

# 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?

## 1. Who we are?

# Who we are?



**chron0 chrono**

JavaScript



**elf-pavlik elf Pavlik**

<https://gitorious.org/~elf-pavlik> [perpetual-tripper@wwelves.org](mailto:perpetual-tripper@wwelves.org) JavaScript, Ruby, CoffeeScript



**yaggi Sebastian Steuer**

[iggy@yaggi.de](mailto:iggy@yaggi.de) Python, JavaScript, Shell



**alice-wl alice**

PHP, Shell, Python



**nilclass Niklas Cathor**

Hamburg JavaScript, Ruby, Shell

2. What do we want?

## What do we want?

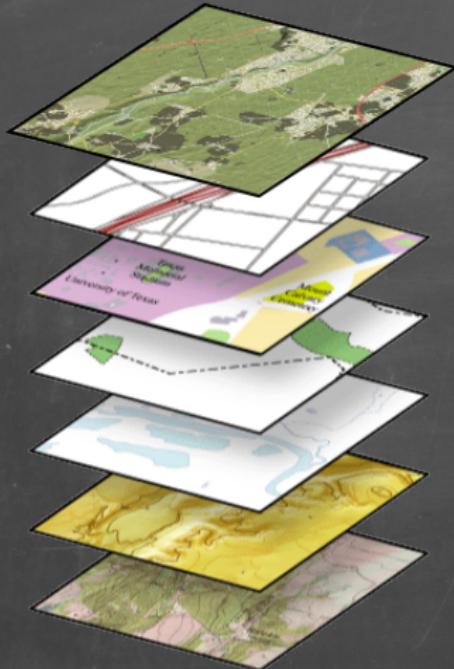
Staging our wants

## Something like a standard

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

- ▶ Federation
- ▶ Free
- ▶ Open Source
- ▶ Loose 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

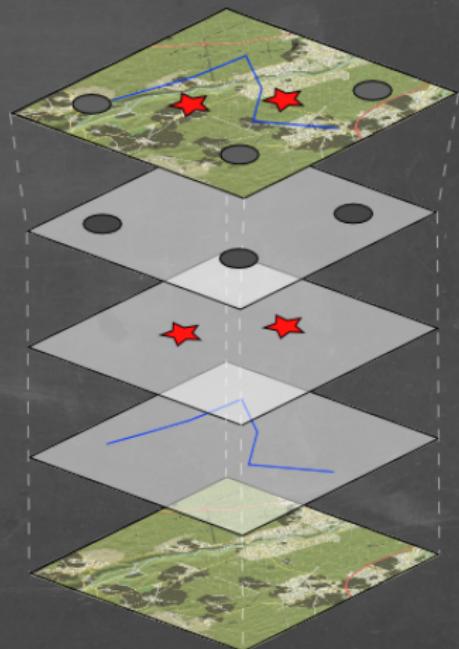


- ▶ Static/Longterm data retention & validity
- ▶ General interest
- ▶ Only one basemap is visible at a time
- ▶ Composition based on region/application
- ▶ Updates are resource intensive
- ▶ Server based rendering

## Overlays

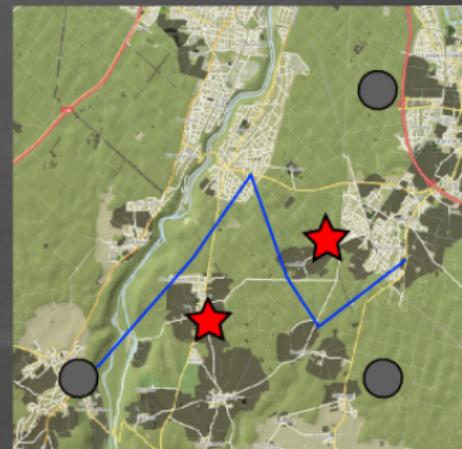
Basemap + Overlay Assembly:

- ▶ 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



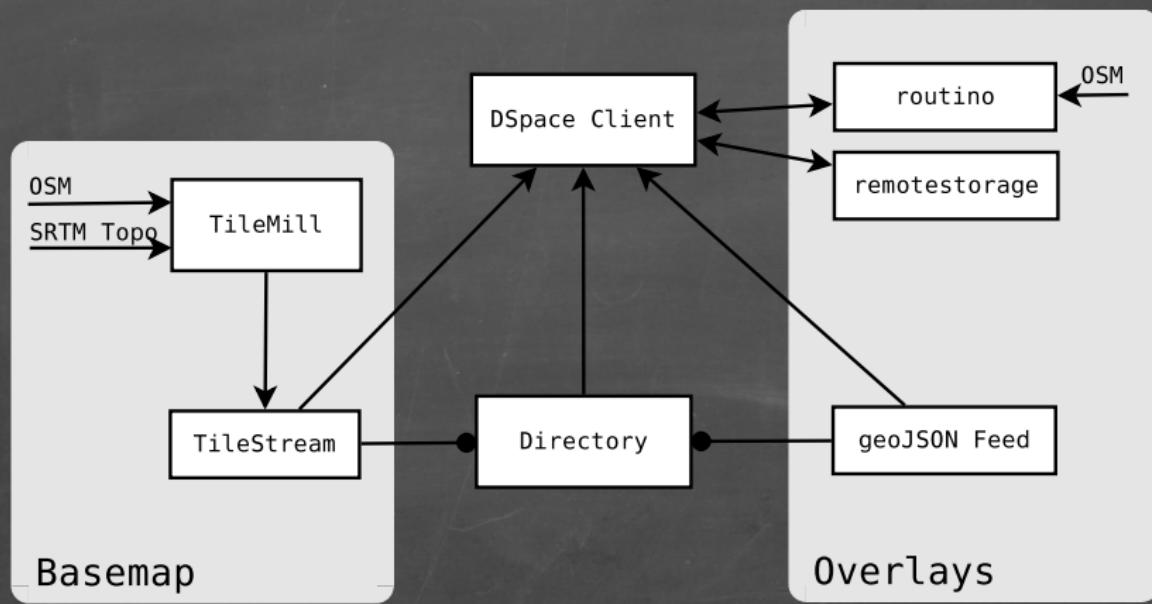
## Basic Overlay applications

- ▶ Where are my friends?
  - ▶ Shared list of realtime geolocations
- ▶ Where are bars my friends like?
  - ▶ Shared list of static geolocations
- ▶ How far is it and how do I get there?
  - ▶ Navigation
- ▶ ¿Wheresss myy hoooome? (show display to taxi driver)
  - ▶ Private list

## Advanced Overlay applications

- ▶ Urban Management
  - ▶ Emergency Response Management (First Responder Setup)
  - ▶ Hitchhiking (linking drivers/hikers in a sector - hitchwiki.org)
  - ▶ Real-Time public transportation information
  - ▶ Realtime Semantic Mapping
- ▶ Resource Management
  - ▶ Food Mapping/Sharing (mundraub/foodshare.org)
  - ▶ Fleet Management
- ▶ Organizing Events
  - ▶ Public congress/camp Overlay for visitors
  - ▶ Private events Overlays for orga
- ▶ Entertainment
  - ▶ Geocaching
  - ▶ AR-MMORPGs, AR-MMO-Strategy-Games

## Architecture Overview



### 3. What do we have?

## What do we have?

Staging our haves

## What do we have?



TileMill



BACKBONE.JS



DLR



REMOTE  
STORAGE



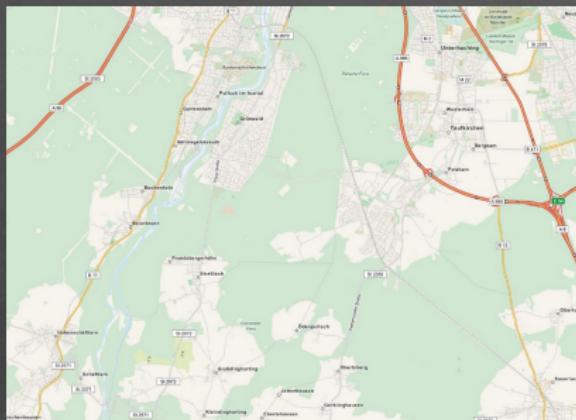
CGIAR CSI  
Consortium for Spatial Information



