

**Rollenbasierte Zugriffssteuerung
(Role Based Access Control)
für .NET-Anwendungen**

Strategien für Authentifizierung und Berechtigungen

1	ZIEL DIESES DOKUMENTS	3
2	HAUPTFUNKTIONEN	3
2.1	AUTHENTIFIZIERUNG	3
2.2	AUTORISIERUNG.....	3
2.3	AUDIT	3
3	BASISKOMPONENTEN.....	4
3.1	SICHERES REPOSITORY ZUR SPEICHERUNG VON RBAC-DATEN.....	4
3.2	INTEGRIERTE ANWENDUNGSKOMPONENTE	4
3.3	VERWALTUNGSKONSOLE.....	4
3.4	DOKUMENTATION FÜR ENTWICKLER UND ADMINISTRATOREN.....	4
4	BERECHTIGUNGSENTWICKLUNG UNABHÄNGIG VOM ANWENDUNGSCODE	5
4.1	WAS BEDEUTET DAS?	5
4.2	ASPEKTE DER LEBENSDAUER	5
4.3	UNABHÄNGIGKEIT FÜR ADMINISTRATOREN.....	5
4.4	ENTLASTUNG FÜR DAS ENTWICKLUNGSTEAM	5
5	SCHLÜSSELFRAGEN FÜR DIE WAHL IHRES RBAC-SYSTEMS.....	6
5.1	BENÖTIGEN SIE EIN ZENTRALES SICHERHEITSREPOSITORY FÜR ANWENDUNGEN?.....	6
5.2	BENÖTIGEN SIE GEMEINSAM VERWENDETE ROLLEN?.....	6
5.3	BENÖTIGEN SIE UNTERSTÜTZUNG FÜR EINMALIGES ANMELDEN (SINGLE SIGN-ON)?	6
5.4	BENÖTIGEN SIE UNTERSTÜTZUNG FÜR MEHRERE TECHNOLOGIEN?.....	6
5.5	BENÖTIGEN SIE BERICHTE ZU BENUTZERKONTEN UND BERECHTIGUNGEN?.....	6
5.6	BENÖTIGEN SIE UNTERSTÜTZUNG FÜR AUDIT-ANFORDERUNGEN (SOX, ISO,...) ?.....	6
5.7	BENÖTIGEN SIE UNTERSTÜTZUNG FÜR EIN GROBES / INTERNATIONALES ENTWICKLUNGSTEAM?.....	6
5.8	BENÖTIGEN SIE EINE FEINE BERECHTIGUNGSGRANULARITÄT?	6
6	UNTERSCHÄTZTE KOSTEN FÜR INSTANDHALTUNG.....	8
6.1	UNTERSTÜTZUNG FÜR ENTWICKLER UND ADMINISTRATOREN	8
6.2	KONSISTENTE VERWALTUNG VON BERECHTIGUNGEN UND ANWENDUNGSCODE	8
6.3	BEREITSTELLUNG NEUER ANWENDUNGSVERSIONEN	8
6.4	VERSIONIERUNG VON SICHERHEITSDATEN	8
7	BETRIEBSINTERNE ENTWICKLUNGEN ALS SICHERHEITSLÖSUNGEN	9
7.1	ENTWICKLUNG UND SUPPORT HÄNGEN AB VON INTERNEN RESSOURCEN.....	9
7.2	ANWENDUNGSSPEZIFISCHE BENUTZERKONTEN	9
7.3	GERINGE GRANULARITÄT BEI ZUGRIFFEN UND BERECHTIGUNGEN	9
7.4	SPEZIFISCHER ANWENDUNGSCODE.....	9
7.5	INSTANDHALTUNGSASPEKTE BEI SEPARATEN ANWENDUNGSLÖSUNGEN.....	9
8	VISUAL GUARD ALS UNTERNEHMENSLÖSUNG FÜR ANWENDUNGSSICHERHEIT	10
8.1	EINHEITLICHE LÖSUNG FÜR SICHERUNG UND ZENTRALISIERUNG ALLER .NET-ANWENDUNGEN	10
8.2	VOM PROGRAMMCODE UNABHÄNGIGE BERECHTIGUNGEN	10
8.3	VERWALTUNGSWERKZEUGE FÜR NICHTTECHNISCHES PERSONAL	10
8.4	WERKZEUGE ZUR BERECHTIGUNGSVERWALTUNG, VERSIONIERUNG UND BEREITSTELLUNG.....	10
8.5	AUDIT-WERKZEUGE FÜR BERECHTIGUNGSBERICHTE UND ANWENDUNGSPROTOKOLLPRÜFUNGEN.....	10
8.6	STANDARDPRODUKT MIT PROFESSIONELLEM SUPPORT UND STÄNDIGEN AKTUALISIERUNGEN.....	11

