



# Pôle de données et services “Surfaces Planétaires” Conversion des protocoles OGC / OV

## Contexte

- **Pôle de données et services CNES/INSU**
- **Périmètre : Surfaces Planétaires**
- **Phase A (juin 2021 - décembre 2023)**
  - Responsable scientifique : Frédéric Schmidt
  - Responsable technique : Jean-Christophe Malapert
  - Objectifs:
    - Besoins et contraintes
    - Architecture
    - Gouvernance

## La communauté des sciences planétaires

Domaine fortement interdisciplinaire, issue à l'origine de l'astronomie et des sciences de la Terre.

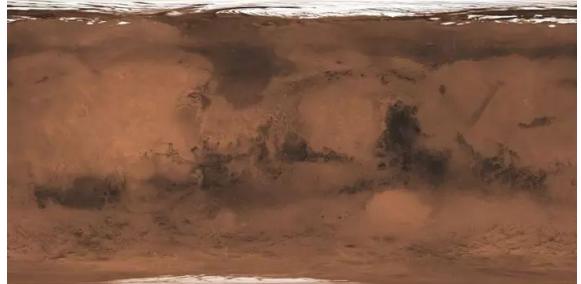
Les scientifiques planétaires sont généralement situés dans les départements d'astronomie et de physique ou de sciences de la Terre des universités ou des centres de recherche



**OGC<sup>®</sup>**  
Making location count.

[www.opengeospatial.org](http://www.opengeospatial.org)

# Type de données



Image

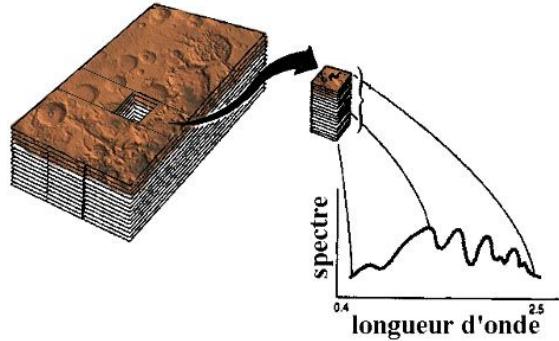
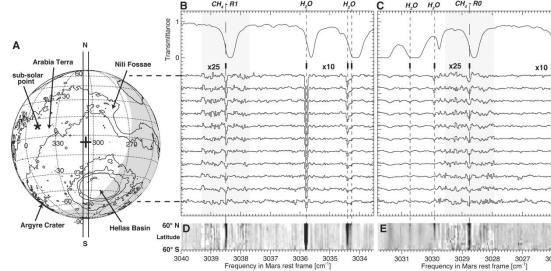


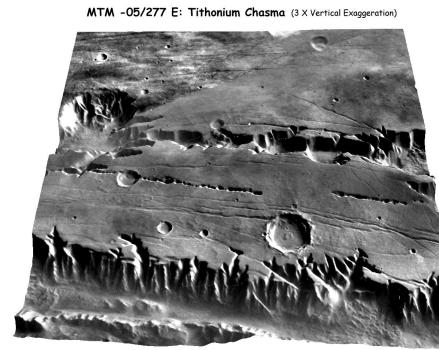
Image  
spectrale



Vecteurs



Spectroscopie



Modèle  
numérique de  
terrain

# Les problématiques

- **La découverte des données spatiales**
  - Lorsqu'un chercheur souhaite se lancer dans une nouvelle recherche, par où commence-t-il ?
  - Perte de temps
  - Reproduction des données existantes
  - Utilisation de données non pertinentes
- **Les standards**
  - Les utilisateurs veulent plus qu'un simple archive statique
  - Interopérabilité dans l'accès, les formats, les systèmes de coordonnées
- **Les données**
  - les systèmes de coordonnées géodésiques, élévation et orthoimagerie

# Objectifs du pôle

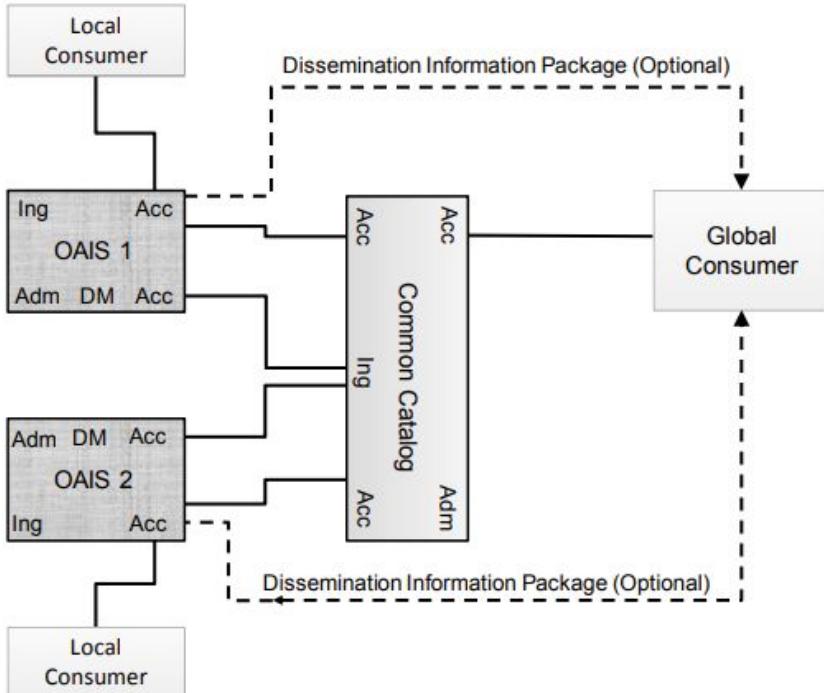
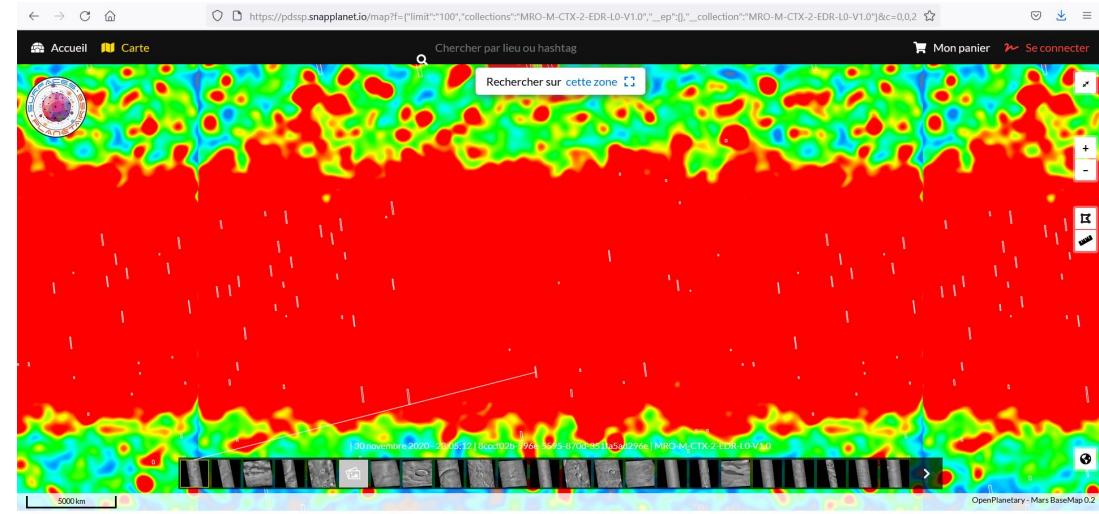
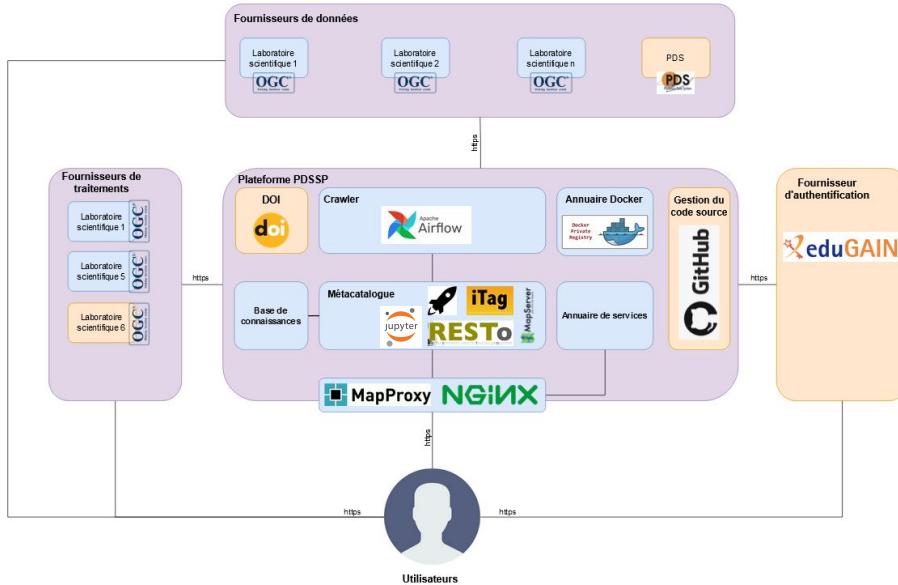


Figure 6-3: An OAIS Federation Employing a Common Catalog

- Accéder aux données de manière centralisée
- Mutualiser les moyens
- Fournir des services et outils à valeur ajoutée (visualisation, traitement)
- Utiliser les données sur le long terme (pérennisation)
- Promouvoir l'utilisation des standards d'interopérabilité, de formats
- Fédérer et animer une communauté scientifique

# Architecture pressenti du pôle - OGC & prototype



# Open Geospatial Consortium

## Who are our members?



The world's leading and comprehensive community  
of experts making location data more findable,  
accessible, interoperable and reusable

OGC

### Commercial

- Business Development
- Competitive Technical Advantage
- Global; Brand Exposure
- Funding for Innovation

### Government

- Innovation and Market Support
- Trusted Advice
- International Partnerships
- Operational Policy, Support, and Certification

### Research & Academia

- Applied Research Partners
- Funding for Innovation
- International Collaboration
- Citations

# Pourquoi les standards OGC ?

- **Maîtrise du CNES et des industriels dans ces standards**
  - Pont réalisable entre les réalisations/études CNES pour l'EO -> planétologie
- **Implementations (beaucoup - propriétaires & Open Sources)**
  - <https://www.ogc.org/resource/products/uncertified>
  - <https://www.ogc.org/resource/products/>
- **Standards répondant aux besoins**
  - Web Map Service (WMS) : <https://www.ogc.org/standards/wms>
  - Web Map Tile Service (WMPS) : <https://www.ogc.org/standards/wmts>
  - Web Coverage Service (WCS) : <https://www.ogc.org/standards/wcs>
  - Web Coverage Processing Service (WCPS) : <https://www.ogc.org/standards/wcps>
  - Web Processing Service (WPS) : <https://www.ogc.org/standards/wps>
  - SpatioTemporal Asset Catalog (STAC) : <https://stacspec.org/>

