Intelligence Network & Secure Platform for Evidence Correlation and Transfer

D2.5 Reference Framework for Standardization of Evidence Representation and Exchange

Document Summary Information

Grant Agreement No	833276	**Acronym**	INSPECTr
Full Title	Reference Framework for Standardization of Evidence Representation and Exchange		
Start Date	01/09/2019	**Duration**	42 months
Project URL	inspectr-project.eu		
Deliverable	2.5		
Work Package	2		
Contractual due date	28/02/2023	**Actual submission date**	22/03/23
Nature	Report	**Dissemination Level**	PU
Lead Beneficiary	CNR		
Responsible Author	Fabrizio Turchi		
Contributions from	Fabrizio Turchi, Sozos Karageorgiou, Ray Genoe		

Revision history (including peer reviewing & quality control)

Version 0.1	Originated by: CNR (Fabrizio Turchi)	On 30/11/2022/
Version 0.2	Extended by: CNR (Fabrizio Turchi)	On 01/12/2022
Version 0.3	Extended by: CNR (Fabrizio Turchi)	On 08/12/2022
Version 0.4	Extended by: CNR (Fabrizio Turchi)	On 12/12/2022
Version 0.5	Extended by: CNR (Fabrizio Turchi)	On 15/12/2022
Version 0.6	Extended by: CNR (Fabrizio Turchi)	On 16/12/2022
Version 0.7	Extended by: CNR (Fabrizio Turchi)	On 18/12/2022
Version 0.8	Extended by: CNR (Fabrizio Turchi)	On 19/12/2022
Version 0.9	Integrated by CNR (Fabrizio Turchi)	On 09/01/2023
Version 1.0	Integrated by CNR (Fabrizio Turchi)	On 10/01/2023
Version 1.1	Amended by CNR (Fabrizio Turchi)	On 12/01/2023
Version 1.2	Peer Reviewed by eBOS (Sozos Karageorgiou)	On 28/02/2023
Version 1.3	Peer Reviewed by CCI (Ray Genoe)	On 28/02/2023
Version 1.4	Integrated by CNR (Fabrizio Turchi)	On 02/03/2023
Version 1.5	Peer Reviewed by eBOS (Sozos Karageorgiou)	On 06/03/2023
Version 1.6	Peer Reviewed by CCI (Ray Genoe)	On 20/03/2023
Version 1.7	Integrated by CNR (Fabrizio Turchi)	On 20/03/2023

Disclaimer

Copyright message

TABLE OF CONTENTS

List of Figures

List of Tables

Glossary of terms and abbreviations used

Acronym	Explanation
AS4 Gateway	On the message level the AS4 standard (established by the OASIS organisation) is used to create an envelope for the messages which can be transferred securely from gateway to gateway. The authenticity and data integrity are ensured with digital signatures and encryption. e-CODEX uses the WS-Security standard (also established by OASIS) to guarantee this security features.
CASE	The open-source Cyber-investigation Analysis Standard Expression (CASE) is a community-developed ontology designed to serve as a standard for interchange, interoperability, and analysis of investigative information in a broad range of cyber-investigation domains, including digital forensic science, incident response, counter-terrorism, criminal justice, forensic intelligence, and situational awareness.
CTF	Capture The Flags (CTFs) are a kind of computer security competition. Teams of competitors (or just individuals) are set up against each other in a test of computer security skills. You, or your team, have to go to the other team's base and steal their team flag and bring it back to your base for points.
ebMS	A communication-protocol neutral method for exchanging electronic business messages. It defines specific enveloping constructs supporting reliable, secure delivery of business information. Furthermore, the specification defines a flexible enveloping technique, permitting messages to contain payloads of any format type.
e-CODEX	The e-Justice Communication via Online Data Exchange. The e-CODEX system is a tool specifically designed to facilitate the cross-border electronic exchange of data in the area of judicial cooperation in civil and criminal matters.

EXIF	Exchangeable image file format, a standard that specifies the formats for images, sound, and ancillary tags used by digital cameras (including smartphones), scanners and other systems handling image and sound files recorded by digital cameras.
GPDR	The General Data Protection Regulation is a European Union regulation on data protection and privacy in the EU and the European Economic Area (EEA).
IRME	Information Request Management Engine used to compose and send request of investigative information.
JSNL-LD	JSON-LD is a lightweight Linked Data format. It is easy for humans to read and write. It is based on the already successful JSON format and provides a way to help JSON data interoperate at Web-scale. JSON-LD is an ideal data format for programming environments, REST Web services, and unstructured databases such as Apache CouchDB and MongoDB.
KDD	Knowledge Discovery in Database refers to a method of finding, transforming, and refining meaningful data and patterns from a raw database in order to be utilised in different domains or applications.
LEA	Law enforcement is the activity of some members of government who act in an organized manner to enforce the law by discovering, deterring, rehabilitating, or punishing people who violate the rules and norms governing that society [from Wikipedia]
UCO	Unified Cyber Ontology. A foundation for standardized information representation across the cyber security domain.

1. Introduction

This document – "Reference Framework for Standardization of Evidence Representation and Exchange" describes the outcome of the activities carried out in Subtasks 2.4.1 and 2.4.2 of Work Package 2 – INSPECTr Reference Framework for Standardization of Evidence Representation and Exchange (SERE) – as explained in the Description of Action (DoA) of the INSPECTr project (Grant agreement no 833276).
The main objective of Work Package (WP) 2 – INSPECTr Reference Framework – is to provide a reference framework to be implemented in the INSPECTr platform which will facilitate standard solutions for forensic investigations across Law Enforcement Agencies (LEAs) within the European Union (EU) and internationally.
The activities focused mainly on the use of UCO/CASE ontologies that have been designed to be a common language between tools/systems and organizations/countries. This have also included some extensions of the ontologies for INSPECTr to accommodate the full scope of cyber investigation information in the Living Labs.
Initially, after the development of the Use Cases (TERRO, CSAM and FRAUD) provided by the LEA representatives within the Consortium, the UCO/CASE ontologies were able to represent almost all information included, taking into account that the standard does not represent investigative activities that do not operate directly on a digital item. Nevertheless, the number and the variety of the data comprised in those Use Cases were limited, as partially illustrated in Section 3.5 (Mobile Forensic Tools). Therefore, in order to cover a wider range of digital artifacts, it has been decided to follow two additional steps to extend the forensic data set available for the Project:

- including seven images/XML reports (five mobile devices, one computer, one takeout from the cloud) from the Catch the Flag (CTF) forensic competitions organised by Cellebrite in 2021 and by Magnet Forensics in 2022 (see Section 3 – Data Set)
- including the image/XML report of a personal smartphone for a wider view of the different artifacts included in a mobile device (see Section 3.1 - Forensic Acquisition action)

Moreover, it has been opted to represent in UCO/CASE all the data included in these additional XML reports and transfer them to the Storage Elastic Search platform regardless if they had already a representation in the UCO/CASE ontologies (see Section 5 – UCO/CASE extensions).

These choices have produced a twofold benefit to the UCO/CASE ontologies and to the development of the Project:

- Improving the standard representation of the investigative information already included in the UCO/CASE ontologies.
- Extending the standard representation adding relevant Artefacts not taking into account yet by the UCO/CASE ontologies (this also thanks to the use of Artificial Intelligence processing, briefly mentioned in Section 6. – UCO/CASE representation of AI processing).

Part of Task 2.4, according to the Description of Action (DoA) included examples of use in Living Labs (LLs) and the associated benefits. However, throughout the lifetime of the INSPECTr project, no issues, with regard to the "common language" provided by the UCO/CASE ontologies, emerged from the LLs (see Section 7 - Examples of use in LLs and the associated benefits) meaning that this has been excluded from this deliverable. As regards the survey of technical issues, the LEA feedback is focused on technical and investigative issues related to:

- CMS (Case Management System, almost developed from scratch) and CORTEX (an Observable (i.e., elements of probative evidence) Analysis and Active Response Engine);
- GAD (Gadget): tools to process the elements of evidence, for instance the parsers developed to convert the XML report generated by forensic tools in UCO/CASE standard;
- CSAM, TERRO, FRAUD: referring to the Use Cases created by the LEAs to understand if the investigators were able to answer specific investigative queries by using the platform;
- GRELLI (Generic Reusable Embeddable Lightweight Widgets etc.): the widgets (table, word cloud, chart, map) developed within the context of the project;
- Survey on AI tools, such as Translation, Automatic Speech Recognition (ASR), Optical Character Recognition (OCR), Named-Entity Recognition (NER).

1.1. Mapping INSPECTr Outputs

The purpose of this section is to map INSPECTr Grant Agreement commitments, both within the formal Deliverable and Task description, against the project's respective outputs and work performed.

INSPECTr GA Component Title	*INSPECTr GA Component Outline*	*Respective Document Chapter(s)*	*Justification*
Deliverable D2.5 Reference Framework for Standardization of Evidence Representation and Exchange	Guidelines and best practices report for multi-jurisdictional and cross-border operations. Data treatment guidelines, Governance, and Provenance specifications and policy definitions for information exchanges, and Necessary UCO/CASE extensions. Examples of use in LLs and the associated Benefits, and lessons learned.	All components addressed throughout this Deliverable where relevant. No issues of LLs have been reported with regard to the standard representation.	Reference framework for standardisation of evidence representation. Data set in use. UCO/CASE extensions. UCO/CASE representation of AI processing

T2.4: Reference Framework for Standardization of Evidence Representation and Exchange. Consolidate outputs from T2.1–T2.3 in the Reference Framework for Standardization of Evidence Representation and Exchange and will support its application, evaluation and refinement through the LLs. **ST2.4.1** Provide a guide for and support orchestration of 'standard' solutions for forensic investigations across EU LEA, in line with requirements specified in WP1. **ST2.4.2** Governance, Provenance and Policies in Information Exchanges.	**a.** Specification for provenance in data integration into INSPECTr and UCO/CASE management system implementation. **b.** Definition of Information Exchange Policy for INSPECTr requirements and implement in CASE standard (data markings) to enable proper data protection enforcement. **c.** UCO/CASE extensions for INSPECTr to accommodate the full scope of cyber-investigation information in LLs.	All components addressed throughout this Deliverable where relevant.	The traceability is integrated into UCO/CASE using the InvestigativeAction and ProvenanceRecord representations. To ensure all analysis results are traceable to their source(s), UCO/CASE keeps track of when, where and who performed which actions and used which tools to perform investigative actions on data sources, and what was the result. UCO/CASE supports classification markings, which are adaptable to different needs, and permit marking both at the overall UCO/CASE bundle level and granular UCO/CASE object level. An example of representing Traffic Light Protocol and a specific Information Exchange Policy (IEF) to mark UCO/CASE data is presented in Casey E, Barnum S, Griffith R, Snyder J, van Beek H, Nelson A (2017). Moreover, in Section 2 Guidelines/Best practices for cross-border operations in multi-jurisdictional scenarios are described relying on

			the deliverables D2.2 (Legislative compliance relating to law- enforcement powers and evidence requirements) and D2.4 (e-CODEX infrastructure evaluation in the context of deployment in LLs). The UCO/CASE extension topic has been illustrated in Section 5 (UCO/CASE extensions) and Section 6 (UCO/CASE representation of AI processing).

Table 1: Adherence to INSPECTr GA Deliverable & Tasks Descriptions

1.2. Deliverable Overview and Report Structure

This deliverable, D2.5 "Reference Framework for Standardization of Evidence Representation and Exchange" describes guidelines and best practices for multi-jurisdictional and cross-border operations. In particular the document has been broken down as follows:

- Data treatment guidelines, Governance and Provenance specifications and policy definitions for information exchanges.
- The UCO/CASE ontologies and the relative model in use within the project platform.
- The necessary UCO/CASE extensions that have been introduced to meet the informative needs.
- The UCO/CASE representation of AI processing.

Section 2 - Cross-border operations in multi-jurisdictional scenarios, guidelines/best practices - has been divided into two parts, the first part is dedicated to the legal perspective, the second part is devoted to the technical perspective.

Section 3 - Data set is dedicated to the extension of the data set compared with the one illustrated in the deliverable D2.3 ("Reference digital forensics Domain Model"). The CTF competitions have been exploited to add relevant XML reports for testing both the parsers and the robustness of the Storage Elastic Search due to the size of the data.

Section 4 - Reference Framework for Standardization of Evidence Representation - details the UCO/CASE ontologies and the related model in use within the Project platform.

Section 5 - Case extension includes all the new classes added to the last release of the ontologies

(1.1.0) that had to be introduced to meet the informative needs of the features of the platform.

Section 6 – UCO/CASE representation of AI processing includes some special artifacts that are related to the AI processing results such as Data Mining, Stylometry and NLP. Even though currently these artifacts are not represented in UCO/CASE yet, there are proposals to include them in the ontology. This section is meant to indicate the most important AI processing taken into account within the Project.

2. Cross-border operations in multi-jurisdictional scenarios: guidelines/best practices.

2.1. Legal perspective

The INSPECTr platform is mainly composed by LEA nodes, and each node is bound by law in their actions, because any platform enabling multi-level and cross-border cooperation should take into account the applicable legal framework.
The deliverable D2.1 (Legislative compliance relating to law-enforcement powers and evidence requirements) provided the initial legislative compliance relating to law enforcement powers and evidence requirements. This initial legal framework is devoted to understanding the legal requirements for law enforcement powers and evidence requirements, i.e. which legal instruments are applicable to investigations and to acquiring evidence, what powers (and restrictions) do law enforcement have in investigating criminal procedures and sharing this with their colleagues across Europe, how do LEAs cooperate with each other and with other parties, which are the relevant data protection implications to be taken into account, etc.
National, European and international laws and regulations, bilateral agreements and multilateral agreements all play a role in regulating the gathering, analysis and exchange of digital evidence. In particular when it comes to exchanging digital evidence across borders, it depends on the countries involved which legal instrument needs to be used for mutual assistance.

According to the European Commission (the Commission), more than half of all criminal investigations today include a cross-border element therefore the Commission proposed new rules with the aim to make the exchange of digital evidence easier and faster for police and judicial authorities. There are two paths followed by the Commission: international negotiations and internal rules.
International negotiations aim to improve cooperation with third (non-EU) countries, including with the United States of America (USA). At this aim, the Commission proposed two sets of negotiations. The first is an agreement between the EU and the USA on cross-border access to digital evidence for judicial cooperation in criminal matters which aims at avoiding conflicting obligations for service providers between the EU and the USA[1]. The second is an authorisation to participate in negotiations on a second Additional Protocol to the Cybercrime Convention[2] which aims at more effective Mutual Legal Assistance[3], including for example direct cooperation

[1] In 2018, the USA enacted the Clarifying Lawful Overseas Use of Data (CLOUD) Act in order to improve procedures for investigators in obtaining access to electronic information held by service providers. This Act allows LEAs in the USA in certain cases access to extraterritorially located data, meaning that European companies may fall within the scope of the CLOUD Act. This Act was highly criticized, both within the USA and outside the USA, including by civil rights groups for reasons of fundamental rights. Council of the European Union, Brussels, 21 May 2019, 9114/19, available at <https://data.consilium.europa.eu/doc/document/ST-9114-2019-INIT/en/pdf>; Recommendation for a Council Decision authorising the opening of negotiations in view of an agreement between the European Union and the United States of America on cross-border access to electronic evidence for judicial cooperation in criminal matters, COM(2019) 70 final. See for the negotiating directive: <https://data.consilium.europa.eu/doc/document/ST-9666-2019-INIT/en/pdf>.

[2] Council of the European Union, Brussels, 21 May 2019, 9116/19, available at <https://data.consilium.europa.eu/doc/document/ST-9116-2019-INIT/en/pdf>; Recommendation for a Council Decision authorising the participation in negotiations on a second Additional Protocol to the Council of Europe Convention on Cybercrime (CETS No. 185), COM(2019) 71 final. See for the negotiating directive: <https://data.consilium.europa.eu/doc/document/ST-9664-2019-INIT/en/pdf>.

[3] Council Act of 29 May 2000 establishing in accordance with Article 34 of the Treaty on European Union the Convention

with service providers in other jurisdictions. These negotiations will be followed throughout the lifetime of the INSPECTr project and reported in the deliverable D2.2 "Provide updates on regulatory developments and support project work streams" whose aim are a) to provide a periodic review of regulatory developments in Europe and United States. and b) to support legal issues and requirements arising from the Living Labs' scenario interactions the final legislative compliance report.

For improving the internal rules to make cross-border evidence gathering within the EU easier and faster, the Commission proposed a Regulation and a Directive[4] for the creation of a European Production and Preservation Orders for digital evidence in criminal matters as well as harmonised rules for legal representatives for gathering evidence in criminal proceedings. These new legal instruments will not replace the European Investigation Order (EIO) Directive[5], but will provide an additional tool for authorities.

These tools are considered to be necessary due to the fact that network-based services can be provided from anywhere in the world.
As such, the investigating authority needs to request the Member State where the service provider is based for mutual assistance. In view of the growing number of digital evidence, these requests through the official channels can take a long time. Combining this with the lack of a clear framework for cooperation with service providers makes it challenging for service providers to comply with LEA requests, in particular LEAs from another country. The new Regulation will allow LEAs to approach the service providers directly, without the involvement of a judicial authority in another Member State.

The Directive will lay down harmonised rules, obliging service providers in the EU to designate at least one legal representative for the receipt of, compliance with and enforcement of production and preservation orders and any other orders issued in the context of gathering evidence in criminal proceedings. Having legal representatives means that LEAs will have a clear point of access to address service providers.
The proposals are currently at the stage of first reading by the European Parliament.

The EU legal instruments discussed above are legislation that is relevant to digital evidence and LEA powers when gathering and handling digital evidence in criminal proceedings.

on Mutual Assistance in Criminal Matters between the Member States of the European Union. <https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32000F0712(02)&from=EN>.

[4] European Production and Preservation Orders for electronic evidence in criminal proceedings and for the execution of custodial sentences following criminal proceedings, direct cross-border requests for e-evidence to service providers in another Member State. Not yet force, is likely to enter into force in 2023. Agreement on compromise text reached in January 2023. Council and European Parliament are expected to vote on the compromise text during 2023. See <https://eur-lex.europa.eu/resource.html?uri=cellar:639c80c9-4322-11e8-a9f4-01aa75ed71a1.0001.02/DOC_1&format=PDF>;
Annex at <https://eur-lex.europa.eu/resource.html?uri=cellar:639c80c9-4322-11e8-a9f4-01aa75ed71a1.0001.02/DOC_2&format=PDF>.

[5] European Investigation Order: cross-border gathering and transmission of evidence, it does not apply in Ireland and Denmark. See <https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32014L0041&from=EN>.

2.2. Technical perspective

The EU Regulation 2022/850[6] established the legal framework for the e-CODEX system and therefore the e-CODEX system officially has become the technological backbone of the digitalisation of EU judicial cooperation in both civil and criminal matters. It comprises a package of software products that allow for secure digital communication between courts, and between citizens and the courts, in particular enabling the secure exchange of judicial documents.

From a technological perspective, the e-CODEX is a decentralised network of access points established with every e-CODEX participant. Thus, e-CODEX does not replace existing 'back-end' systems in the Member, but rather, it interlinks national and European IT systems in the area of justice. Therefore, each e-CODEX participant has to set up its own access point to participate in the communication.

At present, the e-CODEX platform is used for communicating within the scope of four EU legal instruments: 1) European order for payment procedure; 2) European small claims procedure; 3) mutual recognition of financial penalties procedure; and 4) mutual legal assistance in criminal matters and European investigation order.

The products that are part of the e-CODEX system are available free of charge, and the underlying software (Domibus and Domibus Gateway) is offered under the EU public licence – EUPL v1.2, which means it is open source.

Article 5 of the new Regulation outlines the composition of the e-CODEX system, which will be composed of an e-CODEX access point and digital procedural standards. An e-CODEX access point will be composed of a gateway (software, based on a common set of protocols, enabling the secure exchange of information over a telecommunications network with other gateways using the same common set of protocols) and a connector (making it possible to link connected systems to the gateway and consisting of software, based on a common set of open protocols, enabling the structuring, logging and linking of messages, the verification of their integrity and authenticity, and the creation of time- linked evidence of the receipt of the exchanged messages).

For the INSPECTr Living Lag, the e-CODEX system interlinks the various INSPECTr nodes in order to create a network of independent nodes. Any data collection, aggregation, dissemination or interpretation is done inside these nodes. E-Codex assumes that any output of these nodes is compliant to all requirements set for the INSPECTr Living Labs (see deliverables D2.1 and D2.2).

As e-CODEX is completely content agnostic, it performs no checks on the INSPECTr requirement compliancy, nor does e-CODEX read or process any of the information the node output entails. Most likely such output shall be delivered to e-CODEX in encrypted format. E-CODEX just boxes, addresses and dispatches the (encrypted) output for secured delivery at its intended recipient.

