Use Case Objective

Design and implementation of a prototypal Infrastructure for data-intensive seismology, adopting the ADMIRE framework.

Advanced Data Mining and Integration Research for Europe http://www.admire-project.eu/

ADMIRE Goals

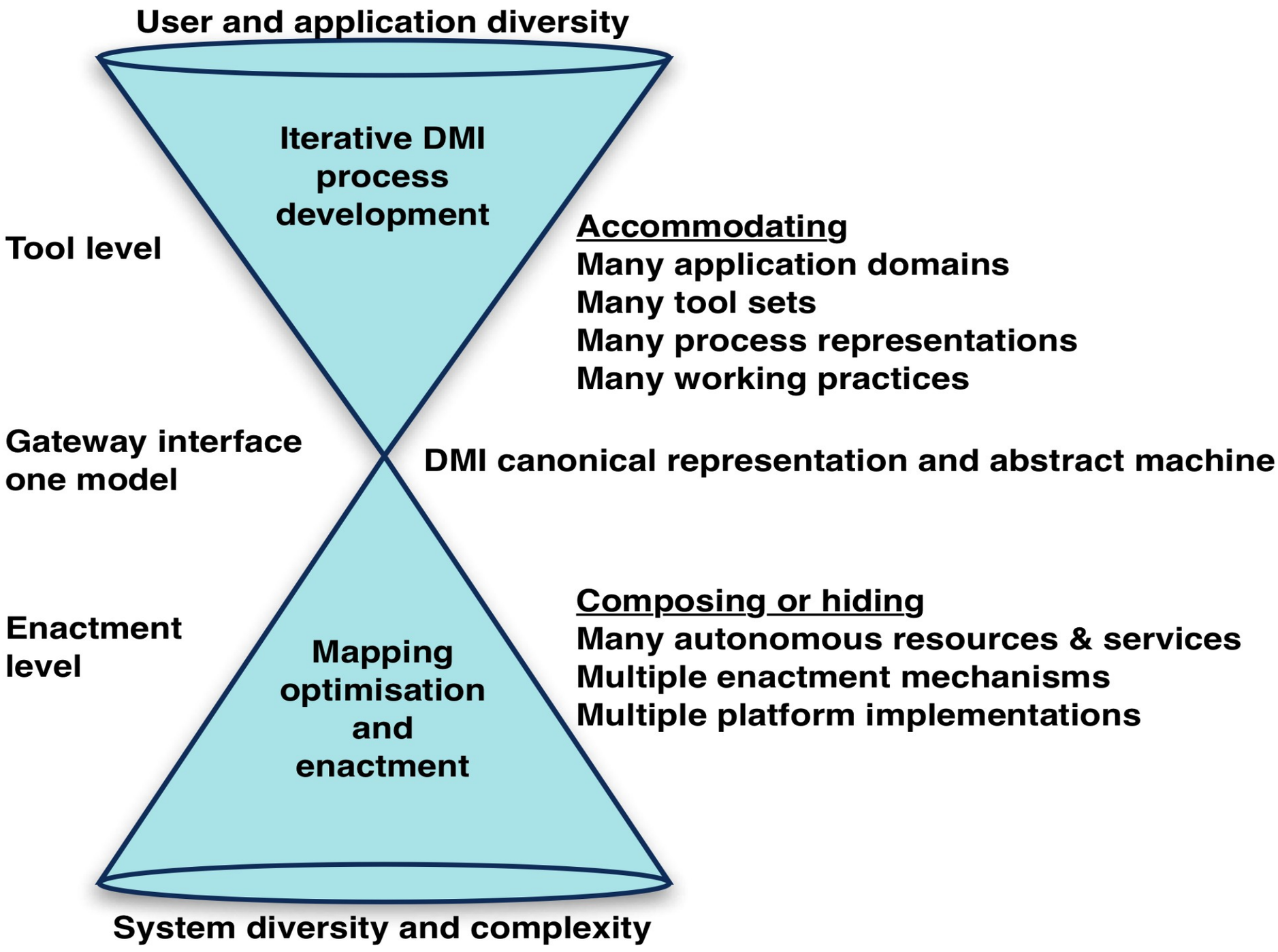
Accelerate access to and increase the benefits from data exploitation

Deliver consistent and easy to use technology for extracting information and knowledge;

Cope with complexity, distribution, change and heterogeneity of services, data, and processes, through **abstract view of data mining and integration**

Provide power to users and developers of data mining and integration processes.

