Extraction et évolution de contraintes d'intégrité dans les Géocommuns

Martin Bodin Ugo Comignani

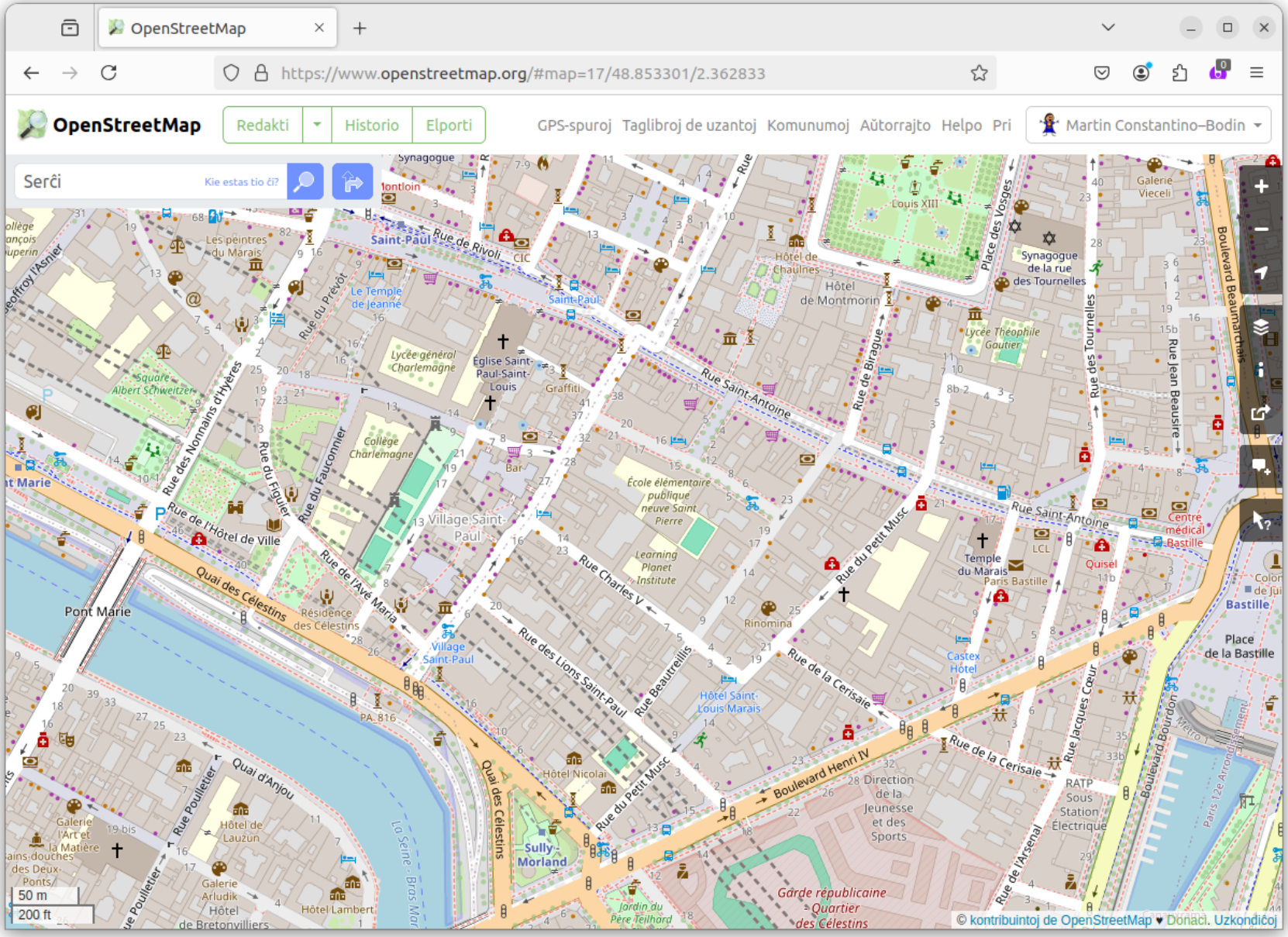
Spades TyRex

Inria

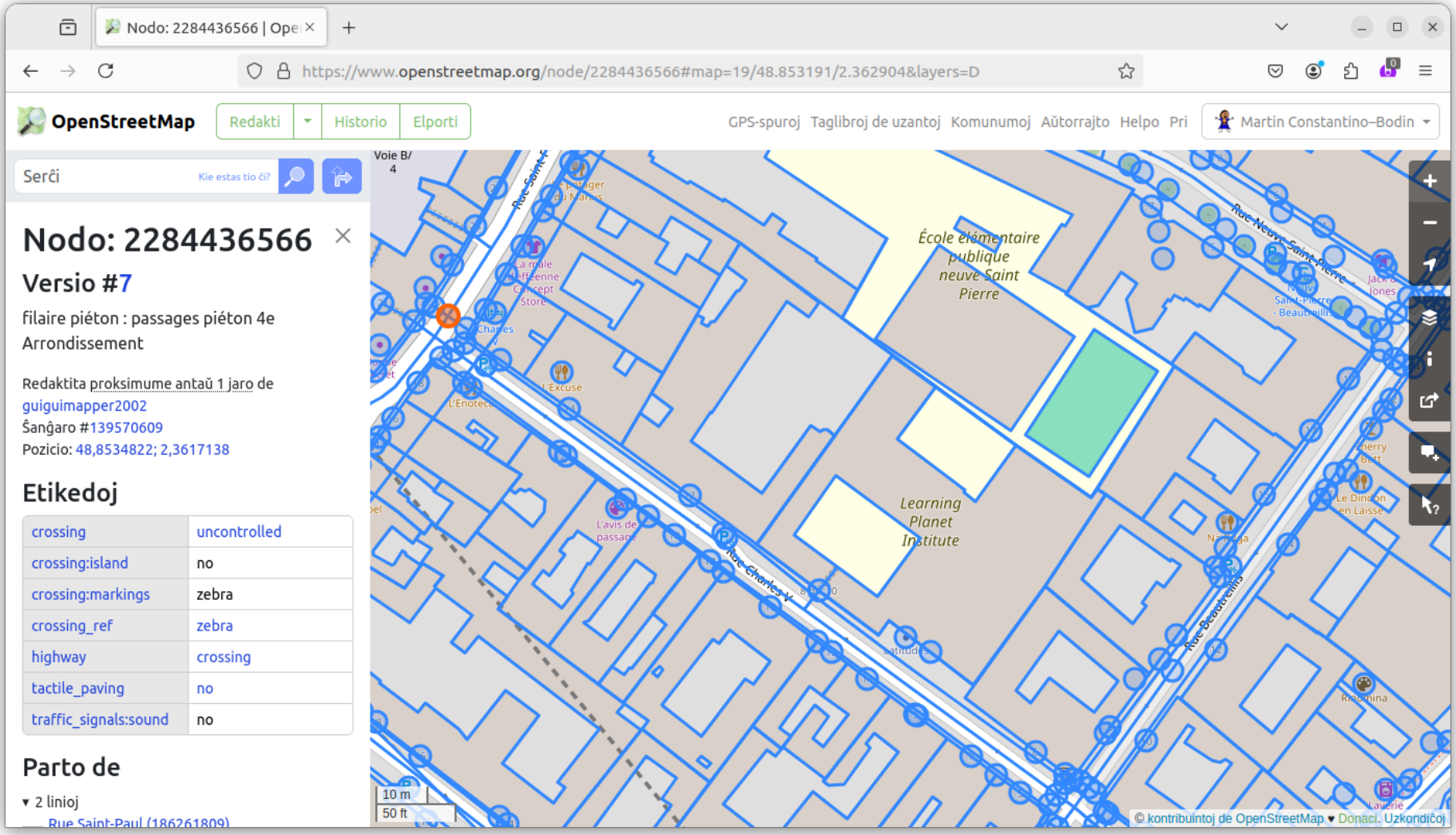
6 Novembre 2024

OpenStreetMap

Le site openstreetmap.org

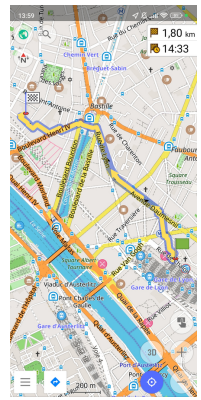


Le site openstreetmap.org



OpenStreetMap est une base de données

Données
cartographiques



Schémas de données

highway	=	crossing
crossing	=	uncontrolled
crossing:markings	=	zebra
crossing:island	=	no
traffic_signals:sound	=	no
tactile_paving	=	no
crossing_ref	=	zebra

