Marlin – Profile and Capability Signaling

Version 1.0
Final

Source	Marlin Engineering Workgroup
Date	July 18, 2008

Notice

Use of this document is subject to the agreement executed between you and the Parties, if any.

Any copyright notices shall not be removed, varied, or denigrated in any manner.

Intellectual Property

A commercial implementation of this specification requires a license from the Marlin Trust Management Organization.

Contact Information

Feedback on this specification should be addressed to: editor@marlin-community.com

Contact information for the Marlin Trust Management Organization can be found at: http://www.marlin-trust.com/

Contents

1 Introduction

This document describes a mechanism by which a Marlin client implementation can signal to a Marlin service the set of mandatory functions and optional features supported by the Marlin client implementation.

1.1 Document Organization

This document is organized as follows:

- Introduction and conventions
- Attribute definitions
- Signaling profile and binding

1.2 Conformance Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this specification are to be interpreted as described in IETF RFC 2119 [RFC2119].

These capitalized key words are used to unambiguously specify requirements and behavior that affect the interoperability and security of implementations. When these key words are not capitalized they are meant in their natural-language sense.

All elements of this specification are considered Normative unless specifically marked Informative. All Normative Elements are Mandatory to implement, except where such an element is specifically marked OPTIONAL. Finally, where Normative elements are described as OPTIONAL, they MAY be omitted from an implementation, but when implemented, they MUST be implemented as described.

1.3 Namespaces and Identifiers

This specification defines schemas conforming to XML Schemas [Schema] normative text to describe the syntax and semantics of XML-encoded objects and protocol messages. In cases of disagreement between the schema documents and the schema listings in this specification the schema documents take precedence. Note that in some cases the normative text of this specification imposes constraints beyond those indicated by the schema documents.

1.3.1 Namespaces and Notation

The table below summarizes the normative schemas defined by this specification, and their XML namespace [XMLns] URIs. These URIs MUST be used by implementations of this specification.

Prefix	XML Namespace
pacs:	urn:marlin:pacs

Table 1: Normative Namespaces

The table below summarizes the external schemas used in this specification:

Prefix	XML Namespace	Description
xsd:	http://www.w3.org/2001/XMLSchema	[Schema]

Prefix	XML Namespace	Description
xsi:	http://www.w3.org/2001/XMLSchema-instance	[Schema]
saml	urn:oasis:names:tc:SAML:1.0:assertion	[SAML1.1]
S11:	http://schemas.xmlsoap.org/soap/envelope	[SOAP11]

Table 2: Supporting Namespaces

As a convention throughout this document we use the namespace prefixes described above to qualify XML elements and attributes that are specified elsewhere. That is the typographical convention is: <MarlinElement>, <ns:ForeignElement>, XMLAttribute, Datatype, OtherKeyword.

1.4 References

1.4.1 Normative References

[MIAR]	Marlin Identifier and Attribute Registry (TBD)
[MBB]	Marlin Engineering Work Group, Marlin Broadband Delivery System Specification, version1.2 and latest errata
[MCS]	Marlin Engineering Work Group, Marlin – Core System Specification, version1.3 and latest errata
[RFC2119]	S. Bradner, Key words for use in RFCs to Indicate Requirement Levels, IETF RFC 2119, March 1997. http://www.ietf.org/rfc/rfc2119.txt
[SAML1.1]	Eve Maler, Prateek Mishra and Rob Philpott, eds., *Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V1.1,* http://www.oasis-open.org/committees/download.php/3405/oasis-sstc-saml-bindings-1.1.pdf
[Schema]	XML Schema Part 1: Structures. W3C Recommendation. D. Beech, M. Maloney, N. Mendelsohn, H. Thompson. May 2001. http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/ XML Schema Part 2: Datatypes W3C Recommendation. P. Biron, A. Malhotra. May 2001. http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/
[SOAP11]	"Simple Object Access Protocol (SOAP) 1.1," Box, Don, Ehnebuske, David , Kakivaya, Gopal, Layman, Andrew, Mendelsohn, Noah, Nielsen, Henrik Frystyk, Winer, Dave, eds. World Wide Web Consortium W3C Note (08 May 2000). http://www.w3.org/TR/2000/NOTE-SOAP-20000508/
[WS-SECSAML]	Phillip Hallam-Baker *et al.*, eds., *Web Services Security: SAML Token Profile,* OASIS Standard, December 2004, http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.0.pdf

2 Marlin Profile and Capability Signaling

This specification defines a general framework to provide Marlin client implementations with a mechanism to communicate support for optional functionality. The general notion is to be able to signal to a relying party the set of well-known Marlin profiles (mandatory functions) and optional features that the Marlin client implementation supports.

