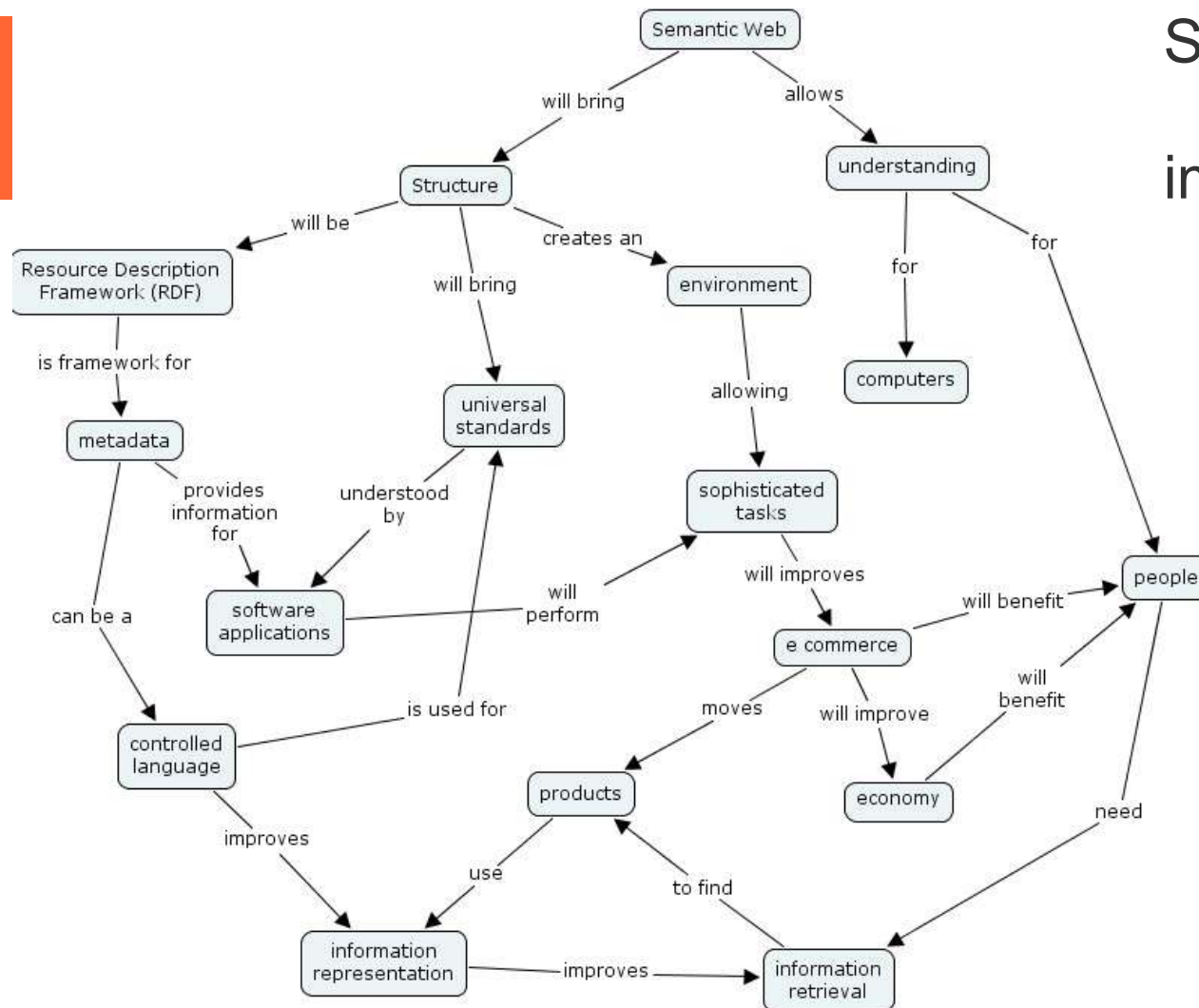


# using Semantic Web related technologies in a business Environment

SW in the life of a company

impact on

