

# White Paper

## Federated Search Using ELISE ID

Leveraging Existing Repositories



elise

# Federated Search Using ELISE ID

## Introduction

In an ideal world, search would mean looking in a single large data repository to find what you are after. In reality, data is often held by different organizations using different software with different data structures and IT architectures, and often with different access rights. Searching across these separate data repositories presents some special challenges and requires special solutions.

In a typical solution, a federated search is performed in which several databases are searched simultaneously, but data from the separate repositories is kept separate. This allows a local organization to control the content, structure and access, while enabling other organizations to search through the local data. An example of this approach would be several local police departments that want to control their own data, but want to collaborate to allow a search across the total body of data held by all police departments.

ELISE ID not only supports the federated search model, it also meets the special challenges of federated search that stem from disparate database software, disparate data structures, and disparate access control requirements. This whitepaper discusses several of these challenges and how ELISE ID can help. First, the typical process flow for a federated search is laid out. Second, a federated search solution based on ELISE ID is described, complete with some of the fundamental architecture options. Third, the challenge of data security and some of the methods of addressing this concern are explained. Fourth, data replication, which helps normalize both data structures and search methodologies, is described. Finally, a section on scalability and reliability shows how ELISE ID

provides infrastructure for these key attributes of any large-scale search system.

## Typical Federated Search Process

In the typical process of a federated search, a central application issues a search query to several local databases. None of the local databases needs to merge its information with other databases since each local database is searched separately. The results are then returned to the central application, which uses some logic or rules to combine them and report them back to the user. This is most effective when each local database has the same structure as the others and uses the same search methodology.

**In a typical solution, a federated search is performed in which several databases are searched simultaneously, but data from the separate repositories is kept separate.**

More likely, however, each local database will have its own data structure and search methodology, so combining the results can be difficult and complex. When control over data access is required, protocols for access rights must be negotiated and implemented at each database to make certain that data is returned only to users that have rights to view it.

## Federated Search with ELISE ID

Though ELISE ID follows much of the typical process for federated search, it offers several significant enhancements to the process.

