Web Services

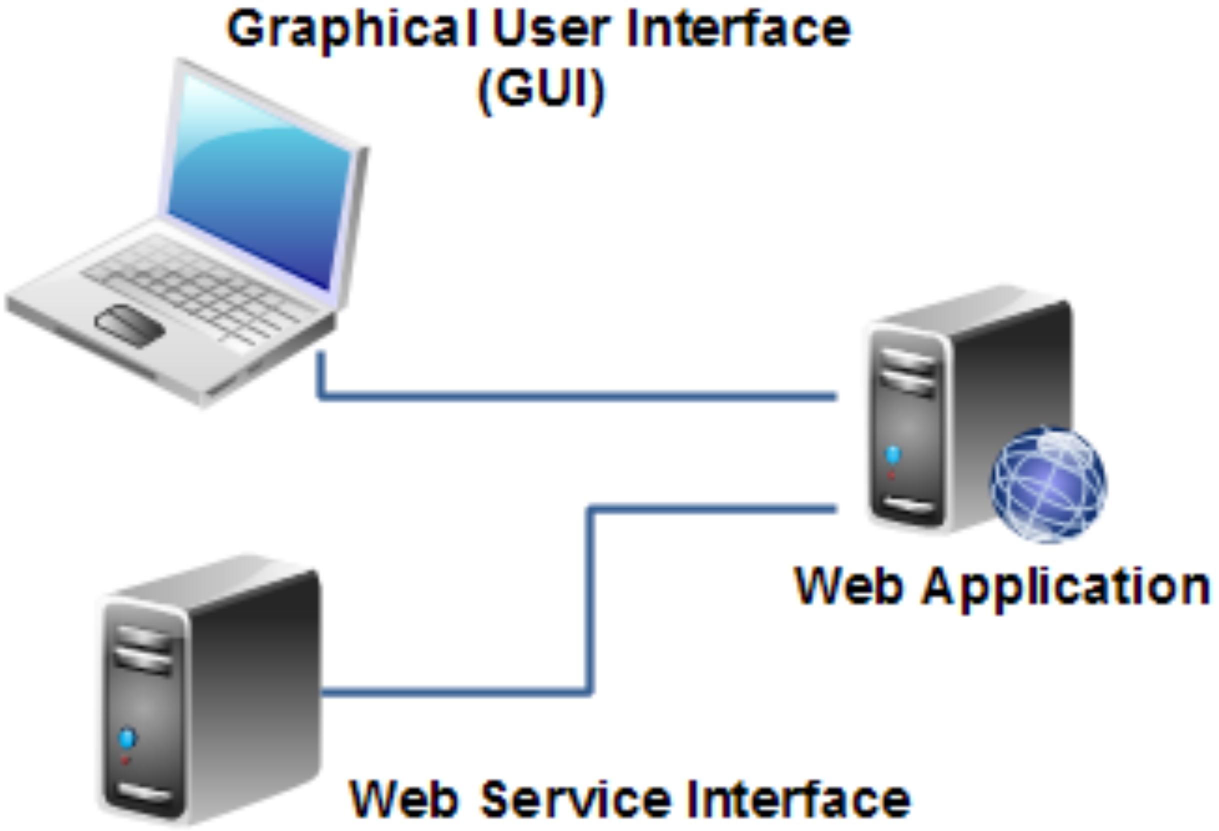
What are web services?

- Programmatic interfaces used for application to application communication.
- Provider exposes service that requestor consumes.
- Implementation details hidden from requestor.

http://www.w3.org/2002/ws/

http://www.w3.org/TR/ws-arch/

What are web services?



