dispatch
Internet-Draft
Intended status: Experimental
Expires: 30 January 2026

N. Lukianets
Open Ethics Initiative

29 July 2025

Open Ethics Transparency Protocol
draft-lukianets-open-ethics-transparency-protocol-08

Abstract

The Open Ethics Transparency Protocol (OETP) is an application-level protocol for publishing and accessing ethical Disclosures of IT Products and their Components. The Protocol is based on HTTP exchange of information about the ethical "postures", provided in an open and standardized format. The scope of the Protocol covers Disclosures for systems such as Software as a Service (SaaS) Applications, Software Applications, Software Components, Application Programming Interfaces (API), Automated Decision-Making (ADM) systems, and systems using Artificial Intelligence (AI). OETP aims to bring more transparent, predictable, and safe environments for the end-users. The OETP Disclosure Schema is an extensible JSON-based format.

Table of Contents

1. Introduction

The Open Ethics Transparency Protocol (OETP or Protocol) describes the creation and exchange of voluntary ethics Disclosures for IT products. It is brought as a solution to increase the transparency of how IT products are built and deployed. This document provides details on how disclosures for data collection and data processing practice are formed, stored, validated, and exchanged in a standardized and open format.

OETP provides facilities for:

* *Informed consumer choices* : End-users able to make informed choices based on their own ethical preferences and product disclosure.

* *Industrial-scale monitoring* : Discovery of best and worst practices within market verticals, technology stacks, and product value offerings.

* *Legally-agnostic guidelines* : Suggestions for developers and product-owners, formulated in factual language, which are legally-agnostic and could be easily transformed into product requirements and safeguards.

* *Iterative improvement* : Digital products, specifically, the ones powered by artificial intelligence could receive nearly real-time feedback on how their performance and ethical posture could be improved to cover security, privacy, diversity, fairness, power balance, non-discrimination, and other requirements.

* *Labeling and certification* : Mapping to existing and future regulatory initiatives and standards.

The Open Ethics Transparency Protocol (OETP) is an application-level protocol for publishing and accessing ethical Disclosures of IT products and their components. The Protocol is based on HTTP exchange of information about the ethical "postures", provided in an open and standardized format. The scope of the Protocol covers

Disclosures for systems such as Software as a Service (SaaS) Applications, Software Applications, Software Components, Application Programming Interfaces (API), Automated Decision-Making (ADM) systems, and systems using Artificial Intelligence (AI). OETP aims to bring more transparent, predictable, and safe environments for the end-users. The OETP Disclosure Schema is an extensible JSON-based format.

2. Requirement Levels

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Terminology

Disclosure: Disclosure (Ethics Disclosure, or self-disclosure) is application-specific information about the data collection, data processing, and decision-making practices of a Product, provided by the Product Vendor (an individual developer or an organization).

Disclosure Feed: A historical sequence of Disclosures, made for a specific Product.

Disclosure Identity Provider: The automated Disclosure processing is enabled by requests to both the Open Ethics Disclosure database powered by Disclosure Identity Providers (DIP) and the Product’s OETP Disclosure file, stored in the product’s website root following OETP specification. DIP serves as a service point to generate and retrieve generated disclosures.

OETP Disclosure Schema: A predefined structure for Disclosure, supplied in the form of the JSON schema. The current version of the JSON schema is available on GitHub https://github.com/OpenEthicsAI/OETP/tree/main/schema (https://github.com/OpenEthicsAI/OETP/tree/main/schema) and is visualized on https://openethics.ai/oetp/ (https://openethics.ai/oetp/).

Transparency Manifest: A JSON file, typically named OETP.json or transparency.json, storing the Disclosure using the defined OETP Disclosure Schema.

Vendor: A legal person (an individual developer or an organization)

that owns one or several end-user Products, or acts as a Supplier and provides Components for other Vendors.

Integrator: A legal person (an individual developer or an organization) that deploys technology-powered services to the end-users based on Product(s) from third-party Vendors.

Product: An IT system in the form of software, software as a service system, application, software component, application programming interface, or a physically embodied automated decision-making agent.

Component: An IT system supplied by Vendor and integrated/embedded into end-user Products. Components themselves do not necessarily interface with end-users.

Upstream Component: A Component that sends its outputs to the Product Downstream in the data processing chain. Disclosure for the Upstream Component is represented as a Child relative to the Disclosure node of the Downstream Product.

Downstream Component: A Component that receives inputs from the Components Upstream in the data processing chain. Disclosure for the Downstream Component is represented as a Parent relative to the Disclosure node of the Upstream Component.

