

**Prescription Monitoring Program Information Exchange**

**( PMIX )**

***Secure*  
Service Interface Description  
Version 1.1**

**July 2013**

**Sponsored by:**



**CONTENTS**

[1 Document Introduction 5](#_Toc362000274)

[2 Background 6](#_Toc362000275)

[3 Physical Model 7](#_Toc362000276)

[4 Service Interaction Requirements 9](#_Toc362000277)

[5 Interface Description Requirements 10](#_Toc362000278)

[6 Message Exchange Patterns 16](#_Toc362000279)

[7 Message Definition Mechanisms 16](#_Toc362000280)

[8 Policies and Contracts 16](#_Toc362000281)

[9 Interface Security 17](#_Toc362000282)

[10 Interface Privacy 18](#_Toc362000283)

[12 Service Testing 19](#_Toc362000284)

[13 Appendix A — References 20](#_Toc362000285)

[14 Appendix B — Acronyms 20](#_Toc362000286)

[15 Appendix C — Document History 21](#_Toc362000287)

**FIGURES**

[Figure 1: Physical Model Overview 7](#_Toc362000317)

[Figure 2: Secure Interface Scope 8](#_Toc362000318)

[Figure 3: WS-Addressing Overview 11](#_Toc362000319)

[Figure 4: Message Level Security 12](#_Toc362000320)

[Figure 4: Message Level Security 17](#_Toc362000321)

[Figure 5: Transport Level Encryption 18](#_Toc362000322)

**TABLES**

[Table 1: Service Interaction Requirements 9](#_Toc362000332)

[Table 2: Interface Description Requirements 10](#_Toc362000333)

**Acknowledgements**

This document was produced as part of the BJA-sponsored IJIS Institute project in support of the Prescription Monitoring Program Information Exchange. Major portions of this document were derived directly from the earlier PMIX Project documents and significant contributions were made by the following organizations:

Office of Justice Programs Bureau of Justice Assistance

Prescription Drug Monitoring Program Training and Technical Assistance Center (PDMP TTAC)

IJIS Institute Prescription Monitoring Program (PMP) Committee

Open Networks, Inc.

This project was supported by Grant No. 2009-PM-BX-K044 awarded by the Bureau of Justice Assistance. The Bureau of Justice Assistance is a component of the Office of Justice Programs, which also includes the Bureau of Justice Statistics, the National Institute of Justice, the Office of Juvenile Justice and Delinquency Prevention, and the Office for Victims of Crime. Points of view or opinions in this document are those of the author and do not represent the official position or policies of the United States Department of Justice.

# Document Introduction

A Service Interface Description describes the physical implementation or service interface used in a specific implementation of a service. This document is a Service Interface Description for the Prescription Monitoring Program Information Exchange (PMIX) Secure Interface.

In the context of a service-oriented architecture (SOA) and, more specifically, the Global Reference Architecture (GRA), a service is the means by which one partner gains access to one or more capabilities offered by another partner. Capabilities generate real-world effects that can be as simple as sharing information or can involve performing a function or changing the state of some other processes. Organizations have numerous capabilities and partner organizations. There are significant benefits for these organizations to have access to each other's capabilities. Each state has its own governance, business needs, applications, hardware, and networks. Achieving interoperability within this environment, and with external partner organizations, calls for alignment of business and technical requirements and capabilities by specifying them consistently across organizational boundaries. The GRA was developed to facilitate interoperability and to assist in meeting other key requirements in an information sharing environment. In order to achieve this goal, there is a strong need to define a consistent approach to identifying and describing services and their interactions that can be implemented in many different technical environments, across multiple government lines of business, at all levels of government and with other partner organizations. GRA web services are based on the Organization for the Advancement of Structured Information Standards (OASIS) Web Services Profiles.

The GRA defines a service interface as "the means for interacting with a service.” It includes the specific protocols, commands, and information exchange by which actions are initiated on the service. A service interface is what a system designer or implementer (programmer) uses to design or build executable software that interacts with the service. That is, the service interface represents the “how” of the interaction. Since the service interface is the physical manifestation of the service, best practices call for a service interface which can be described in an open-standard format (that is, a format which could be automatically processed by a computer).

A Service Specification is a formal document describing the capabilities made available through the service; the service model that defines the semantics of the service by representing its behavioral model, information model, and interactions; the policies that constrain the use of the service, and the service interface that provides a means of interaction with the service. A Service Specification provides stakeholders with an understanding of the structure of the service and the applicability to it interface rules. It gives the service consumers the information necessary for consuming a particular service, and service providers the information necessary for implementing the service in a consistent and interoperable way. The main components of a Service Specification are the Service Description and one or more Service Interface Descriptions. A Service Description describes all aspects of a service that are not directly tied to the physical implementation or service interface.

