Trustmark Framework Technical Specification

Georgia Tech Research Institute

National Strategy for Trusted Identities in Cyberspace (NSTIC) Trustmark Pilot

Version 1.1

December 8, 2016

Table of Contents

1 Introduction

A **Trustmark** is a machine-readable, cryptographically signed digital artifact, issued by a **Trustmark Provider** to a **Trustmark Recipient**, and relied upon by one or more **Trustmark Relying Parties**. A Trustmark represents an official attestation by the Trustmark Provider of conformance by the Trustmark Recipient to a well-defined set of requirements pertaining to trust and/or interoperability for the purpose of interaction with and use of digital information resources and services. A Trustmark Relying Party may rely upon a Trustmark as the basis for third-party trust in the Trustmark Recipient with respect to the set of requirements represented by the Trustmark. A **Trustmark Definition** expresses the specific set of requirements represented by a Trustmark. A Trustmark Provider cryptographically signs and publishes various Trustmarks for organizations (Trustmark Recipients) that wish to obtain and use those Trustmarks as a mechanism for establishing trust with other entities (Trustmark Relying Parties), including partner organizations and individuals. These concepts and others together comprise the **Trustmark Framework**.

The purpose of this **Trustmark Framework Technical Specification** is to provide normative language that governs the structures that comprise the Trustmark Framework and the rules and policies related to the operational use of these structures. The remainder of this document is structured as follows. Section 2 contains basic front matter, including explanations of notation, namespaces, schemas, and data types used. Section 3 introduces the Trustmark Framework in more detail and sets the stage for Sections 4 and 5, which form the heart of this specification. Section 4 provides a detailed description of each artifact in the Trustmark Framework, and Section 5 addresses basic operational considerations for users of the Trustmark Framework. Section 6 contains a list of external references that are cited at various places throughout this document.

2 Notation

2.1 Keywords

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].

2.2 XML Schema Documents

This specification uses XML schema documents conforming to [Schema1] and [Schema2] and accompanying normative text to describe the syntax and semantics of XML-encoded Trustmark Framework documents. In case of disagreement between Trustmark Framework schema documents and listings in this specification, the schema documents shall take precedence. Note that in some cases, the normative text of this specification imposes constraints beyond those indicated by the schema documents.

This specification uses the following namespace prefixes to reference the given namespaces as follows, whether or not a namespace declaration is present in the example.

Prefix	Namespace
`xs`	XML Schema namespace [Schema1] `http://www.w3.org/2001/XMLSchema`
`xsi`	XML Schema Instance namespace [Schema1] `http://www.w3.org/2001/XMLSchema-instance`
`ds`	XML Signature Syntax and Processing namespace [XML Sig] `http://www.w3.org/2000/09/xmldsig#`
`tf`	Trustmark Framework namespace `https://trustmark.gtri.gatech.edu/specifications/trustmark-framework/1.1/schema/`

`my`	Example namespace `https://my.example/`

2.3 XML Schema Data Types

2.3.1 String Values

Trustmark Framework string values are of type `xs:string`. In addition to the requirements specified by [Schema2], a string value MUST be non-empty, and it must include at least one non-whitespace character, in accordance with the definition of whitespace in [XML].

2.3.2 URI and URL Values

Trustmark Framework Uniform Resource Identifier (URI) and Uniform Resource Locator (URL) values are of type `xs:anyURI`. In addition to the requirements specified by [Schema2], both URI and URL values MUST be absolute URIs, as defined in [RFC 2396]. URL values MUST be dereferenceable.

2.3.3 Boolean Values

Trustmark Framework Boolean values are of type `xs:boolean`. As specified by [Schema2], a Boolean value MUST contain a value of `true`, `false`, `1` (which is equivalent to `true`), or `0` (which is equivalent to `false`).

2.4 Typographical Conventions

This specification uses the following typographical conventions in text:

- `<Element>`, for elements in the Trustmark Framework namespace;
- `<ns:ForeignElement>`, for elements in the namespace indicated by the prefix `ns`;
- `Attribute`, for attributes in the Trustmark Framework namespace; and
- `DataType`, for complex types and simple types in the Trustmark Framework Namespace.

Listings of Trustmark Framework XML schema fragments and example XML fragments appear as follows.

```
<xs:element name      = "Identifier"
            type      = "xs:string"
            minOccurs = "1"
            maxOccurs = "1"/>
```

This specification uses sample XML fragments that have been derived from actual Trustmark Definitions and Trust Interoperability Profiles developed by the Georgia Tech Research Institute (GTRI) as part of a National Strategy for Trusted Identities in Cyberspace (NSTIC) Pilot project on the Trustmark concept. To illustrate specific concepts required within specific sections of this document, these sample XML fragments may differ from the corresponding XML fragments in the Trustmark Definitions and Trust Interoperability Profiles from which they were derived. Please note that all XML fragments appearing in this specification are for illustrative purposes only, and any sample XML content herein should not be interpreted to represent or substitute for an actual Trustmark Definition or Trust Interoperability Profile.

3 Basic Trustmark Framework Concepts

This section introduces the basic concepts that comprise the Trustmark Framework. It includes both the **Trustmark Framework Concept Map**, described in Section 3.1, and the **Trustmark Legal Framework**, described in Section 3.2. The normative language in Sections 4 and 5 provides the necessary formal structure to enable the Trustmark Framework to fulfill its intended purpose described in this section.

3.1 The Trustmark Framework Concept Map

Figure 1 illustrates the Trustmark Framework Concept Map, which illustrates the basic elements in the Trustmark Framework. It provides a high-level description of what a Trustmark is, how it is defined, and how it is used.

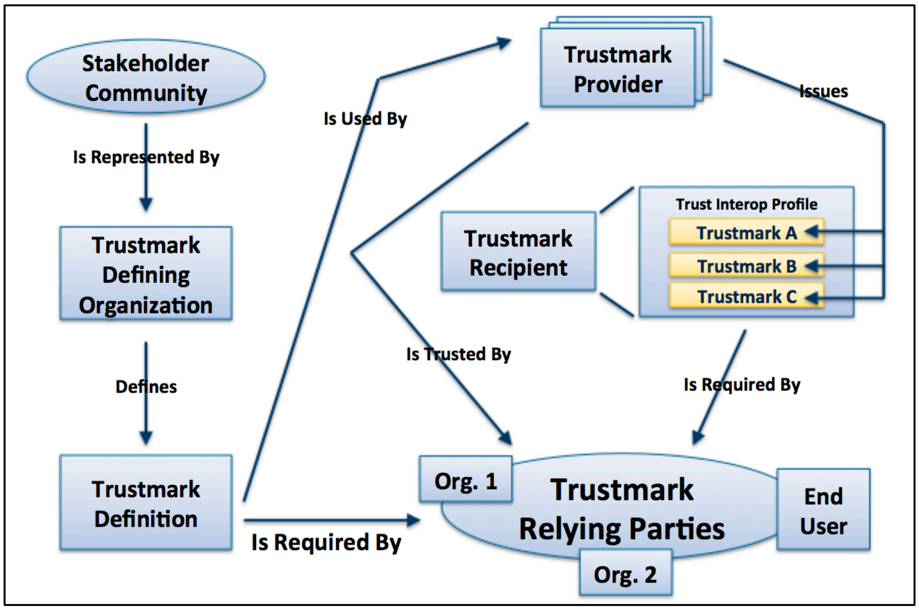

Figure 1: The Trustmark Framework Concept Map

The following terms and concepts are represented in the preceding figure.

A **Trustmark** is a machine-readable, cryptographically signed digital artifact that represents a statement of conformance to a well-scoped set of trust and/or interoperability requirements. It exists as an eXtensible Markup Language (XML) object and conforms to the normative specification defined in Section 4.4 of this document. Its issuer, also called the Trustmark Provider, cryptographically signs it to ensure its integrity.

A **Trustmark Provider** is an organization or other business entity that issues a Trustmark to a **Trustmark Recipient** based on a formal assessment process. The Trustmark serves as a formal attestation by the Trustmark Provider that the Trustmark Recipient conforms to a well-defined set of requirements. The Trustmark is issued under a **Trustmark Policy** (not shown in figure) and is subject to a **Trustmark Recipient Agreement** (also not shown in figure). A Trustmark Recipient is always an organization or other business entity; Trustmarks are not intended for issuance to individuals.

A **Trustmark Definition** specifies the conformance criteria that the Trustmark Recipient must meet, as well as the formal assessment process that the Trustmark Provider must perform to assess whether the Trustmark Recipient qualifies for the Trustmark. There can be many different types of Trustmarks, and each type of Trustmark has its own Trustmark Definition. A Trustmark Definition exists as an eXtensible Markup Language (XML) object and conforms to the normative specification defined in Section 4.3 of this document.

A Trustmark Definition is developed and maintained by a **Trustmark Defining Organization**, which represents the interests of one or more Stakeholder Communities. A Trustmark Defining Organization is similar in function to a Standards Development Organization. A Trustmark Defining Organization does not play an active role in the issuance of a Trustmark, and does not enter into any legal agreement as part of the issuance or use of Trustmarks; its only role is to represent Stakeholder Communities and publish Trustmark Definitions that represent the requirements of those communities.

Possession of a Trustmark by the Trustmark Recipient is required by a **Trustmark Relying Party**, which treats the Trustmark as 3rd-party-verified evidence that the Trustmark Recipient meets the trust and/or interoperability criteria set forth in the Trustmark Definition for the Trustmark. When it relies on a Trustmark, a Trustmark Relying Party enters into a **Trustmark Relying Party Agreement** (not shown in figure) with the Trustmark Provider. A Trustmark Relying Party may be either an organization or an individual.

A **Trustmark Relying Party** defines a **Trust Interoperability Profile** that expresses a trust and interoperability policy in terms of a set of Trustmarks that a Trustmark Recipient must possess, in order to meet its trust and interoperability requirements. A Trust Interoperability Profile exists as an eXtensible Markup Language (XML) object and conforms to the normative specification defined in Section 4.6 of this document.

After issuing a Trustmark, a Trustmark Provider must publish a **Trustmark Status Report** (not shown in figure) that provides status information about the Trustmark, updating the report as needed if the Trustmark's status changes, e.g., from "active" to "revoked" or "expired". A Trustmark Relying Party may request the Trustmark Status Report periodically or as needed to check whether the Trustmark is still valid and suitable for use as the basis for trust. A Trustmark Status Report exists as an eXtensible Markup Language (XML) object and conforms to the normative specification defined in Section 4.5 of this document.

3.2 The Trustmark Legal Framework

Figure 2 illustrates the Trustmark Legal Framework. It builds upon the basic Trustmark Framework Concept Map depicted in Figure 1, adding detail about how Trustmark issuance, use, and reliance work from a legal perspective.

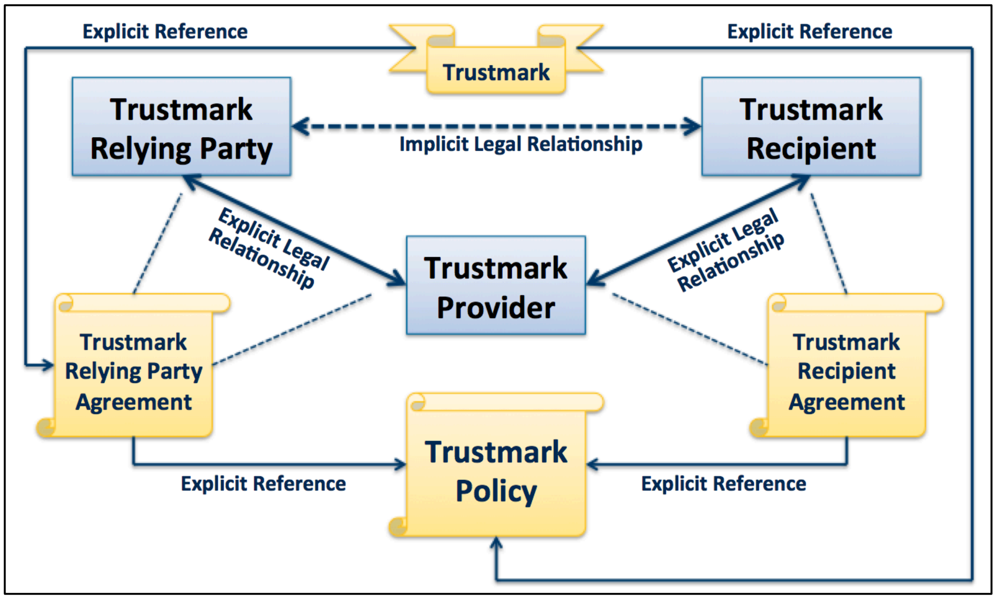

Figure 2: The Trustmark Legal Framework

Within the Trustmark Legal Framework, a Trustmark is issued from a Trustmark Provider to a Trustmark Recipient under a Trustmark Recipient Agreement, which is a standard two-party contract that establishes an explicit legal agreement between the Trustmark Provider and Trustmark Recipient. The Trustmark Recipient Agreement incorporates a Trustmark Policy by reference. The Trustmark Provider and the Trustmark Recipient both must sign the Trustmark Recipient Agreement to execute it.

When a Trustmark Relying Party chooses to rely upon a Trustmark, the Trustmark Relying Party must enter into a Trustmark Relying Party Agreement with the Trustmark Provider. The Trustmark Relying Party Agreement is also a two-party contract; however, it is not a standard two-party agreement that both parties must sign. Instead, it is a "clickwrap" or "clickthrough" agreement that becomes effective by virtue of the Trustmark Relying Party using or relying on a Trustmark issued by the Trustmark Provider. The Trustmark Relying Party Agreement also incorporates the Trustmark Policy by reference.

Note, as indicated by Figure 2, that the Trustmark object contains references to both the Trustmark Policy under which it was issued and the Trustmark Relying Party Agreement to which all Trustmark Relying Parties are subject if they choose to use or rely upon the Trustmark.

Note also that even though the purpose of a Trustmark is to provide a basis for trust between the Trustmark Recipient and Trustmark Relying Party, the Trustmark Legal Framework does not establish an explicit legal relationship between these two entities. Instead, it establishes separate explicit legal relationships between each entity and a third party, the Trustmark Provider.

Establishment of a suitable Trustmark Policy, Trustmark Recipient Agreement, and Trustmark Relying Party Agreement are mandatory for the issuance of a Trustmark, as stipulated in Section 5.2.3 of this specification; however, this specification does not provide any further requirements or guidance as to what structure these three documents must follow or what content they must contain.

4 Trustmark Framework Artifacts

This section defines the normative XML schema structures of all artifacts that comprise the Trustmark Framework. It includes an XML schema as well as accompanying normative rules about each structure. The section proceeds as follows. First, Section 4.1 introduces the schema headers and namespace declarations that are common to all XML artifacts in the framework. Next, Section 4.2 introduces some common data structures that are used throughout the framework. Finally, Sections 4.3, 4.4, 4.5, and 4.6 introduce the normative XML schema structures and accompanying rules for a Trustmark Definition, Trustmark, Trustmark Status Report, and Trust Interoperability Profile, respectively.

4.1 Schema Header and Namespace Declarations

The following schema fragment defines the XML namespaces for the Trustmark Framework schema.

```
<xs:schema targetNamespace=
  "https://trustmark.gtri.gatech.edu/specifications/trustmark-
framework/1.1/schema/"
           xmlns:tf=
  "https://trustmark.gtri.gatech.edu/specifications/trustmark-
framework/1.1/schema/"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
           elementFormDefault="qualified"
           attributeFormDefault="qualified"
           version="1">
```

4.2 Common Data Structures

The XML encodings of Trustmarks, Trustmark Definitions, and Trust Interoperability Profiles share the following schema components.

- `id` attribute
- `ref` attribute
- `EntityType` complex type
- `ContactType` complex type
- `ContactKindType` simple type
- `EntityReferenceType` complex type
- `TrustmarkReferenceType` complex type
- `TrustmarkDefinitionReferenceType` complex type
- `TrustInteroperabilityProfileReferenceType` complex type

- `ExtensionType` complex type

The following sections describe the aforementioned schema components.

4.2.1 Attribute `id`

The `id` attribute described in this section, and the `ref` attribute described in the next section, are used for internal cross-references within XML objects in the Trustmark Framework. The value of the `id` attribute uniquely identifies the element that bears the attribute within the XML encoding of a Trustmark, Trustmark Definition, Trustmark Status Report, or Trust Interoperability Profile. Beyond uniquely identifying the element that bears it, the value of the `id` attribute has no meaning in the Trustmark, Trustmark Definition, Trustmark Status Report, or Trust Interoperability Profile.

The following schema fragment defines the `id` attribute.

```
<xs:attribute name="id"
              type="xs:ID">
  <xs:simpleType>
    <xs:restriction>
      <xs:pattern value="[_A-z][_A-z0-9\-]*"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
```

In addition to the constraints expressed in the schema, any element that carries the `id` attribute MUST adhere to the following rules.

1. The element MUST NOT carry the `ref` attribute.
2. The element MUST NOT carry the `xsi:nil` attribute.

4.2.2 Attribute `ref`

The value of the `ref` attribute matches the value of the `id` attribute of some element in the XML encoding of a Trustmark, Trustmark Definition, Trustmark Status Report, or Trust Interoperability Profile.

The following schema fragment defines the `ref` attribute.

```
<xs:attribute name="ref"
              type="xs:IDREF"/>
```

In addition to the constraints expressed in the schema, any element that carries the `ref` attribute MUST adhere to the following rules.

1. The element MUST NOT carry the `id` attribute.
2. The element MUST carry the `xsi:nil` attribute, and the value of that `xsi:nil` attribute MUST be `true`.
3. The value of the `ref` attribute MUST match the value of the `id` attribute of another element, and the name and type of the element MUST match the name and type of the other element.

4.2.3 Complex Type `EntityType`

`EntityType` is a complex type that describes an organization or a business entity, including Trustmark Defining Organizations, Trustmark Providers, Trustmark Recipients, and Trust Interoperability Profile Issuers. The `EntityType` complex type includes the following elements and attributes.

`<Identifier>` [Required]

> `<Identifier>` is a URL; it is the globally unique identifier of an organization or a business entity. Section 5 contains normative language about how an organization or business entity must establish this identifier for various contexts (Trustmark Providers, Trustmark Recipients, etc.)

<Name> [Required]

<Name> is a string; it is the name for the organization or business entity.

<Contact> [One or More]

<Contact> is a ContactType; it is a point of contact for this organization or business entity. At least one of the <Contact> elements MUST have a <Kind> child element with the value PRIMARY. Section 4.2.4 further describes the ContactType complex type.

The following schema fragment defines the EntityType complex type.

```
<xs:complexType name="EntityType">
  <xs:sequence>
    <xs:element name="Identifier"
                type="xs:anyURI"
                minOccurs="1"
                maxOccurs="1"/>
    <xs:element name="Name"
                type="xs:string"
                minOccurs="1"
                maxOccurs="1"/>
    <xs:element name="Contact"
                type="tf:ContactType"
                minOccurs="1"
                maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

The following example XML fragments illustrate the <TrustmarkDefiningOrganization> element, which is of the EntityType complex type. The first example illustrates the minimum content required by its definition.

```
<tf:TrustmarkDefiningOrganization>
  <tf:Identifier>https://trustmark.gtri.gatech.edu/</tf:Identifier>
  <tf:Name>Georgia Tech Research Institute</tf:Name>
  <tf:Contact>
    <tf:Kind>PRIMARY</tf:Kind>
    <tf:Email>TrustmarkFeedback@gtri.gatech.edu</tf:Email>
  </tf:Contact>
</tf:TrustmarkDefiningOrganization>
```

The second example illustrates additional content permitted by its definition. Additional content in this example appears in **boldface** text.

