

# Le projet IMPEX

## Partager et analyser données observationnelles et simulées

Hess Sébastien<sup>1</sup>, M. Khodachenko<sup>2</sup>, E. Kallio<sup>3</sup>, V. Genot<sup>4</sup>, M. Gangloff<sup>4</sup>, R. Jarvinen<sup>3</sup>, L. Hakkinen<sup>3</sup>, F. Topf<sup>2</sup>, T. Al-ubaidi<sup>2</sup>, W. Schmidt<sup>3</sup>, R. Modolo<sup>1</sup>, I. Alexeev<sup>5</sup> and the IMPEX team.

<sup>1</sup>LATMOS, Université Versailles-St Quentin, France

<sup>2</sup> Space Research Institute, Austrian Academy of Science, Graz, Austria.

<sup>3</sup> Finnish Meteorological Institute, Helsinki, Finland.

<sup>4</sup> IRAP, CNRS/Université Paul Sabatier, Toulouse, France.

<sup>5</sup> Skobektsyn Institute of Nuclear Physics, Moscow, Russian Federation

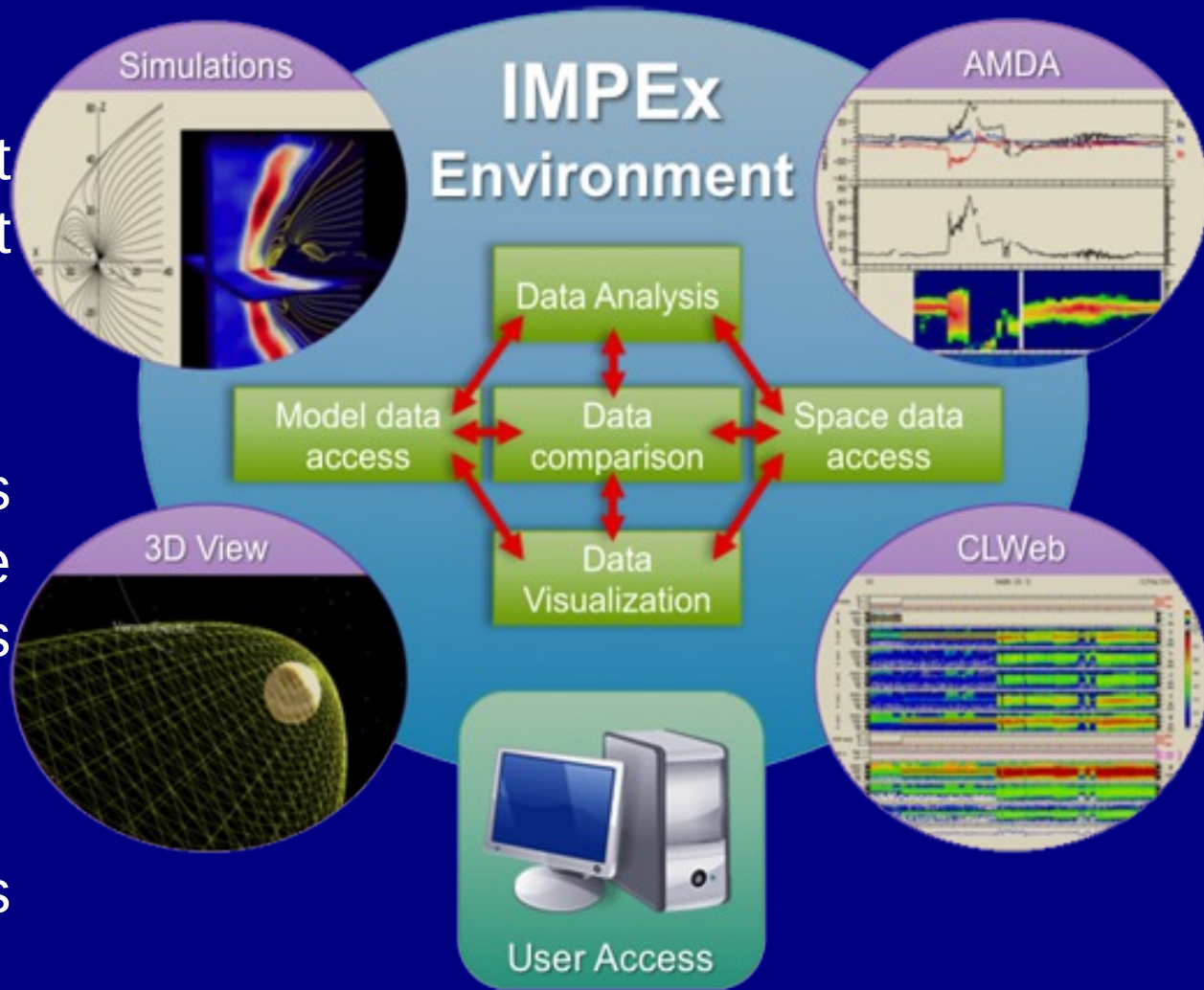
Programme Européen FP7.

Fournit un environnement simple et efficace d'outils et de bases de données.

où résultats de simulations et de modèles peuvent être comparés aux données observationnelles.

