Toward an Identity & Context Virtualization Service:

A Buyer's Guide to Virtual Directories

A blueprint for how to evaluate virtual directories, from use cases and technical parameters to the five questions you should ask yourself before choosing a virtualization solution.

A **Radiant Logic** White Paper

Contents

Introduction....3
New Ways to Manage Identity and Context....3
From Push to Pull: The Challenges of Building an Identity Infrastructure Within a Changing IT Landscape....3
Out of the Silos: Identity and Context as a Service....4
Section 1: What Virtualization Lets You Do....5
Identity & Context Virtualization by Use Case:....5
Level 1: Proxying to Multiple Directories....6
Level 2: Remapping Non-Standard Identity Sources to Standard LDAP....6
Level 3: Aggregating Larger Data Sources....6
Level 4: Enriching Profiles with Context-Driven Attributes....7
Level 5: Creating a Common Identity and Context Service....8
Section 2: Why Speed is Essential and so are Volume and Richness....9
Volume Matters in Authentication....10
The Limitations of Proxy....11
Richness Matters for Authorization and Profile Management....11
Join is the Key to Richness—But Joining Across Systems is Complex and Expensive....12
Section 3: How Different Virtualization Solutions Work....13
Five Questions to Ask Yourself When Considering a Virtual Directory Solution:....13
1. How should my information be represented?....14
2. Is my data clean, and how important is that to me?....14
3. What kind of performance do I need?....14
4. How do I want to propagate information?....14
5. Where's my infrastructure headed?....14
Conclusion: Different Solutions for Different Needs....15
Why You Need Identity and Context Virtualization....15
About the RadiantOne Identity and Context Virtualization Platform....16

Introduction

The world of IT and security is changing. We're moving from highly centralized infrastructures to a more federated approach, where relationships are loosely coupled, identities are scattered across systems, and we all live and work in the cloud. But such a move has exposed weaknesses in our IT systems—and prevented us from taking full advantage of all these technological advances. **The struggle between the push model of our traditional security infrastructure and the pull model of modern architectures demands new ways of thinking.**

As Bob Blakley of The Burton Group explains in his recent paper, *"The Emerging Architecture of Identity Management,"* the foundation for such a "pull-based" model already exists:

1. *"In the first phase,* ***production of identities will be separated from consumption of identities through the introduction of a virtual directory interface."***
2. *"In the second phase,* ***applications will externalize authorization to policy decision points****…which can* ***use contextual authorization to request attributes in real time."***

New Ways to Manage Identity and Context

These ideas tap into something we've long understood: that **virtualization has the power to change how we manage identities in the enterprise**, and better yet, to free the relationships that are trapped in our application silos, enabling a world of new context-driven services. Such infrastructure shifts are necessarily incremental—no company can afford to make a wholesale change to its architecture. We also understand that virtual directory technology will be a main driver of such progressive changes, enabling an evolutionary approach while salvaging as much of the existing infrastructure as possible. **But not all virtual directory solutions are created equal—and we think it's imperative to select the right tool for this job.**

This paper is **a practical guide to choosing the best virtual directory solution to meet your needs**, both now and in the future. Before we dive into the details of virtual directory technology, it's important to step back and look at the problems such technology is designed to solve. We'll begin by exploring how identity and context are siloed across the enterprise, and see how different virtualization strategies address these issues. Then we'll examine the common use cases for virtual directories, the essential parameters to use in evaluating them, and the questions you should address before purchasing an identity virtualization solution.

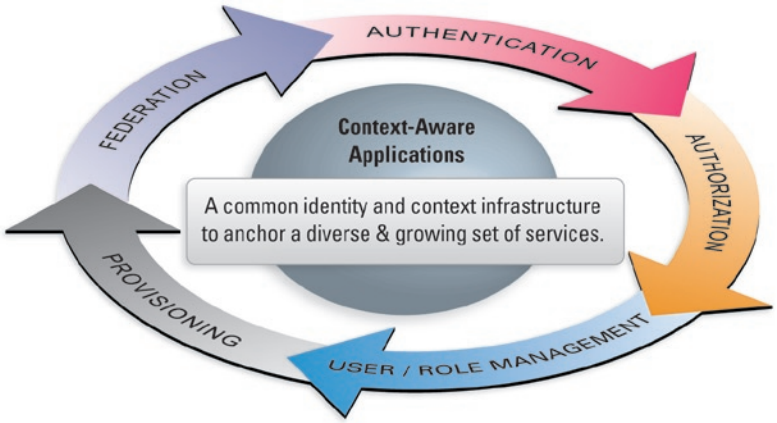
From Push to Pull: The Challenges of Building an Identity Infrastructure Within a Changing Landscape

Enterprises today have heterogeneous environments of legacy applications and increasingly distributed data silos, and they serve diverse user populations both inside and outside the firewall. Amid all this complexity, they must also respond to new demands and opportunities, ranging from mergers and acquisitions to the rise of the cloud. **The first challenge within such an evolving environment is to integrate identity and context for security and privacy.**

The requirement is multiform, going from a GRC focus on managing employee access behind the firewall to extending and controlling access to customers, vendors, and other stakeholders beyond the firewall. As enterprises have added new audiences and acquired new companies, identities can no longer be stored in a centralized, homogeneous location. Now they're managed across distributed and heterogeneous data silos. **What used to work—centralized authentication and authorization—no longer serves today's far-flung, federated, and even cloud-based infrastructures.**

Out of the Silos: Identity and Context as a Service

As the Burton report highlights, identities and security contexts are embedded in a series of different applications and silos, making it difficult to manage overall your identity and security. This issue began with the creation of directories themselves and while there's been evolution in how we handle our silos, this remains a systemic problem at the infrastructure level. In hindsight, **such data should be externalized into a service outside any application.** This is the foundation for an identity service based on virtualization, where, as Burton says: *"…[the] production of identities will be separated from consumption of identities through the introduction of a virtual directory interface."*



In today's enterprise, you must provide a set of security services to many applications, as well as begin supporting increasingly more personalized, context-aware services.