[6] Regulation (EU) 2022/850 of the European Parliament and of the Council of 30 May 2022 on a computerised system for the cross-border electronic exchange of data in the area of judicial cooperation in civil and criminal matters (e-CODEX system), and amending Regulation (EU) 2018/1726. See at <https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32022R0850&from=EN>.

2.2.1. *Creating the pMode configuration files to enable the INSPECTr workflow*

The "pModes" are part of the configuration files created for every participating country or organization necessary to establish the Gateway connections. Those files are centrally created based on the configuration data received by the participants. The pModes contain data such as the public web address of the Gateway, the use-cases that are supported by the system and sender/recipient data. Additionally, the configuration includes public certificates of all participants.[7]
Although it is possible for INSPECTr participants to create the so called pModes within the project, it is strongly advised not to do so. Only if the INSPECTr partners are sure that during the proof of concept or in future run- time, their network will never show any overlap with networks in which e-CODEX partners operate, an INSPECTr specific set of pModes can be created.

Between e-CODEX and related initiatives, overlap has already occurred and has led to addressing and routing problems. Certain parameters need to be uniquely set in order for a gateway to determine which partner information is to be delivered.

To provide a very simple example a so called '*PartyID*' is allocated for all partners in e-CODEX collaborations. This *partyID* is to be unique across all participants. , There are French entities involved in the EU collaborations on Financial Penalties, iSupport (global child maintenance obligations), and eEDES (European Investigation Order) . In all of these a specific French authority has been given the *PartyID* 'FR'. The e-CODEX project team allocated 'FR' to the ANTAI organisation (French fine collecting agency). In eEDES, the European Commission allocated 'FR' to the Public Prosecution Office, and in iSupport16 the HCCH allocated 'FR' to central authority for maintenance obligations. Based on *PartyID* a sending gateway is now unable to determine which receiving FR gateway to address.

2.2.2. *e-CODEX building blocks*

All technical solutions are based on the principle of subsidiarity , e-CODEX doesn't change existing solutions or laws in the participating countries. Therefore, a circle of trust has been introduced. Participating countries accept the legal validity of documents, and of information on identity and signatures of other Member States. To cope with different legal systems the technical infrastructure includes a methodology for mutual equal interpretation of legal terms.

As the technical components are open source, they can be used, advanced and linked to any national system. The main benefits are increased security and reliability along with saving time in completing cross-border processes.

2.2.3. *AS4 Gateway*

The Gateway within e-CODEX is the building block that is responsible for the communication between participants of e-CODEX use cases. A gateway used for participating in e-CODEX must have the following standards implemented:

- ebMS 3.0 standard: Gateway interchange messages complying with the ebXML

[7] *See the e-CODEX official site for further details: https://www.e-codex.eu.*

standard. This standard defines the structure one message header must have to be understood among the e-CODEX infrastructure of participants.

- AS-4 messaging profile: this profile is part of the ebMS 3.0 standard and defines the structure one message must have.
- Non-repudiation: A gateway must ensure that one message must not be delivered twice successfully.
- Reliability: The gateway used implements reliability and "Quality of service" that can be configured since different e-CODEX use cases may have different reliability settings which are defined by the use-case owner.

Whereas it is defined in e-CODEX that any gateway solution fulfilling those requirements mentioned above can be used, e-CODEX implemented its own gateway solution. This gateway is the DOMIBUS Gateway which is now under maintenance of the Connecting Europe Facility (CEF) programme of the European Commission[8].

The DOMIBUS gateway[9] additionally offers features like:

- pMode (Processing Mode) configuration: the way ebXML messages are sent and processed between two Gateways is defined using PMode files. Those files contain configuration information on how the communication is set up in sense of use-case, involved parties, their technical communication parameters and their roles, security, business processes and reliability.
- Logging: The Logging module enables the administrator to configure log levels and choose the log medium to be used.
- Plugin interface: The plugin interface allows implementers to develop their own plugin(s) for communicate with the backend(s).

In order to facilitate the familiarisation process, e-CODEX has developed the e-CODEX *LabBox*. This *LabBox* offers a completely configured Gateway and Connector setup of up to nine instances of Gateway/Connector pairs. It comes with pre-defined *pMode*, certificates and message structures. In this 'sandbox setup' the administrators can build experience in connecting to new partners, configuring message flows etc.
The *LabBox* does not allow to integrate with back-end systems, so it cannot be used for the actual INSPECTr framework implementation. For that purpose, individual Gateway/Connector software needs to be installed for each participant. By training on the *LabBox*, the configuration and interconnection between instances should be less time consuming. e-CODEX *LabBox* also offers an API to transmit messages, and at the moment the platform uses the default *pModes* and security settings it offers.

To handle a request for information that a user (LEA) needs to transmit they log in the Information Request Management Engine (IRME). The IRME also has a backend and uses

[8] The Connecting Europe Facility (CEF) is a key EU funding instrument to promote growth, jobs and competitiveness through targeted infrastructure investment at European level. It supports the development of high performing, sustainable and efficiently interconnected trans-European networks in the fields of transport, energy and digital services. See at <https://commission.europa.eu/funding-tenders/find-funding/eu-funding-programmes/connecting-europe-facility_en>.
[9] Additional details at https://ec.europa.eu/digital-building-blocks/wikis/display/DIGITAL/Domibus.

MongoDB as a database to store outgoing and incoming information requests and their status. The IRME contacts a bespoke Publisher/Subscribe[10] middleware. Then the middleware takes care of transmitting this message to the *LabBox* that, in turn, transmits the message to the e-CODEX node recipient.

Summarising the technical perspective, the exchange is carried out by using the appropriate security levels, confidentiality, integrity and sender authentication, offered by e-CODEX as secure communication channel.

[10] A Publish/Subscribe messaging is an asynchronous service-to-service communication method used in serverless and microservices architectures. Basically, the Pub/Sub model involves: i) a publisher who sends a message, ii) a subscriber who receives the message via a message broker.

3. Data set

The original dataset, described in the deliverable D2.3 (Reference digital forensics Domain Model), has been extended by using the forensic images supplied by the Catch the Flag (CTF) organised by Cellebrite in 2021[11] and by Magnet Forensic in 2022. Capture The Flags are a kind of computer security competition. Teams of competitors (or just individuals) are set up against each other in a test of computer security skills. Participants have to go to the other team's base and steal their team flag and bring it back to your base for points. The competition is used as a learning tool for everyone that is interested in cybersecurity, and it can help sharpen the tools they have learned during their training. In these challenges, the contestant is usually asked to find a specific piece of text that may be hidden on the server, or behind a webpage, in an image etcetera.

The below table provides a brief description of the data set included in the project for testing both the XML parsers and the platform.

Origin	Type of image	XML report size (UFED PA v. 7.34)
Cellebrite CTF 2021	iPhone X (Beth's iPhone X)	567 MB
	Heisenberg Galaxy Note 10	135 MB
	iPhone X (Marsha iPhone X)	2,8 GB
	Marsha PC Windows	317 MB

Table 2: Forensic data set provided by Cellebrite for Catch the Flag competition 2021

Origin	Type of image	XML report size (IEF v. 5.8)
Magnet Forensic 2022	iPhone 8 (Jess's iPhone 8)	120,5 MB
	Google Pixel	82 MB
	Google Takeout	14,7 MB

Table 3: Forensic data set provided by Magnet Forensic for CTF competition 2022

All the provided data is fictitious, so there is no issue at all from the privacy point of view, because the aim of the CTFs is to is to test cybersecurity skills among the digital forensic community at large.
Despite the gathered dataset's size, about 300 GB, all the provided data are fictitious, so they do not represent any issue from the General Data Protection Regulation 2016/679 of the European Parliament and of the Council. Nevertheless, the content of these images and the related XML reports which are the input for the parsers, are not suitable to comprise the multitude of cases that can be encountered when processing real data.
Considering the legal obligation to comply with the regulations laid down by the GPDR we decided to use the personal smartphone of one person from the WP2 working group to carry out the forensic acquisition and extraction of its data to produce significant XML reports based

[11] https://cellebrite.com/en/part-5-and-thats-a-wrap-for-the-2021-capture-the-flag-ctf.

on real data. Therefore, the legal basis is consent (Art.6(1)(a), GDPR). All the technical/forensic operations to accomplish this goal are thoroughly described below along with the actions taken to respect the protection of individuals with regard to the processing of personal data.
The use of real data is also for improving the quality of the Reference digital forensics domain model (Reference Framework for Standardization of Evidence Representation and Exchange - SERE) referring to the task T2.2.

3.1. Forensic Acquisition action

Within the Forensic premises laboratory, located in Genoa (Italy), the physical acquisition of the personal smartphone has been carried out.
The smartphone model, a Huawei, POT-LX1 model, could allow a mobile data recovering with full integrity (physical acquisition) only by using the XRY (MSAB) tool that supports the acquisition of the chipset Kirin 710 with Android 10.0.
The acquisition has been accomplished, opening the smartphone's lid to have direct access to the chip set – Figure 1 and Figure 2 - and allows a direct connection to a computer with a special adapter and the cited forensic tool – Figure 3.
The output obtained by the forensic acquisition has been securely transferred on one of the internal Forensic Lab's servers (the server is accessible from external networks only through a two-factor authentication system)

Figure 1: Smartphone Huawei with the lid open

Figure 2: Smartphone Huawei zoom on model details

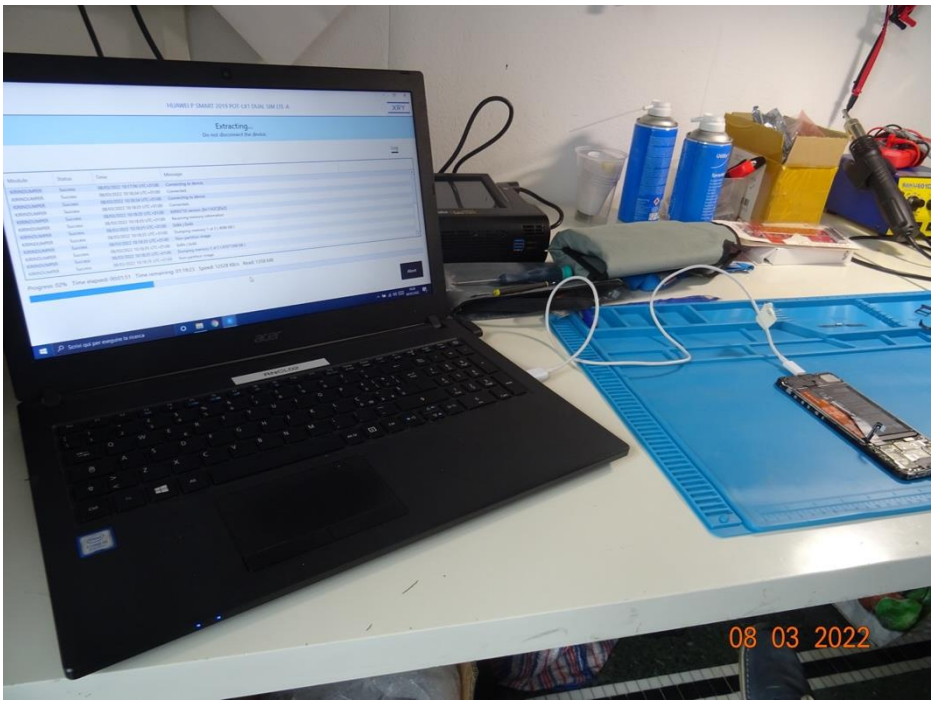

Figure 3: Smartphone Huawei connected to the PC with the XRY tool running

3.2. Forensic Extraction and conversion in UCO/CASE

With the aim to processing the Huawei smartphone data, the output obtained during the forensic acquisition, has been managed by using the XAMN tool (MSAB) to convert it from the original proprietary format (XRY) into a compatible format (tree view structure of the smartphone's data) that could be dealt with by the forensic extraction tools provided by Cellebrite, Oxygen Forensic and Magnet Forensic. The proprietary output of each forensic tool has been maintained on the server mentioned in the previous Section and used to generate, as

export, an XML report. The following XML reports have been generated as a result of this activity:

Report	*Size (in MB)*
Huawei_UFED.xml	1.300
Huawei_AXIOM.xml	511
Huawei_XRY.xml	1.230
Huawei_OXYGEN.xml	3.879

Table 4: XML reports size generated by processing the personal smartphone

The conversion of the XML reports, by using the UFED and AXIOM parsers developed so far, has produced the following files in JSON format, comply with the UCO/CASE:

JSON-LD file	*Size (in MB)*
UFED_Huawei_POT_LX1_Fabrizio.json	921
AXIOM_Huawei_POT_LX1_Fabrizio.json	282
XAMN_Huawei_POT_LX1_Fabrizio.json	1.600 MB
OXYGEN_Huawei_POT_LX1_Fabrizio.json	1.800 MB

Table 5: JSON-LD files size generated by processing the XML reports of the personal smartphone

These JSON files allowed meaningful tests relying on real data. Later the same operation has been extended to the parsers for XRY and OXYGEN, under development.
Figure 4 shows the cyber items represented in JSON UCO/CASE for the UFED report of the Huawei_POT_LX1 (personal smartphone). The testing phase is still in progress and will take some time due to the high number of data extracted and their overall variety. The result of this activity will be to improve the Reference Framework for Standardization of Evidence Representation and Exchange (SERE) and strengthen the quality of the commercial forensic tools parsers.

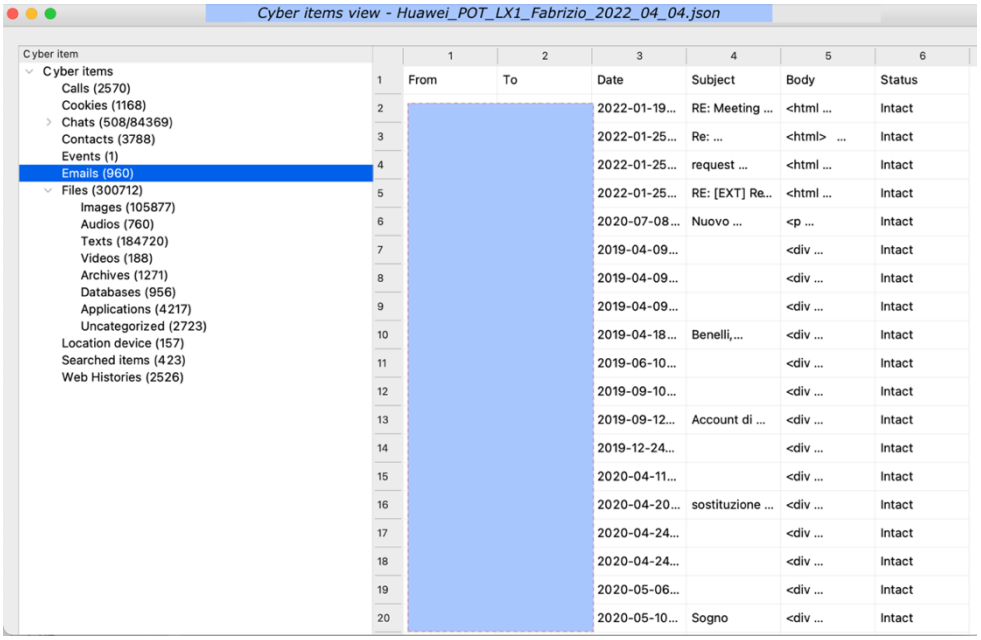

Figure 4: JSON view from the smartphone Huawei UFED XML report conversion

3.3. Final operations for removing personal data

Once all the XML reports have been generated, all the data (proprietary outputs of the forensic acquisition, proprietary outputs of the forensic extraction/processing and XML reports) temporary stored on the internal Forensic Lab's servers have been wiped using the Eraser tool[12], and the Pseudorandom data method. Therefore, the data has been overwritten with random data, indistinguishable from random noise.
The XML reports generated have been maintained on the computer of the mobile phone owner, in compliance with the GPDR and other related regulations.

3.4. Selected forensic tools

This document focuses on the commercial forensic tools, both for mobile devices and computer. The description of the model to cover the Traces/Cyber items involved relying on the analysis of the XML reports generated by the chosen forensic tools during the exporting process, a feature provided by each of the selected tools.

3.5. Mobile Forensic Tools

The mobile forensic tools have been selected on the basis of different criteria:

- survey provided within the project
- the direct experience of the digital forensic experts of the team responsible for this deliverable
- the availability of regular licenses for these kinds of tools, generally licenses are rather expensive
- on the basis of questionnaire, distributed to the potential users in other European projects
- some market analysis

Each acquisition of the forensic images described in Table 2 and Table 3 has been processed with the following four Mobile Forensics tools:

- UFED Physical Analyzer (v. 7.34 and 7.48)
- Oxygen Forensics Detective (v. 12.0 and 12.4)
- Magnet Axiom Process (v. 3.4, 3.8 and 4.01)
- MSAB XAMN (v. 4.4.0)

For each tool, two different reports have been created: a report in XML format and a report in the proprietary format. Figure 5 and Figure 6 show the test performed on UFED, AXIOM, XAMN and OXYGEN parsers using the CTF 2022 by MAGNET Forensics images:

[12] Eraser, version 6.2.0.2993, https://eraser.heidi.ie.

# Evidence	Report type (size)	UFED	AXIOM	XAMN	OXYGEN
E01	Jess_iPhone8 - UFED (175 MB) AXIOM (120 MB) XAMN (132 MB) OXYGEN (77 MB)	JSON size - 175 MB Observables: **96.031** **ACCOUNT** - 69 **CALL** - 2 **CELL_TOWER** - 1.696 **CHAT** - 16 (51) **COOKIE** - 138 **EMAIL** - 81 **EVENT** - 608 **FILE** - 40.032 **LOCATION** - 11.249 **SEARCH** - 82 **SMS** - 0 **WEB** - 2.003 **WIRELESS** - 3.285	JSON size 120 MB Observables: **36.742** **ACCOUNT** - 63 **CALL** - 9 **CELL_TOWER** - 1.767 **CHAT** - 3 (27) **COOKIE** - 0 **EMAIL** - 85 **EVENT** - 226 **FILE** - 17.101 **LOCATION** - 0 **SEARCH** - 20 **SMS** - 24 **WEB** - 2.928 **WIRELESS** - 0	JSON size - 132 MB Observables: **99.874** **ACCOUNT** - 15 **CALL** - 2 **CELL_TOWER** - 0 **CHAT** - 2 (27) **COOKIE** - 140 **EMAIL** - 0 **EVENT** - 608 **FILE** - 84.437 **LOCATION** - 9.570 **SEARCH** - 18 **SMS** - 13 **WEB** - 23 **WIRELESS** - 0	JSON size - 77 MB Observables: **72.242** **ACCOUNT** - 47 **CALL** - 1 **CELL_TOWER** - 0 **CHAT** - 10 (40) **COOKIE** - 0 **EMAIL** - 81 **EVENT** - 0 **FILE** - 67.931 **LOCATION** - 0 **SEARCH** --> **WEB** **SMS** - 0 **WEB** - 2.046 **WIRELESS** - 0
E02	Huawei_Fabri - UFED (1,2 GB) AXIOM (511 MB) XAMN (1,2 GB) OXYGEN (4 GB)	JSON size - 1,2 GB Observables: **416.880** **ACCOUNT** 3.788 **CALL** - 2.570 **CHAT** - 508 (84.369) **COOKIE** - 1.168 **EMAIL** - 960 **FILE** - 300.577 **LOCATION** - 157 **SEARCH** - 158 **SMS** - 423 **WEB** - 3.145	JSON size - 511 MB Observables: **187.648** **ACCOUNT** - 1.701 **CALL** - 149 **CHAT** - 513 (71.939) **COOKIE** - 906 **EMAIL** - 960 **FILE** - 102.997 **LOCATION** - 0 **SEARCH** - 265 **SMS** - 702 **WEB** - 810	JSON size 1,2 GB Observables: **860.162** **ACCOUNT** - 2.906 **CALL** - 2.776 **CHAT** - 439 (82.730) **COOKIE** - 945 **EMAIL** - 977 **FILE** - 762.228 **LOCATION** - 131 **SEARCH** - 72 **SMS** - 1.376 **WEB** - 844	JSON size 1,2 GB Observables: **860.162** **ACCOUNT** - 2.906 **CALL** - 2.776 **CHAT** - 439 (82.730) **COOKIE** - 945 **EMAIL** - 977 **FILE** - 762.228 **LOCATION** - 131 **SEARCH** - 72 **SMS** - 1.376 **WEB** - 844

Figure 5: CTF 2022 Use Case – Test (1/2)

