

Urban Data Visualisation in a web browser

Jérémy Gaillard, Alexandre Vienne, Rémi Baume, Frédéric Pedrinis, Adrien Peytavie, Gilles Gesquière

Laboratoire d'InfoRmatique en Image et Systèmes d'information

jeremy.gaillard@liris.cnrs.fr



INSA



UNIVERSITÉ
LUMIÈRE
LYON 2

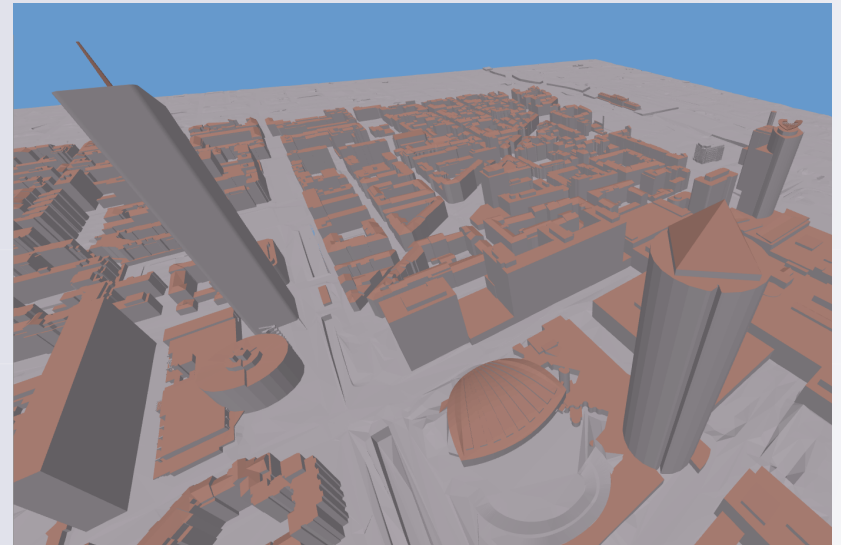


Introduction

- Web technologies mature enough to allow 3D application on the web
- Sharing virtual city models
 - Collaborative urban management
 - Simulation
 - Training
- Use cases need both semantic and geometric information
- Importance of standards

Introduction

- Challenges:
 - Manage gigabytes of data
 - Aggregate data from multiple data stores
 - Allow multiple representation of the data depending on the user's need



Introduction

Our contributions:

1. Fluid visualisation
2. Standard-based solution
 - Access to multiple sources of open data
3. Access to semantic information
4. Data loading prioritisation
5. Progressive textures



Overview




1. Related works
2. Urban data viewer framework
 - a. General architecture
 - b. Data preparation
 - c. Client architecture
 - d. Scheduler
 - e. Additional data
3. Results
4. Conclusion



Related works

3D rendering on the web




3D rendering on the web: declarative approach vs imperative approach

	2D	3D
<p><u>Declarative</u> Scene Graph Part of HTML Document DOM Integration CSS/Events</p>		<p>Declarative 3D for the Web Architecture Community Group</p> 
<p><u>Imperative</u> Procedural API Drawing Context Flexible</p>	<p><canvas></p>	

Extracted from Evans, A., Romeo, M., Bahrehmand, A., Agenjo, J., and Blat, J. 2014. **3D graphics on the web: A survey**. Computers & Graphics

3D rendering on the web

3D rendering on the web: declarative approach vs imperative approach

	2D	3D
<p><u>Declarative</u> Scene Graph Part of HTML Document DOM Integration CSS/Events</p>		<p>Declarative 3D for the Web Architecture Community Group</p> 
<p><u>Imperative</u> Procedural API Drawing Context Flexible</p>	<p><canvas></p>	

Our solution

Extracted from Evans, A., Romeo, M., Bahrehmand, A., Agenjo, J., and Blat, J. 2014. *3D graphics on the web: A survey*. *Computers & Graphics*

Related work

3D GIS data visualisation



2

Building

City

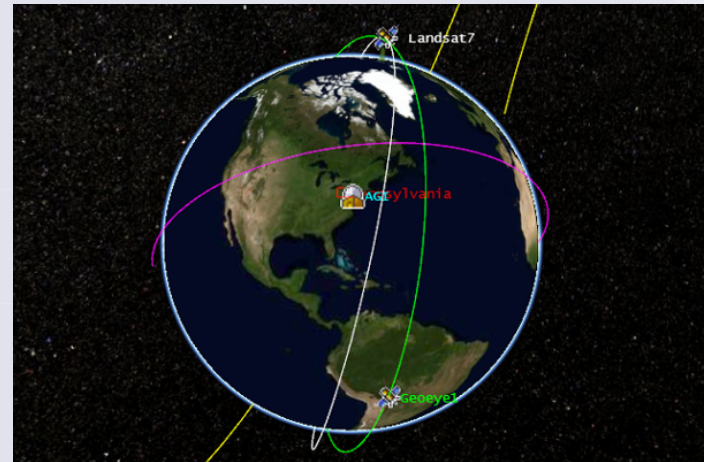
Globe

Scale

- [1] Enodo <http://www.enodo.fr/>
- [2] Vizicities <http://vizicities.com/>
- [3] Cesium <http://cesiumjs.org/>



1



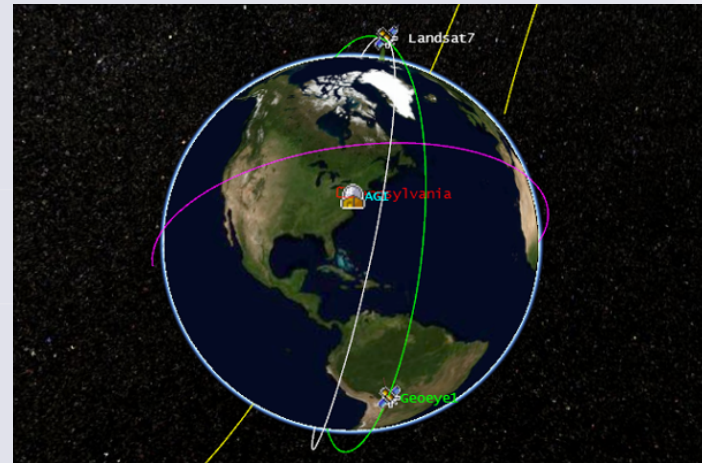
3

Related work

3D GIS data visualisation



- [1] Enodo <http://www.enodo.fr/>
- [2] Vizicities <http://vizicities.com/>
- [3] Cesium <http://cesiumjs.org/>





