



## SEXTEN INTEROP

### SOME EVOLUTIONS IN THE PDL FRAMEWORK IMPLEMENTATIONS

CARLO MARIA ZWÖLF AND PDL CONTRIBUTORS



Laboratoire d'Etude du Rayonnement  
et de la Matière en Astrophysique



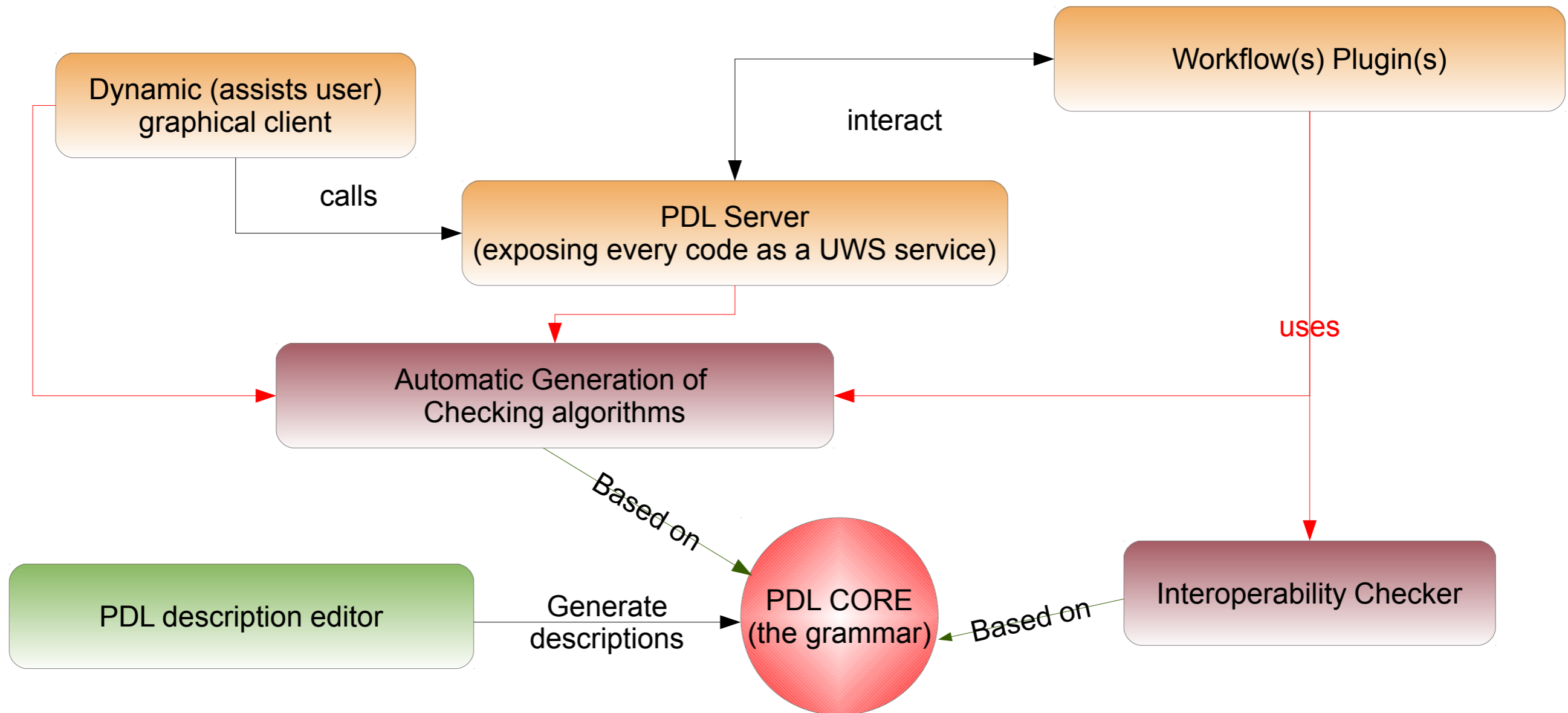
# The component of the PDL framework



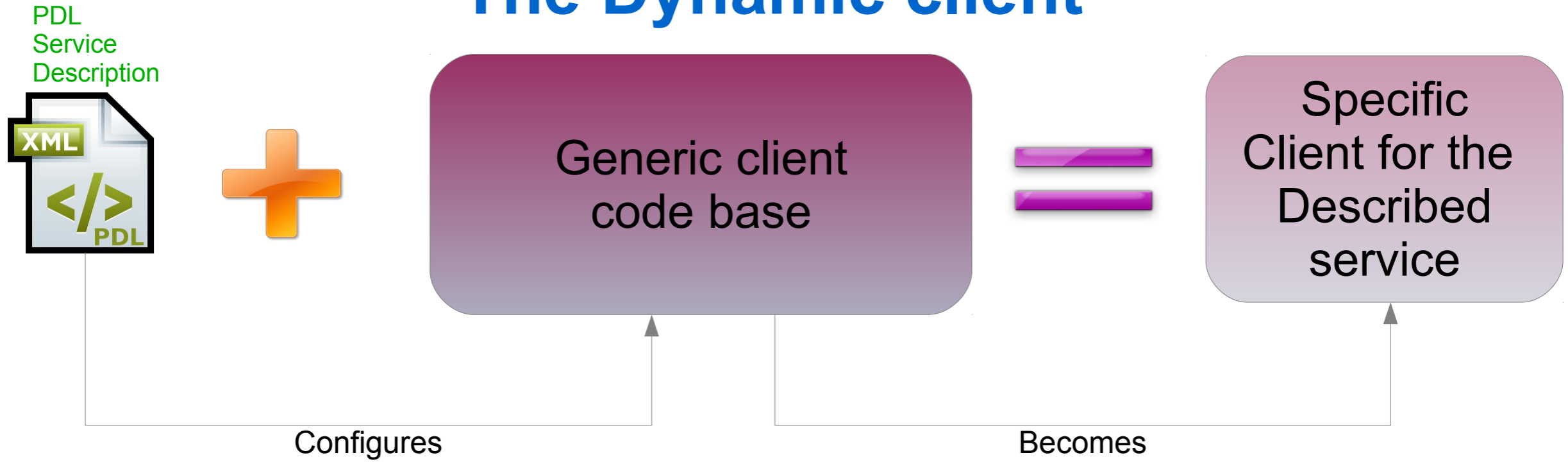
With PDL, the parameters and their related constraints are finely described with fine grained granularity:

- Generic software elements are automatically “configured” by a specific PDL description instance:
  - Services containers (PDL server)
  - Graphical User Interfaces (PDL client)
  - Workflow Plugins (Astro-Taverna plugin include the PDL layer)
- Checking algorithms and interoperability checker between service are automatically generated from descriptions

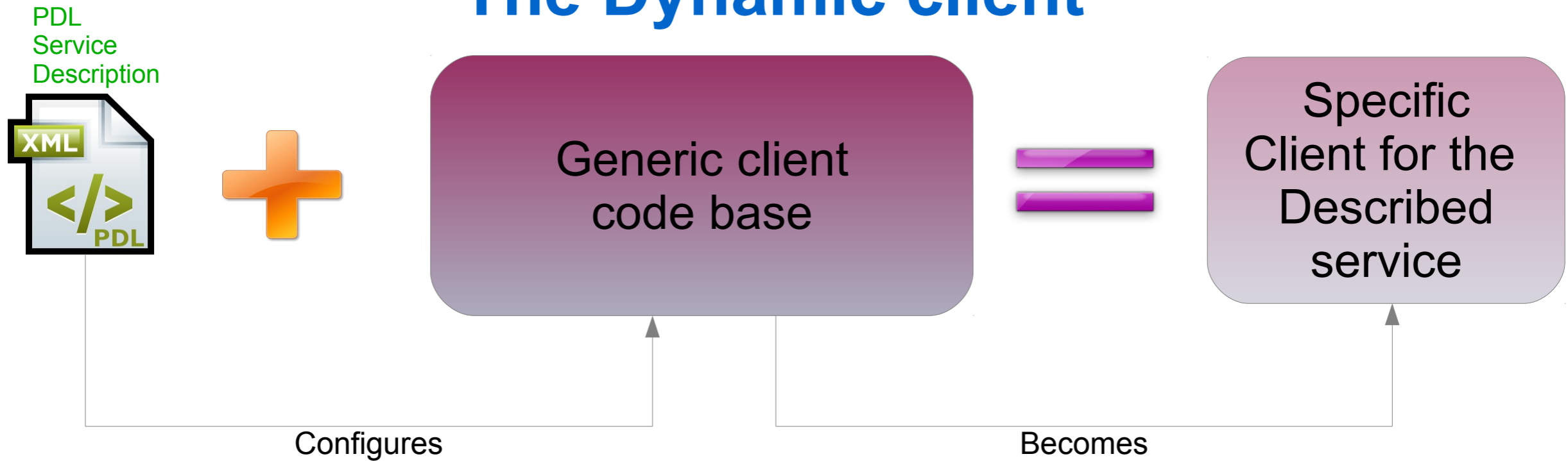
# The component of the PDL framework



# The Dynamic client



# The Dynamic client



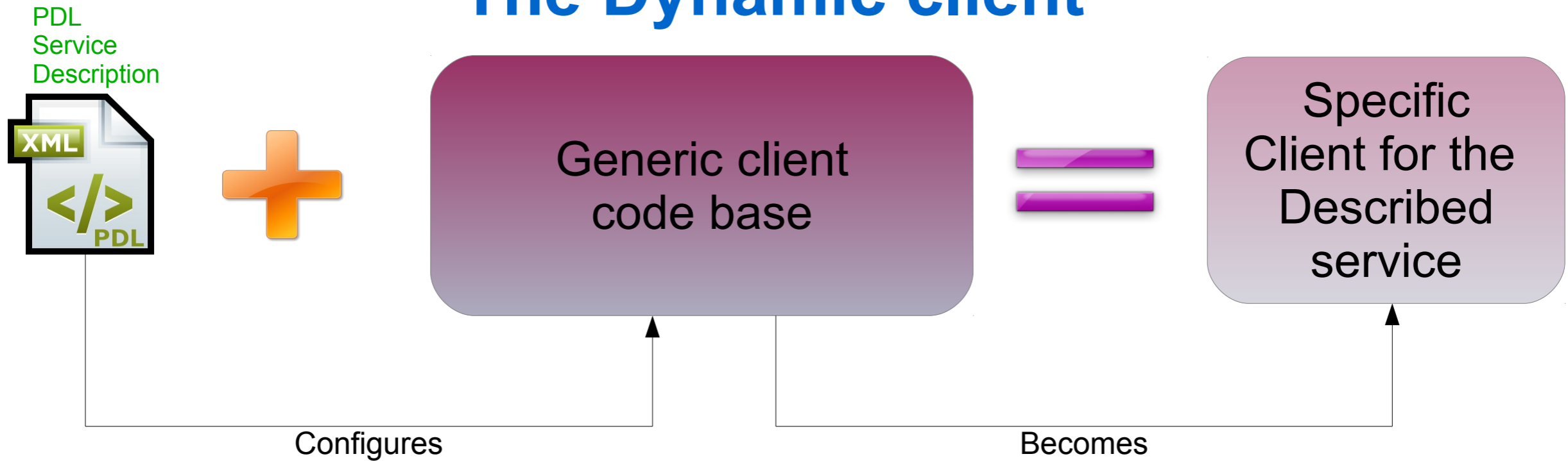
Service description:

- $p_1 \in \mathbb{R}$ ,  $p_2 \in \mathbb{N}$  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.

Automatically Generated Client

P1	<input type="text"/>
P2	<input type="text"/>
P3	<input type="checkbox"/>

# The Dynamic client



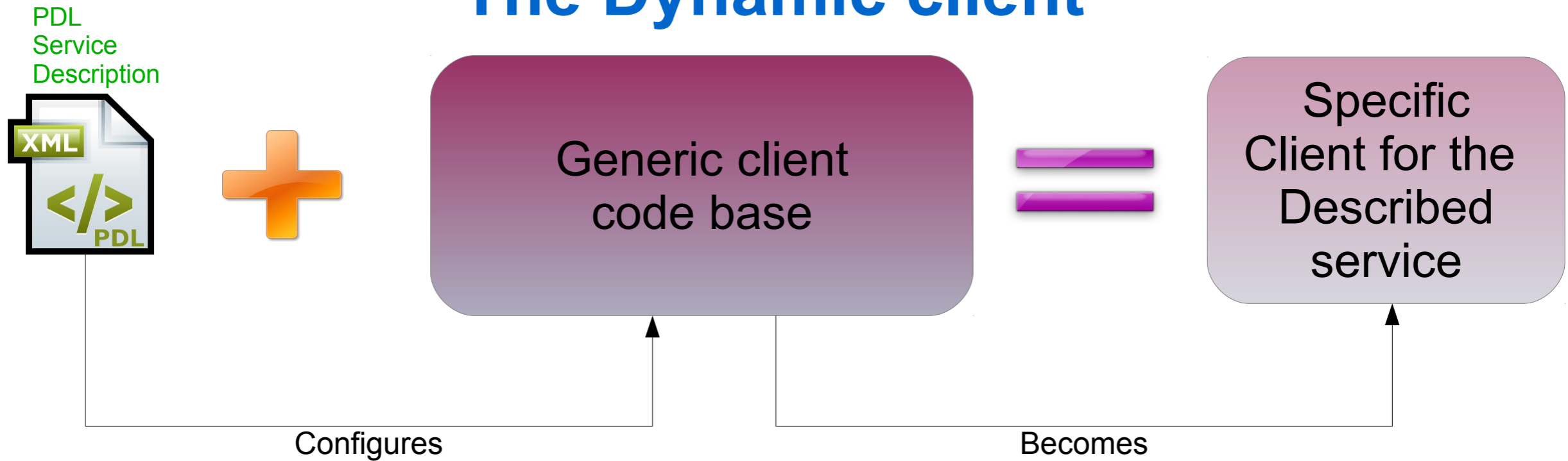
Service description:

- $p_1 \in \mathbb{R}$ ,  $p_2 \in \mathbb{N}$  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.

Automatically Generated Client

P1	<input type="text" value="1"/>
P2	<input type="list" value="2, 4, 6"/>
P3	<input type="checkbox"/>