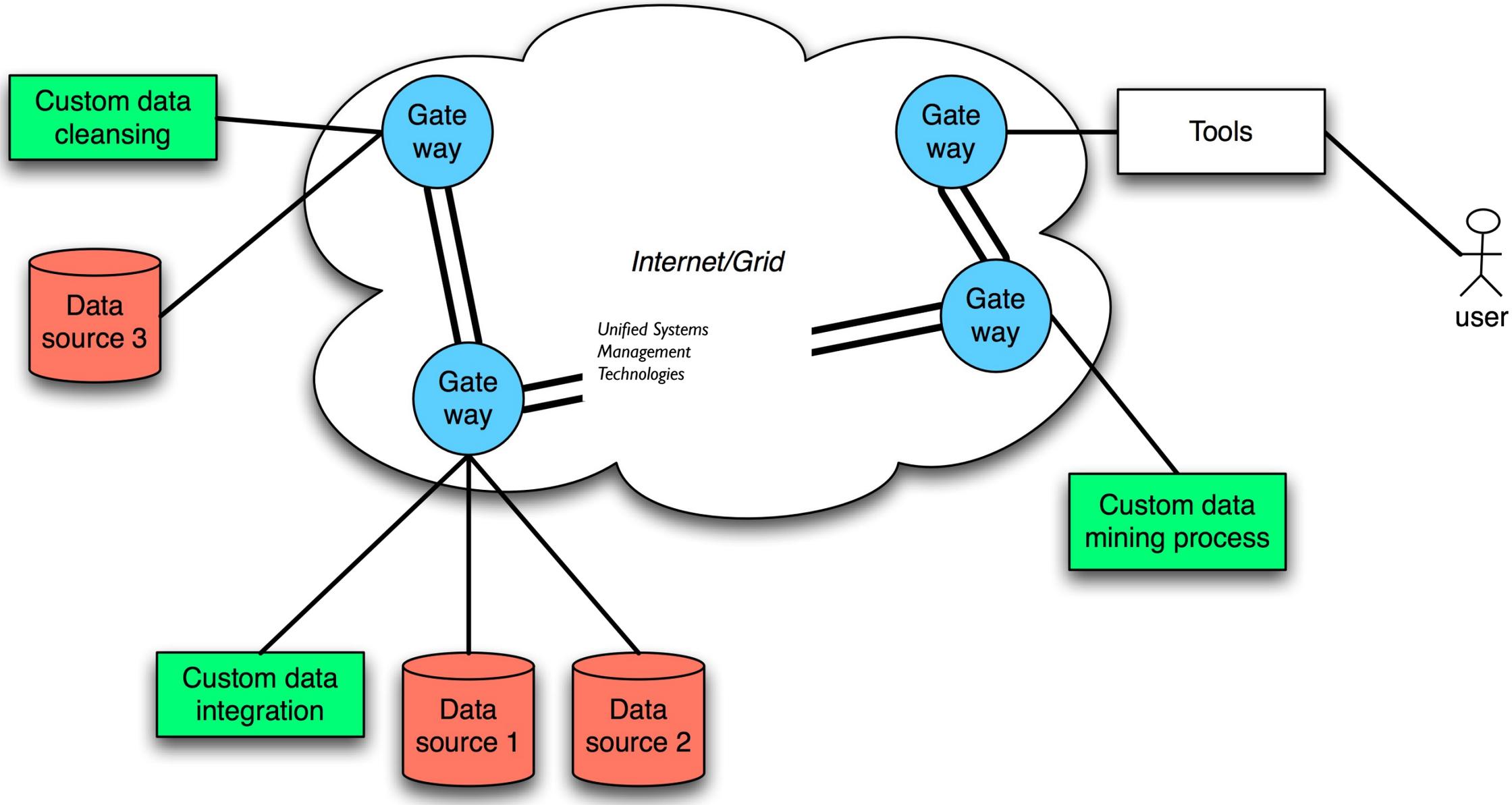
The Hourglass Model



Separation of Concerns

	Tools	Formalisms	Data-Flow Execution
Domain Experts			
Data Mining Experts			
Data Aware Distributed Computing Experts			

ADMIRE's Distributed Infrastructure



DISPEL

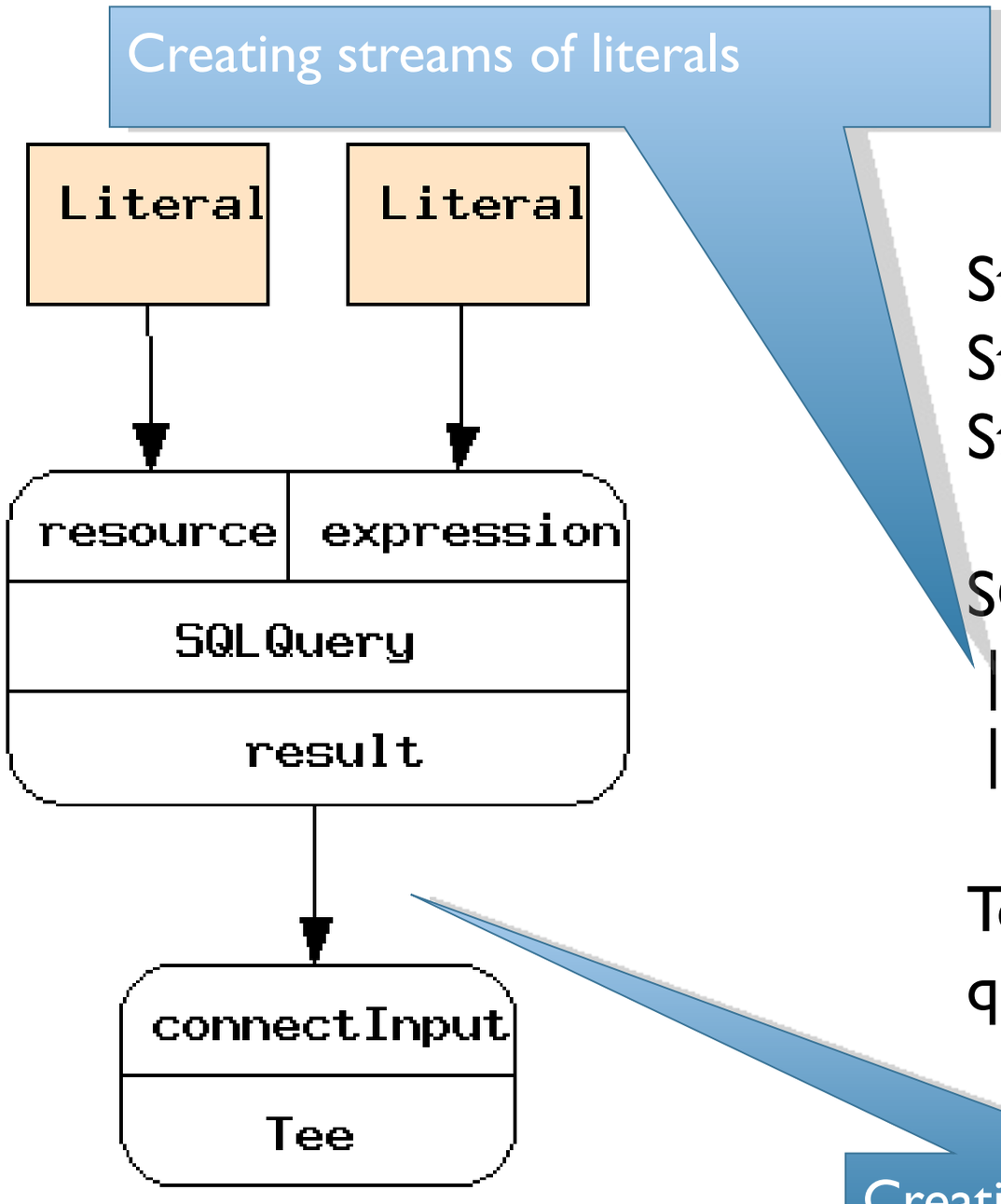
Intensive Systems Process Engineering Language

ADMIRE is offering a solution which is supported by a formal language as the end-user development interface for the underlying infrastructure.

