Pacific Northwest
NATIONAL LABORATORY

PNNL-32414

Universal Utility Data Exchange (UUDEX) – Information Exchange Structures – Rev 1

Cybersecurity of Energy Delivery Systems (CEDS) Research and Development

December 2021

SR Mix
CM Schmidt
S Raju
MJ Rice
C Gonzales-Perez
S Sridhar
D Bharadwaj

U.S. DEPARTMENT OF ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-OE0000190

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor Battelle Memorial Institute, nor any of their employees, makes **any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights**. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062;
ph: (865) 576-8401
fax: (865) 576-5728
email: reports@adonis.osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
email: orders@ntis.gov <https://www.ntis.gov/about>
Online ordering: http://www.ntis.gov

Universal Utility Data Exchange (UUDEX) – Information Exchange Structures – Rev 1

Cybersecurity of Energy Delivery Systems (CEDS) Research and Development

December 2021

SR Mix
MJ Rice
S Sridhar
CM Schmidt
S Raju
C Gonzales-Perez
D Bharadwaj

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99354

Revision History

Revision	Date	Deliverable (Reason for Change)	Release #
00	9/30/2020	Initial release	Initial release
1	12/2021	Second Release	PNNL-32414

Summary

This report summarizes the information structures developed for the demonstration version of UUDEX. They serve as an initial proposal for information exchange models and structures to be proposed for standardization.

Note that this document only describes the models and structures of the information exchanges. The mechanics of the actual exchange, including routing and security are discussed in separate documents.

Acronyms and Abbreviations

ACL	Access Control List
ASCII	American Standard Code for Information Interchange, as defined by ISO/IEC 646
Avro	A data serialization system developed by *The Apache Software Foundation*
BA	Balancing Authority
CISA	Cybersecurity & Infrastructure Security Agency, an agency of the U. S. DHS
CIM	Common Information Model, as defined by EPRI, the Utility Communications Architecture Users Group and as used by IEC 61968 and 61970 series of standards
CN	Common Name field of an X.509 digital certificate
DHS	U. S. Department of Homeland Security
DOE	U. S. Department of Energy
EPRI	The Electric Power Research Institute
ICCP	Inter-control Center Communications Protocol, also known as TASE.2 or IEC 60870-6
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
ISO	International Standardization Organization
ITU	International Telecommunication Union, a specialized agency of the United Nations
JSON	JavaScript Object Notation, as defined by IETF RFC 7159
mRID	Master resource Identifier (as defined by CIM)
OE-417	U.S. DOE Electric Emergency Incident and Disturbance Report
OPC	Open Platform Communications
PDF	Portable Document Format, as specified in ISO 32000
PSIR	Physical Security Incident Report
RC	Reliability Coordinator
RFC	Request for Comment – Used in reference to published IETF standards, which are referred to as RFCs
SCADA	Supervisory Control and Data Acquisition
STIX	Structured Threat Information eXpression, as defined by the OASIS Cyber Threat Intelligence (CTI) Technical Committee
TASE	Telecontrol Application Service Element
TASE.2	Synonym for ICCP

TLP	traffic light protocol as defined by DHS CISA
TOP	Transmission Operator
URL	Universal Resource Locator
UTC-8	Unicode (or Universal Coded Character Set) Transformation Format – 8-bit as defined by ISO 10646
UUDEX	Universal Utility Data Exchange
UUID	Universally Unique IDentifier, as defined by IETF RFC 4122, which is technically equivalent to ITU-T Rec. X.667 and ISO/IEC 9834-8
XML	eXtensible Markup Language, as defined by the W3C

Contents

Figures

Tables

1.0 Introduction

All structures used for managing and transferring information using UUDEX are based on JSON. In order to optimize communication bandwidth, the native JSON structures may be transformed and transmitted using Apache Avro™.

All numbers specified are to be assumed to be decimal (base-10) unless otherwise specified.

Hexadecimal (base-16) numbers are represented by strings of the characters 0 through 9, and A through F, prefixed by 0x, with each character representing four bits of information. The hexadecimal number 0x41 is equivalent to the decimal value 65, or the ASCII character “A”.

Binary (base-2) numbers are represented by strings of the characters 0 and 1, prefixed by 0b, with each character representing one bit of information. The binary number 0b10 is equivalent to the decimal value 2.

The information structures are described using native JSON structures, with formal definitions using JSON schema[1], and Apache Avro schema[2],[3].

The JavaScript Object Notation (JSON) as defined by IETF RFC 8259 is a structured method of conveying information. It consists of sets of comma-separated name/value pairs organized as either individual objects or arrays of objects. The JSON structure does not enforce the order of objects within the structure. Complex object values (i.e., where the value itself is comprised of a set of name/value pairs) are enclosed in braces (“{}”), while arrays are enclosed in brackets (“[]”). Additional information on JSON is available in Appendix A.

By default, JSON field names and values are case sensitive. Field names are generally specified in “camelCase”[4], i.e., they capitalize the first letter of each word in the field name except for the first word. This convention is sometimes called “lowerCamelCase” to distinguish it from “PascalCase”, also called “UpperCamelCase”[5]. Acronyms within field names are capitalized, e.g., ACLDefinition.

Abbreviated field names are all lower case, i.e., the JSON field `powerSystemResourceMRID` may be abbreviated as `psrmrid`.

Some JSON field names are included by reference (STIX) or modeled after (CIM) existing naming constructs, and they adopt the case structures of those standards.

JSON structures and examples are represented using the `Courier` font (a fixed pitch font that allows proper display of indented structures).

---

[1] See https://json-schema.org/ accessed 9/14/2020
[2] See http://avro.apache.org/docs/current/ accessed 09/29/2020
[3] Note – the Apache Avro features have not been included in this version of the document.
[4] See https://www.theserverside.com/feature/A-guide-to-common-variable-naming-conventions for additional information, accessed 11/04/2020
[5] *Ibid*.

Static values and constants should all be specified in upper case, unless their accepted usage is not (e.g., kV). Multi-word constants such as "`BLOCK_NEW`" separate individual words with the underscore character ("_") referred to as "SCREAMING_SNAKE_CASE"[1]. This format is also used in the example power system element descriptive names.

Quoted string values such as descriptive names and text-based report fields should be accepted as mixed case.

Recipients of JSON formatted messages are encouraged, but not required, to accept static values of mixed case unless different cased values intentionally represent different values.

JSON strings are enclosed in double quotes. Some commonly used characters must be "escaped" inside the string quotes with a "\" (backslash) character:

Character	Escaped Character Representation
Double quote	\"
Back slash	\\
Slash	\/
Backspace	\b
Form feed	\f
New line	\n
Carriage return	\r
Tab	\t

Unless enclosed in quote characters, all "white space" characters (i.e., spaces, tabs, new lines, etc.) are ignored by JSON processing. White space and line formatting is presented in this document for reader clarity only; when the structures are transmitted, all the white space is removed to minimize the bandwidth required to transmit the message. An example of this is shown in Section 2.2.

[1] *Ibid*.

2.0 Basic Message Format

All UUDEX Messages (U-Messages) are expressed as JSON structures comprised of a “header” section and a payload section.

2.1 Basic Management Message Structures

As with all U-Messages, management messages consist of a header and a specific named management structure as illustrated by the following minimal JSON structure used for specifying management functions:

```
{
  "header":{},
  "payload section":{}
}
```

The header section is identified using the literal string "`header`", whereas the "*`payload section`*" is identified by specifying one of several payload section identifier strings.

The payload component for the management structure defines either the subject create structure or the ACL structure.

Management structures consist of a header, and the specific named management structure as the payload. These management payload structures are defined in Section 5.0. Currently, management structure payloads are:

- "`subjectCreate`" - for either creating a subject with optional resource constraints and access permissions
- "`ACLDefinition`" - for defining an access control list (ACL) that is used when defining access permissions.

Management payload structures are described in detail Section 5.0.

An example management structure is as follows:

```
{
  "header":{
    "messageID":"4599cadb-aa7b-438e-8b80-bbd8eb232eab",
    "noun":"subjectCreate",
    "verb":"CREATE",
    "origin":"RC",
    "timeStamp":"2020-07-29 10:12:09.209124",
    "version":"1.0",
    "hashType":"SHA-256",
    "hash":"200d5e6d32ac781860038d35728180674ce374fde9c409a0e9f149171567fbae"
  },
  "subjectCreate":{
    "schema":"https://www.uudex.org/uudex/0.1/SubjectPolicy",
    "schemaVersion":"0.1",
    "owner":"Jane",
    "dataType":"STIXElements",
    "action":"ALLOW",
```

```
    "constraints":{
      "maxPriority":3,
      "maxMessageCount":20,
      "broadestAllowedPublisherAccess":{
        "allowOnly":[
          "Bob",
          "Mary",
          "Fred",
          "Paul",
          "Carl"
        ]
      },
      "broadestAllowedSubscriberAccess":{
        "allowExcept":[
          "Jerk",
          "Bad Guy"
        ]
      },
      "broadestAllowedManagerAccess":{
        "allowOnly":[
          "Trusted",
          "Mary"
        ]
      }
      "broadestAllowedDiscoveryAccess":{
        "allowExcept":[
          "Jerk",
          "Bad Guy"
        ]
      }
    }
  }
}
```

2.2 Basic Information Transfer Structures

The payload for information transfers is described in Section 7.0, and is comprised of a header and payload, where the payload always consists of a dataSet containing one or more dataElements, each representing a discrete item (measured value, message, file, etc.) that is to be transferred from a UUDEX Publisher (U-Publisher) to one or more UUDEX Subscriber (U-Subscriber) clients. All the dataElements within a dataSet must be of the same type, as indicated by the noun field of the header.

A minimal JSON structure for UUDEX information transfers would be:

```
{
  "header":{},
  "payload section":{}
}
```

The specification of the *payload section* is different depending on the contents and format of the particular payload. For example, for power system data (i.e., data formerly transmitted by ICCP) the payload is a "dataSet" containing "dataElements" (as described in Section 6.1). Other information exchanges also use the dataElements structure to contain, for example,

OE-417 report forms (as described in Section 6.4.1), exchange of threat, vulnerability, or patch information (as described in Section 6.4.3, Section 6.4.4, and Section 6.4.5). or power system models (as described in Section 6.5),

However, to be syntactically correct, the header and payload structures would each need to contain valid structures, as shown in the following example for exchange of a telemetered power system measurement value:


```
{
  "header":{
    "messageId":"d5d1c892-974a-11e9-b198-b0c090affff",
    "noun":"powerSystemResource",
    "subject":"RC/ICCPData/GEN1",
    "origin":"TransOp1",
    "source":"RC",
    "destination":"TransOp2",
    "timeStamp":"2020-07-29 10:12:09.209124",
    "verb":"CREATE",
    "version":"1.0",
    "hashType":"SHA-256",
    "hash":"9d724973799dc04a5adf71520c91c1331dd0c0fbb940d8f4330f2c45a7d36c8e"
  },
  "dataSet":{
    "dataElements":[
      {
        "powerSystemResource":{
          "powerSystemResourceMRID":"GEN1",
          "powerSystemResourceName":"Generator1",
          "powerSystemResourceAliasName":"",
          "acdcTerminal":{
            "acdcTerminalMRID":"",
            "acdcTerminalName":"",
            "connected":true
          },
          "measurements":[
            {
              "measurementMrid":"GEN1_8323",
              "measurementName":"GEN1-8223_MW",
              "measurementType":"ANALOG",
              "unitSymbol":"MW",
              "unitMultiplier":1,
              "readingType":"15.8.6.1.0.8.0.0.0.6.38",
              "values":[
                {
                  "value":56.5,
                  "quality":{
                    "currentSource":"TELEMETERED",
                    "normalSource":"TELEMETERED",
                    "normalValue":"300.0",
                    "validity":[
                      "VALID"
                    ],
                    "selection":"PRIMARY"
                  },
                  "timeStamp":{
                    "quality":"VALID",
```

```
                    "value":"2020-07-06 09:35:46.305-05:00"
                  },
                  "covCounter":0,
                  "alarms":[
                    {
                      "state":"",
                      "acknowledged":"",
                      "returnToNormal":"",
                      "alarmCovCounter":0
                    }
                  ],
                  "inhibit":[
                    "SCAN",
                    "CONTROL"
                  ]
                }
              ]
            }
          ]
        }
      }
    ]
  }
}
```


2.2.1 Message Sizes

Note that tabs and unquoted white space is only necessary for humans to read the JSON message. The following JSON code is functionally identical to the above example:


```
{"header":{"messageId":"d5d1c892 974a 11e9 b198 b0c090affff","noun":"powerSys
temResource","subject":"RC/ICCPData/GEN1","origin":"TransOp1","source":"RC","
destination":"TransOp2","timeStamp":"2020 07 29 10:12:09.209124","verb":"CREA
TE","version":"1.0","hashType":"SHA 256","hash":"9d724973799dc04a5adf71520c91
c1331dd0c0fbb940d8f4330f2c45a7d36c8e"},"dataSet":{"dataElements":[{"powerSyst
emResource":{"powerSystemResourceMRID":"GEN1","powerSystemResourceName":"Gene
rator1","powerSystemResourceAliasName":"","acdcTerminal":{"acdcTerminalMRID":
"","acdcTerminalName":"","connected":true},"measurements":[{"measurementMrid"
:"GEN1_8323","measurementName":"GEN1-8223_MW","measurementType":"ANALOG","uni
tSymbol":"MW","unitMultiplier":1,"readingType":"15.8.6.1.0.8.0.0.0.6.38","val
ues":[{"value":56.5,"quality":{"currentSource":"TELEMETERED","normalSource":"
TELEMETERED","normalValue":"300.0","validity":["VALID"],"selection":"PRIMARY"
},"timeStamp":{"quality":"VALID","value":"2020 07 06 09:35:46.305 05:00"},"co
vCounter":0,"alarms":[{"state":"","acknowledged":"","returnToNormal":"","alar
mCovCounter":0}],"inhibit":["SCAN","CONTROL"]}]}]}}]}}
```


Removing the unnecessary white space reduced the JSON structure from 2,128 bytes to 1,109 bytes, a reduction of 48% with no loss of function or ability to interpret the information.

Further reduction in the size of the transmitted message can be accomplished by not including optional fields that do not contain values, for example not including the `powerSystemResourceAliasName` field or the `acdcTerminal` structure, or by only including fields with static information, such as the field `powerSystemResourceName`, infrequently, say every 10 messages.

Even further reduction can be accomplished by using standardized abbreviations for static quoted string values, e.g., "T" for "TELEMETERED", or standardized abbreviated field names, e.g., "psrmrid" for "powerSystemResourceMRID".

Using these reduction techniques results in a JSON code that can be expressed as:


```
{
  "header":{
    "messageId":"d5d1c892-974a-11e9-b198-b0c090affff",
    "noun":"psr",
    "subject":"RC/ICCPData/GEN1",
    "origin":"TransOp1",
    "source":"RC",
    "destination":"TransOp2",
    "timeStamp":"2020-07-29 10:12:09.209124",
    "verb":"CREATE",
    "version":"1.0",
    "hashType":"SHA-256",
    "hash":"0e731b890928319d1f7d95802bf8e232b777b7373b030edd310986fca0296c89"
  },
  "ds":{
    "de":[
      {
        "psr":{
          "psrmrid":"GEN1",
          "meas":[
            {
              "mrid":"GEN1_8323",
              "val":[
                {
                  "v":56.5,
                  "q":{
                    "cs":"T",
                    "val":[
                      "V"
                    ]
                  },
                  "ts":{
                    "q":"V",
                    "v":"2020-07-06 09:35:46.305-05:00"
                  }
                }
              ]
            }
          ]
        }
      }
    ]
  }
}
```


or


```
{"header":{"messageId":"d5d1c892-974a-11e9-b198-b0c090affff","noun":"psr","su
bject":"RC/ICCPData/GEN1","origin":"TransOp1","source":"RC","destination":"Tr
ansOp2","timeStamp":"2020-07-29 10:12:09.209124","verb":"CREATE","version":"1
```

```
.0","hashType":"SHA-256","hash":"0e731b890928319d1f7d95802bf8e232b777b7373b03
0edd310986fca0296c89"},"ds":{"de":[{"psr":{"psrmrid":"GEN1","meas":[{"mrid":"
GEN1_8323","val":[{"v":56.5,"q":{"cs":"T","val":["V"]},"ts":{"q":"V","v":"202
0-07-06 09:35:46.305-05:00"}}]}]}}]}}
```


For a size of 1024 bytes in human-readable format, or 525 bytes in compact format.

Adding a second measurement to this reduced format produces:


```
{
  "header":{
    "messageId":"d5d1c892-974a-11e9-b198-b0c090affff",
    "noun":"psr",
    "subject":"RC/ICCPData/GEN1",
    "origin":"TransOp1",
    "source":"RC",
    "destination":"TransOp2",
    "timeStamp":"2020-07-29 10:12:09.209124",
    "verb":"CREATE",
    "version":"1.0",
    "hashType":"SHA-256",
    "hash":"4e8eb7a0c0decedcb161becd5b8ac1dfdc98ca1bfa3f3869aeb95b348f983dbe"
  },
  "ds":{
    "de":[
      {
        "psr":{
          "psrmrid":"GEN1",
          "meas":[
            {
              "mrid":"GEN1_8323",
              "val":[
                {
                  "v":56.5,
                  "q":{
                    "cs":"T",
                    "val":[
                      "V"
                    ]
                  },
                  "ts":{
                    "q":"V",
                    "v":"2020-07-06 09:35:46.305-05:00"
                  }
                }
              ]
            }
          ]
        }
      },
      {
        "psr":{
          "psrmrid":"GEN2",
          "meas":[
            {
              "mrid":"GEN1_8324",
```

```
                "val":[
                  {
                    "v":60.5,
                    "q":{
                      "cs":"T",
                      "val":[
                        "V"
                      ]
                    },
                    "ts":{
                      "q":"V",
                      "v":"2020-07-06 09:35:46.305-05:00"
                    }
                  }
                ]
              }
            ]
          }
        }
      ]
    }
}
```


or


```
{"header":{"messageId":"d5d1c892-974a-11e9-b198-b0c090affff","noun":"psr","su
bject":"RC/ICCPData/GEN1","origin":"TransOp1","source":"RC","destination":"Tr
ansOp2","timeStamp":"2020-07-29 10:12:09.209124","verb":"CREATE","version":"1
.0","hashType":"SHA-256","hash":"4e8eb7a0c0decedcb161becd5b8ac1dfdc98ca1bfa3f
3869aeb95b348f983dbe"},"ds":{"de":[{"psr":{"psrmrid":"GEN1","meas":[{"mrid":"
GEN1_8323","val":[{"v":56.5,"q":{"cs":"T","val":["V"]},"ts":{"q":"V","v":"202
0-07-06 09:35:46.305-05:00"}}]}]}},{"psr":{"psrmrid":"GEN2","meas":[{"mrid":"
GEN1_8324","val":[{"v":60.5,"q":{"cs":"T","val":["V"]},"ts":{"q":"V","v":"202
0-07-06 09:35:46.305-05:00"}}]}]}}]}}
```


For a size of 1597 bytes in human-readable format, or 663 bytes in compact format, an increase of 138 bytes for the second measurement.

The removal of extraneous white space is a step towards formal canonization of the JSON. Full canonization would include ordering the individual name-value pairs alphabetically by their name, and some additional processing of strings. See IETF RFC 8785 for additional information.

Finally, additional compression could be accomplished by using Apache Avro to process and compress the JSON structures for transport when published and perform a similar decompression process when messages are received by subscribers.

2.3 Response Information Structures

The payload for responses transfers is comprised of just an abbreviated header indicating the response and response status. For example,

The JSON structure for UUDEX response would be:

```
{
  "header":{
    "verb":"CREATED",
    "messageId":"14a58f95-c7a4-4263-bd4f-6472c2c78568",
    "correlationId":"d5d1c892-974a-11e9-b198-b0c090affff",
    "status":"",
    "timeStamp":"2020-07-29 10:12:09.209124",
    "version":"1.0",
  }
}
```

In this example, the `verb` is the past tense of the verb the U-Message is responding to, the `correlationId` is the `messageId` of the message being responded to, and the status is the response status, generally "`SUCCESS`" or a string indicating what the failure was.

Note that there is no hash since there is no payload.

2.4 Flexibility in Information Structures

When processing information elements in UUDEX, applications should follow the robustness principle known as Postel's Law: "be conservative in what you do, be liberal in what you accept from others"[1], sometimes rewritten as "be conservative in what you send, be liberal in what you accept". Following this guidance, U-Publishers may start publishing information that may not be understood completely by U-Subscribers, for example a new `powerSystemElement dataElement` (i.e., a power system element identified by a CIM mRID [common information model master resource identifier] that the subscriber cannot map to a point in its system) without coordinating with the subscriber – the subscriber may simply ignore that particular information block while processing all others. When the receiving subscriber has configured its system to understand the new CIM mRID, it is ready to receive and process the new fields with no coordination or configuration update required – the new information is already being transferred.

The flexibility of the JSON structure also allows for additional properties to be included. If they are to be processed, their interpretation must be agreed to by both the publisher and subscriber. If the subscriber cannot interpret them, it is free to ignore them. Note that this flexibility applies at all levels of the JSON structures.

[1] Attributed to Jon Postel in the specification for Transmission Control protocol (TCP), RFC 761.

3.0 Message Header

The header of each message contains information about the message, including the source of the information and actions to be performed on the information by UUDEX. The header information is modeled after the CIM Common Message Envelope as described in IEC 61968-100. Management structures use a simplified version of the message header that is described following the structure used for information transfers

3.1 Information Message Header

The structure of the message header used for information transfers is:

```
{
  "header":{
    "messageID":"",
    "noun":"",
    "verb":"",
    "subject":"",
    "origin":"",
    "source":"",
    "destination":"",
    "timeStamp":"",
    "correlationID":"",
    "context":"",
    "user":"",
    "comment":"",
    "encryption":"",
    "encoding":"",
    "compression":"",
    "properties":"",
    "sensitivity":"",
    "schemaVersion":"",
    "schema":"",
    "replyAddress":"",
    "asyncReplyFlag":"",
    "ackRequired":"",
    "ackReply":"",
    "status": "",
    "expiration":"",
    "hashType":"",
    "hash":"",
    "properties":""
  }
}
```

Table 3.1 provides an overview and explanation of the fields used in the header for information transfers.

Table 3.1: Message Header Field Descriptions

Field	Required / Optional / Recommended	Description
`header`	required	Specifies this is a JSON structure containing metadata about the entire message
`messageID`	required	Unique identifier of the message. This is a Universally Unique Identifier (UUID)..
`noun`	required	Identifies the type of data elements being exchanged. The noun in the header must match the `dataElementType` in the payload tying them together. Note that the `dataElementType` may be abbreviated in some cases, so the `noun` should be similarly abbreviated. U-Publisher Endpoints and U-Consumer Endpoints should always treat the full and abbreviated terms as equivalent.
`verb`	required	Identifies the action to be taken. – create, change, close, cancel, delete etc. For example: • "`CREATE`" • "`CHANGE`" • "`CLOSE`" • "`CANCEL`" • "`DELETE`" • "`ACKNOWLEDGE`" / "`ACKNOWLEDGED`" / "`NACKED`" See Section 3.1 and Table 3.3 for additional information.
`subject`	required	The UUDEX Subject (U-Subject) to which the message is being published.
`origin`	optional	Specifies the name of the UUDEX Participant (U-Participant) original "owner" of the information contained in the message (this could be the name of an organization or node further "upstream" from the U-Publisher node). This is only required of the origin and source are different; if not supplied, the "source" is assumed to be the original owner of the information. Note that the content of this field is informational only.
`source`	required	Specifies the name of the U-Participant publisher UUDEX Instance (U-Instance) of the message which could be the same as the origin for information created and owned by the publisher node/organization For UUDEX Servers (U-Servers) that act as intermediaries, for example a local Reliability Coordinator (RC) that collects measurement data from its Transmission Operator (TOP) members, and shares the information with a neighboring RC, the `origin` would be the

Field	Required / Optional / Recommended	Description
		TOP, while the `source` would be the local RC. Note that the content of this field is informational only. The actual source of the message is contained in the identity field of the X.509 certificate used to authenticate the transfer.
`destination`	optional	Specifies the name of the U-Endpoint to which the information is ultimately being sent. While this is typically the U-Endpoint instance receiving the information, in a bridged environment (i.e., where a U-Endpoint spans multiple UUDEX Infrastructures (U-Infrastructures) or U-Instances, possibly of different versions), the `destination` represents the U-Endpoint to which the bridge U-Endpoint should send the message.
`timeStamp`	required	The message timestamp, formatted using ISO 8601, representing when the U-Message was published.
`correlationID`	optional	Unique identifier used to map related messages. This is the `messageID` of a previous message to which this message is related. An example would be if this message was in response to a query posed by a previous message – that query `messageID` would be stored in `correlationID` to allow the two messages to be correlated.
`context`	optional	The context is used to logically segregate messages that might be used for production, testing, or other purposes. If the field is not specified or is blank or null, "`PROD`" is assumed. Entries include: • "`PROD`" – indicates that the message and payload values are intended to be processed normally (i.e., by production processes or servers) • "`DEV`" – indicates the message and payload values are used for operations development*. • "`TEST`" – indicates the message and payload values are used for operations testing*. • "`TRAIN`" – indicates the message and payload values are used for operations training*. • "`SIM`" – indicates that the message and payload values contain simulated information. • "`UTEST`" – indicates that the message and payload values are to be used for testing or diagnosis of UUDEX functionality, performance, connectivity, etc. • "`EXERCISE`" – indicates that the message and payload values are associated with an exercise and must be processed by systems involved in the exercise, not production or other routine use systems such as testing, training, simulation, or development.

Field	Required / Optional / Recommended	Description
		* “operations” here means that the development, testing, or training contexts are for end-use application environments as opposed to the “UTEST” context that is used to test internal UUDEX functionality. Other values for context must be mutually agreed to by the U-Publisher all U-Subscribers.
user	optional	The user that is responsible for the initiation of the information exchange. The user may only make sense for certain dataSets, as mutually agreed to by the U-Publisher all U-Subscribers. Note that the content of this field is informational only.
comment	optional	A comment is text entered for documentation or diagnostic purposes.
encryption	optional	Specifies whether the entire message payload is encrypted. If not specified, or specified as “NONE”, the payload is not encrypted; otherwise, specifies the method used to encrypt the payload. When specified, the contents of the dataSet{} structure (the entire contents inside the braces, but not including them) is encrypted using the indicated methodology. Specific values for the encryption field are not specified, nor are the mechanisms for encryption key management, which should occur outside the scope of UUDEX. If additional parameters are required, they may be specified as custom extensions to the standard U-Message header. If encryption is specified as anything other than “NONE”, encoding is required, and compression is recommended.
encoding	optional	The encoding used to transfer the information. Supported values are “ASCII” (ISO/IEC 646) or “UTF-8” (ISO 10646) for direct XML transfers; and “BASE64”, or “BASE85” for encoding binary values into printable ISO/IEC 646 characters for transport. When specified, the contents of the dataSet{} structure (the entire contents inside the braces, but not including them) is encoded. If compression is used, it must be performed before the encoding to ensure that all transmitted characters are ISO/IEC 646. If not specified, “ASCII” is assumed, and the native format of the contents *must* be ISO/IEC 646 characters. See Appendix D for additional information.
compression	optional	Compression method used to shrink the information prior to encoding it. Supported values are “NONE”, “ZLIB” and “GZIP” (note ZLIB and GZIP represent the same algorithm – gzip is a stand-alone program that implements the zlib

Field	Required / Optional / Recommended	Description
		compression library). Additional methods such as "LZ4", "LZ77", "LZ78", "LZO", "LZSS" "LZFSE", "LZVN", "LZW", "DEFLATE", "BZIP2", "LZMA", "LZMA2", "PPM", and "RLE", may be used by mutual agreement of both the publisher and subscriber. When specified, the contents of the dataSet{} structure (the entire contents inside the braces, but not including them) is compressed. If not specified, "NONE" is assumed. See Appendix D for additional information.
sensitivity	optional	Specifies the sensitivity level of the message. If not specified, or specified as "NONE", the default value of not sensitive is used. Other values could use the TLP (traffic light protocol) designations[1] formatted as "TLP-WHITE", or other mutually agreeable set of values. Note that enforcement of the sensitivity designation is the responsibility of the publishing and consuming U-Endpoints.
schema	optional	Reference to the JSON schema for the header.
schemaVersion	recommended	The version of the JSON structure used to define the header. This may be a number or letter (e.g., "1.0" or "A"), and may indicate a site-specific version by as a string of characters (e.g., "RC-V1").
replyAddress	optional	The U-Instance to receive a reply when using the request/reply feature of UUDEX
asyncReplyFlag	optional	A Boolean indication that the U-Subscribing client should acknowledge receipt of the message to the U-Publisher client by publishing an "ACKNOWLEDGED" response message to the replyAddress U-Subject specifying the messageID of the received message in the correlationID field. If the replyAddress field is blank or null, no response is sent.
ackRequired	optional	A Boolean indication that the U-Server should acknowledge receipt of the message to the U-Publisher client by publishing a "past tense" verb response message to the replyAddress U-Subject specifying the messageID of the received message in the correlationID field. If the replyAddress field is blank or null, no response is sent. For example, if the U-Publish verb was "CREATE", the response should use the verb "CREATED", and the status of the "CREATE" request would be transmitted in the ackReply field.
ackReply	optional	A Boolean indication required and used only when the ackRequired flag is set to true that indicates the status of

[1] Refer to the DHS / US-CERT CISA TLP description noted in the references section

Field	Required / Optional / Recommended	Description
		the request. For example, if the verb was "CREATE", the message was published successfully, and the ackRequired was true, the ackReply response would be true. If the message was not successfully published, for example because the queue was full, the ackReply response would be false.
status	optional	Only used when the verb is "ACKNOWLEDGED" or "NACKED". Contains the status of the acknowledgement or negative acknowledgement.
expiration	optional	The time, formatted using ISO 8601, after which the message should no longer be considered valid. If supported by the underlying message protocol, the message can be automatically deleted prior to a consume request by a subscriber. If the message cannot be deleted prior to the consume action, the message should be deleted at the time of the consume request and not delivered to the subscriber.
hashType	recommended	The format of the hash field, for example, "SHA-256" or "MD5". Cryptographic hashes should be vetted and should not contain vulnerabilities or back doors. See NIST FIPS-140-2 Annex A (latest version) for an expanded list of acceptable hash algorithms. If the hashType field is not specified or is blank or null, no hash is supplied, and the payload is not checked for tampering or corruption.
hash	recommended	The cryptographic hash of the payload component of the message, e.g., the contents of the "dataSet":{} structure (the entire structure between the braces, not including them), when expressed in "compact" form, i.e., a continuous stream of characters without line breaks or indents for readability, except for strings that are enclosed in quotes. If the hashType field is not blank, this field is required.
properties	optional	Any JSON object (or JSON structure) that is mutually agreed to by the U-Publisher and all U-Subscribers. For example, a custom object could be defined that defines the parameters of a test when using a "TEST" context, the name of the specific exercise or exercise step when using an "EXERCISE" context, or encryption parameters when specifying encryption.

Note that the message header is where the U-Subject or origin are specified, so if information corresponding to different U-Subjects or origins are to be transmitted (for example, a (RC) sending information from multiple TOPs), they must be in different published messages. The flexibility of the U-Message structure allows multiple messages with the same subject but different contents to be published to allow, for example, sending transmission line flows and bus voltages in the same U-Message, or sending telemetered, estimated, and substituted values for the same power system element voltages in the same U-Message.

3.2 Management Message Header

The simplified message header used for management structures is:

```
{
  "header":{
    "messageID":"",
    "noun":"",
    "verb":"",
    "origin":"",
    "replyAddress":"",
    "ackRequired":"",
    "ackReply":"",
    "timeStamp":"",
    "hashType":"",
    "hash":""
  }
}
```

Most fields in the management header are the same as for information transfers, with the following exceptions or explanations. Table 3.2 provides an overview and explanation of the fields used in the simplified header structure used for management structures.

Table 3.2: Simplified Message Header Field Descriptions

Field	Required / Optional / Recommended	Description
`header`	required	Specifies this is a JSON structure containing metadata about the entire message
`messageID`	required	Unique identifier of the message.
`noun`	required	Specifies the management action to be taken, i.e., "`subjectCreate`" or "`ACLDefinition`"
`verb`	required	Identifies the action to be taken. – create, , close, cancel, delete etc. For example: • "`CREATE`" • "`CLOSE`" • "`CANCEL`" • "`DELETE`" • "`ACKNOWLEDGE`" / "`ACKNOWLEDGED`" / "`NACKED`" See Section 3.5 and Table 3.3 for additional information.
`origin`	required	The U-instance requesting the management action. Note that the content of this field is informational only.

