# GDAL 2.1
# What’s new ?

Even Rouault - *SPATIALYS*
Dmitry Baryshnikov - *NextGIS*
Ari Jolma

August 25th 2016

# Plan

- Introduction to GDAL/OGR
- Community
- GDAL 2.1 : new features
- Future directions

# GDAL/OGR : Introduction

- GDAL? Geospatial Data Abstraction Library. The swiss army knife for geospatial.
- Raster (GDAL) and Vector (OGR)
- Read/write access to more than 200 (mainly) geospatial formats and protocols.
- Widely used (FOSS & proprietary): GRASS, MapServer, Mapnik, QGIS, gvSIG, PostGIS, OTB, SAGA, FME, ArcGIS, Google Earth…

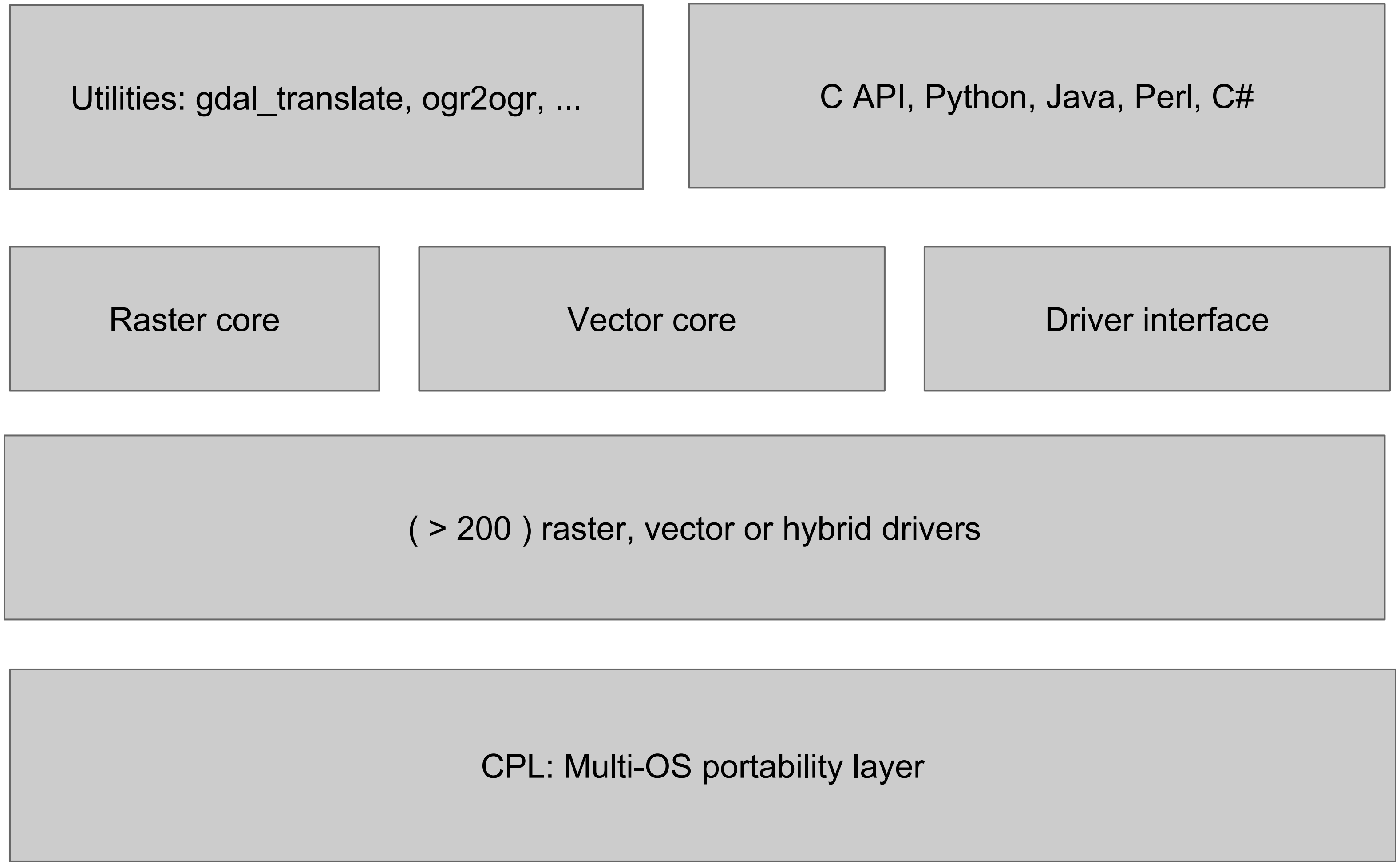
(> 100 http://trac.osgeo.org/gdal/wiki/SoftwareUsingGdal)

