Une interface python pour Graal

https://gite.lirmm.fr/graphik/graal-group/graal/-/tree/Stage-Salembien

Tom Salembien

30/09/2021

EQUIPE PROJET
GraphIK
INRIA Sophia-Antipolis
LIRMM

Objectif

```
import graal
graal = graal.Graal()
kb = graal.make_knowledge_base()
kb.add_atom("human(socrates).")
kb.add_rule("mortal(X) :- human(X).")
for answer in kb.make_answers("?(X):-mortal(X).":
    print(answer['X'])
```

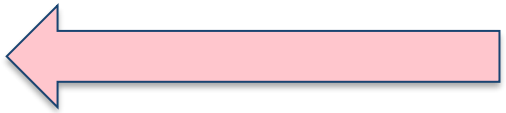
Outil: py4j
https://www.py4j.org/contents.html

```java
import py4j.GatewayServer;

public class Foo {

    public Foo() {
    }
    public String bar(String arg1, String arg2) {
         return arg1 + arg2;
    }
    public static void main(String[] args) {
          GatewayServer server = new GatewayServer(new Foo());
          server.start();
   }
}
```

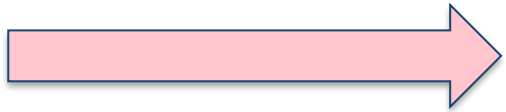
```python
from py4j.java_gateway import JavaGateway

gateway = JavaGateway()
connector = gateway.entry_point

print(connector.bar("Hello", "GraphIK"))

gateway.shutdown()
```

Py4j: Communication

DLGP, filenames



types élémentaires (str, int, …)



Substitutions (JSON str)

Ne peut pas échanger d'objets complexes !

Objets « virtuels » en Python

```
graal = graal.Graal()

kb = graal.make_knowledge_base()

kb.add_rule("mortal(X) :- human(X).")
```

```
public class GraalConnector {


   public  int initKnowledgeBase() {
      KBBuilder base = new KBBuilder();
      int index = getNewIndex();
      this.htKBBuilder.put(index, base);
      return index;
   }


   public int addRule(int ident, String dlgp)  {
```


```
from py4j.java_gateway import JavaGateway
import KnowledgeBase as kb

class Graal:

    def __init__(self, jar = ".\Py4GraalJava.jar"):
        self.proc = subprocess.Popen('java -jar "{}"'.format(jar))
        self.gateway = JavaGateway()
        self.connector = self.gateway.entry_point

    def make_knowledge_base(self):
        kbident = self.connector.initKnowledgeBase()
        return kb.KnowledgeBase(self.connector, kbident)

    def __del__(self):
        self.gateway.shutdown()
        self.proc.terminate()
```

```
class KnowledgeBase:

    def __init__(self, connector, ident):
        self.connector = connector
        self.ident = ident

    def add_rule(self, dlgp):
        self.connector.addRule(self.ident, dlgp)
```

Pythonize Graal: enumerations

```python
for answer in kb.make_answers("?(X):-mortal(X).":
    print(answer['X'])
```