Field	Required / Optional / Recommended	Description
`replyAddress`	optional	The U-instance to receive a reply when using the `ackRequired` feature of UUDEX
`ackRequired`	optional	A Boolean indication that the U-Server should acknowledge receipt of the message to the U-Publisher client by publishing a “past tense” verb response message to the `replyAddress` U-Subject specifying the `messageID` of the received message in the `correlationID` field. If the `replyAddress` field is blank or null, no response is sent. For example, if the U-publish verb was “`CREATE`”, the response should use the verb “`CREATED`”, and the status of the “`CREATE`” request would be transmitted in the `ackReply` field.
`ackReply`	optional	A Boolean indication required and used only when the `ackRequired` flag is set to true that indicates the status of the request. For example, if the verb was “`CREATE`”, the action was successful, and the `ackRequired` was true, the `ackReply` response would be true. If the message was not successfully published, for example because the requestor is not authorized to perform the requested action, the `ackReply` response would be false.
`timeStamp`	required	The message timestamp, formatted using ISO 8601, representing when the U-Message was published. Supplied by the U-publish API.
`hashType`	recommended	The format of the hash field, for example, “`SHA-256`”. Cryptographic hashes should be vetted and should not contain vulnerabilities or back doors. See NIST FIPS-140-2 Annex A (latest version) for an expanded list of acceptable hash algorithms. If the `hashType` field is not specified or is blank or null, no hash is supplied, and the payload is not checked for tampering or corruption.
`hash`	required	The cryptographic hash of the payload component of the message, e.g., the entire "`dataSet":{}`", when expressed in “compact” form, i.e., a continuous stream of characters without line breaks or indents for readability, except for strings that are enclosed in quotes. If the `hashType` field is not blank, this field is required.

3.3 Response Message Header

The response message also uses a simplified message header:

```
{
  "header":{
    "messageID":"",
```

```
    "noun":"",
    "verb":"",
    "origin":"",
    "timeStamp":"",
    "response":"",
    "error":""
  }
}
```

Note that there is no payload section for the response message.

Most fields in the response header are the same as for information transfers, with the following exceptions or explanations. Table 3.3 provides an overview and explanation of the fields used in the simplified header structure used for management structures.

Since these messages are replies, they use the past-tense verbs, i.e., "`CREATED`" is the response to a "`CREATE`" request

Table 3.3: Simplified Message Header Field Descriptions

Field	Required / Optional / Recommended	Description
`header`	required	Specifies this is a JSON structure containing metadata about the entire message
`messageID`	required	Unique identifier of the message associated with this response. This is a UUID generated by the U-Publish API.
`noun`	required	Specifies the `dataElementType` or management action associated with this response.
`verb`	required	Identifies the action that was taken. – created, changed, closed, cancelled, deleted etc. For example: • "`CREATED`" • "`CLOSED`" • "`CANCELLED`" • "`DELETED`" • "`ACKNOWLEDGED`" / "`NACKED`" See Section 3.1 and Table 3.4 for additional information.
`origin`	required	The U-instance that requested the action. Note that the content of this field is informational only.
`timeStamp`	required	The message timestamp, formatted using ISO 8601, representing when the response U-Message was published. Supplied by the U-publish API.

Field	Required / Optional / Recommended	Description
`response`	required	The high-level response for the action. Valid values are” • “`OK`” – the request completed successfully • “`FAILED`” – the request did not complete successfully. Additional information may be provided in the `status` field. • `PARTIAL`” – the request was only partially completed. Additional information may be provided in the `status` field.
`status`	optional	Provides additional information for `FAILED` or `PARTIAL` responses.

3.4 Message Header Extensions

In either case, the header may also optionally contain additional JSON fields or structures that may be used to further describe the data set:

```
{
  "header":{
    "messageID":"",
    "noun":"",
    "verb":"",
    "origin":"",
    "timeStamp":"",
    "property1":"",
    "property2": {
      "property3":"",
      "property4":""
    },
    "hashType":"",
    "hash":""
  }
}
```

If additional properties are included, their interpretation must be agreed to by both the publisher and subscriber. If the subscriber cannot interpret them, it is free to ignore them. Note that this flexibility applies at all levels of the JSON structures (including the header and payload).

3.5 Verbs

Verbs used in UUDEX are modeled after those described in IEC 61968-1 Annex A and IEC 61968-100 Annex B.

The verb field in the message header directs how the U-Server is to process the message. the most common verb is “`CREATE`”, which is used to publish information for eventual subscription fulfillment and transmission to users for their use. Other verbs are described in Table 3.4

Note that although the verbs are described using all capital letters, the verbs are to be interpreted in case-blind mode, i.e., "CREATED", "Created", "created", and "cReAtEd" should all be treated as being the same verb.

Table 3.4: UUDEX Message Verbs

Verb	Description
CREATE	Requests the U-Server to process the published message and make it available for subscription fulfillment For subject creation requests, this requests the creation of the specified subject with parameters specified in the message payload. For subscribe requests, this requests a new subscription to be created with the parameters specified in the message payload.
CREATED	Response from U-Server to the CREATE request. Only used if the ackRequired flag is set to true and the replyAddress field is not blank or null.
CLOSE	Requests the U-Server terminate the connection between it and the U-Endpoint. This is functionally equivalent to the U-Endpoint terminating the connection itself.
CLOSED	Response from U-Server to the CLOSE request. The connection is closed after this message is delivered.
CANCEL	Requests the U-Server to cancel a pending subscription request (i.e., perform an "unsubscribe" action). Can also request UUDEX to cancel a pending publish request that has not been fulfilled by a subscription (essentially an "uncreate"). This makes the most sense for message publications that are served by a notify process.
CANCELLED	Response from U-Server to the CANCEL request. Only used if the ackRequired flag is set to true and the replyAddress field is not blank or null.
DELETE	Requests the U-Server to delete an existing message (identified by the messageID) in the U-Repository (if it is still resident in the U-Server). For subject requests, this requests the specified subject and all messages pending delivery be deleted.
DELETED	Response from U-Server to the DELETE request. Only used if the ackRequired flag is set to true and the replyAddress field is not blank or null.
QUERY	Requests the U-Server to issue a query using keywords or tags specified in the payload against messages in a persistent queue associated with a U-Subject.
RESPONSE	Response from the U-Server containing an array of messageId values that match the query. The messageId of the query is contained in the correlationId of the response message
REQUEST	Requests the U-Server to retrieve the U-Message associated with a specified

Verb	Description
	`messageId`. The `messageId` can be the result of a `QUERY`, or it can be a `messageId` contained in a notification message (e.g., referencing a power system model update).
`REPLY`	Response from the U-Server containing the contents of the message requested in the `REQUEST` message. The `correlationId` of the response is the `messageId` of the request.
`ACKNOWLEDGE`	Requests the U-Server to acknowledge receipt of the message. This functions similar to a “noop” command and can be used by publish applications to verify connectivity to a U-Server and verify permissions to the named subject. (Note that UUDEX has a separate, independent connectivity process that is used to keep connections alive and gather performance statistics.)
`ACKNOWLEDGED`	Response from U-Server to the `ACKNOWLEDGE` request. Only used if the `ackRequired` flag is set to true and the `replyAddress` field is not blank or null. The `ACKNOWLEDGE` response is also provided by U-Subscribe clients when the `asyncReplyFlag` is set to true to provide acknowledgement by the subscribing application that the U-Message has been received at its final destination. If there are multiple U-Subscribers to the message, multiple `ACKNOWLEDGED` responses should be expected. Recipients are identified by their `origin` or `source` fields. Additional information (e.g., the application receiving the information and sending the acknowledgement) can be supplied in the `status` field.
`NACKED`	General negative acknowledgement. This result is returned to indicate that the processing associated with the request has failed. Additional information about why the request failed must be supplied in the return message.

4.0 UUDEX Message Payload

The U-Message payload contains the information that is to be published in the message. Message payloads used for information transfer are formatted so that there is a one-to-one relationship between a message payload and a subscription. That is, a given U-Message contains information to be published for a single subject, although the contents of the information within the subject may be different for each published message. This allows a client to subscribe to a high-level set of information (e.g., line flow values), and receive published information associated with that set of information as it is made available.

The payload of the message is a complicated nested structure containing both metadata blocks as well as a subject block that contains the individual dataElements blocks.

The simple information transfer payload described in section 6.0.

```
{
  "dataSet":{}
}
```

which expands to:

```
{
  "dataSet":{
    "dataElements":[]
  }
}
```

Note that the dataElements structure is an array allowing for multiple individual data elements to be transmitted in the same dataSet payload.

A dataSet may also optionally contain additional JSON structures that may be used to further describe the data set:

```
{
  "dataSet":{
    "property1":"",
    "property2":{
      "property3":"",
      "property4":""
    },
    "dataElements":[]
  }
}
```

As noted for the header, if additional properties are included, their interpretation must be agreed to by both the publisher and subscriber. If the subscriber cannot interpret them, it is free to ignore them.

The dataElements structure contains an array of individual data elements all with the same JSON structure as defined in one of the sections of Section 7.0.

Message payloads for management structures are specific to those structures as discussed in the *Universal Utility Data Exchange (UUDEX) – Security and Administratio*n report.

4.1 UUDEX Message and dataElement Encryption

Note that while U-Messages may be encrypted using TLS encryption cyphers while in transit between UUDEX Clients (U-Clients) and the U-Server, messages stored in the U-Server data store are not encrypted by default while waiting for subscription fulfillment. This means that unless the content of the messages is encrypted prior to being published by the client, the content is accessible by anyone with access to the U-Server data store and will be able to see the content. If message hashes are used, the content is protected against tampering, but not against observation.

If the content of the U-Message is to be protected against observation, it must be encrypted by the U-Publish Client prior to publication, and then decrypted by the U-Subscribe Client after subscription fulfillment. The mechanics of the encryption and associated key management is beyond the scope of the UUDEX specification.

The entire U-Messages may be encrypted by specifying the `encryption` field in the U-Message header. If only a single dataElement is to be encrypted, the `encryption` field in the specific dataElement should be specified.

The specific values for the `encryption` field and any associated processing associated with encryption or key management is beyond the scope of this document.

5.0 Subscriptions and Subjects

In order to simplify client processing, multiple subjects can be grouped together in the form of a U-Subscription. Essentially, a U-Subscription is a collection of one or more U-Subjects a consumer is interested in consuming. This means that a consumer will receive all messages published to any of the U-Subjects in the U-Subscription. This is a convenient way to consume messages since the consumer needs to only make one call to U-Server to receive messages from multiple U-Subjects. Of course, if the consumer only wants to consume a single U-Subject, the option of having a single U-Subject in a U-Subscription is available.

This allows a single client call to receive information that is formatted using different dataSets, since each message within a given Subject maps to the same dataSet description. This would allow, for example, a U-Subscription to be established to receive Physical Security Incident Reports, DOE OE-417 reports, and STIX formatted cybersecurity events. The client could subscribe to the Subscription and receive U-Messages from any of the reporting message structures.

Alternatively, U-Subjects could be set up to receive measurement data from individual TOPs at an RC. Each U-Subject is restricted to allow only a single TOP publish access. The RC would receive all the messages in the U-Subscription which contains data from all TOPs. If an additional TOP were added to the U-instance, a new U-Subject would be created for that TOP, and the U-Subscription modified to add the new U-Subject to the U-Subscription, allowing the RC's subscribe application to receive the additional TOP's data without any programming changes.

Figure 5-1shows the relationships between subjects and subscriptions, and how publishers (producers) and subscribers (consumers) interact with them[1].

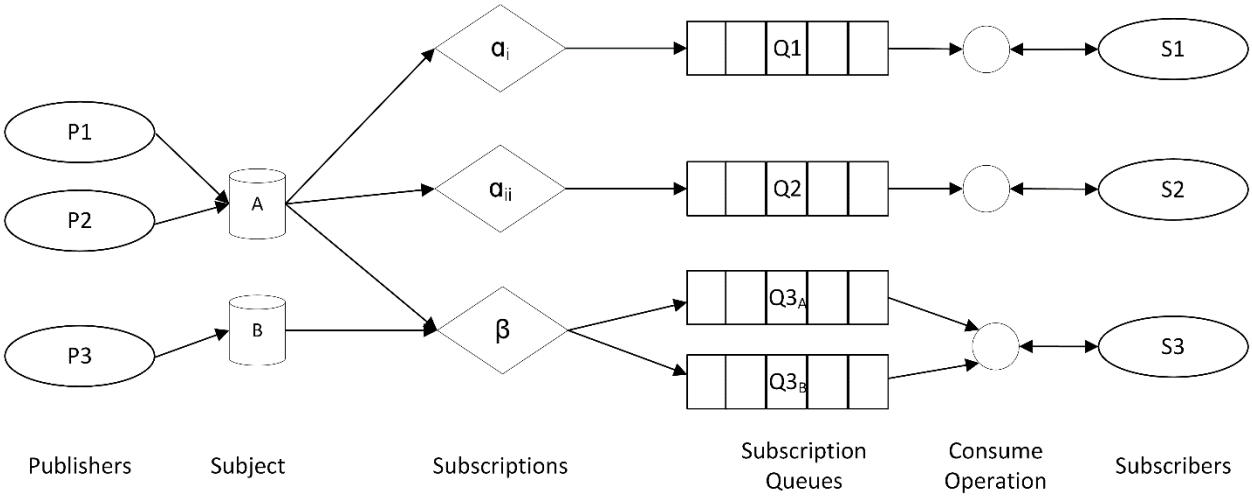

Figure 5-1: Subscription and Queue Processing

[1] Note – this description is based on the RabbitMQ based reference implementation of UUDEX. Other message bus systems may implement queues and subscriptions differently internally, but the interface for the publisher and subscriber should be the same.

In this example, there are three publishers (P1, P2, and P3), two subjects (A and B), two subscriptions (α and β), and three subscribers (S1, S2, and S3).

Publishers P1 and P2 both publish to Subject A, while Publisher P3 publishes to Subject B.

Subscriber S1 creates a subscription and attaches subject A to it as Subscription α_i while Subscriber S2 creates a subscription and attaches it to the same subject, Subject A, as Subscription α_{ii}. Although the subscriptions are identical, internally they are managed as separate subscriptions, each with an independent queue, Queue Q1 for Subscriber S1, and Queue Q2 for Subscriber S2.

Subscriber S3 creates a subscription and attaches both Subject A and Subject B to it as Subscription β. Since there are multiple subjects in the subscription, a separate queue is established for each subject, i.e., Queue $Q3_A$ for Subject A, and Queue $Q3_B$ for subject B.

Whenever a message is published to a subject, it is processed by all the subscriptions attached to that subject and placed on a subscription fulfillment queue that is bound to an individual subscriber. As individual subscribers retrieve data, it is removed from their subscription fulfillment queue, but the subscription fulfillment queues for other subscribers are untouched by that action.

The actions of a publisher and a subscriber are asynchronous, that is, unless prohibited by the fullQueueBehavior's "BLOCK_NEW" processing, publishers can continue to publish new messages to subjects regardless of whether subscribers retrieve them.

For example, assuming that there is no fullQueueBehavior processing:

1. Publisher P1 publishes a message to Subject A
2. The message broker processes the message and notes that it is associated with Subscription α_i, Subscription α_{ii}, and Subscription β.
    a. The message is then placed on Queue Q1, Queue Q2, and Queue $Q3_A$.
3. The same processing would occur when Publisher P2 publishes a message to Subject A.
4. Publisher P3 publishes a message to Subject B.
5. The message broker processes the message and notes that is it associated with Subscription β.
    a. The message is placed on Queue $Q3_B$.

At any time, a subscriber can check to see if there are messages in its subscription queue and retrieve them. When a subscriber calls the consume function to retrieve messages, the subscription queues are checked to determine if there are any messages pending in the queues for that subscription, and if so, all messages are extracted from the queues, formatted as a list of responses, and returned to the subscriber. If there are no messages in the associated queues, and the consume request is asynchronous, a null list of messages is returned; if the consume request is synchronous, the consume function waits until one or more messages arrive in any of the queues and returns them. If there are multiple queues associated with a subscription, all the pending messages are extracted from each queue in turn to create the list of returned messages.

For example:

1. Subscriber S1 represents a real-time responsive application that is constantly waiting for a message to be delivered. It makes a synchronous consume request, and as soon as a message shows up on Queue Q1, Subscriber S1 retrieves the message and processes it. The message broker then removes the message from Queue Q1. Since a single subject is associated with Subscription α_i, all messages retrieved by Subscriber S1 are from subject A
2. Subscriber S2 only checks for messages on its queue once an hour. When it wakes up each hour, it makes an asynchronous consume request to check to see if there are any messages on Queue Q2, and if so, the consume function extracts all the messages from the queue and returns them. If there are no messages on Queue Q2, a null list is returned. The message broker then removes the message from Queue Q1. Since a single subject is associated with Subscription α_{ii}, all messages retrieved by Subscriber S2 are from Subject A. Note that the same message may have been delivered and processed by Subscriber S1 up to an hour before it was retrieved and processed by Subscriber S2.
3. Subscriber S3 represents a manual process that only checks for messages on demand when a human operator makes a request. When the manual application is run, it makes an asynchronous consume request which checks to see if there are any message on Queue $Q3_A$ or $Q3_B$. If there are, the consume function extracts all messages from Queue $Q3_A$, formats them in a list, then extracts all messages from queue $Q3_B$, and adds them to the list of pending messages. Once all the queues are checked and the messages extracted from them, the list of messages is returned to the application. Since multiple subjects are associated with Subscription β, the Subscriber S3 application must be able to process messages from either Subject A or Subject B. As with Subscriber S2, messages from Subject A may have already been processed by Subscriber S1 or Subscriber S2 before Subscriber S3 retrieves and processes it. Note that the since the delivered message list is extracted from each queue in turn, are delivered message order is only maintained with each subject – message delivery order is not maintained across multiple subjects.

Note that if Subscriber S2 also wanted to Retrieve data for Subject B, it would need to either 1) establish a separate subscription causing a separate subscription queue to be established, or 2) attach Subject B to its existing queue and internally process the different message types like Subscriber S3, since multiple subscriptions (i.e., subscribers) cannot share the same subscription queue.

6.0 Information Transfer Structures

The JSON structures UUDEX uses to transfer information can be broken down into two major sets: those that replicate information transfers used in existing ICCP transfers, and additional information transfers that do not map to ICCP.

ICCP information transfers are described using “conformance blocks”, commonly referred to as “Block 1” through “Block 8”. UUDEX currently implements equivalent transfers for Blocks 1 through 4, Blocks 5 through 8 are either not needed or currently not implemented. Table 6.1 shows how the ICCP conformance blocks are implemented in UUDEX.

Table 6.1: ICCP Conformance Block to UUDEX Implementation Mapping

ICCP Conformance Block	UUDEX Implementation
Block 1 – Periodic Power System Data	Implemented as Power System Information exchanges (see section 6.1)
Block 2 – Extended Data Set Condition Monitoring	Combined with Block 1 processing (see section 7.1)
Block 3 – Block Data Transfer	Combined with Block 1 processing (see section 7.1)
Block 4 – Informational Messages	Implemented (see section 7.2)
Block 5 – Device Control	Not implemented
Block 6 – Program Control	Not implemented
Block 7 – Event Reporting	Not implemented
Block 8 – Additional User Objects	Not needed due to the ability to create and define new JSON structures for information transfer

The flexibility that UUDEX offers with self-describing information and JSON structures allows a single message transfer to contain as much or as little information as is required. The use of CIM mRIDs to define each data element allows information to be transferred periodically with all values whether they changed or not, or to transfer only changed values. The CIM mRIDs also allow the information to be transferred in any order. This flexibility allows UUDEX to combine the transfers from ICCP Blocks 1 through 3 into a single transfer mechanism, deferring the selection of information to be transferred to the application creating the message and requesting the transfer.

Informational text messages are also implemented in UUDEX allowing UUDEX users to publish and subscribe to short text messages

The additional information transfers, such as incident reports, power system model exchanges, reliability coordinator information system (RCIS) message transfers, and possibly even exchange of market information are implemented as JSON message exchanges, each with their own formats and specifications. U-instances may define and exchange any other information as

long as it can be contained within a JSON structure and data element. Even unstructured binary data can be exchanged by encoding it and wrapping it in a JSON descriptor – all that is required is agreement on how to encode and decode the information object both the U-Publisher and U-Subscriber clients.

Note: Examples of messages containing these structures are contained in Appendix E.

6.1 Diagnostic and Test Messages

In order to facilitate simple testing, a simple message structure is defined that can be clearly identified as being used for diagnostic and testing capability. This message can be used to test simple connectivity, security ACL processing, and performance testing.

```
{
  "dataSet":{
    "dataElements":[
      {
        "test":{
          "schema":"https://www.uudex.org/uudex/0.1/InformationMessage",
          "schemaVersion":"0.1",
          "sendingReference":"",
          "sequenceNumber":"",
          "encryption":"",
          "encoding":"",
          "compression":"",
          "properties":"",
          "data":{
            "contents":""
          }
        }
      }
    ]
  }
}
```

Diagnostic and Test messages can be created for a variety of test scenarios ranging from simple and small messages containing no payloads that can be used to verify connectivity, to messages with varying or large payloads that can be used to test performance.

The smallest message supported contains only the `schemaVersion` field, while there is no architectural limit to the maximum size of a message that contains payload.

Since the format of the `contents` field is undefined, it can contain arbitrary data, so that in addition to performance tests, it can be used to compare uncompressed and compressed transmission of the same data, and to validate different compression and encoding methods (note that if the `contents` is compressed or contains binary data, it must also be encoded).

Table 6.2 provides an explanation of the fields used in the test message `dataElements` block.

Table 6.2: Diagnostic and Test Messages

Field (abbreviation)	Required / Optional / Recommended	Description
dataSet (ds)	required	Specifies this is a JSON structure containing a power system data set
dataElements (de)	required	The array of information messages being exchanged
test	required	Specifies that the dataElements are diagnostic or test messages
schema	optional	Pointer to the JSON schema being used
schemaVersion (sver)	required	Version of the JSON schema being used. This may be a number or letter (e.g., “1.0” or “A”), and may indicate a site-specific version by as a string of characters (e.g., “RC-V1”).
sendingReference	optional	An arbitrary reference value that can be used to describe the test or tester.
sequenceNumber	optional	A field that may be used to indicate the message sequence number. If multiple test messages are published, and the sequence number is incremented for each message, it can be used to verify that all published messages are received by the subscriber, allowing the subscriber to verify the order of receipt.
encryption	optional	Specifies whether the data portion of the dataElement is encrypted. If not specified, or specified as “NONE”, the data is not encrypted; otherwise, specifies the method used to encrypt the data. When specified, the contents of the data{} structure (the entire contents inside the braces, but not including them) is encrypted using the indicated methodology. Specific values for the encryption field are not specified, nor are the mechanisms for encryption key management, which should occur outside the scope of UUDEX. If additional parameters are required, they may be specified as custom extensions to the standard U-Message header. If encryption is specified as anything other than “NONE”, encoding is required, and compression is recommended.
encoding	optional	The encoding used to transfer the information. Supported values are “ASCII” (ISO/IEC 646) or “UTF-8” (ISO 10646) for direct XML transfers; and “BASE64”, or “BASE85” for encoding binary values into printable ISO/IEC 646 characters for transport. If compression is used, it must be performed before the encoding to ensure that all

Field (abbreviation)	Required / Optional / Recommended	Description
		transmitted characters are ISO/IEC 646. If not specified, "ASCII" is assumed, and the native format of the contents *must* be ISO/IEC 646 characters. See Appendix D for additional information.
compression	optional	Compression method used to shrink the information prior to encoding it. Supported values are "NONE", "ZLIB" and "GZIP" (note ZLIB and GZIP represent the same algorithm – gzip is a stand-alone program that implements the zlib compression library). Additional methods such as "LZ4", "LZ77", "LZ78", "LZO", "LZSS" "LZFSE", "LZVN", "LZW", "DEFLATE", "BZIP2", "LZMA", "LZMA2", "PPM", and "RLE", may be used by mutual agreement of both the publisher and subscriber. If not specified, "NONE" is assumed. See Appendix D for additional information.
properties	optional	Any JSON object (or JSON structure) that is mutually agreed to by the U-Publisher and all U-Subscribers. For example, encryption parameters when specifying encryption.
data	optional	Structure to indicate that contents is specified. Note that if the contents field is used, the data field is required.
contents	optional	An optional field of arbitrary length that can be used to test UUDEX performance or testing of performance constraints. The contents and format of the contents is not defined. If the contents contain binary data, is encrypted, or is compressed, it must be encoded.

6.2 Power System Information

Power system information values are modeled after ICCP Blocks 1, 2, and 3, and where possible use nomenclature and information structures modeled after the Common Information Model - CIM (IEC 61968 and IEC 61970). This information is broken into at least two major categories:

1. information related directly to power system elements such as lines, generators, busses, and breakers
2. information about the power system that is not directly tied to a specific power system element, such as system frequency, total interchange, and total system load.

In the first case, the CIM elements associated with PowerSystemResource, specifically the Measurements, class are specified so that they can be most closely aligned with other uses of the CIM. In the second case, the CIM elements associated with IndicationPoint are used.

Where applicable, ICCP models as defined in 60870-6 specifications will be merged with existing CIM definitions to build a domain model for ICCP data. In CIM, virtually everything inherits from Identified Object, where each object has a unique identifier known as a CIM mRID

(which is a UUID). Identified Objects may have many Names, which are key to the mapping of legible, well-known, legacy, and foreign identifiers for a given object instance.

Object names may be made unique through use of the Name class, where each name may be associated to a specific `NameType`, as is shown in Figure 6-1:

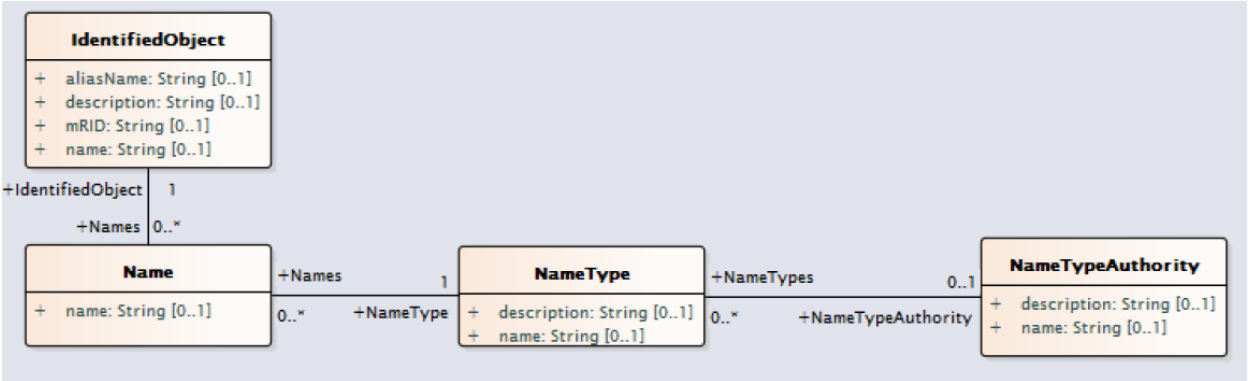

Figure 6-1: CIM IdentifiedObject - Name Relationship

A Measurement represents any measured, calculated or non-measured, non-calculated quantity. Any piece of equipment may contain Measurements, e.g., a substation may have temperature measurements and door open indications, a transformer may have oil temperature and tank pressure measurements, a bay may contain a number of power flow measurements and a breaker may contain a switch status measurement. Some Measurements represent quantities related to a particular sensor location in the network, e.g., a Potential Transformer (PT) at a busbar or a Current Transformer (CT) at the bar between a breaker and an isolator. This is captured by the Measurement - Terminal association that is used to define the sensing location in the network topology. The location is defined by the connection of the Terminal to ConductingEquipment. This is shown in Figure 6-2 (cropped from a larger drawing).

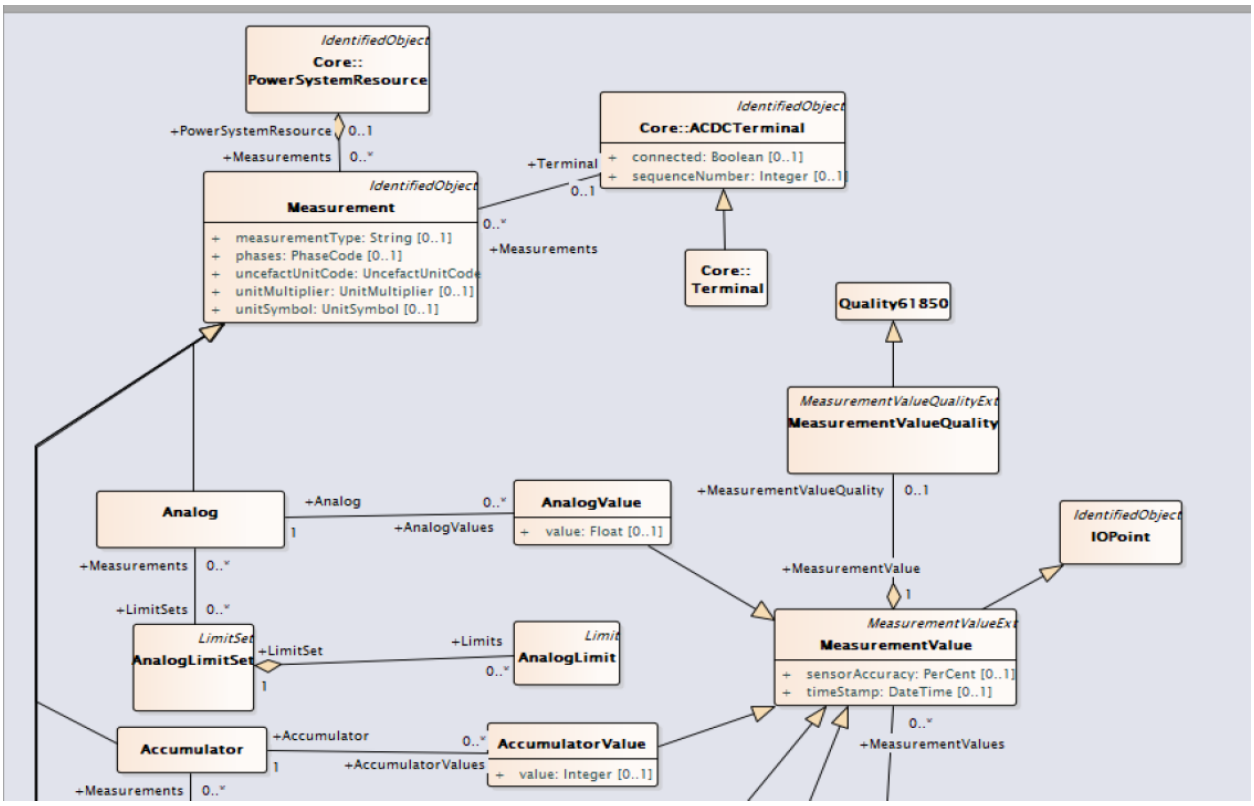

Figure 6-2: CIM Measurement Class relationship

ICCPPoint is partially defined in the CIM model. There are additional attributes defined in ICCP specification that are not defined in the CIM model. To accommodate the additional attributes a new class called “IndicationPoint” will be created that extends the ICCPProvidedPoint class. ICCPProvidedPoint class extends “‘IdentifiedObject” and hence inherits all its attributes. This is shown in Figure 6-3 (cropped from a larger drawing). UUDEX drops the “ICCP” portion of the name to create more generic names.

