



# GDAL 2.3

## 20 ans déjà et la tête dans les nuages !

Even Rouault  
*SPATIALYS*

# Qui suis-je ?

- Contributeur à GDAL/OGR depuis 2007 et responsable du comité de pilotage depuis 2015.
- Contributeur à MapServer, QGIS
- Co-mainteneur: libtiff, libgeotiff, PROJ.4
- Fondateur de Spatialys, SS2L dans la géomatique.  
Forte expertise sur GDAL / MapServer

# GDAL/OGR : Introduction

- GDAL? Geospatial Data Abstraction Library. Le couteau suisse du géomaticien
- Raster (GDAL) et Vecteur (OGR)
- Accès lecture/écriture à plus de 240 formats et protocoles de données (principalement) géospatiales.
- Utilisé très largement (Open Source et propriétaire): QGIS, GRASS, MapServer, Mapnik, PostGIS, OTB, SAGA, FME, ArcGIS, Google Earth...  
(> 100 <http://trac.osgeo.org/gdal/wiki/SoftwareUsingGdal>)
- License open-source MIT/X (permissive)

# GDAL/OGR : Introduction

- GDAL? Geospatial Data Abstraction Library. Le couteau suisse du géomaticien
- Raster (GDAL) et Vecteur (OGR)
- Accès lecture/écriture à plus de 240 formats et protocoles de données (principalement) géospatiales.
- Utilisé très largement (Open Source et propriétaire): QGIS, GRASS, MapServer, Mapnik, PostGIS, OTB, SAGA, FME, ArcGIS, Google Earth...

(> 100 <http://trac.osgeo.org/gdal/wiki/SoftwareUsingGdal>)

- License open-source MIT/X (permissive)
- Débuté en 1998 par Frank Warmerdam



# GDAL/OGR 2.2 en résumé

- V2.2.0: mai 2017 → V2.2.4: mars 2018
- 4 RFCs implémentées dans le cycle 2.0 dont:
  - Gestion des zones sans données dans les pilotes GeoTIFF et VRT
  - Nouvelles géométries: surfaces polyédrales, réseaux irréguliers triangulés (TIN)
  - Distinction entre attribut nul et manquant (pour JSon et GML)
- 7 nouveaux drivers dont:
  - CAD: lecture fichiers DWG R2000
  - DGNv8: lecture/écriture format propriétaire DGN v8 (utilisation SDK propriétaire)
  - GMLAS: lecture/écriture de fichiers XML/GML pilotée par les schémas applicatifs (“Complex features”)

# GDAL/OGR 2.3

- Sortie le 4 mai 2018
- 3801 “commits” (total depuis 1998: 37988)
- 3 RFCs implémentées dans le cycle 2.3
  - Utilisation de C++11
  - Migration de Trac+Subversion vers GitHub
    - La base de tickets existants reste sous Trac
    - Les nouveaux tickets à créer sous GitHub
  - Auto-détection du format de sortie des utilitaires en fonction de l’extension
    - `gdal_translate mon.tif mon.png`
    - `ogr2ogr mon.gpkg mon.shp`

# GDAL/OGR 2.3

- 2 nouveaux pilotes raster
  - PDS4: Planetary Data System v4 (format NASA)
    - Entête XML pour métadonnées + fichier brut pour l'imagerie.
    - Lecture/écriture
  - RDA: DigitalGlobe Raster Data Access. Accès en lecture aux images via l'API REST
- 2 nouveaux pilotes vecteur:
  - MVT: MapBox Vector Tiles
  - WFS3
- Drivers ESRIJSON et TOPOJSON séparés du driver GeoJSON existant (périmètres fonctionnels inchangés)

# Pilote MVT (Mapbox Vector Tiles)

- Lecture/écriture
- Sous-formats: une tuile par fichier ou MBTILES

/metadata.json



Liste couches et attributs

/0/

0/

0.pbf

/1/

0/

0.pbf

1.pbf

1/

0.pbf

1.pbf



# Pilote MVT

- Lecture:

- sur disque ou tuiles sur le réseau

```
ogrinfo MVT:https://free.tilehosting.com/data/v3/1 -oo tile_extension="pbf.pict?key=${YOUR_KEY}" \
-oo metadata_file="https://free.tilehosting.com/data/v3.json?key=${YOUR_KEY}"
```

- clipping de géométrie ou non

- Ecriture:

- Paramétrage des niveaux de zoom
- Choix de la résolution et du buffer
- Paramétrage de la simplification géométrique
- Assignment de couche par niveaux de zoom

