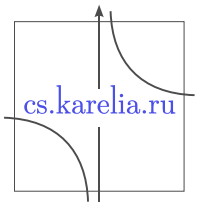
Subscription Operation in Smart-M3

Aleksandr A. Lomov, Dmitry G. Korzun

Petrozavodsk State University
Department of Computer Science

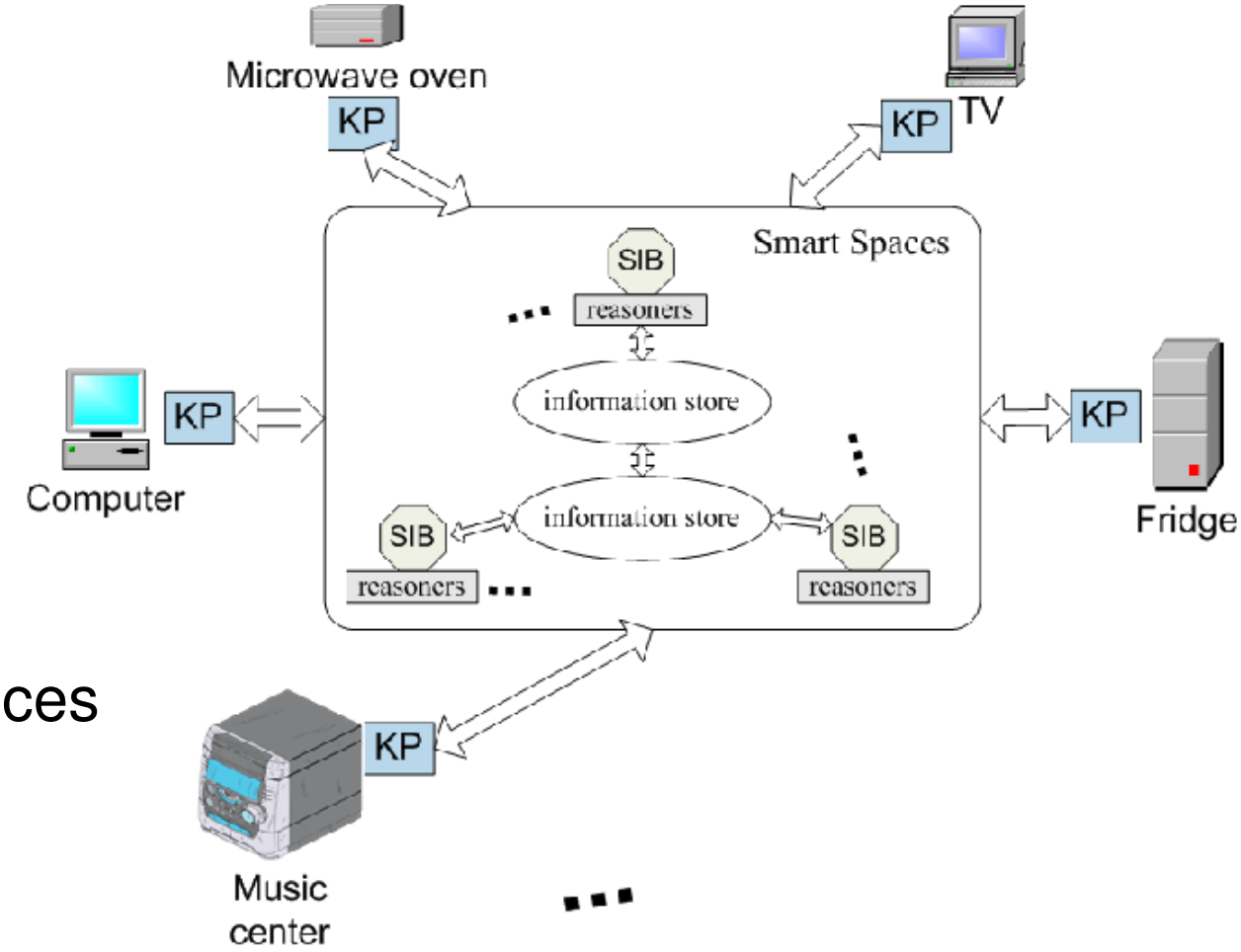
10th FRUCT Conference, November 6–11, Tampere, Finland

Table of Contents



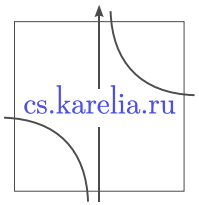
Infrastructure: RDF-based space with pub/sub

- Semantic information brokers (SIBs) maintain space content in low-level RDF triples
- Application consists of several knowledge processors (KPs) running on various devices
- Smart space access protocol (SSAP) for SIB↔KP communication; it supports subscription to RDF triples
- Smart-M3: **M**ultidomain, **M**ultidevice, **M**ultivendor



