

OmniOne Digital ID

OmniOne Digital ID is an open source project that aims to drive global adoption and expansion of self-sovereign identity through a secure identity system based on the technology that built the Republic of Korea's national mobile ID.

OmniOne Digital ID is an open source decentralized identifier (DID) designed to provide developers with the tools they need to integrate self-sovereign identity management into their applications, built to be secure, scalable, and interoperable by adhering to the latest DID standards.

Goals

- We aim to provide a digital identity authentication system through the OPEN DID platform based on trust and responsibility to various members of the digital world.
- Through this, we reduce social costs and enable the socially excluded from the infrastructure to fulfill their "social responsibility for realizing human values."

Structure

The OPEN DID system consists of a Trust Environment for establishing a trust chain between participating entities, a Wallet for storing and managing various certificates based on Digital ID, a Digital ID that includes various certificates, and Security for data protection.

- Trust Environment: The components of OPEN DID define and develop methods for establishing a chain of trust between components to build a trust environment based on Digital IDs.
- Wallet: Define the lifecycle of the wallet and e-signs and develops the architecture for key management, storage management, access control management, and security management of the wallet.
- Digital ID: Define a data model based on W3C's Decentralized Identifier and Verifiable Credentials specifications.
- Security: Ensures a level of security that ensures the authenticity, confidentiality, and integrity of data exchanged between OPEN DID components.

Getting Started

Prerequisites

Before you begin, ensure you have met the following requirements:

- Node.js (version 14 or later)
- npm (version 6 or later)
- Git (for cloning the repository)
- A basic understanding of decentralized identities and blockchain technology is recommended.

Installation

To install OmniOne, follow these steps:

1. Clone the Repository:

```
git clone https://github.com/yourusername/OmniOne.git
cd OmniOne
```

2. Install Dependencies:

```
npm install
```

Usage

OmniOne Digital ID provides a set of commands to manage DIDs. Below are some basic usage examples:

- Start the DID Service:

To start the OmniOne DID service:

```
npm start
```

- Create a New DID:

To create a new DID:

```
npm run create-did
# Output: did:omni:123456789abcdefghi
```

- Resolve an Existing DID:

To resolve and retrieve details of an existing DID:

```
npm run resolve-did did:omni:123456789abcdefghi
# Output: { "@context": "https://www.w3.org/ns/did/v1", "id":
"did:omni:123456789abcdefghi", ... }
```

Examples

Here's how you can use OmniOne Digital ID in your project:

- Integrating with a Web Application: OmniOne can be integrated into a web application to enable users to authenticate using their DIDs.
- Cross-Chain DID Resolution: Utilize OmniOne's interoperability features to resolve DIDs across different blockchain networks.

Contributing

We welcome and appreciate contributions from the community! To contribute to OmniOne Digital ID:

1. Fork the repository.
2. Create a new branch (`git checkout -b feature/your-feature`).
3. Make your changes.
4. Commit your changes (`git commit -m 'Add some feature'`).
5. Push to the branch (`git push origin feature/your-feature`).
6. Open a Pull Request.

Please ensure that your code adheres to our [contribution guidelines](#) and is covered by tests where appropriate.

License

This project is licensed under the Apache License 2.0. See the [LICENSE](#) file for details.
Apache License 2.0 : <http://www.apache.org/licenses/LICENSE-2.0>

Acknowledgments

OmniOne Digital ID was made possible thanks to the contributions of the open-source community and the support of various partners.

Contact

For further inquiries or support, please contact us at technical@omnione.com.

Additional Resources

- Documentation: Detailed documentation is available in the [docs](#) folder.
- Issue Tracking: Please report issues or request features via the [GitHub Issues](#) page.
- Community Discussions: Join our [community forum](#) to discuss OmniOne and decentralized identity.

Roadmap

OmniOne Digital ID is under active development. Here are some features we're planning to add:

- Support for additional DID methods
- Enhanced privacy features
- Integration with more blockchain networks
- Improved scalability and performance

Stay tuned for updates!

Android Wallet Core SDK

Welcome to the Android Wallet Core SDK Repository.
This repository provides an SDK for developing an Android mobile wallet.

