PLR Cadastre

Cadastre RDPPF

ÖREB-Kataster

Cadastre of public-law restrictions on landownership

Cadastre des restrictions de droit public à la propriété foncière

Kataster der öffentlich-rechtlichen Eigentumsbeschränkungen

Definition

The Cadastre of public-law restrictions on landownership (PLR Cadastre) is a reliable, official system providing information about the most important public law restrictions on landownership.

Now I have all the relevant information, should I really buy this new house ?

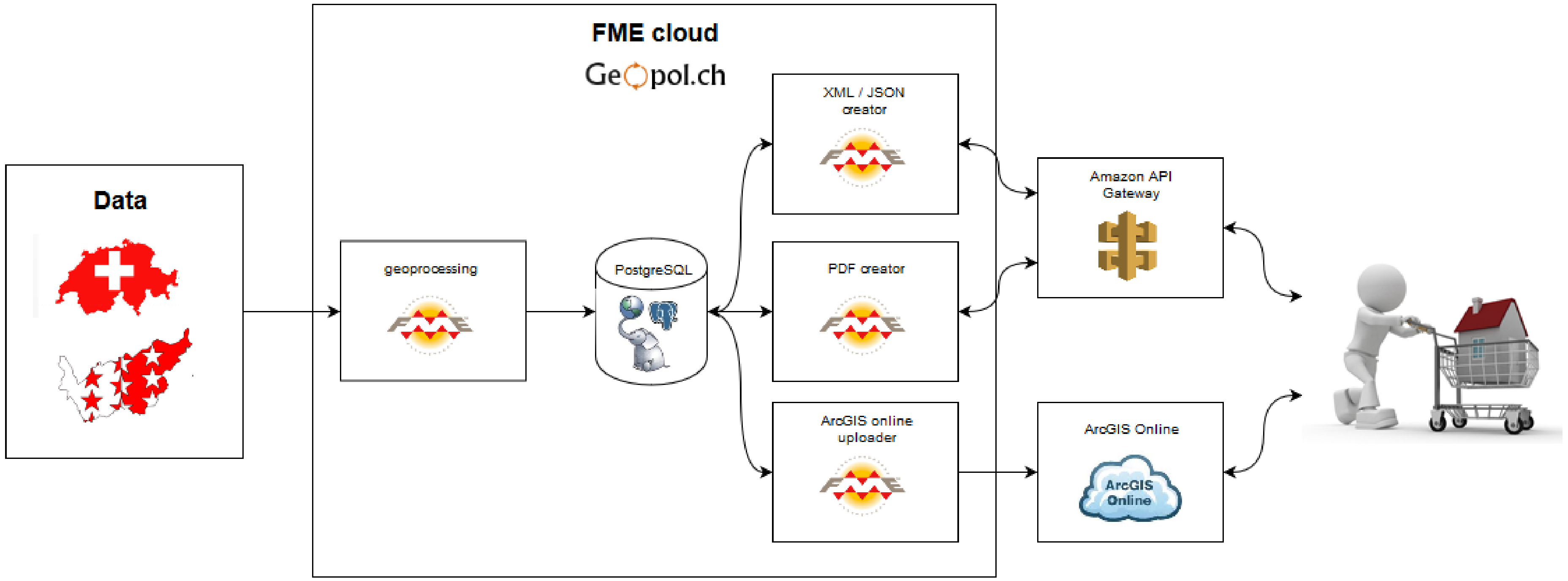
The closest pollution site is 100m away (petrol station)	✓
No restrictions due to railway proximity	✓
The building is close to an airport, number of floors is limited	✗
No groundwater protection area	✓
The noise sensitivity level is quite high, there might be restrictions if constructing part of the building	✗
No restriction due to highway proximity	✓
Only houses with no more than two floors are allowed in this area, solar panels are not allowed	✗
The land is surrounded by a forest, therefore, part of it can not be touched	✗

INSER

Initial requirements PLR

- Output :
  - Static extract : PDF
  - Dynamic extract : Arcgis Online
  - Data extract (REST API) : XML + JSON
- Use of the existing infrastructure (ArcGIS online, FME cloud).
- No local installation (cloud based)
- Fulfill federal standards

Architecture 2017



INSER

Issue n

Auszug aus dem Kataster der öffentlich-rechtlichen Eigentumsbeschränkungen (ÖREB-Kataster)

