Authors:
Blair Dillaway, Microsoft
Marty Humphrey, UVA
Chris Smith, Platform (Editor)
Glenn Wasson, UVA

12/13/2006

HPC Basic Profile, Version 0.2

Status of this Memo

This memo provides information to the Grid community regarding the specification of the HPC Basic Profile. Distribution is unlimited.

Copyright Notice

Abstract

This document defines the HPC Basic Profile, consisting of a set of non-proprietary specifications, along with clarifications, refinements, interpretations and amplifications of those specifications which promote interoperability. The single use-case addressed in this Profile is the “Base Case” (Section 2) of [HPC-U].

Contents

1 Introduction

The HPC Basic Profile is a document that is used to describe how a particular set of specifications are composed in order to solve a basic use case around the use of HPC systems [refer to use case document]. The single use-case addressed in this Profile is the “Base Case” (Section 2) of [HPC-U].

The Profile consists of references to existing specifications, along with any clarifications of the contents of those specifications, restrictions on the use of those specifications, and references to any normative extensions to those specifications. While it is envisioned that many systems will have capabilities above and beyond those described in this profile, this profile describes a basic set of capabilities that can be used as the basis of interoperability testing between systems claiming compliance.

The document is structured as a set of sections, each of which is used to reference a particular aspect of an HPC Basic Profile compliant system. The first is that of job description, which references the Job Submission Description Language, version 1.0 [JSDL10] and the HPC Profile Application Extension [JSDLHPC]. The second is job scheduling and management, which references the OGSA Basic Execution Services specification [BES10].

It is worth noting that this profile is focused on describing the basic capabilities that must be supported by a compliant system. In many cases, the systems in question will support higher levels of functionality than described here, and many systems will support various extensions to the functionality described in the referenced specifications. It is not the goal of this profile to prohibit the use of such extentions, but to define a set of capabilities that can provide a basis for interoperability. As such, this profile may implicitly allow the use of various constructs, but not make any statement about the semantics of such use, and thus these constructs should not be used as the basis of any interoperability testing of HPC Basic Profile compliant systems.

2 Notational Conventions

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

The document refers to an “HPC Basic Profile compliant system” as a “Compliant system”.

This specification uses namespace prefixes throughout; they are listed in Table 2-1. Note that the choice of any namespace prefix is arbitrary and not semantically significant.

Table 2-1: Prefixes and namespaces used in this specification.

Prefix	Namespace
xsd	http://www.w3.org/2001/XMLSchema

3 Job Description

This section describes restrictions and clarifications to the Job Submission Description Language, version 1.0 [JSDL10] and the HPC Profile Application Extension [JSDLHPC] specifications.

The following elements within a JSDL document MUST be supported by a Compliant system. For the purposes of this document, supporting an element has a stronger meaning than with [JSDL10]. In order to support an element, a Compliant system must not only parse the element, but must accept the element as part of the JSDL job definition, and apply the semantics as indicated by the referenced specification with any clarifications or restrictions as described in this section.

JSDL documents MAY include additional elements from [JSDL10] beyond those listed in this section. A Compliant system MAY support any such additional elements should it encounter them in a submitted JSDL document. However, a Compliant system MAY also instead return a JSDL UnsupportedFeature fault in response to encountering any such additional elements from [JSDL10].

3.1 JobDefinition

As in [JSDL10].

3.2 JobDescription

A Compliant system MUST support the JobIdentification, JobName, Application, and Resources sub-elements.

3.2.1 JobIdentification

A Compliant system MUST support the JobName and JobProject sub-elements.

3.2.2 JobName

As in [JSDL10].

3.2.3 JobProject

As in [JSDL10].

3.2.4 Application

A Compliant system MUST support the BasicHPCApplication sub-element, as defined in [JSDLHPC].