Folder Structure

```
opendid-core-sdk-aos |— CHANGELOG.md |— CLA.md |— CODE_OF_CONDUCT.md
```

Name	Description
source	SDK source code project
docs	Documentation
└─ api	API guide documentation
└─ design	Design documentation
sample	Samples and data
README.md	Overview and description of the project
CLA.md	Contributor License Agreement
CHANGELOG.md	Version-specific changes in the project
CODE_OF_CONDUCT.md	Code of conduct for contributors
CONTRIBUTING.md	Contribution guidelines and procedures
LICENSE.dependencies.md	Licenses for the project's dependency libraries
MAINTAINERS.md	General guidelines for maintaining

RELEASE-PROCESS.md	Release process
SECURITY.md	Security policies and vulnerability reporting

Libraries

Libraries can be found in the [releases folder](#).

1. Copy the `opendid-core-sdk-aos-1.0.0.jar` file to the libs of the app project.
2. Add the following dependencies to the build.gradle of the app project.

```
implementation files('libs/opendid-core-sdk-aos-1.0.0.jar')
```

3. Sync `Gradle` to ensure the dependencies are properly added.

API Reference

API Reference can be found [here](#)

Contributing

Please read [CONTRIBUTING.md](#) and [CODE_OF_CONDUCT.md](#) for details on our code of conduct, and the process for submitting pull requests to us.

License

Copyright 2024 Raonsecure

Android Wallet Core SDK Guide

This document is a guide for using the OpenDID Wallet Core SDK, providing functions to generate, store, and manage the keys, DID Document, and Verifiable Credential (VC) information required for Open DID.

S/W Specifications

Component	Requirement
OS	Android 13
Language	Java 17
IDE	Android Studio 4
Build System	Gradle 8.2
Compatibility	Android API level 33 or higher

Build Method

: Execute the export JAR task in the build.gradle file of this SDK project to generate a JAR file.

1. Open the project's `build.gradle` file and add the following `export JAR` task.

```
ext {  
    version = "1.0.0"}  
task exportJar(type: Copy){  
    from
```

2. Open the `Gradle` window in Android Studio, and run the `Tasks > other > exportJar` task of the project.

3. Once the execution is complete, the `opendid-core-sdk-aos-1.0.0.jar` file will be generated in the `release/` folder.

SDK Application Method

1. Copy the `opendid-core-sdk-aos-1.0.0.jar` file to the libs of the app project.
2. Add the following dependencies to the build.gradle of the app project.

```
implementation files('libs/opendid-core-sdk-aos-1.0.0.jar')
```

3. Sync `Gradle` to ensure the dependencies are properly added.

API Specification

Classification	API Document Link
KeyManager	Wallet Core SDK - KeyManager API
DIDManager	Wallet Core SDK - DIDManager API
VCManager	Wallet Core SDK - VCManager API
SecureEncryptor	Wallet Core SDK - SecureEncryptor API
ErrorCode	Error Code

KeyManager

KeyManager provides the functionality to generate and manage key pairs for signing and store them securely.

The main features are as follows:

- **Key Generation:** Generates a new key pair.
- **Key Storage:** Securely stores the generated key.

- **Key Retrieval:** Retrieves the stored key.
- **Key Deletion:** Deletes the stored key.

DIDManager

DIDManager provides the functionality to generate and manage DID Documents.

The main features are as follows:

- **DID Generation:** Generates a new DID.
- **DID Document Management:** Creates, updates, and deletes DID Documents.
- **DID Document Retrieval:** Retrieves information about DID Documents.

VCManager

VCManager provides the functionality to manage and store Verifiable Credentials (VC).

The main features are as follows:

- **VC Storage:** Securely stores the generated VC.
- **VC Retrieval:** Retrieves the stored VC.
- **VC Deletion:** Deletes the stored VC.

SecureEncryptor

SecureEncryptor provides the functionality to encrypt and decrypt data using the Keystore.

