

A Framework for Context-Aware Applications for Smart Spaces.

Natalia Díaz Rodríguez (ndiaz@abo.fi).

TUCS (Turku Centre for Computer Science)

Department of Information Technologies, Åbo Akademi University
Turku, Finland.

SMART SPACE:

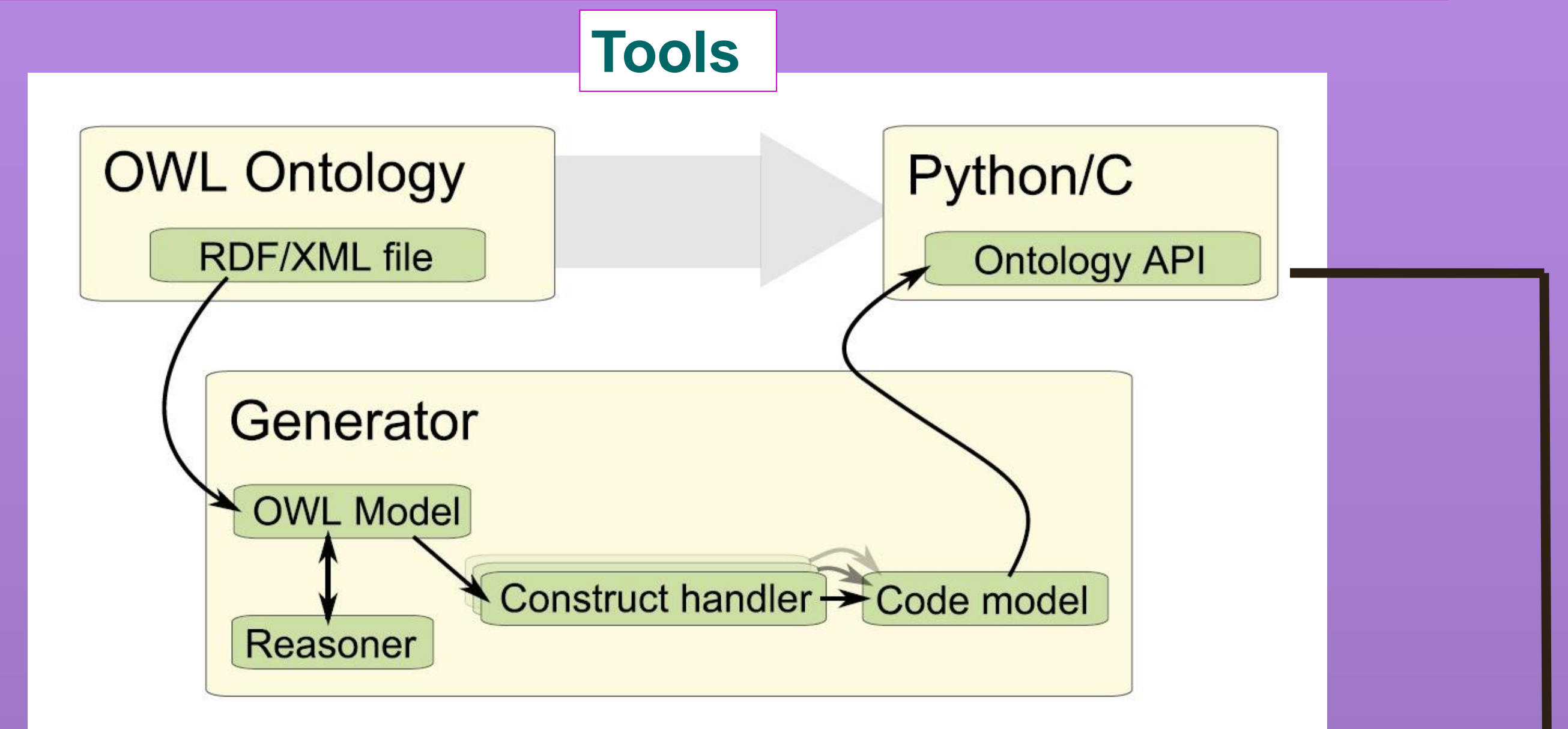
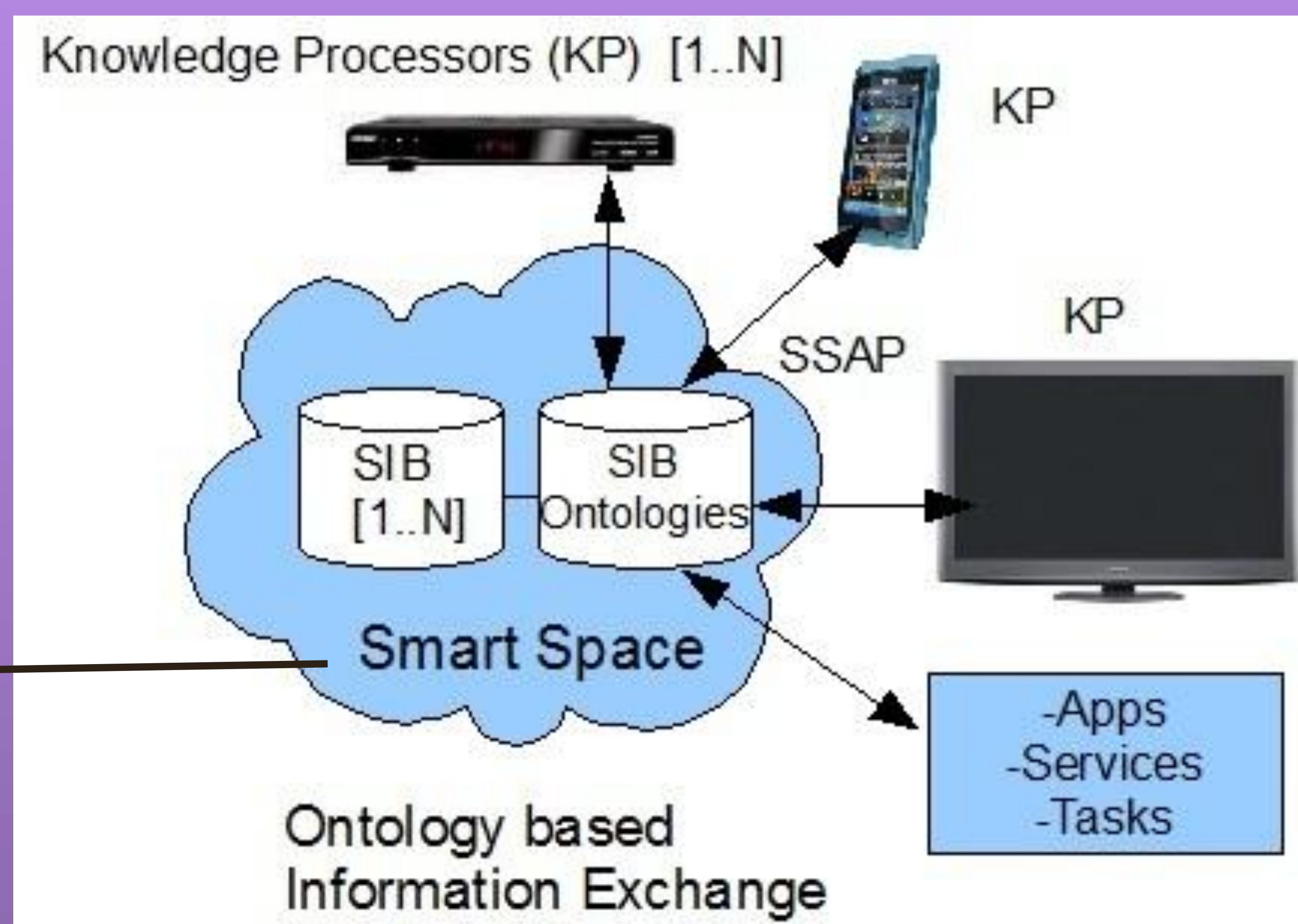
An abstraction of space encapsulating both information from a physical space and access to this information allowing devices to join and leave the space.

