

**Department of Spatial Sciences**

**Landgate API Test**

Testing SLIP's Spatial Web Services  
from a Mobile Device

**Aiden John Price**

**This research project report is presented for the Degree of  
Master of Science (Geospatial Science)  
of  
Curtin University**

**June 2016**

## ABSTRACT

Landgate is Western Australia's premier spatial data provider, the lead organisation and partner of the West Australian Land Information System (WALIS). The Shared Location Information Platform (SLIP) is the WA Government's spatial data portal and the implementation of WALIS's programme to improve access to location information. Previously dependent upon Google Map Engine spatial server infrastructure, Landgate and SLIP had to pivot rapidly to Esri spatial servers to avoid loss of service due to GME's decommissioning.

Academic writing on spatial web service testing naturally aims to test under controlled conditions, to eliminate variables of network connectivity or device speed. Controls tend to poorly reflect real world mobile device usage. This research eschews controls for the greater part, preferring to gather data on the test environment at test time.

Further, academia tends to focus on response time as the main metric of service performance, an objective measure. Few papers consider whether the data returned are correct. Mobile downloads can be interrupted by lost signal, resulting in an incomplete response. Unreliable or inconsistent response data decreases a user's desire to reuse and recommend a mapping application. Were such a situation to occur repeatedly fewer application developers would employ SLIP's services in their mobile applications and fewer end users would access Western Australia's spatial information.

LandgateAPITest is a testing suite composed of an iOS application for frontline mobile device testing and a Google App Engine web application for storage, analysis and presentation of charted results. The suite combines response time measurements with errors in responses to paint a broader picture of Landgate's servers' suitability for mobile devices.

Deploying LandgateAPITest against SLIP's GME, OGC and Esri spatial web service endpoints confirms the findings of earlier studies and finds that the mobile network is the biggest factor in performance and quality of responses.

The end result is a set of mobile suitability metrics. The frequency distribution of response time across a number of web service request types demonstrates performance. The percentage of requests with inconsistent response data compared to a reference measures quality.

Providing these metrics to the SLIP team will aid Landgate and WALIS in objectively measuring progress towards their goal of enabling access to location information.

## ACKNOWLEDGEMENTS

First and foremost, the author acknowledges the unfailing support of his wife and family. This work would not have been possible, or as thorough, without their aid.

Dr David McMeekin has been a most encouraging supervisor. His enthusiasm for the subject is infectious.

The iOS brains trust at Curtin University, Tristan Reed and Jeremy Siao Him Fa, helped cut through knotty iOS and Swift problems.

## TABLE OF CONTENTS

ABSTRACT.....	i
TABLE OF CONTENTS.....	iv
LIST OF FIGURES .....	vi
LIST OF TABLES.....	vii
1 INTRODUCTION .....	1
2 BACKGROUND .....	1
2.1 Web Services .....	1
2.2 Spatial Web Services .....	2
2.2.1 Open Geospatial Consortium Web Map Service .....	2
2.2.2 Open Geospatial Consortium Web Feature Service .....	2
2.2.3 Google Maps Engine.....	3
2.2.4 Esri ArcGIS for Server and ArcGIS REST API .....	3
2.3 Landgate.....	4
3 LITERATURE REVIEW .....	6
3.1 Related Work .....	6
3.2 Web Service Quality and Discovery.....	10
3.3 Web Service Evaluation.....	12
3.4 Acceptance.....	13
4 MATERIALS AND METHODS.....	15
4.1 Generalised Workflow .....	15
4.2 Data Model and Structures.....	18
4.2.1 TestCampaign .....	18
4.2.2 TestMaster.....	20
4.2.3 ReferenceObject.....	22
4.2.4 Vector.....	23
4.2.5 CampaignStats .....	24
4.3 iOS Mobile Application.....	25
4.3.1 Mobile Application Design Principles.....	25
4.3.2 iOS Application Architecture .....	26
4.3.3 Swift Open Source Packages .....	28
4.4 Google Apps Engine Web Service .....	29
4.4.1 Web Service Design Principles.....	29

4.4.2	Python Application Architecture .....	30
4.4.3	Python Open Source Packages.....	30
4.5	Other Applications Deployed .....	31
4.5.1	Paw.....	31
4.5.2	Atom .....	32
4.5.3	Xcode .....	32
5	RESULTS .....	33
5.1	Test Regime .....	33
5.2	Test Device Hardware.....	37
5.3	TestEndpoint Successes and Failures .....	38
5.4	Test Results by Response Time .....	40
5.5	Test Results by Distance Device Travelled .....	49
6	DISCUSSION .....	52
7	RECOMMENDATIONS .....	56
8	FUTURE WORK.....	57
9	CONCLUSIONS .....	59
	REFERENCES .....	61
	APPENDIX A Web Service GET Requests .....	66
	APPENDIX B Web Service POST Requests .....	74
	APPENDIX C List of Abbreviations.....	80
9.1	HTTP Method Names .....	82

## LIST OF FIGURES

Figure 2.1 - Organisation of Landgate and WALIS, adapted from Location Information Strategy Program Coordination Team, 2012 .....	4
Figure 4.1 - LandgateAPITest generalised workflow .....	16
Figure 4.2 - LandgateAPITest data model class diagram .....	19
Figure 4.3 - Typical Model-View-Controller (MVC) design pattern .....	26
Figure 4.4 - LandgateAPITest's modified MVC design pattern incorporating singleton, state machine controller classes as a model management layer .....	26
Figure 4.5 - LandgateAPITest mobile application state machine UML diagram .....	27
Figure 5.1- Sydney Test Heat Map, basemap tiles copyright OpenStreetMap Contributors .....	35
Figure 5.2 - Bathurst Test Heat Map, basemap tiles copyright OpenStreetMap Contributors .....	36
Figure 5.3 - Townsville Test Heat Map, basemap tiles copyright OpenStreetMap Contributors .....	37
Figure 5.4 - TestEndpoints successful (labelled "True") and failed on device (labelled "False") .....	38
Figure 5.5 - Percentage of Vectors with failed reference checks (labelled "False") ..	40
Figure 5.6 - Percentage of Vectors by test type N.B. smaller categories removed for clarity .....	41
Figure 5.7 - A subset of test types where the request calls for either a small sized response or a larger one comparing their response times .....	43
Figure 5.8 - A subset of test types which call upon the server to limit results by a function comparing the distribution of their response times .....	44
Figure 5.9 - Percentage of Vectors by response data type .....	45
Figure 5.10 - Response time distribution by response data type .....	46
Figure 5.11 - Percentage of Vector objects by server type .....	46
Figure 5.12 - Response time distribution by server type .....	47
Figure 5.13 - Percentage of Vectors by HTTP Method (GET and POST) .....	48
Figure 5.14 - HTTP Method (GET and POST) response time distributions in box plot .....	49
Figure 5.15 - A scatterplot of distance device travelled (metres) versus response time (seconds) for each Vector object .....	50

## LIST OF TABLES

Table 5.1 - Subtest counts .....	33
Table 5.2 - Test device characteristics captured by LandgateAPITest iOS application .....	38
Table 5.3 - Percentage of reference checks successful by test type.....	39
Table 9.1 - Esri GET Requests .....	67
Table 9.2 - GME GET Requests .....	69
Table 9.3 - OGC GET Requests .....	71
Table 9.4 - Esri POST Requests .....	75
Table 9.5 - OGC POST Requests .....	77
Table 9.6 - List of Abbreviations.....	81
Table 9.7 - HTTP Method Names.....	82



## 1 INTRODUCTION

Landgate is Western Australia's cadastre authority and foremost spatial data agency. It is the leading organisation in the Western Australian Land Information System (WALIS) and its open data portal; the Shared Location Information Platform (SLIP). As SLIP was built on Google Maps Engine (GME), Google Inc.'s closure of the service forced Landgate and WALIS into a period of rapid change. Esri ArcGIS for Server infrastructure has replaced GME as Landgate's production spatial servers.

In this work, a testing suite was built consisting of a mobile application for front-line testing and a web service for analysis, named LandgateAPITest. The suite was applied to Landgate's spatial server infrastructure before and after the pivot away from GME.

The use of a mobile device as a test platform and its deployment in real world conditions is a departure from much of the academic literature on spatial web service testing. LandgateAPITest eschews experimental controls as found in related works, such as mobile devices simulated on desktop computer hardware. Further, much academic writing reports results as a few descriptive statistics of web service request response time. LandgateAPITest aims to inspect response data for returns inconsistent with reference data sets.

The testing suite provides a set of measurements as its outputs, frequency distributions of response time and a percentage of responses with inconsistent data, termed mobile suitability metrics. These are aligned with a technology acceptance model related in other works such that improvement in these scores should occasion greater user acceptance of SLIP's spatial web services. Through this work, it is hoped to uncover practical and actionable recommendations which can aid Landgate in improving their service to the community.

This study exists in the context of real world limitations, constrained budget, mobile devices and time. The total funds available amounted to zero Australian Dollars. Only a single mobile device could be effectively deployed in the time frame. The pre-set mobile data usage allowance capped the amount of testing possible within a

given month before excess usage charges began eroding the previously described funding.

It is outside the scope of the LandgateAPITest suite to perform load testing (also known as a stress testing). The primary focus is on testing from a mobile device. This restricts the number of concurrent requests which could be launched at Landgate's infrastructure to a figure small enough that it would be unlikely to cause any difficulties. Conversely, such a method would have a significant negative affect on the device's battery life and the owner's mobile data plan.

Hereafter in this section, the organisations and services discussed throughout the paper are clarified. The Literature Review delves into the background of spatial web service testing and related works. The Materials and Methods section outlines the general workflow for the testing suite, it then explains the inner workings of its data model and the mobile and web applications. The Results section shows the test outcomes in charts and maps generated by the web application. The Discussion section places the results in context and leads to Recommendations and a Future Work section which details possible improvements.

## 2 BACKGROUND

### 2.1 Web Services

Web services enable interaction between computer systems over a network (“Web Services Glossary,” 2004). One system may call on another to provide data or a service without requiring a human user to mediate the interaction. 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 easy 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 end consumers.

REST services use HTTP/HTTPS as their transport layer (Castillo et al., 2011). This permits them to employ the standard HTTP response codes to indicate the success of a request. When a server is able to successfully respond to a request it sends the integer 200 along with the response headers and response body payload. The server may reject an improperly formatted request; these are the 400 series response codes. Examples include the often seen 404 code for a missing resource or 403 for failed authorisation. A server-side fault that prevents a proper response is assigned a 500 series response code, such as the catch all 500 Internal Server Error code. Less frequently seen, the 300 series codes indicate that the sought resource has been moved and the requester is directed to the new location.

It should be noted that SOAP messages may use many more transport layers than REST services (Castillo et al., 2011) and as such cannot rely singularly on the HTTP response codes to indicate request failure. As a consequence, they often include error or exception details in the response body while setting the response code to the contrary 200 success value.

## **2.2 Spatial Web Services**

Spatial data are a computer representation of any information with a location dimension (Huisman & de By, 2009). It digitally models the real world. Web services that deliver location data over a network or perform geospatial functions are spatial web services.

### **2.2.1 Open Geospatial Consortium Web Map Service**

The Open Geospatial Consortium (OGC) is an international group of industry, government, academic and community representatives whose aim is to improve business processes through the integration of location data (Reed, 2011). The OGC direct their main supporting efforts towards the creation of location data and service standards and strategies.

A Web Map Service (WMS) composes an image file from server stored vector and raster layers in response to a request in a URL. The open source and standards driven approach meant that WMS was widely adopted and became a cornerstone for web mapping technology. The standard is now quite old; the documentation for version 1.3.0 was finalised in 2006 (la Beaujardiere, 2006).

### **2.2.2 Open Geospatial Consortium Web Feature Service**

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 grants 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.

### 2.2.3 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 of GME was an 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 web site 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.

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 shut the service on the 29th of January 2016.

### 2.2.4 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.).

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 capable of supporting older web service standards such as SOAP.

### 2.3 Landgate

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 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. The success of this strategy has led to a projected 5% increase in the number of datasets served through the 2015/16 financial year (*Statement of Corporate Intent 2015/16*, 2015).

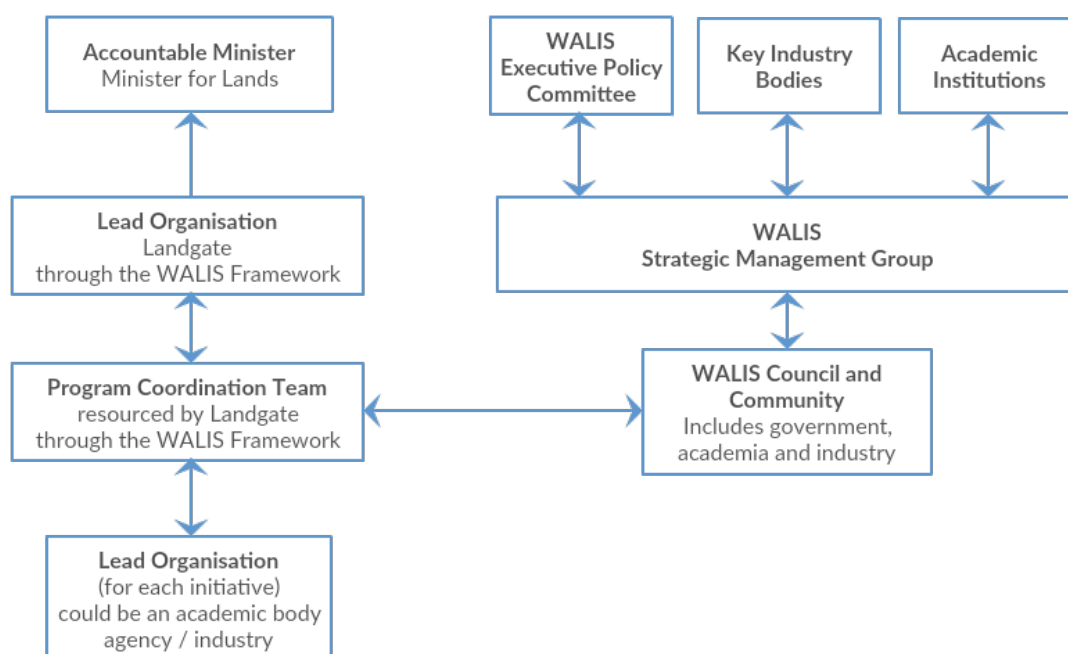


Figure 2.1 - Organisation of Landgate and WALIS, adapted from Location Information Strategy Program Coordination Team, 2012

The Western Australian Land Information System (WALIS) is a partnership between government agencies, the private sector, academia and the community (see figure

2.1). 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 older 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 shut 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 replaced GME's API through a transition period through the end of 2015 and beginning of 2016.

Landgate's mandate to provide spatial data services through SLIP has been somewhat derailed by the deprecation of Google Maps Engine. The work presented within looks at developing a testing suite to aid Landgate and WALIS in measuring their success against SLIP's goals. Related work, explored below, is taken as a guide to design and implementation, detailed in the Materials and Methods section.

### 3 LITERATURE REVIEW

#### 3.1 Related Work

Web services are widely studied (Ioini Nabil, 2015; Qiu, Li, Ji, & Leung, 2015; Tahir, Tosi, & Morasca, 2013). However, the scope of applications for web services is broad. There are, therefore, few studies that examine the intersection of geographic web service performance, mobile device context and a single, 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) compared the performance of SOAP and REST APIs on mobile devices. The measured criteria were response time and transmission size which predictably favoured REST interfaces.

Their experiment design emulated a mobile device on a desktop computer; further, they restricted the simulated mobile network speed. These are useful controls in an experiment designed with a very clear aim of finding which service is faster. 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) designed a server-client system that could 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 was connected to the server via Wi-Fi, Bluetooth or a simulated mobile network. To simulate the increased latency and slower connection speed of a GPRS network, they introduced another server that throttled network speed by artificially delaying messages.

Davis, Kimo and Duarte-Figueiredo (2009) focussed on OGC Web Map Service (WMS) optimisation for mobile devices. They elaborated a service that combined 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 concluded, by extension not experiment, that smaller volumes of transmitted data would result in faster map interaction overall.

Fowler, Hameseder 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 controlled whether response time was dependent upon message size. The results, as is common, were given as mean and standard deviation, descriptive statistics without discussion of error responses.

Fowler, Hameseder and Peterson's (2012) methodology called for the mobile user to remain "fixed" while requesting and receiving the response. This is interpreted to mean stationary, 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 was significant.

Hussain, Wang and Toure (2014) tested the response time and throughput of a variety of real world web services over DSL, Wi-Fi and LAN internet connections. Results were simplified to descriptive statistics, average, minimum and maximum response time. They discussed some tests with unusually long response times and speculated that it may be due to other web traffic. The specific time or other conditions of these particular tests were not elucidated. In fact, the methodology is not sufficiently detailed to provide all the parameters used in web service requests. Repeating their experiments would likely produce different results.

Yang, Cao and Evans (2013) demonstrated 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 made several recommendations, particularly regarding parallel requests and processing. Most helpful to this study, a simple progress bar to indicate to a user that their request is still being processed.

Recent work by Cannata, Antonovic and Molinari (2014) load tested the transalpine HELIDEM project's OGC web services. They measured response time as they programmatically overloaded the GeoServer system using the Locust open source Python testing library.

Tantalisingly, Cannata, Antonovic and Molinari's (2014) mentioned that Locust is capable of reporting testing tasks as failed even if the server returns a 200 "Success" response code with an exception in the response data. Unfortunately, they did not go on to explain this capability, nor did their testing uncover any exceptions which Locust could have revealed to them.

The emergence of RESTful web services engendered many studies comparing performance to entrenched SOAP services (Castillo et al., 2011; Kanagasundaram,

Majumdar, Zaman, Srivastava, & Goel, 2012). By focussing on response performance, 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) compared REST and SOAP service implementations as the intermediary messaging layer for a genetic algorithm and a fitness evaluator. They also presented one of the few papers to elaborate the advantages of the older SOAP API standard against a REST API, but their experiment did not build on this. Their proof of concept methodology introduced a useful control, requests via SOAP and REST sent strings of either 100 or 1,000 characters. The proportional time difference between large and small requests controlled 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. (2011) related the results of their performance tests as average response time with a margin of error. They discussed accuracy, but this concerned their genetic algorithm's accuracy, not the error rate of the web service API.

Kanagasundaram et al. (2012) exposed an ersatz student database as a resource and performed Create, Read, Update Delete operations over SOAP and REST web services. The comparison of response time between different operations lead the team to propose a hybrid SOAP and REST web service that incorporated the security and reliability aspects of SOAP with REST APIs' ease of development.

The experiment design placed the client and server processes on separate cores of a single processor. This worked well as an experimental control but came at the expense of measuring real world performance. Furthermore, the results obtained by Kanagasundaram et al. (2012) were averaged response times from experiments repeated until they achieved a 95% confidence level. However, not all were 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 scalable to requirements. Crowd-based testing employs volunteers or contractors to test apps and can reach reasonable scales; they noted that tests may not be 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 from a single desktop computer. They concluded it was most likely due to the limited bandwidth and processing power of the single testing node.

### **3.2 Web Service Quality and Discovery**

Automated web service discovery aims to support semantic web development (D'Mello & Ananthanarayana, 2010; Palacios et al., 2011). An application should be able to bind a 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 discuss 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 regarding availability (essentially the probability a request will receive a response) with 94% of surveyed papers defining the metric. Response time was second place at 83% coverage and accuracy third with 62%.

Wu et al. (2011) proposed 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 could lead to the selection of a suboptimal web mapping service. The catalogue service periodically interrogated an OGC WMS testing all operations listed in its GetCapabilities document. Successful tests decreased the frequency of future tests while a lack of response increased the frequency.

Wu et al. (2011) modelled 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 stopped after comparing the most recent response time to a weighted average of previously recorded response times. They went on to assess map data quality through a survey method. A GUI program presented returned data to an expert user for assessment and scoring out of five on a series of quality metrics.

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

Miao, Shi and Cao (2011) developed a C based program to crawl 100 WMS servers to measure their performance and stability as per their procedure. The paper

referenced here omitted a table listing the servers assessed. As with Wu et al. (2011), the team left data quality assessment to a survey of expert users.

### **3.3 Web Service Evaluation**

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

- (a) Business Value Quality - the value arising from using a web service as compared to the cost.
- (b) Service Level Measurement Quality - the service responsiveness from a client's point of view, including time and success criteria.
- (c) Interoperability Quality - the degree to which a service conforms to appropriate standards.
- (d) Business Processing Quality - the service's reliability for business use considering transmission integrity and integration with other processes.
- (e) Manageability Quality - management processes to ensure web service quality.
- (f) 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.

The sub-categories include:

- (a) 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.

- (b) Maximum Throughput - the maximum number of requests a service can reliably respond to in a unit of time.
- (c) Availability - the proportion of time the server is operational, the complement of service downtime per measured time.
- (d) 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.
- (e) Successability - the probability of receiving a successful response to a web service request, the number of responses divided by the number of requests.

This research proposes 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.

### **3.4 Acceptance**

Park and Ohm (2014) surveyed mobile map users 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.

Park and Ohm (2014) 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. This study considers that requests for map data that fail due to errors or return mismatching or incomplete data are a direct hindrance to user acceptance. A mobile map application that featured missing map tiles or incomplete geographic feature data prevents a user from determining their location in the map with respect to their real world situation.

Park and Ohm's (2014) definition of service and display quality is an extension of earlier general definitions of service quality; "the degree of general performance of

an information system and related services." As discussed above, quality as measurable in the context of this work includes response time and likelihood of request success.

A LandgateAPITest suite campaign, explained in the following section, produces mobile suitability metrics aligned with both factors as its outputs.



## 4 MATERIALS AND METHODS

### 4.1 Generalised Workflow

The mobile application user chooses to initiate a test against the Landgate servers. There are several types of test, each offers a different combination of subtests on a variety of Landgate endpoints. All subtests are enqueued and assuming that several preconditions are met the testing begins.

Testing proceeds in a cycle (illustrated in the first half of figure 4.1):

- (a) First, a LocationTest determines the device's latitude and longitude
- (b) A NetworkTest queries the device's connection to the mobile network
- (c) A PingTest checks the ping time to a well-known endpoint other than Landgate
- (d) Then the device sends one of the pre-ordered requests to Landgate's servers and captures the response data in an EndpointTest

The cycle repeats until there are no more EndpointTests in the queue.