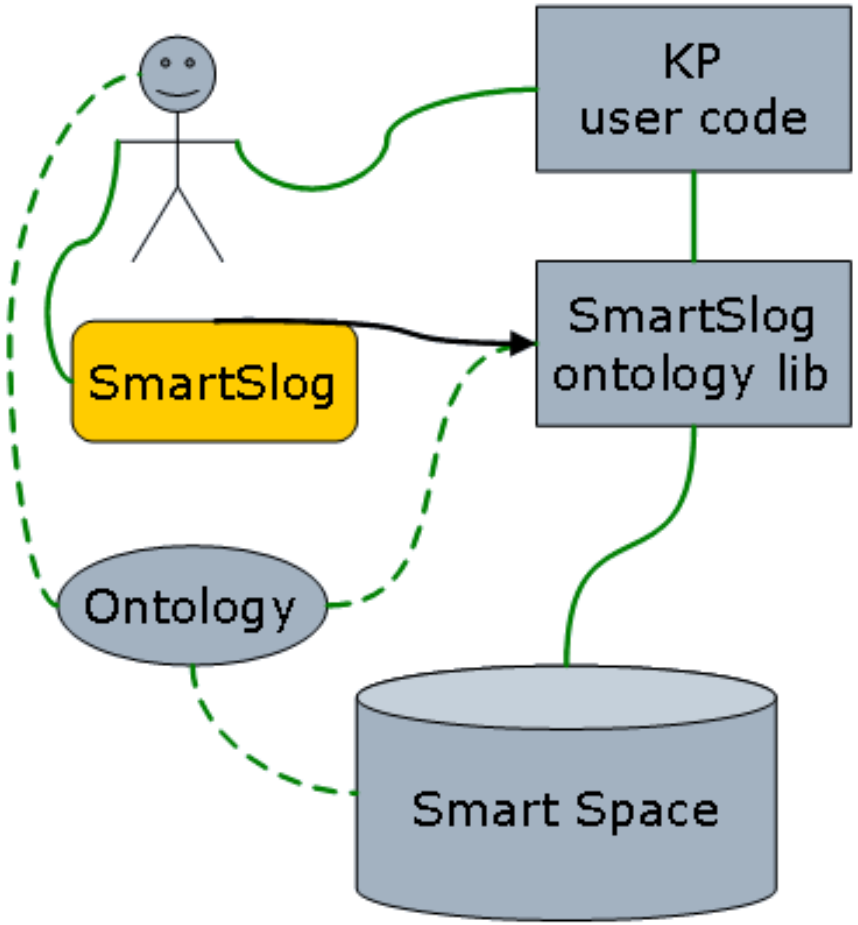
KP development tools

Low-level (RDF triple)	High-level (OWL object & properties)
Whiteboard, Whiteboard-Qt *C/Glib, C/Dbus, C++/Qt (Smart-M3)*	Smart-M3 ontology to C-API generator *C/Glib, C/Dbus (Smart-M3)*
Smart-M3 Java KPI library *Java (University of Bologna and VTT)*	Smart-M3 ontology to Python generator *Python (Smart-M3)*
M3-Python KPI (m3_kp) *Python (Smart-M3 distribution)*	SmartSlog *ANSI C, C# (Petrozavodsk State University)*
C# KPI library *C# (University of Bologna)*	
KPI_low *ANSI C (VTT-Oulu)*	



SmartSlog SDK

- Library generator for Smart Space ontology
- Mapping OWL to code (C, C#):
  - KP uses ontology library
  - ontology abstractions in API
  - modest code
  - low-level KPI is hided (KPI_low)
- High-level communication primitives
  - knowledge patterns
  - **subscription**



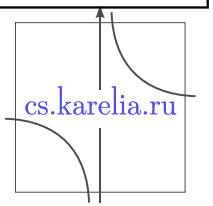
KP programmer can think in abstract ontology terms!

