



**WebPark**  
**IST-2000-31041**

Geographically relevant information for mobile users in  
protected areas

# GI interoperability platform - review of OGC Specifications D3.1.1

Type PU\*

\*Type: CO-consortium, RE-restricted, PU-public

Report Version: 04

Report Preparation Date: 23 May 2002

Contract Start Date: 17 October 2001

Duration: 3 years

Project Co-ordinator: Geodan

Prepared by: Alistair Edwardes, Dirk Burghardt

Partners: CU, LNEC, SNP, EADS, GIUZ



Project funded by the European  
Community under the "Information  
Society Technology" Programme (1998-  
2002)

Project	Nature:	WP:	Delivery nr.	Written by	Delivery date	Contractual delivery
<b>Webpark</b>	<b>R</b>	<b>3100</b>	<b>3.1.1</b>	<b>GIUZ</b>	<b>25/10/02</b>	<b>01/11/02</b>

\*\*Nature: R-Report, P-Prototype, D-Demonstrator, CO-Consortium, PU-Public



**Abstract:**

This report provides a review, comparison and discussion of the OpenGIS Consortium (OGC) initiatives that are relevant to the WebPark project. For the purposes of comparison and discussion the initiatives are split into two groups, Open Web Services (OWS) that aim to facilitate the use of GI-Services over the Internet and the Open Location Services (OpenLS) which aim to facilitate the use of GI-Services over the mobile Internet. The aim of the report is to provide a basis for discussion and decision-making about the interoperability aspects of the WebPark platform.

The report first outlines the OGC Specification development process and goes on to provide an overview of each of the initiatives. It then provides a comparison of the work of the two groups, highlighting where they complement each other and where they diverge. Finally it discusses their potential adoption in the WebPark platforms and makes tentative recommendations on the interoperability strategy that might be followed by WebPark with regard to these discussions.

**Partner owning: GIUZ**

**Partners contributed: Geodan**



## Document information

### Codes

Full document name:	WP_D311_GI_interoperability_platform_OGC_v03.doc
Document code:	

### Revision history

Version number, date	Author(s)	Summary of changes to previous version
01, 23-May-2002	Alistair Edwardes, Dirk Burghardt	First version – draft
02, 17-Oct-2002	Alistair Edwardes, Dirk Burghardt	Final version – include comments from Geodan and reformat document
03, 25-Oct-2002	Debbrah Phan	Reformat document for compliance and report consistency including spelling check and content table
04, 22-Mar-2005	Evert van Kootwijk	Style improvements

### Distribution list

Name	Organisation
All Partners	All Partners

### Approval

Name	Date and signature	Remarks
All Partners		



---

## Table of contents

OGC Specifications and their potential for adoption by WebPark.....	5
Introduction.....	5
The OGC Consensus Specification Development Process .....	5
OpenLS Introduction.....	6
Aims.....	6
Services Overview.....	7
Service Identification and Definition .....	8
XML for Location Services (XLS) .....	9
OGC Web Services .....	10
OGC Web Map Service (WMS) .....	10
Style Layer Descriptors (SLD).....	11
OGC Web Feature Service (WFS).....	11
Comparison .....	12
Discussion .....	13
Recommendations .....	15
Conclusion.....	15
References: .....	16



---

## OGC Specifications and their potential for adoption by WebPark

### *Introduction*

This report provides a review, comparison and discussion of the OpenGIS Consortium (OGC) initiatives that are relevant to the WebPark project. For the purposes of comparison and discussion the initiatives are split into two groups, Open Web Services (OWS) that aim to facilitate the use of GI-Services over the Internet and the Open Location Services (OpenLS) which aim to facilitate the use of GI-Services over the mobile Internet. The aim of the report is to provide a basis for discussion and decision-making about the interoperability aspects of the WebPark platform.

The report first outlines the OGC Specification development process and goes on to provide an overview of each of the initiatives. It then provides a comparison of the work of the two groups, highlighting where they complement each other and where they diverge. Finally it discusses their potential adoption in the WebPark platforms and makes tentative recommendations on the interoperability strategy that might be followed by WebPark with regard to these discussions.