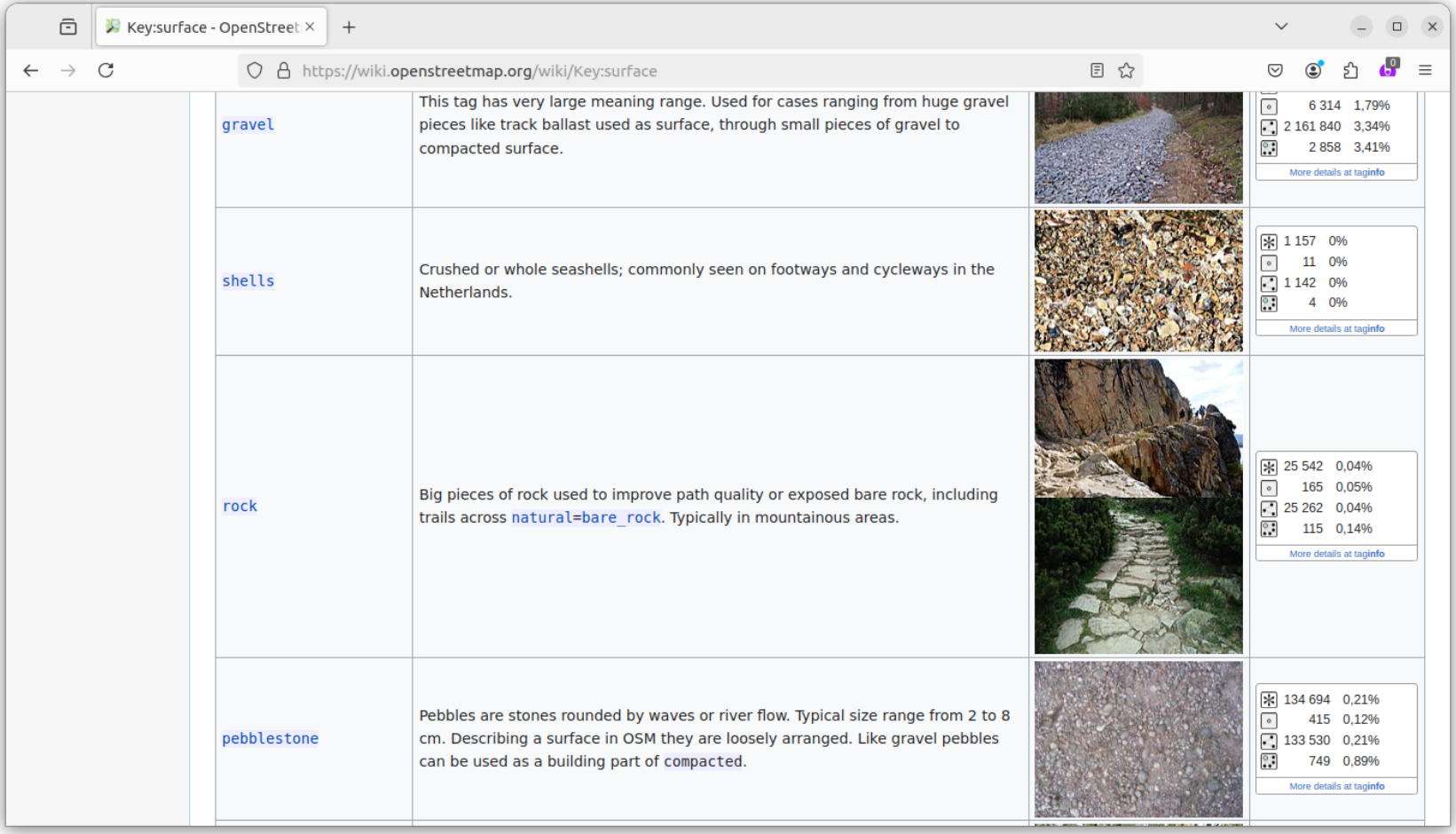
Schémas de données

$$
\begin{aligned}
\texttt{highway} &= \texttt{crossing} \\
\texttt{crossing} &= \texttt{uncontrolled} \\
\texttt{crossing:markings} &= \texttt{zebra} \\
\texttt{crossing:island} &= \texttt{no} \\
\texttt{traffic\_signals:sound} &= \texttt{no} \\
\texttt{tactile\_paving} &= \texttt{no} \\
