A proposal for a Triple Space kernel architecture

Robert Tolksdorf, Elena Paslaru, **Lyndon J B Nixon**
FU Berlin
27.06.06

Outline

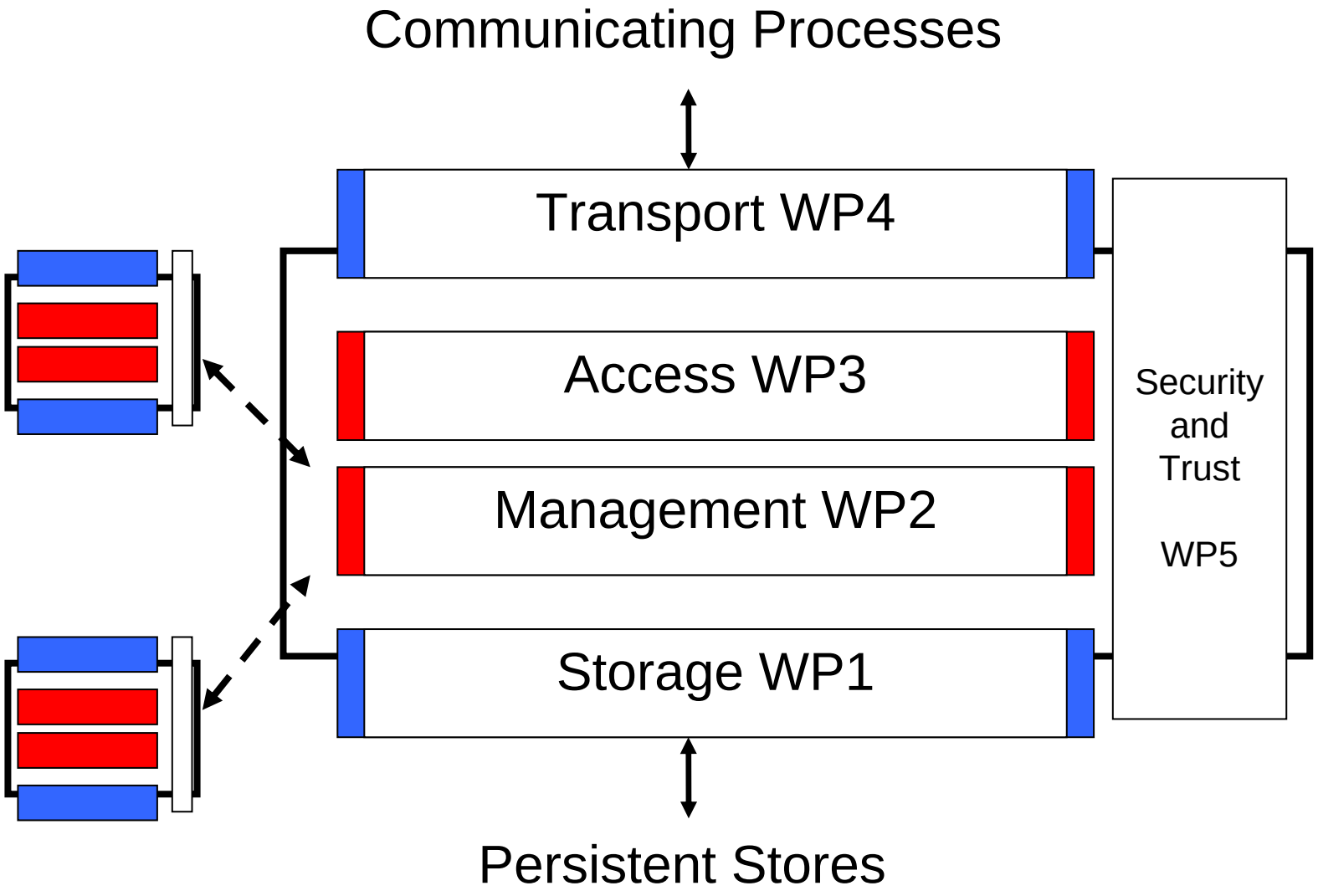
- Main objectives
- Conceptual architecture of a TripleSpace kernel
- Relationship to other WPs
- Action points

Main objectives

- Form a common understanding of internal structure of triplespace kernel
- Agree on the distribution of responsibilities among and within work packages
- Begin a project-wide discussion on the overall direction of development