### *The OGC Consensus Specification Development Process*

The OGC process for all specifications evolves through the same set of sequences. It is an iterative process based on consensus and market requirements. An emerging market for a new technology is identified for standardisation and an initiative is formed. Initially interested commercial sponsors and participants are sought through a "Call for sponsors" and a "Request for Technology". Sponsors responding to these provide business requirements and financing to drive the development of specifications through test beds. The sponsors and the OGC engineers review the domain and identify potential interfaces and core services where interoperability is needed. These interfaces are prioritised into manageable work items for individual test beds, with priority based on the commercial requirements of the sponsors. In addition the preferred architecture for the tested is drafted. A "Call for participation" and a "Request for Quotation" is then issued. Interested participants respond to this, providing a cost quotation and a description of how they would participate in the tested and the resource that they will supply to it. Participants are then selected based on these responses and working groups are identified. A tested is then launched to developed and demonstrate interoperability solutions.

The tested provides a period of experimentation to investigate what constitutes an interface according to OGC practice and participant consensus. The tested involves the formation of candidate interfaces and a demonstration of their operation. The principal outputs of the tested are the Interoperability Program Reports (IPRs). These contain knowledge accrued through the tested and the agreed definitions of demonstration interfaces and encoding in the form of requirements documentation and engineering specifications. These are submitted to the OGC Technical Committee who turns this knowledge into draft specifications. The Technical Committee then issues a "Request for Comments" about these specifications to the industry at large. Comments received are integrated into the draft specifications, which are then published as formal Interface Implementation Specifications.



---

## OpenLS Introduction

### ***Aims***

The OpenLS Initiative is currently at the test bed stage of the standardisation process with work being carried out on candidate interface definitions and their demonstration through the first tested, OpenLS-1 (OLS1). The goal of OpenLS is to develop industry-wide consensus-based interface and encoding specifications in support of interoperable location services. The development work is focused on core location application services, location content and related support services. The principal requirement of OLS1 is the collaborative design of demonstration interface specifications for a subset of core services defined by the test bed sponsors. These are to satisfy the following use cases expressed as questions that a consumer might ask:

1. **What is the location of the nearest 'place/product/service' of a known TYPE?**
2. **Where is a specific 'place/product/service'?**
3. **Which route should I take?**
4. **Where is my route on a map?**
5. **What are the turn-by-turn driving directions for my route?**
6. **Where am I on a map?**
7. **Where is this place on a map?**
8. **What is this feature?**
9. **Where am I?**
10. **What are the traffic conditions along my route or in my area?**

**Services Overview**

Figure 1 describes the functional framework of OpenLS adapted from the “Call for Participation”.

