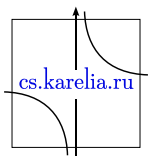
SmartSlog Session Scheme for Smart-M3 Applications

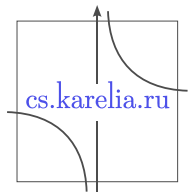
Aleksandr A. Lomov

Petrozavodsk State University
Department of Computer Science



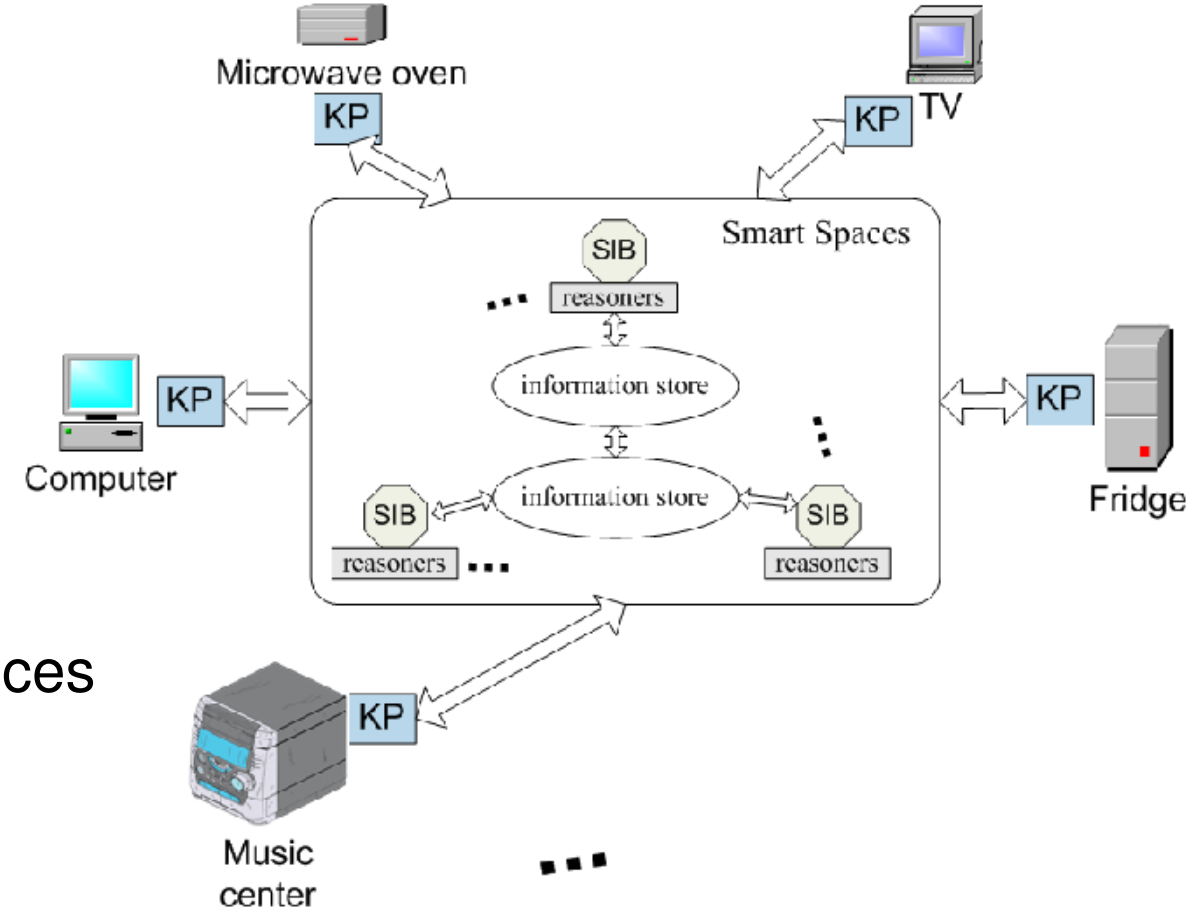
12th FRUCT Conference, November 5–9, Oulu, Finland

