



Open Content Architecture:  
A Pluggable Platform to Accelerate and  
Simplify Content Processing and Delivery

## Table of Contents

- 1 | The Content Challenge
- 2 | The Role of Enterprise Content Management
- 2 | Open Content Architecture
- 3 | Mark Logic in the Open Content Architecture
- 3 | Interoperable Components in the Open Content Architecture
- 4 | System Integrators and the Open Content Architecture
- 4 | The Mark Logic Partner Network
- 5 | Conclusion



# Open Content Architecture: A Pluggable Platform to Accelerate and Simplify Content Processing and Delivery

## The Content Challenge

In the early 1970s, computer scientists introduced a set of technical innovations to improve the storage and querying of numeric data. By the end of that decade, corporations adopted these methods – packaged as relational database management systems – as the standard for information management. Relational databases have improved dramatically in functionality and performance over the past three decades. However, these advancements have not addressed a critical business reality: most corporate information does not fit neatly into relational tables. In fact, over eighty percent of corporate information is content: reports, contracts, proposals, e-mail, employment records, compliance documents, training manuals, and the like. These documents are produced in varying formats – including Microsoft Word, PowerPoint, and Adobe PDF – and cover a wide range of topics.

Managing content using a relational database presents a number of unique challenges. The primary issue is that of structure. Relational database engines and tools assume a highly consistent data structure. For instance, every address in the United States has a street number, city, state, and ZIP code; because the data format is always consistent, address information fits neatly into a relational table. By contrast, corporate documents, even of the same type, are often strikingly heterogeneous.

For example, an installation guide is a short, simple document, often just a couple of pages long. By comparison, a complex training manual may have dozens of chapters with sections and sub-sections, a number of explanatory appendices and hundreds of diagrams and graphs. Both of these documents could be classified as “instructional materials,” yet their internal structures are very different. This heterogeneity makes a relational database a poor choice for content storage.

A second challenge when dealing with content is that its component pieces – sentences, phrases, paragraphs, and so on – do not exist only as discrete elements; hierarchy and context play an important role. In a relational database, a single record has a specific meaning. That John Smith purchased three cordless phones on September 3rd, 2004 is a distinct fact; its meaning does not change with circumstance. However, the fifty-third sentence of the service contract between Hi-Band Telecom and U.S. Enterprises can only be understood in the context of its surrounding content, such as clarifying sentences or definitions of contract terms. When dealing with content, the ability to search through text while maintaining context is critical.

Finally, the purpose for which content is created may differ entirely from the ways it is subsequently used. For example, a vendor

and client may sign a support agreement to ensure the timely delivery of service in the event of a product failure. However, years after the initial signatures, the vendor's sales force may want to use the contract as the basis for its renewal efforts. Meanwhile, when planning a revision to the terms of the support program, the marketing department may browse through those agreements terminated by customers. The needs of later content users are quite distinct from those envisioned when the content – the contract – was originally created.

This quality of varied reuse is further reason not to use a relational database as a content repository, for relational databases cannot store information for which they are not preconfigured. In general, database architects attempt to envision all potential uses for the data at design time, to avoid potentially costly configuration changes after system deployment. But any system designed to leverage the value of content must be flexible enough to allow documents to be reused in ways not envisioned by their original creator.

### **The Role of Enterprise Content Management**

In the late 1990s enterprise content management (ECM) applications emerged to address the special requirements of corporate content. To introduce some control over the creation and deployment of enterprise documents, ECM focused heavily on managing the workflow of information origination and approval.

The ECM business grew rapidly. However, ECM adopters quickly discovered that unlike relational databases, which are open platforms on which other technologies and applications can be layered, ECM systems are relatively closed environments, difficult to enhance with third-party software. These content management applications usually offer a single, proprietary approach to handling documents, and require end users to work with their built-in content store, analysis algorithms, and workflow tools.

As a result of this closed architecture, customers who need a range of content-

handling functionality may end up with multiple systems based on incompatible content silos, even when the content sets largely overlap. The increasing number of discrete content stores introduces the need to segment, replicate and migrate content, and to maintain the consistency of related content sets. Enterprises clearly need a more efficient and flexible model.

### **Open Content Architecture**

Mark Logic advocates the Open Content Architecture (OCA) as an optimal approach to managing and leveraging unstructured and semi-structured corporate information. OCA describes a model of how systems that store, process, enrich, and deliver content should interoperate.

The Open Content Architecture model centers on a standards-based persistent content store (or "repository") with an extensible architecture. The content store and best-of-breed solutions for processing, enriching and delivering content may be quickly assembled by an IT group or systems integrator into a fullfeatured content application.

The OCA content repository must have the characteristics of a high-end database: it must be able to deliver the performance, scalability and programmability for content that relational databases deliver for numeric data. In addition, to address the unique challenges presented by content, a content repository must support element-level query – so it can extract relevant subsections of a document – and full and partial document updating. Finally, since users typically employ search techniques to locate content of interest, a robust content repository must also offer full text search at both the document and element level.

