

Library Information System

Moldova Libraries

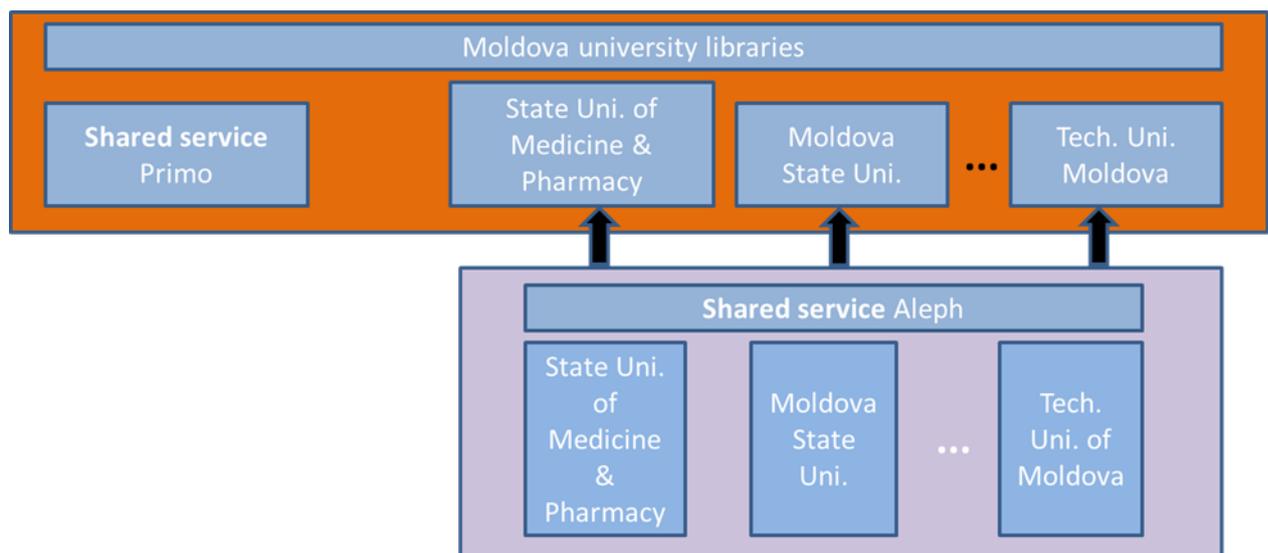
Solution Summary

Executive summary of the offered solution

Ex Libris is pleased to present a solution of information system of the libraries for the following institutions in Moldova:

- **State University of Medicine and Pharmacy "Nicolae Testemitanu"**
- **Moldova State University**
- **Technical University of Moldova**
- **Academy of Economic Studies of Moldova**
- **Alecu Russo Balti State University**
- **Chisinau "Ion Creanga" State Pedagogical University**
- **State Agrarian University of Moldova**

The offered solution is based on Ex Libris integrated library system ALEPH and the delivery and discovery system PRIMO and will be implemented as a shared solution for all above institutions and hosted local in Moldova:



In detail it means that it will be one Aleph installation with 7 Aleph 'databases' and one Primo installation allowing each institution to have an own access to their catalogue and additionally to have a common view about all 7 institutions..

The offered Ex Libris solution fulfils all required general exclusion criteria:

- **The Integrated Library System ALEPH includes**
 - **Acquisitions,**
 - **Cataloguing,**
 - **Circulation,**
 - **Inter-Library Loan,**
 - **Serials**

- **The discovery solution PRIMO includes**
 - **One Stop Search**
 - **Decoupled content**
 - **Search Engine based architecture**
 - **Single Sign-on**
 - **Integration of OPAC functionality (integrated user account/requests)**
 - **FRBR/Title grouping**
 - **Deduplication functionality**

- **ALEPH and PRIMO supports a shared service/multi tenancy architecture: all seven libraries will share one system**

- **ALEPH and PRIMO provides required consortia support, allowing each member of a consortium to have its own interface language, branding and look-and-feel**

- **Local deployment for the entire solution**

- **1st line support will be in the Romanian language.**

The proposed solution covers required 171 ALEPH users and can be anytime extended for additional users. For the discovery solution PRIMO there is no user limitation.

A detailed description of ALEPH and PRIMO can be found on the following pages.

ALEPH® 500 – System Architecture Overview

I. Multi-tier Architecture

ALEPH 500 is a library services agent, providing application services to clients via Application Program Interfaces (APIs). The ALEPH 500 architecture is based on a scalable, distributed logic model and relies on an object-oriented design.

ALEPH 500 runs under a range of UNIX machines:

SUN SOLARIS
LINUX (REDHAT)
IBM/AIX

The ALEPH 500 architecture features a multi-tier client/server model and offers extensive modularity.

- **Multi-tiered, Client/Server model**

ALEPH 500 is split into logical segments with a clearly defined interface based on message passing. Client/Server communication is based on a stateless (self-contained) transaction model. Nonetheless, the ALEPH 500 Application Servers maintain continuous connections (with time-out) to the database to ensure high performance. See *Figure 1 below*.

- **Modularity**

The key notion of the distributed logic design that underlines ALEPH 500 is modularity – both vertical (between tiers) and horizontal (within the tiers). This modularity ensures both maintainability and extensibility of the system and ensures easy integration with new technologies and concepts. ALEPH 500's modularity plays a crucial role in the scalability potential of the product, and is design criteria that libraries should insist on.

ALEPH 500™ Server Multi-tier Architecture

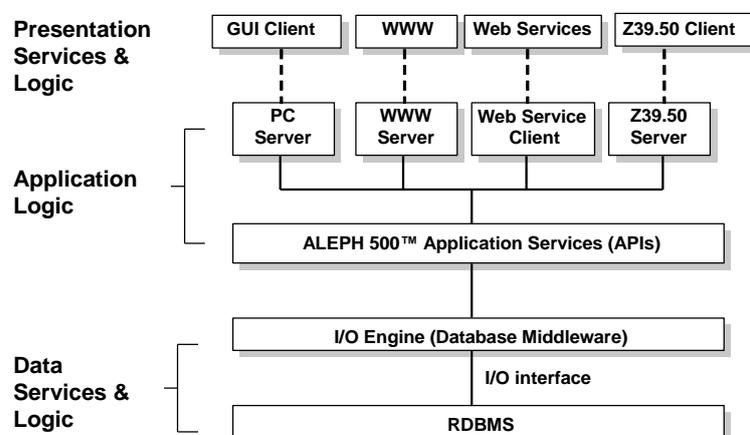


Figure 1

Tiers in ALEPH 500™

- **Client tier** – The “public face” of ALEPH 500, the client software includes a graphical, windows-based client for staff functions, a library-supplied Browser of choice, either Netscape or Microsoft Internet Explorer. ALEPH 500 clients can be run under Windows 98 (as long as Microsoft continues supporting this version), Windows NT, Windows 2000 or Windows XP.
- **Application Servers tier** - A front-end tier that is composed of dedicated client software, which translates queries from the Client software to a uniform format and directs them to the relevant Application Service object (API). Once translated, all queries have the same format, regardless of their origin (Z39.50 client, ALEPH 500client, WWW browser or standard character mode terminal).
- **Application Services tier (APIs)** - The core of ALEPH 500 is the application services tier, composed of sets of services which are then available via the ALEPH API. The API provides library services for the different clients. For example, a FIND API provides FIND services to all clients once a FIND query is invoked by one of the clients (WWW, Z39.50, telnet or ALEPH proprietary). As part of its open system architecture, ALEPH 500 includes mechanisms to integrate new services as well as to extend the Application Services to other clients or applications, providing considerable potential for extensibility.
- **ALEPH 500 Database Middleware (or I/O Engine)** - High level database management layer. A logical server which provides data services to the application services objects. It contains a group of objects, which mediate between the application and the database. The I/O Engine translates an application request to a sequence of database commands. In addition, the I/O Engine provides SQL enhancement which is required because of the textual, non-formatted nature of library catalog data.

