Authors:
Blair Dillaway, Microsoft
Marty Humphrey, University of Virginia
Chris Smith, Platform (Editor)
Marvin Theimer, Microsoft
Glenn Wasson, University of Virginia

2/10/2007

HPC Basic Profile, Version 1.0

Status of this Document

This document 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 High Performance Computing (HPC) systems. 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 extensions, 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
jsdl	http://schemas.ggf.org/jsdl/2005/11/jsdl
jsdl-hpcpa	http://schemas.ggf.org/jsdl/2006/07/jsdl-hpcpa
bes-factory	http://schemas.ggf.org/bes/2006/08/bes-factory
hpcp-bp	http://schemas.ogf.org/hpcp/2007/01/bp

3 Claiming Conformance

Claims of conformance to the HPC Basic Profile 1.0 can be made using the following mechanisms, as described in Conformance Claim Attachment Mechanisms, when the applicable Profile requirements associated with the listed targets have been met:

The conformance claim URI for the Basic Profile 1.0 is as follows, as per the discussions in the "Security Considerations" section of this document:

- Username Token - "http://ogf.org/profiles/hpc-basic/1.0/username-token"
- X.509 Certificate Token - "http://ogf.org/profiles/hpc-basic/1.0/x.509-certificate-token"

A claim of conformance MUST be made with at least one of these two tokens. A claim of conformance MAY be made with both of these two tokens. In addition, a claim of conformance MUST be made for the WS-I basic profile (http://ws-i.org/profiles/basic/1.1).

4 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 BES UnsupportedFeatureFault in response to encountering any such additional elements from [JSDL10].

4.1 JobDefinition

As in [JSDL10].

4.2 JobDescription

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

4.2.1 JobIdentification

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

4.2.2 JobName

As in [JSDL10].

4.2.3 JobProject

As in [JSDL10].

4.2.4 Application

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

4.2.5 Resources

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

4.2.5.1 CandidateHosts

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

4.2.5.2 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.

4.2.5.3 OperatingSystem

The jsdl:OperatingSystem complex type will be supported as described in [JSDL10]. If the consuming system does not provide the requested operating system, or if the JSDL special token “other” is used as the content of the jsdl:OperatingSystemName sub-element, and if the consuming system does not understand the provided extension content, then the consuming system MAY return the BES UnsupportedFeatureFault to the requester.

4.2.5.4 CPUArchitecture

The CPUArchitecture complex type will be supported as described in [JSDL10]. If the consuming system does not provide the requested CPU architecture, or if the JSDL special token “other” is used as the content of the jsdl:CPUArchitectureName sub-element, and if the consuming system does not understand the provided extension content, then the consuming system MUST return the BES UnsupportedFeatureFault to the requester.

4.2.5.5 TotalCPUCount

The description is as in [JSDL10]. A Compliant system MUST support non-negative integer values of the jsdl:Exact element from the jsdl:RangeValue_Type. It MUST support positive integer values of the jsdl:UpperBoundRange and jsdl:LowerBoundRange, and MUST support the exclusiveBound attribute on these elements. It MAY support non-integer values, it MAY support the epsilon attribute of jsdl:Exact, and it MAY support the jsdl:Range element, but MUST instead return a BES UnsupportedFeatureFault in response to encountering such elements.

5 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.

5.1 BES Vector Operations

The BES GetActivitiesStatus, TerminateActivities, and GetActivityDocuments 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 BES UnsupportedFeatureFault in response to input vector lengths greater than 1.

5.2 FactoryResourceAttributesDocument contents

The bes-factory:FactoryResourceAttributesDocument, as returned by the GetFactoryAttributesDocument operation, includes a list of activities currently managed by the BES as well as a list of contained resources that are allocated for the use of these activities. If the numbers of activities or contained resources gets large, then the corresponding size of this document can also be quite

large. Given that repeated requests for this document could incur a large cost for both clients and servers in transferring and parsing this document, the BES MAY choose not to return the ActivityReference or ContainedResource sub-elements of the bes-factory:FactoryResourceAttributesDocument on a request by request basis.

