

Landgate

Good intro

Landgate is the trading name of the Western Australian Land Information Authority, the statutory authority given charge of maintaining the state's land and property information system ("State Records Office of Western Australia," 2004). The organisation is the inheritor of the mandate of various incarnations of the Department of Lands and Surveys, dating back to the original Survey Office in the 19th Century.

Landgate's

Their role incorporates managing property ownership and transfer records, as well as property valuations to government agencies ("Our story - Landgate," 2015). Vital to society in the connected age, Landgate is Western Australia's leading spatial data agency. Landgate has successfully commercialised spatial data creation and access. Their cumulative efforts considerably lessened their dependence on funding from the state government (*Statement of Corporate Intent 2015/16*, 2015). The success of this strategy has lead to a projected 5% increase in the number of datasets served through the 2015/16 financial year. Ref.

A diagram showing WALIS & Landgate would greatly help.

The Western Australian Land Information System (WALIS) is a partnership between government agencies, the private sector and the community. Their aim is to improve access to location information for the betterment of the Western Australian community (Location Information Strategy Program Coordination Team, 2012). The Shared Location Information Platform (SLIP) is WALIS's spatial data portal, the Western Australian government's Spatial Data Infrastructure (SDI), managed by Landgate. The portal presents datasets owned and maintained by authoritative agencies, standardises data formats and simplifies access.

SLIP Future is WALIS's programme to revamp the original SLIP Enabler portal and infrastructure ("SLIP Future Project," 2014). The custom built and open-source based infrastructure was deemed incapable of handling projected usage and implementing new features. WALIS built a new platform around Google's Software as a Service (SaaS) Google Maps Engine (GME). The new environment offered significant advantages in reliability, scalability and feature set ("SLIP Future

Project," 2014).

In January 2015, Google announced the deprecation of Google Maps Engine ("SLIP Stream," 2015b). Further, they planned to shutter the service entirely by the end of January 2016 ("Overview - Google Maps Engine API (Deprecated)," 2015). Landgate and WALIS were left in search of a new provider for the SLIP Future programme. ESRI aggressively sought the business of GME refugee organisations ("Google & Esri," n.d.) offering free software replacements and membership to business partnership programs. In July 2015, Landgate selected Esri's ArcGIS Server and Portal as the replacement for GME ("SLIP Stream," 2015a). Web services offering datasets in Esri's ArcGIS REST APIs will replace GME's API through a transition period through the end of 2015 and beginning of 2016.

Web Services *Will need a little more intro*

The development of powerful and flexible web services was the foundation that allowed the mobile web to blossom. Web services enable interaction between computer systems over a network. One system may call on another to provide data or a service without requiring a human user to mediate the interaction ("Web Services Glossary," 2004). Mobile devices have limited processing power and storage available, so off-device storage and processing empowers on-device applications.

Service Oriented Architectures (SOA) is a standard governing software design that aims to compose a software product from loosely coupled, and hence replaceable, components (Endo & Simao, 2010). Designers commonly employ this pattern as a method for distributed computing (Palacios, García-Fanjul, & Tuya, 2011). A set of descriptive XML documents, such as Web Services Description Language (WSDL), enable service builders to publish to service registries and then consumers to find and bind to services suited to their needs. Communication is carried out with XML-based messages based on the Simple Object Access Protocol (SOAP), another highly capable

standard.

Representational State Transfer (ReST) services are much easier to develop and consume (Castillo, Bernier, Arenas, Merelo, & Garcia-Sanchez, 2011). The adoption of ReSTful services has led to an explosion of data available on the web, particularly with mobile applications in mind as the end consumers.

Open Geospatial Consortium Web Map Service

A Web Map Service (WMS) composes an image file from server stored vector and raster layers in response to a request in a URL. Despite their focus on returning an image file, WMS operations are still dependent upon XML, especially for GetCapabilities, GetFeatureInfo, error descriptions and layer style code.

