

Tuilage de données géospatiales pour le métaverse

Présentation – 09.10.2024

Ian Escher

Table des matières

Tableau noir ?

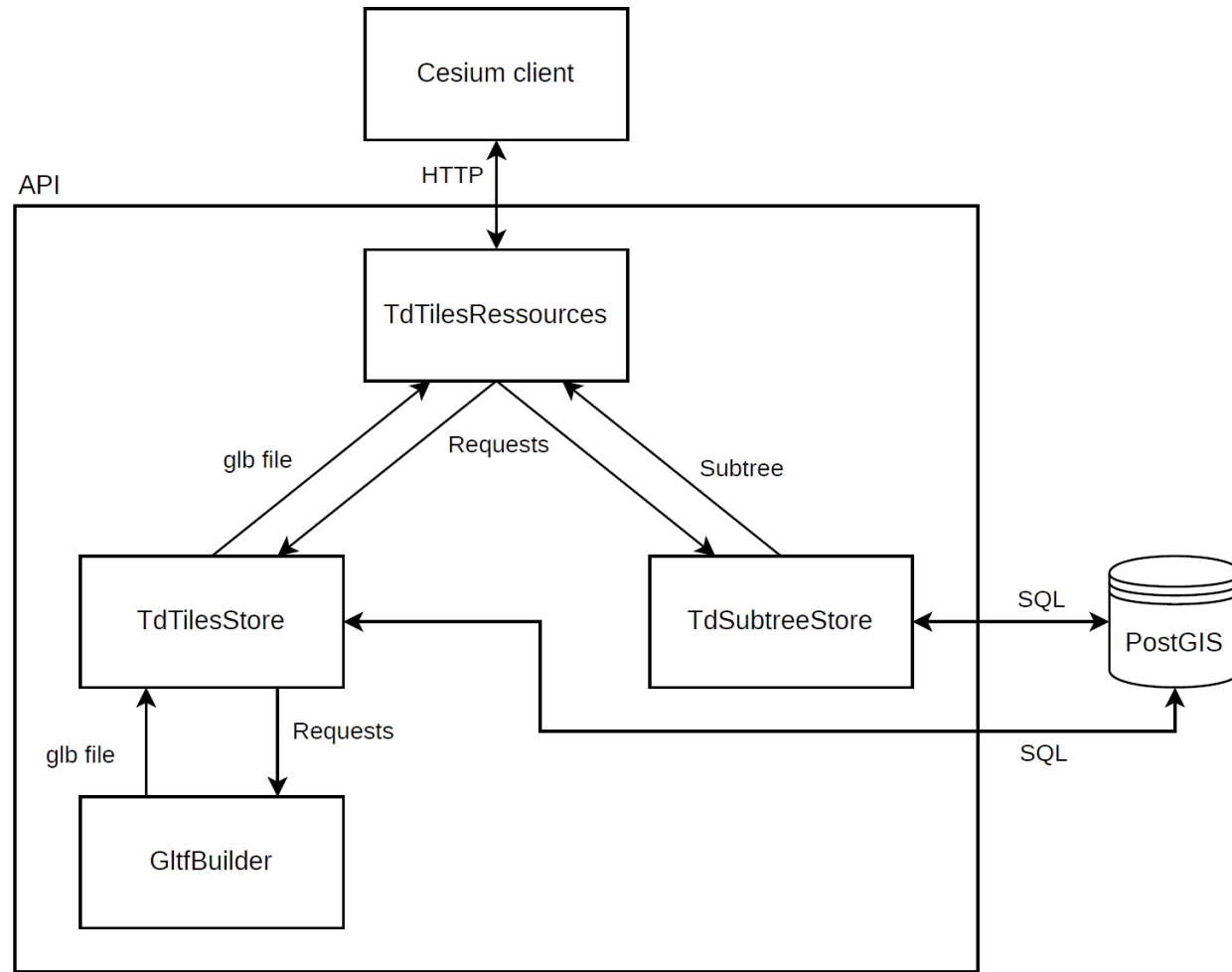
Osef de ce qu'ils connaissent déjà genr OSM buildings

- Introduction – de quoi je vais parler, plus les problèmesArchitecture
 - Globale
 - Nouvelles Tables dans la DB
 - TdTilesRessources/TdTilesStore/TdSubtreeStore -> leurs rôles
- Implicit Tiling -> S2 extensionSubtrees
- Availabilities
 - Morton Indexes
- Fonctionnement interne
- Fichiers glTF
- Levels of detail
- Transmission au client Cesium -> création des buffers + utilisation de l'outil checker
- Mon Ressentit -> CesiumJS et 3DTiles Next

Introduction – Travail effectué

Cahier des charges original très différent
Beaucoup de recherche
Conception d'algorithmes

Introduction – Architecture



Introduction – Architecture

Classes

TdTilesResources

S'occupe de la jointure entre le client CesiumJS et le logiciel à travers le traitement de requêtes HTTP.

TdTilesStore

Fabrique les fichiers glTF demandés par la classe TdTilesRessources.

TdSubtreeStore

Construit les Subtrees ainsi que leurs Availabilities et les donne à TdTilesRessources sous forme de JSON Binaire.