To optimize data update and retrieval, the ALEPH 500's I/O Engine exploits knowledge about the special characteristics of the DBMS. In this logical layer ALEPH 500 incorporates its experience and know-how of libraries' special data structures and formats. Having an intermediate level of the I/O Engine between the application and the DBMS ensures maximum flexibility of DBMS logical and physical design. The ALEPH's I/O Engine layer provides another means of **performance tuning**: It accommodates distribution of services between different hardware servers. For example, services pertaining to the database files can be on a different server than services pertaining to the index files.

- **ALEPH 500 Database** - based on Oracle RDBMS. ALEPH 500 leverages SQL “enrichment” to secure high performance in the face of textual non-formatted library catalog data.

Other Characteristics of the ALEPH 500 Architecture

- **Network orientation** – The ALEPH 500 distributed logic fits into scaleable network configurations. With its distinguished Application Servers tier and Database Middleware tier (the I/O engine), ALEPH 500 is most suitable for intra- and inter-networking.
 - ALEPH 500 supports a range of client access (WWW, Z39.50, and ALEPH 500 proprietary Clients).

- ALEPH 500 supports access to heterogeneous database resources. Special features are included to overcome the diversity of databases, such as MATCH to locate remote/external documents, and Multi-FIND to broadcast a search across multiple databases.
- ALEPH 500 includes enhanced functionality to support centralized, de-centralized and union catalogs. See Figure 2 below.

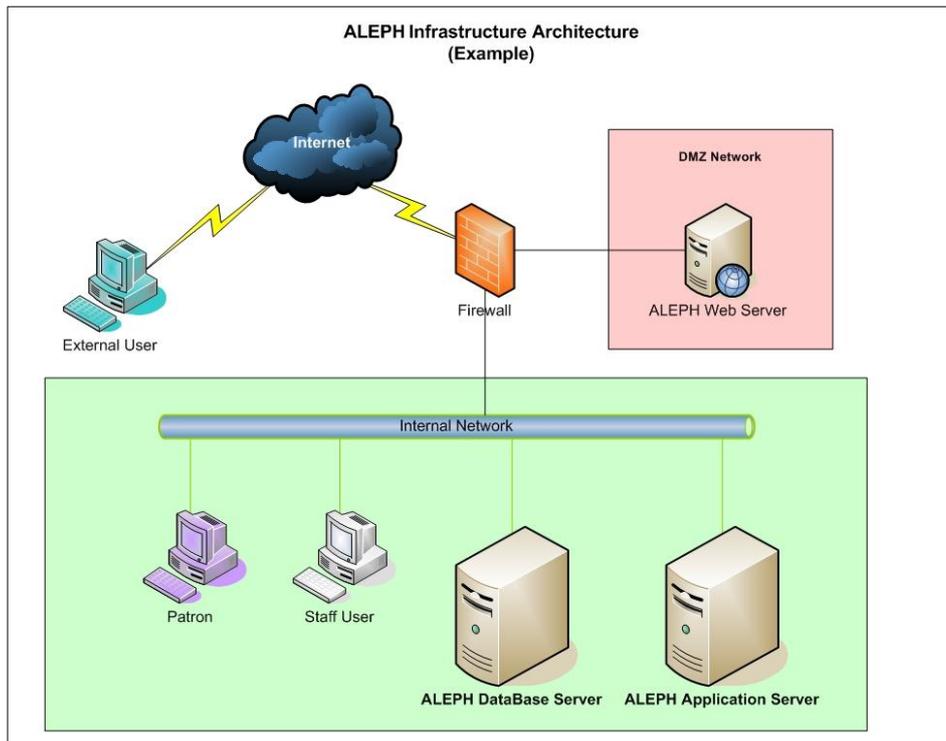


Figure 2

- **Scalability** – the ALEPH 500 multi-tier, distributed architecture supports a wide range of scalability possibilities, including distribution of data across disks/servers, distribution of services across servers or even a multi-server configuration with shared data.

II. Client/Server model:

ALEPH 500 Client/Server communication model is based on a multi-tier transaction processing model (agent). ALEPH 500 supports several client access methods: WWW, Z39.50, and proprietary clients.

ALEPH 500 Proprietary Clients

Communication is based on a stateless transaction model with constant connection (until time-out). In other words, each transaction is self-contained, encapsulated and context-free to the extent that all relevant information is contained within the transaction. This means that communication can stop and resume at any time without any impact on the work-flow.

Since one of the most time consuming elements in a client/server communication overhead are opening/closing of connections between the Client and Transaction Server and between the Transaction Server and Database Server, the ALEPH 500 communication model incorporates "state full" behavior in terms of keeping constant connection between the Client and Transaction Server and between the Transaction Server and the Database Server. Each constant connection is always limited by a time-out parameter.

ALEPH 500/WWW

The ALEPH 500 WWW server is composed of a WWW Front server and one or many WWW Server(s). Although WWW communication is based on a stateless transaction model, in order to reduce time consuming operations of opening/closing database connections, the ALEPH 500 WWW servers maintain a constant connection to the database.

III. Database structure and architecture

General

Each ALEPH 500 site is composed of 5 interrelated, yet separated database units: Authority, Bibliographic, Holdings, Administrative and System-wide administration. Each database unit can have many occurrences with many-to-many links to the other database units. The ALEPH 500 database design supports a wide range of database configurations and implementations, including independent installation of different units on different hosts. The ALEPH 500 database design plays an important role in the scalability potential of the product.

Bibliographic Unit (BIB)

Includes a bibliographic catalog in MARC format (MARC21, UNIMARC, KORMARC or other variation) plus all of the catalog's indexes. Every site can setup as many bibliographic units (databases) as desired. For example: central catalog, special catalogs, union catalog, etc.

Authority Unit (AUT)

A lexicon of authorized headings for bibliographic cataloging. This is a catalog (database) of Authority records containing MARC or non-MARC authority records. It serves two main functions:

1. Enrichment of the headings in the bibliographic data (including references).
2. Automatic correction of headings and bibliographic records.

A site might implement multiple authority databases (e.g. LC Names & Subjects, MESH, etc.).

The underlying structure of an Authority Unit is identical to that of a Bibliographic unit. This enables, for example, application of ALEPH 500 import/export mechanisms to the Authorities database. In addition, it means that a library can define any set of indexes for its authority databases.

Holdings Unit (HOL)

A database of MARC Holdings records. A Holdings Unit is commonly designed per administrative unit to include information such as location, and any other local information. The Holding Unit is the basis for all library administrative activities, such as circulation, acquisition, item control, serial control, etc.

The underlying structure of a Holdings Unit is identical to that of a Bibliographic and Authority units. This enables, for example, application of ALEPH 500import/export mechanisms to Holdings databases as well. It also means that a library can define any set of indexes over its Holdings databases. In addition, it is possible to enrich a library's bibliographic database with fields derived from a Holdings database. This has a very useful consequence of enabling search of a bibliographic database by local search terms, such as location, material attributes, local notes etc.

Local Administrative Unit

Consists of all administrative data pertaining to the institution – Circulation, Acquisition, Items Control, Serials Control and management of Library Patron privileges. This information can be controlled as the level of a single Holding Unit.

ILL Administration Unit

An administrative unit, within the global ALEPH 500 administrative structure, that handles ILL requests for both borrowing and lending requests.

