Web map printing solutions

Evaluating printing solution for web mapping application, with emphasis on the client side.

Marc Monnerat

2018-03-29

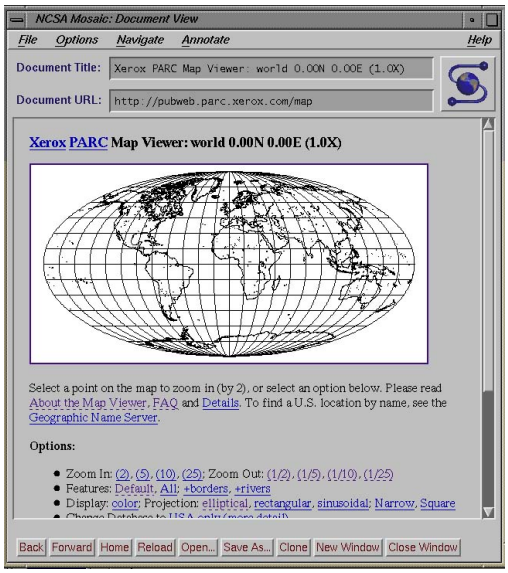
Contents

Abstracts

This document explores various alternatives for generating suitable document for printing to be used in

a web mapping application, with emphasis on the client side.

The mother of all web mapping application, in 1993



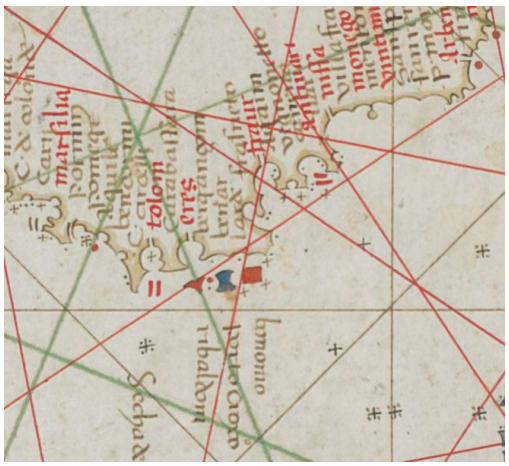
1 Printing maps

In the context on web mapping application, printing is generally seen as as a way to generate a file, like an image or a PDF, suitable to be sent to an office printer. But even in this context, some people may primarly interested in the PDF file, for instance for archiving (PDF/A - ISO 19005 (PDF for Archiving)) or offline use, not a paper impression. Some people use the PDF to get a decent image to be georeferenced, other are well interested in an exact printing at scale, etc.

The advent of 3D data and 3D capable applications as well as the generalisation of vector data also brings new possiblities and challenges for printing.

1.1 Elements to be printed

Since their inceptions, maps have been decorated with more or less useful features (so called *metadata*, *sensu lato*)



1.1.1 Map data

In the GIS world, two things are almost infinite, the number of map projections and the number of data formats.

2D

Raster

Source are generally satelite/aerial imagery, and scan of legacy (printed) maps (historical maps). Vector data are rasterized with Mapserver Served usually served as OGC WMS and WMTS

Vector

In map.geo.admin.ch, there are now about 22 vector layers (for 465 WMTS/WMS layers), loaded as GeoJSON (*e.g.* Water temperature river. Vector files like GPX and KML may also be imported.

Question: rasterize data to print seems obvious, but it is definitely a loss of information and a generally results in a larger file.

3D

- Flattening or not? If so, rasterize or not?
- Render as 3D (export ?) for use with 3D printer or anything else (virtual world, KML, etc.)

Time (4th dimension)

- Data changing over time (Zeitreise)
- Position changing over time (fly path)

How to render: multipage, movie?

Another challenge: mixing data with different rate of change and/or lacking data (not print specific)

1.1.2 Non map data

Non-map data generally added to a document intented for printing.

Logo and corporate identity

Important or not, it gives a professinal look and always generated the most heated discussion.

Disclaimer/copyright

Providing a clear delimitation between which geodatas are from the geoportal and which data are 3rd party. Some reminder of copyright use.

Date

Time of document generation

Title and description

Provide the user the opportunity to give its work a title and small description.

Legend to the map

Some data use complexe symbology and classification, and need an explanation. It could be as *simple* as displaying the classification or add a complexe an lengthy explanation in the case of a geological map as an PDF document.

Scale, scalebar, north arrow

Useful information for the orientation, distance calculation, when using a topographical map or plan.

QRcode, shortlink

Useful to recreate the map in the application online

Table data (reporting)

Some information be easier to display as table. Not used in map.geo.admin.ch, but proposed by some printing applications

File metadata (if applicable)

Metadata embeded in the resulting file (PDF or image) are something useful for search engine if the main goal is to store the PDF.

Grid

Various geographical grids, for orientation, distance calculation and use with GPS.

Information on elements displayed

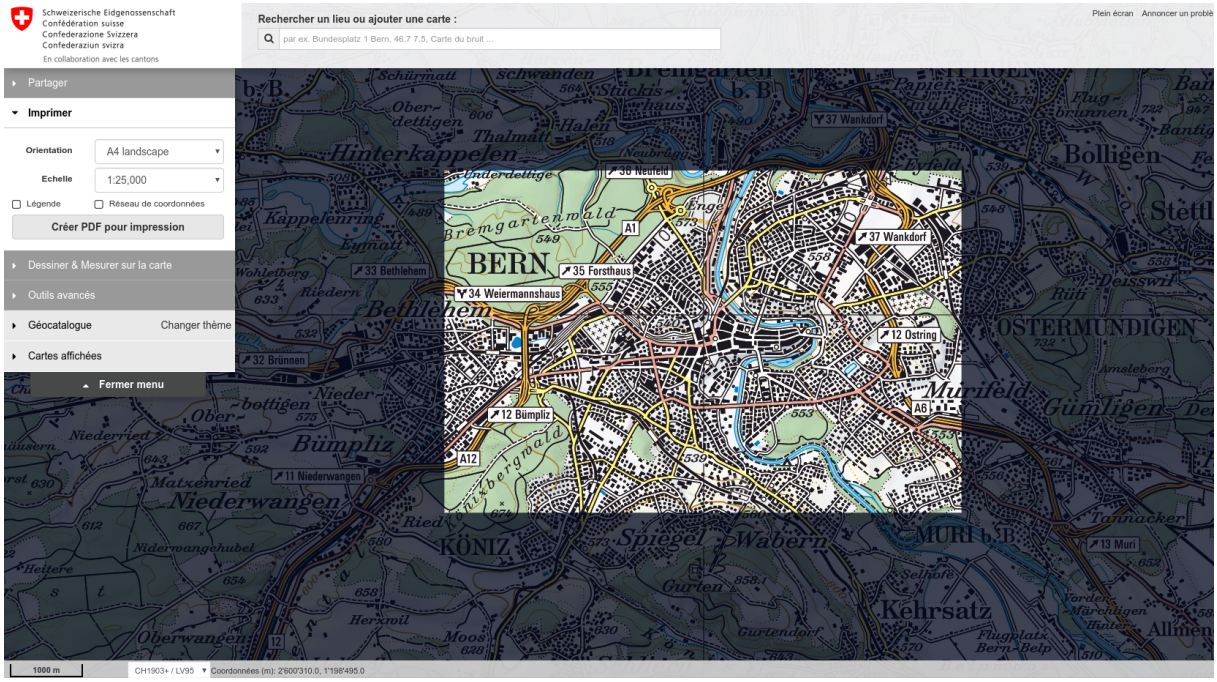
Could be the location of a search term in the simpliest form (Google Map-like marker) or additional information on a highlighted object (tooltips,etc.)

2 Current situation

2.1 map.geo.admin.ch

The Swiss federal geoportal map.geo.admin.ch uses a customed developed server-side printing application (service-print based on a forked of MapFish Print v2 and a Python Flask application to allow for multiserver use and multipage PDF document generation. Currently, it runs in a Docker auto-scaling group, with one server at night and two during the day.

