Spatial Data Computations in a Toolkit to Improve Accessibility for Mobile Applications

Janne Kovanen*, Paul Kelly**, Stuart Ferguson**,
L. Tiina Sarjakoski* and Tapani Sarjakoski*

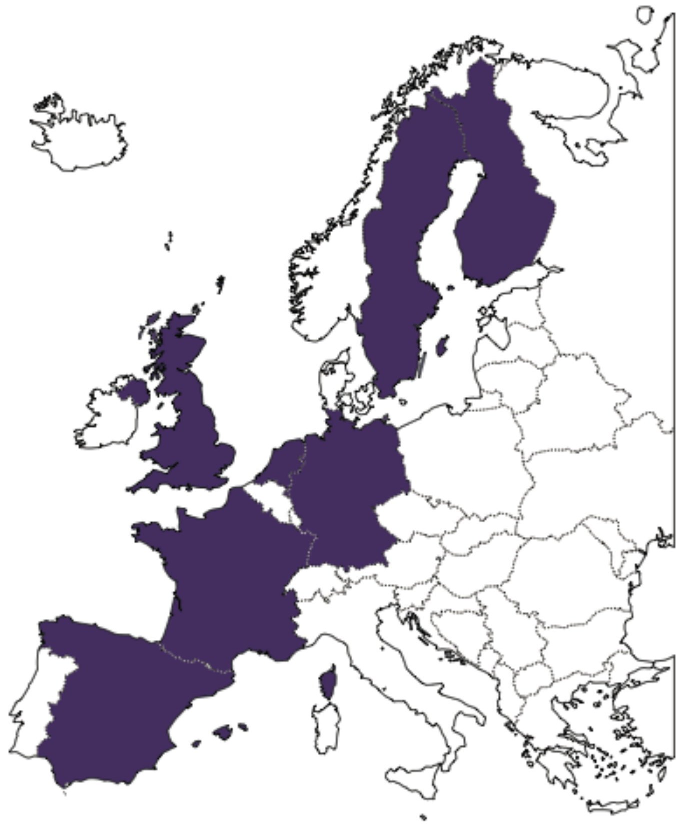
* Finnish Geodetic Institute
Department of Geoinformatics and Cartography

** Queen’s University Belfast
School of Electronics, Electrical Engineering and Computer Science

LBS 2011, Vienna, 21–23 November 2011

HaptiMap

- "Haptic, Audio and Visual Interfaces for Maps and Location Based Services"
- HaptiMap aims at making mobile maps and location based services more accessible by using several senses like touch, hearing and vision
- Receives financial support from the EC in the 7th Framework Programme



The HaptiMap toolkit

- The toolkit provides "tools" for mobile map application developers to enhance the accessibility of their applications
- Tools support several interaction modalities
- Open-source
  - Licensed under an umbrella of OS licenses
  - Subversioning, wiki, mailing list etc. are available
- Cross-platform
  - Android, iPhone (iPad), Windows Mobile, Meego, Symbian, …

