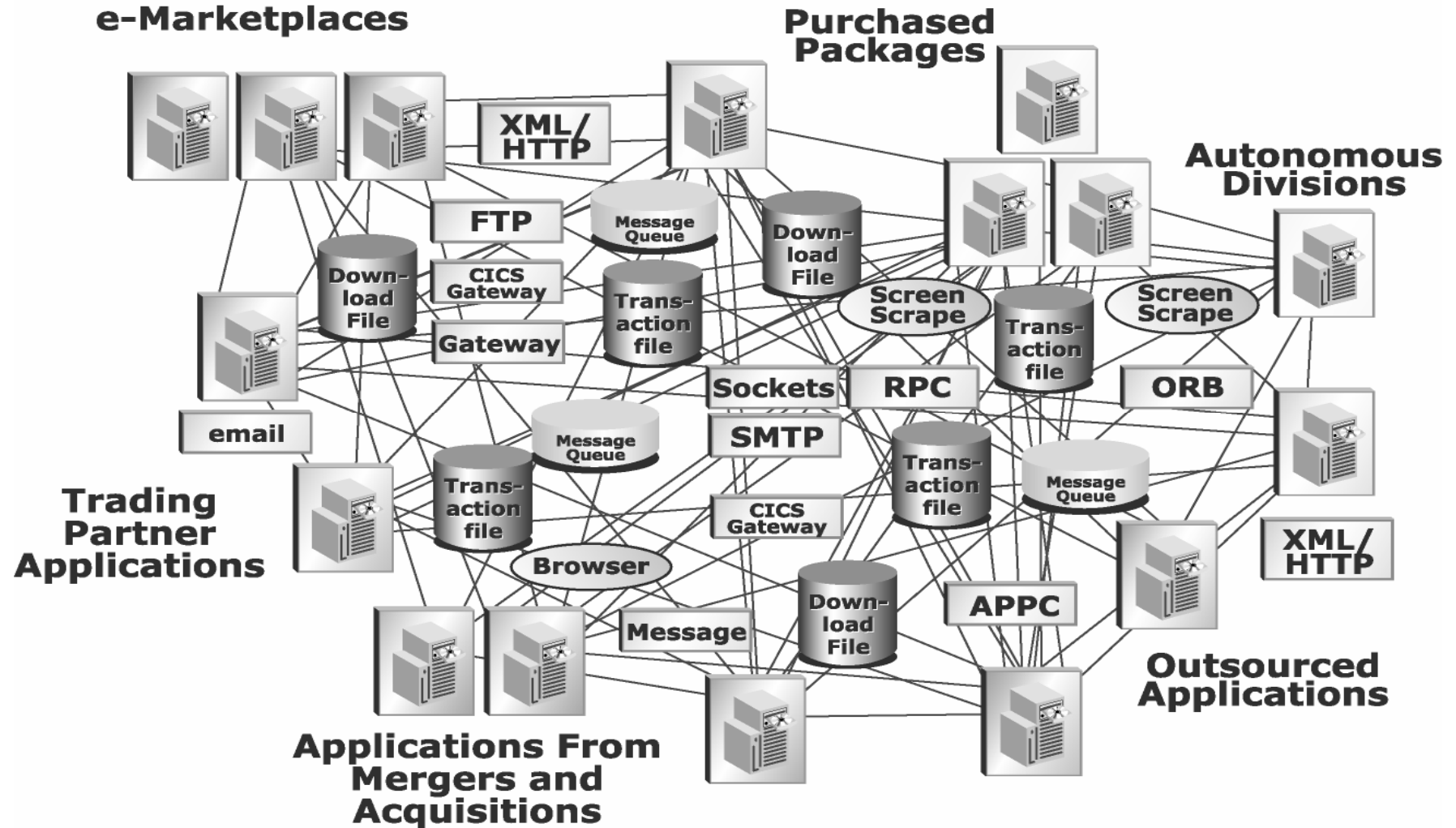


SOAP

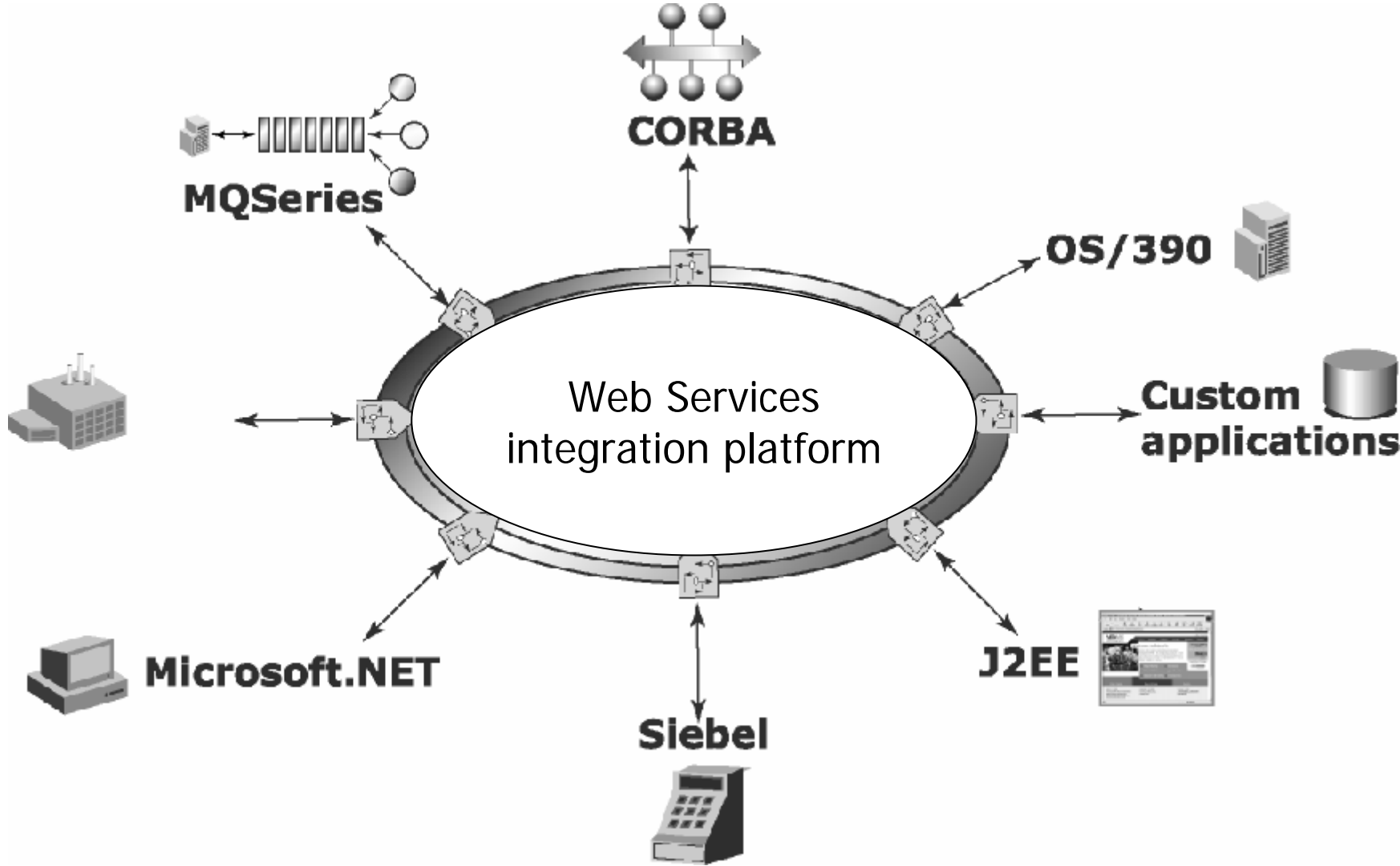
Why SOAP?

- Distributed computing is here to stay
- Computation through communication
- Resource heterogeneity
- Application integration
- Common language for data exchange

Why SOAP?