# Evidence	Report type (size)	UFED	AXIOM	XAMN	OXYGEN
E03	GOOGLE_PIXEL AXIOM (82 MB)	no report generated	**ACCOUNT** - 1.609 / 253 **CALL** - 199 / 213 **CELL_TOWER** - 3.880 / 4.440 **CHAT** - 107 (13.501) / 41 (199) **EMAIL** - 265 / 362 **EVENT** - 8.807 /1.428 **FILE** - 248.229 / 129.997 **LOCATION** - 391.136 / 323 **SEARCHED_ITEM** - 264 / 307 **SMS** - / 501 **SOCIAL** - 727 / _ **WEB** - 4.939 / 481 **WIRELESS** - 77.365 / 77.365	no report generated	no report generated
E04	GOOGLE TAKEOUT AXIOM (14 MB)	no report generated	**ACCOUNT** - 75 **CHAT** - 27 (81) **COOKIE** - 1.553 **EVENT** - 71 **FILE** - 151.311 **SEARCHED_ITEM** - 289 **SMS** - 396 **WEB** - 3.392	no report generated	no report generated

Figure 6: CTF 2022 Use Case – Test (2/2)

Figure 7 and Figure 8 show the test related to the CSAM Use Case, prepared by the LEA representatives of the Project's Consortium

# Evidence	Device type	XML report - size	Observables
E11 ✓	Xiaomi / Samsung Galaxy J5	*CSAM-E11-UFED_File_System - 16 MB / CSAM-E11-UFED - 41 MB*	**ACCOUNT** - 64 / 603 **CALL** - 87 / 40 **CHAT** - 13 (2.251 MSG.) / 25 (839 MSG.) **COOKIE** - 1 / 17 **EMAIL** - 6 / 0 **FILE** - 2.988 / 14.081 **LOCATION** - 2 / 2 **SEARCHED** - 1 **URL** - 14
E12 ✓	SD	*CSAM-E12-AXIOM - 132 KB / CSAM-E12-UFED - 1 MB*	**FILE** - 121 / 98 **LOCATION**: 0 / 90
E13 ✓	Samsung JS logical	*CSAM-E13-UFED - 132 KB*	**ACCOUNT** - 50 **FILE** - 1
E15 ✓	HDD 80GB	*CSAM-E15-AXIOM - 804 MB / CSAM-E15-BIS-AXIOM - 893 MB*	**COOKIE** - 1.237 / 332 **EVENT** - 606 / 298 **FILE** - 51.708 / 47.482 **SEARCHED_ITEM** 539 / 23 **URL** - 2.099 / 935
E16 ✓	SD	*CSAM-E16-AXIOM - 467 KB*	**FILE** - 290
E21 ✓	V1 Viper S4G	*CSAM-E21-UFED - 96 MB*	**ACCOUNT** - 99 **CALL** - 22 **CHAT** - 15 (218 MSG.) **COOKIE** - 183 **EMAIL** - 18 **EVENT** - 17 **FILE** - 37.269 **LOCATION** - 8 **SEARCHED_ITEM** - 33 **SOCIAL** - 139 **URL** - 54

Figure 7: CSAM Use Case – Test (1/2)

# Evidence	Device type	XML report - size	Observables
E22 ✓	SIM logical	*CSAM-E22-UFED - 14 KB*	**ACCOUNT** - 2 **FILE** - 1 (Acquisition) **SIMData** not relevant
E31 ✓	Huawei P9 Lite	*CSAM-E31-UFED - 89 MB* *CSAM-E31-UFED_Android_Generic - 60 MB*	**ACCOUNT** - 409 / 38 **CALL** - 85 / 138 **CHAT** - 59 (813 msg.) / 10 (1.304) **COOKIE** - 740 / 106 **EMAIL** - 199 / 2 **EVENT** - 6 / 11 **FILE** - 23.789 / 17.291 **LOCATION** - 40 / 4 **SEARCHED_ITEM** - 158 / 8 **SMS** - 13 / 0 **SEARCHED** 0 / 8 **URL** - 2.378 / 12
E32 ✓	SIM	*CSAM-E32-UFED - 22 KB*	**ACCOUNT** - 5 **FILE** - 1 (Acquisition) **SIMData** no values
E41 ✓	Samsung (AL)	*CSAM-E41-UFED - 1,5 MB*	**ACCOUNT** - 17 **CALL** - 10 **FILE** - 471
E42 ✓	micro SD Samsung	*CSAM-E42-AXIOM - 288 KB*	**FILE** - 167
E43	SIM SD	*CSAM-E43-UFED - 13 KB*	empty only "SIMData" no values
E44	HD Lenovo	*CSAM-E44-AXIOM - 492 MB*	**COOKIE** -632 **EVENT** - 369 **FILE** - 30.434 **SEARCHED_ITEM** 148 **URL** - 1.072
E51 ✓	HDD Laptop P5	*CSAM-E51-AXIOM - 285 MB*	**COOKIE** - 734 **EVENT** - 583 **FILE** - 62.412 **SEARCHED_ITEM** - 136 **URL** - 2.588
E53 ✓	USB	*CSAM-E53-AXIOM - 43 KB*	**FILE** - 19

Figure 8: CSAM Use Case – Test (2/2)

Figure 9 shows the test related to the TERRO Use Case, prepared by the LEA representatives of the Project's Consortium.

# Evidence	Device type	XML report	Observables
E1 ✓	USB - 1GB	TERRO-E1-AXIOM - 40KB	**FILE** - 12
E2 ✓	HDD - Windows - 42 GB	TERRO-E2-AXIOM - 14MB	**FILE** - 2.651 **SEARCHED_ITEM** 35 **URL** - 10
E3 ✓	Samsung Phone	TERRO-E3-UFED - 1,3 MB	**CALL** - 3 **CONTACT** - 11 **FILE** - 492 **SMS** - 33 **URL** - 5

Figure 9: TERRO Use Case – Test

Figure 10 shows the test related to the FRAUD Use Case, prepared by the LEA representatives of the Project's Consortium.

# Evidence	Device type	XML report - size	Extracted Artifacts
E1			
E2 ✓	Windows HD 325 GB	R-FRAUD-E2-AXIOM - 347 MB	**COOKIE** - 298 **EVENT** - 22 **FILE** - 23.644 **SEARCHED_ITEM** 86 **URL** - 2.181
E3	iPhone SE (D79AP)	R-FRAUD-E3-UFED - 10,5 MB	**ACCOUNT** - 51 **CALL** - 16 **CHAT** - 3 (5) **COOKIE** - 85 **FILE** - 2.417 **LOCATION** - 3 **URL** - 4
E4			
E5	Android Nokia 3.4	R-FRAUD-E5-UFED - 1,9 MB	**ACCOUNT** - 39 **CALL** - 23 **COOKIE** - 2 **FILE** - 735
E9 ✓	Linux.Exx 153 GB	R-FRAUD-E9-AXIOM - 347 MB	**COOKIE** - 30 **FILE** - 25.164 **SEARCHED_ITEMS** - 5 **URL** - 22

Figure 10: FRAUD Use Case – Test

The next sections describe the data model derived by the UCO/CASE ontologies[13] to be included in the domain forensic model, considering the different kind of Cyber items or Observables, extracted by any forensic tool.

[13] UCO/CASE may be used as a data model but it is actually an ontology that is designed to be a common language between tools/systems and organizations/countries. As such, the most common approach to using CASE is to map an existing data model to the CASE standard, and then translate to/from the data model to CASE via import/export functions. During this mapping and implementation process, tool developers often find features of CASE that are useful to incorporate into their own data models.

4. Reference Framework for Standardization of Evidence Representation

The data model is derived from the Unified Cyber Ontology (UCO)[14] and the Cyber-investigation Analysis Standard Expression (CASE)[15]. CASE is a small piece of ontology that goes together with UCO and it is focused on data strictly connected with an investigation. CASE extends the Unified Cyber Ontology (UCO) construct. UCO provides a formalism for representing all cyber artifacts, almost any artifact that can be come across during an investigation.

UCO/CASE is a community-developed ontology designed to provide a standard for interoperability and analysis of investigative information in a broad range of cyber-investigation domains, including digital forensic, incident response, counter-terrorism, criminal justice, forensic intelligence.

The UCO/CASE Community is a consortium of for-profit, academic, government and law enforcement, and non-profit organisations. The consortium can guarantee the maintenance and the updating of the ontologies in the long term. The technological developments go by leaps and bounds and it is essential to keep updated with this progress to support a sustainability plan in the long term.

UCO/CASE uses Facets to represent various properties of the associated Observable Object. UCO/CASE uses the programming concept of 'duck typing', allowing an object to be enriched with any rational combination of Facets. Cyber-investigations can involve various kinds of data, including unexpected combinations of properties in a single object. UCO/CASE uses duck typing which allows data to be defined by its inherent characteristics rather than enforcing strict data typing. UCO/CASE objects can be assigned any rational combination of Facets, such as a file that is an image and a thumbnail. When employing this approach, data types are evaluated with the duck test, allowing data to be represented more truly without imposing a rigid class structure. Simply stated, if it walks like a duck, swims like a duck, quacks like a duck, and looks like a duck, then it probably is a duck. For certain common combinations of Facets, it is possible to assign them a higher-level class, such a PDF File or WhatsApp Message. This flexible approach is favoured over using the OWL concept of inheritance to define an object with various properties. Using inheritance requires permitted properties to be formally defined for each object type, which becomes un-wieldy when unexpected combinations of objects are encountered, such as one type of data embedded within another type of data that was not imagined when the ontology was designed.

UCO/CASE can be used as a data model but it is actually an ontology that is designed to be a common language between tools/systems and organizations/countries. Some visualization tools import and render UCO/CASE natively, effectively using it as a data model to combine information from various sources into a unified repository to strengthen correlation and analysis. This is a design choice, not a requirement for using UCO/CASE. Within the INSPECTr project UCO/CASE has been used a data model includes the most relevant Trace/Cyber items that is possible to extract from the source of evidence (i.e., mobile device, hard disks, USB pen drive, data on cloud) but also as the output of AI processing like machine learning.

[14] UCO is a community-developed ontology/model, which is intended to serve as a consistent foundation for standardised information representation across the cyber security domain/ecosystem. See at <https://unifiedcyberontology.org/>.

[15] CASE is a community-developed evolving standard that provides a structured (ontology-based) specification for representing information commonly analysed and exchanged by people and systems during investigations involving digital evidence. See at <https://caseontology.org/>.

CASE/UCO represent almost all types of information in cyber-investigations. The CASE/UCO community works together in an effort to keep pace with evolving types of information in investigations. Currently, UCO/CASE data representation includes data sources (mobile devices, storage media, memory), event details (browser history, logs), and well-known digital objects such as files and folders, messages (email, chat), documents (PDF, Word), multimedia (pictures, video, audio). Additional support being developed by the CASE/UCO community includes SQLite databases and Windows artifacts. In addition, by treating addresses, accounts, locations, identities, and other entities as nodes in a graph, UCO/CASE represents relationships between objects to support linked data analysis and automated correlation (similarity/repetition detection).

The main classes of the ontologies are depicted in Figure 11:

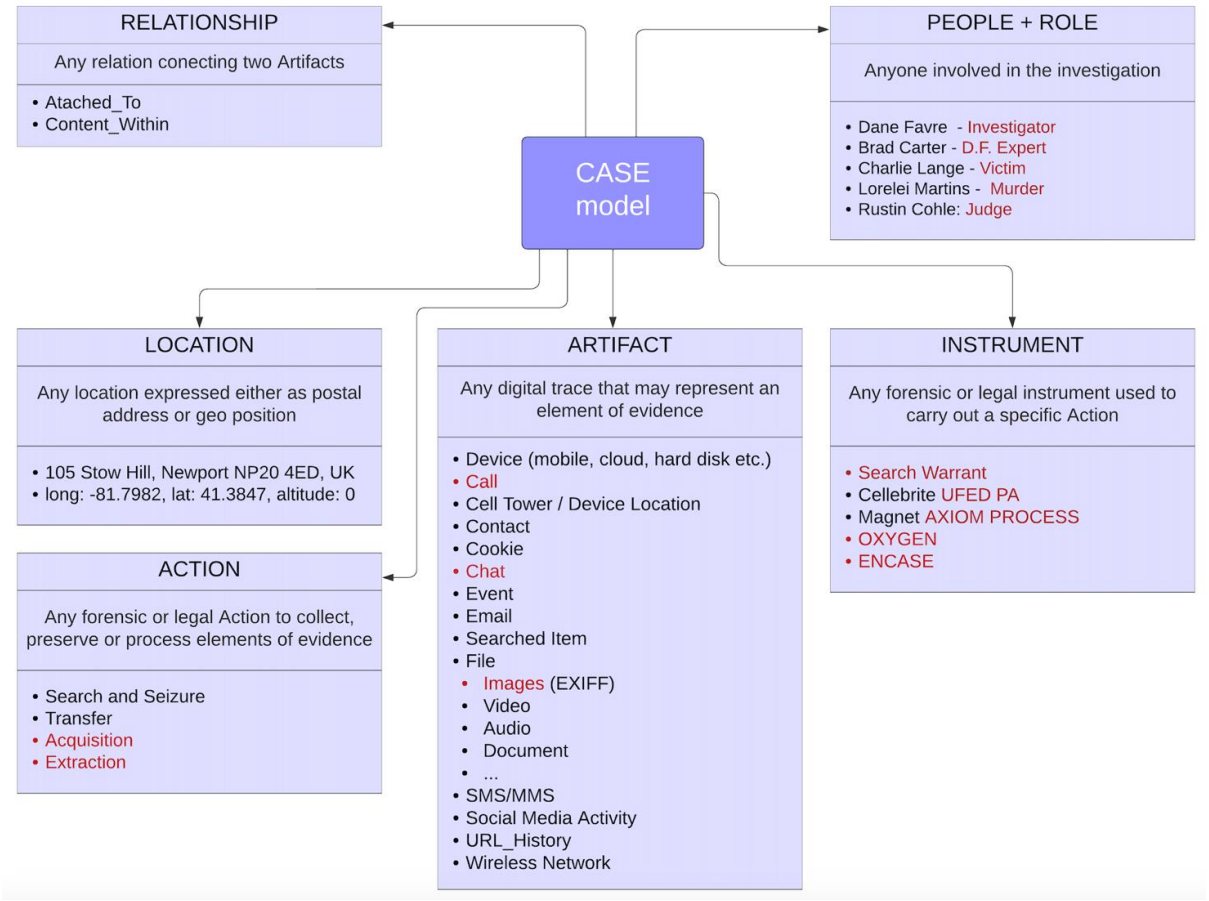

Figure 11: UCO/CASE main classes

The meaning of each class is briefly described below:

- People involved in the evidence life-cycle, from search and seizure to the report before the Court, technical and legal (subjects, victims, authorities, examiners etc.).
- Surrounding information about Legal authorization (i.e., search warrant).
- Information about the Process/Lifecycle (i.e. seizing, acquisition, analysis etc.).
- Information about the Chain of custody by identifying Who did What, When and Where from the moment the Evidence has been gathered.
- Acts performed by people (seizing, acquisition, analysis etc.).
- Source of evidence, that is physical objects involved in the investigative case (e.g. hard disk, smartphone) but even digital source of evidence (e.g. memory dumps).
- Description of the Objects inside the digital evidence and their Relationships (e.g., *Contained_Within*, *Extracted_From* etc.).

4.1. UCO/CASE serialisation

The community responsible for maintaining and updating UCO/CASE ontologies has chosen the JSON-LD[16] format as serialisation because it is possible to validate its compliance with the structures, properties and constraints defined within the ontologies.

Within a JSON-LD file, each Observable Object (henceforth referred to as Object) is assigned an identifier (@id) that can be used to refer to another Object in cases where a property of the first Object is not explicitly defined but refers to the content of a second Object. For example in Figure 12 where we illustrate the representation in JSON-LD of an Object of type *CallFacet* (i.e., telephone call) via the *Whatsapp* application, the property of the calling party (*uco-observable:from property*) is not expressed, as one would expect via a *Whatsapp* account, but with a reference, expressed with the *@id* property to another Object of type *AccountFacet* relative to an Object of type *ApplicationAccountFacet*. This reference is unique because it must represent an entity in an unambiguous manner.

Figure 12 shows a telephone call via *Whatsapp* application, but also includes two *ApplicationAccountFacet* and one *ApplicationFacet* to represent the application used.

[16] JSON-LD is a lightweight Linked Data format. It is easy for humans to read and write. It is based on the already successful JSON format and provides a way to help JSON data interoperate at Web-scale. JSON-LD is an ideal data format for programming environments, REST Web services, and unstructured databases such as Apache CouchDB and MongoDB. See at <https://json-ld.org/>.

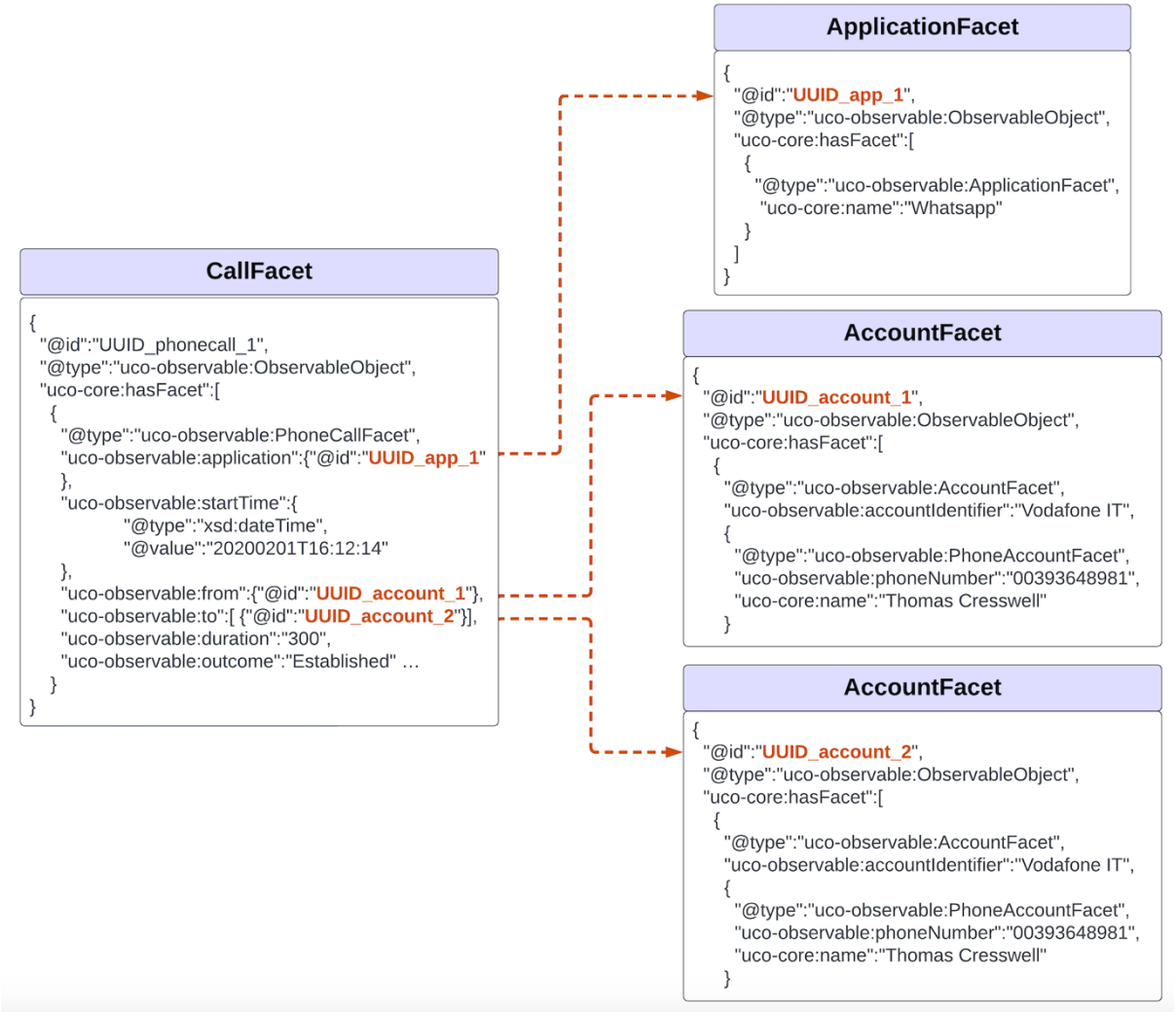

Figure 12: UCO/CASE representation of a CallFacet by Whatsapp

4.2. UCO/CASE model

The model is presented by using a set of images that illustrate the main ontology classes and an example of the corresponding JSON-LD serialization. The figures also include the kind of data (string, *xsd:dateTime, xsd:integer* etc.) for each class's property. A mandatory property/field is marked with an asterisk, according to the cardinality requested by the UCO/CASE ontologies, version 1.0.0, released on August 31st 2022.

4.2.1. Account

An account facet is a grouping of characteristics unique to an arrangement with an entity to enable and control the provision of some capability or service.

