XML-RPC

INTRODUCTION TO XML-RPC, A SIMPLE XML-BASED RPC MECHANISM

Peter R. Egli
INDIGOO.COM

Contents

1. What is XML-RPC?

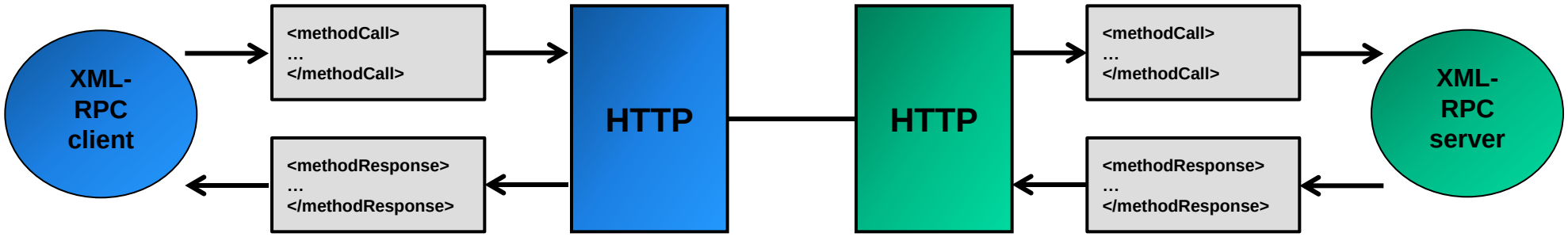
XML-RPC is a remote procedure call protocol using XML as data format and HTTP as transport protocol.

Advantages of XML-RPC:

- **Simple mechanism to call remote procedures on a machine with a different OS.**
- **XML-RPC is language and platform independent. XML-RPC libraries are available in Java and other languages (e.g. .Net: http://xml-rpc.net/).**
- **XML-RPC is not more than its name implies and thus is very simple and lean (very short specification, see http://xmlrpc.scripting.com/spec.html).**

Protocols and techniques behind XML-RPC:

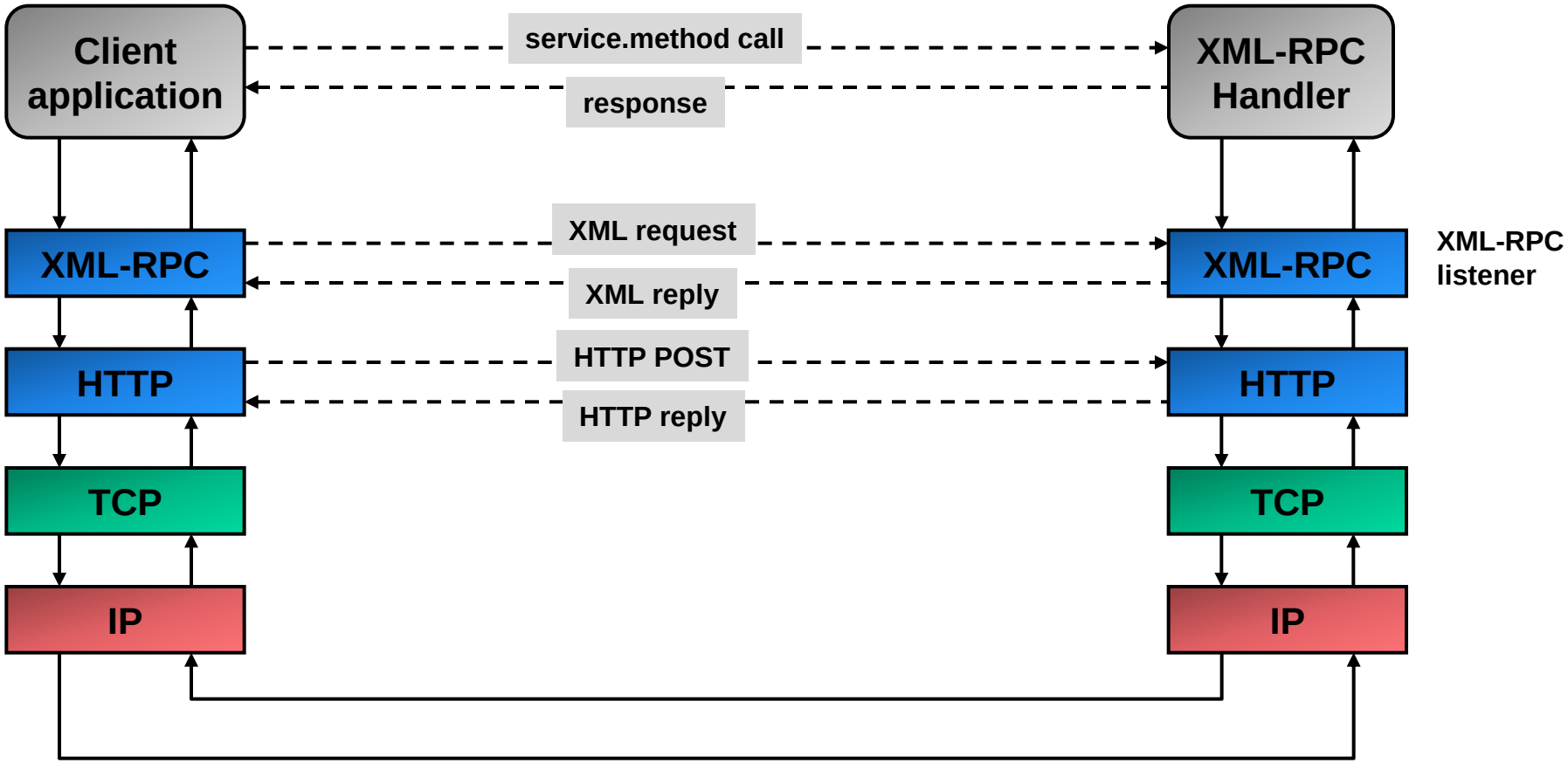
1. XML - Formatting of the request and response and the arguments (wire protocol)
2. RPC - Remote call of procedures.
3. HTTP - Transport protocol for the XML („firewall-friendly“).