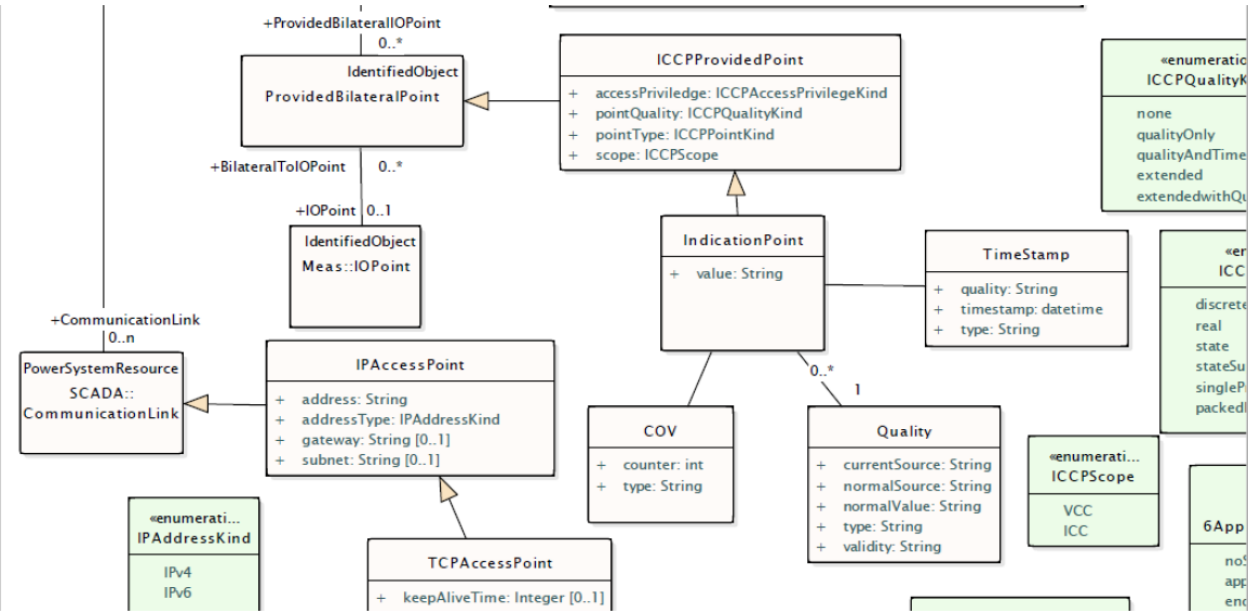

Figure 6-3: CIM ICCPProvidedPoint Class Relationships

All power system information will attempt to follow naming conventions used in the CIM, although the names cannot directly map due to differences between how the CIM is defined (using a variant of XML), and how the UUDEX information structures are defined (using JSON that is expressed using Apache Avro).

Mappings to CIM data objects were selected since they represent a standardized approach to naming and provide for the use of a “master resource identifier”, or mRID that can be used to uniquely identify any CIM object. All CIM mRIDs used by UUDEX must be unique not just within an organization, but across all organizations in a U-Infrastructure (which means that if an organization participates in multiple U-infrastructures, the CIM mRIDs must be unique across all of them). In order to guarantee that, it is recommended that they be expressed in UUID format as defined by IETF RFC 4122, which is technically equivalent to ITU-T Rec. X.667 and ISO/IEC 9834-8. If this is not possible (for example, due to pre-existing agreements by participants), any unique designation may be used.

Some information described for power system data is optional, and therefore not required to be transmitted in each message. Generally, the only required fields are the CIM mRID (used to define the information item being transferred), and the value. Other information fields, such as names and descriptors are optional, and even if transferred, are not required to be sent with each value, rather can be sent infrequently, like every 30 transmissions (in this case providing the descriptor information once a minute for values sent every two seconds.

Other information, while not specifically required, may be transmitted with every message if applicable. Examples of this information include quality flags, alarm indications, and value metadata. For any information that is not published with each message, users should be aware that information may be unsynchronized, especially in cases where the subscribing node is reset, and intermediate messages are lost. For this reason, other than descriptive information that is not required for application processing should be transferred with each message.

The dialect of Avro/JSON used in UUDEX does not support classes, so, while the field names used in UUDEX are modeled after the CIM class nomenclature, there are no inherent classes in UUDEX.

For example, The CIM class element “PowerSystemResource.mRID” representing the CIM mRID object in the class PowerSystemResource, in UUDEX becomes “`PowerSystemResourceMRID`”. This allows a straightforward translation for human readability but will require a translation table for applications to properly reference and store information sent through UUDEX.

Standardized abbreviations are provided for field names and static values. While an attempt is made to make all abbreviations unique, they are only guaranteed unique within their corresponding structures. Standardized abbreviations are only supplied for information transfer data types since they will be the most common and frequently-used information structures exchanged using UUDEX. Other information structures will be transmitted less frequently, so the small amount of bandwidth savings they would represent is offset by the decreased code required to process the abbreviated field names.

6.2.1 Quality, Timestamp, Alarms, Inhibit, ReadingType, and Values Structures

The `quality`, `timeStamp`, `alarms`, `inhibit`, `readingType`, and `values` structures are common to both Power System Resources and Indication Points, and are described here.

6.2.1.1 `Quality`

The `quality` block is used to convey the overall quality of the transmitted value.

Quality and flag codes are used by all power system measured values (analog and status) to indicate the quality, validity, and other information about the measured value. The quality codes represent a combination of values from existing ICCP exchanges, OPC (Open Platform Communications), and industry suggestions.

The codes are divided into two sections – one dealing with characteristics of the value itself, referred to as the “quality”, and the other dealing with how the value was or is processed by the Supervisory Control and Data Acquisition (SCADA) system, referred to as “flags“.

There is the possibility of creating illogical combinations of the validity quality codes. It is the responsibility of both the publisher and subscriber to determine how to process any potential illegal combinations. Ideally, the publisher would not produce illogical combinations (e.g., it should not set both the “`HIGH LIMIT EXCEEDED`” and “`LOW LIMIT EXCEEDED`” codes for the same value), but assumptions made by both the publisher and subscriber may lead to other illegal combinations.

Not all quality codes are appropriate for all `dataElementTypes`. For example, “`HIGH LIMIT EXCEEDED`” doesn’t make sense for the status of a switch or circuit breaker, but it may make sense for a change-of-value count for a breaker.

Not all publishers or subscribers may have the capability to generate all the individual codes – it is the responsibility of the subscriber to be able to map all possible codes (or combinations of codes) into quality codes that make sense to the local applications.

Quality indications, rather than bit masks, are used to allow for variations in how the quality codes are mapped in different SCADA databases.

```
{
  "quality":{
    "currentSource":"",
    "normalSource":"",
    "selection":"",
    "normalValue":"",
    "validity":[
      ""
    ]
  }
}
```

Table 6.3 provides an explanation of the fields used in the `quality` block of the message.

Table 6.3: Quality Block Field Descriptions

Field (abbreviation)	Required / Optional / Recommended	Description
`quality (q)`	required if included	Specifies the start of the `quality` block if included. The `quality` block is optional, but if included this field is required to specify the start of the `quality` block.
`currentSource (cs)`	required	Indicates the current source for the value. (Some entries are from those defined in IEC 61970-301:2020 ed.7, Table 7[1]). Entries include: • "`OPERATIONAL`" – non-specific; the primary value being used by the applications. – May be abbreviated "`OP`". • "`TELEMETERED`" – value received from field equipment via the SCADA system – May be abbreviated "`T`" – Alternative name "`SCADA`", which may be abbreviated "`SC`". This is an IEC 61970-301 code. • "`CCLINK`" – value received from a remote control center. This is an IEC 61970-301 code. – May be abbreviated "`CCL`". • "`MANUAL`" – value entered by an operator (also referred to as manual override, manual entered, substituted, local override, forced).

[1] IEC-International Electrotechnical Commission. "IEC 61970-301." (2020).

Field (abbreviation)	Required / Optional / Recommended	Description
		– May be abbreviated "M". – Alternative name "OPERATOR", which may be abbreviated "OPR". This is an IEC 61970-301 code. – Alternative name "SUBSTITUTED", which may be abbreviated "S". • "ESTIMATED" – value estimated by the State Estimator application – May be abbreviated "E" • "POWERFLOW" – value updated as the result of a power flow application execution. This is an IEC 61970-301 code. – May be abbreviated "PF". • "FORECASTED" – value that is planned or forecasted. This is an IEC 61970-301 code. – May be abbreviated "FC". • "CALCULATED" – value is calculated from other values, for example, Total Net Interchange. – May be abbreviated "C" • "ALLOCATED" – value is calculated by a load allocator. This is an IEC 61970-301 code. – May be abbreviated "ALL".
normalSource (ns)	optional	Indicates the normal source for the value. Valid entries are specified in the currentSource field
selection (sel)	optional	An indication as to which of multiple sources is being used. For example, if multiple frequency sources can be telemetered, this field can specify which one is being used by applications. Entries include: • "PRIMARY" – the designated primary source is selected – May be abbreviated "P" • "BACKUP" – the designated backup source is selected – May be abbreviated "B" • "ALTERNATE" – the designated alternate source is selected – May be abbreviated "A" If not specified, "PRIMARY" is assumed.

Field (abbreviation)	Required / Optional / Recommended	Description
`normalValue (nv)`	optional	Normal measurement value, e.g., used for percentage calculations.
`validity (val)`	required	Specifies the validity of the value. This is an array of quality codes allowing for multiple quality codes to be transferred for the value. Some values are derived from OPC processing[1], others from the list of quality codes summarized in IEC 62361-2[2] . Entries include: • "`VALID`" – the value is valid without any indicated issues (also referred to as Good, within limits, current). This is also an IEC 60870-6 code. – May be abbreviated "`V`" • "`TELEMETRY_FAILED`" – value is no longer being received from primary source, but last good value is retained (also referred to as not updated, static, value is not changing) the value transmitted is the last value received prior to the telemetry failure. The timestamp should specify the time of the last telemetered value. – May be abbreviated "`TF`" • "`TELEMETRY_LOST`" – value is no longer being received from primary source, but no previous value is retained. This is the OPC "telemetry failed – no last good value" quality – May be abbreviated "`TL`" • "`TEST`" – the value has been placed "under test" at the sensor location, and even if telemetry is received, the last good value should be used. This is also an IEC 60870-5 code. – May be abbreviated "`TE`" • "`ABNORMAL`" – the value is in an unspecified non-normal condition or state. – May be abbreviated "`AB`" • "`HIGH_LIMIT_EXCEEDED`" – the value is above the acceptable range for values. – May be abbreviated "`HL`" • "`LOW_LIMIT_EXCEEDED`" – the value is below the acceptable range for values.

[1] See OPC Unified Architecture for Analyser Devices, OPC 10020 among others
[2] IEC-International Electrotechnical Commission. "IEC 62361-2." (2013).

Field (abbreviation)	Required / Optional / Recommended	Description

  - May be abbreviated "LL"

- "NOT_LIMITED" – there are no validity limits associated with this value. No limit checking has been performed. This is an OPC code.
  - May be abbreviated "NL"
- "CONSTANT" – the value cannot change. Note - this different than telemetry failed or static. This is an OPC code.
  - May be abbreviated "C"
- "STATIC" – the value is being telemetered (with periodic updates from field devices) and expected to change but is not changing. A likely cause is an undetected sensor failure.
  - May be abbreviated "ST"
  - Alternative name "NOT TOPICAL", may be abbreviated as "NT". This is the IEC 60870-5 Not Topical/Topical (NT) code.
- "LIMIT_OVERRIDDEN" – the validity limit for the value has been overridden to allow the value to be used.
  - May be abbreviated "LO"
- "RATE-OF-CHANGE LIMIT EXCEEDED" – the value is changing more rapidly than expected.
  - May be abbreviated "ROC"
- "CONFIGURATION_ERROR" – the configuration of the sensor is in error. This is an OPC code.
  - May be abbreviated "CE"
- "INPUT_NOT_CONNECTED_ERROR" – the sensor has detected that that the inputs are not connected. This is an OPC code.
  - May be abbreviated "NC"
- "DEVICE_FAILURE_ERROR" – the RTU has failed. This is an OPC code.
  - May be abbreviated "DF"
- "SENSOR_FAILURE_ERROR" – the sensor has failed. This is an OPC code.
  - May be abbreviated "SF"
- "SUB-NORMAL" – value represents the results of a calculation where less than the minimum number of

Field (abbreviation)	Required / Optional / Recommended	Description
		inputs required for calculated value are available. This is the OPC Uncertain_SubNormal code – May be abbreviated "SN" • "SENSOR_NOT_ACCURATE" – the value is either at one of the sensor limits, or the sensor has determined that the reading is not accurate. This is the OPC Uncertain_SensorNotAccurate code – May be abbreviated "SNA" • "OVERFLOW" – The value of the information object is beyond a predefined range of value (mainly applicable to analog values)[1]. This is the IEC 60870-5 Overflow/No Overflow (OV) code. This is also an IEC 61850-7-3 code. – May be abbreviated "OV" • "BLOCKED" – The value of the information object is blocked for transmission; the value remains in the state that was acquired before it was blocked. Blocking and deblocking may be initiated e.g., by a local lock or a local automatic cause[1]. This is the IEC 60870-5 Blocked/Not Blocked (BL) code. – May be abbreviated "BL" • SUBSTITUTED" – The value of the information object is provided by input of an operator (dispatcher) or by an automatic source[1]. This is the IEC 60870-5 Substituted/Not Substituted (SB) code. – May be abbreviated "SB". • "INVALID" – A value is valid if it was correctly acquired. After the acquisition function recognizes abnormal conditions of the information source (missing or non-operating updating devices) the value is then marked invalid. The value of the information object is not defined under this condition. The mark invalid is used to indicate to the destination that the value may be incorrect and cannot be used[1]. This is the IEC 60870-5 Invalid/Valid (IV) code. – May be abbreviated "IV". • "CARRY" – Counter overflow occurred in the corresponding integration period/no counter overflow occurred in the corresponding integration period.[1] This is the IEC 60870-5 Carry/No Carry (CY) code.

[1] Description from IEC 62361-2:2013

Field (abbreviation)	Required / Optional / Recommended	Description

- May be abbreviated "CY"

- "COUNTER_ADJUSTED" – Counter was adjusted since last reading/Counter was not adjusted since last reading. This is the IEC 60870-5 Counter Was Adjusted/ Counter Was Not Adjusted (CA) code.
  - May be abbreviated "CA"
- "INVALID" – Counter reading is invalid. This is the IEC 60870-5 Invalid/Valid (IV) code. This is also the IEC 60870-6 NOTVALID code.
  - May be abbreviated "IV"
  - Alternative name "ERROR", which may be abbreviated "ER". This is an IEC 60870-5-103 code.
- "OVERFLOW" – the value has overflowed. This is an IEC 60870-5-103 code. This is also an IEC 61850-7-3 code.
  - May be abbreviated "OV".
- "HELD" – the previous data value has been held over. Interpretation is local. This is an IEC 60870-6- code
  - May be abbreviated "H"
- "SUSPECT" – the Data value is questionable. Interpretation is local. This is an IEC 60870-6 code.
  - May be abbreviated "SP"
- "OUT_OF_RANGE" -- indicates the value is beyond a predefined range of values. The server shall decide if validity shall be set to invalid or questionable. The value mat be assumed to be either invalid or questionable. This is an IEC 61850-7-3 code. This is similar to the HIGH_LIMIT_EXCEEDED or LOW_LIMIT_EXCEEDED codes.
  - May be abbreviated "OOR"
- "BAD_REFRENCE" – indicates the value may not be a correct value due to a reference being out of calibration. The value mat be assumed to be either invalid or questionable. This is an IEC 61850-7-3 code.
  - May be abbreviated "BR"
- OSCILLATORY" – indicates the sensor has determined that the binary value is oscillating (e.g., the value changes in a defined time twice in the same direction (from 0 to 1 or from 1 to 0), and the value should be interpreted as questionable. If the value is still oscillating after a defined number of changes, the value shall be left in the state it was in when the oscillatory

Field (abbreviation)	Required / Optional / Recommended	Description
		condition was first indicated, and the value should be interpreted as invalid. This is an IEC 61850-7-3 code. – May be abbreviated “OSC” • “FAILURE” – indicates the sensor has detected an internal or external error. This is similar to the `CONFIGURATION_ERROR`, `INPUT_NOT_CONNECTED_ERROR` `DEVICE_FAILURE_ERROR`, or `SENSOR_FAILURE_ERROR` codes. This is an IEC 61850-7-3 code. – May be abbreviated as “F” • “OLD_DATA” – indicates an update of the value was not made during a specific time interval. The value may be an old value that may have changed in the meantime This may be similar to the `TELEMETRY_LOST` code. This is an IEC 61850-7-3 code. – May be abbreviated as “OD” • “INCONSISTENT” – indicates the sensor has detected an inconsistency. This is an IEC 61850-7-3 code. – May be abbreviated as “INC” • “INACCURATE” – indicates the value does not meet the stated accuracy of the data source. This is an IEC 61850-7-3 code. – May be abbreviated as “INA”

6.2.1.2 `TimeStamp`

The `timeStamp` block is used to convey a timestamp associated with a value

```
{
  "timeStamp":{
    "quality":"VALID",
    "value":"2020-08-24T20:49:18+0000"
  }
}
```

Table 6.4 provides an explanation of the fields used in the `timestamp` block of the message.

Table 6.4: Timestamp Block Field Descriptions

Field (abbreviation)	Required / Optional / Recommended	Description
`timeStamp (ts)`	required if included	Specifies the start of the `timeStamp` block if included. The `timeStamp` block is optional, but if included this field is required to specify the start of the `timeStamp` block.
`quality (q)`	required	Specifies the quality of the time indicated. Possible values include: • "`VALID`" – the time is valid and derived from a synchronized clock – May be abbreviated "`V`" • "`INVALID`" – the time is known not valid for an unspecified reason – May be abbreviated "`I`" • "`UNSYNCH`" – the time is from a clock, but the clock is not synchronized – May be abbreviated "`U`"
`value (v)`	required	The time in ISO 8601 format pertaining to the associated data.

6.2.1.3 Alarms

The `alarms` block is used to convey the alarm status for the point. This is used to indicate whether the point is currently in an alarm state, whether the alarm state has been acknowledged, or whether the point that was previously in alarm state has returned to normal state.

```
{
  "alarms":{
    "state":"",
    "acknowledged":"",
    "returnToNormal":"",
    "alarmCovCounter":0
  }
}
```

Table 6.5 provides an explanation of the fields used in the `alarms` block of the message.

Table 6.5: Alarms Block Field Descriptions

Field (abbreviation)	Required / Optional / Recommended	Description
`alarms (alrm)`	required if included	Specifies the start of the quality block if included. The `alarms` block is optional, but if included this field is required to specify the start of the `alarms` block.
`state (s)`	required	The current alarm state of the point. Entries include: • "`ALARM`" – the point is currently in alarm state – May be abbreviated "`A`" • "`NORMAL`" – the point is currently in normal state – May be abbreviated "`N`"
`acknowledged (ack)`	required	Indication as to whether the alarm state has acknowledged by the operator. This is a Boolean (true/false).
`returnToNormal (rtn)`	required	Indication that the point was in alarm but returned to normal since the last transmission. This could be either that the point was normal, went into alarm, and returned to normal since the last transmission, or the point was in alarm for the last transmission, but has returned to normal for this transmission. This is a Boolean (true/false).
`alarmCovCounter (acovc)`	optional	An integer count of the number of times the alarm status has changed since the last transmission. If the value is an odd number, the alarm value should be different than the last telemetered value. If the value is two or greater, the alarm condition has changed multiple times since the last transmission. If not specified, no counter information should be assumed.

6.2.1.4 `Inhibit`

The `inhibit` block is an array of quoted string values indicating the processing and state inhibits placed on the point.

```
{
  "inhibit":[
    ""
  ]
}
```

Table 6.6 provides an explanation of the fields used in the `inhibit` block of the message.

Table 6.6: Inhibit Block Field Descriptions

Field (abbreviation)	Required / Optional / Recommended	Description
`inhibit (inh)`	required if included	Specifies the start of the `inhibit` block if included. The `inhibit` block is optional, but if included this field is required to specify the start of the `inhibit` block. The inhibit block is an array of values representing processing inhibits placed on the value. A null list indicates that there are no inhibits placed on the point. Entries include: • "`CONTROL`" – indicates that supervisory controls are inhibited for the point. The point should be considered in "lock-out/tag-out" state and no control actions should be issued to the point. – May be abbreviated "`C`" • "`SCAN`" – indicates that scanning for new values is inhibited. – May be abbreviated "`S`" • "`ALARM`" – indicates that even of the point enters a condition that would normally raise an alarm (such as the point's value being too high or too low), an alarm should not be issued for the condition. – May be abbreviated "`A`" • "`OUT_OF_SERVICE`" – indicates that the point has been removed from service, even though it may still be scanned. – May be abbreviated "`OOS`" • "`TEST`" – indicates that the point has been placed in test or maintenance mode, and telemetry should not be processed. – May be abbreviated "`T`" Other values may be specified by mutual agreement between the publisher and subscriber.

6.2.1.5 `ReadingType`

The `readingType` is a "dotted number" string containing an abbreviated list of attributes for the point. This is modeled after IEC 61958-9 Annex C, and describes fields such as units, scaling factors, flow directions, etc.

`"readingType":"15.8.6.1.0.8.0.0.0.6.38",`[1]

[1] Based on IEC 61958-9 Annex C, this example reading type translates to Present Maximum Indicating Forward Demand Real-power Megawatts for all phases with no harmonics indicated.

Table 6.7 provides an explanation of the fields used in the `readingType` block of the message.

Table 6.7: ReadingType Field Descriptions

Field (abbreviation)	Required / Optional / Recommended	Description
`readingType (rt)`	required	Specifies the reading type. The `readingType` is a string of numbers separated by periods that specify information about the value, such as scaling, units, direction, etc. See IEC 61958-9 Annex C for additional information

6.2.1.6 `Values`

The `values` block contains an array of values that all map to the same power system resource or indication point. The array structure allows the transmission of, for example, a telemetered, estimated, and substituted value for each point, each with its own quality codes, flags, and time stamp (if applicable or available).

```
{
  "values":[
    {
      "value":56.5,
      "quality":{
        "currentSource":"TELEMETERED",
        "normalSource":"TELEMETERED",
        "normalValue":"300.0",
        "validity":[
          "VALID"
        ],
        "selection":"PRIMARY"
      },
      "timeStamp":{
        "quality":"VALID",
        "value":"2020-07-06 09:35:46.305-05:00"
      },
      "covCounter":0,
      "alarms":[
        {
          "state":"",
          "acknowledged":"",
          "returnToNormal":"",
          "alarmCovCounter":0
        }
      ],
      "inhibit":[
        ""
      ]
    }
  ]
}
```

Table 6.8 provides an explanation of the fields used in the `quality` block of the message.

Table 6.8: Values Block Field Descriptions

Field (abbreviation)	Required / Optional / Recommended	Description
`values (val)`	required	Specifies the start of the values block. The `values` block is an array of structures all relating to the same point, allowing multiple values (e.g., telemetered and estimated) to be transmitted in the same message, each with their own attribute blocks.
`value (v)`	required	The value transmitted. • For analog points, this is the floating-point representation of the value • For status points, this represents a two-bit integer indicating the value: – "`INDETERMINANT`" or "`TRANSITIONING`" = `0b00` – "`OPEN`" or "`TRIPPED`" = `0b01` – "`CLOSED`" = `0b10` – "`OTHER`" — Invalid value or custom condition = `0b11`
`quality (q)`	optional	(see `quality` description)
`timeStamp (ts)`	optional	(see `timeStamp` description)
`covCounter (covc)`	optional	An integer count of the number of time the value has changed value since last transmission. This makes the most sense for status values but could also be important for analog setpoint values like transformer tap positions.
`alarms (a)`	optional	(see `alarms` description)
`inhibit (i)`	optional	(see `inhibit` description)

6.2.2 Power System Resource Information

The `powerSystemResource` UUDEX object is based on the PowerSystemResource CIM class and is used for all values that are directly associated with power system elements. For example, this includes line real power, reactive power, current, and voltage, generation real and reactive power and frequency, bus voltage and frequency, breaker and switch status, transformer tap position, real and reactive power, and voltages.

```
{
  "dataSet":{
```

```
"dataElements":[
  {
    "powerSystemResource":{
      "schema":"https://www.uudex.org/uudex/0.1/powerSystemResource",
      "schemaVersion":"0.1",
      "encryption":"",
      "encoding":"",
      "compression":"",
      "properties":"",
      "data":{
        "powerSystemResourceMRID":"GEN1",
        "powerSystemResourceName":"ACME1",
        "powerSystemResourceAliasName":"ACME Plant Unit 1",
        "acdcTerminal":{
          "acdcTerminalMRID":"",
          "acdcTerminalName":"",
          "connected":true
        },
        "measurements":[
          {
            "measurementMRID":"GEN1_8323",
            "measurementName":"GEN1-8223_MW",
            "measurementType":"ANALOG",
            "unitSymbol":"MW",
            "unitMultiplier":1,
            "readingType":"15.8.6.1.0.8.0.0.0.6.38",
            "values":[
              {
                "type":"",
                "value":56.5,
                "quality":{
                  "currentSource":"TELEMETERED",
                  "normalSource":"TELEMETERED",
                  "normalValue":"300.0",
                  "validity":[
                    "VALID"
                  ],
                  "selection":"PRIMARY"
                },
                "timeStamp":{
                  "quality":"VALID",
                  "value":"2020-07-06 09:35:46.305-05:00"
                },
                "covCounter":0,
                "alarms":[
                  {
                    "state":"",
                    "acknowledged":"",
                    "returnToNormal":"",
                    "alarmCovCounter":0
                  }
                ],
                "inhibit":[
                  ""
                ]
              }
            ]
```

```
                }
              ]
            }
          }
        }
      ]
    }
}
```

Table 6.9 provides an explanation of the fields used in the `powerSystemResource dataElements` block, while Table 6.10 provides a mapping from the data element names used in UUDEX to the data element names as described in the CIM.

Table 6.9: Power System Resource Field Descriptions

Field (abbreviation)	Required / Optional / Recommended	Description
`dataSet (ds)`	required	Specifies the payload type
`dataElements (de)`	required	The array of data elements contained in the message
`powerSystemResource (psr)`	required	Specifies the payload contains Power System Resource information
`schema (sch)`	optional	Pointer to the JSON schema being used
`schemaVersion (sver)`	required	Version of the JSON schema being used. This may be a number or letter (e.g., “1.0” or “A”), and may indicate a site-specific version by as a string of characters (e.g., “RC-V1”).
`encryption`	optional	Specifies whether the `data` portion of the `dataElement` is encrypted. If not specified, or specified as “`NONE`”, the `data` is not encrypted; otherwise, specifies the method used to encrypt the `data`. When specified, the contents of the `data{}` structure (the entire contents inside the braces, but not including them) is encrypted using the indicated methodology. Specific values for the `encryption` field are not specified, nor are the mechanisms for encryption key management, which should occur outside the scope of UUDEX. If additional parameters are required, they may be specified as custom extensions to the standard U-Message header.

Field (abbreviation)	Required / Optional / Recommended	Description
		If `encryption` is specified as anything other than "`NONE`", `encoding` is required, and `compression` is recommended.
`encoding`	optional	The encoding used to transfer the information. Supported values are "`ASCII`" (ISO/IEC 646) or "`UTF-8`" (ISO 10646) for direct XML transfers; and "`BASE64`", or "`BASE85`" for encoding binary values into printable ISO/IEC 646 characters for transport. If compression is used, it must be performed before the encoding to ensure that all transmitted characters are ISO/IEC 646. If not specified, "`ASCII`" is assumed, and the native format of the contents *must* be ISO/IEC 646 characters. See Appendix D for additional information.
`compression`	optional	Compression method used to shrink the information prior to encoding it. Supported values are "`NONE`", "`ZLIB`" and "`GZIP`" (note ZLIB and GZIP represent the same algorithm – gzip is a stand-alone program that implements the zlib compression library). Additional methods such as "`LZ4`", "`LZ77`", "`LZ78`", "`LZO`", "`LZSS`" "`LZFSE`", "`LZVN`", "`LZW`", "`DEFLATE`", "`BZIP2`", "`LZMA`", "`LZMA2`", "`PPM`", and "`RLE`", may be used by mutual agreement of both the publisher and subscriber. If not specified, "`NONE`" is assumed. See Appendix D for additional information.
`properties`	optional	Any JSON object (or JSON structure) that is mutually agreed to by the U-Publisher and all U-Subscribers. For example, encryption parameters when specifying `encryption`.
`data`	required	Structure to indicate data component of the `dataElement`.
`powerSystemResourceMRID (psrmrid)`	required	The CIM mRID of the power system resource. This is typically a UUID but can be any representation that is unique across all users of the U-instance.

Field (abbreviation)	Required / Optional / Recommended	Description
`powerSystemResourceName (psrn)`	optional	The name of the power system resource
`powerSystemResourceAliasName (psran)`	optional	An alternate name for the power system resource
`acdcTerminal (acdcT)`	optional	For lines, specifies that the power system resource has “terminals”
`acdcTerminalMRID (acdcmrid)`	optional	The CIM mRID for the terminal of the power system resource at which the measurement it taken. This is typically a UUID but can be any representation that is unique across all users of the U-instance.
`acdcTerminalName (acdcTN)`	optional	The name of the terminal for the power system resource at which the measurement is taken
`connected (acdcC)`	optional	Specifies whether the terminal of the power system resource is connected to other elements of the power system. This is a Boolean value.
`measurements (meas)`	required	Specifies the measurements block. This is an array of measurements.
`measurementMRID (mmrid)`	required	The CIM mRID for the measurement. This is typically a UUID but can be any representation that is unique across all users of the U-instance.
`measurementName (mn)`	optional	The name of the measurement
`measurementType (mt)`	required	The type of measurement (e.g., “`ANALOG`”, “`STATUS`”, “`ACCUMULATOR`”, etc.) -- or -- Specifies the type of measurement. For example, this specifies if the measurement represents an indoor temperature, outdoor temperature, bus voltage, line flow, etc. When the `measurementType` is set to "Specialization", the type of Measurement is defined in more detail by the

Field (abbreviation)	Required / Optional / Recommended	Description
		specialized class which inherits from the CIM Measurement class. Values can be either the IEC CIM measurementType strings, the IEC 61850 name, or other value mutually agreed to by all communicating parties. See 61970-301:2020 Clause 4.5.11.3, Table 6 and Clause 6.10.19
`unitSymbol (mus)`	optional	The symbol for the measurement (e.g., “MW”, “MVAR”, “kV”, etc. See IEC 91970-301:2020 Clause 6.2.70
`unitMultiplier (mum)`	optional	The multiplier for the measurement. Possible uses are for scaling or sign flip. Default value is 1. See IEC 91970-301:2020 Clause 6.2.69
`readingType rt)`	optional	(see `readingType` description). If `readingType` is specified (for example, to provide additional detail), `unitSymbol` and `unitMultiplier` are not needed.
`values (val)`	required	(see `values` description)

Table 6.10: PowerSystemResource CIM Mapping

UUDEX Field name	IEC 61968-100 CIM reference
`dataSet`	n/a
`dataElements`	n/a
`powerSystemResource`	powerSystemResource
`schema`	n/a
`schemaVersion`	n/a
`encryption`	n/a
`encoding`	n/a
`compression`	n/a
`data`	n/a

UUDEX Field name	IEC 61968-100 CIM reference
powerSystemResourceMRID	powerSystemResource.mRID
powerSystemResourceName	powerSystemResource.Name
powerSystemResourceAliasName	powerSystemResource.AliasName
acdcTerminal	acdcTerminal
acdcTerminalMRID	acdcTerminal.mRID
acdcTerminalName	acdcTerminal.name
connected	acdcTerminal.connected
measurements	n/a
measurementMRID	powerSystemResource.measurement.mRID
measurementName	powerSystemResource.measurement.name
measurementType	powerSystemResource.measurement.type
unitSymbol	See IEC 91970-301:2020 Clause 6.2.70
unitMultiplier	See IEC 91970-301:2020 Clause 6.2.69
readingType	See IEC 61958-9 Annex C
values	n/a
quality	n/a
timeStamp	n/a
covCounter	powerSystemResource.measurement.covCounter
alarms	n/a
Inhibit	n/a

6.2.3 Indication Point Information

The `indicationPoint` UUDEX object is based on the indicationPoint CIM class and is used for values that do not specifically relate to power system element measurements but are still considered power system information. This information is often calculated from other power system resource element values. Examples of this information includes ACE, total system load, and total net interchange.

```
{
  "dataSet":{
    "dataElements":[
```

```
{
  "indicationPoint":{
    "schema":"https://www.uudex.org/uudex/0.1/IndicationPoint",
    "schemaVersion":"0.1",
    "encryption":"",
    "encoding":"",
    "compression":"",
    "properties":"",
    "data":{
      "identifiedObjectName":"",
      "identifiedObjectMRID":"",
      "identifiedObjectDescription":"",
      "identifiedObjectAliasName":"",
      "providedPointType":"",
      "providedPointScope":"",
      "values":[
        {
          "value":"",
          "quality":{
            "source":"",
            "currentSource":"",
            "normalSource":"",
            "normalValue":"",
            "validity":[
              "VALID"
            ],
            "selection":"PRIMARY"
          },
          "timeStamp":{
            "quality":"",
            "value":""
          },
          "covCounter":"",
          "alarms":[
            {
              "state":"",
              "acknowledged":"",
              "returnToNormal":"",
              "alarmCovCounter":0
            }
          ],
          "inhibit":[
            ""
          ]
        }
      ]
    }
  }
]
}
}
```

Table 6.11 provides an explanation of the fields used in the Indication Point dataElements block, while Table 6.12 provides a mapping from the data element names used in UUDEX to the data element names as described in the CIM.

