Proceedings XI GEOINFO, Campos do Jordão, Brazil, Nov. 28 to Dec. 1, 2010, INPE, p. 57–66.

# Low-cost Satellite-based Products for the Web - the Example of Fire Web Service

**Theodor Foerster[1], Thore Fechner[1], Holger Fritze[1], Frederik Loock[1] & Albert Remke[2]**

[1]Institute for Geoinformatics (ifgi) - University of Muenster
Muenster, Germany

[2]52° North - Initiative for Geospatial Open Source software GmbH
Muenster, Germany

{theodor.foerster,t.fechner,h.fritze,f.loock}@uni-muenster.de

a.remke@52north.org

***Abstract.*** *GEONETCast is a satellite-based dissemination system for a range of in-situ and remote sensing products. Enabling such products for the Web is interesting to ease a broad data access for industry and academia. In this context, standardized Web Service interfaces and data formats play an important role. In this article, we present an architecture for accessing satellite-based MODIS fire data products through the so-called Fire Web Service. A browser-based application allows users to query up-to-date but also historic data about fire events. The presented architecture has been implemented based on Open Source software.*

## 1. Introduction

Broadcasting in-situ and remote sensing data through a satellite network for low-cost is important to support developing countries in planning and monitoring their environment with up-to-date data. One example of such low-cost satellite-based broadcasting system is GEONETCast [Wolf and Williams 2008]. It provides space-based, air-borne and in-situ data, metadata and products usable by diverse communities through a global network of satellite-based data dissemination systems. Consequently, GEONETCast products can be received through standard satellite dishes and are directly accessible through the connected computer. However, these products are only available locally (i.e. on the PC, to which the dish is connected) through mostly proprietary data formats and specific software. With the increasing availability of the Web an interoperable, easy and worldwide access to this data is promising but has not been realized yet.

This paper describes an architecture to serve GEONETCast products on the Web using standardized data formats and Web Service interfaces. The architecture is demonstrated by the use case of a Fire Web Service. Detecting fires through satellite data such as Moderate Resolution Imaging Spectroradiometer (MODIS) is one of the important tasks when monitoring large-scale rural areas. The architecture allows users to query up-to-date and historic fire events through a browser-based application. The presented architecture is based on Open Source software, which also motivates the use of open standards.

Section 2 gives an overview of the products available in GEONETCast and how those products can be received. The architecture and its implementation to serve these

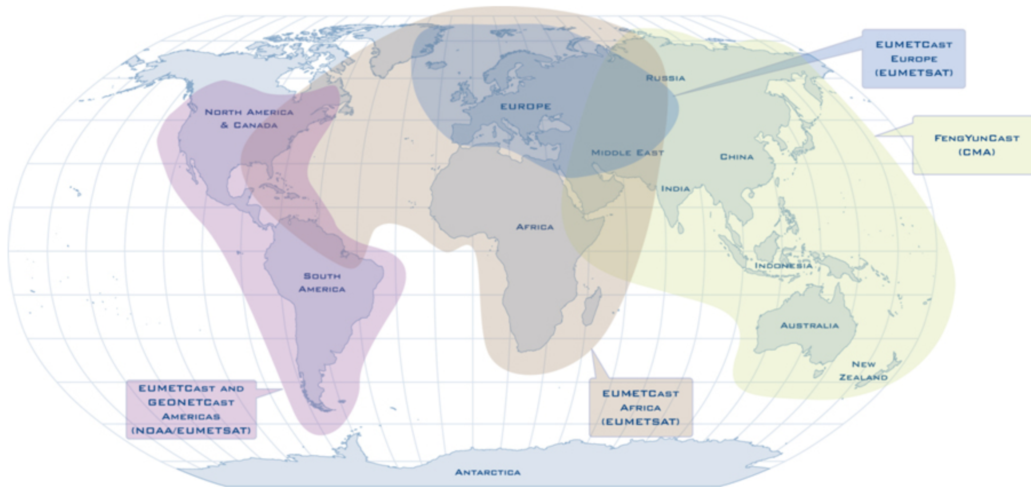


**Figure 1. Overview of GEONETCast satellite systems.**

products on the Web for the example of the Fire Web Service is presented in Section 3. The use case of the Fire Web Service is then presented in 4. The paper ends with an outlook and conclusion of the presented findings.

