DSpace

A new way of handling geolocation based information

iggy & chrono

2013-06-13

Question

”If life were a just computer game with awesome sensory input, which standard game features would be missing in our interface in order to play it well, especially when playing in groups?”

Interface View

Image Street - boring

Interface View

Image Street - boring Image Street - DSpaced

Talk-Contents

Who we are?

What do we want?

What do we have?

- Basemaps
- Overlays
- DSpace Client

What do we need?

Who we are?

chron0 chrono

JavaScript

elf-pavlik elf Pavlik

https://gitorious.org/~elf-pavlik perpetual-tripper@wwelves.org JavaScript, Ruby, CoffeeScript

yggi Sebastian Steuer

iggy@yggi.de Python, JavaScript, Shell

alice-wl alice

PHP, Shell, Python

nilclass Niklas Cathor

Hamburg JavaScript, Ruby, Shell

What do we want?

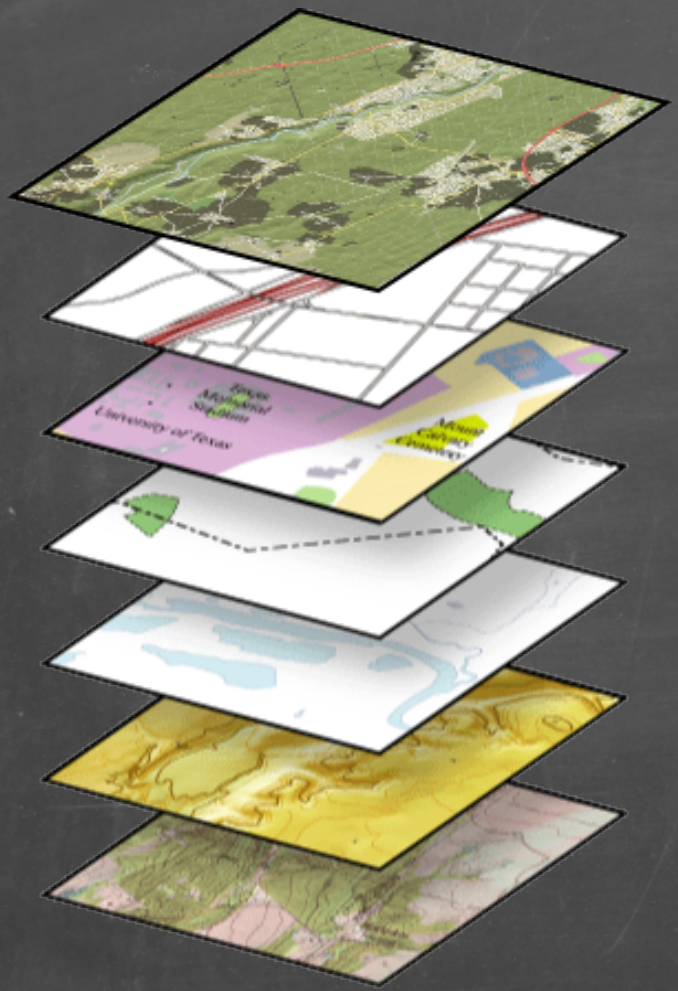
Staging our wants

Something like a standard

Increase the likelyhood and efficiency of adding/sharing information by introducing a standardized framework like the W3c in 1993.

- Federation
- Free
- Open Source
- Lose Bindings
- Modular Extensions

Open Basemaps



Basemap Tile Assembly:

- Roads (OSM)
- Land Usage (OSM)
- Boundaries (OSM)
- Hydrography (OSM/External)
- Topography (NASA/DLR SRTM)
- Land Imagery (NASA Blue Marble)