**W\*S & STAC => GeoAPI**

# Adaptation des standards OGC pour la planétologie

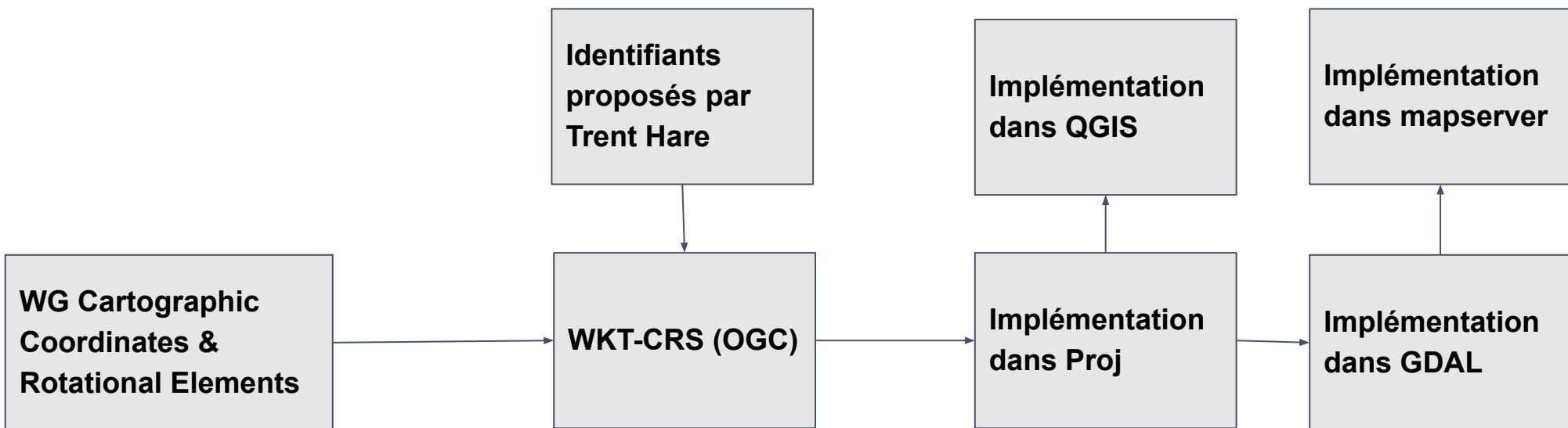


Problèmes :

- paramètre de requête pour le CRS (SRS=EPSG:xxxx)
- métadonnées relatives à WGS84

Responsables du groupe:

Jean-Christophe Malapert & Trent Hare



---

## Et la communauté IVOA ?

### Dans un premier temps : l'imagerie

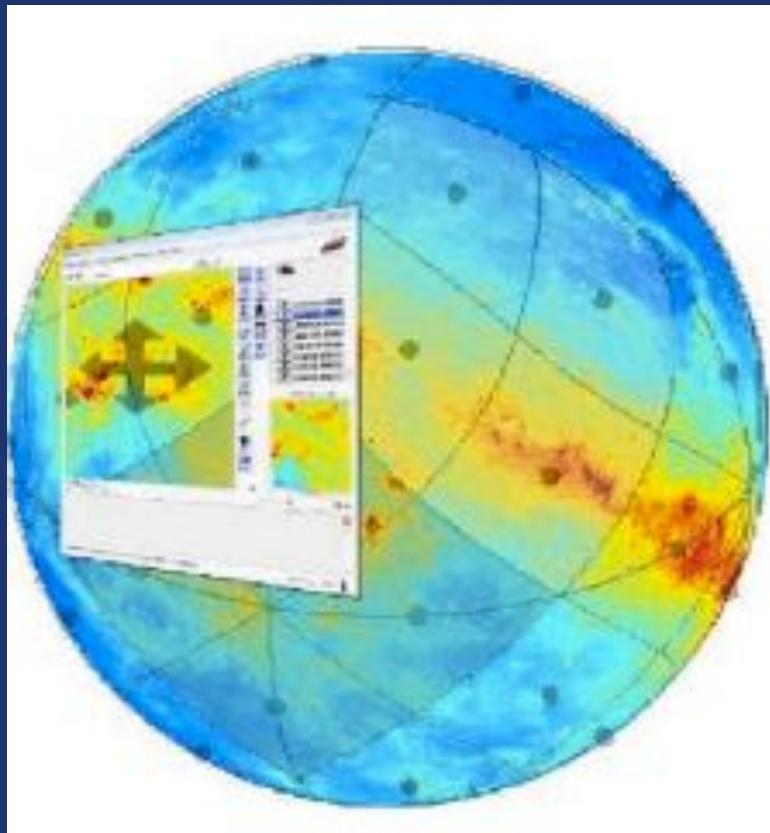
- HIPS produit par l'IVOA (International Virtual Observatory)
- WMS ou WMTS produit par l'OGC (Open Geospatial Consortium)

### Besoins:

- Comment établir une passerelle entre les 2 communautés et leurs protocoles, pour convertir protocole HIPS ↔ WMS/WMTS
- Réalisation d'un prototype de service de conversion pour que les outils utilisent des données venant de l'un ou l'autre standard de manière transparente

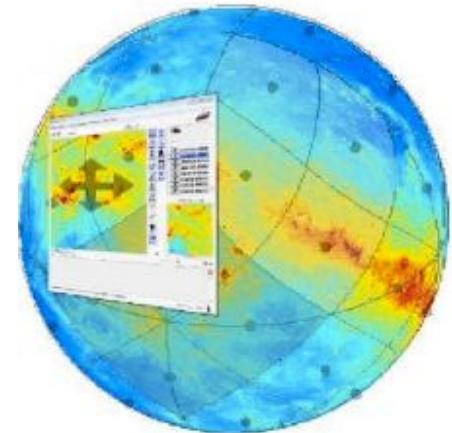
### Dans un second temps : la recherche

# HIPS



## Accès aux images HiPS

- Acronyme de Hierarchical Progressive Survey
- Système hiérarchique pour la description, le stockage et l'accès à des données d'observations du ciel, avec également des applications en planétologie
- <https://ivoa.net/documents/HiPS/20170519/REC-HIPS-1.0-20170519.pdf> « HiPS – Hierarchical Progressive Survey », Version 1.0, IVOA Recommandation du 19 mai 2017.
- Tuilage des régions du ciel (ou plus généralement de n'importe quelle sphère)
- Organisation hiérarchique permettant un raffinement de la résolution spatiale qui permet une visualisation progressive (zoom, déplacement)
- S'appuie sur une tessellation HEALPix de la sphère.



## Healpix : Nombre de cellules par niveau

- 12 cellules au niveau Norder=0 (Nside = 1)
- 48 cellules au niveau Norder=1 (Nside = 2)
- 192 cellules au niveau Norder=2 (Nside = 3)
- ...
- $12 * 2^{(2*Norder)}$  au niveau Norder

On définit aussi Nside=2\*Norder

L'aire d'une cellule au niveau Norder est de

$4 \pi / (12 * 2 ^{(2 * Norder)})$  stéradian.

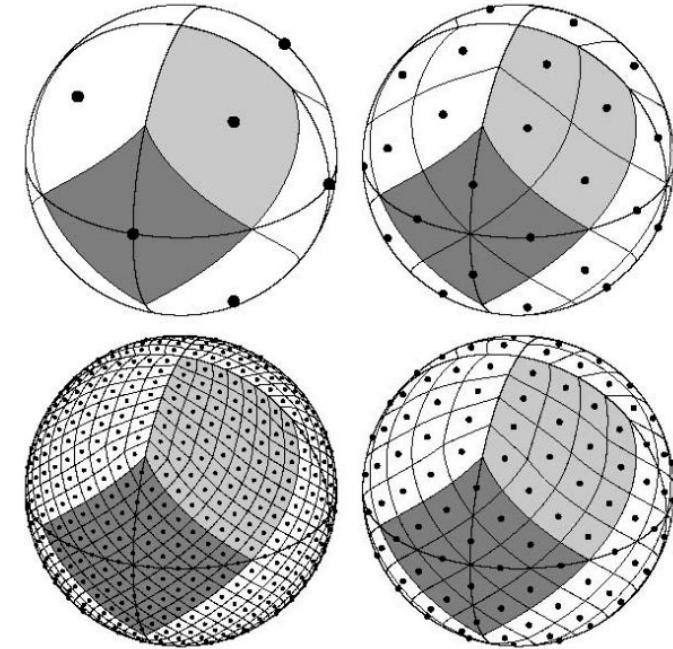
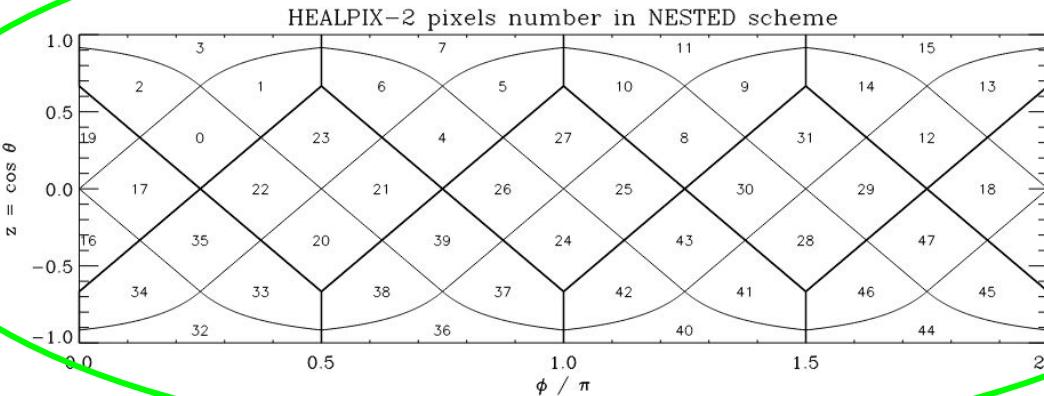
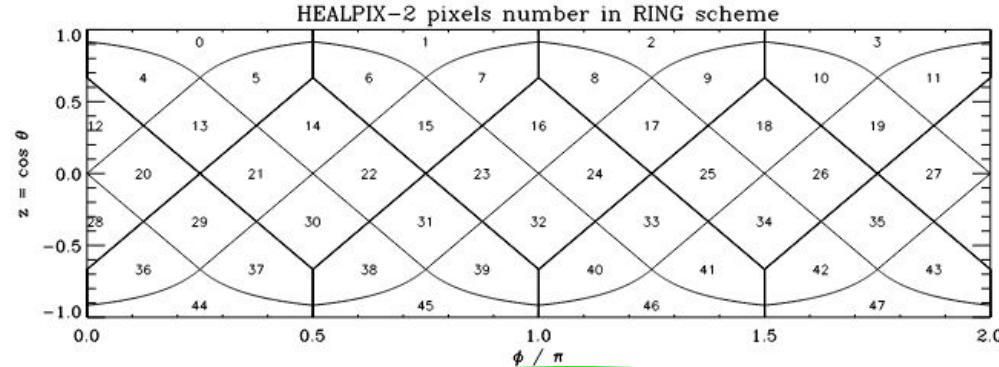


FIG. 3.—Orthographic view of the HEALPix partition of the sphere. The overplot of equator and meridians illustrates the octahedral symmetry of HEALPix. Light gray shading shows one of the 8 (4 north and 4 south) identical polar base-resolution pixels. Dark gray shading shows one of the 4 identical equatorial base-resolution pixels. Moving clockwise from the top left panel, the grid is hierarchically subdivided with the grid resolution parameter equal to  $N_{\text{side}} = 1, 2, 4, 8$ , and the corresponding total number of pixels equal to  $N_{\text{pix}} = 12 \times N_{\text{side}}^2 = 12, 48, 192, 768$ . All pixel centers are located on  $N_{\text{ring}} = 4N_{\text{side}} - 1$  rings of constant latitude. Within each panel the areas of all pixels are identical.

