OGC® DOCUMENT: 24-033
External identifier of this OGC® document: http://www.opengis.net/doc/PER/t20-D001

Open Geospatial Consortium

TESTBED-20: INTEGRITY, PROVENANCE, AND TRUST (IPT) REPORT

ENGINEERING REPORT

PUBLISHED

Submission Date: 2024-10-17
Approval Date: 2024-11-08
Publication Date: 2025-04-02
Editor: Paul Churchyard

Note

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

CONTENTS

I. KEYWORDS vi

II. CONTRIBUTORS vi

III. OVERVIEW vi

IV. FUTURE OUTLOOK vii

V. VALUE PROPOSITION viii

1. INTRODUCTION 2
   1.1. Aims 2
   1.2. Objectives 3

2. TOPICS 5
   2.1. Secure Dimensions – D100 IPT Server Instance 5
   2.2. Spacebel – D100 IPT Server Instance 10
   2.3. UseCases 19
   2.4. Technical Interoperability Experiments 24

3. OUTLOOK 30
   3.1. Integrity, Provenance, and Trust Roadmap 30
   3.2. VC/VP Compliant Self-Sovereign Identity 30
   3.3. Provenance Outlook 31

4. SECURITY, PRIVACY AND ETHICAL CONSIDERATIONS 33
   4.1. Demonstrated FACTS Ecosystem Privacy Considerations 33
   4.2. Demonstrated FACTS Ecosystem Ethical Considerations 34
   4.3. Data Privacy Considerations 34

BIBLIOGRAPHY 36

ANNEX A (NORMATIVE) ABBREVIATIONS/ACRONYMS 39

ANNEX B (INFORMATIVE) SECURE DIMENSIONS – FACTS ECOSYSTEM 42
   B.1. Trusted Teapot Process Example Plot 42
   B.2. FACTS Architecture and Implementation 46
   B.3. The Trusted Teapot Process 54
   B.4. Developers View: Smart Certificate issuing and verification 57
   B.5. Trusted Watermarking Process Execution Example 86

ANNEX C (INFORMATIVE) SPACEBEL — IPT SERVER ARCHITECTURE COMPONENT EXAMPLES 89
C.1. OGC/STAC Catalog Response Example 89
C.2. W3C Decentralized Identifier Document Example 89
C.3. W3C Verifiable Credentials Example 90
C.4. W3C Verifiable Presentations Example 92
C.5. Univerifier VC and VP Validation 95

ANNEX D (INFORMATIVE) SPACEBEL — IPT SERVER DEMO NOTEBOOK 97
D.1. Introduction 97
D.2. Create a DID 98
D.3. Resolve DID 99
D.4. Create a VC 100
D.5. Verify a VC 104
D.6. Create a VP 105
D.7. Verify a VP 109
D.8. Content-integrity Protection 109
D.9. Interactions with Catalogue 113
D.10. Interactions with Indy instance 124

ANNEX E (INFORMATIVE) EU SATCEN — ADDITIONAL USE CASE INFORMATION 130
E.1. Situation Report Use Case Workflow Visuals 130
E.2. FACTS EU Sat-Cen Examples 134
E.3. Overview of Data Usage Conditions from ESA/NASA and EU Regulations 135
E.4. EU 135
E.5. ESA 136
E.6. NASA 136

LIST OF TABLES

Table 2 — TIE Testing Matrix 25

LIST OF FIGURES

Figure 1 — Trusted Teapot Execution Process Flow 7
Figure 2 — Interaction Diagram 9
Figure 3 — Deployment Diagram 11
Figure 4 — Component Diagram 12
Figure 5 — OGC/STAC Catalog Resource Diagram 13

Figure 6 — Structure of a Verifiable Presentation 17
Figure 7 — Sequence diagram VC/VP verifications 18
Figure 8 — Sentinel 2 Catalgue query and AI superresolution 21
Figure B.1 — Trusted Teapot creation of a certificate for the source code 43
Figure B.2 — Trusted Teapot creation of a certificate for the deployment of the process 44
Figure B.3 — Searching for the Trusted Teapot 45
Figure B.4 — Trusted Teapot Execution with Certificates 45
Figure B.5 — Trusted Teapot Execution without Certificates 46
Figure B.6 — FACTS Provenance and Immutable Catalog Architecture 47
Figure B.7 — FACTS Certification Architecture 48
Figure B.8 — FACTS Detailed Architecture 49
Figure B.9 — FACTS Issuer Controller - Main Menue 49
Figure B.10 — FACTS Holder Controller - Main Menue 50
Figure B.11 — FACTS Issuer Controller - Issuing a Certificate 51
Figure B.12 — FACTS Holder Controller - Inspecting Certificates 52
Figure B.13 — FACTS Verifier Controller - Main Menue 53
Figure B.14 — FACTS Verifier Controller - Inspecting Proof Requests 53
Figure B.15 — FACTS Verifier Controller - Creating a Proof Request 54
Figure B.16 — FACTS Detailed Architecture including the Trusted Teapot Process 55
Figure B.17 — FACTS Trusted Teapot Process - Request Example 56
Figure B.19 — QUICKLOOK Image 87
Figure B.20 — Watermarked QUICKLOOK Image 87
Figure C.1 — Univerifier 95
Figure E.1 — EU SatCen reprojection workflow 130
Figure E.2 — EU SatCen gmljp2 vector plus image workflow 131
Figure E.3 — EU SatCen smart certificate issuer 132
Figure E.4 — Document FACTS compliant 133

I KEYWORDS

The following are keywords to be used by search engines and document catalogues.

testbed, integrity, provenance, trust, IPT

II CONTRIBUTORS

All questions regarding this document should be directed to the editor or the contributors:

NAME	ORGANIZATION	ROLE
Paul Churchyard	HSR.health	Editor
Lucio Colaiacomo	EU Sat-Cen	Contributor
Greg Buehler	OGC	Contributor
Carl Reed	OGC	Contributor
Andreas Matheus	Secure Dimensions	Contributor
Yves Coene	Spacebel	Contributor
Philippe Duchesne	Spacebel	Contributor
Iain Burnell	UK DSTL	Contributor
Christopher Budas	UK DSTL	Contributor

III OVERVIEW

In an era of growing data volumes and geospatial analysis demands, maintaining Integrity, Provenance, and Trust (IPT) is critical, especially in distributed systems where data availability can fluctuate. The OGC Testbed 20 IPT activity addresses this need by developing resilient data services that safeguard IPT throughout the data lifecycle. Two IPT Server instances were

demonstrated in the Testbed 20 IPT activity by Secure Dimensions, which used the Federated Agile Collaborative Trusted System (FACTS) to manage Smart Certificates, and Spacebel, which leveraged Decentralized Identifiers (DIDs) and Verifiable Credentials (VCs) for data verification. These servers ensure that data remains reliable and traceable by validating its origin and history, supporting cross-organizational trust without altering legacy data formats.

The FACTS ecosystem that was demonstrated by Secure Dimensions provides a trusted process for managing data through Smart Certificates. The two processes demonstrated, Trusted Teapot and Trusted Watermarking, validated certificates, maintained data integrity, and generated new certificates for outputs. Meanwhile, Spacebel's IPT Server demonstrated verification and authenticity of Earth Observation (EO) data sources, ensuring the integrity of information exchanged within the EO data supply chain. Together, these IPT Server instances provide examples of scalable frameworks for applying IPT principles, supporting the development of resilient data services that uphold reliability and trust across distributed networks.

To validate cross-component compatibility and demonstrate seamless data exchange, nine Technical Interoperability Experiments (TIEs) were conducted between these IPT servers and various tools, including a PDF Generator from SatCen and tools like UniResolver and UniVerifier. These experiments successfully established interoperability between Smart Certificates and Verifiable Credentials, confirming the flexibility and scalability of the IPT systems. Additionally, the TIEs identified challenges such as the need for JSON-LD 1.1 compatibility and suggested alternative encoding strategies, enhancing the potential for IPT systems to support diverse geospatial data services in real-world scenarios.

IV FUTURE OUTLOOK

To address the growing needs for data integrity, provenance, and trust across diverse domains, IPT frameworks combined with agile reference architectures that align with FAIR principles are essential. As demonstrated in the OGC Testbed 20 IPT activity, these frameworks are well-positioned to expand into areas beyond geospatial data, such as public health, where continuous monitoring, data integration, and interoperability are crucial. By leveraging these capabilities, IPT can support the management and use of data across dynamic and complex networks, ensuring that data remains reliable and trustworthy, regardless of fluctuating conditions.

Looking forward, it is recommended that IPT activities be incorporated into 2025 and 2026 Testbed and related activities. This includes a potential code sprint in Q1 2025 aimed at developing additional IPT Building Blocks that remain agnostic to specific encoding formats. Additionally, applying IPT in public health, as seen in the Zoonotic Spillover and Urban Extreme Heat Health Risk Index use cases, underscores the importance of IPT in supporting reliable and resilient data services across domains where data accuracy and provenance are essential.

In addition to integrity and trust, which were key focuses of the Testbed 20 activity, a comprehensive IPT framework must also encompass provenance. Provenance is crucial for tracking the "journey" of data as it undergoes various transformations and transits through networks or local processing environments. Particularly in systems governed by location-based regulations and security standards. Establishing a standardized IPT-geospatial

framework, in collaboration with other standardization bodies like the Defense GEospatial Information Working Group (DGIWG) and World Wide Web Consortium (W3C), could ensure interoperability across different levels of compliance (e.g., authenticity, licensing, and security environment). This approach would enable IPT principles to be effectively applied in geospatial IT and beyond, supporting emerging use cases in AI, emergency response, and other critical areas.

V VALUE PROPOSITION

Integrity, Provenance, and Trust are essential elements of a resilient data service, offering significant value for both clients and services by ensuring data reliability, traceability, and trustworthiness. Provenance captures the data's lineage, recording every change and who made those changes, which verifies the accuracy, completeness, and consistency of the data over time and across various formats. Trust is established through the use of authoritative, analysis-ready data and maintained across the data lifecycle when combined with the Integrity and Provenance components. Together, these elements create a trustworthy data service, underpinned by the frameworks and principles demonstrated in the OGC Testbed 20 activity, which incorporate IPT and FAIR principles as foundational building blocks.

This value was exemplified in the Earth Observation (EO) and Situation Report use cases, highlighting IPT's applicability beyond traditional data management. The EO Data Supply Chain use case illustrates how IPT helps data consumers verify the integrity of data from original sources, addressing both open and proprietary data requirements. Similarly, the EO Traceability Service use case demonstrates how traceability information, preserved through Verifiable Credentials (VCs), supports the verification of data sources and the overall chain of derived products. In the Situation Report use case, IPT principles ensure the reliability of merged geospatial information, from vector and raster data to AI-enhanced imagery, providing a comprehensive, validated data output ready for critical applications.

These use cases, together with the two robust IPT Servers and the insights from the Technical Interoperability Experiments (TIEs), highlight IPT's role and capabilities within resilient, secure, and trustworthy data services adaptable to diverse operational demands.

1 INTRODUCTION

1 INTRODUCTION

The incorporation and prioritization of the principles of Integrity, Provenance, and Trust (IPT) are critical for maintaining the availability, reliability, and credibility of data across various endpoints and operators, especially within distributed systems where access points can fluctuate and data may not always be readily available.

Integrity refers to the accuracy, completeness, and quality of data as it is preserved over time and across different formats. Maintaining integrity involves ongoing efforts to ensure that data remains unaltered by unauthorized users or accidental errors, maintaining its intended state for reliable use. In cybersecurity, integrity ensures that data is trustworthy and has not been modified, which is essential for any data-driven operation. Integrity, therefore, is not just about correctness: Integrity is about ensuring data is consistently fit for its intended purpose and remains so throughout its interactions with various systems.

Provenance is the documentation of the origin and history of data, akin to a chain of custody in legal contexts. Provenance data records the generation, transmission, and storage of information, allowing users to trace the source and journey of data throughout its lifecycle. Provenance is crucial for verifying the credibility, currency, and value of data, enabling users to detect any inconsistencies or "echoes" in the data. By ensuring that the origin and modifications of data are well-documented, provenance supports claims of data accuracy and enhances trust in data outputs.

Trust is the confidence that one system or entity places in another to perform its functions correctly, fairly, and impartially. Trust is built on the assurance that the data and entities involved are genuine and behave as expected. In resilient data services, trust is established by using authoritative and trusted analysis-ready data (ARD), and maintained by integrating the principles of integrity and provenance. Together, IPT ensures that the data and its sources are reliable, which is essential for making informed decisions based on that data.

The Integrity, Provenance, and Trust components are interconnected and mutually reinforcing within a resilient data service. Provenance ensures integrity by documenting all changes made to the data and identifying who made those changes. Integrity guarantees that the data is accurate, complete, and consistent over time. Trust is established through the use of reliable, authoritative data and is maintained by combining the principles of integrity and provenance. The architectures explored in the OGC Testbed 20 IPT activity demonstrate how Resilient Distributed Data Services can serve as building blocks for incorporating IPT principles alongside FAIR principles, ultimately fostering trust in the data.

1.1. Aims

The Testbed 20 IPT activity explored architectures that support Resilient Data Services, focusing on the delivery of data and services that are resistant to climate effects. The key objectives were

to address the challenges of resilience, integration, and interoperability in data services, ensuring that they remain robust even under adverse environmental conditions.

In considering Resilient Data Services, which are tolerant of climate impacts, the following aspects are essential.

- Resilience, Integration, and Interoperability: Develop strategies to overcome the challenges associated with building resilient data services that can seamlessly integrate and interoperate across diverse systems.
- Universal Access: Ensure that data and services are universally accessible for discovery and assurance, allowing users to easily find and trust the information they need.
- Architectural Options: Explore various architectural options for the distribution, discovery, and access to contextual information, supporting heterogeneous information sources. This includes identifying the best approaches for managing and delivering data from multiple sources in a coherent and reliable manner.
- Continuous Integration and Continuous Testing: Enable a Continuous Integration and Continuous Testing approach to maintain the quality and reliability of data services as they evolve, ensuring that any changes or updates do not compromise the service's resilience.
- Network Characteristics and Intelligent Monitoring: Incorporate intelligent monitoring of network characteristics to ensure Quality of Service. This involves using advanced techniques to monitor and manage network performance, guaranteeing that data services meet required standards even under varying conditions.

The architectures explored in Testbed 20 serve as building blocks for incorporating IPT and FAIR principles. Building blocks may include complete OGC API standards, parts of multi-part standards, or more granular functionalities such as data types or parameters. Each building block is designed to be an open and reusable digital solution, whether as a framework, standard, software, or SaaS. These components are defined to meet specific business needs and ensure that the architecture supports the resilient and reliable delivery of data and services.

1.2. Objectives

The primary objectives of the OGC Testbed 20 IPT activity were to design and deploy two IPT Servers, leveraging components from the OGC standards baseline, and previous OGC Testbed developments. In addition to, conducting interoperability testing to ensure seamless communication, integration, and functionality between IPT systems. Whilst also defining initial use cases for utilizing IPT Servers and highlighting security, privacy, and ethical considerations.

2

TOPICS

2 TOPICS

2.1. Secure Dimensions — D100 IPT Server Instance

2.1.1. Objectives

Secure Dimensions' primary objective is to provide a Federated Agile Collaborative Trusted System (FACTS) ecosystem that supports the agile management of Smart Certificates to establish Integrity and Trust in geospatial data products. FACTS operates through Collaborative Objects, self-contained units comprising data products, services, and processes that actively participate in event exchanges and operations, emphasizing data-centric security that focuses on data reliability rather than traditional network or server security.

A FACTS trusted process is a piece of software where the code was inspected, deployed through a trusted operator. The process can be verified to be certified, and to operate on verifiable inputs only by requiring the user to provide a Smart Certificate when executing the process, produces a Smart Certificate for the output product, and records the provenance metadata. A Smart Certificate ensures that the FACTS Collaborative Object is doing what it is supposed to do, supporting verified attestation and processes, such as assuring that the APIs on the Collaborative Object interact with FACTS. A Smart Certificate, acting as a profile of Verifiable Attestation, assures that certification is in place before a data product is released, as defined by the EBSI guidelines for attestation types.

The second objective is to deploy two processes using an implementation of the OGC API — Processes Standard that leverage the FACTS ecosystem to validate Smart Certificates and W3C Verifiable Credentials (VC) for the process input.

The first process is called `Trusted Teapot`, which validates a FACTS Smart Certificate for the input image and produces a Smart Certificate for the created output image. The Smart Certificates are stored in the process caller — a FACTS holder like Alice.

The second process is called `Trusted Watermarking`, which accepts a W3C VC as input. After the successful validation of the VC, the process executes.

A detailed overview of the FACTS Ecosystem and the Trusted Teapot Process as well as the Trusted Watermarking Process is available in Annex B.

2.1.2. FACTS Ecosystem

The idea of a FACTS originates from the previous OGC 2023 Testbed 19. Testbed 19 identifies the need to ensure reliability for geospatial data processing; expressed as IPT. The implementation of Integrity, Provenance and Trust is, along with authenticity, the basis for

establishing reliability between geospatial data producers and consumers. In previous OGC Testbeds, solutions to IPT were implemented using the Data Centric Security (DCS) approach where the security measures are built into the data. The solutions are an excellent choice but require the data format to accommodate security descriptions inside the actual geospatial data. For some use cases it is almost impossible to switch to another data format. Especially when the existing data is already in a standardized format or where data is going to be produced with existing, expensive and operational systems. To enable IPT with existing data instances and without requiring the change of legacy systems, the idea of FACTS as a standalone eco-system was born. FACTS API endpoints can be implemented that support manual, semi-automatic or fully integrated processing of geospatial data that already exist or is produced with existing systems. The key point is the FACTS Smart Certification API implementation which persistently links a geospatial data instance of any format with one or multiple revocable Smart Certificate(s). For ensuring trusted cross-organizational verification, the attested attributes inside a certificate — the certificate structure — is stored on the FACTS distributed ledgers. Also, the public DIDs of certificate issuers are stored on the distributed ledgers.

The issuing of Smart Certificates can be implemented in different ways: The automatic issuing of a Smart Certificate when data is produced requires that the processing service integrates the FACTS API endpoint. This requires — obviously — the integration of the FACTS API for certificate validation and issuing. The automatic issuing of a Smart Certificate for existing data instances is possible if the producer operates a 'provenance' catalog in which all data instances' metadata are recorded that ever got produced. Such a catalog can be operated by each data producer. Alternatively, the FACTS immutable catalog can be used. The FACTS immutable catalogue allows to link a producers metadata record — required to issue a certificate — with a persistent identifier on the FACTS distributed ledgers. Once a data instance is registered in the immutable catalog it becomes a genuine record that allows IPT enablement for the associated data instance.

From a service or user perspective, the FACTS API endpoint distinguishes three different roles: The `issuer` role allows to issue and revoke Smart Certificates; the `holder` role is occupied by the acting user or service that requests certificates of participates in their proof; the `verifier` role is mainly used by IPT enabled services to validate Smart Certificates. This separation of concerns can be found in many Self-Sovereign-Identity (SSI) use cases: The holder decides when to request a certificate and when to use a certificate in a proof initiated by verifiers. The important aspects of privacy and need-to-know are supported by peer-to-peer communication (between issuer-holder and holder-verifier) and the ability of Zero-Knowledge-Proof (ZKP). The latter enables that the holder may withhold certificate attributes inappropriate for a particular proof request. For example, the proof of whether or not a data imagery covers a given area of interest does not require disclosing the entire graphical extend of the data product. It is sufficient that the holder answers spatial predicate proof requests like "does the data intersect with area (0,0,100,100)?"

FACTS could be operated by different organizations that want to establish trust among each other without relying on one single 'super' trusted central organization. In that sense, all operators would participate in the FACTS decentralized distributed ledgers. Operated as a permissioned ledger, it provides 'public' read access to all entities that are configured to have access to the ledgers and controlled write access to the authorized organizations (typically those) that operate the distributed ledgers. The scalability of the FACTS eco-system is natively supported as the distributed ledgers are only storing the issuers public DIDs and the certificate structures. Any other operational interactions take place in the business logic that could be

hosted in any cloud-based environment. Holders store their Smart Certificates in their own secure wallet and decide when to use or delete them. This data economy ensures that Smart Certificates remain manageable in the FACTS eco-system.

2.1.3. The Trusted Teapot Process

The `Trusted Teapot Process` processes an input image if the caller can present a FACTS Smart Certificate for that input image. A FACTS Smart Certificate for this process contains information about the image itself but also includes business logic directives which control the usage of the associated image.

A user that wants to execute this process must be a FACTS user with the role holder. The user must also have stored a Smart Certificate in their wallet which gets validated by the `Trusted Teapot Process.`

The process begins with a developer obtaining FACTS compliance certification for their software, ensuring that it can only process data with valid certificates, creates certificates for output, and records execution metadata in the Immutable Catalog. After certification, the process is deployed in an OGC API Processes framework, where it becomes available for use.

For each execution, the Trusted Teapot verifies the user's Smart Certificate for the input image, processes the image, and generates an output certificate. All associated metadata is logged in the Immutable Catalog, establishing a traceable record of the transaction. The system ensures that only authorized images are processed and that data integrity, provenance, and trust are maintained throughout the operation.

2.1.4. Trusted Teapot Process Execution Example

The Trusted Teapot Process can be executed utilizing the following steps.

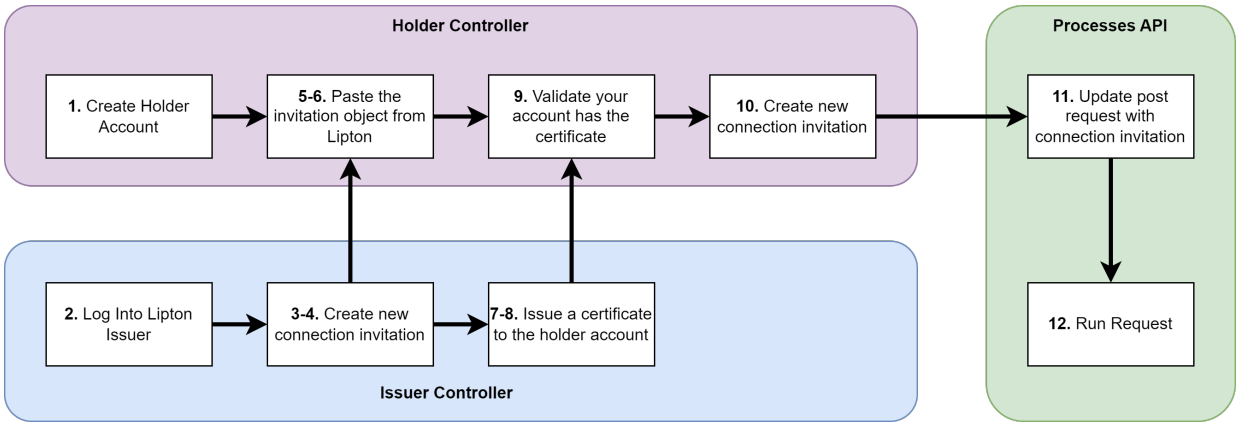


Figure 1 — Trusted Teapot Execution Process Flow

1. Register an account as a holder here: https://holders.ogc.secd.eu/Account/Login
2. Log in to the issuers controller and log in with the Lipton username with the password ThisIsSecure!

- In this case Lipton will be the one who grants the certificate to your holder account that allows for verification

3. In the issuers controller as the Lipton user navigate to connections: https://issuers.ogc.secd.eu/connections/active
4. Create a new connection and copy the invitation object
5. In the holder controller where you are logged into your own account navigate to connections: https://holders.ogc.secd.eu/connections/accept
6. Paste in the invitation object from the Lipton user from the issuers controller
7. Navigate to the certificates in the issuers controller: https://issuers.ogc.secd.eu/credentials/issue
8. Issue a certificate using the connection to your holder account and selecting the teabag2.0 credential definition
    - For the credential attributes set the value of URL to: https://iafs.demo.secure-dimensions.de/f/998845b59db786d3df28ba85b42c6a481da2b3843399ae5c3f49a06c1e564f87
    - Set the value of the hash to: sha256=998845b59db786d3df28ba85b42c6a481da2b3843399ae5c3f49a06c1e564f8
    - Send the certificate
9. In the holders controller navigate to certificates and verify that your account now has a certificate from Lipton for teabag2.0: https://holders.ogc.secd.eu/credentials/list
10. In the holders controller navigate to https://holders.ogc.secd.eu/connections/new Create a new connection invitation and copy the invitation object
11. Navigate to Trusted Tea Pot Processes API: https://processes.ogc.secd.eu/openapi?f=html#/trusted-teapot/executeTrusted-teapotJob and scroll down to the post request in processes
    - Replace the connection invitation with the values copied from step 10
    - Replace the teabag URL with: https://iafs.demo.secure-dimensions.de/f/998845b59db786d3df28ba85b42c6a481da2b3843399ae5c3f49a06c1e564f87
12. Run the POST request

2.1.5. The Trusted Watermarking Process

The objective of the `Trusted Watermarking Process` is to demonstrate the ability of the FACTS eco-system to verify W3C VCs that were created by Spacebel.

For demonstrating the use of W3C VCs with this process, Spacebel generated VCs and stores them in their catalog, accessible under the following OGC API Records endpoint: https://emc.spacebel.be/collections/Aqua_AMSR-E_L3_SSW_1month_0.25deg/items. Any record that includes a "QUICKLOOK" look can be used as input to the `Trusted Watermarking Process`.

A successful execution of the process generates an image with watermark "FACTS <date>" in the upper right corner. The watermarked image is uploaded to the Secure Dimensions Integrity Assured File System: https://iafs.demo.secure-dimensions.de/. The process response in JSON format contains the URL to the watermarked image and the RFC 9530: https://www.rfc-editor.org/rfc/rfc9530 compatible hash. RFC 9530 defines HTTP fields that support integrity digests. The Content-Digest field can be used for the integrity of HTTP message content.

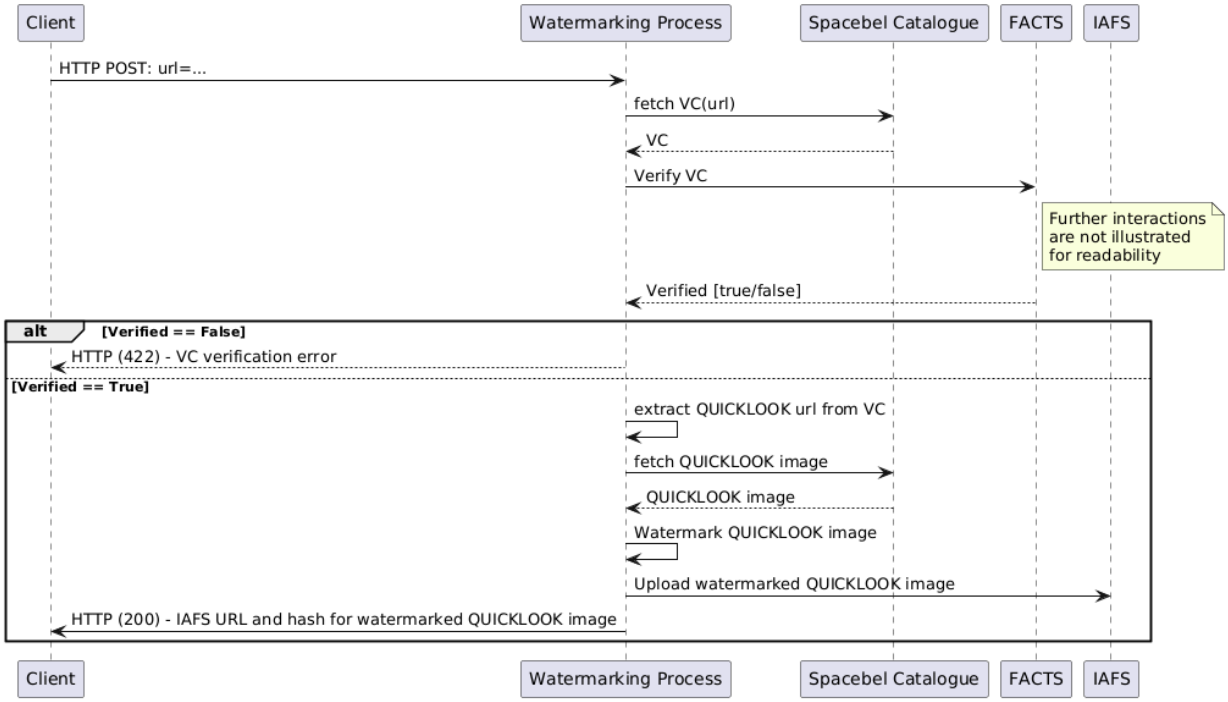


Figure 2 — Interaction Diagram

The `Trusted Watermarking Process` API is described via OpenAPI: https://processes.ogc.secd.eu/openapi?f=html#/trusted-watermarking

2.2. Spacebel — D100 IPT Server Instance

2.2.1. Objective

The objective of the Spacebel IPT Server scenario setup is to support downstream Earth Observation data consumers to verify that data consumed is from the original data provider and thus allow verification of the integrity of the data.

The scenario addresses `integrity`, `provenance`, and `trust` (IPT) defined below.

> "Provenance is information about entities, activities, and people involved in producing a piece of data or thing, which can be used to form assessments about its quality, reliability or trustworthiness". [W3C PROV].
>
> "Data integrity is the opposite of data corruption. The overall intent of any data integrity technique is the same: ensure data is recorded exactly as intended" [Wikipedia].
>
> "Trust is the characteristic that one entity is willing to rely upon a second entity to execute a set of actions and/or to make set of assertions about a set of subjects and/or scopes" [OASIS].

In this scenario, Decentralized Identifiers (DIDs) are used for identifying organizations and EO resources (products). Verifiable Credentials (VC) and Presentations (VP) are exchanged between (VC) issuers, holders and verifiers as depicted below.

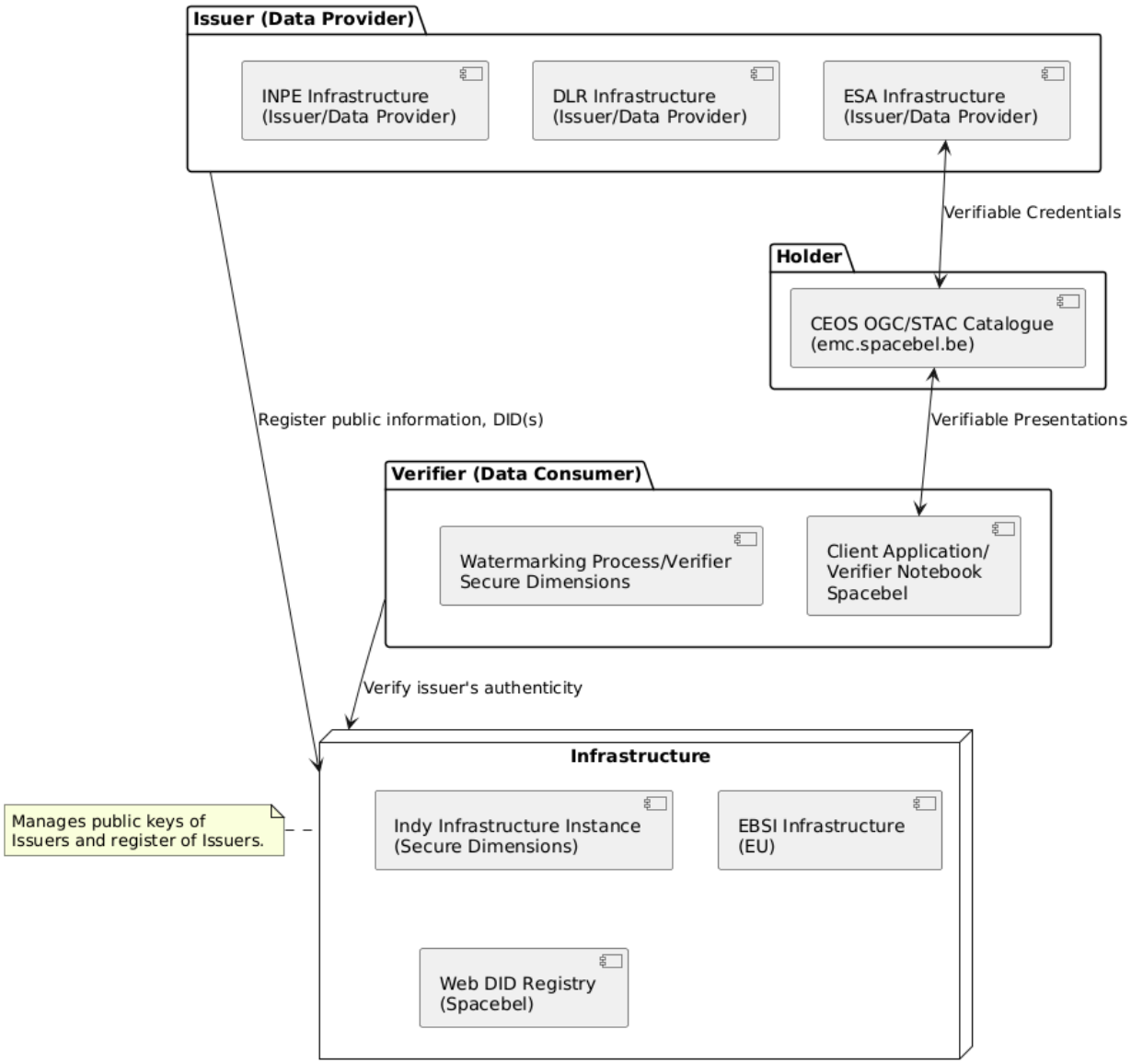


Figure 3 — Deployment Diagram

This section provides an overview of the architecture used in the IPT activity. The details on the interactions in the scenario are included in the Jupyter Notebook implementation which is included in Annex D.

2.2.2. Component Overview

The scenario relies on the following architecture. An overview of the components of the architecture are described below and examples of the components are shown in Annex C.

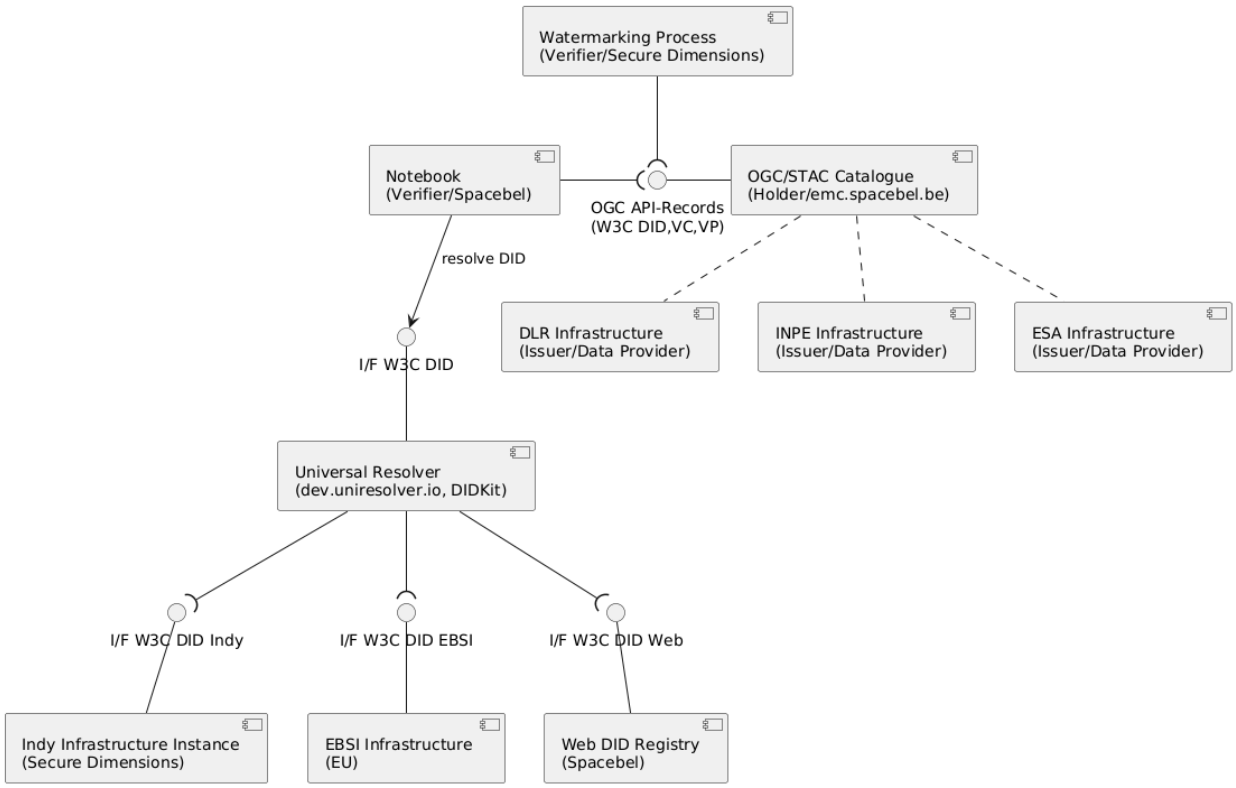


Figure 4 — Component Diagram

2.2.2.1. DID Resolver

According to DID-CORE, a DID resolver is a software and/or hardware component that performs the DID resolution function by taking a DID as input and producing a conforming DID document as output.

In the demonstrator, Spacebel Testbed 20 participants used https://dev.uniresolver.io, https://godiddy.com and DIDKit as Universal Resolvers. This allows for the generation of abstractions of the underlying infrastructure which may be the European Blockchain Service Infrastructure EBSI, an Indy instance, or an infrastructure serving Web DIDs.

One of the design goals of DIDs was indeed to be system- and network-independent and enable entities to use their digital identifiers with any system that supports DIDs and DID methods.

In the scope of the IPT D100 deliverable, the expectation is that involved parties (DID owner, VC holder, signer, verifier) are able to register DIDs and resolve them into DID Documents, according to the W3C DID core recommendation. With the specification of the DID method at stake. In practice, this can be done with a variety of existing tools and libraries, depending on the DID method.

The potential standardization proposals from this deliverable do not make assumptions or mandates on the practical implementations used to register or resolve a DID, or produce a Verifiable Credential or Presentation. Furthermore, it must be noted that at the date of

publication of this Report, none of the available DID resolvers offer the level of universality and generality, across languages and DID methods, expected from a production-grade solution.

2.2.2.2. Spacebel OGC/STAC Catalog

Spacebel's STAC Catalog used in the scenario serves metadata for EO collections and granules (products) via its OGC API-Records or STAC API endpoints. The main resources of the API implementation instances are provided below. Note that the resource `DidDocument` at the path `/did.json` represents a DID Document for the corresponding resource. The Catalog acts as `Web DID Registry` for the DID documents of its resources. A (Web) DID URL such as `did:web:emc.spacebel.be:collections:{collectionId}:items:{featureId}` resolves to the DID document at `https://emc.spacebel.be/collections/{collectionId}/items/{featureId}/did.json`. The Landing Page of the Catalog is accessible at https://emc.spacebel.be. It is based on a development version of the CEOS FedEO Catalog.

The Catalog makes available the granules in Verifiable Credential and Verifiable Presentation formats. The same media types as EBSI are used for this:

- `application/vc+ld+json` : metadata signed by the original issuer (data provider);
- `application/vp+ld+json` : metadata signed by the original issuer (provider) and the holder.

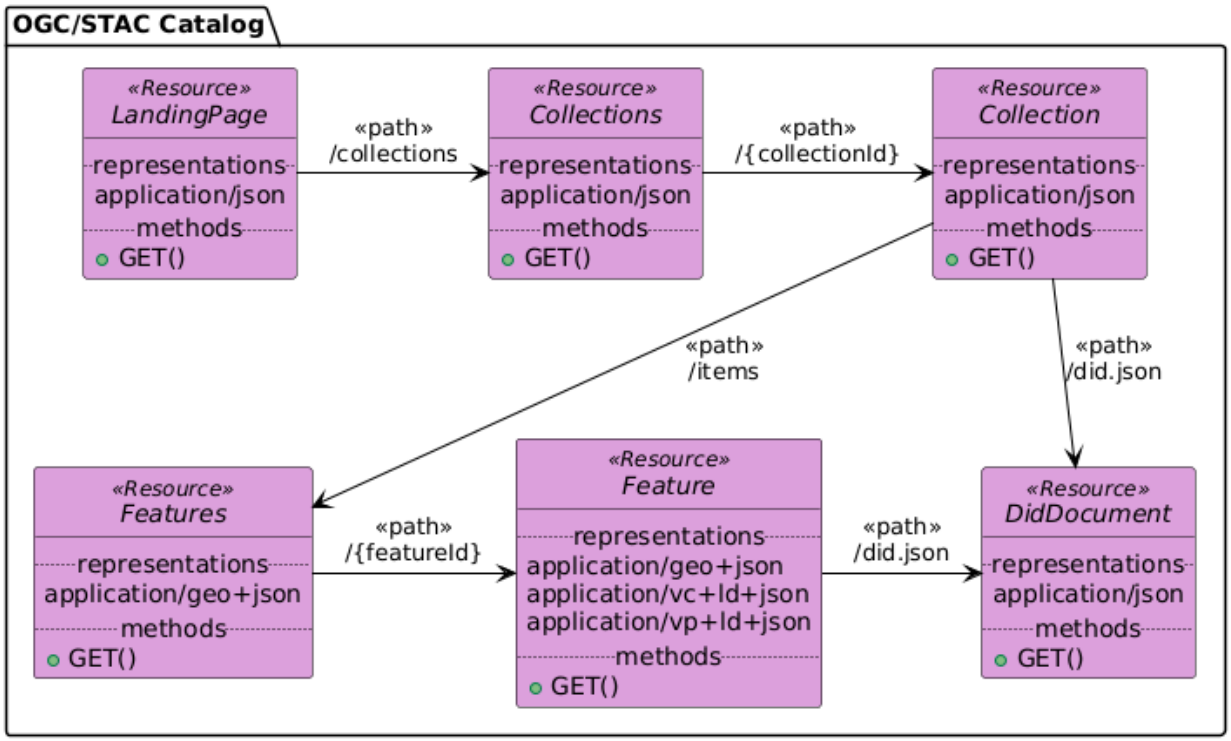


Figure 5 — OGC/STAC Catalog Resource Diagram

2.2.2.3. Data Consumer / Notebook

The Jupyter Notebook (See Annex D) illustrates several operations of an Issuer, Holder and Verifier. As an Issuer, it simulates a Data Provider (e.g., ESA, DLR, INPE, ...) and produces Verifiable Credentials for the (CEOS) Catalog. As a Holder, it produces Verifiable Presentations for data consumers based on Verifiable Credentials issued by Data Providers. As a Verifier, it obtains Verifiable Presentations from the (CEOS) Catalog.