Automated Decision-Making (ADM): Automated decision-making is the process of making a decision by automated means without any human involvement. These decisions can be based on factual data, as well as on digitally created profiles or inferred data.

Validation: A sequence of automated software-based checks to control validity and security elements in the OETP Disclosure.

Auditor: A third-party legal person trusted to perform Verification checks and to issue Verification Proofs.

Auditing software: An automated software-based tool authorized to perform Verification checks and to issue Verification Proofs.

Verification: A procedure to control the correspondence of the elements in the OETP Disclosure and the actual data processing and data collection practices of the Vendors.

Verification Proof: A result of the formal Disclosure Verification procedure presented to a requestor.

Chaining: A process of combining Disclosures of individual

Components into a composite high-level Disclosure for a Product.

Label: User-facing graphical illustrations and textual descriptions of the Product that facilitate understanding of the values and risks the Product carries.

4. Protocol Model

The Disclosure creation and delivery consist of the two parts, starting from (I) the submission of the Disclosure form, chaining of the Suppliers' Disclosures, Signature of the disclosed information, and the delivery part (II) that first checks that the Disclosure is Valid, and then that the information specified in it is Verified by the third-parties. Figure 4 shows disclosure creation steps.

4.1. Creation of Disclosure

The initial Disclosure is created by filling out a standardized disclosure form (for example, see 1. https://openethics.ai/label/ (https://openethics.ai/label/)). A Vendor representative, a Product Owner, or a Developer, MUST submit data-processing and data-collection information about the Product. The information about the end-point URL, as well as a contact email address, MUST be specified. Disclosure MAY also be created in a fully automated way as a part of the CI/CD DevOps pipeline. Figure 5 shows basic disclosure submission process.

4.1.1. Cryptographic Signature

The Disclosure is organized into a predefined data schema and MUST be cryptographically signed by the Signature Generator (Open Ethics or federated providers) using standard SHA3-512 hash implementation. The integrity hash MUST be appended to a disclosure as the OETP.schema.integrity element.

4.1.2. Immutable Storage

Both the signature integrity hash and the Disclosure SHOULD be stored in the log-centric root database and MAY be mirrored by other distributed databases for redundancy and safety.

4.1.3. Visual Labeling

Open Ethics Label SHOULD be automatically generated by mirroring the submitted Disclosure into a set of graphical icons and simple human-readable descriptions. Additional Labels MAY be generated following successful third-party Verification and by mapping the regulatory requirements to Verified Disclosures.

4.2. Access to Disclosure

4.2.1. Initial Request to a Disclosure file

The most recent OETP file SHOULD be stored in the root of the Product's specified end-point URL, allowing requests to the OETP file from third-party domains. When establishing a Vendor relationship, the Integrator or a downstream Vendor MAY examine the Disclosure for their Components using the following HTTP request: GET https://testexample.com/oetp.json, where _testexample.com_ is the URL of the Supplier's end-point.

4.2.2. Access to Visual Trust Labels

A Vendor SHOULD place a visual Label generated as a result of the Disclosure process in the Product informational materials (for example Marketing Materials, User Guides, Safety Instructions, Privacy Policy, Terms of Service, etc). The Label reflects the content of the Disclosure and SHOULD be displayed in any digital media by embedding a software widget. Visual labels in the print media SHOULD carry a visually distinguishable Integrity signature to enable manual Validation by the User.

4.2.3. Requirements for placement of Integrity Signature in Visual Label

* *Labels in the online digital media* MUST be generated automatically based on the content of the Disclosure and MUST contain a URL allowing to check the complete Integrity hash and explore more detailed information about the Disclosure.

* *Labels in the offline media* MUST be generated automatically based on the content of the Disclosure and should carry the first 10 digits of the corresponding Integrity hash.

4.2.4. Conformity assessment marks

Based on the Verification performed for the OETP Disclosure file, the labels MAY include Conformity assessment marks, Certification marks, as well as marks showing adherence to certain standards. These marks MAY be generated and displayed automatically based on the Verification Proofs.

4.2.5. Accessibility considerations

Accessibility of the Labels for the visually impaired Users SHOULD be considered. The OETP Processing system MUST provide alternative forms of the Label so that text-to-speech tools could be used to narrate the Label without the lost of meaning.