Table of Contents



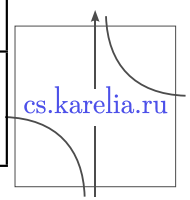
Infrastructure: RDF-based space with pub/sub

- 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



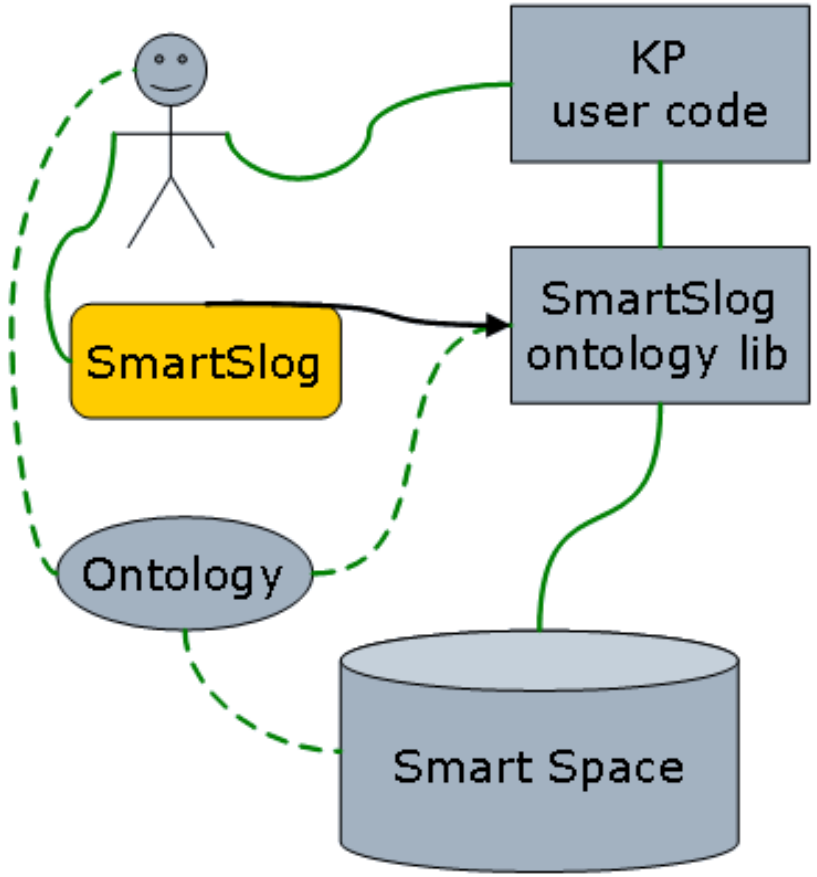
KP development tools

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



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
  - **session**
  - knowledge patterns
  - subscription

