

# Topic Maps Grid:

information sharing  
infrastructure based on Topic  
Maps, Grid and Service Oriented  
Architecture

Dmitry Bogachev  
Extreme Markup 2006  
August 9th

# Agenda

- XTM - Topic Maps interchange format
- TMGrid introduction
- Main TMGrid components
- TMGrid vs. RDF, Contemporary Desktop
- Summary and next steps

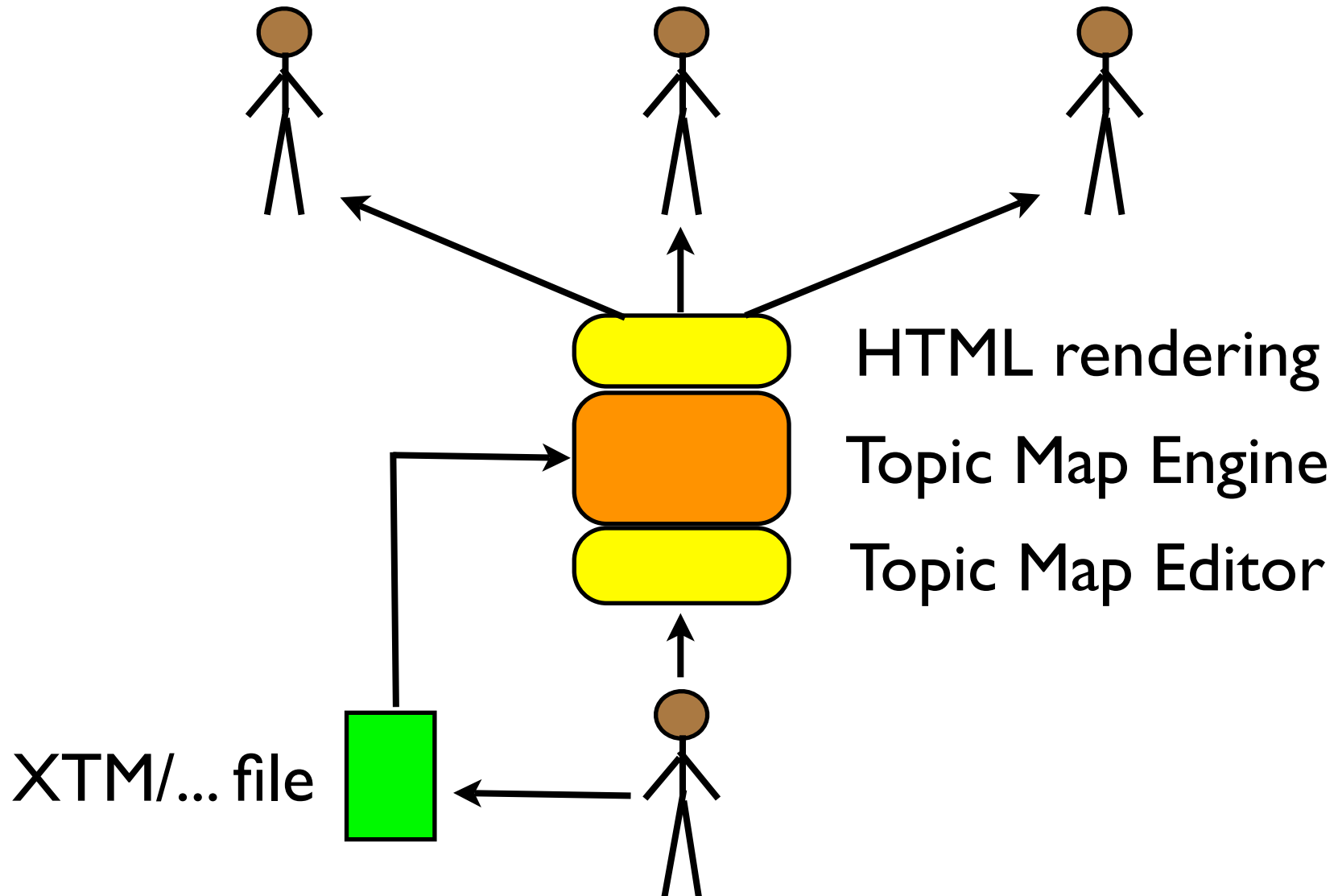
# XTM - Topic Maps interchange format

- allows to record and exchange pieces of information
- assumes minimum semantic commitments
  - enough to support reasonable information exchange
- has rich and clear model for subject identification
- has basic support for re-using existing pieces of knowledge (merging)
- has basic support for knowledge contexts (scope)

# Topic Maps in real life

- Use case: user maintains a database on home computer about some favorite topic and she wants to make it available online as a topic map...
- How can she do this?
- How can other people find this topic map?
- How will other people know about changes in this topic map?
- How can they combine this topic map with other topic maps?
- How they can view existing topic maps...

# Typical Implementation



# RSS analogy

- RSS standard(s) - good basis
- But also:
  - RSS desktop aggregators
  - browser plug-ins
  - online aggregation services
  - RSS auto generation by content management systems, blogs, wikis
  - specialized search and directory services for RSS feeds

# XTM - conclusion

- Additional pieces of infrastructure are required to support knowledge sharing based on Topic Maps paradigm.

