15/09/2015

Deploying a PDL based complete web service

Carlo Maria Zwölf et PDL contributors

PDL: a quick overview

- Parameter Description Language (PDL) is intended to be a lingua franca of parameters:
  - Describes params in a sufficient detail to allow workflow tools to check if parameters can be "piped" between services
    - Physical Properties (Nature, Meaning, unit, precision,...)
    - Computing (Numerical Type, UCD, SKOS concept)
  - Also has capabilities do describe constraints on parameters
    - Physical constraints
    - Arbitrary (including mathematical) constraints
- **Not** a description of parameters "values" (cf. UWS).

PDL Uses				
Generic software components can be 'configured' by a PDL description for creating quickly fully interoperable new services				
Server exposing services as web services	User Interface (for interaction with PDL services)	Auto Generation of checking algorithms from description	Workflow plugin (for WF interaction with PDL services)	A priori computation of interoperability graphs

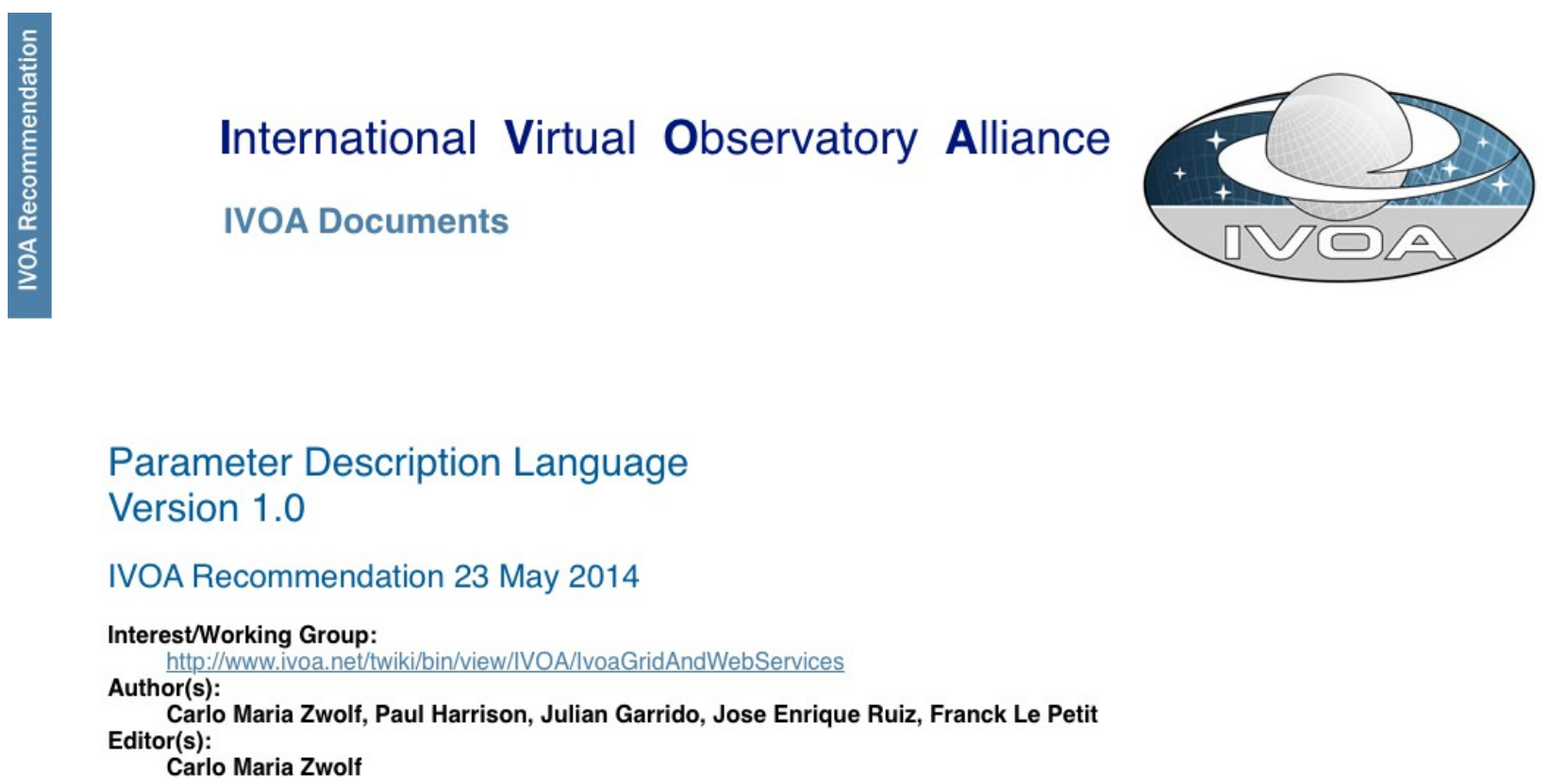
PDL Principles

- The language is based on a *Data Model*
- Each object of the DM corresponds to a syntactic element:
  - Sentences are made by building object-structures;
  - Each sentence is interpreted by a computer by parsing the sentence-related object-structure;
  - With no loss of generality → the DM is fixed into an XML schema.

PDL Principles

- The language is based on a *Data Model*
- Each object of the DM corresponds to a syntactic element:
  - Sentences are made by building object-structures;
  - Each sentence is interpreted by a computer by parsing the sentence-related object-structure;
  - With no loss of generality → the DM is fixed into an XML schema.