MortonIndexes

Contient les fonctions permettant de traiter avec des indexes de Morton.

Subtree

Classe permettant une représentation d'un Subtree avec une méthode de simplification et une de concaténation.

Availability

Classe permettant une représentation d'un Availability avec des méthodes permettant de le générer à partir d'un BitSet et de concaténation

Introduction – Architecture

Base de données

td_subtrees

Colonne	Type
morton_index	bigint
level	integer
binary_file	bytea

td_tile_gltf

Colonne	Type
x	bigint
y	bigint
level	integer
binary_file	bytea

td_tile_buildings_in_tile

Colonne	Type
id	bigint
compression	integer

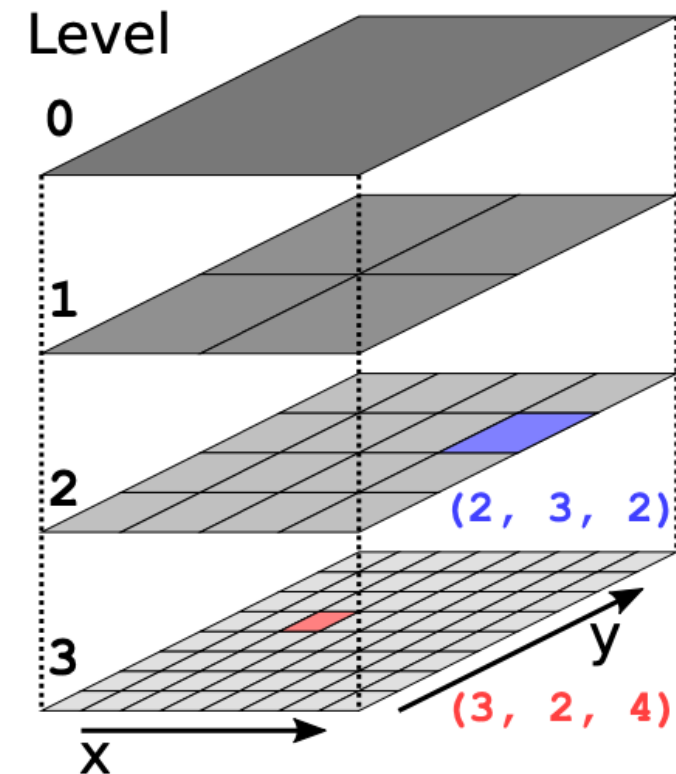
Implicit Tiling – But

Division implicite d'un espace 2D

Traitement de dataset volumineux

Quadtree ou Octree

Définit par un objet JSON



Implicit Tiling – 3DTILES_bounding_volume_S2

Création des bounding
volumes et geometric errors
automatique

Utilise une courbe de Hilbert

```
{
  "asset" : {
    "version" : "1.1"
  },
  "geometricError": 1200000,
  "root" : {
    "boundingVolume": {
      "region": [-3.14, -1.57, 3.14, 1.57, 0, 60]
    },
    "refine": "REPLACE",
    "geometricError": 1200000,
    "content": {
      "uri" : "/content/content_glb_{level}_{x}_{y}.glb"
    },
    "implicitTiling" : {
      "subdivisionScheme" : "QUADTREE",
      "subtreeLevels" : 4,
      "availableLevels" : 19,
      "subtrees" : {
        "uri" : "/subtrees/{level}.{x}.{y}.subtree"
      }
    }
  }
}
```


Implicit Tiling – Semi Implicit Tiling

Utilisation d'Implicit Tiling pour les Tuiles de haut niveau

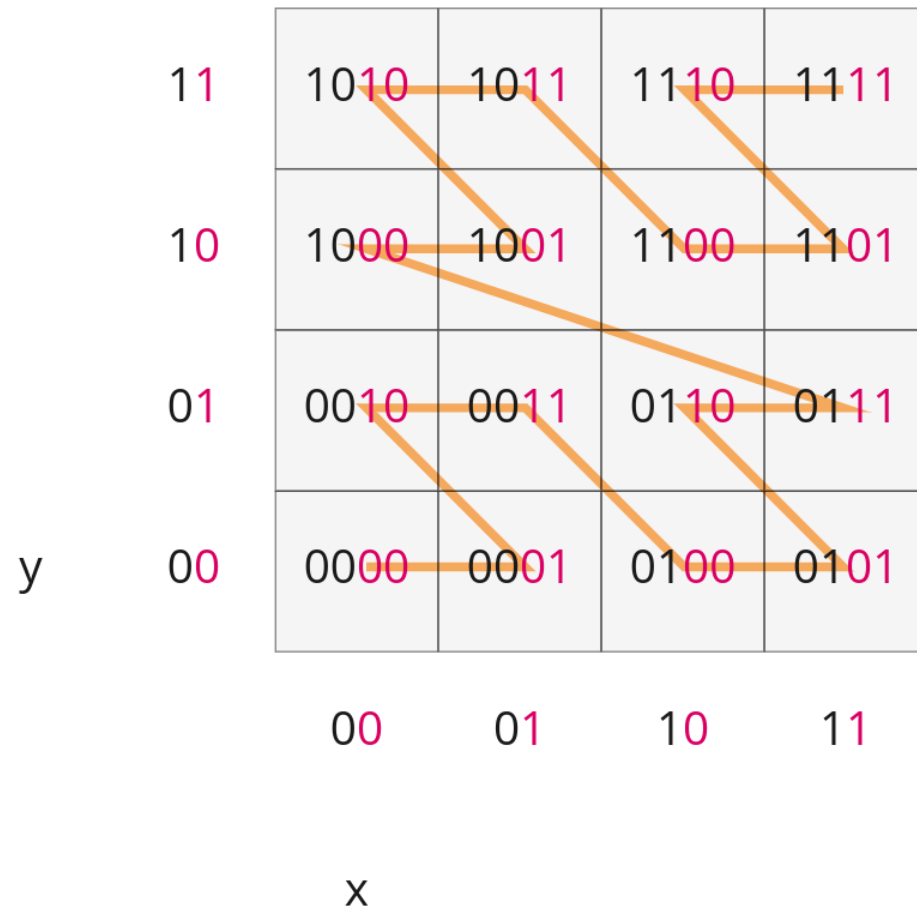
Utilisation de Tileset défini manuellement pour les tuiles contenant les bâtiments

