we make IT simple

Security in IBM Lotus Notes Anwendungen

Manfred Meise

IBM Certified Advanced Developer - Lotus Notes and Domino R3 – R8.5
IBM Certified Advanced Administrator - Lotus Notes and Domino R3 – R8, R8.5
IBM Certified Advanced Instructor – Lotus Notes and Domino R3- R8, R8.5

Certified for IBM Lotus software

Track 4 Session 1: Montag 26.03.2012 13:45 – 15:15

IBM Business Partner

zu meiner Person: Manfred Meise

- Studium Elektrotechnik (Dipl. Ing. (FH))
- Arbeit als Softwareingenieur seit mehr als 30 Jahren bei verschiedenen Computerherstellern und Softwarehäusern
- Gründer und Geschäftsführer der mmi consult gmbh
- Erfahrungen mit Lotus Notes/Domino seit 1992 - Markteinführung in Europa
  (als Leiter Strategische Projekte bei Lotus Development Deutschland)
- IBM Zertifizierungen als Anwendungsentwickler, Systemadministrator, Trainer für die Produktversionen R3 bis R8.5
- Tätigkeitsschwerpunkte im Entwicklungsbereich:
  CRM, Workflow, Objektorientierte Anwendungsarchitekturen, XPages
- Tätigkeitsschwerpunkte als Systemadministrator:
  Domänenzusammenführungen und –trennungen, Betriebshandbücher und Administrationsstandards, Versionswechsel, Infrastruktur-Audits, Client-Rollouts
- Erreichbar unter:
  - ***manfred.meise@mmi-consult.de***
  - ***http://www.mmi-consult.de***
  - ***http://www.mmi-consult.de/faq***

Meine Themen heute …

- Datenschutz und Datensicherheit
- Berechtigungsvergabe durch den Domino Administrator
- Datenbankberechtigungen
- Feldsicherheit
- Management von Berechtigungen, Keys und Tokens
- Resümee und Ausblick

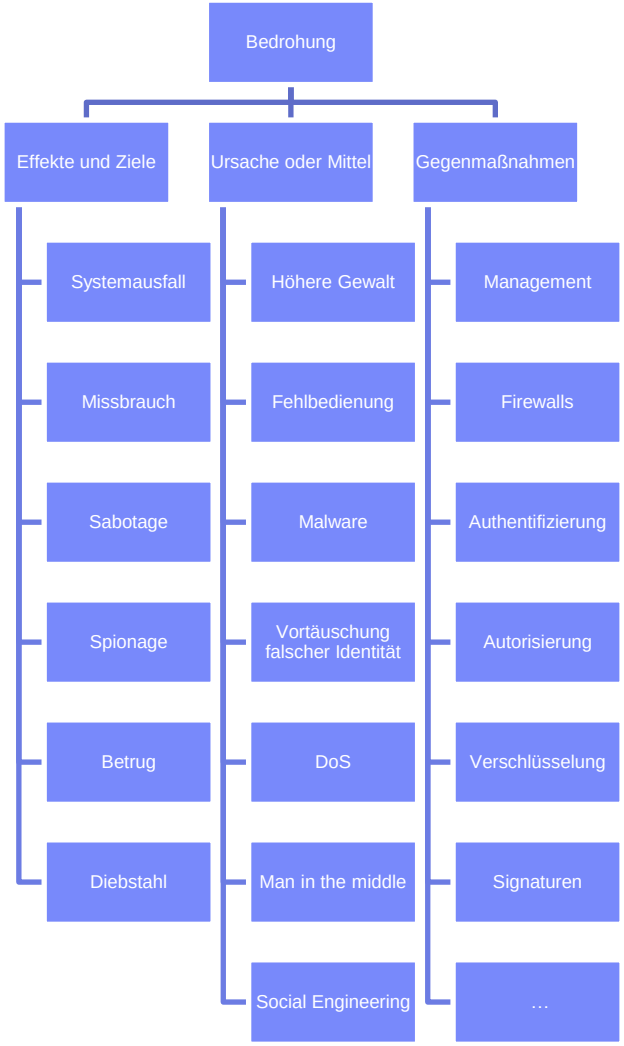
Datenschutz und Datensicherheit

Datenschutz wird leider zu gering bewertet …

Was ist IT-Sicherheit?