- PDL became an IVOA recommendation on May 2014 (Madrid Interop): http://ivoa.net/documents/PDL


IVOA Recommendation

International Virtual Observatory Alliance

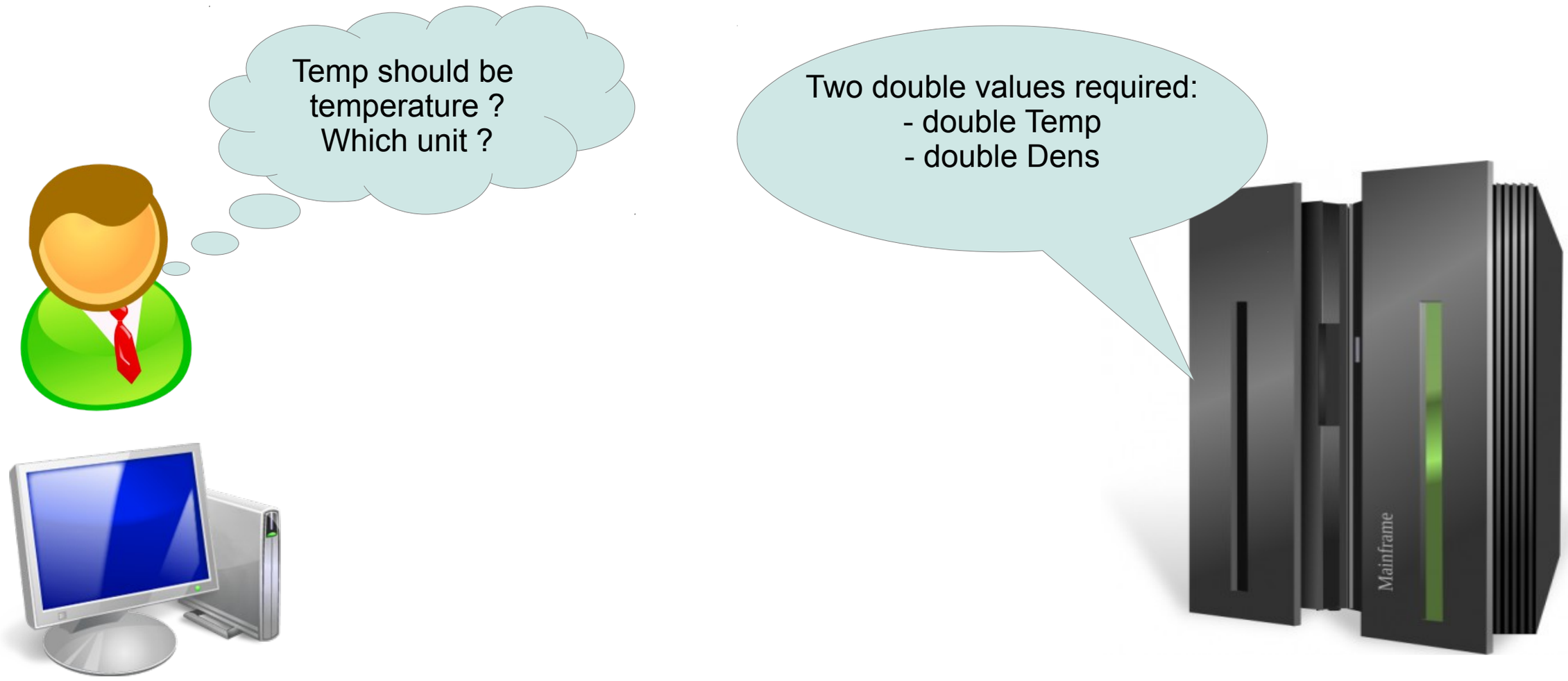
IVOA Documents

Parameter Description Language
Version 1.0

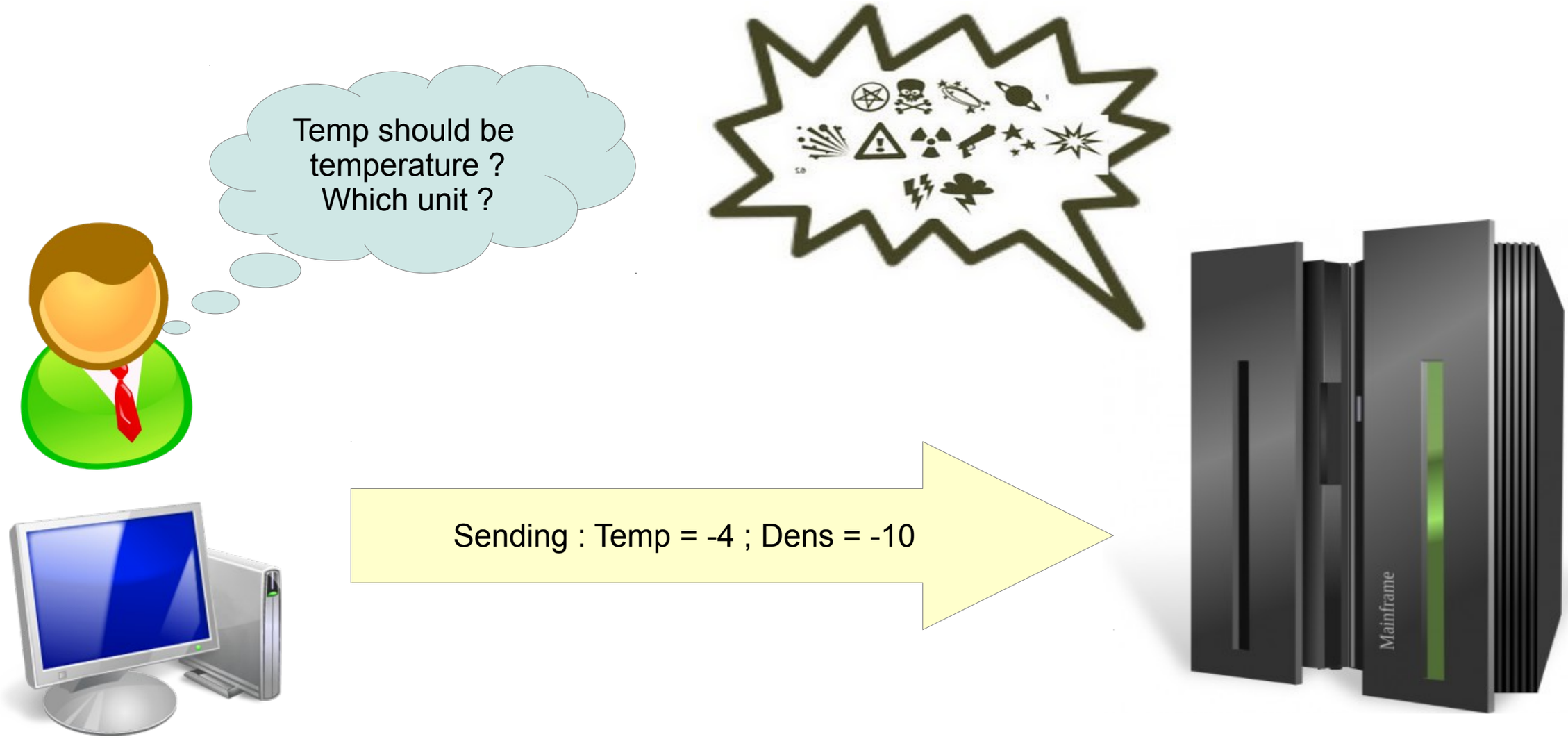
IVOA Recommendation 23 May 2014

Interest/Working Group:
http://www.ivoa.net/twiki/bin/view/IVOA/IvoaGridAndWebServices
Author(s):
Carlo Maria Zwolf, Paul Harrison, Julian Garrido, Jose Enrique Ruiz, Franck Le Petit
Editor(s):
Carlo Maria Zwolf

