Urbana Champaign Interop 2012

Parameter Description Language :

A generic grammar for parameters (and relates constraints) description.

Presenting the generic client and working services

Carlo Maria Zwölf,

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

Franck Le Petit,

Laboratoire Univers et Théories

Paul Harrison.

The University of Manchester

Introduction to PDL

- PDL is a grammar for

Describing parameters				
Describe physical properties of parameters				
Nature	Meaning	Unit	Precision	Range
Describe complex relations involving parameters				
Physical constraints	Arbitrary Conditions	Mathematical Conditions		

Making *transversal* (i.e. cross standard) interoperability straightforward within all the VO components		
Interaction of two services has sense if the parameter sent by the first and expected by the second have same		
Computer type	Physical concept	Unit
Interaction of two services has sense if all preconditions of second service are satisfied by output of first one		

Introduction to PDL

- PDL is a grammar for

Describing parameters				
Describe physical properties of parameters				
Nature	Meaning	Unit	Precision	Range
Describe complex relations involving parameters				
Physical constraints		Arbitrary Conditions		Mathematical Conditions

Making *transversal* (i.e. cross standard) interoperability straightforward within all the VO component		
Interaction of two services has sense if the parameter sent by the first and expected by the second have same		
Computer type	Physical concept	Unit
Interaction of two services has sense if all preconditions of second service are satisfied by output of first one		

Indeed PDL capabilities meet:

- The "*scientific*" description needs
- The "*scientific*" interoperability needs

- Using PDL one could describe
  - All the mathematical constraints on params
  - All the possible conditional sentences
- PDL Descriptions
  - Can be understood easily by humans
  - Can be interpreted and handled by a computer

- Concepts and Details will be presented in GWS session.
- PDL standard working draft available at pdl.obspm.fr

PDL main corollary

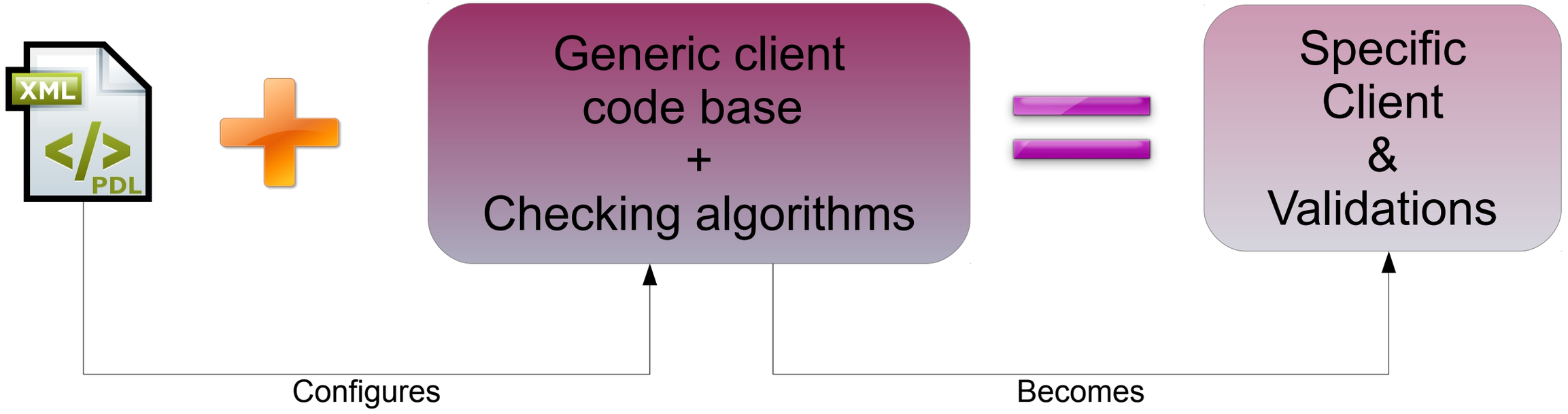
Since parameters are finely described with fine grained granularity

- Interoperability becomes possible in the *smart* sense we need
- Dynamic interactive graphical client can be automatically generated from description
- Checking algorithms for validating data can be automatically generated from description

PDL main corollary

Since parameters are finely described with fine grained granularity

- Interoperability becomes possible in the *smart* sense we need
- Dynamic interactive graphical client can be automatically generated from description
- Checking algorithms for validating data can be automatically generated from description