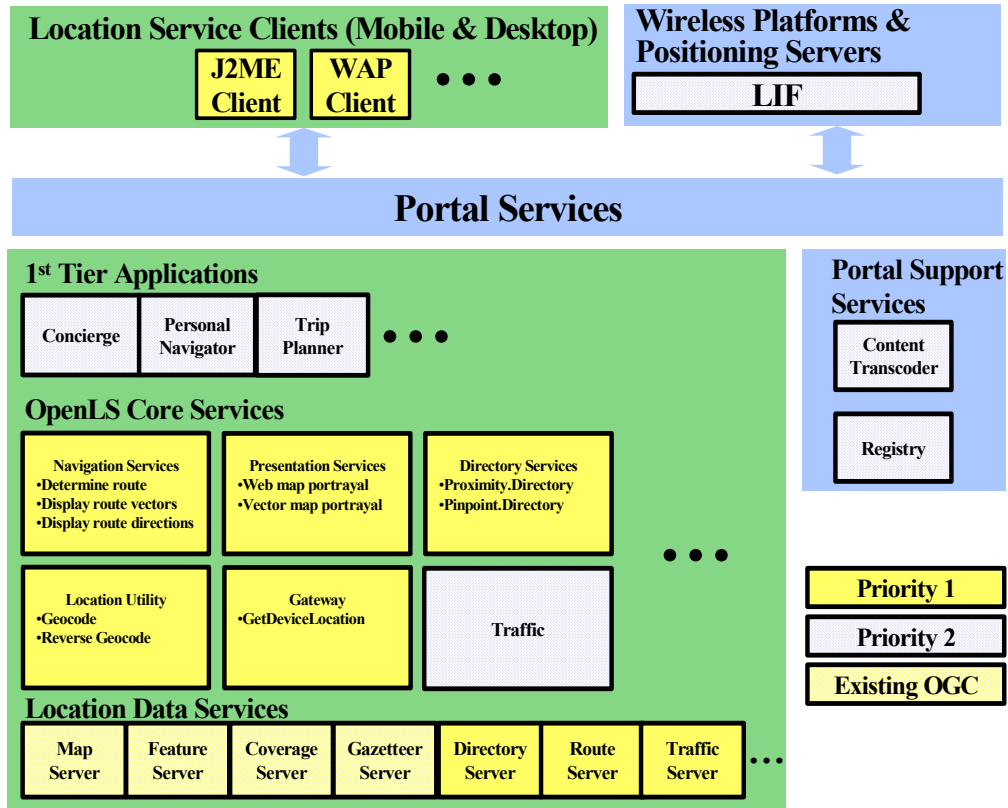


Figure 1. Location Services Framework: A Functional View (Source:[1], Notes: “DS” - Directory Service, “SLD” – Style Layer Descriptor)

Services marked as Priority 1 were the subject of the first OpenLS initiative (OLS1). Services marked Priority 2 are to be developed in future initiatives. Whilst some of the services are marked as “Existing OGC” it was recognised in the test bed that the specifications of these didn’t necessarily fit with the business requirements of OpenLS so work was also undertaken on refining and extending these to satisfy the needs of the test bed.

Figure 2 describes the priority reference architecture chosen by the Sponsors for implementing test bed activities.

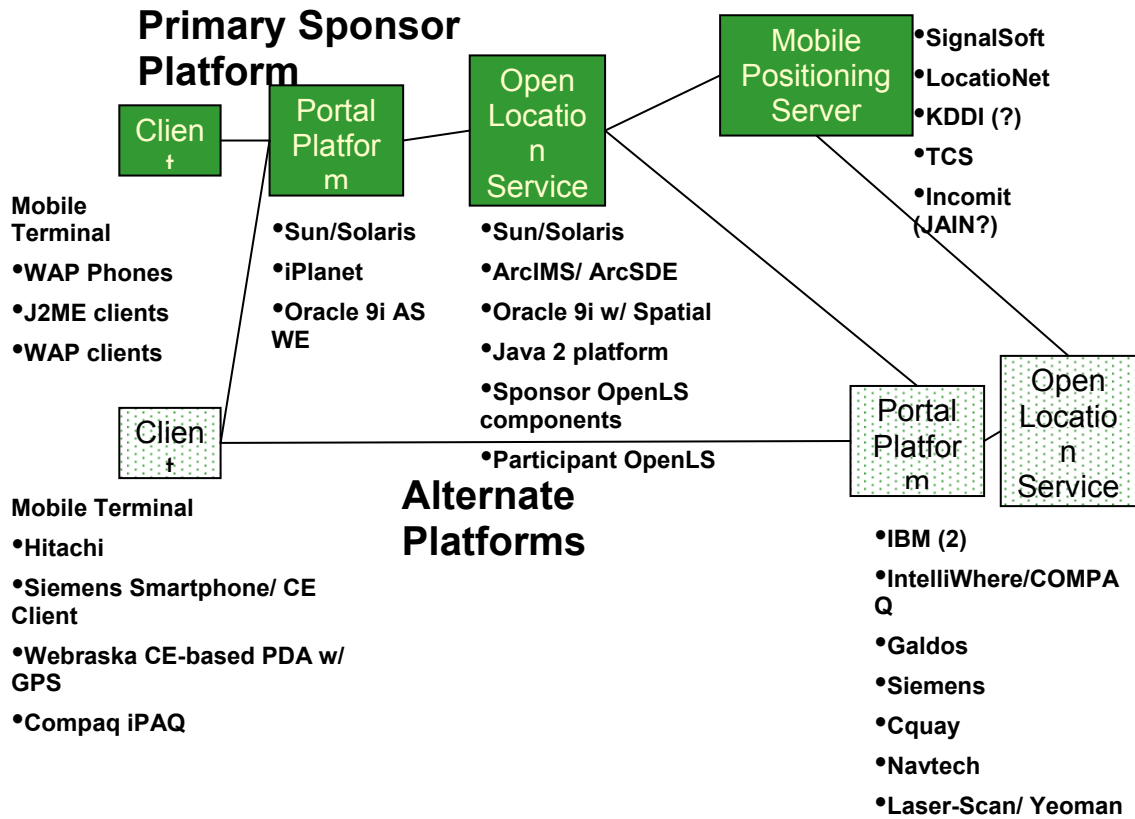


Figure 2. OpenLS-1 Platform (Source: [1])

The framework uses HTTP/POST for inter-process communication. Request-Response interfaces and data types are encoded using XML/Schema. WAP and J2ME are the reference platforms for location service client devices.