Grundstück-Nr.	**1549**
E-GRID	nicht verfügbar
Gemeinde	Martigny (6136)
Sektor	
Fläche	8673 m²
Auszugsnummer	**5080CC393BE2AF1527ECB5C15E4718 6E**
Erstellungsdatum des Auszugs	8.5.2018
Katasterverantwortliche Stelle	Dienststelle für Geoinformation, Av. de la Gare 39, 1950 Sion

6 × 6 mm

40

Sommaire

53

Restrictions à la

Page

Espace 3.8 mm

Espace 1.7 mm

3 Plan d'affect

4 Plan d'amén

5 Objets natu

6 Cadastre d

7

Restrictions

Préplublicatio

Plan des alig

public à

stra Roman / Bold

de police 8.5 pt

cement 11.5 pt

oger Origin

Generation du PDF

te der Gemeinde

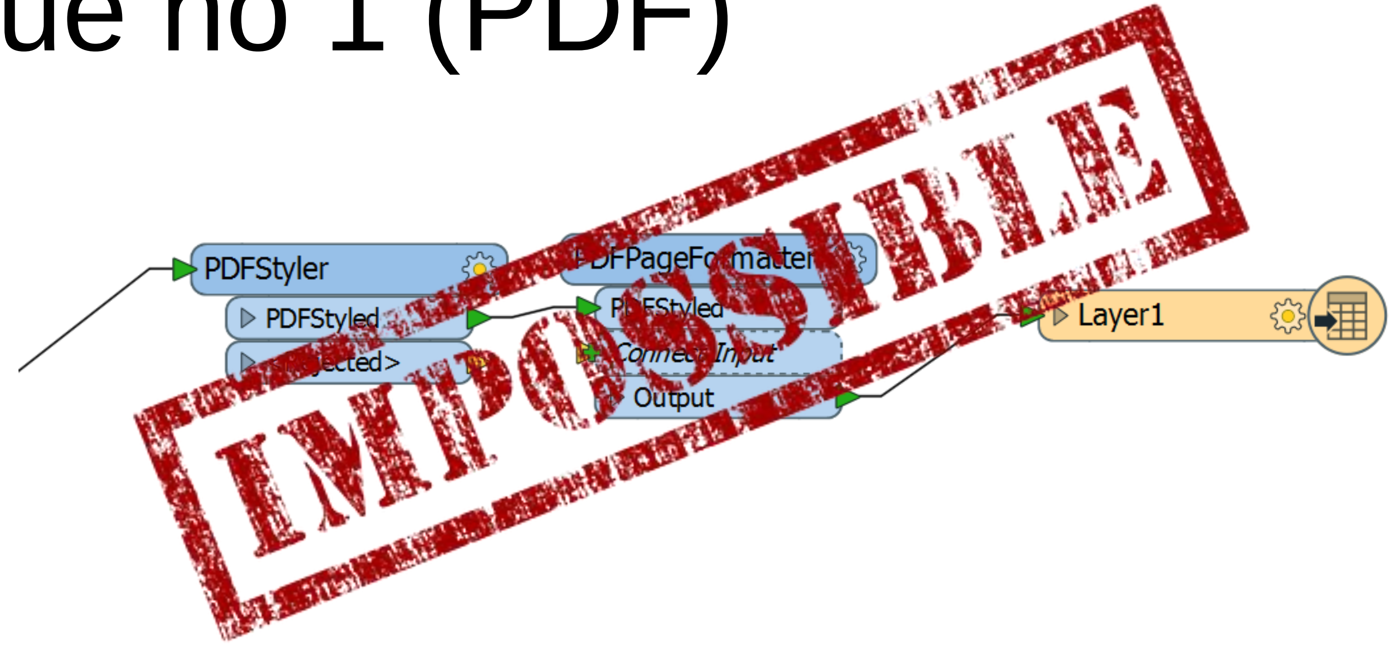
RDPPF /

adastra Bold

18 pt

pt

Issue no 1 (PDF)



