



Camptocamp

Tiling state of the art

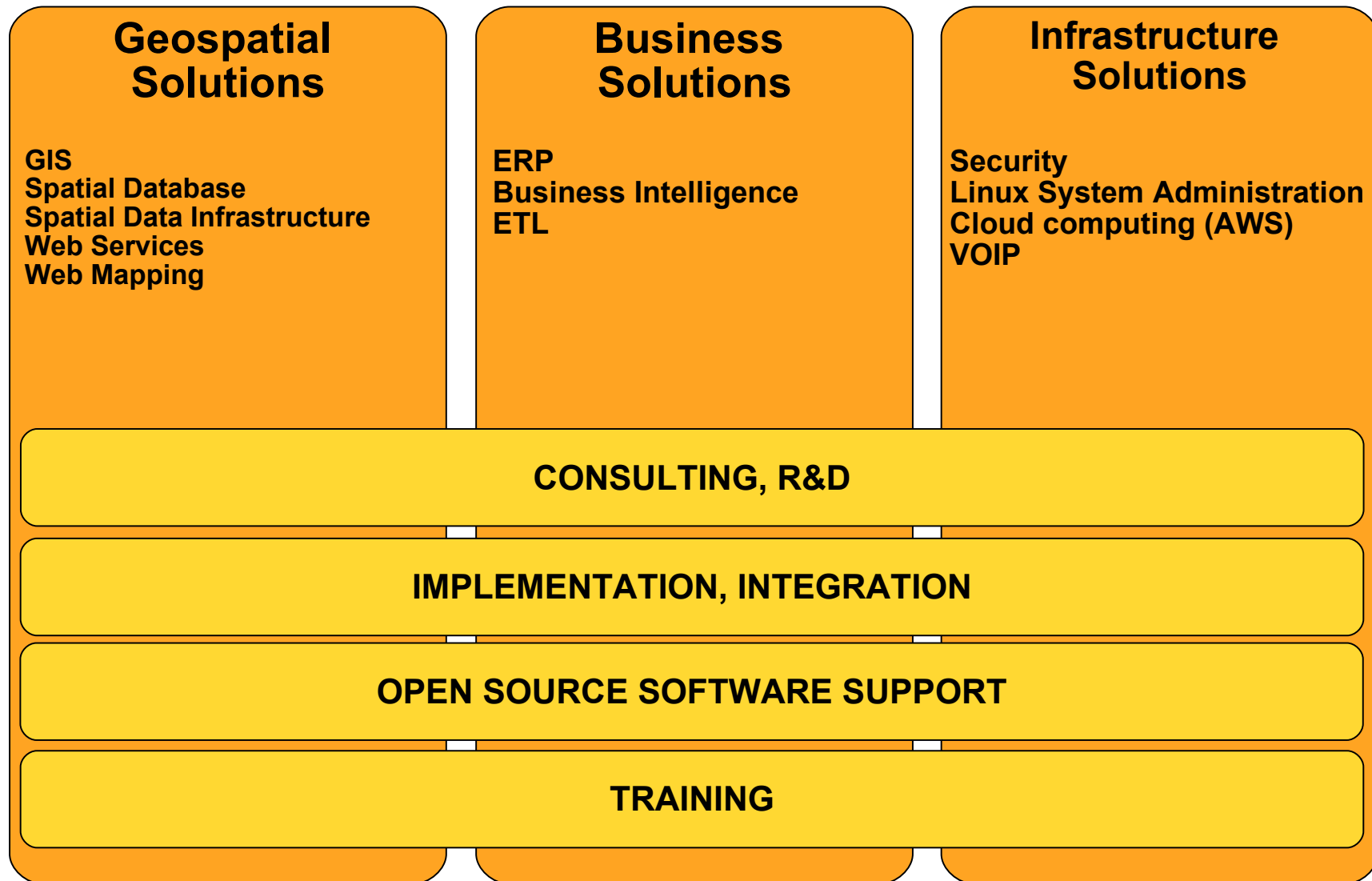
Cédric Moullet, Geospatial Director Camptocamp