This framework is designed to support a case where a Marlin client implementation supports more than one Marlin profile. A Marlin client that implements more than one profile implies the Marlin client implements optional features in one profile that may be mandatory in the other. In this case, it is RECOMMENDED that Marlin client implementations signal the optional features for the less feature-rich profiles.

The framework is designed to be extensible. To this end, signaled identifiers are registered in [MIAR] §2.7 along with a normative reference to the implemented functionality indicated by the signal.

2.1 Attribute Names and Value-space (Informative)

A normative list of identifiers and attributes is maintained in [MIAR]. Table 3 exemplifies the attribute name identifiers and the values that can be signaled by this framework.

Attribute Name	Attribute Value-space
profile	Profile defined URI
topology	urn:marlin:bb:1-2:topology:any
protocols	urn:marlin:bb:1-2:dcs urn:marlin:bb:1-2:dus urn:marlin:mcs:1-3:licensexfer
features	urn:marlin:dmz:1-0

Table 3. Example of attribute value

2.2 Scope and Qualification of Attributes

The utility of the signaling mechanism is twofold. First, it enables a Marlin client implementation to unambiguously indicate the basic set of functionality it implements. This is accomplished by signaling a profile attribute as defined in §2.2.1. Secondly, this mechanism enables a Marlin client to indicate the optional features, qualified by the profile, that the Marlin client also implements. These profile qualified attributes are defined in §2.3.

2.2.1 profile Attribute Name

When signaling, the client is REQUIRED to supply the profile attribute. That is, all Implementations of Marlin profiles MUST signal this attribute using a profile specific URI.

2.3 profile Qualified Attributes

Attributes are scoped by the Marlin profile attribute defined in §2.2.1. One or more attributes defined in this section MAY be signalled along with the MANDATORY profile attribute. Signaling attributes in this manner indicates to the relying party the optional features implemented by the Marlin client. A Marlin client MUST NOT signal attributes deemed mandatory by the designated Marlin profile.

2.3.1 topology Attribute Name

Services represent the supported business models by implementing an Octopus Node/Link topology. Marlin delivery system specifications define services (e.g., Registration Service) that, through DRM Client interactions, issue Octopus Nodes and Links that formulate this topology. DRM Clients implementing a particular delivery system specification are required to support the prescribed topology. However, Octopus Nodes and Links are generic technologies, when implemented accordingly, enable services and clients to formulate and support other node/link topologies beyond the mandatory ones defined by the delivery system specification. This attribute is used to signal this DRM Client capability.

The topology attribute signals that the DRM Client implementation is capable of supporting optional node/link topologies that can be formulated with the Octopus Node types the delivery system specification supports. A DRM Client that asserts this capability indicates that the implementation conforms to all the mandatory protocol and processing requirements to support the optional topologies. That is, the DRM Client will:

- Conform to the requirements of Octopus Node and Link acquisition protocols, including Agent processing, as well as other Octopus Node and Link related protocols defined by the delivery system. For example, the license acquisition protocols (i.e., binding a license to an Octopus Node) or the deregistration protocol (i.e., termination of the link relationship between two Octopus Nodes).
- Enforce link constraints (i.e., securely execute control programs).
- Process and derive Scuba keys distributed within the Octopus Link Objects.

The value-space of the topology identifier MUST be a URI and it is RECOMMENDED that it include delivery system specification version information (e.g., urn:marlin:broadband:1-2:topology:any). This allows for future enhancements to the node/link topology prescribed by the delivery system specification.

2.3.2 protocols Attribute Name

This attribute signals a Marlin client implementation supports the designated communication protocol.

2.3.3 features Attribute Name

This attribute signals the optional features supported by a Marlin client implementation.

3 Signaling with SAML Attributes

Various Marlin specifications utilize SAML Assertions to convey trusted attributes about a system entity (typically a Nemo node). However the lifecycle and security properties of these assertions greatly differ from the requirements of the attributes signaled with this framework. Specifically the lifecycle of, and the type of information signaled with this framework are generally bound to the lifecycle of a client implementation, not its security posture. Therefore, this SAML Assertion Profile is defined here to support the different lifecycle requirements.

The following SAML Assertion Profile defines the mechanism by which a Marlin client implementation MUST convey the attributes defined in §2. That is, a qualified profile and any optional features the implementation supports beyond what is mandated by the qualified profile.

3.1 SAML Assertion Profile

3.1.1 Assertion Composition

The <saml:Assertion> MUST contain a <saml:AttributeStatement> element for each supported Marlin Profile.

3.1.2 AttributeStatement Composition

The <saml:AttributeStatement> MUST contain a single <saml:Subject> element and one or more <saml:Attribute> elements.