## 2. GEONETCast - an Overview

GEONETCast is a satellite-based dissemination system for environmental data created by remote and in-situ sensors. GEONETCast is thereby a part of the Global Earth Observation System of Systems (GEOSS). In particular, GEONETCast is a task in the GEO Work Plan and is led by EUMETSAT, the United States, China, and the World Meteorological Organization (WMO). Many GEO members and participating Organizations contribute to this task. The dissemination of GEONETCast products is based on the following satellites:

- FENYUNCast(Asia)
- EUMETCast (Africa & Europe)
- GEONETCast Americas (North and South America)

An overview of the different satellite systems and their coverage is also depicted in Figure 1.

Section 2.1 will describe the different thematic types of products and their application. The technical setup of a system for receiving such GEONETCast products is presented in Section 2.2.

### 2.1. GEONETCast Products

With over 180 products GEONETCast offers a broad thematic range from spectral transmission and climate measures (i.e. surface temperature, precipitation) to disaster management (e.g. fire monitoring). The data is free for use in research and education.

One of the available products is MODIS data, which has been selected as an example for the presented architecture. MODIS is part of Nasa's Earth Observation System. MODIS provides a set of land surface products, which are described in

[Justice et al. 1998]. In particular, MODIS fire data (MOD14) has been selected for this study and also serves as a basis for other derived products of MODIS.

### 2.2. Technical Setup

This section describes the common technical setup to access GEONETCast products. The setup is depicted in Figure 2 and consists of three steps: receive, store and serve.

The GEONETCast groundstation receives the data from the GEONETCast satellites through a standard TV dish (connected through TV-card) and decodes the received data stream. The GEONETCast Toolbox [Maathuis et al. 2008] acts as a data manager to filter the desired products, which are then stored in a structured way on the data server. The data server is able to manage user access and serves the data through the file system.

It is important to note, that in some setups the data server and the ground receiving station are hosted on the same computer. For the technical setup as used for this implementation these two components are separated on different machines for scalability and maintenance reasons (e.g. the data server has a periodic backup).

Based on this technical setup the architecture is designed and implemented (Section 3). In particular, the data server is the entry point to access the MODIS data for the Fire Web Service.