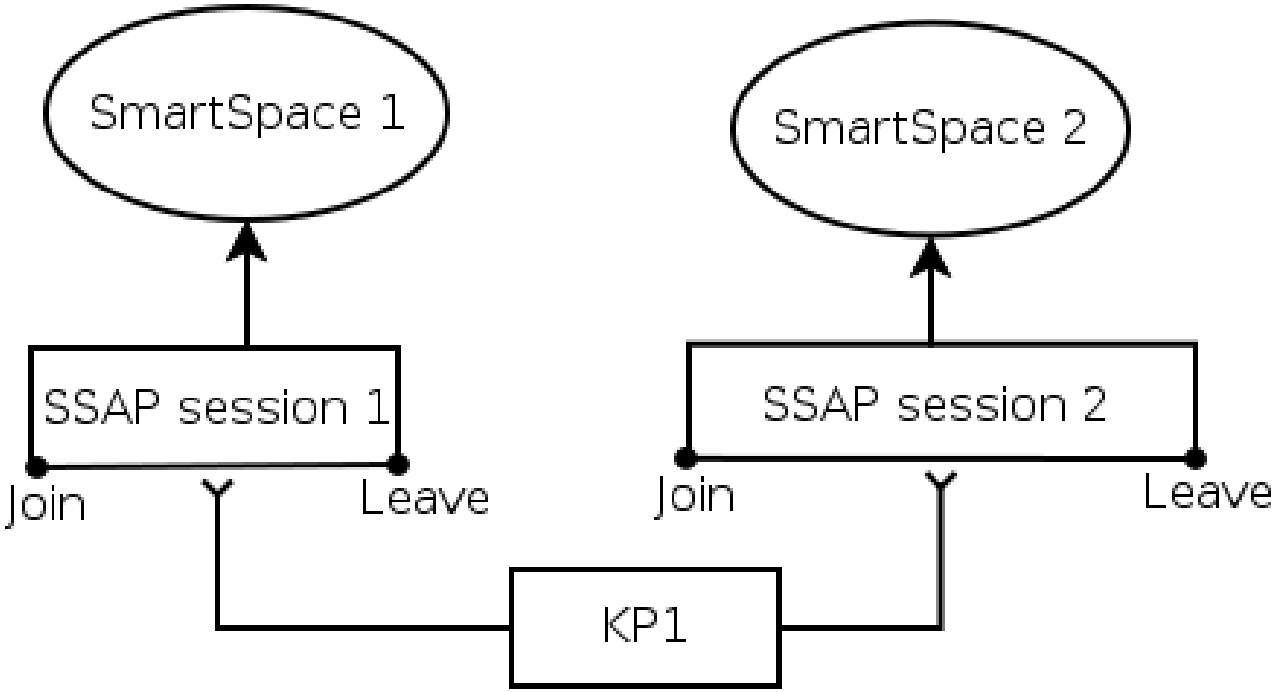


KP programmer can think in abstract ontology terms!

cs.karelia.ru

SSAP session

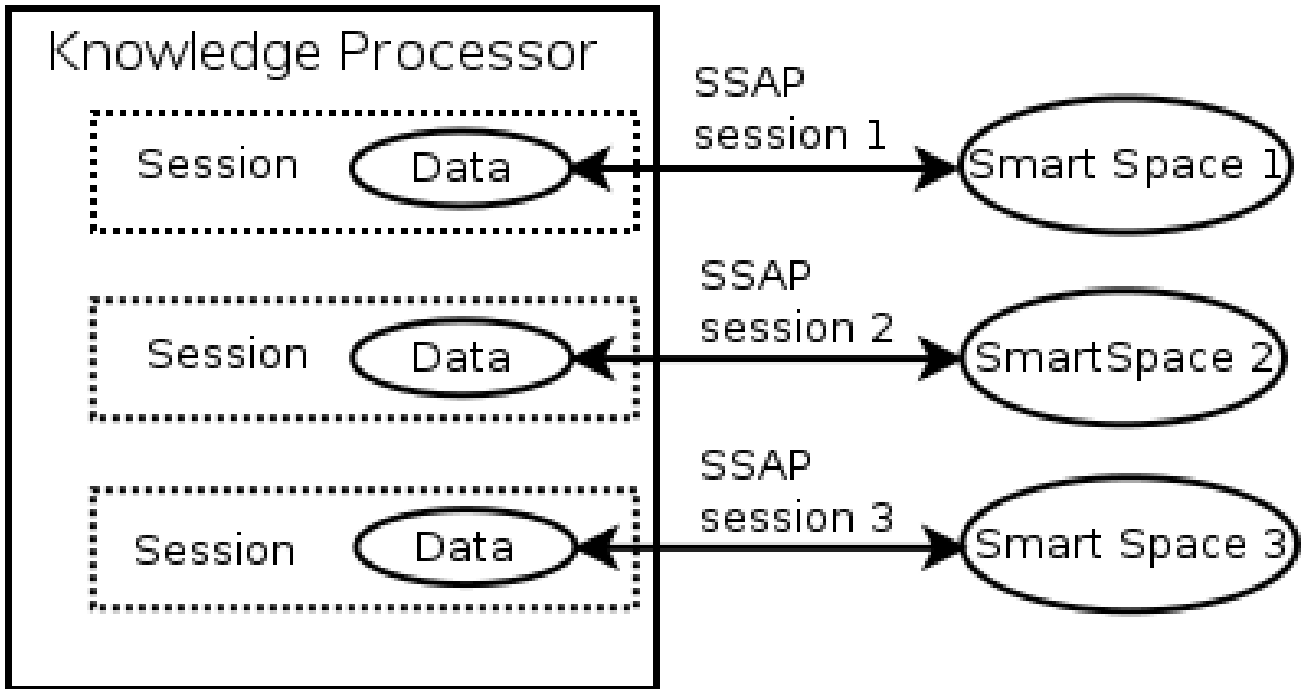
- Start with the Join operation
- Work with the smart space
- Terminate session with Leave operation



