REX 5ième journée Loops

Outils présentés

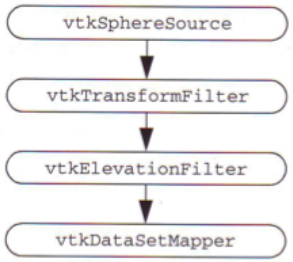
- Prérequis: suite Anaconda
  https://store.continuum.io/cshop/anaconda/
- Outils de visualisation des données
  - VTK
  - vispy
- Outil pour créer une interface graphique
  - pySide
- HFD5
- Bonus Ipython notebook
- LOOPS présentations et tutoriaux:
  http://reseau-loops.github.io/journee_2014_06.html

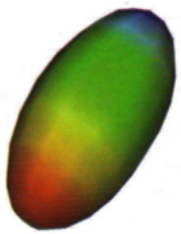
VTK

- Présentation par Sylvain Faure, labo Math Orsay
- Logiciels gratuits Paraview, Visit
- Visualization Toolkit (VTK)
  - Bibliothèques C++
  - 2D, 3D
  - //
  - développement d'applications spécifiques, voire de nouvelles classes
  - Package python
  - http://www.vtk.org/

VTK

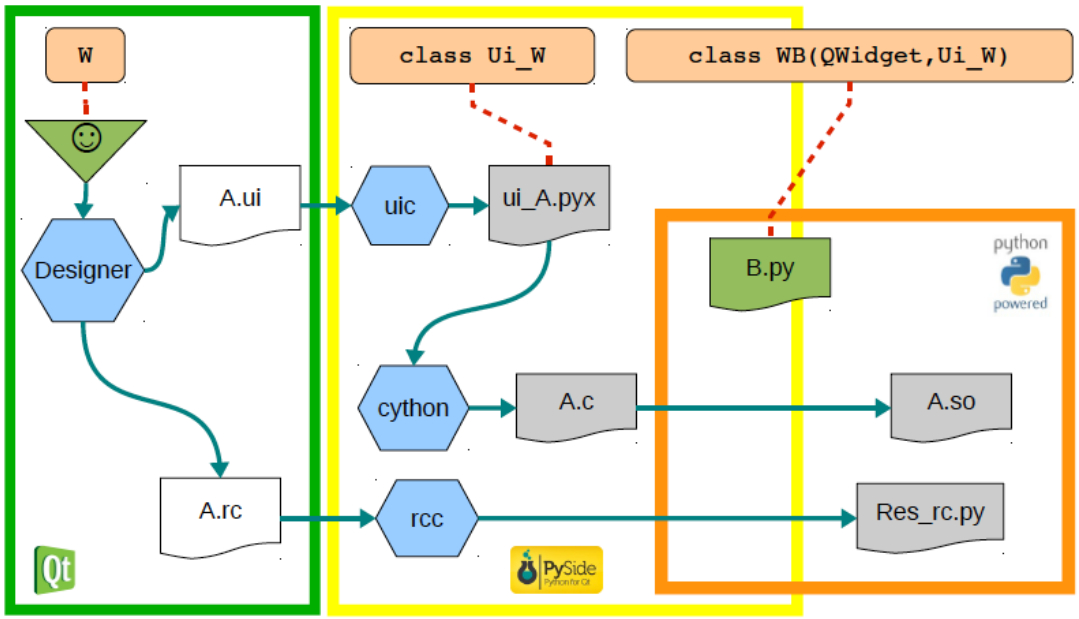
- Structures de données
  - Domaine (maillage) défini par des points (vtkPoints) et des cellules (vtkCells)
  - Données (P,v…) dans des objets de type Data (vtkPointData, vtkCellData)
- Nombreux formats de fichiers compatibles pour créer une structure de données VTK
  - Fichiers VTK : vtkXMLImageDataReader (.vti),
    - vtkXMLStructuredGridReader (.vts),
    - vtkXMLPolyDataReader (.vtp),
    - vtkXMLUnstructuredGridReader (.vtu) ...
  - Autres fichiers : vtkAVSucdReader, vtkPNGReader, vtkPDBReade, vtkNetCDFCFReader , vtkOpenFOAMReader, vtkEnSightReader,vtkFLUENTReader,...
- Notion de pipeline de visualisation et de filtres à connecter entre eux
- Notion de scène, de rendu
- Interactivité





