

P R Ø J

PROJ

Une librairie ouverte de transformation de coordonnées

Even Rouault
SPATIALYS


PROJ en résumé

- Bibliothèque écrite en C/C++ et utilisable depuis ces langages, et autres via bindings
- Licence libre MIT
- Gestion des:
 - Projections cartographiques
 - Conversions: unités, géographiques/géocentriques
 - Transformations: grilles, Helmert, Molodensky, modèles de déformation, modèles triangulés, etc.
- Outils en ligne de commande
- Intégration de multiples registres CRS et transformations: EPSG, IGN, ESRI, ...
- Grilles de transformation

Historique: les débuts

- PROJ 1: 197?-1983
 - Auteur: Gerald E. Evenden (USGS)
 - Ecrit en Ratfor/Fortran
 - Basé sur GCTP (USGS)
- PROJ 2.0 à 4.3: 1983-1995
 - Ecrit en C
 - Domaine publique
 - > 100 projections (formules provenant très souvent de John Parr Snyder / USGS)
 - Transformations directes et indirectes
 - Chaîne PROJ4: “+proj=lcc +lat_0=46.5 +lon_0=3 +lat_1=44 +lat_2=49 +x_0=700000 +y_0=6600000 +ellps=GRS80”
 - Gestion des ellipsoïdes, mais pas des datums

Historique: ère PROJ4

- PROJ 4.4.0 à 4.8.0: 2000-2011
 - Auteur: Frank Warmerdam et communauté
 - License open source MIT
 - Ajout du concept de datum (WGS84, NAD27, NAD83)
 - Transformations de datum via pivot WGS 84:
 - horizontales: via +towgs84 / +nadgrids
 - Vecticales: +geoidgrids
 - Gestion de grilles de transformation (NTv2, GTX)
 - Inclusion du registre EPSG sous forme de fichier texte associant un code à une chaîne PROJ4
 - Premiers tests de non-régression
 - Ajout du registre IGNF en 2008 (Didier Richard. IGN)
 - 2008: sous-projet de MetaCRS, fondation  OSGeo

Historique: transition

- PROJ 4.9.x
 - Auteur: Howard Butler, E. Rouault et communauté: 2011-2017
 - Maintenance / mise à jour registre EPSG
 - Modernisation de la documentation (RST / Sphinx)
 - Système de construction CMake

Historique: pipelines géodésiques

- PROJ 5.0: 2017

- Auteurs: Kristian Evers, Thomas Knudsen (SDFE, Danemark)
- Nouvelle API C: proj.h
- Projet désormais appelé PROJ et non plus PROJ4
- Ajout du concept de pipeline de transformation géodésique

```
+proj=pipeline
```

```
+step +proj=axiswap +order=2,1
```

```
+step +proj=unitconvert +xy_in=deg +xy_out=rad
```

```
+step +proj=hgridshift +grids=fr_ign_ntf_r93.tif
```

```
+step +proj=lcc +lat_0=46.5 +lon_0=3 +lat_1=44 +lat_2=49
```

```
+x_0=700000+y_0=6600000 +ellps=GRS80
```

- Transformations Helmert temporelles (15 paramètres)
- Modèles de déformation avec grilles cinématiques

Historique: modernisation

- PROJ 6.0: 2018-2019
 - Auteur: E. Rouault
 - Modèle objet (datums, CRS, CoordinateOperation) en C++ implémentant ISO 19111:2019
 - Base de données SQLite3 de structure inspirée de celles d'ISO 19111 et registre EPSG, mais permettant la gestion simultanée de plusieurs registres
 - Moteur de calcul de pipelines de transformation de coordonnées pour la transformation entre un CRS source et cible. ~~WGS 84: pivot obligatoire~~
 - Import/export aux formats PROJ.4, WKT ESRI, WKT1, WKT2:2015, WKT2:2019
 - Utilitaire ligne de commande projinfo
 - Financement participatif <https://gdalbarn.com>

Historique: modernisation

- PROJ 6.2: 2019

- Import/export de CRS/CoordinateOperation au format PROJJSON: <https://proj.org/specifications/projjson.html>