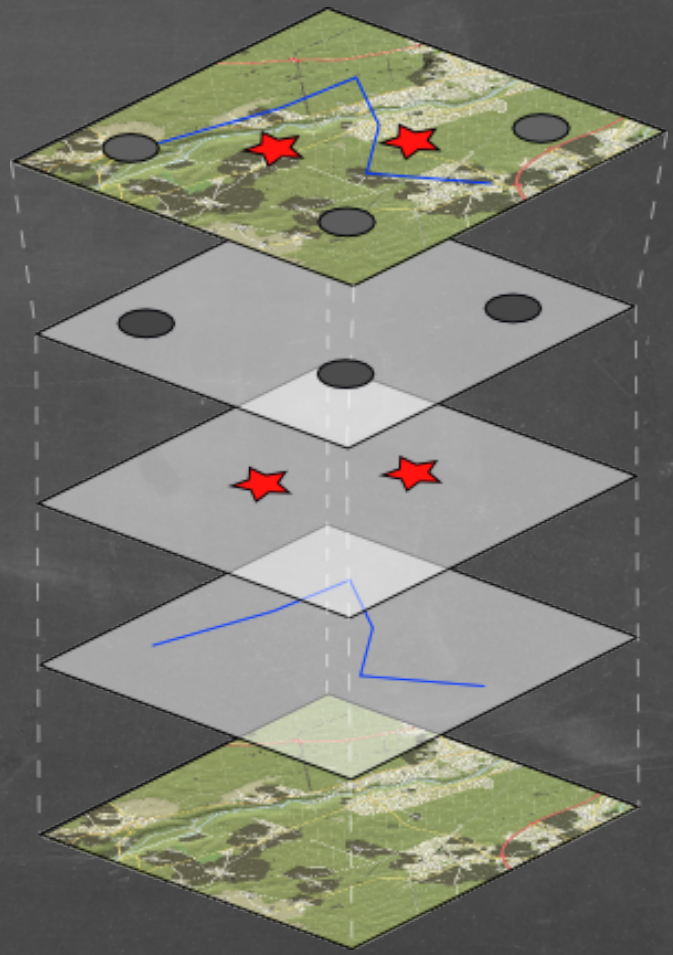
Basemap Properties

- Static/Longterm data retention & validity
- General interest
- Only one basemap is visible at a time
- Composition based on region/application
- Updates are resource intensive
- Server based rendering

Overlays

Basemap + Overlay Assembly:

- POIs
- Real-Time Location tracking
- Waypoints on a route (Navigation)
- User/Overlay selected Basemap



Overlay Properties

- Collections of things at locations
- Public or private
- Can be very dynamic (Real-Time)
- Many can be overlayed at a time
- User-generated and -updated
- Very fast & cheap updates
- Suited for browser/client rendering