# Pilote WFS3

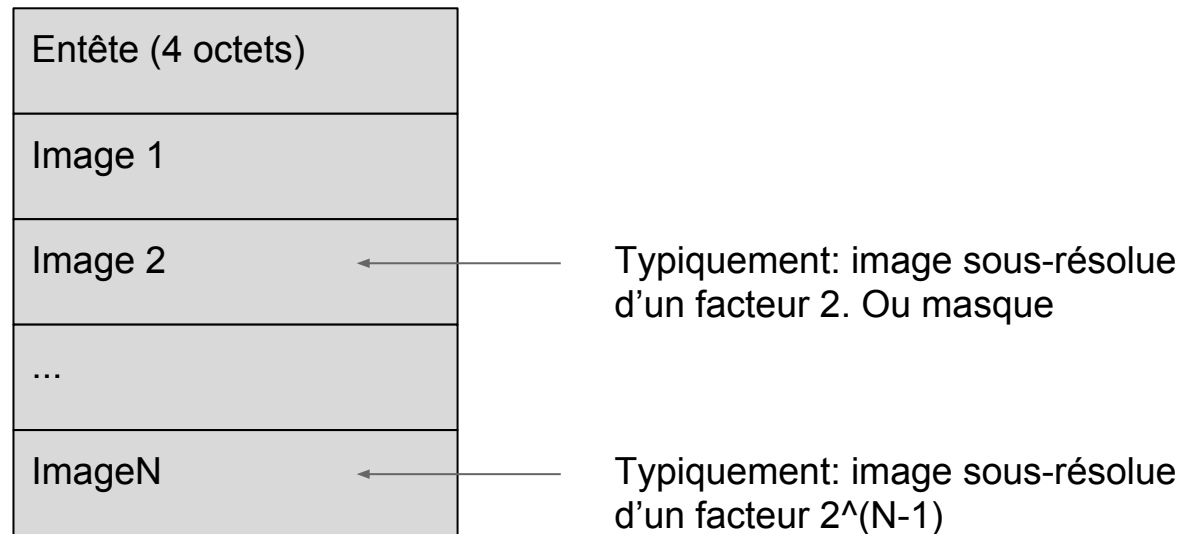
- EXPERIMENTAL! basé sur la version de la spécification alpha de début mars 2018 qui a évolué depuis.
- WFS v3:
  - Rupture par rapport aux versions précédentes
  - Philosophie REST
    - GET /collections
    - GET  
/collections/{name}/items?bbox=160.6,-55.95,-170,-25.89
    - GET /collections/{name}/items/{id}
  - Fin du tout XML. Négociation de contenu
  - OpenAPI 3.0
  - Coeur minimal, facile à implémenter

# Cloud Optimized GeoTIFF (COG)

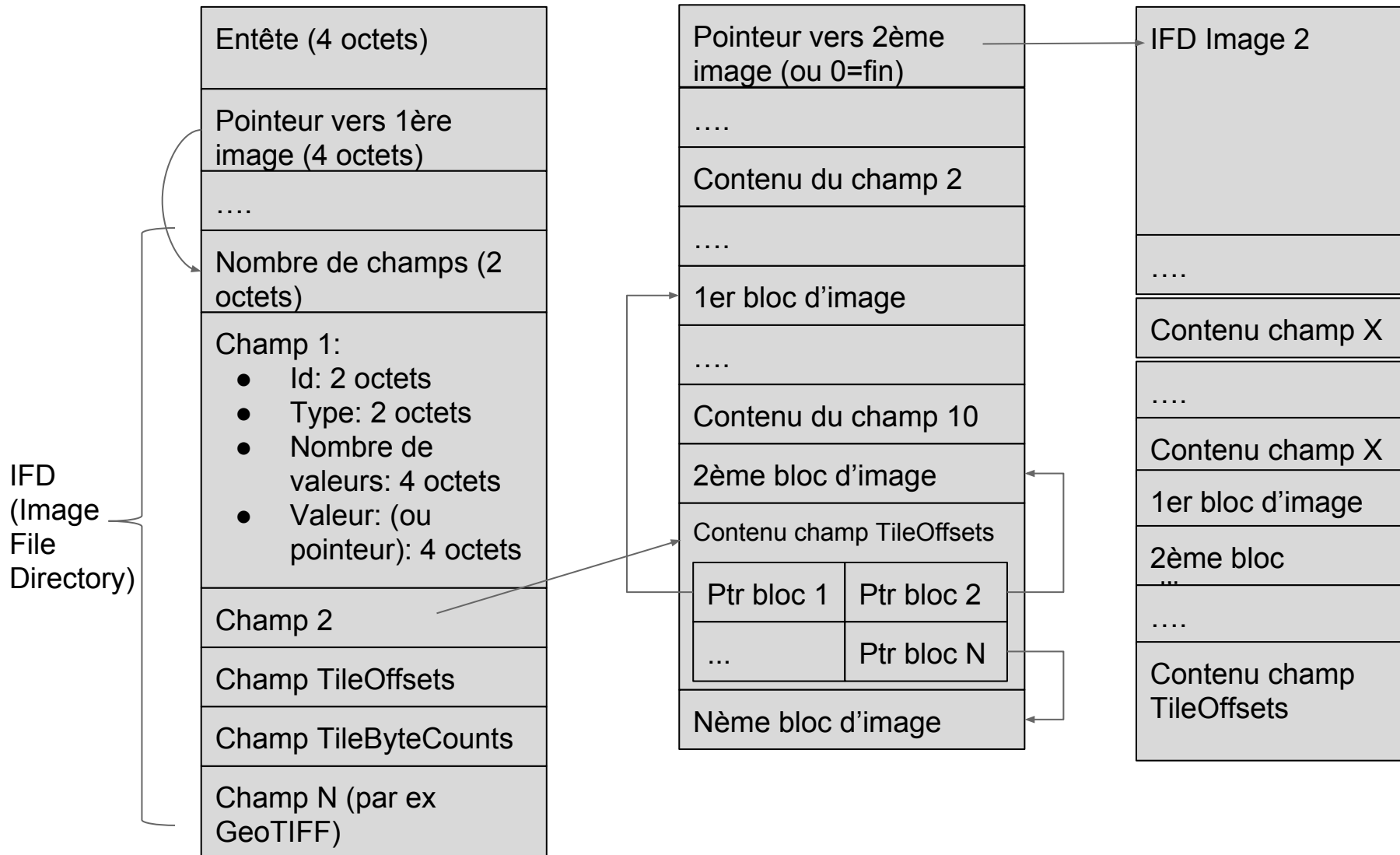
- “Profil” du format TIFF optimisé pour lecture au travers d’HTTP
- But: accès efficace à des parties de fichier sans serveur spécialisé
- ~ WCS avec uniquement serveur de fichier
- Côté client: utilisation de requête HTTP GET avec en-tête Range
- Adoption croissante par les logiciels et fournisseurs de données



