

COLLECTION « CHAÎNE DE LA DONNÉE GÉOSPATIALE » — VOLUME 1

LIVRE BLANC TECHNIQUE — N°1

Panorama complet des formats géospatiaux (1994–2026)

De Shapefile à MLT — Histoire, architecture et guide décisionnel

Ce livre blanc couvre l'évolution complète des formats géospatiaux sur trente ans : du SIG desktop des années 1990 aux formats cloud-native et WebGPU de 2026. Il propose une taxonomie détaillée, une analyse par ère technologique, un guide décisionnel complet avec matrice d'usage, arbre de décision et cinq architectures de référence.

LB1 — Les Formats ←

›

LB2 — Le Stockage

›

LB3 — L'Exposition

Mattieu Pottier

Consultant en transformation digitale — SIG, géomatique, Web, ERP Odoo
MP-i — Mattieu Pottier Indépendant · mp-i.pro

Version 3.0 — Mars 2026
Document distribué librement

Derrière chaque carte web, chaque requête spatiale, chaque export SIG, il y a un format de fichier ou un protocole qui conditionne la performance, la scalabilité et la maintenabilité de tout le système. Choisir le mauvais format, c'est potentiellement des temps de chargement excessifs, une scalabilité impossible et une dette technique durable. Ce premier volume de la collection « Chaîne de la donnée géospatiale » couvre trente ans d'évolution — du SIG desktop des années 1990 aux formats cloud-native et WebGPU de 2026 — avec plus de vingt formats analysés en profondeur, un guide décisionnel complet et cinq architectures de référence.

20+

formats analysés en
profondeur

4

ruptures technologiques

5

scénarios architecturaux

1994

— 2026 couvert



Collection « Chaîne de la donnée géospatiale » — Volume 1

LB1 — LE QUOI ← CE DOCUMENT — Formats Géospatiaux 2026 — Structure, histoire, critères de choix.

LB2 — LE OÙ — Stockage Géospatial — Moteurs, index, coûts, gouvernance.

LB3 — LE COMMENT — Exposition Web — Servir et consommer la donnée.

Ce document est le prérequis de LB2 et LB3. Les formats décrits ici sont référencés dans les volumes suivants sans être redécrits.

SOMMAIRE

→ Préface

→ Introduction — Les formats géospatiaux, sujet stratégique

· Le fil conducteur : quatre ruptures technologiques

→ Partie 1 — Taxonomie et grille de lecture

· Vecteur, Raster, Nuage de points, 3D, Scientifique

· Fichier, API, Tile · Les cinq critères de choix

→ Partie 2 — Ère 1 : Le SIG Desktop (1994–2004)

· GeoTIFF · Shapefile · WMS · GML · KML · WFS · GPX

→ Partie 3 — Ère 2 : L'explosion du web cartographique (2005–2013)

· GeoJSON · MBTiles · WMTS · OSM PBF · TopoJSON · XYZ

→ Partie 4 — Ère 3 : La performance vectorielle web (2014–2019)

· MVT · GeoPackage · FlatGeobuf · OGC API Features · PMTiles

→ Partie 5 — Ère 4 : Cloud, Big Data et WebGPU (2020–2026)

· COG · GeoParquet · GeoArrow · MLT — analyse approfondie

→ Partie 6 — Formats 3D et LiDAR

· LAS/LAZ · 3D Tiles · CityGML · IFC · Convergence BIM-GIS

→ Partie 7 — Formats scientifiques et climatiques

· NetCDF · HDF5

→ Partie 8 — Formats CAD dans l'écosystème géospatial

· DXF/DWG · Flux CAD→SIG · Intégration Odo

→ Partie 9 — Guide décisionnel complet

· Matrice d'usage · Arbre de décision · Compatibilité logiciels

· Cinq scénarios architecturaux types

→ Partie 10 — Architectures de référence 2026

→ Partie 11 — Tendances 2030

→ Conclusion · Références · Glossaire · Auteur

Préface

Les formats de données géospatiales constituent l'un des sujets les plus stratégiques — et les plus sous-estimés — de l'écosystème des systèmes d'information. Depuis près de trente ans, j'ai vu des projets échouer non pas par manque de compétences métier, mais parce que les choix technologiques fondamentaux avaient été faits trop vite, sans recul sur les implications architecturales, de performance ou de maintenabilité.

Ce livre blanc est né d'un double constat. D'abord, la documentation existante sur les formats géospatiaux est souvent fragmentée : les développeurs trouvent des références techniques précises sur chaque format pris isolément, mais rarement une vision d'ensemble qui permette de comprendre pourquoi un format a émergé, quel problème concret il résolvait, et comment il s'inscrit dans la trajectoire longue de l'évolution du SIG vers le web, puis vers le cloud et le GPU.

Ensuite, 2026 marque un tournant concret : l'arrivée de MapLibre Tile (MLT) comme successeur du MVT, l'adoption massive de PMTiles qui supprime le besoin de tile server, et la montée en puissance de GeoParquet dans les pipelines de data engineering spatial. Ce n'est pas une simple évolution incrémentale — c'est une recomposition des fondamentaux.

Ce document s'adresse aux développeurs web et SIG, aux architectes solutions, aux chefs de projet en transformation digitale et à tous ceux qui prennent des décisions techniques impliquant de la donnée géographique. Il ne présuppose pas une expertise préalable en cartographie, mais suppose une familiarité avec les concepts d'API, de base de données et d'architecture web.

Mattieu Pottier — Consultant en transformation digitale, spécialiste Odoo ERP et applications SIG

Introduction — Pourquoi les formats géospatiaux sont un sujet stratégique

La donnée géographique est partout. Dans votre application de livraison quand elle calcule un itinéraire, dans votre plateforme de gestion de patrimoine quand elle localise un équipement, dans votre dashboard de monitoring quand il représente une carte de chaleur. Derrière chaque carte web, derrière chaque requête spatiale, derrière chaque export SIG, il y a un format de fichier ou un protocole qui conditionne la performance, la scalabilité et la maintenabilité de tout le système.



Choisir le mauvais format, c'est potentiellement...

Des performances dégradées — un GeoJSON de 500 Mo servi à chaque requête là où des tuiles vectorielles auraient suffi.

Une scalabilité impossible — un WFS XML exposant 100 000 entités sans pagination ni optimisation.

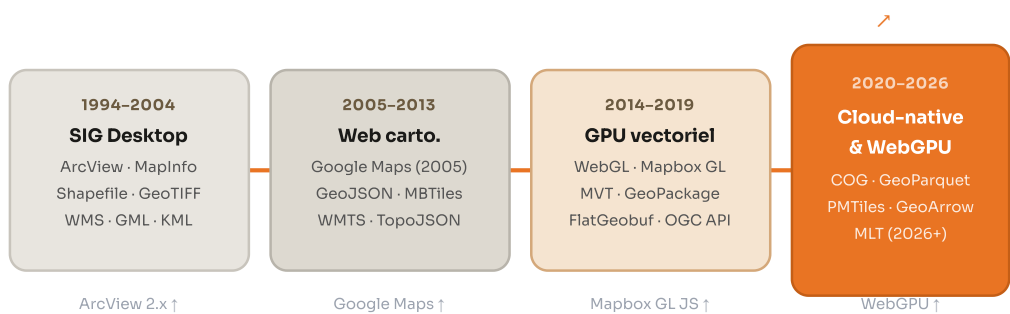
Une dette technique — un Shapefile au cœur d'une API web moderne, avec ses contraintes des années 1990.

Un coût d'infrastructure — un tile server complet là où PMTiles sur CDN aurait coûté dix fois moins.

Le fil conducteur : quatre ruptures technologiques

L'histoire des formats géospatiaux n'est pas une accumulation linéaire de standards. Elle se structure autour de quatre ruptures technologiques majeures, chacune provoquée par un changement de contexte d'usage.

FIGURE 1 — LES QUATRE RUPTURES TECHNOLOGIQUES DES FORMATS GÉOSPATIAUX
(1994–2026)



LES QUATRE RUPTURES TECHNOLOGIQUES DES FORMATS GÉOSPATIAUX

→ **Comment lire ce document**

Lecture linéaire — pour comprendre l'évolution historique et technique, section par section.

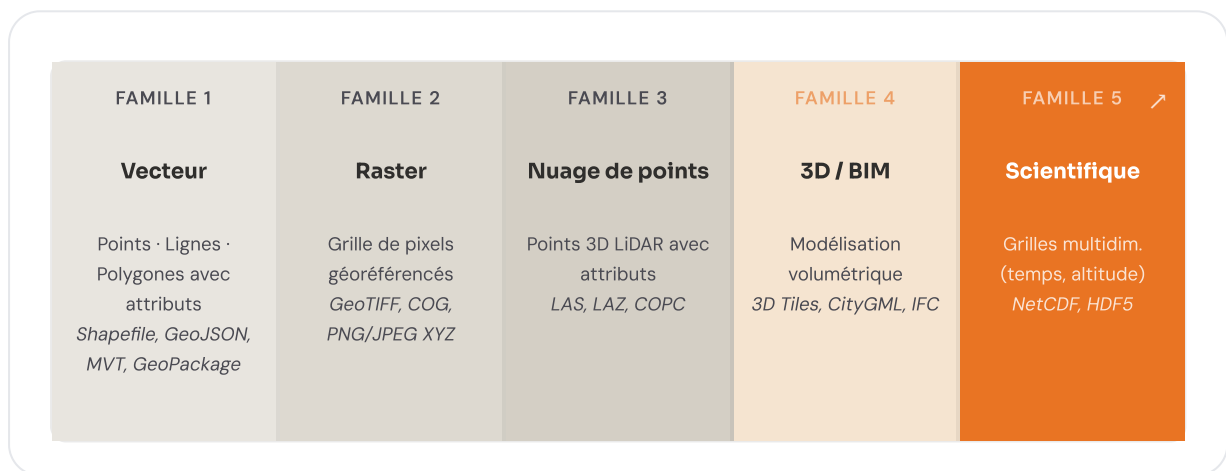
Lecture orientée décision — directement vers la Partie 9 (Guide décisionnel) et la Partie 10 (Architectures de référence) si vous avez un choix technique à faire aujourd'hui.

Partie 1 — Taxonomie et grille de lecture

Avant d'entrer dans le détail de chaque format, il est essentiel de disposer d'une grille de lecture commune. Les formats géospatiaux peuvent être classifiés selon plusieurs dimensions orthogonales.

1.1 Vecteur, Raster, Nuage de points, 3D, Scientifique

La première distinction fondamentale concerne la nature des données représentées.



1.2 Fichier, API, Tile

La deuxième dimension concerne le mode d'accès et de distribution de la donnée.

