LarKC – a platform for large-scale Semantic Web Reasoning

Alexey Cheptsov

High Performance Computing Center Stuttgart

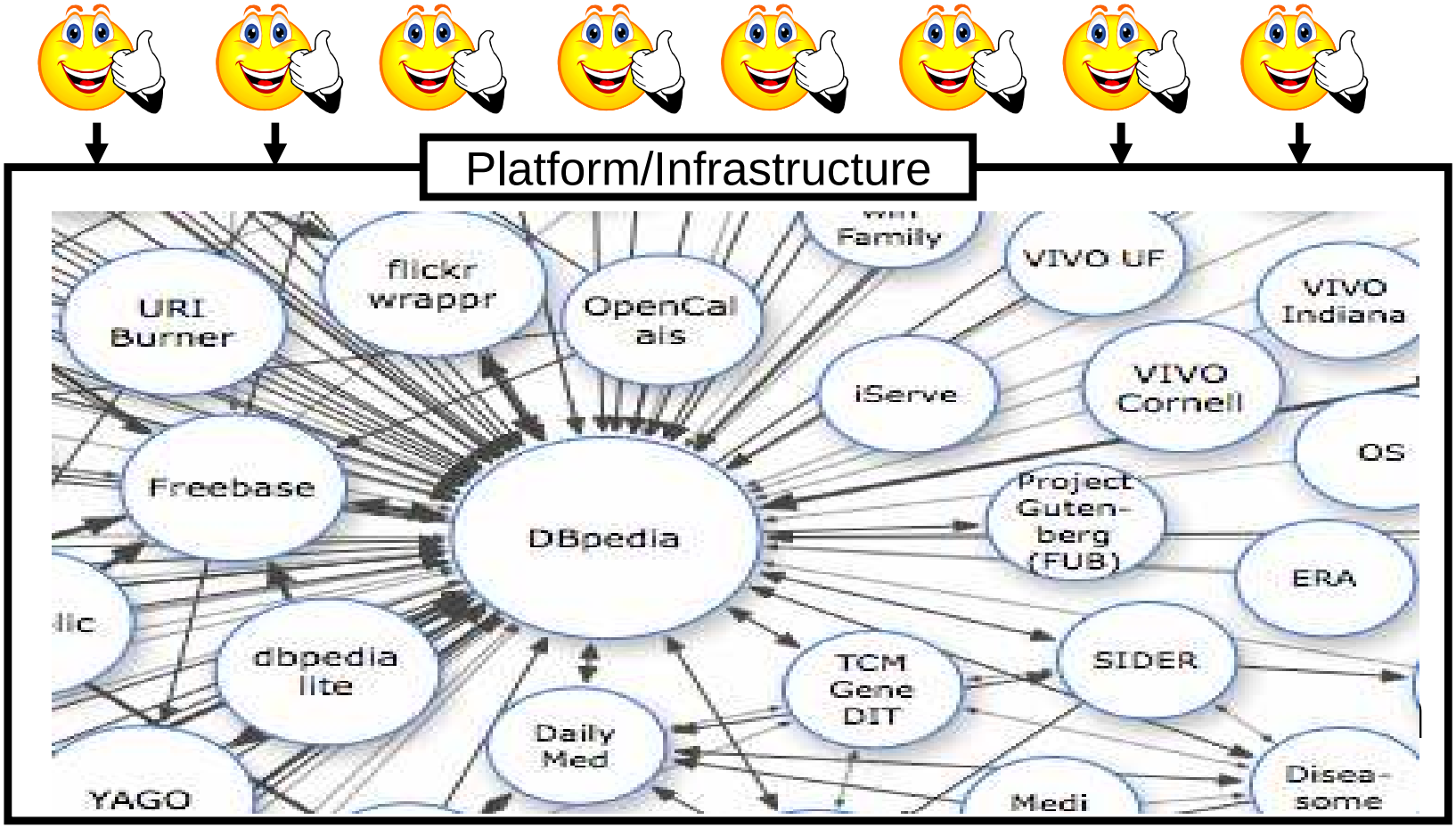
Outline

- **Semantic Web**. New challenges
- **LarKC**. A platform for web-scale Reasoning
  - Main motivation
  - Architecture overview
- **LarKC Platform**. Main subsystems
- **LarKC Software Redistribution**. LarKC@SourceForge
- **Conclusions**.

Large Knowledge Collider

1 *Semantic Web. New challenges*

Semantic Web enables machine-supported inferencing over the WWW-distributed data



Large Knowledge Collider

1 *Semantic Web. New challenges*

Web-scale reasoning

Gartner (May 2007):

"By 2012,
70% of public Web pages will have some level of semantic markup,
20% will use more extensive Semantic Web-based ontologies"

- Semantic Technologies at Web Scale?
  - 20% of 30 billion pages @ 1000 triples per page = **6 trillion triples**
  - 30 billion and 1000 are underestimated, imagine in 6 years from now…
  - data-integration and semantic search at web-scale?

