

M³G: Towards to the interoperable GNSS station metadata catalogue

EUREF Symposium – May 26, 2021

A. Fabian, C. Bruyninx, A. Miglio, J. Legrand

Introduction

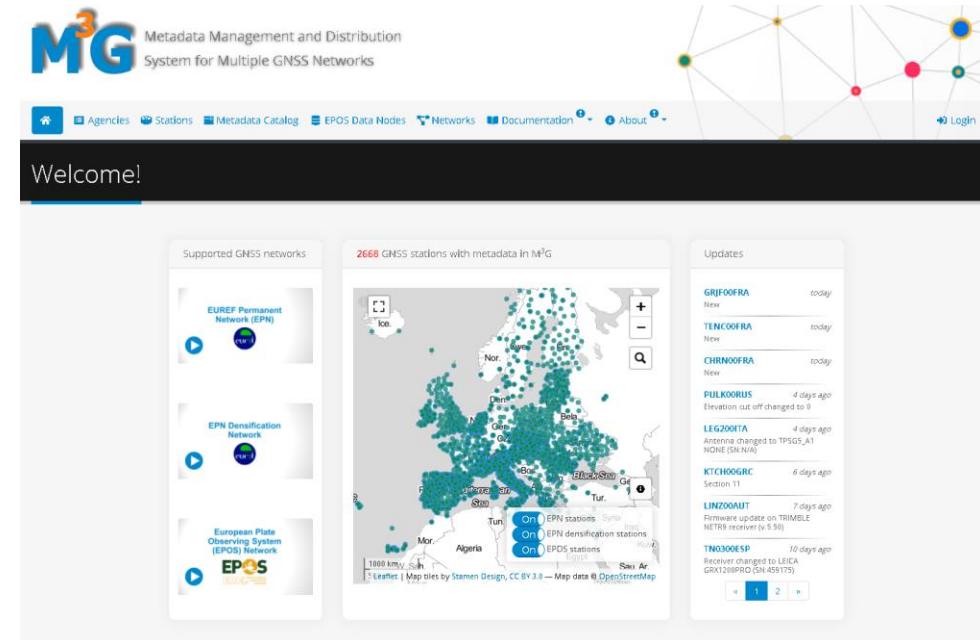
Metadata Management and Distribution System for Multiple GNSS Networks (M³G)

Collect GNSS station metadata

- IGS Site log
- GeodesyML
- Networks information with DOI
- Station pictures
- Data license
- Embargo period

Additional requirements

- Comply with the EU GDPR requirements
- Centralised contact information for each Agency
- Towards to Linked Open Data (LOD)
- Towards to Findable, Accessible, Interoperable, Reusable (FAIR) data