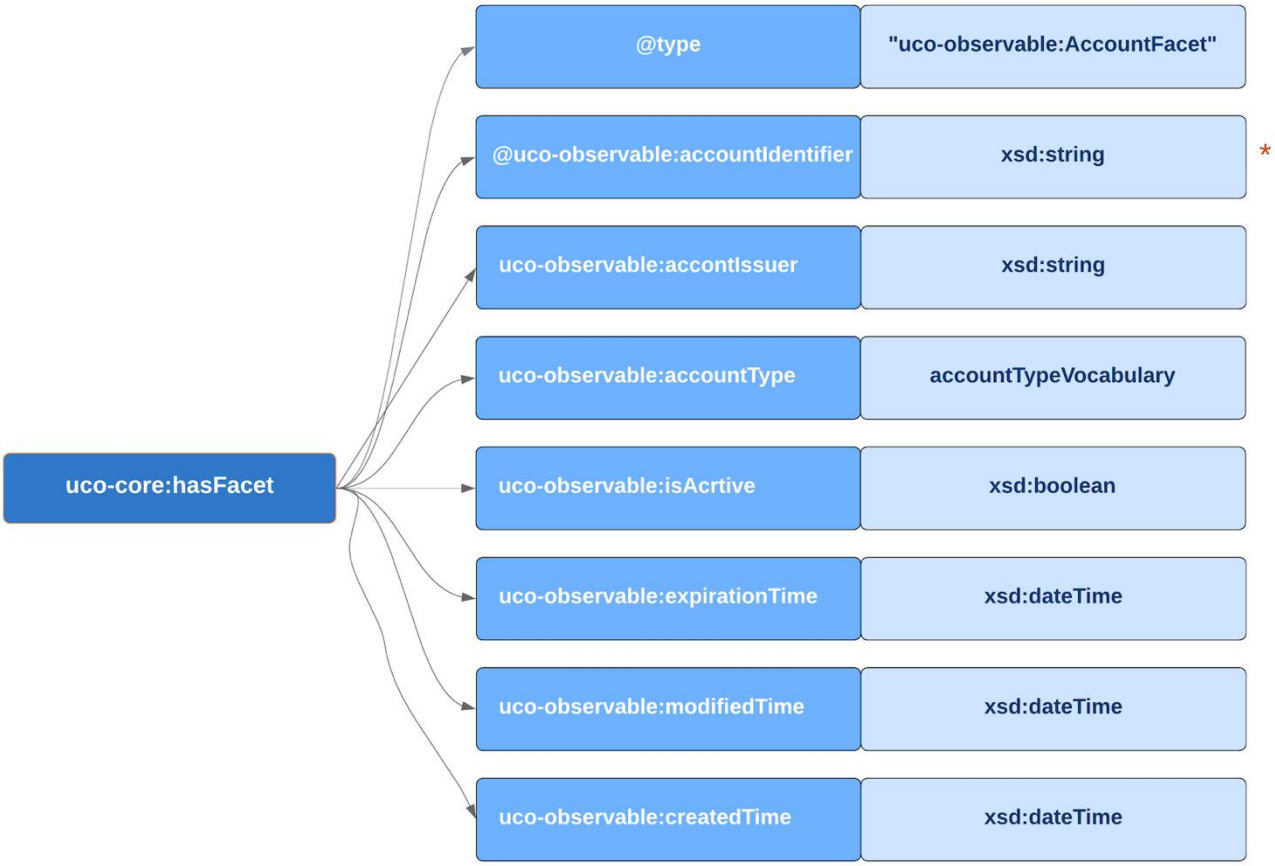

Figure 13: UCO/CASE Artifact Account

An example of JSON-LD serialisation for the *Account Observable* is as follows:

```
{
   "@id": "kb:uuid_whatsapp_org",
   "@type": "uco-identity:Organization",
   "uco-core:name": "Whatsapp"
 },
 {
   "@id": "kb:d3eabebc-bba6-49cf-8d5a-2af44a1ca389",
   "@type": "uco-observable:ObservableObject",
   "uco-core:hasFacet": [
     {
       "@type": "uco-observable:AccountFacet",
       "uco-observable:isActive": {
         "@type": "xsd:boolean",
```

```
            "@value": "True"
          },
          "uco-observable:accountIdentifier": "393457700255@s.whatsapp.net"
        },
         "uco-observable:accountIssuer": {
            "@id": "kb:uuid_whatsapp_org"
          }
        "uco-observable:accountType": {
          "@type": "uco-types:ControlledDictionary",
          "uco-types:entry": [
            {
              "@type": "uco-types:ControlledDictionaryEntry",
              "uco-types:key": "Type",
              "uco-types:value": "Application"
            }
          ]
        },
        {
          "@type": "uco-observable:ApplicationAccountFacet",
          "uco-observable:application": {
            "@id": "kb:af902343-4e5c-49f4-adf0-31da6fd18422"
          }
        },
        {
          "@type": "uco-observable:DigitalAccountFacet",
          "uco-observable:displayName": ""
        }
      ]
    }
```


An example of JSON-LD serialisation for the *Phone Account Observable* is as follows:


```
  {
     "@id": "kb:uuid_providere_org",
     "@type": "uco-identity:Organization",
     "uco-core:name": "Telecom"
   },
   {
     "@id": "kb:d3eabebc-bba6-49cf-8d5a-2af44a1ca389",
     "@type": "uco-observable:ObservableObject",
     "uco-core:hasFacet": [
       {
         "@type": "uco-observable:AccountFacet",
         "uco-observable:isActive": {
           "@type": "xsd:boolean",
           "@value": "True"
         },
         "uco-observable:accountIdentifier": "393457700255 "
       },
        "uco-observable:accountIssuer": {
           "@id": "kb:uuid_whatsapp_org"
```

```
      }
    "uco-observable:accountType": {
      "@type": "uco-types:ControlledDictionary",
      "uco-types:entry": [
        {
          "@type": "uco-types:ControlledDictionaryEntry",
          "uco-types:key": "Type",
          "uco-types:value": "Phone"
        }
      ]
    },
    {
      "@type": "uco-observable:PhoneAccountFacet",
      "uco-observable: phoneNumber": "393283623632"
     "uco-core:name": "Usko Bergdahl"
    },
  ]
}
```

4.2.2. *Application*

An application facet is a grouping of characteristics unique to a particular software program designed for ends users.

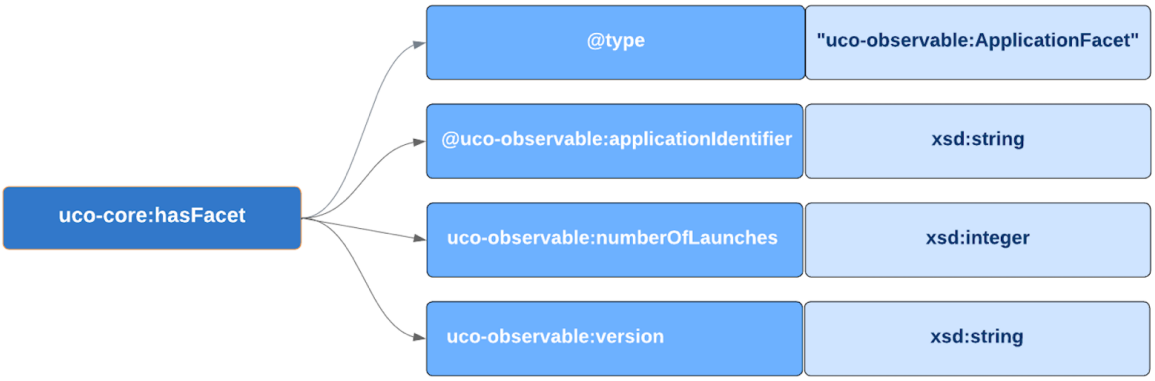

Figure 14: UCO/CASE Artifact Application

An example of JSON-LD serialisation for the *Application Account Observable* is as follows:

```
{
      "@id": "kb:9e4c3149-bb40-4ee9-b895-c53a7c392ae2",
      "@type": "uco-observable:ObservableObject",
      "uco-core:hasFacet": [
        {
          "@type": "uco-observable:ApplicationFacet",
          "uco-observable:applicationIdentifier": "Skype"
          "uco-observable:numberOfLaunches": {
            "@type": "xsd:integer",
            "@value": "188"
          },
          "uco-observable:version": "8.91"
        }
      ]
    }
```

4.2.3. *Call (both Phone or App call)*

A call facet is a grouping of characteristics unique to a connection as part of a real-time cyber communication between one or more parties. This artifact includes both the traditional Call and the Call made by using an application (Whatsapp, Telegram etc.).

Figure 15: UCO/CASE Artifact Call

An example of JSON-LD serialisation for the *Call Observable* is as follows:

```
{
        "@id": "kb:uuid_call",
        "@type": "uco-observable:ObservableObject",
        "uco-core:hasFacet": [
          {
            "@type": "uco-observable:CallFacet",
            "uco-observable:callType": "incoming",
            "uco-observable:allocationStatus": "Intact",
            "uco-observable:startTime": {
              "@type": "xsd:dateTime",
              "@value": "2020-05-23T14:49:19+00:00"
            },
            "uco-observable:duration": {
              "@type": "xsd:integer",
              "@value": "35"
            },
            "uco-observable:application": {
              "@id": "kb:uuid_application "
            },
            "uco-observable:from": {
              "@id": "kb:uid_account_1"
            },
            "uco-observable:participants": {
```

```
            "@id": "kb:uuid_account_2"
          }
        }
      ]
    }
```

4.2.4. Chain of Evidence

The scope of the UCO/CASE ontologies cover investigations in any context, including in criminal, cybersecurity, and intelligence. Digital evidence includes data sources (mobile devices, storage media, memory) and well-known digital objects such as files and folders, messages (email, chat), documents (PDF, Word), multimedia (pictures, video, audio) and logs (browser history, events). UCO/CASE ensures that analysis results can be traced back to their source(s), keeping track of when, where and who used which tools to perform investigative actions on data sources. These details are generally referred to as provenance (e.g., Chain of Custody) and lineage (e.g., Chain of Evidence).
The Chain of Evidence describes from which file a given digital trace (Message, Call, Email etc.) comes from, or, in other words from where it is originated, therefore it is a key information to back up the admissibility of an element of evidence.
UCO/CASE represents the Chain of Evidence through a Relationship of kind "Contained_Within" where the uco-core:source property refers to the element of evidence (in the below example it consists of a CallFacet Observable) and the uco-core:target property refers to a file, in the below example it is the Call Database (CallHistory.storedata-wal), represented by a Write-Ahead Logging[17] file (WAV extension).

An example of JSON-LD serialisation for the *Chain of Evidence* representation is as follows:

```
{
    "@id": "kb:27669cf9-c0f1-49e8-a86e-d7756a666dd8",
    "@type": "uco-observable:ObservableObject",
    "uco-core:hasFacet": [
      {
        "@type": "uco-observable:CallFacet",
        "uco-observable:callType": "incoming",
        "uco-observable:allocationStatus": "Intact",
        "uco-observable:startTime": {
          "@type": "xsd:dateTime",
          "@value": "2021-04-10T23:49:32.021000+00:00"
        },
        "uco-observable:duration": {
          "@type": "xsd:integer",
          "@value": "1"
        },
        "uco-observable:application": {
          "@id": "kb:931ec331-6c44-44a6-8ca8-ff93804d6248"
```

[17] Android from 9 version on, introduced a special mode of SQLiteDatabase called Compatibility WAL (write-ahead logging) that allows a database to use journal_mode=WAL while preserving the behaviour of keeping a maximum of one connection per database. Enabling the WAL journal mode can lead to a significant improvement in performance and reduction in the amount of writes.

```
            },
            "uco-observable:from": {
              "@id": "kb:853b4a65-a4d8-4085-8413-420a8fa54af7"
            },
            "uco-observable:to": {
              "@id": "kb:1574e30c-7162-4374-8a09-23c5cfa277a0"
            }
          }
        ]
},
{
  "@id": "kb:dba9a898-6233-4974-b794-f5654fb4e90c",
  "@type": "uco-observable:ObservableObject",
  "uco-core:hasFacet": [
    {
      "@type": "uco-observable:FileFacet",
      "uco-observable:fileName": "CallHistory.storedata-wal",
      "uco-observable:filePath":
"/root/private/var/mobile/Library/CallHistoryDB/CallHistory.storedata-wal",
      "uco-observable:fileLocalPath": "files/Uncategorized/CallHistory.storedata-
wal",
      "uco-observable:extension": ".storedata-wal",
      "uco-core:objectAccessedTime": {
        "@type": "xsd:dateTime",
        "@value": "2021-02-03T22:46:12+00:00"
      },
      "uco-core:objectCreatedTime": {
        "@type": "xsd:dateTime",
        "@value": "2021-02-03T22:46:12+00:00"
      },
      "uco-core:objectModifiedTime": {
        "@type": "xsd:dateTime",
        "@value": "2021-07-17T21:30:30+00:00"
      },
      "uco-core:tag": [
        "Uncategorized"
      ],
      "uco-observable:sizeInBytes": {
        "@type": "xsd:integer",
        "@value": "3155952"
      }
    },
          {
      "@type": "uco-observable:ContentDataFacet",
      "uco-observable:hash": [
        {
          "@type": "uco-types:Hash",
          "uco-types:hashMethod": {
            "@type": "uco-vocabulary:HashNameVocab",
            "@value": "MD5"
          },
```

```
            "uco-types:hashValue": {
              "@type": "xsd:hexBinary",
              "@value": "0d8564e56ed83ab1d3253974c28f884e"
            }
          }
        ]
      }
    ]
  },
  {
      "@id": "kb:436cff99-5e3b-45f0-ab2c-587e3fe0bdad",
      "@type": "uco-observable:ObservableRelationship",
      "uco-core:isDirectional": {
        "@type": "xsd:boolean",
        "@value": "True"
      },
      "uco-core:kindOfRelationship": "Contained_Within",
      "uco-observable:startTime": {
        "@type": "xsd:dateTime",
        "@value": "1900-01-01T08:00:00+00:00"
      },
      "uco-observable:endTime": {
        "@type": "xsd:dateTime",
        "@value": "1900-01-01T08:00:00+00:00"
      },
      "uco-core:source": {
        "@id": "kb:27669cf9-c0f1-49e8-a86e-d7756a666dd8"
      },
      "uco-core:target": {
        "@id": "kb:dba9a898-6233-4974-b794-f5654fb4e90c"
      }
  }
```


4.2.5. Chat/Message and Thread Messages

A message facet is a grouping of characteristics unique to a discrete unit of electronic communication intended by the source for consumption by some recipient or group of recipients.

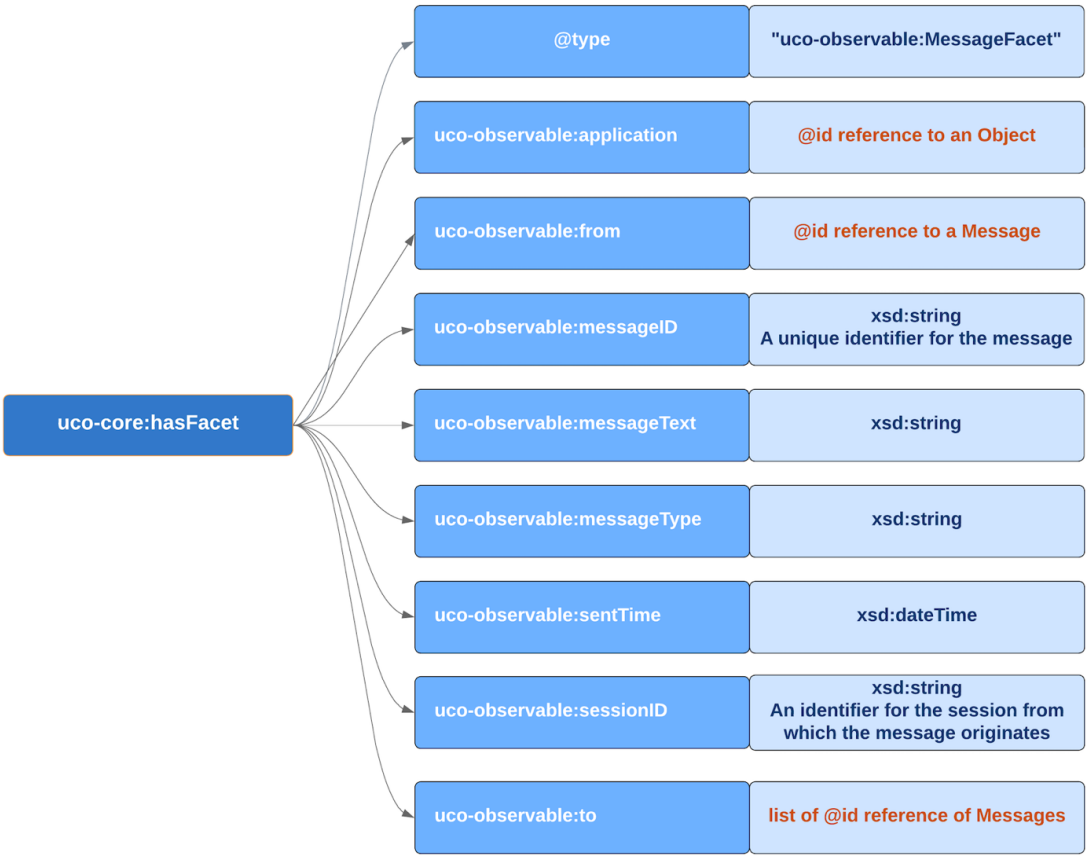

Figure 16: UCO/CASE Artifact Message

An example of JSON-LD serialisation for the *Chat/SMS Message Observable* is as follows:


```
{
        "@id": "kb:2808ffcb-48c0-4a84-9d71-6e86acc76999",
        "@type": "uco-observable:ObservableObject",
        "uco-core:hasFacet": [
          {
            "@type": "uco-observable:MessageFacet",
            "uco-observable:messageText": "\ud83d\udcf7 Beth, Scrap Yard Broke
shared a photo: https://fb.com/l/1G2R0BNUQUk5vjJ",
            "uco-observable:messageType": "CHAT Message",
            "uco-observable:sentTime": {
              "@type": "xsd:dateTime",
              "@value": "2021-07-03T21:48:49+00:00"
            },
            "uco-observable:from": {
              "@id": "kb:d9e41909-83a8-40a6-9ca5-58c529540e9d"
            },
            "uco-observable:to": [
              {
                "@id": "kb:86e6f8f1-0194-4988-adb4-23974e57ab92"
              }
            ],
            "uco-observable:application": {
              "@id": "kb:e0198ab6-c11d-49f0-9ca9-4037735dfc02"
```

```
            }
        }
    ]
}
```

A Message Thread facet is a grouping of characteristics unique to a running commentary of electronic messages pertaining to one topic or question.

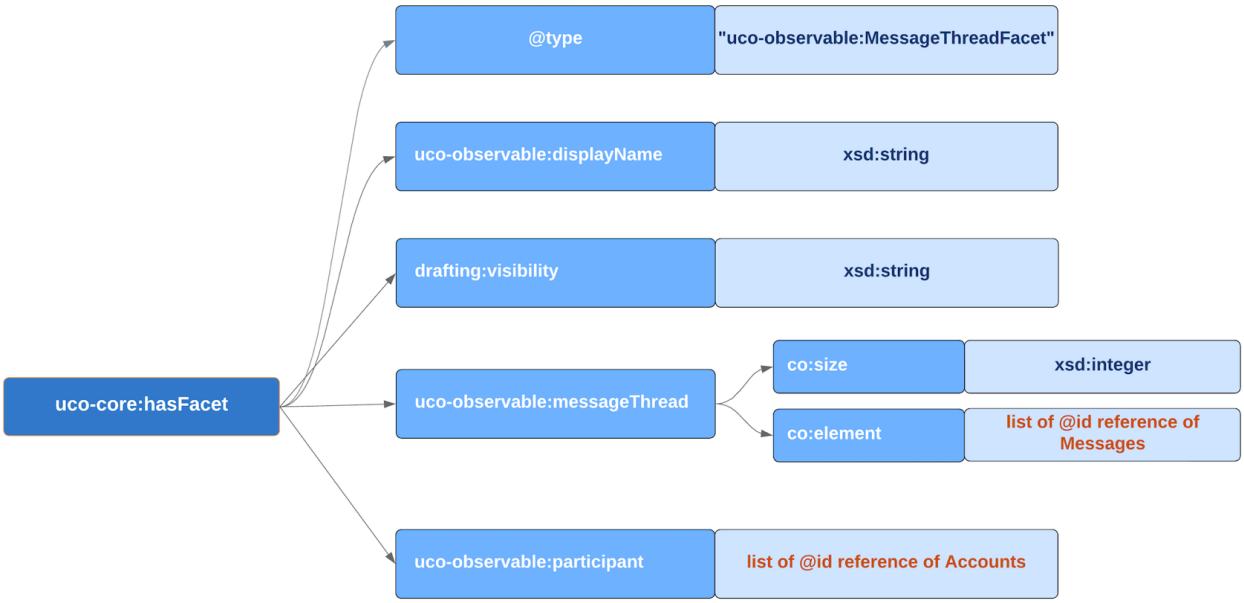

Figure 11: UCO/CASE Artifact Message Thread

An example of JSON-LD serialisation for the *Message Thread Observable* (valid only for Chat Message) is as follows:

```
{
    "@id": "kb:4237ff8b-e049-4a56-bcf7-e89ed2797d83",
    "@type": "uco-observable:ObservableObject",
    "uco-core:hasFacet": [
        {
            "@type": "uco-observable:MessageThreadFacet",
            "uco-observable:displayName": "Best Friend Chat!!",
            "drafting:visibility": "PRIVATE",
            "uco-observable:messageThread": {
                "co:size": {
                    "@type": "xsd:nonNegativeInteger",
                    "@value": "3"
                },
                "co:element": [
                    {
                    "@id": "kb:message-d8330d5a-b8de-4425-
9cd8-a37b038afe81"
                    },
                    {
                    "@id": "kb:message-decea264-014d-4da8-
9d7f-a63231d2c049"
```

```
                                        },
                                        {
                                        "@id": "kb:message-e809578d-3890-4d21-
82ec-afde487b3d7e"
                                        }
                                ],
                        },
                        "uco-observable:participant": [
                                {
                                "@id": "kb:account-3b61cb4c-f5fd-428c-80d7-
79ac841a4f87"
                                },
                                {
                                "@id": "kb:account-16f128ac-7e5b-4cac-908c-
11062488eb06"
                                }
                        ]
                }
        ]
}
```


4.2.6. Cookie

A browser cookie facet is a grouping of characteristics unique to a piece of data sent from a website and stored on the user's computer by the user's web browser while the user is browsing. The class has been lightly extended adding the *not-in-ontology:source* property related to the browser application that generated the cookie.