In addition to the repository, any content environment implementing the OCA model must support the three key steps of unstructured data workflow.

#### **1. Getting content into the system**

In contrast to typical practices in the relational world, where transformations must occur before data can be loaded into the database, a content repository must be

Mark Logic advocates the Open Content Architecture (OCA) as an optimal approach to managing and leveraging unstructured and semi-structured corporate information. OCA describes a model of how systems that store, process, enrich, and deliver content should interoperate.

flexible enough to allow customers to load content without worrying up front about issues such as format conversion and structure normalization. Instead, users must be able to perform content transformations iteratively, as new requirements arise or new processing algorithms become available.

## 2. Enriching the content in the repository

Simple content applications can often be constructed on top of raw content that has been quickly loaded into the database without additional processing. That said, most such applications do benefit from “content enrichment,” the post-processing of content to add value in the form of additional metadata. A common example of content enrichment is “entity extraction,” in which names, dates, locations and other important concepts found in documents are identified and flagged.

Documents in the repository may be enriched in place, which requires the data store to offer update functionality. Alternatively, documents may be routed to external processing routines for enrichment, and then reloaded into the repository. While such external enrichment processes are most often implemented in software, they may include manual steps such as human data validation.

Content can also be iteratively enriched via an application. For example, a user may annotate documents using an application's Web interface; these annotations become part of the metadata, adding value for other users.

## 3. Extracting and presenting content

In this final step in the content workflow, information is extracted from the database and presented to end users. There are many possible extraction schemes, ranging from simple querying of the repository to quantitative analytical processing, but in general, business logic will reside at the extraction layer. For final presentation of the extracted data, raw content is transformed into a view humans can understand.

The Open Content Architecture can be described generally as a content-centric

database with hooks for the seamless integration of third-party software. However, The OCA allows an application designer to build as much sophistication into the content processing logic as needed to achieve their business objectives. For this reason, OCA designs as implemented may vary heavily from project to project. Furthermore, an OCA application may stand alone, or may be integrated into a larger workflow, such as that of an enterprise content management system.

## Mark Logic in the Open Content Architecture

MarkLogic Server is an enterprise-class database optimized to function as the content store for the Open Content Architecture. It is fully transactional, runs in a distributed environment and scales to terabytes of data. In addition, it offers full programmability and content access in XQuery, the query language for XML.

MarkLogic Server is built specifically for content. It is schema independent; any loaded document can be immediately and efficiently queried without normalization. Content can be loaded in any format, and transformations such as conversion and enrichment can be performed at any time. Finally, MarkLogic Server's open-standards interfaces allows for easy integration with a wide range of products for content processing. In addition to supporting XQuery, it also offers APIs in Java and Microsoft .Net.

## Interoperable Components in the Open Content Architecture

The sequence of transformations required to load content into a data store and then publish it via an application is known as the content processing pipeline. Software components that perform specific tasks in the content processing pipeline are critical to the success of an application deployment. Mark Logic has identified a spectrum of such components that offer a wide range of capabilities to OCA project architects. However, the list is not exhaustive, and technical innovation will continue to drive innovations in content processing and management.

MarkLogic Server is built specifically for content. It is schema independent; any loaded document can be immediately and efficiently queried without normalization.

## Document Conversion

Storing content in proprietary binary formats such as Microsoft Word or Adobe PDF traps vital business information inside the document. To unlock the value in these documents, they must be converted and stored in a format that lends itself to querying and enrichment. XML is an open standards format ideal for this purpose; its flexible, hierarchical design incorporates content, structure, and metadata in a single document. Products that can convert content from proprietary binary formats into rich XML play an important role in the content processing pipeline.

Depending on application requirements, the conversion process may range from unnecessary to highly complex. In some instances, “raw” XML, such as that generated by directly saving a Word document as XML, is sufficient. More often, a conversion process that captures the structure and semantics of the document is needed. For example, various third party conversion products can intelligently capture and tag page elements such as chapters, sections, paragraphs, headlines, or sidebars. Enhancing XML content in this fashion can dramatically enhance the business value of content applications.

Voice recognition and optical character recognition may be introduced into the content conversion process if electronic source documents are not available. The output from these components can be converted into XML by a separate conversion step if needed. This “pipelining” – when a series of conversions are sequenced one after another – is an OCA norm and highlights the strength of the architecture.

To help our customers expedite and ease the deployment of content applications, Mark Logic has forged relationships with a range of partners in the above categories, and we are continuously expanding our partner list.

### **Relational Data Integration**

Application designers often wish to present and query content in conjunction with structured data – financial or transactional information, for example – which is typically stored in a relational database. Components that help developers integrate relational data into a content application typically use one of two different approaches: Tabular data can be converted into XML form, loaded into the content repository and treated as a document, or a data integration engine may convert relational data into XML at query time.