- Started in 1998 by Frank Warmerdam
- A project of OSGeo since 2008
- MIT/X Open Source license (permissive)
- > 1M lines of code for library + utilities, ...
- > 150K lines of test in Python

# Main features

- Format support through drivers implemented a common interface
- Support datasets of arbitrary size with limited resources
- C++ library with C API
- Multi OS: Linux, Windows, MacOSX/iOS, Android, ...
- Language bindings: Python, Perl, C#, Java,...
- Utilities for translation,reprojection, subsetting, mosaicing, interpolating, indexing, tiling…
- Can work with local, remote (/vsicurl), compressed (/vsizip/, /vsigzip/, /vsitar), in-memory (/vsimem) files

# General architecture



Utilities: gdal_translate, ogr2ogr, ...

C API, Python, Java, Perl, C#

Raster core

Vector core

Driver interface

( > 200 ) raster, vector or hybrid drivers

CPL: Multi-OS portability layer

# Raster Features

- Efficient support for large images (tiling, overviews)
- Several georeferencing methods: affine transform, ground control points, RPC
- Caching of blocks of pixels
- Optimized reprojection engine
- Algorithms: rasterization, vectorization (polygon and contour generation), null pixel interpolation, filters

# Raster formats

- Images: JPEG, PNG, GIF, WebP, BPG ...
- Georeferenced images: GeoTIFF, .img, NITF, ...
- Wavelets: JPEG 2000, ECW, MrSID, ...
- RDBMS: Oracle Raster, PostGIS Raster, Rasdaman
- Portable DBs: Rasterlite, MBTiles, GeoPackage
- Web Services: WMS, WCS
- Radar: CEOS, Envisat
- Elevation: DTED, USGS DEM, SRTM HGT
- Containers: HDF4, HDF5, NetCDF
- Other: Geospatial PDF
- GDAL specific: memory, VRT (virtual)

⇒ 148 Formats

# Vector features

- Feature and geometry model based on OGC/ISO Single Features model
- GEOS library for geometry operations (buffers, intersections, etc..)
- Reprojection engine
- SQL capabilities
  - OGR SQL or SQLite for all formats
  - SQL pass-through for RDBMS

# Vector formats

- GIS: Shapefile, MapInfo, ESRI Personal/File Geodatabase
- CAD: DXF, DWG, DGN (pre-V8)
- RDBMS: PostGIS, Oracle, MySQL, Ingres, MSSQL, ODBC
- Portable DBs: SQLite/Spatialite, GeoPackage
- Exchange: KML (*GDAL 1.11:reference implementation*) GML, GeoJSON
- Web Service: WFS, Fusion Tables, CartoDB, CouchDB, Cloudant, GME
- National formats: SDTS, NAS, NTF, TIGER/Line, Interlis, VFK, Edigeo, SOSI, SXF, MTK GML, RUIAN GML, INSPIRE Cadastral GML
- Non spatial : CSV, XLS, XLSX, ODS
- GDAL specific: in-memory, VRT (virtual)

CERTIFIED OGC® COMPLIANT RI KML 2.2.0

⇒ 87 formats

# Community activity

- 58 developers with SVN write access
  - 19 active in 12 last months + 65 occasional contributors
  - https://www.openhub.net/p/gdal
- 2237 subscribers to gdal-dev. 2538 messages / 12 last months
- ~ 550 tickets created / 12 last months (6500 total). ~600 opened
- 1 student for GSoC in 2015. 2 in 2016