cs.karelia.ru

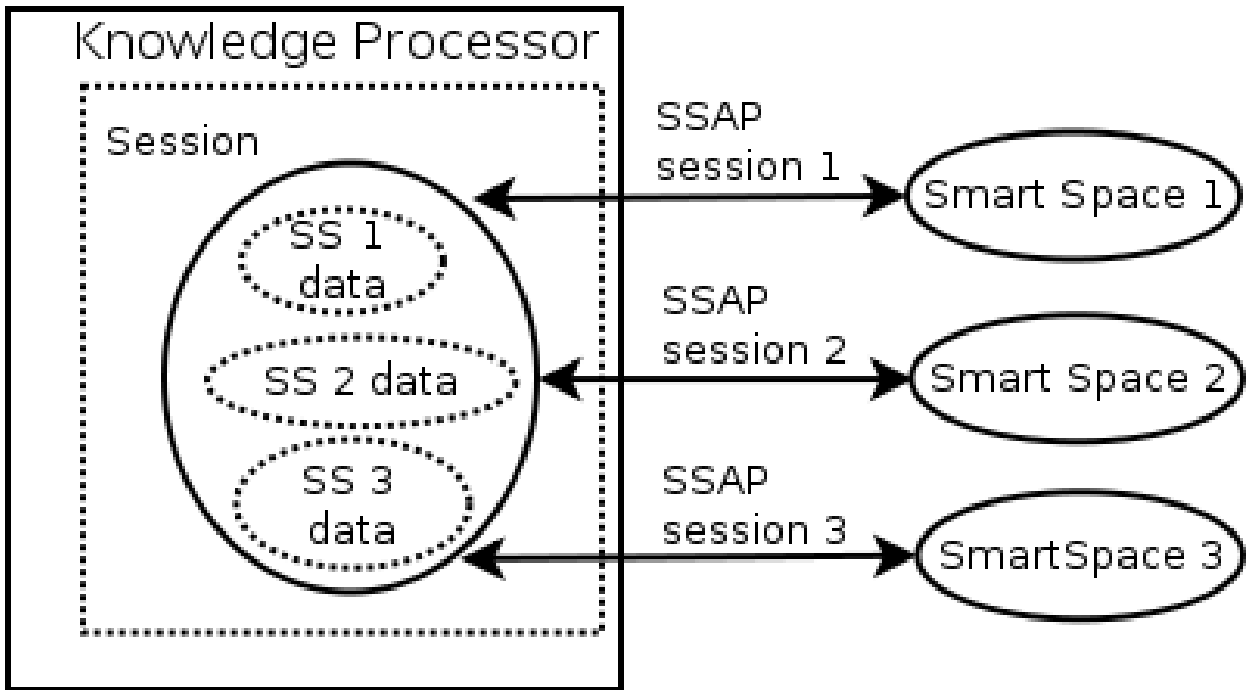
SSAP session drawbacks

- Each session is works independently
- Data do not share between sessions
- There is no scheme to work and manage several sessions
- There is no mapping smart spaces to a one local space