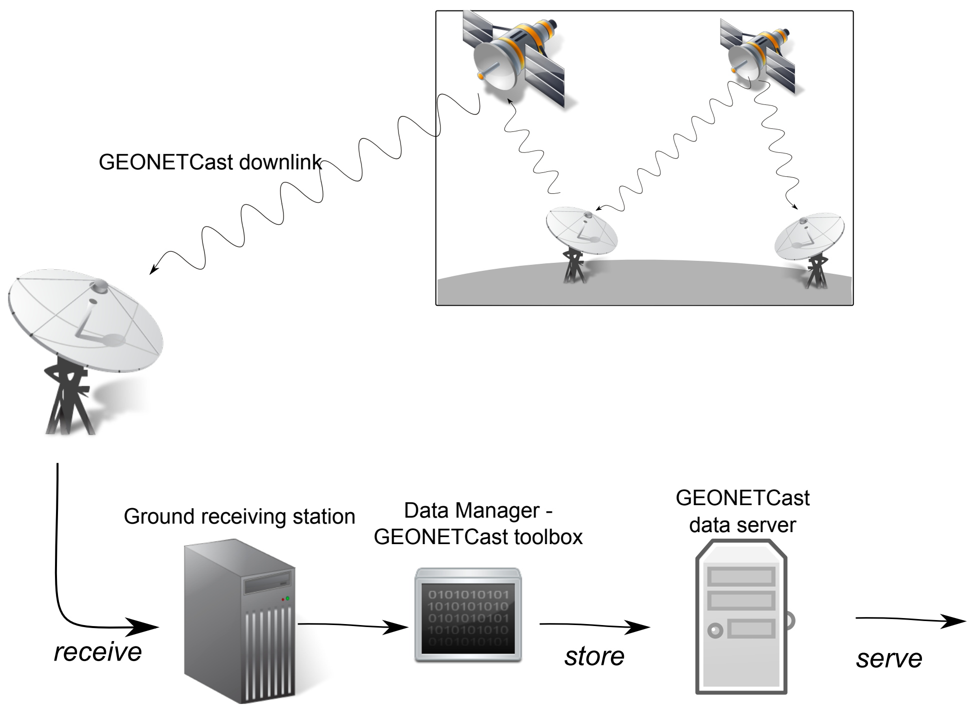


**Figure 2. Overview of the technical setup for receiving GEONETCast products.**

## 3. Fire Web Service Architecture

This section describes the architecture of the Fire Web Service and its implementation. Related work has already been reported by [Davies et al. 2009]. They also used MODIS data for monitoring fires and served this data on the Web. However, their setup involved a lot of manual steps and the access to the data was limited to portrayal (no querying possible). Additionally, the Center for Weather Forecast and Climatic Studies (CPTEC) of the National Institute for Space Research in Brazil (INPE) provides a web portal to access data about fire events. This portal only provides limited querying capabilities and does not allow users to integrate the data available in the portal into other applications. The presented architecture in this article is more comprehensive as it is based on live streaming of MODIS data through GEONETCast using Web Service interfaces. These Web Service interfaces allow users to query the data and also to integrate it into other applications such as Google Earth. Finally, the described implementation is based on common Open Source software and can be re-built with low cost, if required.

### 3.1. Design of the Architecture

For data access on the Web, interoperability enabled by standards plays an important role. Common standards such as established by the Open Geospatial Consortium (OGC) allow users to share data and services out-of-the-box without knowing the implementation details. Therefore, the presented architecture makes excessive use of standards to support the integration of the offered data and services in other applications.

The fire events served by the Fire Web Service are modeled as point-based features. This allows users to integrate low volume data and to query it easily for specific attributes. Both are advantages over the bulky and less structured raster-based data, which requires high network bandwidth for data transmission and advanced tools to extract the required attributes.

For the Fire Web Service and considering the feature-based data for the fire events, the OGC Web Feature Service (WFS) [OGC 2005] has been identified as the appropriate interface. WFS interface allows users to query and access feature data consisting of (multiple) points, lines or polygons. The communication with WFS is based on the Internet Protocol HTTP using an XML-based encoding the so-called WFS Filter encoding. To retrieve specific features from a WFS, the *getFeature* operation is used, which receives messages as WFS Filters. WFS returns feature data (as result of the WFS Filter query) in the Geography Markup Language (GML) or also for instance in KML. KML is the data encoding established and used by Google-based applications. To also create and store new features on a WFS over the Web, a transactional interface has been developed. The additional operations of the so-called WFS-T (T stands for transactional) are insert, update, delete.

The architecture of the Fire Web Service is depicted in Figure 3. The Fire Web Service is accessible via a WFS-T interface as well as through a browser-based application to portray and query the data. Internally, the Fire Web Service uses a spatial database to store the fire data.

The Fire Web Service polls periodically the MODIS data from the GEONETCast data server by searching through the available products and copying the required files locally for further processing. Additionally, the business logic transforms the MODIS data from a raster-based format to the vector-based format and inserts it through the WFS-T interface into the database. The inserted data is then available to the browser-based application of the Fire Web Service but also to external users who can query the fire data

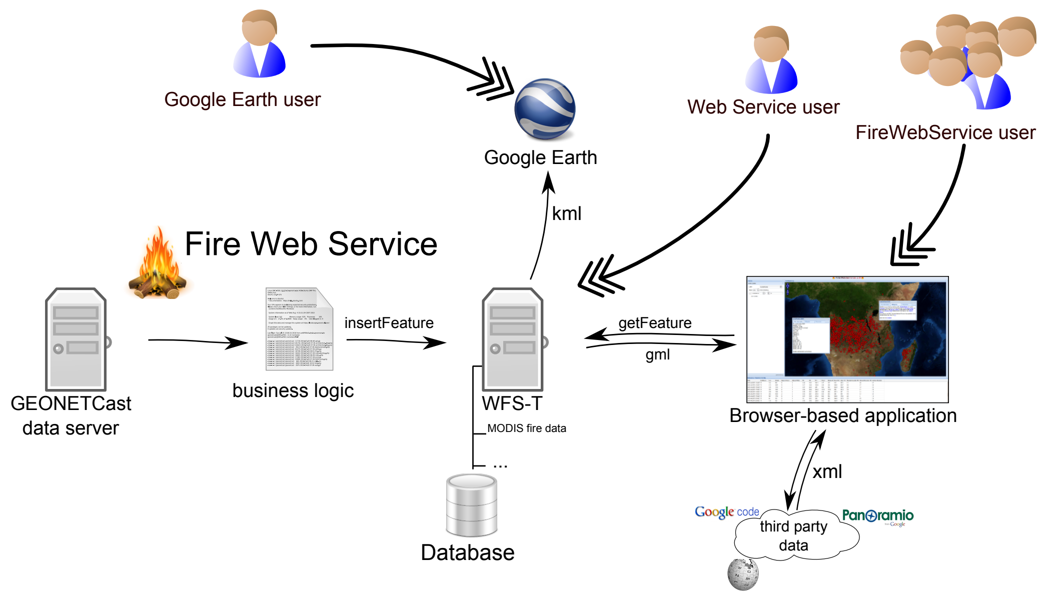


**Figure 3. Architecture of the Fire Web Service.**

directly and integrate it in their application using WFS interface.

