

# Whitepaper

**bi-Cube<sup>®</sup> SSO**

**Erfahrungsbericht zu einem qualifizierten  
Single Sign-On**

Technologien   Lösungen   Trends   Erfahrung



## Inhalt

<b>1</b>	<b>SUMMARY .....</b>	<b>3</b>
<b>2</b>	<b>QUALIFIZIERTES SINGLE SIGN-ON .....</b>	<b>3</b>
2.1	Eigensicherheit .....	5
2.2	Verfügbarkeit.....	5
2.3	Gina – Integration .....	6
<b>3</b>	<b>SSO BRINGT EIN NEUES SECURITY-PROBLEM .....</b>	<b>6</b>
3.1	Duale Authentifikation.....	6
3.2	Aktive Authentifikation.....	7
3.3	Vier-Augen-Prinzip mit Biometrie .....	7
<b>4</b>	<b>VORTEILE EINER Q-SSO-LÖSUNG.....</b>	<b>8</b>
4.1	Erhöhter Komfort für den Nutzer .....	8
4.2	Kostenreduzierung (vor allem im UHD).....	8
4.3	Deutlicher Gewinn an Sicherheit .....	8
<b>5</b>	<b>WICHTIGE WEITERE FUNKTIONEN EINES Q-SSO .....</b>	<b>9</b>
5.1	Q-SSO-API .....	11
5.2	Anmelde-Technologien .....	11
<b>6</b>	<b>KENNWORT-MANAGEMENT .....</b>	<b>12</b>
6.1	Methoden des Kennwort-Managements.....	12
6.2	Kennwortsynchronisation und Q-SSO-Cluster .....	13
6.3	Q-SSO-Versions-Cluster .....	13
<b>7</b>	<b>Q-SSO VERSUS FEDERATED IDENTITY .....</b>	<b>14</b>

## 1 Summary

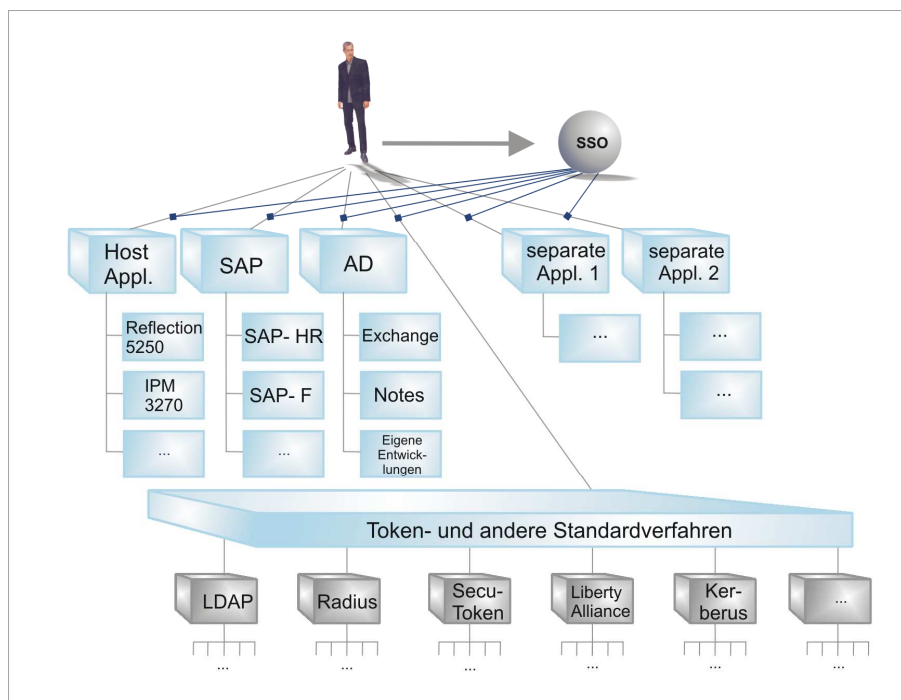
Der Artikel fasst Erfahrungen aus mehreren SSO-Projekten zusammen und stellt vor allem die Anforderungen heraus, die in komplexen IT-Strukturen (und nur diese sind auf SSO-Lösungen angewiesen) unbedingt zu beachten sind, wenn ein SSO-Projekt erfolgreich sein soll. Diese Anforderungen münden in einer neuen Kategorie von SSO-Lösungen, die hier als qualifiziertes SSO (Q-SSO) bezeichnet wird.

**Jeder potentielle Interessent an einer SSO-Lösung sollte sich verdeutlichen, ob er mit einer singulären SSO-Lösung zurechtkommt oder ob seine Gegebenheiten ein Q-SSO erfordern.**

## 2 Qualifiziertes Single Sign-On; Anforderungen und Einsatzszenarien

Die Notwendigkeit, sich mit dem Thema Single Sign-On zu beschäftigen, ergibt sich einerseits aus dem Aspekt der Sicherheitsanforderungen und der Benutzbarkeit der verschiedensten Systeme und Anwendungen mit jeweils eigener Berechtigungsverwaltung.

