Voyager

Understanding the Voyager Enterprise Architecture

June 2013

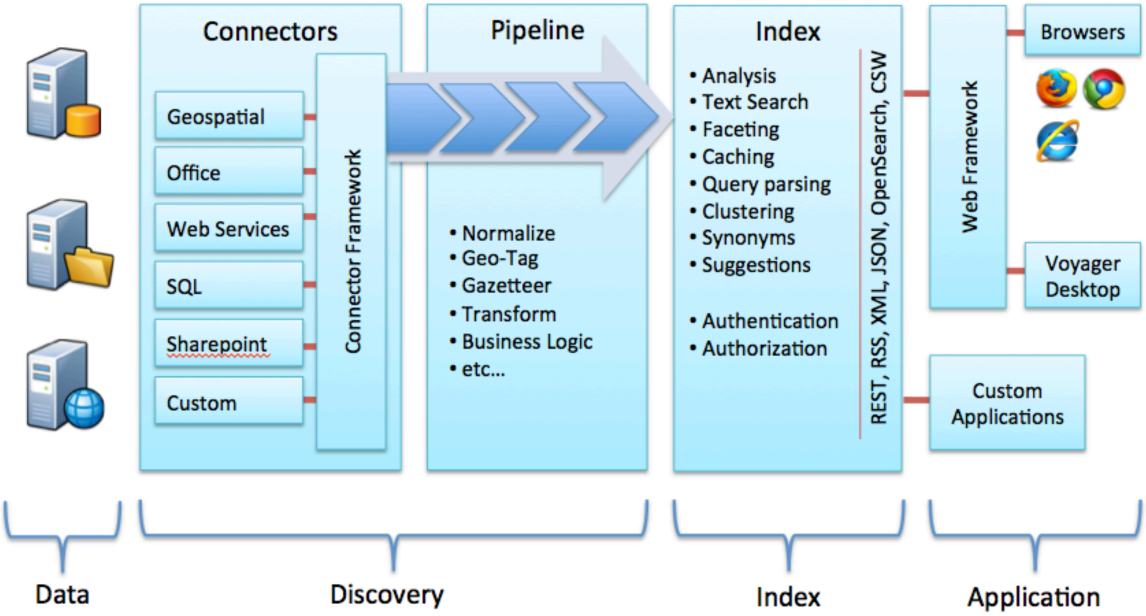
www.voyagersearch.com

www.voyagersearch.com/demo

An Introduction to Voyager

Voyager provides an easy-use, out-of-the-box, enterprise search solution that can bring together a vast variety of content from almost any source. It is open and scalable, but with advanced security. Voyager can provide internal and external customers with the right data at the right time. But most importantly it is the only spatially enabled, enterprise search solution.

This diagram illustrates the Voyager system in more detail and the areas where it can be extended or enhanced. We will discuss the core functionality and the extensions below.



Overall the four areas for extending Voyager are:

1. Connector for accessing other content repositories
2. Adding Extractors for new files types
3. Enriching the processing pipeline.
4. Extending Voyager with custom applications

Data Stores and Connectors

Voyager's unique spatial search openly integrates data from anywhere in the organization. Out of the box, Voyager can create an index of data from the desktop, servers or network attached drives. It can index web services whether they are Catalog Web Services, WMS services, FME web services or ArcGIS Services. Voyager can index ArcGIS Online or its sister, the on premise Portal. Voyager can index file servers. Voyager can index rows in a databases including SQL, Oracle and ArcSDE. And Voyager can link together separate Voyager instances to provide a single relevant search results set.

But what if all of your data isn't stored in these places?

Voyager can also be extended to consume content from other document management systems or search engines. The Voyager connector framework includes support for the Apache Foundation's Manifold CF project. Apache ManifoldCF is an effort to provide an open source framework for connecting source content repositories like Microsoft Sharepoint and EMC Documentum, to target repositories or indexes, such as Voyager, which is built upon Apache Solr. Apache ManifoldCF also defines a security model for target repositories that permits them to enforce source-repository security policies. Voyager can enforce these security policies and integrates with LDAP systems as well as Windows Authentication.

Voyager supports connections to FileNet P8 (IBM), Documentum (EMC), LiveLink (OpenText), Meridio (Autonomy), Windows shares (Microsoft), and SharePoint (Microsoft). Also included are a general CMIS connector, a generic file system connector, a general JDBC connector, an RSS feed connector, a Wiki connector, a DropBox connector, and a general web connector.

If further connections are needed, Voyager professional services will extend our connector framework working with the customer to define the needs.

