PLANMAP
DATA & INTERFACES
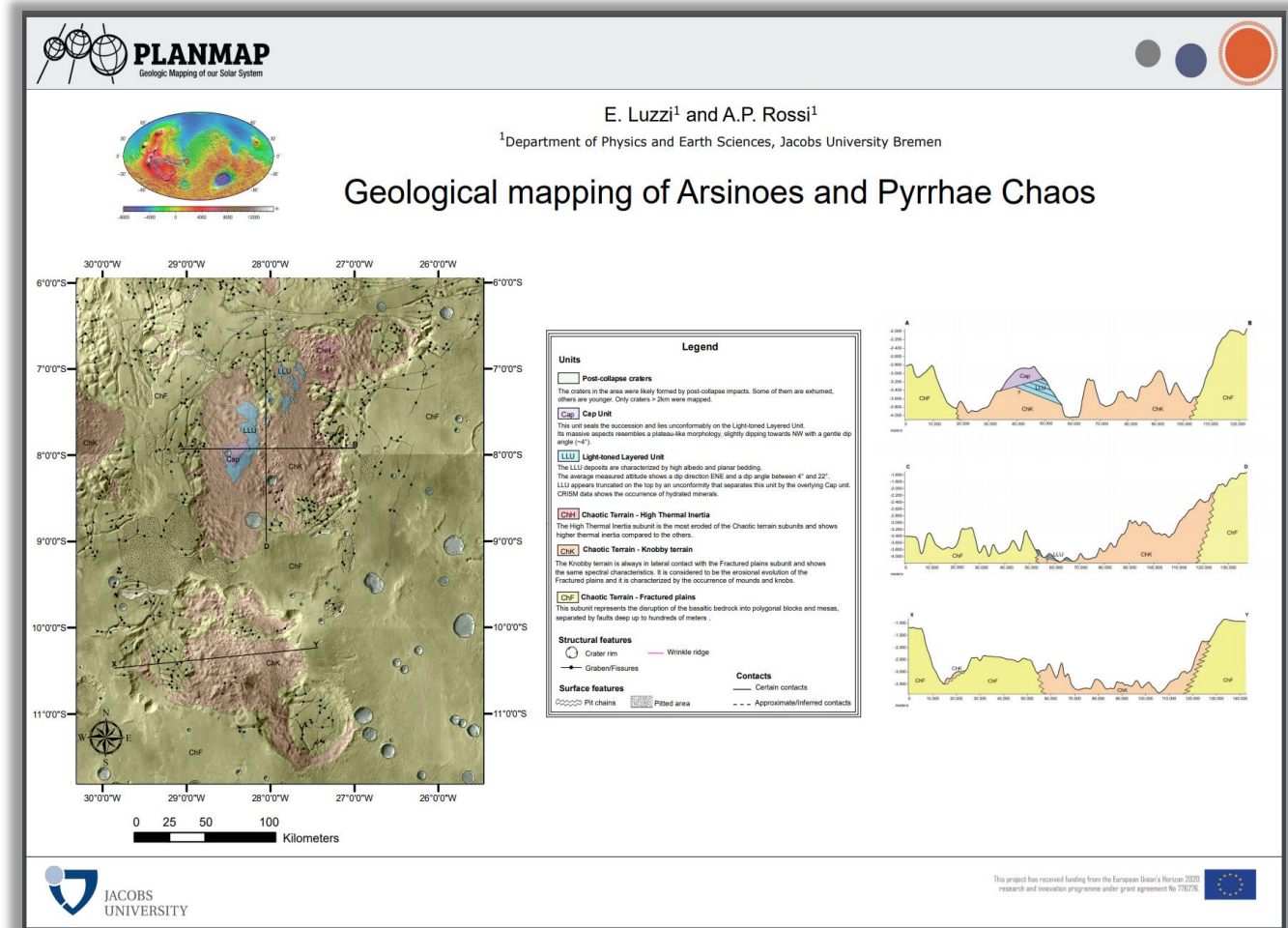## (LESSONS LEARNED TOWARDS FAIR)

Carlos H Brandt
and the PLANMAP consortium

c.brandt@jacobs-university.de

PLANetary **MAP**ping
compose into geological maps data about composition and structure of key planetary bodies: **Mars**, **Mercury**, the **Moon**.