# Structure fichier TIFF: haut niveau



# Structure fichier TIFF: bas niveau



# Structure d'un fichier COG

Métadonnées

Entête (4 octets)

IFD 1: image résolution 100%

Valeur des champs IFD 1  
(dont offsets et taille de blocs)

IFD 2: image résolution 50%

Valeur des champs IFD 2

IFD 3: image résolution 25%

Valeur des champs IFD 3

Imagerie IFD 3

Imagerie IFD 2

Imagerie IFD 1

Block X=1 Y=1

Block X=2 Y=1

Block X=N Y=1

Block X=2 Y=1

Block X=N Y=M

Imagerie

# Génération d'un fichier COG

- Création d'un fichier GeoTIFF:
  - `gdal_translate source temp.tif`
- Création de ses pyramides:
  - `gdaladdo -r average temp.tif`
- Création du COG:
  - `gdal_translate temp.tif cog.tif \`  
    `-co TILED=YES \`  
    `-co COMPRESS=DEFLATE \ (ou JPEG -co`  
    `PHOTOMETRIC=YCBCR)`  
    `-co COPY_SRC_OVERVIEWS=YES`

# Systemes de fichiers virtuels réseau

- /vsicurl/
  - gdalinfo  
/vsicurl/[http://landsat-pds.s3.amazonaws.com/L8/139/045/LC81390452014295LGN00/LC81390452014295LGN00\\_B1.TIF](http://landsat-pds.s3.amazonaws.com/L8/139/045/LC81390452014295LGN00/LC81390452014295LGN00_B1.TIF)
- /vsi3/ : Amazon AWS S3
- /vsigs/ : Google Cloud Storage
- /vsiaz/ : Microsoft Azure Blob storage
- /vsiswift/ : OpenStack SWIFT
- /vsioss/ : Alibaba Object Storage Service
- Tous disponibles en lecture (aléatoire) / écriture (séquentielle)
- Version “streamée” pour la lecture: /vsicurl\_streaming/, /vsi3\_streaming/ etc,...
- Fonction VSIGetSignedURL() pour obtenir une URL signée