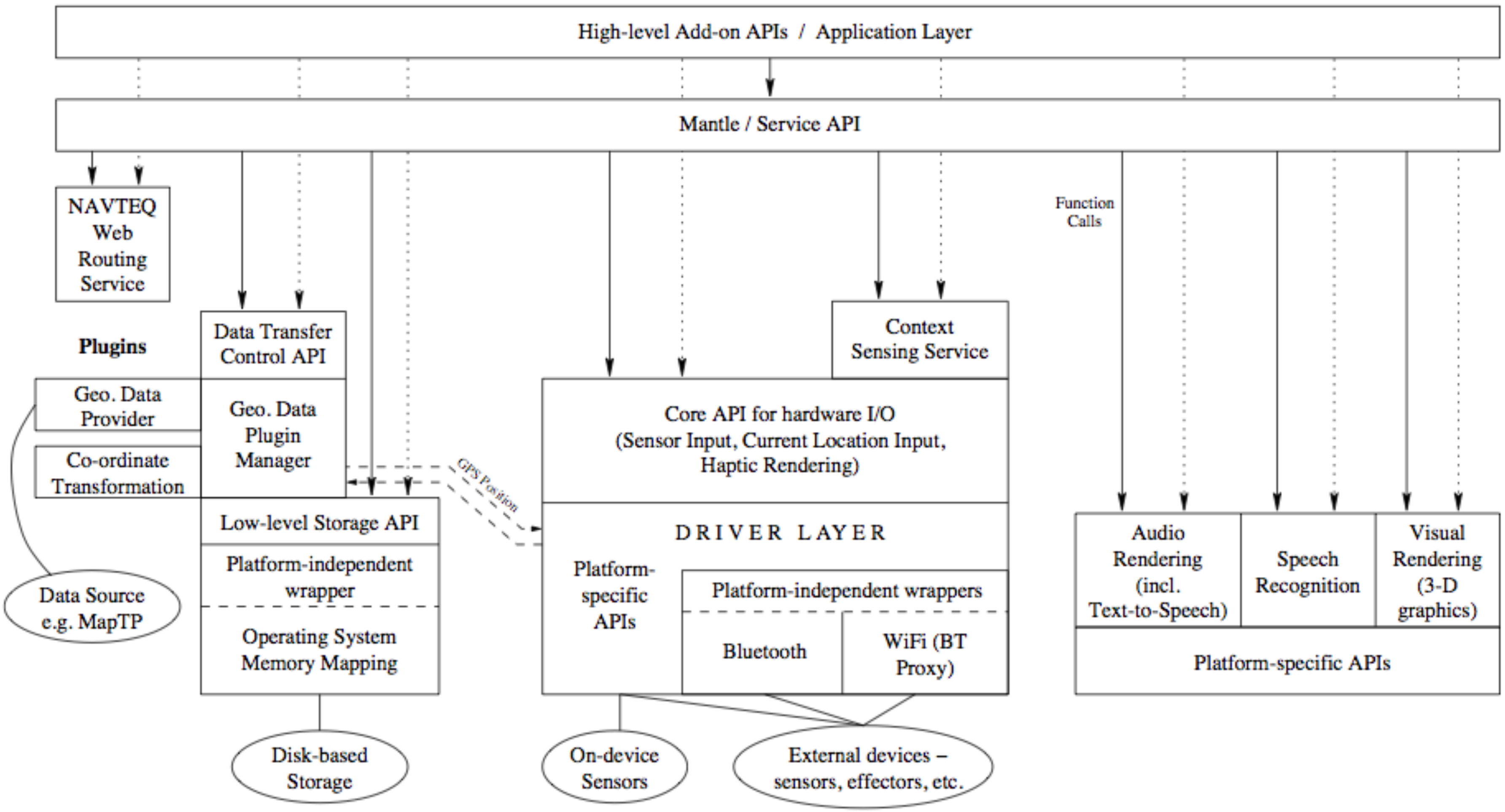
Finnish Geodetic Institute
janne.kovanen@fgi.fi

… The HaptiMap toolkit

- Made “simple” for the end users, such as
  - Human-Computer Interaction developers
  - Developers building on top of existing platforms and having a moderate knowledge of the spatial domain
- Simplicity is a compromise
- Is composed of three principal layers
  - Core, Mantle and Crust
- Plug-ins are a logically separate components
- Is a set of Application Programming Interfaces

