QUEST SOFTWARE®

# Unified and Intelligent Identity and Access Management

Authors
Jackson Shaw
Quest Software, Inc.

WHITE PAPER

Quest Software, Inc.
Attn: Legal Department
5 Polaris Way
Aliso Viejo, CA 92656
**www.quest.com**
E-mail: **legal@quest.com**

Refer to our Web site for regional and international office information.

## Trademarks

# Contents

# Abstract

Part of managing today's complex and diverse IT environments requires that users must be set up with separate identities and associated roles in order to have access to each required application, operating system, database platform, and so on. This approach means users have multiple passwords to remember and the IT staff has to duplicate work to provision and manage users on each system. This impairs productivity and increases the risk that a user may receive inappropriate access to valuable data and other company resources.

A unified and intelligent approach to identity and access management (IAM) offers an alternative. Organizations can consolidate each user's multiple identities into a few or, ideally, just one identity, and create a single set of roles, rules, workflows, and attestation around that one identity. This approach significantly simplifies identity and access management (IAM), improves user and IT productivity, and enhances security and compliance.

This white paper explains how this unified and intelligent approach to IAM simplifies a number of key tasks: provisioning, single sign-on, role management, multifactor authentication, privileged account management, and password management.

# Introduction

Your organization is most likely a diverse mix of applications, operating systems, databases, platforms, and other technology. It's also probable that your enterprise grew to this complex, diverse, disjointed state organically—a new application here, a new platform there.

This growth means that each user now has several, or even dozens, of separate identities spread across your diverse systems, and possibly just as many separate roles associated with his or her job. Each role might be called the same thing on each system, but there is no correlation between those roles. Each identity is an island unto itself, managed by any number of different teams.

Now consider the rules and policies that control user and group access. It's common for policies to be similar but not consistent across all systems. Often they are managed in an ad-hoc fashion by different IT teams and influenced by different business drivers. Therefore, the processes for managing access, like the provisioning processes, must be duplicated across systems. Because those processes are often tedious, error-prone and inconsistent manual processes, the result is more inefficiency. In fact, the process for requesting, approving, and granting access is typically a disjointed collection of true business processes, tribal knowledge, and settling for the easiest way to "just get it done."

The following table illustrates the complexity of managing access when users have separate identities and roles on multiple systems. In the table, each row represents a different physical user, and each column represents a different system that the user must access to do his or her job. The symbols in the cell represent user identity, role, workflow to establish and manage the identity and access, and approvals and attestation to ensure that everything happens in a way that supports business needs, security, and compliance. The colors roughly represent the multiple versions of each component that must be managed for IAM. Notice that there is no rhyme or reason to the identities, roles, workflows, and approvals associated with each physical user. For example, User 1 has the same user role for App 1 and App 3, but a different identity for App 3, even while having the same role across all three.

| | Windows/AD | App 1 | App 2 | App 3 | Unix 1 | Unix 2 | Unix 3 | DB | Mainframe |
|---|---|---|---|---|---|---|---|---|---|
| User 1 |  |  |  |  |  |  |  |  |  |
| User 2 |  |  |  |  |  |  |  |  |  |
| User 3 |  |  |  |  |  |  |  |  |  |
| User 4 |  |  |  |  |  |  |  |  |  |
| User 5 |  |  |  |  |  |  |  |  |  |
| User 6 |  |  |  |  |  |  |  |  |  |
| ● = user identity ◆ = role ◘ = workflow ■ = approvals/attestation | | | | | | | | | |

**Table 1. The complexity of managing access for users with separate identities and roles on multiple systems**

It all comes down to complexity. Simply stated, your enterprise, like almost everyone else's, likely requires:

- Too many identities for any one individual user
- Too many independent roles spread across systems that functionally define what the same user can and can't do on each system
- Too many manual processes performed by too many different IT teams to set up, manage, and terminate access
- Too many similar (yet unrelated) processes to achieve your identity and access management (IAM) goals
- Too many places for line-of-business personnel to request and approve IAM activities for their assigned users
- Too much of IAM being driven by your IT staff and its technology resources and capabilities, rather than being driven by actual business or organizational needs

# The Solution: Identity Intelligence

## What is Identity Intelligence?

These issues mean that, at most organizations, IAM lacks two things: unity and intelligence. Imagine the benefits of unifying identities, roles, workflows, and attestations to provide a single all-powerful intelligent approach to IAM ("identity intelligence") that affects each and every system, user, situation, and requirement (operational, security-related, and compliance-driven). If this were possible, IAM would suddenly transition from a difficult, expensive, and troublesome undertaking to a structured, achievable, and individually optimized approach that moves your business forward rather than holds it back.

## Traditional Strategies for IAM

Traditionally, the complexity and diversity of IAM components has been viewed as a necessary evil. After all, the technologies (applications, databases, and platforms) you need to run your business demand separate identities, roles, workflows, and approvals…don't they? And each requires its own dedicated piece of your IT team to ensure that these assets deliver their promised benefits. And to top it off, you have the ever-changing world of security and compliance that pulls at your established practices, demanding increased levels of control and visibility. It's a constant battle to get things done, at the most affordable price possible, without compromising security and compliance. After all, IAM is a tool to run your organization, not the reason your organization exists in the first place.

Historically, two major strategies have driven efforts to control the complexities of IAM:

- **Using point solutions:** Organizations try to address specific needs on individual systems with point solutions; for example, implement a self-service password reset solution for one system and synchronize it with others
- **Using an IAM framework:** Another approach is to implement an all-encompassing framework upon which IAM can be custom-built for an organization's specific environment, requirements, and goals. Examples include solutions from IBM (Tivoli Identity Manager), Oracle (including the recently acquired Sun solutions), Novell, Computer Associates, and the newest entry into the market, Microsoft Forefront Identity Manager (FIM)

Both approaches can provide value and move organizations closer to their objectives, but neither addresses the underlying cause of all the trouble—too much complexity. Identity, roles, rules, and policies continue to be managed inconsistently and individually (in the case of point solutions) or expensively with custom-coded logic (in the case of an IAM framework). Moreover, compliance and security are often addressed reactively rather than strategically.

### Using Point Solutions

Addressing specific needs with point solutions automates some tasks and increases security and compliance on the target system, but does nothing for those systems not addressed by the solution. The complexity remains, and to address additional needs on other systems, additional point solutions must be implemented—resulting in an even more disjointed environment with more tools and more work for IT.