The client/server interaction has several failure points. It is important for this work to record failed requests as these affect a service's suitability for mobile network traffic. The device may not reach the endpoint at all due to a total lack of connectivity (recorded as response code "0") or if a TestEndpoint is interrupted before its conclusion. Should the device fail to reach the server, or the server respond with a 400 or 500 code, the iOS LandgateAPITest app records the TestEndpoint as a failure. These are referred to as "On Device Failures".

Note that 300 series response codes, the resource moved or redirect codes, are not considered failures. The test continues to the redirected resource where it will eventually earn a 200 success code or a failure code.

Immediately upon the test commencing, the device records the current date and time as a Unix time value, the number of seconds since 00:00 on the first of January 1970. Similarly, when the test concludes (successfully or otherwise) the device records the

current date and time again. The total response time is the difference between these two time values.

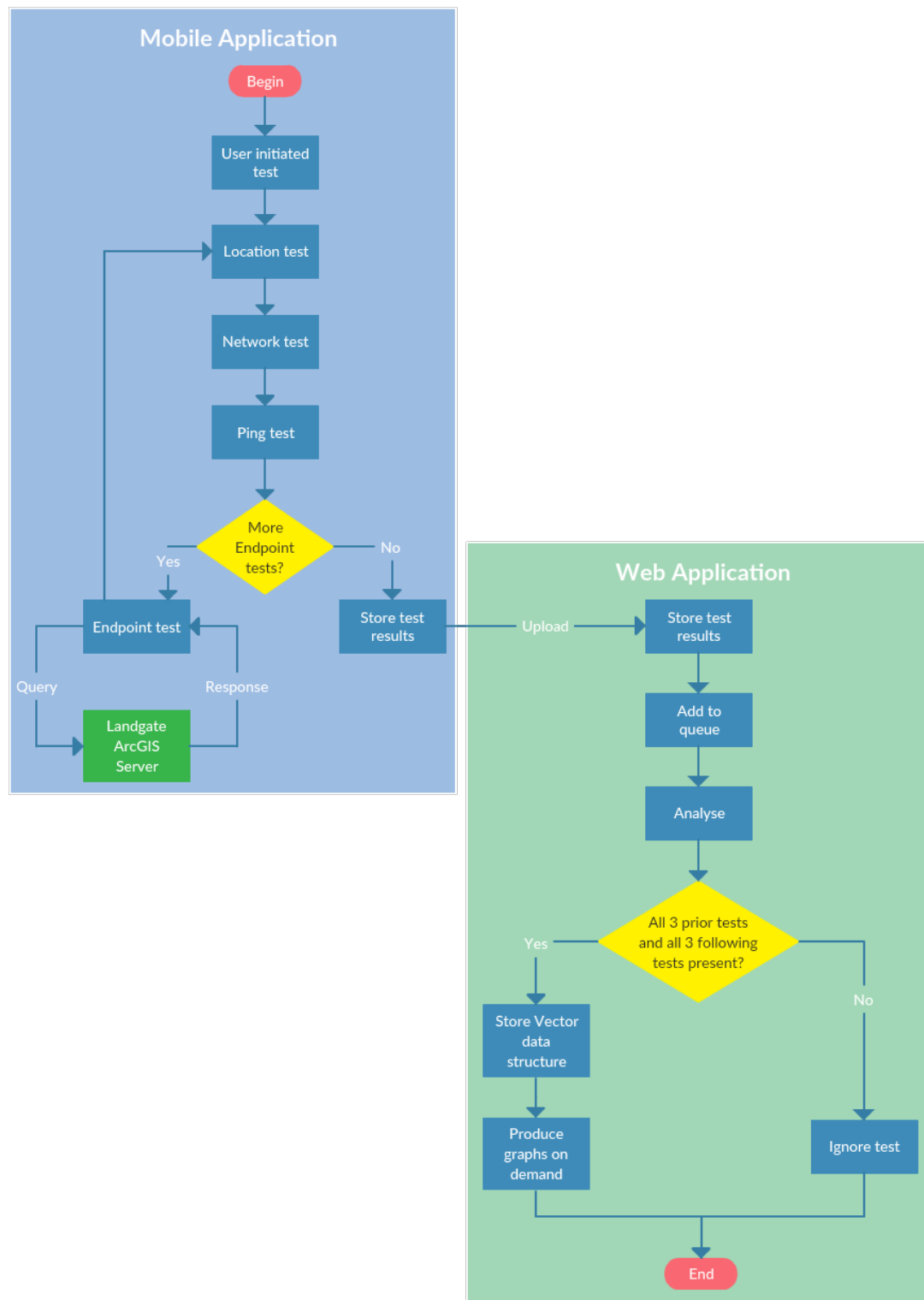


Figure 4.1 - LandgateAPITest generalised workflow

After all tests in the queue are complete the device stores all tests, their details and response data to a local database. The mobile app can query this database to display results to the user.

The user may choose to upload the results to the LandgateAPITest web app at a later time, ideally when the device is connected to Wi-Fi. It is not desirous to immediately upload the result as it could double the data usage on the user's mobile data plan. Should an upload fail, the user may retry as many times as they wish.

LandgateAPITest's web application conducts the analysis on all results for each campaign of testing. When each test is successfully stored in the web app's database, it adds a task to the app's task queue. When the app has spare processing capacity, the queued task fires a request to the app to analyse the result.

In the analysis, the web app attempts to create a new Vector object to contain the results of the analysis. It requires the LocationTest, NetworkTest and PingTest results from directly before and the same from directly after each EndpointTest. Given the sequence of location, network, ping and endpoint tests each Vector shares its following three results with the next EndpointTest.

A gateway decision requires all six tests be present. Should one be missing the analysis is aborted and the result disregarded.

Otherwise, the web app proceeds to check the EndpointTest's response data against a set of reference data. These are exemplar responses to each request which are assumed to be the "true" data.

If the response received by the iOS app is identical to the reference response, then the entire test is considered successful. The iOS application may assume a test is successful given it receives a 200 response code from the Landgate server. Often though OGC servers will respond with a 200 code but send exception text in place of the response data. Tests that fail at this stage are referred to as "Reference Check Failures" and have an appropriate flag set on the record on the web application database.

During execution of the Analyse function, the web app records the percentage of reference check tests successful for each test type. When a test type returns less than 5% successful reference checks it is assumed that there is a process error or an incorrect reference object and disregard the test type entirely. These tests are flagged False for their "ReferenceCheckValid" property to allow them to be filtered out.

The resultant Vector objects are the basis for all analysis and graphical representation. The web application produces pie charts of the various test categories and graphs of response time or distance travelled by category. Each graph is available from the Graph endpoint and responds with the latest information in the database.

## **4.2 Data Model and Structures**

The application code draws a distinction between a Test object and a Result object. The Tests are templates and the action of performing a test. Where Results are concrete records of enacted tests, stored in a database and the subject of analysis.

So a LocationTest would be the template and the act of determining the device's location. A LocationResult is the latitude and longitude output stored on the device and uploaded to the web application.

The following subsections detail each data object as laid out in figure 4.2.

### **4.2.1 TestCampaign**

A unique and human-readable string identifier which groups many TestMasters into a single campaign, potentially from multiple devices. This class combines tests into a unit of work for an individual client.

The TestCampaign class in the web application has no properties other than its name. It serves as the ancestor key for all TestMaster, Vector and CampaignStats objects, simplifying their retrieval from the datastore. In the case of this work the test database used "test\_campaign" and the production database used "production\_campaign" as the test campaign names.

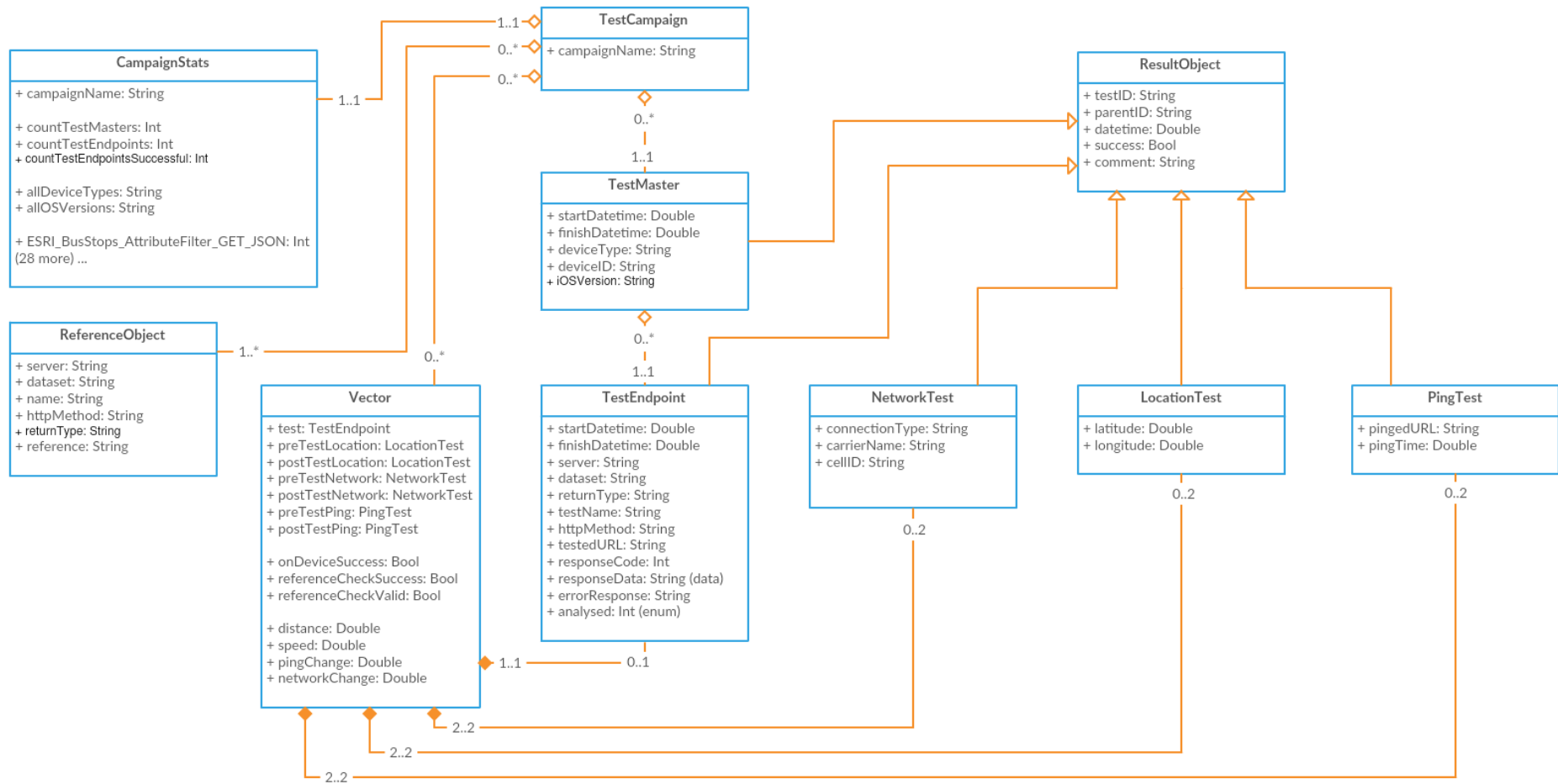


Figure 4.2 - LandgateAPITest data model class diagram

#### 4.2.2 TestMaster

A TestMaster encapsulates all tests undertaken in a single user-initiated test. The LocationTests, NetworkTests, PingTests and EndpointTests performed from a given user test have the same TestMaster as their parent object. This allows all the various subtests to be queried with their fellows in the Analyse function.

TestMasters also group subtests according to the user's perception of the manner in which the tests were done. Thus, allowing the user to review a TestMaster and its children as a single unit of test work in the iOS app interface.

All TestMasters and their children inherit from the ResultObject superclass in both the web and iOS applications. In this manner, they inherit the same properties of datetime, testID, parentTestID and so forth. This is for the sake of convenience and avoiding repeated code.

TestMasters have properties relating to the test device itself which cannot change through the cycle of subtests. The record includes the device type, the version of its operating system and a unique identifier for the device. The device ID is Apple Inc.'s "ID for vendor" a key unique to both the device and the application vendor. This key cannot be traced to a particular device without the application's signed certificate.

The database record corresponding to a TestMaster is a TestMasterResult object.

##### 4.2.2.1 TestEndpoint

The primary focus of the study, TestEndpoints request a set response from the Landgate server. The TestEndpoint records the details of the template (server type, HTTP method, the URL and so forth), the test's outcome (successful or failed on the device), the start and finish time and date, and the data received in the endpoint's response.

TestEndpoints are stored in the iOS application database as EndpointResult objects. Each TestMaster will enact dozens of TestEndpoints. Consequentially, each TestMasterResult will be the parent to dozens of EndpointResults.

#### 4.2.2.2 LocationTest

Acquiring the mobile device's position throughout testing is key to LandgateAPITest's methodology. Information on the environment in which a given test was undertaken can be derived from this data, such as device travel distance and speed.

A LocationTest acquires a latitude and longitude value in the WGS84 coordinate reference system common to GPS devices.

The LandgateAPITest iOS app requests a 10-metre accuracy for location fixes. This accuracy is not guaranteed, should the device be unable to locate with the desired accuracy it will report what it can after timing out. As with all GPS devices, the environment affects location accuracy, particularly when testing indoors. Looser GPS accuracy saves the device's battery power which otherwise would be wasted in trying to acquire a more accurate location fix.

The web application and iOS application databases store each LocationTest's outcome as a LocationResult.

#### 4.2.2.3 NetworkTest

The mobile network connection varies depending on the cell tower, its capabilities and obstacles in the intervening space.

A NetworkTest queries the iOS device for the properties of its network connection. Each NetworkTest records the mobile network provider (called carrier in mobile device parlance) and the class of the mobile network (for example, EDGE, HSDPA, LTE).

LandgateAPITest tests specifically for a Wi-Fi connection before testing for a mobile network connection.

NetworkTest's have a property to hold a unique identifier for a cell tower. In its current form, LandgateAPITest does not record anything in this property. The iOS device reports details of its mobile network connection through Apple's Core

Telephony framework. Intended for the use of mobile communication providers much of Core Telephony's functionality is private. Apps utilising such functions submitted to Apple Inc.'s App Store for review will be summarily rejected. The code LandgateAPITest uses should not be condemned under the current regime. Recording the cell tower id would be cause for rejection, however, so it is excluded.

The result object for a NetworkTest is a NetworkResult.

#### 4.2.2.4 PingTest

As LandgateAPITest cannot directly test network connection speed, a proxy is taken in its place. A PingTest sends a HEAD request to [www.google.com.au](http://www.google.com.au) and measures the time until it receives a response. HEAD requests do not demand any data in the response body, so any intervening time is a factor of the mobile connection and processing time on the Google web server. These two factors are unfortunately inseparable given the app's limitations.

The result object stored in the database for a PingTest is a PingResult.

#### 4.2.3 ReferenceObject

ReferenceObjects hold the correct response data from the Landgate server for each request. Server, returnType and other test template properties identify ReferenceObjects from one another and allow comparison to a TestEndpointResult. The reference property holds the response data in text, either XML, JSON or images converted to base64 text.

These exemplar responses were requested and stored on the 5th of April, 2016. This postdates GME's replacement. References for GME requests were stored in April 2016 from the first test responses in December 2015. Dynamic parts of responses were excluded from the final ReferenceObject, for example, any date or time value that changes between requests.

The administrator uploads text files containing ReferenceObject references to the web application's code repository. A request to the StoreReferences endpoint



enqueues a task to add new and replace old ReferenceObjects with the text file contents.

#### 4.2.4 Vector

The web application's Analyse function parses each TestEndpoint object and attempts to generate a new Vector object. Vectors encapsulate the LocationTest, NetworkTest and PingTest immediately preceding the TestEndpoint along with those immediately following it, retaining pointers to these objects. The function determines the change in Location, Network conditions and Ping response time and takes them as a proxy for the mobile device's changing connectivity environment through the EndpointTest.

The logic to retrieve the six related subtests from the datastore is as follows;

- (a) Query only those records specific to the subtest type (LocationTests, NetworkTests or PingTests)
- (b) Retrieve only results with the same TestMaster key as the TestEndpoint
- (c) Filter results to only those with a datetime property less than the TestEndpoint's datetime for the three preceding subtests (or greater than for the three following tests)
- (d) Sort the results by their datetime, descending for preceding subtests and ascending for following subtests
- (e) Return only the first result, that being the closest in time to the TestEndpoint

If any one of the six subtests is absent, the Vector object cannot be reliably created and the process is aborted. The TestEndpoint is marked "IMPOSSIBLE" to prevent any further automated attempts at analysis. Such objects are not considered any further in this study. Situations like this may arise where the device cancelled a test partway through, precluding the three following tests.

The Haversine distance formula gives a great circle distance travelled between two LocationTests (essentially a straight line at these scales). Dividing this result by the time difference gives an average speed in metres per second. The time between two LocationTests is not the same as the TestEndpoint's total elapsed time. The interval is

greater as it allows for NetworkTests, PingTests and the duration of the LocationTests themselves.

The generation of the mobile network is here taken as a proxy for connection speed. For example, LTE is a fourth-generation network and assigned 4.0 for a networkClass property, where HSDPA would be assigned 3.5, CDMA 2.5 and so on. Wi-Fi is approximated to generation 5.0 due to its higher potential connection speed. Subtracting the following networkClass from the preceding one gives a networkChange value. Positive values reflect improving network generation and negative values degrading.

The change in response time for a HEAD request to google.com.au before and after an EndpointTest is another proxy for change in network connection speed. Subtracting the following test's response time from the preceding test's response time gives a positive pingChange value for improving network speed and a negative one for degrading speed.

The Analyse function also performs the reference check. It assigns the Vector's referenceCheckSuccess property a True value if the TestEndpoint's response data contains the ReferenceObject's reference text or False otherwise.

Vector objects are the basis for all further analysis in this study. All graphs in this work show the Vector rather than the original TestEndpoint or its subtests.

#### 4.2.5 CampaignStats

The CampaignStats class stores counts of TestEndpoints and other subtests, enabling calculation of descriptive statistics for a particular TestCampaign. CampaignStats updates when the iOS application uploads a new TestMaster to the database, adding new counts onto existing values. Also, when the Analyse function creates a new Vector object it updates the CampaignStats properties for reference check success.

The overwhelming majority of CampaignStats properties concern the percentage of successful reference object checks. 0% reference check success rates will exclude the entire test type from further consideration. See the Results section.

## 4.3 iOS Mobile Application

### 4.3.1 Mobile Application Design Principles

The LandgateAPITest iOS application should be quick to build and simple to maintain. The app's design eschews user interface flair in favour of basic views, such as the ubiquitous UITableView. This has the beneficial side-effect of making the app familiar and easy to use for a broader range of the community.

Where possible, open source libraries should replace handwritten code. There being no point to reinventing functionality already well developed and freely available. A common pattern in modern app development, community supported libraries fill narrowly focussed functions. These are easily incorporated into an iOS project through dependency management applications and community code repositories such as GitHub.

LandgateAPITest is not a privacy focussed app. The application collects data that many would consider intrusive, such as location or mobile network connection details. Without this information, the application would not fulfil its objectives. The app's help text encourages users who find such invasiveness unacceptable to uninstall the application.

The application is also a heavy user of the device's battery and mobile network downloads. These are similarly unavoidable, being core to the application's function. The help text highlights that users may upload a test while on a Wi-Fi network to avoid data charges for uploading a TestEndpoint's response data.

Given these two points, users could be justly concerned that the application would be a noticeable drain on their battery and data plan. LandgateAPITest addresses this issue by only testing when the application is in the device's foreground, meaning on the screen while the device is awake. Should the application be sent to the background, it immediately enters an abort state, cancelling active tests and storing what results it has to the database. This can be triggered by the user selecting a different app, clicking the Home button, or by an interrupting phone call.

### 4.3.2 iOS Application Architecture

Apple Inc. advocates the Model-View-Controller (MVC) design pattern in object-oriented code. A controller class intermediates all interaction between the data model layer and the views on the device screen (see figure 4.3). LandgateAPITest follows this pattern by implementing viewcontroller classes for each screen presented to the user.

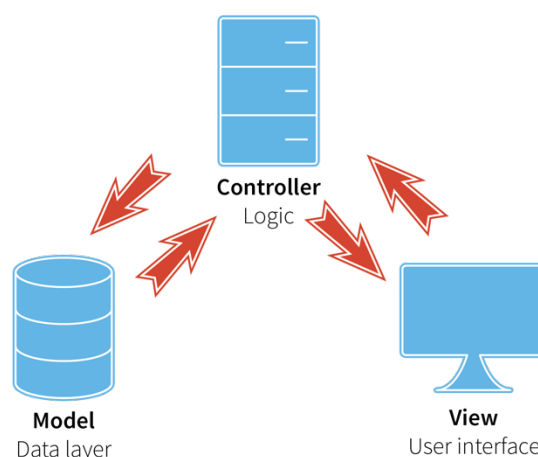


Figure 4.3 - Typical Model-View-Controller (MVC) design pattern

The majority of the application's logic does not reside in viewcontrollers however. It is a common problem in MVC pattern design that the controller classes amass logic to the point of becoming unwieldy and difficult to maintain. LandgateAPITest's viewcontrollers call upon the SingletonTestManager and SingletonUploader classes when the user initiates a test or an upload to the LandgateAPITest web app respectively. These act as an interface for the model layer, abstracting model logic from the point of view of the viewcontrollers.