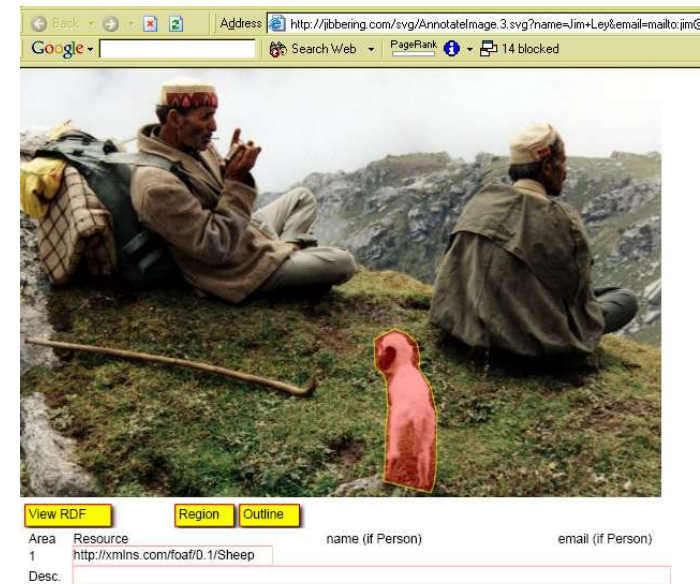
- development
- business



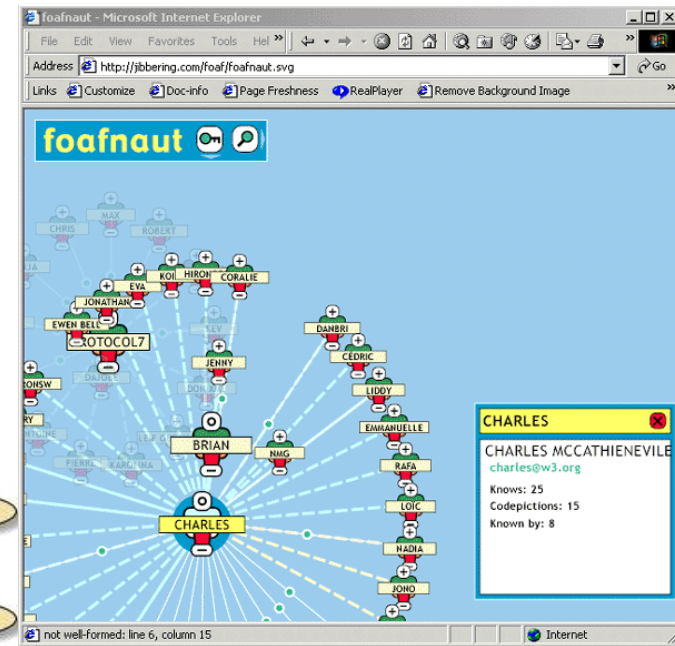
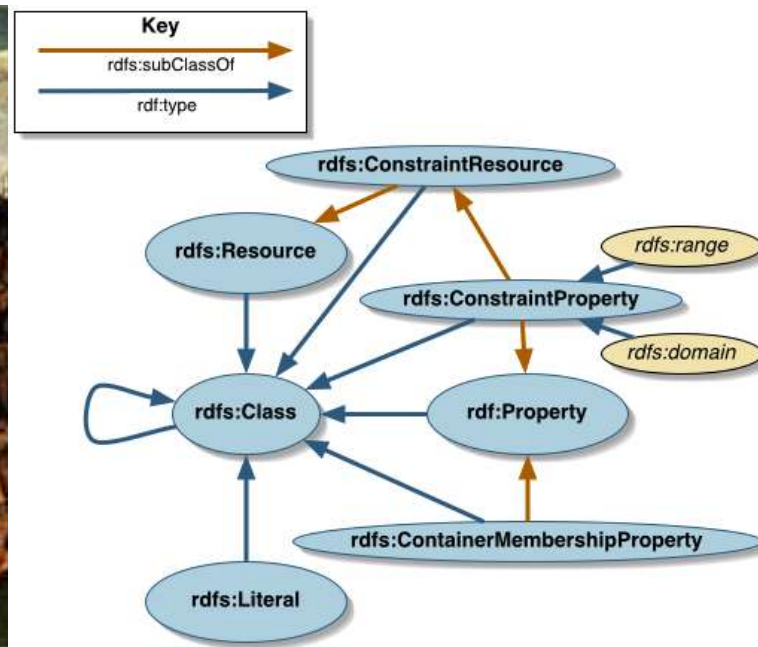
Péter Vojnisek  
et@progos.hu