Table 6.11: IndicationPoint Field Descriptions

Field (abbreviation)	Required / Optional / Recommended	Description
`dataSet (ds)`	required	Specifies the payload type
`dataElements (de)`	required	The array of data elements contained in the message
`indicationPoint (ip)`	required	Specifies the payload contains Indication Point information
`schema (sch)`	optional	Pointer to the JSON schema being used
`schemaVersion (sver)`	optional	Version of the JSON schema being used. This may be a number or letter (e.g., “1.0” or “A”), and may indicate a site-specific version by as a string of characters (e.g., “RC-V1”).
`encryption`	optional	Specifies whether the `data` portion of the `dataElement` is encrypted. If not specified, or specified as “`NONE`”, the `data` is not encrypted; otherwise, specifies the method used to encrypt the `data`. When specified, the contents of the `data{}` structure (the entire contents inside the braces, but not including them) is encrypted using the indicated methodology. Specific values for the `encryption` field are not specified, nor are the mechanisms for encryption key management, which should occur outside the scope of UUDEX. If additional parameters are required, they may be specified as custom extensions to the standard U-Message header. If `encryption` is specified as anything other than “`NONE`”, `encoding` is required, and `compression` is recommended.
`encoding`	optional	The encoding used to transfer the information. Supported values are “`ASCII`” (ISO/IEC 646) or “`UTF-8`” (ISO 10646) for direct XML transfers; and “`BASE64`”, or “`BASE85`” for encoding binary values into printable ISO/IEC 646 characters for transport. If compression is used, it must be performed before the encoding to ensure that all transmitted characters are ISO/IEC 646. If not specified, “`ASCII`” is assumed, and the native format of the contents *must*

Field (abbreviation)	Required / Optional / Recommended	Description
		be ISO/IEC 646 characters. See Appendix D for additional information.
`compression`	optional	Compression method used to shrink the information prior to encoding it. Supported values are "`NONE`", "`ZLIB`" and "`GZIP`" (note ZLIB and GZIP represent the same algorithm – gzip is a stand-alone program that implements the zlib compression library). Additional methods such as "`LZ4`", "`LZ77`", "`LZ78`", "`LZO`", "`LZSS`" "`LZFSE`", "`LZVN`", "`LZW`", "`DEFLATE`", "`BZIP2`", "`LZMA`", "`LZMA2`", "`PPM`", and "`RLE`", may be used by mutual agreement of both the publisher and subscriber. If not specified, "`NONE`" is assumed. See Appendix D for additional information.
`properties`	optional	Any JSON object (or JSON structure) that is mutually agreed to by the U-Publisher and all U-Subscribers. For example, encryption parameters when specifying `encryption`.
`data`	required	Structure to indicate data component of the `dataElement`.
`identifiedObjectName (ion)`	optional	the common name of the CIM object as might appear on a one-line
`identifiedObjectMRID (iomrid)`	required	the unique CIM mRID associated with the CIM object
`identifiedObjectDescription (iod)`	optional	the descriptive detailed name of the CIM object
`identifiedObjectAliasName (ioan)`	optional	an alternate name as stored in the CIM description of the object
`providedPointType (ppt)`	required	the type of point for the CIM object as defined in the ICCPProvidedPoint.PointType attribute of the CIM object – used to determine the specific fields required in the array of value sets
`providedPointScope (pps)`	required	as defined in the ICCPProvidedPoint.scope
`values (val)`	required	(see values description)

Table 6.12: IndicationPoint CIM Mapping

UUDEX Field name	IEC 61968-100 CIM reference
dataSet	n/a
dataElements	n/a
indicationPoint	indicationPoint
schema	n/a
schemaVersion	n/a
encryption	n/a
encoding	n/a
compression	n/a
data	n/a
identifiedObjectName	identifiedObject.Name
identifiedObjectMRID	identifiedObject.mRID
identifiedObjectDescription	indicationPoint.Description
identifiedObjectAliasName	indicationPoint AliasName
providedPointType	ICCPProvidedPoint.PointType
providedPointScope	ICCPProvidedPoint.scope
values	n/a
covCounter	indicationPoint.measurement.covCounter

6.2.4 Example Analog Value

Power system analog values are typically expressed as floating-point numbers that are scaled to engineering units. In the case of discrete positions, they are integer numbers representing the position (e.g., tap position on a transformer).

Attributes are not all required, but at least one value and its associated attributes must be included when a specific dataElements block is published.

The following example shows the exchange of both telemetered and estimated MW and MVAR values (a total of four values) for a transmission line as a single power system resource.

```
{
  "dataSet":{
    "dataElements":[
```

```
{
  "powerSystemResource":{
    "schema":"https://www.uudex.org/uudex/0.1/powerSystemResource",
    "schemaVersion":"0.1",
    "encryption":"",
    "encoding":"",
    "compression":"",
    "properties":"",
    "data":{
      "powerSystemResourceMRID":"4b84a480-af0e-4287-94a9-e06912098256",
      "powerSystemResourceName":"Line A-B",
      "powerSystemResourceAliasName":"Line Alpha to Bravo",
      "acdcTerminal":{
        "acdcTerminalMRID":"f6dfb0b7-2ece-49b8-8bb9-c082bee88cae",
        "acdcTerminalName":"Line A",
        "connected":true
      },
      "measurements":[
        {
          "measurementMRID":"22c0cc70-f32d-4474-bed9-f23f08ba4978",
          "measurementName":"Line_A-B_MW",
          "measurementType":"ANALOG",
          "unitSymbol":"MW",
          "unitMultiplier":1,
          "readingType":"15.8.6.1.0.8.0.0.0.6.38",
          "values":[
            {
              "type":"TELEMETERED",
              "value":56.5,
              "quality":{
                "currentSource":"TELEMETERED",
                "normalSource":"TELEMETERED",
                "normalValue":"300.0",
                "validity":[
                  "VALID"
                ],
                "selection":"PRIMARY"
              },
              "timeStamp":{
                "quality":"VALID",
                "value":"2020-07-06 09:35:46.305-05:00"
              },
              "covCounter":0,
              "alarms":[
                {
                  "state":"NORMAL",
                  "acknowledged":"",
                  "returnToNormal":"",
                  "alarmCovCounter":0
                }
              ],
              "inhibit":[
                ""
              ]
            },
            {
              "type":"ESTIMATED",
```

```
        "value":57.0,
        "quality":{
          "currentSource":"TELEMETERED",
          "normalSource":"TELEMETERED",
          "normalValue":"300.0",
          "validity":[
            "VALID"
          ],
          "selection":"PRIMARY"
        },
        "timeStamp":{
          "quality":"VALID",
          "value":"2020-07-06 09:35:46.305-05:00"
        },
        "covCounter":0,
        "alarms":[
          {
            "state":"NORMAL",
            "acknowledged":"",
            "returnToNormal":"",
            "alarmCovCounter":0
          }
        ],
        "inhibit":[
          ""
        ]
      }
    ]
  },
  {
    "measurementMRID":"cca009e6-0b17-44fd-8f09-a3253aa60ad4",
    "measurementName":"Line_A-B_MVAR",
    "measurementType":"ANALOG",
    "unitSymbol":"MW",
    "unitMultiplier":1,
    "readingType":"15.8.6.1.0.8.0.0.0.6.38",
    "values":[
      {
        "type":"TELEMETERED",
        "value":10.5,
        "quality":{
          "currentSource":"TELEMETERED",
          "normalSource":"TELEMETERED",
          "normalValue":"300.0",
          "validity":[
            "VALID"
          ],
          "selection":"PRIMARY"
        },
        "timeStamp":{
          "quality":"VALID",
          "value":"2020-07-06 09:35:46.305-05:00"
        },
        "covCounter":0,
        "alarms":[
          {
            "state":"NORMAL",
```

```
                    "acknowledged":"",
                    "returnToNormal":"",
                    "alarmCovCounter":0
                  }
                ],
                "inhibit":[
                  ""
                ]
              },
              {
                "type":"ESTIMATED",
                "value":11,
                "quality":{
                  "currentSource":"TELEMETERED",
                  "normalSource":"TELEMETERED",
                  "normalValue":"300.0",
                  "validity":[
                    "VALID"
                  ],
                  "selection":"PRIMARY"
                },
                "timeStamp":{
                  "quality":"VALID",
                  "value":"2020-07-06 09:35:46.305-05:00"
                },
                "covCounter":0,
                "alarms":[
                  {
                    "state":"NORMAL",
                    "acknowledged":"",
                    "returnToNormal":"",
                    "alarmCovCounter":0
                  }
                ],
                "inhibit":[
                  ""
                ]
              }
            ]
          }
        ]
      }
    }
  }
}
```


6.2.5 Example Status Values

Power system status values are binary values (or in some cases three-state values) representing the status of a device – typically either “open” or “closed”, or in the case of a three-state device like a motor operated switch “indeterminate”.

The following example is for the status of a circuit breaker.

```
{
  "dataSet":{
    "dataElements":[
      {
        "powerSystemResource":{
          "schema":"https://www.uudex.org/uudex/0.1/powerSystemResource",
          "schemaVersion":"0.1",
          "encryption":"",
          "encoding":"",
          "compression":"",
          "properties":"",
          "data":{
            "powerSystemResourceMRID":"4b84a480-af0e-4287-94a9-e06912098256",
            "powerSystemResourceName":"CB-Alpha-103-1",
            "powerSystemResourceAliasName":"Circuit Breaker Station Alpha Bus
103 Bay 1",
            "acdcTerminal":{
              "acdcTerminalMRID":"c3b6a4f6-7e58-4585-bd86-2aac54eedd1f",
              "acdcTerminalName":"CB-Alpha-103-1-1",
              "connected":true
            },
            "measurements":[
              {
                "measurementMRID":"e0edee3c-8511-4a36-96e2-20372b734e96",
                "measurementName":"CB-Alpha-103-1",
                "measurementType":"STATUS",
                "readingType":"0.0.0.0.0.43.0.0.0.0.109",
                "values":[
                  {
                    "type":"TELEMETERED",
                    "value":"CLOSED",
                    "quality":{
                      "currentSource":"TELEMETERED",
                      "normalSource":"TELEMETERED",
                      "normalValue":"CLOSED",
                      "validity":[
                        "VALID"
                      ],
                      "selection":"PRIMARY"
                    },
                    "timeStamp":{
                      "quality":"VALID",
                      "value":"2020-07-06 09:35:46.305-05:00"
                    },
                    "covCounter":2,
                    "alarms":[
                      {
                        "state":"NORMAL",
                        "acknowledged":"",
                        "returnToNormal":"",
                        "alarmCovCounter":0
                      }
                    ],
                    "inhibit":[
                      "CONTROL"
                    ]
                  }
```

```
                    ]
                }
            ]
        }
    }
}
    ]
  }
}
```

6.3 Informational Messages

Informational messages are modeled after ICCP block 4, which is used to send text or other information to an application at a remote control center.

The ICCP protocol includes an Information Message object for sending text or other information to an application at a remote control center. An Information Message object consists of a header portion (identifying the source and purpose of the message) and an Info Stream portion, which contains the body of the message.

Following are some of the key attributes for this payload as defined in ICCP specification (IEC 60870-6-802, Clause 5.4):

- Key Attribute: InfoReference
- Attribute: LocalReference
- Attribute: MessageId
- Attribute: InfoStream

Information message is already defined in CIM specification. The class will be extended to include additional fields that are missing in the spec. See Figure 6-4 (cropped from a larger drawing) for the class definition.

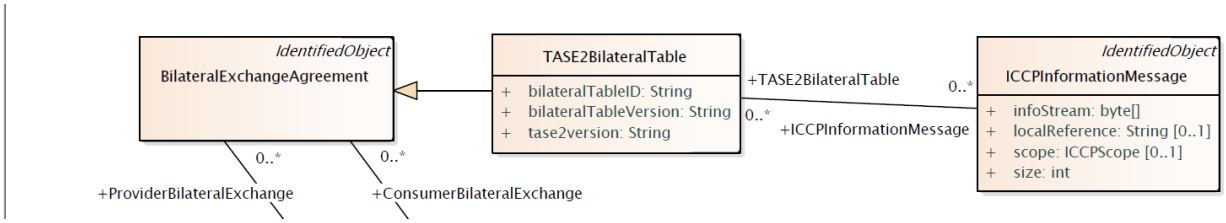

Figure 6-4: ICCPInformation Message Relationship

ICCPInformationMessage is an existing class in IEC 61970. It has been extended to include the additional fields “size” and “infoStream.” This class extends the IdentifiedObject and hence inherits the following fields:

- identifiedObject.Name,
- identifiedObject.mRID,
- identifiedObject.AliasName,
- identifiedObject.Description

The CIM IdentifiedObject class has associations to “Name” class to enable a certain object to have multiple names. But since this is an `informationMessage` there is no need to have multiple names and will be omitted in the payload definition.

```
{
  "dataSet":{
    "dataElements":[
      {
        "informationMessage":{
          "schema":"https://www.uudex.org/uudex/0.1/InformationMessage",
          "schemaVersion":"0.1",
          "encryption":"",
          "encoding":"",
          "compression":"",
          "properties":"",
          "data":{
            "identifiedObjectName":"",
            "identifiedObjectMRID":"",
            "identifiedObjectDescription":"",
            "identifiedObjectAliasName":"",
            "infoReference":"",
            "localReference":"",
            "infoStream":"Hi UUDEX Team"
          }
        }
      }
    ]
  }
}
```

Table 6.13 provides an explanation of the fields used in the Information message `dataElements` block, while Table 6.14 provides a mapping from the data element names used in UUDEX to the data element names as described in the CIM.

Table 6.13: Informational Message Field Descriptions

Field (abbreviation)	Required / Optional / Recommended	Description
`dataSet (ds)`	required	Specifies this is a JSON structure containing a power system data set
`dataElements (de)`	required	The array of information messages being exchanged
`informationMessage (im)`	required	Specifies that the dataElements are information messages
`schema`	optional	Pointer to the JSON schema being used
`schemaVersion (sver)`	required	Version of the JSON schema being used. This may be a number or letter (e.g., “1.0”

Field (abbreviation)	Required / Optional / Recommended	Description
		or “A”), and may indicate a site-specific version by as a string of characters (e.g., “RC-V1”).
`encryption`	optional	Specifies whether the `data` portion of the `dataElement` is encrypted. If not specified, or specified as “`NONE`”, the `data` is not encrypted; otherwise, specifies the method used to encrypt the `data`. When specified, the contents of the `data{}` structure (the entire contents inside the braces, but not including them) is encrypted using the indicated methodology. Specific values for the `encryption` field are not specified, nor are the mechanisms for encryption key management, which should occur outside the scope of UUDEX. If additional parameters are required, they may be specified as custom extensions to the standard U-Message header. If `encryption` is specified as anything other than “`NONE`”, `encoding` is required, and `compression` is recommended.
`encoding`	optional	The encoding used to transfer the information. Supported values are “`ASCII`” (ISO/IEC 646) or “`UTF-8`” (ISO 10646) for direct XML transfers; and “`BASE64`”, or “`BASE85`” for encoding binary values into printable ISO/IEC 646 characters for transport. If compression is used, it must be performed before the encoding to ensure that all transmitted characters are ISO/IEC 646. If not specified, “`ASCII`” is assumed, and the native format of the contents *must* be ISO/IEC 646 characters. See Appendix D for additional information.
`compression`	optional	Compression method used to shrink the information prior to encoding it. Supported values are “`NONE`”, “`ZLIB`” and “`GZIP`” (note ZLIB and GZIP represent the same algorithm – gzip is a stand-alone program that implements the zlib compression library). Additional methods such as “`LZ4`”, “`LZ77`”, “`LZ78`”, “`LZO`”, “`LZSS`” “`LZFSE`”, “`LZVN`”, “`LZW`”, “`DEFLATE`”, “`BZIP2`”, “`LZMA`”, “`LZMA2`”, “`PPM`”, and “`RLE`”, may

Field (abbreviation)	Required / Optional / Recommended	Description
		be used by mutual agreement of both the publisher and subscriber. If not specified, "NONE" is assumed. See Appendix D for additional information.
properties	optional	Any JSON object (or JSON structure) that is mutually agreed to by the U-Publisher and all U-Subscribers. For example, encryption parameters when specifying encryption.
data	required	Structure to indicate data component of the informationMessage.
identifiedObjectName (ion)	optional	The name is any free human readable and possibly non unique text naming the object. Described in IEC 61970-301 clause 6.3.20.
identifiedObjectMRID (iomrid)	optional	Master resource identifier issued by a model authority. The CIM mRID is unique within an exchange context. Global uniqueness is easily achieved by using a UUID, as specified in RFC 4122, for the CIM mRID. The use of UUID is strongly recommended. For CIMXML data files in RDF syntax conforming to IEC 61970-552, the CIM mRID is mapped to rdf:ID or rdf:about attributes that identify CIM object elements. Described in IEC 61970-301 clause 6.3.20.
identifiedObjectDescription (iod)	optional	The identifiedObjectDescription is a free human readable text describing or naming the object. It may be non-unique and may not correlate to a naming hierarchy. Described in IEC 61970-301 clause 6.3.20.
identifiedObjectAliasName (ioan)	optional	The identifiedObjectAliasName is free text human readable name of the object alternative to the CIM IdentifiedObject.name. It may be

Field (abbreviation)	Required / Optional / Recommended	Description
		non-unique and may not correlate to a naming hierarchy. The attribute `aliasName` is retained because of backwards compatibility between CIM releases. It is however recommended to replace `aliasName` with the Name class as `aliasName` is planned for retirement at a future time Described in IEC 61970-301 clause 6.3.20.
`infoReference (ioref)`	optional	The `infoReference` uniquely identifies the `informationMessage`.
`localReference (iolr)`	optional	The `localReference` attribute specifies a value agreed upon between sender and receiver of the `informationMessage`. It further identifies the `informationMessage`. Described in IEC 61970-301 clause 6.3.20.
`infoClass (iocls)`	optional	High-level classification designation for the `informationMessage`, such as “ALARM”, ”OPERATOR MESSAGE” to assist with categorizing, routing, or sorting received messages.
`infoStream (iostr)`	optional	The contents of the `informationMessage`, typically a text message containing printable ISO/IEC 646 (ASCII) or ISO 10646 (UTF-8) characters. (note that JSON sends the message as a quoted string, so the size is not needed.

Table 6.14: IndicationPoint to CIM Mapping

UUDEX Field name	IEC 61968-100 CIM reference
`dataSet`	n/a
`identifiedObjectName`	identifiedObject.name
`identifiedObjectMRID`	identifiedObject.mRID

UUDEX Field name	IEC 61968-100 CIM reference
`identifiedObjectDescription`	identifiedObject.Description
`identifiedObjectAliasName`	identifiedObject.AliasName
`infoReference`	ICCPInformationMessage.infoReference
`localReference`	ICCPInformationMessage.localReference
`infoClass`	n/a
`infoStream`	ICCPInformationMessage.infoStream
n/a	ICCPInformationMessage.scope
n/a	ICCPInformationMessage.size

The following example sends two text messages "Hi UUDEX Team" and "How Are You?" from the U-instance named “RC” to the U-instance named “TransOp2” using the U-Subject "`RC/infoMessage/0001`".

```
{
  "header":{
    "messageID":"d5d1c892-974a-11e9-b198-b0c090affff",
    "noun":"informationMessage",
    "origin":"TransOp1",
    "source":"RC",
    "destination":"TransOp2",
    "timeStamp":"2020-05-13 10:12:09.209124",
    "verb":"CREATED",
    "subject":"RC/infoMessage/0001",
    "encryption":"",
    "encoding":"",
    "compression":"",
    "sensitivity":"",
    "properties":"",
    "hashType":"SHA-256",
    "hash":"hash-value-of-payload"
  },
  "dataSet":{
    "dataElements":[
      {
        "informationMessage":{
          "schema":"https://www.uudex.org/uudex/0.1/InformationMessage",
          "schemaVersion":"0.1",
          "encryption":"",
          "encoding":"",
          "compression":"",
          "properties":"",
          "data":{
            "identifiedObjectName":"",
            "identifiedObjectMRID":"",
```

```
            "identifiedObjectDescription":"",
            "identifiedObjectAliasName":"",
            "infoReference":"",
            "localReference":"",
            "infoStream":"Hi UUDEX Team"
          }
        }
      },
      {
        "InformationMessage":{
          "schema":"https://www.uudex.org/uudex/0.1/InformationMessage",
          "schemaVersion":"0.1",
          "encryption":"",
          "encoding":"",
          "compression":"",
          "properties":"",
          "data":{
            "identifiedObjectName":"",
            "identifiedObjectMRID":"",
            "identifiedObjectDescription":"",
            "identifiedObjectAliasName":"",
            "infoReference":"",
            "localReference":"",
            "infoStream":"How Are You?"
          }
        }
      }
    ]
  }
}
```

6.4 Incident Reports

This section describes the various incident reporting messages that UUDEX supports, including DOE OE-417 reports, a Physical Security Incident Event Report (PSIR), and Structured Threat Information Expression (STIX™) structures used by UUDEX.

6.4.1 Electrical Disturbance Reporting

Electrical disturbance reporting is modeled after the DOE OE-417 report form and is transferred as either the XML representation of the information as entered into the PDF form supplied by DOE, or a native PDF file. Note that the format of the XML structures is not in scope of this document. All fields are required.

Note that since the OE-417 report is file based, a file name is required to be transmitted with the report. Subscribing applications must manage file name collisions when creating files to contain the OE-417 information.

```
{
  "header":{
    "messageID":"",
    "noun":"OE-417",
    "verb":"CREATE",
    "subject":"",
    "origin":"",
```

```
    "source":"",
    "destination":"",
    "timeStamp":"",
    "correlationID":"",
    "context":"",
    "user":"",
    "comment":"",
    "properties":"",
    "schema":"",
    "schemaVersion":"",
    "encryption":"",
    "encoding":"",
    "compression":"",
    "properties":"",
    "sensitivity":"",
    "replyAddress":"",
    "asyncReplyFlag":"",
    "ackRequired":"",
    "expiration":"",
    "tags":[
      ""
    ],
    "hashType":"",
    "hash":""
  },
  "dataSet":{
    "dataElements":[
      {
        "OE-417":{
          "schema":"https://www.uudex.org/uudex/0.1/OE-417",
          "schemaVersion":"0.1",
          "encryption":"",
          "encoding":"",
          "compression":"",
          "properties":"",
          "data":{
            "format":"",
            "name":"",
            "tags":[
              ""
            ],
          }
        }
      }
    ]
  }
}
```

Table 6.15 provides an explanation of the fields used in the OE-417 dataElements block

Table 6.15: OE-417 Report Field Descriptions

Field	Required / Optional / Recommended	Description
dataSet	required	Specifies the payload type
dataElements	required	The array of OE-417 reports contained in the message
OE-417	required	Specifies that the message contains a DOE OE-417 report form
schema	optional	Pointer to the JSON schema being used
schemaVersion	required	Version of the JSON schema being used. This may be a number or letter (e.g., “1.0” or “A”), and may indicate a site-specific version by as a string of characters (e.g., “RC-V1”).
encryption	optional	Specifies whether the data portion of the dataElement is encrypted. If not specified, or specified as “NONE”, the data is not encrypted; otherwise, specifies the method used to encrypt the data. When specified, the contents of the data{} structure (the entire contents inside the braces, but not including them) is encrypted using the indicated methodology. Specific values for the encryption field are not specified, nor are the mechanisms for encryption key management, which should occur outside the scope of UUDEX. If additional parameters are required, they may be specified as custom extensions to the standard U-Message header. If encryption is specified as anything other than “NONE”, encoding is required, and compression is recommended.
properties	optional	Any JSON object (or JSON structure) that is mutually agreed to by the U-Publisher and all U-Subscribers. For example, encryption parameters when specifying encryption.
data	required	Structure to indicate data component of the informationMessage.
format	required	Specifies the format of the information. Supported values are “XML” and “PDF” • “XML” is used to transfer the information used to populate the OE-417 report in a format suitable for parsing and storing into a database. The Adobe Acrobat program can generate the XML file from a completed form and can import the XML into a blank form for viewing and further editing. • “PDF” is used to transfer the report as a PDF file. PDF-formatted files are suitable if the document is scanned.
compression	optional	Compression method used to shrink the information prior to encoding it. Supported values are “NONE”, “ZLIB” and “GZIP” (note ZLIB and GZIP

Field	Required / Optional / Recommended	Description
		represent the same algorithm – gzip is a stand-alone program that implements the zlib compression library). Additional methods such as "LZ4", "LZ77", "LZ78", "LZO", "LZSS" "LZFSE", "LZVN", "LZW","DEFLATE", "BZIP2", "LZMA", "LZMA2", "PPM", and "RLE", may be used by mutual agreement of both the publisher and subscriber. If not specified, "NONE" is assumed. See Appendix D for additional information.
name	required	The name of the file containing the XML or PDF formatted report. Name extensions should match the format but are not required to do so.
tags	optional	A list of tags (keywords) as quoted strings to facilitate discovery of this report in searches.
encoding	optional	The encoding used to transfer the information. Supported values are "ASCII" (ISO/IEC 646) or "UTF-8" (ISO 10646) for direct XML transfers; and "BASE64", or "BASE85" for encoding binary values into printable ISO/IEC 646 characters for transport. If compression is used, it must be performed before the encoding to ensure that all transmitted characters are ISO/IEC 646. If not specified, "ASCII" is assumed, and the native format of the contents *must* be ISO/IEC 646 characters. See Appendix D for additional information.
contents	required	A string variable containing the contents of the OE-417 report. For XML formatted reports, the information may be transferred uncompressed or unencoded, but for PDF formats, encoding is required, and compression is strongly recommended.

6.4.2 Physical Security Incident Reporting

The UUDEX Physical Security Incident Report (PSIR) is designed to demonstrate a format for reporting a physical security incident at an energy sector facility. It is intended to be generated by the victim as a means to share relevant information with peer groups to support awareness of the event and its impacts. It is also intended to facilitate the exchange of indicators so that peers will be better able to detect and prevent similar incidents at their own facilities.

The design of the PSIR incorporates key elements of the DOE OE-417 report form. Because of this, if a party fully fills certain portions of the PSIR, it should be possible to auto-generate most of an OE-417 form. That noted, the PSIR is not intended as a regulatory reporting format. Instead, the focus is to facilitate shared threat intelligence for physical threats, similarly to how the STIX format (see Section 6.4.3) facilitates shared threat intelligence for cyber threats.

Both STIX and the PSIR include sections that are intended to describe indicators that might serve as warnings, alerting peers to signs of similar attacks, which would allow them to take actions to prevent or mitigate them. Both report formats also include fields to characterize details of an attack. However, while STIX includes fields and details that are meant to facilitate forensic analysis efforts, PSIR does not go into a similar level of detail and only provides descriptive information of events rather than analytic details. It is assumed that, in the case of

physical attacks, law enforcement will handle forensic analysis and that energy sector organizations would not have access to, nor necessarily have deep interest in, details of forensic investigations.

Note that only the header contains required fields. All other blocks and fields in those blocks are optional.

Multiple PSIR messages may be related to the same event. In this case, subsequent messages (or supplemental messages filed by different UUDEX Participants) can be linked to the initial report using the "`correlationID`" field in the message header.

6.4.2.1 Report Structure

The report is comprised of several sections, each described below. Except in a very limited number of cases where noted, all fields are optional.

The overall structure of the report is as follows:

```
{
  "header":{
    "messageID":"",
    "noun":"physicalSecurityIncident",
    "verb":"CREATE",
    "subject":"",
    "origin":"",
    "source":"",
    "destination":"",
    "timeStamp":"",
    "correlationID":"",
    "context":"",
    "user":"",
    "comment":"",
    "properties":"",
    "schemaVersion":"",
    "schema":"",
    "encryption":"",
    "encoding":"",
    "compression":"",
    "properties":"",
    "sensitivity":"",
    "replyAddress":"",
    "asyncReplyFlag":"",
    "ackRequired":"",
    "expiration":"",
    "hashType":"",
    "hash":""
  },
  "dataSet":{
    "dataElements":[
      {
        "physicalSecurityIncident":{
        "schema":"https://www.uudex.org/uudex/0.1/PhysicalSecurityIncidentRep
ort",
        "schemaVersion":"0.1",
          "encryption":"",
```

```
"encoding":"",
"compression":"",
"properties":"",
"data":{
  "reportHeader":{
    "uniqueID":"",
    "title":"",
    "summary":"",
    "tags":[
      ""
    ],
    "relatedReports":[
      {
        "reportType":"",
        "reportID":"",
        "reportLocation":""
      }
    ],
    "reportStatus":"",
    "TLP":"",
    "filedBy":{
      "name":"",
      "organization":"",
      "phone":"",
      "email":"",
      "address":{
        "country":"",
        "state":"",
        "code":"",
        "address":""
      }
    },
    "timeOfReport":""
  },
  "incidentSeverity":"",
  "incidentCategory":"",
  "timeOfDetection":"",
  "earliestLikelyOnset":"",
  "targetList":[
    {
      "targetIdentification":"",
      "targetDescription":"",
      "targetOwnerName":"",
      "targetOperatorName":"",
      "targetLocation":{
        "country":"",
        "state":"",
        "code":"",
        "address":""
      },
      "targetImpactState":"",
      "targetRestorationTime":"",
      "targetAdditionalInfo":""
    }
  ],
  "incidentImpact":{
    "functionalImpact":{
```

```
      "severity":"",
      "onsetTime":"",
      "endTime":"",
      "isOngoing":false
    },
    "economicImpact":{
      "severity":"",
      "onsetTime":"",
      "endTime":"",
      "isOngoing":false
    },
    "riskImpact":{
      "severity":"",
      "onsetTime":"",
      "endTime":"",
      "isOngoing":false
    },
    "energySectorImpact":{
      "controlCenterLossOrFailure":false,
      "controlCenterEvacuation":false,
      "lossOfCCMonitoringOrComm":false,
      "damageOrDisruptFacility":false,
      "islanding":false,
      "failureOfTransmissionOrDistribution":false,
      "transmissionInterruption":false,
      "distributionInterruption":false,
      "uncontrolledFirmLoadLoss":0,
      "firmLoadShedding":0,
      "serviceLossToCustomers":0,
      "systemVoltageReduction":false,
      "facilityVoltageDeviation":false,
      "inadequateResourcesForLoad":false,
      "generatingCapacityLoss":0,
      "offSitePowerLossToNuclear":false,
      "potentialImpactToSystemAdequacy":false,
      "vandalismTargetingSecurity":false,
      "fuelSupplyEmergencies":false,
      "physicalThreatPotentiallyDegrades":false,
      "suspiciousDeviceAtFacility":false,
      "otherEnergySectorImpact":""
    },
    "impactedThirdParties":[
      {
        "impactedOrgName":"",
        "impactedOrgRole":"",
        "impactedOrgMagnitude":"",
        "impactedOrgRiskMagnitude":"",
        "impactedOrgAdditionalInfo":""
      }
    ],
    "additionalImpactInformation":""
  },
  "incidentResponse":{
    "responseSummary":"",
    "responseTimeline":{
      "predictableTimeline":[
        {
```

```
          "milestone":"",
          "milestoneTime":""
        }
      ],
      "contingentTimeline":[
        ""
      ],
      "unknownTimeline":"",
      "unrecoverable":""
    },
    "responderNotification":[
      {
        "contact":{
          "name":"",
          "organization":"",
          "phone":"",
          "email":"",
          "address":{
            "country":"",
            "state":"",
            "code":"",
            "address":""
          }
        },
        "timeContacted":"",
        "additionalContactInformation":""
      }
    ],
    "additionalResponseInformation":""
  },
  "attackCharacterization":{
    "attackIntent":{
      "attackIntentService":false,
      "attackIntentEnvironment":false,
      "attackIntentStaffInjury":false,
      "attackIntentPopulationInjury":false,
      "attackIntentFinancialGain":false,
      "attackIntentFinancialLoss":false,
      "attackIntentPolitical":false,
      "attackIntentSpying":false,
      "attackIntentTactical":false,
      "attackIntentUnknown":false,
      "attackIntentOther":""
    },
    "attackActors":[
      {
        "attackerName":"",
        "attackerOrganization":"",
        "attackerTargetRelationship":"",
        "attackerDisposition":"",
        "attackerAdditionalInfo":""
      }
    ],
    "attackMeans":[
      {
        "attackMeansType":"",
        "attackMeansDescription":""
```

```
            }
          ],
          "attackAdditionalInformation":""
        },
        "indicators":{
          "personIndicators":[
            {
              "apparentAge":1,
              "apparentHeight":{
                "measure":0,
                "units":""
              },
              "apparentWeight":{
                "measure":0,
                "units":""
              },
              "apparentGender":"",
              "apparentEthnicity":"",
              "vocalBehavior":"",
              "physicalBehavior":"",
              "otherPersonCharacteristics":""
            }
          ],
          "vehicleIndicators":[
            {
              "vehicleType":"",
              "vehicleMakeModel":"",
              "vehicleColor":"",
              "vehicleIdentifiers":[
                {
                  "identifier":"",
                  "identifierType":""
                }
              ],
              "otherVehicleCharacteristics":""
            }
          ],
          "objectIndicators":[
            {
              "objectDescription":"",
              "objectSize":"",
              "objectLocationFound":"",
              "objectHandlingWarnings":"",
              "otherObjectCharacteristics":""
            }
          ],
          "indicatorDecoys":"",
          "additionalIndicatorInformation":""
        },
        "recommendedCourseOfAction":{
          "recommendation":"",
          "followUp":""
        }
      }
    }
  }
]
```

```
  }
}
```

Specific fields are described in the following sections. Unless specifically noted as required, all fields are optional. Most fields are free-form ISO/IEC 646 (ASCII) or ISO 10646 (UTC-8) text, but some are binary (true/false) or numeric.