Implicit Tiling – Morton Index

Même principe que la courbe de Hilbert

Utilisé par la spécification 3D Tiles Next pour les tuiles

Calcul des indices enfants simple



Subtrees – Introduction

Fichier utilisé par
Cesium

Définit la
disponibilité :

- Des tuiles
- De leur contenu
- De leurs enfants

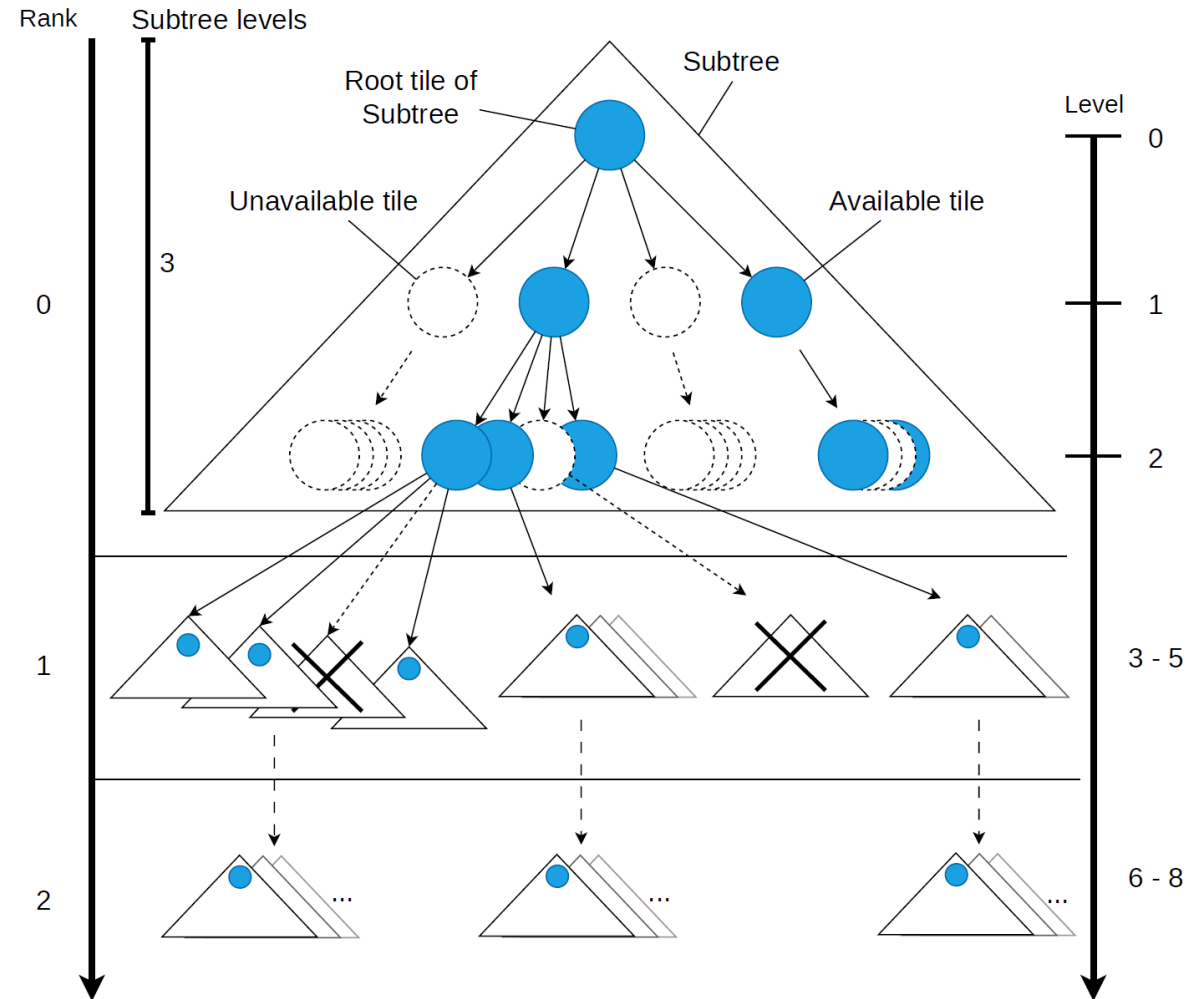


Subtrees – Représentation

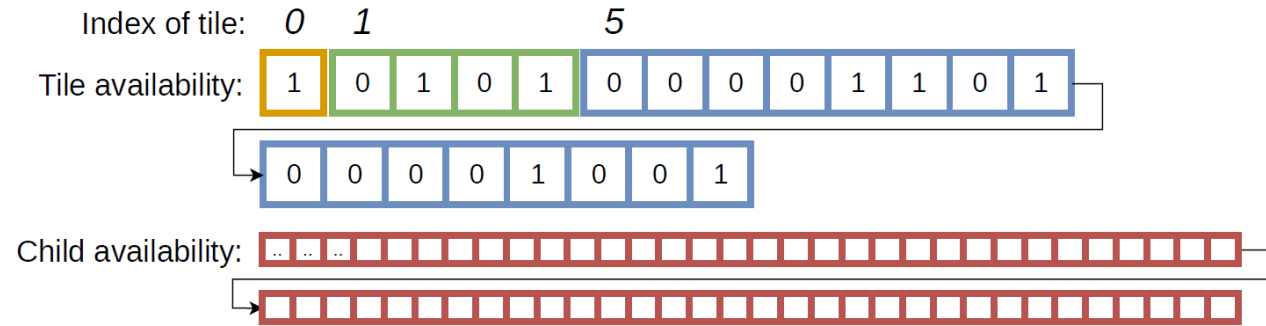
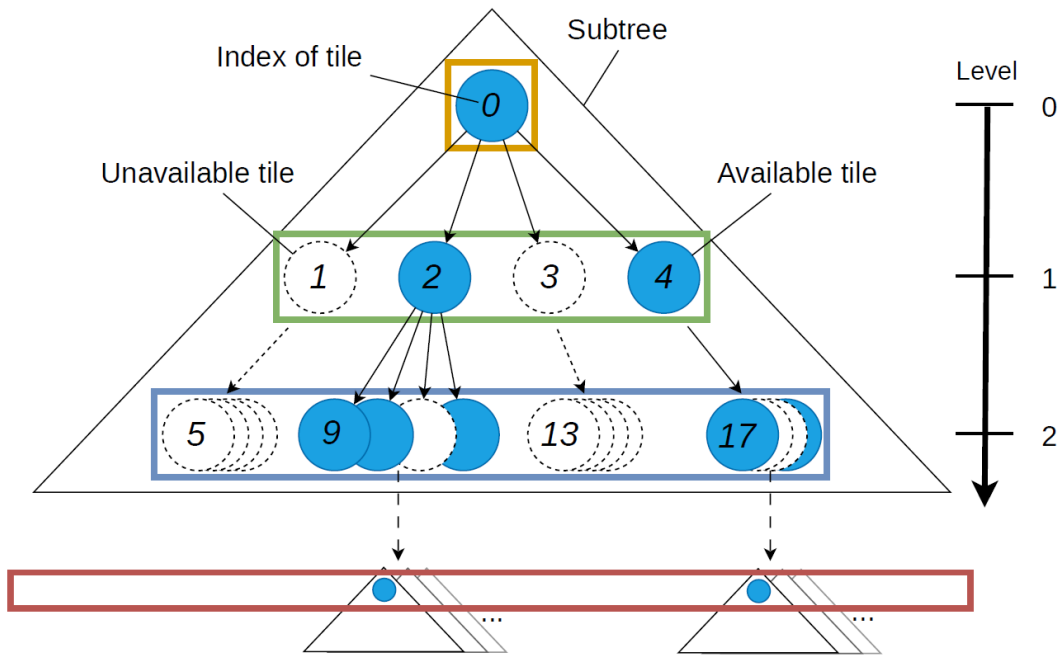
Hiérarchie

