Scalability in TripCom: scalability and the API

Lyndon J B Nixon

TripCom WP3
Vienna Meeting, 2 October 2007









Background

- In D3.1 we did not consider Web scalability, only having an expressive enough co-ordination model
- Trade-offs between scalability and expressiveness
  - Some aspects of the API can be dropped fully
  - Retain as „maximum“ possible, the „minimum“ necessary for the TripCom scenarios
  - Define the minimum necessary for a Web scale Triple Space as the ‚core API‘
  - The rest has been labelled ‚extension API‘

D3.1 API

out (Graph, Space)	rda (Template)	rd (Template)	rdg (Template)
ina (Template)	in (Template)	ing (Template)	subscribe (Template, Space)
create (Space)	destroy (Space)	create Transaction (type)	unsubscribe (Subscription)
get Transaction (ID)	begin Transaction (ID)	commit Transaction (ID)	rollback Transaction (ID)

The maximum necessary

The WP3 meeting in Vienna produced a new API

http://www.tripcom.org/internal/document/707

as the maximum necessary in Triple Space

(NB. Space as parameter, transactions)

out(Triple, Space)
out(Set<Triple>, Space)
rd(Template, Space, Timeout)
rdmultiple(Template, Space, Timeout)
in(Template, Space, Timeout)
inmultiple(Template, Space, Timeout)
subscribe(Template, Space, Callback)
unsubscribe(Subscription)
create(path, Space)
destroy(Space)

Querying

- The impact of rd on scalability will depend on the complexity of the queries allowed
  - Simplest template would be triple patterns without RDFS inference
  - WP3 is benchmarking OWLIM on SPARQL query support with full RDFS inference
  - Other template types will be specified based on
    - Restricting the query language or ontology to gain lower complexity in reasoning for a significant subset of queries in Triple Space
    - Extending the query language or ontology while minimizing the cost in increased complexity in reasoning to support reasoning beyond RDF and SPARQL

Sixth Framework Programme Priority 2
Information Society Technologies (IST)
Specific Targeted Research Project

The minimum necessary

- We defined a „core API“ as the minimum functionality of the Web-scale Triple Space

out (Triple t, URL space) returns void

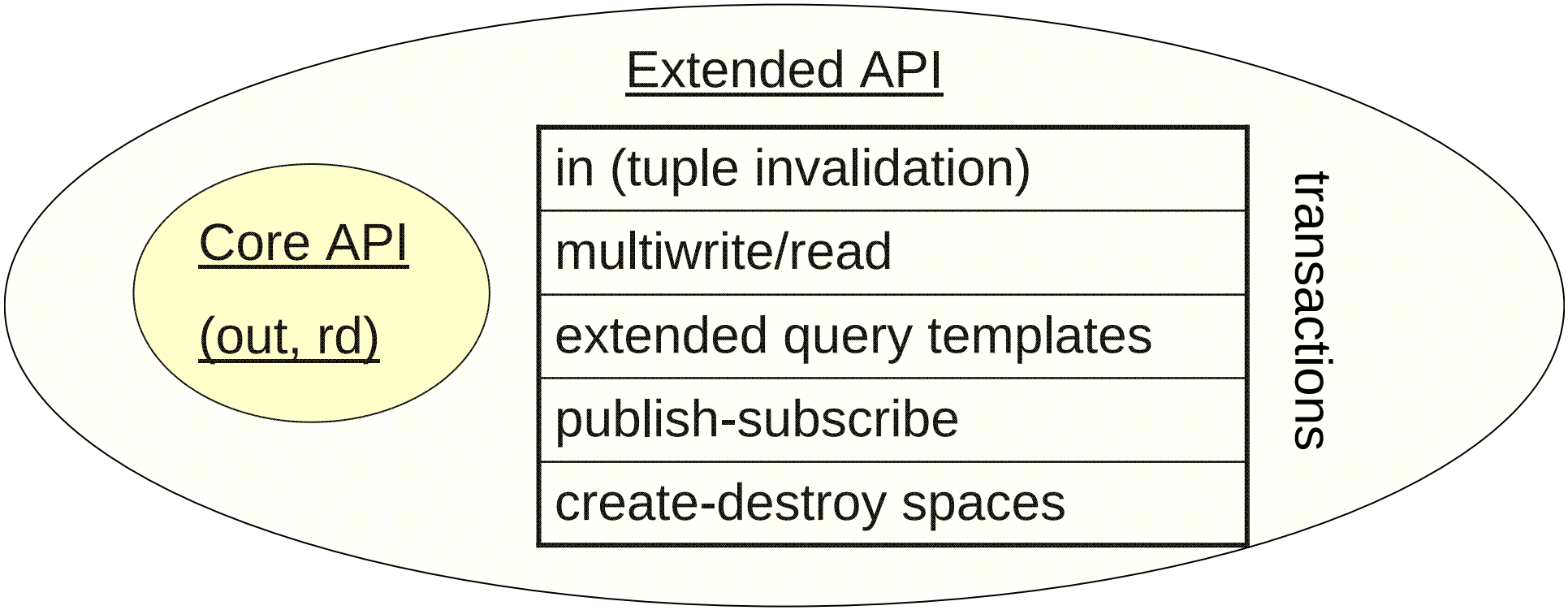
Atomically writes a single triple into the space. The operation makes no guarantee if and when the triple will be available in the space (unordered semantics). The client is immediately free to perform further activities. The client has to provide a resolvable URL which identifies a space.

rd (SingleTemplate t, URL space, Time timeout) returns Set<Triple> s

Returns one match of the given template which is a single triple pattern. The match may be a set of triples, e.g. Concise Bounded Description. The operation makes no guarantee as to when the match would be returned to the client. A timeout is provided to give a temporal bound for returning a match. If no match has been found by the timeout period, an empty set is returned. This does not make any statement regarding the existence of a match in the space. No timeout can be specified by providing a null value to the timeout parameter.

Extensions

- Part of the remaining work in TripCom must be
  - Determining the extensions to the core API which add necessary expressivity to the coordination model
  - **measuring the impact on scalability**



API configurations

Web scale	eHealth (WP8b)	EAI (WP8a)
out rd Single Template	multiwrite/read publish-subscribe SPARQL++ Template	create-destroy transactions SPARQL Template

in(multiple)

Sixth Framework Programme Priority 2
Information Society Technologies (IST)
Specific Targeted Research Project

End of Document