In order to distinguish between the absence of any activities being managed by the BES, and the BES implementation choosing not to return the ActivityReference sub-elements, the BES MUST provide the number of managed activities in the TotalNumberOfActivities sub-element of the bes-factory:FactoryResourceAttributesDocument.

In order to distinguish between the absence of any contained resources available to the BES, and the BES implementation choosing not to return the ContainedResource sub-elements, the BES MUST provide the number of available contained resources in the TotalNumberOfContainedResources sub-element of the bes-factory:FactoryResourceAttributesDocument.

5.3 BasicFilter extension

Since there are cases when a client explicitly requires the complete list of both activities or contained resources, the BES MAY support the hpcp-bp:BasicFilter extension element within the content of the bes-factory:GetFactoryAttributesDocumentType. A BES that chooses to support this extension MUST return a BESExtension sub-element of bes-factory:FactoryResourceAttributesDocument containing the URI "http://schemas.ogf.org/hpcp/2007/01/bp/BasicFilter", and MUST provide the following semantics when encountering a hpcp-bp:BasicFilter element in the bes-factory:GetFactoryAttributesDocumentType.

The hpcp-bp:BasicFilter has the structure (the normative schema is provided in Appendix A):

```
<hpcp-bp:BasicFilter>
    <ActivityReferences> true|false </ActivityReferences>
    <ContainedResources> true|false </ContainedResources>
</hpcp-bp:BasicFilter>
```

There are four possible cases:

1. If both the ActivityReferences and the ContainedResources sub-elements are false in the BasicFilter, then the BES MUST NOT return either ActivityReference or ContainedResource sub-elements in the bes-factory:FactoryResourceAttributesDocument.
2. If the ActivityReferences sub-element is true and the ContainedResources sub-element is false in the BasicFilter, then the BES MUST return an ActivityReference sub-element for each activity managed by the BES, and the BES MUST NOT return any ContainedResource sub-elements in the bes-factory:FactoryResourceAttributesDocument.
3. If the ActivityReferences sub-element is false and the ContainedResources sub-element is true in the BasicFilter, then the BES MUST NOT return any ActivityReference sub-elements, and the BES MUST return a ContainedResource sub-element for each contained resource available to the BES in the bes-factory:FactoryResourceAttributesDocument.
4. If both the ActivityReferences and ContainedResources sub-elements are true in the BasicFilter, then the BES MUST return an ActivityReference sub-element for each activity managed by the BES, and the BES MUST return a ContainedResource sub-element for each contained resource available to the BES.

6 Security Considerations

This section defines interoperable security mechanisms that HPC Basic Profile compliant implementations must support. These mechanisms are limited to those necessary to address the requirements of the "Base Case" (Section 2) of [HPC-U]. Compliant implementations MAY support

additional security mechanisms required for extended functionality as discussed in Section 3 of [HPC-U].

6.1 Security Requirements of the HPC Basic Profile

The environment in which an HPC Basic Profile service and client will operate is described below along with the requirements for securing the HPC Basic Profile messages.

6.1.1 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 [RFC 3280] service certificates following industry standard practice.
    - It is required that users be provisioned with username-password credentials or X.509 certificates. If an organization uses X.509 client certificates, username-password credentials may also 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 and safeguarding any confidential user and activity information.
    - 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 [RFC 3280] and OCSP [RFC 2560] 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 (Domain Name Service) or UDDI (Universal Description Discovery and Integration) 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.

6.1.2 Securing the HPC Profile Messages

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

BES Request Message Authentication – BES services require authentication of clients (may be a user or other 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 the responding service. This is needed to ensure that returned status information or faults can be relied upon.

Integrity Protection – 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 - In some environments, activity details and status information will be considered confidential. As such, it will be mandatory to encrypt the BES messages to prevent disclosure to unauthorized entities. Confidentiality of this information may not be critical in other environments, though message encryption is still acceptable.

6.2 HPC Basic Profile Message Security

This specification takes the position that security interoperability for the Base Use Case is best achieved through 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 TLS/SSL transport layer security as the basis for interoperable secure messages is adopted. This provides greater 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 rely on their use as the primary security mechanism for this profile.

The HPC Basic Profile builds on the “WS-I Basic Security Profile” [WS-I BSP] as the foundation for interoperable message security. In particular, the transport layer security mechanisms identified in Section 4 of that specification are used.

