



# Information Exchange Policy 2.0 JSON Specification



## Copyright Notice

Copyright (c) Forum of Incident Response and Security Teams (FIRST) (2017). All Rights Reserved.

## Abstract

The FIRST Information Exchange Policy (IEP) framework enables information providers to inform users of how they may use the information they receive. IEP ensures that both parties are aware of any restrictions on the use of the shared information, and reduces the likelihood of misunderstandings.

## Co-chairs

The FIRST IEP-SIG Co-chairs at the time of release were:

- Terry MacDonald
- Paul McKitrick
- Merike Kaeo
- Steve Mancini

## Editors

The FIRST IEP v2 JSON Implementation Standard was created and edited by the following people:

- Terry MacDonald
- Paul McKitrick

## Contributors

The following people contributed to the FIRST IEP v2 JSON Implementation Standard:

- Terry MacDonald
- Paul McKitrick



## Contents

Copyright Notice.....	1
Abstract.....	1
Co-chairs .....	1
Editors.....	1
Contributors.....	1
1. Introduction .....	4
1.1 Purpose.....	4
1.2 Requirements .....	4
1.3 Terminology.....	4
2. Architecture.....	6
3. JSON IEP Format .....	6
4. IEP Information.....	7
4.1 IEP Structure .....	7
4.2 Using an IEP .....	7
4.3 Versioning IEPs.....	7
4.4 Caching IEPs.....	7
5. IEP Policy Statements.....	8
5.1 id .....	8
5.2 name .....	8
5.3 version.....	8
5.4 start_date .....	8
5.5 end_date .....	8
5.6 encrypt_in_transit.....	9
5.7 encrypt_at_rest .....	9
5.8 permitted_actions.....	9
5.9 affected_party_notifications .....	10
5.10 tlp.....	11
5.11 attribution .....	11
5.12 obfuscate_affected_parties.....	12
5.13 unmodified_resale .....	12
5.14 external_reference .....	13
6. IEP Example .....	14



- 7. IEP Policy File..... 15
  - 7.1 IEP Policy File Structure..... 15
  - 7.2 IEP Policy File naming..... 17
  - 7.3 IEP Policy File network accessibility..... 17
  - 7.4 Moving IEP Policy Files to a different URL..... 17
  - 7.5 Caching IEP Policy Files..... 17
- 8. IEP Policy Reference ..... 18
  - 8.1 IEP Policy Reference Structure ..... 18
  - 8.2 IEP Policy Reference URL naming ..... 18
  - 8.3 IEP Policy Reference or Embedded IEP Policy ..... 18
  - 8.4 IEP Policy Reference lookups..... 19
- 9. IEP Policy Reference Statements..... 20
  - 9.1 id\_ref..... 20
  - 9.2 url..... 20
  - 9.3 version..... 20
- 10. IEP Policy Reference Complete Example..... 21
- 11. Handling IEP Policy Errors ..... 21
  - 11.1 No IEP..... 21
  - 11.2 Invalid IEP ..... 21
  - 11.3 Missing IEP..... 21
  - 11.4 IEP Policy References pointing to non-existent IEP Policy Files..... 21
  - 11.5 IEP Policy References pointing to non-existent id ..... 22
  - 11.6 Default Unknown IEP ..... 22
- 12. Pre-defined FIRST IEP JSON Policy Files ..... 24
  - 12.1 FIRST IEP-SIG IEP 2.0 TLP Red Policy ..... 24
  - 12.2 FIRST IEP-SIG IEP 2.0 TLP Amber Policy..... 24
  - 12.3 FIRST IEP-SIG IEP 2.0 TLP Green Policy..... 24
  - 12.4 FIRST IEP-SIG IEP 2.0 TLP White Policy..... 24
- 13. Bibliography..... 25



## 1. Introduction

### 1.1 Purpose

Automating the exchange of security and threat information in a timely manner is crucial to the future and effectiveness of the security response community. The timely distribution of sensitive information will only thrive in an environment where both producers and consumers have a clear understanding of how shared information can and cannot be used, with very few variations of interpretation.

The general lack of adequate policy that supports information exchange is increasingly becoming an impediment to timely sharing. This will only be exacerbated as more organizations start actively participating in information exchange communities and the volume of security and threat information being shared continues to grow.

The Traffic Light Protocol<sup>1</sup> (TLP) is the most commonly used method to mark and protect information that is shared. The original intent behind TLP was to speed up the time-to-action on shared information by pre-declaring the permitted redistribution of that information, reducing the need for everyone to ask the producer if it could be "shared with XYZ in my organization" and for that purpose TLP still works.

The challenge for producers of information is that they need to be able to convey more than just the permitted redistribution of the information. There can be a lack of clarity when defining and interpreting the permitted actions and uses of information shared between organizations. This is compounded by the sensitive nature and commercially competitive aspects of security and threat information.

FIRST, interested in enabling the global development and maturation of CSIRTs, recognized that the general lack of adequate policy supporting information exchange is increasingly becoming an impediment to information sharing amongst CSIRT teams.

The FIRST IEP Special Interest Group (IEP-SIG) was formed to develop, oversee and grow the IEP Framework and to ensure it met the needs of the community.