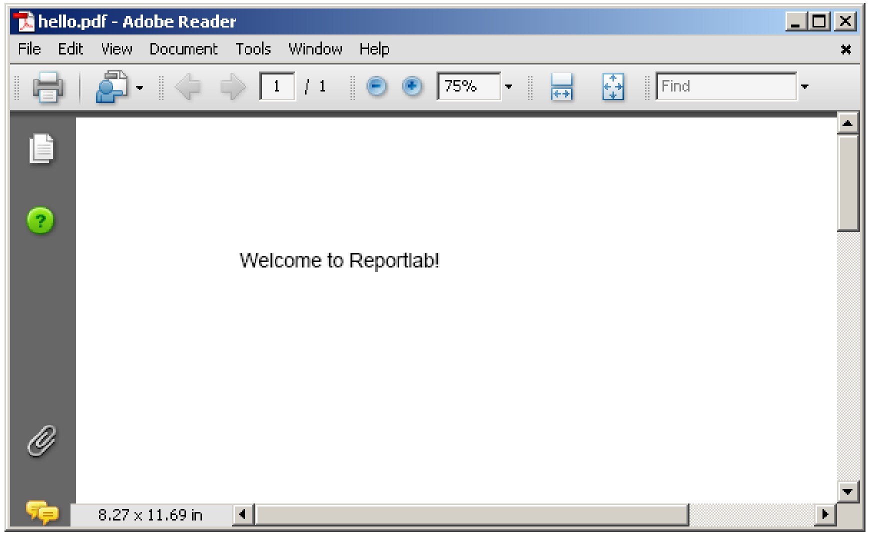
- FME can not create accurate layout
- Not dynamic (unknown number of pages)

FME
python
ReportLab

Issue no 1 (PDF)

```python
from reportlab.pdfgen import canvas

c = canvas.Canvas("hello.pdf")
c.drawString(100,750,"Welcome to Reportlab!")
c.save()
```



Welcome to Reportlab!

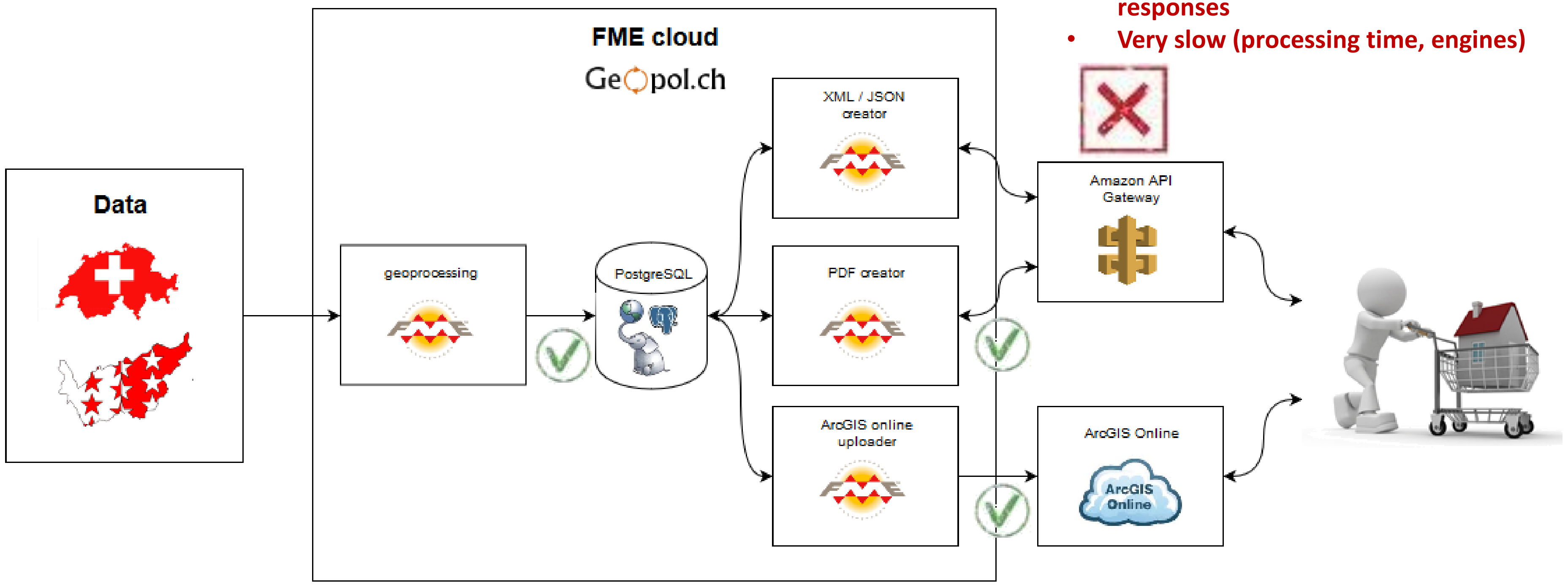
Requirements

- ✓ Millimeter accuracy
- ✓ Font size
- ✓ Shape (box, line, etc)
- ✓ Image
- ✓ Dynamic page number
- ✓ URL link
- ✓ Linux cloud based