DISPEL allows composition of the data analysis steps into composite scenarios that can be considered as a set of, to some degree, independent workf ows

DISPEL represent the connection point of complex application requests and complex enactment systems

DISPEL: Simple Example




```
String sql1 = "SELECT * FROM some_table";
String sql2 = “SELECT * FROM table2”;
String resource = "128.18.128.255";

SQLQuery query = new SQLQuery;
 |- sql1, sql2 -| => query.expression;
 |- resource -| => query.resource;

Tee tee = new Tee;
query.result => tee.connectInput;
```

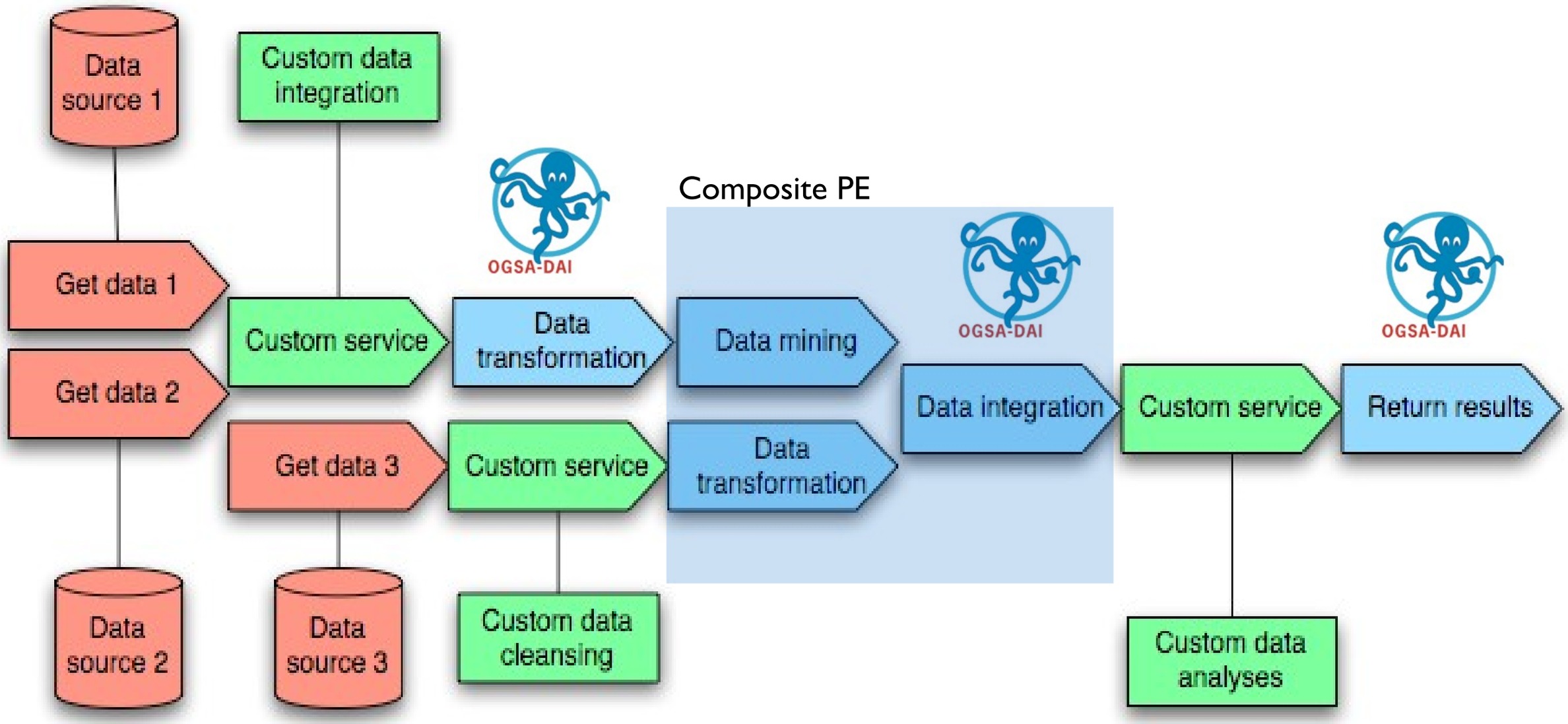
Creating connections

Processing Element

The building blocks of a DISPEL workflow are called Processing Elements (PEs)

The execution of the PEs is optimised and parallelised through The ADMIRE's computational infrastructure, which relies on the OGSA-DAI data-streams processing technology.

