



View500

View500 White Paper The XML Enabled Directory

Author Andrew Sciberras
Date: 22nd January 2009
Address: PO Box 462
Kew VIC 3101
Contact Details: Australia: +61 3 9851 8600
Singapore: +65 6336 4780
Version: 0.6

The copyright of this document belongs to eB2Bcom. This document is intended to inform the reader of eB2Bcom's products, services, or views. Whilst every effort is made to ensure accuracy, the reader should take appropriate and diligent steps to ensure that the information is relevant to the reader's requirements.

CREDENTIALS	3
HISTORY OF VIEW500.....	3
View500 is Born	3
Setting the Standard.....	4
Client Adoption.....	4
XML Adoption	4
XML ENABLED DIRECTORY.....	6
XML Based Directory Protocols.....	7
XML Internet Directly Mapped (IDM)	7
XML Lightweight Directory Access Protocol (LDAP).....	7
XML Rendition of Directory Data	7
User Defined Syntaxes	8
XML Component Matching.....	8
APPLICABILITY AND USAGE EXAMPLES	10
Unified Policy and Identity Store	10
ebXML Registry Service	12
CONCLUSION	13

Credentials

eB2Bcom is the owner and developer of View500 directory technology that is deployed in Government, Defence, and private sector companies throughout the world. This technology has some unique features that deliver superior performance and capabilities for various applications; the following is one such case.

History of View500

It was in the 1980's that electronic directories were born. Research in the early 1980's by Xerox led to the development of Grapevine and Clearinghouse, the first distributed directory. With the Internet came a specialised distributed directory service called the Domain Name System, better known as DNS.

In the 1980's, the International Telegraph and Telephone Consultative Committee (CCITT), now known as International Telecommunications Union (ITU) was interested in a directory to provide a white pages service.

In parallel with this, the International Organization for Standardization (ISO) and the European Computer Manufacturers Association (ECMA) were interested in an OSI (Open Systems Interconnection) name service.

The two initiatives eventually merged and a collaborative effort was formed to define a general purpose directory standard. The first joint meeting between CCITT and the ISO was held in Melbourne in April 1986¹.

View500 is Born

In 1986 the Telecom (now Telstra) Research Laboratories (TRL) developed a prototype electronic directory system based on these emerging international standards and began investigating possible user interfaces for online public directories. It was this effort that marked the beginning of View500.

Consequently, the View500 team at TRL were heavily involved with defining the X.500 Standard from 1986 onwards and in the development of the View500 Directory Server. View500 was designed from "the ground up" to be a hierarchical directory server, to accommodate both the human interface to white pages and the machine interfaces associated with a distributed computing environment. Telecom, at that point Australia's largest telecommunications and directory services provider (a position still maintained) also ensured that View500's design catered for the storage and performance requirements for multimillion user environments, namely, the Australian population.

The final editing meeting for the first edition of the joint standard occurred in October 1988, with TRL continuing its significant contribution².

In 1989 View500 was first used within a commercial environment, being deployed within Telecom as the Corporate Electronic Directory. Based on the X.500 recommendations it delivered extensive facilities for update and retrieval of organizational and staff details via a user interface. It also delivered extensive savings in printing costs.

¹ <http://sec.cs.kent.ac.uk/x500book/Chapter.1/Chapter1.htm>

² <http://www.coxhill.com/trlhistory/history/achiev.htm>

Setting the Standard

Following the publication of Edition 1 of X.500, further enhancements and extensions were incorporated into the standard. The View500 team at TRL led this major international standardisation effort to provide enhanced security, search facilities, and information formatting and retrieval mechanisms. These significant enhancements were included within the X.500 standard in its 2nd Edition.

Currently 5 Editions of the X.500 standard have been published and the 6th Edition drafts are in progress. To this day, eB2Bcom (lead by Dr. Steven Legg, the View500 Product Architect) still actively participates in the X.500 Standards development.

Client Adoption

Whilst the X.500 standardisation provided a solid foundation for the directory information models, security framework, replication and distributed operation services, it had characteristics that didn't compliment the current state of technology. In order for an X.500 client application to interact with an X.500 DSA, the client application had to implement the full OSI protocol stack and be able to interpret complex ASN.1 data types.