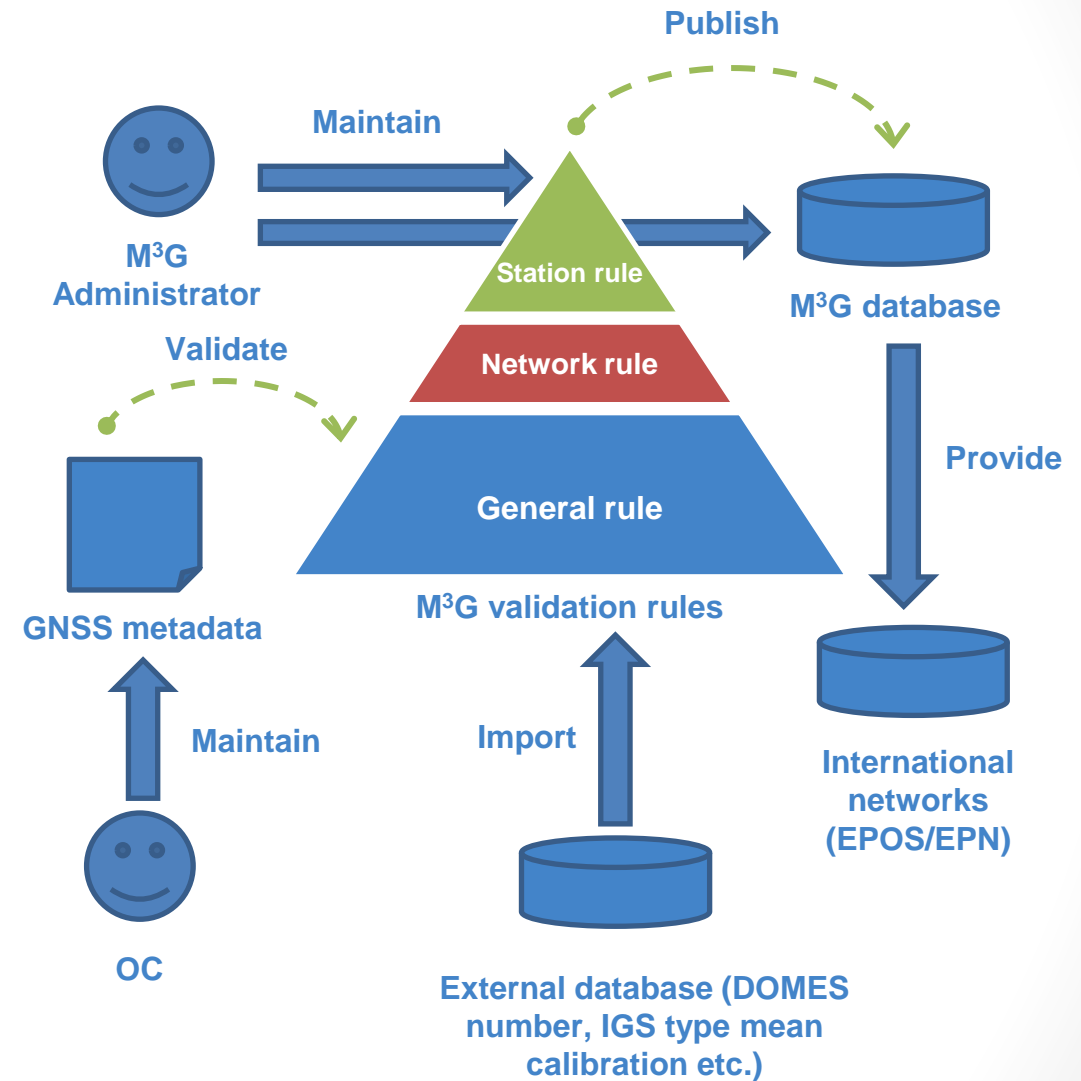
M³G Web portal (<https://gnss-metadata.eu>)

Reliable metadata

1. Operation Centre (OC) is responsible to
 - maintain the station metadata in M³G.

2. M³G system is responsible to
 - import and extend the validation rules (e.g. DOMES number database, Official country names, external networks (IGS, EGVAP)), IGS type-mean antenna calibration),
 - validate the provided information using three sets of validation rules: general rules, network-dependent rules, station-dependent rules,
 - provide the updated valid GNSS station metadata to EPN and EPOS.

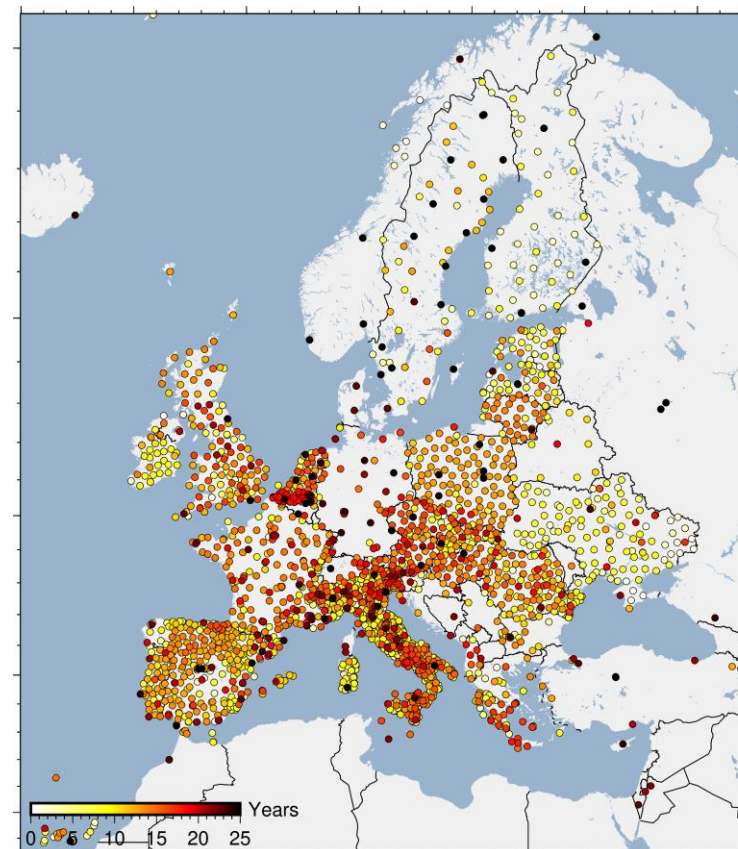
3. M³G Administrator is responsible to
 - modify the station dependent rules,
 - maintain the network dependent rules,
 - maintain the M³G database,
 - check if the station metadata is up-to-date,
 - contact the agency in case of any issue.