Editor and integrator of Open Source software

40 Open Source Engineers



Complimentary products and services



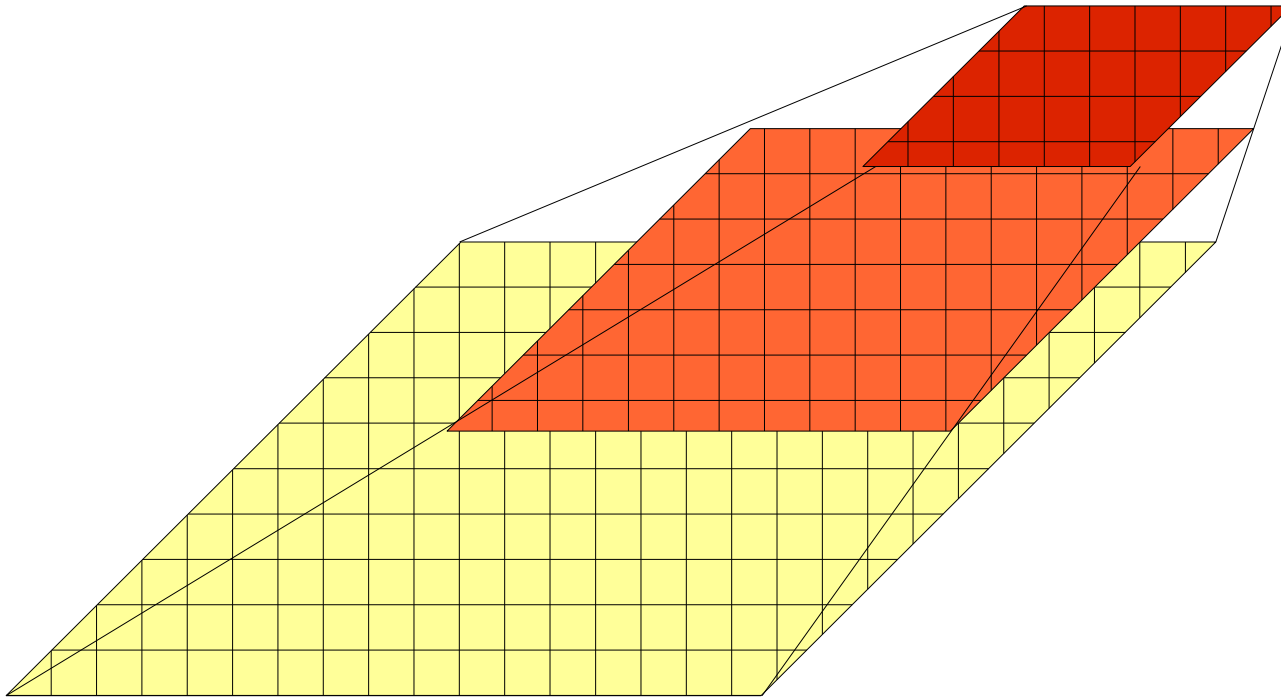
Agenda

- What ?
- Why ?
- How ?

What ?

Tiling: definition

- Tile is a rectangular single piece of an uniform fragmentation of pictorial representation of geographic data that can be accessed by a pair **row**, **column** indices and **scale** value.



Tiling: illustration

Applications Places System ven avr 17, 11:40 Cedric Moulet

Carte La Suisse à vélo - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://map.veloland.ch/ sdi logo

Most Visited Agenda c2c Intranet c2c Demos c2c Fileserver c2c Mail c2c WebAdmin c2c Blogspot Google Analytics Mapfish Blog MapFish Stats https://mgmt0.bgdi.... MapFish Studio

Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options

Camptocamp Mail - Inbox (... ergonomie logo - Recherch... Download Spatial Data Inte... Carte La Suisse à vélo

Bienvenue dans La Suisse à vélo

deutsch
italiano
english

Fermer les thèmes

Carte Photo aérienne Recherche...