This prompted the development of a new standardisation effort called LDAP, the Lightweight Directory Access Protocol. The Lightweight Directory Access Protocol (LDAP) is an Internet protocol for accessing distributed directory services that act in accordance with X.500 data and service models³.

As with the X.500 standardisation effort, the View500 team participated in defining the LDAP standard and implementing it into View500. By implementing LDAP support into View500, the product became readily usable by lightweight client applications.

Two of the editors of the latest core LDAPv3 specifications, published in June 2006, are from the current View500 team. Dr. Steven Legg (View500 Product Architect) is the editor of RFC4517⁴ and Andrew Sciberras (Solution Architect) is the editor of RFC4519⁵.

By being easily accessible to client applications, View500 has become a centralised component within many organizations. LDAP enabled applications can leverage it as a provider of information or security services, including (but not limited to) authentication, authorization data (including security policies), white pages, reporting, graphical charting, provisioning and as a PKI distribution point.

XML Adoption

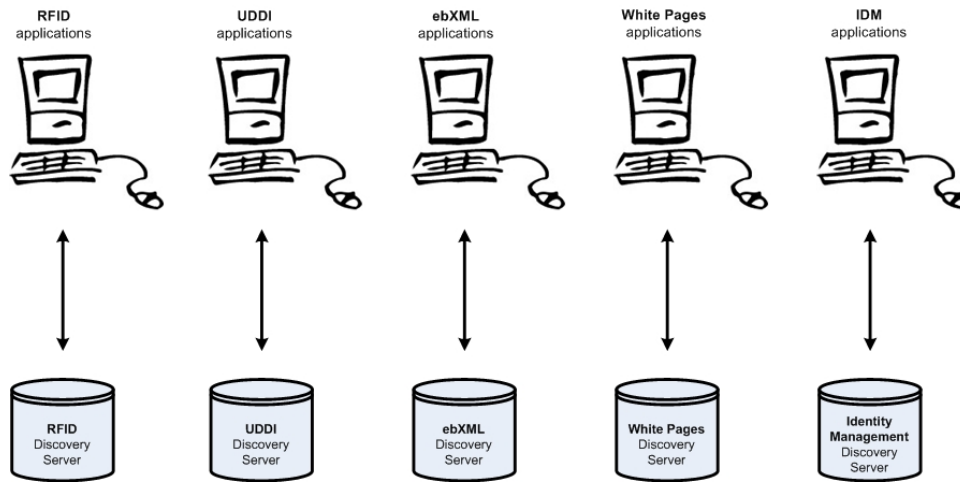
Today, XML centric technologies are being widely adopted and employed. Whilst the X.500 and LDAP technology provide optimised data storage and retrieval capabilities, it is not suited to XML based services. As a result, XML developments are forging ahead and making use of inferior or inappropriate technology.

³ RFC4510 - <http://www.ietf.org/rfc/rfc4510.txt>

⁴ RFC4517 - <http://www.ietf.org/rfc/rfc4517.txt>

⁵ RFC4519 - <http://www.ietf.org/rfc/rfc4519.txt>

The practice of developing technologies or applications that utilise their own data store has led to costly administrative and security issues. Initiatives such as Identity Management are attempting to remedy these issues.



Due to the general lack of XML support in today's X.500 and LDAP directory servers, emerging XML technologies are once again making use of application specific repositories.

However, leveraging proven technology (X.500 and LDAP) and learning from past mistakes, the View500 team (lead by Dr. Steven Legg) have instigated The XML Enabled Directory framework (see <http://www.xmlled.info>).

These capabilities have been built into View500. At the same time, the team has developed and published the XML Enabled Directory standards.

In 2005, version 6 of View500 was released, making it the world's first XML Enabled Directory. The vision behind this release was to continue to provide a general purpose directory server that was capable of accommodating both past, present and emerging technology trends.

This document will discuss the XML Enabled Directory and the benefits that it has brought to View500 users.