Conceptual architecture

Triple Space layers

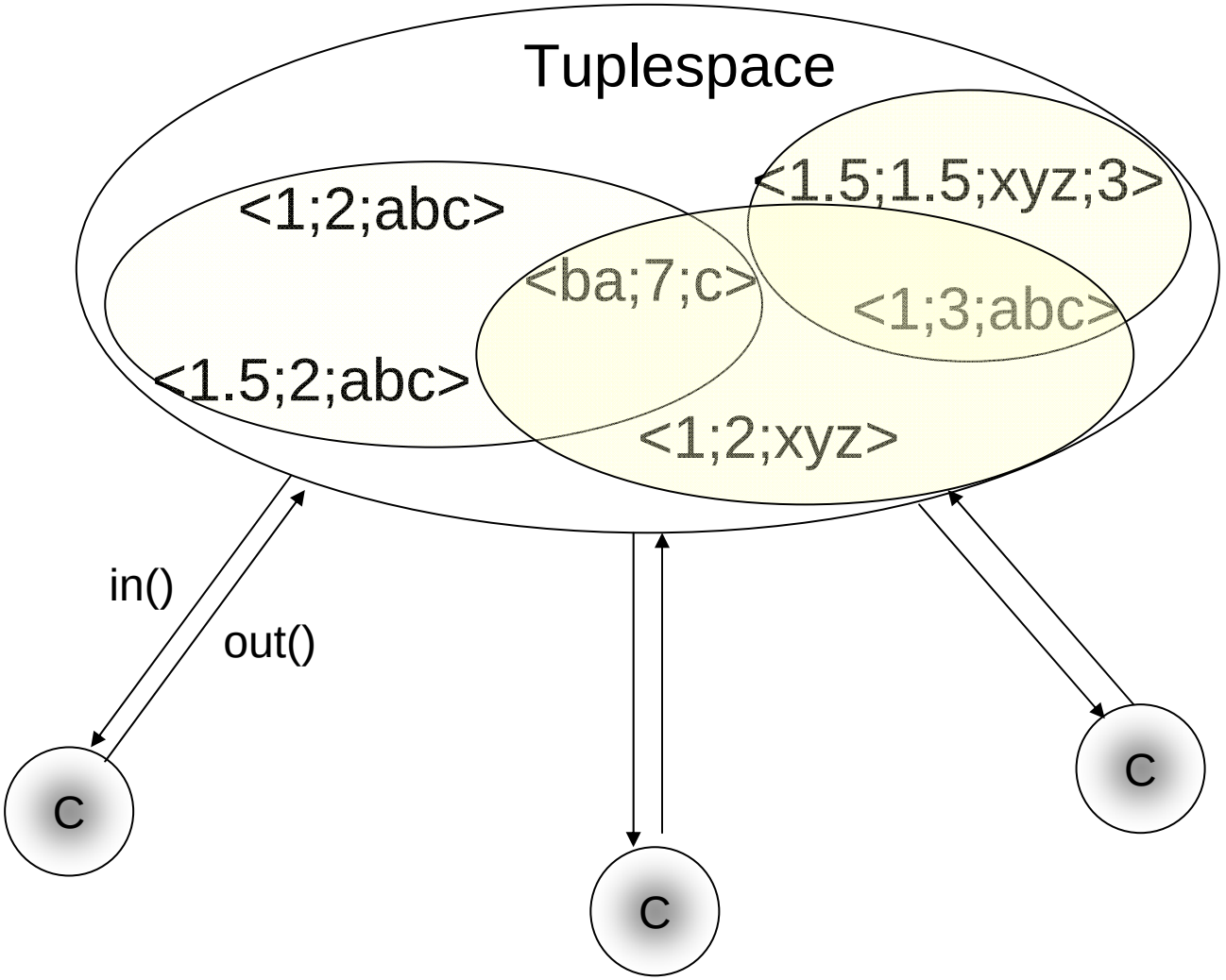


Sixth Framework Programme Priority 2
Information Society Technologies (IST)
Specific Targeted Research Project