```
<tf:TrustmarkDefiningOrganization>
  <tf:Identifier>https://trustmark.gtri.gatech.edu/</tf:Identifier>
  <tf:Name>Georgia Tech Research Institute</tf:Name>
  <tf:Contact>
    <tf:Kind>PRIMARY</tf:Kind>
    <tf:Responder>George P. Burdell</tf:Responder>
    <tf:Email>TrustmarkFeedback@gtri.gatech.edu</tf:Email>
    <tf:Telephone>404-407-8956</tf:Telephone>
    <tf:PhysicalAddress>
      75 5th Street NW,
      Atlanta, GA 30308
    </tf:PhysicalAddress>
    <tf:MailingAddress>
      75 5th Street NW,
      Suite 900,
      Atlanta, GA 30308
    </tf:MailingAddress>
    <tf:WebsiteURL>http://trustmark.gtri.gatech.edu/</tf:WebsiteURL>
    <tf:Notes>The responder may change.</tf:Notes>
  </tf:Contact>
</tf:TrustmarkDefiningOrganization>
```

4.2.4 Complex Type `ContactType`

`ContactType` is a complex type that describes a point of contact for an organization or a business entity. The `ContactType` complex type includes the following elements.

`<Kind>` [Required]

> `<Kind>` is a `ContactKindType`; it indicates the kind of this point of contact: one of `PRIMARY` and `OTHER`. Section 4.2.5 further describes the `ContactKindType` simple type.

`<Responder>` [Optional]

> `<Responder>` is a string; it is the name of a responder (e.g., a person, department, or job title) through which this organization or business entity may be contacted, if any.

`<Email>` [One or More]

> `<Email>` is a string; it is an electronic mailing address at which this organization or business entity may be contacted.

`<Telephone>` [Any Number]

> `<Telephone>` is a string; it is a telephone number at which this organization or business entity may be contacted, if any.

`<PhysicalAddress>` [Any Number]

> `<PhysicalAddress>` is a string; it is the full text of the physical address at which this organization or business entity may be contacted, if any.

`<MailingAddress>` [Any Number]

> `<MailingAddress>` is a string; it is the full text of the mailing address at which this organization or business entity may be contacted, if any.

`<WebsiteURL>` [Any Number]

> `<WebsiteURL>` is a URL; it is a website address at which this organization or business entity may be contacted, if any.

`<Notes>` [Optional]

> `<Notes>` is a string; it contains additional optional text content about this point of contact.

The following schema fragment defines the complex type `ContactType`.

```
<xs:complexType name="ContactType">
  <xs:sequence>
    <xs:element name="Kind"
                type="tf:ContactKindType"
                minOccurs="1"
                maxOccurs="1"/>
    <xs:element name="Responder"
                type="xs:string"
                minOccurs="0"
                maxOccurs="1"/>
    <xs:element name="Email"
                type="xs:string"
                minOccurs="1"
                maxOccurs="unbounded"/>
    <xs:element name="Telephone"
```

```
                type="xs:string"
                minOccurs="0"
                maxOccurs="unbounded"/>
    <xs:element name="PhysicalAddress"
                type="xs:string"
                minOccurs="0"
                maxOccurs="unbounded"/>
    <xs:element name="MailingAddress"
                type="xs:string"
                minOccurs="0"
                maxOccurs="unbounded"/>
    <xs:element name="WebsiteURL"
                minOccurs="0"
                maxOccurs="unbounded"
                type="xs:anyURI"/>
    <xs:element name="Notes"
                type="xs:string"
                minOccurs="0"
                maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
```

The following example XML fragments illustrate the `ContactType` complex type. The first example illustrates the minimum content required by its definition.

```
<tf:Contact>
  <tf:Kind>PRIMARY</tf:Kind>
  <tf:Email>TrustmarkFeedback@gtri.gatech.edu</tf:Email>
</tf:Contact>
```

The second example illustrates additional content permitted by its definition. Additional content in this example appears in **boldface** text.

```
<tf:Contact>
  <tf:Kind>PRIMARY</tf:Kind>
  <tf:Responder>George P. Burdell</tf:Responder>
  <tf:Email>TrustmarkFeedback@gtri.gatech.edu</tf:Email>
  <tf:Telephone>404-407-8956</tf:Telephone>
  <tf:PhysicalAddress>
    75 5th Street NW,
    Atlanta, GA 30308
  </tf:PhysicalAddress>
  <tf:MailingAddress>
    75 5th Street NW,
    Suite 900,
    Atlanta, GA 30308
  </tf:MailingAddress>
  <tf:WebsiteURL>http://trustmark.gtri.gatech.edu/</tf:WebsiteURL>
  <tf:Notes>The responder may change.</tf:Notes>
</tf:Contact>
```

4.2.5 Simple Type `ContactKindType`

The `ContactKindType` simple type constrains the value space of the `<Kind>` element of the `ContactType` complex type to one of two values: `PRIMARY` and `OTHER`. If the value is `PRIMARY`, the point of contact is the primary point of contact for the organization or business entity; if the value is `OTHER`, the point of contact is not the primary point of contact for the organization or business entity.

The following schema fragment defines the `ContactKindType` simple type.

```
<xs:simpleType name="ContactKindType">
```

```
    <xs:restriction base="xs:string">
      <xs:enumeration value="PRIMARY"/>
      <xs:enumeration value="OTHER"/>
    </xs:restriction>
  </xs:simpleType>
```

For an example of the ContactKindType simple type, see the example for the ContactType complex type in Section 4.2.4.

4.2.6 Complex Type EntityReferenceType

EntityReferenceType is a complex type that describes a reference to an organization or a business entity. Trust Interoperability Profiles employ EntityReferenceType to reference an organization or business entity for which they require only the identifier of the organization or business entity, not its name or points of contact. The EntityReferenceType complex type includes the following elements and attributes.

<Identifier> [Required]

> <Identifier> is a URL that is the globally unique identifier of the referenced organization or business entity. Section 5 contains normative language about how an organization or business entity must establish this identifier for various contexts (Trustmark Providers, Trustmark Recipients, etc.)

<Name> [Optional]

> <Name> is a string that is, if present, the name of the referenced organization or business entity. If the <Name> element is present, the value of the <Name> element SHOULD match the value of the <Name> element as it would appear in the EntityType complex type for this organization or business entity.

<Contact> [Any Number]

> <Contact> is a ContactType (see Section 4.2.4) that is, if present, a point of contact for the referenced organization or business entity. If a <Contact> element is present, the content of the <Contact> element SHOULD match the content of a <Contact> element as it would appear in an EntityType complex type for this organization or business entity.

id [Optional]

> id is an XML ID for this reference to an organization or a business entity.

ref [Optional]

> ref is an XML IDREF to a reference to an organization or a business entity.

The following schema fragment defines the EntityReferenceType complex type.

```
<xs:complexType name="EntityReferenceType">
  <xs:sequence>
    <xs:element name="Identifier"
                type="xs:anyURI"
                minOccurs="1"
                maxOccurs="1"/>
    <xs:element name="Name"
                type="xs:string"
                minOccurs="0"
                maxOccurs="1"/>
    <xs:element name="Contact"
                type="tf:ContactType"
                minOccurs="0"
                maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute ref="tf:id"/>
  <xs:attribute ref="tf:ref"/>
```

```
</xs:complexType>
```

The following example XML fragments illustrate the `<ProviderReference>` element, which is of the `EntityReferenceType` complex type. The first example illustrates the minimum content required by its definition.

```
<tf:ProviderReference>
  <tf:Identifier>https://trustmark.gtri.gatech.edu/</tf:Identifier>
</tf:ProviderReference>
```

The second example illustrates additional content permitted by its definition. Additional content in this example appears in **boldface** text.

```
<tf:ProviderReference tf:id="providerReference">
  <tf:Identifier>https://trustmark.gtri.gatech.edu/</tf:Identifier>
  <tf:Name>Georgia Tech Research Institute</tf:Name>
  <tf:Contact>
    <tf:Kind>PRIMARY</tf:Kind>
    <tf:Responder>George P. Burdell</tf:Responder>
    <tf:Email>TrustmarkFeedback@gtri.gatech.edu</tf:Email>
    <tf:Telephone>404-407-8956</tf:Telephone>
    <tf:PhysicalAddress>
      75 5th Street NW,
      Atlanta, GA 30308
    </tf:PhysicalAddress>
    <tf:MailingAddress>
      75 5th Street NW,
      Suite 900,
      Atlanta, GA 30308
    </tf:MailingAddress>
    <tf:WebsiteURL>http://trustmark.gtri.gatech.edu/</tf:WebsiteURL>
    <tf:Notes>The responder may change.</tf:Notes>
  </tf:Contact>
</tf:ProviderReference>
```

The third example illustrates the use of `ref` to refer to the `<ProviderReference>` in the preceding fragment.

```
<tf:ProviderReference tf:ref="providerReference" xsi:nil="true"/>
```

This convention permits a Trust Interoperability Profile to define an element of `EntityReferenceType` complex type once and refer to that element elsewhere in the Trust Interoperability Profile.

4.2.7 Complex Type `TrustmarkReferenceType`

`TrustmarkReferenceType` is a complex type that refers to an XML encoding of a Trustmark. Trustmark Status Reports employ `TrustmarkReferenceType` to reference Trustmarks for which they require only the identifier of the Trustmark, not its other metadata. The `TrustmarkReferenceType` includes the following elements.

`<Identifier>` [Required]

> `<Identifier>` is a URL; it is the globally unique Trustmark Identifier for the referenced Trustmark.

The following schema fragment defines the `TrustmarkReferenceType` complex type.

```
<xs:complexType name="TrustmarkReferenceType">
  <xs:sequence>
    <xs:element name="Identifier"
                type="xs:anyURI"
                minOccurs="1"
                maxOccurs="1"/>
  </xs:sequence>
```

```
</xs:complexType>
```

The following example XML fragment illustrates the `<TrustmarkReference>` element, which is of the `TrustmarkReference` complex type.

```
<tf:TrustmarkReference>
  <tf:Identifier>http://provider.example/trustmark/1</tf:Identifier>
</tf:TrustmarkReference>
```

4.2.8 Complex Type TrustmarkDefinitionReferenceType

`TrustmarkDefinitionReferenceType` is a complex type that refers to an XML encoding of a Trustmark Definition. Trustmarks and Trust Interoperability Profiles employ `TrustmarkDefinitionReferenceType` to reference Trustmark Definitions for which they require only the identifier of the Trustmark Definition, not its other metadata. The `TrustmarkDefinitionReferenceType` includes the following elements.

`<Identifier>` [Required]

> `<Identifier>` is a URL; it is the globally unique Trustmark Definition Identifier for the referenced Trustmark Definition.

`<Name>` [Optional]

> `<Name>` is a string; it is, if present, the name of the referenced Trustmark Definition. If the `<Name>` element is present, the value of the `<Name>` element SHOULD match the value of the `<Name>` element as it would appear in the `<TrustmarkDefinition>` element for this Trustmark Definition.

`<Version>` [Optional]

> `<Version>` is a string; it is, if present, the version of the referenced Trustmark Definition. If the `<Version>` element is present, the value of the `<Version>` element SHOULD match the value of the `<Version>` element as it would appear in the `<TrustmarkDefinition>` element for this Trustmark Definition.

`<Description>` [Optional]

> `<Description>` is a string; it is, if present, the description of the referenced Trustmark Definition. If the `<Description>` element is present, the value of the `<Description>` element SHOULD match the value of the `<Description>` element as it would appear in the `<TrustmarkDefinition>` element for this Trustmark Definition.

The following schema fragment defines the `TrustmarkDefinitionReferenceType` complex type.

```
<xs:complexType name="TrustmarkDefinitionReferenceType">
  <xs:sequence>
    <xs:element name="Identifier"
                type="xs:anyURI"
                minOccurs="1"
                maxOccurs="1"/>
    <xs:element name="Name"
                type="xs:string"
                minOccurs="0"
                maxOccurs="1"/>
    <xs:element name="Version"
                type="xs:string"
                minOccurs="0"
                maxOccurs="1"/>
    <xs:element name="Description"
                type="xs:string"
                minOccurs="0"
                maxOccurs="1"/>
  </xs:sequence>
```

```
</xs:complexType>
```

The following example XML fragments illustrate the <TrustmarkDefinitionReference> element, which is of the TrustmarkDefinitionReferenceType complex type. The first example illustrates the minimum content required by its definition.

```
<tf:TrustmarkDefinitionReference>
  <tf:Identifier>
https://trustmark.gtri.gatech.edu/operational-pilot/trustmark-definitions/ficam-privacy-activity-tracking-requirements-for-csps-and-bae-responders/1.0/
  </tf:Identifier>
</tf:TrustmarkDefinitionReference>
```

The second example illustrates additional content permitted by its definition. Additional content in this example appears in **boldface** text.

```
<tf:TrustmarkDefinitionReference>
  <tf:Identifier>
https://trustmark.gtri.gatech.edu/operational-pilot/trustmark-definitions/ficam-privacy-activity-tracking-requirements-for-csps-and-bae-responders/1.0/
  </tf:Identifier>
  <tf:Name>
    FICAM Privacy Activity Tracking Requirements
    for CSPs and BAE Responders
  </tf:Name>
  <tf:Version>1.0</tf:Version>
  <tf:Description>
    This TD adopts the LOA 2 and LOA 3 Privacy Minimalism trust criteria
    of the FICAM Trust Framework Provider Adoption Process.
  </tf:Description>
</tf:TrustmarkDefinitionReference>
```

4.2.9 Complex Type `TrustInteroperabilityProfileReferenceType`

TrustInteroperabilityProfileReferenceType is a complex type that refers to an XML encoding of a Trust Interoperability Profile. Trust Interoperability Profiles employ TrustInteroperabilityProfileReferenceType to reference Trust Interoperability Profiles for which they require only the identifier of the Trust Interoperability Profile, not its other metadata. The TrustInteroperabilityProfileReferenceType includes the following elements.

<Identifier> [Required]

> <Identifier> is a URL; it is the globally unique Trust Interoperability Profile Identifier of the referenced Trust Interoperability Profile.

<Name> [Optional]

> <Name> is a string; it is, if present, the name of the referenced Trust Interoperability Profile. If the <Name> element is present, the value of the <Name> element SHOULD match the value of the <Name> element as it would appear in the <TrustInteroperabilityProfile> element for the Trust Interoperability Profile to which the enclosing TrustInteroperabilityProfileReferenceType element refers.

<Version> [Optional]

> <Version> is a string; it is, if present, the version of the referenced Trust Interoperability Profile. If the <Version> element is present, the value of the <Version> element SHOULD match the value of the <Version> element as it would appear in the <TrustInteroperabilityProfile> element for

the Trust Interoperability Profile to which the enclosing TrustInteroperabilityProfileReferenceType element refers.

<Description> [Optional]

<Description> is a string; it is, if present, the description of the referenced Trust Interoperability Profile. If the <Description> element is present, the value of the <Description> element SHOULD match the value of the <Description> element as it would appear in the <TrustInteroperabilityProfile> element for the Trust Interoperability Profile to which the enclosing TrustInteroperabilityProfileReferenceType element refers.

id [Optional]

id is an XML ID for this reference to a Trust Interoperability Profile. This id permits the content of the <TrustExpression> element to refer to the Trust Interoperability Profile to which the enclosing TrustInteroperabilityProfileReferenceType element refers. When used in this context, this id MUST appear in the <TrustExpression> element of the Trust Interoperability Profile element within which the enclosing TrustInteroperabilityProfileReferenceType element appears.

The following schema fragment defines the TrustInteroperabilityProfileReferenceType complex type.

```
<xs:complexType name="TrustInteroperabilityProfileReferenceType">
  <xs:sequence>
    <xs:element name="Identifier"
                type="xs:anyURI"
                minOccurs="1"
                maxOccurs="1"/>
    <xs:element name="Name"
                type="xs:string"
                minOccurs="0"
                maxOccurs="1"/>
    <xs:element name="Version"
                type="xs:string"
                minOccurs="0"
                maxOccurs="1"/>
    <xs:element name="Description"
                type="xs:string"
                minOccurs="0"
                maxOccurs="1"/>
  </xs:sequence>
  <xs:attribute ref="tf:id"
                use="optional"/>
</xs:complexType>
```

The following example XML fragments illustrate the <TrustInteroperabilityProfileReference> element, which is of the TrustInteroperabilityProfileReferenceType complex type. The first example illustrates the minimum content required by its definition.

```
<tf:TrustInteroperabilityProfile tf:id="tip1">
  <tf:Identifier>https://trustmark.gtri.gatech.edu/operational-
pilot/trust-interoperability-profiles/ficam-privacy-tip-for-
csp/1.0/</tf:Identifier>
</tf:TrustInteroperabilityProfile>
```

The second example illustrates additional content permitted by its definition. Additional content in this example appears in **boldface** text.

```
<tf:TrustInteroperabilityProfile tf:id="tip1">
  <tf:Identifier>https://trustmark.gtri.gatech.edu/operational-
pilot/trust-interoperability-profiles/ficam-privacy-tip-for-
csp/1.0/</tf:Identifier>
```

```
  <tf:Name>FICAM Privacy TIP for CSP</tf:Name>
  <tf:Version>1.0</tf:Version>
  <tf:Description>
    This Trust Interoperability Profile specifies the FICAM mandated
    privacy requirements to be followed by all FICAM Credential Service
    Providers (CSPs).]]>
  </tf:Description>
</tf:TrustInteroperabilityProfile>
```

4.2.10 Complex Type `ExtensionType`

`ExtensionType` is a complex type that provides additional XML content for a Trustmark Status Report or Trustmark. The `ExtensionType` includes one or more of any XML element from any namespace.

The following schema fragment defines the complex type `ExtensionType`.

```
<xs:complexType name="ExtensionType">
  <xs:sequence>
    <xs:any minOccurs="1"
            maxOccurs="unbounded"
            processContents="skip"/>
  </xs:sequence>
</xs:complexType>
```

The following example XML fragment illustrates the `<Extension>` element, which is of the `ExtensionType` complex type. Note that this example fragment includes the definition of an extension namespace from which the extension schema is assumed to originate.

```
<tf:Extension xmlns:my="my-extension-namespace">
  <my:Element>
    This is additional XML content about the status of the Trustmark.
  </my:Element>
</tf:Extension>
```

4.2.11 Simple Type `ParameterIdentifierType`

The `ParameterIdentifierType` simple type constrains the value space of the `<ParameterIdentifier>` element of the `ParameterDefinitionType` complex type to values containing alphanumeric strings starting with alphabetic characters. It also permits underscores after the first character.

The following schema fragment defines the `ParameterIdentifierType` simple type.

```
<xs:simpleType name="ParameterIdentifierType">
  <xs:restriction base="xs:string">
    <xs:pattern value="[A-z][A-z0-9_]*"/>
  </xs:restriction>
</xs:simpleType>
```

For an example of the `ParameterIdentifierType` simple type, see the example for the `ParameterDefinitionType` complex type in Section 4.3.8.

4.2.12 Simple Type `ParameterKindType`

The `ParameterKindType` simple type constrains the value space of the `<ParameterKind>` element of the `ParameterDefinitionType` complex type to one of six values: `STRING`, `NUMBER`, `BOOLEAN`, `DATETIME`, `ENUM`, and `ENUM_MULTI`.

The following schema fragment defines the `ParameterKindType` simple type.

```
<xs:simpleType name="ParameterKindType">
  <xs:restriction base="xs:string">
```

```
    <xs:enumeration value="STRING"/>
    <xs:enumeration value="NUMBER"/>
    <xs:enumeration value="BOOLEAN"/>
    <xs:enumeration value="DATETIME"/>
    <xs:enumeration value="ENUM"/>
    <xs:enumeration value="ENUM_MULTI"/>
  </xs:restriction>
</xs:simpleType>
```

For an example of the `ParameterKindType` simple type, see the example for the `ParameterDefinitionType` complex type in Section 4.3.8.

4.2.13 Simple Type `EnumValueType`

The `EnumValueType` simple type constrains the value space of the `<EnumValue>` element within a Trustmark Definition to a text string not containing commas.

The following schema fragment defines the `EnumValueType` simple type.

```
<xs:simpleType name="EnumValueType">
  <xs:restriction base="xs:string">
    <xs:pattern value="[^,]*"/>
  </xs:restriction>
</xs:simpleType>
```

For an example of the `EnumValueType` simple type, see the example for the `ParameterDefinitionType` complex type in Section 4.3.8.

4.2.14 Complex Type `ParameterBindingType`

`ParameterBindingType` is a complex type that defines a parameter name/value binding within a Trustmark. The `ParameterBindingType` complex type contains two required attributes: `identifier`, which is a `ParameterIdentifierType` and corresponds to the parameter definition with the same parameter name as defined within the Trustmark Definition for the Trustmark, and `kind`, which is a `ParameterKindType` and provides the data type of the parameter referenced by the `identifier` attribute. The contents of the `ParameterBindingType` represent the value of the bound parameter.

The following schema fragment defines the complex type `ParameterBindingType`.

