

Mobile Authentication Assurance Statement (MAAS)

**Version:** 1.0.0

**Date:** 2020-10-06

**Editor:** Tom Jones

**Contributors****: Jeff Brennan, Salvatore D’Agostino. Jim Kragh, Catherine Schulten, Noreen Whysel, Tom Jones**

**Status:** Implementor’s Report DRAFT 2

Kantara Working Group: Federated Identifiers for Resilient Ecosystems

Comments directed to: <https://github.com/KantaraInitiative/DistributedAssurance/issues>

**Abstract:**

Authentication assurance is a statement of the level of protection that the authenticator enforces to assure that a user retains control of the credentials used in authentication. Protected secret credentials enable user authentication at high levels of assurance. This specification describes the format of a message for mobile assurance along with a healthcare provider use case, involving the user’s acquisition and operation of an app on a smartphone that uses the statement in an authentication protocol.

**Notice:**

This document is not a Kantara recommendation or specification but is a report distributed for trial implementations. Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights with supporting documentation.

Copyright (c) 2020 Kantara and the persons identified as the document contributing authors. All rights reserved.

This document is subject to the [Kantara IPR Policy](https://kantarainitiative.org/confluence/display/GI/Option+Non-Assertion+Covenant)

[Abstract: 1](#_Toc49258208)

[1 Introduction 3](#_Toc49258209)

[1.1 Assumptions 3](#_Toc49258210)

[1.2 Goals 3](#_Toc49258211)

[2 Notations and Abbreviations 5](#_Toc49258212)

[3 Terms and definitions 6](#_Toc49258213)

[3.1 Trustworthy Digital Ecosystem 6](#_Toc49258214)

[3.2 Guardianship 6](#_Toc49258215)

[4 Health care use case 7](#_Toc49258216)

[4.1 Actors in the test use case 7](#_Toc49258217)

[4.2 Scenarios 8](#_Toc49258218)

[4.3 Results 9](#_Toc49258219)

[4.4 User Preparation of the Device for Use 9](#_Toc49258220)

[4.5 Registration Certificate returned to the user Device 10](#_Toc49258221)

[5 Contents of messages 12](#_Toc49258222)

[5.1 Contents of Mobile Assurance Statement (MAAS) 12](#_Toc49258223)

[6 Guidance 13](#_Toc49258224)

[6.1 Presentation and Delivery 13](#_Toc49258225)

[6.2 User Experience with High Assurance UX 13](#_Toc49258226)

[6.3 Open Issues: 13](#_Toc49258227)

[7 Mobile Assurance - JSON 14](#_Toc49258228)

[7.1 JSON Fields 14](#_Toc49258229)

[7.2 JSON Schema 15](#_Toc49258230)

[8 Conformance 16](#_Toc49258231)

[9 Considerations 17](#_Toc49258232)

[9.1 Mobile Authentication Assurance User Experience 17](#_Toc49258233)

[9.2 Privacy Considerations 18](#_Toc49258234)

[9.3 Security Considerations 18](#_Toc49258235)

[9.4 Notices 19](#_Toc49258236)

[10 Acknowledgements 20](#_Toc49258237)

[11 Works Cited 21](#_Toc49258238)

[Appendix A: Categories of Data 22](#_Toc49258239)

[Appendix B: Example Mobile Assurance messages 23](#_Toc49258240)

[B.1 Human-readable Mobile Assurance– Simple 23](#_Toc49258241)

[B.2 Human-readable Mobile Assurance– Fancy 23](#_Toc49258242)

[B.3 JSON Contents 23](#_Toc49258243)

[Revision history 24](#_Toc49258244)

# Introduction

This concept of **Mobile Authentication Assurance (MAA)** is described here along with a use case for attaining [NIST SP 800-63B](https://pages.nist.gov/800-63-3/sp800-63b.html) AAL2 or AAL3 levels of authentication assurance. No person creates their identity in a single place. A person’s identity is formed in the places where they work and play, learn and advocate. So it is unlikely that any one’s identity can ever be completely encompassed by an authenticated identifier in one single [Credential Service Provider](https://wiki.idesg.org/wiki/index.php/Credential_Service_Provider). What people need is a collection of [Verified Claims](https://tcwiki.azurewebsites.net/index.php?title=Verified_Claim) that they can call upon as needed in their online interchanges to protect access to and distribution of their personal information.