PDL Principles



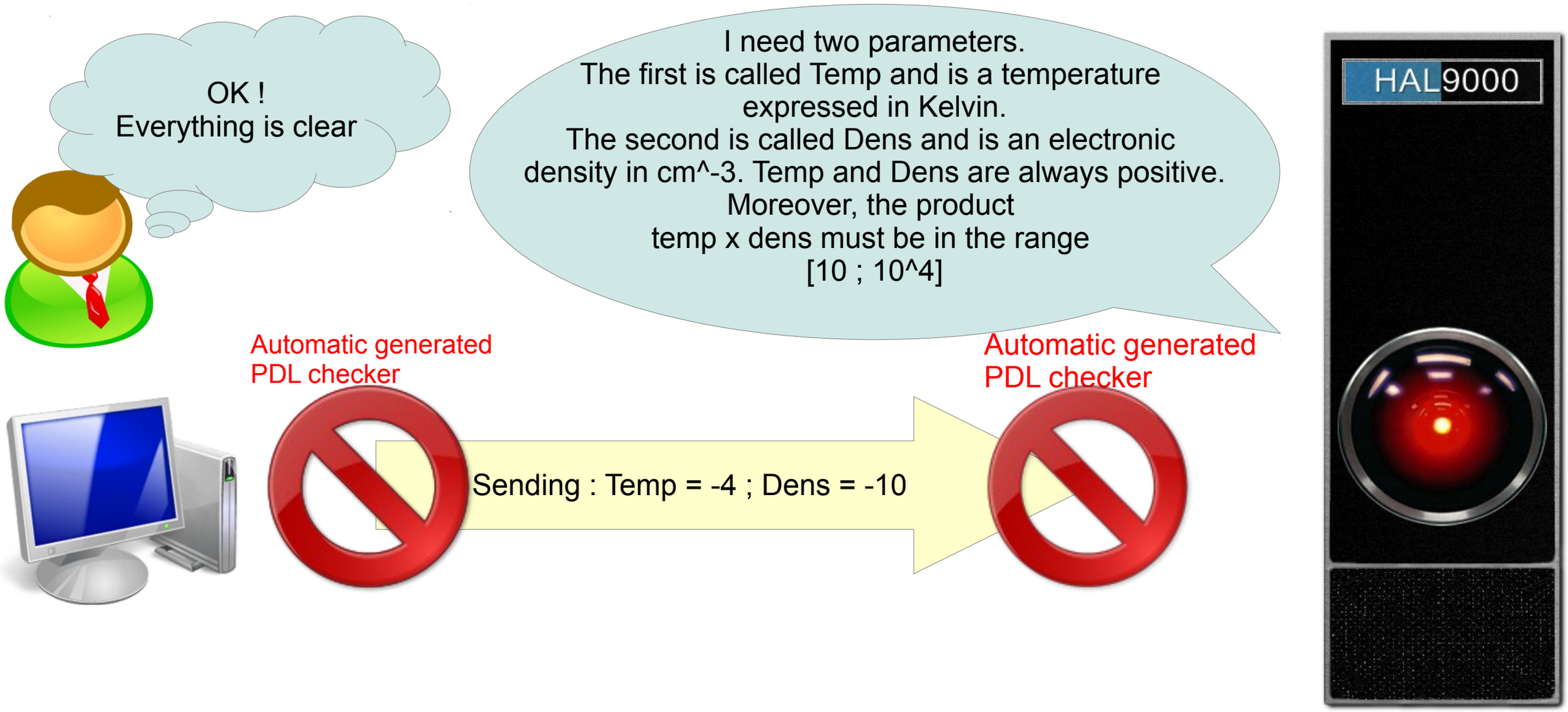
PDL Principles



PDL Principles

I need two parameters.
The first is called Temp and is a temperature expressed in Kelvin.
The second is called Dens and is an electronic density in cm^-3. Temp and Dens are always positive.
Moreover, the product
temp x dens must be in the range
[10 ; 10^4]

PDL Principles



Software components based on PDL

Since parameters and constraints are finely described with fine grained granularity:

- Generic software are automatically “configured” by a specific PDL description instance:
  - Services containers
  - Graphical User Interfaces
  - Workflow Plugins
- Checking algorithms and interoperability checker between service are automatically generated from descriptions