2. XML-RPC architecture

The client application accesses the server through a URL (= location where service resides).

The XML-RPC listener receives requests and passes these to the handler (= user defined class servicing the request).



3. XML-RPC protocol (1/5)

Request (example from www.xmlrpc.com/spec):

- The XML body of the HTTP request contains a single method call (`getStateName`).
- The method is called on a service name under which the method is available.
- The URL does not need to be specified (service is indicated by the string before the dot in the method name element).

```
POST /RPC2 HTTP/1.0
User-Agent: Frontier/5.1.2 (WinNT)
Host: betty.userland.com
Content-Type: text/xml
Content-length: 181
```

HTTP header (request) with required header fields (blue)

```
<?xml version="1.0"?>
<methodCall>
  <methodName>examples.getStateName</methodName>
  <params>
    <param>
      <value>
        <i4>41</i4>
      </value>
    </param>
  </params>
</methodCall>
```

List of request parameters

XML body

3. XML-RPC protocol (2/5)

Response (example from www.xmlrpc.com/spec):

- **The HTTP return code is always „200 OK“, even in case of an XML-RPC fault (see below).**
- **The response contains a single value (like a return argument of a local procedure call).**

```
HTTP/1.1 200 OK
Connection: close
Content-Length: 158
Content-Type: text/xml
Date: Fri, 17 Jul 1998 19:55:08 GMT
Server: UserLand Frontier/5.1.2-WinNT
```

HTTP header (response) with required header fields (blue)

```
<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <value>
        <string>South Dakota</string>
      </value>
    </param>
  </params>
</methodResponse>
```

List of return values

XML body

3. XML-RPC protocol (3/5)

Response with a failure (example from www.xmlrpc.com/spec):

- **Even in case of an XML-RPC level failure the HTTP return code is 200 OK.**
- **The failure is indicated with the `<fault>` element.**

```
HTTP/1.1 200 OK
Connection: close
Content-Length: 426
Content-Type: text/xml
Date: Fri, 17 Jul 1998 19:55:02 GMT
Server: UserLand Frontier/5.1.2-WinNT
```

```
<?xml version="1.0"?>
<methodResponse>
  <fault>
    <value>
      <struct>
        <member><name>faultCode</name><value><int>4</int></value></member>
        <member><name>faultString</name>
          <value><string>Too many parameters.</string>
          </value>
        </member>
      </struct>
    </value>
  </fault>
</methodResponse>
```