# Background

All states have laws and regulations that govern the distribution and handling of controlled substances and other pharmaceuticals. Diversion of, and addiction to, such substances are generally recognized as serious problems throughout the country. States have found that prescription monitoring programs (PMPs), which collect prescription-controlled-substance dispensing data from pharmacies and other authorized dispensers, can be effective tools for identifying and preventing these problems. (In the remainder of this document, the term “prescription” will refer to controlled-substance prescriptions.) These data are then reviewed and analyzed for medical, educational, public health and investigative purposes.

States implementing new PMP systems may find it desirable to conform closely to the PMIX Service Specification in order to help ensure consistency with potential interstate exchange partners in the future.

**Guiding Principles**

Several core principles have been utilized to guide the development of this document and other PMIX artifacts:

* **Distributed data sources:** assumes distributed, rather than centralized, information sources.
* **Maintenance of state-level controls:** PMIX implementation will not impact or modify a state’s control over the operation of the PMP and authorization to access prescription data.
* **End-to-end security:** Mechanisms must be utilized to assure the security of PMIX in-transit data between the sending and receiving endpoints.
* **Information traceability:** PMIX data flows will leave an audit trail, not to include protected health information (PHI), to enable reporting on demand to PMP administrators.
* **Standards-based information sharing:** PMIX standards will leverage open industry standards such as Extensible Markup Language (XML) and the National Information Exchange Model (NIEM) for encoding data to ensure maximal interoperability between future exchange partners.

# Physical Model

This Secure service interface implements the ProvidePrescriptionDrugHistory operation, which is compliant with the GRA Reliable Secure Web Service - Service Interaction Profile (SIP) and the PMIX Execution Context. The service naming is defined from the perspective of the service provider rather than the service consumer or client.

The diagram below shows the physical model including a hub (within the server playing the role of number “3”) which forms the PMIX Execution Context. The PMIX service interface is provided by an Application Program Interface (API), within the server playing the roles of number “1” and number “5”) along with a PMIX State Routing Service (SRS) (within the server playing the roles of number “2” and number “4”). The blue icon on top of number “1” and the blue arrows represent clear text. The red arrows indicate encrypted data. The icons next to numbers “2” and “4” represent security certificates.



Figure 1: Physical Model Overview

The PMIX Secure Service Interface Description Document focuses only on the Secure interface portion of the Physical Model as highlighted in Figure 2: Secure Interface Scope.



Figure 2: Secure Interface Scope

A PMP System has several options when connecting to PMIX via the secure web service interface, including standards based WS-Security, standards based WS-Security with additional XML Encryption and standards based WS-Security with additional PMPi Encryption.

# Service Interaction Requirements

The table below outlines the service interaction requirements and the associated standards that are being used to support this service interface.

|  |  |  |
| --- | --- | --- |
| Requirements | Mandatory | Specification |
| Service Consumer Authentication | *Yes* | **Static IP Address via Firewall** |
| Service Consumer Authorization | *Yes* | **Application specific roles** |
| Identity and Attribute Assertion Transmission | *No* | ***N/A*** |
| Service Authentication | *Yes* | **SSL Certificate Authentication** |
| Message Nonrepudiation | *Yes* | **W3C XML Signature** |
| Message Integrity | *Yes* | **W3C XML Signature** |
| Message Confidentiality | *Yes* | **OASIS Security Profile 1.1**  **with XML Encryption**  **Transport Layer Security** |
| Message Addressing | *Yes* | **WS-Addressing** |
| Reliability | *Yes* | **Implicitly provide by response** |
| Transaction Support | *No* | ***N/A*** |
| Service Metadata Availability | *No* | ***N/A*** |
| Interface Description Requirements | *Yes* | **WSDL 1.1** |

Table 1: Service Interaction Requirements

# Interface Description Requirements

The ProvidePrescriptionDrugHistory operation provides state PMP systems with the capability to retrieve interstate prescription drug history. The API calls and the associated parameters are provided below. The location of each parameter in the SOAP message is also provided in the last column.

