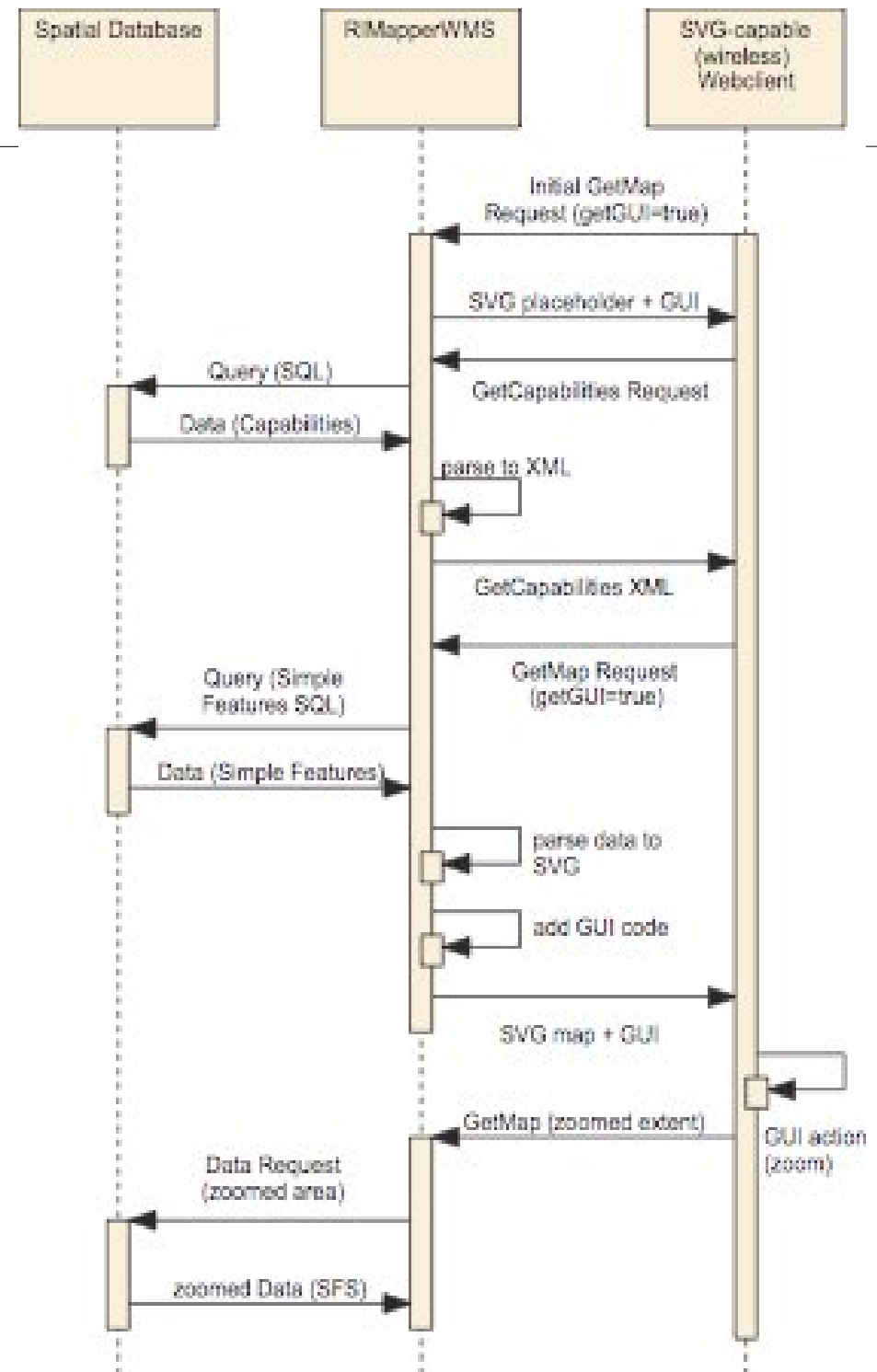


Why a built-in GUI?