The browser-based application links besides the Fire Web Service other additional data sources. In particular, the browser-based application links data from Wikipedia, images from Panoramio and third-party geocoding functionality. This allows users to get comprehensive information for a particular area of interest. The functionality of the browser-based application is described in Section 4.

### 3.2. Implementation

Based on the design of the architecture (Section 3.1), this section describes the implementation. The implementation is based on open source tools to meet the requirements of building a low-cost infrastructure. As the applied open source tools come from different sources and initiatives, standards and interoperability are essential, to make these tools work together.

The Fire Web Service polls the MODIS data from the GEONETCast data server every five minutes by a Cronjob. Every new dataset is transformed from raster data into vector data using the fire detection algorithm for MODIS data [Giglio et al. 2003]. This algorithm transforms the raster data into an ASCII or a shape file. The transformed data is then sent in GML format via a WFS Filter Request to the Fire Web Service using the transactional interface of the WFS. An example of such a request is presented in Listing 1. For this implementation, Geoserver has been chosen as the approriate implementation due to its WFS-T interface and its support for various formats such as KML [Deoliveira 2008]. Geoserver stores the data internally in for instance a PostGIS database[1]. The PostGIS database serves as a backend for the Geoserver to compensate the frequent updates of approximately 15000 fire events per day worldwide.

[1]POSTGIS website: postgis.refractions.net/

**Listing 1. Example of WFS-T insert request to store fire data.**

```
<wfs:Transaction service="WFS" ...">
    <wfs:Insert>
        <de.fws:modisfiredata>
            <de.fws:the_geom>
                <gml:Point>
                    <gml:coordinates decimal="." cs="," ts"">
                    168.143,-16.252</gml:coordinates>
                </gml:Point>
            </de.fws:the_geom>
            <de.fws:appearancedate>2010-06-21T15:13:21Z</de.fws:appearancedate>
            <de.fws:line>1177</de.fws:line>
            <de.fws:sample>506</de.fws:sample>
            <de.fws:confidence>100.0</de.fws:confidence>
            <de.fws:adjcloud>0</de.fws:adjcloud>
            <de.fws:adjwater>0</de.fws:adjwater>
            <de.fws:r2>-1.0</de.fws:r2>
            <de.fws:t21>334.9</de.fws:t21>
            <de.fws:t31>288.4</de.fws:t31>
            <de.fws:power>51.9</de.fws:power>
            <de.fws:mediandt>0.0</de.fws:mediandt>
            <de.fws:meant21>290.3</de.fws:meant21>
            <de.fws:meant31>288.4</de.fws:meant31>
            <de.fws:stddevt21>1.3</de.fws:stddevt21>
            <de.fws:stddevt31>0.5</de.fws:stddevt31>
            <de.fws:stddevdt>1.2</de.fws:stddevdt>
        </de.fws:modisfiredata>
    </wfs:Insert>
</wfs:Transaction>
```

Since the goal was not only to provide access to the data through Web Service interface, but also to visualize it and to combine it with public available data, a browser-based application was created based on OpenLayers[2] and GeoExt[3]. OpenLayers builds a map of the fire data coming from WFS and links it with other sources. The GeoExt library allows us to build a rich web application around those maps including functions such as the built-in support of tables, drop down menus, action bars and pop ups.

The Wiki API[4] was used to provide additional information from Wikipedia about locations with fire events. Every coordinate of a fire is translated on demand into an address via reverse geocoding by the Google Geocoder [5]. The derived address is then used to query Wikipedia for a related article if one is available. If an articles for this specific region is present it is displayed as additional information for the specific fire. The same approach is used for obtaining photos from Panoramio and displaying them in relation to the specific fire.

