Webinar: OGC API Sprint Maps & Tiles Overview and status update

Joan Masó & Jerome St-Louis

May, 2021

#OGCAPI

The world's leading and comprehensive community of experts making location information:

- **F**indable
- **A**ccessible
- **I**nteroperable
- **R**eusable

Introduction (Maps)

OGC API – Maps

- Retrieve a map from a geospatial resource (image)
  - Could be static natural color image or map rendered server-side
- Resource could be a dataset, a feature collection, a coverage, the output of a process or workflow...
- The map can also be of a specific style for such a resource in conjunction with *OGC API – Styles*
- Leveraging building blocks from *OGC API – Common Part 1 & 2*
  - *Dataset landing page, list of collections, collection*

Resources (Maps)

- Relation type to link to a map
  - http://www.opengis.net/def/rel/ogc/1.0/**map**
- {datasetAPI}**/map**
  - A map of the whole dataset *(conf. class: dataset-map)*
- {datasetAPI}**/collections/{collectionId}/map**
  - A map of a particular collection *(conf. class: geodata-map)*
- {datasetAPI}**/map/tiles**
  {datasetAPI}**/collections/{collectionId}/map/tiles**
  - Map tilesets *(conf. class: map-tilesets, dependency on **API - Tiles**)*

Styles (Maps)

- Leveraging ***OGC API – Styles:***

  {datasetAPI}**/styles/{styleId}/map**
  {datasetAPI}**/styles/{styleId}/map/tiles**
  {datasetAPI}**/collections/{collectionId}/styles/{styleId}/map**
  {datasetAPI}**/collections/{collectionId}/styles/{styleId}/map/tiles**

  - maps of a particular style

Query parameters (Maps)

- No parameter required: easily retrieve a default map
- **width** and **height**
  - specify either or both – aspect ratio will be maintained if only one specified
- **transparent**
  - useful for PNG, rendering individual layers to be composited
- **bgcolor**
  - specify a background color
- **datetime**
  - ISO 8601 date/time string

Data set maps: selecting collections (Maps)

- 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**
- The order in the collections list is the default rendering order
  - A style may override this order (or even intertwine elements of those collections)

Sub-setting (Maps)

- Use of a common **subset** building block proposed at last (joint) sprint
  - Used in *Coverages*, could also be used to clip (rather than intersect) with *Features*
  - Discussed in Common
    - https://github.com/opengeospatial/ogcapi-common/issues/167
    - https://github.com/opengeospatial/ogcapi-common/tree/master/proposals/subsetting
  - Example: **subset=Lat(10:60),Lon(-140:-100)**
    - Return only the portion spanning from 10°N to 60°N and 140°W to 100°W
  - Either or both axes can be specified (the full extent is returned)
  - Remaining question on how axes names are determined (Common – collection resource)
- Server can still support **bbox** for compatibility
  - Equivalent example: **bbox=-140,10,-100,60**

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
    - using tileMatrixSetDefinition if custom TileMatrixSet (or as an additional local description of the TileMatrixSet)
  - Limits for each tile matrix sets, min / max tile matrix
  - Layers making up the tiles; for vector tiles: attributes 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)