2.2.2.4. W3C Decentralized Identifiers

The use of Private DID versus Public DID are discussed here:

- Private DIDs can be exchanged between two parties to create a secure channel that no unauthorized person or party can access in that channel.
- Public DIDs are beneficial for parties that need to be publicly identifiable such as a government department that issues IDs for citizens, players in the supply chain (producers, manufacturers, distributors, etc.), or health care providers.

In the current demonstrator, public W3C Decentralized Identifiers (DIDs) are used to identify the following entities in an Earth Observation-related supply chain:

- Organizations (i.e. Legal Persons) acting as data providers or data consumers (issuers, holders, verifiers)
- Earth Observation (EO) products and collections.

DID assigned to a product can be found in the metadata for the product available in the Catalog. The DID document for a product identifies the controlling organization (issuer) of the DID via an organization DID. A Verifiable Credential (VC) related to a product DID by an issuing organization can be cryptographically verified with the public key of the organization issuing the VC. This key is available in the DID document for the organization DID.

2.2.2.5. Issuers

In a real-life operational scenario, a distributed trust model with a trust chain of trusted issuers (TI) may be setup on the European Blockchain Service Infrastructure or EBSI. A Trusted Accreditation Organization (TAO) would then accredit the issuers. TAO are organizations responsible for accrediting trusted issuers in a specific sector or domain in a specific geography and to issue certain types of Verifiable Credentials (VC).

For the current use case, CEOS and/or ESA might act as TAO for accrediting trusted issuers in the Earth Observation domain either globally (CEOS) or in Europe (ESA).

In comparison with the EBSI multi-level Issuers Trust Model, the Spacebel IPT Server scenario utilized limited issuers that were at Level-3 and ignored the accreditation steps and verifiable accreditation required in a real-life scenario.

The following issuers have been defined via W3C DIDs. For simplicity, Web DID have been used, but Indy DID or EBSI DID could be used as well.

- did:web:emc.spacebel.be:organisations:ceos
- did:web:emc.spacebel.be:organisations:esa_esrin
- did:web:emc.spacebel.be:organisations:br_inpe
- did:web:emc.spacebel.be:organisations:de_dlr
- did:web:emc.spacebel.be:organisations:spacebel_sa
- did:web:emc.spacebel.be:organisations:jp_jaxa_saoc

DIDs resolve to DID Documents that contain information about the public keys that is used by consumers of the VC and VP to verify signatures. Three types of key / key representations were used in the demonstrator:

Table 1

DID	PUBLIC KEY REPRESENTATION	COMPATIBLE VERIFIERS
DLR, CEOS	publicKeyJwk (“kty”: “OKP”, “crv”: “Ed25519”)	DIDKit, UniVerifier
ESA/ESRIN, INPE, Spacebel	publicKeyJwk (“kty”: “EC”, “crv”: “secp256k1”)	DIDKit, UniVerifier
JAXA	publicKeyBase58 (“kty”: “OKP”, “crv”: “Ed25519”)	hyperledger/aries-cloudagent-python, Uni Verifier

2.2.2.6. EO Resources

EO products in the Catalog have been assigned a (Web) DID. Below a number of examples.

- did:web:emc.spacebel.be:collections:TropForest:items:KO2_OTPF_KO2_MSC_2F_20091107T04175
- did:web:emc.spacebel.be:collections:AMZ1-WFI-L4-SR-1:items:AMAZONIA_1_WFI_20240618_037_016
- did:web:emc.spacebel.be:collections:F-FSM:items:F_FSM_20160610T000000_20161101T235959_ROCHEFORT

2.2.2.7. W3C Verifiable Credentials

The prototype exchanges claims about EO products (i.e., the `credentialSubject`) as Verifiable Credentials (v1.1). As the Verifiable Credentials Data Model is based on JSON-LD, the OGC EO Dataset Metadata GeoJSON(-LD) Encoding Standard (OGC 17-003r2) encoding, which defines a proper @context document, was used. Alternative EO product metadata encodings such as GeoDCAT-AP or schema.org are also suitable.

```
"type": [
   "VerifiableCredential", "Feature"
],
"@context": [
   "https://www.w3.org/2018/credentials/v1",
   "http://schemas.opengis.net/eo-geojson/1.0/eo-geojson.jsonld"
]
```

Listing 1

In a future version, the STAC JSON-LD encoding might be used as well to encode claims in Verifiable Credentials (v2.0). This encoding is under development as a set of building blocks. See https://ogcincubator.github.io/geodcat-ogcapi-records/build/annotated/geo/geodcat/geodcat-stac-eo/context.jsonld.

```
"type": [
   "VerifiableCredential", "Feature"
],
"@context": [
   "https://www.w3.org/ns/credentials/v2",
   "https://ogcincubator.github.io/geodcat-ogcapi-records/build/annotated/geo/
geodcat/geodcat-stac-eo/context.jsonld"
]
```

Listing 2

The open-source `DIDKit` library cannot resolve external context documents at run-time, but including context files at build-time is recommended. As work-around, a fragment of the context document has been included in-line.

The following is an example Verifiable Credential generated by the Catalog. In this example, some links to content (e.g. download, thumbnail or quicklook links) are content-integrity protected. This can be achieved via URL schemes that enforce content integrity such as Cryptographic Hyperlinks or the Interplanetary File System (IPFS). Alternatively, the relatedResource property with one or more cryptographic digests for each related resource can be used.

The computation of the cryptographic hyperlinks in the example below involved the following steps.

- Generate the raw hash value by processing the resource data using the cryptographic hashing algorithm.
- Generate the multihash value by encoding the raw hash using the Multihash Data Format (sha2-256).

- Generate the multibase hash by encoding the multihash value using the Multibase Data Format (base58btc).
- Output the multibase hash as the resource hash (`?hl=xxxx`).

NOTE:Multihash is a protocol for differentiating outputs from various well-established cryptographic hash functions, addressing size and encoding considerations.

The result of a VC verification by DIDKit corresponds to the structure proposed by the W3C VC-API.

2.2.2.8. W3C Verifiable Presentations

A Verifiable Presentation related to (claims from) one or more Verifiable Credentials. The relation is depicted below.

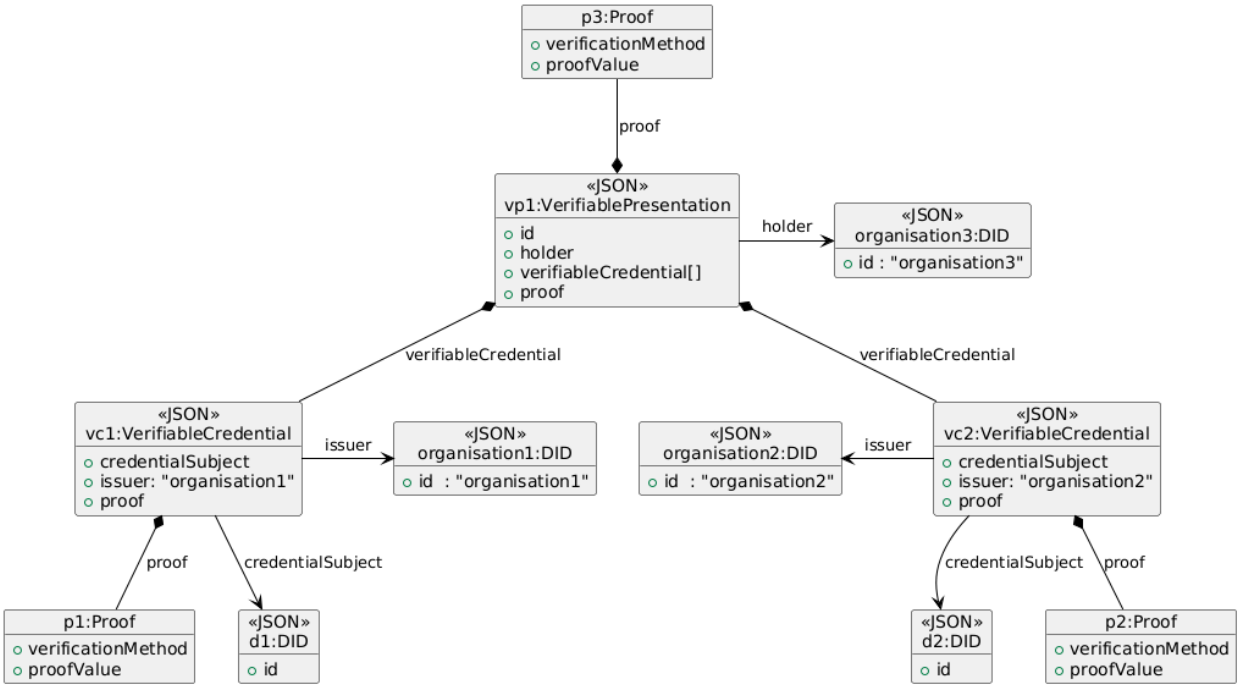


Figure 6 — Structure of a Verifiable Presentation

To verify a VP, the Verifier obtains the public keys of holder and issuer(s) from the corresponding DID documents as depicted below.

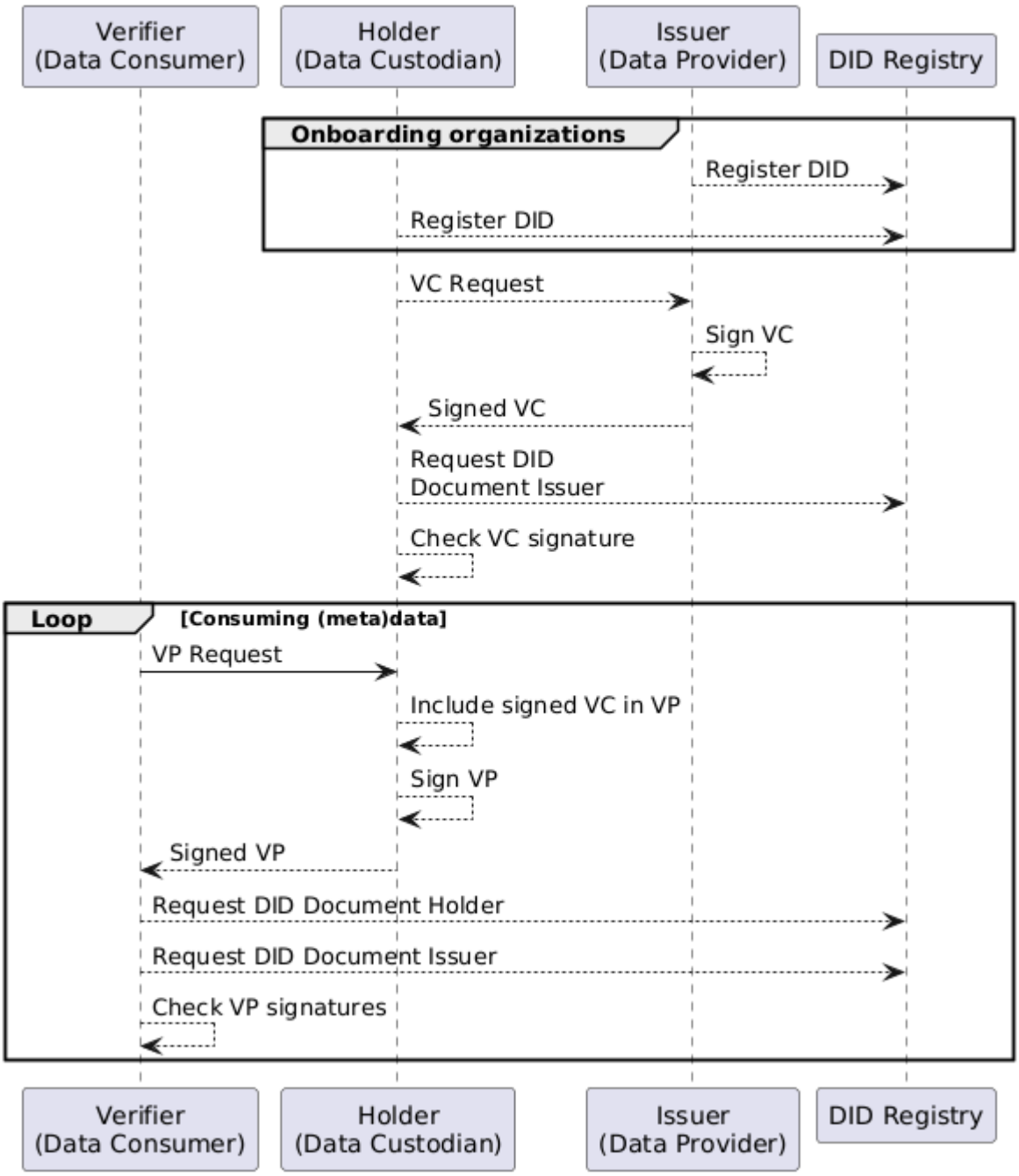


Figure 7 — Sequence diagram VC/VP verifications

The result of a VP verification by DIDKit corresponds to the structure proposed by the W3C VC-API. The VC and VP available in the Catalog were verified with the Univerifier Tool as depicted in Annex C.

2.3. UseCases

2.3.1. EO Use Cases

2.3.1.1. EO Data Supply Chain Use Case (Spacebel)

An important element to be considered is the heterogenous source of data and their destination. It is important to distinguish the applicable requirements for:

- Open data/services
- Proprietary data/services

In general, a double approach considering open and proprietary data seems highly required.

The use case proposed is related to the scenario from the Secure Catalogue activity in Testbed-18. While the Secure Catalogue covered both `Integrity` and `Confidentiality`, the EO Supply Chain use case in the current Testbed is demonstrated with `Open data` and concentrates on `Integrity`, `Provenance`, and `Trust`.

The objective is to allow downstream Earth Observation data consumers to "trust" the data, i.e., allow them to verify that data consumed is from the original data provider and allow verification of the integrity of the data.

2.3.1.1.1. Scenario

- EO Data Provider (VC Issuer)
- EO Catalogue Owner (VC Holder)
- EO Data Consumer (VC Verifier), e.g., EO Data Processor (Third Party / Value adder)

DID identifiers are used for:

- Organizations (data provider, catalogue owner, Data Processing Company)
- EO data products (i.e., metadata record identifiers)

Approach: data consumers can verify (via W3C Verifiable Credentials and Verifiable Presentations) that data consumed is from original data provider (integrity). Content/resource integrity protection to protect integrity of external link targets (external to the Verifiable Credential) is used as well. In addition, all parties in the supply chain identify themselves with W3C Decentralized Identifiers (DID).

For more details, please refer to Annex C.

2.3.1.2. EO Traceability Service Use Case (Spacebel)

This use case focuses on providing information similar to that available from the Copernicus CDSE Traceability Service using DID and VC. See https://documentation.dataspace.copernicus.eu/APIs/Traceability.html and https://github.com/eu-cdse/trace-cli#digital-signatures for more information and the example at https://trace.dataspace.copernicus.eu/api/v1/traces/name/S2A_MSIL1C_20230420T100021_N0509_R122_T33UVP_20230420T120027.SAFE.zip.

The traceability information can be provided via W3C Verifiable Credentials/Verifiable Presentations allowing verification of the integrity of the individual data sources and the integrity of the overall traceability information as well.

While the previous use case uses VC/VP for a single EO product, the traceability use case combines claims about the final product and the intermediate products it was derived from, in a single W3C Verifiable Credential/Verifiable Presentation. In addition, the intermediate products the product traces to, can be protected by content/resource integrity protection, e.g., multibase multihash links, as in the previous use case.

2.3.2. Situation Report Use Case (EU SatCen)

2.3.2.1. Combination of Geospatial Information to support emergencies

In this use case, the focus is on merging into one single file valid information (with DID & smart certificate) coming from:

- Vector datastores (e.g., open street map);
- Raster datastores (e.g., Sentinel2 data, Landsat data);
- Calculating network topology for Open Street Map (OSM) (e.g., OGC-API Routes (as an example of an OGC API ITP enabled and/or osm2pgrouting validate with smart contract)); and
- Applying a super resolution algorithm (validated SContract) for Sentinel 2. Super resolution imaging (SR) is a class of techniques that enhance (increase) the resolution of an imaging system.

Objective

A key focus is reliability of information produced or process results. The latter being strongly dependent from inputs where provenance should be carefully demonstrated and evaluated. Validated results are then fused in a validated map, such as GeoPDF, ready for distribution. The final result should be a valid and ITP enabled PDF derived from all the above. Demonstrating trustability of the data.

Workflow

- Search a catalog for a specific Area of Interest and sensors (assuming ESA Sentinel 2).
- Supersample with Artificial Intelligence algorithms (parameters: algorithm, band combination, scale factor, resulting encoding).

Figure 8 — Sentinel 2 Catalgue query and AI superresolution

- In the case where the specific algorithm is not considering geography, a reprojection is required.
- Query Open Street Map database with image bounding box.
- Merging imagery and vectors in a single file (jp2, GIMI,..) converting to PDF.
- Calling a python script to make the PDF FACTS compliant.

2.3.3. Future Health Use Cases

In health-related scenarios, Integrity, Provenance, and Trust are foundational principles, ensuring that the data used is accurate, traceable, and reliable. These principles are vital in scenarios where data authenticity and secure interactions between systems are paramount. Below, potential health use cases are presented (Zoonotic Spillover and Urban Extreme Heat Health Risk Index) where IPT can ensure the protection, integrity, and trustworthiness of sensitive data at each stage of the process. Each use case ties into the larger framework of Resilient Distributed Data Services, emphasizing the need for reliable and secure data management under real-world conditions. These proposed use cases are intended for consideration in future OGC testbed or pilot activities, as testing them falls outside the scope of the Testbed 20 efforts.

2.3.3.1. Zoonotic Spillover (HSR.health)

The goal of this use case is: The identification of regions at high risk for zoonotic spillover events, which can lead to outbreaks of emerging or endemic diseases.

Data Required:

- Earth Observation (EO) Analysis: Land Use and Land Cover Classification
- Vector Data: Administrative Boundaries, Social, Demographic, Environmental Data, Climate Change Data, Historical Zoonotic Spillover Events, and Chronic Health Conditions

End Users:

- Emergency Response
- Public Health
- Governments at local, regional, central levels
- Multinational authorities
- Environmentalists
- Ranchers, Farmers, Livestock Suppliers
- Loggers
- Businesses

Use Case:

- Identifying regions at risk for zoonotic spillover events, such as areas with high livestock density or significant environmental change.
- Recommending response efforts based on causative factors such as deforestation, livestock movement, and population density.

Recommendation:

IPT is crucial in ensuring that the data used to identify zoonotic spillover risks is authentic, unaltered, and can be traced back to its source. Integrity ensures the accuracy and completeness of environmental, health, and demographic data. Provenance provides a clear chain of custody, documenting where data originated and how it has been processed. Trust is established by ensuring that only authorized entities—such as government health agencies or environmental monitors—have access to or can modify the data, maintaining its reliability.

IPT ensures that data remains resilient even when accessed from different systems or under varying environmental conditions. The trust established between entities sharing zoonotic

data is further reinforced by Resilient Distributed Data Services, which guarantee that the data remains accessible and accurate even in disaster-prone areas.

This use case for integrating the IPT components demonstrated Testbed 20 into the Zoonotic Spillover Risk Index builds upon previous work from the Disaster Pilot 21. Peru was where the Pandemic Health Risk Index and Zoonotic Spillover Risk Index were piloted. The system is ready to be integrated into an IPT-enabled workflow, ensuring secure data provenance and trust throughout the decision-making process in a resilient data-sharing framework. In addition, the Zoonotic Spillover Risk Index has been piloted through Amazon Web Services (AWS) and Snowflake for Peru and Brazil, priming the Zoonotic Spillover Risk Index for piloting the inclusion into an IPT workflow.

2.3.3.2. Urban Extreme Heat Health Risk Index (HSR.health)

The goal of this use case is: To identify populations at high risk from extreme heat exposure and determine which urban areas are most susceptible to extreme heat.

Data Required:

Heat Data:

- Temperature data from sources such as the National Weather Service and Climate Central, including calculations of “felt heat” that incorporate ambient temperature, humidity, heat dome effects, and urban heat islands.

City Environment Data:

- Administrative boundaries, neighborhood data, and information on the urban landscape and built environment.

Population Characteristics:

- Demographic information, social determinants of health, and underlying health conditions.

End Users:

- Emergency Response Teams
- Public Health Authorities
- Urban Planners
- Real Estate Developers
- Economic Development Agencies
- Local Community Representatives

Use Case:

- Determining city-wide risk levels for extreme heat exposure.
- Identifying at-risk populations, such as the elderly, those with pre-existing health conditions, or communities without access to cooling centers.
- Sizing and locating cooling centers, integrating evacuation planning, and optimizing resource allocation, especially for vulnerable populations like the homeless.

Recommendation:

IPT plays a critical role in ensuring that the data driving heat exposure and population risk models is accurate and secure. Integrity guarantees that data collected from disparate sources — temperature readings, population health statistics, and urban landscape information — remains consistent and unaltered. Provenance allows urban planners and emergency response teams to trace the origins of the data, ensuring its credibility, especially when making life-saving decisions about resource allocation and emergency responses. Trust is built through authenticated, reliable data sharing, enabling effective cross-agency collaboration in managing extreme heat risks.

As highlighted in the Security, Privacy, and Ethical Considerations section, Smart Certificates and Zero Knowledge Proofs (ZKP) can be applied to protect sensitive demographic or health-related information. For example, public health officials may need to verify risk without exposing personal health details, using ZKP to demonstrate that certain conditions are met without revealing the underlying data.

This use case builds upon work performed in the 2024 Climate and Disaster Resilience Pilot for Northern Manhattan in New York City and is replicable in other cities. This previous work suggests that the Urban Heat Health Risk Index is ready for piloting its inclusion into an IPT workflow linking the outputs from the testbeds to ongoing pilot activities. The incorporation of IPT principles ensures that the index remains resilient, transparent, and secure as it scales to new environments and populations.

2.4. Technical Interoperability Experiments

As part of OGC Testbed 20, interoperability testing was conducted between various components to ensure seamless communication and functionality across different systems. Specifically, tests were carried out between the Secure Dimensions IPT Server instance and the Spacebel IPT Server instance. Additionally, the PDF Generator from SatCen was tested for interoperability with the Secure Dimensions IPT Server. The objective of these tests was to validate the technical integration, data flow, and cross-system compatibility among these components.

2.4.1. TIE Testing Matrix

The matrix below illustrates the Technical Interoperability Experiments (TIE) conducted during the testbed, including component interactions and the success of each interoperability scenario:

Table 2 — TIE Testing Matrix

FROM / TO	SERVER #2A SPACEBEL	CLIENT #2B NOTEBOOK	SERVER #1A FACTS
SERVER #2A SPACEBEL	X	.	.
CLIENT #2B NOTEBOOK	(3)	X	(4)
SERVER #1A FACTS	(6)	.	X
SERVER #1B TRUSTED TEAPOT	.	.	(2)
SERVER #1C TRUSTED WATERMARKING	(1)	.	(7)
CLIENT #3 PDF GENERATOR	.	.	(5)
CLIENT #4 UNIRESOLVER	(8)	.	(7)
CLIENT #5 UNIVERIFIER	(9)	.	.

2.4.2. Components

The TIE experiments involved the following components:

- IPT Server #1a – FACTS Ecosystem (Secure Dimensions, SatCen)
- IPT Server #1b – Trusted Teapot (Secure Dimensions)
- IPT Server #1c — Trusted Watermarking (Secure Dimensions)
- IPT Server #2a – Catalogue Server (Spacebel) and DID Registry (Spacebel)
- IPT Client #2b – Jupyter Notebook (Spacebel)
- IPT Client #3 – PDF Generator (SatCen)
- IPT Client #4 — UniResolver
- IPT Client #5 — UniVerifier

2.4.3. TIE Results

The TIEs demonstrated successful cross-component communication, data sharing, and validation across the various systems involved and demonstrated that the use of Smart Certificates based on Anonymous Credentials as well as W3C Verifiable Credentials can be used together to build an agile system for the trusted exchange of geospatial data. These experiments confirm that the FACTS ecosystem, Spacebel IPT Server, and SatCen's PDF Generator can function together seamlessly, enhancing the overall integration of IPT components used in this Testbed. Further details on the component interactions tested in the TIEs are detailed in the following Component Interactions section.

The TIEs were successful, as all planned interactions outlined in the TIE Matrix were executed successfully. However, there were some limitations with certain tested interactions. Notably, the Aries Cloud Agent software currently only supports JSON-LD 1.0, which meant that not all VCs from the Spacebel catalog could be utilized. Aries Cloud Agent Python (ACA-Py) is a foundation for building decentralized identity applications and services running in non-mobile environments. Additionally, the use of multi-arrays was not supported, preventing the use of GeoJSON encoding for expressing geometries. To overcome this, Spacebel created specific records for the TIE tests, using the alternative 'hasGeometry' encoding, which is also compliant with OGC standards.

The TIEs confirmed that the FACTS ecosystem can successfully generate DIDs from external services such as the Spacebel IPT Server and Notebook. Furthermore, the FACTS ecosystem demonstrated the ability to issue and verify Smart Certificates based on Anonymous Credentials, in addition to verifying W3C Verifiable Credentials created by the Spacebel IPT Server.

The TIEs showcased successful cross-component communication, data sharing, and validation between the various systems involved. They also demonstrated that Smart Certificates based on Anonymous Credentials and W3C Verifiable Credentials can be integrated to create a flexible system for the trusted exchange of geospatial data.

Further details on the component interactions tested during the TIEs are provided in the following Component Interactions section.

2.4.4. Component Interactions

The following interactions between the components were tested and validated as part of the TIEs.

(1) Trusted Watermarking → Spacebel Server: Fetch VC

The Trusted Watermarking process fetches the W3C VC from the Spacebel Catalogue. To be able to verify the VC, it then (See (6)) must retrieve the DID document corresponding to the issuer of the VC from the Spacebel DID Registry to get the public key of the issuer.

(2) Trusted Teapot → IPT Server #1a — FACTS Ecosystem (Smart Certificate Validation)

The Trusted Teapot process successfully validated Smart Certificates issued by the FACTS ecosystem, demonstrating cross-component credential verification.

(3) Jupyter Notebook → Spacebel Server

The Spacebel Jupyter Notebook successfully interacted with the Spacebel IPT servers (Catalogue and DID Registry), validating the interoperability between these components.

The Notebook interacts with the STAC API / OGC API Records "Catalogue" to discover EO product metadata records from several data providers ("issuers") and obtain the corresponding W3C VC and VP.

The Notebook then interacts with the "DID Registry" (via the DIDKit Resolver) to resolve the DID identifiers used in the VC/VP into DID documents during VC and VP validation. Keys provided in publicKeyJwk representation could be used. However, the key representation publicKeyBase58 could not be consumed by the DIDKit Resolver. Furthermore, the custom JSON-LD context file in the VC and VP had to be expanded in-line as the DIDKit library cannot resolve context documents included via URL.

The public SOV DID created on the ledger was not used in the VC in the catalogue as the DIDKit Resolver used by the Notebook was not able to resolve it without major changes to the DIDKit software, to have the IPT Server #1a ledger included among its backends.

(4) Notebook → IPT Server #1a (Secure Dimensions)

The Spacebel Notebook communicated with Secure Dimensions' IPT Server to create a public Sovereign Identity (SOV) DID, ensuring secure digital identity management. The Notebook could subsequently obtain the DID document for the SOV DID that was created.

(5) PDF Generator → FACTS Ecosystem

SatCen's PDF Generator integrated with the FACTS ecosystem to produce PDFs, verifying the compatibility and data exchange between these components. Create JWT.

(6) FACTS Ecosystem → Uniresolver → Spacebel Server

The FACTS ecosystem utilized the Uniresolver service to communicate with the Spacebel server, ensuring decentralized identity resolution and verification.

The FACTS Ecosystem was able to resolve the issuer Web DID available from the Spacebel DID Registry. However, only key information provided as publicKeyBase58 could be used in subsequent proof verifications.

(7) Trusted Watermarking → Spacebel VC → FACTS Ecosystem → Uniresolver → Spacebel Server

The Trusted Watermarking process initiated the interaction, verifying credentials through the Spacebel Verifiable Credential (VC) server and the FACTS ecosystem, followed by validation through the Uniresolver service and Spacebel server. The Uni Resolver is a utility developed at the Decentralized Identity Foundation (DIF) to resolve Decentralized Identifiers (DIDs) across many different DID methods, based on the W3C DID Core 1.0 and DID Resolution Specifications.

The process obtained an EO product metadata record from the Spacebel OGC/STAC Catalogue which provided the associated VC signed by its issuer (JAXA). The FACTS Ecosystem was able to resolve the issuer DID, i.e. obtain it from the Spacebel DID Registry and use the issuer's public key in base58 format to verify the signature in the VC. Only VC signed by DID with a public key in publicKeyBase58 format could be processed by the FACTS Ecosystem. The publicKeyJwk format was not accepted. Additionally, a simplified encoding for the JSON-LD context in the VC was used to avoid unsupported JSON-LD 1.1 features such as arrays of arrays (for geometry) or nested contexts.

(8) UniResolver → Spacebel DID Registry

The external UniResolver was able to resolve the Web DID created on the Spacebel DID Registry.

(9) UniVerifier → Spacebel Catalogue

The external UniVerifier was able to verify VC and VP produced by the Spacebel Catalogue for all of the organizations for which Web DID were created on the Spacebel DID Registry. The verification process implied also the resolving of the "issuer" and "holder" DID to DID documents and the use of the corresponding public keys provided as publicKeyBase58 of publicKeyJwk in the DID document.

3

OUTLOOK

3 OUTLOOK

3.1. Integrity, Provenance, and Trust Roadmap

The Testbed 20 IPT work established essential components of the Integrity, Provenance, and Trust Building Blocks through prototyping two use cases. These building blocks are intended to be encoding-agnostic, and the work confirms that the IPT concept is valid and should be further integrated into OGC API Standards and activities.

Recommendations:

Consider a Code Sprint in Q1 2025 to develop additional IPT Building Block prototypes.

2025 Activity Recommendations: Prototype and demonstrate IPT within new OGC API Standards. This work aims to secure data services by validating data integrity, building trust, and ensuring provenance verification through the implementation of OGC API standards.

2026 Activity Recommendations: Demonstrate the creation of custom, dynamic services at OGC API endpoints that verify authenticity, reliability, and security of data in terms of integrity, trust, and provenance. This could leverage existing and emerging OGC API Building Blocks, specifically IPT, for data delivery to the edge across various locations.

3.2. VC/VP Compliant Self-Sovereign Identity

Other Self-Sovereign Identity (SSI) aspects in EO use case with W3C compliant VC/VP need to be further explored beyond the Testbed 20 IPT Activity, as follows.

- Integration with OpenID Connect (OIDC) and OAuth: “OpenID for VC/VP” (OID4VC, OID4VP) .
- Decentralized Identifiers for individuals (“natural persons”), recorded in a “wallet.” In Testbed 20, EO use cases are limited to “legal persons” (DID recorded in a Registry instead of a wallet).
- Support for privacy and confidentiality via “Selective disclosure” (i.e., ability of holder to decide what information to share, VC formatted according to verifier’s data schema).

3.3. Provenance Outlook

The IPT concept encompasses both the certification of data authenticity and integrity, as well as the management of data provenance—tracking the data's journey through networks or local processing. This aspect is increasingly vital as geospatial data processing becomes more integrated into IT infrastructures.

Future IPT frameworks should reflect dynamic combinations of processes, APIs, and building blocks, providing flexibility to align with legislation and enablement requirements. Developing a geospatial IPT standard is recommended, offering a minimal framework that can adapt to various infrastructures. This standard should be collaboratively developed with other standardization bodies like DGIWG and W3C, addressing the need for compliance definitions in terms like GENUITY, LICENSED, AOI, CLASSIFICATION, DATA_USE, and SECURITY_ENVIRONMENT to promote interoperable governance.

4

SECURITY, PRIVACY AND ETHICAL CONSIDERATIONS

4 SECURITY, PRIVACY AND ETHICAL CONSIDERATIONS

The following highlight the security, privacy, and ethical considerations of IPT within the context of the FACTS Ecosystem IPT Instance.

4.1. Demonstrated FACTS Ecosystem Privacy Considerations

The first components of the FACTS Ecosystem deployed in this Testbed focus on Integrity, Privacy, and Trust through authentication. Integrity and Trust in the FACTS Ecosystem are maintained and provable through Smart Certificates that are issued and used in a privacy-preserving way, as the certificates are stored in the secure wallet of the user.

To illustrate the process with Secure Dimension's Trusted TeaPot process: Alice likes to brew tea. She finds the Trusted Teapot process in an (OGC API Records endpoint) catalog. From reading the description, she understands that a Smart Certificate from Lipton is required for the input image. Having such a certificate proves the authenticity of the image, as the Smart Certificate was produced by Lipton. So, before executing the Trusted Teapot, Alice contacts Lipton requesting a Smart Certificate based on the schema indicated by the Trusted Process's description. Lipton may offer an automated Smart Certificate issuing service for all images they have produced in the past. So, Alice could simply go to Liptons issuing service and receive a certificate that gets stored in her personal secure wallet. The communication with the Lipton issuing service is secure and privacy conserving, as it is a peer-to-peer communication. By requesting a Smart Certificate, Alice accepts the terms of use and privacy considerations involved in getting the certificate issued.

Once Alice has the Smart Certificate, she can now call the Trusted Process. Submitting a connection invite with the request, the Trusted Process can request Smart Certificate verification from Alice in a private communication. Because of the way the Smart Certificate is signed, using the CL signature scheme, it is possible for Alice to prove to the Trusted Teapot process that she possesses the appropriate Smart Certificate without revealing sensitive attributes.

A key privacy feature is that the certificate is not automatically shared among systems, giving Alice control over the information stored within it. Depending on the verifier's proof request, Alice can combine attributes from different certificates to respond to the request. Moreover, CL signatures allow Alice to commit a proof without involving the issuer, preventing Lipton from tracking where and how she uses the certificate.

Because the Trusted Process is aware of the attributes in the certificate and since this information is on the Blockchain, the proof can be very specific regarding attributes and their values. For example, the hash value in the certificate must match the hash of the input image

that Alice provides via the input URL. Verification of the issuer's DID guarantees that the certificate is genuinely issued by Lipton.

4.2. Demonstrated FACTS Ecosystem Ethical Considerations

An important ethical consideration of the Trusted Teapot Process is that Alice is not forced to disclose all information of a certificate and she can prevent the sharing of any attributes she is uncomfortable with disclosing. She retains control over what she chooses to reveal.

A variation of the verification process allows Alice to submit a proof proposal to the Trusted Process, outlining which attributes will be disclosed and which will remain hidden. While Alice may still choose to release attributes she is hesitant about in order to complete the process (like brewing tea), it is ultimately her decision to do so.

Additionally, Alice can use Zero Knowledge Proof (ZKP) predicates to offer further privacy protection. For example, in the scenario described earlier, Alice could avoid releasing the actual hash value of the image by instead proving that the hash value in the certificate matches a value requested by the Trusted Process, without disclosing the image itself.

Smart Certificates also enhance ethical considerations through the usage control information embedded within them. This could include policies that not only control processing behavior but also govern how privacy and ethical aspects are handled throughout the lifecycle of the certificate.

4.3. Data Privacy Considerations

As the FACTS Ecosystem grows, systems must be capable of managing both open-source and non-open-source data, which may contain sensitive or personal information not intended for public distribution. Protecting personal or sensitive data requires a multilevel security approach and releasability scheme. This is particularly relevant for location and time-based geospatial information that can easily be connected to individual identities or sensitive details. The system must ensure that sensitive information is protected while still allowing for the secure and responsible sharing of data.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Ben Domenico: OGC 10-092r3, *NetCDF Binary Encoding Extension Standard: NetCDF Classic and 64-bit Offset Format*. Open Geospatial Consortium (2011). https://portal.ogc.org/files/?artifact_id=43734.

[2] Akinori Asahara, Ryosuke Shibasaki, Nobuhiro Ishimaru, David Burggraf: OGC 14-084r2, *OGC® Moving Features Encoding Extension: Simple Comma Separated Values (CSV)*. Open Geospatial Consortium (2015). http://www.opengis.net/doc/IS/movingfeatures/csv-extension/1.0.0.

[3] Akinori Asahara, Ryosuke Shibasaki, Nobuhiro Ishimaru, David Burggraf: OGC 14-083r2, *OGC® Moving Features Encoding Part I: XML Core*. Open Geospatial Consortium (2015). http://www.opengis.net/doc/IS/movingfeatures/xmlcore/1.0.0.

[4] OGC: OGC 11-165r2: CF-netCDF3 Data Model Extension standard, 2012

[5] Hylerledger Aries Cloud Agent — https://github.com/hyperledger/aries-cloudagent-python)

[6] Hyperledger Fabric — https://www.hyperledger.org/projects/fabric

[7] Hyperledger Indy — https://www.hyperledger.org/projects/hyperledger-indy)

[8] Standardized Big Data Processing in Hybrid Clouds. In: Proceedings of the 4th International Conference on Geographical Information Systems Theory, Applications and Management — Volume 1: GISTAM, pp. 205–210. SciTePress (2018).

[9] pygeoapi — https://pygeoapi.io/)

[10] Lawrence Livermore National Laboratory: NetCDF CF Metadata Conventions – http://cfconventions.org/

[11] ESIP: Attribute Convention for Data Discovery (ACDD) – http://wiki.esipfed.org/index.php/

[12] W3C Decentralized Identifiers (DIDs) v1.0 — https://www.w3.org/TR/did-core/

[13] W3C did:web Method Specification — https://w3c-ccg.github.io/did-method-web

[14] Indy DID Method — https://github.com/hyperledger/indy-did-method

[15] EBSI DID Method for Legal Entities — https://hub.ebsi.eu/vc-framework/did/legal-entities

[16] Sovrin DID Method Specification, W3C Editor's Draft 22 April 2024 — https://sovrin-foundation.github.io/sovrin/spec/did-method-spec-template.html

[17] W3C Verifiable Credentials Data Model v2.0 — https://www.w3.org/TR/vc-data-model-2.0/

[18] W3C VerifiableCredential Data Integrity 1.0, W3C Candidate Recommendation Draft, 20 June 2024 — https://www.w3.org/TR/vc-data-integrity

[19] W3C Subresource Integrity, W3C Recommendation, 23 June 2016 — https://www.w3.org/TR/SRI/#the-integrity-attribute

[20] Cryptographic Hyperlinks — https://github.com/w3c-ccg/hashlink

[21] InterPlanetary File System — https://en.wikipedia.org/wiki/InterPlanetary_File_System

[22] W3C DID Implementation Guide v1.0 — https://www.w3.org/TR/did-imp-guide

[23] Verifiable Credentials Implementation Guidelines 1.0 — https://w3c.github.io/vc-imp-guide

[24] W3C Verifiable Credentials API v0.3 — https://w3c-ccg.github.io/vc-api

[25] OGC Testbed-19 Agile Reference Architecture Engineering Report — https://docs.ogc.org/per/23-050.html)

A

ANNEX A (NORMATIVE) ABBREVIATIONS/ACRONYMS

A ANNEX A (NORMATIVE) ABBREVIATIONS/ACRONYMS

API	Application Programming Interface
ARA	Agile Reference Architecture
ARD	Analysis Ready Data
BB	Building Block
CEOS	Committee on Earth Observation Satellites
CO	Collaborative Object
DCS	Data Centric Security
DID	Decentralized Identifier
DIF	Decentralized Identity Foundation
EBSI	European Blockchain Service Infrastructure
EO	Earth Observation
ESA	European Space Agency
FACTS	Federated Agile Collaborating Trusted Systems
FAIR	Findable Accessible Interoperable and Reusable
GDAL	Geospatial Data Abstraction Library
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
INPE	National Institute for Space Research (Instituto Nacional de Pesquisas Espaciais)
IPT	Integrity Provenance Trust
ISO	International Organization for Standardization

JSON	JavaScript Object Notation
JWK	JSON Web Key
JWS	JSON Web Signature
JWT	JSON Web Token
OGC	Open Geospatial Consortium
RDF	Resource Description Framework
REST	Representational State Transfer
SaaS	Software as a Service
SatCen	European Union Satellite Center
SC	Smart Certificate
SSI	Self-Sovereign Identity
STAC	SpatioTemporal Asset Catalog
TIE	Technology Integration Experiment
URL	Uniform Resource Locator
VA	Verifiable Attestation
VC	Verifiable Credentials
VP	Verifiable Presentations
W3C	World Wide Web Consortium
XML	eXtensible Markup Language

B

ANNEX B (INFORMATIVE) SECURE DIMENSIONS — FACTS ECOSYSTEM

ANNEX B (INFORMATIVE) SECURE DIMENSIONS – FACTS ECOSYSTEM

B.1. Trusted Teapot Process Example Plot

The following plot describes the entire history of how a developed process (source code) becomes a FACTS Trusted Process, and the interactions at runtime to "brew tea."