cs.karelia.ru

Subscription types

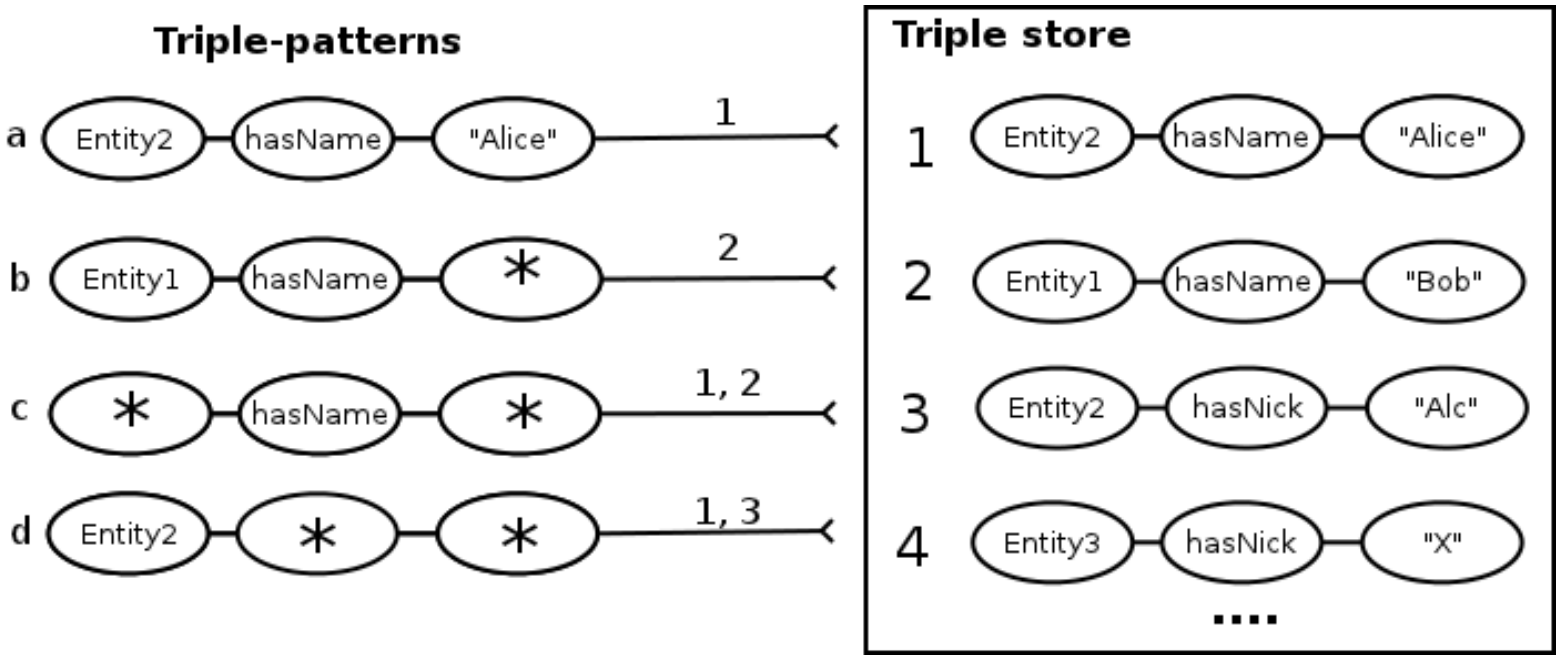
	Type	Description
Low-level (basic subscription of SSAP)		
1	Triple-based subscription	Select triples to subscribe by a triple-pattern
High-level (implemented or planned in SmartSlog)		
2.1	Property-based subscription: data properties	Select some data properties of some individuals to a container
2.2	Property-based subscription: object properties	Select some properties that link some individuals and put them into a container
2.3	Property-based subscription: data, object properties	Mixture of 2.1 and 2.2. A container is a set of some properties; they are treated independent