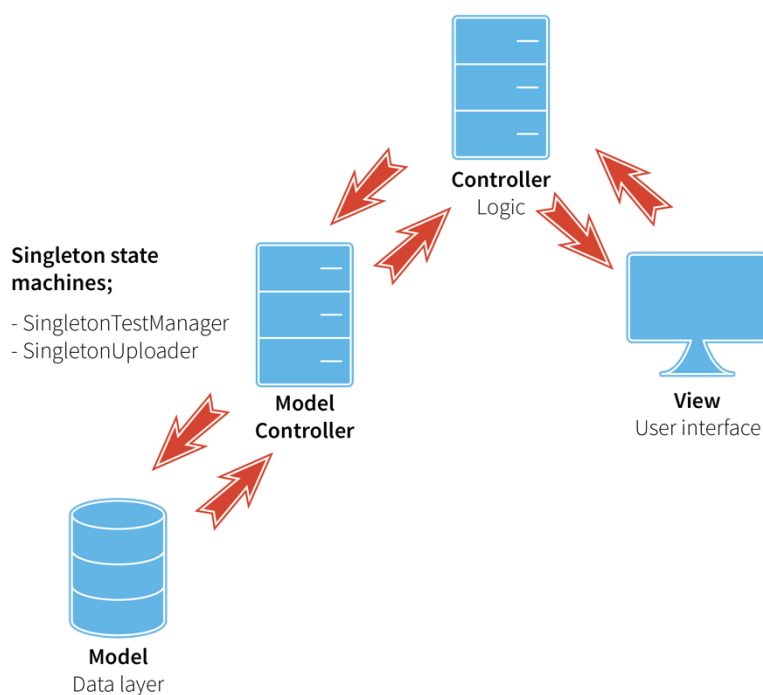


Figure 4.4 - LandgateAPITest's modified MVC design pattern incorporating singleton, state machine controller classes as a model management layer

Firing requests at Landgate's endpoints concurrently, rather than synchronously, would give unreliable response time results. An analysis would not be able to determine what proportion of response time was a factor of the device resolving multiple threads of computation. To avoid this complication LandgateAPITest's iOS app uses a state machine architecture, detailed in figure 4.5.

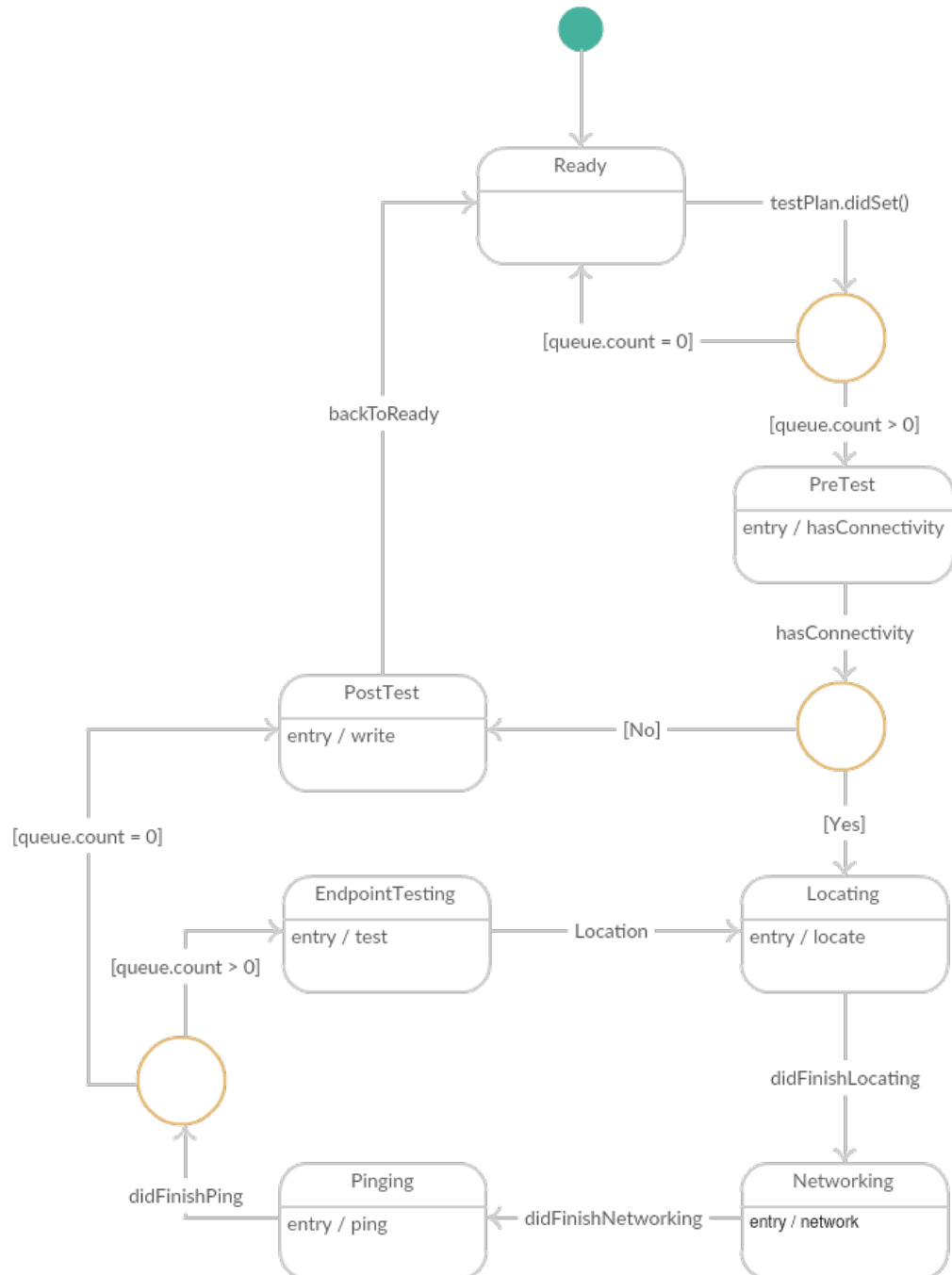


Figure 4.5 - LandgateAPITest mobile application state machine UML diagram

The state machine completes functions sequentially. The completion of each test fires an event function causing the application to change state and after that perform different functions.

When the user initiates a test the SingletonTestManager class switches to its prepareForTest state where it checks preconditions and creates a TestMaster object. From there the SingletonTestManager enters a loop; testing location, network, ping time to google.com.au and then testing a Landgate endpoint (a TestEndpoint) and back to location. The loop continues until the TestMaster's queue of TestEndpoints is exhausted, after which the device writes the TestMaster and all its subtests to its database. Each state performs distinct actions and does not interfere with tests preceding or following as none may start until the earlier test has successfully finished.

At any time the state machine may abort the loop if the preconditions are not met, the app leaves the foreground, or the battery is exhausted. It immediately skips to the post-test state and attempts to save the test results gathered to the device database. Any test (endpoint, location, network or ping) cancelled part-way through is aborted and marked as "Failed On Device."

Uploading in the background of the application uses a similar pattern. The SingletonUploader class changes state from "Ready" to "Uploading" and thence to "Success" or "Failure" depending on the response code from the web app. It settles back to the "Ready" state once its upload queue is emptied.

### 4.3.3 Swift Open Source Packages

#### 4.3.3.1 Realm Mobile Database

The Realm open source mobile database ("realm/realm-cocoa," 2016) is a much simpler on-device storage solution than the proprietary Apple Inc. solution. LandgateAPITest stores all test results in a Realm database file.

#### 4.3.3.2 Transporter State Machine

Denis Telezhkin's Transporter library (Telezhkin, 2016) offers a straightforward state machine implemented in Swift, available under the MIT licence. LandgateAPITest

relies upon Transporter state machines for the SingletonTestManager and SingletonTestUploader classes.

#### 4.3.3.3 Reachability

Ashley Mill's Reachability library queries the device to determine whether and how it is connected to a network (Mills, 2015). LandgateAPITest depends on Reachability's response in two areas. Firstly, having a network connection is a prerequisite to starting a test, either through Wi-Fi or 2G, 3G or 4G mobile network. Secondly, the iOS standard telephony libraries report a mobile network connection regardless of whether there may be an overriding Wi-Fi network connection. LandgateAPITest checks Reachability beforehand for a Wi-Fi connection.

#### 4.3.3.4 KDCircularProgress

Kaan Dedeoglu's KDCircularProgress library initiates a progress indicator that fulfils a circular ring as the task approaches completion (Dedeoglu, 2015). LandgateAPITest uses KDCircularProgress to show the percent completion of a test, updating the indicator each time a sub-test calls its completion delegate method. The library was chosen as it aligned with the design aesthetic of recent iOS releases.

### 4.4 Google Apps Engine Web Service

#### 4.4.1 Web Service Design Principles

The ideal for any web service is to present the latest available data to requesters. LandgateAPITest presents statistics, maps and analysis on objects in the datastore at request time. End users need not await final reports but can check on the status of a testing campaign whenever they wish.

The iOS testing app can produce a large set of testing data in a short timeframe. Cloud computing power is capable of scaling to accommodate excess load, but at a cost. The web application acts to smooth out high load by deferring computationally intensive tasks to avoid peak load and minimise cost.

The web application should analyse TestEndpoint results and not just represent them. Generating actionable information from pure data has much more value to the end user. Similar applications produce appealing graphs with little scientific merit.

LandgateAPITest's charts, while less visually gripping, offer in-depth analysis and proper scientific methodology.

#### 4.4.2 Python Application Architecture

Google App Engine (GAE) Python applications derive their basic functionality from the webapp2 open source library (“Welcome to webapp2! — webapp2 v2.5.1 documentation,” 2011). Developers define web app endpoints and map them to Python classes, so `landgateapitest.appspot.com/database` maps to the Database class in Python code. Then the HTTP method maps to functions within that class. A POST request to the Database endpoint fires the Database class's `post()` function adding the request body to the database. A GET request calls the `get()` function and downloads a TestMaster record to the requester.

Python functions may call for a task to be enqueued. The GAE system will fire the specified request at a later time, ideally when processor load is minimal. A successful POST request to the Database endpoint queues an Analyse endpoint GET request as one of its last tasks. Similar tasks which are anticipated to consume more than trivial resources are deferred as queued tasks in LandgateAPITest, such as importing reference text files to the datastore or updating the model schema.

Google App Engine applications may use any of Google Inc.'s several cloud data storage solutions. LandgateAPITest uses Google Cloud Datastore, a NoSQL database quite distinct from relational databases in that it does not store all records as atomic rows in tables, rather as schema-less objects in distributed documents.

#### 4.4.3 Python Open Source Packages

##### 4.4.3.1 Matplotlib

Well known in academia, Matplotlib is an open source Python library capable of producing detailed graphs from complex datasets (“matplotlib/matplotlib,” 2016). LandgateAPITest's web app leveraged Matplotlib to build graphs of up to the minute Vector object data queried from Google's Cloud Store. Programmatically compiling the figures in this report allows them to be recreated on demand with the latest information from testers.



Matplotlib's analytical power is enhanced when paired with Numpy (“NumPy — Numpy,” 2016). A mathematically focussed library, most useful in LandgateAPITest for its arrays, which are much more analytically oriented than generic Python list collections.

#### 4.4.3.2 Leaflet

The LandgateAPITest web application offers a heatmap of test locations. More precisely, the map shows each Vector object's LocationTest prior to the EndpointTest. Leaflet (Agafonkin, 2016) is a lightweight JavaScript mapping library, it is ideal for a simple application such as this.

Visualising 16,000 test locations as generic markers would result in a confusing map. The Leaflet. Heat (Agafonkin, 2015) library generalises multiple points into a single heat map layer. Closely clustered tests are represented by warmer colours, dispersed ones by cooler colours. OpenStreetMap tiles comprise the base map, available under a CC BY-SA open licence and built by the community of OpenStreetMap contributors.

### 4.5 Other Applications Deployed

Besides code incorporated directly into the product applications, several other applications were instrumental to app development.

#### 4.5.1 Paw

Keeping track of 46 REST requests, each with minor variations from its neighbours, required more than a handwritten list. Paw (version 2.3.3) (“Paw – The most advanced HTTP client for Mac, n.d.) is a Mac application designed for testing REST requests. It simplifies the process of composing the request and its query components. Its most helpful feature is its Swift NSURLSession code output, suitable for directly pasting into an iOS application repository. The code in LandgateAPITest's EndpointTester class where it fires a request to the Landgate server is derived from Paw's example code.

#### 4.5.2 Atom

Atom (version 1.7.2) (“A hackable text editor for the 21st Century”, n.d.) is Github Inc.’s open source text editor used primarily for LandgateAPITest's Python web app development. The Github community has built a plethora of extensions expanding the application's capabilities far beyond the average plain text editor. This entire report was drafted in Markdown styled plain text in Atom.

#### 4.5.3 Xcode

Apple Inc.’s Xcode Integrated Development Environment (IDE) (“Xcode - What's New - Apple Developer,” n.d.) is the orthodox application for developing apps for Apple machines. LandgateAPITest's iOS app was entirely built in Xcode version 7.3.

## 5 RESULTS

### 5.1 Test Regime

The "production\_campaign" featured two main pushes of testing. The first in December 2015 through to January 2016 tested Google Maps Engine endpoints before their shutdown. The campaign's main thrust took place in March 2016 where the majority of tests queried the old OGC endpoints and the new Esri endpoints.

Table 5.1 - Subtest counts

Test Type	Count
Count Test Masters	284
Count Test Endpoints	16144
Count Location Results	16391
Count Network Results	16391
Count Ping Results	16345

The user initiated 284 TestMasters resulting in 16,144, as shown in table 5.1. The count of LocationResults, NetworkResults and PingResults are each over 200 higher than the TestEndpoint count as they are run before the first EndpointTest and after the last one. The small differences account for a few dozen TestMasters cancelled or aborted before finalising the tests at the end.

Tests were undertaken in a broad range of situations common to mobile device use. Situations of varying mobile network signal strength were deliberately sought. Such situations were found while travelling on highways between cities or, Interestingly while crossing the Sydney Harbour Bridge. There were fewer tests conducted while connected to WiFi as the results tend to not be useful except as an upper bound to connection speed.

Device motion ranged from stationary (such as in an office environment) to highway speeds. Note that the chosen level of GPS location accuracy also affected calculation of device speed, particularly where tests complete quickly over a good mobile network but location results less reliable (a common situation in the city).

The latest map information is available from the web service at this URL;  
[https://landgateapitest.appspot.com/map?campaignName=production\\_campaign&mapName=All](https://landgateapitest.appspot.com/map?campaignName=production_campaign&mapName=All)

There were three theatres of action in the campaign. Each test is mapped in a Leaflet web map using the location of its Vector's PreTestLocation (the LocationTest completed before the TestEndpoint began). Visualising 16,000 points would result in an ineffective map, so here closely clustered points are generalised into a heat map. A beneficial side effect of generalisation is to obfuscate precise locations.

The majority of tests took place in Sydney, NSW and its environs, show in figure 5.1. In particular, the regular commute over the harbour to the Central Business District, and the roads and freeways to neighbouring cities.



Several discrete bursts of tests took place in Bathurst, NSW and the highway back and forth to Sydney, NSW. Figure 5.2 shows several clusters of tests in the main streets of Bathurst.

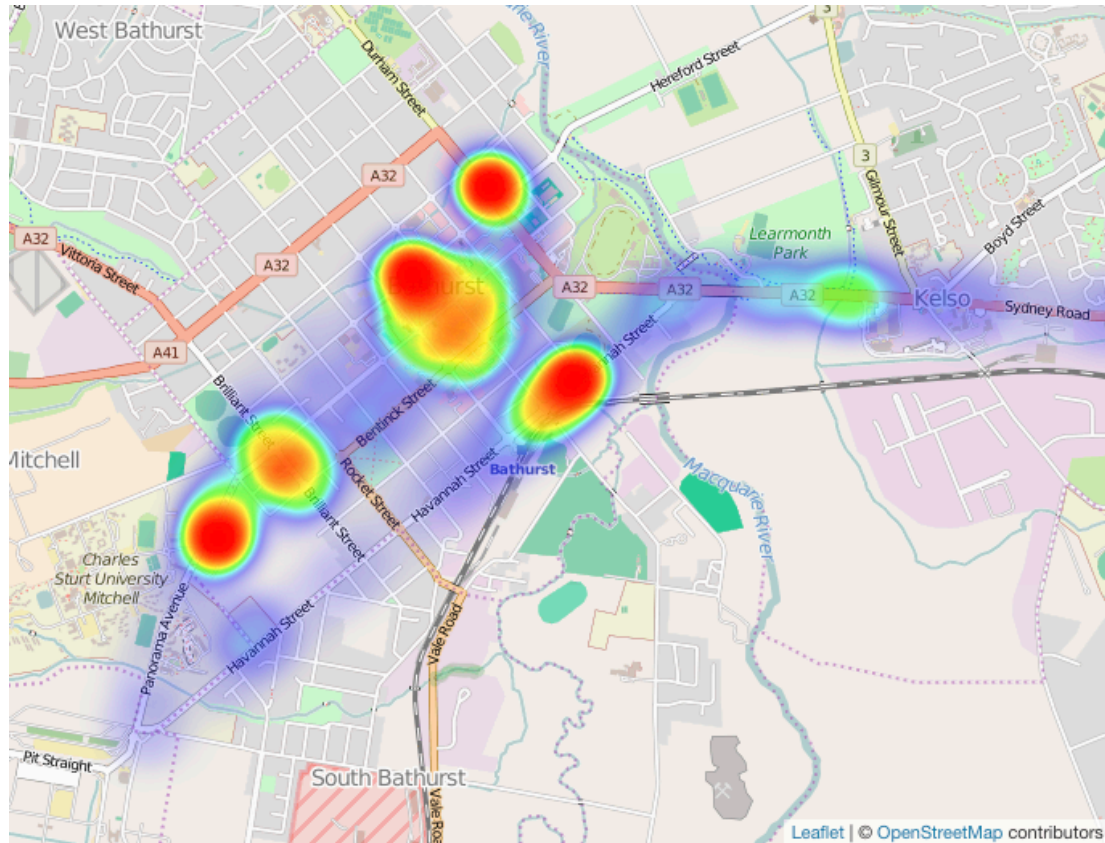


Figure 5.2 - Bathurst Test Heat Map, basemap tiles copyright OpenStreetMap Contributors



Townsville, QLD was the theatre with the least number of tests, but some interesting mobile situations involving ferry crossings and steep terrain on Magnetic Island, illustrated in figure 5.3.

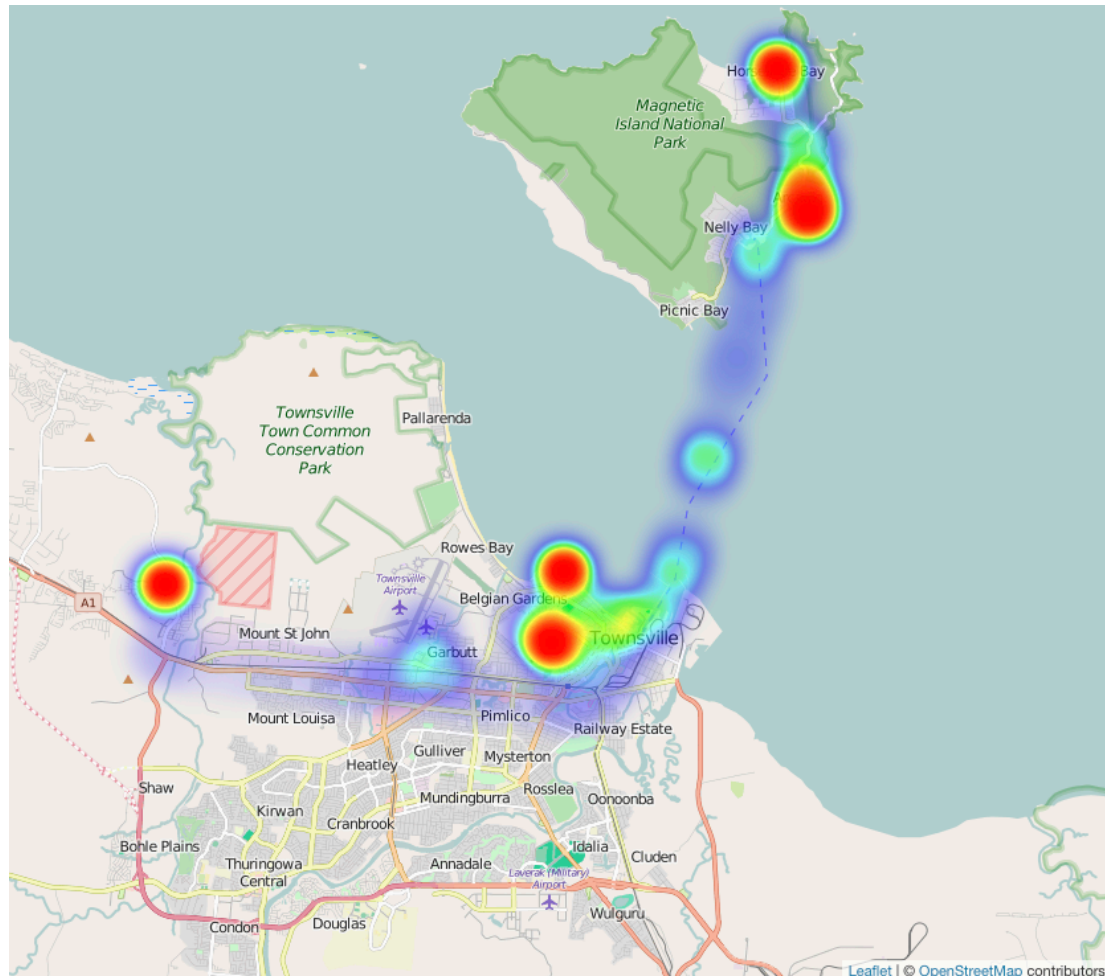


Figure 5.3 - Townsville Test Heat Map, basemap tiles copyright OpenStreetMap Contributors

## 5.2 Test Device Hardware

All tests were performed on an Apple iPhone 6S, model A1688 (a.k.a. iPhone8,1), with 64GB of storage. The standard device comes with a range of mobile radios across many bands; LTE, HSDPA, CDMA, GSM, EDGE, Wi-Fi radios a/b/g/n/ac and GPS and GLONASS receivers (“iPhone 6s – Technical Specifications”, n.d.).

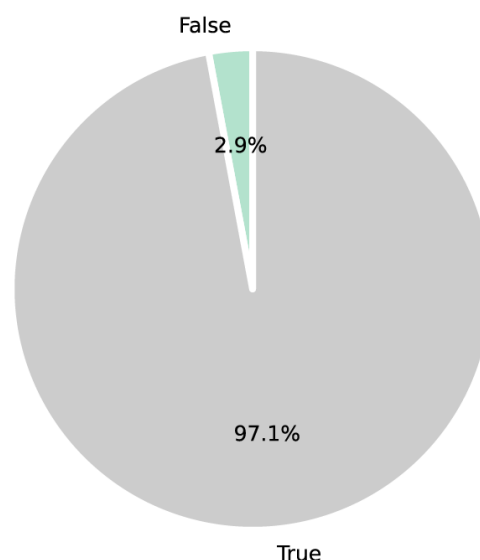
Table 5.2 - Test device characteristics captured by LandgateAPITest iOS application

<b>Campaign Name</b>	production_campaign
<b>All Device Types</b>	iPhone8,1
<b>All iOS Versions</b>	9.1, 9.2, 9.2.1, 9.3, 9.3.1

Table 5.2 shows the operating system changed through the campaign as Apple Inc. updated their software. The first tests launched LandgateAPITest on iOS 9.1, later tests on 9.2, 9.2.1 and later still on 9.3 and 9.3.1.

