COSMetyc : OpenStreetMap en OCaml

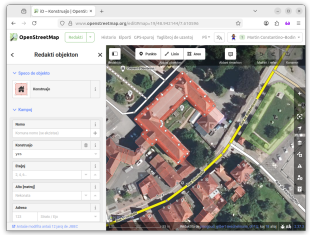
Timothé Baleras, **Martin Bodin**, et Ugo Comignani

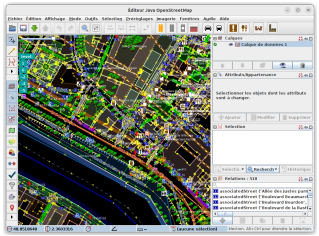
Inria

28 janvier 2026

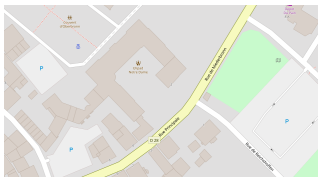
OpenStreetMap

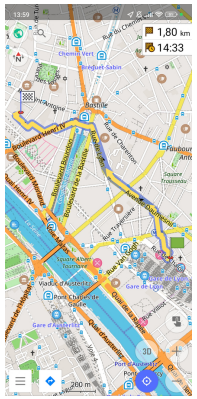
OpenStreetMap est une base de données





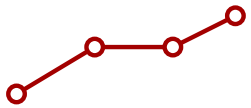
Données cartographiques



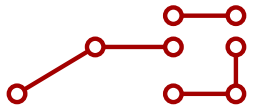


Les 3 types dans OpenStreetMap

name = Buckelstein
tourism = viewpoint
wheelchair = no

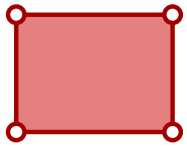


handrail = yes
highway = steps
incline = down

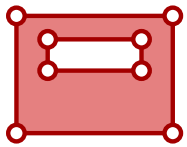


name = Croix verte
network = lwn
route = hiking
symbol = green_cross
type = route

Le $3{,}5^{\text{e}}$ type d'OpenStreetMap



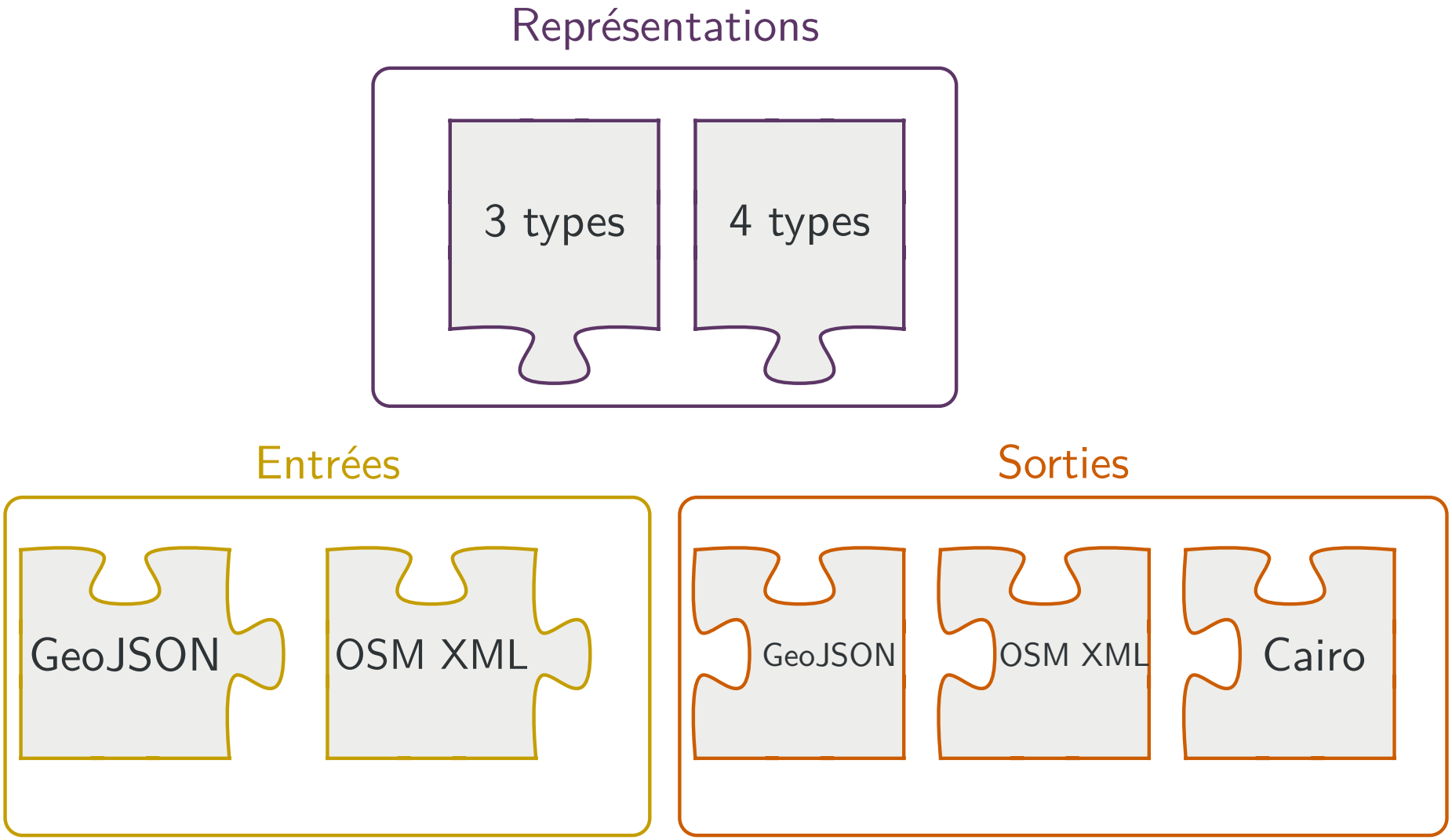
amenity = place_of_worship
building = church
denomination = catholic
religion = christian
wikidata = Q68165164