3	Data property value is subject to constraints (cf. 2.1)	Constrain a data property to detect specific changes (equality, range, threshold, etc.)
4	Subscription to a class	Select classes to detect appearance of their individuals in the smart space

Note: Subscribed elements are treated independently within either a triple set or container



Basic subscription operation of SSAP

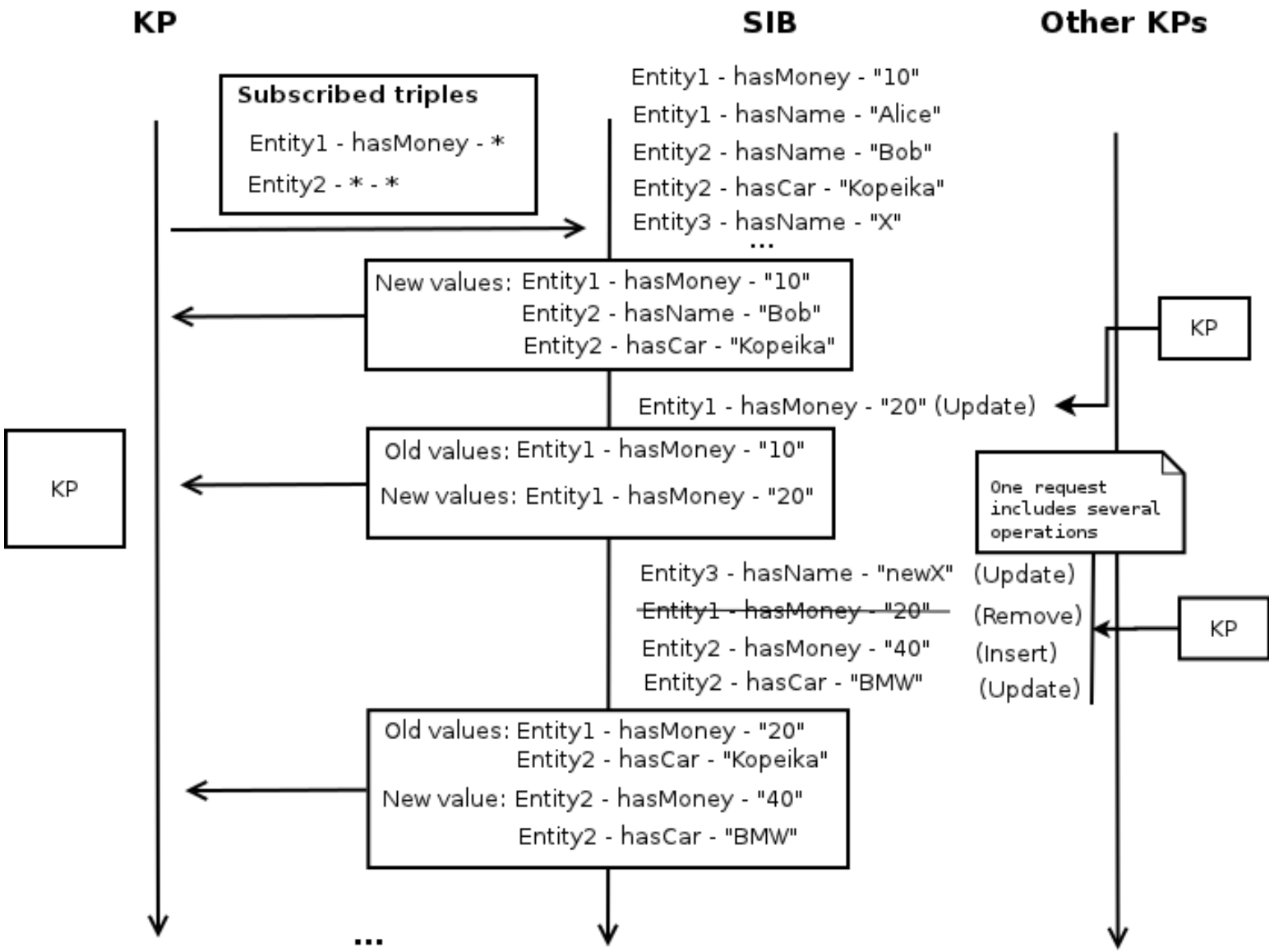
- Triple-pattern is a triple with masks (any value)
- Triple-pattern selects a set of triples for subscription
- Triple-pattern maps to a list of actual triples in the smart space