The Basics

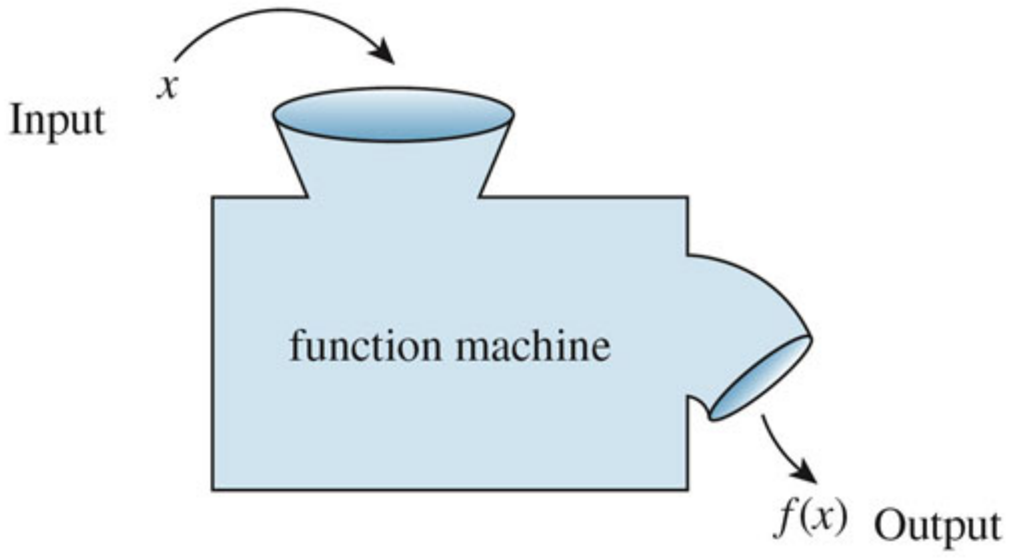
Function Machines and Functions Defined by Formulas.

a. Function Machines.

One way to think of a function is as a machine. You drop a domain element into the input hopper, and it produces a codomain element from the output chute.

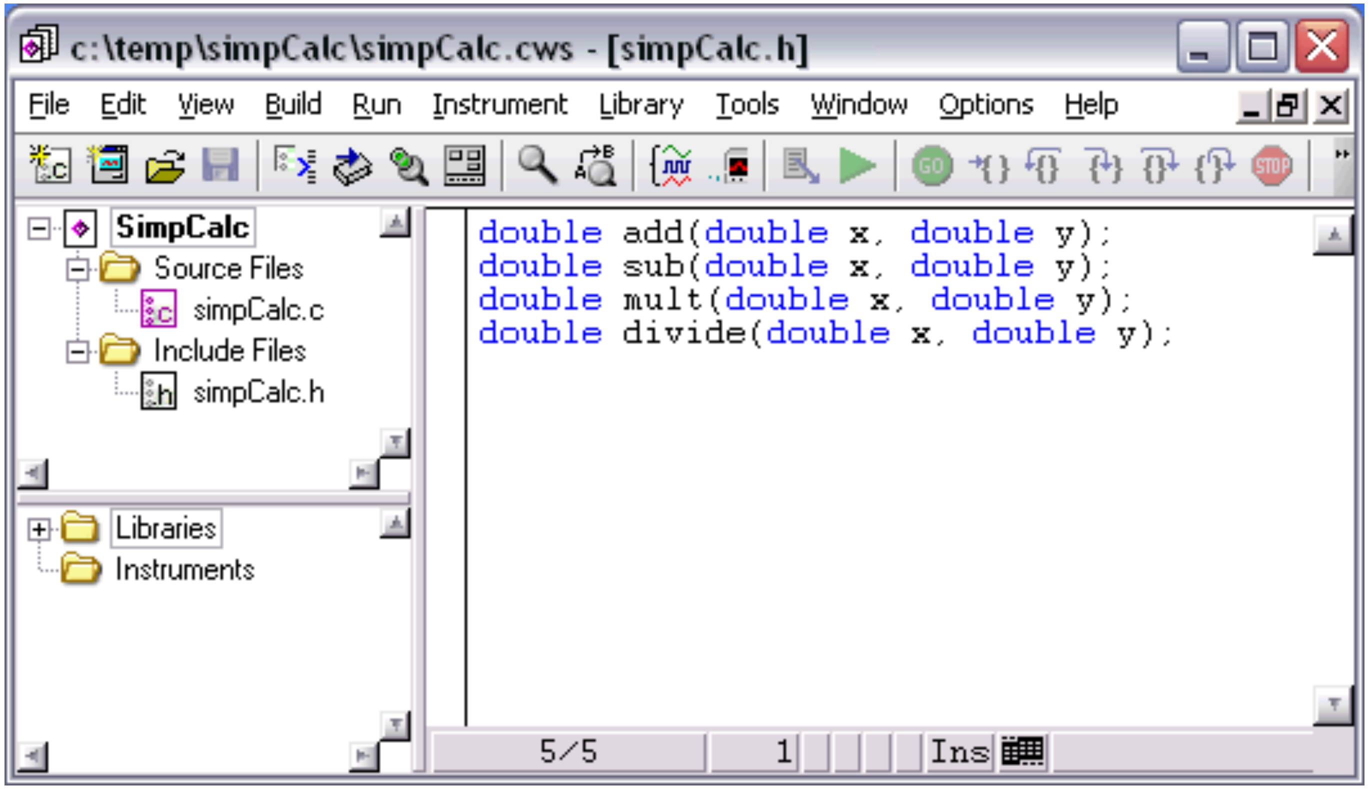
There is a rule or formula hiding inside the machine.

