[CP-04-014] Web GIS

Abstract

Web GIS allows the sharing of GIS data, maps, and spatial processing across private and public computer networks. Understanding web GIS requires learning the roles of client and server machines and the standards and protocols around how they communicate to accomplish tasks. Cloud computing models have allowed web-based GIS operations to be scaled out to handle large jobs, while also enabling the marketing of services on a per-transaction basis.

A variety of toolkits allow the development of GIS-related websites and mobile apps. Some web GIS implementations bring together map layers and GIS services from multiple locations. In web environments, performance and security are two concerns that require heightened attention. App users expect speed, achievable through caching, indexing, and other techniques. Security precautions are necessary to ensure sensitive data is only revealed to authorized viewers.

Many organizations have embraced the web as a way to openly share spatial data at a relatively low cost. Also, the web-enabled expansion of spatial data production by nonexperts (sometimes known as “neogeography”) offers a rich field for alternative mappings and critical study of GIS and society.

Keywords: dissemination of geospatial information, geospatial web services, managing the GI system operations and infrastructure, web mapping, Web service

Author & citation

Quinn, S. (2018). Web GIS. The Geographic Information Science & Technology Body of Knowledge (1st Quarter 2018 Edition), John P. Wilson (ed). DOI: 10.22224/gistbok/2018.1.11

Explanation

1. Definitions
2. Introduction
3. Web GIS Architectures
4. GIS Web Services
5. GIS Client Applications
6. Web GIS Performance
7. Security and Data Integrity in Web GIS
8. GIS Web Services in the Cloud
9. Web GIS and Society

1. Definitions

AJAX: Asynchronous JavaScript and XML. A technique that allows a web page to fetch information from the server without doing a complete refresh. Used in web maps for smooth navigation.

API: Application programming interface. An organized set of programming components that can be used to develop apps using a particular language.

client: A computer or other device that requests information or processing from a server.

cloud: A networked group of computers that can be used to supplement one's own computational infrastructure. Access to clouds is often sold or rented for a metered fee.

HTTP(S): Hypertext Transfer Protocol (Secure). A mutually understood set of commands and parameters often used by clients and servers to communicate over the Internet.

server: A powerful computer that offers specialized tasks or data, which it sends to other computers or devices (clients) upon request.

SSL: Secure Sockets Layer. A web security strategy wherein communication is encrypted and restricted to sites whose identity is proven. Used with the HTTPS protocol.

tiles: Tessellated bundles or packets of geographic data, usually square in shape, that are generated by the server at multiple scales and kept in a cache to improve map delivery speed. Can store vector coordinates or rasterized map images.

VGI: Volunteered geographic information. Geographic data offered by non-experts and collected through everyday activities and common devices, such as smartphone GPS receivers.

web service: Code that runs on the server and can perform a specialized task in response to a client request.

2. Introduction

A web GIS is a computer software and hardware configuration that allows the sharing of maps, spatial data, and geographic processing operations throughout one's own network and beyond using common web communications protocols such as HTTP (Dragićević, 2004) and WebSockets. In a web GIS, resources are often sent from a centralized machine (or "server") upon request. Organizations may choose to implement web GIS to share computing capabilities and resources across departments on a local intranet, while others may use web GIS as a vehicle for disseminating information to a worldwide audience (Plewe, 1997). In some situations the public might even be invited to contribute or update data in an online environment.

Web GIS reduces the need for end users to install sophisticated software to work with geographic information, as most operations can be performed in web browsers or lightweight mobile apps. Thus, the web offers the potential to lower the GIS learning curve and increase the accessibility and exposure of GIS to the public. Implementing web GIS on one's own servers requires a substantial investment in hardware, software, and connective infrastructure, as well as the skills to link and configure those components. As a result, new business models are emerging where GIS is sold as a package of services running on cloud architectures.

3. Web GIS Architectures

A web GIS centers around one or more machines called servers that respond to requests from other machines, called clients. Servers are usually configured with more computing power than clients, and run specialized operations in response to client requests. They may also send data to clients for processing. GIS-focused servers can perform cartography and spatial processing operations such as drawing maps and running geocomputational algorithms, respectively (Peng, 1999). The client must know how to work with the server's response, whether that is an image of a map, some geographic coordinates, a list of attributes, some data that the client itself must process, etc.

Servers must allow inbound and outbound network traffic, necessitating configuration of firewall rules by system administrators. GIS servers must often connect to other types of servers on the network to fully accomplish their tasks; for example, an organization's existing web server needs to be configured to screen and forward incoming requests to the GIS server. Organizations may also maintain a separate data server that must be allowed to communicate with the GIS server. Figure 1 shows one common way that servers and clients are connected in a web GIS architecture.