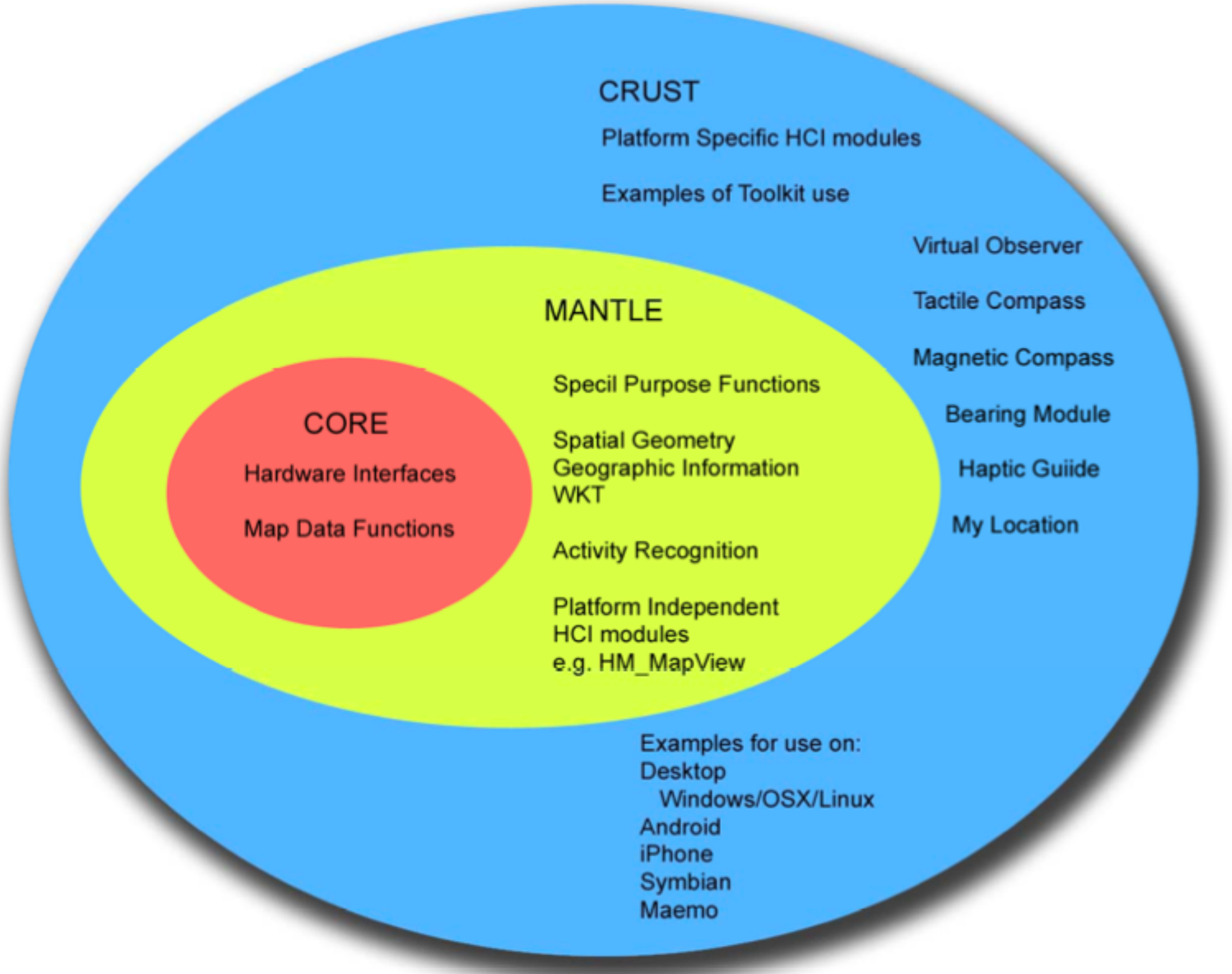
Finnish Geodetic Institute
janne.kovanen@fgi.fi

The architecture



Finnish Geodetic Institute
janne.kovanen@fgi.fi

The architecture



The architecture – Core

- The core contains functions to access both in-built and external sensors
  - Accelerometers, tactile vibrators, speech engines, positioning, digital compass, …
  - Parts of the core are platform-specific!
- The second task is handling and caching of geographic vector data
  - The data is stored in memory-mapped disk files
- Public interfaces for upper layers
- Currently licensed under LGPL

The architecture – Mantle

- Consists of platform-independent human-computer interaction modules
  - Act as building blocks
  - Contain analysis and processing support
- Includes computational geometry functions
  - Supports the HCI modules
  - Data is always read from the internal data storage
- Written in ANSI-C
- Currently licensed under LGPL

