

WEB CARTOGRAPHY

IN A WORLD OF SERVICES, SDI'S AND WEB 2.0

Barend Köbben
<kobben@itc.nl>

EuroSDR workshop, Lund, 5 May 2011



Background (I)

changing role of cartography in a changing world:

- information disseminated in digital ways
- all about sharing, interoperability, web services, SDIs and the modern two-way Web 2.0
- this has consequences on the design of (web)cartography solutions in this environment
- subject of research projects in our group at ITC

Background (2)

Webcartography research projects at ITC

- share the larger aim of improving mapping within the framework of *loosely coupled, distributed webservices*
- fit within our *SDI^{light}* approach

Background (2)

Webcartography research projects at ITC

- share the larger aim of imp framework of *loosely couple*
- fit within our *SD/light* appro

SD/light



Background (2)

Webcartography research projects at ITC

- share the larger aim of imp framework of *loosely couple*
- fit within our *SDI^{light}* appro

SDI



SDI^{light} approach

- a down-to-earth approach towards SDI
- Open Standards whenever available
- Open Source where possible
- used in teaching, projects and research
- provides researchers, students and partners with a platform for relatively simple, low-cost, yet powerful ways of sharing data amongst various stakeholders



SDI^{light} software stack

client–side

application layer
(middle–ware)

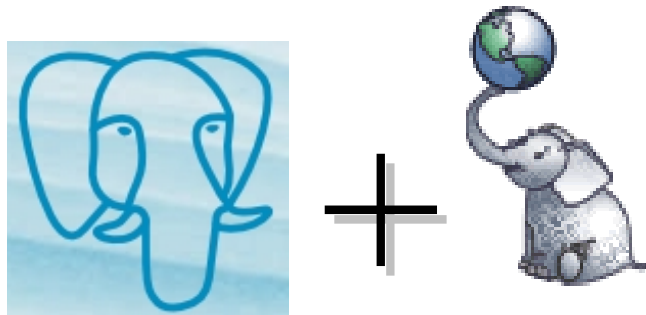
data layer
(back–end)



SDI^{light} software stack

client-side

application layer
(middle-ware)