Urban data viewer framework

Overview

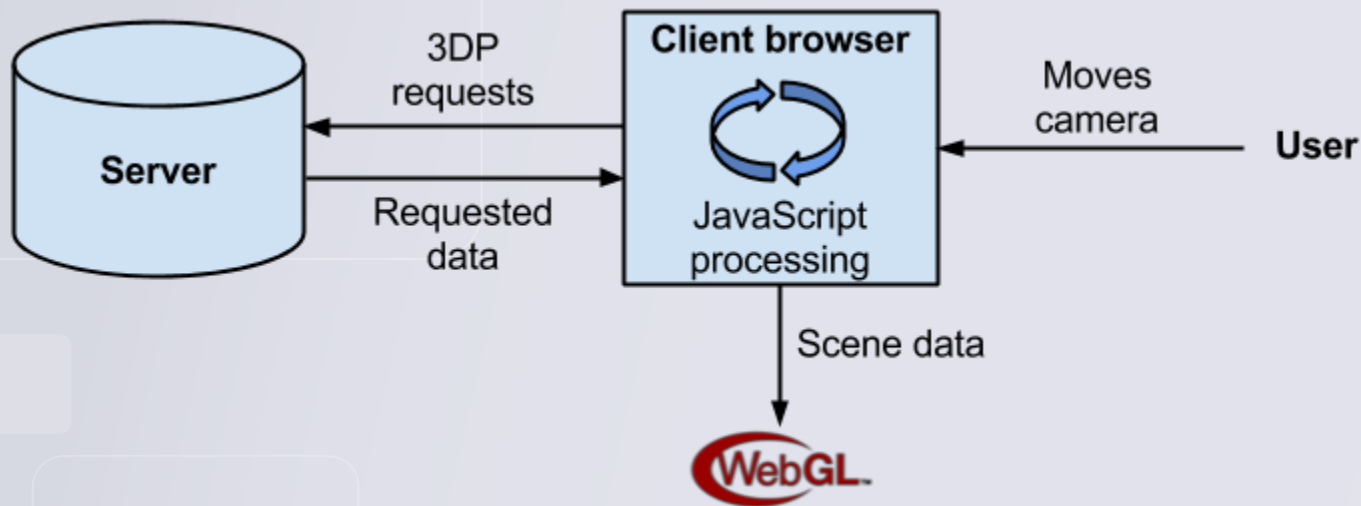
Urban data viewer framework

- a. General architecture
- b. Data preparation
- c. Client architecture
- d. Scheduler
- e. Additional data

Urban data viewer framework

General architecture

Our framework's architecture

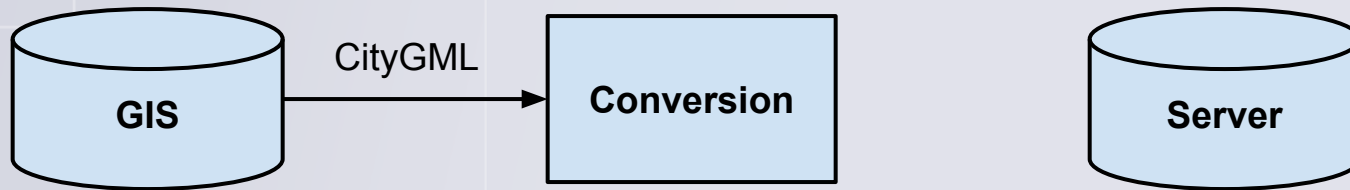


Implementation of the 3D Portrayal pending standard

3DP standard working group: <http://www.opengeospatial.org/projects/groups/3dpswg>

Data preparation

CityGML is not an efficient format for city visualisation



Data preparation

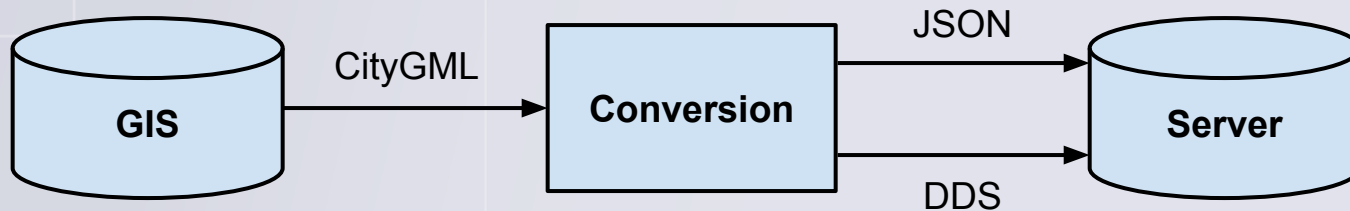
CityGML is not an efficient format for city visualisation



- JSON:
 - compact
 - easy to parse in JavaScript

Data preparation

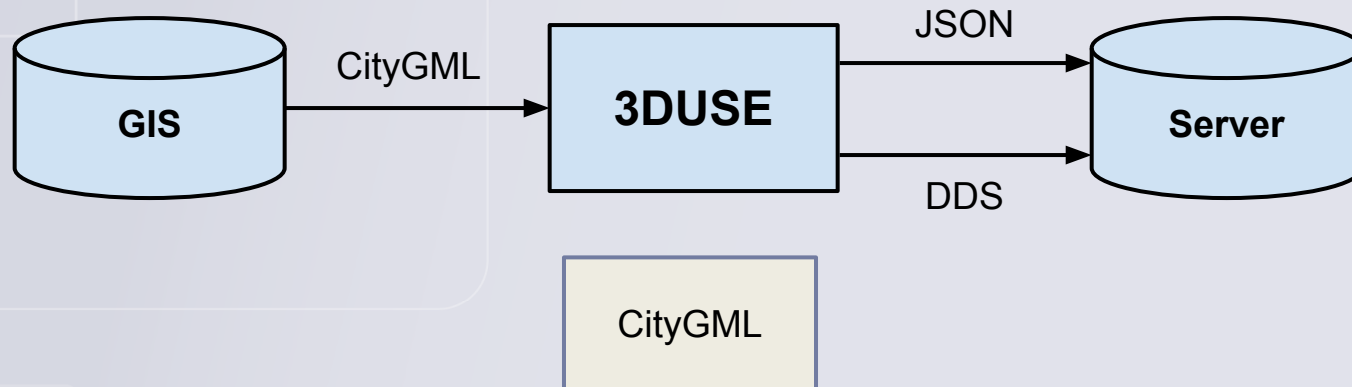
CityGML is not an efficient format for city visualisation