MODE D'ACCÈS	DESCRIPTION	EXEMPLES	CAS D'USAGE
Format fichier statique	Encapsulé en fichiers. Lecture séquentielle ou indexée.	Shapefile, GeoJSON, GeoPackage, GeoTIFF	Échanges, exports, stockage local
Format d'API	Exposé via protocole requête/réponse. Permet filtrage, pagination, requêtes spatiales.	WFS, OGC API Features, REST JSON	Applications interactives, interopérabilité
Format de tuiles	Données découpées en tuiles par niveau de zoom. Optimisé pour le rendu.	XYZ raster, MVT, MLT, PMTiles	Cartographie web performante, grand public



Point clé — le format n'est pas une propriété intrinsèque

Un même jeu de données peut exister sous plusieurs formes : une couche PostGIS peut être exportée en GeoJSON (fichier), exposée via OGC API Features (API), et rendue en MVT (tiles). Le format est un choix architectural en fonction du cas d'usage — pas une propriété de la donnée elle-même.

1.3 Environnement de consommation cible

ENVIRONNEMENT	CARACTÉRISTIQUES	FORMATS ADAPTÉS
Bureau / Desktop	QGIS, ArcGIS — accès fichier local, pas de contrainte réseau	Shapefile, GeoPackage, GeoTIFF, LAS/LAZ
Web navigateur	JavaScript, WebGL/WebGPU — contrainte réseau, interactivité	GeoJSON, MVT, MLT, PMTiles, COG
Cloud / Data	Traitement massif, pipelines analytiques, stockage objet	GeoParquet, GeoArrow, COG, FlatGeobuf
API / Backend	Échange entre services, interopérabilité, standardisation	OGC API Features, REST JSON, WFS, WMS

1.4 Les cinq critères de choix d'un format

Tout choix de format géospatial devrait être évalué selon ces cinq dimensions.



Volumétrie

Combien d'entités, quelle taille de fichier, quel volume de requêtes ? Un GeoJSON convient pour 500 POI, devient problématique au-delà de 50 000.



Interactivité

La carte doit-elle permettre le clic, le survol, le filtre client ? Les formats raster (WMS, XYZ) ne permettent pas d'interaction sur les entités.



Scalabilité

Le service doit-il tenir sous charge ? Les tuiles (MVT, PMTiles) sont naturellement cachables. Les API WFS XML ne le sont pas.



Interopérabilité

Doit-on s'intégrer à des standards OGC, des logiciels bureau, des outils tiers ? Certains formats sont ouverts, d'autres propriétaires ou émergents.



Maturité & écosystème

Existe-t-il des bibliothèques stables, une documentation solide, une communauté active ? MLT offre de meilleures performances mais un écosystème encore restreint en 2026.

Partie 2 — Ère 1 : Le SIG Desktop (1994–2004)

Au milieu des années 1990, le SIG est un outil professionnel réservé aux spécialistes. Les logiciels — ArcView 3.x d'ESRI, MapInfo Professional, Intergraph — tournent sur des postes de travail coûteux. L'internet grand public n'existe pas encore sous sa forme actuelle. Les formats de données géospatiales de cette époque reflètent ces contraintes : accès fichier local, échange entre spécialistes, peu de préoccupation pour le réseau ou l'interopérabilité avec des navigateurs. C'est dans ce contexte que naissent les formats fondateurs, dont certains sont encore actifs trente ans plus tard.

1994 — GEOTIFF

Niles Ritter (JPL/NASA) / Mike Ruth (SPOT Image)

Le raster géoréférencé universel. Étend TIFF avec des balises normalisées stockant projection, emprise et résolution directement dans le fichier image. Standard de fait depuis 1995, adopté OGC en 2019.

~1993 / SPEC 1998 — SHAPEFILE

ESRI

Le format qui refusait de mourir. Trente ans de domination malgré noms de colonnes à 10 chars, limite 2 Go, encodage ambigu. Toujours omniprésent dans les portails open data institutionnels.

2000 — WMS

OGC — Web Map Service

Premier standard de cartographie web. Le client envoie un bounding box, le serveur renvoie une image PNG/JPEG. Traitement entièrement côté serveur. Toujours actif dans les portails institutionnels en 2026.

2000 — GML

OGC — Geography Markup Language

L'ambition XML universelle. Verbeux (5 à 15× GeoJSON pour les mêmes données), il reste le format sous-jacent de WFS et les échanges INSPIRE. GeoJSON l'a supplanté pour les API modernes.

2001 — KML

Keyhole Inc. / Google

Créé pour EarthViewer (Google Earth), popularise la carte 3D grand public. XML avec styles visuels intégrés. Standardisé OGC en 2008. Niche stable pour le partage ponctuel d'annotations géographiques.

2002 — WFS

OGC — Web Feature Service

Accès aux données vectorielles brutes via le web. Hérite de GML — réponses volumineuses, parsing côté client coûteux, pas de pagination native en v1.x. OGC API Features (2019) est son successeur officiel.

2002 — GPX

Topografix — GPS Exchange Format

Standard d'échange universel pour les données GPS terrain. Traces, waypoints, routes avec timestamp et altitude. Support natif Garmin, Strava, Komoot, QGIS. Niche stable, non concerné par les tendances web.

ÈRE 1 — SIG DESKTOP · 1994 · RASTER

GeoTIFF

Le raster géoréférencé universel — le format qui a normalisé la donnée satellite

Niles Ritter (JPL) · Mike Ruth (SPOT Image)

Spec 1.0 — 1995

Standard OGC — 2019

Statut 2026 : Actif

Contexte et problématique

En 1994, les données satellitaires et aériennes numérisées se multiplient, mais chaque fournisseur utilise ses propres conventions pour stocker l'information de géoréférencement. Résultat : une image SPOT ne pouvait pas être superposée directement à une image d'un autre capteur sans retraitement manuel. GeoTIFF répond à ce problème en étendant le format TIFF standard avec des balises normalisées qui stockent le système de coordonnées, la transformation affine et la projection cartographique directement dans le fichier image.

Architecture technique

GeoTIFF est un TIFF enrichi de métadonnées géographiques sous forme de tags privés. L'image est stockée en tuiles ou en bandes, potentiellement compressée (LZW, Deflate, JPEG). Les métadonnées clés : **GeoKeyDirectory** (répertoire des clés de géoréférencement), **ModelPixelScaleTag** (résolution X/Y dans les unités de la projection), **ModelTiepointTag** (correspondance pixel ↔ coordonnée terrain), **ModelTransformationTag** (matrice affine de transformation).

FORCES	LIMITES
Support universel (GDAL, QGIS, ArcGIS, Python rasterio)	Fichier complet à charger — pas de streaming natif
Métadonnées intégrées dans le fichier	Taille importante pour haute résolution
Support multi-canal (RGB, infrarouge, hyper-spectral)	Non adapté aux accès partiels côté web
Standard de fait depuis 1995	COG (2016) résout le problème du streaming



Statut 2026

GeoTIFF reste le format raster de référence pour le stockage et l'échange. Sa version cloud-optimisée (COG) est devenue le standard pour le streaming raster sur infrastructure cloud. Il est peu probable qu'il disparaisse avant longtemps.

Shapefile

Le format qui refusait de mourir — trente ans de domination malgré ses contraintes

ESRI

Multi-fichiers : .shp · .shx · .dbf · .prj · .cpg

Statut 2026 : Legacy ⚠

Architecture technique

Un "fichier Shapefile" est en réalité un ensemble d'au moins trois fichiers avec le même nom de base : `.shp` (géométries vectorielles), `.shx` (index d'accès aux géométries), `.dbf` (attributs en format dBASE III — héritage des années 1980), `.prj` (projection cartographique), `.cpg` (encodage des caractères).



Contraintes structurelles — pourquoi le Shapefile est obsolète

Noms de colonnes limités à 10 caractères — héritage dBASE III 1985, source de troncatures constantes.

Taille maximale de 2 Go par fichier — insuffisant pour les jeux de données modernes.

Pas de support NULL — une valeur nulle est représentée par 0, ambiguïté garantie.

Un seul type de géométrie par fichier — impossible de mélanger points et polygones.

Encodage des caractères ambigu — UTF-8 ou Latin-1 selon les logiciels, source de corruptions.

Pas de topologie — les relations entre entités doivent être recalculées.

→ **Anecdote technique**

La spécification du fichier .dbf (dBASE III Plus) date de 1985. Quand ESRI a créé le Shapefile au début des années 1990, ils ont réutilisé ce format de moins de dix ans pour les attributs — un choix pragmatique à l'époque. En 2026, nous continuons à utiliser des noms de colonnes de 10 caractères hérités de contraintes matérielles des années 1980.

Pourquoi le Shapefile survit malgré ses limitations : support universel (absolument tous les logiciels SIG le lisent), simplicité d'usage (pas de base de données requise), et inertie institutionnelle (décennies de workflows, données archivées et scripts automatisés).



Statut 2026

Le Shapefile est officiellement déconseillé pour tout nouveau projet. **GeoPackage** est son successeur naturel pour les échanges desktop. Il reste cependant omniprésent comme format d'export dans les outils institutionnels et les portails open data.

WMS — Web Map Service

La première cartographie web normalisée — des images à la demande

OGC

GetCapabilities · GetMap · GetFeatureInfo

Statut 2026 : Actif institutionnel

WMS répond à un problème fondamental du web cartographique naissant : comment permettre à un client quelconque de demander une image cartographique d'une zone précise, à une échelle précise, sans télécharger toute la donnée ? Le serveur génère et renvoie une image PNG ou JPEG pour chaque requête. Le traitement est entièrement côté serveur.

FORCES	LIMITES
Standard OGC universellement supporté	CPU serveur sollicité à chaque requête
Rendu serveur — cohérence graphique garantie	Image non interactive (pas de clic sur entité)
Compatible tous clients cartographiques	Difficile à mettre en cache (paramètres variables)



Statut 2026

WMS reste actif dans les contextes institutionnels et les portails de données publiques (IGN, Copernicus, etc.). Pour les applications web modernes à fort trafic, il a été largement supplanté par les tuiles raster XYZ cachables et les tuiles vectorielles MVT/MLT.

GML — Geography Markup Language

L'ambition XML qui a manqué le virage du web

OGC

Verbosité XML : 5–15× GeoJSON

Statut 2026 : Legacy INSPIRE ⚠

En 2000, XML est au sommet de sa popularité. L'OGC publie GML comme format d'échange universel pour les données géographiques complexes. GML est pensé comme le langage universel de l'information géographique — et c'est précisément son problème : la verbosité XML.

GML VS GEOJSON — COMPARAISON

```
<!-- GML — un point avec srsName -->
<gml:Point srsName="urn:ogc:def:crs:EPSG::4326">
  <gml:pos>48.8566 2.3522</gml:pos>
</gml:Point>

// GeoJSON équivalent
{"type": "Point", "coordinates": [2.3522, 48.8566]}

// Rapport de taille : GML est typiquement 5 à 15× plus volumineux.
```



**Statut
2026 :**

GML est utilisé dans les échanges INSPIRE et portails institutionnels. Il reste le format sous-jacent de WFS 1.x et 2.x. Pour les nouvelles API, GeoJSON l'a remplacé dans OGC API Features.