# Healpix : Numérotation des cellules



NESTED⇒  
utilisé par  
HIPS

## Points d'accès Hips

- **/properties: capacités du service**
- **/NorderK/DirD/NpixN{.ext}: service de tuile**
  - K: order HIPS (0, 1, 2 etc)
  - D: floor( $N/10000$ ) \* 10000
  - N: numéro de tuile [0,  $12 * 2^{(2*K)} - 1$ ]
- **/NorderK/Allsky{.ext}: prévisualisation aux ordres faibles**
- **Autres: catalogue de tuiles, fichier Moc.fits, fichier metadata**

## Fichier de properties

Exemple: [http://alasky.u-strasbg.fr/Planets/Mars\\_MOLA/properties](http://alasky.u-strasbg.fr/Planets/Mars_MOLA/properties)

creator\_did = ivo://CDS/P/Mars/MOLA-color

obs\_collection = Mars MOLA

obs\_title = Mars MOLA Shaded Relief / Colorised Elevation

hips\_version = 1.4

hips\_release\_date = 2019-05-21T06:53Z

hips\_frame = mars

hips\_order = 5

hips\_tile\_width = 512

hips\_status = public master clonableOnce

hips\_tile\_format = jpeg

dataproduct\_type = image

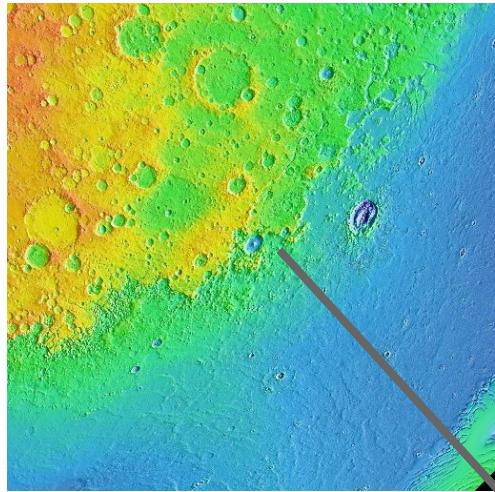
hips\_order\_min = 0

dataproduct\_subtype = color

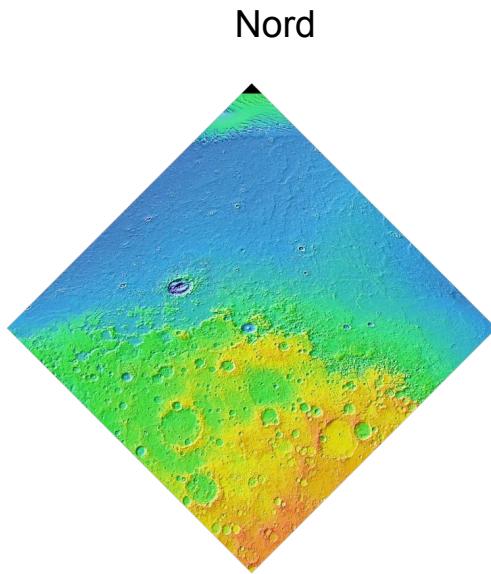
# autres propriétés omises

## Requête de tuile HIPS

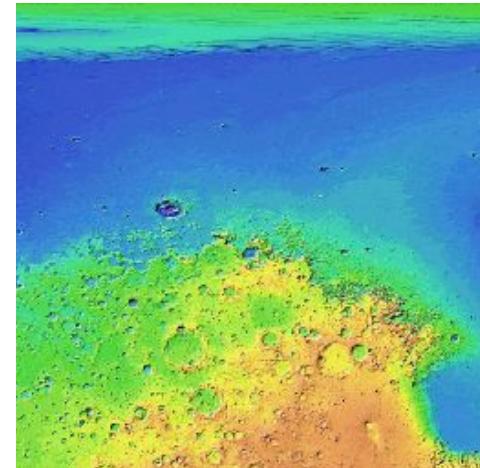
Exemple: [http://alasky.u-strasbg.fr/Planets/Mars\\_MOLA/Norder0/Dir0/Npix0.jpg](http://alasky.u-strasbg.fr/Planets/Mars_MOLA/Norder0/Dir0/Npix0.jpg)



Nord



Nord



Nord

# WMS



## OGC WMS : définition

- **Acronyme de Web Map Service**
- **Standard très mature (<https://www.ogc.org/standards/wms>)**
  - WMS 1.1.1: 2002
  - WMS 1.3.0: 2006
- **Requêtes:**
  - GetCapabilities: capacités du service: liste des couches, de leurs métadonnées, des styles disponibles, de l'étendue spatiale, des systèmes de référencement de coordonnées (Coordinate Reference System),
  - GetMap: obtenir une carte correspondant à une ou plusieurs couches, avec une étendue, un CRS, une taille d'images et un style spécifiés
  - autres: GetFeatureInfo, DescribeLayer, GetLegendGraphic
- **Standard très utilisé (cf <https://www.ogc.org/resource/products/byspec/?specid=107>)**
  - WMS 1.1.1: 483 implémentations serveur
  - WMS 1.3.0: 386 implémentations serveur
  - Sans compter les clients:
    - Légers: OpenLayers, Leaflet
    - Lourds: QGIS, etc.

# OGC WMS : GetCapabilities

Exemple:

[https://planetarymaps.usgs.gov/cgi-bin/mapserv?map=/maps/mars/mars\\_simp\\_cyl.map&SERVICE=WMS&VERSION=1.1.1&REQUEST=GetCapabilities](https://planetarymaps.usgs.gov/cgi-bin/mapserv?map=/maps/mars/mars_simp_cyl.map&SERVICE=WMS&VERSION=1.1.1&REQUEST=GetCapabilities)

Réponse:

```
<?xml version='1.0' encoding="ISO-8859-1" standalone="no" ?>

<WMT_MS_Capabilities version="1.1.1">

    <Service>

        <Name>OGC:WMS</Name>

        <Title>WMS Mars Server</Title>

        <Abstract>Planetary WMS service hosted by Astrogeology, USGS</Abstract>

        <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
xlink:href="https://planetarymaps.usgs.gov/cgi-bin/mapserv?map=/maps/mars/mars_simp_cyl.map&amp;"/>

        <ContactInformation>

        </ContactInformation>

    </Service>
```

&lt;Capability&gt;

&lt;Request&gt;

&lt;GetCapabilities&gt; [...] &lt;/GetCapabilities&gt;

&lt;GetMap&gt;

&lt;Format&gt;image/png&lt;/Format&gt;

&lt;Format&gt;image/jpeg&lt;/Format&gt;

[...]

&lt;Format&gt;image/tiff&lt;/Format&gt;

&lt;DCPType&gt;

&lt;HTTP&gt;

<Get><OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"  
xlink:href="[https://planetarymaps.usgs.gov/cgi-bin/mapserv?map=/maps/mars/mars\\_simp\\_cyl.map&](https://planetarymaps.usgs.gov/cgi-bin/mapserv?map=/maps/mars/mars_simp_cyl.map&)"></Get><Post><OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"  
xlink:href="https://planetarymaps.usgs.gov/cgi-bin/mapserv?map=/maps/mars/mars\_simp\_cyl.map&"></Post>

&lt;/HTTP&gt;

&lt;/DCPType&gt;

&lt;GetMap&gt;

[...]

&lt;/Request&gt;

```
<Layer>
  <Name>Mars_Simple_Cylindrical_Rasters</Name>
  <Title>WMS Mars Server</Title>
  <Abstract>Planetary WMS service hosted by Astrogeology, USGS</Abstract>
  <SRS>EPSG:4326</SRS>
  <SRS>IAU2000:49900</SRS>
  <LatLonBoundingBox minx="-180" miny="-90" maxx="360" maxy="90" />
  <BoundingBox SRS="EPSG:4326"
    minx="-180" miny="-90" maxx="360" maxy="90" />
  <Layer queryable="0" opaque="0" cascaded="0">
    <Name>MOLA_THEMEIS_blend</Name>
    <Title>MOLA Topography and THEMIS IR Day Blend</Title>
    <Abstract>MOLA Colorized DEM blended with the THEMIS IR Day 100m Global Mosaic v12[...]</Abstract>
    <LatLonBoundingBox minx="-180" miny="-89.9992" maxx="359.999" maxy="90" />
    <BoundingBox SRS="EPSG:4326"
      minx="-180" miny="-89.9992" maxx="359.999" maxy="90" />
  </Layer>
<...>
```

[...]

## OGC WMS : GetMap

Exemple:

<https://planetarymaps.usgs.gov/cgi-bin/mapserv?>

map=/maps/mars/mars\_simp\_cyl.map&

SERVICE=WMS&

VERSION=1.1.1&

REQUEST=GetMap&

LAYERS=MOLA\_color&

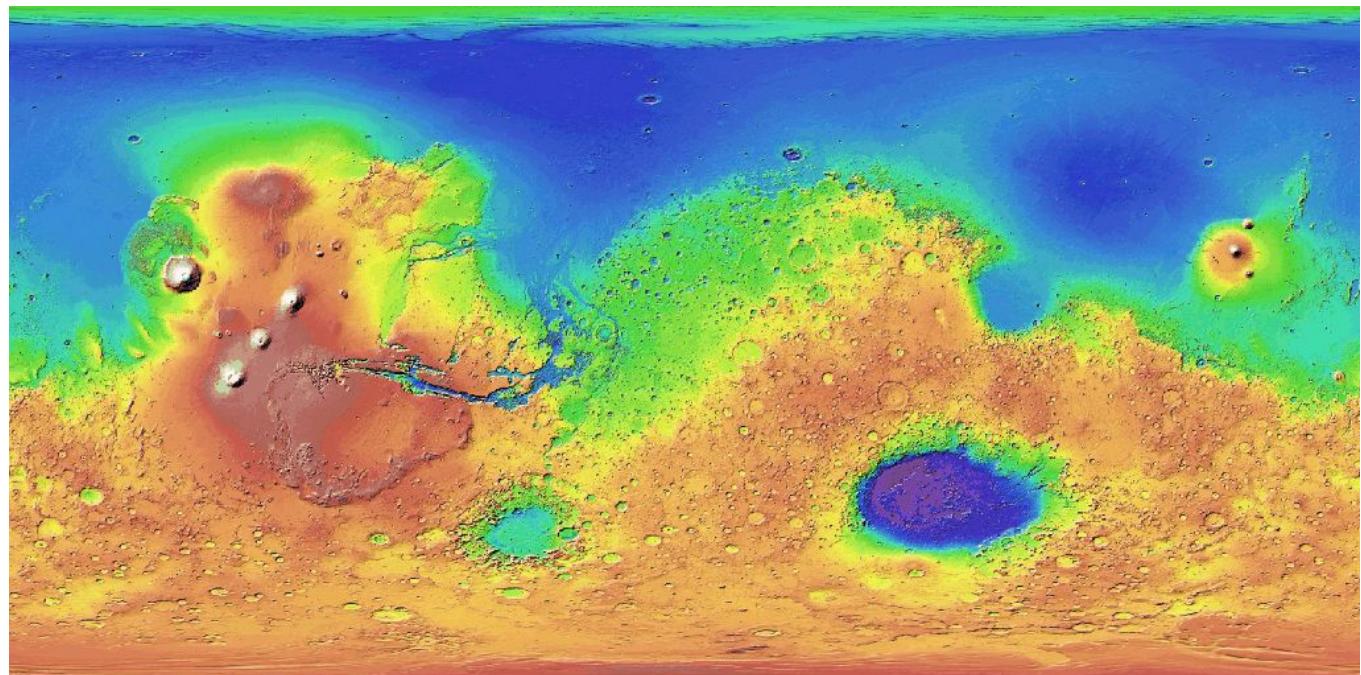
SRS=EPSG:4326&

BBOX=-180,-90,180,90&

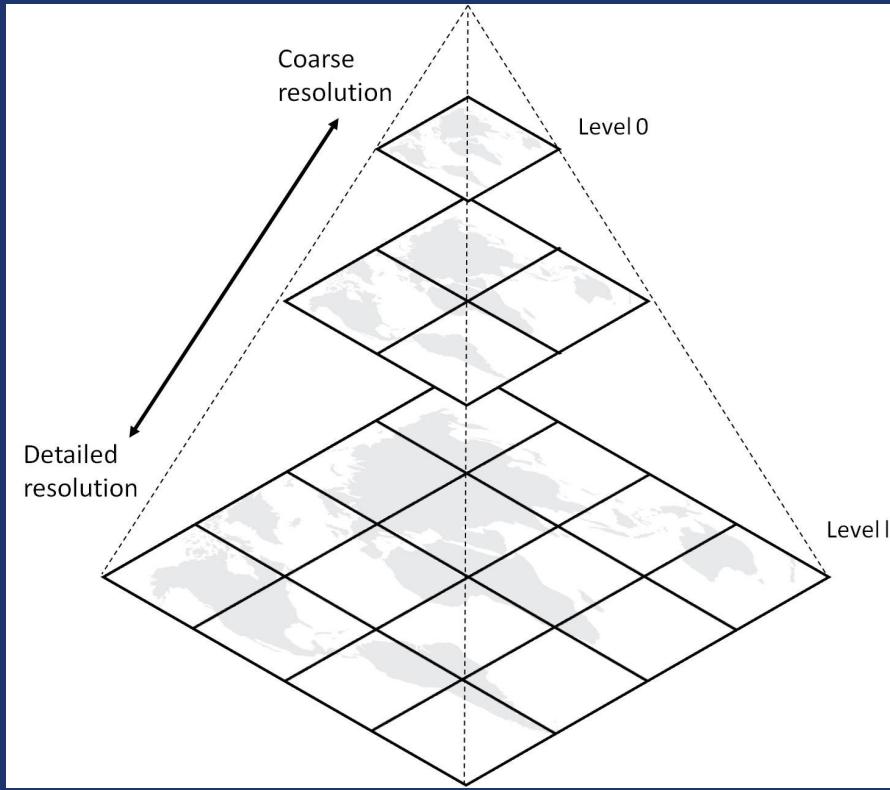
WIDTH=1024&

HEIGHT=512&

FORMAT=image/png



# WMTS



## OGC WMTS: définition

- Acronyme de Web Map Tile Service
- Plus récent que WMS:  
⇒ WMTS 1.0.0, 2007  
(<https://www.ogc.org/standards/wmts>)
- besoins de service de carte rapides comme ceux offerts par les protocoles XYZ Google Maps ou OpenStreetMap
- Se veut plus aisément à implémenter que WMS
- Restriction à un ou plusieurs systèmes de tuilage définis par l'administrateur du service.

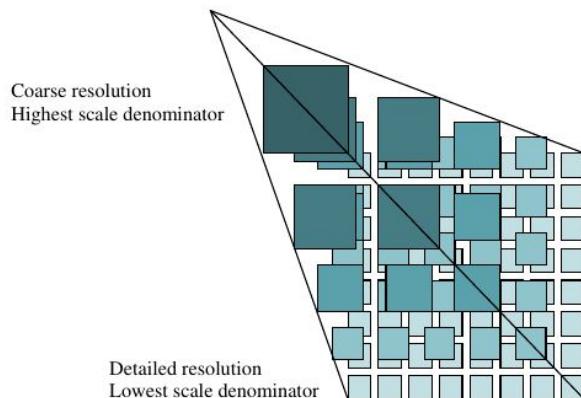
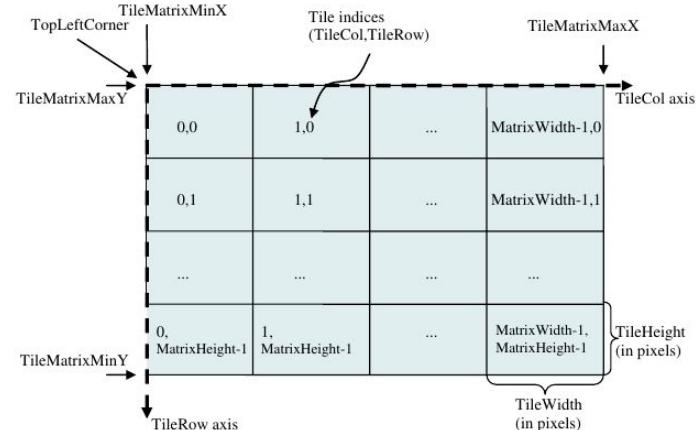


Figure 3 — Tile Matrix Set representation

- **Pré-génération de toute ou partie des tuiles possibles**
- **Profil RESTful permettant une implémentation potentielle avec serveur de fichiers statiques, sans code métier spécifique ⇒ idem pour HIPS**

# OGC WMTS : GetCapabilities

Exemple: <https://mrdata.usgs.gov/mapcache/wmts?SERVICE=WMTS&REQUEST=GetCapabilities>

Réponse:

```
<?xml version="1.0" encoding="UTF-8"?>
<Capabilities [...] version="1.0.0">
  <ows:ServiceIdentification>
    <ows:Title>MRData mapcache service</ows:Title>
    <ows:Abstract>Contains various services for MRData maps</ows:Abstract>
    <ows:ServiceType>OGC WMTS</ows:ServiceType>
    <ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
  </ows:ServiceIdentification>
  <ows:ServiceProvider>
    <ows:ServiceContact>
      <ows>ContactInfo/>
    </ows:ServiceContact>
  </ows:ServiceProvider>
```

```
<ows:OperationsMetadata>
  <ows:Operation name="GetCapabilities">
    [...]
  </ows:Operation>
  <ows:Operation name="GetTile">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="https://mrdata.usgs.gov/mapcache/wmts?">
          <ows:Constraint name="GetEncoding">
            <ows:AllowedValues>
              <ows:Value>KVP</ows:Value>
            </ows:AllowedValues>
          </ows:Constraint>
        </ows:Get>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
  <ows:Operation name="GetFeatureInfo">
    [...]
  </ows:Operation>
</ows:OperationsMetadata>
```

&lt;Contents&gt;

&lt;Layer&gt;

&lt;ows:Title&gt;mrds-LST&lt;/ows:Title&gt;

&lt;ows:WGS84BoundingBox&gt;

&lt;ows:LowerCorner&gt;-180.000000 -90.000000&lt;/ows:LowerCorner&gt;

&lt;ows:UpperCorner&gt;180.000000 90.000000&lt;/ows:UpperCorner&gt;

&lt;/ows:WGS84BoundingBox&gt;

&lt;ows:Identifier&gt;mrds-LST&lt;/ows:Identifier&gt;

&lt;Style isDefault="true"&gt;&lt;ows:Identifier&gt;default&lt;/ows:Identifier&gt;&lt;/Style&gt;

&lt;Format&gt;image/png; mode=8bit&lt;/Format&gt;

&lt;TileMatrixSetLink&gt;

&lt;TileMatrixSet&gt;WGS84&lt;/TileMatrixSet&gt;

&lt;/TileMatrixSetLink&gt;

&lt;TileMatrixSetLink&gt;

&lt;TileMatrixSet&gt;GoogleMapsCompatible&lt;/TileMatrixSet&gt;

&lt;/TileMatrixSetLink&gt;

&lt;ResourceURL format="image/png; mode=8bit" resourceType="tile"

template="https://mrldata.usgs.gov/mapcache/wmts/1.0.0/mrds-LST/default/{TileMatrixSet}/{TileMatrix}/{TileRow}/{TileCol}.png"/&gt;

&lt;/Layer&gt;

```
<TileMatrixSet>
  <ows:Identifier>GoogleMapsCompatible</ows:Identifier>
  <ows:BoundingBox crs="urn:ogc:def:crs:EPSG:6.3:3857">
    <ows:LowerCorner>-20037508.342789 -20037508.342789</ows:LowerCorner>
    <ows:UpperCorner>20037508.342789 20037508.342789</ows:UpperCorner>
  </ows:BoundingBox>
  <ows:SupportedCRS>urn:ogc:def:crs:EPSG:6.3:3857</ows:SupportedCRS>
  <WellKnownScaleSet>urn:ogc:def:wkss:OGC:1.0:GoogleMapsCompatible</WellKnownScaleSet>
  <TileMatrix>
    <ows:Identifier>0</ows:Identifier>
    <ScaleDenominator>559082264.02871787548065185547</ScaleDenominator>
    <TopLeftCorner>-20037508.342789 20037508.342789</TopLeftCorner>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>1</MatrixWidth>
    <MatrixHeight>1</MatrixHeight>
  </TileMatrix>
```

```
<TileMatrix>
  <ows:Identifier>1</ows:Identifier>
  <ScaleDenominator>279541132.01435887813568115234</ScaleDenominator>
  <TopLeftCorner>-20037508.342789 20037508.342789</TopLeftCorner>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>2</MatrixWidth>
  <MatrixHeight>2</MatrixHeight>
</TileMatrix>
[...]
<TileMatrix>
  <ows:Identifier>18</ows:Identifier>
  <ScaleDenominator>2132.72958384978392132325</ScaleDenominator>
  <TopLeftCorner>-20037508.342789 20037508.342789</TopLeftCorner>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>262144</MatrixWidth>
  <MatrixHeight>262144</MatrixHeight>
</TileMatrix>
```

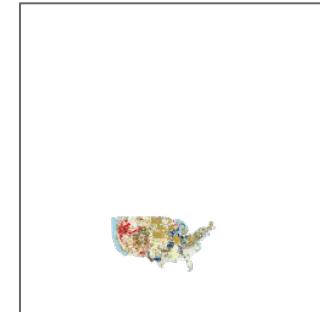
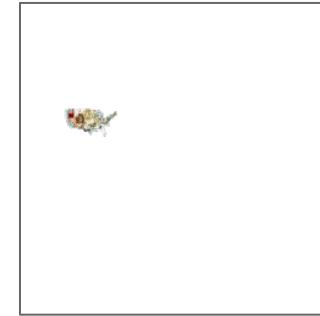
## WMTS: GetTile

Exemple:

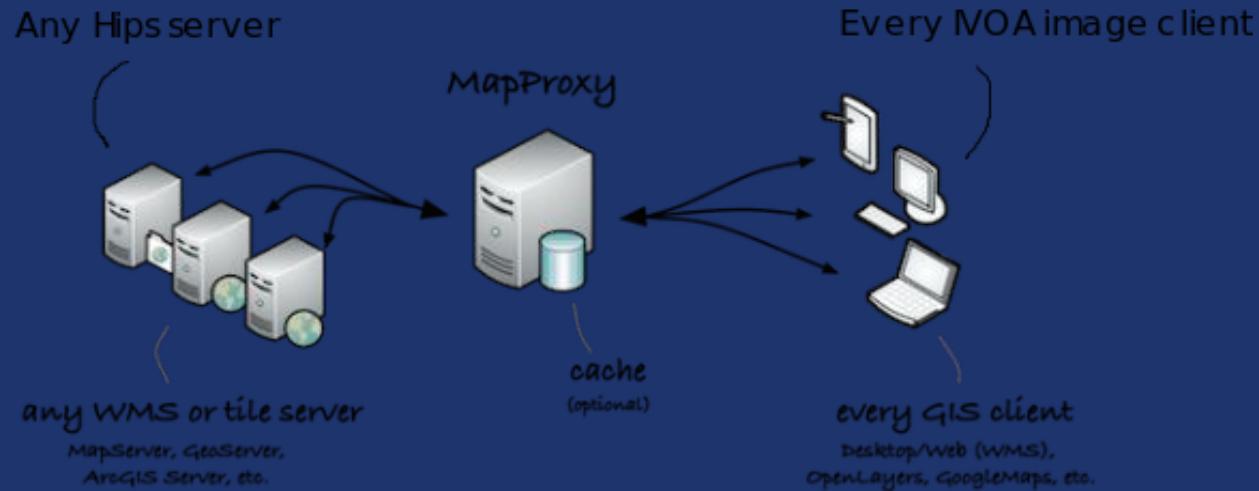
<https://mrdata.usgs.gov/mapcache/wmts/1.0.0/sgmc2/default/GoogleMapsCompatible/0/0/0.png>

