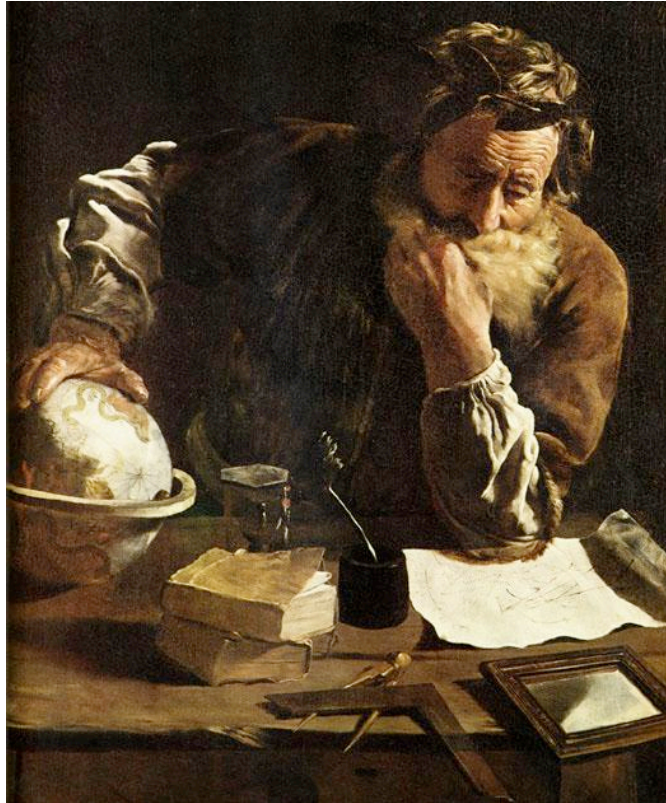


## OVERVIEW OF THE OLE TECHNICAL ARCHITECTURE



ARCHITECTURAL MODEL FOR THE OLE TOOLKITS

OPEN FEDERATION OF OER SYSTEMS

Charlie Kohl

February 2008

# Overview

## Goals for the OLE Technology

The OLE technical architecture defines the framework for the tools that support the overall OLE mission of Universal Basic Education. As a framework, the architecture defines the mechanisms that enable the OLE community to easily access a wide range of existing open source educational technology. The creation of the OLE toolkits is not a monolithic effort to build new tools. Rather, it is a minimalist effort to provide the pathways that connect the tools that already exist in a fashion that makes them accessible to a global community.

Specifically, the framework supports the OLE consortium by:

- Enabling worldwide access to a broad range of digital educational resources through the use of varied delivery technologies, hardware platforms, and access protocols.
- Supporting online, near-line, and offline usage models.
- Providing transparent interfaces to existing OER systems.
- Supporting educators in the collaborative development of educational resources and the reuse of existing resources.
- Supporting distributed learning communities in the use of OLE learning resources in focused educational pursuits.
- Enabling OLE Centers to quickly establish local sites to manage localized views of the global library of educational resources; provide email, website, blog, and forum communications for OLE members; and analyze usage of the OLE resources.
- Providing out-of-the-box tools that enable OLE Centers to manage membership, training, and fundraising.

## Design Principles

The first key design principle is that the OLE Toolkits leverage existing open source technology to the maximum extent possible. Our goal is to build as little as possible. We have not designed a new technology platform for the creation and distribution of open educational resources. The OER community has already developed many excellent tools - there is no need to rebuild this technology to support the OLE mission. Rather, it is our goal to build the *glue*: a framework for collaboration that facilitates the OLE educational mission rather than defining it.



The second design principle for OLE technology is that the OLE software be open and accessible. To achieve the ambitious goals of this effort, technology development must be a collaborative, global effort. We must be able to draw on technology components from many technical communities: networking, database, web application servers, content management systems, multimedia, OER, content delivery, visualization, etc. The integration of such a diverse set of tools would be effectively impossible without access to source code and freedom from proprietary licensing restrictions.

The OLE architecture is *platform gnostic*: we appreciate the power of technology to achieve our mission, but do not want to bind the architecture to any specific platform. The global OLE community will be using a broad range of

server platforms, networking protocols, client architectures, and presentation devices; the architecture must be general enough to be compatible with the broadest possible subset of technologies while still providing the capability to leverage unique features of specific devices and platforms.