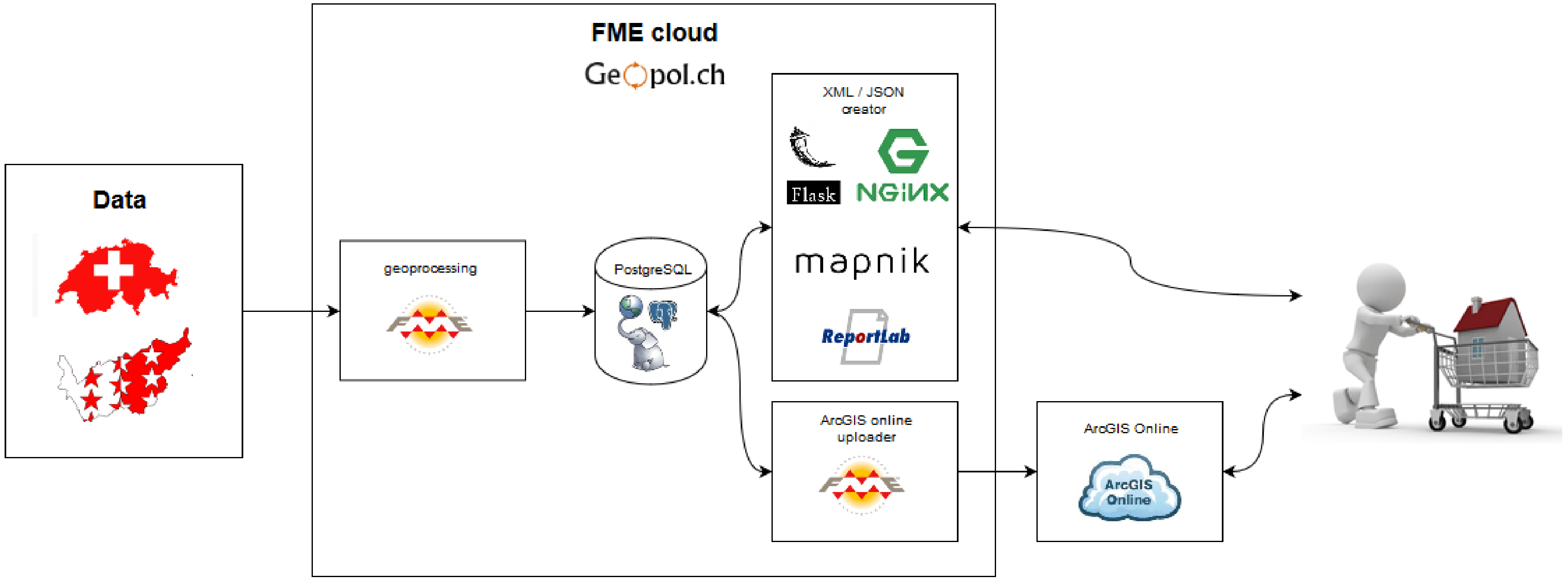
Architecture 2017

Issues no 2 and 3

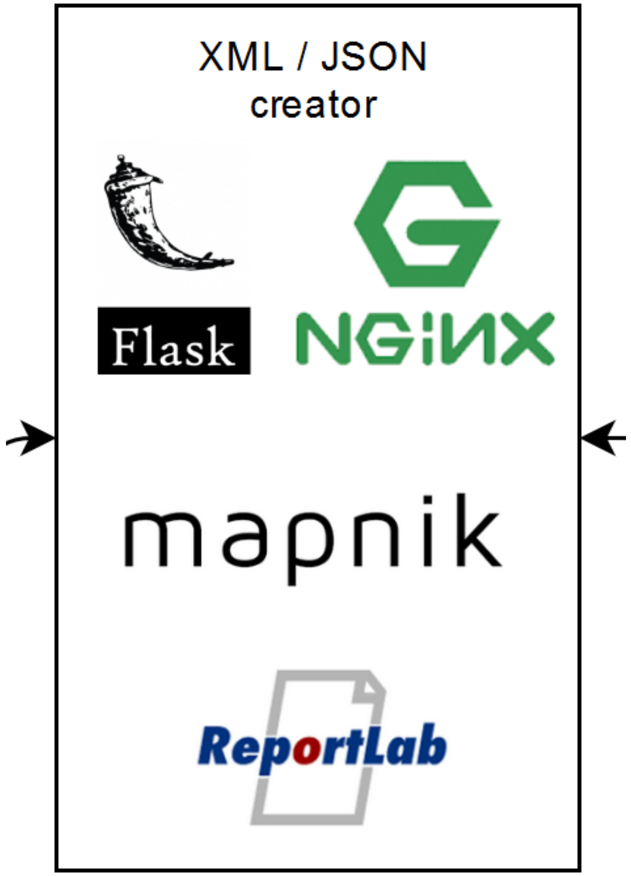
- Difficult to set up
- Unsatisfactory workaround to transmit types file (JSON/XML/PDF)
- Unable to create error codes responses
- Very slow (processing time, engines)