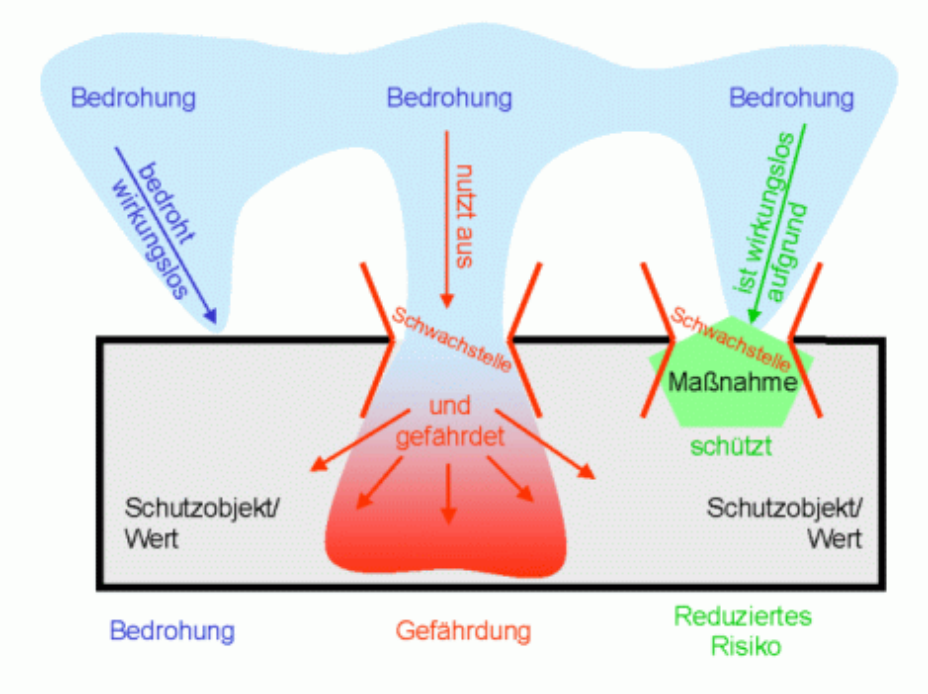
- Vertraulichkeit
  - Daten dürfen lediglich von autorisierten Benutzern gelesen bzw. modifiziert werden, dies gilt sowohl beim Zugriff auf gespeicherte Daten wie auch während der Datenübertragung
- Integrität
  - Daten dürfen nicht unbemerkt verändert werden, bzw. es müssen alle Änderungen nachvollziehbar sein
- Verfügbarkeit
  - Verhinderung von Systemausfällen; der Zugriff auf Daten muss innerhalb eines vereinbarten Zeitrahmens gewährleistet werden

Bedrohungen der IT Sicherheit



Bedrohung + Schwachstelle = Gefährdung

- Eine Bedrohung allein ist wirkungslos
- Eine Bedrohung bei Ausnutzung vorhandener Schwachstellen richtet Schaden am Schutzobjekt an
- Durch gezielte Gegenmaßnahmen werden Schachstellen „gestopft“. Das Risiko für Schaden wird reduziert oder „auf Null“ gesetzt



Publizierte Domino Schwachstellen:
`http://www.ibm.com/developerworks/lotus/security`

Einige Gegenmaßnahmen gegen Bedrohungen

Bedrohung	Gegenmaßnahme
Abhören	Verschlüsselung
Vertraulichkeit	Verschlüsselung
Authentifizierung	Zertifikate, Benutzernamen / Passwörter
Verfälschung	Signatur
Unbefugter Zugriff	ACL / Dokumentenrechte
DoS	Firewalls, Internet-Lockout, Agentensicherheit