Processing Elements Workflow

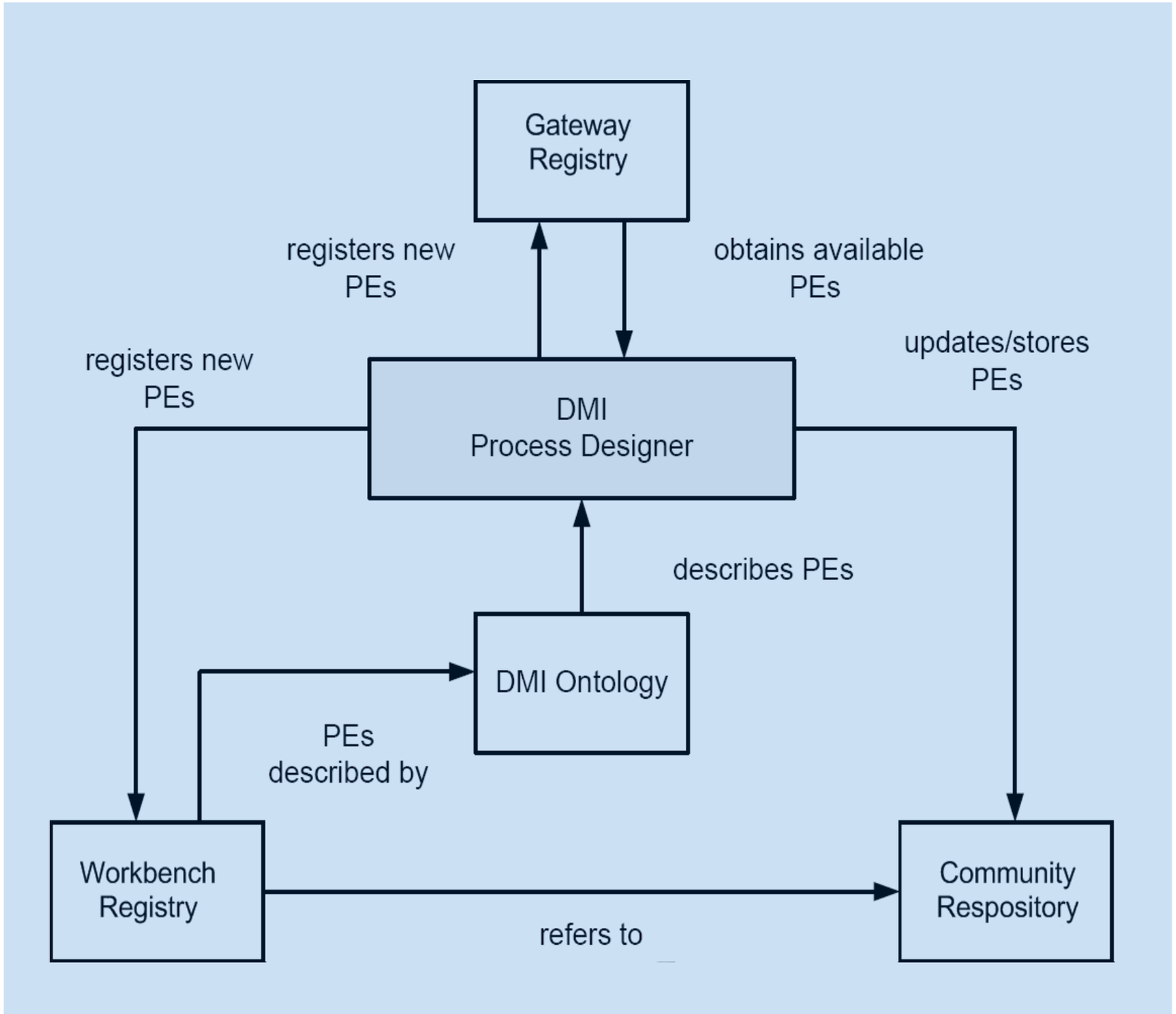


Processing Elements Registration

Goals:

Keep track of the processing elements (PE) created by the users of the ADMIRE tools

Define aspects of the computational context of a DMI stage

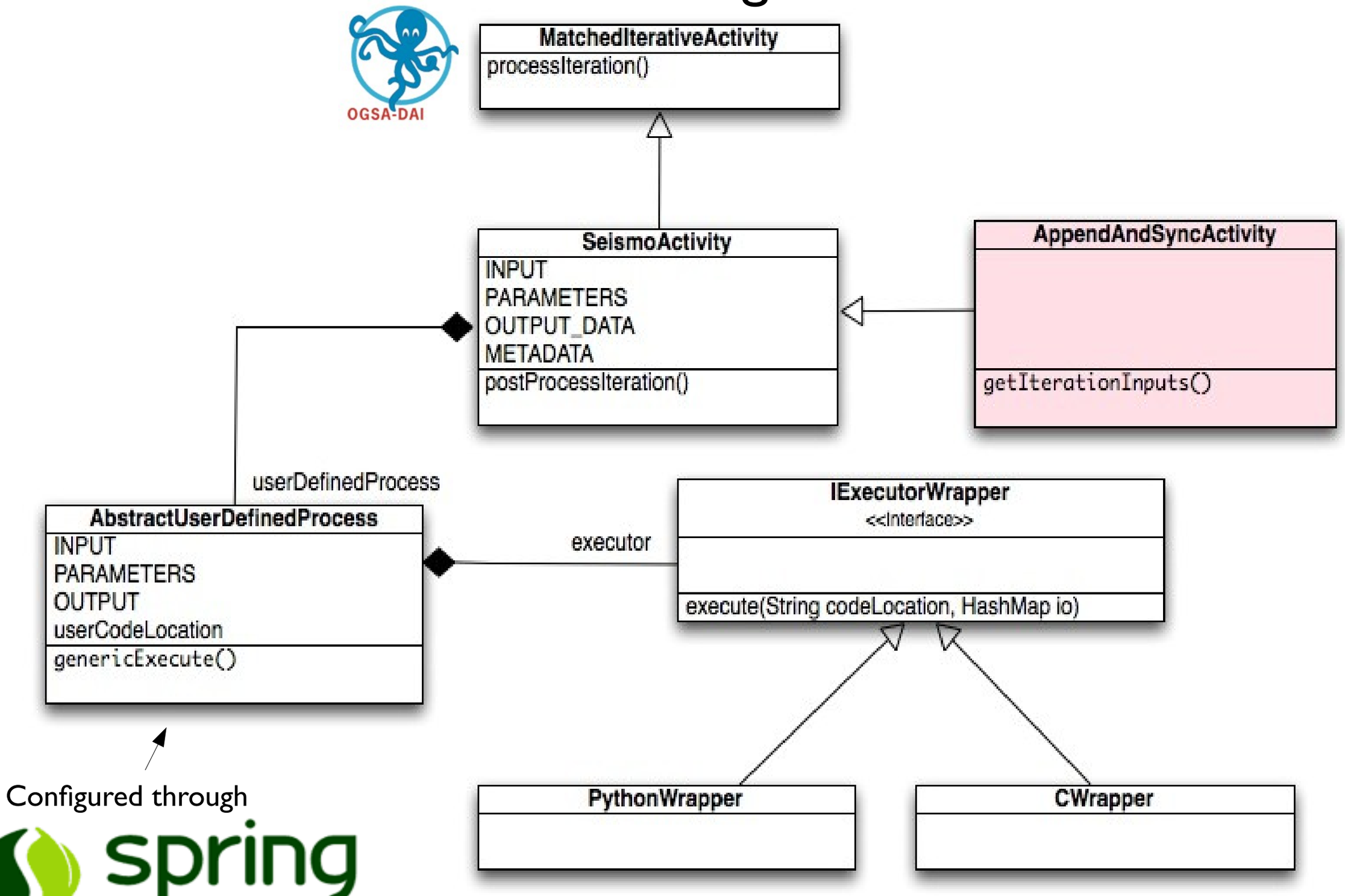


Seismological Processing Element Implementation

Requirement: The seismology scientists expect to develop and use their own analysis code, written with their favorite programming language and API

Integration of the domain specific Python API (ObsPy) into OGSA-DAI Activities, as the low level implementation for the seismological analysis PEs.