### 1.2 Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

An implementation is not compliant if it fails to satisfy one or more of the MUST or REQUIRED level requirements for the protocols it implements. An implementation that satisfies all the MUST or REQUIRED level and all the SHOULD level requirements for its protocols is said to be "unconditionally compliant"; one that satisfies all the MUST level requirements but not all the SHOULD level requirements for its protocols is said to be "conditionally compliant."

### 1.3 Terminology

**Creator:** The organization or individual who defines the IEP Policy, and creates the Policy Statements that define the IEP Policy. The Creator typically creates an Embedded Policy and stores it in an Internet accessible Policy File to allow Providers to reference it.

---

<sup>1</sup> [https://en.wikipedia.org/wiki/Traffic\\_Light\\_Protocol](https://en.wikipedia.org/wiki/Traffic_Light_Protocol)



**Recipient:** An organization or individual that receives Information from another organization or individual.

**IEP:** Information Exchange Policy framework. Enables information providers to inform users of how they may use the information they receive.

**IEP 2.0:** Version 2 of the Information Exchange Policy framework described in this document.

**Implementation:** Software or hardware that implements the Information Exchange Policy framework.

**Policy:** A collection of Policy Statements that together form an IEP Policy.

**Policy Enumeration:** A list of strings, one of which must be selected and used as the value for that Policy Statement.

**Policy File:** A File containing one or more IEP Policies.

**Policy Statement:** A JSON name/value pair that is used to construct an IEP Policy.

**Policy Reference:** An object that references an IEP Policy.

**Policy Reference Statement:** A JSON name/value pair that is used to construct an IEP Policy Reference.

**Provider:** An organization or individual that produces Information that another organization or individual will consume.

## 2. Architecture

The IEP Framework is designed to be succinct, flexible and descriptive. It tries to help producers describe to consumers exactly what they can and can't do with the information that they receive.

The IEP Framework is built from a series of structures that work together to convey the Providers intent.

The Providers intent is documented by an Information Exchange Policy (IEP). The IEP is constructed from a series of Policy Statements that together form an IEP.

An IEP can be created as a standalone IEP Policy File (allowing that file to be referenced from elsewhere), or the IEP can be embedded within another protocol structure such as the STIX Threat Intelligence Sharing Protocol. This difference is shown below in Figure 1 - Embedded vs Referenced.

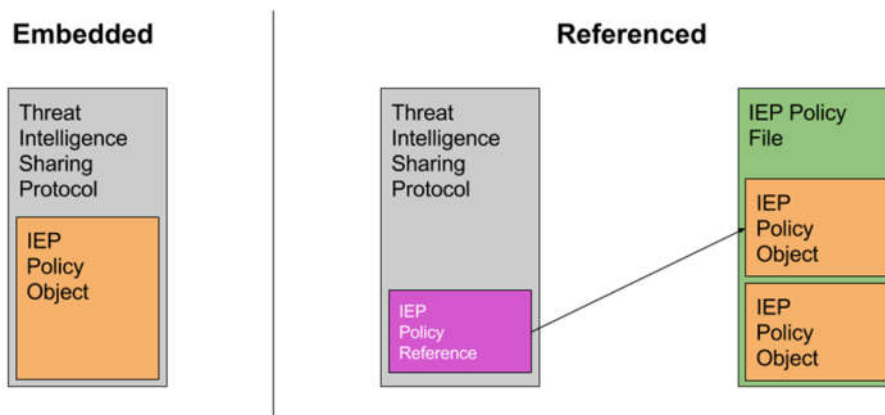


Figure 1 - Embedded vs Referenced

An IEP Policy File **MUST** contain at least one IEP, but **MAY** contain multiple IEP's. Each IEP Policy **MUST** have a unique Policy ID.

A Policy Reference contains a URL and a Policy ID that refers to a particular IEP Policy housed within an Internet accessible IEP Policy File. Policy References are designed to be used within other information sharing standards and protocols to enable easy reuse of common Information Exchange Policies.

## 3. JSON IEP Format

The JSON IEP format uses the JSON text format for the serializations of structured data as defined in RFC7159 [1].

In addition, the IEP Policy Statements naming convention follows the same conventions as described in the STIX Version 2.0 - Part 1 - WD02 standard. The STIX 2.0 standard states that all property names use words separated by underscores (`_`), and all property values that use a mandated vocabulary should be lowercase and separated by hyphens (`-`). This IEP JSON standard follows the same naming structure for its JSON name / value pairs to retain maximum compatibility with JSON and the STIX standard.



## 4. IEP Information

### 4.1 IEP Structure

A JSON IEP MUST be defined as a single JSON Object.

Each JSON IEP JSON Object MUST contain **all** of the IEP Policy Statements as defined in the list below. The mandatory policy statements in IEP Framework 2.0 are:

- id
- name
- version
- start\_date
- end\_date
- encrypt\_in\_transit
- encrypt\_at\_rest
- permitted\_actions
- affected\_party\_notifications
- tlp
- attribution
- obfuscate\_affected\_parties
- unmodified\_resale
- external\_reference

### 4.2 Using an IEP

As mentioned earlier an IEP is defined using an IEP JSON Object. This IEP JSON Object can be used in one of two ways:

- Directly embedded in another intelligence sharing protocol or standard such as STIX, **OR**
- Written into an Internet-accessible IEP Policy File, and referenced using an IEP Policy Reference.

