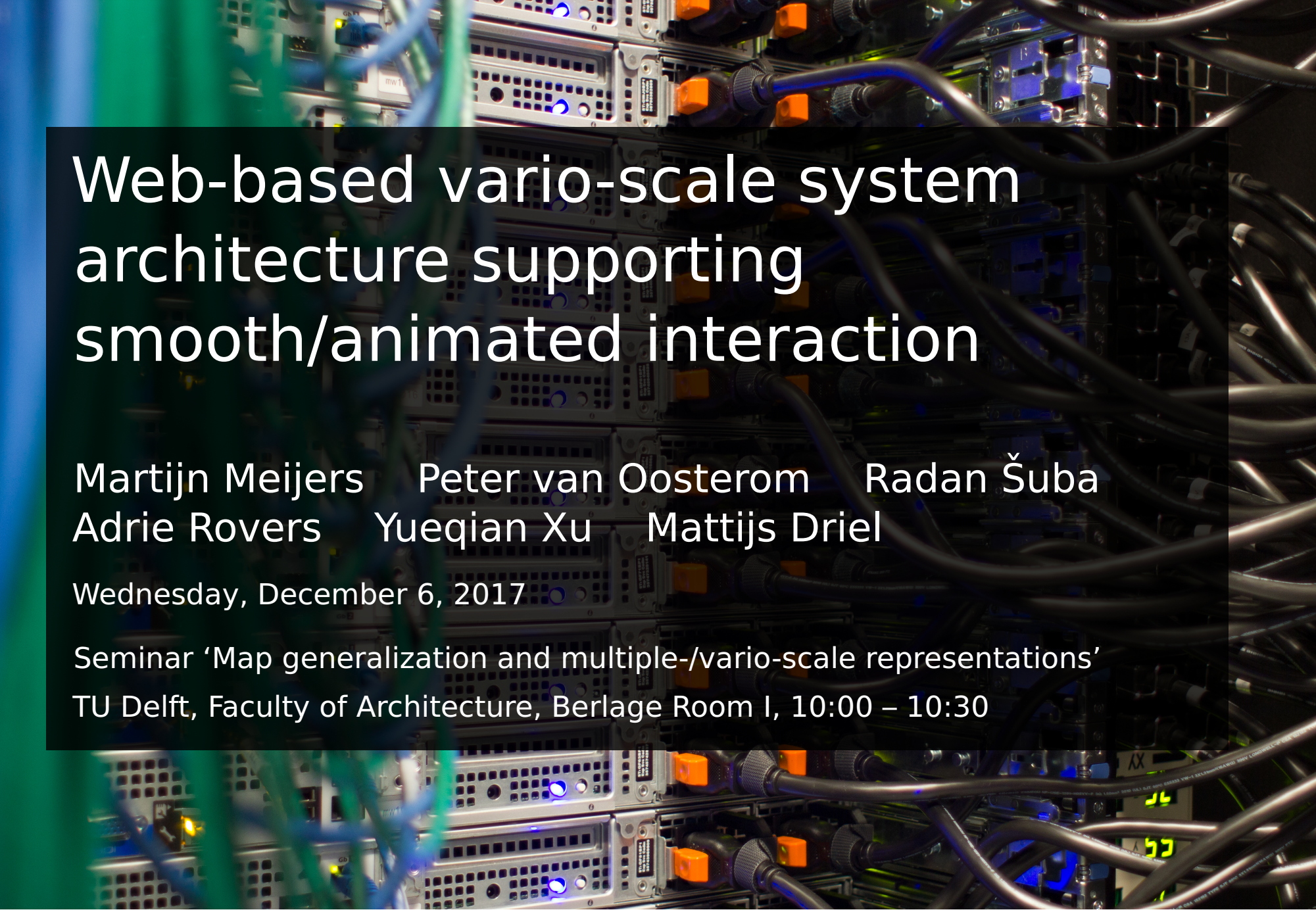


Web-based vario-scale system architecture supporting smooth/animated interaction

Martijn Meijers Peter van Oosterom Radan Šuba
Adrie Rovers Yueqian Xu Mattijs Driel

Wednesday, December 6, 2017

Seminar ‘Map generalization and multiple-/vario-scale representations’

TU Delft, Faculty of Architecture, Berlage Room I, 10:00 – 10:30

TU Delft

Overview

Web-based vario-scale system architecture
supporting **smooth/animated interaction**

Overview

Web-based vario-scale system architecture supporting **smooth/animated interaction**

1. Vario-scale content

Overview

Web-based vario-scale system architecture supporting **smooth/animated interaction**

1. Vario-scale content
2. Web-based system architecture

Overview

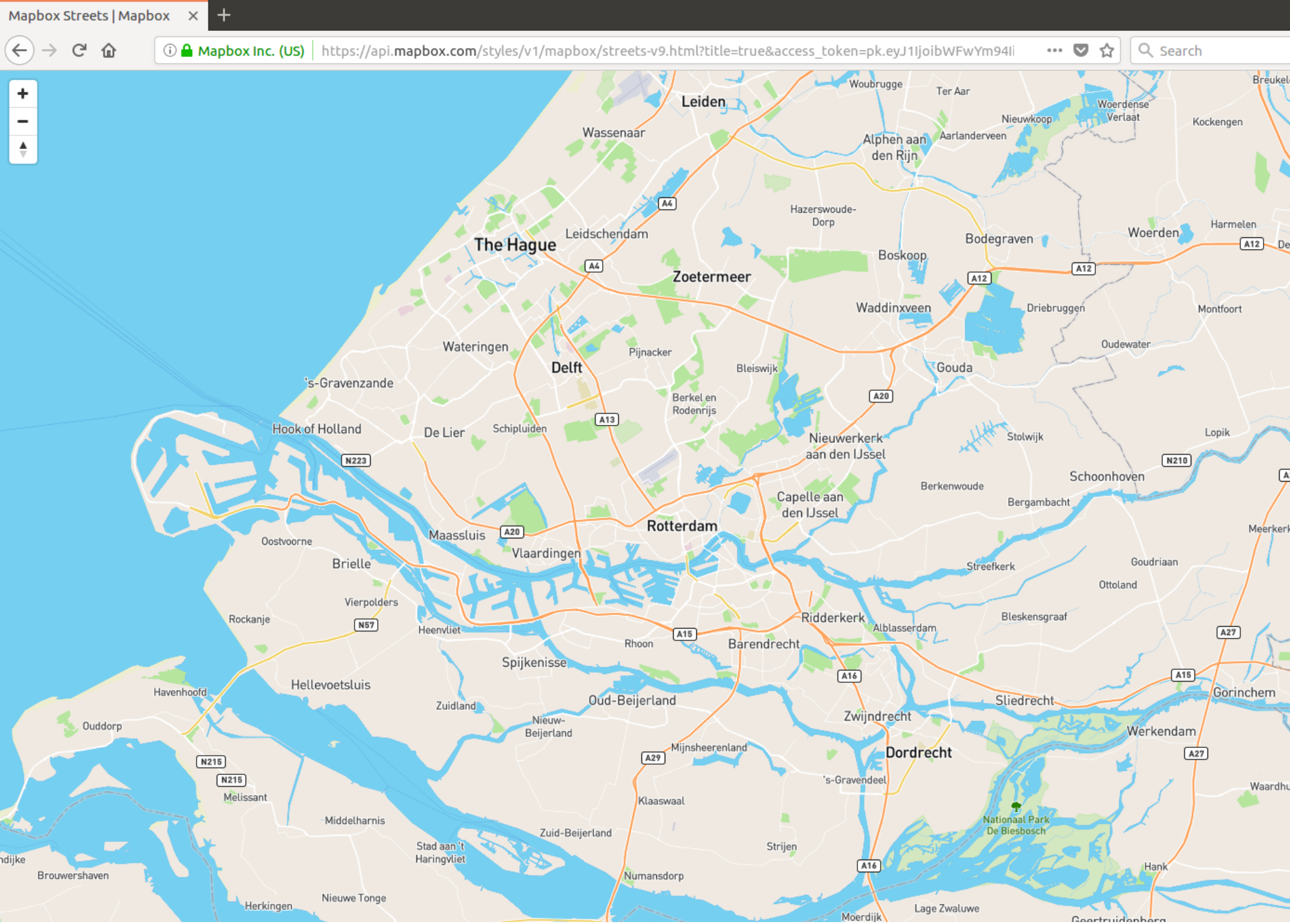
Web-based vario-scale system architecture supporting **smooth/animated interaction**

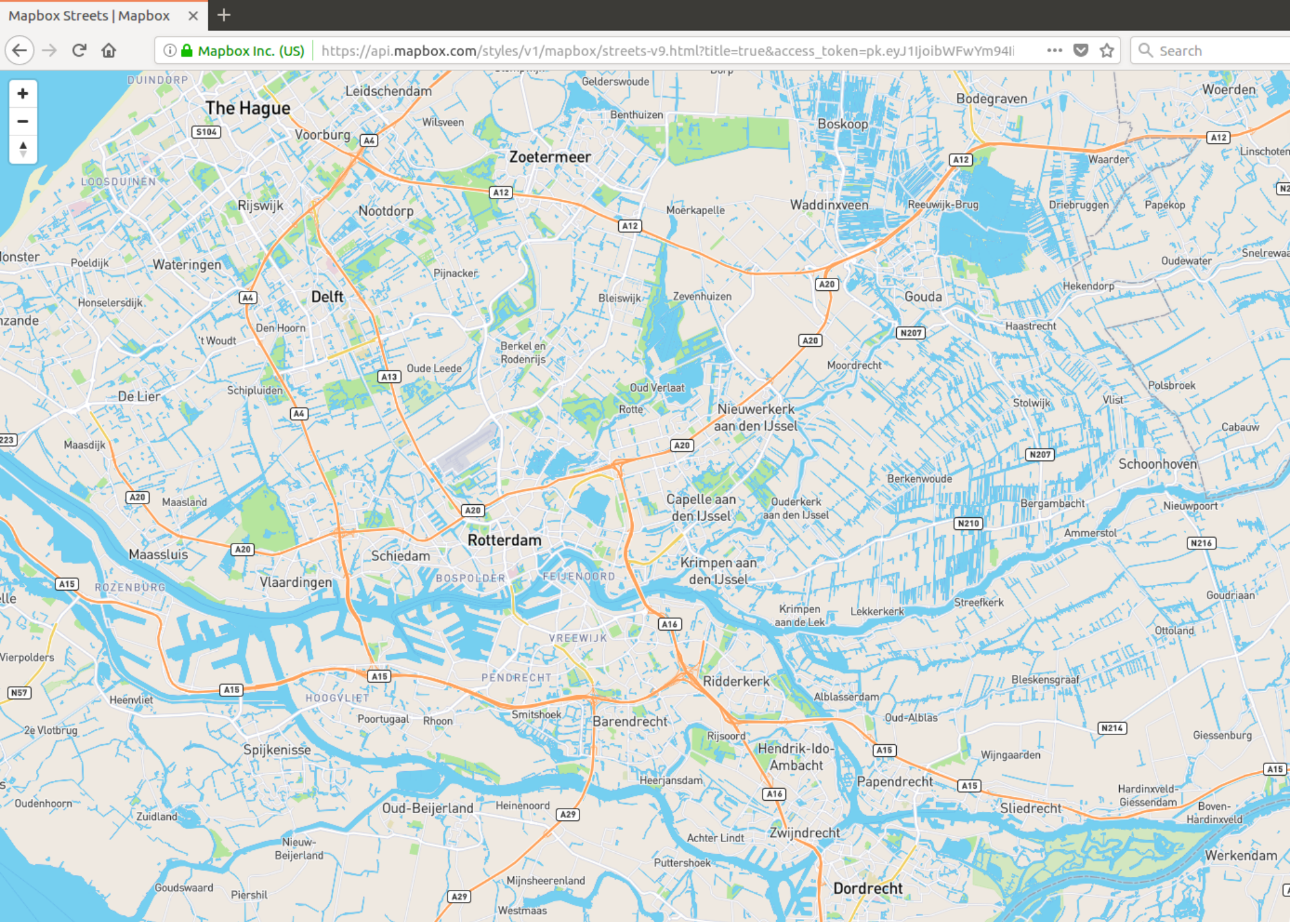
1. Vario-scale content
2. Web-based system architecture
3. Smooth/animated interaction

Vario-scale maps: A definition

Vario-scale maps: A definition

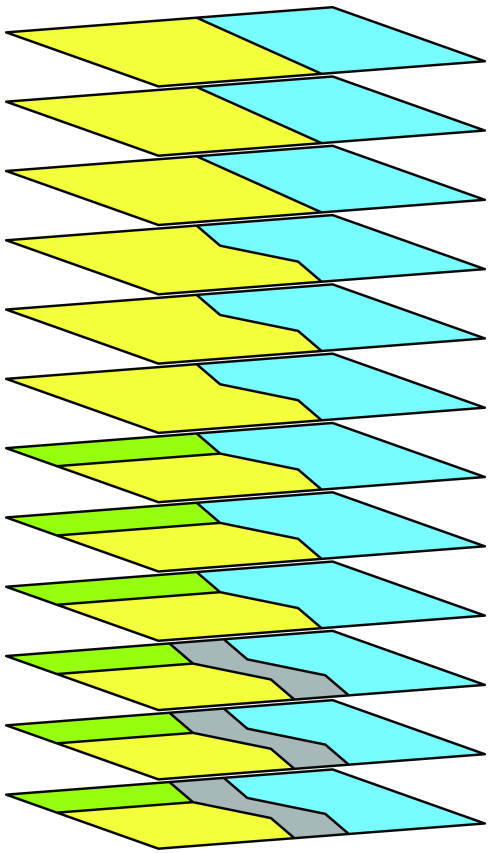
A small delta in map scale,
leads to a small delta in map content