The described implementation is hosted on a Linux operating system for easy maintenance and high availability. In particular, the Fire Web Service is running on a virtual machine with access to a 2 GHz single core processor with 2 Gigabyte RAM and 7 Gigabyte of storage. Based on our experience, this light-weight configuration provides suf-

[2]OpenLayers website: www.openlayers.org

[3]GeoExt website: www.geoext.org/

[4]Wiki API website: jwikiapi.sourceforge.net/

[5]Google Geocoder website: code.google.com/intl/en-US/apis/maps/documentation/geocoding/

ficient computational performance since the transformation process from MODIS raster data (0.5 MB) to vector data (max. 200 points) takes less than 10 seconds as well as the insert operations into the WFS-T. The transformation and the insert operation are performed every 5 minutes. The system is scalable since every component (e.g. database, WFS-T, business logic) can be hosted on a separate computational node. Some operations such as rendering the map, the requesting geocoding functionality and additional information from wikipedia are performed by the client (without interaction with the Fire Web Service). Consequently, the only task of the Fire Web Service is to transform the data and to provide this data through WFS-T interface. In the future, if the number of users increases, the number of requests to the WFS-T would increase. This can be compensated by a more powerful server configuration hosting the WFS-T. The architecture with its components does not need to be changed.

## 4. Fire Monitoring Use Case

Fire is a natural phenomenon and poses a threat especially when reaching built-up areas. Although it is a natural hazard a rigorous supression leads to even more severe fires. Using fire for cultivating agricultural land has become an established element. However, an excessive application also leads to severe problems. Finally, monitoring such fires is required and different data is already available such as the MODIS fire products [Justice et al. 2002].

For easy and customized access of this data, Web Service technology can be used. To demonstrate the use case, the described architecture of Section 3 is applied to monitor fire events and to integrate the fire data with other third party sources such as Wikipedia or images from for instance Panoramio. Integrating the data with other third party sources is necessary to provide comprehensive information to the user and is possible due to established standards for data and Web Services. In particular, users can access and query the data through a browser-based client and can use additionally Google Earth to combine the data with other sources.

To provide an easy and user-friendly access, a browser-based application has been designed. An example of the browser-based application is depicted in Figure 4. The different parts of the interface are attached to the several tasks, which are described in the following:

a) *Query data* - The user is able to query the fire data available on the Fire Web Service based on its attributes (including time). By querying the temporal dimension, it is possible to not only receive up-to-date data about fire events, but also historic data. Another important attribute is the confidence value of a fire event, which indicates the possibility of a fire. The confidence value is a result of the applied fire detection algorithm performed on the MODIS data.
b) *Map data* - The result of the query can be inspected in the map view, in which the fire events are located on freely-available imagery such as from Blue Marble [Stockli et al. 2005]. Each fire event on the map can be clicked for further details.
c) *Inspect attribute table* - Based on the query, the data can also be inspected using a tabular view.

The fire data can also be integrated into Google Earth by KML format as supported by the WFS interface. An example of such integration is depicted in Figure 5. The