\texttt{crossing\_ref} &= \texttt{zebra}
\end{aligned}
$$

$$
\left(\quad \texttt{clé} \quad = \quad \texttt{valeur} \quad \right)^{\star}
$$

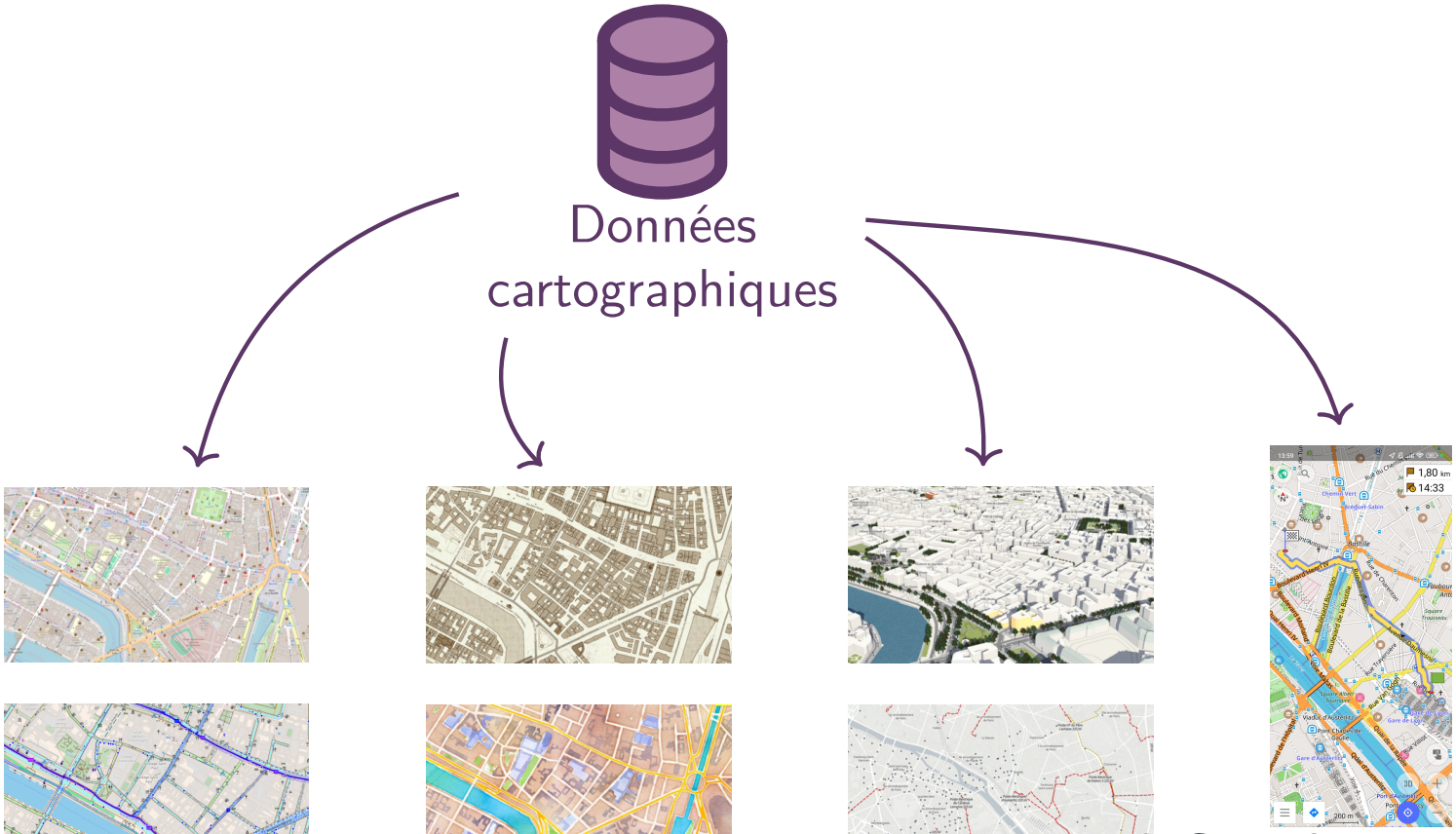
$$
\begin{aligned}
\texttt{clé} &\overset{\text{def}}{=} \texttt{string} \\
\texttt{valeur} &\overset{\text{def}}{=} \texttt{string}
\end{aligned}
$$