More information about the difference between embedded and referenced IEP Policies can be found in section 8.3.

### 4.3 Versioning IEPs

You cannot update an IEP once it has been applied or associated with a piece of information. IEPs are immutable once they have been associated or applied to a piece of information. If a Producer wishes to change any value within an IEP then a new IEP MUST be created under a new Policy ID, and the new IEP used instead.

### 4.4 Caching IEPs

A Recipient SHOULD keep a cached copy of all IEPs that mark information stored within its information repository, so that all information marked with IEPs will have a corresponding IEP locally available.

A Recipient MAY keep a cached copy of all IEPs that mark information not currently stored within its information repository if it so chooses.



## 5. IEP Policy Statements

Each IEP Policy Statement is a JSON name / value pair (also known as a member), where the name is a string, and it is separated from the value by a colon.

All IEP Policy Statement names use lowercase words separated by underscores (\_), and all IEP Policy Statement values that are policy enumeration strings (i.e. they are part of a list that users can select from) are lowercase and separated by hyphens (-).

An example IEP Policy Statement for the `encrypt_in_transit` Policy Statement is shown below:

```
"encrypt_in_transit": "may"
```

### 5.1 id

The `id` statement is used to identify the IEP, and to allow the IEP to be referenced from other protocols and standards. The `id` statement MUST be a JSON name/value pair. The `id` statement MUST be included in an IEP. The `id` statement name MUST be the JSON string "id", and it must be in lowercase. The `id` statement value MUST be a UUIDv4 or UUIDv5 identifier as defined in RFC4122 [2].

### 5.2 name

The `name` statement is a human readable name for the IEP. The `name` statement MUST be a JSON name/value pair. The `name` statement MUST be included in an IEP. The `name` statement name MUST be the JSON string "name", and it must be in lowercase. The `name` statement value MAY be written in any language.

### 5.3 version

The `version` statement describes which version of the Information Exchange Policy framework that the IEP adheres to. The `version` statement MUST be a JSON name/value pair. The `version` statement MUST be included in an IEP. The `version` statement name MUST be the JSON string "version", and it must be in lowercase. The `version` statement value MUST be the JSON number "2.0" (without the quotes).

### 5.4 start\_date

The `start_date` statement describes when the IEP begins to apply to information that references the IEP.

The `start_date` statement MUST be a JSON name/value pair. The `start_date` statement MUST be included in an IEP. The `start_date` statement name MUST be the JSON string "start\_date", and it must be in lowercase. The `start_date` statement value MUST either be a date string in Co-ordinated Universal Time (UTC) as per RFC3339 [3], or the JSON literal null value **null**. **null** MUST be used as the `start_date` statement value when the producer wishes the IEP to take effect as soon as the recipient receives it.

### 5.5 end\_date

The `end_date` statement describes when the IEP ceases to apply to information that references the IEP.

The `end_date` statement MUST be a JSON name/value pair. The `end_date` statement MUST be included in an IEP. The `end_date` statement name MUST be the JSON string "end\_date", and it must be in lowercase. The `end_date` statement value MUST either be a date string in Co-ordinated Universal Time



(UTC) as per RFC3339 [3], or the JSON literal null value **null**. **null** MUST be used as the end\_date statement value when the producer wishes the IEP to apply to the information forever.

## 5.6 encrypt\_in\_transit

The encrypt\_in\_transit statement is used to inform the recipient if the received information must be encrypted when it is retransmitted by the recipient.

The encrypt\_in\_transit statement MUST be a JSON name/value pair. The encrypt\_in\_transit statement MUST be included in an IEP. The encrypt\_in\_transit statement name MUST be the JSON string "encrypt\_in\_transit", and it must be in lowercase.

The encrypt\_in\_transit statement value MUST be set to one of the following two Policy Enumeration strings:

- "must"
- "may"

As the encrypt\_in\_transit statement value is a policy enumeration string, it MUST be lowercase.

## 5.7 encrypt\_at\_rest

The encrypt\_at\_rest statement is used to inform the recipient if the received information must be encrypted when it is stored by the recipient.

The encrypt\_at\_rest statement MUST be a JSON name/value pair. The encrypt\_at\_rest statement MUST be included in an IEP. The encrypt\_at\_rest statement name MUST be the JSON string "encrypt\_at\_rest", and it must be in lowercase.

The encrypt\_at\_rest statement value MUST be set to one of the following two Policy Enumeration strings:

- "must"
- "may"

As the encrypt\_at\_rest statement value is a policy enumeration string, it MUST be lowercase.

## 5.8 permitted\_actions

The permitted\_actions statement is used to inform of what actions they may take with the information they receive.

The permitted\_actions statement MUST be a JSON name/value pair. The permitted\_actions statement MUST be included in an IEP. The permitted\_actions statement name MUST be the JSON string "permitted\_actions", and it must be in lowercase.

The permitted\_actions statement value MUST be set to one of the following five Policy Enumeration strings:

- "none"
- "contact-for-instruction"
- "internally-visible-actions"



- "externally-visible-indirect-actions"
- "externally-visible-direct-actions"

As the permitted\_actions statement value is a policy enumeration string, it MUST be lowercase.