Mapbox Streets | Mapbox
Mapbox Inc. (US)
https://api.mapbox.com/styles/v1/mapbox/streets-v9.html?title=true&access_token=pk.eyJ1IjoibWFwYm94li
Search
Leiden
Woubrugge
Ter Aar
Wassenaar
Nieuwkoop
Woerdense Verlaat
Kockengen
Alphen aan den Rijn
Aarlanderveen
Hazerswoude-Dorp
Leidschendam
The Hague
Bodegraven
Woerden
Harmelen
Boskoop
Zoetermeer
Waddinxveen
Driebruggen
Montfoort
Oudewater
Wateringen
Pijnacker
Delft
Bleiswijk
Gouda
's-Gravenzande
Berkel en Rodenrijs
Hook of Holland
De Lier
Schipluiden
Nieuwerkerk aan den IJssel
Stolwijk
Lopik
Schoonhoven
Berkenwoude
Bergambacht
Capelle aan den IJssel
Rotterdam
Maassluis
Vlaardingen
Oostvoorne
Brielle
Streefkerk
Goudriaan
Ottoland
Vierpolders
Rockanje
Heenvliet
Ridderkerk
Alblasserdam
Bleskensgraaf
Rhoon
Barendrecht
Spijkenisse
Hellevoetsluis
Havenhoofd
Oud-Beijerland
Sliedrecht
Gorinchem
Zuidland
Zwijndrecht
Ouddorp
Nieuw-Beijerland
Werkendam
Mijnsheerenland
Dordrecht
's-Gravendeel
Melissant
Klaaswaal
Middelharnis
Zuid-Beijerland
Nationaal Park De Biesbosch
Stad aan 't Haringvliet
Strijen
Brouwershaven
Numansdorp
Hank
Herkingen
Nieuwe Tonge
Moerdijk
Lage Zwaluwe


Mapbox Streets | Mapbox
Mapbox Inc. (US)
https://api.mapbox.com/styles/v1/mapbox/streets-v9.html?title=true&access_token=pk.eyJ1IjoibWFwYm94li
Search
The Hague
Leidschendam
Voorburg
Wilsveen
Gelderswoude
Benthuizen
Zoetermeer
Boskoop
Bodegraven
Woerden
Waarder
Linschoten
Rijswijk
Nootdorp
Moerkapelle
Waddinxveen
Reeuwijk-Brug
Driebruggen
Papekop
Wateringen
Poeldijk
Pijnacker
Delft
Bleiswijk
Zevenhuizen
Gouda
Oudewater
Hekendorp
Honselersdijk
Den Hoorn
't Woudt
Berkel en Rodenrijs
Haastrecht
Oude Leede
Moordrecht
De Lier
Schipluiden
Oud Verlaat
Rotte
Nieuwerkerk aan den IJssel
Stolwijk
Vlist
Polsbroek
Cabauw
Maasdijk
Schoonhoven
Berkenwoude
Maasland
Capelle aan den IJssel
Ouderkerk aan den IJssel
Bergambacht
Nieuwpoort
Ammerstol
Rotterdam
Maassluis
Schiedam
Krimpen aan den IJssel
Vlaardingen
BOSPOLDER
FEIJENOORD
ROZENBURG
Krimpen aan de Lek
Lekkerkerk
Streefkerk
Goudriaan
Ottoland
VREEWIJK
Vierpolders
PENDRECHT
Ridderkerk
Bleskensgraaf
Alblasserdam
HOOGVLIET
Heenvliet
Poortugaal
Rhoon
Smitshoek
Barendrecht
Oud-Alblas
2e Vlotbrug
Rijsoord
Giessenburg
Spijkenisse
Hendrik-Ido-Ambacht
Wijngaarden
Heerjansdam
Papendrecht
Oudenhoorn
Zuidland
Oud-Beijerland
Heinenoord
Sliedrecht
Hardinxveld-Giessendam
Boven-Hardinxveld
Achter Lindt
Zwijndrecht
Nieuw-Beijerland
Puttershoek
Werkendam
Goudswaard
Piershil
Mijnsheerenland
Dordrecht
Westmaas

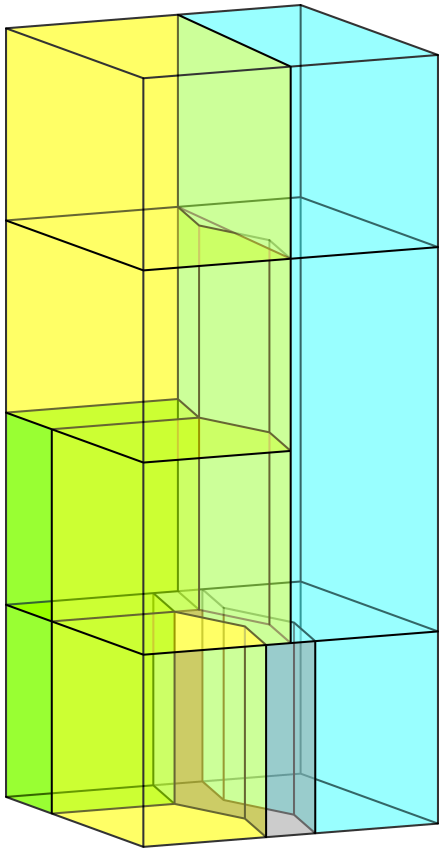
Vario-scale: Delta scale → Delta map content

- Changing in small steps key to smooth map content!
- Changes for transitions as small as possible?



Vario-scale: Delta scale → Delta map content

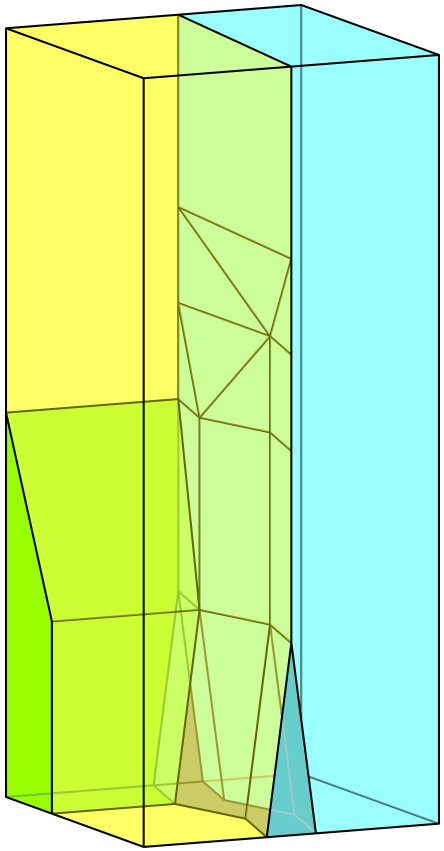
- Changing in small steps key to smooth map content!
- Changes for transitions as small as possible?



Space Scale Cube (SSC)

Vario-scale: Delta scale → Delta map content

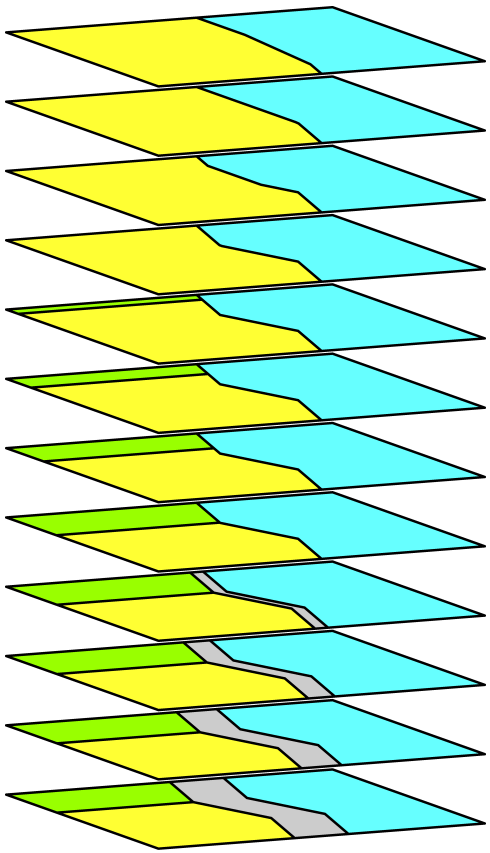
- Changing in small steps key to smooth map content!
- Changes for transitions as small as possible!!



Smooth Space Scale Cube

Vario-scale: Delta scale → Delta map content

- Changing in small steps key to smooth map content!
- Changes for transitions as small as possible!!



Vario-scale maps: Generate Content

Automated generalization
Making the map simpler with elementary operations

Vario-scale maps: Generate Content

Automated generalization
Making the map simpler with elementary operations

1. Merge areas: 2 areas become 1

Vario-scale maps: Generate Content

Automated generalization
Making the map simpler with elementary operations

1. Merge areas: 2 areas become 1
2. Split: 1 area split over its neighbours

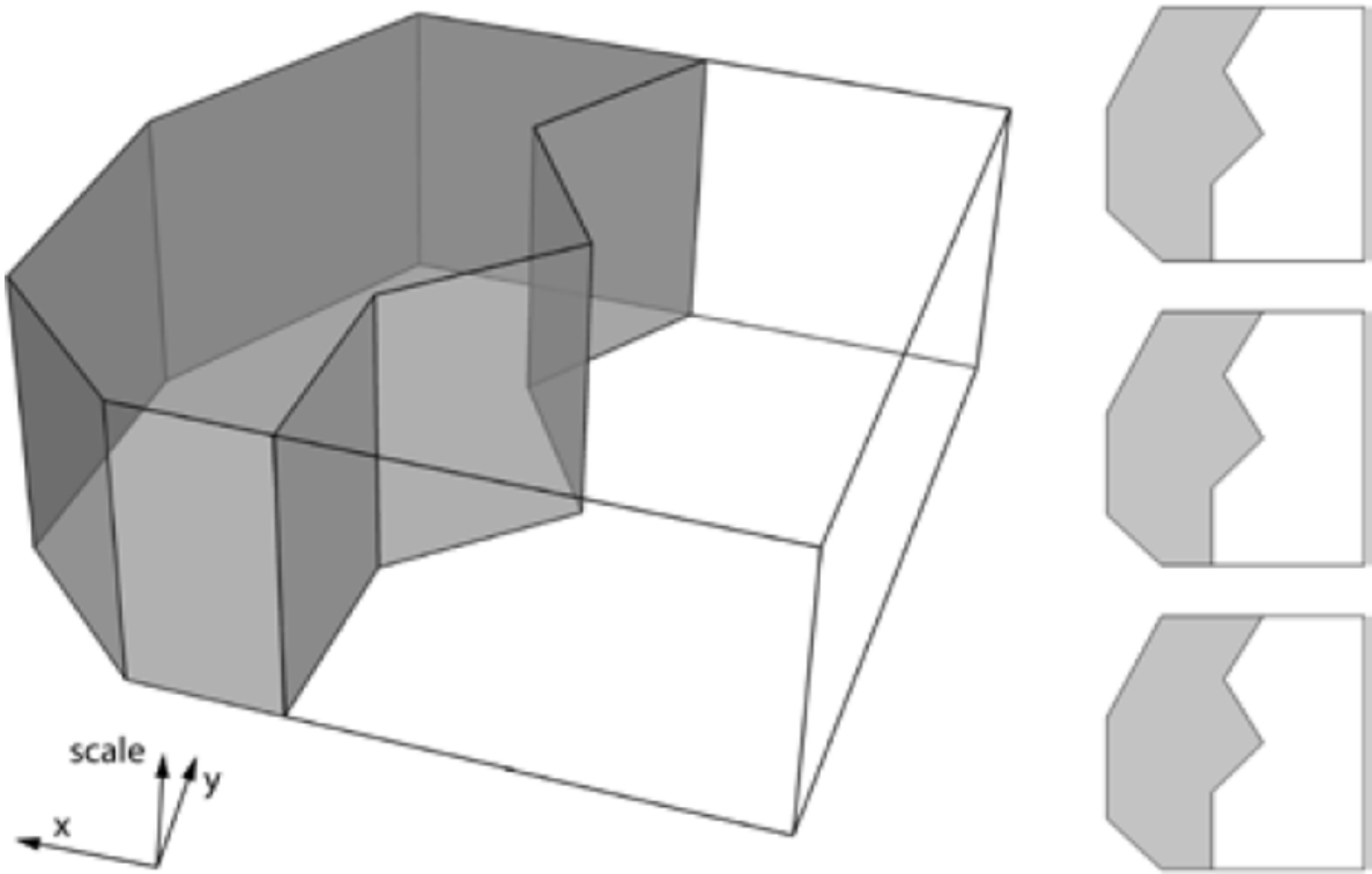
Vario-scale maps: Generate Content

Automated generalization
Making the map simpler with elementary operations

1. Merge areas: 2 areas become 1
2. Split: 1 area split over its neighbours
3. Simplify: simplify border of area object / line object