Domino Sicherheit muss durch jeden umgesetzt werden

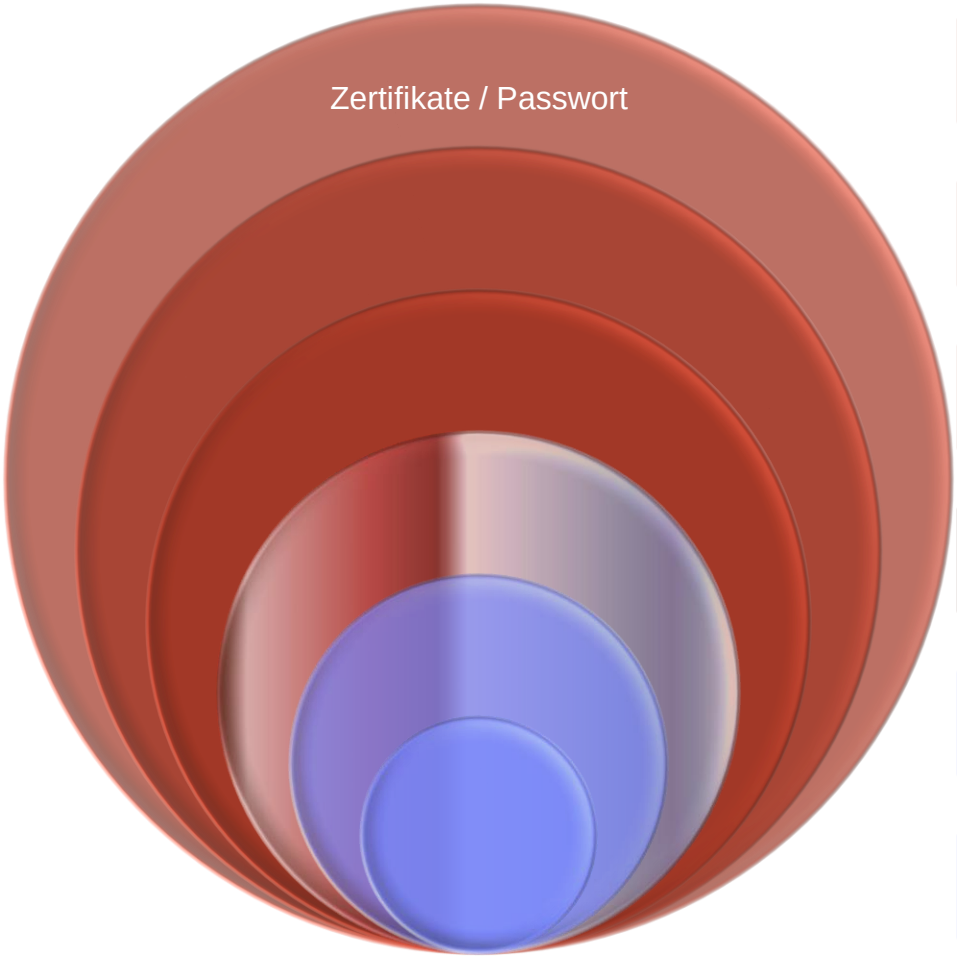
- Sicherheit muss etabliert werden
  - System-/Netzwerkadministration
  - Datenbankentwicklung
  - Helpdesk
  - Benutzer

Demo- /Testumgebung

- IBM Lotus Domino Server
- 2 User mit unterschiedlichem Zugriff (z:b. umschaltbare Windows-Clients)
- Diskussionsschablone mit
  - Ansichtssicherheit
  - Maskensicherheit
  - Leser- / Autorennamen
  - Abschnitten / Signatur
  - Verschlüsselung mit
    - Dokumentenschlüssel
    - Public Key des Benutzers
- AdminP
  - zur Umbenennung bei korrekten Feldtypen
  - Löschung bei korrekten Feldtypen

Berechtigungsvergabe
durch den Domino Administrator

Das Notes/Domino Sicherheitsmodell



Authentifiziert den Benutzer / Server

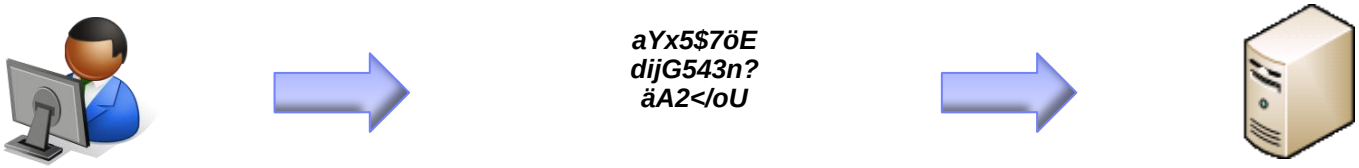
Authentifizierung von Benutzern

- Benutzer von Lotus Notes Clients
  - verwenden stets die in ihren Notes-IDs gespeicherten Zertifikate
- Benutzer von Web Clients
  - verwenden Benutzername / Passwort
  - Verwendung von SSO (z.B. SPNEGO)
  - könnten Client Zertifikate verwenden
- Andere Server
  - verwenden stets die in ihren Notes-IDs gespeicherten Zertifikate