Why SOAP?



What is SOAP?

- SOAP is an XML-based protocol for exchanging information between computers
- Primary focus of SOAP is Remote Procedure Calls transported via HTTP
- Similar to DCOM, CORBA, and Java RMI; the main difference is that SOAP messages are written entirely in XML
- SOAP is therefore uniquely platform and language independent
- For example, a SOAP Java client running on Linux or a Perl client running on Solaris can connect to a Microsoft SOAP server running on Windows 2000

SOAP facts

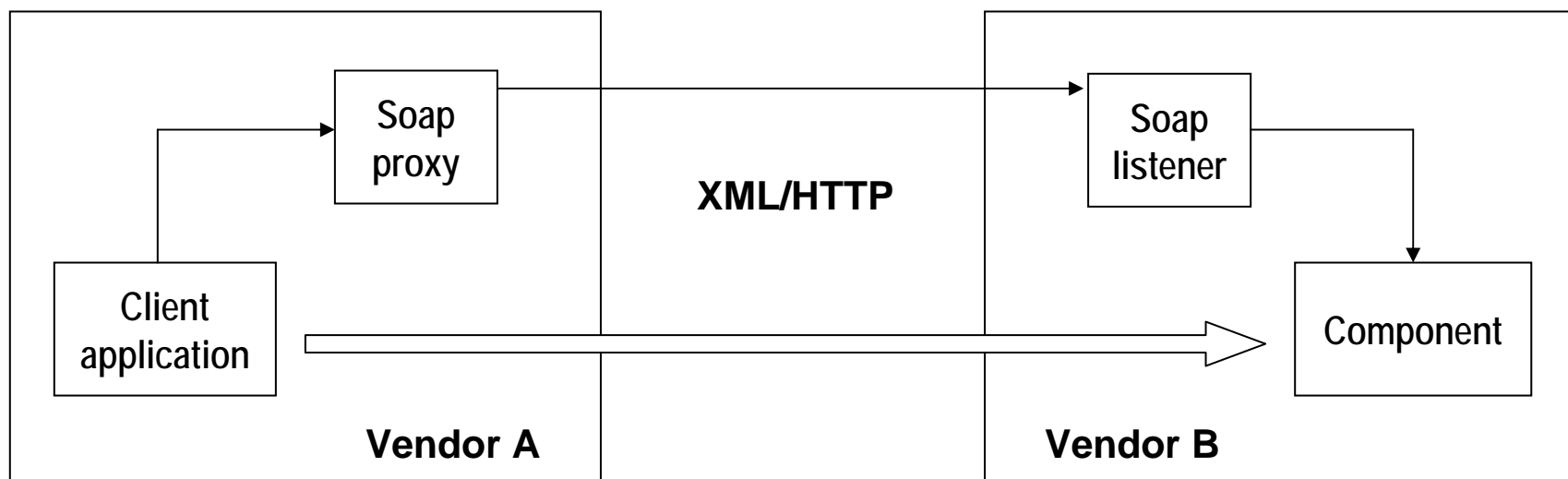
- Although still in its infancy, SOAP has received widespread industry support
- Dozens of SOAP implementations now exist, including implementations for Java, COM, Perl, C# and Python
- SOAP has received an official recommendation by the W3C