a, b, c, d are triple-patterns, 1, 2, 3, 4 are actual triples

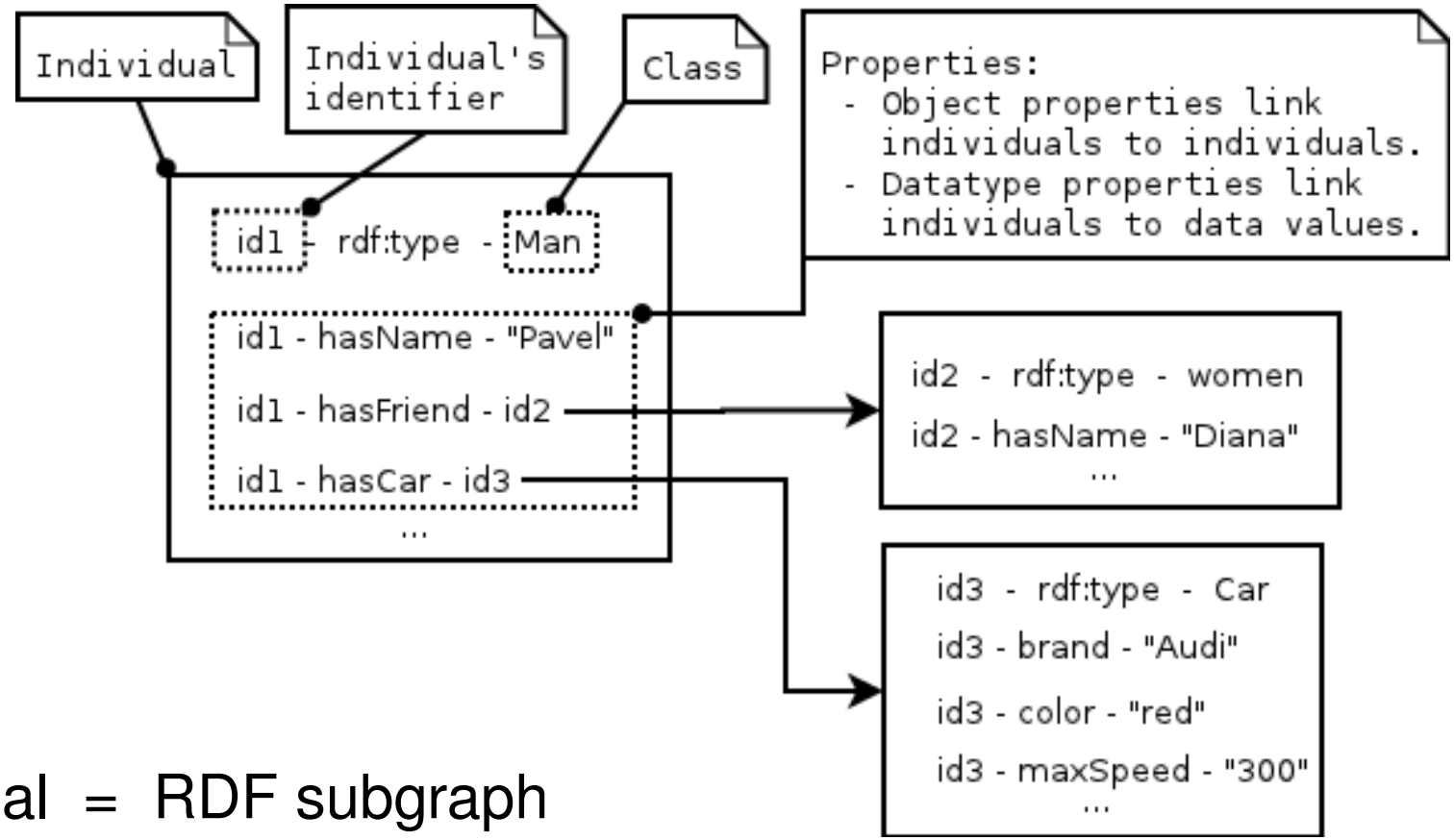
cs.karelia.ru

Triple-based subscription process



Ontological representation of an individual

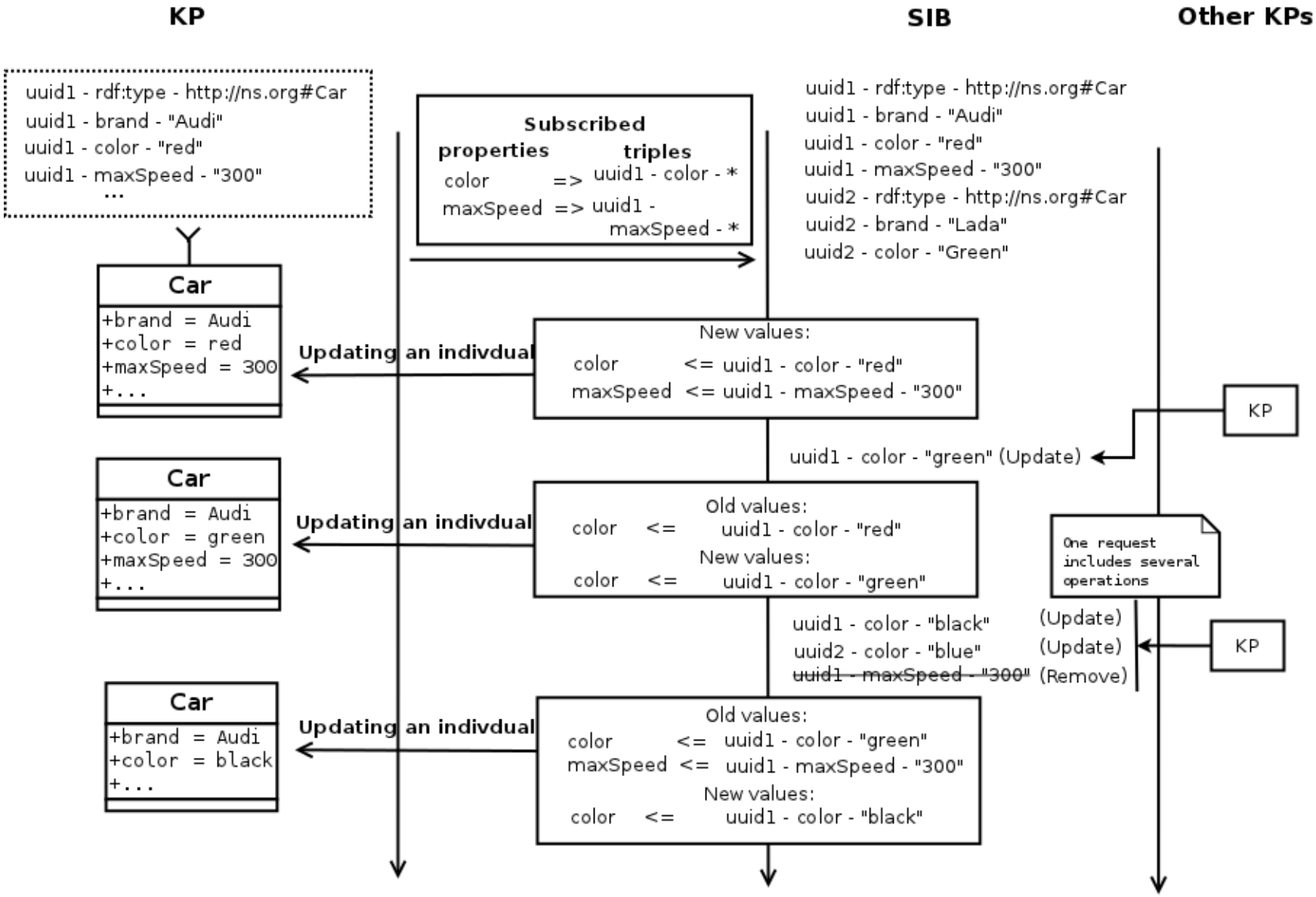
- rdf:type is a triple to represent an individual
- it links individual's ID with its class
- given individual's ID, properties of the individual are accessible