## 5.9 affected\_party\_notifications

The affected\_party\_notifications statement is used to tell the recipient if they may contact parties who are affected by the information received by the recipient.

**PLEASE NOTE: Setting this Policy Statement to "may" does not allow the recipient to forward full copies of the information received directly to the affected party, but it instead allows the recipient to contact the affected party via any means and inform them of the parts of the information that directly affects the affected party. Please see section 9.2 of the Information Exchange Policy 2.0 Framework Definition document for more information.**

**PLEASE NOTE: The sharing restrictions set by the tlp Policy Statement can be overridden by the setting of the affected\_party\_notifications Policy Statement in some situations. Please see section 9.2 of the Information Exchange Policy 2.0 Framework Definition document for more information.**

The affected\_party\_notifications statement MUST be a JSON name/value pair. The affected\_party\_notifications statement MUST be included in an IEP. The affected\_party\_notifications statement name MUST be the JSON string "affected\_party\_notifications", and it must be lowercase.

The affected\_party\_notifications statement value MUST be set to one of the following two Policy Enumeration strings:

- "may"
- "must-not"

As the affected\_party\_notifications statement value is a policy enumeration string, it MUST be lowercase.

If the affected\_party\_notifications statement value is set to "may", then the recipient MAY contact parties who are affected by the information received by the recipient. Note - this does not allow the recipient to forward the information directly to the affected party, but it instead allows the recipient to contact the affected party and tell them the part of the information that affects them.

If the affected\_party\_notifications statement value is set to "must-not", then the recipient MUST NOT contact any affected parties in regards to the information they received. In some cases, an affected party may be a member of the sharing community that received the original information. If this is the case then the tlp setting takes precedence, and the affected party may be treated as a community member rather than as an affected party.

Please note that in some cases the recipient is required to report the information to law enforcement or other officials due to laws in their local jurisdiction, and those laws will take precedence over this IEP Policy Statement.



### 5.10 tlp

The tlp statement is used to inform the recipient who they may re-share copies of the information with. IEP uses the FIRST TLP-SIG definition of TLP [4].

**PLEASE NOTE: The sharing restrictions set by the tlp Policy Statement can be overridden by the setting of the affected\_party\_notifications Policy Statement in some situations. Please see section 9.2 of the Information Exchange Policy 2.0 Framework Definition document for more information.**

The tlp statement MUST be a JSON name/value pair. The tlp statement MUST be included in an IEP. The tlp statement name MUST be the JSON string "tlp", and it must be in lowercase.

The tlp statement value MUST be set to one of the following four Policy Enumeration strings:

- "red"
- "amber"
- "green"
- "white"

As the tlp statement value is a policy enumeration string, it MUST be lowercase.

### 5.11 attribution

The attribution statement allows the provider to communicate if it wants to be known as the producer of the information if the information is re-shared, or if it wants to remain anonymous.

The attribution statement MUST be a JSON name/value pair. The attribution statement MUST be included in an IEP. The attribution statement name MUST be the JSON string "attribution", and it must be in lowercase.

The attribution statement value MUST be set to one of the following three Policy Enumeration strings:

- "may"
- "must"
- "must-not"

As the attribution statement value is a policy enumeration string, it MUST be lowercase.

If the attribution statement value is set to "must", then the recipient MUST ensure that any information that refers to the IEP MUST maintain attribution of the information to the producer when the information is re-shared.

If the attribution statement value is set to "must-not", then the recipient MUST ensure that any information re-shared MUST NOT attribute the information to the producer in any way when the information is re-shared. **Warning: It still may be possible attribution will still be derived from the information itself.**

If the attribution statement value is set to "may", then it is up to the recipient if they wish to attribute the information to the producer when the information is re-shared.



### 5.12 obfuscate\_affected\_parties

The obfuscate\_affected\_parties statement allows the provider to communicate if the consumer needs to obfuscate the identities of the affected parties contained within the information if the information is re-shared.

*It is up to each information sharing protocol that makes use of IEP to determine what level of obfuscation is acceptable for use in that information sharing protocol.*

The obfuscate\_affected\_parties statement MUST be a JSON name/value pair. The obfuscate\_affected\_parties statement MUST be included in an IEP. The obfuscate\_affected\_parties statement name MUST be the JSON string "obfuscate\_affected\_parties", and it must be in lowercase.

The obfuscate\_affected\_parties statement value MUST be set to one of the following three Policy Enumeration strings:

- "may"
- "must"
- "must-not"

As the obfuscate\_affected\_parties statement value is a policy enumeration string, it MUST be lowercase.

If the obfuscate\_affected\_parties statement value is set to "must", then the recipient MUST ensure that any information that refers to the IEP MUST obfuscate the identities of affected parties mentioned within the information when the information is re-shared.

If the obfuscate\_affected\_parties statement value is set to "must-not", then the recipient MUST ensure that any information that refers to the IEP MUST NOT obfuscate the identities of affected parties mentioned within the information when the information is re-shared. **Warning:** *It still may be possible affected parties will be identified from the derived information even after obfuscation.*

If the obfuscate\_affected\_parties statement value is set to "may", then it is up to the recipient if they wish to obfuscate the identities of affected parties mentioned within the information when the information is re-shared.