Library Administration Unit

Consists of all administration data common to all libraries of a single ALEPH site. For examples: Authorizations, Profiles (defaults) and User registration.

There is a link between the bibliographic record in the bibliographic database and one or more Holdings records in the Holdings database (see also the description of the Bibliographic records, in the section below). Item records (copies or issues) are linked to Holdings records.

The databases in ALEPH can have one-to-one, one-to-many, or many-to-many relationships, allowing for great flexibility in setup. For example, multiple bibliographic databases can share a single authority database. Systems that have multiple administrative databases can share a single patron record. In library environments where several libraries share a single bibliographic database but wish to maintain autonomy for their acquisitions and serials information, they can have multiple administrative units, keeping funds, vendors, orders, and serials check in separate while still sharing bibliographic data. Each library can have its own bibliographic records (i.e. duplicate records per title), with a single merged view for library patrons.

ALEPH Catalog Structure

Each ALEPH 500 database (BIB, AUT and HOL) is composed of four logical components:

- Records in a MARC format.
- Headings records (Bibliographic indexes, sometimes named Access in ALEPH)
- One off index records (one index record is linked to a single BIB record)
- Free-text words

An ALEPH Catalog - (BIB, AUT or HOL Unit/s)

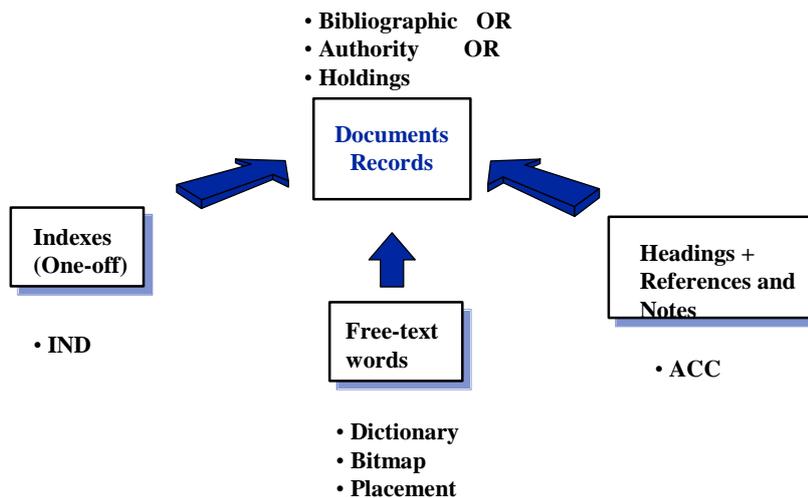


Figure 3

The logical components in the ALEPH 500 databases divide into two types: source data (mainly, the Bibliographic records) and derived data (Indexes records, Free-text words records, Headings records). This division is designed to ensure database consistency (by means of data redundancy): Document indexes are maintained as derived data. This means that, at any point, indexes and free-text words can be recreated, on the basis of the document records.

The special structure of a library database and its unique characteristics, in particular:

- No limit on the number of characters
- No limit on the number of fields per bibliographic record
- Non-limited repeated fields of varying length
- Fields containing unstructured sub-fields
- No limit on number of access paths indexing to the document record
- Fields composing more than one heading list
- Heading lists composed of more than one field

The ALEPH 500 database is constructed especially to deal with these unique characteristics. The following paragraphs present the concepts and organization of the ALEPH 500 database divided into different components:

A. Bibliographic Records

ALEPH 500 bibliographic records are stored in **DOC** files. There is a separate DOC file for every type of database BIB, AUT and HOL. AUT (The Authority database) and HOL (the Holdings database) conform to the same structure and support full search possibilities (according to their parameter table setup). Headings and Index files, for global management, can be defined as well. In addition to standard MARC tags, the records can also contain other fixed/variable length fields and management flags, as necessary.

Several **Logical Bases** can be defined within a single Bibliographic database to enable grouping bibliographic records with common characteristics into sets.

A bi-directional link between a bibliographic record (in BIB) and its corresponding Holdings records (in HOL) is provided. This link also enables a **virtual** addition of holdings data relevant for search into corresponding bibliographic record, to enrich search possibilities. For example, free words extracted from a holdings record fields, will be listed as part of the corresponding bibliographic record in the BIB free-text words file. As a result, it is possible to search within BIB records on non-bibliographic criteria, such as note fields (e.g. autograph by Dali, call number, etc.), or to compose a Boolean search based on both bibliographic and non-bibliographic criteria (e.g. a book written by Hugo with Dali's autograph). More details about the BIB's word file are given in the **Free-text words record** section, below.

The ALEPH 500I/O Engine can handle a separation of the DOC file into several physical files (in addition, or instead of the ORACLE feature) to enable better I/O optimization. This can also enable parallel processing of the DBMS to accomplish different tasks (e.g., reorganization, reports, etc.). Implementation of such parallel processing features can be done only after careful system analysis.

B. Headings (Bibliographic Indexes)

A Heading (Access) list consists of a list of headings (authors, titles, subject headings, etc.) which are either extracted from specific fields of bibliographic records, or created from the AUT database. Access lists serve multiple purposes in ALEPH:

- Access lists to the bibliographic records
- Working files for global database maintenance

A heading has several characteristics:

- It can be up to 2000 bytes in length
- One heading might have different occurrences.
- There is a need to sort a certain heading in different ways (for e.g. for the professional and nonprofessional users).
- In certain cases, there is a need to combine several types of headings into one index, in order to compose one list, e.g., titles and series tracing.

Because of the above characteristics, ALEPH 500 maintains a derived index file.

There are 2 logical files: Directory of Headings (ACC) and pointers (ACCDOC):

- ACC file contains the list of headings. Each heading (title, subject, author, etc.) has a pointer to ACCDOC.
- ACCDOC contains a list of documents containing the specific heading.

An ACC record is created either as a derived data from a bibliographic record, or from a corresponding AUT record field.

The library can define which Headings lists should be created and, for each Headings list, which bibliographic fields will be included in the list.

Publications related to a heading are sequenced on-the-fly according to the user's choice. A different default sort can be set for cache heading list. Each Headings list has a filing procedure assigned to it for sequencing of entries within the list. The procedure sets up rules for filing (such as treatment of special characters). In addition, the cataloger can alter the filing sequence of individual headings by indicating to the system to ignore

certain characters while filing (for examples: treat the string 'The law of Torts' as 'Torts'; 'The book' as 'book').

A bibliographic field can be "sent" to one or more headings lists (for example, series field sent to both titles and series lists). A headings list can be built from one or more bibliographic fields or sub-fields (for example, titles list can be built to include titles, sub-titles, series and other fields).

In the OPAC, a headings list is displayed when the SCAN command is invoked. When a list of headings is displayed, the user can browse forward and backward, display related documents, or display cross-references and notes.

C. Index Records

Classic indexes are derived from the bibliographic information. They allow direct access to a bibliographic record by an index term (such as ISBN or call number). Each BIB record field creates a separate occurrence in the index. Index entries are not necessarily unique.

The classic index information in ALEPH 500 is stored in the INDEX file. The ALEPH INDEX record contains index terms built from the list of fields defined as indexes (defined by the library) and the pointers to their related documents. The indexes text is transformed through a predefined filing routine.

