

## **SAGA API Extension: Remote Procedure Calls, Version 2**

### Status of This Document

This document provides information to the grid community, proposing a standard for an extension to the Simple API for Grid Applications (SAGA). As such it depends upon the SAGA Core API Specification [1]. This document is intended to be used as input to the definition of language specific bindings for this API extension, and as reference for implementors of these language bindings. Distribution of this document is unlimited.

### Copyright Notice

Copyright © Open Grid Forum (2007). All Rights Reserved.

### Abstract

This document specifies an updated version of the SAGA Remote Procedure Calls, which in particular includes the data handle capabilities as defined by the OGF GridRPC group.

## **Contents**

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>SAGA Service Discovery API</b>	<b>3</b>
2.1	Introduction . . . . .	3
2.2	Specification . . . . .	3
<b>3</b>	<b>Intellectual Property Issues</b>	<b>6</b>
3.1	Contributors . . . . .	6

3.2	Intellectual Property Statement . . . . .	6
3.3	Disclaimer . . . . .	7
3.4	Full Copyright Notice . . . . .	7
<b>References</b>		<b>8</b>

## 1 Introduction

## 2 SAGA Service Discovery API

### 2.1 Introduction

### 2.2 Specification

---

```
package saga.rpc {

    enum io_mode
    {
        In      = 1, // input parameter
        Out     = 2, // output parameter
        InOut   = 3  // input and output parameter
    }

    enum data_mode
    {
        Volatile      = 1, // don't keep
        Sticky        = 2, // keep
        StickyReturn   = 3, // keep and copy to client
        Persistent    = 4  // keep and migrate
        PersistentReturn = 5 // keep and migrate and copy to client
    }

    class remote_parameter : extends saga::parameter
                          extends saga::async
                          // from parameter saga::buffer
                          // from buffer    saga::object
                          // from object    saga::error_handler
    {
        CONSTRUCTOR (in    url          src    = "",
                     in    url          tgt    = "",
                     in    io_mode      iomode = In,
                     in    data_mode    dmode  = Volatile,
                     in    array<byte>  data   = "",
                     in    int          size   = 0,
                     out   buffer       obj);

        set_data_mode (in    data_mode    dmode);
        get_data_mode (out   data_mode    dmode);

        set_src_url   (in    url          src);
    }
}
```

---

```

        get_src_url    (out    url            src);

        set_tgt_url    (in     url            tgt);
        get_tgt_url    (out    url            tgt);

        // manage data
        unbind          (void);
        retrieve        (in     url            src);
        store           (in     url            tgt);
    }

    // replaces saga::rpc from the SAGA Core API specification
    class rpc : implements saga::object
                implements saga::async
                implements saga::permissions
                // from object saga::error_handler
    {
        CONSTRUCTOR (in     session            s,
                     in     saga::url          url = "",
                     out    rpc                obj          );
        DESTRUCTOR  (in     rpc                obj          );

        // rpc method invocation
        call         (inout array<remote_parameter> parameters);

        // handle management
        close        (in     float             timeout = 0.0);
    }
}

```

---

### 2.2.1 Specification Details

#### Enum data\_mode

The `data_mode` enum specifies the storage properties of the `rpc::remote_parameter` instances:

##### Volatile

the paramater data are not stored on server sideafter computation

##### Sticky

the parameter data are stored on the rpc server after the rpc call finishes, and can be re-used for subsequent calls. That implies that `InOut` and `Out` parameter get their `src` url set and pointed to the intermediate results, but do not have a copy of the data stored.

#### **StickyReturn**

As `Sticky`, but a copy of the data are returned after the call, i.e., `InOut` and `Out` parameter have a copy of the data, and their `src` url is set and points to the intermediate results. The data remain at that single service location.

#### **Persistent**

As `Sticky`, but the data can be migrated to other service instances as needed.

#### **PersistentReturn**

As `Persistent`, but a copy of the data are returned after the call, i.e., `InOut` and `Out` parameter have a copy of the data, and their `src` url is set and points to the intermediate results.

#### **Class `remote_parameter`**

The `parameter` class inherits the `saga::parameter` class, adds an additional readonly state attribute: `data_mode`, and two additional URLs, `srs` and `tgt`. The `remote_parameter` uses these additional information to allow for data persistency between subsequent RPC calls, which can significantly improve performance.

#### **Class `rpc`**

This class replaces the `saga::rpc` class from the SAGA Core API specification [1], but now accepts both arrays of `saga::parameter` and `saga::remote_parameter` for rpc calls. The implementation is responsible to manage the data persistency for the remote parameters, according to their `data_mode` and `src` and `tgt` URLs.

As this class does not add any syntax to the original `rpc` class, no detailed specification is given here.

### **2.2.2 Examples**

## 3 Intellectual Property Issues

### 3.1 Contributors

This document is the result of the joint efforts of several contributors. The authors listed here and on the title page are those committed to taking permanent stewardship for this document. They can be contacted in the future for inquiries about this document.

**Andre Merzky**  
**andre@merzky.net**  
Center for Computation and Technology  
Louisiana State University  
216 Johnston Hall  
70803 Baton Rouge  
Louisiana, USA

In particular, the document build heavily on the specifications of the OGF GridRPC Working Group – we want to thank Eddy Caron, Craig Lee, Hidemoto Nakata and Yusuke Tanimura for their input and cooperation.

### 3.2 Intellectual Property Statement

The OGF 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. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the OGF Secretariat.

The OGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the OGF Executive Director.

### 3.3 Disclaimer

This document and the information contained herein is provided on an "As Is" basis and the OGF 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 of merchantability or fitness for a particular purpose.

### 3.4 Full Copyright Notice

Copyright (C) Open Grid Forum (2007). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the OGF or other organizations, except as needed for the purpose of developing Grid Recommendations in which case the procedures for copyrights defined in the OGF Document process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the OGF or its successors or assignees.

## References

- [1] T. Goodale, S. Jha, H. Kaiser, T. Kielmann, P. Kleijer, A. Merzky, J. Shalf, and C. Smith. A Simple API for Grid Applications (SAGA). Grid Forum Document GFD.90, 2007. Open Grid Forum.