Ruby Topic Maps in Action

Topic Maps 2008, Oslo
April 2nd, 2008

Dmitry Bogachev

Agenda

- Why
  - Topic Maps, Ruby, Ruby on Rails, RTM
- Traditional Rails application with XTM export
  - Subject-centric blog (demo and details)
  - PSI server (demo)
- Subject-centric blog aggregator
  - Implementing application based on RTM
- Subject-centric computing
  - Topic Maps Grid

Installation instructions

- Ruby (with RubyGem)
- Ruby on Rails
- RTM gem

“Subject-centric” blog

SUBJECT-CENTRIC

What if every subject that we think about can have explicit representation in our computers?

Home | Index | About

Thursday, February 28, 2008, 09:47 PM

Authoring topic maps using Ruby-based DSL: CTM, the way I like it

Designing and using Domain Specific Languages (DSL) is a popular programming style in Ruby community. I am experimenting with Ruby-based DSL for authoring topic maps. Surprisingly, the result is very close to my view on the “ideal” CTM (Compact Topic Maps syntax).

I just would like to share a sample that should demonstrate main ideas of this approach. It is a piece of Ruby code that generates topic maps (behind the scenes).

First topic map defines some simple ontology.

```
# some definitions to support DSL
# should be included

topic_map :ontology_tm do

  tm_base "http://www.example.com/topic_maps/people/"

  topic(:person) {
    sid   "http://psi.example.com/Person"
    name  "Person"
    isa :topic_type
  }

  topic(:first_name) {
    sid   "http://psi.example.com/first_name"
```

Ontopedia

Syndication

RSS, XTM 1.0, XTM 2.0

XTM 1.0 (in Omnigator)

Recent subjects

CTM
Domain Specific Language
iPhone
iPod touch
ISO 13250
Multi-touch interaction
OS X Leopard
Resource-Oriented Architecture
RESTful Web Services
Robotics
Ruby (programming language)
Service-oriented architecture (SOA)
Subject-centric computing
TMCL
TMQL
Topic Maps
XTM

“Subject-centric” blog: differentiation

- Explicit representation of relationships between posts and subjects (subjects are first class “citizens”)
- Subjects are in a form of Published Subject Identifiers (PSI)
- Integration with PSI Server (psi.ontopedia.com)
  - the blog uses subjects identified on the PSI Server
- Usage of Dublin Core and Blogging ontologies
- “Traditional” Web app but with “Export as XTM” feature
  - XTM 1.0
  - XTM 2.0

Published Subject Identifiers

- A published subject identifier is an IRI that by definition is deemed to identify a single subject and that was expressly created in order to serve as the identifier of that subject
- A published subject identifier resolves to a human-interpretable document, known as a published subject indicator, that is intended to convey a compelling and unambiguous "indication" of the identity of its subject and explicitly declares itself to be a subject indicator
  - Source: http://www.ontopia.net/topicmaps/materials/The_Case_for_Published_Subjects.pdf

Published Subject Identifiers: why

- Identifiers exist for the benefit of computers that need to be able to ascertain whether two pieces of information (or assertions) are about the same subject: if the identifiers are identical, this can be assumed to be the case; if they are not, no such assumption can be made
- Indicators exist for the benefit of humans whose job it is to assign identifiers to information resources; by examining and interpreting the contents of the subject indicator, a human can gain a notion of the identity of the subject that is sufficient for deciding whether or not to use that PSI.
  - Source: http://www.ontopia.net/topicmaps/materials/The_Case_for_Published_Subjects.pdf

“Subject centric” blog: demo

SUBJECT-CENTRIC

What if every subject that we think about can have explicit representation in our computers?

Home | Index | About

Subject: Subject-centric computing

It is a branch of computing theory and practice which concentrates on explicit representations of subjects that are important for human activities in computer systems.

PSI: http://psi.ontopedia.net/Subject-centric_computing

Related posts:

Monday, December 24, 2007, 08:32 PM

Subject-centric computing and robotics: Osaka will soon be known as the capital of the robotics world..?

I was in Kyoto for three days in December. Osaka-Kobe-Kyoto is a region with high concentration of companies involved in robotics. I cannot stop thinking about robotics and Subject-centric computing after this trip. Traditionally, when we talk about Subject-centric computing (SCC) and Topic Maps (as enabling technology), we assume more or less slowly evolving models. In the world of robotics, models are evolving in real time. ...

· Robotics · Subject-centric computing ·

Ontopedia

Syndication

RSS, XTM 1.0, XTM 2.0

XTM 1.0 (in Omnigator)

Recent subjects

CTM
Domain Specific Language
iPhone
iPod touch
ISO 13250
Multi-touch interaction
OS X Leopard
Resource-Oriented Architecture
RESTful Web Services
Robotics
Ruby (programming language)
Service-oriented architecture (SOA)
Subject-centric computing
TMCL
TMQL
Topic Maps
XTM

Implementation Details Overview

- MySQL database with domain specific tables
  - Posts, Subjects, Tagging
  - Sorry, no Topic Maps at this level :)
- Ruby on Rails
  - Traditional usage of ActiveRecord (ORM)
- RSS, XTM export
  - XML Builder
- Doubling Core and Blogging Ontologies are “embedded” in XTM export
  - no explicit representation in database

Implementation details

- Database demo
- Rails demo
  - Models, Controllers, Views, Routing (and nice URIs)
- RSS export
- XTM export
  - XTM 1.0
  - XTM 2.0

Ontopedia PSI server

- Joint project with Steve Pepper
- Based on
  - The Case for Published Subjects
  - TMGrid
- Directory of Published Subject Identifiers

