

# How Security Binding Choices Impact Everything in Enterprise Search

Bryan Pham, CTO  
Cloudtenna  
July 25, 2018

## Executive Summary

Few non-software architects understand the outsized impact security binding has on enterprise search performance, scalability, multi-tenancy, hardware, supporting infrastructure, capital expenditures (CapEx), operating expenditures (OpEx), and the total cost of ownership (TCO). The wrong security binding choice can add hundreds of thousands to millions USD to the TCO. From additional expensive hardware, supporting infrastructure, maintenance, software licensing, training, power, cooling, shelf space, rack space, cables, conduit, transceivers, and allocated overhead, the costs can be shockingly high.

This document examines the pros, cons, tradeoffs, consequences, and workarounds, for each of three different security binding choices – **late binding, early binding, and real-time binding**. What it finds is that Cloudtenna DirectSearch real-time binding provides the query-time performance of early binding with the security accuracy of late binding.

Real-time binding is up to 100x faster in reflecting proper security compared to early-binding and returns user queries up to 10x faster than late (query-time) binding for large datasets. This document outlines why perturbation performance and fast query-times are crucial for multi-silo enterprise search.

## Introduction

Effective Enterprise search software has to deliver on several fronts if users are to have a satisfactory experience. The fundamental challenges Enterprise search must resolve include:

- Providing search results sub-second<sup>1</sup> (an absolute must for human productivity based on IBM's "Economic Value of Rapid Response Time");
- Scaling the search engine index from millions of files (single tenant) to billions of files (multi-tenant) providing the ability to search keywords and phrases without compromising performance;
- Scaling to thousands of users without reducing performance;
- Ensuring results are accurate and relevant to the query;
- Safeguarding security by delivering only query results that the user is authorized to see based on access control lists (ACL) a.k.a. user permissions.

---

<sup>1</sup> Per IBM's "Economic Value of Rapid Response Times" <https://jlelliotton.blogspot.com/p/the-economic-value-of-rapid-response.html>, user productivity increases substantially when response times are sub-second. User attention and productivity drops precipitously when response times climb above 1 second and fall off a cliff when they equal or exceed 3 seconds. It ties to the human short-term memory buffer. When response times move to 1 second or more, the buffer empties and humans must retrace their steps. The differences between 3 seconds and .3 seconds response times is more than 2x in productivity.


The last of the itemized challenges is a function of security binding. The security binding a.k.a. access control, correlates queries to ACLs or user permissions. Security binding filters query results based on permissions/access rights before they're returned to the user. Files and permissions are stored in separate indexes. Correlating the two requires a join operation. When and how that security binding operation occurs directly effects scalability, accuracy, performance, and user satisfaction. Architectural choices are highly consequential. This is acutely accurate regarding the ways an Enterprise Search product implements security binding.

This document explores the pros and cons of the security binding architectural choices, tradeoffs, and consequences.

## Security Binding Choices, Pros, Cons, Tradeoffs, and Consequences

The following are the 3 security binding choices:

- **Late (query-time) binding:** Content and file permissions are reconciled at time of user search query.
- **Early binding:** File permissions are preprocessed on a schedule, typically 24 hours.
- **Real-time binding:** File permissions are preprocessed immediately upon recognized attribute change.

	<i>For large datasets</i>	
	Sub-second search queries	File permissions updated in near real-time
 Real-time binding	✓	✓
Query-time binding	!	✓
Early binding	✓	!

### Late Binding – Sometimes called “on-demand binding”

In late (query-time) binding, user permissions are calculated and bound to the files at the time of query. Search results are by default permissions-accurate because the security binding takes place at the time of query. It doesn't matter if user permissions or a property change.

But security binding is CPU and memory intensive because it is a join operation that can span potentially billions of rows in two separate indexes. This can take a substantial amount of time. Query response times can be lengthy, especially as the number of files and/or users scale. Query-time binding attempts to filter on the results rather than on shards and then calculates visibility and, in some rare case, it can cause extreme delay. Queries that are supposed to be sub-second are commonly several seconds to dozens of seconds long. They are known to stretch into minutes when the either the number of files exceeds a billion or the number of users grows quite large. As per the IBM report on the “Economic Value of Rapid Response Times,” these response times are unacceptable.

The rigid limitations for database joins required of late binding enterprise search makes multi-tenancy problematic at best and generally impracticable. Database efficient joins regularly max out at tens of millions of rows for a specified period of time. For a single tenant with limited scalability, it is generally permissible. But for multi-tenants its scale limitations makes performance unacceptable.

There are a few workarounds to late binding response time penalty. The first is to utilize in-memory caching. Caching speeds up response times significantly for common and repeated queries by not having to repeat the binding process. But caching in DRAM is quite costly. To match performance, an oversized infrastructure stack must be maintained at all times (most idling) because query-time-join is paid on (done for) every request. Then there's the issue of cache coherency. As the Enterprise Search application scales, it needs more nodes. Each node has its own cache. The cache of each node must be coordinated and synchronized. Multi-node cache coherency is exceedingly difficult and cumbersome. This why many late binding Enterprises are limited in the number of files and users they can scale.