<https://mrdata.usgs.gov/mapcache/wmts/1.0.0/sgmc2/default/GoogleMapsCompatible/1/0/0.png>

<https://mrdata.usgs.gov/mapcache/wmts/1.0.0/sgmc2/default/GoogleMapsCompatible/2/1/0.png>

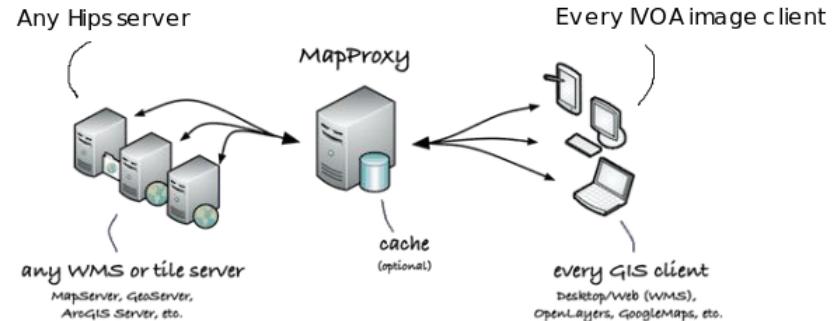


## Une solution



# Mapproxy

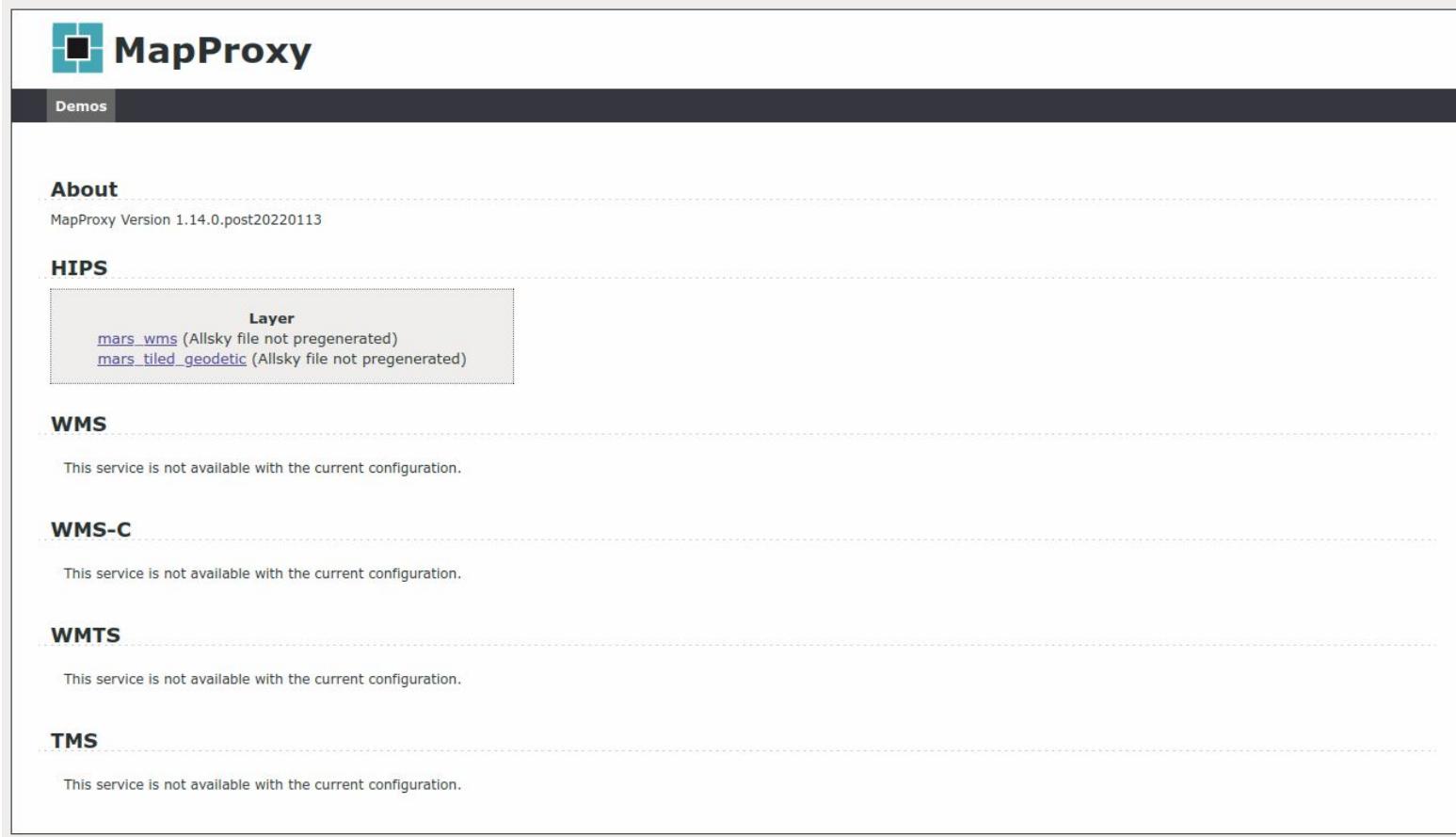
- **Mature (plus de 10 ans) et toujours maintenu**
- **Utilisé en production**
- **Documentation d'utilisation détaillée:**  
<https://mapproxy.org/docs/nightly/>
- **tests unitaires et d'intégrations.**
- **Architecture stable et claire, avec 3 composants:**
  - services (WMS, WMTS, KML, tuiles TMS, etc.),
  - caches (cache fichier, SQLite, Redis, etc.)
  - sources (WMS, WMTS/TMS, Google Maps, Bing/Virtual Earth, ArcGIS REST)
- **Python: utilisation aisée d'healpy**



## Développements réalisés:

- **Exposer un service OGC en tant que service HIPS. Utilisation de 2 clients de test**
  - client léger Javascript, Aladin Lite, qui s'intègre dans des pages HTML:  
<https://aladin.u-strasbg.fr/AladinLite>
  - un client « lourd », écrit en Java, Aladin Desktop :  
<https://aladin.cds.unistra.fr/AladinDesktop/>
- **Exposer un service HIPS en tant que service OGC**

# MapProxy: mise à disposition d'un service WMS via HIPS



**MapProxy**

Demos

**About**  
MapProxy Version 1.14.0.post20220113

**HIPS**

Layer

[mars\\_wms](#) (Allsky file not pregenerated)  
[mars\\_tiled\\_geodetic](#) (Allsky file not pregenerated)

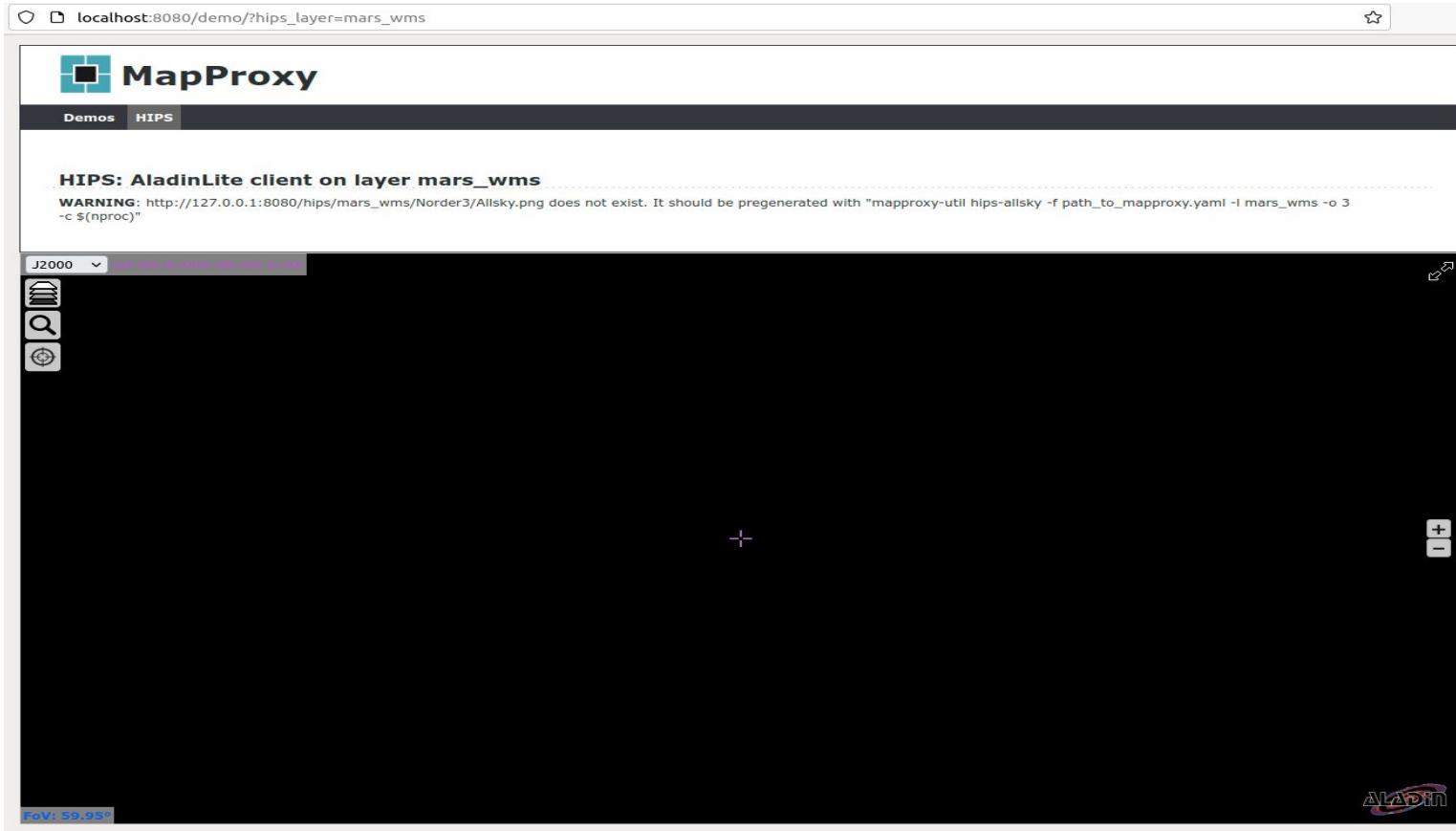
**WMS**  
This service is not available with the current configuration.

