



LI³DS

(Large Input 3D System)

Un entrepôt dynamique de données 3D

Mathieu Brédif
Eric Lemoine

<https://github.com/LI3DS>

Acquisition 3D : Plateformes



Mobile Mapping



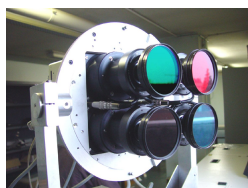
*Véhicules
légers*



Drones



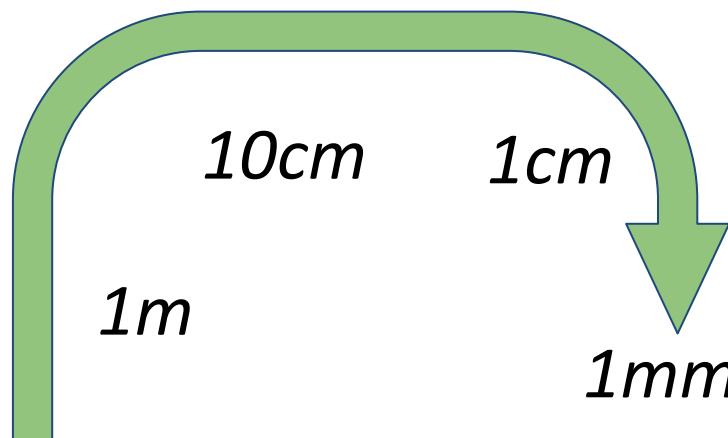
*Portatif, Sac à dos,
smartphone...*



Aérien



Satellite



Acquisition 3D : Capteurs

capteurs images



capteurs LiDAR



véhicule / plateforme



capteurs "georéférencement direct"

Acquisition 3D : Capteurs

Images



Nuages de point



véhicule / plateforme



Trajectoires

Acquisition 3D

Exemple du mobile mapping



Trajectoire

- centrale inertielle + GPS + odomètre → position et orientation à 200Hz

Images

- 9 images totalisant 28MPix tous les 2m

Laser

- 300 000 points/s
- 84 octets/point (non compressé)
- ~100Go/H d'acquisition
- 6H par jour ...