An <saml:AttributeStatement> MUST have one or more <saml:Attribute> elements. In the set of <saml:Attribute> elements one MUST signal the profile attribute. This profile attribute qualifies the remaining <saml:Attribute> elements within the <saml:AttributeStatement>.

If a Marlin client supports more than one profile then it MUST communicate this fact in a separate <saml:AttributeStatement> elements.

3.1.3 Subject Identification

The <saml:Subject> element MUST contain a single <saml:NameIdentifier> element to identify the entity for which attributes apply. The Format XML attribute of the <saml:NameIdentfier> element SHOULD be;

http://nemo.intertrust.com/2004/saml/name-format/uri

The value of the <saml:NameIdentifier> element MUST be a URI. This URI SHOULD be managed under the “urn:marlin:organization” namespace. The <saml:NameIdentifer> SHOULD reflect the same identity as the subject of the client’s NEMO Client Keys.

3.1.4 Attribute Naming

For this assertion profile, the AttributeNamespace XML attribute in all <saml:Attribute> elements MUST be;

urn:marlin:pacs

The AttributeName XML attribute in the <saml:Attribute> elements MUST be one of the attribute names defined in §2.

3.1.5 Attribute Values

The schema type of the contents of the <saml:AttributeValue> element MUST be indicated with the xsi:type attribute. All attribute values MUST be represented as xsd:string. Multiple values for an attribute MUST be represented with multiple <saml:AttributeValue> elements. An example follows:

```
<Attribute AttributeNamespace="urn:marlin:pacs"
        AttributeName="protocols">
   <AttributeValue xsi:type=”xsd:string”>urn:marlin:bb:1-2:dcs
   </AttributeValue>
   <AttributeValue xsi:type=”xsd:string”>urn:marlin:bb:1-2:dus
   </AttributeValue>
</Attribute>
```

The value space of the <saml:AttributeValue> element SHOULD be one or more of the attribute values defined [MIAR].

3.1.6 Assertion Signature

The <saml:Assertion> MAY be independently signed. When signed, the guidance given in [MCS] §12.2 and [SAML1.1] §5 SHALL be followed.

If signed, the signer of the assertion MAY be either the authority that signs the DRM Client Role Attribute Assertion [MCS] or the DRM Client with its NEMO Client Signing Key.

4 NEMO Basic Secure Message Binding

These attribute assertions SHOULD be conveyed in the Request Message of each NEMO Basic Secure Message exchange.

The assertion SHALL be placed as a direct child element of a <wsse:Security> element, and SHALL be referenced from a <wsse:KeyIdentifier> element in a <wsse:SecurityTokenReference> element, as specified in [WS-SECSAML] §3.3.

The <wsse:SecurityTokenReference> element that references these SAML attribute assertions SHALL contain a nemosec:Usage attribute with the following value:

```
urn:marlin:pacs:1.0:profile-capability-attributes:assertion
```

5 SAML Profile and Capability Example (Informative)

5.1 Signaling Mandatory and Optional Features

```
<Assertion xmlns="urn:oasis:names:tc:SAML:1.0:assertion"
           xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           AssertionID="AQAjCGNl" IssueInstant="2007-06-19T18:36:47.942Z"
           Issuer=" urn:marlin:organization:phony:model"
           MajorVersion="1" MinorVersion="1">
  <AttributeStatement>
    <Subject>
      <NameIdentifier
          Format=" http://nemo.intertrust.com/2004/saml/name-format/uri">
        urn:marlin:organization:phony:model:pd-x1:firmware:v1.0
      </NameIdentifier>
    </Subject>
    <!-- The MANDATORY profile attribute -->
    <Attribute AttributeNamespace="urn:marlin:pacs"
                    AttributeName="profile">
      <AttributeValue xsi:type="xsd:string">
        urn:marlin:profile:jimmyjet:1-0
      </AttributeValue>
    </Attribute>
    <!-- Signal additional protocols implemented by the client -->
    <Attribute AttributeNamespace="urn:marlin:pacs"
                    AttributeName="protocols">
      <AttributeValue xsi:type="xsd:string">urn:marlin:bb:1-2:dcs
      </AttributeValue>
      <AttributeValue xsi:type="xsd:string">urn:marlin:bb:1-2:dus
      </AttributeValue>
    </Attribute>
    <!-- Signal implementation can understand any nodes types -->
    <Attribute AttributeNamespace="urn:marlin:pacs"
            AttributeName="topology">
      <AttributeValue xsi:type=”xsd:string”>
          urn:marlin:broadband:1-2:any
      </AttributeValue>
    </Attribute>
</AttributeStatement>
</Assertion>
```