D. Free-text Words Records

Free Text Words Indexes

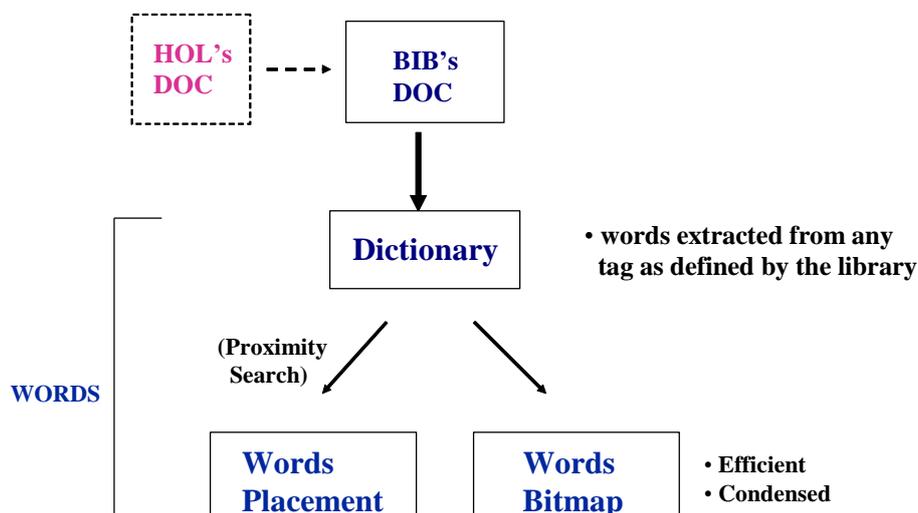


Figure 4

A free-text words list is based on individual words extracted from selected record fields (e.g. subjects and/or titles and/or sub-titles and/or abstract, etc.). The choice of the fields is flexible and determined by the application manager via the ALEPH 500 configuration table. The list is sequenced alphabetically and each word in it is unique (appears only once in the list).

The free-text words data is distributed in 3 logical files:

- Words Dictionary: each record consists of: the word and pointers to the Words Bitmap and to the Words Placement Table.
- Words Bitmap: An efficient condensed bitmap file of all words occurrences within the documents.
- Words Placement Table: Each word record contains the placement of the word per document. This information allows proximity search.

Update of free text words is part of the ongoing indexing process that runs in the background. Free text indexing of the Bibliographic Unit (BIB) database can be enhanced to include also the following:

- Local item information: words from the BIB's corresponding HOL record/s + selected words from its/their corresponding item records.
- Related terms: Words from relevant cross references
- Linked information: Words from documents in external sources (full text retrieval systems) which are linked to the BIB record.

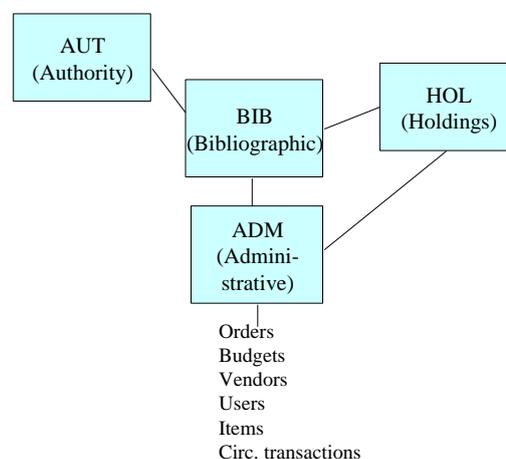
ALEPH 500: The Ideal Solution for Consortia

The ALEPH system data is stored in various databases – the “building blocks” of the system. These databases include Bibliographic, Holdings, Authority, and Administrative databases – the Administrative database (or databases) contain item, order, fund, vendor, and serial check-in information. A standard ALEPH implementation consists of a single Bibliographic, Holdings, Authority and Administrative database. Licensing for the system is made up of a combination of the base software license and concurrent user licenses.

Any ALEPH installation can be broken into multiple regions accommodating multiple libraries within a single instance of the software.

The ALEPH 500 architecture easily accommodates a variety of system configurations, and for that reason it is ideal for consortia. The ALEPH 500 system data is stored in various databases – the “building blocks” of the system. These databases include Bibliographic, Holdings, Authority and Administrative databases – the Administrative database (or databases) contain item, order, fund, vendor, and serial check-in information. It is possible to create relationships between the various databases linking them in a one-to-one, a one-to-many or a many-to-many relationship.

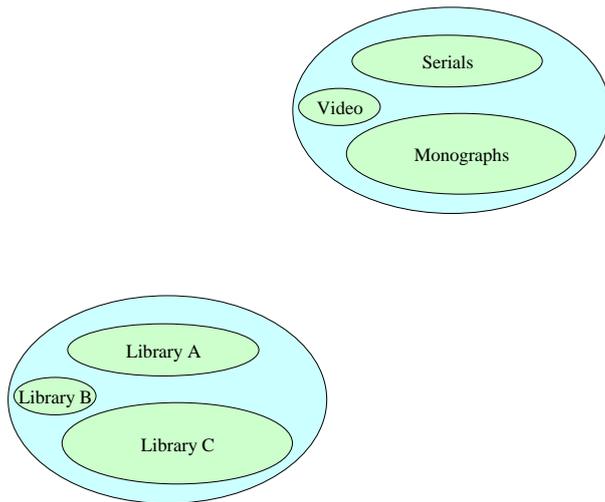
The basic database relationships are illustrated below:



The different databases in a single installation are integrated so that information from one can be displayed together with the other. The Bibliographic record is the central record and the Authority, Holdings and Administrative records are linked to it. Search indexes are derived from the Bibliographic record, with additional holdings and item information virtually integrated into the bib for indexing and display purposes.

The Bibliographic database contains bibliographic records, which describe the bibliographic material of all types. Ownership of records is determined by using the OWN field, an ALEPH-specific field in the MARC record. This OWN field will correspond to the owning institution. Display and edit rights are based on the OWN field code and the password privileges of the staff member logging in to the ALEPH system.

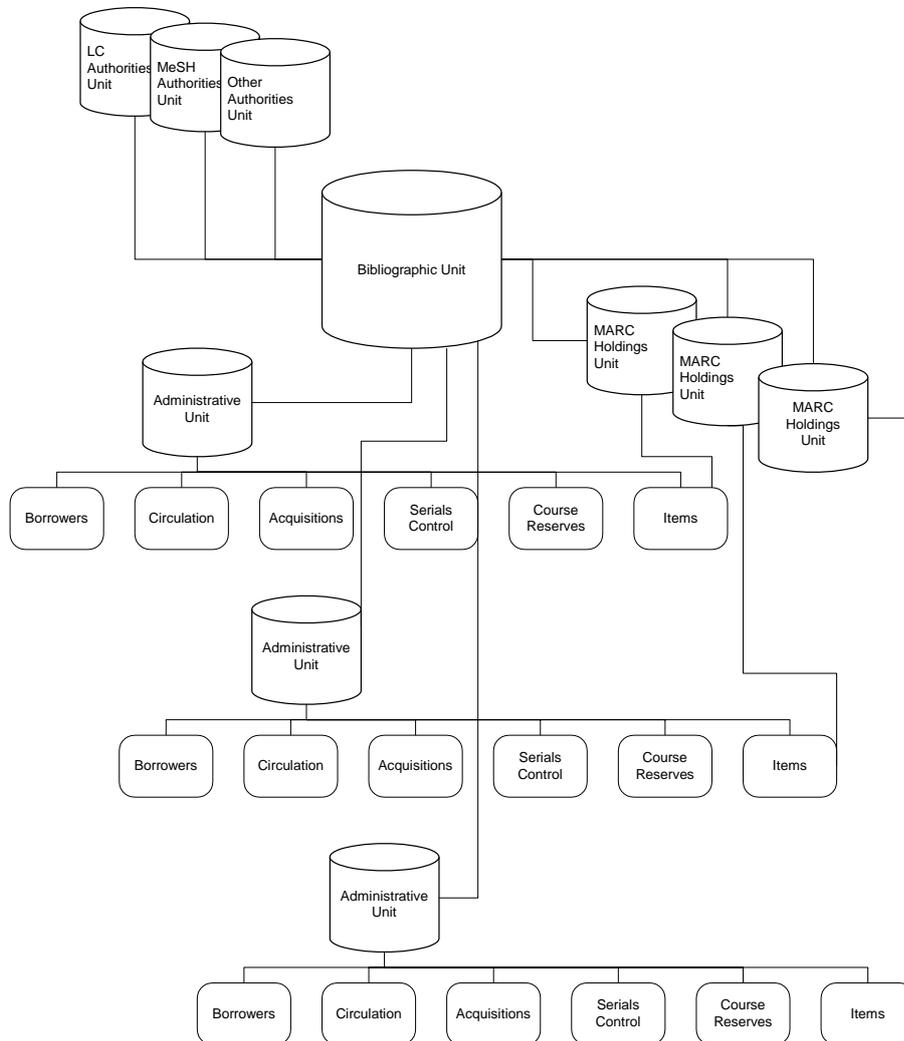
A central concept of the Bibliographic database is that of “logical bases”. A logical base is a sub division of the Bibliographic database. Logical bases can be defined for practically any group of records – by subject, material type, language, or location.