In einer solchen Systemvielfalt, die sich in den meisten Unternehmen sehr schnell ermitteln lässt, muss ein Nutzer in der Regel 3-6 verschiedene Anmeldeprozeduren mit differenzierten Parametern (User-ID und Kennwort) den Vorschriften entsprechend, im Kopf haben. Da dies sehr schnell unhandlich wird, greift der User zu Hilfsmitteln, die in der Regel gegen die Security-Richtlinien verstoßen.



Ein qualifiziertes SSO kombiniert bzw. optimiert die verschiedenen Möglichkeiten der übergreifenden Accountverwaltung mit dem Ziel, dem Nutzer eine komfortable Anmeldung zu ermöglichen und gleichzeitig die Sicherheit des gesamten IT-Systems zu erhöhen. Dabei übernimmt das Q-SSO auf der obersten Ebene die Anmeldung für den User an die Teilsysteme, für die es keine systemübergreifenden PW-Synchronisationen bzw. Identitätsübergaben gibt.

**Abbildung 1:** Position des Q-SSO in Kombination mit anderen Identity-Management Möglichkeiten

Ein weiteres negatives Ergebnis dieser Systemlandschaft lässt sich jeden Montag oder nach längerer Abwesenheit (Urlaub, Krankheit) im UHD (User-Help-Desk) verfolgen, wenn nämlich die Zahl der Kennwort-Rücksetz-Anforderungen merklich ansteigt.

Vielfach vertrösten die IT-Verantwortlichen auf die geplanten Portal-Entwicklungen, die dies alles auf einen Schlag regeln sollen. Doch selbst unter diesem Aspekt (wenn er denn in endlicher Zeit realisiert wird), verbleiben noch mehrere Anmeldungen bzw. Authentifikationen (LAN, Mail, nicht portalfähige Anwendungen, das Portal selbst, usw.). Selbst wenn ein Portal die Ziel-Architektur ist, bleibt ein Q-SSO eine sinnvolle Ergänzung.

Die hier vorgestellten Lösungen und der zusammenfassende Erfahrungsbericht beziehen sich auf ein Q-SSO-System, das plattformübergreifend mit diversen Zusatzfunktionen sowohl die Security als auch die Benutzbarkeit der IT-Ressourcen erhöhen soll. Die konkrete Lösung geht dabei über diverse andere Ansätze deutlich hinaus, die unter diesem Kürzel, Zusatzfunktionen von Standardsystemen (SAP, Microsoft, Novell,...) anbieten, die aber oft nur eine Synchronisation der Accounts innerhalb einer Plattform realisieren.

Des Weiteren sollten diverse Nebenbedingungen erfüllt und Zusatzfunktionen enthalten sein. Eine Q-SSO-Lösung, die allen diesen Anforderungen gerecht wird, wird als qualifiziertes Q-SSO (Q-Q-SSO) bezeichnet.

Es bestand die Vorgabe, in einer bestehenden heterogenen Umgebung vom LAN auf Basis Win2000 / AD und NT über Unix und Host möglichst alle wesentlichen Anwendungen in einem Q-SSO-Desktop zusammenzufassen. Mit ‚wesentlich‘ sind hier die Applikationen gemeint, die einer hinreichenden Anzahl von Usern (die sog. 80% Grenze) bereitgestellt werden. Oder anders herum, einzelne Spezialanwendungen, die nur von einigen wenigen Usern genutzt werden, wie z.B. ein Expertensystem zur Wertermittlung im Bereich KFZ- Schäden wird z.B. nicht mit erfasst, da nur ca. 10 User diese Anwendung nutzen. Der Zugang zum Schaden-Leistungssystem, das von vielen Sachbearbeitern einer Versicherung genutzt wird, gehört deshalb unbedingt zu den SSO-Systemen.

Das System muss in einer Terminal-Server-Umgebung (z.B. Citrix) nutzbar sein, Funktionen eines Application Launchers besitzen, IKS (Internes Kontroll-System)-Funktionen wie das Vier-Augen-Prinzip, die die Weiterleitung unterstützen und diverse Zusatzfunktionen wie der schnelle Benutzerwechsel, Dateiverschlüsselung und personalisiertes Q-SSO realisieren.

Es muss in einer gemischten LAN-Welt (NT mit ADS /win2000) eine Kennwort-Synchronisation ermöglichen. Eine solche gemischte LAN-Welt ist regelmäßig in großen Unternehmen anzutreffen, deren Migration zu ADS/win 2000 sich über einen längeren Zeitraum erstreckt und die User z.B. schon im ADS verwaltet werden und die Filespaces aber z.T. noch in einer NT-Welt realisiert sind.

Das Q-SSO sollte neben dem primären (zentralen) Profil dem Nutzer die Möglichkeit bieten, sich ein eigenes **personifiziertes SSO-Profil** anzulegen. Damit können nebengelagerte Systeme, die aus der Gesamtsicht des Unternehmens nicht SSO-relevant sind, durch den Nutzer selbst und individuell aufgenommen werden.