### 5.13 unmodified\_resale

The unmodified\_resale statement allows the provider to communicate if the consumer is permitted to resell the information information they receive from the producer within their own product offerings in an unmodified state or one that is semantically equivalent. Semantically equivalent in this context means transposing the data from one format to another i.e. transposing the information from a CSV file format to a JSON file format would be considered semantically equivalent.

The unmodified\_resale statement MUST be a JSON name/value pair. The unmodified\_resale statement MUST be included in an IEP. The unmodified\_resale statement name MUST be the JSON string "unmodified\_resale", and it must be in lowercase.



The `unmodified_resale` statement value MUST be set to one of the following two Policy Enumeration strings:

- "may"
- "must-not"

As the `unmodified_resale` statement value is a policy enumeration string, it MUST be lowercase.

If the `unmodified_resale` statement value is set to "may", then the consumer MAY sell the information they receive in an unmodified (or **semantically equivalent format**) without restriction.

If the `unmodified_resale` statement value is set to "must-not", then the recipient MUST NOT sell or resell any information that refers to the IEP in an unmodified (or **semantically equivalent format**).

**PLEASE NOTE: Setting the `unmodified_resale` statement value to "must-not" does not restrict the consumer from deriving their own information from the information provided by the producer, and then selling their own derived information.**

#### 5.14 `external_reference`

The `external_reference` statement allows the provider to provide a URL that provides further human readable information about the policy described in the IEP. This allows a provider to provide more context about the IEP and to document why the certain IEP Policy Statement values were selected, and what the IEP was developed to control.

The `external_reference` statement MUST be a JSON name/value pair. The `external_reference` statement MUST be included in an IEP. The `external_reference` statement name MUST be the JSON string "external\_reference", and it must be in lowercase.

The `external_reference` statement MUST be a URL as defined in RFC3986 [5].



## 6. IEP Example

The following is an example of a complete IEP JSON Object:

```
{
  "id": "01bc4353-4829-4d55-8d52-0ab7e0790df9",
  "name": "FIRST IEP-SIG TLP-AMBER",
  "version": 2.0,
  "start_date": "2017-01-01T00:00:00Z",
  "end_date": null,
  "encrypt_in_transit": "may",
  "encrypt_at_rest": "may",
  "permitted_actions": "externally-visible-direct-actions",
  "affected_party_notifications": "may",
  "tlp": "amber",
  "attribution": "must-not",
  "obfuscate_affected_parties": "may",
  "unmodified_resale": "must-not",
  "external_reference": " https://www.first.org/tlp"
}
```



## 7. IEP Policy File

### 7.1 IEP Policy File Structure

A JSON IEP Policy file MUST contain at least one IEP within it. A JSON IEP file MAY contain multiple IEPs within it if desired. A JSON IEP Policy file MUST NOT contain any other JSON structure other than one or more IEPs.

If the JSON IEP Policy File contains a single IEP Policy, then the file MUST only contain the single IEP Policy JSON object, e.g.

```
{
  "id": "01bc4353-4829-4d55-8d52-0ab7e0790df9",
  "name": "FIRST IEP-SIG TLP-AMBER",
  "version": 2.0,
  "start_date": "2017-01-01T00:00:00Z",
  "end_date": null,
  "encrypt_in_transit": "may",
  "encrypt_at_rest": "may",
  "permitted_actions": "externally-visible-direct-actions",
  "affected_party_notifications": "may",
  "tlp": "amber",
  "attribution": "must-not",
  "obfuscate_affected_parties": "may",
  "unmodified_resale": "must-not",
  "external_reference": " https://www.first.org/tlp"
}
```

If the JSON IEP Policy File contains multiple IEP Policies, then the IEP Policy File MUST only contain a single JSON array containing all the IEP Policy JSON objects as members of the JSON array, e.g.

```
[
  {
    "id": "01bc4353-4829-4d55-8d52-0ab7e0790df9",
    "name": "FIRST IEP-SIG TLP-AMBER",
    "version": 2.0,
    "start_date": "2017-01-01T00:00:00Z",
    "end_date": null,
    "encrypt_in_transit": "may",
```





```
    "encrypt_at_rest": "may",
    "permitted_actions": "externally-visible-direct-actions",
    "affected_party_notifications": "may",
    "tlp": "amber",
    "attribution": "must-not",
    "obfuscate_affected_parties": "may",
    "unmodified_resale": "must-not",
    "external_reference": " https://www.first.org/tlp"
  },
  {
    "id": "5e607e88-ab70-4977-8c1b-ee3a16b0f68c",
    "name": "FIRST IEP-SIG TLP-RED",
    "version": 2.0,
    "start_date": "2017-01-01T00:00:00Z",
    "end_date": null,
    "encrypt_in_transit": "must",
    "encrypt_at_rest": "may",
    "permitted_actions": "internally-visible-direct-actions",
    "affected_party_notifications": "must-not",
    "tlp": "red",
    "attribution": "must-not",
    "obfuscate_affected_parties": "must",
    "unmodified_resale": "must-not",
    "external_reference": " https://www.first.org/tlp"
  }
]
```



## 7.2 IEP Policy File naming

JSON IEP Policy Files SHOULD end with a ".iepj" file extension where possible.