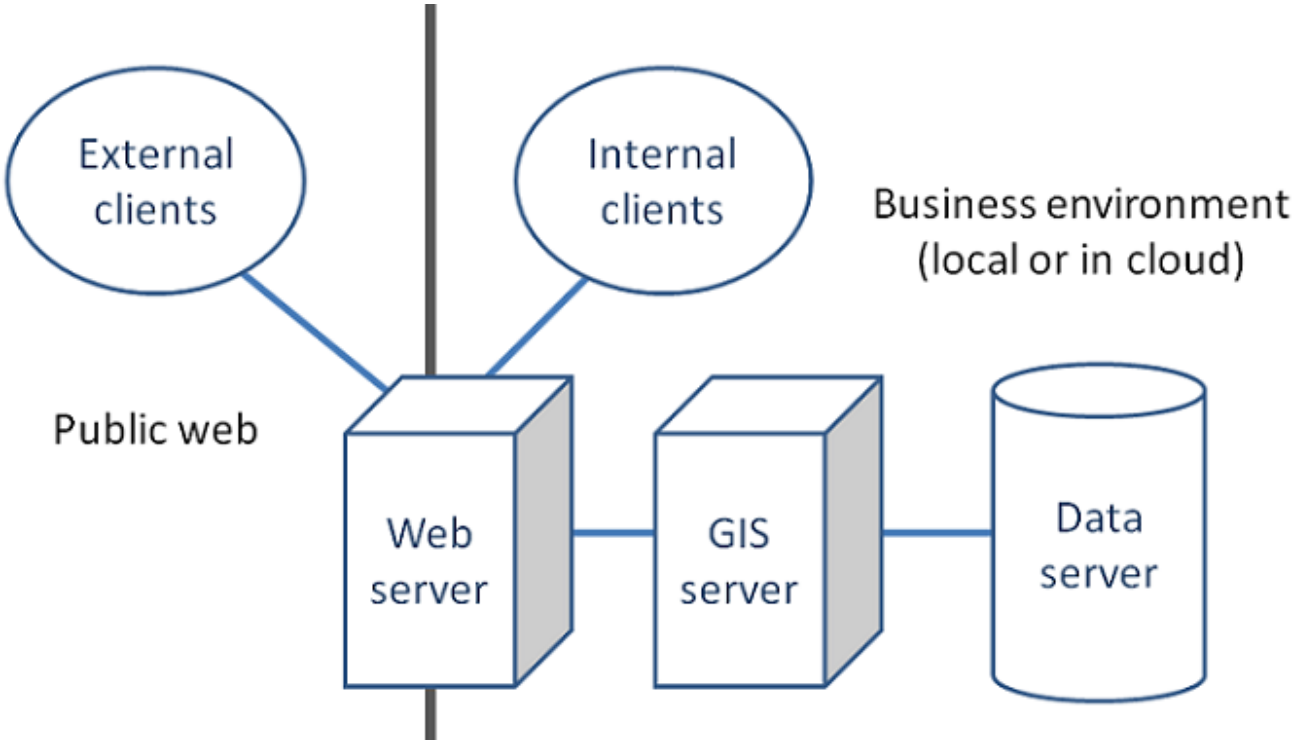


Figure 1. Elements of a common web GIS architecture.

4. GIS Web Services

A variety of commercial and open source software options are available for setting up GIS servers. Examples include GeoServer, MapServer, and Esri ArcGIS Enterprise. Their core common feature is the ability to create web services for GIS and mapping. A web service is a piece of code that runs on the server and can perform some action in response to a client

request. For example, a web service might receive some bounding coordinates and an image format from a client, draw a map image, and then send the image back to the client. Another type of web service might receive some vector feature coordinates, calculate an intersection of those features, and send back to the client the coordinates of the intersecting feature set.

In order for a client and server to interact successfully, both must understand the format and protocol for issuing the request and receiving the response. Most web GIS transactions are executed over hypertext transfer protocol (HTTP) or its encrypted counterpart HTTPS. Furthermore, the client and server must understand a common syntax for the parameters of the request and the format of the data to be returned. To facilitate this, the Open Geospatial Consortium (OGC) has defined open specifications for GIS web services. These include the web map service (WMS) and web map tile service (WMTS) for requesting maps, the web feature service (WFS) for requesting vector feature geometries and attributes, the web coverage service (WCS) for raster data requests, and the web processing service (WPS) for spatial data processing operations. In the realm of proprietary software, the Esri GeoServices specification outlines very similar types of services for map delivery, vector feature transfer, “geoprocessing”, 3D scene viewing, and other GIS-related operations.