The main features are as follows:

- **Data Encryption:** Securely encrypts the data.
- **Data Decryption:** Decrypts the encrypted data.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein

shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [2024] [Raonsecure]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

```
http://www.apache.org/licenses/LICENSE-2.0
```

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Contribution Guidelines

We love your input! We want to make contributing to this project as easy and transparent as possible, whether it's:

- Reporting a bug
- Discussing the current state of the code
- Submitting a fix
- Proposing new features

We Develop with Github

We use github to host code, to track issues and feature requests, as well as accept pull requests.

We Use [Github Flow](#), So All Code Changes Happen Through Pull Requests

Pull requests are the best way to propose changes to the codebase (we use [Github Flow](#)). We actively welcome your pull requests:

1. Fork the repo and create your branch from `master`.
2. If you've added code that should be tested, add tests.
3. If you've changed APIs, update the documentation.
4. Ensure the test suite passes.
5. Make sure your code lints.
6. Issue that pull request!

Report bugs using Github's [issues](#)

We use GitHub issues to track public bugs. Report a bug by [opening a new issue](#); it's that easy!

Write bug reports with detail, background, and sample code

Bug Reports tend to have:

- A quick summary and/or background
- Steps to reproduce
 - Be specific!
 - Give sample code if you can.
- What you expected would happen
- What actually happens
- Notes (possibly including why you think this might be happening, or stuff you tried that didn't work)

People *love* thorough bug reports. I'm not even kidding.

Contributor License Agreement ("CLA")

In order to accept your pull request, we need you to submit a CLA. You only need to do this once to work on any of OpenDID open source projects.

Complete your CLA here: [Contributor License Agreement](#)

Coding Style

Our code style guidelines are based on the [OpenDID Coding Style](#).

Commit Message Guidelines

Our commit rule guidelines are based on the [OpenDID Commit Rule](#).

License

Copyright 2024 Raonsecure

Contributor Covenant Code of Conduct

Our Pledge

We as members, contributors, and leaders pledge to make participation in our community a harassment-free experience for everyone, regardless of age, body size, visible or invisible disability, ethnicity, sex characteristics, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, religion, or sexual identity and orientation.

We pledge to act and interact in ways that contribute to an open, welcoming, diverse, inclusive, and healthy community.

Our Standards

Examples of behavior that contributes to a positive environment for our community include:

- Demonstrating empathy and kindness toward other people
- Being respectful of differing opinions, viewpoints, and experiences
- Giving and gracefully accepting constructive feedback
- Accepting responsibility and apologizing to those affected by our mistakes, and learning from the experience
- Focusing on what is best not just for us as individuals, but for the overall community

Examples of unacceptable behavior include:

- The use of sexualized language or imagery, and sexual attention or advances of any kind
- Trolling, insulting or derogatory comments, and personal or political attacks

- Public or private harassment
- Publishing others' private information, such as a physical or email address, without their explicit permission
- Other conduct which could reasonably be considered inappropriate in a professional setting

Enforcement Responsibilities

Community leaders are responsible for clarifying and enforcing our standards of acceptable behavior and will take appropriate and fair corrective action in response to any behavior that they deem inappropriate, threatening, offensive, or harmful.

Community leaders have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, and will communicate reasons for moderation decisions when appropriate.

Scope

This Code of Conduct applies within all community spaces, and also applies when an individual is officially representing the community in public spaces. Examples of representing our community include using an official e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event.

Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported at [opendDID.to](https://openddid.to) to the community leaders responsible for enforcement.

All complaints will be reviewed and investigated promptly and fairly.

All community leaders are obligated to respect the privacy and security of the reporter of any incident.

Enforcement Guidelines

Community leaders will follow these Community Impact Guidelines in determining the consequences for any action they deem in violation of this Code of Conduct:

1. Correction

Community Impact: Use of inappropriate language or other behavior deemed unprofessional or unwelcome in the community.

Consequence: A private, written warning from community leaders, providing clarity around the nature of the violation and an explanation of why the behavior was inappropriate. A public apology may be requested.

2. Warning

Community Impact: A violation through a single incident or series of actions.