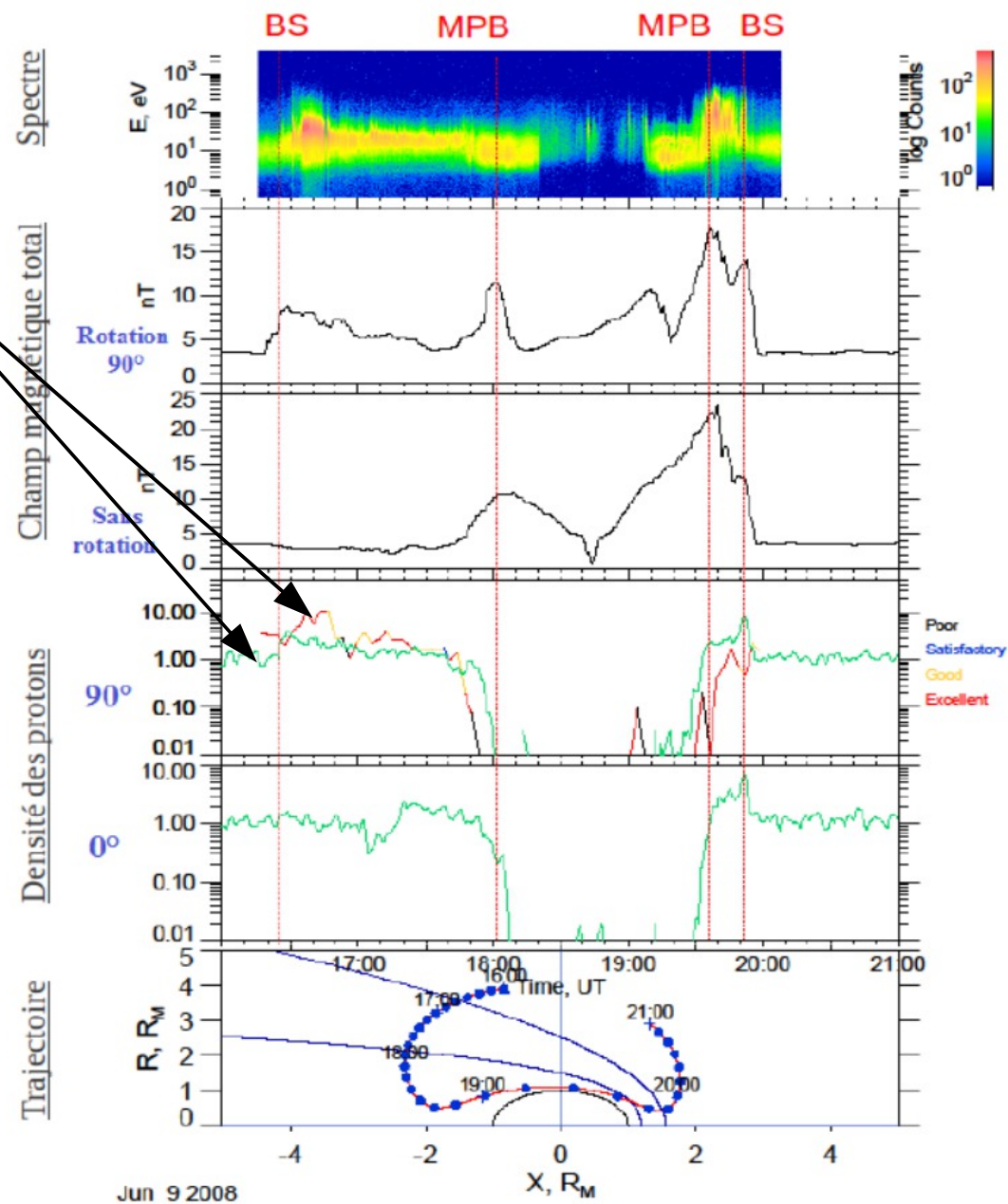
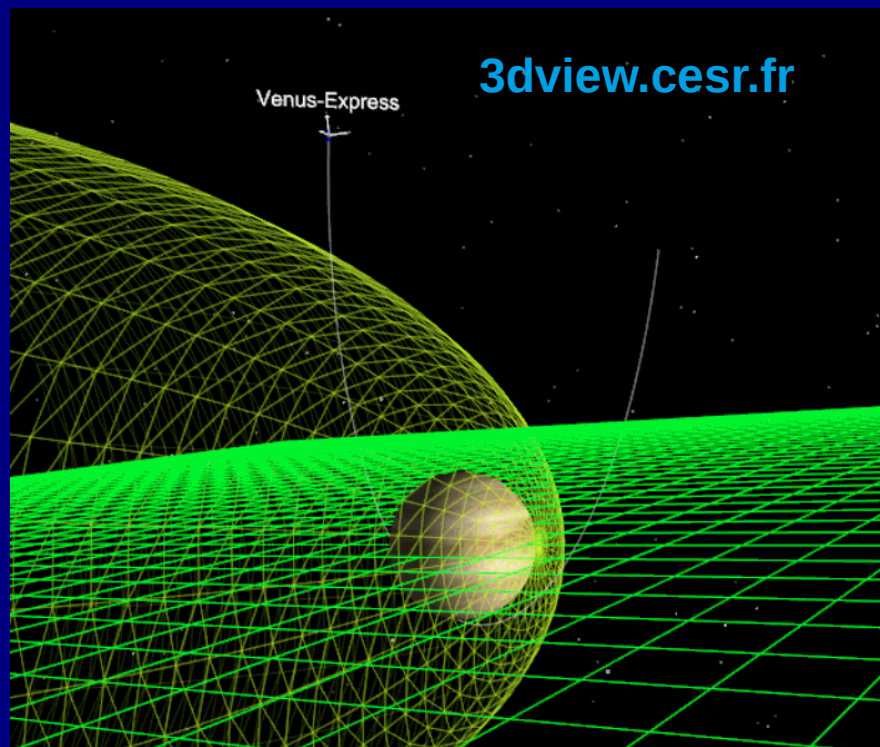
3 bases de données (LATMOS, FMI, SINP)