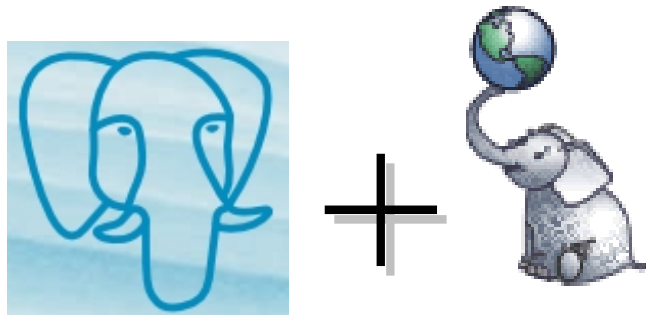


SDI^{light} software stack

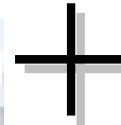
client-side



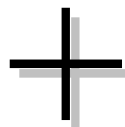
MAPSERVER



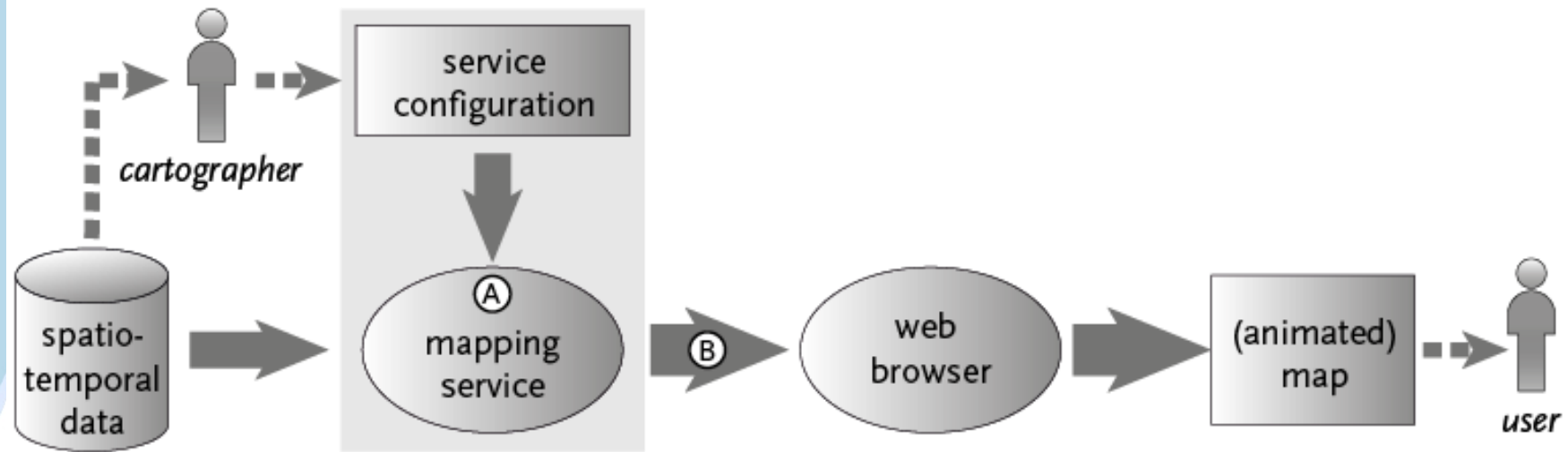
SDI^{light} software stack



MAPSERVER

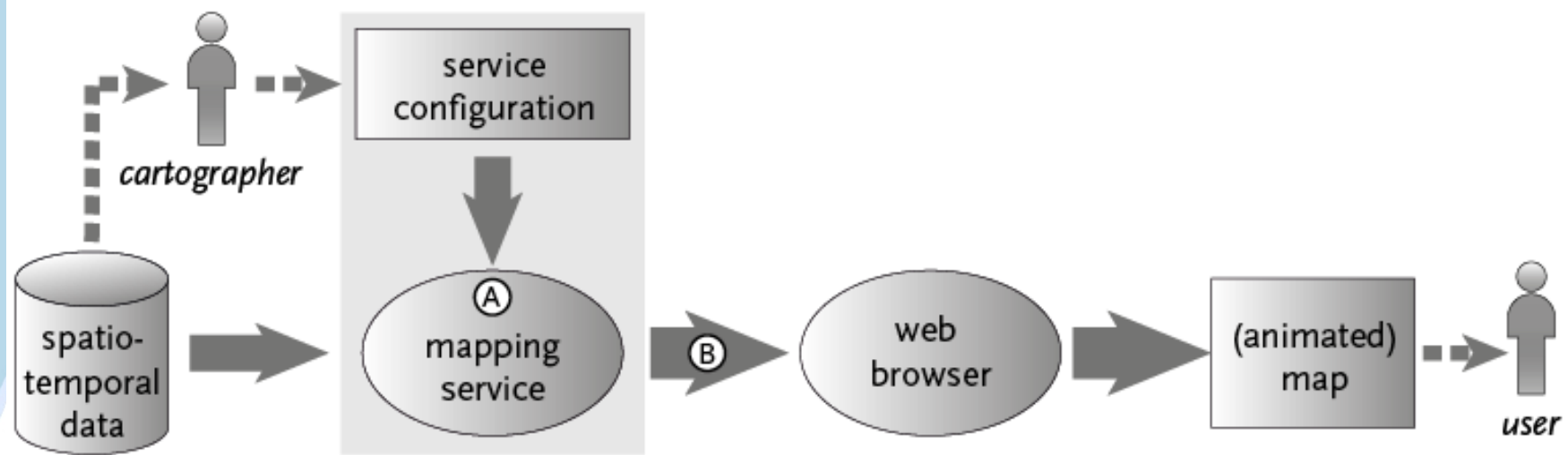


Mapping in a webservices environment