Authentifizierung: Phase 1

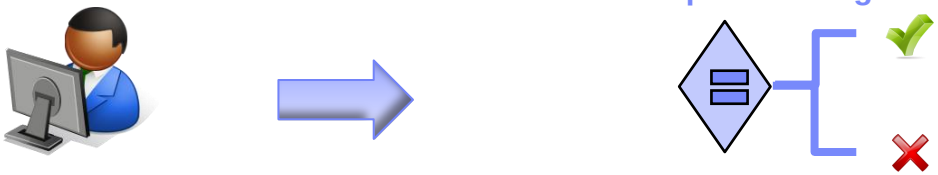
- Der Client stellt sicher, einen vertrauenswürdigen Server zu kontaktieren

Client erzeugt eine Zufallszahl und bittet den Server um Signatur



Server signiert die Nachricht und schickt sie zurück

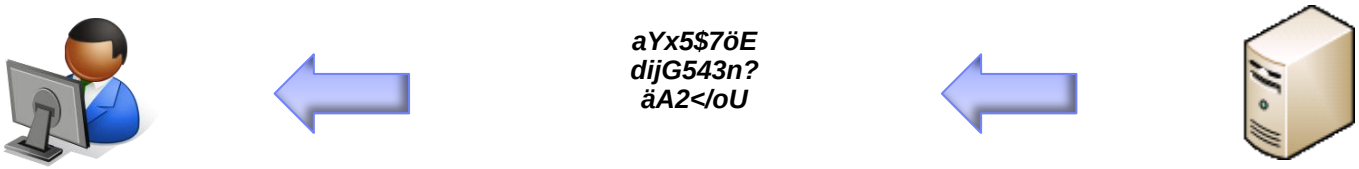
Client überprüft die Signatur



Authentifizierung: Phase 2

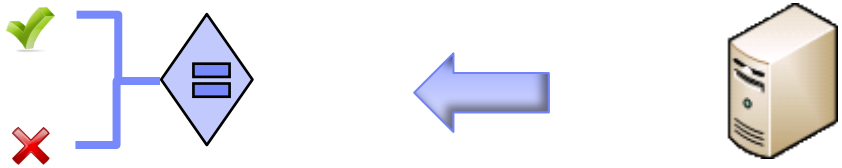
- Der Server stellt sicher, einen bekannten Benutzer zu kontaktieren