- JSON:
 - compact
 - easy to parse in JavaScript
- DDS: texture format natively read by the GPU
- Progressive textures with mipmaps

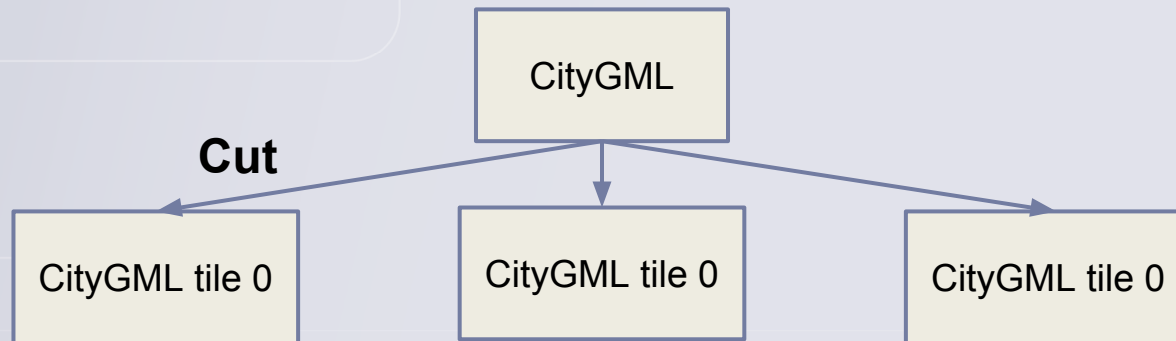
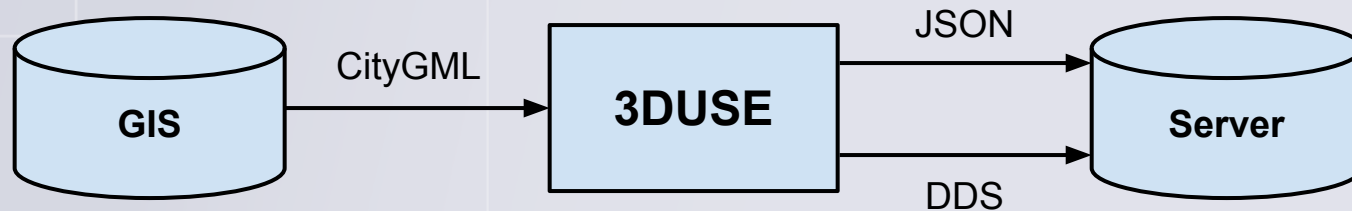
Data preparation

3DUSE: LIRIS' software to read and process CityGML



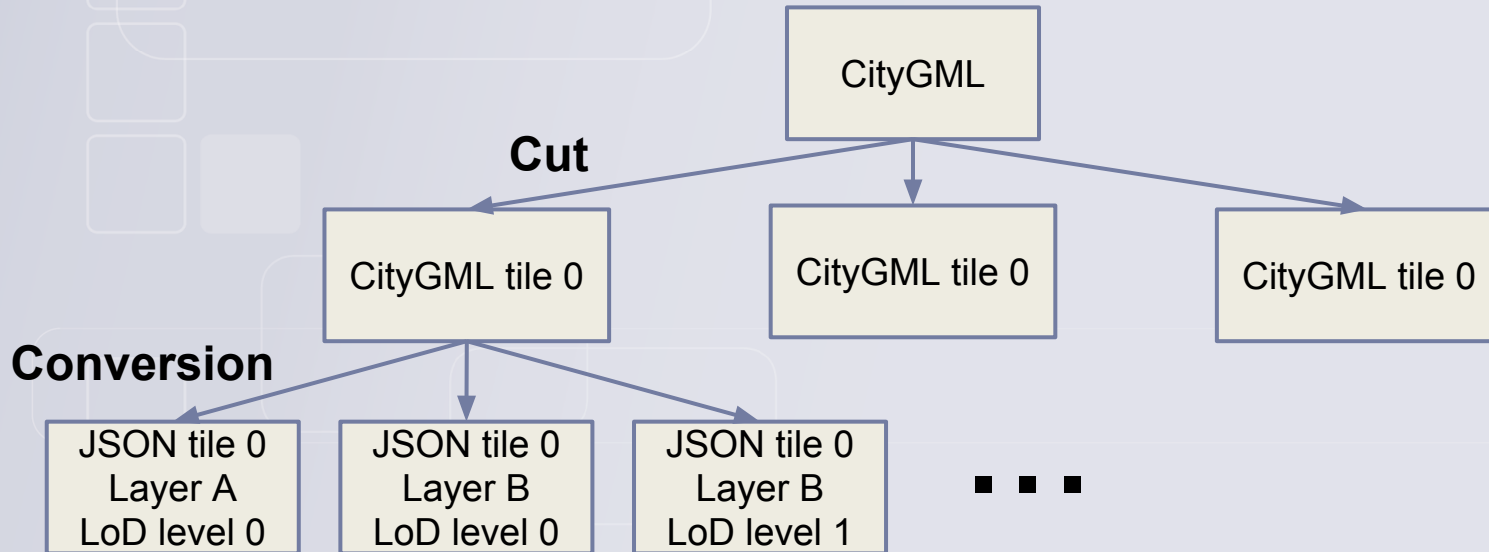
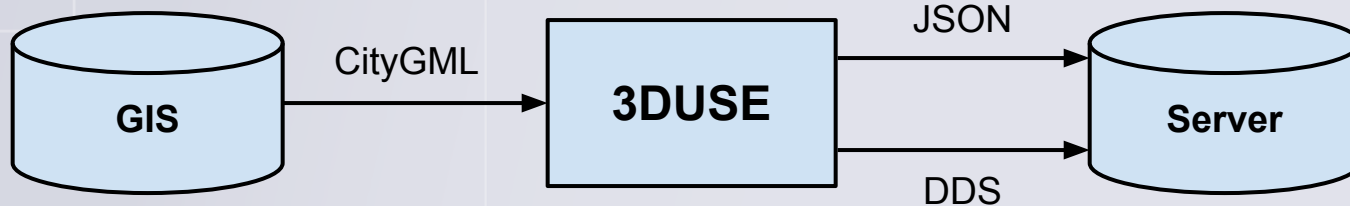
Data preparation