### Using an IAM Framework

IAM frameworks have often been considered to be the “only” real way to do IAM. However, traditionally they have been custom built, making them very expensive with long development and deployment cycles. The vast majority of organizations simply cannot afford a framework or choose not to undertake a project of such epic scope. Moreover, like point solutions, an IAM framework does little to eliminate the underlying complexity that causes all the trouble in IAM. If a single user requires ten disparate identities to access ten different systems, the framework will still require all ten identities, and also add an eleventh identity in a metadirectory that controls all the others. If there are five different “versions” of the user’s role across the required systems, the framework will require custom-built business logic to negotiate the differences and idiosyncrasies of each. And, finally, all workflows and attestations need to be custom built in the framework—often duplicating those that already exist or requiring additional components to replace the ad-hoc ones.

## A Unified and Intelligent Approach to IAM

If all components of an IAM strategy were unified—meaning one identity per user across all systems, one set of roles that are applied universally wherever they are needed, a single set of workflows regardless of the systems involved, and one set of approvals/attestations driven by business needs rather than technology capabilities—managing identity and access would be simple, cost-effective, secure, and compliant.

Can an organization get there? It would be nearly impossible to throw out everything that has been built over the years and restart with a clean slate and a single source of IAM authority. No one is in a position to rebuild their entire environment around only Microsoft technologies, or only Oracle solutions, or only Linux options. In fact, much of the value of technology lies in the diversity of options and the opportunity to choose the best technology for a given need.

Fortunately, unifying identity and access management is not an all-or-nothing proposition. It is possible to dramatically reduce the number of identities in the enterprise. It is possible to condense disjointed and ad-hoc roles, workflows, and attestations into a more unified and consistent set. In other words, it is possible to maintain your desired technical diversity while simplifying the critical components of IAM—identities, roles, rules, policies, workflows, and attestations.

### Key Features of a Unified and Intelligent Approach to IAM

Key aspects of a new unified and intelligent approach to IAM include:

- Consolidating each identity to a single, already established identity where possible. (Many organizations choose to consolidate around Active Directory.)
- Creating a single set of roles, rules, workflows, and attestation around that one identity, which now controls a much larger portion of the enterprise
- Addressing key challenges with point solutions that perfectly support the unified identity namespace. For example, you can use AD-based enterprise single sign-on for systems that cannot be unified with AD, and platform-specific privileged account management that draws on AD roles and identity for targeted delegation of rights
- Wrapping the whole thing (unified and non-unified systems) with identity intelligence that takes all existing roles, rules, workflows, and attestations and interprets and converts them to a single set that accomplishes IAM with business objectives—not technical capabilities— as the driver

Put more simply, this new approach to IAM encourages you to:

- Get to one (or at least as close to one as possible) identity for identity administration and access control
- Make the whole solution intelligent by implementing a single, powerful, and all-encompassing structure upon which to build IAM including roles, rules, policy, workflows, and attestations

## An Illustration

The following table illustrates a unified and intelligent approach to IAM, with a single user identity and one set of roles, workflows, and approval/attestations applied consistently across the entire enterprise:

| | Windows/AD | App 1 | App 2 | App 3 | Unix 1 | Unix 2 | Unix 3 | DB | Mainframe |
|---|---|---|---|---|---|---|---|---|---|
| User 1 | ● ◆ □ ■ | | | | | | | | |
| User 2 | ● ◆ □ ■ | | | | | | | | |
| User 3 | ● ◆ □ ■ | | | | | | | | |
| User 4 | ● ◆ □ ■ | | | | | | | | |
| User 5 | ● ◆ □ ■ | | | | | | | | |
| User 6 | ● ◆ □ ■ | | | | | | | | |
| ● = user identity ◆ = role □ = workflow ■ = approvals/attestation | | | | | | | | | |

**Table 2. The simplicity of managing access for users with a unified approach**

## Using a Unified and Intelligent Approach to IAM

The remainder of this paper discusses how a unified and intelligent approach to IAM affects the most common tasks:

- Provisioning
- Single sign-on
- Role management
- Multifactor authentication
- Privileged account management
- Password management

This paper will also address the capabilities of the Quest One suite of identity and access management solutions. Quest One delivers the power of targeted point solutions along with the scope of an IAM framework but without their limitations. Quest One includes best-in-class solutions for IAM in Active Directory environments, one-time password (OTP) multifactor authentication, Active Directory bridge technology, and enterprise single sign-on.

In addition Quest One also provides the level of identity intelligence that makes the whole thing work together—based on your business objectives; taking into account your existing practices, capabilities, and processes; and with an emphasis on configuration rather than customization that delivers time-to-value in a matter of months, not years.

# Provisioning

For many organizations, IAM starts with provisioning and its three flavors:

- **Provisioning** – Setting up user accounts, group memberships, and rights
- **Re-provisioning** – Managing each identity throughout its lifecycle and changes in user roles and responsibilities
- **De-provisioning** – Terminating access when an employee leaves the organization (de-provisioning) is critical to most IAM projects

## Challenges of Traditional Provisioning

Provisioning is often one of the most complex and challenging IT tasks in large organizations. Some typical challenges include:

- Identity and access must be provisioned to multiple systems, with differing capabilities and requirements. There is no consistency across these systems for what is and is not required in an identity to grant access, resulting in a single physical user having many identities, each having no relation to the others
- Specific provisioning tasks may be performed by different IT teams, some of which may not perform the action in a timely fashion—a major compliance and security concern for de-provisioning
- Rules, roles, and policies vary from system to system and have no single, common controlling structure behind them
- Much of the provisioning process is based on “tribal knowledge,” and when individuals with that knowledge leave the organization, often the knowledge goes with them
- There is no consistency in how provisioning actions are performed, who does the work, who approves the action, and what checks and balances are in place to ensure compliance and security
- Traditional provisioning automation solutions require significant custom coding to ensure consistency, security, and proper workflow
- Manual provisioning processes are tedious, error-prone, and lack control

# A Unified and Intelligent Approach to Provisioning

Applying the unified and intelligent approach to IAM significantly improves the provisioning process. Imagine if only one identity was required for multiple systems—for example, if Unix, Linux, Java, and Mac accounts were eliminated in favor of a single Active Directory account. Only a single provisioning action would be required to establish access to numerous systems, eliminating dozens of disjointed, non-correlated tasks. And if the technology that automates provisioning was based on business needs and didn't require large amounts of custom coding, that one provisioning action could be more accurately and thoroughly executed. In addition, a single, all-powerful workflow, attestation procedure, and set of controlling polices and rules could ensure that the provisioned account is set up correctly with minimal IT involvement, granting only appropriate access for the user. In addition, this structure moves tribal knowledge and "we do the best we can with what we've got" to a controlled and intelligent foundation for all of provisioning.