Architecture 2018



Architecture 2018



Flask is a micro web framework written in Python

Nginx is a free and open-source web server

Mapnik is an open source mapping toolkit

Reportlab a software library that lets you directly create documents in Adobe's Portable Document Format (PDF)

100 % python

Flask

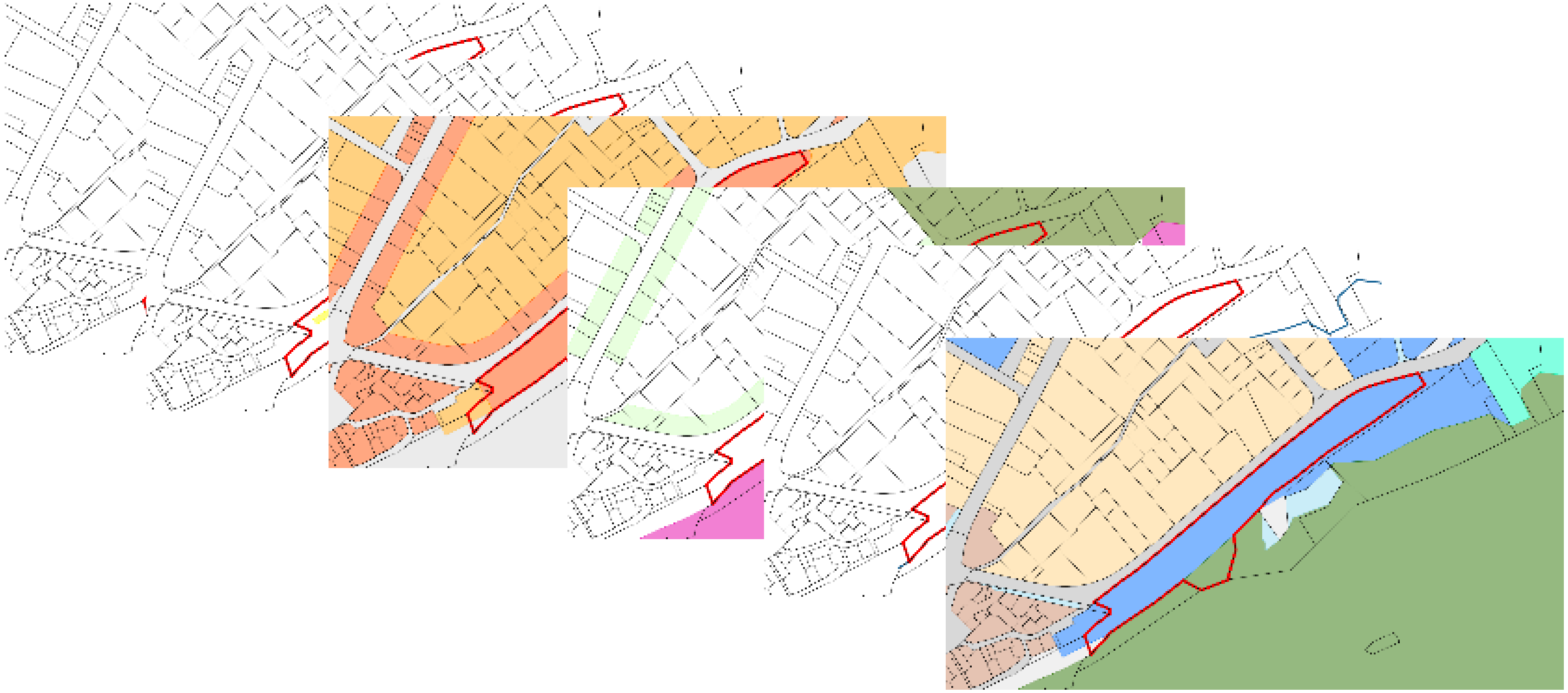
```python
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello World!"
```

- Built-in development server and fast debugger
- Jinja2 templating
- Flask documentation is comprehensive, full of examples and well structured.
- It is super easy to deploy Flask in production
- High Flexibility