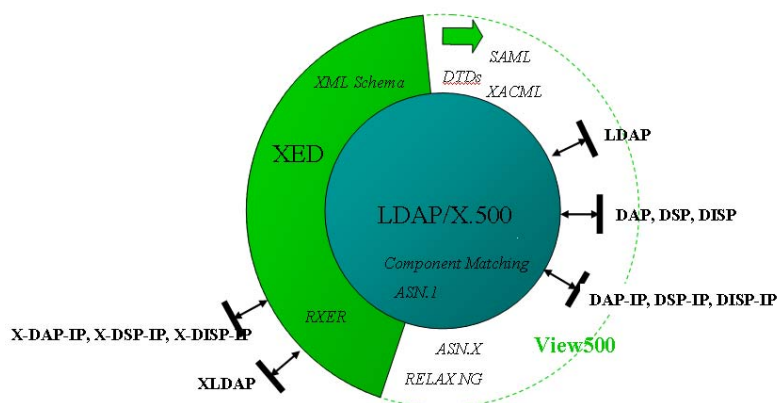
XML Enabled Directory

It was a logical step in the advancement of directory technology to adapt and become XML enabled due to the emergence of XML based technologies. To prevent application developers repeating the mistakes of the past, a general purpose, XML enabled, distributed directory service was required.

In 2003, after a significant amount of joint research between the View500 architects and Deakin University Melbourne, Australia, the initial revision of the XML Enabled Directory specifications were published through the Internet Engineering Task Force (IETF). The XML Enabled Directory (abbreviated to XED and pronounced 'zed') initiative was pioneered by Dr Steven Legg, who remains the View500 Product Architect at eB2Bcom today. Andrew Sciberras, a member of the View500 team at eB2Bcom, is also an author of a XED specification.

Some directory vendors have XML enabled themselves by introducing support for the Directory Service Markup Language. Where DSMLv2 provides an XML based protocol (that doesn't seem to accommodate for XML data), XED provides a complete specification for enabling directories to facilitate XML based technologies.

The XML Enabled Directory (XED) leverages existing Lightweight Directory Access Protocol (LDAP) and X.500 directory technology to create a directory service that stores, manages, and transmits Extensible Markup Language (XML) format data, while maintaining interoperability with LDAP clients and X.500 agents. The following diagram shows how XED is built on top of View500's core LDAP/X.500 capabilities. Whilst still offering the core capabilities through the LDAP/X.500 protocols, a new XML based offering is also provided, leveraging the same powerful capabilities.



The main features of XED are:

- semantically equivalent XML renditions of existing directory protocols,
- XML renditions of directory data,
- the ability to accept at run time, user-defined attribute syntaxes specified in a variety of XML schema languages,
- the ability to perform filter matching on the parts of XML-format directory attribute values,
- the flexibility for implementers to develop XED clients using only their favoured XML schema language.

The remainder of this document will discuss these features of XED and how they apply to and enhance View500.

XML Based Directory Protocols

An important component of the XED specification is the ability to provide semantically equivalent XML renditions of existing directory protocols. The existing directory protocols offered through the LDAP and X.500 standard required information to be transferred in a binary format called the Basic Encoding Rules (BER). Whilst both LDAP and X.500 offer directory protocol services over TCP/IP, XML based services are offered over HTTP and SOAP based connections.

In order for the directory to be accessible by XML clients it needed to be able to offer XML centric communication methods. This feature of XED provides XML based versions of the X.500 protocols.

XML Internet Directly Mapped (IDM)

The Internet Directly Mapped (IDM) protocol was included in the 4th Edition of X.500 whereby X.500 protocol operations can be exchanged between directory agents using TCP/IP with minimal encapsulation.

Protocol operations in the IDM protocol are encoded according to the Basic Encoding Rules (BER) of ASN.1. The XED framework introduces a new, exclusively XML-based protocol, called the XML Internet Directly Mapped (XIDM) protocol, which differs from the IDM protocol only in that the protocol operations are encoded using RXER instead of BER.

XML Lightweight Directory Access Protocol (LDAP)

Whilst the IDM protocol allowed a simple substitution of the encoding rules to create a uniformly XML-formatted protocol operation, LDAP does not. A straight application of RXER to an LDAP operation would inconveniently force directory attribute values, among other things, to be represented as hexadecimal strings.

DSMLv2 allows an LDAP directory to be accessed through an XML based protocol. However, it suffers the limitation that, should the directory store XML information, the XML data will need to be transported in a hexadecimal format.

Considering that the intention of an XML based access protocol is to facilitate XML based environments, the inability for technologies such as DSMLv2 to handle XML information is a severe limitation. It is functionally deficient because it adds additional encoding and decoding overheads to both the client and the server. It also leads to security complications, in that any XML appliance making stateful inspections of the directory data will not be able to interrogate the hexadecimal information as readily as it can inspect the native XML information.