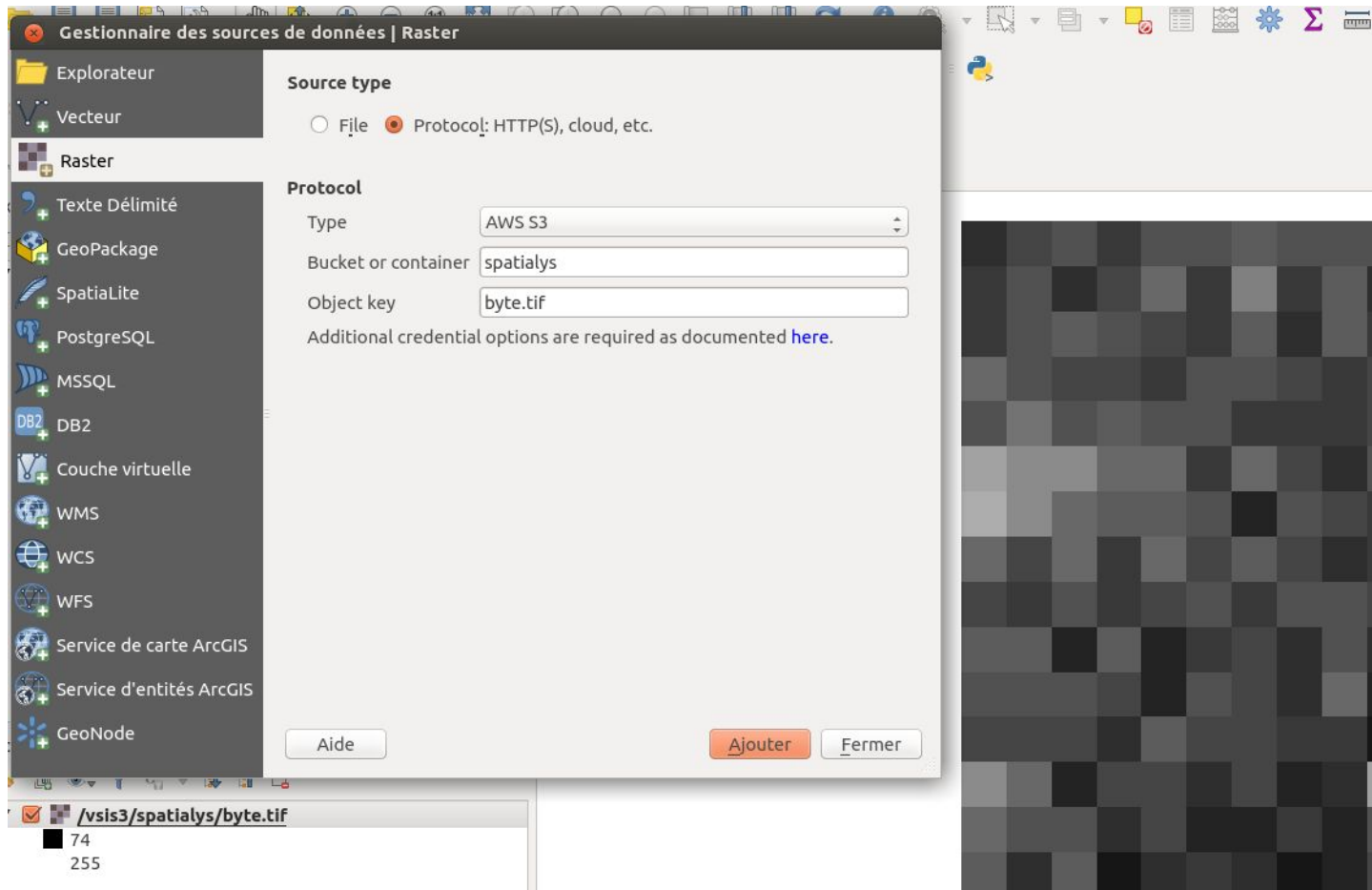


# Systemes de fichiers virtuels réseau

- Plusieurs méthodes d'authentification gérées
  - /vsi3:
    - AWS\_SECRET\_ACCESS\_KEY + AWS\_ACCESS\_KEY\_ID
    - ~/.aws/credentials (AWS\_PROFILE)
    - ~/.aws/config
    - Identification automatique sur AWS EC2
  - /vsigs:
    - GS\_SECRET\_ACCESS\_KEY + GS\_ACCESS\_KEY\_ID
    - GDAL\_HTTP\_HEADER\_FILE
    - GS\_OAUTH2\_REFRESH\_TOKEN
    - GS\_OAUTH2\_PRIVATE\_KEY + GS\_OAUTH2\_CLIENT\_EMAIL
    - ~/.boto
    - Identification automatique sur Google Compute Engine
- Utilitaires:
  - <https://github.com/OSGeo/gdal/tree/master/gdal/swig/python/samples>
    - gdal\_ls.py
    - gdal\_cp.py
    - gdal\_rm.py

# Systèmes de fichiers virtuels réseau

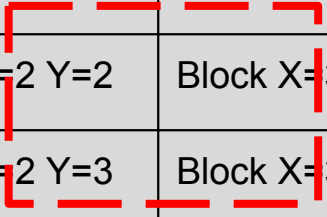
- Lecture facilitée dans QGIS 3.2 (*Mathieu Pellerin, iMHere Asia*)



# Systemes de fichiers virtuels réseau

Amélioration du driver GeoTIFF pour requêtes HTTP plus efficaces

Block X=1 Y=1	Block X=2 Y=1	Block X=3 Y=1
Block X=1 Y=2	Block X=2 Y=2	Block X=3 Y=2
Block X=1 Y=3	Block X=2 Y=3	Block X=3 Y=3



Block X=N Y=1
Block X=N Y=2
Block X=N Y=3

Block X=1 Y=M	Block X=2 Y=M	Block X=3 Y=M
---------------	---------------	---------------

Block X=N Y=M
---------------

⇒ 3 requêtes GET Range parallélisées

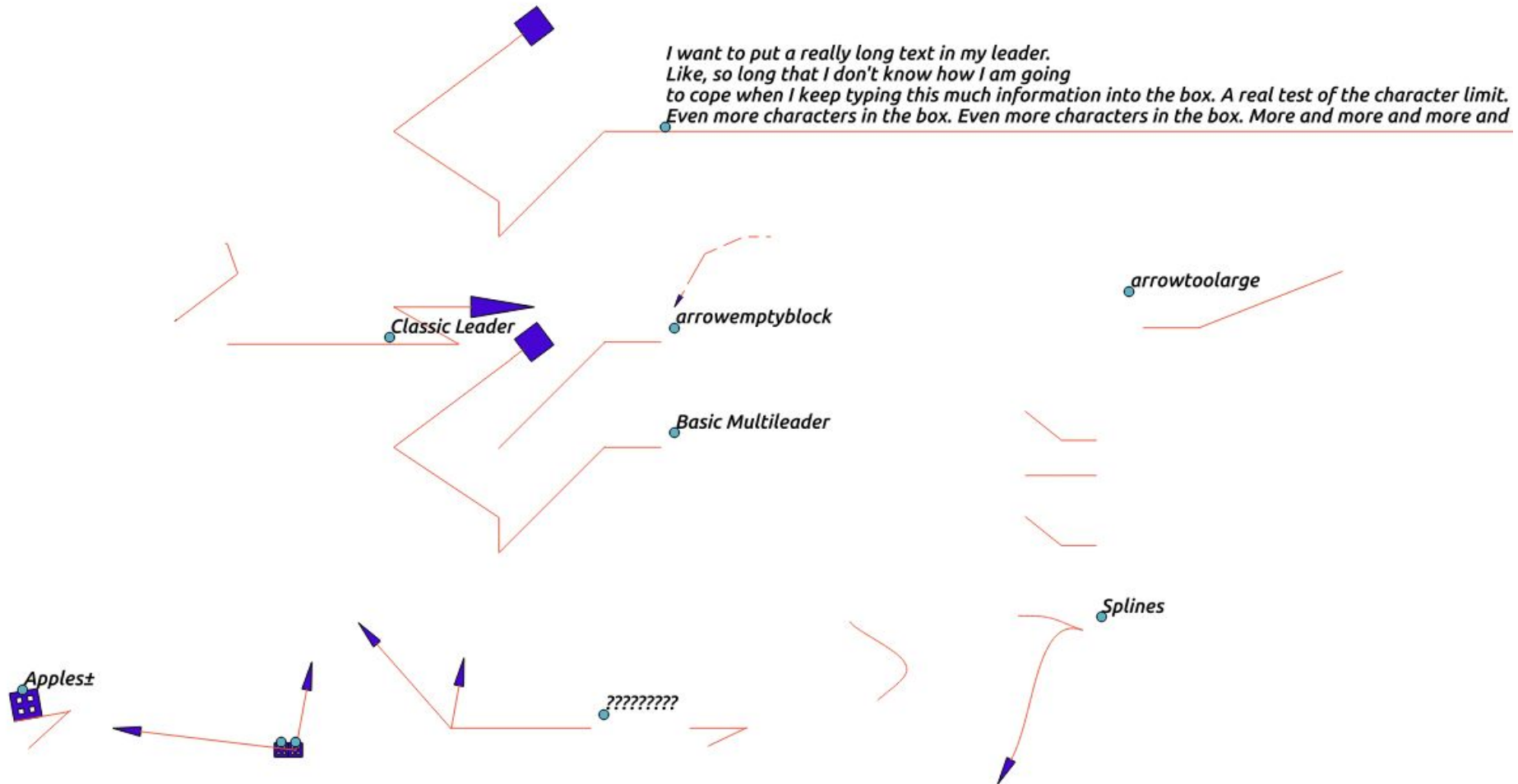
# COG et systèmes de fichiers: liens utiles