## Federated Search Using ELISE ID

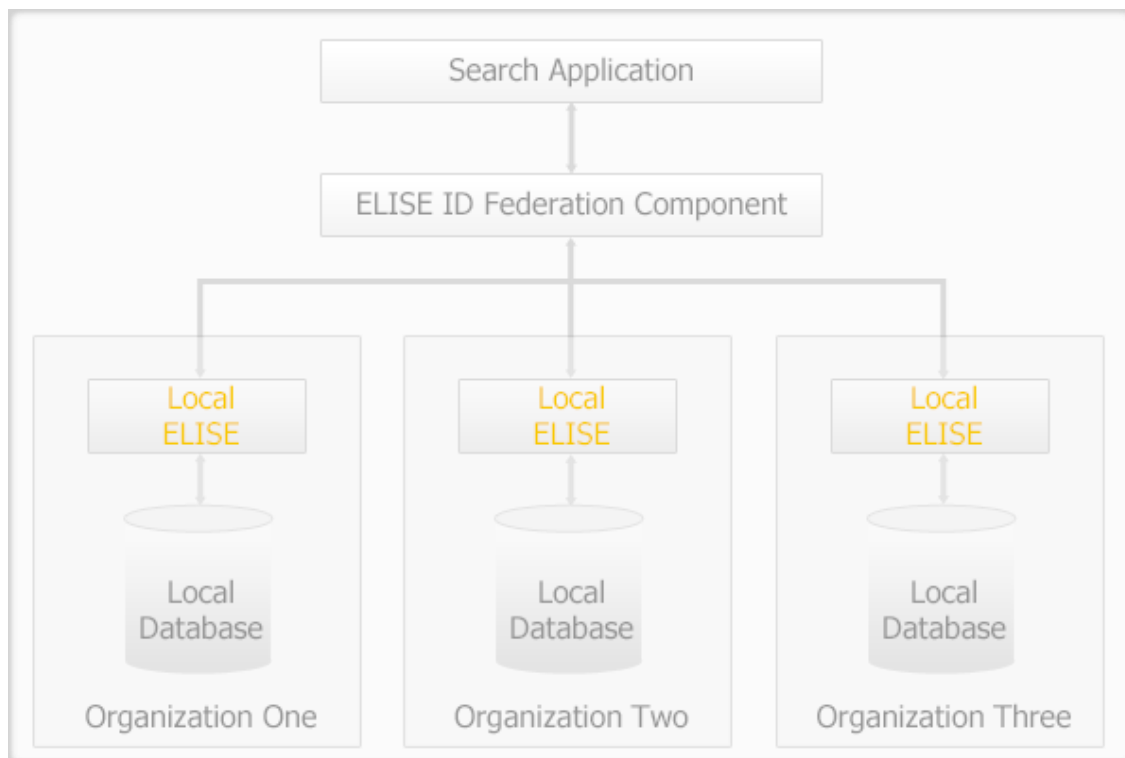


Figure 1 - Architecture of a Federated Search Solution Using ELISE ID

The architecture diagram above (Figure 1) shows an overview of the architecture of a federated search solution using ELISE ID. It shows three databases that are controlled locally and made available for search through the ELISE ID Federation component.

With ELISE ID, a Federated Search Component accepts the search query from the central application and distributes it to several instances of ELISE ID, one at each local database.

Regardless of the local database software used or the structure of the data held, each local instance of ELISE ID provides a unified search methodology that can be optimized for the actual data in the repository. This makes combining the results a more straightforward process and provides results that are more consistent.

Since ELISE ID is a Smart Search & Match system that uses match scoring, data security can be preserved by providing search results that contain only the scores and the unique identifiers of the records found. This makes it possible to separate the security protocols from the search or the data, thereby simplifying the control of data access within the application layer.

**The ELISE ID Federated Search Component combines scores according to customizable fusion algorithms and returns results that are above the customizable threshold to the central application.**

# Federated Search Using ELISE ID

In step-by-step form, here is how the process looks:

1. Central application issues query to ELISE ID Federated Search Component
2. ELISE ID Federated Search Component issues a query to multiple local databases
3. Local search done by ELISE ID on the local databases
4. Scores and unique identifiers are returned to the ELISE ID Federated Search Component
5. ELISE ID Federated Search Component combines scores according to customizable fusion algorithms and returns those above the customizable threshold to the central application.
6. Protocols for obtaining details on the matches that meet certain criteria are followed in the application layer to provide detailed results

To really understand how ELISE ID enhances federated search, let's take a closer look at each of the challenges and the ELISE ID answer.

## Implications of Data Structure

### Using a Common Data Schema

In reality, it is not likely that the different local databases will share a common schema. To address this issue, a unified schema for ELISE ID can be used. In this approach, all local ELISE ID databases will share a common data schema, and the data from all local databases will be mapped onto this schema. For example if one of the local databases has a property for eye color and another database does not have that property, it could be decided to include the eye color property in the unified data schema for ELISE ID.

ELISE ID provides a specialized component for synchronization with the local reference

database, so more details on the mapping process are provided below in the section "Data Synchronization".

The benefit of this approach is that the matching behavior can be specified centrally rather than having to define it for every local ELISE ID instance. This approach also makes use of ELISE ID's native capability for handling missing and unknown data.



Figure 2 – The Process of Matching

# Federated Search Using ELISE ID

When a user performs a query from his search application, the query is sent from the federation component to all systems that are part of the federation. Each member of the federation calculates a top N list on the data that it contains and sends that back to the federation component where it is combined to the overall result. ELISE ID's built-in functionality for handling unknown or missing data makes sure that in case a certain property is not present in one of the member systems, it will be handled properly.

## Using Local Data Schemas

An alternative approach is to use individual, local schemas for each ELISE ID instance that closely parallels the schema of the local database. In this approach, the queries would be mapped by the federation component onto the schemas of the local ELISE ID instances.

When using individual data schemas, the process of performing a match follows the general pattern described previously, but two additional steps are required:

First, prior to executing the query on the local ELISE ID system, it would be converted to the format of the local schema. Second, the percentages that are sent back to the federation component may be adjusted so they can be correctly compared with percentages from other systems. For instance, if the original query contained a field such as fingerprint next to person name and address, and the local system does not have fingerprint information, an 80% match from a non-fingerprint database would be rescaled to rank it correctly against an 80% score from a database that does have fingerprint information.