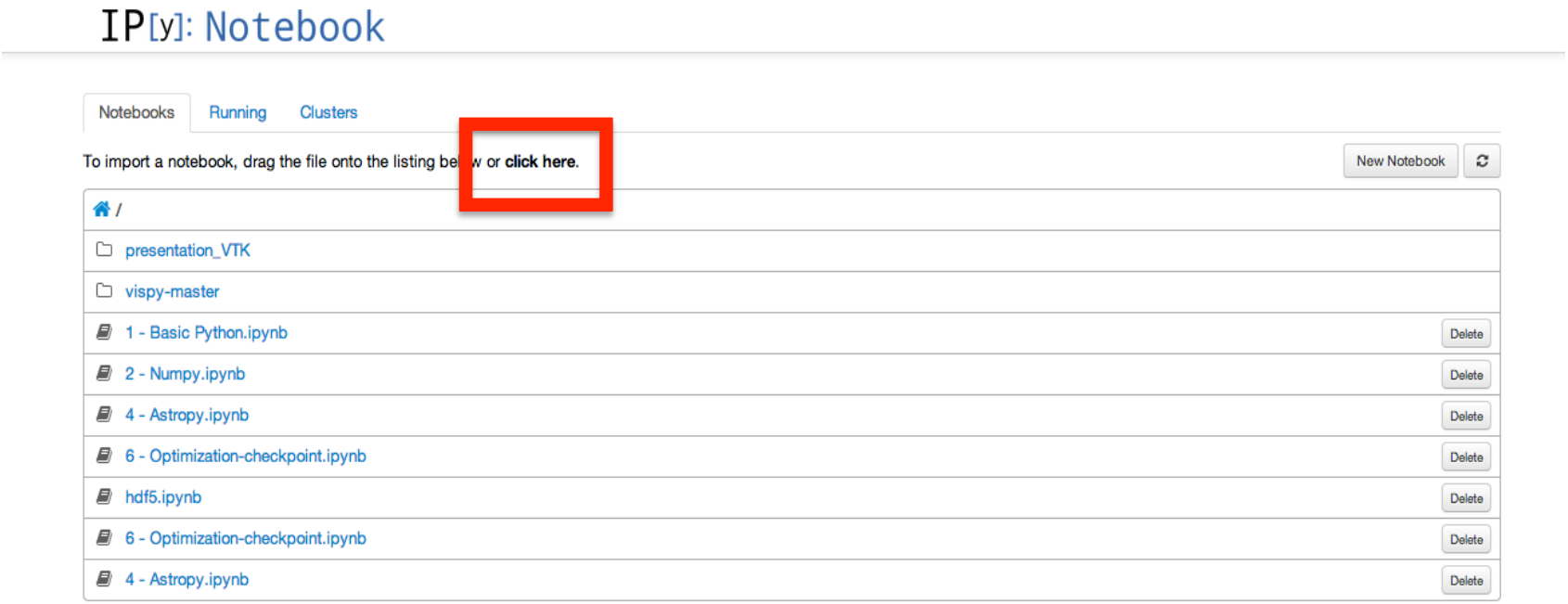
pySide

- Présentation de Marc Poinot (ONERA)
- conda install pyside (pas par défaut dans la suite anaconda)
- Principe de base: Qt (librairies C++)
- Python binding: pyQt / pySide
- Notion de production process designer/cython/python



HDF5 1/2

- présentation de Cyrille Rossant, University College London
- https://github.com/rossant/hdf5-tutorial
- extension .ipynb
- > ipython notebook



HDF5 2/2

- http://www.hdfgroup.org/HDF5/whatishdf5.html
- h5dump --help
- https://dpservis.wordpress.com/tag/hdf5/

« But what's important to understand here is the different scope of the two systems: a database handles efficiently large numbers of transactions consisting of small pieces of data. HDF5 handles one or only a few transactions consisssting of large amounts of data. It is important to identify the right tool for the right task. »