Although the initial challenge is identity management, **the rise of the silos—and the difficulty of integrating across them—covers many aspects of your business information**. As enterprise IT has matured, more of your business-critical data is being managed in specialized application silos. For the most part, this is a good thing: such applications are expert at what they do, making it easier to accomplish specialized functions, such as CRM, HR, or billing. But just as technology helps us do more, it can also hold us back. While silos are excellent at collecting, maintaining and organizing information on a local level, integrating data across them is a huge technical challenge.

The need to integrate application contexts across silos is the foundation for a context service, where ***"…applications will externalize authorization to policy decision points****…which can* ***use contextual authorization to request attributes in real time."*** Such an identity and context service needs to serve many different applications, with different expectations and interfaces and hence support different access protocols.

The virtual directory began life as a lightweight, flexible point solution aimed at solving part of this problem by bridging user directories. The traditional virtual directory does this job well, but it is part of a continuum of identity and context virtualization solutions that have evolved to meet new demands and increasingly sophisticated use cases. Such solutions treat identity and context as a service, enabling enterprises to **get a complete global view of their information**, which is essential for analyzing data and business intelligence, targeting upsell and cross-sell opportunities, and improving operational efficiency.

Section 1: What Virtualization Lets You Do

When considering a virtual directory solution, it's important to understand your requirements because there are several products in the category, each representing different approaches to virtualization and addressing different needs at different price points. **The more you clarify your requirements, the better you can optimize your investment** and ensure that it scales along with future needs.

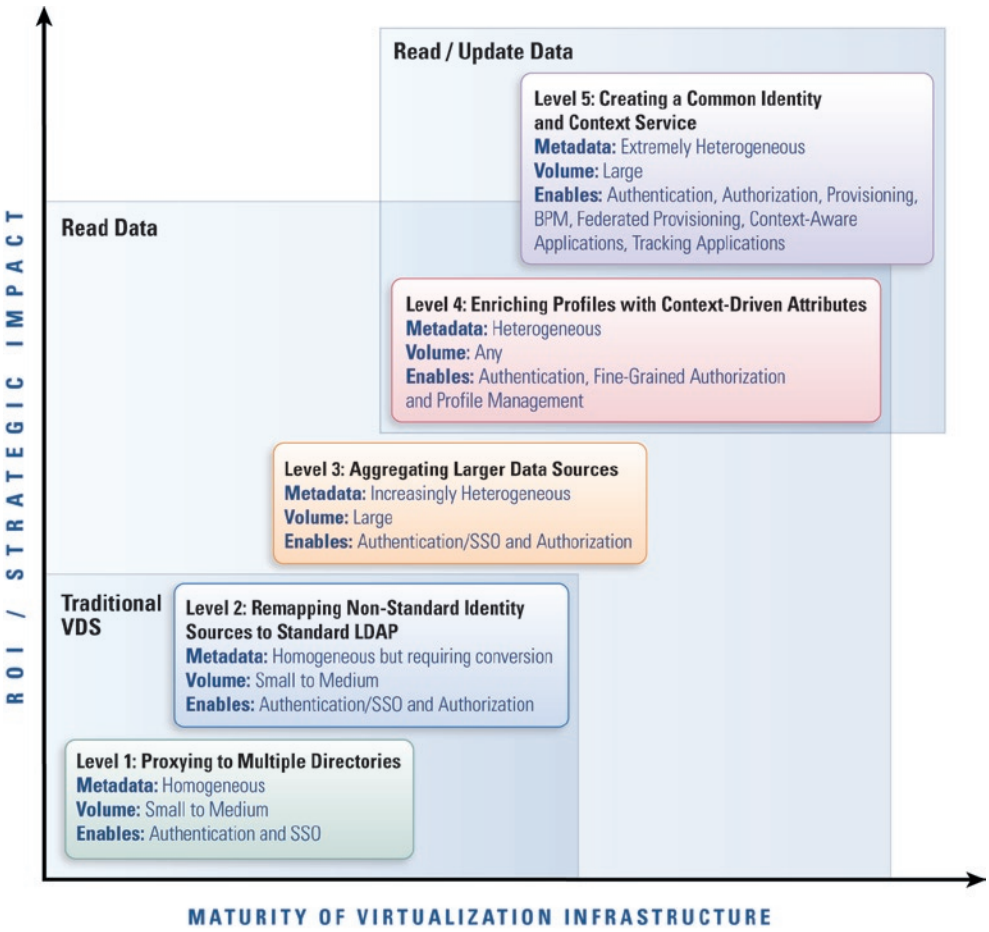
For instance, if you require simple directory aggregation within a relatively homogeneous environment, and you know your integration needs will not grow over time, a classical virtual directory solution should serve your needs. But **if you have a heterogeneous high-volume environment, you'll need a more robust virtualization solution;** one that can scale across diverse data sources, millions of users, and increasing demand.