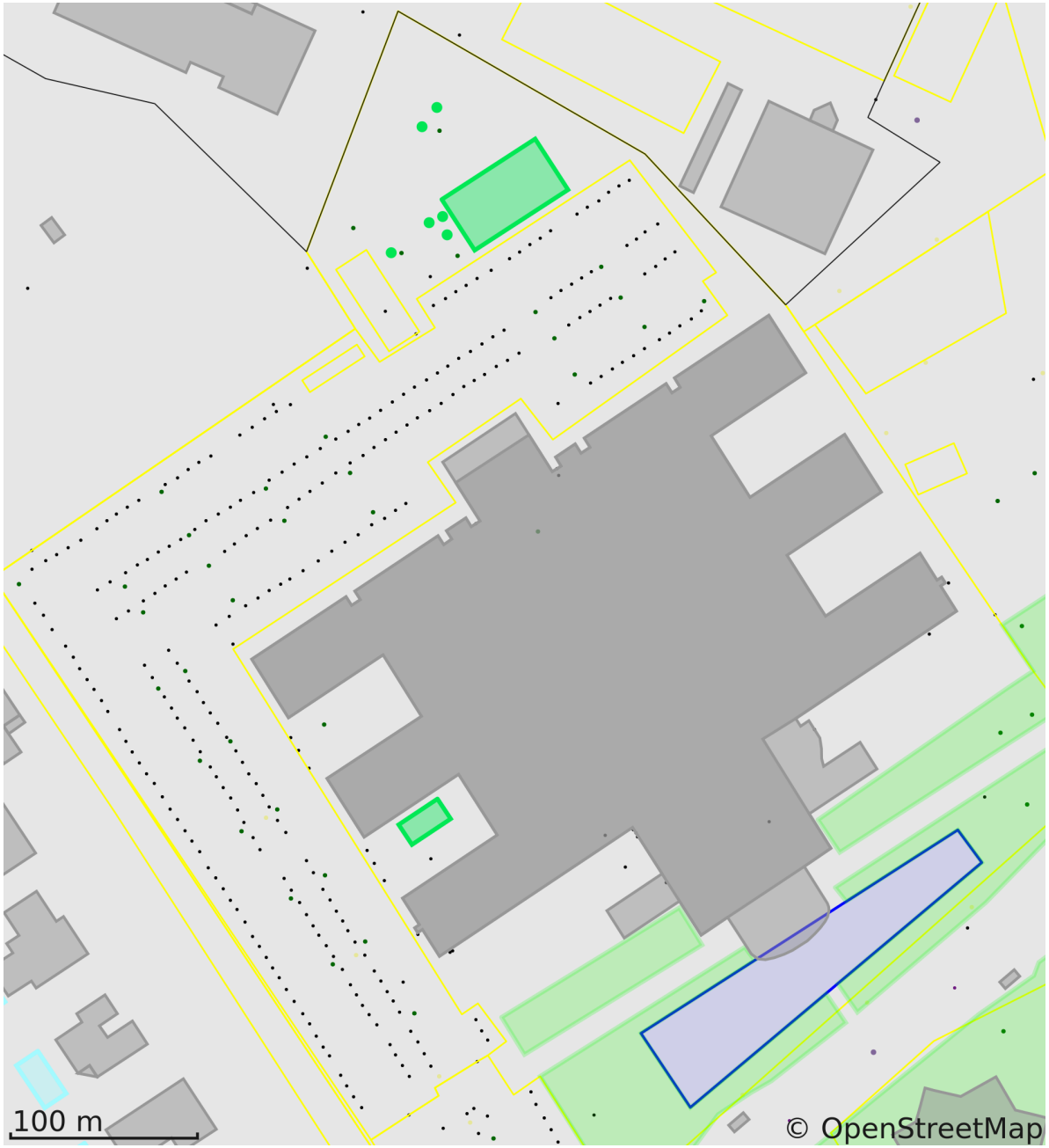
CLC:code = 112
CLC:id = FR-26157
CLC:year = 2006
landuse = residential
type = multipolygon