# what can be SW technologies used for?

- annotation, metadata
  - anything which has a unique identifier
- data handling – content and knowledge
  - storing data
  - searching, mining
  - data transformation
- process description
  - choreography of services
  - business process, supply chain
- software integration
  - sharing & exchanging information
- etc...



# impact on software development 1.



- open and shared **standards, ontologies**

# impact on software development 2.

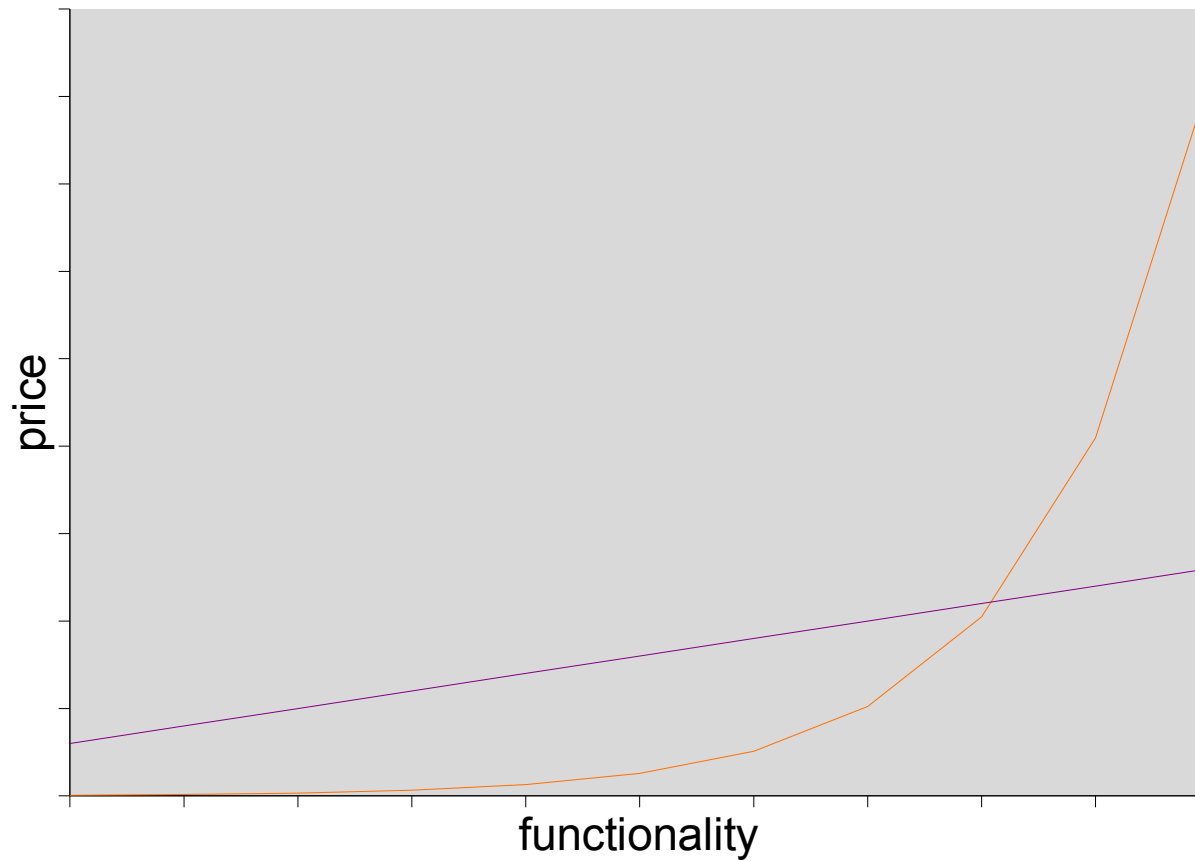
- **reusable** software modules
- integrating **opensource** 3<sup>rd</sup> party products
- more efficient **Rapid Application Development**
- fast **prototyping**