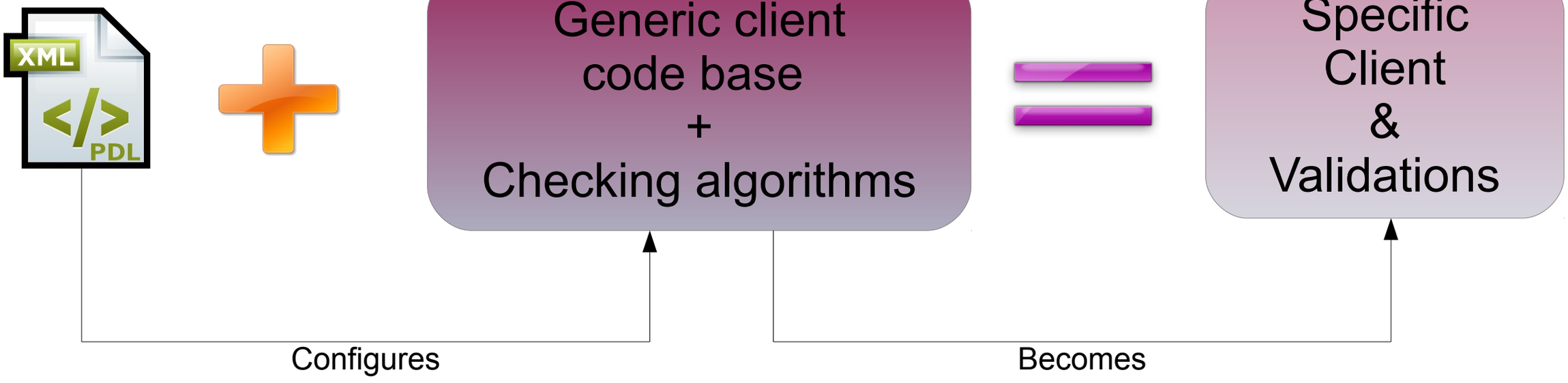


- Generic client can be *configured* on the fly using PDL description

PDL main corollary

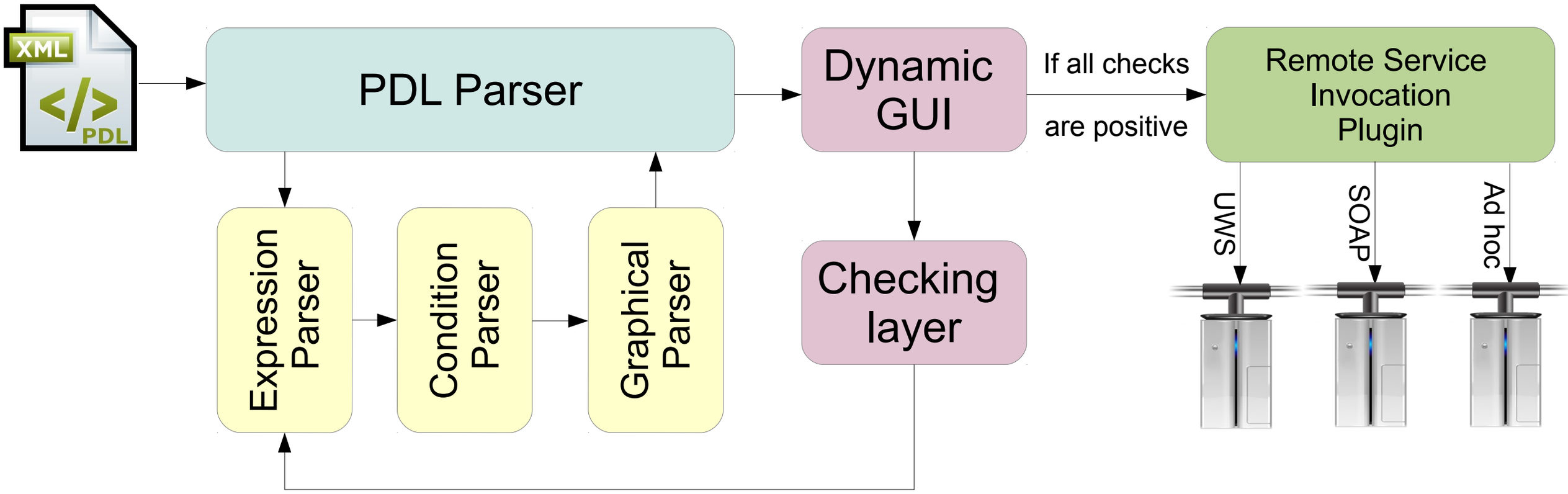
Since parameters are finely described with fine grained granularity

- Interoperability becomes possible in the *smart* sense we need
- Dynamic interactive graphical client can be automatically generated from description
- Checking algorithms for validating data can be automatically generated from description

