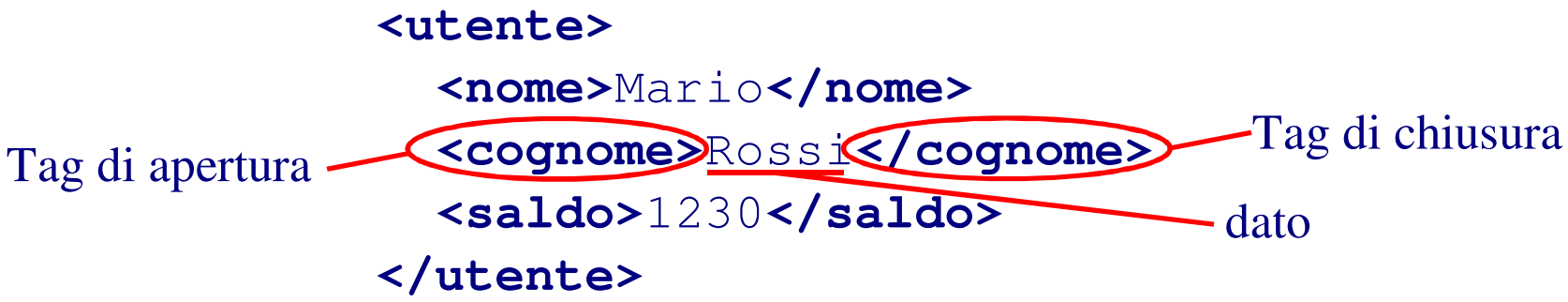
Interoperabilità e linguaggio XML

- Nel laboratorio precedente abbiamo visto come tramite BPMN sia possibile istruire un sistema informatico a gestire i flussi di attività.
- Si tratta però di attività interne ad un Pool, ossia ad una specifica azienda o ente. Come possono più aziende (Pool) comunicare scambiandosi messaggi? Tramite il formato **XML** e i **Servizi Web**.
- XML (sigla di *eXtensible* **Markup** Language) è un linguaggio **a marcatori** che consente di definire dei **tag** (o **elementi**) sui dati, per consentire a tali dati di essere correttamente interpretati.
- Si tratta di un linguaggio *estensibile*, ossia è possibile aggiungere altri tag per estendere i dati ricevuti con altre informazioni.
- Es. supponiamo di dover inviare un messaggio con i seguenti dati: "L'utente di nome Mario Rossi ha un saldo di € 1230". Una possibile traduzione XML è la seguente:



- Si noti che il dato (es. Rossi) è racchiuso tra il tag di apertura (es. **<cognome>**) e quello di chiusura (**</cognome>**). Il dato composto da più tag viene comunque racchiuso a sua volta in un tag (es. **<utente>**…**</utente>**).

- Se tutti i saldi scambiati tra le aziende sono espressi in Euro non occorre specificare la valuta, ma se operiamo in un contesto globale, con più possibili valute, si può aggiungere una **opzione** (o **attributo**) all'interno del tag che esprime la valuta:

```
<saldo valuta="Euro">1230</saldo>
```

- Quando manca il dato, allora il tag di apertura e chiusura diventano un tag unico, si ha il **tag vuoto**. Quindi al posto della seguente forma:

```
<saldo valuta="Euro"></saldo>
```

- Occorre adoperare la seguente forma:

```
<saldo valuta="Euro"/>
```

- Es. "Una Cantina contiene vini di categoria Rosso. Ciascun Vino è caratterizzato da una denominazione e una data di produzione, da un Produttore e un Prezzo. Un Produttore è caratterizzato da nome e cognome, mentre il Prezzo è caratterizzato dalla valuta e dall'importo".

- Versione XML:

```
<Cantina>
  <categoria>Rosso</categoria>
  <Vino>
    <denominazione>Chianti Classico</denominazione>
    <data giorno = "23" mese = "11" anno = "1998"/>
    <Produttore formato = "nome cognome">Giorgio Rossi</Produttore>
    <Prezzo importo = "14.50 Euro"/>
  </Vino>
  <Vino>
    <denominazione>Lambrusco Frizzante</denominazione>
    <data giorno = "15" mese = "04" anno = "2000"/>
    <Produttore formato = "nome cognome">Gianni Frizzi</Produttore>
    <Prezzo importo = "20.11 Euro"/>
  </Vino>
</Cantina>
```

- Questa versione XML non è ottimale per la macchina, e rende l'accesso inefficience. Ecco delle **regole per un formato XML efficiente**. Nella scelta di rappresentare un dato come tag o come opzione, considerarlo:

a) un tag, se il dato è strutturato su linee multiple, o cambia spesso.

b) una opzione se il dato è semplice e non cambia spesso, oppure appartiene a un set limitato di possibilità predefinite.