Options for Logical bases

The power of the logical bases means that in a consortia environment, individual “catalogs” can be defined for each library, while still allowing for easy expansion to searching the combined holdings of the institutions.

Various relationships can be created between databases. One recommended environment for a multi-library site includes separate Holdings and Administrative databases for each institution as illustrated below:



By maintaining separate administrative database for each institution, all circulation policy tables for individual libraries can be separately administered allowing each library to determine loan periods, fine policies, etc. Universal borrowing can also be supported if desired.

PRIMO – A brief description

1) Primo Search

The information resources that libraries make available to users are varied. In their effort to provide a comprehensive selection of quality information resources, libraries, museums and archives offer their own collections – physical or digital – as well as collections from sources such as other libraries, primary and secondary publishers and aggregators, organizations holding repositories of scanned e-books, images, audiovisual materials and Web sites.

Information professionals face numerous challenges in choosing appropriate information resources, licensing them when necessary, and presenting these heterogeneous collections to patrons in a compelling and easy-to-use way. A single point of access to all of their collections is the ultimate goal.

The Ex Libris Primo® discovery and delivery solution meets this goal by providing several complementary technologies that leverage an institution's unique collections and position these collections prominently in the information landscape of the institution. A library, archive or museum can deploy any combination of the Primo technologies; the ones that it chooses depend on the specific collections that it wants to offer its users, the technical resources that are at its disposal, the collaborative relationships that it can establish with other institutions, and its agreements with information providers. Regardless of the technologies deployed, users have a similar, though not identical, experience when looking for information across all resources.

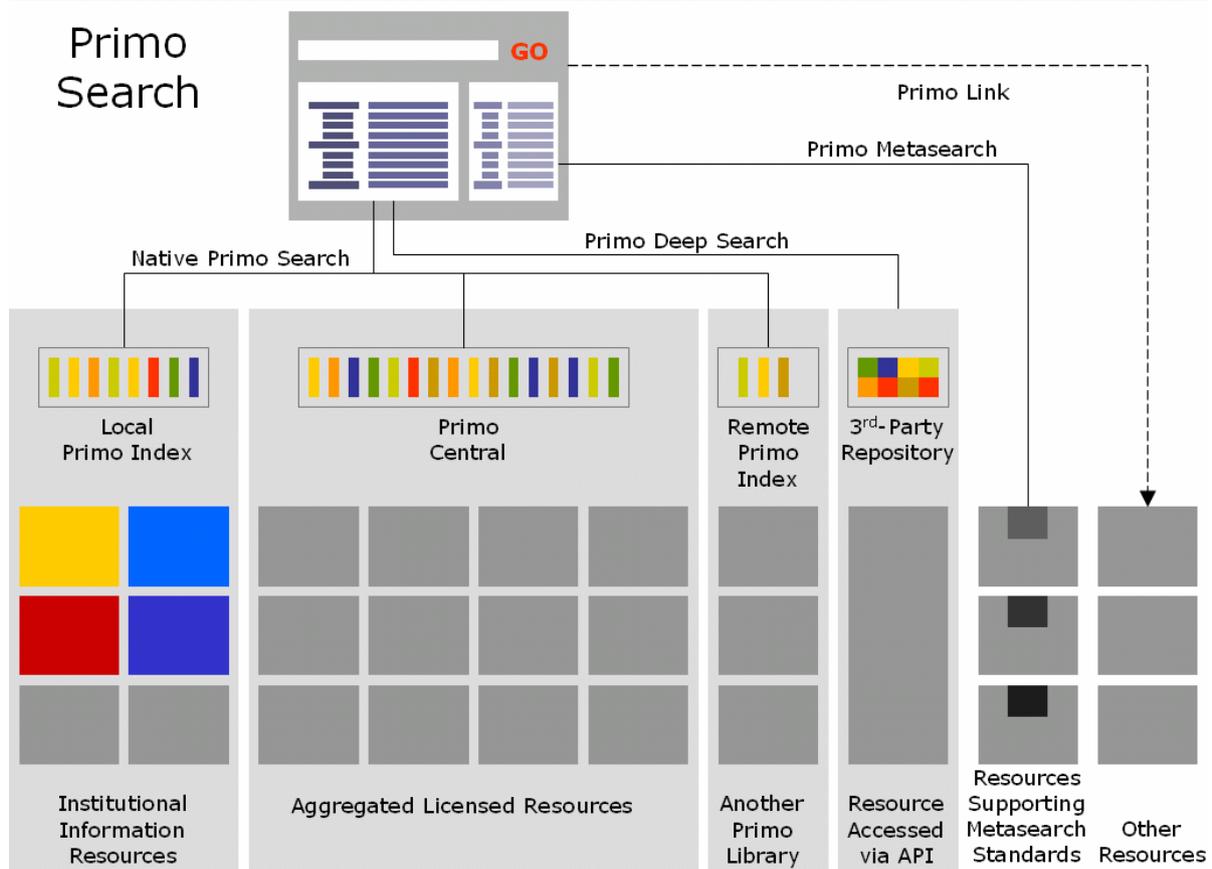
In deploying Primo with its range of search technologies, institutions have full control over the selection and description of their collections, the way in which the collections are presented, and the discovery and delivery options offered to users. Based on an infrastructure that is independent of commercial publishing and handles the full spectrum of scholarly materials, Primo encourages collaboration among libraries, archives, museums, information providers, vendors, and other stakeholders. With Primo, institutions benefit from other collections without having to replicate the work done elsewhere and can focus their energy on exposing their unique collections, which they

can contribute to the community effort to provide as much scholarly data as possible to users.

The search technologies that Primo enables institutions to deploy are as follows:

- **“Native” Primo search**, enabling Primo to search in one or more Primo indexes and blend the results into one result list. Several deployment options are available for the “native” Primo search technology, and an institution can choose to deploy one or more of them:
 - Search in a **local Primo index**, an index that is derived from the library catalog and other collections under the governance of the library or the institution.
 - Search in **Primo Central**, a centralized Primo index that encompasses data of global or regional significance harvested from primary and secondary publishers and aggregators. Primo Central is maintained by Ex Libris and offered as a service.
 - Search in a **remote Primo index**, an index of another Primo institution.
- **Primo Deep Search**, which enables Primo to perform optimized just-in-time searches in remote resources that are not indexed by Primo and are compliant with the Deep Search requirements.
- **Primo Metasearch**, which enables Primo to perform just-in-time searches in remote resources that are not available through Primo Central or through Primo Deep Search. The Primo Metasearch technology uses standard metasearch protocols.
- **Primo Link**, which enables Primo to direct users to the native user interface of a specific information resource.