Die Q-SSO-Funktionalität muss auch zur Verfügung stehen, wenn der jeweilige Client nicht mit dem Q-SSO-Server bzw. dem Netz verbunden ist. Letzteres ist z.B. für den Außendienst erforderlich. Aus dieser Sicht heraus, muss der Q-SSO-Client die jeweiligen Systemumgebungen erkennen und separat darstellen.

Selbstverständlich muss ein Q-SSO-System eine hohe Eigensicherheit und Verfügbarkeit besitzen.

## 2.1 Eigensicherheit

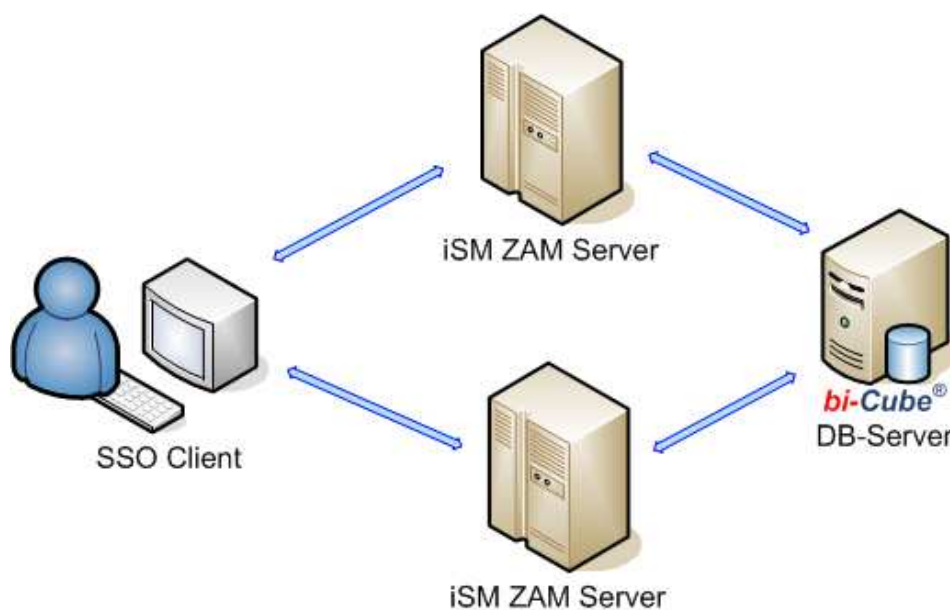
Das gesamte Q-SSO-System, beginnend bei der Rechteverwaltung der Administration, über die Datenspeicherung, den Transport der Anmeldedaten zum Client bis hin zur Speicherung auf dem Client muss einem lückenlosen Security-Konzept unterworfen sein.

## 2.2 Verfügbarkeit

Unter dem Aspekt, dass ein Nutzer sein Kennwort nicht mehr kennen kann, da es automatisch generiert wurde oder dass er es durch die Nutzung des Q-SSO ganz einfach vergessen hat, kommt der Verfügbarkeit der Q-SSO-Funktionalität natürlich ein hoher Stellenwert zu.

Dies wird durch eine Clustering der Q-SSO-Server mittels Load Balancing und einer lokalen Speicherung des Q-SSO-Profiles in hinreichendem Maße erreicht. Verbunden muss dies mit einem äußerst stabilen Q-SSO-Client sein.

Zur Verfügbarkeit gehört auch das Kriterium der richtigen Funktion des Q-SSO-Clients. Das heißt, er muss immer in der Lage sein, die Anmeldung an Stelle des Users richtig auszuführen. Hier ist das qualifizierte Kennwort-Management ein zu lösendes Problem, das dadurch erschwert wird, dass es für fast jedes Zielsystem individuell zu lösen ist. (vergl. dazu den Teil Kennwort-Management)



**Abbildung 2:** Load Balancing und Applications-Cluster eines Q-SSO

### 2.3 Gina – Integration

Ein Q-SSO muss mit der Anmeldung an das System zusammen spielen. Die Funktionen der Gina (Graphical Identification and Authentication) und des SSO sind in einem Q-SSO miteinander optimal abgestimmt. Deshalb kommt ein Q-SSO regelmäßig mit einer eigenen Gina zur Anwendung. Dies ist insbesondere zur Eliminierung der im Folgenden beschriebenen Sicherheitsproblematik von Bedeutung.

## 3 Führt SSO zu einem neuen Security Problem?

Vielfach wird der Einsatz einer SSO-Lösung abgelehnt, da dadurch ein neues Security-Problem generiert wird.