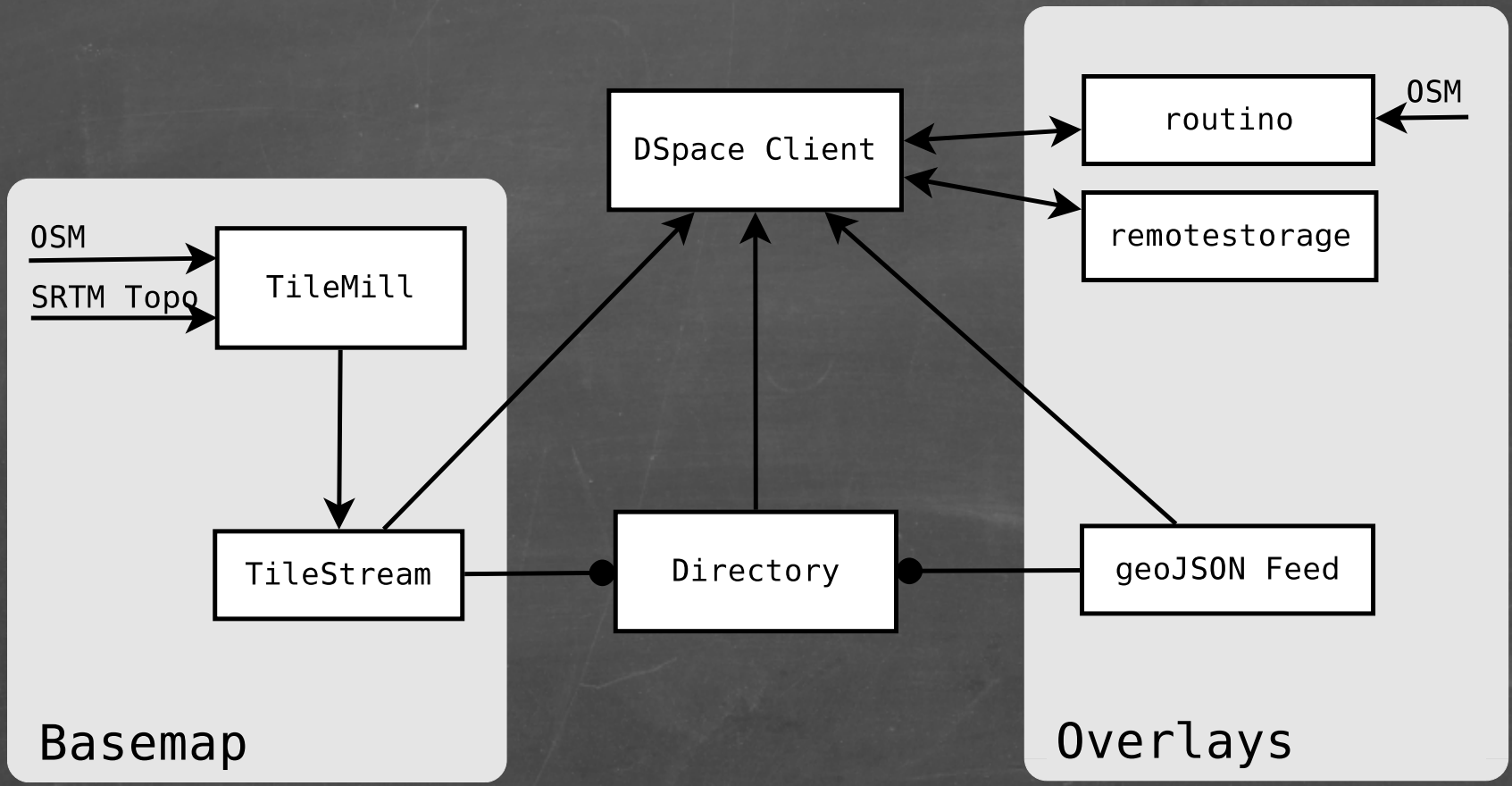
Overlay ideas

- Urban Management
  - Emergency Response Management (First Responder Setup)
  - Hitchhiking (linking drivers/hikers in a sector - hitchwiki.org)
  - Real-Time public transportation information
  - Real-Time risk distribution
- Resource Management
  - Food Mapping/Sharing (mundraub/foodshare.org)
  - Dumpster Diving (trashwiki.org)
  - Fleet Management
  - Open Access Mapping (openwifimap.net)
- Organizing Events
  - Public congress/camp Overlay for visitors
  - Private engel Overlays for orga

Even more Overlay ideas

- Realtime Semantic Mapping Heat mapping twitter hashtags (i.e. heatmap #earthquake to find current EQ reports and positions)
- Private group overlays for the area of activity (i.e. MuCCC)
- Drone GCS Interfacing Localization and interactive Mission/WayPoint Management
- Entertainment Geocaching, AR-MMORPGs, AR-MMO-Strategy-Games
- Open Network Access Mapping Access Points (http://openwifimap.net)-http://map.pberg.freifunk.net/ + ham-radio repeater information
- ADS-B Airplane Mapping Overlay
- Use your imagination

Architecture Overview



What do we have?

Staging our haves

What do we have?