**ProvidePrescriptionDrugHistory Operation:**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Description | Input or output | Location |
| MetaData   * Requestor * Requestor Role * RoutingData   + RequestID   + RequestingState   + DisclosingState * RequestProfileName | The Requestor Role is used for authorization of search requests  RequestID is a unique message identifier with the first two characters corresponding to the requesting state postal code  Requesting/disclosing state two character postal code  The Request Profile Name element reflects the search profile used in the RequestData body message | API input parameters | SOAP header |
| RequestData | XML request message consistent with the PMIX NIEM IEPD | API input parameter | SOAP body |
| ResponseStatus | Provided, NotFound, Deferred | API return parameter | SOAP header |
| ResponseData | XML response message consistent with the PMIX NIEM IEPD | API output parameter | SOAP body |

Table 2: Interface Description Requirements

**Addressing**

The PMIX Secure Interface web service requires the WS-Addressing “To” element to contain the proper address to enable routing to the endpoint PMP system through any number of PMIX compliant intermediate hub systems. Figure 3, WS-Addressing Overview, below depicts the addressing conventions used for accessing the PMIX web services using WS-Addressing.



Figure 3: WS-Addressing Overview

**Security Options**

1. **Standard WS-Security**

The PMIX Secure service interface supports dual privacy protection and message security mechanisms. First, the communication interface is protected and encrypted using a transport level security binding that implements secure socket layer (SSL). Additionally, WS-Security is used to provide end-to-end message level security by signing and encrypting the exchanges between the states.



Figure 4: Message Level Security

The following example depicts the WS-Security header used by the PMIX Secure Interface.

<s:Envelope xmlns:s=<http://www.w3.org/2003/05/soap-envelope>

xmlns:a="http://www.w3.org/2005/08/addressing">  
<s:Header>  
...  
<o:Security s:mustUnderstand="1"   
 xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">  
 <u:Timestamp u:Id="uuid-5c69e42a-5b2c-4b08-bc0a-2c3d71d11b9a-5"   
 xmlns:u="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">  
 <u:Created>2012-02-15T16:13:07.908Z</u:Created>  
 <u:Expires>2012-02-15T16:18:07.908Z</u:Expires>  
 </u:Timestamp>  
 <o:BinarySecurityToken>  
 <!-- Removed-->  
 </o:BinarySecurityToken>  
 <e:EncryptedKey Id="\_0" xmlns:e="http://www.w3.org/2001/04/xmlenc#">  
 <e:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p">  
 <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"   
 xmlns="http://www.w3.org/2000/09/xmldsig#"></DigestMethod>  
 </e:EncryptionMethod>  
 <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">  
 <o:SecurityTokenReference>  
 <o:KeyIdentifier ValueType="http://docs.oasis-open.org/wss/oasis-wss-soap-message-security-1.1#ThumbprintSHA1" EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary">S7nuqp3KsqErQq0k2hkS9DrdtnE=</o:KeyIdentifier>  
 </o:SecurityTokenReference>  
 </KeyInfo>  
 <e:CipherData>  
 <e:CipherValue>VXljqtcV/lyS1 ... VZQ=</e:CipherValue>  
 </e:CipherData>  
 <e:ReferenceList>  
 <e:DataReference URI="#\_2"></e:DataReference>  
 <e:DataReference URI="#\_5"></e:DataReference>  
 </e:ReferenceList>  
 </e:EncryptedKey>  
 <e:EncryptedData Id="\_5" Type="http://www.w3.org/2001/04/xmlenc#Element" xmlns:e="http://www.w3.org/2001/04/xmlenc#">  
 <e:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes256-cbc">

</e:EncryptionMethod>  
 <e:CipherData>  
 <e:CipherValue>HOZgoh+30b ... A==</e:CipherValue>  
 </e:CipherData>  
 </e:EncryptedData>  
 </o:Security>  
</s:Header>

<s:Body u:Id="\_1" xmlns:u="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">  
 <e:EncryptedData Id="\_2" Type="http://www.w3.org/2001/04/xmlenc#Content" xmlns:e="http://www.w3.org/2001/04/xmlenc#">  
 <e:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes256-cbc">

</e:EncryptionMethod>  
 <e:CipherData>  
 <e:CipherValue>Dc7eZIVlRW0H+ba ... A==</e:CipherValue>  
 </e:CipherData>  
 </e:EncryptedData>  
</s:Body>  
</s:Envelope>

The <e:EncryptedData> element contains the three components required to access to the token used to encrypt the actual data.