What SOAP is not

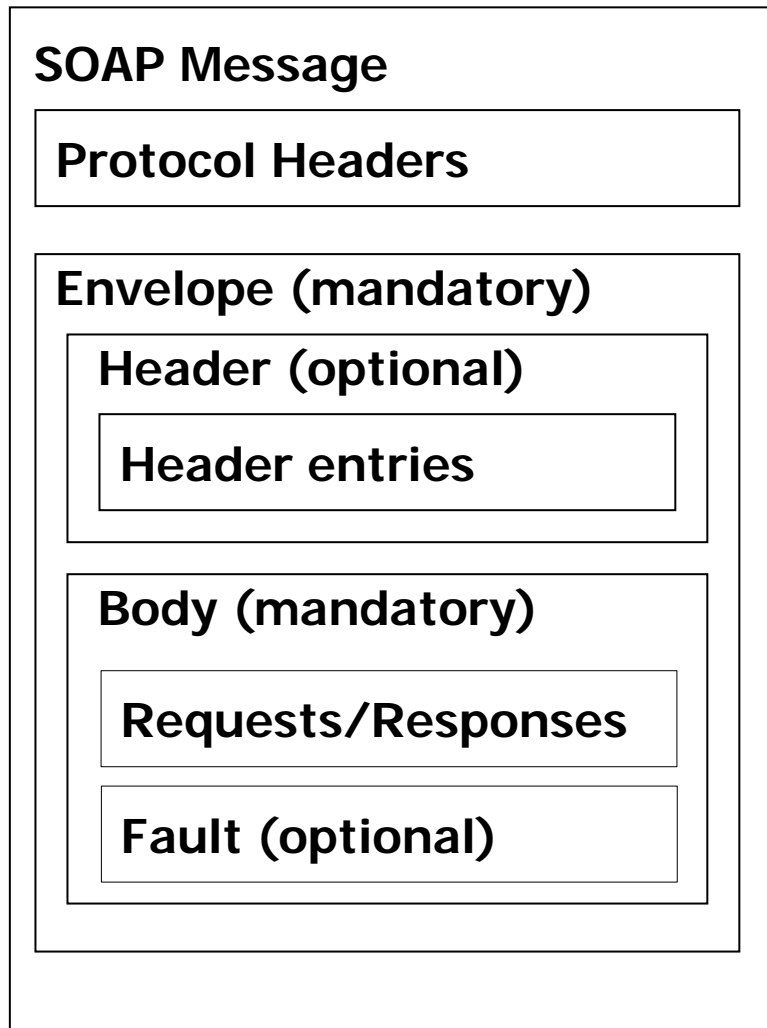
- Object activation
- Bi-directional communications
- Distributed garbage collection
- Language bindings
 - Good for interoperability
 - Source of payload is immaterial

SOAP Messaging

- A SOAP conversation includes:
 - SOAP Request
 - specify method name, method parameters, etc.
 - SOAP Response
 - specify return values or error conditions
- All SOAP messages are encoded in XML