1 Ziel dieses Dokuments

Dieses Dokument bietet Informationen für den Entwurf und die Umsetzung eines Systems zur rollenbasierten Zugriffssteuerung (Role Based Access Control, RBAC).

2 Hauptfunktionen

Ein RBAC-System bietet drei Arten von Leistungsmerkmalen: Authentifizierung, Autorisierung und Audit.

2.1 Authentifizierung

Dieses Leistungsmerkmal dient zur Bestätigung der Identität eines Benutzers. Hiermit wird sichergestellt, dass nur identifizierte Personen eine Anwendung nutzen können. Das Verfahren besteht aus zwei Schritten. Im ersten Schritt, der Identifikation, gibt der Benutzer an, wer er ist. Im zweiten Schritt, der Authentifizierung, wird sichergestellt, dass die Identifikation korrekt ist. Dies geschieht üblicherweise über Benutzerkonten und Kennwörter. Dies ist die erste Ebene der Sicherheit.

2.2 Autorisierung

Autorisierungen definieren, was ein Benutzer in einer Anwendung tun kann. Sie definieren damit grundsätzlich, was ein Benutzer in einer Anwendung sehen, tun und ändern darf.

Bei der Definition von Autorisierungen entscheiden Sie zwischen zwei Vorgehensweisen:

- Der sicherste Ansatz besteht darin, standardmäßig alles zu verbieten und dann gezielte Berechtigungen für gewollte Aktionsmöglichkeiten zu erteilen. Sollten Sie bei diesem Ansatz einmal vergessen, eine Berechtigung zu definieren, kann ein Benutzer etwas Gewolltes nicht tun. Dafür kann der Benutzer aber auch nicht zufällig etwas tun, was nicht gewollt ist.
- Der schnellere Ansatz besteht darin, standardmäßig alles zu erlauben und dann gezielte Beschränkungen für ungewollte Aktionsmöglichkeiten zu erteilen. Dieser Ansatz ist schneller, da es meist weniger Beschränkungen als Berechtigungen gibt.

Dies ist die zweite Ebene der Sicherheit. Sie bildet den umfangreichsten Teil eines RBAC-Entwurfs, da jede Berechtigung/Beschränkung codiert werden muss.

2.3 Audit

Audits basieren auf der Protokollierung und Prüfung kritischer Anwendungstransaktionen. Audit-Funktionalitäten können erforderlich sein, um unternehmensinterne Vorschriften, gesetzliche Anforderungen (z.B. SOX) oder Zertifizierungsverfahren (z.B. ISO) einzuhalten.

Audit-Funktionalitäten sollen Ihnen Prüfungen dahin gehend ermöglichen, **wer wann was** in einer Anwendung tat, und wer welchem Benutzer welche Berechtigung erteilte.

3 Basiskomponenten

RBAC-Systeme für Unternehmensanwendungen besitzen folgende Komponenten:

3.1 Sicheres Repository zur Speicherung von RBAC-Daten

Benötigt wird ein sicherer Verwahrungsort, um Benutzernamen, Kennwörter, Rollen und Berechtigungen zu speichern.