Print dialog and print extent as implemented in map.geo.admin.ch:



The application is basically a proxy, no geodata are store locally. It receives a specification file, describing the extent, the layers (and how to get them), various text and flags, requests all geodata it needs and assembles a PDF document.

2.1.1 Capabilities

- Printing all 2D layers from map.geo.admin.ch, including import GPX/KML and most WMTS/WMS
- Selecting a predefined scale (from 1:500 to 1:2'500'000)
- Generating an A4/A3 PDF 1.3 at 150 DPI (not a technical limitation), portrait or landscape orientation.
- Generating a multipage PDF for time series data, e.g. *Zeitreise* (historical maps, ranging from 1864 to present day, one year per page, about 20 pages)
- Synchronous print for single page, asynchronous for multipage
- 247'198 PDF page generated the last 90 days (500-5'000 per day) and 1'383 multipage print (0 to 50 per day)
- Dockerized application running on a auto-scaling cluster (time-based, two containers during the day, one at night)
- Using an extend version of the standard Mapfish print procotol used by GeoMapFish/GeoExt/-GeoServer
- Printing legends, if needed (classes)

- Printing a *LV95* and *WSG1984* grid if need (the grid is generated by Mapserver)
- Merging PDF legends of complex layers (geology) to the end of the PDF document.
- Logo, QRcode image, shorlink URL, scalebar and scalenumber, various disclaimer and copyright texts.

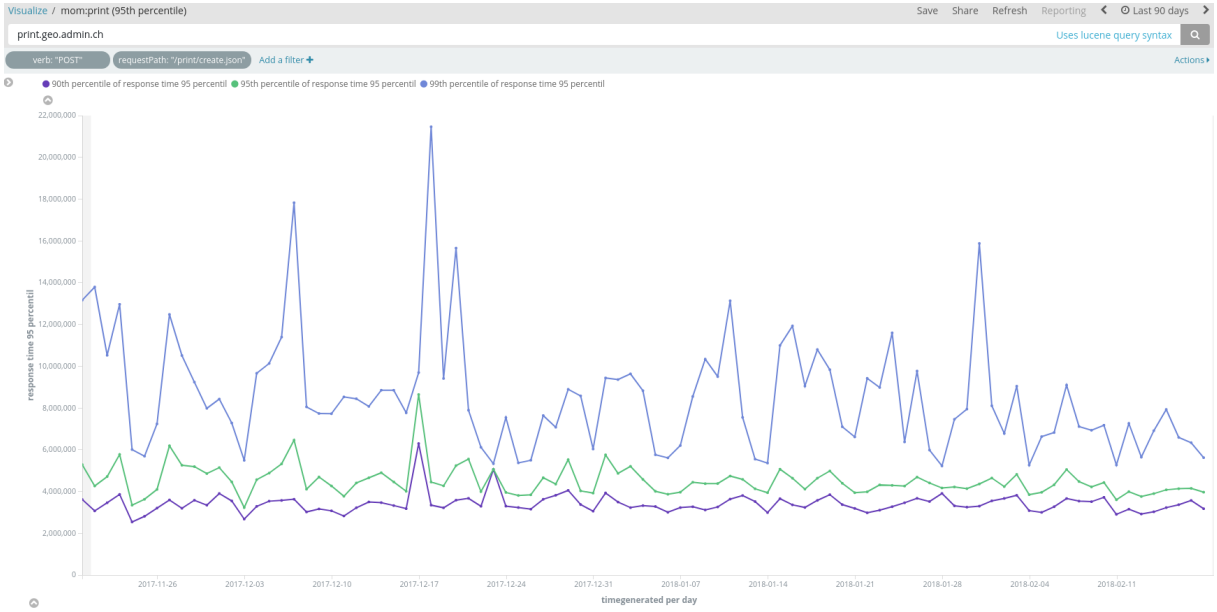
2.1.2 Shortcomings

- Print only 2D map
- Imported WMS and WMTS layers won't be printed if remote server do not support LV95 projection (EPSG:2056). These layers may be still displayed in the application
- WMTS layers are added as indivual tiles to PDF, for every layers (way too much information if some layer are fully opaque) 232 MB for Zeitreise only, 376MB (25 pages)
- Very limited support for Vector. Luckily, imported vector do not support many styles.
- Symbols size for Raster layer (Vector symbol are adapted to print resolution)
- MapFish v2 is not developped anymore. swisstopo is patching a fork.
- Support for retry and multiserver is sub-optimal.
- Not 100% WYSIWIG. The print extent is always correct, but some people are confused because both the *ch.swisstopo.pixelkarte-farbe* and the *ch.swisstopo.pixelkarte-grau* are combination of maps of various scales which are not necessarly printed.

2.1.3 Performances

Averall print jobs

The 95th percentile for all print jobs are between 3.7 and 5.0 s (only time for generation, without download)



https://kibana.bgdi.ch/goto/3566fae450682eea244826f2ec49e477

Error rate: last month 1’082 errors for 97’988 success (about 1.09%)

Simple print job (cron)

Printing a standard A4 landscape page at 1:25’000

Generation time (POST to json response): 1.21 ? 0.05 s (n=4578, every 5 minutes)

Errors: 27 (2 DNS failures!) out of 8294 jobs (0.32%) (about one month)

Pingdom

Same spec file as above sent every 5 minutes from Pingdom (many locations within Europe)

1.166 s average response time (time to get JSON response) 1.279 s max, 1.104 s min response time 7 outages totalising 35 minutes uptime 99.91%

Pingodm report

Testing

Many spec file examples are availables on github at mapfish-print-examples

A bash script may target the remote print server, tomcat only (no multiprint) or the local jar file.

```
./test_print_server.sh remote lv95_versoix_10000_draw
```

Examples PDF

Some PDF output from print.geo.admin.ch

A4 1:25’000 150 dpi

Non standard print, at 96 and 300 dpi, for comparison:

A4 1:25’000 96 dpi

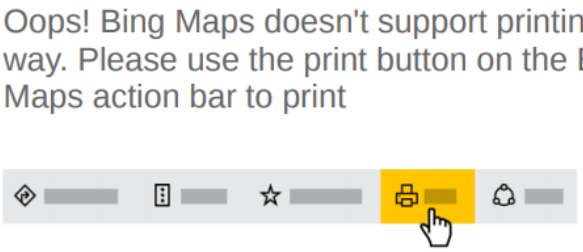
A4 1:25’000 300 dpi

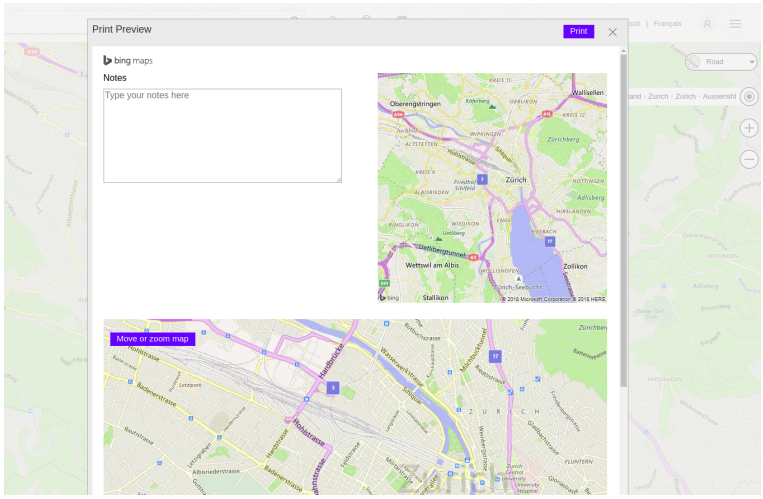
2.2 Leading web map application

2.2.1 Bing

Bings Maps

is not supported

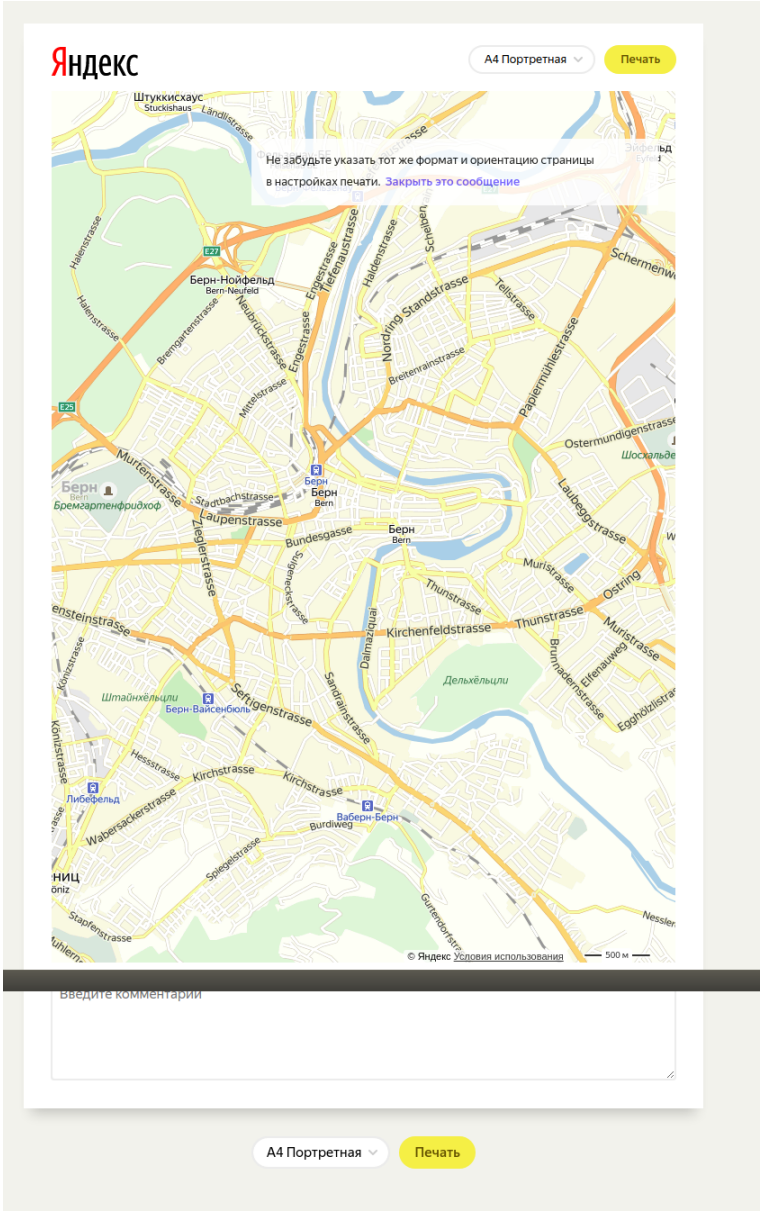




2.2.2 Yandex

Yandex Maps

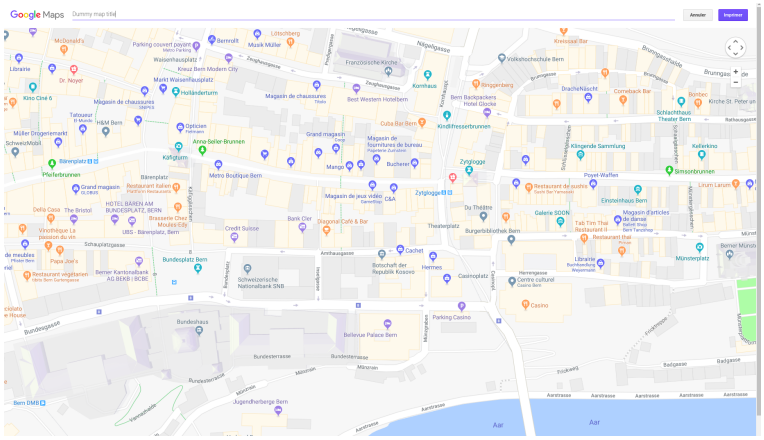
Via or print buttom. Preview in new tab. Possibility to change page size and orientation



2.2.3 Google Map

Google Maps

Printing via ,possibility to add a title



2.2.4 Here

Here

2.2.5 CalTopo

Caltopo

Print to Geospatial PDF

```
gdalinfo img/caltopo.pdf

```

```
Driver: PDF/Geospatial PDF
Files: img/caltopo.pdf
Size is 2338, 1654
Coordinate System is:
PROJCS["WGS 84 / World Mercator",
    GEOGCS["WGS 84",
        DATUM["WGS_1984",
            SPHEROID["WGS 84",6378137,298.257223563,
                AUTHORITY["EPSG","7030"]],
            AUTHORITY["EPSG","6326"]],
        PRIMEM["Greenwich",0,
            AUTHORITY["EPSG","8901"]],
        UNIT["degree",0.01745329251994328,
            AUTHORITY["EPSG","9122"]],
        AUTHORITY["EPSG","4326"]],
    UNIT["metre",1,
        AUTHORITY["EPSG","9001"]],
    PROJECTION["Mercator_1SP"],
    PARAMETER["central_meridian",0],
    PARAMETER["scale_factor",1],
    PARAMETER["false_easting",0],
    PARAMETER["false_northing",0],
    AUTHORITY["EPSG","3395"],
    AXIS["Easting",EAST],
    AXIS["Northing",NORTH]]
Origin = (739770.531363048939966,5963816.915579564869404)
Pixel Size = (93.015211288114500,-92.730207471186560)
Metadata:
  DPI=200
  NEATLINE=POLYGON ((751397.432875239 5835941.9567583,751397.432875239
      5952112.28762264,945437.500334236
      5952112.28762264,945437.500334236 5835941.9567583,751397.432875239
       5835941.9567583))
  PRODUCER=CalTopo
Corner Coordinates:
Upper Left  (  739770.531, 5963816.916) (  6d38'43.70"E, 47d19'29.16"N)
Lower Left  (  739770.531, 5810441.152) (  6d38'43.70"E, 46d22'46.49"N)
Upper Right (  957240.095, 5963816.916) (  8d35'56.52"E, 47d19'29.16"N)
Lower Right (  957240.095, 5810441.152) (  8d35'56.52"E, 46d22'46.49"N)
Center      (  848505.313, 5887129.034) (  7d37'20.11"E, 46d51'15.36"N)
Band 1 Block=2338x1 Type=Byte, ColorInterp=Red
  Mask Flags: PER_DATASET ALPHA
Band 2 Block=2338x1 Type=Byte, ColorInterp=Green
```

```
  Mask Flags: PER_DATASET ALPHA
Band 3 Block=2338x1 Type=Byte, ColorInterp=Blue
  Mask Flags: PER_DATASET ALPHA
Band 4 Block=2338x1 Type=Byte, ColorInterp=Alpha
```

2.2.6 MangoMap

MangoMap

[MangoMap generated PDF(pdfs/Fire Districts - Interactive Web Map.pdf)

3 Approaches

3.1 CSS: @media print

@media in MDN web docs

```
> The @media CSS at-rule associates a set of nested statements, in a
  CSS block
> that is delimited by curly braces, with a condition defined by a
  media query.
> The @media at-rule may be use
```

3.1.1 Browser print function

Easiest solution, used by Google Map, Bing Map. WYSIWIG

All it takes is to define a CSS stylesheed for print (for inspiration Paper CSS).