For integrating with other search solutions, Voyager supports OpenSearch. OpenSearch is a collection of simple formats for the sharing of search results. You can use OpenSearch formats to help users discover and use a search engine as well as to syndicate search results across the web or between networks. OpenSearch was built by A9.com, an Amazon company. Search engines crawl only the surface of the mountains of data within an organization or across the web. It can glean only a small fraction of the content. Further different search engines, with different algorithms and specialties do different things well, but when they need to share results, OpenSearch let's them communicate using a common language. The OpenSearch format is now in use by hundreds of search engines and search applications around the Internet.

What Voyager Can Index and How They Use It

Voyager is also known for its ability to index geospatial data while maintaining the data's connections to documents or layers, as well as to maintain spatial reference. All of this information can be used for searching. But Voyager can also index any Internet media type, also known as a MIME type. In cases where Voyager is unfamiliar with a format, Voyager can use the simple file indexer to index whatever it can from the header of that file and then we work with customers to extend our framework to include a new file type so that we can examine it more thoroughly. A complete list of MIME types that are supported can be found here.[1]

Once the data is in the index, the user has many tools at his or her disposal for customizing the presentation of data and reporting on the data. Users can create saved searches and custom presentations of fields. They can add synonym libraries for extending keyword search. And much more! A Voyager administrator can control what users can see the data and how they can use it. For instance, is it OK for them to know that it exists, but not be able to download it? Or should it be clearly restricted?

Voyager engineers strive to make Voyager flexible and extensible while providing a core solution that is simple to use and gain value from right out of the box.

Enriching the Processing Pipeline

Once data is in the Voyager index, we can show you customized saved searches, like what documents have broken links or what data is duplicated. But we also have a processing pipeline that allows users to do things like clip and ship the data in the index, create mosaic map tiles and much more. If you don't have what you need, this pipeline is extensible.

It can be extended, for example, to not only show you where the broken links are in a map document, but fix them by pointing to another data source. Or you could use Voyager to schedule a job that would test map services and delete the ones you'd published to ArcGIS Online if the links were broken.

[1] https://en.wikipedia.org/wiki/Internet_media_type

Custom Applications

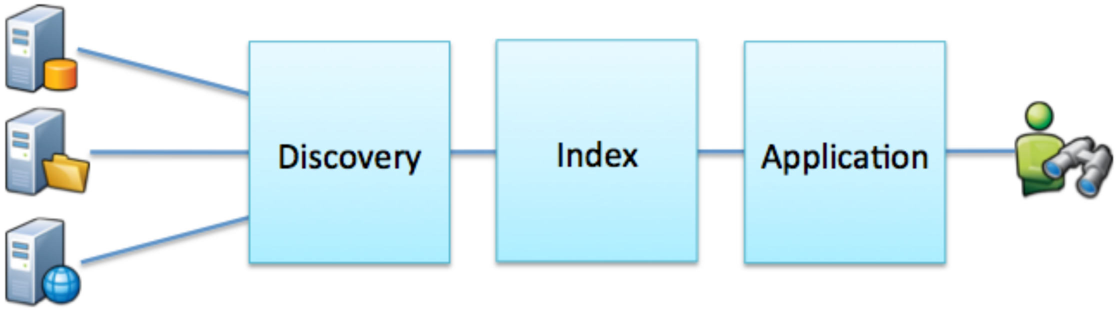
Most Voyager users access the system through a simple web browser. But what if they want to take the index further?

Voyager supports integration with custom applications or other off the shelf software. Voyager is tightly integrated with the ArcGIS family of software. In the 1.5 version, users can open layers or maps in ArcGIS directly. In 1.9, they will also be able to publish one or many map services into ArcGIS Online, or open maps directly in a GeoCortex custom application. Voyager strives to be an open platform and provides a robust API where developer could extend Voyager providing custom applications or extending the “open with…” or “publish to…” capabilities.