Division en plusieurs
rangs

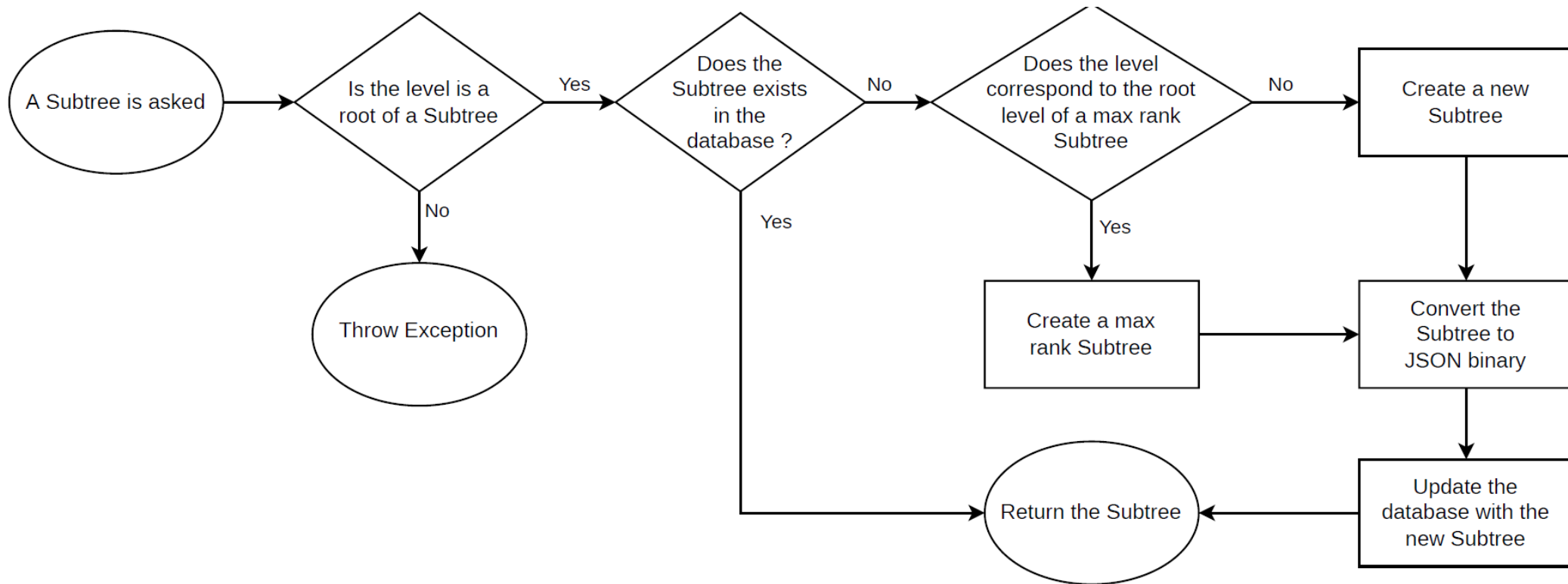
Liaison entre eux par
des nœuds
représentant le Subtree
enfant