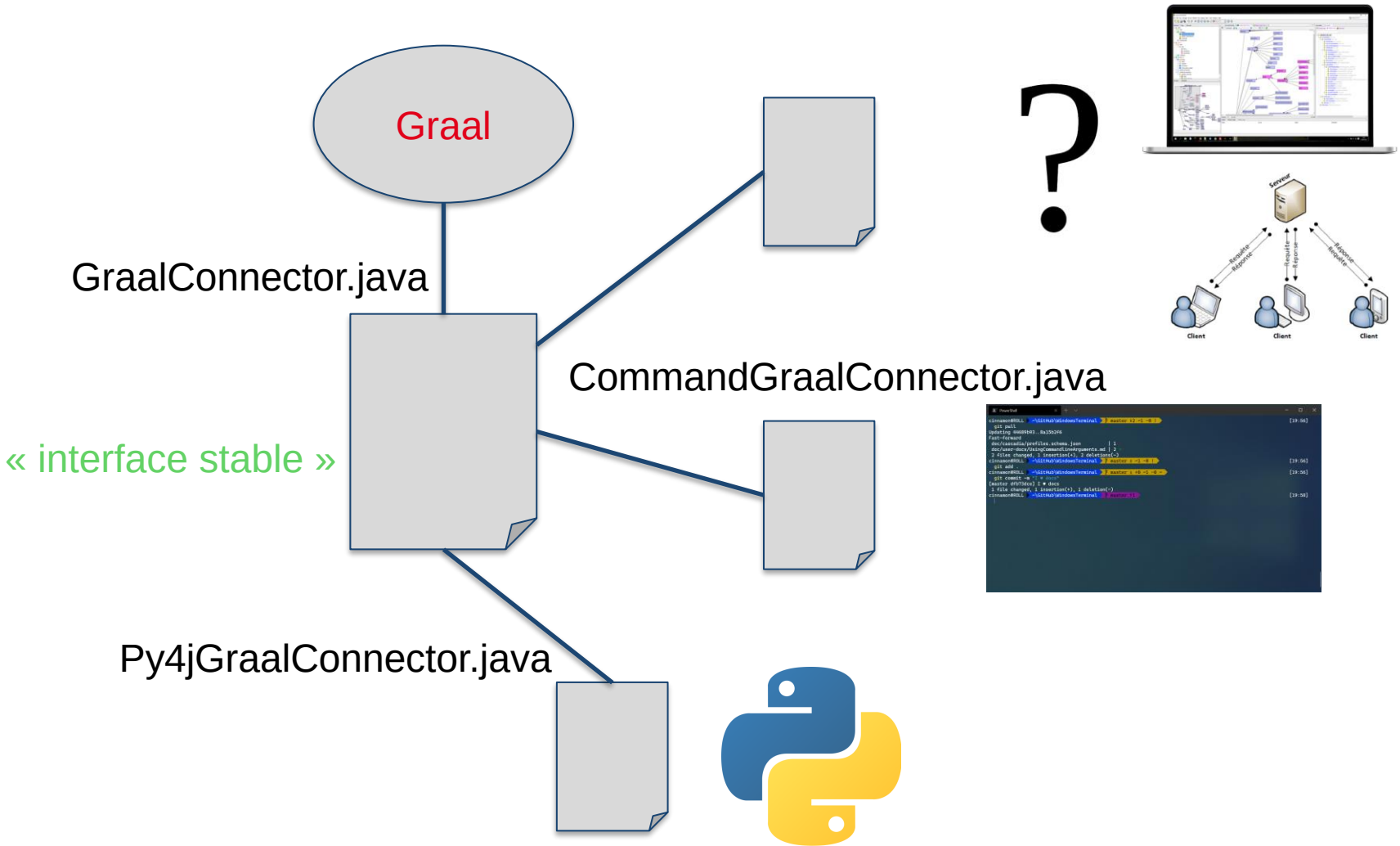
Class KnowledgeBase

```java
public  int initAnswers(int ident, String dlgp) {
    KnowledgeBase kb
        = this.htKnowledgeBase.get(ident);
    ConjunctiveQuery query
        = DlgpParser.parseQuery(dlgp);
    CloseableIterator<Substitution> iterator
        = kb.query(query);
    int index = getNewIndex();
    this.htAnswers.put(index, iterator);
    return index;
}

public String nextAnswer(int index) {
    CloseableIterator<Substitution> iterator
        = htAnswers.get(index);
    String res = new String();
    if (iterator.hasNext()) {
        return subsToString(iterator.next();
    }
    else {
        return "end";
    }
}
```


```python
def make_answers(self, dlgpquery):
    ident = self.connector.initAnswers(self.ident, dlgpquery)
    return ans.Answers(self.connector, dlgpquery, ident)
```

```python
class Answers:

    def __init__(self, connector, dlgp, ident):
        self.connector = connector
        self.ident = ident
        self.dlgp = dlgp


    def __iter__(self):
        return self


    def __next__(self):
        res = self.connector.nextAnswer(self.ident)
        if res != "end":
            return (json.loads(res))
        else:
            raise StopIteration
```

Une architecture au-delà de Python



Fusion nécessaire de 2 stages