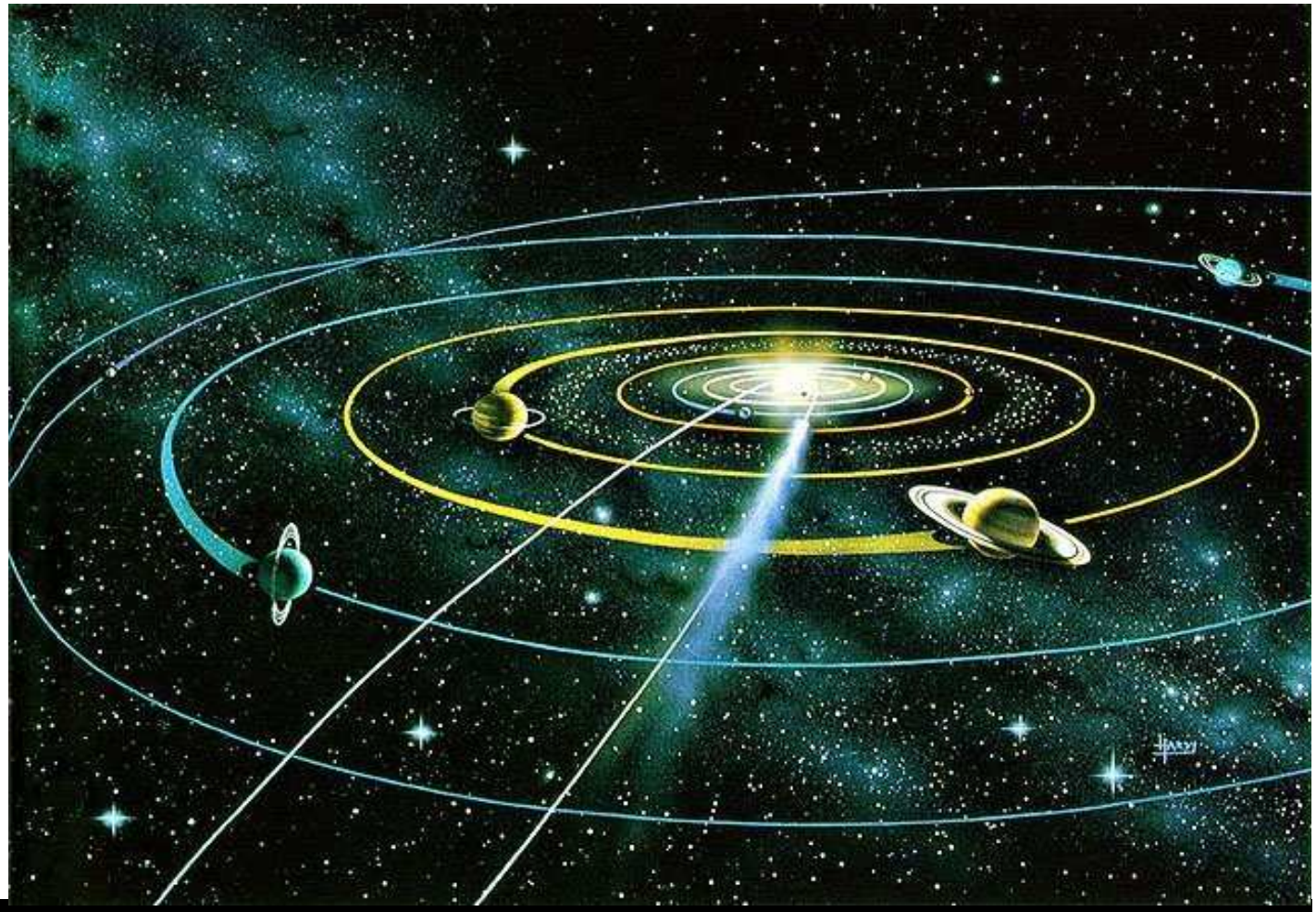
Large Knowledge Collider

1 *Semantic Web. New challenges*

Web-scale reasoning

1 Triple

~10^{14} Triples



Large Knowledge Collider

1 *Semantic Web. New challenges*

Data on the web scale

not only from

large numbers

- from performant data layer (OWLIM)
- from parallel deployment and execution of reasoners (IRIS)
- from load-balancing strategies
- …

but also from

interaction of multiple components
allowing for **incompletenes** and **anytime behaviour**
in the reasoning process

Large Knowledge Collider

❷ *The idea of LarKC*

An experimental platform for large-scale reasoning

requires

not only: deductive inference over given axioms

but also:

Reasoning + Search

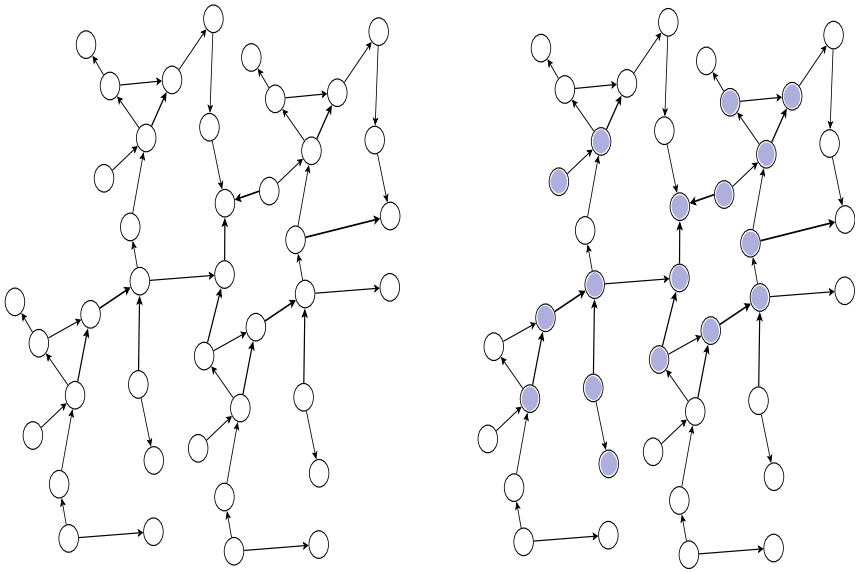
where do the axioms come from? (**IDENTIFY**)
which part of knowledge & data is required (**SELECT**)
when is an answer "good enough" or "best possible" (**DECIDE**)
non-deductive inference (inductive, statistical) (**REASON**)

Large Knowledge Collider

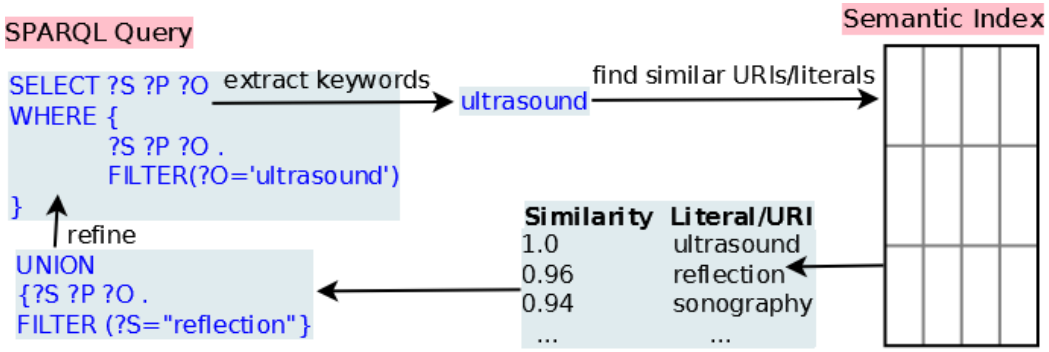
❷ *The idea of LarKC*

Statistical Semantics tasks

Subsetting

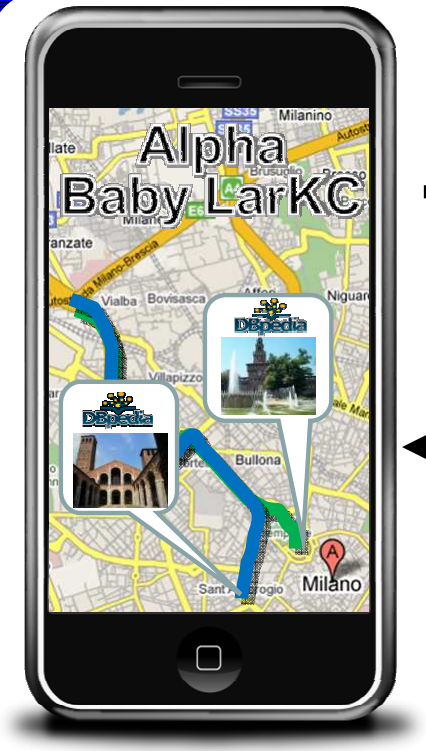


Query expansion



Large Knowledge Collider

❷ *The idea of LarKC*



REST request

JSON response

Urban Computing Environment

Interface

SPARQL query

SPARQL result

Pipelines Config.