3 outils (CDPP)



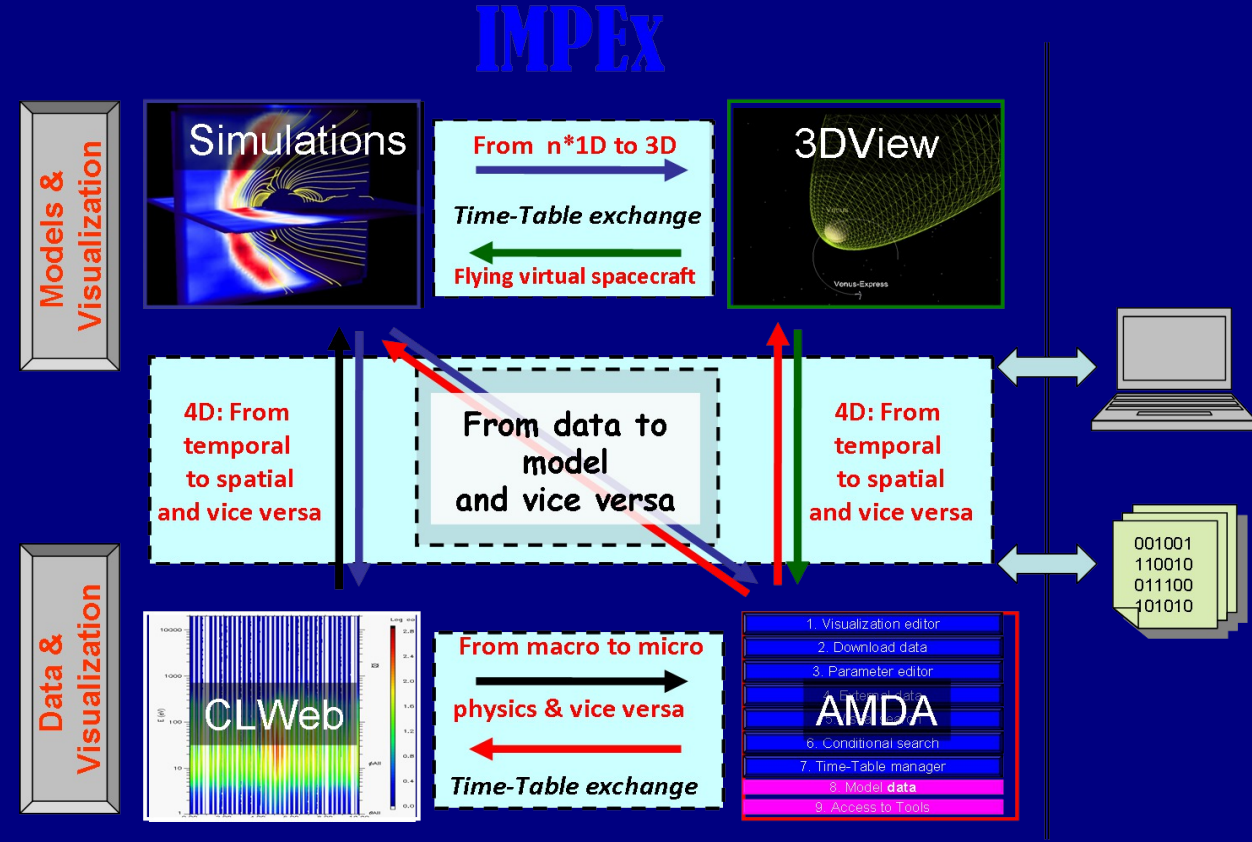
Comparaison données simulations pour plusieurs orientations de l'IMF