### 5.3 TestEndpoint Successes and Failures

Of the 16,144 TestEndpoints 15,670 were successful on device (97.06%). These were able to complete the test and received a 200 response code from the Landgate server. The 2.94% of on device failures either could not reach the server (response code 0) or received a server error response (code 500 and above).



LandgateAPITest's Analyse function compared each TestEndpoint's response data to the stored reference data and determined that 13,220 of them match, setting the resultant Vector's referenceCheckSuccess flag to True.

Figure 5.4 - TestEndpoints successful (labelled "True") and failed on device (labelled "False")

Closer examination of referenceCheckSuccess by test type (detailed in table 5.3) showed 9 test types that consistently failed their reference checks (less than 5% passed). All such Vectors had their ReferenceCheckValid flag set to False to exclude them en masse from further analysis on the assumption that there was a systematic issue with their reference data.



Notably, the GetCapabilities tests rarely passed reference checks. Likely causes include changes to services offered during the test campaign or possibly conflicting timestamps buried in the XML response and reference.

Table 5.3 - Percentage of reference checks successful by test type

Test Name	Percent Reference Check Successful
ESRI - BusStops - AttributeFilter - GET - JSON	98.79%
ESRI - BusStops - AttributeFilter - POST - JSON	98.20%
ESRI - BusStops - FeatureByID - GET - JSON	99.00%
ESRI - BusStops - FeatureByID - POST - JSON	98.76%
ESRI - BusStops - GetCapabilities - GET - JSON	97.64%
ESRI - BusStops - GetCapabilities - POST - JSON	98.99%
ESRI - BusStops - IntersectFilter - GET - JSON	97.31%
ESRI - BusStops - IntersectFilter - POST - JSON	98.21%
ESRI - BusStops - Small - GET - JSON	96.98%
ESRI - BusStops - Small - POST - JSON	97.89%
ESRI - Topo - Big - POST - Image	99.45%
ESRI - Topo - Small - GET - Image	99.32%
ESRI - Topo - Small - POST - Image	100%
GME - AerialPhoto - Big - GET - Image	96.97%
GME - AerialPhoto - GetTileKVP - GET - Image	95.65%
GME - AerialPhoto - GetTileKVP2 - GET - Image	93.75%
GME - AerialPhoto - GetTileKVP3 - GET - Image	90.63%
GME - AerialPhoto - GetTileKVP4 - GET - Image	90.91%
GME - AerialPhoto - Small - GET - Image	86.36%
GME - AerialPhoto - WMSGetCapabilities - GET - XML	0%
GME - AerialPhoto - WMTSGetCapabilities - GET - XML	0%
GME - BusStops - AttributeFilter - GET - JSON	83.87%
GME - BusStops - Big - GET - JSON	0%
GME - BusStops - DistanceFilter - GET - JSON	93.75%
GME - BusStops - FeatureByID - GET - JSON	96.55%
GME - BusStops - IntersectFilter - GET - JSON	90.63%
GME - BusStops - Small - GET - JSON	0%
OGC - AerialPhoto - GetTileKVP - GET - Image	98.85%
OGC - AerialPhoto - GetTileRestful - GET - Image	98.36%
OGC - BusStops - AttributeFilter - GET - JSON	0.83%
OGC - BusStops - AttributeFilter - GET - XML	99.72%
OGC - BusStops - AttributeFilter - POST - JSON	98.63%
OGC - BusStops - AttributeFilter - POST - XML	99.45%
OGC - BusStops - Big - GET - JSON	100%
OGC - BusStops - Big - GET - XML	100%

OGC - BusStops - Big - POST - JSON	100%
OGC - BusStops - Big - POST - XML	100%
OGC - BusStops - FeatureByID - GET - JSON	99.30%
OGC - BusStops - FeatureByID - GET - XML	98.85%
OGC - BusStops - FeatureByID - POST - JSON	100%
OGC - BusStops - FeatureByID - POST - XML	99.54%
OGC - BusStops - GetCapabilities - GET - XML	2.88%
OGC - BusStops - GetCapabilities - POST - XML	1.26%
OGC - BusStops - IntersectFilter - GET - JSON	99.73%
OGC - BusStops - IntersectFilter - GET - XML	99.46%
OGC - BusStops - IntersectFilter - POST - JSON	99.46%
OGC - BusStops - IntersectFilter - POST - XML	100%
OGC - BusStops - Small - GET - JSON	98.81%
OGC - BusStops - Small - GET - XML	100%
OGC - BusStops - Small - POST - JSON	99.76%
OGC - BusStops - Small - POST - XML	98.83%
OGC - Topo - Big - GET - Image	3.63%
OGC - Topo - Small - GET - Image	3.48%

See Appendices A and B for a list of URLs for each request.

Of the remaining tests only 79 (0.6%) failed a reference check. And 2.9% of the referenceCheckValid tests failed on the device itself.

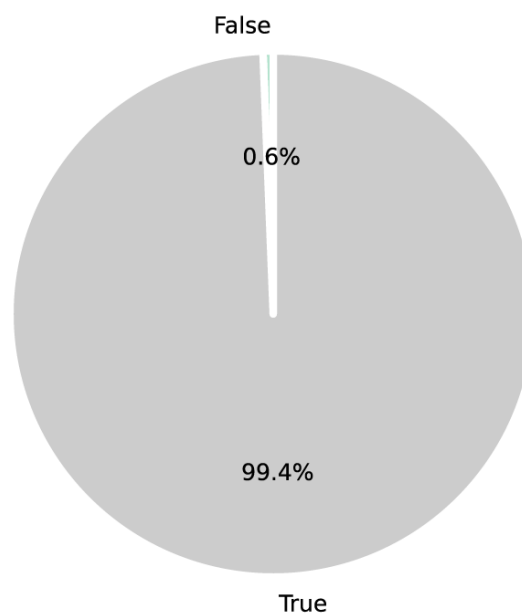


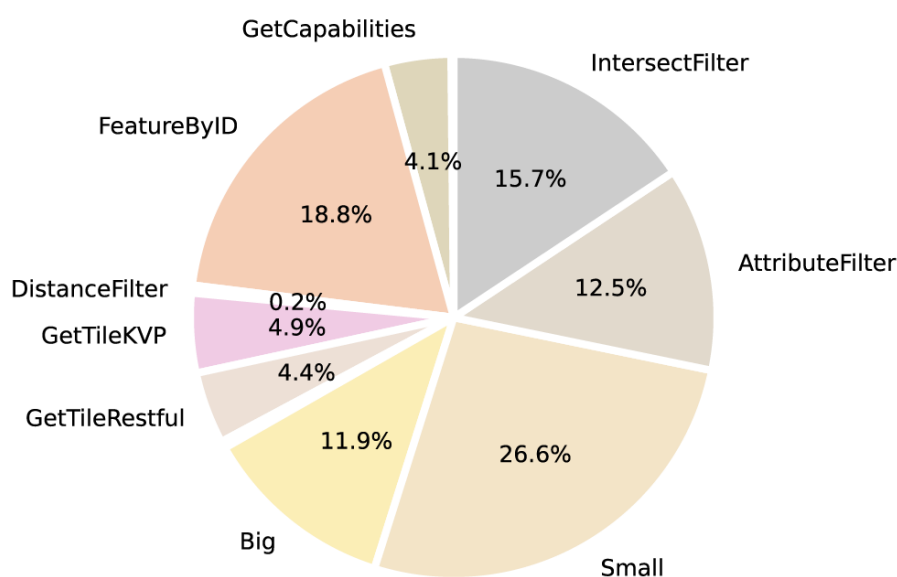
Figure 5.5 - Percentage of Vectors with failed reference checks (labelled “False”)

#### 5.4 Test Results by Response Time

The LandgateAPITest web application produces charts of current data on request to the Graph endpoint. All charts in this text are saved versions of these as they stood

on the 7th of May, 2016. Links are provided in the text to the endpoint for each chart displayed so that the reader may receive the latest information.

The various requests are subcategorised by their test name, a general description denoting near identical requests across the three server types. A FeatureByID request returns the same data from all three servers, though it may not be in the same format (GML, Esri JSON, GeoJSON for example). Figure 5.6 (available at this link; [https://landgateapitest.appspot.com/graphs?campaignName=production\\_campaign&graphName=graph3](https://landgateapitest.appspot.com/graphs?campaignName=production_campaign&graphName=graph3)) was modified to exclude some of the smallest percentage test types to aid reading clarity.



(N.B. Some low percentage tests removed for clarity)

Figure 5.6 - Percentage of Vectors by test type N.B. smaller categories removed for clarity

The most elucidating test types in figure 5.6 are explained in the following paragraphs.

Aligned with Fowler, Hameseder and Peterson's (2012) experimental control showing that response data size affects response time, LandgateAPITest requests "Small" and "Big" responses. Small requests are either for a few features in GML or

JSON or an image only a few tens of pixels in dimension. Big requests ask for 100 vector features or images 500 pixels in dimension.

The distribution of their response times are shown in figure 5.7 as a box and whiskers chart (available at this link; [https://landgateapitest.appspot.com/graphs?campaignName=production\\_campaign&graphName=graph25](https://landgateapitest.appspot.com/graphs?campaignName=production_campaign&graphName=graph25)). Box and whiskers charts show the interquartile range of a distribution. The dataset is divided into four equal parts around the median value, shown as a red line. The first quartile (Q1) to the third quartile (Q3) are contained within the blue box and contain 50% of the points in the dataset. The "whiskers" above and below the box show the range of the data and in a normal distribution would contain over 99% of the data points. Skewed distributions end up excluding the outliers from the interquartile range, shown here as blue crosses.

The "Big" requests have a similar Q1 to Q3 (interquartile range) to "Small" ones. The lowest values in the whiskers are significantly slower to arrive. Both have a significant number of outliers above the maximum response time whisker.

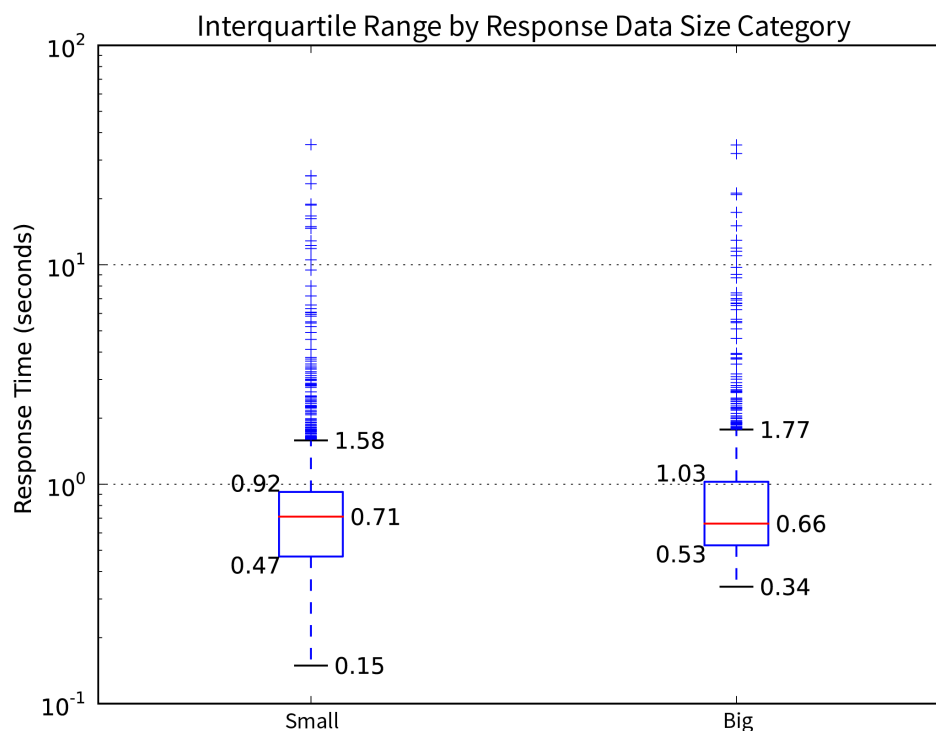


Figure 5.7 - A subset of test types where the request calls for either a small sized response or a larger one comparing their response times

Spatial servers can filter results either by a function of the attributes of each feature, returning features from any location meeting certain criteria of their properties. Features may instead be filtered by a spatial function, returning features from a specific location of any attribute value. The response time frequency distributions for four test types which call upon the server to filter results are shown in figure 5.8 as a box and whiskers chart (available at this link; [https://landgateapitest.appspot.com/graphs?campaignName=production\\_campaign&graphName=graph26](https://landgateapitest.appspot.com/graphs?campaignName=production_campaign&graphName=graph26)).

- Feature by ID calls for a single feature with an exactly matching ID
- Attribute Filter test requests features with a text "location" property containing the word "Curtin"
- Spatial intersect requests provide an envelope (minX, min Y, max X and max Y) covering the Curtin University Bentley campus and request only features intersecting the envelope

- Distance Filter was only requested from GME servers, returning only the closest feature to a point within Curtin University's Bentley campus, the operation involved sorting the entire table by distance and selecting the closest

The two attribute filters generally show a distribution of response times shorter than the two spatial filters. The confidence in this result is not great. Firstly, all have a significant number of high outliers denoting skewed distributions. The spatial filter medians are only 2 to 3 tenths of a second slower than the two attribute filters. The much smaller Distance Filter sample size makes it less worthy of consideration.

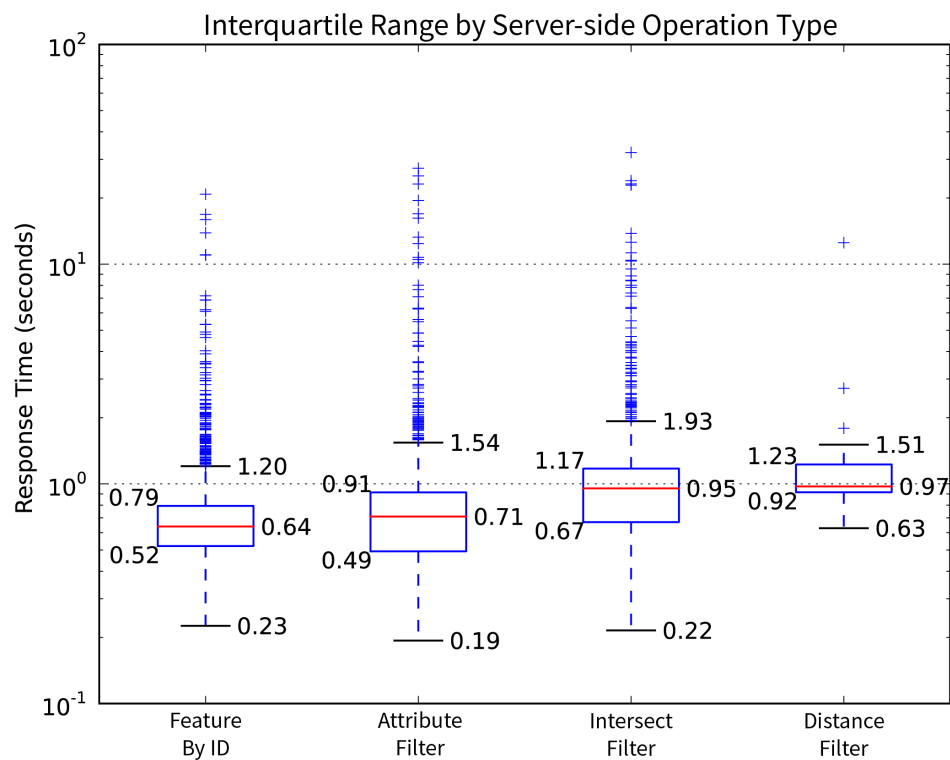


Figure 5.8 - A subset of test types which call upon the server to limit results by a function comparing the distribution of their response times

Figure 5.9 (available from this link;

[https://landgateapitest.appspot.com/graphs?campaignName=production\\_campaign&graphName=graph4](https://landgateapitest.appspot.com/graphs?campaignName=production_campaign&graphName=graph4)) shows that JSON

response data dominated the requests, being the only format available across all three server types. XML's

geographic subset, GML, is only routinely served by OGC endpoints through SLIP.

Images were not requested as often, there being fewer server-side filtering functions available.

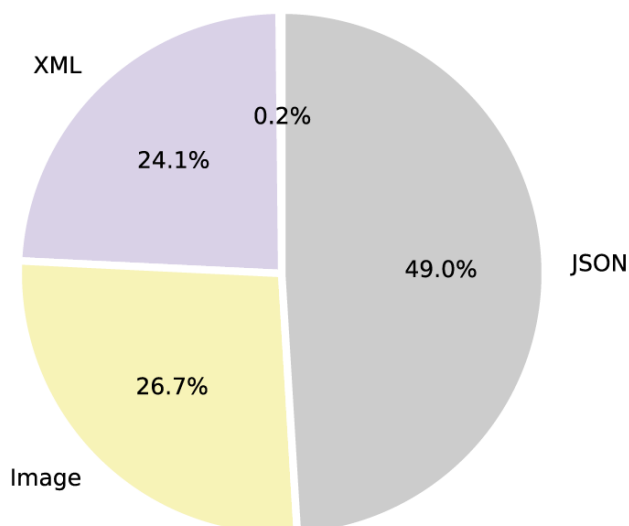


Figure 5.9 - Percentage of Vectors by response data type

Figure 5.10 (available at this link:

[https://landgateapitest.appspot.com/graphs?campaignName=production\\_campaign&graphName=graph24](https://landgateapitest.appspot.com/graphs?campaignName=production_campaign&graphName=graph24)) shows the response time distribution for XML responses is tighter and higher overall than the similar JSON and image request response time distributions. All three have a significant number of outliers in their upper ranges, showing clearly skewed distributions with most requests achieved in short time frames.

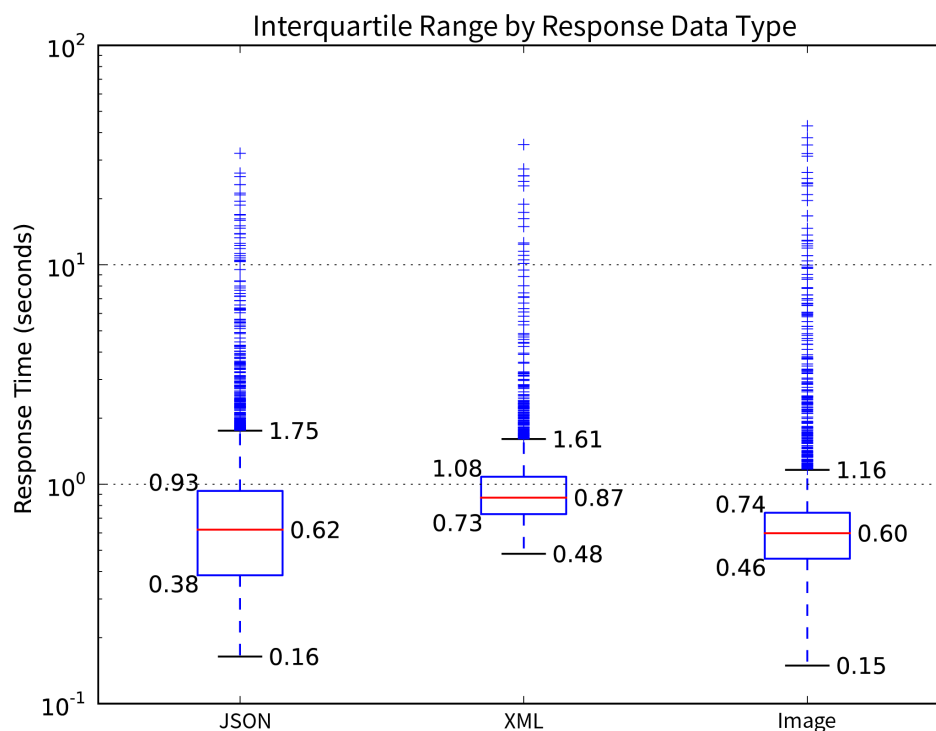


Figure 5.10 - Response time distribution by response data type

The Esri and OGC portion of the test campaign in March 2016 was more vigorous than the earlier GME part in December 2015. Figure 5.11 (available at this link; [https://landgateapitest.apps.pot.com/graphs?campaignName=production\\_campaign&graphName=graph1](https://landgateapitest.apps.pot.com/graphs?campaignName=production_campaign&graphName=graph1)) shows how clearly the Esri and OGC tests overwhelm the fewer GME ones.

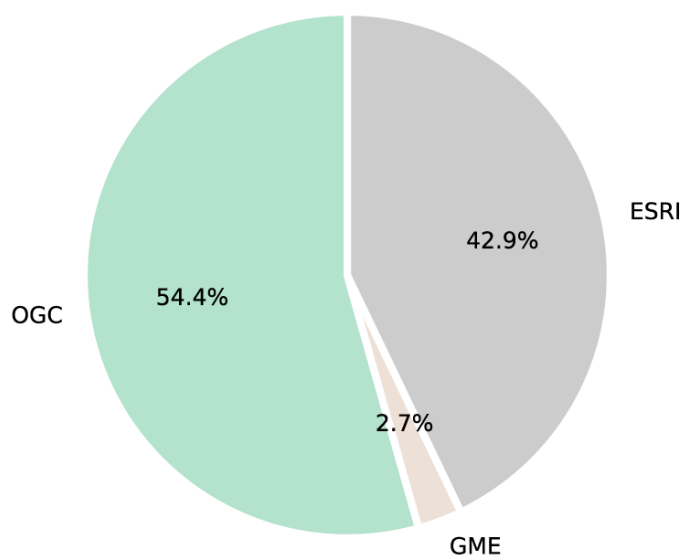


Figure 5.11 - Percentage of Vector objects by server type



The box and whiskers chart in figure 5.12 (available at this link; [https://landgateapitest.appspot.com/graphs?campaignName=production\\_campaign&graphName=graph22](https://landgateapitest.appspot.com/graphs?campaignName=production_campaign&graphName=graph22)) appears to show a clear performance win for the Esri servers over the OGC servers, having a much lower median and interquartile range. There is a significant consideration here that Esri servers do not supply heavier payload XML/GML responses where OGC ones do. As the response data type and response size charts show, on average larger responses have slower response times.