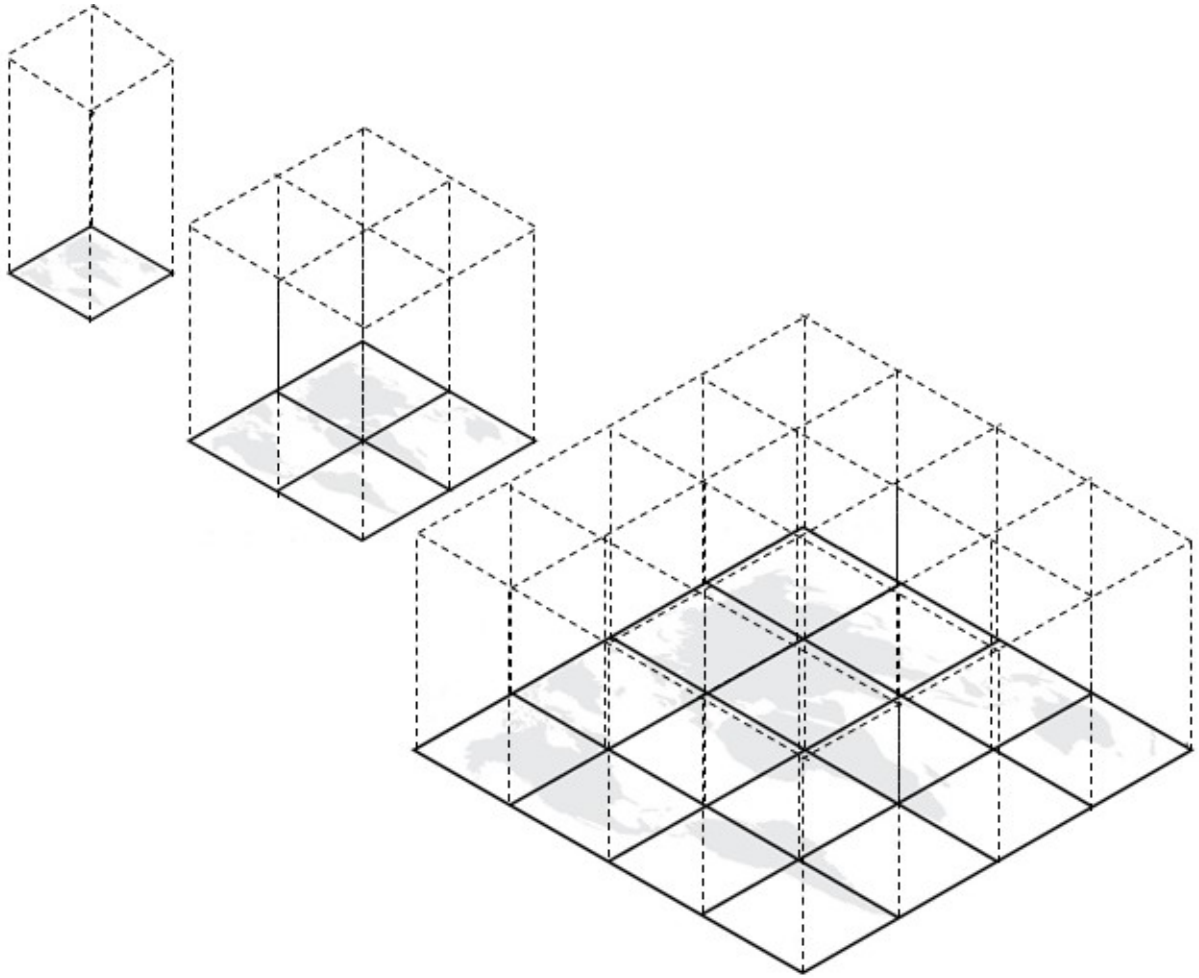
- No parameter required: easily retrieve a tile
- **transparent**
  - (for Map tiles) useful for PNG, rendering individual layers to be composited
- **bgcolor**
  - (for Map tiles) specify a background color
- **datetime**
  - ISO 8601 date/time string
  - One mechanism to support temporal datasets
-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

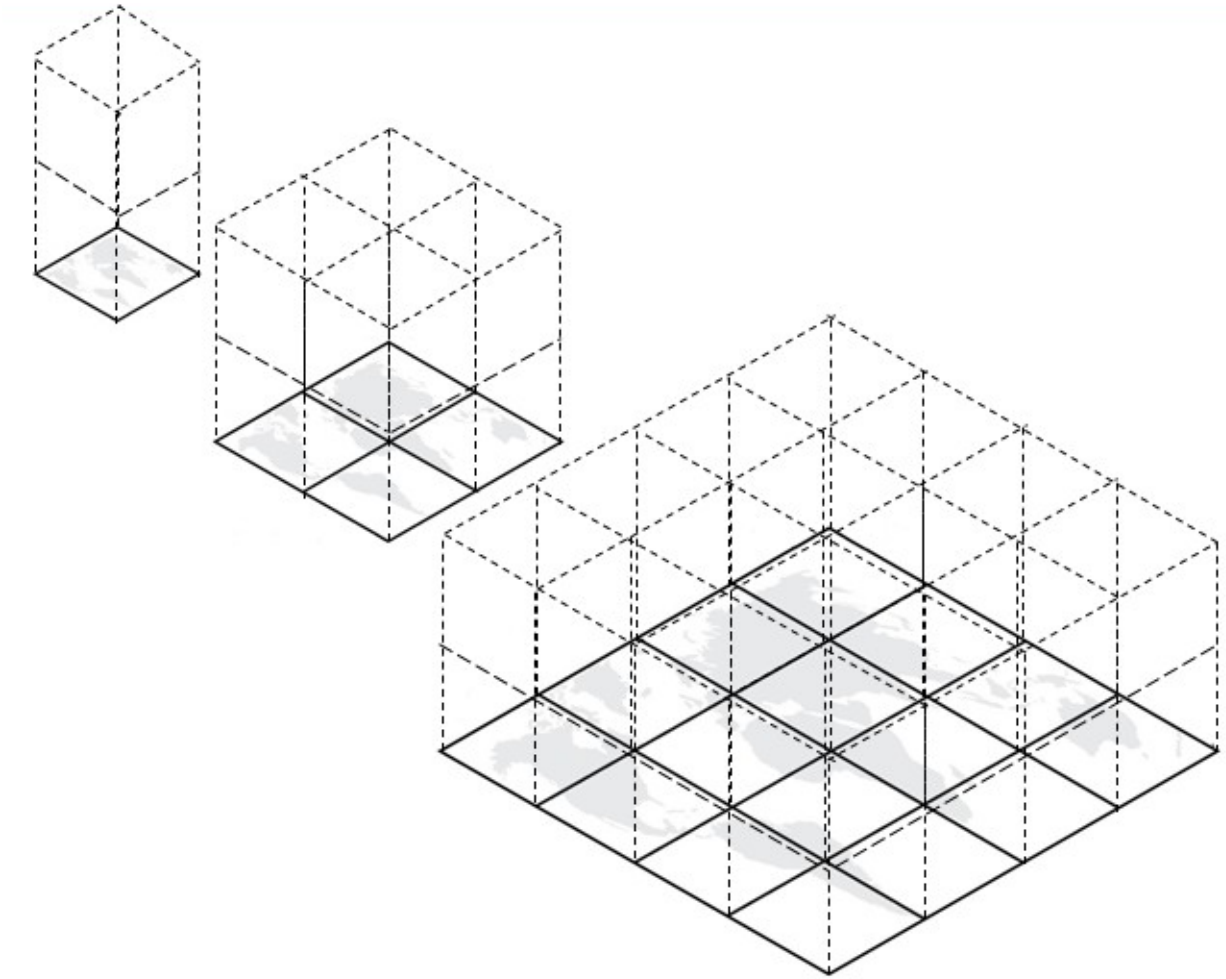
Extending TileMatrixSets to other dimensions