### Service Identification and Definition

The framework distilled the following core services with interfaces for standardisation

Service	Operations	Description
Navigation Services	Determine Route Service	Determine and store a route
	Display Route Vectors Service	Display a route geometry
	Display Route Directions Service	Provide turn-by-turn driving directions
Presentation Services	Web Map Portrayal Service	Display raster map
	Vector Map Portrayal Service	Display vector map
Directory Services	Proximity.Directory Service	Find nearest points of interest
	Pinpoint.Directory Service	Find the location of a specific point of interest
Location Utility Services	Geocoder service	Find position of an address



	<b>Reverse-Geocoder Service</b>	Find address of a position
	<b>Get Location Service</b>	Describe location of a position
<b>Gateway Services</b>	<b>Get Device Location Service</b>	Get the location of a device from a mobile positioning server

Each interfaces is defined in XML/Schema as a Request-Response pair. The interfaces pass information encoded in XML/Schema Abstract (or Complex) Data Types (ADTs). Specifying data types in XML/Schema provides a clean object-oriented approach to data structure definition. The data types and their relationships (inheritance, association, aggregation etc) can be modeled first in the Unified Modeling Language (UML) and these diagrams then converted into XML/Schema ADTs. The ADTs provide the key mechanism for passing data between services. Collectively these ADTs make up the encoding 'XML for Location Services' (XLS).

**XML for Location Services (XLS)**

XLS currently consists principally of the following ADTs

Abstract Data Type	Description
<b>Position ADT</b>	Point location in well-known coordinate system
<b>Address ADT</b>	Street address or intersection
<b>Point of Interest (POI) ADT</b>	The location where someone can find place, product or service
<b>Area of Interest (AOI) ADT</b>	A polygon, bounding box or circle used as a search template
<b>Location ADT</b>	A location (Position, Address or POI)
<b>Map ADT</b>	The portrayal of maps and feature overlays (routes & POI)
<b>Route Summary ADT</b>	Metadata pertaining to a route
<b>Route Geometry ADT</b>	Geometry data for a route
<b>Route Maneuvers ADT</b>	Navigation maneuver data for a route
<b>Route Directions ADT</b>	Turn-by-turn navigation instructions for a route

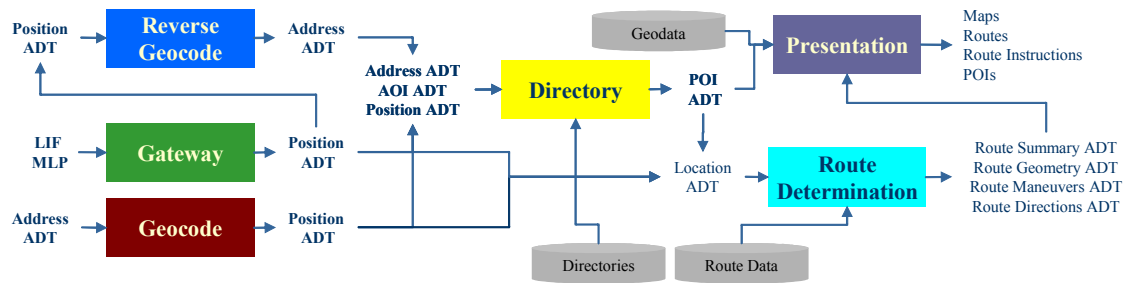


Figure 3. OpenLS Information Model

## OGC Web Services

Whilst originally test bed programs in themselves, the OGC Web Map Service and the OGC Feature Service can now be seen within the broader standardisation effort of the OGC Web Services Initiative (OWS). Figure 3. Describes how these services fit together in the OWS framework