Of course, whether a common schema or local schemas are used, the local instance of ELISE ID could be used independently for local search, in addition to supporting the federated search.

## Data Access Control

In many cases it is necessary to restrict data access based on the role of the user that is performing the search. For example, personnel with a lower security clearance may not be allowed to know that a certain local database contains a hit on a particular person's name. However, it is not likely that the local databases share the same access control mechanism or have a common view of all possible users and their associated privileges. This can make integrating with the local access control mechanism challenging and typically calls for solution specific integration.

In general ELISE ID can be configured in such a way that it only returns a controlled amount of information on the search results, for instance, it can return only the key (unique identifier) of the record and a match score. This allows the federation component to merge the different result sets, while protecting the privacy of the records in the database. If more details about a record from the result set are needed, the search application can use the key and separate access control protocols to control retrieval the appropriate information..

## Methods

There are two well accepted approaches within the search and match industry for data access control both of which can be implemented with ELISE ID: early and late binding, which are detailed below.

# Federated Search Using ELISE ID

### Early binding of security credentials

When using early binding, the ELISE ID search engine stores a token that specifies which users are allowed to access an element of data with that element. As part of the match request, the security token of the user making the request is sent into the local ELISE IDs, which compares the user's token to the tokens in its database to only return those documents that the user has access to. This has the advantage that the engine only returns results that the user is actually allowed to see.

### Late binding of security credentials

When using late binding, there are no security tokens stored in the ELISE ID database. Rather,

for every element that is found to be relevant for a given query, a callout is made to a separate system containing access control information to determine whether the user performing the query is actually allowed to access that particular element. The advantage is that security information does not have to be made available to ELISE ID, but the drawback is that potentially a lot of elements will have to be verified before a top 100 list can be composed.

When using late binding, there is the option of performing the check either at the level of the local databases, or at the level of the federation component.