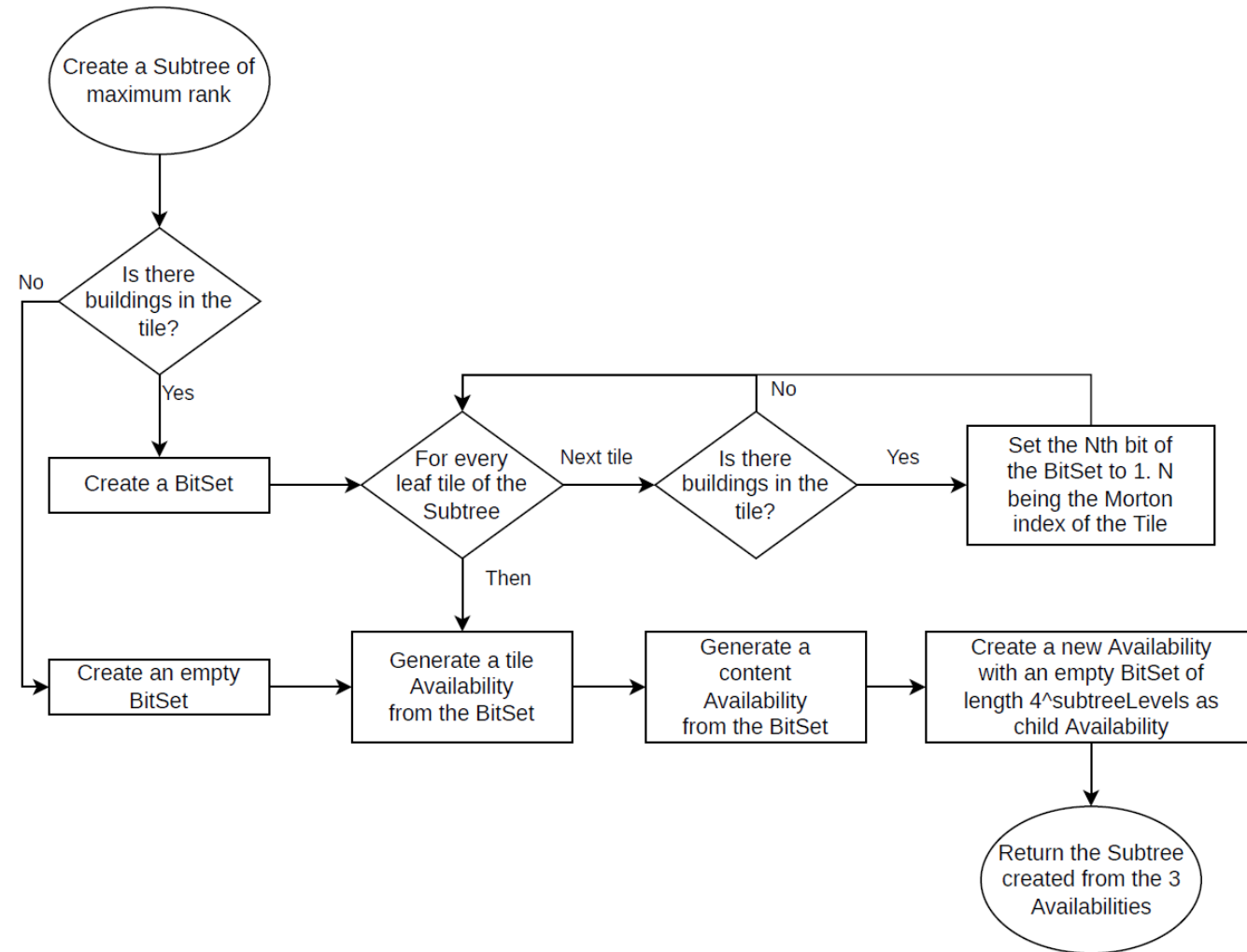
Availabilities – Représentation



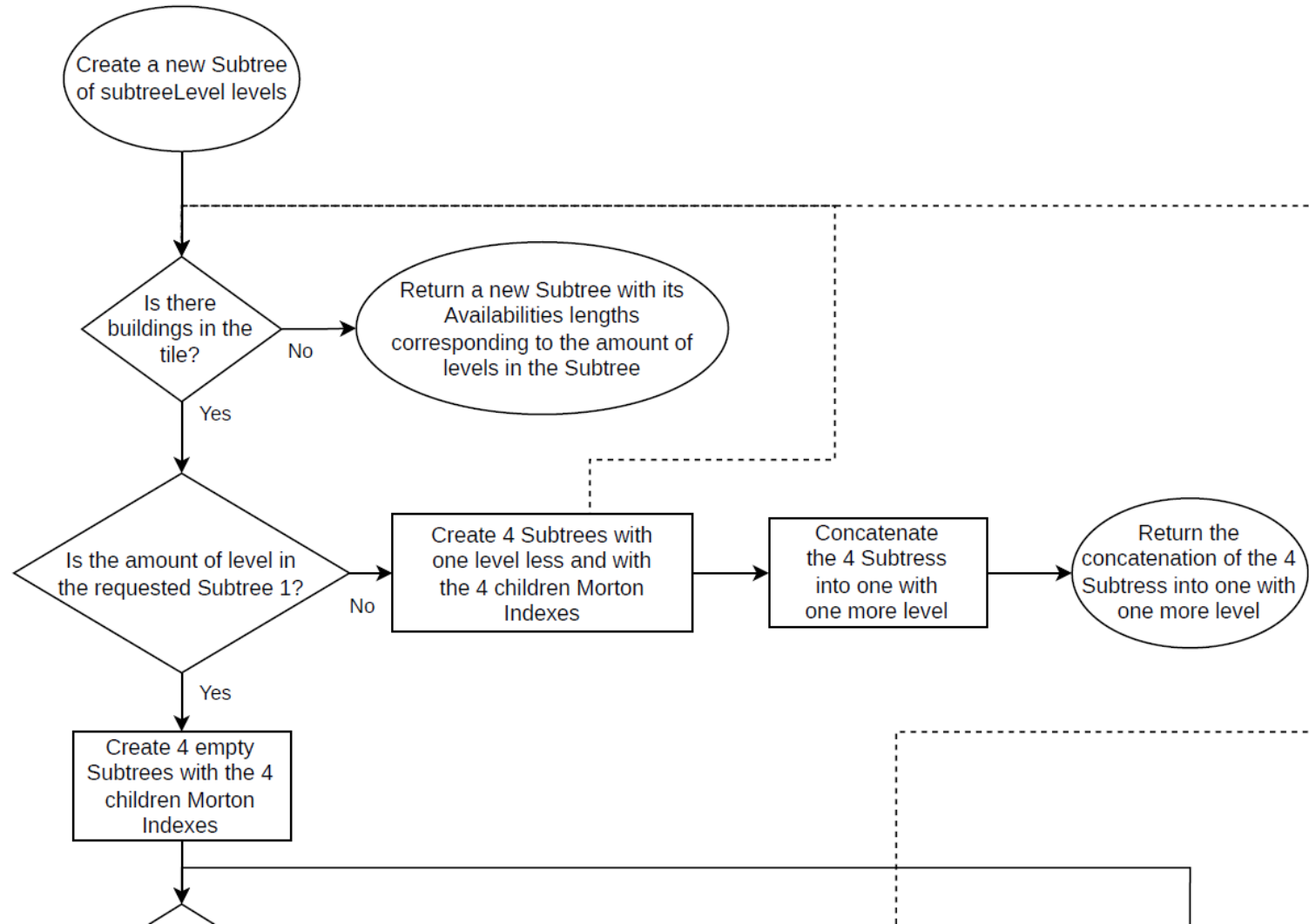