COSMetyc

Architecture modulaire



Exemple d'usage



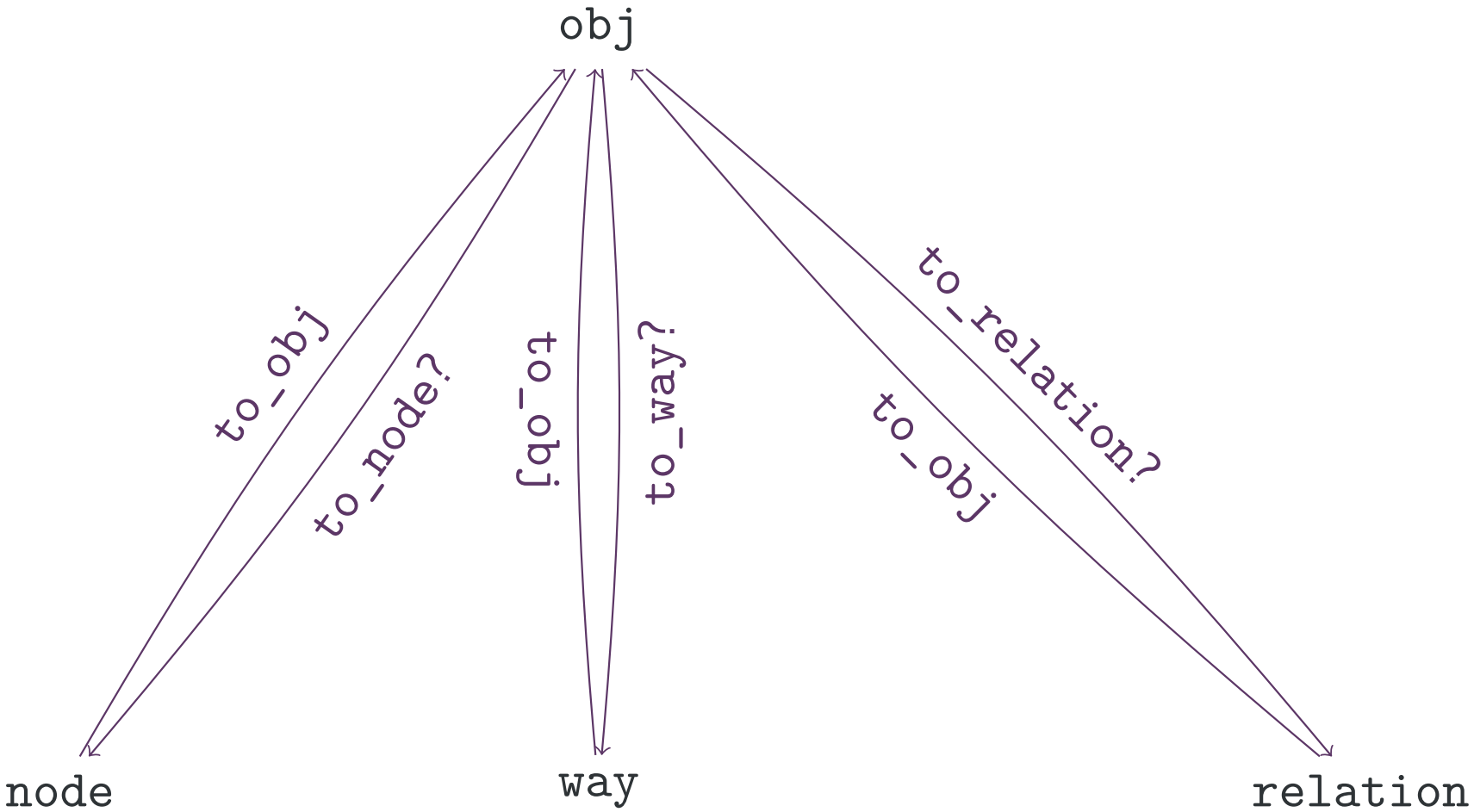
Exemple d'usage

```
1  open Cosmetyc
2  module Rep = Repr.Raw
3  module In = Input.OSM_XML.Make (Rep)
4  module Out = Output.Cairo_render.Default (Rep)
5
6  let input_flux =
7    In.open_in (In_channel.open_bin "data.osm") in
8  let objs = In.get_objects input_flux in
9  let bbox = In.get_bbox input_flux in
10 In.close input_flux;
11
12 let output_flux =
13   Out.open_out (Out.SVG "output.svg")
14     bbox 10. ~background:(0.9, 0.9, 0.9, 1.0) () in
15 Out.write_objects output_flux objs ;
16 Out.close output_flux_png
```