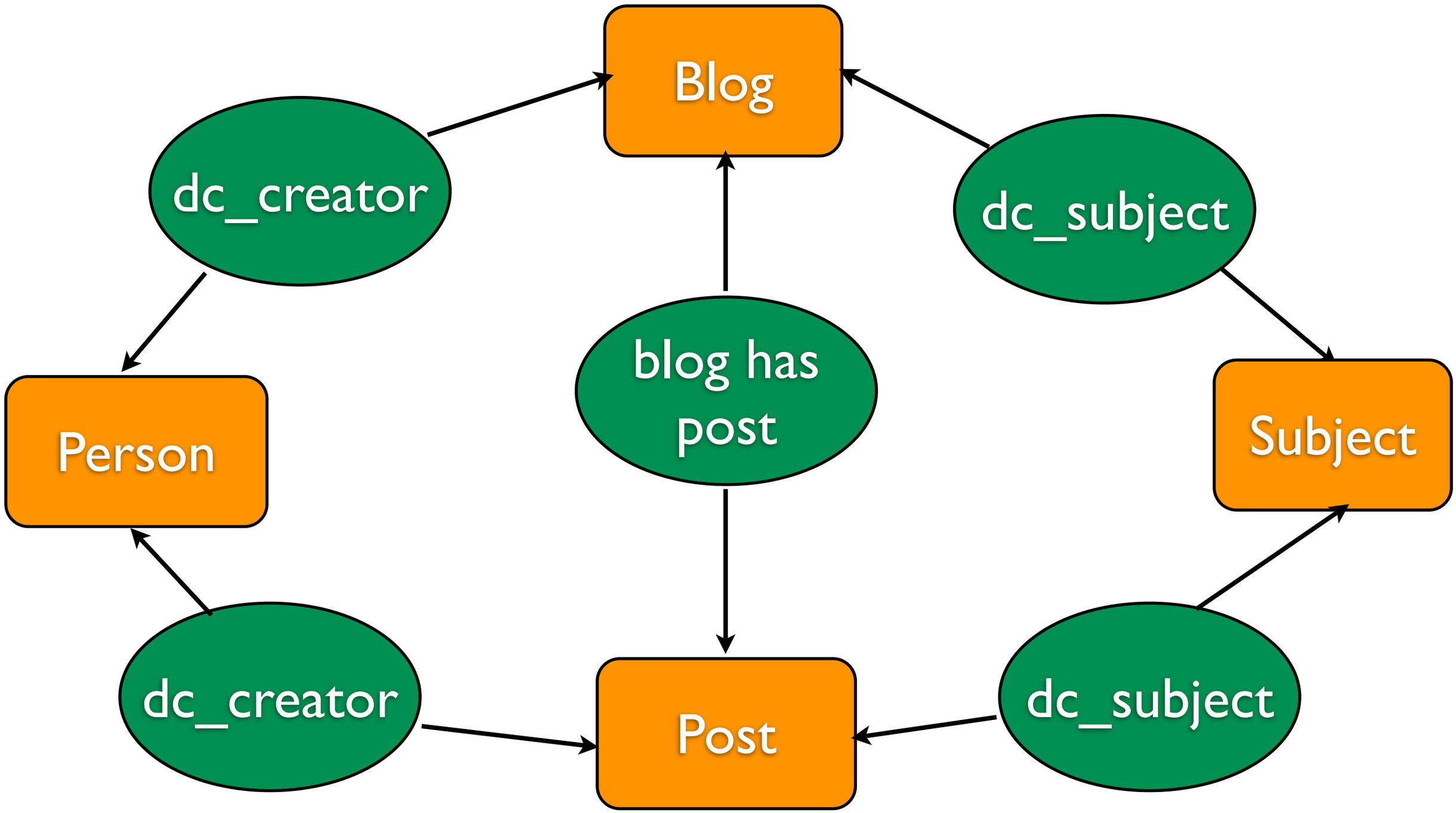
Ontopedia PSI server

- Demo
  - http://psi.ontopedia.net/

Doublin Core and Blogging Ontologies

- "Subject-centric" utilizes Dublin Core and basic Blogging ontology
  - Expressing Dublin Core Metadata Using Topic Maps ISO recommendation draft
  - http://www.ontopedia.net/pepper/papers/DCinTopicMaps.pdf
- Demo in Omnigator and Vizigator
  - http://www.ontopedia.net/omnigator/models/topic_nontopoly.jsp?tm=subject-centric.xtm&id=id_d1386424 4f388d31a24c79ad3f1603dd

Ontology Diagram



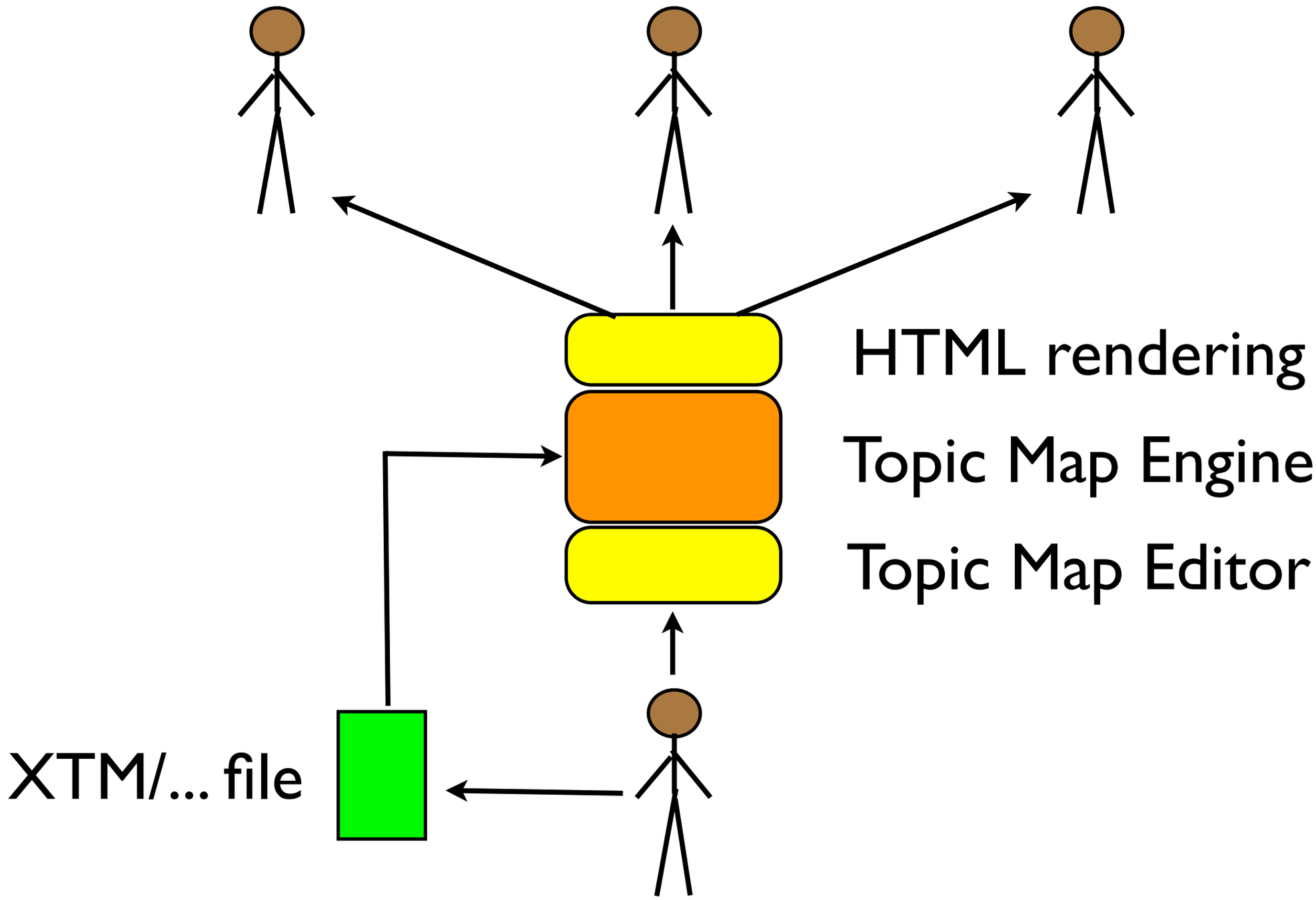
Subject Stream

- Subject-centric blog aggregator
- Demo
- Let's create it!