- Enfin un logo ! P R Ø J

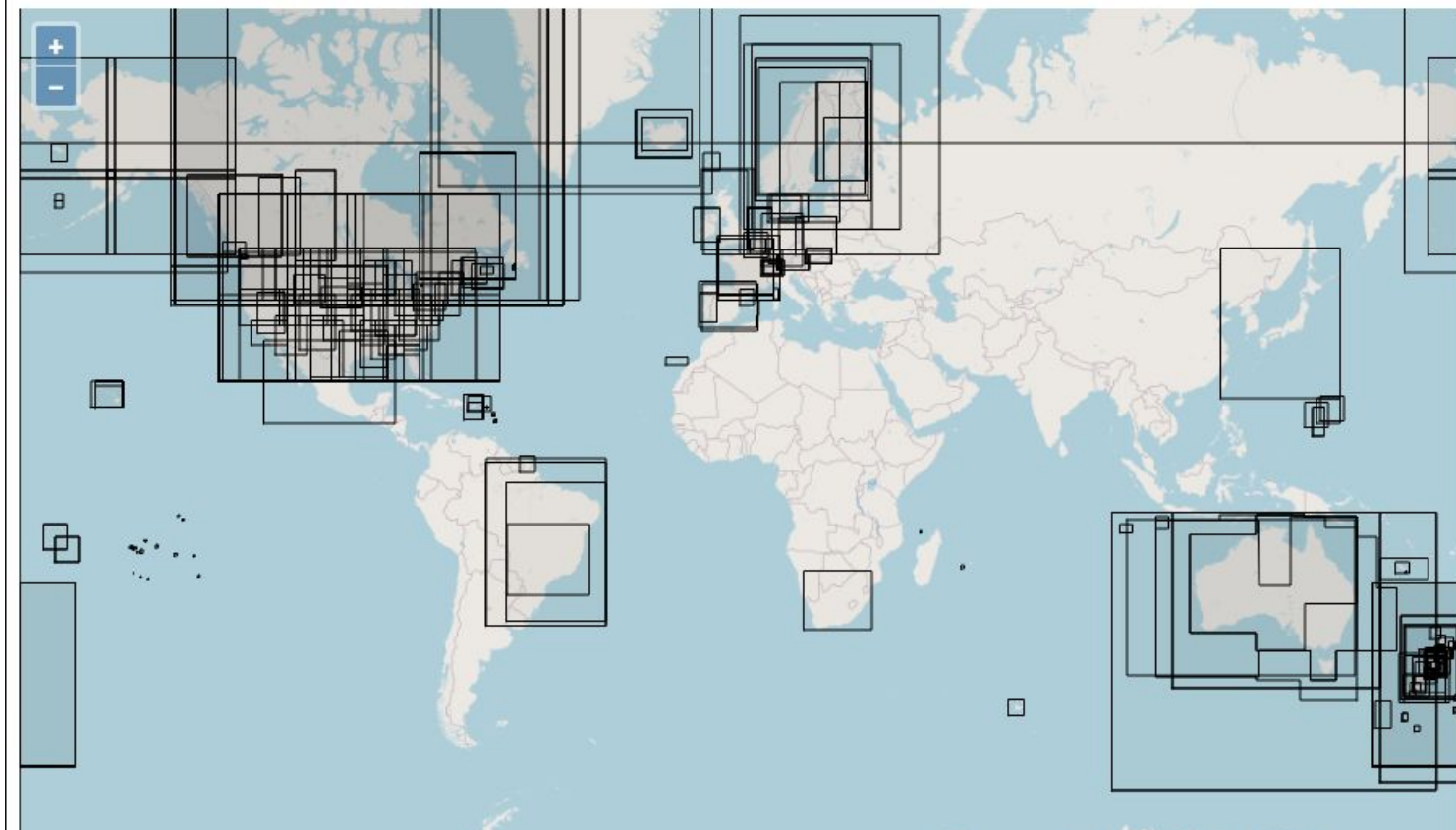
- PROJ 7.0: 2020

- Gestion de grilles au format GeoTIFF (GTG): <https://proj.org/specifications/geodetictiffgrids.html>
- 300 grilles / 500 MB provenant de 33 agences sous license libre disponibles sur <https://cdn.org/> / <https://github.com/OSGeo/PROJ-data>
- Utilisation en local ou accès à la demande par réseau

Grilles hébergées sur cdn.proj.org

Types: Horizontal shift grids Geoid models Vertical shifts Velocity grids Deformation models Other datasets

Agencies: at_bev au_ga au_icsm be_ign br_ibge ca_nrc ca_que_mern ch_swisstopo de_adv de_geosn
 de_lgl_bw de_lgvl_saarland dk_sdfe es_cat_icgc es_ign eur_nkg fi_nls fr_ign is_lmi jp_gsi mx_inegi
 nc_dittt nl_nsgi no_kv nz_linz pl_gugik pt_dgt se_lantmateriet sk_gku uk_os us_nga us_noaa za_cdngi