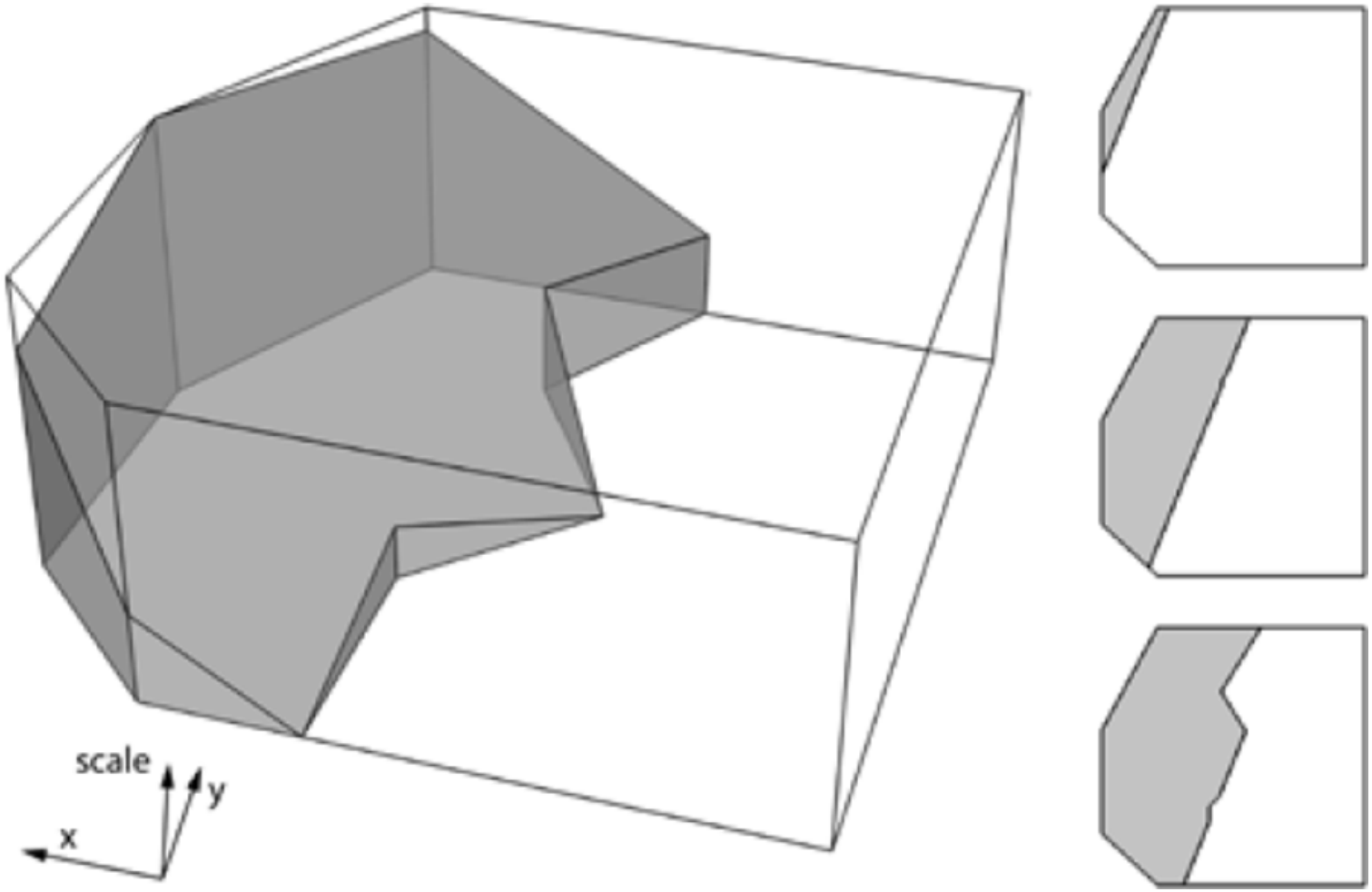
Vario-scale maps: Generate Content

Transition for merging 2 area objects



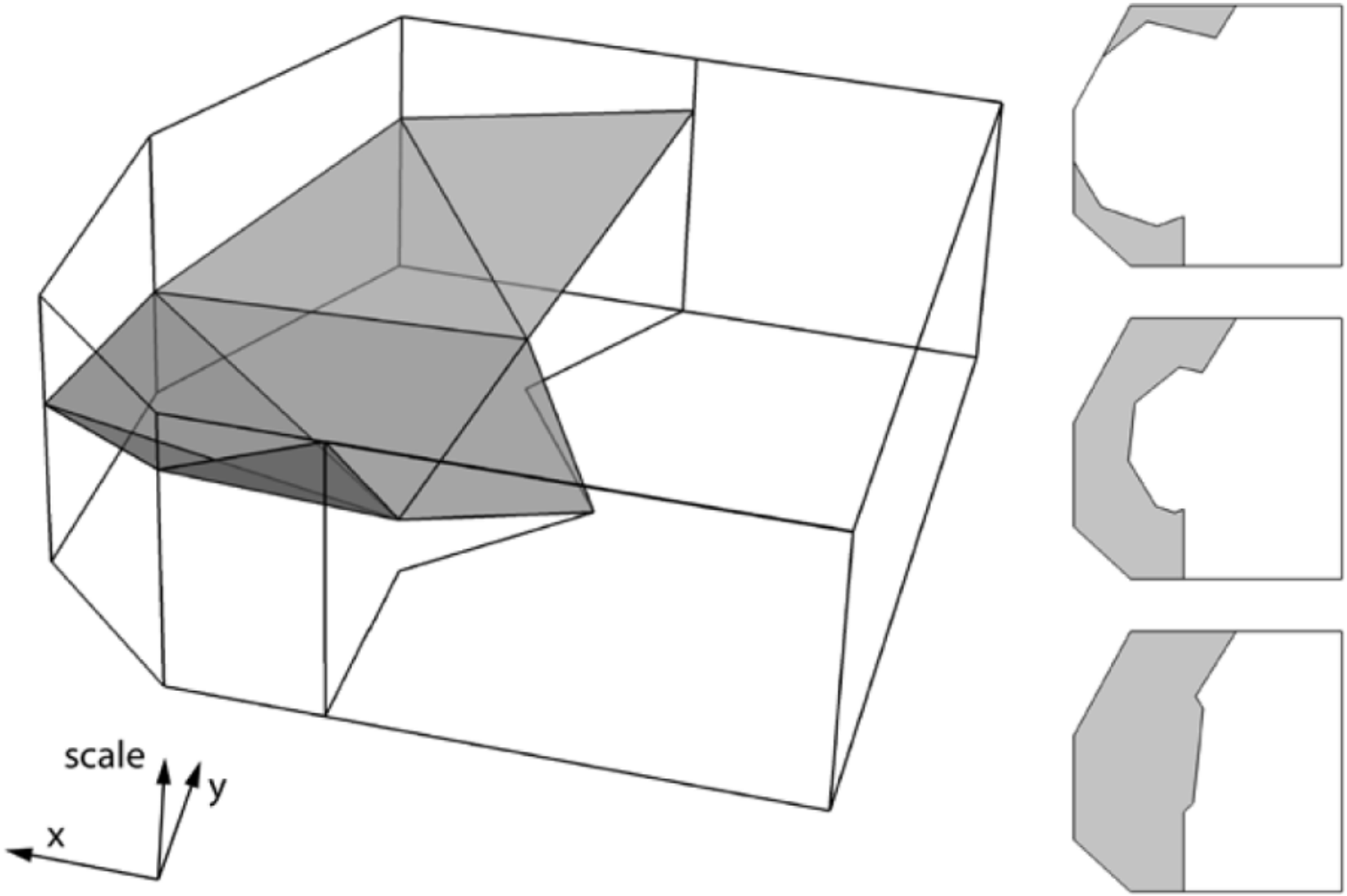
Vario-scale maps: Generate Content

Transition for merging 2 area objects



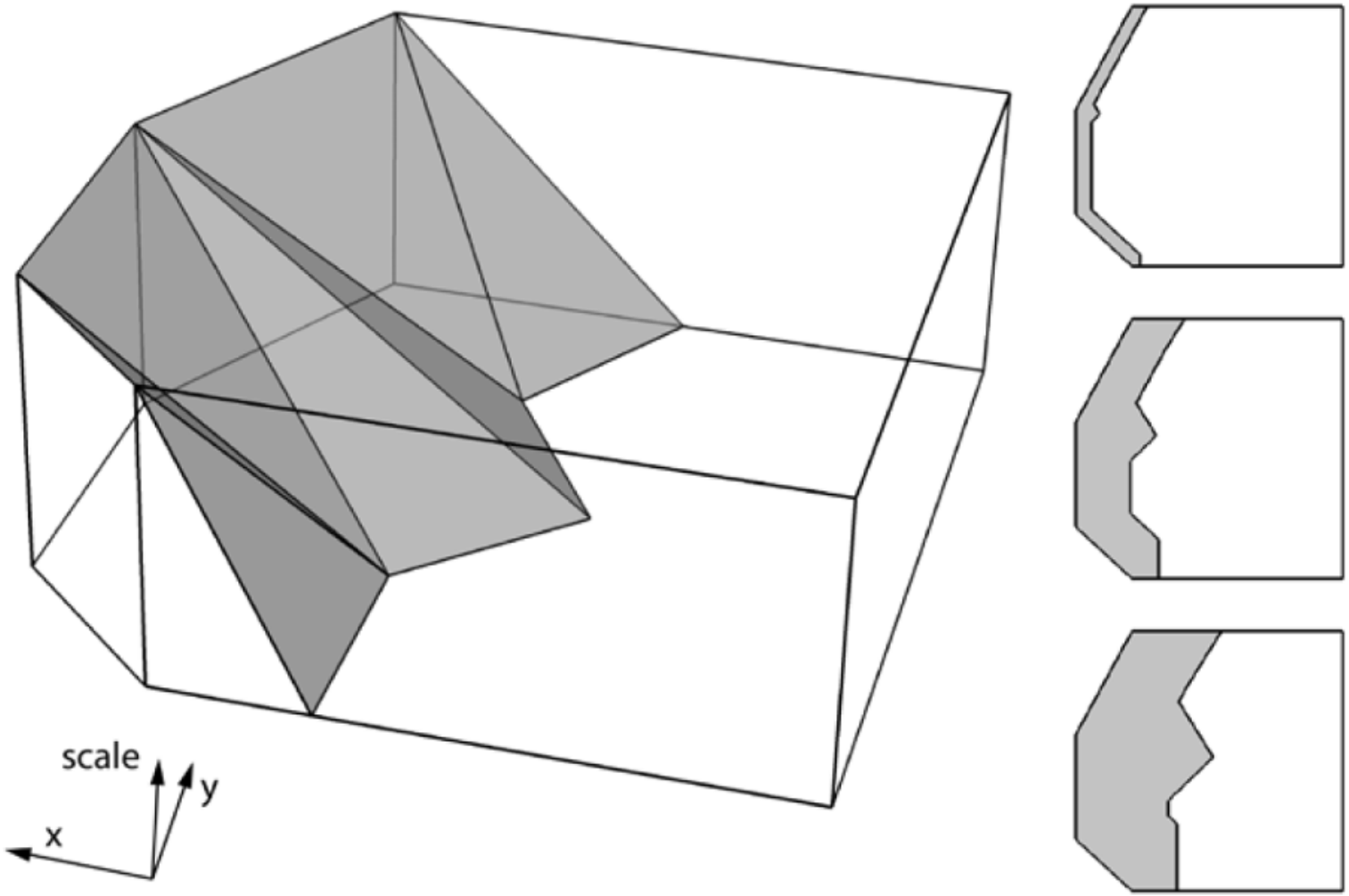
Vario-scale maps: Generate Content

Transition for merging 2 area objects



Vario-scale maps: Generate Content

Transition for merging 2 area objects

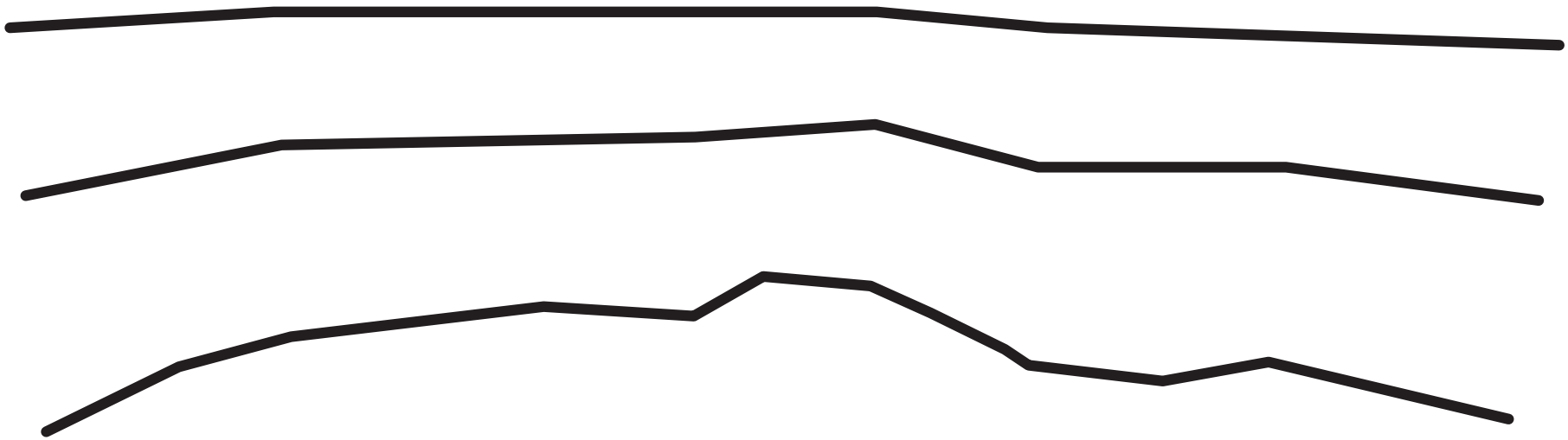


Vario-scale maps: Generate Content

Transition for simplifying line object

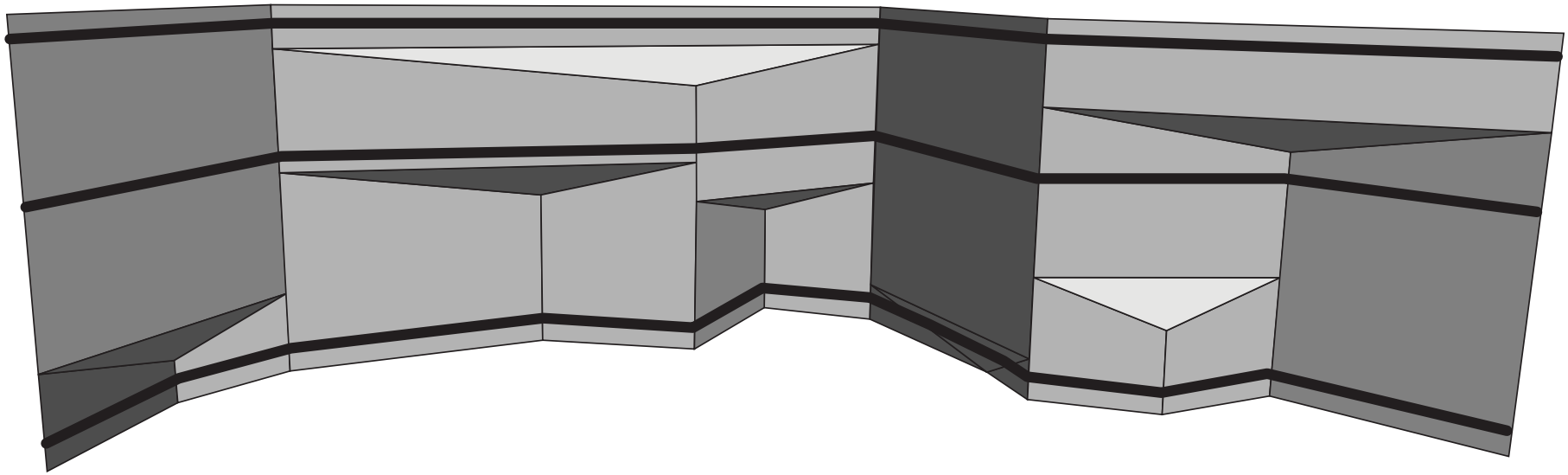
Vario-scale maps: Generate Content

Transition for simplifying line object



Vario-scale maps: Generate Content

Transition for simplifying line object



Web-based system architecture