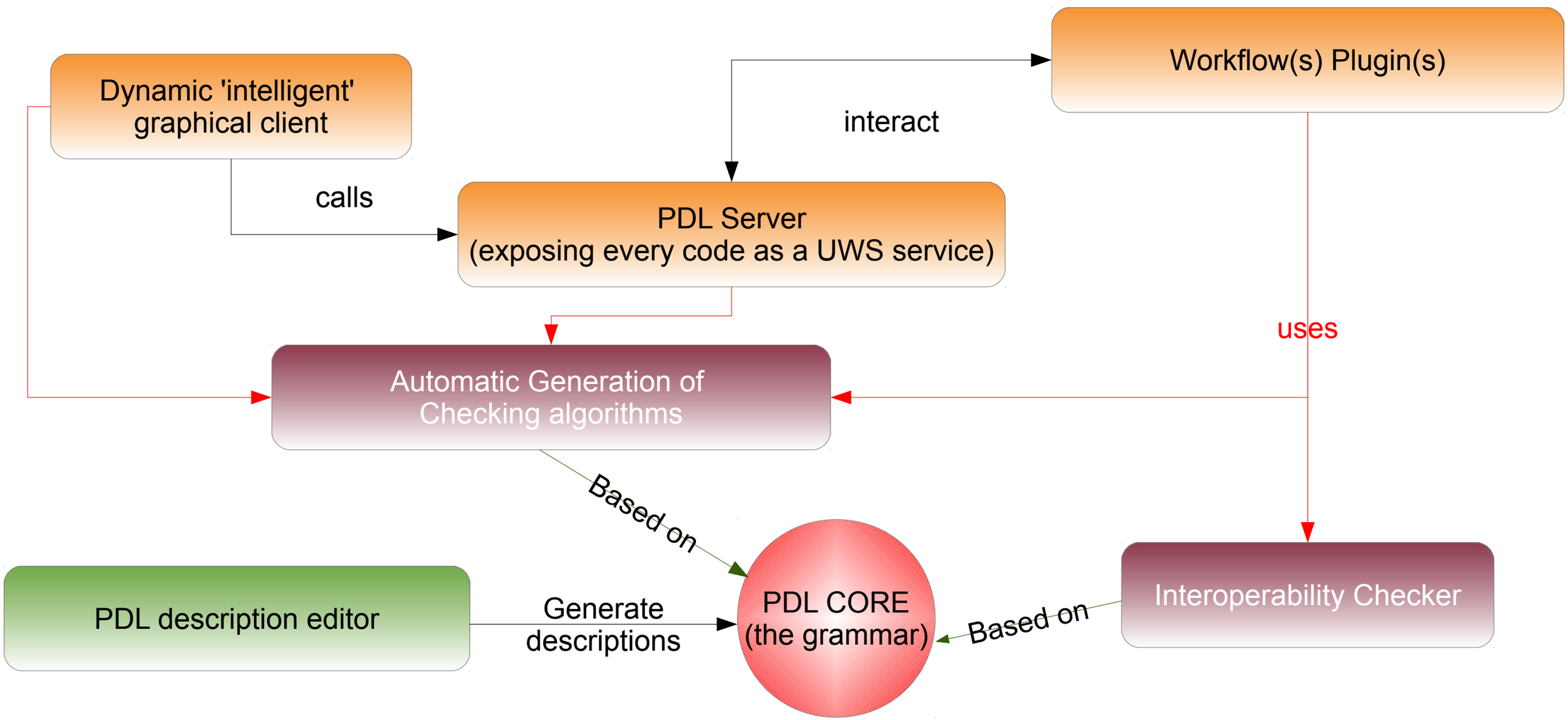
PDL: a quick overview

- Parameter Description Language (PDL) is intended to be a lingua franca of parameters:
  - Describes params in a sufficient detail to allow workflow tools to check if parameters can be "piped" between services
    - Physical Properties (Nature, Meaning, unit, precision,...)
    - Computing (Numerical Type, UCD, SKOS concept)
  - Also has capabilities do describe constraints on parameters
    - Physical constraints
    - Arbitrary (including mathematical) constraints
- **Not** a description of parameters "values" (cf. UWS).

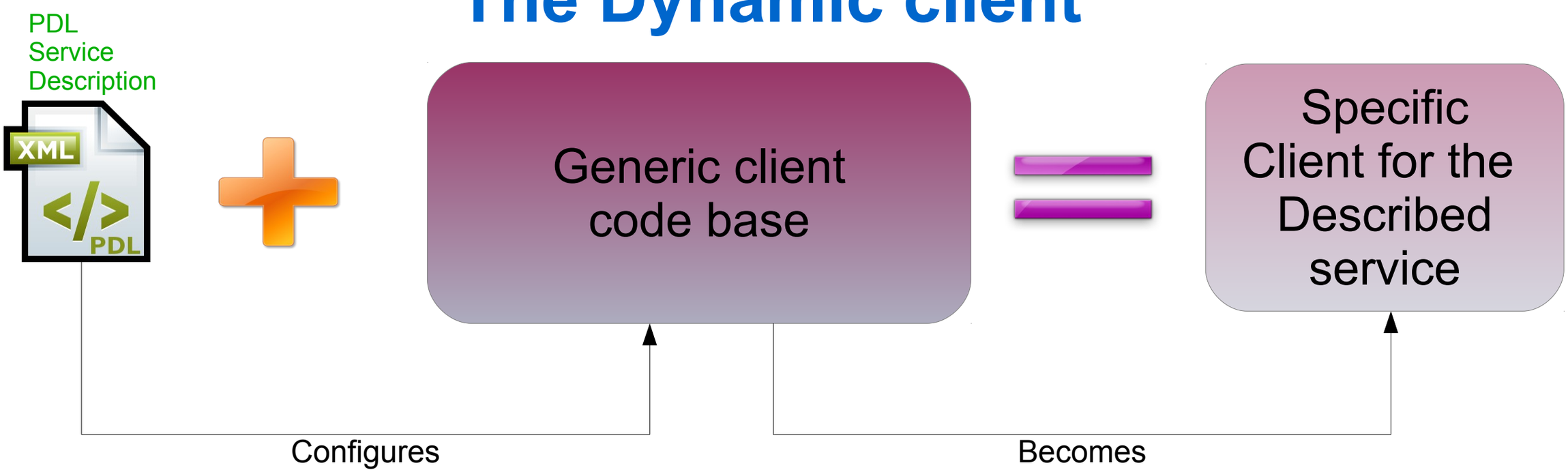
Software components based on PDL

Since parameters and constraints are finely described with fine grained granularity:

- Generic software elements could be automatically “configured” by a specific PDL description instance:
  - Services containers
  - Graphical User Interfaces
  - Workflow Plugins
- Checking algorithms and interoperability checker between service are automatically generated from descriptions