LarKC platform

Request data

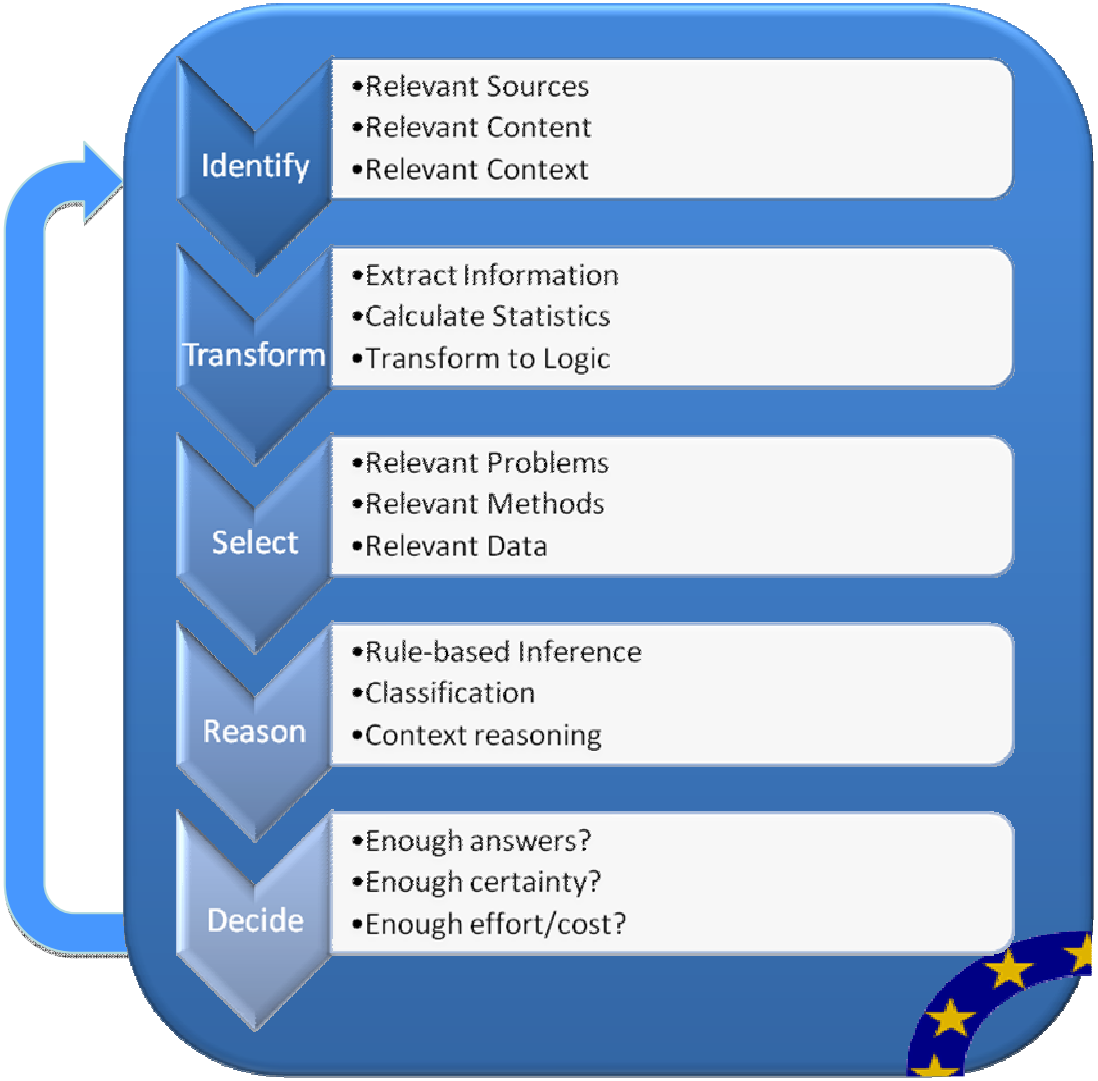
Data

PROBLEM: Which Milano monuments or events or friends can I quickly get to from here?