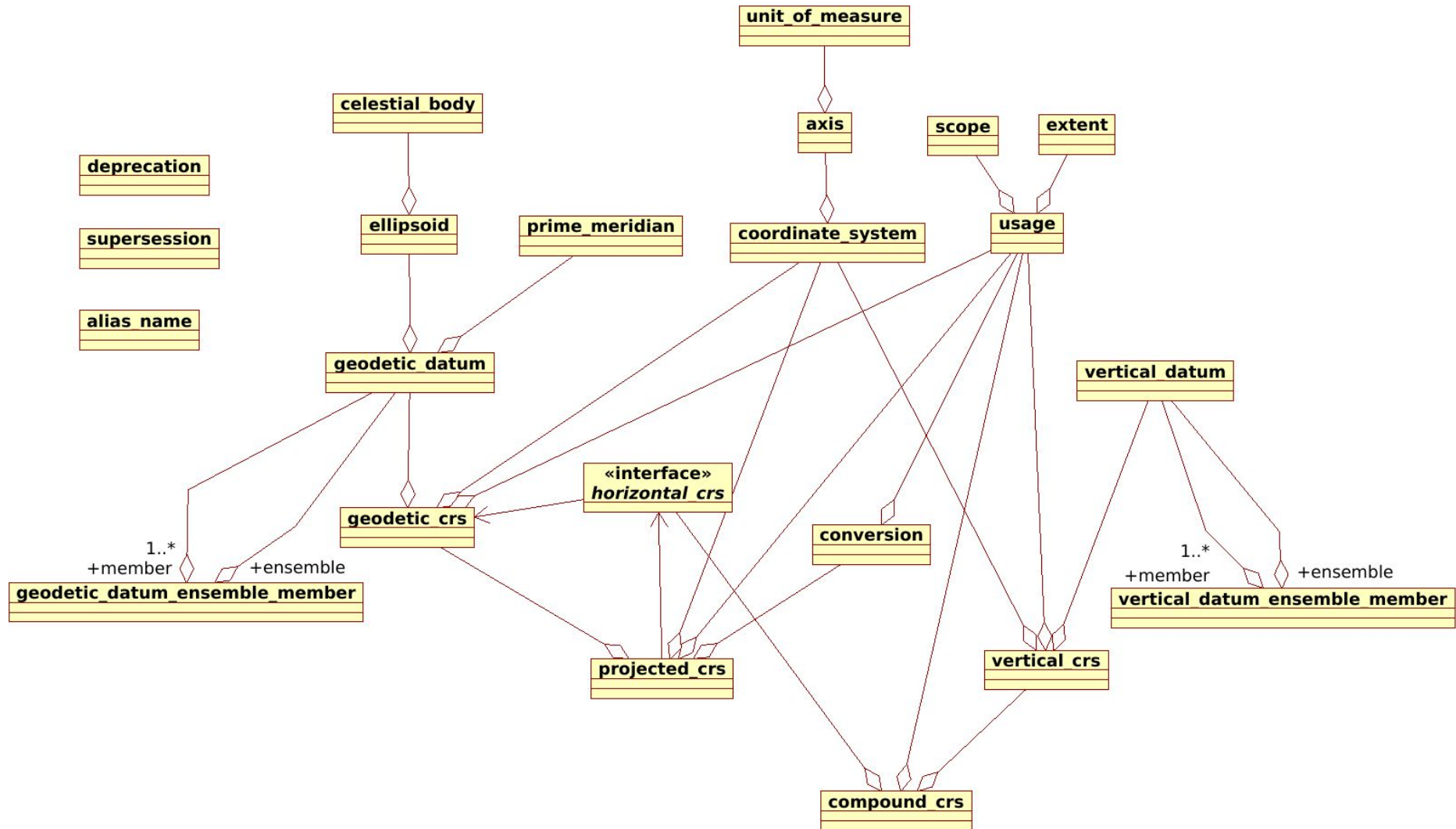
Historique: développements récents

- PROJ 7.1: 2020
 - Modèle de déformation temporel et multi-composantes.
(<https://proj.org/operations/transformations/defmodel.html>)
Cas d'utilisation: NZGD2000 ⇒ ITRF96
- PROJ 7.2: 2020
 - Transformations à base de triangulation irrégulière (TIN)
- PROJ 8.0: 2021
 - Retrait de l'API historique proj_api.h
 - Ajout des CRS planétaires du registre IAU 2015
- PROJ 9.0 / 2022
 - Retrait du système de build autotools

