

SOAP

A Bubble Bath For RPC

Keith Maull

Outline

- Background & History
- Protocol Details
- What SOAP does well ...
- Criticisms of SOAP
- SOAP Alternatives
- The forecast for tomorrow is ...

A Little History

- Initial work began in 1998 ...
 - Dave Winer, Don Box, Bob Atkinson, Mohsen Al-Ghosein
 - Solving COM/DCOM issues for Internet operability
 - General interest in overcoming GIOP/IIOP
 - XML-RPC was the first incarnation (incidentally called SOAP according to Box)
 - SOAP evolved shortly after
 - Version 1.2 (2003)

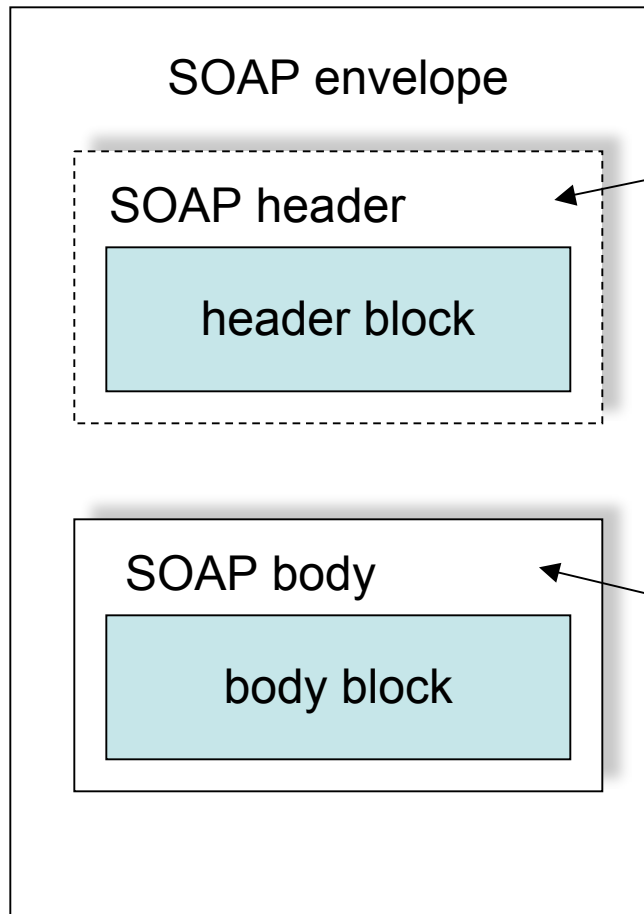
A Little Background

- EAI-to-Web Services problems
 - Standardization?
 - Syntax
 - Interaction mechanisms
 - Service descriptions
 - Naming and service lookup
- Solution :
 - XML + SOAP + WSLD + UDDI = Basic “Web Services”

SOAP

- “SOAP's original intent was fairly modest: to codify how to send *transient* XML documents to trigger operations or responses on remote hosts.” – *Don Box*
- RPC over HTTP via XML
 - Data encoded in XML for one-way communication over HTTP (but sometimes SMTP or TCP/IP)
 - Implements the RPC interaction pattern and defines how clients will talk to remote server
 - Defines processing rules for messages received by server and what to do after messages are received
 - Defines transport bindings for HTTP, SMTP

SOAP : Structure and Content



- Context of the message
- Transaction instructions, identification information, etc.
- **Examples:**
`role=next|none|ultimate`
`Receiver,`
`mustUnderstand=1|0`

- Core contents of the procedure call, including method name, parameters, types, etc.
- May be document-style or RPC-style content