# The Dynamic client



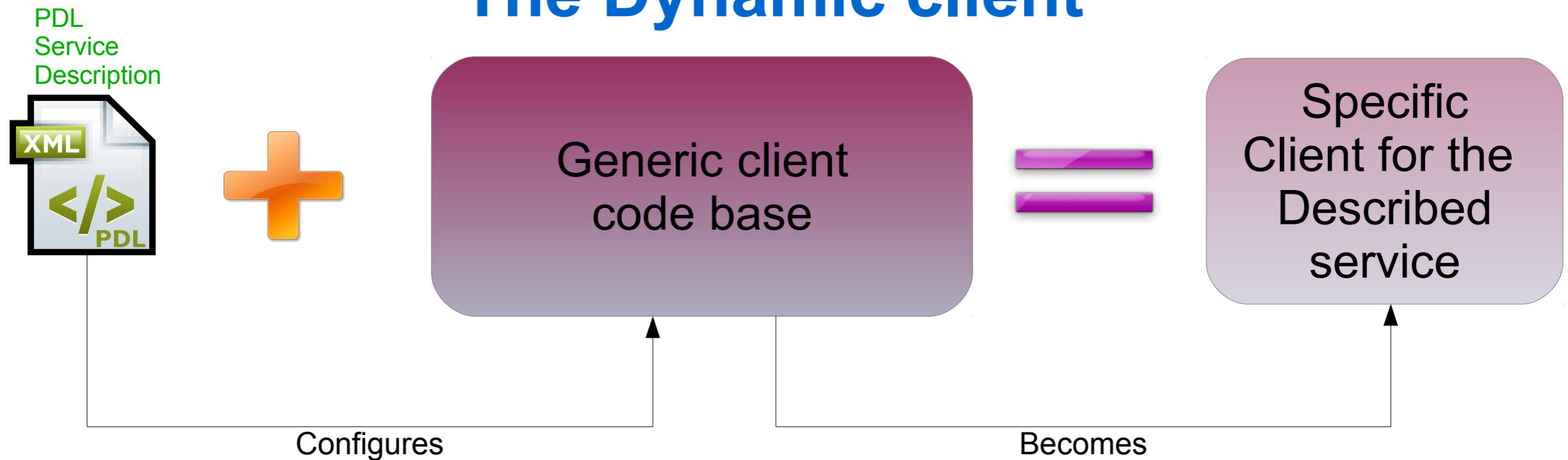
Service description:

- $p_1 \in \mathbb{R}$ ,  $p_2 \in \mathbb{N}$  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.

Automatically Generated Client

P1	<input type="text" value="-1"/>
P2	<input type="text"/>
P3	<input checked="" type="checkbox"/>

# The Dynamic client



- The client implementation realized for RFC is based on Java Swing
- A French SME (Artenum, [www.artenum.com](http://www.artenum.com)) adopted PDL and has developed a Java FX version of the dynamic client, based on our core components (released with a free license).



# The Dynamic client

Keridwen [Global Parameters Editor]

ARTENUM, PARIS  
Science & Groupware

MUSCA SEP<sup>3</sup>

Load a draft  
Load  
Save a draft  
Save

Groups

- Technologie
  - Type ERN
    - Proton
    - Ion lourd isotrope
    - Ion lourd unidirectionnel
      - caracteritiques ion
  - Blindage
    - Uniforme
    - Non uniforme
  - Modele
    - RPP
    - Diff Coll
  - Parametres

Parameters

Name	Value	Unit	Type
Particule *	Proton		string
ModeERN *	Automatique		string
CaracDir *	Isotrope		string
	Isotrope		
	Unidirectionnel		

Validate

Constraints

- Particule = Proton or Ion lourd
- CaracDir = Isotrope or Ion Unidirectionnel
- ModeERN = Automatique or Manuel

Parameter description

CaracDir : Isotrope ou unidir

# The Dynamic client

Keridwen [Global Parameters Editor]

ARTENUM, PARIS  
Science & Groupware

Example

Load a draft  
Load  
Save a draft  
Save

Groups

- ▼ G1
  - G1.1
  - G1.2
- ▼ G2
  - G2.1

Parameters

Name	Value	Unit	Type
✎ P1 *	Choice1	P1 Unit	string
	Choice1		
	Choice2		

Validate

Constraints

✎ P1 = Choice1 or Choice2

Parameter description

P1 : P1 description

# The Dynamic client

Keridwen [Global Parameters Editor]

ARTENUM, PARIS  
Science & Groupware

Example

Load a draft  
Load  
Save a draft  
Save

Groups

- ✓ G1
  - ✗ G1.1
  - 🔒 G1.2
- ✎ G2
  - 🔒 G2.1

Parameters

Name	Value	Unit	Type
✓ P2 *	2.72	P2 Unit	real
✓ P3 *	0	P3 Unit	real

Validate

Constraints

✗ If P2 is different from P3 then,  $\log|P2 - P3|$  must be smaller than 1.

Parameter description

# The Dynamic client

The screenshot shows the Keridwen [Global Parameters Editor] window. The title bar includes the text "Keridwen [Global Parameters Editor]" and standard window controls. The interface features the ARTENUM, PARIS Science & Groupware logo on the left and the word "Example" in the center. On the right side, there are four buttons: "Load a draft", "Load", "Save a draft", and "Save".

The main area is divided into four sections:

- Groups:** A tree view showing a hierarchy of groups: G1 (expanded), G1.1, G1.2 (locked), G2 (expanded), and G2.1 (locked).
- Parameters:** A table with columns for Name, Value, Unit, and Type. It contains two entries: P2 \* (value 2.71, unit P2 Unit, type real) and P3 \* (value 0, unit P3 Unit, type real). A "Validate" button is located below the table.
- Constraints:** A list of constraints, including one that states: "If P2 is different from P3 then,  $\log|P2 - P3|$  must be smaller than 1."
- Parameter description:** A large empty text area for describing parameters.

A scrollbar is visible at the bottom left of the interface.

# The Dynamic client

The screenshot shows the 'Keridwen [Global Parameters Editor]' window. The title bar includes standard window controls. The interface features the 'ARTENUM, PARIS Science & Groupware' logo on the left and the word 'Example' in the center. On the right side, there are four buttons: 'Load a draft', 'Load', 'Save a draft', and 'Save'. The main area is divided into two columns: 'Groups' and 'Parameters'. The 'Groups' column contains a tree view with nodes G1, G1.1, G1.2, G2, and G2.1. The 'Parameters' column contains a table with columns 'Name', 'Value', 'Unit', and 'Type'. The first row is highlighted in blue and contains 'P6 \*', an empty input field, 'P6 Unit', and 'integer'. A tooltip 'P6 : P6 description' is visible over the 'Unit' cell. Below the table is a 'Validate' button. At the bottom, there is a 'Parameter description' section containing the text 'P6 : P6 description'.