3DUSE: LIRIS' software to read and process CityGML



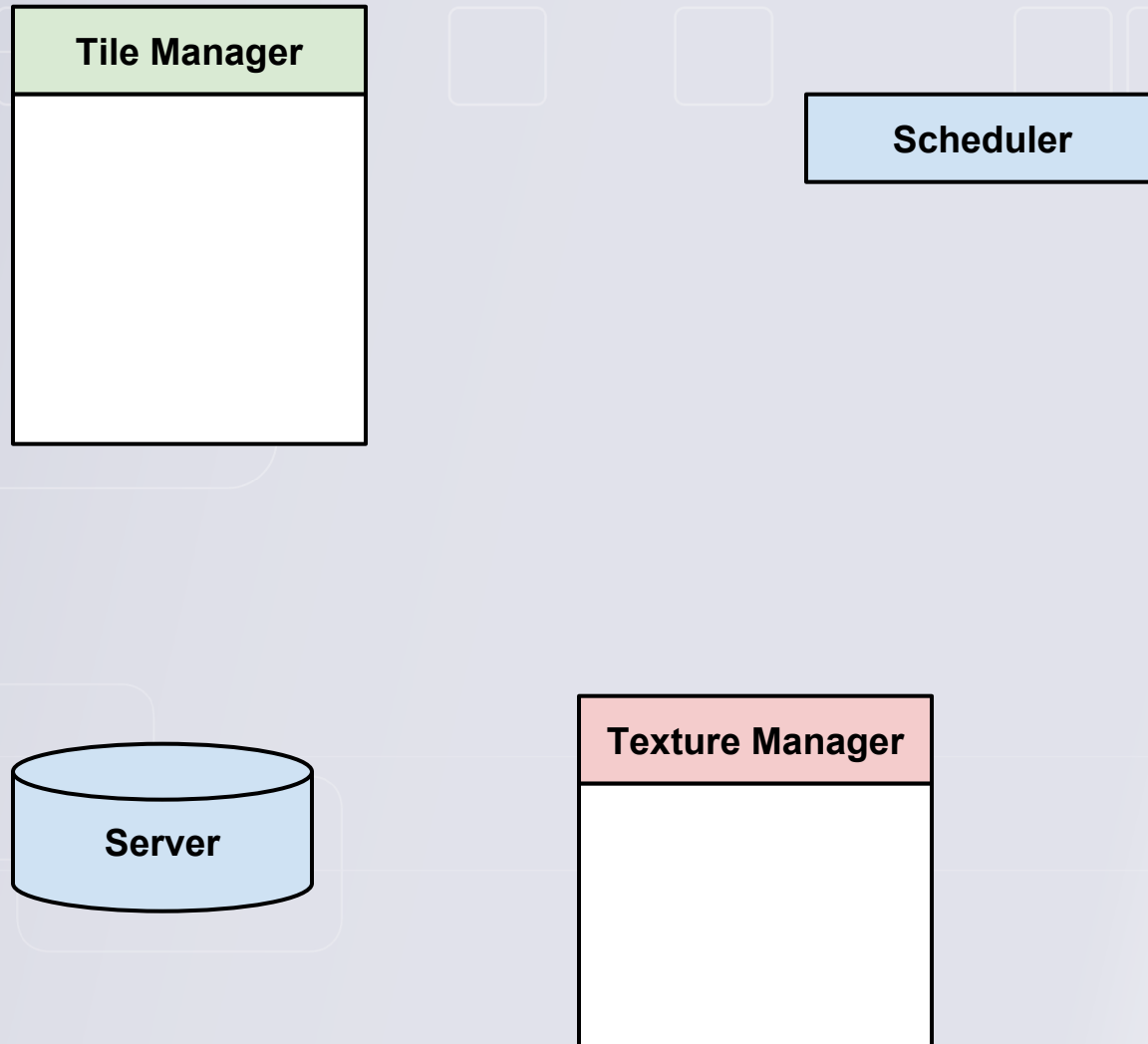
Data preparation

3DUSE: LIRIS' software to read and process CityGML



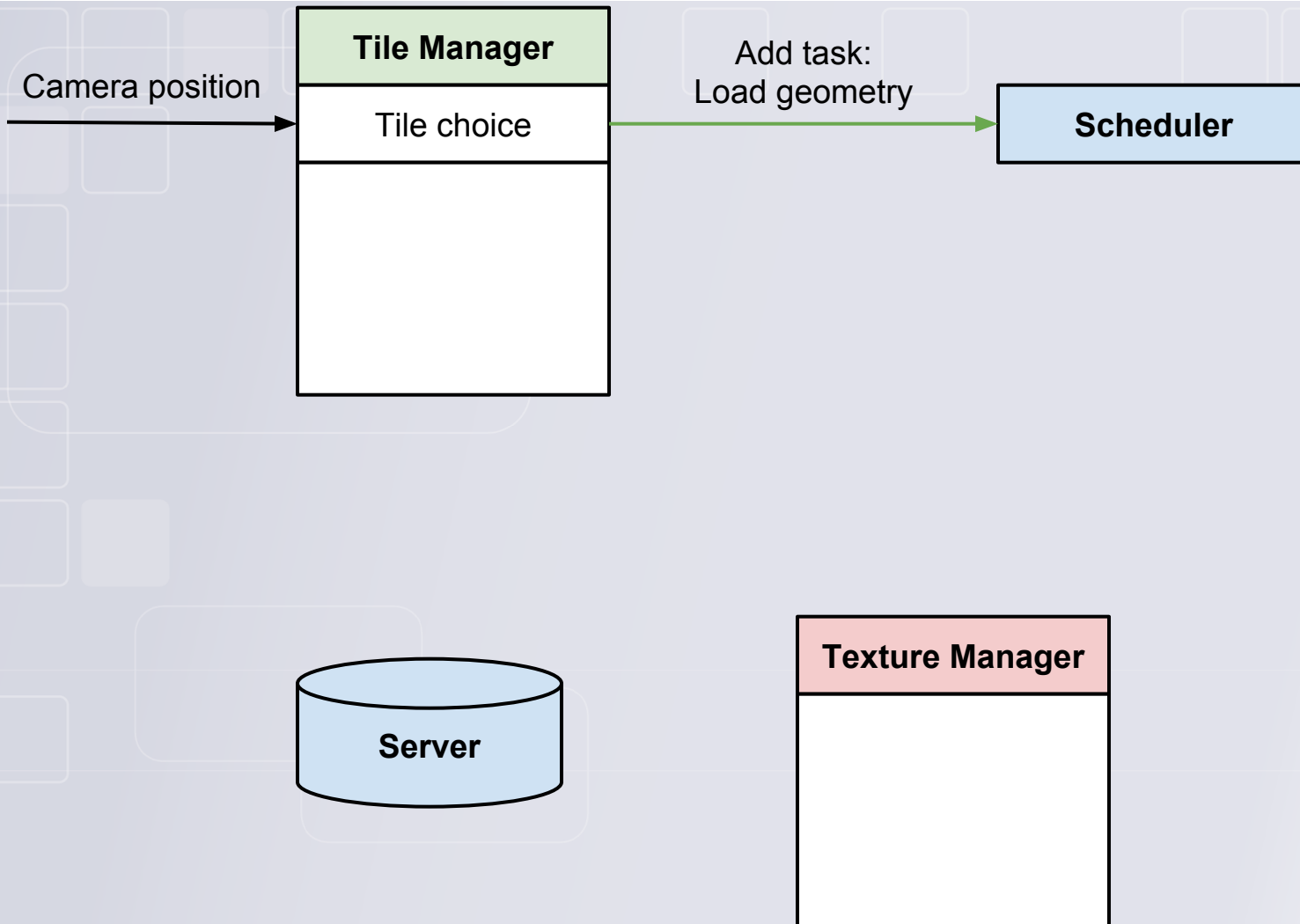
Urban data viewer framework

Detailed client architecture