The open source and standards driven approach meant that WMS was widely adopted and became a cornerstone web mapping technology. The standard is now quite old; it was last updated to version 1.3.0 in 2006 (la Beaujardiere, 2006).

Open Geospatial Consortium Web Feature Service

Who manages the standards?

A Web Feature Service (WFS) returns geographic vector data in GML (Geographic Markup Language, a derivative of XML) in response to a URL request. It is a more complex and capable service than WMS. If fully deployed, WFS allows external users full create, read, update and delete (CRUD) access to a geographic database (Vretanos, 2005).

The WFS standard is of a similar age to WMS. Version 1.1.0 is most commonly deployed, dating from 2005 (Vretanos, 2005). Version 2.0 from 2010 gained capability and complexity from GML3 and somewhat simpler use from stored queries but was not widely adopted in the years after its release.

Google Maps Engine

Google Inc.'s Google Maps Engine (GME) service enabled the creation of more sophisticated web mapping services on top of the Google Maps interface. Whereas the familiar Google Maps application only allowed one or two additional map layers to be displayed, Google Maps Engine could host many hundreds of datasets in the cloud and perform multi-layer geographic analysis ("Google Maps Engine," 2014). The full version was a true enterprise application and cloud service, with off-the-shelf or bespoke solutions created to suit a client's needs. The scalability and reliability of Google's service were a significant attractor to geospatial providers, such as Landgate.

From the 22nd of November 2014, Google redirected GME website bound traffic to their Google Maps for Work service ("Google Maps for Work - Mapping Software & Applications," 2014), a more streamlined approach to providing enterprise mapping solutions than the previous five separate products.

In a commercial decision, Google Inc. announced the deprecation of the GME API on the 29th of January 2015 ("Overview - Google Maps Engine API (Deprecated)," 2015). Google shuttered the service on the 29th of January 2016.

Esri ArcGIS for Server and ArcGIS ReST API

ArcGIS for Server is Esri's enterprise level product for intra/internet GIS and provisioning web services ("ArcGIS for Server Features," n.d.). ArcGIS^{for} Server has a long development history and has been an established (or entrenched rather) product in geospatial enterprises for many years.

The ArcGIS REST API exposes ArcGIS for Server data and functions as web services. This modern API has been fully developed since 2010 ("The ArcGIS REST API," 2015). ArcGIS^{for} Server's age means it is also fully capable of supporting older web service standards such as

SOAP.

Web Service Evaluation

The OASIS Web Services Quality Factors (Kim et al., 2012) defines six quality factors with 28 sub-categories.

1. Business Value Quality - the value arising from using a web service as compared to the cost.
2. Service Level Measurement Quality - the service responsiveness from a client's point of view, including time and success criteria.
3. Interoperability Quality - the degree to which a service conforms to appropriate standards.
4. Business Processing Quality - the service's reliability for business use considering transmission integrity and integration with other processes.
5. Manageability Quality - management processes to ensure web service quality.
6. Security Quality - the service's ability to prevent intrusion, interception or destruction of the service itself or its messages.

Taken all together these represent all factors that affect a client's decision to consume a web service.

The scope of this study is limited to those aspects of Landgate's service that affect their suitability for use on mobile devices. The business process and value, management, interoperability and security factors cannot be tested with a mobile device. These are more suited to desktop studies and surveys of existing clients. Only Service Level Measurement Quality is within the purview of this study.

The sub-categories include;

1. Response Time - the time interval between the transmission of a request and the receipt of a response. The total time is composed of the time taken for the client to compose the request and decompose the response plus the network transmission time to and from the server plus the time taken for the server to process the request and formulate a response.
2. Maximum Throughput - the maximum number of requests a service can reliably respond to in a unit of time.
3. Availability - the proportion of time the server is operational, the complement of service downtime per measured time.
4. Accessibility - the probability of the web service can be reached when the system is operational, quantified as the number of received acknowledgement messages divided by the total number of requests.
5. Successability - the probability of receiving a successful response to a web service request, the number of responses divided by the number of requests.