Consequence: A warning with consequences for continued behavior. No interaction with the people involved, including unsolicited interaction with those enforcing the Code of Conduct, for a specified period of time. This includes avoiding interactions in community spaces as well as external channels like social media. Violating these terms may lead to a temporary or permanent ban.

3. Temporary Ban

Community Impact: A serious violation of community standards, including sustained inappropriate behavior.

Consequence: A temporary ban from any sort of interaction or public communication with the community for a specified period of time. No public or private interaction with the people involved, including unsolicited interaction with those enforcing the Code of Conduct, is allowed during this period. Violating these terms may lead to a permanent ban.

4. Permanent Ban

Community Impact: Demonstrating a pattern of violation of community standards, including sustained inappropriate behavior,

harassment of an individual, or aggression toward or disparagement of classes of individuals.

Consequence: A permanent ban from any sort of public interaction within the community.

Attribution

This Code of Conduct is adapted from the [Contributor Covenant](https://www.contributor-covenant.org/version/2/0/code_of_conduct.html), version 2.0, available at https://www.contributor-covenant.org/version/2/0/code_of_conduct.html.


Community Impact Guidelines were inspired by [Mozilla's code of conduct enforcement ladder](#).

For answers to common questions about this code of conduct, see the FAQ at <https://www.contributor-covenant.org/faq>. Translations are available at <https://www.contributor-covenant.org/translations>.

Security Policy

Supported Versions

In this project, the following versions have undergone security vulnerability assessments:

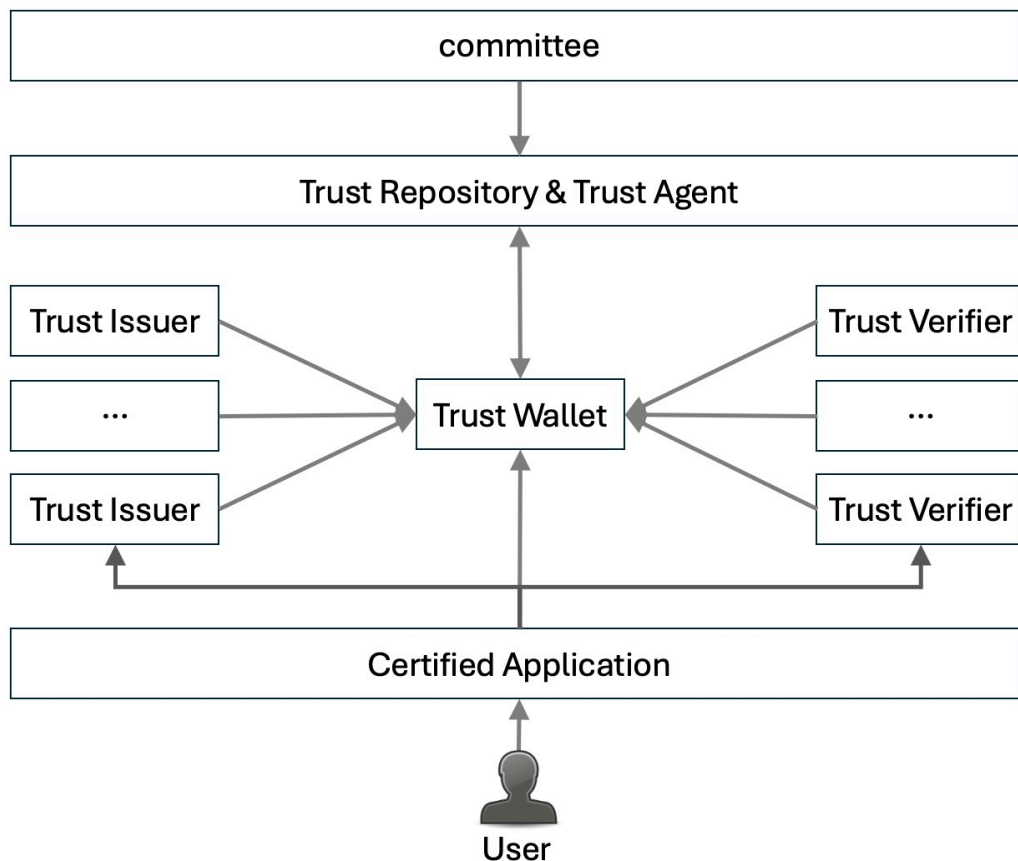
Version	Supported
1.0.0	

Reporting a Vulnerability

If you discover a new security vulnerability, please report it via [opendid mail](#).