<e:EncryptedData Id="\_2" Type="http://www.w3.org/2001/04/xmlenc#Content" xmlns:e="http://www.w3.org/2001/04/xmlenc#">

<e:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes256-cbc">

</e:EncryptionMethod>  
 <e:CipherData>  
 <e:CipherValue>Dc7eZIVlRW0H+ba ... A==</e:CipherValue>  
 </e:CipherData>  
</e:EncryptedData>

First, the <e:EncryptionMethod> contains the method used to encrypt the token.

<e:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes256-cbc" />

In this case, it is **AES-256-cbc—a symmetric key encryption algorithm**—in which both parties use the same key for encryption and decryption. The symmetric key is randomly generated by the originator and is called a transient key. This is encrypted and stored in the<e:EncryptedKey> element, which is wrapped by a <KeyInfo> element.

Asymmetric encryption is sufficiently slower than symmetric encryption; therefore the WS-Security specification utilizes a symmetric algorithm to encrypt the data. The encryption solution also sends the symmetric key, which has been encrypted with the token. Encrypting only the key with the asymmetric algorithm substantially lowers the amount of processing required at the server.

The symmetric key is encrypted with the replying party’s public key (from their certificate). The transient key’s encryption method is found in the <e:EncryptionMethod>  element.

In this case, **the RSA-OAEP-MGF1P algorithm** is used for encryption. The key for this operation is found in the following <KeyInfo> element.

<KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">  
 <o:SecurityTokenReference>  
 <o:KeyIdentifier ValueType="http://docs.oasis-open.org/wss/oasis-wss-soap-message-security-1.1#ThumbprintSHA1" EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary"> S7nuqp3KsqErQq0k2hkS9DrdtnE=

</o:KeyIdentifier>  
 </o:SecurityTokenReference>  
 </KeyInfo>

The <o:KeyIdentifier> allows the sender to inform the relying party which certificate’s key was used to encrypt the symmetric key, by putting its **thumbprint (base64 encoded)** into the element. The replying party then uses its private key to decrypt the data in the <e:CipherData> - <e:CypherValue> element. After retrieving the symmetric key, the token itself is decrypted using the symmetric key.

For additional information, please refer to the underlying specification documents:

**Web Services Security: SOAP Message Security 1.1**

<http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-os-SOAPMessageSecurity.pdf>

1. **WS-Security with additional XML-Encryption**

The “WS-Security with optional XML-Encryption” interface alternative combines the standard WS-Security protocol attributes, defined in Security Option 1, Standard WS-Security,” above, with an extra layer of message encryption based on the XML-Encryption 1.1 specification. XML Encryption 1.1 is a W3C specification that defines how to represent encrypted data and key material in an XML framework. The PMIX implementation of XML-Encryption the 1.1 specification utilizes the secure AES256-GCM symmetric block encryption algorithm (http://www.w3.org/2009/xmlenc11#aes256-gcm) along with the RSA Optimal Asymmetric Encryption Padding (OAEP) standard (<http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p>). The resulting cipher values, associated keying information and related algorithms are stored in <EncryptedData> and <EncryptedKey> elements of the XML.

Refer to the XML Encryption 1.1 specification for additional information.

1. **WS-Security with optional PMPi-Encryption**

The “WS-Security with optional PMPi-Encryption” interface alternative combines the standard WS-Security protocol attributes, defined in Security Option 1, Standard WS-Security,” above, with the ad-hoc PMPi envelope and encryption mechanism defined in the proprietary PMPi specification. The PMPi specification defines a custom security solution for cryptography, a nonstandard key management mechanism and a proprietary access control solution. The custom cryptography solution uses a basic key encapsulation mechanism (KEM) that uses an RSA asymmetric (public-key) encryption algorithm to secure symmetric cryptographic key material and data for transmission. The PMPi encryption specification defines the symmetric Advanced Encryption Standard (AES) 256-bit algorithm operating in the CBC (Cipher Block Chaining) mode. The wrapped key along with the encrypted data are then transported in a PMPi specific XML envelope format.

Refer to the PMPInterconnect Technical Specification package for additional information.

# Message Exchange Patterns

The PMIX service interface uses a request/response “message exchange pattern” (referred to as the MEP in the standards specifications) to provide simple processing in the case that the prescription drug history is readily available.

# Message Definition Mechanisms

All information exchanges are conformant to the National Information Exchange Model (NIEM) V2.0.

The PMIX service interface messages are defined using a NIEM IEPD. The PMIX service interface is largely defined by the WSDLs. The PMIX Administrator can provide the PMIX IEPD and Service Specification Package artifacts.