Keridwen [Global Parameters Editor]

ARTENUM, PARIS  
Science & Groupware

Example

Load a draft  
Load  
Save a draft  
Save

Groups

- ▼ ✓ G1
  - ✓ G1.1
  - 🔒 G1.2
- ▼ ✓ G2
  - ✎ G2.1

Parameters

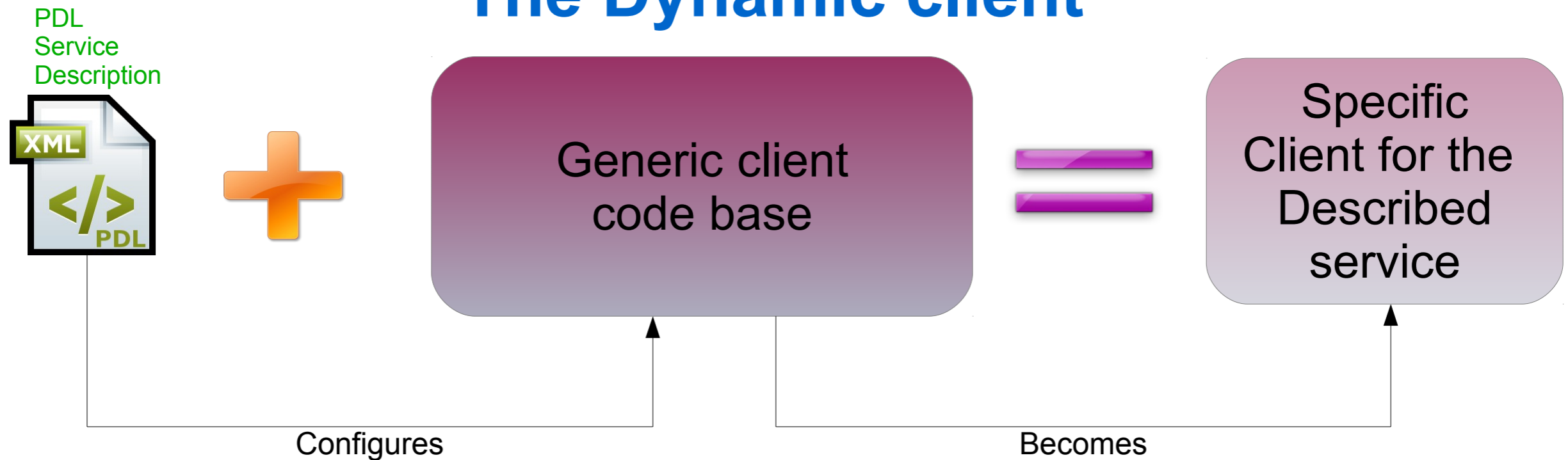
Name	Value	Unit	Type
✎ P6 *	<input type="text"/>	P6 Unit	integer
	<input type="text"/>		
	<input type="text"/>		