We describe briefly the various Primo search technologies and explain how only by offering a central index and a combination of such search technologies can a search solution respond completely to the needs of academic and research institutions that rely on a multitude of information resources, information providers, material types, and access methods.



a) Native Primo Search

The Primo publishing platform creates and maintains a Primo index by harvesting and processing bibliographic metadata and any other available data stored in local or remote resources. Primo can divide its index into several nodes, or “sub” indexes, to improve load balancing and to distribute data-harvesting responsibility among various collections. Because Primo can perform distributed searches in multiple Primo indexes, it can search simultaneously in very large institutional collections, in collections that other institutions control, and in a central repository managed by Ex Libris. Primo blends the results originating from multiple Primo indexes and displays them as one result list.

b) Local Primo Index

By harvesting and processing metadata from a library’s catalogue, its digital collections, its course materials, and any other harvestable information resources, Primo creates a

local Primo index for the institution. The pre-processing of the bibliographic data and the complete control that the library has over the hardware and software that it uses to create and maintain the local index enable the library to present its collections to best advantage and to optimize the searching capabilities in these collections for the user community.

In the process of building a Primo index, an institution can apply various configuration and adjustment options to normalize, augment, de-duplicate, and group the harvested bibliographic records. Furthermore, the institution can control the information that is used for the search mechanism, for the display, and for the faceted categorization of result lists, and it can tailor the relevance ranking of the results to the institution's profile.

The Primo search engine is based on the Lucene search routines, which have been enhanced to take full advantage of the specific nature of library, archive and museum materials. The indexing and the search process take into account metadata standards, such as MARC, Dublin Core, and MAB; library-oriented classification, such as the various subject headings; and relationships between items, such as various expressions of a single work. As a result, Primo offers sophisticated searching and an extremely rapid response time.

To create a local Primo index, the institution needs to harvest, process, and store metadata. The institution might also choose to harvest the primary data – for example, from local repositories of articles. However, because institutions offer a wide range of collections, harvesting all of them to bring them into the local Primo index is not always feasible. Ex Libris optimizes the harvesting of information resources that are relevant for all Primo customers by offering a centralized index, Primo Central.

c) Primo Central

Primo Central is a centralized Primo index that is managed by Ex Libris in a cloud environment and offered to Primo institutions as a service. The information that is harvested and described in the Primo Central index – primarily bibliographic metadata, abstracts, and the full text of scholarly materials – is of great relevance to the global Primo community.

Technologically, managing Primo Central is similar to managing any other Primo index at an institution; however, the Primo Central index is managed by Ex Libris in collaboration with information providers – primary and secondary publishers and aggregators – for the initial and ongoing harvesting and processing of their data and for the handling of authorized access to licensed resources. Primo uses the harvested data to provide the discovery capabilities to users; however, delivery of the full text is carried out by the information provider’s system. Where applicable, the delivery is facilitated by an institutional link resolver such as SFX.

The Primo Central index is readily available to all Primo customers and is highly scalable, covering hundreds of millions of scholarly materials, primarily articles and e-books. When required, authentication and authorization apply when a user accesses this central repository to obtain information based on the subscriptions of the user’s institution.

The Primo search engine uses the Primo Central index along with the local Primo index. The search results from the central repository are blended with the results from the local index to form one set of results. Primo optimizes searches in both the local index and the centralized index and provides performance that is similar to that of searching only in a local index.

d) Remote Primo Index

The Primo solution can use Primo indexes of other institutions – remote Primo indexes – as well. By configuring Primo to aggregate results from multiple Primo indexes and blend them into one result list, institutions can extend their offering and leverage the unique collections of other institutions, thus broadening the scope of scholarly data in a simple and cost-effective manner.

e) Primo Deep Search

With the Primo Deep Search technology, an institution can offer access to remote, third-party information resources that are not available for Primo Central indexing but are Deep Search compliant. Using this technology, searches in remote resources resemble searches in the institution’s local Primo index: the response time is very short; the result lists are sorted by relevance, enabling Primo to combine them with results obtained from the local Primo index, the Primo Central index, and other remote Primo indexes; and

relevant facet information, when provided by the remote resources, is available to Primo for display.

Institutions can deploy the Deep Search technology to access information resources that comply with the following requirements:

- The response time is less than a second.
- The results are sorted by relevance.
- The information resource's index can be accessed directly through an application programming interface (API).
- (Strongly recommended) Facet information is provided, including facet titles and the number of records per title.

The Deep Search component provides an efficient way to access remote resources that Primo does not harvest for Primo Central, such as the DTV Article Database Service (DADS) repository of tens of millions of scholarly articles, maintained by the Technical University of Denmark and accessed by Primo at the Royal Library of Denmark; and the repository of 1.8 million e-books at Duxiu.com, accessed by Primo at Shanghai Jiao Tong University. The deployment of the Deep Search technology requires an institution to obtain the cooperation of information providers in furnishing the requisite API.

f) Primo Metasearch

To enable users to search in remote resources that an institution cannot or does not want to harvest and are not available through Primo Central or Primo Deep Search, the Primo solution makes use of a metasearch component – the metasearch engine of the Ex Libris MetaLib® gateway and metasearch system. This component enables a patron to use the Primo search interface to launch a search in multiple heterogeneous resources. The metasearch component, employing standard protocols or special programs, converts the unified Primo query to the diverse search syntaxes that are supported by the information resources in which the user wants to search; delivers the query to the search engine of each search target; and collects the results once they are available at the search target. Primo then processes the results and either combines them with the results obtained from the Primo indexes or displays them as a separate result list.

The metasearch technology has been deployed by Ex Libris since 2001 and relies on a comprehensive knowledge base that enables the MetaLib metasearch engine to invoke a

search in hundreds of databases. Implemented by 1500 institutions around the world, MetaLib is one of the most widespread metasearch systems in use today.

g) Primo Link

Primo Link enables users to discover information resources in Primo, link to a resource, and search in the resource's native interface.

Achieving a single access point for searching in all the information resources that a library offers its users is not always possible, given the number and diversity of resources. Even when used together, the local and remote Primo indexes, Primo Central, Primo Deep Search, and Primo Metasearch might fall short in addressing the technological, legal, or other aspects of some resources – for example, subject guides and resources that do not have a bibliographic structure, such as data sets. Furthermore, users sometimes want to access an information resource that addresses a specific need or that provides unique services. Primo, as the library's information gateway, grants such direct access. By harvesting the institution's SFX and MetaLib knowledge bases, Primo obtains a list of institutional e-resources (e-journals, e-books, and databases) and incorporates them in the Primo index, thus enabling users to look for information resources and link directly to them.

h) Primo – the Best of All Worlds

No single technology can cover all the discovery needs of an institution. Primo offers a spectrum of technologies and deployment options, as well as a centralized index, that complements each other and together provides a compelling and friendly search environment.

A combination of all these options provides a complete environment and enables institutions to emphasize their uniqueness and to maintain their control over a number of factors: the selection of resources; the way in which resources are presented and made available to users; and the institution's investment in collections, hardware, software, and the maintenance of systems and data, be it a local or hosted environment.

2) Primo System Architecture

Primo provides users with a universal solution for the discovery and delivery of print and digital information sources, regardless of format and location.

Primo offers advanced, high quality search results based on existing metadata. After finding the desired materials, Primo can facilitate the delivery of the physical item from the library or provide immediate access to digital copies.