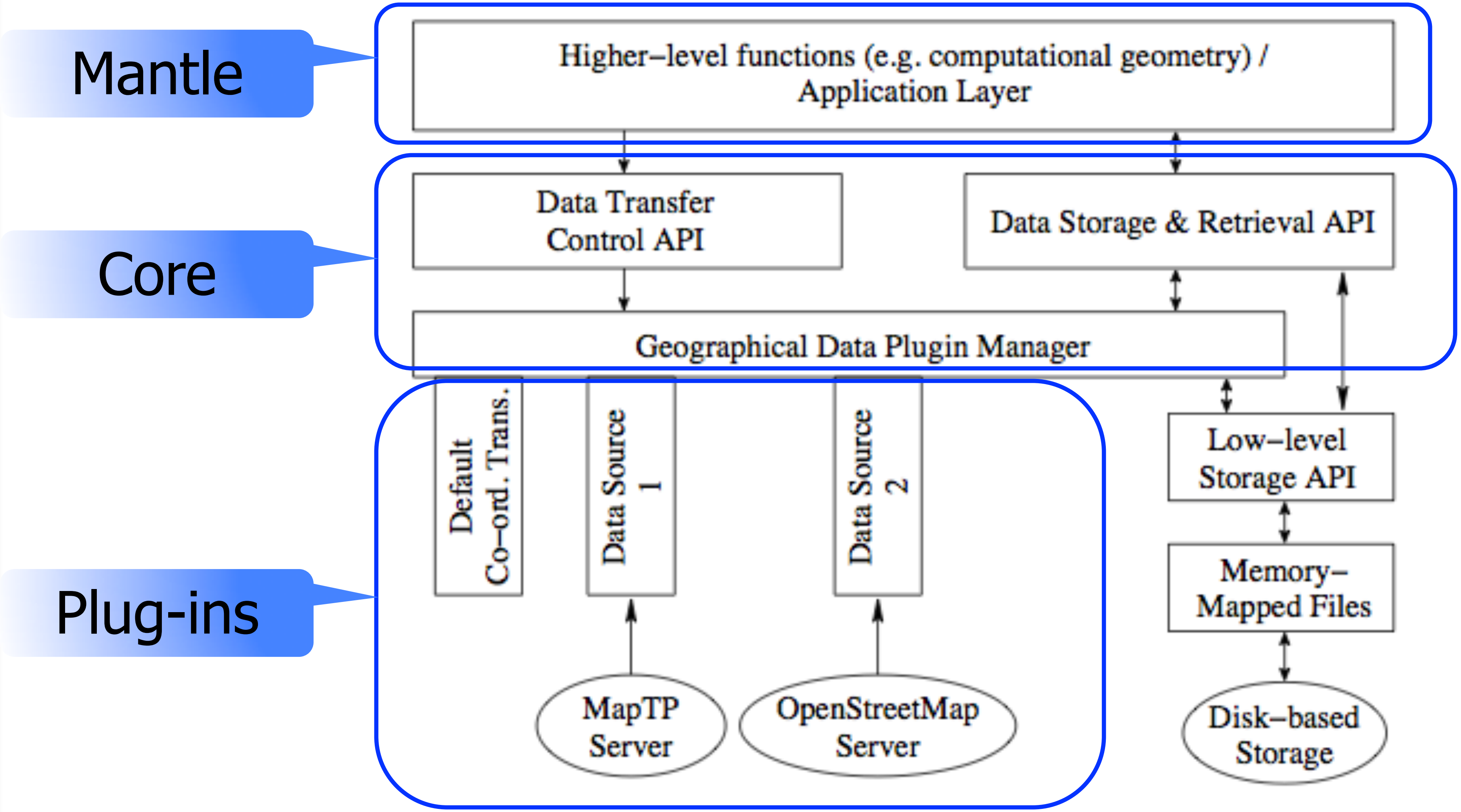
Finnish Geodetic Institute
janne.kovanen@fgi.fi

The architecture – Crust & Plug-ins

- Crust contains platform-specific components
  - Human-Computer Interaction modules
  - Views, view controllers, view activities, fragments, …
  - Examples
- Plug-ins are used for
  - Reading data from external data sources
  - Perform model transformation to the internal model
  - Co-ordinate reference system support
  - The leading plug-in defines the reference system
- Any suitable license may be used