Data & Index | Events | Monuments | Streets

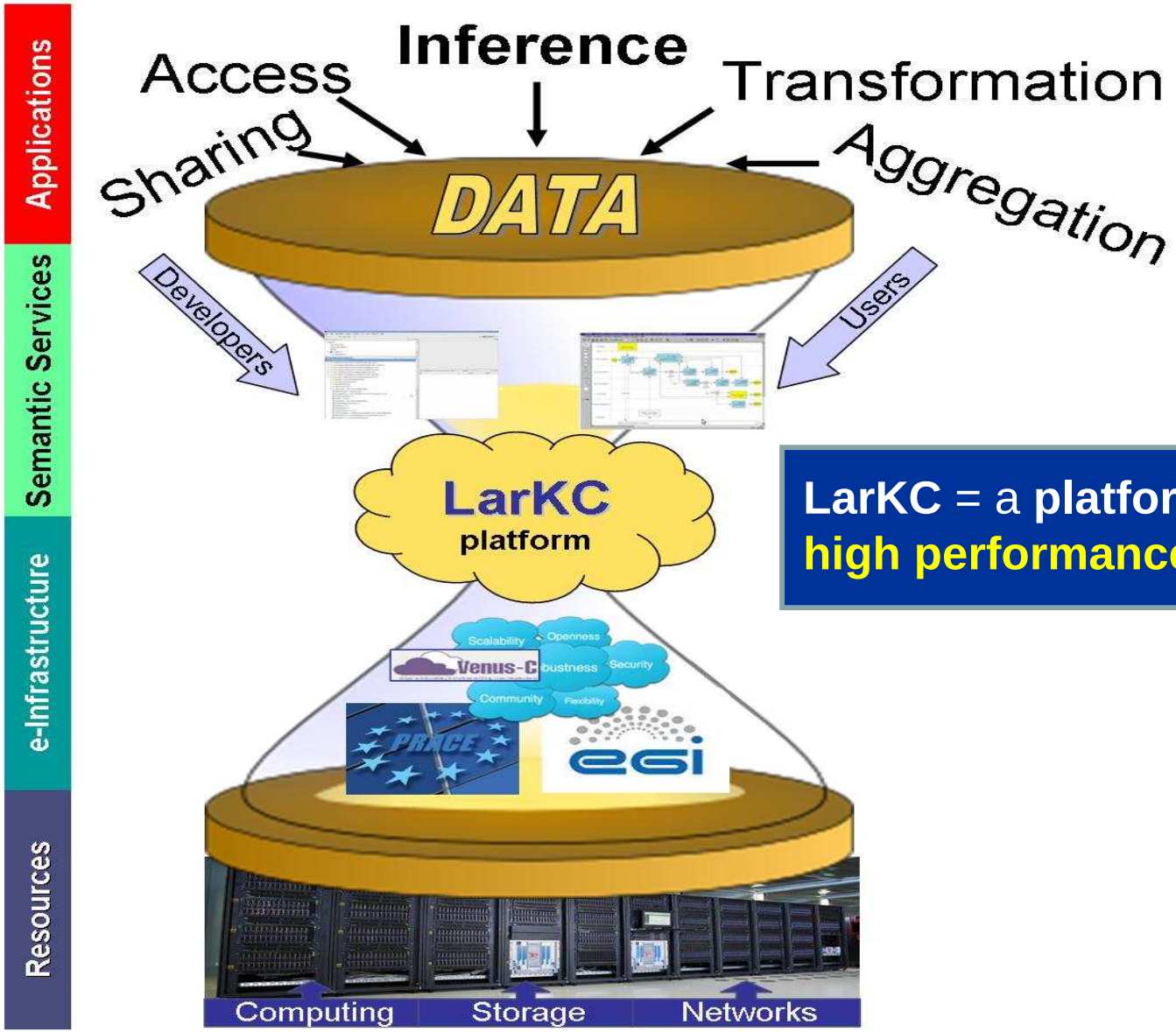
Large Knowledge Collider

❷ *The idea of LarKC*



Large Knowledge Collider

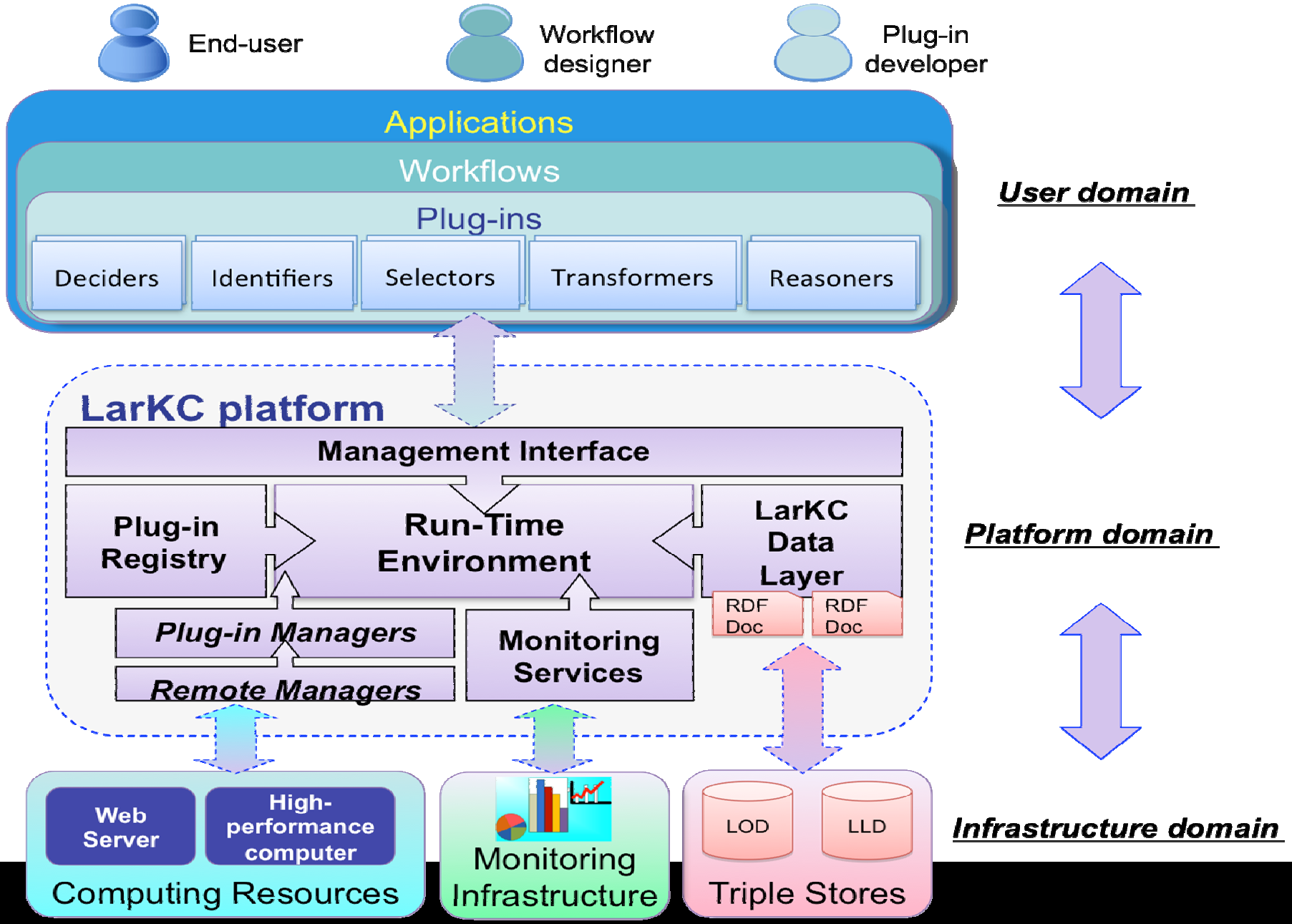
❷ *The idea of LarKC*



LarKC = a **platform** for **large scale, high performance reasoning**

Large Knowledge Collider

❸ *The architecture of LarKC*



Large Knowledge Collider

❸ *The architecture of LarKC*

Plug-in Registry	Management of plug-ins
Management Interface	Workflow specification and management
LarKC RTE	Initialisation and invocation of workflows
Plug-in Manager	Plug-in execution
Data Layer	Data management
Instrumentation	Monitoring of core components