Server erzeugt eine Zufallszahl und bittet den Client um Signatur

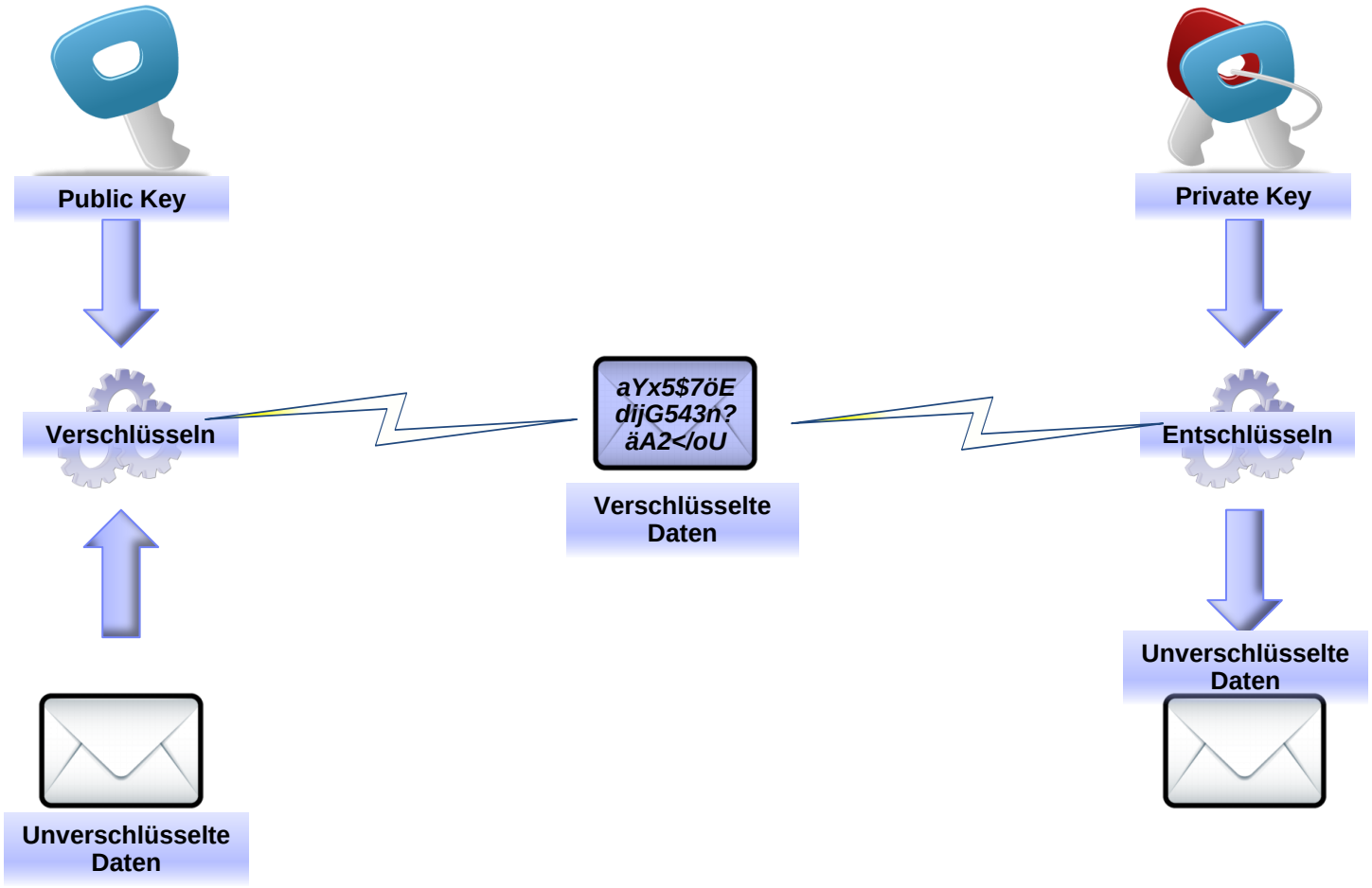


Client signiert die Nachricht und schickt sie zurück

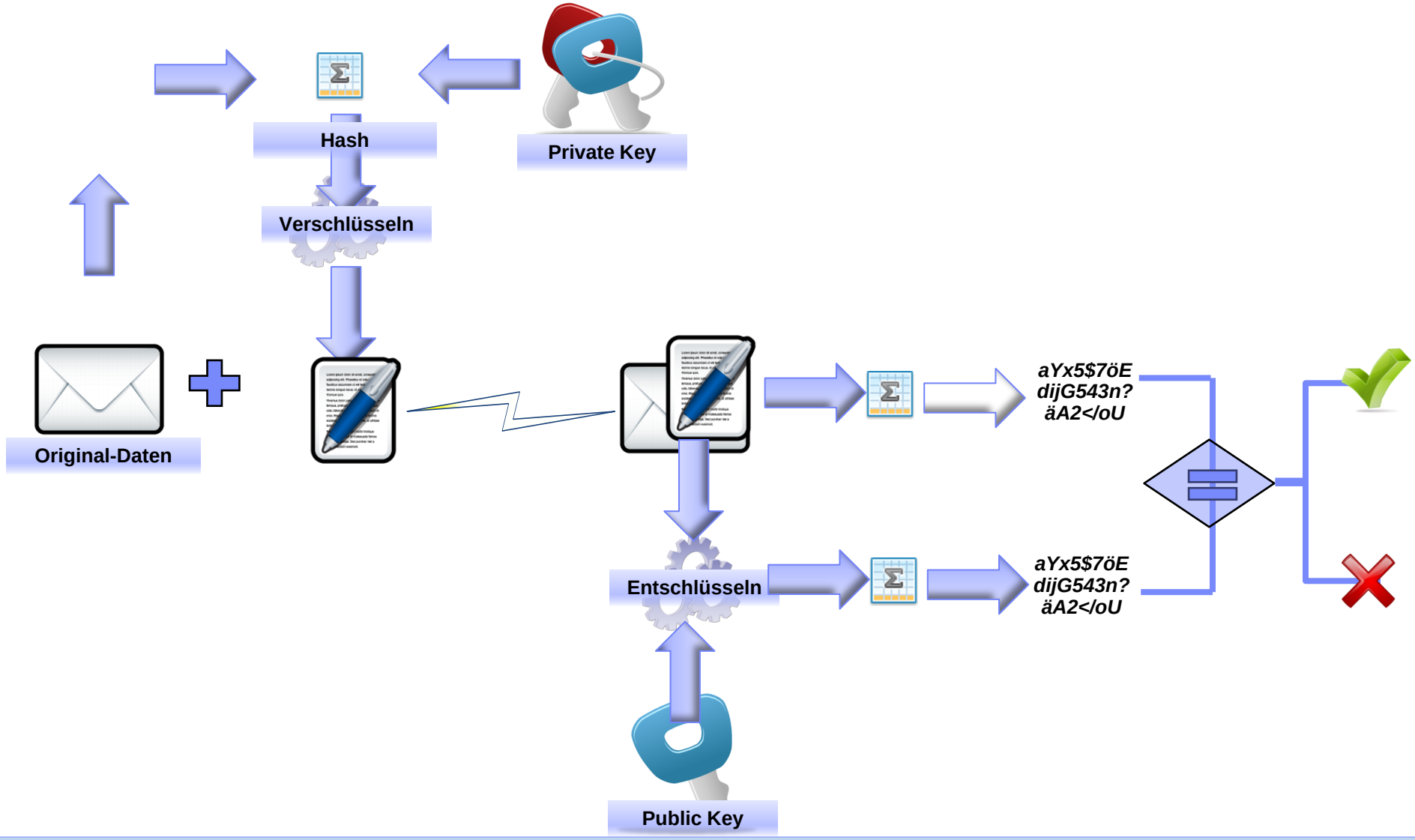