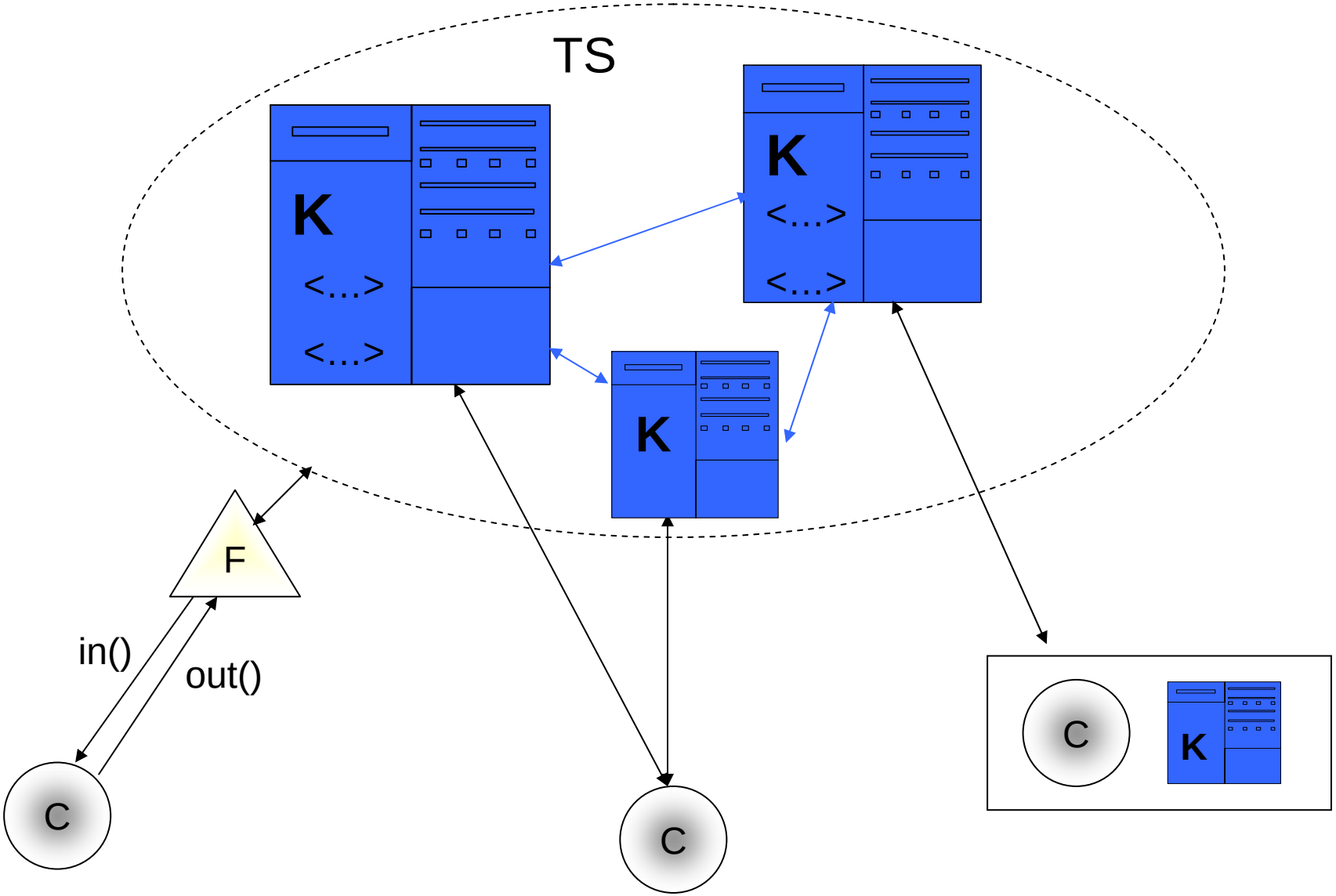
Access and management layers

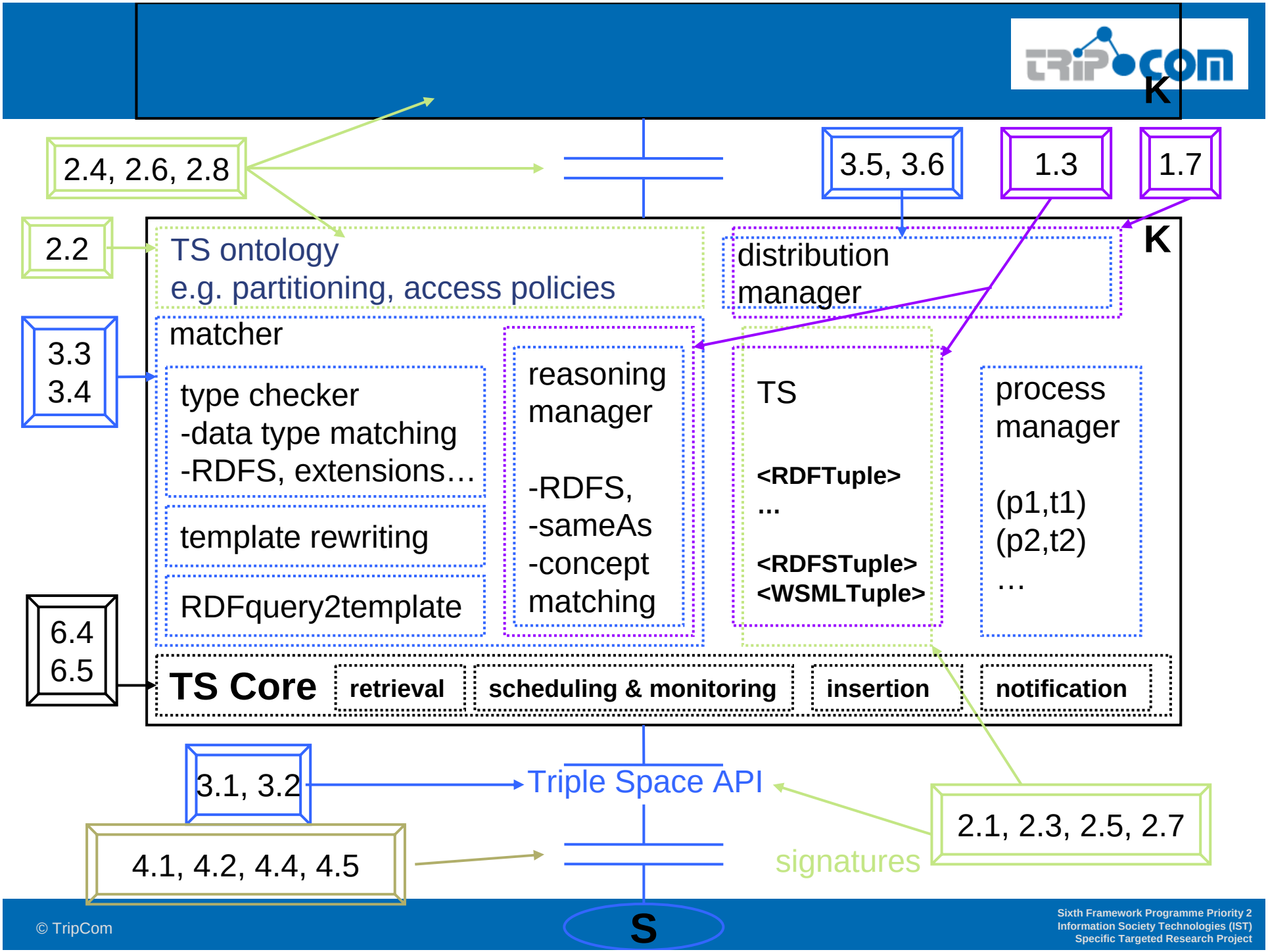
Component	Role	Functionality
Tuple/Tuplespace Management	To maintain the shared tuplespace and manage the tuples which exist in it	▪ tuple format ▪ tuple type ▪ tuple organisation within a single process ▪ tuple distribution across processes ▪ access policies
Tuplespace Access	To support a standardized set of coordination primitives which enable clients to perform operations on the tuplespace	▪ coordination primitives ▪ parameter handling ▪ input/output of tuples from/to clients