Wiki OpenStreetMap

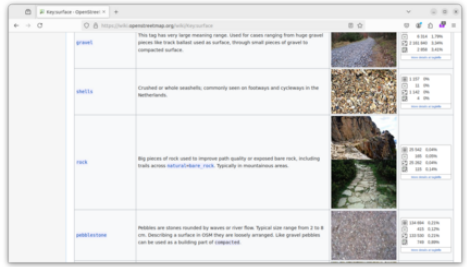


wiki.openstreetmap.org

La communauté OpenStreetMap

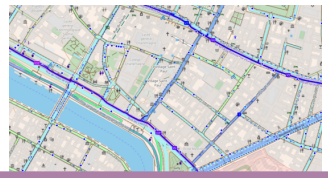


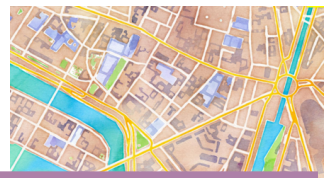
La communauté OpenStreetMap

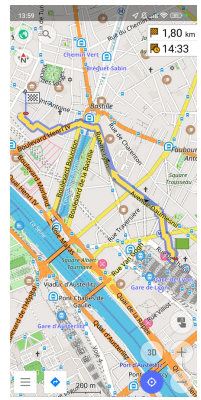


wiki.openstreetmap.org

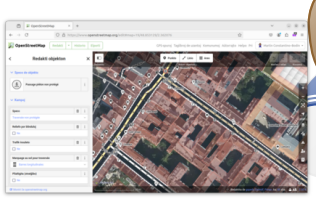
Données cartographiques

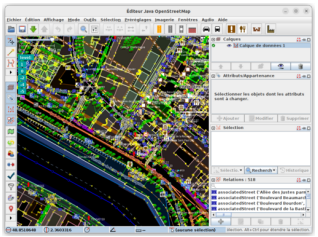


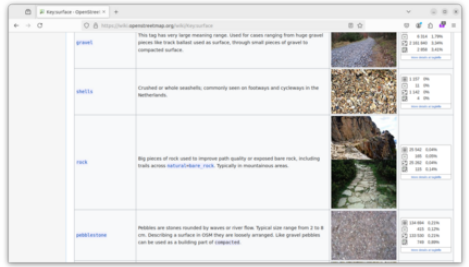




La communauté OpenStreetMap

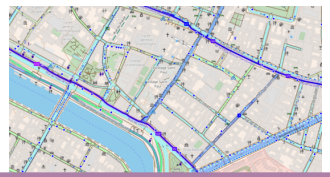


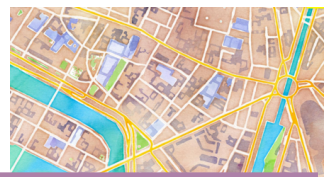


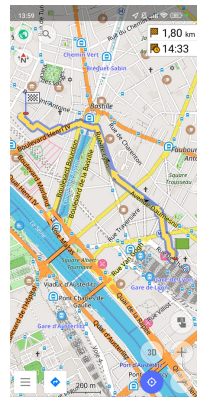


wiki.openstreetmap.org

Données cartographiques







crossing_ref

- zebra

- pelican

- pegasus

- toucan

- puffin

- tiger

crossing_ref

- zebra

- pelican

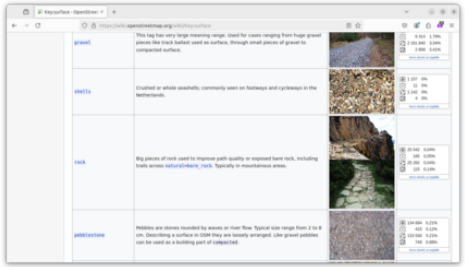
- toucan

- puffin

Évolution des pratiques

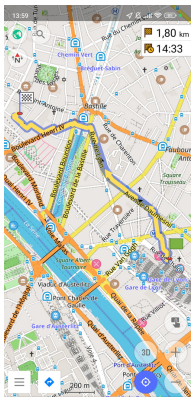
$$\begin{aligned} \texttt{crossing} &= \texttt{pelican} \\ &\wr \\ \texttt{crossing} &= \texttt{traffic\_signals} \\ \texttt{crossing:markings} &= \texttt{dots} \\ \texttt{button\_operated} &= \texttt{yes} \\ \texttt{traffic\_signals:sound} &= \texttt{yes} \end{aligned}$$

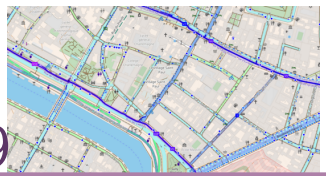
L'usage prime

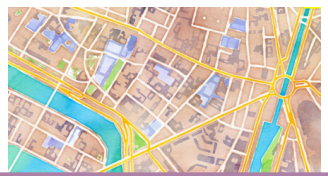


wiki.openstreetmap.org

Données cartographiques







Contraintes d'intégrité

Incohérences dans la base de données

Fautes de frappe

$$\texttt{corrsing} = \texttt{unmarked} \quad \rightsquigarrow \quad \texttt{crossing} = \texttt{unmarked}$$

Clés dépréciées

$$\texttt{crossing} = \texttt{pelican} \rightsquigarrow \begin{array}{r} \texttt{crossing} = \texttt{traffic\_signals} \\ \texttt{crossing:markings} = \texttt{dots} \\ \texttt{crossing:signals} = \texttt{yes} \\ \texttt{crossing\_ref} = \texttt{pelican} \end{array}$$