Visualisation de la trajectoire en 3D





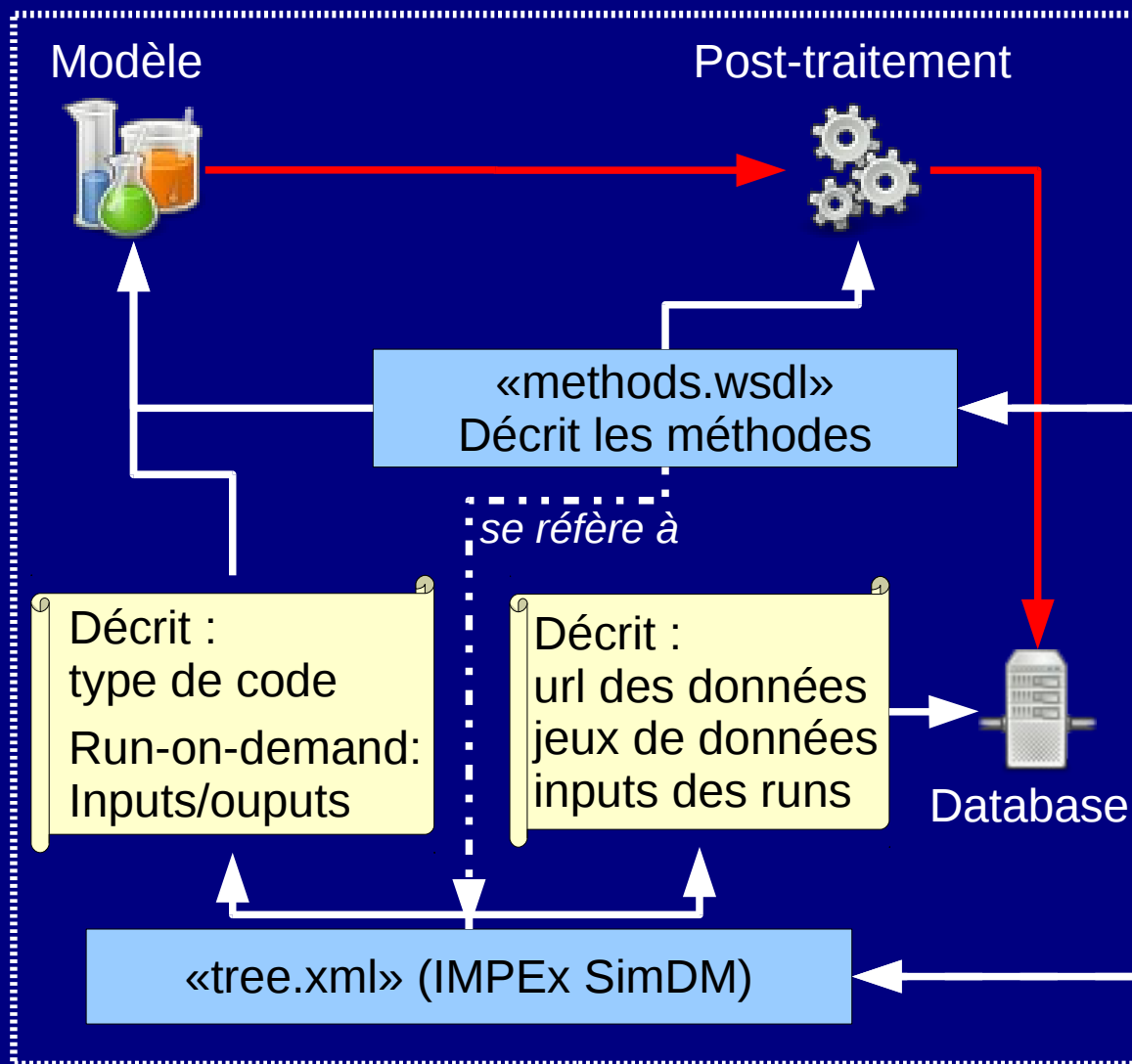
- Les outils doivent pouvoir :
- accéder aux données observationnelles
- accéder aux données simulées
- les mettre en relations  
⇒ Métadonnées décrivant :  
Les données  
Les inputs des runs
- Ils doivent aussi pouvoir demander :
- des post-traitements ad-hoc  
(interpolation sur une trajectoire,...)
- des Runs-On-Demand



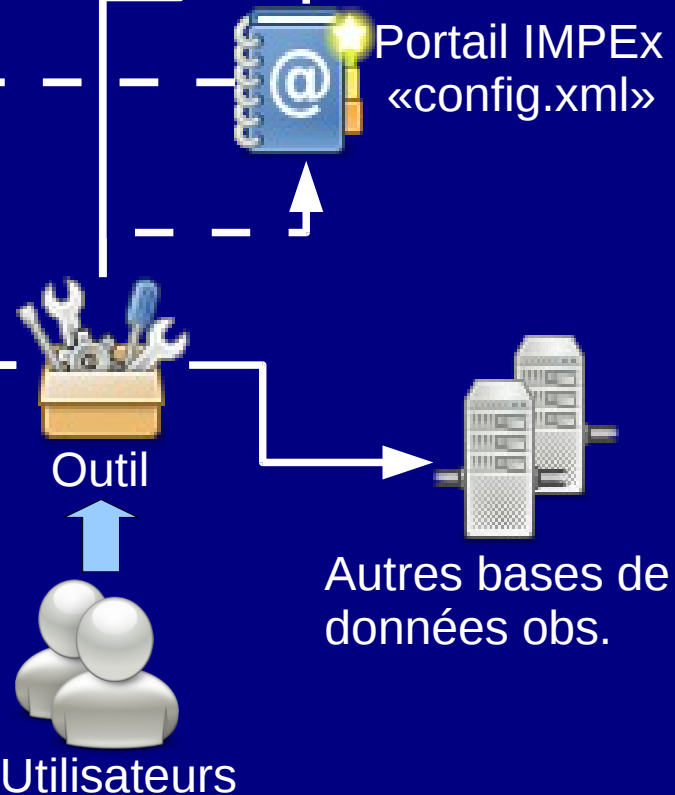
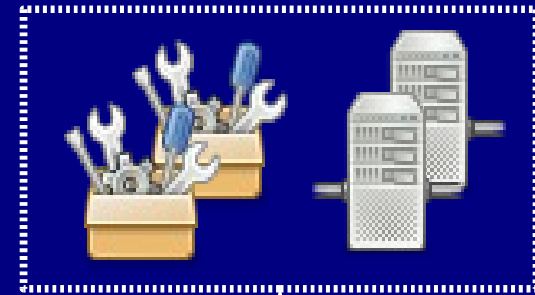
Pour cela nous avons développé :

- Un datamodel pour les métadonnées (IMPEX SimDM)
- Un jeu d'API pour les post-traitements et les RODs

## Détail d'une base de données IMPEX



## Autres outils et DB IMPEX



IMPEX Sim-DM est basé sur le data model SPASE. SPASE fournit :

Des éléments décrivant des ressources :

NumericalData, DisplayData,...