- Documentation générale
  - <http://www.cogeo.org>
  - <https://trac.osgeo.org/gdal/wiki/CloudOptimizedGeoTIFF>
- Clients Javascript:
  - <https://geotiffjs.github.io/cog-explorer> (EOX)
  - <http://www.cogeo.org/map/> (Radiant Earth) + tiles.rdnt.io (serveur)
- Validation:
  - <http://cog-validate.radiant.earth/html>
  - [https://github.com/rouault/cog\\_validator](https://github.com/rouault/cog_validator)
- Documentation driver GeoTIFF GDAL:
  - [http://gdal.org/frmt\\_gtiff.html](http://gdal.org/frmt_gtiff.html)
- Documentation systèmes de fichiers virtuels:
  - [http://gdal.org/gdal\\_virtual\\_file\\_systems.html](http://gdal.org/gdal_virtual_file_systems.html)

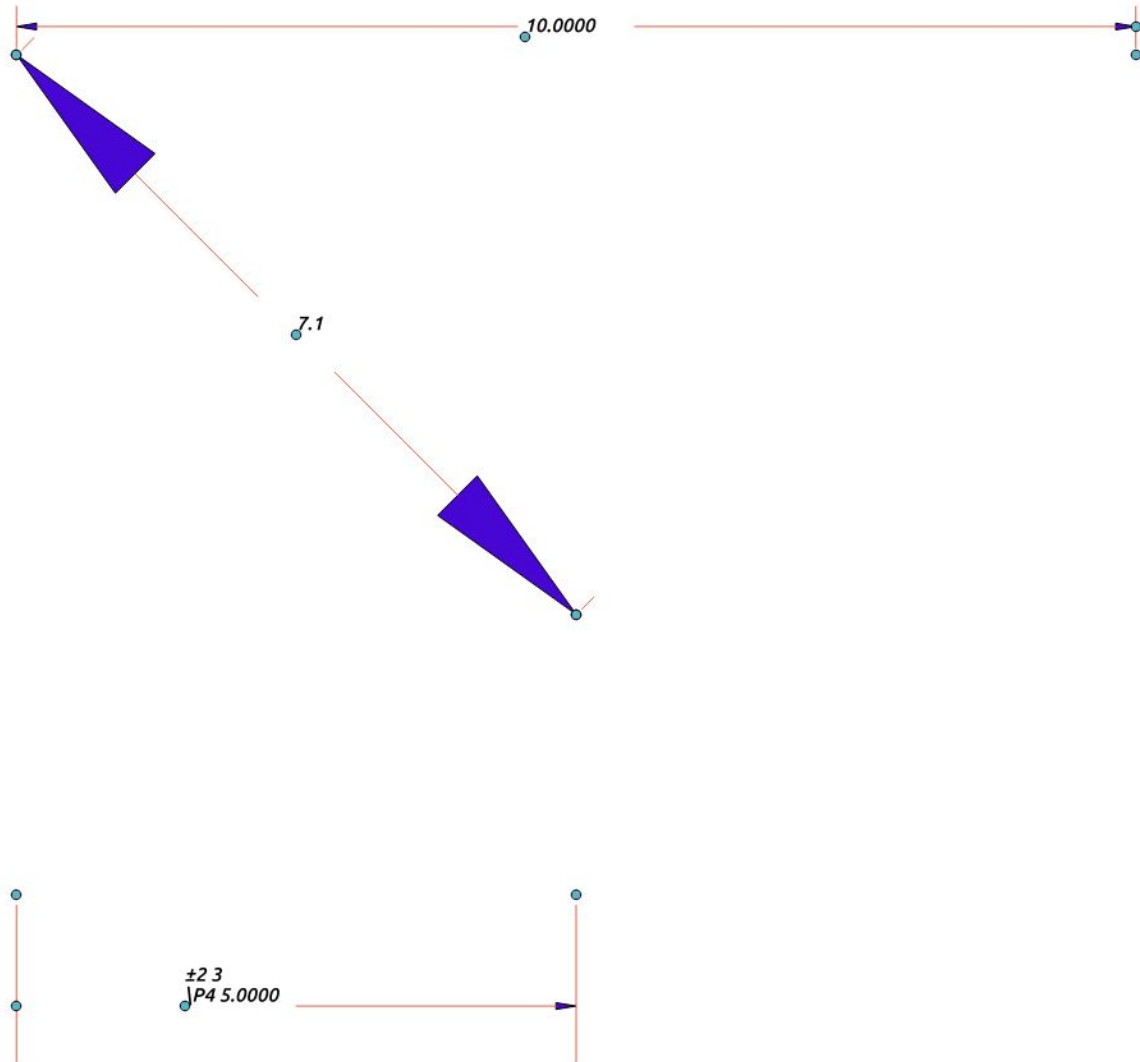
# GDAL/OGR 2.3: autres changements

- Gestion de PROJ v5
- Mise à jour vers base de SRS EPSG v9.2
- Meilleure identification des codes EPSG pour les shapefiles
- Améliorations dans les pilotes:
  - GeoJSON: lecture de fichiers arbitrairement grands
  - GRIB: création de fichiers GRIB2
  - WCS:
    - ajout de la gestion de WCS 2.0(*Ari Jolma*)
    - Cache disque de métadonnées
    - Multiples options pour gérer non-conformités
  - DXF: multiples améliorations (*Alan Thomas / ThinkSpatial*  
*thinkspatial.com.au*)