1) A Label MUST contain a title. Title could be either marked by the aria-label attribute for the narration software or be labeled by another content tag(s) present via aria-labelledby attribute, pointing to the ID(s) describing the label content.

```
<!-- Open Ethics Label snippet: visible content -->
<a href="https://openethics.ai/label" target="_blank">
    <div id="label" aria-label="Open Ethics Label">
        <img src="/src/images/Open Ethics logo.svg" class="oesign svg" height="30" alt
="Open Ethics Logo" role="img" aria-hidden="true">
        <!-- Dynamic Content of the Open Ethics Label goes here -->
    </div>
</a>
```

Figure 1: Example Label Snippet Content

2) Every icon that is present in the visual Label MUST contain a title, describing the property illustrated by the icon. A more extended description MAY be provided when necessary. The following patterns are suggested:

* Pattern for images embedded using SVG tags: <img> + role="img" + alt="[title text here]" OR <img> + role="img" + aria-label="[title text here]"

* Pattern for images embedded using IMG tags: <svg> + role="img" + <title> + <desc> + aria-labelledby="[ID]"

```
<svg width="16" height="16" viewBox="0 0 16 16" fill="none" xmlns="http://www.w3.org/2
000/svg" role="img" aria-labelledby="iconOpenSourceCodeTitle iconOpenSourceCodeDescriptio
n">
<title id="iconOpenSourceCodeTitle">Algorithms and code libraries: Open Source Code</t
itle>
<desc id="iconOpenSourceCodeDescription">This product/component has disclosed that it
mainly uses the Open Source Code as a part of its codebase.</desc>
<circle cx="8" cy="8" r="7" fill="#FFFFFF" class="branding-accent" />
<path d="M10.56 4.83221C9.70667 4.08053 8.64 3.75839 7.46667 3.97315C5.44 4.18792 3.84
 6.01342 3.84 8.16107V8.26845C3.84 8.37584 3.84 8.48322 3.84 8.5906C3.94666 9.98658 4.8 1
1.1678 6.08 11.9195C6.4 11.1678 6.61333 10.5235 6.82667 9.87919C6.18667 9.44966 5.86667 8
.80537 5.86667 8.05369C5.86667 7.51678 6.08 7.08725 6.4 6.7651C7.14667 6.01342 8.21333 5.
79866 9.06667 6.33557C9.81333 6.7651 10.1333 7.51678 10.1333 8.37584C10.0267 9.02013 9.70
667 9.55704 9.17333 9.87919C9.38667 10.5235 9.70667 11.1678 9.92 11.9195C10.4533 11.5973
10.9867 11.1678 11.4133 10.6309C12.0533 9.77181 12.2667 8.69798 12.16 7.62416C12.0533 6.5
5033 11.52 5.47651 10.56 4.83221Z" fill="#333333" class="branding-main" />
<path d="M8 0C3.62667 0 0 3.54362 0 7.94631C0 12.349 3.62667 16 8 16C12.3733 16 16 12.
4564 16 7.94631C16 3.43624 12.3733 0 8 0ZM9.92 12.8859C9.81333 12.8859 9.70667 12.9933 9.
49333 12.8859C9.38667 12.8859 9.28 12.7785 9.17333 12.5638L8.10667 9.77181C8.10667 9.5570
5 8.21333 9.34228 8.42667 9.2349C8.85333 9.02013 9.06667 8.80537 9.17333 8.37584C9.17333
8.05369 9.17333 7.73154 8.96 7.51678C8.74667 7.19463 8.42667 6.97987 8.10667 6.97987C7.46
667 6.97987 6.93333 7.30201 6.93333 7.94631C6.82667 8.48322 7.04 8.91275 7.57333 9.2349C7
.89333 9.34228 8 9.55705 7.89333 9.87919L6.82667 12.6711C6.72 12.8859 6.61333 12.8859 6.6
1333 12.9933C6.61333 12.9933 6.50667 12.9933 6.4 12.9933C6.29333 12.9933 6.29333 12.9933
6.18667 12.9933C5.01333 12.5638 4.05333 11.7047 3.52 10.5235C2.98667 9.55705 2.88 8.80537
 2.88 8.37584V8.26846C2.88 5.58389 4.8 3.43624 7.36 3.11409C10.1333 2.68456 12.5867 4.617
45 13.12 7.30201C13.5467 9.66443 12.2667 12.0268 9.92 12.8859Z" fill="#333333" class="bra
```

```
nding-main" />
   </svg>
```

Figure 2: Example of the SVG icon with ARIA attributes for Accessibility

4.3. Verification and Validation of Disclosure

4.3.1. Automated Disclosure processing

The automated Disclosure processing is enabled by requests to both the Open Ethics Disclosure database powered by Disclosure Identity Providers and the Product’s OETP Disclosure file.