Finnish Geodetic Institute
janne.kovanen@fgi.fi

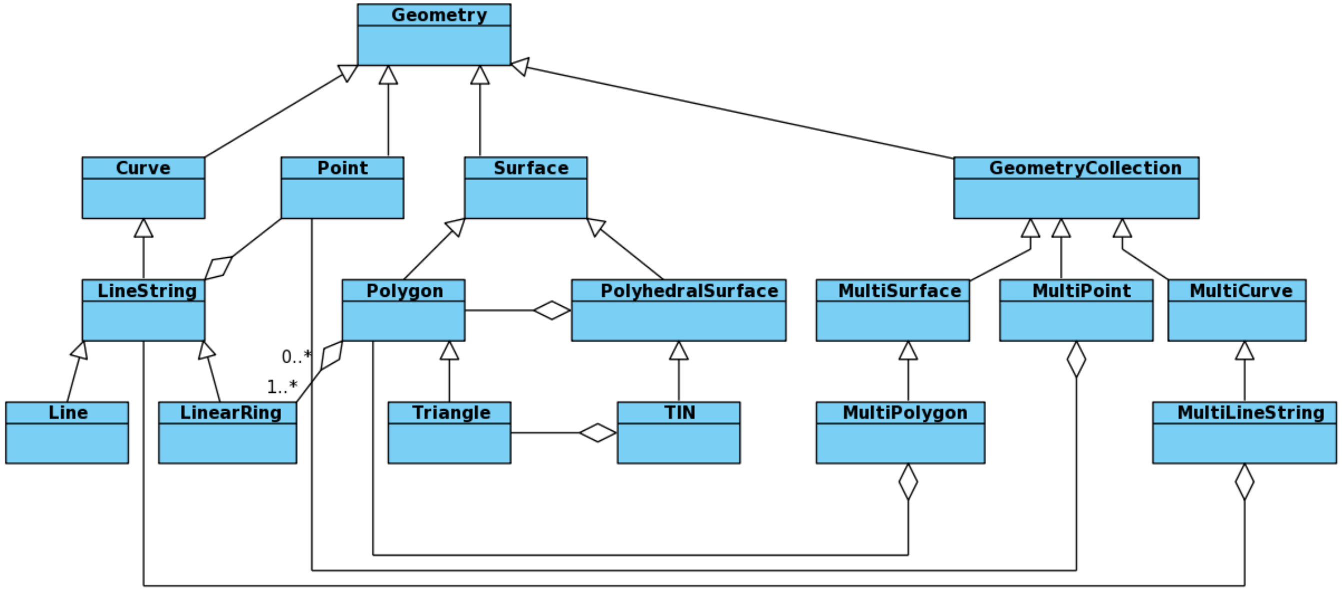
Geographical data loading

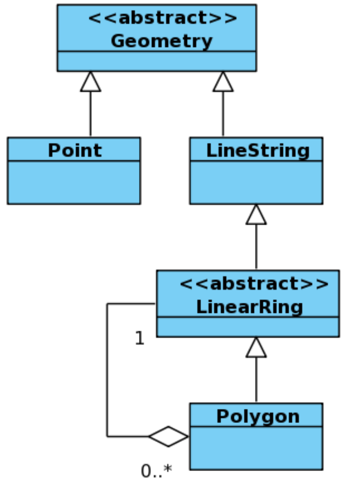


Finnish Geodetic Institute
janne.kovanen@fgi.fi

The geometry model & data types

- Geometries are 2D (+1D) points and linestrings
- Polygons are defined by the internal data storage
- Co-ordinates are stored as 32-bit integers
- The internal Unit of Measure is centimeters





Finnish Geodetic Institute
janne.kovanen@fgi.fi

Computational geometry functions

```
/* Metric methods */
HM_RESULT hm_geom_area(hm_t *hm, int lid, double *area);
HM_RESULT hm_geom_bearing(hm_t *hm, int pid1,int pid2, double *angle);
HM_RESULT hm_geom_distance(hm_t *hm, enum HM_GEOMETRY_TYPE gtype1, int fid1,
          enum HM_GEOMETRY_TYPE gtype2, int fid2,double *dist);
HM_RESULT hm_geom_distance_hausdorff(hm_t *hm,enum HM_GEOMETRY_TYPE gtype1,
          int fid1,enum HM_GEOMETRY_TYPE gtype2, int fid2,double *dist);
HM_RESULT hm_geom_length(hm_t *hm, int lid, double *length);
/* Spatial predicates */
HM_RESULT hm_geom_contains(hm_t *hm, int polyfid,
          enum HM_GEOMETRY_TYPE gtype, int fid, int *r);
HM_RESULT hm_geom_within(hm_t *hm, int fid,int polyfid,
          enum HM_GEOMETRY_TYPE gtype, int *r);
HM_RESULT hm_geom_intersects(hm_t *hm,enum HM_GEOMETRY_TYPE gtype1,
          int fid1, enum HM_GEOMETRY_TYPE gtype2, int fid2,int *r);
/* Overlay methods */
HM_RESULT hm_geom_intersection(hm_t *hm,
         enum HM_GEOMETRY_TYPE gtype1, int fid1,
         enum HM_GEOMETRY_TYPE gtype2, int fid2,
         enum HM_GEOMETRY_TYPE *r_type, int *r);
/* Buffering */
HM_RESULT hm_geom_buffer(hm_t *hm, int fid, enum HM_GEOMETRY_TYPE gtype,
         double buffer_width, int *r);
/* Generalisation etc */
HM_RESULT hm_geom_simplify(hm_t *hm, int lid,double tolerance, int *r);
HM_RESULT hm_geom_centroid(hm_t *hm, int lid, int *r);
HM_RESULT hm_geom_interior_point(hm_t *hm, int lid, int *r);
HM_RESULT hm_geom_convex_hull(hm_t *hm, int lid, int *r);
HM_RESULT hm_geom_mbr(hm_t *hm, int lid, int *mbr_id);
HM_RESULT hm_geom_ray_intersection(hm_t *hm, int lid, int pid,
          double angle, double *distance);
```