ÈRE 1 — SIG DESKTOP · 2001 · VISUALISATION GÉOGRAPHIQUE

KML — Keyhole Markup Language

Google Earth change la donne — la cartographie grand public

Keyhole Inc. / Google

Standard OGC — 2008

KMZ : archive ZIP

Statut 2026 : Niche stable

KML est créé en 2001 pour EarthViewer (qui deviendra Google Earth). Il popularise la carte 3D interactive auprès du grand public et introduit massivement la notion de couche cartographique partageable. Conçu pour la présentation visuelle avec styles, icônes et animations. Sa forme compressée KMZ facilite la distribution.

FORCES	LIMITES
Grand public — simple à créer et partager	XML verbeux, peu performant pour gros volumes
Support natif Google Earth / Maps	Pas adapté aux API web modernes
Style visuel intégré (couleurs, icônes)	Peu de support pour les attributs complexes

ÈRE 1 — SIG DESKTOP · 2002 · SERVICE VECTORIEL

WFS — Web Feature Service

Accès vectoriel OGC — les limites structurelles de XML

OGC

WFS-T : transactions

Successeur : OGC API Features

Statut 2026 : Legacy institutionnel ⚠

WFS répond à la question : comment accéder aux données vectorielles brutes (géométries + attributs) via le web, de manière standardisée ? Il hérite de GML pour la représentation — chaque feature retournée est encodée en XML GML. Les conséquences sont directes : requêtes volumineuses, parsing côté client coûteux, pas de pagination native en WFS 1.x, complexité du filtre OGC.



**Statut
2026 :**

WFS reste présent dans les portails institutionnels existants. OGC API Features (2019) est son successeur officiel avec une API REST JSON moderne qui résout la plupart de ses problèmes structurels.

ÈRE 1 — SIG DESKTOP · 2002 · GPS TERRAIN

GPX — GPS Exchange Format

Le standard terrain GPS — simple, robuste, universel

Topografix

Traces · Waypoints · Routes

Timestamp + altitude natifs

Statut 2026 : Niche stable 

GPX émerge avec la démocratisation des récepteurs GPS portables (Garmin, Magellan).
Format d'échange universel XML pour les données de navigation GPS : traces avec timestamp et altitude, waypoints, routes. Support natif sur Garmin, Strava, Komoot, QGIS. Sa niche est stable — il n'est pas concerné par les tendances web.

Partie 3 — Ère 2 : L'explosion du web cartographique (2005–2013)

Février 2005 marque un tournant historique : Google lance Google Maps sur le web. Pour la première fois, une carte interactive glissante (*slippy map*) est disponible dans un simple navigateur, sans plugin. La technique des tuiles raster combinée à du JavaScript asynchrone révolutionne l'expérience cartographique. L'écosystème web cartographique explose : OpenLayers (2006) puis Leaflet (2011) démocratisent la cartographie JavaScript open source. Ces nouveaux usagers ont besoin de formats adaptés au web : légers, JSON-natifs, faciles à manipuler.

GeoJSON

JSON natif web — le tournant qui a tout changé

Howard Butler · Sean Gillies et al.

RFC 7946 — 2016

WGS84 obligatoire

Statut 2026 : Standard API web

Structure et architecture

GEOJSON — FEATURECOLLECTION

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [2.3522, 48.8566]
      },
      "properties": {
        "name": "Paris",
        "population": 2161000
      }
    }
  ]
}
```

// Types géométriques : Point, MultiPoint, LineString, MultiLineString,
// Polygon, MultiPolygon, GeometryCollection
// RFC 7946 : WGS84 obligatoire, polygones en sens antihoraire

Pourquoi GeoJSON a gagné

1

Zero parsing

- ✓ `JSON.parse()` intégré dans tous les moteurs JS depuis ES5
- ✓ Pas de librairie externe requise

2

Lisible humainement

- ✓ Inspectable dans n'importe quel éditeur de texte
- ✓ Débogage trivial

3

Intégration native

- ✓ Leaflet, Mapbox GL, OpenLayers — natif
- ✓ PostGIS, GDAL, Python fiona/geopandas

4

Tooling riche

- ✓ geojson.io, Turf.js, GDAL, ogr2ogr
- ✓ Écosystème mature et stable

Les limites de GeoJSON

! **Volumétrie** — au-delà de 50 000 entités, les performances de parsing et de rendu se dégradent significativement

! **Pas de compression native** — un GeoJSON de données OSM d'une ville peut peser plusieurs centaines de Mo

! **Pas d'index spatial** — interroger un GeoJSON côté serveur nécessite de le charger entièrement en mémoire

! **WGS84 obligatoire** — RFC 7946 impose ce CRS, incompatible avec certains contextes (Lambert 93, etc.)



Statut 2026

GeoJSON est le format vectoriel dominant pour les API web, les échanges de données légères et les applications avec moins de quelques dizaines de milliers d'entités. Pour les gros volumes, MVT/MLT lui ont pris sa place dans le rendu cartographique.

WMTS — Web Map Tile Service

WMS tuilé — le pont entre le serveur et le CDN

OGC

Grille TileMatrix normalisée

Successeur : OGC API Tiles (2022)

Statut 2026 : Actif institutionnel

WMS génère une image à la demande pour chaque requête — ce qui est coûteux en CPU et impossible à cacher efficacement. WMTS standardise la technique des tuiles : on pré-calcule des tuiles 256×256 pour chaque niveau de zoom et chaque position dans une grille fixe. Ces tuiles sont identiques pour tous les utilisateurs et parfaitement cachables sur CDN.

FORCES	LIMITES
Tuiles cachables sur CDN — scalabilité excellente	Non interactif comme WMS (image raster)
Standard OGC — support institutionnel large	Moins flexible que WMS (grille fixe)
Compatible QGIS, ArcGIS, navigateurs	Protocole plus complexe que le schéma XYZ simple

MBTiles

SQLite pour les tuiles — le container tile standard

Mapbox / Development Seed

SQLite : table tiles (zoom, x, y, data)

Raster PNG/JPEG ou vectoriel MVT

Statut 2026 : Actif offline

Quand on génère des tuiles pour un zoom mondial, on peut facilement produire des millions de fichiers PNG individuels. Stocker des millions de petits fichiers sur un système de fichiers est inefficace. MBTiles encapsule toutes les tuiles dans une base SQLite unique, portable, indexée et directement servable via TileServer GL. PMTiles (2022) offre une alternative sans serveur via HTTP Range Requests.

OSM PBF — OpenStreetMap Protocolbuffer Binary Format

La matière première des pipelines vectoriels planet-scale

OpenStreetMap Foundation

Planétaire : ~80 Go

Parsing 5–6× plus rapide que XML OSM

Statut 2026 : Standard OSM

OSM distribue l'intégralité de sa base mondiale sous forme de dumps. Le format PBF (Protocol Buffers), adopté en 2010, divise la taille par 2–3 et multiplie la vitesse de parsing par 5–6 par rapport au XML.



Deux usages distincts du terme "PBF" — à ne pas confondre

OSM PBF (.osm.pbf) — le format de dump complet d'OpenStreetMap : nœuds, chemins, relations. C'est la donnée source brute, non structurée comme un SIG classique.

MVT/PBF — les tuiles vectorielles MVT encodées en Protocol Buffers. Ici "PBF" désigne le codec de sérialisation, pas le format OSM. La confusion est fréquente et source d'erreurs.

OUTIL	ENTRÉE	SORTIE
Tippecanoe	GeoJSON, FlatGeobuf	MVT (MBTiles ou PMTiles)
Planetiler	OSM PBF, GeoPackage, Shapefile	MVT ou MLT (PMTiles)
osmium-tool	OSM PBF	OSM PBF filtré, GeoJSON
osm2pgsql	OSM PBF	PostGIS



Sans OSM PBF, construire une carte mondiale en MVT/MLT nécessiterait >500 Go de GeoJSON. Le PBF planétaire pèse ~80 Go.

TopoJSON

Compression topologique — la niche de la dataviz

Mike Bostock (D3.js)

2–4× plus compact que GeoJSON

Statut 2026 : Niche D3.js stable

TopoJSON encode la topologie explicitement : au lieu de stocker des polygones fermés, il stocke des arcs (segments de frontière) référencés par les polygones adjacents. Une frontière commune entre deux départements n'est stockée qu'une seule fois. Pour une carte choroplèthe classique, TopoJSON est typiquement 2 à 4× plus petit que GeoJSON équivalent. Sa niche est stable : excellent pour la dataviz D3.js et les cartes choroplèthes web légères.

3.4 Le schéma XYZ — le tuilage dominant non-OGC

Parallèlement aux standards OGC, un schéma de facto s'impose par sa simplicité : le schéma XYZ, popularisé par Google Maps et OpenStreetMap. L'URL d'une tuile suit le pattern `https://tile.openstreetmap.org/{z}/{x}/{y}.png` où `z` est le niveau de zoom (0–22) et `x`, `y` la position dans la grille de Mercator. Ce schéma est directement supporté par Leaflet, Mapbox GL et tous les frameworks cartographiques modernes. Il n'a pas de spécification formelle OGC mais est devenu un standard de facto universel.

Partie 4 — Ère 3 : La performance vectorielle web (2014–2019)

En 2013–2014, la carte web est confrontée à un mur de performance. Les tuiles raster gèrent parfaitement les fonds de carte statiques, mais posent deux problèmes majeurs : elles ne sont pas interactives, et elles doivent être régénérées entièrement si le style change. WebGL, disponible dans les navigateurs depuis 2011 et stable depuis 2013, ouvre une nouvelle possibilité : utiliser le GPU du client pour le rendu cartographique. Si on envoie des données vectorielles brutes au client et qu'on les rend avec WebGL, le style peut être modifié en temps réel côté client sans régénérer les tuiles. C'est cette idée qui donne naissance aux vector tiles.

MVT — Mapbox Vector Tiles

Le standard de performance web vectoriel — binaire, GPU-ready, universel

Mapbox

Protocol Buffers (protobuf)

Grille 4096×4096 par tuile

Statut 2026 : Standard dominant

Architecture technique

Une tuile MVT est un fichier binaire encodé en Protocol Buffers. Elle contient des **layers** (couches thématiques : routes, bâtiments, eau), des **features** (entités avec attributs), des **géométries** (coordonnées entières relatives à la grille 0–4096) et des **clés/valeurs compressées** (les attributs strings sont dédupliqués via un dictionnaire interne). Le fichier est encodé row-based — limitation que MLT résoudra.

L'écosystème MVT — pourquoi il a tout conquis

CATÉGORIE	OUTILS / LIBRAIRIES
Génération serveur	Tippecanoe · Planetiler · PostGIS ST_AsMVT · Martin · pg_tileserv
Rendu client web	Mapbox GL JS · MapLibre GL JS · OpenLayers · Deck.gl
Rendu client mobile	Mapbox Maps SDK · MapLibre Native (iOS, Android)
Serveur de tuiles	TileServer GL · Martin · tegola · pg_tileserv
Stockage	MBTiles · PMTiles · S3/CDN
Bureau	QGIS — support direct des sources MVT

FORCES	LIMITES
Style modifiable côté client en temps réel	Row-based — décodage moins efficace pour gros volumes
Interactif — clic sur features, filtres clients	Pas de support 3D natif (altitude)
Cachable sur CDN comme les tuiles raster	Pas de support des types complexes (nested)
Écosystème massif et mature	Pas pensé pour WebGPU (latence de décodage)



Statut 2026

MVT est le standard dominant pour le rendu vectoriel web. Il est déployé sur toutes les grandes plateformes cartographiques (Mapbox, Google Maps, HERE, Esri). Son successeur MLT est en cours d'adoption mais MVT restera dominant pendant plusieurs années encore.

GeoPackage

Le Shapefile moderne enfin — SQLite standardisé par l'OGC

OGC

SQLite — 1 seul fichier .gpkg

Multi-couches · UTF-8 · ACID

Statut 2026 : Standard desktop 

GeoPackage est publié par l'OGC en 2014 comme remplacement officiel du Shapefile. Un GeoPackage est une base SQLite avec une structure de tables normalisée : `gpkg_contents` (métadonnées des couches), `gpkg_geometry_columns` (description des colonnes géométriques), `gpkg_spatial_ref_sys` (projections), tables métier avec géométries WKB + attributs typés, et tables optionnelles pour les tuiles raster.

CRITÈRE	SHAPEFILE	GEOPACKAGE
Nombre de fichiers	Minimum 3 (.shp, .shx, .dbf)	1 seul fichier .gpkg
Noms de colonnes	10 caractères max	Illimités
Taille max	2 Go	1,1 To (spec OGC)
Types de données	Limités dBASE III	SQL complet (INTEGER, REAL, TEXT, BLOB, DATE)
Valeurs NULL	Ambiguës (0 = null)	NULL SQL natif
Multi-couches	Un fichier par couche	Plusieurs couches dans un seul fichier
Transactions	Non	Oui (SQLite ACID)
Encodage	Ambigu (Latin-1 ou UTF-8)	UTF-8 garanti



Statut 2026

GeoPackage est le format recommandé pour l'échange et le stockage desktop. QGIS, ArcGIS Pro, FME, GDAL le supportent tous. Il n'est pas adapté au rendu web direct — c'est son rôle : remplacer Shapefile dans les workflows bureautiques.

FlatGeobuf

Streaming binaire avec index spatial — le format méconnu

Björn Harrtell et al.

FlatBuffers (Google)

Index Hilbert R-Tree intégré

HTTP Range Requests

Statut 2026 : Technique solide, adoption limitée

FlatGeobuf répond à un problème spécifique : diffuser un grand jeu de données vectorielles sur le web en permettant des requêtes spatiales sans serveur. GeoJSON nécessite de charger le fichier entier. FlatGeobuf intègre un **index Hilbert R-Tree** directement dans le fichier, permettant des requêtes bbox en n'accédant qu'aux octets pertinents via HTTP Range Requests. Structure : Header (schéma, CRS, emprise), Index spatial, Features ordonnées selon la courbe de Hilbert.

FORCES	LIMITES
Streaming spatial sans serveur (HTTP Range)	Support limité dans les navigateurs web
Index spatial intégré — requêtes $O(\log n)$	Peu de tooling grand public
Format binaire compact	Pas adapté au rendu cartographique (pas de tiles)
Support GDAL/OGR complet	Adoption freinée par écosystème client limité

OGC API Features

REST moderne — le successeur officiel de WFS

OGC

REST · JSON · OpenAPI 3.0

Pagination native · Filtrage CQL2

Statut 2026 : Standard recommandé 

OGC API Features est conçu selon les principes REST modernes : URLs lisibles (`/collections/{id}/items/{featureId}`), JSON natif (GeoJSON par défaut), pagination via liens next/prev, filtrage CQL2, documentation OpenAPI auto-générée, et rendu HTML pour la navigation humaine.

OGC API FEATURES — EXEMPLE DE REQUÊTE