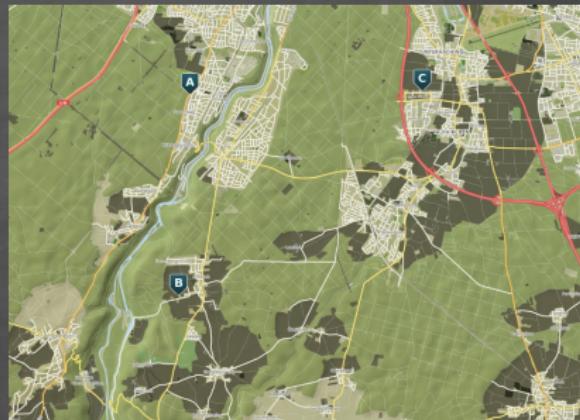
And many more projects by countless people who provide open-source software and data to merge and assemble into DSpace

# Why not use OpenStreetMap Map-Servers?

Not everything is in OSM:



Not everything belongs in OSM:



- ▶ Topography
- ▶ Aerial Imagery
- ▶ All non-STREETmap data

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

# Map Forge

- ▶ NodeJS
- ▶ Mapnik + TileMill + OSM-Bright
- ▶ PostgreSQL + PostGIS + imposm
- ▶ OpenStreetMap 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

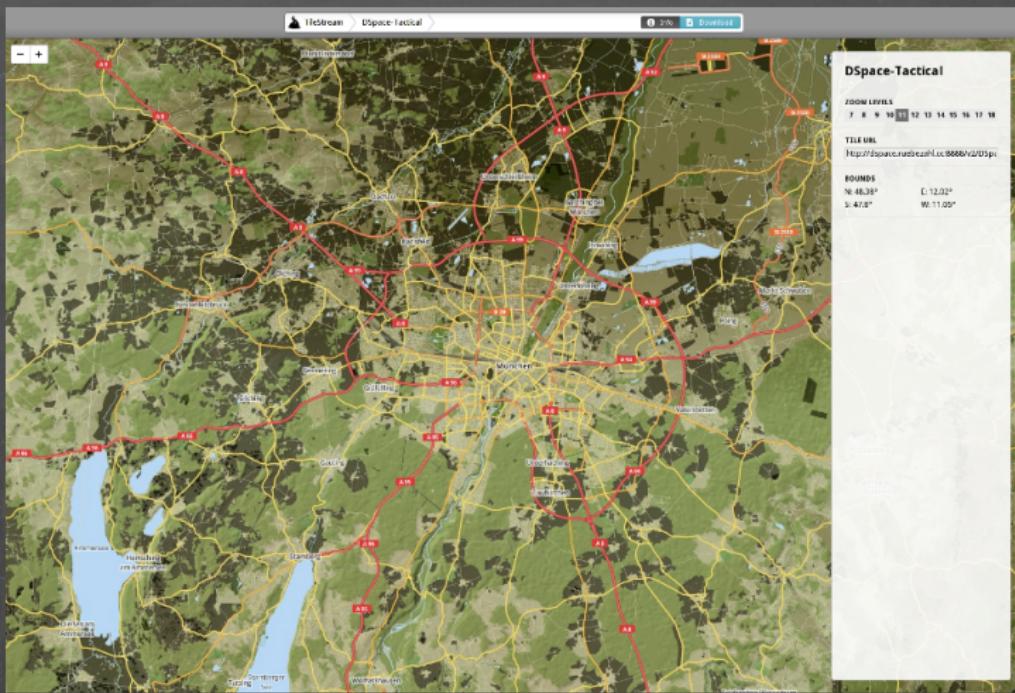
TileStream is also written in NodeJS and very easy to set up:

- ▶ 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

```
1 {
2   "type": "FeatureCollection",
3   "name": "Pools",
4   "features": [
5     {
6       "type": "Feature",
7       "geometry": {
8         "type": "Point",
9         "coordinates": [11.62876, 48.15471]
10      },
11      "properties": {
12        "type": "Pool",
13        "indoor": "yes",
14        "outdoor": "no",
15        "title": "Cosimabad",
16        "address": "Cosimastr. 5",
17        "zip_code": "80925",
18        "website": "http://www.swm.de/hallenbaeder/cosima-wellenbad.html"
19      }
20    },
21    [...]
```

# 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:

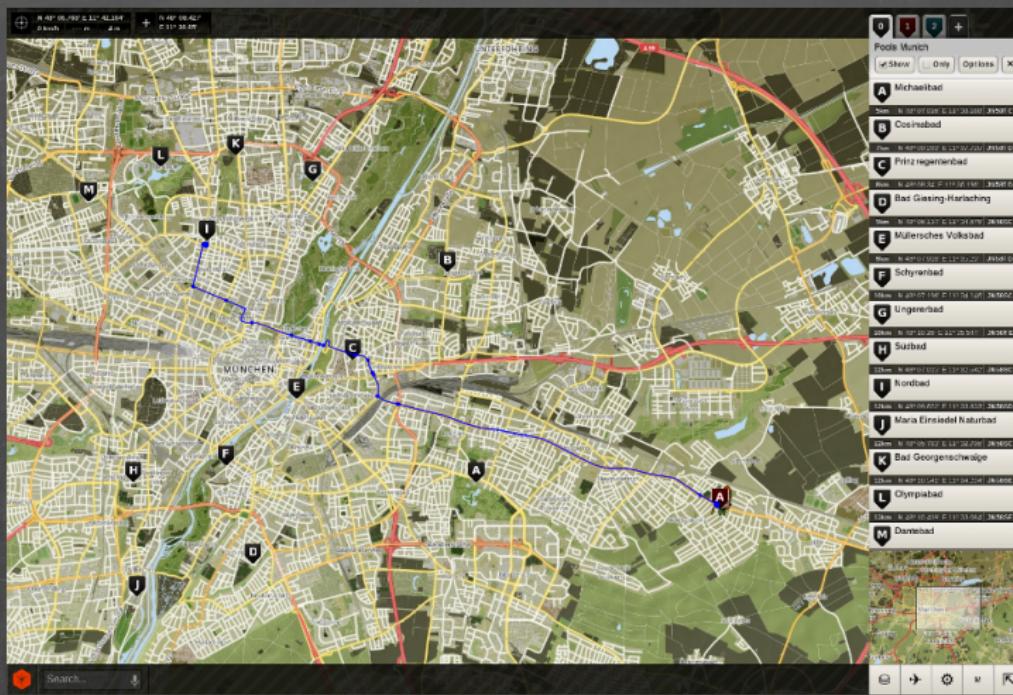
- ▶ HTML5/JavaScript
- ▶ ModestMaps

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 basemaps and overlay feeds

- ▶ Federated
- ▶ Searchable
- ▶ Ranked
- ▶ Geobound
- ▶ Tagged

Attributes of a directory entry:

- ▶ url
- ▶ Type (overlay or basemap)
- ▶ Topleft, bottomright
- ▶ Name
- ▶ Description
- ▶ Tags
- ▶ Rank

## Client

- ▶ More overlay functionality
  - ▶ Polygons
  - ▶ (Translated) images
  - ▶ 3D
- ▶ Better mobile integration
  - ▶ iOS
  - ▶ Android
  - ▶ Glass!
- ▶ Overlay browser
- ▶ Encryption

## Collaboration

People ...

- ▶ ... forging and serving basemaps for their area
- ▶ ... exposing existing geodata as dspace overlay feeds
- ▶ ... helping with docs, bugs, issues, features

## The End

Thanks for your attention. May the daemon be with you.

[QR + URL]

Questions?