```
<xs:complexType name="ParameterBindingType">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute
        name="identifier" type="tf:ParameterIdentifierType" />
      <xs:attribute name="kind" type="tf:ParameterKindType" />
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

The following example XML fragments illustrate the `ParameterBindingType` complex type in conjunction with various values of `ParameterKindType`.

```
<tf:ParameterBinding
  identifier="min_pw_len"
  kind="NUMBER">8</tf:ParameberBinding>

<tf:ParameterBinding
  identifier="firewall_filter_policy_type"
  kind="ENUM">WHITELIST</tf:ParameberBinding>

<tf:ParameterBinding
```

```
identifier="sensitive_info_types"
kind="ENUM_MULTI">PII,CJI,HIPAA</tf:ParameberBinding>
```

4.3 Trustmark Definition

This section describes the XML encoding of a Trustmark Definition. As noted in Section 3, a Trustmark Definition specifies the conformance criteria that a Trustmark Recipient must meet and a formal assessment process that the Trustmark Provider must perform to assess whether the Trustmark Recipient qualifies for a Trustmark. The XML encoding of a Trustmark Definition employs the following schema components to encode this information, as well as additional metadata about the Trustmark Definition.

- `<TrustmarkDefinition>` element
- `AssessmentStepType` complex type
- `ConformanceCriterionType` complex type
- `CitationType` complex type
- `SourceType` complex type
- `TermType` complex type
- `ParameterDefinitionType` complex type
- `ParameterIdentifierType` simple type
- `ParameterKindType` simple type

The following sections describe the aforementioned schema components.

4.3.1 Element `<TrustmarkDefinition>`

The `<TrustmarkDefinition>` element is an XML encoding of a Trustmark Definition. It contains the following elements.

`<ds:Signature>` [Optional]

> `<ds:Signature>` is an optional XML signature that ensures the integrity and the authenticity of this Trustmark Definition. If included, the signature MUST conform to XML signature normative requirements as specified in [XML Sig]. In addition, if included, the signature must reference and pertain to the outermost XML element (`<TrustmarkDefinition>`) in the Trustmark Definition XML object. As such, if a `<ds:Signature>` element appears in the `<TrustmarkDefinition>` element, then the `<TrustmarkDefinition>` element MUST carry the `id` attribute.

`<Metadata>` [Required]

> `<Metadata>` is a container element for Trustmark Definition metadata. It contains the following elements.
>
> `<Identifier>` [Required]
>
> > `<Identifier>` is a URL; it is the globally unique Trustmark Definition Identifier for this Trustmark Definition. Section 5.1 contains normative language pertaining to the selection of Trustmark Definition Identifiers by Trustmark Defining Organizations.
>
> `<TrustmarkReferenceAttributeName>` [Required]
>
> > `<TrustmarkReferenceAttributeName>` is a URI; it is the name of the attribute that refers to a Trustmark issued under this Trustmark Definition.
>
> `<Name>` [Required]
>
> > `<Name>` is a string; it is the name of the Trustmark Definition.

`<Version>` [Required]

`<Version>` is a string; it is the version of the Trustmark Definition.

`<Description>` [Required]

`<Description>` is a string; it is the description of the Trustmark Definition.

`<PublicationDateTime>` [Required]

`<PublicationDateTime>` is the date and time at which the Trustmark Defining Organization published this Trustmark Definition.

`<TrustmarkDefiningOrganization>` [Required]

`<TrustmarkDefiningOrganization>` is an `EntityType`; it is the Trustmark Defining Organization Identifier for the Trustmark Defining Organization that defined and published this Trustmark Definition. Section 4.2.3 further describes the `EntityType` complex type. Section 5.1 contains normative language pertaining to the selection of Trustmark Defining Organization Identifiers.

`<TargetStakeholderDescription>` [Optional]

`<TargetStakeholderDescription>` is a string; it is an optional description of the intended communities and stakeholder groups to which the Trustmark Definition may apply.

`<TargetRecipientDescription>` [Optional]

`<TargetRecipientDescription>` is a string; it is an optional description of the intended organizations to which Trustmarks would be issued under this Trustmark Definition.

`<TargetRelyingPartyDescription>` [Optional]

`<TargetRelyingPartyDescription>` is a string; it is an optional description of the intended Trustmark Relying Parties for Trustmarks issued under this Trustmark Definition.

`<TargetProviderDescription>` [Optional]

`<TargetProviderDescription>` is a string; it is an optional description of the intended organizations that would act as Trustmark Providers and issue Trustmarks under this Trustmark Definition.

`<ProviderEligibilityCriteria>` [Optional]

`<ProviderEligibilityCriteria>` is a string; it is an optional description of the criteria that an organization must meet to become eligible to act as a Trustmark Provider and issue Trustmarks under this Trustmark Definition. If this element is absent, any organization may act as a Trustmark Provider and issue Trustmarks under this Trustmark Definition.

`<AssessorQualificationsDescription>` [Optional]

`<AssessorQualificationsDescription>` is a string; it is an optional description of the qualifications that an individual must possess to act as an assessor on behalf of a Trustmark Provider that issues Trustmarks under this Trustmark Definition. If this element is absent, any individual that is an employee or contractor for the Trustmark Provider may act as an assessor on behalf of a Trustmark Provider that issues Trustmarks under this Trustmark Definition.

`<TrustmarkRevocationCriteria>` [Optional]

`<TrustmarkRevocationCriteria>` is a string; it is an optional description of the criteria that, if triggered, would require that the Trustmark Provider revoke a Trustmark issued under this Trustmark Definition. If this element is absent, the Trustmark Provider must revoke a Trustmark issued under this Trustmark Definition upon discovery that the Trustmark

Recipient no longer fulfills one or more of the conformance criteria in this Trustmark Definition

`<ExtensionDescription>` [Optional]

`<ExtensionDescription>` is a string; it is an optional description of the normative requirements for populating the Extension element of a Trustmark issued under this Trustmark Definition.

`<LegalNotice>` [Optional]

`<LegalNotice>` is a string; it is a legal notice for this Trustmark Definition, if any.

`<Notes>` [Optional]

`<Notes>` is a string; it is additional optional text content about this Trustmark Definition.

`<Supersessions>` [Optional]

`<Supersessions>` is a container element for references to other Trustmark Definitions that either supersede this Trustmark Definition or have been superseded by this Trustmark Definition. If present, it contains the following elements.

`<Supersedes>` [Zero or More]

`<Supersedes>` is a `TrustmarkDefinitionReferenceType`; it is a reference to another Trustmark Definition that this Trustmark Definition supersedes or replaces. Multiple `<Supersedes>` elements can be included to indicate that this Trustmark Definition supersedes or replaces multiple other Trustmark Definitions.

`<SupersededBy>` [Zero or More]

`<SupersededBy>` is a `TrustmarkDefinitionReferenceType`; it is a reference to another Trustmark Definition that supersedes or replaces this Trustmark Definition. Multiple `<SupersededBy>` elements can be included to indicate that multiple other Trustmark Definitions supersede or replace this Trustmark Definition.

`<Deprecated>` [Optional]

`<Deprecated>` is an optional Boolean element that indicates whether this Trustmark Definition has been deprecated. Absence of this element is equivalent to the presence of this element with a value of `false` or `0`.

`<Satisfies>` [Zero or More]

`<Satisfies>` is a `TrustmarkDefinitionReferenceType`; it is a reference to another Trustmark Definition, and is intended to indicate that if an entity satisfies the conformance criteria for the enclosing Trustmark Definition, then the entity may also satisfy the conformance criteria for the referenced Trustmark Definition. This element can be used to indicate that there exists an object-oriented "subclass" or "is-a" relationship between the enclosing Trustmark Definition and the referenced Trustmark Definition. This can commonly occur when the referenced Trustmark Definition contains vague, high-level conformance criteria (e.g., "must use encryption to protect data") and the enclosing Trustmark Definition contains more specific conformance criteria (e.g., "must use AES 256-bit encryption to protect data").

In practice, this reference SHOULD be interpreted by a Trustmark Provider as a hint meaning that, if an entity satisfies the criteria for the enclosing Trustmark Definition, then the Trustmark Provider should also strongly consider assessing the entity for the referenced Trustmark Definition and granting a trustmark if the assessment indicates conformance.

`<Keywords>` [Optional]

<Keywords> is a container element for keywords that pertain to this Trustmark Definition. Keywords are intended to facilitate search and discovery of Trustmark Definitions within registries. If present, it contains the following elements.

<Keyword> [One or More]

<Keyword> is a string; it contains a keyword that pertains to this Trustmark Definition.

<Terms> [Optional]

<Terms> is a container element for Trustmark Definition terms. If present, it contains the following elements.

<Term> [One or More]

<Term> is a TermType; it is a term used by this Trustmark Definition. Section 4.3.7 further describes the TermType complex type.

<Sources> [Optional]

<Sources> is a container element for Trustmark Definition sources. If present, it contains the following elements.

<Source> [One or More]

<Source> is a SourceType; it is an authoritative source for a conformance criterion in the Trustmark Definition. The <Source> child element of the <Sources> element MUST NOT carry the ref attribute. Section 4.3.6 further describes the SourceType complex type.

<ConformanceCriteria> [Required]

<ConformanceCriteria> is a container element for Trustmark Definition conformance criteria. It contains the following elements.

<Preface> [Optional]

<Preface> is a string; it is optional prefatory text that applies to every conformance criterion.

<ConformanceCriterion> [One or More]

<ConformanceCriterion> is a ConformanceCriterionType; it is a conformance criterion: a normative requirement that a Trustmark Recipient must meet to receive a Trustmark issued under this Trustmark Definition. The <ConformanceCriterion> child element of the <ConformanceCriteria> element MUST carry the id attribute. Section 4.3.4 further describes the ConformanceCriterionType complex type.

<AssessmentSteps> [Required]

<AssessmentSteps> is a container element for Trustmark Definition assessment steps. It contains the following elements.

<Preface> [Optional]

<Preface> is a string; it is optional prefatory text that applies to every assessment step.

<AssessmentStep> [One or More]

<AssessmentStep> is an AssessmentStepType; it is an assessment step: a step in the formal assessment process that the Trustmark Provider must perform to assess whether a Trustmark Recipient qualifies for a Trustmark. Section 4.3.2 further describes the AssessmentStepType complex type.

<IssuanceCriteria> [Required]

<IssuanceCriteria> is a string; it is a Boolean expression that indicates whether a Trustmark Provider may issue a Trustmark to a Trustmark Recipient, based on the results of a formal assessment process.

If the issuance criteria evaluate to true, the Trustmark Provider MAY issue a Trustmark under this Trustmark Definition to the Trustmark Recipient; if the issuance criteria evaluate to false, the Trustmark Provider MUST NOT issue the Trustmark.

Appendix B describes the syntax and semantics of this Boolean expression.

id [Optional]

id is an XML ID for this Trustmark.

The following schema fragment defines the <TrustmarkDefinition> element.

```
<xs:element name="TrustmarkDefinition">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ds:Signature"
                  minOccurs="0"
                  maxOccurs="1"/>
      <xs:element name="Metadata"
                  minOccurs="1"
                  maxOccurs="1">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Identifier"
                        type="xs:anyURI"
                        minOccurs="1"
                        maxOccurs="1"/>
            <xs:element name="TrustmarkReferenceAttributeName"
                        type="xs:anyURI"
                        minOccurs="1"
                        maxOccurs="1"/>
            <xs:element name="Name"
                        type="xs:string"
                        minOccurs="1"
                        maxOccurs="1"/>
            <xs:element name="Version"
                        type="xs:string"
                        minOccurs="1"
                        maxOccurs="1"/>
            <xs:element name="Description"
                        type="xs:string"
                        minOccurs="1"
                        maxOccurs="1"/>
            <xs:element name="PublicationDateTime"
                        type="xs:dateTime"
                        minOccurs="1"
                        maxOccurs="1"/>
            <xs:element name="TrustmarkDefiningOrganization"
                        type="tf:EntityType"
                        minOccurs="1"
                        maxOccurs="1"/>
            <xs:element name="TargetStakeholderDescription"
                        type="xs:string"
                        minOccurs="0"
                        maxOccurs="1"/>
            <xs:element name="TargetRecipientDescription"
                        type="xs:string"
                        minOccurs="0"
                        maxOccurs="1"/>
            <xs:element name="TargetRelyingPartyDescription"
                        type="xs:string"
                        minOccurs="0"
                        maxOccurs="1"/>
            <xs:element name="TargetProviderDescription"
                        type="xs:string"
                        minOccurs="0"
                        maxOccurs="1"/>
            <xs:element name="ProviderEligibilityCriteria"
                        type="xs:string"
                        minOccurs="0"
                        maxOccurs="1"/>
            <xs:element name="AssessorQualificationsDescription"
                        type="xs:string"
                        minOccurs="0"
                        maxOccurs="1"/>
            <xs:element name="TrustmarkRevocationCriteria"
                        type="xs:string"
                        minOccurs="0"
                        maxOccurs="1"/>
            <xs:element name="ExtensionDescription"
                        type="xs:string"
                        minOccurs="0"
                        maxOccurs="1"/>
            <xs:element name="LegalNotice"
                        type="xs:string"
                        minOccurs="0"
                        maxOccurs="1"/>
            <xs:element name="Notes"
                        type="xs:string"
                        minOccurs="0"
                        maxOccurs="1"/>
            <xs:element name="Supersessions"
                        minOccurs="0"
                        maxOccurs="1">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="Supersedes"
                              type="tf:TrustmarkDefinitionReferenceType"
                              minOccurs="0"
                              maxOccurs="unbounded"/>
                  <xs:element name="SupersededBy"
                              type="tf:TrustmarkDefinitionReferenceType"
                              minOccurs="0"
                              maxOccurs="unbounded"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="Deprecated"
                        type="xs:boolean"
                        minOccurs="0"
                        maxOccurs="1"/>
            <xs:element name="Satisfies"
                        type="tf:TrustmarkDefinitionReferenceType"
                        minOccurs="0"
                        maxOccurs="unbounded">
            </xs:element>
            <xs:element name="Keywords"
                        minOccurs="0"
                        maxOccurs="1">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="Keyword"
                              type="xs:string"
                              minOccurs="1"
                              maxOccurs="unbounded"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Terms"
                  minOccurs="0"
                  maxOccurs="1">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Term"
                        type="tf:TermType"
                        minOccurs="1"
                        maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Sources"
                  minOccurs="0"
                  maxOccurs="1">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Source"
                        type="tf:SourceType"
                        minOccurs="1"
                        maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="ConformanceCriteria"
                  minOccurs="1"
                  maxOccurs="1">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Preface"
                        type="xs:string"
                        minOccurs="0"
                        maxOccurs="1"/>
            <xs:element name="ConformanceCriterion"
                        type="tf:ConformanceCriterionType"
                        minOccurs="1"
                        maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="AssessmentSteps"
                  minOccurs="1"
                  maxOccurs="1">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Preface"
                        type="xs:string"
                        minOccurs="0"
                        maxOccurs="1"/>
            <xs:element name="AssessmentStep"
                        type="tf:AssessmentStepType"
                        minOccurs="1"
```

```
                        maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="IssuanceCriteria"
                  type="xs:string"
                  minOccurs="1"
                  maxOccurs="1"/>
</xs:sequence>
    <xs:attribute ref="tf:id"
                  use="optional"/>
  </xs:complexType>
</xs:element>
```

The following example XML fragments illustrate the `<TrustmarkDefinition>` element. The first example illustrates the minimum content required by its definition.

```
<tf:TrustmarkDefinition
  xmlns:tf="https://trustmark.gtri.gatech.edu/specifications/trustmark-
framework/1.1/schema/"
  xmlns="http://www.w3.org/1999/xhtml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <tf:Metadata>
    <tf:Identifier>
https://trustmark.gtri.gatech.edu/operational-pilot/trustmark-
definitions/ficam-privacy-minimal-attribute-release-requirements-for-
csps-and-bae-responders/1.0/
    </tf:Identifier>
    <tf:TrustmarkReferenceAttributeName>
https://trustmark.gtri.gatech.edu/operational-pilot/trustmark-
definitions/ficam-privacy-minimal-attribute-release-requirements-for-
csps-and-bae-responders/1.0/trustmark-reference/
    </tf:TrustmarkReferenceAttributeName>
    <tf:Name>
      FICAM Privacy Minimal Attribute Release Requirements
      for CSPs and BAE Responders
    </tf:Name>
    <tf:Version>1.0</tf:Version>
    <tf:Description>
      This TD adopts the LOA 2 and LOA 3 Privacy Minimalism trust
      criteria of the FICAM Trust Framework Provider Adoption
      Process.
    </tf:Description>
    <tf:PublicationDateTime>2014-09-30T00:12:58</tf:PublicationDateTime>
    <tf:TrustmarkDefiningOrganization>
      <tf:Identifier>https://trustmark.gtri.gatech.edu/</tf:Identifier>
      <tf:Name>Georgia Tech Research Institute</tf:Name>
      <tf:Contact>
        <tf:Kind>PRIMARY</tf:Kind>
        <tf:Email>TrustmarkFeedback@gtri.gatech.edu</tf:Email>
      </tf:Contact>
    </tf:TrustmarkDefiningOrganization>
  </tf:Metadata>
  <tf:ConformanceCriteria>
    <tf:ConformanceCriterion tf:id="Minimalism">
      <tf:Number>1</tf:Number>
      <tf:Name>Transmission of Minimal Attributes</tf:Name>
      <tf:Description>
        <![CDATA[<p>The CSP or BAE Responder MUST transmit only those
attributes that were explicitly requested by an attribute
requester.</p>]]>
      </tf:Description>
    </tf:ConformanceCriterion>
  </tf:ConformanceCriteria>
  <tf:AssessmentSteps>
    <tf:AssessmentStep tf:id="MinimalismPolicy">
      <tf:Number>1</tf:Number>
      <tf:Name>Policy for minimalism requirements</tf:Name>
      <tf:Description>
        <![CDATA[<p>Does the trustmark applicant have established
policies or procedures that demonstrate that it transmits only those
attributes that were explicitly requested by an attribute requester?
Document references to sections of policies and procedures that
demonstrate conformance, and provide annotations that justify
conformance.</p>]]>
      </tf:Description>
      <tf:ConformanceCriterion tf:ref="Minimalism" xsi:nil="true"/>
    </tf:AssessmentStep>
  </tf:AssessmentSteps>
  <tf:IssuanceCriteria>yes(all)</tf:IssuanceCriteria>
</tf:TrustmarkDefinition>
```