Fonctionnement interne – GetSubtree



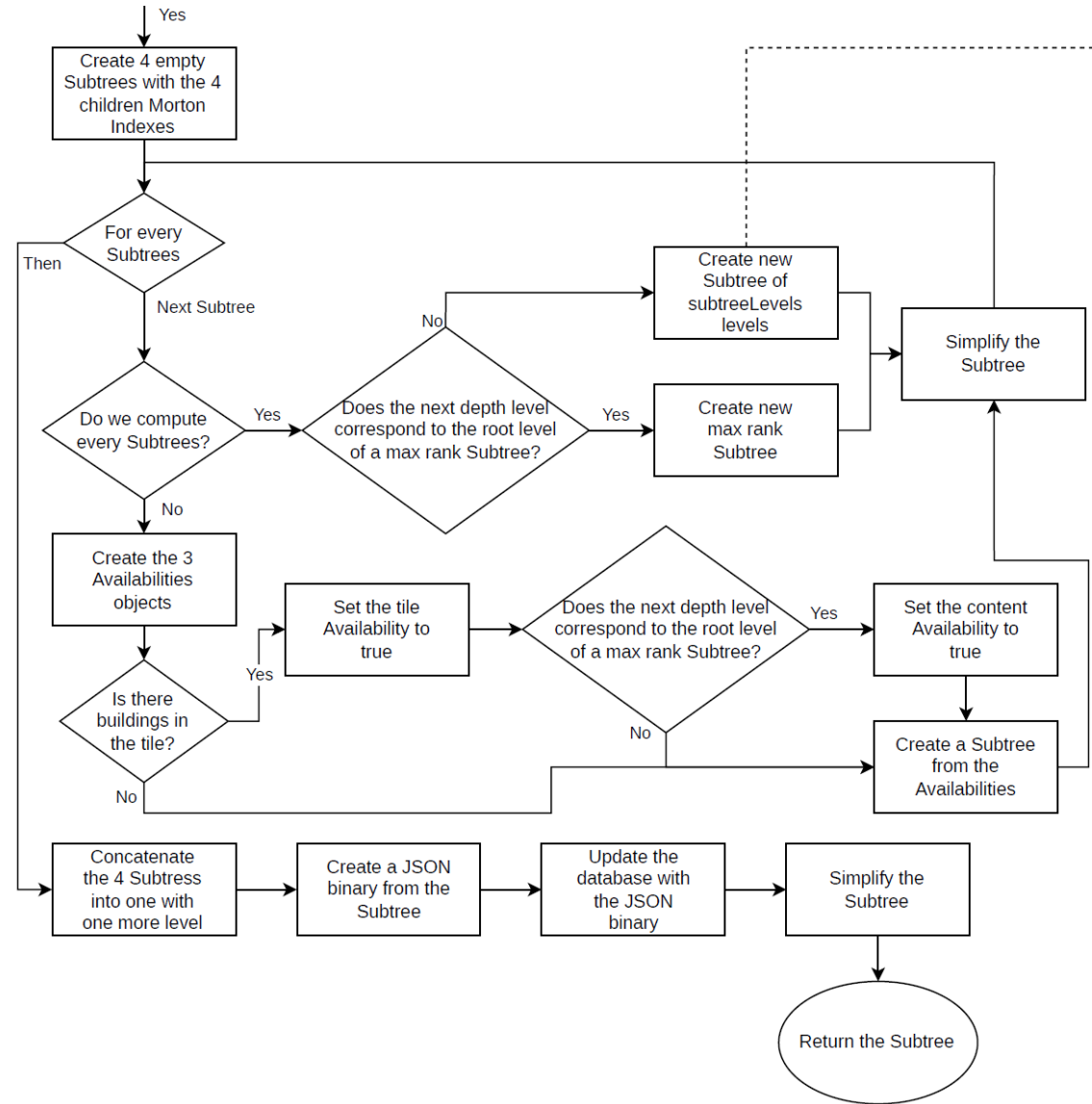
Fonctionnement interne – createMaxRankSubtree