Die Argumentation geht dahin, dass bei verschiedenen Accounts (User-ID und Kennwort) das Risiko eines „geknackten“ Zugangs sich immer nur auf das konkrete System beschränkt und nicht auf alle Systeme gleichzeitig, die durch das SSO erfasst werden. Hierfür muss ein Q-SSO-System eine entsprechende Lösung anbieten! Diese Lösung besteht zumindest aus den beiden folgenden Komponenten:

### 3.1 Duale Authentifikation

Für ein Q-SSO ist ein gesicherter Zugang zum SSO-Client zu schaffen. Hierzu gibt es verschiedene Ansätze, die unter dem Begriff „Hochsicherheits-Authentifikation“ zusammengefasst werden. Dazu gehören alle Arten der dualen Authentifikation und vor allem die Biometrie.

### 3.2 Aktive Authentifikation

Die Biometrie hat außerdem den Vorteil der aktiven Authentifikation. Dies bedeutet, dass sie zu jedem Zeitpunkt abgefordert werden kann und immer die direkte Anwesenheit des Users erfordert, was bei einigen anderen dualen Verfahren nicht immer gegeben ist. Ein duales Verfahren nützt wenig, wenn es auch in Abwesenheit des Users (z.B. durch das Abfragen eines Tokens) aktiviert werden kann.

Die aktive Authentifikation ist in einem gesicherten Q-SSO durch die Modellierung vor jede besonders zu sichernde Anwendung „vorzuschalten“. Dies bedeutet, dass der Start einer solchen Anwendung im SSO-Client vorher die aktive Authentifikation des Users erfordert. Er muss also im Moment des Startens persönlich anwesend sein und entweder eine biometrische Authentifikation vornehmen oder zumindest sein LAN-Passwort noch einmal aktiv eingeben, bevor die eigentliche Anmeldung startet. Dieses Verfahren muss natürlich auch bei der SSO-Konfiguration selbst zur Anwendung kommen. (s. Bild)



**Abbildung 3:** Biometrie-Authentifikation VOR dem Start einer zu sichernden Anwendung durch das Q-SSO

### 3.3 Vier-Augen-Prinzip mit Biometrie

Unter Anwendung der Biometrie- und Q-SSO-Technologien können Anforderungen nach einer Zusatzfreigabe innerhalb eines Geschäftsprozesses (z.B. Freigabe von höheren Auszahlungen) eingefügt werden, ohne dass in die Anwendung eingegriffen werden muss. Der Freigebende nutzt ein Biometrie-Device, womit dann in **bi-Cube®** das Kennwort für die Freigabe ermittelt wird. Dieses wird dann mit der Anmeldetechnologie von Q-SSO in das Kennwort-Eingabefeld automatisch eingefügt. Dies ohne, dass der Freigebende dieses Kennwort kennen muss.

## 4 Vorteile einer Q-SSO- Lösung

### 4.1 Erhöhter Komfort für den Nutzer

Dieser Effekt ist unbestritten und bei einem sicher funktionierenden SSO durch eine hohe Akzeptanz der User belegt.

### 4.2 Kostenreduzierung (vor allem im UHD)

Dieser Faktor wird von allen Anbietern als ein primärer Effekt angegeben. Die Angaben zu den Einsparungen im UHD schwanken hier je nach Mut des Anbieters zwischen 30% und 60%. Auf alle Fälle wird durch ein Q-SSO eine wesentliche Reduzierung der UHD-Calls erreicht.

Merklich wird dieser Effekt allerdings dadurch gesteigert, wenn mit dem Q-SSO flankierende IPM-Maßnahmen zum Einsatz kommen (IPM = Identity- and Provisioning-Management)

Hierzu zählen vor allem:

- Verfahren des Kennwort- Self Service
- Server-getriggerte Aktualisierung des lokalen Profils
- keine Information des Users bei Zuteilung neuer Software (Start-Account)

Zusätzlich ist zu sehen, dass ein Q-SSO durch den Wegfall der Arbeits-Ausfälle des Nutzers die durch verhinderte Systemzugänge bedingt sind, die Verfügbarkeit der IT-Infrastruktur insgesamt verbessert.

Ein Q-SSO kann außerdem wesentlich zur Reduzierung der Lizenzkosten beitragen

### 4.3 Deutlicher Gewinn an Sicherheit

Ein qualifiziertes SSO (das u.a. obige Aspekte der Eigensicherheit usw. berücksichtigen muss) führt zu einer deutlichen Erhöhung des Security-Niveaus.

- Der User muss nicht mehr gegen Richtlinien der Handhabung von Kennwörtern verstoßen.
- Die Kennwort-Regeln der einzelnen Systeme können sehr hoch angesetzt werden, da der User sich die Kennwörter nicht mehr merken muss.
- Die allgemein bekannte Sicherheitslücke der Übermittlung von Start-Kennwörtern (per Mail oder Telefon) entfällt völlig.

## 5 Wichtige weitere Funktionen eines Q-SSO

Neben den oben bereits genannten funktionellen Anforderungen, wie z.B. gemischter Betrieb unter Citrix gehören folgende Funktionen zu einem Q-SSO:

### 1. Schneller Benutzerwechsel

Durch den schnellen Benutzerwechsel kann die Wartezeit beim Ummelden des Benutzers am Betriebssystem vermieden werden. Dazu meldet sich am Betriebssystem ein nicht personalisierter Nutzer und am SSO-Client eine Person an. Voraussetzung ist, dass die Anwendungen keine persönlichen Verzeichnisse benötigen. Diese Anforderung ergab sich vor allem beim Einsatz in Krankenhäusern und im operativen Bankbetrieb.