Stations in M³G

Stations in M ³ G				
Network	Number of stations in M ³ G	Required to use M ³ G	Station Metadata	Registered data license
EPN	435	✓	Mandatory	72%
EPOS	2026	✓	Mandatory	100%
EPN Densification	2026/3848	✓	Recommended	65 %
E-GVAP	1554/5960	-	Optional	-
IGS	118/506	-	Mandatory	-

Stations in M³G



20 years of GNSS station history in M³G
(2687 stations in M3G)

Interoperable service

Provide updated metadata to

- International network (EPN/EPOS)
- Update the station metadata on local FTP (<ftp://gnss-metadata.eu>)

M³G provides a wide range of services to download metadata:

- FTP (<ftp://gnss-metadata.eu>)
- Metadata Catalog (<https://gnss-metadata.eu/site/metadata>)
- REST API (<https://gnss-metadata.eu/site/api-docs>)

Metadata Catalog

Filter the stations for which you want to download metadata and/or create a new metadata catalog

Sitelog GeodesyML {all 4/9-char station ids} Belgium (BEL) EPN Reset Submit

Showing 1-6 of 6 items. ▲

#	SELECT ALL/NONE	STATION	FILENAME	MD5
0	<input type="checkbox"/>	WARE00BEL	ware00bel_20210217.log	43197e4e13106fb31df2e163dc952ec8
1	<input type="checkbox"/>	REDU00BEL	redu00bel_20210119.log	8247f1895e8fff4ad65aa961620ff90d
2	<input type="checkbox"/>	DOUR00BEL	dour00bel_20210201.log	6c664ac12cb70cec1a5581a87f7ca1ea
3	<input type="checkbox"/>	DENT00BEL	dent00bel_20210121.log	5b97865d2d4e98529b391d2b0e3f85f9
4	<input type="checkbox"/>	BRUX00BEL	brux00bel_20210420.log	fa86d5425ecce02e4cdccb39b1c14d78
5	<input type="checkbox"/>	BRUS00BEL	brus00bel_20200604.log	3552475920a2c5685816fa1c17e48d3f

[View metadata catalog above in text format](#)
[Download selected sitelog/GeodesyML](#)

Metadata Catalog

Sitelog GeodesyML {all 4/9-char station ids} Belgium (BEL) EPN Reset Submit

View metadata catalog above in text format Download selected sitelog/GeodesyML

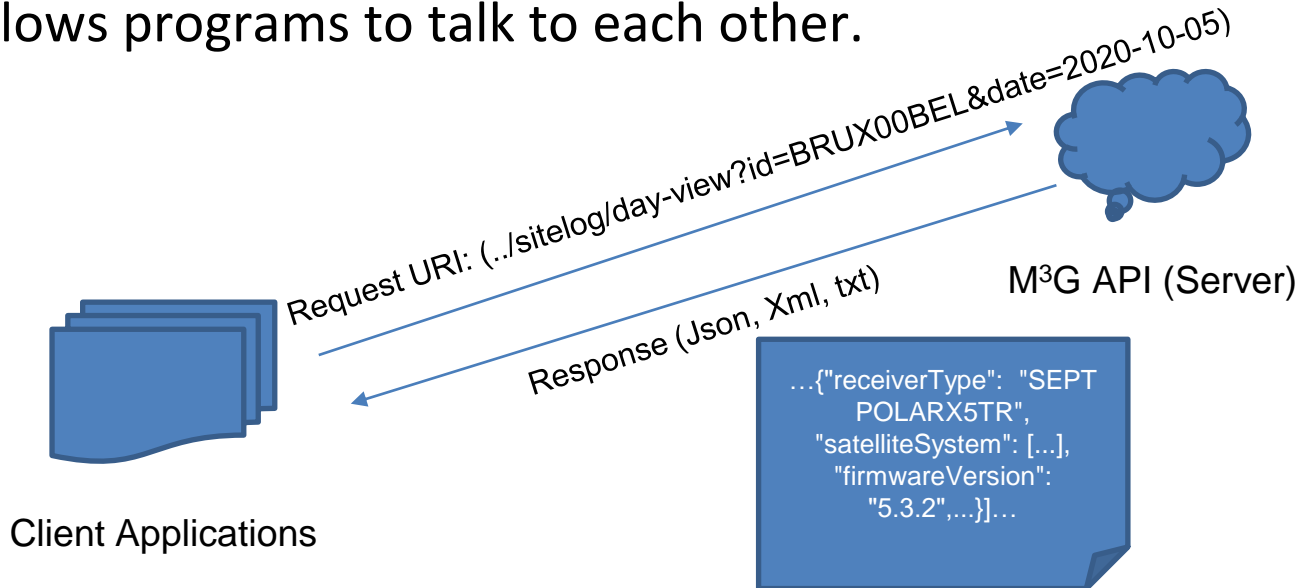
<https://gnss-metadata.eu/v1/sitelog/metadata-list?downloadFormat=log&validMetadata=1&network=EPN&country=BEL&stationId=>