Voyager System Overview

On the most basic level, a single instance of Voyager performs three fundamental functions:

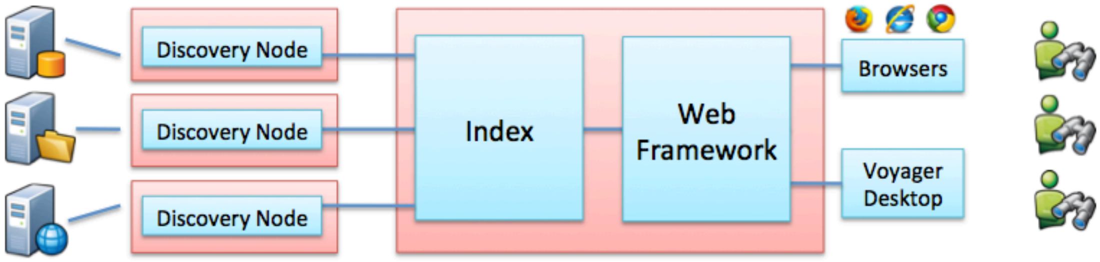
1. Discovery content
2. Hosting and maintaining the index of content
3. Providing an application — be it a desktop client or browser – for end users



On a small scale, Voyager performs just fine doing all three things. But in larger organization or with large data sets, these three functions can be broken apart to achieve optimal performance and to support an organization's needs.

Option 1: Add Voyager Instances for Discovering Data Only

In cases where the data are located in different cities or on different continents or the data are vast or dynamic, it might be useful to enhance the discovery capabilities of Voyager.



Let many machines handle initial discovery and extraction of data

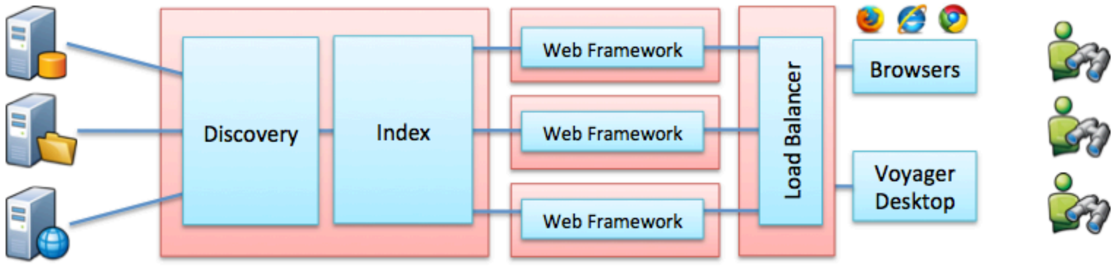
Recommendation:

Increase the discovery horsepower. It is best to have an instance of Voyager installed near the data that it is indexing. Otherwise, the initial discovery and indexing speed will pay a latency tax simply getting Voyager to the data. You can mitigate this problem by locating Voyager near the data it is trying to index.

For organizations with large and ever changing data, it is also recommended that you devote one or more instances of Voyager exclusively for discovery. Voyager's Federated search can then merge the indices created by each instance into a single set of search results for the end user.

Option 2: Load-Balanced, Failover System

In organizations where Voyager is a part of a larger system that might need to be taken offline for maintenance; where thousands of users are hitting Voyager and high-availability is required, or where Voyager is mission-critical, you should consider a load-balanced, failover system.



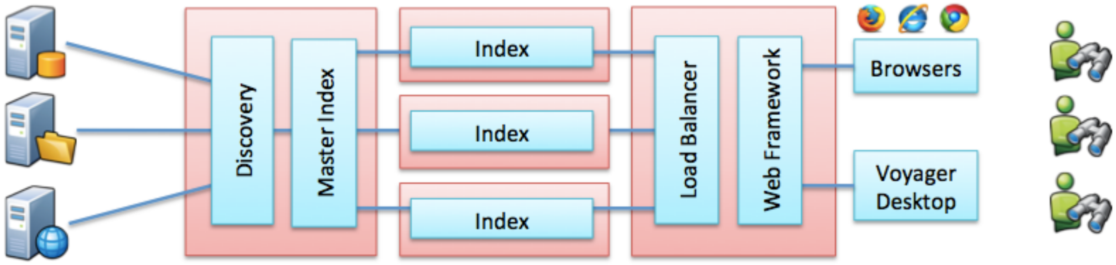
Multiple Application Servers for high availability and better performance

Recommendation:

Create a load-balanced, failover version of Voyager to handle peak load times. Have a master Voyager instance and one or many slave indices and web servers on a load-balanced framework. This provides redundancy for when a computer needs to be rebooted. It also allows more people to access the Voyager website.

Option 3: Index Replication

In organizations where Voyager must be able to continue serving customers through hardware failures or configuration changes or where indexes become large enough that it become advantageous to divide and distribute them across multiple servers.