The GME tests fill a broader interquartile range and have fewer outliers. A larger sample set of these requests could have increased confidence in this result.

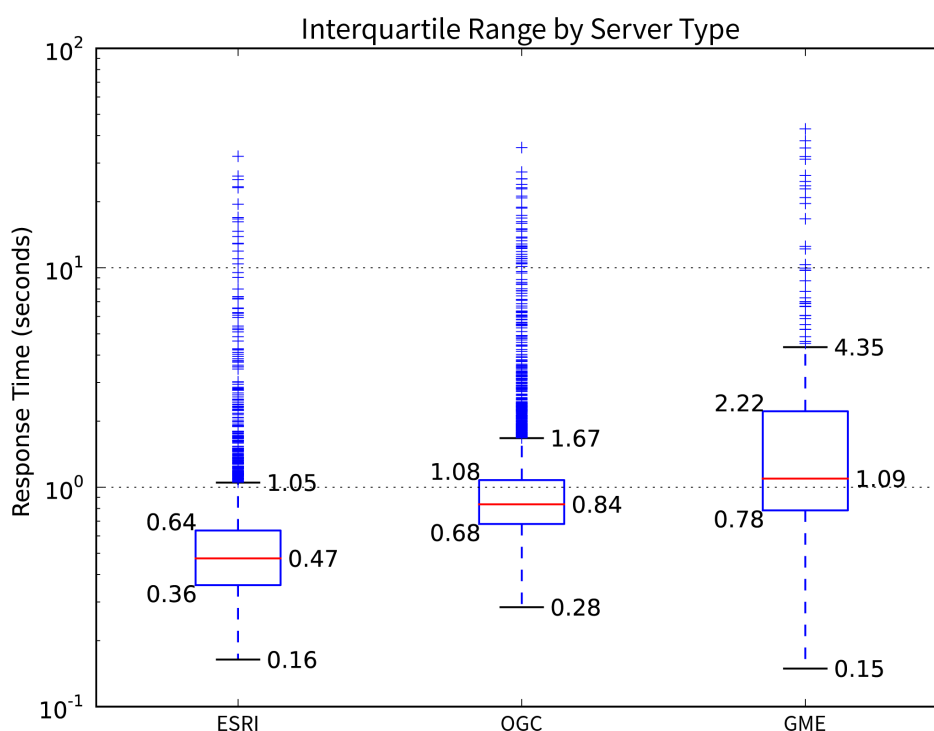


Figure 5.12 - Response time distribution by server type

Figure 5.13 (available at this link;

[https://landgateapitest.appspot.com/graphs?campaignName=production\\_campaign&graphName=graph2](https://landgateapitest.appspot.com/graphs?campaignName=production_campaign&graphName=graph2)) shows tests

were almost evenly split between the two HTTP methods favoured by spatial servers; GET and POST. The greater proportion of GET requests are partly due to the lack of POST requests

created for the GME server and the map tile requests mostly being GET's with key value coding or straight RESTful endpoints.

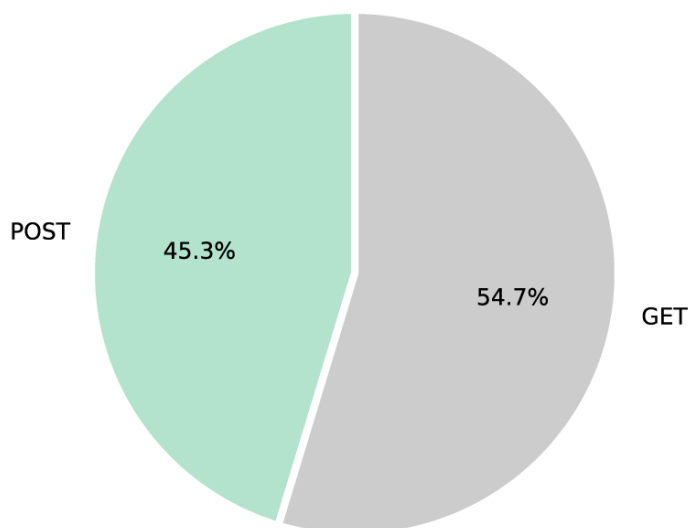


Figure 5.13 - Percentage of Vectors by HTTP Method (GET and POST)

There was no distinct difference in response time between the two methods as can be seen in figure 5.14 (available at this link; [https://landgateapitest.appspot.com/graphs?campaignName=production\\_campaign&graphName=graph23](https://landgateapitest.appspot.com/graphs?campaignName=production_campaign&graphName=graph23)). The medians and boundaries of interquartile ranges are similar enough that differences could be rounding errors.

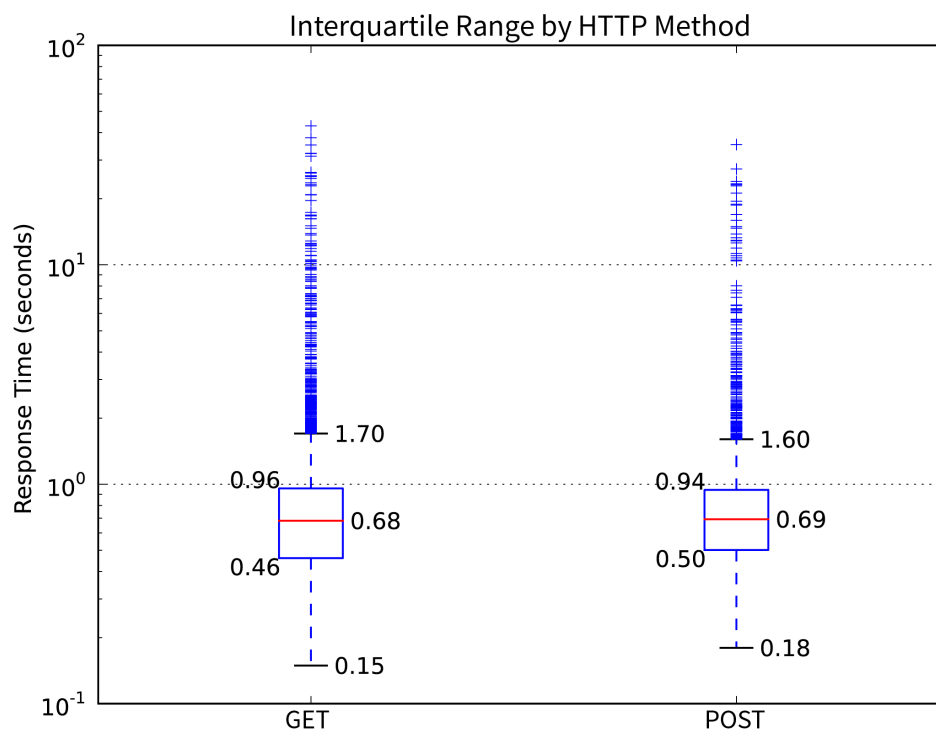


Figure 5.14 - HTTP Method (GET and POST) response time distributions in box plot

An important disparity to note, Esri POST requests required Form-URL-encoded bodies (i.e. content type = application/x-www-form-urlencoded), whereas OGC POST requests were all XML in plain text (content type = text/xml).

### 5.5 Test Results by Distance Device Travelled

The test device deployed determined its location through GPS. The Vector object considers the distance between the LocationTest prior to an EndpointTest and the LocationTest afterwards. By comparing each Vector's distance property to its response time, a scatter plot is produced. Then the web application categorises the points by the EndpointTest's success, on device failure or reference check failure.

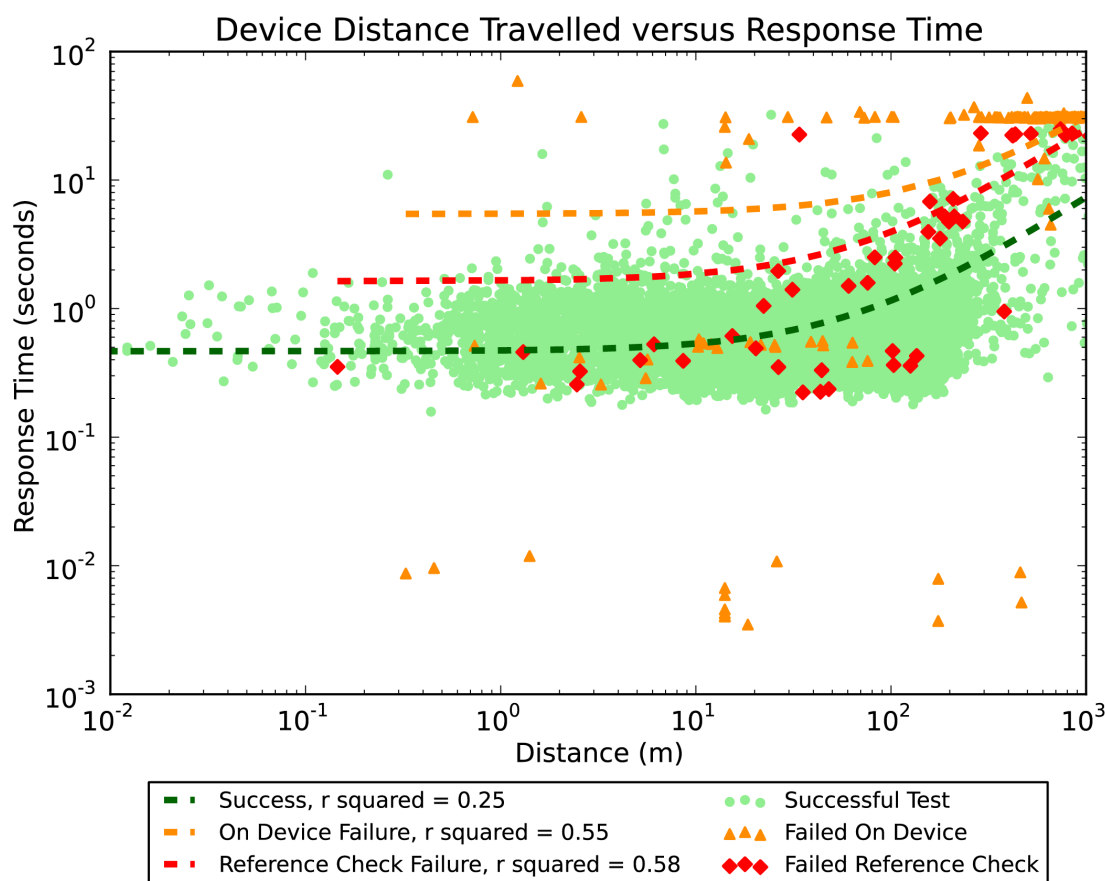


Figure 5.15 - A scatterplot of distance device travelled (metres) versus response time (seconds) for each Vector object

Figure 5.15 (available at this link; [https://landgateapitest.appspot.com/graphs?campaignName=production\\_campaign&graphName=graph12](https://landgateapitest.appspot.com/graphs?campaignName=production_campaign&graphName=graph12)) shows the green successful tests show a loose trend of increasing response time in line with increasing distance travelled.

The orange "failed on device" category exhibit three distinct horizontal bands of response times. The bottommost band are the shortest response times, notably so as they are at most 1/100th of a second. These are the cases where the device had no mobile network connection at all and the operating system aborted the request immediately without even an attempt to send it to the server.

The middle band of "on device failures" are those with similar response times to many successful requests. These are the tests cancelled before completion. They had a successful link to the server but the test was interrupted by an incoming phone call

or the app was otherwise switched to the background. In line with the application's design goals these tests were aborted, their response time recorded and marked as on device failures.

The uppermost band forms a clear line around 30 seconds in response time. This is the standard time-out length for a web service request on an iOS device. Requests without a response are aborted by the system. Interestingly, the majority of these failures occurred when the device travelled more than 100m. In such cases, the device is more likely to change mobile network cell tower or enter areas of weaker signal, preventing completion of the exchange.

Tests with failed reference checks returned data inconsistent with the reference files in the web application's archive. Similarly, the majority occurred when the device travelled more than approximately 50m in the test period. This indicates changing mobile network environment interrupts reception of response data.

The scatterplot shows enough noise to produce R-squared values that are less than ideal. Each value of distance produces a range of response times due to several factors, most notably the response payload size varies by request type. Distance values are not perfect either as consideration must be given to the GPS receiver's desired and possible accuracy.

## 6 DISCUSSION

The results of deploying the LandgateAPITest suite to test SLIP and Landgate's spatial web services show that the mobile network environment is the major determinant of request success and response speed. Greater distance travelled during a test increases response time, whereas GET and POST requests and other server and request type dimensions have minimal affect.

The frequency distributions of response times are skewed towards shorter timeframes. Regardless of the dimension studied, for instance OGC versus Esri, XML versus JSON versus image data, the bulk of response times fell within the same order of magnitude and less than a second. The fact that the result charts need a logarithmic axis for response time to show the interquartile range indicates that Landgate's servers are suitable for the range of mobile situations investigated.

Measurement of response time is useful for determining Park and Ohm's (2014) service and display quality factor. The frequency distribution of response times is an objective measurement which Landgate can use to show improvement of their service delivery.

As other studies showed, JSON responses are lighter and faster to download. They are thus better suited to mobile devices where data caps and slower mobile networks are real limitations. XML and GML suit situations where strict adherence to schema is critical to process success.

Each test type asked the same question of the three server types. The AttributeFilter test should have the same response from GME, OGC and Esri servers as the fundamental query is the same. Yet, each server type responds with differently formatted data, necessitating different referenceObjects for each. The lack of a common format makes developing applications to take advantage of services more complex. More so, the change from GeoJSON in OGC and GME endpoints to the incompatible Esri JSON requires external developers to rework their apps just to maintain existing functionality.

Esri's purely RESTful implementation denotes errors and exceptions in a manner more familiar to web and app developers. Exceptions result in a response code other than 200, namely a 400 series or 500 series integer. Whereas, OGC servers often bury exceptions within the response data itself and supply a 200 response code. Programming an application to react to a non-200 response code is much more straightforward than having to test response data for exception information.

The finding that failures are more frequent with increasing distance travelled is not an issue with Landgate's servers. Longer distances travelled during tests are an outcome of highway speed travel where the device is more likely to encounter signal interruptions or inferior signal strength.

LandgateAPITest discovered only 79 reference check failures. This is, admittedly, a small sample set from which to draw conclusions on whether spatial servers return inconsistent or incomplete data in specific circumstances. A few orders of magnitude more such errors could substantiate conclusions. Unfortunately, this would require more time and data download limit than this study has resources to allow.

According to this work, inconsistent data are delivered only 0.6% of the time. If this finding is extended, it can be assumed that there are few situations where this would be a critical hindrance for a mobile device user.

Reliable and complete response data helps ensure users can orient themselves in the map compared to their real world location, Park and Ohm's (2014) perceived locational accuracy. The percentage of responses with failed reference checks is a second metric which demonstrates successful web service data delivery.

Measurements of response time frequency distribution and percentage of reference check failures are LandgateAPITest's mobile suitability metrics. Improving these metrics should lead to greater mobile map user acceptance of SLIP's services, even though they may not realise where the underlying data originates. Accordingly, greater acceptance leads to a greater intention to make use of the services. Should more users and more application developers make use of SLIP's web services

Landgate could well be said to be succeeding their shared goal with WALIS to improve access to spatial information.

The OASIS web service quality standard (Kim et al., 2012) calls for calculation of Availability, Accessibility and Successability (among others). These are predicated on the assumption that the testing device is guaranteed access to the internet to perform its tests. In other words, the testing device is assumed to be infallible while the tested service is not. This is entirely possible to achieve in controlled conditions, the testing machine simply does not send a request when it is not certain of success, or ignores tests where certain preconditions of controlled experiment are not met. The output then is a percentage of tests where the tester was able to contact the service, the difference from 100% being entirely the fault of the service.

LandgateAPITest's methodology does not assume as given nor control its connection to the internet. The application proceeds with a test so long as there is some connectivity, regardless of its reliability. A failed request which timed out (and hence received a "0" response code from the iOS application) could either be the fault of server downtime or lack of a mobile network connection on the device's part.

As such LandgateAPITest is not able to reliably determine Accessibility or Availability. These are common testing metrics and future versions should address this shortcoming.

Successability is similar in that LandgateAPITest cannot reliably determine whether the lack of a response to a request is the fault of the server or the network. However, the Oasis standard assumes WSDL responses are "error-free". LandgateAPITest again does not make this assumption and interrogates the response data for errors.

Overall, LandgateAPITest is not an everyday testing suite. There are suitable applications, as shown in the literature review, capable of determining such oft required statistics. This app more closely tests the mobile device user's experience with the data served by a geographic web service. Environmental factors of network connectivity, constrained device processing power and others have a larger effect on whether the service will successfully deliver data to meet the user's needs.



LandgateAPITest's inability to calculate Successability and related metrics is a hindrance to its deployment as the sole testing application for spatial data infrastructure. It should be deployed alongside conventional testing packages to calculate this information.

## 7 RECOMMENDATIONS

Esri ArcGIS Servers can provision OGC and KML web services alongside their Esri REST services. Landgate has enabled WMS services for their Public ArcGIS MapServers, but not WFS or KML. Doing so would improve interoperability for open source apps such as QGIS at little incremental cost. Older infrastructure could then be decommissioned without reduced service to the community.

Esri JSON is not the same format as the open standard GeoJSON served by GME and OGC endpoints (note, this is not an OGC standard (Reed, 2011)). The JSON output from Esri endpoints represents the same data as a response from an OGC or GME endpoint but is laid out differently and must be parsed into an in-memory geometry object before the two can be directly compared. Replacing GeoJSON with Esri JSON requires all applications which depend on these endpoints to adapt to the new format.

The latest version of ArcGIS Server (10.4) can supply GeoJSON in response to a Query request (“What's new in ArcGIS 10.4.1 for Server—Documentation | ArcGIS for Server,” n.d.). Landgate employs the previous generation ArcGIS Server, 10.3.1 for their production server infrastructure (“SLIP Public ArcGIS REST Services Directory,” n.d.).

Should older OGC servers be decommissioned Landgate could offer WFS services from ArcGIS servers. This would allow them to continue to offer OGC standard GML despite changes to underlying server software.

Any recommendations for improving mobile suitability metrics (response time frequency distributions and reference check percentage) would require detailed discussion between Landgate and their internal and external stakeholders. Given such, these recommendations will not be addressed in this writing.

## 8 FUTURE WORK

This work is by no means exhaustive. There remain several possibilities for expanding the application's reach and improving the depth of information generated from a campaign.

The foremost improvement to LandgateAPITest's functionality would be a more detailed investigation of the tests that failed their reference check, i.e. their response data did not match that stored in the web app database. This process could be semi-automated. Response data could be scanned for keywords such as "exception" in XML key/values to identify exception responses in place of the proper response data. Partial responses could be identified by longest common string comparison functions. Analysis of these categories of failed responses could provide more accurate insight into Landgate's server's suitability for mobile device traffic. This, in turn, would improve SLIP's mobile suitability metrics for Landgate and WALIS.

A closer collaboration with the Landgate team could permit the investigators to inspect the server logs for the web app's requests. The causes of failed requests could be discerned from a combination of the LandgateAPITest app data and the server's log data. Especially, testers could relate server error conditions and logs to requests with a 500 response.

The OASIS Web Services Quality Factors (Kim et al., 2012) identify three types of latency in requests, ClientLatency, NetworkLatency and ServerLatency. Their sum being the total response time (that which LandgateAPITest records). Analysis of server logs could differentiate ServerLatency and NetworkLatency and compare latency in a variety of situations common to mobile devices.

Further development work could create a version of the LandgateAPITest mobile app suitable for Android OS devices. This would broaden the base of testers available and add another dimension to the result data. Investigation could show whether the different spatial server types offer better performance and reliability for iOS or Android devices.

This study only compared OGC services provisioned by Landgate's earlier server infrastructure. It would be instructive to see whether Esri provisioned OGC services compared favourably with other OGC servers.

Reference data can be considered in a more adaptable methodology than a straightforward equal/not equal test. The minor changes to GetCapabilities documents that caused their Reference Checks to fail wholesale could be averted through cleverer comparisons. Advanced string comparison techniques, such as longest common substring or edit distance, could be given a threshold similarity to assign success. For example, strings that are 90% the same could pass.

Success in collaborating with Western Australia's Landgate authority could be replicated with other Australian state cadastre authorities, such as NSW Land and Property Information. Determination of response time frequency distributions and rates of reference check failures would provide similar guidance to other organisations and offer opportunities to learn from one another which could lead to improved services for all Australian states.

## 9 CONCLUSIONS

This 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.

This work contributes to three aspects of the body of web service evaluation literature.

Firstly, LandgateAPITest tests 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. This testing campaign showed that the majority of responses were received in under a second and is suitable for mobile use.

Secondly, more aspects of the interaction between client and server than merely average response time are examined. Determining the rate of inconsistent responses provides Landgate with a second metric indicative of user acceptance of SLIP's services. The campaign detailed in this work showed only 0.6% of responses returned inconsistent responses, but LandgateAPITest in its current form is incapable of determining the cause. Given the increasing frequency of reference check failures with increasing distance travelled it is assumed that the mobile network environment is the main determinant.

Lastly, the examination of a single service provider, Landgate and WALIS's SLIP service, lead to practical and actionable recommendations to improve its suitability for mobile users and application developers. Google Inc.'s deprecation of Google Maps Engine and SLIP's enforced change to Esri servers caused some disruption due to a change in JSON data format. Landgate can smooth such transitions by continuing to serve OGC data from the new Esri servers.

This is not the first work to study each of these aspects, as is evident in the related work section. It is the intersection of these three aspects that grants LandgateAPITest the opportunity to produce actionable recommendations for Landgate and provide mobile suitability metrics for objective goal success measurement.