Validate

Parameter description

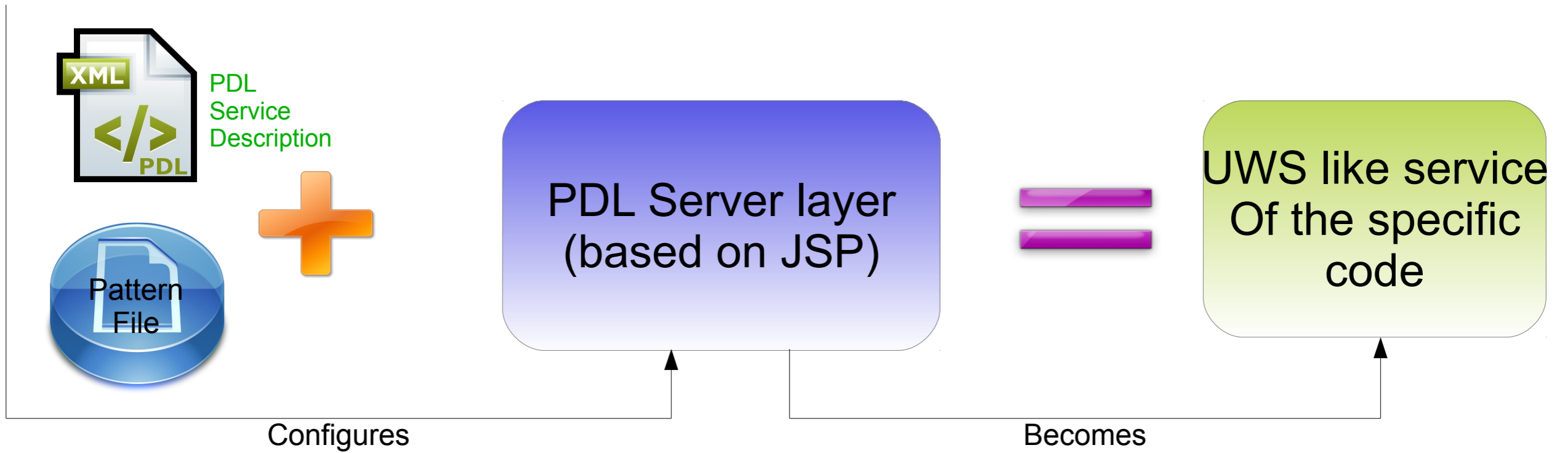
P6 : P6 description

# The Dynamic client

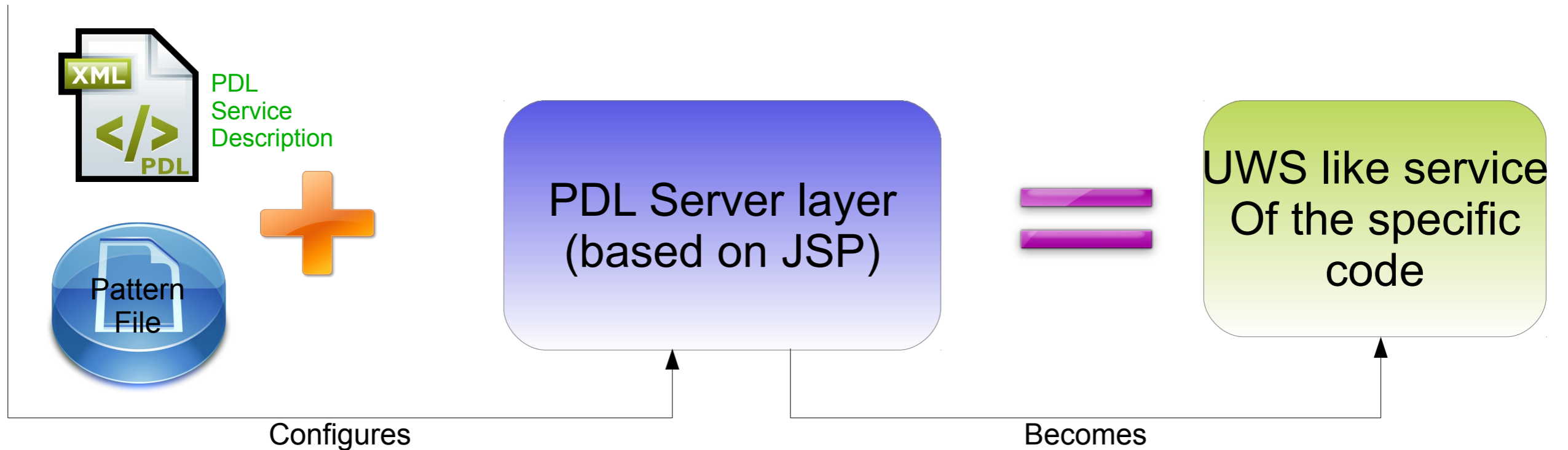


- The client implementation realized for RFC is based on Java Swing
- A French SME (Artenum, [www.artenum.com](http://www.artenum.com)) adopted PDL and has developed a Java FX version of the dynamic client, based on our core components (released with a free license).
- I am thinking at providing a full web version of the client, and started testing prototypes with Google Web Toolkit.
  - Do you think this is useful?
  - Do you see any particular feature to include? (e.g. embedded workflow engine, cf. Final remarks)