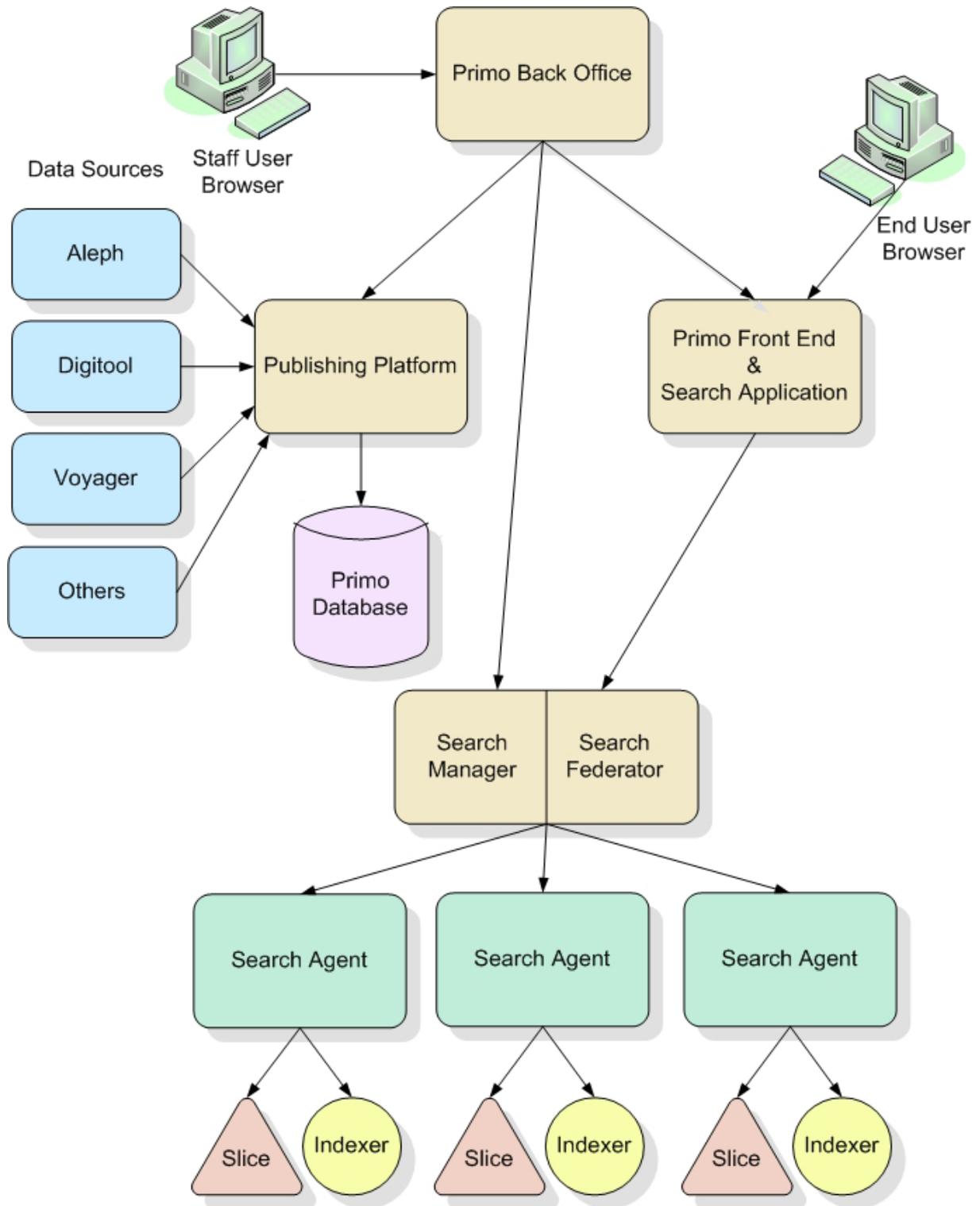
Primo is a web-based solution that runs under UNIX. It comes with a built-in publishing platform that is used to process raw data (MARC, XML, MAB, and Dublin Core) and turn it into high-quality, indexed information that is saved in Primo Normalized XML (PNX) format. The processing of the data source information is performed by pipes that understand the various library source formats. The PNX data can be quickly and efficiently searched by the search engine.

a) Components

Primo consists of the following components:

- Publishing Platform
- Indexer
- Back Office
- Search Federator
- Search Agents
- Search Engine
- Front End
- Database

The following figure illustrates the logical view of Primo, including its various components. Each of the system components is described in the following sections.



b) Publishing Platform

The Publishing Platform enables the institution to consolidate the full range of institutional resources, including print collections, digital repositories, and electronic resources. The Publishing Platform manages the harvesting of this raw data from various data sources and its transformation into high quality, indexed information that can be quickly and efficiently searched by the Primo search engine. This data is stored in the Primo Normalized XML (PNX) format.

Primo can harvest and normalize any data in standard XML format. Standard formats, for example, MARC, MAB and Dublin Core, have built-in template mappings. Templates can be customized during the implementation process. The processing of each data source is performed by pipes that recognize various source formats.

The Publishing Platform performs:

- Intelligent harvesting of data via FTP, file copy, or OAI.
- Normalization of the data to the PNX format, which is stored in the Primo database.
- Enrichment based on algorithms and external information.
- De-duplication based on algorithms.
- FRBRization.

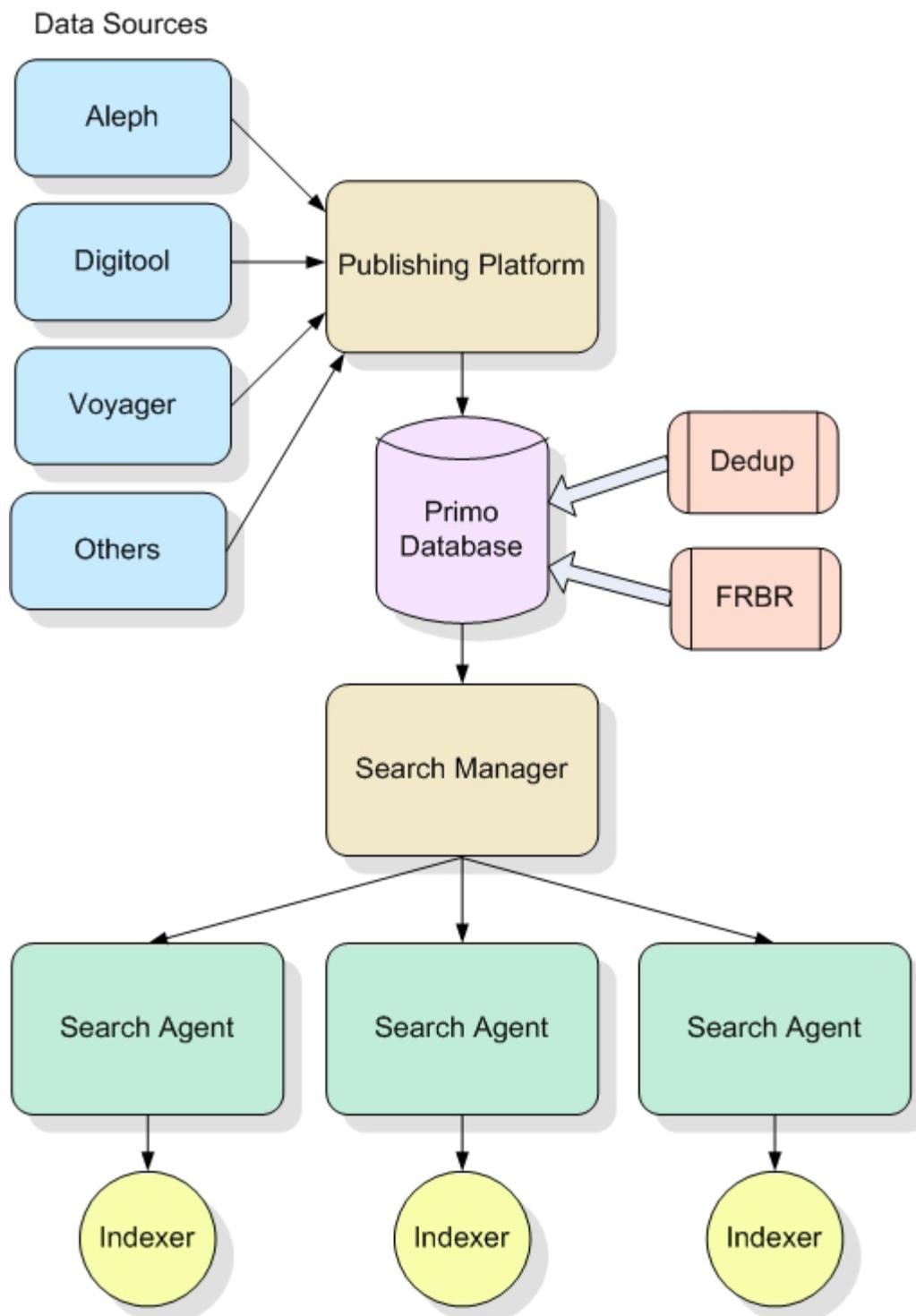
The Normalization Mappings and Enrichments are configurable using the Primo Back Office. The Publishing Platform supports scheduled and unattended harvesting and processing of different data formats, while enabling interactive monitoring and control over the entire set of activities.