New session scheme

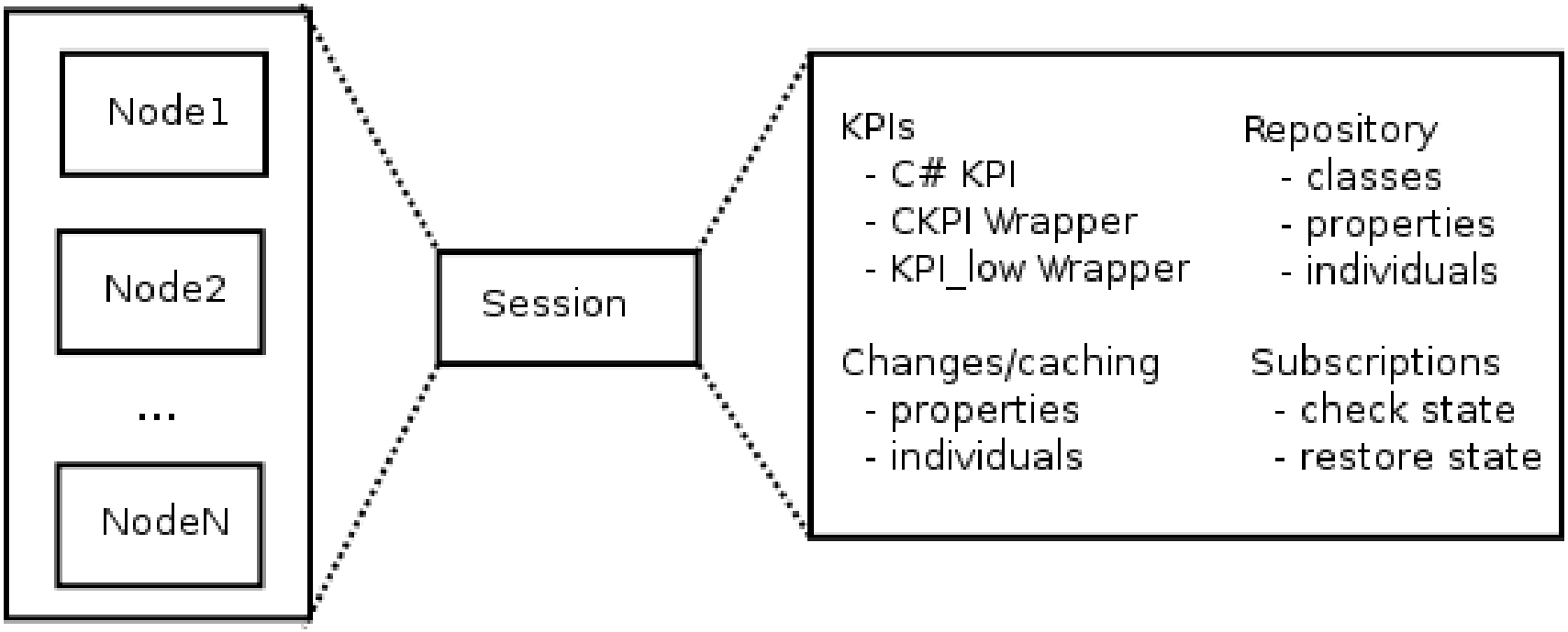
- SSAP sessions work in one session
- one store for all SSAP session
- local store maps to several smart spaces