High level overview

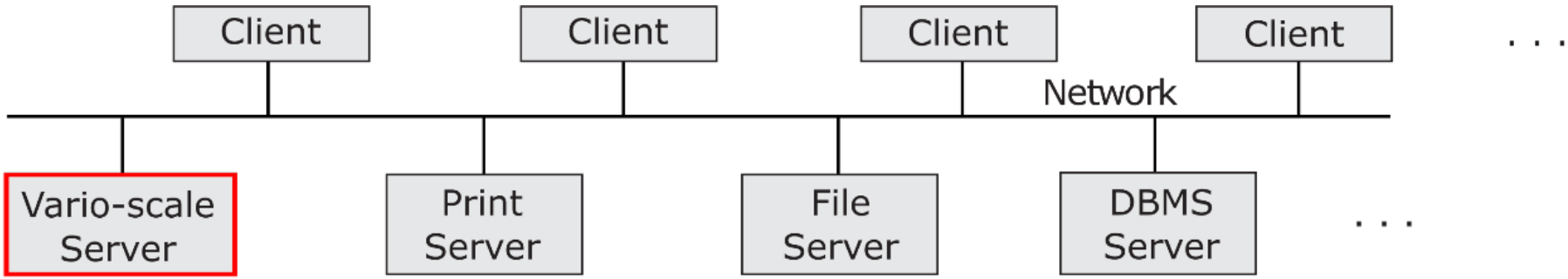


Figure adapted from Elmasri and Navathe (2010).

Web-based system architecture

High level overview

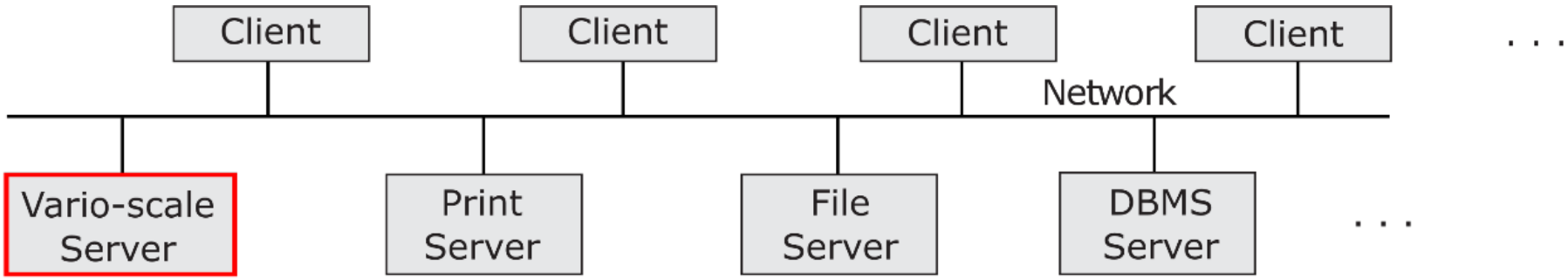
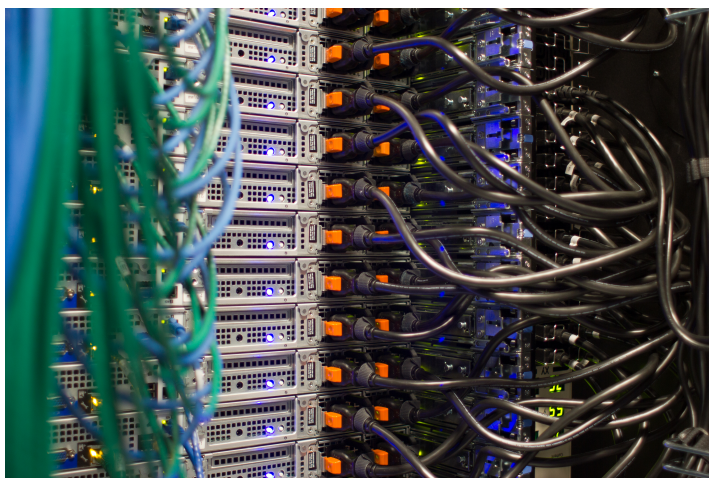


Figure adapted from Elmasri and Navathe (2010).



1. Server (data) – Client (interaction)

Web-based system architecture

High level overview

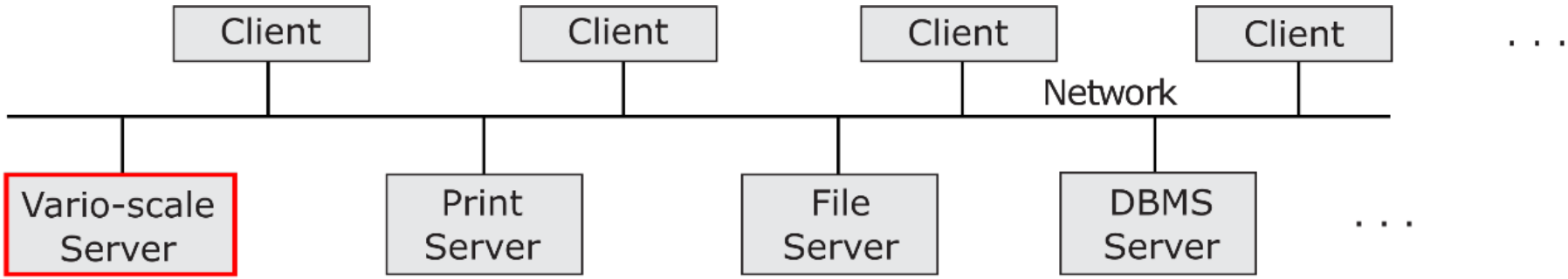


Figure adapted from Elmasri and Navathe (2010).

1. Server (data) – Client (interaction)
2. Server requirements: Scalable (many clients, no state to remember)

Web-based system architecture

High level overview

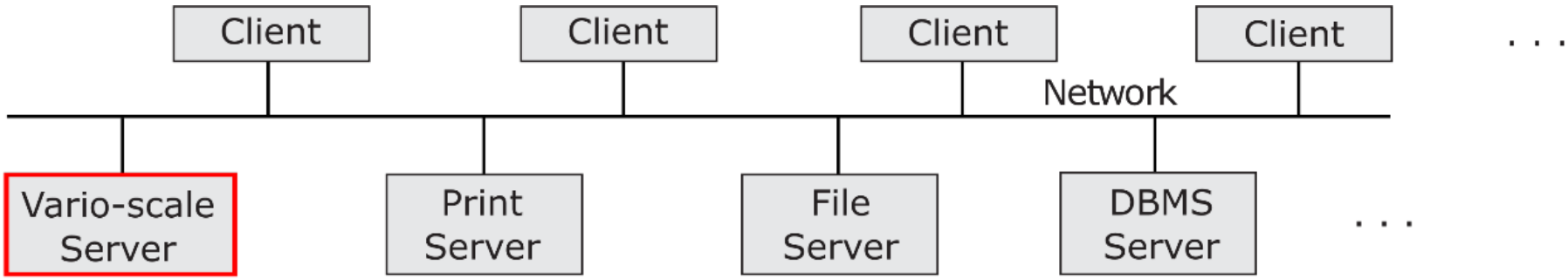


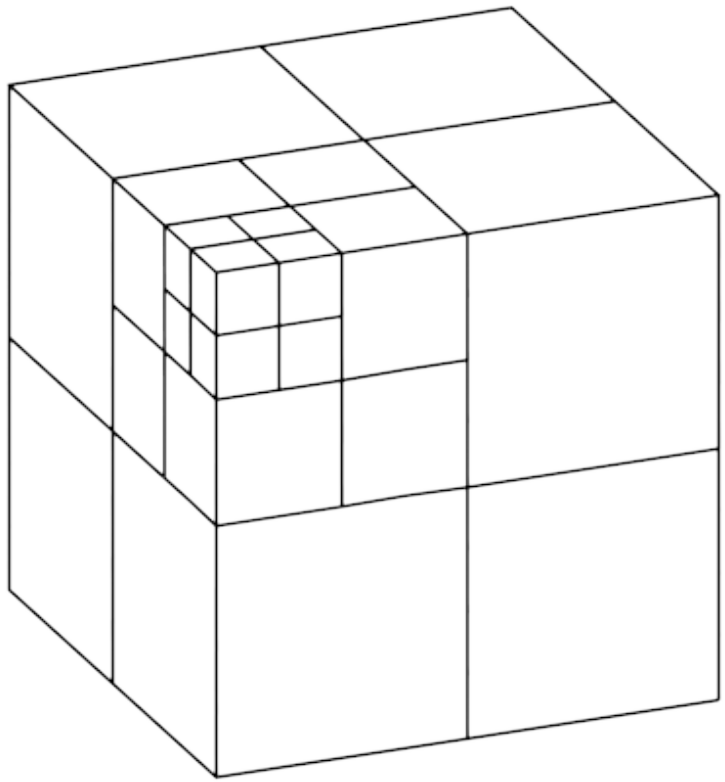
Figure adapted from Elmasri and Navathe (2010).