The second example illustrates additional content permitted by its definition. Additional content in this example appears in **boldface** text.

```
<tf:TrustmarkDefinition
  xmlns:tf="https://trustmark.gtri.gatech.edu/specifications/trustmark-
framework/1.1/schema/"
  xmlns="http://www.w3.org/1999/xhtml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:Signature>
    <!-- The example omits the XML signature content. -->
  </ds:Signature>
  <tf:Metadata>
    <tf:Identifier>
https://trustmark.gtri.gatech.edu/operational-pilot/trustmark-
definitions/ficam-privacy-minimal-attribute-release-requirements-for-
csps-and-bae-responders/1.0/
    </tf:Identifier>
    <tf:TrustmarkReferenceAttributeName>
https://trustmark.gtri.gatech.edu/operational-pilot/trustmark-
definitions/ficam-privacy-minimal-attribute-release-requirements-for-
csps-and-bae-responders/1.0/trustmark-reference/
    </tf:TrustmarkReferenceAttributeName>
    <tf:Name>
      FICAM Privacy Minimal Attribute Release Requirements
      for CSPs and BAE Responders
    </tf:Name>
    <tf:Version>1.0</tf:Version>
    <tf:Description>
      This TD adopts the LOA 2 and LOA 3 Privacy Minimalism trust
      criteria of the FICAM Trust Framework Provider Adoption
      Process.
    </tf:Description>
    <tf:PublicationDateTime>2014-09-30T00:12:58</tf:PublicationDateTime>
    <tf:TrustmarkDefiningOrganization>
      <tf:Identifier>https://trustmark.gtri.gatech.edu/</tf:Identifier>
      <tf:Name>Georgia Tech Research Institute</tf:Name>
      <tf:Contact>
        <tf:Kind>PRIMARY</tf:Kind>
        <tf:Responder>George P. Burdell</tf:Responder>
        <tf:Email>TrustmarkFeedback@gtri.gatech.edu</tf:Email>
        <tf:Telephone>404-407-8956</tf:Telephone>
        <tf:PhysicalAddress>
          75 5th Street NW,
          Atlanta, GA 30308
        </tf:PhysicalAddress>
        <tf:MailingAddress>
          75 5th Street NW,
          Suite 900,
          Atlanta, GA 30308
        </tf:MailingAddress>
        <tf:WebsiteURL>http://trustmark.gtri.gatech.edu/</tf:WebsiteURL>
        <tf:Notes>The responder may change.</tf:Notes>
      </tf:Contact>
    </tf:TrustmarkDefiningOrganization>
    <tf:TargetStakeholderDescription>
      Stakeholders of the U.S. federal notions of levels of assurance as
      defined by the National Institute of Standards and Technology and
      the FICAM TFS.
    </tf:TargetStakeholderDescription>
    <tf:TargetRecipientDescription>
      Credential Service Providers that wish to assert identities to
      either federal relying parties or relying parties that utilize the
      U.S. federal notions of levels of assurance.
    </tf:TargetRecipientDescription>
    <tf:TargetRelyingPartyDescription>
      U.S. federal relying parties and other relying parties that utilize
      the U.S. federal notions of levels of assurance.
    </tf:TargetRelyingPartyDescription>
    <tf:TargetProviderDescription>
      Current or prospective FICAM-approved trust framework providers
      (TFPs), or Trustmark Providers that specialize in assessments
      concerning the U.S. federal notions of levels of assurance.
    </tf:TargetProviderDescription>
    <tf:ProviderEligibilityCriteria>
      <![CDATA[<div> <p>Any organization or business entity may act as a
Trustmark Provider for trustmarks under this Trustmark Definition.</p>
</div>]]>
    </tf:ProviderEligibilityCriteria>
    <tf:AssessorQualificationsDescription>
      <![CDATA[<div> <p>Any individual employed or contracted by the
Trustmark Provider may act as the assessor for trustmarks under this
Trustmark Definition.</p> </div>]]>
    </tf:AssessorQualificationsDescription>
    <tf:TrustmarkRevocationCriteria>
      <![CDATA[<div> <p>For any trustmark issued under this Trustmark
Definition, the Trustmark Provider must revoke the trustmark upon any
condition whereby one or more Conformance Criteria cease to be
satisfied.</p> </div>]]>
    </tf:TrustmarkRevocationCriteria>
    <tf:ExtensionDescription>
      <![CDATA[<div> <p>This TD requires no TDO defined extension
data.</p> </div>]]>
    </tf:ExtensionDescription>
    <tf:LegalNotice>
      This document and the information contained herein is provided on
      an "AS IS" basis, and the Georgia Tech ResearchInstitute disclaims
      all warranties, express or implied, including but not limited to
      any warranty that the use ofthe information herein will not
      infringe any rights or any implied warranties or merchantability or
      fitness for aparticular purpose. In addition, the Georgia Tech
      Research Institute disclaims legal liability for any loss
      incurredas a result of the use or reliance on the document or the
      information contained herein.
    </tf:LegalNotice>
    <tf:Notes>
      The Georgia Tech Research Institute (GTRI) has published this
      document with the support of the National Strategy forTrusted
      Identities in Cyberspace (NSTIC) via the National Institute of
      Standards and Technology (NIST). The viewsexpressed herein do not
      necessarily reflect the official policies of GTRI, NIST or NSTIC;
      nor does mention of tradenames, commercial practices, or
      organizations imply endorsement by the U.S. Government.
    </tf:Notes>
    <tf:Supersessions>
      <tf:Supersedes>
        <tf:Identifier>
https://trustmark.gtri.gatech.edu/operational-pilot/trustmark-
definitions/ficam-privacy-minimal-attribute-release-requirements-for-
csps-and-bae-responders/0.9/
        </tf:Identifier>
        <tf:Name>
          FICAM Privacy Minimal Attribute Release Requirements
          for CSPs and BAE Responders
        </tf:Name>
        <tf:Version>0.9</tf:Version>
      </tf:Supersedes>
      <tf:SupersededBy>
        <tf:Identifier>
https://trustmark.gtri.gatech.edu/operational-pilot/trustmark-
definitions/ficam-privacy-minimal-attribute-release-requirements-for-
csps-and-bae-responders/1.1/
        </tf:Identifier>
        <tf:Name>
          FICAM Privacy Minimal Attribute Release Requirements
          for CSPs and BAE Responders
        </tf:Name>
        <tf:Version>1.1</tf:Version>
      </tf:SupersededBy>
    </tf:Supersessions>
    <tf:Deprecated>true</tf:Deprecated>
    <tf:Keywords>
      <tf:Keyword>FICAM</tf:Keyword>
      <tf:Keyword>Privacy</tf:Keyword>
      <tf:Keyword>Attributes</tf:Keyword>
      <tf:Keyword>Minimalism</tf:Keyword>
    </tf:Keywords>
  </tf:Metadata>
  <tf:Terms>
    <tf:Term>
      <tf:Name>credential service provider</tf:Name>
      <tf:Abbreviation>CSP</tf:Abbreviation>
      <tf:Definition>
        <![CDATA[<p>A trusted entity that issues or registers subscriber
tokens and issues electronic credentials to subscribers. The CSP may
encompass registration authorities and verifiers that it operates. A CSP
may be an independent third party, or may issue credentials for its own
use.</p>]]>
      </tf:Definition>
    </tf:Term>
  <tf:Terms>
  <tf:Sources>
    <tf:Source tf:id="TFPAP_Privacy">
      <tf:Identifier>TFPAP Privacy</tf:Identifier>
      <tf:Reference>
```

```
        <![CDATA[<p>FICAM TFS Trust Framework Provider Adoption Process
(TFPAP) For All Levels of Assurance, v2.0.2. March, 14, 2014, Appendix A-
2, Privacy</p>]]>
      </tf:Reference>
    </tf:Source>
  </tf:Sources>
  <tf:ConformanceCriteria>
    <tf:Preface>
      <![CDATA[<p>The following are the normative requirements for this
Trustmark Definition.</p>]]>
    </tf:Preface>
    <tf:ConformanceCriterion tf:id="Minimalism">
      <tf:Number>1</tf:Number>
      <tf:Name>Transmission of Minimal Attributes</tf:Name>
      <tf:Description>
        <![CDATA[<p>The CSP or BAE Responder MUST transmit only those
attributes that were explicitly requested by an attribute
requester.</p>]]>
      </tf:Description>
      <tf:Citation>
        <tf:Source tf:ref="TFPAP_Privacy" xsi:nil="true"/>
        <tf:Description>
          <![CDATA[<em>Item 3</em>]]>
        </tf:Description>
      </tf:Citation>
    </tf:ConformanceCriterion>
  </tf:ConformanceCriteria>
  <tf:AssessmentSteps>
    <tf:Preface>
      <![CDATA[<p>For assessment steps that require the verification that
there are established policies or procedures that demonstrate conformance
to a particular conformance criterion, there is an implied requirement to
verify that there are no established policies or procedures that
demonstrate non-conformance to that criterion. This can be facilitated by
documenting references to sections of established policies and procedures
that explicitly enumerate the entirety of the relevant items that need to
be verified for conformance.</p> <p>To answer an assessment step as not
applicable, the assessor must document references to, and explanatory
annotations of, sections of established policies or procedures of the
Trustmark Applicant that provide supporting evidence of why the
assessment step should be answered as not applicable.</p>]]>
    </tf:Preface>
    <tf:AssessmentStep tf:id="MinimalismPolicy">
      <tf:Number>1</tf:Number>
      <tf:Name>Policy for minimalism requirements</tf:Name>
      <tf:Description>
        <![CDATA[<p>Does the trustmark applicant have established
policies or procedures that demonstrate that it transmits only those
attributes that were explicitly requested by an attribute requester?
Document references to sections of policies and procedures that
demonstrate conformance, and provide annotations that justify
conformance.</p>]]>
      </tf:Description>
      <tf:ConformanceCriterion tf:ref="Minimalism" xsi:nil="true"/>
      <tf:Artifact>
        <tf:Name>Annotated References</tf:Name>
        <tf:Description>
          <![CDATA[<p>Annotated references to the conforming policies and
procedures.</p>]]>
        </tf:Description>
      </tf:Artifact>
    </tf:AssessmentStep>
  </tf:AssessmentSteps>
  <tf:IssuanceCriteria>yes(all)</tf:IssuanceCriteria>
</tf:TrustmarkDefinition>
```

4.3.2 Complex Type AssessmentStepType

AssessmentStepType is a complex type that describes an assessment step: a step in the formal assessment process that the Trustmark Provider must perform to assess whether a Trustmark Recipient qualifies for a Trustmark. It includes a number for the step, a name for the step, a question, and an artifact. The answer to the question helps determine whether the Trustmark Recipient meets the conformance criteria for this Trustmark, and the artifact serves as evidence in support of the answer provided by the Trustmark Provider. The AssessmentStepType complex type includes the following elements.

<Number> [Required]

<Number> is a positive integer; it is the sequence number of this assessment step.

<Name> [Required]

<Name> is a string; it is a short descriptor of this assessment step.

<Description> [Required]

<Description> is a string; it is a question to the Trustmark Provider about conformance of the Trustmark Recipient to the referenced conformance criteria. The text of the question MUST be structured such that the answer is exactly one of "yes", "no", and "not applicable". This question MAY include instructions that the Trustmark Provider must follow when completing the assessment step.

<ConformanceCriterion> [One or More]

<ConformanceCriterion> is a ConformanceCriterionType; it is a reference to a conformance criterion that motivates this assessment step. The <ConformanceCriterion> child element of the <AssessmentStep> element MUST carry the ref attribute. Section 4.3.4 further describes the ConformanceCriterionType complex type.

<Artifact> [Any Number]

<Artifact> is an ArtifactType; it is an artifact that serves as evidence that the Trustmark Recipient meets the referenced conformance criteria of the Trustmark Definition. Section 4.3.3 further describes the ArtifactType complex type.

<ParameterDefinitions> [Optional]

<ParameterDefinitions> is an optional container element for assessment step parameter definitions. It contains the following elements.

<ParameterDefinition> [One or More]

<ParameterDefinition> is a ParameterDefinitionType; it defines a parameter for this assessment step. An assessment step parameter enables a Trustmark Provider to encode a specific, well-defined data attribute about a Trustmark Recipient in a machine-understandable format when issuing a Trustmark under this Trustmark Definition.

id [Required]

id is an XML ID for this assessment step. This id MUST appear in the <IssuanceCriteria> element. The value of the id attribute MUST NOT be ALL or NONE, as these are reserved words within the Issuance Criteria expression language used by the Trustmark Definition's <IssuanceCriteria> element. Appendix B describes the syntax and semantics of the contents of the <IssuanceCriteria> element.

The following schema fragment defines the AssessmentStepType complex type.

```
<xs:complexType name="AssessmentStepType">
  <xs:sequence>
    <xs:element name="Number"
                type="xs:positiveInteger"
                minOccurs="1"
                maxOccurs="1"/>
    <xs:element name="Name"
                type="xs:string"
                minOccurs="1"
                maxOccurs="1"/>
    <xs:element name="Description"
                type="xs:string"
                minOccurs="1"
                maxOccurs="1"/>
    <xs:element name="ConformanceCriterion"
                type="tf:ConformanceCriterionType"
                minOccurs="1"
                maxOccurs="unbounded"
                nillable="true"/>
    <xs:element name="Artifact"
                type="tf:ArtifactType"
                minOccurs="0"
                maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:element name="ParameterDefinitions"
              minOccurs="0"
              maxOccurs="1">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ParameterDefinition"
                    type="tf:ParameterDefinitionType"
                    minOccurs="1"
                    maxOccurs="unbounded">
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:attribute ref="tf:id" use="required"/>
</xs:complexType>
```

The following example XML fragments illustrate the `<AssessmentStep>` element, which is of the `AssesmentStepType` complex type. The first example illustrates the minimum content required by its definition.

```
<tf:AssessmentStep tf:id="MinimalismPolicy">
  <tf:Number>1</tf:Number>
  <tf:Name>Policy for minimalism requirements</tf:Name>
  <tf:Description>
        <![CDATA[<p>Does the trustmark applicant have established
policies or procedures that demonstrate that it transmits only those
attributes that were explicitly requested by an attribute requester?
Document references to sections of policies and procedures that
demonstrate conformance, and provide annotations that justify
conformance.</p>]]>
  </tf:Description>
  <tf:ConformanceCriterion tf:ref="Minimalism" xsi:nil="true"/>
</tf:AssessmentStep>
```

The second example illustrates additional content permitted by its definition. Additional content in this example appears in **boldface** text.

```
<tf:AssessmentStep tf:id="MinimalismPolicy">
  <tf:Number>1</tf:Number>
  <tf:Name>Policy for minimalism requirements</tf:Name>
  <tf:Description>
        <![CDATA[<p>Does the trustmark applicant have established
policies or procedures that demonstrate that it transmits only those
attributes that were explicitly requested by an attribute requester?
Document references to sections of policies and procedures that
demonstrate conformance, and provide annotations that justify
conformance.</p>]]>
  </tf:Description>
  <tf:ConformanceCriterion tf:ref="Minimalism" xsi:nil="true"/>
  <tf:Artifact>
    <tf:Name>Annotated References</tf:Name>
    <tf:Description>
      <![CDATA[<p>Annotated references to the conforming policies and
procedures.</p>]]>
    </tf:Description>
  </tf:Artifact>
</tf:AssessmentStep>
```

4.3.3 Complex Type `ArtifactType`

`ArtifactType` is a complex type that describes an artifact that serves as evidence that the Trustmark Recipient meets the conformance criteria of the Trustmark Definition. It includes the name and the description of the artifact. The `ArtifactType` complex type includes the following elements.

`<Name>` [Required]

> `<Name>` is a string; it is the name of the artifact that serves as evidence that the Trustmark Recipient meets a normative requirement of the Trustmark Definition.

`<Description>` [Required]

> `<Description>` is a string; it is the description of this artifact, including its content and its file type.

The following schema fragment defines the `ArtifactType` complex type.

```
<xs:complexType name="ArtifactType">
  <xs:sequence>
    <xs:element name="Name"
                type="xs:string"
                minOccurs="1"
                maxOccurs="1"/>
    <xs:element name="Description"
                type="xs:string"
                minOccurs="1"
                maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
```

The following example XML fragment illustrates the `<Artifact>` element, which is of the `ArtifactType` complex type.

```
<tf:Artifact>
  <tf:Name>Annotated References</tf:Name>
  <tf:Description>
      <![CDATA[<p>Annotated references to the conforming policies and
procedures.</p>]]>
  </tf:Description>
</tf:Artifact>
```

4.3.4 Complex Type `ConformanceCriterionType`

`ConformanceCriterionType` is a complex type that describes a conformance criterion: a normative requirement that a Trustmark Recipient must meet to receive a Trustmark issued under this Trustmark Definition. It includes a number for the requirement, a name for the requirement, a description of the requirement, and zero or more citations of the authoritative source(s) of the requirement. The `ConformanceCriterionType` complex type includes the following elements.

`<Number>` [Required]

> `<Number>` is a positive integer; it is the sequence number of this normative requirement.

`<Name>` [Required]

> `<Name>` is a string; it is a short descriptor for this normative requirement.

`<Description>` [Required]

> `<Description>` is a string; it contains a statement of the normative requirement that a Trustmark Recipient must meet to receive a Trustmark issued under this Trustmark Definition.

`<Citation>` [Any Number]

> `<Citation>` is a `CitationType`; it is a citation of an authoritative source of this normative requirement. Section 4.3.5 further describes the `CitationType` complex type.

`id` [Optional]

> `id` is an XML ID for this conformance criterion.

`ref` [Optional]

> `ref` is an XML IDREF to a conformance criterion.

The following schema fragment defines the `ConformanceCriterionType` complex type.

```
<xs:complexType name="ConformanceCriterionType">
  <xs:sequence>
    <xs:element name="Number"
                type="xs:positiveInteger"
                minOccurs="1"
                maxOccurs="1"/>
    <xs:element name="Name"
                type="xs:string"
                minOccurs="1"
                maxOccurs="1"/>
    <xs:element name="Description"
                type="xs:string"
                minOccurs="1"
                maxOccurs="1"/>
    <xs:element name="Citation"
                type="tf:CitationType"
                minOccurs="0"
                maxOccurs="unbounded"
                nillable="true"/>
  </xs:sequence>
  <xs:attribute ref="tf:id"/>
  <xs:attribute ref="tf:ref"/>
</xs:complexType>
```

The following example XML fragments illustrate the `<ConformanceCriterion>` element, which is of the `ConformanceCriterionType` complex type. The first example illustrates the minimum content required by its definition.

```
<tf:ConformanceCriterion tf:id="Minimalism">
  <tf:Number>1</tf:Number>
  <tf:Name>Transmission of Minimal Attributes</tf:Name>
  <tf:Description>
    <![CDATA[<p>The CSP or BAE Responder MUST transmit only those
attributes that were explicitly requested by an attribute
requester.</p>]]>
  </tf:Description>
</tf:ConformanceCriterion>
```

The second example illustrates additional content permitted by its definition. Additional content in this example appears in **boldface** text.

```
<tf:ConformanceCriterion tf:id="Minimalism">
  <tf:Number>1</tf:Number>
  <tf:Name>Transmission of Minimal Attributes</tf:Name>
  <tf:Description>
    <![CDATA[<p>The CSP or BAE Responder MUST transmit only those
attributes that were explicitly requested by an attribute
requester.</p>]]>
  </tf:Description>
  <tf:Citation>
    <tf:Source tf:ref="TFPAP_Privacy" xsi:nil="true"/>
    <tf:Description>
      <![CDATA[<em>Item 3</em>]]>
    </tf:Description>
  </tf:Citation>
</tf:ConformanceCriterion>
```

The third example illustrates the use of `ref` to refer to the `<ConformanceCriterion>` in the preceding fragment.

```
<tf:ConformanceCriterion tf:ref="Minimalism" xsi:nil="true"/>
```

This convention permits a Trustmark Definition to define an element of `ConformanceCriterionType` once and refer to that element elsewhere in the Trustmark Definition, as illustrated in the example for `AssessmentStepType`.

4.3.5 Complex Type `CitationType`

`CitationType` is a complex type that describes an authoritative source, or a part of an authoritative source, for a conformance criterion in a Trustmark Definition, including a reference to the authoritative source, an optional identifier for the part of the authoritative source, and a description of the location of the part in the authoritative source. The `CitationType` complex type includes the following elements.

`<Source>` [Required]

> `<Source>` is a `SourceType`; it is a reference to the authoritative source for a conformance criterion. The `<Source>` child element of the `<Citation>` element MUST carry the `ref` attribute. Section 4.3.6 further describes the `SourceType` complex type.

`<Description>` [Optional]

> `<Description>` is a string; it is the optional description of the location of the part in the authoritative source, such as a page number.

The following schema fragment defines the `CitationType` complex type.

```
<xs:complexType name="CitationType">
  <xs:sequence>
    <xs:element name="Source"
                type="tf:SourceType"
                minOccurs="1"
                maxOccurs="1"
                nillable="true"/>
<xs:element name="Description"
                type="xs:string"
                minOccurs="0"
                maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
```

The following example XML fragments illustrate the `<Citation>` element, which is of the `CitationType` complex type. The first example illustrates the minimum content required by its definition.

```
<tf:Citation>
  <tf:Source tf:ref="TFPAP_Privacy" xsi:nil="true"/>
</tf:Citation>
```

The second example illustrates additional content permitted by its definition. Additional content in this example appears in **boldface** text.

```
<tf:Citation>
  <tf:Source tf:ref="TFPAP_Privacy" xsi:nil="true"/>
  <tf:Description>
    <![CDATA[<em>Item 3</em>]]>
  </tf:Description>
</tf:Citation>
```

4.3.6 Complex Type `SourceType`

`SourceType` is a complex type that describes the authoritative source for a conformance criterion in the Trustmark Definition, including the identifier and the bibliographic information for the authoritative source. The `SourceType` complex type includes the following elements.