This specification is designed to work with devices that are carried with the user and network attached, such as a smartphone.

This concept is formalized in this "Kantara Mobile Authentication Assurance Statement".

This specification is a part of a series of evolving Kantara specifications on distributed identifiers that will all be available [at the work group draft recommendations page](https://kantarainitiative.org/confluence/display/WT/Draft+Recommendations).

Assumptions

The following assumptions on the existence of a trustworthy ecosystem are further described in section 3. The ecosystem itself is not the subject of this specification.

Trust Federations all start with a set of terms and conditions or a [Code of Conduct](https://wiki.idesg.org/wiki/index.php/Code_of_Conduct) that define their concept of a trustworthy ecosystem for the members of the federation.

* There is a Federation Code of Conduct, a trust anchor and a collection of service providers which are registered as compliant with the code.
* The test use case for this statement is the US [Trustworthy Healthcare Ecosystem](https://wiki.idesg.org/wiki/index.php/Trustworthy_Healthcare_Ecosystem), but it is intended that it apply to other digital ecosystems as well.
  + The user has acquired a mobile computing device (such as a smartphone) that can protect the user credentials needed for authentication.
  + The user can download an app that is known to comply with the Code of Conduct which, for the [health ecosystem, has been drafted by CARIN.](https://www.carinalliance.com/wp-content/uploads/2019/05/2019_CARIN_Code_of_Conduct_05082019.pdf)
  + The user should have some level of identity assurance (IAL), but that is not part of this specification
  + Service providers that contain a user’s sensitive personal information will expect them to provide a high level of authentication assurance to access that information.
  + The user will be able to access the identity and protection levels of any service provider before the user is requested to supply any personal information.

## Goals

The Mobile Authentication Assurance Statement (MAAS) is a structured document that describes the application and mobile device which protects the user’s authentication secrets.

The goal for this specification is to enable the online users of one service provider to leverage their existing Authentication Assurance with other service providers in a secure and privacy-preserving process.

The user holds the protected authentication secrets they need to give consent to intentionally choose to move protected data among service providers that have communicated a need to acquire and protect that data.

The service providers can use this Mobile Authentication Assurance Statement to show due diligence in authenticating user’s before sharing access to stored personal information.

# Notations and Abbreviations

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", “NOT RECOMMENDED”, "MAY", and "OPTIONAL" in this document are to be interpreted as described in [[RFC 2119](http://www.rfc-editor.org/info/rfc2119)].

All JSON [[RFC 7159]](http://www.rfc-editor.org/info/rfc7159) properties and values are case sensitive. JSON data structures defined by this specification MAY contain extension properties that are not defined in this specification. Any entity receiving or retrieving a JSON data structure SHOULD ignore extension properties it is unable to understand.

AAL2,3 – Authentication Assurance Levels 2 and 3 from [NIST 800-63-3B](https://pages.nist.gov/800-63-3/sp800-63b.html)

CR Consent Receipt [Specification Version: 1.1.0 Date: 2018-02-20](https://kantarainitiative.org/file-downloads/consent-receipt-specification-v1-1-0/)

GDPR General Data Protection Regulation

IPR Intellectual Property Rights

JSON JavaScript Object Notation in [IETF RFC 7159](https://tools.ietf.org/html/rfc7159)

JSON Canonicalization Scheme (JCS) [IETF RFC 8785](file:///\\WDMYCLOUDMIRROR\Public\Backup\tomj_f\Documents\Kragh\JSON%20Canonicalization%20Scheme%20(JCS))

JWE JSON Web Encryption in IETF RFC 7516

JWK JSON Web Key in IETF RFC 7517

JWKS JSON Web Key Set in IETF RFC 7517

JWS JSON Web Signature in IETF RFC 7515

JWT JSON Web Token in [IETF RFC 7797](https://tools.ietf.org/html/rfc7519)

PII Personally Identifiable Information

PHI Private Health Information (PII that applies to the specific use case)

# Terms and definitions

This specification uses terminology and definitions from OpenID Connection and other specifications for JWT, JWE, JWS and JWK. In addition, OAuth **[RFC 6749]** and other specifications listed in the normative references at the end of this specification have defined terms.