Individual = RDF subgraph

cs.karelia.ru

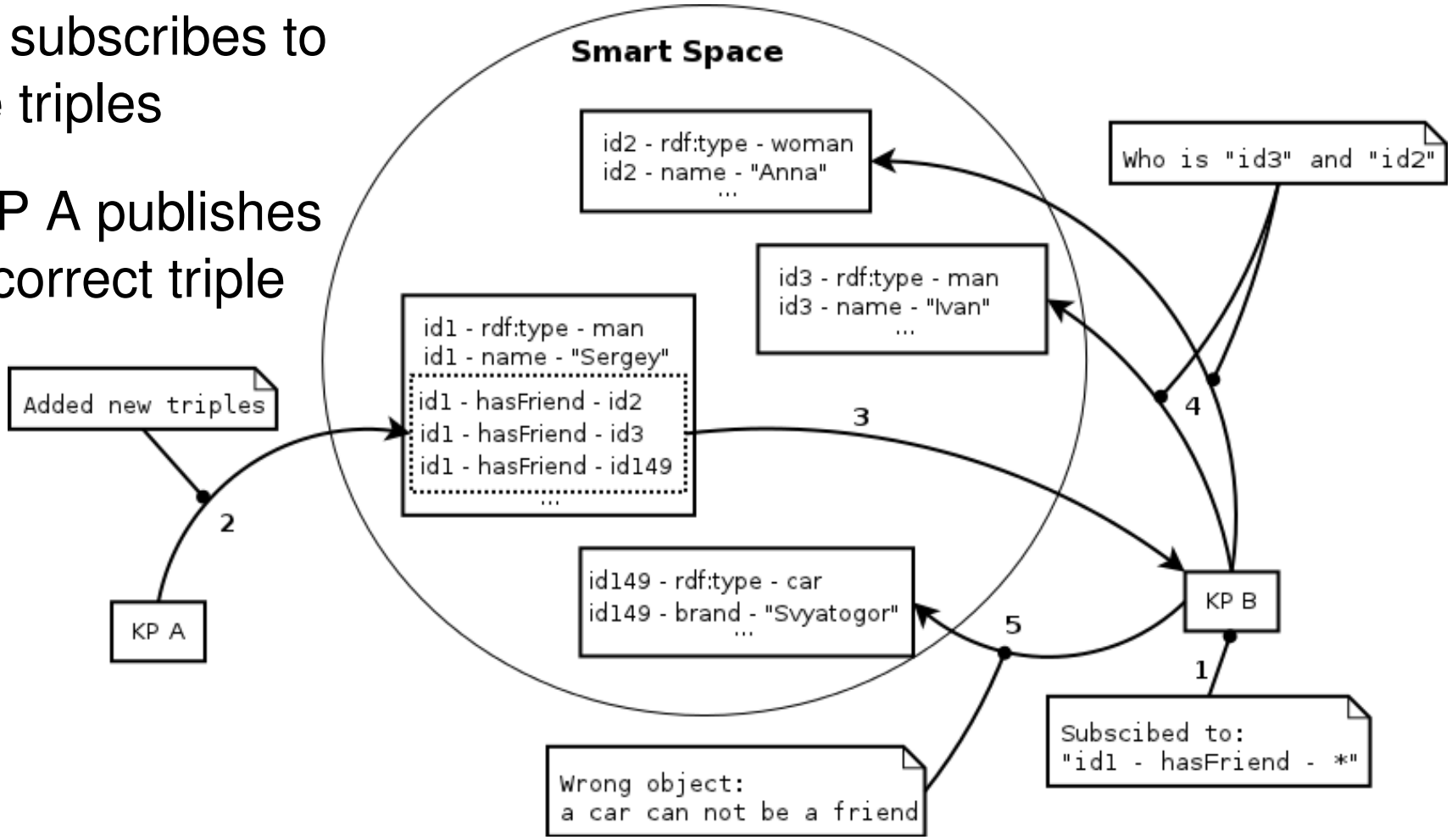
Property-based subscription process



cs.karelia.ru

Content inconsistency in smart space

- KP B subscribes to some triples
- . . . KP A publishes an incorrect triple

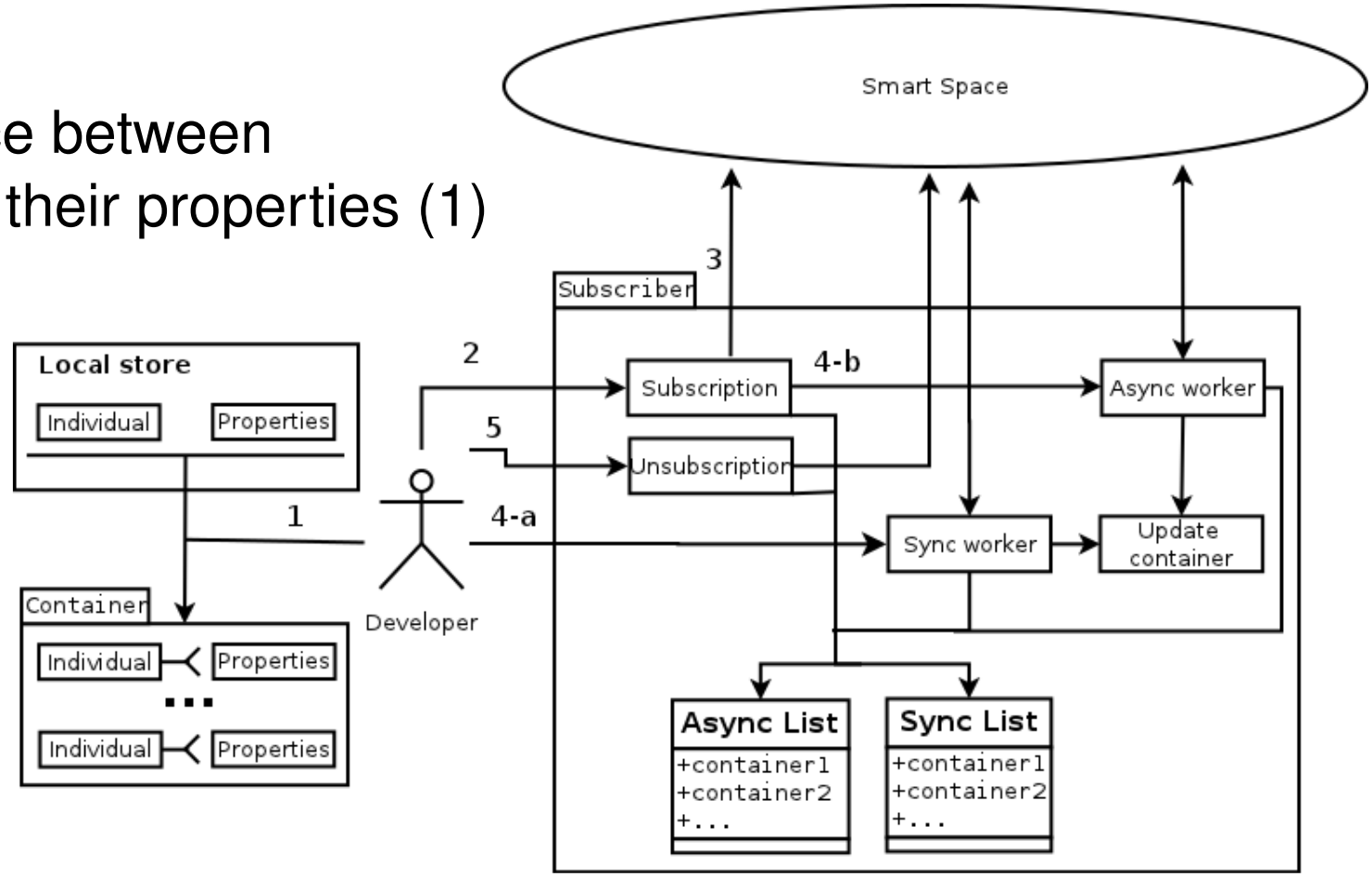