Actors in the plot:

- Long John Silver (LJS) is the developer of the Trusted Process
- Trusted Processes Open Foundation` (TPOF) is an organization that verifies a process implementation to be FACTS compliance
- Trusted Processes Inc. (TPI) is an organization that offers self-deployment of trusted processes into an OGC API Processes framework
- Alice is the actual real user who likes to brew a good pot of tea
- Joe is the coffee drinker that likes to brew coffee with the teapot
- Teapot is the actor that represents the trusted process in the FACTS ecosystem. Teapot actually acts as a verifier and as an issuer

The plot begins…

Long John Silver (LJS) is the developer of a piece of software called the `Trusted Teapot` that he wants to be deployed as a FACTS trusted process via the OGC API Processes standard. In order to obtain a Smart Certificate for the implementation, Long John Silver contacts `The Trusted Processes Open Foundation` (TPOF). The role of TPOF is to assure that the implementation provided by LJS is FACTS compliant. What does that mean? FACTS compliance means that the implementation essentially does three things: (i) only process input data, images in particular, for which the executing user has a valid Smart Certificate; (ii) the implementation creates a Smart Certificate for the produced output, in particular for images; and (iii) records provenance (execution metadata) in the FACTS Immutable Catalogue.

Once TPOF has successfully finished the implementation introspection, they issue a compliance certificate to LJS and list the implementation as FACTS compliance.

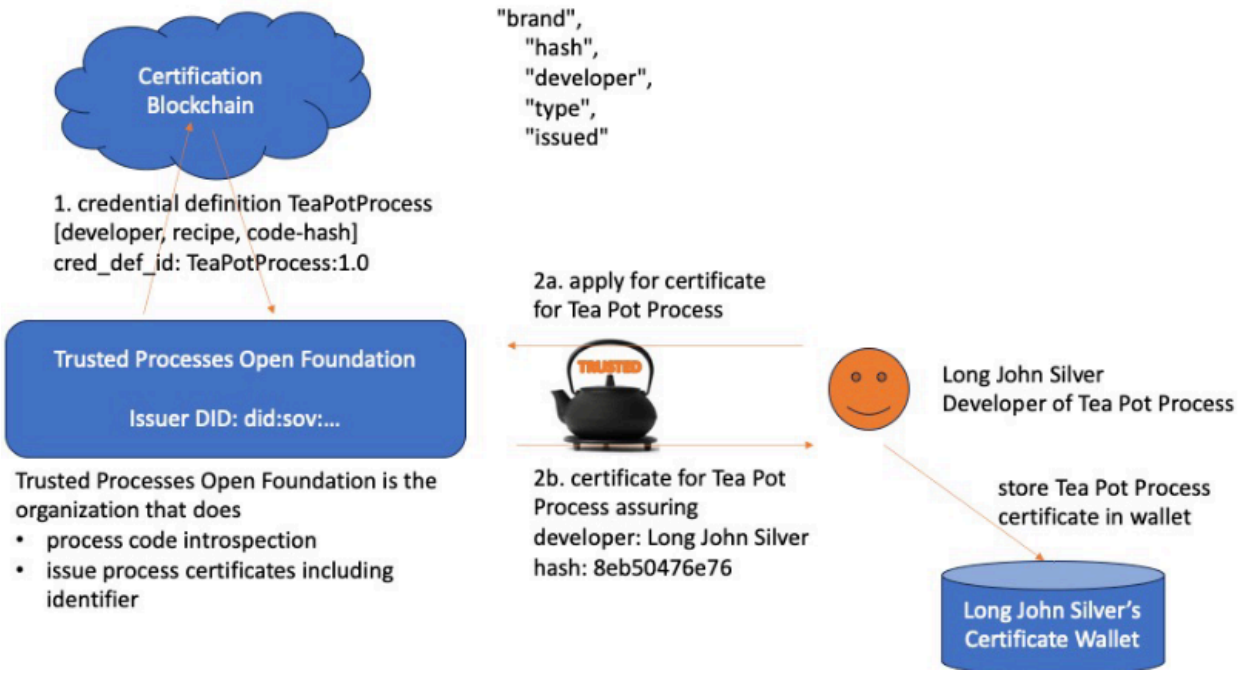


Figure B.1 — Trusted Teapot creation of a certificate for the source code

Long John Silver is now in the position to apply for a deployment of his implementation. LJS contacts the `Trusted Processes Inc.` (TPI) which operates an OGC API Processes framework that allows self-deployment of implementations if a compliance certificate from TPOF can be presented. As LJS is in possession of such a certificate, he is able to deploy the implementation, e.g. as a python script code, using the OGC API Processes — Part 2.

TPI acknowledges the successful deployment of the process by issuing a certificate of deployment to LJS and add the process to the catalogue of trusted processes.

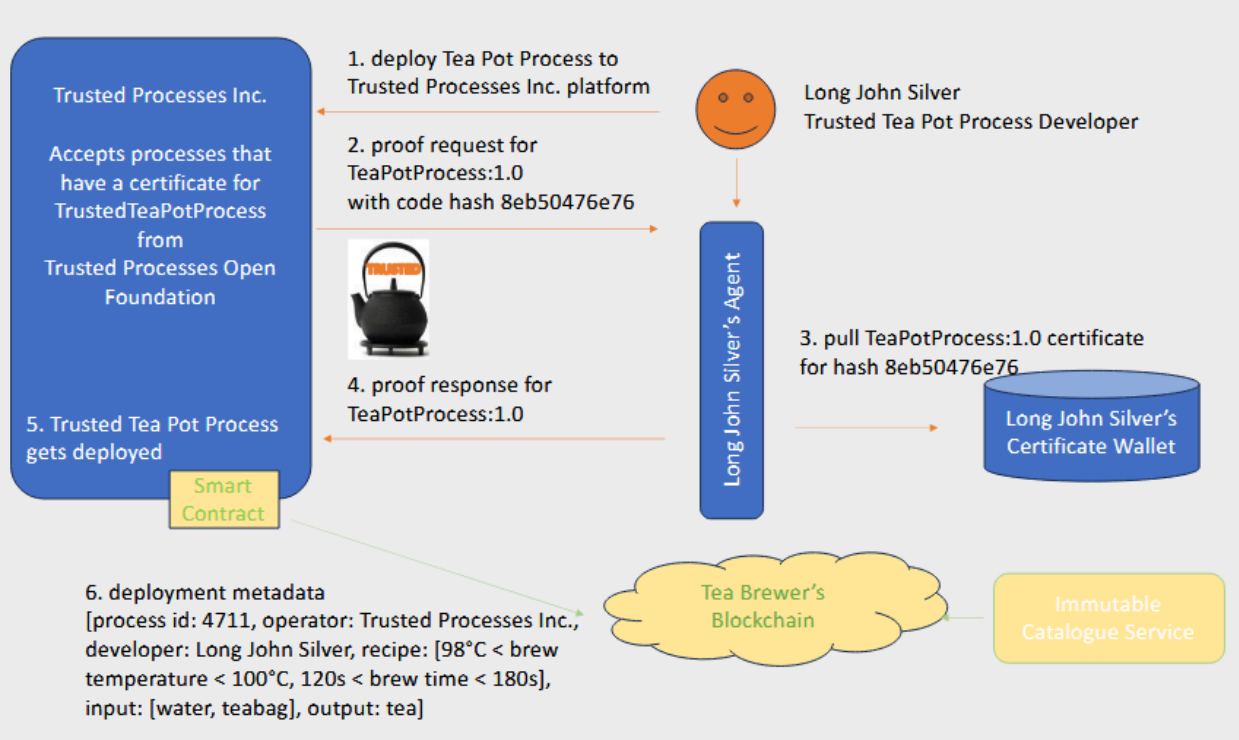


Figure B.2 — Trusted Teapot creation of a certificate for the deployment of the process

Lipton Tea produces images of tea bags of high quality issues FACTS Smart Certificates for these images to assert genuine verification. It could also be the purpose that Lipton only issues certificates to legitimate owners of their images or they could include processing and redistribution conditions in their certificates, but for simplicity in this plot, a certificate just states that a particular image has been produced by Lipton.

For this plot, two different images got created for which genuine proof is possible:

- The Lipton Earl Grey Classic image which is stored at IAFS URL https://iafs.demo.secure-dimensions.de/f/998845b59db786d3df28ba85b42c6a481da2b3843399ae5c3f49a06c1e564f87 and is identified by SHA-256 value `mYhFtZ23htPfKLqFtCxqSB2is4Qzma5cP0mgbB5WT4c=`.
- The Lipton Earl Grey Lemon image which is stored at IAFS URL https://iafs.demo.secure-dimensions.de/f/b278ee930b22baec09f977a625c1d51919575c36184e06d61bc74c952c243da0 and is identified by SHA-256 value `snjukwsiuuwJ+XemJcHVGRlXXDYYTgbWG8dMlSwkPaA=`.

In addition, there is one image for which no genuine proof is possible as no Smart Certificate will be available:

- The First computer Bug image which is stored at IAFS URL https://iafs.demo.secure-dimensions.de/f/511c63bee9bd062ef8280ca98fb00fbe5d00df0b97ba95ca6ad2bf01f55dd292

Alice is the user that likes to brew a good pot of tea. But in order to be sure it's genuine tea, Alice searches the FACTS Immutable Catalog to find the Trusted Teapot Process. As Alice can verify the developer of the process and can verify the deployment, she decides to use the Trusted Teapot from LJS deployed at TPI.

Figure B.3 — Searching for the Trusted Teapot

Once Alice has obtained the execution URL and read the process description, she understands that she needs a FACTS Smart Certificate for the input image representing her teabag. As Alice likes to brew a pot of Lipton Earl Grey Lemon, she contacts Lipton to get a certificate issued.

Equipped with a certificate for the Earl Grey Classic image, Alice can call the Trusted Teapot to have brewed a pot of tea.

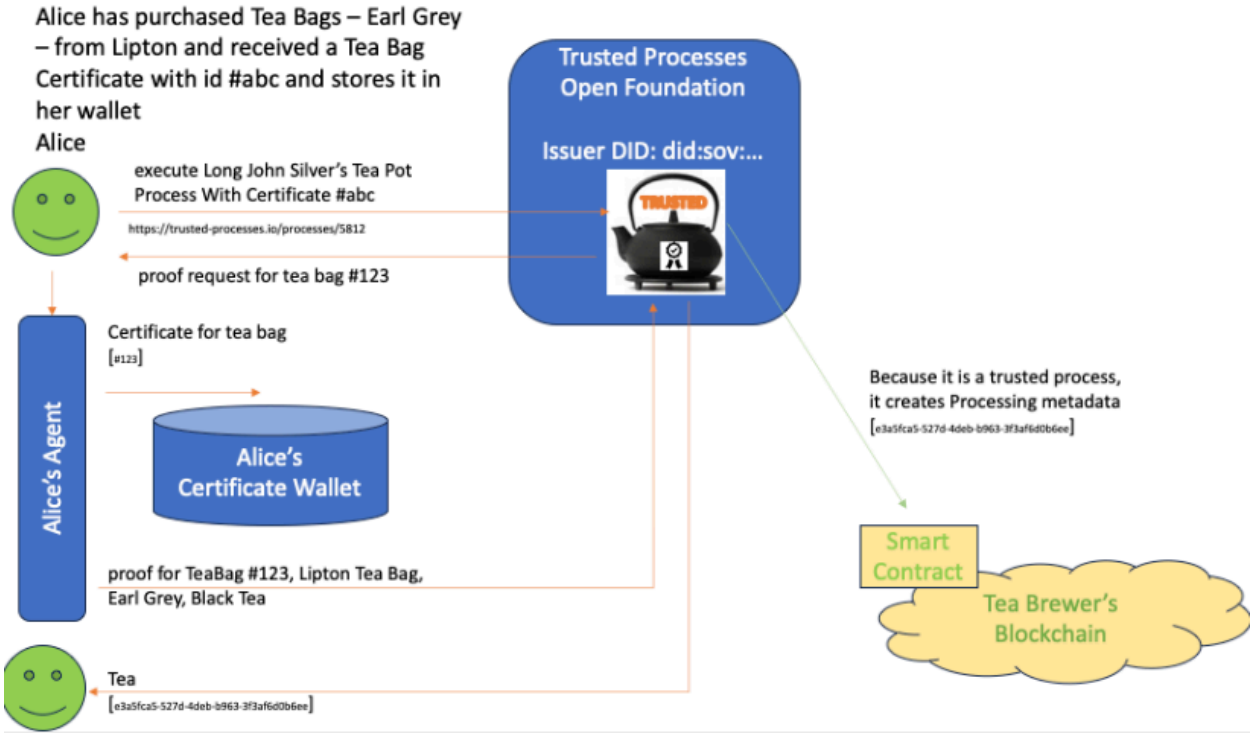


Figure B.4 — Trusted Teapot Execution with Certificates

The Trusted Teapot can verify that Alice is in possession of a Smart Certificate for the Lipton Earl Grey Classic image and produces the output image and the associated certificate. In addition — as a FACTS compliant process — the processing metadata is stored at the Immutable Catalog. This ensures that any user can — at a later time — request a certificate for the produced image. Inserting the execution metadata including the image hash to the Immutable Catalog associates the produced image with the Teapot process (or whoever the actual identified user is going to be).

Any attempt by `Bob` to brew coffee with the Trusted Teapot Process fails, as the process accepts images with certificates based upon the `Teabag 2.0` schema. Therefore, any attempt by Bob to use coffee pad images based on a different schema get rejected as requests are not associated with any teabag 2.0 Smart Certificate. Such an example is the First Computer Bug image.

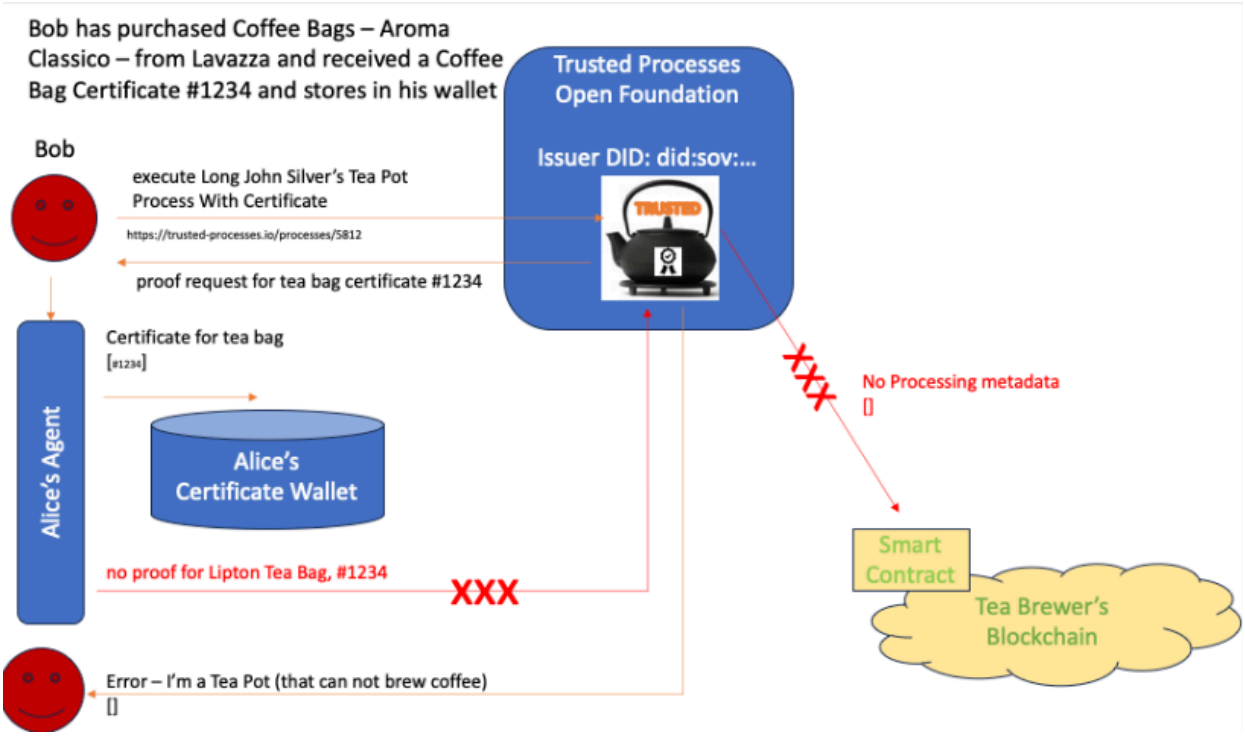
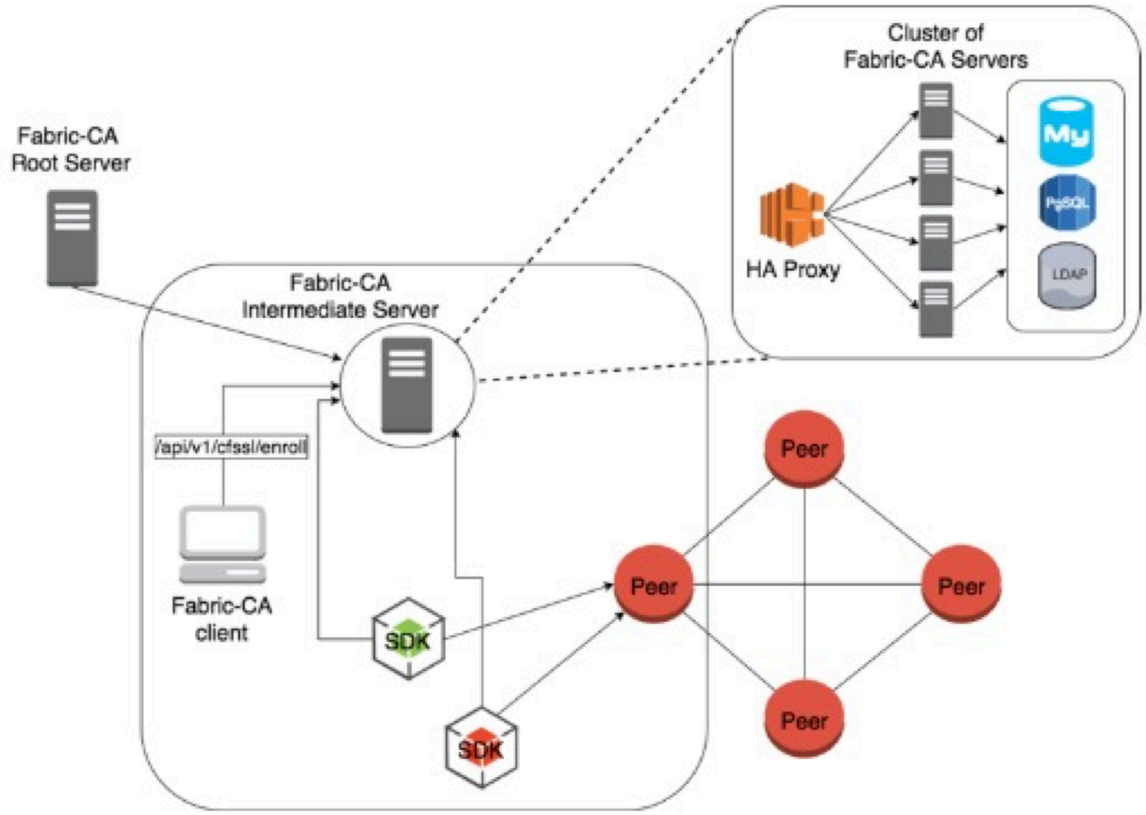


Figure B.5 — Trusted Teapot Execution without Certificates

B.2. FACTS Architecture and Implementation

TheFACTS architecture demonstrates the use of OGC API Standards to support an Immutable Catalog, Certification and Trusted Services. The Immutable Catalog is based on the Open Source Hyperledger-Fabric project and the certification is based on the Open Source Hyperledger-Indy project. Each fundamentally different distributed ledger is fit for purpose for its operational needs: The Fabric distributed ledger is designed to record provenance as transactions on the ledger and supports the implementation of an immutable catalog; the Indy distributed ledger is designed to store ledger-based identities and to support S-S-I.

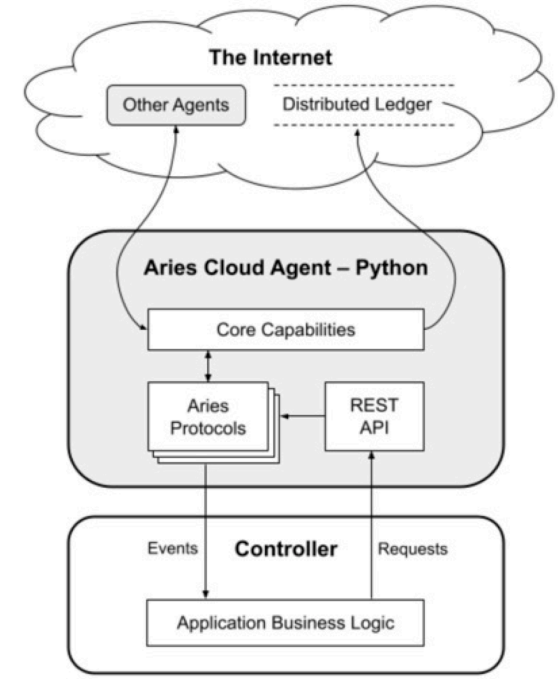
The architecture for the Provenance recording and the Immutable Catalogue are illustrated in the figure below.



https://editor.analyticsvidhya.com/uploads/440372.png

Figure B.6 — FACTS Provenance and Immutable Catalog Architecture

The architecture for the Smart Certification is mainly established by implementing the business logic based on the Open Source Hylerledger-Aries-Cloud-Agent API as illustrated in the following figure.



https://github.com/hyperledger/aries-cloudagent-python/README.md

Figure B.7 — FACTS Certification Architecture

During OGC Testbed 20, general purpose business logic was implemented that illustrates the roles issuer, holder and verifier.

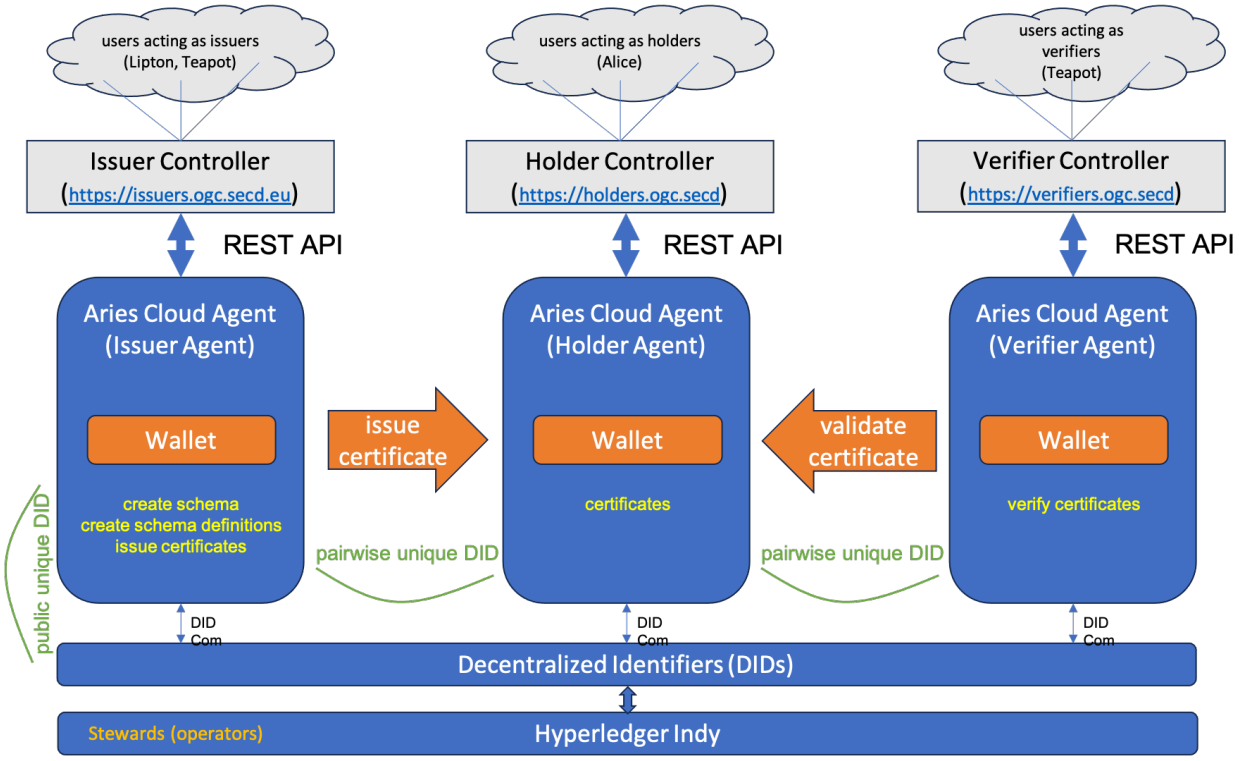


Figure B.8 — FACTS Detailed Architecture

B.2.1. Issuer and Holder Controller

A simple generic FACTS Issuer and Holder Controller was implemented as a Web-Application using Microsoft .NET 8. For supporting demonstration purposes, the application supports multi tenancy. To act as a FACTS issuer or holder, a user must first register and then login. For any user acting as an issuer, a public DID and a secure wallet is generated where the issued certificate structures (schemas) and the schema definitions (signed schemas) are stored for each user. For any user acting as a holder, a secure wallet is generated to store the certificates. The implementation is available here: https://issuers.ogc.secd.eu

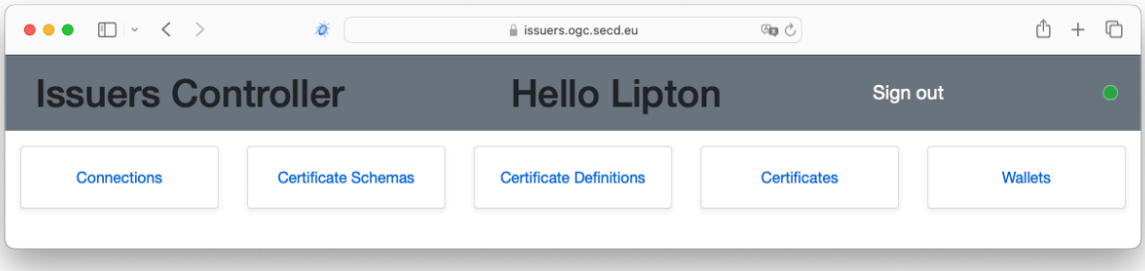


Figure B.9 — FACTS Issuer Controller - Main Menue

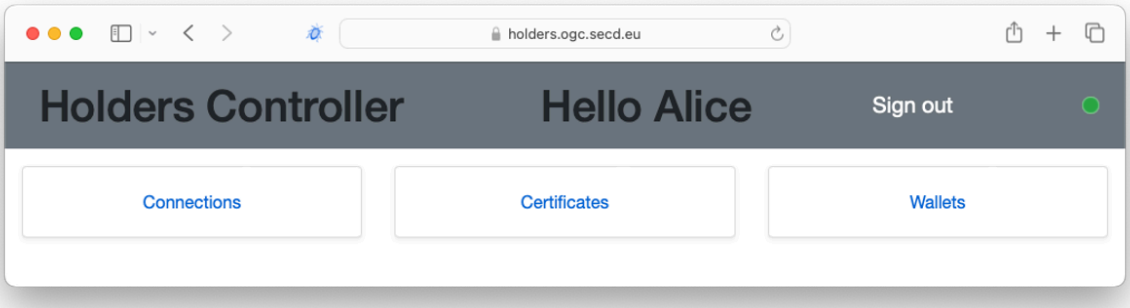


Figure B.10 — FACTS Holder Controller - Main Menue

The Issuer Controller supports the issuing of schemas and schema definitions that are stored on the ledger and in the issuer's wallet. For issuing Smart Certificates, the application supports connection management with holders and of course, the actual issuing of certificates. This proof of concept does not support the revocation of certificates.

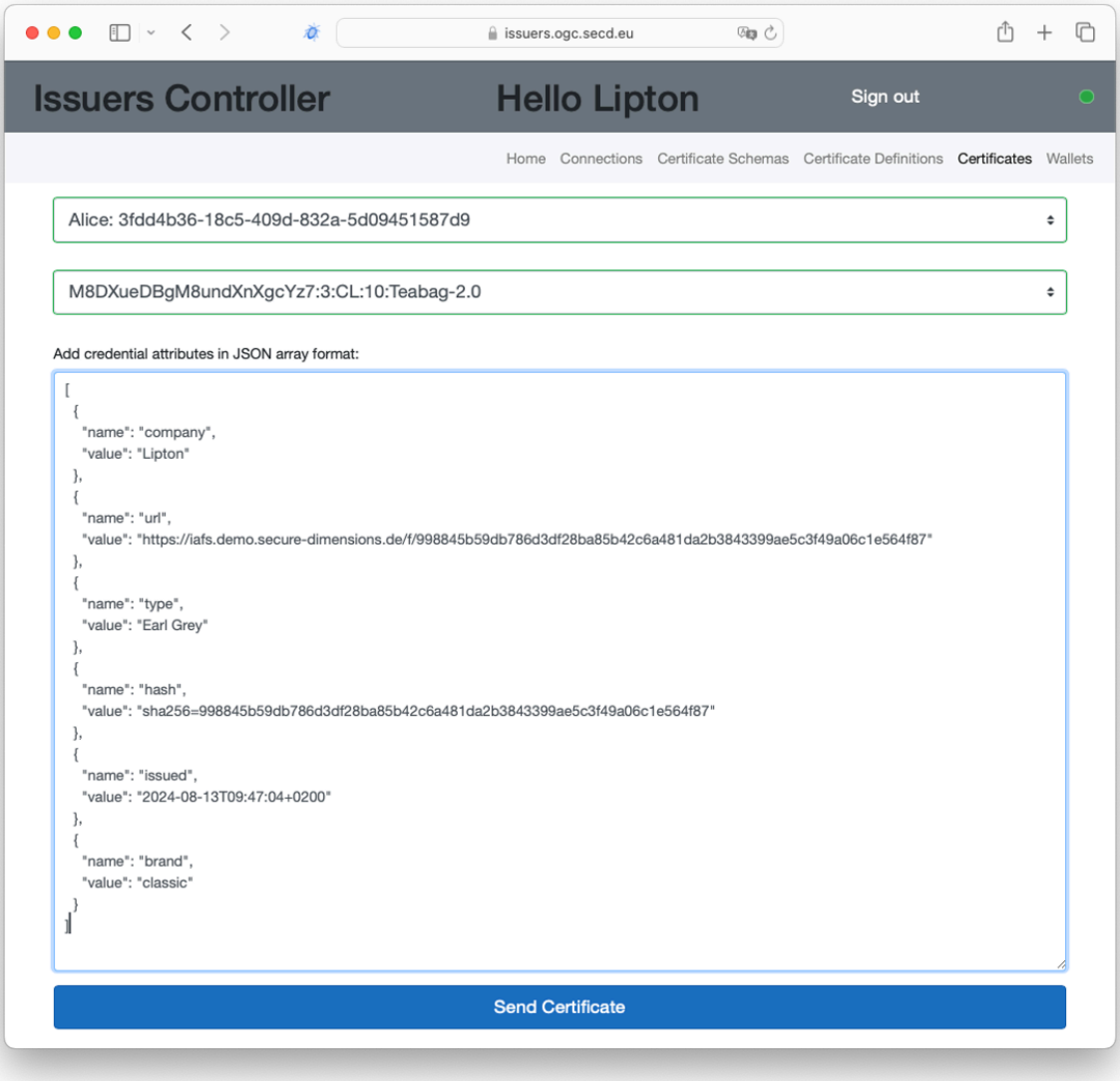


Figure B.11 — FACTS Issuer Controller - Issuing a Certificate

The Holder Controller supports the storage of Smart Certificates in the secure wallet of each user and to undertake connection management with issuers and verifiers. Different than the issuers, holders have no public DIDs. This is unnecessary as these users are not able to issue certificates. The implementation for the holder controller is available here: https://holders.ogc.secd.eu

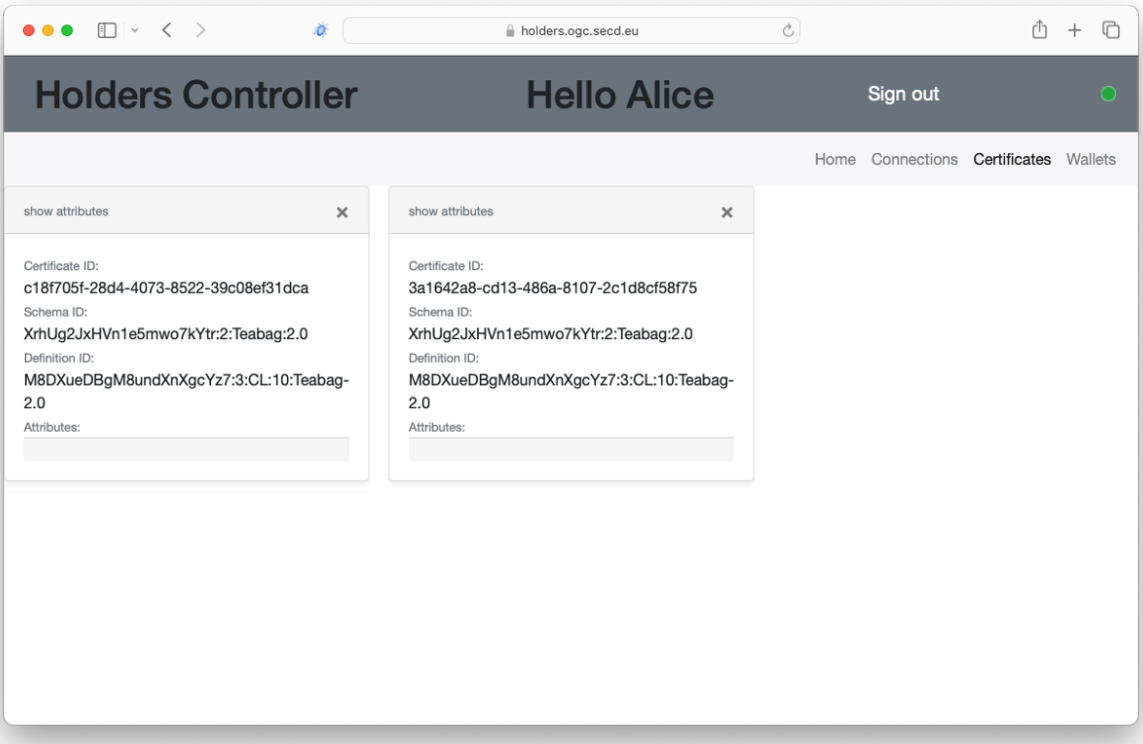


Figure B.12 — FACTS Holder Controller - Inspecting Certificates

B.2.2. Verifier Controller

A simple generic FACTS Verifier Controller was implemented as a Web-Application using NodeJS. For supporting demonstration purposes, the application supports multi tenancy. To act as a FACTS verifier, a user must first register and then login. As for holders, users acting as verifiers are not identifiable via a public ledger based DID. The application provides logic to list ledger-based schemas and their definitions. The ability to see existing schemas is important to create appropriate proof requests. To undertake an actual proof request, the implemented web-app includes a generic page that constructs a real proof request from simplified user input. The implementation of the verifier controlle is available here: https://verifiers.ogc.secd.eu

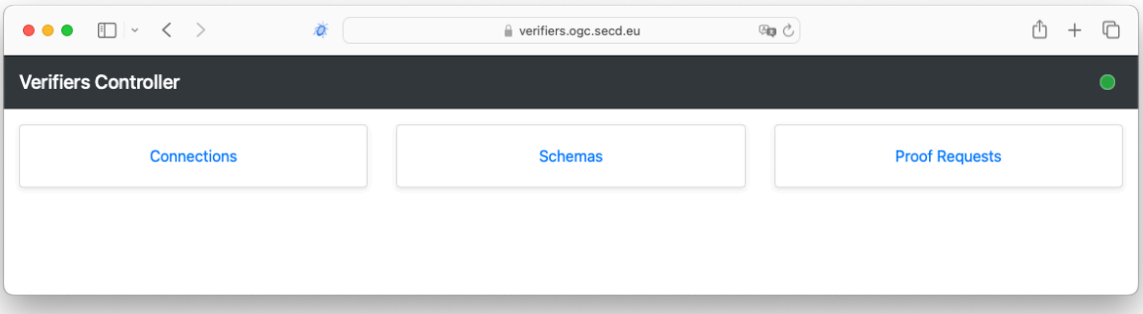


Figure B.13 — FACTS Verifier Controller - Main Menue

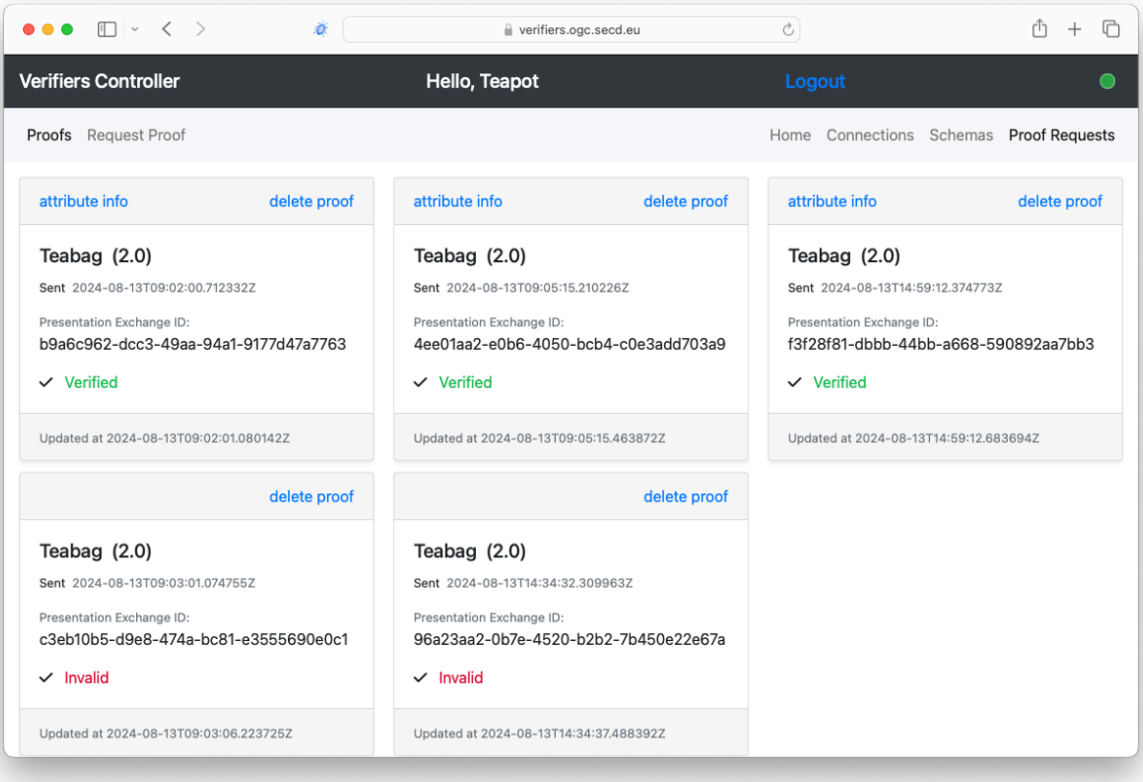


Figure B.14 — FACTS Verifier Controller - Inspecting Proof Requests

The following figure illustrates the construction of a proof request to user Alice (holder). The proof request is for four attributes of the schema (`company`, `url`, `type` and `hash`). The proof request has two conditions: (i) The `company` value must be equal to `Lipton` and (ii) the `hash` value must be equal to `sha256=998845b59db786d3df28ba85b42c6a481da2b3843399ae5c3f49a06c1e564f87` where the issuer must be `PqfTpsD1ck4VcrGfmbHQvK:3:CL:10:Teabag-2.0` (Lipton).