Report Header

The `reportHeader` block is used to capture information about the report itself. It has the following fields:

```
{
  "dataSet":{
    "dataElements":[
      {
        "physicalSecurityIncident":{
        "schema":"https://www.uudex.org/uudex/0.1/PhysicalSecurityIncidentRep
ort",
        "schemaVersion":"0.1",
          "reportHeader":{
            "uniqueID":"",
            "title":"",
            "summary":"",
            "tags":[
              ""
            ],
            "relatedReports":[
              {
                "reportType":"",
                "reportID":"",
                "reportLocation":""
              }
            ],
            "reportStatus":"",
            "TLP":"",
            "filedBy":{
              "name":"",
              "organization":"",
              "phone":"",
              "email":"",
              "address":{
                "country":"",
                "state":"",
                "code":"",
                "address":""
              }
            },
            "timeOfReport":""
          }
        }
      }
    }
  }
}
```

Table 6.16 provides an explanation of the fields used in the `reportHeader` block for the Physical Security Incident Reporting structure.

Table 6.16: Physical Security Incident Reporting - Report Header Field Descriptions

Field	Required / Optional / Recommended	Description
`dataSet`	required	Specifies the payload type
`dataElements`	required	The array of data elements contained in the message
`uniqueID`	required	This field provides a universally unique identifier for the document. A UUID is recommended for this purpose. Ideally, the UUID will include the namespace of the author to prevent identifier collision, but if the author desires anonymity, another source (such as an anonymity server that will generate UUIDs in its own namespace for others to use) can be employed.
`schema`	optional	Pointer to the JSON schema being used
`schemaVersion`	required	Version of the JSON schema being used. This may be a number or letter (e.g., “1.0” or “A”), and may indicate a site-specific version by as a string of characters (e.g., “RC-V1”).
`encryption`	optional	Specifies whether the `data` portion of the `dataElement` is encrypted. If not specified, or specified as “`NONE`”, the `data` is not encrypted; otherwise, specifies the method used to encrypt the `data`. When specified, the contents of the `data{}` structure (the entire contents inside the braces, but not including them) is encrypted using the indicated methodology. Specific values for the `encryption` field are not specified, nor are the mechanisms for encryption key management, which should occur outside the scope of UUDEX. If additional parameters are required, they may be specified as custom extensions to the standard U-Message header. If `encryption` is specified as anything other than “`NONE`”, `encoding` is required, and `compression` is recommended.
`encoding`	optional	The encoding used to transfer the information. Supported values are “`ASCII`” (ISO/IEC 646) or “`UTF-8`” (ISO 10646) for direct XML transfers; and “`BASE64`”, or “`BASE85`” for encoding binary values into printable ISO/IEC 646 characters for transport. If compression is used, it must be performed before the encoding to ensure that all transmitted characters are ISO/IEC 646. If not specified, “`ASCII`” is assumed, and the native format of the contents *must* be ISO/IEC 646 characters. See Appendix D for additional information.
`compression`	optional	Compression method used to shrink the information prior to

Field	Required / Optional / Recommended	Description
		encoding it. Supported values are "NONE", "ZLIB" and "GZIP" (note ZLIB and GZIP represent the same algorithm – gzip is a stand-alone program that implements the zlib compression library). Additional methods such as "LZ4", "LZ77", "LZ78", "LZO", "LZSS" "LZFSE", "LZVN", "LZW", "DEFLATE", "BZIP2", "LZMA", "LZMA2", "PPM", and "RLE", may be used by mutual agreement of both the publisher and subscriber. If not specified, "NONE" is assumed. See Appendix D for additional information.
properties	optional	Any JSON object (or JSON structure) that is mutually agreed to by the U-Publisher and all U-Subscribers. For example, encryption parameters when specifying encryption.
data	required	Structure to indicate data component of the physicalSecurityIncident.
title	required	A short description of the report as might be used in an index of reports.
summary	optional	A longer description of the incident, summarizing the key elements of the report.
tags	optional	A list of tags (keywords) as quoted strings to facilitate discovery of this report in searches.
relatedReports	optional	This is a list of triples consisting of a type of report (e.g., the schema used to format the report), the report's unique identifier, and a pointer to where the report might be found.
reportStatus	required	Whether this report is preliminary, supplemental, or final.
TLP	required	Distribution indicator using TLP (traffic light protocol) markings. TLP markings are: • "WHITE" – no restriction on distribution • "GREEN" – distribution limited to known trusted peers and partners (i.e., "the community" • "AMBER" – restricted distribution restricted to participant organizations only • "RED" – restricted distribution Additional information is available from DHS CISA
filedBy	required	Contact information for the person/organization filing the report.
timeOfReport	required	When the report was filed.

Incident Characteristics

The `incidentCharacteristics` block consists of a few fields that capture key characteristics of the incident. These include:

```
{
  "incidentSeverity":"",
  "incidentCategory":"",
  "timeOfDetection":"",
  "earliestLikelyOnset":""
}
```

Table 6.17 provides an explanation of the fields used in the `incidentCharacteristics` block for the Physical Security Incident Reporting structure.

Table 6.17: Physical Security Incident Reporting - Incident Characteristics Field Descriptions

Field	Required / Optional / Recommended	Description
`incidentSeverity`	optional	The overall severity of the incident, denoted as nuisance, minor, major, or catastrophic.
`incidentCategory`	optional	The general category of the incident, such as asset-removal (theft), threats, assault, trespass, and other.
`timeOfDetection`	optional	Time when the incident was detected in ISO 8601 format
`earliestLikelyOnset`	optional	The earliest point when the incident might have actually begun in ISO 8601 format

Incident Target Information

The `targetList` block consists of information about the assets targeted in the incident. In consists of a list of objects. Each object may contain the following fields:

```
{
  "targetList":[
    {
      "targetIdentification":"",
      "targetDescription":"",
      "targetOwnerName":"",
      "targetOperatorName":"",
      "targetLocation":{
        "country":"",
        "state":"",
        "code":"",
        "address":""
      },
      "targetImpactState":"",
      "targetRestorationTime":"",
      "targetAdditionalInfo":""
```

```
    }
  ]
)
```

Table 6.18 provides an explanation of the fields used in the `targetList` block for the Physical Security Incident Reporting structure.

Table 6.18: Physical Security Incident Reporting - Incident Target Information Field Descriptions

Field	Required / Optional / Recommended	Description
`targetIdentification`	optional	A short name or identifier for the given target.
`targetDescription`	optional	A short description of the target.
`targetOwnerName`	optional	The name of the person or organization that owns the target.
`targetOperatorName`	optional	The name of the person or organization that operates the target.
`targetLocation`	optional	This can either be a street address or GPS coordinates.
`targetImpactState`	optional	The targets state following the incident. Allowed values are lost, destroyed, degraded, damaged (without being degraded), or unimpacted.
`targetRestorationTime`	optional	The estimated time when the target will resume normal operations in ISO 8601 format
`targetAdditionalInfo`	optional	Any additional information about the target.

This section may list any number of targets.

Incident Impact

The `incidentImpact` block describes the impact of the incident. It contains the following fields:

```
{
  "incidentImpact":{
    "functionalImpact":{
      "severity":"",
      "onsetTime":"",
      "endTime":"",
      "isOngoing":false
    },
    "economicImpact":{
      "severity":"",
      "onsetTime":"",
      "endTime":"",
```

```
      "isOngoing":false
    },
    "riskImpact":{
      "severity":"",
      "onsetTime":"",
      "endTime":"",
      "isOngoing":false
    },
    "energySectorImpact":{
      "controlCenterLossOrFailure":false,
      "controlCenterEvacuation":false,
      "lossOfCCMonitoringOrComm":false,
      "damageOrDisruptFacility":false,
      "islanding":false,
      "failureOfTransmissionOrDistribution":false,
      "transmissionInterruption":false,
      "distributionInterruption":false,
      "uncontrolledFirmLoadLoss":0,
      "firmLoadShedding":0,
      "serviceLossToCustomers":0,
      "systemVoltageReduction":false,
      "facilityVoltageDeviation":false,
      "inadequateResourcesForLoad":false,
      "generatingCapacityLoss":0,
      "offSitePowerLossToNuclear":false,
      "potentialImpactToSystemAdequacy":false,
      "vandalismTargetingSecurity":false,
      "fuelSupplyEmergencies":false,
      "physicalThreatPotentiallyDegrades":false,
      "suspiciousDeviceAtFacility":false,
      "otherEnergySectorImpact":""
    },
    "impactedThirdParties":[
      {
        "impactedOrgName":"",
        "impactedOrgRole":"",
        "impactedOrgMagnitude":"",
        "impactedOrgRiskMagnitude":"",
        "impactedOrgAdditionalInfo":""
      }
    ],
    "additionalImpactInformation":""
  }
}
```

Table 6.19 provides an explanation of the fields used in the `incidentImpact` block for the Physical Security Incident Reporting structure.

Table 6.19: Physical Security Incident Reporting - Incident Impact Field Descriptions

Field	Required / Optional / Recommended	Description
`functionalImpact`	optional	The magnitude of the impact on the mission or operation of the impacted organization
`severity`	optional	The magnitude of the impact (nuisance, minor, major, or catastrophic).
`onsetTime`	optional	The date/time value to indicate the onset of the impact in ISO 8601 format.
`endTime`	optional	The date/time value to indicate when the impact ended or is expected to end in ISO 8601 format.
`isOngoing`	optional	A flag indicating whether impact was ongoing at the time of the report.
`economicImpact`	optional	The magnitude of the impact on the financial health of the impacted organization. This field has the same structure as the `functionalImpact` field.
`riskImpact`	optional	The magnitude of new risk caused by the incident. This field has the same structure as the `functionalImpact` field.
`energySectorImpact`	optional	This consists of a series of fields that correspond to the options in line K (Impact) of the OE-417 report form, extended slightly to capture impacts from lines 1-24 not captured in line K. In most cases, these fields are Boolean, where true would indicate the corresponding box in form OE-417 would be checked. Some fields record the magnitude of the impact, such as the number of impacted customers or the number of MW of load lost. If not specified, the flag is assumed to be false (for Boolean values) or 0 (for numeric values)
`controlCenterLossOrFailure`	optional	Flag indicating that the incident resulted in loss or failure of the control center
`controlCenterEvacuation`	optional	Flag to indicate that the incident caused the evacuation of the control center
`lossOfCCMonitoringOrComm`	optional	Flag to indicate that the incident caused loss of control center monitoring or communications
`damageOrDisruptFacility`	optional	Flag to indicate that the incident resulted in damage or disruption to a facility

Field	Required / Optional / Recommended	Description
islanding	optional	Flag to indicate that the incident resulted in islanding of the transmission system
failureOfTransmissionOrDistribution	optional	Flag to indicate that the incident resulted in a failure of the transmission or distribution system
transmissionInterruption	optional	Flag to indicate that the incident resulted in an interruption to the transmission system
distributionInterruption	optional	Flag to indicate that the incident resulted in a disruption to the distribution system
uncontrolledFirmLoadLoss	optional	Amount of uncontrolled firm load loss in MW
firmLoadShedding	optional	Amount of firm load shed
serviceLossToCustomers	optional	Number of customers out of power
systemVoltageReduction	optional	Flag to indicate that the incident resulted in a reduction of system voltage
facilityVoltageDeviation	optional	Flag to indicate that the incident resulted in a reduction of facility voltage
inadequateResourcesForLoad	optional	Flag to indicate that the incident caused inadequate resources to serve load
generatingCapacityLoss	optional	Flag to indicate that the incident resulted in generation capacity loss
offSitePowerLossToNuclear	optional	Flag to indicate that the incident resulted in loss of offsite power to a nuclear generation station
potentialImpactToSystemAdequacy	optional	Flag to indicate that the incident has the potential to impact system adequacy
vandalismTargetingSecurity	optional	Flag to indicate that the incident is a result of vandalism
fuelSupplyEmergencies	optional	Flag to indicate that the incident is related to a fuel system emergency
physicalThreatPotentiallyDegrades	optional	Flag to indicate that the incident is a physical threat
suspiciousDeviceAtFacility	optional	Flag to indicate that the incident is a suspicious activity at a facility
otherEnergySectorImpact	optional	Flag to indicate that the incident is not categorized

Field	Required / Optional / Recommended	Description
`impactedThirdParties`	optional	This consists of a list meant to capture important third parties who are impacted by the consequences of the reported incident. This would be used to note impacts on parties such as hospitals, police/fire, or other critical elements. Each party is identified and described using the following fields:
`impactedOrgName`	optional	The name of the impacted party.
`impactedOrgRole`	optional	A description of the party's role or significance.
`impactOrgMagnitude`	optional	The severity of the impact on the third party.
`impactedOrgRiskmagnitude`	optional	The degree to which the incident creates risk of negative impact on the party.
`impactedOrgAdditionalInfo`	optional	Any additional information about the impacted party.
`additionalImpactInformation`	optional	Any additional information about the impact of this incident.

Incident Response

The `incidentResponse` block describes the response to the incident. It includes the following fields:

```
{  "incidentResponse":{
    "responseSummary":"",
    "responseTimeline":{
      "predictableTimeline":[
        {
          "milestone":"",
          "milestoneTime":""
        }
      ],
      "contingentTimeline":[
        ""
      ],
      "unknownTimeline":"",
      "unrecoverable":""
    },
    "responderNotification":[
      {
        "contact":{
          "name":"",
          "organization":"",
          "phone":"",
          "email":"",
          "address":{
```

```
            "country":"",
            "state":"",
            "code":"",
            "address":""
          }
        },
        "timeContacted":"",
        "additionalContactInformation":""
      }
    ],
    "additionalResponseInformation":""
  }
}
```

Table 6.20 provides an explanation of the fields used in the `incidentResponse` block for the Physical Security Incident Reporting structure.

Table 6.20: Physical Security Incident Reporting - Incident Response Field Descriptions

Field	Required / Optional / Recommended	Description
`responseSummary`	optional	An overview of the response to the incident.
`responseTimeline`	optional	There are four possible variations of values for this field depending on what is known at the time of the report. This field will contain exactly one of the following subfields:
`predictableTimeline`	optional	This indicates that the timeline for restoration of capabilities is mostly known. It is followed by a list of milestone-date/time pairs, where each pair describes a milestone in the restoration process with an associated estimated date in ISO 8601 format.
`contingentTimeline`	optional	This indicates that the recovery timeline is dependent on factors that create unknown timelines. This is followed by a list of the factors on which the development of any timeline will depend.
`unknownTimeline`	optional	This indicates that, at the time of the report, the process of restoring the service is not yet known. This is followed by an explanation of the reasons.
`unrecoverable`	optional	This indicates that no restoration of this service is anticipated. This is followed by a description of the reasons.
`responderNotification`	optional	This field consists of a list of external parties contacted in response to the incident. Likely parties include law enforcement and other emergency services. Each entry in the list identifies and describes the engagement with one responder. Each such entry includes the following

Field	Required / Optional / Recommended	Description
		fields:
`contact`	optional	Contact information (including name, organization, phone number, email, etc.) of the party contacted.
`timeContacted`	optional	The date and time when the contact was made in ISO 8601 format
`additionalContactInformation`	optional	Additional information about the engagement with this contact.
`additionalResponseInformation`	optional	This field contains any additional information about the incident response.

Attack Characterization

The `attackCharacterization` block is used to describe the incident or attack. It includes the following fields:

```
{
  "attackCharacterization":{
    "attackIntent":{
      "attackIntentService":false,
      "attackIntentEnvironment":false,
      "attackIntentStaffInjury":false,
      "attackIntentPopulationInjury":false,
      "attackIntentFinancialGain":false,
      "attackIntentFinancialLoss":false,
      "attackIntentPolitical":false,
      "attackIntentSpying":false,
      "attackIntentTactical":false,
      "attackIntentUnknown":false,
      "attackIntentOther":""
    },
    "attackActors":[
      {
        "attackerName":"",
        "attackerOrganization":"",
        "attackerTargetRelationship":"",
        "attackerDisposition":"",
        "attackerAdditionalInfo":""
      }
    ],
    "attackMeans":[
      {
        "attackMeansType":"",
        "attackMeansDescription":""
      }
    ],
    "attackAdditionalInformation":""
  }
}
```

Table 6.21 provides an explanation of the fields used in the `attackCharacterization` block for the Physical Security Incident Reporting structure.

Table 6.21: Physical Security Incident Reporting - Attack Characterization Field Descriptions

Field	Required / Optional / Recommended	Description
`attackIntent`	optional	This consists of several fields each associated with a possible attack objective. (E.g., service denial, financial gain, inflicting financial loss, etc.) Each of these subfields is a Boolean value, where a value of true indicates that the associated goal is believed to be part of the attack objective. There is also a text field to describe objectives other than those enumerated.
`attackActors`	optional	This section is used to describe the parties behind the attack, if known. It consists of a list, where each entry represents a known or suspected actor. Each actor is described using the following fields:
`attackerName`	optional	The name of an individual involved in the attack.
`attackerOrganization`	optional	The organization believed to be behind the individual's actions. if individuals are not known, this field can be used to identify an organization in general.
`attackerTargetRelationship`	optional	This field captures whether the attacking individual had a relationship with the targeted organization, or an organization affiliated with the target, either directly or through friends or relatives.
`attackerDisposition`	optional	The state of the attacker at the time of the report. Example values include "at large", incarcerated, or unknown.
`attackerAdditionalInfo`	optional	Additional information about the attacker.
`attackMeans`	optional	This section captures information about how the attack was carried out. It consists of a list of individual means. Each identified means has the following fields:
`attackMeansType`	optional	A standard categorization of a means or method. Examples of field values include projectiles (such as firearms), explosives, drones, personnel-compromise (i.e., suborning members of the target organization), etc.
`attackMeansDescription`	optional	This text field provides additional information about the tool or method.
`attackAdditionalInformation`	optional	This field contains additional information about the attack.

Indicators

The `indicators` block is used to capture information about attack indicators. An indicator is intended to represent something that others might watch for in order to detect similar attack attempts against their organizations. The following indicator classes are described:

```
{
  "indicators":{
    "personIndicators":[
      {
        "apparentAge":1,
        "apparentHeight":{
          "measure":0,
          "units":""
        },
        "apparentWeight":{
          "measure":0,
          "units":""
        },
        "apparentGender":"",
        "apparentEthnicity":"",
        "vocalBehavior":"",
        "physicalBehavior":"",
        "otherPersonCharacteristics":""
      }
    ],
    "vehicleIndicators":[
      {
        "vehicleType":"",
        "vehicleMakeModel":"",
        "vehicleColor":"",
        "vehicleIdentifiers":[
          {
            "identifier":"",
            "identifierType":""
          }
        ],
        "otherVehicleCharacteristics":""
      }
    ],
    "objectIndicators":[
      {
        "objectDescription":"",
        "objectSize":"",
        "objectLocationFound":"",
        "objectHandlingWarnings":"",
        "otherObjectCharacteristics":""
      }
    ],
    "indicatorDecoys":"",
    "additionalIndicatorInformation":""
  },
  "recommendedCourseOfAction":{
    "recommendation":"",
    "followUp":""
  }
```

```
}
```

Table 6.22 provides an explanation of the fields used in the `indicators` block for the Physical Security Incident Reporting structure.

Table 6.22: Physical Security Incident Reporting - Indicators Field Descriptions

Field	Required / Optional / Recommended	Description
`personIndicators`	optional	This consists of a list of persons associated with the incident who may be involved in future incidents. Each list entry describes a person. The following descriptors are employed:
`apparentAge`	optional	The apparent age of the person.
`apparentHeight`	optional	The apparent height of the person.
`apparentWeight`	optional	The apparent weight of the person.
`apparentGender`	optional	The apparent gender of the person.
`apparentEthnicity`	optional	The apparent ethnicity (e.g., skin tone, hair color) of the person.
`vocalBehavior`	optional	Any noticeable elements of the person's voice or speech, such as accent, notable phrases used, etc.
`physicalBehavior`	optional	Any noticeable elements of the person's physical behavior, such as limps, tics, or other actions.
`otherPersonCharacteristics`	optional	Any additional distinguishing features of the person, such as tattoos, scars, etc.
`vehicleIndicators`	optional	This consists of a list of vehicles associated with the incident that might be used in future events. Each list entry describes a vehicle. The following descriptors are employed:
`vehicleType`	optional	The general class of vehicle, such as car, truck, drone, boat, airplane, etc.
`vehicleMakeModel`	optional	The specific make and model of the vehicle.
`vehicleColor`	optional	The color of the vehicle.
`vehicleIdentification`	optional	This consists of a list of identifiers for the vehicle. Examples include license plate numbers of cars or tail numbers for aircraft. Each entry consists of a pair of values: the identifier value and the type of the identifier (e.g., "license plate").

Field	Required / Optional / Recommended	Description
otherVehicleCharacteristics	optional	This field contains any other identifying characteristics of the vehicle, such as observed damage, notable decorations, etc.
objectIndicators	optional	This consists of a list of objects or devices associated with the incident that might be recognizable in future incidents. Each list entry describes an object. The following descriptors are employed:
objectDescription	optional	A description of the object.
objectSize	optional	A description of the size of the object.
objectLocationFound	optional	Where the object was discovered during the incident.
objectHandlingWarnings	optional	This field is used to convey any warnings about handling the object in question. For example, warnings would note if the object was flammable or explosive.
otherObjectCharacteristics	optional	This field contains additional observations about the nature of the object or device.
indicatorDecoys	optional	This field describes any actions that appeared to be meant to draw attention away from the attack target before or during the incident.
additionalIndicatorInformation	optional	This field contains any additional information about indicators associated with the incident.

Recommended Course of Action

The `recommendedCourseOfAction` is section use used by the report author to recommend actions for their peers to take. This section can also be used to request additional information related to the incident. This section has the following fields:

```
{
  "recommendedCourseOfAction":{
    "recommendation":"",
    "followUp":""
  }
}
```

Table 6.23 provides an explanation of the fields used in the `recommendedCourseOfAction` block for the Physical Security Incident Reporting structure.

Table 6.23: Physical Security Incident Reporting - Recommended Course of Action Field Descriptions

Field	Required / Optional / Recommended	Description
`recommendation`	optional	The recommendations as to courses of action to take or information to provide.
`followUp`	optional	This field provides a quick indication of what peers are expected to do. Possible options are none (no action), action (take steps described in the recommendation), or information (provide information identified in the recommendation).

6.4.3 Cybersecurity Threat and Incident Reporting

Cybersecurity threat and incident reporting use the JSON structures developed and supported by the STIX V2.1 specification. The JSON representations are not reproduced here but are available at https://oasis-open.github.io/cti-documentation/stix/intro.html (accessed 09/24/2020).

Incident reports using the STIX formats would be comprised of a dataSet containing one or more different STIX dataElements corresponding to the JSON structures for STIX from the following list.

- attack-pattern
- campaign
- course-of-action
- grouping
- identity
- indicator
- infrastructure
- intrusion-set
- location
- malware
- malware-analysis
- note
- observed-data
- opinion
- report
- threat-actor
- tool
- vulnerability

- relationship
- sighting
- artifact
- autonomous-system
- directory
- domain-name
- email-addr
- email-message
- file
- ipv4-addr
- ipv6-addr
- mac-addr
- mutex
- network-traffic
- process
- software
- url
- user-account
- windows-registry-key
- x509-certificate
- bundle

UUDEX wraps the standard STIX 2.1 structures into a structure similar to the existing `dataSet` structures but identifies it as a STIX structure. The following shows an example extracted from the STIX website in Section 4.1.2 of the web page[1] wrapped inside of the UUDEX dataSet structure.

The `STIXElements` structure is an array containing sub-component structures that are assumed to be related to the same incident. By using the `dataElements` array structure, multiple independent `STIXElements` structured incident reports can be combined into a single U-Message. It is expected that the UUDEX application that publishes or subscribes to the Cybersecurity Threat and Incident Reporting messages will need to process the STIX format directly, since there is no metadata associated with the STIX document name included in the message.

Note that STIX messages are internally linked by the use of the defined fields "`id`", "`source_ref`" and "`target_ref`". Use of the "`correlationID`" field of the U-Message

[1] See https://docs.oasis-open.org/cti/stix/v2.1/cs01/stix-v2.1-cs01.html (accessed 09/24/2020)

header may also be used, but in this case is not required. However, the "correlationID" field should be used to link a STIX message to a PSIR message for the same incident.

Note also that the time format used is specified in the STIX description, and may differ from other time formats used by UUDEX.

Note also that encryption for individual STIX element objects is not specified. If encryption is desired, whole-message encryption should be specified in the header.


```
{
  "header":{
    "messageID":"d5d1c892-974a-11e9-b198-b0c090affff",
    "noun":"STIXElements",
    "origin":"RC",
    "source":"RC",
    "destination":"E-ISAC",
    "timeStamp":"2020-05-13 10:12:09.209124",
    "verb":"create",
    "subject":"E-ISAC/VulnerabilityReport/0001",
    "encryption":"",
    "encoding":"",
    "compression":"",
    "properties":"",
    "sensitivity":"",
    "hashType":"SHA-256",
    "hash":"hash-value-of-payload"
  },
  "dataSet":{
    "dataElements":[
      {
        "STIXElements":[
          {
            "type":"attack-pattern",
            "spec_version":"2.1",
            "id":"attack-pattern--7e33a43e-e34b-40ec-89da-36c9bb2cacd5",
            "created":"2016-05-12T08:17:27.000Z",
            "modified":"2016-05-12T08:17:27.000Z",
            "name":"Spear Phishing as Practiced by Adversary X",
            "description":"A particular form of spear phishing where the atta
cker claims that the target had won a contest, including personal details, to
get them to click on a link.",
            "external_references":[
              {
                "source_name":"capec",
                "external_id":"CAPEC-163"
              }
            ]
          },
          {
            "type":"relationship",
            "spec_version":"2.1",
            "id":"relationship--57b56a43-b8b0-4cba-9deb-34e3e1faed9e",
            "created":"2016-05-12T08:17:27.000Z",
            "modified":"2016-05-12T08:17:27.000Z",
            "relationship_type":"uses",
```

```
            "source_ref":"intrusion-set--0c7e22ad-b099-4dc3-b0df-2ea3f49ae2e6
",
            "target_ref":"attack-pattern--7e33a43e-e34b-40ec-89da-36c9bb2cacd
5"
          },
          {
            "type":"intrusion-set",
            "spec_version":"2.1",
            "id":"intrusion-set--0c7e22ad-b099-4dc3-b0df-2ea3f49ae2e6",
            "created":"2016-05-12T08:17:27.000Z",
            "modified":"2016-05-12T08:17:27.000Z",
            "name":"Adversary X"
          }
        ]
      }
    ]
  }
}
```


Note that the STIXElements structure is an array of individual STIX blocks.

6.4.4 Vulnerability Notification

Vulnerability notifications also use the JSON structures from the STIX documentation, or may use structures defined by NIST for the National Vulnerability Database (NVD)[1].

6.4.5 Patch Notification

Some patch information may be available from the NVD or in STIX feeds. It may also be available from vendors and third-party patch aggregators using customized structures.

6.5 Power System Model Exchange

A UML representation of Power System models is shown in Figure 6-5. The main class is called “PowerSystemModels” and this object inherits from the “core:IdentifiedObject.” This enables the PowerSystemModels to inherit the following fields along with other fields:

- IdentifiedObject.name
- IdentifiedObject.aliasName
- IdentifiedObject.description
- IdentifiedObject.mRID

[1] See https://nvd.nist.gov/General/News/JSON-1-1-Vulnerability-Feed-Release, accessed 8/19/2020

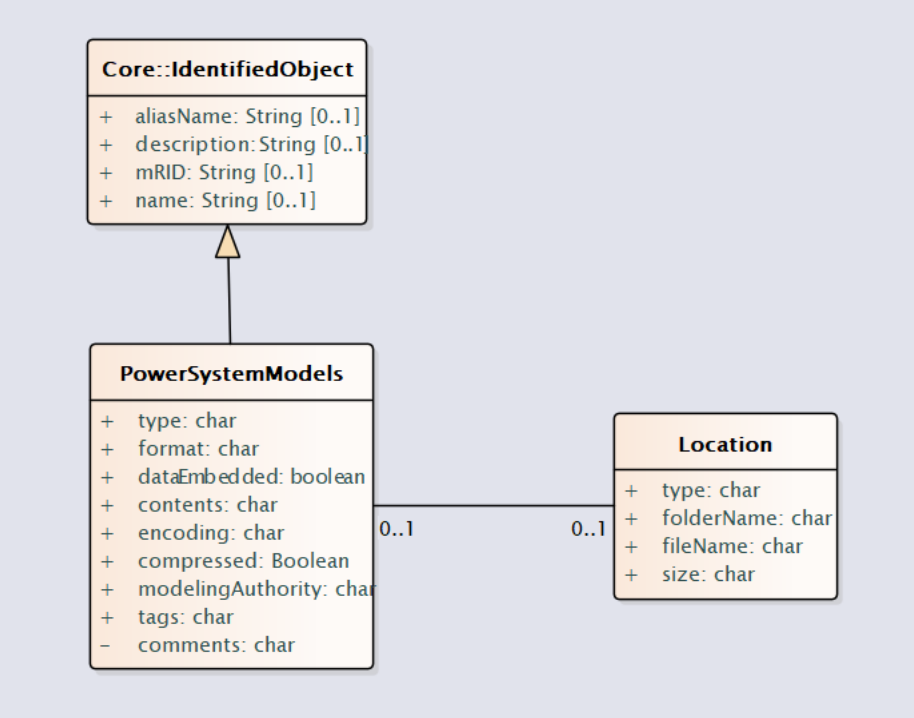

Figure 6-5: Power System Model Exchange Information Model

Note that the dialect of Avro/JSON used in UUDEX does not support classes, so, while the field names used in UUDEX are modeled after the CIM class nomenclature, there are no inherent classes in UUDEX.

Each PowerSystemModel object has a 1:1 association to a “Location” object. This model can be used to exchange either a full or partial model. Depending upon the type of model, certain fields are required to be populated.

The Power System Model Exchange format allows for either including the model file in the U-Message or specifying a location where the model file(s) can be obtained. This is accomplished using the `dataEmbedded` field: if the `dataEmbedded` field is set to True, the model file is contained in the `contents` block and the `location` block is ignored, while if the `dataEmbedded` field is False, the model file(s) are located elsewhere as specified by the `location` block, and the `contents` field is ignored.

When the `dataEmbedded` field is set to True, the Power system model exchanges are accomplished by encoding the power system model file inside a “binary blob”, using a similar compression and encoding mechanism as is used for OE-417 PDF files.