High-level subscription can handle this inconsistency

cs.karelia.ru

SmartSlog subscriber architecture

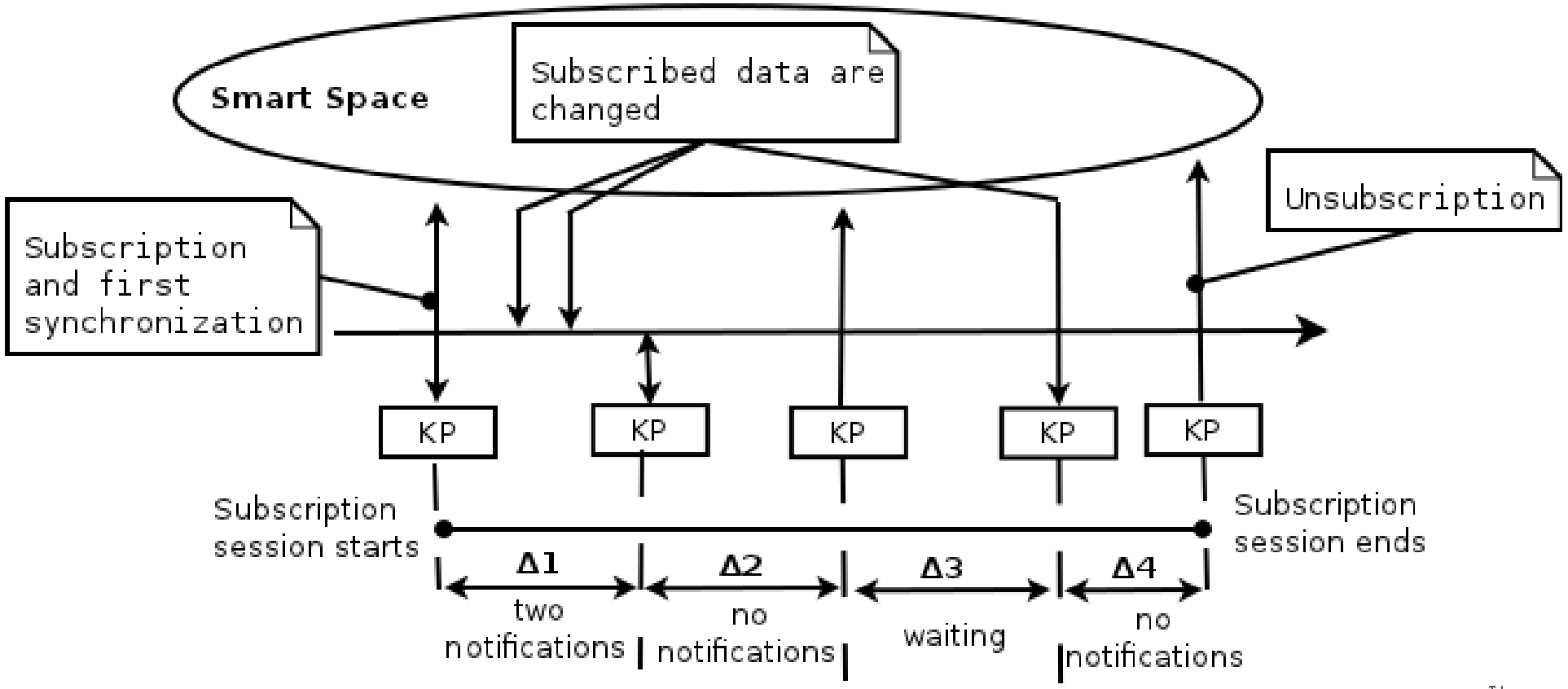
- Correspondence between individuals and their properties (1)
- Subscription to container (2, 3)
- Handling subscription (4-a or 4-b)
- Unsubscription from container (5)