- Extend in other dimensions; Use TMS as is, reduce content with level
- TMS Issue #27



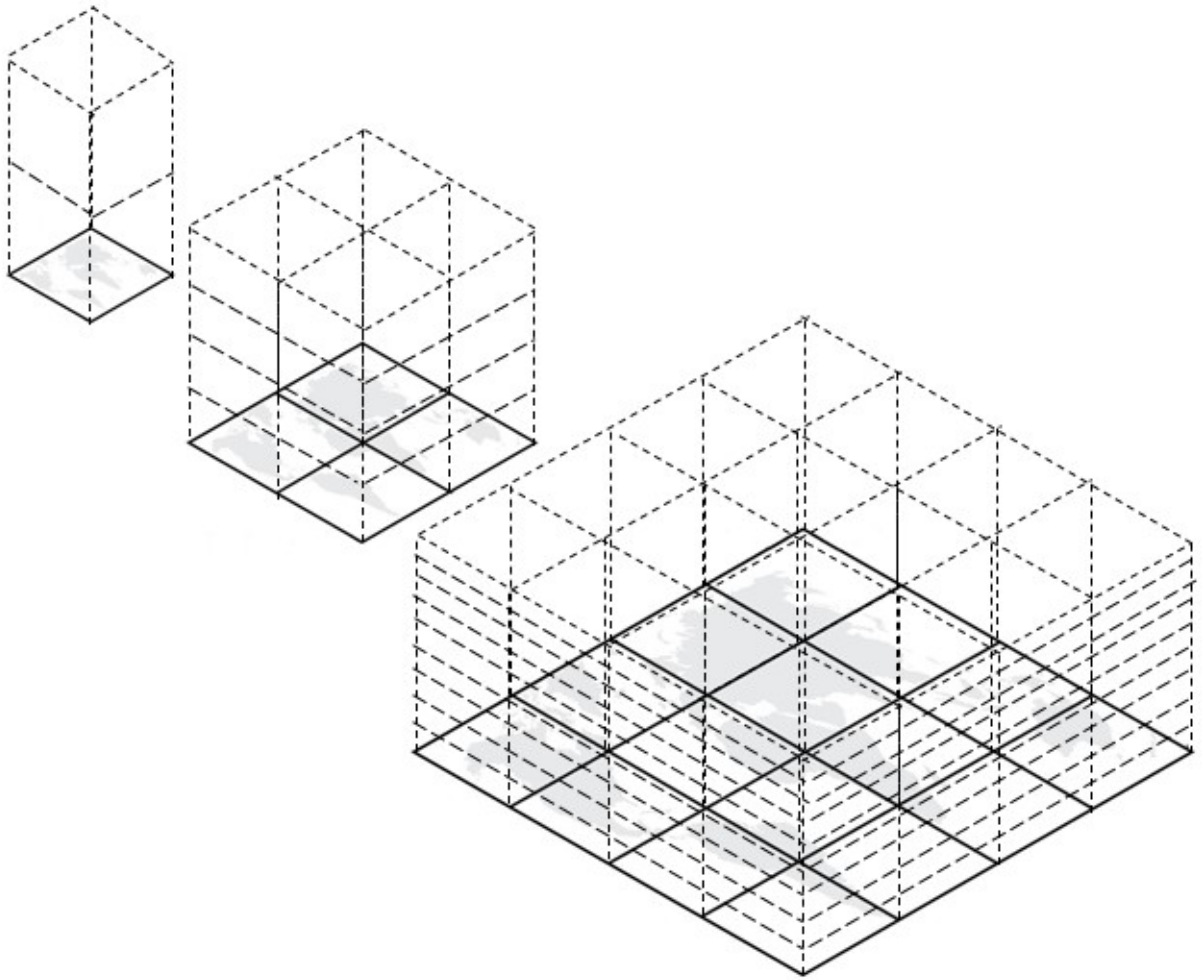
Extending TileMatrixSets to other dimensions