```
// Lister les communes de plus de 10 000 habitants dans une emprise
GET /collections/communes/items
  ?bbox=2.0,48.5,3.0,49.0
  &filter=population>10000
  &limit=100
  &offset=0

// Accéder à une feature spécifique
GET /collections/communes/items/75056

// Implémenter dans pygeoapi (Python)
// GeoServer, ldproxy, 52°North — implémentations disponibles
```

FORCES	LIMITES
REST JSON moderne — accessible aux développeurs web	Maturité variable selon les implémentations
Pagination native	Moins de fonctionnalités transactionnelles que WFS-T
Documentation OpenAPI auto-générée	Adoption encore partielle dans les outils bureau
Standard OGC — interopérabilité garantie	Part 4 (édition) encore en développement



Statut 2026

OGC API Features est le standard recommandé pour exposer des données vectorielles via API. GeoServer, pygeoapi, Idproxy, 52°North l'implémentent. Pour un nouveau projet API géospatiale en 2026, c'est le choix de référence.

PMTiles

Single-file tile archive — CDN sans serveur

Protomaps — Brandon Liu

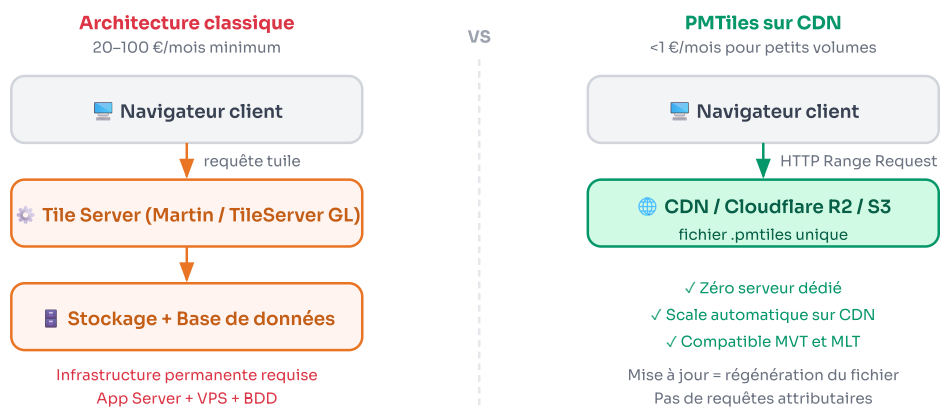
HTTP Range Requests

Compatible MVT + MLT

Statut 2026 : Adoption rapide ✓

Servir des tuiles cartographiques a longterm nécessite un serveur dédié. PMTiles supprime ce besoin : le format organise toutes les tuiles dans un fichier unique avec un index spatial en en-tête. Pour récupérer une tuile, le client envoie une requête HTTP `Range` pour télécharger uniquement les octets correspondants. Les CDN et stockages objets supportent nativement les Range Requests.

FIGURE 9 — ARCHITECTURE TILE SERVER CLASSIQUE VS PMTILES SUR CDN



ARCHITECTURE TILE SERVER CLASSIQUE VS PMTILES SUR CDN — IMPACT COÛT ET INFRASTRUCTURE



PMTiles dans l'écosystème MLT

PMTiles est format-agnostique : il peut contenir des tuiles MVT (Protocol Buffers) ou des tuiles MLT (le nouveau format de 2026). La combinaison **PMTiles + MLT** représente l'architecture serverless la plus performante disponible en 2026.



Statut 2026

PMTiles est en adoption rapide. Il est devenu le format de distribution recommandé pour les applications cartographiques légères et les projets open data. Son intégration dans Planetiler et le support natif dans MapLibre GL JS en font une option de production crédible.

Partie 5 — Ère 4 : Cloud, Big Data et WebGPU (2020–2026)

Deux tendances convergent au début des années 2020. D'un côté, le volume de données géospatiales explose : satellites Sentinel et Landsat en open access, données ADS-B, traces de smartphones, LiDAR des villes. Les formats binaires compacts orientés colonnes (Apache Parquet, Arrow) dominent l'ingénierie des données — il devient naturel de les étendre au domaine spatial. De l'autre côté, WebGPU arrive dans les navigateurs — successeur de WebGL, il ouvre des possibilités nouvelles pour le rendu cartographique si le format des données est adapté à la structure GPU.

ÈRE 4 — CLOUD-NATIVE · 2016 (ADOPTION ~2020) · RASTER CLOUD

COG — Cloud Optimized GeoTIFF

Le raster cloud-native — streaming partiel sans serveur

OSGeo / Cloud Native Geo

HTTP Range Requests

Tuiles internes + aperçus pré-calculés

Standard ESA/USGS/NASA

Statut 2026 : Standard cloud raster ✓

GeoTIFF standard présente un problème pour le cloud : pour accéder à une petite zone d'une image de 10 Go, un client doit télécharger le fichier entier. COG résout ce problème avec HTTP Range Requests — la même technique que PMTiles. Un COG est un GeoTIFF réorganisé avec deux caractéristiques clés : **tuiles internes** (256×256 ou 512×512 px, répertoire d'offsets en en-tête) et **aperçus pré-calculés** (plusieurs niveaux de résolution dans le même fichier).



Impact concret — Sentinel-2 (~500 Mo par bande)

Visualisation d'une ville (zoom 15) avec COG sur S3 : ~200 Ko téléchargés.

Vue continentale (zoom 6) : ~50 Ko (aperçu basse résolution).

Sans COG (GeoTIFF classique) : 500 Mo à télécharger pour chaque accès.

RÔLE	OUTILS
Génération	GDAL <code>gdal_translate -co TILED=YES -co COPY_SRC_OVERVIEW S=YES</code> · rio-cogeo
Stockage	AWS S3 · Google Cloud Storage · Azure Blob (Range Requests natifs)
Serveur	rio-tiler · TiTiler (FastAPI) · Terracotta
Client Python	rasterio · stackstac (time series)
Catalogage	STAC (SpatioTemporal Asset Catalog)



Statut 2026

COG est devenu le standard de facto pour la donnée raster cloud. Les agences spatiales (ESA, USGS, NASA) distribuent leurs archives en COG. **STAC + COG** forme l'architecture de référence pour les plateformes de données satellitaires.

ÈRE 4 — CLOUD-NATIVE · 2021 / SPEC 1.0 SEPT. 2023 · DATA ENGINEERING

GeoParquet

Apache Parquet pour le spatial — le Big Data géospatial

Joris Van den Bossche · Kyle Barron · Dewey Dunnington et al.

Géométries WKB dans colonne bytes

Standard Overture Maps Foundation

Statut 2026 : Adoption rapide

Apache Parquet s'est imposé comme le format standard du data engineering : orienté colonnes, compressé, optimisé pour les requêtes analytiques. GeoParquet l'étend avec une convention pour stocker des géométries (en WKB dans une colonne bytes) et des métadonnées géospatiales (CRS, emprise, type) dans les métadonnées Parquet. DuckDB, Spark, BigQuery, Pandas, Polars — tous le lisent nativement.

SQL – DUCKDB + GEOPARQUET (REQUÊTE SPATIALE)

```
-- Compter les bâtiments dans Paris intra-muros
LOAD spatial;

SELECT COUNT(*) FROM read_parquet('buildings.parquet')
WHERE ST_Within(
  geometry,
  ST_GeomFromText('POLYGON((2.224 48.815, 2.469 48.815,
                          2.469 48.902, 2.224 48.902,
                          2.224 48.815))')
);
-- Sur 100 millions de bâtiments : résultat en ~2 secondes
-- Sans infrastructure serveur, directement sur fichier local ou S3