We propose to track these factors through a series of frequent but irregularly timed tests from a mobile device deployed in situations common to the mobile network milieu.

Similar Work

Web services are widely studied. However, the scope of applications for web services is broad to say the least. There are, therefore, few studies that examine the intersection of geographic web service performance, mobile device context and an individual, state-level spatial data infrastructure. Following are a cross-section of papers with aims partially

aligned to those of this work.

Hamas, Saad and Abed (2010) compare the performance of SOAP and ReST APIs on mobile devices. The measured criteria are response time and transmission size which predictably favour ReST interfaces.

Their experiment design emulates a mobile device on a desktop computer; further they restrict the simulated mobile network speed. These are useful controls in an experiment designed with a very clear aim of finding which service is faster. But real world complications such as heavy network traffic or poor signal are not addressed as a factor in the outcome. As an example, SOAP's WS-ReliableMessaging protocol may reduce overall transfer time in areas with weak signal by minimising the number of failed message attempts.

Tian et al. (2004) design a server-client system that can optionally compress responses to save the client's download limit or skip compression when the server is under heavy load to minimise timed out requests.

Working in the pre-smartphone era the team simulated an iPAQ Pocket PC, emulating the device on a Pentium III laptop. The laptop emulating the client is connected to the server via WiFi, Bluetooth or a simulated mobile network. To simulate the increased latency and slower connection speed of a GPRS network, they introduce another server that throttles network speed by artificially delaying messages.

Davis, Kimo and Duarte-Figueiredo (2009) focus on OGC Web Map Service (WMS) optimisation for mobile devices. They elaborate a service that combines the multi-layer composition of WMS with the mobile device response speed of AJAX-based web maps such as Google Maps.

Their experiment implemented the proposed service and interacted with it from a custom application deployed on a Nokia N95. Given the focus on minimising data sent and received from the device, the results

vindicated their hypothesis. Unfortunately, the team declined to study response time results due to “severe fluctuations” that they attributed to an overcrowded network. They conclude, by extension not experiment, that smaller volumes of transmitted data would result in faster map interaction overall.

Fowler and Peterson (2012) built a custom iPhone application to test the performance of SOAP and ReST versions of a public transportation web service in Hamburg over a typical working day. They measured response time, data serialisation/deserialisation time and response size on the device itself and returned the results to their own web service. Simple and detailed messages of significantly different response size control whether response time is dependent upon message size. The results, as is common, are given as mean and standard deviation, descriptive statistics without discussion of error responses.

Fowler and Peterson’s methodology called for the mobile user to remain “fixed” while requesting and receiving the response, which we interpret as stationary. This is contrary behaviour for mobile device use. There are countless situations in which a mobile user would be active and moving while concurrently requesting data from a web service.

Provisioning web services from a mobile device faces similar network and device limitations as consuming a service from a mobile device. Nguyen, Jørstad and van Thanh (2008) explored web service performance on an emulated mobile device. While investigating the influence of varied simulated mobile network speeds, they concluded that testing on an actual device would provide ideal settings for their network simulation. Indeed, the subsequent experiment showed considerable differences between emulated and real network speed influence on web service performance. Even after modifying their simulated network speed to approximate real world network speed the difference is significant.

Hussain, Wang and Toure (2014) test the response time and throughput of a variety of real world web services over DSL, WiFi and LAN internet

connections. Results are simplified to descriptive statistics, average, minimum and maximum response time.

Hussain, Wang and Toure discuss some tests with unusually long response times and speculate that it may be due to other web traffic. The specific time or other conditions of these particular tests are not elucidated. In fact, the methodology is not sufficiently detailed to provide all the parameters used in web service requests. Repeating their experiments will likely produce different results.