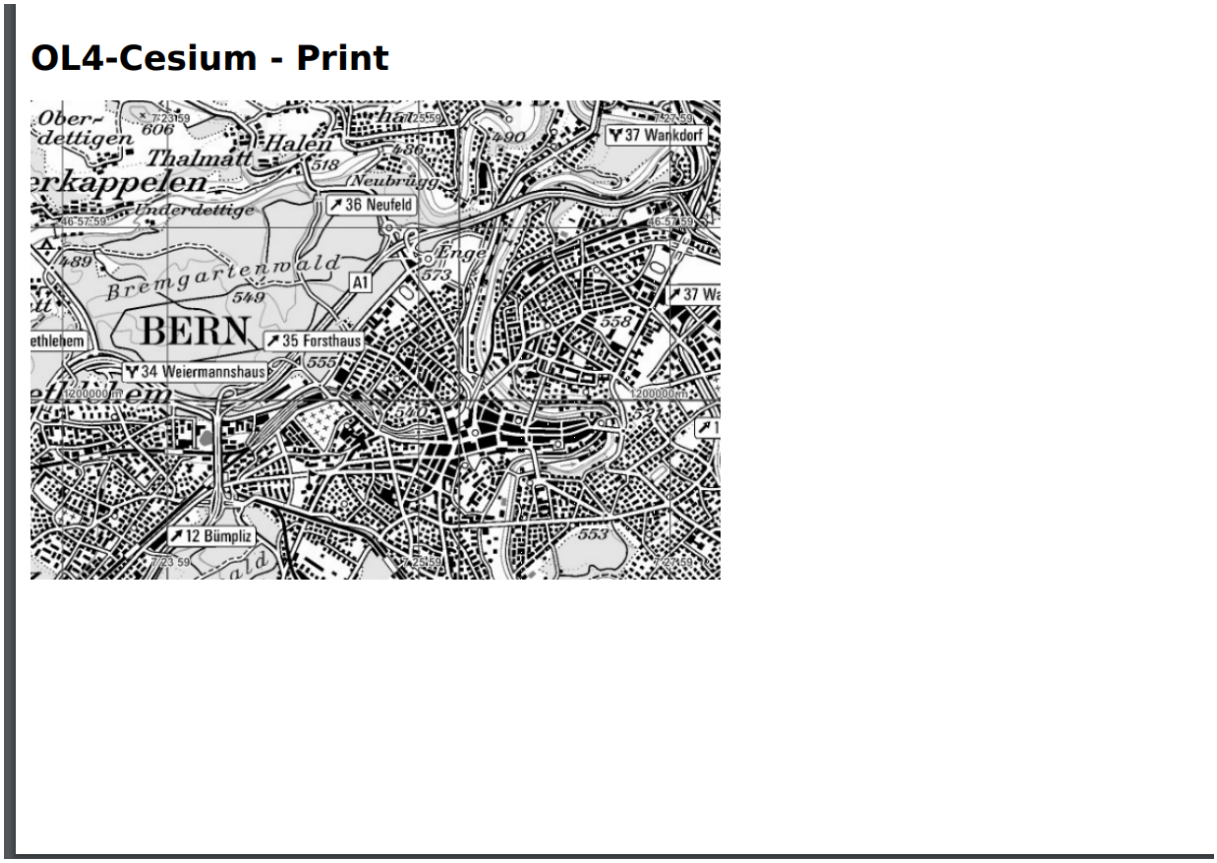
HTML5, *toBlob()*, *toDataURL()*

A CSS at-rule @page to define page-specific rules when printing web pages, such as margin per page and page dimensions. Not supported by Safari (all versions).

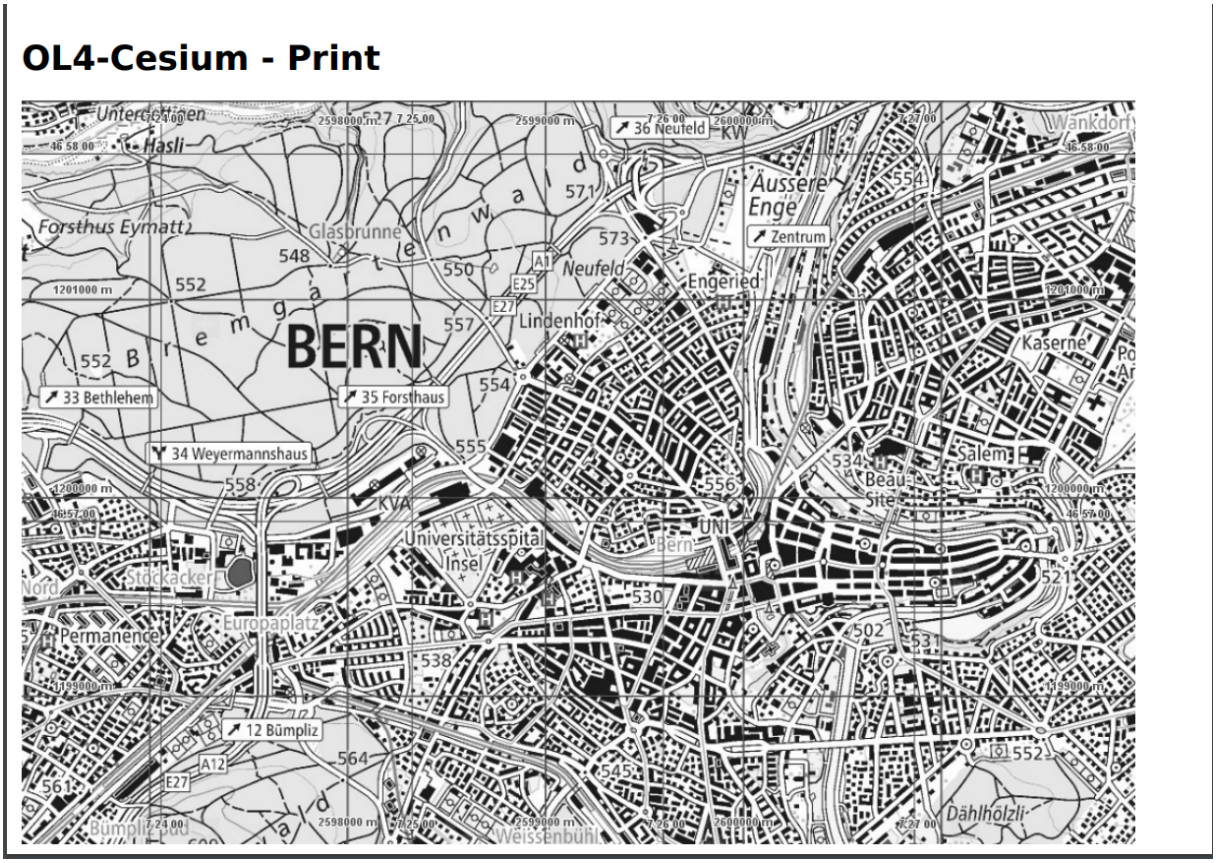
```
page[size="A4"][layout="landscape"]  {
    width: 21cm;
    height: 29.7cm;
}
```

But, as we do not have any influence on the size of the image generated in the browser, we have to make trade off in term of scale, print resolution and image aspect ratio. To print at 150dpi on A4, we need an image of 1750x1250 pixel approximatly, which is OK on a deskop computer, but probably not on a laptop or tablet.

Canvas of 650x450px not fitting an A4 page



Canvas of 1050x750px fitting an A4 page



A bigger display will be cropped

With a map covering the whole screen (`width: 100%; height: 100%`), the challenge is to get an absolute width and height, to be fitted in the print page.

Challenges

- Print to many different paper sizes and orientations
- Make browser recognize size and orientation
- Print quality, on smaller display *e.g.* Laptop with 14” screen

Works well on Chrome, Firefox and Opera (only 2D for the latter), when preview is activated.