We do have to specify what the domain is for the rule, so we don't drop things into the machine that might "break" it. (It may not know how to handle certain inputs.)



Function prototypes & header files

A function definition specifies what a function does, a function prototype can be thought of as specifying its *interface*.


c:\temp\simpCalc\simpCalc.cws - [simpCalc.h]
File Edit View Build Run Instrument Library Tools Window Options Help
SimpCalc
Source Files
simpCalc.c
Include Files
simpCalc.h
Libraries
Instruments
5/5
1
Ins

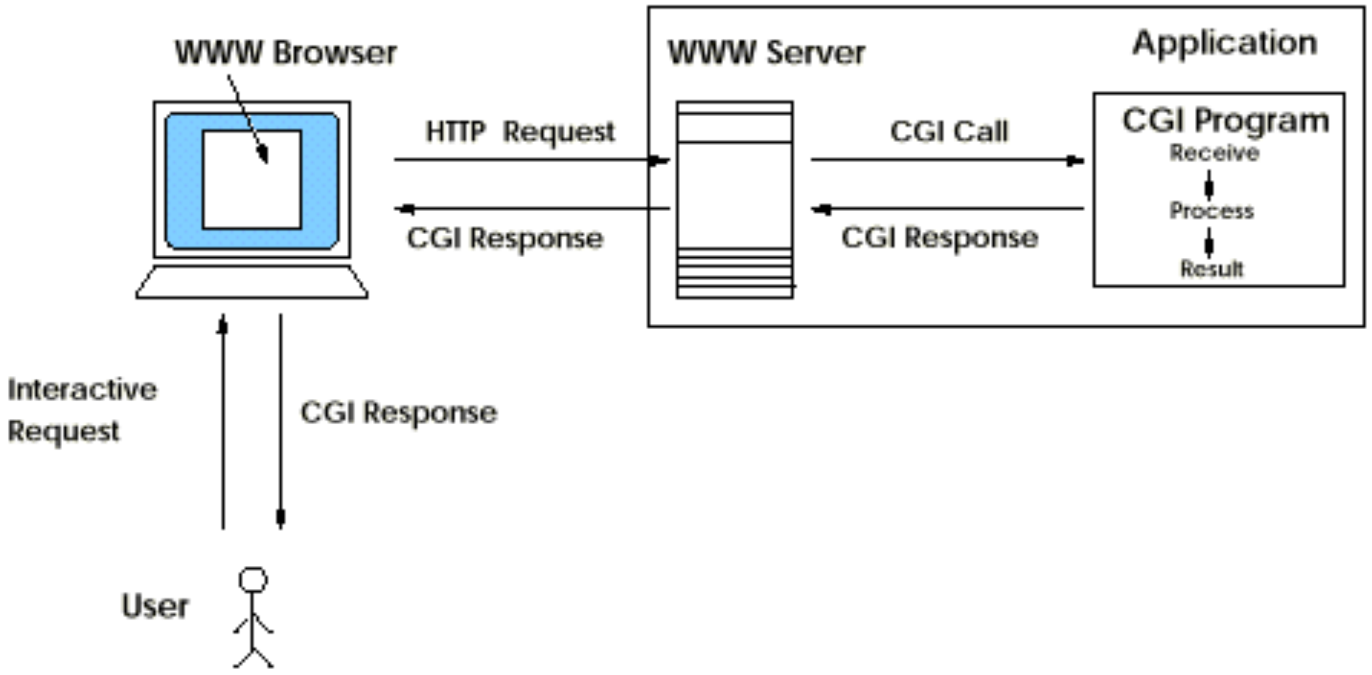
```
double add(double x, double y);
double sub(double x, double y);
double mult(double x, double y);
double divide(double x, double y);
```