Identity and Context Virtualization by Use Case

A virtual directory is only as valuable as what it allows you to do. Again, it's important to consider your needs. The complexity of use cases—along with the value of your solution—rises along with the diversity of your data sources and the number of users you serve. Authentication is the basic use case for a virtual directory, but new model-driven approaches have enabled a world of fine-grained authorization, context-driven profile management, and other personalized services.

We've developed five evolving levels of virtualization, designed to address the spectrum of solutions covered by identity and context virtualization technology. While most virtual directories offer only limited directory aggregation and mapping capabilities, there are many more levels to a true virtualization solution. The first two levels are addressed by mapping and proxy virtualization and the other three are addressed by model-driven virtualization, plus infrastructure support to ensure scalability, performance, and stability.

Value rises along with complexity of the use case and maturity of the IdM infrastructure.



Level 1: Proxying to Multiple Directories

Metadata: Homogeneous

Volume: Small to Medium

Enables: Authentication, SSO

Let's look at where virtual directories began, as a tactical point solution to enable authentication and SSO. Our starting point is the need to search for an identity across different data sources. The first real application for this "federated directory" is in the identification phase of the authentication process, where you're looking to see if a given log-on or identifier exists. Because we're in the security space, and security requires speed, you'll find this information in a directory. But what happens if there's more than one directory?

- You either build a **global directory** where you centralize the contents of multiple directories, which is the most robust and scalable solution, but which leads to complex maintenance and synchronization.
- Or else you've got to search each directory in turn, even if they're built on the same schemas. The **virtual directory-as-a-proxy** was developed to address this, quickly scanning each datasource and translating where needed to create a single logical view of disparate sources, enabling the federation of queries where calls are routed to the correct distributed silos.

While you may lose a tiny bit of speed with a virtual directory, you don't have the same synchronization issues you do with a metadirectory, and speed is not an issue, because your underlying data sources tend to be fast, especially when you're dealing with small volumes. **This kind of solution appeared first in the LDAP proxy category and was a precursor to the traditional virtual directory we see in level 2.**

Level 2: Remapping Non-Standard Identity Sources to Standard LDAP

Metadata: Homogeneous but requiring conversion

Volume: Small to Medium

Enables: Authentication, SSO, Authorization

At the most basic level, security applications such as your Web Access Management solution generally expect identity to be stored in a directory. So **the first requirement is to remap something that is not a directory into something that looks like a directory**, so the application can consume the data within. At the simplest level, this can involve converting a propriety directory, such as Active Directory, into a standard LDAP directory, such as Sun.

This level can also involve mapping a **SQL database** to an **LDAP directory.** At this level, virtualization is a lightweight approach that's great for small volumes, but cannot scale to large databases containing millions of identities. **This proxy level refers to the traditional virtual directory and is the category of most virtual directory products on the market.**

Level 3: Aggregating Larger Data Sources

Metadata: Increasingly Heterogeneous

Volume: Large

Enables: Authentication, SSO, Authorization

But what if your directories are larger—what happens then? While directories scale well, not all directories scale the same way. For instance, Sun Directory (10 million+ users) scales better than Active Directory (1-2 million maximum), because AD is burdened with extra data for managing the network. Basically, AD trades volume for richness of information. Because of this, when you're storing identities for externally-facing initiatives, you would traditionally store it in a standard LDAP directory, such as Netscape, Sun, or Oracle Directory.

So let's say you combine an external Sun directory full of your customers with an internal AD directory full of employees. Now your aggregate virtualization speed is a lot slower using the proxy method, because you are constrained by the weakest link. If you can live with this slowdown, then the proxy approach will still work for you.

But such projects—**large-scale WAM and portal deployments that must integrate large, heterogeneous populations**, as an example—usually require a more strategic approach. You'll either need heavy hard-coded synchronizations or the smart synchronization you get from a more robust virtualization solution, with a **transparent, model-based approach and sophisticated caching technology.**

Level 4: Enriching Profiles with Context-Driven Attributes

As you begin to scale up to more complex use cases, such as finer-grained authorization and profile management, virtualization necessarily becomes the center point of your identity management infrastructure. In such cases, it's essential to integrate not only identity, but also its application context, to deliver the attributes needed for richer, more personalized services.

Metadata: Heterogeneous

Volume: Any

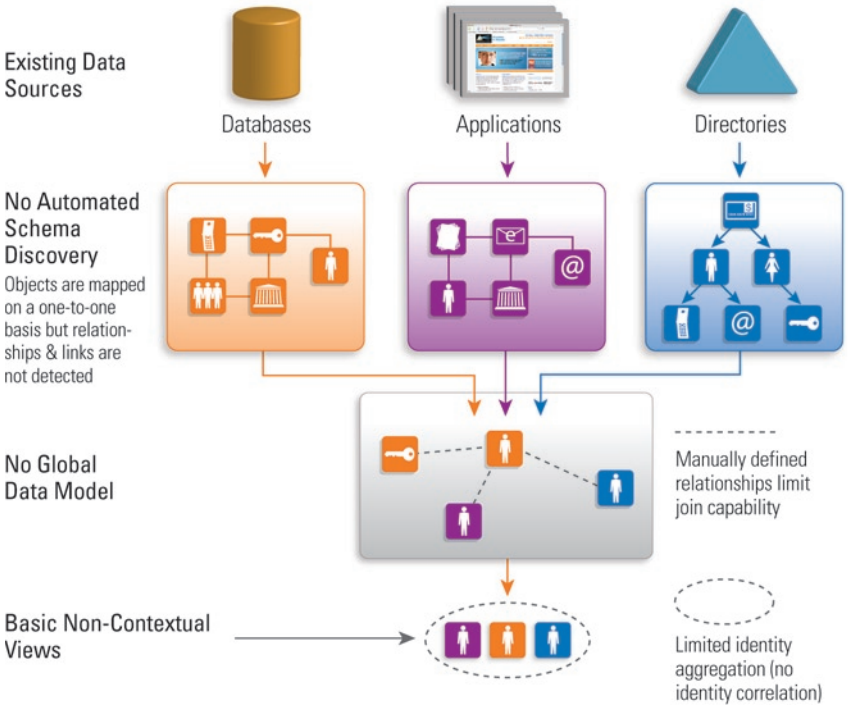
Enables: Authentication, Fine-Grained Authorization, Profile Management