Observatory, Service, ...

Des éléments permettant de définir les propriétés de ces ressources (ID,URLs,Contents,...)

Dictionnaire de termes permettant de définir les valeurs de ces propriétés

Permet de décrire la plupart des données acquises par une sonde dans le domaine de la physique spatiale (time series).

Mais ne peut pas décrire les modèles et les données sur un domaine spatial.



# Impex Simulation Data Model



IMPEX Sim-DM est basé sur le data model SPASE. SPASE fournit :

Des éléments décrivant des ressources :

NumericalData, DisplayData,...

Observatory, Service, ...

Des éléments permettant de définir les propriétés de ces ressources (ID,URLs,Contents,...)

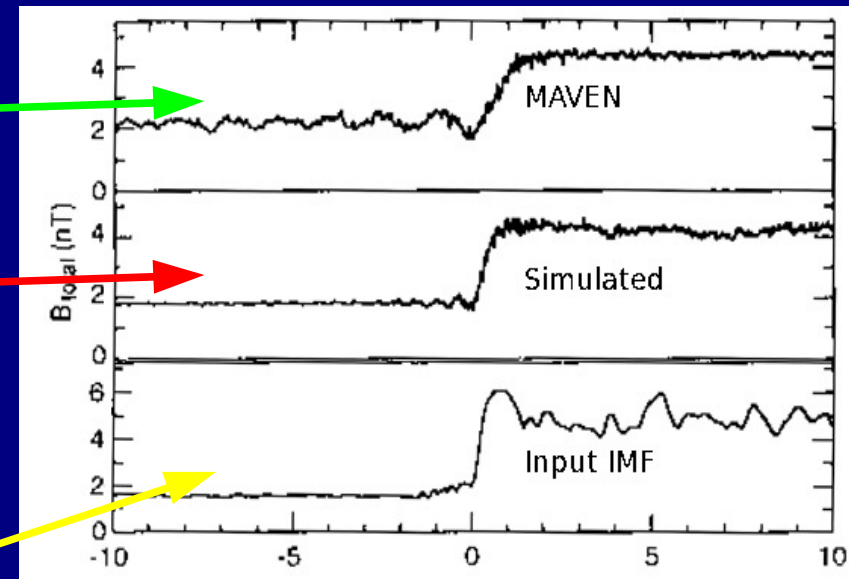
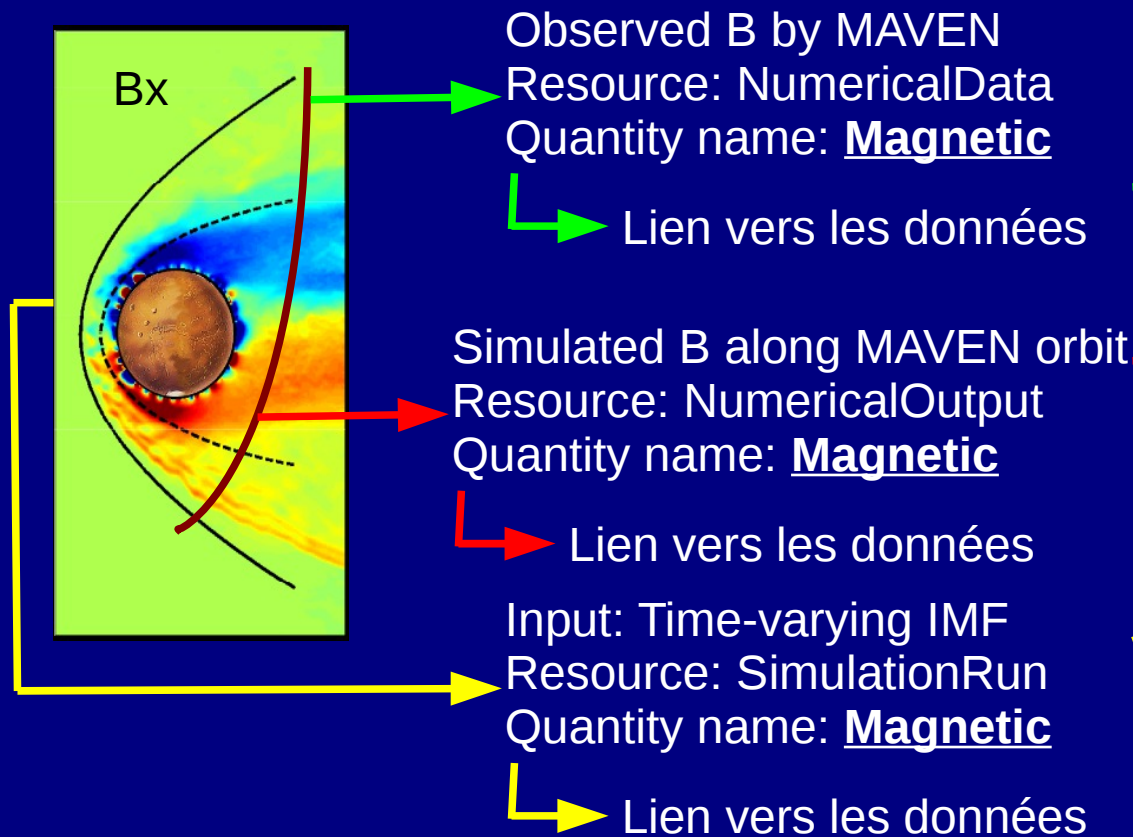
Dictionnaire de termes permettant de définir les valeurs de ces propriétés

**Garde les ressources SPASE mais en ajoute de nouvelles**  
*(Spase files are compliant with IMPEXDM)*

**Les éléments, termes et définitions de Spase sont gardées et utilisées pour décrire de nouvelles ressources. De nouveaux éléments sont créés, mais peu de nouveaux termes**  
*(ex: MagneticField signifie champ magnetique pour la description des données et des entrées de simulations.)*

Intérêt d'un dictionnaire limité et cohérent : données, entrées et sorties peuvent être traitées de la même manière.