-- Lire depuis Cloudflare R2 / S3 sans téléchargement complet
SELECT name, ST_Area(geometry) as area_m2
FROM read_parquet('s3://bucket/communes.parquet')
WHERE ST_Intersects(geometry, ST_GeomFromText('POINT(2.35 48.86)'))
LIMIT 10;
```

Architecture column-oriented — pourquoi c'est révolutionnaire

1 Compression par colonne

- ✓ Une colonne d'attributs du même type compresse 5 à 20× mieux
- ✓ Compression RLE, Dictionary, Zstd par colonne

2 Lecture partielle

- ✓ 3 colonnes sur 50 = lecture de seulement 6% des données
- ✓ HTTP Range Requests compatible (S3, R2)

3 Pushdown de prédicats

- ✓ Statistiques min/max par row group
- ✓ Blocs entiers sautés sans lecture

4 Compatibilité universelle

- ✓ DuckDB, Spark, BigQuery, Pandas, Polars
- ✓ Overture Maps distribue en GeoParquet

FORCES	LIMITES
Requêtes analytiques massives ultra-rapides	Pas adapté au rendu cartographique direct
Compressé et compact	Pas de mise à jour partielle (append ou régénération)
Écosystème data engineering universel	Index spatial moins efficace que PostGIS pour requêtes complexes
Compatible DuckDB, Spark, BigQuery, Polars	Courbe d'apprentissage data stack



Statut 2026

GeoParquet est en adoption rapide dans les pipelines data engineering spatial. Overture Maps Foundation distribue l'intégralité de ses données mondiales (bâtiments, routes, lieux) en GeoParquet. C'est devenu le format de référence pour les pipelines ETL géospatiaux.

ÈRE 4 — CLOUD-NATIVE · 2023 · MÉMOIRE GPU

GeoArrow

Apache Arrow pour le spatial — le pont entre mémoire et GPU

Kyle Barron · Joris Van den Bossche · Dewey Dunnington et al.

Zero-copy CPU→GPU

Buffers Arrow alignés mémoire

Statut 2026 : Émergent stratégique ⚡

Apache Arrow est le standard de représentation des données analytiques en mémoire : un format orienté colonnes avec un buffer mémoire commun partageable entre processus (zero-copy). GeoArrow définit une extension Arrow native pour les géométries stockées directement dans des buffers Arrow typés. L'intérêt fondamental : les coordonnées géométriques sont dans des buffers numériques qui peuvent être transférés au GPU sans conversion. Dans le contexte WebGPU, les vertex buffers GPU attendent exactement ce format.

FORCES	LIMITES
Zero-copy entre CPU et GPU	Spécification encore jeune (2023)
Interopérabilité Python (geopandas, shapely 2.0)	Adoption limitée hors du monde Python data
Performances SIMD sur opérations géométriques	Écosystème en construction



Statut 2026 :

GeoArrow est émergent mais stratégique. Il forme le lien entre GeoParquet (stockage) et les moteurs de rendu GPU (WebGPU, deck.gl). À surveiller comme brique fondamentale de la prochaine génération de visualisation géospatiale.

MLT — MapLibre Tile

Le vector tile conçu pour WebGPU — la prochaine ère commence

MapLibre / Microsoft / AWS

Column-oriented encoding

SIMD + WebGPU natif

Coordonnées 3D natives

Statut 2026 : Adoption précoce ⚡

Genèse et contexte

En 2014, MVT a résolu le rendu vectoriel web avec WebGL. En 2026, MLT résout le problème suivant : MVT a été conçu quand les volumes correspondaient à une ville. Douze ans plus tard, trois contraintes ont changé : **volumes planet-scale** (Overture Maps, LiDAR nationaux), **WebGPU** (nécessite des données structurées différemment de WebGL), et l'efficacité du **row-based vs column-based**.

Architecture technique — la différence column-oriented

STRUCTURE MÉMOIRE — MVT ROW-BASED VS MLT COLUMN-ORIENTED

```
// MVT – Row-based : chaque feature est un bloc complet
Feature 1: [geometry] [name: "rue de Rivoli"] [class: "primary"] [oneway: true]
Feature 2: [geometry] [name: "bd Haussmann"] [class: "secondary"] [oneway: false]
Feature 3: [geometry] [name: "rue du Faubourg"] [class: "primary"] [oneway: true]
// → Lire "class" de toutes les features = parcourir toute la structure