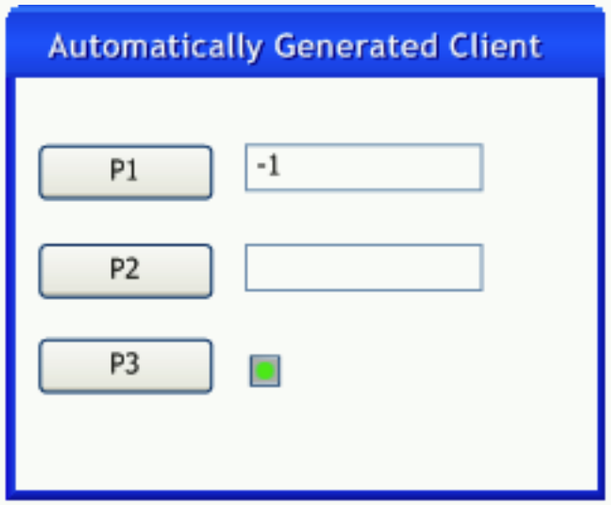


The Dynamic client

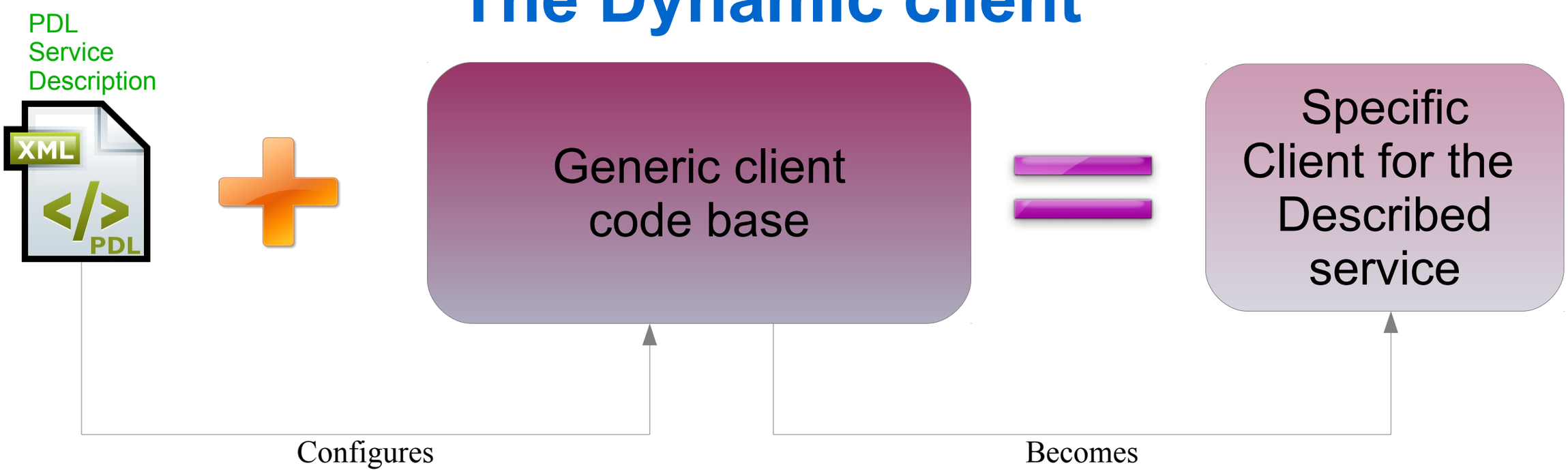


Service description:

- $p_1 \in \mathbb{R}$, $p_2 \in \mathbb{N}$ and p_3 is boolean.
- if $p_1 > 0 \implies p_2 \in \{2; 4; 6\}$ and p_3 must be false.
- if $p_1 < 0 \implies p_3$ must be true.

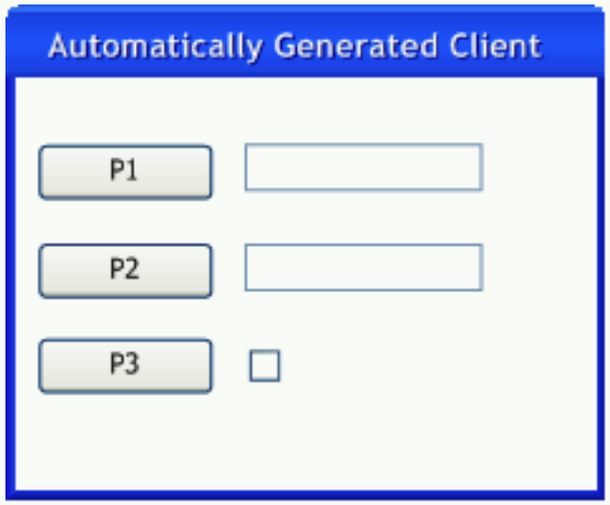


The Dynamic client

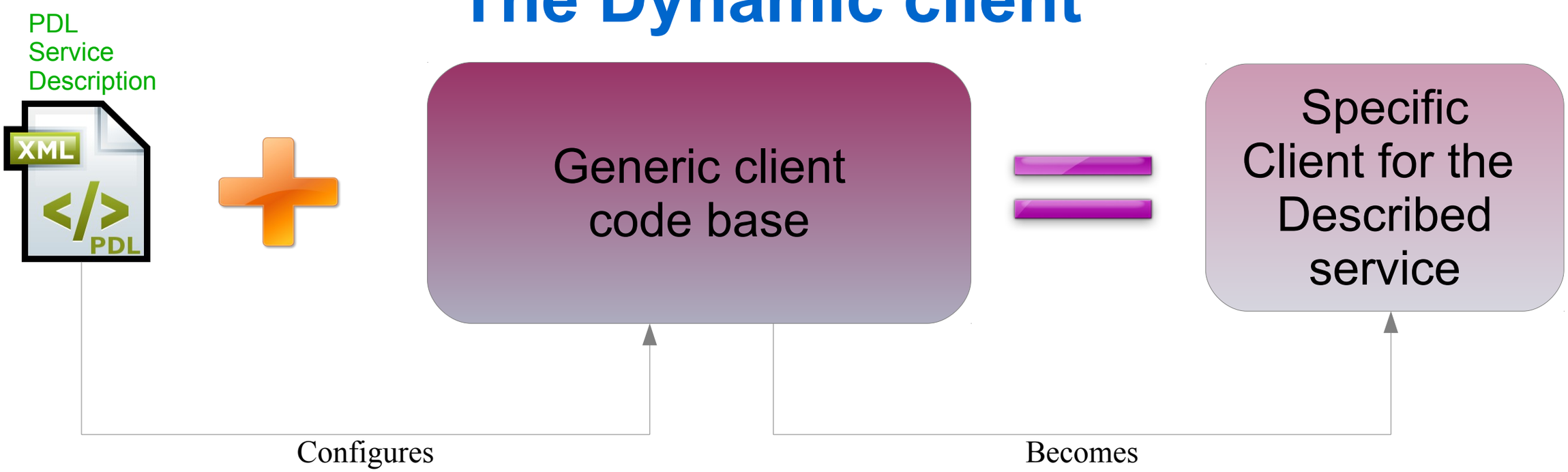


Service description:

- $p_1 \in \mathbb{R}$, $p_2 \in \mathbb{N}$ and p_3 is boolean.
- if $p_1 > 0 \implies p_2 \in \{2; 4; 6\}$ and p_3 must be false.
- if $p_1 < 0 \implies p_3$ must be true.