No need for separate client application:
“output = application”

- simple WMS conformant interface to the data
- data includes built-in client-side GUI
- GUI handles the map interaction and generates further requests



General setup of RIMapperWMS

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



spatial database back-end (PostGIS)

column
*PK id: integer = nextval('svg_st...')
name:
style:

PK
+ css_styles_pkey(integer)

column
abstract:
classes:
*PK id: integer = nextval('wms_st...')
legend_url_format:
legend_url_height: smallint
legend_url_online_resource:
legend_url_width: smallint
* name:
styleattribute:
* styletype: = 'single':chara...
svgstyles:
* title:

PK
+ id(integer)

column
order: varchar(9999)
* id: integer
* name: = 'roads':character v...
type: varchar(32) = 'character v...

WMS styling

column
auth_name: varchar(256)
auth_srid: integer
proj4text: varchar(2048)
*PK srid: integer
srs_text: varchar(2048)

PK
+ spatial_ref_sys_pkey(integer)

column
* coord_dimension: integer
*PK f_geometry_column: varchar(256)
*PK f_table_catalog: varchar(256)
*PK f_table_name: varchar(256)
*PK f_table_schema: varchar(256)
* srid: integer
* type: varchar(30)

PK
+ geometry_columns_pk(varchar, varchar, varchar, varchar)

PostGIS spatial metadata

column
abstract:
access_constraints: = 'none':charact...
contact_electronic_mail_address:
fees: = 'none':charact...
*PK id: integer = nextval('servic...')
keyword_list:
name: = 'OGC:NMS':char...
title:

PK
+ id_pkey(integer)

column
abstract:
* geom_col: = 'ogc_geom':cha...
*PK id: integer = nextval('wms_la...')
keyword_list:
metadata_url:
* name:
opaque: smallint = 0
pkey: = 'gid':characte...
queryable: smallint = 0
scalehint:
srs_epsg_list:
style_list:
title:

PK
+ wms_layers_id_pkey(integer)

WMS metadata

column
layer n
layer ...
layer 1
column
*PK gid: integer = nextval('roads_...')
the_geom:
type: varchar(32)

check
+ enforce_dims_the_geom()
+ enforce_geotype_the_geom()
+ enforce_srid_the_geom()
PK
+ roads_pkey(integer)

spatial & attribute data per 'layer'