# Proposed approaches

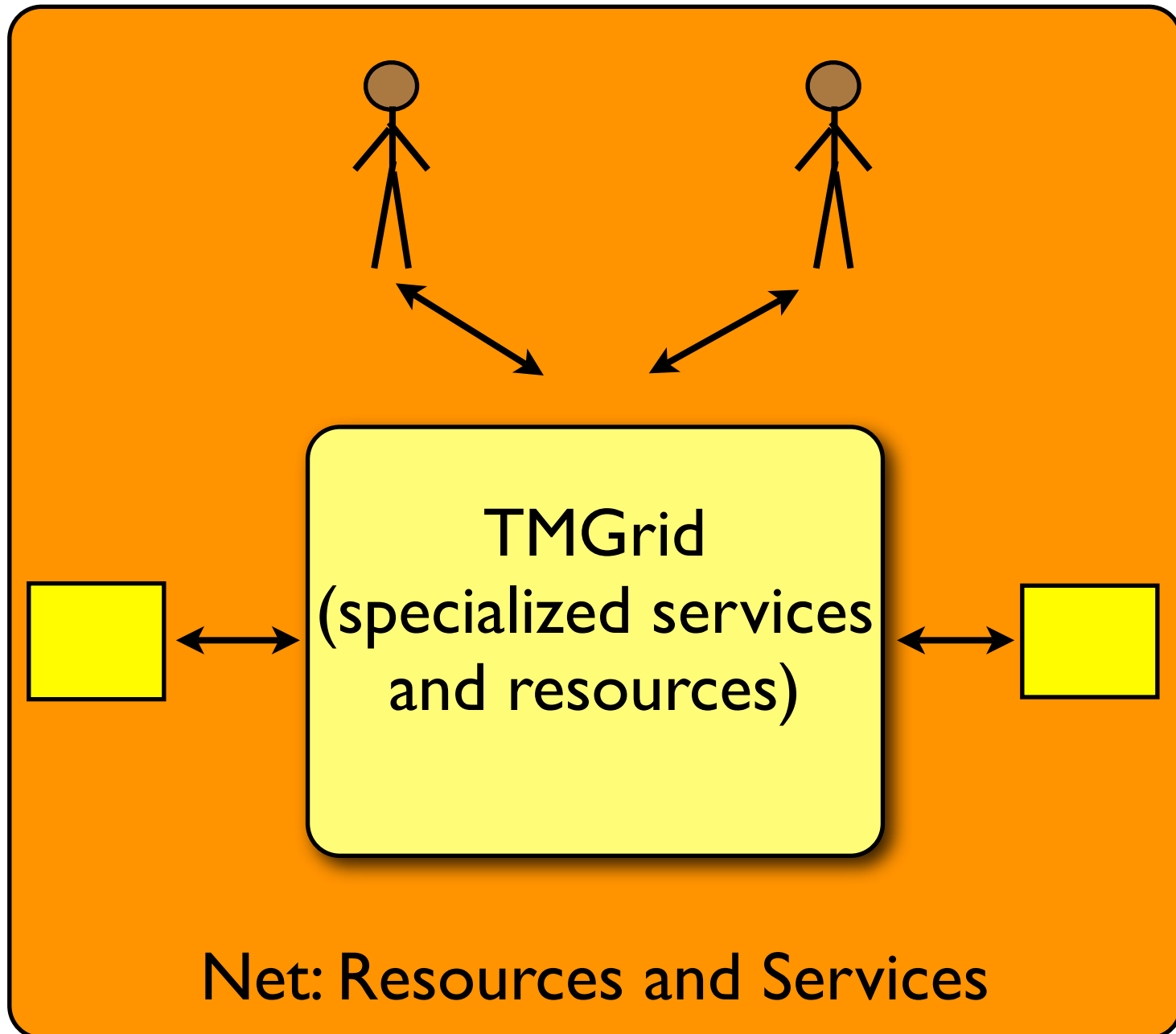
- TMRAP - Topic Maps Remote Access Protocol
- TMIP, A RESTful Topic Maps Interaction Protocol
- TMShare - Topic Map Fragment Exchange In a Peer-To-Peer Application
- Jabber integration
- Basic Web services
- Integration with WS-Notifications and WS-Eventing
- Mapping with relational databases
- Mapping with RDF
- ◆ It is a list of possible solution blocks
- ◆ We need a comprehensive architecture to connect all pieces together



# Our Proposal - Topic Maps Grid (TMGrid)

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

# TMGrid services

- gateways
- transformers
- validators
- aggregators
- directory
- provenance
- my topics
- portal

# Fundamental activities on TMGrid

- Creating topics with Public 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 RSS feed with topic map fragments for updates
- Aggregator can process RSS 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 gateways

- Provide access to existing on the internet assertion sets
- Existing assertion sets can have different format
- Topic Maps are rich enough to represent factual information encoded in different assertion sets
- “Knowledge level” gateways

# TMGrid gateway types

- Relational database
- RSS/Atom
- Wiki
- RDF
- Domain specific XML (including Web 2.0)
- OpenCyc (OpenCyc 1.0 is available!)
- WS-Resource
- WS-Transfer/Enumeration/Eventing
- 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
- 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

# Wiki gateway

- Wiki can be a very important source for contributing information to TMGrid.
- OpenCyc: using users to translate Wikipedia articles to CycL
- Interesting ideas about extending Wiki links to include RDF like assertions
- Semantic MediaWiki -
  - [http://wiki.ontoworld.org/index.php/Semantic\\_MediaWiki](http://wiki.ontoworld.org/index.php/Semantic_MediaWiki)
- TMWiki
  - <http://www.topic-maps.org/projects:tmwiki>