- Ecco come si può rappresentare meglio la versione XML precedente:

```
<Cantina categoria = "Rosso">
  <Vino>
    <denominazione>Chianti Classico</denominazione>
    <data>1998-11-23</data>
    <Produttore>
     <nome>Giorgio</nome>
     <cognome>Rossi</cognome>
    </Produttore>
    <Prezzo valuta = "Euro" >14.50</Prezzo>
  </Vino>
  <Vino>
    <denominazione>Lambrusco Frizzante</denominazione>
    <data>2000-04-15</data>
    <Produttore>
     <nome>Gianni</nome>
     <cognome>Frizzi</cognome>
    </Produttore>
    <Prezzo valuta = "Euro" >20.11</Prezzo>
  </Vino>
</Cantina>
```

- Infatti: la categoria di vino e la valuta sono dati che non cambiano spesso e appartengono ad un set limitato di possibilità predefinite (es. vino rosso, bianco, rosato, valuta Euro, Dollaro, …). Per cui sono delle opzioni.

- Tutto il resto è fatto di dati complessi, o che cambiano spesso, o che appartengono a set infiniti di possibilità.

SOA client for Firefox

- Grazie ad XML è possibile per due macchine scambiarsi dei messaggi ed eseguire dei servizi web. Esiste uno standard detto SOA (Service Oriented Architecture) che permette alle macchine si eseguire detti servizi.

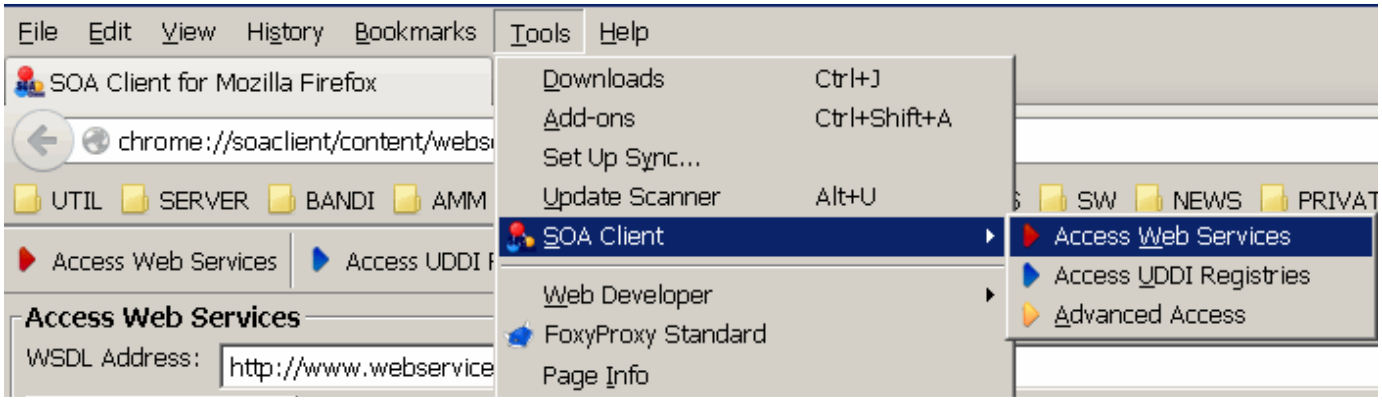
- Allo stesso modo in cui a un sito web è possibile chiedere una pagina HTML, è possibile chiedere dati e servizi a un **servizio web**, inviando un messaggio XML con le richieste e ottenendo il risultato.

- Un servizio web non si accede navigando con il browser. Ad esempio all'indirizzo: http://www.webservicex.net/geoipservice.asmx?WSDL
  è presente un servizio web. Ma se si digita tale indirizzo sulla barra degli indirizzi di un browser, apparirà una sequenza di dati in XML pressochè incomprensibile, perché un servizio web non è fatto per essere direttamente acceduto da un essere umano, ma da una macchina, ossia da un software.