Server überprüft die Signatur



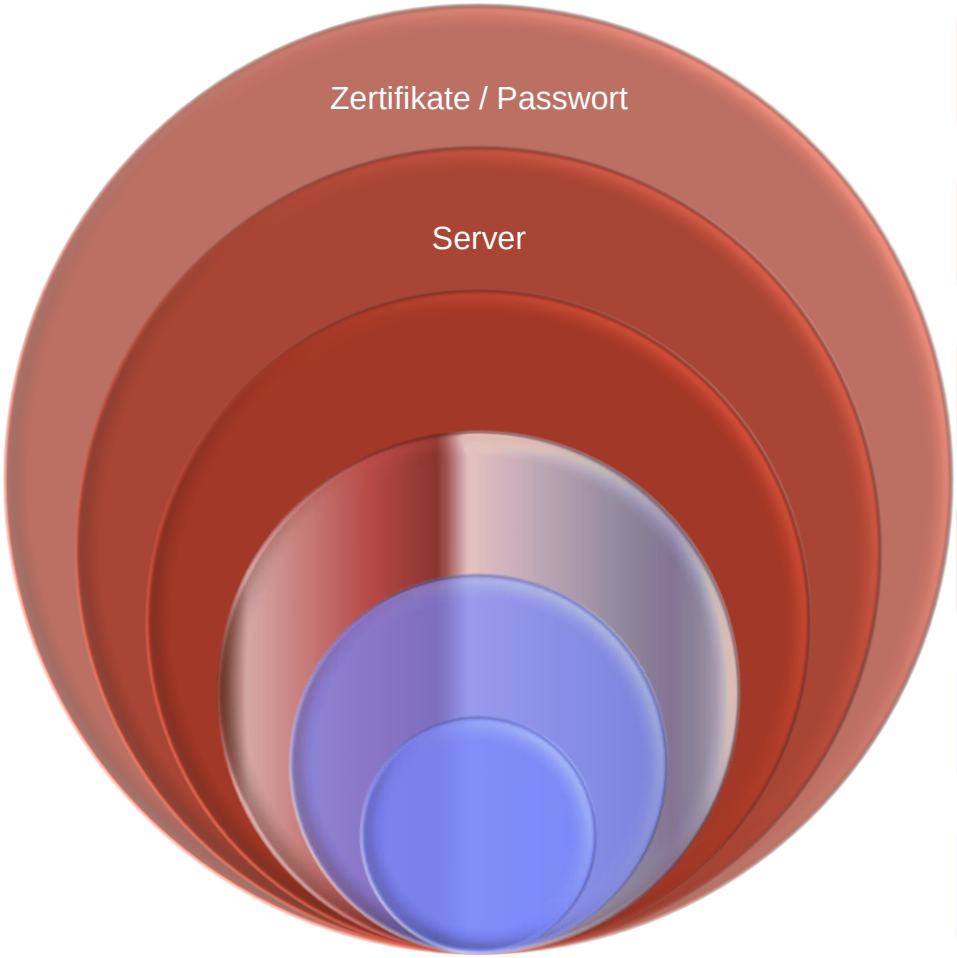
Dual key encryption



Signatur


Hash
Private Key
Verschlüsseln
Original-Daten
aYx5$7öE
dijG543n?
äA2</oU
Entschlüsseln
aYx5$7öE
dijG543n?
äA2</oU
Public Key