Clés incompatibles

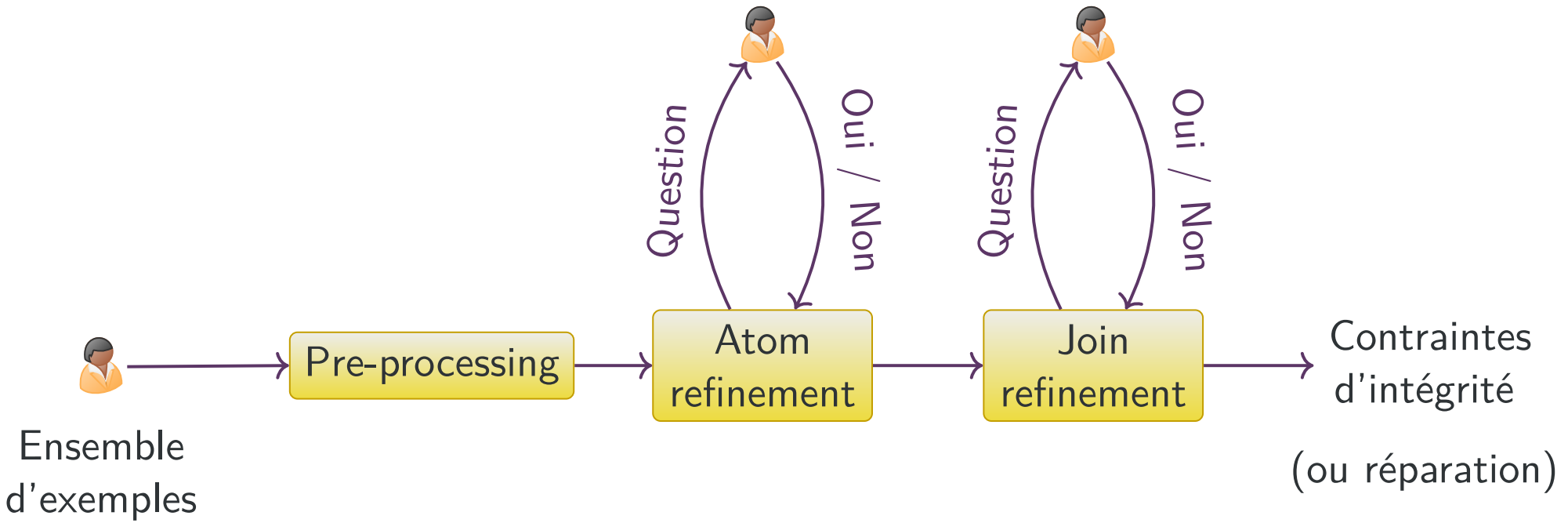
$$\begin{array}{r} \texttt{crossing} = \texttt{unmarked} \\ \texttt{highway} = \texttt{motorway} \end{array}$$

Incohérences entre objets

$$\texttt{sidewalk} = \texttt{no} \quad \text{à côté de} \quad \texttt{footway} = \texttt{sidewalk}$$

Interactive mapping specification and repairing framework

Travaux antérieurs



Interactive mapping specification and repairing framework

Exemple

Company:

IdCompany	*Name*	*Town*
C1	AA	Paris
C2	Ev	Lyon

Flight:

Departure	*Arrival*	*IdCompany*
Lyon	Paris	C1
Paris	Lyon	C2

Travel Agency:

IdAgency	*Name*	*Town*
A1	TC	L.A.

$\Rightarrow$

Carrier:

Id	*Name*	*Town*
Id1	AA	Paris
Id2	Ev	Lyon
Id3	TC	L.A.

Departure:

Town	*IdCarrier*
Lyon	Id1
Paris	Id2

Arrival:

Town	*IdCarrier*
Paris	Id1
Lyon	Id2

$$
\begin{aligned}
&\mathit{Company}(\mathtt{c1}, \mathtt{aa}, \mathtt{paris}) \\
&\wedge\ \mathit{Company}(\mathtt{c2}, \mathtt{ev}, \mathtt{lyon}) \\
&\wedge\ \mathit{Flight}(\mathtt{lyon}, \mathtt{paris}, \mathtt{c1}) \\
&\wedge\ \mathit{Flight}(\mathtt{paris}, \mathtt{lyon}, \mathtt{c2}) \\
&\wedge\ \mathit{TravelAgency}(\mathtt{a1}, \mathtt{tc}, \mathtt{la})
\end{aligned}
\Rightarrow
\begin{aligned}
&\exists \mathtt{id1}, \mathtt{id2}, \mathtt{id3}, \\
&\mathit{Carrier}(\mathtt{id1}, \mathtt{aa}, \mathtt{paris}) \\
&\wedge\ \mathit{Departure}(\mathtt{lyon}, \mathtt{id1}) \\
&\wedge\ \mathit{Arrival}(\mathtt{paris}, \mathtt{id1}) \\
&\wedge\ \mathit{Carrier}(\mathtt{id2}, \mathtt{ev}, \mathtt{lyon}) \\
&\wedge\ \mathit{Departure}(\mathtt{paris}, \mathtt{id2}) \\
&\wedge\ \mathit{Arrival}(\mathtt{lyon}, \mathtt{id2}) \\
&\wedge\ \mathit{Carrier}(\mathtt{id3}, \mathtt{tc}, \mathtt{la})
\end{aligned}
$$

Interactive mapping specification and repairing framework

Exemple

Company:

IdCompany	*Name*	*Town*
C1	AA	Paris
C2	Ev	Lyon

Flight:

Departure	*Arrival*	*IdCompany*
Lyon	Paris	C1
Paris	Lyon	C2

Travel Agency:

IdAgency	*Name*	*Town*
A1	TC	L.A.

$\Rightarrow$

Carrier:

Id	*Name*	*Town*
Id1	AA	Paris
Id2	Ev	Lyon
Id3	TC	L.A.