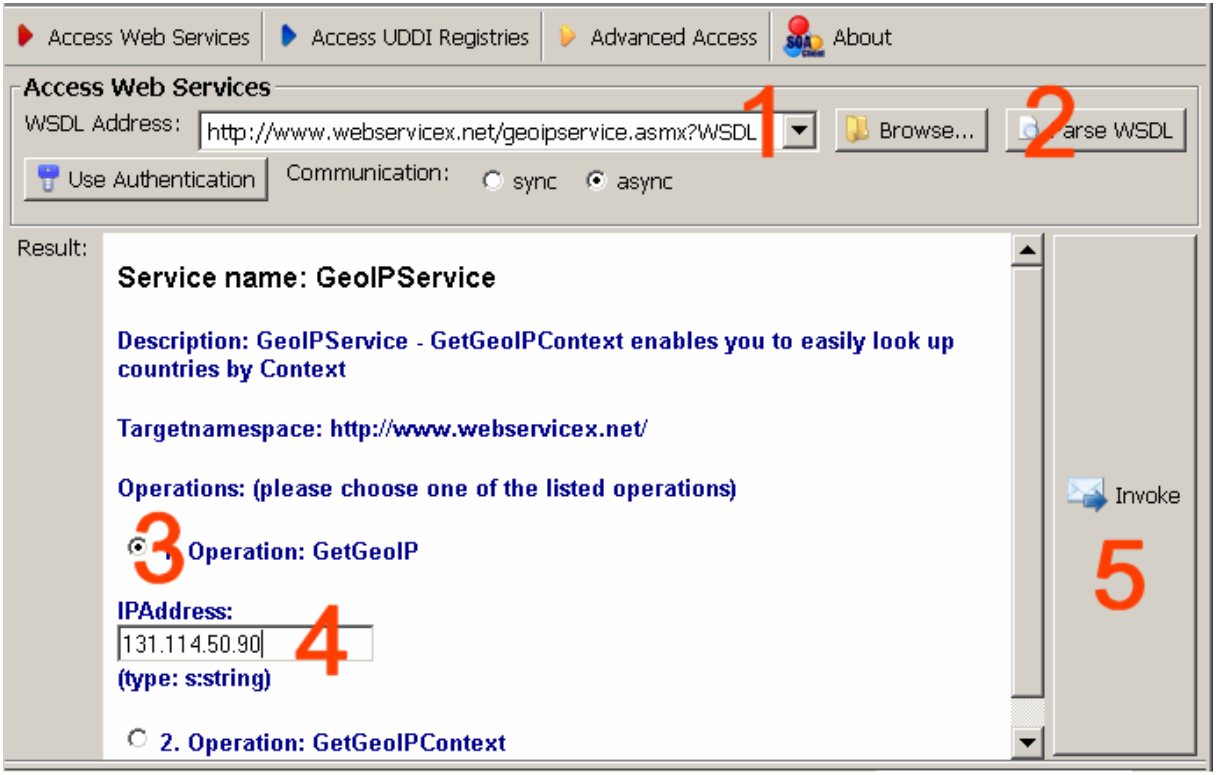
- Nel seguito proveremo a invocare un servizio web tramite una applicazione (SOA Client) che si può installare su Firefox, e vedremo lo scambio di messaggi in XML.

- Download: https://addons.mozilla.org/en-us/firefox/addon/soa-client/. Firefox si accorge che si tratta di un add-ons e ne chiede l'installazione. Dopodichè occorre riavviare Firefox.

