the following types of Platforms:

- Platforms that do not use blockchain.
- Platforms developed with a Turing complete programming language.
- Platforms that use 2D graphics depiction.
- Platforms that use 3D graphics depiction.

Rationale: To ensure that IWPS gets wide adoption and the metaverse includes as many worlds as possible, it is imperative that it supports the most commonly used programming paradigms and graphical interfaces used by Platforms.

1.2.   Game Engine Compatibility - IWPS SHALL provide developer hooks to integrate with Platforms that use:

- Unreal Engine
- Unity engine
- Playcanvas engine

Rationale: These are widely used game engines and having compatibility with them ensures a wider user base and ease of integration.

1.3.   Browser Compatibility - IWPS SHALL provide developer hooks that allow integration into Platforms which:

- Run in a Chromium-based browser
- Run in a Firefox browser

Rationale: Popular browsers should be supported to ensure maximum reach and usability.

1.4.   Mobile and Desktop OS Compatibility - IWPS SHALL provide developer hooks that allow integration into Platforms which run natively on:

- iOS
- MacOS
- Android
- Windows

Rationale: These are the major operating systems for both mobile and desktop environments. Supporting them ensures nearly universal compatibility.

1.5.   Other compatibility - IWPS SHOULD provide developer hooks that allow integration into Platforms with the following characteristics:

- ■ Utilize blockchains with median finality of under one minute.
- ■ Are installed via app store or direct download.

Rationale: Supporting fast-finality blockchains enhances the IWPS's ability to facilitate quick and seamless portaling experiences. Additionally, allowing for diverse installation methods expands the system's compatibility across various Platforms.

1.6. Additional Compatibility - IWPS MAY provide developer hooks that allow integration into Platforms with the following characteristics:
- ■ Developed using any game engine.
- ■ Runs natively on any operating system.

Rationale: It is valuable to capture any additional markets and future-proof the platform.

2. <u>Performance Requirements</u>

A system that does not perform well results in a poor user experience. These requirements ensure the system meets user expectations in responsiveness.

2.1. IWPS SHOULD have a maximum latency of 1 second for displaying Destination Platform information.

Rationale: Quick display of Destination Platform information is essential for user engagement and efficient operations.

2.2. IWPS SHOULD have a maximum latency of 20 seconds for Users that already have an account in the Destination Platform.

Rationale: Returning users expect faster interactions and this will ensure a smooth user experience for them.

2.3. IWPS SHOULD have a maximum latency of 1 minute for Users that do not have an account in the Destination Platform but have downloaded the Destination Platform client.

Rationale: New users with the client should experience good performance to encourage continued use.

2.4. IWPS MAY communicate the latency for different stages of portaling to Users.

Rationale: This allows users to be informed and manage their expectations during the teleportation process.

2.5.   IWPS SHALL allow the Originating Platform to communicate the availability status of the Destination Platform to the User.

   Rationale: Availability status is critical for user decision-making and helps to manage user expectations.

2.6.   IWPS MAY provide a rate limiter with the following functionalities:
   ■   Prevents IWPS from executing more than a certain number of teleportations per second.
   ■   Takes into account outstanding outbound teleportation requests on the Originating Platform.
   ■   Takes into account inbound teleportation requests on the Destination Platform.

   Rationale: Rate limiting ensures that the system remains stable and responsive under various conditions.

2.7.   IWPS MAY enable Platforms to display a visual queue reflecting waiting times at both Originating and Destination Platforms

   Rationale: A visual queue helps manage user expectations and could improve user experience.

2.8.   IWPS MAY enable Platforms to display a visual queue reflecting waiting times at both Originating and Destination Platforms.

   Rationale: A visual queue helps manage user expectations and could improve user experience.

2.9.   IWPS SHOULD have an uptime of 95%

   Rationale: High uptime ensures availability and trust in the service.

2.10.   IWPS SHOULD implement DDoS protection mechanisms.

   Rationale: Protection against DDoS attacks ensures service availability and user trust.

2.11.   IWPS SHOULD be resistant to censorship.

   Rationale: Censorship resistance ensures that the platform is accessible to a broad user base. Censorship resistance is a core Web3 value.

2.12.   IWPS SHOULD be decentralized with no single point of failure controlled by a single entity.

> Rationale: Decentralization adds an additional layer of reliability and security to the platform. Decentralization is also a core Web3 value.