3.2 Integrierte Anwendungskomponente

Diese Komponente kommuniziert mit dem RBAC-Repository und passt eine Anwendung entsprechend der Benutzerberechtigungen an.

3.3 Verwaltungskonsole

Diese Anwendung ist für nichttechnisches Personal ausgelegt, um Benutzerkonten zu verwalten und Berechtigungen zu erteilen. Sie bietet eine benutzerfreundliche Oberfläche, um die Information auf einfache Weise zu verwalten und Entwickler von der täglichen Zugriffsverwaltung des Systems zu entlasten.

3.4 Dokumentation für Entwickler und Administratoren

Eine Dokumentation sollte allen Mitarbeitern zur Verfügung stehen, die am Sicherheitsverfahren der Anwendung beteiligt sind . Sinnvoll sind beispielsweise ein Integrationshandbuch, ein Benutzerhandbuch, FAQ usw.

4 Berechtigungsentwicklung unabhängig vom Anwendungscode

4.1 Was bedeutet das?

Damit Berechtigungen unabhängig vom Anwendungsprogrammcode sind, dürfen sie nicht in diesem definiert werden. Es kann hierfür notwendig sein, einen Aufruf im Programmcode einer Komponente einzufügen, um das RBAC-System zu aktivieren. Aber dieser Aufruf stellt nicht die Berechtigung selbst dar.

Wird dagegen der Programmcode, der die Berechtigungen definiert, mit dem Programmcode der Anwendung vermengt, können langfristig gravierende Instandhaltungsprobleme auftreten.

4.2 Aspekte der Lebensdauer

Es kann häufiger erforderlich sein, die Einstellungen der Zugriffssteuerung zu aktualisieren, beispielsweise für neue Benutzerkonten und Berechtigungsänderungen. Dagegen erfolgen Änderungen an einer Anwendung unregelmäßig, und eine vorhandene Anwendungsversion kann viele Monate im Einsatz sein. In der Regel ist die Lebensdauer einer Berechtigung deutlich kürzer als die Lebensdauer einer Anwendung. Werden Berechtigungen mit dem Programmcode einer Anwendung vermengt, müssen Endbenutzer auf eine neue Anwendungsversion warten, um neue Berechtigungen nutzen zu können.

Die Definition oder die Erteilung von Berechtigungen sollte nicht von neuen Anwendungsversionen abhängig sein. Sie sollten in der Lage sein, eine Berechtigung hinzuzufügen oder zu erteilen, ohne den Anwendungscode zu ändern (wofür ein voller Entwicklungszyklus erforderlich wäre: Programmieren, Testen, Debuggen, Bereitstellung, ...). Werden neue Berechtigungen dynamisch berücksichtigt (anstelle fest in der Anwendung codiert zu sein), können Sie neue Berechtigungen definieren und erteilen, während die Anwendung im Einsatz ist. Und diese werden sofort wirksam!

4.3 Unabhängigkeit für Administratoren

Lokale Administratoren kennen die Endbenutzer und ihre Berechtigungen, besonders bei entfernten Standorten. Sie benötigen ein benutzerfreundliches Administrationsprogramm, um Benutzerkonten und Berechtigungen zu verwalten, ohne vom Entwicklungsteam abhängig zu sein. Dies ist nur möglich, wenn das Hinzufügen neuer Berechtigungen keine Änderung des Anwendungsprogrammcodes erfordert.

4.4 Entlastung für das Entwicklungsteam

Wenn sich die Endbenutzer und Administratoren selbst um die tägliche Verwaltung von Benutzerkonten und Berechtigungen kümmern, kann sich das Entwicklungsteam auf die Entwicklung konzentrieren.

5 Schlüsselfragen für die Wahl Ihres RBAC-Systems

5.1 Benötigen Sie ein zentrales Sicherheitsrepository für Anwendungen?

Wollen Sie alle Ihre verschiedenen Anwendungen auf Basis eines gemeinsamen Repositories zentralisieren? Sollen Administratoren einen Überblick über alle Anwendungen und Benutzer Ihres Unternehmens haben?

