VTS 3D Map Streaming and Rendering Stack

Ondřej Procházka, CEO Melown Technologies SE
I. Introducing VTS

II. VTS Architecture

III. Getting Started with VTS

+ Q and A
I. Introducing VTS
What VTS is

• An integrated platform for 3D map application development
  – A set of 2D and 3D geospatial data fusion tools
  – A set of server-side components for 3D geospatial data streaming
  – A set of client-side libraries for interactive 3D map rendering

Think Google Earth Enterprise, but on 2017 technology standards
What VTS is not:

• Google Maps / Google Earth
• Cesium JS
• ArcGIS Online, Agency9, SkylineGlobe, etc

but

• interoperability is encouraged and supported whenever possible
• VTS may be used as both an alternative and a complement to these systems
What you can do with VTS:
stream and render photorealistic landscapes at street level detail and at planetary scale
render place and street labels, stylable and multi-resolution
dynamically generate and stream 3D globes based on DEMs and DSMs
create complex GIS integrations
visualize 3D remote sensing data interactively
and have fun writing games!

VTS Racing on Android
Coordinate system agnosticism

webmerc reference frame
Coordinate system agnosticism

melown2015 reference frame
Coordinate system agnosticism

earth-qsc reference frame
More VTS selling points

• JavaScript and C++11 client libraries
• GDAL rasters and OGR vectors with stylesheets
• WMS/WMTS and Mapbox vector tile support
• Open source under BSD 2-clause license
Melown Cloud

• a cloud 3D map development platform operated by Melown Tech atop of VTS
• a point-and-click interface to a subset of VTS functionality
• ideal for smaller projects and less technically savvy users

www.melown.com/cloud
II. VTS Architecture
VTS architecture

- Streaming servers: *mapproxy* and *vtsd*
- Rendering libraries: *vts-browser.js* (JavaScript) and *libvts-browser* (C++)
- Data fusion and management tools: *vts*, generatevrtwo, etc.
- Encoders: vts2vts, vef2vts, slpk2vts, etc.
VTS streaming servers: mapproxy*)

- An HTTP server
- Performs on the fly conversion of non-VTS GIS formats to VTS streaming formats
- (a powerful SRS transforming TMS server)

**Input formats:** GDAL rasters, OGR vectors, Mapbox vector tiles
**Output formats:** surfaces, bound layers, tiled geodata

*) github.com/Melown/vts-mapproxy
VTS streaming servers: vtsd*)

• An HTTP server
• Streams static tilesets as surfaces or free layers
• Translates storage views into streamable VTS map configurations
• Implements TILeARchive – efficient format for 3D tile hierarchy storage

*) github.com/Melown/vts-vtsd
VTS data management tools: vts*

A Swiss army knife of VTS storage management:

$ vts --create mystorage --referenceFrame melown2015
$ vts mystorage --add --tileset local:<path-to-tileset> --top
$ vts mystorage -add http://<mapproxy-surface-url> --top

If mystorage is inside vtsd root, mystorage/mapConfig.json yields a map configuration with both surfaces in proper stacking order.

*) github.com/Melown/vts-tools
VTS rendering libraries: vts-browser-js*

• All encompassing WebGL-based VTS client-side implementation
• Comprehensive API
• Currently about 176 kB gzipped and minified
• Works in all modern browsers
• Rudimentary mobile support

*) github.com/Melown/vts-browser-js
VTS rendering libraries: libvts-browser*)

- a recent C++ client library
- very small footprint, lightweight
- separate from actual rendering layer
- developed and tested on GNU/Linux desktop, awaits ports to other platforms

*) github.com/Melown/vts-vtsbrowser-cpp
III. Getting Started with VTS
Getting started with VTS JavaScript API

*) github.com/Melown/vts-browser-js/wiki

**) https://github.com/Melown/vts-browser-js/wiki/Examples

JS Fiddle live examples**
Setting up a mapconfig

A.) use one of the public URLs available from Melown Tech, or
B.) use Melown Cloud (melown.com/cloud), or
C.) install and configure VTS streaming servers
Getting started with VTS streaming servers

Set up your own data management and streaming: *)

$ apt install vts-backend

*) more details at melown.com/vtstutorials
Getting started with VTS C++ desktop API

For C++ desktop application development:

*)

```
$ apt install libvts-browser-dev vts-browser-glfw
$ vts-browser-glfw <map configuration url>
```

Then use the Qt5 sample application** as a starting point.

*) more details at github.com/Melown/vts-browser-cpp

**) github.com/Melown/vts-browser-cpp/tree/master/browser/src/vts-browser-qt
Sources of VTS information

melown.com/vts
github.com/Melown

Getting involved in VTS development

contact: community@melown.com
or fork us on GitHub ;-

“An Unlikely Road to an Advanced 3D Mapping Open Source Technology”

Today at 13:30 at Waterfront 2