SOAP message structure



Versioning is based
on XML namespaces

Sample SOAP Request

```
<?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:getTemp xmlns:ns1="urn:xmethods-Temperature"
      SOAP-ENV:encodingStyle=
        "http://schemas.xmlsoap.org/soap/encoding/" >
      <zipcode xsi:type="xsd:string" >10016</zipcode>
    </ns1:getTemp>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

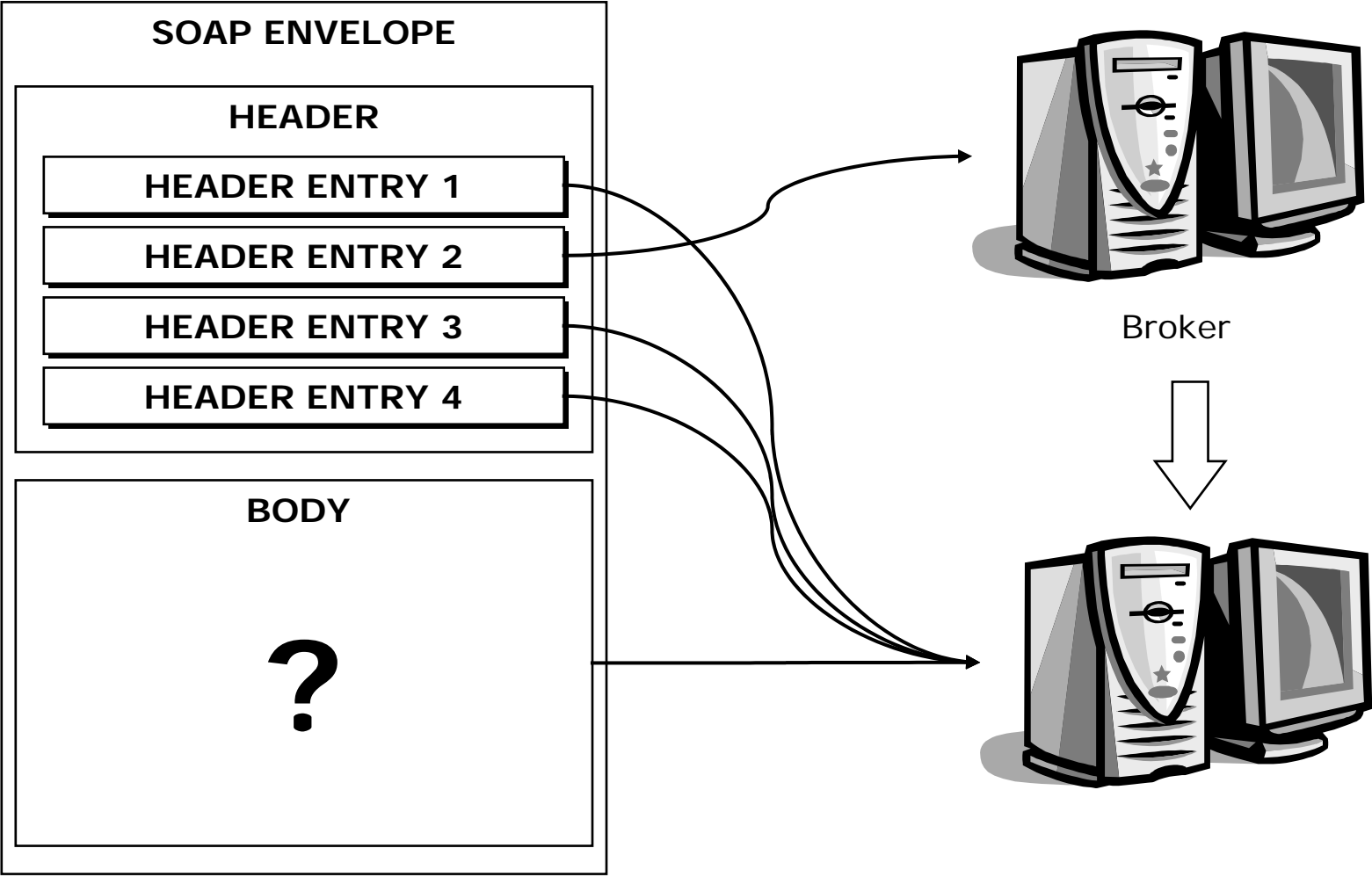
Sample SOAP Response

```
<?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:getTempResponse xmlns:ns1="urn:xmethods-Temperature"
      SOAP-ENV:encodingStyle=
        "http://schemas.xmlsoap.org/soap/encoding/" >
      <return xsi:type="xsd:float">71.0</return>
    </ns1:getTempResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