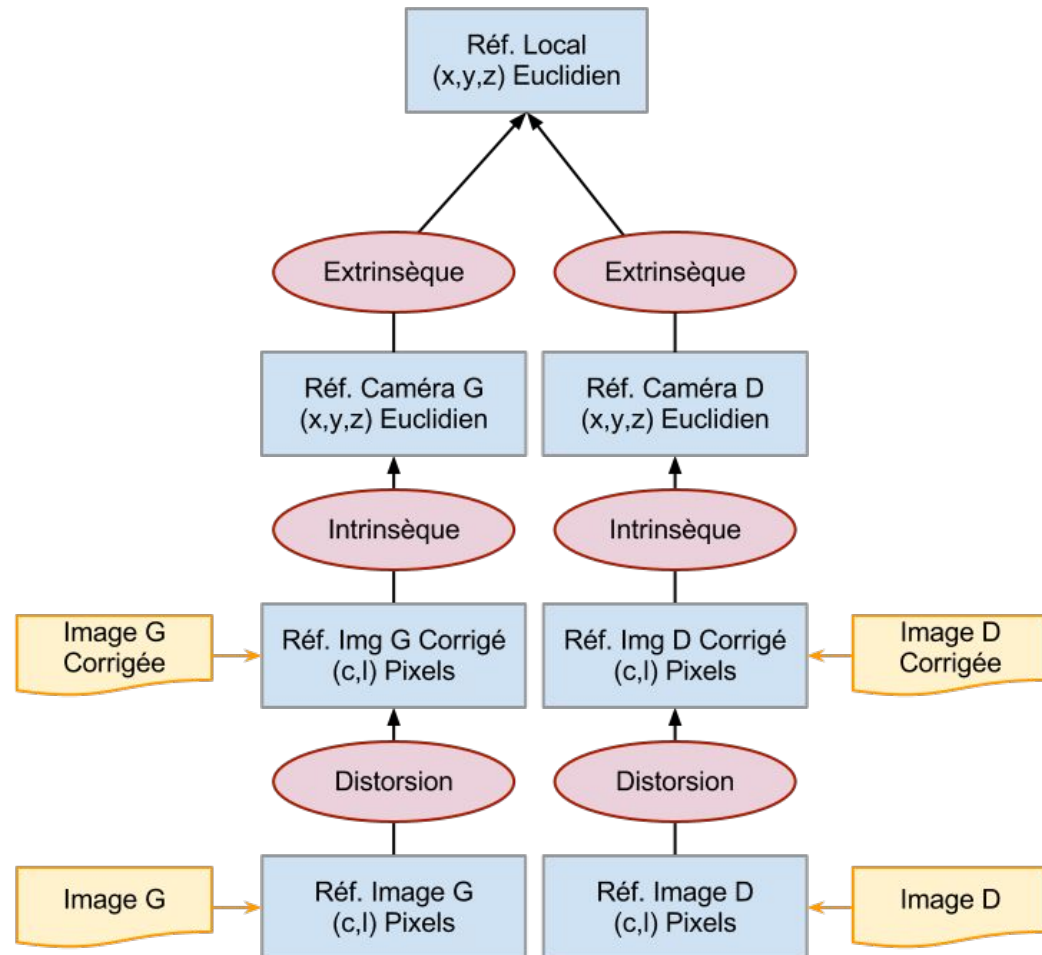


Caméra stéréo

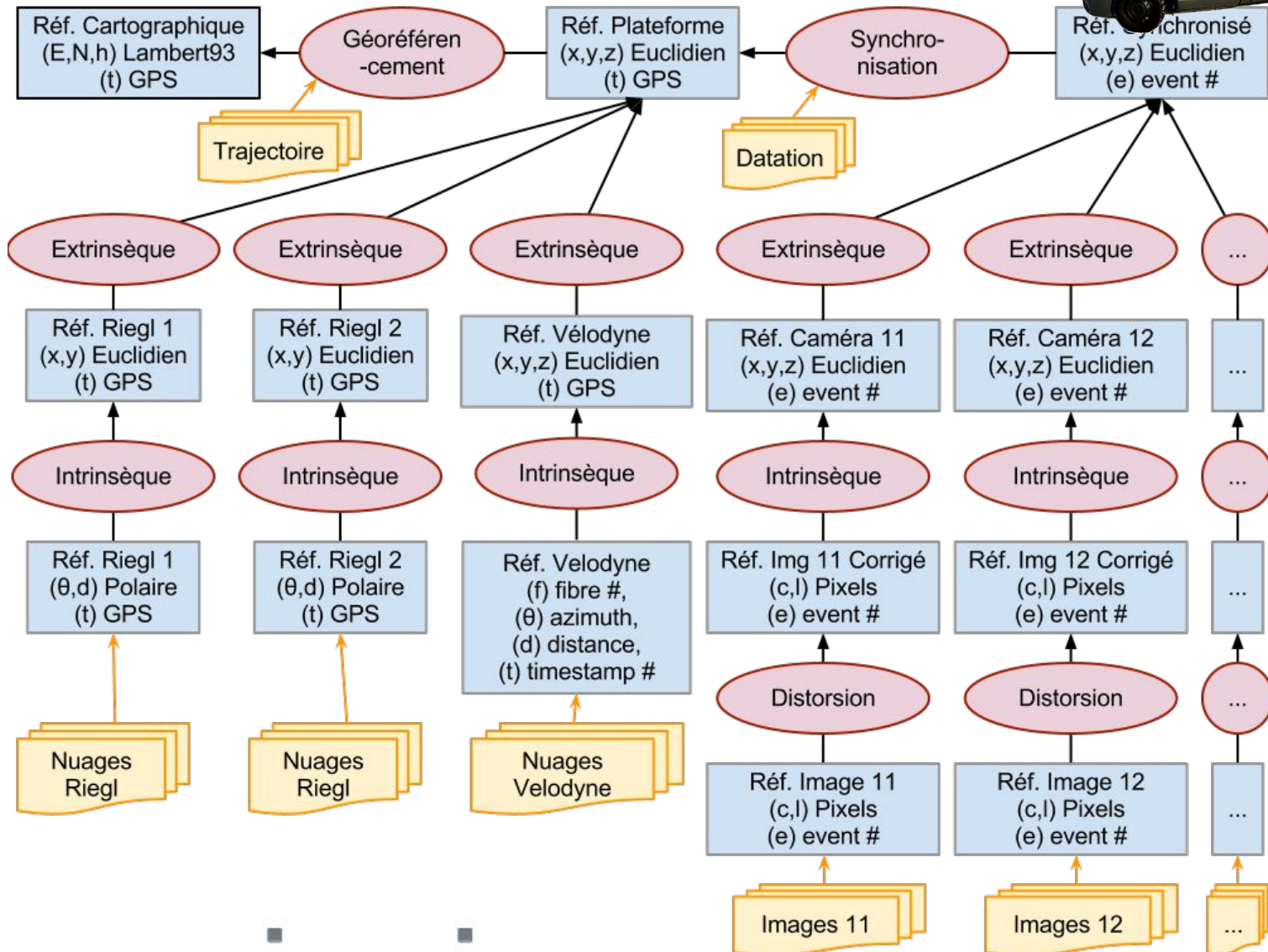
Trans-
formation

Donnée brute

Référentiel



Mobile Mapping



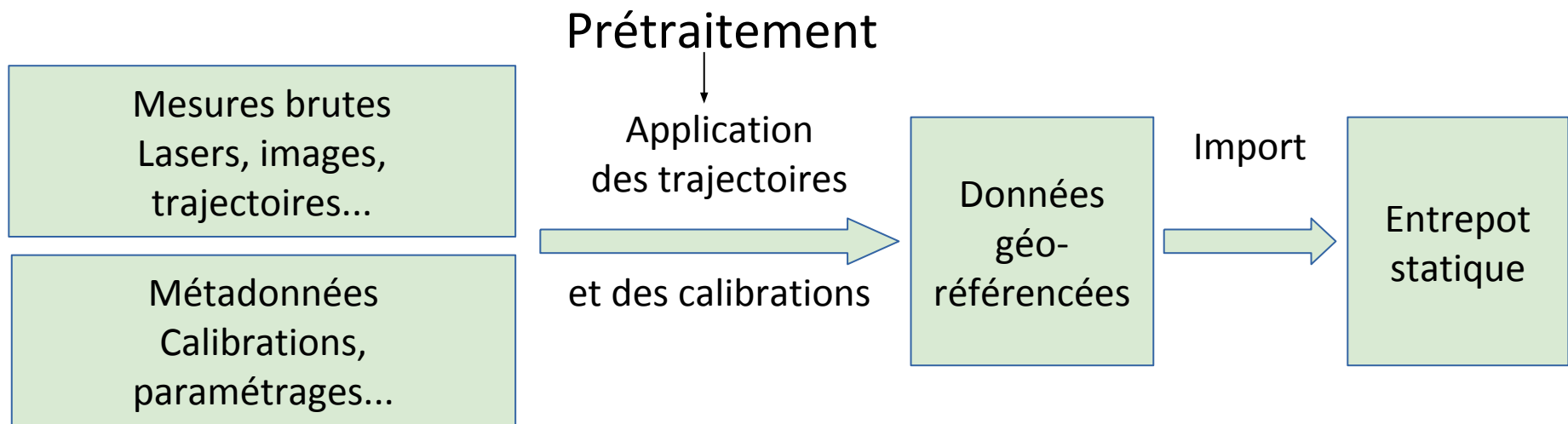
Entrepôt de données LI³DS

Stocke

- **Données d'acquisition 3D**: images, nuages de point, trajectoires
- **Métadonnées** : calibrations images et laser, réglages des capteurs, poses relatives...