But we all know that getting to a single identity (and thus only one identity to provision) is not entirely achievable. But anything that can be done to eliminate redundant identities is a major improvement. And the unifying force of identity intelligence—getting to "one" for roles, rules, policy, workflow, and attestation—is achievable and can dramatically improve the security, efficiency, and compliance of enterprise provisioning done through automated tools. And if that single controlling set of identity intelligence is available and based on your real-world business and organizational requirements (rather than IT or technology limitations), a world of possibilities opens up.

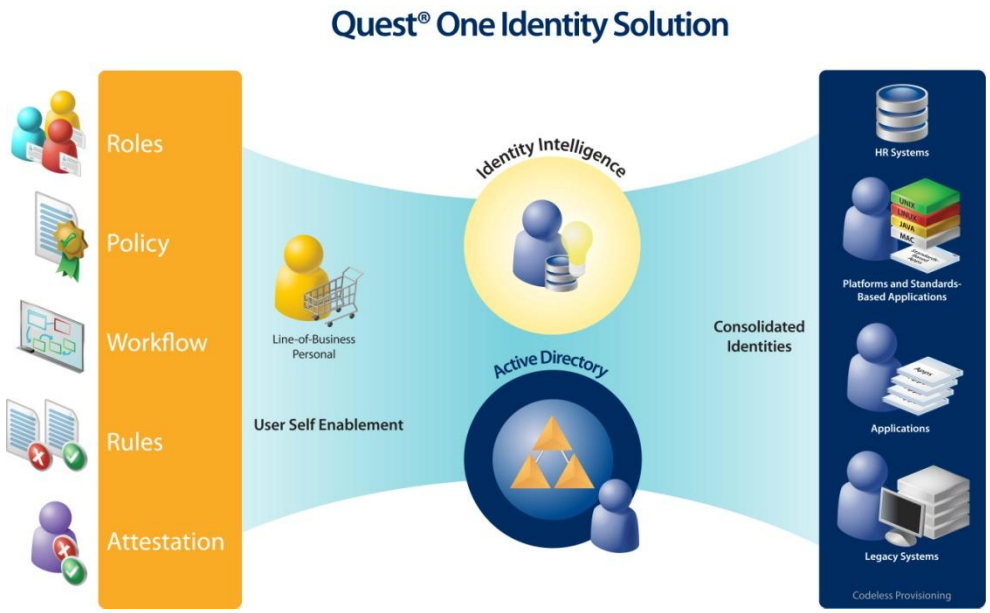


**Figure 1. The Quest One unified approach simplifies provisioning.**

The following table explains how a unified and intelligent IAM solution can improve key aspects of the provisioning process:

| | Unification | Identity Intelligence |
|---|---|---|
| Account provisioning, re-provisioning, and de-provisioning | Unix, Linux, Java, and Mac identities can be eliminated by consolidating them into the Active Directory identity. The result is significantly fewer identities to provision. | Codeless provisioning, with an emphasis on configuration rather than customization, builds on existing practices, policy, workflows, etc. to deliver rapid time to value. |
| Physical provisioning | | Self-service and "shopping cart" processes for end users and line-of-business personnel ensures that physical provisioning occurs rapidly and in perfect alignment with identity provisioning processes, workflows, attestations, and charge-back procedures. |
| Roles | Active Directory roles can be more granularly defined and used to control provisioning on Windows (including AD, Exchange, etc.), Unix, Linux, Java, and Mac systems. | Unifying disparate roles into a single, all-encompassing set ensures consistency and efficiency of provisioning actions while supporting stronger security and compliance. |
| Workflows | Workflows established in Active Directory can be leveraged to also cover AD-joined Unix, Linux, Java, and Mac systems. | Unifying all workflows into a single set ensures proper processes occur with maximum efficiency, automation, security, and compliance. |
| Rules | Established Active Directory rules (improved through more granular management than natively available) can also control Unix, Linux, Java, and Mac system access and provisioning actions. | Unifying rules into a single set ensures that provisioning actions and access control occurs consistently and in line with business objectives enterprise-wide. |
| Policy | When Unix, Linux, Java, and Mac identities are unified with Active Directory, the stronger provisioning policy (security, workflow, attestation, etc.) available through AD can be applied to those non-Windows systems. | Combining previously disparate provisioning policy collections into a single set ensures that all provisioning actions are executed according to their pre-defined and business-driven purposes. |
| Attestation | Existing attestation and approval structures can be expanded beyond Active Directory to also include AD-joined Unix, Linux, Java, and Mac systems. | Unifying all attestation and approvals across varied and diverse systems ensures that proper approvals always occur, regardless of the target system, provisioning action, user profile, or established workflows, policies, and rules. |

# Single Sign-on

Perhaps the most obvious evidence of the complexity inherent in identity and access management is the number of logins and passwords associated with individual users, the amount of management required to maintain the access granted by those logins, and the disparity of how those passwords are controlled and administered. Industry trends reveal that a user in a typical 10,000-employee organization will have between 5 and 14 different passwords, and the cost of resetting those passwords (usually requiring IT staff involvement) is between $20 and $45 per incident.