Logical view of Triple Space



Machine view of Triple Space

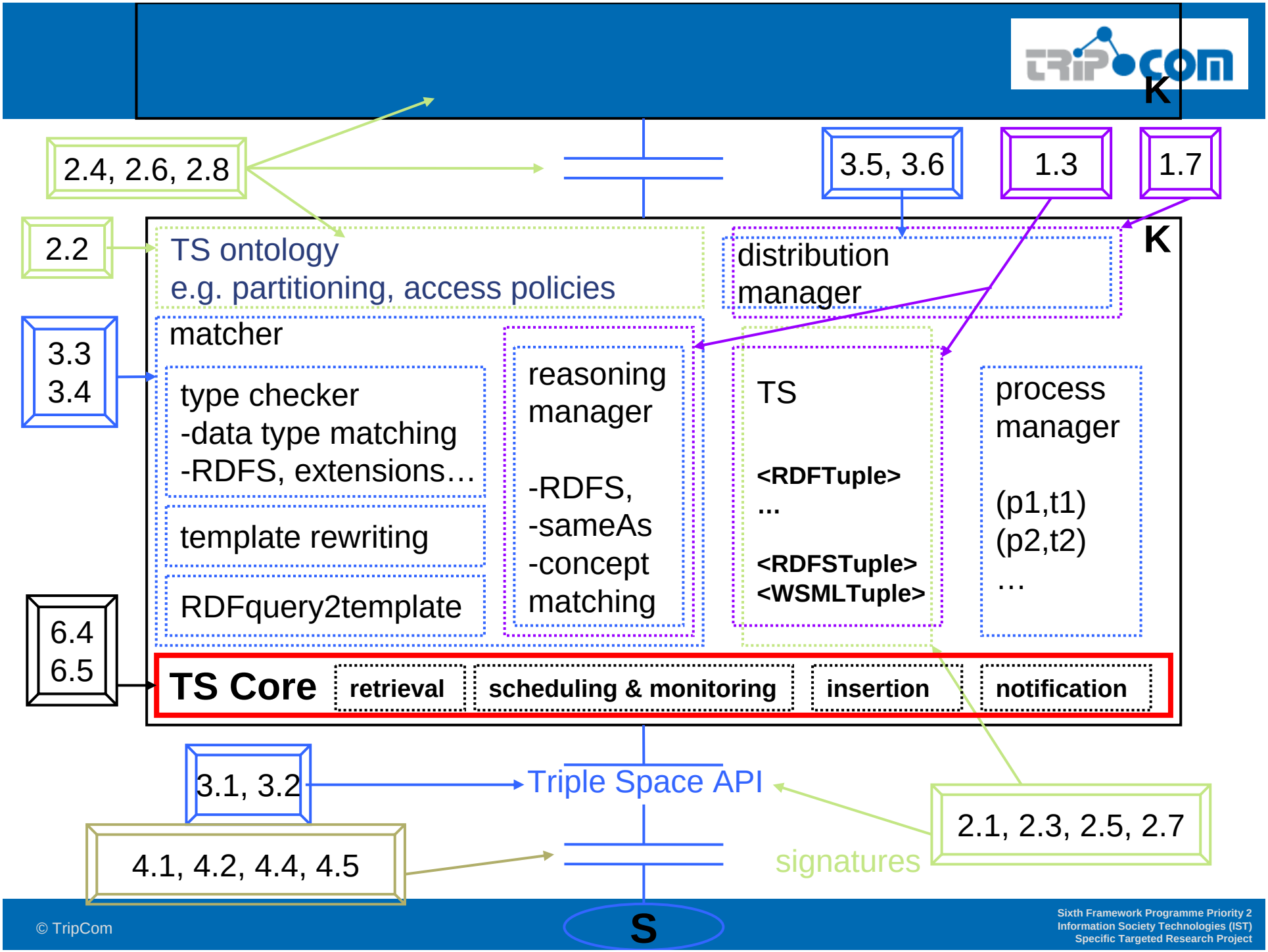



2.4, 2.6, 2.8
3.5, 3.6
1.3
1.7
K
2.2
TS ontology
e.g. partitioning, access policies
distribution manager
K
3.3
3.4
matcher
type checker
-data type matching
-RDFS, extensions…
template rewriting
RDFquery2template
reasoning manager
-RDFS,
-sameAs
-concept matching
TS
<RDFTuple>
...
<RDFSTuple>
<WSMLTuple>
process manager
(p1,t1)
(p2,t2)
…
6.4
6.5
TS Core
retrieval
scheduling & monitoring
insertion
notification
3.1, 3.2
Triple Space API
2.1, 2.3, 2.5, 2.7
signatures
4.1, 4.2, 4.4, 4.5
S

Triple Space API

- TripleSpace requires a coordination model and a language which defines the model
- We plan to take the tuplespace paradigm and the associated Linda coordination language for these purposes

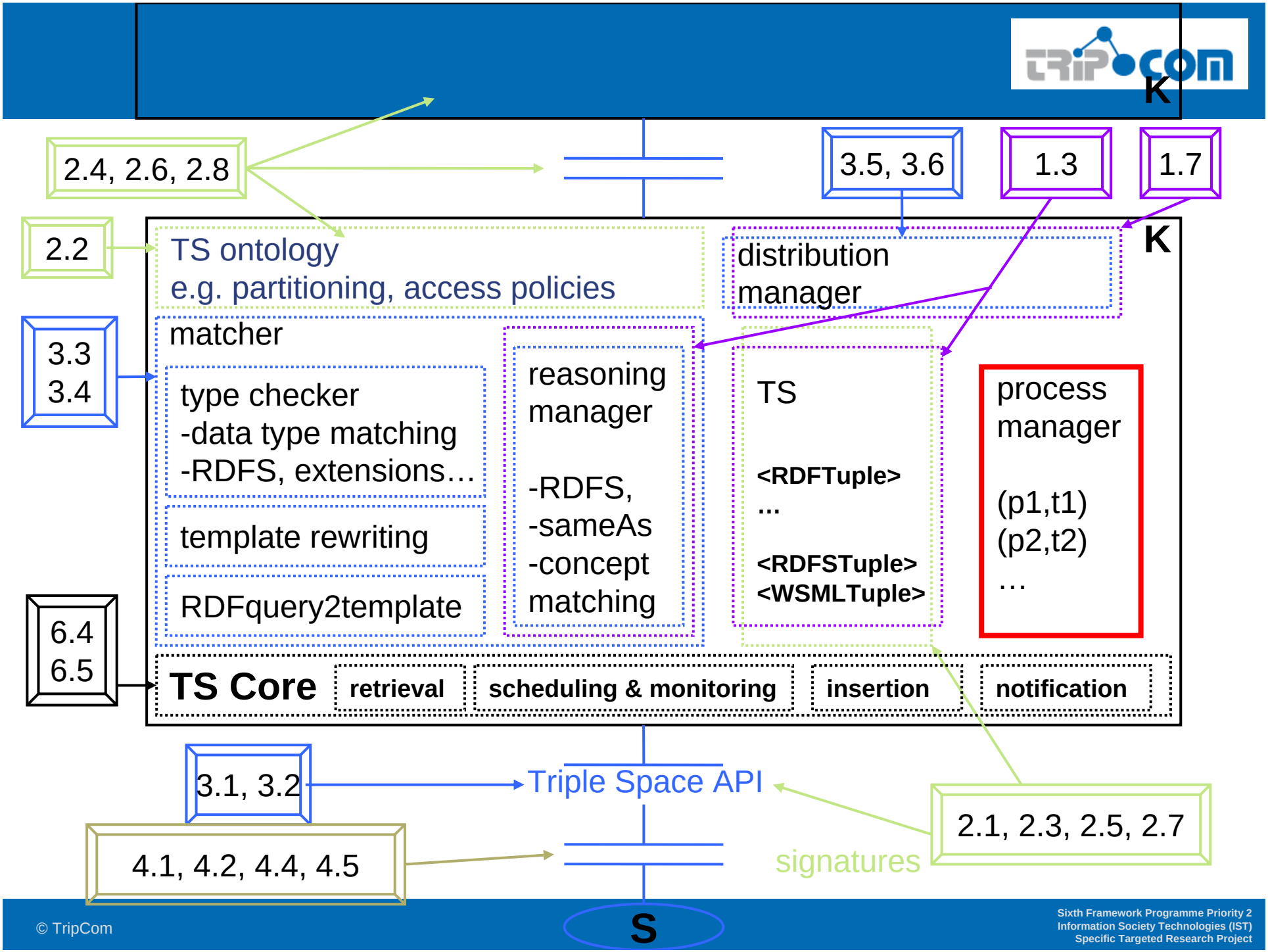
- Which extensions and revisions are required for TripCom?
- How will it map onto Web Service interaction?