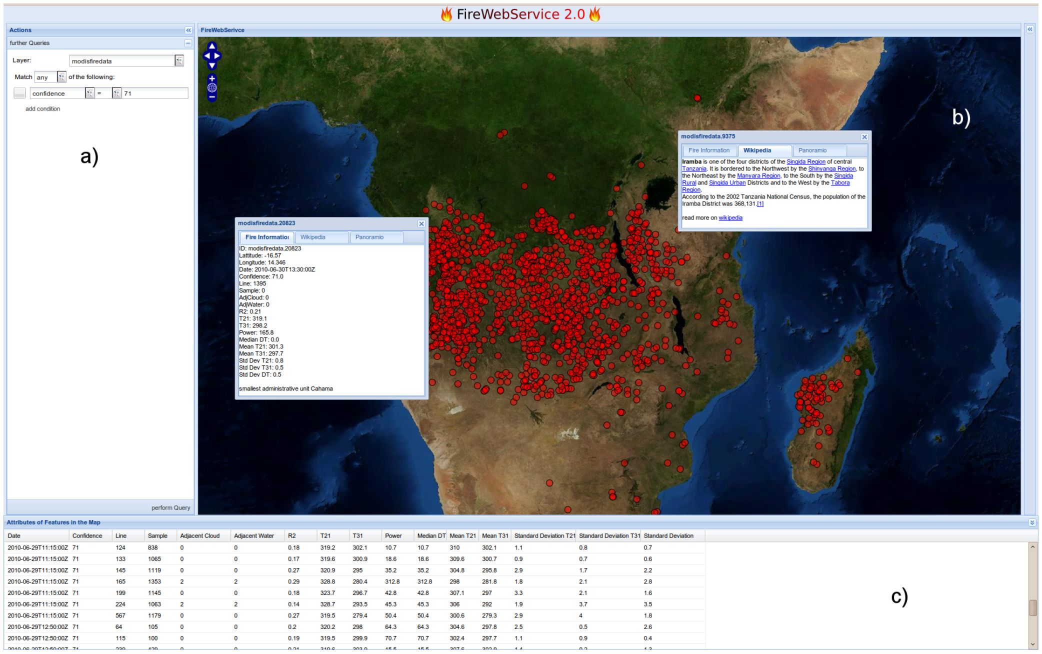


**Figure 4. Screenshot of the browser-based application for the Fire Web Service with map of sub-Sahara region - different views: a) query view, b) map view, c) tabular view.**

attributes of the fire can be inspected directly by the user. However, a query mechanism for the data such as provided by the browser-based application is not available.

## 5. Discussion & Conclusion

This article describes a web-based architecture to access satellite-based products for environmental monitoring such as wild fires. In particular, it describes how data products available from the GEONETCast data stream can be integrated into web-based applications. The architecture is demonstrated by the application of a Fire Web Service and the use case of fire monitoring. As the architecture uses low-cost satellite-based products such as GEONETCast and is implemented using Open Source software, it can be re-built with low cost. Additionally, as this architecture is based on open standards for data and Web Services such as GML, WFS and KML, it can be applied easily in Spatial Data Infrastructures, if required. The presented architecture is not exclusively suitable for infrastructure with web access, but also suitable for the intranet. Therefore, this architecture supports also large organizations in development countries with only limited web access.

The GEONETCast data stream and the presented architecture offer various applications from meteorology and hydrology, which can be realized in the future. Especially, serving the GEONETCast products directly on the Web seems to be promising in the future. *In-situ sensor data* from GEONETCast might become available through the Sensor Web [Botts et al. 2008] and thereby be easily integrated with other in-situ data stemming for instance from local organizations. One example, in which MODIS fire products (not accessed through GEONETCast) are accessed through the Sensor Web, is the Advanced Fire Information System (AFIS) of South Africa. The AFIS use case might be enhanced