Exemple : IMF variable dans le temps en entrée, et champ magnétique interpolé le long d'une trajectoire en sortie peuvent être traités de la même façon que des observations par un outil.



*(just an illustration, obviously not real data)*



## SPASE DM fournit des ressources

- Pour décrire les données et leur accès:
  - NumericalData,
  - DisplayData,
  - Granule
- Des métadonnées pour l'exploitation des données :
  - Catalog,
  - Observatory,...
- Pour décrire la structure de la base de données :
  - Repository,
  - Service, ...

## IMPEX Simulation DM ajoute :

- NumericalOutput, DisplayOutput  
clone Spase+ description domaine spatial  
et propriétés liées au post-traitement
- Granule (overrides Spase's)  
ajoute la description du domaine spatial
- SimulationModel  
Infos générales sur le modèle  
(pas sur les algorithmes)  
+ liste entrées/sorties pour les RODs
- SimulationRun  
Décrit les paramètres des simulations  
à l'origine des données diffusées

## SPASE DM fournit des ressources

- Pour décrire les données et leur accès:
  - NumericalData,
  - DisplayData,
  - Granule
- Des métadonnées pour l'exploitation des données :
  - Catalog,
  - Observatory,...
- Pour décrire la structure de la base de données :
  - Repository,
  - Service, ...

## IMPEX Simulation DM ajoute :

- NumericalOutput, DisplayOutput  
clones + description du domaine spatial  
et propriétés liées au post-traitement
- Granule (overrides Spase's)  
ajoute la description du domaine spatial
- SimulationModel  
Infos générales sur le modèle  
(pas sur les algorithmes)  
+ liste entrées/sorties pour les RODs
- SimulationRun  
Décrit les paramètres des simulations  
à l'origine des données diffusées

## Structure de la ressource SimulationRun

<SimulationRun>

<ResourceID />

ID dans la base de donnée

<ResourceHeader />

Header

<Model />

Lien vers le modèle (<SimulationModel>)

<SimulationTime />

Début, fin,...

<SimulationDomain />

Taille du domaine, système de coordonnées,...

<SimulatedRegion/>

Cible de la simulation

<InputParameter 1 />

Liste des paramètres de simulation

<InputParameter 2 />

....

</SimulationRun>



- Les paramètres correspondent à des quantités Spase et ont plusieurs propriétés. Ces paramètres peuvent être pré-définis ou libres.
- Les paramètres usuels ou utilisés pour des recherches automatiques ont des structures pré-définies: liste limitée de quantités (ex :Population= Electron, Ion,...)  
propriétés pré-définies (ex : Mass, Charge,... )
  - ▶ facilite l'analyse automatique par les outils.
- Paramètres libres : moins limités dans les quantités représentées  
Chaque propriétés est déclarée avec sa propre quantité Spase.
  - ▶ peut décrire presque n'importe quelle simulation.

▶ L'élément quantité permet aux outils de « comprendre » ce que représente le paramètre et chacunes de ses propriétés

*Parameter*

*Quantity=ActivityIndex*

*Property*

*Quantity=SolarUVFlux*

*Value=....*





```

<InputParameter>
<Name>Solar UV Flux index</Name>
<ParameterQuantity>ActivityIndex</ParameterQuantity>
<Property>
<Name>Solar F10.7 Flux</Name>
<PropertyQuantity>ActivityIndex</PropertyQuantity>
<PropertyValue>170</PropertyValue>
</Property>
<Property>
<Name>Solar F10.7 Flux Average</Name>
<PropertyQuantity>ActivityIndex</PropertyQuantity>
<PropertyValue>170</PropertyValue>
</Property>
</InputParameter>
    
```

```

<InputField>
<Name>IMF</Name>
<Description/>
<SimulatedRegion>Heliosphere</SimulatedRegion>
<FieldQuantity>Magnetic</FieldQuantity>
<Units>nT</Units>
<InputLabel>Bx By Bz</InputLabel>
<InputValue>1.634 2.516 0.0</InputValue>
</InputField>
    
```

```

<InputProcess>
<Name>H+ + H > H+ + H</Name>
<ProcessType>ChargeExchange</ProcessType>
<Units>cm2</Units>
<ProcessCoefficient>2.5e15</ProcessCoefficient>
<ProcessCoeffType>CrossSection</ProcessCoeffType>
</InputProcess>
    
```



## WebServices :

\_ pour obtenir des post-traitements ad-hoc (interpolation le long d'une trajectoire...)