### 2. Erweiterte Logmöglichkeiten durch das Q-SSO-Event-Management

Mit dem erweiterten Log ist es möglich, den Start des SSO-Clients und den Start von Anwendungen aus dem SSO-Client heraus in der Datenbank mitzuloggen. Diese Datensätze können später über den Report aufgelistet und ausgewertet werden und dann z.B. zur Lizenzoptimierung genutzt werden. Der SSO-Client eines Q-SSO sollte für das Event-Management folgende Funktionen bieten:

1. Logging der Anmelde- Abmeldezeiten an das Q-SSO
2. Logging der Anmeldezeiten an ein System (zur Lizenzoptimierung)

Die Events lassen sich abhängig von den Auswertezielen aktivieren:

- Starten des Q-SSO-Clients
- Beenden des Q-SSO-Clients
- Aktualisieren des Q-SSO-Clients
- Starten einer Applikation durch das Q-SSO

Standard-Auswertungen können nach folgenden Punkten erfolgen:

- Analyse nach Useranzahlen / Usern die ein ausgewähltes System länger als x Tage nicht genutzt haben
- Durchschnittliche Nutzungsrate (x mal in 30 Tagen) eines Systems,
- und der Rollen, die dieses System zugeordnet haben

### 3. Reaktion auf LAN-Verfügbarkeit

Beim Start des SSO-Clients wird die Verfügbarkeit des SSO-Servers getestet. Ist er nicht verfügbar, erfolgt eine Abfrage, welche Anwendungsprofile (persönlich und/oder lokal) weiter benutzt werden sollen. Bei manueller Aktualisierung wird erneut getestet und ggf. der online-Zustand wiederhergestellt.

### 4. Reaktion auf Netzgeschwindigkeit

Auf verminderte Netzgeschwindigkeit (z.B. bei VPN) wird durch Weglassen von zeitaufwendigen, aber nicht unbedingt notwendigen Aktionen (Test der Existenz von Dateien, Lesen der Icons) reagiert.

### 5. Automatische Aktualisierung des Anwendungsprofils (Trigger)

Bei Änderung des Profils in der IPM-Datenbank wird eine Benachrichtigung an den SSO-Client gesendet, damit dieser das Anwendungsprofil aktualisiert. Hat der entsprechende Nutzer aktuell keinen SSO-Client gestartet, so wird die Benachrichtigung vorgemerkt. Damit erübrigt sich die Security-lastige Übermittlung von Start-Accounts an den User.

## 6. User-Selbst-Registrierung

Die **User-Selbst-Registrierung** (USR) ermöglicht dem Nutzer, dem Q-SSO-System einen bestehenden Account zu übergeben, der diesem noch nicht bekannt ist. Dieses Verfahren kann zur sanften Migration der Accounts in das Q-SSO-System genutzt werden. Dies ist oft die einzige Lösung, erstmalig das Kennwort eines Users für eine Applikation dem Q-SSO-System bekannt zu geben. Dabei wählt der User ein durch das Q-SSO verwaltetes System aus, zu dem er meint, einen Zugang zu besitzen. Durch Eingabe von User-ID und Kennwort ist das Q-SSO-System in der Lage, eine automatisierte Anmeldung (im Namen dieses Users) vorzunehmen. Wenn diese Anmeldung erfolgreich ist, was durch das Q-SSO-System zu überprüfen ist, werden die Anmeldedaten im Q-SSO-Server auf gültig gesetzt und stehen damit zur Verwendung bereit.

## 7. Verdecktes SSO

Mit dem **Verdeckten SSO** können irgendwann im Verlauf der Nutzung einer Applikation erforderliche Authentifikationen realisiert werden. Es geht also nicht darum, den Start der Anmeldung durch ein automatisches Logon zu unterstützen, sondern innerhalb einer Anwendung erforderliche Authentifizierungen vorzunehmen. Diese können in einer Anwendung mehrfach und unterschiedlich erforderlich sein (z.B. e-Bay). Bei Notes kann der User z.B. mit F5 seinen Desktop sperren und muss diesen bei Weiterarbeit durch Eingabe des Kennwortes wieder entsperren. Da ihm im Regelfall das Kennwort aber nicht mehr bekannt ist, muss dies auch durch ein Q-SSO geleistet werden.

## 8. Varianten durch verschiedene Betriebssysteme

Im Gegensatz zu Programmversionen, die unterschiedliche SSO-Funktionalitäten bedingen und die bereits auf dem Server selektiert werden können, müssen alle betriebssystemabhängigen Inputstrings eines Q-SSO-Systems auf dem Client zur Verfügung stehen, da nur dort entschieden werden kann, welches Betriebssystem der Client hat. Dies bedeutet, dass der SSO-Client unterscheiden muss auf welcher Windowsversion er aktuell läuft.

