OGC API - Tiles

Tiles in the floor of the monument of discovery in Lisbon, Portugal.
(Lee Cannon April 2010, CC-BY-SA, https://www.flickr.com/photos/leecannon/5127274297)

Tiles in the floor of the terminal 2 of the Prague Airport, Czech Republic.
(Joan Masó, September 2022, CC0)

Tiles – Part 1: Core

GET

`.../tiles`

http://www.opengis.net/def/rel/ogc/1.0/tilesets-*

Server Response

- List of tilesets

200

Access any data tiled based on 2D Tile Matrix Set

Map Tiles, Vector Tiles, Coverage Tiles...

Main Conformance Classes

Core

Template based on tile *Matrix*, *Row*, *Column*

Tile Set

Tile set described using standard metadata

Tile Sets list

List of tile sets

GeoData Tile Sets List

OGC API Common Collection-level list of tilesets

Dataset Tile Sets List

OGC API Common Dataset-level list of tilesets

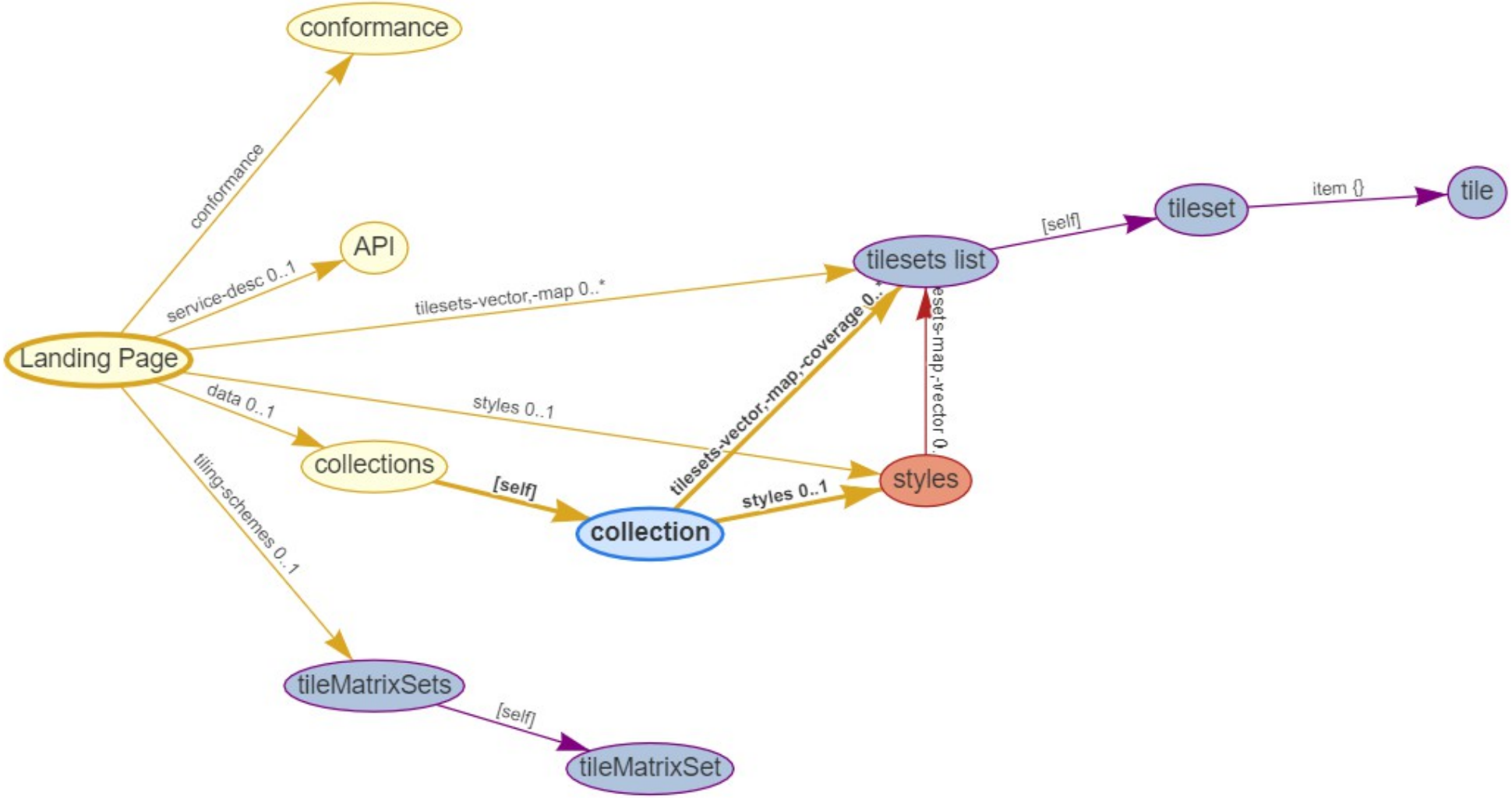
Formats Conformance Classes

PNG
JPEG
GeoTIFF
netCDF
GeoJSON
Mapbox Vector Tiles
(additional formats always allowed)