// MLT – Column-oriented : chaque attribut est un tableau contigu
col_geometry : [geom1_bytes] [geom2_bytes] [geom3_bytes] ...
col_name      : ["rue de Rivoli"] ["bd Haussmann"] ["rue du Faubourg"] ...
col_class     : ["primary"] ["secondary"] ["primary"] ...
col_oneway    : [true] [false] [true] ...
// → Lire "class" = lire un seul tableau contigu → SIMD possible
// → Compression RLE : ["primary", "secondary", "primary"] → très efficace
```

Performances mesurées (benchmarks OSM)

MÉTRIQUE	MVT	MLT	AMÉLIORATION
Taille tuile (OSM planet dense)	100%	20–35%	Jusqu'à 6× plus petit
Décodage JS (sans SIMD)	100%	~50%	2× plus rapide
Décodage avec SIMD CPU	100%	~30%	3× plus rapide
Transfert CPU→GPU (WebGPU)	Conversion requise	Direct (aligned buffers)	Latence réduite significativement
Support types complexes	Limité	Nested, lists, maps	Extension fonctionnelle
Coordonnées 3D	Non natif	Oui (élévation)	Nouvelle capacité

Fonctionnalités étendues

MLT ajoute des capacités structurelles absentes de MVT : **coordonnées 3D natives** (altitude comme type de première classe — bâtiments extrudés, terrain, LiDAR), **nested types** (attributs contenant objets ou listes — horaires OSM, relations routières), **linear referencing** et **M-values** (positionnement kilométrique pour les données routières), et **maps et listes** pour les attributs polymorphes.

Guide de décision MLT vs MVT

CONTEXTE	RECOMMANDATION	RAISON
Application web standard, données modérées	MVT	Écosystème mature, tooling riche
Volume planet-scale, performance critique	MLT	Compression 3–6×, décodage 2–3× plus rapide
Rendu WebGPU avancé	MLT	Buffers alignés GPU, moins de conversion
Données 3D / élévation	MLT	Support natif 3D
Contrainte de compatibilité maximale	MVT	Support universel tous serveurs/clients
Projet greenfield haute performance 2026+	MLT + PMTiles	Architecture optimale 2026

Statut 2026 — MLT

MLT est en phase d'adoption précoce. Les implémentations MapLibre sont disponibles et stables, mais l'écosystème de génération côté serveur est encore limité. Pour la production standard, MVT reste le choix sûr. MLT est le choix optimal pour les applications hautes performances, les gros volumes, et les projets WebGPU. **Projection : maturité comparable à MVT d'ici 2028–2029.**

Partie 6 — Formats 3D, LiDAR et Nuages de Points

La troisième dimension s'est imposée avec la démocratisation du LiDAR aéroporté et les besoins des smart cities et du BIM. Ces formats répondent à des problématiques radicalement différentes des formats 2D : volumétrie massive (des milliards de points pour un scan LiDAR), modélisation sémantique des bâtiments, visualisation 3D temps réel de scènes urbaines.

FORMATS LIDAR · LAS : 2003 / LAZ : 2010–2011

LAS / LAZ — et COPC

Le standard LiDAR — des milliards de points géoréférencés

ASPRS (LAS) · Martin Isenburg / LASzip (LAZ)

LAZ : compression lossless ~75–85%

COPC : cloud-native (HTTP Range)

Statut 2026 : Standard LiDAR

LAS est publié par l'ASPRS en 2003 pour les nuages de points LiDAR (coordonnées XYZ, intensité, classification, timestamp). LAZ (vers 2010–2011 par Martin Isenburg) est la version compressée sans perte : réduction de 70 à 85% sans sacrifier la précision. LAZ est aujourd'hui le format standard de distribution. Un successeur émerge : COPC (Cloud Optimized Point Cloud) — applique à LAS la même idée que COG à GeoTIFF, avec un index spatial permettant des requêtes partielles via HTTP Range Requests.

CARACTÉRISTIQUE	LAS	LAZ
Compression	Non	Oui (lossless, ~75–85% de réduction)
Décodage	Direct	Décompression requise (LASzip lib open source)
Support logiciels	Universel	Universel
Usage recommandé	Traitement local intensif	Distribution, stockage, archivage

FORMATS 3D · 2015 / OGC STANDARD 2022

3D Tiles

Visualisation 3D massive – smart city et jumeaux numériques

Cesium Inc.

Streaming hiérarchique (octree/quadtree)

glTF 2.0 (v1.1)

Statut 2026 : Standard jumeau numérique ✓

Comment visualiser une ville entière en 3D dans un navigateur ? Les modèles BIM d'un quartier peuvent contenir des millions de triangles. 3D Tiles résout ce problème avec un streaming hiérarchique basé sur un arbre spatial (octree ou quadtree) : les tuiles les plus grossières sont chargées en premier, puis raffinées à mesure que l'utilisateur se rapproche. Chaque nœud peut contenir des meshes 3D glTF (bâtiments), des nuages de points LiDAR, ou des objets instanciés (arbres, lampadaires).

FORMATS 3D · 2002 / OGC 1.0 EN 2008

CityGML / CityJSON

Modélisation urbaine sémantique – le BIM du SIG

OGC

5 niveaux de détail LoD 0→4

CityJSON (2019) : version JSON moderne

Statut 2026 : Institutionnel + smart city

CityGML représente non seulement la géométrie des bâtiments, mais aussi leur structure sémantique (toits, murs, portes, fenêtres), leurs attributs (usage, date de construction) et leurs relations. Il définit cinq niveaux de détail LoD 0 (empreinte au sol) à LoD 4 (modèle intérieur complet). La plupart des projets smart city utilisent LoD 2. Sa complexité XML est un frein — CityJSON (2019) est la version JSON qui gagne en popularité.

IFC — Industry Foundation Classes

Le standard BIM — la convergence avec le SIG

buildingSMART International

IFC 4.3 : géolocalisation native

Convergence BIM-GIS active

Statut 2026 : Standard BIM incontournable

IFC est le standard d'échange du BIM. Un modèle IFC contient la géométrie précise d'un bâtiment avec ses métadonnées (matériaux, propriétés thermiques, coûts). IFC 4.3 intègre nativement la géolocalisation. La chaîne IFC → CityGML/CityJSON → 3D Tiles pour la visualisation web est maintenant opérationnelle.

WORKFLOW	DESCRIPTION
IFC → GeoPackage	Extraction des empreintes et attributs pour SIG bureautique
IFC → CityJSON/CityGML	Modélisation urbaine avec sémantique BIM
IFC → 3D Tiles	Visualisation web via CesiumJS / iTwin
IFC + PostGIS	Requêtes spatiales sur composants BIM

Partie 7 — Formats Scientifiques et Climatiques

Les sciences de la Terre produisent des données multidimensionnelles qui dépassent le cadre classique du SIG 2D : données climatiques sur une grille régulière avec une dimension temporelle, profils océanographiques avec une dimension de profondeur, données de modèle atmosphérique sur plusieurs niveaux de pression.

FORMATS SCIENTIFIQUES · 1989 · CLIMATOLOGIE / OCÉANOGRAPHIE

NetCDF — Network Common Data Form

Le standard des données climatiques et océanographiques

Unidata (UCAR)

Variables n-dimensionnelles

NetCDF-4 basé sur HDF5

Statut 2026 : Standard ECMWF/Copernicus 

NetCDF est un format binaire auto-descriptif pour les données scientifiques multidimensionnelles. Il contient des variables (tableaux n-dimensionnels), des dimensions (latitude, longitude, temps, altitude) et des attributs. Un fichier de prévisions météo peut contenir : dimensions lat[721] × lon[1440] × time[240] × pressure_level[37] avec les variables temperature, wind_u, wind_v, humidity — soit ~37 milliards de valeurs dans un seul fichier. NetCDF-4 (basé sur HDF5) supporte la compression interne, l'accès parallèle et les groupes de variables. Les bibliothèques Python `xarray` et `netCDF4-python` permettent une manipulation expressive.

FORMATS SCIENTIFIQUES · 1998 · DONNÉES VOLUMINEUSES

HDF5 — Hierarchical Data Format 5

Conteneur scientifique universel

The HDF Group

Groupes + datasets n-dimensionnels

NetCDF-4 construit sur HDF5

Statut 2026 : LiDAR aéroporté + simulation

HDF5 organise les données en groupes (comme des répertoires) et datasets (tableaux n-dimensionnels), avec compression, chunking et accès parallèle. NetCDF-4 est construit sur HDF5. HDF5 est utilisé pour les données de télédétection (HDF-EOS pour MODIS/VIIRS), les données LiDAR aéroportées et les données de simulation numérique.

Partie 8 — Formats CAD dans l'écosystème géospatial

La frontière entre CAD et SIG s'est progressivement estompée. Les projets d'infrastructure (routes, réseaux, bâtiments) sont souvent conçus en CAD puis intégrés dans des systèmes SIG pour la gestion patrimoniale, la maintenance et l'analyse territoriale.

FORMATS CAD · DXF : 1982 / DWG : 1982 · CONSTRUCTION & INFRASTRUCTURE

DXF / DWG

Le format CAD dominant — incontournable en construction

Autodesk

DWG : binaire propriétaire

DXF : échange texte/binaire

Statut 2026 : Incontournable construction

DWG est le format natif binaire d'AutoCAD (propriétaire, spécification partiellement publique). DXF est la version d'échange texte publiée par Autodesk pour l'interopérabilité. L'intégration CAD-SIG pose plusieurs défis : coordonnées relatives (pas géographiques), structure en couches et blocs (pas en features typées), géométries complexes (splines, arcs, NURBS), et attributs dans les blocs à mapper vers des attributs SIG.

Le flux CAD → SIG en pratique

1 Export DXF depuis AutoCAD

Sélection des calques pertinents, nettoyage des entités inutiles.

2 Import QGIS ou FME

Conversion des entités CAD en features SIG, mapping des attributs de blocs.

3 Géoréférencement

Application de la transformation de coordonnées locale → Lambert 93 / WGS84.

4 Nettoyage topologique

Fermeture des polygones, correction des doublons, nœuds.

5 Chargement PostGIS

Vers une base de données spatiale pour la gestion patrimoniale et la GMAO.

→ Pertinence Odoo-AutoCAD

Dans le contexte d'un ERP de gestion industrielle ou de construction (Odoo), l'intégration AutoCAD-SIG permet de lier les plans d'équipement (blocs AutoCAD) aux fiches produits et nomenclatures (BOM) de l'ERP. Un bloc AutoCAD représentant une pompe peut référencer directement le composant Odoo correspondant avec son stock, sa maintenance planifiée et son historique d'interventions.

Partie 9 — Guide décisionnel complet

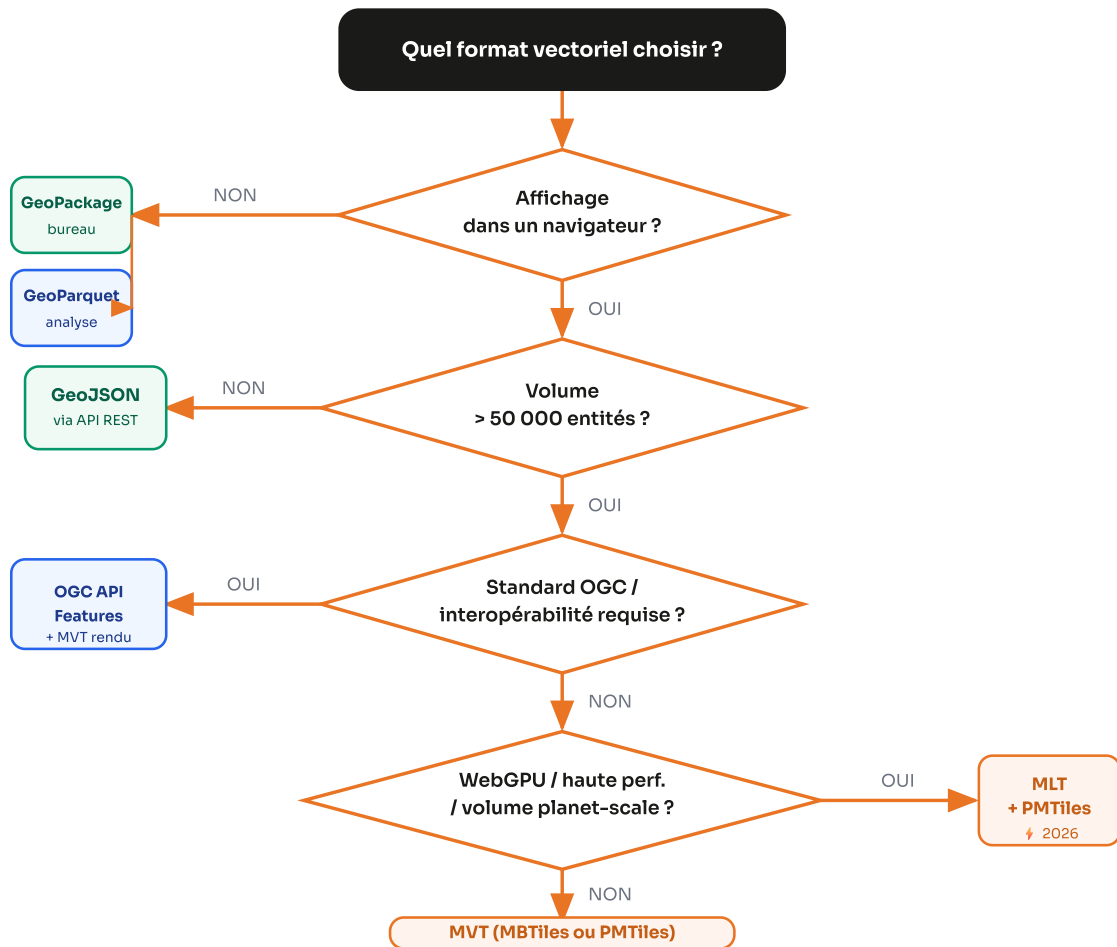
Ce guide synthétise les recommandations pratiques de l'ensemble du document. Il est conçu pour être utilisé comme référence rapide lors d'un choix architectural.

9.1 Matrice décisionnelle par cas d'usage

CAS D'USAGE	VOLUME	FORMAT RECOMMANDÉ	ALTERNATIVE
POI simples (restaurants, commerces)	<5 000	GeoJSON via API REST	CSV + coordonnées
Données de localisation légères	<50 000	GeoJSON ou OGC API Features	FlatGeobuf
Application web interactive	<100 000	MVT (tiles) + PostGIS	GeoJSON paginé
Cartographie volumineuse web	>100 000	MVT ou MLT	PMTiles + MVT/MLT
Haute performance / WebGPU	Tout volume	MLT + PMTiles	MVT si MLT non disponible
Portail institutionnel OGC	Variable	OGC API Features	WFS 2.0
Fond de carte statique	Planet-scale	XYZ raster ou PMTiles raster	WMTS
Échange bureautique / QGIS	Tout volume	GeoPackage	Shapefile (legacy)
Export / archivage	Tout volume	GeoPackage ou Shapefile	GeoJSON si petit
Pipeline data engineering	>1 million	GeoParquet + DuckDB	PostGIS