3. XML-RPC protocol (4/5)

Parameter types (1/2):

Base types:

XML-RPC has a very limited set of base types:

Type	*Description*	*Example*
`<i4> or <int>`	Four-byte signed integer	-12
`<boolean>`	0 (false) or 1 (true)	1
`<string>`	ASCII string (string is default)	Hi!
`<double>`	Double-precision	3.1415
`<dateTime.iso8601>`	Date/time	19980717T14:08:55
`<base64>`	Base64-encoded binary	eW91IGNhbid

Structs:

Structs contain elements (<member>) with a <name> and <value> element ("named" values).
Structs may be recursive (value in a struct may be a struct again).

```
<struct>
  <member>
    <name>lowerBound</name>
    <value><i4>18</i4></value>
  </member>
  <member>
    <name>upperBound</name>
    <value><i4>139</i4></value>
  </member>
</struct>
```

3. XML-RPC protocol (5/5)

Parameter types (2/2):

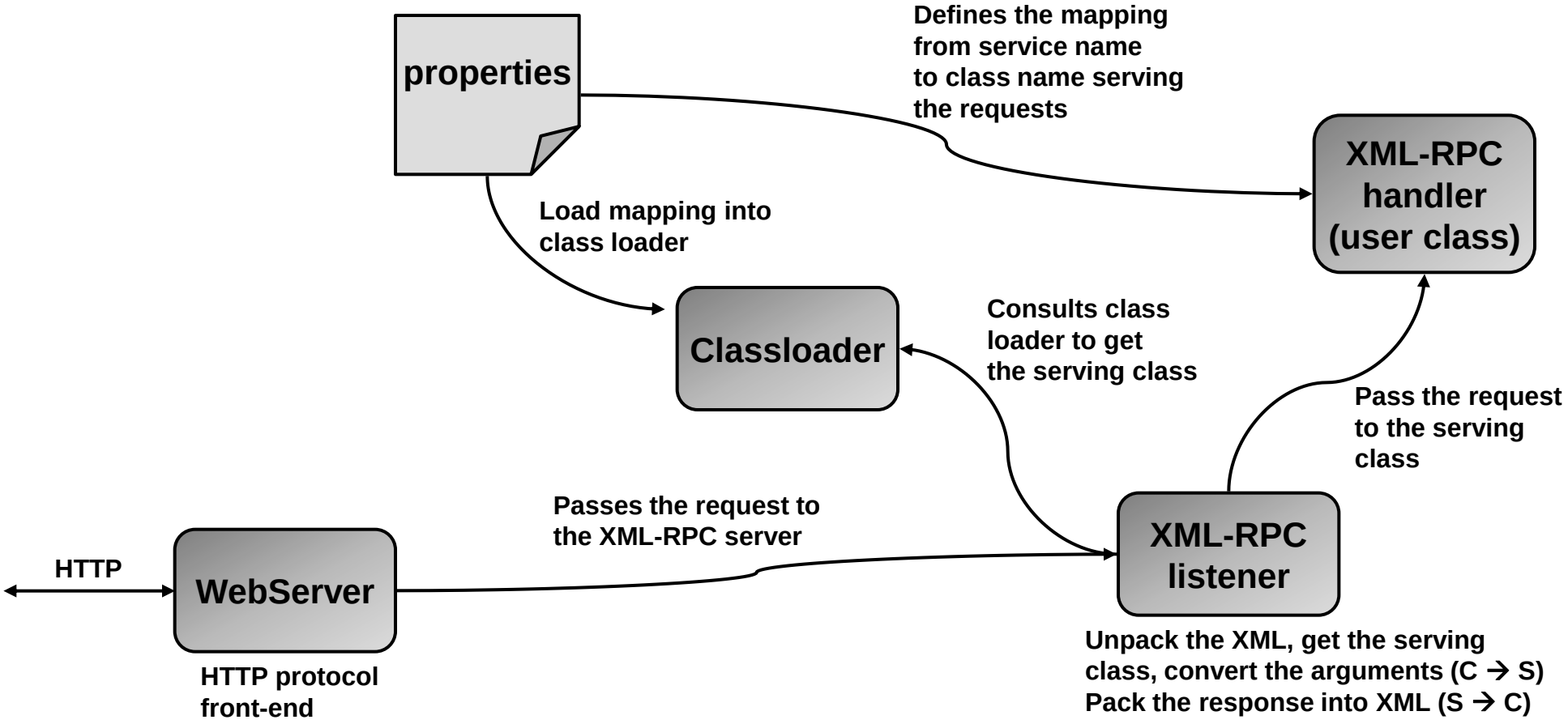
Arrays:

An array contains a single <data> element that in turn contains an array of <value> elements.
An array differs from a struct in that its elements are simple values (without <name> element).
Arrays may be recursive (value in array may be an array or also a struct).

```
<array>
  <data>
    <value><i4>12</i4></value>
    <value><string>Egypt</string></value>
    <value><boolean>0</boolean></value>
    <value><i4>-31</i4></value>
  </data>
</array>
```

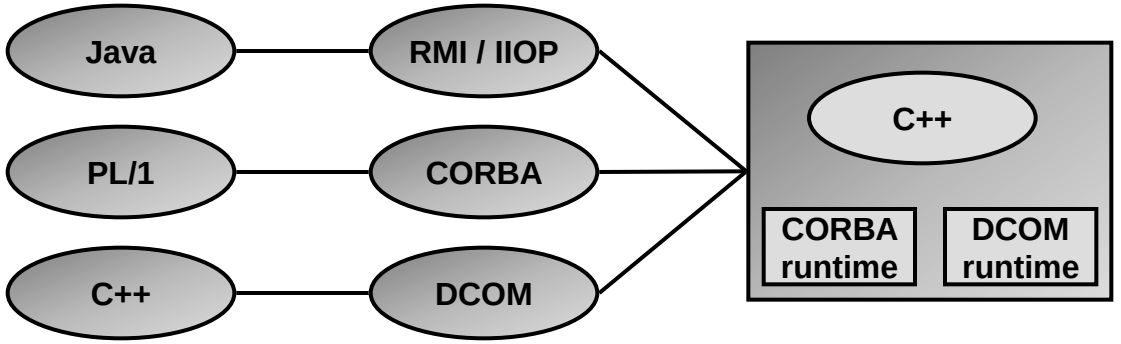
4. XML-RPC server implementation in Java

The class loader provides lookup service (find the serving class from request name).
The XML-RPC listener does unmarshalling / marshalling of parameters.
The XML-RPC handler is the user class that handles the request (implementation of the actual procedure call).

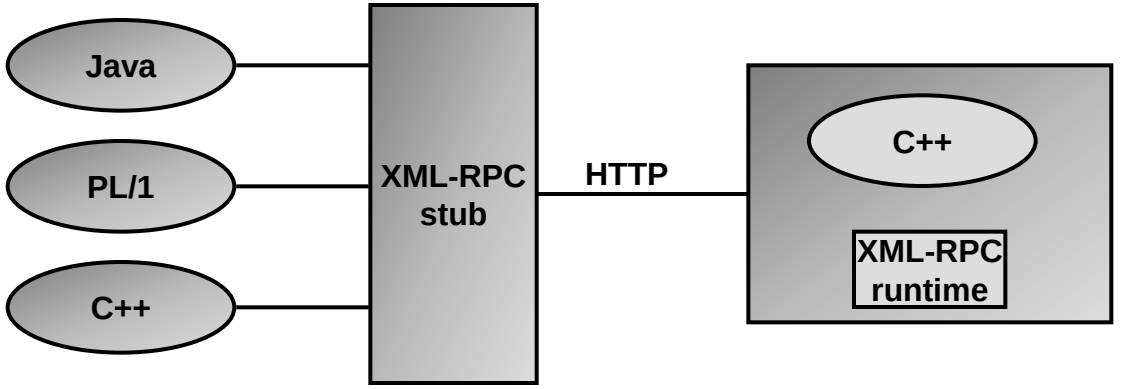


5. Where to use XML-RPC

- XML-RPC may be suited for simple applications or situations where clients implemented in different technologies need to interact with a server with simple read-write operations where a more complex middleware technology would be overkill.
- XML-RPC is a solution to integrate different platforms with a simple middleware.
- XML-RPC is very simple so it can be implemented also for platforms without open source or commercially available XML-RPC libraries.



Integration of different platforms leads to complexity on the server side



XML-RPC as a common integration technology reduces complexity on the server side (but adds complexity on the client side)