

# 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

Navigation

DSpace Client

What do we need?

## 1. Who we are?

# Who we are?



**chrono0 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?

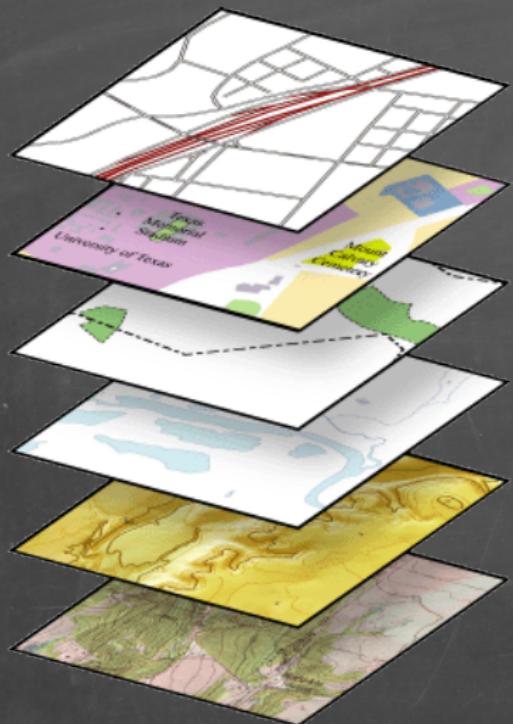
Staging our wants

## Something like a standard

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

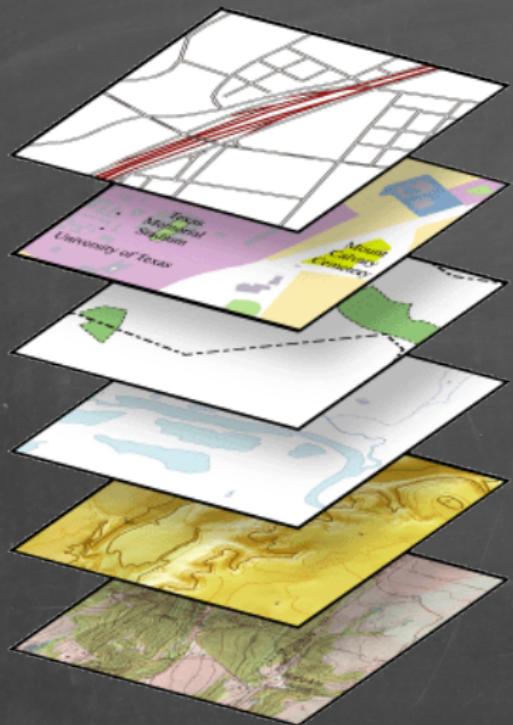
- ▶ Federation
- ▶ Free
- ▶ Open Source
- ▶ Lose Bindings
- ▶ Modular Extensions

## Basemaps



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

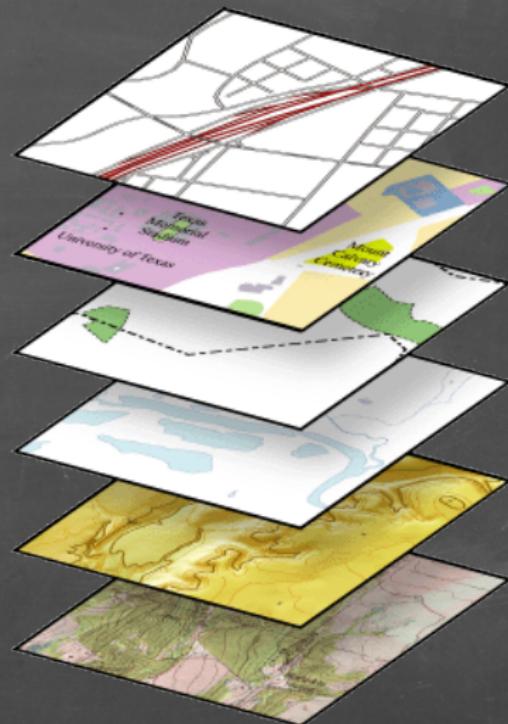
## Basemaps II



- ▶ Static / Longterm data retention validity
- ▶ General interest
- ▶ Only one map is visible at a time
- ▶ Composition depends on region/application
- ▶ Updates are resource intensive (Rendering)