3.   Governance Requirements

The Governance Section outlines the organizational and administrative principles essential for the effective functioning and trustworthiness of the Inter World Portaling System (IWPS).

3.1.   IWPS SHALL distinguish between certified and uncertified Platforms.

> Rationale: Differentiating between certified and uncertified platforms establishes a trust mechanism and ensures compliance with set standards to guarantee interoperability.

3.2.   IWPS SHOULD support the following permissions policies: Decentralized infrastructure can be run by any OMA3 Creator or Sponsor member.

> Rationale: Allowing only OMA3 Creator or Sponsor members to run the decentralized infrastructure ensures a certain level of quality and trust, while still being inclusive. IWPS can always become more permissionless over time whereas going from permissionless to permissioned is a harder transition.

4.   Usability Requirements

The Usability Section of this document is dedicated to defining standards that prioritize user-centric design, ease of use, and the overall user experience in the Inter World Portaling System (IWPS). The guidelines set forth in this section aim to reduce barriers to entry, simplify processes, and enhance the fluidity of interactions for users. Areas covered include the need for extra downloads, text entry, click interactions, visual media requirements, monetization options, avatar visibility, and notifications. These factors contribute to the creation of a seamless, intuitive, and flexible environment for users navigating across multiple platforms in the Metaverse.

4.1.   IWPS SHOULD NOT require any extra downloads other than the downloading of Platform applications.

> Rationale: This requirement streamlines the user experience and reduces entry barriers for users.

4.2.   IWPS SHOULD enable teleportations that do not require any text entry other than entering the Destination Platform and Destination Location.

Rationale:Reducing the time spent entering text will simplify the teleportation process and improve usability and accessibility.

4.3. IWPS MAY enable teleportations that do not require additional clicks during the teleportation process.

Rationale: Minimizing user actions during teleportation enhances user experience and gives the truest experience of a seamless and interconnected metaverse.

4.4. IWPS SHOULD NOT require an image or video of the Destination Platform inside an Originating Platform's Portal UI.

Rationale: Reducing the need for visual media from the Destination Platform reduces complexity and dependency between platforms. It also reduces the minimum requirements for Platforms to be compatible with the system.

4.5. IWPS SHALL allow Platforms to enforce paywalls when users arrive at the Destination Location.

Rationale: This allows Destination Platforms to have flexibility with their monetization strategy.

4.6. IWPS SHALL NOT enforce the condition that an Avatar can only be visible in one location across all IWPS Platforms.

Rationale: This ensures that users have the freedom to multitask across multiple platforms and gives Platforms flexibility on their experience.

4.7. IWPS SHOULD NOT enforce any UX elements on the Originating Platform.

Rationale: This gives Originating Platforms the flexibility to maintain their unique user experiences.

4.8. IWPS SHOULD implement a notification system for informing Users.

Rationale: A notification system keeps users informed about critical updates, enhancing the user experience.

4.9. IWPS SHOULD allow Destination Platforms to automatically launch without requiring the User to manually open the Destination Platform application

Rationale: This requirement streamlines the teleportation process and gives Users the feeling of an interconnected metaverse.

4.10.   IWPS SHOULD allow a User to understand the privacy policy of the Destination Platform before initiating the teleportation.

4.11.   IWPS SHOULD allow a User to understand the intellectual property policy of the Destination Platform before initiating the teleportation.

4.12.   IWPS SHOULD allow a User to understand the requirements of launching the Destination Platform, including client software downloads, registration, and native token ownership, before initiating the teleportation.

4.13.   IWPS SHOULD standardize legal agreements that gives Platforms flexibility but at the same time allows Users to understand the implications without having to read the full agreement (e.g.- Apache 2 is well understood).

5.   Messaging System Requirements

This section outlines the essential requirements for messaging systems that IWPS will rely on for communication across various Platforms, applications, and devices. The requirements definitions aim to ensure IWPS runs on secure messaging systems with enough flexibility to meet compatibility requirements. Each subsection is designed to address a different aspect of communication, ranging from intradevice messaging to secure Internet-based communication, thereby ensuring a robust and adaptable messaging systems that IWPS can rely on.

5.1.   IWPS SHALL support messaging systems that allow communication between applications running on the same operating system.

Rationale:To enable teleportation and communications between different applications within the same operating system.