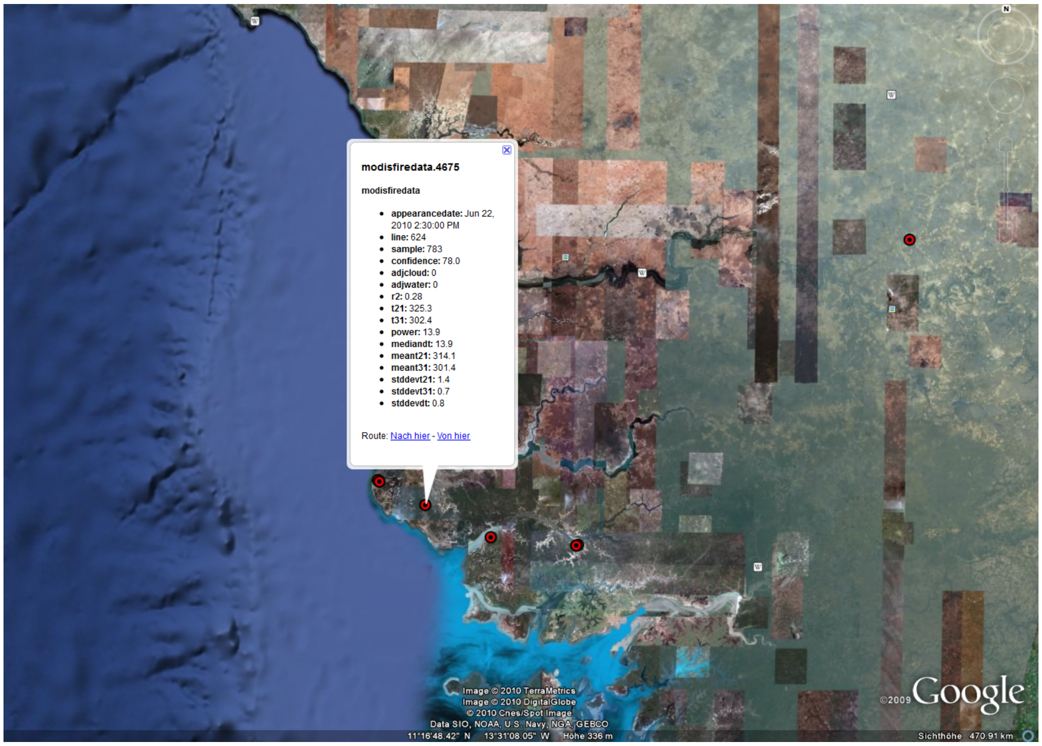


**Figure 5. Screenshot of Google Earth with data from the Fire Web Service.**

with GEONETCast capability to provide real-time fire data [McFerren et al. 2007]. *Remote sensing data* of GEONETCast should be available through OGC Web Coverage Service, which provides customized access to coverage data such as Eumetsat data. Users are then able to receive specific bands of the coverage for a specific area of interest. These customized coverages are then ready for instance to be processed using a web-based geoprocess model, typically accessible as OGC Web Processing Service.

The next step of enabling web-based access is to provide a notification mechanism, whenever the GEONETCast data stream contains a significant event such as a large fire. Therefore an event-based architecture is required to register for specific events (described as patterns) and to notify accordingly.

Overall, data and processing functionality will become available on the Web as Web Services through architectures as described in this article to enhance the sharing of data and functionality. Thereby the web will become a platform to support environmental monitoring and to improve the awareness of global change. In this context, low-cost satellite-based products will play an important role for development countries in the future.

## References

Botts, M., Percivall, G., Reed, C., and Davidson, J. (2008). OGC sensor web enablement: Overview and high level architecture. In *GeoSensor Networks*, number 4540 in Lecture Notes in Computer Science, pages 175–190. Springer, Heidelberg.

Davies, D. K., Ilavajhala, S., Wong, M. M., and Justice, C. O. (2009). Fire Information for Resource Management System: Archiving and Distributing MODIS Active Fire Data. *IEEE Transactions on Geoscience and Remote Sensing*, 47:72–79.

Deoliveira, J. (2008). GeoServer: uniting the GeoWeb and spatial data infrastrucutures. In *Proceedings of the 10th International Conference for Spatial Data Infrastructure*, St. Augustine, Trinidad.

Giglio, L., Descloitres, J., Justice, C. O., and Kaufman, Y. J. (2003). An enhanced contextual fire detection algorithm for modis. *Remote Sensing of Environment*, 87(2-3):273 – 282.

Justice, C. O., Giglio, L., Korontzi, S., Owens, J., Morisette, J. T., Roy, D., Descloitres, J., Alleaume, S., Petitcolin, F., and Kaufman, Y. (2002). The modis fire products. *Remote Sensing of Environment*, 83(1-2):244 – 262.

Justice, C. O., Vermote, E., Townshend, J. R. G., Defries, R., Roy, D. P., Hall, D. K., Salomonson, V. V., Privette, J. L., Riggs, G., Strahler, A., Lucht, W., Myneni, R. B., Knyazikhin, Y., Running, S. W., Nemani, R. R., Wan, Z., Huete, A. R., van Leeuwen, W., Wolfe, R. E., Giglio, L., Muller, J., Lewis, P., and Barnsley, M. J. (1998). The Moderate Resolution Imaging Spectroradiometer (MODIS): land remote sensing for global change research. *IEEE Transactions on Geoscience and Remote Sensing*, 36:1228–1249.

Maathuis, B., Mannaerts, C., and Retsios, B. (2008). The ITC geonetcast-toolbox approach for less developed countries. In *Proceedings of Commission VII*, volume XXXVII of *The international archives of the photogrammetry, remote sensing and spatial information sciences*, pages 1301–1306, Beijing, China. International Society for Photogrammetry and Remote Sensing.

McFerren, G., Roos, S., and Terhorst, A. (2007). Fire alerts for the geospatial web. In Scharl, A. and Tochtermann, K., editors, *The Geospatial Web*, Advanced Information and Knowledge Processing Series. Springer, London, UK.

OGC (2005). Web feature service implementation specification. Implementation specification OGC 04-094.

Stockli, R., Vermote, E., Saleous, N., Simmon, R., and Herring, D. (2005). The blue marble next generation - a true color earth dataset including seasonal dynamics from MODIS. Technical report.

Wolf, L. and Williams, M. (2008). GEONETCast - delivering environmental data to users worldwide. *IEEE Systems Journal*, 2(3):401–405.