2.4, 2.6, 2.8
3.5, 3.6
1.3
1.7
K
2.2
TS ontology
e.g. partitioning, access policies
distribution manager
K
3.3
3.4
matcher
type checker
-data type matching
-RDFS, extensions…
template rewriting
RDFquery2template
reasoning manager
-RDFS,
-sameAs
-concept matching
TS
<RDFTuple>
...
<RDFSTuple>
<WSMLTuple>
process manager
(p1,t1)
(p2,t2)
…
6.4
6.5
TS Core
retrieval
scheduling & monitoring
insertion
notification
3.1, 3.2
Triple Space API
4.1, 4.2, 4.4, 4.5
signatures
2.1, 2.3, 2.5, 2.7
S

TS Core

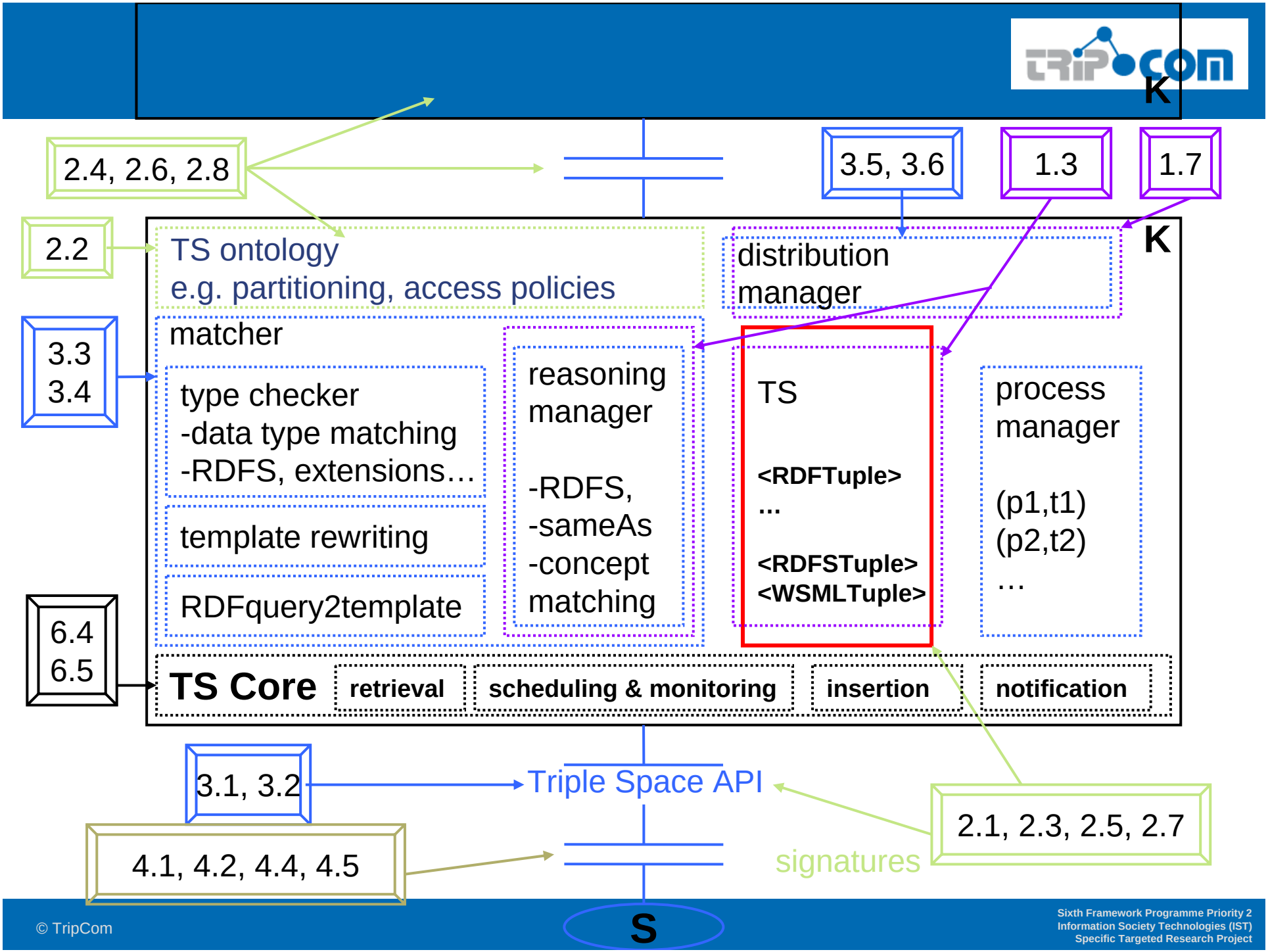
- Handles incoming operations
- Hands off tasks to appropriate components in the TripleSpace kernel

- Core functionality:
  - Insertion
  - Retrieval
  - Scheduling and monitoring
  - Notification
  - …


2.4, 2.6, 2.8
3.5, 3.6
1.3
1.7
K
2.2
TS ontology
e.g. partitioning, access policies
distribution manager
K
3.3
3.4
matcher
type checker
-data type matching
-RDFS, extensions…
template rewriting
RDFquery2template
reasoning manager
-RDFS,
-sameAs
-concept matching
TS
<RDFTuple>
...
<RDFSTuple>
<WSMLTuple>
process manager
(p1,t1)
(p2,t2)
…
6.4
6.5
TS Core
retrieval
scheduling & monitoring
insertion
notification
3.1, 3.2
Triple Space API
2.1, 2.3, 2.5, 2.7
signatures
4.1, 4.2, 4.4, 4.5
S