5.2.   IWPS MAY support messaging systems that allow communication across operating systems running on the same device.

Rationale: Multi-OS devices are uncommon, but there is benefit to supporting this scenario.

5.3.   IWPS SHOULD support messaging systems that allow communication between different devices.

Rationale: A cross-metaverse IWPS is severely limited without inter-device teleportation.

5.4. IWPS SHALL support messaging systems that allow communication over the Internet.

Rationale: To facilitate global reach and interconnectivity.

5.5. IWPS SHALL only support messaging systems that allow a Platform to use TLS communications.

Rationale: Secure communications prevent adversaries from compromising IWPS.

5.6. IWPS MAY support messaging systems that allow communications between smart contracts on different blockchains.

Rationale: IWPS asset transfers may need cross-chain communications.

6. Registry Requirements

The Registry is an IWPS component that keeps track of "certified" Platforms. This section lays out the requirements for registry functionality, ensuring a streamlined and secure experience for all Platforms.

6.1. The Registry SHALL allow Platforms to communicate with it via web-based API accessible from any Platform with access to the Internet.

Rationale: An API is the most common method to programmatically access a resource.

6.2. The Registry MAY allow communication via smart contract API accessible from any smart contract on the same blockchain.

Rationale: It is unlikely the Registry's smart contracts will reside on the same blockchain as a Platform, but if they do smart contract APIs are more efficient than off-chain APIs.

6.3. The Registry SHALL only list certified Platforms.
6.4. The Registry SHALL only allow certified Platforms to access the Registry.

Rationale: For the best user experience only Platforms that are certified as interoperable with other Platforms and compliant to IWPS specifications should be part of IWPS. Note that this could be a temporary "beta" requirement only and over time IWPS gives the option for Users or Platforms to incorporate

non-certified Platforms as well. OMA3 would like to receive comments on these requirements from the community.

7.   Security Requirements

This section delineates the security requirements for deterring unapproved use of IWPS. These requirements are fundamental to preserving the integrity, confidentiality, and availability of data and services within the system.

7.1.   All data transferred between Actors SHALL be encrypted (TLS ciphersuites, AES).

7.2.   All data transferred SHALL use authentication (X.509 certificates, PKI, plain public/private (JWT), symmetric shared keys).

> Rationale: To ensure the secure transmission of data between platforms.

7.3.   Destination Platforms SHALL authenticate themselves to Users wishing to teleport to them from an Originating Platform.

> Rationale:To confirm the authenticity of the destination, enhancing user security and preventing hijacking of teleportation.

8.   Identity Service Requirements

This section outlines the comprehensive set of requirements governing identity management within the system. IWPS needs to be flexible enough to support commonly used Web3 identity systems as well as Web3 identity systems as metaverse Platforms are diverse in terms of how they handle identity.

8.1.   IWPS SHOULD integrate an Identity Service abstraction layer that gives a unified interface to one or more approved Identity Services.

> Rationale: IWPS should be flexible enough to support different Identity Services, but Identity Services should be approved to protect Users from exploit.

8.2.   IWPS SHOULD NOT require Platforms to support a separate Identity Service, and instead allow the Destination Platforms to use its own authentication.

> Rationale: To provide flexibility in identity management for various Platforms.

8.3.   IWPS SHALL allow Platforms to use any identity method they want.

> Rationale: To provide flexibility in identity management for various Platforms.

8.4.   IWPS SHOULD require Destination Platforms to communicate which authentication methods they support.

Rationale: Users should know the options they have for identity in a Platform.

8.5.   IWPS SHOULD track which Originating Platform IDs are tied to which Destination Platform IDs while conforming to 8.17, 8.18, and section 10 below.

Rationale: This ensures IWPS works in a consistent manner from a User perspective.

8.6.   IWPS SHOULD give Platforms the cybersecurity information (e.g., certifications) Platforms need to make their own determination on which Identity Service they use.

Rationale: This gives Platforms more sovereignty over their Identity Service integration.

8.7.   IWPS SHALL give Platforms the privacy information (e.g., certifications) Platforms need to make their own determination on which Identity Service they use.

Rationale: This gives Platforms more sovereignty over their Identity Service integration.

8.8.   IWPS SHALL only integrate with Identity Services that support all of the following sign-on and account sharing methods:
- Username/password with 2FA (text, mobile app, FIDO hardware, etc.)
- Mobile crypto wallets
- Browser crypto wallets

Rationale: These methods are secure and commonly used today, so Identity Services that integrate with IWPS need to support them to ensure a smooth and secure user experience.