To allow efficient decentralization and access to the disclosures of autonomous systems, such as AI systems powered by trained machine learning models, the vendor (or a developer) MUST send requests to a Disclosure Identity Provider. Disclosures MAY be resolved using URIs. To satisfy the mentioned requirements for disclosure RI, it is proposed in [OETP-RI] to use the following formats:

* oetp://<hash> - Here integrity <hash> is the SHA3-512 generated during the disclosure process.

* oetp://<component>@<alias>[:<disclosure>] - Here <component> is the ID assigned via Disclosure Identity Provider under its <alias> during the first disclosure.

* oetp://<domain>[:<disclosure>] - For verified domains (Domain Validation), disclosure could be accessed using product’s <domain> instead of <component>@<alias>.)

4.3.2. Validation of Vendor’s Disclosures

The OETP Processing system MUST compare integrity hashes in the Open Ethics Disclosure database and entries that arrive as a result of the Disclosure Request response.

4.3.3. Verification of Vendor’s Disclosures

Every disclosure SHOULD be checked for the existence of the external Verification from Auditors for the entire Disclosures or one of the Disclosure elements.

4.3.4. Progressive Verification

To raise a level of trust in a Disclosure, a Vendor MAY decide to opt-in for a third-party Disclosure Verification. OETP suggests a Progressive Verification scheme where multiple independent external Verification Proofs COULD be issued by third parties to confirm the information specified in the Disclosure.

The Progressive Verification applies to a whole Disclosure, or to specific elements of the Disclosure.

Figure 6 displays a general scheme for Disclosure requests and responses.

The following elements MAY serve as sources for various kinds of Verification proofs: * Qualified Auditor reports * Qualified Vendor of Auditing software tests * Certification Authority assessments * Conformity assessments * User Feedback * Market Brokers * Real-time Loggers

4.4. End-to-end transparency and formation of the composite Disclosure

The IT industry is getting more mature with Vendors becoming more specialized. Surface-level transparency is not sufficient as supply chains are becoming more complex and distributed across various Components. The following steps MUST be satisfied for the end-to-end transparency:

4.4.1. Open Supplier Policy

Every Integrator or a Vendor SHOULD disclose the information about their Suppliers (sub-processing Vendors), indicating the scope of the data processing in the Components they provide.

If the Supplier information is not provided, Disclosure SHOULD contain information that a Vendor (Integrator) has not provided Supplier information.

4.4.1.1. First-party Components

For greater transparency, Vendors may decide to reveal Components even if they originate from themselves (first-party Components). For the first-party Component, the Supplier identity information SHOULD NOT be provided because it was already disclosed earlier.

Required: (Section 4.4.1.3.2) only

4.4.1.2. Third-party Components

When disclosing Components originating from the third-party Vendors SHOULD provide both the Supplier identity information and Component information

Required: (Section 4.4.1.3.1, Section 4.4.1.3.2)

4.4.1.3. Elements of Supplier disclosure

4.4.1.3.1. Supplier identity

* Vendor Name
* Vendor URL
* Vendor ID
* Vendor DPO Contact Email

4.4.1.3.2. Component information

* Component Scope of use
* Personal Data Being Processed by Component
* Is a Safety Component (YES)/(NO)
* Component URL (if different from the Vendor URL)
* Component Disclosure URL (if different from the default Component URL/oetp.json)
* Component DPO Contact (if different from Vendor DPO Contact Email)

4.4.2. Request for Supplier’s Disclosures

The OETP Processing system MUST send GET requests to the URLs of each Component to obtain their Disclosures. Based on the response to each Disclosure request, the OETP Processing system MUST specify which Components have Disclosures and which don’t have Disclosures.

Figure 7 shows the process of how Disclosure Chaining requests and responses happen.

4.4.3. Disclosure Chaining

The same Request-response operation applies recursively for Components of the Components, for the Components of the Components of the Components, etc. It is proposed to view the supply chain as a tree-like hierarchical data structure, where the information about Components is assembled using Level Order Tree Traversal algorithm.

In this tree: * Node is a structure that contains the Component’s Disclosure; * Root is the top Node representing a Product’s Disclosure information; * Edge is the connection between one Node and another, representing the scope of the Data Processing by the Component.

Figure 8 displays the order of the Disclosure Chaining with Level Order Tree Traversal algorithm.