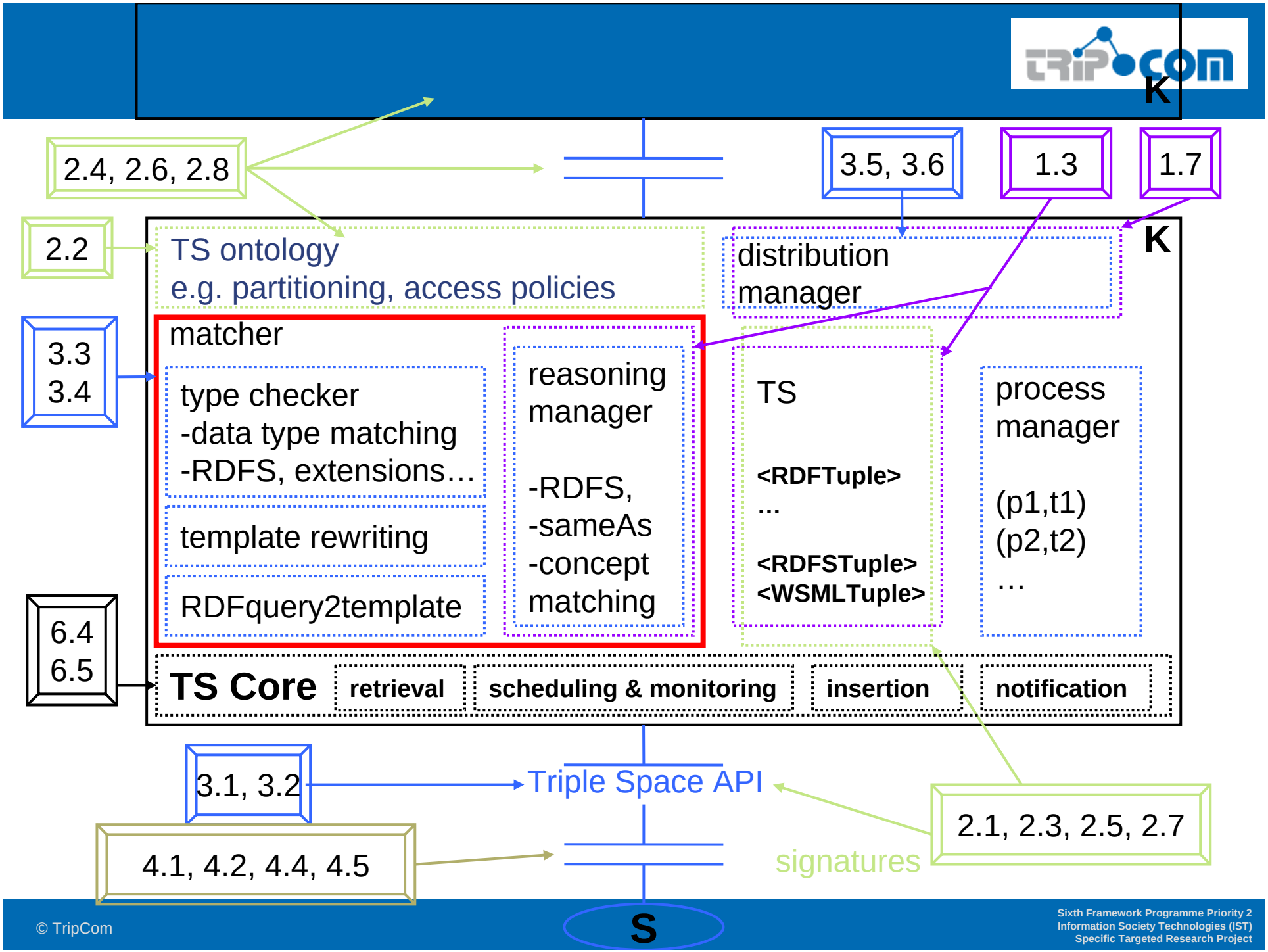
Process manager

- Maintains tables for outstanding retrieval operations (blocking)
- Timeouts
- Notification mechanism


K
2.4, 2.6, 2.8
3.5, 3.6
1.3
1.7
2.2
TS ontology
e.g. partitioning, access policies
distribution manager
K
3.3
3.4
matcher
type checker
-data type matching
-RDFS, extensions…
template rewriting
RDFquery2template
reasoning manager
-RDFS,
-sameAs
-concept matching
TS
<RDFTuple>
...
<RDFSTuple>
<WSMLTuple>
process manager
(p1,t1)
(p2,t2)
…
6.4
6.5
TS Core
retrieval
scheduling & monitoring
insertion
notification
3.1, 3.2
Triple Space API
4.1, 4.2, 4.4, 4.5
signatures
2.1, 2.3, 2.5, 2.7
S

Triple Space (TS)

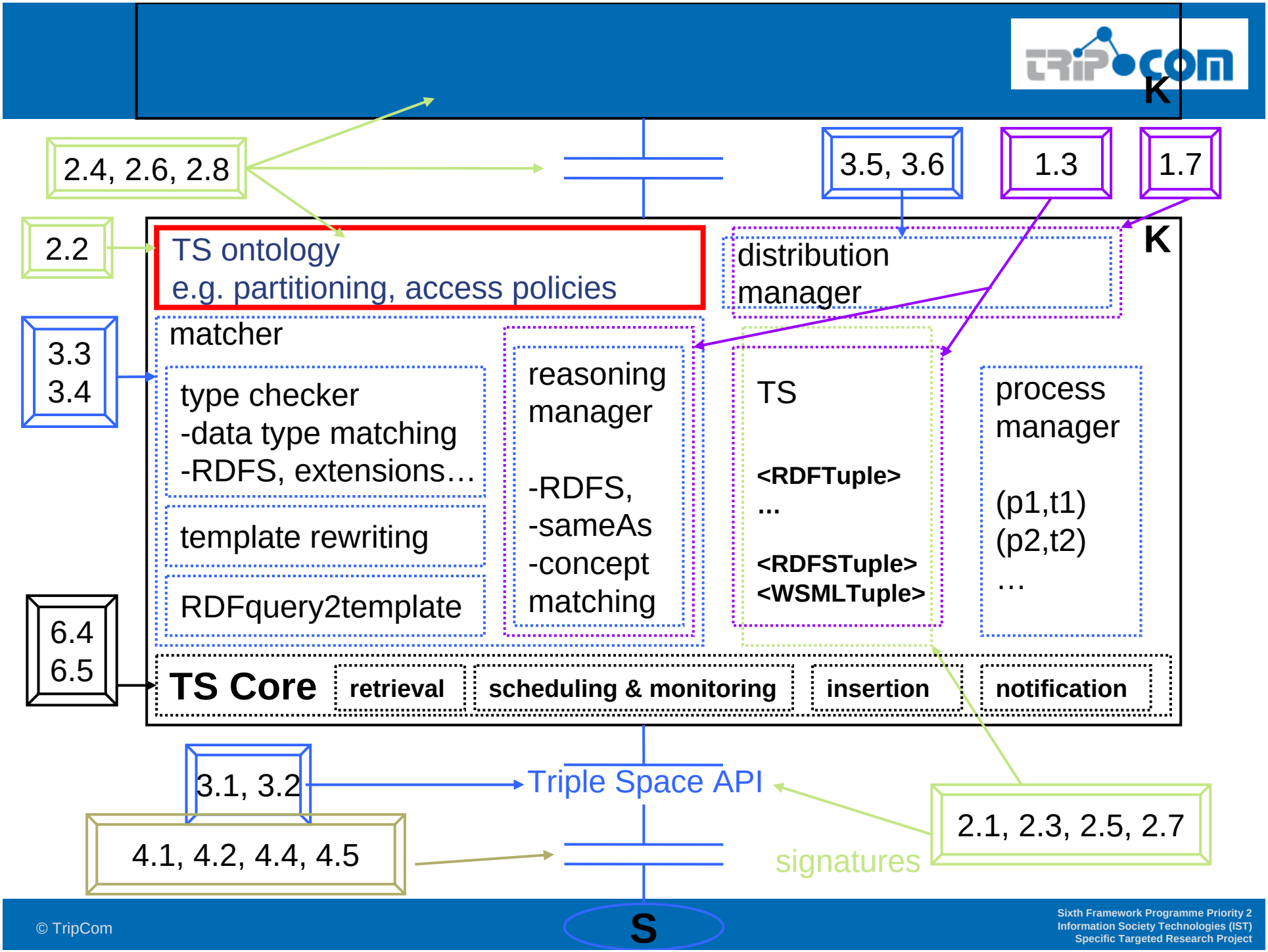
- Virtual shared space containing tuples
  - Data represented as tuples
  - Service represented as (reactive) tuples
  - Mapping tuples
  - Administrative data according to the tuplespace ontology
- The space can be further structured according to
  - Access policies
  - Tuple types
  - Agent-defined views
  - …