8.9.   IWPS SHOULD support Identity Services that support any of the following sign-on and account sharing methods:
- OAUTH2/SAML/OpenID Connect
- Hardware crypto wallets
- Web3 identity
- DIDs

Rationale: To enable broader compatibility with existing and future Identity Services.

8.10.   IWPS MAY have hooks to support Identity Services that support the following sign-on and account sharing methods:

- MPC crypto wallets
- Smart contract crypto wallets, including account abstraction
- One time code
- Sign In with asymmetric signature (e.g., public key pair JWT)

Rationale: To offer extensive flexibility in identity solutions.

8.11.   IWPS SHALL support Identity Services that use DID verified credentials to log in users to Platforms.

Rationale: DIDs give Users control over privacy and security.

8.12.   IWPS SHOULD have the ability to track multiple Avatars controlled by one User and notify the Destination Platform which Avatar the User is currently using to teleport.

Rationale: Users often maintain more than one Avatar on a Platform. This requirement ensures IWPS supports this kind of identity model.

8.13.   IWPS SHOULD support Identity Services that use Web3 domains to identify Users.

Rationale: To support newer, blockchain-based methods of identification.

8.14.   IWPS SHALL integrate with Identity Services that support different types of private keys (Neo vs EVM vs BTC).

Rationale: To ensure compatibility across various blockchain technologies.

8.15.   IWPS SHALL support identities across different blockchains.

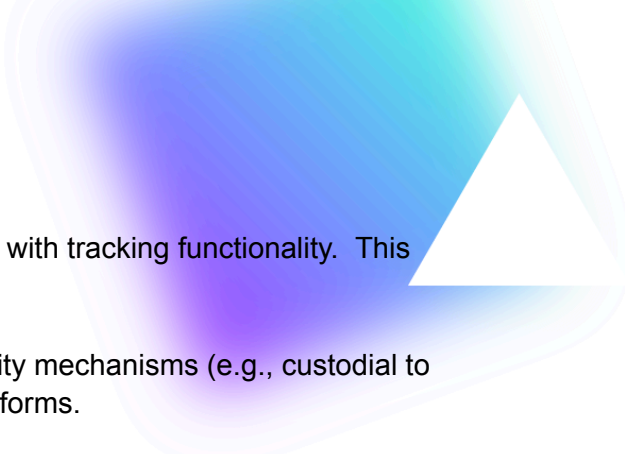Rationale: To enable a seamless cross-blockchain identity experience for users.

8.16.   IWPS SHALL allow a User to create an account on the Destination Platform if the User does not have one.

Rationale: This will allow Users to explore the metaverse without limitations.

8.17.   IWPS SHALL require Identity Services to adhere to all privacy regulations pertaining to the User's jurisdiction.

Rationale: To comply with legal requirements and ensure user privacy.

8.18.   IWPS SHALL require Identity Services to adhere to privacy requirements in this document.

Rationale: Web3 often tramples on User privacy with tracking functionality. This is against OMA3 principles.

8.19. IWPS MAY support the User migrating to different identity mechanisms (e.g., custodial to self-custody) while maintaining the same identity in Platforms.

Rationale: In Web3 users often go through a migration journey from custodial Web2-style identity mechanisms to self-custody identity mechanisms. It is advantageous for IWPS to support this migration.

8.20. IWPS SHOULD enable future support for tying digital assets (besides avatars) to Users and Avatars.

Rationale: It is beneficial to allow Avatars to take their Items from Platform to Platform.

8.21. IWPS SHOULD allow users to log in to the Destination Platform with their credentials if the Identity Service is not able to automatically log a User into the Destination Platform.

Rationale: To ensure that Users are not locked out in case the Identity System fails to automatically log in a User.

8.22. IWPS SHOULD require Identity Services to make available to Users information about their policies.

Rationale: To ensure transparency and informed user choice.