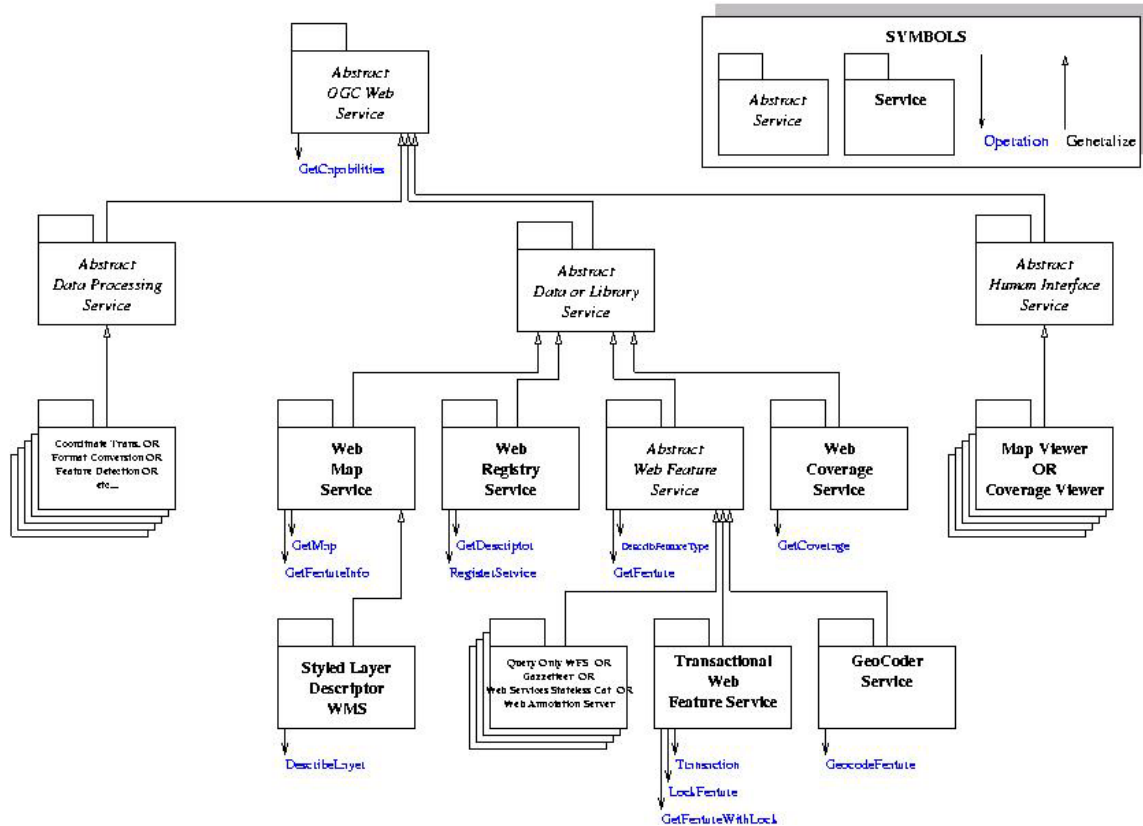


Figure 4. OWS Architecture [2]

OWS provides the framework to discover, access and integrate services residing on the web using standard protocols such as HTTP and XML. The aim is to provide seamless integrated access to geo-spatial data sources and geo-processing. OWS is at the stage of the second test bed (OWS1.2).

### OGC Web Map Service (WMS)

WMS has published formal implementation specifications for the production and delivery of maps over the web [3]. The specifications are mature and have been two iterations of specification ([4],[5]) as well as being adopted by ISO/TC211. The specification standardises the way in which a client may request a map and a server may publish information on its data holdings. The service defines three operations.



Operation	Description
<b>GetCapabilities</b>	Provides service-level metadata describing content of a data store and acceptable request parameters
<b>GetMap</b>	Provides mapping in response to geospatial and dimensional parameters
<b>GetFeatureInfo</b>	Allows a map to be queryable by providing information about features present in a map

The service may be accessed using HTTP GET or POST, though commonly most current implementations prefer the GET method. A client may specify the map layers it wants to see and styles in which they should be rendered, the bounding box of the map, it's spatial reference system, its width and height in pixels, the background color and whether the map should be transparent to allow overlay.

Web Map Servers maybe distributed on the web and maybe be cascading; that is different servers behave simultaneously as both clients and servers to other Web Map Servers.