In such instances, we must distinguish between two cases:

1. One consists of object mapping, or "identity virtualization" for authentication.
2. The other is object/relationship mapping, which is about externalizing contextual information from existing applications and linking this information to identities for authorization.

In both cases, you will need tools to support a clear representation of the data model of a given data source. This diagram shows how the traditional virtual directory is not equipped to build a data model.

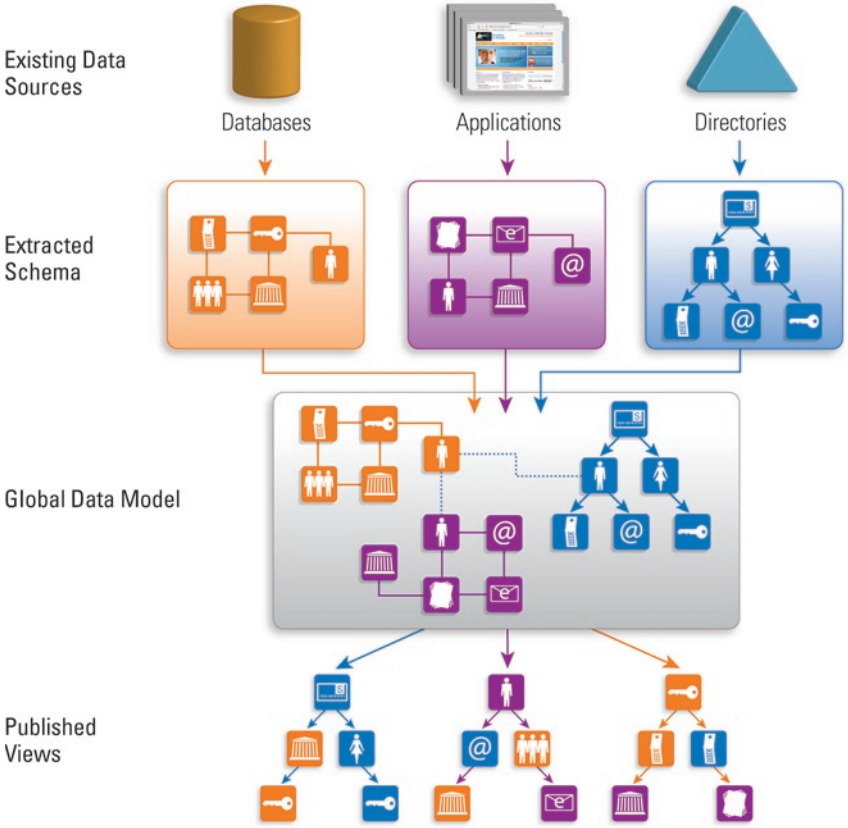
THE LIMITS OF THE TRADITIONAL VIRTUAL DIRECTORY



Objects can be mapped and linked/joined, but joins are limited by the scalability and speed of the operation, duplicate identities aren't correlated, and there's very limited support for views and context. While a traditional virtual directory may be able reproduce an existing directory hierarchy, it can't produce a virtual view that's more tailored to the needs of specific applications.

Once these specific models are "externalized" into a common model, or neutral representation, you can begin to recombine them. With this larger model, you can deliver a "virtual representation" of your data that exactly matches the needs of the consuming application, but which could be quite different from the starting sources. Virtualization through model allows you to deliver the view you need for your application, without having to worry about how it's structured by the producing application.

BUILDING THE DATA MODEL & PUBLISHING CONTEXTUAL VIEWS



How model-driven virtualization works:

- ***First, you discover and extract a model for each local source, where not only objects but also their relationships are mapped and virtualized.***
- ***Then these local models are linked through common identities into a global model.***
- ***Relationships are the key enablers to link/join the identity of a subject to its relevant application context.***
- ***These links/joins expose the attributes needed for finer-grained, context-driven authorization and context-aware applications.***

Identity can by linked and extended dynamically, with attributes coming from applications.

Level 5: Creating a Common Identity and Context Service

Metadata: Extremely Heterogeneous

Volume: Large

Enables: Authentication, Authorization, Provisioning, BPM, Federated Provisioning, Context-Aware Applications, Tracking Applications

The highest level of virtualization is aimed at **solving one of the thorniest challenges in identity management and IT: The full integration of data silos.** By creating a common identity and context layer at this level, you have now a complete picture of your system, where each object can be reached and understood in its relevant context. The opportunities at this level are huge, because **for the first time you have a 360-degree view of your processes and how they interrelate.**

In a first step, you could limit the features to reading the state of the system but such a service enables you to write and update, as well. An identity and context service becomes a common abstraction layer that hides the complexity of your infrastructure from the rest of the application world. The integration of this layer is provided by the support of multiple protocols, including LDAP, SQL, web services, and the transaction support that's essential to distributed operations and federated provisioning.