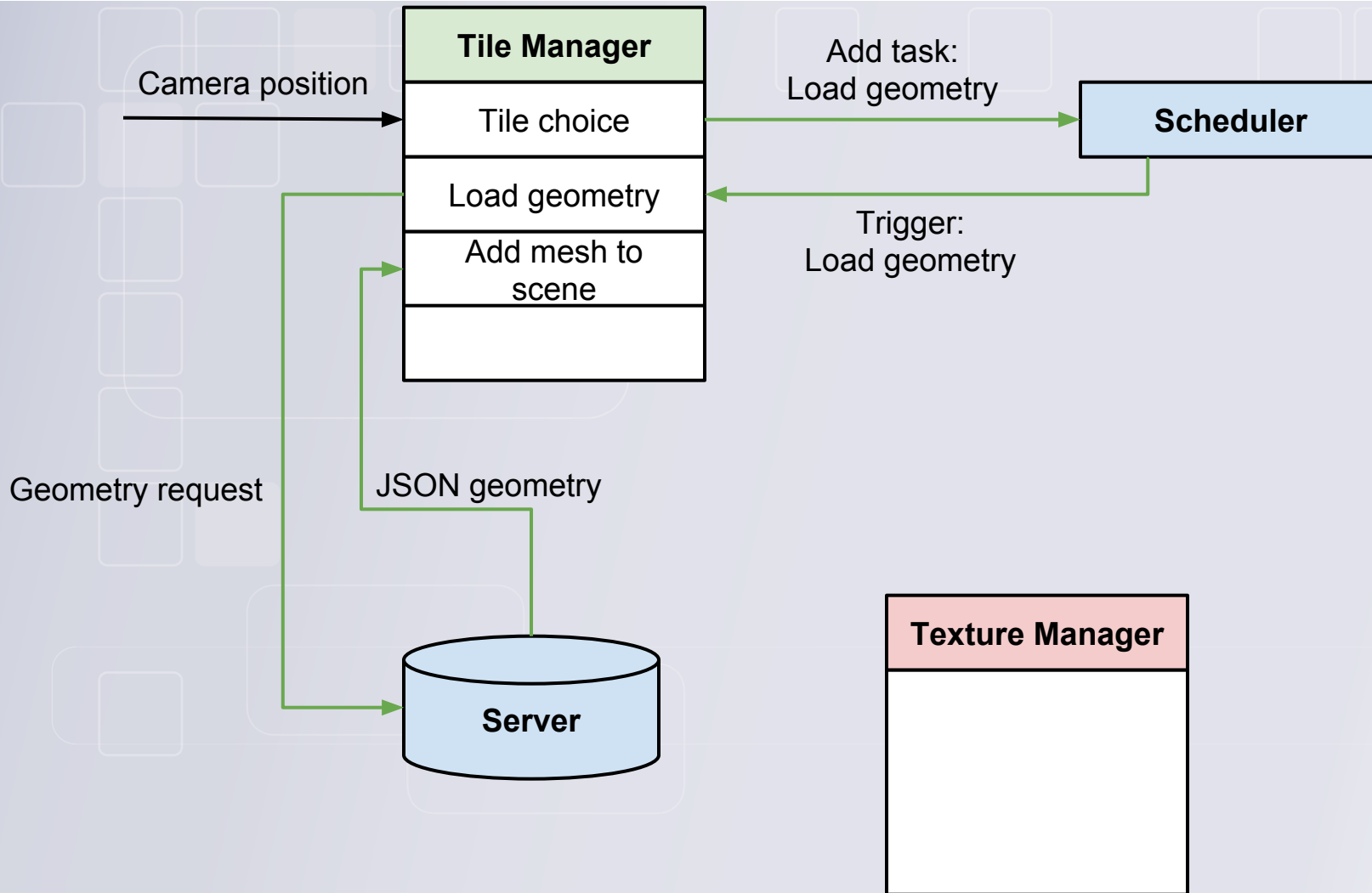
Urban data viewer framework

Detailed client architecture



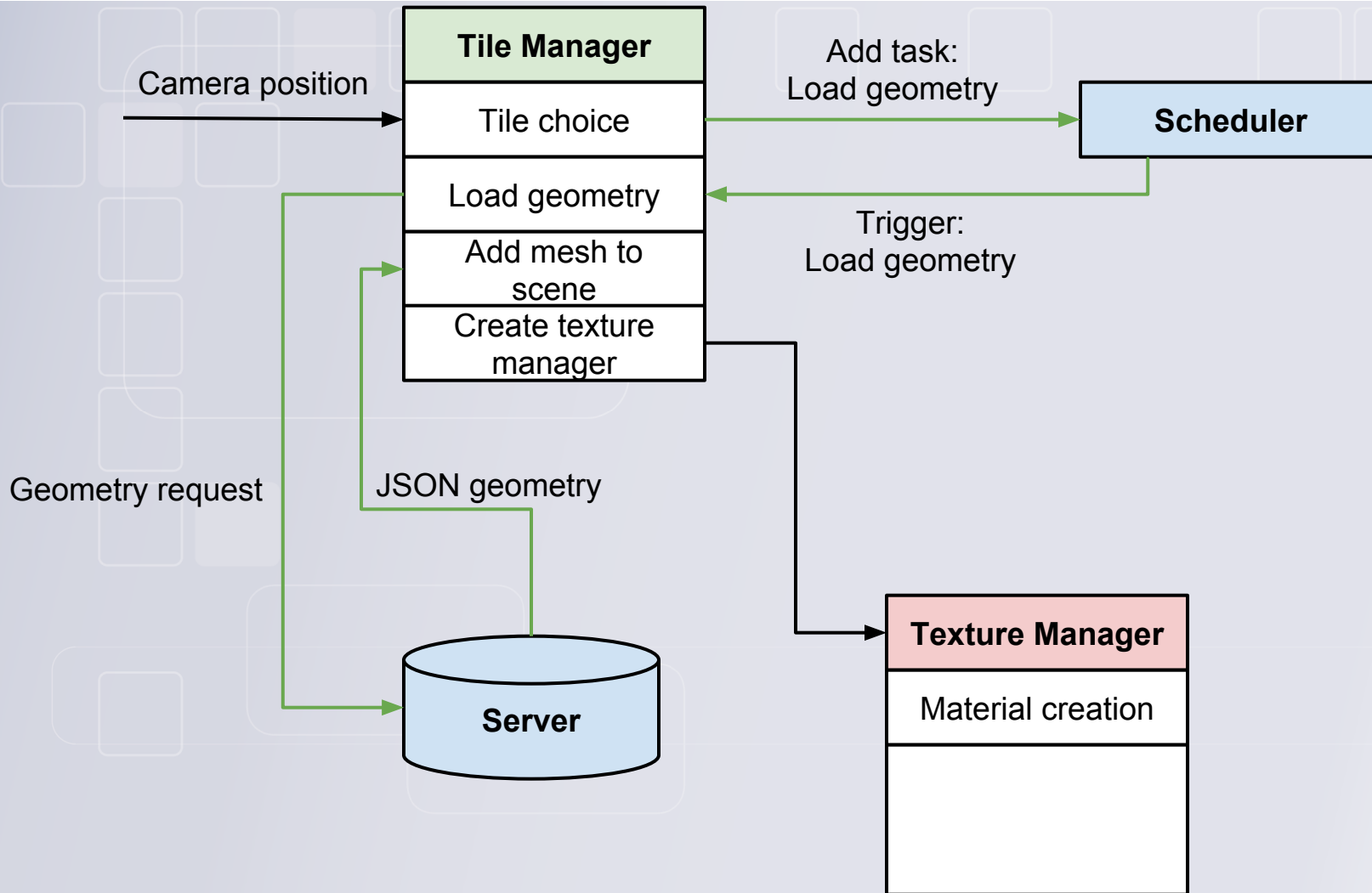
Urban data viewer framework

Detailed client architecture



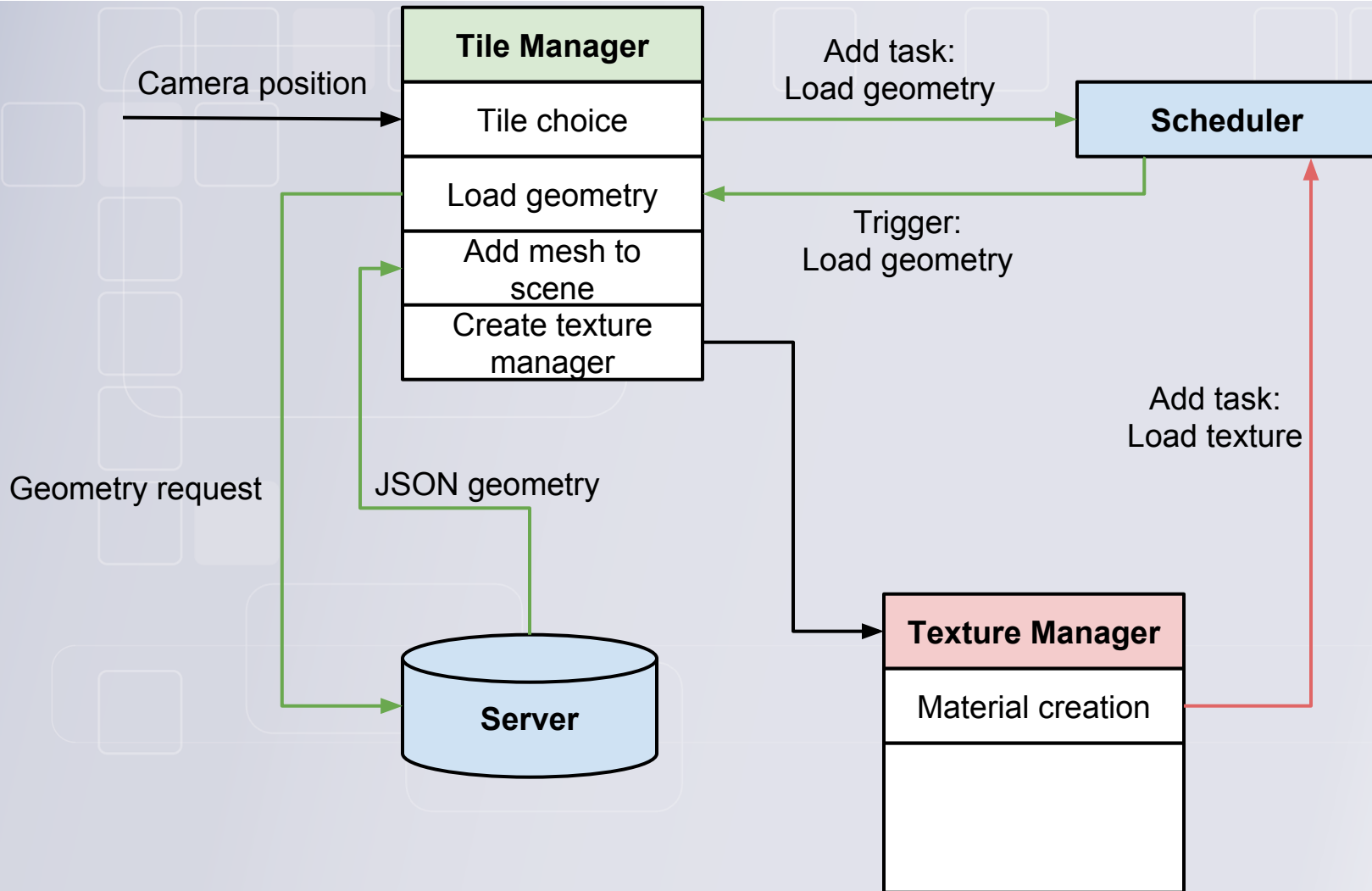