Mapnik

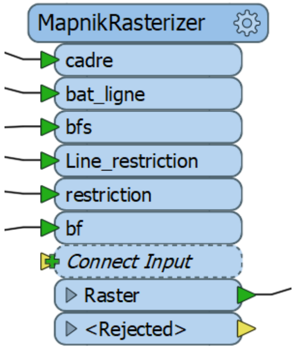
Last challenge, generate rasters

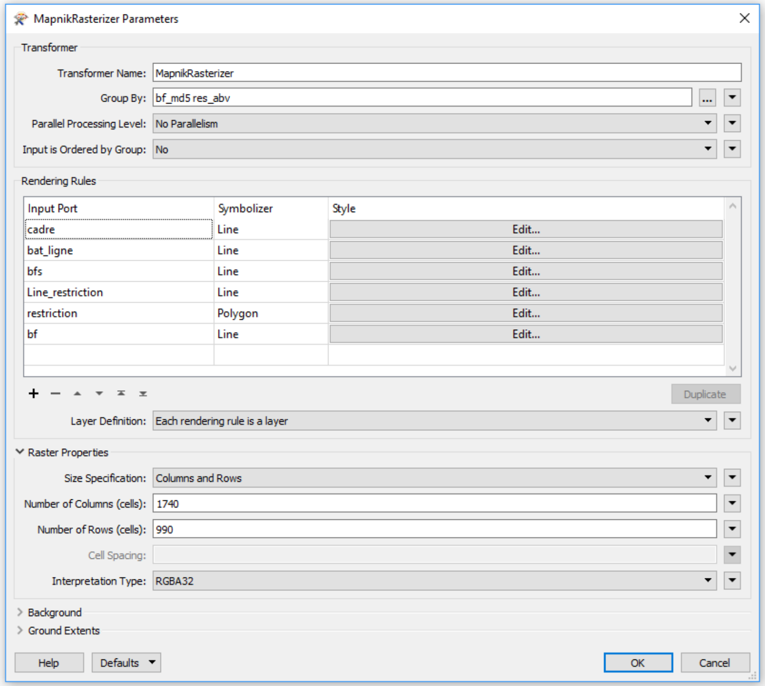


Mapnik on FME

FME transformer based on Mapnik, a Free Toolkit for developing mapping applications.

Draws input points, lines, polygons, and rasters features onto a raster using the Mapnik toolkit.




MapnikRasterizer Parameters

Transformer

Transformer Name:	MapnikRasterizer
Group By:	bf_md5 res_abv
Parallel Processing Level:	No Parallelism
Input is Ordered by Group:	No

Rendering Rules

Input Port	Symbolizer	Style
cadre	Line	Edit...
bat_ligne	Line	Edit...
bfs	Line	Edit...
Line_restriction	Line	Edit...
restriction	Polygon	Edit...
bf	Line	Edit...

Duplicate

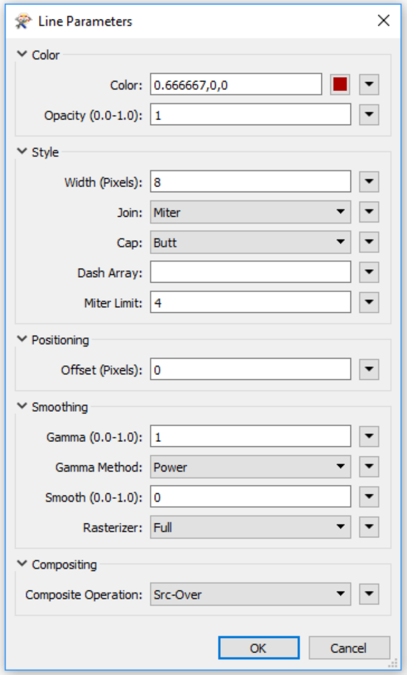
Layer Definition: Each rendering rule is a layer

Raster Properties

Size Specification:	Columns and Rows
Number of Columns (cells):	1740
Number of Rows (cells):	990
Cell Spacing:	
Interpretation Type:	RGBA32