Yang, Cao and Evans (2013) demonstrate that WMS servers struggle with heavy loads of simultaneous requests. They recorded response times to 1, 5, 10, 30, 50 and 100 concurrent requests to six important WMS servers. They found that response times increased with the number of requests until many servers either blocked all incoming requests to handle the load or simply timed out. They make several recommendations, particularly regarding parallel requests and processing. Most helpfully to this study, a simple progress bar to indicate to a user that their request is still being processed.

The emergence of ReSTful web services engendered many studies comparing performance to entrenched SOAP services. Few experimental designs take advantage of SOAP's inherent advantages, i.e. a message layer and orchestrated distributed computing, such a design would allow SOAP APIs to be compared more favourably with ReST APIs.

Castillo et al. (2011) compare ReST and SOAP service implementations as the intermediary messaging layer for a genetic algorithm and a fitness evaluator. They also present one of the few papers to elaborate the advantages of the older SOAP API standard against a ReST API, but their experiment doesn't build on this. Their proof of concept methodology introduces a useful control, requests via SOAP and ReST send strings of either 100 or 1,000 characters. The proportional time difference between large and small requests controls whether the response time depends upon the amount of data sent and received,

illuminating how much response time overhead is due to the API employed.

Like other researchers, Castillo et al. relate the results of their performance tests as average response time with a margin of error. They discuss accuracy, but this is in terms of their genetic algorithm's accuracy, not the error rate of the web service API.

Kanagasundaram et al. (2012) expose a student database as a resource and perform Create, Read, Update Delete operations over SOAP and ReST web services. The comparison of response time between different operations leads the team to propose a hybrid SOAP and ReST web service that incorporates the security and reliability aspects of SOAP with ReST APIs' ease of development.

The experiment design places the client and server processes on separate cores of a single processor. This works well as an experimental control but comes at the expense of measuring real world performance. Furthermore, the results obtained by Kanagasundaram et al. (2012) are averaged response times from experiments repeated until they achieved a 95% confidence level. But not all are presented in the paper, only a select subset. Presumably messages resulting in errors were excluded from the averaged results.

Expedients such as simulated mobile networks or emulated mobile devices can be more easily understood when considered in the context of testing a production server. Stress testing and automated test case generation, for example, both benefit from a quantity of tests unachievable with a few handheld devices. Services such as Load Impact ("Mobile Performance Testing | Load Impact," n.d.) emulate thousands of mobile users on a variety of devices with a range of network connectivity.

Mobile application testing is a rapidly growing field. Gao et al. (2014) identified four classes of mobile application testing approaches. Emulation-based testing, as seen above, where a device simulator runs

on a desktop computer, is commonplace as emulators ship with most app development environments. Device-based testing runs the application and its services on a range of real devices to find edge case failures. Cloud testing suits large-scale tests scaleable to requirements. Crowd-based testing employs volunteers or contractors to test apps and can reach reasonable scales, they note that tests may not be completely thorough.

Yan et al. (2012) built a cloud testing suite, an approach they called Web Service - Testing-as-a-Service (WS-TaaS). Their experiment showed that cloud tested web services were capable of responding to significantly more requests than one tested on a single desktop computer. Most likely due to the limited bandwidth and processing power of the single testing node.

To build a table would be excellent to give tabular format Web Service Quality and Discovery to these articles.

Automating web service discovery is a much more active field of research with the aim of supporting semantic web development. An application should be able to bind the web service without supervision from the end user. How then, though, should the application choose which web service to employ from the multitude available, also without requiring user intervention. The investigators below are proponents of systematically and automatically applied quality metrics as a basis for deciding which web services should be bound.

Palacios, Garcia-Banjul and Tuya (2011) surveyed Service Oriented Architecture literature to find articles focussed on dynamic binding. 57% of the 33 articles detected faults in web services and thereby excluded non-responsive services.