Within the XED framework the LDAP ASN.1 specification is transformed to create a new specification without the discontinuities. By converting the bland OCTET STRING containers used in LDAP into open types and specific types used by X.500, the directory data can be represented in a meaningful format.

The XML Lightweight Directory Access Protocol (XLDAP) is the result of applying RXER to instances of the message types of the transformed specification. Apart from the change in syntax, XLDAP is semantically equivalent to LDAP.

Since the XED protocols are algorithmically generated from the LDAP and X.500 specifications, all future extensions to LDAP and X.500 automatically acquire a XED protocol representation.

XML Rendition of Directory Data

X.500 and LDAP directories hold data that is defined by Abstract Syntax Notation One (ASN.1) and transported using BER (a binary representation). In X.500 the data is stored and provided to the client application in its complex format.

LDAP attempted to make the data more readily usable by clients by defining simple string representations for different types of information. Whilst this was somewhat effective it couldn't cover all data types and was specified on a case by case basis.

The Generic String Encoding Rules⁶ (GSER), which was standardised by Dr. Steven Legg in 2003, provided an algorithmic approach that allowed a string representation to be created for arbitrary data types.

In an XML centric environment, directory servers need to be able to provide clients with XML encoded information. Learning from past mistakes, the method in which the information is encoded needs to be algorithmic and robust enough to handle anything the directory may store.

XED defines an algorithmic approach (like GSER) called the Robust XML Encoding Rules (RXER) which allows the directory content to be represented in an XML format.

Whilst DSMLv1, allows schema, entries and attribute types to be represented in an XML format, it does not provide any means for the attribute values to be represented in XML. DSMLv1 relies on the fact that LDAP servers will simplistically use simple text strings and that anything else (including XML data) is to be treated as a binary BLOB.

RXER provides algorithmic rules that allow data structures defined in ASN.1, XML Schema, Relax NG and DTD to be represented in an XML format. This allows protocols that are represented in BER, to be encoded in XML (e.g. XIDM and XLDAP).

Thus RXER allows View500 to store, search, retrieve and represent data in an XML format, regardless of its syntax.

User Defined Syntaxes

As mentioned previously, the data held within X.500 and LDAP directories is defined by ASN.1, in other words, the range of valid syntaxes of data are all defined by ASN.1.

General purpose directories have knowledge of hundreds of different ASN.1 types and in doing so allow a large variety of information types to be stored and used. Other directories, for instance, will not understand what a certificate is and will simply store it as a meaningless binary object.

Regardless of how many syntaxes a directory supports, LDAP and X.500 do not provide a means to add additional definitions. Therefore, directories can only offer a finite set of built in syntaxes, which cannot be extended.

The XED framework defines additional schema operational attributes in order to hold XML Schema, RELAX NG documents, DTDs and ASN.X definitions. These definitions, supplied by the user, will be instantly understood by View500 and subsequently available for use as the syntaxes of new attributes, making View500 the only directory available with an unlimited extensible schema.

XML Component Matching

One of the greatest strengths of a Directory Server is its ability to search information. Having the ability to store XML information without the capability to search it would not be very beneficial. The XED specification also defines Component Matching rules that allow the inner content of an XML syntax to be targeted by search requests.

⁶ GSER - RFC3641 - <http://www.ietf.org/rfc/rfc3641.txt>

The following is an example of XML data that could represent a person's telephone contact information:

```
<telephoneContact>
  <name> ANDREW SCIBERRAS </name>
  <home> 03 9596 6367 </home>
  <mobile> 0412 098 771 </mobile>
</telephoneContact>
```

If this information was to be stored in a conventional directory server, it would be stored as a string. As a string, any text based value will be permitted, without any verification of whether the value conforms to the XML Schema definition. . In View500 the value would be stored as structured XML data ensuring quality and conformity to the XML syntax.

A conventional directory server can only allow simple string based searching to occur on the XML value, such as "Start With", "Ends With", "Contains" and "Equals". Whilst these searching capabilities will allow the XML to be searched, it is not possible to target specific elements or attributes of the XML structure, such as the `home`, `name` or `mobile` elements of the `telephoneContact` type. With View500, search operations can utilize component matching capabilities that allow the desired elements to be precisely targeted and matched appropriately. For example, View500 can process a request looking for XML `telephoneContact`'s that contain an element called **name** with a value that sounds like "**androo**".