cs.karelia.ru

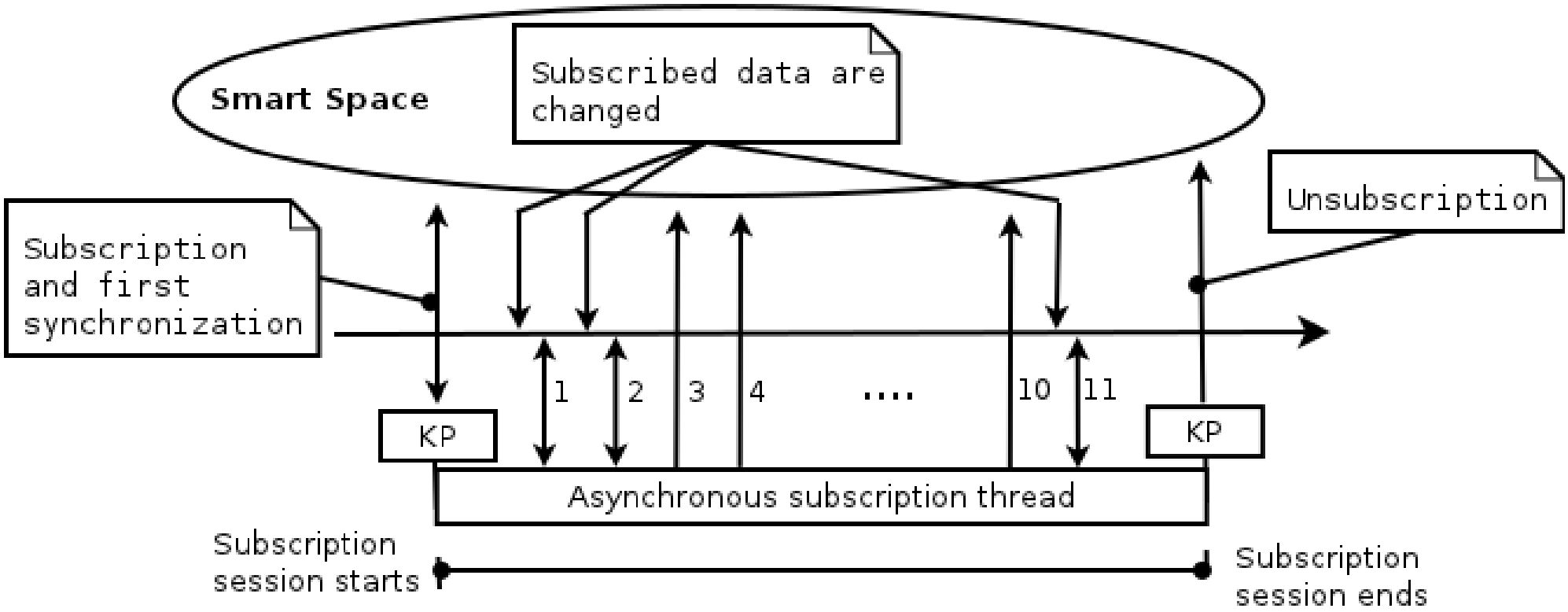
Synchronous subscription

Subscription blocks the current thread, and KP waits for updated data



Asynchronous subscription

Additional thread updates data in background



Conclusion

- Two main types of the subscription
  - Low-level (triple-based)
  - High-level (ontology-based on object and data properties)
- Future directions
  - Subscription based on classes
  - Supporting constrains to subscribed value
- SmartSlog developers wiki:
  `http://oss.fruct.org/wiki/SmartSlog/`
- Open source code:
  `http://sourceforge.net/projects/smartslog/`

cs.karelia.ru

Thank you!