c) Indexer

The Indexer is part of the Search Engine and is used to create slices. The Search Engine supports multiple slices of search data. Slices are an efficient way to allocate groups of equal-sized chunks of memory. The Indexer automatically swaps itself with search machines so that slices can be made available without down time. The Indexer splits the actual data into manageable slices, which are loaded into memory. Each slice is searched by a dedicated thread; however, multiple slices can reside on the same machine so that

all CPUs are utilized. Multiple machines may be used so that the system is not limited to the memory of one machine.

The following figure illustrates the indexing process in the Primo system.



d) Back Office

The Primo Back Office enables configuration and monitoring of all Primo components in an easy-to-use web-based graphical interface. The configuration of Primo in the Back Office is organized by the life-cycle of Primo, and includes:

- Initial configuration
- Ongoing maintenance

In addition, you can configure elements of the system individually by accessing them through the appropriate wizard. A site map is provided for direct access to the individual tasks for the advanced user.

e) Search Federator

The Search Federator coordinates the search, utilizing all slices, and combines the search results into a unified result set.

f) Search Agents

Search agents are located on remote machines. These agents stop and start the search instance and the indexing process. When an agent starts, it sends a registration request to the federator. The agent and the federator communicate using the Java RMI protocol.

g) Search Engine

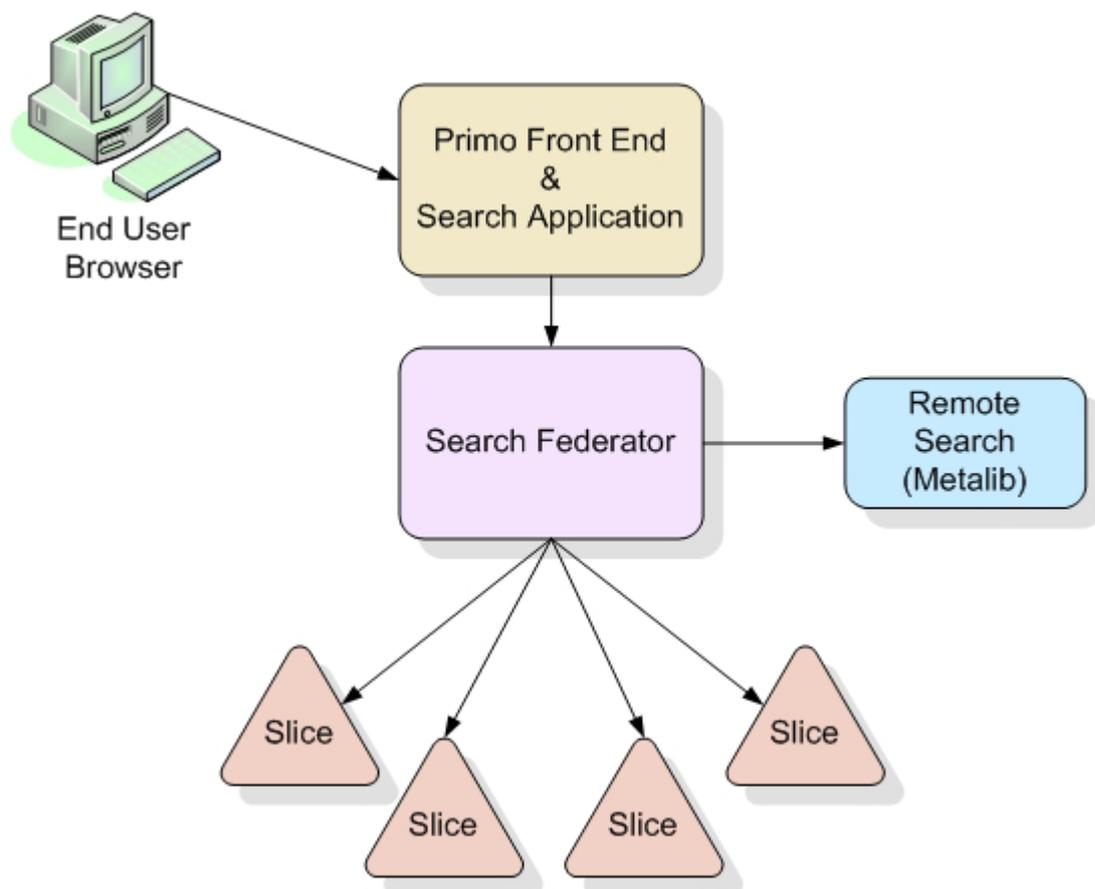
The Primo Search Engine retrieves metadata from the local PNX database table and transforms it into useful information. The Search Engine is based on Lucene technology; however, it extends Lucene functionality and supports multiple slices for very large data sets. These slices are prepared by the Indexer. Each slice is searched by a dedicated thread; however, multiple slices can reside on the same machine, so that all CPUs are utilized. Multiple machines may be used so that you are not limited to the memory of one machine. Search functionality includes faceted navigation, 'did you mean', paging, and sorting.

h) Front End

The Primo Front End user interface is responsible for all interactions with the end user. It is a search tool that is both powerful and easy-to-use. Each institution can have its own fully customized view. Every view can have one or more tabs. Tabs enable a site to divide the Primo repository and records from remote resources into resource groups or types. Within a tab, several search scopes can be defined. Search scopes group records so they can be searched together.

Using the Front End user interface, the end user searches the PNX database table for relevant items. After discovery, Primo indicates the real-time availability of the resource in the source system and interacts with the source system to provide more information about the resource or delivers the resource to the end user. The Front End generates the actual HTML pages viewed by the end user.

The following figure illustrates the querying process in the Primo system.



i) Primo Database

The Primo database is based on Oracle RDBMS. The Oracle database contains the following primary types of content:

- Primo PNX records and user-contributed information (reviews, tags).
- Monitoring information, including statistics, detailed information on searches, etc.
- Primo configuration information.