❸ *The architecture of LarKC*

Plug-Ins

Plug-in description
- ❖Functional properties
- ❖Non-functional properties
- ❖WSDL description

Plug-in

```
+ URI getIdentifier()
+ SetOfStatements getPluginParameter()
+ QoSInformation getQoSInformation()
+ SetOfStatements invokeInternal()
```

- Plug-ins are assembled into workflows to realise a LarKC experiment or application
- Plug-ins are identified by a URI (Uniform Resource Identifier)
- Plug-ins provide metadata about what they do and what they consume/produce (Functional properties): e.g. type = Selecter
- Plug-ins provide information about their needs, including QoS information (Non-functional properties): e.g. Throughput, MinMemory, etc.
- Plug-ins are equipped with functionalities for data caching and messaging

Large Knowledge Collider

❸ *The architecture of LarKC*

New plug-in API – both input and output are represented in RDF

Automatic parallelization (thread-based) on the statement level

Data caching, instrumentation and event processing

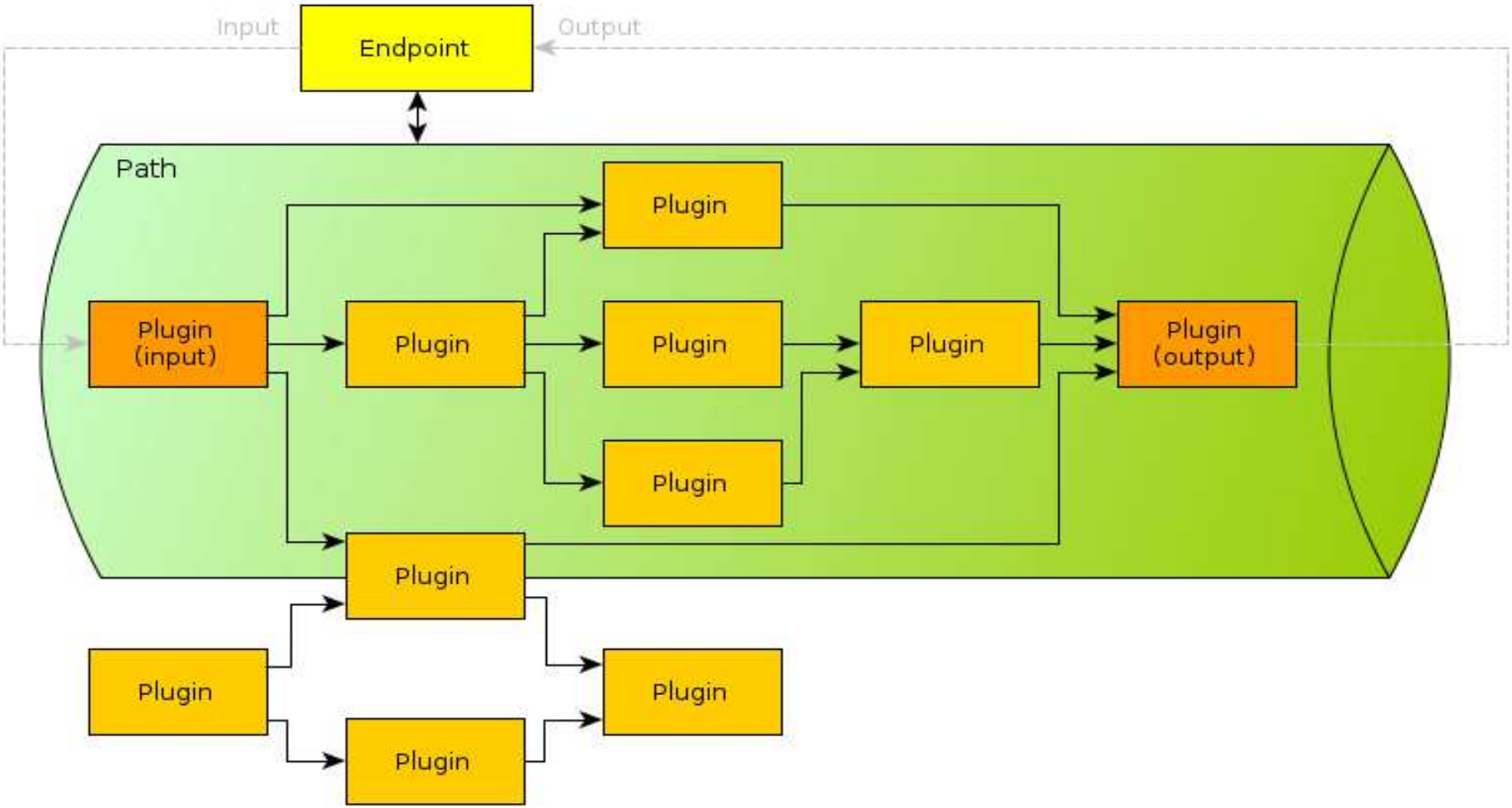
Maven based build system

- Improved version controlling and dependency management
- Simplified procedure of new plug-in creation

Large Knowledge Collider

❸ *The architecture of LarKC*

Workflows



Workflow Ontology

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix larkc: <http://larkc.eu/schema#> .

# Define two plug-ins
_:plugin1 a <urn:eu.larkc.plugin.identify.TestIdentifier> .
_:plugin2 a <urn:eu.larkc.plugin.transform.TestTransformer> .

# Connect the plug-ins
_:plugin1 larkc:connectsTo _:plugin2 .

# Define a path to set the input and output of the workflow
_:path a larkc:Path .
_:path larkc:hasInput _:plugin1 .
_:path larkc:hasOutput _:plugin2 .