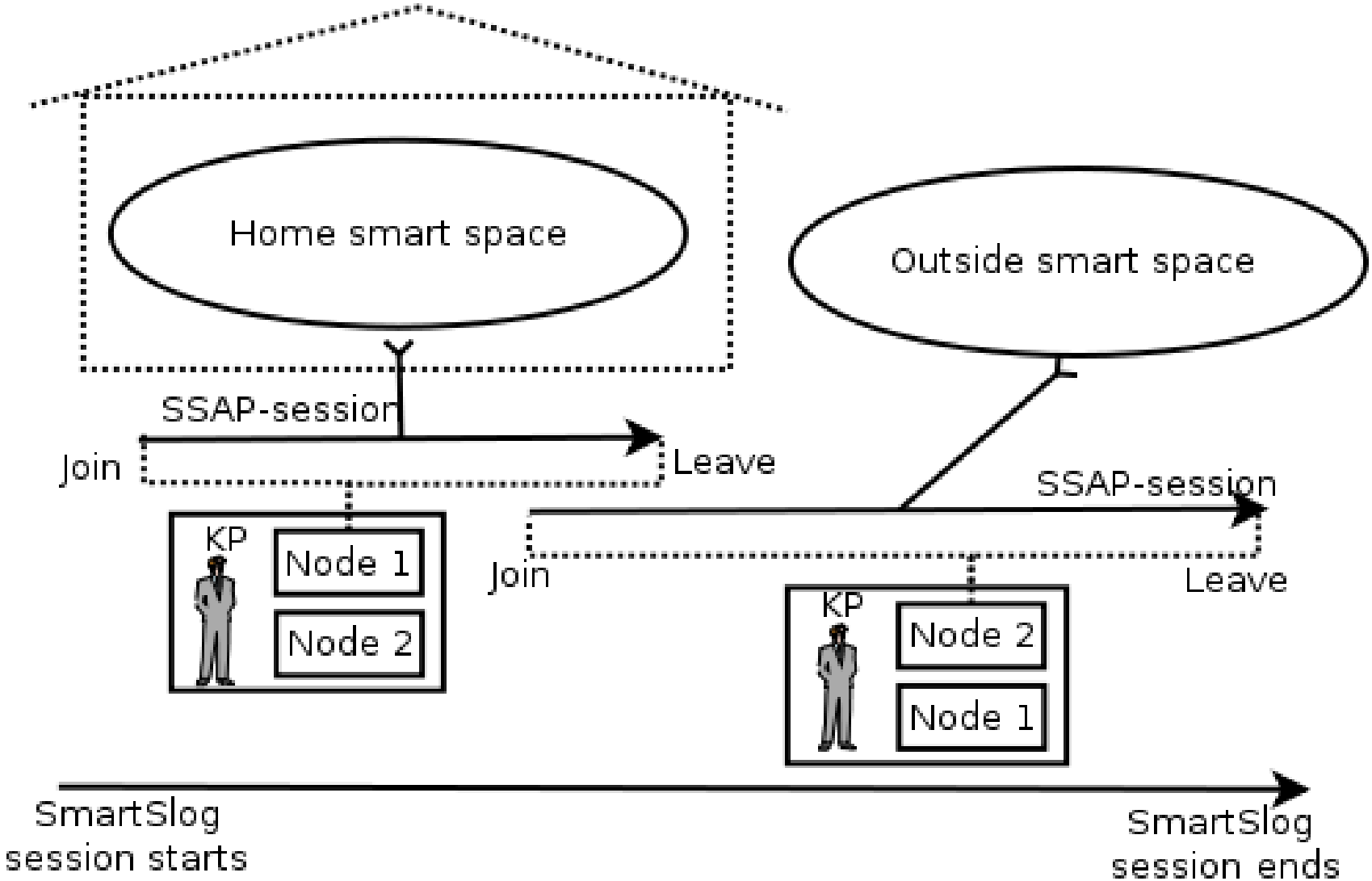
cs.karelia.ru

New session scheme implemented in the SmartSlog

- Node represents a SSAP session
- Repository is a local store for ontology entities
- KPI represents different KP interfaces
- Information about changes (caching data)
- Information about subscriptions (manage subscriptions)