1. Server (data) – Client (interaction)
2. Server requirements: Scalable (many clients, no state to remember)
3. Client requirements: Fat client (powerful, enough memory, GPU). Responsible for: data retrieval, rendering

Web-based system architecture

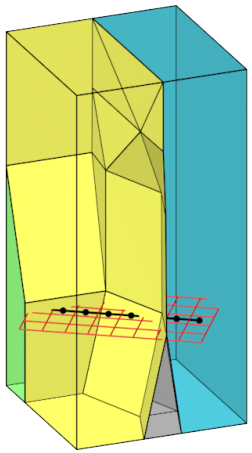
Server:

- Makes smooth data content available
- Data is too large to be retrieved in one go (e.g. map for all of Europe)
- Solution: Retrieve in parts, e.g. Octree subdivision



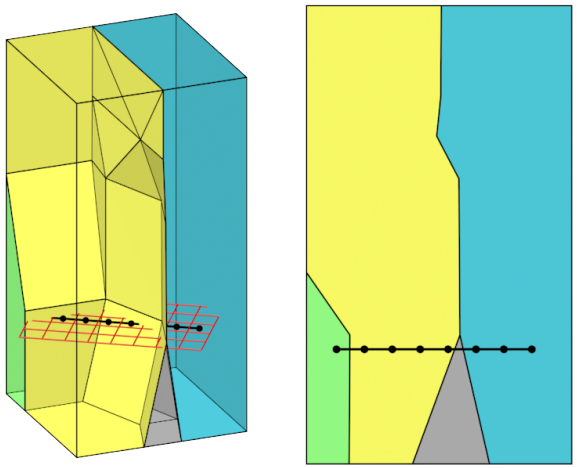
Web-based system architecture

Intermezzo: Principle of rendering at client side



Web-based system architecture

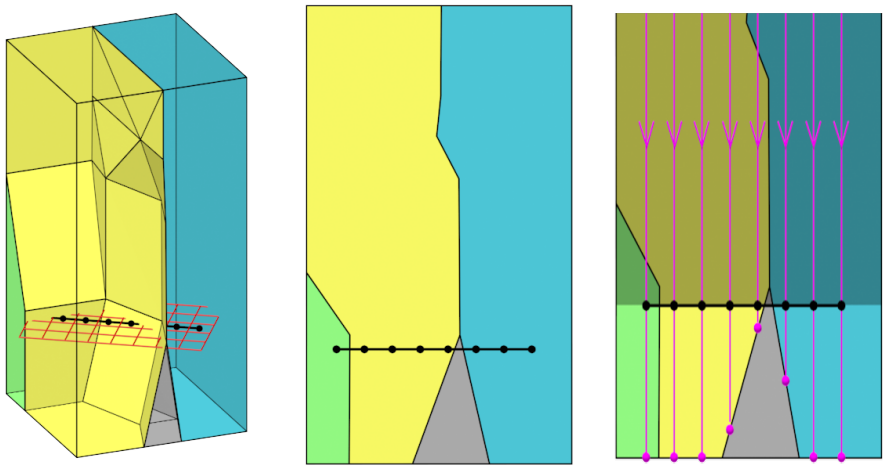
Intermezzo: Principle of rendering at client side



1. Determine slice plane (raster)

Web-based system architecture

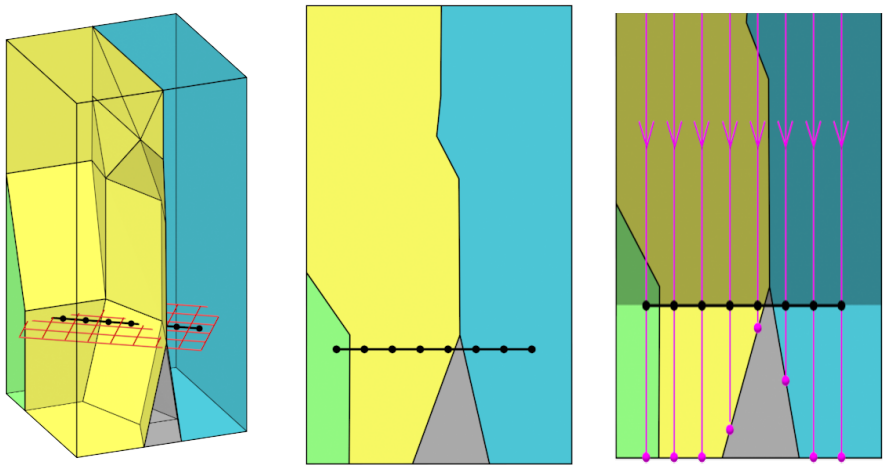
Intermezzo: Principle of rendering at client side



1. Determine slice plane (raster)
2. Discard part that is above slice plane

Web-based system architecture

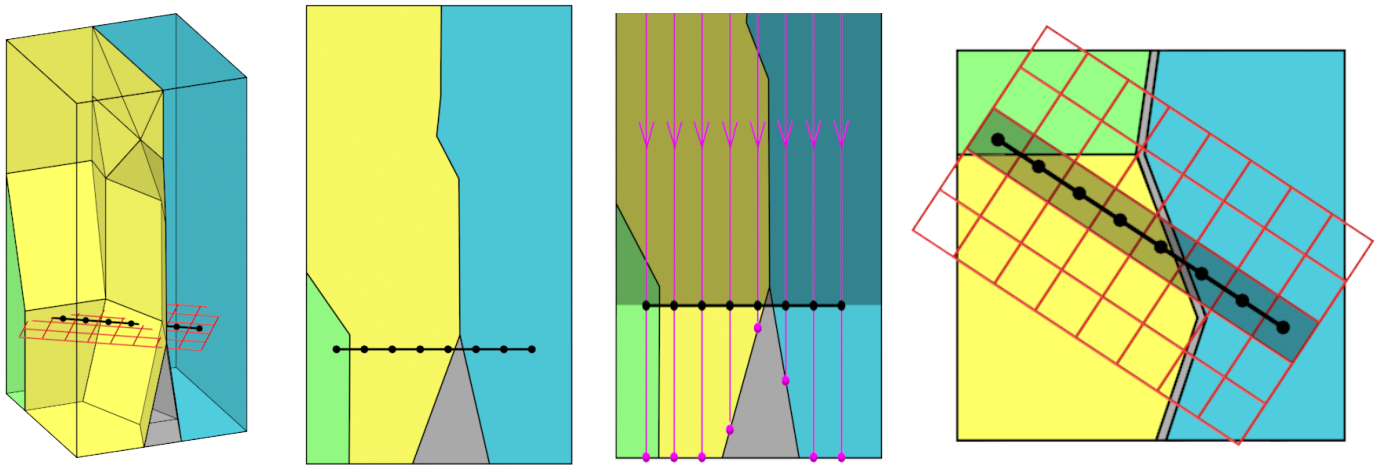
Intermezzo: Principle of rendering at client side



1. Determine slice plane (raster)
2. Discard part that is above slice plane
3. Shoot rays from sliceplane (each raster cell) towards bottom of the cube

Web-based system architecture

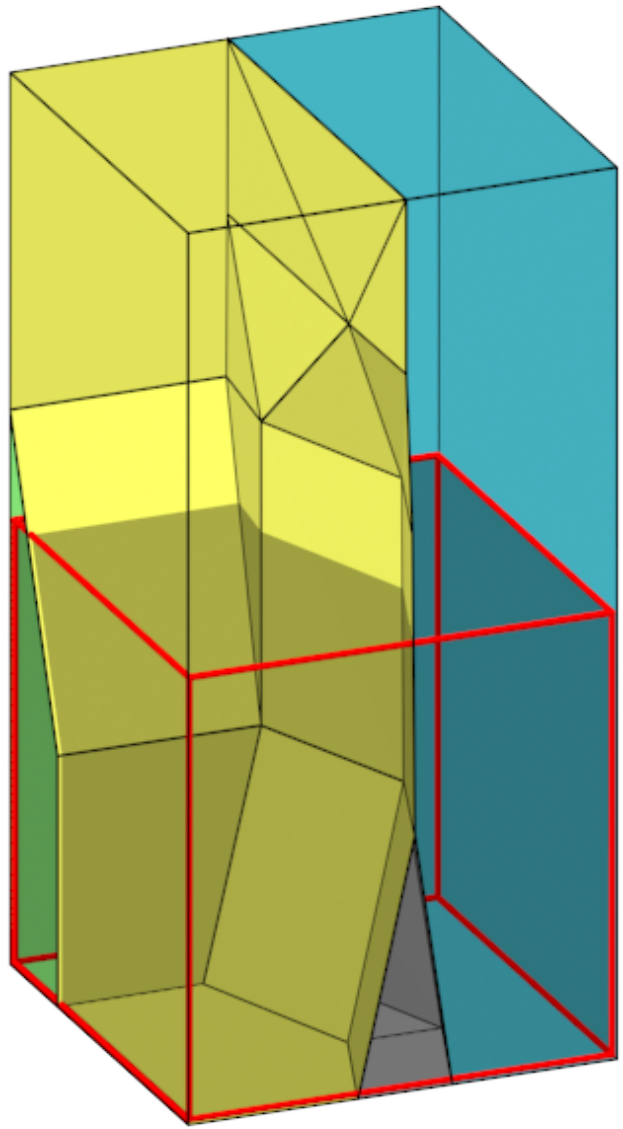
Intermezzo: Principle of rendering at client side



1. Determine slice plane (raster)
2. Discard part that is above slice plane
3. Shoot rays from sliceplane (each raster cell) towards bottom of the cube
4. First intersection with a triangle determines colour of pixel in raster

