Integrating Mashups and Open Source Web-GIS

Pirmin Kalberer
Sourcepole AG, Bad Ragaz, Switzerland
www.sourcepole.ch
Sourcepole

- Linux based software development
- GIS-Knoppix: first GIS live CD
- UMN Mapserver
  - Initial Ruby bindings
  - SDE raster support
- OGR / GDAL
  - Interlis driver
  - Schema support for PostGIS driver
Agenda

- Mashups versus GIS
- Combined architecture
- geopole.org tour
- Used OSS components
- Scalability & performance
Mashups versus GIS

- **Map Mashups**
  - Application data displayed in a map (e.g. Google Maps)
  - Javascript API's -> client side
  - application-centric

- **Classic Web-GIS**
  - Application- and base map data in one database
  - Server-side map creation
  - map-centric
Mashup Architecture

**Clients**
- Browser
- Javascript viewer

**Web server**
- Custom application

**Javascript-API**
- google/yahoo/etc.
Typical Web-GIS Architecture

Clients

Browser

Server

Apache/PHP

viewer & custom application

geodata (DB, Shape, TIFF)
Drawbacks

- Mashups
  - Slow processing for high number of features
  - API dependency

- Typical Web GIS
  - Own map data needed
  - GIS knowledge
  - Integration of custom application in map viewer
"Web 2.0 GIS-Architecture"
Issues to address

- Client-side combination of OGC services and Javascript API's
- Mercator projection of Google Maps
- Support for multiple base map sources
- Support for local geodata
- Integration of map viewer into custom application
geopole.org tour 1/5

- Place search
- Multiple layer overlays
- GeoRSS overlay
Choice of base map
>1000 features
KML- & WMS publication
geopole.org tour 3/5

- Mouse-over feature info
- Multiple features at one position
- OpenID login
- Drag&Drop import from spreadsheet
- Custom columns
geopole.org tour 5/5

- Feature info with HTML elements
- Custom table editor
- Edit/Create/Delete rows
Commercial add-ons

- File upload
- GIS data import (Shapefiles, etc.)
- Dedicated PostgreSQL schema
- PostGIS-Administration
- Non-public maps
- Customized embedding
Architecture

Internet

Proxy server

http

http

Fastcgi

App. server

App. server

WMS server

WMS server

DB server

DB server
Used OSS components

- Operating system: Linux with VServer
- Database: PostgreSQL with PostGIS
- Application framework: Ruby on Rails
- Map viewer: OpenLayers
- WMS server: UMN Mapserver
- HTTP server/proxy: Mongrel/nginx
Ruby on Rails

- Framework for database-backed web application
- Model-View-Control pattern
- Ruby language
- Built-in Ajax support
- GIS libraries:
  - GeoRuby: PostGIS access
  - Graticule, GeoKit: geocoding
  - Ruby bindings for GDAL, Mapserver, etc.
Scalability/Performance

- Load-balancing
  - Application webserver
  - UMN Mapserver (Fast-CGI)
  - Tile server
- Caching
  - map tiles
  - REST protocols
Web 2.0 GIS platform proposals:
- Combination of OGC protocols and Javascript APIs
- State-of-the-art application frameworks
- Use of basic internet protocols gives interoperability and scalability
- Avoiding service vendor lock-in
- Intuitive map interaction
Links

- http://www.geopole.org/
- http://www.rubyonrails.org/
- http://www.openlayers.org/
- http://postgis.refractions.net/
- http://mapserver.gis.umn.edu/
- http://nginx.net/
Questions?

*Live demo*

Thanks!

Pirmin Kalberer

<pka at sourcepole.ch>