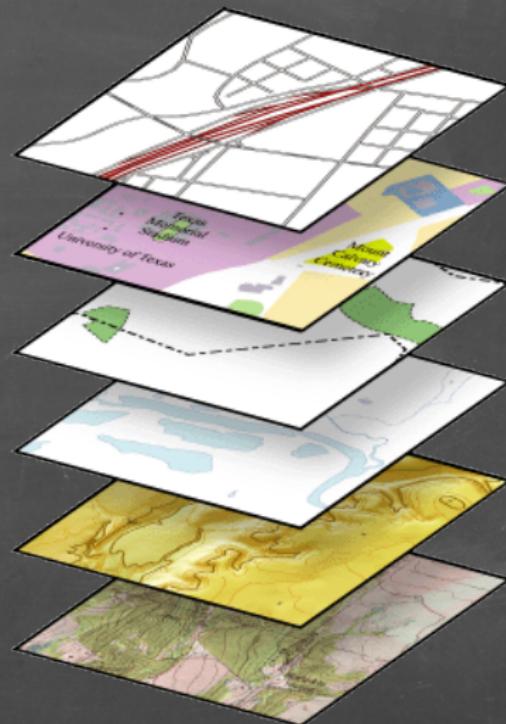
## Overlays

- ▶ POIs
- ▶ Location tracking
- ▶ Waypoints on a route
- ▶ (Basemap)



## Overlays

- ▶ collections of things at locations
- ▶ public or private
- ▶ can be very dynamic (e.g. realtime tracking)
- ▶ many can be visible (overlaid) at a time
- ▶ can be user-generated and -updated
- ▶ Very fast & cheap updates (local browser renders)



## 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

Image of Architecture TileMill -> TileStream -> DSpace Client |-  
Directory Server -Remotestorage -feeds -Routino

3. What do we have?

## What do we need?

Staging our haves

### 3. What do we have?

## What do we have?

Before we start to re-invent the wheel, let's have a look at what other generous people already have developed and shared with the rest of humankind.

## What do we have?



TileMill



BACKBONE.JS



DLR



REMOTE  
STORAGE



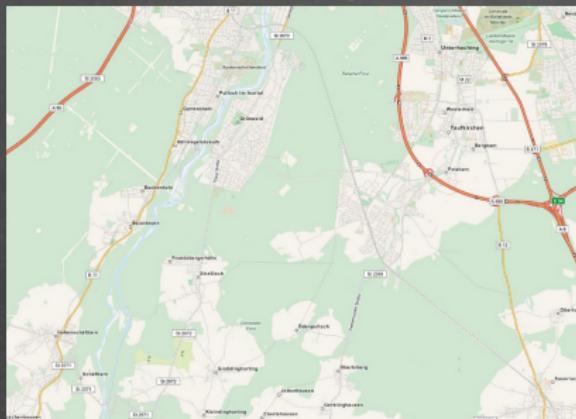
CGIAR CSI  
Consortium for Spatial Information



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

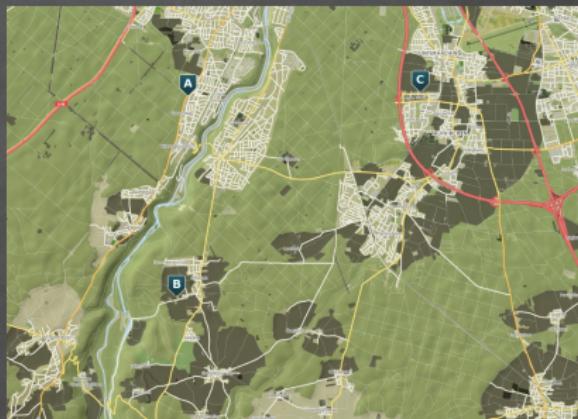
# Why not use OpenStreetMap Map-Servers?