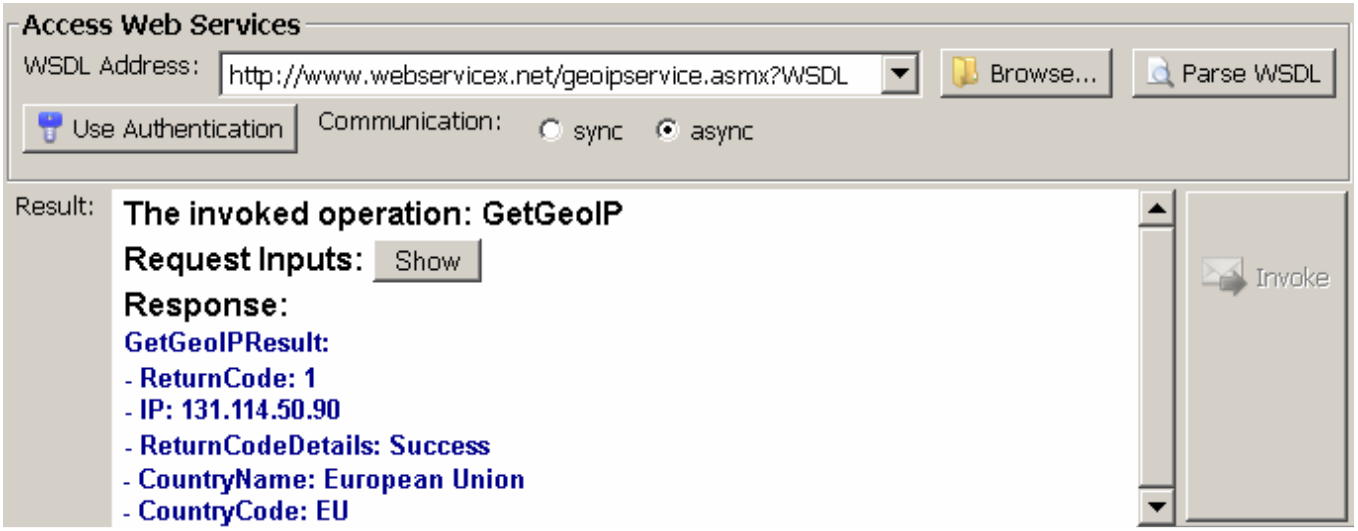
- L'applicazione si accende dal menu Tools => SOA Client => Access Web Services.



- Esempio di web service: GeoIP: dando l'indirizzo IP di una macchina risponde indicando in che paese si trova quella macchina.
- Passi da compiere (vedere immagine seguente):
  1) Inserire l'indirizzo http://www.webservicex.net/geoipservice.asmx?WSDL nel campo WSDL address.
  2) Cliccare su Parse WSDL. Appare nel riquadro in basso un form per inserire i parametri.
  3) Selezionare l'opzione GetGeoIP
  4) Inserire il campo IPaddress.
  5) Cliccare sul pulsante Invoke.



- Es. dando 131.114.50.90 restituisce Europa



- Cliccando in basso sulle schede Raw request Body e Raw response Body è possibile vedere direttamente lo scambio di messaggi XML tra le due macchine. In particolare il tag **<soap:Body>…</soap:Body>** racchiude il corpo del messaggio vero e proprio.

- Facendone un "copia/incolla" e **indentando** (ossia mettendo il codice su linee multiple, ciascuna con un rientro che ne evidenzi l'annidamento dei tag) si ottiene:

- Raw request Body:

```
<GetGeoIP xmlns="http://www.webservicex.net/">
  <IPAddress>131.114.58.30</IPAddress>
</GetGeoIP>
```

- Raw response Body

```
<GetGeoIPResponse xmlns="http://www.webservicex.net/">
   <GetGeoIPResult>
     <ReturnCode>1</ReturnCode>
     <IP>131.114.58.30</IP>
     <ReturnCodeDetails>Success</ReturnCodeDetails>
     <CountryName>European Union</CountryName>
     <CountryCode>EU</CountryCode>
   </GetGeoIPResult>
</GetGeoIPResponse>
```

- Esercizio: tradurre in linguaggio XML i seguenti dati: “Un container ha all'interno un lotto di frutta con 23 cassette di marca Bonita, con data di produzione (espressa in formato gg/mm/aaaa) 30/04/2013, data di scadenza 30/05/2013, temperatura massima delle ultime 24 ore è stata 4 (espressa in °C), e lo spedizioniere che verrà a prelevarlo è Rossi s.n.c, con partita IVA 123456789”.

```
<container>
 <lotto_di_frutta>
  <numero_cassette>23</numero_cassette>
  <marca>Bonita</marca>
  <data_produzione formato="gg-mm-aaaa">
    30/04/2013
  </data_produzione>
  <data_scadenza>30/05/2013</data_scadenza>
  <temperatura_max_24h unita_misura="°C">
    4
  </temperatura_max_24h>
  <spedizioniere>
    <denominazione>Rossi s.n.c</denominazione>
    <partita_iva>123456789</partita_iva>
  </spedizioniere>
 </lotto_di_frutta>
</container>
```