Framework for UserDefined OGSA-DAI Processing Activities

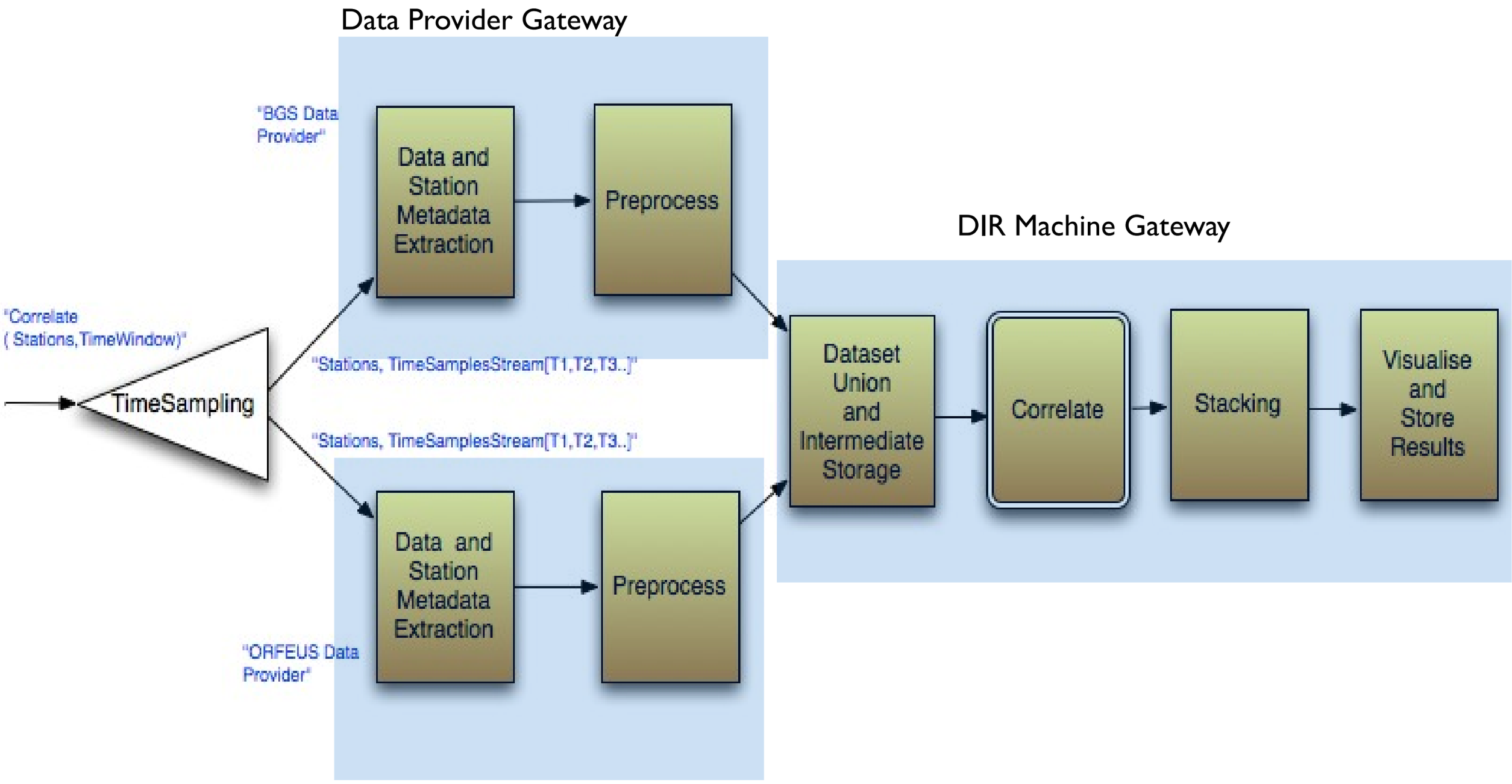


Configured through

spring source

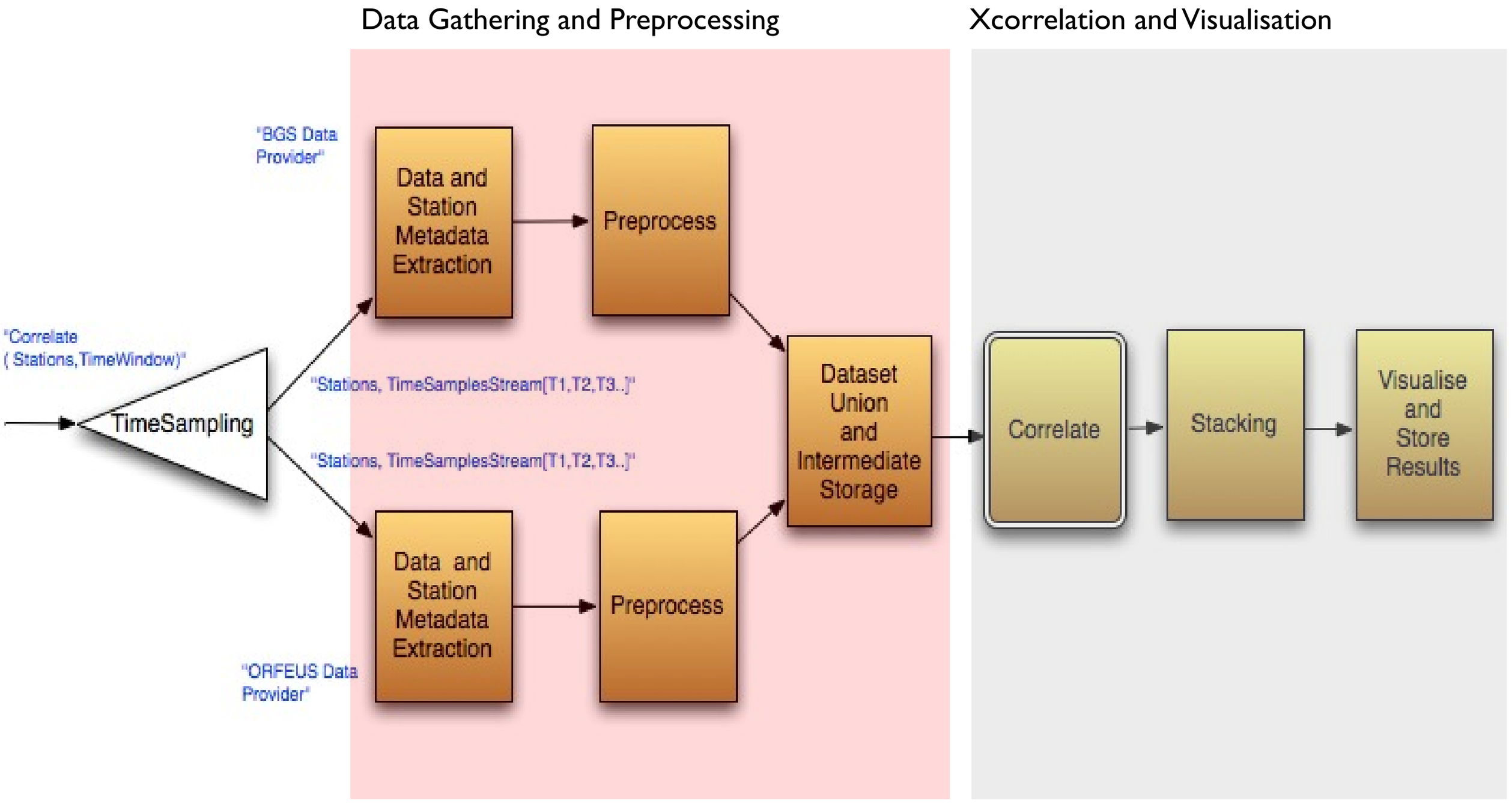