# impact on software development 3.

- build on top of existing software
- **authentication, security**
- less time on **documentation**, administration
- **independent** from data formats or syntax
- less dependencies in a software

# impact on the business 1.

development costs



# impact on the business 2.

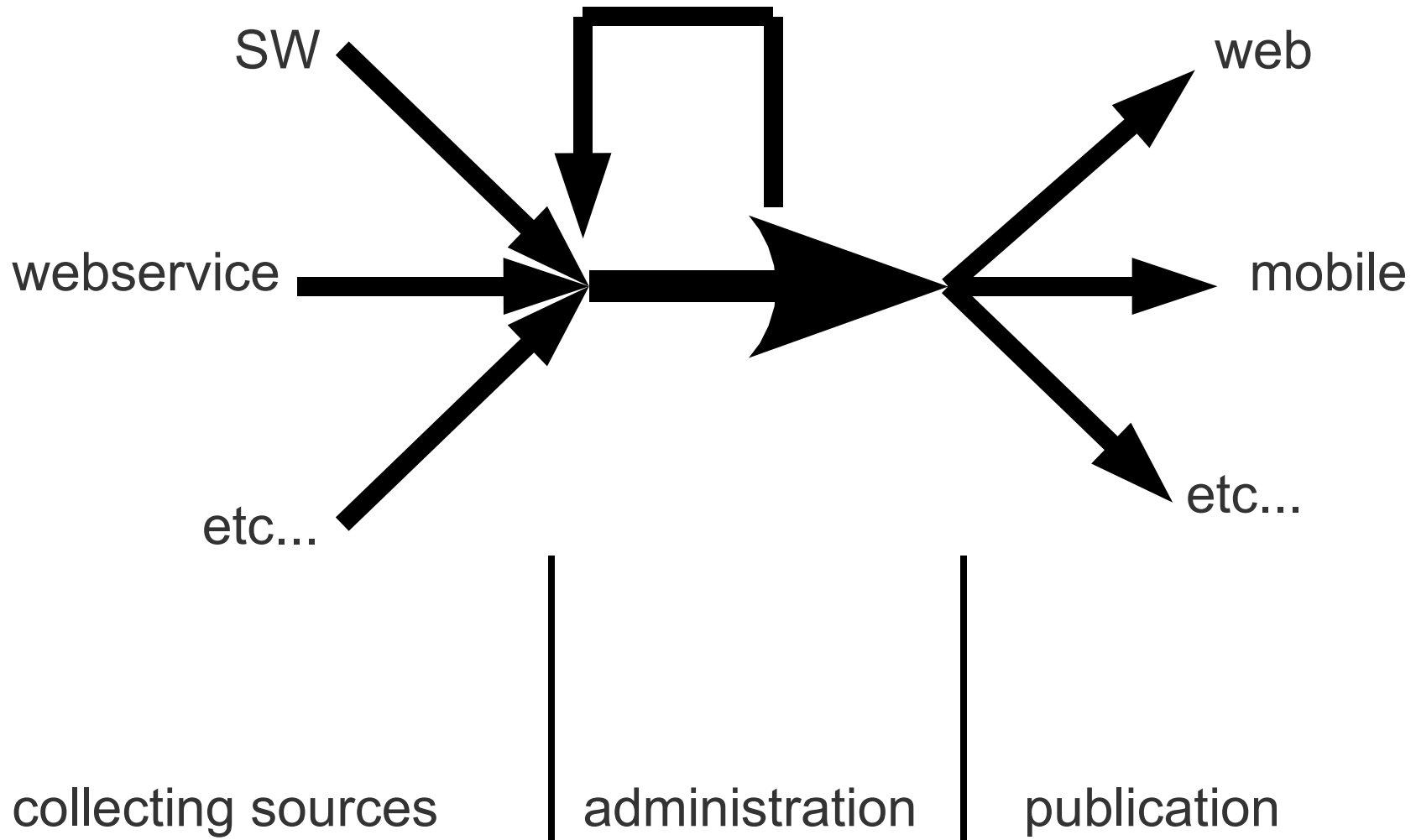
- reduce costs – long term
  - development
  - integration work
- better and wider functionality
  - future-proof solutions
- scalable applications
- become more independent
  - other developer companies
  - 3<sup>rd</sup> party products
- community
- huge potential