**WMS-C**  
This service is not available with the current configuration.

**WMPS**  
This service is not available with the current configuration.

**TMS**  
This service is not available with the current configuration.

# Service démo : visualiseur AladinLite



## Génération allSky

```
$ time mapproxy-util hips-allsky -f ./hips_examples/ogc_as_hips/mapproxy.yaml -l mars_wms -o 3 -c $(nproc)

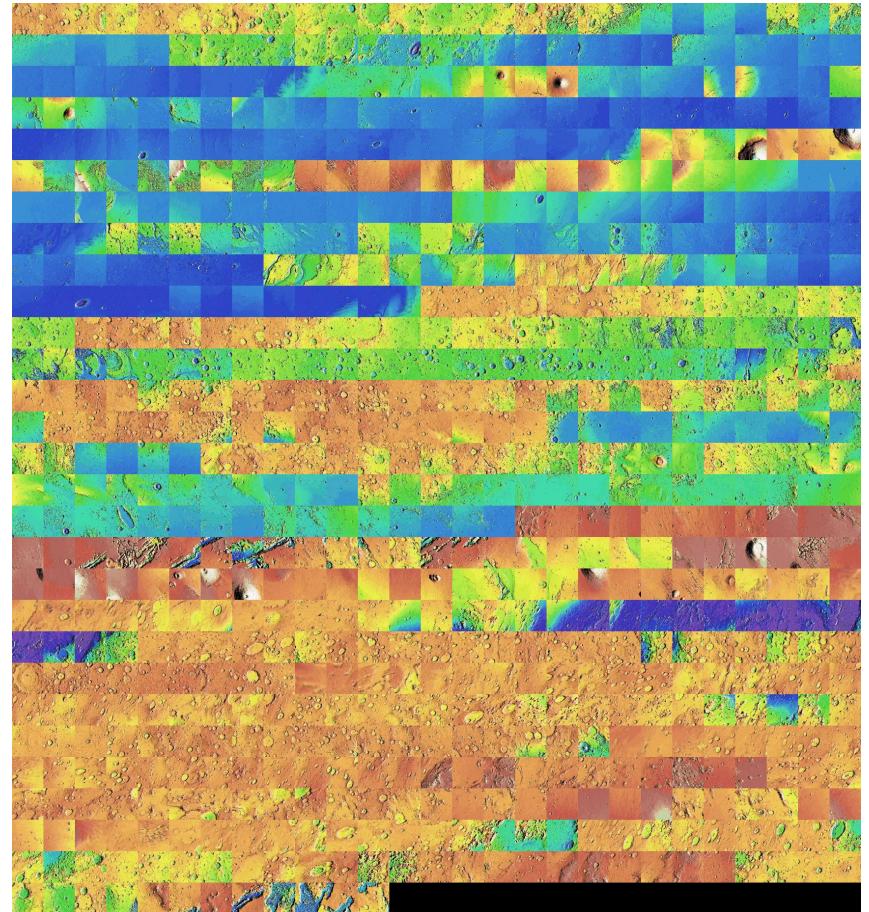
[2022-02-21 20:18:08,916] mapproxy.config - INFO - reading:
/home/even/spatialys/cnes/2021/hips/mapproxy/hips_examples/ogc_as_hips/mapproxy.yaml
[2022-02-21 20:18:08,966] mapproxy.hips - INFO - Generating 27x29 tiles of size 64x64
[2022-02-21 20:18:13,701] mapproxy.source.request - INFO - GET
https://planetarymaps.usgs.gov/cgi-bin/mapserv?map=/maps/mars/mars_simp_cyl.map&&layers=MOLA_color&bbox=-163.12500
000000003,19.47122063449069,-151.87500000000003,29.99999999999993&width=98&height=98&srs=EPSG%3A4326&format=image%2Fpng&request=GetMap&version=1.1.1&service=WMS&styles= 200 - 4691
[...]
[2022-02-21 20:19:43,969] mapproxy.source.request - INFO - GET
https://planetarymaps.usgs.gov/cgi-bin/mapserv?map=/maps/mars/mars_simp_cyl.map&&layers=MOLA_color&bbox=-50.625,-9.
594068226860458,-39.375,0.0&width=98&height=98&srs=EPSG%3A4326&format=image%2Fpng&request=GetMap&version=1.
1.1&service=WMS&styles= 200 - 791

real    1m36,890s
user    4m31,348s
sys     0m3,871s
```

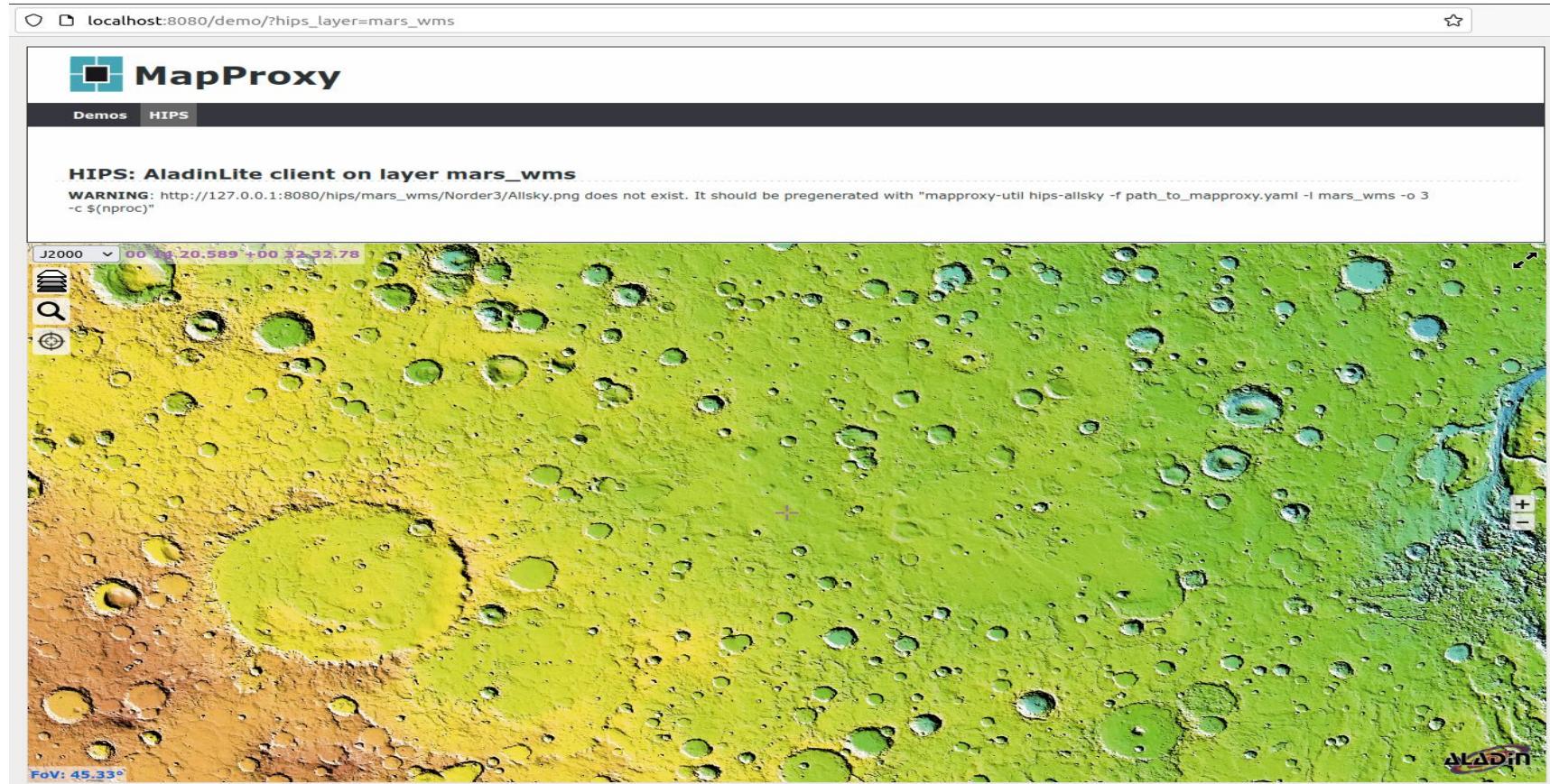
## Fichier Allsky

A l'ordre 3:

- Image de 1728x1856 pixels
- 768 tuiles HIPS de taille 64x64 pixels chacune



# Service démo : visualiseur AladinLite (après génération allsky)

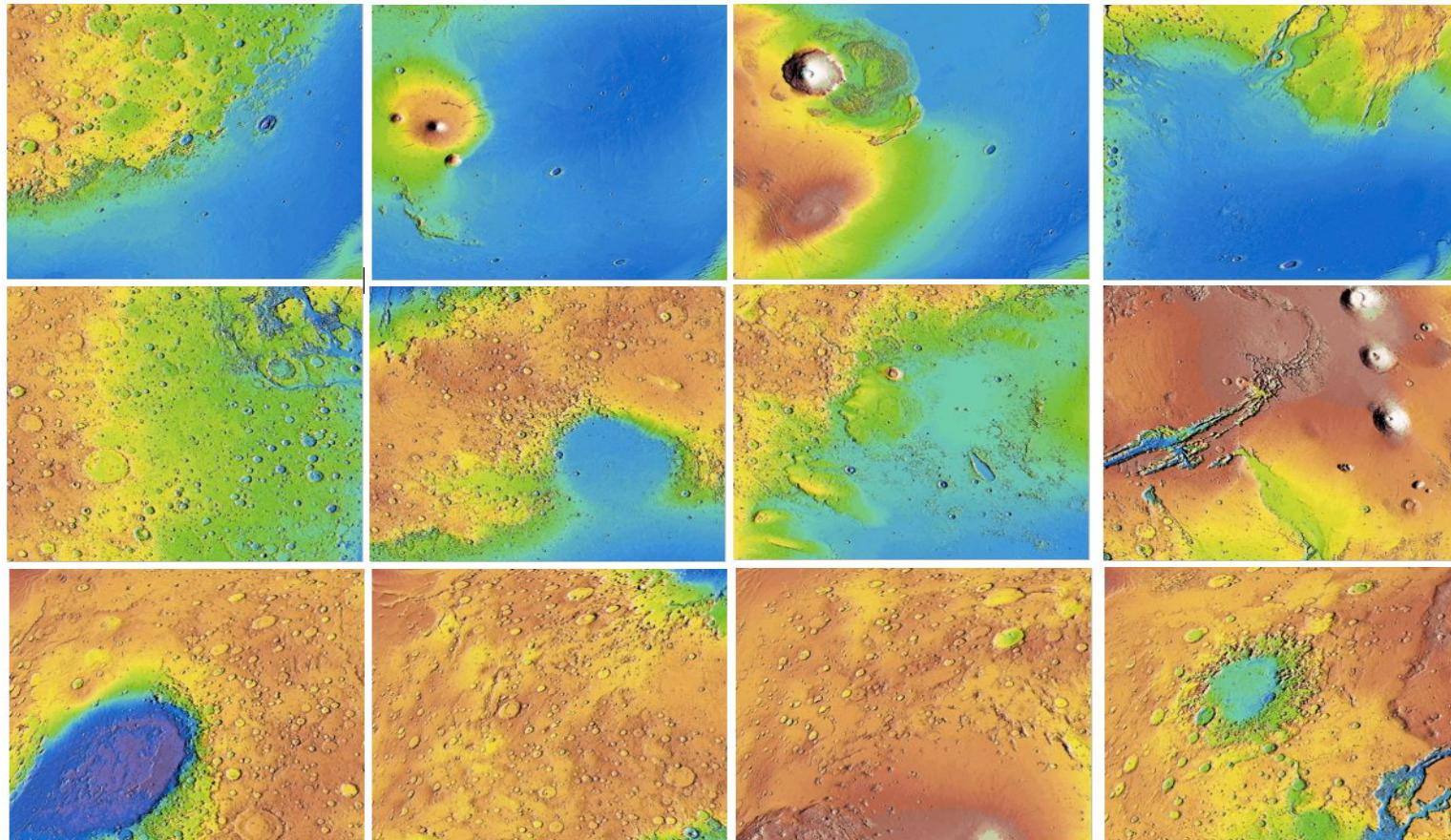


## Requête /properties avec passerelle

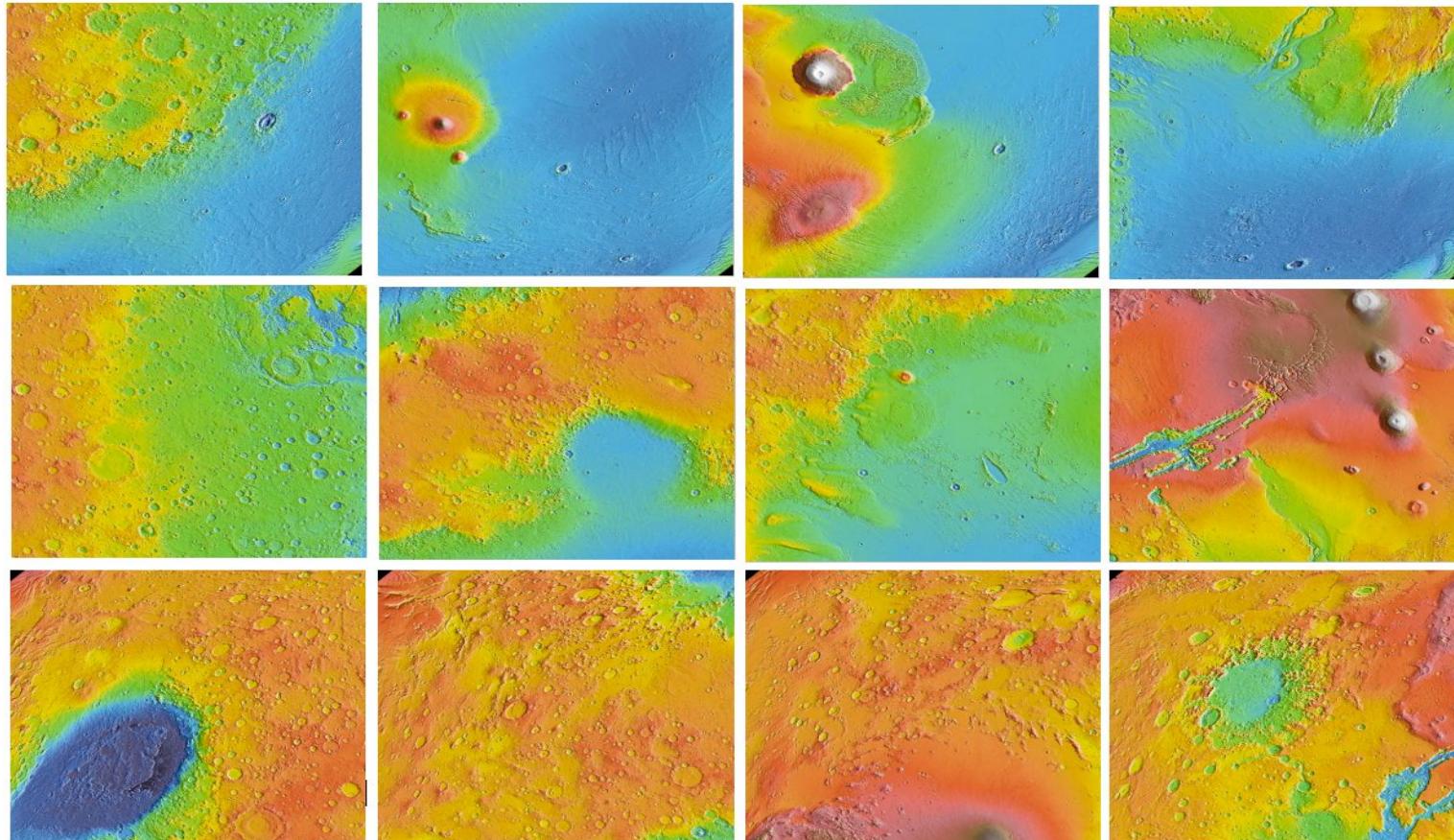
**http://localhost:8080/hips/mars\_wms/properties:**

```
creator_did=ivo://unknown.authority/mars_wms
obs_title=MARS MOLA color
dataproduct_type=image
hips_version=1.4
hips_release_date=2022-01-18T18:25:09Z
hips_status=public master clonableOnce
hips_tile_format=png jpeg
hips_order=5
hips_tile_width=512
hips_frame=planet
dataproduct_subtype=color
foo=bar
```

## Images ordre 0 de notre passerelle



## Image ordre 0 du service natif



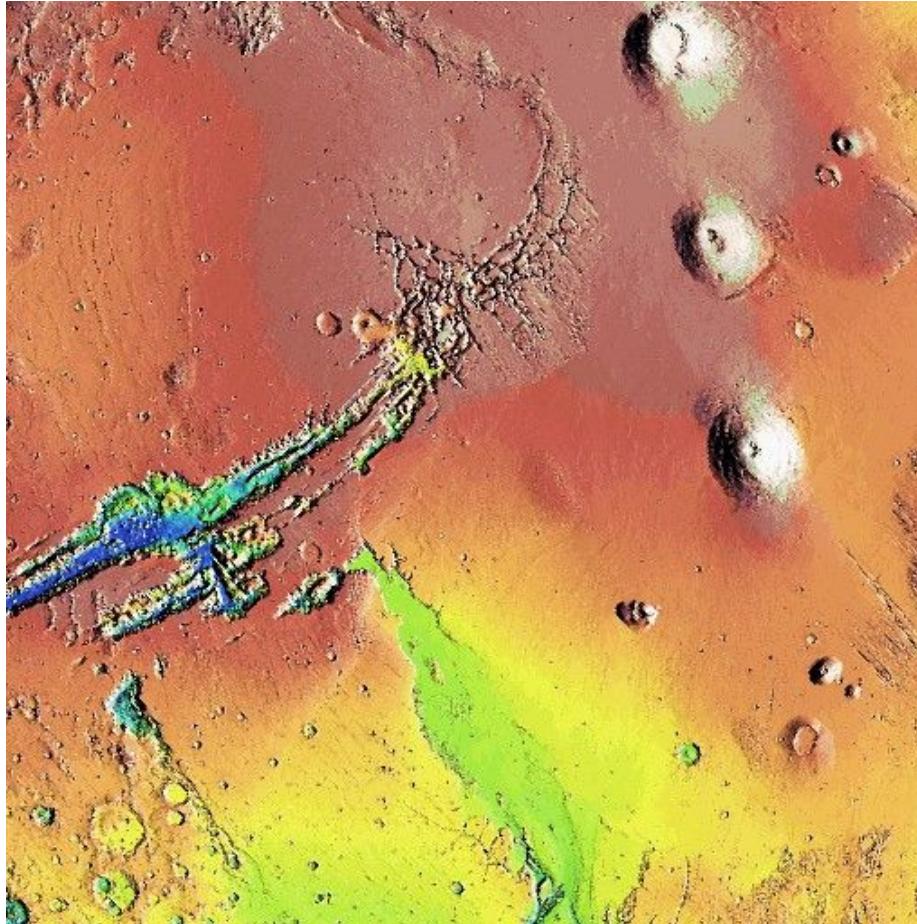
## Utilitaire de préénération des tuiles

```
$ mapproxy-util hips-seed -f path_to_mapproxy.yaml -l mars_wms -o 3 -c $(nproc)
```

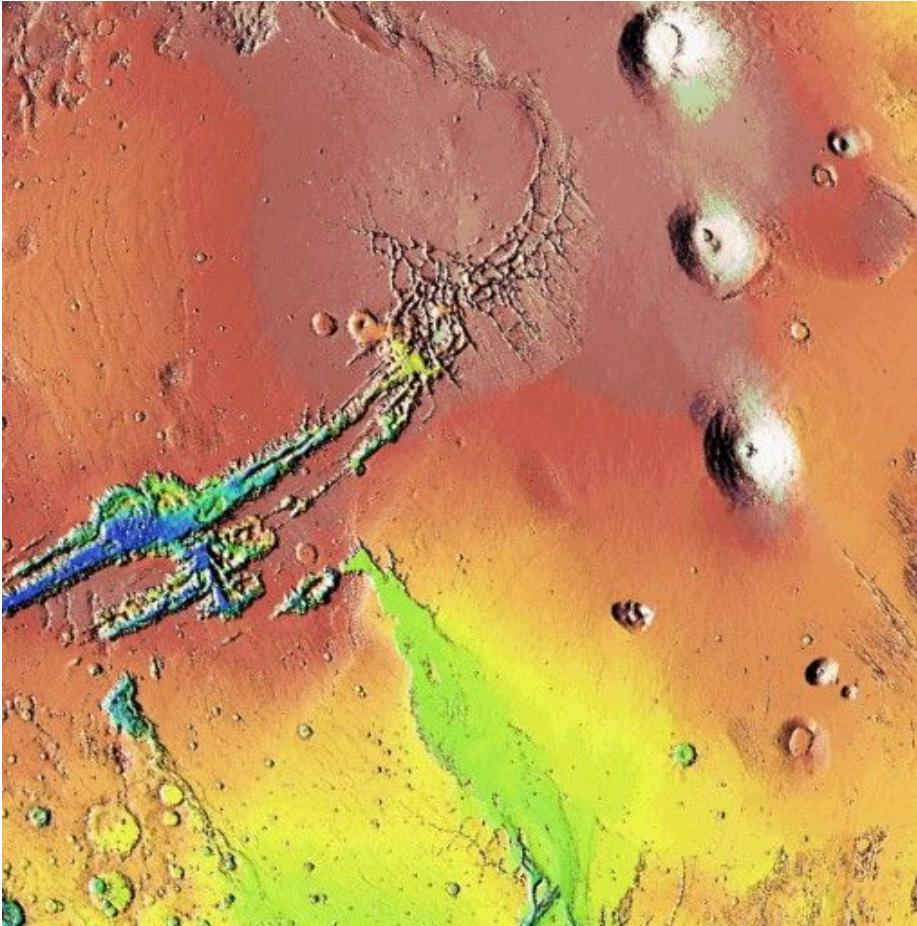
## Rééchantillonnage

- Portage des méthodes bilinéaires et bicubiques du logiciel GDAL/OGR
- Prise en compte du facteur d'échelle en X/Y
- Utilisation de numba: <https://numba.pydata.org/>
- Temps d'exécution pour une tuile:
  - Plus proche voisin : ~ 300 ms
  - Bilinéaire: ~ 510 ms
  - Bicubic: ~ 550 ms

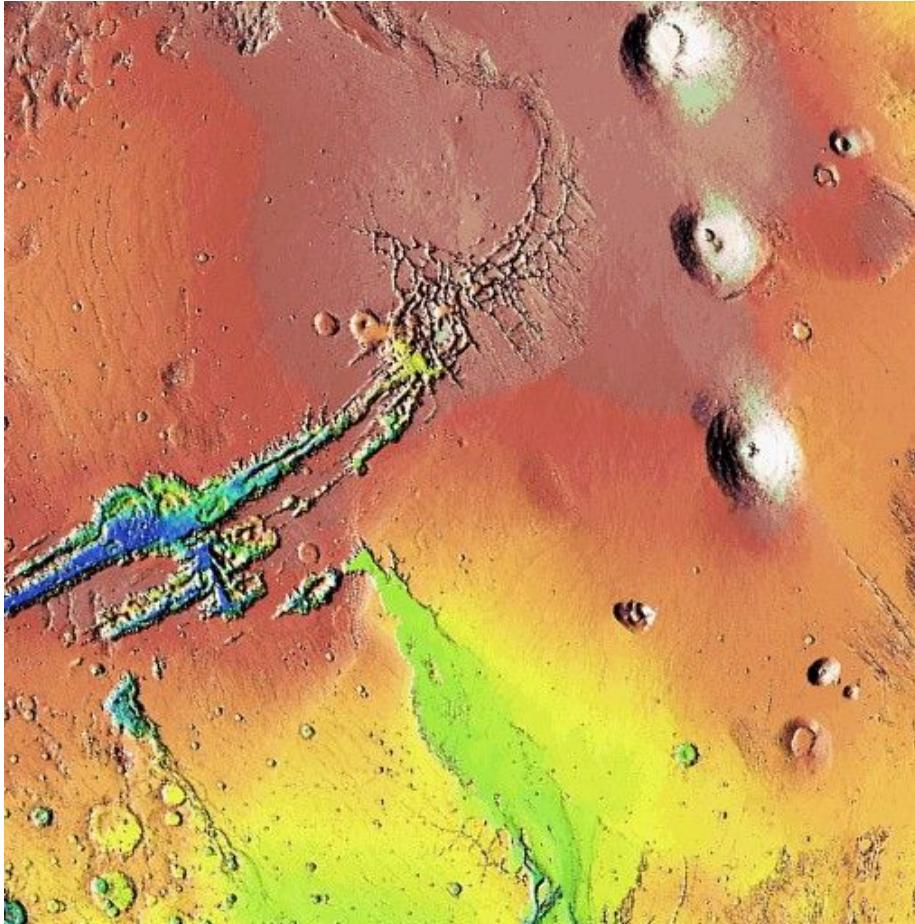
## Rééchantillonnage : plus proche voisin