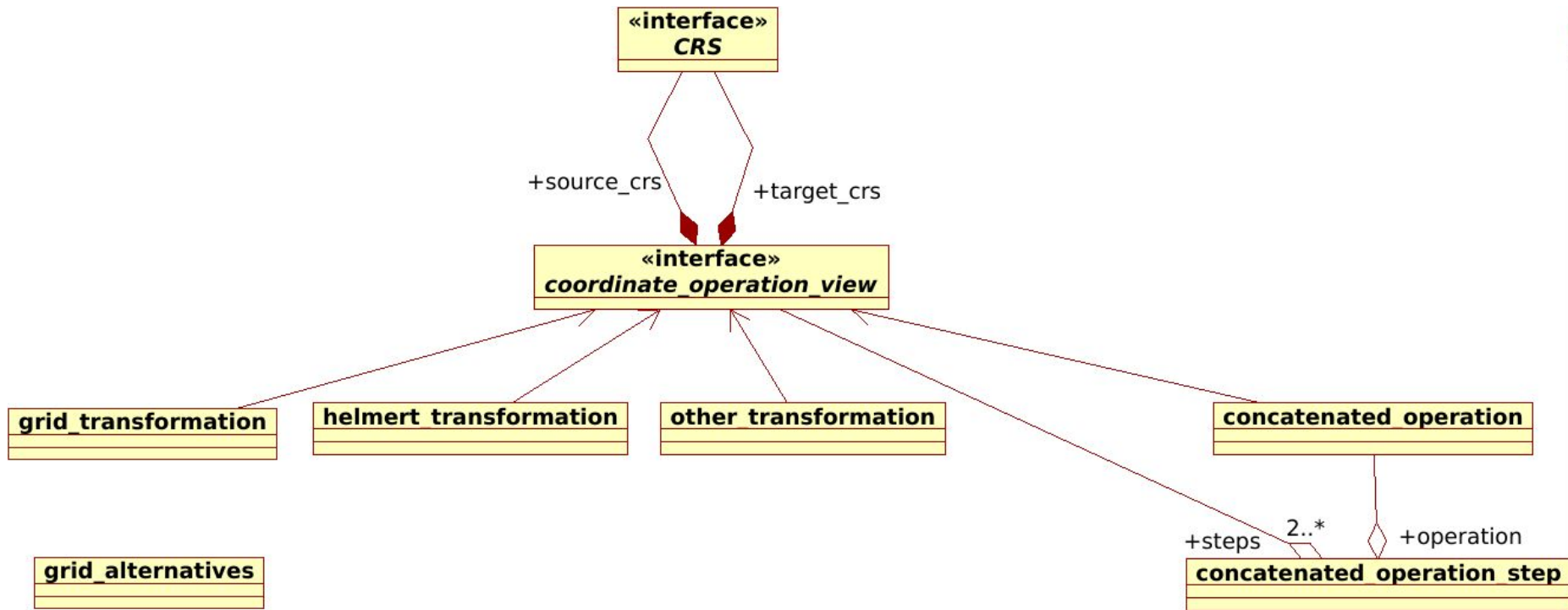
Registres géodésiques intégrés

- EPSG:
 - Script Python d'import des dumps PostgreSQL. Version 10.076 (MàJ régulière)
- IGN France:
 - Script Python d'import du registre XML (version 3.1.0)
 - Contributions bienvenues pour mise à jour!
- ESRI:
 - Script Python d'import des fichiers CSV
- IAU 2015 (CRS planétaires):
 - Script Python d'import du fichier CSV
- NKG (Nordic Geodetic Commission)
 - Fichier SQL écrit "à la main"

Base de données: côté CRS



Base de données: côté transformations



Utilitaires en ligne de commande

- **proj**: application de projection cartographique

```
$ echo 2 49 | proj +proj=utm +zone=31 +ellps=WGS84  
426857.99 5427937.52
```

- **cs2cs**: transformation entre 2 CRS

```
$ echo 2 49 | cs2cs +proj=longlat +datum=WGS84 +to +proj=utm +zone=31 +datum=WGS84  
426857.99 5427937.52 0.00
```

```
$ echo 49 2 | cs2cs EPSG:4326 EPSG:32631  
426857.99 5427937.52 0.00
```

```
$ echo 49 2 | cs2cs "WGS 84" "WGS 84 / UTM zone 31N"  
426857.99 5427937.52 0.00
```

- **cct**: application de pipeline géodésique

```
$ echo 2 49 0 | cct -d 2 +proj=utm +zone=31 +ellps=WGS84  
426857.99 5427937.52 0.00
```

```
$ echo 2 49 0 | cct -d 2 EPSG:16031  
426857.99 5427937.52 0.00
```

projinfo: interrogation base de données

\$ projinfo RGF93

PROJ.4 string:

```
+proj=longlat +ellps=GRS80 +towgs84=0,0,0,0,0,0 +no_defs +type=crs
```

WKT2:2019 string:

```
GEOGCRS["RGF93 v1",  
    DATUM["Reseau Geodesique Francais 1993 v1",  
        ELLIPSOID["GRS 1980",6378137,298.257222101, LENGTHUNIT["metre",1]],  
        PRIMEM["Greenwich",0,ANGLEUNIT["degree",0.0174532925199433]],  
        CS[ellipsoidal,2],  
        AXIS["geodetic latitude (Lat)",north, ORDER[1], ANGLEUNIT["degree",0.0174532925199433]],  
        AXIS["geodetic longitude (Lon)",east, ORDER[2], ANGLEUNIT["degree",0.0174532925199433]],  
        USAGE[  
            SCOPE["Horizontal component of 3D system."],  
            AREA["France - onshore and offshore, mainland and Corsica (France métropolitaine including  
Corsica)."],  
            BBOX[41.15,-9.86,51.56,10.38]],  
    ID["EPSG",4171]]
```

projinfo: déduction pipelines géodésiques

- Composition transformations horizontales et verticales

```
$ projinfo -s "NTF (Paris) + NGF IGN69 height" -t "RGF93 v2b" --3d
```

Operation No. 1:

unknown id, NTF (Paris) to NTF (1) + Inverse of RGF93 v2b to NTF (2) + Inverse of RGF93 v2b to NGF-IGN69 height (5), 1.01 m, France - mainland onshore., at least one grid missing