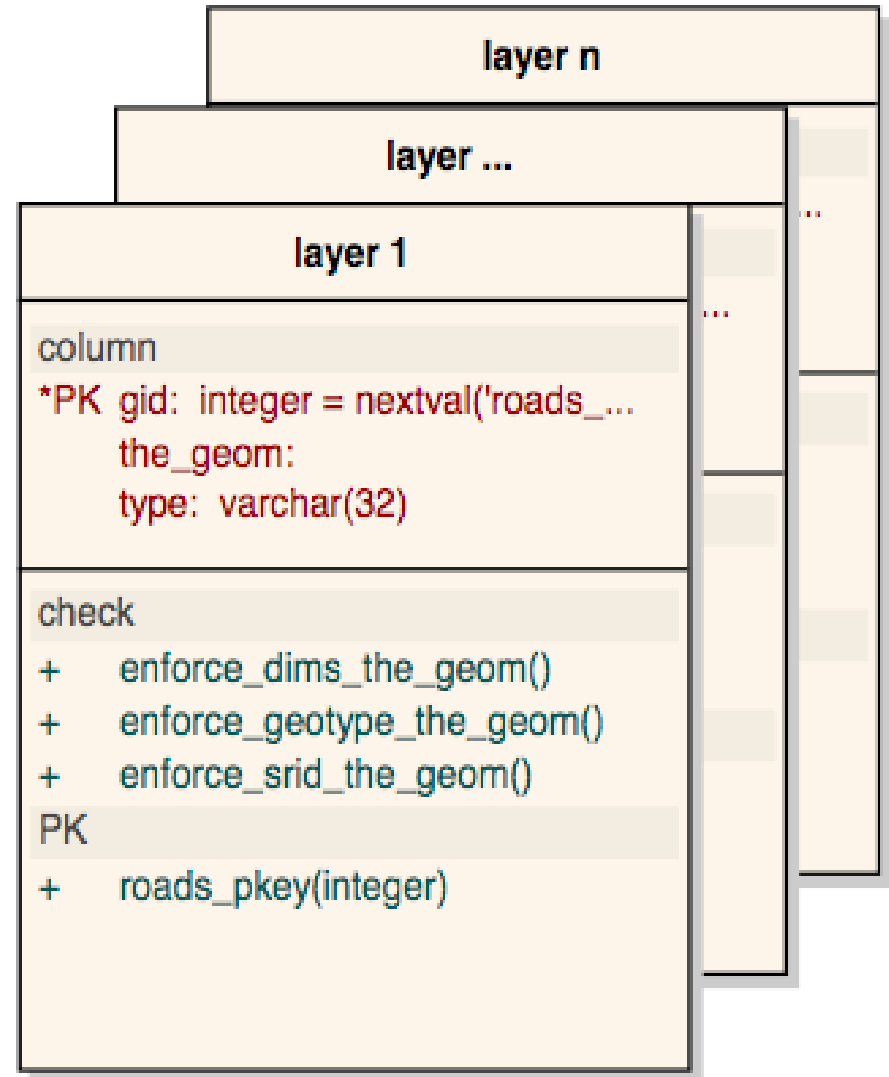
Spatial data layer tables

Object geometries in PostGIS
GEOMETRY objects

- follows OGC Simple Features Specification
- spatially indexed
- (re-)projectable

Object attributes

Can come from many data sources (eg. shp2pgsql)



WMS metadata tables

- Defines the WMS instance metadata
- Lists available layers and their:
 - projection data
 - extent
 - styles
 - etc...

service_metadata	
column	
	abstract:
	access_constraints: = 'none':character...
	contact_electronic_mail_address:
	fees: = 'none':character...
	*PK id: integer = nextval('servic...
	key...
*	na...
*	titl...
	PK
+	id

wms_layers	
column	
	abstract:
*	geom_col: = 'ogc_geom':character...
*PK	id: integer = nextval('wms_la...
	keyword_list:
	metadata_url:
*	name:
	opaque: smallint = 0
*	pkey: = 'gid':character...
	queryable: smallint = 0
	scalehint:
*	srs_epsg_list:
	style_list:
*	title:
	PK
+	wms_layers_id_pkey(integer)

WMS styling tables

- Defines available styles from WMS perspective
- Defines underlying SVG graphic styles
- Multi-purpose table for SVG & script fragments (eg. GUI elements, interactivity event handlers, ...)

svg_styles	
column	
*PK id: integer = nextval('svg_st...)	
* name:	
style:	
PK	
+ css_styles_pkey(integer)	

wms_styles	
column	
abstract:	
classes:	
*PK id: integer = nextval('wms_st...)	
legend_url_format:	
legend_url_height: smallint	
legend_url_online_resource:	
legend_url_width: smallint	
* name:	
styleattribute:	
* styletype: = 'single'::chara...	
svgstyles:	
* title:	
PK	
+ id(integer)	

Status: 1.1

- Adheres to OGC WMS *Basic* 1.1.1 specification
- Supports `GetCapabilities` & `GetMap` requests
- `TIME` parameter supported: data can be selected based on time extent, as well as spatial extent
- Additional vendor-specific `getGUI` capability
- TimeMapper code included to offer SMIL animated SVG for point type data
- Known limitations & issues:
 - Output formats other than SVG (png,jpeg) not possible in combination with `GetGUI=true`.
 - External (cascaded) WMS layers are loaded in the background, without checking for succes or progress report
 - Extension to ISO 8601:1988(E) not fully implemented: Only contiguous `TIME` periods (`datetime_begin/datetime_end`) for now.
 - Animation GUI timeslider only works in Opera and FireFox (basic animation works also in Safari and other WebKit based browsers).
- Free, open source (*creative commons* license)

Outlook

Near future plans:

- WMS setup application for Database
- WMS 1.3.0 support (depends on Proj4 library)

and further...?

- More animation types
- Text labelling
- Styled Layer Descriptor & Web Map Context
- ...?

The WORKshop part:

- Install (described on the RIMapper website:
 1. PostgreSQL
 2. PostGIS
 3. PgAdminIII (on Mac, other installers have it included)
 4. Apache Tomcat
- Deploy the RIMapper Java application (RIMapper.war)
- Set up the example database schema
- Try it on <http://localhost:8080/RIMapper/testURL.html>

- If time permist we'll do some editing:
 - changing the animation style settings
 - changing a 'single' style
 - adding a 'chorochromatic' style
 - ...