4.4.4. Generation of the Composite Disclosure

The current consensus from the user & developer community suggests that Composite Disclosure should follow The "Weakest Link" model. According to this model, the risk that the Product is carrying should not be considered any less than the risk for each of the Components. A similar approach in dealing with software licenses has been successful by allowing to generate Software Bills of Materials (SBOMs) by providing package information in the [SPDX] files.

Formally this approach could be illustrated with the use of a conjunction table for risk modeling (see Table 1). The Truth Table for Logical AND operator below takes one risk factor and evaluates risk outcomes as High (H) or Low (L) for hypothetical Disclosure options of the Product(P) and its Component(C).

Disclosed risk of P	Disclosed risk of C	Composite P & C
L	L	*L*
L	H	*H*
H	L	*H*
H	H	*H*

Table 1: Conjunction Table for Risk Modeling

Further evaluation of this approach is required.

5. Example OETP Disclosure File

```
{
    "schema": {
        "name": "Open Ethics Transparency Protocol",
        "version": "0.9.3 RFC",
        "integrity": "156d624b8f2dbea87128a2147f255842652475c5dc595c79f64c90c7ff486d59
007c3e18c993e3163395812e26b70ea70dfc413f7ca128869d115f12e5699bf2"
    },
    "snapshot": {
        "product": {
            "url": "testexample.com",
            "description": ""
        },
        "timestamp": 1608273946,
        "generator": {
            "name": "Open Ethics",
            "alias": "oe",
            "type": "root",
            "website": "https://openethics.ai"
        },
        "label": {
            "data": {
                "type": "open",
                "practice": ""
            },
            "source": {
                "type": "open",
                "practice": ""
            },
            "decision": {
                "type": "restricted",
                "practice": ""
            }
        }
    }
}
```


Figure 3: Example OETP Disclosure File

6. Security Considerations

6.1. Response content

OETP exchanges data using JSON [RFC8259] which is a lightweight data-interchange format. A JSON-based application can be attacked in multiple ways such as sending data in an improper format or embedding attack vectors in the data. It is important for any application using JSON format to validate the inputs before being processed. To mitigate this attack type, the JSON Key Profile is provided for OETP responses.

6.2. Spoofing

OETP Processors should be aware of the potential for spoofing attacks where the attacker publishes an OETP disclosure with the OETP.snapshot value from another product, or, perhaps with an outdated OETP.snapshot.label element. For example, an OETP Processor could suppress the display of falsified entries by comparing the snapshot integrity from the submission database and a calculated hash for the OETP.snapshot object. In that situation, the OETP Processor might also take steps to determine whether the disclosures originated from the same publisher require further investigation of the Disclosure Feed and alert the downstream OETP Processors.

6.3. Falsification

Dishonest or falsified Disclosures is a problem that is hard to address generally. The approach to it is public control and systematic checks. Vendors or user-facing applications and services could further raise the level of trust in their Disclosures by implementing programmatic control scoring mechanisms, as well as the external verification by trusted Auditors.

7. IANA Considerations

Disclosures MAY be resolved using their URIs. To allow this requirement, the oetp:// URI scheme should be registered with IANA.

8. Areas for Future Study

The following topics not addressed in this version of the document are possible areas for the future study:

* Extensibility of the OETP Disclosure Schema.
* Evaluate other methods of Generation of the Composite Disclosure based on the Disclosure Tree
* Disclosure Chaining mechanisms and various use-cases.
* Typical scenarios and templates for Disclosure submissions.
* Mapping of the regulatory requirements and future Disclosure elements.
* Standardizing Privacy Disclosure and PII data-collection practices.

* Enhancing Label accessibility with ARIA W3C Recommendation and other approaches.

* Use of the OETP Disclosure in the ADM explainability (XAI).

* Disclosure Schemas for families of "Generative AI" technologies such as Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), Conditional Variational Autoencoders (CVAEs), Attention Mechanisms, Transformer-based Models.

* Interoperability of the agentic disclosure and the Model Context Protocol (MCP)

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

9.2. Informative References

[OETP-RI] Open Ethics Initiative, "Resource Identifier Scheme for OETP", 2022, <https://github.com/OpenEthicsAI/OETP-RI-scheme>.

[RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/rfc/rfc8259>.

[SPDX] The Linux Foundation, "SPDX Specification Version 2.2", 2020, <https://spdx.dev/>.

Appendix A. Appendix

A.1. Figures

Diagrams could be built from code using the below *.puml files automatically using PlantUML (https://plantuml.com/).