K
2.4, 2.6, 2.8
3.5, 3.6
1.3
1.7
2.2
TS ontology
e.g. partitioning, access policies
distribution manager
K
3.3
3.4
matcher
type checker
-data type matching
-RDFS, extensions…
template rewriting
RDFquery2template
reasoning manager
-RDFS,
-sameAs
-concept matching
TS
<RDFTuple>
...
<RDFSTuple>
<WSMLTuple>
process manager
(p1,t1)
(p2,t2)
…
6.4
6.5
TS Core
retrieval
scheduling & monitoring
insertion
notification
3.1, 3.2
Triple Space API
2.1, 2.3, 2.5, 2.7
signatures
4.1, 4.2, 4.4, 4.5
S

Template matcher

- Tuple matching must be extended to handle RDF and semantic queries
- Reasoning on application data can be handled in the storage layer
- Reasoning on administrative data is handled by the tuplespace kernel on the basis of the TS ontology

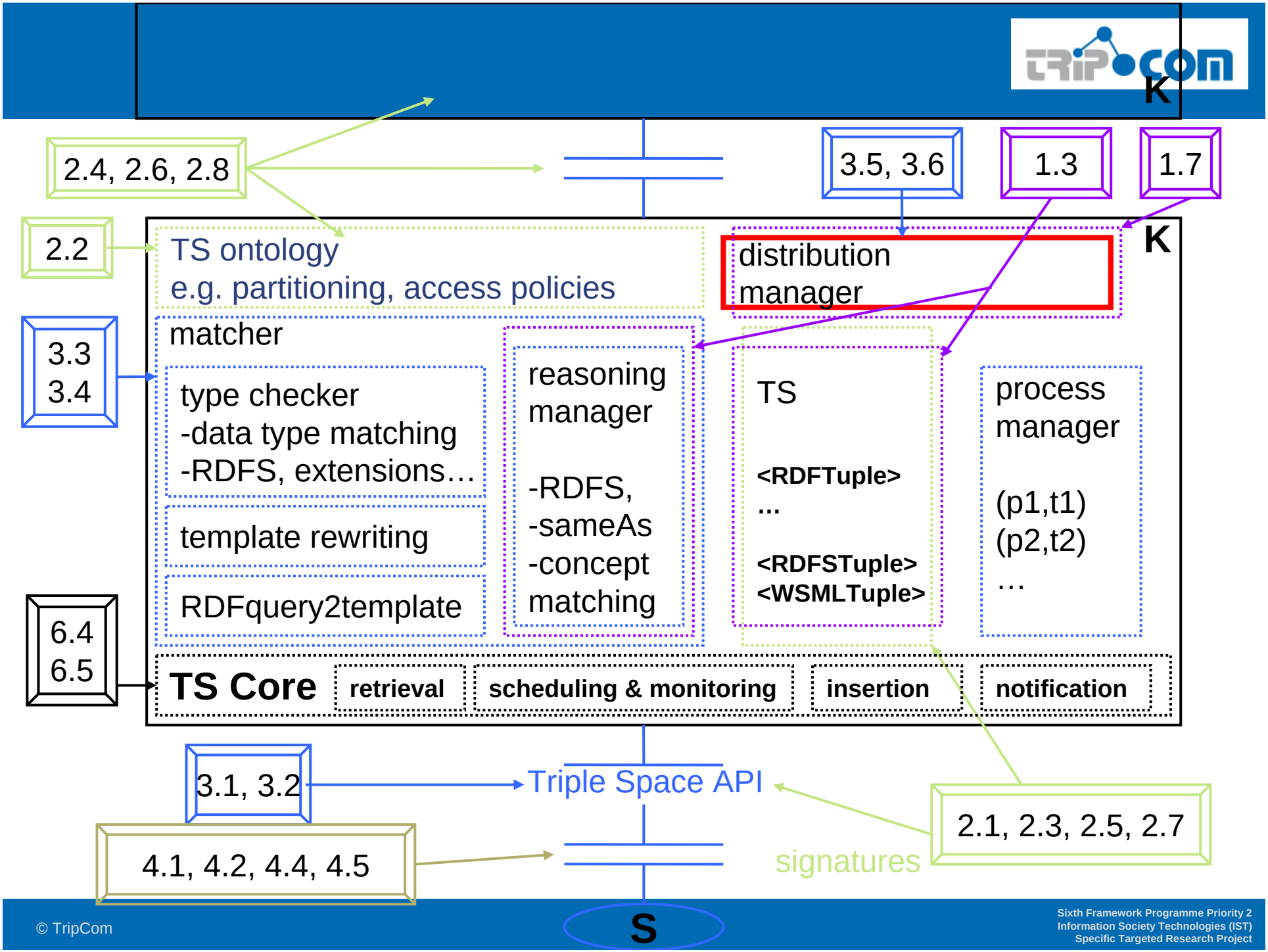
- TS API may use templates, while the storage layer will require (SPARQL) queries
- TS API could also provide support for SPARQL queries
  - When is an intermediary mapping to template necessary?