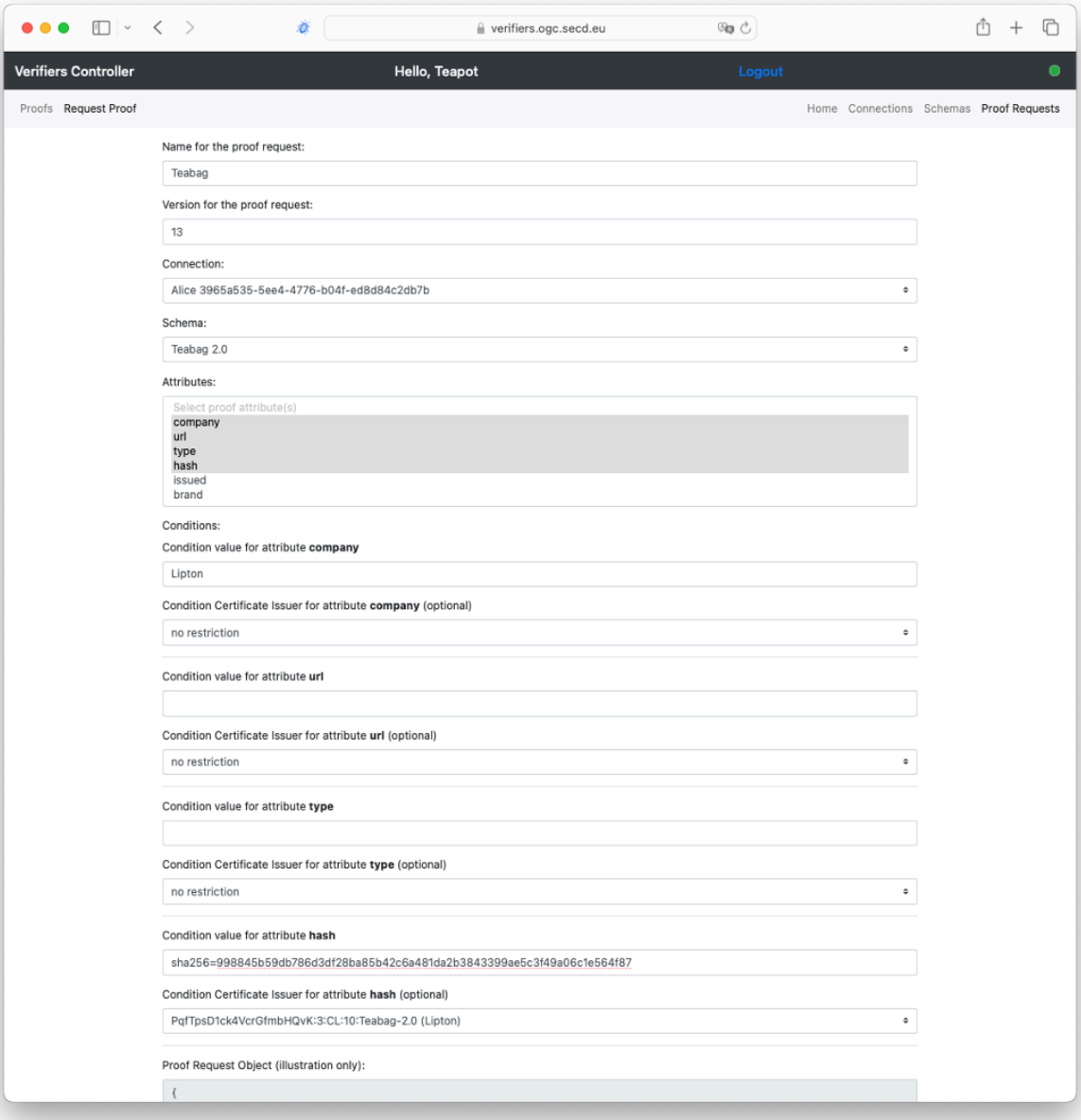


Figure B.15 — FACTS Verifier Controller - Creating a Proof Request

B.3. The Trusted Teapot Process

The OGC Testbed 20 `Trusted Teapot Process` (TTP) is part of the FACTS proof-of-concept implementation. The TTP is a python implementation, deployed as a process in the pygeoapi software stack. The implementation leverages implementations of the FACTS APIs for Certification to validate the Smart Certificate associated with the input image and to issue

a Smart Certificate for the generated output image. The process can be executed via an implementation of the OGC API Processes standard: https://processes.ogc.secd.eu/

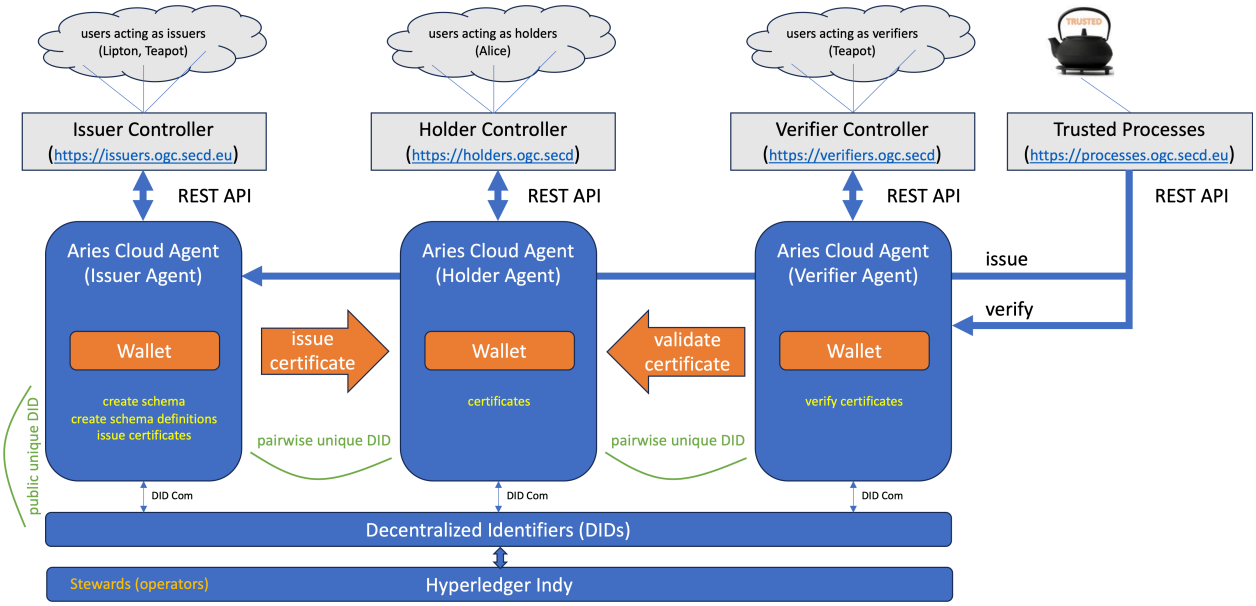


Figure B.16 — FACTS Detailed Architecture including the Trusted Teapot Process

A request to the Trusted Teapot requires — beside the input image and parameters — two FACTS connection invites from the caller. The first connection invitation (`sc_in`) is used to establish a connection for verifying the Smart Certificate associated with the input. The Trusted Teapot process acts as verifier `Teapot` on this connection. The second connection invitation is used to establish a connection for issuing a Smart Certificate for the produced image. The Trusted Teapot process acts as issuer `Teapot` on this connection.

Two new connection invitations can be created with the Holder's Controller (https://holders.ogc.secd.eu/connections/new). Please note that each execution of the process requires a fresh (new) connection invitation. The current implementation does no re-use existing connections.

The following screenshot illustrates the execution request input via the OpenAPI document.

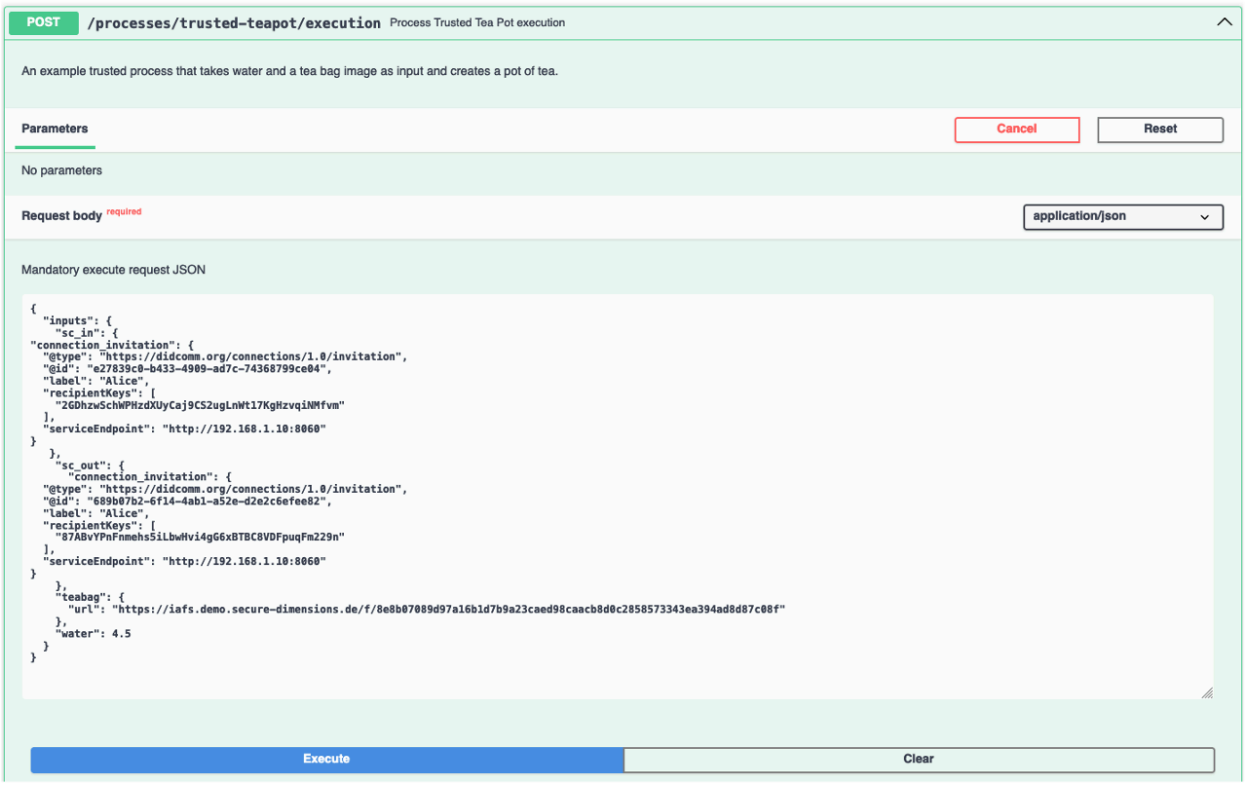


Figure B.17 — FACTS Trusted Teapot Process - Request Example

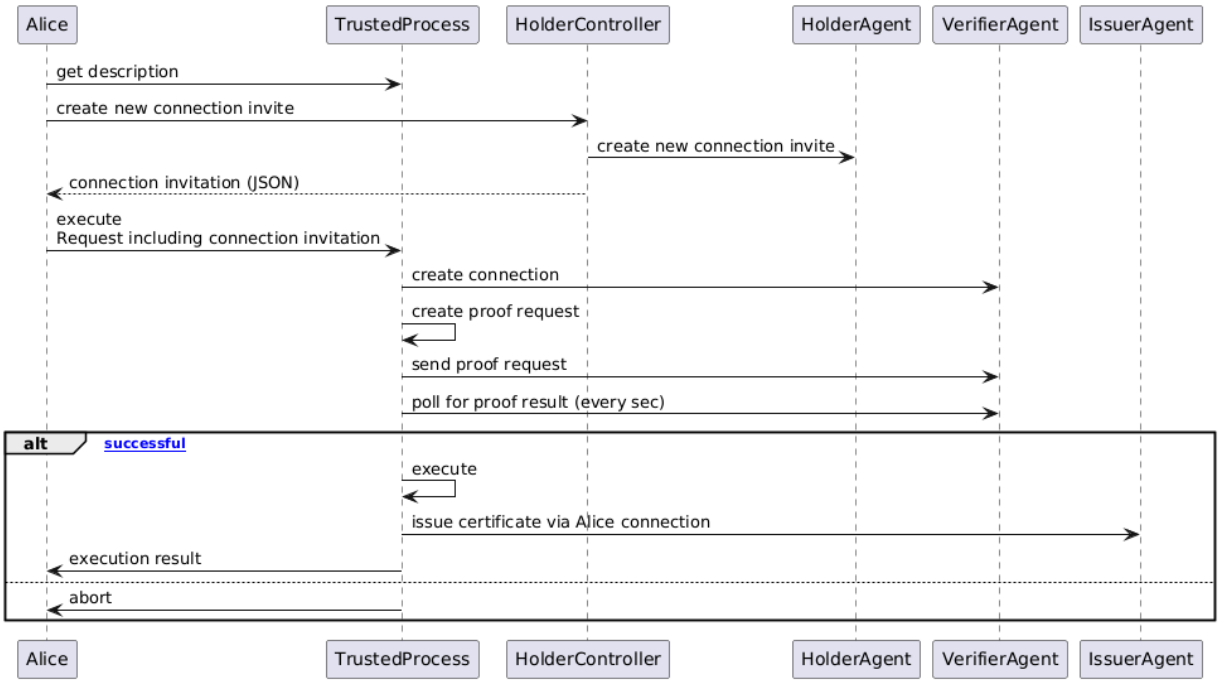


Figure B.18

A successful execution returns the following response:


```
{
  "id": "92c0a415-4fd6-4b9b-9c40-cc1286d617e4",
```

```
  "value": {
    "tea": {
      "amount": 4.254517959428636,
      "brand": "classic",
      "company": "Lipton",
      "type": "Earl Grey"
    },
    "image": {
      "url": "https://iafs.demo.secure-dimensions.de/f/b278ee930b22baec09f977a625
c1d51919575c36184e06d61bc74c952c243da0",
      "hash": "b278ee930b22baec09f977a625c1d51919575c36184e06d61bc74c952c243da0"
    },
    "asset": {
      "assetId": "92c0a415-4fd6-4b9b-9c40-cc1286d617e4",
      "type": "Tea Pot",
      "smartCertificate": {
        "cred_def_id": "QowX66AWGjeVd3H5fNryxd:3:CL:146:Deployment-2.0",
        "connection_invite_url": "https://issuers.ogc.secd.eu/connections/create-
invitation"
      },
      "productionDate": "2024-08-18T15:30:45.737794",
      "description": "I just created Lipton tea of brand classic and type Earl
Grey",
      "temperature": 99.8651465748281,
      "processingTime": 141.27943420379455,
      "water": {
        "source": "Mangfall",
        "hash": "32209ccbf8a8e509b9027698cc173343a2695e8ecdbe899bf5335a3100c956fc
"
      },
      "teabag": {
        "cred_def_id": "M8DXueDBgM8undXnXgcYz7:3:CL:10:Teabag-2.0",
        "hash": "54a02cb1b34de92594c37bc48c74ea4425a3362a371507927addcd988ec32067
"
      },
      "tea": {
        "hash": "0309cf77a8fe9071b856ab1c52de33cd43fc8b72535f12551953a43c0a996a73
"
      }
    }
  }
}
```


Listing B.1

B.4. Developers View: Smart Certificate issuing and verification

The following sub-sections illustrate the round-trip of issuing a Smart Certificate for an image using the Issuer and Holder Agent and verifying the certificate via the Verifier Agent. So, instead of using the user friendly controllers, the interactions are all based on the Hylerledger-Aries-Cloud-Agent REST API.

The script of interactions is:

- First, the issuer `Lipton` registers a schema called `Teabag 3.0`. This schema is then be used to issue certificates to the holder `Alice`.
- Second, the Issuer `Lipton` must register a schema credential definition upon which the actual certificate issuing can take place.
- Third, the Holder `Alice` contacts `Lipton` to get a Smart Certificate for the data image stored at https://iafs.demo.secure-dimensions.de/f/998845b59db786d3df28ba85b42c6a481da2b3843399ae5c3f49a06c1e564f87
- Four, the verifier `Teapot` requests `Alice` to proof being in possession of a FACTS Smart Certificate for that particular image

B.4.1. Issuer, Holder and Verifier Agents

For the illustration how to use the FACTS API, individual agents are deployed that act as dedicated Issuer, Holder and Verifier.

Issuer OpenAPI: https://issuer.ogc.secd.eu/api/doc

The issuer `Lipton`

- API-Key: <Issuer API-Key>
- Public DID: XrhUg2JxHVn1e5mwo7kYtr

Holder OpenAPI: https://holder.ogc.secd.eu/api/doc

The holder `Alice`

- API-Key: <Holder API-Key>
- Public DID: n/a

Verifier OpenAPI: https://verifier.ogc.secd.eu/api/doc

The verifier `Teapot`

- API-Key: <Verifier API-Key>
- Public DID: LyzzRZR9i9p1QnbrPzm4rT

B.4.2. Step I: Lipton creates Teabag schema

The issuer `Lipton` creates a schema (attribute structure) for a Smart Certificates to be used in the Trusted Teapot example.

```
curl -X 'POST' \
  'https://issuer.ogc.secd.eu/schemas' \
```

```
  -H 'accept: application/json' \
  -H 'X-API-KEY: <Issuer API-Key>' \
  -H 'Content-Type: application/json' \
  -d '{
  "attributes": [
    "company", "type", "brand", "issued", "hash", "url", "license"
  ],
  "schema_name": "Teabag",
  "schema_version": "3.0"
}'
```

Listing B.2 — Create certificate schema definition


```
{
    "sent": {
        "schema_id": "XrhUg2JxHVn1e5mwo7kYtr:2:Teabag:3.0",
        "schema": {
            "ver": "1.0",
            "id": "XrhUg2JxHVn1e5mwo7kYtr:2:Teabag:3.0",
            "name": "Teabag",
            "version": "3.0",
            "attrNames": [
                "brand",
                "issued",
                "url",
                "hash",
                "company",
                "license",
                "type"
            ],
            "seqNo": 175
        }
    },
    "schema_id": "XrhUg2JxHVn1e5mwo7kYtr:2:Teabag:3.0",
    "schema": {
        "ver": "1.0",
        "id": "XrhUg2JxHVn1e5mwo7kYtr:2:Teabag:3.0",
        "name": "Teabag",
        "version": "3.0",
        "attrNames": [
            "brand",
            "issued",
            "url",
            "hash",
            "company",
            "license",
            "type"
        ],
        "seqNo": 175
    }
}
```


Listing B.3 — Create certificate schema response

Next, the issuer `Lipton` binds the schema to the own public DID (applying signatures to the schema). This entitles the issuer to issue Smart Certificates upon the `Teabag:3.0` schema.

```
curl -X 'POST' \
  'https://issuer.ogc.secd.eu/credential-definitions' \
  -H 'accept: application/json' \
  -H 'X-API-KEY: <Issuer API-Key>' \
  -H 'Content-Type: application/json' \
  -d '{
```

```
  "revocation_registry_size": 1000,
  "schema_id": "XrhUg2JxHVn1e5mwo7kYtr:2:Teabag:3.0",
  "support_revocation": true,
  "tag": "Teabag-3.0"
}'
```

Listing B.4 — Create certificate schema definition

```
{
    "sent": {
        "credential_definition_id": "XrhUg2JxHVn1e5mwo7kYtr:3:CL:175:Teabag-3.0"
    },
    "credential_definition_id": "XrhUg2JxHVn1e5mwo7kYtr:3:CL:175:Teabag-3.0"
}
```

Listing B.5 — Create certificate schema definition response

Response contains credential_definition_id: `XrhUg2JxHVn1e5mwo7kYtr:3:CL:175:Teabag-3.0` that now can be used to issue certificates.

B.4.3. Step II: Creating a Connection between Issuer and Holder

The issuer can only issue a certificate to a holder via a private, secure peer-to-peer connection. The creation of a connection is a two step sequence between the issuer and the holder.

Usually, the holder contacts an issuer to have a Smart Certificate issued. Therefore, the first step is to create a so called connection invitation that is to be transmitted to the issuer. This could either take place via a simple Web form offered by the issuer's website or via email, etc.

The holder `Alice` creates a connection invite

```
curl -X 'POST' \
  'https://holder.ogc.secd.eu/out-of-band/create-invitation?auto_accept=
true&create_unique_did=true' \
  -H 'accept: application/json' \
  -H 'X-API-KEY: <Holder API-Key>' \
  -H 'Content-Type: application/json' \
  -d '{
  "accept": [
    "didcomm/aip1",
    "didcomm/aip2;env=rfc19"
  ],
  "alias": "Lipton",
  "goal": "To have issued a Teabag Smart Certificate",
  "goal_code": "issue-vc",
  "handshake_protocols": [
    "https://didcomm.org/didexchange/1.0"
  ],
  "metadata": {},
  "my_label": "Invitation to Issuer",
  "protocol_version": "1.1",
  "use_did_method": "did:peer:2",
  "use_public_did": false
}'
```

Listing B.6 — Holder creates a connection invitation

```
{
    "state": "initial",
```

```
    "trace": false,
    "invi_msg_id": "3efff3ad-cee4-46c0-8bd2-505e6ca449a7",
    "oob_id": "23797d22-18d4-43ed-a303-92fe6443735d",
    "invitation": {
        "@type": "https://didcomm.org/out-of-band/1.1/invitation",
        "@id": "3efff3ad-cee4-46c0-8bd2-505e6ca449a7",
        "label": "Invitation to Issuer",
        "handshake_protocols": [
            "https://didcomm.org/didexchange/1.0"
        ],
        "accept": [
            "didcomm/aip1",
            "didcomm/aip2;env=rfc19"
        ],
        "services": [
            "did:peer:2.Vz6MkjVQLtFobEPAvYqvH6nVgXKHvtzbCEK4WmnUW94R6N4Su.Ez6LSdm
DjW44v7Tcu3FY2UiMkqVR4Bo1Nx8EXjNGQ9SHw5pBB.SeyJpZCI6IiNkaWRjb21tLTAiLCJ0IjoiZGlkL
WNvbW11bmljYXRpb24iLCJwcmlvcml0eSI6MCwicmVjaXBpZW50S2V5cyI6WyIja2V5LTEiXSwiciI6W1
0sInMiOiJodHRwOi8vMTkyLjE2OC4xLjEwOjgwMzAifQ"
        ]
    },
    "invitation_url": "http://192.168.1.10:8030?oob=eyJAdHlwZSI6ICJodHRwczovL2RpZ
GNvbW0ub3JnL291dC1vZi1iYW5kLzEuMS9pbnZpdGF0aW9uIiwgIkBpZCI6ICIzZWZmZjNhZC1jZWU0LT
Q2YzAtOGJkMi01MDVlNmNhNDQ5YTciLCAibGFiZWwiOiAiSW52aXRhdGlvbiB0byBJc3N1ZXIiLCAiaGF
uZHNoYWtlX3Byb3RvY29scyI6IFsiaHR0cHM6Ly9kaWRjb21tLm9yZy9kaWRleGNoYW5nZS8xLjAiXSwg
ImFjY2VwdCI6IFsiZGlkY29tbS9haXAxIiwgImRpZGNvbW0vYWlwMjtlbnY9cmZpY
2VzIjogWyJkaWQ6cGVlcjoyLlZ6Nk1ralZRTHRGb2JFUEF2WXF2SDZuVmdYS0h2dHpiQ0VLNFdtblVXOT
RSNk40U3UuRXo2TFNkbURqVzQ0djdUY3UzRlkyVWlNa3FWUjRCbzFOeDhFWGpOR1E5U0h3NXBCQi5TZXl
KcFpDSTZJaU5rYVdSamIyMXRMVEFpTENKMElqb2laR2xrTFdOdmJXMTFibWxqWVhScGIyNGlMQ0p3Y21s
dmNtbDBlU0k2TUN3aWNtVmphWEJwWlc1MFMyVjVjeUk2V3lJamEyVjVMVEVpWFN3aWNpSTZXMTBzSW5Na
U9pSm9kSFJ3T2k4dk1Ua3lMakUyT0M0eExqRXdPamd3TXpBaWZRIl19"
}
```


Listing B.7 — Holder's connection invitation response

To finalize the connection, the issuer `Lipton` accepts the invitation.


```
curl -X 'POST' \
  'https://issuer.ogc.secd.eu/out-of-band/receive-invitation?alias=Holder' \
  -H 'accept: application/json' \
  -H 'X-API-KEY: <Issuer API-Key>' \
  -H 'Content-Type: application/json' \
  -d '{
        "@type": "https://didcomm.org/out-of-band/1.1/invitation",
        "@id": "3efff3ad-cee4-46c0-8bd2-505e6ca449a7",
        "label": "Invitation to Issuer",
        "handshake_protocols": [
            "https://didcomm.org/didexchange/1.0"
        ],
        "accept": [
            "didcomm/aip1",
            "didcomm/aip2;env=rfc19"
        ],
        "services": [
            "did:peer:2.Vz6MkjVQLtFobEPAvYqvH6nVgXKHvtzbCEK4WmnUW94R6N4Su.Ez6LSdm
DjW44v7Tcu3FY2UiMkqVR4Bo1Nx8EXjNGQ9SHw5pBB.SeyJpZCI6IiNkaWRjb21tLTAiLCJ0IjoiZGlkL
WNvbW11bmljYXRpb24iLCJwcmlvcml0eSI6MCwicmVjaXBpZW50S2V5cyI6WyIja2V5LTEiXSwiciI6W1
0sInMiOiJodHRwOi8vMTkyLjE2OC4xLjEwOjgwMzAifQ"
        ]
    }'
```


Listing B.8 — Issuer accept the connection invitation

```
{
    "state": "deleted",
    "created_at": "2024-08-19T08:07:43.598544Z",
    "updated_at": "2024-08-19T08:07:43.598544Z",
    "trace": false,
    "oob_id": "d715bb03-7f7b-498f-8bea-63bedfa40ffc",
    "invi_msg_id": "3efff3ad-cee4-46c0-8bd2-505e6ca449a7",
    "invitation": {
        "@type": "https://didcomm.org/out-of-band/1.1/invitation",
        "@id": "3efff3ad-cee4-46c0-8bd2-505e6ca449a7",
        "label": "Invitation to Issuer",
        "handshake_protocols": [
            "https://didcomm.org/didexchange/1.0"
        ],
        "accept": [
            "didcomm/aip1",
            "didcomm/aip2;env=rfc19"
        ],
        "services": [
            "did:peer:2.Vz6MkjVQLtFobEPAvYqvH6nVgXKHvtzbCEK4WmnUW94R6N4Su.Ez6LSdm
DjW44v7Tcu3FY2UiMkqVR4Bo1Nx8EXjNGQ9SHw5pBB.SeyJpZCI6IiNkaWRjb21tLTAiLCJ0IjoiZGlkL
WNvbW11bmljYXRpb24iLCJwcmlvcml0eSI6MCwicmVjaXBpZW50S2V5cyI6WyIja2V5LTEiXSwiciI6W1
0sInMiOiJodHRwOi8vMTkyLjE2OC4xLjEwOjgwMzAifQ"
        ]
    },
    "connection_id": "148300fa-2421-40b8-8f8f-2a93cc72a02d",
    "role": "receiver",
    "multi_use": false
}
```


Listing B.9 — Issuer's connection acceptance response

The interactions above result in the following connection details:

- Issuer→Holder: connection_id: `148300fa-2421-40b8-8f8f-2a93cc72a02d`

The issuer can verify the connection state to be `active` via the Agents API based upon the response.

```
curl -X 'GET' \
  'https://issuer.ogc.secd.eu/connections/148300fa-2421-40b8-8f8f-2a93cc72a02d' \
  -H 'accept: application/json' \
  -H 'X-API-KEY: <Holder API-Key>'
```

Listing B.10 — Issuer verifies the connection status


```
{
  "state": "active",
  "created_at": "2024-08-19T08:07:43.504111Z",
  "updated_at": "2024-08-19T08:07:43.897670Z",
  "connection_id": "148300fa-2421-40b8-8f8f-2a93cc72a02d",
  "my_did": "LVSyC5bQMLESNGWpQoNh3a",
  "their_did": "T9ZX4uYKnbWuEE4t5KW1EJ",
  "their_label": "Invitation to Issuer",
  "their_role": "inviter",
  "connection_protocol": "didexchange/1.0",
  "rfc23_state": "completed",
  "invitation_key": "639JJ1Z9tqgTSM5aRDXqgDjw5RKLpRpA5mZaJnT5SqfX",
  "invitation_msg_id": "3efff3ad-cee4-46c0-8bd2-505e6ca449a7",
  "request_id": "6c802a0b-9ed8-4b0f-800e-9e99e134956c",
  "accept": "auto",
```

```
  "invitation_mode": "once",
  "alias": "Holder",
  "their_public_did": "did:peer:2.Vz6MkjVQLtFobEPAvYqvH6nVgXKHvtzbCEK4WmnUW94R6N4
Su.Ez6LSdmDjW44v7Tcu3FY2UiMkqVR4Bo1Nx8EXjNGQ9SHw5pBB.SeyJpZCI6IiNkaWRjb21tLTAiLCJ
0IjoiZGlkLWNvbW11bmljYXRpb24iLCJwcmlvcml0eSI6MCwicmVjaXBpZW50S2V5cyI6WyIja2V5LTEi
XSwiciI6W10sInMiOiJodHRwOi8vMTkyLjE2OC4xLjEwOjgwMzAifQ"
}
```


Listing B.11 — Issuer's connection details response

The holder however must search in the wallet if the connection invitation was exchanged into a connection.

```
curl -X 'GET' \
  'https://holder.ogc.secd.eu/connections?invitation_msg_id=3efff3ad-cee4-46c0-
8bd2-505e6ca449a7&limit=100&offset=0' \
  -H 'accept: application/json' \
  -H 'X-API-KEY: <Holder API-Key>'
```

Listing B.12 — Holder searches for the connection and status


```
{
  "results": [
    {
      "state": "active",
      "created_at": "2024-08-19T07:58:50.810729Z",
      "updated_at": "2024-08-19T08:07:43.958233Z",
      "connection_id": "ed3d7ffb-ab78-495d-815d-455131da95b5",
      "my_did": "T9ZX4uYKnbWuEE4t5KW1EJ",
      "their_did": "LVSyC5bQMLESNGWpQoNh3a",
      "their_label": "Issuer.agent",
      "their_role": "invitee",
      "connection_protocol": "didexchange/1.0",
      "rfc23_state": "completed",
      "invitation_key": "639JJ1Z9tqgTSM5aRDXqgDjw5RKLpRpA5mZaJnT5SqfX",
      "invitation_msg_id": "3efff3ad-cee4-46c0-8bd2-505e6ca449a7",
      "request_id": "6c802a0b-9ed8-4b0f-800e-9e99e134956c",
      "accept": "auto",
      "invitation_mode": "once",
      "alias": "Lipton"
    }
  ]
}
```


Listing B.13 — Holders's connections response

This indicates an active connection with the issuer:

- Holder→Issuer: connection_id: `3efff3ad-cee4-46c0-8bd2-505e6ca449a7`

B.4.4. Step III: Issuing a Smart Certificate

The issuer `Lipton` is going to issue a Smart Certificate to the holder `Alice` for the image of a teabag `Lipton Earl Grey` of type `classic`: https://iafs.demo.secure-dimensions.de/f/b278ee930b22baec09f977a625c1d51919575c36184e06d61bc74c952c243da0 using the established peer-to-peer connection from above.

```
curl -X 'POST' \
```

```
  'https://issuer.ogc.secd.eu/issue-credential-2.0/send' \
  -H 'accept: application/json' \
  -H 'X-API-KEY: <Issuer API-Key>' \
  -H 'Content-Type: application/json' \
  -d '{
  "auto_remove": true,
  "comment": "string",
  "connection_id": "148300fa-2421-40b8-8f8f-2a93cc72a02d",
  "credential_preview": {
    "@type": "issue-credential/2.0/credential-preview",
    "attributes": [
      {
        "name": "company",
        "value": "Lipton"
      },
      {
        "name": "brand",
        "value": "Earl Grey"
      },
      {
        "name": "type",
        "value": "Classic"
      },
      {
        "name": "issued",
        "value": "Mon Aug 19 10:34:45 CEST 2024"
      },
      {
        "name": "hash",
        "value": "sha256=998845b59db786d3df28ba85b42c6a481da2b3843399ae5c3f49a06c
1e564f87"
      },
      {
        "name": "url",
        "value": "https://iafs.demo.secure-dimensions.de/f/998845b59db786d3df28ba
85b42c6a481da2b3843399ae5c3f49a06c1e564f87"
      },
      {
        "name": "license",
        "value": "CC-BY 4.0"
      }
    ]
  },
  "filter": {
    "indy": {
      "cred_def_id": "XrhUg2JxHVn1e5mwo7kYtr:3:CL:175:Teabag-3.0",
      "issuer_did": "XrhUg2JxHVn1e5mwo7kYtr",
      "schema_id": "XrhUg2JxHVn1e5mwo7kYtr:2:Teabag:3.0",
      "schema_issuer_did": "XrhUg2JxHVn1e5mwo7kYtr",
      "schema_name": "Teabag",
      "schema_version": "3.0"
    }
  },
  "trace": false,
  "verification_method": "string"
}'
```

Listing B.14 — Lipton issues Certificate to Alice


```
{
    "state": "offer-sent",
    "created_at": "2024-08-19T08:39:36.258049Z",
    "updated_at": "2024-08-19T08:39:36.258049Z",
```

```
    "trace": false,
    "cred_ex_id": "0ea956b1-39b4-4e6d-9f82-9ab86aaae01c",
    "connection_id": "148300fa-2421-40b8-8f8f-2a93cc72a02d",
    "thread_id": "053ee666-e18f-4774-a6f6-a582f73bd5c9",
    "initiator": "self",
    "role": "issuer",
    "cred_preview": {
        "@type": "https://didcomm.org/issue-credential/2.0/credential-preview",
        "attributes": [
            {
                "name": "company",
                "value": "Lipton"
            },
            {
                "name": "brand",
                "value": "Earl Grey"
            },
            {
                "name": "type",
                "value": "Classic"
            },
            {
                "name": "issued",
                "value": "Mon Aug 19 10:34:45 CEST 2024"
            },
            {
                "name": "hash",
                "value": "sha256=998845b59db786d3df28ba85b42c6a481da2b3843399ae5c
3f49a06c1e564f87"
            },
            {
                "name": "url",
                "value": "https://iafs.demo.secure-dimensions.de/f/998845b59db786
d3df28ba85b42c6a481da2b3843399ae5c3f49a06c1e564f87"
            },
            {
                "name": "license",
                "value": "CC-BY 4.0"
            }
        ]
    },
    "cred_proposal": {
        "@type": "https://didcomm.org/issue-credential/2.0/propose-credential",
        "@id": "d4c38f3e-d793-4a66-b3ee-6f929cc99183",
        "comment": "string",
        "credential_preview": {
            "@type": "https://didcomm.org/issue-credential/2.0/credential-
preview",
            "attributes": [
                {
                    "name": "company",
                    "value": "Lipton"
                },
                {
                    "name": "brand",
                    "value": "Earl Grey"
                },
                {
                    "name": "type",
                    "value": "Classic"
                },
                {
                    "name": "issued",
```

```
                    "value": "Mon Aug 19 10:34:45 CEST 2024"
                },
                {
                    "name": "hash",
                    "value": "sha256=998845b59db786d3df28ba85b42c6a481da2b3843399
ae5c3f49a06c1e564f87"
                },
                {
                    "name": "url",
                    "value": "https://iafs.demo.secure-dimensions.de/f/998845b59d
b786d3df28ba85b42c6a481da2b3843399ae5c3f49a06c1e564f87"
                },
                {
                    "name": "license",
                    "value": "CC-BY 4.0"
                }
            ]
        },
        "formats": [
            {
                "attach_id": "indy",
                "format": "hlindy/cred-filter@v2.0"
            }
        ],
        "filters~attach": [
            {
                "@id": "indy",
                "mime-type": "application/json",
                "data": {
                    "base64": "eyJjcmVkX2RlZl9pZCI6ICJYcmhVZzJKeEhWbjFlNW13bzdrWX
RyOjM6Q0w6MTc1OlRlYWJhZy0zLjAiLCAiaXNzdWVyX2RpZCI6ICJYcmhVZzJKeEhWbjFlNW13bzdrWXR
yIiwgInNjaGVtYV9pZCI6ICJYcmhVZzJKeEhWbjFlNW13bzdrWXRyOjI6VGVhYmFnOjMuMCIsICJzY2hl
bWFfaXNzdWVyX2RpZCI6ICJYcmhVZzJKeEhWbjFlNW13bzdrWXRyIiwgInNjaGVtYV9uYW1lIjogIlRlY
WJhZyIsICJzY2hlbWFfdmVyc2lvbiI6ICIzLjAifQ=="
                }
            }
        ]
    },
    "cred_offer": {
        "@type": "https://didcomm.org/issue-credential/2.0/offer-credential",
        "@id": "053ee666-e18f-4774-a6f6-a582f73bd5c9",
        "~thread": {},
        "comment": "create automated v2.0 credential exchange record",
        "credential_preview": {
            "@type": "https://didcomm.org/issue-credential/2.0/credential-
preview",
            "attributes": [
                {
                    "name": "company",
                    "value": "Lipton"
                },
                {
                    "name": "brand",
                    "value": "Earl Grey"
                },
                {
                    "name": "type",
                    "value": "Classic"
                },
                {
                    "name": "issued",
                    "value": "Mon Aug 19 10:34:45 CEST 2024"
                },
```

```
                {
                    "name": "hash",
                    "value": "sha256=998845b59db786d3df28ba85b42c6a481da2b3843399
ae5c3f49a06c1e564f87"
                },
                {
                    "name": "url",
                    "value": "https://iafs.demo.secure-dimensions.de/f/998845b59d
b786d3df28ba85b42c6a481da2b3843399ae5c3f49a06c1e564f87"
                },
                {
                    "name": "license",
                    "value": "CC-BY 4.0"
                }
            ]
        },
        "formats": [
            {
                "attach_id": "indy",
                "format": "hlindy/cred-abstract@v2.0"
            }
        ],
        "offers~attach": [
            {
                "@id": "indy",
                "mime-type": "application/json",
                "data": {
                    "base64": "eyJzY2hlbWFfaWQiOiAiWHJoVWcySnhIVm4xZTVtd283a1l0cj
oyOlRlYWJhZzozLjAiLCAiY3JlZF9kZWZfaWQiOiAiWHJoVWcySnhIVm4xZTVtd283a1l0cjozOkNMOjE
3NTpUZWFiYWctMy4wIiwgImtleV9jb3JyZWN0bmVzc19wcm9vZiI6IHsiYyI6ICI1MDU4MzAzNzkxMDEx
OTI4NzY1OTA1OTk4Mjg0MDE5MDM2Nzc1OTU4ODkzNzgwNDUzMjc3OTE0OTE2NjI3NjMyMjEzNzMwMjYwO
TI2NyIsICJ4el9jYXAiOiAiODk1MzE5MTkyOTI3MDczOTg1MTAyNjIyODA5Nzk2OTA5NDUzMDQ5MTkxNz
Y2NTUxNDc0MjA5NDE1MTM0OTgxOTM2NTM2NjI0NjU0NjU1NzY5ODM1NTQwOTU4OTA1NzE2OTk5NzA5NTk
4NzQ3NjI5Mjk4OTQ2NjUzNjU1MzgyOTQyOTI4MDE5MzAzOTQ0Njk0OTg3MzQ4MzM1NDg4MDAyNTQ4NjA1
MjM1NDM4NTkzOTUzMDg1MTI1MDI5NDIyODY1ODQ1Mzg5MTMzNDkwNTY4NTE1NjkyMjU2Mjg0NDgyNDQzN
jk4MDQxMzUyMzM4MjYzMzA1MjU4NzkwNjk0MzMwNDYyNDgxNDI4Mzg0MzQ0ODk2NTMyODIzNDY1NjQwNj
g4NzQwNDkxNDI3MzAwOTI1Nzg3NjY1Mjg1ODY4Njk2MjcwNTE4MzIwNDU3NDYyNTk5Nzc5NDMzOTgyOTU
4OTQ5Mjc0ODY0OTQ3MTM0MTk5MzYzNzg2NjQ4MjgwODUwMDczOTMzOTA0ODI0NjI3NTI0MjIyNzY3OTkx
MjU2MzI1MzE5MjUxMjY2NDM1ODQzNDI5NDEwMzA3NTAyNTI5NTI5NDM3MTQyMDk3ODIxNzQ5MzQ1ODA1O
DUzNzI5MzY5ODQ5Nzk1MzAyNTE1NTU4ODQyMDgyMDIyMDk3MjEzMTg0NDM3MzY5MDYxMDAyNTExMDYwND
I1OTE5MzU3NzA5NjM5MTQwOTY4ODI5NDI1MDE3ODUwNTc5MDI3NTIyOTI1ODI5NzIyNTA3ODI5MjEyNTI
0MDE0ODk3NjMwMTIzNjM3MDI3OTAzMTQyNjg1NTk3NDEyNDk2NzY1ODA2NTgxMTkxNTcyNTc0NzA2MzI5
Mzg0MjY5NDU5NzE3ODg4MDYxNDMwMTE4MjQ2MTEzMTg4MTI1NTM1MDI0IiwgInhyX2NhcCI6IFtbInVyb
CIsICI5OTU3ODg5NjAzMTczMDgxMTUwNDcwMjQzMTI2OTQ5MDM3NzA5ODczNjI3NDk4Mjc3ODA5NTQ4ND
g3NDIzNTIzMzgzMDI5MTk1MDc5NjE0OTY5MTY4NjkwMTk4ODc3ODc2Njk4OTIyMTM3MTY3MTE3NjU1Mjk
0MjAwOTgwMzAxMDc2MjI4OTU5MDI1MTg0MjAyODU3NzgzMTQ3MzM1ODQwNDk2NzcyNzI1ODU5MzEzMTQ1
MTMyOTQyMjAzMjUwMzY2MjQyMTg3MTQxMTk2MTA5ODU2NDAyNTU0NzYxOTUyOTAwOTI0Mjk5NTU1MzU1M
zc0MTE5OTAxNjQyMTkxOTQwNTU3NTAwMDM3MTQ5NzgzODIzOTg0NjM0Njg5MTcyMjYwOTIyNzY1NzIxNj
Q0NzUyMjM1NDU0MzQ0NTAyMzQ1NTEzMTY1MzMwMTA1MjQ4NDM0MDI5NDcyMDIwMTU5MjAwODg1NzIzODE
wNTQwOTI3MDUyMzQ3Njc4ODg0MDI2MzY2NTQzOTIxNzk3MDI5OTEyODc4Nzc5NTE1ODY4OTExMzQxNDM5
NTg4ODgyOTk0NzIxOTM2NTgwODgwNzE1NjY0NTc2MDg4MjUzOTkyMzM4MTk5MDkwNjI2ODk4MDMwNzYxM
DgyNDcxMjIzNzQ5MjM1MzkyNTc0MTYwMDA1MjE0ODE2MjA5MjkyNzg3NzU2MzU4Mzg3OTAwMDk0NzQ4OD
UyNDc5MzYzMzI2ODczMjUxNDQzNDg2NTk3NjQyMjk5NzgwMTc4NzI1NjA2NDM1NzQwMDk1MzA1NjM2ODY
4NjAwNjY5NzY5NDY4Mjg4MjU5MjExMzg2OTgyNDgzODgwNTA5MDUyNTE4MDY2ODkwODE4MDc1MTExODQx
NTQ0Njg4MDY0NDE5MDA3Mjk4ODQwNDE5NDM3NiJdLCBbImNvbXBhbnkiLCAiNTA4NzIxODkxMjI5ODA1M
jQ2NTczMjkzOTQwNjE4Mjk2OTg2ODE4MzQ1NTY4NDI3MDcyNjI2NTU3MTUwODMzMTM1NDU2Mjk2MTYxNT
YzMDQyMzA1ODUwMDc1MzE5Mzg0Mzc4MDQxNTAzMzU5MjI1MDk0NTA0OTQ2MzQ3MzMzMzk0MzU1MjQ0MDU
4MzA3NTQ0MTgwNDk0NDAwNTgwMzAyNDAwMjEzNjM3NDE1NTE5ODc0MjQzMzA2OTQ1OTQzMzQyOTUwNjE1
NjMxMjY5MTgxMjcxODA2ODQ0MjA1NjM3MzE1NTg2OTk5Mjg4ODM2NTc5MjM5MzM2MzQ1Mzc2NjA1ODUyN
jc2MTQ2MjYyMzQ5MjQwMzgwMzQ1NTM4Nzc0ODM4NDU2OTc1ODI1ODczMTQxMzU3NTkxMTI1MTc3Njk2ND
c0Nzg2ODU4NTMzNTM3MzcyMTk5MzczODI1OTYyMzMxMTgxMzI0Mjk2MDM4OTEwMzM1NDQ3MzIyMDU3ODI
5OTA3MDc0MzcwMjc5OTEyMjMzMTU0MDQ1MDUyNTQyODA5MzIzNjgyMTQyMzAyNjM2OTkyMDIxOTUyODc4
NjQzMTYxMzUxMTUxMjA0MjA0NzEwNzQ5MTUwMDU3MzY4MjAxNzQyNjEwOTI1ODQwNjMxMTU0MzA1NjI1N
```