View500 can be configured to index specific portions of an XML syntax to ensure the utmost efficiency when handling component level searches.

This allows View500 to provide all of its capabilities and functionality as a high speed Directory Server for any application that needs to store, search and obtain XML content.

Applicability and Usage Examples

Unified Policy and Identity Store

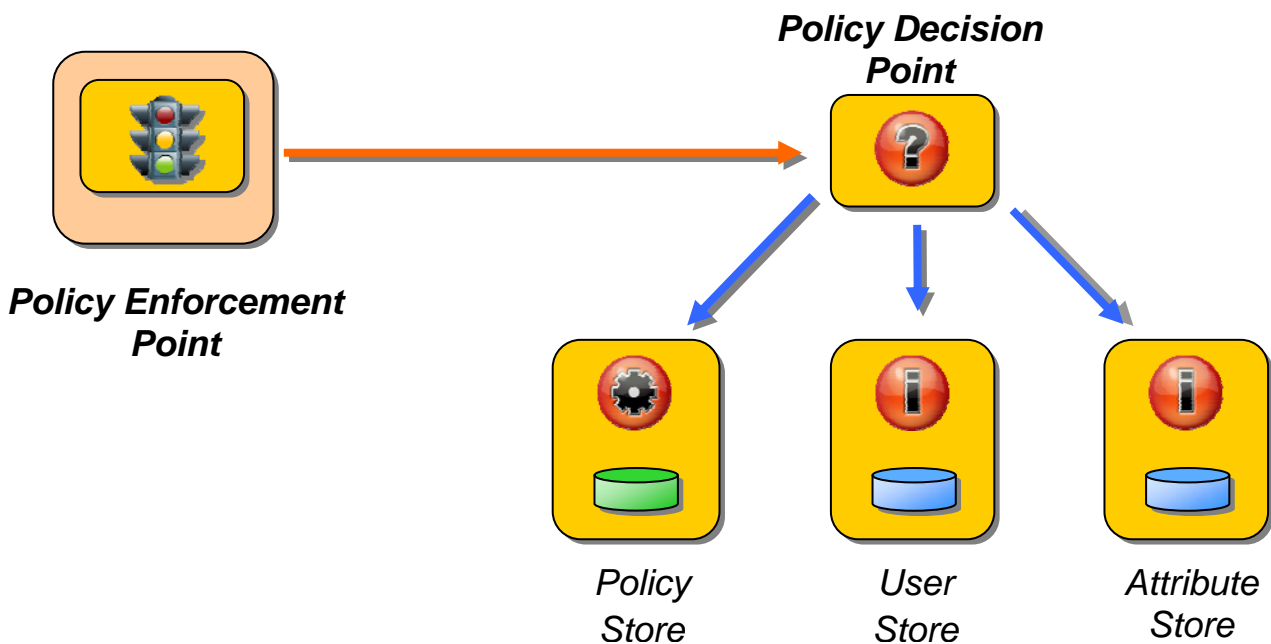
Most organizations will, in one form or another, have an Identity Store. Typically, the Identity Store is a standardised implementation of LDAP that is used to identify users and their roles. Sometimes, this may be a single directory server in centralized environments, whilst in others it may be a combination of a User Store for authentication, plus an Attribute Store holding additional information about the identities.

Directory server products typically used as an Identity Store include Microsoft's Active Directory, SUN's Java System Directory Server, Novell's eDirectory, the Red Hat Directory Server, the Fedora Directory Server, the Apache Directory Server, IBM's Tivoli Directory Server. Considering that the Identity Store is generally used to store information about users for authentication and lower level access control decisions, it makes sense that most of these products are made available by operating system vendors.

Whilst these technologies can suitably store directory user information and proprietary policies that are used to govern access to the resources that the operating system protects, they are less suitable for the storage of generic modern day fine grained access control policies. In some cases, these technologies are unsuitably rigid and consequently not suitable to be used to hold additional attributes of a user. Hence the need for a separate Attribute Store.