Topic Maps Grid (TMGrid)

- Presentation on Extreme Markup in 2006
  - http://homepage.mac.com/dmitryv/TopicMaps/TMGRID/TopicMapsGrid.pdf
- From Topic Maps theory to building distributed topic maps infrastructure
- We are on the way of implementing these ideas

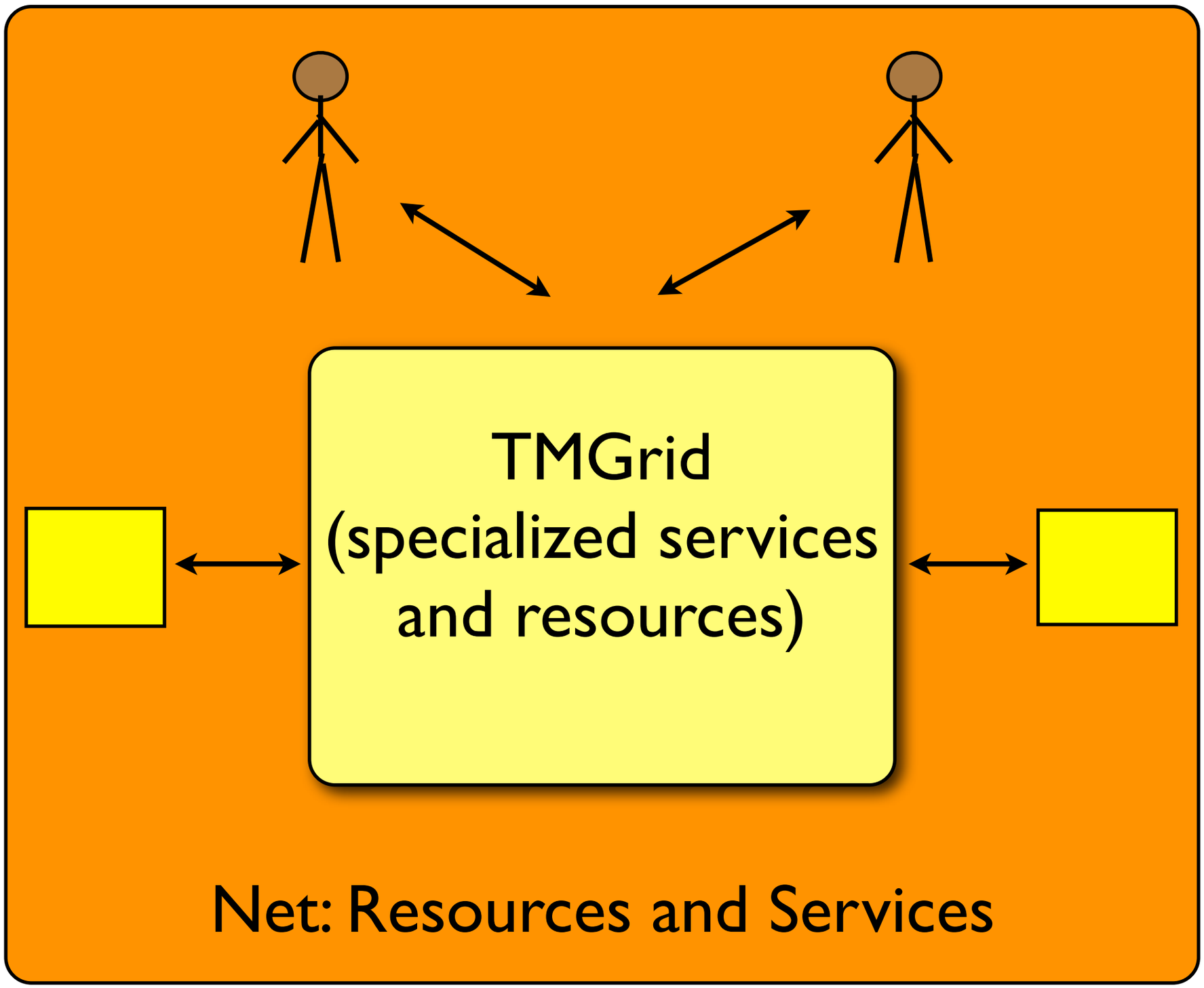
Topic maps: Typical Implementation



Topic Maps Grid

- Cooperating system of services covering different aspects of providing and consuming information based on Topic Maps paradigm
- Each service has well defined contract
- Relatively easy to implement
- Can be re-used for building required activities

Consumers and Providers



TMGrid

- Leveraging ideas of SOA/REST
  - composite applications
- and Grid computing
  - effective usage of distributed environment (storage, services)
- It is not about one centralized "database"
  - even nice one like Freebase
- Support for Presentation and User interaction layer
- Personal topics of interest
- Support for "what if scenarios", fictitious contexts

TMGrid services

- native Topic Maps publishers
- gateways
- transformers
- validators
- aggregators
- directory
- provenance
- my topics
- portal

Fundamental activities on TMGrid

- Creating topics with Published Subject Identifiers (PSIs) for events, people, locations etc., with Subject Locators for resources on the Net
- Mapping between PSIs produced by different PSI providers
- Making assertions about topics (often as a result of interpretation of existing texts)
- Using topics and assertions for solving various problems

Publishing alternatives: “Export as XTM”

- Small assertion sets can be published as XTM files
- Example: "Export as XTM"
- These assertion sets can be consumed by TMGrid aggregators
- and registered in TMGrid directory

Publishing alternatives: What about changes?

- Topic Map aggregators can reload small topic maps from time to time, but...
- Possible solution:
  - publish XTM file with versions (snapshots)
  - add Atom feed with topic map fragments for updates
- Aggregator can process Atom feed and “re-play” updates

Publishing Alternatives: large assertion sets

- Become a "Dynamic Publisher" which supports:
  - request/response
  - transactions recording
  - pub/sub functionality
- Use one of the gateways
- Use native Topic Map server
- Dynamic Publishers use topic map fragments for communication with TMGrid Aggregators

TMGrid gateway types

- Relational database
- RSS/Atom
- Wiki
- RDF
- Domain specific XML (including Web 2.0)
- OpenCyc
- IRC, Jabber, IM channels
- ...

TMGrid database gateway

- Implements gateway to existing databases
  - (mostly in relational form)
- Often is installed "close to database server"
- We would like to do configuration, not programming
- Often includes factual information enriched by mapping to ontology

Database gateway: Ontology reverse engineering

- In most of cases relational databases have "hidden" ontology
  - If we are lucky, we have related UML diagrams
- It is required to do "ontology reverse engineering":
  - identify types
  - occurrences/properties
  - associations
- Identification schema should be clarified and mapped to PSIs (local -> global)

RDF Gateway

- It is optimized for integration of RDF-based resources and distributed networks such as SemanticGrid
- Can be located close to RDF server/storage or it can call RDF server/storage remotely
- Mapping should be based on recommendations from W3C RDF/Topic Maps Interoperability Task Force
  - "A Survey of RDF/Topic Maps Interoperability Proposals" - W3C Working Draft, March 29th 2005,
  - new work item in WG3
- Preferable solution: "semantic mapping"
  - vs. "object mapping"

RSS gateway

- RSS (in combination with gateway) is probably the easiest way to contribute to TMGrid
- RSS has a "category" element which can be used to create occurrences related to specific topics
- Example:

```
<item>
    ...
     <category domain="some domain PSI">
          some topic PSI
     </category>
    ...
</item>
```

RSS gateway: using extensibility

- We can create a module with some TM specific vocabulary which allows to specify additional information such as occurrence type and scope
- RSS module also can be used to attach TM fragments (metadata and/or resource interpretation)
- One of the interesting activities:
  - read an article
  - create topic map interpretation (summary)
  - publish interpretation as an item in RSS/TM feed

Atom feed gateway

- Atom feeds have more semantic commitments (vs. RSS)
- Explicit representation of people (author, contributor), posts, feeds, subjects
- It is easy to convert to XTM