Live demo

3.1.2 PDF generation in client

PDF.js (Open Source) and jsPDF (commercial), PSPDFKit (comercial)

3.1.3 Rendering PDF on a server

Generating PDF from XML/HTML and CSS - A tutorial and showcase for CSS Paged Media

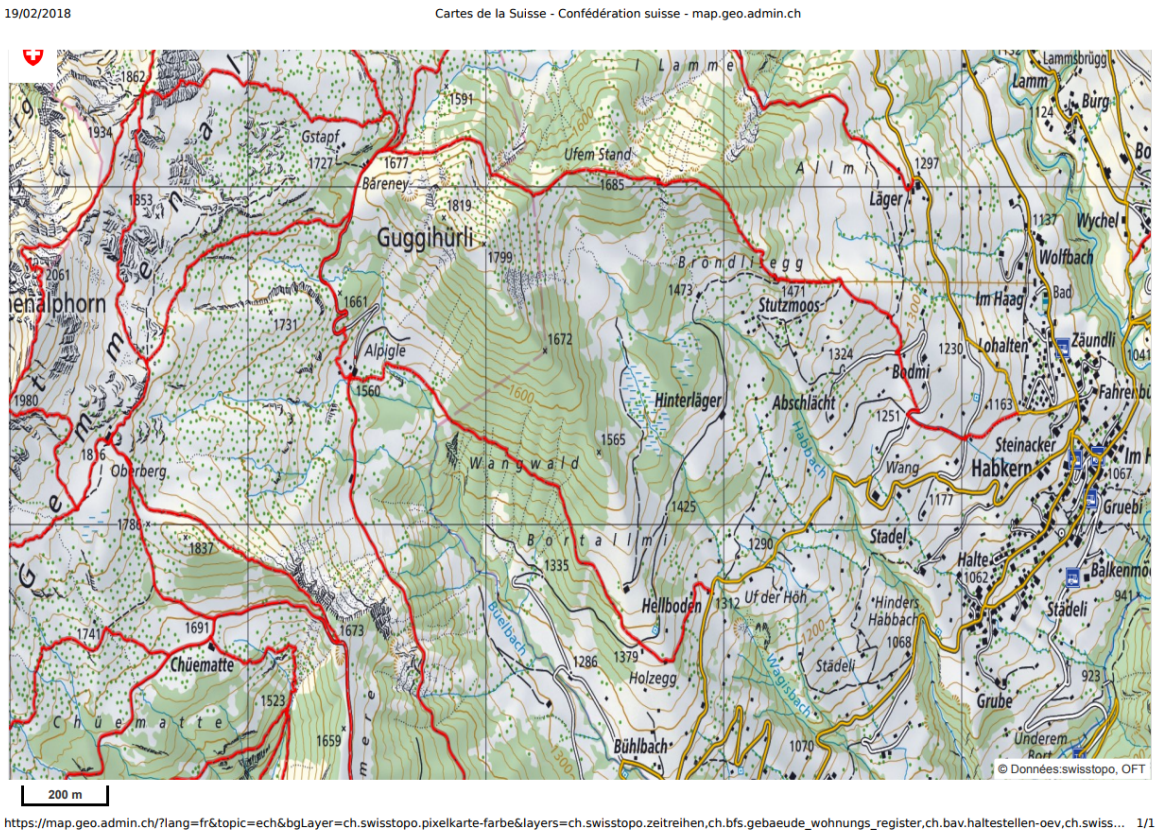
Print CSS rocks

Rendering HTML page to PDF using the full CSS Page media standard with commercial tool like PDF-reactor, PrinceXML, Antennahouse CSS Formatter or DocRaport. Some are based on XSL-FO other on Webkit.

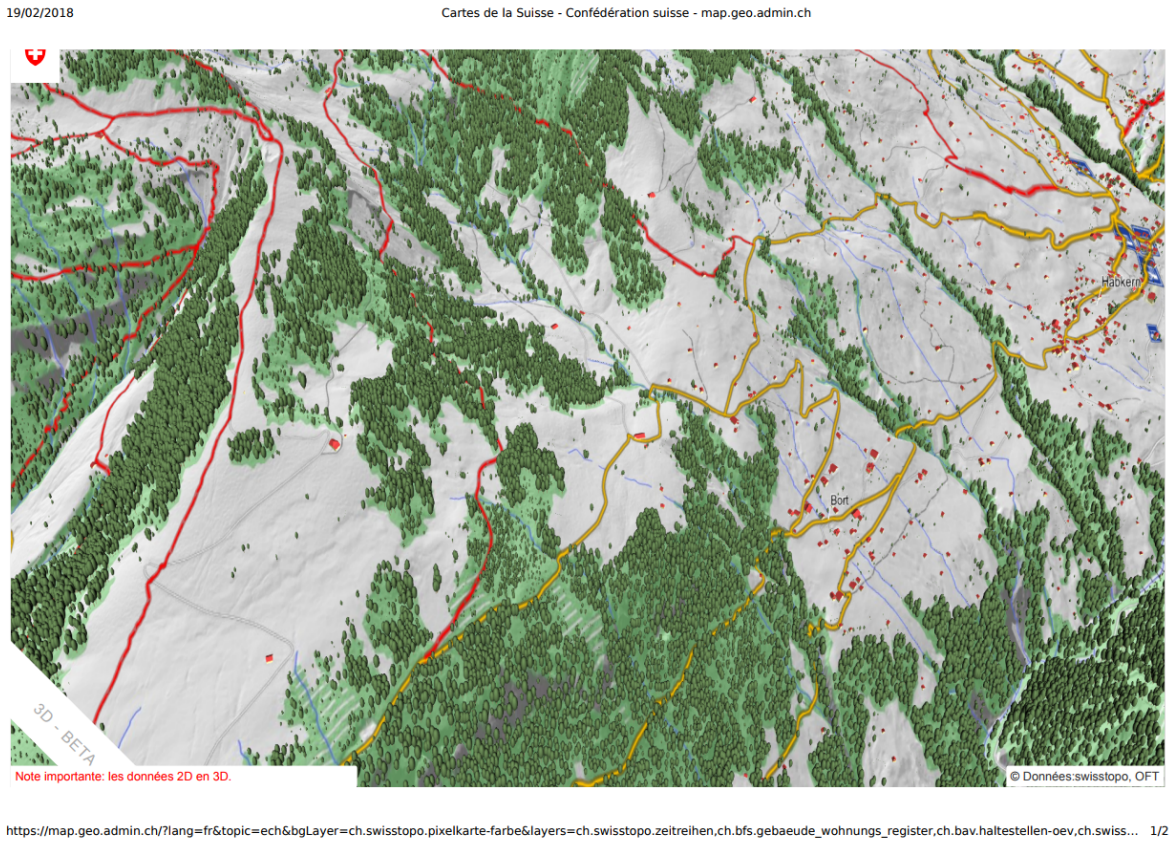
3.2 Export canvas as image

3.2.1 Export canvas as image

Printing in 2D



Printing in 3D



Live demo. This is OL-Cesium, you may toggle 2D/3D.

3.2.2 Export image + popup

Create a simple page in a new popup window, with additional elements, and copy the map as an image HTML template for printing

3.2.3 Better resolution

An approach to gain control over the generated image is to recreate a hidden canvas to generate an image that suit the need. This is the approach of the “High DPI print” for Mapbox (print-maps).

Maybe OK, for simple 2D applications, more difficult for complexe 3D applications with user defined content. What about performance ?

3.2.4 Export canvas to a vector format

As SVG or PDF (what libraries?)

3.2.5 Examples

Browser print

Browser print

Mapbox GL using print-maps by Matthew Petroff.

Cesium and swisstopo terrain

OL-Cesium in 2D and 3D mode

OL4-Cesium - Print

Leaflet Easy Print Demo

3.2.6 Discussions

Shortlinks

Mabe useful to recreate the application elsewhere, when needed.

Browser support

But 3D and WebGL is forcing to use modern browsers.

Workload is offloaded to clients

No more server

Raster only

Raster only, but smaller images

WYSIWIG

Especially in Chrome with the print preview function, the user sees exactly what is going to be printed. A print preview may also be impletemented for other browser.

No special code for rendering

If we considere CSS is no code, yes…

Resolution, aspect ratio, screen size, pixel density

Basically, your get the canvas at dispostion and try to fit in an A4 page. A small screen means a poors print quality, while a large screen means a decent or good print quality.

Some are trying to get a lager image by recreating a large hidden map canvas. It may work in 2D, but will consume much resource in 2D. Remember, A4 at 150 dpi is 1750x 1250px, and A3 at 150 DPI is 1750x2480px.