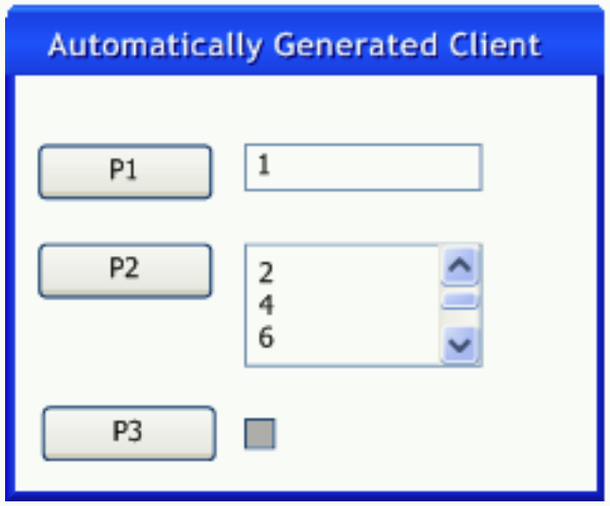


The Dynamic client

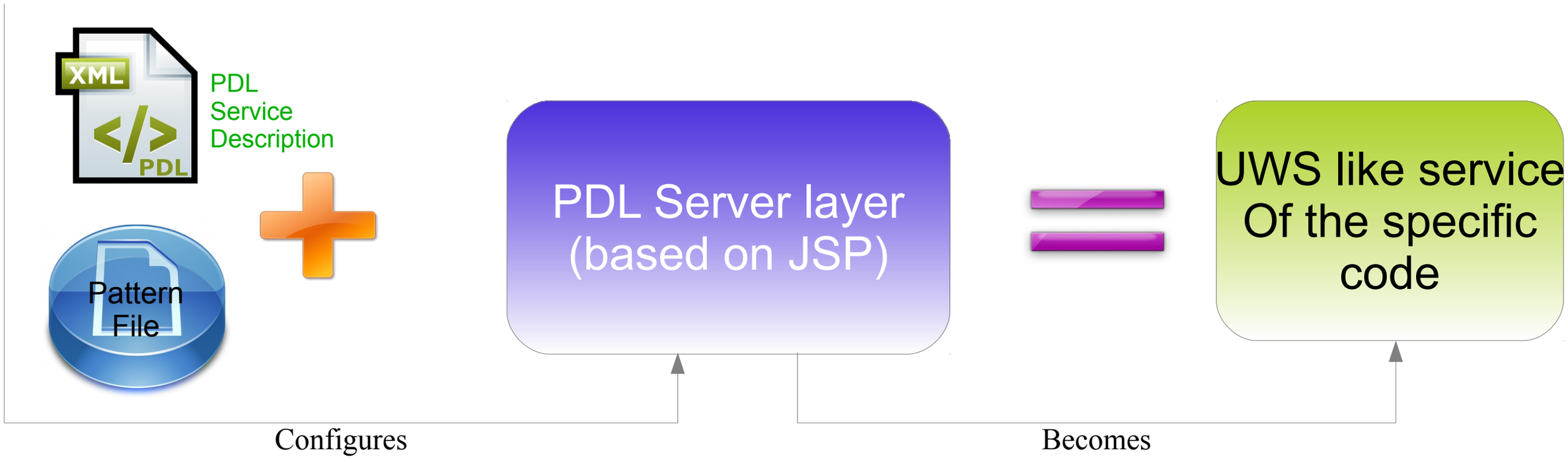


Service description:

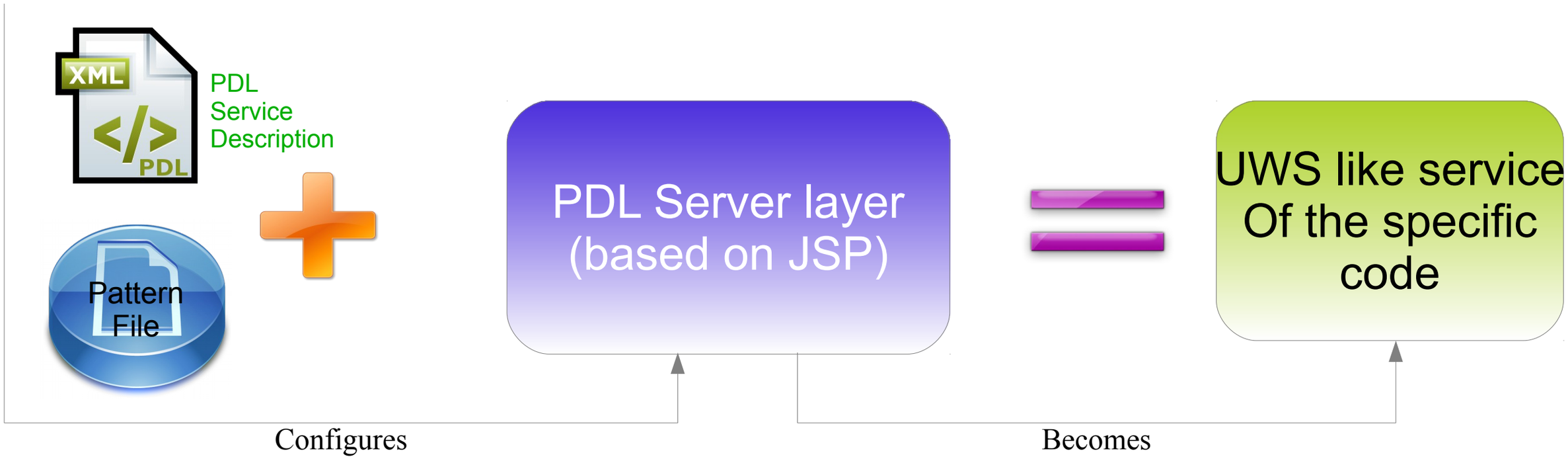
- $p_1 \in \mathbb{R}$, $p_2 \in \mathbb{N}$ and p_3 is boolean.
- if $p_1 > 0 \implies p_2 \in \{2; 4; 6\}$ and p_3 must be false.
- if $p_1 < 0 \implies p_3$ must be true.