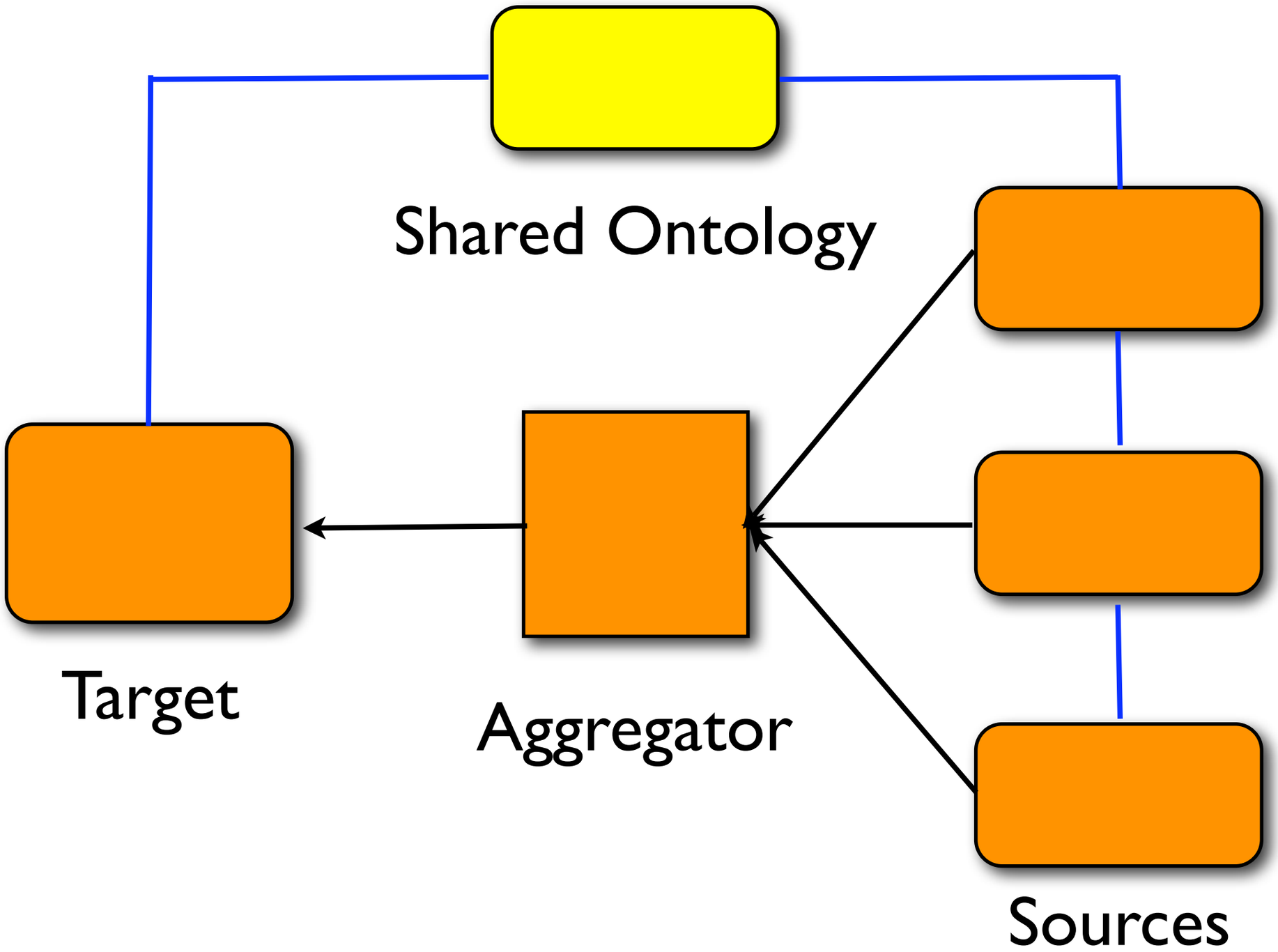
Wiki gateway

- Wiki can be a very important source for contributing information to TMGrid.
- Wikipedia allows to create structured content (“facts”)
  - Infoboxes
  - http://dbpedia.org
- New generation of wiki/topic maps integrated tools
  - Occurrences can keep wiki-like markup
    - “Subject-centric” uses Textile, for example

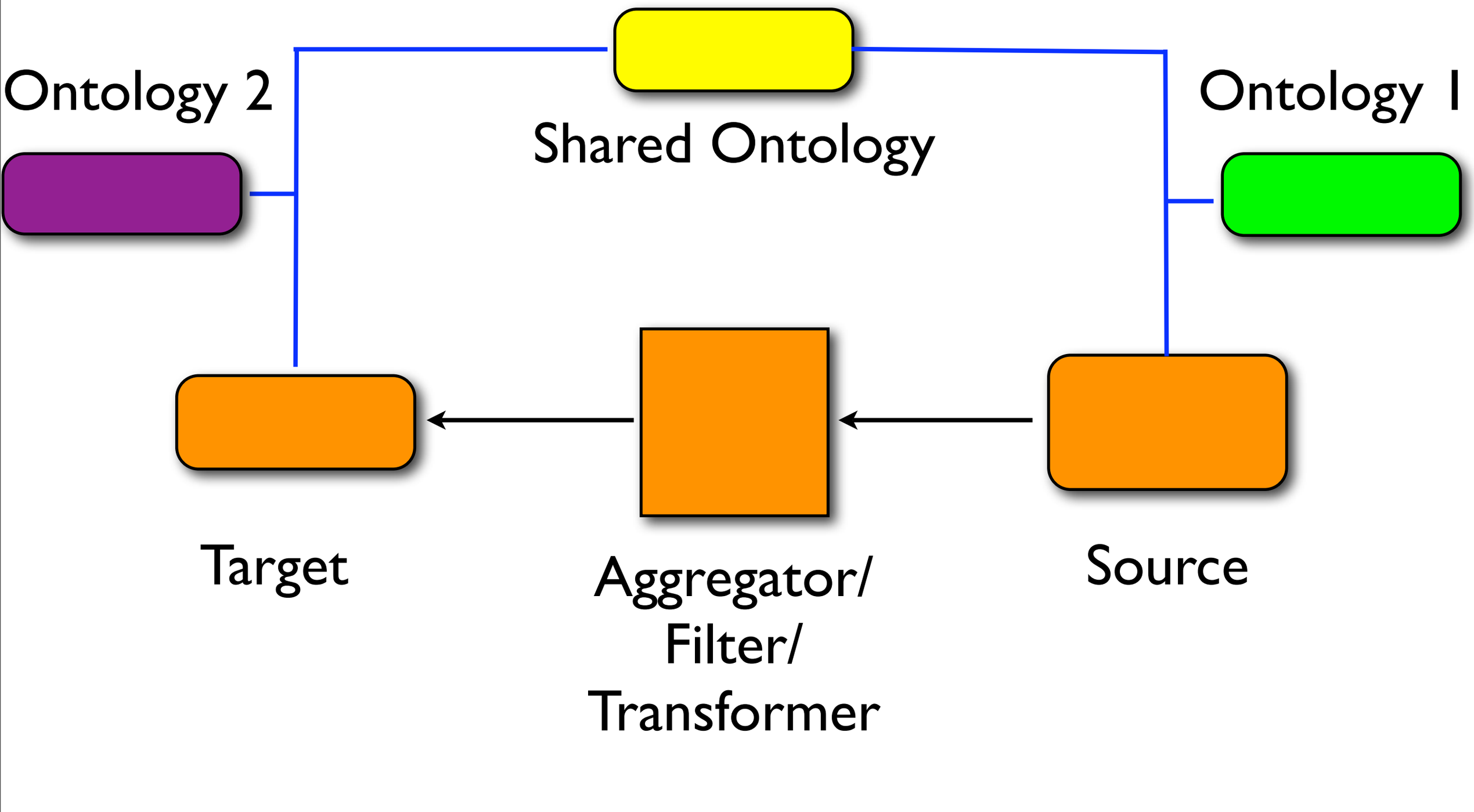
TMGrid Aggregator: basics

- Aggregation by "coping and merging"
- Aggregation with filtering and transformation
  - it is possible to define quite complex relationships between sources and results of aggregation
- "On the fly" aggregation of fragments
  - coping may not be optimal for huge assertion sets