Pixel density

A paper map is read at 25cm, screen at 50-70cm 14’“and 75 - 105 cm for 20/21”’ The maximal resolution is about 300 dpi at 25cm, 152 dpi at 50cm and 76 dpi at 100cm. So while a resolution of about 100dpi is acceptable on a desktop, it makes no sense to print above 300dpi for instance.

Table 1: Relation between paper size (mm) and image size (pixels). *dpi* (dot per inch)

Paper	96 dpi	150 dpi	300 dpi
A5 (210x148mm)	793x563	1240x880	2408x1760
A4 (297x210mm)	1112x793	1753x1240	3507x2408
A3 (420x297mm)	1587x1112	2408x1753	4360x3507

Table 2: PPI for some smartphone and tablets

Device	Screen (in)	Resolution (px)	PPI
galaxy S8 (2017)	6.2	2960x1440	530
galaxy S9 (2018)	5.8	2960x1440	567
ipad (2017)	9.7	2048x1536	263
ipad pro (2017)	12.9	2732x2048	264
iphone 5S (2013)	4.0	1136x640	325
iphone 6 (2014)	4.7	1334x750	325
iphone 6s (2014)	4.7	1334x750	325
iphone 7 (2016)	4.7	1334x750	325
iphone 8 (2017)	5.5	1920x1080	400
iphone X (2017)	5.8	2436x1125	462

Conserving the scale

The next three images have the same spatial extent (2500 meters wide), but have different sizes. If we want to keep the scale, here 1:25'000, we have to print the image of 100mm, regardless of the pixels of the image. The result is a low quality for smaller images.

Image of 377x188 pixels, printed @96 dpi (100x50mm)

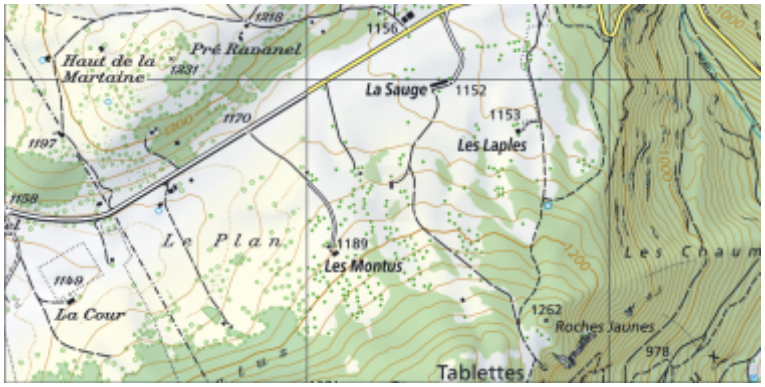


Image of 590x295 pixels, printed @150 dpi (100x50mm)

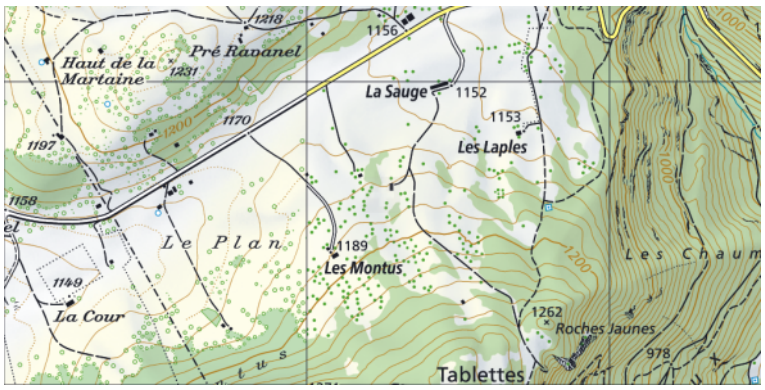
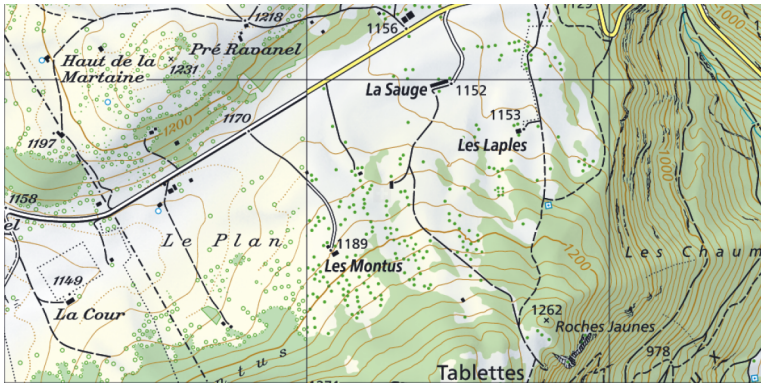


Image of 1181x590 pixels, printed @300 dpi (100x50mm)



Conserving the print resolution

The next three images have the same spatial extent (2500 meters wide), but have different sizes. In this example, we want to keep a good print quality (300 dpi), so we have to decrease the size of the final printed image. The result is loosing the original map scale of 1:25'000.

Image of 377x188 pixels, printed @300 dpi (32x16 mm), print scale is about 1:78'125

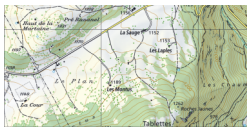


Image of 590x295 pixels, printed @300 dpi dpi (50x25 mm), print scale is 1:50'000

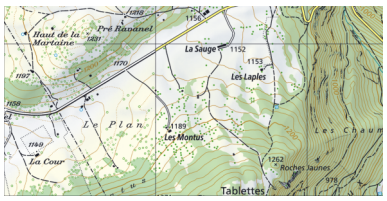
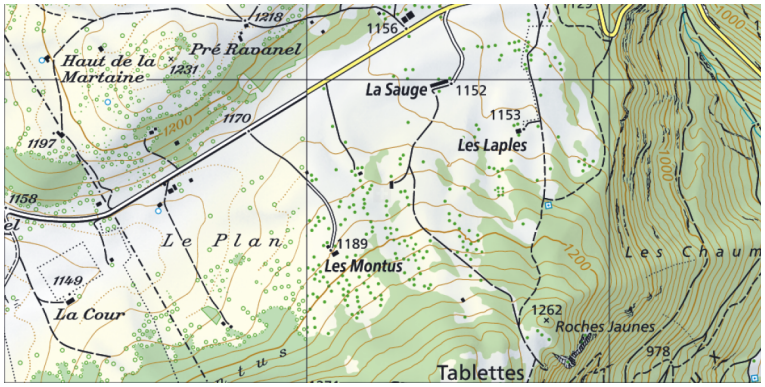


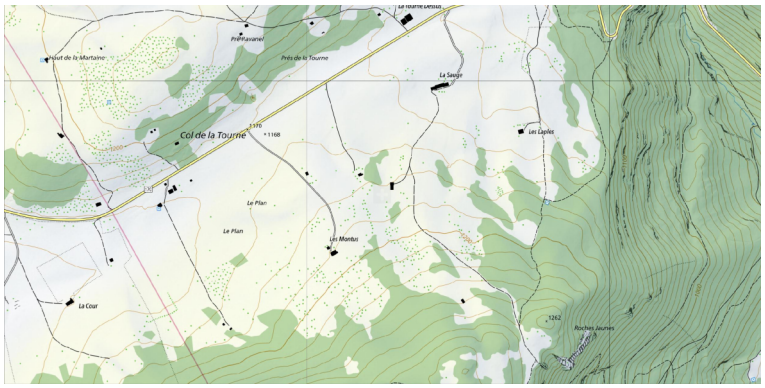
Image of 1181x590 pixels, printed @300 dpi (100x50mm), print scale is 1:25'000



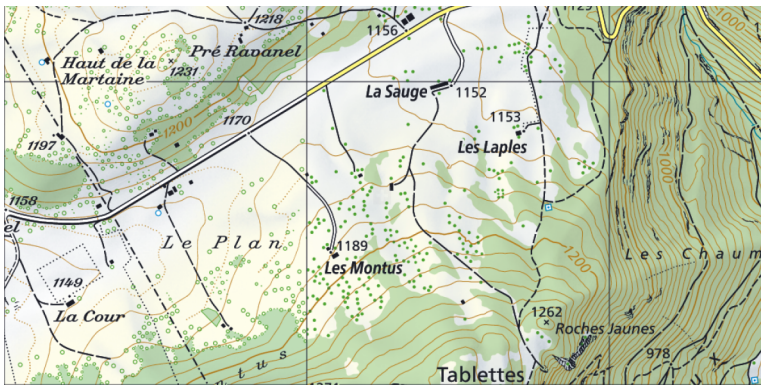
Effect of style

In this example we use the *Landeskarte/Pixelkarte* because their style is targeted for a very specific scale: 1:10'000, 1:25'000 and 1:50'000. The three following picture are printed at 300dpi for a scale of 1:25'000.