id	md5_sitelog	name_sitelog	update(system-time)	url_sitelog
WARE00BEL	43197e4e13106fb31df2e163dc952ec8	ware00bel_20210217.log	2021-02-17T14:48Z	../v1/sitelog/exportlog?id=WARE00BEL
REDU00BEL	8247f1895e8fff4ad65aa961620ff90d	redu00bel_20210119.log	2021-01-19T15:54Z	../v1/sitelog/exportlog?id=REDU00BEL
.				
.				

REST API

Representational State Transfer (REST) Application Interface (API)

- The API is a set of rules that allows programs to talk to each other.



You can send a request with any programming language.

- Javascript (Fetch `$.ajax()`)
- Perl (`HTTP::Request`)
- Python (`Requests`)
- Command line tool (`cURL`)

API use cases - Request

A. Get antenna/receiver configuration for a specific date and for a specific network

Request:

```
> curl -X GET "https://gnss-metadata.eu/v1/sitelog/day?network=EPN&date=2021-05-19" -H "accept: application/json"
```

Response:

```
...{"receiverType": "SEPT  
POLARX5TR",  
"satelliteSystem": [...],  
"firmwareVersion":  
"5.3.2",...}]...
```

B. List all site logs updated since specific (system) date for a specific network

Request:

```
> curl -X GET "https://gnss-metadata.eu/v1/sitelog/metadata-list?dateUpdate=2021-05-01T00%3A00Z&network=EPOS" -H "accept: text/plain"
```

Response:

id	name_sitelog	url_sitelog
WTZR00DEU	wtZR00deu_20210518.log	https://gnss-metadata.eu/v1/sitelog/exportlog?id=WTZR00DEU
WRLG00DEU	wrlg00deu_20210518.log	https://gnss-metadata.eu/v1/sitelog/exportlog?id=WRLG00DEU
...		

API use cases - Request

C. List of all changes in chronological orders for a specific network/station

Request:

```
> curl -X GET "https://gnss-metadata.eu/v1/sitelog/change?network=EPN" -H "accept:
  "application/json"
```

Response:

```
[
  {
    "id": "ZYWI00POL",...
    "changes": [
      { "status": "installed", "dateFrom": "2003-01-30T00:00Z", "note": "Installed", ... },
      { "status": "firmware_updated", "dateFrom": "2005-10-12T00:00Z", "note": "Firmware changed to CK00", ..}
      { "status": "station_inactive", "dateFrom": "2007-11-25T21:00Z", "note": "inactive" ...},
      { "status": "receiver_changed", "dateFrom": "2007-11-27T00:00Z", "note": "Receiver changed to TRIMBLE NETR5 (SN:4716K05678)" ... }...
```

API use cases - Update

D. Change the firmware for a specific station

Request:

```
> curl -X PUT "https://gnss-metadata.eu/v1/sitelog/firmware-change?id=BRUX00BEL" -H "accept: application/json" -H "Authorization: Bearer <agency token>" -H "Content-Type: application/json" -d "{\"updateMadeBy\":\"Carine Bruyninx\",\"endOfTheLastSection\":\"2020-08-15T13:56Z\",\"changedTo\":\"5.3.2\",\"startOfTheNewSection\":\"2020-08-31T13:56Z\"}"
```