The *maps* being published are provided in individual packages, according to the covered region and specific content:

Stratigraphic

Morphologic

Compositional

Geo-structural

Digital outcrop

E. Luzzi[1] and A.P. Rossi[1]

[1]Department of Physics and Earth Sciences, Jacobs University Bremen

Geological mapping of Arsinoes and Pyrrhae Chaos

Infrastruture

Geoserver – https://geoserver.planmap.eu

Wiki – https://wiki.planmap.eu

Webservers

Homepage – https://www.planmap.eu

Maps-app – https://maps.planmap.eu

Storymaps – https://stories.planmap.eu

Data archive – https://data.planmap.eu

- Nextcloud (internal use)

- Social networks
  - Facebook: facebook.com/planmap.eu
  - Twitter: twitter.com/planmap_eu

FAIR

Findable   Accessible   Interoperable   Reusable

○ Basically, we want our data to be used as much as possible, efficiently as possible, to optimise its use.

○ How do we give it visibility so that people actually find it?

  ○ What are the tools and channels to communicate to potential users?

○ Which interface(s) to provide for its access, exploration and download

○ And then on using it, decisions on metadata and file formats are as well important for its (better) use.

On access and use

- Graphical interactive map interface
- Standard services/file formats
  - GeoPackage
  - GeoTiff
  - OGC WFS/WMS
- Package naming convention
- Data archive structure
- Vector data/metadata convention

Package naming

- Each data package is named uniquely considering type of data provided, planetary body, region covered and a free-form substring to further specify the content:

  PM_<body>_<type>_<toponym>_<note> :

  - <body> is a tree-letters string representing the body of interest: MOO MER MAR ;
  - <type> represents the data content type: S M C D G I ;
    - More than one type can be provided per package
  - <toponym> stands for the name of the region covered, e.g, **Hokusai** or **H05**
  - <note> – *optional* – for a further refinement of the package name when two or more packages would otherwise conflict name:
    - PM-MER-MS-H05_3cc_01
   - PM-MER-MS-H05_5cc_01

Data archive

- Each package is composed by a set of documents and data files distributed in directories accordingly:

- Document – provides the map itself as PDF and optionally other complementary files

- Raster – provides basemaps or compositional arrays

- Vector – provides features table(s)