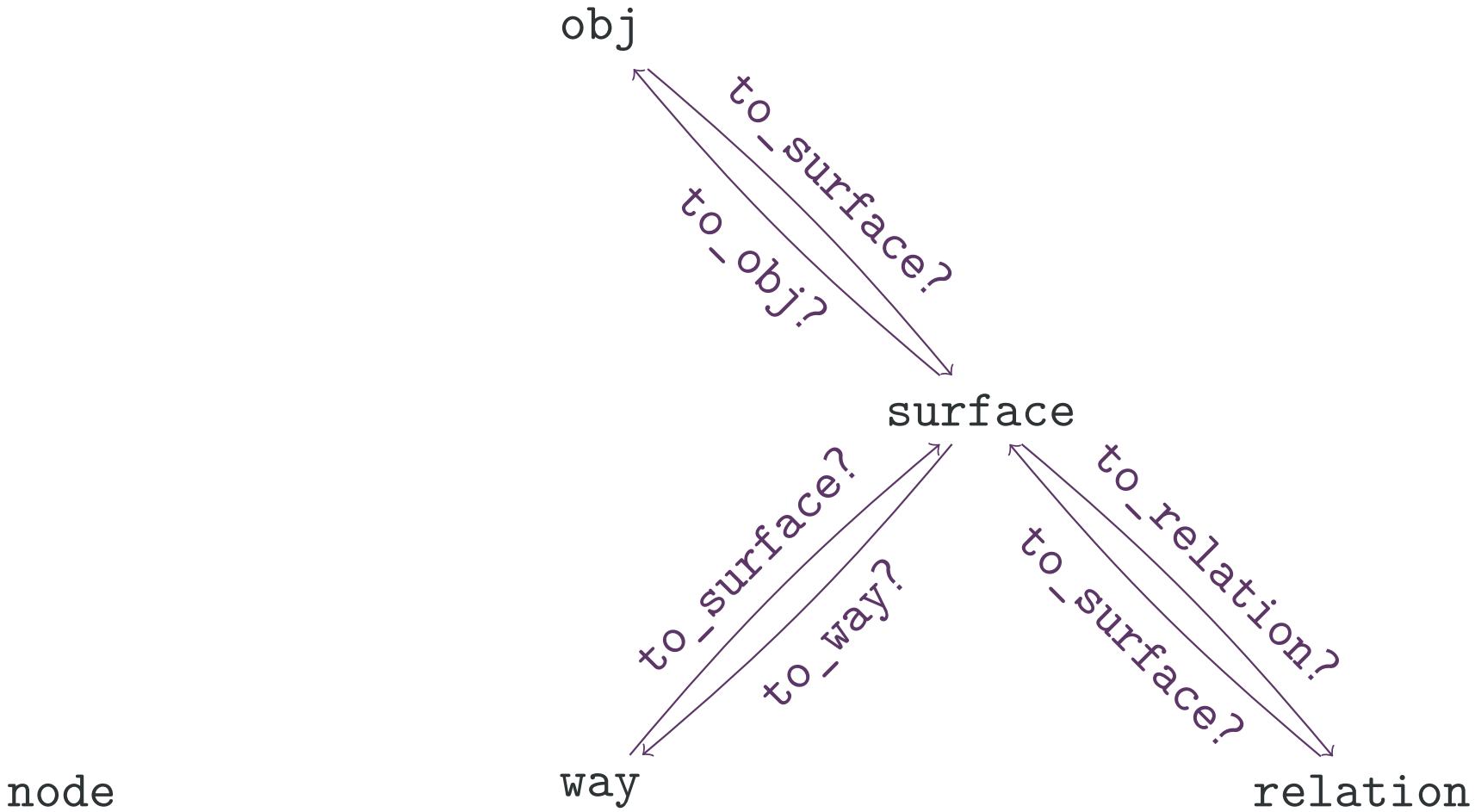
Exemple d'usage

```
1  open Cosmetyc
2  module Rep = Repr.Surface
3  module In = Input.OSM_XML.Make (Rep)
4  module Out = Output.Cairo_render.Default (Rep)
5
6  let input_flux =
7    In.open_in (In_channel.open_bin "data.osm") in
8  let objs = In.get_objects input_flux in
9  let bbox = In.get_bbox input_flux in
10 In.close input_flux;
11 let objs = Seq.filter Rep.Object.is_surface objs in
12 let output_flux =
13   Out.open_out (Out.SVG "output.svg")
14     bbox 10. ~background:(0.9, 0.9, 0.9, 1.0) () in
15 Out.write_objects output_flux objs ;
16 Out.close output_flux_png
```