## Types of SSO Solutions

Single sign-on (SSO) could overcome these costs. But is SSO the "holy grail" of IAM—a mythical and magical objective that does not really exist? No; SSO comes in different flavors, and its value is directly tied to how it addresses the underlying complexity that is the root of most IAM challenges. Typical SSO solutions include:

- **Password synchronization ("same sign-on")** — This approach ensures that all passwords across the environment are the same. Often called "same sign-on," password synchronization is the oldest and most popular form of SSO
- **Enterprise SSO (ESSO)** — This technology securely stores all passwords across the environment and automatically enters them where and when needed. Sometimes called "login automation," ESSO solutions have been considered the next generation of SSO
- **True SSO** — This approach actually enables multiple, disparate systems to use the same login, password, identity, and credential. Most often these solutions rely on the Kerberos capabilities of Microsoft Active Directory. True SSO is the most secure, efficient, and compliant form of SSO

## Comparing the SSO Approaches

Each of these SSO options had its advantages and disadvantages:

| SSO Type | Advantages | Disadvantages |
|---|---|---|
| Password synchronization | Widespread use<br><br>Numerous vendors to choose from<br><br>Often included as part of an IAM framework<br><br>Can play a role in enterprise provisioning | Requires users to still log in separately for each system (albeit with the same password each time)<br><br>Does not eliminate the underlying complexity of multiple identities and passwords<br><br>Requires a "lowest common denominator" approach to password complexity<br><br>Requires significant investments in infrastructure (connectors) and often custom coding |
| Enterprise SSO | End users only login once (subsequent logins are automated under the covers)<br><br>Maintains individual system password complexity rules<br><br>Can leverage advanced security and delegation practices<br><br>Initiated with AD login | Does not address the underlying complexity of multiple identities and passwords<br><br>Complex tool may be difficult to implement on some systems |
| True SSO | Eliminates complexity by unifying identities and password<br><br>Leverages the Kerberos standard for security<br><br>Supports a unified approach to additional IAM tasks (such as provisioning, string authentication, password management, and audit)<br><br>Eliminates the need for additional infrastructure and management tools | Not all systems can be integrated through this approach |

**Table 3. Comparing the three approaches to SSO**

### True SSO

The most secure, efficient, and compliant solution to the single sign-on dilemma is the true SSO option—a strategy that fully supports the unified approach to IAM discussed in this paper. This is what Microsoft implemented as it moved from Windows NT to Windows XP with the true SSO environment of Active Directory. Subsequent technologies have been developed (starting with Quest Software's Authentication Services solution) that enable non-Windows systems to participate in the AD/Kerberos SSO environment as "true citizens." Today a high-number of systems can "join" AD for SSO, including:

- Unix
- Linux
- Mac OS X
- Java
- SAP (SAPgui and NetWeaver)
- Siebel
- DB2
- Samba
- PuTTY
- Apache
- Any application that supports the Kerberos standard
- Any application that supports LDAP
- Any application that provides "pluggable" authentication (such as GSS-API)

## The Best Option: a Blended Approach

However, not every system is equipped to participate in true SSO. For example, mainframes and mid-range systems (RACF), Oracle applications, and a number of off-the-shelf applications require their own proprietary authentication methods and cannot use AD's Kerberos SSO capabilities.

Therefore, a unified and intelligent approach to SSO would actually be blend of two approaches:

- True SSO for as much of the environment as possible
- AD-based enterprise SSO for those systems that cannot be entirely unified

This blended approach ensures the very best SSO option for each and every piece of the enterprise. Both options can use identity intelligence to improve and control user access while streamlining administration through unified roles, rules, policy, workflow, and attestations that influence how users authenticate and what they are able to authenticate to. In addition, both options can further strengthen authentication through intelligent and unified implementation of two-factor authentication.

Quest One offers this blended approach to SSO, as illustrated below:

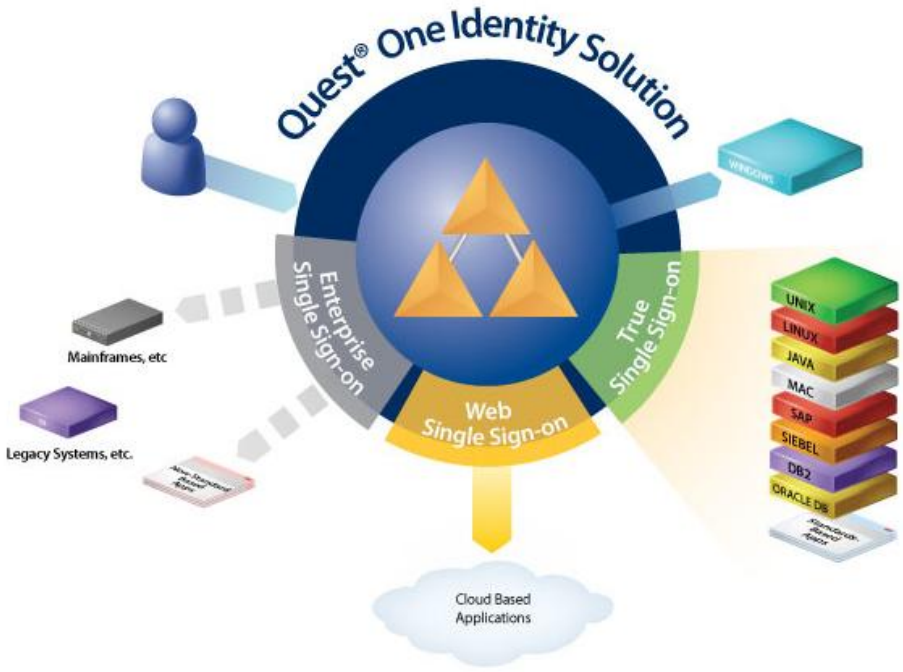


**Figure 2. Quest One offers a blended approach to SSO.**

The following table explains how a unified and intelligent IAM solution can provide SSO for different platforms and types of applications:

| | Unification | Identity Intelligence |
|---|---|---|
| Unix, Linux, Mac | Unix, Linux, Java, and Mac systems can achieve true SSO by being consolidated into the Active Directory identity | Access through any form of single sign-on can be controlled based on identity intelligence principles (roles, policy, etc.) centered on the unified IAM processes and identities |
| Standards-based applications | Many standards-based applications can authenticate through the AD/Kerberos credential generated at AD login | |
| Legacy systems | Legacy systems can use login automation (ESSO) based on an initial AD login | |
| Non-standards-based applications | These applications can use login automation (ESSO) based on an initial AD login | |

# Role Management

The roles associated with users provide the foundation for appropriate access and efficient administration: role-based access control (RBAC) requires that access to resources be controlled by the role associated with an individual user. Unfortunately, roles often proliferate just like identities, making it difficult to achieve RBAC.

To illustrate the challenge of effective role management, let's use the example of a new employee named John Doe at Company One. John has been hired as a data analyst to help relieve the workload of Jane Smith. In the company's HR system, both John and Jane have the job title "data analyst." John's new boss, Bill Jones, informs the Windows help desk that he has hired a new data analyst who needs an Active Directory account and access to Exchange and other resources and that John's rights, group memberships, etc. should match Jane's. Without an AD-based role management solution, the process of setting up John's accounts to match Jane's is a tedious, error-prone manual process.