3.2.5 Resources

A Compliant system MUST support the following sub-elements within the Resources element: CandidateHosts, ExclusiveExecution, OperatingSystem, CPUArchitecture, IndividualCPUCount, IndividualPhysicalMemory, IndividualVirtualMemory, TotalCPUCount and TotalResourceCount.

The IndividualCPUCount, IndividualPhysicalMemory, IndividualVirtualMemory, TotalCPUCount and TotalResourceCount sub-elements all MUST support non-negative integer values of the jsdl:exact element from the jsdl:RangeValue_Type. They MAY support non-integer values and they MAY support other jsdl:RangeValue_Type elements, but MAY instead return a Not_Supported fault in response to encountering such elements.

3.2.5.1 CandidateHosts

The CandidateHosts complex type will be supported as described in [JSDL10].

3.2.5.2 ExclusiveExecution

As in [JSDL10], with the clarification that the resources being allocated to the job are “hosts”. That is, if a job runs exclusively on a host, then no other jobs may run concurrently on the same host.

3.2.5.3 OperatingSystem

The OperatingSystem complex type will be supported as described in [JSDL10].

3.2.5.4 CPUArchitecture

The CPUArchitecture complex type will be supported as described in [JSDL10].

3.2.5.5 IndividualCPUCount

The description is as in [JSDL10]. A Compliant system MUST support integer values of the jsdl:exact element from the jsdl:RangeValue_Type for this element's value, as described in more detail above.

3.2.5.6 IndividualPhysicalMemory

As in [JSDL10], with the clarification that this element refers to a requirement of an activity being described and represents the amount of physical memory (i.e. RAM) that the activity expects to need. A Compliant system MUST support integer values of the jsdl:exact element from the jsdl:RangeValue_Type for this element's value, as described in more detail above.

3.2.5.7 IndividualVirtualMemory

As in [JSDL10], with the clarification that this element refers to a requirement of an activity being described and represents the amount of virtual memory that the activity expects to need. A Compliant system MUST support integer values of the jsdl:exact element from the jsdl:RangeValue_Type for this element's value, as described in more detail above.

3.2.5.8 TotalCPUCount

The description is as in [JSDL10]. A Compliant system MUST support integer values of the jsdl:exact element from the jsdl:RangeValue_Type for this element's value, as described in more detail above.

3.2.5.9 TotalResourceCount

The description is as in [JSDL10]. A Compliant system MUST support integer values of the jsdl:exact element from the jsdl:RangeValue_Type for this element's value, as described in more detail above.

4 Job Scheduling and Management Services

This section describes restrictions and clarifications to the OGSA Basic Execution Services specification [BES10].

A Compliant system MUST support the BES base case specification. It MAY additionally support BES extension profiles.

The BES GetActivitiesStatus, TerminateActivities, and GetJSDLDocuments operations include a vector input parameter that specifies the set of activities that the operation should be applied to. A Compliant system MUST support a vector length of 1. A Compliant system SHOULD support input vector lengths greater than 1 but MAY return a Not_Supported fault in response to input vector lengths greater than 1.

5 Security Considerations

In this section, we define interoperable security mechanisms which HPC Basic Profile compliant implementations must support. The mechanisms defined are limited to those necessary to address the requirements of the "Base Case" (Section 2) of [HPC-U]. The "Common Cases" (Section 3) of [HPC-U] are *not* explicitly supported in this document.

5.1 Security Requirements of the HPC Basic Profile

In this section, we first describe the environment in which an HPC Basic Profile service/client will operate, and then identify the requirements for securing the HPC Basic Profile messages.

5.1.1 Environment Assumptions

In addressing the Base Case some common assumptions are made about the environment and relationships between the users and BES web service schedulers. The security mechanisms defined in this specification build on this environment.