The HPC Basic Profile message security mechanisms and requirements are defined in Section 6.3 and 6.4. Compliant implementations are required to fully implement one of these mechanisms, though they may support both. The terminology of the WS-I BSP is used to define compliant implementations. Specifically, a conforming INSTANCE is software which implements the mechanisms defined in this specification; identifies its conformance using the mechanism defined in Section 3; and "implements a wsdl:port or a uddi:bindingTemplate". The last requirement follows terminology of the WS-I BSP used to define compliant implementations.

6.3 TLS/SSL using X.509 Certificate Based Mutual Authentication

This specification supports use of the Transport Layer Security (TLS 1.0 [RFC 2246] and TLS 1.1 [RFC 4346]) or Secure Sockets Layer (SSL 3.0) protocol for BES message security with mutual authentication of the sender and receiver based on X.509 v3 certificates. This is done in accordance with the recommendations of WS-I BSP. Faults shall be handled in accordance with the TLS/SSL specifications.

Specific requirements of this specification are:

R0631: An INSTANCE MUST support TLS 1.0, SHOULD support SSL 3.0, and SHOULD support TLS 1.1.

R0632: An INSTANCE MUST support the FIPS-140 compliant Ciphersuites TLS_RSA_FIPS-WITH_3DES_EDE_CBC_SHA and TLS_RSA_FIPS_WITH_3DES_EDE_CBC_SHA

R0633: An INSTANCE SHOULD support TLS_RSA_WITH_AES_128_CBC_SHA and TLS_RSA_WITH_AES_128_CBC_SHA.

R0634: An INSTANCE MUST support X.509 v3 certificates using RSA cryptographic keys and RSA/SHA-1 (http://www.w3.org/2000/09/xmldsig#**rsa-sha1)** digital signatures.

R0635: An INSTANCE SHOULD support X.509 v3 certificates using RSA cryptographic keys and RSA/SHA-256 (http://www.w3.org/2001/04/xmldsig-more#rsa-sha256) digital signatures.

R0636: An INSTANCE MUST use TLS/SSL encryption key agreement based on the RSA algorithm. Diffie-Hellman key agreement SHALL NOT be used.

R0637: An INSTANCE MUST support server authentication using X.509 v3 certificates.

6.4 TLS/SSL with Username-Password Client Authentication

This specification supports use of the TLS or SSL protocol for BES message security with X.509 server authentication and username-password based client authentication. When using this mechanism, a secure TLS/SSL session with the BES service must be first established. This is done in conformance with the recommendations contained in the WS-I BSP and requirements R0631 through R0637 above. 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 transmit an HPC Basic Profile supported request messages, including their username-password authentication information as specified in the Username Token Profile 1.1 specification [WSS-UP].

Specific requirements of this specification are:

R0641: An INSTANCE MUST support client authentication using username/password credentials with cleartext (http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText) type encoding.

R0642: An INSTANCE MAY support client authentication using username/password credentials with digest (http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordDigest) type encoding.

Since all password information is communicated within a secure transport layer by compliant implementations, this specification does not specify use of message-level encryption. Also, use of nonces or creation times to prevent replay attacks is not required by this specification and these may be omitted from a password digest calculation.

Faults occurring during TLS/SSL negotiation shall be handled in accordance with the TLS/SSL specifications. If faults arise based on processing of the clients username-password credential by the service, the service may silently drop the request message or respond with a SOAP fault message. When responding with a fault message, if the service is unable to validate the supplied credentials a SOAP fault with faultcode 'Client' should be returned otherwise a fault with faultcode 'Server' shall be returned. Compliant BES service implementations may wish to implement mechanisms to limit the number of invalid authentication attempts for a given username to prevent password guessing attacks.