### **Style Layer Descriptors (SLD)**

An extension to the Web Map Server is the Style Layer Descriptor Specification that is currently at the stage of "Request for Comments" [6]. This provides a more powerful method for cartographic visualisation, allowing more control over how and what features are to be rendered. It is an XML encoding that enables a WMS to query and retrieve features from a web server and apply client-defined user styling preferences to them. It also provides a cleaner separation between the presentation of data and the description of content. SLD defines the additional operations of a web map server; DescribeLayer, GetLegendGraphic, GetStyles, PutStyles

### **OGC Web Feature Service (WFS)**

The OGC Web Feature Service is an Implementation Specification [7] that provides interfaces for supporting transactions and accessing and manipulating OGC simple features [8] over the web. Figure 4. describes the WFS architecture



Figure 5. WFS architecture (adapted from [7])

The WFS uses GML [9] to encode the Simple Feature abstraction [8] and the OGC Filter Encoding Specification [10] to allow the formation of complex queries. It presents the following interfaces

Interface	Description
<b>GetCapabilities</b>	Provides information on what Feature types can be served and what operations are supported
<b>DescribeFeatureType</b>	Describe structure of a Feature type (returns description in XML/Schema)



<b>GetFeature</b>	Retrieve instance of a Feature, which properties and how to constrain the query (spatially and spatially)
<b>Transaction</b>	Provides support for database transactions
<b>Lock Feature</b>	Lock a feature during a transaction in order to delete or update it – performs serialisation of request to maintain database integrity during simultaneous transactions.

In practice WFSes generally come in two forms, those that support transactions and those that don't. Those that support transactions allow Features to be created, deleted and updated. Those that don't are essentially 'read only' WFSes that only allow data stores to be queried and Features to be retrieved.

As with WMSes, WFSes may be cascading with a single request querying more than one data store. Such requests are termed Heterogeneous requests. Requests on a single data store are termed Homogeneous.

### **Comparison**

Key to the distinction between OpenLS and OGC Web Services is that OpenLS provides interfaces to core location application services in response to the OpenLS use cases and not interfaces to location data services such as WMS and WFS. The aim for OpenLS is to demonstrate and standardise interfaces to the services that that add value to data by integrating geo-processing with geospatial data. Hence OpenLS operates at a higher tier of the service reference model ([11],[12]) than OGC Web services. However in implementation terms this distinction isn't so strong, for example a WFS or a WMS could be used as the basis for the OpenLS services or an alternative non-WFS/non-WMS solution could also be used. Hence OpenLS does not aim to define the architecture for location services but instead a framework.

Associated with this distinction was the need for OpenLS interfaces to shield users, developers and wireless providers from "GIS Complexity". Companies operating in the location based services market could not be assumed to have or indeed need to have, an in depth knowledge of handling spatial data. Since their core competences are likely to lie in other areas such as consumer marketing and service provision. Therefore, in order to achieve the widest use of location in mobile services, what are required are tools, which allow the transparent integration of geographic information within service-oriented applications. For example, the lack of geospatial knowledge of a company wishing to provide a service for finding hotels using a mobile device, should not preclude it from including location based tools in its portal to assist this. Hence a core goal of OpenLS is to provide simple and essential interfaces to the mainstream.

The second distinction is in OpenLS's use of 'legacy' OGC specifications and encoding and its use of specifications currently under development by other OGC groups. Because the aim of the OpenLS test bed is to develop services and interface in response to specific business requirements, there are parts of existing, complementary specifications which are irrelevant and other parts that are not sufficiently comprehensive to meet these needs. OpenLS does look to integrating relevant parts from existing specifications as far as possible, and where sufficiently stable, those currently in the process of being developed by other groups. However, there are likely to be areas where there is significant divergence, particularly in the case where similar emerging standards being developed by other initiatives. For example whilst the OpenLS Web Map Portrayal Service interface takes account of richer control for styles, layers and map overlays. The technique they ultimately choose with which to provide this functionality will



possibly be quite different from the Style Layer Descriptors specification. Such differences could represent significant factors in the decision to adopt one specification over the other.

The third distinction is that all the different specifications discussed are at different stages in the standardisation process. OpenLS in particular is a long way from becoming an implementation specification. This is not to say that commercial parties will not produce software in accordance to the demonstration interfaces being developed, they certainly will given that this is one of the major incentives of being a participant in the process. However it does mean a true comparison amongst specifications is bound to be somewhat flawed and that the availability of stable interfaces to parties outside the process with which to base development is very limited.

