UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II
Facoltà di Ingegneria
Corso di Studi in Ingegneria Informatica

tesi di laurea

Web 2.0 e Web Services nell'era dei Social Network. Snal: un caso di studio

2010/2011

relatore
Ch.mo prof. Porfirio Tramontana

candidato
Marcello Roerhssen di Cammerata
Matr. 534/001550

Problematiche affrontate:

- Dimostrare che i Social Network possono essere veri e propri Web Services

- Sviluppare un Web Service in grado di astrarre le interazioni con i Social Network

Il punto sui Web Services

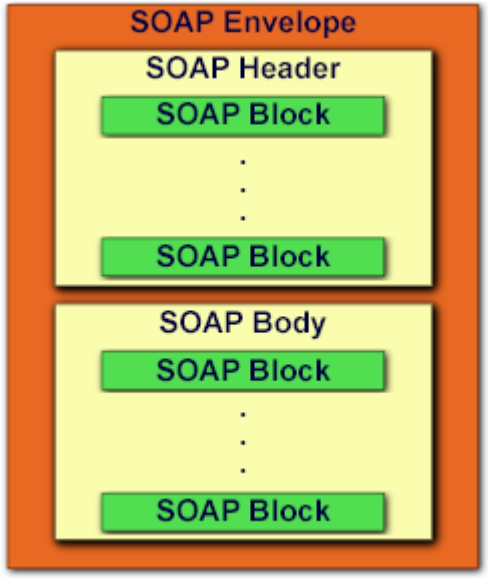
Protocolli esistenti:

- **SOAP**
  - Il messaggio è “imbustato” in XML
  - Estensibile e decentralizzato
- **XML-RPC**
  - Usa XML per lo scambio di messaggi
  - Consente di effettuare una chiamata come se fosse locale
- **Rest**
  - Le funzionalità sono divise in Risorse accessibili tramite URI
  - Stateless, Cachable a livelli

Un Esempio di Web Service SOAP: Amazon AWS

Mostriamo un esempio di Web Service prendendo in considerazione uno dei più famosi fornitori di servizio: AMAZON AWS

- **Protocollo di comunicazione: SOAP**
- **Sicurezza garantita da firma digitale e da certificati di sicurezza X.509**



Normale SOAP envelope

```
<DescribeDBInstancesResponse
xmlns="http://rds.amazonaws.com/admin/2009
-10-16/">
  <DescribeDBInstancesResult>
    <DBInstances>
      <DBInstance>I-f8n42</DBInstance>
      <DBInstance>I-f8n43</DBInstance>
      <DBInstance>I-f8n44</DBInstance>
    <DBInstances>
  </DescribeDBInstancesResult>
  <ResponseMetadata>
    <RequestId>946cda70-c3f1-11de-807a-
79c03c55f7d4</RequestId>
  </ResponseMetadata>
</DescribeDBInstancesResponse>
```

SOAP envelope di AWS

Social Network e Web Services

FLICK

- Utilizza, come protocollo, sia Rest che XML-RPC a seconda dei parametri passati
- Utilizza OAuth per l'autenticazione
- I formati di risposta possono essere sia XML che JSON a seconda dei parametri passati

```
jsonFlickrApi({
    "stat": "ok",
    "blogs": {
    "blog": [{
        "id" : "73",
        "name"  : "Bloxus test",
        "needspassword" : "0",
        "url" : "http://remote.bloxus.com/"
    }]
)
```

TWITTER

- Utilizza OAuth per l'autenticazione
- Restituisce risposte in JSON
- Le API core di Twitter sono utilizzate dagli stessi sviluppatori nel sito www.twitter.com

```
TwitterJsonApi({
    "completed_in":0.088,
    "max_id":114000917262643200,
    "max_id_str":"114000917262643200",
    "page":1,
    "query":"%40twitterapi+%40anywhere",
    "refresh_url":"?since_id=114000917262643200&q=
%40twitterapi%20%40anywhere",
    "results"[],
    "results_per_page":15,
    "since_id":0,
    "since_id_str":"0"
})
```

Facebook come Web Service

- Utilizza il protocollo RESTful
- Utilizza OAuth per l'autenticazione
- Le risposte sono di tipo json
  - Ogni oggetto di risposta appartiene ad un grapho, cioè ogni oggetto è legato agli altri appartenenti al dominio API di Facebook.
- Prima di poter essere utilizzata, una applicazione deve essere autorizzata dall'utente

- Le risposte fornite sono Json

```
{
  "id": "1565911306",
  "name": "Marcello Roehrssen",
  "username": "marcello2",
  "first_name": "Marcello",
  "last_name": "Roehrssen",
  "link":
"http://www.facebook.com/marcel
lo2",
  "gender": “male”,
  "locale": “it_IT”
}
```

Web Service tramite Thrift

Cos'è Thrift :

- una software library
- un insieme di strumenti di generazione di codice

Come nasce:

- sviluppato da Facebook per accelerare lo sviluppo di web-services efficienti e scalabili.