## 7.3 IEP Policy File network accessibility

Creators SHOULD ensure that IEP Policy Files are made available at the network accessible URLs.

Providers SHOULD ensure that when IEP Policy References are used to mark information that Recipients will be able to access the referenced IEP Policy Files. To clarify, IEP Policy Files MAY be publicly accessible on the Internet, or MAY be housed within a private or restricted network – the only requirement is that the Recipient of the information marked with the IEP has the ability to access them.

## 7.4 Moving IEP Policy Files to a different URL

If an IEP Policy File needs to be moved to a different URL, then a URL redirection SHOULD be made to ensure that implementations that ingest old information marked with an IEP will still work as they will be redirected to the new IEP Policy File location.

If a URL redirection is not possible then Providers SHOULD reissue new versions of the old information marked with an IEP Policy Reference that points to the new IEP Policy File location. It is up to Providers to determine which old information is worth reissuing with an updated IEP Policy.

## 7.5 Caching IEP Policy Files

Recipients SHOULD keep a cached copy of all IEP Policy Files that contain IEP Policy Objects that mark information stored within its information repository, so that all information marked with IEPs will have a corresponding IEP locally available.

Recipients SHOULD periodically access remote IEP Policy Files that contain IEP Policy Objects that mark information stored within its information repository, and update any IEP Policy Files in the IEP Policy File cache that were changed. It is up to Implementations' to determine how often this cache update takes place.

## 8. IEP Policy Reference

IEP Policy References are used to reference an IEP located in a different place. This functionality was designed to allow the development of community-shared IEPs to enable faster, automated sharing, and to reduce the communication overhead of embedding the same IEPs over and over again.

### 8.1 IEP Policy Reference Structure

A JSON IEP Policy Reference MUST be defined as a single JSON Object.

Each JSON IEP Policy Reference JSON Object MUST contain one of each of the IEP Policy Reference Statements as defined in the list below. The mandatory policy reference statements in IEP Framework 2.0 are:

- `id_ref`
- `url`
- `version`

### 8.2 IEP Policy Reference URL naming

As mentioned in section 0, JSON IEP Policy Files SHOULD end with a ".iepj" file extension where possible. This in turn means that JSON Policy Reference URL SHOULD also end with a ".iepj" file extension where possible.

### 8.3 IEP Policy Reference or Embedded IEP Policy

It is up to the protocol or standard using IEP to decide if will apply the IEP to information contained within the information sharing protocol by embedding the IEP JSON Objects directly within the protocol or standard, or if it will make use of the IEP Policy Reference feature, or if it will support both.

Any protocols or standards that leverage IEP SHOULD support both embedded IEPs and IEP Policy References. This provides Providers with the greatest flexibility in how they apply the IEP Policy to the information they are sharing.

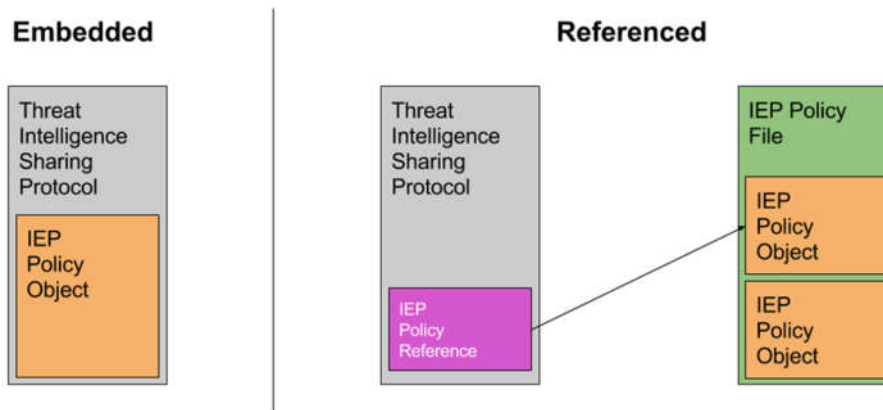


Figure 2 - Embedded vs Referenced



## 8.4 IEP Policy Reference lookups

IEP Policy Reference lookups SHOULD use the following steps to resolve the references to retrieve the IEP identified by the IEP Policy Reference:

1. The id\_ref Policy Reference Statement is read and the id\_ref extracted.
2. The Implementation checks if it has a cached version of the IEP whose id matches the id\_ref.
3. If the Implementation has a cached copy of the IEP, then the cached copy of the IEP is used.
4. If the Implementation does not have a cached copy of the IEP, then the url Policy Reference Statement is read and the URL extracted.
5. The Implementation checks if it has a cached version of the IEP Policy File whose URL matches the url Policy Reference Statement.
6. If the Implementation has a cached version of the IEP Policy File whose URL matches the url Policy Reference Statement, and the cache timer has not expired, then the IEP Policy File is checked for an IEP whose id matches the id\_ref extracted earlier.
7. If the Implementation does not have a cached copy of the IEP Policy File, then the Implementation accesses the URL and downloads the IEP Policy File at the URL.
8. The Implementation checks the IEP Policy File to ensure that it is valid.
9. If the IEP Policy File is invalid then the process in section 11 – "Handling IEP Policy Errors" is followed.
10. If the IEP Policy File is valid, then a copy of the IEP Policy File is stored in the IEP Policy File cache, and the IEP Policy File is checked for an IEP whose id matches the id\_ref extracted earlier.
11. If the IEP has the correct id and is valid, then a local cached copy of the IEP is stored and used, and the Implementation lookup has ended.
12. If the IEP Policy File is valid but there is no IEP with the correct id within the Policy File then the process in section 11 – "Handling IEP Policy Errors" is followed.