Figure 17: UCO/CASE Artifact Browser Cookie

An example of JSON-LD serialisation for the *Browser Cookie Observable* is as follows:


```
{
    "@id": "kb:ddc42145-2048-46d6-93ff-a8ec0ef8da18",
    "@type": "uco-observable:ObservableObject",
    "uco-core:hasFacet": [
      {
        "@type": "uco-observable:BrowserCookieFacet",
        "uco-observable:cookieName": "amzn-app-id",
        "uco-observable:cookiePath": "Amazon.com/18.3.0.100/18.0.225.0",
        "uco-observable:observableCreatedTime": {
          "@type": "xsd:dateTime",
          "@value": "2021-07-14T08:54:07+00:00"
        },
        "uco-observable:lastAccessTime": {
          "@type": "xsd:dateTime",
          "@value": "2022-02-11T16:28:44+00:00"
        },
        "uco-observable:expirationTime": {
          "@type": "xsd:dateTime",
          "@value": "2022-12-31T08:00:00+00:00"
        },
        "not-in-ontology:source": {
          "@id": "kb:e5fc0226-677b-4c56-923a-14b473463805"
        },
        "uco-observable:cookieDomain": {
          "@id": "kb:d0f934cb-a3a4-4835-959d-510685b3883e"
        }
      }
    ]
  }
```


4.2.7. Mobile Device

A device facet is a grouping of characteristics unique to a piece of equipment or a mechanism designed to serve a special purpose or perform a special function. This section describes a mobile device facet that is a grouping of characteristics unique to a portable computing device.

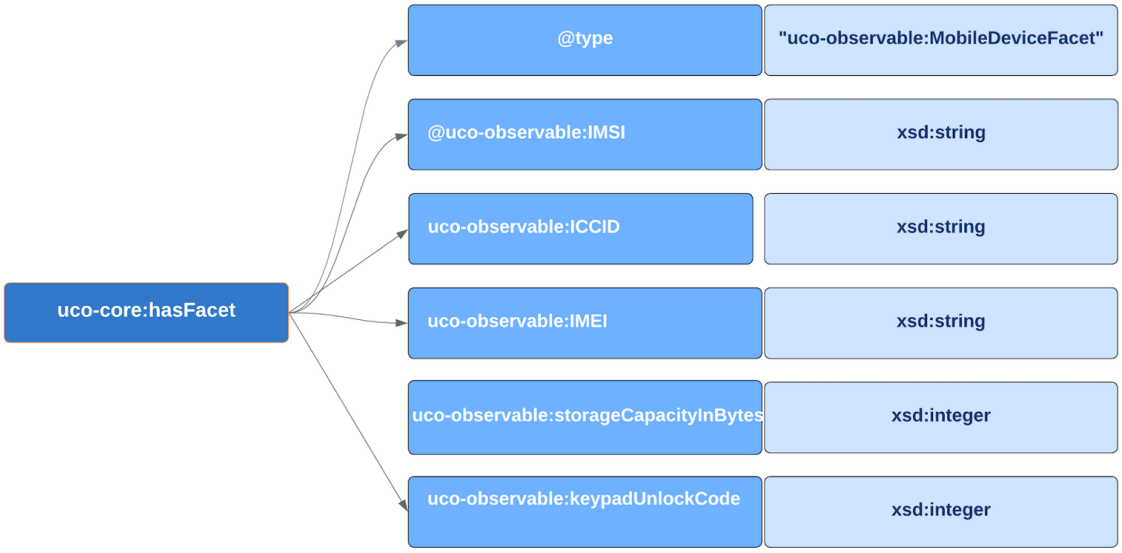

Figure 18: UCO/CASE Artifact Mobile Device

An example of JSON-LD serialisation for the *Mobile Device Observable* also including the *OperatingSystem*, the *BluetoothAddress* and the *WifiAddress* Observables is as follows:


```
{
        "@id": "kb:d7369240-5755-402b-8c69-fb169ebd504f",
        "@type": "uco-observable:ObservableObject",
        "uco-core:hasFacet": [
          {
            "@type": "uco-observable:DeviceFacet",
            "uco-observable:deviceType": "Mobile phone",
            "uco-observable:model": "Huawei POT-LX1",
            "uco-observable:serialNumber": "c8e1e8110f316eec"
          },
          {
            "@type": "uco-observable:MobileDeviceFacet",
            "uco-observable:IMSI": "222108503288987",
            "uco-observable:ICCID": "89391044100165786 32F",
            "uco-observable:IMEI": "862094049983802",
            "uco-observable: storageCapacityInBytes":
            {
              "@type": "xsd:integer",
              "@value": "67108864"
            }
          },
          {
            "@type": "uco-observable:OperatingSystemFacet",
            "uco-core:name": "Android",
            "uco-observable:manufacturer": "Huawei",
            "uco-observable:version": "12.0.0.226"
          },
          {
```

```
            "@type": "uco-observable:BluetoothAddressFacet",
            "uco-core:name": "huaweiFabri",
            "uco-observable:addressValue": "34:B2:0A:22:58:FA"
        },
        {
            "@type": "uco-observable:WifiAddressFacet",
            "uco-observable:addressValue": "34:B2:0A:22:47:65"
        }
    ]
}
```

4.2.8. Email

An email message facet is a grouping of characteristics unique to a message that is an instance of an electronic mail correspondence conformant to the internet message format described in RFC 5322[18] and related RFCs.

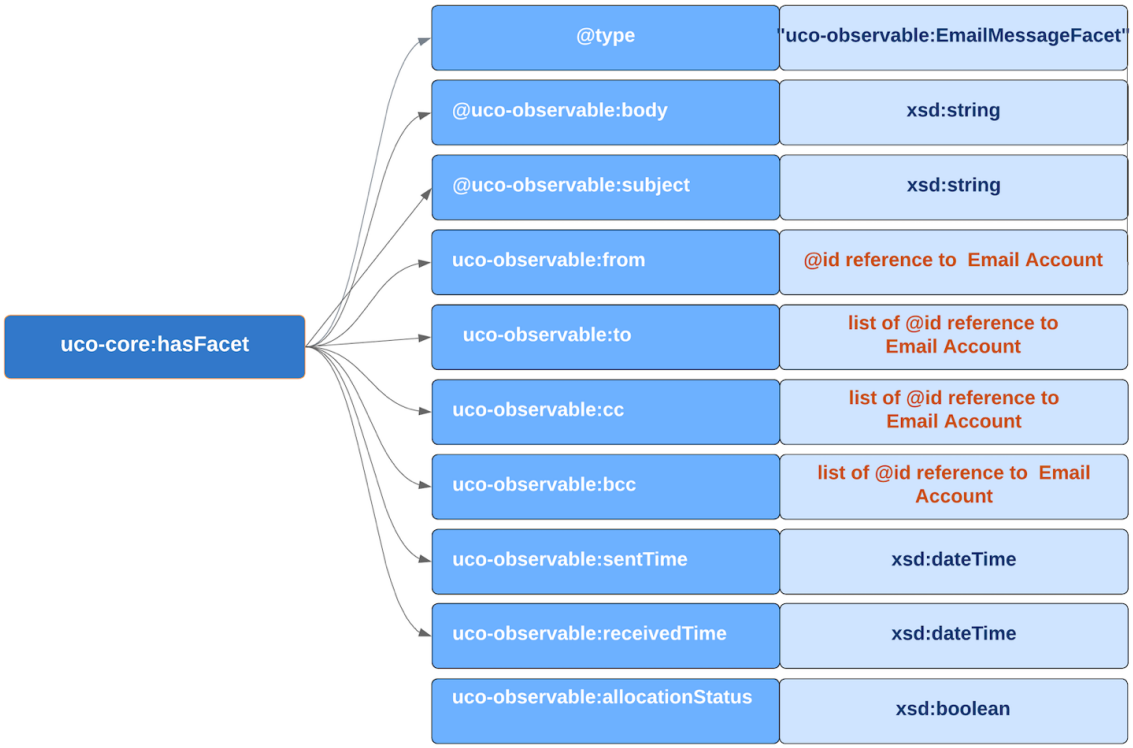

Figure 19: UCO/CASE Artifact Email Message

The Figure 19 illustrates the main property of the EmalMessageFacet class. The EmailAccountfacet referred in the "from", "to", "cc" and "bcc" properties are represented in the serialization example below.

An example of JSON-LD serialisation for the *Email Message Observable*, also including the *Account*, the *EmailAccount* and the *EmailAddress Observables* is as follows:

[18] Internet Message Format (IMF). See at <https://www.rfc-editor.org/rfc/rfc5322>.

```
{
        "@id": "kb:d560ed96-4598-4cbe-b47f-76bacb32c5cb",
        "@type": "uco-observable:ObservableObject",
        "uco-core:hasFacet": [
          {
            "@type": "uco-observable:EmailAddressFacet",
            "uco-observable:addressValue": "grace.vanpet@cbi.california.org"
          }
        ]
},
{"@id":"kb:93450be3-30f5-4e17-a920-6b1db60369a3",
  "@type":"uco-observable:ObservableObject",
  "uco-core:hasFacet":[
    {
      "@type": "uco-observable:AccountFacet",
      "uco-observable:accountIdentifier": "Sugar Ray - California"
    },
    {
      "@type": "uco-observable:EmailAccountFacet",
      "uco-observable:emailAddress": {
        "@id": "kb:d560ed96-4598-4cbe-b47f-76bacb32c5cb"
      }
    }
  ]
},
{
        "@id": "kb:2049151d-90ee-48fd-9f91-2374e5e33eec",
        "@type": "uco-observable:ObservableObject",
        "uco-core:hasFacet": [
          {
            "@type": "uco-observable:EmailAddressFacet",
            "uco-observable:addressValue": "gemma.propols@gmail.com"
          }
        ]
     },
{
        "@id": "kb:6d9efdf1-4710-464e-8b7d-7e217910c4d5",
        "@type": "uco-observable:ObservableObject",
        "uco-core:hasFacet": [
          {
            "@type": "uco-observable:AccountFacet",
            "uco-observable:isActive": {
              "@type": "xsd:boolean",
              "@value": "True"
            },
            "uco-observable:accountIdentifier": "Gemma Propols"
          },
          {
            "@type": "uco-observable:EmailAccountFacet",
            "uco-observable:emailAddress": {
              "@id": "kb:2049151d-90ee-48fd-9f91-2374e5e33eec"
```

```
            }
          }
        ]
      },
{
  "@id":"kb:24177e43-3303-441e-8d24-eab9e9eda815",
  "@type":"uco-observable:ObservableObject",
  "uco-core:hasFacet":[
   {
     "@type":"uco-observable:EmailMessageFacet",
     "uco-observable:sentTime":{
       "@type":"xsd:dateTime",
       "@value":"2020-02-01T16:12:14"
     },
     "uco-observable:from":{
       "@id":"kb:93450be3-30f5-4E17-a920-6b1db60369a3"
     },
     "uco-observable:to":[
      {"@id":"kb:cb360760-d200-4384-ad9f-417abe60f85e"}
     ],
     "uco-observable:cc":[
            {"@id":"kb: 6d9efdf1-4710-464e-8b7d-7e217910c4d5"}
     ],
     ],
     "uco-observable:bcc":[],
     "uco-observable:body":"Rose, I was never mean to you. Yes, I might have
teased you a little. But I was never mean to you on purpose.",
     "uco-observable:subject":"Rose, my property!",
     "uco-observable:allocationStatus":"Intact"
   }
  ]
}
```


4.2.9. Event

An event facet is a grouping of characteristics unique to something that happens in a digital context (e.g., operating system events).
This class has been extended introducing the "*not-in-ontology:observableStartTime*" and the "*not-in-ontology:observableEndTime*" properties.

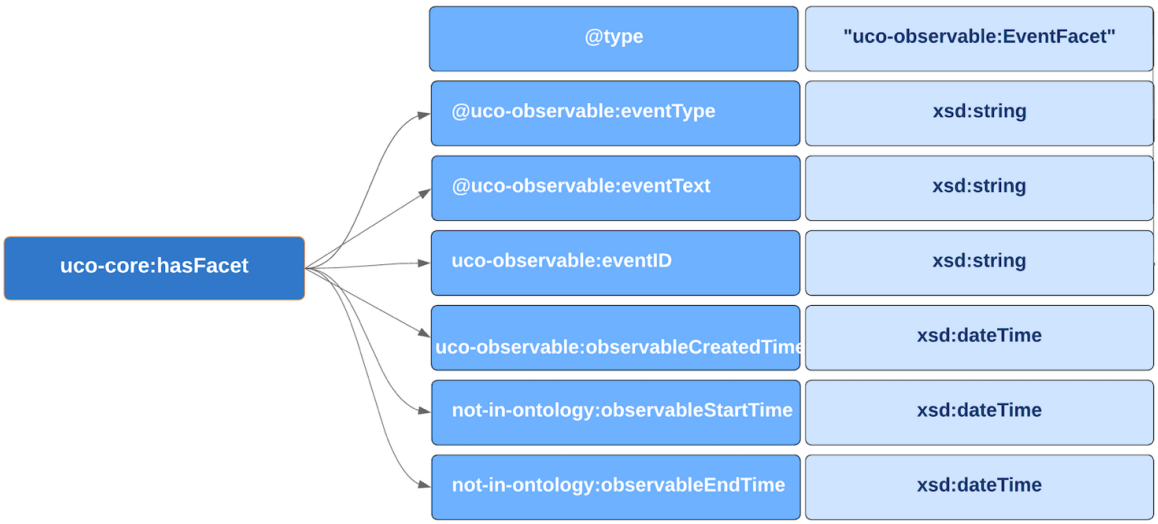

Figure 20: UCO/CASE Artifact Event

An example of JSON-LD serialisation for the *Event Observable*: is as follows:

```
{
    "@id": "kb:fe688202-4b21-48d2-b07d-170e14971935",
    "@type": "uco-observable:ObservableObject",
    "uco-core:hasFacet": [
      {
        "@type": "uco-observable:EventFacet",
        "uco-observable:eventType": "PowerEvent",
        "uco-observable:eventText": "Power on",
        "uco-observable:observableCreatedTime": {
          "@type": "xsd:dateTime",
          "@value": "2021-05-18T11:31:46+00:00"
        }
      }
    ]
  }
```

4.2.10. File and EXIF

A file facet is a grouping of characteristics unique to the storage of a file (computer resource for recording data discretely in a computer storage device) on a file system (process that manages how and where data on a storage device is stored, accessed and managed).

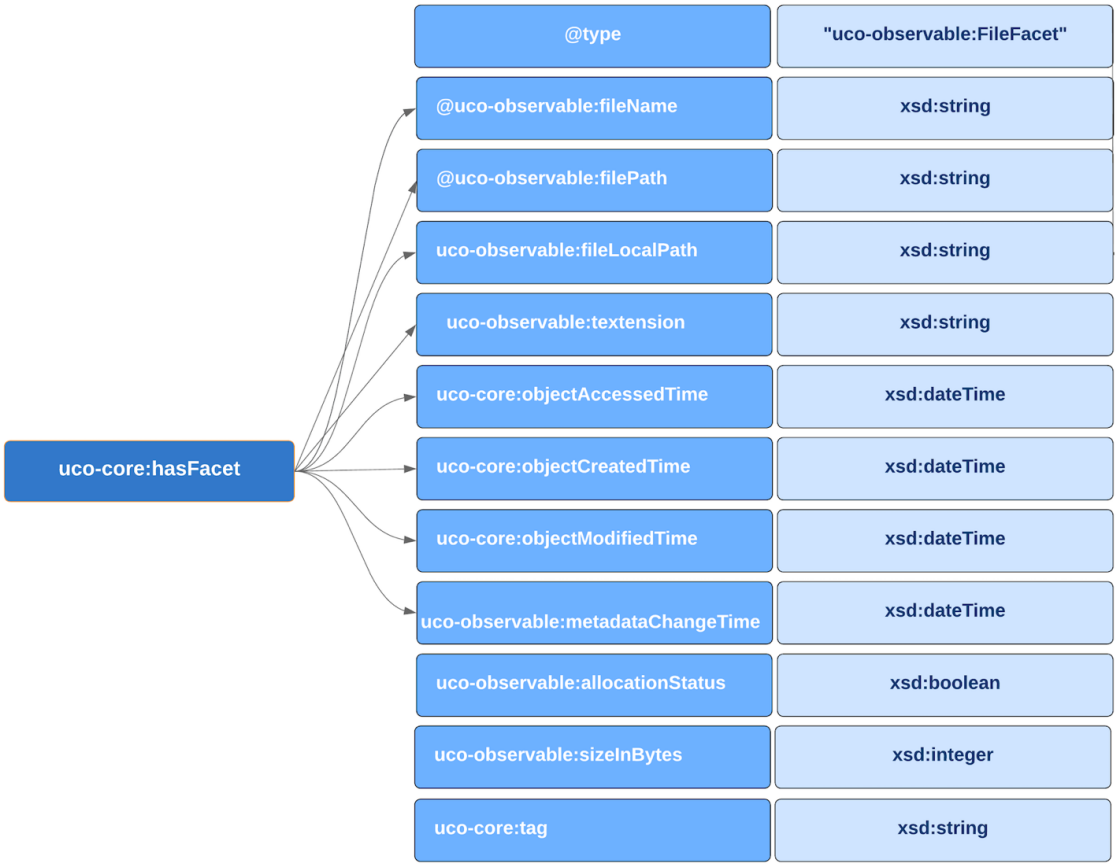

Figure 21: UCO/CASE Artifact File

Actually the "*uco-observable:fileLocalPath*" property is not part of the ontologies' classes, for this property the "not-in-ontology" space should be used instead. The property refers to the local path to the physical files extracted by the forensic tool. For instance, by using the UFED PA by Cellebrite, the physical files are stored in a folders structure whose root is "file" and inside it there are other folders that depend on the kind of files such as Image, Video, Audio etc.