Types of web services

- Old School text-based
  - HTTP + HTML + CGI
- Modern text-based
  - REpresentational State Transfer (REST)
    Typically XML or JSON
    Uses variety of HTTP request verbs
    http://en.wikipedia.org/wiki/REST
  - SOAP-based (XML)
    (Originally Simple Object Access Protocol)
    Uses Web Services Description Language (WSDL)
    http://en.wikipedia.org/wiki/SOAP
    http://en.wikipedia.org/wiki/Web_Services_Description_Language
- Alternatives and hybrids
  - Remote Procedure Calls (RPC)
  - Service Component Architecture (SCA)
  - Service Data Objects (SDO)
  - Windows Communication Foundation (WCF)
  - CORBA, GIOP, ICE, DCOM (older binary formats)

Common Gateway Interface (CGI) over HTTP/S

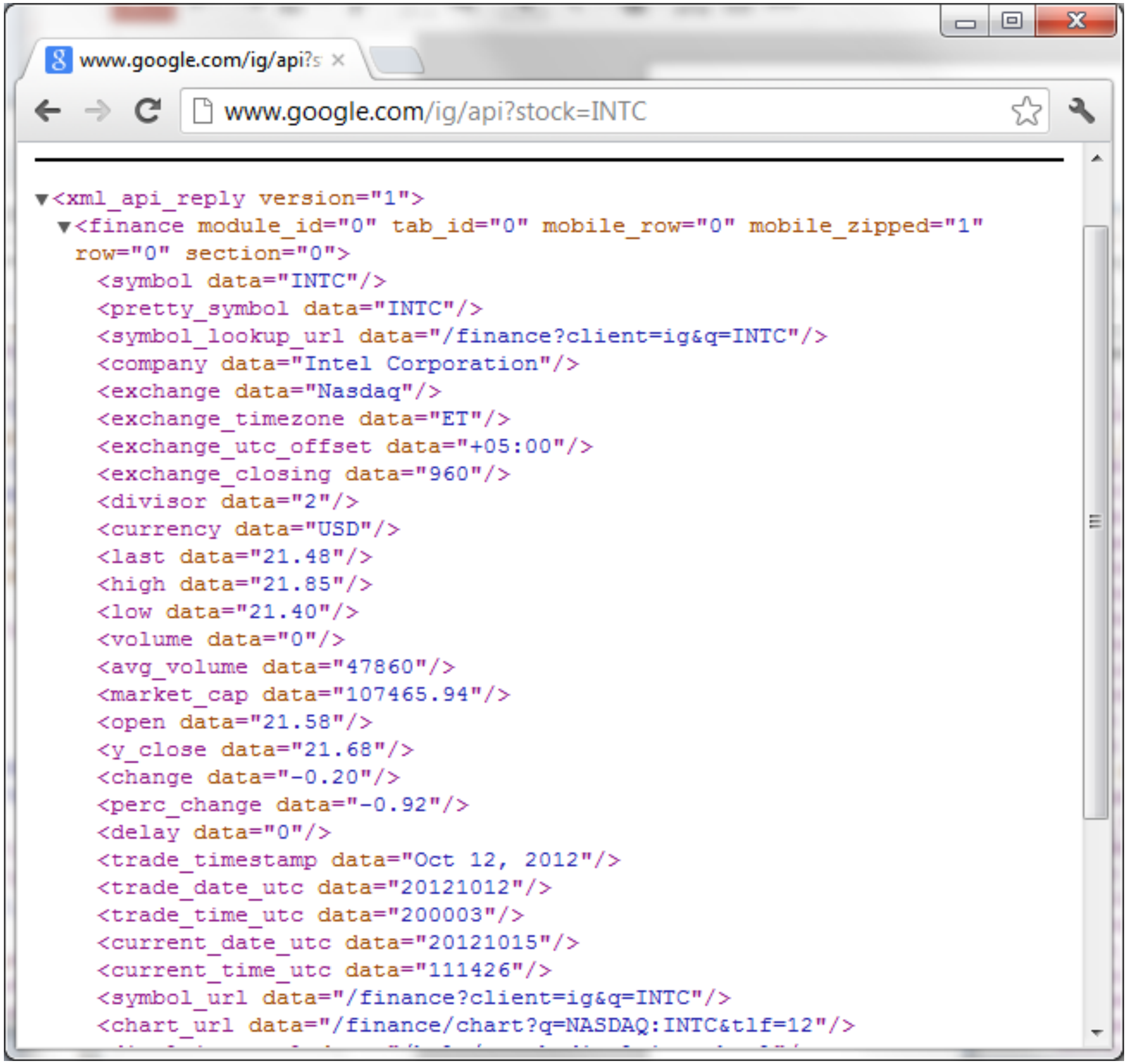
The CGI program usually did all of the UI, Business, and Data processing (Single Tier).