## REFERENCES

- Agafonkin, V. (2015, October 26). Leaflet/Leaflet.heat. Retrieved May 2, 2016, from <https://github.com/Leaflet/Leaflet.heat>
- Agafonkin, V. (2016, April 18). Leaflet. Retrieved from [leafletjs.com/](http://leafletjs.com/)
- ArcGIS for Server Features. (n.d.). ArcGIS for Server Features. Retrieved October 3, 2015, from <http://www.esri.com/software/arcgis/arcgisserver/features>
- Cannata, M., Antonovic, M., & Molinari, M. E. (2014). Load testing of HELIDEM geo-portal: an OGC open standards interoperability example integrating WMS, WFS, WCS and WPS. *International Journal of Spatial Data Infrastructures Research*, 6, 2011, (9), 107–130. <http://doi.org/10.2902/1725-0463.2014.09.ART5>
- 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*.
- D'Mello, D. A., & Ananthanarayana, V. S. (2010). A Review of Dynamic Web Service Description and Discovery Techniques (pp. 246–251). Presented at the ICIIC '10: Proceedings of the 2010 First International Conference on Integrated Intelligent Computing, IEEE Computer Society. <http://doi.org/10.1109/ICIIC.2010.57>
- 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>
- Dedeoglu, K. (2015, November 11). [kaandedeoglu/KDCircularProgress](https://github.com/kaandedeoglu/KDCircularProgress).
- 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-lp>
- 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*.
- Huisman, O., & de By, R. (2009). Principles of Geographic Information Systems. ITC Educational Textbook Series.
- 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>
- Ioini Nabil, El. (2015). Specifications for Web Services Testing: A Systematic Review. Presented at the SERVICES '15: Proceedings of the 2015 IEEE World Congress on Services, IEEE Computer Society.
- 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/wsrm/WS-Quality-Factors/v1.0/cos01/WS-Quality-Factors-v1.0-cos01.html>
- la 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%202.0%20November%202012.pdf>
- matplotlib/matplotlib. (2016, January 11). matplotlib/matplotlib.
- 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>
- Mills, A. (2015, December 11). ashleymills/Reachability.swift.
- 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>
- NumPy — Numpy. (2016). NumPy — Numpy. *Numpy.org*.
- 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>
- Qiu, D., Li, B., Ji, S., & Leung, H. (2015). Regression Testing of Web Service: A Systematic Mapping Study. *ACM Computing Surveys (CSUR)*, 47(2), 21–46. <http://doi.org/10.1145/2631685>
- realm/realm-cocoa. (2016, March 26). realm/realm-cocoa.
- Reed, C. (2011). OGC standards. In *Advances in Web-based GIS, Mapping Services and Applications* (pp. 327–348). CRC Press. <http://doi.org/10.1201/b11080-26>
- 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](https://www2.landgate.wa.gov.au/c/document_library/get_file?uuid=18853442-51bf-4a11-9ccc-76a61d891e51&groupId=10136)
- SLIP Public ArcGIS REST Services Directory. (n.d.). SLIP Public ArcGIS REST Services Directory. Retrieved July 11, 2016, from <https://services.slip.wa.gov.au/public/rest/services>
- 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](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](http://www0.landgate.wa.gov.au).
- Tahir, A., Tosi, D., & Morasca, S. (2013). A systematic review on the functional testing of semantic web services. *Journal of Systems and Software*, 86(11), 2371–2393. <http://doi.org/10.1016/j.jss.2013.04.021>
- Telezhkin, D. (2016, January 9). DenHeadless/Transporter.
- 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>
- Welcome to webapp2! — webapp2 v2.5.1 documentation. (2011). Welcome to webapp2! — webapp2 v2.5.1 documentation. *Webapp-Improved.Appspot.com*.
- What's new in ArcGIS 10.4.1 for Server—Documentation | ArcGIS for Server. (n.d.). What's new in ArcGIS 10.4.1 for Server—Documentation | ArcGIS for Server. Retrieved July 11, 2016, from <http://server.arcgis.com/en/server/latest/get-started/windows/what-s-new-in-arcgis-for-server.htm>
- 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>
- Xcode - What's New - Apple Developer. (n.d.). Xcode - What's New - Apple Developer. Retrieved April 20, 2016, from <https://developer.apple.com/xcode/>
- 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>

## **APPENDIX A WEB SERVICE GET REQUESTS**

Table 9.1 - Esri GET Requests

Test Name	URL
ESRI - BusStops - AttributeFilter - GET - JSON	<a href="https://services.slip.wa.gov.au/public/rest/services/Landgate_Public_Maps/Cultural_and_Society/MapServer/1/query?where=location%20LIKE%20%27%25Curtin%25%27%20AND%20status%3D%27Active%27%20AND%20suburb%3D%27BENTLEY%27&amp;text=&amp;objectIds=&amp;time=&amp;geometry=&amp;geometryType=esriGeometryEnvelope&amp;inSR=&amp;spatialRel=esriSpatialRelIntersects&amp;relationParam=&amp;outFields=stopid%2Cstopnumber%2Cstatus%2Cstopname%2Clocation%2Cstoptype%2Csuburb%2Cbay%2Cshelter%2Casset%2Cweekpam%2Cweekppm%2Cweekoff%2Cweekrem%2Csatpam%2Csatppm%2Csatoff%2Csatrem%2Csunpham%2Csunhppm%2Csunphoff%2Csunphrem%2Coid&amp;returnGeometry=true&amp;returnTrueCurves=false&amp;maxAllowableOffset=&amp;geometryPrecision=&amp;outSR=4326&amp;returnIdsOnly=false&amp;returnCountOnly=false&amp;orderByFields=&amp;groupByFieldsForStatistics=&amp;outStatistics=&amp;returnZ=false&amp;returnM=false&amp;gdbVersion=&amp;returnDistinctValues=false&amp;resultOffset=&amp;resultRecordCount=999&amp;f=json">https://services.slip.wa.gov.au/public/rest/services/Landgate_Public_Maps/Cultural_and_Society/MapServer/1/query?where=location%20LIKE%20%27%25Curtin%25%27%20AND%20status%3D%27Active%27%20AND%20suburb%3D%27BENTLEY%27&amp;text=&amp;objectIds=&amp;time=&amp;geometry=&amp;geometryType=esriGeometryEnvelope&amp;inSR=&amp;spatialRel=esriSpatialRelIntersects&amp;relationParam=&amp;outFields=stopid%2Cstopnumber%2Cstatus%2Cstopname%2Clocation%2Cstoptype%2Csuburb%2Cbay%2Cshelter%2Casset%2Cweekpam%2Cweekppm%2Cweekoff%2Cweekrem%2Csatpam%2Csatppm%2Csatoff%2Csatrem%2Csunpham%2Csunhppm%2Csunphoff%2Csunphrem%2Coid&amp;returnGeometry=true&amp;returnTrueCurves=false&amp;maxAllowableOffset=&amp;geometryPrecision=&amp;outSR=4326&amp;returnIdsOnly=false&amp;returnCountOnly=false&amp;orderByFields=&amp;groupByFieldsForStatistics=&amp;outStatistics=&amp;returnZ=false&amp;returnM=false&amp;gdbVersion=&amp;returnDistinctValues=false&amp;resultOffset=&amp;resultRecordCount=999&amp;f=json</a>
ESRI - BusStops - FeatureByID - GET - JSON	<a href="https://services.slip.wa.gov.au/public/rest/services/Landgate_Public_Maps/Cultural_and_Society/MapServer/1/query?where=status%3D%27Active%27&amp;text=&amp;objectIds=1540&amp;time=&amp;geometry=&amp;geometryType=esriGeometryEnvelope&amp;inSR=&amp;spatialRel=esriSpatialRelIntersects&amp;relationParam=&amp;outFields=%2A&amp;returnGeometry=true&amp;returnTrueCurves=false&amp;maxAllowableOffset=&amp;geometryPrecision=&amp;outSR=4326&amp;returnIdsOnly=false&amp;returnCountOnly=false&amp;orderByFields=&amp;groupByFieldsForStatistics=&amp;outStatistics=&amp;returnZ=false&amp;returnM=false&amp;gdbVersion=&amp;returnDistinctValues=false&amp;resultOffset=&amp;resultRecordCount=25&amp;f=json">https://services.slip.wa.gov.au/public/rest/services/Landgate_Public_Maps/Cultural_and_Society/MapServer/1/query?where=status%3D%27Active%27&amp;text=&amp;objectIds=1540&amp;time=&amp;geometry=&amp;geometryType=esriGeometryEnvelope&amp;inSR=&amp;spatialRel=esriSpatialRelIntersects&amp;relationParam=&amp;outFields=%2A&amp;returnGeometry=true&amp;returnTrueCurves=false&amp;maxAllowableOffset=&amp;geometryPrecision=&amp;outSR=4326&amp;returnIdsOnly=false&amp;returnCountOnly=false&amp;orderByFields=&amp;groupByFieldsForStatistics=&amp;outStatistics=&amp;returnZ=false&amp;returnM=false&amp;gdbVersion=&amp;returnDistinctValues=false&amp;resultOffset=&amp;resultRecordCount=25&amp;f=json</a>
ESRI - BusStops - GetCapabilities - GET - JSON	<a href="https://services.slip.wa.gov.au/public/rest/services/Landgate_Public_Maps/Cultural_and_Society/MapServer/1?f=json">https://services.slip.wa.gov.au/public/rest/services/Landgate_Public_Maps/Cultural_and_Society/MapServer/1?f=json</a>
ESRI - BusStops - IntersectFilter - GET - JSON	<a href="https://services.slip.wa.gov.au/public/rest/services/Landgate_Public_Maps/Cultural_and_Society/MapServer/1/query?where=status%3D%27Active%27&amp;text=&amp;objectIds=&amp;time=&amp;geometry=115.8860718954193%2C-32.01350664629256%2C115.9009673001542%2C-31.99754322345886&amp;geometryType=esriGeometryEnvelope&amp;inSR=4326&amp;spatialRel=esriSpatialRelIntersects&amp;relationParam=&amp;outFields=stopid%2Cstopname%2Cstoptype%2Cstatus%2Clocation%2Csuburb&amp;returnGeometry=true&amp;returnTrueCurves=false&amp;maxAllowableOffset=&amp;geometryPrecision=&amp;outSR=4326&amp;returnIdsOnly=false&amp;returnCountOnly=false&amp;orderByFields=stopname&amp;groupByFieldsForStatistics=&amp;outStatistics=&amp;returnZ=false&amp;returnM=false&amp;gdbVersion=&amp;returnDistinctValues=false&amp;resultOffset=&amp;resultRecordCount=75&amp;f=json">https://services.slip.wa.gov.au/public/rest/services/Landgate_Public_Maps/Cultural_and_Society/MapServer/1/query?where=status%3D%27Active%27&amp;text=&amp;objectIds=&amp;time=&amp;geometry=115.8860718954193%2C-32.01350664629256%2C115.9009673001542%2C-31.99754322345886&amp;geometryType=esriGeometryEnvelope&amp;inSR=4326&amp;spatialRel=esriSpatialRelIntersects&amp;relationParam=&amp;outFields=stopid%2Cstopname%2Cstoptype%2Cstatus%2Clocation%2Csuburb&amp;returnGeometry=true&amp;returnTrueCurves=false&amp;maxAllowableOffset=&amp;geometryPrecision=&amp;outSR=4326&amp;returnIdsOnly=false&amp;returnCountOnly=false&amp;orderByFields=stopname&amp;groupByFieldsForStatistics=&amp;outStatistics=&amp;returnZ=false&amp;returnM=false&amp;gdbVersion=&amp;returnDistinctValues=false&amp;resultOffset=&amp;resultRecordCount=75&amp;f=json</a>
ESRI - BusStops - Small - GET - JSON	<a href="https://services.slip.wa.gov.au/public/rest/services/Landgate_Public_Maps/Cultural_and_Society/MapServer/1/query?where=status%3D%27Active%27&amp;text=&amp;objectIds=&amp;time=&amp;geometry=&amp;geometryType=esriGeometryEnvelope&amp;inSR=&amp;spatialRel=esriSpatialRelIntersects&amp;relationParam=&amp;outFields=stopid%2Cstatus%2Cstopname%2Cstoptype&amp;returnGeometry=true&amp;returnTrueCurves=false&amp;maxAllowableOffset=&amp;geometryPrecision=&amp;outSR=4326&amp;returnIdsOnly=false&amp;returnCountOnly=false&amp;orderByFields=&amp;groupByFieldsForStatistics=&amp;outStatistics=&amp;returnZ=false&amp;returnM=false&amp;gdbVersion=&amp;returnDistinctValues=false&amp;resultOffset=&amp;resultRecordCount=25&amp;f=json">https://services.slip.wa.gov.au/public/rest/services/Landgate_Public_Maps/Cultural_and_Society/MapServer/1/query?where=status%3D%27Active%27&amp;text=&amp;objectIds=&amp;time=&amp;geometry=&amp;geometryType=esriGeometryEnvelope&amp;inSR=&amp;spatialRel=esriSpatialRelIntersects&amp;relationParam=&amp;outFields=stopid%2Cstatus%2Cstopname%2Cstoptype&amp;returnGeometry=true&amp;returnTrueCurves=false&amp;maxAllowableOffset=&amp;geometryPrecision=&amp;outSR=4326&amp;returnIdsOnly=false&amp;returnCountOnly=false&amp;orderByFields=&amp;groupByFieldsForStatistics=&amp;outStatistics=&amp;returnZ=false&amp;returnM=false&amp;gdbVersion=&amp;returnDistinctValues=false&amp;resultOffset=&amp;resultRecordCount=25&amp;f=json</a>

ESRI - Topo - Big - GET - Image	<a href="https://services.slip.wa.gov.au/public/rest/services/Landgate_Public_Services/Medium_Scale_Topo_Public_Services/MapServer/export?bbox=115%2C-35.5%2C127%2C-13.5&amp;bboxSR=4283&amp;layers=show%3A0%2C1%2C2%2C3&amp;layerDefs=&amp;size=500%2C500&amp;imageSR=4283&amp;format=png&amp;transparent=true&amp;dpi=&amp;time=&amp;layerTimeOptions=&amp;dynamicLayers=&amp;gdbVersion=&amp;mapScale=&amp;f=image">https://services.slip.wa.gov.au/public/rest/services/Landgate_Public_Services/Medium_Scale_Topo_Public_Services/MapServer/export?bbox=115%2C-35.5%2C127%2C-13.5&amp;bboxSR=4283&amp;layers=show%3A0%2C1%2C2%2C3&amp;layerDefs=&amp;size=500%2C500&amp;imageSR=4283&amp;format=png&amp;transparent=true&amp;dpi=&amp;time=&amp;layerTimeOptions=&amp;dynamicLayers=&amp;gdbVersion=&amp;mapScale=&amp;f=image</a>
ESRI - Topo - Small - GET - Image	<a href="https://services.slip.wa.gov.au/public/rest/services/Landgate_Public_Services/Medium_Scale_Topo_Public_Services/MapServer/export?bbox=115%2C-35.5%2C127%2C-13.5&amp;bboxSR=4283&amp;layers=show%3A0%2C1%2C2%2C3&amp;layerDefs=&amp;size=50%2C50&amp;imageSR=4283&amp;format=png&amp;transparent=true&amp;dpi=&amp;time=&amp;layerTimeOptions=&amp;dynamicLayers=&amp;gdbVersion=&amp;mapScale=&amp;f=image">https://services.slip.wa.gov.au/public/rest/services/Landgate_Public_Services/Medium_Scale_Topo_Public_Services/MapServer/export?bbox=115%2C-35.5%2C127%2C-13.5&amp;bboxSR=4283&amp;layers=show%3A0%2C1%2C2%2C3&amp;layerDefs=&amp;size=50%2C50&amp;imageSR=4283&amp;format=png&amp;transparent=true&amp;dpi=&amp;time=&amp;layerTimeOptions=&amp;dynamicLayers=&amp;gdbVersion=&amp;mapScale=&amp;f=image</a>

Table 9.2 - GME GET Requests

Test Name	URL
GME - AerialPhoto - Big - GET - Image	<a href="https://mapsengine.google.com/09372590152434720789-03311849775732631692-4/wms/?service=wms&amp;version=1.1.1&amp;request=GetMap&amp;srs=EPSG%3A4326&amp;format=image%2Fpng&amp;height=500&amp;width=500&amp;bbox=115.894855%2C-32.007438%2C115.895456%2C-32.006877&amp;layers=09372590152434720789-02344374277596842979-4&amp;styles=&amp;key=">https://mapsengine.google.com/09372590152434720789-03311849775732631692-4/wms/?service=wms&amp;version=1.1.1&amp;request=GetMap&amp;srs=EPSG%3A4326&amp;format=image%2Fpng&amp;height=500&amp;width=500&amp;bbox=115.894855%2C-32.007438%2C115.895456%2C-32.006877&amp;layers=09372590152434720789-02344374277596842979-4&amp;styles=&amp;key=</a>
GME - AerialPhoto - GetTileKVP - GET - Image	<a href="https://mapsengine.google.com/09372590152434720789-03311849775732631692-4/wmts/?service=wmts&amp;version=1.0.0&amp;request=GetTile&amp;exceptions=XML&amp;layer=09372590152434720789-02344374277596842979-4&amp;style=default&amp;format=image%2Fpng&amp;tilematrixset=EPSG%3A900913&amp;tilematrix=EPSG%3A900913%3A19&amp;tilerow=311390&amp;tilecol=430928&amp;key=">https://mapsengine.google.com/09372590152434720789-03311849775732631692-4/wmts/?service=wmts&amp;version=1.0.0&amp;request=GetTile&amp;exceptions=XML&amp;layer=09372590152434720789-02344374277596842979-4&amp;style=default&amp;format=image%2Fpng&amp;tilematrixset=EPSG%3A900913&amp;tilematrix=EPSG%3A900913%3A19&amp;tilerow=311390&amp;tilecol=430928&amp;key=</a>
GME - AerialPhoto - GetTileKVP2 - GET - Image	<a href="https://mapsengine.google.com/09372590152434720789-03311849775732631692-4/wmts/?service=wmts&amp;version=1.0.0&amp;request=GetTile&amp;exceptions=XML&amp;layer=09372590152434720789-02344374277596842979-4&amp;style=default&amp;format=image%2Fpng&amp;tilematrixset=EPSG%3A900913&amp;tilematrix=EPSG%3A900913%3A19&amp;tilerow=311389&amp;tilecol=430929&amp;key=">https://mapsengine.google.com/09372590152434720789-03311849775732631692-4/wmts/?service=wmts&amp;version=1.0.0&amp;request=GetTile&amp;exceptions=XML&amp;layer=09372590152434720789-02344374277596842979-4&amp;style=default&amp;format=image%2Fpng&amp;tilematrixset=EPSG%3A900913&amp;tilematrix=EPSG%3A900913%3A19&amp;tilerow=311389&amp;tilecol=430929&amp;key=</a>
GME - AerialPhoto - GetTileKVP3 - GET - Image	<a href="https://mapsengine.google.com/09372590152434720789-03311849775732631692-4/wmts/?service=wmts&amp;version=1.0.0&amp;request=GetTile&amp;exceptions=XML&amp;layer=09372590152434720789-02344374277596842979-4&amp;style=default&amp;format=image%2Fpng&amp;tilematrixset=EPSG%3A900913&amp;tilematrix=EPSG%3A900913%3A19&amp;tilerow=311388&amp;tilecol=430930&amp;key=">https://mapsengine.google.com/09372590152434720789-03311849775732631692-4/wmts/?service=wmts&amp;version=1.0.0&amp;request=GetTile&amp;exceptions=XML&amp;layer=09372590152434720789-02344374277596842979-4&amp;style=default&amp;format=image%2Fpng&amp;tilematrixset=EPSG%3A900913&amp;tilematrix=EPSG%3A900913%3A19&amp;tilerow=311388&amp;tilecol=430930&amp;key=</a>
GME - AerialPhoto - GetTileKVP4 - GET - Image	<a href="https://mapsengine.google.com/09372590152434720789-03311849775732631692-4/wmts/?service=wmts&amp;version=1.0.0&amp;request=GetTile&amp;exceptions=XML&amp;layer=09372590152434720789-02344374277596842979-4&amp;style=default&amp;format=image%2Fpng&amp;tilematrixset=EPSG%3A900913&amp;tilematrix=EPSG%3A900913%3A19&amp;tilerow=311388&amp;tilecol=430931&amp;key=">https://mapsengine.google.com/09372590152434720789-03311849775732631692-4/wmts/?service=wmts&amp;version=1.0.0&amp;request=GetTile&amp;exceptions=XML&amp;layer=09372590152434720789-02344374277596842979-4&amp;style=default&amp;format=image%2Fpng&amp;tilematrixset=EPSG%3A900913&amp;tilematrix=EPSG%3A900913%3A19&amp;tilerow=311388&amp;tilecol=430931&amp;key=</a>
GME - AerialPhoto - Small - GET - Image	<a href="https://mapsengine.google.com/09372590152434720789-03311849775732631692-4/wms/?service=wms&amp;version=1.1.1&amp;request=GetMap&amp;srs=EPSG%3A4326&amp;format=image%2Fpng&amp;height=50&amp;width=50&amp;bbox=115.894855%2C-32.007438%2C115.895456%2C-32.006877&amp;layers=09372590152434720789-02344374277596842979-4&amp;styles=&amp;key=">https://mapsengine.google.com/09372590152434720789-03311849775732631692-4/wms/?service=wms&amp;version=1.1.1&amp;request=GetMap&amp;srs=EPSG%3A4326&amp;format=image%2Fpng&amp;height=50&amp;width=50&amp;bbox=115.894855%2C-32.007438%2C115.895456%2C-32.006877&amp;layers=09372590152434720789-02344374277596842979-4&amp;styles=&amp;key=</a>
GME - AerialPhoto - WMSGetCapabilities - GET - XML	<a href="https://mapsengine.google.com/09372590152434720789-03311849775732631692-4/wms/?service=wms&amp;version=1.3.0&amp;request=GetCapabilities&amp;exceptions=XML&amp;key=">https://mapsengine.google.com/09372590152434720789-03311849775732631692-4/wms/?service=wms&amp;version=1.3.0&amp;request=GetCapabilities&amp;exceptions=XML&amp;key=</a>
GME - AerialPhoto - WMTSGetCapabilities - GET - XML	<a href="https://mapsengine.google.com/09372590152434720789-03311849775732631692-4/wmts/?service=wmts&amp;version=1.0.0&amp;request=GetCapabilities&amp;exceptions=XML&amp;key=">https://mapsengine.google.com/09372590152434720789-03311849775732631692-4/wmts/?service=wmts&amp;version=1.0.0&amp;request=GetCapabilities&amp;exceptions=XML&amp;key=</a>
GME - BusStops - AttributeFilter - GET - JSON	<a href="https://www.googleapis.com/mapsengine/v1/tables/09372590152434720789-08620406515972909896/features?version=published&amp;where=LOCATION%20CONTAINS%20%27Curtin%27%20AND%20SUBURB%20CONTAINS%20%27BENTLEY%27%20AND%20STATUS%20CONTAINS%20%27Active%27&amp;key=">https://www.googleapis.com/mapsengine/v1/tables/09372590152434720789-08620406515972909896/features?version=published&amp;where=LOCATION%20CONTAINS%20%27Curtin%27%20AND%20SUBURB%20CONTAINS%20%27BENTLEY%27%20AND%20STATUS%20CONTAINS%20%27Active%27&amp;key=</a>
GME - BusStops - Big - GET - JSON	<a href="https://www.googleapis.com/mapsengine/v1/tables/09372590152434720789-08620406515972909896/features?version=published&amp;maxResults=500&amp;key=">https://www.googleapis.com/mapsengine/v1/tables/09372590152434720789-08620406515972909896/features?version=published&amp;maxResults=500&amp;key=</a>