An example of JSON-LD serialisation for the *File Observable*: is as follows:

```
{
        "@type": "uco-observable:FileFacet",
        "uco-observable:fileName": "homescreenPreview.png",
        "uco-observable:filePath":
"/Root/media/0/Android/data/com.sec.android.app.launcher/cache/homescreenP
review.png",
        "uco-observable:fileLocalPath": "files/Image/homescreenPreview.png",
        "uco-observable:extension": ".png",
        "uco-core:objectAccessedTime": {
          "@type": "xsd:dateTime",
          "@value": "2018-06-10T14:36:27+00:00"
        },
        "uco-core:objectCreatedTime": {
          "@type": "xsd:dateTime",
```

```
            "@value": "2018-06-10T14:36:27+00:00"
          },
          "uco-core:objectModifiedTime": {
            "@type": "xsd:dateTime",
            "@value": "2018-12-10T08:55:27+00:00"
          },
          "uco-core:tag": [
            "Image"
          ],
          "uco-observable:sizeInBytes": {
            "@type": "xsd:integer",
            "@value": "525099"
          }
        }
      ]
    }
```


Figure 22 represent the structure of the "*uco-observable:EXIFFacet*"class:

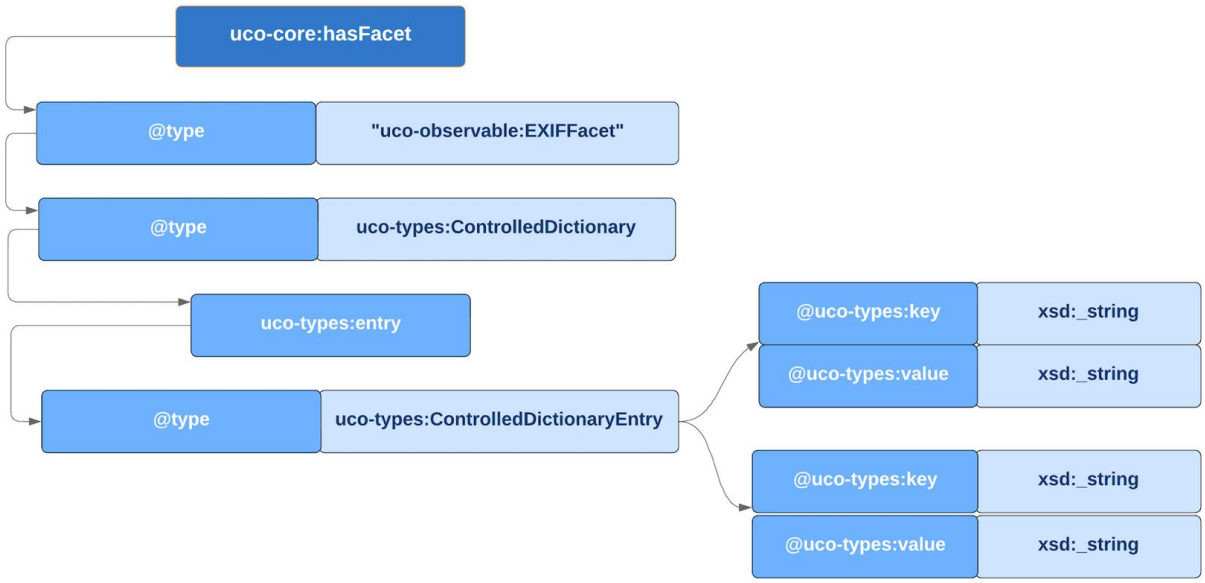

Figure 22: UCO/CASE Artifact EXIF Data

An example of JSON-LD serialisation for the *EXIF Observable*: is as follows:


```
{
          "@type": "uco-observable:EXIFFacet",
          "uco-observable:exifData": {
            "@type": "uco-types:ControlledDictionary",
            "uco-types:entry": [
              {
                "@type": "uco-types:ControlledDictionaryEntry",
                "uco-types:key": "Make",
                "uco-types:value": "Apple"
              },
              {
                "@type": "uco-types:ControlledDictionaryEntry",
```

```
                "uco-types:key": "Model",
                "uco-types:value": "iPhone X"
            },
            {
                "@type": "uco-types:ControlledDictionaryEntry",
                "uco-types:key": "LatitudeRef",
                "uco-types:value": "N"
            },
            {
                "@type": "uco-types:ControlledDictionaryEntry",
                "uco-types:key": "Latitude",
                "uco-types:value": "39, 3, 10.48"
            },
            {
                "@type": "uco-types:ControlledDictionaryEntry",
                "uco-types:key": "LongitudeRef",
                "uco-types:value": "W"
            },
            {
                "@type": "uco-types:ControlledDictionaryEntry",
                "uco-types:key": "Longitude",
                "uco-types:value": "77, 26, 47.52"
            },
            {
                "@type": "uco-types:ControlledDictionaryEntry",
                "uco-types:key": "Altitude",
                "uco-types:value": "86.2744565217391"
            }
        ]
    }
}
```

4.2.11. Identity

An identity facet is a grouping of characteristics unique to a particular aspect of an identity. The class is a general container for other more peculiar classes such as: AddressFacet, OrganizationDetailsFacet, PersonalDetailsFacet, SimpleNameFacet and others. The structure of these classes is very simple, here only the SImpleNamefacet JSON-LD serialization is provided:

```
{
    "@id": "kb:8e4f771d-4fa0-4f70-b593-20d8f00e0461",
    "@type": "uco-observable:ObservableObject",
    "uco-core:hasFacet": [
        {
            "@type": "uco-identity:SimpleNameFacet",
            "uco-identity:givenName": "Jane",
            "uco-identity:familyName": "Austin"
        }
    ]
}
```

4.2.12. *Location device and Geo Coordinates*

The Location device is the global positioning system (GPS) that enables the cell phone to know the location coordinates at all times. This data is represented in UCO/CASE by using the Relationships Object of kind "Mapped_By" between the Mobile Device (see Section 3.7) and the geo coordinates represented as a "*LatLongCoordinatesFacet*" class (see below).

An example of JSON-LD serialisation for the *Relationship Observable* is as follows:

```
{
    "@id": "kb:d78785be-d1d0-44b8-bf34-7a8667cb8308",
    "@type": "uco-observable:ObservableRelationship",
    "uco-core:isDirectional": {
      "@type": "xsd:boolean",
      "@value": "True"
    },
    "uco-core:kindOfRelationship": "Mapped_By",
    "uco-observable:startTime": {
      "@type": "xsd:dateTime",
      "@value": "2021-05-04T22:49:11+00:00"
    },
    "uco-observable:endTime": {
      "@type": "xsd:dateTime",
      "@value": "1900-01-01T08:00:00+00:00"
    },
    "uco-core:source": {
      "@id": "kb:e4912cd1-abc3-4ad4-9d73-f54379752e02"
    },
    "uco-core:target": {
      "@id": "kb:a7121daf-3f10-4e50-a77f-63cbc7a37e34"
    }
  }
```

The "uco-core:source" property refers to the MobileDevice Object and the "uco-core:target" refers to a "*LatLongCoordinatesFacet*" class.
The GeoCoordinates are represented as follows:

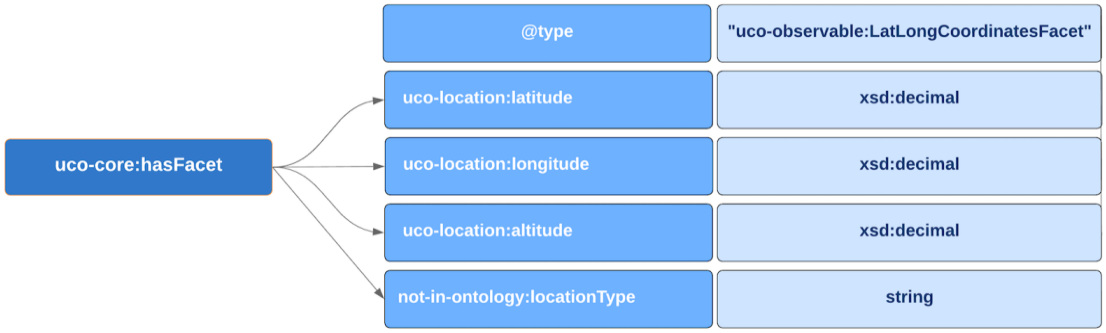

Figure 23: UCO/CASE Artifact GEO Coordinates class

An example of JSON-LD serialisation for the *LatLong Coordinates Observable* is as follows:


```
{
        "@id": "kb:924b8c85-7016-439e-b6f8-962fb5af1496",
        "@type": "uco-observable:ObservableObject",
        "uco-core:hasFacet": [
          {
            "@type": "uco-location:LatLongCoordinatesFacet",
            "uco-location:latitude": {
              "@type": "xsd:decimal",
              "@value": "40.11259078"
            },
            "uco-location:longitude": {
              "@type": "xsd:decimal",
              "@value": "-75.65714263"
            },
            "uco-location:altitude": {
              "@type": "xsd:decimal",
              "@value": "0.0"
            },
            "not-in-ontology:locationType": "Cell Tower"
          }
        ]
      }
```


4.2.13. Network Connection

A network connection facet is a grouping of characteristics unique to a connection (complete or attempted) across a digital network (a group of two or more computer systems linked together).

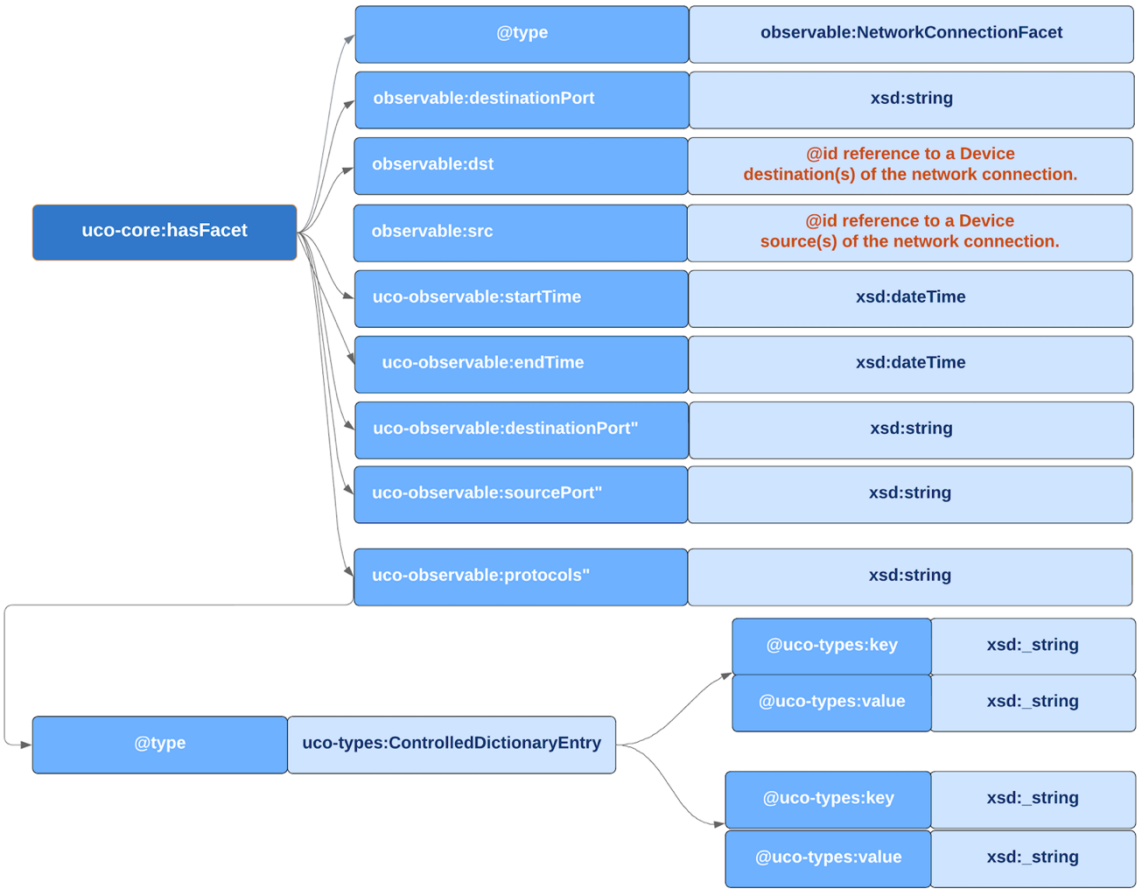

Figure 24: UCO/CASE Network Connection Artifact

An example of JSON-LD serialisation for the *Network Connection Observable* is as follows:

```
{
    "@id": "kb:66f723eb-81bd-439c-a106-949805005d8d",
    "@type": "uco-observable:ObservableObject",
    "uco-core:createdBy": {
        "@id": "kb:investigator-b132f44d-6417-46b6-8158-b8e03d948357"
    },
    "uco-core:objectCreatedTime": {
        "@type": "xsd:dateTime",
        "@value": "2017-09-29T11:47:54.2889922Z"
    },
    "uco-core:hasFacet": [
    {
        "@id": "kb:network-connection-facet-f6b13cbb-ed93-41a0-ad25-
c8a0cd8f13a8",
        "@type": "uco-observable:NetworkConnectionFacet",
        "uco-observable:startTime": {
            "@type": "xsd:dateTime",
            "@value": "2009-04-03T02:29:25.6264620Z"
        },
        "uco-observable:endTime": {
            "@type": "xsd:dateTime",
            "@value": "2009-04-03T02:29:25.6369450Z"
```

```
                },
                "uco-observable:dst": {
                "@id": "kb:destination-host-e7857c18-9d8a-4257-9eac-
b75d5a5bf8f0"
                },
                "uco-observable:destinationPort": {
                        "@type": "xsd:integer",
                        "@value":139
                },
                "uco-observable:src": {
                "@id": "kb:source-host-e44d685c-56fe-417c-a898-a2af0026268e"
                },
                "uco-observable:sourcePort": {
                        "@type": "xsd:integer",
                        "@value":52961
                },
                "uco-observable:protocols": {
                        "@type": "uco-types:ControlledDictionary",
                        "uco-types:entry": [
                                {
                                        "@type": "uco-
types:ControlledDictionaryEntry",
                                        "uco-types:key": "Transport Layer",
                                        "uco-types:value": "TCP"
                                },
                                {
                                        "@type": "uco-
types:ControlledDictionaryEntry",
                                        "uco-types:key": "Session Layer",
                                        "uco-types:value":
"NETBIOSSESSIONSERVICE "
                                }
                        ]
                }
        }
        ]
}
```


4.2.14. Place as Simple Address

A simple address facet is a grouping of characteristics unique to a geolocation expressed as an administrative address.

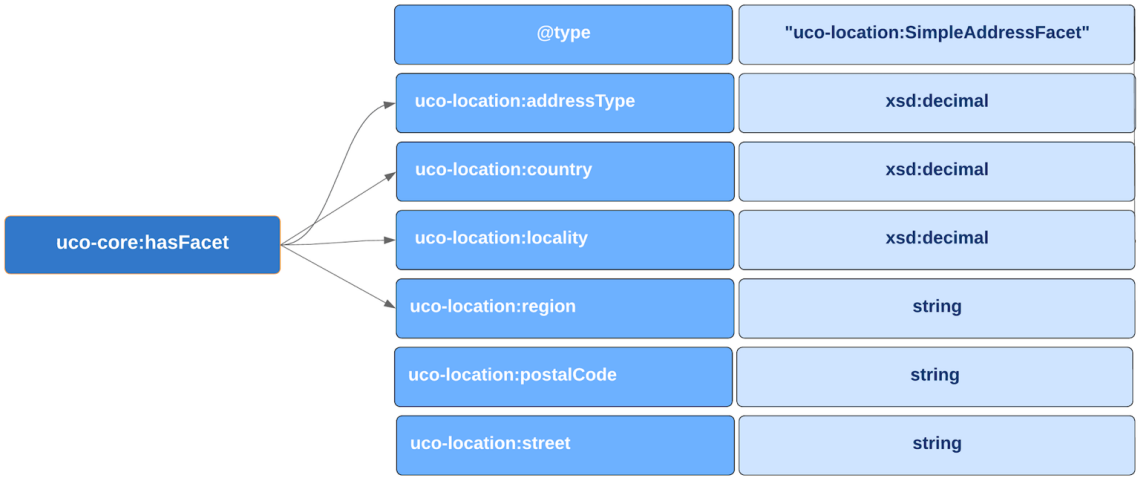

Figure 25: UCO/CASE Artifact Simple Address class

An example of JSON-LD serialisation for the *Simple Address Observable* is as follows:

```
{
  "@id":"kb:C69DC60D-5FCA-4221-8753-B574727A454C",
  "@type":"uco-location:Location",
  "uco-core:hasFacet":[
    {
      "@type":"uco-location:SimpleAddressFacet",
      "uco-location:addressType":"Work",
      "uco-location:country":"Italy",
      "uco-location:locality":"Florence",
      "uco-location:region":"Tuscany",
      "uco-location:postalCode":"50199",
      "uco-location:street":"via dei Benci, 8"
    }
  ]
}
```

4.2.15. Role

A role is a usual or customary function based on contextual perspective. The most common use of the Role class is related to the people or organisation involved in carrying out specific forensic actions within the evidence timeline. For instance, it is important to know the role of who carried out the acquisition or the extraction of the elements of evidence from the original device or the role of who did the search and seizure or the role of who issued the search warrant.
The structure of the class is very simple, therefore only JSON-LD serialization is provided:

```
{
  "@id":"kb:DC35012F-43FB-42FF-A8FA-AE74F4160ACB",
  "@type":"uco-role:Role",
  "uco-role:name":"Forensic Expert",
  "uco-role:description":"A digital forensic expert in dealing with mobile devices",
}
```

4.2.16. *Provenance (Chain of Custody)*

A provenance record is a grouping of characteristics unique to the provenance (chronology of the ownership, custody or location) connection between an investigative action and a set of observations (items and/or actions) or interpretations that result from it.

In any investigation, it is important to maintain links between data sources, their treatment, and analysis results. UCO/CASE includes concepts for keeping track of when, where and who used which tools to perform investigative actions on data sources. These details are generally referred to as provenance (e.g., chain of custody). UCO/CASE tracks provenance with the combination of *ProvenanceRecords* and *InvestigativeActions* as explained later.

UCO/CASE aligns with the PROV-O ontology[19] to support enhanced provenance tracking that is needed in more mature operating environments. The PROV-O enhanced representing of the Urgent Evidence scenario is depicted here.

UCO/CASE represents this data by using the *ProvenanceRecord* class. The Input (*uco-action:object* property) and the Output (*uco-action:result* property) of the *InvestigativeAction* are represented as an *@id* reference to a *ProvenanceRecord*. An example of this class, represented in a graphical simplified manner, related to a Forensic Acquisition and a Forensic Extraction action, is illustrated in Figure 26:

ACTION - Mobile device - Forensic Acquisition			
Data	*CASE Observable*	*CASE property*	*CASE value*
What	InvestigativeAction	description	Forensic acquisition
Who	SimpleNameFacet	familyName	Josie Loren
Where	SimpleAddressFacet	locality description ...	California Hal Moon Bay San Mateo Rd. 18
When	InvestigativeAction	startTime endTime	- 2022-09-19T14:21+00:00
Instrument	DeviceFacet	@id reference --> DeviceFacet	Cellebrite UFED 4PC version 7.1
Input	uco-action:object	array of @id reference --> DeviceFacet	deviceType": "Mobile phone" model": "Huawei POT-LX1"
Output	FileFacet	array of @id reference --> FileFacet	fileName: E01_Hauwaei.zip sizeInBytes: 64GB

ACTION - Mobile device - Forensic Extraction/Processing			
Data	*CASE Observable*	*CASE property*	*CASE value*
What	InvestigativeAction	description	Forensic processing
Who	SimpleNameFacet	familyName	Josie Loren
Where	SimpleAddressFacet	locality description ...	California Hal Moon Bay San Mateo Rd. 18
When	InvestigativeAction	startTime endTime	- 2022-09-20T08:15+00:00
Instrument	DeviceFacet	@id reference --> DeviceFacet	Cellebrite UFED PA version 10.5
Input	uco-action:object	array of @id reference --> DeviceFacet	fileName: E01_Hauwaei.zip sizeInBytes: 64GB
Output	FileFacet	array of @id reference --> FileFacet	list of files (@id) extracted

Figure 26: UCO/CASE Chain of Custody for the Acquisition and Extraction actions

Figure 26 shows two investigative actions, an Acquisition and the subsequent Extraction/Processing that takes as input the outcome obtained by the previous action (the Forensic Acquisition). Each action includes the data related to What (Description property), Who, Where, When, Instrument, Input and Output related to the action. Relying

[19] The PROV Ontology (PROV-O) expresses the PROV Data Model [PROV-DM] using the OWL2 Web Ontology Language (OWL2) [OWL2-OVERVIEW]. It provides a set of classes, properties, and restrictions that can be used to represent and interchange provenance information generated in different systems and under different contexts. It can also be specialized to create new classes and properties to model provenance information for different applications and domains. See https://www.w3.org/TR/prov-o for further details

on an investigative action data is possible to answer relevant investigative questions, depicted in Figure 27:

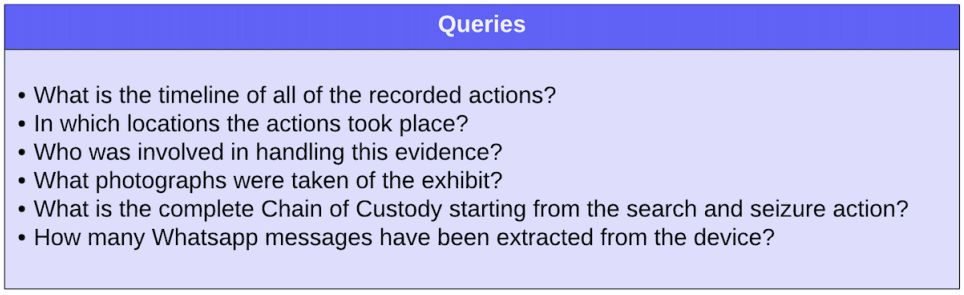

Figure 27: Possible queries based on Investigative Action data

Figure 28 illustrates the properties of the *InvestigationAction* class:

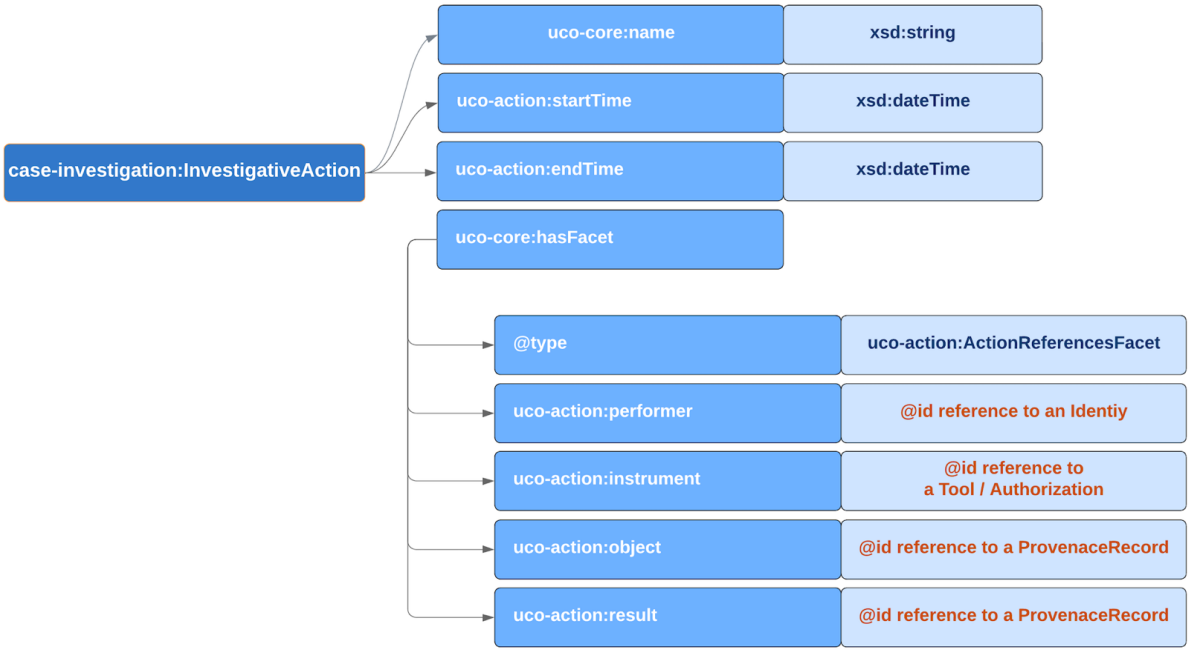

Figure 28: UCO/CASE Investigative Action class

For an Investigative Action related to a Forensic Acquisition, the *ProvenanceRecord* of the *uco-core:object* property (Input of the Action) contains the reference to the original source of evidence, that is the device under processing. Whilst the *ProvenanceRecord* of the *uco-core:result* property (Output of the Action) contains the reference to all the files obtained from the acquisition process.

An example of JSON-LD serialisation for the *Investigation Action Class* is as follows:

```
{
    "@id": "kb:0d549b0e-4484-4858-9e48-fb21e3f317f7",
    "@type": "case-investigation:InvestigativeAction",
    "uco-core:name": "Forensic mobile device acquisition",
    "uco-action:startTime": {
      "@type": "xsd:dateTime",
```

```
        "@value": "2021-07-29T12:28:49+00:00"
      },
      "uco-action:endTime": {
        "@type": "xsd:dateTime",
        "@value": "2021-07-29T12:43:44+00:00"
      },
      "uco-core:hasFacet": [
        {
          "@type": "uco-action:ActionReferencesFacet",
          "uco-action:performer": {
            "@id": "kb:8e4f771d-4fa0-4f70-b593-20d8f00e0461"
          },
          "uco-action:instrument": {
            "@id": "kb:4252f4ee-d2bd-4d83-bbb4-2669a7be8286"
          },
          "uco-action:result": [
            {
              "@id": "kb:9ebe7b98-5323-4bf5-b44a-d499c928b93d"
            }
          ],
          "uco-action:object": {
            "@id": "kb:6e4a276b-0b5e-472c-b1cf-a4f3dfd4e5d8"
          }
        }
      ]
}
```


4.2.17. URL history

Specifies a URL history record stored in the browser's history. The class also includes the observable:URLHistoryEntry class that represents a grouping of characteristics unique to the properties of a single URL history entry for a particular browser.

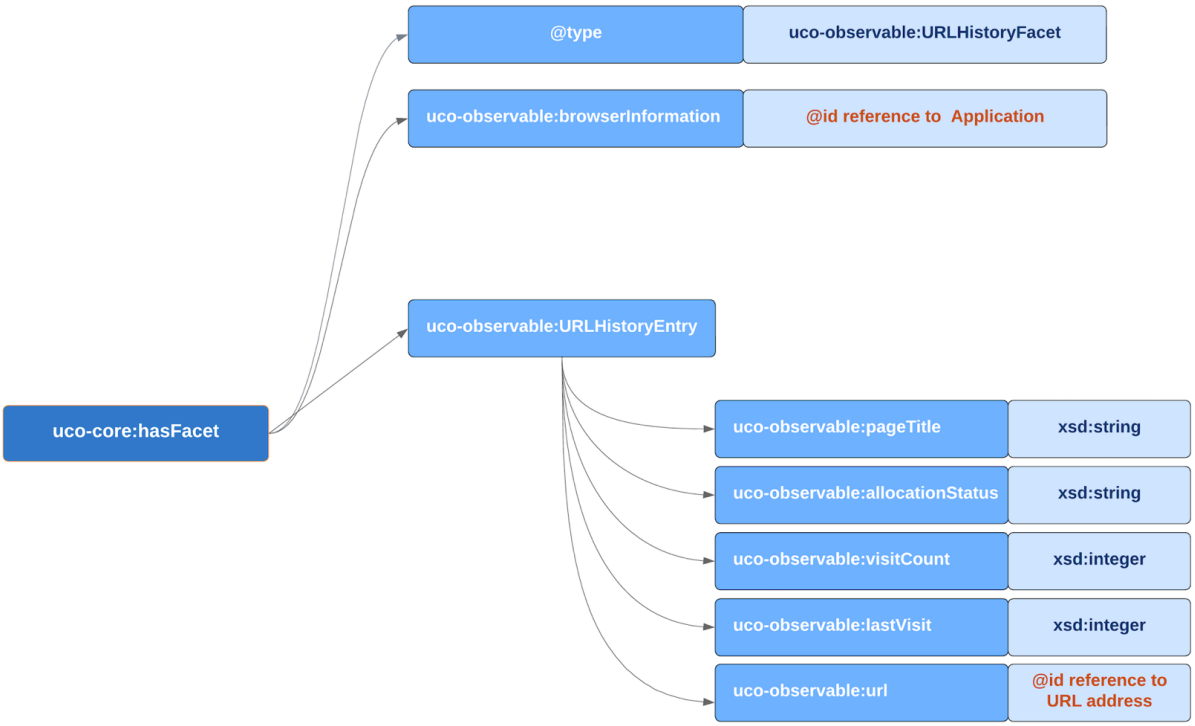

Figure 29: UCO/CASE Artifact URL History

An example of JSON-LD serialisation for the *URL History Observable*: is as follows:

```
{
  "UCO-CASE comment": " URLFacet  Object."
},

{
        "@id": "kb:1ad5773d-fc2c-4004-9c86-d2056c60089c",
        "@type": "uco-observable:ObservableObject",
        "uco-core:hasFacet": [
          {
            "@type": "uco-observable:URLFacet",
            "uco-observable:fullValue": "https://www.photographyblog.com/"
          }
        ]
    },
{
  "UCO-CASE comment": " ApplicationFacet Object."
},

{
        "@id": "kb:9c4cf79b-069d-44f2-abed-c999042b5c3d",
        "@type": "uco-observable:ObservableObject",
        "uco-core:hasFacet": [
          {
            "@type": "uco-observable:ApplicationFacet",
            "uco-core:name": "Safari"
          }
```

```
        ]
      },
{
  "UCO-CASE comment": " URLHistoryFacet Object."
},

{
        "@id": "kb:2f941e71-1f29-445a-b5f2-5535b41aa739",
        "@type": "uco-observable:ObservableObject",
        "uco-core:hasFacet": [
          {
            "@type": "uco-observable:URLHistoryFacet",
            "uco-observable:browserInformation": {
              "@id": "kb:9c4cf79b-069d-44f2-abed-c999042b5c3d"
            }
          },
          {
            "@type": "uco-observable:URLHistoryEntry",
            "uco-observable:pageTitle": "Photography Blog",
            "uco-observable:allocationStatus": "Intact",
            "uco-observable:visitCount": {
              "@type": "xsd:integer",
              "@value": "15"
            },
            "uco-observable:lastVisit": {
              "@type": "xsd:dateTime",
              "@value": "2021-06-14T20:08:45+00:00"
            },
            "uco-observable:url": {
              "@id": "kb:1ad5773d-fc2c-4004-9c86-d2056c60089c"
            }
          }
        ]
}
```

4.2.18. Web bookmark

A browser bookmark facet is a grouping of characteristics unique to a saved shortcut that directs a WWW (World Wide Web) browser software program to a particular WWW accessible resource.

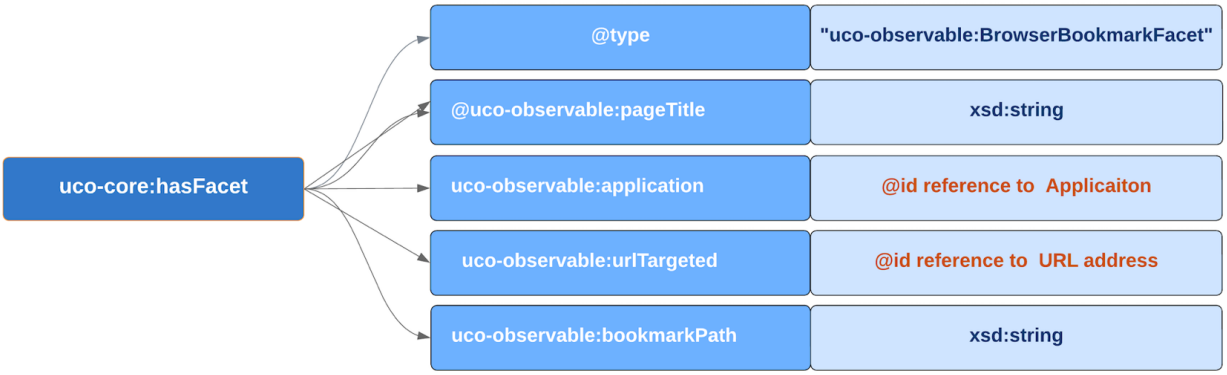

Figure 30: UCO/CASE Artifact Browser Bookmark

An example of JSON-LD serialisation for the *Browser Bookmark Observable*: is as follows:

```
{
  "UCO-CASE comment": " ApplicationFacet Object."
},
{
        "@id": "kb:931ec331-6c44-44a6-8ca8-ff93804d6248",
        "@type": "uco-observable:ObservableObject",
        "uco-core:hasFacet": [
          {
            "@type": "uco-observable:ApplicationFacet",
            "uco-core:name": "Google Chrome"
          }
        ]
      },

{
  "UCO-CASE comment": " URLFacet Object."
},
{
        "@id": "kb:3e05a2bc-ce85-4a72-9c83-93e1c92ddfa9",
        "@type": "uco-observable:ObservableObject",
        "uco-core:hasFacet": [
          {
            "@type": "uco-observable:URLFacet",
            "uco-observable:fullValue": "
https://docwiki.embarcadero.com/RADStudio/Seattle/en/Mobile_Tutorial:_Set_Up_Your_Development_Environment_on_the_Mac_(iOS)"
          }
        ]
      },

{
  "UCO-CASE comment": " BrowserBookmarkFacet Object."
},

{
        "@id": "kb:27669cf9-c0f1-49e8-a86e-d7756a666dd8",
```

```
        "@type": "uco-observable:ObservableObject",
        "uco-core:hasFacet": [
          {
            "@type": "uco-observable:BrowserBookmarkFacet",
            "uco-observable:pageTitle": "Mobile Tutorial: Set Up Your
Development Environment on the Mac (iOS)",
            "uco-observable:bookmarkPath": "/root/Library/Application
Support/Google/Chrome/Default",
            "uco-observable:observableCreatedTime": {
              "@type": "xsd:dateTime",
              "@value": "2022-04-10T23:49:32.021000+00:00"
            }
            "uco-observable:application": {
              "@id": "kb:931ec331-6c44-44a6-8ca8-ff93804d6248"
            },
            "uco-observable:urlTargeted": {
              "@id": "kb:853b4a65-a4d8-4085-8413-420a8fa54af7"
            }
          }
        ]
      }
```


4.2.19. Wireless network connection

A wireless network connection facet is a grouping of characteristics unique to a connection (completed or attempted) across an IEEE 802.11[20] standards-conformant digital network (a group of two or more computer systems linked together).

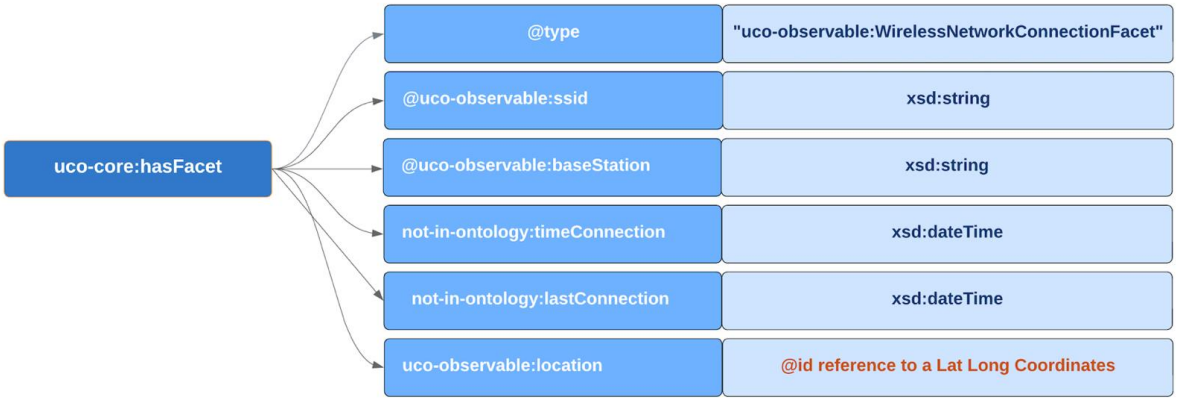

Figure 31: UCO/CASE Artifact Wireless Network Connection

An example of JSON-LD serialisation for the *Wireless Network Connection Observable*: is as follows:


```
{
        "@id": "kb:37445ac8-7fd9-4ab4-ba82-231eba274480",
        "@type": "uco-observable:ObservableObject",
        "uco-core:hasFacet": [
```


[20] IEEE 802.11 is the most popular standard for wireless LANs. See at <https://www.ieee802.org/11/>.

```
            {
              "@type": "uco-location:LatLongCoordinatesFacet",
              "uco-location:latitude": {
                "@type": "xsd:decimal",
                "@value": "40.05659866"
              },
              "uco-location:longitude": {
                "@type": "xsd:decimal",
                "@value": "-75.67047119"
              },
              "uco-location:altitude": {
                "@type": "xsd:decimal",
                "@value": "0.0"
              },
              "not-in-ontology:locationType": "Wireless Networks"
            }
          ]
    },
    {
          "@id": "kb:c64804f4-bd52-4194-8409-08018386374f",
          "@type": "uco-observable:ObservableObject",
          "uco-core:hasFacet": [
            {
              "@type": "uco-observable:WirelessNetworkConnectionFacet",
              "not-in-ontology:ssid": "78:BC:1A:36:25:E0",
              "not-in-ontology:timeConnection": {
                "@type": "xsd:dateTime",
                "@value": "2021-07-29T13:42:11+00:00"
              },
              "not-in-ontology:lastConnection": {
                "@type": "xsd:dateTime",
                "@value": "2021-12-14T18:32:28+00:00"
              },
              "uco-observable:location": {
                "@id": "kb:37445ac8-7fd9-4ab4-ba82-231eba274480"
              }
            }
          ]
    }
```


4.2.20. Windows Registry

The Microsoft Windows Registry is a central hierarchical database used to store information that is necessary to configure the system for one or more users, applications, and hardware devices. The Registry contains information that Windows continually references during operation, such as profiles for each user, the applications installed on the computer and the types of documents that each can create, property sheet settings for folders and application icons, what hardware exists on the system, and the ports that are being used.

The Microsoft Window Registry is represented by using three different classes:

- WindowsRegistryHiveFacet
- WindowsRegistryKeyFacet
- WindowsRegistryValueFacet

The WindowsRegistryHiveFacet class is a grouping of characteristics unique to a particular logical group of keys, subkeys, and values in a Windows registry (a hierarchical database that stores low-level settings for the Microsoft Windows operating system and for applications that opt to use the registry). The class has one single property:

- observable:hiveType of kind string

The WindowsRegistryKeyFacet class is a grouping of characteristics unique to a particular key within a Windows registry (A hierarchical database that stores low-level settings for the Microsoft Windows operating system and for applications that opt to use the registry). The class has the following properties:

- observable:creator: it refers to an @id of an Identity Object
- observable:key of kind string
- observable:modifiedTime of kind xsd: dateTime
- observable:numberOfSubkeys of kind integer
- observable:registryValues: it refers to a list of @id of WindowsRegistryValueFacet

The WindowsRegistryValueFacet class is a grouping of characteristics unique to a particular value within a Windows registry (a hierarchical database that stores low-level settings for the Microsoft Windows operating system and for applications that opt to use the registry. The class has the following properties:

- core:name —> string
- observable:data —> string
- observable:dataType —> string

5. UCO/CASE extensions

This Section describes all the ontologies extensions used in the project to represent significant artifacts that have not been approved yet by the UCO/CASE Community. At this aim the not-in-ontology namespace has been used to represent classes or properties not comprised in the official version of the ontologies. The UCO/CASE community uses a similar method by using the *drafting* name space. Probably the last version of the data ingested in the project's platform will convert the *not-in-ontology* namespace into the *drafting* namespace for compatibility with the UCO/CASE community data.

5.1. Automatic Number Plate Recognition

The Automatic Number Plate Recognition (ANPR) is a highly accurate system capable of reading vehicle number plates without human intervention Through the use of high speed image capture with supporting illumination, detection of characters within the images provided, verification of the character sequences as being those from a vehicle license plate, character recognition to convert image to text; so ending up with a set of metadata that identifies an image containing a vehicle license plate and the associated decoded text of that plate. The class is not part of the ontology, below the representation that has been defined in the in the data model:

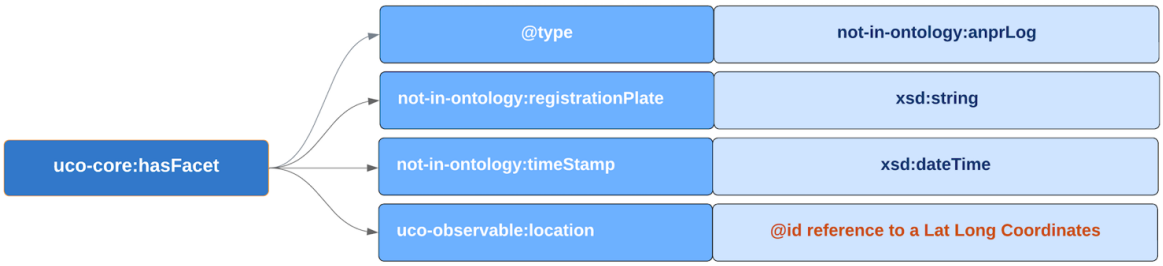

Figure 32: UCO/CASE Artifact ANPR

5.2. Calendar

A calendar entry facet is a grouping of characteristics unique to an appointment, meeting, or event within a collection of appointments, meetings, and events. The class is part of the latest UCO ontology, but it has been extended to include some relevant properties in the "*not-in-ontology*" name space.