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



Extending TileMatrixSets to other dimensions

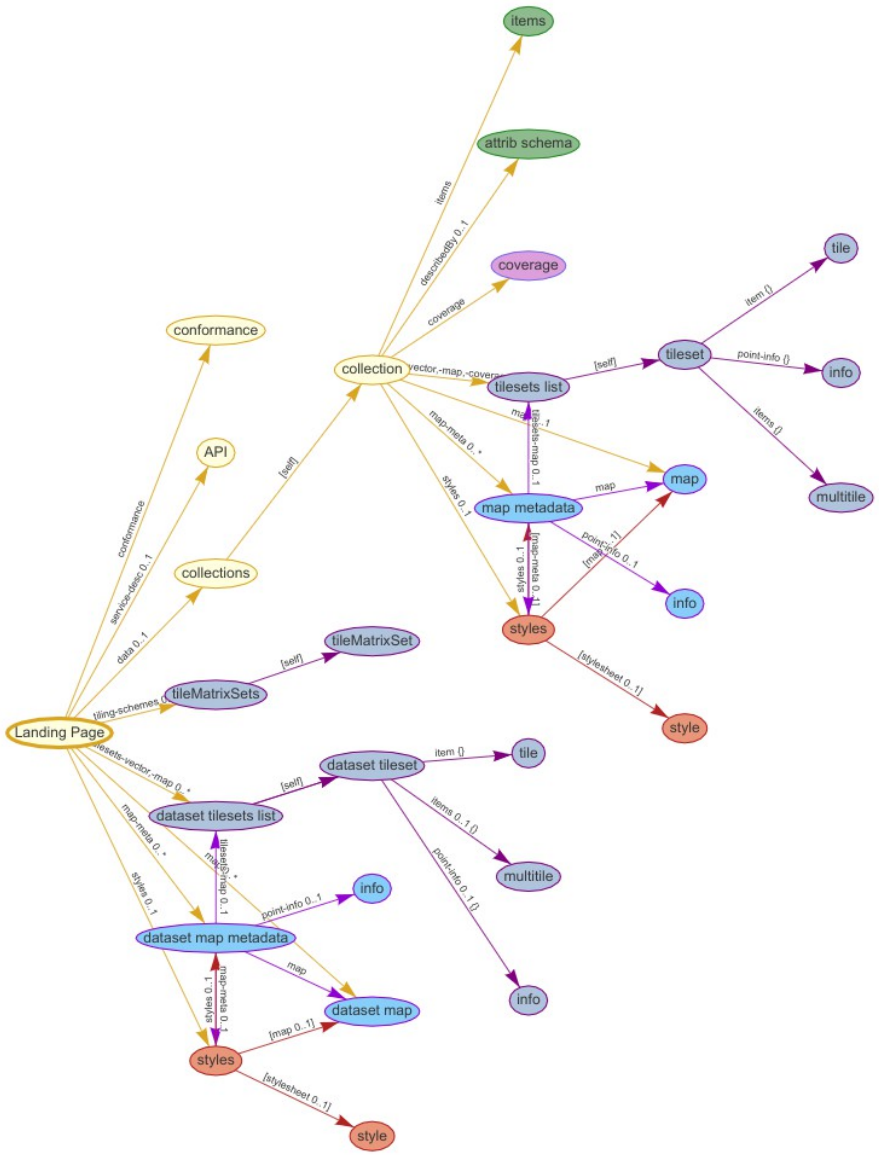
- Change the number of slices in other dimension as well at lower levels
  (turning into octree
  or hyperoctree)



Navigating OGC APIs

http://joanma.uab.cat/temp/graphs/OGCAPItiles.htm

- Resource paths and relation types at a glance



Objectives for the sprint

- Validate implementations against latest changes for Maps & Tiles
- Validate new TileMatrixSet and TileSet Metadata schemas
- Validate integration of Styles and Maps / Tiles APIs
- Validate vector and coverage tiles

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!