XACML (eXtensible Access Control Markup Language) is an access control policy language based on XML. Any organization wishing to use XACML and one of the typical Identity Store LDAP Directories, will need to deploy yet another information repository. Typical Identity Stores cannot adequately store the XML policy data, because they do not possess the capability required to validate, search or comprehend the content of the XACML policy beyond it being a nonsensical string.

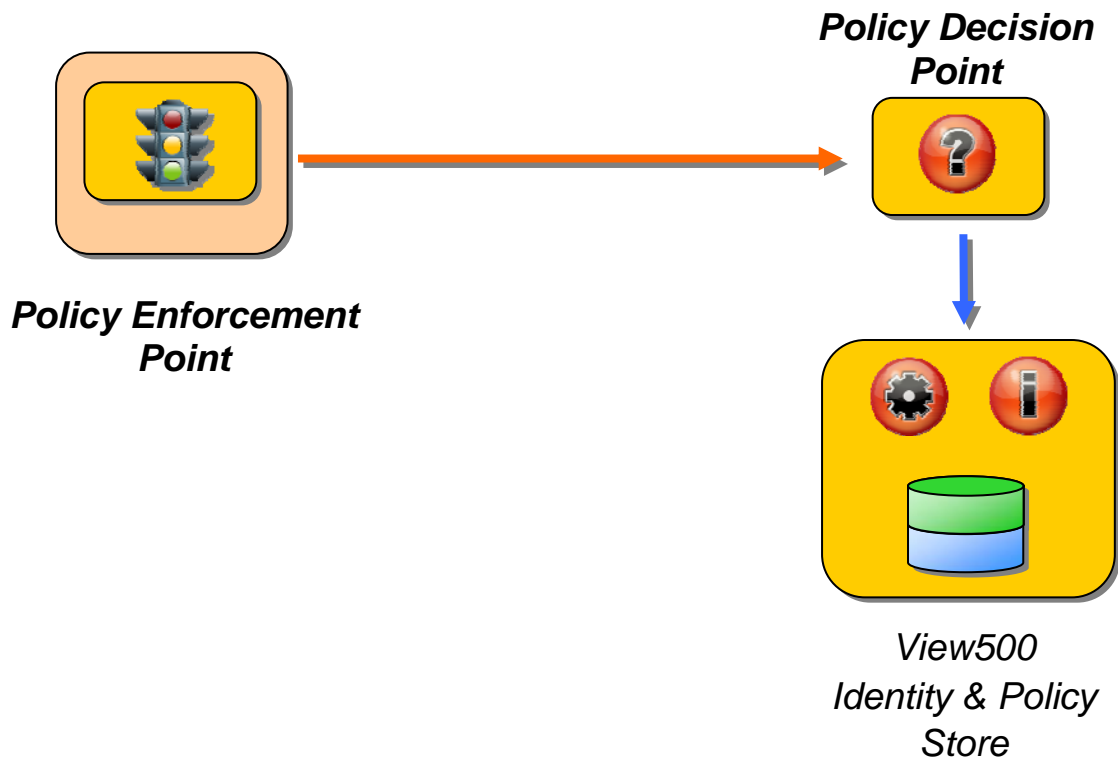
An environment enforcing XACML policies within a **non-View500** environment may look like this:



In some cases the LDAP directory used may be suitable enough to store user authentication and additional attribute information, in which case the User and Attribute store are merged into a single Identity Store.

However, since typical LDAP directories cannot suitably store the policy data, another repository will be introduced. This increases the complexity of the Policy Decision Point, introduces more points of failure, increases latency in response times and reduces the manageability and audit ability of access control policy information within an organization.

Using a View500 XML Enabled Directory in an environment enforcing XACML policies would look like this:



View500's support for the storage, search and retrieval of XML information such as XACML allows it to be used in the above example as both the Identity and Policy store, providing a centralised management of both information and policy.

Also since View500 supports both RBAC (role Based Access Control) and ABAC (Attribute Access Control (and combinations)) tight authenticated access to those security policies stored in View500 can be maintained

The XML Enabled Directory capabilities of View500 allow the XACML policies to be searched quickly and efficiently, offering Policy Decision Points with high speed service.