Definitions of the following terms are taken from the [Kantara Identity Assurance Framework](https://kantarainitiative.org/confluence/display/IAWG/IAF+1050+-+Glossary+and+Overview): (Kantara Identity Assurance Work Group, 2019) Assessor, Assurance, Authentication, and Credential Service Provider (CSP). Note that no distinction is made in this specification between Applicant and Subject, since the Subject Identifier (sub) is chosen by the user before the CSP is contacted. This sub is not of much use, however, until a CSP issues a certificate for the sub and it is always possible for more than one CSP to issue a certificate for the same sub with different levels of assurance.

The healthcare use case described here uses the term Patient to describe the owner of the protected health information (PHI) which is the PII described above, called the Electronic Health Records (EHR) at the provider. The user of the service is either the patient (aka subject) who owns the PHI, or a guardian which is delegated authority over the PHI in the EHR by the patient or appropriate sovereign entity. User access to their PHI will be controlled by a **Credential** in their possession. In this use case the credential is instantiated by a private key protected from disclosure in the mobile device. This is more fully described in [Phone as a Healthcare Credential](https://wiki.idesg.org/wiki/index.php/Phone_as_Health_Care_Credential). More detail can be found in the Distributed Identity Assurance doc [at this site](https://kantarainitiative.org/confluence/display/WT/Draft+Recommendations).

## Trustworthy Digital Ecosystem

It helps to understand how Mobile Assurance fits into the broader picture of a Trustworthy Digital Ecosystem by starting from the top of the trust chain and working down.

The simplest form of a digital ecosystem starts with a single **Trust Anchor,** which could be viewed as the one node to rule them all. This is the single source of trust from which all other entities in the ecosystem can trace the provenance of their trust. This is not about the centralized naming system of the internet, which will be assumed to be in operation, but rather about the trust that one node of the network can have with other nodes.

## Guardianship

The terms [guardian](https://wiki.idesg.org/wiki/index.php/Guardian), [delegate](https://wiki.idesg.org/wiki/index.php/Delegation#Full_Title_or_Meme) or similar are defined in the documents that can be accessed at that site. In general, a delegation statement is required when one user requests to view or alter information about another user.

# Health care use case

(This section is non-normative)

## Actors in the test use case

The Actors in the use case are identified by numbers in the list and figure below except for the mobile app which runs on the mobile device.

1. User (Patient or Guardian) of one healthcare provider seeking services at another (referred) provider. In technical terms the Patient is the resource owner of the personally identifiable information (PII).
2. A mobile device (e. g. smartphone) with an operating system that can connect to the internet and securely run a mobile app (8). This device will also need to protect the user's private key when AAL2 or higher is required.
3. Assessor of claim of high assurance with provided Identifier (herein called the Credential Service Provider - CSP).
4. Existing provider of healthcare services. (The source of Identifier Assurance and later of the user’s PII.)
5. Another provider of healthcare services. (The sink of Identifier Assurance and later of the user’s PII.)
6. Trust Registry which can assure users that all digital actors are trustworthy by issuing digital trust indicators that attest to their membership in the trust ecosystem.
7. Medical Records Locator Service can find the Patient’s records wherever they are located.
8. A mobile user agent app that is certified by an appropriate federation authority to correctly relay a user's intent to any web site that needs the assurances described in this specification.

The terms “Source” and “Sink” are used in the sense of network data flows. A solid line arrow in the following diagram goes from the source of the data to the sink for the data. In IETF standards, like the OAuth 2.0 Authorization Framework, the Source is called a Resource Server and the Sink is called the user’s Client.