Background
Ground Extents

Help | Defaults | OK | Cancel


Line Parameters

Color

Color:	0.666667,0,0
Opacity (0.0-1.0):	1

Style

Width (Pixels):	8
Join:	Miter
Cap:	Butt
Dash Array:	
Miter Limit:	4

Positioning

Offset (Pixels):	0

Smoothing

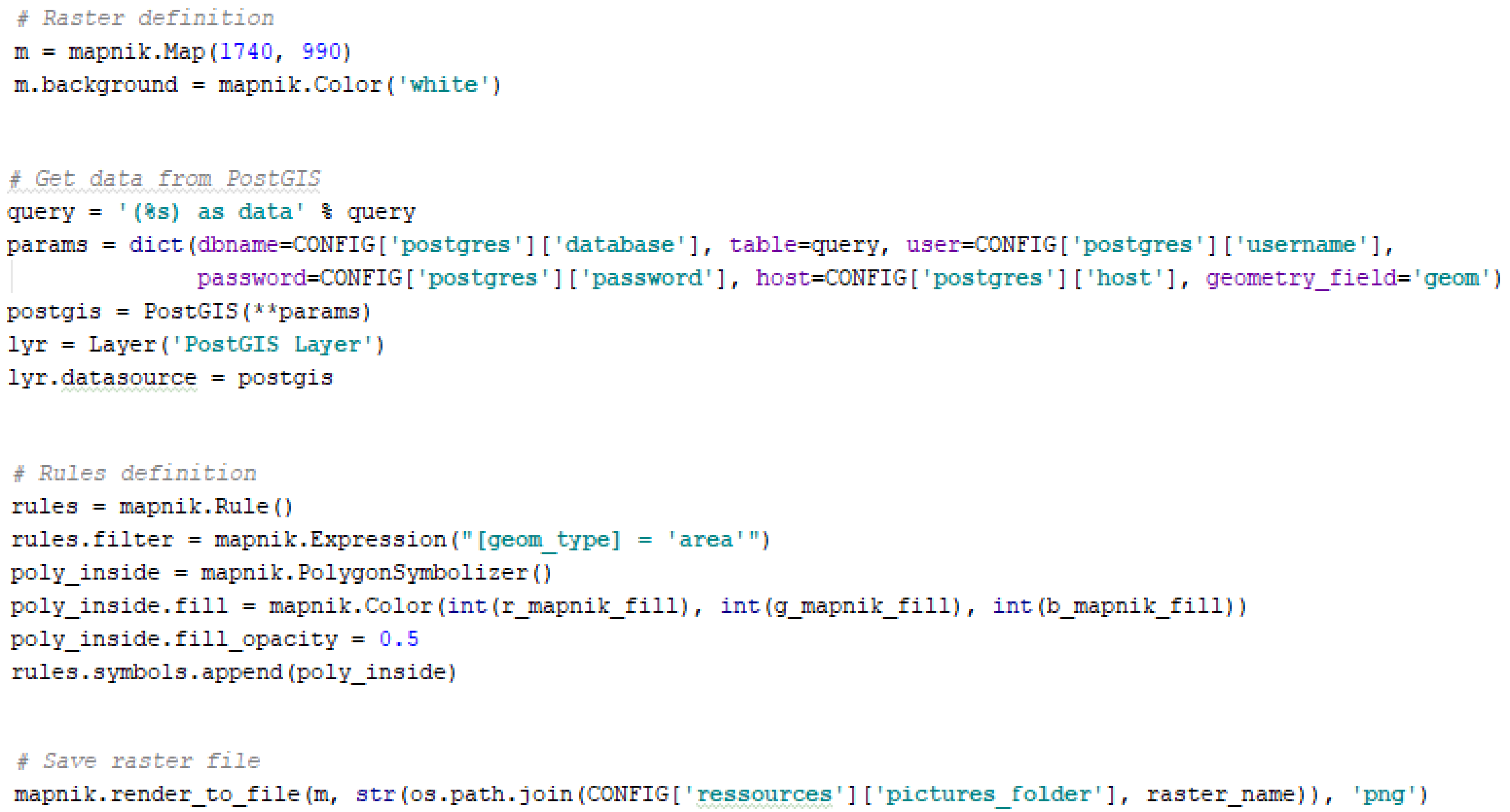
Gamma (0.0-1.0):	1
Gamma Method:	Power
Smooth (0.0-1.0):	0
Rasterizer:	Full