Additional Conformance Classes

Collection selection, Date and Time
Open API 3.0, XML

Following the links (resources & relation types)



Objectives for the sprint

- Validate / update implementations against published *OGC API – Tiles*
- Support development of test suite and new implementations
- Technology Integration Experiments

Introduction (Tiles)

OGC API – Tiles

- Retrieve tiles from a geospatial data resource
- Resource could be a map, a dataset, a feature collection, a coverage, the output of a process or workflow...
- Could be data tiles (e.g. vector tiles, coverage tile) or map tiles
- Tile sets are defined using *Tile Matrix Sets & TileSet Metadata*
- A style could also be used to change the appearance (e.g. maps) or content (e.g. filtering for vector tiles) with *OGC API – Styles*
- Stand-alone or attached to *Common, Maps, Coverages, Styles*

Minimalist Core (Tiles)

- A minimalist **Core** conformance class:
  - Tiles retrievable according to some *Tile Matrix Set* definition
  - A templated URL with variable identifiers should allow to express the path to individual tiles
    Example: {someAPI}/{column}/{row}/{level}.png
  - In *Core*, no specific identifiers or order specified, but they correspond to the tile matrix (zoom level), tile row and tile column
  - This enables most tile-based web mapping platform to conform
  - No mechanism to communicate *Tile Matrix Set* definition or templated URL – done out of bounds via other *OGC API – Tiles* conformance classes, or other mechanism like Mapbox *TileJSON*

Tileset (Tiles)

- Servers supporting **Tileset** conformance class defines a tileset according to the schema defined in *TileMatrixSet & Tileset Metadata*
  - Specifies TileMatrixSet used for the tiles
    - using tileMatrixSetURI if registered with OGC NA tile matrix set registry
    - always have a [ogc-rel:tiling-scheme] link for TileMatrixSet definition
  - Limits for each tile matrix sets, min / max tile matrix
  - Layers making up the tiles; for vector/coverage tiles: properties schema
  - Templated link (rel: **"item"**) for tiles using **{tileMatrix}**, **{tileRow}** and **{tileCol}**
  - Additional metadata about the tileset (e.g. dataType: *map, coverage, vector*)
  - Example resource: {datasetAPI}**/tiles/{tileMatrixSetId}**

Tilesets List (Tiles)

- Servers supporting **Tilesets List** conformance class list one or more tileset(s) available
  - Relation types to link to a list of tilesets:
    - http://www.opengis.net/def/rel/ogc/1.0/tilesets-map (map tiles)
    - http://www.opengis.net/def/rel/ogc/1.0/tilesets-coverage (coverage tiles)
    - http://www.opengis.net/def/rel/ogc/1.0/tilesets-vector (vector tiles)
  - The *list of tilesets* resource consists of a ***tilesets*** key for which the value is an array of tilesets, each defined as per the same schema as for a single tileset, but featuring a minimal amount of information: tileMatrixSetURI / tileMatrixSetDefinition and dataType
  - Each element in the list must contain link with rel: **"self"** to the tileset resource
  - Example resource: {datasetAPI}**/tiles**

Tilesets Resources (Tiles)

- {datasetAPI}/**tiles**
  - Dataset tilesets (e.g. multi-layer vector tiles) (*dataset-tilesets* conf. class)
- {datasetAPI}/**map/tiles**
  - Dataset map tilesets (*API – Maps map-tilesets* conf. class)
- {datasetAPI}/**{collectionId}/tiles**
  - Collection data tilesets (e.g. vector tiles) (*geodata-tilesets* conf. class)
- {datasetAPI}/**{collectionId}/coverage/tiles**
  - Collection coverage tilesets (*API - Coverages coverage-tilesets* conf. class)
- {datasetAPI}/**{collectionId}/map/tiles**
  - Collection map tilesets (*API – Maps map-tilesets* conf. class)

Styled Tilesets Resources (Tiles)

- {datasetAPI}/styles/{styleId}/**tiles**
  - Styled dataset tilesets (e.g. multi-layer vector tiles)
- {datasetAPI}/styles/{styleId}/**map/tiles**
  - Styled dataset map tilesets
- {datasetAPI}/styles/{styleId}/**{collectionId}/tiles**
  - Styled collection data tilesets (e.g. vector tiles filtered by style)
- {datasetAPI}/styles/{styleId}/**{collectionId}/map/tiles**
  - Styled collection map tilesets