Departure:

Town	*IdCarrier*
Lyon	Id1
Paris	Id2

Arrival:

Town	*IdCarrier*
Paris	Id1
Lyon	Id2

$$\begin{aligned} &\mathit{Company}(\mathtt{c1}, \mathtt{aa}, \mathtt{paris}) \\ &\wedge\ \mathit{Company}(\mathtt{c2}, \mathtt{ev}, \mathtt{lyon}) \\ &\wedge\ \mathit{Flight}(\mathtt{lyon}, \mathtt{paris}, \mathtt{c1}) \\ &\wedge\ \mathit{Flight}(\mathtt{paris}, \mathtt{lyon}, \mathtt{c2}) \\ &\wedge\ \mathit{TravelAgency}(\mathtt{a1}, \mathtt{tc}, \mathtt{la}) \end{aligned} \Rightarrow \begin{aligned} &\exists \mathtt{id1}, \mathtt{id2}, \mathtt{id3}, \\ &\mathit{Carrier}(\mathtt{id1}, \mathtt{aa}, \mathtt{paris}) \\ &\wedge\ \mathit{Departure}(\mathtt{lyon}, \mathtt{id1}) \\ &\wedge\ \mathit{Arrival}(\mathtt{paris}, \mathtt{id1}) \\ &\wedge\ \mathit{Carrier}(\mathtt{id2}, \mathtt{ev}, \mathtt{lyon}) \\ &\wedge\ \mathit{Departure}(\mathtt{paris}, \mathtt{id2}) \\ &\wedge\ \mathit{Arrival}(\mathtt{lyon}, \mathtt{id2}) \\ &\wedge\ \mathit{Carrier}(\mathtt{id3}, \mathtt{tc}, \mathtt{la}) \end{aligned}$$

Split-reduction

$$\begin{aligned} \{&\exists \mathtt{id1}, \mathit{Carrier}(\mathtt{id1}, \mathtt{aa}, \mathtt{paris}) \wedge \mathit{Departure}(\mathtt{lyon}, \mathtt{id1}) \wedge \mathit{Arrival}(\mathtt{paris}, \mathtt{id1}); \\ &\exists \mathtt{id2}, \mathit{Carrier}(\mathtt{id2}, \mathtt{ev}, \mathtt{lyon}) \wedge \mathit{Departure}(\mathtt{paris}, \mathtt{id2}) \wedge \mathit{Arrival}(\mathtt{lyon}, \mathtt{id2}); \\ &\exists \mathtt{id3}, \mathit{Carrier}(\mathtt{id3}, \mathtt{tc}, \mathtt{la})\} \end{aligned}$$

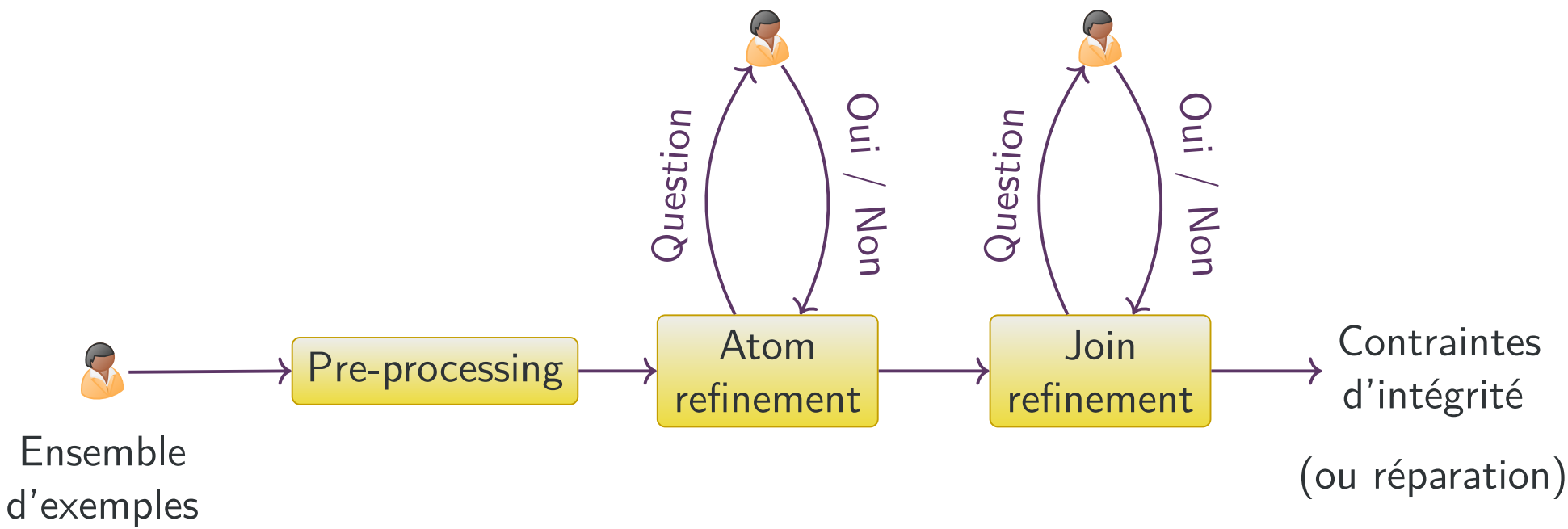
Removing superfluous atoms

$$\begin{aligned} &\mathit{Company}(\mathtt{c1}, \mathtt{aa}, \mathtt{paris}) \wedge \cancel{\mathit{Company}(\mathtt{c2}, \mathtt{ev}, \mathtt{lyon})} \\ &\wedge\ \mathit{Flight}(\mathtt{lyon}, \mathtt{paris}, \mathtt{c1}) \wedge \mathit{Flight}(\mathtt{paris}, \mathtt{lyon}, \mathtt{c2}) \wedge \mathit{TravelAgency}(\mathtt{a1}, \mathtt{tc}, \mathtt{la}) \\ &\Rightarrow \exists \mathtt{id1}, \mathit{Carrier}(\mathtt{id1}, \mathtt{aa}, \mathtt{paris}) \wedge \mathit{Departure}(\mathtt{lyon}, \mathtt{id1}) \wedge \mathit{Arrival}(\mathtt{paris}, \mathtt{id1}) \end{aligned}$$