# Connect an endpoint to the path
_:ep a <urn:eu.larkc.endpoint.sparql> .
_:ep larkc:links _:path .
```

Management Interface

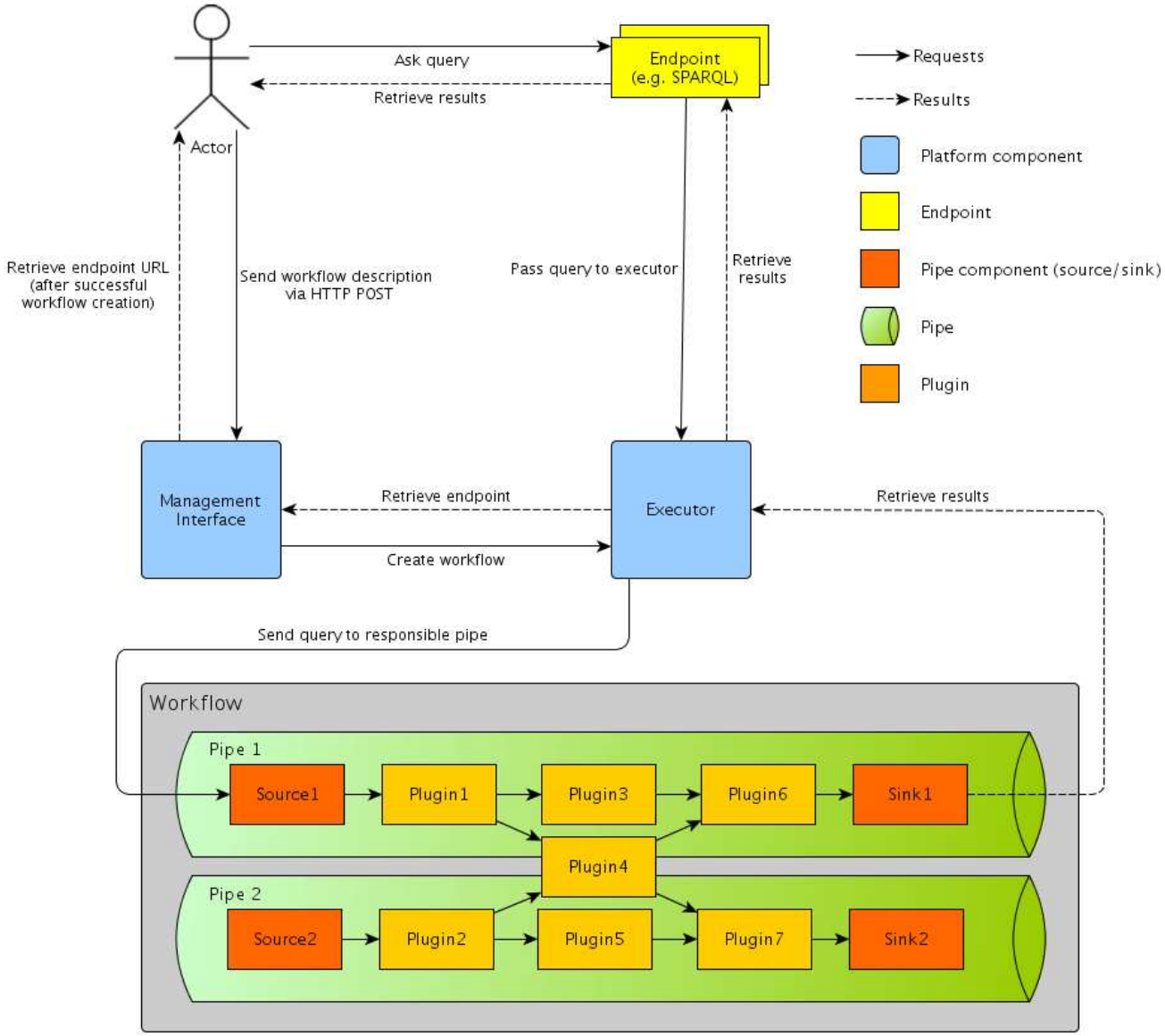
- implemented as a RESTful service
- allows the submission/monitoring/deletion of workflows
- supports N3 and RDF/XML workflow representations
- equipped with additional features for
  - retrieving all registered plug-ins from the platform
  - retrieving configuration templates of all supported remote hosts
- accessible via a simple HTML interface

Large Knowledge Collider

❸ *The architecture of LarKC*

Run-Time Environment

- comprises the
  components (
- responsible fo
  workflows
- collaborates w
  Managers
- takes care of
  loops, workflo

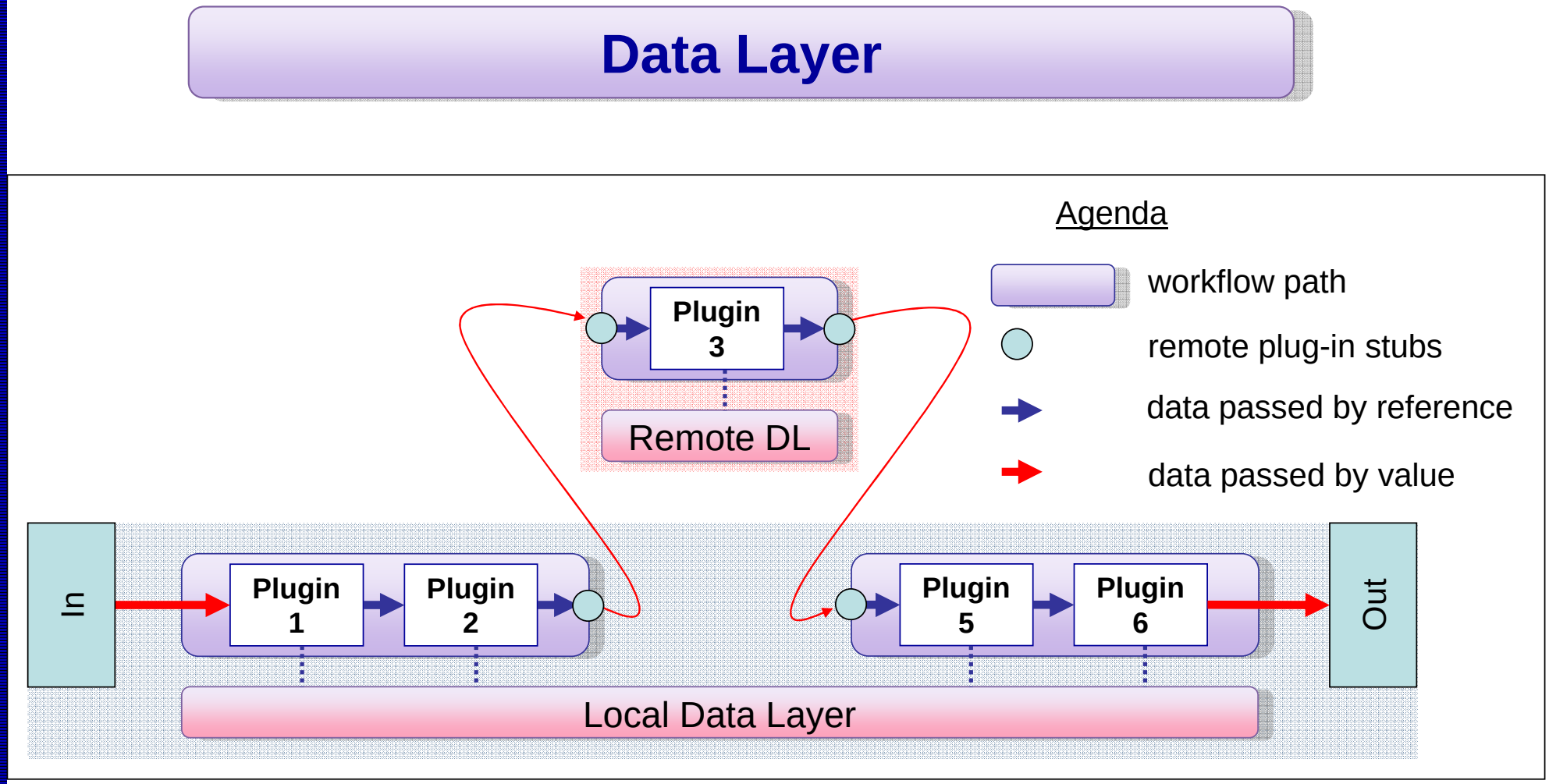


Data Layer

- The Data Layer supports LarKC plug-ins:
  - storage, retrieval and light-weight inference on top of large volumes of data
    - Reference implementation of ORDI data model
    - Retrieval data exposed via standard SPARQL endpoints
    - Configurable forward-chaining reasoning OWL2-RL
  - automates the exchange of RDF data by reference and by value
  - offers other utility tools to manage data (e.g. data streaming, querying remote data)
  - can run in cluster mode to improve resilience and provide scalable query answering

Large Knowledge Collider

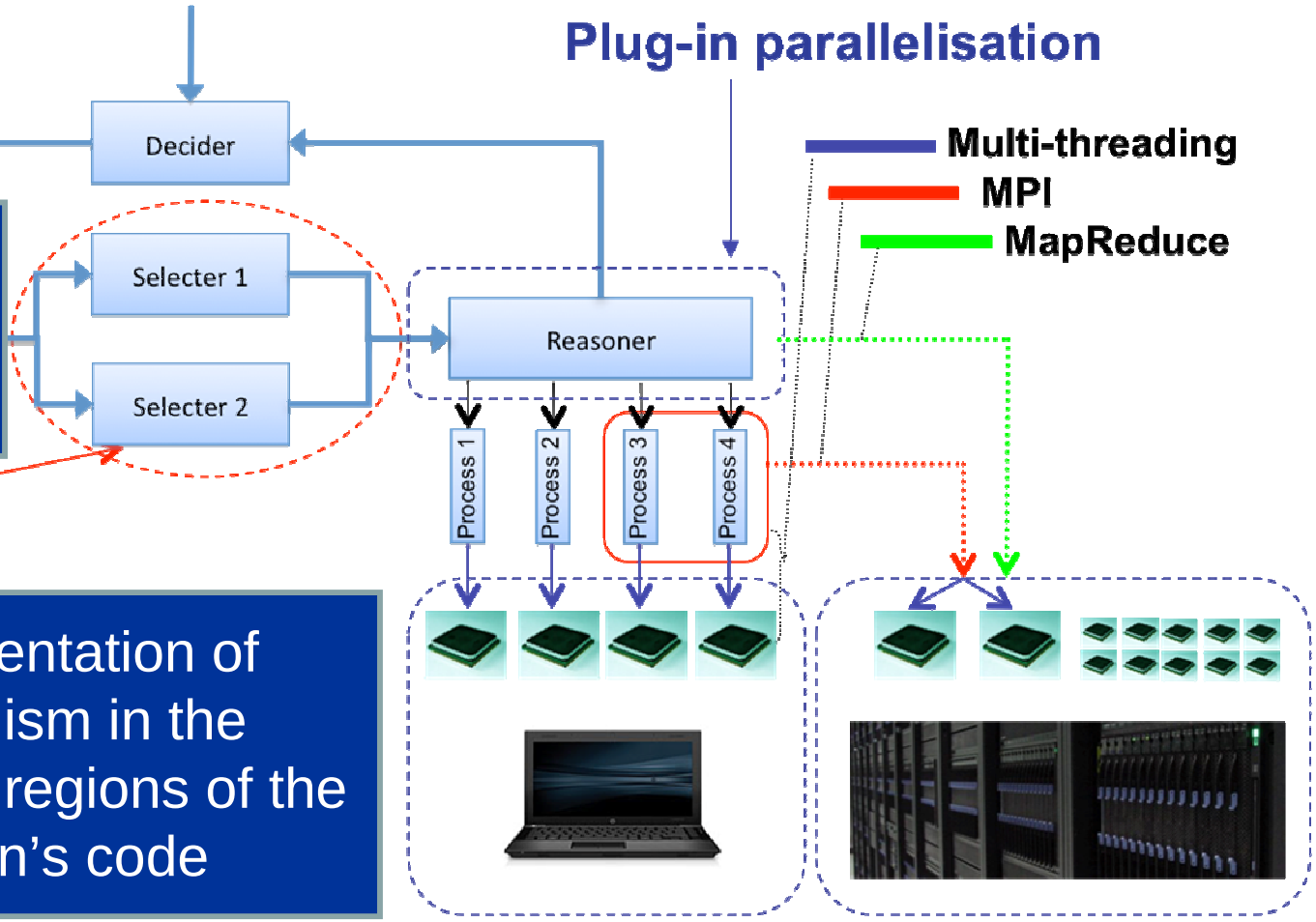