# difficulties

- start with a new philosophy can be **risky**
- choosing **development tools**
- must **train your stuff**
- **ontologies** availability, variety
- **little SW content** is available
- **convince** your partners
- **multilingual** content is a problem



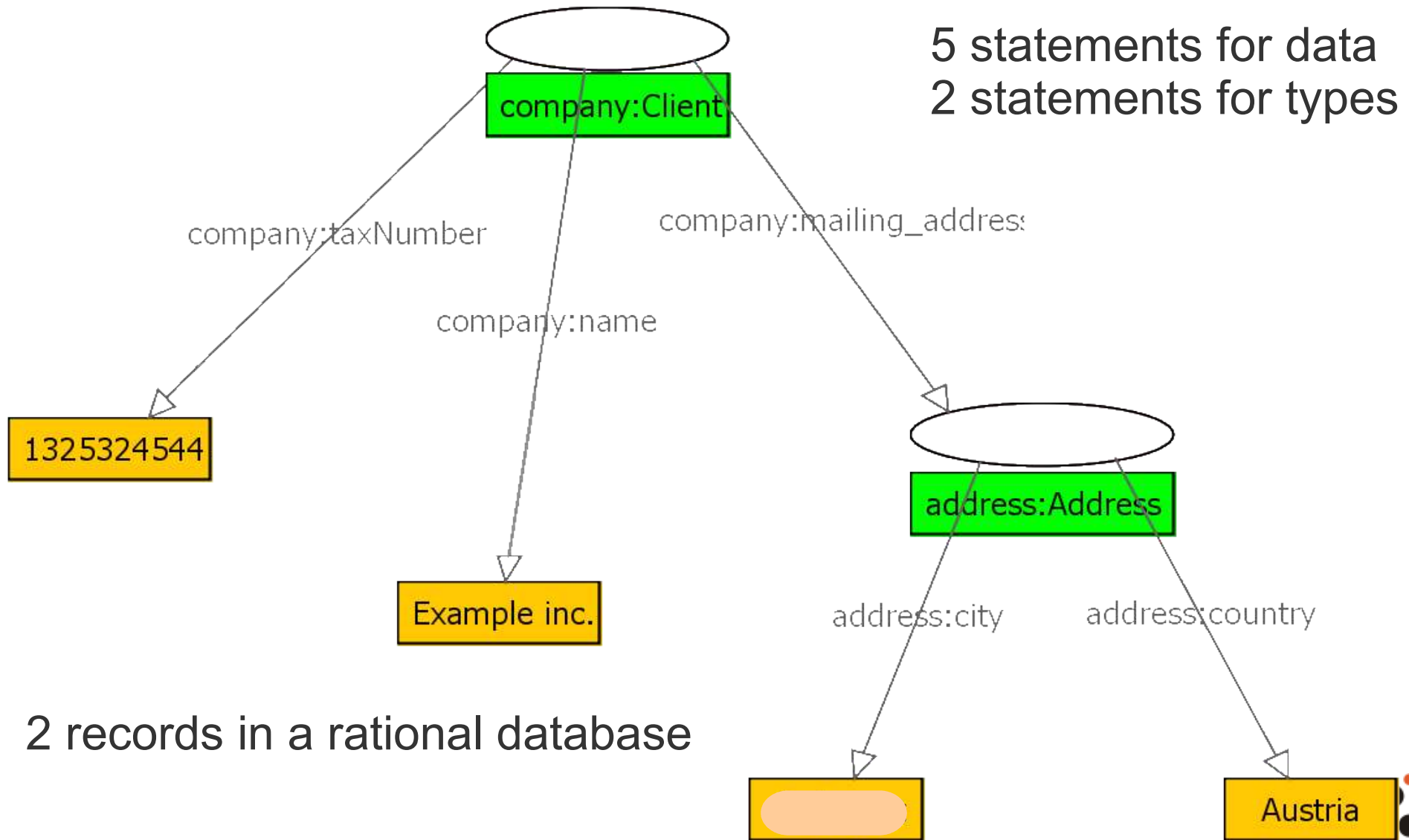
# content management system



# relational database

- relational database
  - table: Clients
    - name – String
    - tax number – String
    - mail address – address
  - table: Addresses
    - country – String
    - city - String

# the same in RDF graph



# comparing relational database to RDF graph

Imagine a database:

- 30 tables
- 15 fields per each (average)
- 10 000 records per table average

Rational database

- 30 x 10 000  