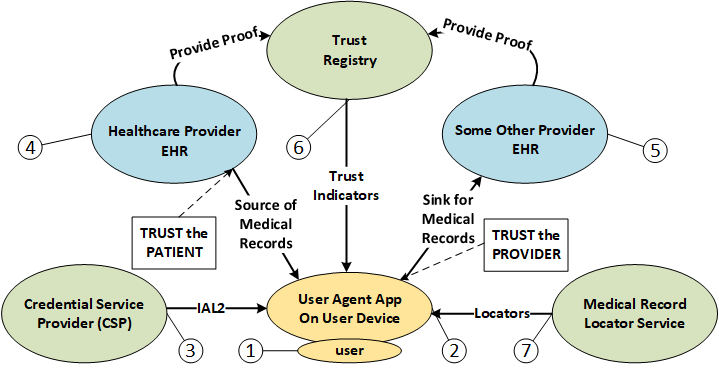


Figure 1 - The US Healthcare Assurance Framework

This specification uses the term “US Healthcare Assurance Framework” rather than the more common “[Federation](https://tcwiki.azurewebsites.net/index.php?title=Federation_Trust_Registry)” due to the large number of entities, governments and associations that are involved. Many of the concepts of federation are applicable to this test use case. The US Healthcare Assurance Framework is used as a working placeholder for the trust framework element used in the MAAS document with the assumption that the final term will be selected by relevant federation community.

Scenarios

Primary Scenario:

1. Patient visits a HIPAA covered healthcare provider where they are known as a Patient. After the medical visit they receive a paper or other notification with instructions for establishing a strong authentication credential on their smartphone.
2. Patient uses their smartphone to acquire the native app specific to their device that has been attested compliant with the code of conduct for such apps. (Today this would be the Apple or Android app store.)
3. Patient launches the app and picks a friendly name for a new identifier that will be assigned by an associated process.
4. Patient asks the app to "bind" the new identifier to the Mobile Identity Assurance claim that is also on the paper received in step one.
5. The MAAS is sent with other identification information to a CSP for registration of this device with this MAAS.
6. The user can send this assurance from the CSP to any relying party to prove that their authentication of themselves meets NIST SP 800-63B criteria.

A different path using biometrics: is under development.

Failed Paths:

1. Patient has no tolerance for technology and ignores or misunderstands the instructions or the purpose of the exercise.

## Results

Accepted Risks:

1. The Patient loses the paper allowing some other person to attempt to steal their identity - mitigated by sign up process as described.
2. Recovery TK

Post Condition:

1. If validation accepted by the CSP, the Patient has a phone that can be used for sign in to any participating healthcare provider.

Dependencies:

1. Web Sites must be trusted before any user information is released.
2. Trust federations can be used to help users make informed decisions.
3. User consent and trust must begin with no user information transferred.
4. Standards exist to collect needed attributes where-ever they may be.

## User Preparation of the Device for Use

This message is sent by a user agent app on the user’s phone with some information known to the user to assure the CSP that the message comes from the user, and a software statement to indicate the level of protection and user-presence is adequate assurance of authentication level 2 (AAL2).

**Registration Ceremony**

The user needs to install the app on their mobile device before completing this step. The instructions from the EHR will tell the user how to acquire the app from the app store specific to their phone supplier. After the agent app is running the user will chose to create an identifier and add a binding of that identifier to the CSP. For IAL2 they will need the DIA code from the EHR as described above. For authentication assurance (AAL2) they will need to establish that their identifier is bound to a private key held in the Trusted Execution Environment on the Phone, called the KeyStore on Android. The Agent needs to have its own certificate informing the CSP that the app can be trusted to reliably report this information as well as user consent to proceed. The application MUST provide information to the CSP to prevent use of the code by anyone other than the Patient or guardian. The following is an example of one implementation of the user experience in providing proof that they are entitled to access the EHR record on themselves.

**User Consent**

Note – must have notification address email or SMS phone number

The following is a non-normative example of what might be displayed to the user. This is only used to verify that this is the user that is identified in the activation code.