Python Seismological Lib: ObsPy

Ambient Noise Xcorrelation, Implementation overview



Data Provider Gateway

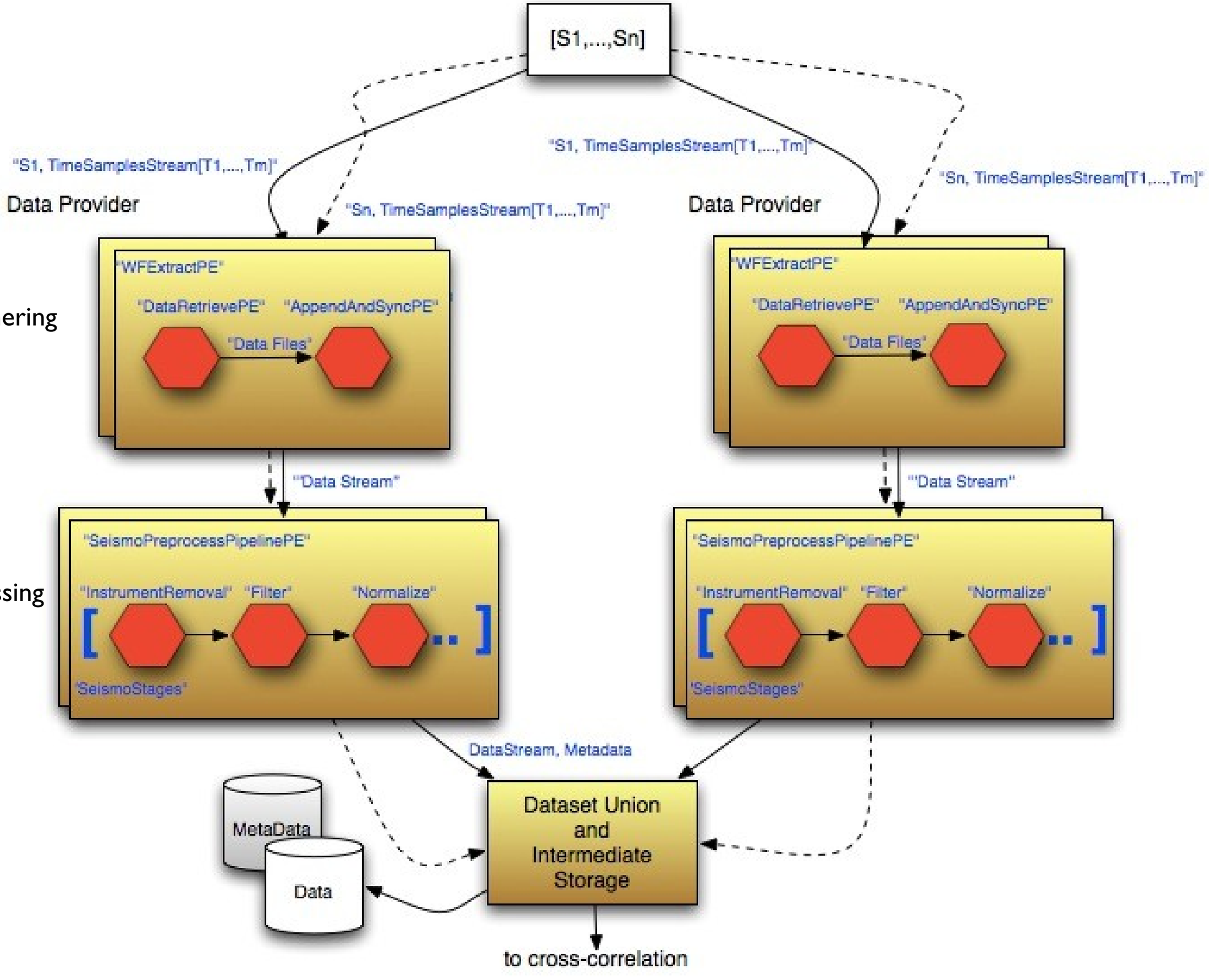
Ambient Noise Xcorrelation, Implementation overview