Ecosystem of OmniOne Open DID

The underlying illustration is the architecture of the OmniOne Open DID ecosystem and process overview flow between various roles:



The roles are :

1. Trust Repository

A trust Repository operated by a committee guarantees the role of each entity.

2. Trust Agent

A role delegated by the committee to resolve Oracle issues when registering entities in a trust Repository.

3. Trust Wallets

A user wallet where various credentials can be stored. There may be various level of assurance depending on their reliability.

4. Trust Issuers

It is role for proof the user's credentials. Like wallets, there may be issuing agencies with various levels of assurance.

5. Trust Verifiers

It is role for verifying the user's credentials.

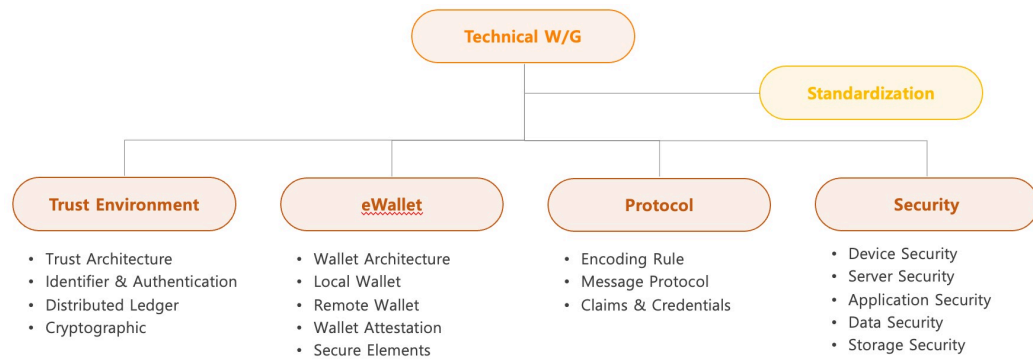
6. Certified applications

Users can interact with various entities, including wallets, through the Certified applications.

Roadmap

Track	2024	2025				2026			
	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
Technical	SDK & Sample Launch	Improvement use cases		Research Interoperability(ISO/IEC)		mDoc launch			
		Product(dual license)							
Standardization	KS Standardization			Global Standardization					
Operation	Community & improvement								
	Seminar		Support project						
	Technical Working group			Sub Group					

* refer. Technical working group



Maintainers

About This File

You can find out who's contributed recently just by looking at GitHub's [contributors list](#). But there are a few more things you ought to know about who maintains this code, and how they do it:

1. Make sure you read our [contributor guidelines](#) so you understand how we work and how to collaborate effectively. This includes instructions about pull request and code review protocols, and it explains what we mean by calling someone a "maintainer" in this file.
2. Be aware that individual folders in the project may have more specific maintainers; if you see another `MAINTAINERS.md` in a subfolder, that governs the tree below it.

Who To Contact

For ordinary questions, we suggest you contact [active contributors](#)

Maintainers

Name	GitHub	email
Sangjun Kim	sjkim	sjkim@raoncorp.com

Emeritus Maintainers

Name	GitHub	email

Changelog

v1.0.0 (2024-08-22)

New Features

- DID Document management(Generation, Retrieval, Deletion)
- VerifiableCredential management(Storage, Retrieval, Deletion)
- VerifiablePresentation generation
- Key management for encryption, decryption and signing

OmniOne Governance

- This document outlines the governance structure, decision-making processes, and roles within the OmniOne project.
- This governance model is inspired by the Apache Software Foundation's practices to ensure transparency, meritocracy, and sustainability

1. Overview

- The OmniOne project is an open-source initiative under the Apache 2.0 License.
- The project is governed by a group of contributors, committers, and a Project Management Committee (PMC) that ensures the project's long-term health and success.