Imprimer

La Suisse à vélo

- Itinéraires
 - National
 - Régional
- Etapas
 - Offres locales
 - Vélo de location
 - Service vélo

La Suisse à pied

La Suisse à VTT

La Suisse en rollers

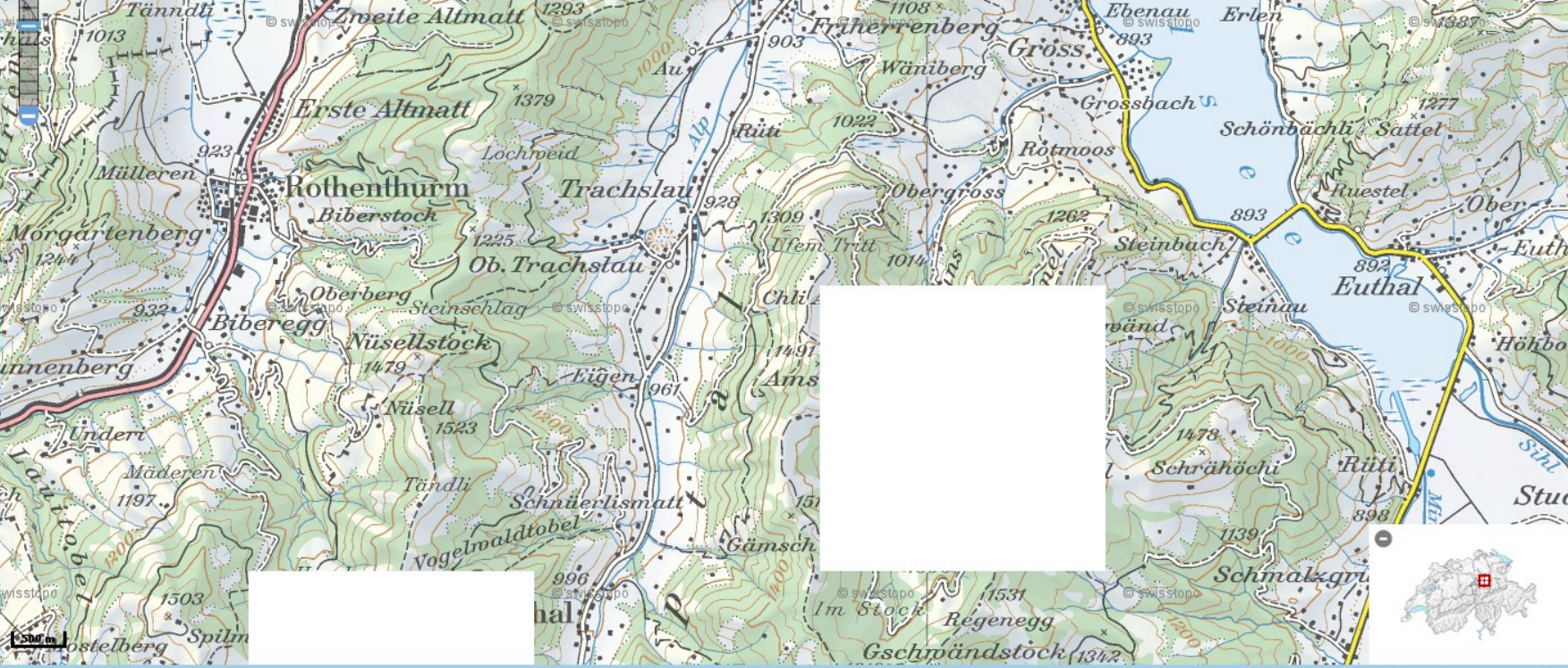
La Suisse en canoë

Trains, bus, bateaux

Lieux

Hébergements

Curiosités



Coordonnées (m): 699215, 216858

Impressum | Copyright & protection des données | Contact Légendes

Transferring data from tile0.bgdi.admin.ch...