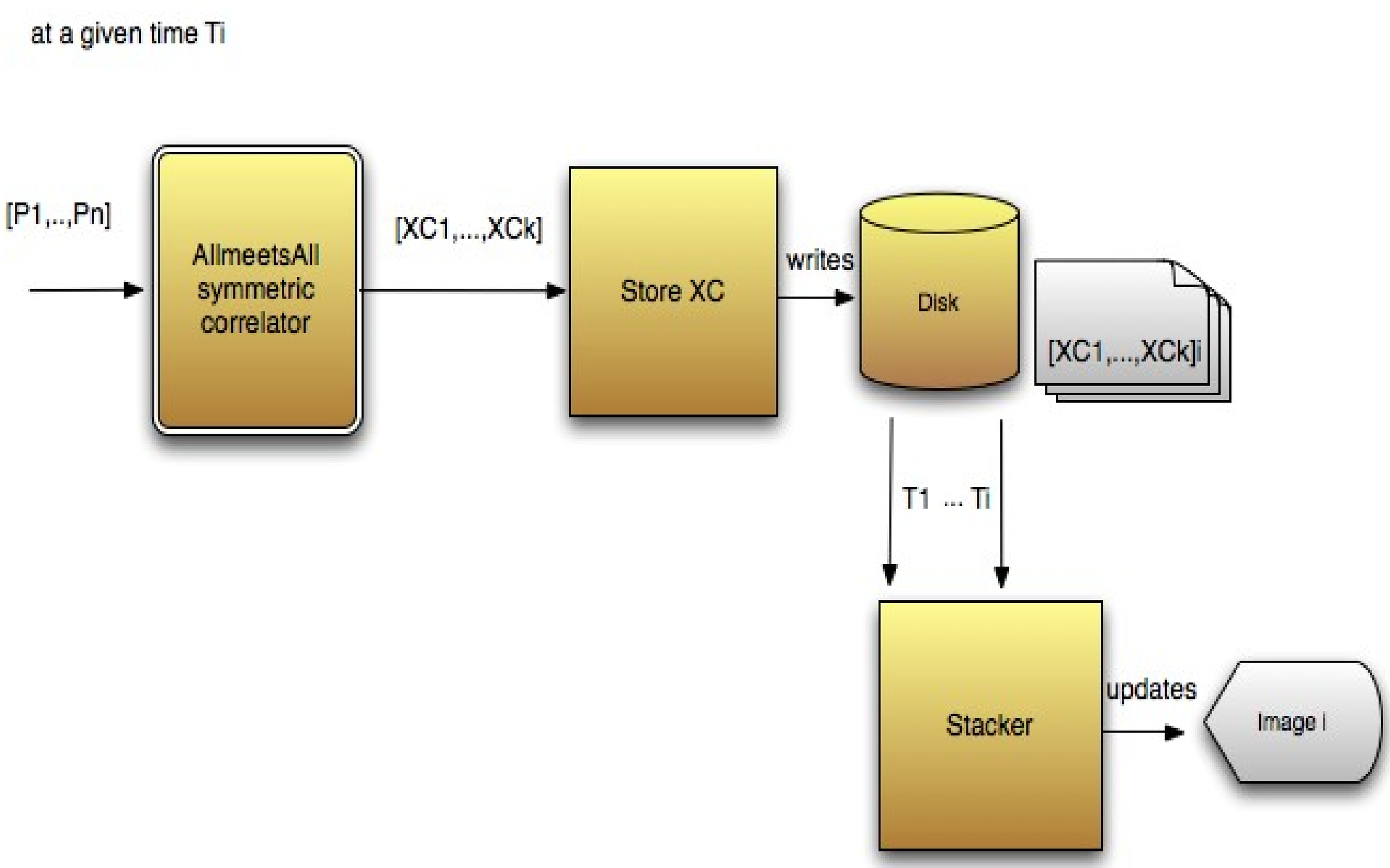
Data Gathering and Preprocessing

Data Gathering

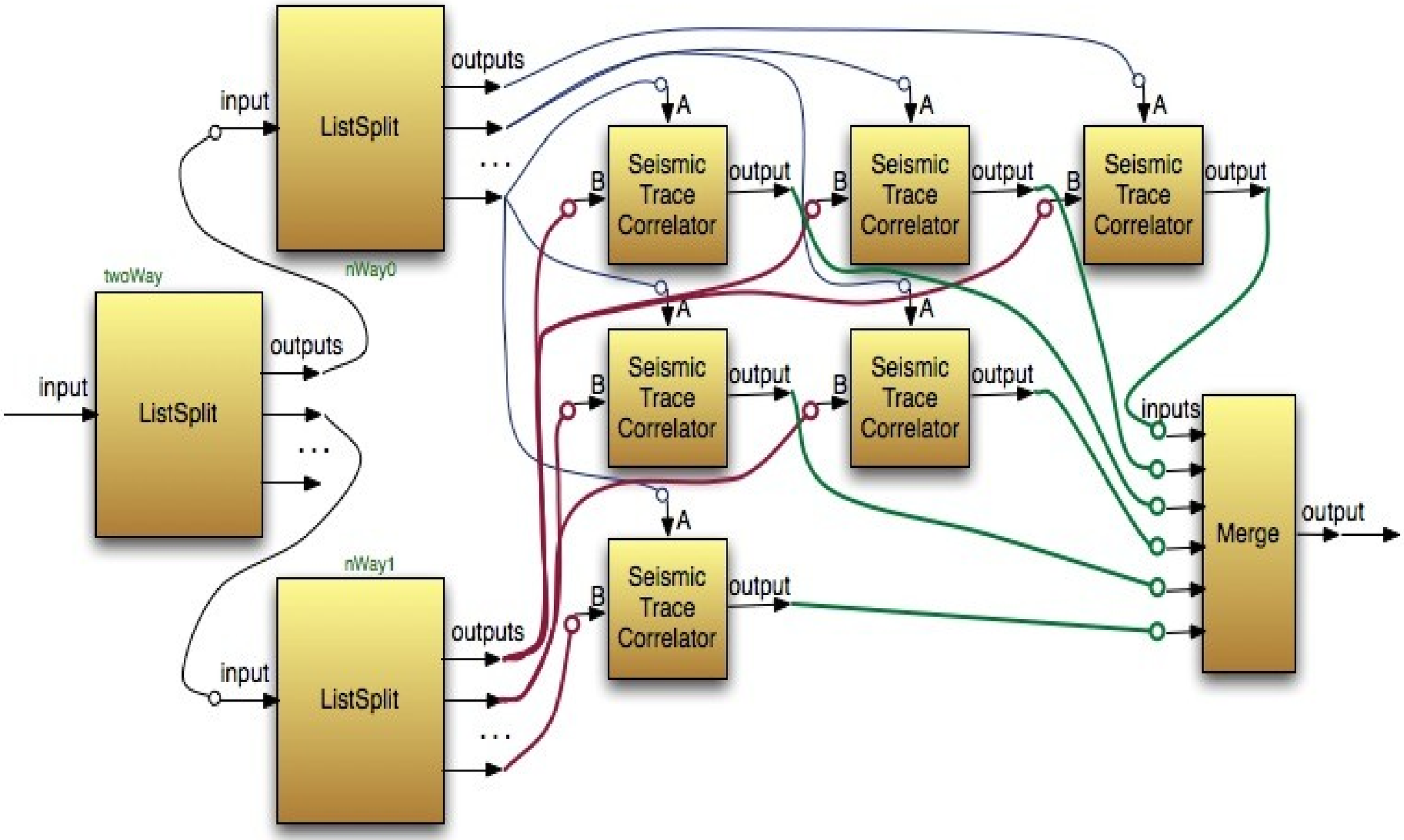
Preprocessing Pipeline



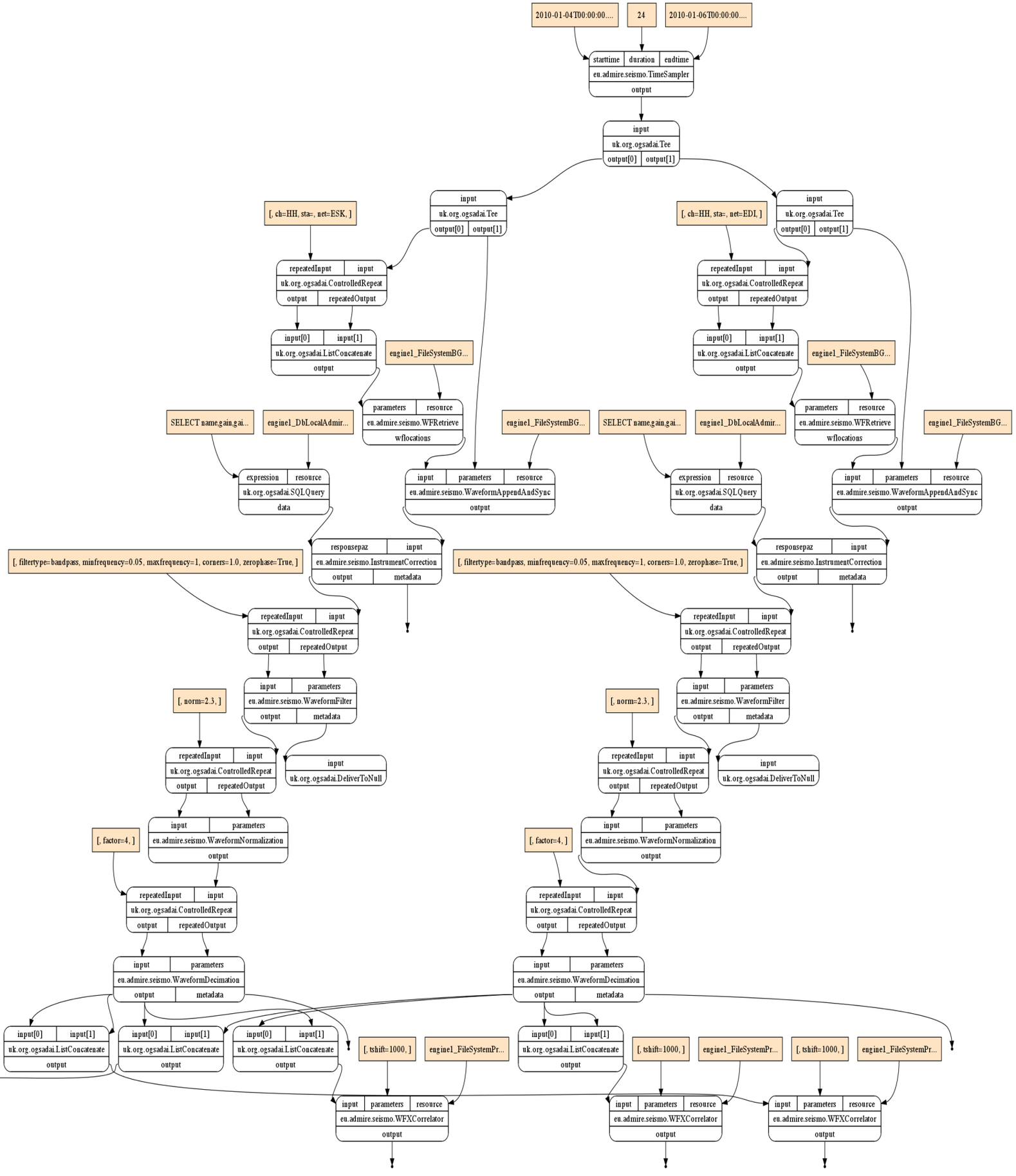
Xcorrelation and Visualisation



AllMeetsAll



All Together in DISPEL



Suggested Future Works

Dynamic integration of User-Def ned PEs:
Easier deployment and validation of user's analysis code within a deployed ADMIRE infrastructure
(Based on the RapidSeis idea: http://research.nesc.ac.uk/node/586).

Better support for data streams I/O within the ObsPy API
(Currently, serialised Python Objects are exchanged among PEs)

Optimisation of parallel workf ow's subgraphs execution, depending on the available computing resources