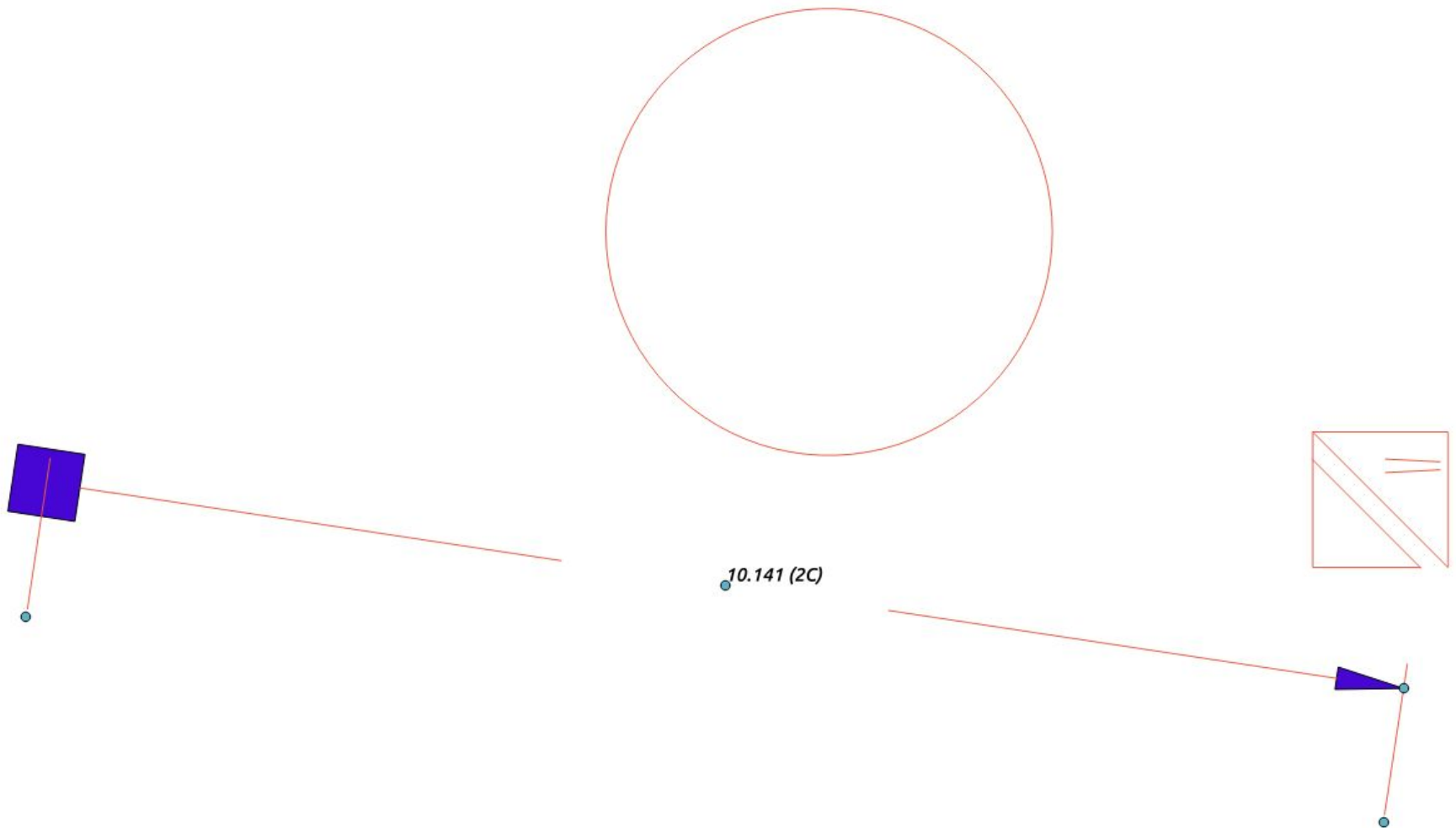
# DXF: LEADER / Ligne de repères



# DXF: DIMENSION / Ligne de cotes

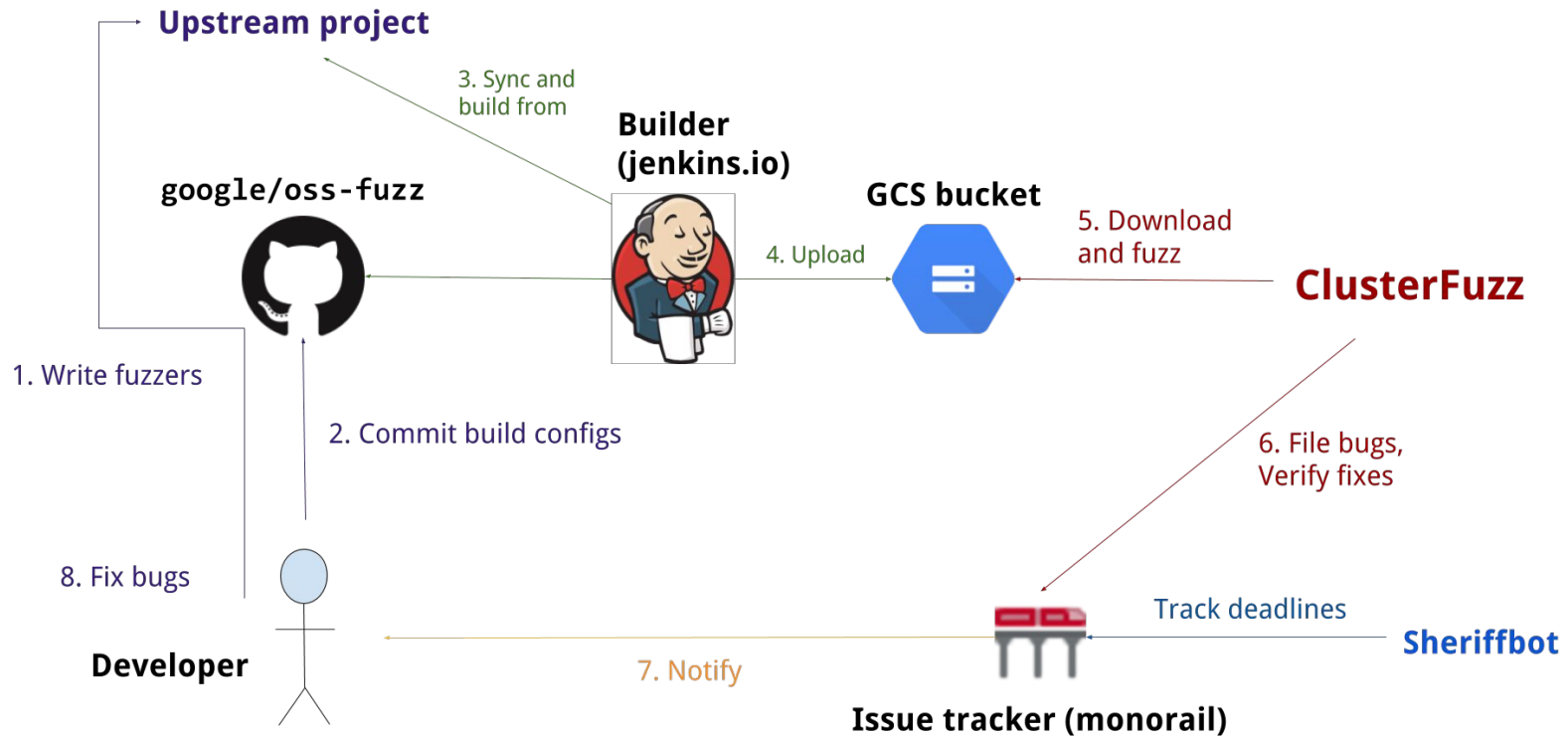


# DXF: BLOCK





# Intégration au projet oss-fuzz



(source: <https://github.com/google/oss-fuzz>)

> 1000 problèmes corrigés

# Activité communautaire

- 19 développeurs avec accès GitHub
- 95 contributeurs durant le cycle 2.3.0
- 2295 inscrits à gdal-dev.
  - 1924 messages mai 2017→ mai 2018
- ~470 tickets créés / 12 derniers mois (7310 au total).
  - ~480 ouverts

