

Server

This section covers the server side infrastructure of the system. It currently includes the following components:

- A **Geotemporal Data Service** to store and retrieve data using a JSON RESTful web interface
- The **Medici** content management system to store the original data sets and metadata about them

A preliminary overview and design is presented in this [slide set](#).

Geotemporal Data Service

The geotemporal data service is a RESTful web service to store and retrieve data points that are defined both in space and time. The goal is to make a service that horizontally scales, has a flexible data model (variables don't need to be predefined) and support complex geospatial queries (including polygon based queries). In the current design the data points will reside in database that has scaling built-in (currently [MongoDB](#)), with a dedicated geospatial index in [PostGIS](#) to take be able to have sophisticated geospatial queries (something MongoDB has only basic support for). The end user, in this case the application programmer, will only have to learn the JSON API. The details of how things are stored will be hidden from them. This will also make it easier to change underlying implementations later on.

JSON API

The current design only includes a JSON interface, but ideally more serialization mechanisms could be added later on. The serialization of geospatial data into JSON tries to follow the [GeoJSON](#) standard. Currently only a subset of the standard is supported, with the idea of slowly building up to the full spec.

Method	URL	Description
POST	/datapoints	Add a datapoint
GET	/datapoints	Search via query params (<i>default: all datapoints</i>)
GET	/datapoints/{id}	Retrieve a specific datapoint
POST	/sensors	Add a sensor
GET	/sensors	Search via query params (<i>default: all sensors</i>)
GET	/sensors/{id}	Retrieve a specific sensor
GET	/sensors/{id}/stats	Retrieve stats for a sensor

Datapoint

```
{
  "time": "2004-09-04T18:06:22Z",
  "location": {
    "geometryType": "Point",
    "coordinates": [1.1, 0.5]
  },
  "data": {
    "temp": "42",
    "conductivity": "0.008489"
  }
}

{
  "time": "2004-09-04T18:06:22Z",
  "location": {
    "featureType": "FeatureCollection",
    "features": [
      {
        "geometryType": "Point",
        "coordinates": [1.1, 0.5]
      }
    ]
  },
  "data": {
    "temp": "42",
    "conductivity": "0.008489"
  }
}
```

Code

Source code: <https://opensource.ncsa.illinois.edu/stash/projects/GS/repos/geostream-service/browse>

The current implementation is written in [Scala](#) using the [Scalatra](#) framework. The data is currently being stored in [MongoDB](#) with a geotemporal index in [Postgres](#) + [Postgis](#) (work in progress).

It uses the following libraries:

- [lift-json](#) - json serialization
- [salat](#) - scala case classes to MongoDB serialization
- [specs2](#) - unit testing