Compositing

Composite Operation:	Src-Over

OK | Cancel

Mapnik



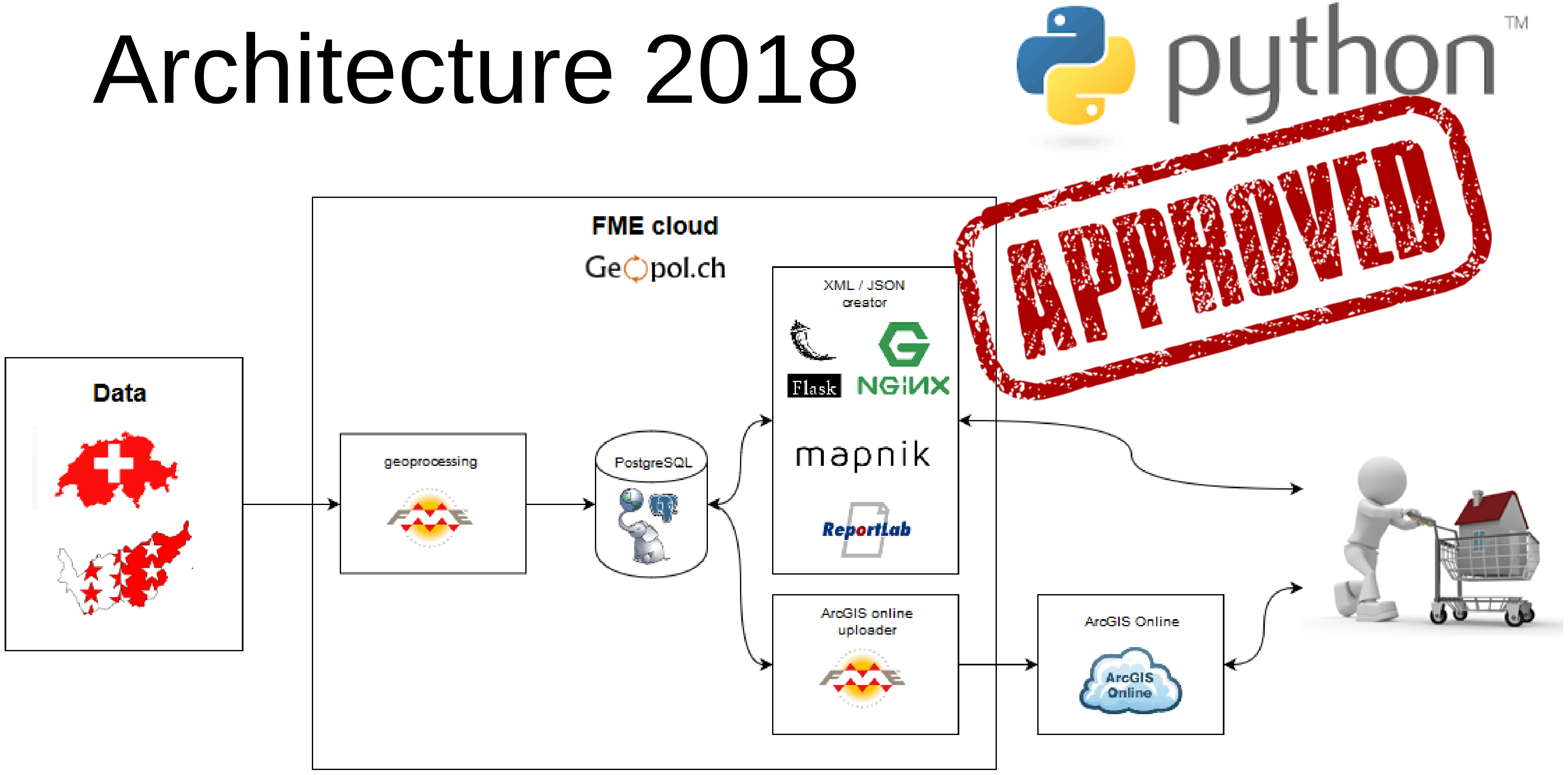
```python
# Raster definition
m = mapnik.Map(1740, 990)
m.background = mapnik.Color('white')


# Get data from PostGIS
query = '(%s) as data' % query
params = dict(dbname=CONFIG['postgres']['database'], table=query, user=CONFIG['postgres']['username'],
              password=CONFIG['postgres']['password'], host=CONFIG['postgres']['host'], geometry_field='geom')
postgis = PostGIS(**params)
lyr = Layer('PostGIS Layer')
lyr.datasource = postgis


# Rules definition
rules = mapnik.Rule()
rules.filter = mapnik.Expression("[geom_type] = 'area'")
poly_inside = mapnik.PolygonSymbolizer()
poly_inside.fill = mapnik.Color(int(r_mapnik_fill), int(g_mapnik_fill), int(b_mapnik_fill))
poly_inside.fill_opacity = 0.5
rules.symbols.append(poly_inside)


# Save raster file
mapnik.render_to_file(m, str(os.path.join(CONFIG['ressources']['pictures_folder'], raster_name)), 'png')
```

Architecture 2018



https://www.vs.ch/web/egeo/cadastre_rdppf