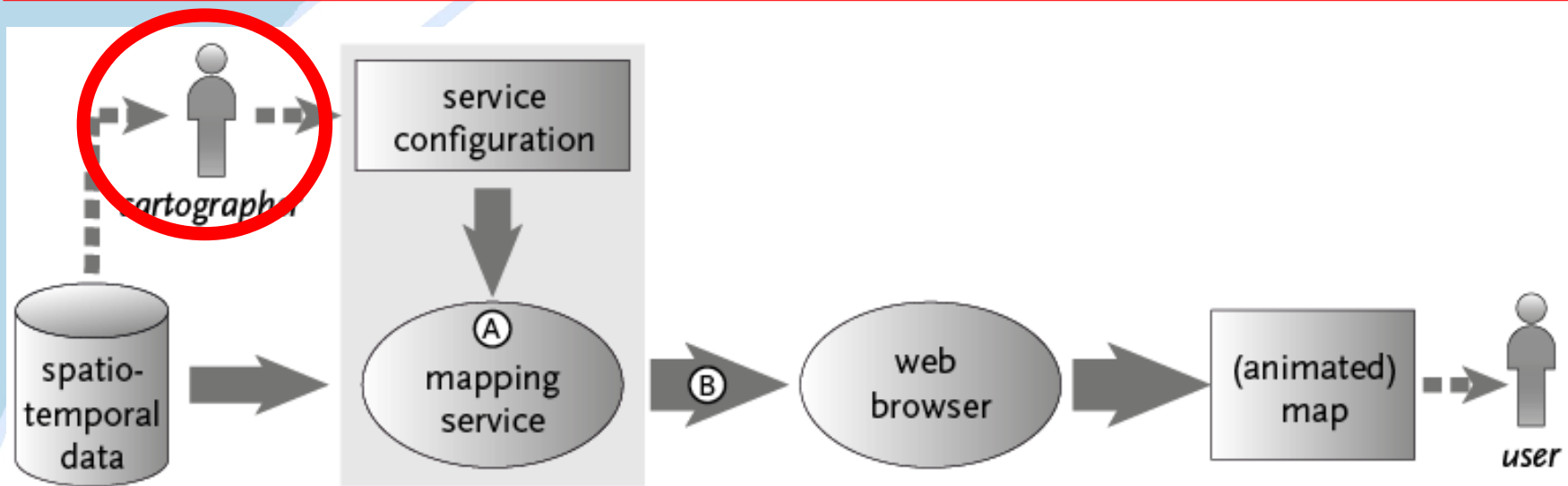
- possibilities for **direct** and *automatic* production of maps
- where 'direct' means:
 - generated case-by-case and on-the-fly from the data, no conversion or pre-processing needed for purpose of visualisation only
 - ▶ important for system to be an SDI node
 - ▶ able to consume data from any other SDI node