# GDAL/OGR 2.0 in a nutshell

- V2.0.0: june 2015
- 10 RFCs implemented in 2.0 cycle, including:
  - Unification of GDAL and OGR at driver and dataset level
  - Bilinear, cubic resampling kernel available for overviews or pixel operations
  - Curve geometries
  - 64 bit integer for OGR features
- 11 new drivers including:
  - GeoPackage Raster
  - Full support for GeoPackage Vector

# GDAL/OGR 2.1

- Released in May 2016. 2.1.1 release in July
- 4461 “commits” (total since 1998: 30689)
- 6 RFCs implemented in 2.1 cycle
- 7 new raster drivers
    - CALS Type 1 : read-only. Legacy archiving format
    - IBM DB2 : read/write. Tiled rasters
    - ISCE : read-only. Used by JPL in their Interferometric SAR Scientific Computing Environment
    - MRF : Meta Raster Format. read/write. Developed par NASA Global Imagery Browse Services.
    - SAFE: read-only. ESA Sentinel-1 (SAR) products.
    - SENTINEL2: read-only. ESA Sentinel-2 L1B/L1C/L2A products.
    - WMTS : read-only. Client for OGC Web Map Tile Service

# GDAL/OGR 2.1

- 5 new vector drivers
    - AmigoCloud: read/write. “geospatial as a service” platform
    - IBM DB2: read/write
    - MongoDB : read/write. “no-SQL” DB with spatial capabilities
    - netCDF: read/write. Points and vertical profiles, following CF (Climate and Forecast) conventions
    - VDV: read/write. Model and data format for public transportation (GTFS competitor). + specialization for Austrian ITF format.

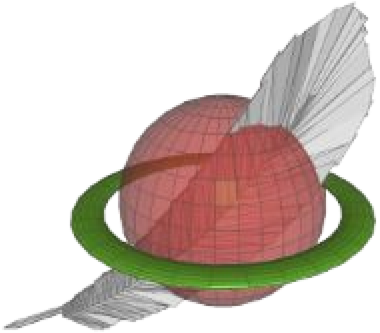
# GDAL/OGR 2.1

- Improvements in existing drivers:
  - CSV: support for full editing (update & deletion of records, through full rewriting of the file)
  - GeoJSON: full editing+ extensions to core GeoJSON preserved
  - ElasticSearch: reading mode added. Support for all geometry types in reading/writing
  - MBTiles: write support added
  - PDF: addition of the new PDFium backed (BSD licensed)
  - PLScenes:  API V1 of Planet Labs added (scene catalog)
  - VRT(raster): on-the-fly pansharpening
  - GeoTIFF: multi-threaded compression available as an option (useful for DEFLATE)

# GNM: the problem

pgRouting

graphhopper



ArcGIS Network

OSRM

OPEN SOURCE ROUTING MACHINE

# GNM: the solution

Set of special network classes in GDAL

Main purposes of these classes:

1. Abstraction for network data
2. Provide network functionality for those formats that lack it

The main work was done during GSoC 2014 by Mikhail Gusev

RFC 48: Geographical networks support was adopted and implemented in GDAL 2.1.

Documentation at http://gdal.org – GNM Architecture and GNM Tutorial sections

# GNM: GNMNetwork & GNMGenericNetwork

GNMNetwork methods:

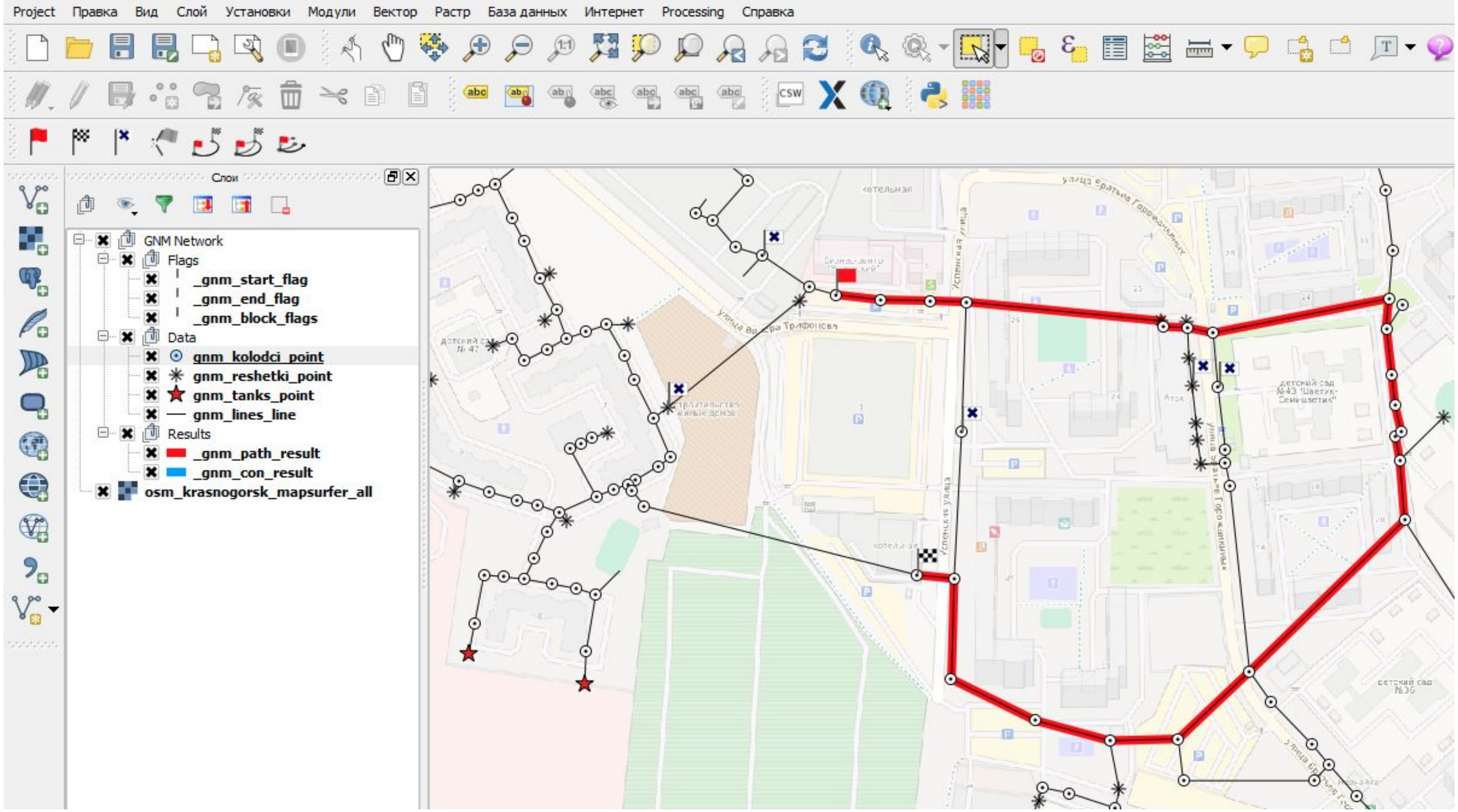
1. Create
2. Open
3. Delete
4. GetName
5. GetVersion
6. DisconnectAll
7. GetFeatureByGlobalFID
8. GetPath

GNMGenericNetwork features:

- The most general type of graph is used.
- Any feature in any layer may be vertex or edge. Virtual edges.
- All features in all layers in network have unique id.
- Special rules to guide network graph creation.
- Use common spatial reference.
- Network is always created empty.
- Deleting a network deletes all layers.

# GNM: use case

## QNetwork plugin


Project Правка Вид Слой Установки Модули Вектор Растр База данных Интернет Processing Справка
Слои
улица Братьев Горожанкиных
котельная
Успенская улица
улица Виктора Трифонова
детский сад № 47
детский сад №43 "Цветик-Семицветик"
детский сад №36