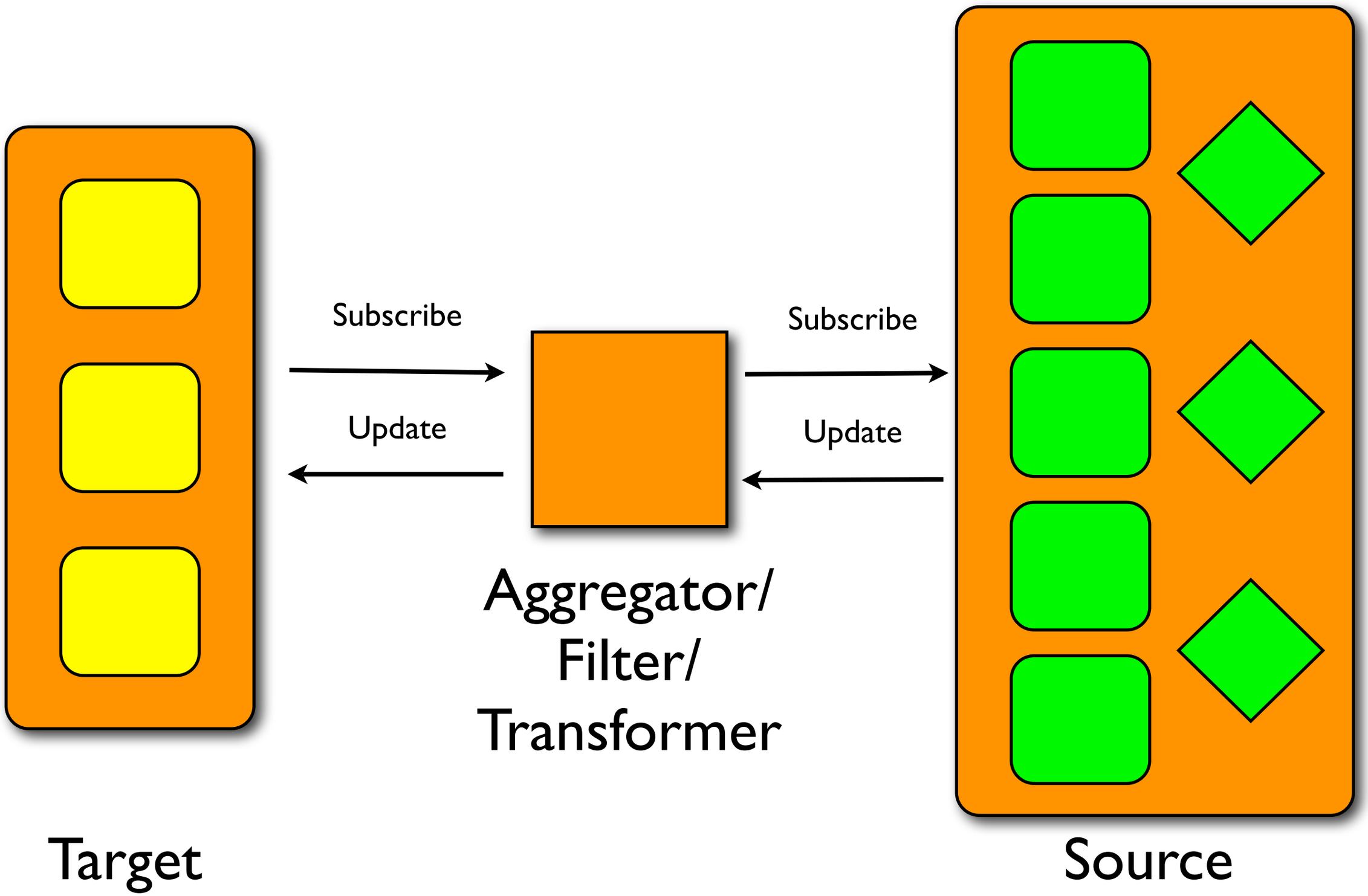
Simple “copy and merge” aggregation



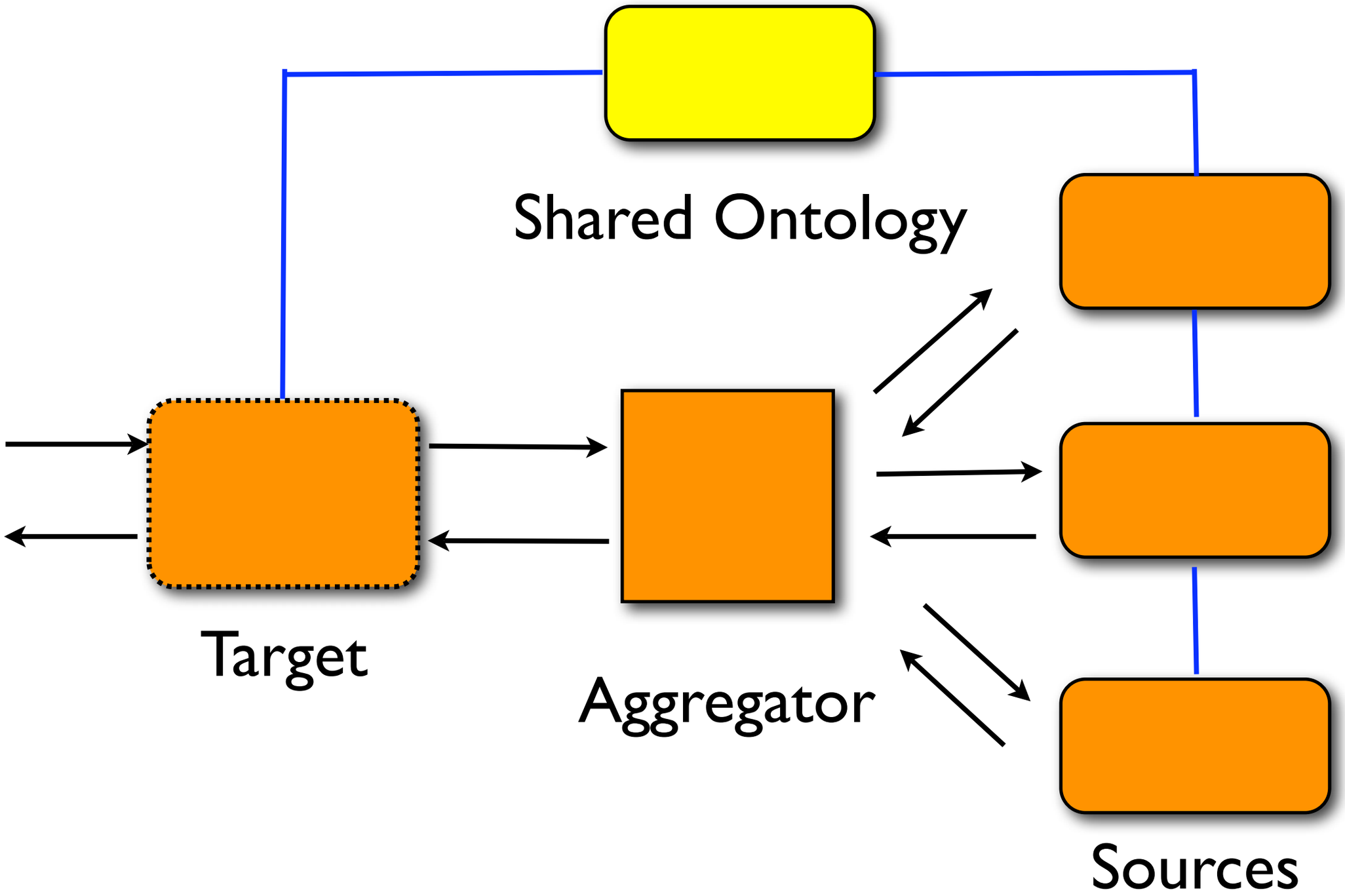
Filtering and transformation



Fragment-based filtering and transformation



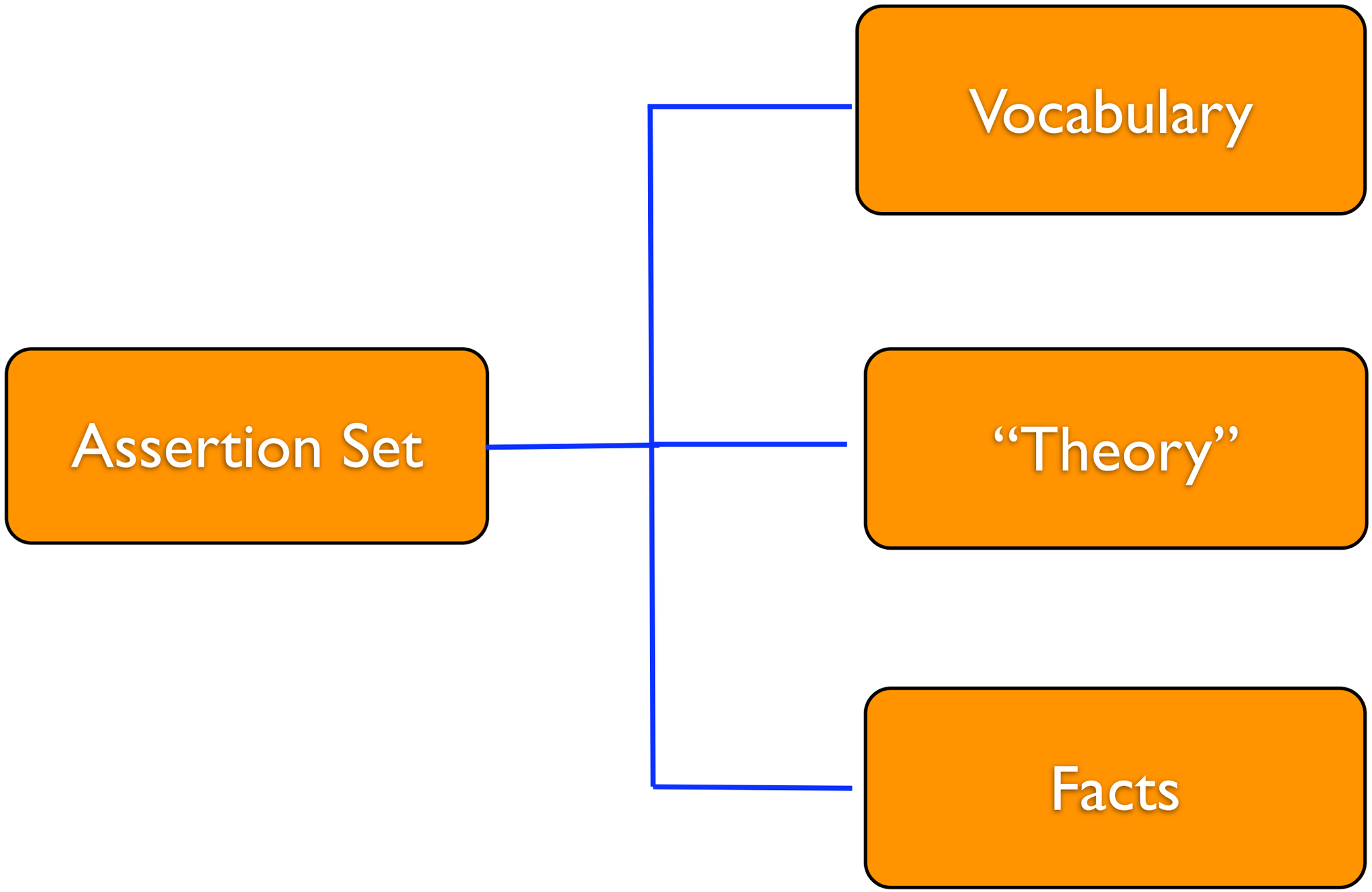
“On the fly” aggregation