# Focus on the PDL server



# Focus on the PDL server



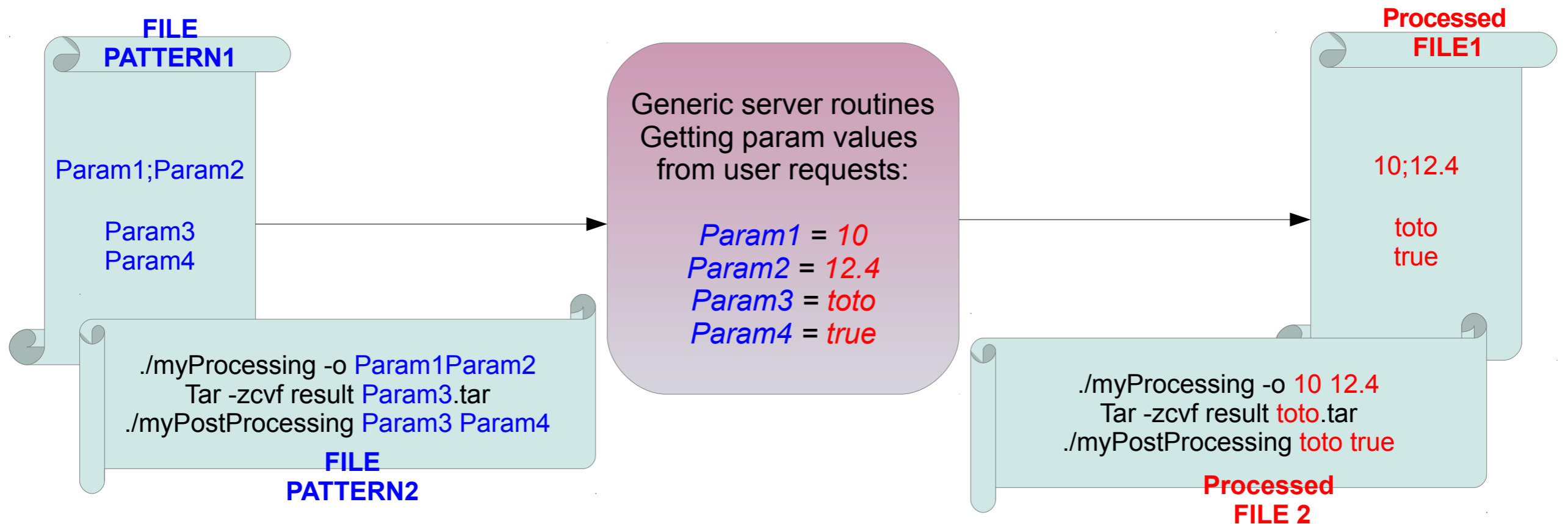
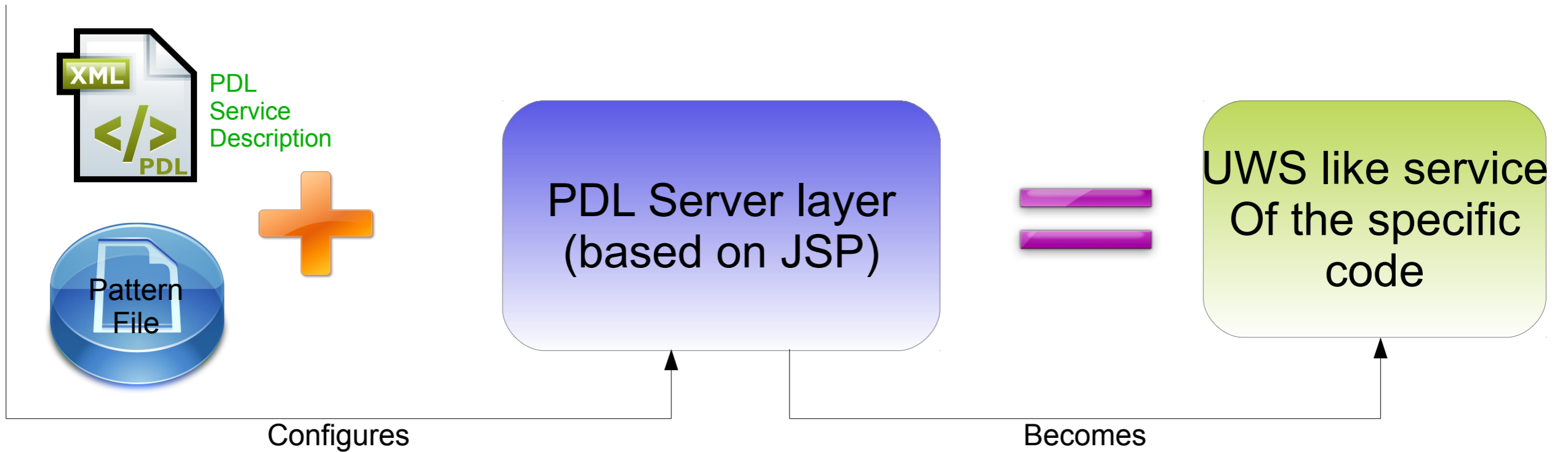
- Read the PDL description and
  - For each expected parameter, try to get the parameter provided by the user
  - Verify if the set of the provided parameters verify all the PDL constraints
    - **Ok** → the new job is created
    - **No** → PDL errors are notified to user as a server response

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.