## 9. Weitere optionale Funktionen eines Q-SSO

- unterschiedliche Gültigkeitsdauer des Kennwortes für jedes System
- Widerruf der Zuordnung eines Systems bei Nichtanmeldung an das System innerhalb von x Tagen auf Ebene des Q-SSO (Revoke)
- systemspezifische Gültigkeitsdauer der lokalen Anmeldung ohne Verbindung zum Server (wichtig für Außendienst)
- systemspezifische Angabe, ob ein System per Remote Access genutzt werden darf
- systemspezifische Angabe, ob ein System im Q-SSO gleich per Autostart genutzt werden soll
- Über Gültig ab / Gültig bis kann je System festgelegt werden, ab wann bzw. bis wann es im Q-SSO zur Verfügung steht und dies unabhängig von der Zuordnung des Systems. Damit kann z.B. ein System durch einen einzigen Eintrag aus der Testphase für alle User bereitgestellt werden bzw. der Zugriff wieder unterbunden werden.
- verschiedenste User - Aktionen können ein Alerting an eine je System definierte Mailadresse auslösen.
- Es kann für jedes System angegeben werden, von welchem System das Kennwort „vererbt“ wird.
- Trennung der Systeme nach Systemumgebung (Application Launcher)

## 5.1 Q-SSO-API

Für die wichtigsten Steuer- und Datentransfer-Operationen stellt das Q-SSO-System eine API zur Verfügung, um das System möglichst effektiv in vorhandene Verwaltungssysteme zu integrieren. Dies ist insbesondere dann erforderlich bzw. sinnvoll, wenn das Q-SSO nicht in ein IPM integriert ist, sondern ein Provisioningsystem eines anderen Herstellers oder unternehmenseigene Verwaltungssysteme genutzt werden.

## 5.2 Anmelde-Technologien

Die Bereitstellung verschiedener Technologien zur Realisierung der automatischen Interaction des SSO-Systems mit der jeweiligen Applikation, um eine möglichst breite Palette von Anwendungen zu integrieren. Dies wird als horizontale Integrationsfähigkeit bezeichnet. Ein Q-SSO-System sollte dem Administrator mindestens folgende Anmeldetechnologien zur Verfügung stellen:

### **Security-Token**

Die eleganteste und sicherste Methode der automatisierten Anmeldung eines Users an ein System mit eigener Berechtigungsverwaltung erfolgt mittels eines Security-Tokens (von manchen Herstellern auch als Zertifikat bezeichnet). Dabei generiert der SSO-Client ein Einmal-Kennwort, das noch einige weitere Parameter (User-ID, Timestamp, System-ID des aufrufenden Systems) verschlüsselt enthält. Dieses Token hat eine begrenzte Lebensdauer (Standard =5 min) und wird vom Zielsystem wieder in seine einzelnen Parameter zerlegt und geprüft. Wenn die Laufzeit (5 min) überschritten oder das aufrufende System dem Zielsystem nicht bekannt ist, wird das Token abgewiesen.

### **Anmeldung über ein API**

Diese Technologie der automatischen Useranmeldung ist die sicherste Methode und wenn möglich, die Methode der Wahl. Dabei bietet die Applikation ein Programm-Interface, das sowohl die Nutzeranmeldung als auch die Verwaltung der Logon-Daten im Zielsystem ermöglicht. Typische Beispiele hierfür sind (SAP und 3270).

### **Simulation der User-Anmeldung**

Bei diesem Verfahren werden Windows-Funktionen genutzt, um die erforderlichen Daten direkt in die Eingabefenster „reinzuschreiben“. Dieses Verfahren erfordert ein mehr oder weniger kompliziertes Scripting, sicheres Erkennen der Fenster, die eine Eingabe erwarten und eine sinnreiche Zeitsteuerung (Wie lange wartet der Q-SSO-Client auf ein Fenster bzw. ab wann wird das Warten so interpretiert, dass die Anwendung nicht gestartet ist.)

Anwendungen mit variablem Fenstertitel und mehr-schrittige Anmeldeprozeduren komplizieren diesen Prozess merklich. Die Erstellung der Scripte sollte unbedingt durch einen Scanner unterstützt werden, der die Anmeldung an das Zielsystem „mitschreibt“, das Script automatisch generiert und in die Datenbank zu den Systemdaten ablegt.

Für den Internet-Browser ist ein separater Scanner erforderlich, der dann noch verschiedene Darstellungstechnologien (HTML, Java,...) der jeweiligen Anmelde-GUI unterstützen muss. Eine „Unterart“ dieser Technologie ist die Simulation der Tastatur, die immer dann zur Anwendung gelangt, wenn alle anderen Verfahren nicht einsetzbar sind.