zM1MzY2OTQxODQwNjQxODYyMTU1Mzc5Mzg3Njg0Mzc4MjU0MTg4NDY1NzQ1OTI4MjI2OTM1NzgwNDMyMD
AwOTMwNDIzNjk5ODAxNzc3NjQwMjczNDc2MDA2MjI4MzM2OTU0MTkxNzYyNjc5MTQyMDk3Nzc0Mzk5MTc
1MzE4NjI0ODc1ODE0MTQ1NTkxODQzOTU1NDgwMjc3MzM2NjMzMTQxNzA0MTQ5NzUwNDc5OTc4NjYxNDcz
MjUzODYwMjQzIl0sIFsiaXNzdWVkIiwgIjYxNDY3OTk5MTc3MTQzMDU5MTU4MTg5MzA5MzE4NjA0NjY2M
zE2OTIyNDE2Nzk0MTY4NzE0NDkyODEzNzE0OTQzNjE4NzA0MzQxMzY1NDM1NjQzNzc0OTQyODYyMTgxNT
AzMzMzNDU3NzM4NTY5NDE4Nzk4NTc1NjI4NjIyODU4NDM3MTMyMTAxMjg3OTgyODUyMzY3NzEzNjU2Mzk
zODE2OTE3MTk2NDExMTY2NDE0NzMzMzU1OTQ0NDUxOTU4NTYyODM0NTg2Njc0NTU0MDU3MDgzNDg2MzUz
NjExODk4Njg3MDk4MDEzOTU2OTc0MzQyNjI1OTg3NTA0NDE2ODI1NzgzOTQ5NDQyNjAwMjQ0NzQyMDE2N
zAwNjE0NzA1OTE2ODE1NzA1MDIwMTcxMzEwMzczNzYzNzk4Mzk2MDgxNDA3MzQzODMyNzI4NzgyMTkwMD
cyOTAzMjg1ODEzMjgzMTQwMTIwOTU5MDMwMzYxMDAxMDE3NDU5NzMzNzU4NjkyMTAyNjE1MjMzNzg2MDc
4NzIwOTU2ODEyNzU3MjY4OTg3NjMyMTYwOTg4Nzg2NDQ0MTYwNzMyODcyNzI1OTQxNTM3MzQ2ODA5NTAw
MTIyMzg1ODE4NjI0NzU4MDcwNjcwOTUyMDE3NTA5NDY4Mjk3MjQzNjQyNzY4NTM3MzgxNjMyMTcyMzYzO
DU2NTE4NzA2NzI5MDY5MzU2MTE3NzU3ODA2OTc5NDQ4OTA5NTYwOTQ0NTg2MTE3NzAyNjIxMjYxNDUwMD
M1OTI4NjU0OTY0ODEwMDU1OTE5MTE3OTE3ODU3NjQzOTQ1NDgyMzY1MjY4MzY5OTIyNDU3OTk5MzQ1NTU
4MDQxMDY5MzU4Njg2Nzg1MjUxNDUxOTQ5NDgzODc0MDE0NzQxOTY2MDkxMTI4ODYxIl0sIFsiYnJhbmQi
LCAiMTgwOTEzNjA5MTU1NTI1MjU5NzcxMTcwOTM1OTg2NDI5MDY0NTE0OTM2MTU5Mzk4NDQxOTY1MjY2O
TQ0MTMwNDI0MjQ4ODcyODA4NTIwMTAxNDI1Nzk3MjIyMzMwMDc4OTc1MDk2MzkzNzk3ODIxMzQ5MDgzND
Q5MjQ5OTU4NzQ0NjgyOTA1MjYxMDQxNDEzMzg3MjYwMzUzNzE5NTE2MjAzMDA4ODY1NjQzOTA5NDk3MzI
xNTMzNjM2ODAyNjkwMDY0NDU0NDc3NzIwMDk3MzYzNjUxMDM4Nzg2OTQ3OTYxODcxNTg4OTU3MTU5MDc4
MjgwMzE0MzIzNzkwNzQ5MzEzMTE5ODI3NzM3MTQ2NDA3Nzg2MjMxODc2MTczNDIyMDAyMTk1NjI5NjkwO
TEwNDQ2NDY2MTIyNDA2NzI0NzczMzczMTA1NzMxNDI4NTQxMTQwMjc5NjA3MTMyOTg5OTc4MDE5MjkzNj
gxOTA2MjQ1MzY4NjA3NjQ5NjE4ODg3NjMwODAwMjg0OTcwNTkyMzE1MzA5OTY5NDY0Njg3MjY3MzUxMDM
yMzgyNzA2MzY4ODU0NzAzMTY5OTUyNjc4Njg3Njk0NzU3NTM2ODM5NzA4OTE4Nzk3NzQ4NDAwMzM5MTc5
NDYzNjUyNzM5ODY3ODQzMTEwMTE0NTA3MzE5MzEyNzUwNzYxMTc1Mjc0NjI3OTc5OTAxNDQ1NjkyMDU3N
TgyNjU0NzA3NTc2MTM5ODQ1MzUwMTk5MTAxNzc4MzIwMDY4NzM4NzQ4NDUzNDMwNTk5MDk4MzM5MDE2OT
I3NzY2Mjc2NzI2OTMxMzEwODc2MTM5Njk5MTc2MDYxNTAyMzY5NTA2NTk2MzY1MTgyOTQ2MjQ5NDgwMTE
5MjM3OTIyMDA5NTEwMDg0NTc1ODcxMDgyMDc5Il0sIFsidHlwZSIsICI4MDg2NzA5NjIwODk4ODAxODIz
NDkyOTY5NTA1MjkxNjMyMzUyODc1NDAwODQ3NTM1NzUwNDU4Mjk1Nzk5MjIwNTk2NDgxNjg5NDYwMjA0N
Dk4NTQzMDkxMzQwODEyNDM3NDQyNTc2Nzc2MjA5MDgxMDE1MjE3NTcyODY3MDI3MDI5NDgwMDc4MjI4ND
Y4OTAxODczMjQzMzU2MTk5NTkyMjkzODkwOTM2OTczNDY2OTI3MDgxMzEwMDA5ODQ4MjA1ODA5MjEzNzY
5NjY1MTgyMzgxNzcxODMwNDk4MTIwNzI3NjY0MTE0OTE5Mzk1MDU1MzkzNzMwMDA1MDUyMTc4MjMzODAw
NDk5MDc0MzQzNTczMjcyNjg1ODI2Mjg5MDExNDkzOTA1NzcwOTM3NTY3NTgzNjg5NjMzMTQ5MjgwODY5O
DQyNjY1NDMzOTI3NTUwNTc0Mzc0ODE4Njg3NjgxNjE0MjI2MDY2MTA0NjQ5MjQyNTI2NjczMTAwMTExOT
c2ODAwMzEzOTExNDEyMzExNjY5NzY3OTM1NjI1MTIwMTA4NDM1MTU3OTAyMTAzMDE4MDEyODk0MTE1MzI
2MjI3MTEzNjM2NzQ4ODYzMzc2MjkxODAwNzgwNDA3NzY3ODU2MzgyNzMyMTUxNDE1ODEyNjYzMzAxMDMw
MDMzODAwMjI4MDE2ODY0MjgyMTQxMDYzMTAxODQzNTMzNDM0ODQwNDI2MzE4MTIyNDMzMDQ4MzQ0MjMwM
DMwMDgxMjEzNzczODA2MjAxMzI3NjYxMTY1NDMyNDk4NTIyOTY1NzU0MDk3Mzg4ODE5MjY5NjA3MjA3OT
kzOTMxNzM4NzU3MDg4NjUwMDkyMDY4NDIyNzgwMTI4MTc5NzkxNDkxODQ5MTU4NzU1OTY4MzE0MjQ2OTY
xNjM0OTciXSwgWyJtYXN0ZXJfc2VjcmV0IiwgIjYwMzA0MzQ0NzgyNTU4ODI5Njc4ODQzMDU1MzUzMTYw
OTM1NDQwMjg4MTU3ODk0MDU0MjAwNjE2ODg0NDExOTMwMTM1MzI0MTQwODkxNjg1NzczOTIyMDM2MjI0N
zU2MjA4MzU2NTc2NTQxNzE0ODE1Nzk1MDM5OTQ4Mjg1OTcwMjY1NjE2MDkzOTIzODA4MTU4MjQ2NjM3OT
A4NDMxNzIwOTAwNjQ1MDY5MTE1NjYzNDM0MzA3NzI1MTc5NjczMDM3NTI3NDU2MjM5NzM2NjMzMjkxMzc
4OTYyMzEzNDM2NzI2MDU1MDAyNTYyMTc3MzczODI1ODc3NDUwNjk3MjYyMDkxMTU1Mjg2MjU2ODA1NDU1
ODM0MjY3MjU2NzY0NDA2MDcxOTA0NzIyNDE2OTY1ODk1MjE0ODM1NDg5MTE4ODQ4NjkzNjAyNjY4NTg3M
jYxMzU1ODc4NjkzMzc4NjAzMTQ5MDk2MTQyMjk3MzY4NzgzNDE1NjI1NDI1MDM3OTI5OTA2NzExMjE1ND
M4MjQxOTk4MTY5MDk4Njg5MDk4MDgyNzkwMzA2Mjk0NTQ4MzU2OTA5MDA3NDI4MTgzMzQxMDI4NDE5NDM
4NzAwODE2MTA1NTc3MjkwMTY2MDY4MDc1NTI5NDQ5NTQwNDM4MjkxNzAxODM2NjA1NTc1MzI1MTAxMTA2
Njc5ODk5OTIyNTgxMDk1NTY4ODM2MzYxNzA5NDc5NzgwMjA1OTgwNDUyNjYyODAwODcxMDUyNTQ2NzgzN
zg5NDMxMTcwODc0MTg0MTczMDQxMjg5MDA5MjM0ODY2NjcxMzY2NTcxMzE2MTQzMjk1MDEwODExMTYxMT
YwNzI2MDQzNjU0MDYyMzA5Mjk3NTcxNDQ5MjI3NDQzNjA4MzYxNjMzMTQyOTIwOTgxODU0NiJdLCBbImx
pY2Vuc2UiLCAiNjg0MDU1MjYzNzIwOTIzMjk3MzkyNzIzOTcxNjQ0MTM5NDgyNzA2MjkzODc3NjI5OTYy
Mzg1NTIzNTk1MTQxMDg3MTYzMTA1MzMwMjQ4NDk2Njc3ODEyODY3NDk2Mjg3ODIwNzUxMzY3MzE4MDE4N
Tc5NzgxMTE1ODE5NDkxMTM5ODY4NjQ0NjAyMjk2NTk2ODgxMjAxNjE1MTI1MTYxMDk0MTk0ODg4MTMwMz
E3ODQwMDk0MDI1Nzc4ODM0OTIzNDQ4NTcwNDY5ODMzNTczMTM1ODUwMzYyODIxMDYwMTM1Mjk4OTcwNDc
xNDYzMzMwMzM0MTU1NjEyNzQ0MDEyMzUyMjMzNjYyMDY2MTIxMjQ4NDkyNjI4NTU3MjE4NTgwMTkyNTA0
NDQ0MzE0NjkxOTAzMzA3MjgxOTUxMDc0Mzk3NjM2MDMwMjc4NzExNDUxMTY4MDAzMzc0NzEwNzA5MjQ4O
TQwMDYyNDAwNjI5NDUyODU3OTQ4OTk0MDk5MDM4MjQxNzQ4NjgxMDY5OTkyMTY5MDk4NTU1NjU4OTQzND
QxMjQwOTk3ODg3MTgxMzA3MTkzODQ2MTEwMDYyOTU4MTQxNDI3MTE1MzM3NTUyMTcyMjkzODAyMTA2MzA
4NTIxNzYyMDY0Mzc0NDgzNTQwMzAzMDQ4NjEzMTQzNTI0MTc2NTUyMTM5MjEwNzQ1MjA3NzY3NTA5MTIz
NjUxNjczOTAzNTk5MjI4NDY2MzA3NzQ4NzAyMjY5ODEzNDY3MjA4Nzg4NTU2NDU0MDE1NTA3MzYzNzgyM
jI1ODAzODIxMzgzNjAzMzU5OTU3NjAzMDIwMTIxMDE5ODMwNjg1MzAyMTY1NDI3NjU2NzIwNTc5Mjg1MD
UzMzQ1NTk2MTg2OTA3NTcwNzAyNzUxMTQ4MDE1MDMzMTM1Il0sIFsiaGFzaCIsICI0NDkzMTE5NTQzMzE
3MDUyNzM4MjI0Mzc0ODA2NTE2MDc1OTIzNDk4NzIwNzYwMDc5NTcwMjc2MTM1NjU1MTExMzQwNzA5MTU3

```
OTcwNDA3NjIyNTYyNTMwMjU4MTM3NDcyMjUyNDI0MzkxMDE3NjQ5ODA5NjEwOTcyNzIxMTY0MTczNzIxM
DEwMTQ4NjQyODYxOTc1MTUwNzg3NDgzOTA2NDA5MjA0MzU2MzcxODM1NjYxMDYyODYyNTU3MTQ5ODY5OD
QxMDQwODQ0MTcxMTA3NTg0NzEwNzgwNDI5NDI4MDg3NDIxNjMyMjkzNjIxMDg2NzAyNjgzMjYyMzM3MDk
5NDYxNTQwNTE1NTcyODY2NzYyNTc1NTI2NDA4MjU3NDIwNzI5MDAwMzU2NzU4OTY3NDUwNTczMTUxNTc2
MTAxNjkxMzQ5MDY1NDgxNTgxNzUwMDcwODI1NzI0MDAzODc3NTg1NjQ3MzgxNzU1NTg2NTY3ODcxODQxN
zcwNTI4ODU0NTI4MDQxODU4OTA0MzI2NzMzMjQyNjY3OTQ5NDk3MDE5NjY0NzEyNTA1NTA0NzIwMjg5OD
k1NDIxMzMzOTA2ODI1NjY0OTAzNjkzODc1MTc1NzU5MDM0OTE1MDU2NDk1MjMyMzAwNzUyNzkwOTIxMDA
0ODM0MTQ4MjM5MTY3OTcxMDMyODk1MTc0ODkyMTkwNjk5OTE4NjcwNzI2NjQ4NTkxMDM1MDk2NTEzNjIy
ODk2NDY5NTAwOTQxNDU2Mjg2MjY2MDkyNDkxMjE2OTQ4OTMwMTkxODAxMDEzNDc3NzM2MjI4MDU3NjY2O
DU4NjEwOTc1MjY3MDY5NDgyMDg4MTk5MjQ1MDIyNDY3NDQyOTU0MjM2NzgwMTU3NjI3MDg1MjE4MzU2NT
g5NjI5MTI2ODczMzYiXV19LCAibm9uY2UiOiAiNzI0ODY1MTYzMDYyNzM1MDgwMDUxNzAyIn0="
                }
            }
        ]
    },
    "by_format": {
        "cred_proposal": {
            "indy": {
                "cred_def_id": "XrhUg2JxHVn1e5mwo7kYtr:3:CL:175:Teabag-3.0",
                "issuer_did": "XrhUg2JxHVn1e5mwo7kYtr",
                "schema_id": "XrhUg2JxHVn1e5mwo7kYtr:2:Teabag:3.0",
                "schema_issuer_did": "XrhUg2JxHVn1e5mwo7kYtr",
                "schema_name": "Teabag",
                "schema_version": "3.0"
            }
        },
        "cred_offer": {
            "indy": {
                "schema_id": "XrhUg2JxHVn1e5mwo7kYtr:2:Teabag:3.0",
                "cred_def_id": "XrhUg2JxHVn1e5mwo7kYtr:3:CL:175:Teabag-3.0",
                "key_correctness_proof": {
                    "c": "505830379101192876590599828401903677595889378045327791491662762322137302609267",
                    "xz_cap": "895319192927073985102622809796909453049191766551474209415134981936536624654655769835540958905716999709598747629298946653655382942928019303944694987348335488002548605235438593953085125029422865845389133490568515692256284482443698041352338263305258790694330462481428384344896532823465640688740491427300925787665285868696270518320457462599779433982958949274864947134199363786648280850073933904824627524222767991256325319251266435843429410307502529529437142097821749345805853729369849795302515558842082022097213184437369061002511060425919357709639140968829425017850579027522925829722507829212524014897630123637027903142685597412496765806581191572574706329384269459717888061430118246113188125535024",
                    "xr_cap": [
                        [
                            "url",
                            "995788960317308115047024312694903770987362749827780954848742352338302919507961496916869019887787669892213716711765529420098030107622895902518420285778314733584049677272585931314513294220325036624218714119610985640255476195290092429955535537411990164219194055750003714978382398463468917226092276572164475223545434450234551316533010524843402947202015920088572381054092705234767888402636654392179702991287877951586891134143958888299472193658088071566457608825399233819909062689803076108247122374923539257416000521481620929278775635838790009474885247936332687325144348659764229978017872560643574009530563686860066976946828825921138698248388050905251806689081807511184154468806441900729884041943760"
                        ],
                        [
                            "company",
                            "508721891229805246573293940618296986818345568427072626557150833135456296161563042305850075319384378041503359225094504946347333394355244058307544180494400580302400213637415519874243306945943342950615631269181271806844205637315586999288836579239336345376605852676146262349240380345538774838456975825873141357591125177696474786858533537372199373825962331181324296038910335447322057829907074370279912233154045052542809323682142302636992021952878643161351151204 2
```

```
047107491500573682017426109258406311543056257353669418406418621553793876843782541
884657459282269357804320009304236998017776402734760062283369541917626791420977743
991753186248758141455918439554802773366331417041497504799786614732538602 43"
                        ],
                        [
                            "issued",
                            "614679991771430591581893093186046663169224167941687 1
449281371494361870434136543564377494286218150333345773856941879857562862285843713
210128798285236771365639381691719641116641473335594445195856283458667455405708348
635361189868709801395697434262598750441682578394944260024474201670061470591681570
502017131037376379839608140734383272878219007290328581328314012095903036100101745
973375869210261523378607872095681275726898763216098878644416073287272594153734680
950012238581862475807067095201750946829724364276853738163217236385651870672906935
611775780697944890956094458611770262126145003592865496481005591911791785764394548
236526836992245799934555804106935868678525145194948387401474196609112886 1"
                        ],
                        [
                            "brand",
                            "180913609155525259771170935986429064514936159398441 9
652669441304242488728085201014257972223300789750963937978213490834492499587446829
052610414133872603537195162030088656439094973215336368026900644544777200973636510
387869479618715889571590782803143237907493131198277371464077862318761734220021956
296909104464661224067247733731057314285411402796071329899780192936819062453686076
496188876308002849705923153099694646872673510323827063688547031699526786876947575
368397089187977484003391794636527398678431101145073193127507611752746279799014456
920575826547075761398453501991017783200687387484534305990983390169277662767269313
108761396991760615023695065963651829462494801192379220095100845758710820 79"
                        ],
                        [
                            "type",
                            "808670962089880182349296950529163235287540084753575 0
458295799220596481689460204498543091340812437442576776209081015217572867027029480
078228468901873243356199592293890936973466927081310009848205809213769665182381771
830498120727664114919395055393730005052178233800499074343573272685826289011493905
770937567583689633149280869842665433927550574374818687681614226066104649242526673
100111976800313911412311669767935625120108435157902103018012894115326227113636748
863376291800780407767856382732151415812663301030033800228016864282141063101843533
434840426318122433048344230030081213773806201327661165432498522965754097388819269
607207993931738757088650092068422780128179791491849158755968314246961634 97"
                        ],
                        [
                            "master_secret",
                            "603043447825588296788430553531609354402881578940542 0
061688441193013532414089168577392203622475620835657654171481579503994828597026561
609392380815824663790843172090064506911566343430772517967303752745623973663329137
896231343672605500256217737382587745069726209115528625680545583426725676440607190
472241696589521483548911884869360266858726135587869337860314909614229736878341562
542503792990671121543824199816909868909808279030629454835690900742818334102841943
870081610557729016606807552944954043829170183660557532510110667989992258109556883
636170947978020598045266280087105254678378943117087418417304128900923486667136657
131614329501081116116072604365406230929757144922744360836163314292098185 46"
                        ],
                        [
                            "license",
                            "684055263720923297392723971644139482706293877629962 3
855235951410871631053302484966778128674962878207513673180185797811158194911398686
446022965968812016151251610941948881303178400940257788349234485704698335731358503
628210601352989704714633303341556127440123522336620661212484926285572185801925044
443146919033072819510743976360302787114511680033747107092489400624006294528579489
940990382417486810699921690985556589434412409978871813071938461100629581414271153
375521722938021063085217620643744835403030486131435241765521392107452077675091236
516739035992284663077487022698134672087885564540155073637822258038213836033599576
030201210198306853021654276567205792850533455961869075707027511480150331 35"
```

```
                        ],
                        [
                            "hash",
                            "44931195433170527382243748065160759234987207600795702761356551113407091579704076225625302581374722524243910176498096109727211641737210101486428619751507874839064092043563718356610628625571498698410408441711075847107804294280874216322936210867026832623370994615405155728667625755264082574207290003567589674505731515761016913490654815817500708257240038775856473817555865678718417705288545280418589043267332426679494970196647125055047202898954213339068256649036938751757590349150564952323007527909210048341482391679710328951748921906999186707266485910350965136228964695009414562862660924912169489301918010134777362280576668586109752670694820881992450224674429542367801576270852183565896291268733 6"
                        ]
                    ]
                },
                "nonce": "72486516306273508005 1702"
            }
        }
    },
    "auto_offer": false,
    "auto_issue": true,
    "auto_remove": true
}
```


Listing B.15 — Issuing a certificate response

The holder `Alice` can now obtain the Smart Certificate from the own wallet

```
curl -X 'GET' \
  'https://holder.ogc.secd.eu/credentials' \
  -H 'accept: application/json' \
  -H 'X-API-KEY: <Holder API-Key>'
```

Listing B.16 — Alice lists all Smart Certificates from the wallet


```
{
    "results": [
        {
            "referent": "03f56ec2-f73b-4661-b78a-4098cf11a191",
            "schema_id": "XrhUg2JxHVn1e5mwo7kYtr:2:Teabag:3.0",
            "cred_def_id": "XrhUg2JxHVn1e5mwo7kYtr:3:CL:175:Teabag-3.0",
            "rev_reg_id": "XrhUg2JxHVn1e5mwo7kYtr:4:XrhUg2JxHVn1e5mwo7kYtr:3:
CL:175:Teabag-3.0:CL_ACCUM:6cf7a24f-9d79-4792-a7e6-8f5a9f361485",
            "cred_rev_id": "1",
            "attrs": {
                "type": "Classic",
                "issued": "Mon Aug 19 10:34:45 CEST 2024",
                "hash": "sha256=998845b59db786d3df28ba85b42c6a481da2b3843399ae5c3
f49a06c1e564f87",
                "company": "Lipton",
                "url": "https://iafs.demo.secure-dimensions.de/f/998845b59db786d3
df28ba85b42c6a481da2b3843399ae5c3f49a06c1e564f87",
                "brand": "Earl Grey",
                "license": "CC-BY 4.0"
            }
        }
    ]
}
```


Listing B.17 — Certificates from the wallet response

B.4.5. Step IV: Verifying a Smart Certificate

Finally, the holder `Alice` can use the Smart Certificate to execute the Trusted Teapot Process. The following interactions simulate the verification steps manually.

First, the verifier `Teapot` needs a peer-to-peer connection with the holder `Alice`. As `Alice` is the active part, this starts with creating a connection invitation.

```
curl -X 'POST' \
  'https://holder.ogc.secd.eu/out-of-band/create-invitation?auto_accept=
true&create_unique_did=true' \
  -H 'accept: application/json' \
  -H 'X-API-KEY: <Holder API-Key>' \
  -H 'Content-Type: application/json' \
  -d '{
  "accept": [
    "didcomm/aip1",
    "didcomm/aip2;env=rfc19"
  ],
  "alias": "Teabag",
  "goal": "To have a Teabag Smart Certificate verified",
  "goal_code": "issue-vc",
  "handshake_protocols": [
    "https://didcomm.org/didexchange/1.0"
  ],
  "metadata": {},
  "my_label": "Invitation to Teapot",
  "protocol_version": "1.1",
  "use_did_method": "did:peer:2",
  "use_public_did": false
}'
```

Listing B.18 — Holder creates a connection invitation


```
{
    "state": "initial",
    "trace": false,
    "invi_msg_id": "00f43256-75f8-46c3-8953-c2aa5e21517c",
    "oob_id": "d2d93cf4-9b4e-433e-afac-4c1d7b86ac5c",
    "invitation": {
        "@type": "https://didcomm.org/out-of-band/1.1/invitation",
        "@id": "00f43256-75f8-46c3-8953-c2aa5e21517c",
        "label": "Invitation to Issuer",
        "handshake_protocols": [
            "https://didcomm.org/didexchange/1.0"
        ],
        "accept": [
            "didcomm/aip1",
            "didcomm/aip2;env=rfc19"
        ],
        "services": [
            "did:peer:2.Vz6Mku1tgKm4epmxp81YZ44iAH29BjEqNNRnABnHW3MhvWdCT.Ez6LSnK
Jr9nSYkYS3HSXx9bBXF94kjU8pSqqbjvSumLU8nwR9.SeyJpZCI6IiNkaWRjb21tLTAiLCJ0IjoiZGlkL
WNvbW11bmljYXRpb24iLCJwcmlvcml0eSI6MCwicmVjaXBpZW50S2V5cyI6WyIja2V5LTEiXSwiciI6W1
0sInMiOiJodHRwOi8vMTkyLjE2OC4xLjEwOjgwMzAifQ"
        ]
    },
    "invitation_url": "http://192.168.1.10:8030?oob=eyJAdHlwZSI6ICJodHRwczovL2RpZ
GNvbW0ub3JnL291dC1vZi1iYW5kLzEuMS9pbnZpdGF0aW9uIiwgIkBpZCI6ICIwMGY0MzI1Ni03NWY4LT
```

```
Q2YzMtODk1My1jMmFhNWUyMTUxN2MiLCAibGFiZWwiOiAiSW52aXRhdGlvbiB0byBJc3N1ZXIiLCAiaGF
uZHNoYWtlX3Byb3RvY29scyI6IFsiaHR0cHM6Ly9kaWRjb21tLm9yZy9kaWRleGNoYW5nZS8xLjAiXSwg
ImFjY2VwdCI6IFsiZGlkY29tbS9haXAxIiwgImRpZGNvbW0vYWlwMjtlbnY9cmZjMTkiXSwgInNlcnZpY
2VzIjogWyJkaWQ6cGVlcjoyLlZ6Nk1rdTF0Z0ttNGVwbXhwODFZWjQ0aUFIMjlCakVxTk5SbkFCbkhXM0
1odldkQ1QuRXo2TFNuS0pyOW5TWWtZUzNIU1h4OWJCWEY5NGtqVThwU3FxYmp2U3VtTFU4bndSOS5TZXl
KcFpDSTZJaU5rYVdSamIyMXRMVEFpTENKMElqb2laR2xrTFdOdmJXMTFibWxqWVhScGIyNGlMQ0p3Y21s
dmNtbDBlU0k2TUN3aWNtVmphWEJwWlc1MFMyVjVjeUk2V3lJamEyVjVMVEVpWFN3aWNpSTZXMTBzSW5Na
U9pSm9kSFJ3T2k4dk1Ua3lMakUyT0M0eExqRXdPamd3TXpBaWZRIl19"
}
```


Listing B.19 — Holder's connection invitation response

To finalize the connection, the verifier `Teapot` accepts the invitation.

```
curl -X 'POST' \
  'https://verifier.ogc.secd.eu/out-of-band/receive-invitation?alias=Holder' \
  -H 'accept: application/json' \
  -H 'X-API-KEY: <Verifier API-Key>' \
  -H 'Content-Type: application/json' \
  -d '{
        "@type": "https://didcomm.org/out-of-band/1.1/invitation",
        "@id": "00f43256-75f8-46c3-8953-c2aa5e21517c",
        "label": "Invitation to Issuer",
        "handshake_protocols": [
            "https://didcomm.org/didexchange/1.0"
        ],
        "accept": [
            "didcomm/aip1",
            "didcomm/aip2;env=rfc19"
        ],
        "services": [
            "did:peer:2.Vz6Mku1tgKm4epmxp81YZ44iAH29BjEqNNRnABnHW3MhvWdCT.Ez6LSnK
Jr9nSYkYS3HSXx9bBXF94kjU8pSqqbjvSumLU8nwR9.SeyJpZCI6IiNkaWRjb21tLTAiLCJ0IjoiZGlkL
WNvbW11bmljYXRpb24iLCJwcmlvcml0eSI6MCwicmVjaXBpZW50S2V5cyI6WyIja2V5LTEiXSwiciI6W1
0sInMiOiJodHRwOi8vMTkyLjE2OC4xLjEwOjgwMzAifQ"
        ]
    }'
```

Listing B.20 — Teapot accept the connection invitation


```
{
    "state": "deleted",
    "created_at": "2024-08-19T09:30:06.553408Z",
    "updated_at": "2024-08-19T09:30:06.553408Z",
    "trace": false,
    "oob_id": "dae26972-1ce9-476a-bb5a-0dcc65308ac8",
    "invi_msg_id": "00f43256-75f8-46c3-8953-c2aa5e21517c",
    "invitation": {
        "@type": "https://didcomm.org/out-of-band/1.1/invitation",
        "@id": "00f43256-75f8-46c3-8953-c2aa5e21517c",
        "label": "Invitation to Issuer",
        "handshake_protocols": [
            "https://didcomm.org/didexchange/1.0"
        ],
        "accept": [
            "didcomm/aip1",
            "didcomm/aip2;env=rfc19"
        ],
        "services": [
            "did:peer:2.Vz6Mku1tgKm4epmxp81YZ44iAH29BjEqNNRnABnHW3MhvWdCT.Ez6LSnK
Jr9nSYkYS3HSXx9bBXF94kjU8pSqqbjvSumLU8nwR9.SeyJpZCI6IiNkaWRjb21tLTAiLCJ0IjoiZGlkL
WNvbW11bmljYXRpb24iLCJwcmlvcml0eSI6MCwicmVjaXBpZW50S2V5cyI6WyIja2V5LTEiXSwiciI6W1
0sInMiOiJodHRwOi8vMTkyLjE2OC4xLjEwOjgwMzAifQ"
```

```
        ]
    },
    "connection_id": "fbea1839-f202-40ad-a0a3-6d96804c4e9c",
    "role": "receiver",
    "multi_use": false
}
```


Listing B.21 — Verifier's connection acceptance response

Connection between `Verifier` and `Holder`

- Verifier→Holder: connection_id: `fbea1839-f202-40ad-a0a3-6d96804c4e9c`

Connection between `Holder` and `Verifier` must be searched for in the holder's wallet.

```
curl -X 'GET' \
  'https://holder.ogc.secd.eu/connections?invitation_msg_id=00f43256-75f8-46c3-
8953-c2aa5e21517c&limit=100&offset=0' \
  -H 'accept: application/json' \
  -H 'X-API-KEY: <Holder API-Key>'
```

Listing B.22 — Holder searches for the connection and status


```
{
    "results": [
        {
            "state": "active",
            "created_at": "2024-08-19T09:29:39.119392Z",
            "updated_at": "2024-08-19T09:30:06.851913Z",
            "connection_id": "fd60d0b5-31d5-4d0b-86ef-18fcc1c869b9",
            "my_did": "xrrD4ZQ5dv8pzCCECErE7",
            "their_did": "XXBgiCcVWvQpH6QVTx4bLy",
            "their_label": "Verifier.agent",
            "their_role": "invitee",
            "connection_protocol": "didexchange/1.0",
            "rfc23_state": "completed",
            "invitation_key": "FZddjWpDVEUM1WhrNVkKRvbBufZWxYXoVmNaD5jubQR5",
            "invitation_msg_id": "00f43256-75f8-46c3-8953-c2aa5e21517c",
            "request_id": "bb0b703d-2a92-469f-b716-058b934361cf",
            "accept": "auto",
            "invitation_mode": "once",
            "alias": "Lipton"
        }
    ]
}
```


Listing B.23 — Holders's connections response

This indicates the following active connection: * Holder→Verifier: connection_id: `fd60d0b5-31d5-4d0b-86ef-18fcc1c869b9`

The verifier `Teapot` can finally request proof that the holder `Alice` is in possession of a Smart Certificate for the image stored at IAFS URL https://iafs.demo.secure-dimensions.de/f/998845b59db786d3df28ba85b42c6a481da2b3843399ae5c3f49a06c1e564f87

The image hash is used to link the Smart Certificate with the image that is stored at an external provider. Because the SHA-256 algorithm was used when issuing the Smart Certificate, the proof request must use the same hash algorithm.

The Trusted Teapot Process also asks the holder to disclose additional attributes from the Smart Certificate as they are required for processing. These attributes are `company`, `brand` and `type`.

```
curl --location 'https://verifier.ogc.secd.eu/present-proof/send-request' \
--header 'Content-Type: application/json' \
--header 'X-API-Key: <Verifier API-Key>' \
--data '{
    "connection_id": "fbea1839-f202-40ad-a0a3-6d96804c4e9c",
    "auto_remove": false,
    "auto_verify": true,
    "proof_request": {
        "name": "Teabag",
        "version": "3.0",
        "requested_attributes": {
            "hash": {
                "name": "hash",
                "restrictions": [
                    {
                        "cred_def_id": "XrhUg2JxHVn1e5mwo7kYtr:3:CL:175:Teabag-
3.0",
                        "attr::hash::value": "sha256=998845b59db786d3df28ba85b42c
6a481da2b3843399ae5c3f49a06c1e564f87"
                    }
                ]
            },
            "brand": {
                "name": "brand",
                "restrictions": []
            },
            "company": {
                "name": "company",
                "restrictions": []
            },
            "type": {
                "name": "type",
                "restrictions": []
            }
        },
        "requested_predicates": {}
    }
}'
```

Listing B.24 — Verifier sends proof request to Holder

```
{
    "state": "request_sent",
    "created_at": "2024-08-19T09:32:23.355452Z",
    "updated_at": "2024-08-19T09:32:23.355452Z",
    "trace": false,
    "presentation_exchange_id": "2d0feb52-d627-4582-8b7c-91d4c705bc33",
    "connection_id": "fbea1839-f202-40ad-a0a3-6d96804c4e9c",
    "thread_id": "8bc115b4-abca-4487-bccf-d05d833f3ff2",
    "initiator": "self",
    "role": "verifier",
    "presentation_request": {
        "nonce": "387713795381214155782152",
        "name": "Teabag",
        "version": "3.0",
        "requested_attributes": {
            "hash": {
                "name": "hash",
                "restrictions": [
                    {
```

```
                        "cred_def_id": "XrhUg2JxHVn1e5mwo7kYtr:3:CL:175:Teabag-
3.0",
                        "attr::hash::value": "sha256=998845b59db786d3df28ba85b42c
6a481da2b3843399ae5c3f49a06c1e564f87"
                    }
                ]
            },
            "brand": {
                "name": "brand",
                "restrictions": []
            },
            "company": {
                "name": "company",
                "restrictions": []
            },
            "type": {
                "name": "type",
                "restrictions": []
            }
        },
        "requested_predicates": {}
    },
    "presentation_request_dict": {
        "@type": "https://didcomm.org/present-proof/1.0/request-presentation",
        "@id": "8bc115b4-abca-4487-bccf-d05d833f3ff2",
        "request_presentations~attach": [
            {
                "@id": "libindy-request-presentation-0",
                "mime-type": "application/json",
                "data": {
                    "base64": "eyJuYW1lIjogIlRlYWJhZyIsICJ2ZXJzaW9uIjogIjMuMCIsIC
JyZXF1ZXN0ZWRfYXR0cmlidXRlcyI6IHsiaGFzaCI6IHsibmFtZSI6ICJoYXNoIiwgInJlc3RyaWN0aW9
ucyI6IFt7ImNyZWRfZGVmX2lkIjogIlhyaFVnMkp4SFZuMWU1bXdvN2tZdHI6MzpDTDoxNzU6VGVhYmFn
LTMuMCIsICJhdHRyOjpoYXNoOjp2YWx1ZSI6ICJzaGEyNTY9OTk4ODQ1YjU5ZGI3ODZkM2RmMjhiYTg1Y
jQyYzZhNDgxZGEyYjM4NDMzOTlhZTVjM2Y0OWEwNmMxZTU2NGY4NyJ9XX0sICJicmFuZCI6IHsibmFtZS
I6ICJicmFuZCIsICJyZXN0cmljdGlvbnMiOiBbXX0sICJjb21wYW55IjogeyJuYW1lIjogImNvbXBhbnk
iLCAicmVzdHJpY3Rpb25zIjogW119LCAidHlwZSI6IHsibmFtZSI6ICJ0eXBlIiwgInJlc3RyaWN0aW9u
cyI6IFtdfX0sICJyZXF1ZXN0ZWRfcHJlZGljYXRlcyI6IHt9LCAibm9uY2UiOiAiMzg3NzEzNzk1MzgxM
jE0MTU1NzgyMTUyIn0="
                }
            }
        ]
    },
    "auto_present": false,
    "auto_verify": true,
    "auto_remove": false
}
```


Listing B.25 — Proof Request response

Once the request has been received by the holder's agent, the attempted to auto-verify the proof request is started. The result is either a positive proof response or an exception message in case that no auto-proof is possible.