# 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/"
}
```

# Focus on the PDL server

## PDL Service

Job list for user antoine.gusdorf@googlemail.com

Job Id	Job Phase	Demand Date	End Date
233	finished	2015/04/08 15:13:26	2015/04/10 09:45:02
232	running	2015/04/08 11:47:13	
201	finished	2014/10/14 11:14:52	2014/10/14 12:30:03
182	finished	2014/04/21 11:52:16	2014/05/02 22:21:03
181	finished	2014/04/21 11:33:51	2014/05/02 22:12:02
180	finished	2014/04/21 11:32:59	2014/05/02 21:41:02
179	finished	2014/04/21 11:32:22	2014/05/02 21:27:02
178	finished	2014/04/21 11:29:30	2014/05/01 20:57:03
172	finished	2014/04/21 11:44:29	2014/04/27 00:01:02
169	finished	2014/04/21 11:45:05	2014/04/25 20:56:02
138	finished	2014/04/21 11:43:39	2014/04/19 19:02:02
135	finished	2014/04/21 11:42:55	2014/04/19 16:51:03
132	finished	2014/03/24 15:42:30	2014/03/24 16:49:02
131	finished	2014/03/24 15:40:25	2014/03/24 16:32:03
130	finished	2014/03/24 15:40:07	2014/03/24 16:19:03

# Focus on the PDL server

Job list for user antoine.gusdorf@googlemail.com

Job Id	Job Phase	Demand Date	End Date
233	finished	2015/04/08 15:13:26	2015/04/10 09:45:02
232	running	2015/04/08 11:47:13	
201	finished	2014/10/14 11:14:52	2014/10/14 12:30:03
182	finished	2014/04/21 11:52:16	2014/05/02 22:21:03
181	finished	2014/04/21 11:33:51	2014/05/02 22:12:02
180	finished	2014/04/21 11:32:59	2014/05/02 21:41:02
179	finished	2014/04/21 11:32:22	2014/05/02 21:27:02
178	finished	2014/04/21 11:29:30	2014/05/01 20:57:03
172	finished	2014/04/21 11:44:29	2014/04/27 00:01:02
169	finished	2014/04/21 11:45:05	2014/04/25 20:56:02
138	finished	2014/04/21 11:43:39	2014/04/19 19:02:02
135	finished	2014/04/21 11:42:55	2014/04/19 16:51:03
132	finished	2014/03/24 15:42:30	2014/03/24 16:49:02
131	finished	2014/03/24 15:40:25	2014/03/24 16:32:03
130	finished	2014/03/24 15:40:07	2014/03/24 16:19:03

1-15 of 37

Detail for the selected Job (Id=233)

Delete this job

ParisDurhamFileResult: <http://pdl-calc.obspm.fr:8081/ParisDurham/output/233.pdshock.tgz>

Parameter Name	Parameter Value
xll	1e9
shockType	C
nHi	1e4
ikinH2	2
iforH2	1
epsV	1e-8
Zeta	5e-17
Vs	26.5
Vdi	1e3
TimeJ	2e7
Ti	10
Tg	15
SOS	FD
OpH2	3
NstepW	5
NstepMax	10000
Nfluids	3
NH2Lines	200
NH2Lev	150
MaxTimeN	1e6
LIOS	integrated
LEOS	ln(N/g)
Bbeta	1

1-23 of 23

# Focus on the PDL server

Job list for user antoine.gusdorf@googlemail.com

Job Id	Job Phase	Demand Date	End Date
233	finished	2015/04/08 15:13:26	2015/04/10 09:45:02
232	running	2015/04/08 11:47:13	
201	finished	2014/10/14 11:14:52	2014/10/14 12:30:03
182	finished	2014/04/21 11:52:16	2014/05/02 22:21:03
181	finished	2014/04/21 11:33:51	2014/05/02 22:12:02
180	finished	2014/04/21 11:32:59	2014/05/02 21:41:02
179	finished	2014/04/21 11:32:22	2014/05/02 21:27:02
178	finished	2014/04/21 11:29:30	2014/05/01 20:57:03
172	finished	2014/04/21 11:44:29	2014/04/27 00:01:02
169	finished	2014/04/21 11:45:05	2014/04/25 20:56:02
138	finished	2014/04/21 11:43:39	2014/04/19 19:02:02
135	finished	2014/04/21 11:42:55	2014/04/19 16:51:03
132	finished	2014/03/24 15:42:30	2014/03/24 16:49:02
131	finished	2014/03/24 15:40:25	2014/03/24 16:32:03
130	finished	2014/03/24 15:40:07	2014/03/24 16:19:03

1-15 of 37

Detail for the selected Job (Id=233)

Delete this job

ParisDurhamFileResult: <http://pdl-calc.obspm.fr:8081/ParisDurham/output/233.pdshock.tgz>

▼ Parameter Name Parameter Value

Bbeta	1
LEOS	ln(N/g)
LIOS	integrated
MaxTimeN	1e6
NH2Lev	150
NH2Lines	200
Nfluids	3
NstepMax	10000
NstepW	5
OpH2	3
SOS	FD
Tg	15
Ti	10
TimeJ	2e7
Vdi	1e3
Vs	26.5
Zeta	5e-17
epsV	1e-8
iforH2	1
ikinH2	2
nHi	1e4
shockType	C
xll	1e9

1-23 of 23

# Focus on the PDL server

It is based on UWS but:

- Uses Java servlets for job management (functional architecture and not REST) Recall, REST is just a binding example (actually the only) for UWS. It is not the core part of the norm and historically a soap binding was proposed.
- Has extra features for dealing with
  - Grids of jobs (e.g. search jobs by GridID)
  - User authentication/authorisation
  - Live notification of violated constraints.



# Focus on the PDL server

It is based on UWS but:

- Uses Java servlets for job management (functional architecture and not REST) Recall, REST is just a binding example (actually the only) for UWS. It is not the core part of the norm and historically a soap binding was proposed.
- Has extra features for dealing with
  - Grids of jobs (e.g. search jobs by GridID)
  - User authentication/authorisation
  - Live notification of violated PDL constraints.

What could be done to approach (and ideally converge) the UWS and the PDL server?

Is it convenient to do so?

# PDL provides some answers to issues discussed into Massive and complex data session - Final remarks-

- It is a very convenient way for exposing codes (thus bringing processes to data).
  - It is fast to deploy services using PDL framework
  - PDL avoids “dummy computation” (runs with non-sense parameters): parameters verifications performed before jobs creation.
  - PDL server is ready to work with computer clusters, cloud & computing grids.
- Enable “transversal interoperability” between services (PDL may be seen as a meta-language for describing workflows, cf. PDL presentation of [interop@Pune](#))

## Ongoing efforts:

- Should a new client embed a sort of graphical workflow engine.
  - Users can actually build “script-oriented” workflows, using the Json interface of PDL server.
- What can we do in GWS to boost the operational adoption of PDL?
  - Politically?
  - Practically? (e.g. how to register PDL services into registries?)