Carte La Suisse à vél... seminaire_2009 - File... 2009_Seminaire_We... Documents - File Bro... GIMP Downloads

YSLOW

Why ?

Tiling: pros and cons

Pros:

- Speed, speed, speed !
 - * No cartographic rendering
 - * No data access
- Low (ridiculous...) server load
- Use client and server cache
- Ergonomy
- Web

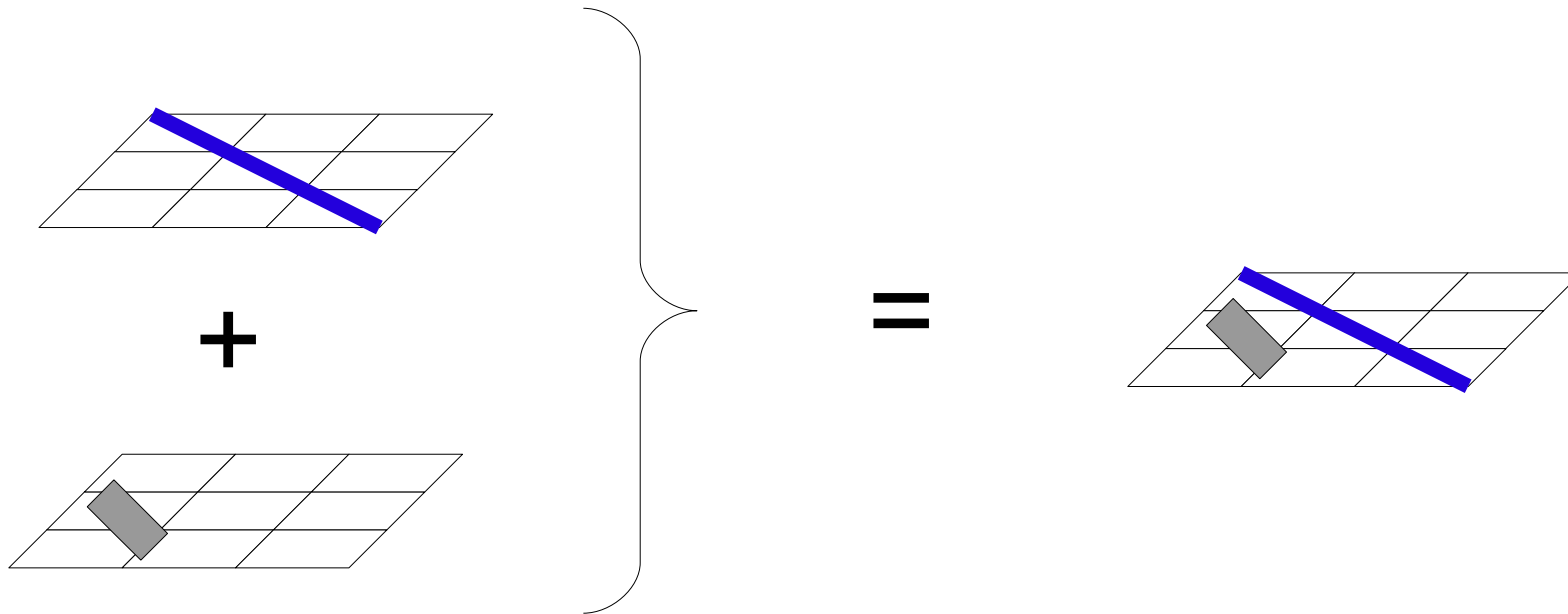
Cons:

- Generation
- Update
- Browser support
- Storage
- Fixed scale levels
- Fixed styling
- Fixed Spatial Reference System (SRS)

Tiling: limitations

WMS mode to avoid

Problem if too many layers client side (max 5-8) -> solution
tile merger



**User wants fast
and reactive
applications**

How ?