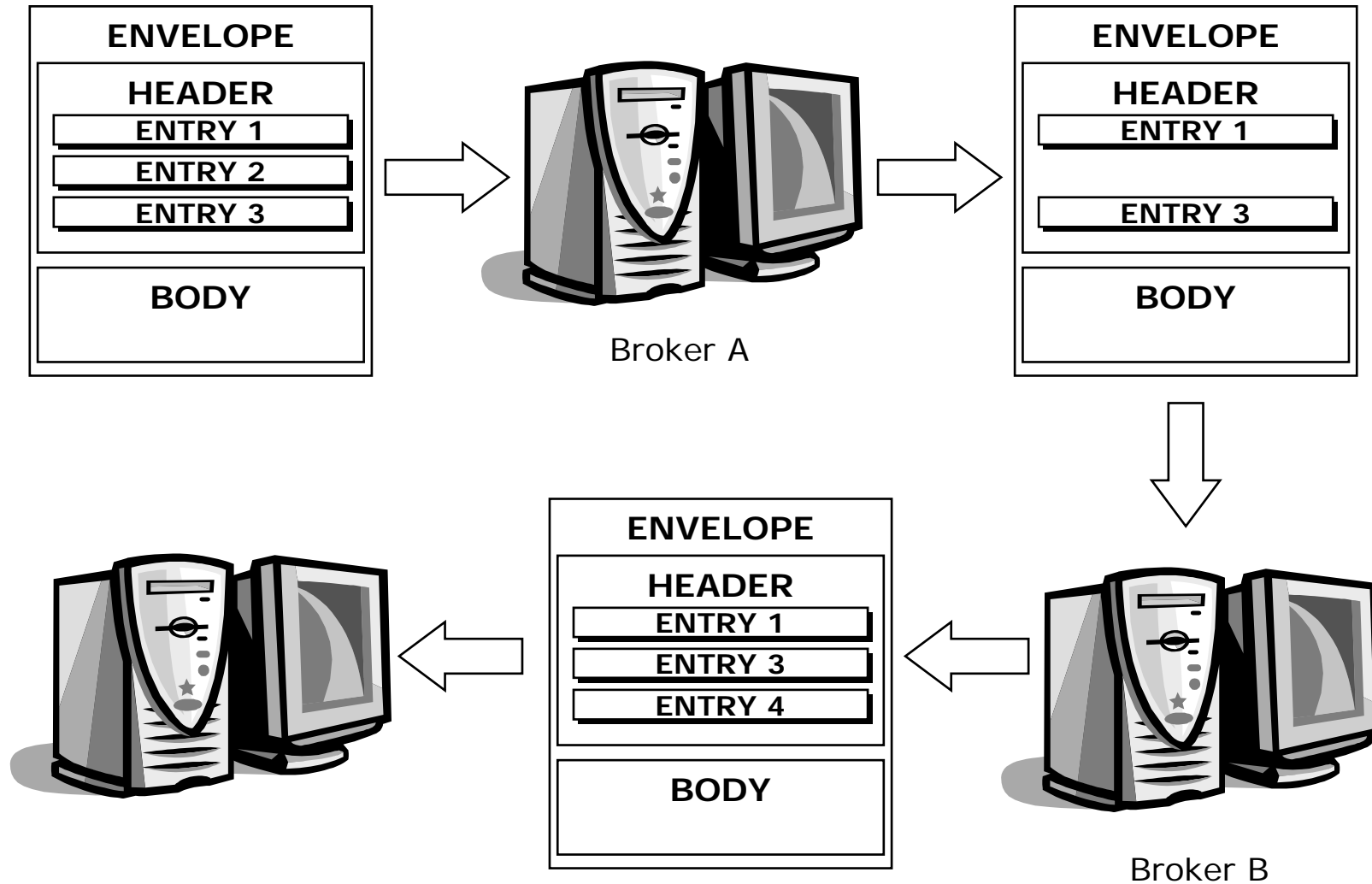
Header

- Flexible framework for additional application-level requirements
- E.g., authentication, transaction management, payment authorization
- Currently not much used
- Entries are open-ended: only two attributes defined
 - actor: recipient of the entry
 - mustUnderstand: if set to true, the recipient must process the header