Urban data viewer framework

Detailed client architecture



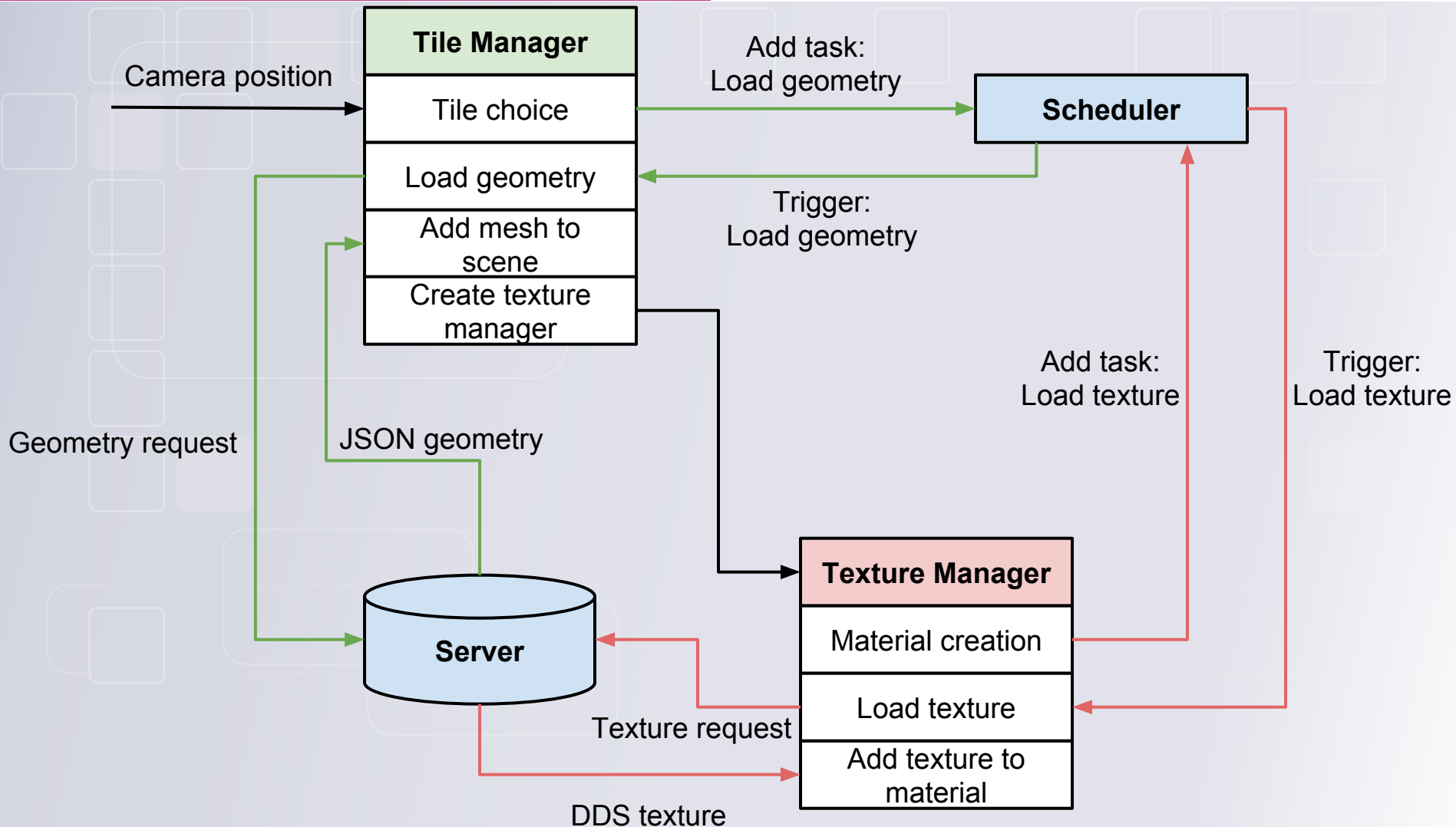
Urban data viewer framework

Detailed client architecture



Urban data viewer framework

Detailed client architecture



Urban data viewer framework