Analyse massive spatiale	Téraoctets	GeoParquet (Spark/BigQuery)	PostGIS partitionné
Données raster cloud	Tout volume	COG (Cloud Optimized GeoTIFF)	GeoTIFF local
Streaming raster partiel	Tout volume	COG sur S3/CDN	WMS avec cache
Données GPS terrain	N/A	GPX	GeoJSON
Nuage de points LiDAR	Milliards de points	LAZ (stockage) / COPC (cloud)	LAS local
Visualisation 3D bâtiments	Urbain	3D Tiles via CesiumJS	CityJSON + Three.js
BIM géolocalisé	Bâtiment	IFC → 3D Tiles	IFC → CityGML
Données climatiques	Multidimensionnel	NetCDF	HDF5
Plans CAD → SIG	N/A	DXF → GeoPackage via QGIS/FME	DWG → PostGIS

9.2 Arbre de décision — Format vectoriel web

FIGURE 11 — ARBRE DE DÉCISION : CHOISIR LE BON FORMAT VECTORIEL WEB



ARBRE DE DÉCISION — FORMAT VECTORIEL WEB 2026

9.3 Compatibilité par logiciel

LOGICIEL / FRAMEWORK	FORMATS SUPPORTÉS NATIVEMENT
QGIS	GeoPackage, Shapefile, GeoJSON, WMS, WFS, OGC API, MVT, GeoTIFF, LAZ
ArcGIS Pro	GeoPackage, Shapefile, GeoJSON, WMS, WFS, MVT, GeoTIFF
Leaflet.js	GeoJSON, WMS, XYZ raster — via plugins : MVT, WFS
MapLibre GL JS	MVT, MLT, GeoJSON, XYZ raster, PMTiles, GeoTIFF (raster source)
Mapbox GL JS	MVT, GeoJSON, XYZ raster
CesiumJS	3D Tiles, KML, GeoJSON, WMS, WMTS, Terrain quantized-mesh
deck.gl	GeoJSON, MVT, GeoArrow, GeoParquet (via loaders.gl)
PostGIS	Import/export : GeoJSON, Shapefile, GeoPackage, WKB, WKT
GDAL/OGR	200+ formats — référence universelle
osmium-tool	OSM PBF (lecture, écriture, filtrage, conversion GeoJSON/GeoPackage)
Python geopandas	Shapefile, GeoJSON, GeoPackage, GeoParquet, PostGIS, FlatGeobuf
DuckDB spatial	GeoParquet, GeoJSON, Shapefile, WKT, WKB

9.4 Cinq scénarios architecturaux types

SCÉNARIO 1

Application web légère

<5 000 entités · POI, capteurs IoT, annuaire

PostgreSQL + PostGIS



API REST / OGC API Features



GeoJSON → Leaflet / MapLibre

Infrastructure : VPS simple, pas de tile server

SCÉNARIO 2

Portail institutionnel OGC

Open data collectivité · INSPIRE · Géoportail

PostgreSQL + PostGIS



GeoServer / pygeoapi (OGC API)



GeoJSON + MVT → OpenLayers

Infrastructure : serveur dédié, Docker

SCÉNARIO 3

Haute performance >100k

Réseaux infrastructure · Mobilité urbaine · Tracking

PostGIS + index GiST



Tippecanoe / Planetiler → PMTiles



MVT → MapLibre GL JS

Infrastructure : Serverless tiles (R2/S3), API légère

SCÉNARIO 4

Pipeline data engineering

Satellitaire · Mobilité nationale · ETL géospatial

GeoParquet sur S3



DuckDB (single) / Spark (distribué)



PMTiles (rendu) · GeoParquet (analytics)

Infrastructure : cloud object storage + orchestration Airflow/Prefect

SCÉNARIO 5

Jumeau numérique / Smart City

Maquette numérique · Patrimoine bâti · Smart city

IFC 4.3 géolocalisé (BIM source)



FME / py3dtiles → 3D Tiles



CesiumJS (3D) + MapLibre (2D) + Odoo
ERP

Infrastructure : Cesium ion / iTwin / self-hosted

Partie 10 — Architectures de référence 2026

Cette partie présente les stacks techniques recommandées par profil d'usage. Le socle commun incontournable reste **PostgreSQL + PostGIS** : maturité, richesse fonctionnelle (ST_Intersects, ST_Buffer, ST_AsMVT), écosystème universel (GDAL, QGIS, GeoServer, SQLAlchemy).

10.2 Architecture standard — PME / Startup

COUCHE	TECHNOLOGIE RECOMMANDÉE	ALTERNATIVE
Base de données	PostgreSQL 16 + PostGIS 3.4	SQLite + SpatiaLite (si offline)
API données	FastAPI + GeoJSON / OGC API (pygeoapi)	Express.js + Knex
Génération tiles	pg_tileserv ou Martin (MVT dynamique)	Tippecanoe (statique)
Distribution tiles	PMTiles sur Cloudflare R2	MBTiles + TileServer GL
Frontend	MapLibre GL JS	Leaflet + plugins
Style	MapLibre Style Spec (JSON)	Maputnik (éditeur visuel)
Fond de carte	OpenMapTiles (PMTiles)	Maptiler Cloud

10.3 Architecture cloud-native — volumétrie élevée

COUCHE	TECHNOLOGIE	JUSTIFICATION
Stockage données brutes	GeoParquet sur S3 / R2	Columnar, coût minimal
Traitement ETL	DuckDB + spatial extension	SQL massif sans infra serveur
Base opérationnelle	PostgreSQL + PostGIS (RDS ou Supabase)	ACID, fonctions spatiales
Tiles vectorielles	Planetiler → PMTiles (MLT ou MVT)	Génération offline, distribution CDN
Tiles raster	COG + TiTiler sur Lambda/Cloud Run	Serverless, auto-scaling
API	OGC API Features (pygeoapi)	Interopérabilité OGC
Frontend	MapLibre GL JS + deck.gl	Performance GPU, grandes données
CDN	Cloudflare R2 + Workers	Latence globale minimale

10.4 Architecture institutionnelle / OGC-compliant

COUCHE	TECHNOLOGIE	JUSTIFICATION
Serveur OGC	GeoServer 2.x ou MapServer	WMS, WFS, WMTS, OGC API
Base de données	PostgreSQL + PostGIS	Standard de production
Formats sortie	GML/GeoJSON (WFS), PNG/JPEG (WMS)	Interopérabilité OGC
Portail de données	GeoNode ou CKAN avec plugin géo	Open data, catalogue INSPIRE
Frontend	OpenLayers (support OGC natif)	Standards complets
Métadonnées	GeoNetwork (CSW)	Catalogue INSPIRE-compliant

Partie 11 — Tendances et horizon 2030

Les tendances identifiées ici ne sont pas spéculatives — elles sont déjà visibles en 2026, certaines en phase de production, d'autres en adoption précoce. Elles permettent d'anticiper les choix architecturaux pour les 3 à 5 prochaines années.

🕒 TENDANCE 1 — WEBGPU COMME NOUVEAU TERRAIN DE JEU

Rendu avancé et compute spatial côté client

WebGPU est disponible dans Chrome, Firefox et Safari depuis 2023–2024. Pour le SIG web, cela ouvre le rendu 3D physiquement réaliste, le compute spatial côté client (agrégation, clustering, interpolation exécutés sur GPU dans le navigateur) et des transferts de données optimisés vers le GPU. MLT est le premier format tile conçu pour WebGPU. GeoArrow fournit les buffers alignés pour les transferts CPU→GPU. **Horizon 2030** : analyses spatiales complexes sur des millions d'entités en quelques millisecondes, sans serveur.

🕒 TENDANCE 2 — CONVERGENCE BIM-GIS ACCÉLÉRÉE

Jumeaux numériques nationaux et IFC géolocalisé

Les projets de jumeau numérique national (France DGALN, Singapour, Dubai) poussent la convergence BIM-GIS vers une maturité industrielle. IFC 4.3 géolocalisé supprime le besoin de transformation manuelle. La collaboration OGC-buildingSMART converge vers des standards communs IFC-CityGML. Dans les ERP (Odo), l'intégration des référentiels géospatiaux dans la gestion patrimoniale et la GMAO devient un standard.

🕒 TENDANCE 3 — GEOPARQUET + DUCKDB : LE SPATIAL SANS INFRASTRUCTURE

Démocratisation de l'analyse spatiale massive

La combinaison GeoParquet + DuckDB permet d'exécuter des requêtes spatiales analytiques sur des centaines de millions d'entités sur un simple laptop, sans base de données serveur. DuckDB-WASM permet d'exécuter des requêtes GeoParquet dans le navigateur. GeoParquet sur S3 + DuckDB = pipeline ETL spatial sans serveur, scalable et économique.

TENDANCE 4 — PMTILES + CDN : LA FIN DES TILE SERVERS ?

Architecture serverless pour les données statiques

Pour les projets n'ayant pas besoin de tiles dynamiques, PMTiles est probablement l'architecture optimale. Coût : fraction d'un tile server dédié, performances comparables grâce au réseau edge. **Limite actuelle** : pas de mises à jour partielles (régénération du fichier entier requise), pas de tiles dynamiques. Pour les données temps réel (tracking, IoT), un tile server reste nécessaire.

TENDANCE 5 — L'IA SPATIALE ET SES IMPLICATIONS

Segmentation satellite, embeddings géospatiaux, génération de tiles

Les modèles de segmentation (SAM, SegmentAnything) produisent des géométries vectorielles — GeoParquet avec GeoArrow est adapté. Les embeddings géospatiaux (représentations vectorielles ML de zones géographiques pour la recherche par similarité) nécessitent des extensions des formats existants. Les modèles de diffusion génèrent des fonds de carte stylisés via le MapLibre Style Spec.

11.6 Synthèse — Les formats à surveiller d'ici 2030

FORMAT / TECHNOLOGIE	TRAJECTOIRE	HORIZON
MLT (MapLibre Tile)	Adoption croissante, écosystème en développement	2027–2028 : maturité production
GeoParquet	Adoption rapide, standard Overture Maps. Spec 1.0 publiée 2023.	Déjà en production massive
PMTiles	Standard serverless tile, adoption en cours	2026–2027 : standard de facto
COPC	Standard LiDAR cloud émergent	2027 : adoption institutionnelle
GeoArrow	Brique fondamentale GPU, adoption Python	2027 : standard data-GPU
OGC API Tiles / Maps	Successeur WMTS/WMS, standard OGC futur	2028 : remplacement progressif
Shapefile	Déclin continu mais lent	2035+ : fin de vie dans les nouvelles API
WFS 1.x/2.x	Remplacement par OGC API Features	2028 : marginalisé dans les nouveaux projets

Conclusion générale

En parcourant trente ans d'évolution des formats géospatiaux — du Shapefile des années 1990 à MLT en 2026 — une trajectoire se dessine avec une clarté remarquable : chaque génération de formats répond à un changement structurel du contexte technologique et des usages.

Le SIG desktop des années 1990 avait besoin de formats stables pour des logiciels spécialisés — Shapefile et GeoTIFF ont parfaitement rempli ce rôle. L'explosion du web cartographique a exigé des formats JSON-natifs — GeoJSON a répondu et reste dominant pour sa niche. La révolution WebGL de 2014 a donné naissance à MVT et l'écosystème des tuiles vectorielles. Aujourd'hui, en 2026, WebGPU, GeoParquet, PMTiles et MLT posent les bases de la quatrième ère.

→ **Ce qu'il faut retenir**

Il n'existe pas de format universel. Shapefile pour l'échange bureautique, GeoJSON pour les API légères, MVT/MLT pour le rendu web volumineux, GeoParquet pour l'analyse massive, COG pour le raster cloud — chaque format a sa niche optimale.

Le choix d'un format est un choix architectural. Il conditionne la performance, la scalabilité, les coûts d'infrastructure et la maintenabilité sur le long terme.

PostgreSQL + PostGIS reste le socle. Quelle que soit l'architecture cible, la base de données spatiale est le cœur du système. Les formats de distribution (MVT, PMTiles, GeoParquet) sont des couches au-dessus.

Investir dans les formats modernes maintenant. GeoParquet, PMTiles et MLT ne sont plus expérimentaux. Les projets démarrés aujourd'hui devraient les considérer comme des options de production sérieuses.



La maîtrise des formats géospatiaux est une **compétence différenciante** pour tout professionnel du SIG et de la transformation digitale. Elle permet de faire des choix éclairés, d'anticiper les évolutions technologiques et de concevoir des architectures qui résistent au temps.

— Mattieu Pottier, consultant SIG & transformation digitale, MP-i

Références et ressources

Standards et spécifications

OGC Standards	Ensemble des standards OGC (WMS, WFS, OGC API Features, 3D Tiles...)	ogc.org/standards (https://www.ogc.org/standards/)