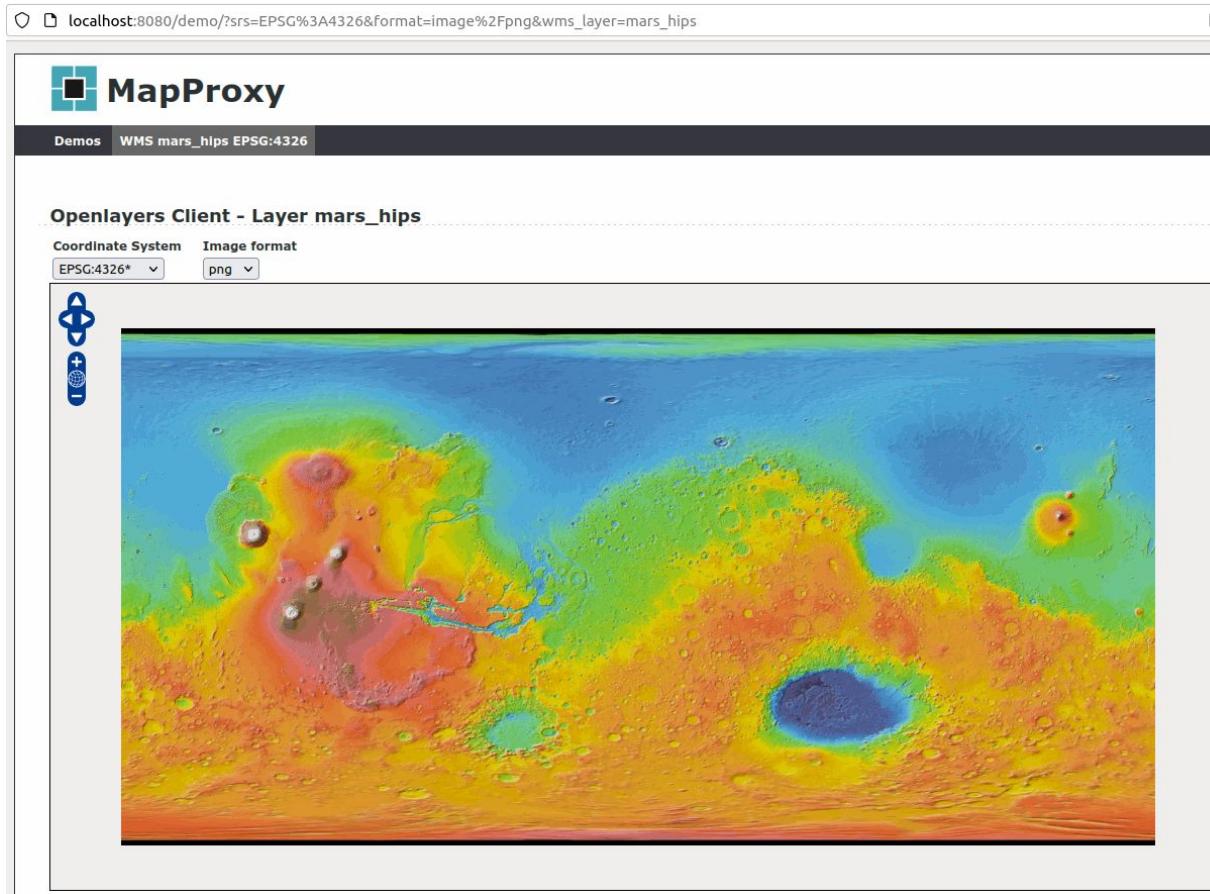
## Rééchantillonnage: bilinéaire

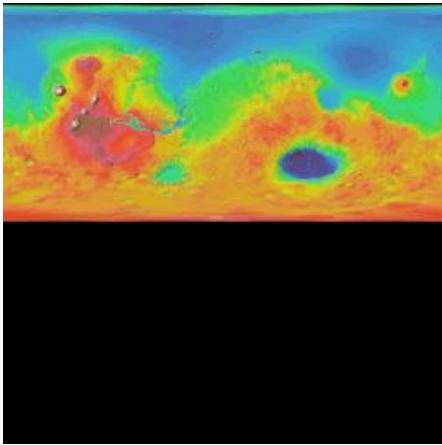


## Rééchantillonnage: bicubique

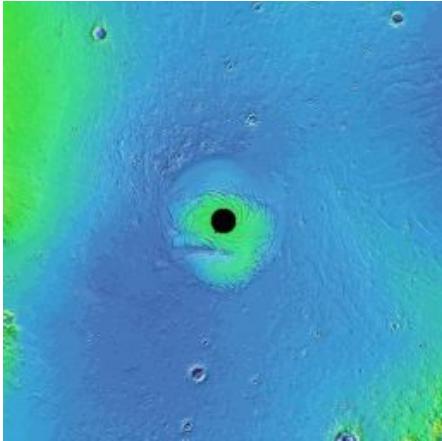


# MapProxy: Mise à disposition d'un service HIPS via WMS



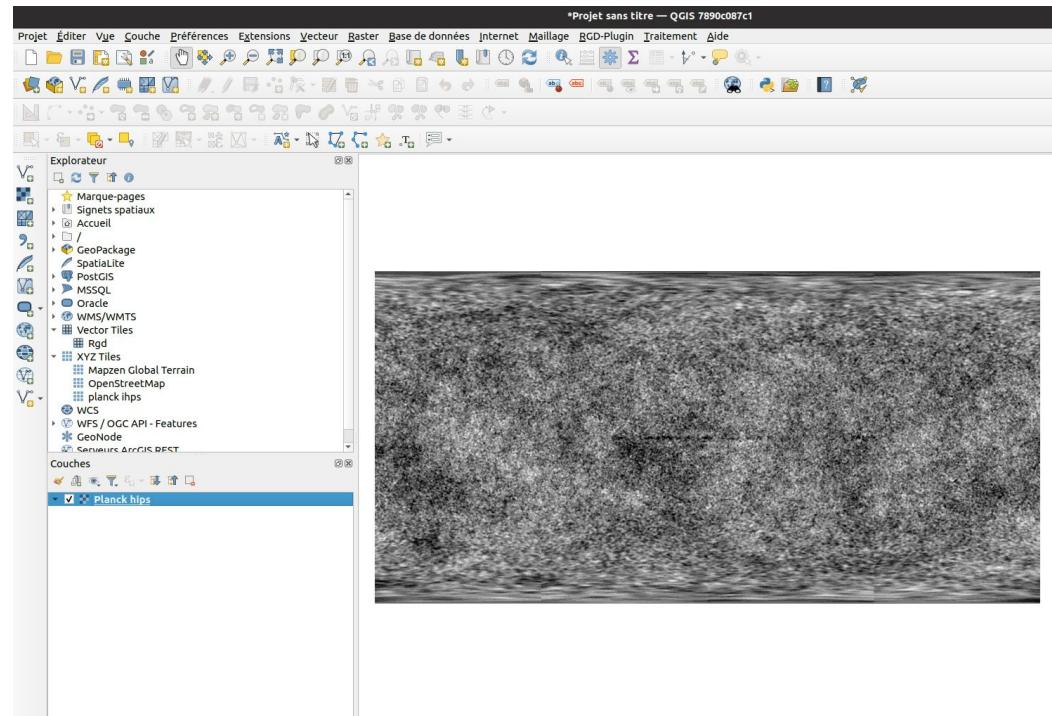
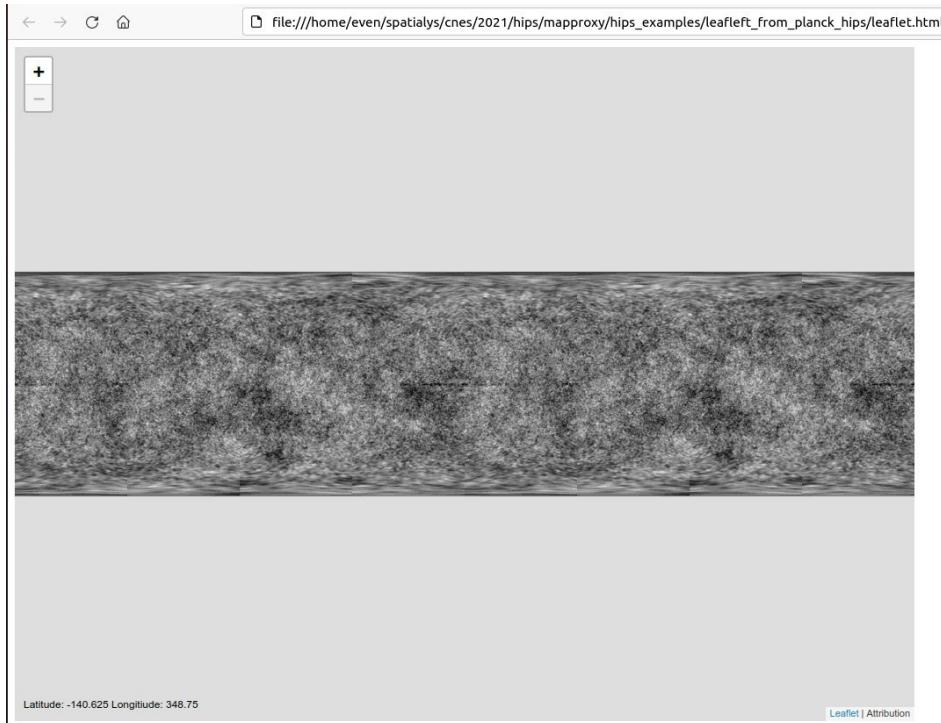


```
curl "http://localhost:8080/wmts/mars_hips/geodetic/0/0/0.png" > out.png
```



```
curl "http://localhost:8080/wmts/mars_hips_north_polar/north_polar/0/0/0.png" > out.png
```

# Visualisation des données astro avec leaflet / QGIS



- <http://voparis-vespa-hips-wms.obspm.fr:8080/demo/#>
- [https://docs.google.com/document/d/1fJJYVNYg2uWhCaN3fxIBLq09vFND0YL\\_LUiGRJ\\_fCh5c/edit?usp=sharing](https://docs.google.com/document/d/1fJJYVNYg2uWhCaN3fxIBLq09vFND0YL_LUiGRJ_fCh5c/edit?usp=sharing)
- <https://hub.docker.com/repository/docker/pdssp/maproxy>

## Développements futurs (prévus pour la fin de l'année)

- En contact avec la communauté mapproxy pour pérenniser les développements
- Ajout d'un système de plugins dans mapproxy
- Transformation des développements réalisés en tant que plugin
- CRS IAU:
  - Mapproxy s'appuie sur Pyproj (<https://pyproj4.github.io/pyproj/stable/>) dont les versions récentes gèrent les CRS IAU intégrés à PROJ 8.2
  - Cependant des tests montrent qu'il y a des présuppositions dans MapProxy qu'un CRS XXXX:CODE est EPSG:CODE
  - ⇒ développements nécessaires