L'objectif est de cataloguer et intégrer des **volumes** très importants de données hétérogènes issues de plateformes multiples : aérien, drone, mobile mapping, portatif...

Entrepôt statique



En pratique, il est faut republier (dupliquer?) la donnée à chaque:

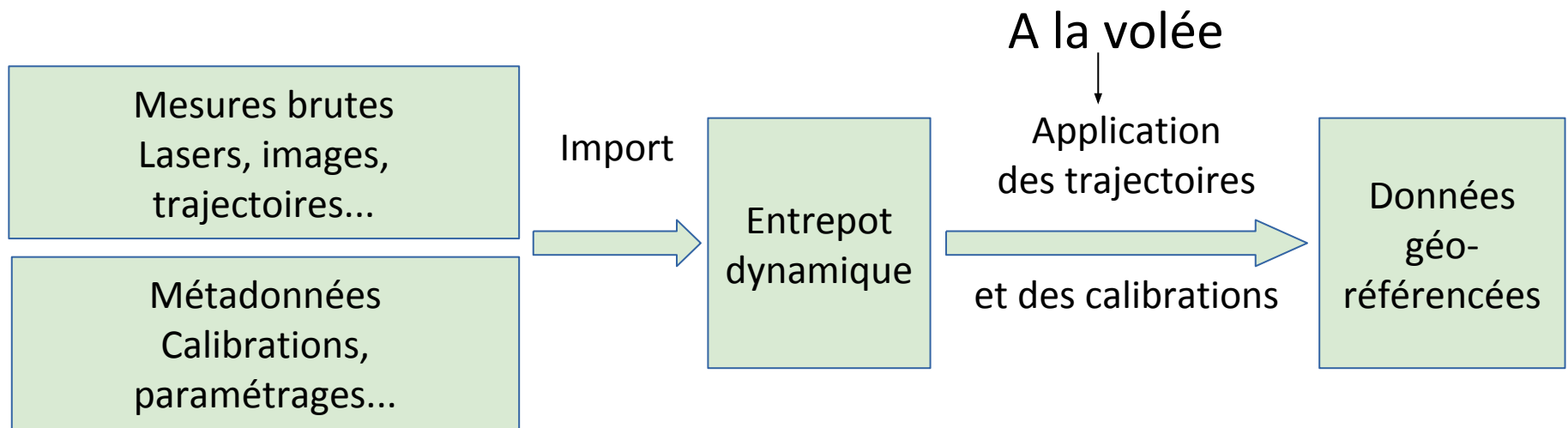
- recalibration

- un capteur a bougé (maintenance, remontage...)
- Amélioration de la calibration post-acquisition