Ex : getDataPointValue, ResourceID, Variable, url\_xyz, extraParams

↓ ↓  
Définis dans le «tree.xml»

↓  
URL du fichier contenant  
la trajectoire à interpoler

Rotation,  
Format,...  
définis dans  
le wsdl

\_ pour obtenir des Run-On-Demand

Ex : calculateDataPointValue, ResourceID, Variable, url\_xyz, extraParams

↓ ↓  
Définis dans le «tree.xml»

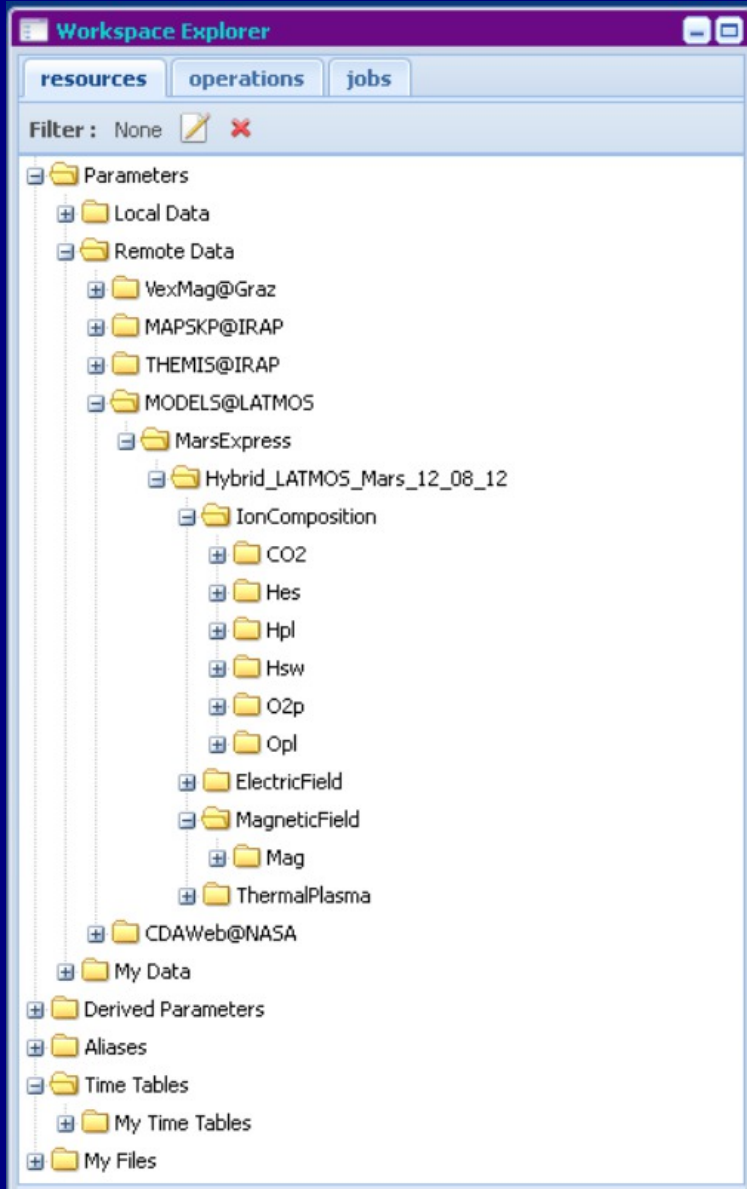
↓  
URL du fichier contenant  
la trajectoire à interpoler

↓  
Définis dans le «tree.xml»  
par des <Property> dans  
<SimulationModel>

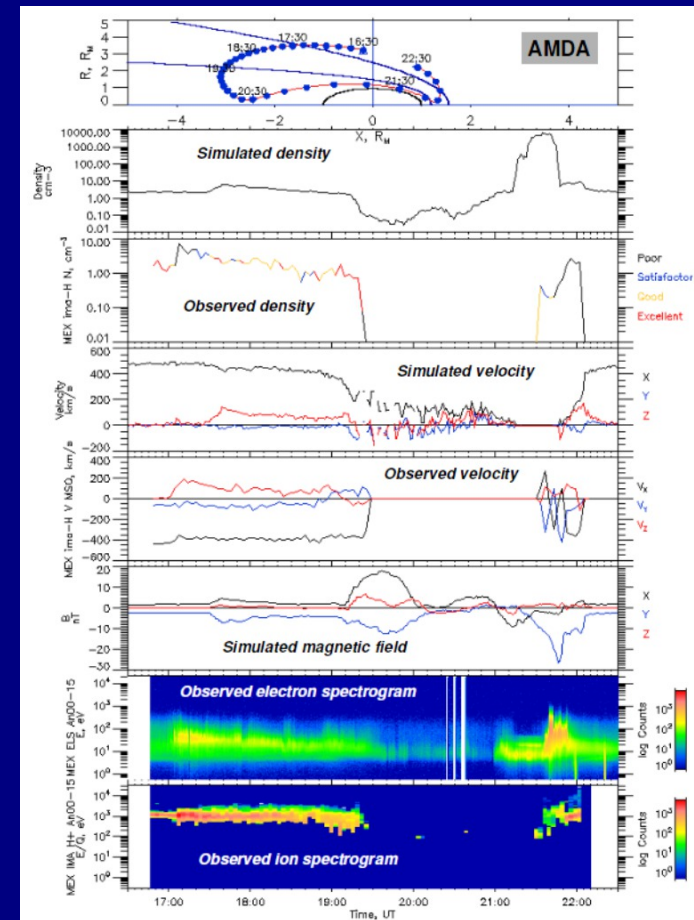


```
<SimulationModel>
  <ResourceID>
  <ResourceHeader />
  <Versions/>
  <SimulationType>Hybrid</SimulationType>
  <TemporalDependence>Yes</TemporalDependence>
  <SimulatedRegion>Earth</SimulatedRegion>
  <InputProperties>
    <Property>
      <PropertyName>SW_Velocity</PropertyName>
      <PropertyQuantity>FlowSpeed</PropertyQuantity>
      <Units>km/s</Units>
      <ValidMin>300</ValidMin>
      <ValidMax>1000</ValidMax>
    </Property>
    <Property>
      <PropertyName>SW_Density</PropertyName>
      <PropertyQuantity>Density</PropertyQuantity>
      <Units>cm-3</Units>
      <ValidMin>1</ValidMin>
      <ValidMax>100</ValidMax>
    </Property>
  </InputProperties>
  <OutputParameters />
</SimulationModel>
```