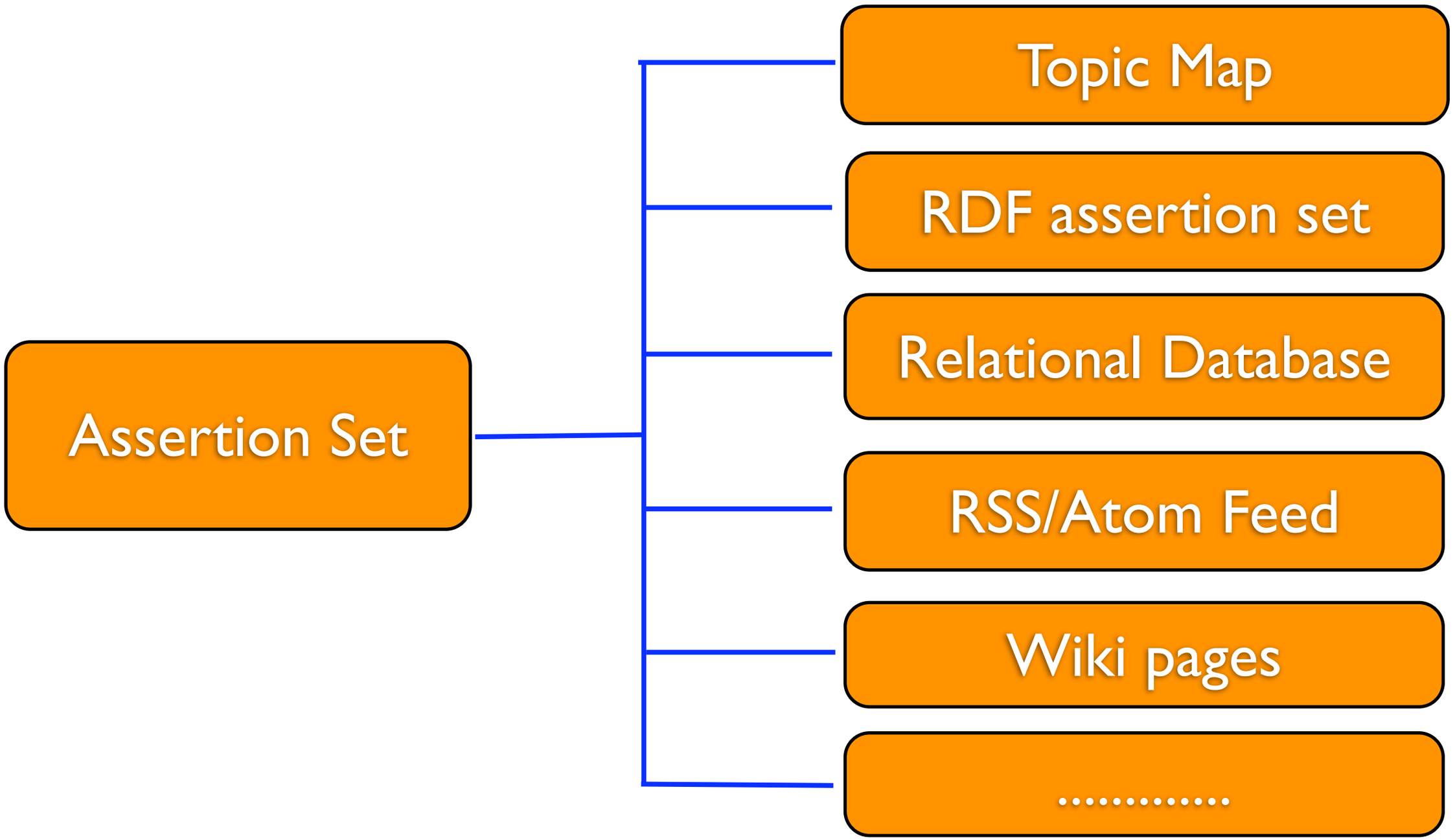
TMGrid Directory

- Repository of Published Subject Identifiers (PSI)
- Is a "map of maps"
  - repository of topic maps and other assertion sets (including available metadata)
- Repository of different services on the TMGrid (gateways, aggregators, transformers ...)