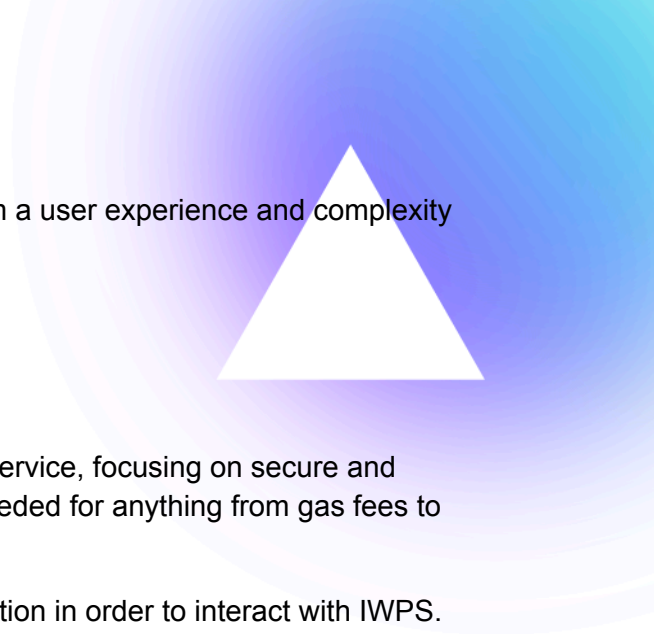8.23. IWPS MAY adopt or create a blockchain ID standard that includes a network ID

Rationale: To avoid confusion on which chain an ID is located on.

8.24. IWPS SHOULD adopt specifications that deter or eliminate unresolvable contention on namespaces.

Rationale: Contention in namespaces causes confusion with developers and users.

8.25. IWPS MAY incorporate ID systems that do not rely on the DNS system.

Rationale: The DNS system has some points of centralization.

8.26.   IWPS SHOULD place minimum burden on users from a user experience and complexity perspective

Rationale: To ensure ease of use.

9.   Payment Service Requirements

This section outlines the requirements for the Payment Service, focusing on secure and efficient transactional capabilities. Payments could be needed for anything from gas fees to Platform subscriptions.

9.1.   A Payment System SHALL get cybersecurity certification in order to interact with IWPS.

Rationale: To ensure only secure payment systems are used with IWPS.

9.2.   A Payment System SHALL allow Users to pay for the following fees:
- ■   Platform subscription
- ■   Gas.

Rationale: To provide essential payment functionalities within the platform.

9.3.   A Payment System SHOULD allow Users to pay for the following fees if required:
- ■   Asset transfer
- ■   IWPS teleportation.

Rationale: These fees may also be required in IWPS.

9.4.   A Payment System SHOULD allow for automatic payments.

Rationale: To streamline the payment process and improve user experience.

9.5.   A Payment System MAY allow for recurring payments.

Rationale: To offer users the convenience of subscription-based services.

10.   Privacy Requirements

This section ensures that both System and Actors operate within the confines of data privacy laws and best practices. These requirements focus on the necessity of user consent before data collection and stringent access control measures for stored data. Special attention is given to the protection of sensitive real-world identities and compliance with jurisdiction-specific data protection regulations.

10.1.   IWPS SHALL require Data Collectors to seek permission before collecting data.

Rationale: To comply with data privacy laws and user consent.

10.2. Data stored by Data Collectors SHALL have access control and be protected from unauthorized access.

Rationale: To ensure data security and prevent breaches.

10.3. Real-world identities and associated data (e.g., KYC data) SHALL be protected from unauthorized access.

Rationale: To safeguard sensitive identity information from unauthorized users.

10.4. Data collection SHALL comply with regulations of the User's or Platform's jurisdiction.

Rationale: To ensure compliance with local and international data protection laws.

11. Asset Transfer Requirements

This section outlines the guidelines and requirements concerning asset transfers within the IWPS ecosystem. It emphasizes the need to standardize asset transfer rules and ensure compatibility across platforms. The aim is to establish a seamless, user-friendly, and secure environment for asset management and transfers between different platforms.

11.1. IWPS SHOULD support a mechanism that regulates how assets created by the creating Platform are treated in other Platforms, and Destination Platforms either have to accept those terms or deny the asset transfer.