WMTS: Web Map Tiling Service

- Status
 - Candidate OpenGIS® Implementation Standard (26.02.2009)
 - Not an OGC standard, based on the OGC Web Services Common Specification
- WMS Brother
- Origin: OSGEO TileMapService and OnEarth Tiled WMS
- 3 requests
 - ServiceMetadata (a **GetCapabilities** response in procedure oriented architecture style) (required) – It allows a client to request and receive back service metadata (or Capabilities) documents that describe the abilities of the specific server implementation. In procedure oriented architecture style this operation also supports negotiation of the standard version being used for client-server interactions.
 - Tile (a **GetTile** response in procedure oriented architecture style) (required) – It allows a client to request a tile map in a similar way that WMS does but in a **more constrained way**.
 - FeatureInfo (a **GetFeatureInfo** response in procedure oriented architecture style) (optional) – It allows a client to request information about a particular pixel of a tile map in a similar way that WMS does.

WMTS: Web Map Tiling Service

- WMTS encoding using HTTP KVP

- <http://www.maps.cat/maps.cgi?service=WMTS&version=1.0.0&request=GetCapabilities>
- http://www.maps.cat/maps.cgi?service=WMTS&request=GetTile&version=1.0.0&layer=coastlines&style=default&format=image/png&TileMatrixSet=WholeWorld_CRS_84&TileMatrix=10m&TileRow=1&TileCol=3&J=86&I=132&InfoFormat=text/xml
- http://www.maps.cat/maps.cgi?service=WMTS&request=GetFeatureInfo&version=1.0.0&layer=coastlines&style=default&format=image/png&TileMatrixSet=WholeWorld_CRS_84&TileMatrix=10m&TileRow=1&TileCol=3&J=86&I=132&InfoFormat=text/xml

- WMTS using SOAP encoding

- Ridiculous in web context (image content in xml...)

- WMTS using RESTful

- {WMTSBaseURL}/{version}/WMTSCapabilities.{format_extension}
- {WMTSBaseURL}/{layer}/{style}/{firstDimension}/{...}/{lastDimension}/{TileMatrixSet}/{TileMatrix}/{TileRow}/{TileCol}.{format_extension}
 - http://www.maps.cat/etopo2/WholeWorld_CRS_84/10m/1/3.png
- {WMTSBaseURL}/{layer}/{style}/{firstDimension}/{...}/{lastDimension}/{TileMatrixSet}/{TileMatrix}/{TileRow}/{TileCol}/{J}/{I}.{InfoFormat_extension}
 - http://www.maps.cat/etopo2/WholeWorld_CRS_84/10m/1/3/86/132.xml

WMS-C: WMS constrained

- WMS-C (Web Mapping Service - Cached) is a **constrained** (non official) profile of OGC WMS that permits servers to optimize their image generation, and allows tiles to be cached at intermediate points.
- A WMS-C service would likely only deliver images for bounding boxes aligned to a given rectangular origin and grid, and only at particular scale levels.
- http://wiki.osgeo.org/wiki/WMS_Tile_Caching /
http://wiki.osgeo.org/wiki/WMS_Tiling_Client_Recommendation

Tile Map Service: TMS

Access by URL: http://tms_blablabla/myTile.png

```
getURL: function (bounds) {  
    bounds = this.adjustBounds(bounds);  
    var res = this.map.getResolution();  
    var x = Math.round((bounds.left - this.tileOrigin.lon) / (res * this.tileSize.w));  
    var y = Math.round((bounds.bottom - this.tileOrigin.lat) / (res * this.tileSize.h));  
    var z = this.serverResolutions != null ?  
        OpenLayers.Util.indexOf(this.serverResolutions, res) :  
        this.map.getZoom();  
    var path = this.serviceVersion + "/" + this.layername + "/" + z + "/" + x + "/" + y + "." + this.type;  
    var url = this.url;  
    if (url instanceof Array) {  
        url = this.selectUrl(path, url);  
    }  
    return url + path;  
},
```