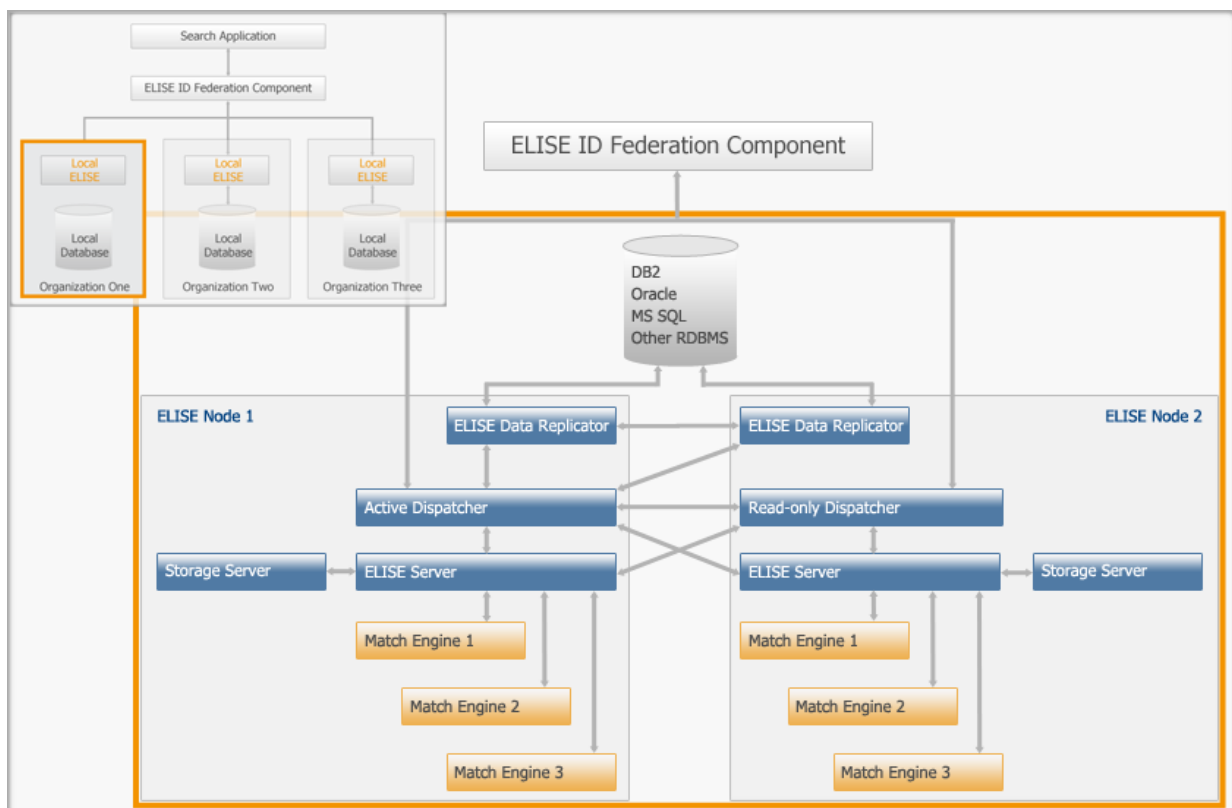


Figure 3 - Data Synchronization Within an Organization

# Federated Search Using ELISE ID

## Physical security

Another critical element in the federated search architecture is the physical security of the overall system. As connections have to be established between networks of different organizations, potentially over public networks, controlling who can connect to which machine is key in keeping information safe. Using encrypted communication channels and tightly configured firewalls, this can be achieved. While a full description of how to set up such a system is outside the scope of this whitepaper, please contact WCC to learn how some of our customers have set up their multi-datacenter environments.

## Data Synchronization

As most search systems, the data that should be searched needs to be ingested in ELISE ID and remain synchronized afterwards. In a federated environment, the data synchronization is limited to the local agency level only. Within this agency, a regular mapping from the RDBMS (or any other data source) to ELISE ID is created using our standard component ELISE Data Replicator (EDR). This mapping contains the specification of which data from the source database should be made available in ELISE ID for searching. After all, not necessarily all data from the source

database needs to be available in ELISE ID. EDR ensures that the content of the source database and ELISE ID remain synchronized. Typically this is done by first instructing EDR to do perform a “full load” which loads all data from the source database into ELISE ID. After the initial full load, EDR is instructed to synchronize any changes in the sources database with ELISE ID. This includes all inserts, updates and deletes.

## Scalability and Failover

The scalability, high availability, redundant architecture and use of industry standard servers make ELISE ID ideal for mission critical deployments in a federated architecture.

ELISE ID’s high availability/redundant architecture allows multiple installations in physically separate locations on different network