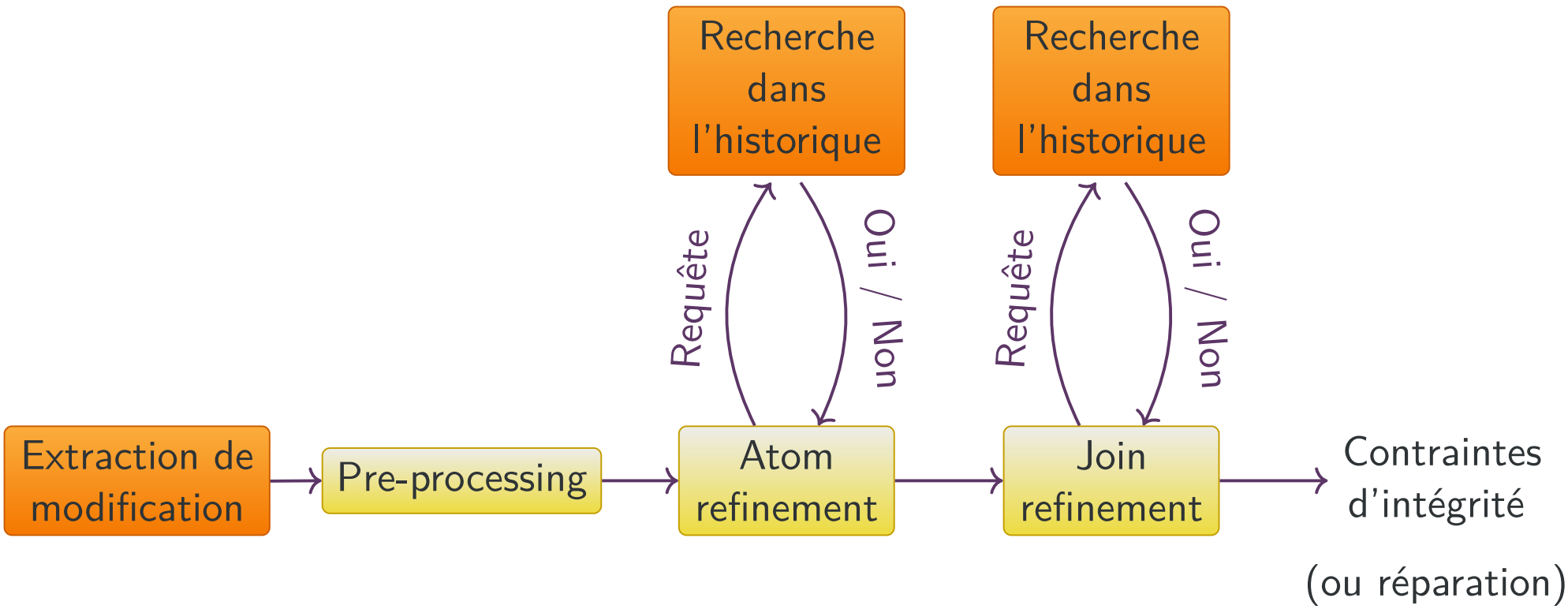
Join refinement

$$\mathit{Company}(\mathtt{c1}, \mathtt{aa}, \mathtt{paris}_1) \wedge \mathit{Flight}(\mathtt{lyon}, \mathtt{paris}_2, \mathtt{c1}) \Rightarrow \exists \mathtt{id1}, \mathit{Carrier}(\mathtt{id1}, \mathtt{aa}, \mathtt{paris}_3) \wedge \mathit{Departure}(\mathtt{lyon}, \mathtt{id1}) \wedge \mathit{Arrival}(\mathtt{paris}_4, \mathtt{id1})$$