## 9. IEP Policy Reference Statements

Each IEP Policy Reference Statement is a JSON name / value pair (also known as a member), where the name is a string, and it is separated from the value by a colon. An example IEP Policy Reference Statement for the url Policy Reference Statement is shown below:

```
"url": "https://www.first.org/iep/2.0/first-iep-sig-tlp-amber.iepj"
```

### 9.1 id\_ref

The id\_ref statement is used to identify the IEP that resides within the IEP Policy File referenced and located at the URL also provided. The id\_ref statement MUST be a JSON name/value pair. The id\_ref statement MUST be included in a IEP Policy Reference object. The id\_ref statement name MUST be the JSON string "id\_ref", and it must be in lowercase. The id\_ref statement value MUST be a UUIDv4 or UUIDv5 identifier as defined in RFC4122 [2], and MUST be an identifier that exists (or had existed in the past) within the IEP Policy File referenced and located at the URL also included within the IEP Policy Reference object.

### 9.2 url

The url statement is used to specify the URL that will enable the recipient to access and read the IEP Policy File that contains the IEP that the information has been marked with.

The url statement MUST be a JSON name/value pair. The url statement MUST be included in a IEP Policy Reference object. The url statement name MUST be the JSON string "url", and it must be in lowercase. The url statement value MUST be a URL as defined in RFC3986 [5], and MUST point to an Internet accessible IEP Policy File.

### 9.3 version

The version statement describes the IEP Framework version that this Policy Reference is.

The version statement MUST be a JSON name/value pair. The version statement MUST be included in a IEP Policy Reference object. The version statement name MUST be the JSON string "version", and it must be in lowercase. The version statement value MUST be the JSON number "2.0" (without the quotes).



## 10. IEP Policy Reference Complete Example

An example IEP Policy Reference JSON object is shown below:

```
{
  "id_ref": "01bc4353-4829-4d55-8d52-0ab7e0790df9",
  "url": "https://www.first.org/iep/2.0/first-iep-sig-tlp-amber.iepj",
  "version": 2.0
}
```

## 11. Handling IEP Policy Errors

This section provides guidance on how to handle IEP errors gracefully.

### 11.1 No IEP

If some information is received, and that information is not marked with an IEP, then the recipient **MUST** follow whatever guidelines they have agreed with the Provider. In this case no IEP has been applied to the information, and IEP is not being used to control how the recipient is allowed to use the received information.

### 11.2 Invalid IEP

If some information is received, and that information is marked with an IEP, but that IEP is invalid, then the recipient **MUST** contact the Provider to clarify what restrictions they should apply to the received information. The Creator or Provider **SHOULD** correct the Invalid IEP and the Provider **SHOULD** reissue **an updated version of** the information with a valid IEP.

### 11.3 Missing IEP

If some information is received, and that information is marked with an embedded IEP, but that **embedded IEP is missing or otherwise unavailable**, then the recipient **MUST** contact the Provider to clarify what restrictions they should apply to the received information. **The Provider SHOULD reissue the information with a valid embedded IEP.**

### 11.4 IEP Policy References pointing to non-existent IEP Policy Files

If some information is received, and that information is marked with an IEP Policy Reference, but that IEP Policy Reference points to a URL that is **currently** unreachable, then the following rules apply:

1. If the Recipient had previously successfully accessed the IEP Policy File at the URL defined in the IEP Policy Reference, and the Policy File was valid, and one of the IEPs contained within the IEP Policy File had the same id as the id\_ref contained within the IEP Policy Reference, and the Recipient has a cached copy of the IEP, then the Recipient **MAY** continue to use the previous cached copy of the IEP as if the IEP Policy Lookup worked correctly.
2. If the Recipient had **never** successfully accessed the IEP Policy File at the URL defined in the IEP Policy Reference, then the Recipient **MUST** contact the Provider to clarify what restrictions they should apply to the received information.



3. If the Recipient had previously successfully accessed the IEP Policy File at the URL defined in the IEP Policy Reference, and the IEP Policy File was valid, and the Recipient has a cached copy of the IEP, but the `id_ref` contained within the IEP Policy Reference does not match any of the `id`'s within the IEP Policy File, then the Recipient **MUST** contact the Provider to clarify what restrictions they should apply to the received information.
4. A Recipient **SHOULD** keep a cached copy of all IEPs that mark information stored within its information repository, so that all information marked with IEPs will have a corresponding IEP locally available.
5. If a recipient has attempted to contact the Provider for clarification on use of the information but has been unable to get a response, or if the recipient is unable (or unwilling) to contact the Provider, then the **Default Unknown IEP** applies (see section 11.6 - "Default Unknown IEP" later in this document).

The Creator **SHOULD** ensure that the missing IEP Policy File is made available at the URL. If an IEP Policy File needs to move to a different URL, then a URL redirection **SHOULD** be made to ensure that old information marked with an IEP will be redirected to the new location.