**300 000** records

RDF graph

- 30 x 10 000 type declaration
- 30 x 15 x 10 000 data  
**4 800 000** statements!!!

# how can we store data more efficiently?

- use the advantages of both solutions!
  - use relational base RDF stores!
- use ontologies
  - database mapping

# Querying on RDF graphs

- simple query
  - well supported by APIs
- structured query
  - matching on a shape



# RDF APIs

- Java
  - **Jena**
  - **Thingy** - [progos.hu](http://progos.hu)
- C
  - **RDFStore**
  - **Redland**
- Python
  - **pyrple**
- PHP
  - **RAP**
- .NET
  - **Drive**



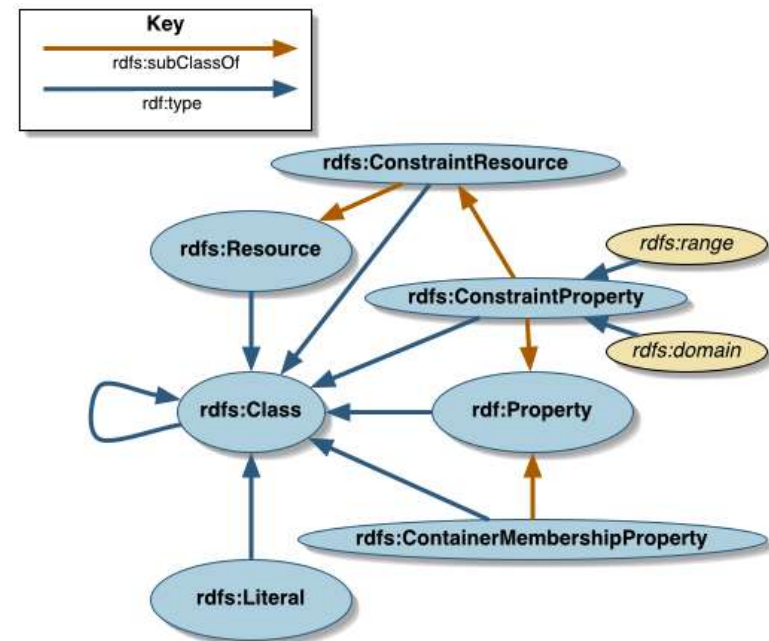
# RDF databases

- RDFStore
  - <http://rdfstore.sourceforge.net/>
- Jena
  - <http://jena.sourceforge.net/>
- Sesame
  - <http://www.openrdf.org/>