5.2 Benötigen Sie gemeinsam verwendete Rollen?

Eine gemeinsam verwendete Rolle kann für mehrere Anwendungen genutzt werden, wogegen eine spezifische Rolle sich nur für eine Anwendung eignet. Gemeinsam verwendete Rollen sind besonders nützlich, wenn die Mitarbeiter Ihres Unternehmens mit mehreren Anwendungen arbeiten. Es ist dann nicht notwendig, für jede Anwendung eine Rolle zu verwalten, auf die ein Benutzer zugreifen kann.

5.3 Benötigen Sie Unterstützung für Einmaliges Anmelden (Single Sign-On)?

Nutzen Sie ein Active Directory zur Verwaltung von Benutzerkonten, ist meist ein Sicherheitsverfahren zur einmaligen Anmeldung sinnvoll, die aus Sicht der Endbenutzer den größten Komfort bietet. Sobald sich ein Benutzer einmal an einer Windows-Sitzung angemeldet hat, startet jede Anwendung, ohne nochmals nach weiteren Anmeldeinformationen zu fragen.

5.4 Benötigen Sie Unterstützung für mehrere Technologien?

Muss Ihr RBAC-System in den folgenden Jahren verschiedene Techniken unterstützen? Ein breites Spektrum an Entwicklungstechniken wird von .NET unterstützt: Winform, Webforms, Webservices... Ist Ihr System nur auf kurzfristige Anforderungen hin ausgelegt (beispielsweise Winform-Anwendungen), könnten langfristig zusätzliche Entwicklungslösungen für andere Technologien erforderlich werden, sodass am Ende mehrere RBAC-Systeme vorhanden sind und erhebliche Kosten für Entwicklung, Schulung und Instandhaltung entstehen.

5.5 Benötigen Sie Berichte zu Benutzerkonten und Berechtigungen?

Möglicherweise benötigen Sie detaillierte Beschreibungen zu vorhandenen Benutzerkonten, Rollen und zugewiesenen Berechtigungen.

5.6 Benötigen Sie Unterstützung für Audit-Anforderungen (SOX, ISO,...) ?

Zur Einhaltung gesetzlicher Anforderungen oder Zertifizierungsverfahren (Sarbanes-Oxley Act, ISO, CMMI, ITIL, ...) kann eine Protokollierung erforderlich sein, um nachweisen zu können, wer wann was in einer Anwendung tat. Beispiele: Wer hat eine bestimmte Anwendung gestartet? Wer hat ein bestimmtes Formular aufgerufen? Wer hat einen bestimmten Datensatz geändert? Weiterhin kann nach der Protokollierung eine Prüfung kritischer Transaktionen erforderlich sein, beispielsweise bei Zahlungstransaktionen.

5.7 Benötigen Sie Unterstützung für ein großes / internationales Entwicklungsteam?

Haben Sie ein großes Entwicklungsteam? Ist Ihr Entwicklungsteam über mehrere Standorte verteilt? In diesem Fall kann die Koordination sehr aufwendig sein, besonders bei Zeit- und Sprachunterschieden. Weitere Aspekte hierbei können Personalwechsel und Wissensmanagement sein.

5.8 Benötigen Sie eine feine Berechtigungsgranularität?

Bei der Festlegung einer Zugriffssteuerung beginnen Sie normalerweise damit, Menüoptionen zu deaktivieren. Darüber hinaus kann ein breites Spektrum von Aktionen in Ihren Anwendungen

erforderlich sein, wie beispielsweise das Deaktivieren von Optionen oder die Filterung von Daten entsprechend der Benutzerberechtigungen.

Welche der folgenden Aktionen könnten Sie benötigen? Verbergen oder Deaktivieren von Feldern, Menüoptionen, Registerkarten oder Steuerelementen? Unterschiedliche Filterung einer Liste in Abhängigkeit vom jeweiligen Benutzer? Anpassung von Unternehmensrichtlinien? Weitere ...?