2.4, 2.6, 2.8
3.5, 3.6
1.3
1.7
K
2.2
TS ontology
e.g. partitioning, access policies
distribution manager
K
matcher
3.3
3.4
type checker
-data type matching
-RDFS, extensions…
template rewriting
RDFquery2template
reasoning manager
-RDFS,
-sameAs
-concept matching
TS
<RDFTuple>
...
<RDFSTuple>
<WSMLTuple>
process manager
(p1,t1)
(p2,t2)
…
6.4
6.5
TS Core
retrieval
scheduling & monitoring
insertion
notification
3.1, 3.2
Triple Space API
4.1, 4.2, 4.4, 4.5
signatures
2.1, 2.3, 2.5, 2.7
S

TS Ontology

- Means to structure administrative data about the tuplespace
- Provides a basis for optimization

- Administrative data could include:
  - Types of tuples and tuplespaces
  - Agents and access policies
  - Operation history, logging
  - Distribution strategy


2.4, 2.6, 2.8
3.5, 3.6
1.3
1.7
K
2.2
TS ontology
e.g. partitioning, access policies
distribution manager
K
3.3
3.4
matcher
type checker
-data type matching
-RDFS, extensions…
template rewriting
RDFquery2template
reasoning manager
-RDFS,
-sameAs
-concept matching
TS
<RDFTuple>
...
<RDFSTuple>
<WSMLTuple>
process manager
(p1,t1)
(p2,t2)
…
6.4
6.5
TS Core
retrieval
scheduling & monitoring
insertion
notification
3.1, 3.2
Triple Space API
2.1, 2.3, 2.5, 2.7
signatures
4.1, 4.2, 4.4, 4.5
S

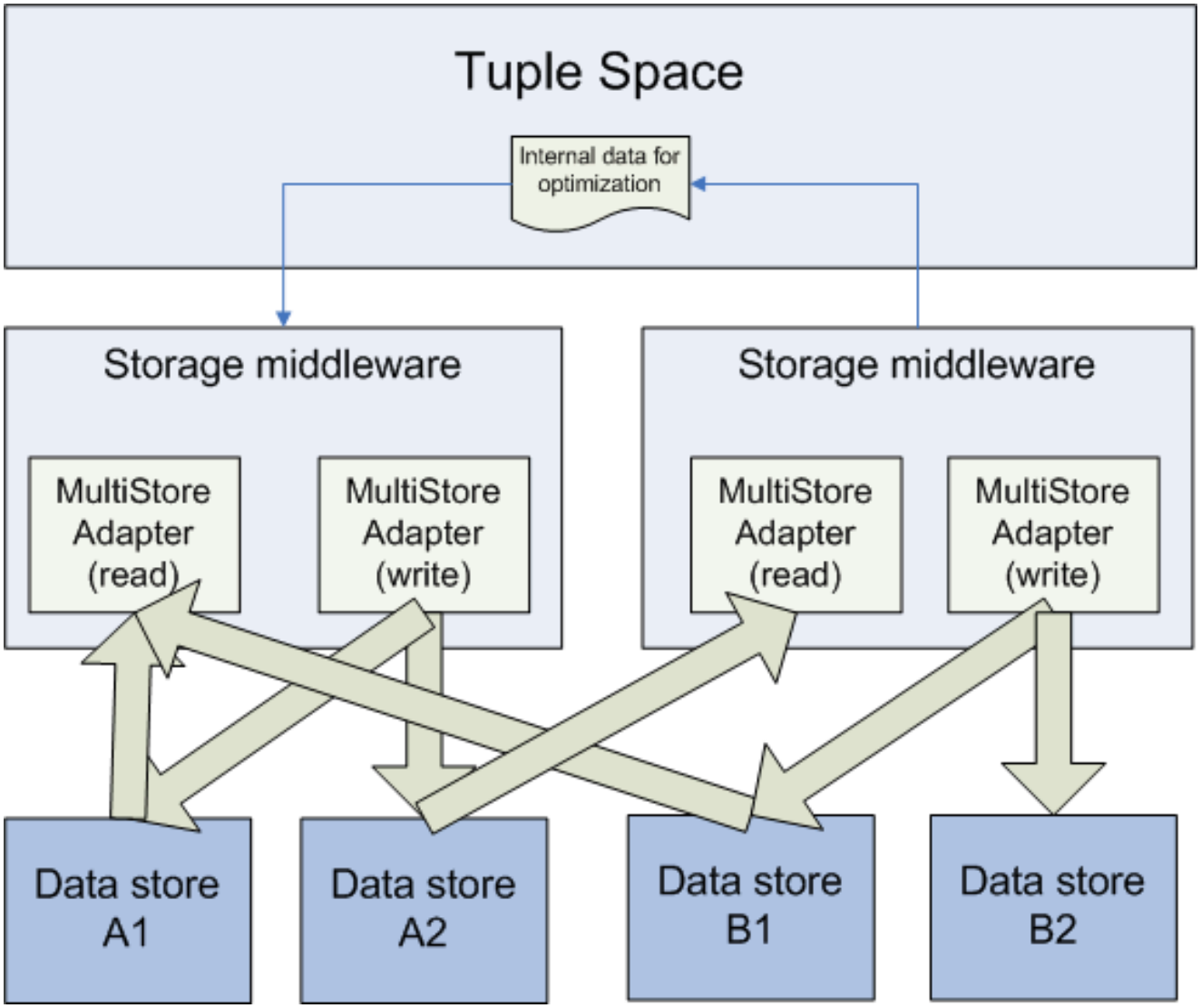
Distribution manager

- Operation distribution according to some distribution function
- API for inter-kernel communication

- Which is optimal relationship between the distribution manager within the tuplespace kernel and the storage layer?

Relationship to other WPs

Relationship to Storage (WP1)



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

Relationship to Web Services (WP4)

- To be specified
  - Mapping of the WS-communication into tuplespace
  - Mapping of the WS-registry into the tuplespace
  - Data, message and protocol mediation

- Can we specify Triple Space independently of the processes that will use it?
- Do SWS raise any requirements that must be explicitly considered in the architecture?

Relationship to Security/Trust (WP5)

- Security and trust measures include:
  - Agent authentication
  - Access policy
  - Data encryption
  - Reputation

- Which components need to be extended?
- Do any aspects of the architecture require modification?

Action points

Action points

- What concrete form should the architecture take (component-oriented, service-oriented…)?
- Are the presented components sufficient for the realization of the project goals?
- How do the activities of others WPs impact on the presented architecture?

So how do we bring Triple Space to its full potential???