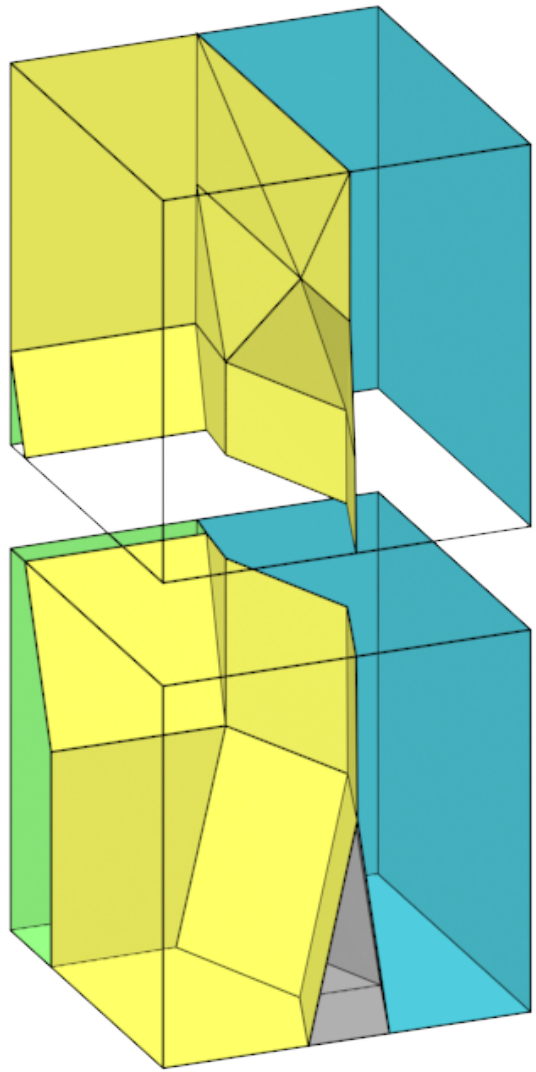
Web-based system architecture

Additional bottoms needed when splitting with Octree



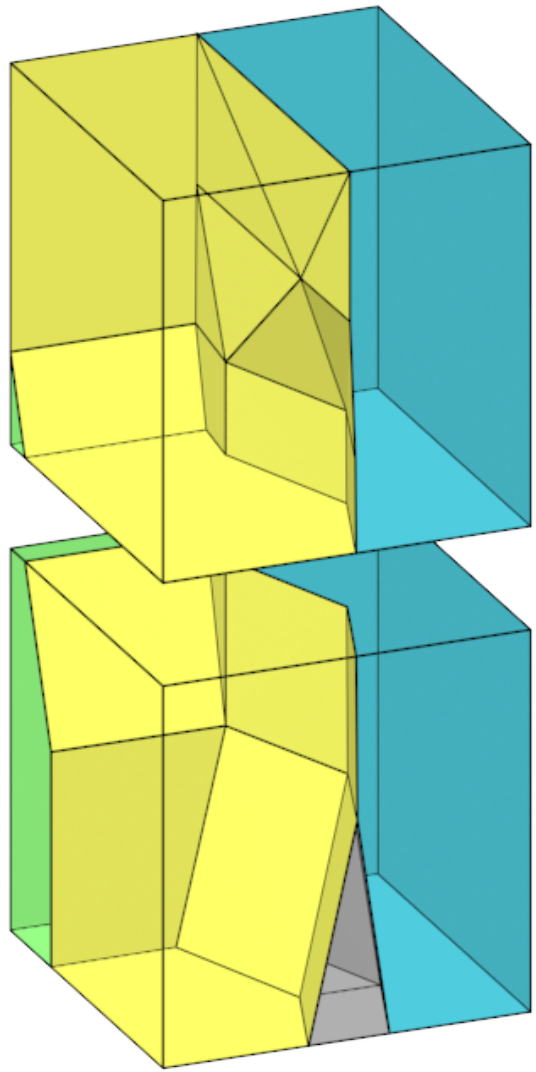
Web-based system architecture

Additional bottoms needed when splitting with Octree



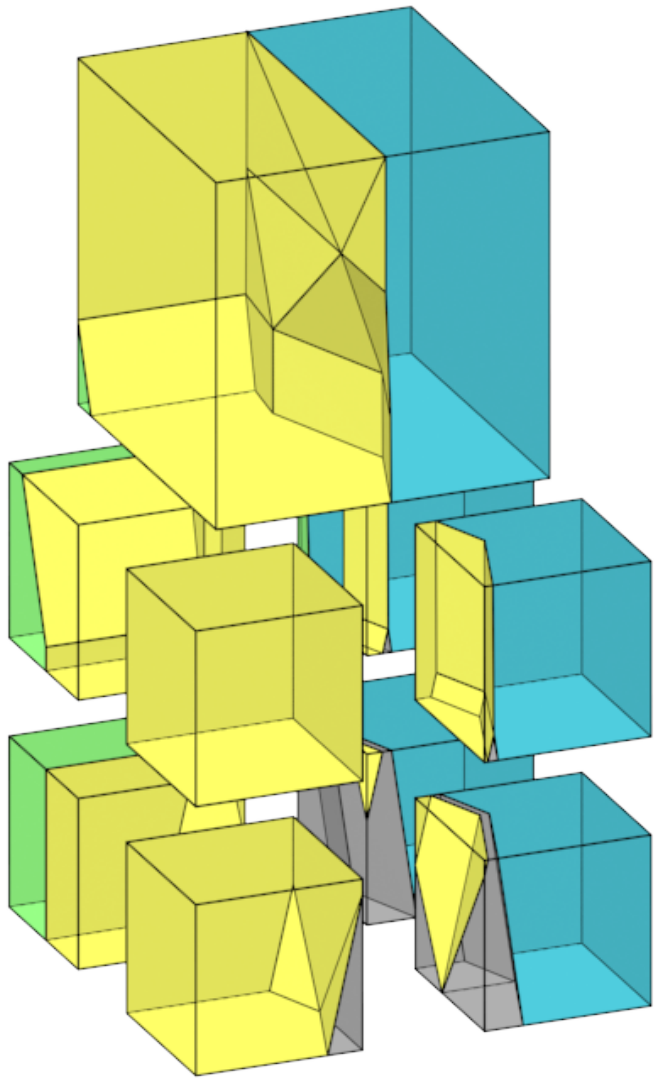
Web-based system architecture

Additional bottoms needed when splitting with Octree



Web-based system architecture

Additional bottoms needed when splitting with Octree



Web-based system architecture

Server: Make data blocks available

- Implemented: Octree

Web-based system architecture

Server: Make data blocks available

- Implemented: Octree
- Splitting introduces additional geometry:
  We have observed increase of 6× original data size

Non-split versus Split data set size

Dataset	Size (MB)	
	1 block	*n* blocks
Leiden	0.7	0.93
Limburg (9x9km)	40	233

Web-based system architecture

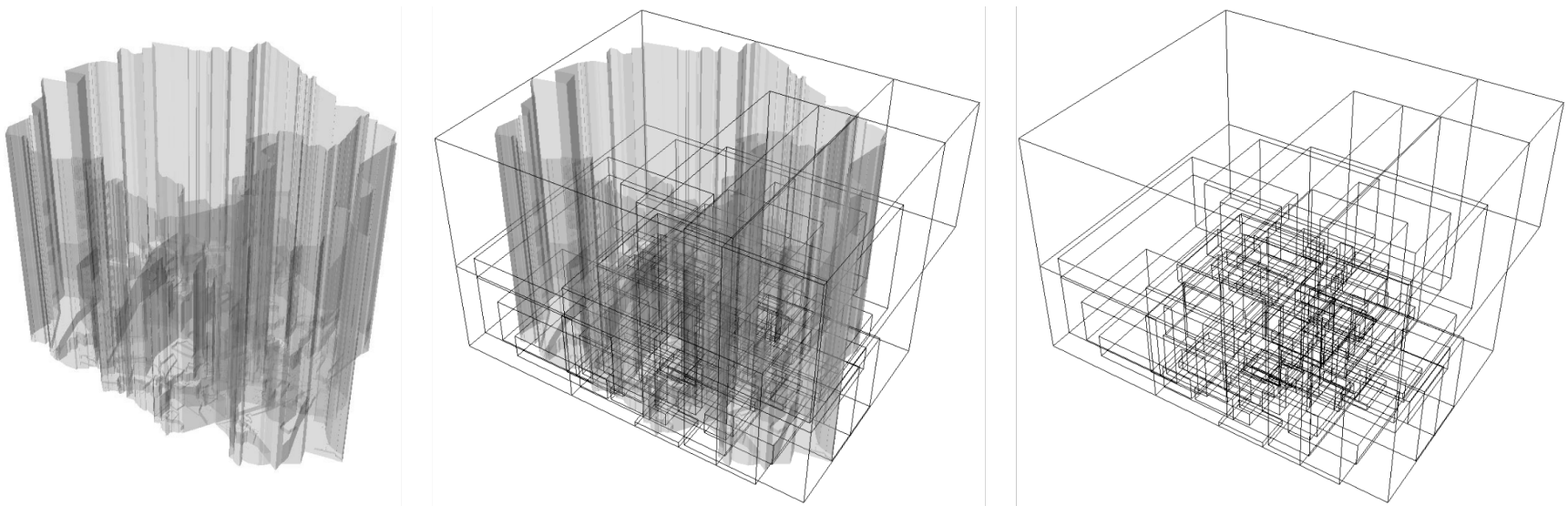
Server: Make data blocks available

- Work in progress: Organize blocks by not cutting object geometry
- Keep 1 object completely in 1 block
- Put objects that are close in space and scale together in 1 block

Web-based system architecture

Server: Make data blocks available

- Work in progress: Organize blocks by not cutting object geometry
- Keep 1 object completely in 1 block
- Put objects that are close in space and scale together in 1 block



Web-based system architecture

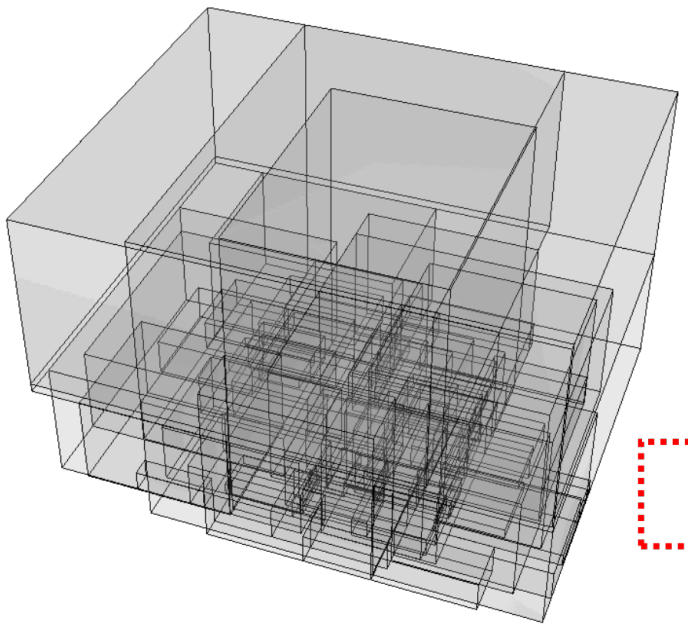
Server: Make data blocks available

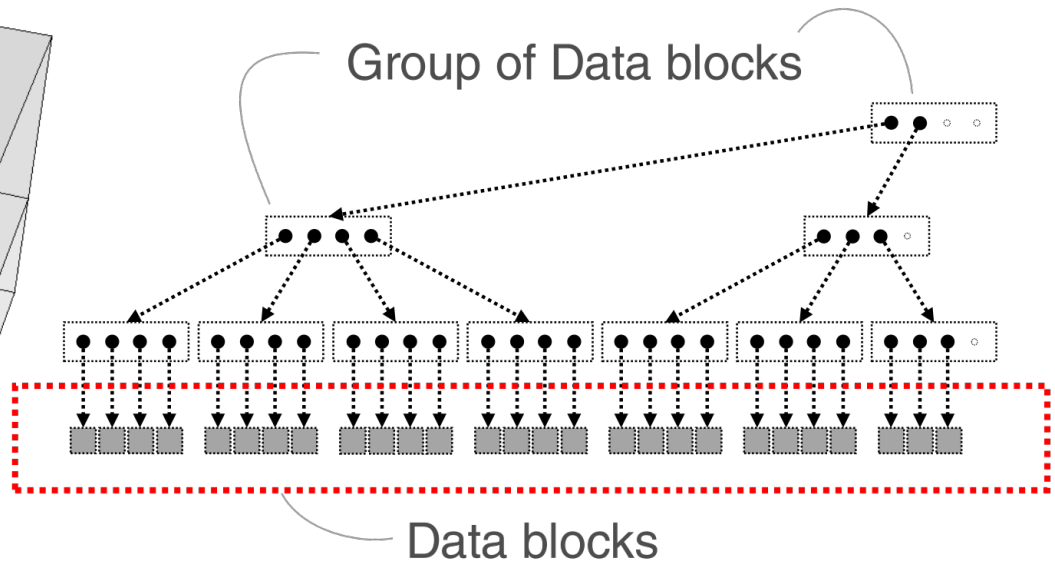
- Which blocks are available: Block index
- Block index $=$ Tree structure on the blocks

Web-based system architecture

Server: Make data blocks available

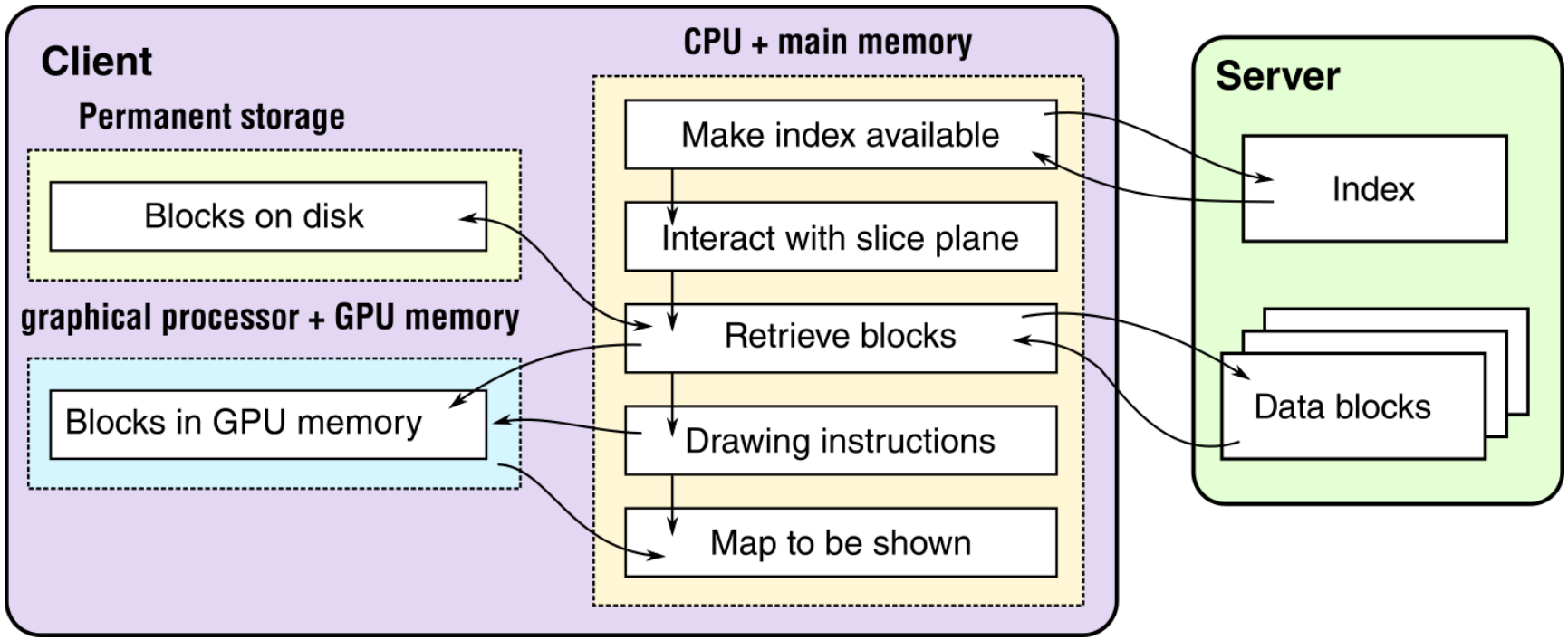
- Which blocks are available: Block index
- Block index = Tree structure on the blocks
- Using the tree it is possible to:
  - Determine fast which blocks are needed
  - Record status of block: In transfer, Succesfully transferred, In GPU memory . . .