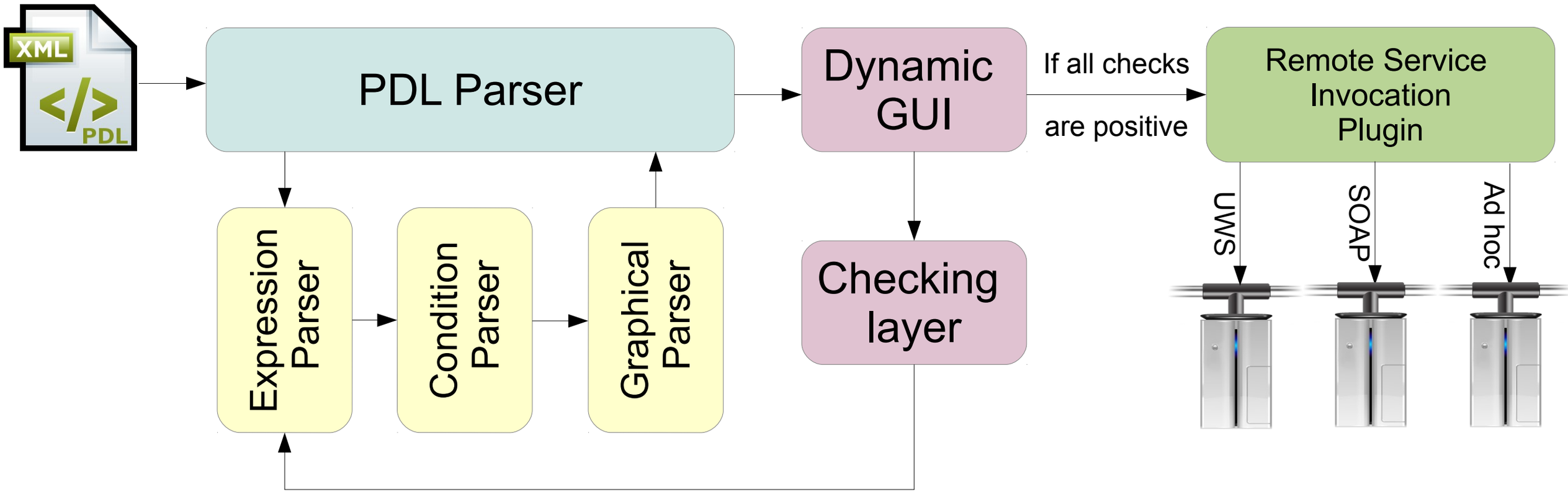


Some Statistics on code	
Lines of code	**~ 10 000 total** ~ 3 000 automatically generated
Classes	128
Methods	835