- 3dmodels – provides structural models

- README.md -- provides map data description information about data sources, publication projection used and other relevant metadata

```
PM-MAR-MS-Arsinoes_02
├── README.md
├── document
│   └── PM-MAR-MS-Arsinoes_02.pdf
├── raster
│   └── PM-MAR-MS-Arsinoes_02.tiff
└── vector
    └── PM-MAR-MS-Arsinoes_02.gpkg
```
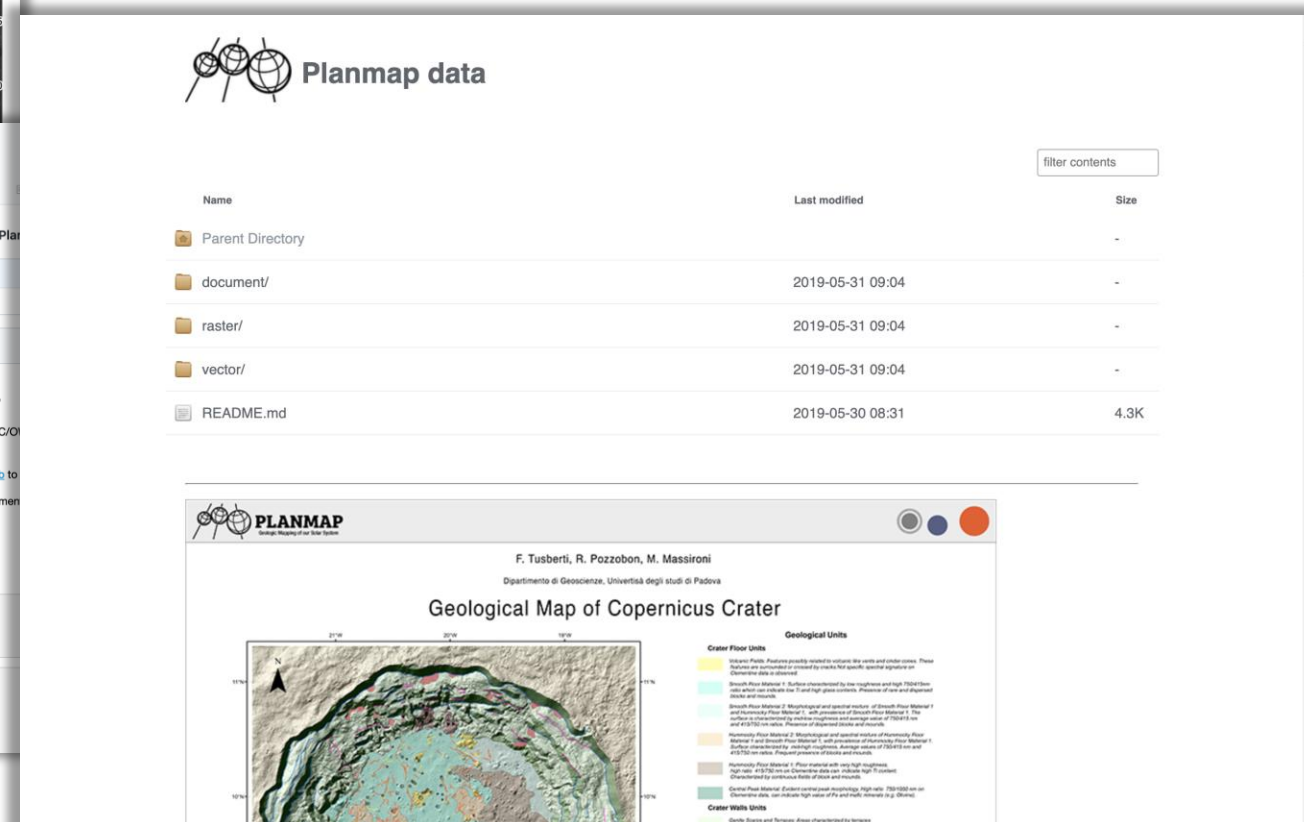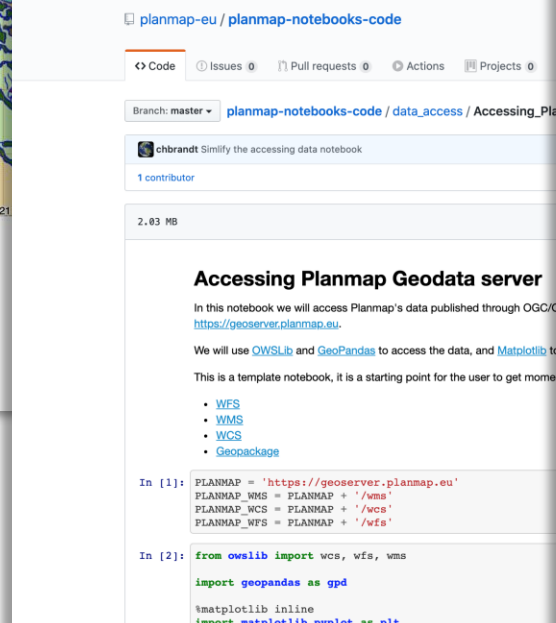
Vector data/metadata