**Et après ?**





***[gdalbarn.com](http://gdalbarn.com)***



# “GDAL SRS barn”: [gdalbarn.com](http://gdalbarn.com)

- Projet de modernisation et refonte profonde des systèmes de coordonnées de référence dans GDAL, libgeotiff et PROJ
- Opportunité liée à la sortie de PROJ 5.0
- Adoption du standard OGC WKTV2 (12-063r5) / ISO 19162
  - Résolution de problèmes d'interopérabilité
  - Gestion de la dimension temporelle
  - Meilleure gestion de la dimension verticale



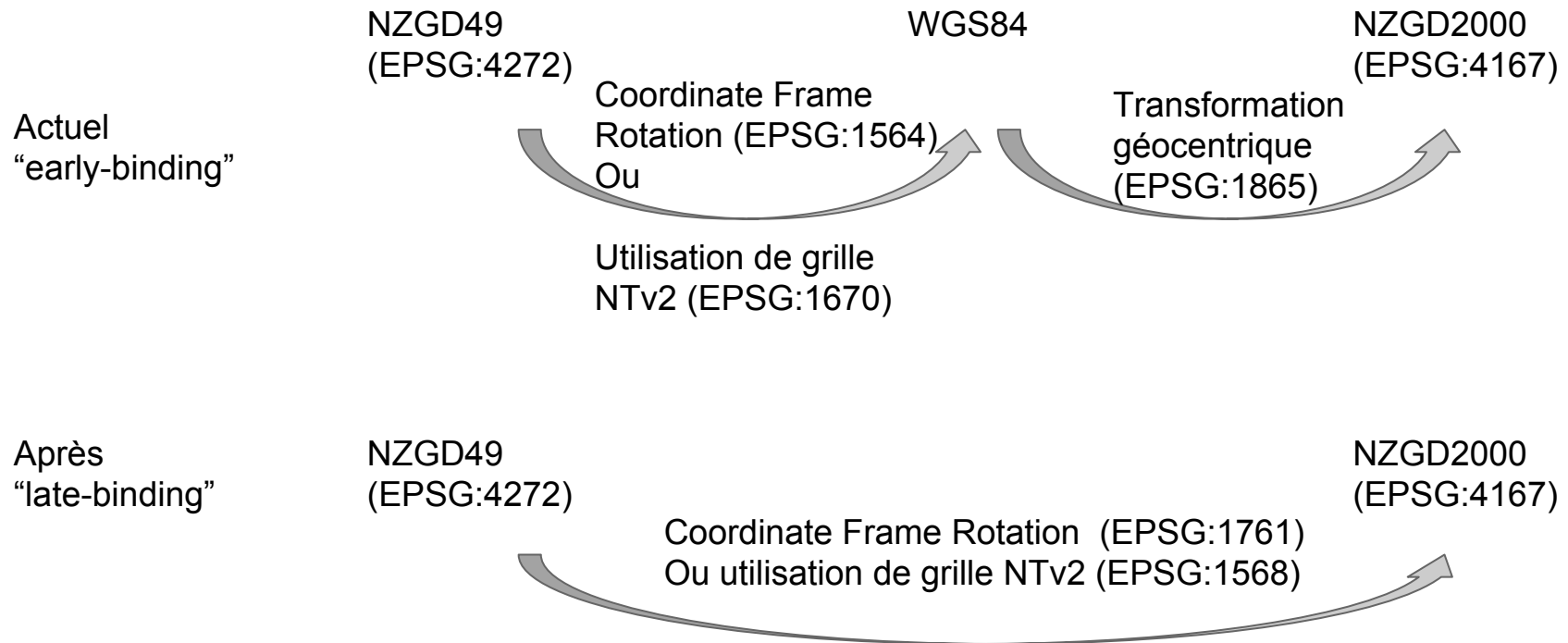
# “GDAL SRS barn”: [gdalbarn.com](http://gdalbarn.com)