- ☒ GNM Network
  - ☒ Flags
    - ☒ _gnm_start_flag
    - ☒ _gnm_end_flag
    - ☒ _gnm_block_flags
  - ☒ Data
    - ☒ ⊙ gnm_kolodci_point
    - ☒ ✳ gnm_reshetki_point
    - ☒ ★ gnm_tanks_point
    - ☒ — gnm_lines_line
  - ☒ Results
    - ☒ 🟥 _gnm_path_result
    - ☒ 🟦 _gnm_con_result
- ☒ osm_krasnogorsk_mapsurfer_all

http://plugins.qgis.org/plugins/qnetwork/

# RFC 59.1: Utilities available as library functions

- No more :
  - os.system(“gdal_translate src.tif dst.tif”)
- Now :
  - gdal.Translate(‘dst.tif’, ‘src.tif’)
  - gdal.Translate(‘dst.tif’, src_dataset)
  - gdal.Translate(‘dest_mem’, src_dataset, format = ‘MEM’)
  - gdal.Translate(dst, src, callback = my_progress_func)
- Available for gdal_translate, gdalwarp, gdalinfo, ogr2ogr, gdaldem, gdalbuildvrt, nearblack, gdalgrid, gdal_rasterize
- Available in C/C++, Python, Java, Perl, (C#)
- Selling points :
  - Work on in-memory datasets. Useful for processing chaining without requiring on disk serialization
  - Progression report and cancellation
  - No more headaches with the path of the utilities

# Other RFCs:

- RFC 26: use of hash tables to index cached blocks.
  - Needed for huge datasets (up to 2 billion x 2 billion pixels), for WMS/WMTS
- RFC 58: DeleteNoDataValue()
  - Deletion of null value metadata into an existing dataset
  - Implemented for GeoTIFF, MEM, VRT, and Persitant Auxiliary Metadata (.aux.xml)
- RFC 60: Improvement in data preservation when converting between same OGR formats
  - Concept of “native data” attached to a feature
  - Implemented for GeoJSON
- RFC 61: Management of measured geometries (M dimension)
  - For ex: POINT M (1 2 3), POINT ZM (1 2 3 4)
  - Shapefile, PostgreSQL/PostGIS, PGDump, MEM, SQLite, GeoPackage, FileGDB, OpenFileGDB, CSV, VRT

# Other changes

- Upgrade to V8.8 of EPSG database
- Linear interpolation algorithm (based on libqhull) in gdal_grid (super fast and “beautiful” results)
- New virtual file systems: /vsis3/, /vsis3_streaming/, /vsicrypt/
- Bash completion scripts for command line utilities
- Internal changes:
    - Huge code cleaning effort,
    - No warning compilation,
    - Fixing warnings of static code analyzers
    - Fixing of numerous security vulnerabilities in dozens of drivers when dealing with corrupted/hostile files.

# GSoC 2016

- DWG driver
  - Based on a new libopencad library (licence X/MIT)
  - Aims at DWG R2000, R13/R14 compatibility
  - https://trac.osgeo.org/gdal/wiki/DWG_driver

- Geometry model fully supporting ISO SQL/MM Part 3
  - Addition of Triangle, TIN (Triangulated Irregular Networks), PolyhedralSurface geometry types
  - Implemented in Shapefile, PostGIS, GML and DXF
  - Used SFCGAL for 3D operations