2. Roles and Responsibilities

2.1. Contributors

- Definition: Anyone who contributes code, documentation, or other valuable resources to the project.
- Responsibilities:
 - Submit patches or pull requests.
 - Participate in discussions and reviews.
 - Adhere to the project's contribution guidelines.

2.2. Committers

- Definition: Contributors who have been granted write access to the project's repository due to their sustained contributions and involvement. Committer is composed of Technical Working Members from the Korea Digital Certification Association
- Responsibilities:
 - Review and merge contributions from other contributors.
 - Ensure that code quality and project standards are maintained.
 - Engage with the community to foster collaboration and growth.
 - Propose new features, changes, or improvements.

2.3. Project Management Committee (PMC)

- Definition: Contributors who have been granted write access to the project's repository due to their sustained contributions and involvement. The PMC is composed of Technical Working Members from the Korea Digital Certification Association
- Definition: A group of active committers who are responsible for the strategic direction, overall governance, and decision-making of the project.
- Responsibilities:
 - Set project policies and guidelines.
 - Make decisions on releases, roadmaps, and major changes.
 - Resolve disputes or issues within the community.
 - Represent the project in external communications and collaborations.

3. Decision-Making Process

3.1. Consensus-Based Decision Making

- Decisions are typically made by consensus. This means that after open discussion, if no committers or PMC members object, the decision is accepted.
- For significant changes, a formal vote may be conducted within the PMC.

3.2. Voting

- Lazy Consensus: If no objections are raised within a specified period (e.g., 72 hours), the proposal is accepted.
- Formal Votes: Major decisions, such as a new release or adding a new committer, require a formal vote with the following possible outcomes:
 - +1 (Yes): In favor of the proposal.
 - 0 (Neutral): No opinion or abstaining from the vote.
 - -1 (No): Opposed to the proposal (must provide reasoning).
- A simple majority is typically required for a proposal to pass.

4. Releases

- Releases are managed by the PMC, with input from the community.
- A release plan is proposed, discussed, and approved via consensus or formal vote.
- The release manager is responsible for coordinating the release process, ensuring that all criteria and quality standards are met.

5. Conflict Resolution

- In the event of a disagreement or conflict, the PMC will mediate and attempt to reach a resolution that is in the best interest of the project.
- If necessary, a formal vote within the PMC may be conducted to resolve the issue.

6. Community Guidelines

- All participants are expected to adhere to the project's Code of Conduct, fostering a respectful and inclusive environment.
- Contributions should align with the project's goals and quality standards.

7. Amendments

- This governance document may be amended by the PMC through a consensus or formal vote.
- All amendments should be documented and communicated to the community.

Release Process

This document outlines the steps involved in the release process for the Wallet core SDK project.

Versioning

The project follows the format `X.Y.Z` for versioning.

X (Major): In case of incompatibility with lower versions

Y (Minor): When new features are added while being compatible with lower versions

Z (Patch): Bug fixes and small changes while compatible with lower versions

When the major version is changed, minor and patch are initialized to 0.

When the minor version is changed, the patch is initialized to 0.

Releasing a New Version

1. Create an issue to define and track release-related activities.
Choose a title that follows the format `X.Y.Z`.
2. Stop merging any new work into the main branch.
3. Check the release draft under the [CHANGE LOG](#) page to ensure that everything is in order.
4. Create and push the release tag in the format `X.Y.Z`:

```
git tag -a vX.Y.Z -m "Release vX.Y.Z"git push origin vX.Y.
```

As a result, the CI/CD pipeline will publish the release.

Individual Software Grant and Contributor License Agreement

Based on the Apache Software Foundation Software Grant and Individual Contributor License Agreement

<http://www.apache.org/licenses/>

Thank you for your interest in the OpenDID project (the "Project"). In order to clarify the intellectual property license granted with Contributions (defined below) from any person or entity, the Project must have a Contributor License Agreement ("CLA") on file that has been signed by each Contributor, indicating agreement to the license terms below. This license is for your protection as a Contributor as well as the protection of the Project and its users; it does not change your rights to use your own Contributions for any other purpose.