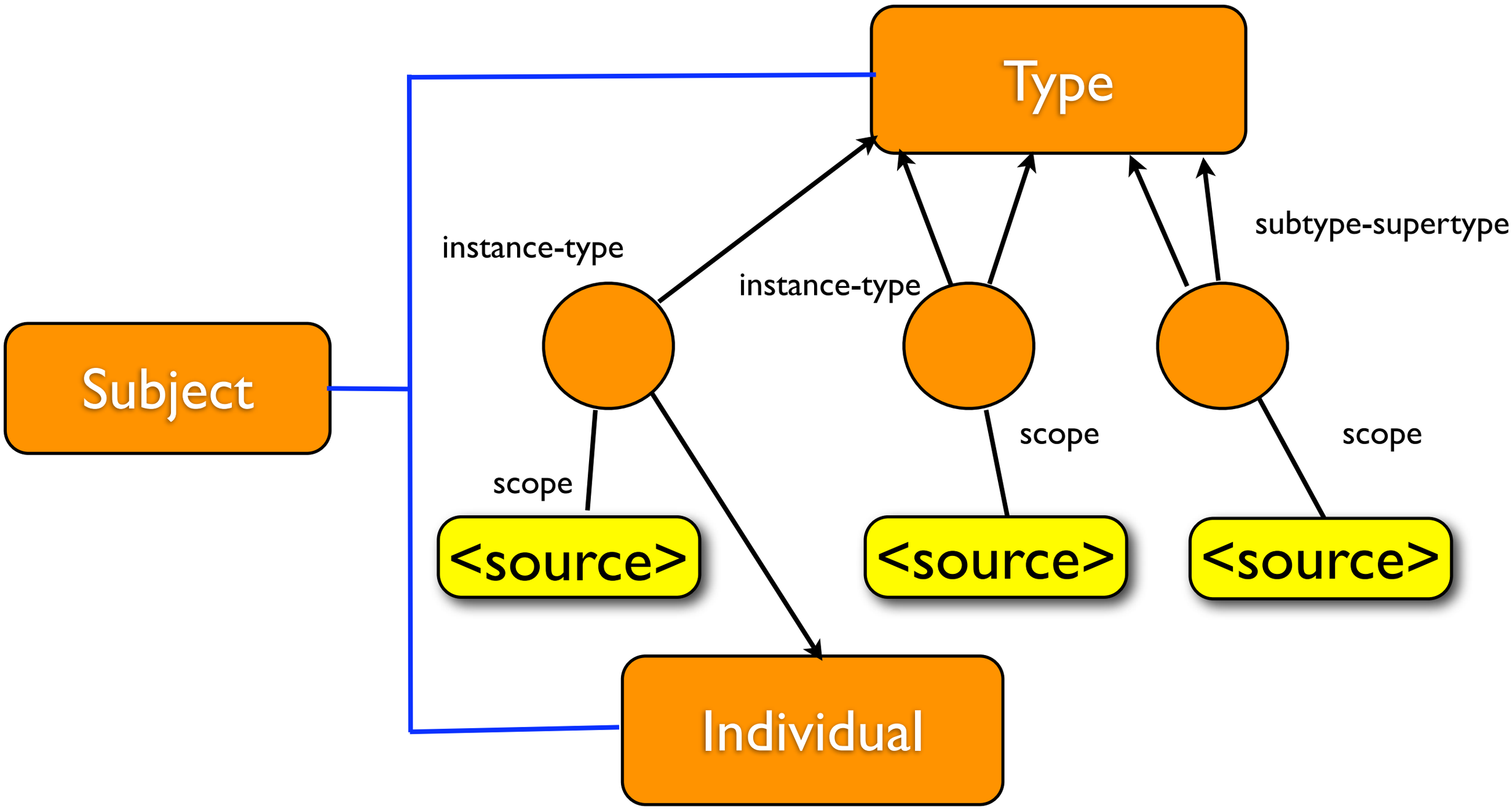
Directory: Assertion Set



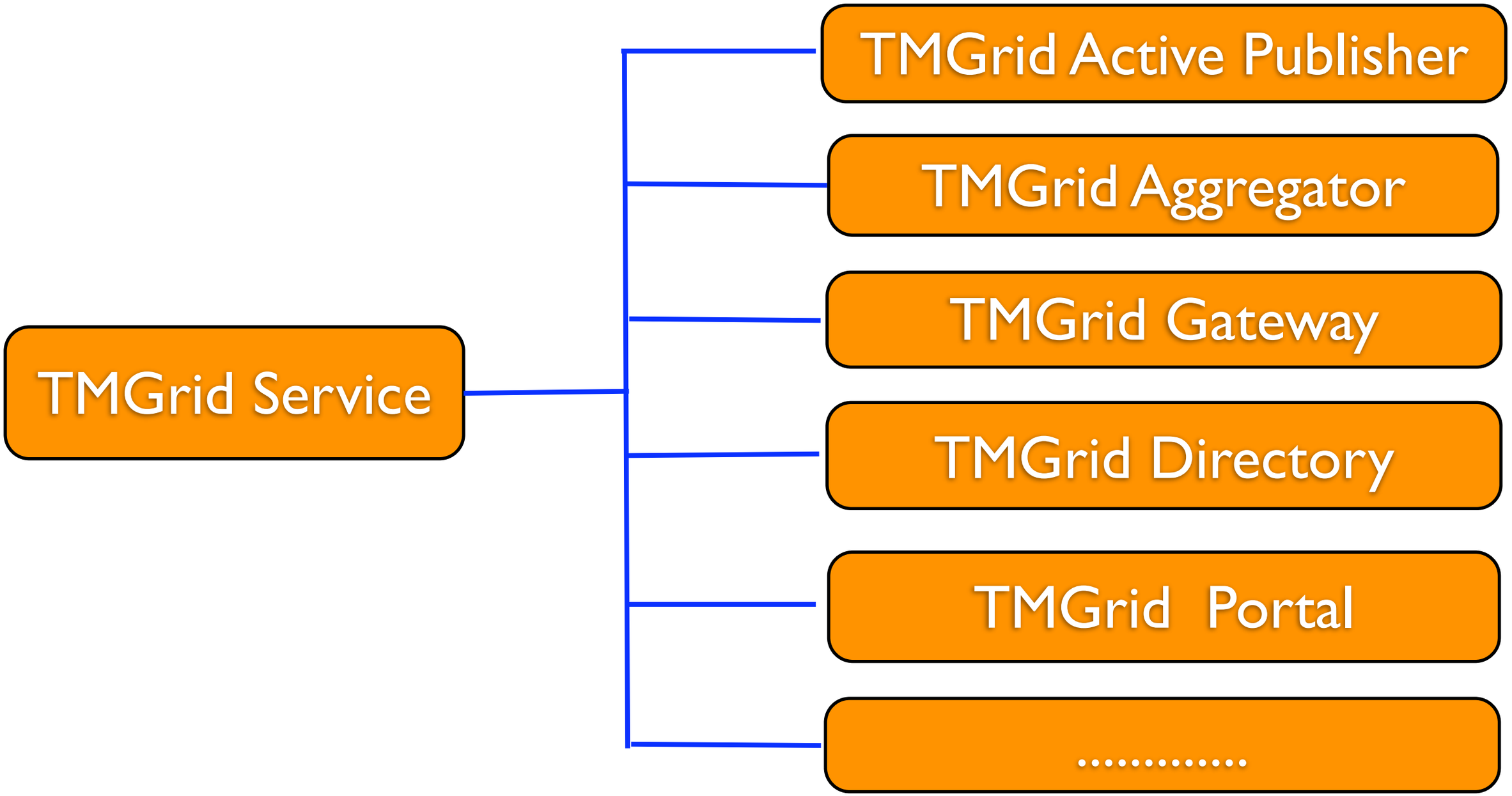
Directory: Assertion Set



Directory: Subjects



Directory: Services



TMGrid Directory

- allows to find PSIs of different topics
- allows to identify sources of information for specific individual topics or types (identified by PSIs)
- keeps track of "who knows what"
  - what kinds of instances are described in specific sources
  - what kinds of assertions
- allows to find topic map related services

TMGrid Portal

- Provides navigation, browsing and rendering facilities, user interaction layer
- Should be implemented as a system of sub-services
- Subject-centric, implements 360° view for subjects, also can include dashboards with summaries
- Search, faceted navigation should help findability
- Has topic map editor (general or based on custom forms, based on fragments)
- It should be sensitive to "new" information (like RSS/ Atom)

TMGrid Portal: integration points

- Integration with directory for smart search (topics, documents, assertion sets/topic maps)
- Aggregation is outsourced to Aggregators
- Integration with My Topics service. User can select some topics and organize them based on personal preferences
- Integration with Provenance service: Portal can present, help to navigate provenance information
- Support for offline component and roaming(user can take favorite topics on a laptop, sync later with the grid, users can use multiple computers)

TMGrid My Topics

- TMGrid must be able to scale to billions of topics. It is a serious problem for efficient browsing, search and navigation
- Users should be able to specify and manage set of topics of personal interest
- My Topics may support context sensitivity. Users can wear different "hats" during the day, so different topics can be in the focus of attention
- May use usage/activity statistics: unused topics become less and less active, topic activation can be propagated to related topics

TMGrid Provenance

- This service is responsible for managing data/info provenance
- Topic map - based inference engines, editors, gateways can use this service to record provenance information for different assertions
- It makes sense to have it as a separate service, optimized for recording, querying and presenting provenance information to the users
- It can be distributed

Why Grid?

- virtualization of data (kind of data grid)
- virtualization of services (kind of service grid)
- distributed inference (kind of computational grid)

- "what if" scenarios
- Example: create a topic map that is based on specific ontology,
- inherits some factual assertion sets,
- and have some additional assertions
  - example: person X is the same as person Y

TMGrid vs. Web 2.0

- general assertion model + explicit ontologies vs. custom domain specific XML
- universal merging vs. custom connections between different applications
- TMGrid should have gateways that allow to map domain specific vocabularies to topic map constructs (semantic mapping)
- But... it is better to start with domain ontology

TMGrid vs. RDF-based grids

- "right" level of basic semantic commitments
  - it works without additional schemas