Non-normative Example:

{

"iss": "https://trustregistry.us/csp",

"roles": "ial2 aal2",

"sub": "cc395e3a-32f1-4b90-8cf6-a1e087fca9e6",

"personal\_info": {

"email": "way2long@nowhere",

"phone\_number": "1234567898",

"birth\_date": "20200217"

},

"doi": "net.azurewebsites.controls.5741770546174656480",

"iat": 1576358115,

"exp": 1582414622,

"nonce": 1712055122,

"jwk": {

"kty": "RSA",

"alg": "RS256",

"e": "AQAB",

"n": "ofgWCu…b4p4fAkd"

}

}

## Registration Certificate returned to the user Device

This certificate from the CSP is what makes the phone a credential with IAL2 and AAL2 assurance. The user is informed by their agent app that the binding of their credential to IAL2 and AAL2 has been successful, or given instructions on how to remedy the problem.

Note that actual JSON canonicalized form of these message would have the MAAS in JSON canonicalized form (see RFC 8785) and jose (see RFC 7515) encoding with a header and signature. The example shown below expands the MAAS expanded into its component parts but would be transmitted encoded.

Non-normative example:

{

"iss":"https://trustregistry.us/csp",

"roles":"ial2 aal2",

"sub":"cc395e3a-32f1-4b90-8cf6-a1e087fca9e6",

"nbf":1581980372,

"exp":1582412367,

"nonce":730747597,

"trust\_framework":"US Healthcare Assurance Framework",

"status":"success",

"iat":1576358115,

"maas": {

"name": "us.trustworthy.agent",

"version": "1",

"platform": "Android",

"min\_platform": "23",

"source": null,

"jurisdiction": "us-wa",

"user\_authn": null,

"date": 1576358115,

"url": "https://trustregistry.us/csp",

"trust\_registry": "US Healthcare Assurance Framework"

},

"jwk": {

"kty": "RSA",

"alg": "RS256",

"e": "AQAB",

"n": "zY0FA1lpMoJy3RLWb……KUB3EIh/9npVvf3J3w=="

}

# Contents of messages

## Contents of Mobile Authentication Assurance Statement (MAAS)

|  |  |  |  |
| --- | --- | --- | --- |
| **Mobile Assurance Statement** Contents of the Mobile Assurance Statement | | | |
| Field Name | Definition | Guidance | Required |
| Header | Part of the jose structure | {alg:RSA} | MUST |
| iat | Date that the MAAS was created | This field is partially intended to create a unique Id for the doc. It is calculated at printing time. | MUST |
| Name | "us.trustworthy.agent", | Name of app as known to app store | MUST |
| Version | integer | Assigned by the app author – monotonically increasing | MUST |
| Platform | "Android" | Source of the operating system | MUST |
| Min Platform | 23 | Required minimum functionality | MAY |
| Author | URN | Name of the responsible author | MUST |
| Device | Identifier reported by the device |  | MAY |
| Source | null |  | MAY |
| Jurisdiction | "us-wa", | Typically the ISO country and state. | MUST |
| URL | "https://trustregistry.us/csp", |  | MUST |
| Trust Federation | Healthcare Assurance Framework" | Pointer to the code of conduct | MUST |
| Signature | JWS Part of the jose structure | Signed by an authority | MUST |

Table 1: Mobile Assurance fields

TK - Data structure will be provided after the schema is fixed

Figure 3: Mobile Assurance data structure

# Guidance

## Presentation and Delivery

Although an Assurance can be provisioned in any manner that is feasible or expected based on the context, an Assurance MUST be provided to the Subject of data (the Patient in the case) or the Guardian in a human-readable format either on screen, or delivered to the data subject, or both. A JSON encoded Consent Receipt (CR} SHOULD also be delivered to the PII Principal.

NOTE: Issues such as language translation, localization, human-readable layout and formatting, and delivery mechanisms are out-of-scope for this document.

## User Experience with High Assurance UX

One of the major challenges with getting user acceptance of high assurance projection … TK

## Open Issues:

1. How does the user agent determine which CSP to call?
2. How should we specify anti-hammering requirements?
3. An example protocol flow where the statement is used.

# Mobile Assurance - JSON

## JSON Fields

This specification uses “named object” data types to describe the principal concepts within the Mobile Assurance and allows for extension by implementers.