Publish-subscribe methods are used in these dynamically changing environments.

NOKIA'S SMART-M3 (An implementation of Smart Space):

- A Multi part, Multi device and Multi vendor (M3) **open source cross-domain platform** consisting of a space-based communication mechanism for independent agents to communicate.
- Semantic Information Broker (SIB)**: The central repository of **RDF triples** responsible for information storage, sharing and management through the Smart Space Access Protocol (SSAP).

- KNOWLEDGE PROCESSORS (KPs)** entities implement functionality and interact with the Smart Space by inserting/retrieving/querying common information.
- An **APPLICATION** is constructed by aggregating KPs which perform tasks.
- COMMUNICATION** happens not device to device but through the SIB.



1) An **Ontology Library Generator** (in **Python** and **C**): creates a static API from an OWL-DL ontology containing Classes and its properties' Get & Set methods.

2) A **Middleware framework**: Abstracts the communication with the persistence layer. Provides to the generated ontology API: RDF triple handling, synchronous and asynchronous querying.

Rule Expression embedding into Python language

Since the end-user should not deal with the RDF store directly, a PythonRules module is presented to translate Python logic expressions to the SIB API (*Query, Subscribe, Insert, Remove, Update*).

AIM: Design a **Rule syntax** for allowing users -with knowledge of basic programming- easy definition of Rules to model Smart Space applications.

Python Rule Syntax:

With() // When() >> Then()

- With** class handles **Existence Assumptions** in the Smart Space.
- When** class handles **Conditions**.
- Then** class handles **Actions**.

Knowledge Processor Programming

```
def main(args):
    app = QtGui.QApplication(sys.argv)
    smartSpace = ('x', (TCPConnector, ('127.0.0.1', 10010)))
    phoneKP = PhoneKP.create(smartSpace)
    #Definition of Rules
    sys.exit(app.exec_())
```

```
user = User(1, "Researcher", "Peter", True, False)
room = Room("B4050", "ICT House", "Turku", True)

condition1 = lambda: user.isBusy()
condition2 = lambda: room.getOccupied()
conditions = [condition1, condition2]
action = lambda: user.setVoiceMail(True)
myRule = With([user, room]) // When(conditions) >> Then(action)
diem.addRule(myRule)
```

RULES are stored and passed to a rule **Inference Engine** which, based on given/sensed (sensors, RFID, devices, etc) atomic context information, will **infer** higher level **Context Information**.

APPLICATION DOMAIN for the Smart Space Development framework:

- Office Domain
- Home Automation
- Elderly Monitoring Systems, etc.