Orion, Marco and French (2014) reviewed the state of the art in quality of service models for web services, surveying 65 papers written between 2001 and 2012. They showed that most researchers were assessing web services quality in terms of availability (essentially the probability a request will receive a response) with 94% of surveyed papers defining

Citation

the metric. Response time was second place at 83% coverage and accuracy third with 62%.

Wu et al. (2011) propose web service registries evaluate the quality of services they expose. Service downtime, mismatches between catalogued metadata and current capability or inconsistencies when registered in multiple catalogues can lead to the selection of a suboptimal web mapping service. The catalogue service periodically interrogates an OGC WMS testing all operations listed in its GetCapabilities document. Successful tests decrease the frequency of future tests while a lack of response increases the frequency.

Wu et al. model quality factors using a hierarchical tree, proposed by Hanwu Zhang, one of the authors in their Ph.D. dissertation from 2008 (reference not found in English translation). This is a helpful concept for automated quality analysis as branches in the hierarchy can be weighted differently, emphasising categories of metrics (the “leaves”) over others. Unfortunately, Wu et al.’s automated analysis stops after comparing the most recent response time to a weighted average of previously recorded response times. They go on to assess map data quality through a survey method. A GUI program presents returned data to an expert user for assessment and scoring out of five on a series of quality metrics.

Miao, Shi and Cao (2011) build upon Wu et al.’s method of adaptable testing frequency by halving test intervals when the response time is greater than the average of earlier tests. They go further to parameterise the proportion of failed requests. Their process is explained well in a flow chart, omitted from many similar papers.

Miao, Shi and Cao developed a C based program to crawl 100 WMS servers to measure their performance and stability as per their procedure. The conference paper referenced here omitted a table listing the servers assessed. As with Wu et al., the team leaves data quality assessment to a survey of expert users.

Again maybe a table also displaying it.

Acceptance

Park and Ohm (2014) used survey data to construct a technology acceptance model to investigate users' acceptance of mobile mapping applications. They found that acceptance and hence intention to use a mobile mapping service depended to a large degree upon two factors; perceived locational accuracy and processing speed. Park and Ohm defined perceived locational accuracy as how well users envision their location in the map, essentially the degree to which mapped features correspond with a user's mental model of the world and where they are in it.

Conclusion

Our study's contexts are in a state of nearly constant change. Web services have lost complexity due to the dominance of ReST APIs, but the number of services available has grown exponentially. Mobile devices have more processing power, larger batteries and faster connections to the internet. There are whole new categories of mobile devices, such as smartwatches, that change the mobile device landscape. Web service evaluation studies must be continually revisited just to keep pace with change.

Similarly, service providers must prepare for a changing landscape that may not even be partially realised yet. Landgate has evolved and changed direction since early in the decade to build the SLIP Future program. They must continue to adopt new technologies and processes so as to remain relevant to their customers.

In this work we contribute to three aspects of the body of web service evaluation literature.

Firstly, we test services from an actual mobile device in real-world mobile usage situations. Tests are therefore as close to the real mobile experience as possible. This contrasts with the bulk of academic literature that performs tests on desktop computers emulating mobile

devices.

Secondly, we examine more aspects of the interaction between client and server than merely average response time. Examining error responses and the conditions that lead to them will give valuable insight into mobile-specific situations likely to result in errors.

Lastly, our examination of a single service provider, Landgate and WALIS's SLIP service, should produce practical and actionable recommendations to improve its suitability for mobile users.

We will need to tie in all of our metrics etc. to do this & **Bibliography** explain it.

ArcGIS for Server Features. (n.d.). ArcGIS for Server Features.

Retrieved October 3, 2015, from <http://www.esri.com/software/arcgis/arcgisserver/features>

Castillo, P. A., Bernier, J. L., Arenas, M. G., Merelo, J. J., & Garcia-Sanchez, P. (2011, May 25). SOAP vs REST: Comparing a master-slave GA implementation. *arXiv.org*.