`<Identifier>` [Required]

> `<Identifier>` is a string; it is an identifier for the authoritative source. The text of the Trustmark Definition, such as the conformance criteria and the assessment steps, may use this identifier to refer to the source itself.

`<Reference>` [Required]

> `<Reference>` is a string; it is the bibliographic information for the authoritative source.

`id` [Optional]

> `id` is an XML ID for this source.

`ref` [Optional]

> `ref` is an XML IDREF to a source.

The following schema fragment defines the `SourceType` complex type.

```
<xs:complexType name="SourceType">
  <xs:sequence>
    <xs:element name="Identifier"
                type="xs:string"
                minOccurs="1"
                maxOccurs="1"/>
    <xs:element name="Reference"
                type="xs:string"
                minOccurs="1"
                maxOccurs="1"/>
  </xs:sequence>
  <xs:attribute ref="tf:id"/>
  <xs:attribute ref="tf:ref"/>
</xs:complexType>
```

The following example XML fragments illustrate the `<Source>` element, which is of the `SourceType` complex type. The first example illustrates the minimum content required by its definition.

```
<tf:Source>
  <tf:Identifier>TFPAP Privacy</tf:Identifier>
  <tf:Reference>
      <![CDATA[<p>FICAM TFS Trust Framework Provider Adoption Process
(TFPAP) For All Levels of Assurance, v2.0.2. March, 14, 2014, Appendix A-
2, Privacy</p>]]>
  </tf:Reference>
</tf:Source>
```

The second example illustrates additional content permitted by its definition. Additional content in this example appears in **boldface** text.

```
<tf:Source tf:id="TFPAP_Privacy">
  <tf:Identifier>TFPAP Privacy</tf:Identifier>
  <tf:Reference>
      <![CDATA[<p>FICAM TFS Trust Framework Provider Adoption Process
(TFPAP) For All Levels of Assurance, v2.0.2. March, 14, 2014, Appendix A-
2, Privacy</p>]]>
  </tf:Reference>
</tf:Source>
```

The third example illustrates the use of `ref` to refer to the `<Source>` in the preceding fragment.

```
<tf:Source tf:ref="TFPAP_Privacy" xsi:nil="true"/>
```

This convention permits a Trustmark Definition to define an element of the `SourceType` complex type once and refer to that element elsewhere in the Trustmark Definition, as illustrated in the example for `CitationType`.

4.3.7 Complex Type `TermType`

`TermType` is a complex type that describes a term used by a Trustmark Definition, including the name of the term, any abbreviations for the term, and the meaning of the term within the context of the Trustmark Definition. The `TermType` complex type includes the following elements.

`<Name>` [Required]

`<Name>` is a string; it is the name of this term.

`<Abbreviation>` [Any Number]

`<Abbreviation>` is a string; it is a case-sensitive abbreviation of this term.

`<Definition>` [Required]

`<Definition>` is a string; it is the meaning of this term in the context of this Trustmark Definition.

The following schema fragment defines the `TermType` complex type.

```
<xs:complexType name="TermType">
  <xs:sequence>
    <xs:element name="Name"
                type="xs:string"
                minOccurs="1"
```

```
                 maxOccurs="1"/>
    <xs:element name="Abbreviation"
                 type="xs:string"
                 minOccurs="0"
                 maxOccurs="unbounded"/>
    <xs:element name="Definition"
                 type="xs:string"
                 minOccurs="1"
                 maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
```

The following example XML fragments illustrate the `<Term>` element, which is of the `TermType` complex type. The first example illustrates the minimum content required by its definition.

```
<tf:Term>
  <tf:Name>personally identifiable information</tf:Name>
  <tf:Definition>
    <![CDATA[<p>Information which can be used to distinguish or trace an
individual's identity, such as their name, social security number,
biometric records, etc. alone, or when combined with other personal or
identifying information which is linked or linkable to a specific
individual, such as date and place of birth, mother's maiden name,
etc.</p>]]>
  </tf:Definition>
</tf:Term>
```

The second example illustrates additional content permitted by its definition. Additional content in this example appears in **boldface** text.

```
<tf:Term>
  <tf:Name>personally identifiable information</tf:Name>
  <tf:Abbreviation>PII</tf:Abbreviation>
  <tf:Definition>
    <![CDATA[<p>Information which can be used to distinguish or trace an
individual's identity, such as their name, social security number,
biometric records, etc. alone, or when combined with other personal or
identifying information which is linked or linkable to a specific
individual, such as date and place of birth, mother's maiden name,
etc.</p>]]>
  </tf:Definition>
</tf:Term>
```

4.3.8 Complex Type `ParameterDefinitionType`

`ParameterDefinitionType` is a complex type that defines a parameter within a Trustmark Definition. The `ParameterDefinitionType` complex type includes the following elements.

`<Identifier>` [Required]

> `<Identifier>` is a `ParameterIdentifierType`; it indicates the machine-readable identifier of this parameter. Section 4.2.11 further describes the `ParameterIdentifierType` simple type.
>
> Note that `<Identifier>` must be unique, not only within an `<AssessmentStep>`, but also across the entire `<TrustmarkDefinition>` within which it appears.

`<Name>` [Required]

> `<Name>` is a string; it is the human-readable name of the parameter defined by the `ParameterDefinitionType` object.

`<Description>` [Required]

<Description> is a string; it is a human-readable description of the parameter defined by the ParameterDefinitionType object.

<ParameterKind> [Required]

<ParameterKind> is a ParameterKindType; it indicates the data type of this parameter. Section 4.2.12 further describes the ParameterKindType simple type.

<EnumValues> [Optional]

<EnumValues> is a container element for an enumerated list of data type values. It is required only if <ParameterKind> indicates that this parameter is an ENUM (i.e., an enumerated data type) or an ENUM_MULTI (i.e., an enumerated data type that permits multi-valued parameter values). It contains the following elements.

<EnumValue> [One or More]

<EnumValue> is an EnumValueType; it names one value for an enumerated data type.

<Required> [Required]

<Required> is a Boolean; it indicates whether a Trustmark is required to contain a value for this parameter.

The following schema fragment defines the complex type ParameterDefinitionType.

```
<xs:complexType name="ParameterDefinitionType">
  <xs:sequence>
    <xs:element name="Identifier"
                type="tf:ParamaterIdentifierType"
                minOccurs="1"
                maxOccurs="1"/>
    <xs:element name="Name"
                type="xs:string"
                minOccurs="1"
                maxOccurs="1"/>
    <xs:element name="Description"
                type="xs:string"
                minOccurs="1"
                maxOccurs="1"/>
    <xs:element name="ParameterKind"
                type="tf:ParameterKindType"
                minOccurs="1"
                maxOccurs="1"/>
    <xs:element name="EnumValues"
                minOccurs="0"
                maxOccurs="1">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="EnumValue"
                      type="tf:EnumValueType"
                      minOccurs="1"
                      maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="Required"
                type="xs:boolean"
                minOccurs="1"
                maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
```

The following example XML fragments illustrate the `ParameterDefinitionType` complex type. The first example illustrates the minimum content required by its definition.

```
<tf:ParameterDefinition>
  <tf:Identifier>min_pw_len</tf:Identifier>
  <tf:Name>Minimum Password Length</tf:Name>
  <tf:Description>Minimum required password length</tf:Description>
  <tf:ParameterKind>NUMBER</tf:ParameterKind>
  <tf:Required>true</tf:Required>
</tf:ParameterDefinition>
```

The second example illustrates additional content permitted by its definition.

```
<tf:ParameterDefinition>
  <tf:Identifier>SSO_Platform</tf:Identifier>
  <tf:Name>Single Sign-On Platform</tf:Name>
  <tf:Description>Software platform for the SSO system</tf:Description>
  <tf:ParameterKind>ENUM</tf:ParameterKind>
  <tf:EnumValues>
    <tf:EnumValue>Shibboleth</tf:EnumValue>
    <tf:EnumValue>PingFederate</tf:EnumValue>
    <tf:EnumValue>DataPower</tf:EnumValue>
    <tf:EnumValue>Other</tf:EnumValue>
  </tf:EnumValues>
  <tf:Required>false</tf:Required>
</tf:ParameterDefinition>
```

4.4 Trustmark

This section describes the XML encoding of a Trustmark. As noted in Section 3, a Trustmark is a machine-readable, cryptographically signed digital artifact that represents a statement of conformance to a well scoped set of trust and interoperability requirements. The XML encoding of a Trustmark employs the `<Trustmark>` element schema component to encode this information.

4.4.1 Element `<Trustmark>`

The `<Trustmark>` element is an XML encoding of a Trustmark. It includes the following elements.

`<ds:Signature>` [Optional]

> `<ds:Signature>` is an XML signature that ensures the integrity and the authenticity of this Trustmark. The signature MUST conform to XML signature normative requirements as specified in [XML Sig]. In addition, if included, the signature must reference and pertain to the outermost XML element (`<Trustmark>`) in the Trustmark XML object. As such, if a `<ds:Signature>` element appears in the `<Trustmark>` element, then the `<Trustmark>` element MUST carry the `id` attribute.

`<Identifier>` [Required]

> `<Identifier>` is a URL; it is the globally unique Trustmark Identifier for this Trustmark. Section 5.3 further describes the normative requirements pertaining to Trustmark Identifiers.

`<TrustmarkDefinitionReference>` [Required]

> `<TrustmarkDefinitionReference>` is a `TrustmarkDefinitionReferenceType`; it is a reference to the Trustmark Definition under which this Trustmark was issued.

`<IssueDateTime>` [Required]

> `<IssueDateTime>` is the date and time at which the Trustmark Provider issued this Trustmark to the Trustmark Recipient.

`<ExpirationDateTime>` [Required]

<ExpirationDateTime> is the date and time at which this Trustmark expires.

<PolicyURL> [Required]

<PolicyURL> is the URL of the Trustmark Policy under which this Trustmark was issued.

<RelyingPartyAgreementURL> [Required]

<RelyingPartyAgreementURL>is the URL of the Trustmark Relying Party Agreement under which this Trustmark was issued.

<StatusURL> [Required]

<StatusURL> is the URL at which the Trustmark Relying Party may query the Trustmark Provider for the current status of this Trustmark.

<Provider> [Required]

<Provider> is an EntityType; it is Trustmark Provider that issued this Trustmark. Note that the content of the <Identifier> of this EntityType is the Trustmark Provider Identifier, as described in Section 5.2.1.

<Recipient> [Required]

<Recipient> is an EntityType; it is the Trustmark Recipient to which and about which this Trustmark was issued. Note that the content of the <Identifier> of this EntityType is the Trustmark Recipient Identifier, as described in Section 5.2.4.

<ExceptionInfo> [Zero or More]

<ExceptionInfo> is an optional string that provides information about any exceptions that pertain to this Trustmark. The presence of one or more of these elements within a Trustmark indicates that one or more exceptions exist, and absence of this element indicates that no exceptions exist. An exception indicates that, while the Trustmark Provider believes that the Trustmark Recipient satisfies the overall spirit and intent of the Trustmark Definition under which this Trustmark was issued, there exist one or more conformance criteria within the Trustmark Definition to which the Trustmark Recipient does not fully conform. If present, this element contains descriptive text about the exception(s) pertaining to this Trustmark. Note that, while the contents of this element are not expected to be machine-interpretable, the presence or absence of this element can be used as part of a trust decision about a Trustmark. When using this element, a Trustmark Provider SHOULD use a separate <ExceptionInfo> element for each individual exception reported within the Trustmark.

<ParameterBindings> [Optional]

<ParameterBindings> is a container element for a set of parameter binding objects that indicate specific parameter names and values bound to the Trustmark. It contains the following elements.

<ParameterBinding> [One or More]

<ParameterBinding> is a ParameterBindingType; it indicates a specific parameter name and value that are bound to the Trustmark, as well as the parameter's "kind" (i.e., its data type). Section 4.2.14 further describes the ParameterBindingType complex type.

<DefinitionExtension> [Optional]

<DefinitionExtension> is an ExtensionType; it is additional XML content about this Trustmark, as normatively defined by the Trustmark Definition under which this Trustmark was issued.

<ProviderExtension> [Optional]

<ProviderExtension> is an ExtensionType; it is additional XML content about this Trustmark, provided at the discretion of the Trustmark Provider that issued this Trustmark.

id [Required]

`id` is an XML ID for this Trustmark.

The following schema fragment defines the `<Trustmark>` element.

```
<xs:element name="Trustmark">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ds:Signature"
                  minOccurs="0"
                  maxOccurs="1"/>
      <xs:element name="Identifier"
                  type="xs:anyURI"
                  minOccurs="1"
                  maxOccurs="1"/>
      <xs:element name="TrustmarkDefinitionReference"
                  type="tf:TrustmarkDefinitionReferenceType"
                  minOccurs="1"
                  maxOccurs="1"/>
      <xs:element name="IssueDateTime"
                  type="xs:dateTime"
                  minOccurs="1"
                  maxOccurs="1"/>
      <xs:element name="ExpirationDateTime"
                  minOccurs="1"
                  maxOccurs="1"
                  type="xs:dateTime"/>
      <xs:element name="PolicyURL"
                  minOccurs="1"
                  maxOccurs="1"
                  type="xs:anyURI"/>
      <xs:element name="RelyingPartyAgreementURL"
                  minOccurs="1"
                  maxOccurs="1"
                  type="xs:anyURI"/>
      <xs:element name="StatusURL"
                  type="xs:anyURI"
                  minOccurs="1"
                  maxOccurs="1"/>
      <xs:element name="Provider"
                  type="tf:EntityType"
                  minOccurs="1"
                  maxOccurs="1"/>
      <xs:element name="Recipient"
                  type="tf:EntityType"
                  minOccurs="1"
                  maxOccurs="1"/>
      <xs:element name="ExceptionInfo"
                  type="xs:string"
                  minOccurs="0"
                  maxOccurs="1"/>
      <xs:element name="ParameterBindings"
                  minOccurs="0"
                  maxOccurs="1">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ParameterBinding"
                        type="tf:ParameterBindingType"
                        minOccurs="1"
                        maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="DefinitionExtension"
```

```
                  type="tf:ExtensionType"
                  minOccurs="0"
                  maxOccurs="1"/>
      <xs:element name="ProviderExtension"
                  type="tf:ExtensionType"
                  minOccurs="0"
                  maxOccurs="1"/>
    </xs:sequence>
    <xs:attribute ref="tf:id"
                  use="required"/>
  </xs:complexType>
</xs:element>
```

The following example XML fragments illustrate the `<Trustmark>` element. The first example illustrates the minimum content required by its definition.

```
<tf:Trustmark tf:id="trustmark"
  xmlns:tf="https://trustmark.gtri.gatech.edu/specifications/trustmark-
framework/1.1/schema/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <tf:Identifier>https://provider.example/trustmark/1</tf:Identifier>
  <tf:TrustmarkDefinitionReference>
    <tf:Identifier>
https://trustmark.gtri.gatech.edu/operational-pilot/trustmark-
definitions/ficam-privacy-minimal-attribute-release-requirements-for-
csps-and-bae-responders/1.0/
    </tf:Identifier>
  </tf:TrustmarkDefinitionReference>
  <tf:IssueDateTime>2014-01-01T00:00:00</tf:IssueDateTime>
  <tf:ExpirationDateTime>20015-01-01T00:00:00</tf:ExpirationDateTime>
  <tf:PolicyURL>https://provider.example/policy</tf:PolicyURL>
  <tf:RelyingPartyAgreementURL>
https://provider.example/relying-party-agreement
  </tf:RelyingPartyAgreementURL>
  <tf:StatusURL>
https://provider.example/trustmark/1/status
  </tf:StatusURL>
  <tf:Provider>
    <tf:Identifier>https://provider.example/</tf:Identifier>
    <tf:Name>Example Provider</tf:Name>
    <tf:Contact>
      <tf:Kind>PRIMARY</tf:Kind>
      <tf:Email>contact@provider.example</tf:Email>
    </tf:Contact>
  </tf:Provider>
  <tf:Recipient>
    <tf:Identifier>https://recipient.example</tf:Identifier>
    <tf:Name>Example Recipient</tf:Name>
    <tf:Contact>
      <tf:Kind>PRIMARY</tf:Kind>
      <tf:Email>contact@recipient.example</tf:Email>
    </tf:Contact>
  </tf:Recipient>
</tf:Trustmark>
```

The second example illustrates additional content permitted by its definition. Additional content in this example appears in **boldface** text.

```
<tf:Trustmark tf:id="trustmark"
  xmlns:tf="https://trustmark.gtri.gatech.edu/specifications/trustmark-
framework/1.1/schema/"
```

```
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:my="https://my.example/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:Signature>
    <!-- The example omits the XML signature content. -->
  </ds:Signature>
  <tf:Identifier>https://provider.example/trustmark/1</tf:Identifier>
  <tf:TrustmarkDefinitionReference>
    <tf:Identifier>
https://trustmark.gtri.gatech.edu/operational-pilot/trustmark-
definitions/ficam-privacy-minimal-attribute-release-requirements-for-
csps-and-bae-responders/1.0/
    </tf:Identifier>
    <tf:Name>
      FICAM Privacy Minimal Attribute Release Requirements
      for CSPs and BAE Responders
    </tf:Name>
    <tf:Version>1.0</tf:Version>
    <tf:Description>
      This TD adopts the LOA 2 and LOA 3 Privacy Minimalism trust
      criteria of the FICAM Trust Framework Provider Adoption
      Process.
    </tf:Description>
  </tf:TrustmarkDefinitionReference>
  <tf:IssueDateTime>2014-01-01T00:00:00</tf:IssueDateTime>
  <tf:ExpirationDateTime>20015-01-01T00:00:00</tf:ExpirationDateTime>
  <tf:PolicyURL>https://provider.example/policy</tf:PolicyURL>
  <tf:RelyingPartyAgreementURL>
https://provider.example/relying-party-agreement
  </tf:RelyingPartyAgreementURL>
  <tf:StatusURL>
https://provider.example/trustmark/1/status
  </tf:StatusURL>
  <tf:Provider>
    <tf:Identifier>https://provider.example/</tf:Identifier>
    <tf:Name>Example Provider</tf:Name>
    <tf:Contact>
      <tf:Kind>PRIMARY</tf:Kind>
      <tf:Responder>George P. Burdell</tf:Responder>
      <tf:Email>contact@provider.example</tf:Email>
      <tf:Telephone>404-555-1234</tf:Telephone>
      <tf:PhysicalAddress>
        123 Main Street
        Somewhere, GA 30000
      </tf:PhysicalAddress>
      <tf:MailingAddress>
        123 Main Street
        Somewhere, GA 30000
      </tf:MailingAddress>
      <tf:WebsiteURL>https://provider.example</tf:WebsiteURL>
      <tf:Notes>The responder may change.</tf:Notes>
    </tf:Contact>
  </tf:Provider>
  <tf:Recipient>
    <tf:Identifier>https://recipient.example</tf:Identifier>
    <tf:Name>Example Recipient</tf:Name>
    <tf:Contact>
      <tf:Kind>PRIMARY</tf:Kind>
      <tf:Responder>Burdell P. George</tf:Responder>
      <tf:Email>contact@recipient.example</tf:Email>
      <tf:Telephone>404-555-4321</tf:Telephone>
      <tf:PhysicalAddress>
        456 Main Street
        Somewhere, GA 30000
      </tf:PhysicalAddress>
      <tf:MailingAddress>
        456 Main Street
        Somewhere, GA 30000
      </tf:MailingAddress>
      <tf:WebsiteURL>https://recipient.example/</tf:WebsiteURL>
      <tf:Notes>The responder may change.</tf:Notes>
    </tf:Contact>
  </tf:Recipient>
  <tf:ExceptionInfo>
[Descriptive text about exceptions (if any) goes here.]
  </tf:ExceptionInfo>
<tf:DefinitionExtension>
    <my:DefinitionExtension>
      This is additional XML content about the Trustmark, as required by
      the Trustmark Definition.
    </my:DefinitionExtension>
  </tf:DefinitionExtension>
  <tf:ProviderExtension>
    <my:ProviderExtension>
      This is additional XML content about the Trustmark, as provided by
      the Trustmark Provider.
    </my:ProviderExtension>
  </tf:ProviderExtension>
</tf:Trustmark>
```

4.5 Trustmark Status Report

This section describes the XML encoding of a Trustmark Status Report. As noted in Section 3, a Trustmark Status Report provides status information about a Trustmark, and can be updated as needed if the status of the Trustmark changes from "active" to "revoked" or "expired". The XML encoding of a Trustmark Status Report employs the following schema components to encode this status information.