Recommendation:

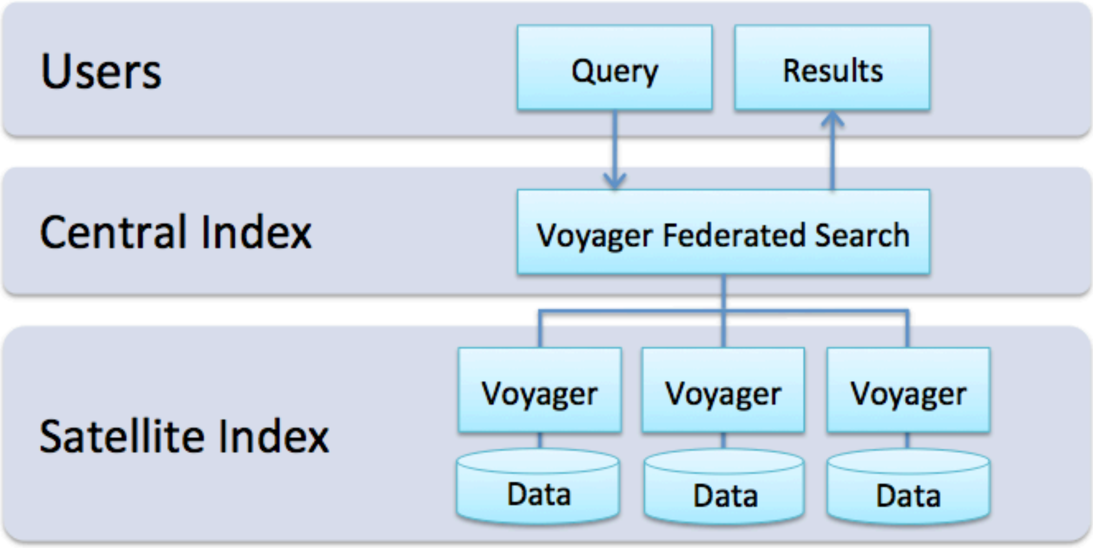
In this configuration, a replicated index can ensure high-availability in the wake of hardware configuration changes or failed hardware.

Load-balanced indexing will help your index scale to unlimited proportions. By dividing and distributing the index across multiple servers, the total number of records in each index will be smaller. The reduction in index size results in performance increases.

Voyager’s Federated Search

Voyager’s federated search links indices together to enrich results and simplify the user experience. Users can investigate multiple Voyager indexes with a single query, and gain access to more data, efficiently and accurately. By installing Voyager at each location and then linking the indexes together, an organization can achieve the one-search vision.

Architectural Diagram of Federated Search



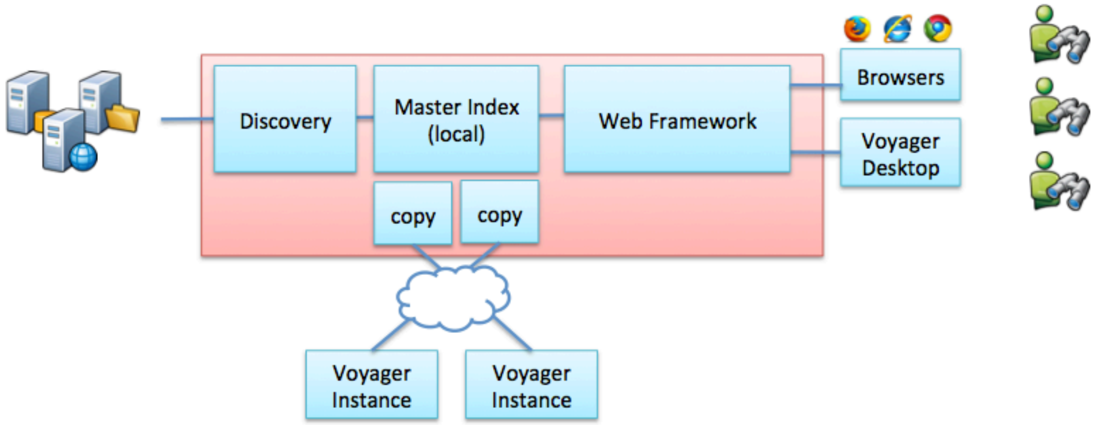
Voyager's federated search allows a large organization to search multiple Voyager indices simultaneously. A single request then provides unified, relevant results from multiple Voyager instances without storing all of the data in one place or even storing it in one way.

Voyager can be pointed to wherever the data exists. Once content is indexed, data owners can use Voyager to search remote data as easily as searching their own hard drive. Each Voyager instance can also be linked together to see other content all while honoring existing data dissemination and security infrastructure.