# Resources, services and Topic Maps

- Data can be available in specialized formats (relational data sets, time series, video streams etc.)
- We do not need to represent every artifact in topic maps, we can make resources available on TMGrid by providing Subject Locators and metadata for these resources
  - Example: `stock-price-data-set(ticker,from-date,to-date,interval)`
- Resource Subject Locator can be passed to different services on the grid which can do something meaningful with this resource
- Services also can produce new resources which become available on the grid for consumption

# What we represent on TMGrid

- Ontologies (lightweight approach)
- Reference Data
  - More or less stable objects, properties and relationships
  - Example: Apple has a product: 12' PowerBook? G4 (scope: 2003-2006)
  - Objects are evolving in time, evolution is covered by events
  - We can use “scope” for representing time sensitive information

# What else?

- Events
  - something that happens at a given place and time
  - include participants, sub-events
  - casual, time relationships between events
  - events often are described in news articles
- And
  - Change assertions (comparisons between states in time)
  - ...

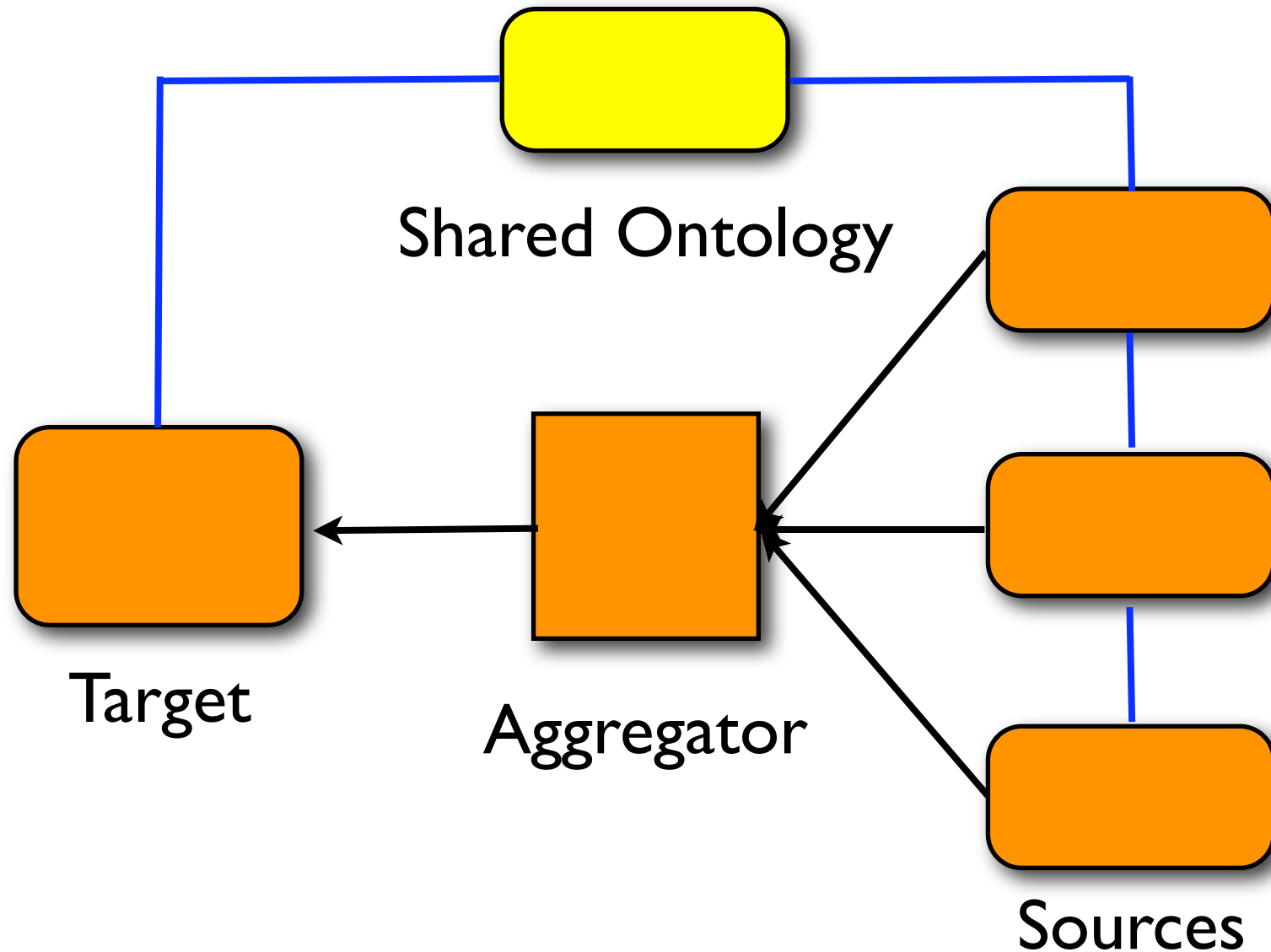
# TMGrid Aggregator

- Aggregates information from multiple sources
  - XTM/LTM/... files , RSS/TM
  - active topic map publishers
    - response/request, publish/subscribe
  - different gateways
  - other aggregators
- Often filtering and transformations are required
- Propagates updates between topic maps
- Keeps track of original source of information