## 6 Kennwort-Management

Wesentliches Qualitätsmerkmal eines Q-SSO ist die Leistungsfähigkeit des Kennwort-Managements, das sich zwangsläufig dadurch ergibt, dass der User sein Kennwort nicht mehr kennt. Kennwort-Management ist dann erforderlich, wenn das Kennwort im SSO und im Zielsystem nicht mehr synchron ist. Dies kann nicht völlig ausgeschlossen werden, wenn im Zielsystem der Kennwort-Wechsel durch den User selbst vorgenommen werden kann. In diesem Fall muss dem User die Synchronisation möglich sein.

Die **SSO-Synchronisation** ermöglicht es dann, dem User ein durch ihn evtl. verändertes Kennwort dem SSO wieder bekannt zu geben. Als letzte Lösung bleibt immer noch das Rücksetzen des Kennwortes durch SSO-Funktionalität.

Weiterhin ist Kennwort-Management für Systeme erforderlich, die einen Kennwort-Wechsel erzwingen und es nicht möglich oder vertretbar ist, diesen abzustellen. Das Kennwort-Management soll vom Q-SSO-System so weit als irgend möglich unterstützt werden. Das bedeutet, dass das Q-SSO-System es dem Nutzer abnimmt, in definierten Zeitabständen sein Kennwort zu wechseln

Das Kennwort-Management ist vor allem durch die Spezifität jedes einzelnen Systems gekennzeichnet und dadurch natürlich auch recht kompliziert in Hinblick auf eine umfassende Realisierung. Deshalb ist ein Q-SSO dadurch gekennzeichnet, dass es eine breite Palette von technischen Möglichkeiten für das Kennwort-Management anbietet.

### 6.1 Methoden des Kennwort-Managements

#### Manueller Wechsel (Methode 0)

Der Nutzer wechselt selbständig im Anwendungsclient das Kennwort. War der Wechsel erfolgreich, muss er das neue Kennwort über den SSO-Client auch in **bi-Cube<sup>®</sup>** eintragen. Diese Methode ist sicher nur als Übergang zu einer Q-Lösung sinnvoll aber mit jeder Anwendung möglich.

#### Manueller Kennwortwechsel mit automatischem Eintrag im IPM (Methode 1)

In dieser Variante des manuellen Kennwortwechsels erkennt der SSO-Client das Auftreten des Kennwortwechselfensters und schneidet die Eingaben des Benutzers mit. Der SSO-Client trägt dann das neue Kennwort automatisch in **bi-Cube<sup>®</sup>** ein.

#### Dezentraler Kennwortwechsel (Methode 2)

Der SSO-Client überwacht die Kennwortgültigkeit. Droht ein Kennwort abzulaufen, löst der SSO-Client den Kennwortwechsel aus. Dazu wird ein neues Kennwort erzeugt und dann die Nutzeraktionen im Anwendungsclient simuliert.

#### Dezentraler Kennwortwechsel über API (Methode 3)

Diese Methode entspricht der Methode 2. Nur wird hier die Eintragung des neuen Kennworts über die API der Anwendung vorgenommen. Die Nutzung der API des Anwendungsprogramms ist jeder anderen Methode vorzuziehen, da sie die höchste Sicherheit in der Funktionalität bietet. Schwierigkeiten ergeben sich mitunter dadurch, dass der Client mit der Anwendung in einem System-Kontext kommunizieren muss, um die Funktion des Kennwort-Wechsels ausführen zu dürfen.

### Zentraler Kennwortwechsel (Methode 4)

Das Kennwort wird vom zentralen Passwort Manager gewechselt. Über den Nachrichtenraum und einen entsprechenden Output-Connector erfolgt der Wechsel in der Anwendung. Nach Bestätigung durch die Anwendung wird das neue Kennwort auch in **bi-Cube<sup>®</sup>** eingetragen bzw. gültig gemacht und ist damit für das SSO nutzbar. Der Nutzer wird im Normalfall nicht vom Wechsel in Kenntnis gesetzt.

Voraussetzung ist ein Output-Connector für die Anwendung, der in der Lage ist, eine Kennwortänderung durchzuführen. Die Gültigkeitsdauer in **bi-Cube<sup>®</sup>** muss so eingestellt werden, dass der Passwort-Manager dem Ablauf des Kennwortes in der Anwendung zuvorkommt oder der Verfall der Kennworte kann in der Anwendung abgeschaltet werden.

Diese Methode zeichnet ein Q-SSO besonders aus, da sie sicher und für den Nutzer transparent ist. Sie wird unabhängig davon, ob der Nutzer irgendwann angemeldet ist oder nicht, ausgeführt. Diese Möglichkeit ist nur in solchen Systemen gegeben, die einen integrierten IPM-SSO-Architekturansatz verfolgen. Derzeit bietet dies nur die **bi-Cube<sup>®</sup>**-Lösung des iSM.