Rationale: IWPS respects the rights of creators, which means all Platforms must accept the terms creators establish for assets they create or not use them at all.
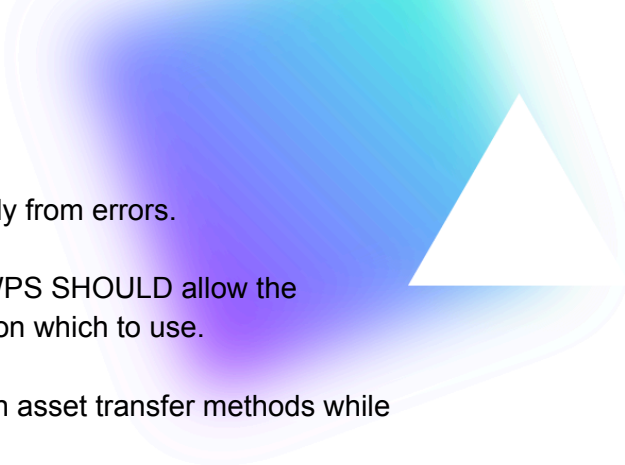
11.2. Incompatibilities in asset transfers SHOULD be communicated to Users before they initiate the teleportation.

Rationale: To keep users informed and prevent unexpected issues during asset transfers.

11.3. IWPS SHOULD support bilateral agreements between Platforms on Asset transfers.

Rationale: Bilateral agreements streamline the asset transfer process between Platforms.

11.4. IWPS SHALL ensure (to the extent possible) that Avatars and all Items show up in the Originating Platform if there is an error (as determined by the Destination Platform) before the Avatar appears in the Destination Platform.

Rationale: This allows IWPS to recover gracefully from errors.

11.5. If there are different methods for transferring Assets, IWPS SHOULD allow the Originating Platform and Destination Platform to agree on which to use.

Rationale: This gives Platforms the flexibility with asset transfer methods while also maintaining a seamless user experience.

11.6. System SHALL NOT force Platforms to transfer Items through the System unless it is a return transfer (see below).

Rationale: Give Platforms complete sovereignty over their asset management.

11.7. Destination Platform SHALL give Users the ability to transfer Assets from Destination Platform back to the Originating Platform.

Rationale: To ensure users maintain control over their assets while using IWPS.

11.8. IWPS SHALL require Originating Platforms to allow Users to control when an asset transfer happens. An example of such control would be a setting, or a one-time permission such as giving a mobile application permission to use your location data.

Rationale: To ensure users maintain control over their assets while using IWPS.

12. IWPS Economy Requirements

IWPS will support an economy that will support functionality such as fees for teleportation, staking for portal creation, and a compensation and trust system in platform registration and certification. The IWPS economy may leverage a fungible token for some of this functionality.
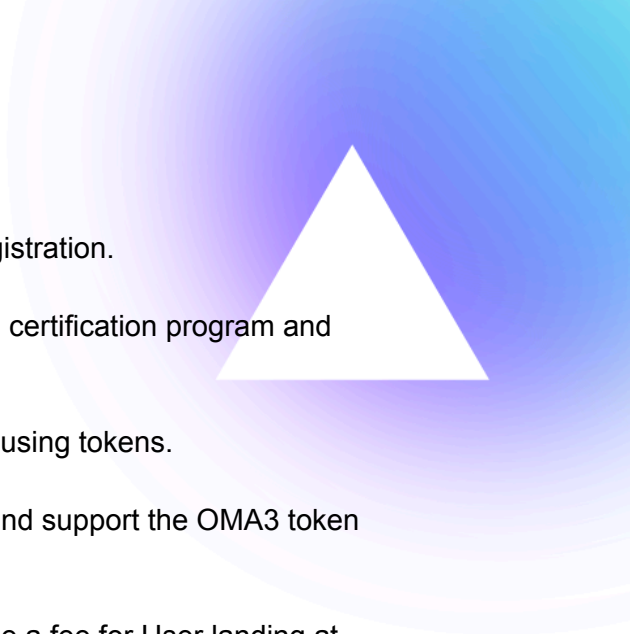
12.1. IWPS MAY charge a fee for teleportation.

Rationale: To cover the operational costs associated with the teleportation feature.

12.2. If there is a fee, IWPS SHOULD allow Platforms to pay teleportation fees for the User.

Rationale: To improve the user experience by reducing or eliminating costs they may incur.

12.3. IWPS MAY establish a staking fee for portal creation.

Rationale: To allow Platforms to monetize IWPS and prevent spamming portals.

12.4.   IWPS MAY allow Platforms to pay a fee or stake for registration.

Rationale: To pay for the cost of maintaining the certification program and requiring Platforms to have "skin in the game".

12.5.   IWPS MAY allow Platform certification costs to be paid using tokens.

Rationale: To streamline transactions in IWPS and support the OMA3 token architecture.