## LATMOS simulations in AMDA



Actual link to be implemented soon

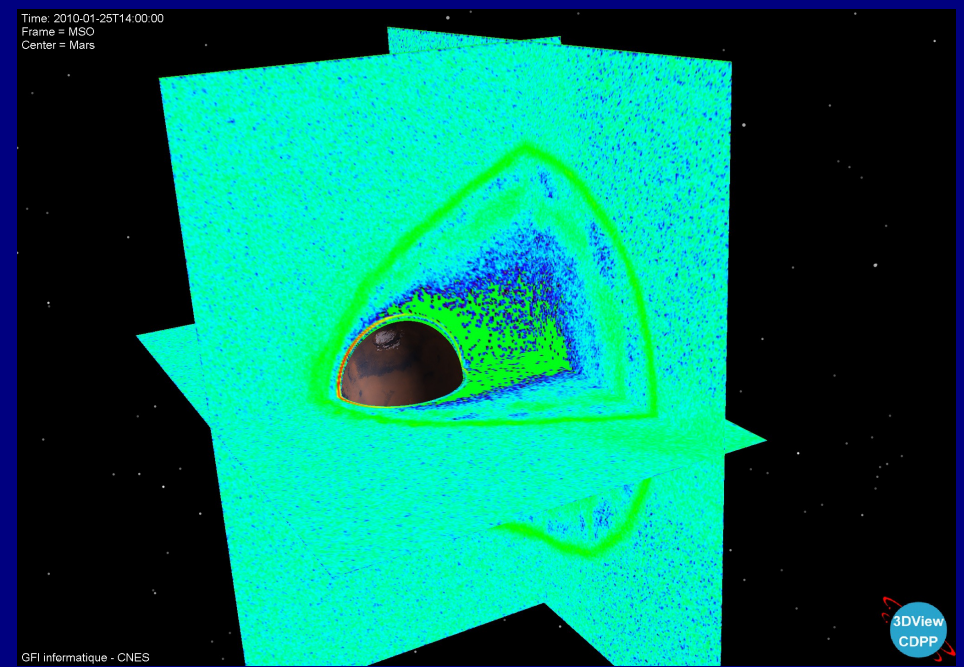
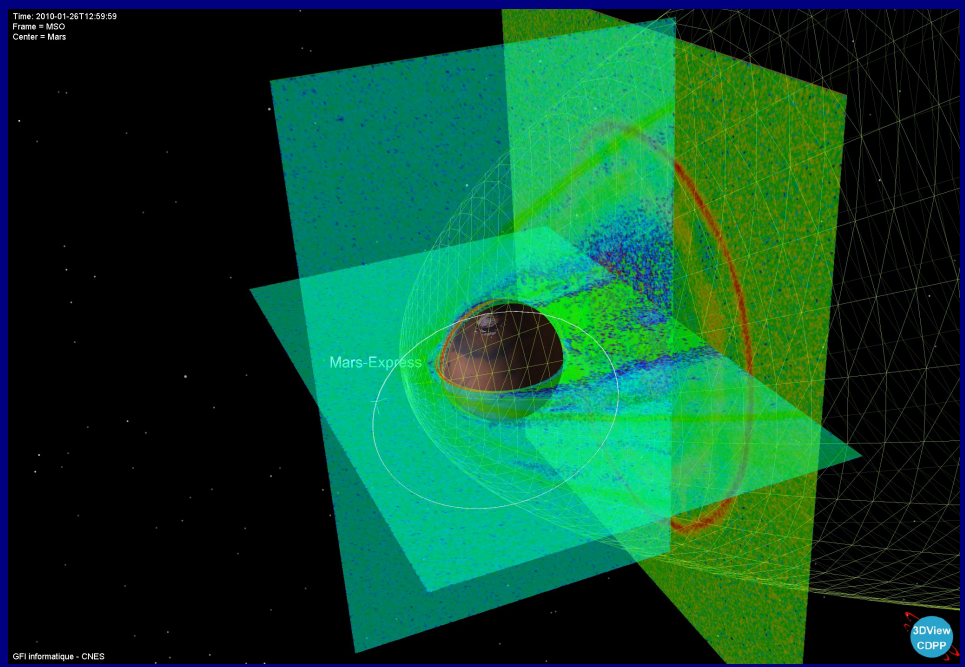








## LATMOS simulations in 3DView



*Prototype version*

## A développer : Connexion dynamique AMDA – 3DView

*Will use SAMP*

*Will use AMDA WebService*

**AMDA**