Virtualization in this case radically improves productivity and simplifies the data and context integration process, but while the value is high, the deployments are necessarily more complex, as well. Essentially, your virtualization layer must be complemented by systems such as an offline identity correlation engine, featuring transaction and synchronization support. (For more information, learn about our Identity Correlation and Synchronization Server.)

You can only achieve this level at the end of the of the maturity cycle, because each step along the way is critical to building the infrastructure needed to truly deliver advanced use cases, such as provisioning, BPM, federated provisioning, and the creation of context-aware applications. The failure of current provisioning initiatives illustrates the need for such an evolution; today's efforts tend to be too ambitious and built on infrastructures that have not been developed or allowed to mature.

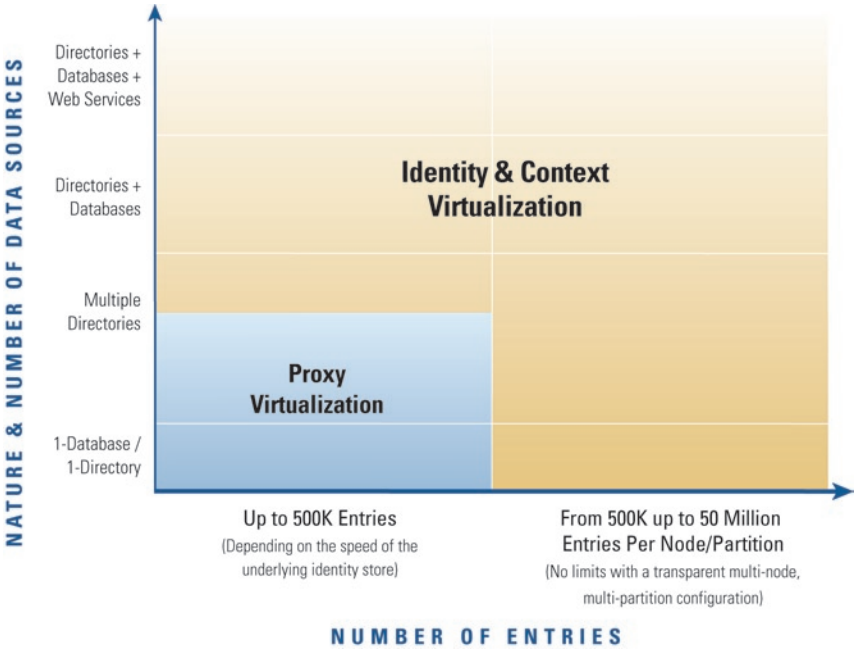
By virtualizing both identity and context, you can **leverage the investments you've already made, while creating a unified infrastructure for future initiatives.**

Section 2: Why Speed is Essential—and so are Volume and Richness

Underlying all these use cases are three main parameters: volume, richness, and speed. Speed is a key requirement for any directory—and this is true for a virtual directory, as well. While all directories need to be fast, two essential parameters can affect your speed:

- **Volume** of data, users and user types, and queries.
- **Richness** of data and user profiles.

As we'll demonstrate, volume and richness both have an impact on speed. These parameters also influence other important factors, such as scalability, availability, and stability.



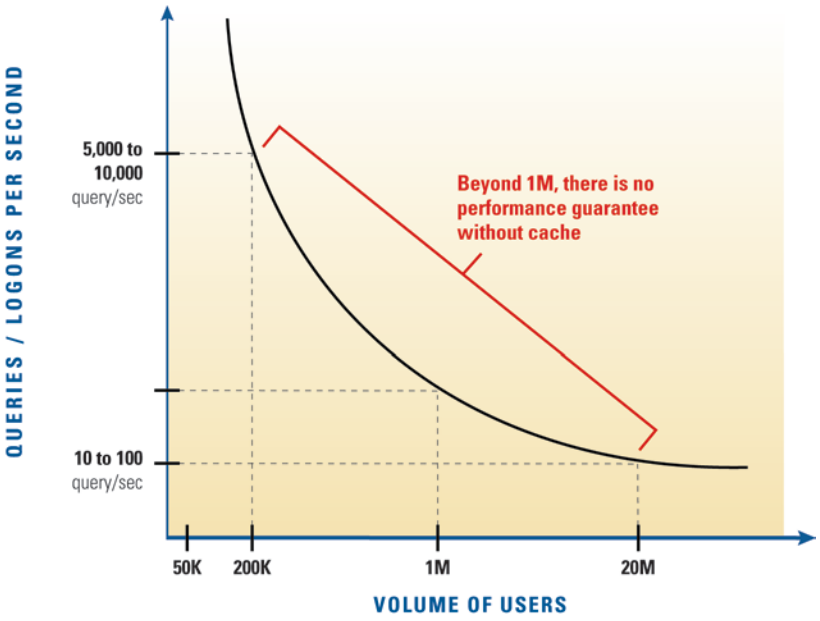
Model-driven virtualization becomes essential as volume and richness rise.

Volume Matters in Authentication

When preparing identities for authentication, your main challenge is building a single list from across all your data sources. During this process, volume comes into play in several ways:

- You may have multiple heterogeneous data sources, including some that are only accessible via SQL.
- You may need to authenticate many people, some of who may be outside the firewall or distributed over the cloud.
- Your system may need to handle a spike in queries, depending on business initiatives and other factors.

While richness is not as important for the authentication process, volume is key. You need to decide whether your infrastructure should be able to handle multiple data types and scale to millions of users.