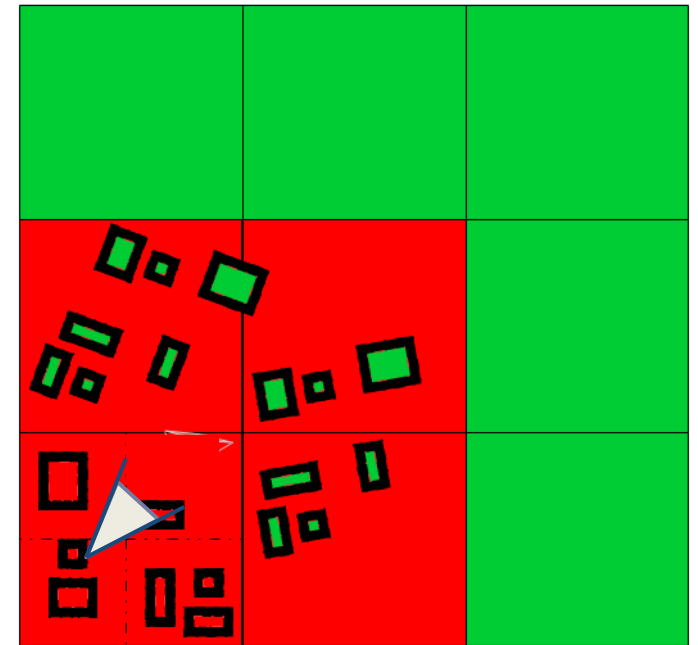
Scheduler

Three priority types:

- Top priority: data unloading
- High priority: depends on strategy
- Low priority: depends on strategy

Strategy decision according to:

- Task nature
- Layer type
- Distance to tile
- etc.