Mapping in a webservices environment



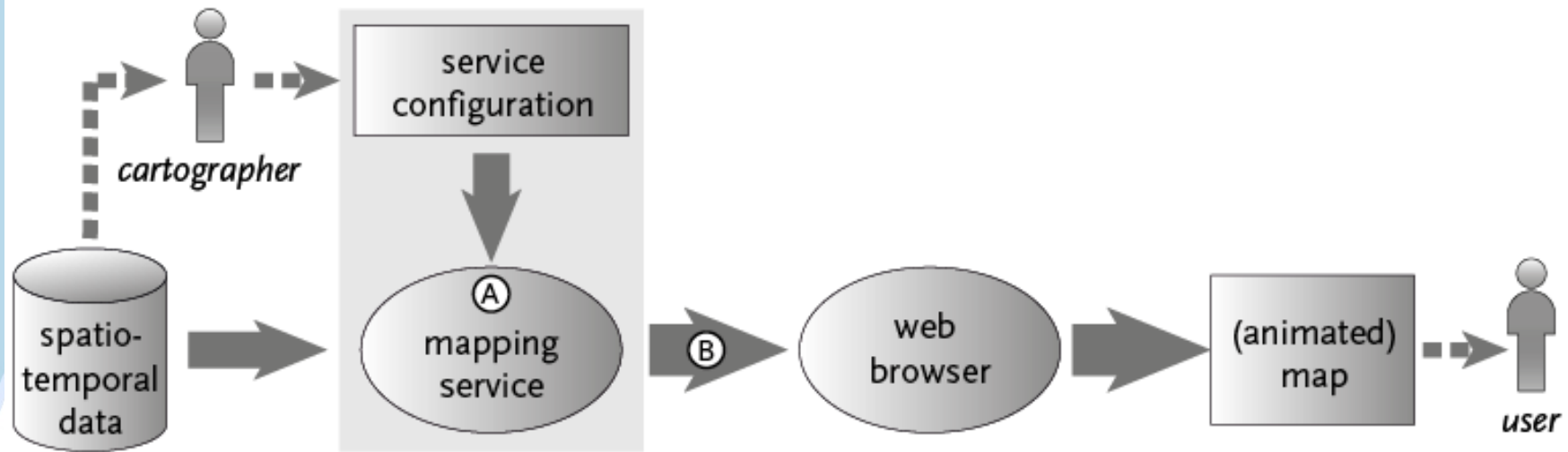
- possibilities for *direct* and **automatic** production of maps
- where ‘automatic’ means:
maps will be generated from the spatio-temporal data by the system “*working by itself with little or no direct human control*” (Concise Oxford Dictionary of Current English)

Mapping in a webservices environment



- possibilities for *direct* and **automatic** production of maps
- where 'automatic' does NOT mean:
the system "*simulates human action*" (Oxford English Dictionary)
 - ▶ including the cartographic decisions as to what type of map and what map properties to use for different data–types and data–instances

Mapping in a webservices environment



- nowadays a very important dissemination channel
- but partly takes us back to “the old days”:
 - ▶ “pre-cooked” maps in a one-way process
 - ▶ little user influence on design and content
 - ▶ little interactivity and exploration possibilities
 - ▶ “cartographer” (map-maker) \neq user determines most of the map design and usability

How to increase interactivity & user input..?

using rich map formats:

Scalable Vector Graphics

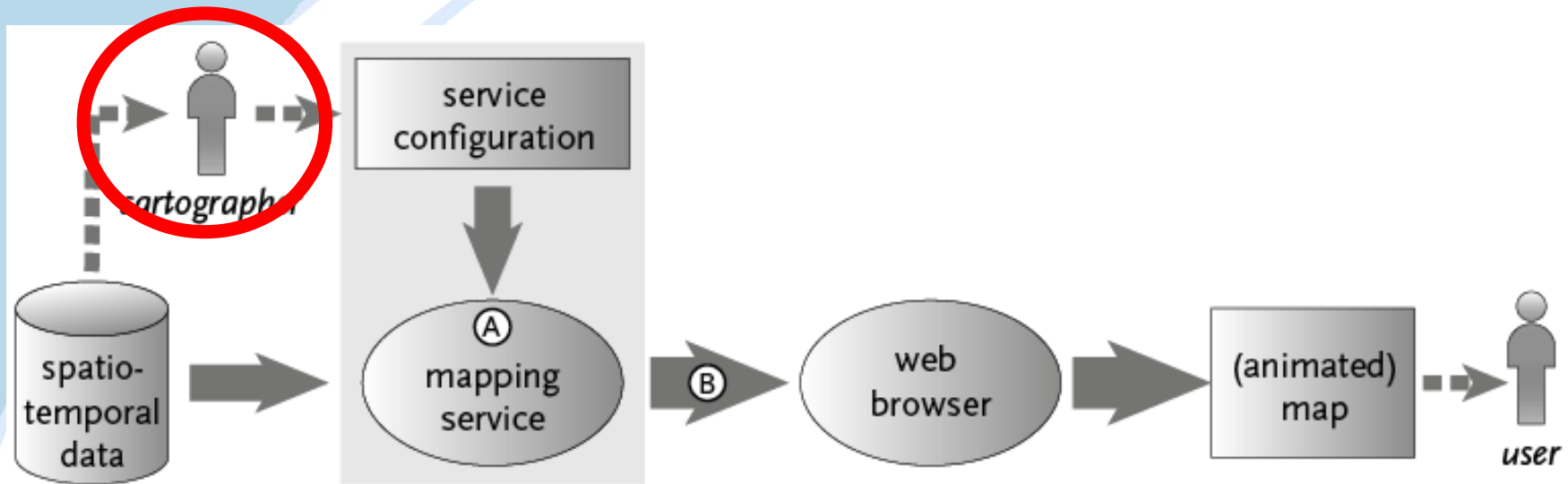
- SVG is open standard XML-based vector graphics
 - ▶ High quality (carto)graphics & attribute info
- Some WMS exist with (limited) SVG, but all treat SVG as 'static graphics format' only
 - ▶ SVG also can hold attribute data
 - ▶ SVG also can provide animation
 - ▶ SVG also can provide application logic
- ➔ Can support built-in Graphical User Interface (GUI)
- ➔ Can support animated maps