- réestimation de trajectoire

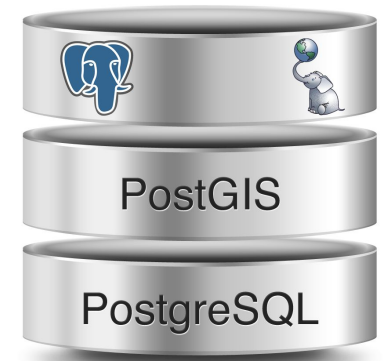
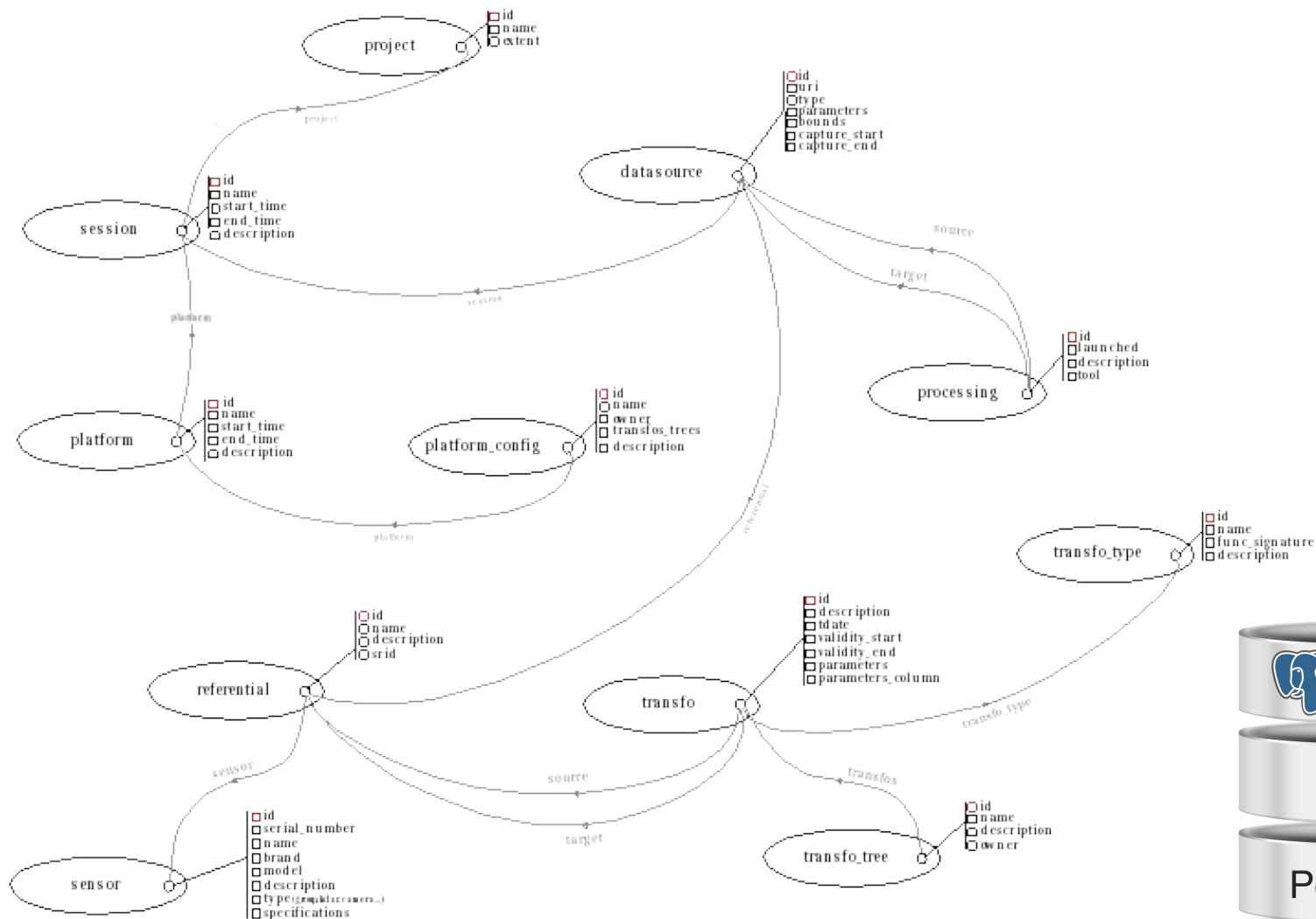
- Trajectographie GPS brute → Post traitée (différentiel)
- Compensation avec données externes...