TileMill

BACKBONE.JS

NASA

DLR

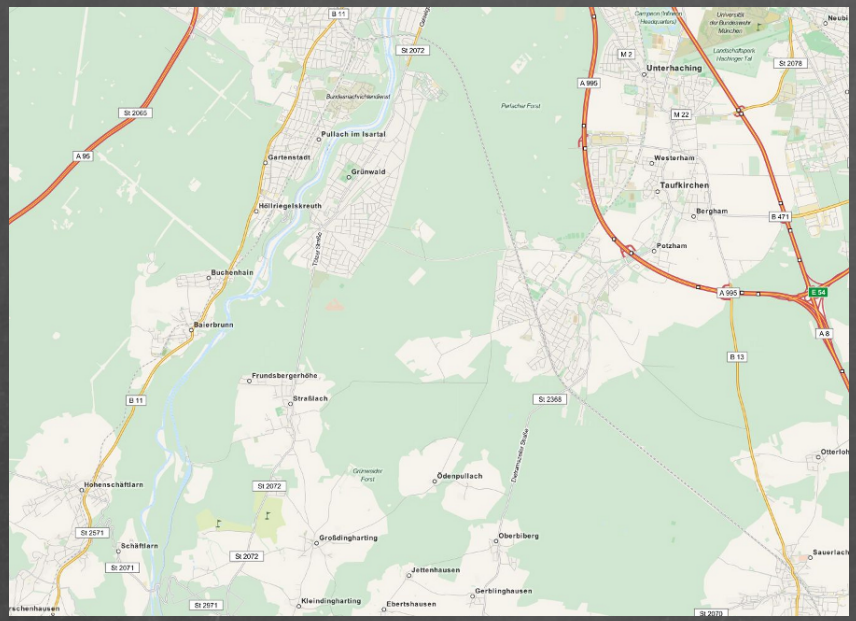
REMOTE STORAGE

CGIAR CSI
Consortium for Spatial Information

And many more projects by countless people who provide open-source software and data to merge and assemble into DSpace

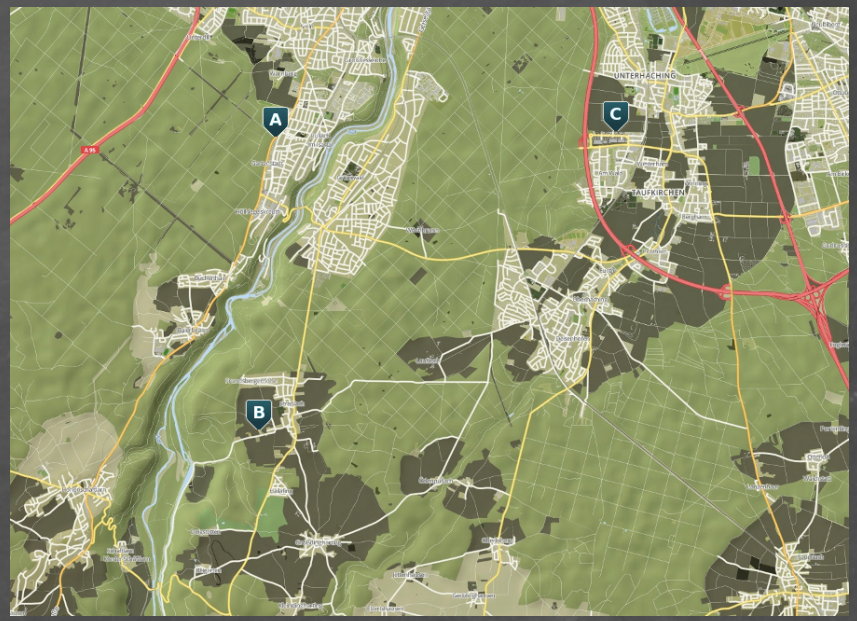
Why not use OpenStreetMap Map-Servers?