Moreover, AD is not the only system John needs to access. For example, he also requires access to the company's Oracle systems. So Bill contacts the Oracle team and informs them that John needs access that matches Jane's. Once again, IT is diverted from their core responsibilities to manually research Jane's permissions and group memberships and set up John's account to match. The same process is repeated multiple times to grant John appropriate access to several other applications, the mainframe, and some Unix servers and the databases they host.

In the end, John is defined as a "data analyst" in AD, Oracle, several other applications, the mainframe, and some Unix servers and databases. But each designated "data analyst" role was set up independently of the others, and there is actually no correlation between them; John's "data analyst" role as defined in AD has absolutely nothing to do with the "data analyst" role defined for John in Oracle. Even if the role is called the same thing on each system and the same people belong to the data analysts group, functionally Company One has John (and Jane) associated with a different role on each and every system, and a change to the role in one system does not affect the others.

Of course, this issue is not unique to data analysts; the term "DBA" or "Developer" or "Manager" might be used to define roles in all the company's systems, but mean different things in each, as illustrated here:

| | AD | Unix | Oracle | Mainframe | App 1 | App 2 |
|---|---|---|---|---|---|---|
| User 1 | DBA | DBA | DBA | DBA | DBA | DBA |
| User 2 | Analyst | Analyst | Analyst | Analyst | Analyst | Analyst |
| User 3 | Admin | Admin | Admin | Admin | Admin | Admin |
| User 4 | Manager | Manager | Manager | Manager | Manager | Manager |
| User 5 | Developer | Developer | Developer | Developer | Developer | Developer |
| User 6 | Marketing | Marketing | Marketing | Marketing | Marketing | Marketing |

**Table 4. Roles might have the same name on different systems, but they are still unconnected.**

## Using Custom Business Logic as a Workaround

One approach to address the role management challenge is to build complex, custom business logic to deal with the relationships between all these disparate roles. This approach is similar to the synchronization scenario offered by traditional IAM frameworks. Unfortunately, this strategy relies almost exclusively on custom coding, and therefore it is very expensive and typically takes years to fully develop and deploy. Consequently, many organizations avoid this approach and continue to make do with their current disjointed method of manual role management, which makes RBAC virtually impossible to achieve.

## A Unified and Intelligent Approach to Role Management

A unified and intelligent approach to role management, on the other hand, can help organizations achieve RBAC quickly and at a fraction of the cost of traditional frameworks. Imagine if a number of disparate identities could be consolidated into a single, well-controlled directory (Active Directory for example). With Unix, Linux, Java, Mac, SAP, and other systems consolidated into AD, a single set of well-managed roles can affect all those systems. And by adding a layer of identity intelligence to the entire enterprise, roles across the whole environment can be correlated, resulting in a unified set of roles with universal applicability and dynamic adjustment for changes, additions, and evolution.

Using this approach, when Company One hires John and designates him a data analyst, each system automatically associates the correct permissions, group memberships, and even exceptions to John's role within the company. This approach enables Company One to achieve RBAC easily, and it is driven by real business needs rather than IT or solution capabilities.

| | AD | Unix | Oracle | Mainframe | App 1 | App 2 |
|---|---|---|---|---|---|---|
| User 1 | DBA | | | | | |
| User 2 | Analyst | | | | | |
| User 3 | Admin | | | | | |
| User 4 | Manager | | | | | |
| User 5 | Developer | | | | | |
| User 6 | Marketing | | | | | |

**Table 5. A unified approach assigns the same role across systems (not different roles with the same name).**

Additional advantages of the unified and intelligent approach to role management include:

- Role mining to discover, consolidate, and correlate roles across the entire enterprise
- Individuals and roles can be managed and maintained by those who are responsible for them (line of business personnel rather than the IT staff)
- A configurable foundation that does not require custom coding to achieve role management
- A role structure that can be used not only for RBAC but also for provisioning, audit, password management, and the rest of IAM


**Figure 3. Quest One offers a unified and intelligent approach to role management**

# Multifactor Authentication

A number of government and industry regulations and security best practices demand that user authentication be secured beyond the level offered by a simple user name and its password logins. This strong authentication approach to verifying a user's identity prior to granting access includes a number of two- and multi-factor options such as smart cards, one-time password (OTP) tokens, and even biometrics. Each option augments the user name and password with "something you know," "something you have" (smart card or OTP token), and/or "something you are" (fingerprint, retinal scan, etc.).

Implementing multifactor authentication solutions can be expensive because of the additional proprietary infrastructure required by traditional solutions. The solutions can also be difficult to manage, particularly when multifactor authentication is needed on multiple, non-integrated systems. It is not uncommon for users in highly secure IT environments to have multiple smart cards hanging around their necks or several OTP tokens dangling from their key chains, one for each system that must be accessed.

Many multifactor authentication vendors have convinced their customers that the proprietary architecture and limited life of multifactor authentication solutions is necessary. However, innovations in both technology and thinking have revealed that multifactor authentication does not have to rely on proprietary technology and that form factors can be non-expiring. New OTP solutions base all multifactor authentication on Active Directory, not an additional, proprietary directory, and use standards to ensure that organizations can choose form factors from a number of vendors.

## A Unified and Intelligent Approach to Multifactor Authentication

Real advancements in multifactor authentication, however, lie in the unified approach to IAM. For example, Unix, Linux, Mac, and Java systems that have "joined" Active Directory can be secured through the same multifactor authentication solution that secures AD access, eliminating the need for a wad of tokens or a collection of smart cards. Quest Defender, for instance, provides one-time password (OTP) authentication for any Unix, Linux, Mac, or Java system that has become part of the AD "trusted realm" through Quest Authentication Services. In addition, for organizations with an existing Windows smart card installation, Authentication Services extends its scope to Unix and Linux systems as well. In addition, Quest Enterprise Single Sign-on can initiate SSO with any multifactor authentication option (smart cards, OTP, or biometrics).

While the identity intelligence approach doesn't impact multifactor authentication directly, it can add a layer of control and visibility to multifactor authentication based on the unified identities, roles, rules, policy, workflows and attestations that may require multifactor authentication for access, or be necessary to provision and manage the multifactor authentication solution.