See the JSON schema for object implementation.

| **JSON name** | **CR name** | **Data Type** | **Format/Example** |
| --- | --- | --- | --- |
| version | Version | string |  |
|  |  |  |  |
|  |  |  |  |

Table 2: Mobile Assurance JSON fields

## JSON Schema

{

TBD

{

# Conformance

tbd

# Considerations

Assurance level is a measure used by digital systems to evaluate claims about identity, security and privacy. This specification addresses protocol interoperability of assurance information as the primary objective. The following considerations are needed for a complete solution and are offered here as suggested guidelines.

With each ecosystem policy and a Mobile Assurance implementation, there are different UX, legal, privacy, and security-related considerations. This document uses an example from health care but that would not apply in other federations.

## Mobile Authentication Assurance User Experience

A mobile auth protects the user’s private key credential

A mobile auth allows the user to carry their credential with them and use it to gain access to their protected resources both in the physical and digital world

A mobile auth provides assurances to the service provider that only the one true end-user is in possession of the credential

### Identifiers (non-normative)

It is the objective of the user's agent to store authentication credentials and other user claims needed to provide the user with the level of assurance for identity and authentication needed for demanding web sites. Each persona that the user creates has the level of support that it requires. For example the following personas might be appropriate for Fred if he want to have a normal persona, as well as a high assurance medical persona and to act as guardian for his under 13 year-old daughter and a parent who has arranged for him to act as guardian.

1. Fred-personal = a common self-issued persona with no special assurances
2. Fred-medical = a high assurance self-issued persona with level 2 identity and authentication assurances for access to his own medical records.
3. Daughter-annie = a high assurance self-issued (by Fred) persona with level 2 identity and authentication assurances and access to Annie's medical records.
4. Grandfather-john = a high assurance self-issued (by Fred) persona with level 2 identity and authentication assurances for access to Fred's father's medical records.

While many use cases will have multiple personas as shown above, there is no reason that a single persona user agent should not be produced as well. In any case each persona must be able to handle identity assurances from any compliant provider, and record locators at any number of other compliant providers. Acting as identifier provider for the user, the agent in the user's device may also crate any number of subject identifiers spanning a range of networks including decentralized identifiers (DIDs) defined over block chains; but none of this complexity need to be normally visible to the user. It can all be hidden inside user-opaque digital protocols.

### Consent Data Categories

While it is not clear what data categories will be in the various data stores as each data ecosystem will likely create their own, for the US Healthcare system the best information available at the time this was written point to the data sensitivity levels from the FHIR documentation which have been summarized [here](https://wiki.idesg.org/wiki/index.php/Trustworthy_Healthcare_Provider#Data_Categorization).

## Privacy Considerations

The major objective of this section is to assure that solution can be configured by most developers to meet the privacy requirements of the evolving privacy legislation as well as the IDEF guidelines. TK link to IDEF.

In this document, sensitive data collection is indicated. NOTE: In multiple jurisdictions, there are categories listed as sensitive personal information. If you use, collect or disclose sensitive personal information these have legal requirements, require explicit consent and can have jurisdiction-specific legal notice requirements to be informed. For example, user information about sexually transmitted diseases would need particular care before being released.

For example the sensitivity levels are described in [Trustworthy Healthcare Provider](https://wiki.idesg.org/wiki/index.php/Trustworthy_Healthcare_Provider) document.

## Security Considerations

The major objective of this section is to assure that solution can be configured by most developers to meet the security requirements of the NIST 800-63-3B level AAL2 as well as the IDEF guidelines. The challenge is to encode the authentication assurances in a manner that can be reliably tested by the CSP and passed on to the relying parties. The specification relies on cryptographically signed JSON Web Tokens (JWT) documents and does not depend on TLS (HTTPS) in order to establish trust. TLS may still be advisable to prevent leakage of private information.

### Private Key Protection

All user private keys must be enabled for AAL2 protection if requested. To prevent a stolen user device from allowing unauthorized use of private key, the user must be authenticated to the device before the private (or secret) keys are made available for use.