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#PasswordText">Ernie</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
    <wsa:Action>
       http://schemas.ggf.org/bes/2006/08/bes-factory/CreateActivity
    </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:ActivityDocument>
                <jsdl:JobDefinition>
                    {Any valid JSDL document}
                </jsdl:JobDefiniton>
            </bes-factory:ActivityDocument>
        </bes-factory:CreateActivity>
  </s11:Body>
</s11:Envelope>
```

7 Author Information

Blair Dillaway
Software Architect, Microsoft Corporation
One Microsoft Way, Redmond, WA 98052
blaird@microsoft.com

Marty Humphrey
Department of Computer Science
University of Virginia
Charlottesville, VA 22094
humphrey@cs.virginia.edu

Chris Smith
Platform Computing Inc.
3760 14th Avenue
Markham, Ontario
Canada L3R 3T7
csmith@platform.com

Marvin Theimer
Microsoft Corporation
One Microsoft Way, Redmond, WA 98052
theimer@microsoft.com

Glenn Wasson
Department of Computer Science
University of Virginia
Charlottesville, VA 22094
wasson@virginia.edu

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): Jim Basney, Rich Ciapala, Ian Foster, Andrew Grimshaw, Hiro Kishimoto, Peter Lane, Allessandro Maraschini, Satoshi Matsuoka, Mark Morgan, Steven Newhouse, Bill Nitzberg, Stephen Pickles, Andreas Savva, Von Welch, Luigi Zangrando, Liang Zhong.

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.

9 Intellectual Property Statement

10 Disclaimer

11 Full Copyright Notice

12 References

[BES10] I. Foster, A. Grimshaw, P. Lane, W. Lee, S. Newhouse, S. Pickles, S. Pulsipher, C. Smith, M. Theimer. *OGSA Basic Execution Service Version 1.0*. GFD-R. Mar 17 2007.

[HPC-U] M. Theimer, C. Smith, and M. Humphrey. *HPC Job Scheduling: Base Case and Common Cases*. GFD-I.100. July 1, 2006.

[JSDL10] A. Anjomshoaa, F. Brisard, M. Drescher, D. Fellows, A. Ly, S. McGough, D. Pulsipher, and A. Savva (ed.) *Job Submission Description Language (JSDL) Specification, Version 1.0,* Global Grid Forum, Lemont, Illinois, U.S.A., GFD-R.56, 7 November 2005.

[JSDLHPC] M. Humphrey, C. Smith, M. Theimer, and G. Wasson. *JSDL HPC Profile Application Extension, V1.0.* GFD-R. Oct 2, 2006.

[RFC 2119] S. Bradner. *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

[RFC 2246] T. Dierks, C. Allen. *The TLS Protocol Version 1.0*. Internet Engineering Task Force, RFC 2246, January 1999. Available at http://www.ietf.org/rfc/rfc2246.txt

[RFC 2560] M. Myers, R. Ankney, A. Malpani, S. Galperin, C. Adams. *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP.* Internet Engineering Task Force, RFC 2560, March 1999. Available at http://www.ietf.org/rfc/rfc2560.txt

[RFC 3820] S. Tuecke, V. Welch, D. Engert, L. Pearlman, M. Thompson.*Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile.* Internet Engineering Task Force, RFC 3820, June 2004. Available at http://www.ietf.org/rfc/rfc3820.txt

[RFC 4346] T. Dierks, E. Rescorla. *The Transport Level Security (TLS) Protocol Version 1.1.* Internet Engineering Task Force, RFC 4346, April 2006. Available at http://www.ietf.org/rfc/rfc4346.txt

[WSI BSP] WS-I Basic Security Profile Version 1.0, Working Group Draft,2006-08-17.

[WSS-UP] Web Services Security UsernameToken Profile, Working Draft 2, OASIS, 23 Feb 2003.

Appendix 1 HPC Basic Profile XML Schema

```
<xsd:schema
    targetNamespace="http://schemas.ogf.org/hpcp/2007/01/bp"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:hpcp-bp="http://schemas.ogf.org/hpcp/2007/01/bp"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified">

    <!-- Filter Types -->
    <xsd:complexType name="BasicFilterType">
        <xsd:sequence>
            <xsd:element name="ActivityReferences" type="xsd:boolean"/>
            <xsd:element name="ContainedResources" type="xsd:boolean"/>
            <xsd:any namespace="##other" processContents="lax"
                minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:anyAttribute namespace="##other" processContents="lax"/>
    </xsd:complexType>

    <xsd:element name="BasicFilter" type="hpcp-bp:BasicFilterType"/>
</xsd:schema>
```