Data formats

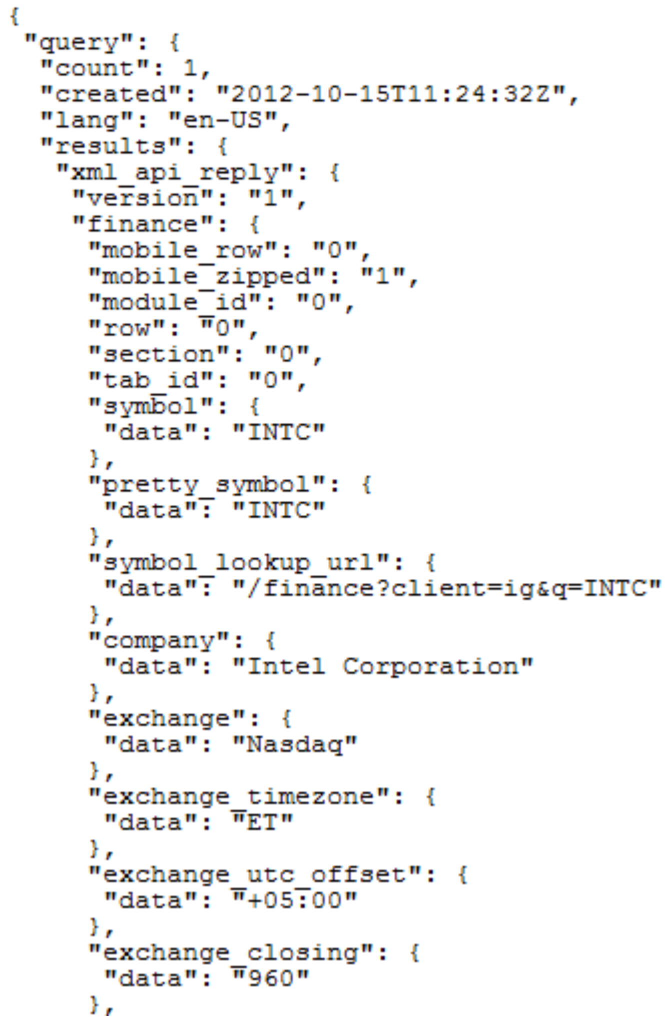
- Binary
  - Images, Audio, Video
  - Documents (RTF, DOC, PDF)
- Text-based
  - HTML
  - Plain text
  - CSV
  - XML (RSS, Atom)
  - JSON

XML



```
▼<xml_api_reply version="1">
  ▼<finance module_id="0" tab_id="0" mobile_row="0" mobile_zipped="1"
   row="0" section="0">
     <symbol data="INTC"/>
     <pretty_symbol data="INTC"/>
     <symbol_lookup_url data="/finance?client=ig&q=INTC"/>
     <company data="Intel Corporation"/>
     <exchange data="Nasdaq"/>
     <exchange_timezone data="ET"/>
     <exchange_utc_offset data="+05:00"/>
     <exchange_closing data="960"/>
     <divisor data="2"/>
     <currency data="USD"/>
     <last data="21.48"/>
     <high data="21.85"/>
     <low data="21.40"/>
     <volume data="0"/>
     <avg_volume data="47860"/>
     <market_cap data="107465.94"/>
     <open data="21.58"/>
     <y_close data="21.68"/>
     <change data="-0.20"/>
     <perc_change data="-0.92"/>
     <delay data="0"/>
     <trade_timestamp data="Oct 12, 2012"/>
     <trade_date_utc data="20121012"/>
     <trade_time_utc data="200003"/>
     <current_date_utc data="20121015"/>
     <current_time_utc data="111426"/>
     <symbol_url data="/finance?client=ig&q=INTC"/>
     <chart_url data="/finance/chart?q=NASDAQ:INTC&tlf=12"/>
```