The verifier `Teabag` can check the proof status by searching the wallet's proofs based on the `presentation_exchange_id`.

```
curl -X 'GET' \
  'https://verifier.ogc.secd.eu/present-proof/records/2d0feb52-d627-4582-8b7c-
91d4c705bc33' \
  -H 'accept: application/json' \
```

```
  -H 'X-API-KEY: <Verifier API-Key>'
```

Listing B.26 — Verifier searches wallet for the status


```
{
  "state": "verified",
  "created_at": "2024-08-19T09:32:23.355452Z",
  "updated_at": "2024-08-19T09:32:23.756088Z",
  "trace": false,
  "presentation_exchange_id": "2d0feb52-d627-4582-8b7c-91d4c705bc33",
  "connection_id": "fbea1839-f202-40ad-a0a3-6d96804c4e9c",
  "thread_id": "8bc115b4-abca-4487-bccf-d05d833f3ff2",
  "initiator": "self",
  "role": "verifier",
  "presentation_request": {
    "nonce": "387713795381214155782152",
    "name": "Teabag",
    "version": "3.0",
    "requested_attributes": {
      "hash": {
        "name": "hash",
        "restrictions": [
          {
            "cred_def_id": "XrhUg2JxHVn1e5mwo7kYtr:3:CL:175:Teabag-3.0",
            "attr::hash::value": "sha256=998845b59db786d3df28ba85b42c6a481da2b384
3399ae5c3f49a06c1e564f87"
          }
        ]
      },
      "brand": {
        "name": "brand",
        "restrictions": []
      },
      "company": {
        "name": "company",
        "restrictions": []
      },
      "type": {
        "name": "type",
        "restrictions": []
      }
    },
    "requested_predicates": {}
  },
  "presentation_request_dict": {
    "@type": "https://didcomm.org/present-proof/1.0/request-presentation",
    "@id": "8bc115b4-abca-4487-bccf-d05d833f3ff2",
    "request_presentations~attach": [
      {
        "@id": "libindy-request-presentation-0",
        "mime-type": "application/json",
        "data": {
          "base64": "eyJuYW1lIjogIlRlYWJhZyIsICJ2ZXJzaW9uIjogIjMuMCIsICJyZXF1ZXN0
ZWRfYXR0cmlidXRlcyI6IHsiaGFzaCI6IHsibmFtZSI6ICJoYXNoIiwgInJlc3RyaWN0aW9ucyI6IFt7I
mNyZWRfZGVmX2lkIjogIlhyaFVnMkp4SFZuMWU1bXdvN2tZdHI6MzpDTDoxNzU6VGVhYmFnLTMuMCIsIC
JhdHRyOjpoYXNoOjp2YWx1ZSI6ICJzaGEyNTY9OTk4ODQ1YjU5ZGI3ODZkM2RmMjhiYTg1YjQyYzZhNDg
xZGEyYjM4NDMzOTlhZTVjM2Y0OWEwNmMxZTU2NGY4NyJ9XX0sICJicmFuZCI6IHsibmFtZSI6ICJicmFu
ZCIsICJyZXN0cmljdGlvbnMiOiBbXX0sICJjb21wYW55IjogeyJuYW1lIjogImNvbXBhbnkiLCAicmVzd
HJpY3Rpb25zIjogW119LCAidHlwZSI6IHsibmFtZSI6ICJ0eXBlIiwgInJlc3RyaWN0aW9ucyI6IFtdfX
0sICJyZXF1ZXN0ZWRfcHJlZGljYXRlcyI6IHt9LCAibm9uY2UiOiAiMzg3NzEzNzk1MzgxMjE0MTU1Nzg
yMTUyIn0="
        }
      }
```

```
    ]
  },
  "presentation": {
    "proof": {
      "proofs": [
        {
          "primary_proof": {
            "eq_proof": {
              "revealed_attrs": {
                "brand": "811648150937182330231127524387529542165859965915634805522321629292642853967799",
                "company": "13367127887689907803703433490445133802506253177320659009082545591321827153332",
                "hash": "97205814620398442380070821315995013409374460451206276838547095689552290864939",
                "type": "78769287913063788254918834259019030160534368657209252279629795053278321624233"
              },
              "a_prime": "10058907944478034388163112977649960851342433435788469325012725742759747308348428507993850748346839735581082236594170354462805387125364948950730250088926125124927891649627221525552550705560777467670291082532261563640501502183050289090492293733961148504887398786801695215770900293714362830175859993306807387426791148812927811813113602404799312135786497574408745551277366047121391369324839915374856623837639739001633572445602371495663951016852614565005192507275630211964718591967045720238762943673177253879372519242937501139274221892441103446570861012829484424389477894621696430198823537169910294754400639694516459829711",
              "e": "3812642408741370763966467308188618990555991106707075807184445133641478819468986052097425973031855670968049434335306973357391790523203261 9",
              "v": "9946686984463392390432322710631656548741549754767329266247291435157980947806296140925907470831455230243992932999788693216264635377655733400068196834205677596438399059211220397677638029684556220076733663964937267311773615666123573742208729807539394043973291554791063785028553692288620591136063448365094228513569805521669439216239075934408477211997564642352784045291934187146747636542105471296141392785115160542075272228037197306225696151133677962740213685893678788029509309824807811101163886321676934291298418310236703264608768036778690228773073144248703922130931464436317875277457589397276892168369844383395402807108438759927259419362056694021586023082213257843092796704986021138302195225901453803763995062952932312211224526693803485456171101852271274178543865460437606086338796257503282139809713606651197444724599327196048147403421163970344166926999552734696269989625109186062436225774005283579293079500168722405856630 01",
              "m": {
                "url": "515316599757575828517151748145959051311385337936683845532260740502941916847390617045736124398237609458956592868068410723525753140045760223471143133298661931131137441279422538848 1",
                "issued": "551060190159461459095922239192005930825046050064254837060716547086328332700166163039074967288329344202100368889377537743751826718629446959865127779664886017527804775065006043928 0",
                "license": "3707551283680131698641160030503119249255852926827927082959225849824252194904097498513348119604682725687012171196277663827752233353648741705995043519567125554348671005012357674 27",
                "master_secret": "2809220823069227329180235054527728569362423352941234420728055764657134562948017286693825594081176292338292481093194542655205675518451315280987799685876420439154324221273473898882"
              },
              "m2": "4589058904215654001399965821498136036607844682935236508906159922547394000909695921339323002194613077644096440927131684512909954731484984983768587656459313240243169076522363769516857645211792287172903491253969339631395698248745584501081457877511264132842772229525132176160115180021519082142398456243904194744882123510121338420725889678005710676660218627854247149724794896808806915322201546288286280122137137628246882125526155164365010474009295928686954023169444980477356940379751435958702563642503169940886901930187917883669451148538807062402433272979832139062478013955237005413333223862225191877405730429231106087667230087247 92
```

```
835825634614819816991895621911800755784544445524854654030390893241223394430471942
314213973233256589146698"
            },
            "ge_proofs": []
          }
        }
      ],
      "aggregated_proof": {
        "c_hash": "102721855043054376590530863218649285679148214711933784773750971986753235501612",
        "c_list": [
          [
            79,
            174,
            144,
            249,
            54,
            40,
            33,
            35,
            230,
            79,
            62,
            227,
            26,
            34,
            182,
            106,
            110,
            0,
            76,
            247,
            3,
            4,
            1,
            126,
            191,
            103,
            60,
            66,
            143,
            40,
            116,
            0,
            238,
            99,
            67,
            163,
            60,
            87,
            62,
            141,
            191,
            238,
            12,
            219,
            249,
            136,
            132,
            251,
            163,
            106,
            151,
```

```
12,
62,
214,
158,
44,
103,
172,
15,
82,
226,
213,
213,
61,
32,
84,
137,
218,
65,
44,
22,
199,
85,
164,
64,
30,
101,
148,
104,
158,
215,
163,
231,
98,
254,
24,
68,
233,
48,
225,
219,
199,
207,
116,
56,
167,
194,
185,
243,
170,
6,
33,
27,
158,
47,
63,
200,
70,
254,
136,
153,
63,
216,
101,
```

```
57,
11,
201,
250,
184,
126,
71,
12,
207,
255,
37,
228,
34,
196,
53,
223,
255,
34,
246,
23,
174,
159,
145,
95,
92,
157,
19,
9,
96,
221,
43,
242,
16,
129,
89,
162,
244,
78,
237,
133,
115,
67,
181,
153,
94,
162,
199,
51,
246,
175,
32,
137,
23,
29,
94,
99,
116,
165,
128,
97,
227,
181,
140,
```

```
246,
178,
174,
92,
39,
162,
229,
182,
49,
131,
38,
234,
169,
129,
186,
74,
83,
17,
216,
159,
42,
117,
100,
129,
164,
175,
89,
108,
47,
9,
47,
245,
9,
147,
125,
217,
50,
147,
68,
56,
38,
71,
11,
247,
81,
49,
247,
185,
7,
196,
176,
78,
45,
122,
160,
159,
125,
26,
171,
185,
23,
138,
31,
```

```
            73,
            17,
            210,
            85,
            91,
            200,
            97,
            131,
            14,
            42,
            100,
            187,
            23,
            218,
            121,
            207
          ]
        ]
      }
    },
    "requested_proof": {
      "revealed_attrs": {
        "company": {
          "raw": "Lipton",
          "encoded": "133671278876899078037034334904451338025062531773206590090825455591321827153332",
          "sub_proof_index": 0
        },
        "hash": {
          "raw": "sha256=998845b59db786d3df28ba85b42c6a481da2b3843399ae5c3f49a06c1e564f87",
          "encoded": "97205814620398442380070821315995013409374460451206276838547095689552290864939",
          "sub_proof_index": 0
        },
        "type": {
          "raw": "Classic",
          "encoded": "78769287913063788254918834259019030160534368657209252279629795053278321624233",
          "sub_proof_index": 0
        },
        "brand": {
          "raw": "Earl Grey",
          "encoded": "81164815093718233023112752438752954216585996591563480552232162929264285396799",
          "sub_proof_index": 0
        }
      },
      "self_attested_attrs": {},
      "unrevealed_attrs": {},
      "predicates": {}
    },
    "identifiers": [
      {
        "schema_id": "XrhUg2JxHVn1e5mwo7kYtr:2:Teabag:3.0",
        "cred_def_id": "XrhUg2JxHVn1e5mwo7kYtr:3:CL:175:Teabag-3.0",
        "rev_reg_id": "XrhUg2JxHVn1e5mwo7kYtr:4:XrhUg2JxHVn1e5mwo7kYtr:3:CL:175:Teabag-3.0:CL_ACCUM:6cf7a24f-9d79-4792-a7e6-8f5a9f361485"
      }
    ]
  },
  "verified": "true",
  "verified_msgs": [],
```

```
  "auto_present": false,
  "auto_verify": true,
  "auto_remove": false
}
```

Listing B.27 — Proof request response

The above indicates that the request was auto-verified which means that Alice did prove to the Teapot Process to be in possession of a Smart Certificate that is associated with the image SHA-256 hash `998845b59db786d3df28ba85b42c6a481da2b3843399ae5c3f49a06c1e564f87`.

To assert that the auto-verification does actually refuse a proof request for another image, a proof request was sent for another image: https://iafs.demo.secure-dimensions.de/f/b278ee930b22baec09f977a625c1d51919575c36184e06d61bc74c952c243da0

Contrary to the first image which shows the Lipton Earl Grey classic teabag, the latter image shows the Lipton Earl Grey lemon teabag. And naturally, the hash of the images differ!

```
curl --location 'https://verifier.ogc.secd.eu/present-proof/send-request' \
--header 'Content-Type: application/json' \
--header 'X-API-Key: <Verifier API-Key>' \
--data '{
    "connection_id": "fbea1839-f202-40ad-a0a3-6d96804c4e9c",
    "auto_remove": false,
    "auto_verify": true,
    "proof_request": {
        "name": "Teabag",
        "version": "3.0",
        "requested_attributes": {
            "hash": {
                "name": "hash",
                "restrictions": [
                    {
                        "cred_def_id": "XrhUg2JxHVn1e5mwo7kYtr:3:CL:175:Teabag-
3.0",
                        "attr::hash::value": "sha256=b278ee930b22baec09f977a625c1
d51919575c36184e06d61bc74c952c243da0"
                    }
                ]
            },
            "brand": {
                "name": "brand",
                "restrictions": []
            },
            "company": {
                "name": "company",
                "restrictions": []
            },
            "type": {
                "name": "type",
                "restrictions": []
            }
        },
        "requested_predicates": {}
    }
}'
```

Listing B.28 — Verifier sends proof request to Holder

From the response (not shown for readability) we use the `presentation_exchange_id` to query the proof from the wallet.

```
curl -X 'GET' \
  'https://verifier.ogc.secd.eu/present-proof/records/480dbaa3-ea56-40a2-b27b-
8476be203034' \
  -H 'accept: application/json' \
  -H 'X-API-KEY: <Verifier API-Key>'
```

Listing B.29 — Verifier searches wallet for the status


```
{
    "state": "request_sent",
    "created_at": "2024-08-19T09:49:52.781330Z",
    "updated_at": "2024-08-19T09:49:52.781330Z",
    "trace": false,
    "presentation_exchange_id": "480dbaa3-ea56-40a2-b27b-8476be203034",
    "connection_id": "fbea1839-f202-40ad-a0a3-6d96804c4e9c",
    "thread_id": "43f70240-4b9e-4f9e-9ab7-3baeb9208ab1",
    "initiator": "self",
    "role": "verifier",
    "presentation_request": {
        "nonce": "10706826616321917410827 66",
        "name": "Teabag",
        "version": "3.0",
        "requested_attributes": {
            "hash": {
                "name": "hash",
                "restrictions": [
                    {
                        "cred_def_id": "XrhUg2JxHVn1e5mwo7kYtr:3:CL:175:Teabag-
3.0",
                        "attr::hash::value": "sha256=b278ee930b22baec09f977a625c1
d51919575c36184e06d61bc74c952c243da0"
                    }
                ]
            },
            "brand": {
                "name": "brand",
                "restrictions": []
            },
            "company": {
                "name": "company",
                "restrictions": []
            },
            "type": {
                "name": "type",
                "restrictions": []
            }
        },
        "requested_predicates": {}
    },
    "presentation_request_dict": {
        "@type": "https://didcomm.org/present-proof/1.0/request-presentation",
        "@id": "43f70240-4b9e-4f9e-9ab7-3baeb9208ab1",
        "request_presentations~attach": [
            {
                "@id": "libindy-request-presentation-0",
                "mime-type": "application/json",
                "data": {
                    "base64": "eyJuYW1lIjogIlRlYWJhZyIsICJ2ZXJzaW9uIjogIjMuMCIsIC
JyZXF1ZXN0ZWRfYXR0cmlidXRlcyI6IHsiaGFzaCI6IHsibmFtZSI6ICJoYXNoIiwgInJlc3RyaWN0aW9
ucyI6IFt7ImNyZWRfZGVmX2lkIjogIlhyaFVnMkp4SFZuMWU1bXdvN2tZdHI6MzpDTDoxNzU6VGVhYmFn
```

```
LTMuMCIsICJhdHRyOjpoYXNoOjp2YWx1ZSI6ICJzaGEyNTY9YjI3OGVlOTMwYjIyYmFlYzA5Zjk3N2E2M
jVjMWQ1MTkxOTU3NWMzNjE4NGUwNmQ2MWJjNzRjOTUyYzI0M2RhMCJ9XX0sICJicmFuZCI6IHsibmFtZS
I6ICJicmFuZCIsICJyZXN0cmljdGlvbnMiOiBbXX0sICJjb21wYW55IjogeyJuYW1lIjogImNvbXBhbnk
iLCAicmVzdHJpY3Rpb25zIjogW119LCAidHlwZSI6IHsibmFtZSI6ICJ0eXBlIiwgInJlc3RyaWN0aW9u
cyI6IFtdfX0sICJyZXF1ZXN0ZWRfcHJlZGljYXRlcyI6IHt9LCAibm9uY2UiOiAiMTA3MDY4MjY2MTYzM
jE5MTc0MTA4Mjc2NiJ9"
                    }
                }
            ]
        },
        "auto_present": false,
        "auto_verify": true,
        "auto_remove": false
}
```


Listing B.30 — Proof request response

The proof request remains in the `request_sent` state. This illustrates the correctness of the auto-verification for this example.

B.5. Trusted Watermarking Process Execution Example

The following Trusted Watermarking Process execution uses a VC from the Spacebel catalogue using this URL: https://emc.spacebel.be/collections/Aqua_AMSR-E_L3_SSW_1month_0.25deg/items/P1AME020600A_P3SSW000700E0?httpAccept=application/vc%2Bld%2Bjson

```
curl -X 'POST' \
  'link:++https://processes.ogc.secd.eu/processes/trusted-watermarking/execution+
+[]' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "inputs": {
    "url": "link:++https://emc.spacebel.be/collections/Aqua_AMSR-E_L3_SSW_1month_
0.25deg/items/P1AME020600A_P3SSW000700E0?httpAccept=application/vc%2Bld%2Bjson+
+[]"
  }
}'
```

Listing B.31

The following is the QUICKLOOK image extracted from the VC:

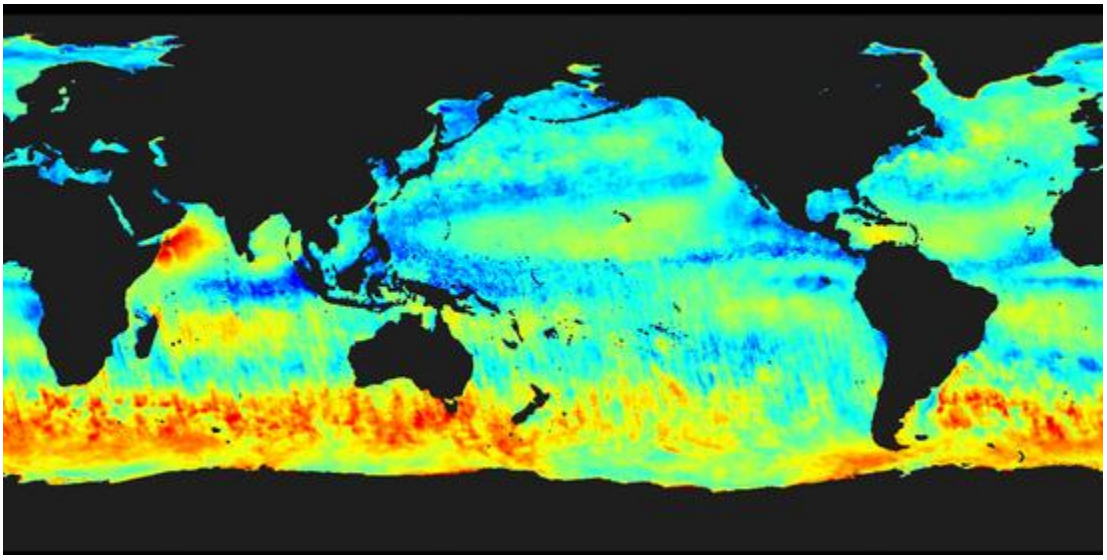


Figure B.19 — QUICKLOOK Image

The following is the watermarked output generated by the process:

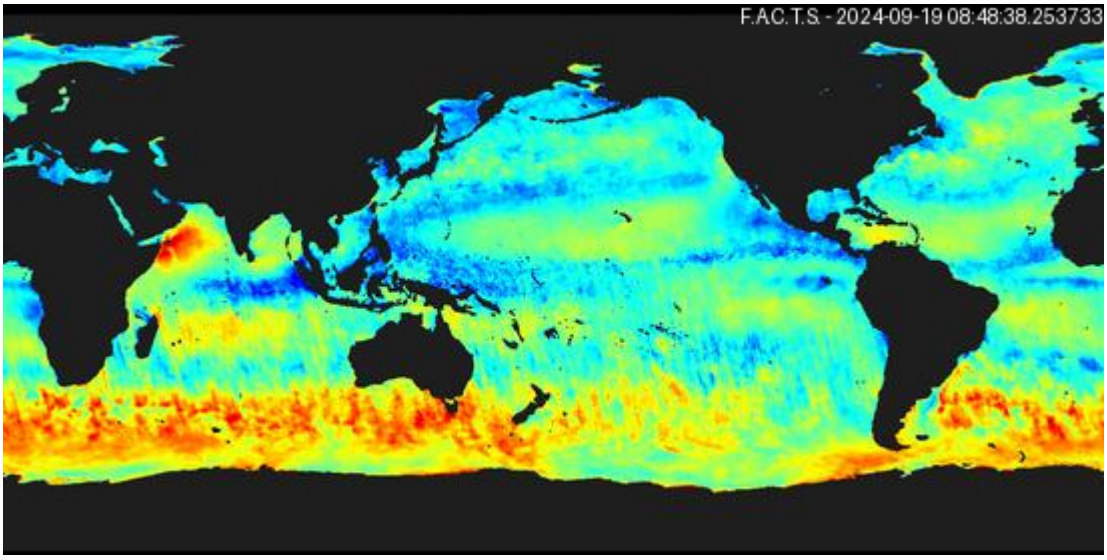


Figure B.20 — Watermarked QUICKLOOK Image

C

ANNEX C (INFORMATIVE) SPACEBEL — IPT SERVER ARCHITECTURE COMPONENT EXAMPLES

ANNEX C (INFORMATIVE) SPACEBEL – IPT SERVER ARCHITECTURE COMPONENT EXAMPLES

C.1. OGC/STAC Catalog Response Example

Below an extract of the additional information for an EO product metadata record in a Catalog response providing access to the DID, VC and VP.


```
    {
      "rel": "alternate",
      "href": "https://emc.spacebel.be/collections/TropForest/items/KO2_OTPF_KO2_
MSC_2F_20090831T125932_20090831T125932_016509_W038_S006?httpAccept=application/vc
%2Bld%2Bjson",
      "type": "application/vc+ld+json",
      "title": "Verifiable Credential"
    },
    {
      "rel": "alternate",
      "href": "https://emc.spacebel.be/collections/TropForest/items/KO2_OTPF_KO2_
MSC_2F_20090831T125932_20090831T125932_016509_W038_S006?httpAccept=application/vp
%2Bld%2Bjson",
      "type": "application/vp+ld+json",
      "title": "Verifiable Presentation"
    },
    {
      "rel": "describes",
      "href": "did:web:emc.spacebel.be:collections:TropForest:items:KO2_OTPF_KO2_
MSC_2F_20090831T125932_20090831T125932_016509_W038_S006",
      "type": "application/did+json",
      "title": "DID"
    }
```


Listing C.1

C.2. W3C Decentralized Identifier Document Example


```
{
  "@context": [
    "https://www.w3.org/ns/did/v1",
```

```
    "https://w3id.org/security/suites/jws-2020/v1"
  ],
  "id": "did:web:emc.spacebel.be:organisations:esa_esrin",
  "alsoKnownAs": [
    "https://gcmd.earthdata.nasa.gov/kms/concept/c56b4a86-82f8-4f15-98ba-
c5f7abe8ee5a",
    "https://yago-knowledge.org/resource/European_Space_Agency",
    "https://dbpedia.org/resource/European_Space_Agency",
    "https://ror.org/03wd9za21"
  ],
  "verificationMethod": [
    {
      "id": "did:web:emc.spacebel.be:organisations:esa_esrin#owner",
      "type": "JsonWebKey2020",
      "controller": "did:web:emc.spacebel.be:organisations:esa_esrin",
      "publicKeyJwk": {
        "kty": "EC",
        "crv": "secp256k1",
        "x": "yKemFmBtShtFrYfZHj9D3h83FntzLcCbLm8n1O4VOyI",
        "y": "ssJjEkBMsd4OFejeKNiCZRhJYuTaITyEGz93Ti0gfqw"
      }
    }
  ],
  "assertionMethod": [
    "did:web:emc.spacebel.be:organisations:esa_esrin#owner"
  ]
}
```


Listing C.2

C.3. W3C Verifiable Credentials Example


```
{
  "credentialSubject": {
    "date": "2009-08-31T12:59:32Z/2009-08-31T12:59:32Z",
    "bbox": [
      -38.09186965,
      -6.09027783,
      -37.91002331,
      -5.91282213
    ],
    "geometry": {
      "coordinates": [[
        [
          -38.09186965,
          -5.91282213
        ],
        [
          -38.09186965,
          -6.09027783
        ],
        [
          -37.91002331,
          -6.09027783
        ],
        [
          -37.91002331,
          -5.91282213
```

```
        ],
        [
          -38.09186965,
          -5.91282213
        ]
      ]],
      "type": "Polygon"
    },
    "links": {
      "data": [{
        "href": "https://tpm-ds.eo.esa.int/oads/data/Tropforest/KO2_OTPF_KO2_MSC_
2F_20090831T125932_20090831T125932_016509_W038_S006.ZIP",
        "title": "Download",
        "type": "application/x-binary"
      }],
      "previews": [
        {
          "href": "http://tpm-ds.eo.esa.int/oads/meta/Tropforest/browse/KO2_OTPF_
KO2_MSC_2F_20090831T125932_20090831T125932_016509_W038_S006.ZIP_BID.JPG?hl=zQmS4B
GNFENx911i6FwS4yAoJGbFb1TfutUNNSEyEEEznTe",
          "title": "QUICKLOOK",
          "type": "image/jpeg"
        },
        {
          "href": "http://tpm-ds.eo.esa.int/oads/meta/Tropforest/thumbnail/KO2_
OTPF_KO2_MSC_2F_20090831T125932_20090831T125932_016509_W038_S006.ZIP_TIMG.jpg?hl=
zQmRfPerwQvNuKL7D2rfx4cA6E5dGJtfNcHV1AvjAJwHTgy",
          "title": "THUMBNAIL",
          "type": "image/jpeg"
        }
      ]
    },
    "id": "did:web:emc.spacebel.be:collections:TropForest:items:KO2_OTPF_KO2_MSC_
2F_20090831T125932_20090831T125932_016509_W038_S006",
    "title": "KO2_OTPF_KO2_MSC_2F_20090831T125932_20090831T125932_016509_W038_
S006",
    "updated": "2024-04-30T11:32:32Z"
  },
  "issuanceDate": "2024-09-20T14:05:29Z",
  "id": "did:web:emc.spacebel.be:collections:TropForest:items:KO2_OTPF_KO2_MSC_
2F_20090831T125932_20090831T125932_016509_W038_S006",
  "proof": {
    "created": "2024-09-20T14:05:29.665Z",
    "jws": "eyJhbGciOiJFUzI1NksiLCJraWQiOiJiQnpXOHlHeEt1RUNMal9JeFNyTkN1eVJKOUppZ
2ptTU9DeDJGZHpDcnFBIiwiY3JpdCI6WyJiNjQiXSwiYjY0IjpmYWxzZX0..X2hsUtb6LdPfLLushOAMp
UbkkEVI0TrKM2gaUqmkZfBMyY7uumTGPthIEKtrpuV0ULE0v-FDaMiU_8ssU1-V5w",
    "proofPurpose": "assertionMethod",
    "type": "EcdsaSecp256k1Signature2019",
    "verificationMethod": "did:web:emc.spacebel.be:organisations:esa_esrin#owner"
  },
  "type": [
    "VerifiableCredential",
    "Feature"
  ],
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    {
      "date": "dct:date",
      "asWKT": "gsp:asWKT",
      "gj": "https://purl.org/geojson/vocab#",
      "data": "iana:enclosure",
      "gsp": "http://www.opengis.net/ont/geosparql#",
      "bbox": {
```

```
        "@id": "gj:bbox",
        "@container": "@list"
      },
      "coordinates": "gj:coordinates",
      "icon": "iana:icon",
      "title": "dct:title",
      "Feature": "gj:Feature",
      "hasGeometry": "gsp:hasGeometry",
      "dct": "http://purl.org/dc/terms/",
      "previews": "iana:icon",
      "iana": "http://www.iana.org/assignments/relation/",
      "geometry": "gj:geometry",
      "links": {
        "@id": "owc:links",
        "@context": {
          "@vocab": "http://www.iana.org/assignments/relation/",
          "type": "atom:type"
        }
      },
      "owc": "http://www.opengis.net/ont/owc/1.0/",
      "href": "@id",
      "id": "@id",
      "Polygon": "gj:Polygon",
      "atom": "http://www.w3.org/2005/Atom/",
      "updated": "dct:modified"
    }
  ],
  "issuer": "did:web:emc.spacebel.be:organisations:esa_esrin"
}
```


Listing C.3

C.4. W3C Verifiable Presentations Example


```
{
  "holder": "did:web:emc.spacebel.be:organisations:ceos",
  "proof": {
    "created": "2024-09-20T14:07:12.567Z",
    "jws": "eyJhbGciOiJFZERTQSIsImNyaXQiOlsiYjY0Il0sImI2NCI6ZmFsc2V9..i7CMONFjaCb
Lby60ZImTUX2IvLkPnFtPCe5BiVisiiIm4Yw62wbH0PtGCTETqa4tatctPfBkBVTPA7cM_TV8AA",
    "proofPurpose": "authentication",
    "type": "Ed25519Signature2018",
    "verificationMethod": "did:web:emc.spacebel.be:organisations:ceos#owner"
  },
  "type": ["VerifiablePresentation"],
  "@context": ["https://www.w3.org/2018/credentials/v1"],
  "verifiableCredential": [{
    "credentialSubject": {
      "date": "2009-08-31T12:59:32Z/2009-08-31T12:59:32Z",
      "bbox": [
        -38.09186965,
        -6.09027783,
        -37.91002331,
        -5.91282213
      ],
      "links": {
        "data": [{
```

```
          "href": "https://tpm-ds.eo.esa.int/oads/data/Tropforest/KO2_OTPF_KO2_
MSC_2F_20090831T125932_20090831T125932_016509_W038_S006.ZIP",
          "title": "Download",
          "type": "application/x-binary"
        }],
        "previews": [
          {
            "href": "http://tpm-ds.eo.esa.int/oads/meta/Tropforest/browse/KO2_
OTPF_KO2_MSC_2F_20090831T125932_20090831T125932_016509_W038_S006.ZIP_BID.JPG?hl=z
QmS4BGNFENx911i6FwS4yAoJGbFb1TfutUNNSEyEEEznTe",
            "title": "QUICKLOOK",
            "type": "image/jpeg"
          },
          {
            "href": "http://tpm-ds.eo.esa.int/oads/meta/Tropforest/thumbnail/KO2_
OTPF_KO2_MSC_2F_20090831T125932_20090831T125932_016509_W038_S006.ZIP_TIMG.jpg?hl=
zQmRfPerwQvNuKL7D2rfx4cA6E5dGJtfNcHV1AvjAJwHTgy",
            "title": "THUMBNAIL",
            "type": "image/jpeg"
          }
        ]
      },
      "geometry": {
        "coordinates": [[
          [
            -38.09186965,
            -5.91282213
          ],
          [
            -38.09186965,
            -6.09027783
          ],
          [
            -37.91002331,
            -6.09027783
          ],
          [
            -37.91002331,
            -5.91282213
          ],
          [
            -38.09186965,
            -5.91282213
          ]
        ]],
        "type": "Polygon"
      },
      "id": "did:web:emc.spacebel.be:collections:TropForest:items:KO2_OTPF_KO2_
MSC_2F_20090831T125932_20090831T125932_016509_W038_S006",
      "title": "KO2_OTPF_KO2_MSC_2F_20090831T125932_20090831T125932_016509_W038_
S006",
      "updated": "2024-04-30T11:32:32Z"
    },
    "issuanceDate": "2024-09-20T14:07:12Z",
    "id": "did:web:emc.spacebel.be:collections:TropForest:items:KO2_OTPF_KO2_MSC_
2F_20090831T125932_20090831T125932_016509_W038_S006",
    "proof": {
      "created": "2024-09-20T14:07:12.482Z",
      "jws": "eyJhbGciOiJFUzI1NksiLCJraWQiOiJiQnpXOHlHeEt1RUNMal9JeFNyTkN1eVJKOUp
pZ2ptTU9DeDJGZHpDcnFBIiwiY3JpdCI6WyJiNjQiXSwiYjY0IjpmYWxzZX0..NbAmO9Sx56_9AWa3Ngd
bKjW296tYwq4Hfroyi0bfRrR20FJNj7f07o4AIFfgll8P9ojo7veA1bdEdKgOptBDqg",
      "proofPurpose": "assertionMethod",
      "type": "EcdsaSecp256k1Signature2019",
```

```
      "verificationMethod": "did:web:emc.spacebel.be:organisations:esa_
esrin#owner"
    },
    "type": [
      "VerifiableCredential",
      "Feature"
    ],
    "@context": [
      "https://www.w3.org/2018/credentials/v1",
      {
        "date": "dct:date",
        "asWKT": "gsp:asWKT",
        "gj": "https://purl.org/geojson/vocab#",
        "data": "iana:enclosure",
        "gsp": "http://www.opengis.net/ont/geosparql#",
        "bbox": {
          "@id": "gj:bbox",
          "@container": "@list"
        },
        "coordinates": "gj:coordinates",
        "icon": "iana:icon",
        "title": "dct:title",
        "Feature": "gj:Feature",
        "hasGeometry": "gsp:hasGeometry",
        "dct": "http://purl.org/dc/terms/",
        "previews": "iana:icon",
        "geometry": "gj:geometry",
        "iana": "http://www.iana.org/assignments/relation/",
        "links": {
          "@id": "owc:links",
          "@context": {
            "@vocab": "http://www.iana.org/assignments/relation/",
            "type": "atom:type"
          }
        },
        "owc": "http://www.opengis.net/ont/owc/1.0/",
        "id": "@id",
        "href": "@id",
        "atom": "http://www.w3.org/2005/Atom/",
        "Polygon": "gj:Polygon",
        "updated": "dct:modified"
      }
    ],
    "issuer": "did:web:emc.spacebel.be:organisations:esa_esrin"
  }]
}
```


Listing C.4

C.5. Univerifier VC and VP Validation

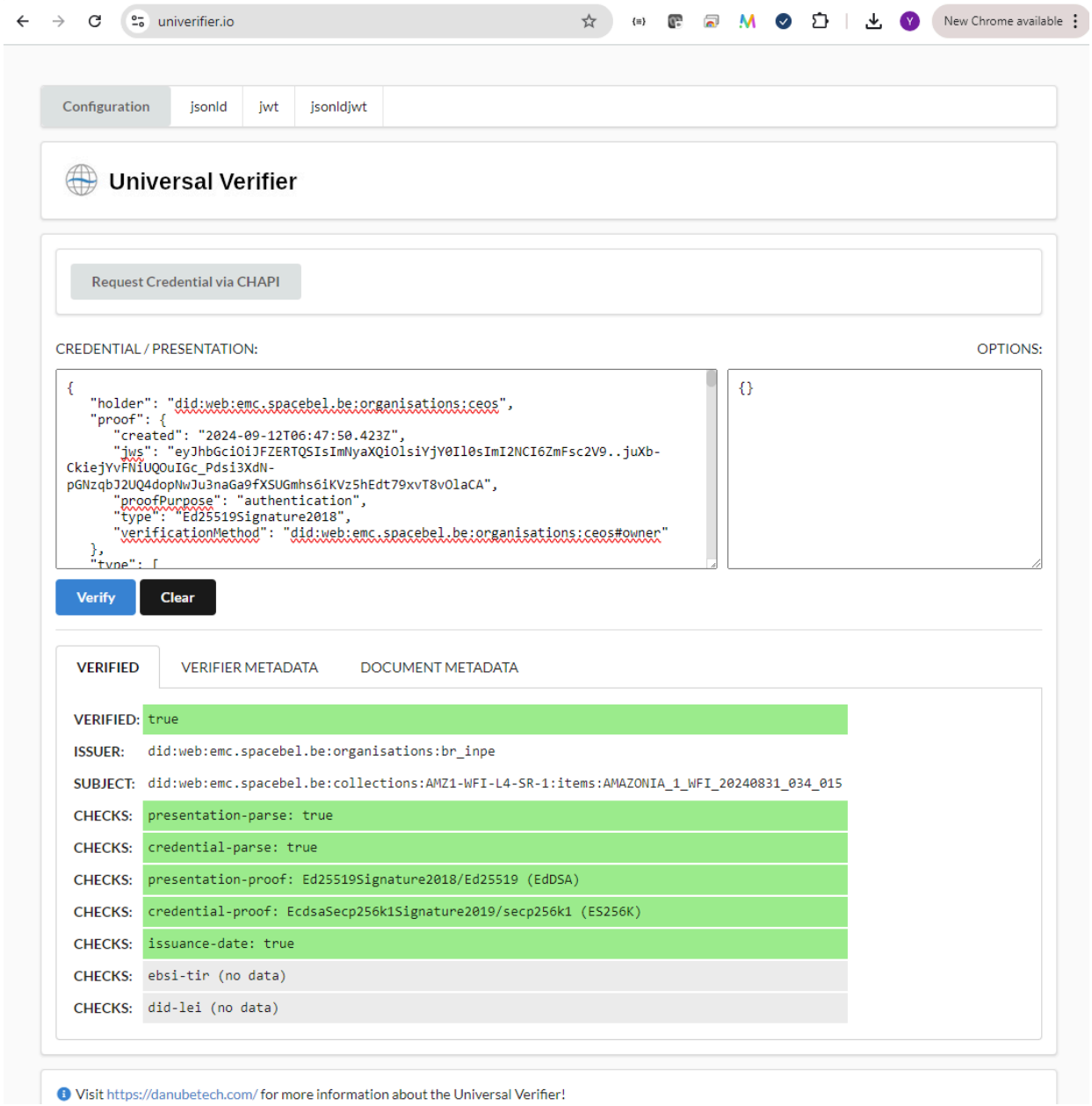


Figure C.1 — Univerifier

D

ANNEX D (INFORMATIVE) SPACEBEL — IPT SERVER DEMO NOTEBOOK

ANNEX D
(INFORMATIVE)
SPACEBEL — IPT SERVER DEMO NOTEBOOK

D.1. Introduction

D.1.1. Definitions

A DID resolver is a software and/or hardware component that performs the DID resolution function by taking a DID as input and producing a conforming DID document as output.

```
# First resolver is publicly accessible but cannot handle query parameters.
DID_RESOLVER_1 = "https://dev.uniresolver.io/1.0/identifiers/"

# The second DID resolver requires an access token and has access constraints
(calls per hour).
DID_RESOLVER_2 = "https://api.godiddy.com/0.1.0/universal-resolver/identifiers/"

DID_RESOLVER = DID_RESOLVER_1
```

Listing D.1

D.1.2. Issuers

The following issuers have been defined via W3C DIDs. For simplicity, Web DID was used, but Indy DID or EBSI DID could be used as well.

- did:web:emc.spacebel.be:organisations:ceos
- did:web:emc.spacebel.be:organisations:esa_esrin
- did:web:emc.spacebel.be:organisations:br_inpe
- did:web:emc.spacebel.be:organisations:spacebel_sa
- did:web:emc.spacebel.be:organisations:de_dlr

D.1.3. EO Resources

All EO products in the FedEO Catalogue have been assigned a (Web) DID. Below are a number of examples.

- did:web:emc.spacebel.be:collections:TropForest:items:KO2_OTPF_KO2_MSC_2F_20091107T04175(
- did:web:emc.spacebel.be:collections:AMZ1-WFI-L4-SR-1:items:AMAZONIA_1_WFI_20240618_037_016
- did:web:emc.spacebel.be:collections:F-FSM:items:F_FSM_20160610T000000_20161101T235959_ROCHEFORT

D.2. Create a DID

D.2.1. Create key pair for use with DID

This section shows how to prepare a set of keys (private/public) to be included in a DID document. The publication of the DID is handled off-line and differs for each DID method.

```
# OKP is used in W3C examples of VC and DID
# key = jwk.JWK.generate(kty='OKP', crv='Ed25519')
# k = key.export(private_key=True)
# k

key = jwk.JWK.generate(kty='EC', crv='secp256k1')
k = key.export(private_key=True, as_dict=True)
k
```

Listing D.2

```
{'kty': 'EC', 'crv': 'secp256k1', 'x':
'g3IKCADcDxoN8TTDsBpTqskrRwfa6nEIXfMC8eW3dF4', 'y':
'g0l7Wpqc1oTjeT129_1Ql2L3NO6-zrznCCIYYEzBCRc', 'd':
'N0NaJ1ZHT9Znfsd9qxmNA2aOnvlbBAbvSTWNMhiiFrA'}
```

Figure D.1

JWK Thumbprint is preferred key name, for consistency (See EBSI DID conventions).