Not everything is in OSM:



- Topography
- Aerial Imagery
- All non-STREETmap data

Not everything belongs in OSM:



- Tracking
- Personal/Private POIs
- Dynamic/Real-Time data

Map Forge

- NodeJS
- Mapnik + TileMill + OSM-Bright
- PostgreSQL + PostGIS + imposm
- OpenStreepMap data
- NASA/CGIAR SRTM SIR-C-Band V41 90m Topo data
- DLR SRTM X-Band SAR 25m Topo data (+/- 60 coverage)
- TerraSAR-X/TanDEM-X data (Future)

```
imposm -U gisuser -d gis -m \
/tmp/osm-bright/imposm-mapping.py --overwrite-cache --read --write --optimize \
--deploy-production-tables planet-latest.osm.pbf
```

Map Forge Screenshot

Include picture of Map Forge in action

Map Delivery

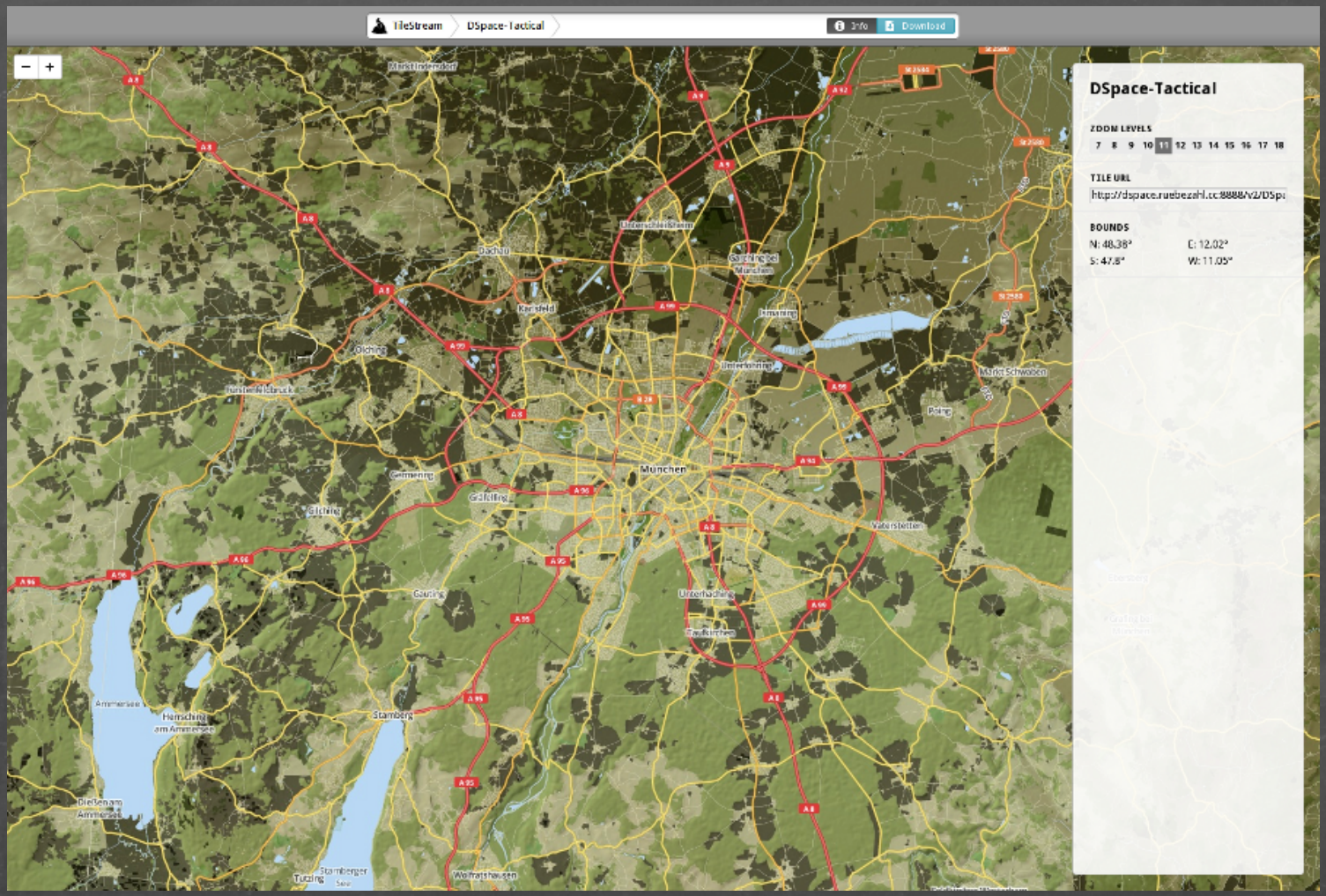
TileStream is also written in NodeJS and very easy to set up:

- Copy maps from TileMill to your TileStream map folder
- Start TileStream*

```
$ cp /tmp/my_map_from_tilemill.mbtiles /home/tilestream/maps/
$ ./index.js --host my.fully.qualified.domainname --tiles=/home/tilestream/maps/
Started [Server Tile].
Started [Server Core:8888].
```

*Obviously, this should be running as a service :)

Live DEMO of the TileStream WebUI



Overlays - GeoJSON

```
"type": "FeatureCollection",
  "name": "Pools",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [11.62876, 48.15471]
      },
      "properties": {
        "type": "Pool",
        "indoor": "yes",
        "outdoor": "no",
        "title": "Cosimabad",
        "address": "Cosimastr. 5",
        "zip_code": "80925",
        "website": "http://www.swm.de/hallenbaeder/cosima-wellenbad.html"
      }
    },
    ...
```

Overlays

GeoJSON Feed

- simple http readonly list

SpaceAPI

- global list of hackerspaces, with live state info

remotestorage.io

- unbound private/public data storage

Navigation

- Routino
- OpenStreetMap dump as basis (just like TileMill)

Client

Technologies:

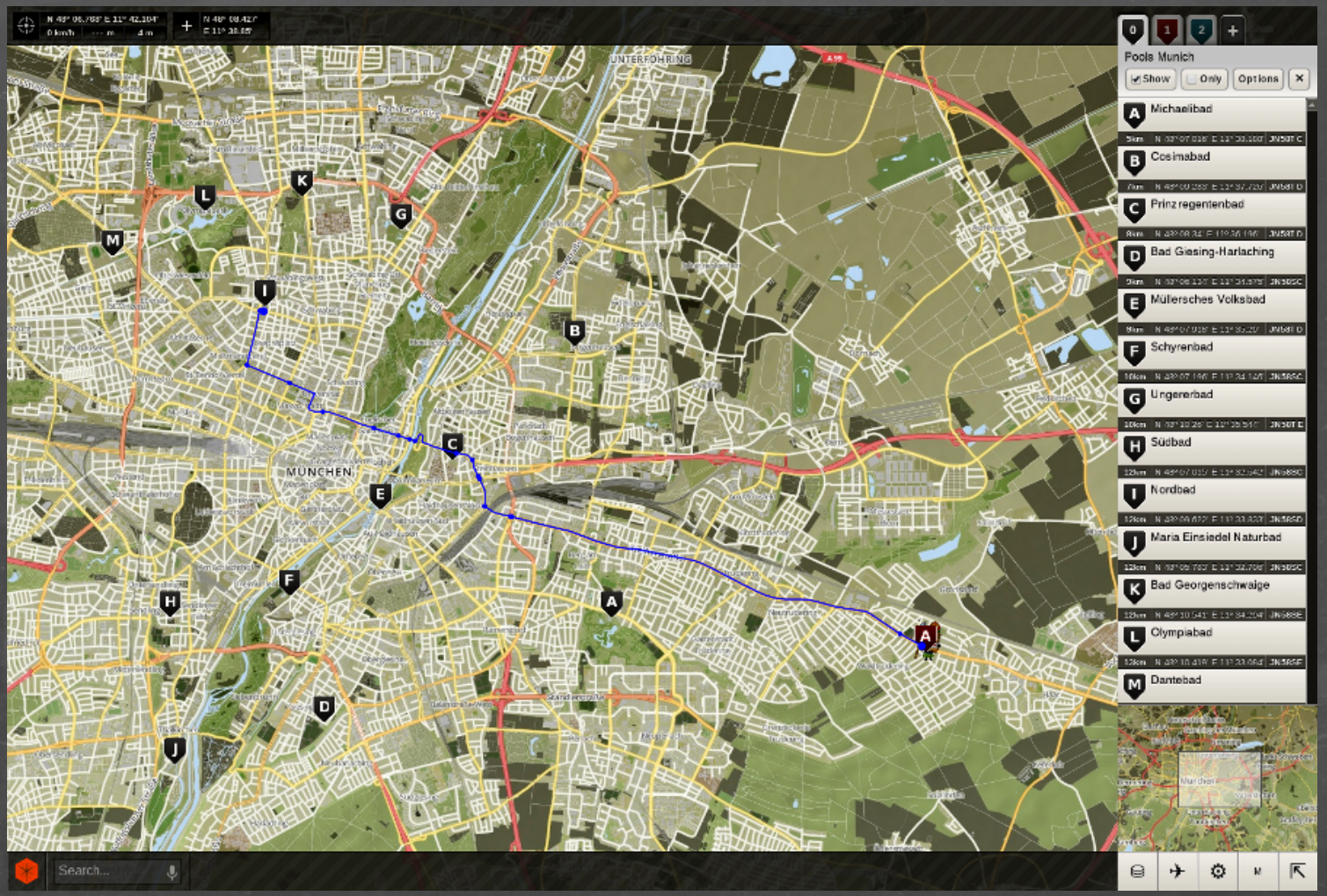
- HTML5/JavaScript
- ModestMaps

Focus on:

- As lightweight as possible
- Powerful Plugin-API
- Mobile Readiness/Integration

Assembled, built and packaged with NodeJS.

DSpace Client Live DEMO



NPM Package Overview

```
almond@0.2.4
backbone@0.9.10
bean@1.0.3
bonzo@1.3.5
csso@1.3.7
domready@0.2.11
+ ender@1.1.0
+ ender-js@0.4.4-1 extraneous
handlebars@1.0.8
  optimist@0.3.5
    wordwrap@0.0.2
  uglify-js@1.2.6
morpheus@0.6.7
qwery@3.4.1
requirejs@2.1.4
reqwest@0.6.4
underscore@1.4.4
```