- <TrustmarkStatusReport> element
- TrustmarkStatusCodeType simple type

The following sections describe the aforementioned schema components.

4.5.1 Element <TrustmarkStatusReport>

The <TrustmarkStatusReport> element is an XML encoding of a Trustmark Status Report. It includes the following elements.

<ds:Signature> [Optional]

> <ds:Signature> is an optional XML signature that protects the integrity and authenticity of this Trustmark Status Report. If included, the signature MUST conform to XML signature normative requirements as specified in [XML Sig]. In addition, if included, the signature must reference and pertain to the outermost XML element (<TrustmarkStatusReport>) in the Trustmark Status Report XML object. As such, if a <ds:Signature> element appears in the <TrustmarkStatusReport> element, then the <TrustmarkStatusReport> element MUST carry the id attribute.

<TrustmarkReference> [Required]

> <TrustmarkReference> is a TrustmarkReferenceType; it is the reference to the Trustmark that is the subject of this Trustmark Status Report.

<StatusCode> [Required]

> `<StatusCode>` is a `TrustmarkStatusCodeType` that is current status code of the Trustmark indicated by `<TrustmarkReference>`.

`<StatusDateTime>` [Required]

> `<StatusDateTime>` date and time at which the Trustmark Provider published this Trustmark Status Report.

`<SupersederTrustmarkReference>` [Any Number]

> `<SupersederTrustmarkReference>` is a `TrustmarkReferenceType`; it is a reference to a Trustmark that supersedes the Trustmark indicated by `<TrustmarkReference>`, if any.

`<Notes>` [Optional]

> `<Notes>` is a string; it is additional text content about the status of the Trustmark indicated by `<TrustmarkReference>`, if any.

`<Extension>` [Optional]

> `<Extension>` is an `ExtensionType`; it is additional XML content about the status of the Trustmark indicated by `<TrustmarkReference>`, if any.

`id` [Optional]

> `id` is an XML ID for this Trustmark.

The following schema fragment defines the `<TrustmarkStatusReport>` element.

```
<xs:element name="TrustmarkStatusReport">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ds:Signature"
                  minOccurs="0"
                  maxOccurs="1"/>
      <xs:element name="TrustmarkReference"
                  type="tf:TrustmarkReferenceType"
                  minOccurs="1"
                  maxOccurs="1"/>
      <xs:element name="StatusCode"
                  type="tf:TrustmarkStatusCodeType"
                  minOccurs="1"
                  maxOccurs="1"/>
      <xs:element name="StatusDateTime"
                  type="xs:dateTime"
                  minOccurs="1"
                  maxOccurs="1"/>
      <xs:element name="SupersederTrustmarkReference"
                  type="tf:TrustmarkReferenceType"
                  minOccurs="0"
                  maxOccurs="unbounded"/>
      <xs:element name="Notes"
                  type="xs:string"
                  minOccurs="0"
                  maxOccurs="1"/>
      <xs:element name="Extension"
                  type="tf:ExtensionType"
                  minOccurs="0"
                  maxOccurs="1"/>
    </xs:sequence>
    <xs:attribute ref="tf:id"
                  use="optional"/>
  </xs:complexType>
</xs:element>
```

The following example XML fragments illustrate the `<TrustmarkStatusReport>`. The first example illustrates the minimum content required by its definition.

```
<tf:TrustmarkStatusReport
  xmlns:tf="https://trustmark.gtri.gatech.edu/specifications/trustmark-
framework/1.1/schema/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <tf:TrustmarkReference>
    <tf:Identifier>https://provider.example/trustmark/1</tf:Identifier>
  </tf:TrustmarkReference>
  <tf:StatusCode>ACTIVE</tf:StatusCode>
  <tf:StatusDateTime>2014-01-01T00:00:00</tf:StatusDateTime>
</tf:TrustmarkStatusReport>
```

This second example illustrates additional content permitted by its definition. Additional content in this example appears in **boldface** text.

```
<tf:TrustmarkStatusReport tf:id="tsr1"
  xmlns:tf="https://trustmark.gtri.gatech.edu/specifications/trustmark-
framework/1.1/schema/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:my="https://my.example/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:Signature>
    <!-- The example omits the XML signature content. -->
  </ds:Signature>
  <tf:TrustmarkReference>
    <tf:Identifier>https://provider.example/trustmark/1</tf:Identifier>
  </tf:TrustmarkReference>
  <tf:StatusCode>EXPIRED</tf:StatusCode>
  <tf:StatusDateTime>2014-01-01T00:00:00</tf:StatusDateTime>
  <tf:SupersederTrustmarkReference>
    <tf:Identifier>https://provider.example/trustmark/0</tf:Identifier>
  </tf:SupersederTrustmarkReference>
  <tf:Notes>
    Notes.
  </tf:Notes>
  <tf:Extension>
    <my:Extension>
      This is additional XML content about the status of the Trustmark.
    </my:Extension>
  </tf:Extension>
</tf:TrustmarkStatusReport>
```

4.5.2 Simple Type `TrustmarkStatusCodeType`

The `TrustmarkStatusCodeType` simple type constrains the value space of the `<StatusCode>` child element of the `<TrustmarkStatusReport>` element to one of the following string values: `ACTIVE`, `REVOKED`, and `EXPIRED`.

The following schema fragment defines the `TrustmarkStatusCodeType` simple type.

```
<xs:simpleType name="TrustmarkStatusCodeType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="ACTIVE"/>
    <xs:enumeration value="REVOKED"/>
    <xs:enumeration value="EXPIRED"/>
  </xs:restriction>
</xs:simpleType>
```

For an illustration of `TrustmarkStatusCodeType`, see the illustration for the `<TrustmarkStatusReport>` element.

4.6 Trust Interoperability Profile

This section describes the XML encoding of a Trust Interoperability Profile. As noted in Section 3, a Trust Interoperability Profile expresses a trust and interoperability policy in terms of a set of Trustmarks that a Trustmark Recipient must possess in order to meet the policy's trust and interoperability requirements. The XML encoding of a Trust Interoperability Profile employs the following schema components to encode these requirements.

- `<TrustInteroperabilityProfile>` element
- `ReferencesType` complex type
- `TrustmarkDefinitionRequirementType` complex type

The following sections describe the aforementioned schema components.

4.6.1 Element `<TrustInteroperabilityProfile>`

The `<TrustInteroperabilityProfile>` element is an XML encoding of a Trust Interoperability Profile. It includes the following elements.

`<ds:Signature>` [Optional]

> `<ds:Signature>` is an optional XML signature that ensures the integrity and the authenticity of this Trust Interoperability Profile. If included, the signature MUST conform to XML signature normative requirements as specified in [XML Sig]. In addition, if included, the signature must reference and pertain to the outermost XML element (`<TrustInteroperabilityProfile>`) in the Trust Interoperability Profile XML object. As such, if a `<ds:Signature>` element appears in the `<TrustInteroperabilityProfile>` element, then the `<TrustInteroperabilityProfile>` element MUST carry the `id` attribute.

`<Identifier>` [Required]

> `<Identifier>` is a URL; it is the globally unique Trust Interoperability Profile Identifier for this Trust Interoperability Profile. Section 5.6 contains normative language pertaining to the selection of Trust Interoperability Profile Identifiers.

`<Name>` [Optional]

> `<Name>` a string; it is the name of this Trust Interoperability Profile, if any.

`<Version>` [Optional]

> `<Version>` is a string; it is the version of this Trust Interoperability Profile, if any.

`<Description>` [Optional]

> `<Description>` is a string; it is the description of this Trust Interoperability Profile, if any.

`<PublicationDateTime>` [Required]

> `<PublicationDateTime>` is the date and time at which the Trust Interoperability Profile Issuer published this Trust Interoperability Profile.

`<LegalNotice>` [Optional]

> `<LegalNotice>` is a string; it is the legal notice for this Trust Interoperability Profile, if any.

`<Notes>` [Optional]

> `<Notes>` is a string; it is additional optional text content about this Trust Interoperability Profile.

`<Issuer>` [Optional]

> `<Issuer>` is an `EntityType`; if provided, it is the Trust Interoperability Profile Issuer Identifier for the Trust Interoperability Profile Issuer that issued this Trust Interoperability Profile. Section 5.6 contains normative language pertaining to the selection of Trust Interoperability Profile Issuer Identifiers.

`<Supersessions>` [Optional]

> `<Supersessions>` is a container element for references to other Trust Interoperability Profiles that either supersede this Trust Interoperability Profile or have been superseded by this Trust Interoperability Profile. If present, it contains the following elements.
>
> `<Supersedes>` [Zero or More]
>
> > `<Supersedes>` is a `TrustInteroperabilityProfileReferenceType`; it is a reference to another Trust Interoperability Profile that this Trust Interoperability Profile supersedes or replaces. Multiple `<Supersedes>` elements can be included to indicate that this Trust Interoperability Profile supersedes or replaces multiple other Trust Interoperability Profiles.
>
> `<SupersededBy>` [Zero or More]
>
> > `<SupersededBy>` is a `TrustInteroperabilityProfileReferenceType`; it is a reference to another Trust Interoperability Profile that supersedes or replaces this Trust Interoperability Profile. Multiple `<SupersededBy>` elements can be included to indicate that multiple other Trust Interoperability Profiles supersede or replace this Trust Interoperability Profile.

`<Deprecated>` [Optional]

> `<Deprecated>` is an optional Boolean element that indicates whether this Trust Interoperability Profile has been deprecated. Absence of this element is equivalent to the presence of this element with a value of `false` or `0`.

`<Satisfies>` [Zero or More]

> `<Satisfies>` is a `TrustmarkDefinitionReferenceType`; it is a reference to a Trustmark Definition, and is intended to indicate that if an entity satisfies the trust expression for the enclosing Trust Interoperability Profile, then the entity may also satisfy the conformance criteria for the referenced Trustmark Definition. This element can be used to indicate that there exists an object-oriented "subclass" or "is-a" relationship between the enclosing Trust Interoperability Profile and the referenced Trustmark Definition. This can commonly occur when the referenced Trustmark Definition contains vague, high-level conformance criteria (e.g., "must use encryption to protect data") and the enclosing Trust Interoperability Profile's trust expression contains more specific criteria based on the conformance criteria that appear within the Trustmark Definitions that it includes (e.g., "must use AES 256-bit encryption to protect data").
>
> In practice, this reference SHOULD be interpreted by a Trustmark Provider as a hint meaning that, if an entity satisfies the trust expression for the enclosing Trust Interoperability Profile, then the Trustmark Provider should also strongly consider assessing the entity for the referenced Trustmark Definition and grant that trustmark if the assessment indicates conformance.

`<Keywords>` [Optional]

> `<Keywords>` is a container element for keywords that pertain to this Trust Interoperability Profile. Keywords are intended to facilitate search and discovery of Trust Interoperability Profiles within registries. If present, it contains the following elements.
>
> `<Keyword>` [One or More]
>
> > `<Keyword>` is a string; it contains a keyword that pertains to this Trust Interoperability Profile.

`<References>` [Required]

`<References>` is a `ReferencesType`; it is a set of references to the Trustmark Definition Requirements and Trust Interoperability Profiles used by the trust expression for this Trust Interoperability Profile.

`<TrustExpression>` [Required]

`<TrustExpression>` is a string; it is a Boolean expression that indicates whether an entity satisfies this Trust Interoperability Profile based on its possession of specific Trustmarks issued under the referenced Trustmark Definitions and on its satisfaction of the referenced Trust Interoperability Profiles.

If the trust expression evaluates to `true`, the entity satisfies the Trust Interoperability Profile; if the trust expression evaluates to `false`, the entity does not satisfy the Trust Interoperability Profile.

Appendix C describes the syntax and semantics of this Boolean expression.

`id` [Optional]

`id` is an XML ID for this Trustmark.

The following schema fragment defines the `<TrustInteroperabilityProfile>` element.

```
<xs:element name="TrustInteroperabilityProfile">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ds:Signature"
                  minOccurs="0"
                  maxOccurs="1"/>
      <xs:element name="Identifier"
                  type="xs:anyURI"
                  minOccurs="1"
                  maxOccurs="1"/>
      <xs:element name="Name"
                  type="xs:string"
                  minOccurs="0"
                  maxOccurs="1"/>
      <xs:element name="Version"
                  type="xs:string"
                  minOccurs="0"
                  maxOccurs="1"/>
      <xs:element name="Description"
                  type="xs:string"
                  minOccurs="0"
                  maxOccurs="1"/>
      <xs:element name="PublicationDateTime"
                  type="xs:dateTime"
                  minOccurs="1"
                  maxOccurs="1"/>
      <xs:element name="LegalNotice"
                  type="xs:string"
                  minOccurs="0"
                  maxOccurs="1"/>
      <xs:element name="Notes"
                  type="xs:string"
                  minOccurs="0"
                  maxOccurs="1"/>
      <xs:element name="Issuer"
                  type="tf:EntityType"
                  minOccurs="0"
                  maxOccurs="1"/>
      <xs:element name="Supersessions"
                  minOccurs="0"
                  maxOccurs="1">
```

```
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Supersedes"
                        type=
                          "tf:TrustInteroperabilityProfileReferenceType"
                        minOccurs="0"
                        maxOccurs="unbounded"/>
            <xs:element name="SupersededBy"
                        type=
                          "tf:TrustInteroperabilityProfileReferenceType"
                        minOccurs="0"
                        maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Deprecated"
                  type="xs:boolean"
                  minOccurs="0"
                  maxOccurs="1"/>
      <xs:element name="Satisfies"
                  type="tf:TrustmarkDefinitionReferenceType"
                  minOccurs="0"
                  maxOccurs="unbounded">
      </xs:element>
      <xs:element name="Keywords"
                  minOccurs="0"
                  maxOccurs="1">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Keyword"
                        type="xs:string"
                        minOccurs="1"
                        maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="References"
                  type="tf:ReferencesType"
                  minOccurs="1"
                  maxOccurs="1"/>
      <xs:element name="TrustExpression"
                  type="xs:string"
                  minOccurs="1"
                  maxOccurs="1"/>
    </xs:sequence>
    <xs:attribute ref="tf:id"
                  use="optional"/>
  </xs:complexType>
</xs:element>
```

The following example XML fragments illustrate the `<TrustInteroperabilityProfile>` element. The first example illustrates the minimum content required by its definition.

```
<tf:TrustInteroperabilityProfile
  xmlns:tf="https://trustmark.gtri.gatech.edu/specifications/trustmark-
framework/1.1/schema/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <tf:Identifier>
    https://trustmark.gtri.gatech.edu/operational-pilot/trust-
interoperability-profiles/ficam-privacy-tip-for-csp/1.0/
  </tf:Identifier>
  <tf:PublicationDateTime>2014-10-01T14:41:38</tf:PublicationDateTime>
  <tf:References>
    <tf:TrustmarkDefinitionRequirement tf:id="TD_1">
      <tf:TrustmarkDefinitionReference>
        <tf:Identifier>
https://trustmark.gtri.gatech.edu/operational-pilot/trustmark-
definitions/ficam-privacy-activity-tracking-requirements-for-csps-and-
bae-responders/1.0/
        </tf:Identifier>
      </tf:TrustmarkDefinitionReference>
    </tf:TrustmarkDefinitionRequirement>
  </tf:References>
  <tf:TrustExpression>
    <![CDATA[TD_1]]>
  </tf:TrustExpression>
</tf:TrustInteroperabilityProfile>
```

The second example illustrates additional content permitted by its definition. Additional content in this example appears in **boldface** text.

```
<tf:TrustInteroperabilityProfile
  xmlns:tf="https://trustmark.gtri.gatech.edu/specifications/trustmark-
framework/1.1/schema/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:Signature>
    <!-- The example omits the XML signature content. -->
  </ds:Signature>
  <tf:Identifier>
    https://trustmark.gtri.gatech.edu/operational-pilot/trust-
interoperability-profiles/ficam-privacy-tip-for-csp/1.0/
  </tf:Identifier>
  <tf:Name>FICAM Privacy TIP for CSP</tf:Name>
  <tf:Version>1.0</tf:Version>
  <tf:Description>
    <![CDATA[This Trust Interoperability Profile specifies the FICAM
mandated privacy requirements to be followed by all FICAM Credential
Service Providers (CSPs).]]>
  </tf:Description>
  <tf:PublicationDateTime>2014-10-01T14:41:38</tf:PublicationDateTime>
  <tf:LegalNotice>
    This document and the information contained herein is provided on an
    "AS IS" basis, and the Georgia Tech Research Institute disclaims all
    warranties, express or implied, including but not limited to any
    warranty that the use of the information herein will not infringe any
    rights or any implied warranties or merchantability or fitness for a
    particular purpose. In addition, the Georgia Tech Research Institute
    disclaims legal liability for any loss incurred as a result of the
    use or reliance on the document or the information contained herein.
  </tf:LegalNotice>
  <tf:Notes>
    The Georgia Tech Research Institute (GTRI) has published this
    document with the support of the National Strategy for Trusted
    Identities in Cyberspace (NSTIC) via the National Institute of
    Standards and Technology (NIST). The views expressed herein do not
    necessarily reflect the official policies of GTRI, NIST or NSTIC; nor
    does mention of trade names, commercial practices, or organizations
    imply endorsement by the U.S. Government.
  </tf:Notes>
  <tf:Issuer>
    <tf:Identifier>https://trustmark.gtri.gatech.edu/</tf:Identifier>
    <tf:Name>GTRI NSTIC Trustmark Pilot</tf:Name>
    <tf:Contact>
      <tf:Kind>PRIMARY</tf:Kind>
      <tf:Responder>George P. Burdell</tf:Responder>
      <tf:Email>TrustmarkFeedback@gtri.gatech.edu</tf:Email>
      <tf:Telephone>404-407-8956</tf:Telephone>
      <tf:PhysicalAddress>
        75 5th Street NW,
        Atlanta, GA 30308
      </tf:PhysicalAddress>
      <tf:MailingAddress>
        75 5th Street NW,
        Suite 900,
        Atlanta, GA 30308
      </tf:MailingAddress>
      <tf:WebsiteURL>http://trustmark.gtri.gatech.edu/</tf:WebsiteURL>
      <tf:Notes>The responder may change.</tf:Notes>
    </tf:Contact>
  </tf:Issuer>
  <tf:Supersessions>
    <tf:Supersedes>
      <tf:Identifier>
https://trustmark.gtri.gatech.edu/operational-pilot/trust-
interoperability-profiles/ficam-privacy-tip-for-csp/0.9/
      </tf:Identifier>
      <tf:Name>FICAM Privacy TIP for CSP</tf:Name>
      <tf:Version>0.9</tf:Version>
    </tf:Supersedes>
    <tf:SupersededBy>
      <tf:Identifier>
https://trustmark.gtri.gatech.edu/operational-pilot/trust-
interoperability-profiles/ficam-privacy-tip-for-csp/1.1/
      </tf:Identifier>
      <tf:Name>FICAM Privacy TIP for CSP</tf:Name>
      <tf:Version>1.1</tf:Version>
    </tf:SupersededBy>
  </tf:Supersessions>
  <tf:Deprecated>false</tf:Deprecated>
  <tf:Keywords>
    <tf:Keyword>FICAM</tf:Keyword>
    <tf:Keyword>Privacy</tf:Keyword>
    <tf:Keyword>Credential Service Provider</tf:Keyword>
    <tf:Keyword>CSP</tf:Keyword>
</tf:Keywords>
  <tf:References>
    <tf:TrustmarkDefinitionRequirement tf:id="TD_1">
      <tf:TrustmarkDefinitionReference>
        <tf:Identifier>
https://trustmark.gtri.gatech.edu/operational-pilot/trustmark-
definitions/ficam-privacy-activity-tracking-requirements-for-csps-and-
bae-responders/1.0/
        </tf:Identifier>
        <tf:Name>
          FICAM Privacy Activity Tracking Requirements
          for CSPs and BAE Responders
        </tf:Name>
        <tf:Version>1.0</tf:Version>
        <tf:Description>
          This TD adopts the LOA 2 and LOA 3 Privacy Activity Tracking
          trust criteria of the FICAM Trust Framework Provider Adoption
          Process.
        </tf:Description>
      </tf:TrustmarkDefinitionReference>
      <tf:ProviderReference>
        <tf:Identifier>https://trustmark.gtri.gatech.edu/</tf:Identifier>
      </tf:ProviderReference>
  </tf:TrustmarkDefinitionRequirement>
  <tf:TrustExpression>
    <![CDATA[TD_1]]>
  </tf:TrustExpression>
</tf:TrustInteroperabilityProfile>
```

4.6.2 Complex Type `ReferencesType`

`ReferencesType` is a complex type for the references to the Trustmark Definition requirements and Trust Interoperability Profiles referenced by the trust expression of a Trust Interoperability Profile. It includes at least one of the following elements.