Focus on the PDL server



Focus on the PDL server

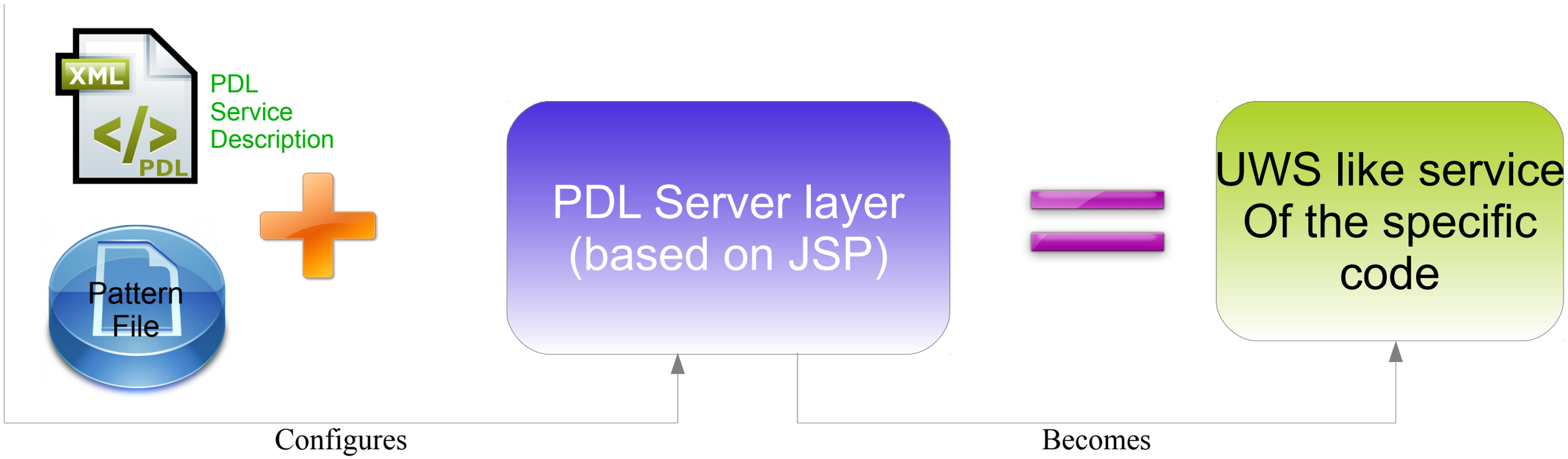


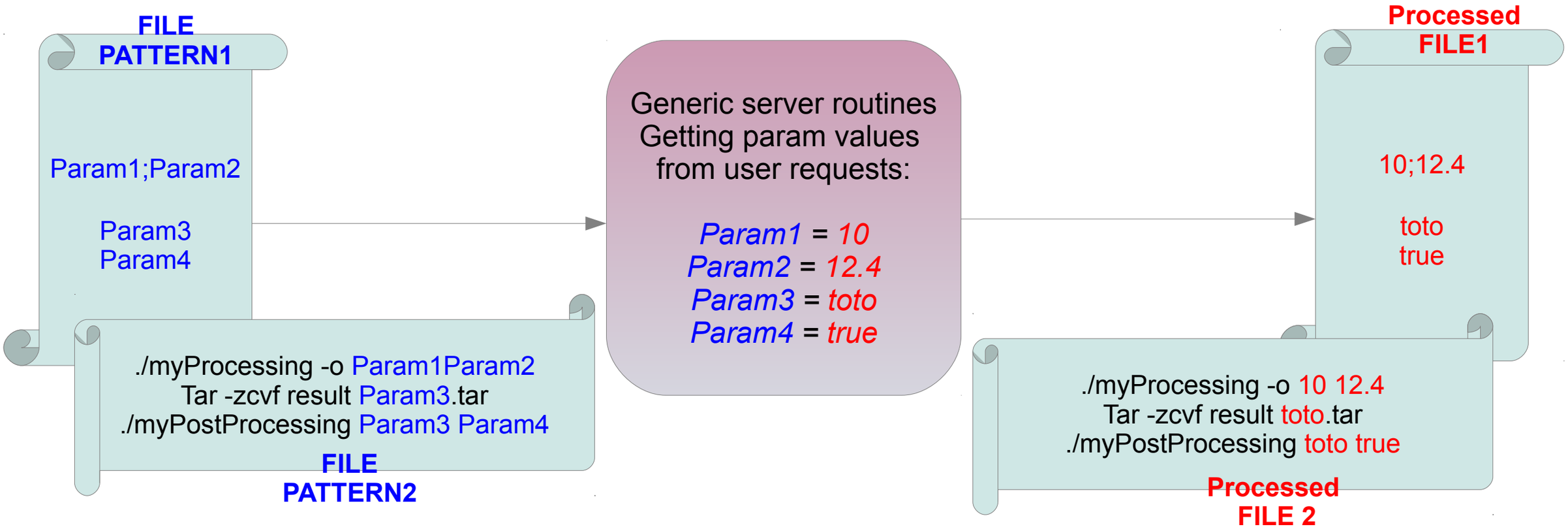
- Read the PDL description and
  - For each expected parameter, try to get the parameter provided by the user
  - Verify if the set of the provided parameters verify all the PDL constraints
    - Ok → the new job is created
    - No → PDL errors are notified to user as a server response

Generic server routines
Getting param values
from user requests:

Param1 = *10*
Param2 = *12.4*
Param3 = *toto*
Param4 = *true*

Focus on the PDL server





Focus on the PDL server

PDL server main features:

- It supports user authentication (a user cannot see the jobs or jobs lists of other users).
- It supports Grids of models
  - Jobs for parametric studies may be grouped into arbitrary sets of runs (GridID for each grid).

Focus on the PDL server

PDL server main features:

- It supports user authentication (a user cannot see the jobs or jobs lists of other users).

- It supports Grids of models
  - Jobs for parametric studies may be grouped into arbitrary sets of runs (GridID for each grid).

- It has three interfaces for job administration:
  - Two machine oriented
    - The first “speaking XML” (e.g. used by Taverna plugin)
    - The second “speaking Json” (for alternate clients e.g. PDR-code client).