Response:

```
{
  "id": "BRUX00BEL",
  "uri": "https://xxxxx...",
  "preparedDate": "2021-05-19",
  "dateUpdate": "2021-05-19T16:38:09.648Z",
}
```

API use cases - Update

E. Upload site log from the local disk

Request:

```
> curl -X PUT "https://gnss-metadata.eu/v1/sitelog/upload-sitelog?id=AILT00FRA" -H "accept: application/json" -H "Authorization: Bearer <agency token>" -H "Content-Type: text/plain; charset=utf-8" --data-binary "@ailt_20210415.log"
```

Response:

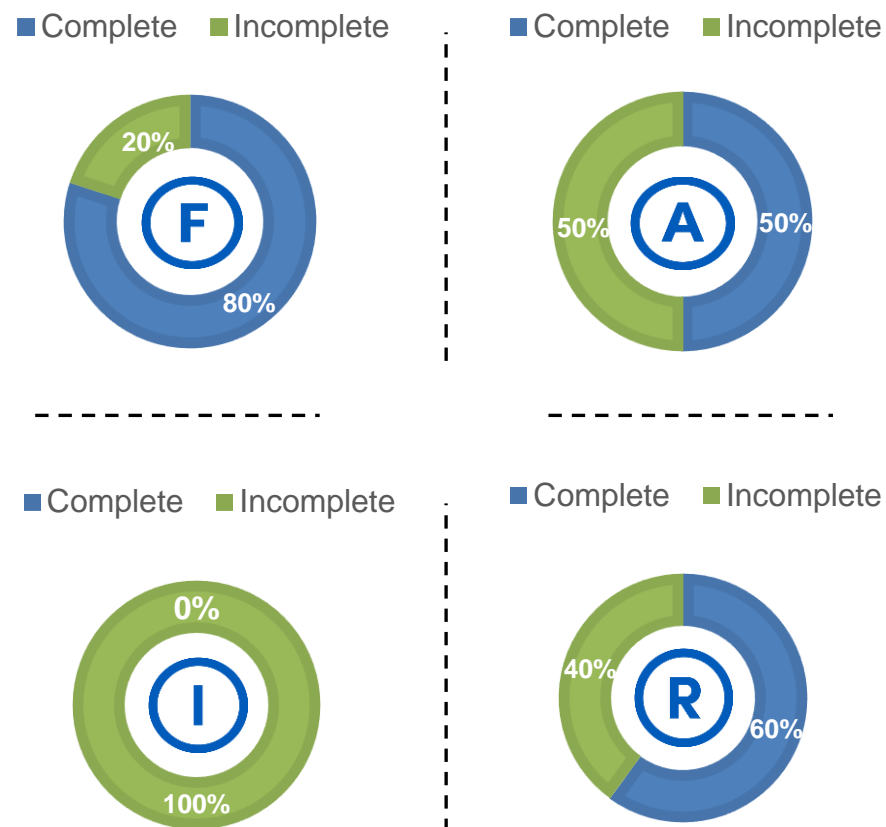
```
{  
  "id": "AILT00FRA",  
  "uri": "https://xxxx",  
  "preparedDate": "2021-05-19",  
  "dateUpdate": "2021-05-19T16:38:09.648Z",  
}
```

FAIR

One of the Aim in M3G is to turn GNSS metadata to FAIR (Findable, Accessible, Interoperable and Reusable) Data Object.

There are several more requirements that need to be implemented:

- PID (Persistent identification of the GNSS metadata) for the resources.
- Metadata description schema to ensure the interoperability of metadata.



Findability, Accessibility, Interoperability, and Reuse metrics score in M³G according to the FAIRsFAIR Data Object Assessment Metrics (<https://github.com/FAIRsFAIR/DataObjectMetrics>)

Thank you

- Contact:
 - Web: <https://gnss-metadata.eu>
 - Email: m3g@oma.be
- M³G documentation:
 - <https://gnss-metadata.eu/site/quickstart>
 - <https://gnss-metadata.eu/site/faq>
- M³G API documentation/example:
 - <https://176paws.github.io/doc4m3g/>
 - <https://gnss-metadata.eu/site/api-docs>