Comfortable Build Process

```
# make init
Rebuilding GIT submodules... [OK]
Building local deps... [OK]
Building AMD Deps... [OK]
Assembling JS components... [OK]

# make deps
Building Ender... [OK]
Building local deps... [OK]
Building AMD Deps... [OK]
Assembling JS components... [OK]

# make build
Building Ender... [OK]
Building local deps... [OK]
Building AMD Deps... [OK]
Assembling JS components... [OK]
Cleaning up build/... [OK]
Build & minify dspace-client.js... [OK]
Copying Assets... [OK]
Copying Plugin Assets... [OK]
Merging and compressing dspace-client.css... [OK]
>>> Client build complete
```

Ops friendly deploy

Easy and structured deployment leaves flexibility for different setups and simple rewrites.

```
+ assets
    + css
    - dspace-client.js
    + icons
    + images
index.html
+ plugins
  + remotestorage
    + assets
      - remoteStorageIcon.svg
      - style.css
  + search
    + assets
```

What do we need?

Staging our needs

Directory Server

Serves a list of basemap and overlay feeds

- Federated
- Searchable
- Ranked
- Geobounded
- Tagged

Attributes of an Directory Entry:

- url
- type (overlay or basemap)
- topleft, bottomright
- name
- description
- tags
- rank

Client

- more overlay functionality
  - polygons
  - (translated) images
  - 3D
- mobile integration
  - iOS
  - android
  - glass!
- Overlay browser

Collaboration

People ...

- ... forging and serving basemaps for their area
- ... exposing existing geodata as dspace overlay feeds
- ... helping with docs, bugs, issues, features (mostly on the client for now)

THEEND

Thanks for your attention.
[QR + URL]
Discussion