6 Unterschätzte Kosten für Instandhaltung

Bei der Wahl einer Lösung stehen normalerweise die Anlagekosten im Vordergrund: Wie viel würde es kosten, die richtige Lösung zu entwickeln oder zu kaufen?

Oft wird nicht realisiert, dass diese Kosten nur 10-20 % der Gesamtkosten darstellen! Die meisten Kosten entstehen über einen längeren Zeitraum: Kosten entstehen vor allem durch die **Nutzung** und die **Instandhaltung** einer Anwendung.

Forrester-Research schätzt, dass 78 % des IT-Budgets für Instandhaltungskosten aufgewendet werden. Dies gilt sowohl für Unternehmensanwendungen als auch für RBAC-Systeme! Bei der Auslegung Ihres Systems sollten Sie die wichtigsten Kostenursachen berücksichtigen, die in den nächsten 10-20 Jahren auf Sie zukommen:

6.1 Unterstützung für Entwickler und Administratoren

Entwickler benötigen Unterstützung bei der Integration von Sicherheitsfunktionen in Anwendungen, bei der Definition von Berechtigungen, bei der Implementation des Authentifizierungsprozesses, bei neuen Versionen des .NET Frameworks... Administratoren benötigen Unterstützung bei der Verwaltung von Benutzerkonten, Rollen und Berechtigungen.

Sie benötigen eine Lösung, um Entwickler und Administratoren über Jahre zu unterstützen. Für betriebsinterne Lösungen benötigen Sie ein Supportteam und müssen über Jahre hinweg wechselndes Personal und Wissenstransfers bewältigen.

6.2 Konsistente Verwaltung von Berechtigungen und Anwendungscode

Während der Entwicklung einer Anwendung muss sichergestellt werden, dass die Berechtigungen und der Anwendungscode zueinander konsistent bleiben. Wird beispielsweise ein Steuerelement geändert, ist zu überprüfen, ob alle mit diesem Element verbundenen Beschränkungen weiterhin richtig funktionieren. Jede neue Version einer Anwendung erfordert eine vollständige Prüfung aller Berechtigungen. Einige RBAC-Systeme bieten ein automatisiertes Berechtigungsprüfverfahren, das Sie bei der Qualitätssicherung Ihrer Anwendung einsetzen können.

6.3 Bereitstellung neuer Anwendungsversionen

Jede neue Version der Anwendung stellt einen neuen Satz an Berechtigungen bereit. Diese neuen Berechtigungen sollten mit der neuen Version der Anwendung bereitgestellt werden. Leider können Sie nicht einfach das alte Repository durch das neue ersetzen, weil so alle vom Administrator während der Nutzungsphase eingegebenen Berechtigungen verloren gehen würden (Benutzerkonten, Rollen und entsprechende Berechtigungen).

Die neuen Berechtigungen müssen also sorgfältig in das bestehende Repository eingefügt werden. Hierfür werden manuelle Exporte und Importe erforderlich, falls Ihr RBAC-System Ihnen kein automatisiertes Bereitstellungswerkzeug bietet.

6.4 Versionierung von Sicherheitsdaten

Während der Bereitstellung einer neuen Anwendungsversion kann es sein, dass einige Benutzer noch eine ältere Version der Anwendung verwenden, und andere Benutzer schon die neue Version einsetzen. Während des Migrationsprozesses sollten sowohl das alte als auch das neue Berechtigungsrepository verfügbar sein. Sie sollten also sicherstellen, dass beide Versionen verfügbar bleiben, und jeder Benutzer nur auf die jeweils geeignete Version zugreift.

7 Betriebsinterne Entwicklungen als Sicherheitslösungen

Betriebsinterne Lösungen bieten einige Funktionalitäten, bringen aber Probleme mit sich:

7.1 Entwicklung und Support hängen ab von internen Ressourcen

Dies ist für Sie von besonderer Bedeutung, wenn Sie eine betriebsinterne Lösung einsetzen. Sie können es sich nicht leisten, durch Personalwechsel an Wissen zu verlieren oder gar ohne Sicherheitsspezialisten arbeiten zu müssen. Folglich müssen Sie ein Team unterhalten, das alle mit Ihrem Sicherheitssystem verbundenen Fragen, Veränderungen und Krisen handhaben kann.