RIMapperWMS and TimeMapper

- spatial database back-end (PostgreSQL/PostGIS)
 - ▶ spatial and attribute data
 - ▶ Web Map Service configuration
- server application (Java)
 - ▶ responds to WMS compliant requests
 - ▶ provides output in SVG
 - ▶ with built-in GUI
 - ▶ with built-in animation
- mobile or desktop web client
 - ▶ renders interactive & dynamic SVG maps

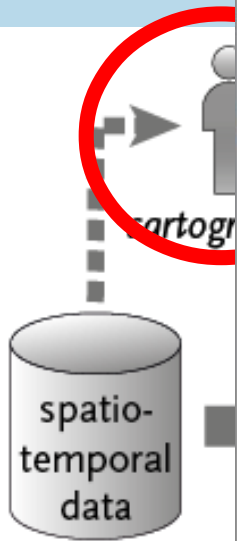


How to automate it fully..?



- needed for mapping services that can adjust to changing circumstances
 - ▶ e.g. data source changes
 - ▶ e.g. data instance updates

How



The screenshot shows a web-based GIS application interface. On the left, a menu titled "KIES ONDERWERP ..." lists various data layers. The layer "Aantal inwoners per gemeente" is selected and highlighted in blue. Below the menu is a search bar labeled "ZOEK ..." with a search icon. The main area is a map of the Netherlands, where the size and color of squares represent population density per municipality. On the right, a "LEGENDA" panel shows the title "Aantal inwoners per gemeente" and the source "Bron: ITC". Below the legend is a "MEER INFORMATIE ..." panel displaying technical details such as "HTTPStatus is 0", "finish loading at: 15h:19m:4,198s", and "Map-object creation took 4.349 seconds." At the bottom of the map, a status bar indicates "Het laden van de kaart duurde 7.541 sec." and "© Geografiek/Landkaartje, 11-11-08, v0.8".