# Ontology editors

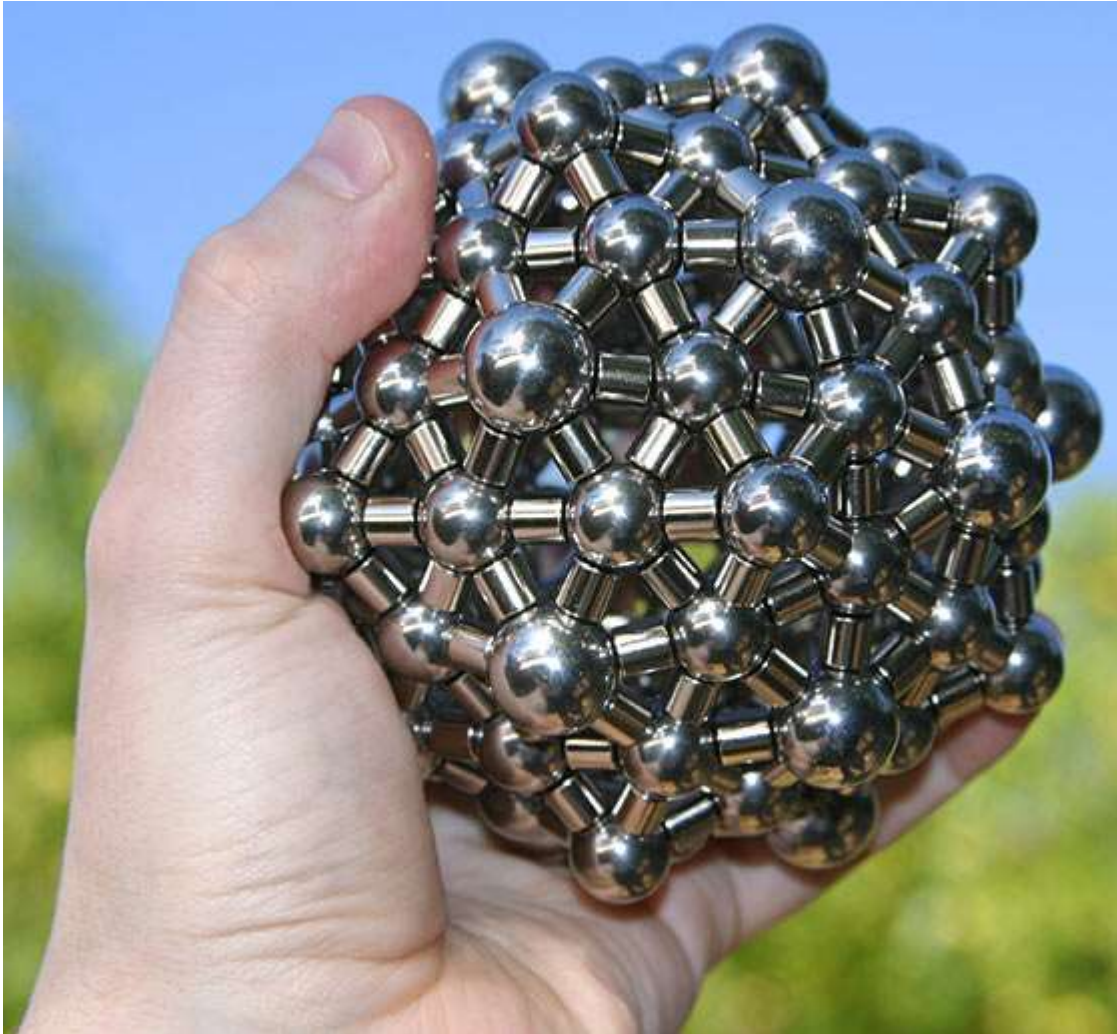
- SWOOP
  - easy to use
- MR3
  - graphical editor
  - manipulate RDF graphs
- Protégé



# Other tools

- RDF validator
  - validates an RDF graph
  - no ontology validation
  - <http://www.w3.org/RDF/Validator/>
- Ontology generator
  - generates ontology from instances
  - <http://progos.hu/tools/og>

# Thank you for your attention!



Péter Vojnisek  
et@progos.hu