Example Envelope (Beheaded)

```
<SOAP-ENV:Envelope xmlns:SOAP-
  ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:doGoogleSearch xmlns:ns1="urn:GoogleSearch"
      SOAP-
      ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <key xsi:type="xsd:string">000000000000000000000000000000000000</key>
      <q xsi:type="xsd:string">shrdlu winograd maclisp teletype</q>
      <start xsi:type="xsd:int">0</start>
      <maxResults xsi:type="xsd:int">10</maxResults>
      <filter xsi:type="xsd:boolean">true</filter>
      <restrict xsi:type="xsd:string"></restrict>
      <safeSearch xsi:type="xsd:boolean">>false</safeSearch>
      <lr xsi:type="xsd:string"></lr>
      <ie xsi:type="xsd:string">latin1</ie>
      <oe xsi:type="xsd:string">latin1</oe>
    </ns1:doGoogleSearch>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example Response

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:doGoogleSearchResponse xmlns:ns1="urn:GoogleSearch" SOAP-
      ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <return xsi:type="ns1:GoogleSearchResult">
        <documentFiltering xsi:type="xsd:boolean">>false</documentFiltering>
        <estimatedTotalResultsCount xsi:type="xsd:int">3</estimatedTotalResultsCount>
        <directoryCategories xmlns:ns2="http://schemas.xmlsoap.org/soap/encoding/"
          xsi:type="ns2:Array" ns2:arrayType="ns1:DirectoryCategory[0]"></directoryCategories>
        <searchTime xsi:type="xsd:double">0.194871</searchTime>
        <resultElements xmlns:ns3="http://schemas.xmlsoap.org/soap/encoding/"
          xsi:type="ns3:Array" ns3:arrayType="ns1:ResultElement[3]">
          <item xsi:type="ns1:ResultElement">
            <cachedSize xsi:type="xsd:string">12k</cachedSize>
            <hostName xsi:type="xsd:string"></hostName>
            <snippet xsi:type="xsd:string"> &lt;b&gt;...&lt;/b&gt; on a simple dialog
              (via &lt;b&gt;teletype&lt;/b&gt;) with ... vintage 1970, and to
              &lt;b&gt;...&lt;/b&gt;</snippet>
            <directoryCategory xsi:type="ns1:DirectoryCategory">
              <specialEncoding xsi:type="xsd:string"></specialEncoding>
              <fullViewableName xsi:type="xsd:string"></fullViewableName>
            </directoryCategory> ...
          </item>
        </resultElements>
      </return>
    </ns1:doGoogleSearchResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```


Transport Binding

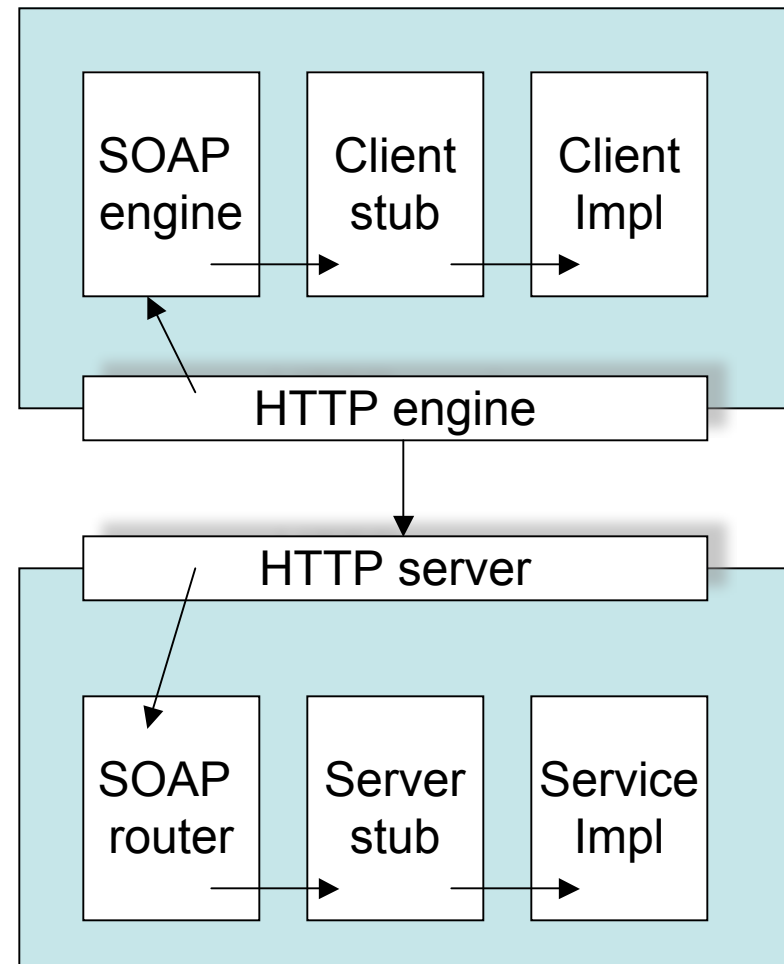
- Specify how underlying protocol transports SOAP message
- HTTP or SMTP
- Addressing
 - Specified by the target URL (HTTP)
 - By the To: recipient (SMTP)
- Routing
 - Path taken is same as underlying protocol

Asynchronous SOAP

- B2B bent
 - Processing of a message over time
 - Transactional messages that may require many intermediate processors
- Some implementations
 - SMTP
 - Threads and callbacks (AXIS2 non-blocking)
 - Message queue like MOM

SOAP Implementation

- SOAP client invokes local call, prepares message inside SOAP engine packages message into HTTP and sends to server
- SOAP server handles requests, router parses message and invokes the stub which invokes the implementation of the request



What's SOAP good for anyway?

- Complex data requiring standards
 - Translation = “the enterprise”
- Complex transactions
 - Translation = “how will I get paid?”
- Content-based routing
- The promise of WSDL
- ... and maybe take a bath with it

Criticism of SOAP

- RPC-style NOT loosely coupled
 - “Service interface” vs API?
 - What is the intent and expectation of each?
- Too complex for XML over HTTP
 - Added complexity, but the benefits?
- Commercial interests are at stake
 - Microsoft? IBM?

Alternatives

- POX
 - Plain old XML
- XML-RPC
 - SOAP on a diet (the spec is 2 pages)
 - Simple types, arrays
- JSON-RPC
 - Ajaxian
- REST
 - rq: `http://potofgold/money/showMeTheMoney`
 - rsp: `<justforyou>22,000,000</justforyou>`

Free Implementations

- Java/C : Apache AXIS (1 & 2)
- PHP-Soap
- COM/C++ : PocketSOAP
- Perl : SOAP::Lite