Finnish Geodetic Institute
janne.kovanen@fgi.fi

Performance comparison

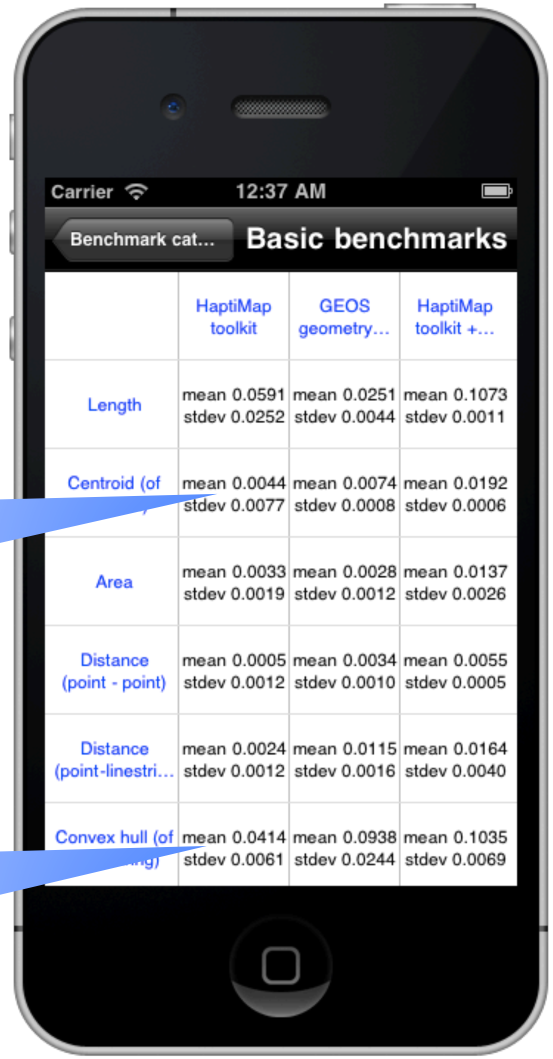
- To validate our approach we performed a performance comparison between solutions
- We applied both unit testing and benchmarks
- The benchmarks were run on the iPhone & iPad
- Three different cases were benchmarked
  - Toolkit – Data already stored in the internal storage
  - GEOS geometry engine (v. 3.2.2) – Data conversion was performed before running the benchmarking
  - Wrapping GEOS – Data in the internal storage was converted during benchmarking from the internal data model to data model of GEOS

Performance comparison results

Results of the first trial versus final (stdev in brackets)

0,058 vs. 0,004
(0,520 vs. 0,008)

0,522 vs. 0,041
(1,268 vs. 0,006)



… Performance comparison results

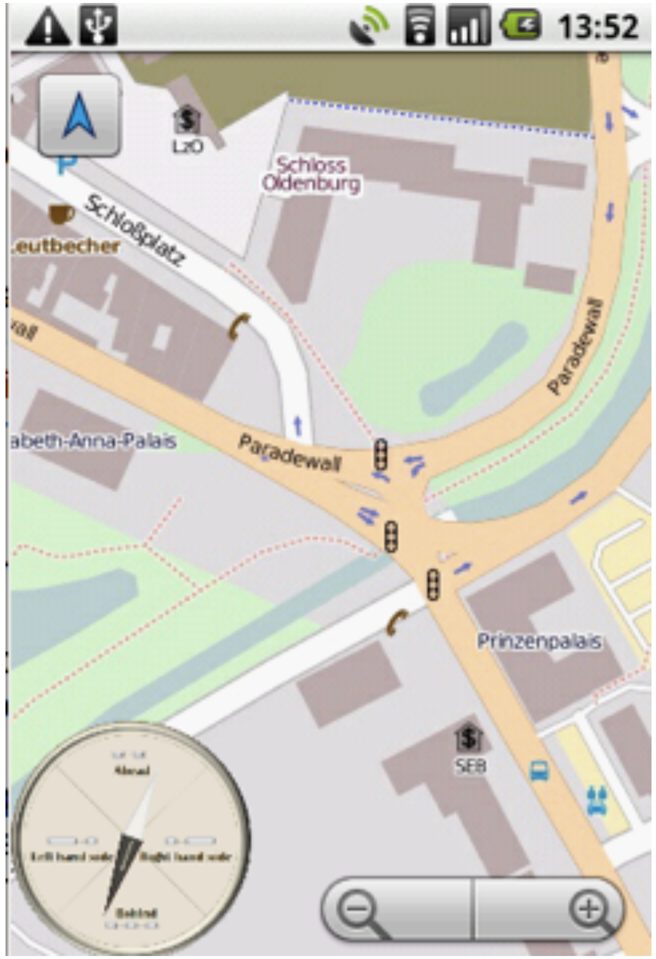
- The internal data storage size increment should be modifiable
- The benchmarking proved the module to be in general faster compared to GEOS
- The solution was tested to be 2-20 times faster than wrapping GEOS functions!