GeoJSON RFC 7946	Spécification officielle IETF du format GeoJSON	IETF RFC 7946 (https://datatracker.ietf.org/doc/html/rfc7946)
MVT Spec	Spécification Mapbox Vector Tiles (protobuf)	github.com/mapbox (https://github.com/mapbox/vector-tile-spec)
MLT Spec	Spécification MapLibre Tile (column-oriented, WebGPU)	github.com/maplibre (https://github.com/maplibre/maplibre-tile-spec)
GeoParquet	Spécification GeoParquet v1.0 — données géospatiales dans Apache Parquet	geoparquet.org (https://geoparquet.org/)
PMTiles	Documentation format PMTiles — single-file tile archive	protomaps.com (https://protomaps.com/docs/pmtiles)
COG	Cloud Optimized GeoTIFF — spécification et outils	cogeo.org (https://cogeo.org/)
STAC	SpatioTemporal Asset Catalog — standard de catalogage des COG	stacspec.org (https://stacspec.org/)

Librairies et outils de référence

GDAL/OGR	La librairie de conversion géospatiale universelle — 200+ formats	gdal.org (https://gdal.org/)
MapLibre GL JS	Moteur de rendu open source MVT/MLT — fork de Mapbox GL JS	maplibre.org (https://maplibre.org/)
Tippecanoe	Génération MVT haute qualité depuis GeoJSON/FlatGeobuf/PostGIS	github.com/felt (https://github.com/felt/tippecanoe)
Planetiler	Génération MVT/MLT planet-scale depuis OSM PBF — très performant	onthegomap/planetiler (https://github.com/onthegomap/planetiler)
DuckDB spatial	SQL spatial sans serveur — extension géospatiale pour DuckDB	duckdb.org (https://duckdb.org/docs/extensions/spatial)
rasterio / rio-cogeo	Création et validation de COG en Python	cogeotiff.github.io (https://cogeotiff.github.io/rio-cogeo/)
geopandas	Géospatial Python DataFrame — GeoJSON, GeoPackage, GeoParquet	geopandas.org (https://geopandas.org/)
PostGIS	Extension géospatiale pour PostgreSQL — référence production	postgis.net (https://postgis.net/)
Overture Maps	Données géospatiales mondiales ouvertes distribuées en GeoParquet	overturemaps.org (https://overturemaps.org/)

Glossaire

BBox / Bounding Box

Emprise rectangulaire d'une géométrie ou d'une requête spatiale, définie par (minX, minY, maxX, maxY).

BIM

Building Information Modeling — modélisation numérique des bâtiments avec données sémantiques et géométriques. Standard IFC porté par buildingSMART.

CDN

Content Delivery Network — réseau de distribution de contenu avec serveurs en périphérie (edge). Utilisé pour distribuer les tiles PMTiles et COG à faible latence.

COG

Cloud Optimized GeoTIFF — GeoTIFF avec index interne permettant des requêtes partielles via HTTP Range Requests. Standard de fait pour le raster cloud en 2026.

Column-oriented

Structure de stockage où les données d'une même colonne sont contiguës en mémoire. Optimal pour les opérations analytiques (OLAP). Base de GeoParquet et MLT. Opposé de row-based.

COPC

Cloud Optimized Point Cloud — extension du format LAS avec index spatial HTTP Range compatible. Équivalent COG pour les données LiDAR.

CRS

Coordinate Reference System — système de référence de coordonnées (projection cartographique + datum). WGS84 (EPSG:4326) est obligatoire en GeoJSON RFC 7946.

DXF

Drawing Exchange Format — format d'échange CAD texte ou binaire, créé par Autodesk pour AutoCAD. Version ouverte de DWG pour l'interopérabilité SIG.

FlatGeobuf

Format binaire vectoriel avec index Hilbert R-Tree intégré. Permet des requêtes spatiales bbox via HTTP Range Requests sans serveur. Basé sur FlatBuffers (Google).

GeoArrow

Extension Apache Arrow pour les géométries spatiales. Buffers mémoire alignés permettant des transferts zero-copy CPU→GPU. Brique fondamentale pour WebGPU et deck.gl.

GeoJSON

Format JSON pour les données géographiques vectorielles, standardisé en RFC 7946. WGS84 obligatoire. Standard de facto pour les API REST géospatiales.

GeoPackage

Format fichier OGC basé sur SQLite. Successeur du Shapefile pour les échanges SIG desktop. Un seul fichier, multi-couches, UTF-8, ACID. Spec OGC 2014.

GeoParquet

Extension Apache Parquet pour les données géospatiales. Standard pour le data engineering spatial. Spec v1.0 publiée en septembre 2023. Adopté par Overture Maps Foundation.

GML

Geography Markup Language — format XML de l'OGC pour les données géospatiales. Verbeux (5–15× GeoJSON). Format sous-jacent de WFS. Remplacé par GeoJSON dans OGC API Features.

HTTP Range Requests

En-tête HTTP permettant de télécharger une partie d'un fichier (bytes spécifiques). Fondement technique de COG, PMTiles, FlatGeobuf et COPC — permet le streaming sans serveur.

IFC

Industry Foundation Classes — standard d'échange BIM de buildingSMART International. IFC 4.3 (2021) intègre la géolocalisation native. Pivot de la convergence BIM-GIS.

KML

Keyhole Markup Language — format XML de visualisation géographique, créé par Keyhole/Google pour EarthViewer. Standard OGC depuis 2008. Niche : annotations grand public.

LAZ

Compression lossless du format LAS pour les données LiDAR. Réduction ~75–85% de la taille sans perte de précision. Développé par Martin Isenburg (LASzip). Standard de distribution LiDAR.

LiDAR

Light Detection And Ranging — technique de mesure par laser produisant des nuages de points 3D géoréférencés avec intensité, classification et timestamp.

MLT

MapLibre Tile — format vector tile column-oriented, successeur de MVT, conçu pour WebGPU. Compression 3–6× meilleure que MVT. Coordonnées 3D natives. Types complexes (nested). Annoncé janvier 2026.

MBTiles

Container SQLite pour tuiles cartographiques raster ou vectorielles. Format Mapbox, portable, indexé. Remplacé progressivement par PMTiles pour la distribution web.

MVT

Mapbox Vector Tiles — format de tuiles vectorielles binaires (protobuf), standard web depuis 2014. Row-based. Écosystème massif (Mapbox, MapLibre, OpenLayers, QGIS). Standard dominant en 2026.

OGC

Open Geospatial Consortium — organisation de standardisation des formats et protocoles géospatiaux (WMS, WFS, OGC API Features, 3D Tiles, GeoPackage, etc.).

OGC API Features

Standard OGC REST moderne pour exposer des données vectorielles. Successeur de WFS. API JSON, pagination, filtrage CQL2, documentation OpenAPI. Recommandé pour toute nouvelle API géospatiale.

OSM PBF

OpenStreetMap Protocolbuffer Binary Format — format de dump complet d'OpenStreetMap. Planétaire : ~80 Go. 2–3× plus compact et 5–6× plus rapide à parser que le XML OSM.

PMTiles

Single-file tile archive permettant la distribution de tiles sur CDN sans serveur dédié. HTTP Range Requests. Compatible MVT et MLT. Développé par Protomaps (Brandon Liu), 2022.

PostGIS

Extension PostgreSQL ajoutant des types géométriques, index GiST spatial et 800+ fonctions spatiales (ST_Intersects, ST_Buffer, ST_AzMVT...). Socle de toute architecture SIG professionnelle.

Protobuf

Protocol Buffers — format de sérialisation binaire Google. Utilisé par MVT pour l'encodage des tuiles vectorielles. Compact et rapide mais nécessite une librairie de décodage.

Row-based

Structure de stockage où toutes les colonnes d'une ligne sont contiguës en mémoire. Optimal pour les transactions OLTP. Architecture de MVT — limitation pour les gros volumes que MLT résout.

Shapefile

Format vectoriel ESRI (~1993, spec 1998). Multi-fichiers (.shp, .shx, .dbf). Noms de colonnes 10 chars max, 2 Go max, encodage ambigu. Legacy mais encore omniprésent dans les portails open data.

SIMD

Single Instruction Multiple Data — exécution parallèle de la même opération sur plusieurs données. Rendu possible par la structure column-oriented de MLT pour le décodage GPU.

STAC

SpatioTemporal Asset Catalog — standard de catalogage des données géospatiales cloud (COG, GeoParquet). Utilisé par ESA, USGS, NASA pour leurs archives satellitaires.

TopoJSON

Extension JSON de GeoJSON encodant la topologie (arcs partagés entre polygones adjacents). 2–4× plus compact pour les cartes choroplèthes. Niche D3.js. Créé par Mike Bostock, 2012.

Vector Tiles

Tuiles vectorielles — données géométriques encodées par tuile pour le rendu GPU côté client. Permettent le style dynamique côté client et l'interactivité sur les features. MVT et MLT.

WebGL

Web Graphics Library — API JavaScript pour le rendu 2D/3D via GPU dans le navigateur. Base de Mapbox/MapLibre GL JS depuis 2013. Successeur : WebGPU (disponible 2023–2024).

WebGPU

Successeur de WebGL — API GPU généraliste dans le navigateur. Support compute shaders (GPGPU). Base de la prochaine génération de rendu cartographique. MLT est conçu pour en tirer parti.

WFS

Web Feature Service — standard OGC pour l'accès aux données vectorielles via le web. Réponses GML/JSON. Successeur officiel : OGC API Features (2019).

WMS

Web Map Service — standard OGC (2000) pour la diffusion d'images cartographiques générées côté serveur. Traitement CPU à chaque requête. Toujours actif dans les portails institutionnels en 2026.

WMTS

Web Map Tile Service — standard OGC (2010) pour la diffusion de tuiles raster pré-calculées. Cachable sur CDN. Successeur : OGC API Tiles (2022).

WKB / WKT

Well-Known Binary / Well-Known Text — formats d'encodage des géométries en base de données. WKB est utilisé par GeoParquet pour stocker les géométries dans les colonnes Parquet.

XYZ

Schéma d'URL de facto pour les tuiles cartographiques : `{z}/{x}/{y}.png`. Standard non OGC mais universel, supporté nativement par Leaflet, MapLibre, Mapbox. Popularisé par Google Maps (2005).

À propos de l'auteur

Mattieu Pottier est consultant indépendant en transformation digitale, spécialisé dans les ERP (Odoo V18/V19), les systèmes d'information géographique (QGIS, développement QGIS), et les applications web (WordPress, FastAPI, Python, JavaScript). Il intervient sous la marque **MP-i — Mattieu Pottier Indépendant** auprès d'organisations souhaitant intégrer des outils open source dans leurs processus métier.

La trilogie de livres blancs techniques MP-i répond à un besoin terrain récurrent : disposer d'une référence synthétique, factuellement vérifiable et architecturalement actionnable sur la chaîne complète des données géospatiales — du format de stockage à l'exposition web.

Historique des révisions

VERSION	DATE	AUTEUR	MODIFICATIONS
v1.0.0	3 mars 2026	Mattieu Pottier	Version initiale – publication
v2.0.0	3 mars 2026	Mattieu Pottier	Ajout section MLT complète, mise à jour statuts 2026, enrichissement guide décisionnel
v3.0.0	3 mars 2026	Mattieu Pottier	Révision approfondie : architecture MLT column-oriented, benchmarks, GeoArrow, COPC, convergence BIM-GIS IFC 4.3, glossaire complet

Mattieu Pottier

Consultant SI indépendant – MPI Ingénierie & Numérique

Expert SIG, architecture système et Odoo. J'aide les collectivités, PME et bureaux d'études à construire des systèmes d'information calibrés sur leurs besoins réels – sans surcoût ni dépendance éditeur.