You accept and agree to the following terms and conditions for Your present and future Contributions submitted to the Project. In return, the Project shall not use Your Contributions in a way that is contrary to the public benefit or inconsistent with its nonprofit status and bylaws in effect at the time of the Contribution. Except for the license granted herein to the Project and recipients of software distributed by the Project, You reserve all right, title, and interest in and to Your Contributions.

1. Definitions. "You" or "Your" shall mean the copyright owner or legal entity authorized by the copyright owner that is making this CLA with the Project. For legal entities, the entity making a Contribution and all other entities that control, are controlled by, or are under common control with that entity are considered to be a single Contributor. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether

by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity. "**Contribution**" shall mean the code, documentation or other original works of authorship expressly identified in Schedule A, as well as any original work of authorship, including any modifications or additions to an existing work, that is intentionally submitted by You to the Project for inclusion in, or documentation of, any of the products owned or managed by the Project (the "**Work**"). For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Project or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Project for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by You as "Not a Contribution."

2. Grant of Copyright License. Subject to the terms and conditions of this CLA, You hereby grant to the Project and to recipients of software distributed by the Project a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, sublicense, and distribute Your Contributions and such derivative works.

3. Grant of Patent License. Subject to the terms and conditions of this CLA, You hereby grant to the Project and to recipients of software distributed by the Project a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by You that are necessarily infringed by Your Contribution(s) alone or by combination of Your Contribution(s) with the Work to which such Contribution(s) was submitted. If any entity institutes patent litigation against You or any other entity (including a cross-claim or counterclaim in a lawsuit) alleging that your Contribution, or the Work to which you have contributed, constitutes direct or contributory patent infringement, then any patent licenses granted to

that entity under this CLA for that Contribution or Work shall terminate as of the date such litigation is filed.

4.Representations with Respect to the License and

Contributions. You represent that you are legally entitled to grant the above license. If your employer(s) has rights to intellectual property that you create that includes your Contributions, you represent that you have received permission to make Contributions on behalf of that employer, that your employer has waived such rights for your Contributions to the Project, or that your employer has executed a separate Corporate CLA with the Project.

5.Additional Representations with Respect to Contributions. You represent that each of Your Contributions is Your original creation (see section 7 for submissions on behalf of others). You represent that Your Contribution submissions include complete details of any third-party license or other restriction (including, but not limited to, related patents and trademarks) of which you are personally aware and which are associated with any part of Your Contributions.

6.Disclaimers. You are not expected to provide support for Your Contributions, except to the extent You desire to provide support. You may provide support for free, for a fee, or not at all. Unless required by applicable law or agreed to in writing, You provide Your Contributions on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE.

7.Non-Original Creations. Should You wish to submit work that is not Your original creation, You may submit it to the Project separately from any Contribution, identifying the complete details of its source and of any license or other restriction (including, but not limited to, related patents, trademarks, and license agreements) of which you are personally aware, and conspicuously marking the work as "Submitted on behalf of a third-party: [named here]."

8. Agreements. You agree to notify the Project of any facts or circumstances of which you become aware that would make these representations inaccurate in any respect.

License Information

This project uses the following third-party libraries, each under its own license:

Gson: Apache License 2.0

`com.google.code.gson:gson:2.10.1`

(<https://github.com/google/gson/blob/master/LICENSE>)

Spongy Castle Core: MIT License

`com.madgag.spongycastle:core:1.54.0.0`

(<https://github.com/rtyley/spongycastle/blob/spongy-master/LICENSE.html>)

Spongy Castle Provider: MIT License

`com.madgag.spongycastle:prov:1.54.0.0`

(<https://github.com/rtyley/spongycastle/blob/spongy-master/LICENSE.html>)

Spongy Castle PKIX: MIT License

`com.madgag.spongycastle:pkix:1.54.0.0`

(<https://github.com/rtyley/spongycastle/blob/spongy-master/LICENSE.html>)

Spongy Castle PG: MIT License

`com.madgag.spongycastle:pg:1.54.0.0`

(<https://github.com/rtyley/spongycastle/blob/spongy-master/LICENSE.html>)