Note that encryption for an individual Power System Model element objects is not specified. If encryption is desired, whole-message encryption should be specified in the header, or the power system model file itself may be encrypted outside of UUDEX prior to encapsulating it in the U-Message.

```
{
  "header":{
```

```
    "messageID":"",
    "noun":"powerSystemModel",
    "verb":"CREATE",
    "subject":"",
    "origin":"",
    "source":"",
    "destination":"",
    "timeStamp":"",
    "correlationID":"",
    "context":"",
    "user":"",
    "comment":"",
    "properties":"",
    "schemaVersion":"",
    "version":"",
    "replyAddress":"",
    "asyncReplyFlag":"",
    "ackRequired":"",
    "expiration":"",
    "encryption":"",
    "encoding":"",
    "compression":"",
    "properties":"",
    "sensitivity":"",
    "hashType":"",
    "hash":""
  },
  "dataSet":{
    "dataElements":[
      {
        "powerSystemModel":{
          "schema":"https://www.uudex.org/uudex/0.1/powerSystemModel",
          "schemaVersion":"0.1",
          "identifiedObjectName : "",
          "identifiedObjectMRID: "",
          "identifiedObjectDescription: "",
          "identifiedObjectAliasName: "",
          "format":"",
          "type":""
          "name":"",
          "modelingAuthority": "",
          "description":"",
          "tags":["",""],
          "comments": "",
          "dataEmbedded": False,
          "encoding":"",
          "compression":"",
          "location": {
            "type": "",
            "site": "",
            "folderName": "",
            "fileName": ["",""],
            "size": ""
          }
          "contents":"",
        }
      }
```

```
    ]
  }
}
```

Table 7.24 describes the fields used for transferring power system models.

Table 6.24: Power System Model Field Descriptions

Field	Required / Optional / Recommended	Description
dataSet	required	Specifies the payload type
dataElements	required	The array of power system model files contained in the message. Typically, only one model file is contained in a single U-Message, but for small incremental or partial models, multiple model files may be contained in a single message.
powerSystemModel	required	Specifies that the message contains a Power System Model file
schema	optional	Pointer to the JSON schema being used
schemaVersion	required	Version of the JSON schema being used. This may be a number or letter (e.g., “1.0” or “A”), and may indicate a site-specific version by as a string of characters (e.g., “RC-V1”).
identifiedObjectName (ion)	optional	the common name of the CIM object as might appear on a one-line
identifiedObjectMRID (iomrid)	required	the unique CIM mRID associated with the CIM object
identifiedObjectDescription (iod)	optional	the descriptive detailed name of the CIM object
identifiedObjectAliasName (ioan)	optional	an alternate name as stored in the CIM description of the object
format	required	Specifies the format of the information. The format is typically the name of the modeling software used (e.g., “PSS/E”, “PSLF”) or a generic format (e.g., “CIM/9”, “CGMES”) but different formats may be transmitted for the same software type and a suitable format must be specified to indicate the differences. If the version of the modeling software is important, it must also be included in the format specification. Specific details of the format must be negotiated between the publisher and subscriber.
type	optional	A string indicating the type of power system model

Field	Required / Optional / Recommended	Description
		file is contained in the message. Expected values are "FULL" or "PARTIAL". Other types may be supported by modeling software. Other values for the type field must be mutually agreed to by the publisher and subscriber. If not specified, "FULL" Is assumed.
name	required / optional	The name of the file containing the power system model. This is used to re-construct the power system model file on the subscribing computer. This field is required if dataEmbedded is set to True; it is optional if dataEmbedded is set to False.
modelingAuthority	required	The name of the modeling authority that created or approved this power system model.
description	recommended	A description of the power system model. The description could also describe any changes between this version of the model and previous versions (e.g., the addition of a line or retirement of a station).
tags	optional	A list (array) of tags or keywords that can be searched to assist in determining whether the model file is needed. Publishers and subscribers should agree on the spelling and format of the tags to ensure that they are useful.
comments	optional	Additional comments about or description of power system model.
dataEmbedded	required	A Boolean flag indication whether the U-Message contains the power system mode information, or whether the U-Message contains a pointer to the power system model information. If the value of dataEmbedded is TRUE, then the power system model is contained in the contents field of the message, and the location block is ignored. If the value of dataEmbedded is FALSE, then the power system model is at the location specified by the location block, and the contents field is ignored.
encoding	required	The encoding used to transfer the information. Supported values are "BASE64", or "BASE85" for encoding binary values into printable ISO/IEC 646 (ASCII) characters for transport. If compression is used, it must be performed before the encoding to ensure that all transmitted characters are ISO/IEC 646. See Appendix D for additional information.

Field	Required / Optional / Recommended	Description
compression	optional	Compression method used to shrink the information prior to encoding it. Supported values are "NONE", "ZLIB" and "GZIP" (note ZLIB and GZIP represent the same algorithm – gzip is a stand-alone program that implements the zlib compression library). Additional methods such as "LZ4", "LZ77", "LZ78", "LZO", "LZSS" "LZFSE", "LZVN", "LZW", "DEFLATE", "BZIP2", "LZMA", "LZMA2", "PPM", and "RLE" may be used by mutual agreement of both the publisher and subscriber. If not specified, "NONE" is assumed. See Appendix D for additional information.
contents	required / optional	A string variable containing the contents of the power system model information if the dataEmbedded field is TRUE. Because the model files are large and most contain binary data, encoding is required, and compression is strongly recommended.
location	required / optional	The data block used to specify the location of the power system model file(s) if the dataEmbedded field is FALSE. Note that credentials such as the username and password used to access the power system model information should not be included in the U-Message and should be communicated to the subscriber using alternate methods such as email or telephone.
type	required	String indicating the mechanism used to store the power system model file. Examples include "SFTP", "DROPBOX", "HTTPS". The type must be mutually agreed to by the publisher and subscriber prior to use.
site	required	The internet-based site where the power system model file is stored. The site should be specified using uniform resource indicator (URI).
folderName	required	Folder at the site where the power system model file(s) are stored.
fileName	required	Name of the power system model file in folderName. If more than 1 file, then use an array of file names.
size	optional	The size of the power system model information.

6.6 Reliability Coordinator Information System (RCIS)

The Reliability Coordinator Information System (RCIS) is used primarily by RCs in North America to post information concerning reliable operations of the Bulk Electric System (BES). The current RCIS is a web-based system that functions much like a message bulletin board, allowing users to post messages to the system, and providing access for other users to view posted messages. Some messages are generated and reported autonomously by software at the RC. Users can either monitor the web interface for new activity or can sign up to receive emails when new messages are posted. Most RCIS messages are free-form text messages, although some message types have a limited message structure.

The following RCIS messages are in the scope of this document.

- Energy Emergency Alert (EEA) reporting
- Frequency Deviations
- Geomagnetic Disturbance (GMD)
- System Emergency
- Transmission Outage
- Generation Outage
- Time Error Correction
- Transmission Loading Relief
- Weather Advisory
- Other types (Free Form)

6.6.1 RCIS Information Model

RCIS messages are implemented as a special case of the IndicationPoint CIM object used in the UUDEX Information Message (see Section 6.3).

A UML representation of RCIS models is shown in Figure 6-5. The main class is called “RCIS” that contains all the fields that are common to different types of RCIS types. This object inherits from the “core::IdentifiedObject.” This enables all RCIS types to inherit the following fields along with RCIS specific fields:

- IdentifiedObject.name
- IdentifiedObject.aliasName
- IdentifiedObject.description
- IdentifiedObject.mRID

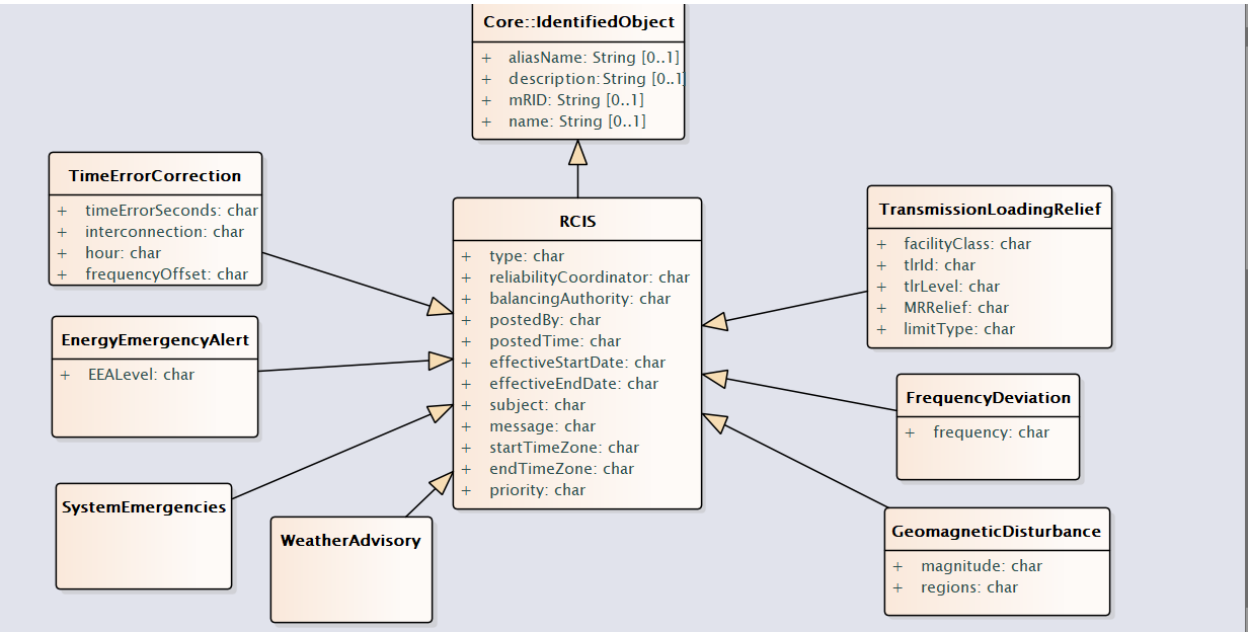

Figure 6-6: RCIS Information Model

The fields specified in Section 6.6.2.1 are defined in the “RCIS” class and thus are inherited by all the different types of RCIS messages. These fields are referred to as “commonFields.”

Note that the dialect of Avro/JSON used in UUDEX does not support classes, so, while the field names used in UUDEX are modeled after the CIM class nomenclature, there are no inherent classes in UUDEX.

6.6.2 RCIS Messages

The following sections describe the message formats used to convey the RCIS messages.

Note that encryption for individual RCIS element objects is not specified. If encryption is desired, whole-message encryption should be specified in the header.

6.6.2.1 RCIS Common Fields

As noted above, all RCIS messages contain a core set of common fields. These common fields are described in this section.

```
{
  "dataSet":{
    "dataElements":[
      {
        "RCIS":{
          "schema":"https://www.uudex.org/uudex/0.1/RCIS",
          "schemaVersion":"0.1",
          "identifiedObjectName":"",
          "identifiedObjectMRID":"",
          "identifiedObjectDescription":"",
          "identifiedObjectAliasName":"",
          "postedBy":"",
          "postedAt":"",
```

```
          "effectiveStartTime":"",
          "effectiveEndTime":"",
          "subject":"",
          "reliabilityCoordinator":"",
          "balancingAuthority":"",
          "message":""
        }
      }
    ]
  }
}
```

Table 6.25 provides an explanation of the common message fields used in the RCIS message block.

Table 6.25: RCIS Message Common Fields

Field (abbreviation)	Required / Optional / Recommended	Description
`dataSet (ds)`	required	Specifies the payload type
`dataElements (de)`	required	The array of data elements contained in the message
`RCIS`	required	Specifies the payload contains RCIS information (note – the specific RCIS message is specified in the following sections).
`schema (sch)`	optional	Pointer to the JSON schema being used
`schemaVersion (sver)`	optional	Version of the JSON schema being used. This may be a number or letter (e.g., “1.0” or “A”), and may indicate a site-specific version by as a string of characters (e.g., “RC-V1”).
`identifiedObjectName (ion)`	optional	the common name of the CIM object as might appear on a one-line
`identifiedObjectMRID (iomrid)`	required	the unique CIM mRID associated with the CIM object
`identifiedObjectDescription (iod)`	optional	the descriptive detailed name of the CIM object
`identifiedObjectAliasName (ioan)`	optional	an alternate name as stored in the CIM description of the object
`postedBy (pby)`	required	Identity of the organization that posted the RCIS message
`postedAt (pat)`	required	Time when the RCIS message was posted in ISO 8601 format

Field (abbreviation)	Required / Optional / Recommended	Description
`effectiveStartTime (est)`	optional	Time when the condition contained in RCIS message is to start in ISO 8601 format
`effectiveEndTime (eet)`	optional	Time when the condition contained in RCIS message is to end in ISO 8601 format
`subject`	required	The subject of the RCIS message, a free form field
`reliabilityCoordinator (rc)`	optional	Identity of the impacted RC
`balancingAuthority (ba)`	optional	Identity of the impacted Balancing Authority (BA).
`message`	optional / required	Free form field containing required or supplemental information for the RCIS message. For some RCIS messages, this field is required.

6.6.2.2 Energy Emergency Alert (EEA)

Energy Emergency Alert (EEA) reporting by RCs required by NERC Standard EOP-002 from the time such an alert is issued to the time the alert has been cancelled. To declare an EEA, the RCIS message is used with a positive `EEALevel` value. To cancel an EEA, the same RCIS message is issued with an `EEALevel` of "0".

```
{
  "dataSet":{
    "dataElements":[
      {
        "RCIS-EEA":{
          "schema":"https://www.uudex.org/uudex/0.1/RCIS",
          "schemaVersion":"0.1",
          "identifiedObjectName":"",
          "identifiedObjectMRID":"",
          "identifiedObjectDescription":"",
          "identifiedObjectAliasName":"",
          "postedBy":"",
          "postedAt":"",
          "effectiveStartTime":"",
          "effectiveEndTime":"",
          "subject":"",
          "reliabilityCoordinator":"",
          "balancingAuthority":"",
          "message":"",
          "EEALevel":""
        }
      }
    ]
  }
```

```
}
```

Table 6.26 provides an explanation of the common message fields used in the RCIS Energy Emergency Alert (EEA) message block.

Table 6.26: RCIS Energy Emergency Alert (EEA) Message Fields

Field (abbreviation)	Required / Optional / Recommended	Description
`dataSet (ds)`	required	Specifies the payload type
`dataElements (de)`	required	The array of data elements contained in the message
`RCIS-EEA`	required	Specifies the payload an RCIS Energy Emergency Alert (EEA) message
`schema (sch)`	optional	Pointer to the JSON schema being used
`schemaVersion (sver)`	optional	Version of the JSON schema being used. This may be a number or letter (e.g., “1.0” or “A”), and may indicate a site-specific version by as a string of characters (e.g., “RC-V1”).
`identifiedObjectName (ion)`	optional	See Table 6.25
`identifiedObjectMRID (iomrid)`	required	See Table 6.25
`identifiedObjectDescription (iod)`	optional	See Table 6.25
`identifiedObjectAliasName (ioan)`	optional	See Table 6.25
`postedBy (pby)`	required	See Table 6.25
`postedAt (pat)`	required	See Table 6.25
`effectiveStartTime (est)`	optional	See Table 6.25
`effectiveEndTime (eet)`	optional	See Table 6.25
`Subject (sub)`	required	See Table 6.25
`reliabilityCoordinator (rc)`	required	See Table 6.25
`balancingAuthority (ba)`	required	See Table 6.25
`message (msg)`	optional	See Table 6.25
`EEAlevel (eea)`	required	The Energy Emergency Alert (EEA) level

Field (abbreviation)	Required / Optional / Recommended	Description
		being declared by this message. EEA levels and meanings are specified in NERC Standard EOP-002.

6.6.2.3 Frequency Deviation

Frequency Deviation reports are used for communicating system events that have or could result in a rapid change in frequency that significantly impacts system operation; also used to report changes in frequency for which the cause is unknown.

```
{
  "dataSet":{
    "dataElements":[
      {
        "RCIS-FD":{
          "schema":"https://www.uudex.org/uudex/0.1/RCIS",
          "schemaVersion":"0.1",
          "identifiedObjectName":"",
          "identifiedObjectMRID":"",
          "identifiedObjectDescription":"",
          "identifiedObjectAliasName":"",
          "postedBy":"",
          "postedAt":"",
          "effectiveStartTime":"",
          "effectiveEndTime":"",
          "subject":"",
          "reliabilityCoordinator":"",
          "balancingAuthority":"",
          "message":"",
          "frequency":""
        }
      }
    ]
  }
}
```

Table 6.27 provides an explanation of the common message fields used in the RCIS Frequency Deviation message block.

Table 6.27: RCIS Frequency Deviation Message Fields

Field (abbreviation)	Required / Optional / Recommended	Description
`dataSet (ds)`	required	Specifies the payload type
`dataElements (de)`	required	The array of data elements contained in the message
`RCIS-FD`	required	Specifies the payload contains an RCIS

Field (abbreviation)	Required / Optional / Recommended	Description
		Frequency Deviation message.
schema (sch)	optional	Pointer to the JSON schema being used
schemaVersion (sver)	optional	Version of the JSON schema being used. This may be a number or letter (e.g., “1.0” or “A”), and may indicate a site-specific version by as a string of characters (e.g., “RC-V1”).
identifiedObjectName (ion)	optional	See Table 6.25
identifiedObjectMRID (iomrid)	required	See Table 6.25
identifiedObjectDescription (iod)	optional	See Table 6.25
identifiedObjectAliasName (ioan)	optional	See Table 6.25
postedBy (pby)	required	See Table 6.25
postedAt (pat)	required	See Table 6.25
effectiveStartTime (est)	optional	See Table 6.25
effectiveEndTime (eet)	optional	See Table 6.25
subject (sub)	required	See Table 6.25
reliabilityCoordinator (rc)	optional	See Table 6.25
balancingAuthority (ba)	optional	See Table 6.25
message (msg)	optional	See Table 6.25
frequency (frq)	required	The observed system frequency corresponding to this reported event.

6.6.2.4 Geomagnetic Disturbance (GMD)

Geomagnetic Disturbance (GMD) reports capture GMD information originated from the National Oceanic and Atmospheric Administration (NOAA) Space Weather Prediction Center (SWPC) and is made available by designated RCs to receive and disseminate notifications of possible GMDs to RCs, Balancing Authorities, and TOPs.

```
{
  "dataSet":{
    "dataElements":[
      {
```

```
        "RCIS-GMD":{
          "schema":"https://www.uudex.org/uudex/0.1/RCIS",
          "schemaVersion":"0.1",
          "identifiedObjectName":"",
          "identifiedObjectMRID":"",
          "identifiedObjectDescription":"",
          "identifiedObjectAliasName":"",
          "postedBy":"",
          "postedAt":"",
          "effectiveStartTime":"",
          "effectiveEndTime":"",
          "subject":"",
          "reliabilityCoordinator":"",
          "balancingAuthority":"",
          "message":"",
          "magnitude":"",
          "region":""
        }
      }
    ]
  }
}
```

Table 6.28 provides an explanation of the common message fields used in the RCIS Geomagnetic Disturbance message block.

Table 6.28: RCIS Geomagnetic Disturbance Message Fields

Field (abbreviation)	Required / Optional / Recommended	Description
dataSet (ds)	required	Specifies the payload type
dataElements (de)	required	The array of data elements contained in the message
RCIS-GMD	required	Specifies the payload an RCIS Geomagnetic Disturbance message.
schema (sch)	optional	Pointer to the JSON schema being used
schemaVersion (sver)	optional	Version of the JSON schema being used. This may be a number or letter (e.g., “1.0” or “A”), and may indicate a site-specific version by as a string of characters (e.g., “RC-V1”).
identifiedObjectName (ion)	optional	See Table 6.25
identifiedObjectMRID (iomrid)	required	See Table 6.25
identifiedObjectDescription (iod)	optional	See Table 6.25

Field (abbreviation)	Required / Optional / Recommended	Description
identifiedObjectAliasName (ioan)	optional	See Table 6.25
postedBy (pby)	required	See Table 6.25
postedAt (pat)	required	See Table 6.25
effectiveStartTime (est)	optional	See Table 6.25
effectiveEndTime (eet)	optional	See Table 6.25
Subject (sub)	required	See Table 6.25
reliabilityCoordinator (rc)	optional	See Table 6.25
balancingAuthority (ba)	optional	See Table 6.25
message (msg)	optional	See Table 6.25
magnitude (mag)	required	Magnitude of the geomagnetic disturbance (typically a reference to a K index, or G scale value[1])
region (reg)	required	String containing the list of impacted regions

6.6.2.5 System Emergency

System Emergency messages are used to provide notification when a RC foresees transmission problems (such as a System operation Limit [SOL] or interconnection operating reliability limit [IROL] violation, loss of reactive reserves, etc.), when results of operational studies for the current day or the next day indicate that there is potential for SOL or IROL violations, or when an interconnected system separation, system islanding, or blackout has occurred.

```
{
  "dataSet":{
    "dataElements":[
      {
        "RCIS-SE":{
          "schema":"https://www.uudex.org/uudex/0.1/RCIS",
          "schemaVersion":"0.1",
          "identifiedObjectName":"",
          "identifiedObjectMRID":"",
          "identifiedObjectDescription":"",
          "identifiedObjectAliasName":"",
```

[1] See https://www.swpc.noaa.gov/noaa-scales-explanation, accessed 02/01/2021

```
            "postedBy":"",
            "postedAt":"",
            "effectiveStartTime":"",
            "effectiveEndTime":"",
            "subject":"",
            "reliabilityCoordinator":"",
            "balancingAuthority":"",
            "message":""
          }
        }
      ]
    }
}
```

Table 6.29 provides an explanation of the common message fields used in the RCIS System Emergency message block.

Table 6.29: RCIS System Emergency Message Fields

Field (abbreviation)	Required / Optional / Recommended	Description
dataSet (ds)	required	Specifies the payload type
dataElements (de)	required	The array of data elements contained in the message
RCIS-GMD	required	Specifies the payload an RCIS System Emergency message.
schema (sch)	optional	Pointer to the JSON schema being used
schemaVersion (sver)	optional	Version of the JSON schema being used. This may be a number or letter (e.g., “1.0” or “A”), and may indicate a site-specific version by as a string of characters (e.g., “RC-V1”).
identifiedObjectName (ion)	optional	See Table 6.25
identifiedObjectMRID (iomrid)	required	See Table 6.25
identifiedObjectDescription (iod)	optional	See Table 6.25
identifiedObjectAliasName (ioan)	optional	See Table 6.25
postedBy (pby)	required	See Table 6.25
postedAt (pat)	required	See Table 6.25
effectiveStartTime (est)	optional	See Table 6.25

Field (abbreviation)	Required / Optional / Recommended	Description
effectiveEndTime (eet)	optional	See Table 6.25
Subject (sub)	required	See Table 6.25
reliabilityCoordinator (rc)	optional	See Table 6.25
balancingAuthority (ba)	optional	See Table 6.25
message (msg)	required	The details of the declared system emergency.

6.6.2.6 Transmission Outage

Transmission Outage reports contain messages relating to transmission line outages for facilities greater than 230kV, automatically generated and sent to the NERC System Data exchange (SDX) database then posted on the RCIS.

```
{
  "dataSet":{
    "dataElements":[
      {
        "RCIS-TO":{
          "schema":"https://www.uudex.org/uudex/0.1/RCIS",
          "schemaVersion":"0.1",
          "identifiedObjectName":"",
          "identifiedObjectMRID":"",
          "identifiedObjectDescription":"",
          "identifiedObjectAliasName":"",
          "postedBy":"",
          "postedAt":"",
          "effectiveStartTime":"",
          "effectiveEndTime":"",
          "subject":"",
          "reliabilityCoordinator":"",
          "balancingAuthority":"",
          "message":"",
          "eventType":"",
          "facility":""
        }
      }
    ]
  }
}
```

Table 6.30 provides an explanation of the common message fields used in the RCIS Transmission Outage message block.

Table 6.30: RCIS Transmission Outage Message Fields

Field (abbreviation)	Required / Optional / Recommended	Description
dataSet (ds)	required	Specifies the payload type
dataElements (de)	required	The array of data elements contained in the message
RCIS-TO	required	Specifies the payload an RCIS Transmission Outage message.
schema (sch)	optional	Pointer to the JSON schema being used
schemaVersion (sver)	optional	Version of the JSON schema being used. This may be a number or letter (e.g., “1.0” or “A”), and may indicate a site-specific version by as a string of characters (e.g., “RC-V1”).
identifiedObjectName (ion)	optional	See Table 6.25
identifiedObjectMRID (iomrid)	required	See Table 6.25
identifiedObjectDescription (iod)	optional	See Table 6.25
identifiedObjectAliasName (ioan)	optional	See Table 6.25
postedBy (pby)	required	See Table 6.25
postedAt (pat)	required	See Table 6.25
effectiveStartTime (est)	optional	See Table 6.25
effectiveEndTime (eet)	optional	See Table 6.25
Subject (sub)	required	See Table 6.25
reliabilityCoordinator (rc)	optional	See Table 6.25
balancingAuthority (ba)	optional	See Table 6.25
message (msg)	optional	See Table 6.25
eventType (evt)	required	The type of transmission outage being reported.
Facility (fac)	required	The transmission facility being declared out of service by this message.

6.6.2.7 Generation Outage

Generation Outage messages relating to generation facility outages greater than 300MW, automatically generated and sent to the SDX database then posted on the RCIS.

```
{
  "dataSet":{
    "dataElements":[
      {
        "RCIS-GO":{
          "schema":"https://www.uudex.org/uudex/0.1/RCIS",
          "schemaVersion":"0.1",
          "identifiedObjectName":"",
          "identifiedObjectMRID":"",
          "identifiedObjectDescription":"",
          "identifiedObjectAliasName":"",
          "postedBy":"",
          "postedAt":"",
          "effectiveStartTime":"",
          "effectiveEndTime":"",
          "subject":"",
          "reliabilityCoordinator":"",
          "balancingAuthority":"",
          "message":"",
          "eventType":"",
          "facility":""
        }
      }
    ]
  }
}
```

Table 6.31 provides an explanation of the common message fields used in the RCIS Generation Outage message block.

Table 6.31: RCIS Generation Outage message Fields

Field (abbreviation)	Required / Optional / Recommended	Description
`dataSet (ds)`	required	Specifies the payload type
`dataElements (de)`	required	The array of data elements contained in the message
`RCIS-GO`	required	Specifies the payload an RCIS Generation Outage message.
`schema (sch)`	optional	Pointer to the JSON schema being used
`schemaVersion (sver)`	optional	Version of the JSON schema being used. This may be a number or letter (e.g., “1.0” or “A”), and may indicate a site-specific version by as a string of characters (e.g.,

Field (abbreviation)	Required / Optional / Recommended	Description
		"RC-V1").
identifiedObjectName (ion)	optional	See Table 6.25
identifiedObjectMRID (iomrid)	required	See Table 6.25
identifiedObjectDescription (iod)	optional	See Table 6.25
identifiedObjectAliasName (ioan)	optional	See Table 6.25
postedBy (pby)	required	See Table 6.25
postedAt (pat)	required	See Table 6.25
effectiveStartTime (est)	optional	See Table 6.25
effectiveEndTime (eet)	optional	See Table 6.25
Subject (sub)	required	See Table 6.25
reliabilityCoordinator (rc)	optional	See Table 6.25
balancingAuthority (ba)	optional	See Table 6.25
message (msg)	optional	See Table 6.25
eventType (evt)	required	The type of generation outage being reported.
facility (fac)	required	The generation facility being declared out of service by this message.

6.6.2.8 Time Error Correction (TEC)

Time Error Correction (TEC) messages are used as the indication of the start and end of time error correction within an interconnection.

```
{
  "dataSet":{
    "dataElements":[
      {
        "RCIS-TEC":{
          "schema":"https://www.uudex.org/uudex/0.1/RCIS",
          "schemaVersion":"0.1",
          "identifiedObjectName":"",
          "identifiedObjectMRID":"",
          "identifiedObjectDescription":"",
          "identifiedObjectAliasName":"",
```

```
          "postedBy":"",
          "postedAt":"",
          "effectiveStartTime":"",
          "effectiveEndTime":"",
          "subject":"",
          "reliabilityCoordinator":"",
          "balancingAuthority":"",
          "message":"",
          "timeErrorSeconds":"",
          "interconnection":"",
          "startHour":"",
          "frequencyOffset":""
        }
      }
    ]
  }
}
```

Table 6.32 provides an explanation of the common message fields used in the RCIS Time Error Correction (TEC) message block.

Table 6.32: RCIS Time Error Correction Message Fields

Field (abbreviation)	Required / Optional / Recommended	Description
dataSet (ds)	required	Specifies the payload type
dataElements (de)	required	The array of data elements contained in the message
RCIS-TEC	required	Specifies the payload an RCIS Time Error Correction message.
schema (sch)	optional	Pointer to the JSON schema being used
schemaVersion (sver)	optional	Version of the JSON schema being used. This may be a number or letter (e.g., “1.0” or “A”), and may indicate a site-specific version by as a string of characters (e.g., “RC-V1”).
identifiedObjectName (ion)	optional	See Table 6.25
identifiedObjectMRID (iomrid)	required	See Table 6.25
identifiedObjectDescription (iod)	optional	See Table 6.25
identifiedObjectAliasName (ioan)	optional	See Table 6.25
postedBy (pby)	required	See Table 6.25

Field (abbreviation)	Required / Optional / Recommended	Description
postedAt (pat)	required	See Table 6.25
effectiveStartTime (est)	optional	See Table 6.25
effectiveEndTime (eet)	optional	See Table 6.25
Subject (sub)	required	See Table 6.25
reliabilityCoordinator (rc)	optional	See Table 6.25
balancingAuthority (ba)	optional	See Table 6.25
message (msg)	optional	See Table 6.25
timeErrorSeconds	required	The time error in seconds (positive or negative)
interconnection	required	The interconnections reporting the time error correction, e.g., EI, WECC, ERCOT, QUEBEC.
startHour	required	The hour number at which the frequency adjustment will start to correct the time error.
frequencyOffset	required	The frequency offset to be applied to correct the time error. A return to normal is indicated by specifying a value of “0”, i.e., no offset.

6.6.2.9 Transmission Loading Relief (TLR)

Transmission Loading relief (TLR) messages relating to transmission loading relief following NERC Standard IRO-006, automatically generated by the Interchange Distribution Calculator (IDC) then posted to the RCIS.

```
{
  "dataSet":{
    "dataElements":[
      {
        "RCIS-TLR":{
          "schema":"https://www.uudex.org/uudex/0.1/RCIS",
          "schemaVersion":"0.1",
          "identifiedObjectName":"",
          "identifiedObjectMRID":"",
          "identifiedObjectDescription":"",
          "identifiedObjectAliasName":"",
          "postedBy":"",
          "postedAt":"",
          "effectiveStartTime":"",
```

```
          "effectiveEndTime":"",
          "subject":"",
          "reliabilityCoordinator":"",
          "balancingAuthority":"",
          "message":"",
          "facilityClass":"",
          "tlrId":"",
          "tlrLevel":"",
          "mrRelief":"",
          "limitType":""
        }
      }
    ]
  }
}
```

Table 6.33provides an explanation of the common message fields used in the RCIS Transmission Loading Relief (TLR) message block.

Table 6.33: RCIS Transmission Loading Relief (TLR) Message Fields

Field (abbreviation)	Required / Optional / Recommended	Description
`dataSet (ds)`	required	Specifies the payload type
`dataElements (de)`	required	The array of data elements contained in the message
`RCIS-TLR`	required	Specifies the payload an RCIS Transmission Loading Relief message.
`schema (sch)`	optional	Pointer to the JSON schema being used
`schemaVersion (sver)`	optional	Version of the JSON schema being used. This may be a number or letter (e.g., “1.0” or “A”), and may indicate a site-specific version by as a string of characters (e.g., “RC-V1”).
`identifiedObjectName (ion)`	optional	See Table 6.25
`identifiedObjectMRID (iomrid)`	required	See Table 6.25
`identifiedObjectDescription (iod)`	optional	See Table 6.25
`identifiedObjectAliasName (ioan)`	optional	See Table 6.25
`postedBy (pby)`	required	See Table 6.25
`postedAt (pat)`	required	See Table 6.25

Field (abbreviation)	Required / Optional / Recommended	Description
effectiveStartTime (est)	optional	See Table 6.25
effectiveEndTime (eet)	optional	See Table 6.25
Subject (sub)	required	See Table 6.25
reliabilityCoordinator (rc)	optional	See Table 6.25
balancingAuthority (ba)	optional	See Table 6.25
message (msg)	optional	See Table 6.25
facilityClass	required	The facility class associated with the TLR.
tlrId	required	The ID assigned to the TLR report. This ID is generated by the reporting entity.
tlrLevel	required	The TLR level being reported.
mrRelief	required	The relief for the TLR.
limitType	required	The type of limit that is causing the TPOR to be reported.