Not everything is in OSM:



Topography, Aerial Imagery

Not everything belongs in OSM:



Tracking, Personal/Private POIs

# 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
- ▶ TerraSAR-X/TanDEM-X data (Future)

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

## Map Forge Screenshot

Include picture of Map Forge in action

## Map Delivery

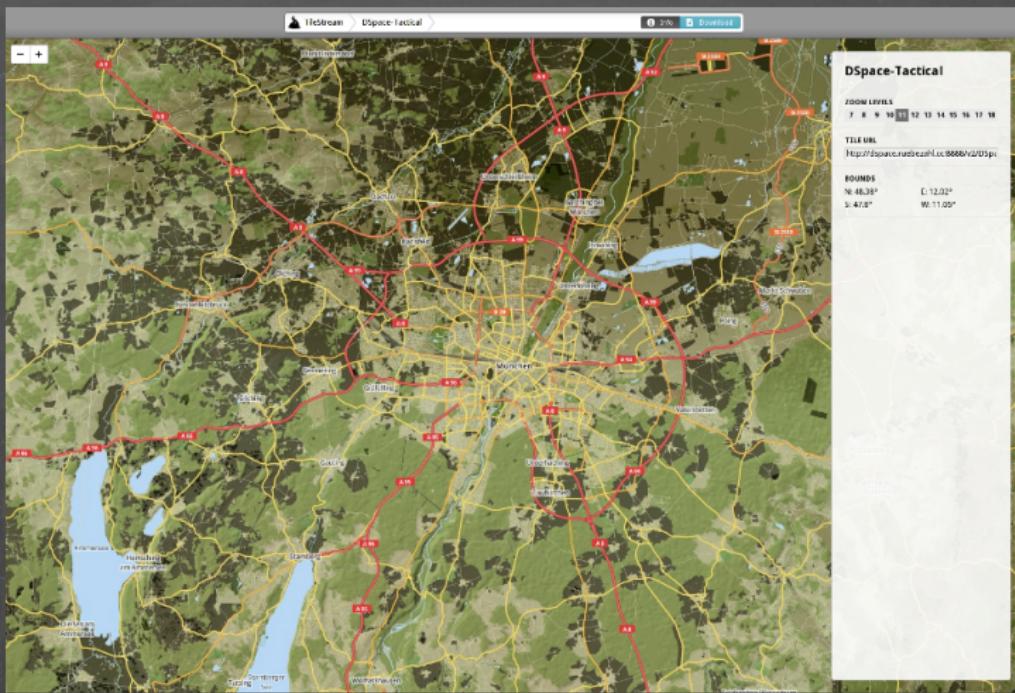
TileStream is also written in NodeJS and very easy to set up:

- ▶ 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



## Read-Only Overlays

- ▶ Simple HTTP GeoJSON Feed
- ▶ SpaceAPI

## Read-Write Overlays

- ▶ remotestorage.io

## Navigation

- ▶ Routino
- ▶ OpenStreetmap import

## DSpace Client

Introduction on the Client now we have nice basemaps and sources for overlays  
presentation comes together in the client

# Client

Live DEMO

## Client

Client-side js assembled, built and packaged in node.js focus on:  
as lightweight as possible Powerful Plugin-API Mobile  
Readiness/Integration

# 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

Taking care of easy and structured deployment to leave flexibility for different setups and potential rewrite issues.

```
+ 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

Federated searchable ranked, geobounded, tagged list of basemaps and overlay feeds ...

## Client

more overlay functionality (polygons, 3D, translated images ...)  
mobile integration (ios, android, glass) Overlay browser

## Collaboration

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

#### 4. What do we need?

## Utopia

Augmented Reality glasses (contact lens FTW!) ...

#### 4. What do we need?

THEEND

Thanks for your attention.  
Discussion