PROJ string:

```
+proj=pipeline
```

```
+step +proj=axisswap +order=2,1
```

```
+step +proj=unitconvert +xy_in=grad +xy_out=rad
```

```
+step +inv +proj=longlat +ellps=clrk80ign +pm=paris
```

```
+step +proj=hgridshift +grids=fr_ign_ntf_r93.tif
```

```
+step +proj=vgridshift +grids=fr_ign_RAF20.tif +multiplier=1
```

```
+step +proj=unitconvert +xy_in=rad +xy_out=deg
```

```
+step +proj=axisswap +order=2,1
```


projinfo: déduction pipelines géodésiques

```
$ echo 49 2 60 | PROJ_NETWORK=ON cs2cs -d 8 -l \  
    "NTF (Paris) + NGF IGN69 height" "RGF93 v2b" \  
54.44451971 -0.37389654 16.15800095
```

```
$ echo 49 2 60 2022 | PROJ_NETWORK=ON cct -d 8 -l +proj=pipeline \  
+step +proj=axiswap +order=2,1 \  
+step +proj=unitconvert +xy_in=grad +xy_out=rad \  
+step +inv +proj=longlat +ellps=clrk80ign +pm=paris \  
+step +proj=hgridshift +grids=fr_ign_ntf_r93.tif \  
+step +proj=vgridshift +grids=fr_ign_RAF20.tif +multiplier=1 \  
+step +proj=unitconvert +xy_in=rad +xy_out=deg \  
+step +proj=axiswap +order=2,1 \  
54.44451971 -0.37389654 16.15800095 2022
```

projinfo: déduction pipelines géodésiques

- Utilisation de datums intermédiaires:

```
$ projinfo -s "KKJ + N43 height" -t "ETRS89 + N2000 height" --grid-check none
```

unknown id, N43 height to N60 height + N60 height to N2000 height + KKJ to ETRS89 (using PROJ:YKJ_TO_ETRS35FIN), 0.12 m, Finland - onshore mainland south of approximately 66°N., at least one grid missing

PROJ string:

```
+proj=pipeline
```

```
+step +proj=axiswap +order=2,1
```

```
+step +proj=unitconvert +xy_in=deg +xy_out=rad
```

```
+step +proj=tmerc +lat_0=0 +lon_0=27 +k=1 +x_0=3500000 +y_0=0 +ellps=intl
```

```
+step +proj=tinshift +file=fi_nls_n43_n60.json
```

```
+step +proj=tinshift +file=fi_nls_n60_n2000.json
```

```
+step +proj=tinshift +file=fi_nls_ykj_etr35fin.json
```

```
+step +inv +proj=utm +zone=35 +ellps=GRS80
```

```
+step +proj=unitconvert +xy_in=rad +xy_out=deg
```

```
+step +proj=axiswap +order=2,1
```

(Quelques) utilisateurs de PROJ

- GDAL: <https://gdal.org>
- GRASS GIS: <https://grass.osgeo.org/>
- QGIS: <https://qgis.org>
- PDAL: <https://pdal.io>
- Mapserver: <https://mapserver.org>
- Pyproj (Python): <https://pyproj4.github.io/pyproj>
- PROJ-JNI (Java): <https://github.com/OSGeo/PROJ-JNI>
- Ecosystème R: <https://rspatial.org/>
- FME (Safe Software):
<https://www.safe.com/transformers/proj-reprojector/>
- Agences géodésiques: SDFE (Danemark), NLS Finlande et Islande, Kartverket (Norvège), NSGI (Pays-Bas), LINZ (Nouvelle Zélande), NOAA, USGS, ...
- Liste publique: <https://proj.org/users.html>

PROJ en chiffres

- 40 ans d'existence
- 139 contributeurs
- 179 méthodes de projection cartographique, conversion et transformation
- 871 pages de documentation PDF
- ~ 25 000 lignes de coordonnées de référence (couple coordonnée en entrée / sortie)
- ~ 55 000 lignes de code source C++ de tests
- ~ 75 000 lignes de SQL pour la BDD
- ~ 134 000 lignes de code source C++ pour la bibliothèque logicielle et utilitaires

Questions?

Liens:

Documentation: <https://proj.org/>

Dépôt de code: <https://github.com/OSGeo/PROJ/>

Cette présentation:



Contact: even.rouault@spatialys.com



**Diapos supplémentaires:
présentation des travaux “gdalbarn”
/ PROJ 6 / GDAL 3**



PROJ



Revamp of CRS management in the OSGeo C/C++ stack

Even Rouault
SPATIALYS

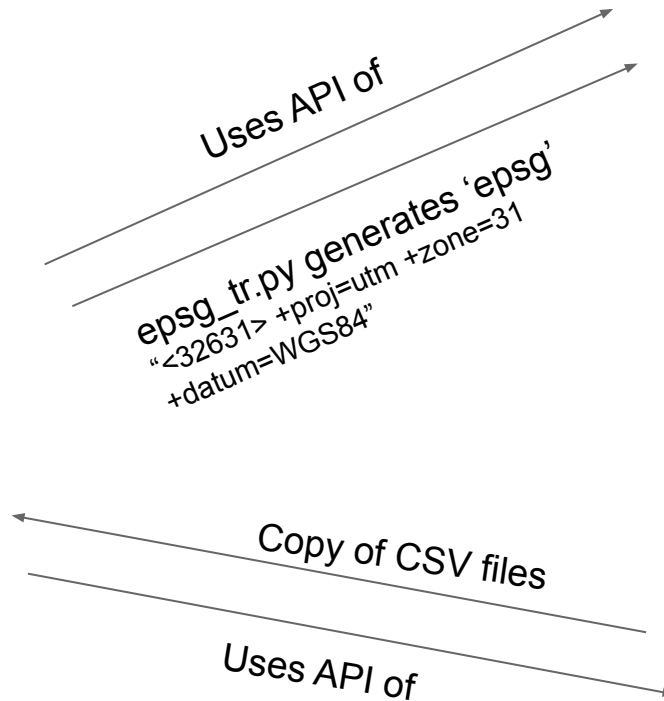
August 28th 2019



Past situation: CRS dictionaries



Many CSV files with
EPSG database
CSV files for ESRI



PROJ

Several text dictionaries:
epsg, IGNF, esri

Uses API
(some services)

libgeotiff

Ingests EPSG database, do
various manipulation and
generates many CSV files

Past situation: coordinate transformation and WGS84 pivot



Source: https://en.wikipedia.org/wiki/File:Good_the_bad_and_the_ugly_poster.jpg

Past situation: coordinate transformation and WGS84 pivot

The good

UTM zone 31 /
WGS84
(EPSG:32631)

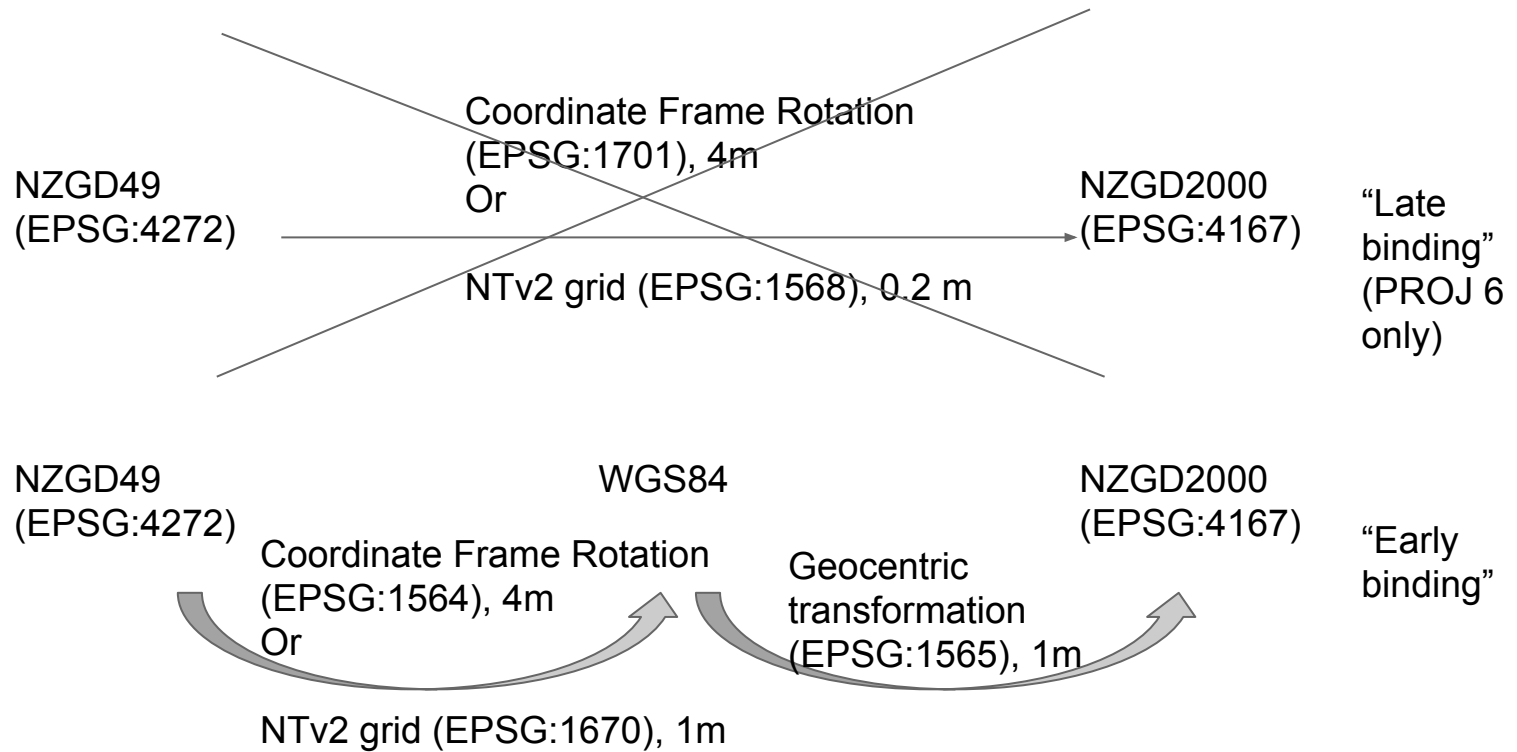
—————→
Inverse map
projection

WGS84 lat/long
(EPSG:4326)