  - instance, type-subtype in TM core (vs. RDFS, OWL, SKOS?,...)
  - automatic access to both sides of binary relationships
  - no need for inference/propogation/triggers
  - not only binary relationships
  - automatic support of symmetric relationships

TMGrid vs. RDF-based grids

- native representation of contextual information
  - Example: Apple has a product: 12' PowerBook G4 (scope: 2003-2006)
  - not only time - any contexts can be defined
  - ability to add/shift assertion context
    - example: add “source” into context and merge
  - can be implemented as a service
- clear concept of subject identity

TMGrid vs. contemporary desktops, personal information managers, RSS

- Subject-centric vs. application/document-centric
- TMGrid desktop: explicit representations of topics such as people, events, projects, tasks ....
- TMGrid desktop: subject-centric faceted navigation and search
- duality of subjects and resources
  - often we search for topics, not "documents"
- applications expose functions which can be used to work with different topics

Summary: TMGrid benefits

- Effective and user friendly infrastructure for producing, consuming and sharing information
- Efficient usage of distributed infrastructure
- More simple services, ability to leverage existing functionality, “plug-and-play”

The near future of TMGrid and Subject-centric blogging

- Progress in entity extraction and understanding the meaning of texts
- Online services for text analysis
  - Reuters
    - ClearForest
    - http://opencalais.mashery.com/
  - TextWise
    - http://www.semantichacker.com/

Generating assertions from text

- Text analysis/understanding tools can generate assertions.
- These assertions can be reviewed by humans
- Topic Maps provide simple standard based mechanism to record these assertions (at different stages of processing)
- Batch processing mode
  - Create text then run it through text analysis service
  - Provide ability to review results

Integration with authoring tools

- Generate assertions "on the fly" during text authoring
  - Side bar with identified subjects and assertions
  - "real time" interpretation
- Integration with personal and global PSI servers
  - we can define subjects that are important for us and "tune" interpretation
- One of the approaches:
  - Each PSI has a related "agent" that knows how to recognize presence of specific PSI in a text
  - Combination of content and model driven recognition

New generation of user interfaces: Multi-touch

- Jeff Han (Perceptive Pixel)
  - Link to video
- Apple iPhone and iPod Touch
  - http://www.apple.com/iphone/features/index.html#touch
- Microsoft Surface
  - http://www.microsoft.com/surface/index.html

Summary

- Ruby is a great dynamic programming language
  - perfect fit for working with Topic Maps
- RTM implements Topic Maps engine on top of ActiveRecord
- Ruby on Rails is a powerful framework for web applications
  - Can be used traditional "Rails" way with XTM export
  - Can be used with RTM as Topic Maps engine on top of ActiveRecord
- Subject-centric blogging is part of Topic Maps Grid - distributed Topic Maps infrastructure
  - It is just a beginning