Related to this distinction is also the fact that different specifications make use of different technologies with which to define interfaces and encoding. In particular, XML/Schema wasn't available for the implementation specification of WMS and WFS, though it is now being integrated. Instead these mainly use DTD XML and HTTP to define interfaces. Whilst on the one hand this distinction may be resolvable through the XLST conversions or wrappers to the web services, on the other the use of XML/Schema features such as namespaces and complex data types could make conversions cumbersome and lossy. For example the WMS GetMap interface defines access to maps using parameters passed in a standard HTTP Request, either URL encoded for GET or in the request body for POST, whereas the OpenLS Web Map Portrayal Service encodes request parameters in and XML/Schema complex data type (MapADT) POSTed in the request body.

The manner in which XML has been used in the different specifications has also been strongly influenced by the business requirements of the respective initiatives. For OpenLS a requirement was that client devices should be able to parse XML directly, using parsers such as kDOM and kXML. This meant that in designing Schema data types a consideration of the limited resources available to clients has been made. For example, XLS encoding are designed such that they have a small footprint and multiple nesting in element trees and properties has been avoided. These requirements thus place constraints on simplified feature model supported.

### **Discussion**

(Note: This section will need to be expanded on with consortium partner comments)

In considering the use of OGC Specifications in WebPark the following issues are considered;

- The appropriateness of the specifications to WebPark's needs
- The volatility of the specifications
- The availability of specifications on which to perform development
- The level of effort required to use the specification in WebPark.

The appropriateness of the specifications is affected by several factors; the synergy between requirements defining the initiatives, the future developments in specifications that will take place during the course of the WebPark project, and the framework for the design of specifications.

Most important is how well the business requirements that direct the various initiatives marry with the needs of WebPark. For OpenLS the requirements and their priorities very tightly define the framework in which specification development is being undertaken. The fact that these specifications are being developed for the location based services industry should in itself add



weight to their consideration since they should have a priori commonalities. However, the priorities guiding much of the current development work are not really relevant to WebPark's needs. For example, the current work on interfaces and encoding for navigation, location utility (geocoding and reverse geocoding) and directory services has little obvious scope within WebPark since these are being designed for different environments, both in terms of mobility (mainly in-vehicle) and landscape (mainly urban streets), to that of a National Park. On the other hand development work on map portrayal and core encoding relating to the simple feature model are directly relevant. This is not to say that future work and future test beds will not have a greater synergy with WebPark, just that OpenLS aims to prioritise the development of specifications by their market relevance. The most currently relevant markets have been identified by sponsors as being initially for 'Where's my nearest?' and 'How do I get to?' type services. However future test beds are likely to be focused on more advanced services and frameworks.

For OWS based services the goal is more with the provision of geospatial data over the web to both consumer and business enterprises. Because of the nature of these initiatives a far greater level of GIS complexity can be afforded, though as with all OGC practice care is taken to avoid creating barriers to adoption in mainstream IT. Hence the interfaces are more detailed and comprehensive, aiming to satisfy a broader array of different types of deployments. However this also means that the encodings are heavier and may include functionality not required by WebPark. In addition because these services are at the data management level of the reference model [11, 12] they will not be complete enough to satisfy interoperability needs in the application tier.

The final consideration with regard to appropriateness is in the design of specifications themselves and their envisaged deployment. OpenLS envisages the use of XML to communicate data amongst applications and in addition makes allowances for its use directly by clients, both in presenting responses and formulating requests. This assumption puts considerable weight on the thickness of clients and the weight of the data encoding which may not be appropriate for WebPark. Of course, this function can be carried out by proxy on the server side, but this may create unnecessary complexity in the WebPark architecture and in creating a light encoding may not be sufficiently complete. That said it will be necessary to design encodings to communicate information between services in the application layer of the WebPark architecture anyway and mechanism (Schema ADTs) and encodings of OpenLS could present an attractive method for achieving this. The prospect of future convergence between the requirements of WebPark and OpenLS also make this channel more appealing. OWS on the other hand accounts for no such mechanism, instead providing more comprehensive generic encodings, such as GML. However these may be too detailed for the requirements of WebPark and do not provide the necessary lightweight encodings conducive to passing information between services at the application layer.