Web-based system architecture

Architecture at client-side more in-depth



Implemented with HTML5 and Javascript

Client side: Interaction principles

Javascript is *event driven*
When an event is emitted a function (event handler) is run

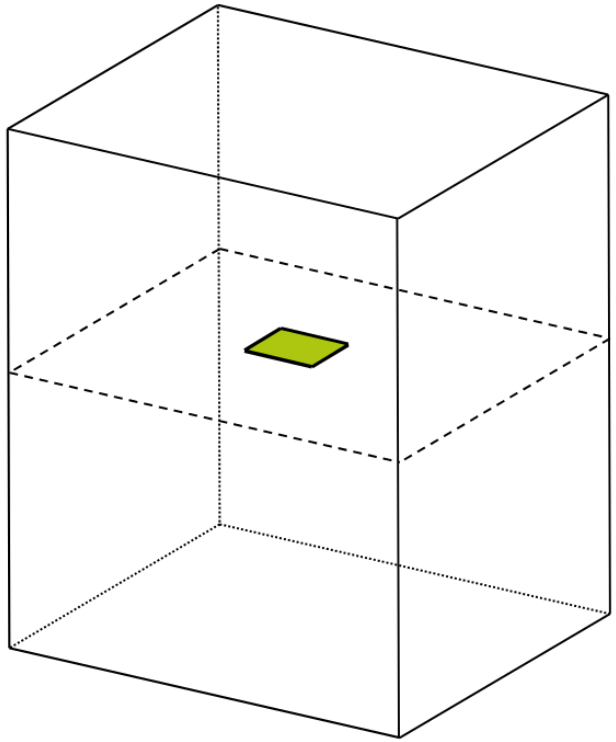
Client side: Interaction principles

Javascript is *event driven*
When an event is emitted a function (event handler) is run
For example:

- Left mouse button pressed and mouse is moved: Event handler for panning will be run
- Mousewheel clicks forward or backward: Event handler for zoom in/out is invoked

Client side: Interaction principles

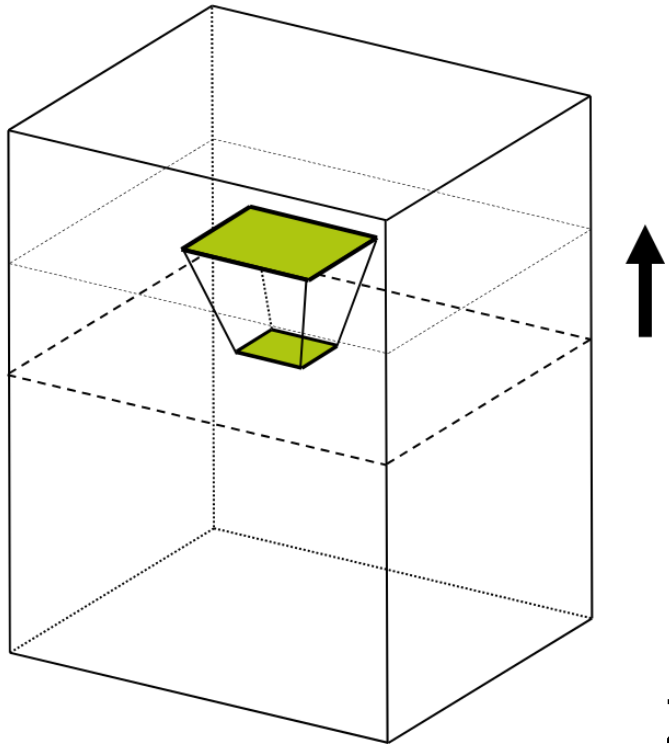
Interaction with the slice plane = showing a map



Initial position

Client side: Interaction principles

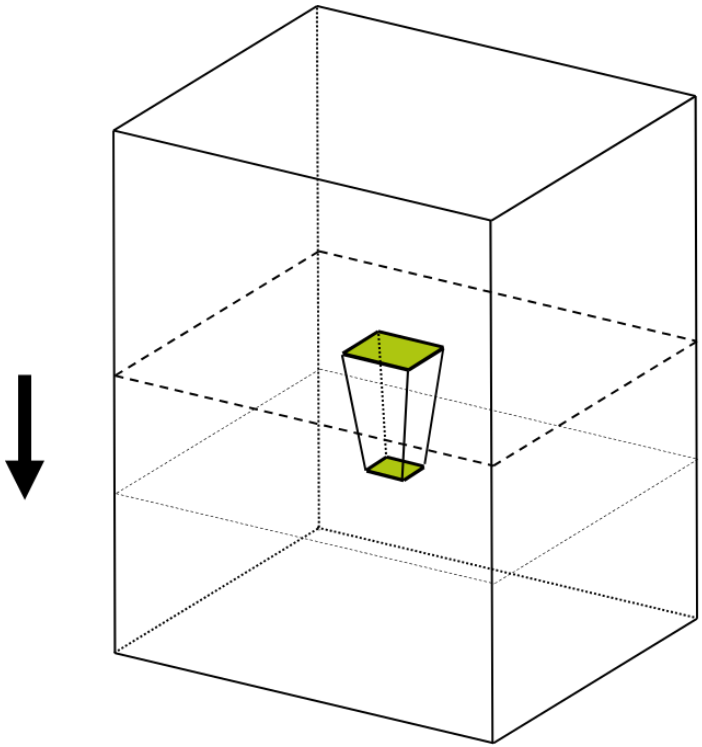
Interaction with the slice plane = showing a map



Zooming out

Client side: Interaction principles

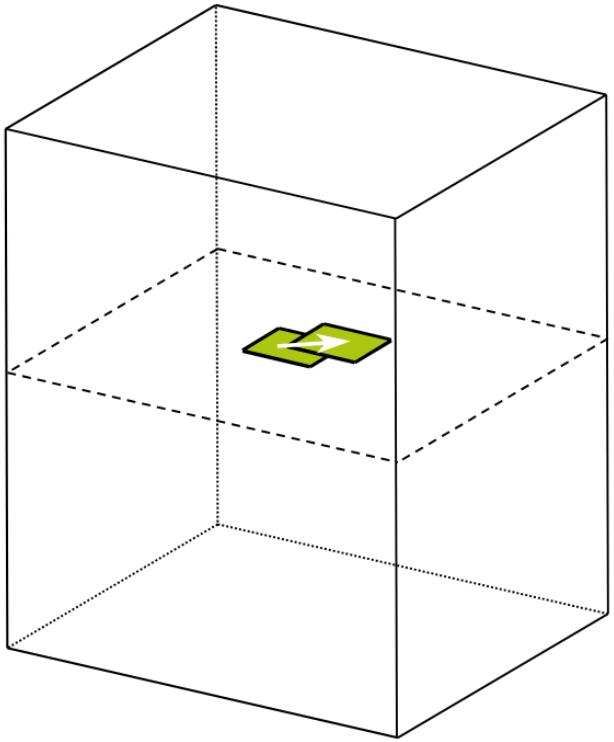
Interaction with the slice plane = showing a map



Zooming in

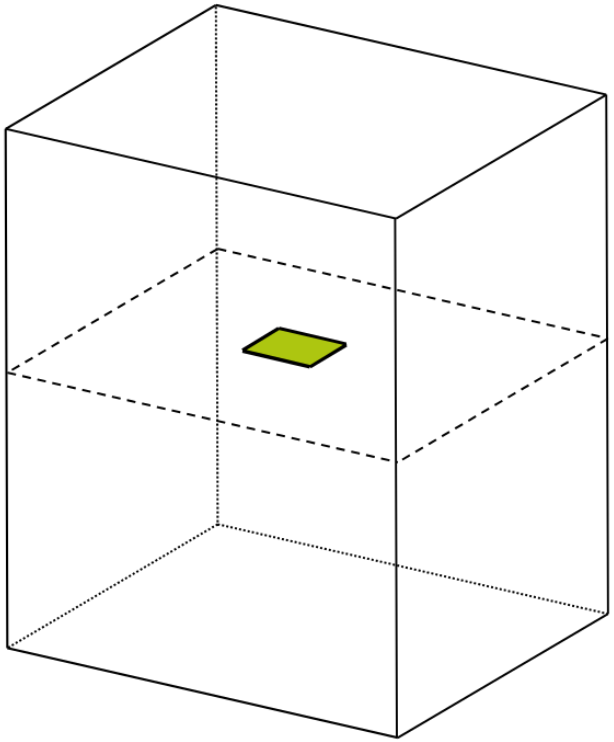
Client side: Interaction principles

Interaction with the slice plane = showing a map

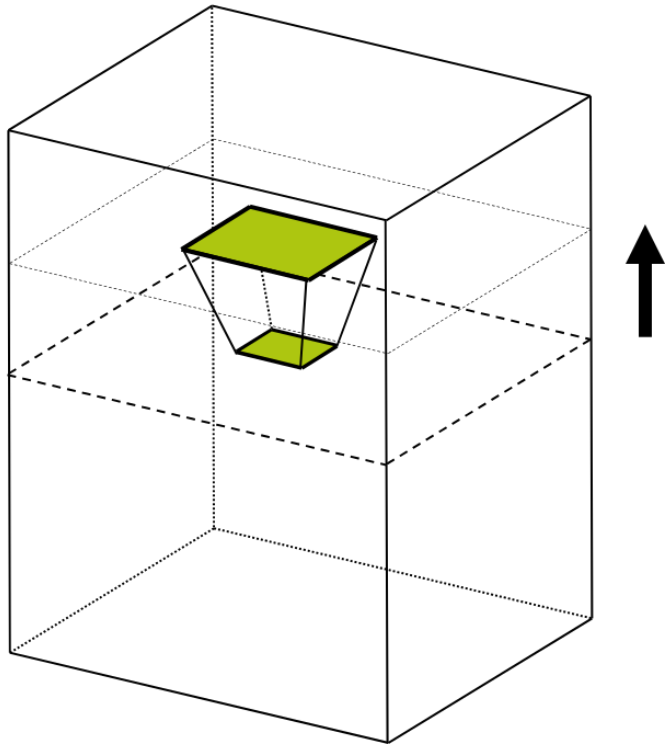


Panning

Client side: Need for animation

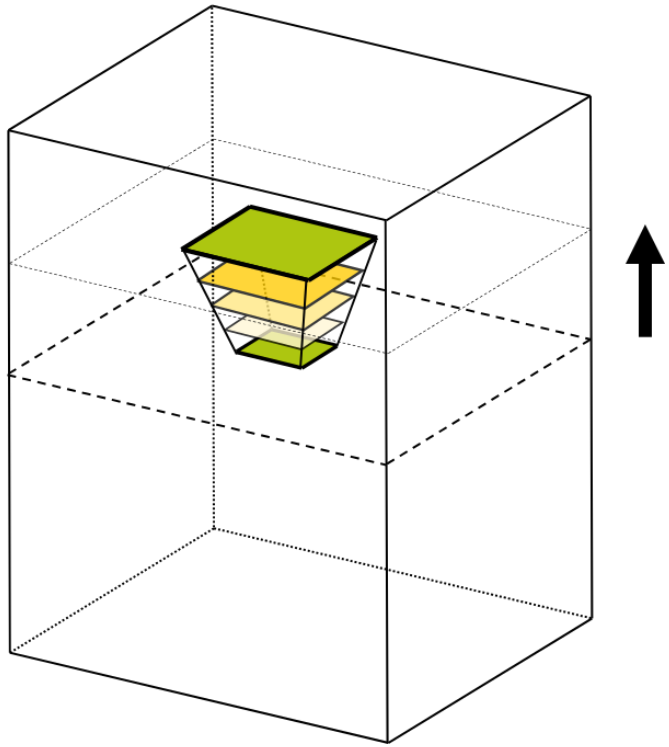


Client side: Need for animation



User skips a part of the modelled smooth transition

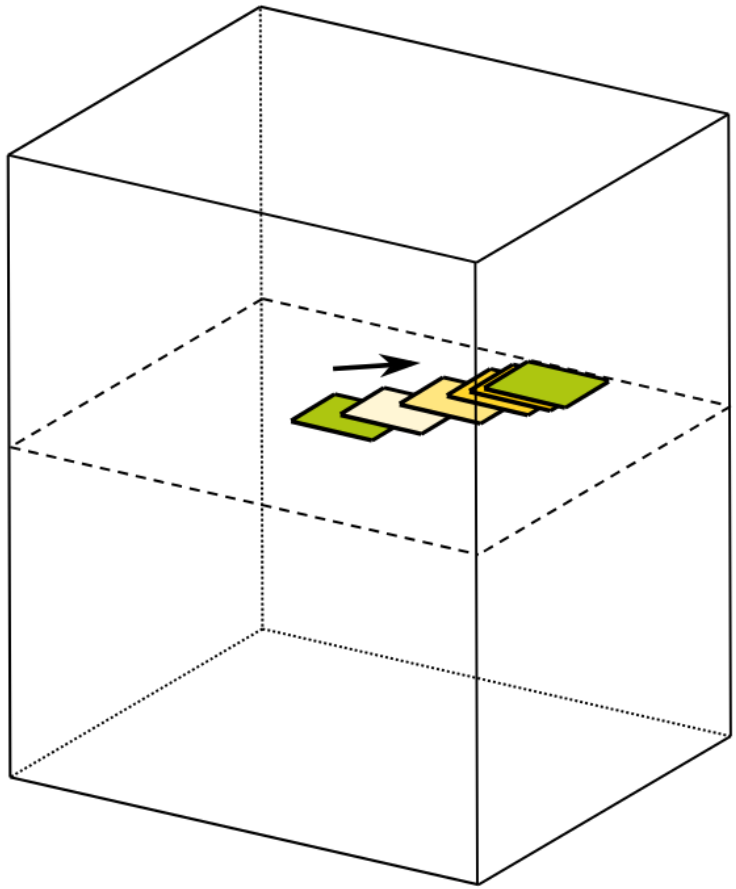
Client side: Need for animation



User skips a part of the modelled smooth transition
Need for animating the transition

