

Introduction

Vizmo provides easy visualization of geospatial data in web applications. A fundamental goal is to solve a common requirement surprisingly unaddressed by most mapping libraries: **displaying large sets of geospatial data** (thousands or tens of thousands of points), **and grouping or “clustering” in a semantically-meaningful way** -- namely, by hierarchical administrative districts like cities and provinces and countries rather than simply, and dumbly, by adjacency.

Vizmo has two components:

- 1) a set of javascript user interface widgets
- 2) a server application that provides JSON data for consumption by those widgets

Javascript and server are designed to work together but each may be used independently. The only “contract” between them is a set of specified JSON formats, the most important of which is an extension of the standard GeoJSON.

Mapping Widgets

The key UI widget is a javascript mapping component based on OpenLayers. The map displays location points from a GeoJSON source, with additional metadata to support hierarchical clustering. Data transport is ajax-style without refreshing the page, triggered by map viewport changes as well as explicit user input.

The clustering logic is implemented server-side; the map widget simply specifies a bounding rectangle and, optionally, an explicit grouping level.

The widget offers a relatively simple library for UI implementors, in contrast to complexity of OpenLayers, although the full power of OpenLayers is accessible for expert users.

Data Server

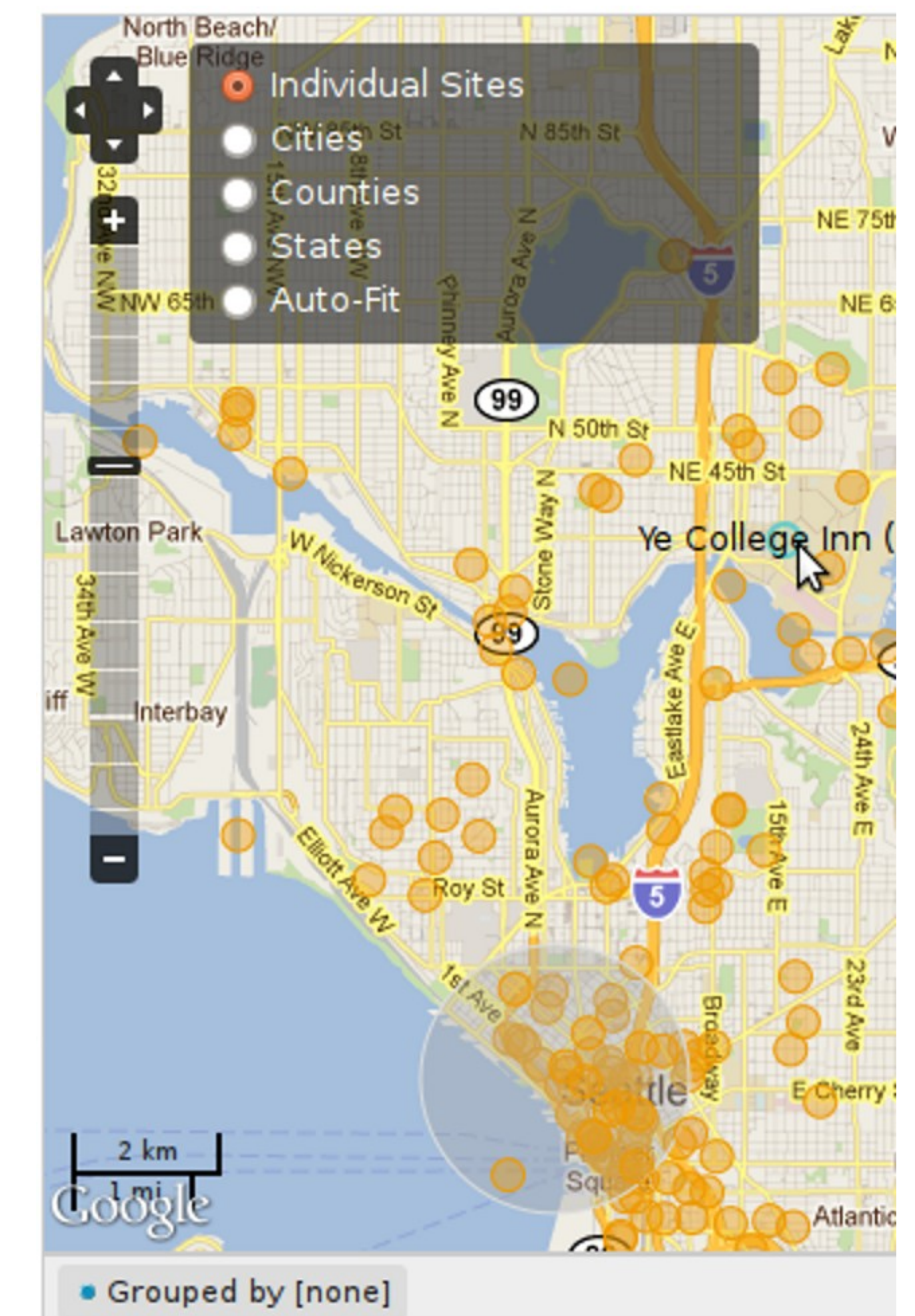
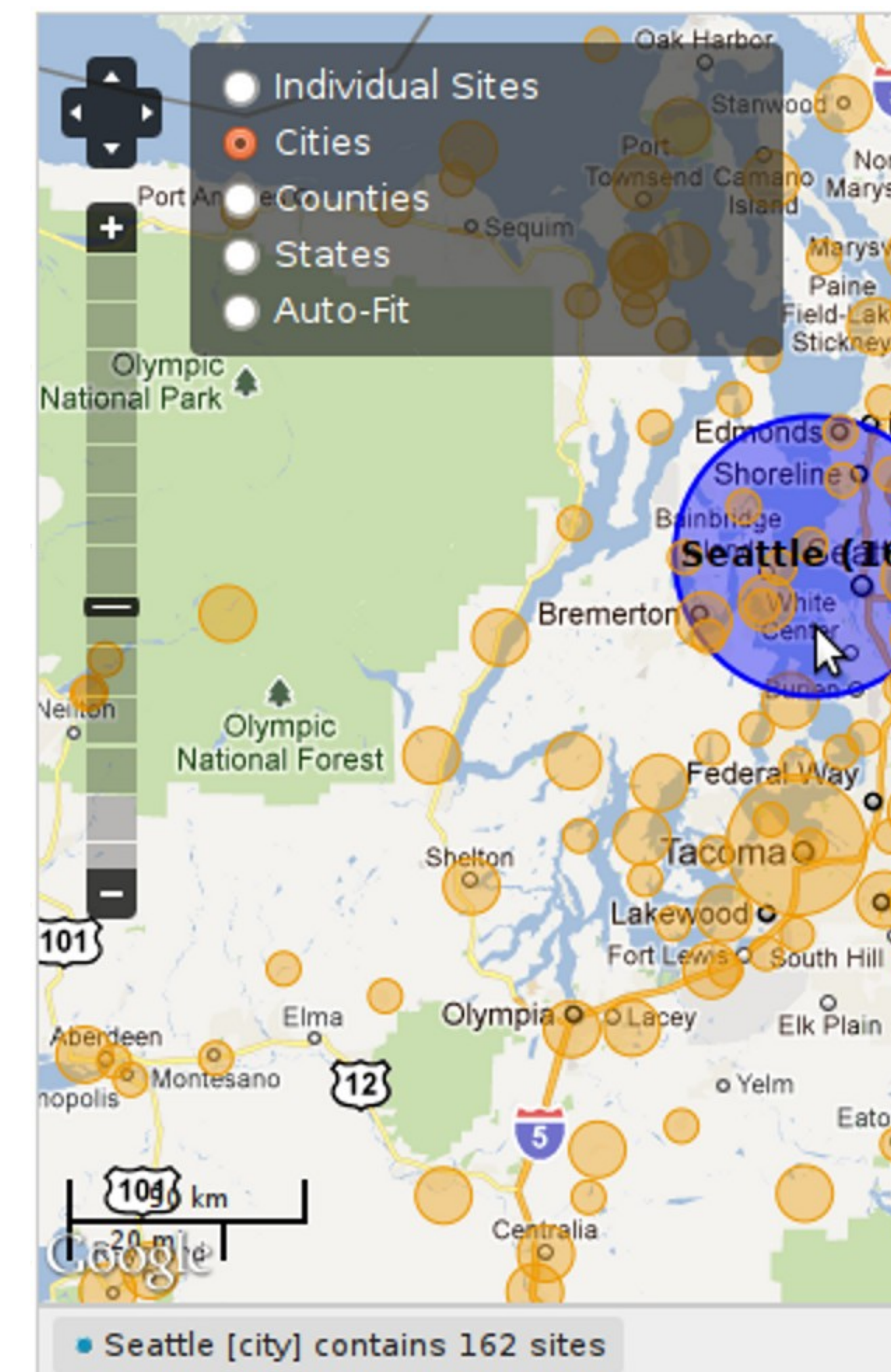
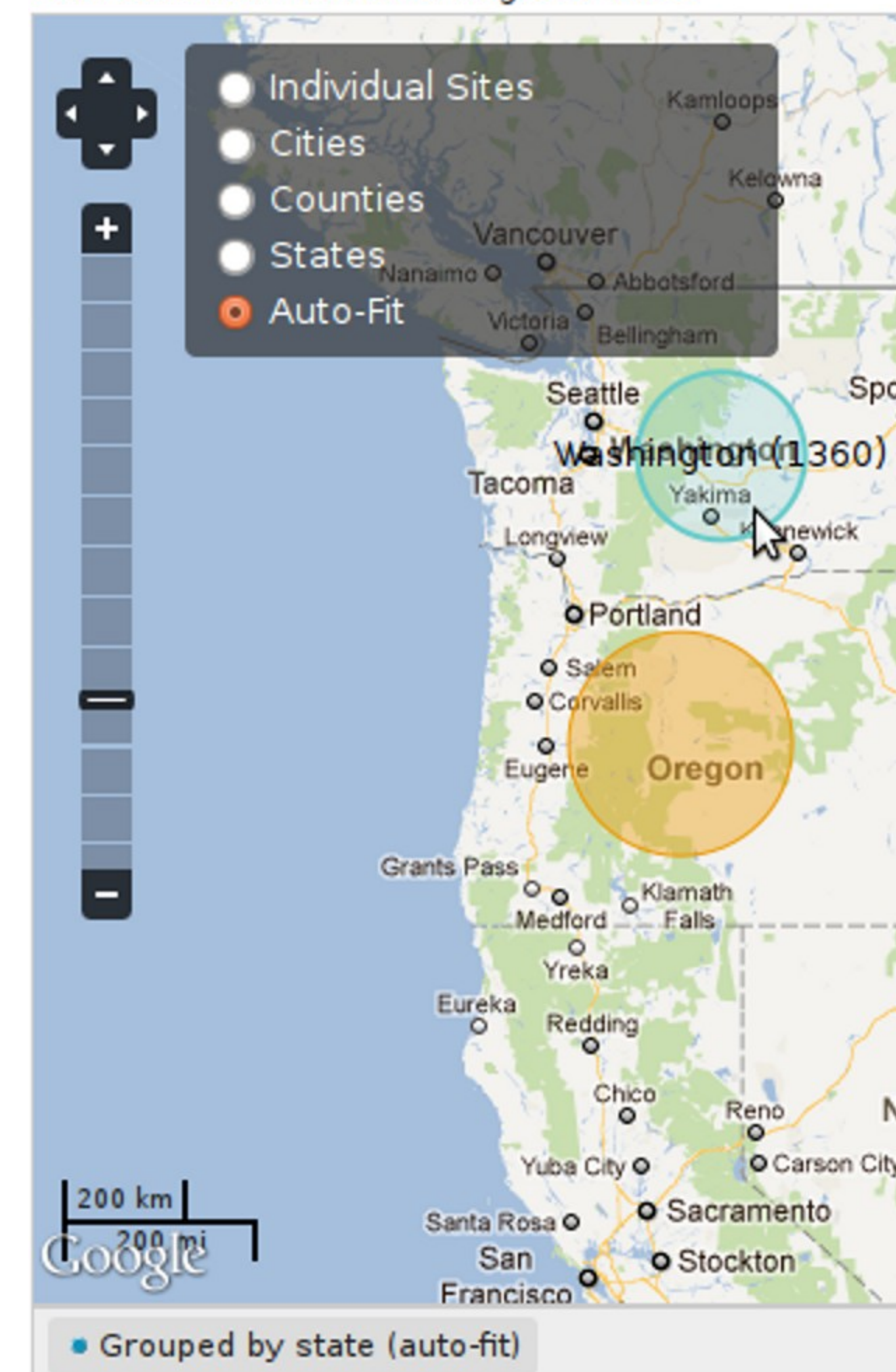
The Vizmo server provides data services for the Vizmo UI widgets. Given a conventional database schema for hierarchical data, queries can be defined simply through configuration files without writing any code.