#### Android key protection (non-normative and may change over time)

The best working API for creating a Private Key in Android appears to be KeyGenParameterSpec added in Android M (API 23) as described in this post.<https://doridori.github.io/android-security-the-forgetful-keystore/#sthash.NijopPIO.dpbs> It is required that the Android device is protected with a device-lock set to prevent access without the code. This is a good security feature, but requires a minimum level of Android M. Fingerprint support is not solid until Android N.

#### Apple Key Protection (non-normative and may change over time)

### Protection for messages transmitted by any actor

The transmission of a JSON Mobile Assurance MUST enable validation of the integrity and authenticity of the receipt using the following specifications:

* JSON Web Token (JWT) [RFC 7519]
* JSON Web Encryption (JWE) [RFC 7516]
* JSON Web Signature (JWS) [RFC 7515]
* FHIR JSON Signature rules in [HL7 section 2.23.0.17.2](https://www.hl7.org/fhir/datatypes.html#signature) which contains options which are consistent with this specification. Note that detached signature are used in the listed messages, but not for the messages in this specification which are significantly smaller.

Notices

# Acknowledgements

The Mobile Assurance effort has been developed in the Kantara Community, supported by people who have invested in making this specification open and free to use. It is free so that people can have a common way to see their data control and sharing. If you wish to provide feedback, you may join the Kantara Working Group, and then email us tbd

In addition to Kantara, we wish to thank the following contributors to the Mobile Assurance effort:

list of names

The Mobile Assurance specification effort has been developed with the support of many communities, as noted in our acknowledgments section, and leverages best of breed standards, legal regulation and technical practices in its design and development, as noted in the references section.

# Works Cited

Group, K. I. (2019, 4 30). *Kantara Identity Assurance Framework: IAF-1050 2 Glossary and Overview* . Rtrieved from Kantara Initiative: https://kantarainitiative.org/confluence/display/IAWG/IAF+1050+-+Glossary+and+Overview?preview=/113869056/113869057/Kantara%20IAF-1050%20Overview%20Glossary%20v0.5.4.pdf

9, May 2015, <https://tools.ietf.org/html/rfc7519>  **[RFC 2119]** Bradner, S., “*Key words for use in RFCs to Indicate Requirement Levels*”, BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997 <http://www.rfc-editor.org/info/rfc2119>

**[RFC 4122]** P. Leach, M. Mealling, R. Salz, “*A Universally Unique IDentifier (UUID) URN Namespace*”, RFC 4122, 10.17487/RFC4122, July 2005, <https://tools.ietf.org/html/rfc4122>

**[RFC 6749]** D. Hardt, “*The OAuth 2.0 Authorization Framework*”, RFC 6749, 2010-10 <https://tools.ietf.org/html/rfc6749#section-5.2>

**[RFC 7159]** Bray, T., Ed., “*The JavaScript Object Notation (JSON) Data Interchange Format*”, RFC 7159, DOI 10.17487/RFC7159, March 2014, <http://www.rfc-editor.org/info/rfc7159>

**[RFC 7515]** M. Jones, J. Bradley, N. Sakimura, “*JSON Web Signature (JWS)*”, RFC 7515, May 2015, <https://tools.ietf.org/html/rfc7515>

**[RFC 7516]** M. Jones, J. Hildebrand, “*JSON Web Encryption (JWE)*”, RFC 7516, May 2015, <https://tools.ietf.org/html/rfc7516>

**[RFC 7519]** M. Jones, J. Bradley, N. Sakimura, “*JSON Web Token (JWT)*”, RFC 7519, DOI 10.17487/RFC751

[JOSE] Javascript Object Signing and Encryption (jose) <https://datatracker.ietf.org/group/jose/documents/>

1. Categories of Data

(Explainers/Examples)

1. Example Mobile Assurance messages
   1. Human-readable Mobile Assurance– Simple

* 1. Human-readable Mobile Assurance– Fancy
  2. JSON Contents

{

}

# Revision history

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Summary of Substantive Changes** |
| 1.0.0 DRAFT 1 | 2020-08-25 | Initial v1.0 draft |
| DRAFT 2 | 2020-10-06 | Implementor’s Report with editorial changes |