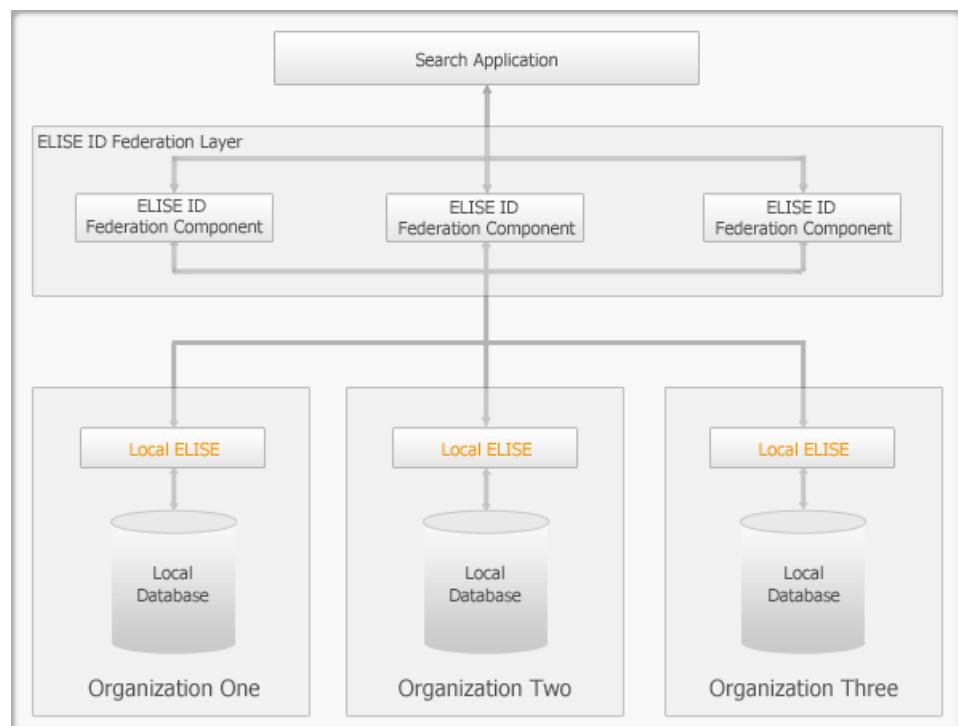


Figure 4 - Failover of ELISE ID Federation Component

## Federated Search Using ELISE ID

infrastructure to prevent downtime due to failure of hardware or network infrastructure. It has been designed in such a way that there is no single point of failure in the architecture. ELISE ID scales easily by simply adding favorable price/performance COTS servers to accommodate larger data volumes, increased loads and match speed performance considerations.

The logical architecture that was presented above in Figure 1 can be expanded to show how the ELISE ID architecture accomplishes this at a more detailed level. The diagram above zooms in

on the architecture of the Federation component.

From this architecture diagram, you can see that the ELISE ID Federation component can be installed on multiple machines, offering load balancing and failover. When a user wants to perform a search, the search application sends the search request to one of the Federation components, which distributes the request to the local ELISE ID databases and consolidates the results. Search requests are distributed evenly across the multiple components, offering load balancing if needed. In case one Federated search component is temporarily not available,