Davis, C. A. J., Kimo, Y. J., & Duarte-Figueiredo, F. L. P. (2009). OGC Web Map Service implementation challenges for mobile computers.

2009 17th International Conference on Geoinformatics (pp. 1–6).

IEEE. <http://doi.org/10.1109/GEOINFORMATICS.2009.5293410>

Endo, A. T., & Simao, A. S. (2010). A systematic review on formal testing approaches for web services. *Brazilian Workshop on Systematic and*

....

Fowler, S., Hameseder, K., & Peterson, A. (2012). An Empirical Evaluation of Web System Access for Smartphone Clients. *Journal of*

- Networks*, 7(11), 1–15. <http://doi.org/10.4304/jnw.7.11.1700-1713>
- Gao, J., Bai, X., Tsai, W.-T., & Uehara, T. (2014). Mobile Application Testing: A Tutorial. *Computer*, 47(2), 46–55. <http://doi.org/10.1109/MC.2013.445>
- Google & Esri. (n.d.). Google & Esri. Retrieved September 27, 2015, from <http://www.esri.com/landing-pages/products/google-ip>
- Google Maps Engine. (2014, October 14). Google Maps Engine. Retrieved September 30, 2015, from <https://web.archive.org/web/20141014085033/http://www.google.com/work/mapsearch/products/mapsengine.html>
- Google Maps for Work - Mapping Software & Applications. (2014, November 22). Google Maps for Work - Mapping Software & Applications. Retrieved October 3, 2015, from <http://web.archive.org/web/20141122022312/https://www.google.com/work/mapsearch/>
- Hamad, H., Saad, M., & Abed, R. (2010). Performance Evaluation of RESTful Web Services for Mobile Devices. *Int Arab J E-Technol.*
- Hussain, S., Wang, Z. S., & Toure, I. K. (2014). Performance Analysis of Web Services in Different Types of Internet Technologies. *Applied Mechanics and Materials*, 513-517, 1431–1436. <http://doi.org/10.4028/www.scientific.net/AMM.513-517.1431>
- Kanagasundaram, R., Majumdar, S., Zaman, M., Srivastava, P., & Goel, N. (2012). Exposing resources as Web services: A performance oriented approach (pp. 1–10). Presented at the Performance Evaluation of Computer and Telecommunication Systems (SPECTS),

- 2012 International Symposium on, IEEE.
- Kim, E., Lee, Y., Kim, Y., Park, H., Kim, J., Moon, B., et al. (2012). *Web Services Quality Factors Version 1.0* (pp. 1–29). Retrieved from <http://docs.oasis-open.org/wsqm/WS-Quality-Factors/v1.0/cos01/WS-Quality-Factors-v1.0-cos01.html>
- Ia Beaujardiere, de, J. (2006). *OpenGIS® Web Map Server Implementation Specification* (1st ed.) (pp. 1–85).
- Location Information Strategy Program Coordination Team. (2012, November 1). A Location Information Strategy for Western Australia. Retrieved April 16, 2015, from <http://www.walis.wa.gov.au/projects/location-information-strategy-for-wa/locationstrategyassets/LIS%202012.0%20November%202012.pdf>
- Miao, L., Shi, X., & Cao, S. (2011). A scoring model for evaluating geospatial web services ; A case study of OGC Web Map Service (pp. 1–4). Presented at the 2011 19th International Conference on Geoinformatics, IEEE. <http://doi.org/10.1109/GeoInformatics.2011.5980738>
- Mobile Performance Testing | Load Impact. (n.d.). Mobile Performance Testing | Load Impact. Retrieved October 11, 2015, from <https://loadimpact.com/mobile-testing>
- Nguyen, T. T., Jørstad, I., & Thanh, D. V. (2008). Security and Performance of Mobile XML Web Services (pp. 261–265). Presented at the Fourth International Conference on Networking and Services (icns 2008), IEEE. <http://doi.org/10.1109/ICNS.2008.41>