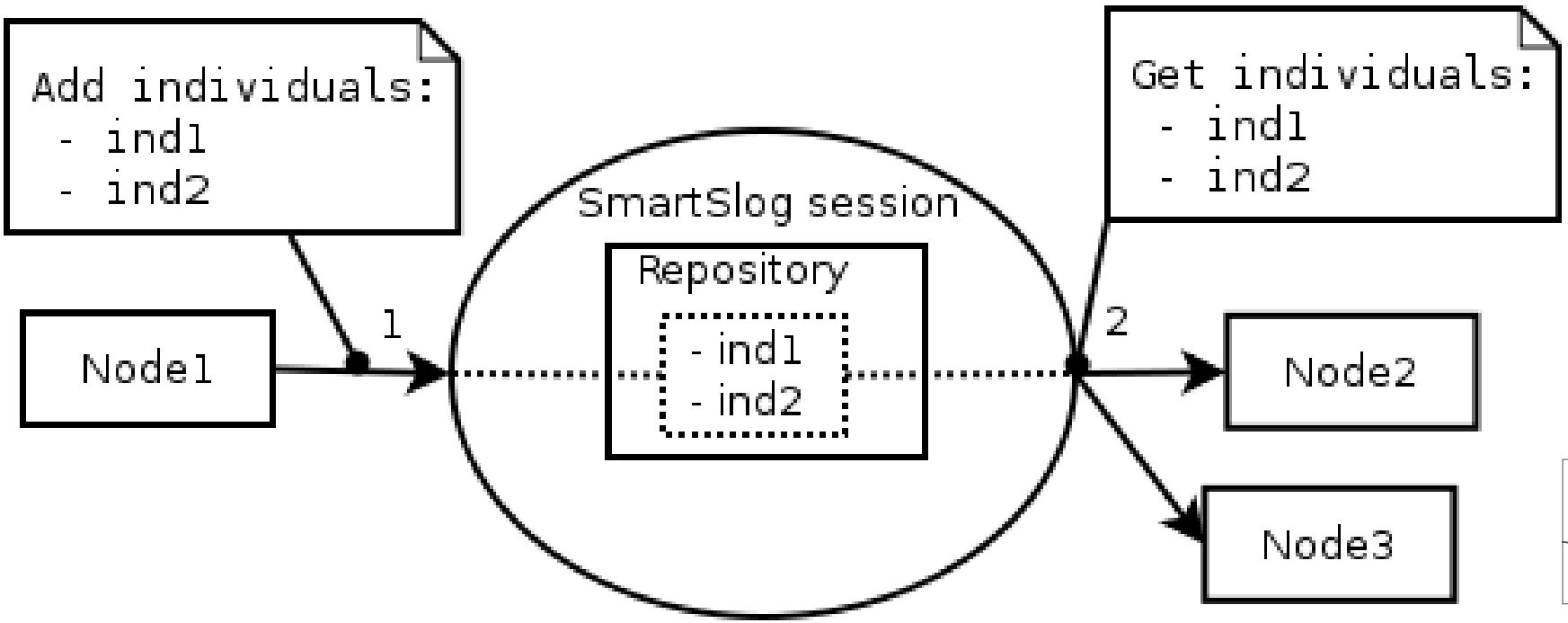


SmartSlog session



Session repository

- Storing data for several SSAP sessions (Nodes)
- Mapping several smart spaces to one local space
- Transferring data between smart spaces