The PMIX exchange incorporates the concept of message metadata, i.e. data about the data or the exchange, e.g. requestor name and role. The message metadata is maintained separate from the prescription drug history request and response. The message metadata is not subject to the same privacy concerns and may be recorded and retained.

# Policies and Contracts

The presumption made throughout this document is that individual state legislation and particular state-to-state agreement(s) will address issues of privacy and security (beyond the transport and message-level security described below).

## Policies

No automated polices have been identified at this time.

## Automated Service Contracts

Web Services Description Language (WSDL) defines an automated service contract..

No automated service contracts exist beyond those specified in the WSDLs.

## Non-automated Service Contracts

No non-automated service contracts have been identified at this time.

## Umbrella Agreements

No umbrella agreements have been identified at this time.

# Interface Security

The PMIX hub supports transport level security using X.509 certificates when communicating with state systems. In a typical configuration with a state using a Microsoft platform, the SRS would establish the transport level security with the hub.

In addition to transport level security with static IP addresses, message level security is used to sign and encrypt exchanges between the states (see Sequence Diagram).



Figure 4: Message Level Security

# Interface Privacy

PMIX exchanges are encrypted for the entire transmission between states (including at the hub). In addition, transport level encryption is also used when information is traversing the public network.



Figure 5: Transport Level Encryption

# Service Testing

The PMIX Service Testing must follow the test policies and procedures reflected in the PMIX Service Conformance Package. In particular, the PMIX Service Conformance Specification describes the conformance testing required to verify interoperability with the Prescription Monitoring Program Information Exchange (PMIX) Service. The document provides the testing information required for government and industry organizations to engage in the conformance testing process.

The Service Conformance Specification document identifies the conformance targets, associated requirements and the related test suite / test cases (with assertions) for both aspects of the PMIX information exchange architecture. As such, government and industry organizations will identify the conformance targets and the test suites applicable to them depending on their role in the PMIX information exchange.

In addition to documenting the conformance targets, requirements and assertions, the Service Conformance Specification also provides guidelines for using the Conformance Test Tool. The Conformance Test Tool, soapUI, is an open-source, cross-platform, web service test application, which provides a number of features that will facilitate the PMIX conformance testing and reporting process. soapUI provides well established test scenarios which issue requests, receive responses, apply assertions and generate fully automated, built-in conformance test reports.

Refer to the PMIX Service Conformance Package for additional information.

# Appendix A — References

* PMIX Service Description v1.1.0, July 2013
* PMIX Secure Service Interface Description v1.1.0, July 2013
* PMIX Execution Context Document v1.0, April 2012
* PMIX NIEM 2.0 Information Exchange Package Documentation
* ijis\_pmix\_III\_niem\_2\_0\_iepd\_v0\_20\_20101207
* PMIX Service Conformance Package v1.0
* GRA Web Services Service Interaction Profile v1.3
* pmp-interconnect-technical-specification-v03.00.01

# Appendix B — Acronyms

API Application Programming Interface

GFIPM Global Federated Identity and Privilege Management

GRA Global Reference Architecture

NIEM National Information Exchange Model

OASIS Organization for the Advancement of Structured Information  
Standards

PMIX Prescription Drug Monitoring Information Exchange

RS WS SIP GRA Reliable Secure Web Services Service Interaction Profile

SOAP Simple Object Access Protocol

SRS State Routing Service

W3C World Wide Web Consortium

WCF Windows Communication Framework

W/S Web Services

WSDL  Web Services Description Language

XML [eXtensible Markup Language](http://en.wikipedia.org/wiki/XML)

# Appendix C — Document History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Editor | Change |
| 1/2011 | 1.0 Rev B | Bob Slaski, Open Networks | For PMP Committee review |
| 3/2011 | 0.1 | Scott Serich, IJIS Institute | Incorporating PMP Committee feedback |
| 5/2011 | 0.2 | Scott Serich, IJIS Institute | Incorporating additional PMP Committee feedback |
| 6/2011 | 0.3 | Scott Serich, IJIS Institute | Incorporating final PMP Committee feedback |
| 7/2011 | 1.0 DRAFT | Scott Serich, IJIS Institute | Technical editing pass |
| 6/2012 | 1.0.1 | Bob Slaski, Open Networks | Updated to SSG 1.0 |
| 9/2012 | 1.0.2 | Bob Slaski, Open Networks | Updated to reflect community feedback |
| 7/2013 | 1.1.0 | Todd Seymour, Open Networks | Revisions to reflect consolidation of State-to-Hub and Hub-to-Hub specifications. |