  - One human readable
    - The old one (based on java servlet) has been redesigned using Google Web Toolkit
      - Three static web pages have been replaced by a unique dynamic page.

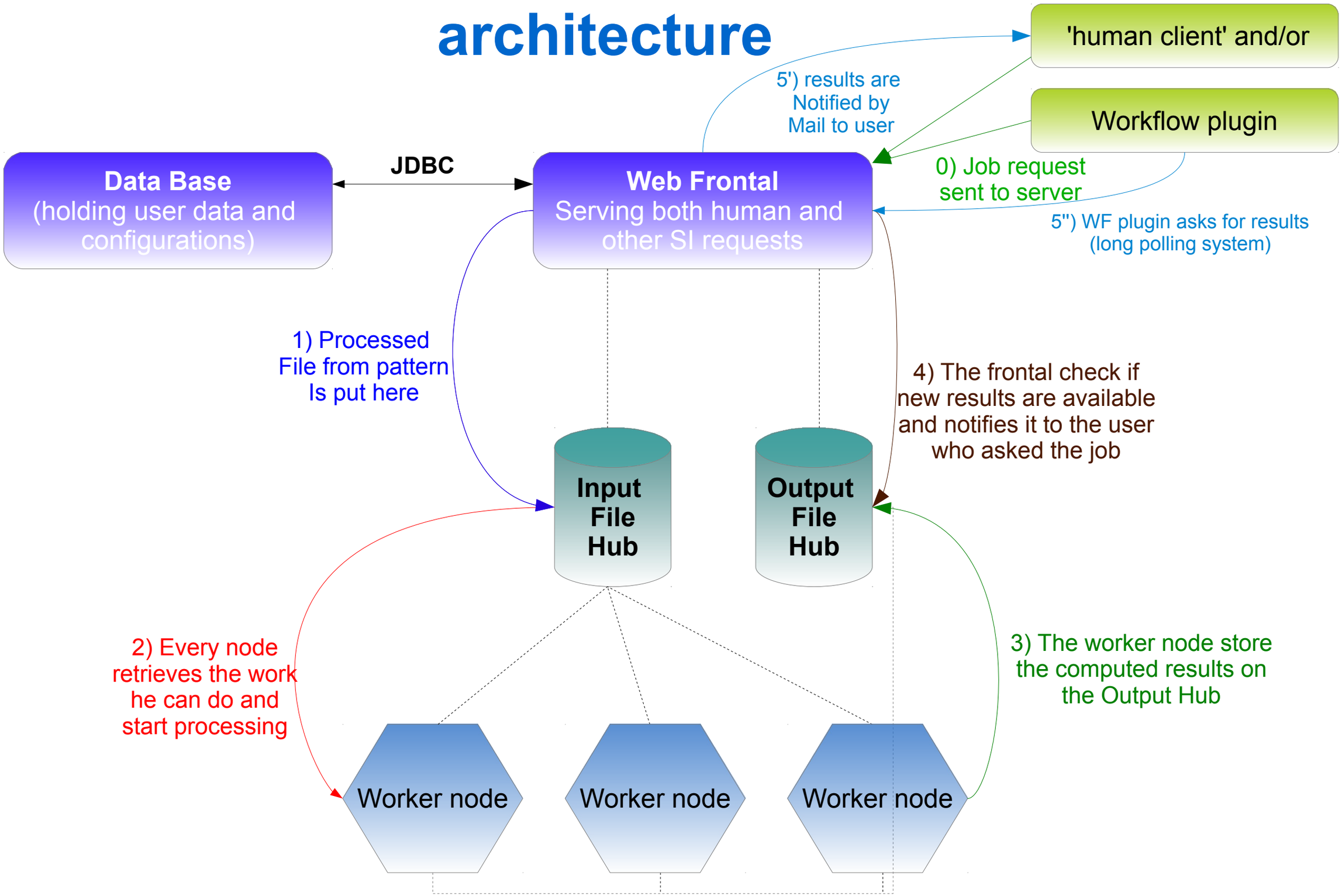
- **Designed to be straightforwardly deployed on local server, clusters,grid and cloud architectures**

Focus on the PDL server


```
{
   "errors": [
      {
         "errorMessage": "the following condition is not verified in the Grains Properties group: Grains max radius
                                 belongs to range 1e-6 - 1e-4",
         "involvedParameter(s)": [
            "los_ext",
            "rrr",
            "metal",
            "cdunit",
            "gratio",
            "q_pah",
            "alpgr",
            "rgrmin",
            "rgrmax",
            "F_DUST_P"
         ]
      }
   ]
}
```

```
{
   "ExpectedResultsURLs": [
      "http://tepig.obspm.fr:8081/pdrlight//output/PDRlight.zip"
   ],
   "UserMail": "test-pdr@obspm.fr",
   "JobID": 8,
   "ManagementURL": "http://tepig.obspm.fr:8081/pdrJobManager/userId=27&mail=test-pdr@obspm.fr",
   "UserID": 27,
   "ServiceId": "http://tepig.obspm.fr:8081/pdrlight/"
}
```

PDL Server: a distributed architecture



Remote filesystem mounting

Practical work

It is easy and quick to deploy from scratch a full working PDL service (client and server), even for non computer-science expert.

We are going through the following steps

- Configuration of the generic client using an instance of description
- Configuration if the server
  - Internal Database
  - Edition of pattern files
- Running the service

Practical work

It is easy and quick to deploy from scratch a full working PDL service (client and server), even for non computer-science expert.

We are going through the following steps

- Configuration of the generic client using an instance of description

- Configuration if the server
  - Internal Database
  - Edition of pattern files

- Running the service

The example service has the following features:

- Takes 4 parameters
  - Temperature (Real, K), Density (Real, cm-3), InitialLevel (Number Integer), FinalLevel (Number Integer)
  - Constrain InitialLevel < FinalLevel

- For the computation
  - The value of Temperature must be contained into a file having the .temp extension
  - The value of Density must be contained into a file having the .dens extension
  - The value of the two levels must be contained into a file having .levels extension
  - A file .sh is used for driving the computation server-side

- Have your own PDL service working now! Follow the 15 steps of the HowTo contained into the zip file at the url: http://pdl.obspm.fr/download/TutorialPDL2015.zip