`<TrustmarkDefinitionRequirement>` [Required]

> `<TrustmarkDefinitionRequirement>` is a `TrustmarkDefinitionRequirementType`; it is a reference to a Trustmark Definition requirement.

`<TrustInteroperabilityProfileReference>` [Required]

> `<TrustInteroperabilityProfileReference>` is a `TrustInteroperabilityProfileReferenceType`; it is a reference to a trust interoperability profile.

The following schema fragment defines the `ReferencesType` complex type.

```
<xs:complexType name="ReferencesType">
  <xs:choice minOccurs="1"
             maxOccurs="unbounded">
    <xs:element name="TrustmarkDefinitionRequirement"
                type="tf:TrustmarkDefinitionRequirementType"/>
    <xs:element name="TrustInteroperabilityProfileReference"
                type="tf:TrustInteroperabilityProfileReferenceType"/>
  </xs:choice>
</xs:complexType>
```

For an example of the `ReferencesType` complex type, see the example for the `<TrustInteroperabilityProfile>` element in Section 4.6.1.

4.6.3 Complex Type `TrustmarkDefinitionRequirementType`

`TrustmarkDefinitionRequirementType` is a complex type for a Trustmark Definition requirement. It includes the following elements.

`<TrustmarkDefinitionReference>` [Required]

> `<TrustmarkDefinitionReference>` is a `TrustmarkDefinitionReferenceType`; it is a reference to the Trustmark Definition that is the subject of this Trustmark Definition requirement.

`<ProviderReference>` [Any Number]

> `<ProviderReference>` is an `EntityReferenceType;` it is an optional reference to a Trustmark Provider that issues Trustmarks under the Trustmark Definition that is the subject of this Trustmark Definition Requirement.

`id` [Required]

> `id` is an XML ID for this Trustmark Definition requirement. This id permits the content of the `<TrustExpression>` element to refer to this Trustmark Definition requirement. This id MUST appear in the `<TrustExpression>` element of the enclosing Trust Interoperability Profile.

The following schema fragment defines the `TrustmarkDefinitionRequirementType` complex type.

```
<xs:complexType name="TrustmarkDefinitionRequirementType">
  <xs:sequence>
    <xs:element name="TrustmarkDefinitionReference"
                type="tf:TrustmarkDefinitionReferenceType"
                minOccurs="1"
                maxOccurs="1"/>
    <xs:element name="ProviderReference"
                type="tf:EntityReferenceType"
                minOccurs="0"
                maxOccurs="unbounded"
                nillable="true"/>
  </xs:sequence>
  <xs:attribute ref="tf:id"
                use="required"/>
</xs:complexType>
```

The following example XML fragments illustrate the `<TrustmarkDefinitionRequirement>` element, which is of the `TrustmarkDefinitionRequirementType` complex type. The first example illustrates the minimum content required by its definition.

```
<tf:TrustmarkDefinitionRequirement tf:id="TD_1">
  <tf:TrustmarkDefinitionReference>
    <tf:Identifier>
https://trustmark.gtri.gatech.edu/operational-pilot/trustmark-
definitions/ficam-privacy-activity-tracking-requirements-for-csps-and-
bae-responders/1.0/
    </tf:Identifier>
  </tf:TrustmarkDefinitionReference>
</tf:TrustmarkDefinitionRequirement>
```

The second example illustrates additional content permitted by its definition. Additional content in this example appears in **boldface** text.

```
<tf:TrustmarkDefinitionRequirement tf:id="TD_1">
  <tf:TrustmarkDefinitionReference>
    <tf:Identifier>
https://trustmark.gtri.gatech.edu/operational-pilot/trustmark-
definitions/ficam-privacy-activity-tracking-requirements-for-csps-and-
bae-responders/1.0/
    </tf:Identifier>
    <tf:Name>
      FICAM Privacy Activity Tracking Requirements
      for CSPs and BAE Responders
    </tf:Name>
    <tf:Version>1.0</tf:Version>
    <tf:Description>
      This TD adopts the LOA 2 and LOA 3 Privacy Activity Tracking trust
      criteria of the FICAM Trust Framework Provider Adoption Process.
    </tf:Description>
  </tf:TrustmarkDefinitionReference>
  <tf:ProviderReference>
    <tf:Identifier>https://trustmark.gtri.gatech.edu/</tf:Identifier>
    <tf:Name>Georgia Tech Research Institute</tf:Name>
    <tf:Contact>
      <tf:Kind>PRIMARY</tf:Kind>
      <tf:Responder>George P. Burdell</tf:Responder>
      <tf:Email>TrustmarkFeedback@gtri.gatech.edu</tf:Email>
      <tf:Telephone>404-407-8956</tf:Telephone>
      <tf:PhysicalAddress>
        75 5th Street NW,
        Atlanta, GA 30308
      </tf:PhysicalAddress>
      <tf:MailingAddress>
        75 5th Street NW,
        Suite 900,
        Atlanta, GA 30308
      </tf:MailingAddress>
      <tf:WebsiteURL>http://trustmark.gtri.gatech.edu/</tf:WebsiteURL>
      <tf:Notes>The responder may change.</tf:Notes>
    </tf:Contact>
  </tf:ProviderReference>
</tf:TrustmarkDefinitionRequirement>
```

5 Trustmark Framework Operational Considerations

The purpose of this section is to define a set of normative requirements that govern the operational issuance of Trustmarks, including prerequisites for Trustmark Definition identifier values, Trustmark issuance, requirements for Trustmark issuance and publication, requirements for Trustmark validation, and requirements for Trust Interoperability Profile identifier values.

5.1 Identifier Requirements for Trustmark Definitions

The following requirements apply to any Trustmark Defining Organization that publishes one or more Trustmark Definitions.

1. Before publishing a Trustmark Definition, a Trustmark Defining Organization MUST establish a Trustmark Defining Organization Identifier that uniquely identifies it. The Trustmark Defining Organization Identifier MUST be a URL on a Domain Name Service (DNS) domain that is under the control of the Trustmark Defining Organization.
2. A Trustmark Defining Organization SHOULD use the same Trustmark Defining Organization Identifier for each Trustmark Definition that it publishes.
3. When publishing a Trustmark Definition, a Trustmark Defining Organization MUST include its Trustmark Defining Organization Identifier in the `<TrustmarkDefiningOrganization>` element of the Trustmark Definition.
4. When publishing a Trustmark Definition, a Trustmark Defining Organization MUST create a unique Trustmark Definition Identifier for the Trustmark Definition, and MUST include the Trustmark Definition Identifier in the `<Identifier>` element of the Trustmark Definition. The Trustmark Definition Identifier SHOULD be a sub-path of the Trustmark Defining Organization Identifier that is used in the Trustmark Definition.

5.2 Prerequisites for Trustmark Issuance

There are several prerequisites that the Trustmark Provider and/or the Trustmark Recipient MUST satisfy before the Trustmark Provider can issue a Trustmark to the Trustmark Recipient. The following subsections enumerate these prerequisites.

5.2.1 Establishment of a Unique Trustmark Provider Identifier

Before a Trustmark Provider can issue Trustmarks to Trustmark Recipients, the Trustmark Provider MUST establish one or more Trustmark Provider Identifiers that uniquely identify it to Trustmark Relying Parties. The Trustmark Provider Identifier MUST be a URL on a Domain Name Service (DNS) domain that is under the control of the Trustmark Provider.

5.2.2 Establishment of a Trustmark Signing Certificate

Before a Trustmark Provider can issue Trustmarks to Trustmark Recipients, the Trustmark Provider MUST establish one or more Trustmark Signing Certificates. A Trustmark Signing Certificate is an X.509 ([X509]) Public Key Infrastructure (PKI) certificate that a Trustmark Provider can use to digitally sign Trustmarks that it issues. Trustmark Relying Parties can use a Trustmark Signing Certificate to uniquely identify the Trustmark

Provider as the publisher of a Trustmark and cryptographically verify a Trustmark's authenticity and integrity. The following rules and guidelines apply to each Trustmark Signing Certificate.

1. The public/private key pair used for the Trustmark Signing Certificate MUST be generated by the Trustmark Signing Certificate itself.
2. The Trustmark Provider MUST maintain positive control over the private key for the Trustmark Signing Certificate at all times prior to and during operational use of the Trustmark Signing Certificate.
3. The Trustmark Provider MUST publish a Trustmark Signing Certificate Policy describing the security policies that it follows to ensure that it maintains positive control over the private key for the Trustmark Signing Certificate. The Trustmark Provider SHOULD publish its Trustmark Signing Certificate Policy at a well-defined, publicly available URL, and the Trustmark Provider MUST make its Trustmark Signing Certificate Policy available for inspection by any Trustmark Relying Party upon request.
4. The Trustmark Signing Certificate MAY be signed by other entities, e.g., organizations that wish to endorse the Trustmark Provider for the benefit of Trustmark Relying Parties.
5. The Trustmark Signing Certificate's Common Name MUST match the DNS domain name used by the Trustmark Provider for one of its Trustmark Provider Identifiers. It is RECOMMENDED that a Trustmark Provider establish one Trustmark Signing Certificate for each unique DNS domain name used in its Trustmark Provider Identifiers, so that each Trustmark it issues can be signed by a Trustmark Signing Certificate with a Common Name that is consistent with the Trustmark Provider Identifier in the Trustmark.
6. The Trustmark Provider SHOULD publish the Trustmark Signing Certificate at a URL that is a sub-path of one of the Trustmark Provider's Trustmark Provider Identifiers. It is RECOMMENDED that a Trustmark Provider publish each Trustmark Signing Certificate at a URL that is a sub-path of the DNS domain name that appears in the Trustmark Signing Certificate's Common Name.

5.2.3 Establishment and Publication of Trustmark Policy and Agreements

Before a Trustmark Provider can issue Trustmarks to Trustmark Recipients, the Trustmark Provider MUST establish and publish one or more Trustmark Policies. A Trustmark Policy MUST define the rules that govern the issuance, usage, and lifecycle management of a Trustmark from the Trustmark Provider that published the policy. Also, the Trustmark Policy MUST define roles, responsibilities, and limitations on liability for damages incurred as a result of usage or reliance on a Trustmark from the Trustmark Provider that published the policy. The Trustmark Provider MUST publish each Trustmark Policy at a well-defined, publicly available URL prior to invoking it via an issued Trustmark. The published location of the Trustmark Policy SHOULD be a sub-path of one of the Trustmark Provider's Trustmark Provider Identifiers. See Section 5.3 for more information about how a Trustmark Provider can invoke a Trustmark Policy via a Trustmark that it issues.

In addition, before a Trustmark Provider can issue Trustmarks to Trustmark Recipients, the Trustmark Provider MUST execute a Trustmark Recipient Agreement with that Trustmark Recipient. A Trustmark Recipient Agreement serves to legally bind the Trustmark Provider and Trustmark Recipient to the terms and conditions expressed in a Trustmark Policy within the context of all Trustmarks issued to the Trustmark Recipient by the Trustmark Provider. A Trustmark Recipient Agreement SHOULD be a signed, two-party contract between the Trustmark Provider and Trustmark Recipient.

Finally, before a Trustmark Provider can issue Trustmarks to Trustmark Recipients, the Trustmark Provider MUST establish one or more Trustmark Relying Party Agreements. A Trustmark Relying Party Agreement serves to legally bind the Trustmark Provider and a Trustmark Relying Party to the terms and conditions expressed in a Trustmark Policy within the context of a specific Trustmark. A Trustmark Relying Party Agreement is binding upon use of a Trustmark by the Trustmark Relying Party. The Trustmark Provider MUST publish each Trustmark Relying Party Agreement at a well-defined, publicly available URL prior to invoking it via an issued Trustmark. The published location of the Trustmark Relying Party Agreement SHOULD be a sub-path of one of the Trustmark Provider's Trustmark Provider Identifiers. See Section 5.3 for more information about how a Trustmark Provider can invoke a Trustmark Relying Party Agreement via a Trustmark that it issues.

5.2.4 Establishment of a Unique Trustmark Recipient Identifier

Before a Trustmark Provider can issue Trustmarks to a prospective Trustmark Recipient, the Trustmark Provider and Trustmark Recipient MUST establish a Trustmark Recipient Identifier that uniquely identifies the Trustmark Recipient. The process for establishing a Trustmark Recipient Identifier is as follows.

1. The prospective Trustmark Recipient chooses its proposed Trustmark Recipient Identifier and notifies the Trustmark Provider of its choice. The following rules and guidelines apply to the Trustmark Recipient Identifier.
    a. The proposed Trustmark Recipient Identifier MUST be a URL on a DNS domain that is under the control of the Trustmark Recipient.
    b. The proposed Trustmark Recipient Identifier SHOULD be chosen so as to uniquely identify the prospective Trustmark Recipient as an organization, even if the organization is a department, subunit, or subsidiary of a larger organization.

       For example, Georgia Tech might choose "https://gatech.edu/" as its proposed Trustmark Recipient Identifier. But the Georgia Tech Office of Information Technology, which is a department of Georgia Tech, might choose "https://oit.gatech.edu/" to distinguish itself from its larger parent organization.
    c. A Trustmark Recipient that plans to obtain Trustmarks from multiple Trustmark Providers SHOULD use the same Trustmark Recipient Identifier for each Trustmark Provider.
2. The Trustmark Provider MUST verify that the prospective Trustmark Recipient controls the URL proposed as the Trustmark Recipient Identifier. The following rules and guidelines apply to the Trustmark Recipient Identifier verification process.
    a. The Trustmark Provider MUST verify Trustmark Recipient control of the URL via a simple challenge-response process, in which: (i) the Trustmark Provider provides a long random number or other hard-to-guess data to the TR, (ii) the Trustmark Recipient publishes the data temporarily at the proposed URL or a sub-path of it, and (iii) the Trustmark Provider performs an HTTP request at the URL to verify that the Trustmark Recipient was able to successfully publish the data as required.
    b. The Trustmark Provider MAY perform additional steps to verify Trustmark Recipient control of the URL, e.g., verification of DNS domain name registration for the URL via WHOIS lookup if appropriate.

5.3 Trustmark Issuance Requirements for Trustmark Providers

After a Trustmark Provider has established a Trustmark Provider Identifier in accordance with the rules defined in Section 5.2.1, and the Trustmark Provider has established a Trustmark Signing Certificate in accordance with the rules defined in Section 5.2.2, and the Trustmark Provider has established appropriate policies and executed appropriate agreements as stipulated in Section 5.2.3, and the Trustmark Provider and a prospective Trustmark Recipient have established a Trustmark Recipient Identifier as per the process defined in Section 5.2.4, the Trustmark Provider MAY begin to issue Trustmarks to the Trustmark Recipient. The following rules and guidelines apply to the issuance of Trustmarks.

1. Each Trustmark issued by the Trustmark Provider MUST uniquely identify the Trustmark Provider by including one of its Trustmark Provider Identifiers within the Trustmark structure.
2. Each Trustmark issued by the Trustmark Provider to the Trustmark Recipient MUST uniquely identify the Trustmark Recipient by including the Trustmark Recipient Identifier within the Trustmark structure.
3. Each Trustmark issued by the Trustmark Provider MUST contain the URL of the Trustmark Policy that pertains to the Trustmark.
4. Each Trustmark issued by the Trustmark Provider MUST contain the URL of the Trustmark Relying Party Agreement that pertains to the Trustmark.

5. Each Trustmark issued by the Trustmark Provider MUST contain a unique Trustmark Identifier. This identifier MUST be the URL at which the Trustmark resides, and SHOULD be a sub-path of one of the Trustmark Provider's Trustmark Provider Identifiers. In addition, prior to issuing the Trustmark to the Trustmark Recipient, the Trustmark Provider MUST publish the Trustmark at this URL.
6. Each Trustmark issued by the Trustmark Provider MUST contain a unique, Trustmark specific Trustmark Status URL (see Section 5.4 for requirements for Trustmark Status Reports that live at Trustmark Status URLs).
7. Each Trustmark issued by the Trustmark Provider MUST contain a digital signature created by the Trustmark Provider using one of its Trustmark Signing Certificates.

5.4 Trustmark Publication and Revocation Requirements for Trustmark Providers

The following requirements apply for any Trustmark published by a Trustmark Provider.

1. For each Trustmark that it publishes, a Trustmark Provider SHOULD maintain an online copy of the Trustmark at the URL indicated by the Trustmark's unique identifier, for as long as the Trustmark remains non-revoked and non-expired.
2. The Trustmark Provider MAY remove the Trustmark from its published location at any time after the Trustmark expires or is revoked.
3. After removing the Trustmark from its published location, the Trustmark Provider SHOULD return a HTTP 404 or 410 code in response to any request for the Trustmark at its published location, and SHOULD NOT return any type of HTTP 3xx redirect code.
4. The Trustmark Provider MUST publish a Trustmark Status Report for each Trustmark that it issues, and MUST update the Trustmark Status Report as appropriate when the Trustmark's status changes. The Trustmark Status Report for a Trustmark MUST reside online at the Trustmark's Status URL.
5. Every Trustmark Status Report published by a Trustmark Provider MUST reside at a TLS-protected HTTPS endpoint. In addition, the Trustmark Provider SHOULD provide a convenient mechanism for HTTP clients to verify the validity and trustworthiness of the TLS certificate for this endpoint. It is RECOMMENDED that the Trustmark Provider protect this endpoint with a TLS certificate that was issued by a Certificate Authority (CA) that is well known and widely trusted, e.g., a CA for which the certificate is pre-installed by default in modern versions of popular Web browsers such as Mozilla Firefox, Google Chrome, and Microsoft Internet Explorer.
6. The Trustmark Provider MUST continue to publish a Trustmark Status Report for a Trustmark until the Trustmark's expiration date, even if the Trustmark was revoked prior to its expiration.
7. The Trustmark Provider MAY continue to provide a Trustmark Status Report following the Trustmark's expiration; in this case, the Trustmark Status Report SHOULD include the unique identifier(s) of any superseding Trustmark(s) if applicable.
8. After removing the Trustmark Status Report from its published location, the Trustmark Provider SHOULD return a HTTP 404 or 410 code in response to any request for the Trustmark Status Report, and SHOULD NOT return any type of HTTP 3xx redirect code.
9. If one of the Trustmark Provider's Trustmark Signing Certificates expires or is revoked, the Trustmark Provider MUST update the Trustmark Status Report for every active Trustmark signed by that Trustmark Signing Certificate, to indicate that the Trustmark's status is now "REVOKED".
10. After a Trustmark's status changes from "ACTIVE" to "REVOKED" or "EXPIRED", the Trustmark Provider MUST NOT change the Trustmark's status back to "ACTIVE".

5.5 Trustmark Validation Requirements for Trustmark Relying Parties

Before relying on a Trustmark for any trust or interoperability policy decision, a Trustmark Relying Party MUST perform the following validation steps. If any validation step fails, the Trustmark Relying Party MUST reject the Trustmark and MUST NOT rely upon it.