Entrepôt dynamique

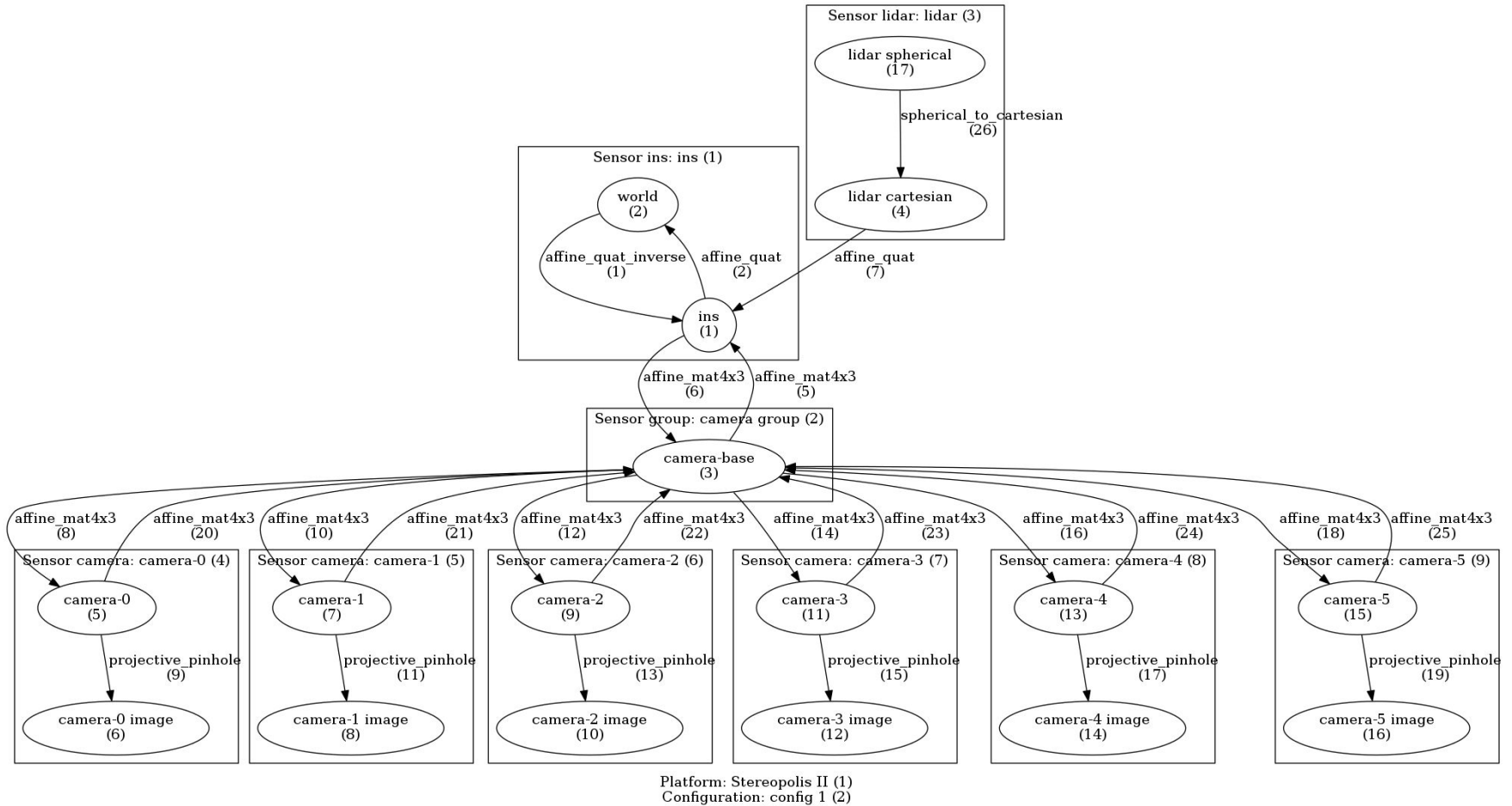


- Pas de duplication (entre données brutes + géoréférencées)
- Trajectoires et calibrations modifiables sans recalcul
- À la volée, mais caching/vue matérialisée si nécessaire (optimisation)
- Description fine des métadonnées
 - Historisation/versionnement des calibrations et des trajectoires (date de validité, sémantique...)