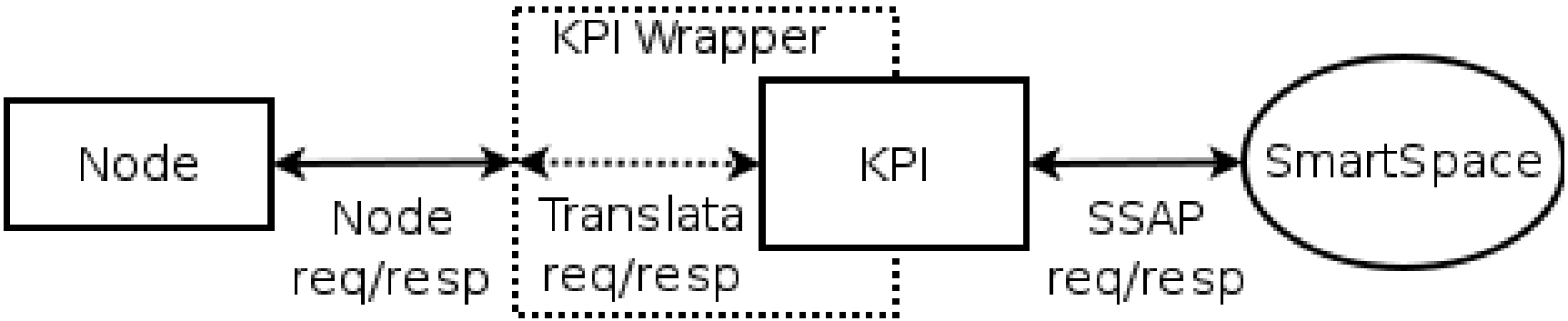


KPI's support

Session loads KPIs in run-time and initializes them. Nodes get the KPI from the session and work with it through KPI Wrapper (C# version).

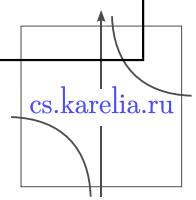
KPI Wrapper assembly contains:

- Simple implementation of Triple and BlankNode classes
- Some interfaces and abstract classes to simplify implementation Wrapper for KPI



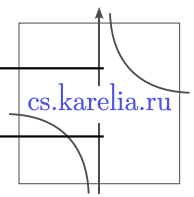
ANSI C and C# versions comparison

ANSI C 0.41alpha	**C# 0.17alpha**
One local store for all sessions.	One local store for each session.
It needs to switch between several sessions manually by using a special function.	Appropriate objects represent sessions.
Session works with one SSAP-session.	Session contains several SSAP-sessions.
Different low-level KPIs can not be used in the run-time.	Session can work with different KPIs (using Wrapper) in the run-time.



Session examples

ANSI C version
sslog_ss_init_new_session_with_parameters(”X”, ”127.0.0.1”, 10010, ”Home”); sslog_ss_init_new_session_with_parameters(”X”, ”194.85.173.9”, 10012, ”Bar”);
sslog_switch_sib(”Home”); individual_t *timo = sslog_new_individual(CLASS_MAN); sslog_add_property(timo, PROP_FNAME, ”Timo”); ... sslog_ss_insert_individual(timo);
sslog_switch_sib(”Bar”); sslog_ss_add_property(timo, PROP_DRINKS, ”beer”);
sslog_ss_leave_session_all(); sslog_repo_clean_all();
C# version
Node timoNode = new Node(nodeNameTimo, smartSpaceName, address, port); Individual timo = timoNode.CreateIndividual(OntStructure.Man); node.Insert(timo);
ISession wifeSession = SessionManager.CreateSession(); Node wifeNode = new Node(wifeSession, ...); Individual wife = wifeNode.CreateIndividual(OntStructure.Woman); wifeNode.Insert(wife);
SessionManager.DestroySession(wifeSession);



Conclusion

- New session scheme
  - several SSAP session
  - repository - one local store
- Future directions
  - Subscriptions managing
  - Property caching
- SmartSlog developers wiki:
  `http://oss.fruct.org/wiki/SmartSlog/`
- Open source code:
  `http://sourceforge.net/projects/smartslog/`

Thank you!