Cosa consente di fare

- Il suo obiettivo primario è quello di consentire una comunicazione efficiente ed affidabile attraverso linguaggi di programmazione differenti astraendo quelle porzioni di codice proprietarie di ogni linguaggio, che tendono a richiedere la maggior manutenzione

Stack di comunicazione di Thrift:

- **Tipi:** Nasce dalla necessità che uno sviluppatore possa essere in grado di scambiare oggetti più o meno complessi in maniera del tutto trasparente. Se un programmatore java vuole scambiare un oggetto di tipo List<T> non deve scrivere codice di controllo e non deve entrare nei sottolivelli che non gli competono.
- **Trasporto:** il livello di trasporto dovrebbe essere anch'esso del tutto trasparente. Lo sviluppatore non deve interessarsi del fatto che il protocollo di trasporto sia implementato tramite Filesystem, socket o TCP
- **Protocollo:** al fine della comunicazione non è importante neanche che lo sviluppatore conosca il protocollo di trasporto, Esso può essere implementato tramite XML, o protocollo binario senza che il developer se ne renda conto.
- **Processors**: Deve essere possibile che gli sviluppatori implementino le chiamate RPC come meglio credono. Unico vincolo deve essere l'interfaccia del servizio definito nell' IDL.

Sistemi a confronto

- **SOAP. XML-based**.
  - Progettato per i servizi web tramite HTTP,
  - comporta un eccessivo overhead sul parsing dell' XML.
- **CORBA**.
  - Relativamente completo,
  - eccessivamente sovradimensionato.
  - Comporta l'installazione di software ingombrante .
- **COM**.
  - Strettamente legato al software, sopratutto nei Client Windows.
  - Non è una soluzione completamente aperta.
- **Thrift**.
  - Prevede la distribuzione della libreria al client
  - velocità di scrittura
  - portabilità alta

Progettazione e sviluppo di SNAL

Requisiti funzionali:

- Gestire una grossa base di utenti appartenenti ad un network di applicazioni presenti su Facebook
- Garantire operazioni fondamentali:
  - GET
  - Check
  - Insert
  - Delete
- L'utente deve avere queste caratterstiche fondamentali:
  - Nome completo, data di nascita, localizzazione geografica, relazione sentimentale, grado di studio
  - Dovrà essere possibile effettuare statistiche su gli utenti iscritti

Scelte progettuali

- Utilizzo di istanze Amazon EC2.

Questo garantisce la velocità nella comunicazione e la sicurezza del servizio.

- Utilizzo di AAS e di ELB.

Garantiremo la scalabilità del servizio in base al carico.

- Utilizzo di Thrift.

Ciò garantirà l'accesso al servizio in maniera semplice generando il codice di comunicazione per il Server.

- Per consentire l'accesso al servizio di applicazioni differenti è necessario che ogni applicazione sia identificata univocamente.

SNAL: Servizio Web

Di seguito un estratto dell'IDL da cui poi Thrift genererà il codice sorgente del servizio

```
service Snal {
    void insert(1:long userid, 2:access_token ac) throws (1:InvalidUser ouch),
    bool check(1:long userid) throws (1:InvalidUser ouch),
    void remove(1:long userid) throws (1:InvalidUser ouch),
    User get(1:long userid) throws (1:InvalidUser ouch)
}
```

Interfaccia Thrift

Codice generato

```
public interface Iface {

  public void insert(long userid, String ac) throws InvalidUser, org.apache.thrift.TException;

  public boolean check(long userid) throws InvalidUser, org.apache.thrift.TException;

  public void remove(long userid) throws InvalidUser, org.apache.thrift.TException;

  public User get(long userid) throws InvalidUser, org.apache.thrift.TException;

}
```

Conclusioni

- Il mondo dei Web Services si è arricchito di un nuovo aspetto:

i Social Network

- Thrift solleva gli sviluppatori dall'onere di scrivere codice di controllo nei Web Services.

Il developer dovrà solo implementare la logica di business

Sviluppi futuri

- Creare un web sempre più “user-centered”

- Ogni web application avrebbe indubbi vantaggi dall'utilizzo dei Social Network come servizi Web