# Introduction

The Semantic Web is envisioned as a decentralized worldwide information space for sharing machine-readable data with minimum integration costs.

Resources are identified by ([URIs](http://www.ietf.org/rfc/rfc3986.txt)). This modeling approach is at the heart of the Resource Description Framework ([RDF](http://www.w3.org/TR/2004/REC-rdf-primer-20040210/) -  standard model for data interchange on the Web)

RDF extends the linking structure of the Web to use URIs to name the relationship between things and the two ends of the link (usually referred to as a “triple”). Using this simple model, allows structured and semi-structured data to be mixed, exposed, and shared across different applications.

# URIs for Web Documents

**http://www.example.com/**

the homepage of Example Inc.

**http://www.example.com/people/alice**

the homepage of Alice

**http://www.example.com/people/bob**

the homepage of Bob

# HTTP and Content Negotiation

Web clients and servers use the HTTP protocol to request representations of Web documents and send back the responses.

**content negotiation** – *a mechanism for offering different formats and language versions of the same Web document*

an HTML or XHTML representation of http://www.example.com/people/alice in English or German:

GET /people/alice HTTP/1.1

Host: www.example.com

Accept: text/html, application/xhtml+xml

Accept-Language: en, de

The server could answer:

HTTP/1.1 200 OK

Content-Type: text/html

Content-Language: en

Content-Location: http://www.example.com/people.en.html

Content negotiation is often implemented with a twist: Instead of a direct answer, the server redirects to another URL where the appropriate representation is found:

HTTP/1.1 302 Found

Location: http://www.example.com/people/alice.en.html

(302- Status Code)

# URIs for Real-World Objects

On the Semantic Web, URIs identify not just Web documents, but also real-world objects like people and cars, and even abstract ideas and non-existing things like a mythical unicorn. We call these **real-world objects**.

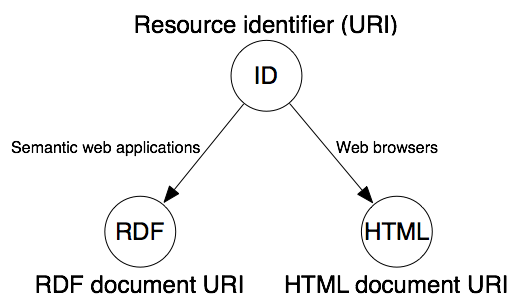
Inc. wants to publish the contact data of their employees on the Semantic Web so their business partners can import it into their address books. For example, the published data would contain these statements about Alice, written here in:

**<URI-of-alice>** a **foaf:Person**;

foaf:name **"Alice"**;

foaf:mbox **<mailto:alice@example.com>**;

foaf:homepage **<http://www.example.com/people/alice>**



# Hash URIs

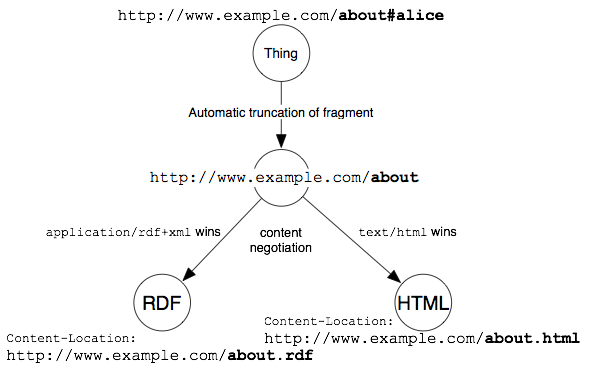
URIs can contain a fragment, a special part that is separated from the rest of the URI by “#”

**http://www.example.com/about#exampleinc**

Example Inc., the company

**http://www.example.com/about#bob**

Bob, the person



# Cool URIs

**Simplicity**

Short, mnemonic, and easier to remember, e.g. when debugging your Semantic Web server.

**Stability**

Once you set up a URI to identify a certain resource, it should remain this way as long as possible.

**Manageability**

One good practice is to include the current year in the URI path, so that you can change the URI-schema each year without breaking older URIs.

# Linking

All the URIs related to a single real-world object—resource identifier, RDF document URL, HTML document URL—should also be explicitly linked with each other to help information consumers understand their relation

Two of them are Web document URLs. The RDF document located at http://www.example.com/data/alice might contain these statements (expressed in N3):

**<**<http://www.example.com/id/alice>**>**

foaf:page **<http://www.example.com/people/alice>**;

rdfs:isDefinedBy **<http://www.example.com/data/alice>**;

a **foaf:Person**;

foaf:name **"Alice"**;

foaf:mbox **<mailto:alice@example.com>**;

...