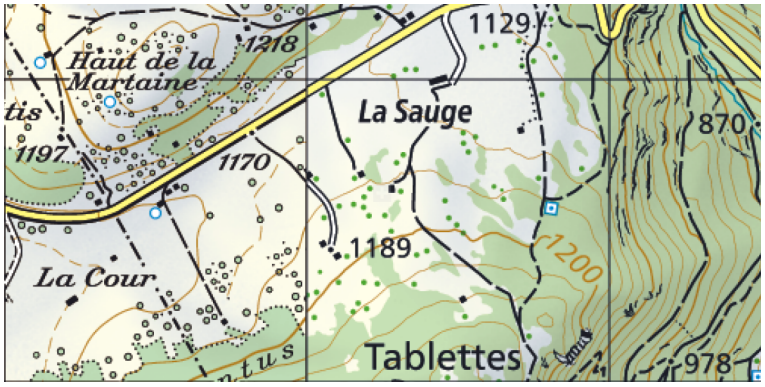
LK10, designed for 1:10'000 Labels and features are very small, hard to read.



PK25, designed for 1:25'000 Labels and features are easily readable, making a joyful impression.



PK50, designed for 1:50'000 Features and labels are big. You may clearly have more information.



3.2.7 PDF compression

Compression with jsPDF of an A4 PDF page containing a single image of 12'236'896 bytes (pk25.noscale-3271-2244-@300.png)

The following compressions are tested: `NONE`, `FAST`, `MEDIUM` and `SLOW`. For comparison, the uncompressed PDF file has been compressed with `ps2pdf`. Five repetitions for each methods.

Table 3: Time needed to compress an image at 300 dpi

Compression	Time (s)	PDF (bytes)	Size (%)
NONE	6.99 ? 0.10	29363689	100.0
FAST	8.84 ? 0.08	11978739	40.8
MEDIUM	11.61 ? 0.41	11282507	38.4

Compression	Time (s)	PDF (bytes)	Size (%)
SLOW	67.91 ? 2.37	10469602	35.6
ps2pdf	< 0.5	1227882	4.18

Another example with an image targeted for 150 dpi 3’609’601 bytes(pk25.noscale-1635-1122-@150.png)

Table 4: Time needed to compress an image at 150 dpi

Compression	Time (s)	PDF (bytes)	Size (%)
NONE	2.01 ? 0.07	7341071	100.0
FAST	2.91 ? 0.20	3554307	48.4
MEDIUM	3.69 ? 0.35	3465019	47.2
SLOW	12.33 ? 2.24	3290648	44.8
ps2pdf	< 0.25	400244	5.4

Templating

Performance

Hard to find when the page is fully rendered…

Security

Cross-Origin Resource Sharing (CORS) and Content-Security-Policy (CSP)

3.2.8 Conclusion

If the added value is not much more than what a screenshot offers, there not point to provide this functionality

3.3 Server-side rendering of HTML

Either vector or raster.

3.3.1 Rasterizing of canvas

Challenges:

- Pretty slow
- How do you know the map is fully loaded and rendered?

SlimerJS (Firefox based)

- WebGL support
- Not truely headless (?)

PhantomJS (WebKit based)

- No WebGL support

Headless Chrome

Getting Started with Headless Chrome

Chromedriver

Print services

Several commercial services are able to generated PDF from web pages:

- PDFreactor
- PrinceXML
- Antennahouse CSS Formatter
- DocRaport

3.4 Server side

3.4.1 Rasterizing/exporting data

Several tools are able to rasterize vector data or export them to another vector format (PDF, SVG). Some of them may deal both with vector and raster geodata, other not.

Mapserver

The grand-mother of all cartographic server: Mapserver.

Cartographical Symbol Construction with MapServer

Mapserver is hower able to render to vector format like SVG and PDF (when using PDFlib)

Mapnik

Map Markup Language file) is a YAML or JSON Mapnik XML, Cascadenik MML, Carto MML Mapnik was the original tool to generate OSM tiles for the so-called slippy map. Mapnik is also able to genrate PDF when compiled with Cairo.

It uses Mapnik XML as configuration, also for styles. Cascading Sheets Of Style for Mapnik aka Cascadenik is a preprocessor for Mapnik, using cascading style sheet for map definition.

CartoCSS is a language for map design. It is similar in syntax to CSS, but builds upon it with specific abilities to filter map data and by providing things like variables. It targets the Mapnik renderer and is able to generate Mapnik XML or a JSON variant of Mapnik XML.

It is now deprecated (The end of CartoCSS) by its parent company, Mapbox.

Tileserver GL

A Mapbox Style Specification is a document that defines the visual appearance of a map: what data to draw, the order to draw it in, and how to style the data when drawing it. A style document is a JSON object with specific root level and nested properties. This specification defines and describes these properties.

Vector and raster maps with GL styles. Server side rendering by Mapbox GL Native. Map tile server for Mapbox GL JS, Android, iOS, Leaflet, OpenLayers, GIS via WMTS, Tileserver GL

Safe FME

Safe FME is a plateform to convert from and to any GIS data formats and automomate workflows.

Creating PDF Cartographic Output

The Adobe Geospatial PDF Reader/Writer allows FME to read and write Adobe?? Portable Document Format (PDF) with vector drawings and geospatial information.

3.4.2 Other tools

GDAL/Rasterio

OWSlib

3.5 Print server

3.5.1 Mapfish print

The purpose of Mapfish Print 3 is to create reports that contain maps (and map related components) within them. The project is a Java based servlet/library/application based on the mature Jasper Reports Library.

3.5.2 Geoserver print

The Geoserver Printing Module allows easy hosting of the Mapfish printing service within a GeoServer instance. The Mapfish printing module provides an HTTP API for printing that is useful within JavaScript mapping applications. User interface components for interacting with the print service are available from the Mapfish and GeoExt projects.

3.6 Mixed approach

Some elements are rendered client-side and then sent to the server, or only some operation are done server-side.

3.7 Integrated approach

Tools like QGis and ArcGIS used to configure the layers. Configuration files, including styles, are used in the web application. As both client and server are using the same configuration, printing may be done server-side.