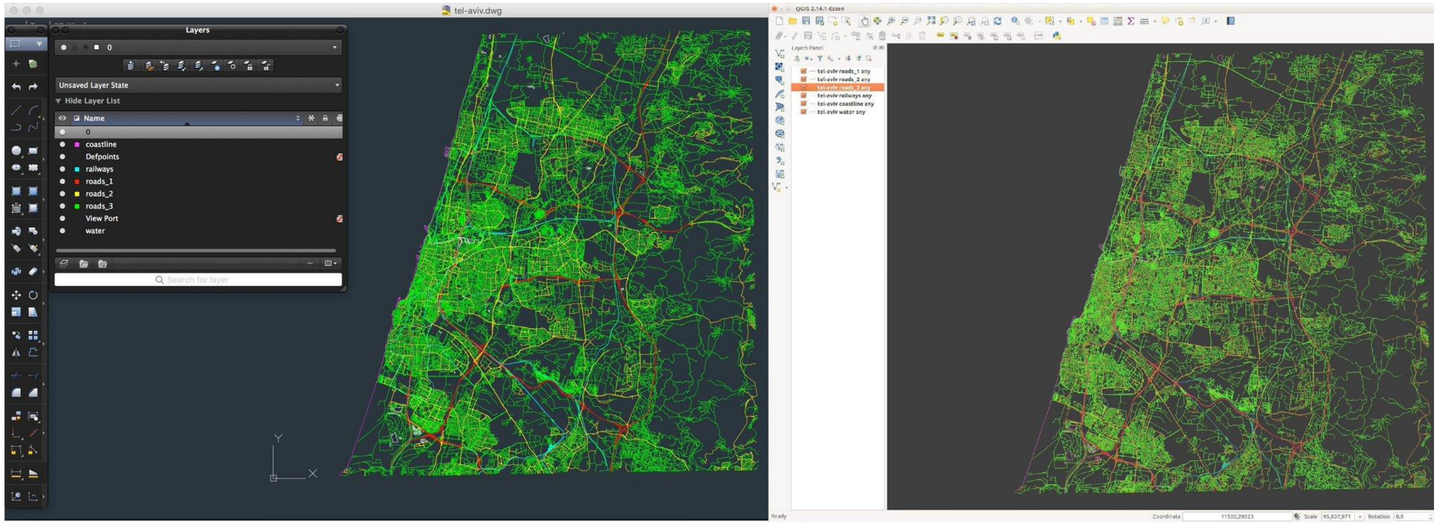
# GSoC 2016: CAD driver

OpenCAD features:

- Special reader classes.
- There are 3 open modes - ALL, FAST and FASTEST.
- When parsing the file, library does not store any information it does not need.
- Currently library support most (but not all) objects of CAD files.

GDAL CAD Driver features:

- OpenOptions - open mode and some behaviour.
- All CAD header mapped to the GDALDataset metadata.
- Create OGRLayer for each DWG Layer with objects.
- Raster subdatasets are also supported.
- Spatial reference extracted from DWG metadata or from prj file.
- CAD block reference attributes mapped to OGRLayer fields.
- Predefined fields - CAD feature type, color in hex, text, etc.
- Override reader class using GDAL VSI functions.



# GDAL 2.2 preview

- Management of GML Complex Features application schemas (INSPIRE dataset): on-going work
- Derived subdatasets to easily compute derived quantities for dataset with complex (real+imaginary) data: amplitude, intensity, log-amplitude, phase, …
- Performance improvements in raster statistics computations
- RFC 63: Sparse dataset improvements

# Potential future directions

- CMake build system
  - Unified build systems for Unix&Windows
  - Out-of-tree builds, correct header dependency
  - Effort lead by Dmitriy Baryshnikov: https://github.com/nextgis-borsch/lib_gdal
  - Tomorrow presentation: "Borsch: modern build system for C/C++ GIS projects" (10h30, room Tunnel)

- Per-dataset raster block cache:
  - For lock-less multi-threaded use
  - To solve multi-threaded dataset writing
  - https://trac.osgeo.org/gdal/wiki/rfc47_dataset_caching (Blake Thompson)

# Potential future directions

- GNM
  - Add more network drivers (pgRouting, OSRM, …)
  - Conversions between network formats (PGRouting, Spatialite, …)
  - Robustness work

- Planar topology:
  - New abstraction based on related ISO SQL/MM Part3 modeling
  - Topology primitives: nodes, edges, faces
  - TopoGeometry build on primitives / hierarchical TopoGeometry
  - Building of topology from geometries
  - Geometry ←→ TopoGeometry conversions
  - Interface with PostGIS, GRASS, Oracle, GML, Spatialite, TopoJSON
  - Conversion : topo2topo

# Potential future directions

- Raster map algebra
- OpenFileGDB write support
- OpenFileGDB raster read support
- GeoJSON: support for the IETF RFC 7946 revision. (see Sean Gillies’s talk “GeoJSON and the IETF”, tomorrow 12h00, room Tunnel)
- GeoJSON driver compatible of arbitrary large files on reading
- Improvement in spatial reference system management: guessing of EPSG codes, proposing appropriate datum shifts according to location, …
- CRS WKT 2 / ISO 19162 standard management
- Alternative geometry engine : Boost::Geometry
- New drivers, performance improvements, ...

# Questions?

Links:

- http://www.gdal.org/
- https://trac.osgeo.org/gdal/wiki/RfcList

Contacts: even.rouault@spatialys.com
bishop.dev@gmail.com (Dmitry Baryshnikov)
ari.jolma@gmail.com