12.6.   IWPS SHOULD allow the Destination Platform to charge a fee for User landing at specific locations (premium vs free).

Rationale: To enable platforms to monetize specific features or locations.

12.7.   IWPS MAY allow Users to purchase priority teleportation passes.

Rationale: To offer users expedited services at a premium.

12.8.   IWPS MAY allow teleportation fees to be split between Originating Platform, Destination Platform, other Actors, and IWPS.

Rationale: To give Platforms ultimate flexibility with fees and fairly distribute the financial burden and benefits among all stakeholders.

Change Log

Version	Date	Comments
1.0	23-11-05	First draft to public

Authors

- ▲ Batis Samadian (Space, OMA3)
- ▲ Alfred Tom (Wivity, OMA3)
- ▲ Judy Chen (Wivity, OMA3)
- ▲ Alaina Bryan (Lamina1)
- ▲ Mic Mann (Africarare)
- ▲ Federico Costa (Freename)
- ▲ Priscilla Angelique (APT)
- ▲ Marc McGinley (Animoca)
- ▲ Zara Zamani (Neoki)
- ▲ Martin Riedel (Identity.com)
- ▲ Anton Knode (Crosstech)
- ▲ Riccardo Sibani (Alice)
- ▲ Sebastien Bru (Ilogos)
- ▲ Jörg Brakensiek (Wivity)
- ▲ Charles Adelman (XRDNA)
- ▲ Angelo Laub (Dacoco)
- ▲ Joel Dietz (MultiversalME)
- ▲ Idan Zuckerman (Upland)
- ▲ Jerry Lai (Animoca)
- ▲ Jung Wookyung (Etri)
- ▲ Phillip Shoemaker (Identity.com)
- ▲ Vincent Ghossoub (Nifty Craft)
- ▲ Saiid El Hajj Chehade (Nifty Craft)
- ▲ Gino Cingolani (Decentraland)
- ▲ William Mougayar (MetaMapp)
- ▲ Somnath Banerjee (MetaJuice)
- ▲ Anis Chohan (Virtua)
- ▲ Paul-David Oosthuizen (MFG)
- ▲ Davide Vicini (Freename)

Appendix: Existing Art

This appendix is a partial list of technology, practices, and solutions that are relevant to IWPS. The goal is to add to this list so it is more comprehensive as this is a work in progress.

1. Developer Integration Hooks
    1.1. Plugins: Pre-built code that can be easily integrated into existing platforms to extend functionalities.
    1.2. Extensions: Add-ons that users can install to extend the capabilities of a software application.
    1.3. Source Code: Direct access to the source code for customization.

2. Messaging Service
    2.1. Communication Layer - Protocols
        2.1.1. IPSME/Messaging: A specialized messaging protocol.
        2.1.2. Phone OS messaging: Deep linking.
        2.1.3. Headsets (e.g.- MS Hololens, Apple VisionOS).
        2.1.4. HTTPS/TCP/IP: Includes Web Sockets, REST, SOAP, FTP, and SOAP.
        2.1.5. WebRTC: Real-time communication.
        2.1.6. gRPC: High-performance, universal remote procedure call (RPC) framework.
    2.2. Structure Layer - Data Structures
        2.2.1. JSON: Widely used data interchange format.
        2.2.2. GraphQL over JSON: Provides a more efficient, powerful and flexible alternative to the traditional REST API.
        2.2.3. XML: Another widely-used data interchange format.
        2.2.4. Binary: For compressed, faster data interchange.
    2.3. Content Layer - Types of Content
        2.3.1. Mapping of content to worlds: Determining location.
        2.3.2. Transferring identity and assets attached to the avatar passing through the portal.
        2.3.3. Communicating portaling intent: Information about the incoming avatar.
        2.3.4. Error Communication: Notifications and logs.
        2.3.5. Capacity information: Real-time data on system load.
    2.4. Transaction Models
        2.4.1. Open sessions with constant communication: Suited for real-time interactions.
        2.4.2. API-style individual sessions: Suited for sporadic, on-demand interactions.

3. Identity Service
    3.1. Master Seed Phrase: Same master seed phrase for all chains but can derive different addresses from it.
    3.2. Custodial Wallet Services: Cross-platform support.
    3.3. Token Bound Contract: ERC 6551 vs SBTs.