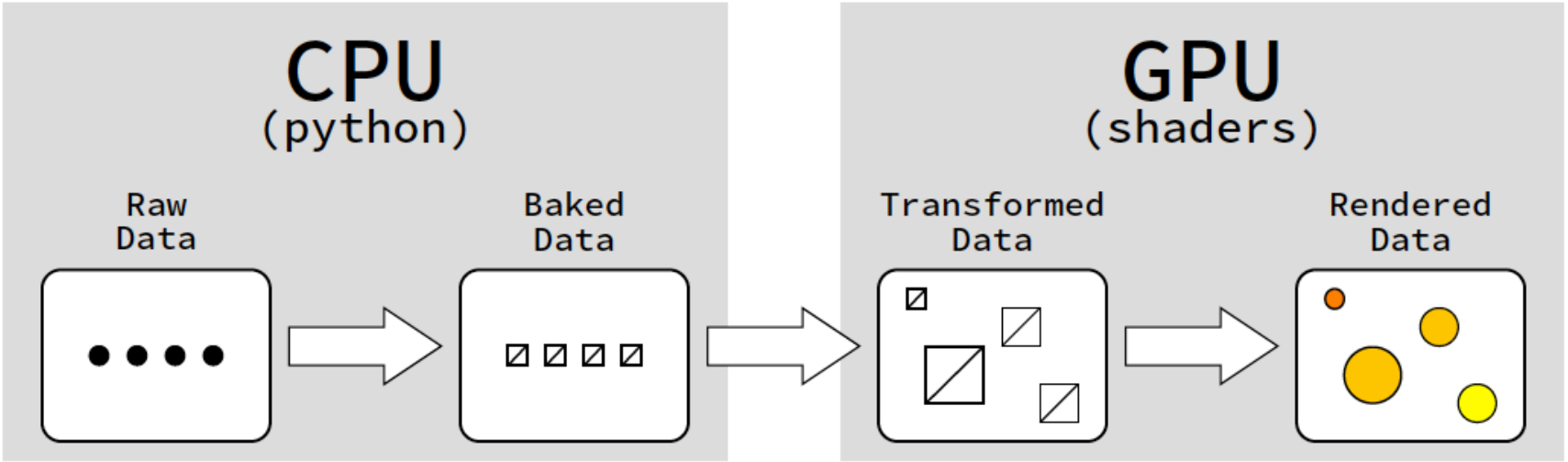
Conclusion

- Visualisation 3D:
  - cas IAS applicables? Planéto, solaire, cosmo
  - Comparaison avec l'existant.
- HDF5 en astro: utopique de détrôner les FITS ou les PDS
- Ipython notebook: idéal pour tutorial sur une nouvelle librairie.
  - Jake Vanderplas – Python in the Browser Age: Data exploration in the IPython Notebook : https://www.youtube.com/watch?feature=player_embedded&v=NzX7DDRkecU

VISPY

- Librairie de visualisation Python basée sur OpenGL
- ↑ Adapté pour visualiser, **explorer** un **gros volume** de données.
- ↓ API de bas niveau donc besoin de temps de développement (surtout concernant les parties shaders OpenGL en C-like)
- ↑ Librairie Matplotlib-like en cours de développement pour les utilisateurs scientifique.
- http://vispy.org/
- https://github.com/vispy

Pipeline Standard



Développements importants:

- baking (transformation de la structure des données pour la visualisation) si besoin
- codage des shaders

Shaders

Partie de programme OpenGL compilé par le GPU et exécuté pendant le rendu.

Shaders de bases:

- **vertex** shader

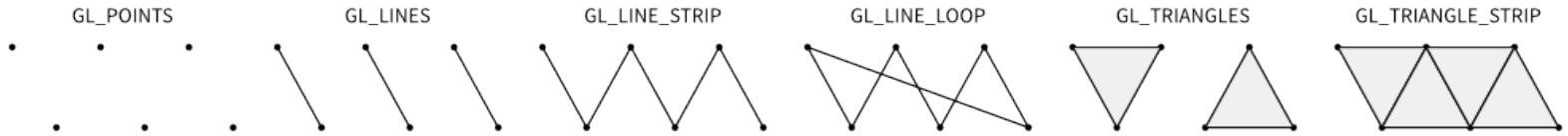
  positions des différents sommets.

```
void main(){
     gl_Position = vec4(0.0,0.0,0.0,1.0);
}
```

- **fragment** shader.

  couleur et textures entre les sommets (en fonction de la primitive de remplissage)

```
void main(){
     gl_FragColor = vec4(0.0,0.0,0.0,1.0);
}
```



2D and 3D Exemples