The next consideration is in the volatility of the specifications, that is how stable they are to develop against. OpenLS is still at the pre-demonstration stage, though it can be assumed that the basic interface and encodings currently defined will persist through the specification process. However currently, even the core encodings are incomplete and the readiness of these with respect to the time-line of WebPark would need to be investigated further, that said the test bed is scheduled for completion by the summer of 2002 with demonstrations of the interfaces in the autumn. Despite this, developing against beta or "Version 1.0" releases always carries risks and has the overhead of requiring close coordination with the OGC to keep track of modifications. The relevant OWS specifications on the other hand are stable and ready for commercial implementation, with the exception of Style Layer Descriptors specification which is at the final stages. However since in the main WebPark would be looking to use COTS implementations of



these at the data layer their readiness may not be as important as their availability in commercial software packages.

Considering availability, the OWS specifications are publicly available to the public and OGC members on the OGC website. The OpenLS specifications are currently only available to sponsors and participants of OLS1, membership of which is now closed. The WebPark consortium would therefore have to either; wait until the specifications were publicly available, have a partner or partners participate in the next stage of the test bed (OLS1.1), or purchase an implementation from OLS participant or work with one to gain access to OpenLS knowledge. In the two former cases this would only be possible in 2003, in the later case a suitable third-party would have to be found.

Finally, considering the level of effort required, in both cases the application layer will largely have to be developed in house. This would in either case, almost certainly, be on top of an OWS data layer (WMS and WFS). If the OpenLS specifications were used to guide application service development, savings could be made at the analysis and design stage. On the one hand this would also ensure that the WebPark architecture adhered to a set of standards but on the other could be at the risk of incompatibility with other emerging OGC specifications, e.g. the SLD and GML3.0, which may hamper efforts at a later date. In addition this may carry significant integration overheads to align the two OGC models. Alternatively if the application service layer was built directly on top of an OWS data service layer more effort would be required in the design stage of the application layer and this would also introduce proprietary interfaces. However, it would ensure a better fit with the requirements of the WebPark consortium, be built on mature and stable standards platform and reduce the risk of incompatibility with other OWS services.

### ***Recommendations***

Use OWS (WMS, WFS, SLD, GML) to provide access to data services – this will ensure development on a stable set of standards through COTS software  
Adopt the framework of OpenLS for the application service layer (XML/Schema, ADTs) and where relevant the ADT encodings. This will ensure that if an OpenLS compliant platform is desired at a future point this can be achieved relatively easily through XLST translations.  
Participate in or formally observe OLS1.1 – This would provide early access to specifications and enable the consortium to adopt new more relevant encodings that are developed. In addition it will provide an opportunity to standardise the consortium's own interfaces.

### ***Conclusion***

To be added after a discussion and decision has been reached.



---

**References:**

- [1] Open Location Services 1.0, 'Call for Participation', <http://www.openls.org>
- [2] Open Web Services 1.2 "Request for Quotation" <http://ip.opengis.org/ows1.2/index.html>
- [3] Open Web Map Server Interfaces Implementation Specification 1.1.1, <http://www.opengis.org/techno/implementation.htm>
- [4] Open Web Map Server Interfaces Implementation Specification 1.1.0, <http://www.opengis.org/techno/implementation.htm>
- [5] Open Web Map Server Interfaces Implementation Specification 1.0.0, <http://www.opengis.org/techno/implementation.htm>
- [6] OpenGIS Styled Layer Descriptor Implementation Specification <http://www.opengis.org/info/techno/rfc14info.htm>
- [7] Web Feature Server Specification <http://www.opengis.org/techno/implementation.htm>
- [8] Open GIS Abstract Feature Specification <http://www.opengis.org/techno/abstract.htm>
- [9] Geography Markup Language (GML) Implementation Specification <http://www.opengis.org/techno/implementation.htm>
- [10] Filter Encoding Implementation Specification <http://member.opengis.org/tc/archive/arch01.htm>
- [11] ISO 19101 Geographic Information – Reference Model
- [12] Open Systems Environment (OSE): Architectural Framework for Information Infrastructure, Fredrik Shultz, NIST Special Publication 500-232 December 1995