5. GIS Client Applications

Clients of a web GIS can include locally installed desktop apps, browser-based apps, and native mobile apps. Open source and proprietary application programming interfaces (APIs) abound for creating custom web and mobile applications that consume GIS web services. These include Open Layers, Leaflet, and the ArcGIS APIs. In order to provide an intuitive user experience, these client apps and APIs often hide the details of all the web service requests occurring in the background. For example, a user of a client app may pan and zoom a map without needing to know all the required parameters for the back-end web service to redraw the map with a new scale and bounding coordinates.

Many maps displayed in client apps originate from multiple web services that are each treated as layers in the final displayed map. For example, a common workflow is to request a base map image from one web service and a thematic map overlay from another web service. The thematic layer can be requested either as an image or as vector data to be drawn by the client. The client app combines the layers and may allow the end user to toggle their visibility or order. Decoupling the base map from the thematic layers allows clients to re-use base maps in many applications, or employ base maps from well-known third parties, such as Google Maps. The term “mashup” is often used to describe a map created from two or more web services in this way.

The server can deliver vector data to client apps either as a standalone file or a streaming set of features. Popular file formats for this kind of data include GeoJSON, Esri GeoServices JSON, and KML. The OGC has also defined a specification for vector features called Geography Markup Language (GML), although the verbosity of this format has led to performance concerns (Yang, Wong, Yang, Kafatos, & Li, 2005). The approach of drawing graphics on the client promotes interactivity and generally improves speed, although it can also slow down apps when the files contain many features.

Some client apps may allow the editing of vector features transacted back and forth

between a GIS database. Web GIS administrators must decide whether to expose the production copy of their database for editing, or restrict editors to modifying an intermediate copy that is quality-checked by an analyst. Policies must also be established for conflict resolution when multiple editors are working on the same database at the same time, especially if these editors commonly upload bulk changes after offline edit sessions. A simple solution is a “last in wins” policy where the most recently uploaded edit takes precedence. More complex use cases may require data versioning or conflict review on a feature-by-feature basis.

GIS users are increasingly deploying mobile devices such as smartphones and tablets as clients of GIS servers. Because these devices are easily portable, they can send location information to the server in real time as the device is carried from place to place; however, mobile clients are also useful in circumstances where connections to the server are intermittent or unavailable. To support these offline or partially-connected scenarios, the server sometimes allows a data extract or download functionality so that some data may be placed locally on the device in preparation for field work. The server might also allow a data upload or synchronization action wherein edits made in the field can be sent back to the server and reconciled with the server’s copy of the data.

6. Web GIS Performance

Users of the web expect a fast response from their apps, which can be challenging with the massive amounts of data and processing required by GIS. Administrators can improve speed by reducing the number of transactions between clients and servers, as well as the amount of information in each transaction. Only the minimum necessary map layers and attribute data should be included in server requests and responses. Simplification of map symbols, vector coordinate geometries, and decimal numbers (through rounding) can reduce the size, or payload, of web transactions, with the awareness that such measures will trade away some precision.

Strategies for speeding web map performance include the use of tree indexing, multithreading, and cluster computing (Yang et al., 2005). Another tactic is to precalculate the results of common requests. For example, if web users will frequently need to view the area measurements of polygons, these areas can be precalculated using desktop software and written into an attribute field. The server can return this value faster than it could perform the area calculation.

Another common example of precalculation is the division of data into a multi-scale pyramid of tessellated regions, or “tiles”, that the server caches and passes out to clients without having to extract or draw data in real time. The integration of pre-drawn rasterized map tiles with asynchronous JavaScript and XML (AJAX) fetching techniques creates a map navigation experience so fast that it can appear “seamless”. Pioneered by Google Maps to great popular acclaim in the mid-2000s, this approach is now common in web maps.

More recently, the tile motif has been expanded to vector data packets which are drawn by the client, allowing a more flexible restyling of cartography than is possible with rasterized tiles. WebGL (Web Graphics Library) technology is often used for drawing vector tiles or thematic features on the client side in a way that is faster than the traditional SVG (scalable vector graphics) approach.

7. Security and Data Integrity in Web GIS

Web GIS administrators must ensure that sensitive data is not revealed to a larger audience than intended. Security rules can be applied at various tiers, and may restrict access to end user applications, GIS web services, map layers, geographic features, or feature attributes (Bertino, Thuraisingham, Gertz, & Damiani, 2008). When vector data is proprietary or copyright-protected, it may be desirable to show only a rasterized image of map data rather than allow the download of each vertex coordinate. Limits should be placed on the scope of web service requests to avoid web scraping, mass downloads, or enormous data processing jobs that overburden the server (whether intentionally or unintentionally).