Remplacer les questions par une recherche dans l'historique



Remplacer les questions par une recherche dans l'historique



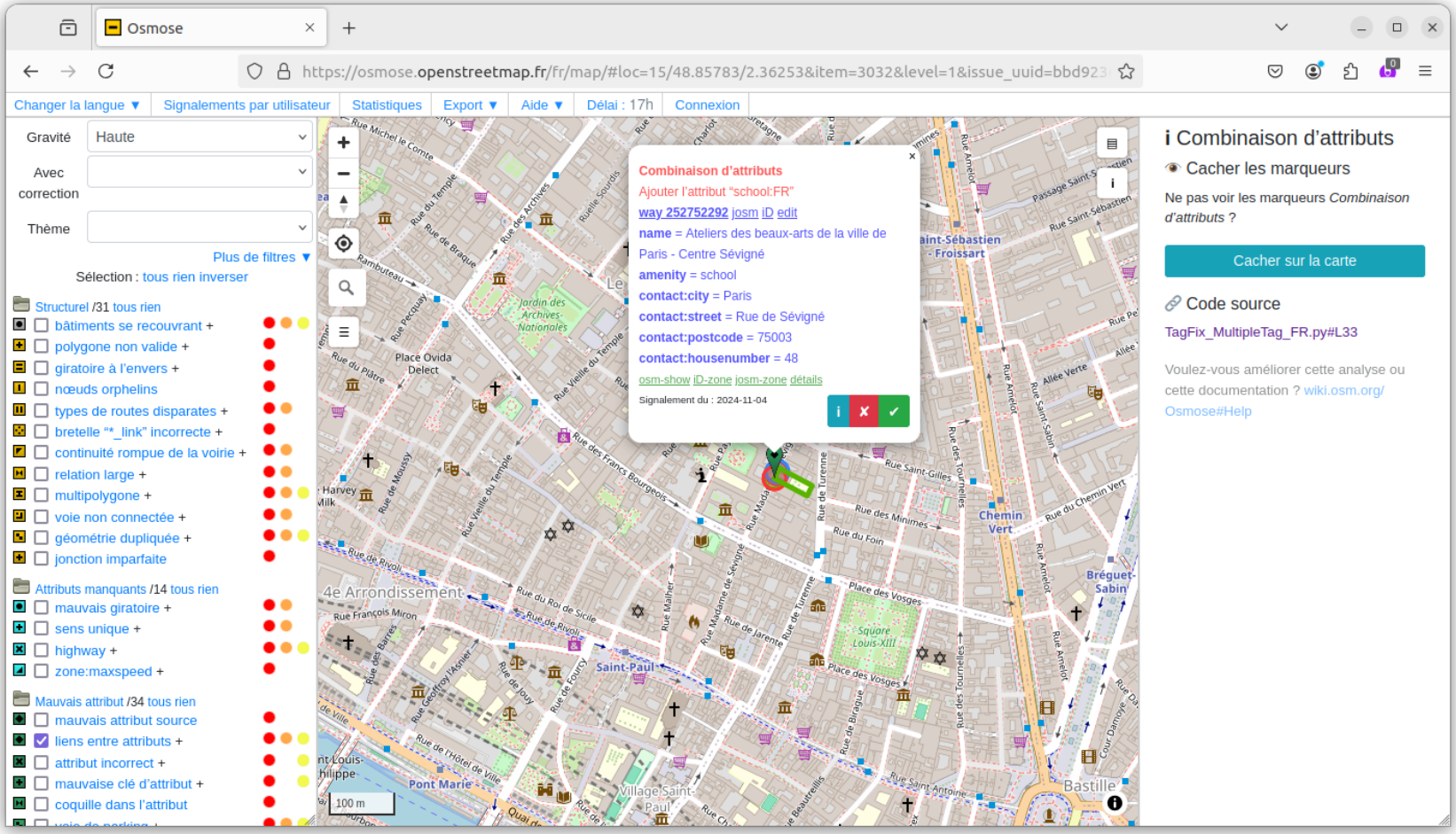
Interagir avec la communauté

Que faire des contraintes inférées ?

- Les ajouter sur le wiki,
- Automatiquement modifier la base de donnée.

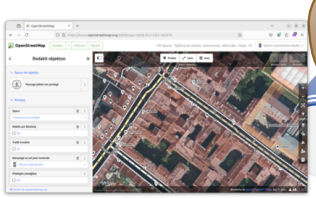
Que faire des contraintes inférées ?

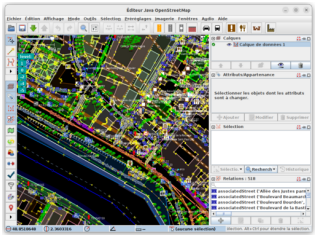
- ~~Les ajouter sur le wiki,~~
- ~~Automatiquement modifier la base de donnée.~~
- Les proposer à la communauté.

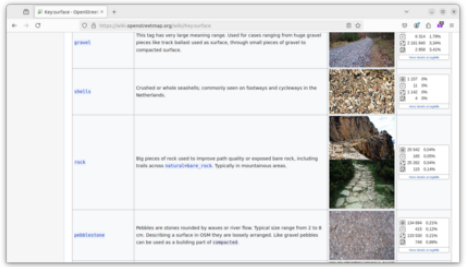


osmose.openstreetmap.fr

Inférer des règles pour la communauté





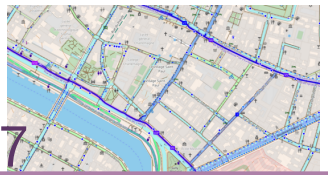


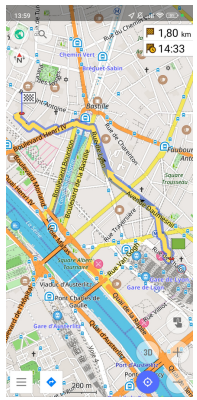
wiki.openstreetmap.org

Nous

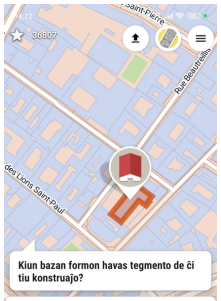
Osmose

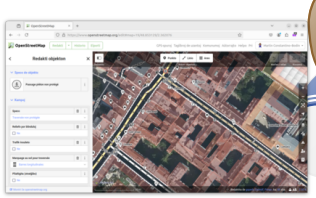
Données cartographiques

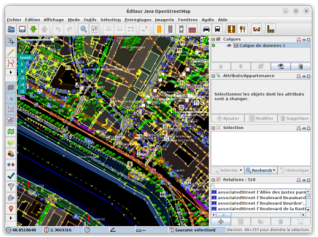


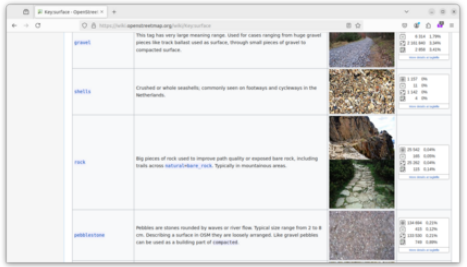


Merci pour votre attention







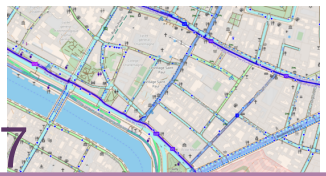


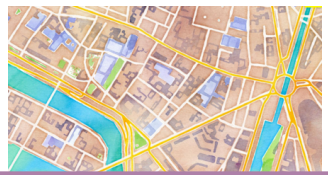
wiki.openstreetmap.org

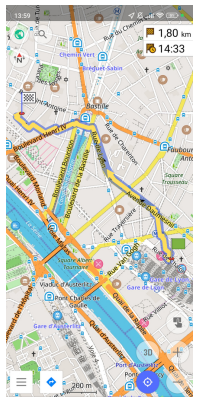
Nous

Osmose

Données cartographiques







1. OpenStreetMap
2. Contraintes d'intégrité
3. Interagir avec la communauté