The workaround to scale and database join limitations is to implement multiple Enterprise search instances and then assign each group or department its own instance. This workaround provides better response times within a group or department but has severe limitations and issues. Each enterprise search instance is its own search silo. This means for large multi-tenant organizations, there will be no global search engine that encompasses all users and files. It means users must utilize multiple tools to search multiples repositories. Security binding for each user will be duplicated in each enterprise search instance where they have access. Management becomes more complicated. Efficiency drops, storage consumption increases, and costs spiral. CapEx increases geometrically (exponentially) as software inefficiencies demand faster and more hardware in addition to supporting infrastructure purchases. OpEx in hardware maintenance, power, cooling, and software instance license costs (subscription or perpetual plus maintenance) also becomes untenable even if large discounts are applied.

Late binding is generally considered the most security accurate, but slowest and costliest security binding methodology.

### Early Binding

In early binding, user permissions are calculated and bound to the files at the time of indexing as part of the ingress pipeline. Unlike late-binding which must pay on every request, early-binding is a one-time cost. When a user queries a file, the security has already been determined. This makes results much faster than late binding and enables sub-second response times. As previously mentioned, security binding is CPU- and memory-intensive and takes quite a bit of time. By scheduling the binding during low use off hours, the binding process does not affect other operations or slow down queries.

The downside to early binding is inaccuracy. When a permission or property change occurs (commonly known as a perturbation,) the permissions will be inaccurate until the next scheduled binding. If that schedule is once a day or once a week, the security is no longer accurate for that period of time. A good example of this would be when an individual changes groups or departments. Their file access permissions will change. But the files they should be able to see in their new function will be inaccessible. The files they should no longer be able to see will still show up in the search results. These security violations will not change until the next scheduled early binding process aggravating the user and hamstringing their productivity. This is likely to be unacceptable to most enterprises.

The workaround to schedule security bindings more frequently is typically not sustainable. It has the undesirable effect of noticeably slowing other applications during binding causing a cascade of consequences. Slower applications cause users and customers to complain. This sets off a series of actions that usually lead to the purchase of more expensive but faster server hardware, memory, and storage. In turn that new hardware needs more or upgraded supporting infrastructure – networks, cables, transceivers, rack space, conduit, allocated data center overhead, etc. This represents a major CapEx investment. It's also a major OpEx investment in for the new more

powerful hardware and infrastructure. OpEx in the form of more maintenance, more power and cooling, additional management training, more real estate consumed, etc. The hardware increase costs are not static. As the file numbers grow so do the security binding demands placing ongoing pressure. This cost pressure makes the workarounds unsustainable causing a relapse of either lengthier security violations or slower applications. Neither is an acceptable outcome.

Early binding is typically has the fast query performance, but is also a costly and commonly inaccurate security binding methodology.

### Real-time Binding

Late and Early security binding have been around for a quite some time. Their shortcomings are profound for the enterprise search use-case due to the complexity required to return personalized results that accurately reflect individual file access permissions. This is why Cloudfenna is seeking a new way to accomplish Security Binding. One that provides the performance of early binding and the security accuracy of late binding without the consequential penalties of either. This is what Cloudfenna calls “real-time binding.”

What makes Cloudfenna’s DirectSearch binding “real-time” is how it handles perturbations in file ACLs, user properties, group access, or SaaS application extended visibility properties. DirectSearch does not wait for a scheduled binding event. It binds when the change occurs. A file ACL change or SaaS application change is reflected in the binding within 15 to 20 seconds, e.g. real-time. A change in user properties or group access takes a little longer, but still in real time when compared to the early binding scheduled window of 24 hours or more.

Cloudfenna DirectSearch security binding real-time capabilities is the direct result of innovative in-memory joins instead of drive-based joins utilizing the Apache Spark fast data platform, as opposed to on-media joins of a relational database system or big data (Hadoop) approach. This methodology is extremely efficient and as a result much less CPU intensive. That translates into vastly reduced hardware and supporting infrastructure requirements than early or on-demand binding Enterprise Search products. Reduced hardware means much lower CapEx and OpEx. When compared to late binding, Cloudfenna DirectSearch is up to 100 times faster. It provides the best of both worlds.

This unique Real-time binding is currently only available from Cloudfenna DirectSearch.

## Conclusion

For users to experience satisfactory Enterprise search that search must utilize a security binding architecture that empowers:

- Sub-second query performance;
- Scalability into billions of files and thousands of users without user noticeable performance degradation;
- Accurate and relevant query results;
- Real-time accurate security.

Today that means Cloudfenna DirectSearch.

For more detailed information about Cloudfenna DirectSearch please go to:

- Website: [www.cloudfenna.com](http://www.cloudfenna.com)
- Email: [contact@cloudfenna.com](mailto:contact@cloudfenna.com)
- Phone: (800) 298-5215