We approach the divergent goals of platform generality and device specificity by decoupling system functionality from presentation through the use of the REST (REpresentational State Transfer) and the Model / View / Controller design patterns. With these patterns, we strictly separate **data** from **behavior** from **control and presentation**. Data (educational assets and the metadata describing those assets) are stored as *resources* in distributed *repositories* that can be accessed through a standard syntax independent of the actual repository technology. Behavior is provided by *web services*, distributed stateless processes that are invoked through a standard set of calls over the web to provide specific functionality. These web services access the educational data using the standard repository syntax, so the services themselves do not necessarily know (or care) where or how the data is stored.

Users interact with web services through client applications that invoke series of web services to produce desired functionality. Since all interaction between the client applications and the services is managed through a specific, well-defined set of service messages, the web services themselves require no knowledge of the client application that is requesting the behavior. A request for a specific educational resource from a cell phone is identical to a request for the same resource from a print server or from a three dimensional holographic display. Different user workflows (e.g., finding and downloading a slideshow vs. taking an interactive online course) can be supported by the creation of multiple client applications that use the same set of web services in different ways.

	
Asynchronous Delivery Device	Synchronous Delivery Device

### OLE Toolkits

OLE has defined three toolkits that will be created as deployable packages making use of the services provided in the OLE technical architecture. These are the OLE Knowledge Toolkit, the OLE Community Toolkit, and the OLE Sustainability Toolkit.

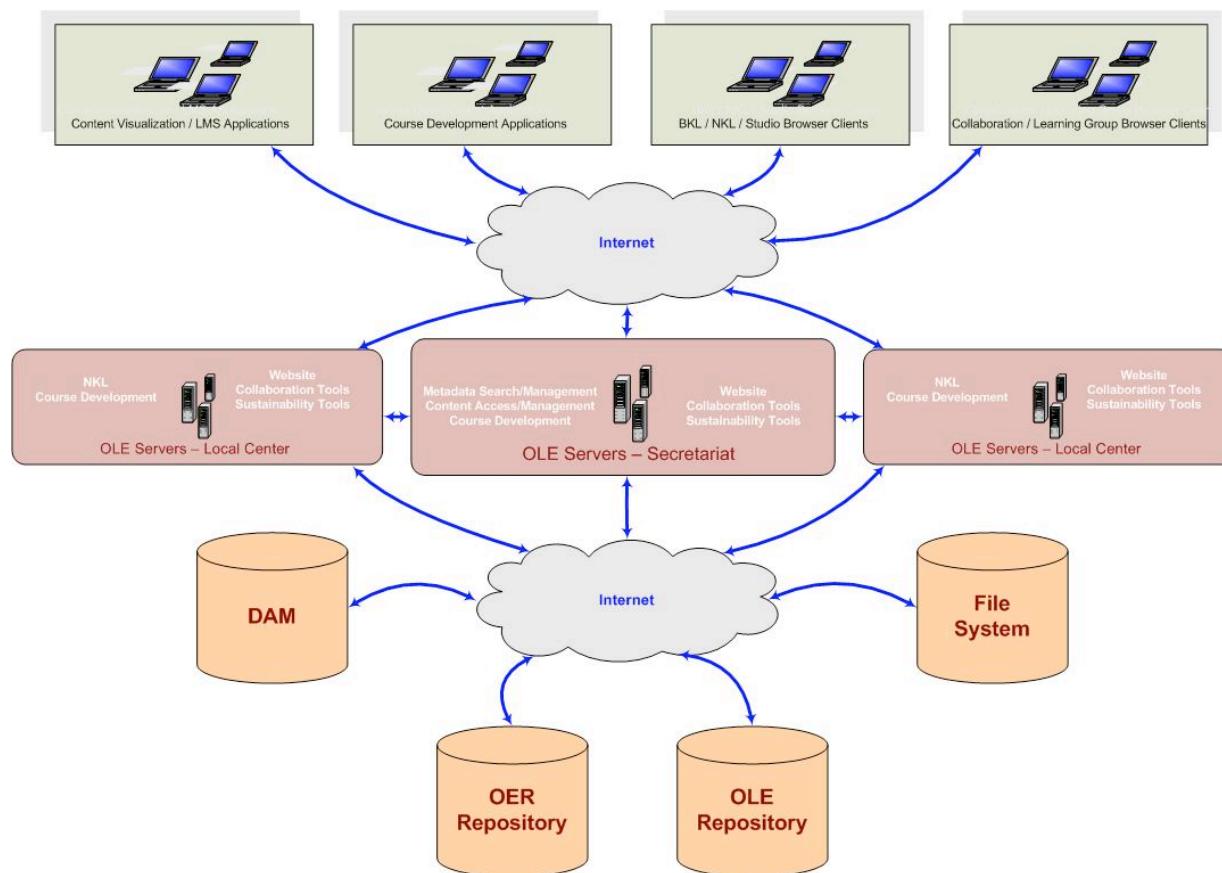
- The **OLE Knowledge Toolkit** provides support for the development, storage, and use of the educational resources available through the Billion Kids Library (BKL - an open global repository of all educational resources available to the OLE community) and the National Kids Libraries (NKLs - filtered views of the BKL providing access to resources appropriate to a particular OLE organization). This toolkit provides mechanisms for: finding resources; creating resources; delivering and using the resources; evaluating resources; analyzing the usage of resources; and localizing, versioning and archiving those resources.