- Base de données SQLite pour stocker les définitions de SRS
  - Plus de fichiers CSV dupliqués entre logiciels
  - Meilleure capacité d'interrogation
- Utilisation des aires de validité
- Gestion de la composante temporelle
  - Transformation de Helmert avec termes dérivés
  - Transformations avec modèles de déformation



# “GDAL SRS barn”: gdalbarn.com

- Fin de WGS84 comme système pivot pour les transformations entre datum

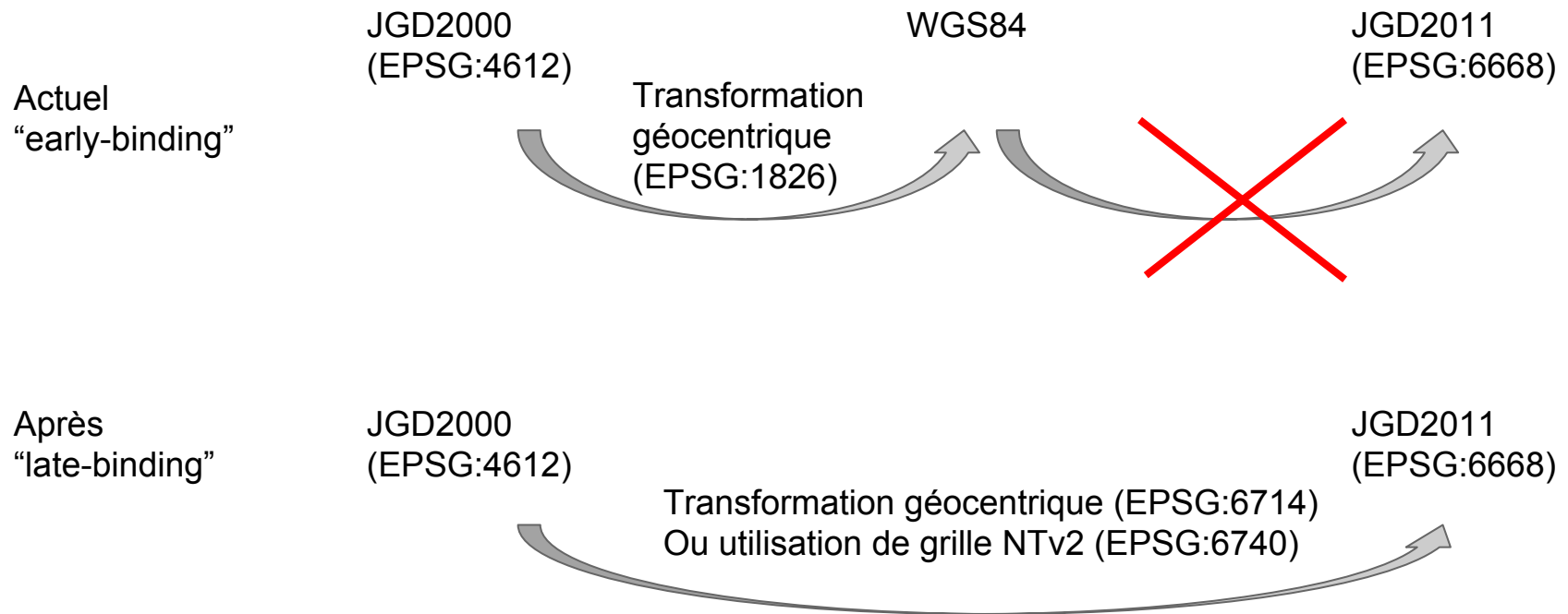






# “GDAL SRS barn”: gdalbarn.com

- Fin de WGS84 comme système pivot pour les transformations entre datum



# Merci aux sponsors de GDAL barn !



**SAFE SOFTWARE™**



Timoney Group



# Questions?

Liens:

<http://www.gdal.org/>

<https://trac.osgeo.org/gdal/wiki/Release/2.3.0-News>

<https://trac.osgeo.org/gdal/wiki/RfcList>

Contact: [even.rouault@spatialys.com](mailto:even.rouault@spatialys.com)