Das Notes/Domino Sicherheitsmodell



Authentifiziert den Benutzer / Server

Prüft Nutzungsberechtigung des spez. Servers (j/n)

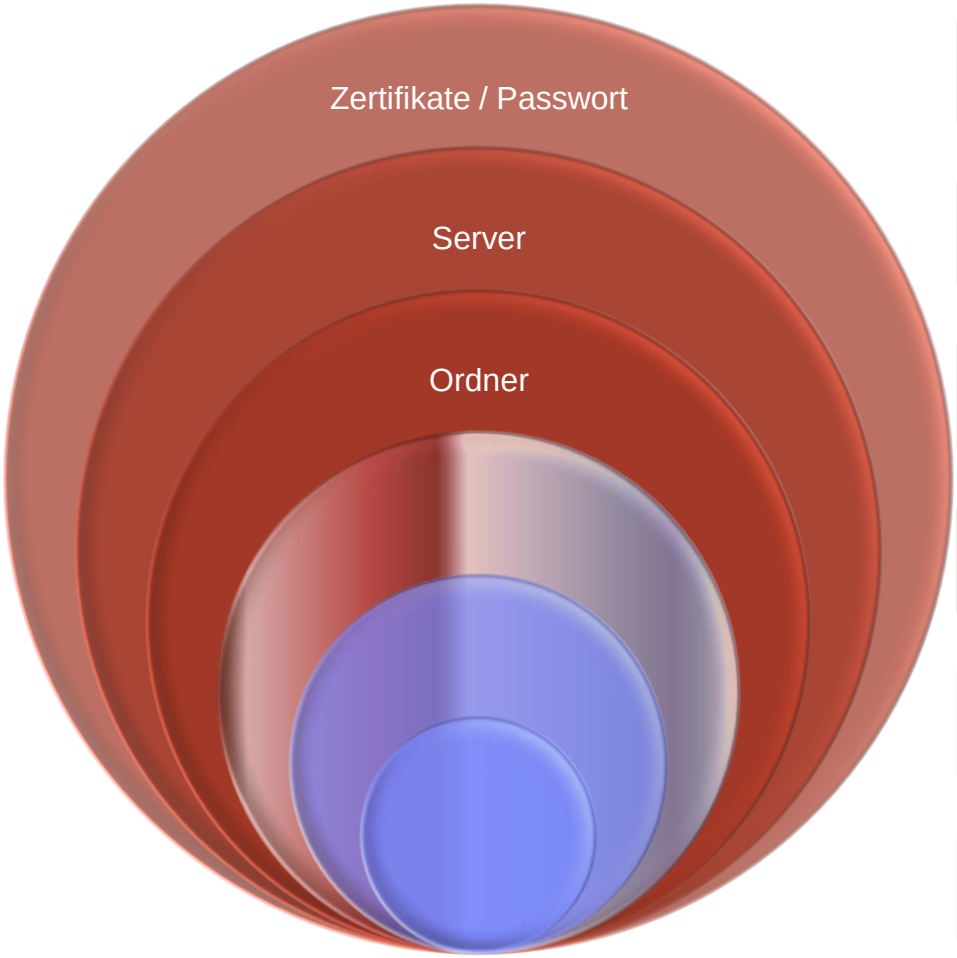
Authentifizierung von Benutzern/Servern

- Notes-IDs werden in einer Zulassungshierarchie mit Schlüsseln/Zertifikaten aus einem Basisschlüssel vergeben
- Authentifizierung wenn Benutzer / Server einen gemeinsames Basiszertifikat besitzen – sonst Querzulassung erforderlich
- Lotus Notes Clients interpretieren die Notes-IDs und führen nach Passworteingabe stets Authentifizierung bei Serverkontakt durch.
- Browser benötigen alternative Verfahren
  - Benutzername / Passwort (hinterlegt im Domino Directory)
  - SPNEGO
  - Client Zertifikate
- Erfordert individuelle Identifizierung einzelner Benutzer
  - keine gemeinsame „Abteilungs-“ oder „Gruppen-IDs“
- Steht und fällt mit dem Benutzerverhalten
  - Weitergabe von Notes-IDs
  - Weitergabe von Kennwörtern