D.3. Resolve DID

D.3.1. Resolve DID for an issuer (data provider)

For this demonstration the did:web method was used. An operational implementation could be based on did:indy or did:esbi DIDs. The resolver Spacebel used should accept any of these DID types.

```
DID_ISSUER[INDEX_ESA]
```

Listing D.3

```
'did:web:emc.spacebel.be:organisations:esa_esrin'
```

Figure D.2

```
response = requests.get( DID_RESOLVER + DID_ISSUER[INDEX_ESA],
    verify=True,
    headers={ 'Accept': 'application/json' })

data = json.loads(response.text)
jstr = json.dumps(data, indent=3)
md("```json\n" + jstr + "\n```\n")
```

Listing D.4


```
{
   "@context": [
      "https://www.w3.org/ns/did/v1",
      "https://w3id.org/security/suites/jws-2020/v1"
   ],
   "assertionMethod": [
      "did:web:emc.spacebel.be:organisations:esa_esrin#owner"
   ],
   "id": "did:web:emc.spacebel.be:organisations:esa_esrin",
   "verificationMethod": [
      {
         "id": "did:web:emc.spacebel.be:organisations:esa_esrin#owner",
         "type": "JsonWebKey2020",
         "controller": "did:web:emc.spacebel.be:organisations:esa_esrin",
         "publicKeyJwk": {
            "kty": "EC",
            "crv": "secp256k1",
            "x": "yKemFmBtShtFrYfZHj9D3h83FntzLcCbLm8n1O4VOyI",
            "y": "ssJjEkBMsd4OFejeKNiCZRhJYuTaITyEGz93Ti0gfqw"
         }
      }
   ],
   "alsoKnownAs": [
      "https://gcmd.earthdata.nasa.gov/kms/concept/c56b4a86-82f8-4f15-98ba-
c5f7abe8ee5a",
      "https://yago-knowledge.org/resource/European_Space_Agency",
      "https://dbpedia.org/resource/European_Space_Agency",
      "https://ror.org/03wd9za21"
   ]
```

```
}
```

Listing D.5

D.3.2. Resolve DID for an EO product

Example: 10
> Resolve a DID or dereference a DID URL (JSON) for a resource with the following DID.

```
DID_PRODUCT[0]
```

Listing D.6

```
'did:web:emc.spacebel.be:collections:TropForest:items:KO2_OTPF_KO2_MSC_2F_2009
```

Figure D.3

```
response = requests.get( DID_RESOLVER + DID_PRODUCT[0] ,
    verify=True,
    headers={ 'Accept': 'application/json' })

data = json.loads(response.text)
jstr = json.dumps(data, indent=3)
md("```json\n" + jstr + "\n```\n")
```

Listing D.7

```
{
   "@context": [
      "https://www.w3.org/ns/did/v1",
      "https://w3id.org/security/suites/jws-2020/v1"
   ],
   "controller": "did:web:emc.spacebel.be:organisations:ESA",
   "id": "did:web:emc.spacebel.be:collections:TropForest:items:KO2_OTPF_KO2_MSC_
2F_20091107T041750_20091107T041750_017498_E082_N028",
   "alsoKnownAs": "https://emc.spacebel.be/collections/TropForest/items/KO2_OTPF_
KO2_MSC_2F_20091107T041750_20091107T041750_017498_E082_N028"
}
```

Listing D.8

D.4. Create a VC

```
# OKP is used in W3C examples of VC and DID
# the exp_key_dlr was created using this algorithm.
key = jwk.JWK.generate(kty='OKP', crv='Ed25519')
k = key.export(private_key=True)
k
```

Listing D.9

```
'{"crv":"Ed25519","d":"PmZjogtiwYLEuQ2dvBhbLId0M53fZMtexUVFTHm-
xxs","kty":"OKP","x":"exSjbI1eIjBXuvL21Zh6B6QhoTn1LOk-LDzpXHUa7RI"}'
```

Figure D.4

The `credentialSubject'' supports including specific properties about the subject which is identified by the` id". The properties must be defined using a JSON-LD @context. https://docs.ogc.org/is/17-003r2/17-003r2.html#78 defines a JSON-LD representation for all OGC EO Dataset Metadata GeoJSON(-LD) Encoding Standard (OGC 17-003r2) properties. Due to limitations of the `DIDKit` library, the @context cannot be included by reference to https://schemas.opengis.net/eo-geojson/1.0/eo-geojson.jsonld, but we include a subset of properties in the VC.

```
import didkit

# https://www.sprucekit.dev/verifiable-credentials/didkit/didkit-packages/
python#examples

# use Web did instead of key did (with similar key info as key example)
did = "did:web:emc.spacebel.be:organisations:de_dlr"

# additional properties still be added using proper JSON-LD...
# EO properties are defined as JSON-LD in the @context at https://docs.ogc.org/
is/17-003r2/17-003r2.html#78

credential = {
    "@context": [  "https://www.w3.org/2018/credentials/v1",
                   # "https://www.w3.org/ns/credentials/v2",
        {
            "date": "dct:date",
            "gj": "https://purl.org/geojson/vocab#",
            "data": "iana:enclosure",
            "bbox": {
                "@id": "gj:bbox",
                "@container": "@list"
            },
            "coordinates": "gj:coordinates",
            "title": "dct:title",
            "Feature": "gj:Feature",
            "dct": "http://purl.org/dc/terms/",
            "previews": "iana:icon",
            "owc": "http://www.opengis.net/ont/owc/1.0/",
            "iana": "http://www.iana.org/assignments/relation/",
            "geometry": "gj:geometry",
            "links": {
                "@id": "owc:links",
                "@context": {
                    "@vocab": "http://www.iana.org/assignments/relation/",
                    "type": "atom:type"
                }
            },
            "href": "@id",
            "id": "@id",
            "atom": "http://www.w3.org/2005/Atom/",
            "Polygon": "gj:Polygon",
            "updated": "dct:modified"
        }
    ],
    "id": DID_PRODUCT[0],
    "type": ["VerifiableCredential", "Feature"],
    "issuer": did,
    "issuanceDate": "2020-08-19T21:41:50Z",
    # "relatedResource": [{
    #     "id": "https://tpm-ds.eo.esa.int/oads/data/Tropforest/KO2_OTPF_KO2_MSC_
2F_20091107T041750_20091107T041750_017498_E082_N028.ZIP",
    #     # dummy value
```

```
    #     "digestMultibase": "uEres1usWcWCmW7uolIW2uA0CjQ8iRV14eGaZStJL73Vz"
    # }],
    "credentialSubject": {

       "id": DID_PRODUCT[0],
       # "extra" : "test"
       "bbox": [
            81.91450641,
            27.91026471,
            82.10769379,
            28.08365828
       ],
       "links": {
            "data": [ {
                "href": "https://tpm-ds.eo.esa.int/oads/data/Tropforest/KO2_OTPF_
KO2_MSC_2F_20091107T041750_20091107T041750_017498_E082_N028.ZIP", #,
                # "type": "application/zip",
                "title": "Download"
                }
            ]
       }
    }
}
```

Listing D.10

```
# resolve DID
response = requests.get( DID_RESOLVER + did,
    verify=True,
    headers={ 'Accept': 'application/json' })

data = json.loads(response.text)
jstr = json.dumps(data, indent=3)
md("```json\n" + jstr + "\n```\n")
```

Listing D.11

```
{
   "@context": [
      "https://www.w3.org/ns/did/v1",
      {
         "Ed25519VerificationKey2018": "https://w3id.org/security#Ed25519Verifica
tionKey2018",
         "publicKeyJwk": {
            "@id": "https://w3id.org/security#publicKeyJwk",
            "@type": "@json"
         }
      }
   ],
   "assertionMethod": [
      "did:web:emc.spacebel.be:organisations:de_dlr#z6MkwQSWK8dfis9Kp9kWUb4g2pG5i
aab92PafzbgL2nN1TS1"
   ],
   "id": "did:web:emc.spacebel.be:organisations:de_dlr",
   "verificationMethod": [
      {
         "id": "did:web:emc.spacebel.be:organisations:de_dlr#z6MkwQSWK8dfis9Kp9kW
Ub4g2pG5iaab92PafzbgL2nN1TS1",
         "type": "Ed25519VerificationKey2018",
         "controller": "did:web:emc.spacebel.be:organisations:de_dlr",
         "publicKeyJwk": {
            "kty": "OKP",
            "crv": "Ed25519",
```

```
            "x": "-93DZ8xtjHPn9VV4eohyXjgE0OWGnQcoJB6y8DEMB4I"
         }
      }
   ],
   "alsoKnownAs": [
      "https://gcmd.earthdata.nasa.gov/kms/concept/2f9d7c12-c02d-41fb-a168-
4d91794187f7",
      "https://dbpedia.org/resource/German_Aerospace_Center",
      "https://ror.org/04bwf3e34",
      "https://yago-knowledge.org/resource/German_Aerospace_Center"
   ],
   "authentication": [
      "did:web:emc.spacebel.be:organisations:de_dlr#z6MkwQSWK8dfis9Kp9kWUb4g2pG5i
aab92PafzbgL2nN1TS1"
   ]
}
```

Listing D.12

```
# The credential generated contains "jws" instead of "proofValue".
# jws appears in VC Model 1.1 (https://www.w3.org/TR/vc-data-model/).
# It no longer appears in Model 2.0 (https://www.w3.org/TR/vc-data-model/) ?
signed_credential = await didkit.issue_credential(
        json.dumps(credential),
        json.dumps({}),
        exp_key_dlr)
```

Listing D.13

```
jstr = json.dumps(json.loads(signed_credential), indent=3)
md("```json\n" + jstr + "\n```\n")
```

Listing D.14

```
{
   "@context": [
      "https://www.w3.org/2018/credentials/v1",
      {
         "owc": "http://www.opengis.net/ont/owc/1.0/",
         "data": "iana:enclosure",
         "Feature": "gj:Feature",
         "coordinates": "gj:coordinates",
         "title": "dct:title",
         "Polygon": "gj:Polygon",
         "dct": "http://purl.org/dc/terms/",
         "date": "dct:date",
         "iana": "http://www.iana.org/assignments/relation/",
         "previews": "iana:icon",
         "bbox": {
            "@container": "@list",
            "@id": "gj:bbox"
         },
         "gj": "https://purl.org/geojson/vocab#",
         "href": "@id",
         "id": "@id",
         "links": {
            "@context": {
               "@vocab": "http://www.iana.org/assignments/relation/",
               "type": "atom:type"
            },
            "@id": "owc:links"
         },
         "updated": "dct:modified",
```

```
            "atom": "http://www.w3.org/2005/Atom/",
            "geometry": "gj:geometry"
         }
      ],
      "id": "did:web:emc.spacebel.be:collections:TropForest:items:KO2_OTPF_KO2_MSC_
2F_20091107T041750_20091107T041750_017498_E082_N028",
      "type": [
         "VerifiableCredential",
         "Feature"
      ],
      "credentialSubject": {
         "id": "did:web:emc.spacebel.be:collections:TropForest:items:KO2_OTPF_KO2_
MSC_2F_20091107T041750_20091107T041750_017498_E082_N028",
         "bbox": [
            81.91450641,
            27.91026471,
            82.10769379,
            28.08365828
         ],
         "links": {
            "data": [
               {
                  "href": "https://tpm-ds.eo.esa.int/oads/data/Tropforest/KO2_OTPF_
KO2_MSC_2F_20091107T041750_20091107T041750_017498_E082_N028.ZIP",
                  "title": "Download"
               }
            ]
         }
      },
      "issuer": "did:web:emc.spacebel.be:organisations:de_dlr",
      "issuanceDate": "2020-08-19T21:41:50Z",
      "proof": {
         "type": "Ed25519Signature2018",
         "proofPurpose": "assertionMethod",
         "verificationMethod": "did:web:emc.spacebel.be:organisations:de_dlr#z6MkwQS
WK8dfis9Kp9kWUb4g2pG5iaab92PafzbgL2nN1TS1",
         "created": "2024-09-12T06:45:35.394Z",
         "jws": "eyJhbGciOiJFZERTQSIsImNyaXQiOlsiYjY0Il0sImI2NCI6ZmFsc2V9.
.O5rIpf6oFk_jrMHXGbgMlNHxBeYeEuogT5z8npzc9X2zGssE8yBzwwM945VLkXqtwwjo07KP7njzIDYy
r8lSAA"
      }
}
```


Listing D.15

D.5. Verify a VC

The verifier of a VC extracts the `proofPurpose` and `verificationMethod` from the `proof` in the VC.

```
# Get the proof section of the VC (and remove it from the document)
proof = json.loads(signed_credential).pop("proof")
print(json.dumps(proof, indent=2))
```

Listing D.16


```
{ "type": "Ed25519Signature2018", "proofPurpose": "assertionMethod",
"verificationMethod":
```

```
"did:web:emc.spacebel.be:organisations:de_dlr#z6MkwQSWK8dfis9Kp9kWUb4g2pG5iaab
"created": "2024-09-12T06:45:35.394Z", "jws":
"eyJhbGciOiJFZERTQSIsImNyaXQiOlsiYjY0Il0sImI2NCI6ZmFsc2V9..O5rIpf6oFk_jrMHXGbg
```

Figure D.5

```
options = {
    # "verificationMethod": "did:web:emc.spacebel.be:organisations:dl#z6MkwQSWK8d
fis9Kp9kWUb4g2pG5iaab92PafzbgL2nN1TS", # proof['verificationMethod'],
    # "proofPurpose": proof['proofPurpose']
  }

await didkit.verify_credential(signed_credential, json.dumps(options))
```

Listing D.17

```
'{"checks":["proof"],"warnings":[],"errors":[]}'
```

Figure D.6

The verification result output is a VerificationResult JSON object as specified in vc-http-api: https://w3c-ccg.github.io/vc-http-api/

D.6. Create a VP

The CEOS catalogue is the holder of a VC that was issued by ESA (or DLR). Create a presentation for a verifier.

Create presentation embedding verifiable credential. Prepare to present the verifiable credential by wrapping it in a Verifiable Presentation. The id here is an arbitrary URL for example purposes.

```
presentation_unsigned = {
    "@context": ["https://www.w3.org/2018/credentials/v1"],
    "id": "http://example.org/presentations/1560",
    "type": ["VerifiablePresentation"],
    "holder": "did:web:emc.spacebel.be:organisations:ceos",
    "verifiableCredential": json.loads(signed_credential)
}
```

Listing D.18

```
jstr = json.dumps(presentation_unsigned, indent=3)
md("```json\n" + jstr + "\n```\n")
```

Listing D.19

```
{
   "@context": [
      "https://www.w3.org/2018/credentials/v1"
   ],
   "id": "http://example.org/presentations/1560",
   "type": [
      "VerifiablePresentation"
   ],
   "holder": "did:web:emc.spacebel.be:organisations:ceos",
   "verifiableCredential": {
```

```
      "@context": [
         "https://www.w3.org/2018/credentials/v1",
         {
            "owc": "http://www.opengis.net/ont/owc/1.0/",
            "data": "iana:enclosure",
            "Feature": "gj:Feature",
            "coordinates": "gj:coordinates",
            "title": "dct:title",
            "Polygon": "gj:Polygon",
            "dct": "http://purl.org/dc/terms/",
            "date": "dct:date",
            "iana": "http://www.iana.org/assignments/relation/",
            "previews": "iana:icon",
            "bbox": {
               "@container": "@list",
               "@id": "gj:bbox"
            },
            "gj": "https://purl.org/geojson/vocab#",
            "href": "@id",
            "id": "@id",
            "links": {
               "@context": {
                  "@vocab": "http://www.iana.org/assignments/relation/",
                  "type": "atom:type"
               },
               "@id": "owc:links"
            },
            "updated": "dct:modified",
            "atom": "http://www.w3.org/2005/Atom/",
            "geometry": "gj:geometry"
         }
      ],
      "id": "did:web:emc.spacebel.be:collections:TropForest:items:KO2_OTPF_KO2_
MSC_2F_20091107T041750_20091107T041750_017498_E082_N028",
      "type": [
         "VerifiableCredential",
         "Feature"
      ],
      "credentialSubject": {
         "id": "did:web:emc.spacebel.be:collections:TropForest:items:KO2_OTPF_
KO2_MSC_2F_20091107T041750_20091107T041750_017498_E082_N028",
         "bbox": [
            81.91450641,
            27.91026471,
            82.10769379,
            28.08365828
         ],
         "links": {
            "data": [
               {
                  "href": "https://tpm-ds.eo.esa.int/oads/data/Tropforest/KO2_
OTPF_KO2_MSC_2F_20091107T041750_20091107T041750_017498_E082_N028.ZIP",
                  "title": "Download"
               }
            ]
         }
      },
      "issuer": "did:web:emc.spacebel.be:organisations:de_dlr",
      "issuanceDate": "2020-08-19T21:41:50Z",
      "proof": {
         "type": "Ed25519Signature2018",
         "proofPurpose": "assertionMethod",
```

```
        "verificationMethod": "did:web:emc.spacebel.be:organisations:de_dlr#z6Mk
wQSWK8dfis9Kp9kWUb4g2pG5iaab92PafzbgL2nN1TS1",
        "created": "2024-09-12T06:45:35.394Z",
        "jws": "eyJhbGciOiJFZERTQSIsImNyaXQiOlsiYjY0Il0sImI2NCI6ZmFsc2V9.
.O5rIpf6oFk_jrMHXGbgMlNHxBeYeEuogT5z8npzc9X2zGssE8yBzwwM945VLkXqtwwjo07KP7njzIDYy
r8lSAA"
      }
   }
}
```

Listing D.20

Issue the verifiable presentation. Pass the unsigned verifiable presentation to DIDkit to be issued as a verifiable presentation. DIDKit signs the presentation with a linked data proof, using the given key pair, verification method and proof type.

```
# Holder (CEOS) signs the Presentation with its private key.

options = {
    "verificationMethod": "did:web:emc.spacebel.be:organisations:ceos#owner",
    "proofPurpose": "authentication"
}

# Option 1 - use default options
# presentation_signed = await didkit.issue_presentation(  proof_options=json.
dumps({}),  presentation=json.dumps(presentation_unsigned), key=exp_key_ceos )

# Option 2 - use specific options
presentation_signed = await didkit.issue_presentation(  \
    proof_options=json.dumps(options),  \
    presentation=json.dumps(presentation_unsigned),  \
    key=exp_key_ceos )
```

Listing D.21

```
jstr = json.dumps(json.loads(presentation_signed), indent=3)
md("```json\n" + jstr + "\n```\n")
```

Listing D.22

```
{
   "@context": [
      "https://www.w3.org/2018/credentials/v1"
   ],
   "id": "http://example.org/presentations/1560",
   "type": [
      "VerifiablePresentation"
   ],
   "verifiableCredential": {
      "@context": [
         "https://www.w3.org/2018/credentials/v1",
         {
            "title": "dct:title",
            "Polygon": "gj:Polygon",
            "iana": "http://www.iana.org/assignments/relation/",
            "bbox": {
               "@container": "@list",
               "@id": "gj:bbox"
            },
            "href": "@id",
            "links": {
               "@context": {
```

```
                  "@vocab": "http://www.iana.org/assignments/relation/",
                  "type": "atom:type"
               },
               "@id": "owc:links"
            },
            "updated": "dct:modified",
            "Feature": "gj:Feature",
            "geometry": "gj:geometry",
            "atom": "http://www.w3.org/2005/Atom/",
            "owc": "http://www.opengis.net/ont/owc/1.0/",
            "data": "iana:enclosure",
            "coordinates": "gj:coordinates",
            "dct": "http://purl.org/dc/terms/",
            "gj": "https://purl.org/geojson/vocab#",
            "previews": "iana:icon",
            "id": "@id",
            "date": "dct:date"
         }
      ],
      "id": "did:web:emc.spacebel.be:collections:TropForest:items:KO2_OTPF_KO2_
MSC_2F_20091107T041750_20091107T041750_017498_E082_N028",
      "type": [
         "VerifiableCredential",
         "Feature"
      ],
      "credentialSubject": {
         "id": "did:web:emc.spacebel.be:collections:TropForest:items:KO2_OTPF_
KO2_MSC_2F_20091107T041750_20091107T041750_017498_E082_N028",
         "bbox": [
            81.91450641,
            27.91026471,
            82.10769379,
            28.08365828
         ],
         "links": {
            "data": [
               {
                  "href": "https://tpm-ds.eo.esa.int/oads/data/Tropforest/KO2_
OTPF_KO2_MSC_2F_20091107T041750_20091107T041750_017498_E082_N028.ZIP",
                  "title": "Download"
               }
            ]
         }
      },
      "issuer": "did:web:emc.spacebel.be:organisations:de_dlr",
      "issuanceDate": "2020-08-19T21:41:50Z",
      "proof": {
         "type": "Ed25519Signature2018",
         "proofPurpose": "assertionMethod",
         "verificationMethod": "did:web:emc.spacebel.be:organisations:de_dlr#z6Mk
wQSWK8dfis9Kp9kWUb4g2pG5iaab92PafzbgL2nN1TS1",
         "created": "2024-09-12T06:45:35.394Z",
         "jws": "eyJhbGciOiJFZERTQSIsImNyaXQiOlsiYjY0Il0sImI2NCI6ZmFsc2V9.
.O5rIpf6oFk_jrMHXGbgMlNHxBeYeEuogT5z8npzc9X2zGssE8yBzwwM945VLkXqtwwjo07KP7njzIDYy
r8lSAA"
      }
   },
   "proof": {
      "type": "Ed25519Signature2018",
      "proofPurpose": "authentication",
      "verificationMethod": "did:web:emc.spacebel.be:organisations:ceos#owner",
      "created": "2024-09-12T06:45:35.710Z",
```

```
      "jws": "eyJhbGciOiJFZERTQSIsImNyaXQiOlsiYjY0Il0sImI2NCI6ZmFsc2V9..3d15xLVlk
JjV1XAjnuqXDcNQWSL3wamX9YLTcqFWOY6axJEjKMqXOsr3CrdavQqvxJUoi3FQ1x37ZNJBIxugBQ"
   },
   "holder": "did:web:emc.spacebel.be:organisations:ceos"
}
```

Listing D.23

D.7. Verify a VP

Verifier checks the signature of the Presentation with the public key of the Holder.

```
options = {
    "verificationMethod": "did:web:emc.spacebel.be:organisations:ceos#owner",
    "proofPurpose": "authentication"
}

# options can be found in the VP.

# option 1 - default options
await didkit.verify_presentation(presentation_signed, json.dumps({}))
# option 2 - specific options
# await didkit.verify_presentation(presentation_signed, json.dumps(options))
```

Listing D.24

```
'{"checks":["proof"],"warnings":[],"errors":[]}'
```

Figure D.7

D.8. Content-integrity Protection

The VC 2.0 Data Model proposes a `relatedResource` property with multibase digest as per https://www.w3.org/TR/vc-data-integrity/#resource-integrity. The VC 1.1 Data Model proposes a multibase multihash value e.g. provided as value of a `hl` query parameter using https://datatracker.ietf.org/doc/html/draft-sporny-hashlink-07. See also See https://w3c-ccg.github.io/hashlink/#hl-url-params.

The value of the resource hash can be generated by utilizing the following algorithm:

- Generate the raw hash value by processing the resource data using the cryptographic hashing algorithm.
- Generate the multihash value by encoding the raw hash using the Multihash Data Format.
- Generate the multibase hash by encoding the multihash value using the Multibase Data Format.
- Output the multibase hash as the resource hash.

```
# Generate the multihash value by encoding the raw hash using the Multihash Data
Format.

# https://github.com/hashberg-io/multiformats
# https://multiformats.readthedocs.io/en/latest/

# import hashlib
# from urllib.request import urlopen
from multiformats import multihash

def get_remote_multihash(url,algorithm):
    # only for "sha2-256"
    hex_digest = get_remote_hash(url,algorithm).hexdigest()
    mh = multihash.get(algorithm)
    # raw_digest = bytes.fromhex("c0535e4be2b79ffd93291305436bf889314e4a3f")
    raw_digest = bytes.fromhex(hex_digest)
    # sha2_256.wrap(raw_digest).hex()
    return mh.wrap(raw_digest)

mh = get_remote_multihash(DATAFILE, 'sha2-256')
print ("Multihash: ", mh.hex() )
```

Listing D.25

```
Multihash:
122089c45c7677a72635657488018110595 3b48a8ff56275d24add8b6cabdcd516b9
```

Figure D.8

```
# 12201e6bb7a9f0bdb593a4c7da34271d34a14fbbf85a199499eae89bb5de9aa5790b
```

Listing D.26

```
from multiformats import multibase
from multiformats.multibase import Multibase
from multiformats import varint

ENCODING_BASE = "base58btc"
# ENCODING_BASE = "base32"
```

Listing D.27

```
multibase.exists( ENCODING_BASE )
```

Listing D.28

```
True
```

Figure D.9

```
multibase.get( ENCODING_BASE )
```

Listing D.29

```
Multibase(name='base58btc', code='z', status='final',
description='Base58 Bitcoin')
```

Figure D.10

```
# Generate the multibase hash by encoding the multihash value using the
Multibase Data Format.
# Output the multibase hash as the resource hash.
```

```
# https://multiformats.readthedocs.io/en/latest/multibase.html

# from multiformats import multibase
# from multiformats.multibase import Multibase

def get_remote_multibase_hash(url, algorithm, base):
    # only for "sha2-256"
    mh = get_remote_multihash(url, algorithm)
    return multibase.encode(mh, base)

mbh = get_remote_multibase_hash(DATAFILE, 'sha2-256', ENCODING_BASE)
print ("Multibase hash: ", mbh )
```

Listing D.30

```
Multibase hash: zQmXcSE7jarcSJPGS3qvhuwAkesHJpfrB5wq39NDX1w2pqv
```

Figure D.11

```
# show details of multibase string.
multibase.from_str(mbh)
```

Listing D.31

```
Multibase(name='base58btc', code='z', status='final',
description='Base58 Bitcoin')
```

Figure D.12

```
digest = multibase.decode(mbh)
digest
```

Listing D.32

```
b'\x12 \x89\xc4\\vw\xa7&5et\x88\x01\x81\x10YS\xb4\x8a\x8f\xf5bu\xd2J
\xdd\x8bl\xab\xdc\xd5\x16\xb9'
```

Figure D.13

```
hex_function = digest.hex()[0:2]  # https://www.ietf.org/archive/id/draft-
multiformats-multihash-07.html#name-hash-function-identifier
print(f"Multihash function code\t: { hex(varint.decode(bytes.fromhex(hex_
function))) } (hex)")

hex_length = digest.hex()[2:4]    # https://www.ietf.org/archive/id/draft-
multiformats-multihash-07.html#name-digest-length
print(f"Multihash length\t: { hex(varint.decode(bytes.fromhex(hex_length))) }
(hex)")
```

Listing D.33

```
Multihash function code : 0x12 (hex) Multihash length : 0x20 (hex)
```

Figure D.14

```
# show details of self-describing multihash string.
multihash.from_digest(digest).codec
```

Listing D.34

```
Multicodec(name='sha2-256', tag='multihash', code='0x12',
status='permanent', description='')
```

Figure D.15

D.8.1. VC Data Model 1.1

```
{
   "@context": [
      "https://www.w3.org/2018/credentials/v1"
   ],
   "credentialSubject": {
      "links": {
         "data": [
            {
               "href": "https://data.cci.ceda.ac.uk/thredds/fileServer/esacci/
land_surface_temperature/data/SENTINEL3A_SLSTR/L3C/0.01/v3.00/daily/2016/05/06/
ESACCI-LST-L3C-LST-SLSTRA-0.01deg_1DAILY_DAY-20160506000000-fv3.00.nc&hl=zQmXcSE7
jarcSJPGS3qvhuwAkesHJpfrB5wq39NDX1w2pqv",
               "title": "Download"
            }
         ]
      }
   }
}
```

Listing D.35

D.8.2. VC Data Model 2.0

```
{
   "@context": [
      "https://www.w3.org/ns/credentials/v2"
   ],
   "credentialSubject": {
      "links": {
         "data": [
            {
               "href": "https://data.cci.ceda.ac.uk/thredds/fileServer/esacci/
land_surface_temperature/data/SENTINEL3A_SLSTR/L3C/0.01/v3.00/daily/2016/05/06/
ESACCI-LST-L3C-LST-SLSTRA-0.01deg_1DAILY_DAY-20160506000000-fv3.00.nc",
               "title": "Download"
            }
         ]
      }
   },
   "relatedResource": [
      {
         "id": "https://data.cci.ceda.ac.uk/thredds/fileServer/esacci/land_
surface_temperature/data/SENTINEL3A_SLSTR/L3C/0.01/v3.00/daily/2016/05/06/ESACCI-
LST-L3C-LST-SLSTRA-0.01deg_1DAILY_DAY-20160506000000-fv3.00.nc",
         "digestMultibase": "zQmXcSE7jarcSJPGS3qvhuwAkesHJpfrB5wq39NDX1w2pqv"
      }
   ]
}
```

Listing D.36

D.9. Interactions with Catalogue

D.9.1. Access OGC API-Records

```
w = Records('https://emc.spacebel.be')
w.conformance()
```

Listing D.37


```
{'conformsTo': ['http://www.opengis.net/spec/ogcapi-features-1/1.0/
conf/core', 'http://www.opengis.net/spec/ogcapi-features-1/1.0/conf/
oas30', 'http://www.opengis.net/spec/ogcapi-features-1/1.0/conf/
geojson', 'http://www.opengis.net/spec/ogcapi_common-2/1.0/conf/
collections', 'http://www.opengis.net/spec/ogcapi-common-2/1.0/conf/
simple-query', 'http://www.opengis.net/spec/ogcapi-records-1/1.0/
req/cql-filter', 'http://www.opengis.net/spec/ogcapi-features-1/1.0/
conf/geojson', 'http://www.opengis.net/spec/ogcapi-features-3/1.0/
conf/features-filter', 'https://api.stacspec.org/v1.0.0-rc.2/
core', 'https://api.stacspec.org/v1.0.0-rc.2/stac-search',
'https://api.stacspec.org/v1.0.0-rc.2/stac-response', 'https://
api.stacspec.org/v1.0.0-rc.2/collection-search', 'https://
api.stacspec.org/v1.0.0-rc.2/collection-search#filter', 'https://
api.stacspec.org/v1.0.0-rc.1/collection-search#free-text', 'https://
api.stacspec.org/v1.0.0-rc.1/collection-search#sort', 'https://
api.stacspec.org/v1.0.0-rc.2/item-search', 'https://api.stacspec.org/
v1.0.0-rc.2/item-search#filter', 'http://www.opengis.net/spec/
cql2/1.0/conf/cql2-text', 'http://www.opengis.net/spec/cql2/1.0/conf/
basic-cql2', 'https://api.stacspec.org/v1.0.0/item-search#sort',
'https://api.stacspec.org/v1.0.0/ogcapi-features#sort']}
```


Figure D.16

```
# Search within INPE Amazonia-1 collection
COLLECTION_ID = 'AMZ1-WFI-L4-SR-1'
query = w.collection_items(COLLECTION_ID)
query['features'][0]['id']
```

Listing D.38


```
'AMAZONIA_1_WFI_20240831_034_015'
```


Figure D.17

```
url = "https://radiantearth.github.io/stac-browser/#/external/emc.spacebel.be/
collections/" + COLLECTION_ID
md("View this collection with a Catalogue client at " + "[" + url + "](" + url +
").  All granules have DID and VC information as additional resources accessible
from the `links` section of the metadata.\n")
```

Listing D.39

View this collection with a Catalogue client at https://radiantearth.github.io/stac-browser/#/external/emc.spacebel.be/collections/AMZ1-WFI-L4-SR-1. All granules have DID and VC information as additional resources accessible from the `links` section of the metadata.

```
query['features'][0]['links']
```

Listing D.40


```
[{'rel': 'self', 'href': 'https://emc.spacebel.be/collections/
AMZ1-WFI-L4-SR-1/items/AMAZONIA_1_WFI_20240831_034_015',
'type': 'application/geo+json'}, {'rel': 'enclosure',
'href': 'https://data.inpe.br/bdc/data/amazonia_wfi/2024_08/
AMAZONIA_1_WFI_RAW_2024_08_31.13_13_30_CB10/034_015_0/4_BC_LCC_WGS84/
AMAZONIA_1_WFI_20240831_034_015_L4_CMASK_GRID_SURFACE.tif',
'type': 'image/tiff; application=geotiff; profile=cloud-
optimized', 'title': 'CMASK'}, {'rel': 'enclosure', 'href':
'https://data.inpe.br/bdc/data/amazonia_wfi/2024_08/
AMAZONIA_1_WFI_RAW_2024_08_31.13_13_30_CB10/034_015_0/4_BC_LCC_WGS84/
AMAZONIA_1_WFI_20240831_034_015_L4_BAND1_GRID_SURFACE.tif',
'type': 'image/tiff; application=geotiff; profile=cloud-
optimized', 'title': 'BAND1'}, {'rel': 'enclosure', 'href':
'https://data.inpe.br/bdc/data/amazonia_wfi/2024_08/
AMAZONIA_1_WFI_RAW_2024_08_31.13_13_30_CB10/034_015_0/4_BC_LCC_WGS84/
AMAZONIA_1_WFI_20240831_034_015_L4_BAND2_GRID_SURFACE.tif',
'type': 'image/tiff; application=geotiff; profile=cloud-
optimized', 'title': 'BAND2'}, {'rel': 'enclosure', 'href':
'https://data.inpe.br/bdc/data/amazonia_wfi/2024_08/
AMAZONIA_1_WFI_RAW_2024_08_31.13_13_30_CB10/034_015_0/4_BC_LCC_WGS84/
AMAZONIA_1_WFI_20240831_034_015_L4_BAND3_GRID_SURFACE.tif',
'type': 'image/tiff; application=geotiff; profile=cloud-
optimized', 'title': 'BAND3'}, {'rel': 'enclosure', 'href':
'https://data.inpe.br/bdc/data/amazonia_wfi/2024_08/
AMAZONIA_1_WFI_RAW_2024_08_31.13_13_30_CB10/034_015_0/4_BC_LCC_WGS84/
AMAZONIA_1_WFI_20240831_034_015_L4_BAND4_GRID_SURFACE.tif',
'type': 'image/tiff; application=geotiff; profile=cloud-
optimized', 'title': 'BAND4'}, {'rel': 'preview', 'href':
'https://data.inpe.br/bdc/data/amazonia_wfi/2024_08/
AMAZONIA_1_WFI_RAW_2024_08_31.13_13_30_CB10/034_015_0/4_BC_LCC_WGS84/
AMAZONIA_1_WFI_20240831_034_015.png', 'type': 'image/png',
'title': 'Preview'}, {'rel': 'canonical', 'href': 'https://
data.inpe.br/bdc/stac/v1/collections/AMZ1-WFI-L4-SR-1/items/
AMAZONIA_1_WFI_20240831_034_015', 'type': 'application/geo+json',
'title': 'Original metadata'}, {'rel': 'collection', 'href': 'https://
emc.spacebel.be/collections/AMZ1-WFI-L4-SR-1?httpAccept=application/
json', 'type': 'application/json', 'title': 'AMZ1-WFI-L4-SR-1'},
{'rel': 'parent', 'href': 'https://emc.spacebel.be/collections/
AMZ1-WFI-L4-SR-1?httpAccept=application/json', 'type': 'application/
json', 'title': 'AMZ1-WFI-L4-SR-1'}, {'rel': 'up', 'href': 'https://
emc.spacebel.be/collections/AMZ1-WFI-L4-SR-1', 'type': 'application/
geo+json', 'title': 'OGC 17-069r3 metadata'}, {'rel': 'alternate',
'href': 'https://emc.spacebel.be/collections/AMZ1-WFI-L4-SR-1/items/
AMAZONIA_1_WFI_20240831_034_015?httpAccept=application/atom%2Bxml',
'type': 'application/atom+xml', 'title': 'Atom format'}, {'rel':
'alternate', 'href': 'https://emc.spacebel.be/collections/AMZ1-WFI-L4-
SR-1/items/AMAZONIA_1_WFI_20240831_034_015?httpAccept=application/gml
```

%2Bxml&recordSchema=om', 'type': 'application/gml+xml;profile="http://www.opengis.net/spec/EOMPOM/1.1"', 'title': 'OGC 10-157r4 metadata'}, {'rel': 'alternate', 'href': 'https://emc.spacebel.be/collections/AMZ1-WFI-L4-SR-1/items/AMAZONIA_1_WFI_20240831_034_015?httpAccept=application/gml%2Bxml&recordSchema=om10', 'type': 'application/gml+xml;profile="http://www.opengis.net/spec/EOMPOM/1.0"', 'title': 'OGC 10-157r3 metadata'}, {'rel': 'alternate', 'href': 'https://emc.spacebel.be/collections/AMZ1-WFI-L4-SR-1/items/AMAZONIA_1_WFI_20240831_034_015?mode=owc', 'type': 'application/geo+json;profile="http://www.opengis.net/spec/eo-geojson/1.0"', 'title': 'OGC 17-003r2 metadata'}, {'rel': 'alternate', 'href': 'https://emc.spacebel.be/collections/AMZ1-WFI-L4-SR-1/items/AMAZONIA_1_WFI_20240831_034_015?httpAccept=application/geo%2Bjson;profile=https://stacspec.org', 'type': 'application/geo+json;profile="https://stacspec.org"', 'title': 'STAC metadata'}, {'rel': 'alternate', 'href': 'https://emc.spacebel.be/collections/AMZ1-WFI-L4-SR-1/items/AMAZONIA_1_WFI_20240831_034_015?httpAccept=application/vnd.iso.19139%2Bxml', 'type': 'application/vnd.iso.19139+xml', 'title': 'ISO 19139 metadata'}, {'rel': 'alternate', 'href': 'https://emc.spacebel.be/collections/AMZ1-WFI-L4-SR-1/items/AMAZONIA_1_WFI_20240831_034_015?httpAccept=application/ld%2Bjson', 'type': 'application/ld+json', 'title': 'JSON-LD metadata'}, {'rel': 'alternate', 'href': 'https://emc.spacebel.be/collections/AMZ1-WFI-L4-SR-1/items/AMAZONIA_1_WFI_20240831_034_015?httpAccept=application/ld%2Bjson;profile=https://schema.org', 'type': 'application/ld+json;profile="https://schema.org"', 'title': 'JSON-LD (schema.org) metadata'}, {'rel': 'alternate', 'href': 'https://emc.spacebel.be/collections/AMZ1-WFI-L4-SR-1/items/AMAZONIA_1_WFI_20240831_034_015?httpAccept=application/ld%2Bjson;profile=http://data.europa.eu/930/', 'type': 'application/ld+json;profile="http://data.europa.eu/930/"', 'title': 'JSON-LD (GeoDCAT-AP) metadata'}, {'rel': 'alternate', 'href': 'https://emc.spacebel.be/collections/AMZ1-WFI-L4-SR-1/items/AMAZONIA_1_WFI_20240831_034_015?httpAccept=application/rdf%2Bxml', 'type': 'application/rdf+xml', 'title': 'RDF/XML metadata'}, {'rel': 'alternate', 'href': 'https://emc.spacebel.be/collections/AMZ1-WFI-L4-SR-1/items/AMAZONIA_1_WFI_20240831_034_015?httpAccept=application/rdf%2Bxml;profile=https://schema.org', 'type': 'application/rdf+xml;profile="https://schema.org"', 'title': 'RDF/XML (schema.org) metadata'}, {'rel': 'alternate', 'href': 'https://emc.spacebel.be/collections/AMZ1-WFI-L4-SR-1/items/AMAZONIA_1_WFI_20240831_034_015?httpAccept=application/rdf%2Bxml;profile=http://data.europa.eu/930/', 'type': 'application/rdf+xml;profile="http://data.europa.eu/930/"', 'title': 'RDF/XML (GeoDCAT-AP) metadata'}, {'rel': 'alternate', 'href': 'https://emc.spacebel.be/collections/AMZ1-WFI-L4-SR-1/items/AMAZONIA_1_WFI_20240831_034_015?httpAccept=text/turtle', 'type': 'text/turtle', 'title': 'Turtle metadata'}, {'rel': 'alternate', 'href': 'https://emc.spacebel.be/collections/AMZ1-WFI-L4-SR-1/items/AMAZONIA_1_WFI_20240831_034_015?httpAccept=text/turtle;profile=https://schema.org', 'type': 'text/turtle;profile="https://schema.org"', 'title': 'Turtle (schema.org) metadata'}, {'rel': 'alternate', 'href': 'https://emc.spacebel.be/collections/AMZ1-WFI-L4-SR-1/items/AMAZONIA_1_WFI_20240831_034_015?httpAccept=text/turtle;profile=http://data.europa.eu/930/', 'type':

```
'text/turtle;profile="http://data.europa.eu/930/"', 'title':
'Turtle (GeoDCAT-AP) metadata'}, {'rel': 'alternate', 'href':
'https://emc.spacebel.be/collections/AMZ1-WFI-L4-SR-1/items/
AMAZONIA_1_WFI_20240831_034_015?httpAccept=text/html', 'type':
'text/html', 'title': 'HTML'}, {'rel': 'alternate', 'href':
'https://emc.spacebel.be/collections/AMZ1-WFI-L4-SR-1/items/
AMAZONIA_1_WFI_20240831_034_015?httpAccept=application/vc%2Bld
%2Bjson', 'type': 'application/vc+ld+json', 'title': 'Verifiable
Credential'}, {'rel': 'alternate', 'href': 'https://emc.spacebel.be/
collections/AMZ1-WFI-L4-SR-1/items/AMAZONIA_1_WFI_20240831_034_015?
httpAccept=application/vp%2Bld%2Bjson', 'type': 'application/vp+ld
+json', 'title': 'Verifiable Presentation'}, {'rel': 'describes',
'href': 'did:web:emc.spacebel.be:collections:AMZ1-WFI-L4-
SR-1:items:AMAZONIA_1_WFI_20240831_034_015', 'type': 'application/did
+json', 'title': 'DID'}]
```

Figure D.18

D.9.2. Obtain DID for EO resource

DID information can be included in GeoJSON or XML encoded metadata records (e.g. STAC, OGC 17-003r2) as a link. `did:` is a valid URI scheme. Relation `describes` is used.

```
# To find DID for a resource (granule), extract link with type "application/did
+json"
# take first metadata record in the response
data = query['features'][0]

from jsonpath_ng.ext import parse

expression = parse("$.links[?(@.type) == 'application/did+json']")
r = expression.find(data)
r[0].value
```

Listing D.41

```
{'rel': 'describes', 'href':
'did:web:emc.spacebel.be:collections:AMZ1-WFI-L4-
SR-1:items:AMAZONIA_1_WFI_20240831_034_015', 'type': 'application/did
+json', 'title': 'DID'}
```

Figure D.19

A possible encoding of the DID identifier (identifying an EO collection) in an ISO-19139 encoding is shown below. This collection has a DOI as well as a DID identifier.

```
<gmd:MD_DataIdentification>
  <gmd:citation>
    <gmd:CI_Citation>
      <gmd:title>
        <gco:CharacterString>TropForest- ALOS, GEOSAT-1 &amp;amp; KOMPSAT-2 optical
coverages over tropical forests</gco:CharacterString>
      </gmd:title>
      <gmd:identifier>
        <gmd:RS_Identifier>
          <gmd:code>
            <gco:CharacterString>10.5270/esa-qoe849q</gco:CharacterString>
```

```
          </gmd:code>
          <gmd:codeSpace>
            <gco:CharacterString>https://doi.org</gco:CharacterString>
          </gmd:codeSpace>
        </gmd:RS_Identifier>
      </gmd:identifier>
      <gmd:identifier>
        <gmd:RS_Identifier>
          <gmd:code>
            <gco:CharacterString>id:web:fedeo.ceos.org:collections:Tropforest</
gco:CharacterString>
          </gmd:code>
          <gmd:codeSpace>
            <gco:CharacterString>https://www.w3.org/ns/did/v1</gco:
CharacterString>
          </gmd:codeSpace>
        </gmd:RS_Identifier>
      </gmd:identifier>
    </gmd:CI_Citation>
  </gmd:citation>
```

Listing D.42

D.9.3. Obtain VC for EO resource

VC information can be included in GeoJSON or XML encoded metadata records (e.g. STAC, OGC 17-003r2) as a link with type `application/vc+ld+json`. This type is proposed by EBSI at https://hub.ebsi.eu/vc-framework/data-models/vcdm-version-update#new-media-type.

```
# To find VC for a resource (granule), extract link with type "application/vc+ld
+json"
# take first metadata record in the response
data = query['features'][0]

from jsonpath_ng.ext import parse

expression = parse("$.links[?(@.type) == 'application/vc+ld+json']")
r = expression.find(data)
r[0].value
```

Listing D.43

```
{'rel': 'alternate', 'href': 'https://emc.spacebel.be/collections/
AMZ1-WFI-L4-SR-1/items/AMAZONIA_1_WFI_20240831_034_015?
httpAccept=application/vc%2Bld%2Bjson', 'type': 'application/vc+ld
+json', 'title': 'Verifiable Credential'}
```

Figure D.20

```
r[0].value['href']
```

Listing D.44

```
'https://emc.spacebel.be/collections/AMZ1-WFI-L4-SR-1/items/
AMAZONIA_1_WFI_20240831_034_015?httpAccept=application/vc%2Bld%2Bjson'
```

Figure D.21

The context document https://schemas.opengis.net/eo-geojson/1.0/eo-geojson.jsonld is not included in the VC @context due to a limitation of the DID Python library used.

```
response = requests.get( r[0].value['href'] ,
    verify=True,
    headers={  })

data = json.loads(response.text)
jstr = json.dumps(data, indent=3)
md("```json\n" + jstr + "\n```\n")
```