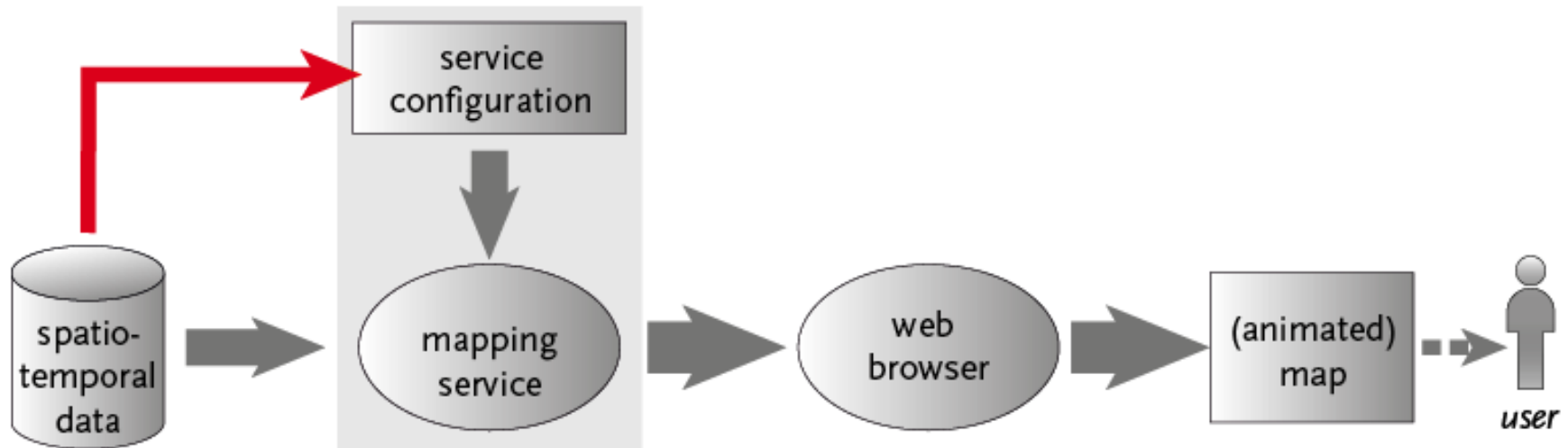
■ neede
circur

- ▶ e.g.
- ▶ e.g.

became apparent in project
3rd edition National Atlas for the Netherlands



How to automate it fully..?



- create service configuration from data
- *direct*, based on:
 - ▶ cartographic knowledge
 - ▶ intended user and/or usage of the map
 - ▶ properties of the data

This is not a recent challenge...!

- ▶ B.J. Köbben (1988): Choro-Expert: a front-end expert system determining data-appropriateness for choropleth mapping (MSc thesis, Utrecht)
- 1980s: the DLM-DCM paradigm was introduced
Digital Landscape Model → *Digital Cartographic Model*
- the automatic generation of DCMs from DLMs has been subject of a small surge of research
 - ▶ in the 80s using Knowledge Based / Expert Systems
- it somehow this never took off ...
... but it should(?) re-emerge in a service environment

Why did this not happen (yet)...?

We think it's a case of missing information:

- cartographic knowledge
- intended user and/or usage of the map
- properties of the data



Why not...?

We think it's a case of missing information:

- cartographic knowledge
→ *solved for 'simple' maps (most common cases)*
- intended user and/or usage of the map
- properties of the data



Why not...?

We think it's a case of missing information:

- cartographic knowledge
 - *solved for 'simple' maps (most common cases)*
- intended user and/or usage of the map
 - *solvable for 'simple' use goals (most common cases)*
- properties of the data



Why not...?

We think it's a case of *missing information*:

- cartographic knowledge
 - *solved for 'simple' maps (most common cases)*
- intended user and/or usage of the map
 - *solvable for 'simple' use goals (most common cases)*
- properties of the data
 - *this is the **main culprit**:*
 - ▶ *meta-data is a problem (in real life) anyway*
 - ▶ *automatic meta-data generation even more*
 - ▶ *the meta-data needed is not the usual set only (e.g. measurement level)*