❸ *The architecture of LarKC*



Large Knowledge Collider

❸ *The architecture of LarKC*

Parallelisation



Running multiple instances of the same plug-in simultaneously

Workflow branching

Implementation of parallelism in the concurrent regions of the plug-in's code

Large Knowledge Collider

❸ *The architecture of LarKC*

User-level Tools

LarKC WorkFlow Designer | New | Load | Initialize | Show Description | Execute Query | About

LarKC Components

Plugins
- TestRemoteIdentifier
- TestIdentifier
- TestTransformer
- TestDecider
- RISearchPlugin
- SpanningWorkflowPlugin
- LLDSelector
- QueryExpansionPlugin
- SparqlQueryEvaluationReasoner

Path
- Input
- Output

Hosts
- GlobusToolkit4
- Local
- Tomcat
- SshWitUserName
- SshWithPublicKey

Local
click to edit...

Input

RISearchPlugin
urn:eu.larkc.plugin.RISearchPlugin
Parameters
0-4/lld1-docs-vectors.sspace" remove
larkc:nrOfWords "20" remove
Add

QueryExpansionPlugin
urn:eu.larkc.plugin.QueryExpansionPlugin
Parameters
Add

LLDSelector
urn:eu.larkc.plugin.LLDSelector
Parameters
Add

SparqlQueryEvaluationReasoner
urn:eu.larkc.plugin.SparqlQueryEvaluationReasoner
Parameters
Add

Tomcat
click to edit...
larkc:hostType
larkc:JEE
larkc:jeeUri
<http://localhost:8080>

Output

Query
SELECT ?s ?p ?o WHERE { { ?s ?p ?o . ?s ?p "asthm

Endpoints
Endpoint Sparql

Minimap

Result
Result: download RDF