Listing D.45


```
{
   "credentialSubject": {
      "date": "2024-08-31T00:00:00.000000Z/2024-08-31T00:00:00.000000Z",
      "bbox": [
         -44.078494,
         -3.931139,
         -34.874524,
         4.18639
      ],
      "links": {
         "data": [
            {
               "href": "https://data.inpe.br/bdc/data/amazonia_wfi/2024_08/
AMAZONIA_1_WFI_RAW_2024_08_31.13_13_30_CB10/034_015_0/4_BC_LCC_WGS84/AMAZONIA_1_
WFI_20240831_034_015_L4_CMASK_GRID_SURFACE.tif",
               "title": "CMASK",
               "type": "image/tiff; application=geotiff; profile=cloud-optimized"
            },
            {
               "href": "https://data.inpe.br/bdc/data/amazonia_wfi/2024_08/
AMAZONIA_1_WFI_RAW_2024_08_31.13_13_30_CB10/034_015_0/4_BC_LCC_WGS84/AMAZONIA_1_
WFI_20240831_034_015_L4_BAND1_GRID_SURFACE.tif",
               "title": "BAND1",
               "type": "image/tiff; application=geotiff; profile=cloud-optimized"
            },
            {
               "href": "https://data.inpe.br/bdc/data/amazonia_wfi/2024_08/
AMAZONIA_1_WFI_RAW_2024_08_31.13_13_30_CB10/034_015_0/4_BC_LCC_WGS84/AMAZONIA_1_
WFI_20240831_034_015_L4_BAND2_GRID_SURFACE.tif",
               "title": "BAND2",
               "type": "image/tiff; application=geotiff; profile=cloud-optimized"
            },
            {
               "href": "https://data.inpe.br/bdc/data/amazonia_wfi/2024_08/
AMAZONIA_1_WFI_RAW_2024_08_31.13_13_30_CB10/034_015_0/4_BC_LCC_WGS84/AMAZONIA_1_
WFI_20240831_034_015_L4_BAND3_GRID_SURFACE.tif",
               "title": "BAND3",
               "type": "image/tiff; application=geotiff; profile=cloud-optimized"
            },
            {
               "href": "https://data.inpe.br/bdc/data/amazonia_wfi/2024_08/
AMAZONIA_1_WFI_RAW_2024_08_31.13_13_30_CB10/034_015_0/4_BC_LCC_WGS84/AMAZONIA_1_
WFI_20240831_034_015_L4_BAND4_GRID_SURFACE.tif",
               "title": "BAND4",
               "type": "image/tiff; application=geotiff; profile=cloud-optimized"
            }
         ],
         "previews": [
            {
```

```
                "href": "https://data.inpe.br/bdc/data/amazonia_wfi/2024_08/
AMAZONIA_1_WFI_RAW_2024_08_31.13_13_30_CB10/034_015_0/4_BC_LCC_WGS84/AMAZONIA_1_
WFI_20240831_034_015.png",
                "title": "Preview",
                "type": "image/png"
             }
          ]
       },
       "geometry": {
          "coordinates": [
             [
                [
                   -44.078494,
                   -3.931139
                ],
                [
                   -44.078494,
                   4.18639
                ],
                [
                   -34.874524,
                   4.18639
                ],
                [
                   -34.874524,
                   -3.931139
                ],
                [
                   -44.078494,
                   -3.931139
                ]
             ]
          ],
          "type": "Polygon"
       },
       "id": "did:web:emc.spacebel.be:collections:AMZ1-WFI-L4-SR-1:items:AMAZONIA_
1_WFI_20240831_034_015",
       "title": "AMAZONIA_1_WFI_20240831_034_015",
       "updated": "2024-09-05T19:03:04.887222Z"
    },
    "issuanceDate": "2024-09-12T06:47:20Z",
    "id": "did:web:emc.spacebel.be:collections:AMZ1-WFI-L4-SR-1:items:AMAZONIA_1_
WFI_20240831_034_015",
    "proof": {
       "created": "2024-09-12T06:47:21.017Z",
       "jws": "eyJhbGciOiJFUzI1NksiLCJjcml0IjpbImI2NCJdLCJiNjQiOmZhbHNlfQ.
.ParvZXf2mVJ_Ip2FuCjpl26eGJokz5cABS-bJEXFrWEBeMGlJilZJiCGBLBV6Zwzhs8QCLsi9HXgofrk
eE4j6A",
       "proofPurpose": "assertionMethod",
       "type": "EcdsaSecp256k1Signature2019",
       "verificationMethod": "did:web:emc.spacebel.be:organisations:br_inpe#owner"
    },
    "type": [
       "VerifiableCredential",
       "Feature"
    ],
    "@context": [
       "https://www.w3.org/2018/credentials/v1",
       {
          "date": "dct:date",
          "gj": "https://purl.org/geojson/vocab#",
          "data": "iana:enclosure",
          "bbox": {
```

```
            "@id": "gj:bbox",
            "@container": "@list"
         },
         "coordinates": "gj:coordinates",
         "title": "dct:title",
         "Feature": "gj:Feature",
         "dct": "http://purl.org/dc/terms/",
         "previews": "iana:icon",
         "geometry": "gj:geometry",
         "iana": "http://www.iana.org/assignments/relation/",
         "links": {
            "@id": "owc:links",
            "@context": {
               "@vocab": "http://www.iana.org/assignments/relation/",
               "type": "atom:type"
            }
         },
         "owc": "http://www.opengis.net/ont/owc/1.0/",
         "href": "@id",
         "id": "@id",
         "Polygon": "gj:Polygon",
         "atom": "http://www.w3.org/2005/Atom/",
         "updated": "dct:modified"
      }
   ],
   "issuer": "did:web:emc.spacebel.be:organisations:br_inpe"
}
```

Listing D.46

D.9.4. Verify VC for EO resource

```
# verify the proof included in the VC (if any)

options = {}
await didkit.verify_credential( json.dumps(data), json.dumps(options))
```

Listing D.47

```
'{"checks":["proof"],"warnings":[],"errors":[]}'
```

Figure D.22

```
# To find VP for a resource (granule), extract link with type "application/vp+ld
+json"
# take first metadata record in the response
data = query['features'][0]

from jsonpath_ng.ext import parse

expression = parse("$.links[?(@.type) == 'application/vp+ld+json']")
r = expression.find(data)
r[0].value
```

Listing D.48

```
{'rel': 'alternate', 'href': 'https://emc.spacebel.be/collections/
AMZ1-WFI-L4-SR-1/items/AMAZONIA_1_WFI_20240831_034_015?
```

```
httpAccept=application/vp%2Bld%2Bjson', 'type': 'application/vp+ld
+json', 'title': 'Verifiable Presentation'}
```

Figure D.23

```
response = requests.get( r[0].value['href'] ,
    verify=True,
    headers={  })

data = json.loads(response.text)
jstr = json.dumps(data, indent=3)
md("```json\n" + jstr + "\n```\n")
```

Listing D.49


```
{
   "holder": "did:web:emc.spacebel.be:organisations:ceos",
   "proof": {
      "created": "2024-09-12T06:47:50.423Z",
      "jws": "eyJhbGciOiJFZERTQSIsImNyaXQiOlsiYjY0Il0sImI2NCI6ZmFsc2V9..juXb-
CkiejYvFNiUQOuIGc_Pdsi3XdN-pGNzqbJ2UQ4dopNwJu3naGa9fXSUGmhs6iKVz5hEdt79xvT8vOlaCA
",
      "proofPurpose": "authentication",
      "type": "Ed25519Signature2018",
      "verificationMethod": "did:web:emc.spacebel.be:organisations:ceos#owner"
   },
   "type": [
      "VerifiablePresentation"
   ],
   "@context": [
      "https://www.w3.org/2018/credentials/v1"
   ],
   "verifiableCredential": [
      {
         "credentialSubject": {
            "date": "2024-08-31T00:00:00.000000Z/2024-08-31T00:00:00.000000Z",
            "bbox": [
               -44.078494,
               -3.931139,
               -34.874524,
               4.18639
            ],
            "geometry": {
               "coordinates": [
                  [
                     [
                        -44.078494,
                        -3.931139
                     ],
                     [
                        -44.078494,
                        4.18639
                     ],
                     [
                        -34.874524,
                        4.18639
                     ],
                     [
                        -34.874524,
                        -3.931139
                     ],
                     [
                        -44.078494,
```

```
                        -3.931139
                     ]
                  ]
               ],
               "type": "Polygon"
            },
            "links": {
               "data": [
                  {
                     "href": "https://data.inpe.br/bdc/data/amazonia_wfi/2024_08/
AMAZONIA_1_WFI_RAW_2024_08_31.13_13_30_CB10/034_015_0/4_BC_LCC_WGS84/AMAZONIA_1_
WFI_20240831_034_015_L4_CMASK_GRID_SURFACE.tif",
                     "title": "CMASK",
                     "type": "image/tiff; application=geotiff; profile=cloud-
optimized"
                  },
                  {
                     "href": "https://data.inpe.br/bdc/data/amazonia_wfi/2024_08/
AMAZONIA_1_WFI_RAW_2024_08_31.13_13_30_CB10/034_015_0/4_BC_LCC_WGS84/AMAZONIA_1_
WFI_20240831_034_015_L4_BAND1_GRID_SURFACE.tif",
                     "title": "BAND1",
                     "type": "image/tiff; application=geotiff; profile=cloud-
optimized"
                  },
                  {
                     "href": "https://data.inpe.br/bdc/data/amazonia_wfi/2024_08/
AMAZONIA_1_WFI_RAW_2024_08_31.13_13_30_CB10/034_015_0/4_BC_LCC_WGS84/AMAZONIA_1_
WFI_20240831_034_015_L4_BAND2_GRID_SURFACE.tif",
                     "title": "BAND2",
                     "type": "image/tiff; application=geotiff; profile=cloud-
optimized"
                  },
                  {
                     "href": "https://data.inpe.br/bdc/data/amazonia_wfi/2024_08/
AMAZONIA_1_WFI_RAW_2024_08_31.13_13_30_CB10/034_015_0/4_BC_LCC_WGS84/AMAZONIA_1_
WFI_20240831_034_015_L4_BAND3_GRID_SURFACE.tif",
                     "title": "BAND3",
                     "type": "image/tiff; application=geotiff; profile=cloud-
optimized"
                  },
                  {
                     "href": "https://data.inpe.br/bdc/data/amazonia_wfi/2024_08/
AMAZONIA_1_WFI_RAW_2024_08_31.13_13_30_CB10/034_015_0/4_BC_LCC_WGS84/AMAZONIA_1_
WFI_20240831_034_015_L4_BAND4_GRID_SURFACE.tif",
                     "title": "BAND4",
                     "type": "image/tiff; application=geotiff; profile=cloud-
optimized"
                  }
               ],
               "previews": [
                  {
                     "href": "https://data.inpe.br/bdc/data/amazonia_wfi/2024_08/
AMAZONIA_1_WFI_RAW_2024_08_31.13_13_30_CB10/034_015_0/4_BC_LCC_WGS84/AMAZONIA_1_
WFI_20240831_034_015.png",
                     "title": "Preview",
                     "type": "image/png"
                  }
               ]
            },
            "id": "did:web:emc.spacebel.be:collections:AMZ1-WFI-L4-SR-1:items:
AMAZONIA_1_WFI_20240831_034_015",
            "title": "AMAZONIA_1_WFI_20240831_034_015",
            "updated": "2024-09-05T19:03:04.887222Z"
```

```
         },
         "issuanceDate": "2024-09-12T06:47:50Z",
         "id": "did:web:emc.spacebel.be:collections:AMZ1-WFI-L4-SR-1:items:
AMAZONIA_1_WFI_20240831_034_015",
         "proof": {
            "created": "2024-09-12T06:47:50.334Z",
            "jws": "eyJhbGciOiJFUzI1NksiLCJjcml0IjpbImI2NCJdLCJiNjQiOmZhbHNlfQ..U
QgwqSuwxrEStUli7I7k8uEOY3yVPzv7QtLLfhQkjsY9gY1ZkPXaeAbKdQiM1AKBC3YP61dIYbdskUmy8D
t4yQ",
            "proofPurpose": "assertionMethod",
            "type": "EcdsaSecp256k1Signature2019",
            "verificationMethod": "did:web:emc.spacebel.be:organisations:br_
inpe#owner"
         },
         "type": [
            "VerifiableCredential",
            "Feature"
         ],
         "@context": [
            "https://www.w3.org/2018/credentials/v1",
            {
               "date": "dct:date",
               "gj": "https://purl.org/geojson/vocab#",
               "data": "iana:enclosure",
               "bbox": {
                  "@id": "gj:bbox",
                  "@container": "@list"
               },
               "coordinates": "gj:coordinates",
               "title": "dct:title",
               "Feature": "gj:Feature",
               "dct": "http://purl.org/dc/terms/",
               "previews": "iana:icon",
               "geometry": "gj:geometry",
               "iana": "http://www.iana.org/assignments/relation/",
               "links": {
                  "@id": "owc:links",
                  "@context": {
                     "@vocab": "http://www.iana.org/assignments/relation/",
                     "type": "atom:type"
                  }
               },
               "owc": "http://www.opengis.net/ont/owc/1.0/",
               "href": "@id",
               "id": "@id",
               "atom": "http://www.w3.org/2005/Atom/",
               "Polygon": "gj:Polygon",
               "updated": "dct:modified"
            }
         ],
         "issuer": "did:web:emc.spacebel.be:organisations:br_inpe"
      }
   ]
}
```


Listing D.50

D.9.5. Verify VP for EO resource

Verifier checks the signature of the Presentation with the public key of the Holder that it finds in the DID document of the Holder. In addition, the signature of the issuer of the included Verifiable Credential is checked as well.

```
# options are found in the VP.

# option 1 - default options
await didkit.verify_presentation(json.dumps(data), json.dumps({}))
```

Listing D.51

```
'{"checks":["proof"],"warnings":[],"errors":[]}'
```

Figure D.24

D.10. Interactions with Indy instance

D.10.1. Prepare keys and identifier for DID

Onboarding of organizations (data providers), could be done by creating a public DID for each of them. DID (e.g. for organizations) are stored on a ledger as Verinym (`NYM`), that is associated with the legal identify of an identity owner. In the demonstrator, an Indy instance is used.

An Indy NYM transaction includes an identifier (dest), an ED25519 verification key (verkey).

```
key = jwk.JWK.generate(kty='OKP', crv='Ed25519')
k = key.export(private_key=False)
k
```

Listing D.52

```
'{"crv":"Ed25519","kty":"OKP","x":"QxKGXXxgyos7i8BhKKBw6B36LkGJKTp1Xc5ERTb3VMA
```

Figure D.25

```
x = json.loads(k)['x']
x
```

Listing D.53

```
'QxKGXXxgyos7i8BhKKBw6B36LkGJKTp1Xc5ERTb3VMA'
```

Figure D.26

What is the encoding of the ``x" in the JWK JSON representation ? This is defined in https://datatracker.ietf.org/doc/html/rfc7517. The x and y coordinates are the base64url-encoded values shown.

```
# decode public key and check it is 32 bytes
content = base64.urlsafe_b64decode(x + '=' * (4 - len(x) % 4))
len(content)
```

Listing D.54

```
32
```

Figure D.27

```
# Re-encode as Base58 for use with Indy
pk58 = base58.b58encode(content)
pk58
```

Listing D.55

```
b'5WphZDsSXZ71GBVCEmvifNuHMYFpKunzNdPAnzpRKg95'
```

Figure D.28

```
verkey = pk58.decode(encoding="utf-8")

# According to the documentation, the did identifier `nym` has to be derived
from the verkey.
# Get first 16 bytes
key = base58.b58decode(verkey)
first16 = key[:16]

nym = base58.b58encode(first16)
nym
```

Listing D.56

```
b'9HNvypZzHip7mYqLV8YRuy'
```

Figure D.29

D.10.2. Register NYM

```
data = '{  }'
response = requests.post('https://issuer.ogc.secd.eu/ledger/register-nym?did=' +
nym.decode(encoding="utf-8") + '&alias=dlr&verkey=' + verkey , data = data,
    verify=True,
    headers={ 'Accept': 'application/json', 'X-API-KEY': 'e5fb814b388f3aa1d8a0c33
56d235f29' })

response.text
```

Listing D.57

```
'{"success": true}'
```

Figure D.30

```
data = json.loads(response.text)
jstr = json.dumps(data, indent=3)
md("```json\n" + jstr + "\n```\n")
```

Listing D.58

```
{
```

```
   "success": true
}
```

Listing D.59

D.10.3. Get VERKEY from ledger

```
# Get VERKEY from ledger

response = requests.get('https://issuer.ogc.secd.eu/ledger/did-verkey?did=' +
nym.decode(encoding="utf-8"),
    verify=True,
    headers={ 'Accept': 'application/json', 'X-API-KEY': 'e5fb814b388f3aa1d8a0c33
56d235f29' })

data = json.loads(response.text)
jstr = json.dumps(data, indent=3)
md("```json\n" + jstr + "\n```\n")
```

Listing D.60

```
{
   "verkey": "5WphZDsSXZ71GBVCEmvifNuHMYFpKunzNdPAnzpRKg95"
}
```

Listing D.61

```
# Get DID endpoint

response = requests.get('https://issuer.ogc.secd.eu/ledger/did-endpoint?did='+
nym.decode(encoding="utf-8") + '&endpoint_type=Profile',
    verify=True,
    headers={ 'Accept': 'application/json', 'X-API-KEY': 'e5fb814b388f3aa1d8a0c33
56d235f29' })

data = json.loads(response.text)
jstr = json.dumps(data, indent=3)
md("```json\n" + jstr + "\n```\n")
```

Listing D.62

```
{
   "endpoint": null
}
```

Listing D.63

```
# get role

# get NYM role from ledger

response = requests.get('https://issuer.ogc.secd.eu/ledger/get-nym-role?did=' +
nym.decode(encoding="utf-8"),
    verify=True,
    headers={ 'Accept': 'application/json', 'X-API-KEY': 'e5fb814b388f3aa1d8a0c33
56d235f29' })

data = json.loads(response.text)
jstr = json.dumps(data, indent=3)
```

```
md("```json\n" + jstr + "\n```\n")
```

Listing D.64

```
{
   "role": "USER"
}
```

Listing D.65

D.10.4. DID Document Assembly

The above information is sufficient to assemble a W3C DID document for the Indy DID that was just created. The process is explained in https://hackmd.io/@kdenhartog/S1eUS2BQw#DIDDoc-Assembly-Steps.

```
# https://issuer.ogc.secd.eu/api/doc

# Resolve DID

response = requests.get('https://issuer.ogc.secd.eu/resolver/resolve/did:sov:' +
nym.decode(encoding="utf-8"),
    verify=True,
    headers={ 'Accept': 'application/json', 'X-API-KEY': 'e5fb814b388f3aa1d8a0c33
56d235f29' })

data = json.loads(response.text)
jstr = json.dumps(data, indent=3)
md("```json\n" + jstr + "\n```\n")
```

Listing D.66

```
{
   "did_document": {
      "@context": [
         "https://www.w3.org/ns/did/v1"
      ],
      "id": "did:sov:9HNvypZzHip7mYqLV8YRuy",
      "verificationMethod": [
         {
            "id": "did:sov:9HNvypZzHip7mYqLV8YRuy#key-1",
            "type": "Ed25519VerificationKey2018",
            "controller": "did:sov:9HNvypZzHip7mYqLV8YRuy",
            "publicKeyBase58": "5WphZDsSXZ71GBVCEmvifNuHMYFpKunzNdPAnzpRKg95"
         }
      ],
      "authentication": [
         "did:sov:9HNvypZzHip7mYqLV8YRuy#key-1"
      ],
      "assertionMethod": [
         "did:sov:9HNvypZzHip7mYqLV8YRuy#key-1"
      ]
   },
   "metadata": {
      "resolver_type": "native",
      "resolver": "IndyDIDResolver",
      "retrieved_time": "2024-09-12T06:45:40Z",
      "duration": 31
   }
```

```
}
```

Listing D.67

E

ANNEX E (INFORMATIVE) EU SATCEN – ADDITIONAL USE CASE INFORMATION

ANNEX E
(INFORMATIVE)
EU SATCEN — ADDITIONAL USE CASE INFORMATION

E.1. Situation Report Use Case Workflow Visuals

E.1.1. Supersampling Reprojection

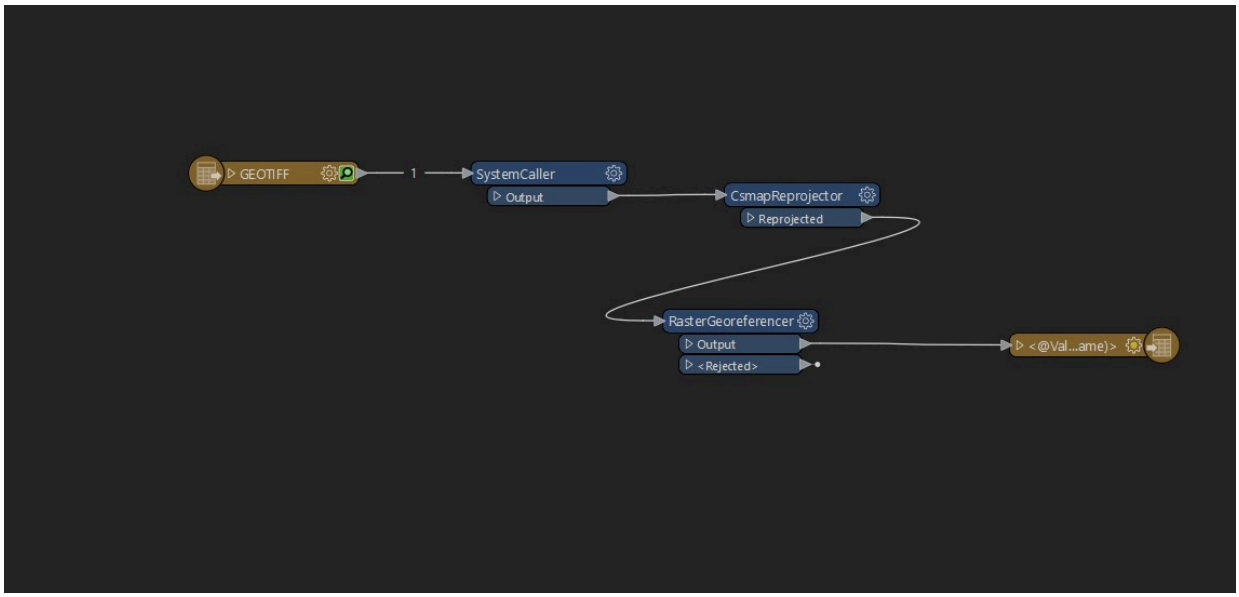


Figure E.1 — EU SatCen reprojection workflow

E.1.2. OSM Query Based on Image Bounding Box

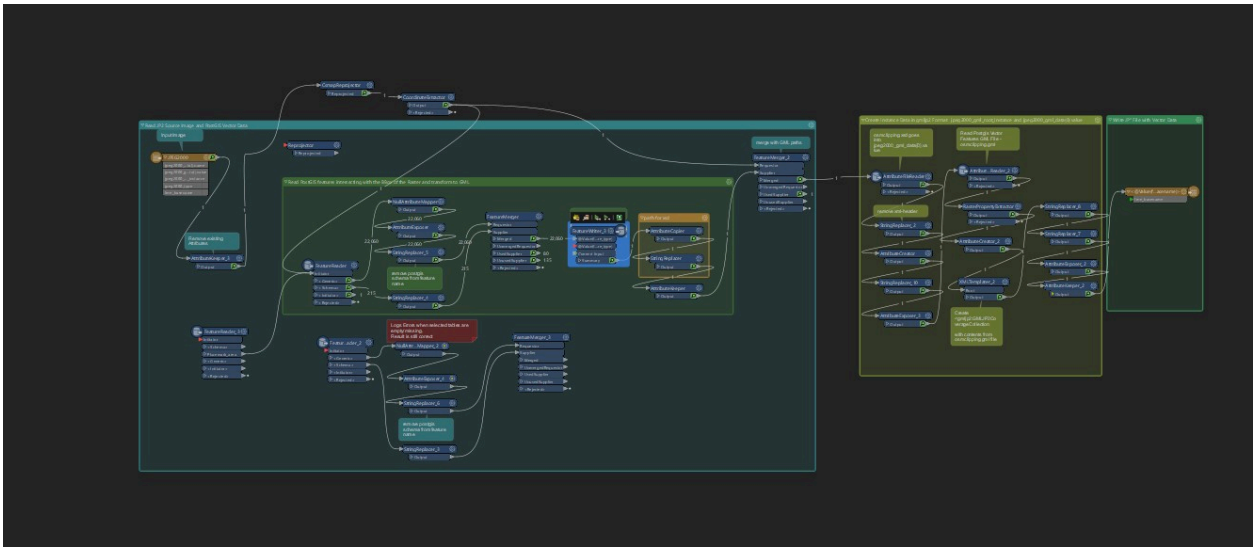


Figure E.2 — EU SatCen gmljp2 vector plus image workflow

E.1.3. Imagery and Vector Merging

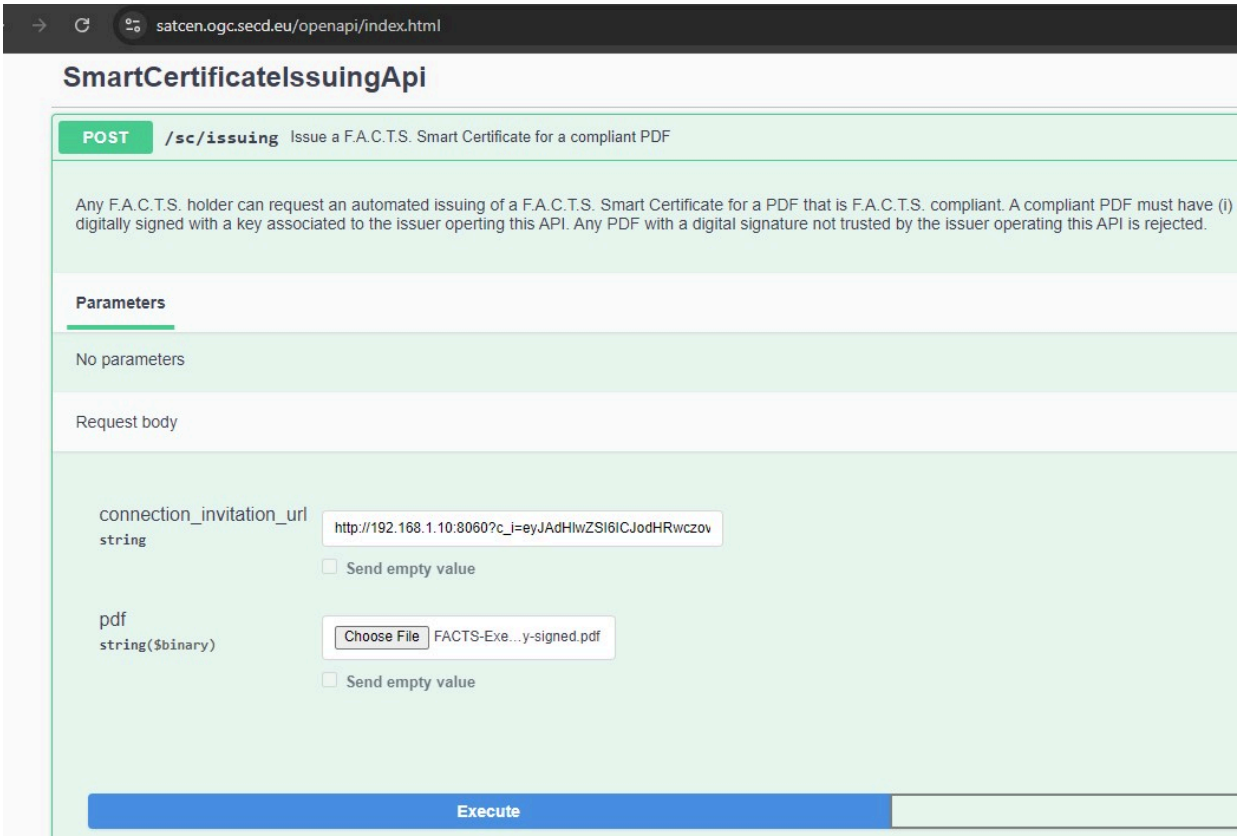


Figure E.3 — EU SatCen smart certificate issuer

E.1.4. FACTS Compliant Python Formatted PDF

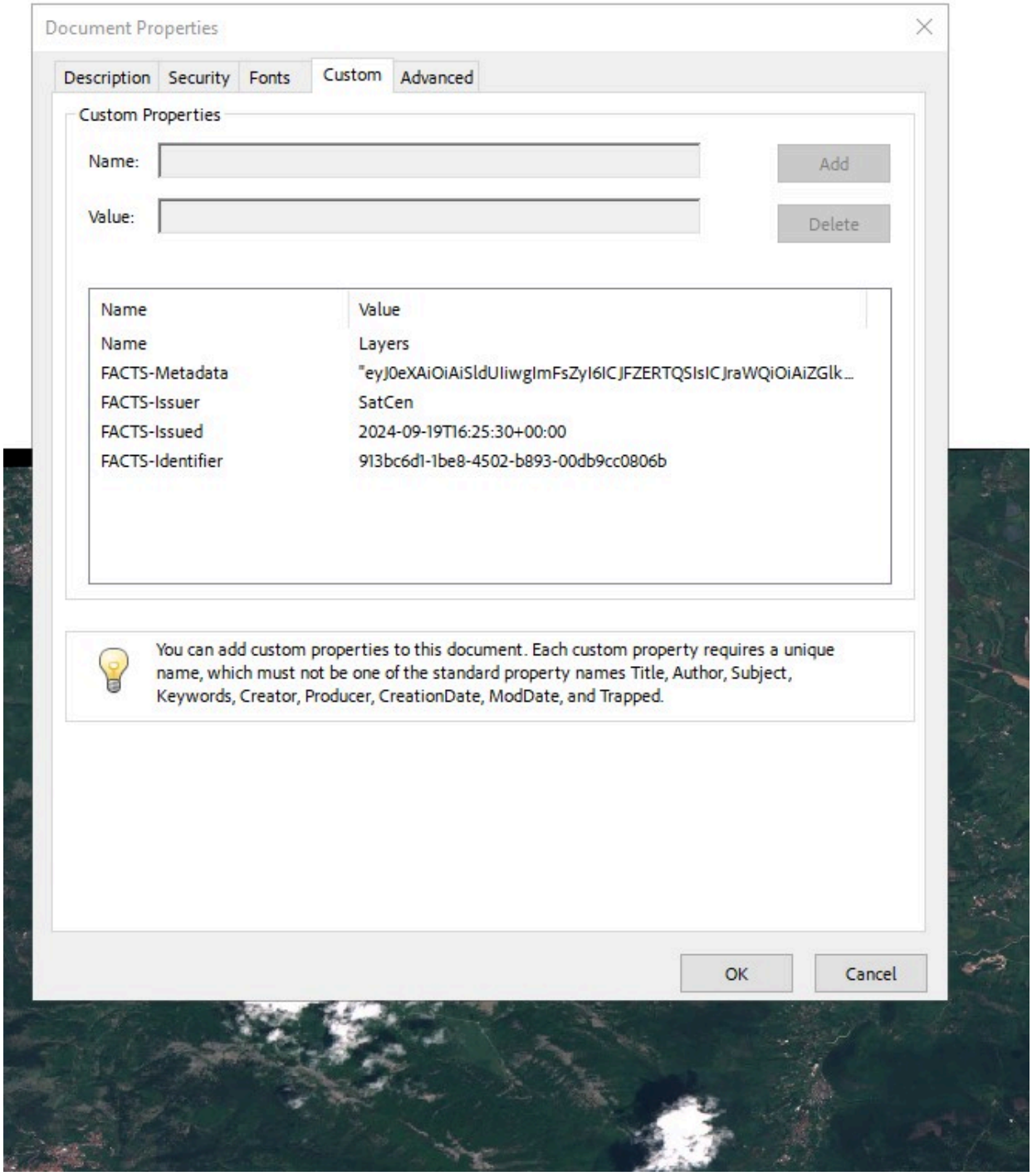


Figure E.4 — Document FACTS compliant

E.2. FACTS EU Sat-Cen Examples

```
<ipt:Metadata xmlns:ipt="http://www.www.www/ipt/1.0/gdb/1.0" schemaLocation=
"http:// www.www.www /ipt/1.0/gdb/1.0 GDB-IPT.xsd">
    <ipt:Identifier>c66d5c20-4c28-4959-b3f6-c3ad3e77003b</ipt:Identifier>
    <ipt:Publisher>Long John Silver</ipt:Publisher>
    <ipt:Published>2024-01-17T09:18:47Z</ipt:Published>
    <ipt:Created>1970-01-01T00:00:00Z</ipt:Created>
    <ipt:DataModel>my cool data model</ipt:DataModel>
    <ipt:FeatureCount>123456789</ipt:FeatureCount>
    <ipt:Aoi>Point(0 0)</ipt:Aoi>
    <ipt:Crs>CRS84</ipt:Crs>
    <ipt:License>CC-BY</ipt:License>
    <ipt:DataEncoding>application/x-gdb</ipt:DataEncoding>
    <ipt:MetadataEncoding>text/xml</ipt:MetadataEncoding>
    <ipt:Classification>ultra open</ipt:Classification>
    <ipt:HolderDID>FpsXsfj64R8N5gRYJjPdSE</ipt:HolderDID>
    <ipt:CredentialDefinition>3zC3MBQ31EV5Wom3UwUamj:3:CL:81:Vector-1.1</ipt:
CredentialDefinition>
</ipt:Metadata>
```

Listing

Schema to be stored into blockchain

```
{
  "attributes": [
    "publisher", "identifier", "gdb-hash", "xml-hash", "created", "published",
"classification", "aoi", "crs", "license", "data-encoding", "metadata-encoding",
"feature-count", "datamodel"
  ],
  "schema_name": "Vector",
  "schema_version": "1.1"
}

schema_id: 3zC3MBQ31EV5Wom3UwUamj:2:Vector:1.1

credential_definition_id: 3zC3MBQ31EV5Wom3UwUamj:3:CL:81:Vector-1.1
```

Listing

Verifiable credential example

```
{
  // set the context, which establishes the special terms we will be using
  // such as 'issuer' and 'alumniOf'.
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://www.w3.org/2018/credentials/examples/v1"
  ],
  // specify the identifier for the credential
  "id": "http://example.edu/credentials/1872",
  // the credential types, which declare what data to expect in the credential
  "type": ["VerifiableCredential", "AlumniCredential"],
  // the entity that issued the credential
  "issuer": "https://example.edu/issuers/565049",
  // when the credential was issued
  "issuanceDate": "2010-01-01T19:23:24Z",
  // claims about the subjects of the credential
```

```
  "credentialSubject": {
    // identifier for the only subject of the credential
    "id": "did:example:ebfeb1f712ebc6f1c276e12ec21",
    // assertion about the only subject of the credential
    "alumniOf": {
      "id": "did:example:c276e12ec21ebfeb1f712ebc6f1",
      "name": [{
        "value": "Example University",
        "lang": "en"
      }, {
        "value": "Exemple d'Université",
        "lang": "fr"
      }]
    }
  },
  // digital proof that makes the credential tamper-evident
  // see the NOTE at end of this section for more detail
  "proof": {
    // the cryptographic signature suite that was used to generate the signature
    "type": "RsaSignature2018",
    // the date the signature was created
    "created": "2017-06-18T21:19:10Z",
    // purpose of this proof
    "proofPurpose": "assertionMethod",
    // the identifier of the public key that can verify the signature
    "verificationMethod": "https://example.edu/issuers/565049#key-1",
    // the digital signature value
    "jws": "eyJhbGciOiJSUzI1NiIsImI2NCI6ZmFsc2UsImNyaXQiOlsiYjY0Il19..TCYt5X
      sITJX1CxPCT8yAV-TVkIEq_PbChOMqsLfRoPsnsgw5WEuts01mq-pQy7UJiN5mgRxD-WUc
      X16dUEMGlv50aqzpqh4Qktb3rk-BuQy72IFLOqV0G_zS245-kronKb78cPN25DGlcTwLtj
      PAYuNzVBAh4vGHSrQyHUdBBPM"
  }
}
```

Listing

E.3. Overview of Data Usage Conditions from ESA/NASA and EU Regulations

E.4. EU

The Council decision on the security rules for protecting EU classified information (EUCI) stipulates that communication and information systems need to handle EUCI in accordance with the concept of information assurance.

Information assurance in the field of communication and information systems is defined as the confidence that such systems will protect the information they handle and will function as they need to, when they need to, under the control of legitimate users. Effective information

assurance must ensure appropriate levels of confidentiality, integrity, availability, non-repudiation and authenticity.

E.5. ESA

Legal notice on the use of Copernicus Sentinel Data and Service Information

EU law grants free access to Copernicus Sentinel Data and Service Information for the purpose of the following use in so far as it is lawful:

- reproduction;
- distribution;
- communication to the public;
- adaptation, modification and combination with other data and information;
- any combination of previous points.

E.5.1. EU law allows for specific limitations of access and use in the rare cases of security concerns, protection of third party rights or risk of service disruption.

Copernicus Marine Service information

In application of the Regulation (EU) n° 1159/2013 of the 12 July 2013 supplementing Regulation (EU) n° 911/2010 of the European Parliament and of the Council on the European Earth monitoring programme, the

E.5.2. Licensee will communicate to the public the source of the products and services by crediting the Copernicus Marine Environment Monitoring Service;

E.6. NASA

The NASA ESDIS Project has a free and open policy for data and information generated under NASA sponsorship.

E.6.1. Non-NASA data is subject to the license arrangements of the sponsoring organization; users are encouraged to validate the source and associated use permissions.

The following is a statement with regard to the access of NASA-sponsored data managed by NASA ESDIS.

NASA ESDIS content, including but not limited to images, audio, video, and computer files used in the rendition of 3-dimensional models, such as texture maps and polygon data in any format, observation data, metadata, products, information, algorithms, including scientific source code, documentation, models, images, and research results, generally are not copyrighted. You may use this material for educational or informational purposes, including photo collections, textbooks, public exhibits, computer graphical simulations and Internet Web pages. This general permission extends to personal Web pages.

NASA ESDIS content used in a factual manner that does not suggest or imply endorsement may be used without explicit permission. NASA should be acknowledged as the source of the material.

ESDIS content that is subject to usage restrictions, such as a license agreement, shall be labeled as such and the use of that data shall be in accordance with the designated license.

E.6.2. Unless the content is marked with a use restriction or license, data provided from a NASA led mission are licensed as Creative Commons Zero (CC0).

There are no restrictions on the use of these data.

E.6.3. NASA occasionally uses copyrighted material by permission on its website. NASA's use does not convey any rights to others to use the same material. Those wishing to use copyrighted material must contact the copyright holder directly.

NASA does not license the use of NASA materials or sign licensing agreements. The agency generally has no objection to the reproduction and use of these materials, subject to the following conditions:

NASA material may not be used to suggest or imply endorsement by NASA or by any NASA employee of a commercial product, service, or activity, or used in any manner that might mislead. Please see NASA Advertising Guidelines and Merchandising Guidelines for more information. It is unlawful to falsely claim copyright or other rights in NASA material. NASA shall in no way be liable for any costs, expenses, claims, or demands arising out of the use of NASA material by a recipient or a recipient's distributees. NASA does not indemnify nor hold harmless users of NASA material, nor release such users from copyright infringement, nor grant exclusive use rights with respect to NASA material.

E.6.4. NASA material is not protected by copyright within the United States, unless noted.

E.6.5. If copyrighted, permission should be obtained from the copyright owner prior to use.

E.6.6. If not copyrighted, NASA material may be reproduced and distributed without further permission from NASA.

Citations and Acknowledgments

E.6.7. NASA should be acknowledged as the source of the material where applicable;

Users are encouraged to follow the following instructions, as guidance may vary depending on the originating source and scientific discipline.

NASA ESDIS has a free and open policy for data and information generated under NASA sponsorship. NASA data are freely accessible; however, when you publish these data or works based on the data, we request that you cite the datasets within the text of the publication and include a reference to them in your reference list. References to datasets should have enough detail to provide readers of your publication with the ability to obtain the datasets and conduct their own studies based on your work. For complete provenance and understanding of specifically which parts of data were used and how, it also may be necessary to describe in detail, within the body of the publication, exactly how the data were used.

Each discipline within Earth Science has its own unique approach to data. There are discipline-specific examples on how to cite and reference data and services. The links to this information are given below.

Alaska Satellite Facility (ASF) DAAC Atmospheric Science Data Center (ASDC) Crustal Dynamics Data Information System (CDDIS) Global Hydrometeorology Resource Center (GHRC) DAAC Goddard Earth Sciences Data and Information Services Center (GES DISC) Land Processes DAAC (LP.DAAC) Level 1 and Atmosphere Archive and Distribution System (LAADS) DAAC National Snow and Ice Data Center (NSIDC) DAAC Oak Ridge National Laboratory (ORNL) DAAC Ocean Biology DAAC (OB.DAAC) Physical Oceanography DAAC (PO.DAAC) Socioeconomic Data and Applications Center (SEDAC) In addition, the following cross-DAAC services provided by NASA ESDIS may be cited as shown below.

International Directory Network (IDN) Please refer to How to cite the International Directory Network for more information.

Land, Atmosphere Near real-time Capability for Earth Observations (LANCE) Please refer to LANCE Citation, Acknowledgements, and Disclaimer for more information.

Global Imagery Browse Services (GIBS) Please refer to the GIBS Data Use Policy and Acknowledgements for more information.

ESIP Guidelines for Referencing Data The Earth Science Information Partners (ESIP) provides a clear, concise guidelines document that may also help you in determining the format of the data set reference.

Open Data and the Importance of Data Citations In addition to providing free and open access to data, NASA's Earth Science Data Systems Program values transparency and reproducibility in scientific research, as do organizations with similar objectives such as the National Science Foundation (NSF), the Earth Science Information Partners (ESIP), the Coalition on Publishing Data in the Earth and Space Sciences (COPDESS), and the Global Earth Observation System of Systems (GEOSS).

As such, NASA recognizes the importance of authors using NASA-provided datasets to clearly indicate which datasets were used and provide access to these datasets to readers. While there are several ways of accomplishing this objective, citing datasets unambiguously is among the best.

NASA SMD will continue to take steps proactively to improve these open data and data citation policies to remain in line with the policies of our community. Our overarching objectives are to ensure that data from NASA's Earth Science Data Systems Program can easily be accessed and that research based on NASA-supported datasets clearly cites the sources of these data.