Dynamic client & Checking layer software architectures



Dynamic client & Checking layer software architectures

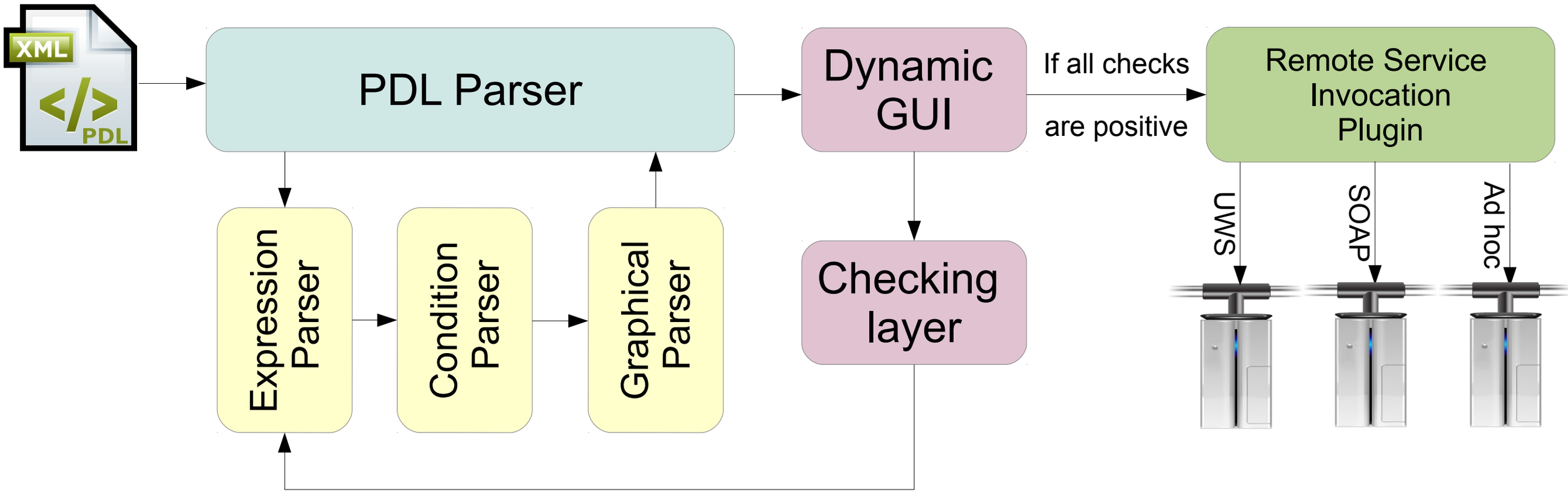


Major difficulty we have to deal with:

PDL sentences

- Could be very complex;
- Could have an arbitrary (non predictable) number of elements.

Dynamic client & Checking layer software architectures



Major difficulty we have to deal with:

PDL sentences

- Could be very complex;
- Could have an arbitrary (non predictable) number of elements.

- All the Parsers are based on *Neuron Network* algorithms
- These are implemented using factory patterns
- The parser itself build the ad-hoc network for parsing the expression
- Every neuron performs an atomic interpretation

The existing services

Three existing services are exposed using the PDL framework

The existing services

Three existing services are exposed using the PDL framework

- The Meudon PDR Code

PDR CODE | PDR DATABASE | PDR TOOLS | TIPS | DOCUMENTATIONS | CREDITS

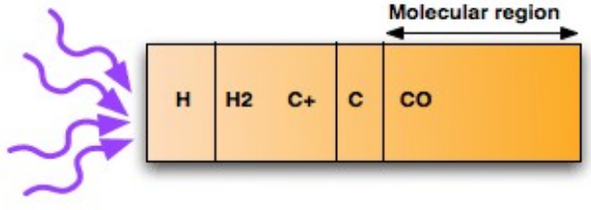
The Meudon PDR code

Description

The **Meudon PDR code** computes the atomic and molecular structure of interstellar clouds.

The code considers a stationary plane-parallel slab of gas and dust illuminated by a radiation field coming from one or both sides of the cloud. The incident radiation field can be the Interstellar Standard Radiation Field (ISRF) and/or a star.

It solves at each point in the cloud, the radiative transfer in the UV taking into account the absorption in the continuum by dust and in discrete transitions of H and H2. The model computes the thermal balance taking into account heating processes such as the photoelectric effect on dust, chemistry, cosmic rays, etc. and cooling resulting frominfrared and millimeter emission of the abundant species.
Chemistry is solved for any number of species and reactions.