### 11.5 IEP Policy References pointing to non-existent id

If some information is received, and that information is marked with an IEP Policy Reference, and that IEP Policy Reference points to a valid IEP Policy File, but the `id_ref` contained within the IEP Policy Reference does not match any of the `id`'s within the IEP Policy File, then the following rules apply:

1. If the Recipient had previously successfully accessed the IEP Policy File at the URL defined in the IEP Policy Reference, and the Policy File was valid, and one of the IEPs contained within the IEP Policy File had the same `id` as the `id_ref` contained within the IEP Policy Reference, and the Recipient has a cached copy of the IEP, then the Recipient **MAY** continue to use the previous cached copy of the IEP as if the IEP Policy Lookup worked correctly.
2. If the Recipient had previously successfully accessed the IEP Policy File at the URL defined in the IEP Policy Reference, and the Policy File was valid, and the Recipient has a cached copy of the IEP, but the `id_ref` contained within the IEP Policy Reference does not match any of the `id`'s within the IEP Policy File, then the Recipient **MUST** contact the Provider to clarify what restrictions they should apply to the received information.

Implementations **SHOULD** periodically try to access IEP Policy Files that IEP Policy References with missing `ids` were referring to in order to check if an IEP with the correct `id` now exists within the IEP Policy File at the URL.

### 11.6 Default Unknown IEP

This IEP is designed to be the most restrictive as possible, as it is only used when Implementations know that an IEP was applied, but are unable to find out what it was, no longer have a cached copy of the IEP, and are unable to contact the Provider of the information to provide guidance as to which IEP should be applied.

In this case the IEP Framework applies a default restrictive policy to the information to ensure that it cannot be shared to any other entity other than the Recipient.

The Default Unknown IEP is below:



```
{
  "id": "e4eb1db1-e0fb-4200-9f4c-4c713bb197aa",
  "name": "FIRST IEP-SIG Unknown IEP",
  "version": 2.0,
  "start_date": "2017-01-01T00:00:00Z",
  "end_date": null,
  "encrypt_in_transit": "must",
  "encrypt_at_rest": "may",
  "permitted_actions": "internally-visible-actions",
  "affected_party_notifications": "must-not",
  "tlp": "red",
  "attribution": "must-not",
  "obfuscate_affected_parties": "must",
  "unmodified_resale": "must-not",
  "external_reference": " https://www.first.org/iep"
}
```





## 12. Pre-defined FIRST IEP JSON Policy Files

The FIRST IEP-SIG have developed some standard IEP Policy Files and have made them Internet accessible to help Implementers standardize on a common set of IEPs. This will aid adoption and ensure all parties within a information sharing community know what behaviour is expected of them.

These policies are based on the FIRST TLP-SIG "TLP FIRST Standards Definitions and Usage Guidance — Version 1.0" available at <https://www.first.org/tlp>.

### 12.1 FIRST IEP-SIG IEP 2.0 TLP Red Policy

Policy Name	FIRST IEP-SIG IEP TLP Red
Policy ID	5e607e88-ab70-4977-8c1b-ee3a16b0f68c
Policy URL	<a href="https://www.first.org/iep/2.0/first-iep-sig-tlp-red.iepj">https://www.first.org/iep/2.0/first-iep-sig-tlp-red.iepj</a>

### 12.2 FIRST IEP-SIG IEP 2.0 TLP Amber Policy

Policy Name	FIRST IEP-SIG IEP TLP Amber
Policy ID	01bc4353-4829-4d55-8d52-0ab7e0790df9
Policy URL	<a href="https://www.first.org/iep/2.0/first-iep-sig-tlp-amber.iepj">https://www.first.org/iep/2.0/first-iep-sig-tlp-amber.iepj</a>

### 12.3 FIRST IEP-SIG IEP 2.0 TLP Green Policy

Policy Name	FIRST IEP-SIG IEP TLP Green
Policy ID	3903ce63-674c-4b70-9457-8c5527dd9115
Policy URL	<a href="https://www.first.org/iep/2.0/first-iep-sig-tlp-green.iepj">https://www.first.org/iep/2.0/first-iep-sig-tlp-green.iepj</a>

### 12.4 FIRST IEP-SIG IEP 2.0 TLP White Policy

Policy Name	FIRST IEP-SIG IEP TLP White
Policy ID	0d783790-b221-40c1-840a-5787330612c1
Policy URL	<a href="https://www.first.org/iep/2.0/first-iep-sig-tlp-white.iepj">https://www.first.org/iep/2.0/first-iep-sig-tlp-white.iepj</a>



## 13. Bibliography

- [1] "RFC7159," March 2014. [Online]. Available: <https://tools.ietf.org/html/rfc7159>.
- [2] "RFC4122," July 2005. [Online]. Available: <https://www.ietf.org/rfc/rfc4122.txt>.
- [3] "RFC3339," July 2002. [Online]. Available: <https://tools.ietf.org/html/rfc3339>.
- [4] F. TLP-SIG, "Traffic Light Protocol," August 2016. [Online]. Available: <https://www.first.org/tlp>.
- [5] "RFC3986," January 2005. [Online]. Available: <https://tools.ietf.org/html/rfc3986>.