In a pure dynamic query/proxy approach, performance will drop very quickly with volume. The aggregated average speed is conditioned by the slowest link.

See how speed is affected by volume.

The Limitations of Proxy

Unfortunately, **if you aggregate information by proxying back to the data sources, your speed will drop as the volume of users increases.** Because each data source has an intrinsically different speed, aggregating by proxy tends to drive speeds to an average that's conditioned by the weakest link. Let's say you aggregate five millions user in one fast directory with 500 users from a slow database. When queries are done serially, the result is a slower query time for everybody. While you could mitigate this slowdown to an extent by sending requests in parallel, you could also overwhelm the slower source with non-relevant queries.

Imagine you're a bank offering online services to customers, employees, vendors, and other stakeholders. Your customers expect an instantaneous log-on, but you may have a small database of investment professionals from your money management department. Using the proxy method, each log-on would necessitate either a serial search through all your data sources, or a parallel query of each source. In either case, your speed drops to unacceptable levels, because the system has to hit each source, and that database is not designed for the instant log-on your customer base demands—and it can't handle the traffic of millions of queries.

Richness Matters for Authorization and Profile Management

Attributes are key. **The more information you have about each user, the better you're able to secure your resources and offer new services to your customers.** But that information is often scattered across different systems.

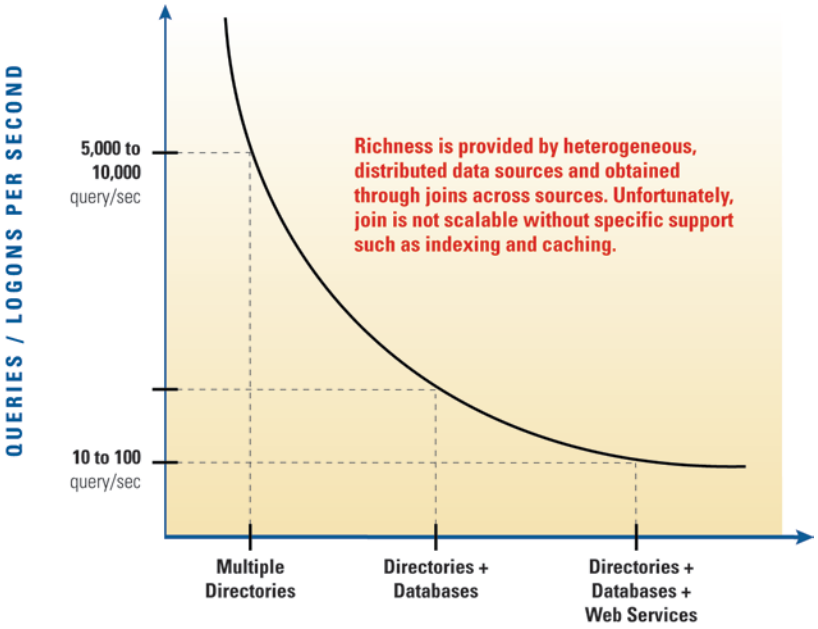
To truly deliver on fine-grained, context-driven services referenced in Burton's coming IdM infrastructure, you need to build a complete profile, bringing together all the attributes for each person, regardless of where or how they're stored. If this is important, then the virtual directory you select needs to be able to build a global profile. Ask yourself:

- Do I want to perform finer-grained authorization and enforce policies contextually?
- Will I need a cross-functional understanding of my users to identify cross-sell and upsell opportunities and better develop targeted new services?

If this is important to your business—or may become more important—then you need a virtualization solution that enables you to boost the richness of your user profiles.

Join is the Key to Richness—But Joining Across Systems is Complex and Expensive

Join is essential for enabling authorization—it's how you bring together different attributes from across all your disparate sources, such as directories, SQL databases, and web services. But like so many things, the more valuable the outcome is to your organization, the more costly the operation is to your backend. Under the load of complex joins, even the fastest sources can become too slow for identity and security applications.



The key to richness is your ability to join across diverse sources. Unfortunately, join is a costly, complex operation.

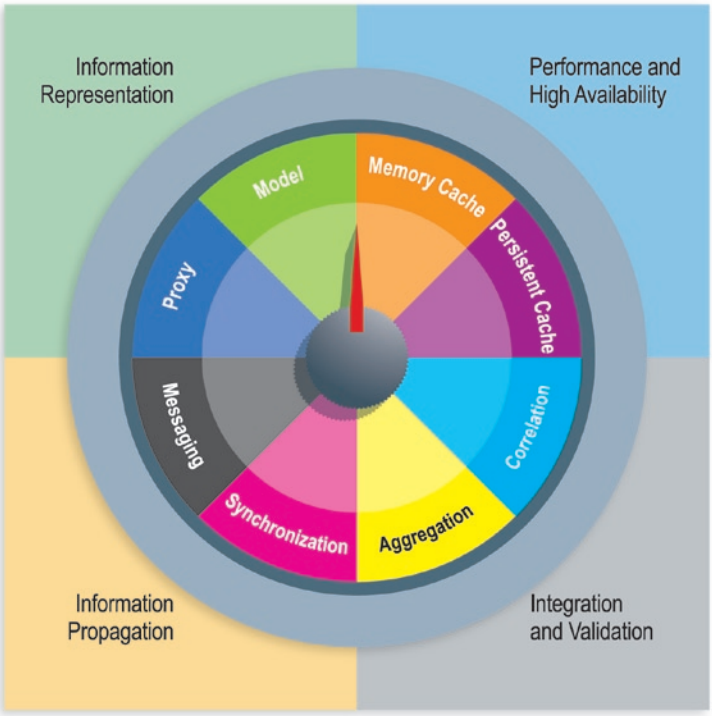
See how rich data and diverse sources affect speed.