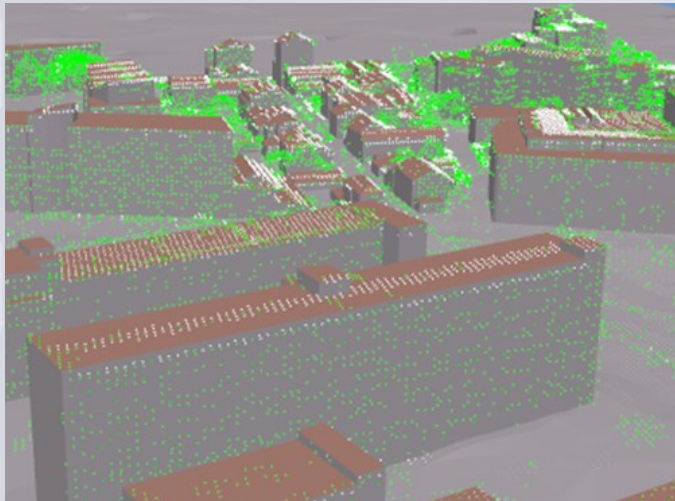
Legend

- Tile field loaded with high priority
- Tile field loaded with low priority
- Buildings loaded with high priority
- Buildings loaded with low priority

Urban data viewer framework

Additional data

- WFS streams with openlayers
- Point cloud (LiDAR)





Results

Demonstration

Results

Data size and loading time for a single tile:

	JSON Size	Compressed size	Geometry download	Parsing	Processing
DEM	2,2 Mb	0,341 Mb	113 ms	12 ms	57 ms
Buildings	2,5 Mb	0,401 Mb	84 ms	14 ms	32 ms
Remarkable buildings	1,2 Mb	0,187 Mb	47 ms	7 ms	15 ms
LiDAR	21 Mb	2,5 Mb	2520 ms	92 ms	17366 ms

	Texture download							
	Resolution 0		Resolution 1		Resolution 2		Resolution 3	
	Size	Download	Size	Download	Size	Download	Size	Download
DEM	0,68 Mb	76 ms	2,05 Mb	347 ms	8,2 Mb	1175 ms	32,8 Mb	4118 ms
Buildings	1,92 Mb	3280 ms	5,75 Mb	4998 ms	23 Mb	11808 ms	106 Mb	41090 ms
Remarkable buildings	1,21 Mb	2563 ms	3,63 Mb	3624 ms	14,5 Mb	8850 ms	64,8 Mb	29160 ms

Intel© i5 4590 @ 3.3GHz CPU, NVidia GTX970 GPU

Results

Data size and loading time for a single tile:

	JSON Size	Compressed size	Geometry download	Parsing	Processing
DEM	2,2 Mb	0,341 Mb	113 ms	12 ms	57 ms
Buildings	2,5 Mb	0,401 Mb	84 ms	14 ms	32 ms
Remarkable buildings	1,2 Mb	0,187 Mb	47 ms	7 ms	15 ms
LiDAR	21 Mb	2,5 Mb	2520 ms	92 ms	17366 ms

	Texture download							
	Resolution 0		Resolution 1		Resolution 2		Resolution 3	
	Size	Download	Size	Download	Size	Download	Size	Download
DEM	0,68 Mb	76 ms	2,05 Mb	347 ms	8,2 Mb	1175 ms	32,8 Mb	4118 ms
Buildings	1,92 Mb	3280 ms	5,75 Mb	4998 ms	23 Mb	11808 ms	106 Mb	41090 ms
Remarkable buildings	1,21 Mb	2563 ms	3,63 Mb	3624 ms	14,5 Mb	8850 ms	64,8 Mb	29160 ms

Intel© i5 4590 @ 3.3GHz CPU, NVidia GTX970 GPU

Low parsing time

Results

Data size and loading time for a single tile:

	JSON Size	Compressed size	Geometry download	Parsing	Processing
DEM	2,2 Mb	0,341 Mb	113 ms	12 ms	57 ms
Buildings	2,5 Mb	0,401 Mb	84 ms	14 ms	32 ms
Remarkable buildings	1,2 Mb	0,187 Mb	47 ms	7 ms	15 ms
LiDAR	21 Mb	2,5 Mb	2520 ms	92 ms	17366 ms

	Texture download							
	Resolution 0		Resolution 1		Resolution 2		Resolution 3	
	Size	Download	Size	Download	Size	Download	Size	Download
DEM	0,68 Mb	76 ms	2,05 Mb	347 ms	8,2 Mb	1175 ms	32,8 Mb	4118 ms
Buildings	1,92 Mb	3280 ms	5,75 Mb	4998 ms	23 Mb	11808 ms	106 Mb	41090 ms
Remarkable buildings	1,21 Mb	2563 ms	3,63 Mb	3624 ms	14,5 Mb	8850 ms	64,8 Mb	29160 ms

Intel© i5 4590 @ 3.3GHz CPU, NVidia GTX970 GPU

High time disparity

Conclusion and future works

Introduced a framework allowing:

- Efficient visualisation of data from different sources using standards
- GPU-friendly progressive textures
- Data loading prioritisation

Future works (joint works with Oslandia):

- Investigate other texture compression formats
- Webworker integration
- Individual building picking



Urban Data Visualisation in a web browser

Jérémy Gaillard, Alexandre Vienne, Rémi Baume, Frédéric Pedrinis, Adrien Peytavie, Gilles Gesquière

Laboratoire d'InfoRmatique en Image et Systèmes d'information

jeremy.gaillard@liris.cnrs.fr



INSA



UNIVERSITÉ
LUMIÈRE
LYON 2