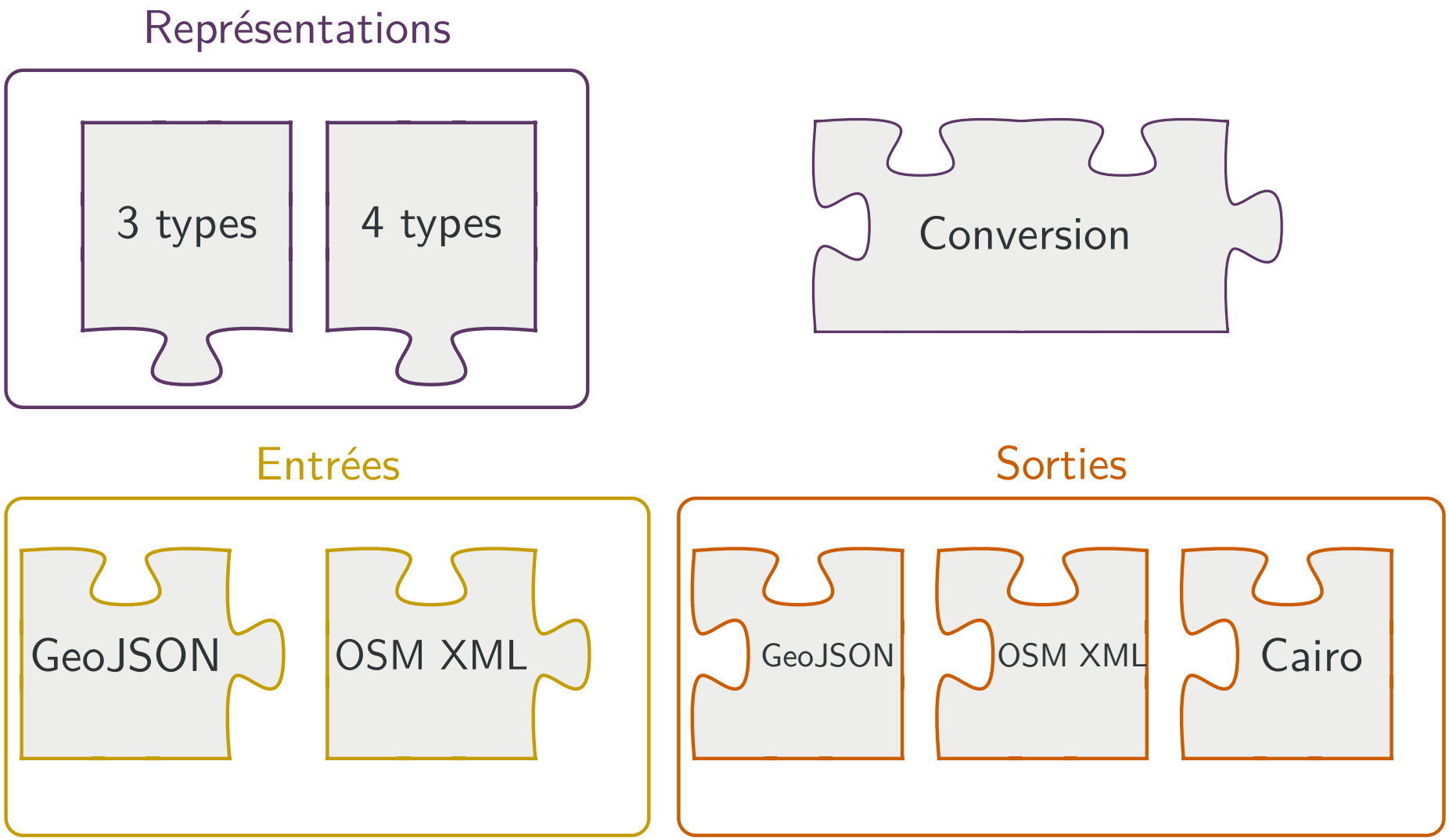
Signatures de modules



Signatures de modules



Architecture modulaire



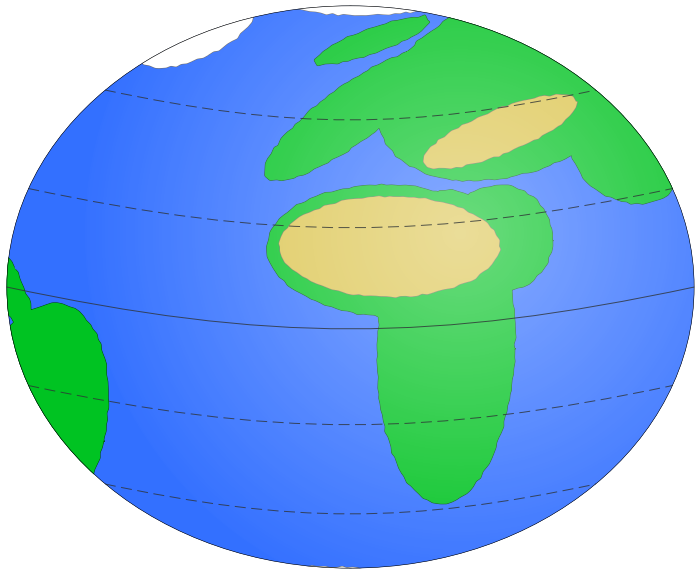
Garanties sur la sortie, à la TyXML


```
type +'a attrib = Xml.attrib
type +'a elt = Xml.elt
```