7.2 Anwendungsspezifische Benutzerkonten

Die Implementierung eines Verfahrens zur einmaligen Anmeldung in .NET ist komplex. Gleiches gilt für die Kommunikation mit einem Active Directory und die Nutzung von Windows-Benutzerkonten, um die Identität von Benutzern zu prüfen. Deshalb verlassen sich die meisten Anwendungen auf speziell für eine Anwendung erstellte und in der Anwendungsdatenbank gespeicherte Benutzerkonten. Diese Lösung ist wahrscheinlich am leichtesten zu entwickeln, jedoch ist sie am schwersten zu verwalten. Benutzerkonten müssen hierbei vom Entwicklungsteam verwaltet werden, wogegen Active Directory / Windows-Benutzerkonten bereits existieren und von Systemadministratoren verwaltet werden.

7.3 Geringe Granularität bei Zugriffen und Berechtigungen

Eine betriebsinterne Lösung an alle Unternehmensanforderungen anzupassen, ist oft kompliziert. Führungskräfte neigen dazu, vor allem die Funktionalitäten zu sehen, aber deren technische Auswirkungen eher wenig abschätzen zu können. Beispielsweise sind Menüoptionen schnell erdacht, die Umsetzung kann aber komplex sein.

7.4 Spezifischer Anwendungscode

Betriebsinterne Lösungen realisieren Berechtigungen meist direkt im Anwendungscode. Das Ergebnis derartiger Entwicklungen ist ein komplexes Sicherheitssystem, das kostspielig zu verwalten und schwierig zu aktualisieren ist. Sollen Sicherheitsdaten eingefügt oder aktualisiert werden, muss der Entwickler den Programmcode abrufen, die entsprechende Stelle finden, sie ändern und die Anwendung wieder testen und neu bereitstellen. Dieses Verfahren ist weder einfach noch flexibel.

Zudem werden viele Sicherheitsbelange oft erst lange nach der Entwicklungsphase deutlich. Es ist meist sehr schwierig oder gar unmöglich, alle Erfordernisse bereits in der Entwicklungsphase vorherzusehen und zu berücksichtigen. Oft werden komplexe Berechtigungsanforderungen erst formuliert, sobald die Anwendung im Einsatz ist und schnelle Problemlösungen gefordert sind.

7.5 Instandhaltungsaspekte bei separaten Anwendungslösungen

Theoretisch ist es erforderlich, eine klare Vorstellung davon zu haben, wie sich ein Unternehmen in den nächsten 5-10 Jahren entwickelt. In der Praxis es jedoch kaum möglich, dieses vorherzusehen. Es ist gut möglich, dass Sie eine neue Anwendung entwickeln müssen, die nicht auf einer von Ihrem vorherigen Sicherheitssystem unterstützten Technik basiert. Hierbei würden Sie wahrscheinlich ein neues Sicherheitssystem für die neue Anwendung maßschneidern. Schließlich werden Sie ein separates Sicherheitssystem für jede Anwendung haben und damit umfangreiche Instandhaltungskosten.

8 Visual Guard als Unternehmenslösung für Anwendungssicherheit

Novalys entwickelt seit 15 Jahren Authentifizierungs- und Berechtigungslösungen für Unternehmensanwendungen. Auf Grundlage dieser Erfahrungen haben wir Visual Guard als Sicherheitslösung für .NET-Anwendungen beliebiger Komplexität entwickelt.