Data set tiles: selecting collections (Tiles)

- Server may decide which collections to return by default
  - It does not need to include all of them
- **collections** query parameter allows client to select collections
  - e.g. **collections=AgricultureSrf,TransportationGroundCrv**
- For map tiles, the order in the collections list is default order
  - A style may override this order (or even intertwine elements of those collections)

Query parameters (Tiles)

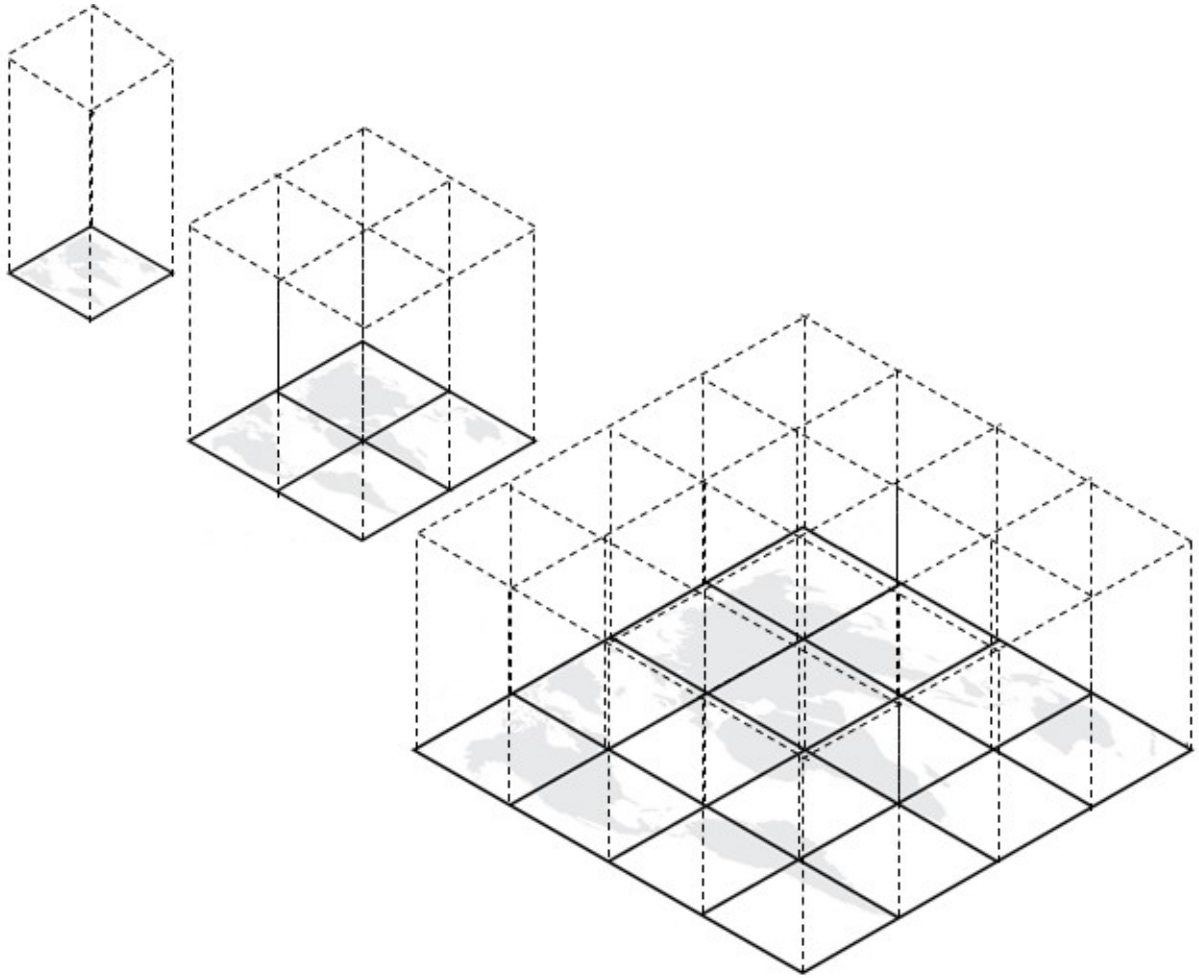
- No parameter required: easily retrieve a tile
- **transparent**
  - (for Map tiles) useful for PNG, rendering individual layers to be composited
- **bgcolor**
  - (for Map tiles) specify a background color
- **datetime**
  - ISO 8601 date/time string
  - One mechanism to support temporal datasets
- **subset**
  - For coverage tiles: can be used to subset (trim or slice) extra dimensions