GME - BusStops - DistanceFilter - GET - JSON	<a href="https://www.googleapis.com/mapsengine/v1/tables/09372590152434720789-08620406515972909896/features?version=published&amp;select=STOPID%2CSTOPNAME%2CSTOPTYPE%2CSTATUS%2CLOCATION%2CSUBURB%2Cgeometry%2CST_Distance%28geometry%2CST_POINT%28115.895095%2C-32.007143%29%29%20as%20distance&amp;orderBy=distance&amp;limit=1&amp;where=STATUS%20CONTAINS%20%27Active%27&amp;key=">https://www.googleapis.com/mapsengine/v1/tables/09372590152434720789-08620406515972909896/features?version=published&amp;select=STOPID%2CSTOPNAME%2CSTOPTYPE%2CSTATUS%2CLOCATION%2CSUBURB%2Cgeometry%2CST_Distance%28geometry%2CST_POINT%28115.895095%2C-32.007143%29%29%20as%20distance&amp;orderBy=distance&amp;limit=1&amp;where=STATUS%20CONTAINS%20%27Active%27&amp;key=</a>
GME - BusStops - FeatureByID - GET - JSON	<a href="https://www.googleapis.com/mapsengine/v1/tables/09372590152434720789-08620406515972909896/features/1847?version=published&amp;maxResults=25&amp;key=">https://www.googleapis.com/mapsengine/v1/tables/09372590152434720789-08620406515972909896/features/1847?version=published&amp;maxResults=25&amp;key=</a>
GME - BusStops - IntersectFilter - GET - JSON	<a href="https://www.googleapis.com/mapsengine/v1/tables/09372590152434720789-08620406515972909896/features?version=published&amp;select=STOPID%2CSTOPNAME%2CSTOPTYPE%2CSTATUS%2CLOCATION%2CSUBURB%2Cgeometry&amp;orderBy=STOPNAME&amp;where=STATUS%20CONTAINS%20%27Active%27&amp;intersects=POLYGON%28%28115.8861773031899%20-31.99754322345886%2C%20115.8880417106566%20-32.00575228301201%2C%20115.8860718954193%20-32.01293450013328%2C%20115.894379415096%20-32.01350664629256%2C%20115.8939605952021%20-32.0119340941473%2C%20115.8934324720389%20-32.01154462279226%2C%20115.8937844220442%20-32.01021176821627%2C%20115.897005592728%20-32.0078328487639%2C%20115.8998850276332%20-32.00893175550439%2C%20115.9009673001542%20-32.00825477399166%2C%20115.8971881688856%20-32.00516242141832%2C%20115.8963344811994%20-32.0038082005204%2C%20115.8954071313707%20-32.0013269008619%2C%20115.8934248136717%20-31.99897245405646%2C%20115.890330338636%20-31.99790449476763%2C%20115.8861773031899%20-31.99754322345886%29%29&amp;limit=75&amp;key=">https://www.googleapis.com/mapsengine/v1/tables/09372590152434720789-08620406515972909896/features?version=published&amp;select=STOPID%2CSTOPNAME%2CSTOPTYPE%2CSTATUS%2CLOCATION%2CSUBURB%2Cgeometry&amp;orderBy=STOPNAME&amp;where=STATUS%20CONTAINS%20%27Active%27&amp;intersects=POLYGON%28%28115.8861773031899%20-31.99754322345886%2C%20115.8880417106566%20-32.00575228301201%2C%20115.8860718954193%20-32.01293450013328%2C%20115.894379415096%20-32.01350664629256%2C%20115.8939605952021%20-32.0119340941473%2C%20115.8934324720389%20-32.01154462279226%2C%20115.8937844220442%20-32.01021176821627%2C%20115.897005592728%20-32.0078328487639%2C%20115.8998850276332%20-32.00893175550439%2C%20115.9009673001542%20-32.00825477399166%2C%20115.8971881688856%20-32.00516242141832%2C%20115.8963344811994%20-32.0038082005204%2C%20115.8954071313707%20-32.0013269008619%2C%20115.8934248136717%20-31.99897245405646%2C%20115.890330338636%20-31.99790449476763%2C%20115.8861773031899%20-31.99754322345886%29%29&amp;limit=75&amp;key=</a>
GME - BusStops - Small - GET - JSON	<a href="https://www.googleapis.com/mapsengine/v1/tables/09372590152434720789-08620406515972909896/features?version=published&amp;maxResults=25&amp;select=STOPID%2CSTOPNAME%2CSTOPTYPE%2Cgeometry&amp;key=">https://www.googleapis.com/mapsengine/v1/tables/09372590152434720789-08620406515972909896/features?version=published&amp;maxResults=25&amp;select=STOPID%2CSTOPNAME%2CSTOPTYPE%2Cgeometry&amp;key=</a>



Table 9.3 - OGC GET Requests

Test Name	URL
OGC - AerialPhoto - GetTileKVP - GET - Image	<a href="https://imagery.slip.wa.gov.au/service?service=WMTS&amp;version=1.0.0&amp;request=GetTile&amp;layer=Locate&amp;style=default&amp;format=image%2Fpng&amp;tilematrixset=wa&amp;tilematrix=19&amp;tilerow=311390&amp;tilecol=430928">https://imagery.slip.wa.gov.au/service?service=WMTS&amp;version=1.0.0&amp;request=GetTile&amp;layer=Locate&amp;style=default&amp;format=image%2Fpng&amp;tilematrixset=wa&amp;tilematrix=19&amp;tilerow=311390&amp;tilecol=430928</a>
OGC - AerialPhoto - GetTileRestful - GET - Image	<a href="https://imagery.slip.wa.gov.au/wmts/Locate/wa/19/430928/311390.png">https://imagery.slip.wa.gov.au/wmts/Locate/wa/19/430928/311390.png</a>
OGC - BusStops - AttributeFilter - GET - JSON	<a href="https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs?service=WFS&amp;version=1.1.0&amp;request=GetFeature&amp;typeName=slip%3AAPT-008&amp;outputFormat=json&amp;maxFeatures=25&amp;filter=%3Cfilter%3E%3CPropertyIsLike%20wildCard%3D%27%2A%27%20singleChar%3D%27.%27%20escape%3D%27%21%27%3E%3CPropertyName%3Elocation%3C%2FPropertyName%3E%3CLiteral%3ECurtin%2A%3C%2FLiteral%3E%3C%2FPropertyIsLike%3E%3C%2FFilter%3E">https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs?service=WFS&amp;version=1.1.0&amp;request=GetFeature&amp;typeName=slip%3AAPT-008&amp;outputFormat=json&amp;maxFeatures=25&amp;filter=%3Cfilter%3E%3CPropertyIsLike%20wildCard%3D%27%2A%27%20singleChar%3D%27.%27%20escape%3D%27%21%27%3E%3CPropertyName%3Elocation%3C%2FPropertyName%3E%3CLiteral%3ECurtin%2A%3C%2FLiteral%3E%3C%2FPropertyIsLike%3E%3C%2FFilter%3E</a>
OGC - BusStops - AttributeFilter - GET - XML	<a href="https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs?service=WFS&amp;version=1.1.0&amp;request=GetFeature&amp;typeName=slip%3AAPT-008&amp;maxFeatures=25&amp;filter=%3Cfilter%3E%3CPropertyIsLike%20wildCard%3D%27%2A%27%20singleChar%3D%27.%27%20escape%3D%27%21%27%3E%3CPropertyName%3Elocation%3C%2FPropertyName%3E%3CLiteral%3ECurtin%2A%3C%2FLiteral%3E%3C%2FPropertyIsLike%3E%3C%2FFilter%3E">https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs?service=WFS&amp;version=1.1.0&amp;request=GetFeature&amp;typeName=slip%3AAPT-008&amp;maxFeatures=25&amp;filter=%3Cfilter%3E%3CPropertyIsLike%20wildCard%3D%27%2A%27%20singleChar%3D%27.%27%20escape%3D%27%21%27%3E%3CPropertyName%3Elocation%3C%2FPropertyName%3E%3CLiteral%3ECurtin%2A%3C%2FLiteral%3E%3C%2FPropertyIsLike%3E%3C%2FFilter%3E</a>
OGC - BusStops - Big - GET - JSON	<a href="https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs?service=WFS&amp;version=1.1.0&amp;request=GetFeature&amp;typeName=slip%3AAPT-008&amp;maxFeatures=500&amp;outputFormat=json&amp;filter=%3Cfilter%3E%3CPropertyIsEqualTo%3E%3CPropertyName%3Estatus%3C%2FPropertyName%3E%3CLiteral%3EActive%3C%2FLiteral%3E%3C%2FPropertyIsEqualTo%3E%3C%2FFilter%3E">https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs?service=WFS&amp;version=1.1.0&amp;request=GetFeature&amp;typeName=slip%3AAPT-008&amp;maxFeatures=500&amp;outputFormat=json&amp;filter=%3Cfilter%3E%3CPropertyIsEqualTo%3E%3CPropertyName%3Estatus%3C%2FPropertyName%3E%3CLiteral%3EActive%3C%2FLiteral%3E%3C%2FPropertyIsEqualTo%3E%3C%2FFilter%3E</a>
OGC - BusStops - Big - GET - XML	<a href="https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs?service=WFS&amp;version=1.1.0&amp;request=GetFeature&amp;typeName=slip%3AAPT-008&amp;maxFeatures=500&amp;filter=%3Cfilter%3E%3CPropertyIsEqualTo%3E%3CPropertyName%3Estatus%3C%2FPropertyName%3E%3CLiteral%3EActive%3C%2FLiteral%3E%3C%2FPropertyIsEqualTo%3E%3C%2FFilter%3E">https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs?service=WFS&amp;version=1.1.0&amp;request=GetFeature&amp;typeName=slip%3AAPT-008&amp;maxFeatures=500&amp;filter=%3Cfilter%3E%3CPropertyIsEqualTo%3E%3CPropertyName%3Estatus%3C%2FPropertyName%3E%3CLiteral%3EActive%3C%2FLiteral%3E%3C%2FPropertyIsEqualTo%3E%3C%2FFilter%3E</a>
OGC - BusStops - FeatureById - GET - JSON	<a href="https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs?service=WFS&amp;version=1.1.0&amp;request=GetFeature&amp;typeName=slip%3AAPT-008&amp;maxFeatures=25&amp;outputFormat=json&amp;filter=%3Cfilter%3E%3CPropertyIsEqualTo%3E%3CPropertyName%3Egid%3C%2FPropertyName%3E%3CLiteral%3E1540%3C%2FLiteral%3E%3C%2FPropertyIsEqualTo%3E%3C%2FFilter%3E">https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs?service=WFS&amp;version=1.1.0&amp;request=GetFeature&amp;typeName=slip%3AAPT-008&amp;maxFeatures=25&amp;outputFormat=json&amp;filter=%3Cfilter%3E%3CPropertyIsEqualTo%3E%3CPropertyName%3Egid%3C%2FPropertyName%3E%3CLiteral%3E1540%3C%2FLiteral%3E%3C%2FPropertyIsEqualTo%3E%3C%2FFilter%3E</a>
OGC - BusStops - FeatureById - GET - XML	<a href="https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs?service=WFS&amp;version=1.1.0&amp;request=GetFeature&amp;typeName=slip%3AAPT-008&amp;maxFeatures=25&amp;filter=%3Cfilter%3E%3CPropertyIsEqualTo%3E%3CPropertyName%3Egid%3C%2FPropertyName%3E%3CLiteral%3E1540%3C%2FLiteral%3E%3C%2FPropertyIsEqualTo%3E%3C%2FFilter%3E">https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs?service=WFS&amp;version=1.1.0&amp;request=GetFeature&amp;typeName=slip%3AAPT-008&amp;maxFeatures=25&amp;filter=%3Cfilter%3E%3CPropertyIsEqualTo%3E%3CPropertyName%3Egid%3C%2FPropertyName%3E%3CLiteral%3E1540%3C%2FLiteral%3E%3C%2FPropertyIsEqualTo%3E%3C%2FFilter%3E</a>

OGC - BusStops - GetCapabilities - GET - XML	<a href="https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs?service=WFS&amp;version=1.1.0&amp;request=GetCapabilities">https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs?service=WFS&amp;version=1.1.0&amp;request=GetCapabilities</a>
OGC - BusStops - IntersectFilter - GET - JSON	<a href="https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs?service=WFS&amp;version=1.1.0&amp;request=GetFeature&amp;typeName=slip%3AAPT-008&amp;outputFormat=json&amp;maxFeatures=500&amp;filter=%3Cwfs%3AGetFeature%20xmlns%3Aawfs%3D%22http%3A%2F%2Fwww.opengis.net%2Fwfs%22%20xmlns%3Aogc%3D%22http%3A%2F%2Fwww.opengis.net%2Fogc%22%20xmlns%3Aqml%3D%22http%3A%2F%2Fwww.opengis.net%2Fqml%22%20xmlns%3Acgf%3D%22http%3A%2F%2Fwww.opengis.net%2Fcite%2Fgeometry%22%20outputFormat%3D%22json%22%20version%3D%221.1.0%22%20service%3D%22wfs%22%3E%3Cwfs%3AQuery%20typeName%3D%22slip%3AAPT-008%22%3E%3Cg%3AFilter%3E%3Cg%3ABBOX%3E%3Cg%3APropertyName%3Ethe_geom%3C%2Fogc%3APropertyName%3E%3Cgml%3ABox%20srsName%3D%22EPSG%3A4326%22%3E%3Cgml%3Acoordinates%3E115.8860718954193%2C-32.01350664629256%20115.9009673001542%2C-31.99754322345886%3C%2Fgml%3Acoordinates%3E%3C%2Fgml%3ABox%3E%3C%2Fogc%3ABBOX%3E%3C%2Fogc%3AFilter%3E%3C%2Fwfs%3AQuery%3E%3C%2Fwfs%3AGetFeature%3E">https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs?service=WFS&amp;version=1.1.0&amp;request=GetFeature&amp;typeName=slip%3AAPT-008&amp;outputFormat=json&amp;maxFeatures=500&amp;filter=%3Cwfs%3AGetFeature%20xmlns%3Aawfs%3D%22http%3A%2F%2Fwww.opengis.net%2Fwfs%22%20xmlns%3Aogc%3D%22http%3A%2F%2Fwww.opengis.net%2Fogc%22%20xmlns%3Aqml%3D%22http%3A%2F%2Fwww.opengis.net%2Fqml%22%20xmlns%3Acgf%3D%22http%3A%2F%2Fwww.opengis.net%2Fcite%2Fgeometry%22%20outputFormat%3D%22json%22%20version%3D%221.1.0%22%20service%3D%22wfs%22%3E%3Cwfs%3AQuery%20typeName%3D%22slip%3AAPT-008%22%3E%3Cg%3AFilter%3E%3Cg%3ABBOX%3E%3Cg%3APropertyName%3Ethe_geom%3C%2Fogc%3APropertyName%3E%3Cgml%3ABox%20srsName%3D%22EPSG%3A4326%22%3E%3Cgml%3Acoordinates%3E115.8860718954193%2C-32.01350664629256%20115.9009673001542%2C-31.99754322345886%3C%2Fgml%3Acoordinates%3E%3C%2Fgml%3ABox%3E%3C%2Fogc%3ABBOX%3E%3C%2Fogc%3AFilter%3E%3C%2Fwfs%3AQuery%3E%3C%2Fwfs%3AGetFeature%3E</a>
OGC - BusStops - IntersectFilter - GET - XML	<a href="https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs?service=WFS&amp;version=1.1.0&amp;request=GetFeature&amp;typeName=slip%3AAPT-008&amp;maxFeatures=500&amp;filter=%3Cwfs%3AGetFeature%20xmlns%3Aawfs%3D%22http%3A%2F%2Fwww.opengis.net%2Fwfs%22%20xmlns%3Aogc%3D%22http%3A%2F%2Fwww.opengis.net%2Fogc%22%20xmlns%3Aqml%3D%22http%3A%2F%2Fwww.opengis.net%2Fqml%22%20xmlns%3Acgf%3D%22http%3A%2F%2Fwww.opengis.net%2Fcite%2Fgeometry%22%20outputFormat%3D%22json%22%20version%3D%221.1.0%22%20service%3D%22wfs%22%3E%3Cwfs%3AQuery%20typeName%3D%22slip%3AAPT-008%22%3E%3Cg%3AFilter%3E%3Cg%3ABBOX%3E%3Cg%3APropertyName%3Ethe_geom%3C%2Fogc%3APropertyName%3E%3Cgml%3ABox%20srsName%3D%22EPSG%3A4326%22%3E%3Cgml%3Acoordinates%3E115.8860718954193%2C-32.01350664629256%20115.9009673001542%2C-31.99754322345886%3C%2Fgml%3Acoordinates%3E%3C%2Fgml%3ABox%3E%3C%2Fogc%3ABBOX%3E%3C%2Fogc%3AFilter%3E%3C%2Fwfs%3AQuery%3E%3C%2Fwfs%3AGetFeature%3E">https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs?service=WFS&amp;version=1.1.0&amp;request=GetFeature&amp;typeName=slip%3AAPT-008&amp;maxFeatures=500&amp;filter=%3Cwfs%3AGetFeature%20xmlns%3Aawfs%3D%22http%3A%2F%2Fwww.opengis.net%2Fwfs%22%20xmlns%3Aogc%3D%22http%3A%2F%2Fwww.opengis.net%2Fogc%22%20xmlns%3Aqml%3D%22http%3A%2F%2Fwww.opengis.net%2Fqml%22%20xmlns%3Acgf%3D%22http%3A%2F%2Fwww.opengis.net%2Fcite%2Fgeometry%22%20outputFormat%3D%22json%22%20version%3D%221.1.0%22%20service%3D%22wfs%22%3E%3Cwfs%3AQuery%20typeName%3D%22slip%3AAPT-008%22%3E%3Cg%3AFilter%3E%3Cg%3ABBOX%3E%3Cg%3APropertyName%3Ethe_geom%3C%2Fogc%3APropertyName%3E%3Cgml%3ABox%20srsName%3D%22EPSG%3A4326%22%3E%3Cgml%3Acoordinates%3E115.8860718954193%2C-32.01350664629256%20115.9009673001542%2C-31.99754322345886%3C%2Fgml%3Acoordinates%3E%3C%2Fgml%3ABox%3E%3C%2Fogc%3ABBOX%3E%3C%2Fogc%3AFilter%3E%3C%2Fwfs%3AQuery%3E%3C%2Fwfs%3AGetFeature%3E</a>
OGC - BusStops - Small - GET - JSON	<a href="https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs?service=WFS&amp;version=1.1.0&amp;request=GetFeature&amp;typeName=slip%3AAPT-008&amp;propertyName=stopid%2Cstatus%2Cstopname%2Cstoptype%2Cthe_geom&amp;maxFeatures=25&amp;outputFormat=json&amp;filter=%3CFilter%3E%3CPropertyIsEqualTo%3E%3CPropertyName%3Estatus%3C%2FPropertyName%3E%3CLiteral%3EActive%3C%2FLiteral%3E%3C%2FPropertyIsEqualTo%3E%3C%2FFilter%3E">https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs?service=WFS&amp;version=1.1.0&amp;request=GetFeature&amp;typeName=slip%3AAPT-008&amp;propertyName=stopid%2Cstatus%2Cstopname%2Cstoptype%2Cthe_geom&amp;maxFeatures=25&amp;outputFormat=json&amp;filter=%3CFilter%3E%3CPropertyIsEqualTo%3E%3CPropertyName%3Estatus%3C%2FPropertyName%3E%3CLiteral%3EActive%3C%2FLiteral%3E%3C%2FPropertyIsEqualTo%3E%3C%2FFilter%3E</a>
OGC - BusStops - Small - GET - XML	<a href="https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs?service=WFS&amp;version=1.1.0&amp;request=GetFeature&amp;typeName=slip%3AAPT-008&amp;propertyName=stopid%2Cstatus%2Cstopname%2Cstoptype%2Cthe_geom&amp;maxFeatures=25&amp;filter=%3CFilter%3E%3CPropertyIsEqualTo%3E%3CPropertyName%3Estatus%3C%2FPropertyName%3E%3CLiteral%3EActive%3C%2FLiteral%3E%3C%2FPropertyIsEqualTo%3E%3C%2FFilter%3E">https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs?service=WFS&amp;version=1.1.0&amp;request=GetFeature&amp;typeName=slip%3AAPT-008&amp;propertyName=stopid%2Cstatus%2Cstopname%2Cstoptype%2Cthe_geom&amp;maxFeatures=25&amp;filter=%3CFilter%3E%3CPropertyIsEqualTo%3E%3CPropertyName%3Estatus%3C%2FPropertyName%3E%3CLiteral%3EActive%3C%2FLiteral%3E%3C%2FPropertyIsEqualTo%3E%3C%2FFilter%3E</a>
OGC - Topo - Big - GET - Image	<a href="https://www2.landgate.wa.gov.au/ows/wmspublic?service=WMS&amp;version=1.3.0&amp;request=GetMap&amp;layers=LGATE-010%2CLGATE-052%2CLGATE-053%2CLGATE-054&amp;srs=EPSG%3A4283&amp;bbox=110%2C-35.5%2C132%2C-13.5&amp;width=500&amp;height=500&amp;format=image%2Fpng&amp;transparent=true">https://www2.landgate.wa.gov.au/ows/wmspublic?service=WMS&amp;version=1.3.0&amp;request=GetMap&amp;layers=LGATE-010%2CLGATE-052%2CLGATE-053%2CLGATE-054&amp;srs=EPSG%3A4283&amp;bbox=110%2C-35.5%2C132%2C-13.5&amp;width=500&amp;height=500&amp;format=image%2Fpng&amp;transparent=true</a>