Accuracy: ~ nanometre

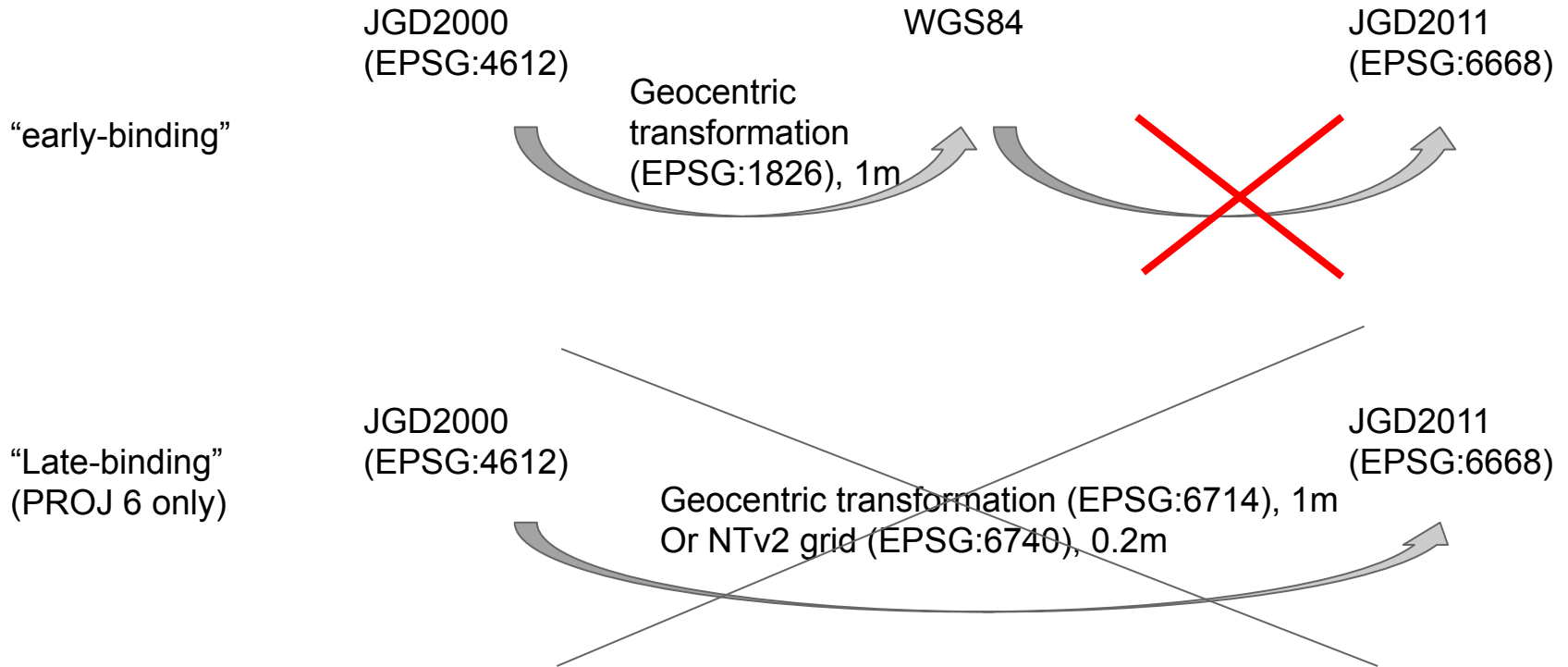
Past situation: coordinate transformation and WGS84 pivot

The ugly
(il brutto)