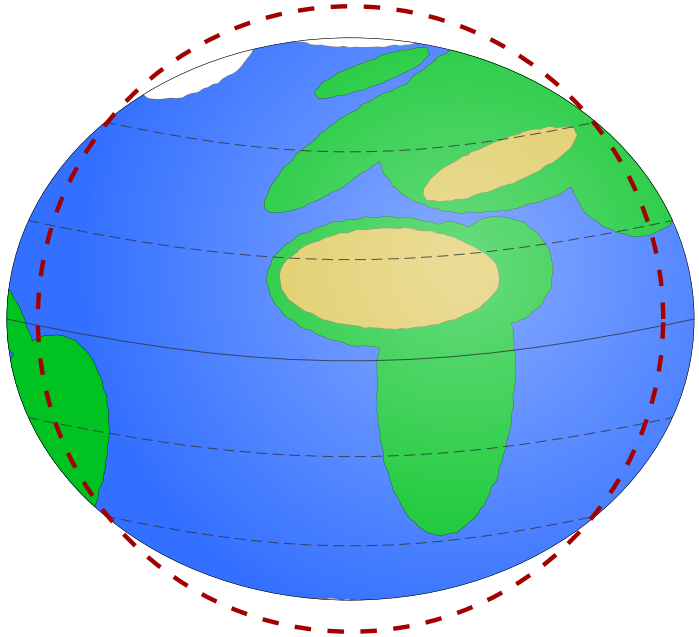
```
val a_date : string -> [> `Date ] attrib
val a_uid : int -> [> `Uid ] attrib
val a_user : string -> [> `User ] attrib
```

```
val comment : ([< `Date | `Uid | `User ],
               [< `Text ],
               [> `Comment ]) star
```