Be sure to ask yourself where your proposed solution fits within these sectors.

No system can afford to do all these complicated joins across heterogeneous sources in real-time. With a **persistent cache**, you can render and persist a global data model—or series of materialized hierarchical views—without constraining query performance with complex joins and searches across multiple data sources. To build such a data model, your system must be able to externalize and represent the information that's specific to each application. If you want to deliver more attributes across diverse systems, **you need to understand the links between objects and their contexts within existing systems**. Then, by linking them across other systems, you're able to form a global data model that can be used to build new representations of data.

Section 3: How Different Virtual Directories Work

Virtualization solutions follow different patterns across several broad categories, and these patterns determine what you can do with your virtual directory. This dial diagram illustrates the different approaches across key sectors, including how information is represented, integrated and validated, and propagated, and how performance and availability are ensured.



Be sure to ask yourself where your proposed solution fits within these sectors.

Five Questions to Ask Yourself When Considering a Virtual Directory Solution:

1. How should my information be represented?

There are two ways to present information in a virtualized world:

- **A proxy view** is a remapping of objects as they exist in your data repositories, which means your virtual world is the same as the actual one.
- **A model view** virtualizes both objects and relationships from across diverse, distributed sources, enabling you to create infinite new views of your data.

If richness is not—and will never be—an issue, the proxy view will serve you well. However, if richness is key for your initiatives, you will need the ability to model your data and create new views from it. **The advantage of the model-driven approach is that you can create exactly the views you want.**

2. Can I just aggregate identities, or do I need to integrate? Is data quality an issue?

When you're virtualizing multiple data sources, it's important to consider how your system will integrate and validate this data. There are two different approaches:

- **Data aggregation,** which means the system is brought together but remains as it is, with no integration or disambiguation.
- **Data integration,** where the system is brought together, overlaps are detected, and the data is correlated, disambiguated, and logically regrouped.

When you aggregate, you're merely bringing data together, without doing anything to improve its quality. Such an approach works well when you're dealing with a smaller number of data sources with relatively clean and/or known data. But when the target is to integrate your identity into a logical structure that gives you a single version of truth, then you must integrate identities. As your data volumes rise, data quality starts to matter more, particularly **when you're trying to authenticate users from multiple overlapping systems.**

It's important to note that some of this process cannot be done on the fly, because it consumes too much processor time. Such correlations are performed as an offline staging operation. **This is where virtualization meets classical synchronization**, such as that offered by metadirectories or MDM/CDI solutions. In this case, you need more than a simple lightweight virtual directory; such a job requires a complete virtualization solution. The advantage of data integration is that you get the data quality you want.

3. What kind of performance do I need?

When looking at views of your data, you need to render your virtual world as quickly as possible. In this instance, both volume and richness play a role.

If you have small volumes and you're not looking for rich profiles, a classical virtual directory with a memory cache will work for you.

But if you have a sizable, rich system with an extensive set of profiles, then high availability, scalability, and stability are essential. In such a case, **you need a complete virtualization solution with persistent cache**, because you don't want this virtual world to disappear after the next power outage. And because such worlds are dynamic, cache refresh needs to happen in real-time—so if something happens in the real world, you see it reflected in the virtual world immediately. The advantage to such a solution is that you get the levels of performance you need.

4. How do I want to propagate information?

With any system, it's important to stay synchronized, with changes propagating as quickly as possible.

If you don't need immediate and constant synchronization, then a classical virtual directory will work for you.

But **if you're dealing with high volumes, then data synchronization is essential.** The problem in a distributed system is that there's always failure between nodes. Your system must be resilient against failure, so you need a solution with a messaging system that collects any changes in a queue. So even when the system's down, your changes are noted and propagated when the system's back online. The advantage to messaging is that your data stays synchronized, even if the system fails.

5. Where's my infrastructure headed?

Ideally, any solution you select allows you to scale beyond your initial requirements, in ways that you may or may not be able to foresee. Over the past decade, one thing we've seen is that **needs always outpace initial requirements—there are always more identities, more diverse data sources, and more demanding integrations.** Ask yourself:

- Could my company merge or acquire other companies?
- Will I want to extend access beyond the firewall?
- Do I need to offer expanded service to customers?

If you answer yes to any of these, it's important that your solution scale to accommodate tomorrow's new demands and architectures. **The advantage to such an approach is that your system has the future built in.**