A key feature is its implementation of a clustering algorithm that determines a “best fit” grouping for a given map bounds and geospatial dataset. Several parameters are configurable to optimize performance, including a maximum number of points/clusters per map, and a hard limit to prevent map point overload.

Built with the Symfony2 php framework, the server was designed to work with a MySQL backend. MySQL, in spite of rather weak geospatial features, offers enormous advantages in terms of resource requirements and ease of deployment, particularly in shared hosting environments. This approach is not meant to replace more sophisticated solutions like Geoserver or MapServer or Postgres, but rather to offer a workable alternative for simpler applications.

Hierarchical Clustering by Meaningful Units — in this example U.S. states, counties, cities.

U.S. National Historical Register Sites



Test Dataset from the National Register of Historic Places, National Park Service, U.S. Department of the Interior (www.nps.gov/nr/research/). Approximately 80,000 sites, grouped into 20,000 administrative districts (states, counties, and cities).

Map-Aware Widgets and Other Features

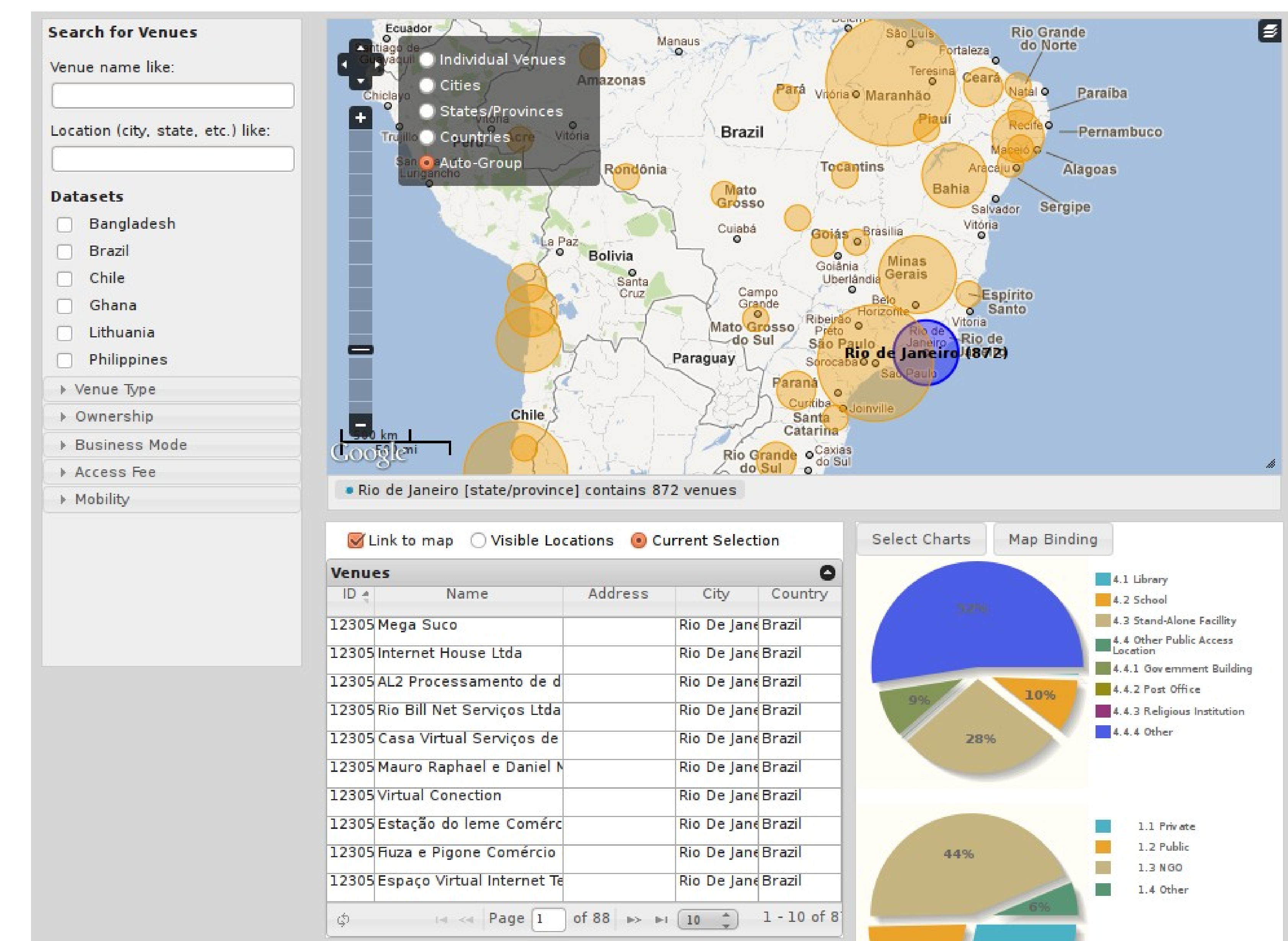
Vizmo includes chart and list view widgets that can be “linked” to a map, so that the display is dynamically updated to reflect map state. Binding operates in one of two modes: widget state reflects either (1) what is *currently visible* on the map viewport, or (2) the *currently-selected* map feature.

These widgets are built with the jQuery-based jqPlot and jqGrid libraries. As with the use of OpenLayers, they provide a simple API to support common use cases but also make accessible the powerful features of the underlying libraries.

The map provides an API for binding to a data entry form, so that the features displayed can be easily filtered by related attributes.

The server also provides data query components to support the chart and list widgets, following the same approach as used for the geospatial queries.

Deployed in combination these widgets can provide rich data visualization capabilities in a web application.



Global Impact Study inventory web application, version 2.0, prototype. Public release expected late 2011.

Project History and Affiliations

These tools were developed for the forthcoming version 2.0 release of the inventory web application of the Global Impact Study of Public Access to Information & Communication Technologies www.globalimpactstudy.org, a project implemented by the University of Washington Technology & Social Change Group tascha.uw.edu with support from Canada's International Development Research Centre (IDRC) and a grant to IDRC from the Bill & Melinda Gates Foundation. Sources will be hosted at tascha.bitbucket.org. Contact: atulinsk@uw.edu, chriskr@uw.edu