## 6.2 Kennwortsynchronisation und Q-SSO-Cluster

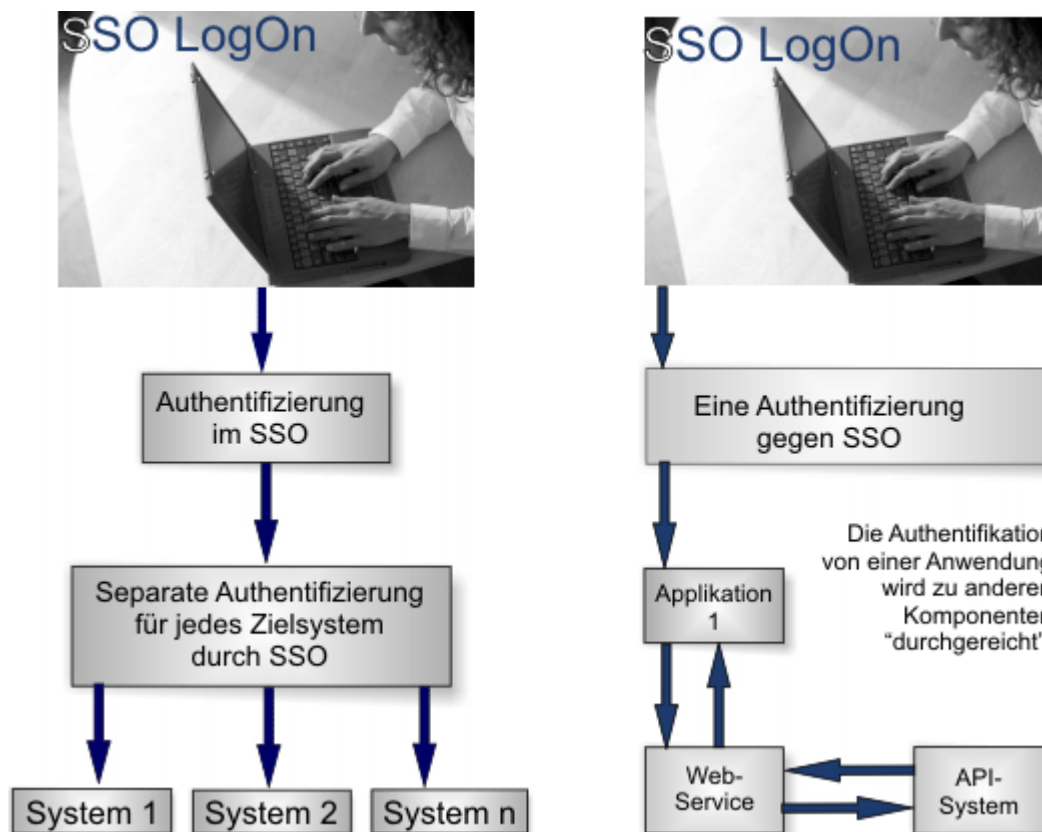
Eine generelle Kennwort-Synchronisation anstelle einer Q-SSO-Lösung ist aus Sicherheitsgründen prinzipiell ungeeignet. Innerhalb der Q-SSO-Lösung können jedoch System-Cluster gebildet werden, zwischen denen eine Kennwort-Synchronisation vorgenommen werden kann und sinnvoll ist. Beispielsweise, zwischen allen Komponenten einer Produktfamilie, die sowieso gleichen Kennwort-Regeln unterliegen.

## 6.3 Q-SSO-Versions-Cluster

Eine spezielle Variante der Q-SSO-Cluster ergibt sich dann, wenn eine Applikation in verschiedensten Versionen verwaltet und zugeteilt werden kann. Ein Beispiel ist Notes, das in unterschiedlichen Releases und Sprachversionen existiert und den Nutzern zugeteilt werden kann. Dieser Fall kann nur in einer realistischen Weise durch die enge Integration des SSO mit einer IPM-Lösung beherrscht werden. In diesem Fall wird Notes einmal als System und die Versionen / Varianten als Applikationen im IPM definiert, wobei der Account nur an das System gebunden ist; die Kennwort-Verteilung sich allerdings nach der Applikation richtet. Die Modellierung als Cluster sichert immer die richtige Anmeldung und erlaubt vor allem den Versionswechsel ohne Änderung der Accountdaten.

## 7 Q-SSO versus Federated Identity

Eine SSO-Lösung ist grundsätzlich eine Notlösung für fehlende einheitliche Architekturkonzepte die eine einmalige und einheitliche Authentifizierung über ALLE Anwendungen ermöglichen. (vergl. auch Bild 1). Für bestimmte Bereiche und Applikationsgruppen (z.B. eines Herstellers wie SAP) wird intern eine zentralisierte Identität bereitgestellt und von allen Anwendungen, die diese kennen, auch nutzbar gemacht. In der Ebene darüber bleibt aber immer (d.h. zumindest für absehbare Zeiten) ein Bereich von Anwendungen übrig, die nur durch ein SSO in Richtung einer Anmeldung konsolidierbar sind. Aus dieser Sicht bleiben SSO und Federated Identity zwei Wege und teilweise auch 2 Ziele.



**Abbildung 4:** Gegenüberstellungen der Funktion eines SSO und der Federated Identity