Conclusion: Different Solutions for Different Needs

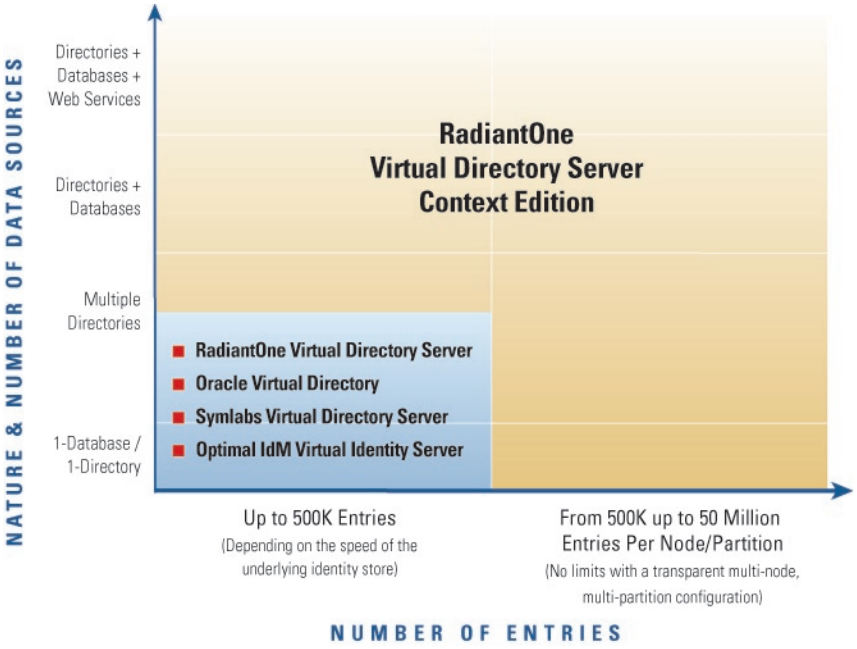
When looking for a virtual directory solution, be sure to consider your needs and your architecture, and ask yourself:

- Does it support your environment, whether it's relatively homogenous or increasingly heterogeneous and distributed?
- Can it handle the users you serve, whether you're authenticating 40,000 employees or managing profiles for millions of customers?
- Will it deliver the performance you need, even as those needs continue to grow?

There are many virtual directories to choose from, and we hope we've given you some helpful parameters by which to evaluate the different options that are available.

Why You Need Identity and Context Virtualization

Of course, we're a bit biased: While there are several options in the virtual directory market, we know **there's only one solution that allows you to virtualize both identity and context, creating an infrastructure that satisfies today's needs and tomorrow's opportunities.** The RadiantOne Identity and Context Virtualization platform combines the best of both meta and virtual directories, delivering a global view of identity and context, while enforcing security at the local level, as close to the sources of services as possible.

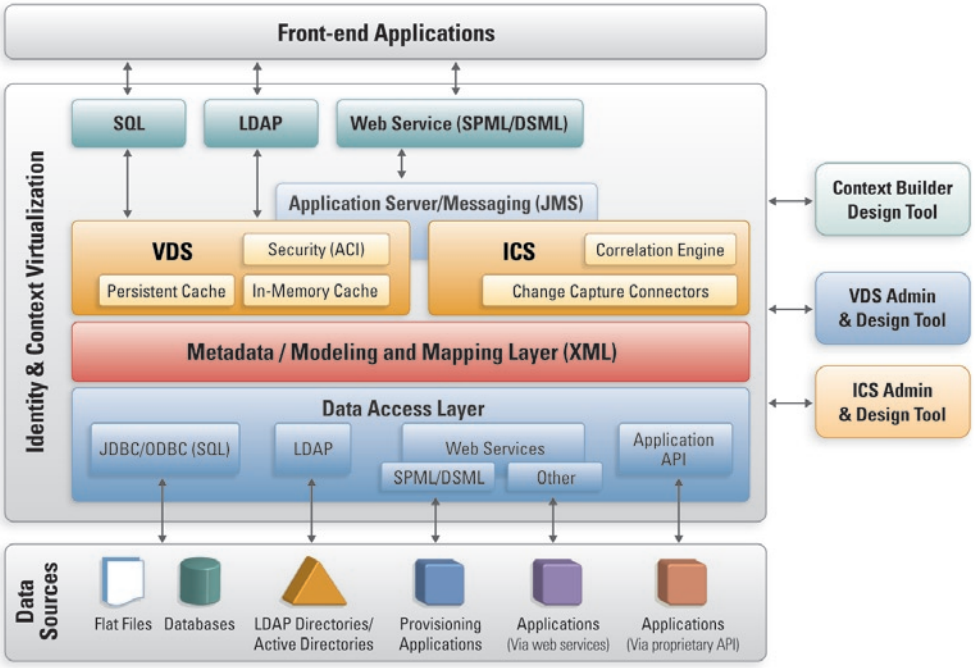


RadiantOne is the only virtual directory solution proven to scale to 20 million users with guaranteed performance.

With this innovation, Radiant Logic moves virtual directory technology beyond traditional proxy-driven directory aggregation into a data model-driven solution for complete identity integration and context management. **Identity and Context Virtualization meets the challenges of identity integration in today's high-volume, increasingly heterogeneous identity environments**, with multiple user populations, such as customers, partners, and employees, and disparate data sources, including directories, databases, and web services.

We'd love to show you what Radiant has to offer, from a proxy-driven point solution to a complete virtualization platform that scales along with your growing needs. No matter what you need now, only RadiantOne Identity and Context Virtualization allows you to take an incremental approach to shifts in your infrastructure, respecting what's already there even as it enables sophisticated new technologies and as-yet-undiscovered use cases.

About The RadiantOne Identity and Context Virtualization Platform



Feature	VDS	VDS Context Edition	RadiantOne Suite
Directory-to-Directory Mapping	x	x	x
Database-to-Directory-Mapping	x	x	x
Directory-to-Database Mapping (VRS)		x	x
Simple Join	x	x	x
Extended Join	x	x	x
Context Virtualization		x	x
Modeling New Directory Views		x	x
Modeling New Database Views		x	x
Memory Cache	x	x	x
Persistent Cache		x	x
Real-Time Cache Refresh		x	x
Identity Correlation with Unique Identifier		x	x
Identity Correlation without Unique Identifier			x
Point-to-Point Identity Synchronization			x