

**Greek School
Network**

**National Technical University of
Athens**



Moving LDAP Writes to Web Services

Kostas Kalevras

National Technical University of Athens, Network Operations Center

kkalev@noc.ntua.gr

1st LDAP Conference 2007, Köln Germany 6-7 September 2007



Agenda

- **Greek School Network – E-School Development Environment**
- **Problems with direct LDAP writes**
- **Why move to Web Services**
- **LDAP Reads – Authentication**
- **LDAP User Management Service PHP API**
- **Conclusion**



Greek School Network

- Interconnects all Greek schools and provides Internet access
- Provides school and personal accounts
- Email, Dialup, VoIP, web pages services



LDAP Service

- Based on Sun One Directory Server
- Central authentication repository for all user services
- Contains the Organizational Hierarchy
- 170,000 entries
- School accounts, teacher accounts
- Student accounts scheduled



User Administration

- Central Web-based interface
- Written in PHP and Javascript
- Provides an object and form editor/creator
- One form is created per object type (object types are abstract types like student, teacher, adsl router, etc)
- LDAP tree browser and data manipulation (add, edit) forms are provided to administrators
- Delegated administration of entries



Interface features

- **Computed attributes based on other attribute values**
- **Computation formula: Any valid PHP expression or even function**
- **Attribute uniqueness**
- **Referential integrity**
- **Post operations (moving user home directories, welcome emails, etc)**



E-School framework

- **Services on top of the current network**
- **Provided services:**
 - **Web portal (sPortal) for student parents**
 - *Parents register and can check out their child's progress and status*
 - **PKI infrastructure**
 - **School Administration platform**
 - *Move all school operations to the electronic world (student enrollment, classroom management, grading)*
 - *Central personnel and student database*
 - *Interface (.NET) running on all schools communicates changes with the central database*



New entry sources

- **Old days: Accounts were created through the central web interface**
- **E-School: Accounts are created from more than one sources now:**
 - **sPortal creates parent accounts**
 - **School Administration platform creates teacher, student accounts and maintains the organizational hierarchy**
 - **School accounts (official school email account) still need to be created 'by hand'**



Why Direct LDAP access is bad

- Each service only knows it's own little world (and attributes). sPortal for instance only needs a username/password pair and nothing more
- No easy way to perform post-operation tasks
- Apart from ACIs there's no other control over what is written (no real constraints)
- Changes to the entry schema need to be integrated in ALL outside sources
- No way to expire an entry instead of deleting it
- Services code and operation are outside our administration domain



Web Services to the rescue

- Create web service functional interface around the user interface
- Provide functions accessible through HTTP(s)-SOAP (declarations in WSDL)
- Web services written in PHP nuSoap
- Map all abstract operations (i.e. Parent Creation) to functions in the web services
- User interface provides general object interaction functions in PHP (ldap add/modify/delete)
- All complex features are already present and configured in the user interface



Example

- **createParent()**
- **Input: Parent name, surname, username, password**
- **Check arguments, username uniqueness**
- **Log all operations**
- **Call internal object creation routine**
- **Routine handles all complex operations (like computed attributes, etc)**
- **Output: Status Code, Error Message if present**



Advantages

- One function backend for both the e-school services and the user interface
- Complete logging is available. No more looking through million lines of directory server logs
- Computed attributes are available
- Pre and Post operation tasks can be performed (calling outside scripts/web services)
- All operations pass through a central point. We can set any constraints on the provided values



Advantages (2)

- Outside service need not know our schema. They call a function with the minimum set of arguments. We can change the entry schema whenever we want
- We can have our own expiration policy. EntryDelete() could just set *active=false*
- WSDL is clear and precise. LDAP is abstract and parties need to agree on how to perform operations.



LDAP Reads

- **Web services could be used for complex reads too**
- **One function for every complex search operation**
- **Group Membership, LDAP browsing are perfect candidates**
- **Advantage: Schema abstraction, functional interface**
- **DSML could be used to carry back entry information**



Authentication

- **HTTP authentication is used**
- **Credentials are mapped to LDAP entries**
- **Web Service binds with the HTTP credentials**
- **Which credentials to use?**
 - **Special service user in case of synchronization mechanisms**
 - **User entry for which the operation is requested (i.e. change password operation)**

LDAP User Management Service (LUMS)

- A PHP LDAP Entry Management API has been created for another project
- Provides:
 - A set of basic LDAP API functions (search, add, delete, modify, rename, change password)
 - A strong configuration language
- Administrator defines ldap object types and their corresponding attributes



LDAP User Management Service (2)

- **Options available for each attribute**
 - Define as required, multivalued
 - Set attribute type (string, binary, dn, telephone, email, etc)
 - Define attribute value source: User inserted, constant, auto increment, function created
 - Allow for attribute uniqueness
 - Define extra syntax checking function
 - Define virtual attributes which can be used to create attribute mappings
- **Pre and Post operation functions can be defined**
- **Automatic handling of non English charsets**



LDAP and XML integration

- DSML has been available for quite some time and is starting to get used
- XML Enabled Directory envision moving the entire LDAP protocol to XML space
- Looks like LDAP and XML integration will be even tighter in the near future



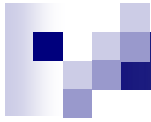
Conclusion

- **A web service functional interface can provide significant benefits if:**
 - **There are more than one entry sources**
 - **Sources are heterogeneous and possibly multiplatform**
 - **Sources are usually outside out administration domain and control**
 - **Information synchronization is not based on human interaction**
 - **A strong and configurable LDAP API is provided for use by the Web Service**



References

- **Greek School Network:** <http://www.sch.gr/>
- **NTUA NOC:** <http://www.noc.ntua.gr/>
- **LUMS:** <http://www.sourceforge.net/projects/lums>
- **Blog:** <http://kkalev.wordpress.com/>



Thank you!