1. **Verification of the Trustmark Provider Identifier**: Verify that the Trustmark Provider Identifier on the Trustmark is a valid URL and identifies one of the Trustmark Providers trusted by the Trustmark Relying Party.
2. **Verification of the Trustmark's Digital Signature**: Verify that the digital signature on the Trustmark is cryptographically consistent with the Trustmark's contents.
3. **Verification of the Trustmark Signing Certificate's Common Name**: Verify that the Trustmark Signing Certificate used to sign the Trustmark contains a Common Name that is consistent with the Trustmark Provider Identifier in the Trustmark.
4. **Verification of the Trustmark Signing Certificate's Status**: Verify that the Trustmark Signing Certificate used to sign the Trustmark has not expired or been revoked. Note, as per Section 5.4, item 9, that a Trustmark Provider is required to revoke the Trustmark if its Trustmark Signing Certificate expires or is revoked.
5. **Verification of the Trustmark's Identifier**: Verify that the Trustmark's identifier is a valid URL that indicates a sub-path of the URL specified for the Trustmark Provider Identifier for the Trustmark.
6. **Verification of Trustmark Non-Expiration**: Verify, via the expiration date-time on the Trustmark, that the Trustmark is not yet expired.
7. **Verification of Trustmark Non-Revocation**: Verify, through the Trustmark Status Report at the Trustmark's Status URL, that the Trustmark has not been revoked. Note that a Trustmark Relying Party MAY perform additional revocation checks periodically or as needed throughout the duration of its period of reliance upon the Trustmark.
8. **Verification of Proper Organizational Scope via the Trustmark Recipient Identifier**: Verify that the Trustmark Recipient Identifier matches and/or logically corresponds to a known URL for the entity about which the Trustmark was (or is assumed to have been) issued, and for which the Trustmark conveys trust.
9. **Verification of Proper Operational Scope via the Trustmark Definition**: Verify that the purpose for which the Trustmark will be used, or the purpose for which it will be relied upon, is consistent with the Trustmark's meaning and intended usage as per its Trustmark Definition.

In addition to completing the steps listed previously, the Trustmark Relying Party SHOULD determine whether the Trustmark contains any exceptions by checking for the presence of an `<ExceptionInfo>` element. If such an element is present, the Trustmark Relying Party SHOULD make a risk-based decision about whether to trust the Trustmark, in accordance with its risk profile.

5.6 Identifier Requirements for Trust Interoperability Profiles

The following requirements apply to any Trust Interoperability Profile Issuer that publishes one or more Trust Interoperability Profiles.

1. Before publishing a Trust Interoperability Profile, a Trust Interoperability Profile Issuer MUST establish a Trust Interoperability Profile Issuer Identifier that uniquely identifies it. The Trust Interoperability Profile Issuer Identifier MUST be a URL on a Domain Name Service (DNS) domain that is under the control of the Trust Interoperability Profile Issuer. If the Trust Interoperability Profile Issuer is also a Trustmark Recipient, the Trust Interoperability Profile Issuer Identifier SHOULD be the same as one of the Trust Interoperability Profile Issuer's Trustmark Recipient Identifiers. See Section 5.2.4 for more information about Trustmark Recipient Identifiers.
2. A Trust Interoperability Profile Issuer SHOULD use the same Trust Interoperability Profile Issuer Identifier for each Trust Interoperability Profile that it publishes.
3. When publishing a Trust Interoperability Profile, a Trust Interoperability Profile Issuer MUST include its Trust Interoperability Profile Issuer Identifier in the `<Issuer>` element of the Trust Interoperability Profile.

4. When publishing a Trust Interoperability Profile, a Trust Interoperability Profile Issuer MUST create a unique Trust Interoperability Profile Identifier for the Trust Interoperability Profile, and MUST include the Trust Interoperability Profile Issuer Identifier in the `<Identifier>` element of the Trust Interoperability Profile. The Trust Interoperability Profile Identifier SHOULD be a sub-path of the Trust Interoperability Profile Issuer Identifier that is used in the Trust Interoperability Profile.

6 References

The following sources are referenced by this document.

Document ID	Document Reference
[ISO 14977]	Information Technology - Syntactic Metalanguage - Extended BNF. ISO/IEC14977, December 1996.
[RFC 2119]	S. Bradner. Key words for use in RFCs to Indicate Requirement Levels. IETF RFC 2119, March 1997. http://www.ietf.org/rfc/rfc2119.txt
[RFC 2396]	T. Berners-Lee. Uniform Resource Identifiers (URI): Generic Syntax. IETF RFC 2396, August 1998. http://www.ietf.org/rfc/rfc2396.txt
[Schema1]	H. S. Thompson et al. XML Schema Part 1: Structures. World Wide Web Consortium Recommendation, May 2001. http://www.w3.org/TR/xmlschema-1/
[Schema2]	P. V. Biron et al. XML Schema Part 2: Datatypes. World Wide Web Consortium Recommendation, May 2001. http://www.w3.org/TR/xmlschema-2/
[XML]	T. Bray, et al. Extensible Markup Language (XML) 1.0 (Second Edition). World Wide Web Consortium, October 2000. http://www.w3.org/TR/REC-xml
[XML Sig]	D. Eastlake et al., XML-Signature Syntax and Processing, World Wide Web Consortium, February 2002. http://www.w3.org/TR/xmldsig-core/
[X509]	ITU-T Recommendation X.509 (1997 E): Information Technology - Open Systems Interconnection - The Directory: Authentication Framework, June 1997.

Appendix A: Sponsor Acknowledgment and Disclaimer

The Georgia Tech Research Institute (GTRI) developed this document through its NSTIC Trustmark Pilot under the support of the National Strategy for Trusted Identities in Cyberspace (NSTIC) via the National Institute of Standards and Technology (NIST). The views expressed herein do not necessarily reflect the official policies of NIST or NSTIC; nor does mention by trade names, commercial practices, or organizations imply endorsement by the U.S. Government.

Appendix B: Issuance Criteria Language for Trustmark Definitions

The content of the `<IssuanceCriteria>` element in a Trustmark Definition is a Boolean expression that indicates whether a Trustmark Provider may issue a Trustmark to a Trustmark Recipient, based on the results of the assessment process. This appendix describes the syntax and semantics of the language used to write these Boolean expressions.

The following grammar expresses the syntax of the issuance criteria in Extended Backus-Naur Form [ISO 14977].

```
issuance-criteria = predicate
                  | issuance-criteria , ' and ' , issuance-criteria
                  | issuance-criteria , ' or ' , issuance-criteria
                  | '!' , issuance-criteria
                  | 'not ' , issuance-criteria
                  | 'not(' , issuance-criteria , ')'
                  | '(' , issuance-criteria , ')'
predicate         = predicate-yes
                  | predicate-no
                  | predicate-na
predicate-yes     = 'yes(' , step-id , ')'
                  | 'yes(' , step-id-range , ')'
                  | 'yes(' , step-id-sequence , ')'
                  | 'yes(ALL)'
                  | 'yes(NONE)'
                  | step-id
predicate-no      = 'no(' , step-id , ')'
                  | 'no(' , step-id-range , ')'
                  | 'no(' , step-id-sequence , ')'
                  | 'no(ALL)'
                  | 'no(NONE)'
predicate-na      = 'na(' , step-id , ')'
                  | 'na(' , step-id-range , ')'
                  | 'na(' , step-id-sequence , ')'
                  | 'na(ALL)'
                  | 'na(NONE)'
step-id-sequence  = step-id , ',' , step-id-sequence
step-id-range     = step-id , '...' , step-id
step-id           = ? an XML ID for an assessment step ?
```

The preceding grammar defines the issuance criteria as a Boolean expression consisting of the Boolean operators and a set of predicates.

The Boolean operators are "and", "or", "not", "!", "(", and ")". The operator “and” is the binary logical conjunction operator. The operator “or” is the binary logical disjunction operator. The operators “not” and “!” are each the unary logical negation operator. The parentheses operators are used for grouping. The order of operations is as follows.

1. “(“, “)”
2. “not”, “!”
3. “and”
4. “or”

A predicate may be a function applied to an XML ID for an assessment step, to an ellipsis-separated range of XML IDs for assessment steps, to a comma-separated sequence of XML IDs for assessment steps, to the string `ALL`, or to the string `NONE`.

For example, given that `step1` and `step2` are the XML IDs of assessment steps, the following are predicates.

```
step1
```

```
yes(step1)
no(step1,step2)
na(ALL)
```

The Boolean value of a predicate is as follows.

1. If the predicate matches the first line of the `predicate-yes` production, the value of the predicate is `true` only if the answer to the assessment step referenced by `step-id` is "yes"; the value of the predicate is `false` only if the answer to the assessment step referenced by `step-id` is "no" or "not applicable".
2. If the predicate matches the second line of the predicate-yes production, the value of the predicate is true only if the answer to every assessment step from the assessment step referenced by the first step-id in the range to the assessment step referenced by the second step-id in the range, inclusive, is “yes”; the value of the predicate is false the answer to any assessment step in the range is “no” or “not applicable”.
3. If the predicate matches the third line of the `predicate-yes` production, the value of the predicate is `true` only if the answer to every assessment step referenced in `step-id-sequence` is "yes"; the value of the predicate is `false` if the answer to any assessment step referenced by `step-id-sequence` is "no" or "not applicable".
4. If the predicate matches the fourth line of the `predicate-yes` production, the value of the predicate is `true` only if the answer to every assessment step in the Trustmark Definition is "yes"; the value of the predicate is `false` if the answer to any assessment step in the Trustmark Definition is "no" or "not applicable".
5. If the predicate matches the fifth line of the `predicate-yes` production, the value of the predicate is `true` only if the answer to every assessment step in the Trustmark Definition is either "no" or "not applicable"; the value of the predicate is `false` if the answer to any assessment step in the Trustmark Definition is "yes".
6. If the predicate matches the first line of the `predicate-no` production, the value of the predicate is `true` only if the answer to the assessment step referenced by `step-id` is "no"; the value of the predicate is `false` only if the answer to the assessment step referenced by `step-id` is "yes" or "not applicable".
7. If the predicate matches the second line of the predicate-no production, the value of the predicate is true only if the answer to every assessment step from the assessment step referenced by the first step-id in the range to the assessment step referenced by the second step-id in the range, inclusive, is “no”; the value of the predicate is false the answer to any assessment step in the range is “yes” or “not applicable”.
8. If the predicate matches the third line of the `predicate-no` production, the value of the predicate is `true` only if the answer to every assessment step referenced in `step-id-sequence` is "no"; the value of the predicate is `false` if the answer to any assessment step referenced by `step-id-sequence` is "yes" or "not applicable".
9. If the predicate matches the fourth line of the `predicate-no` production, the value of the predicate is `true` only if the answer to every assessment step in the Trustmark Definition is "no"; the value of the predicate is `false` if the answer to any assessment step in the Trustmark Definition is "yes" or "not applicable".
10. If the predicate matches the fifth line of the `predicate-no` production, the value of the predicate is `true` only if the answer to every assessment step in the Trustmark Definition is either "yes" or "not applicable"; the value of the predicate is `false` if the answer to any assessment step in the Trustmark Definition is "no".

11. If the predicate matches the first line of the predicate-na production, the value of the predicate is true only if the answer to the assessment step referenced by step-id is “not applicable”; the value of the predicate is false only if the answer to the assessment step referenced by step-id is "yes" or "no".
12. If the predicate matches the second line of the predicate-na production, the value of the predicate is true only if the answer to every assessment step from the assessment step referenced by the first step-id in the range to the assessment step referenced by the second step-id in the range, inclusive, is “not applicable”; the value of the predicate is false the answer to any assessment step in the range is “yes” or “no”.
13. If the predicate matches the third line of the predicate-na production, the value of the predicate is true only if the answer to every assessment step referenced in step-id-sequence is “not applicable”; the value of the predicate is false if the answer to any assessment step referenced by step-id-sequence is "yes" or "no".
14. If the predicate matches the fourth line of the predicate-na production, the value of the predicate is true only if the answer to every assessment step in the Trustmark Definition is “not applicable”; the value of the predicate is false if the answer to any assessment step in the Trustmark Definition is "yes" or "no".
15. If the predicate matches the fifth line of the predicate-na production, the value of the predicate is true only if the answer to every assessment step in the Trustmark Definition is either "yes" or "no"; the value of the predicate is false if the answer to any assessment step in the Trustmark Definition is “not applicable”.

For example, given a Trustmark Definition with exactly two assessment steps, the first with XML ID step1 and the second with XML ID step2, and given the answer to step1 is "yes" and the answer to step2 is "no", the following table describes the Boolean value of predicates derived from these steps.

Predicate	**Boolean Value**
step1	true
step2	false
yes(step1)	true
no(step1)	false
na(step1)	false
yes(step2)	false
no(step2)	true
na(step2)	false
yes(NONE)	false
no(NONE)	false
na(NONE)	true

Boolean algebra determines the value of the Boolean expression derived from these predicates and the Boolean operators.

Appendix C: Trust Expression Language for Trust Interoperability Profiles

The content of the `<TrustExpression>` element in a Trust Interoperability Profile is a Boolean expression that indicates whether an entity satisfies this Trust Interoperability Profile based on: (1) its possession of specific Trustmarks issued under the referenced Trustmark Definitions, (2) its satisfaction of specific logical comparisons based on parameter values within the Trustmarks it possesses, and (3) its satisfaction of the referenced Trust Interoperability Profiles. This appendix describes the syntax and semantics of the language used to write these Boolean expressions.

C.1 Extended Backus-Naur Form Grammar

The following grammar approximately expresses the syntax of the issuance criteria in Extended Backus-Naur Form (EBNF) [ISO 14977]. The grammar defines the trust expression as a Boolean expression consisting of the Boolean operators and a set of predicates.

```
trust-expr = predicate
           | trust-expr , ws , "and" , ws , trust-expr
           | trust-expr , ws , "or" , ws , trust-expr
           | "not" , ws , trust-expr
           | "(" , [ ws , ] trust-expr , [ ws , ] ")"
predicate  = tdr-id
           | tipr-id
           | comp-expr
comp-expr  = comp-term , [ ws , ] op , [ ws , ] comp-term
comp-term  = tdr-id , [ ws , ] "." , [ ws , ] param-name
           | constant
           | func , [ ws , ] "(" , [ ws , ] func-args , [ ws , ] ")"
func-args  = comp-term , [ "," , [ ws , ] func-args ]
tdr-id     = ? an ID for a
             TrustmarkDefinitionReferenceType element ?
tipr-id    = ? an ID for a
             TrustInteroperabilityProfileReferenceType element ?
op         = "==" | "!=" | ">" | "<" | ">=" | "<="
param-name = ? value of a ParameterName
             "name" attribute from a Trustmark ?
constant   = ? a constant value ?
func       = ? a name of a built-in function ?
ws         = ? white space characters ?
```

C.2 Semantics and Order of Operations for Evaluation of Boolean Operators

The Boolean operators are “and”, “or”, “not”, “(“, and “)”. The operator “and” is the binary logical conjunction operator. The operator “or” is the binary logical disjunction operator. The operator “not” is the unary logical negation operator. The parentheses operators are used for grouping. The order of operations is as follows.

1. “(“, “)”
2. “not”
3. “and”
4. “or”

C.3 Semantics for Evaluation of Boolean Predicates

A predicate may be an ID for a Trustmark Definition requirement, an ID for a reference to a Trust Interoperability Profile, or a Boolean comparison expression that can be computed based on the values of zero or more Trustmark parameters bound from Trustmarks possessed by the entity.

The Boolean value of a predicate is as follows.

1. If the predicate matches the `tdr-id` production, the value of the predicate is `true` only if the entity possesses a Trustmark issued under the Trustmark Definition referenced by the Trustmark Definition requirement and exactly one of the following is true.

    a. The Trustmark Definition requirement does not specify a Trustmark Provider reference; or

    b. The Trustmark Definition requirement references the Trustmark Provider that issued the Trustmark.

    Otherwise, the value of the predicate is `false`.

2. If the predicate matches the `tipr-id` production, the value of the predicate is `true` only if the value of the trust expression of the referenced Trust Interoperability Profile when applied to the entity is `true`. Otherwise, the value of the predicate is `false`.

3. If the predicate matches the `comp-expr` production, the value of the predicate is `true` only if the value of the referenced Boolean comparison expression is true when applied to the entity and bound to all referenced Trustmark parameter values. Otherwise, the value of the predicate is `false`.

Boolean algebra determines the value of the Boolean expression derived from these predicates and the Boolean operators.

C.4 Syntax and Semantics by Parameter Data Type

As noted in the EBNF grammar in Section C.1, trust expressions can include comparison operators and constant values. The following table provides semantic rules for constants and comparison operators by parameter data type.

Parameter **Data Type**	**Constants**	**Operator Semantics for** **"==" and "!="**	**Operator Semantics for** **">", "<", ">=", and "<="**
`NUMBER`	Values must conform to the XML `xs:decimal` data type.	Behavior follows numeric type comparison rules of the Java programming language.	
`STRING`	Values must be enclosed in quotation marks.	Perform standard equality/inequality comparisons. Comparisons for `STRING` values are case-sensitive.	Invalid; behavior undefined.
`BOOLEAN`	Values must conform to the XML `xs:boolean` data type.		
`DATETIME`	Values must conform to the ISO 8601 standard, must be enclosed in quotation marks, and must use the `datetime` built-in function.	Behavior follows logical comparison rules wherein prior/earlier dates and times are less than subsequent/later dates and times.	
`ENUM`	Values must be enclosed in quotation marks and conform to the `EnumValueType` simple type syntax as defined in Section 4.2.13.	Perform standard equality/inequality comparisons.	Invalid; behavior undefined.
`ENUM_MULTI`		Invalid when used with `ENUM_MULTI` values. Only the "`contains`" built-in function can be used with `ENUM_MULTI` values. See Section C.5.	

C.5 Semantics for Evaluation of Built-In Functions

As noted in the EBNF grammar in Section C.1, trust expressions can include certain built-in functions. Parser implementations for this trust expression language MUST support the following built-in functions.

Function Name	Description	Example
`exists`	Used to test for the existence of a specific named parameter within a specific type of Trustmark held by an entity.	`exists(TD.param_name)`
`contains`	Used to test for the existence of a specific parameter value for a specific named `ENUM_MULTI` parameter within a specific type of Trustmark held by an entity.	`contains(TD.param_name, param_value)`
`datetime`	Used to create a `DATETIME` constant from an ISO 8601 conformant string.	`datetime("2016-09-30")`

C.6 Examples

Each of the following examples illustrate valid syntax for a trust expression.

Example #1:

```
TD1 and (TD2 or TD3)
```

Example #2:

```
( TD1 or TD2 ) and ( TIP1 )
```

Example #3:

```
TD1 and exists(TD1.min_pw_len)
```

Example #4:

```
(TD1 or TD2) and not TIP2
```

Example #5:

```
TD1 and TD1.min_pw_len >= 10
```

Example #6:

```
TD2 and contains(TD2.sensitive_info_types,"HIPAA")
```

Example #7:

```
TD3 and TD3.certification_date > datetime("2015-12-31T23:59:59Z")
```

The examples shown here would produce valid Boolean results at runtime, under the following assumptions.

1. TD1, TD2, and TD3 are all IDs for Trustmark Definition requirements within the Trust Interoperability Profile that contains the <TrustExpression>.
2. TIP1 and TIP2 are IDs for references to Trust Interoperability Profiles within the Trust Interoperability Profile that contains the <TrustExpression>.
3. The Trustmark Definition referenced by TD1 contains a <ParameterDefinition> for a parameter with the name min_pw_len and the data type NUMBER.
4. The Trustmark Definition referenced by TD2 contains a <ParameterDefinition> for a parameter with the name sensitive_info_types, with data type ENUM_MULTI and an <EnumValue> of "HIPAA".
5. The Trustmark Definition referenced by TD3 contains a <ParameterDefinition> for a parameter with the name certification_date and the data type DATETIME.

Appendix D: Notices

The Georgia Tech Research Institute (GTRI) takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights.

GTRI invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this specification. Please address any such notices to the GTRI NSTIC Trustmark Pilot team at the following email address.

TrustmarkFeedback@gtri.gatech.edu