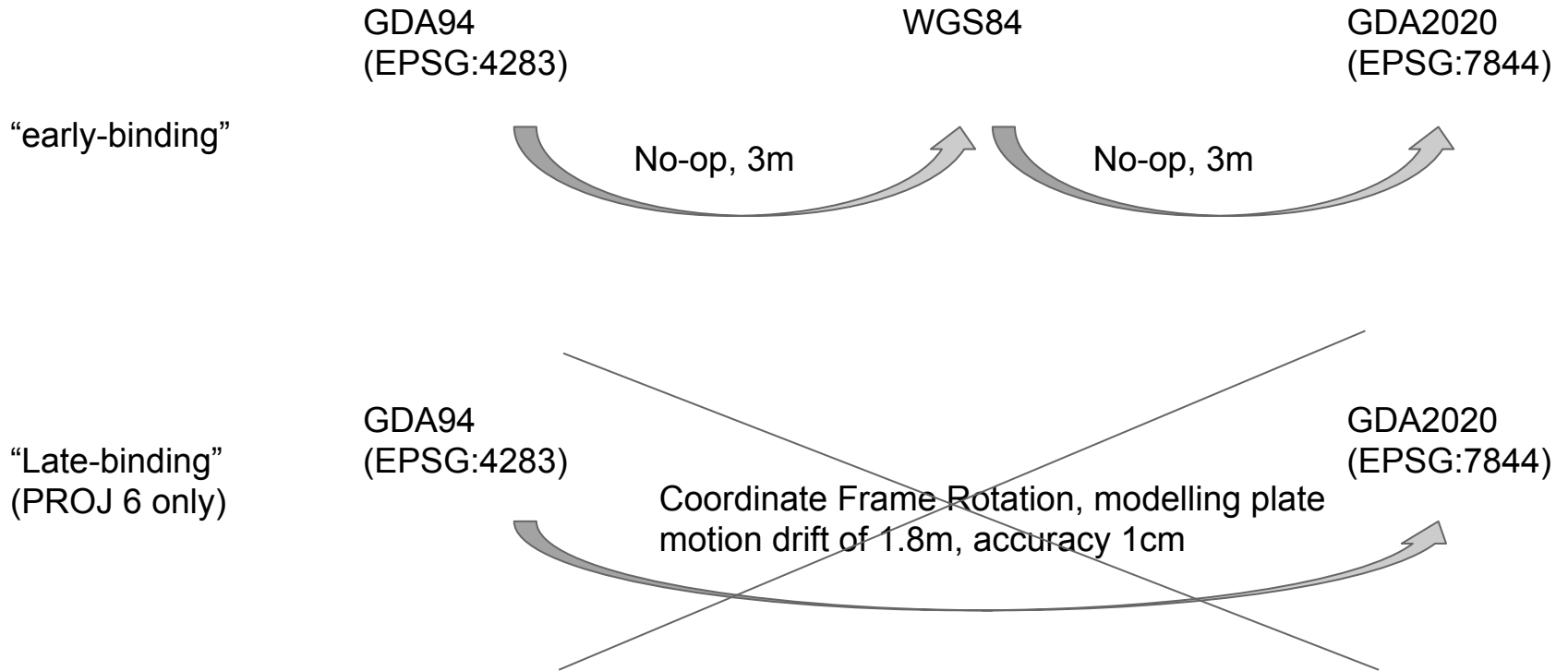
Past situation: coordinate transformation and WGS84 pivot

The bad



Past situation: coordinate transformation and WGS84 pivot

Another bad





gdalbarn.com

PROJ 6 work

- Manages directly WKT (previously in GDAL)
 - ESRI WKT
 - WKT1 (GDAL variant)
 - WKT2:2015 (OGC 12-063r5)
 - WKT2:2019 (OGC 18-010)
- C++ class hierarchy implementing OGC Abstract Topic 2 (Referencing by coordinates) / ISO-19111:2019
- # of LOC doubled !

PROJ 6 work

- SQLite database to store SRS definition
 - No longer CSV files duplicated among software
 - Better query capabilities
 - Multi-catalog: EPSG, ESRI, IGNF
- “Late-binding” CRS transformations
- Use of area of uses
- Temporal component handling

PROJ 6 work

- New command line utility: projinfo
 - Query for CRS or coordinate operation
 - Guess EPSG codes from PROJ or WKT
- Output as PROJ or WKT strings

- Examples:

- projinfo EPSG:4326
- projinfo -s GDA94 -t GDA2020 --summary

Candidate operations found: 5

EPSG:8048, GDA94 to GDA2020 (1), 0.01 m, Australia - GDA

EPSG:8447, GDA94 to GDA2020 (2), 0.05 m, Australia - onshore, at least one grid missing

EPSG:8446, GDA94 to GDA2020 (3), 0.05 m, Australia - onshore, at least one grid missing

EPSG:8445, GDA94 to GDA2020 (5), 0.05 m, Cocos (Keeling) Islands - onshore, at least one grid missing

EPSG:8444, GDA94 to GDA2020 (4), 0.05 m, Christmas Island - onshore, at least one grid missing

Attendance participation needed...

Question: what is the CRS designated by this PROJ string ?

```
+proj=longlat +ellps=GRS80  
+towgs84=0,0,0,0,0,0,0 +no_defs
```

Do **NOT** use PROJ strings for CRS !

Answer:

- ETRS89, (Europe)
 - GDA94, (Australia)
 - GDA2020, (Australia)
 - JGD2000, (Japan)
 - NZGD2000, (New Zealand)
 - NAD83(2011), (North America),
 - SIRGAS 2000, (Latin America)
 - And many others !
- ⇒ Use EPSG codes or WKT

GDAL 3 work

- PROJ 6 is a required dependency
- All WKT support has been moved to PROJ.
Support for WKT2:2015 and :2019
- No longer any CSV file related to CRS →
use PROJ database
- Time-dependent coordinate operations
possible.
- `-ct {proj_pipeline}` switch to `ogr2ogr`,
`gdalwarp`, `gdaltransform`
- EPSG axis order compliant by default !

libgeotiff 1.5 work

- PROJ 6 is a required dependency
- No longer any CSV file related to CRS → use PROJ database

Adoption

- PROJ 6.1, GDAL 3.0.1 and libgeotiff 1.5.1 are out
- PROJ 6 adoption:
 - pyproj 2.0
 - rasterio 1.0.25
 - QGIS 3.8 can use PROJ 6 capabilities (require manual building for now)
 - GRASS 7.8
 - PostGIS 3.0dev
 - PDAL 2
 - MapServer 8
 - Full list at <https://github.com/OSGeo/PROJ/wiki/proj.h-adoption-status>

Thanks to the sponsors of GDAL barn !