- Vector data is stored in Geopackage files
  - Geopackage is an open format for storing different types of (geospatial) data in an SQLite data store.
  - Choice for GeoPackage instead of Shapefiles:
    - Open format
    - One file to store them all
- Towards a more homogeneous structure tables, standards for tables and columns
  - Geological Units
    - Name, Code, Geometry (Polygon)
  - Geological Contacts
    - Type, Geometry (Linestring)
  - Surface Features
    - Type, Geometry (Polygon)
  - Linear Features
    - Type, Geometry (Linestring
  - Layer Styles
    - StyleQML (from QGIS)
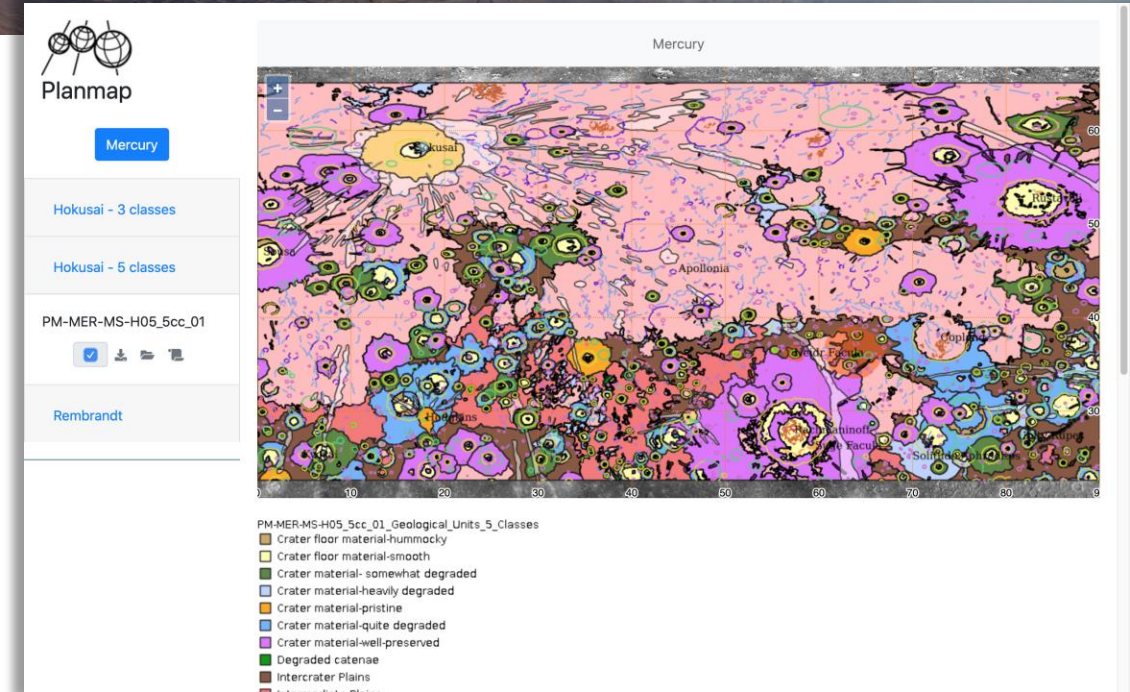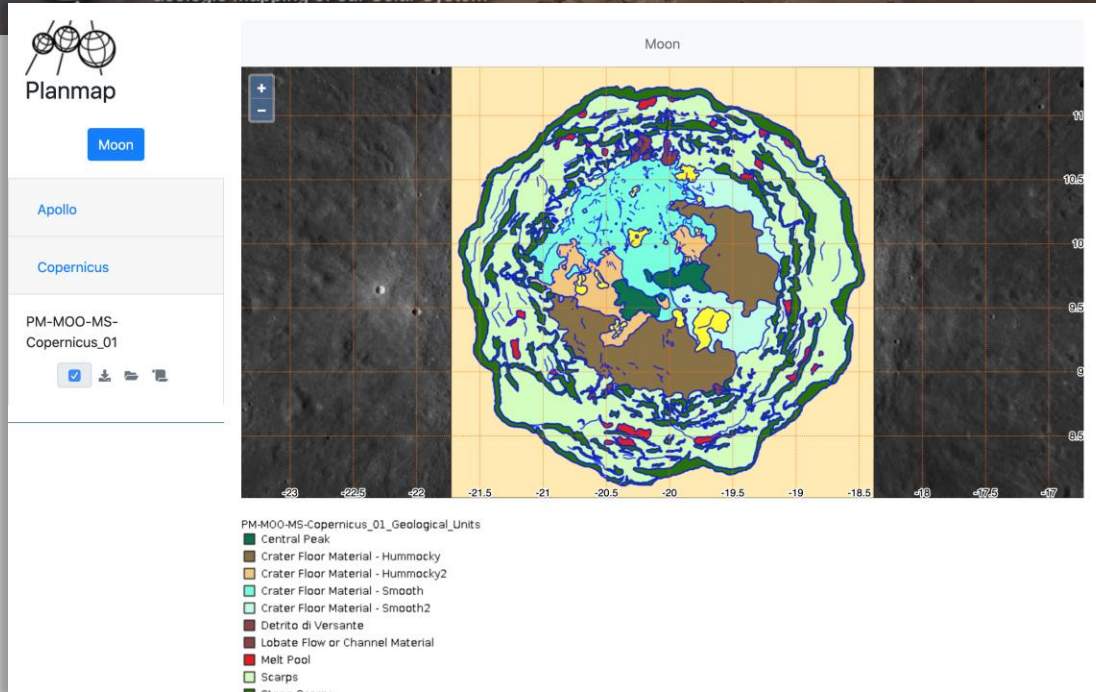
MAPS-APP

- Interactive graphical web interface to explore the maps on each planet
- Links to
  - Data archive (@ data.planmap.eu
  - Jupyter notebooks (@ github.com/planmap-eu)