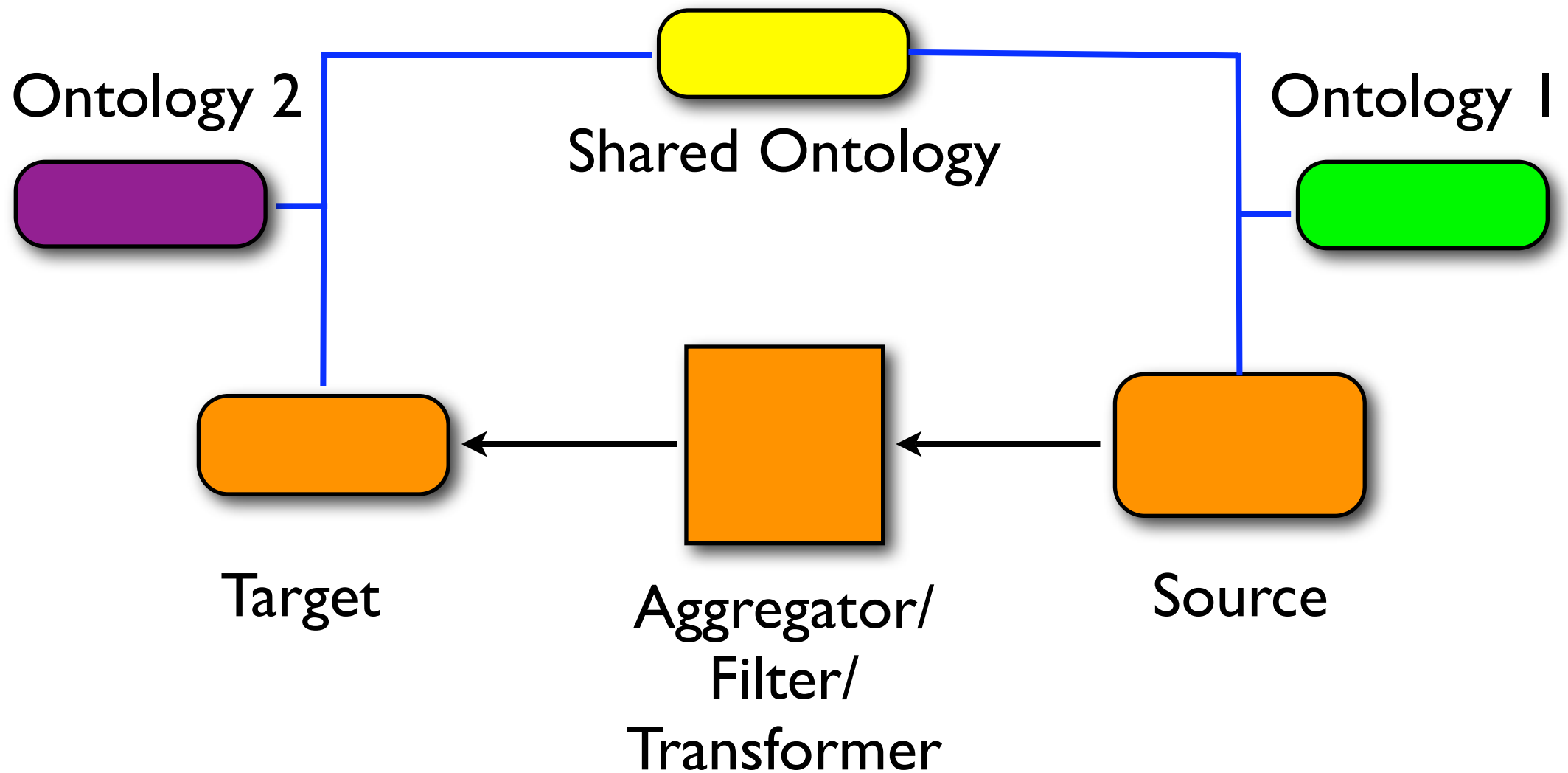
# TMGrid Aggregator: basics

- 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

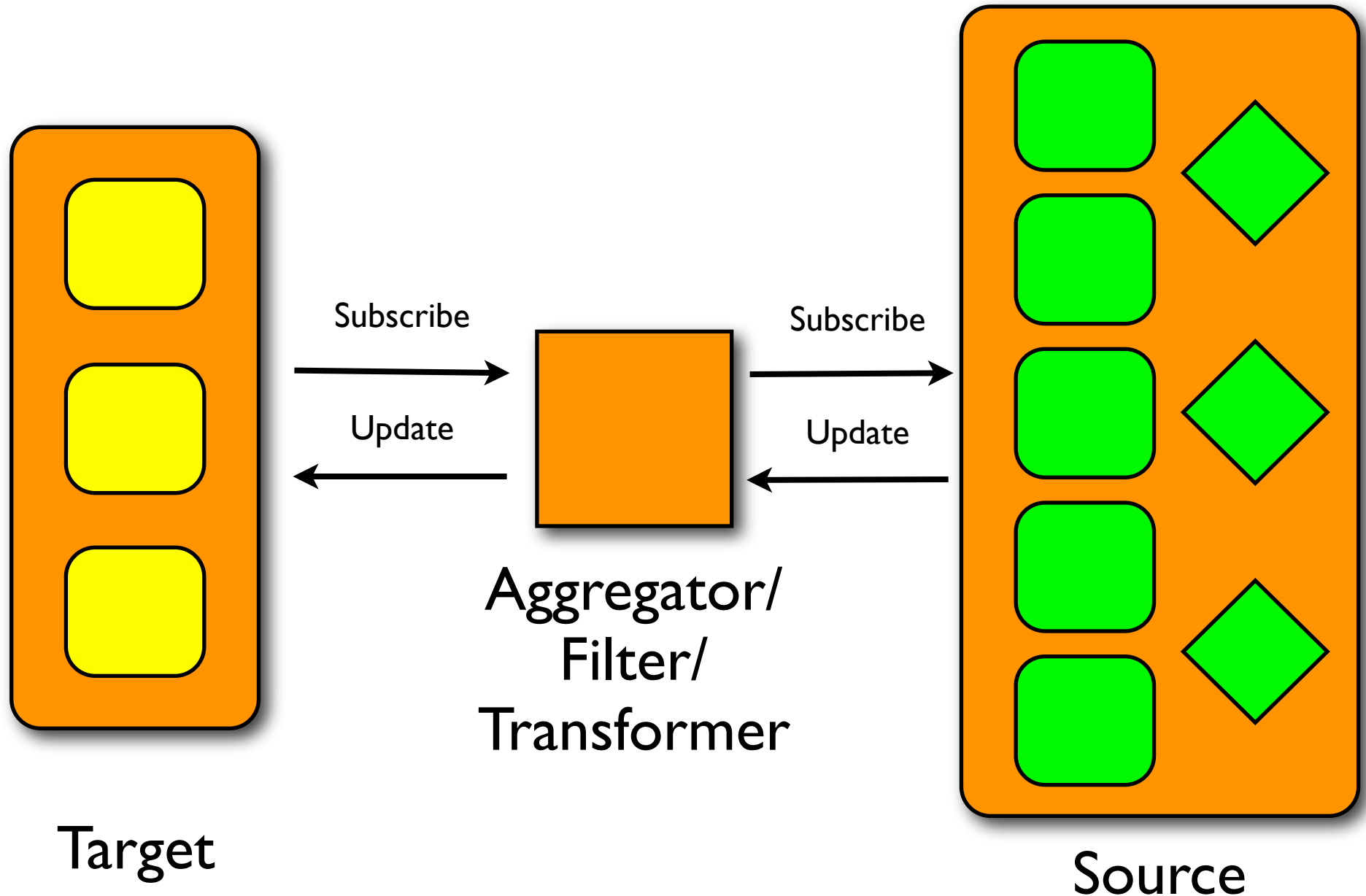
# Simple “copy and merge” aggregation



# Filtering and transformation

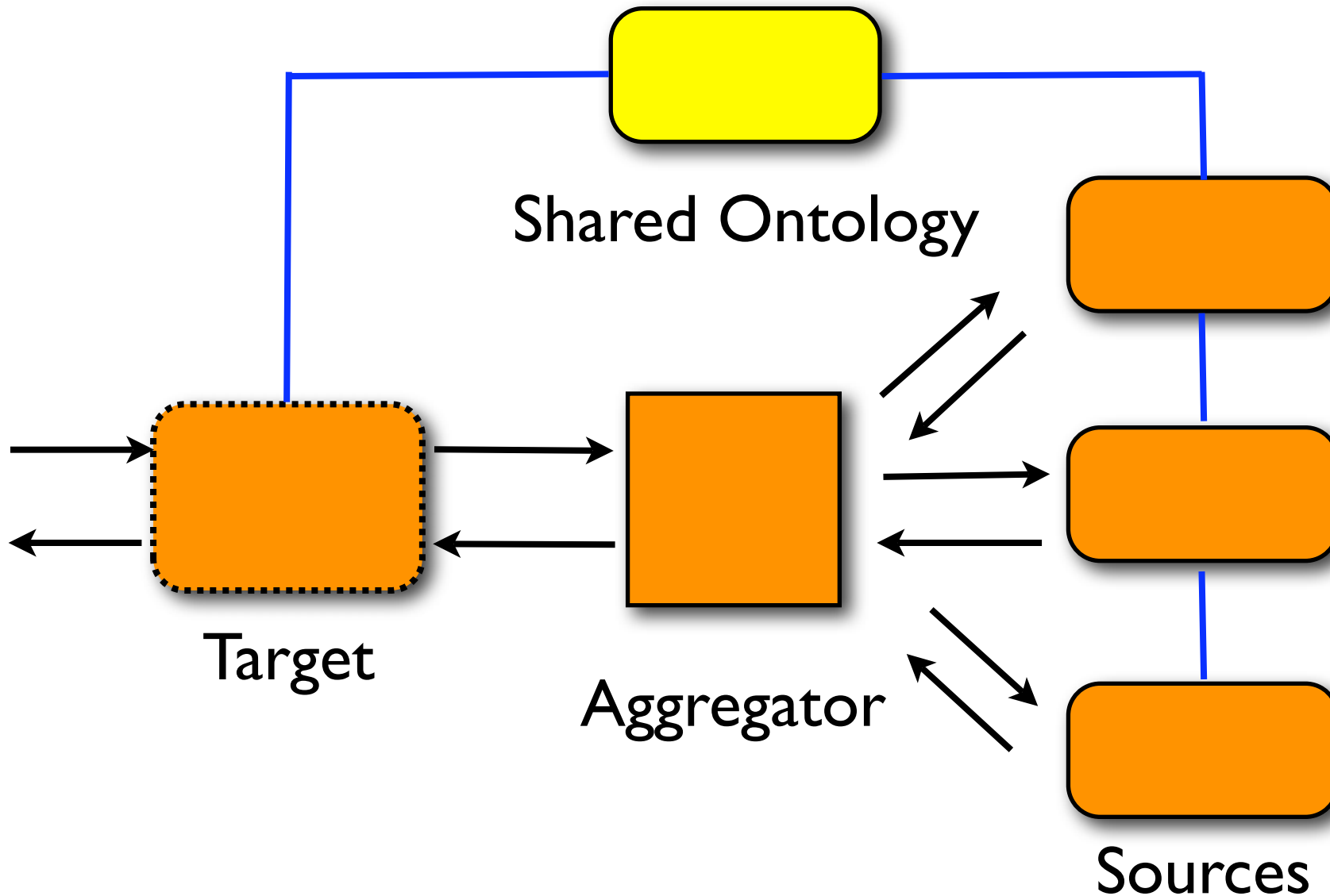