	HaptiMap toolkit	GEOS geometry…	HaptiMap toolkit +…
Length	mean 0.2709 stdev 0.0894	mean 0.1156 stdev 0.0075	mean 1.3575 stdev 0.0174
Centroid (of polygon)	mean 0.0389 stdev 0.0064	mean 0.0600 stdev 0.0175	mean 0.2009 stdev 0.0031
Area	mean 0.0337 stdev 0.0060	mean 0.0178 stdev 0.0052	mean 0.1551 stdev 0.0028
Distance (point - point)	mean 0.0031 stdev 0.0007	mean 0.0315 stdev 0.0026	mean 0.0559 stdev 0.0025
Distance (point-linestri…	mean 0.0119 stdev 0.0018	mean 0.0954 stdev 0.0069	mean 0.1347 stdev 0.0099
Convex hull (of a linestring)	mean 0.2113 stdev 0.0723	mean 0.4453 stdev 0.0250	mean 0.5973 stdev 0.0255

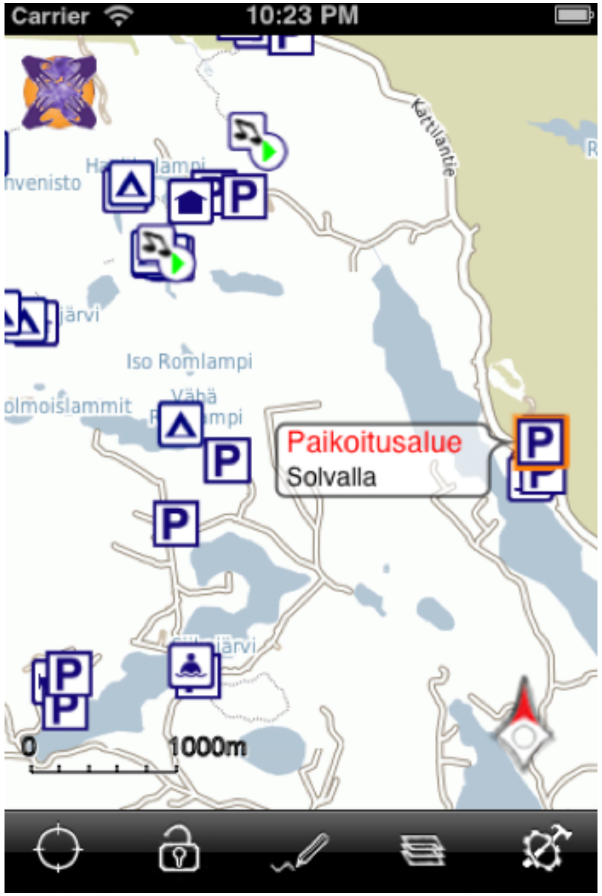
Figure: Results on the iPad 2

Conclusions

- The HaptiMap toolkit may be used to advance accessibility
- Our approach of implementing own optimized computational geometry handling is
  - Significantly faster in comparison to alternative solutions
  - Allows taking into account specific HCI needs/requirements
  - May be extended by wrapping complex functions



Finnish Geodetic Institute
janne.kovanen@fgi.fi



Thank you!

HaptiMap

www.haptimap.org

Finnish Geodetic Institute
janne.kovanen@fgi.fi