Web GIS administrators must work closely with their organization's existing IT management staff to ensure that web services are secure. For greater administrative control and end-user convenience, web GIS systems and apps should be integrated with the organization's existing login infrastructure when feasible. Standard web safety practices can boost the security of a GIS, such as granting users only the minimum privileges necessary to do their jobs, restricting physical access to server machines, requiring strong passwords that are changed on a regular basis, and so forth. All passwords and any sensitive spatial or tabular data should be transferred in encrypted form via secure sockets layer (SSL) connections. Any user input allowed into the web GIS should be screened by client and server code for malicious intentions, such as structured query language (SQL) injection attacks.

8. GIS Web Services in the Cloud

GIS and mapping web services can run on onsite servers or cloud-based hardware. Cloud computing also enables the commoditization of GIS through a software as a service (SaaS) model, wherein companies offer GIS web services in exchange for a usage fee. Examples include Esri ArcGIS Online, CARTO, and Mapbox. Fee-based services offered on the cloud might include data storage, online map viewing, geocoding, routing, and the use of spatial processing algorithms. A benefit of these offerings is that there is no need to acquire and administer onsite GIS servers; however, users still need to pay attention to Internet connectivity speed and bandwidth, as well as the size and security of data transfers.

Another advantage of GIS in the cloud is elasticity, or the ability to expand back-end computing resources to meet the scope of large jobs. Organizations with large cyberinfrastructure could integrate GIS services into their own clouds to reap the benefits of elasticity.

Users of cloud GIS services must pay attention to economics to understand whether the fees associated with the cloud services are lower than the costs of deploying a server onsite. These calculations should include the cost of the hardware, software, infrastructure (such as physical building and utilities), and human resources needed to operate a server.

9. Web GIS and Society

Makers of maps and GIS data often rely on the web as a simple way to deliver content to the public. This includes the sharing of downloadable vector and raster data files through searchable websites sometimes called geoportals. Where governments are mandated to share data, doing so online can be more economical than using traditional hard media formats (Plewe 1997). The web also augments the potential of public participation GIS (PPGIS); for example, inviting national park visitors to point and click on a web map to identify areas of high intrinsic value and places lacking maintenance (Brown & Weber, 2011).

The increasing interactivity of browser-based and mobile apps, along with the relative inexpense of satellite navigation receivers, has allowed mass public creation of spatial data through the web. This is notable because such activities were traditionally the prerogative of trained experts. The changing tides are evident in the many user-generated layers shared in Google Earth, the emergence of volunteered geographic information (VGI) projects such as OpenStreetMap, or the simple attachment of geographic coordinates to social media posts. Furthermore, ready-made web application templates and embeddable frames have made it easier for non-experts to introduce maps and GIS into client applications. To what degree these forms of “neogeography” have actually democratized mapping and spatial data creation is a point of active discussion (Turner, 2006; Goodchild, 2007; Haklay, 2013), although the web has certainly enabled alternative cartographies and counter mappings to reach a wider audience than would otherwise be possible.

References

Bertino, E., Thuraisingham, B., Gertz, M., & Damiani, M. L. (2008). Security and Privacy for Geospatial Data: Concepts and Research Directions. In Proceedings of the SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS (pp. 6-19). New York, NY, USA: ACM.

Brown, G. & Weber, D. (2011). Public Participation GIS: A new method for national park planning. Landscape and Urban Planning, 102(1):1-15.

Dragićević, S. (2004). The Potential of Web-based GIS. Journal of Geographical Systems, 6(2), 79-81.

Goodchild, M. F. (2007). Citizens as sensors: the world of volunteered geography. GeoJournal, 69(4), 211-221.

Haklay, M. (2013). Neogeography and the delusion of democratisation. Environment and Planning A, 45(1), 55-69.

Peng, Z.-R. (1999). An Assessment Framework for the Development of Internet GIS. Environment and Planning B: Planning and Design, 26(1), 117-132.

Plewe, B. (1997). GIS Online: Information Retrieval, Mapping, and the Internet (1st edition). Santa Fe, NM: OnWord Press.

Turner, A. (2006). Introduction to Neogeography. O'Reilly Media, Inc.

Yang, C., Wong, D. W., Yang, R., Kafatos, M., & Li, Q. (2005). Performance-improving

techniques in web-based GIS. International Journal of Geographical Information Science, 19(3), 319-342.