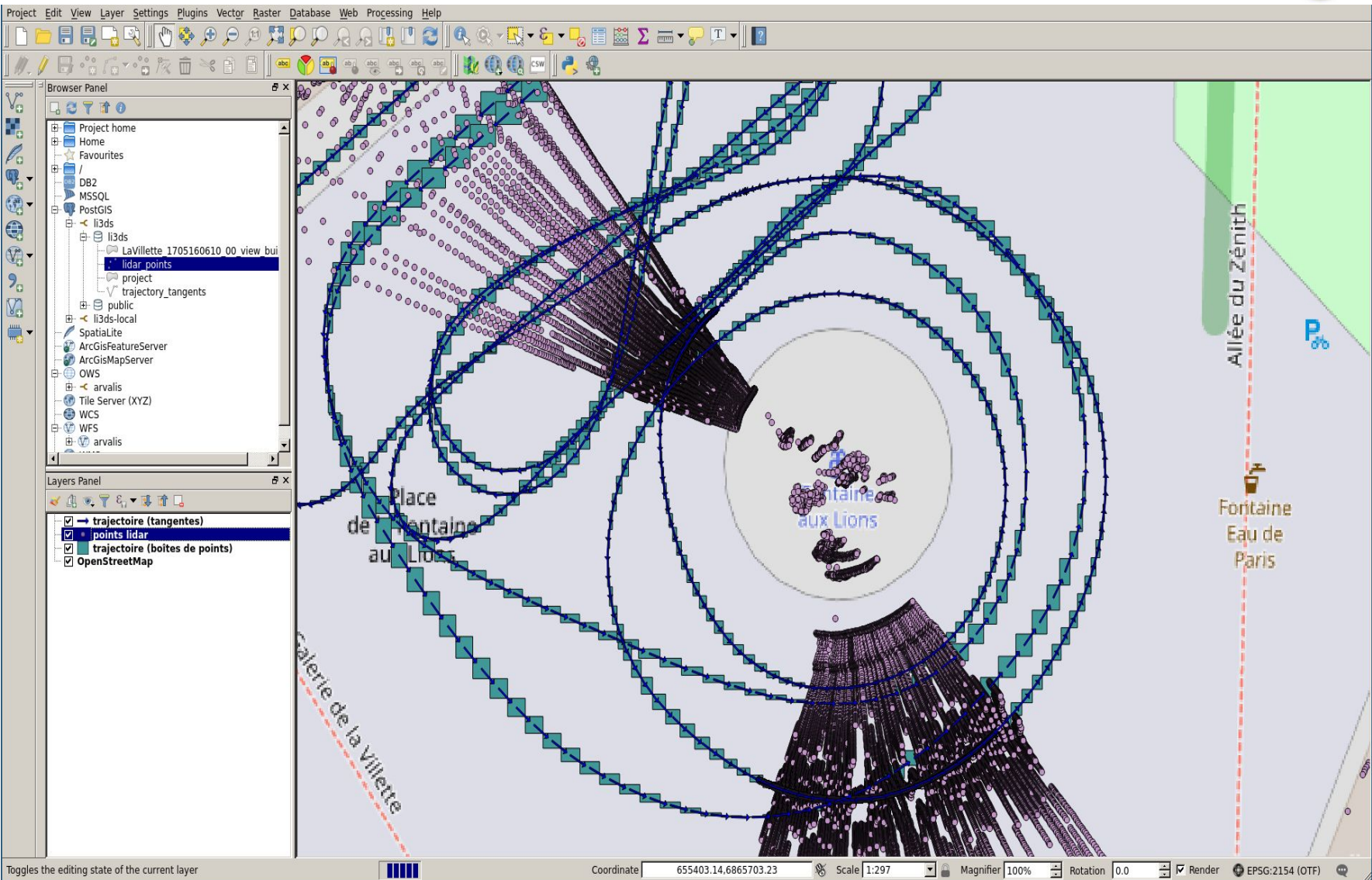
Modèle de données / métadonnées



Référentiels et transformations



Visualisation et analyse



Visualisation et analyse

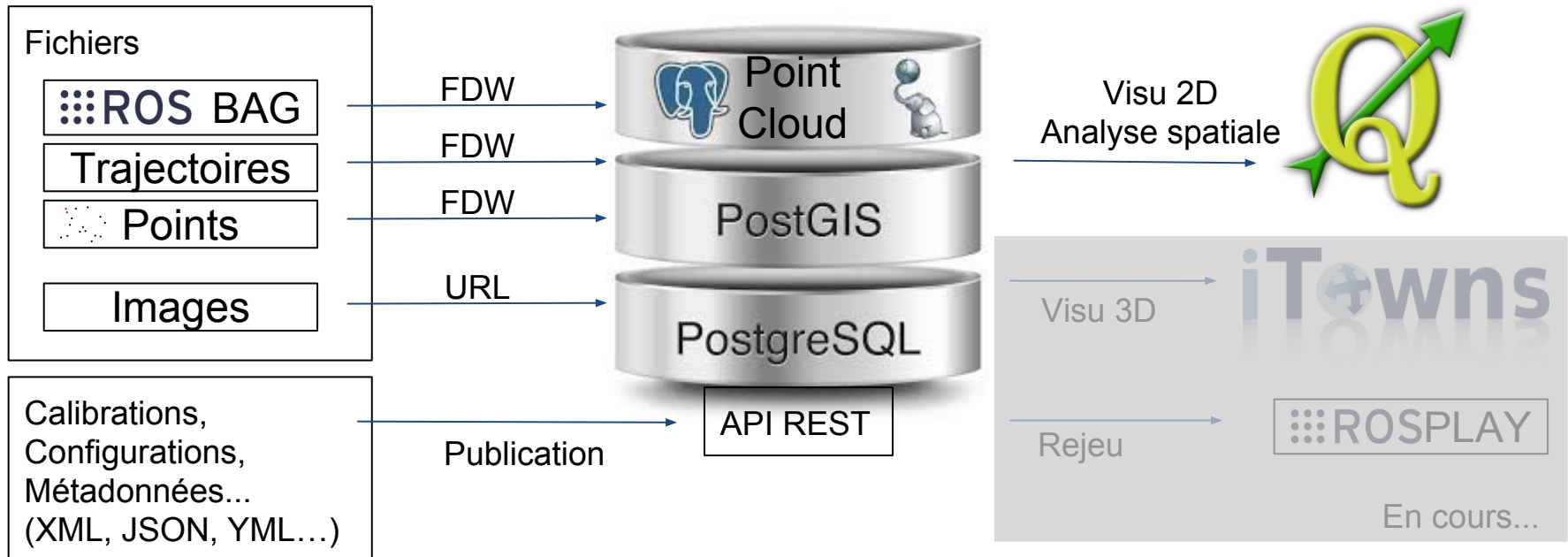


« Obtenir tous les points laser qui intersectent la Fontaine aux Lions »

```
WITH building AS (  
  SELECT 'BOX3D(655432.2 6865674.2 0, 655444.4 6865685.5 300)>::BOX3D  
  AS box  
) , times AS (  
  SELECT min(PC_PatchMin(tr.points, 'time')) AS tmin,  
         max(PC_PatchMax(tr.points, 'time')) AS tmax  
  FROM "LANDINS_20170516_075157_PP_view" tr, building b  
  WHERE PC_BoundingDiagonalGeometry(tr.points) &&& ST_Expand(b.box,  
30)  
)  
SELECT id, tpoints::pcpatch(100)  
  FROM "LaVillette_1705160610_00_view"  
    , PC_PatchAvg(points, 'time'::text) points_time  
    , times  
    , building  
    , li3ds.Transform(PC_Transform(points, 100), 2, 17, 2, points_time) tpoints  
WHERE PC_Patchavg(points, 'time') BETWEEN times.tmin AND times.tmax  
AND PC_BoundingDiagonalGeometry(tpoints) &&& building.box;
```

Implémentation

<https://github.com/LI3DS>



- **pg-li3ds** : modèle de donnée et fonctions PostgreSQL
- **api-li3ds** : API HTTP d'accès à l'entrepôt
- **cli-li3ds** : import des données (ligne de commande)
- **pointcloud** : fork de l'extension PostgreSQL
- **fdw-li3ds** : gestion des tables étrangères
- **docker-li3ds** : image Docker de l'entrepôt

Questions ?



LI³DS