```
<xmp><?xml version="1.0" encoding="U
<sparql xmlns="http://www.w3.org/200
  <head>
    <variable xmlns:ns0="http://www.
    <variable xmlns:ns0="http://www.
    <variable xmlns:ns0="http://www.
  </head>
  <results>
    <result>
      <binding xmlns:ns0="http://www
        <uri>http://linkedlifedata.c
      </binding>
      <binding xmlns:ns0="http://www
        <uri>http://www.w3.org/2004/
      </binding>
      <binding xmlns:ns0="http://www
        <literal>asthma</literal>
      </binding>
    </result>
    <result>
      <binding xmlns:ns0="http://www
        <uri>http://rdf.freebase.com/
      </binding>
      <binding xmlns:ns0="http://www
        <uri>http://rdf.freebase.com/
      </binding>
```

A B C D E F G

Large Knowledge Collider

4 *The LarKC redistributions*

Release 2.5

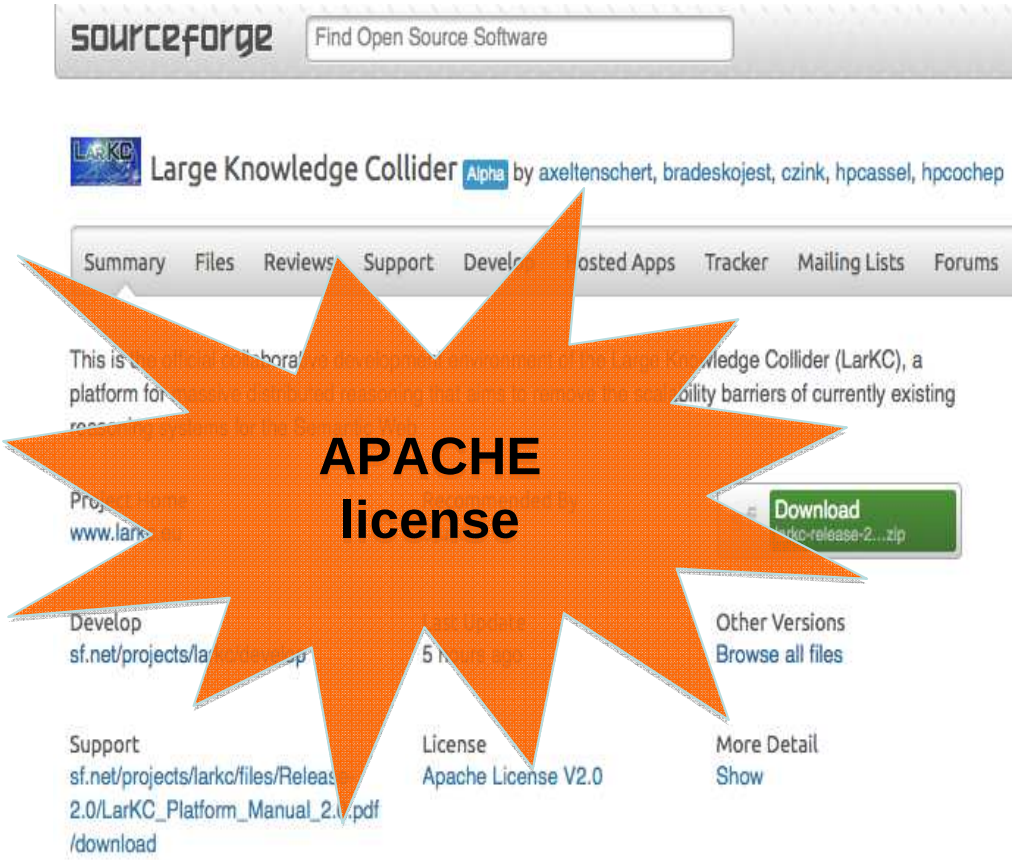
- Plug-in / workflow descriptions and plug-in parameter are in RDF
- Separation of workflow specification and execution
- Integration of various endpoints (e.g. SPARQL) and applications
- Workflow branching, conditional loops, splits / merges of data flow
- (Remote) plug-in execution, parallelisation support, anytime behaviour
- Data caching, instrumentation and event processing
- Data storage, data streaming, parallel request handling

Large Knowledge Collider

④ *The LarKC redistributions*

LarKC@SourceForge

- Software releases
  - Platform binaries
  - User and developer guide
- Source code
  - Maven based build system
  - Subversion repository
- Additional Support Tools
  - Mailing lists
  - Discussion forums
  - Tracker



Large Knowledge Collider

④ *The LarKC redistributions*

Plug-In Market Place

Plug-In Marketplace

Plug-ins

Name	Platform Version	Type	Description	www	Download	Contact person
Base-Line Full-Text Search Selecter	1.0	select				
CRION reasoner	Alpha release	reason				
DIGReasoner	1.0	reason				
EventIdentifier	1.0, 1.1	identify				
GWADecider	1.1	decide				
GWASIdentifier	1.1	identify				
GWASQueryTransformer	1.1	transform				
Information Retrieval Selecter	1.0	select				
Interest-Based Reasoner	Alpha Release	reason				
Interest-Based Selecter	1.0	select				
Keyphrase Selecter	1.0	select				
OWLAPI Reasoner	1.0	reason				
PION Reasoner	1.0	reason				
RDF2MatrixTransformer	1.0	transform				
Random Indexing Reasoner	1.0	reason				
RandomIndexingDecider	1.1	decide				
RandomIndexingIdentifier	1.1	identify				

> 25 Plug-ins available

Large Knowledge Collider

5 *Conclusions*

LARKC is …

… A platform for rapid semantic web application prototyping

- based on light-weight flexible components (plug-ins)
- easy integration in a workflow
- powered by platform's features (data layer, remote execution, etc.)

… An infrastructure that enables web scale and high performance

… A friendly developer team looking forward to your requests