JSON



```
{
 "query": {
  "count": 1,
  "created": "2012-10-15T11:24:32Z",
  "lang": "en-US",
  "results": {
   "xml_api_reply": {
    "version": "1",
    "finance": {
     "mobile_row": "0",
     "mobile_zipped": "1",
     "module_id": "0",
     "row": "0",
     "section": "0",
     "tab_id": "0",
     "symbol": {
      "data": "INTC"
     },
     "pretty_symbol": {
      "data": "INTC"
     },
     "symbol_lookup_url": {
      "data": "/finance?client=ig&q=INTC"
     },
     "company": {
      "data": "Intel Corporation"
     },
     "exchange": {
      "data": "Nasdaq"
     },
     "exchange_timezone": {
      "data": "ET"
     },
     "exchange_utc_offset": {
      "data": "+05:00"
     },
     "exchange_closing": {
      "data": "960"
     },
```

RESTful web services

- HTTP interactions are stateless
- HTTP has OPTIONS, GET, HEAD, POST, PUT, DELETE, and TRACE methods
- HTTP uses a MIME-like envelope format to encode representations

```
# Request
POST /quotegen HTTP/1.1
Host: www.example.org
Content-Type: application/x-www-form-urlencoded

fname=...&lname=...&..
```

```
# Response
HTTP/1.1 200 OK
Content-Type: application/xml;charset=UTF-8

<quote xmlns:atom="http://www.w3.org/2005/Atom">
  <driver>
    ...
  </driver>
  <vehicle>
    ...
  </vehicle>
  <offer>
    ...
    <valid-until>2009-10-02</valid-until>
    <atom:link href="http://www.example.org/quotes/buy?quote=abc1234"
          rel="http://www.example.org/rels/quotes/buy"/>
  </offer>
</html>
```

SOAP

Web Services Description Language

```
<?xml version="1.0" encoding="l
<definitions name="AktienKurs":
  targetNamespace="http://loca
  xmlns:xsd="http://schemas.xmlsoap.or
  xmlns="http://schemas.xmlsoap.org/wsd
  <service name="AktienKurs">
    <port name="AktienSoapPort" binding
      <soap:address location="http://loc
    </port>
    <message name="Aktie.HoleWert">
      <part name="body" element="xsd:Tra
    </message>
    …
  </service>
</definitions>
```

Filename extension	.wsdl
Internet media type	application/wsdl+xml
Developed by	World Wide Web Consortium
Contained by	XML
Standard(s)	2.0 Recommendation

A SOAP request:

```
POST /InStock HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

<soap:Body xmlns:m="http://www.example.org/stock">
  <m:GetStockPrice>
    <m:StockName>IBM</m:StockName>
  </m:GetStockPrice>
</soap:Body>

</soap:Envelope>
```

The SOAP response:

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

<soap:Body xmlns:m="http://www.example.org/stock">
  <m:GetStockPriceResponse>
    <m:Price>34.5</m:Price>
  </m:GetStockPriceResponse>
</soap:Body>

</soap:Envelope>
```

Code Examples

- http://www.wrox.com/WileyCDA/WroxTitle/Expert-PHP-and-MySQL.productCd-0470563125,descCd-DOWNLOAD.html
- http://www.google.com/search?q=php+web+services
- http://developer.yahoo.com/yql/console/?q=show%20tables&env=store://datatables.org/alltableswithkeys&debug=true#h=select%20*%20from%20google.igoogle.stock%20where%20stock%3D%27intc%27%3B

Group Activity

Which web services might your use cases consume? For each use case:

- Which teams will need to work together to agree on an interface?
- Suggest a URI (or WS technology)
- What are the required parameters?
- What is the expected output?

Data validation and verification

http://en.wikipedia.org/wiki/Data_validation