- The **OLE Community Toolkit** provides support for “learning communities” utilizing the OLE tools for individual or group educational activities. The toolkit includes: documentation of best practices; teacher training materials; discovery tools for finding other members or groups; and membership management, group management, and collaboration tools to support online group collaboration. These collaboration tools are all open source third-party modules supporting shared calendars, RSS feeds, text messaging, voice communication, blogs, forums, shared documents, and shared whiteboards or desktops.

- The **OLE Sustainability Toolkit** packages a set of tools supporting the creation, growth, and management of OLE Centers throughout the world. The services provided through this toolkit can either be run on the servers of the OLE Secretariat or on local servers managed by an individual OLE Center. The toolkit provides a generic “out-of-the-box” website, communication and collaboration tools, fundraising tools, analytic tools, and best practices documentation for the management of an OLE Center. As with the Community Toolkit, the modules provided in this toolkit are largely existing packages developed by the open source community.

### System Architecture

The technical architecture that is currently being designed and developed by the OLE consortium can be summarized by the following diagram.



At the lowest level of this architecture, we see the federated, web-based systems for the management and storage of educational resources and associated metadata. content and metadata storage. The XML Schemas and web services

of the OLE framework define a canonical mechanism for resource management that supports the federation of a wide range of resource repositories including: Digital Asset Management systems, OER repositories such as Fedora or Moodle), OER projects such as MIT OCW, COSL, Connexions, or eduCommons, databases, or even file systems. We are currently implementing a Fedora repository for new OLE content, but envision a number of technologies for OLE managed repositories.

The middle layer of the architecture provides the web services that support users in the creation and use of the federated educational repositories for the Knowledge Toolkit. Service requests for content made to the OLE servers are completely isolated from the content repositories, thereby providing the generality and extensibility that enables the use of federated resources. The server layer is distributed and mirrored, providing access either through the servers of the OLE Secretariat or through servers managed by local OLE Centers.

The middle layer also uses the web servers and web application servers to run the modules of the Sustainability and Community Toolkits. The current implementation utilizes Tomcat and JRun, but the underlying architecture is not tied to these platforms. These modules of the Sustainability Toolkit utilize local MySQL databases to maintain RSS feeds, shared documents, CRM data, and analytic data specific to the local Center. The Community Toolkit utilizes mirrored MySQL databases to maintain member data and learning group data that must be accessible (subject to appropriate authorization) across the OLE community in order to support global collaboration.

The top layer of the architecture supports thin (browser-based) clients and thick client applications (standalone applications - e.g., LMS) that enable members to access and manage the modules of the OLE toolkits. The thin client applications utilize HTML, AJAX, and Flash in standard browsers to provide interaction with application processes running on the servers. Standard server-side localization technologies and the use of cascading style sheets enable local OLE Centers to easily customize the thin client applications to their individual needs.

Thick clients are applications running on the local laptop, and use web services to access and manage resources in the repositories and OLE databases. Because all communication between these applications and the servers is defined by the explicit set of web services, developers throughout the OLE community are free to create OLE applications using whatever client technology they choose. Thick client applications also provide the client side mechanisms to support asynchronous workflows where the member is creating or using educational resources offline.

## Summary

The core of the OLE technical architecture is a minimalist set of web services designed to support generic access to existing educational resources and to support collaborative development of new resources appropriate to the mission of Universal Basic Education. These services utilize the *facade* design pattern to provide a single access protocol that hides the complexities of individual repositories. A wide variety of client applications, content delivery, and learning management tools can then be constructed to deliver the educational resources in a fashion appropriate for specific local educational structures, languages, cultures, and technology infrastructures.

This core architecture is the foundation for a potentially broad range of client applications that will be created by the OLE community to provide specific solutions tailored to local needs. The three independent layers of the architecture enable us to extend the functionality of the OLE Toolkits and client applications as new requirements are identified. The architecture is designed to be lightweight, fluid, and flexible; as the community evolves, the infrastructure will be able to evolve in parallel to meet the emerging needs of Universal Basic Education.