# Fragment-based filtering and transformation





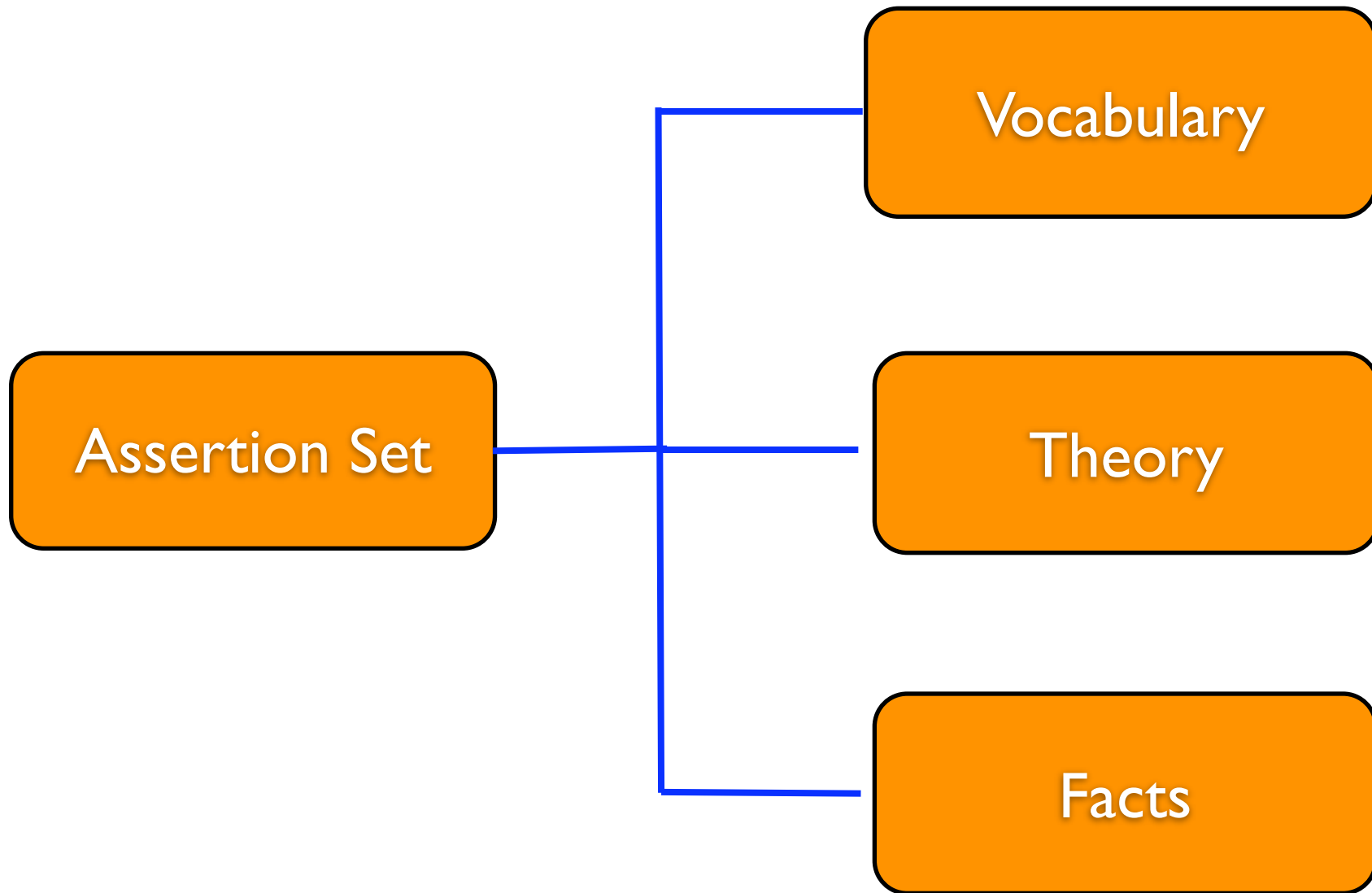
# “On the fly” aggregation



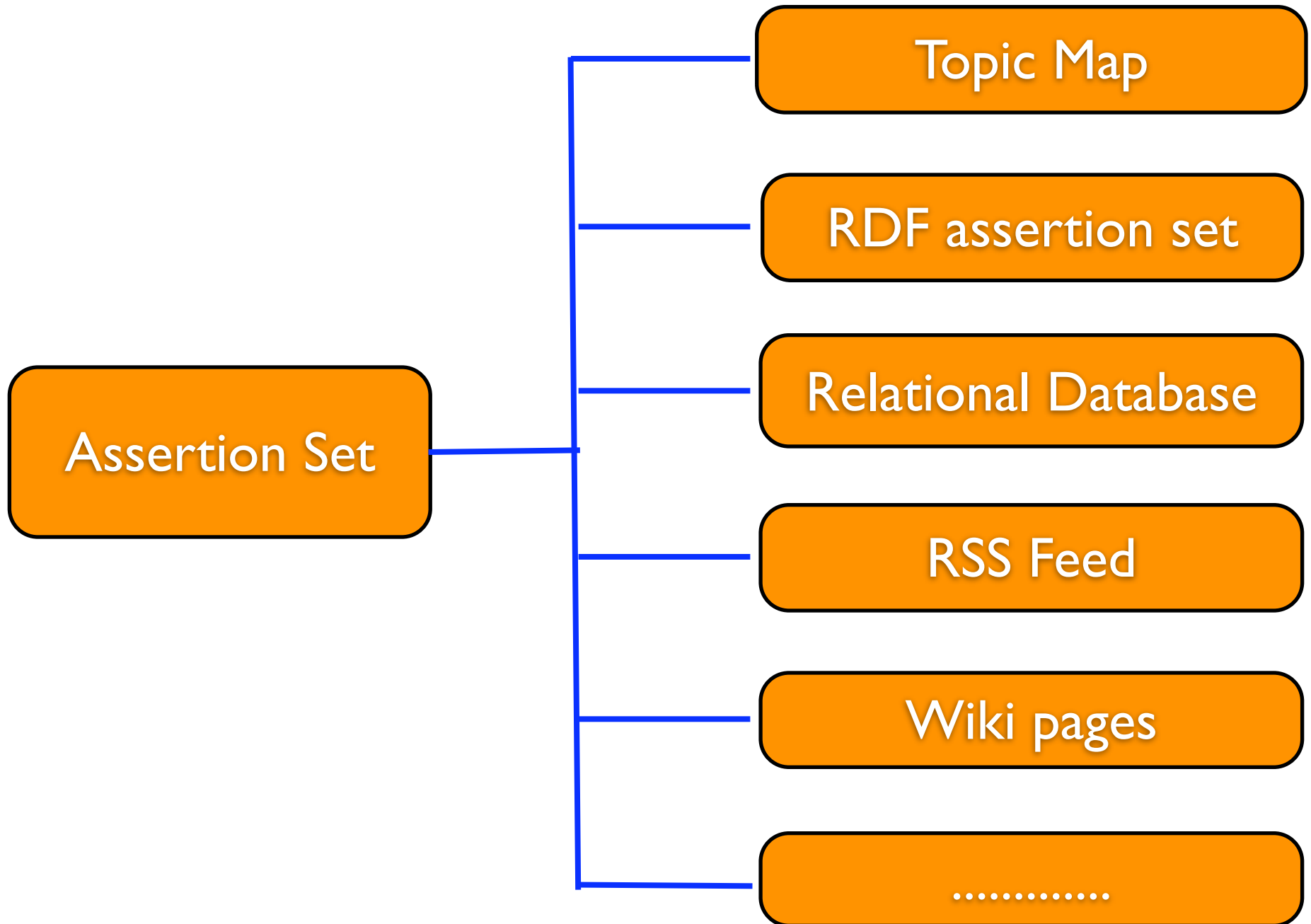
# TMGrid Directory

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