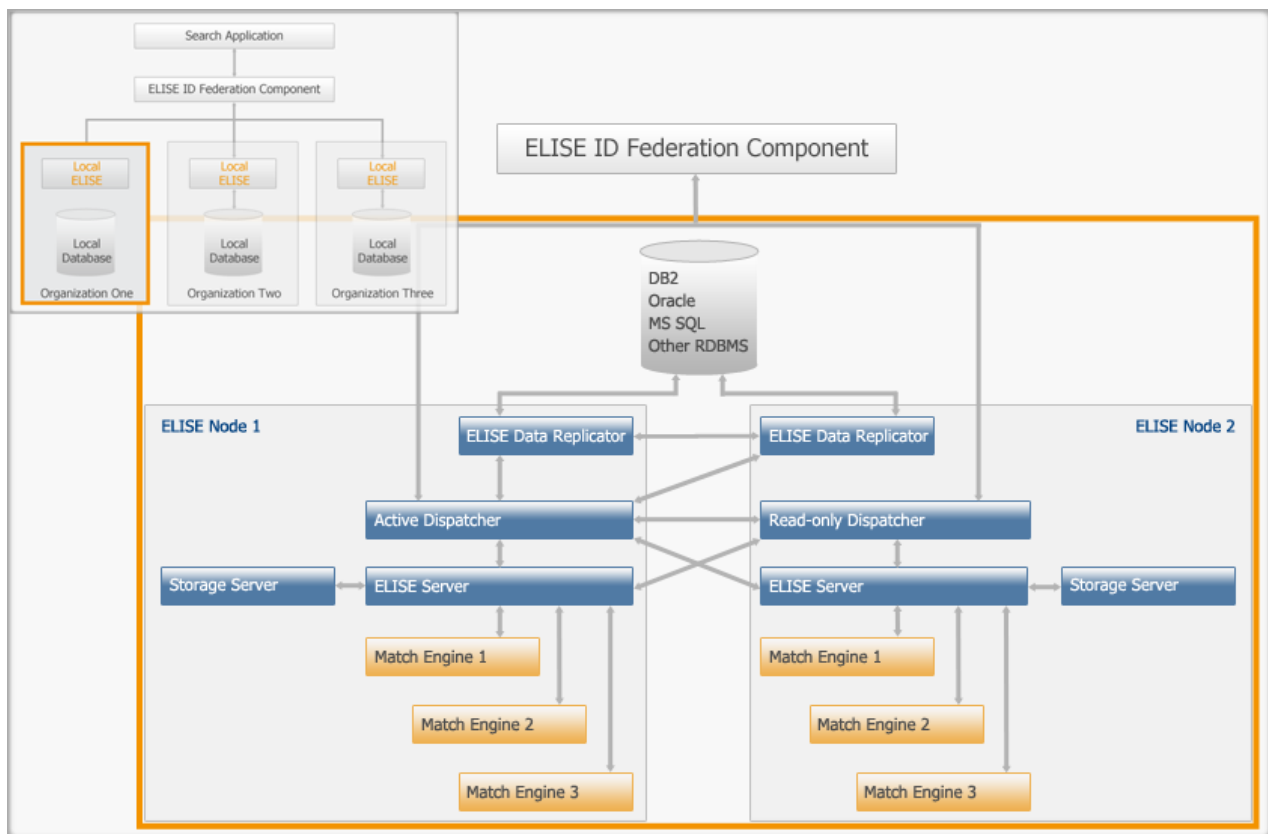


Figure 5 - The ELISE ID Architecture Diagram

## Federated Search Using ELISE ID

the application automatically switches to a component that is still available, and switches back the moment the component is on line again.

In Figure 5 you find the ELISE ID architecture diagram. This is the architecture that would be used in each local setup.

From this architecture diagram, you can see that the ELISE ID architecture has been designed with scalability and high availability in mind. With regard to performance, ELISE ID is scalable as follows:

- ❖ Greater capacity (in terms of matches per second) is attained by adding additional nodes to the ELISE ID cluster
- ❖ Shorter response times are attained by adding additional servers with MatchEngines to individual nodes
- ❖ Higher data volume (more items in the database) is attained by adding more MatchEngines and/or MatchEngine memory, and/or more disk space for the StorageServer

The nodes of the ELISE ID cluster are automatically kept synchronized by the Dispatcher component. In case a node in the cluster is temporarily off line because of for instance hardware maintenance, the Dispatcher will automatically resynchronize the node using the data available in other nodes.

### About WCC and ELISE ID

Founded in 1996, WCC Smart Search & Match specializes in the development of search and match software for identification. Its flagship product, ELISE ID, delivers fast and accurate identification using multiple biometrics and/or

biographic criteria. ELISE ID is used by government agencies for immigration, border security and customs control.

ELISE ID, the proven flagship product of WCC Smart Search and Match for large-scale identification, delivers:

- ❖ True score level fusion of biometric, biographic, and contextual identifying data, using unlimited criteria
- ❖ Ultra-high speed in searching through large amounts of structured and unstructured data
- ❖ High accuracy with large number of criteria by fusing individual match scores
- ❖ Analytics to understand match scores and results ranking
- ❖ Powerful, real-time matching capabilities for finding similar words, names, concepts, numbers, with complete control over rules, weights, value ranges, and other factors
- ❖ Transparent, explainable, repeatable results for adjudication, query tuning, system tuning
- ❖ Extendable architecture to include any third party matching algorithm (e.g. graphics matchers, alternative phonetic matchers, biometrics, etc.)
- ❖ Prevents technology lock-in and obsolescence by allowing the easy integration of future biometric and fusion algorithms
- ❖ Scalability to handle multiple, very large disparate databases in either consolidated or federated fashion

WCC Services B.V.  
Savannahweg 17  
Utrecht ■ 3542 AW ■ NL  
Tel. +31 30 7503200  
Fax. +31 30 7503200  
info@wcc-group.com  
[www.wcc-group.com](http://www.wcc-group.com)

Washington DC Office:  
Two Fountain Square  
Reston Town Center  
11921 Freedom Drive ■ Suite 550  
Reston ■ VA 20190  
Tel. +1 703-904-4320  
info@wcc-group.com  
[www.multimodalfusion.com](http://www.multimodalfusion.com)

*WCC is a market leader in developing and licensing high-performance search & match software helping businesses / governments and their customers find people, products or services that best fit their requirements in sub-second response time. WCC's technology is used by many of the largest staffing firms, job boards and Departments of Labor in the world. Its flagship product ELISE is also utilized in the security and content management industries.*