Figure 33: UCO/CASE Artifact Calendar

An example of JSON-LD serialisation for the *Calendar Observable*: is as follows:

```
{
      "@id": "kb:f6ebbd3d-b300-488d-aecd-fa7f4fb36016",
      "@type": "uco-observable:ObservableObject",
      "uco-core:hasFacet": [
        {
          "@type": "uco-observable:ApplicationFacet",
          "uco-core:name": "Google Calendar"
        }
      ]
    },
{
      "@id": "kb:5973f597-3ccc-4611-828a-e85cec877894",
      "@type": "uco-observable:ObservableObject",
      "uco-core:hasFacet": [
        {
          "@type": "uco-observable:CalendarEntryFacet",
          "uco-observable:application": {
            "@id": "kb:f6ebbd3d-b300-488d-aecd-fa7f4fb36016"
          },
          "not-in-ontology:group": "INSPECTr project",
          "observable:subject": "Weekly technical meeting",
          "not-in-ontology:details": "Description of activities progress",
```

```
            "not-in-ontology:repeatInterval": "every week",
            "uco-observable:eventStatus": "Intact",
            "uco-observable:startTime": {
              "@type": "xsd:dateTime",
              "@value": "2020-06-19T10:30:00+00:00"
            },
            "uco-observable:remindTime": {
              "@type": "xsd:dateTime",
              "@value": "2020-06-19T10:00:00+00:00"
            },
            "uco-observable: repeatUntil": {
              "@type": "xsd:dateTime",
              "@value": "2023-02-28T10:30:00+00:00"
            },
            "uco-observable:endTime": {
              "@type": "xsd:dateTime",
              "@value": "2020-06-19T12:30:00+00:00"
            }
          }
        ]
      }
```


5.3. Cell Tower

A Cell Tower facet aims to represent mobile devices record details about cell sites they connect to, providing clues about the location of the device at a specific time. There are two basic technologies in mobile phones, CDMA and GSM. GSM stands for Global System for Mobile communication, while CDMA stands for Code Division Multiple Access GSM uses FDMA (Frequency division multiple access) and TDMA (Time division multiple access). GSM supports transmitting data and voice both at once, but CDMA does have not this feature. The main distinction between GSM and CDMA is that in GSM, the customer information is put on a SIM card which can be moved to a new mobile phone. Whereas only mobile phones from a set of whitelisted companies can be used with a CDMA network. To localize the sector of the base station (Cell ID) the following parameters can be used:

- GSM technology
- MCC — a Mobile Country Code. This code identifies the country.
- MNC - a Mobile Network Code. This code identifies the mobile operator.
- LAC - Location Area Code is a unique number of current location area. A location area is a set of base stations that are grouped together to optimize signalling.
- CID (Cell ID) — is a generally unique number used to identify each Base transceiver station (BTS) or sector of a BTS within a Location area code.
- CDMA technology
- NID – the Network Identification Number
- BID – the Billing Identification
- SID – the System Identification

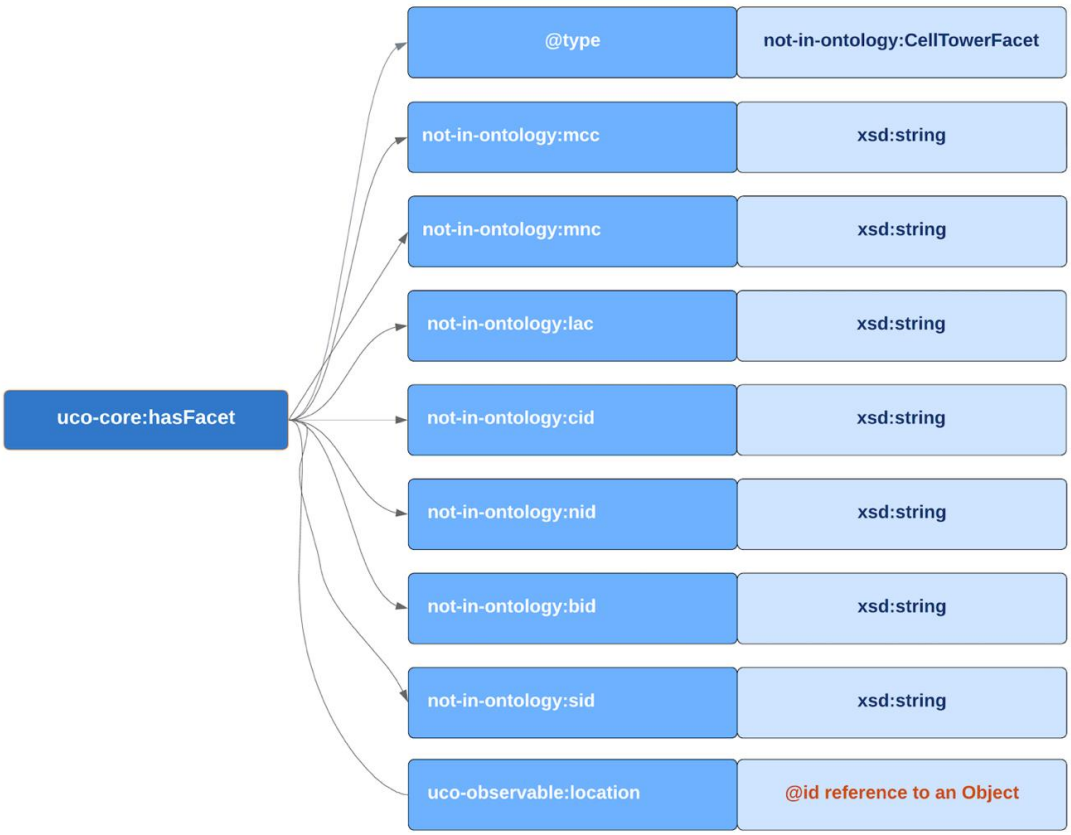

Figure 34: UCO/CASE Cell Tower Artifact

An example of JSON-LD serialisation for the *Cell Tower Observable*: is as follows:

```
{
        "@id": "kb:924b8c85-7016-439e-b6f8-962fb5af1496",
        "@type": "uco-observable:ObservableObject",
        "uco-core:hasFacet": [
          {
            "@type": "uco-location:LatLongCoordinatesFacet",
            "uco-location:latitude": {
              "@type": "xsd:decimal",
              "@value": "40.11259078"
            },
            "uco-location:longitude": {
              "@type": "xsd:decimal",
              "@value": "-75.65714263"
            },
            "uco-location:altitude": {
              "@type": "xsd:decimal",
              "@value": "0.0"
            },
            "not-in-ontology:locationType": "Cell Tower"
          }
        ]
      },
{
```

```
        "@id": "kb:76e023a7-149a-414d-8382-ccef631b4dce",
        "@type": "uco-observable:ObservableObject",
        "uco-core:hasFacet": [
          {
            "@type": "not-in-ontology:CellTowerFacet",
            "not-in-ontology:mcc": "272",
            "not-in-ontology:mnc": "01",
            "not-in-ontology:lac": "2953",
            "not-in-ontology:cid": "187589293",
            "uco-observable:location": {
              "@id": "kb:924b8c85-7016-439e-b6f8-962fb5af1496"
            }
          }
        ]
      }
```


5.4. Searched item

The class represents all the items searched through the use of a browser web.

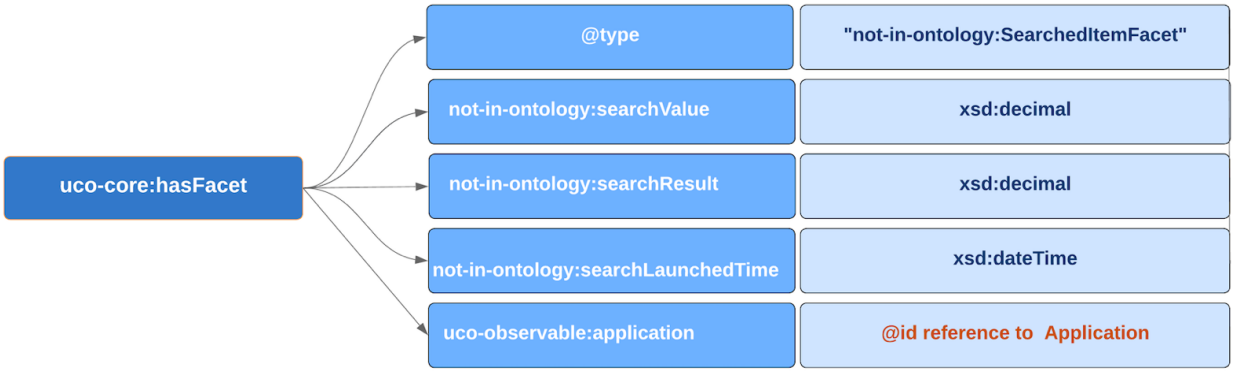

Figure 35: UCO/CASE Searched Item Artifact

An example of JSON-LD serialisation for the *Searched Item Observable*: is as follows:


```
{
        "@id": "kb:b4255f90-6cc4-487f-b894-232c1fc2303d",
        "@type": "uco-observable:ObservableObject",
        "uco-core:hasFacet": [
          {
            "@type": "not-in-ontology:SearchedItemFacet",
            "not-in-ontology:searchValue": "let it snow lyrics",
            "not-in-ontology:searchResult":
"https://www.youtube.com/watch?v=mSe6iZF_UfE;
https://www.youtube.com/watch?v=7hHOtPsNNNA;
https://www.youtube.com/watch?v=Km-Nlo5bgCQ;
https://www.41051.com/xmaslyrics/letitsnow.html; https://genius.com/Dean-
martin-let-it-snow-let-it-snow-let-it-snow-lyrics;
https://www.christmassongsandcarols.com/products/let-it-snow-lyrics",
            "not-in-ontology:searchLaunchedTime": {
              "@type": "xsd:dateTime",
```

```
            "@value": "2021-12-31T06:30:39+00:00"
          },
          "uco-observable:application": {
            "@id": "kb:6cf6d209-47e5-4962-aa6f-349764920d91"
          }
        }
      ]
    },
{
      "@id": "kb:6cf6d209-47e5-4962-aa6f-349764920d91",
      "@type": "uco-observable:ObservableObject",
      "uco-core:hasFacet": [
        {
          "@type": "uco-observable:ApplicationFacet",
          "uco-core:name": "Google Chrome"
        }
      ]
    }
```


5.5. Social media activity

The class represents all the data related to the activity done within a social network, such as the number of the reactions to a post, the number of shares, the comments and the author and the other users involved in these activities.

Figure 36: UCO/CASE Social Media Activity Artifact

An example of JSON-LD serialisation for the *Social Media Activity Observable*: is as follows:

```
{
        "@id": "kb:ffa320ed-4159-4d69-93f4-a4b07a31993d",
        "@type": "uco-observable:ObservableObject",
        "uco-core:hasFacet": [
          {
            "@type": "uco-observable:ApplicationFacet",
            "uco-core:name": "TikTok"
          }
        ]
},
{
        "@id": "kb:3e05a2bc-ce85-4a72-9c83-93e1c92ddfa9",
        "@type": "uco-observable:ObservableObject",
        "uco-core:hasFacet": [
          {
            "@type": "uco-observable:URLFacet",
            "uco-observable:fullValue":
"https://m.tiktok.com/v/6930762821563124998.html?u_code=dh0a11d1063hcc&pr
```

```
eview_pb=0&language=en&_d=dh0a06mjf77af8&share_item_id=693076282156312
4998"
          }
        ]
},
{
        "@id": "kb:1a06956d-5fd3-4d71-baf9-5cda18a0e86b",
        "@type": "uco-observable:ObservableObject",
        "uco-core:hasFacet": [
          {
            "@type": "not-in-ontology:SocialMediaActivityFacet",
            "uco-observable:body": "",
            "uco-observable:pageTitle": "",
            "not-in-ontology:authorIdentifier": "6925437509506696198",
            "not-in-ontology:authorName": "user156314810284",
            "not-in-ontology:reactionsCount": "158",
            "not-in-ontology:sharesCount": "120",
            "not-in-ontology:activityType": "Post",
            "not-in-ontology:commentCount": "86",
            "uco-observable:accountIdentifier": "Etabeta on Saturn",
            "uco-observable:observableCreatedTime": {
              "@type": "xsd:dateTime",
              "@value": "2021-02-19T00:19:41+00:00"
            },
            "uco-observable:application": {
              "@id": "kb:ffa320ed-4159-4d69-93f4-a4b07a31993d"
            },
            "uco-observable:url": {
              "@id": "kb:3e05a2bc-ce85-4a72-9c83-93e1c92ddfa9"
            }
          }
        ]
}
```

6. UCO/CASE representation of AI processing

In this section are included some special artifacts that are the AI processing result such as Data Mining, Stylometry and Natural Language Processing (NLP)[21]. Volume of information is increasing every day, therefore a system capable of extracting essence of information available and that can automatically generate report is needed. At the moment these artifacts are not yet represented in UCO/CASE, but there are proposals to include them in the ontology. The following sections are meant to indicate the most important AI processing taken into account within the project.

6.1. Knowledge Discovery in Database

Data Mining also known as Knowledge Discovery in Databases, refers to the non-trivial extraction of implicit, previously unknown and potentially useful information from data stored in databases. The Knowledge Discovery in Database (KDD) refers to a method of finding, transforming, and refining meaningful data and patterns from a raw database in order to be utilised in different domains or applications.

6.2. Machine Translation

Machine Translation involves the automatic translation of text from one natural language to another using a computer application. As the technology behind machine translation has progressed, new approaches have become available. One of the most common methods is the Statistical MT (SMT)[22]. This method uses large volumes of existing translated texts and statistical models to generate translations. However, it's quickly being overshadowed by other approaches because it's time and resource intensive. Another emerging approach is the Neural MT (NMT)[23]. This is a newer approach that is built on deep neural networks. It generally creates translations that are more fluent and grammatically accurate. However, it struggles translating rare words and terminology.

[21] NLP refers to the branch of computer science—and more specifically, the branch of artificial intelligence (AI) giving computers the ability to understand text and spoken words in much the same way human beings can. NLP combines computational linguistics—rule-based modeling of human language—with statistical, machine learning, and deep learning models. Together, these technologies enable computers to process human language in the form of text or voice data and to 'understand' its full meaning, complete with the speaker or writer's intent and sentiment.

[22] SMT learns how to translate by analysing existing human translations (known as bilingual text corpora). In contrast to the Rules-Based Machine Translation (RBMT) approach that is usually word-based, most modern SMT systems are phrase-based and assemble translations using overlap phrases. In phrase-based translation, the aim is to reduce the restrictions of word-based translation by translating whole sequences of words, where the lengths may differ. The sequences of words are called phrases, but typically are not linguistic phrases, but phrases found using statistical methods from bilingual text corpora.

[23] NMT is a state-of-the-art machine translation approach that utilises neural network techniques to predict the likelihood of a set of words in sequence. This can be a text fragment, complete sentence, or with the latest advances an entire document.

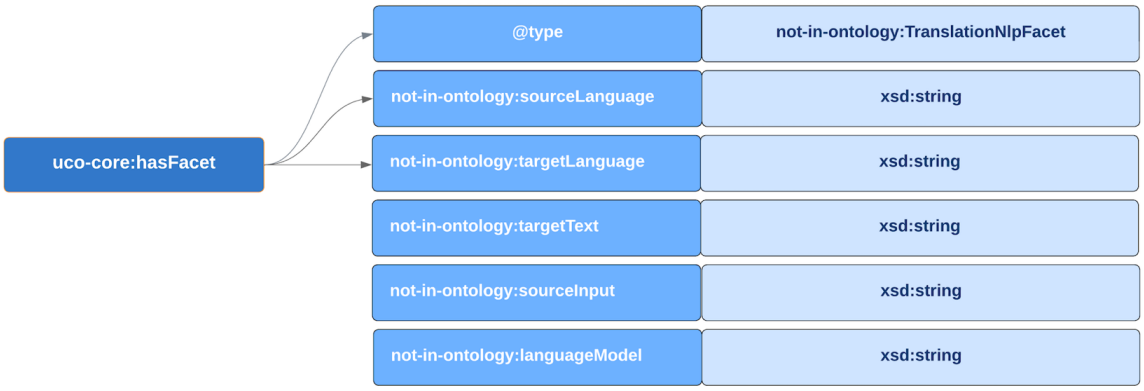

Figure 37: UCO/CASE TranslationNlp Artifact

6.3. Stylometry

Stylometry is the quantitative study of literary style through computational distant reading methods. It is based on the observation that authors tend to write in relatively consistent, recognizable and unique ways. The forensic stylometry model allows to identify the author of a text by their unique stylistic and linguistic "fingerprint". It needs a relatively long text to make an accurate identification[24].

[24] Further details can be found at https://programminghistorian.org/en/lessons/introduction-to-stylometry-with-python.

7. Examples of use in LLs and the associated benefits

UCO/CASE provides a structured (ontology-based) specification for representing information commonly analysed and exchanged by people and systems during investigations involving digital evidence. The primary purpose of UCO/CASE is to automate normalisation and combination of differing data sources to facilitate analysis and exploration of investigative questions (who, when, how long, where). To ensure all analysis results are traceable to their source(s), UCO/CASE keeps track of when, where and who used which tools to perform investigative actions on data sources, and what was the result

An investigation can involve many different tools and data sources, effectively creating separate silos of information. Manually pulling together information from these various data sources and tools is labour intensive, time consuming, and error prone. Tools that support UCO/CASE can extract and ingest data, along with their context, in a normalised format that can be automatically combined into a unified collection to strengthen correlation and analysis. This structured data and context increases visibility and insights across all information sources, giving analysts a more comprehensive and cohesive picture. UCO/CASE provides an enriched latticework of information, opening new opportunities for searching, pivoting, contextual analysis, pattern recognition, machine learning, and visualisation.

7.1. Living Labs feedback

Information related to "Examples of use in LLs and the associated Benefits, and lessons learned" has not been included in the deliverable as the LLs surveys were focused on technical/investigative issues and the questions are related to the following points:

- CMS (Case Management System) and CORTEX (an Observable Analysis and Active Response Engine.
- GAD (Gadget): The represents tools to process the elements of evidence, for instance the parsers developed to convert the XML reports generated by forensic tools in JSON-LD UCO/CASE standard.
- CSAM, TERRO, FRAUD. They refer to the Use Cases created by the LEA of the Consortium to understand if the investigators were able to answer specific investigative queries, by using the platform.
- GRELLI (Generic Reusable Embeddable Lightweight Widgets etc.). They are the widgets (table, word cloud, chart, map).
- Survey on AI tools. An example is “In your day-to-day tasks, do you see tools like Translation, Automatic Speech Recognition (ASR), Optical Character Recognition (OCR) or Named-Entity Recognition (NER) incorporated into your work pipeline?”.

They do not contain any specific questions/feedback from LEAs related to the UCO/CASE standardisation.

8. Conclusion

This deliverable has presented the main artifacts represented by using the Reference Framework for Standardisation of Evidence Representation and Exchange. An investigation generally involves a large number of digital devices, and, consequently, the amount of information to be analysed grows as the number of people involved in the investigation case increases. Therefore, each inquiry involves different subjects and organisations, different forensic tools and also different sources of evidence (i.e., mobile devices, disks, USB sticks, cloud data, etc.) generating, de facto, separate repositories of information, due to the incompatible proprietary formats produced by the forensic tools. Putting this data together, in manual mode, is time-consuming and demanding in terms of human resources, inevitably generating errors that can compromise the admissibility of an item of evidence. To effectively carry out an investigation involving digital devices, it is essential to harmonise the way information relevant to computer investigations is represented and exchanged.

Representing digital information collected during an investigation in a standardised manner solves, or at least addresses, one of the biggest problems investigators face when receiving relevant information from a variety of sources and in different formats. Within the forensic community, the need for a standard to represent the results of forensic processing and exchange the evidence thus obtained has become an increasingly pressing need as the complexity of investigations and digital devices involved in the cases under scrutiny has increased. The standard language UCO/CASE ontologies[25] meet these needs by enabling the processing of large volumes of information from different data sources and facilitating the development of sophisticated applications capable of finding correlations between different cases or within a single case, accurately and efficiently.

[25] See https://ontology.unifiedcyberontology.org/documentation/entities-tree-classes.html for all details on UCO/CASE ontologies classes.

References

- Casey E., Barnum S., Griffith R., Snyder J., van Beek H., Nelson A. (2017). Advancing coordinated cyber-investigations and tool interoperability using a community developed specification language. Digital Investigation 22:14–45.
- Casey E., Biasiotti M. A., Turchi F. (2017). Using Standardization and Ontology to Enhance Data Protection and Intelligent Analysis of Electronic Evidence. ICAIL 17 Proceedings of the 16th edition of the International Conference on Artificial Intelligence and Law, p 10.
- Francesconi E. (2012). Supporting Transnational Judicial Procedures be- tween European Member States: the e-Codex Project. In JURIX, vol 2012, p 41–50.
- Giardiello G., Turchi F. (2022) About Developing a Cross-Check System for Judicial Case Searching and Correlation. European Law Enforcement Research Bulletin - Special Conference Edition Nr. 6.
- Pangalos G., Salmatzidis I., Pagkalos I. (2014). Using IT to provide easier access to cross-border legal procedures for citizens and legal professionals-implementation of a European payment order E-CODEX pilot. In IJCA, vol 6, p 43.