Use the Meudon PDR code

October 2011 : Release of **PDR code version 1.4.3**

- Download and install the code

Recent changes : **Mon, 06 Feb 2012**
Latest revision : **662**

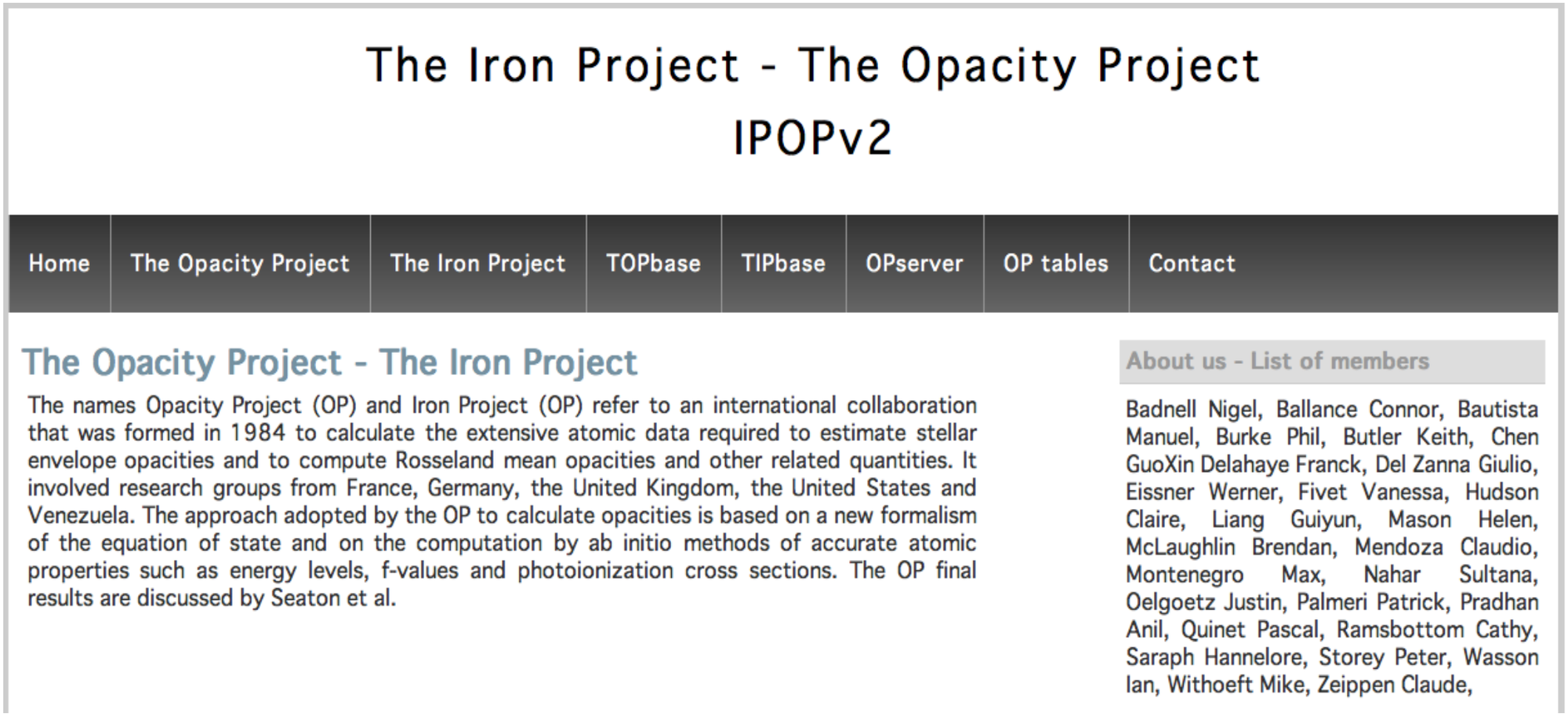
- Follow recent modifications on the track

Chemistry

The existing services

Three existing services are exposed using the PDL framework

- The Meudon PDR Code
- The opacity table service from the *Opacity Project*



The existing services

Three existing services are exposed using the PDL framework

- The Meudon PDR Code online
- The opacity table service from the *Opacity Project*
- The Meudon Broadening Stark H computation service

Hydrogen Stark broadening calculation for astrophysical applications

The Stark line profiles are calculated using the Model Microfield Method of Brissaud and Frisch (J. Quant. Spectrosc. Rad. Transfer 11, 1767 (1971)) which allows to include in a natural way the effects of the motions of the plasmas ions.

Parameters

Email :

Initial level :

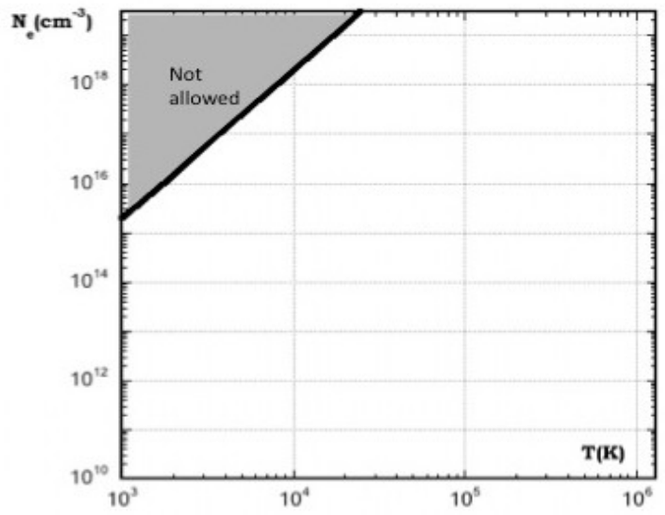
Final level :

Temperature :

Density :

submit

The parameters must verify the Debye approximation : $0.09*Ne(cm^{-3})^{1/6}/T(K)^{1/2} < 1$



Allowed Parameters in N_e & T
(grey zone corresponds to the condition : r_0< Debye)

The existing services

Three existing services are exposed using the PDL framework

- The Meudon PDR Code online
- The opacity table service from the *Opacity Project*
- The Meudon Broadening Stark H computation service

For these services

- Computations are remotely driven using Java Servlet technologies
- Results
  - are "*buffered"* for avoiding expensive recalculations
  - are sent by e-mail to user

The PDL remote service invocation plugin call the servlet with the parameters coming from the PDL client