6.6.2.10 Weather Advisory

Weather Advisory messages are notifications of approaching or existing severe or extreme weather conditions that have the potential to affect system reliability. These conditions could include severe heat or cold, insulator ice bridging, large thermal generation limitations (due to fuel restrictions), tower damage (due to tornado, hurricane, or flooding), extensive ice storms, galloping on transmission circuits, forest fires etc.

```
{
  "dataSet":{
    "dataElements":[
      {
        "RCIS-WA":{
          "schema":"https://www.uudex.org/uudex/0.1/RCIS",
          "schemaVersion":"0.1",
          "identifiedObjectName":"",
          "identifiedObjectMRID":"",
          "identifiedObjectDescription":"",
          "identifiedObjectAliasName":"",
          "postedBy":"",
          "postedAt":"",
          "effectiveStartTime":"",
          "effectiveEndTime":"",
          "subject":"",
          "reliabilityCoordinator":"",
          "balancingAuthority":"",
```

```
          "message":""
        }
      }
    ]
  }
}
```

Table 6.34 provides an explanation of the common message fields used in the RCIS Weather Advisory message block.

Table 6.34: RCIS Weather Advisory Message Fields

Field (abbreviation)	Required / Optional / Recommended	Description
dataSet (ds)	required	Specifies the payload type
dataElements (de)	required	The array of data elements contained in the message
RCIS-WA	required	Specifies the payload an RCIS Weather Advisory message.
schema (sch)	optional	Pointer to the JSON schema being used
schemaVersion (sver)	optional	Version of the JSON schema being used. This may be a number or letter (e.g., “1.0” or “A”), and may indicate a site-specific version by as a string of characters (e.g., “RC-V1”).
identifiedObjectName (ion)	optional	See Table 6.25
identifiedObjectMRID (iomrid)	required	See Table 6.25
identifiedObjectDescription (iod)	optional	See Table 6.25
identifiedObjectAliasName (ioan)	optional	See Table 6.25
postedBy (pby)	required	See Table 6.25
postedAt (pat)	required	See Table 6.25
effectiveStartTime (est)	optional	See Table 6.25
effectiveEndTime (eet)	optional	See Table 6.25
Subject (sub)	required	See Table 6.25
reliabilityCoordinator (rc)	optional	See Table 6.25

Field (abbreviation)	Required / Optional / Recommended	Description
balancingAuthority (ba)	optional	See Table 6.25
message (msg)	required	The text of the weather advisory

6.6.2.11 Free Form

Free Form (also referred to as Other) messages are designed to capture situations that a RC determines to be appropriate to communicate with other RCs, Balancing Authorities or TOPs regarding an issue that is not directly related to any of the other message board categories available through other RCIS message types. A number of free-form message classes have come into common use, even though there is not a specific RCIS message class for them. If there is sufficient interest, UUDEX could consider specifically creating message classes including, but not limited to, the following:

- Test & Maintenance – messages indicating testing of the RCIS system, such as test notification, or testing of successful message submission, or maintenance of the RCIS.
- Drills & Exercises – notification of operational drills, such as testing of backup control center locations, testing of new communications facilities, (parallel) testing of new platforms and applications, etc. (Note – actual evacuation should be reported as an Emergency).
- Software Issue – notification that the RC, BA, or TOP is experiencing Energy Management System or software issues, such as State Estimator not converging, or installation or testing of major software updates.
- Timing Integrity Issue – notification that a RC, BA, or TOP has detected a timing integrity issue such as that caused by spoofing the time from GPS receiver. The issue must be of sufficient magnitude to affect the timing alignment of SCADA, PMU, or other time sensitive data across a RC, BA, or TOP geographic area.
- Theft, Burglary, Vandalism – Reports of (mostly) nuisance events that do not have a direct impact on operations, like copper theft, substation break-in, and bomb threats, that may also be report in more detail using the PSIR format.
- Non-Transmission Emergency – Similar to System Emergency, but not for transmission issues. Examples include control room evacuations due to fire or bomb threat, or physical damage to a transmission station (like fire or flood) that does not necessarily induce an IROL violation, islanding, or cascading (therefore not qualifying as a System Emergency).

The specific type of free form message should be indicated by the subject field of the message.

Even if specific RCIS messages are created for these message types, the Free Form message should be retained for other less frequent and unspecified messages.

```
{
  "dataSet":{
    "dataElements":[
      {
        "RCIS-OTHER":{
```

```
            "schema":"https://www.uudex.org/uudex/0.1/RCIS",
            "schemaVersion":"0.1",
            "identifiedObjectName":"",
            "identifiedObjectMRID":"",
            "identifiedObjectDescription":"",
            "identifiedObjectAliasName":"",
            "postedBy":"",
            "postedAt":"",
            "effectiveStartTime":"",
            "effectiveEndTime":"",
            "subject":"",
            "reliabilityCoordinator":"",
            "balancingAuthority":"",
            "message":""
          }
        }
      ]
    }
}
```

Table 6.35 provides an explanation of the common message fields used in the RCIS Free Form message block.

Table 6.35: RCIS Free Form Message Fields

Field (abbreviation)	Required / Optional / Recommended	Description
`dataSet (ds)`	required	Specifies the payload type
`dataElements (de)`	required	The array of data elements contained in the message
`RCIS-OTHER`	required	Specifies the payload an RCIS Free Form (other) message.
`schema (sch)`	optional	Pointer to the JSON schema being used
`schemaVersion (sver)`	optional	Version of the JSON schema being used. This may be a number or letter (e.g., “1.0” or “A”), and may indicate a site-specific version by as a string of characters (e.g., “RC-V1”).
`identifiedObjectName (ion)`	optional	See Table 6.25
`identifiedObjectMRID (iomrid)`	required	See Table 6.25
`identifiedObjectDescription (iod)`	optional	See Table 6.25
`identifiedObjectAliasName (ioan)`	optional	See Table 6.25

Field (abbreviation)	Required / Optional / Recommended	Description
`postedBy (pby)`	required	See Table 6.25
`postedAt (pat)`	required	See Table 6.25
`effectiveStartTime (est)`	optional	See Table 6.25
`effectiveEndTime (eet)`	optional	See Table 6.25
`Subject (sub)`	required	See Table 6.25
`reliabilityCoordinator (rc)`	optional	See Table 6.25
`balancingAuthority (ba)`	optional	See Table 6.25
`message (msg)`	required	The text of the free form RCIS message

7.0 Approach for Generic Information Encapsulation

The information exchange models used by UUDEX are extremely flexible and allow for the exchange of numerous types of structured and unstructured information. This approach can be used to define additional information exchanges not specified in this document, such as for exchange of market information, or the exchange of new unanticipated information.

In many cases, the embedded structures are referred to as “documents” (e.g., JSON documents or XML documents). The documents when extracted from the U-Message in their native format can be processed directly by applications expecting them in that format.

This section discusses generalized versions of transfers that have already been discussed in previous sections.

7.1 Native JSON Structured Information

Embedding a JSON structure with a U-Message is the most straightforward approach and is what was used to encapsulate the STIX messages in Section 6.4.3. This approach has the advantage of not needing to do any translation or wrapping of the native format into the U-Message structure, and there are no translation issues associated with creating the JSON message. The native JSON “document” is inserted into the U-Message as an embedded structure and can be extracted from it and used directly.

If information structures do not already exist (or can be replaced by updated versions), using native JSON structures with a UUDEX environment makes the most sense, since it minimizes the amount of external processing or file creation required to produce or consume the information. For complicated JSON structures, a JSON schema (or a reference to an external JSON schema file) could be included in the structure to aid the interpretation of the message.

```
{
  "header":{
    "messageID":"",
    "noun":"xyxy",
    "verb":"create",
    "origin":"",
    "source":"",
    "destination":"",
    "timeStamp":"",
    "subject":"",
    "hashType":"",
    "hash":""
  },
  "dataSet":{
    "dataElements":[
      {
        "xyxy":{
          "schema":"https://www.uudex.org/uudex/0.1/xyxy",
          "schemaVersion":"0.1",
          "encryption":"",
          "encoding":"",
          "compression":"",
          "properties":"",
          "data":{
```

```
            "field1":"",
            "field2":"",
            "field3":""
          }
        }
      }
    ]
  }
}
```

7.2 Structured Non-JSON Information

The next most straightforward approach is to wrap a structured document in the UUDEX JSON message format, similar to how the DOE OE-417 XML format in Section 6.4.1.

This approach makes sense in cases where existing structures and their associate processing (i.e., creation and consumption of the information by existing applications) will not be changed. This approach also allows UUDEX and non-UUDEX transfers to occur in parallel, for example while transitioning from a legacy information exchange to using UUDEX.

An alternative approach could be to translate the structured format (e.g., XML document) to native JSON, but in many cases, the translation is not straightforward, and is not reversable. Translated documents cannot be used directly by the original application, requiring a reverse translation, which is not always workable. For example, XML documents can be translated into JSON, but JSON does not support “classes”, so any XML class structures are lost.

An example message for embedding a generic structured document follows. The `format` field specifies the native structured format, with the document name specified by the `name` field. Since many structured documents contain printable ISO/IEC 646 characters (rather than binary data), the `encoding` and `compression` fields will probably not be necessary but could be included for use if necessary. For example, a large IEC 61850 “SCL” file of over 230000 human readable XML records is over 8MB in size, but if compressed using gzip and encoded using BASE64 encoding, it shrinks to a little over 500 kB. This is especially true if the native file contains significant “white space” to aid human readability of the contents.

The U-publish client will read the contents of the structured format document file specified in the `file` field and embed the contents as a string in the `contents` field (optionally compressing and encoding it), while the UUDEX Subscribe client will extract the contents (decoding and decompressing it if specified) and create a file based on the specified `name` field. The native application can then operate directly on the structured information in the document file. The contents of the structured non-JSON document are all contained in the ISO/IEC 646 characters that comprise the string of the `contents` field of the message.

```
{
  "header":{
    "messageID":"",
    "noun":"xyxy",
    "verb":"CREATE",
    "subject":"",
    "origin":"",
    "source":"",
    "destination":"",
```

```
    "timeStamp":"",
    "correlationID":"",
    "context":"",
    "user":"",
    "comment":"",
    "properties":"",
    "schemaVersion":"",
    "version":"",
    "replyAddress":"",
    "asyncReplyFlag":"",
    "ackRequired":"",
    "expiration":"",
    "hashType":"",
    "hash":""
  },
  "dataSet":{
    "dataElements":[
      {
        "xyxy":{
          "schema":"https://www.uudex.org/uudex/0.1/xyxy",
          "schemaVersion":"0.1",
          "encryption":"",
          "encoding":"",
          "compression":"",
          "properties":"",
          "format":"",
          "name":"",
          "tags":[
            "", ""
          ],
          "data":{
            "contents":""
          }
        }
      }
    ]
  }
}
```

7.3 Binary File Information

The least straightforward approach is to wrap an encoded and compressed version of a binary file in the UUDEX JSON message format, similar to how power system model files are exchanged in Section 6.5.

An example message for embedding a generic structured document follows. The `format` field specifies the native structured format, with the document name specified by the `name` field. Since there is no visible structure to the binary data, it must be compressed and encoded from binary to printable ISO/IEC 646 characters (rather than binary data), so the `encoding` and `compression` fields are required. The U-Publish client will read the contents specified in the `file` field, compress and encode it, and embed the resultant string in the `contents` field, while the UUDEX Subscribe client will extract the string from the `contents` field decode and decompress it, and create a file based on the specified `name` field. The native application can then operate directly on information in the document file.

The U-Message structure used would be essentially the same as the one discussed in Section 7.2.

This approach can be used to transfer any kind of file or format whether structured or not and can even transmit encrypted or password protected data since no interpretation of the binary format is performed. This could be used, for example, to transfer word processing files, image, or video files, or even database backups.

Additional fields could be defined to assist with the interpretation of the file. These fields might include the following:

- File type (as may relate to a specific application or defined schema)
- File name (need not be unique, may be hierarchical)
- File ID (unique key, e.g., a UUID)
- File source (organization that is owner or creator of the file)
- Created by (optional, person within the organization that created the file)
- File format (e.g., CSV, XML, JSON, PDF, text, image, binary)
- Schema reference (for structured documents, optional)
- File creation date (ISO 8601 timestamp, set by submitter)
- File submission date (ISO 8601 timestamp, set by UUDEX)
- File expiration date (after which file is no longer valid)
- Abstract (short description of file contents)
- Keywords (that may be useful for searches)
- Priority
- Status (default = ACTIVE)
- Version (default = 1)
- Obsoletes (optional, UUID of file version that this replaces)
- ObsoletedBy (optional, UUID of file version that this is replaced by).

8.0 References

- UUDEX Functional Requirements
- UUDEX Workflow design
- UUDEX Protocol design
- UUDEX Security design

- IEC 60870-6-802 (ICCP)
- IEC 60870-6-503 (ICCP)
- IEC 61968 family: (CIM)
  - IEC 61968-9 - Application integration at electric utilities – System interfaces for distribution management – Part 9: Interfaces for meter reading and control
  - IEC 61968-100
- IEC 61970 family (CIM)
  - IEC 61970-301
- IEC 62361-2 (??) - Power systems management and associated information exchange – Interoperability in the long term – Part 2: End to end quality codes for supervisory control and data acquisition (SCADA)

- Apache Avro™ Specification - https://avro.apache.org/docs/current/spec.html
- OPC
- STIX
- NVD

- IETF RFC 8259 (JSON) (https://tools.ietf.org/html/rfc8259) (also ECMA-404)
- IETF RFC 4648 (BASE64)
- IETF RFC 1924 (BASE85)
- IETF RFC 4122 (UUID) (which is technically equivalent to ITU-T Rec. X.667 and ISO/IEC 9834-8)
- IETF RFC 1630 (URI - Universal Resource Identifiers in WWW)
- IETF RFC 8820 (URI Design and Ownership)
- IETF RFC 1738 (Uniform Resource Locators (URL)) and updates
- IETF RFC 8785 (JSON Canonicalization Scheme (JCS))

- FIPS 140-2 Annex A (https://csrc.nist.gov/CSRC/media/Publications/fips/140/2/final/documents/fips1402annexa.pdf)
- DHS / US-CERT CISA TLP description available at https://us-cert.cisa.gov/sites/default/files/tlp/tlp-v1-letter.pdf

- ISO 8601-x:2019 Date and time — Representations for information interchange

Appendix A JSON and JSON Schema Description

(Informational)

The information in this appendix is derived from the w3schools.com tutorial on "What is JSON"[1], and the JSON Schema documentation available from the json-schema website[2]. The most current version of the JSON schema specifications as of the writing of this report is "draft 2019-09".

JSON is a flexible and lightweight method for storing and transmitting data and information. It is based on the format that JavaScript uses to store date, that is, but "name/value" pairs. The name/value pairs consist of a field name in a double-quoted string, and a value, separated by a colon (":"), such as:

```
"firstName":"John"
```

JSON may also contain objects, which are sets of name/value pairs separated by commas, and enclosed in curly braces:

```
{"firstName":"John","lastName":"Smith"}
```

The JSON object may have also have a name making the structure itself the value:

```
"name":{"firstName":"John","lastName":"Smith"}
```

JSON also allows for arrays of values, specified as JSON objects separated by commas and enclosed in square brackets:

```
"employees":[
  {"firstName":"John","lastName":"Smith"},
  {"firstName":"Anna","lastName":"Doe"},
  {"firstName":"Mary","lastName":"Jones"}
]
```

JSON Schemas are used to describe the named fields, their structure, and format, and whether the structures are required or optional.

For example, the JSON array structure named "employees" has the following JSON schema:

```
{
  "$schema":"http://json-schema.org/draft-04/schema#",
  "type":"object",
  "properties":{
    "employees":{
      "type":"array",
      "items":[
        {
          "type":"object",
          "properties":
```

[1] See https://www.w3schools.com/whatis/whatis_json.asp, accessed 09/14/2020
[2] See https://json-schema.org/, accessed 09/14/2020

```
            "firstName":{
              "type":"string"
            },
            "lastName":{
              "type":"string"
            }
          },
          "required":[
            "firstName",
            "lastName"
          ]
        }
      ]
    }
  },
  "required":[
    "employees"
  ]
}
```

The JSON schema shows that the “employees” structure is a required array comprised of the required objects firstName and lastName, both of which are strings.

JSON structures can be much more complex and can also describe optional objects. For a complete discussion of JSON schemas, refer to the latest document describing JSON Schemas available from the json-schema.org website. While the specification is still in draft form, it has been implemented by numerous tools and websites.

Appendix B Avro and Avro Schema Description

(Informational)

Avro is an Apache specification for transmitting and storing serialized data[1]. It provides for compression of the data during transmission and storage. It uses a data description, called a schema, to allow the transmission of the data with no additional overhead, while allowing for modifications to the structure to be automatically communicated from the sending application to the receiving application. When data is stored, the schema is also stored allowing the data to be retrieved even if the schema changes over time.

Avro schemas are described using JSON syntax

The Avro schema for the employees JSON array structure from Appendix A follows:

```
{
  "name":"MyClass",
  "type":"record",
  "namespace":"com.acme.avro",
  "fields":[
    {
      "name":"employees",
      "type":{
        "type":"array",
        "items":{
          "name":"employees_record",
          "type":"record",
          "fields":[
            {
              "name":"firstName",
              "type":"string"
            },
            {
              "name":"lastName",
              "type":"string"
            }
          ]
        }
      }
    }
  ]
}
```

[1] See http://avro.apache.org/ accessed 09/14/2020.

Appendix C – JSON and Avro Schemas for UUDEX Structures

(Normative)

To be supplied and finalized after demonstration project(s) are complete

C.1 Header Schema

```
{
  "$schema":"http://json-schema.org/draft-04/schema#",
  "type":"object",
  "properties":{
    "header":{
      "type":"object",
      "properties":{
        "messageID":{
          "type":"string"
        },
        "noun":{
          "type":"string"
        },
        "verb":{
          "type":"string"
        },
        "subject":{
          "type":"string"
        },
        "origin":{
          "type":"string"
        },
        "source":{
          "type":"string"
        },
        "destination":{
          "type":"string"
        },
        "timeStamp":{
          "type":"string"
        },
        "correlationID":{
          "type":"string"
        },
        "context":{
          "type":"string"
        },
        "user":{
          "type":"string"
        },
        "comment":{
          "type":"string"
        },
        "properties":{
          "type":"string"
        },
```

```
        "version":{
          "type":"string"
        },
        "replyAddress":{
          "type":"string"
        },
        "asyncReplyFlag":{
          "type":"string"
        },
        "ackRequired":{
          "type":"string"
        },
        "expiration":{
          "type":"string"
        },
        "hashType":{
          "type":"string"
        },
        "hash":{
          "type":"string"
        }
      },
      "required":[
        "messageID",
        "noun",
        "verb",
        "subject",
        "source",
        "destination",
        "timeStamp",
        "hashType",
        "hash"
      ]
      "$comment":"While not strictly required, version is strongly
recommended"
    }
  },
  "required":[
    "header"
  ]
}
```

C.2 Subject Create Schema

```
{
  "$schema":"http://json-schema.org/draft-04/schema#",
  "title":"JSON Schema for UUDEX Subject Policy",
  "$id":"https://www.uudex.org/uudex/0.1/SubjectPolicy",
  "definitions":{
    "def_permission_target_list":{
      "description":"An expression of parties allowed a given access right",
      "type":"object",
      "properties":{
        "allowOnly":{
          "description":"Contains a list of user identifiers that are to be
allowed access; all others denied",
```

```
          "type":"array",
          "items":{
            "type":"string"
          },
          "uniqueItems":true
        },
        "allowExcept":{
          "description":"Contains a list of user identifiers that are to be
denied access; all others allowed",
          "type":"array",
          "items":{
            "type":"string"
          },
          "uniqueItems":true
        },
        "allowAll":{
          "description":"Allow everyone this access",
          "type":"null"
        },
        "allowNone":{
          "description":"Allow no one this access",
          "type":"null"
        }
      },
      "additionalProperties":false,
      "maxProperties":1,
      "minProperties":1
    }
  },
  "type":"object",
  "properties":{
    "schemaVersion":{
      "description":"The version of the schema used to create this policy
record",
      "type":"string",
      "enum":[
        "https://www.uudex.org/uudex/0.1/SubjectPolicy"
      ]
    },
    "owner":{
      "description":"The UUDEX Participant identifier to whom this access
control applies",
      "type":"string"
    },
    "dataType":{
      "description":"The data type for the subject(s) for which this access
control applies",
      "type":"string"
    },
    "action":{
      "description":"Whether an attempt by the named participant to create or
change a UUDEX Subject for the given data type should be allowed, denied, or
subject to review.",
      "type":"string",
      "enum":[
        "ALLOW",
        "DENY",
```

```
        "REVIEW"
      ]
    },
    "constraints":{
      "description":"If allowed or subject to review, the constraints provide
additional limits during Subject creation or modification",
      "type":"object",
      "properties":{
        "maxQueueSizeKB":{
          "description":"The maximum allowable size of the UUDEX Subject's
queue in kB. The value of 0 is special and indicates explicitly that no
constraint is placed.",
          "type":"number",
          "minimum":0
        },
        "maxMessageCount":{
          "description":"The maximum number of allowed messages in a subject
at any one time. The value of 0 is special and indicates explicitly that no
constraint is placed.",
          "type":"number",
          "minimum":0
        },
        "fullQueueBehavior":{
          "description":"Constrain the subject's behavior when it has a full
queues. Options are to block new values or purge old values.",
          "type":"string",
          "enum":[
            "BLOCK_NEW",
            "PURGE_OLD",
            "NO_CONSTRAINT"
          ]
        },
        "maxPriority":{
          "description":"The maximum delivery priority that can be assigned
to this queue (noting that lower numbers are higher priorities. The value of
0 is special and indicates explicitly that no constraint is placed.",
          "type":"number",
          "minimum":0
        },
        "broadestAllowedPublisherAccess":{
          "description":"Only UUDEX Participants listed here are allowed to
publish to this subject (but the subject owner/manager could request fewer
than this list)",
          "$ref":"#/definitions/def_permission_target_list"
        },
        "broadestAllowedSubscriberAccess":{
          "description":"Only UUDEX Participants listed here are allowed to
subscribe to this subject (but the subject owner/manager could request fewer
than this list)",
          "$ref":"#/definitions/def_permission_target_list"
        },
        "broadestAllowedManagerAccess":{
          "description":"Only UUDEX Participants listed here are allowed to
manage this subject (but the subject owner/manager could request fewer than
this list)",
          "$ref":"#/definitions/def_permission_target_list"
        }
```

```
      }
    }
  },
  "required":[
    "schemaVersion",
    "action"
  ]
}
```

C.3 ACL Schema

```
{
  "$schema":"http://json-schema.org/draft-04/schema#",
  "title":"JSON Schema for UUDEX Subject ACLs",
  "$id":"https://www.uudex.org/uudex/0.1/SubjectACL",
  "definitions":{
    "def_permission_target_list":{
      "description":"An expression of parties allowed a given access right",
      "type":"object",
      "properties":{
        "allowOnly":{
          "description":"Contains a list of user identifiers that are to be
allowed access; all others denied",
          "type":"array",
          "items":{
            "type":"string"
          },
          "uniqueItems":true
        },
        "allowExcept":{
          "description":"Contains a list of user identifiers that are to be
denied access; all others allowed",
          "type":"array",
          "items":{
            "type":"string"
          },
          "uniqueItems":true
        },
        "allowAll":{
          "description":"Allow everyone this access",
          "type":"null"
        },
        "allowNone":{
          "description":"Allow no one this access",
          "type":"null"
        }
      },
      "additionalProperties":false,
      "maxProperties":1,
      "minProperties":1
    },
    "def_subject_identification":{
      "description":"Uniquely identifies a subject",
      "type":"object",
      "properties":{
        "owner":{
```

```
          "description":"The user identity of the party associated with the
subject",
          "type":"string"
        },
        "dataType":{
          "description":"The data type associated with this subject",
          "type":"string"
        },
        "groupKey":{
          "description":"The group key of this subject",
          "type":"string"
        }
      },
      "required":[
        "owner",
        "dataType",
        "groupKey"
      ]
    }
  },
  "type":"object",
  "properties":{
    "schemaVersion":{
      "description":"The version of the schema used to create this record",
      "type":"string",
      "enum":[
        "https://www.uudex.org/uudex/0.1/SubjectACL"
      ]
    },
    "subject":{
      "description":"The subject that is the target of this ACL",
      "$ref":"#/definitions/def_subject_identification"
    },
    "privilege":{
      "description":"The access privileges granted to this subject",
      "type":"object",
      "properties":{
        "publish":{
          "description":"Specify who is allowed to publish (write) to this
subject (owner and UUDEX Admin always implicitly allowed)",
          "$ref":"#/definitions/def_permission_target_list"
        },
        "subscribe":{
          "description":"Specify who is allowed to subscribe (read) to this
subject (owner and UUDEX Admin always implicitly allowed)",
          "$ref":"#/definitions/def_permission_target_list"
        },
        "manage":{
          "description":"Specify who is allowed to manage this subject (owner
and UUDEX Admin always implicitly allowed)",
          "$ref":"#/definitions/def_permission_target_list"
        },
        "discover":{
          "description":"Specify who is allowed to discover this subject (all
entities with other rights implicitly allowed, including the owner and UUDEX
Admin)",
          "$ref":"#/definitions/def_permission_target_list"
```

```
          }
        }
      }
    },
    "required":[
      "subject"
    ]
}
```

Appendix D Encoding and Compression

D.1 Encoding

(Informational)

JSON requires that all transfers be made using printable (e.g., ASCII(ISO/IEC 646) or UTF-8 (ISO 10646)) characters. This works well for text data, and individual values that can be represented as a string of numerals. However, many file formats that UUDEX will be transmitting (e.g., PDF documents, power system model files, software patches) are binary in nature, and must be converted from a binary representation to a character representation in order to be transferred. This problem was solved long ago when the Base64 encoding scheme was adopted, initially in the Multipurpose Internet Mail Extensions (MIME) format initially specified in RFC 2045, but more recently updated by RFC 2231 and RFC 6532.

The Base64 scheme represents binary data 6 bits at a time as a printable ISO/IEC 646 (ASCII) character. In order to transmit 8-bit bytes, the bytes must be combined and re-parsed as 6-bit chunks. This means that the 24 bits representing three binary characters is transformed into four characters each containing 6 bits of the original string, a 33% increase in the message size. Padding characters may be needed if the total message size is not a multiple of three. When re-grouping the bits into printable characters, the first 62 characters are specified as “A-Z”, “a-z”, and “0-9”. The remaining two characters vary from implementation to implementation, for example, MIME uses "+" and "/", while RFC 4648 uses "-" and "_". UUDEX specifies the RFC 4648 variant. Internet RFC 4648[1] provides more detail on the Base64 encoding algorithm.

An alternate but less popular encoding method called Base85 (also known as “Ascii85”) recognizes that there are more than 64 printable characters in the ISO/IEC 646 (ASCII) alphabet, and uses a similar combination and re-parsing process to turn four bytes of binary data into five ISO/IEC 646 (ASCII) characters for transmission for a 25% increase in message size (as opposed to the 33% increase when using Base64). As with Base64, padding characters may be needed if the total message size is not a multiple of four. The characters used in Base85 correspond to the ISO/IEC 646 (ASCII) characters “!” (33 or 0x21) through “u” (117 or 0x75) skipping the space character (32 or 0x20) and proceeding through for 86 consecutive ISO/IEC 646 (ASCII) printable characters. Base85 operates by taking the characters 32-bits at a time, convert it to a radix 85 format, and add 33 (0x21) to each element to arrive at the five Base85 characters. For example, as described by John D. Cook[2]:

> Suppose we start with the word 0x89255d9, equal to 143807961 in decimal.
>
> $$143807961 = 2\times85^4 + 64\times85^3 + 14\times85^2 + 18\times85 + 31$$
>
> and so, the radix 85 representation is (2, 64, 14, 18, 31). Adding 33 to each we find that the ISO/IEC 646 (ASCII) values of the characters in the Base85 representation are (35, 97, 47, 51, 64), or (“#”, “a”, “/”, “3”, “‘@”) and so ”#a/3@” is the Base85 encoding of 0x89255d9.

[1] See https://tools.ietf.org/html/rfc4648 accessed 10/01/2020
[2] See https://www.johndcook.com/blog/2019/03/05/base85-encoding/ (accessed 10/02/2020)

Base85 is used by Adobe in its Postscript and PDF formats, and git uses it to encode patches. Internet RFC 1924[1] provides more detail on the Base85 encoding algorithm.

Other encoding methods exist, but they are less efficient than Base64 and Base85, and are therefore not discussed.

D.2 Compression

Many files (whether binary or containing text) include repeated patterns, also referred to as "statistical redundancy". Using data compression algorithms, these repeated patterns can be compressed to reduce file sizes and correspondingly reduce both the number of bits in a message transmission and the length of time it takes to transmit the bits. A reverse process called decompression restores the file to its original format and size when received.

Data compression can be either "lossless' or "lossy". Lossy compression is useful where the fidelity of the resulting file is of less importance, for example, when transmitting a high-resolution photograph to a low-resolution display device. Once compressed using a lossy algorithm, the original high-fidelity picture cannot be restored.

However, lossless compression algorithms allow for the complete reconstruction of the contents of a compressed file resulting in a bit-for-bit identical result following the compression – decompression process. For these reasons, only lossless compression algorithms are supported by UUDEX. (If a user wishes to transmit a compressed photograph using a lossy compression, that compression must be performed outside of the UUDEX workflow.)

There are a great many lossless compression algorithms available. Selection of the correct algorithm requires an analysis of the tradeoff between compression speed and required memory resources to the resultant file size. Often, compression algorithms that result in the smallest file sizes require the most compute and memory resources, while algorithms that are fast and compact produce files with less compression. Algorithm selection may also impact the resultant file sizes – in some cases the "compressed" file is larger than the original if it contains bit patterns that do not effectively compress, for example, if the file has already been compressed.

Many compression algorithms are based on work done by Abraham Lempel and Jacob Ziv in 1977[2].

Some algorithms focus on either compression or decompression speed, or resultant file size. Since data in UUDEX will be generally compressed and decompressed the same number of times, an algorithm that favors one over the other may not be appropriate. (However, many algorithms naturally decompress faster than they compress since less data analysis is involved in the decompression process.)

Some algorithms are more efficient on different data types or patterns. For example, files containing long strings of the same value (e.g., patterns like "aaaabbbbccccdddd") may compress very efficiently using a run-length encoding compression, while files containing

---

[1] See https://tools.ietf.org/html/rfc1924 accessed 10/01/2020

[2] Ziv, Jacob; Lempel, Abraham (May 1977). "A Universal Algorithm for Sequential Data Compression". IEEE Transactions on Information Theory. 23 (3): 337–343. CiteSeerX 10.1.1.118.8921. doi:10.1109/TIT.1977.1055714.

repeated patterns of complex strings or values (e.g., patterns like “abcdabcdabcd”) compress better using a Lempel-Ziv based algorithm like LZ77.

Many compression algorithms provide for specifying a compression level that is used by the algorithm to trade off performance for resulting file size, with the understanding that additional processing time and other compute resources may be required to produce smaller resulting file sizes. Decompression processing time is generally independent of the compression level.

UUDEX provides for specifying different compression algorithms due to the differing performance and resulting compression rations available for different kinds of data.

UUDEX requires support for “GZIP” and “ZLIB” to specify either the gzip compression program, or the zlib library, which is the underlying code used by gzip. The zlib code uses a modified DEFLATE algorithm, that is itself a combination of LZ77 and Huffman coding. Internet RFC 6713[1] provides additional detail on the gzip program.

Other potential compression algorithms include the following:

- BZIP2 – an open source program developed by Julian Seward and maintained by Federico Mena that uses the Burrows–Wheeler algorithm[2]. It is generally more efficient than previous algorithms like DEFLATE but is slower.
- DEFLATE – a compression algorithm developed by Phil Katz for use in Version 2 of the PKZIP program, that uses a combination of LZSS and Huffman coding. It is documented in RFC 1951[3]
- LZ4 – a compression algorithm that is focused on compression and decompression speed. It is based on LZ77.
- LZ77 – an early compression algorithm developed in 1977 by Abraham Lempel and Jacob Ziv[4]
- LZ78 – a follow on to the LZ77 algorithm developed in 1978 by Abraham Lempel and Jacob Ziv[5]
- LZMA – the Lempel–Ziv–Markov chain algorithm used primarily in the “7z” file format of the 7-zip program. It is based on the LZ77 algorithm with a variable length compression dictionary
- LZMA2 – a container format that can include both compressed data using LZMA and uncompressed data.