A.1.1. Creation of Disclosure

```
@startuml

title Disclosure Creation Process

skinparam roundCorner 15

actor "Supplier A" as SA
actor "Supplier B" as SB
actor Vendor as V

component "Component A" as CA
component "Component B" as CB
file "Disclosure A" as DA
file "Disclosure B" as DB
file "Composite Disclosure" as D

V-right->(Creation):disclose
SA-up->CA
SB-up->CB
CA-up->DA
CB-up->DB
DA-up->(Chaining)
DB-up->(Chaining)
(Creation)->(Chaining)
(Chaining)->(Validation)
(Validation)->(Verification)
(Verification)->D
@enduml
```

Figure 4: Creation of the Disclosure

A.1.2. Basic Disclosure Submission

```
@startuml
title Basic Disclosure Submission

skinparam roundCorner 15
autonumber

actor Vendor
database "Disclosure Identity Provider" as ID
control "Signature Generator" as SIG
database "Federated Identity Provider" as DIS

Vendor -> ID: Request with Disclosure payload
ID -> ID: Validate input
ID -> SIG: Structured Data, Initialized

ID <-- SIG: SHA3-512 integrity hash
    group Distributed Identity Storage
DIS <-- SIG: SHA3-512 integrity hash
end
ID -> ID: Log OETP file and a corresponding intgrity hash
Vendor <-- ID: OETP Disclosure File
@enduml
```

Figure 5: Basic Disclosure Submission

A.1.3. Progressive Verification Scheme for Disclosures

```
@startuml

title Progressive Verification with multiple Auditors

skinparam roundCorner 15
autonumber
actor User
User -> Vendor: Disclosure Request
User <-- Vendor: OETP Disclosure File

database "Disclosure Identity Provider" as ID

User -> ID: Disclosure Validation and Verification Request

group Progressive Disclosure Verification
    ID -> ID: Retrieve and Compare Disclosure Integrity
    ID -> "Auditor 1": Disclosure Verification Request
    ID <-- "Auditor 1": Verification Proof 1
    ID -> "Auditor N": Disclosure Verification Request
    ID <-- "Auditor N": Verification Proof N
end

User <-- ID: Verification response

User -> Vendor: Service Request
User <-- Vendor: Service Response
@enduml
```

Figure 6: Progressive Verification Scheme for Disclosures

A.1.4. Disclosure Chaining: Request-Response

```
@startuml
title Disclosure Chaining: Request-Response


start
repeat
  :Request Component’s Disclosure;
    if (Disclosure Obtained?) then (yes)
      :Validate Disclosure;
      :Verify Disclosure;
      :Chain Disclosure;
      :Obtain list of Child Components;
      if (Supplier information exists?) then (yes)
        :Update Tree with (yet)
        Unchained Disclosures;
      else (no)
        #Gold:**Alert** "Vendor has not provided
        Supplier information";
      endif
    else (no)
      #pink:**Alert** "Vendor has not provided
      any Disclosure";
      stop
    endif
repeat while (Unchained Disclosures in the Disclosure Tree?) is (yes) not (no)
:**Generate**
Composite Disclosure;
#palegreen:**Display** Label for "Composite Disclosure";
stop

@enduml
```

Figure 7: Disclosure Chaining: Request-Response

A.1.5. Disclosure Chaining: Level Order Traversal

```
@startmindmap
title Disclosure Chaining: Level Order Traversal

skinparam roundCorner 15
* Root (Product)
        * 1 (Component)
                * 3 (Component)
                        * 7 (Component)
                * 4 (Component)
        * 2 (Component)
                * 5 (Component)
                        * 8 (Component)
                        * 9 (Component)
                * 6 (Component)
@endmindmap
```

Figure 8: Disclosure Chaining: Level Order Traversal

Acknowledgments

Part of this work related to Verification and Validation of Disclosure and Disclosure Chaining was supported by the H2020 Programme of the European Commission under Article 15 of Grant Agreement No. 951972 StandICT.eu 2023

The Open Ethics community and expert volunteers contributed with their valuable feedback, discussions, and comments. Thank you Ashley Duque Kienzle, Angela Kim, Ioannis Zempekakis, Karl Mdespacher, Ida Varoanec, Claudia Del Pozo, Joerg Buss, Mariia Kriuchok, Minhaaj Rehman, Oleksii Molchanovskyi, Roberta Barone, Phil Volkofsky and others.

Author’s Address

Nikita Lukainets
Open Ethics Initiative
Email: n.lukianets@openethics.ai