### **Content Enrichment**

While there is a broad selection of content enrichment solutions, they typically fall into two major categories:

- Extraction tools automatically identify syntactic or semantic constructs in the content. Examples of syntactic entities include names, companies, products, location and dates. Sophisticated extractors can even identify relationships between entities in a document, performing concept or fact analysis. For instance, while a basic entity extractor could recognize Bank of America and Fleet Bank as corporations mentioned in a document, a more advanced product might further infer that the document refers specifically to the merger between the two banks.
- Clustering or categorization tools automatically associate documents with appropriate categories. These categories may be manually defined, automatically assembled from analysis of a larger body of content, or a mix of the two. Categorization adds vital information to content and enables more sophisticated business applications downstream.

### **Presentation and Visualization**

The final component of the content processing workflow is visualization. Software components are available to help application designers deliver content in a myriad of formats ranging from Web pages to printed media to Microsoft Office documents to cell

phones. Presentation components may also transform content from one XML format to another for integration with external applications.

Visualization tools can help users graphically understand the relationships between documents and the entities they contain. So-called “guided search” systems – which let users browse documents in the content store by category – come into play when simple key word search falls short.

### **System Integrators and the Open Content Architecture**

System integrators and value-added resellers play an important role in the OCA ecosystem. System integrators specializing in content processing have expertise in designing overall content workflows and helping companies determine how to develop and maintain mission critical business processes. They also understand the pros and cons of different analytic approaches – from entity extraction to categorization – and how these approaches can be combined into a complete content application. Integrators can be contracted to build individual components of the architecture, end-to-end systems, or standalone applications.

### **The Mark Logic Partner Network**

To help our customers expedite and ease the deployment of content applications, Mark Logic has forged relationships with a range of partners in the above categories, and we are continuously expanding our partner list. Organizations facing a particular content challenge can choose from a broad range of vendors familiar with the Open Content Architecture and with MarkLogic Server. Mark Logic is also pleased to help enterprises considering the deployment of a content application understand how different technologies fit into the Open Content Architecture model.

## Conclusion

For corporations with content management challenges, Open Content Architecture provides the ideal framework for development efforts:

- It is non-proprietary and standards-based, minimizing IT investment risks.
- Processing steps may be plugged in as needed. Vendor capabilities or proprietary algorithms may

be enhanced by stringing together multiple services. Individual components may be improved or upgraded at will, without requiring an overhaul of the entire environment.

- Multiple independent applications can be deployed on top of a single content repository, allowing a company to leverage its investment in the architecture.
- It is optimized for the rapid deployment of a small “proof-of-concept” application followed by continuous improvement, an approach that ensures maximum project visibility and flexibility.

Prior to the introduction of the Open Content Architecture, organizations have had to solve content problems via other means. Their limited options include file systems, search engines, relational database servers, and enterprise content management systems. But these alternatives each have significant drawbacks:

- Storing content as binary data in a file system or database locks away valuable information. Proprietary binary formats prevent applications from accessing document internals or retrieving only relevant subsections. Content is “trapped” inside the document, limiting the scope and flexibility of content applications.
- Relational databases are optimized for numeric data, not content. Storing information in a relational database requires knowing in advance the structure of the data, a requirement that is generally incompatible with the variability and heterogeneity of enterprise content. In order to force content into a relational database

it must be shredded – converted to a rigid relational schema. Shredding is notoriously inflexible, demanding significant IT investment to update the database when the content’s structure evolves.

- Search engines offer very limited functionality. While straightforward to deploy, search engines cannot fully address corporations’ content challenges. Search tools generally offer only a list of links to relevant documents; they have no way to transform, update or annotate content or synthesize multiple documents into a new publication.
- Enterprise content management applications create incompatible content silos. Content management systems are generally delivered as a closed architecture, incorporating a proprietary content store, search environment, conversion capabilities and analytical tools. ECM packages are difficult to extend or enhance with best-of-breed software components. Customers with specific needs are often forced to purchase multiple incompatible systems, each with its own proprietary content store.

In contrast, applications implementing the OCA model are open, extensible, future-

proof, and flexible, built on a platform that provides extreme scalability, reliability, and performance. Each component of the architecture is integrated using open standards, so OCA applications can be assembled rapidly and modified easily when conditions, content, or budgets change.

MarkLogic Server is the ideal content repository for an OCA implementation. Just as a relational database allows organizations to build mission-critical applications on highly-structured data and other “predictable” information, MarkLogic Server allows organizations to rapidly build mission-critical applications using enterprise content and “unpredictable” document types. MarkLogic Server is the optimal platform for a new class of content-centric applications that help enhance existing revenue streams, create new products and services, extract value from existing content and streamline business processes. As the core of the Open Content Architecture, MarkLogic Server connects seamlessly to a range of third-party tools for content conversion, enrichment, extraction, and visualization, providing the ultimate foundation for any content application.

**MarkLogic Server is the optimal platform for a new class of content-centric applications that help enhance existing revenue streams, create new products and services, extract value from existing content and streamline business processes.**

**Mark Logic Corporation**

[www.marklogic.com](http://www.marklogic.com)

**Headquarters**

999 Skyway Road, Suite 200  
San Carlos, CA 94070

+1 650 655 2300

**New York**

+1 646 378 2104

**United Kingdom**

+44 (0) 207 643 1712