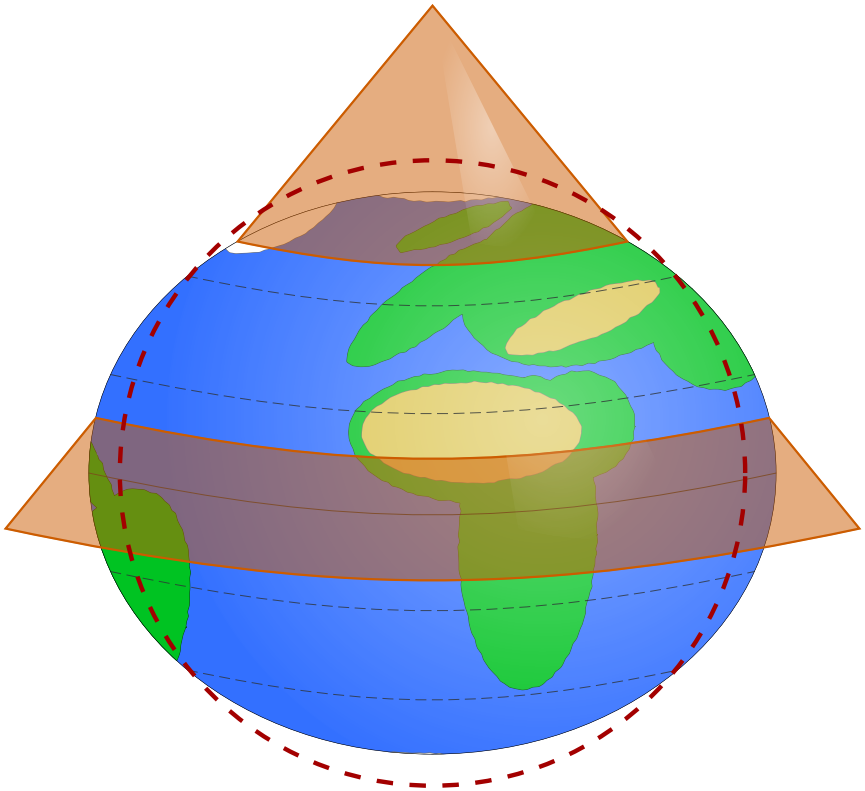
Le monde est compliqué



Le monde est compliqué



Le monde est compliqué



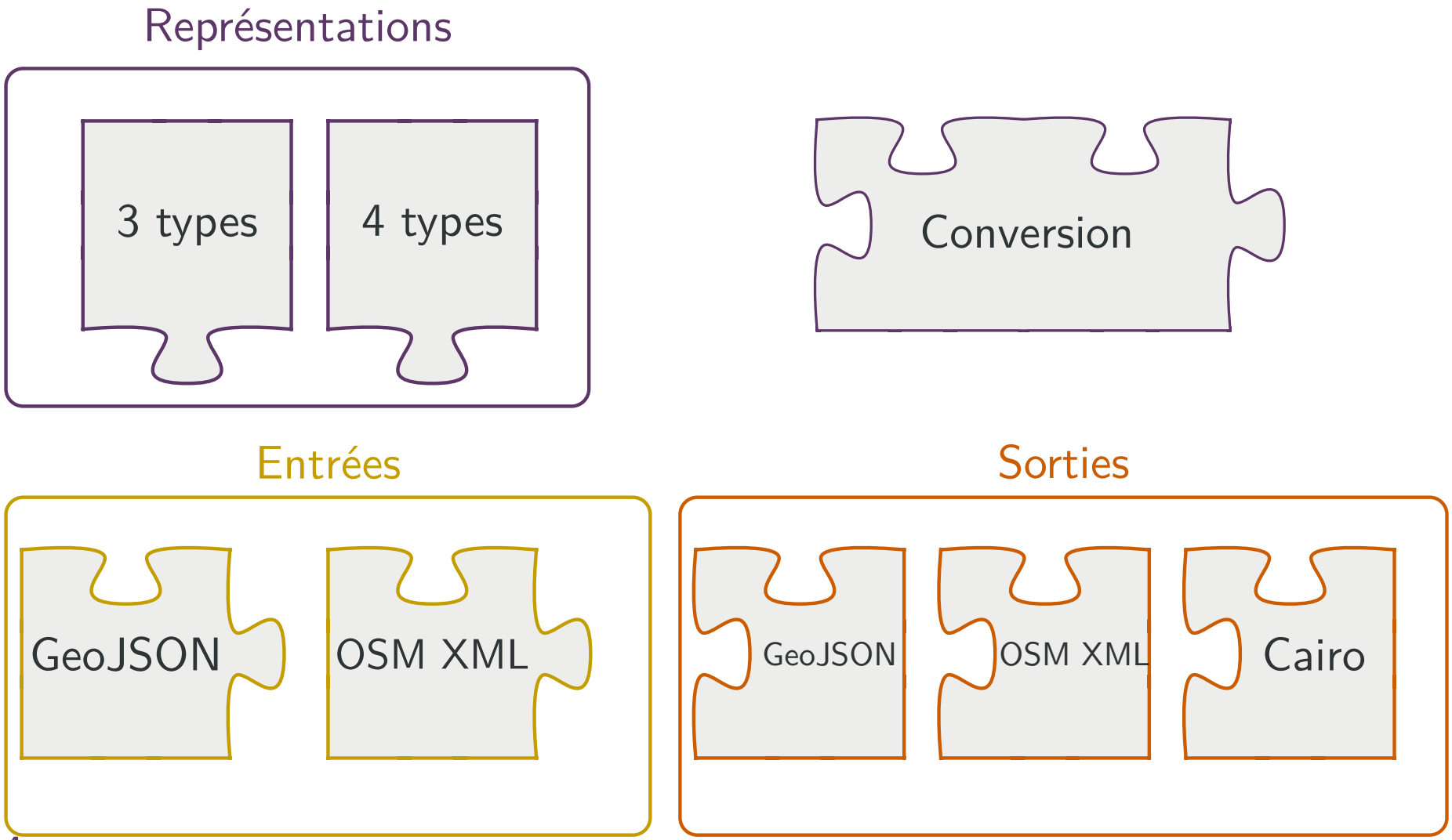
Et maintenant ?