Tile: XYZ rule

Access by URL: http://xyz_blablabla/myTile.png
OpenStreetMap

```
getURL: function (bounds) {
    var res = this.map.getResolution();
    var x = Math.round((bounds.left - this.maxExtent.left) / (res * this.tileSize.w));
    var y = Math.round((this.maxExtent.top - bounds.top) / (res * this.tileSize.h));
    var z = this.map.getZoom();
    var limit = Math.pow(2, z);
    var url = this.url;
    var s = " + x + y + z;
    if (url instanceof Array)
    {
        url = this.selectUrl(s, url);
    }
    var path = OpenLayers.String.format(url, {'x': x, 'y': y, 'z': z});
    return path;
},
OpenLayers.Layer.OSM = OpenLayers.Class(OpenLayers.Layer.XYZ, {
    name: "OpenStreetMap",
    attribution: "Data CC-BY-SA by <a href='http://openstreetmap.org/'>OpenStreetMap</a>",
    sphericalMercator: true,
    url: 'http://tile.openstreetmap.org/${z}/${x}/${y}.png',
    CLASS_NAME: "OpenLayers.Layer.OSM"
});
```

Tile: TileCache rule

Access by URL: http://tilecache_blablabla/myTile.png

```
getURL: function(bounds) {
  var res = this.map.getResolution();
  var bbox = this.maxExtent;
  var size = this.tileSize;
  var tileX = Math.round((bounds.left - bbox.left) / (res * size.w));
  var tileY = Math.round((bounds.bottom - bbox.bottom) / (res * size.h));
  var tileZ = this.serverResolutions != null ?
    OpenLayers.Util.indexOf(this.serverResolutions, res) :
    this.map.getZoom();
  /**
   * Zero-pad a positive integer.
   * number - {Int}
   * length - {Int}
   *
   * Returns:
   * {String} A zero-padded string
   */
  function zeroPad(number, length) {
    number = String(number);
    var zeros = [];
    for(var i=0; i<length; ++i) {
      zeros.push('0');
    }
    return zeros.join("").substring(0, length - number.length) + number;
  }
  var components = [
    this.layername,
    zeroPad(tileZ, 2),
    zeroPad(parseInt(tileX / 1000000), 3),
    zeroPad((parseInt(tileX / 1000) % 1000), 3),
    zeroPad((parseInt(tileX) % 1000), 3),
    zeroPad(parseInt(tileY / 1000000), 3),
    zeroPad((parseInt(tileY / 1000) % 1000), 3),
    zeroPad((parseInt(tileY) % 1000), 3) + '.' + this.extension
  ];
  var path = components.join('/');
  var url = this.url;
  if (url instanceof Array) {
    url = this.selectUrl(path, url);
  }
  url = (url.charAt(url.length - 1) == '/') ? url : url + '/';
  return url + path;
},
```

Tile Generation

- Preseeded or not ?
 - It depends... but preseeded, if possible
- TileForge
 - Usage of cloud computing for tile generation parallelization
 - WMS Server
 - Style
 - Data
 - map.geo.admin.ch: 200'000'000 tiles...
 - geoportail.fr: > 1'000'000'000 tiles...
- Alternative for raster data
 - Specific tools: image « cutter »

Practical considerations

- Tile generation has to be planned
- Data workflow has to be defined
- Use image format to minimize image size (rule of thumb: jpeg for raster, png for vector)
- Create only necessary tiles
- Parallelize tile generation
- Avoid tiling on demand
 - *The WMTS service aim is to be scalable. Therefore, servers must be able to quickly return tiles. The best way to achieve that is use pre-rendered tiles that will no require any image manipulation as times as possible.*
- Tiled WMS is evil
- Use version in request to avoid client cache problems
- Use subdomains to make parallel requests from browser
 - Call always the same subdomain for the same tile (client cache)

Questions ?





Thanks !

Camptocamp

cedric.moulet@camptocamp.com

+33 5 16 57 10 21