QGis Print Composer

3.8 Integrated print tools

Need more testing…

3.8.1 swisstopo Print Flex

No information there…

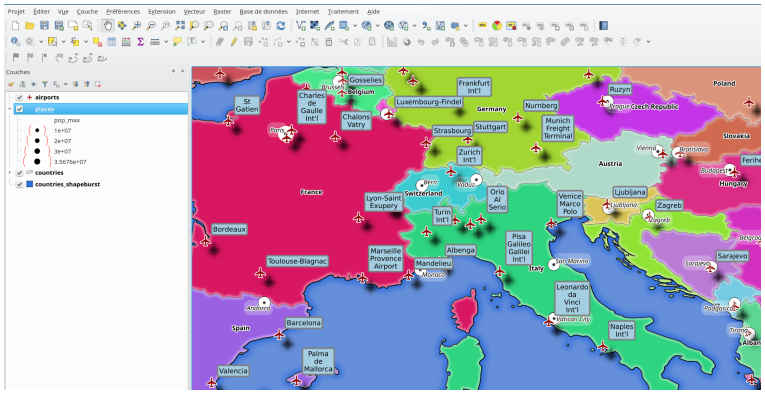
3.8.2 QGIS print

Quantum GIS a fairly advanced print capability

QGIS Server

One interesting aspect of QGis Server is that it may use QGis Desktop Project (.qgs file) as a source of data for server

A simple project in QGis desktop…



…and the same project served as a WMS image:

QGis print composer

Qgis has an advanced Print composer to generate image or PDF export. The composer may add many items to a composition:

- Map, inclusive grids, rotation,

- Labels
- Images, inclusive a north arrow
- Scalebar
- Legends
- Shapes and arrows
- Table items (attributes)
- HTML frame

Code to print load a project and generate a PDF with a custom template

```
#!/usr/bin/env python
#-*- coding: utf-8 -*-

import sys
import os
from qgis.core import (
    QgsProject, QgsComposition, QgsApplication, QgsProviderRegistry,
    QgsComposerMap, QgsRectangle)
from qgis.gui import QgsMapCanvas, QgsLayerTreeMapCanvasBridge
from PyQt4.QtCore import QFileInfo
from PyQt4.QtXml import QDomDocument

gui_flag = True
app = QgsApplication(sys.argv, gui_flag)
app.setPrefixPath("/usr", True)

dir_path = os.path.dirname(os.path.realpath(__file__))

# Make sure QGIS_PREFIX_PATH is set in your env if needed!
app.initQgis()

project_path =  os.path.join(dir_path, 'data', 'world', 'world.qgs')
template_path = os.path.join(dir_path, 'world.qpt')

def make_pdf():
    canvas = QgsMapCanvas()
    # Load our project
    QgsProject.instance().read(QFileInfo(project_path))
    bridge = QgsLayerTreeMapCanvasBridge(
        QgsProject.instance().layerTreeRoot(), canvas)
    bridge.setCanvasLayers()

    template_file = file(template_path)
```

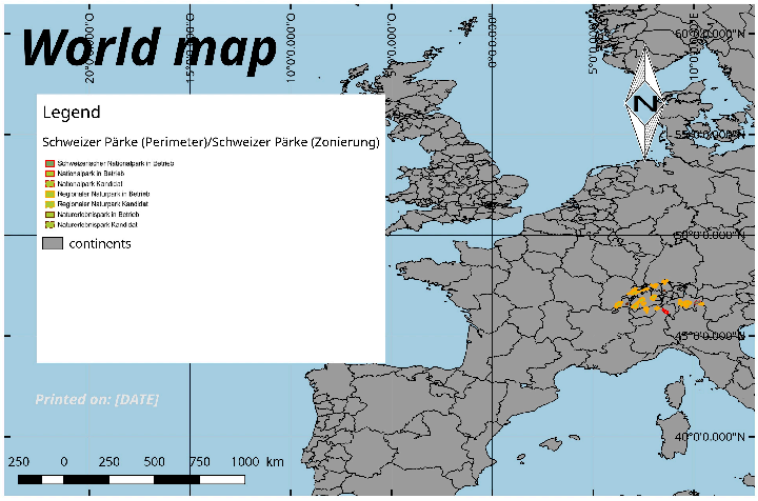
```
    template_content = template_file.read()
    print template_content
    template_file.close()
    document = QDomDocument()
    document.setContent(template_content)
    composition = QgsComposition(canvas.mapSettings())

    composition.loadFromTemplate(document, {})

    map_item = composition.getComposerMapById(0)
    map_item.setMapCanvas(canvas)


    map_item.zoomToExtent(QgsRectangle(-6,39,16,51))
    # You must set the id in the template
    legend_item = composition.getComposerItemById('legend')
    legend_item.updateLegend()
    composition.refreshItems()
    composition.exportAsPDF('report.pdf')
    QgsProject.instance().clear()


make_pdf()
```

See 

3.8.3 ArcGis print

ArcGIS Enterprise includes a geoprocessing service called *PrintingTools*. Web applications invoke the *PrintingTools* service and get a printable document in return (see Printing in Web application):

TODO: needs more investigation

4 Discussion

Table 5: Summary of features

Parameter	Image export	Ctr-P	Export+HTML/PDF	Server-side
WYSIWIG	+	+/-	+/-	+/-
Scale conservation	-	-	+	+
Resolution	+/-	-	+/-	+
Plateform	+	-	+	++
PDF	0	0	+/-	++ (if $$)
Movie	-	-	-	+
Vector export	0	0	-/+ (?)	+
Legacy/low end	+	+/-	+/-	+
Mashup	0	+/-	+	++
User style	++	++	++	+
3D	++	++	+/-	+/-
Side-by-side comaprison	++	++	++	–

Table 6: Summary for adminsitration

Parameter	Image export	Ctr-P	Export+HTML/PDF	Server-side
External computer power	++	++	++	–
Development effort	++	+	+/-	++
Time to result	+	+	+/-	+/-

Parameter	Image export	Ctr-P	Export+HTML/PDF	Server-side

4.1 Scale and resolution

- Is a simple image export enough?
- Should it printed as a PDF, at a given size and/or resolution? What quality?
- Is scale important? Difference in resolution may be important between screen and paper.

4.2 Plateform support

Should not be an issue, as a 3D web application run only on quite recent web browser.

4.3 WebGL/canvas

4.4 Vectors in PDF

PDF is basically a vector format. Must vector layers rendered as vector or must be rasterized?

4.5 Support for user defined style

With vector data, it is easy to apply user defined style, though defining styles for complexes dataset is a daring undertaking.

4.6 PDF standard

- What PDF version? Most client-side libraries are PDF 1.3 (Acrobat 4.x) capable.
- Accessibility considerations?
- Support of PDF/2, PDF/XPDF or PDF/A? No, open source library available. For Python, see PDFlib.
- GeoPDF
- – Long term archiving PDF/A - ISO 19005 (PDF for Archiving)

4.7 Legacy client

For some legacy client, we must eventually also provide raster tiles (from vector layer)? Or anyway, as WMTS. This implies a solution for style management, which could be used for printing

4.8 Vector style definition

How are the styles for vector layer defined? Where? And how it is applied? What standard.

Currently, all style are defined in Mapserver's Mapfile definition.

4.9 Complexe symbols

Especially complexe labels placement is hard (what to render, at what scale, collision avoidance). Advanced label placement is done server-side.

4.10 Mashup

Maps should be mashed up with other sorts of infos (diagram, plot, data tables)

4.11 Movie

Zeitreise, Fly along path, etc. Large PDF files?

4.12 Compute power

Externalize compute power on client or not?

4.13 Performance

Synchronous or asynchronous printing (server-side printing)

4.14 WYSIWIG

Symbol scaling, which LK to use, grid display, etc.

4.15 Grid

Grid for various projection system.

4.16 Projection issue

One selling point of vector tiles and 3D is that 2D in only a special case when pitch is 0 (or 90?). But the 3D world is a WGS1984 only world, which translate in the infamous Equirectangular (or plate carr?e)projection. Using the same projection for printing as in the browser (projection of the data, target projection). When using a Webmercator, deformation, scale, with LK

4.17 Rasterize as a data protection tool

Rasterizing vector data was also a way to *protect* the more valuable original vector datasets (_e.g._WMS).

4.18 Export to other format

Only PDF, or other format as well (SVG, GeoTIFF)

4.19 Use on smartphone and tablet

Client only priting on small scale and/or less powerfull devices is challenging to get nice printed results.

4.20 Printing API

Provinding an API for 3rd party

4.21 Multiserver use, autoscaling, etc.

Most print server are meant to run on a single machine.

4.22 Local data

Files added to project with drag-n-drop

5 Conclusion

Colophon

This document was created with pandoc and eisvogel, with the following commands:

```
$ pandoc -f markdown --latex-engine=xelatex   --number-sections \
  --template eisvogel  ”web-map-printing.md” --toc -o ”web-map-printing
     .pdf”
```