1. There is an identity management infrastructure deployed for provisioning users and services with identity credentials.
    - Web services are provisioned with X.509 service certificates following industry standard practice.
    - It is only required that users be provisioned with username-password credentials (possibly linked to an organizational Kerberos infrastructure). Organizations may, but are not required, to provision users with other credentials such as X.509 certificates. If an organization uses X.509 client certificates, username-password credentials may additionally be utilized but are not required.
2. Trust relationships are pre-configured and uniform
    - Users trust the CA(s) issuing X.509 service certificates and services trust the authority provisioning username-password credentials or the CA(s) issuing X.509 user certificates.
    - All BES Web services are fully trusted with respect to managing and executing activities within the environment.
    - Users may not fully trust each other. They may require their activities be free from tampering by other users, or in some cases that the details of their activities (job type, data source, ..) not be exposed to other users.
3. X.509 certificate revocation may be supported using industry standard mechanism such as CRLs and OCSP responders. It is up to the relying party whether to take advantage of revocation information.
4. It is assumed BES services are well-known to users and other services and may be located using commonly deployed mechanisms such as DNS or UDDI look-ups.
5. Authorization is based on authenticated user/service identities and attributes carried in the provisioned identity credentials. The authorization mechanism employed is outside the scope of this specification.

5.1.2 Securing the HPC Profile Messages

There is a need to secure messages exchanged between the users and schedulers to support the Base Case. The security mechanisms must support required message sender authentication (BES requests and responses), integrity protection, and/or confidentiality. These are summarized below:

BES Request Message Authentication – BES services require authentication of clients (or user or service) invoking their services to ensure only authorized actions are performed. This includes, limiting who may create an activity, cancel an activity, and query an activity's status.

BES Response Message Authentication – Entities requesting BES services will require authentication of response sender. This is needed to ensure that returned status information or faults can be relied upon. It is not generally required that a BES service be authenticated by a client prior to making a request, i.e., there is no danger to the client in first authenticating to the service.

Integrity Protection – Some form of high assurance message integrity is necessary to prevent attackers from modifying activity definitions for purposes such as creating incorrect billing or denial of service.

Confidentiality - For some of the use cases, confidentiality about activity details and status are not considered confidential. As such, it is not mandatory to encrypt the BES messages to prevent disclosure via captured messages. Encryption must be an option to support those cases where the activity details are important.

5.2 HPC Basic Profile Message Security

This specification takes the position that security interoperability for the use cases of interest is best achieved through use of a few widely deployed, standards-based, technologies and vetted implementation guidance. It is not a goal of this specification to innovate in the security area or drive adoption of new technologies.

To that end, use of transport layer security as the basis for interoperable secure messages is adopted. This provides more functionality that absolutely required for some environments, but minimizes the number of mechanisms which must be supported. It is not believed the tools, and supporting infrastructure, for interoperable message level security (based on the WS-* family of specifications) have reached the level of adoption and deployment needed to require their use as part of this specification.

The HPC Basic Profile builds on the WS-I Basic Security Profile BSP) [WS-I Basic Security Profile Version 1.0, Working Group Draft,2006-08-17] as the foundation for interoperable message security profile. In particular, the transport layer security mechanisms identified in Section 4 of that specification are used. The more restrictive usage guidelines specified in the "OGSA Basic Security Profile 1.0 - Secure Channel" are also adopted as described below.

(Note: The "OGSA Basic Security Profile 1.0 - Core" specification is not used as that addresses the binding of key information to an endpoint reference [in WS-Addressing], which is not relevant when using transport layer security.)