`https://gitlab.inria.fr/mbodin1/logic-osm`

- Importation de lampadaires via COSMetyc.
- Analyse de l'historique :
  - Prédiction de changements futurs.
  - Estimation de la qualité de données.
- Outils pour la communauté.

Merci pour votre attention !

`https://gitlab.inria.fr/mbodin1/logic-osm`



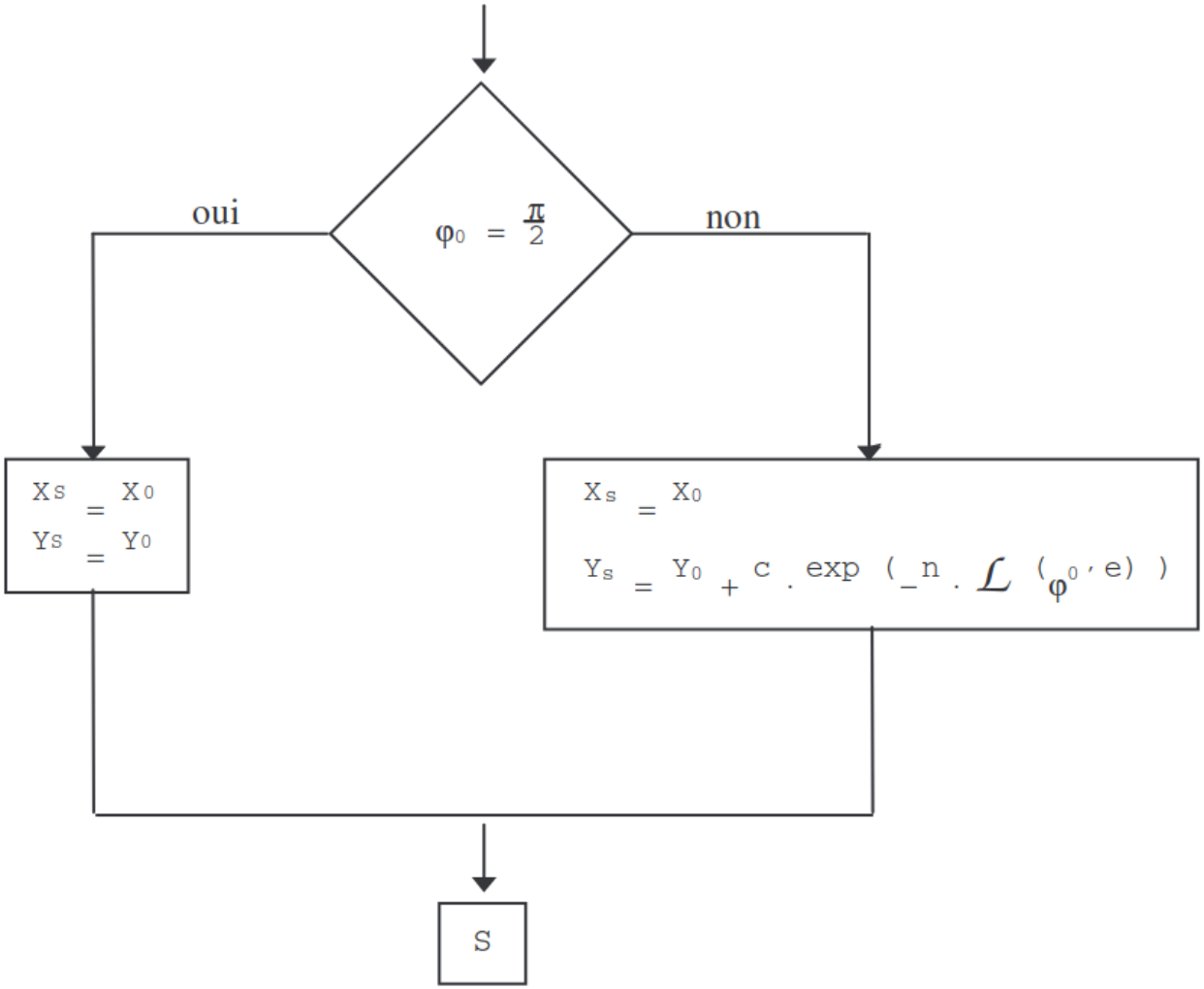
1. OpenStreetMap

2. COSMetyc
   - Architecture modulaire
   - Signatures de modules
   - Garanties à la TyXML
   - Projections

3. Futur

Bonus

Spécifications de l'IGN



1. OpenStreetMap

2. COSMetyc
   - Architecture modulaire
   - Signatures de modules
   - Garanties à la TyXML
   - Projections

3. Futur