Through the use of View500's replication capabilities, the Identity and Policy information can be replicated either fully or partially to other View500 servers. By replicating a complete copy of the data, multiple View500 Identity & Policy stores can be employed for high availability, load balancing and/or failover reasons.

ebXML Registry Service

Electronic Business using eXtensible Markup Language (ebXML) is an XML based technology that enables the interoperable, secure and consistent usage of electronic business information by all trading partners.

One component of the ebXML technology is the ebXML Registry Service, which holds electronic business information about traders and makes it accessible. The ebXML Registry Service needs to be accessed by traders to publish information. Consumers will access the registry service to find information about traders.

The requirements of the ebXML registry service are synonymous with the requirements of a typical directory service, whereby information is published infrequently and used in query and discovery operations. The importance of the service's ability to provide accurate and efficient searching is of the utmost importance in such systems. Whilst LDAP and X.500 Directory Servers have typically been based on binary information, the ebXML Registry Service is based on XML technology.

This disjoint between binary directory services and XML registry services has caused some vendors to reinvent the wheel or to use inferior relational database technology for an ebXML Registry Service. eB2Bcom can provide an efficient ebXML Registry Service through the underlying capabilities and features of View500.

Providing support for ebXML within View500 was a simple process that built upon solid foundations. The core competency of the product allows high volumes of information to be stored and organized within View500.

The Directory Service competencies allow View500 to index the ebXML electronic business information and provide sophisticated search facilities.

The XML Enabled Directory framework allows View500 to provide support for the XML protocols used to interrogate and populate electronic business information. The information stored within the registry is also based on XML, which View500 can readily store and search.

Conclusion

By definition, a directory server is a hierarchical database that is optimised for query and read based operations based on reasonably static data. In order for a directory to achieve these basic goals, it needs to be accessible, accurate and efficient.

In order for an XML Enabled Directory to function, it too must be able to maintain XML data within the hierarchical database and provide an optimised search capability.

An XML Enabled Directory only becomes useful if it supports an access protocol that is XML based and easily supportable by client applications. The manner in which the protocol elements are encoded using XML should be based on an algorithmic approach.

LDAP and XLDAP are extensible and allow the protocol to be extended in a standardised and compliant manner. RXER is an algorithmic XML encoding rule, which means that even if the LDAP/XLDAP protocol is changed or extended, RXER enabled clients will be able to automatically adapt to the changes. A DSMLv2 enabled client will however have to wait for the DSMLv2 standard to get updated.

The XML communication protocol also needs to handle XML data natively. A protocol such as XLDAP seamlessly allows XML values to be handled and transferred in their native XML format. DSML on the other hand, despite being an XML based technology, cannot handle XML directory values and requires them to be BASE64 encoded.

An XML Enabled Directory that is used for the storage of XML information should ensure the accuracy of stored data and be able to guarantee the accuracy of search results. Only directory servers that support XED's ability to define XML based syntaxes will ensure that an XML value conforms to its schema definition.

Without the ability to dynamically add XML Schema definitions into the directory and allow them to be used as attribute syntaxes, there is nothing preventing non conformant and poor quality data contaminating the directory.

Being able to obtain accurate search results from XML data is of vital importance, otherwise there is no point in storing XML in a directory in the first place. The ability to control precisely which aspect of the XML is being searched and with what flavour of matching rule is only achievable by directory servers through Component Matching technology.

When storing XML data and allowing clients to search it, the operation must be fast and efficient. Using a directory that can index the inner components of a complex XML structure will guarantee its ability to deliver accurate results efficiently.

As it can be seen from the two business examples, View500's ability to store Attribute data and XML data in the same directory server provides a unique ability in the market today. Organisations considering the use of any XML based information such as the GJXDM⁷ (Global Justice XML Data), or Military XML-Message Text Format⁸ (XML-MTF) where they both have a need to combine the efficient capabilities of a Directory service and the ability to store, search, understand and retrieve XML data should be considering the use of View500.

Today the View500 XML Enabled Directory Server offers an **accessible, accurate** and **efficient** service. View500 supports the XED technology including RXER, XLDAP, XML Syntaxes and Component Matching. As such, View500 is currently the only directory server available that can truly claim to be **fully XML Enabled**.

⁷ GJXDM - <http://www.it.ojp.gov/jxdm/>

⁸ XML-MTF - <http://faculty.ed.umuc.edu/~meinkej/inss690/moffatt/moffatt/military.htm>