Changes to TileMatrixSet

- Repository: https://github.com/opengeospatial/2D-Tile-Matrix-Set/
- TileMatrixSet schema
- Both *cellSize* and *scaleDenominator* must be specified
- identifier ➔ id
- boundingBox *lowerCorner* ➔ *lowerLeft*, *upperCorner* ➔ *upperRight*
- No more *type*
- New ***cornerOfOrigin*** [optional, default: **topLeft**]
- *topLeftCorner* ➔ *pointOfOrigin* (consistent with the conceptual model)

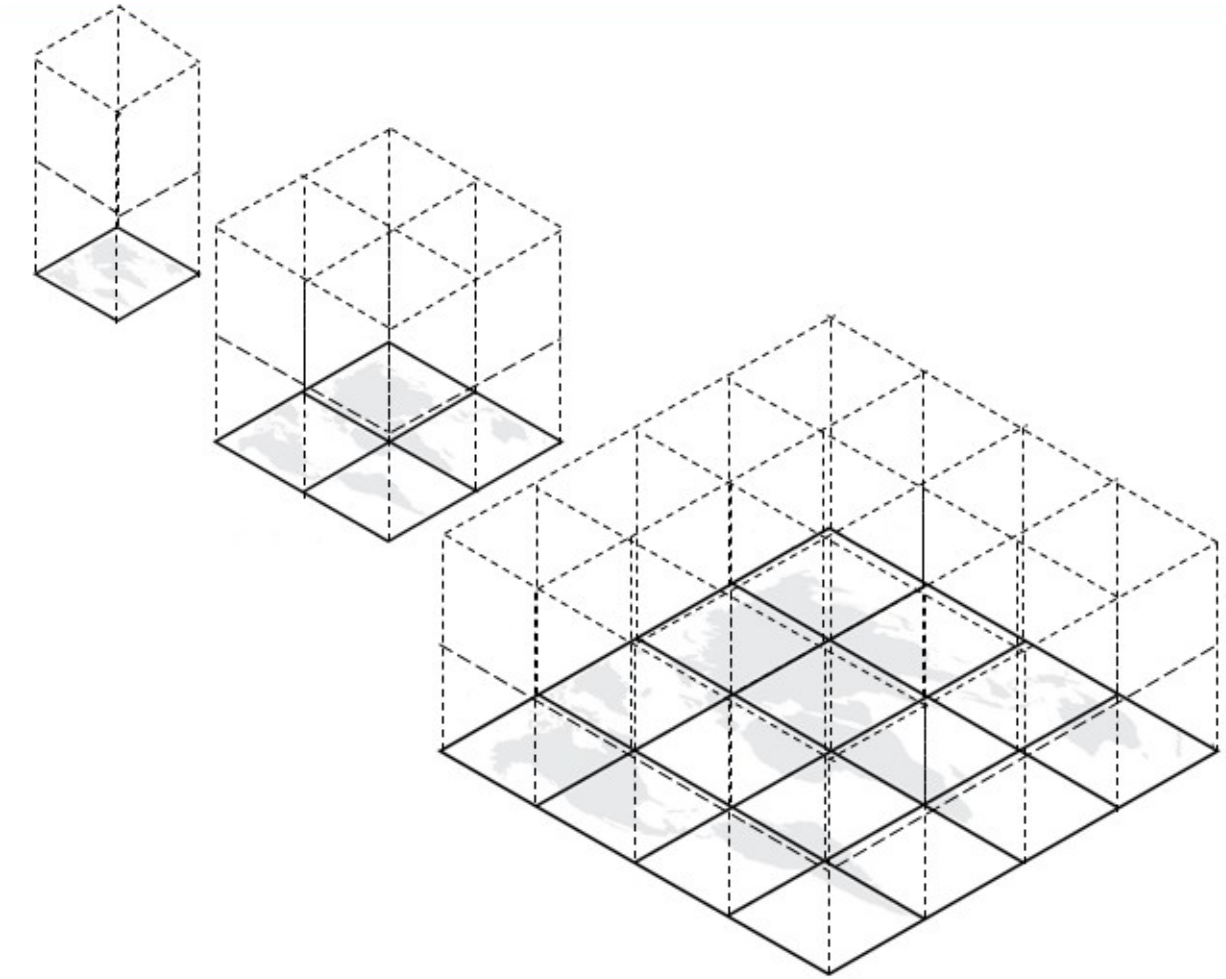
Extending TileMatrixSets to other dimensions

- Extend in other dimensions; Use TMS as is, reduce content with level
- Annex J of 2DTMS