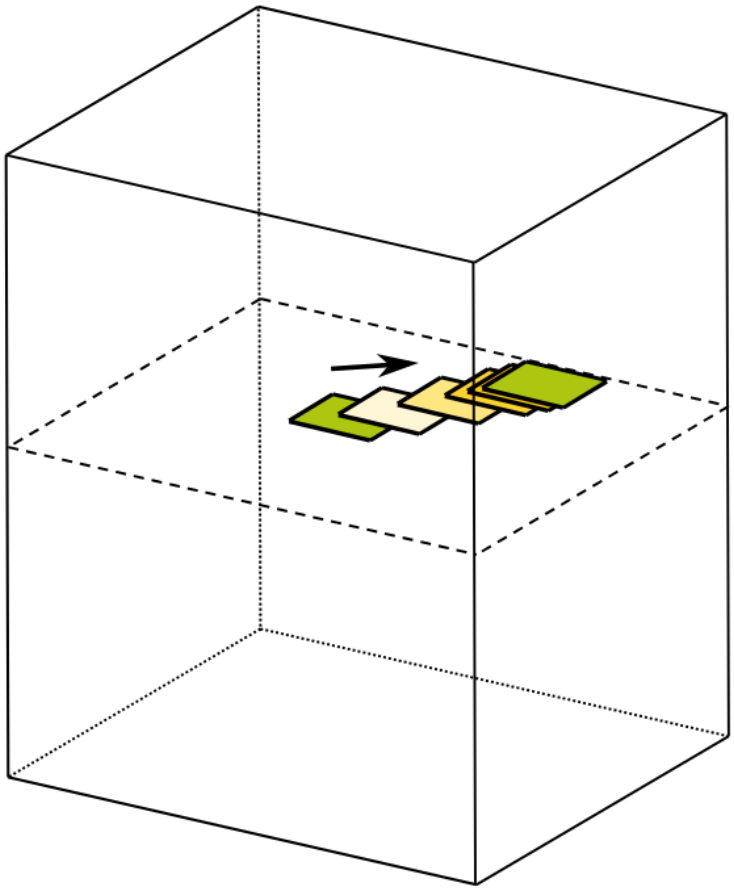
Client side: Easing the animation

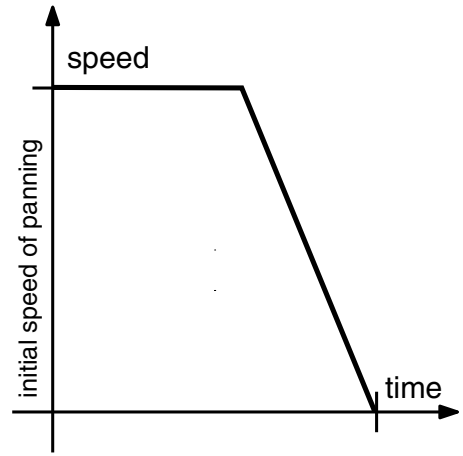
Make interaction even more smooth:
Easing – Explained for panning the map



Client side: Easing the animation

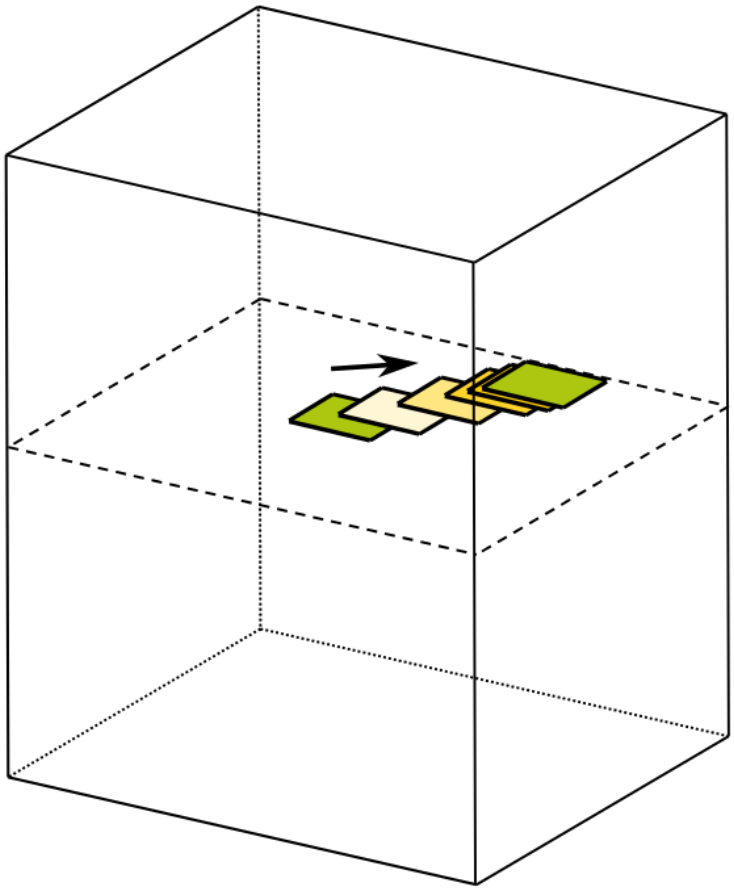
Make interaction even more smooth:
Easing – Explained for panning the map

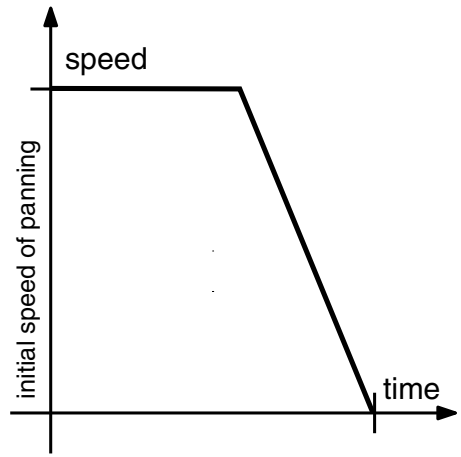


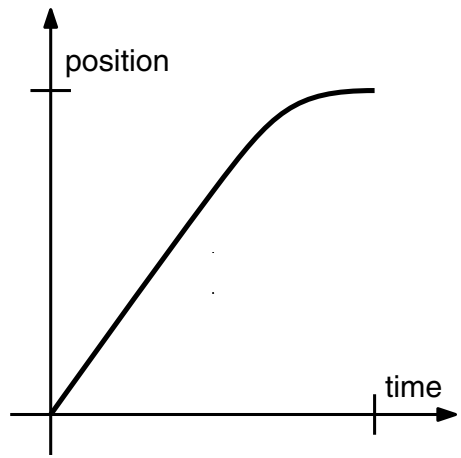


Client side: Easing the animation

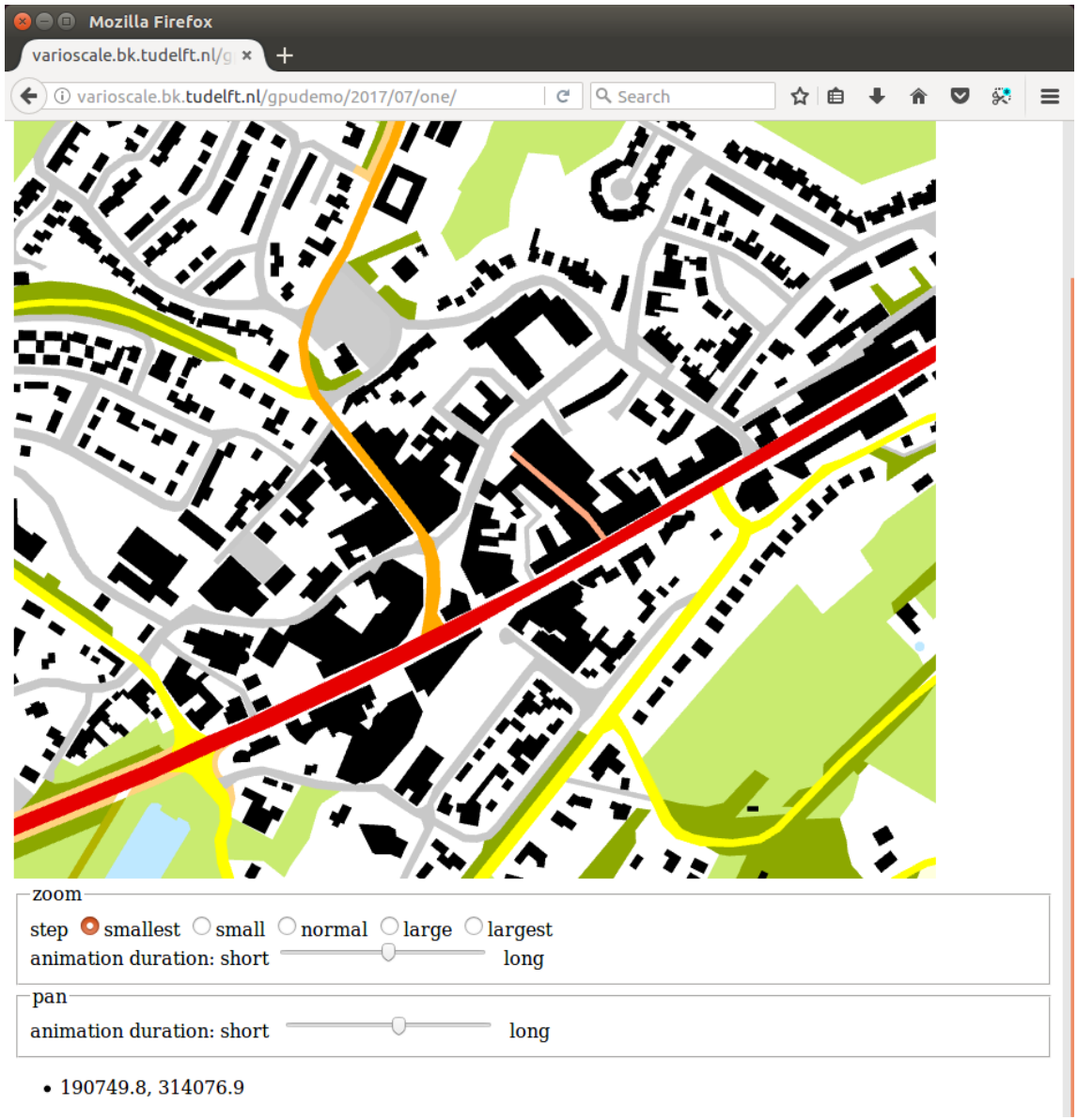
Make interaction even more smooth:
Easing – Explained for panning the map







Demo


Mozilla Firefox
varioscale.bk.tudelft.nl/g
varioscale.bk.tudelft.nl/gpudemo/2017/07/one/
Search
zoom
step smallest small normal large largest
animation duration: short long
pan
animation duration: short long
190749.8, 314076.9

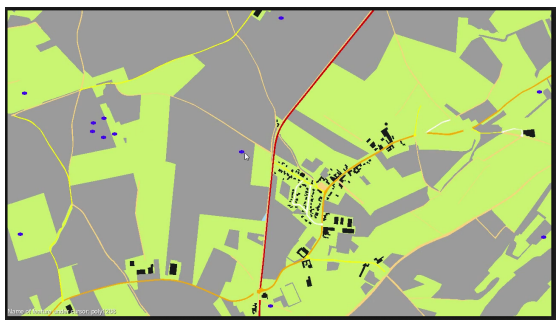
Future work

1. Implement blocks (+ tree structure) for large dataset

Future work

1. Implement blocks (+ tree structure) for large dataset
2. Advanced techniques (already developed in earlier desktop client — put on the web)
   - Smooth colour blending

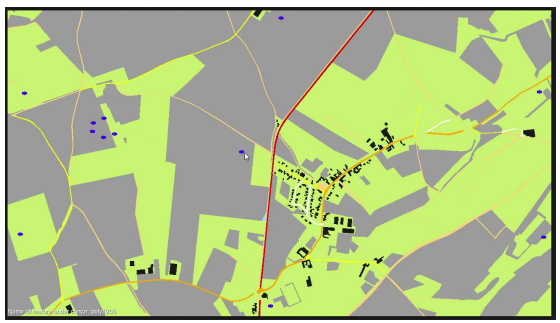
   - Put more detail around cursor (non-planar slice plane)



Future work

1. Implement blocks (+ tree structure) for large dataset
2. Advanced techniques (already developed in earlier desktop client — put on the web)
   - Smooth colour blending

   - Put more detail around cursor (non-planar slice plane)

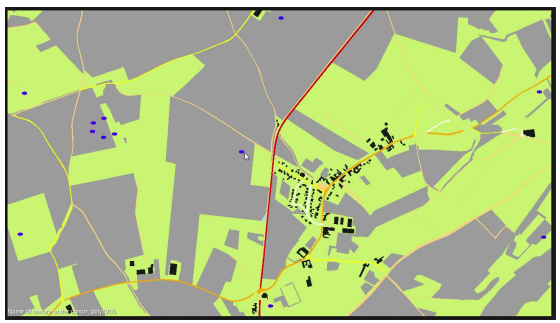


3. Make smooth transitions better visible

Future work

1. Implement blocks (+ tree structure) for large dataset
2. Advanced techniques (already developed in earlier desktop client — put on the web)
   - Smooth colour blending

   - Put more detail around cursor (non-planar slice plane)



3. Make smooth transitions better visible
4. Interaction on mobile/tablet + More user tests

Questions?

- dr.ir. Martijn Meijers
  b.m.meijers@tudelft.nl
  http://www.gdmc.nl/martijn/
  tel. (+31) 15 27 856 42

- Delft University of Technology
  Faculty of Architecture and the Built Environment
  OTB – Research for the built environment
  GIS Technology