Quest One's unified and intelligent approach to multifactor authentication is illustrated below:

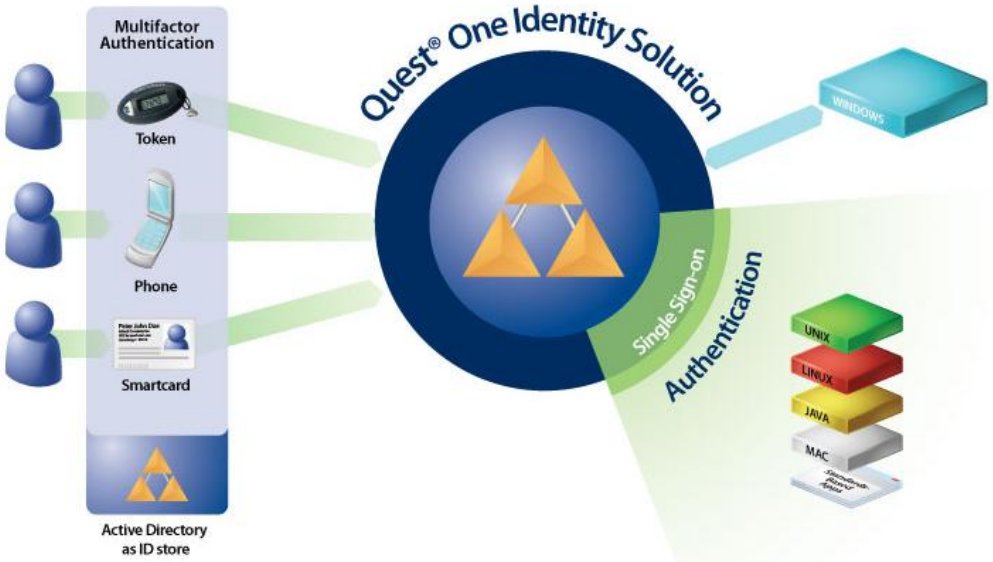


**Figure 4. Quest One's multifactor authentication using your choice of method.**

# Password Management

When users have multiple identities, they also have multiple passwords. This presents the following challenges:

- **Inconsistent password policies** – With separate passwords required on different systems, it is very difficult (and sometimes impossible) to establish a single, secure policy for all passwords. Some systems simply cannot support the same password complexity as other systems, and IT teams may have more important things to do than tear down and rebuild password policies to match other pieces of the enterprise
- **Expensive password resets** – The more passwords users have to remember, the more likely they are to forget them, resulting in lost productivity for employees and increased IT workload. In fact, a domain-expert IT staff member is often required to reset passwords, which is rarely part of the staff member's core job
- **Security and compliance risks** – When users have many passwords to remember and the rules for choosing passwords vary on different systems, users take steps like writing down their passwords. The result is often serious security and compliance holes that make inappropriate access very easy

So the real problems are:

- Users have too many passwords and those passwords are all different
- The organization has too many password policies and they are all different
- Expensive IT staff is often diverted from core responsibilities to address password issues for users
- Most user efforts to alleviate the problem of managing multiple passwords compromise security and compliance

## One Option: Self-Service Password Management

A self-service password reset solution that enables users to reset their own passwords can dramatically reduce the costs and risks associated with the traditional password management approach. Users who can reset their own passwords get back to work faster and are less tempted to write down their passwords, enabling IT staff members to focus on their core responsibilities.

## A Unified and Intelligent Approach to Password Management

Simply enabling users to reset their multiple passwords, however, does not go far enough—the goal has to be reducing the number of passwords the user must remember and IT must manage. The unified approach to IAM does exactly that: by consolidating identities for Unix, Linux, Mac, Java, and many standards-based applications, this approach eliminates passwords for those systems. Therefore, there is no need to reset those passwords, and no need for users to violate security and compliance rules by writing them down. In addition, a single, strong password policy can be implemented across the entire range of AD-integrated systems, and multifactor authentication can be used to further enhance security.

The unified and intelligent approach to IAM consolidates passwords for many non-Windows systems into AD; the rest are handled with enterprise SSO. This approach improves password management in the following ways:

| | Unification | Identity Intelligence |
|---|---|---|
| Inconsistent password policy | Consolidate password policy for a number of non-Windows systems into AD | Unify all password policies around a single, secure set of roles, rules, policies, etc. |
| Password resets | Reduce the number of passwords that must be reset by implementing unification technology and enterprise SSO | Empower users to securely reset their own passwords across the entire environment |
| Security and compliance | Consolidate less-secure policies for a number of systems into the existing secure AD policy structure, and enhance secure logins with multifactor authentication | Arrive at a single, secure and compliant set of password policies and influence those with unified roles, rules, workflows, attestations, etc. |

Quest One consolidates identities so users have just one password to remember for access to multiple systems, and offers self-service password reset to reduce productivity losses, IT workload, and security risks.