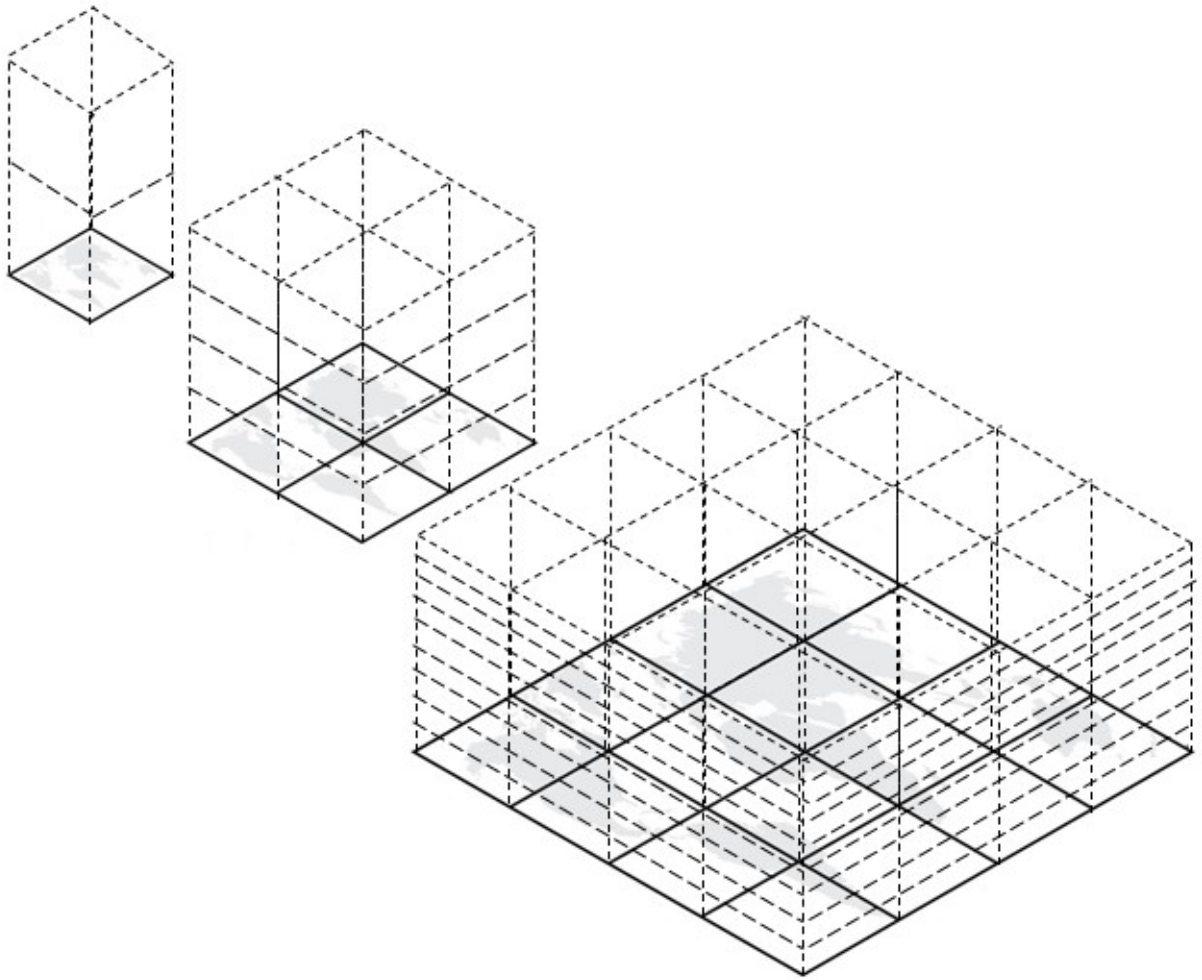
Extending TileMatrixSets to other dimensions

- Add slices in other dimension (but overviews still based only on other dimensions)



Extending TileMatrixSets to other dimensions

- Change the number of slices in other dimension as well at lower levels
  (turning into octree
  or hyperoctree)



Thank You!

Community

500+ International Members
110+ Member Meetings
60+ Alliance and Liaison partners
50+ Standards Working Groups
45+ Domain Working Groups
25+ Years of Not for Profit Work
10+ Regional and Country Forums

Innovation

120+ Innovation Initiatives
380+ Technical reports
Quarterly Tech Trends monitoring

Standards

65+ Adopted Standards
300+ products with 1000+ certified implementations
1,700,000+ Operational Data Sets Using OGC Standards

Contact info@ogc.org to schedule a meeting for an in-depth discussion with OGC staff and join our community today!