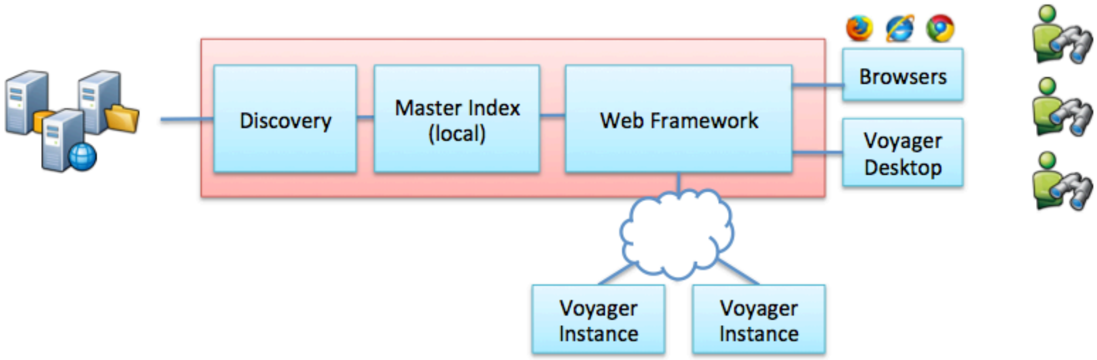
By acting as a catalog, rather than a storage system, content can stay wherever it is and existing systems will continue to work. This eliminates costly data migration and system re-engineering. At the same time Voyager can be used to find and promote authoritative content along with delivering processes to migrate data from tucked away systems.

Offline Environments and Federated Search

It is also possible to have Federated Search for different networks or offline environments.



In some cases one Voyager instance might not be able to talk to the other instance if they are not physically connected through a network, when firewalls exist or when the network performance is slow. In these cases you can copy the Voyager index where it can then be combined with the index of another voyager system.



Alternatively, a Voyager index can be configured for remote access. The user would send a request to the remote index or indices. The remote index would then return the results to the Voyager web application where they would then be collated with the results of other Voyager instances. This configuration is easy to manage and given that Voyager is talking to other Voyager instances, search result relevancy is properly handled.

System Sizing Recommendations

System sizing is an inexact science. With that in mind, here is what we recommend as the minimum requirements for most Voyager systems:

1. Hardware

- 8 CPU cores
- 10 GB memory
- 5 GB of disk space for the Voyager index (This must be on a very fast disk, such as where the OS is installed.)
- 20+ GB disk space for thumbnails (This can be stored on Network Attached Storage or other ancillary hard drives.)

2. Operating Systems

Any 32- or 64-bit edition Windows XP, Windows Vista, Windows 7 or Windows Server.

- 64-bit operating systems are recommended
- Virtual Machines are both supported and often recommended. Virtual Machine allow you to scale indexing resources (CPU cores) during initial indexing and then throttle them back later, when the system is in maintenance mode.

3. Additional Software

Voyager requires a license of ArcGIS Desktop versions 9.3.1, 10, 10.1, for users who wish to index Esri proprietary formats such as .MXD, .LYR, geodatabases, etc. Users who do not wish to index Esri proprietary formats do not require any additional software.

Additional Considerations

1. Voyager can leverage as many CPUs as you have available. Thus the more processor cores the greater number of indexing processes that we will be able to run simultaneously.
2. For average size indexes (100K-1M records), Voyager will need 2-3 GB of memory; for the web service, 1-2 GB per indexing process. The more processes the more memory needed; however, this is also dependent on the types of data that you are indexing (some formats require more memory).
3. Every additional block of 1 million records will require about 512 MB - 1 GB of additional system memory. This is not a linear equation. For systems indexing more than 30 million files, configuration recommendations will vary. (Please inquire at support@voyagergis.com for more information)
4. Access to data. Voyager should have quick access to the data that it is indexing. When the data are on a local network attached storage device or the local machine, this is usually not a problem, but if Voyager is indexing data on remote disks or across a slow network, for example, indexing speed could be impaired.
5. Thumbnail generation. Voyager generates thumbnails as it indexes files. Depending on image compression these files can range from 30 KB to 120 KB in size. Voyager's Discovery Settings allow users to pre-generate thumbnails or generate them on the fly.

   By default, Voyager will to generate thumbnails on the fly so that they are created when the user requests the data, and thumbnails are not created for data that is never called. But if this proves to slow down searching, then users may choose to pre-generated thumbnails for all files. Keep in mind that this might require adding additional store capacity to for storing the thumbnails. We recommend that thumbnails be stored on cheaper, ancillary, storage. See Voyager.vmoptions file for more configuration details.