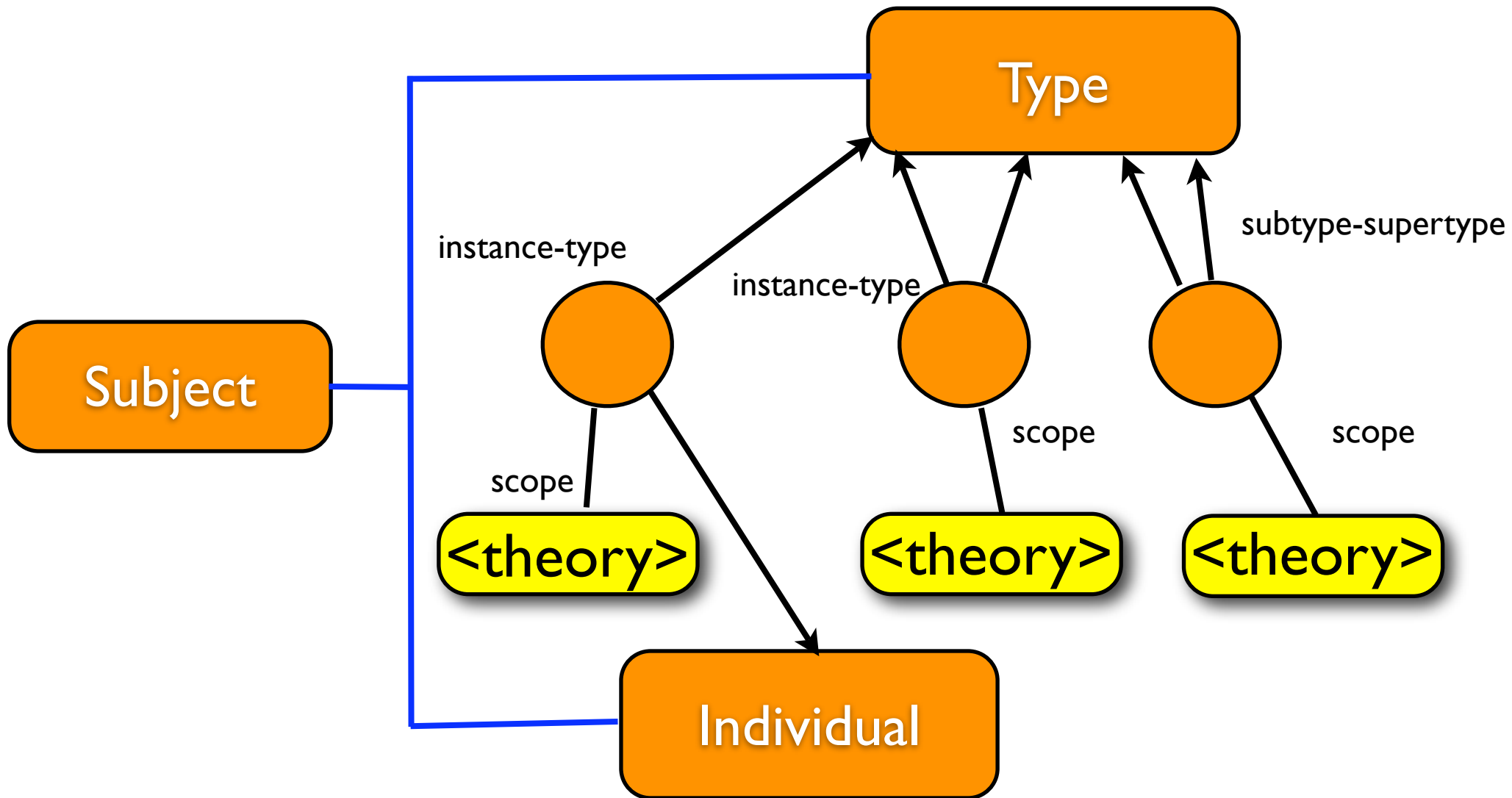
# Directory: Assertion Set



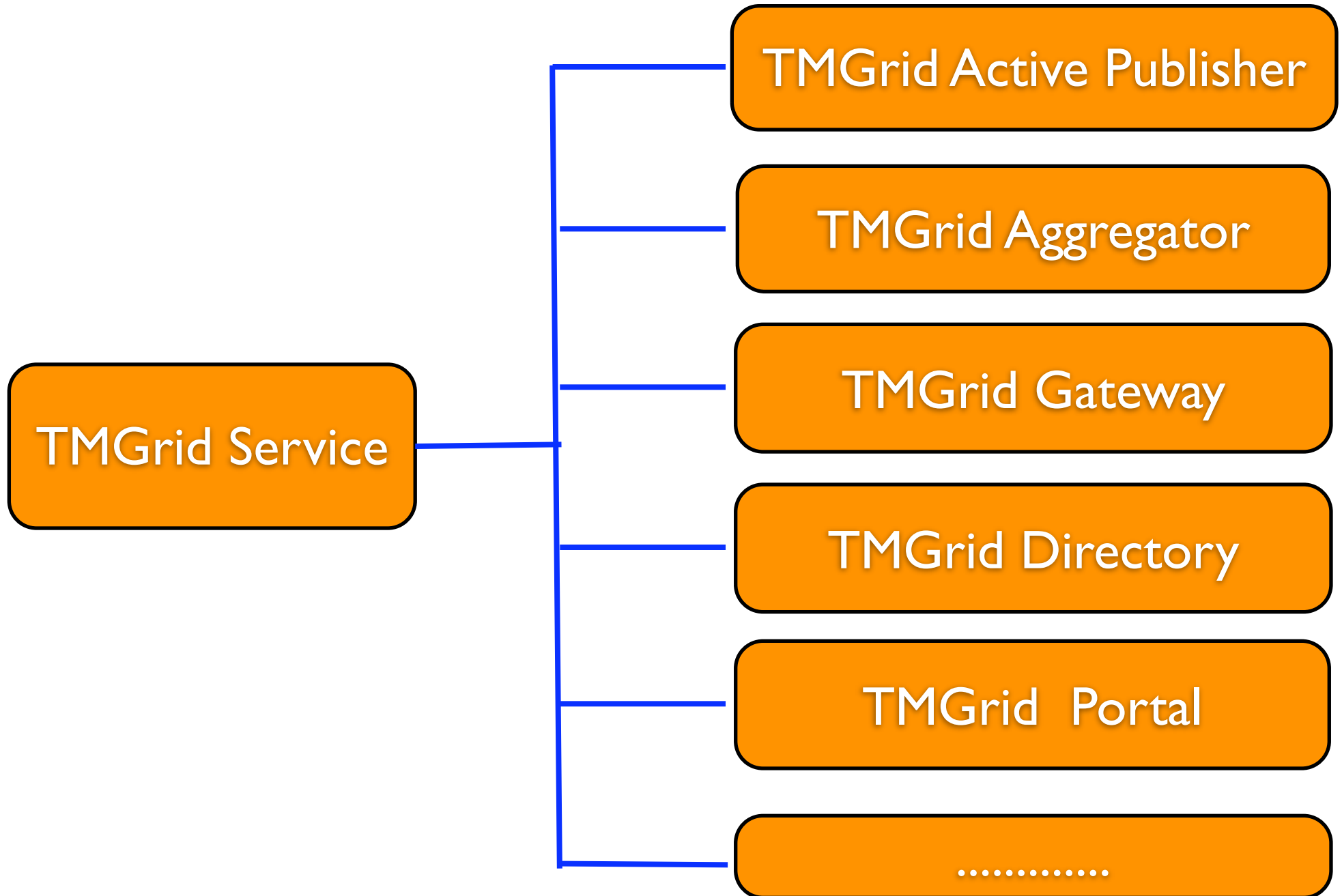
# Directory: Assertion Set



# 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, treemaps, heatmaps - should help findability and efficient presentation
- Has topic map editor (general or based on custom forms, based on fragments)
- It should be sensitive to "new" information (like RSS)



# 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/propagation/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”

# TMGrid: next steps

- Community
- Connecting several existing servers - active publishers
- Publishing reusable services (transformers, gateways, validators...)
- Experimenting with directory, aggregators, portals, personalization