- Oriol, M., Marco, J., & Franch, X. (2014). Quality models for web services: A systematic mapping. *Information and Software Technology*, 56(10), 1167–1182. <http://doi.org/10.1016/j.infsof.2014.03.012>
- Our story - Landgate. (2015, August 28). Our story - Landgate. Retrieved September 24, 2015, from <http://www0.landgate.wa.gov.au/about-us/our-story>
- Overview - Google Maps Engine API (Deprecated). (2015, February 18). Overview - Google Maps Engine API (Deprecated). Retrieved April 15, 2015, from <https://developers.google.com/maps-engine/>
- Palacios, M., García-Fanjul, J., & Tuya, J. (2011). Testing in Service Oriented Architectures with dynamic binding: A mapping study. *Information and Software Technology*, 53(3), 171–189. <http://doi.org/10.1016/j.infsof.2010.11.014>
- Park, E., & Ohm, J. (2014). Factors influencing users' employment of mobile map services. *Telematics and Informatics*, 31(2), 253–265. <http://doi.org/10.1016/j.tele.2013.07.002>
- SLIP Future Project. (2014). SLIP Future Project. Retrieved September 27, 2015, from https://www2.landgate.wa.gov.au/c/document_library/get_file?uuid=18853442-51bf-4a11-9ccc-76a61d891e51&groupId=10136
- SLIP Stream. (2015a, July). SLIP Stream. Retrieved August 12, 2015, from <http://slip.landgate.wa.gov.au/SiteAssets/Pages/SLIPStream/SLIPStream%20July%202015.pdf>

- SLIP Stream. (2015b, January 31). SLIP Stream. Retrieved April 16, 2015, from <http://slip.landgate.wa.gov.au/SiteAssets/Pages/SLIPStream/SLIPStream%20January%202015.pdf>
- State Records Office of Western Australia. (2004, August 23). State Records Office of Western Australia. Retrieved September 24, 2015, from http://aeon.sro.wa.gov.au/Investigator/Details/Agency_Detail.asp?Entity=Global&Search=lands%20and%20surveys&Op=All&Page=1&Id=1576&SearchPage=Global
- Statement of Corporate Intent 2015/16*. (2015). *Statement of Corporate Intent 2015/16*. www0.landgate.wa.gov.au.
- The ArcGIS REST API. (2015, August 31). The ArcGIS REST API. Retrieved October 1, 2015, from <http://resources.arcgis.com/en/help/arcgis-rest-api/index.html>
- Tian, M., Voigt, T., Naumowicz, T., Ritter, H., & Schiller, J. (2004). Performance considerations for mobile web services. *Computer Communications*, 27(11), 1097–1105. <http://doi.org/10.1016/j.comcom.2004.01.015>
- Vretanos, P. P. A. (2005). *Web Feature Service Implementation Specification* (1st ed.) (pp. 1–131).
- Web Services Glossary. (2004, February 11). Web Services Glossary. Retrieved October 4, 2015, from <http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/#webservice>
- Wu, H., Li, Z., Zhang, H., Yang, C., & Shen, S. (2011). Monitoring and evaluating the quality of Web Map Service resources for optimizing

map composition over the internet to support decision making.

Computers & Geosciences, 37(4), 485–494. <http://doi.org/10.1016/j.cageo.2010.05.026>

Yan, M., Sun, H., Wang, X., & Liu, X. (2012). WS-TaaS: A Testing as a Service Platform for Web Service Load Testing. Presented at the ICPADS '12: Proceedings of the 2012 IEEE 18th International Conference on Parallel and Distributed Systems, IEEE Computer Society.

Yang, P., Cao, Y., & Evans, J. (2013). Web Map Server Performance and Client Design Principles. *GIScience & Remote Sensing*, 44(4), 320–333. <http://doi.org/10.2747/1548-1603.44.4.320>