Fonctionnement interne – createSubtree



Fonctionnement interne – createSubtree



Fonctionnement interne – Autres fonctions utilitaires

Concaténation de Subtree

Concaténation de Availability

Génération de Availability à partir d'un BitSet

Manipulations de Morton Index

Génération des fichiers glTF

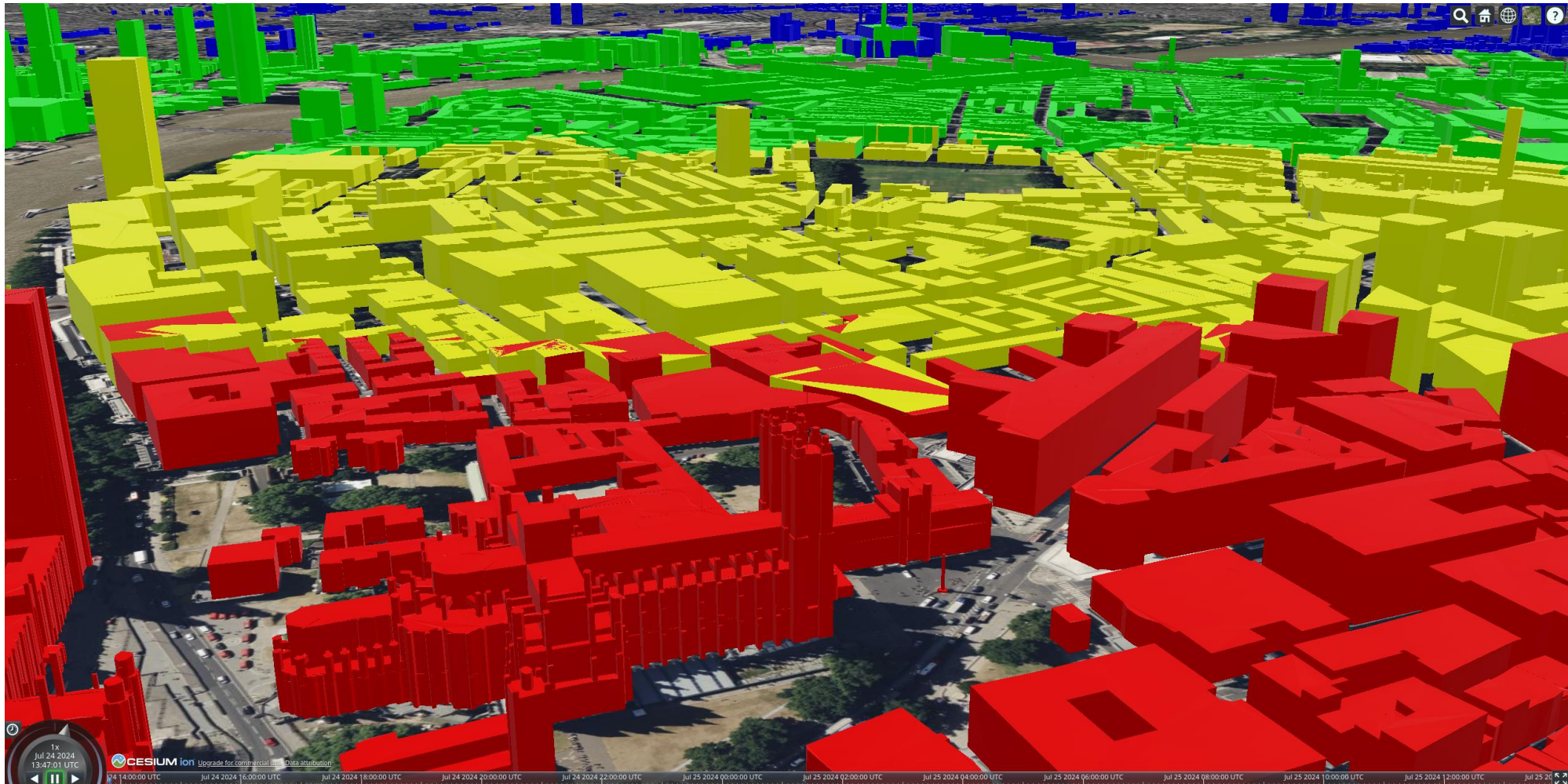
Basé sur la géométrie du bâtiment au sol

Basé sur les nombreux tags fournis par OSM

Génération de plusieurs niveaux de détails grâce à la fonction
TopologyPreservingSimplifier

Un fichier glTF par Tuile

Levels of details



Transmission au format JSON Binaire

24-byte header

Magic (uint32)	Version (uint32)	JSON byte length (uint64)	Binary byte length (uint64)	JSON (padded with spaces to 8-byte multiple)	Binary (padded with zeros to 8-byte multiple)
-------------------	---------------------	------------------------------	--------------------------------	----------------------------------------------------	-----------------------------------------------------

- **Magic** : 4 Bytes, *magic number* identifiant ce fichier comme un Subtree. A toujours une valeur de 0x74627573.
- **Version** : 4 Bytes, version du format binaire. Actuellement, le 24.07.2024, il s'agit de 1.
- **JSON Byte Length** : 8 Bytes, longueur de la partie JSON en Bytes.
- **Binary Byte Length** : 8 Bytes, longueur de la partie binaire en Bytes.

Utilisation de l'outil 3D Tiles Validator :

<https://github.com/CesiumGS/3d-tiles-validator>

Source : <https://github.com/CesiumGS/3dtiles/tree/main/specification/ImplicitTiling>

Demo

Mon ressenti

Documentation parfois incomplète

Certains comportements anormaux non discuté / documenté

Problèmes de performances «out of the box» de CesiumJS sans l'utilisation de Cesium ION

Peu d'extensions avec la version 1.1

Conclusion

Génération de Subtree robuste

Prototype fonctionnel mais lent

Reste un développement à être fait au niveau des performances ainsi que sur la représentation des objets 3D

Cahier des charges original

- *Objectifs fonctionnels*

- Le rendu 3D par Cesium est accessible par navigateur.
- Le niveau de détails est géré pour qu'il soit adapté à chaque distance du sol.
- Il est possible à l'utilisateur de se déplacer librement.
- L'utilisation du logiciel est fluide.

- *Objectifs non-fonctionnels*

- Les données sont représentées correctement. L'affichage 3D correspond à la réalité dans la mesure de la base de données mise à disposition.
- Si la direction prise est le rendu du terrain :
 - La densité de triangles du terrain dépend de l'éloignement de la caméra au terrain.
 - Les différentes tuiles du terrain sont combinées pour former un terrain uni.
 - Les images satellites sont appliquées à la texture du terrain.
- Si la direction prise est le rendu des bâtiments :
 - Les bâtiments ont leur forme décrite dans la base de données. Leurs murs et leur toit ont la bonne forme.
 - Les bâtiments ont une texture appliquée sur leurs faces ou une couleur dépendant de leur matériel.

- *Objectifs complémentaires "nice-to-have"*

- Il est possible à l'utilisateur de régler la qualité graphique qu'il souhaite. Il est possible à l'utilisateur de désactiver certaines parties de l'affichage 3D.
- Une combinaison des deux directions ensemble.
- Une intégration de la météo au rendu par le biais d'ombres et de colorimétrie.

Merci