---

[1] See https://tools.ietf.org/html/rfc6713 (accessed 10/01/2020)
[2] See http://www.bzip.org/ (accessed 10/02/2020) and https://www.ncbi.nlm.nih.gov/IEB/ToolBox/CPP_DOC/lxr/source/src/util/compress/bzip2/README (accessed (10/02/2020)
[3] See https://tools.ietf.org/html/rfc1951 (accessed 10/01/2020)
[4] Ziv, Jacob; Lempel, Abraham (May 1977). "A Universal Algorithm for Sequential Data Compression". IEEE Transactions on Information Theory. 23 (3): 337–343. CiteSeerX 10.1.1.118.8921. doi:10.1109/TIT.1977.1055714.
[5] Ziv, Jacob; Lempel, Abraham (September 1978). "Compression of Individual Sequences via Variable-Rate Coding". IEEE Transactions on Information Theory. 24 (5): 530–536. CiteSeerX 10.1.1.14.2892. doi:10.1109/TIT.1978.1055934.

- LZO – a compression algorithm that is focused on decompression speed. It was developed by Markus Franz Xaver Johannes Oberhumer, based on earlier algorithms by Abraham Lempel and Jacob Ziv. It was originally published in 1996, and the code is copyrighted by Oberhumer, but available from his website[1].
- LZSS – a derivative of LZ77 created by James A. Storer and Thomas Szymanski[2]. It differs from LZ77 in that in certain cases, the replacement string for a compressed sequence could be longer than the sequence it replaces. LZSS does not perform the replacement in this case.
- LZFSE – the Lempel–Ziv Finite State Entropy algorithm[3] introduced by Apple first released in IOS 9. Apple claims that LZFSE is as efficient as zlib for compression, but decompression is 2-3 times faster using fewer resources.
- LZW – a compression algorithm developed by Abraham Lempel, Jacob Ziv, and Terry Welch published in 1984[4] as an improvement to the LZ78 algorithm. It was patented in 1983[5]
- PPM – the prediction by partial matching compression algorithm
- RLE – a compression technique that compresses repeated sequences of characters as a single data value and count, rather than as the original sequence of repeated characters.

A note on compression and data formats: A sample OE-417 PDF report used for analysis is 274 kB in size. The XML data extracted from the report is 4 kB. The PDF report when compressed using gzip shrinks to 137 kB. When the binary file is encoded using BASE64, the file expands to 183 kB, while if only expands to 173 kB using BASE85 encoding.

---

[1] See http://www.oberhumer.com/opensource/lzo/ (accessed 10/01/2020)
[2] Storer, James A.; Szymanski, Thomas G. (October 1982). "Data Compression via Textual Substitution". *Journal of the ACM*. **29** (4): 928–951. doi:10.1145/322344.322346.
[3] Bainville, Eric (2016-06-07). "LZFSE compression library and command line tool". GitHub. Retrieved 2016-07-04.
[4] Welch, Terry (1984). "A Technique for High-Performance Data Compression" (PDF). Computer. 17 (6): 8–19. doi:10.1109/MC.1984.1659158.1983
[5] See https://patents.google.com/patent/US4558302 accessed 10/01/2020

Appendix E Example Information

E.1 Threat Report

(informational)

This section to be supplied and finalized after demonstration efforts are completed.

This example packages the “Identifying a Threat Actor Profile” example STIX report from the OASIS Open website located at https://oasis-open.github.io/cti-documentation/stix/examples.html.


```
{
  "header":{
    "messageID":"d5d1c892-974a-11e9-b198-b0c090affff",
    "noun":"STIX-Report",
    "subject":"EISAC/STIX/Report",
    "origin":"Utility1",
    "source":"Utility1",
    "destination":"E-ISAC",
    "timeStamp":"2020-05-13 10:12:09.209124",
    "verb":"created"
  },
  "payload":{
    "STIX-Report":{
      "type":"bundle",
      "id":"bundle--601cee35-6b16-4e68-a3e7-9ec7d755b4c3",
      "objects":[
        {
          "type":"threat-actor",
          "spec_version":"2.1",
          "id":"threat-actor--dfaa8d77-07e2-4e28-b2c8-92e9f7b04428",
          "created":"2014-11-19T23:39:03.893Z",
          "modified":"2014-11-19T23:39:03.893Z",
          "name":"Disco Team Threat Actor Group",
          "description":"This organized threat actor group operates to create
profit from all types of crime.",
          "threat_actor_types":[
            "crime-syndicate"
          ],
          "aliases":[
            "Equipo del Discoteca"
          ],
          "roles":[
            "agent"
          ],
          "goals":[
            "Steal Credit Card Information"
          ],
          "sophistication":"expert",
          "resource_level":"organization",
          "primary_motivation":"personal-gain"
        },
```

```
        {
          "type":"identity",
          "spec_version":"2.1",
          "id":"identity--733c5838-34d9-4fbf-949c-62aba761184c",
          "created":"2016-08-23T18:05:49.307Z",
          "modified":"2016-08-23T18:05:49.307Z",
          "name":"Disco Team",
          "description":"Disco Team is the name of an organized threat actor
crime-syndicate.",
          "identity_class":"organization",
          "contact_information":"disco-team@stealthemail.com"
        },
        {
          "type":"relationship",
          "spec_version":"2.1",
          "id":"relationship--a2e3efb5-351d-4d46-97a0-6897ee7c77a0",
          "created":"2020-02-29T18:01:28.577Z",
          "modified":"2020-02-29T18:01:28.577Z",
          "relationship_type":"attributed-to",
          "source_ref":"threat-actor--dfaa8d77-07e2-4e28-b2c8-92e9f7b04428",
          "target_ref":"identity--733c5838-34d9-4fbf-949c-62aba761184c"
        }
      ]
    }
  }
}
```


E.2 Measured Value Information Exchange

This example shows how information previously exchanged using ICCP is exchanged using UUDEX. The example shows information originating from and transmitted by Utility1 to its reliability coordinator, RC1.


```
{
  "header":{
    "messageID":"d5d1c892-974a-11e9-b198-b0c090affff",
    "noun":"ICCP-Block-1",
    "origin":"Utility1",
    "source":"Utility1",
    "destination":"RC1",
    "timeStamp":"2020-05-13 10:12:09.209124",
    "verb":"created"
  },
  "payload":{
    "ICCP-Block-1":{




    }
  }
)
```

E.3 OE-417 Report Information (XML)

This example shows how the XML information component of the DOE OE-417 report is encapsulated inside of a UUDEX JSON message.


```
{
  "header":{
    "messageID":"d5d1c892-974a-11e9-b198-b0c090affff",
    "noun":"OE-417",
    "verb":"CREATE",
    "subject":"DOE/OE-417/XML",
    "origin":" Utility1",
    "source":" Utility1",
    "destination":"RC1",
    "timeStamp":"2020-05-13 10:12:09.209124",
    "hashType":"",
    "hash":""
  },
  "dataSet":{
    "dataElements":[
      {
        "OE-417":{
          "schema":"https://www.uudex.org/uudex/0.1/OE-417",
          "schemaVersion":"0.1",
          "format":"XML",
          "name":"OE-417-Example.xml",
          "encoding":"ASCII",
          "compression":"NONE",
          "contents":"<?xml version="1.0" encoding="UTF-8"?><topmostSubform><
cbincidenttype1>Off</cbincidenttype1><cbincidenttype2>Off</cbincidenttype2><c
bincidenttype3>Off</cbincidenttype3><cbincidenttype4>Off</cbincidenttype4><cb
incidenttype5>Off</cbincidenttype5><cbincidenttype6>
   .
   .
   .
>jdoe@electric-power-inc-fake.com</tbEmail><tbNarrative>Gunshots targeting Su
b1 Xfr1</tbNarrative><tbEstRestMonth/><tbEstRestDay/><tbEstRestYear/><tbGener
ator>Sub1 Xfr1</tbGenerator><cbNotifyNERC>Yes</cbNotifyNERC><cbNotifyEISAC>Ye
s</cbNotifyEISAC><txtTimeStamp4/></topmostSubform>"
        }
      }
    ]
  }
}
```


E.4 OE-417 Report Information (PDF)

This example shows how the same OE-417 report transmitted as an XML document in section E.3 can be transmitted as a native PDF file.


```
{
  "header":{
    "messageID":"d5d1c892-974a-11e9-b198-b0c090affff",
    "noun":"OE-417",
    "verb":"CREATE",
```

```
    "subject":"DOE/OE-417/PDF",
    "origin":" Utility1",
    "source":" Utility1",
    "destination":"RC1",
    "timeStamp":"2020-05-13 10:12:09.209124",
    "hashType":"",
    "hash":""
  },
  "dataSet":{
    "dataElements":[
      {
        "OE-417":{
          "schema":"https://www.uudex.org/uudex/0.1/OE-417",
          "schemaVersion":" 0.1",
          "format":"PDF",
          "name":"OE-417-Example.pdf",
          "encoding":"BASE85",
          "compression":"GZIP",
          "contents":" <~+,^C1X92U4!!?"c1bgpr7W3<a?SF>n0etF<0d&%j;c6=A0etO;E+
EP\ls"`p<:0Q)4+f*+XsPgY;!+9(H\ngIqbd_Tb`$D_0Jc*pG*OtkQBr@d8r:Vep+7a/ZDK2#/lDQ
shL79E>@$.#fB&(-^YbmE\Lo\aIElp%oA'*=H(N?1TBB%Y2De_BC>hQ\LPFR;7H#@)O$s$S]"=$%[
PlRhRmn_[]Q]N1#S:^1J8bVm(5&t\J62l/1d[j3-p3$e7%f\9S9#g+8"(Qj&6Lb1!MVidMiNfGR7?
3"6@Aa9mt!h.?ghDop7RVjejP67P:[[!WI]LT-<m<(K1tk0@*oht&CSrl)LGCcA-UJL8n=(I/jYpE
3b]Pu!<rNF!mi>(('gN*n-4t<)N?L-5[eW)4r2YqL-PfRN
    .
    .
    .
Scg>986jY"."s/jbpYb<Jq'?um'RKc#jPbs+%k8^fH4O)Y\nE?c32OBX!Rh\GNJ'IT$RVheB/KGk_
V0$9m5X<qG6r.HWT&FUgW'3h87JFR&e$h*q_].MU1Or_7V`<\,e/aF/.\bEN9m)$TdmOhnR':EZ_j
f+6TZb>TYgjOs?-hS[lLEBHt.oH;7L'mU68lAE3cjt"J=7BJZqdD!l/?Ku47Vh8KE*Vq1lQ-?E5+O
!RKsOdMnDpbG^k7E3k5FU`RACJ`GHS:A9c#A^4Y$qYSHQZhorI:hhd;eZ^_*J/:@J"'^jA-OHs!O'
"]'!"cq8K?L3JS7K6s'9F&W1C+f[.Gr!2EDfb`5`F3Vr(#J,/pT&D<`++HGc'cPUD?Spp86Qne_Kp
bI2`FNt1dW]YNF"fL<63)ShOUroi7HMIL&H32E-Ko?<70ES~>"
        }
      }
    ]
  }
}
```


E.5 Physical Security Incident Report (PSIR)

This example shows a Physical Security Incident Report that is formatted as a JSON document and transmitted to the E-ISAC. Note that this report only employs a small subset of the overall fields. It is likely that most reports would only make use of some of the available fields based on the nature of the incident, the information available when the report was written, and the time and resources available to the report author.


```
{
  "header":{
    "messageID":"3a645f92-76d2-4db8-9b05-e11d651560bc",
    "noun":"physicalSecurityIncident",
    "verb":"CREATE",
    "subject":"EISAC/PSIR/initial",
    "origin":"TOP1",
    "source":"TOP1",
    "destination":"E-ISAC",
```

```
    "timeStamp":"2020-09-23 10:12:09.209124",
    "user":"",
    "hashType":"SHA-256",
    "hash":"74CA5D8491283401222901E03D85BB15A6769D29FDB239C2C0DED69D89D3B8E6"
  },
  "dataSet":{
    "dataElements":[
      {
        "physicalSecurityIncident":{
          "reportHeader":{
            "uniqueID":"e523a2b3-3124-47c9-ad42-eb006f7e734f",
            "schema":"https://www.uudex.org/uudex/0.1/PhysicalSecurityIncident
tReport",
            "schemaVersion":"0.1",
            "title":"Drone surveillance at Friendly, Minnesota substation",
            "summary":"Two drones were observed overflying a substation in Fr
iendly, Minnesota for a period of 15 minutes. They were observed making coord
inated passes over the site from multiple angles.",
            "reportStatus":"FINAL",
            "TLP":"WHITE",
            "filedBy":{
              "name":"John Smith",
              "organization":"Friendly Power and Light",
              "phone":"123-456-7890",
              "email":"jsmith@fpandl.com"
            },
            "timeOfReport":"2020-08-24T20:49:18+0000"
          },
          "incidentSeverity":"NUISANCE",
          "incidentCategory":"SURVEILLANCE",
          "timeOfDetection":"2020-08-24T20:49:18+0000",
          "targetList":[
            {
              "targetIdentification":"Friendly Minnesota substation #8694",
              "targetDescription":"Local substation",
              "targetOwnerName":"Friendly Power and Light",
              "targetOperatorName":"Friendly Power and Light",
              "targetLocation":{
                "country":"USA",
                "state":"Minnesota",
                "code":"55432",
                "address":"1234 East 5th Ave SW"
              }
            }
          ],
          "incidentImpact":{
            "functionalImpact":{
              "severity":"NUISANCE"
            },
            "riskImpact":{
              "severity":"MINOR"
            }
          },
          "incidentResponse":{
            "responseSummary":"Notification of law enforcement.",
            "responderNotification":[
              {
```

```
                "contact":{
                  "name":"Officer Jane Smith",
                  "organization":"Friendly Police Department",
                  "phone":"123-555-3213"
                },
                "timeContacted":"2020-08-24T20:49:18+0000",
                "additionalContactInformation":"Officer arrived at the scene
20 minutes after call. By this time the incident had ended. Police issued a b
ulletin to be on the lookout for the van associated with the incident."
              }
            ]
          },
          "attackCharacterization":{
            "attackIntent":{
              "attackIntentSpying":true,
              "attackIntentTactical":true
            },
            "attackMeans":[
              {
                "attackMeansType":"DRONE",
                "attackMeansDescription":"Two small quadcopter drones, flying
in formation"
              }
            ]
          },
          "indicators":{
            "personIndicators":[
              {
                "apparentHeight":{
                  "measure":69,
                  "units":"INCHES"
                },
                "apparentWeight":{
                  "measure":180,
                  "units":"POUNDS"
                },
                "apparentGender":"MALE",
                "otherKeyObservableFeatures":"Figure was seen at a distance v
ia a security camera. Information provided is a best estimate"
              }
            ],
            "vehicleIndicators":[
              {
                "vehicleType":"DRONE",
                "vehicleColor":"black",
                "otherVehicleCharacteristics":"Two small, quadcopter drones o
f unknown make. They appear to be about 6 inches in length. Both drones moved
in tandem suggesting their behavior was based on a single controller"
              },
              {
                "vehicleType":"TRUCK",
                "vehicleColor":"white",
                "otherVehicleCharacteristics":"Large van of a model used for
service delivery (no passenger windows). Rear bumper appeared to be askew."
              }
            ]
          },
```

```
          "recommendedCourseOfAction":{
            "recommendation":"Be on the lookout for a white delivery van in t
he vicinity of sensitive infrastructure. Alert law enforcement if this happen
s."
          }
        }
      }
    ]
  }
}
```


This report describes a hypothetical incident involving surveillance of a substation using drones.

The header section provides an identifier, schema version, title, and summary. It notes that the report is final and is releasable as TLP "white". It also notes who submitted the report and when.

The subsequent fields provide a high-level classification of the incident. They note that the incident is only of "nuisance" severity, that it was a "surveillance" incident, and the time at which the incident was detected. (The time event began is not given, presumably because it is not known.)

The target list identifies only a single target – a power substation. The section identifies the target, who owns and operates it, and where it is located.

The impact section notes that there was negligible functional impact but noted the possibility of minor risk increase due to the surveillance activities. No other fields are employed - given the limited impact of the event, no OE-417 impacts would be applicable, no other parties were impacted, and there was no financial impact.

The incident response section notes that law enforcement was contacted and identifies the relevant contact and when engagement occurred.

The attack characterization section noted that the goals of the incident were likely to be surveillance, but that the action might also be "tactical" as a preparation for future actions. The section notes that drones were utilized in the attack.

The indicators section has multiple entries. It describes a person who was seen during the incident and provides a rough description. The section also describes both the drones and a white van that was seen at the time of the incident and was suspected to be involved.

The recommended course of action section recommends that peers be on the lookout for the white van near sensitive sites, but otherwise requests no action or information from them.

Appendix F OE-417 XML Schema

(Informational)

The PDF report format containing an OE-417 report contains information described by an XML schema. The information in an OE-417 PDF report can thus be extracted from the PDF report for transmission and processing without requiring the entire PDF report be transferred. Similarly, the information can be imported into the form to produce a printable version of the original form.

The XML schema used is:

```
<xs:schema attributeFormDefault="unqualified"
           elementFormDefault="qualified"
           xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="topmostSubform">
    <xs:complexType>
      <xs:sequence>

        <!-- From OE-417 form expiring on 05/31/2021 -->
        <!-- https://www.oe.netl.doe.gov/docs/OE417_Form_05312021.pdf -->

        <!-- Schedule 1 -->
        <!-- Alert Criteria -->

        <!-- Form Completion date and time (form header) -->
        <xs:element type="xs:string" name="txtTimeStamp1"/>

        <!-- Incident type -->

        <!-- Emergency Alert -->

        <!-- Physical attack that causes major interruptions or impacts -->
        <!-- to critical infrastructure facilities or to operations -->
        <!-- checkbox: "Off" or "Yes" -->
        <xs:element type="xs:string" name="cbincidenttype1"/>

        <!-- Cyber event that causes interruptions of electrical system -->
        <!-- operations -->
        <!-- checkbox: "Off" or "Yes" -->
        <xs:element type="xs:string" name="cbincidenttype2"/>

        <!-- Complete operational failure or shut-down of the -->
        <!-- transmission and/or distribution electrical system -->
        <!-- checkbox: "Off" or "Yes" -->
        <xs:element type="xs:string" name="cbincidenttype3"/>

        <!-- Electrical System Separation (Islanding) where part or -->
        <!-- parts of a power grid remain(s) operational in an -->
        <!-- otherwise blacked out area or within the partial -->
        <!-- failure of an integrated electrical system -->
        <!-- checkbox: "Off" or "Yes" -->
        <xs:element type="xs:string" name="cbincidenttype4"/>
```

```
<!-- Uncontrolled loss of 300 Megawatts or more of firm system -->
<!-- loads for 15 minutes or more from a single incident -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbincidenttype5"/>

<!-- Firm load shedding of 100 Megawatts or more implemented --?
<!-- under emergency operational policy -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbincidenttype6"/>

<!-- System-wide voltage reductions of 3 percent or more -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbincidenttype7"/>

<!-- Normal Report -->

<!-- Public appeal to reduce the use of electricity for -->
<!-- purposes of maintaining the continuity of the Bulk -->
<!-- Electric System -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbincidenttype8"/>

<!-- Physical attack that could potentially impact electric -->
<!-- power system adequacy or reliability; or vandalism which -->
<!-- targets components of any security systems -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbincidenttype9"/>

<!-- Cyber event that could potentially impact electric power -->
<!-- system adequacy or -->
<!-- reliability -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbincidenttype10"/>

<!-- Loss of electric service to more than 50,000 customers -->
<!-- for 1 hour or more -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbincidenttype11"/>

<!-- Fuel supply emergencies that could impact electric power-->
<!-- system adequacy or -->
<!-- reliability -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbincidenttype12"/>

<!-- System Report -->

<!-- Damage or destruction of a Facility within its -->
<!-- Reliability Coordinator Area, Balancing Authority Area -->
<!-- or Transmission Operator Area that results in action(s) -->
<!-- to avoid a Bulk Electric System Emergency. -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbincidenttype13"/>

<!-- Damage or destruction of its Facility that results from -->
<!-- actual or suspected intentional human action. -->
<!-- checkbox: "Off" or "Yes" -->
```

```
<xs:element type="xs:string" name="cbincidenttype14"/>

<!-- Physical threat to its Facility excluding weather or -->
<!-- natural disaster related threats, which has the potential -->
<!-- to degrade the normal operation of the Facility. Or -->
<!-- suspicious device or activity at its Facility. -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbincidenttype15"/>

<!-- Physical threat to its Bulk Electric System control -->
<!-- center, excluding weather or natural disaster related -->
<!-- threats, which has the potential to degrade the -->
<!-- normal operation of the control center. -->
<!-- Or suspicious device or activity at its Bulk Electric -->
<!-- System control center. -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbincidenttype16"/>

<!-- Bulk Electric System Emergency resulting in voltage -->
<!-- deviation on a Facility; A voltage deviation equal to -->
<!-- or greater than 10% of nominal voltage sustained -->
<!-- for greater than or equal to 15 continuous minutes. -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbincidenttype17"/>

<!-- Uncontrolled loss of 200 Megawatts or more of firm -->
<!-- system loads for 15 minutes or more from a single -->
<!-- incident for entities with previous year's peak demand -->
<!-- less than or equal to 3,000 Megawatts -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbincidenttype18"/>

<!-- Total generation loss, within one minute of: greater -->
<!-- than or equal to 2,000 Megawatts in the Eastern or -->
<!-- Western Interconnection or greater than or equal -->
<!-- to 1,400 Megawatts in the ERCOT Interconnection. -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbincidenttype19"/>

<!-- Complete loss of off-site power (LOOP) affecting a -->
<!-- nuclear generating station per the Nuclear Plant -->
<!-- Interface Requirements. -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbincidenttype20"/>

<!-- Unexpected Transmission loss within its area, contrary -->
<!-- to design, of three or more Bulk Electric System ->
<!-- Facilities caused by a common disturbance (excluding -->
<!-- successful automatic reclosing). -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbincidenttype21"/>

<!-- Unplanned evacuation from its Bulk Electric System -->
<!-- control center facility for 30 continuous minutes or more. -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbincidenttype22"/>
```

```
<!-- Complete loss of Interpersonal Communication and -->
<!-- Alternative Interpersonal Communication capability -->
<!-- affecting its staffed Bulk Electric System control -->
<!-- center for 30 continuous minutes or more. -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbincidenttype23"/>

<!-- Complete loss of monitoring or control capability at -->
<!-- its staffed Bulk Electric System control center for 30 -->
<!-- continuous minutes or more. -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbincidenttype24"/>


<!-- A. Alert Status-->

<!-- Emergency Alert - 1 hour -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbalert1"/>

<!-- Normal Report - 6 hours -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbalert2"/>

<!-- System Report 1 - Business Day -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbalert3"/>

<!-- Update - As required -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbalert4"/>

<!-- Final - 72 hours -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbalert5"/>


<!-- B. Organization Name -->
<!-- Text -->
<xs:element type="xs:string" name="tbOrgName"/>


<!-- C. Address of Principal Business Office -->
<!-- Text -->
<xs:element type="xs:string" name="tbOrgAddress"/>

<!-- Incident and Disturbance Data -->

<!-- Form Completion date and time (form header) -->
<xs:element type="xs:string" name="txtTimeStamp2"/>


<!-- D. Geographic Area(s) Affected (County, State) -->
<!-- Text -->
<xs:element type="xs:string" name="tbGeoRegions"/>

<!-- E. Date/Time Incident Began (mm-dd-yy/hh:mm) using -->
```

```
<!-- 24-hour clock -->
<!-- Date - month -->
<xs:element type="xs:string" name="tbBeginMonth"/>

<!-- Incident Began -->
<!-- Date - day -->
<xs:element type="xs:string" name="tbBeginDay"/>

<!-- Incident Began -->
<!-- Date - year -->
<xs:element type="xs:string" name="tbBeginYear"/>

<!-- Incident Began -->
<!-- Time - hour -->
<xs:element type="xs:string" name="tbBeginHour"/>

<!-- Incident Began -->
<!-- Time - minute-->
<xs:element type="xs:string" name="tbBeginMin"/>

<!-- Incident Began Timezone Eastern -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="chkBeginEastern"/>

<!-- Incident Began Timezone Central -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="chkBeginCentral"/>

<!-- Incident Began Timezone Mountain -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="chkBeginMountain"/>

<!-- Incident Began Timezone Pacific -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="chkBeginPacific"/>

<!-- Incident Began Timezone Alaska -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="chkBeginAlaska"/>

<!-- Incident Began Timezone Hawaii -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="chkBeginHawaii"/>


<!-- F. Date/Time Incident Ended (mm-dd-yy/hh:mm) using -->
<!-- 24-hour clock -->
<!-- Date - month -->
<xs:element type="xs:string" name="tbEndMonth"/>

<!-- Incident Ended -->
<!-- Date - day -->
<xs:element type="xs:string" name="tbEndDay"/>

<!-- Incident Ended -->
<!-- Date - year -->
<xs:element type="xs:string" name="tbEndYear"/>
```

```
<!-- Incident Ended -->
<!-- Time - hour -->
<xs:element type="xs:string" name="tbEndHour"/>

<!-- Incident Ended -->
<!-- Time - minute -->
<xs:element type="xs:string" name="tbEndMin"/>

<!-- Incident Ended timezone Eastern -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="chkEndEastern"/>

<!-- Incident Ended timezone Central -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="chkEndCentral"/>

<!-- Incident Ended timezone Mountain -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="chkEndMountain"/>

<!-- Incident Ended timezone Pacific -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="chkEndPacific"/>

<!-- Incident Ended timezone Alaska -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="chkEndAlaska"/>

<!-- Incident Ended timezone Hawaii -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="chkEndHawaii"/>


<!-- G. Did the incident/disturbance originate in your -->
<!-- system/area? -->

<!-- (check one) -->

<!-- Did originate in my area -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbOriginatedYes"/>

<!-- Did not originate in my area -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbOriginatedNo"/>

<!-- Unknown Origination -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbOriginatedUnk"/>


<!-- H. Estimate of Amount of Demand Involved (Peak Megawatts) -->
<xs:element type="xs:string" name="tbPeak"/>

<!-- check if no demand involved -->
<!-- checkbox: "Off" or "Yes" -->
```

```
<xs:element type="xs:string" name="cbPeakZero"/>

<!-- check if demand unknown -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbPeakUnk"/>


<!--I.  Estimate of Number of Customers Affected -->
<xs:element type="xs:string" name="tbCust"/>

<!-- check if zero customers impacted -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbCustZero"/>

<!-- check if customer count is Unknown -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbCustUnk"/>


<!-- Type of Emergency -->


<!-- J - Cause -->

<!-- Unknown -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbCause1"/>

<!-- Physical attack -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbCause2"/>

<!-- Threat of physical attack -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbCause3"/>

<!-- Vandalism -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbCause4"/>

<!-- Theft -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbCause5"/>

<!-- Suspicious activity -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbCause6"/>

<!-- Cyber event (information technology) -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbCause7"/>

<!-- Cyber event (operational technology) -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbCause8"/>

<!-- Fuel supply emergencies, interruption, or deficiency -->
```

```
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbCause9"/>

<!-- Generator loss or failure not due to fuel supply -->
<!-- interruption or deficiency or transmission failure -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbCause10"/>

<!-- Transmission equipment failure (not including substation -->
<!-- or switchyard) -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbCause11"/>

<!-- Failure at high voltage substation or switchyard -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbCause12"/>

<!-- Weather or natural disaster -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbCause13"/>

<!-- Operator action(s) -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbCause14"/>

<!-- Other -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbCause15"/>

<!-- Additional Information/Comments supplied: -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbCause16"/>

<!-- Additional Information/Comments (text): -->
<!-- Text -->
<xs:element type="xs:string" name="tbCauseOther"/>


<!-- K. Impact -->

<!-- None -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbImpact1"/>

<!-- Control center loss, failure, or evacuation -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbImpact2"/>

<!-- Loss or degradation of control center monitoring or -->
<!-- communication systems -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbImpact3"/>

<!-- Damage or destruction of a facility -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbImpact4"/>
```

```
<!-- Electrical system separation (islanding) -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbImpact5"/>

<!-- Complete operational failure or shutdown of the -->
<!-- transmission and/or distribution system -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbImpact6"/>

<!-- Major transmission system interruption (three or more -->
<!-- BES elements) -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbImpact7"/>

<!-- Major distribution system interruption -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbImpact8"/>

<!-- Uncontrolled loss of 200 MW or more of firm system -->
<!-- loads for 15 minutes or more -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbImpact9"/>

<!-- Loss of electric service to more than 50,000 customers -->
<!-- for 1 hour or more -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbImpact10"/>

<!-- System-wide voltage reductions or 3 percent or more -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbImpact11"/>

<!-- Voltage deviation on an individual facility of =10% -->
<!-- for 15 minutes or more -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbImpact12"/>

<!-- Inadequate electric resources to serve load -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbImpact13"/>

<!-- Generating capacity loss of 1,400 MW or more -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbImpact14"/>

<!-- Generating capacity loss of 2,000 MW or more -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbImpact15"/>

<!-- Complete loss of off-site power to a nuclear generating -->
<!-- station -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbImpact16"/>

<!-- Other -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbImpact17"/>
```

```
<!-- Additional Information/Comments: -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbImpact18"/>

<!-- Additional Information/Comments (text): -->
<!-- Text -->
<xs:element type="xs:string" name="tbImpactOther"/>


<!-- L. Action Taken -->

<!-- None -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbAction9"/>

<!-- Shed Firm Load: Load shedding of 100 MW or more -->
<!-- implemented under emergency operational policy (manually -->
<!-- or automatically via UFLS or remedial action scheme) -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbAction1"/>

<!-- Public appeal to reduce the use of electricity for the -->
<!-- purpose of maintaining the continuity of the electric -->
<!-- power system -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbAction10"/>

<!-- Implemented a warning, alert, or contingency plan -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbAction11"/>

<!-- Voltage reduction -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbAction12"/>

<!-- Shed Interruptible Load -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbAction5"/>

<!-- Repaired or restored -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbAction13"/>

<!-- Mitigation implemented -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbAction14"/>

<!-- Other -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbAction8"/>

<!-- Additional Information/Comments -->
<!-- checkbox: "Off" or "Yes" -->
<xs:element type="xs:string" name="cbAction99"/>

<!-- Additional Information/Comments (text) -->
```

```
<xs:element type="xs:string" name="tbActionOther"/>

<!-- Form Completion date and time (form header) -->
<xs:element type="xs:string" name="txtTimeStamp3"/>

<!-- Schedule 2 -->
<!-- Narrative Description -->

<!-- Name of official that should be contacted for follow-up -->
<!-- or any additional information -->


<!-- M. Name -->
<!-- Text -->
<xs:element type="xs:string" name="tbName"/>


<!-- N. Title -->
<!-- Text -->
<xs:element type="xs:string" name="tbTitle"/>


<!-- O. Telephone Number -->
<!-- Phone - area code -->
<xs:element type="xs:string" name="tbPhone1"/>

<!-- Phone - exchange -->
<xs:element type="xs:string" name="tbPhone2"/>

<!-- Phone - number -->
<xs:element type="xs:string" name="tbphone3"/>


<!-- P. Fax Number -->
<!-- Phone - area code -->
<xs:element type="xs:string" name="tbFax1"/>

<!-- Phone - exchange -->
<xs:element type="xs:string" name="tbFax2"/>

<!-- Phone - number -->
<xs:element type="xs:string" name="tbFax3"/>


<!-- Q. E-mail Address -->
<!-- email address-->
<xs:element type="xs:string" name="tbEmail"/>


<!-- R. Narrative (free form text) -->
<!-- Text -->
<xs:element type="xs:string" name="tbNarrative"/>


<!-- S. Estimated Restoration Date for all Affected Customers -->
<!-- Who Can Receive Power -->
<!-- Date - month -->
```

```
        <xs:element type="xs:string" name="tbEstRestMonth"/>

        <!-- Restoration Date -->
        <!-- Date - day -->
        <xs:element type="xs:string" name="tbEstRestDay"/>

        <!-- Restoration Date -->
        <!-- Date - year -->
        <xs:element type="xs:string" name="tbEstRestYear"/>


        <!-- T. Name of Assets Impacted (list) -->
        <!-- Text -->
        <xs:element type="xs:string" name="tbGenerator"/>


        <!--U. Notify NERC/E-ISAC -->

        <!-- Notify NERC -->
        <!-- checkbox: "Off" or "Yes" -->
        <xs:element type="xs:string" name="cbNotifyNERC"/>

        <!-- Notify E-ISAC -->
        <!-- checkbox: "Off" or "Yes" -->
        <xs:element type="xs:string" name="cbNotifyEISAC"/>

        <!-- Form Completion date and time (form header) -->
        <xs:element type="xs:string" name="txtTimeStamp4"/>

      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

PNNL-32414

Pacific Northwest National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99352
1-888-375-PNNL (7665)

www.pnnl.gov