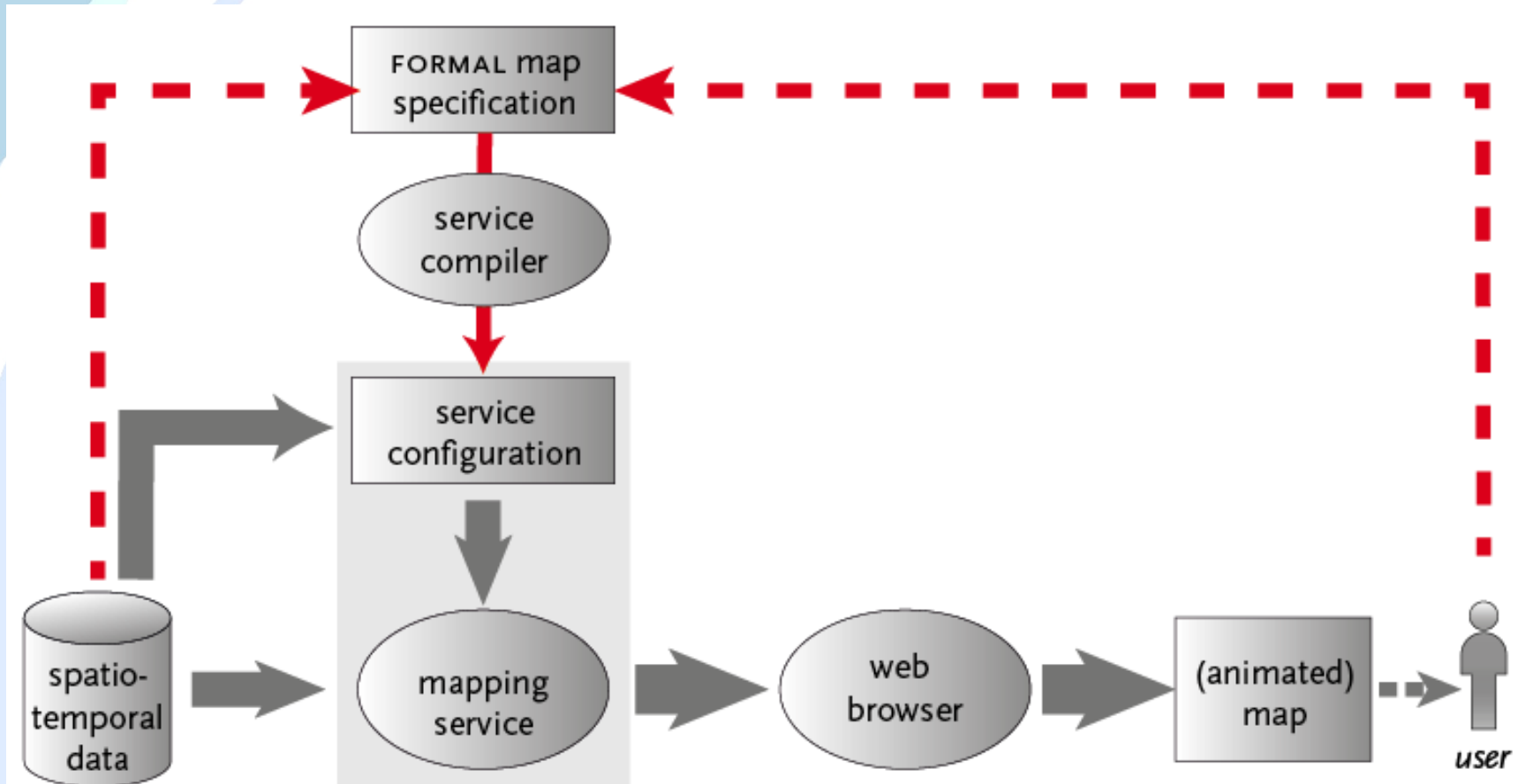
Why not...?

- another missing part:
a *FORMAL map specification language*
 - ▶ \neq service configuration file (e.g. SLD)
 - this is what created by a compiler *based* on the formal map spec, plus (meta–)data and user input
 - ▶ \neq traditional map specifications (e.g. topomap specs)
 - these are focussed on producing a specific *map product*, we want a focus on *information output*
 - ▶ *formalised specification* in the computer science sense
 - defines an outcome using a formal (declarative) language
 - with degrees of freedom (e.g. ranges of acceptable values)
 - in a controlled and consistent manner



Towards automatic mapping in services environment

- possible use of a *FORMAL map specification language*



QUESTIONS...?

Barend Köbben

kobben@itc.nl

<http://www.itc.nl/>

<http://kartoweb.itc.nl/kobben/>

<http://www.nationaleatlas.nl/>

<http://kartoweb.itc.nl/RIMapper/>

<http://kartoweb.itc.nl/TimeMapper/>