The HPC Basic Profile requires compliance with the following security requirements. The terminology of the WS-I BSP is used to define compliant implementations. Specifically, a conforming INSTANCE is "software that implements a wsdl:port or a uddi:bindingTemplate".

```
R0501: An INSTANCE MUST support TLS 1.0, SHOULD support SSL 3.0,
      and SHOULD support TLS 1.1.
R0502: An INSTANCE MUST support the FIPS-140 compliant
      ciphersuites.
R0503: An INSTANCE MUST support TLS_RSA_WITH_AES_128_CBC_SHA.
R0504: An INSTANCE MUST support service authentication using
      X.509 certificates using RSA cryptographic keys and the
       SHA-1 digest algorithm
R0505: An INSTANCE MUST support either client authentication
      using username/password credentials or X.509 certificates
      using RSA cryptographic keys and the SHA-1 digest algorithm
R0506: An INSTANCE must use TLS/SSL encryption key agreement
      based on the RSA algorithm. Diffie-Helman key agreement
      shall not be used.
R0507: Client authentication based on username/password must use
      a password digest and conform to the Web Services Security
      Username Token Profile 1.1.
```

5.3 X.509 Certificate Based Mutual Authentication

When supporting mutual authentication based on X.509 certificates, it will be done in accordance with the recommendations of WS-I BSP and the more restrictive guidance of the OGSA-BSP Secure Channel specification. This specification requires support for X.509 v3 client and service certificates only.

5.4 Username-Password Client Authentication

When supporting username-password client authentication, a secure TLS/SSL session with the BES service must be first established. This is done in conformance to the requirements stated

above and the WS-I BSP. That is, service authentication is done using an X.509 service certificate and a channel encryption key negotiated using RSA key transport.

Once an encrypted and integrity protected transport layer channel has been established, the client may send one of the HPC Basic Profile supported request messages including their username-password authentication information as specified in the Username Token Profile 1.1 specification [XXX].

Since this information is communicated within a secure transport layer, we do not specify a protocol for negotiating a nonce or other values to prevent replay attacks, and in accordance with the WS-I BSP, the nonce and creation-time fields are omitted from the digest calculation.

An example CreateActivity message, including a username and digest password is shown below.

```
<s11:Envelope
  xmlns:s11="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:bes-“factory=”http://schemas.ggf.org/bes/2006/08/bes-factory”
  xmlns:wsse=”http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd”
  xmlns:wsu=”http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd” >
  <s11:Header>
    <wsse:Security>
      <wsse:UsernameToken xmlns:wsse='http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd' >
        <wsse:Username>Bert</wsse:Username>
        <wsse:Password  Type='http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordDigest'>
             B5twk47KwSrjeg==
        </wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
    <wsa:Action>
       http://schemas.ggf.org/bes/2006/08/bes-
factory/GetActivitiesStatus
    </wsa:Action>
    <wsa:To s11:mustUnderstand=1>
        http://www.bes.org/BESFactory
    </wsa:To>
  </s11:Header>
  <s11:Body wsu:Id='TheBody'>
        <bes-factory:CreateActivity>
            <bes-factory:activityDescriptionDocument>
                <bes-factory:ActivityDocument>
                    {Any valid JSDL document}
                </bes-factory:ActivityDocument>
            </bes-factory:activityDescriptionDocument>
        </bes-factory:CreateActivity>
  </s11:Body>
</s11:Envelope>
```

6 Author Information

Blair Dillaway
Microsoft Corp.

Marty Humphrey
University of Virginia

Chris Smith
Platform Computing, Inc.

Marvin Theimer
Microsoft Corp.

Glenn Wasson
University of Virginia

7 Contributors

We gratefully acknowledge the contributions made to this specification by [insert names].

8 Acknowledgements

We are grateful to numerous colleagues for discussions on the topics covered in this document, in particular (in alphabetical order, with apologies to anybody we've missed) [insert names].

We would like to thank the people who took the time to read and comment on earlier drafts. Their comments were valuable in helping us improve the readability and accuracy of this document.

Full Copyright Notice

Intellectual Property Statement

Normative References

[RFC 2119] Bradner, S. *Key words for use in RFCs to Indicate Requirement Levels*. Internet Engineering Task Force, RFC 2119, March 1997. Available at http://www.ietf.org/rfc/rfc2119.txt

[JSDL10] Available at http://www.ggf.org/documents/GFD.56.pdf