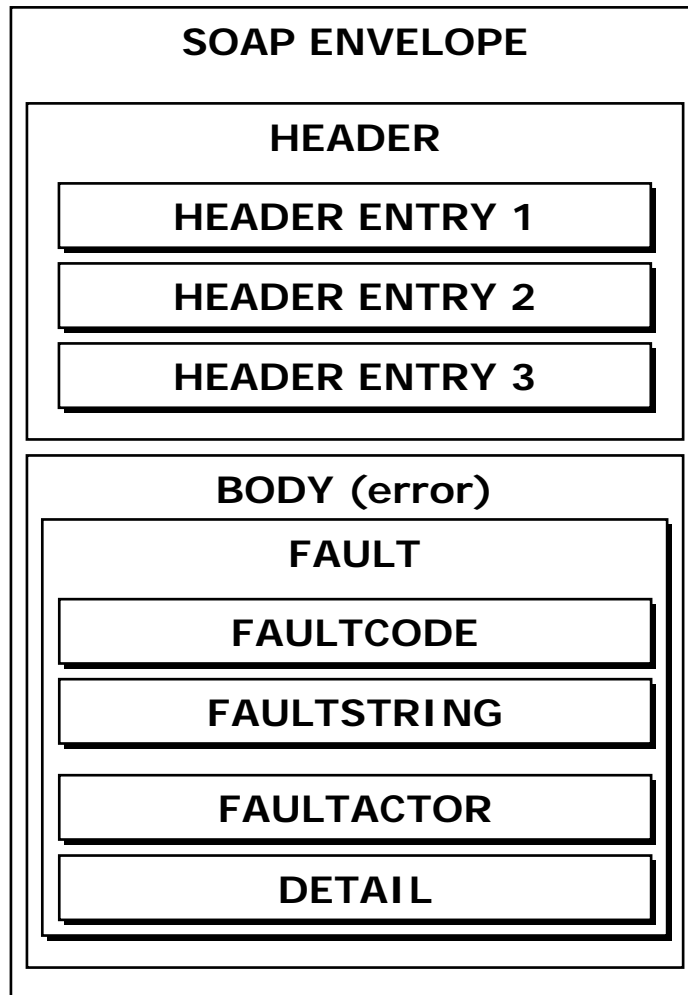
Header



Actors



Faults



Fault sub-elements

Sub-element	Description
<faultcode>	A code for identifying the fault
<faultstring>	A human readable explanation of the fault
<faultactor>	Information about who caused the fault to happen
<detail>	Holds application specific error information in <i>detail entries</i>

Fault codes

Code	Description
VersionMismatch	Found an invalid namespace for the Envelope element
MustUnderstand	An immediate child element of the Header element, with the mustUnderstand attribute set to "true", was not understood
Client	The message was incorrectly formed or contained incorrect information
Server	There was a problem with the server so the message could not proceed

SOAP encoding

- Scalar types
 - Encoded using XML Schema *simple types*
- Compound types
 - Arrays: entries are identified by position
 - Structs: entries are identified by name

Arrays

- Arrays require size and element type to be specified
- Multidimensional arrays not supported by all implementations

```
<return xmlns:ns2="http://schemas.xmlsoap.org/soap/encoding"  
        xsi:type="ns2:Array" ns2:ArrayType="xsd:double[2]" >  
  <item xsi:type="xsd:double">54.99</item>  
  <item xsi:type="xsd:double">...</item>  
</return>
```

Struts

- Each element is specified with a unique “accessor” element

```
<return xmlns:ns3="urn:examples" xsi:type="ns3:product" >  
    <name xsi:type="xsd:string" >...</name>  
    <price xsi:type="xsd:double" >...</price>  
</return>
```

Literal encoding

- The SOAP encoding style is not mandatory
- An XML fragment can be embedded
- *In the following fragment we use the Apache SOAP namespace for literal XML*

```
<ns4:getProductResponse
  xmlns:ns4="urn:examples:ProductService"
  SOAP-ENV:encodingStyle=
    "http://xml.apache.org/xml-soap/literalxml" >
  <return>
    <product code="..." >
      <name>...</name>
      <price>...</price>
    </product>
  </return>
</ns4:getProductResponse>
```

SOAP via HTTP

- HTTP is the most popular SOAP transport protocol
- Requests and responses must have text/xml as content type
- Client must specify a SOAPAction header
 - Server-specific URI to indicate the intent of the request
 - Frequently used by firewall and dispatchers
- HTTP status codes
- SOAP 1.2 makes SOAPAction optional

SOAP implementations

- *Apache SOAP*
 - Open source, based on IBM SOAP4J
- *Microsoft SOAP toolkit*
 - COM implementation, for COM-compliant languages
- *SOAP::Lite for Perl*
- *GLUE (Mind Electric)*
 - Java implementation