OGC - Topo - Small - GET - Image	<a href="https://www2.landgate.wa.gov.au/ows/wmspublic?service=WMS&amp;version=1.3.0&amp;request=GetMap&amp;layers=LGATE-010%2CLGATE-052%2CLGATE-053%2CLGATE-054&amp;srs=EPSG%3A4283&amp;bbox=110%2C-35.5%2C132%2C-13.5&amp;width=50&amp;height=50&amp;format=image%2Fpng&amp;transparent=true">https://www2.landgate.wa.gov.au/ows/wmspublic?service=WMS&amp;version=1.3.0&amp;request=GetMap&amp;layers=LGATE-010%2CLGATE-052%2CLGATE-053%2CLGATE-054&amp;srs=EPSG%3A4283&amp;bbox=110%2C-35.5%2C132%2C-13.5&amp;width=50&amp;height=50&amp;format=image%2Fpng&amp;transparent=true</a>
-------------------------------------	---

## **APPENDIX B WEB SERVICE POST REQUESTS**

Table 9.4 - Esri POST Requests

Test Name	ESRI - BusStops - AttributeFilter - POST - JSON	ESRI - BusStops - FeatureByID - POST - JSON	ESRI - BusStops - GetCapabilities - POST - JSON	ESRI - BusStops - IntersectFilter - POST - JSON
URL	<a href="https://services.slip.wa.gov.au/public/rest/services/Landgate_Public_Maps/Cultural_and_Society/MapServer/1/query">https://services.slip.wa.gov.au/public/rest/services/Landgate_Public_Maps/Cultural_and_Society/MapServer/1/query</a>	<a href="https://services.slip.wa.gov.au/public/rest/services/Landgate_Public_Maps/Cultural_and_Society/MapServer/1/query">https://services.slip.wa.gov.au/public/rest/services/Landgate_Public_Maps/Cultural_and_Society/MapServer/1/query</a>	<a href="https://services.slip.wa.gov.au/public/rest/services/Landgate_Public_Maps/Cultural_and_Society/MapServer/1/">https://services.slip.wa.gov.au/public/rest/services/Landgate_Public_Maps/Cultural_and_Society/MapServer/1/</a>	<a href="https://services.slip.wa.gov.au/public/rest/services/Landgate_Public_Maps/Cultural_and_Society/MapServer/1/query">https://services.slip.wa.gov.au/public/rest/services/Landgate_Public_Maps/Cultural_and_Society/MapServer/1/query</a>
Body	where=location+LIKE+%27%25Curtin%25%27+AND+status%3D%27Active%27+AND+suburb%3D%27BENTLEY%27&text=&objectIds=&time=&geometry=&geometryType=esriGeometryEnvelope&inSR=&spatialRel=esriSpatialRelIntersects&relationParam=&outFields=stopid%2Cstopnumber%2Cstatus%2Cstopname%2Clocation%2Cstoptype%2Csuburb%2Cbay%2Cshelter%2Casset%2Cweekpam%2Cweekppm%2Cweekoff%2Cweekrem%2Csatpam%2Csatppm%2Csatoff%2Csatrem%2Csunphpam%2Csunphppm%2Csunphoff%2Csunphrem%2Coid&returnGeometry=true&returnTrueCurves=false&maxAllowableOffset=&geometryPrecision=&outSR=4326&returnIdsOnly=false&returnCountOnly=false&orderByFields=&groupByFieldsForStatistics=&outStatistics=&returnZ=false&returnM=false&gdbVersion=&returnDistinctValues=false&resultOffset=&resultRecordCount=999&f=json	where=status%3D%27Active%27&text=&objectIds=1540&time=&geometry=&geometryType=esriGeometryEnvelope&inSR=&spatialRel=esriSpatialRelIntersects&relationParam=&outFields=%2A&returnGeometry=true&returnTrueCurves=false&maxAllowableOffset=&geometryPrecision=&outSR=4326&returnIdsOnly=false&returnCountOnly=false&orderByFields=&groupByFieldsForStatistics=&outStatistics=&returnZ=false&returnM=false&gdbVersion=&returnDistinctValues=false&resultOffset=&resultRecordCount=25&f=json	{ "f": "json" }	where=status%3D%27Active%27&text=&objectIds=&time=&geometry=115.8860718954193%2C-32.01350664629256%2C115.9009673001542%2C-31.99754322345886&geometryType=esriGeometryEnvelope&inSR=4326&spatialRel=esriSpatialRelIntersects&relationParam=&outFields=stopid%2Cstopname%2Cstoptype%2Cstatus%2Clocation%2Csuburb&returnGeometry=true&returnTrueCurves=false&maxAllowableOffset=&geometryPrecision=&outSR=4326&returnIdsOnly=false&returnCountOnly=false&orderByFields=&groupByFieldsForStatistics=&outStatistics=&returnZ=false&returnM=false&gdbVersion=&returnDistinctValues=false&resultOffset=&resultRecordCount=75&f=json

Test Name	ESRI - BusStops - Small - POST - JSON	ESRI - Topo - Big - POST - Image	ESRI - Topo - Small - POST - Image
URL	<a href="https://services.slip.wa.gov.au/public/rest/services/Landgate_Public_Maps/Cultural_and_Society/MapServer/1/query">https://services.slip.wa.gov.au/public/rest/services/Landgate_Public_Maps/Cultural_and_Society/MapServer/1/query</a>	<a href="https://services.slip.wa.gov.au/public/rest/services/Landgate_Public_Services/Medium_Scale_Topo_Public_Services/MapServer/export">https://services.slip.wa.gov.au/public/rest/services/Landgate_Public_Services/Medium_Scale_Topo_Public_Services/MapServer/export</a>	<a href="https://services.slip.wa.gov.au/public/rest/services/Landgate_Public_Services/Medium_Scale_Topo_Public_Services/MapServer/export">https://services.slip.wa.gov.au/public/rest/services/Landgate_Public_Services/Medium_Scale_Topo_Public_Services/MapServer/export</a>
Body	where=status%3D%27Active%27&text=&objectIds=&time=&geometry=&geometryType=esriGeometryEnvelope&inSR=&spatialRel=esriSpatialRelIntersects&relationParam=&outFields=stopid%2Cstatus%2Cstopname%2Cstoptype&returnGeometry=true&returnTrueCurves=false&maxAllowableOffset=&geometryPrecision=&outSR=4326&returnIdsOnly=false&returnCountOnly=false&orderByFields=&groupByFieldsForStatistics=&outStatistics=&returnZ=false&returnM=false&gdbVersion=&returnDistinctValues=false&resultOffset=&resultRecordCount=25&f=json	bbox=115%2C-35.5%2C127%2C-13.5&bboxSR=4283&layers=show%3A0%2C1%2C2%2C3&size=500%2C500&imageSR=4283&format=png&transparent=true&f=image	bbox=115%2C-35.5%2C127%2C-13.5&bboxSR=4283&layers=show%3A0%2C1%2C2%2C3&size=50%2C50&imageSR=4283&format=png&transparent=true&f=image

Table 9.5 - OGC POST Requests

Test Name	OGC - BusStops - AttributeFilter - POST - JSON	OGC - BusStops - AttributeFilter - POST - XML	OGC - BusStops - Big - POST - JSON	OGC - BusStops - Big - POST - XML
URL	https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs	https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs	https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs	https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs
Body	<pre>&lt;wfs:GetFeature xmlns:wfs="http://www.opengis.net/wfs" xmlns:ogc="http://www.opengis.net/ogc" xmlns:gml="http://www.opengis.net/gml" xmlns:cgf="http://www.opengis.net/cite/geom etry" outputFormat="json" version="1.1.0" service="wfs"&gt; &lt;wfs:Query typeName="slip:PTA-008"&gt; &lt;ogc:Filter&gt; &lt;ogc:And&gt; &lt;ogc:PropertyIsEqualTo&gt;  &lt;ogc:PropertyName&gt;status&lt;/ogc:PropertyName&gt; &lt;ogc:Literal&gt;Active&lt;/ogc:Literal&gt; &lt;/ogc:PropertyIsEqualTo&gt; &lt;ogc:PropertyIsEqualTo&gt;  &lt;ogc:PropertyName&gt;suburb&lt;/ogc:PropertyName&gt; &lt;ogc:Literal&gt;BENTLEY&lt;/ogc:Literal&gt; &lt;/ogc:PropertyIsEqualTo&gt; &lt;ogc:PropertyIsLike wildCard='*' singleChar='.' escape='! '&gt;  &lt;ogc:PropertyName&gt;location&lt;/ogc:PropertyName&gt; &lt;ogc:Literal&gt;*Curtin*&lt;/ogc:Literal&gt; &lt;/ogc:PropertyIsLike&gt; &lt;/ogc:And&gt; &lt;/ogc:Filter&gt; &lt;/wfs:Query&gt; &lt;/wfs:GetFeature&gt;</pre>	<pre>&lt;wfs:GetFeature xmlns:wfs="http://www.opengis.net/wfs" xmlns:ogc="http://www.opengis.net/ogc" xmlns:gml="http://www.opengis.net/gml" xmlns:cgf="http://www.opengis.net/cite/geom etry" version="1.1.0" service="wfs"&gt; &lt;wfs:Query typeName="slip:PTA-008"&gt; &lt;ogc:Filter&gt; &lt;ogc:And&gt; &lt;ogc:PropertyIsEqualTo&gt;  &lt;ogc:PropertyName&gt;status&lt;/ogc:PropertyName&gt; &lt;ogc:Literal&gt;Active&lt;/ogc:Literal&gt; &lt;/ogc:PropertyIsEqualTo&gt; &lt;ogc:PropertyIsEqualTo&gt;  &lt;ogc:PropertyName&gt;suburb&lt;/ogc:PropertyName&gt; &lt;ogc:Literal&gt;BENTLEY&lt;/ogc:Literal&gt; &lt;/ogc:PropertyIsEqualTo&gt; &lt;ogc:PropertyIsLike wildCard='*' singleChar='.' escape='! '&gt;  &lt;ogc:PropertyName&gt;location&lt;/ogc:PropertyName&gt; &lt;ogc:Literal&gt;*Curtin*&lt;/ogc:Literal&gt; &lt;/ogc:PropertyIsLike&gt; &lt;/ogc:And&gt; &lt;/ogc:Filter&gt; &lt;/wfs:Query&gt; &lt;/wfs:GetFeature&gt;</pre>	<pre>&lt;wfs:GetFeature xmlns:wfs="http://www.opengis.net/wfs" xmlns:ogc="http://www.opengis.net/ogc" xmlns:gml="http://www.opengis.net/gml" xmlns:cgf="http://www.opengis.net/cite/geom etry" outputFormat="json" version="1.1.0" service="wfs" maxFeatures="500"&gt; &lt;wfs:Query typeName="slip:PTA-008"&gt; &lt;ogc:Filter&gt; &lt;ogc:PropertyIsEqualTo&gt;  &lt;ogc:PropertyName&gt;status&lt;/ogc:PropertyName&gt; &lt;ogc:Literal&gt;Active&lt;/ogc:Literal&gt; &lt;/ogc:PropertyIsEqualTo&gt; &lt;/ogc:Filter&gt; &lt;/wfs:Query&gt; &lt;/wfs:GetFeature&gt;</pre>	<pre>&lt;wfs:GetFeature xmlns:wfs="http://www.opengis.net/wfs" xmlns:ogc="http://www.opengis.net/ogc" xmlns:gml="http://www.opengis.net/gml" xmlns:cgf="http://www.opengis.net/cite/geom etry" version="1.1.0" service="wfs" maxFeatures="500"&gt; &lt;wfs:Query typeName="slip:PTA-008"&gt; &lt;ogc:Filter&gt; &lt;ogc:PropertyIsEqualTo&gt;  &lt;ogc:PropertyName&gt;status&lt;/ogc:PropertyName&gt; &lt;ogc:Literal&gt;Active&lt;/ogc:Literal&gt; &lt;/ogc:PropertyIsEqualTo&gt; &lt;/ogc:Filter&gt; &lt;/wfs:Query&gt; &lt;/wfs:GetFeature&gt;</pre>

Test Name	OGC - BusStops - FeatureByID - POST - JSON	OGC - BusStops - FeatureByID - POST - XML	OGC - BusStops - GetCapabilities - POST - XML	OGC - BusStops - IntersectFilter - POST - JSON
URL	https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs	https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs	https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs	https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs
Body	<pre> &lt;wfs:GetFeature   xmlns:wfs="http://www.opengis.net/wfs"   xmlns:ogc="http://www.opengis.net/ogc"   xmlns:gml="http://www.opengis.net/gml"   xmlns:cgf="http://www.opengis.net/cite/geometry" outputFormat="json" version="1.1.0"   service="wfs" maxFeatures="25"&gt;   &lt;wfs:Query typeName="slip:PTA-008"&gt;     &lt;ogc:Filter&gt;       &lt;ogc:PropertyIsEqualTo&gt;  &lt;ogc:PropertyName&gt;gid&lt;/ogc:PropertyName&gt;       &lt;ogc:Literal&gt;1540&lt;/ogc:Literal&gt;     &lt;/ogc:PropertyIsEqualTo&gt;     &lt;/ogc:Filter&gt;   &lt;/wfs:Query&gt; &lt;/wfs:GetFeature&gt; </pre>	<pre> &lt;wfs:GetFeature   xmlns:wfs="http://www.opengis.net/wfs"   xmlns:ogc="http://www.opengis.net/ogc"   xmlns:gml="http://www.opengis.net/gml"   xmlns:cgf="http://www.opengis.net/cite/geometry" version="1.1.0" service="wfs"   maxFeatures="25"&gt;   &lt;wfs:Query typeName="slip:PTA-008"&gt;     &lt;ogc:Filter&gt;       &lt;ogc:PropertyIsEqualTo&gt;  &lt;ogc:PropertyName&gt;gid&lt;/ogc:PropertyName&gt;       &lt;ogc:Literal&gt;1540&lt;/ogc:Literal&gt;     &lt;/ogc:PropertyIsEqualTo&gt;     &lt;/ogc:Filter&gt;   &lt;/wfs:Query&gt; &lt;/wfs:GetFeature&gt; </pre>	<pre> &lt;GetCapabilities service="WFS"   xmlns="http://www.opengis.net/wfs"    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"    xsi:schemaLocation="http://www.opengis.net/wfs   http://schemas.opengis.net/wfs/1.1.0/wfs.xsd" /&gt; </pre>	<pre> &lt;wfs:GetFeature   xmlns:wfs="http://www.opengis.net/wfs"   xmlns:ogc="http://www.opengis.net/ogc"   xmlns:gml="http://www.opengis.net/gml"   xmlns:cgf="http://www.opengis.net/cite/geometry" outputFormat="json" version="1.1.0"   service="wfs"&gt;   &lt;wfs:Query typeName="slip:PTA-008"&gt;     &lt;ogc:Filter&gt;       &lt;ogc:BBOX&gt;  &lt;ogc:PropertyName&gt;the_geom&lt;/ogc:PropertyName&gt;       &lt;gml:Box srsName="EPSG:4326"&gt;         &lt;gml:coordinates&gt;115.8860718954193,- 32.01350664629256 115.9009673001542,- 31.99754322345886&lt;/gml:coordinates&gt;       &lt;/gml:Box&gt;     &lt;/ogc:BBOX&gt;     &lt;/ogc:Filter&gt;   &lt;/wfs:Query&gt; &lt;/wfs:GetFeature&gt; </pre>



Test Name	OGC - BusStops - IntersectFilter - POST - XML	OGC - BusStops - Small - POST - JSON	OGC - BusStops - Small - POST - XML
URL	<a href="https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs">https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs</a>	<a href="https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs">https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs</a>	<a href="https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs">https://www2.landgate.wa.gov.au/ows/wfspublic_4326/wfs</a>
Body	<pre> &lt;wfs:GetFeature   xmlns:wfs="http://www.opengis.net/wfs"   xmlns:ogc="http://www.opengis.net/ogc"   xmlns:gml="http://www.opengis.net/gml"   xmlns:cgf="http://www.opengis.net/cite/geometry" version="1.1.0" service="wfs"&gt;   &lt;wfs:Query typeName="slip:PTA-008"&gt;     &lt;ogc:Filter&gt;       &lt;ogc:BBOX&gt;  &lt;ogc:PropertyName&gt;the_geom&lt;/ogc:PropertyName&gt;       &lt;gml:Box srsName="EPSG:4326"&gt;         &lt;gml:coordinates&gt;115.8860718954193,- 32.01350664629256 115.9009673001542,- 31.99754322345886&lt;/gml:coordinates&gt;       &lt;/gml:Box&gt;     &lt;/ogc:BBOX&gt;   &lt;/ogc:Filter&gt; &lt;/wfs:Query&gt; &lt;/wfs:GetFeature&gt; </pre>	<pre> &lt;wfs:GetFeature   xmlns:wfs="http://www.opengis.net/wfs"   xmlns:ogc="http://www.opengis.net/ogc"   xmlns:gml="http://www.opengis.net/gml"   xmlns:cgf="http://www.opengis.net/cite/geometry" outputFormat="json" version="1.1.0"   service="wfs" maxFeatures="25"&gt;   &lt;wfs:Query typeName="slip:PTA-008"&gt;  &lt;wfs:PropertyName&gt;stopid&lt;/wfs:PropertyName&gt;  &lt;wfs:PropertyName&gt;status&lt;/wfs:PropertyName&gt;  &lt;wfs:PropertyName&gt;stopname&lt;/wfs:PropertyName&gt;  &lt;wfs:PropertyName&gt;stoptype&lt;/wfs:PropertyName&gt;  &lt;wfs:PropertyName&gt;the_geom&lt;/wfs:PropertyName&gt;     &lt;ogc:Filter&gt;       &lt;ogc:PropertyIsEqualTo&gt;  &lt;ogc:PropertyName&gt;status&lt;/ogc:PropertyName&gt;       &lt;ogc:Literal&gt;Active&lt;/ogc:Literal&gt;     &lt;/ogc:PropertyIsEqualTo&gt;     &lt;/ogc:Filter&gt;   &lt;/wfs:Query&gt; &lt;/wfs:GetFeature&gt; </pre>	<pre> &lt;wfs:GetFeature   xmlns:wfs="http://www.opengis.net/wfs"   xmlns:ogc="http://www.opengis.net/ogc"   xmlns:gml="http://www.opengis.net/gml"   xmlns:cgf="http://www.opengis.net/cite/geometry" version="1.1.0" service="wfs"   maxFeatures="25"&gt;   &lt;wfs:Query typeName="slip:PTA-008"&gt;  &lt;wfs:PropertyName&gt;stopid&lt;/wfs:PropertyName&gt;  &lt;wfs:PropertyName&gt;status&lt;/wfs:PropertyName&gt;  &lt;wfs:PropertyName&gt;stopname&lt;/wfs:PropertyName&gt;  &lt;wfs:PropertyName&gt;stoptype&lt;/wfs:PropertyName&gt;  &lt;wfs:PropertyName&gt;the_geom&lt;/wfs:PropertyName&gt;     &lt;ogc:Filter&gt;       &lt;ogc:PropertyIsEqualTo&gt;  &lt;ogc:PropertyName&gt;status&lt;/ogc:PropertyName&gt;       &lt;ogc:Literal&gt;Active&lt;/ogc:Literal&gt;     &lt;/ogc:PropertyIsEqualTo&gt;     &lt;/ogc:Filter&gt;   &lt;/wfs:Query&gt; &lt;/wfs:GetFeature&gt; </pre>

## **APPENDIX C LIST OF ABBREVIATIONS**

Table 9.6 - List of Abbreviations

Abbreviation	Expansion
AJAX	Asynchronuos JavaScript and XML
API	Application Programming Interface
CC BY-SA	Creative Commons Attribution Share Alike licence
CDMA	Code Division Multiple Access
CRUD	Create Read Update Delete
DSL	Digital Subscriber Line
EDGE	Enhanced Data rates for GSM Evolution
ESRI	No longer an acronym (previously Environmental Systems Research Institute)
GAE	Google Apps Engine
GeoJSON	Geographic JavaScript Object Notation
GIS	Geographic Information System
GLONASS	Globalnaya Navigazionnaya Sputnikovaya Sistema
GME	Google Maps Engine
GML	Geographic Markup Language
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile Communications (previously Groupe Special Mobile)
GUI	Graphic User Interface
HELIDEM	Helvetia-Italy Digital Elevation Model
HSDPA	High Speed Download Packet Access
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure (or HTTP over TLS, or HTTP over SSL)
ID	Shorthand for identity
IDE	Integrated Development Environment
JSON	JavaScript Object Notation
KML	Keyhole Markup Language
KVP	Key Value Pair
LAN	Local Area Network
LTE	Long Term Evolution
MIT	Massachusetts Institute of Technology
MVC	Model View Controller
NoSQL	Not strictly an acronym, a term for databases which eschew the standard relational database approach to storage and retrieval
NSW	New South Wales
OASIS	Organisation for the Advancement of Structured Information Standards
OGC	Open Geospatial Consortium
OS	Operating System

PC	Personal Computer
QGIS	No longer an acronym (previously Quantum GIS)
QLD	Queensland
REST	Representational State Transfer
SaaS	Software as a Service
SDI	Spatial Data Infrastructure
SLIP	Shared Location Information Platform
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
TaaS	Testing as a Service
TOC	Table of Contents
UML	Unified Modelling Language
URL	Uniform Resource Locator
WA	Western Australia
WALIS	Western Australian Land Information System
WFS	Web Feature Service
WMS	Web Map Service
WS	Web Service
WSDL	Web Service Description Language
XML	eXtensible Markup Language

### 9.1 HTTP Method Names

The following HTTP method names are printed in all capital letters by convention, this may confuse the reader causing them to believe they are acronyms.

Table 9.7 - HTTP Method Names

HTTP Method
GET
POST
PUT
PATCH
HEAD
OPTIONS
DELETE
TRACE
CONNECT