MAPS-APP

- Links to
  - Data archive (@ data.planmap.eu
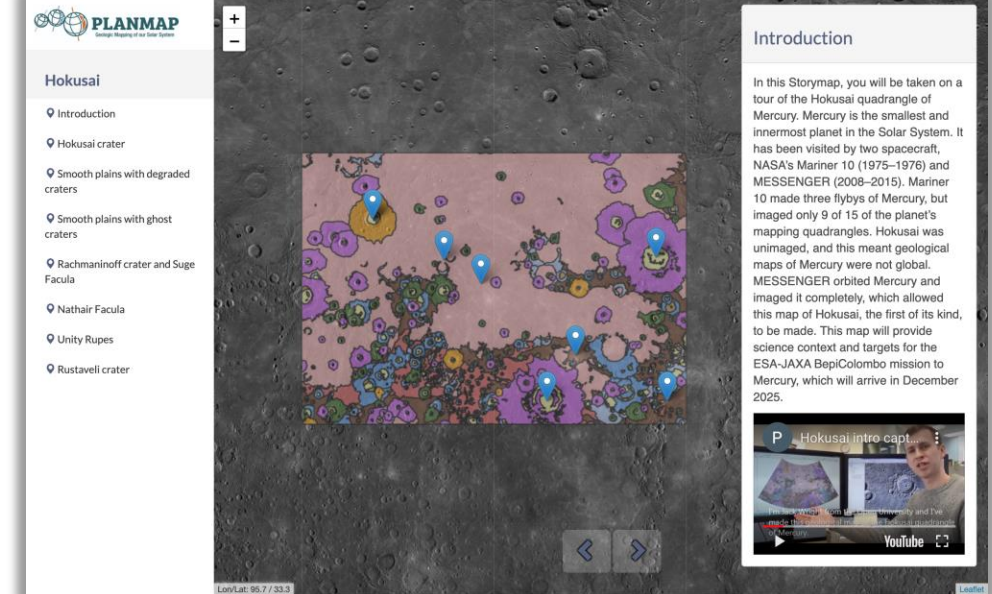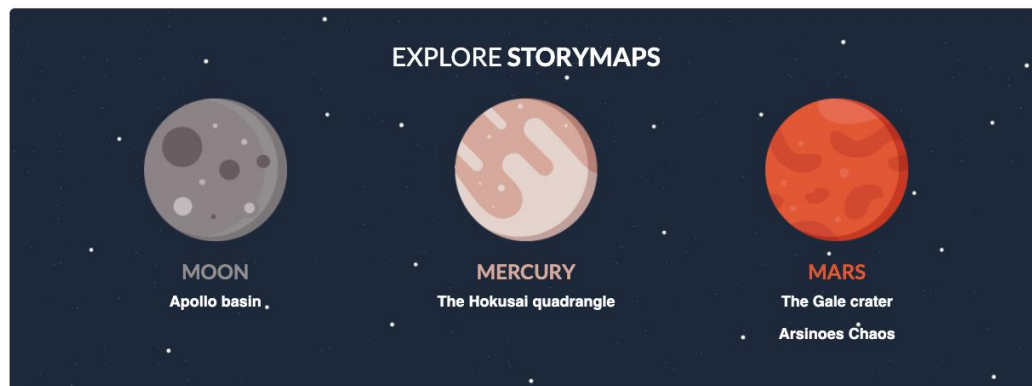 - Jupyter notebooks (@ github.com/planmap-eu)

Storymaps

Present high-level explanations of the maps or narratives about the data and the science behind it for non-experts

Data publication pipeline

- Data producers put data in our cloud (Nextcloud)
- Data package is verified, moves to a "release" area
- Data package is uploaded to
  - data.planmap.eu/pub
  - data.planmap.eu/zip
- Raster and Vector data are ingested in geoserver.planmap.eu
- Maps-app is updated to query new map layers

- Update of story-maps follows a markdown template using Github issues
- Then it is formatted in a json document for publishing on stories.planmap.eu

Lessons learned / todos

- Automation and a better data sharing
  - Cloud * Validation --> Data | GeoServer + REST ingestion
  - Maps <-- GeoServer get-capabilities / stores | layers

- Package data/metadata validation: a summary viewer as from the standards
  - Data producers to verify their content as seeing by the system (and the users).

- Storymaps to provide an interactive interface for writing & upload stories

(Other) Resources

Maps-app source code & backend

https://github.com/planmap-eu/planmap-app-client

https://github.com/planmap-eu/planmap-app-server

Stories source code

https://github.com/planmap-eu/storymaps

Docker geoserver

https://github.com/chbrandt/docker-geoserver

Apaxy PLANMAP

https://github.com/chbrandt/apaxy/tree/planmap

www.planmap.eu

c.brandt@jacobs-university.de