8.1 Einheitliche Lösung für Sicherung und Zentralisierung aller .NET-Anwendungen

Visual Guard bietet Ihnen eine einheitliche Lösung alle Anwendungsarten, wie wie Winform, Webform und Webservices. Sie können die Verwaltung all Ihrer verschiedenen Anwendungen und Benutzerkonten in einem einzigen, sicheren Repository zentralisieren. Visual Guard bietet eine gebrauchsfertige Lösung inklusive eines Verfahrens zur einmaligen Anmeldung auf Basis von Windows-Benutzerkonten. Zur weiteren Vereinfachung der Berechtigungsverwaltung können Sie gemeinsam verwendete Rollen erstellen, die Benutzerberechtigungen für mehrere Anwendungen aufnehmen.

8.2 Vom Programmcode unabhängige Berechtigungen

Mit Visual Guard schreiben Sie keinen Programmcode in Ihre Anwendung, um Berechtigungen zu definieren. Dies bedeutet, dass Sie Berechtigungen definieren und implementieren können, ohne durch einen vollen Entwicklungszyklus gehen zu müssen (Programmieren, Testen, Bereitstellen, ...). Sie können Berechtigungen jederzeit definieren, selbst wenn eine Anwendung verwendet wird: Neue Berechtigungen werden sofort wirksam.

8.3 Verwaltungswerkzeuge für nichttechnisches Personal

Die Visual Guard Konsole macht die tägliche Sicherheitsverwaltung auch für nichttechnisches Personal einfach. Die Verwaltung von Benutzerkonten, Rollen und Berechtigungen erfordert keine technischen Qualifikationen.

Bei der Auswahl einer für die Sicherheit verantwortlichen Person sind Sie damit nicht mehr von technischen Fähigkeiten abhängig und können Ihre Unternehmensabläufe freier gestalten: Als Administratoren agieren können Sicherheitsbedienstete, Systemadministratoren, Führungskräfte entfernter Standorte usw.

Für eine noch weiter gehende Flexibilität können Sie mehrere Administrator-Berechtigungsstufen definieren.

Sie können auch Ihr eigenes Administratorformular erstellen und es in Ihre Anwendung integrieren, um hierüber das Visual Guard API zur Verwaltung von Benutzerkonten und Rollen aufzurufen.

8.4 Werkzeuge zur Berechtigungsverwaltung, Versionierung und Bereitstellung

Visual Guard bietet Entwicklern eine breite Palette an Werkzeugen und Assistenten. Sie können Berechtigungen mit wenigen Mausklicks erstellen, die Konsistenz zwischen Anwendungen und Berechtigungen automatisch überprüfen, neue Berechtigungen ohne Beeinträchtigung vorhandener Sicherheitsdaten in bestehende Repositories bereitstellen, mehrere Versionen eines Repositories bei der Bereitstellung einer Anwendung verwalten usw.

8.5 Audit-Werkzeuge für Berechtigungsberichte und Anwendungsprotokollprüfungen

Auditoren besitzen einen schreibgeschützten Zugriff, um vorhandene Berechtigungen, Rollen und Benutzerkonten zu untersuchen und diesbezüglich detaillierte Berichte zu generieren. Sie können auf Anwendungsprotokolle zugreifen, um bestimmte Transaktionen zu überprüfen, und auf Protokolle der Verwaltungskonsole zugreifen, um zu überprüfen, wer wem welche Berechtigungen gab.

8.6 Standardprodukt mit professionellem Support und ständigen Aktualisierungen

Warum das Rad neu erfinden? Visual Guard stellt gebrauchsfertige Funktionalitäten für die Anwendungssicherheit zur Verfügung. In 15 Jahren Entwicklungszeit ist es zur Industriestandardlösung gereift.

Visual Guard ist schnell und einfach zu verstehen und zu implementieren.

Novalys bietet internationalen Support: Sie können sich auf Ihre Kernaufgaben konzentrieren, während wir uns um die Sicherheit Ihrer Anwendungen kümmern. Novalys ist darüber hinaus bestrebt, die kontinuierlichen Innovationen von Microsoft (.NET Framework, Betriebssysteme, ...) bestmöglich zu nutzen.