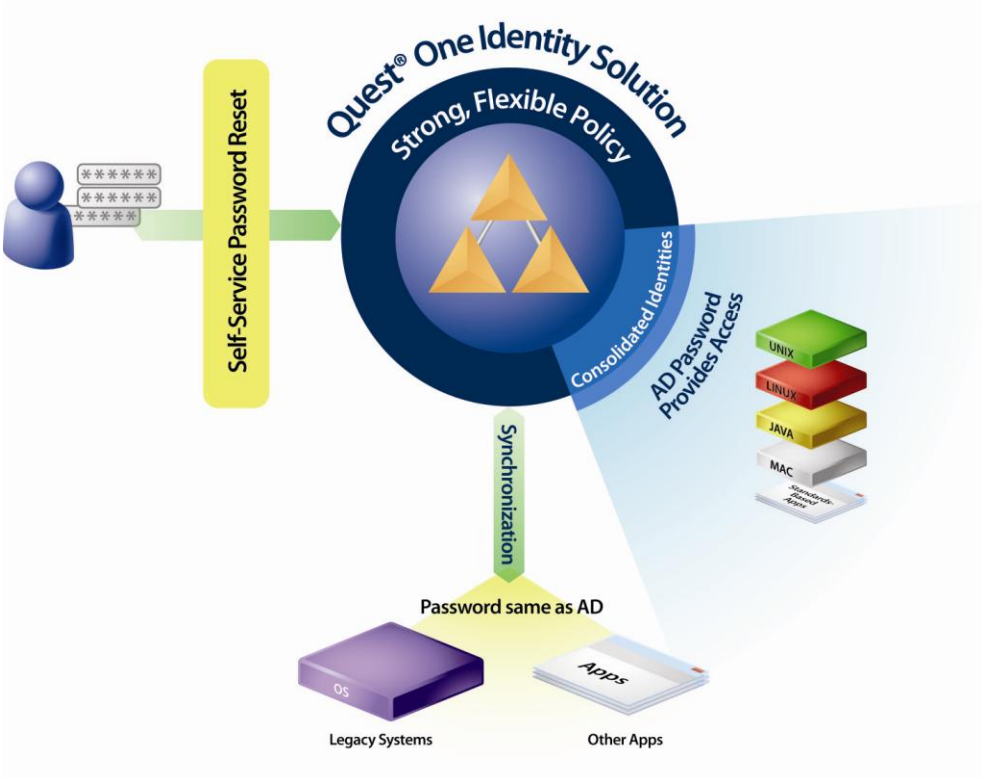


**Figure 5. Quest One not only automates password management it also unified identities to reduce the number of passwords in the environment.**

# Privileged Account Management

Critical systems and platforms require an all-powerful login that enables administrators to perform certain important administrative tasks. In the Windows world, this is called the administrator account. In Unix and Linux, it's the root account; virtually all other systems have their own variation. Some more advanced systems offer varying degrees of privileged access, but it is rare that they offer the granularity of control that real-world organizations need.

To illustrate the issue, consider the Unix/Linux world. There, the all-powerful root account is an all-or-nothing proposition: either an administrator has root privilege and can do anything and everything required to administer the system (including activities that violate security policy such as installing Trojan Horses, or editing and viewing logs), or the administrator does not have root access and cannot do anything. Consequently organizations are left with a choice: either tightly control root access and grant only a few highly trusted administrators the root credential, or give everyone that may need elevated access the root password and trust them to do the right thing.

Either choice has its drawbacks. In the tightly controlled option, menial tasks such as password resets or account provisioning that require root access can be performed only by the few administrators, which will pull them away from their entrusted tasks. On the other hand, giving many administrators the root credential increases the risk of catastrophic actions (either deliberate or accidental).

The same issues apply to the Windows administrator or any other privileged account.

## A Unified and Intelligent Approach to Privileged Account Management

A unified and intelligent approach to IAM addresses the issues of privileged account management by providing the control required to satisfy security and compliance demands while granting administrators the appropriate rights to do their jobs—nothing more, nothing less. By unifying on a single, all-encompassing set of roles and intelligently correlating those roles to granularly control entitlements, organizations can eliminate the troublesome problem of privileged accounts having the "keys to the kingdom."

Specifically, to properly handle privileged accounts, an IAM solution should:

- Provide the granularity to issue exactly the right level of access to each administrator
- Be closely tied to enterprise roles and controlled through centralized policy and common management interfaces
- Provide full audit trails of who has rights to what, who granted them those rights, and what they do with those rights—even down to the keystroke level if possible. This proactive compliance stance not only enables an organization to prove compliance but to continually achieve and maintain that compliance

In addition, privileged account management can be significantly enhanced by implementing identity consolidation technologies, such as technologies that enable Active Directory roles and Windows Group Policy to be used to control delegation of Unix/Linux root access. Privileged account management can also benefit from multifactor authentication as an additional layer of verification before certain sensitive actions can be performed.

Quest One provides centralized, granular control of administrator rights and a complete, secure audit trail:

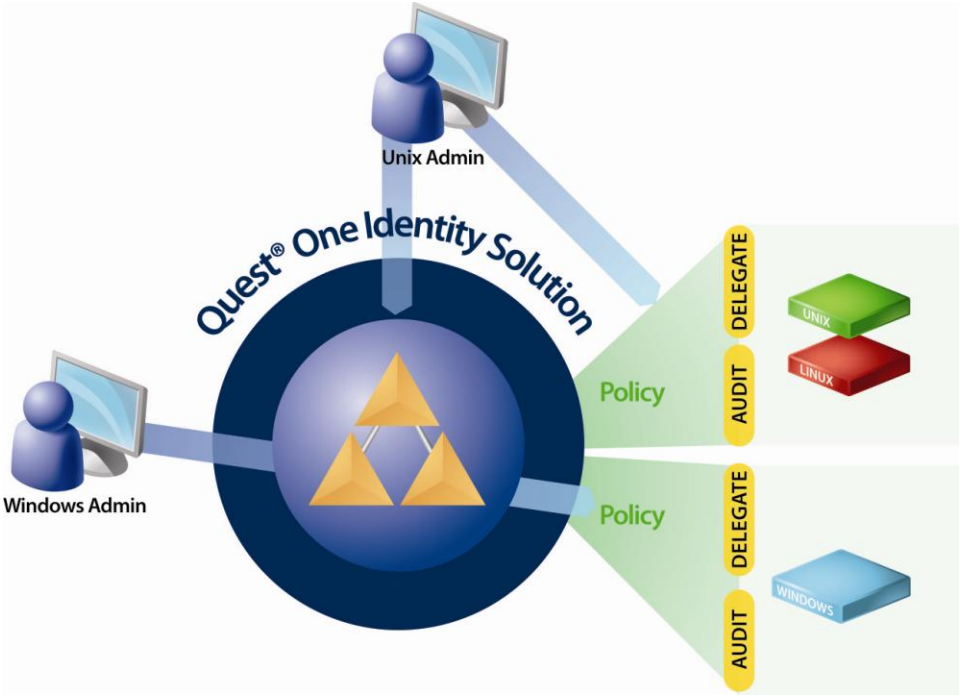


**Figure 6. Quest One provides intelligent privileged account management**

# What if I Already Have an IAM Framework?

Many organizations have already embarked on an IAM project with a traditional framework from IBM, Oracle (now including Sun), Novell, or CA, or with Microsoft Forefront Identity Manager (FIM). All of these solutions are custom built to meet the individual IAM needs of specific organizations. As a result, they can be complex and require significant investments in custom development.

Fortunately, a unified and intelligent approach to IAM can improve the performance of an IAM framework while reducing costs and accelerating time-to-value. Key areas of value for users of established IAM frameworks include the following:

- **Reduced complexity** – A unified approach can eliminate the need for custom-built connectors for many systems. By integrating Unix, Linux, Mac and Java with Active Directory, a single, well-designed connector for AD can take care of the provisioning, policy, and role management needs of a much larger portion of the enterprise
- **Optimized IAM for AD** – AD-optimized solutions for delegation, security, role management, and provisioning of AD (and AD-joined systems) can dramatically increase the value of an IAM framework and accelerate its deployment
- **No need for custom coding** – Unifying roles, rules, policy, workflows, and attestations through configurable identity intelligence eliminates the need for custom business process coding and one-off logic
- **Filling of functionality gaps** – IAM frameworks simply do not handle all areas of IAM. By augmenting an existing framework with unified and intelligent solutions for single sign-on, multifactor authentication, and privileged account management, organizations can achieve a more complete IAM solution quickly and affordably

The following figure illustrates how Quest One can augment an existing IAM framework:

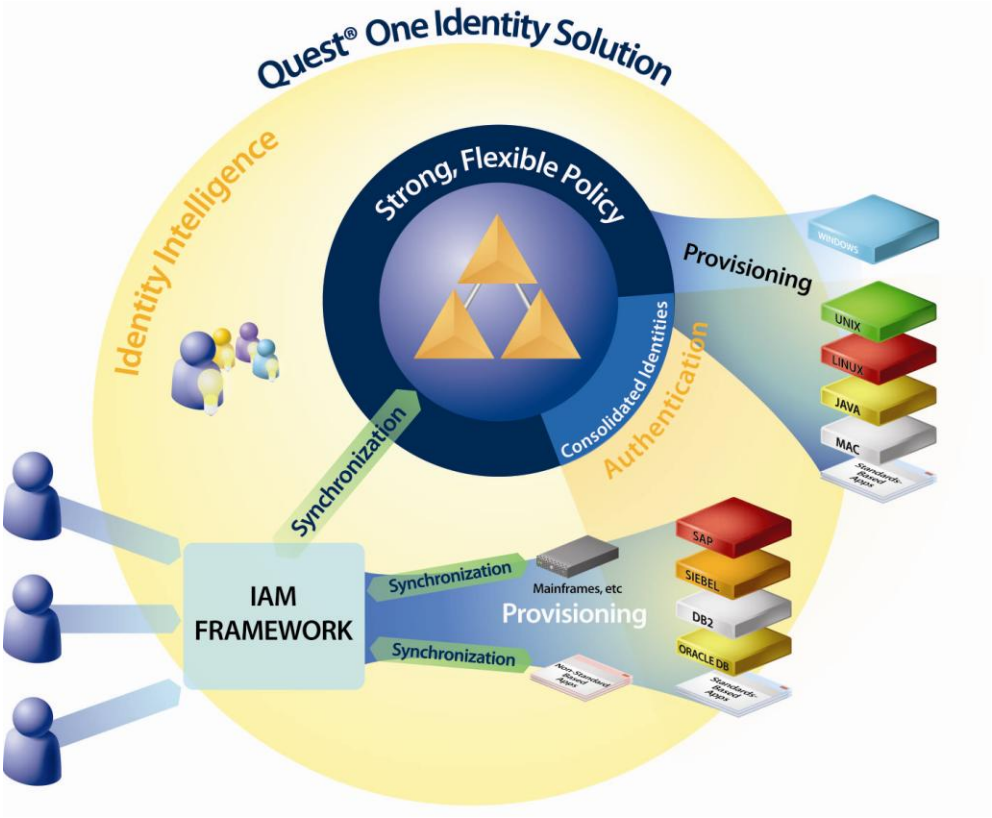


**Figure 7. Quest One accelerates the time-to-value for an IAM framework through reducing complexity and adding identity intelligence**

# Conclusion

Identity and access management is complex, particularly when a diverse IT infrastructure demands multiple identities for each user. This complexity often results in disjointed approaches to role management, password management, and provisioning, as well as “do the best we can with what we have” approaches to multifactor authentication and privileged account management.

A unified and intelligent approach to IAM reduces that complexity, streamlines operations, and adds a layer of control dictated by organizational objectives instead of by IT or technology capabilities. Specifically, this approach:

- Reduces the number of identities associated with any one individual
- Enables provisioning based on business processes and configuration rather than custom-built logic
- Delivers single sign-on ideally suited to each organization’s individual environment, needs, and pressures
- Establishes unified roles, rules, policy, workflow, and attestation processes across systems
- Automates key identity administration tasks such as provisioning, password resets, and audits
- Strengthens security through multifactor authentication
- Eliminates the “keys to the kingdom” problem by granularly delegating administrative access and providing an audit trail of administrative rights and activities
- Consolidates user passwords and offers self-service password resets, enhancing security and improving productivity
- Augments an existing IAM framework for less complexity, which lowers costs, increases control, and speeds time-to-value

## About Quest Software, Inc.

Quest Software (Nasdaq: QSFT) simplifies and reduces the cost of managing IT for more than 100,000 customers worldwide. Our innovative solutions make solving the toughest IT management problems easier, enabling customers to save time and money across physical, virtual and cloud environments. For more information about Quest solutions for application management, database management, Windows management, virtualization management, and IT management, go to **www.quest.com**.

## Contacting Quest Software

PHONE 800.306.9329 (United States and Canada)
If you are located outside North America, you can find your local office information on our Web site.

E-MAIL **sales@quest.com**

MAIL Quest Software, Inc.
World Headquarters
5 Polaris Way
Aliso Viejo, CA 92656
USA

## Contacting Quest Support

Quest Support is available to customers who have a trial version of a Quest product or who have purchased a commercial version and have a valid maintenance contract.

Quest Support provides around-the-clock coverage with SupportLink, our Web self-service. Visit SupportLink at **https://support.quest.com**.

SupportLink gives users of Quest Software products the ability to:

- Search Quest's online Knowledgebase
- Download the latest releases, documentation, and patches for Quest products
- Log support cases
- Manage existing support cases

View the Global Support Guide for a detailed explanation of support programs, online services, contact information, and policies and procedures.

QUEST SOFTWARE®

5 Polaris Way, Aliso Viejo, CA 92656 | PHONE 800.306.9329 | WEB www.quest.com | E-MAIL sales@quest.com
If you are located outside North America, you can find local office information on our Web site.