Agenten

- Signaturstrategie von Anwendungen
  - legt fest in wessen Namen der Agent läuft
  - Server-ID
  - Spezialisierte Pseudo-IDs mit abgestuften Rechten
  - Fremdanwendungen gar nicht signieren?
- Agenten haben Ausführungsrechte
  - festgelegt im Serverdokument auf der Basis der Signatur
  - implementiert durch den Servertask „AgentManager“
- Agenten haben Zugriffsrechte
  - festgelegt durch Signatur der Datenbank und ACL der Datenbank, auf die zugegriffen wird
- Agenten können in anderem Namen ausgeführt werden
  - aktueller WebBenutzer
  - benannter Notes-Benutzer

Das Notes/Domino Sicherheitsmodell



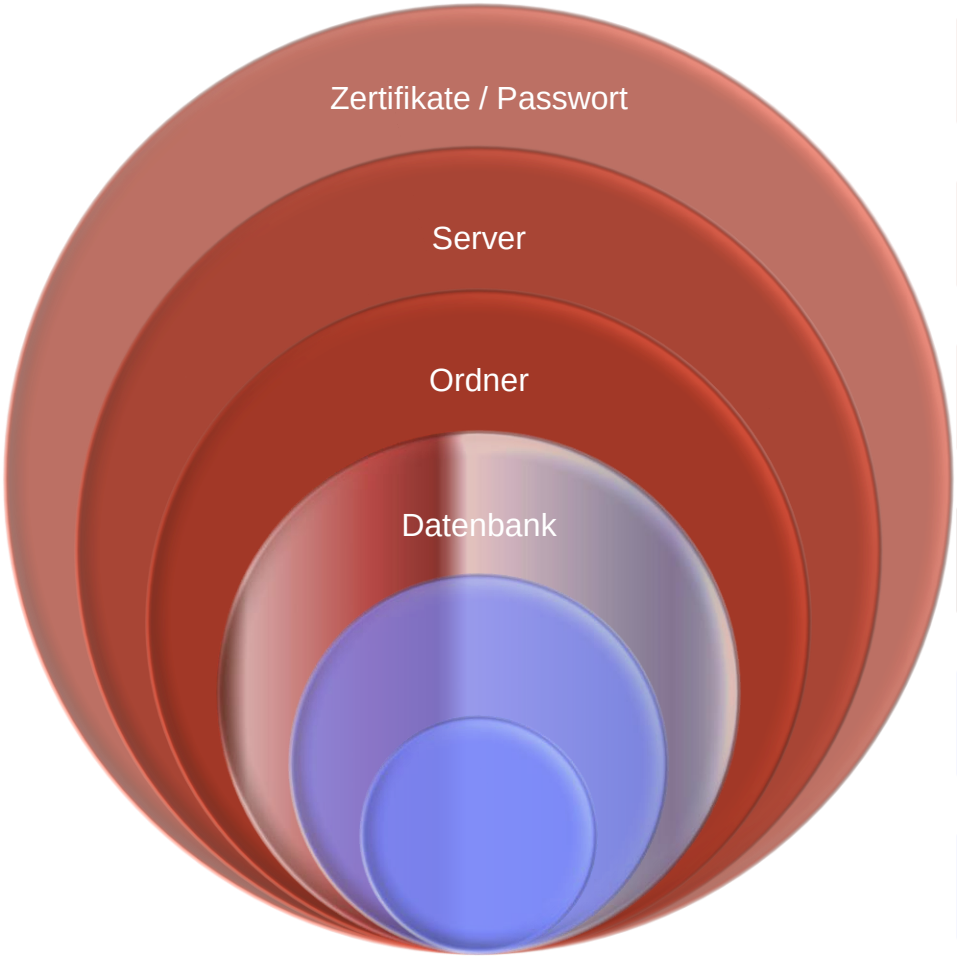
Authentifiziert den Benutzer / Server

Prüft Nutzungsberechtigung des spez. Servers (j/n)

Prüft Nutzungsberechtigung spez. Unterverzeichnisse des Servers (j/n)

Datenbankberechtigungen

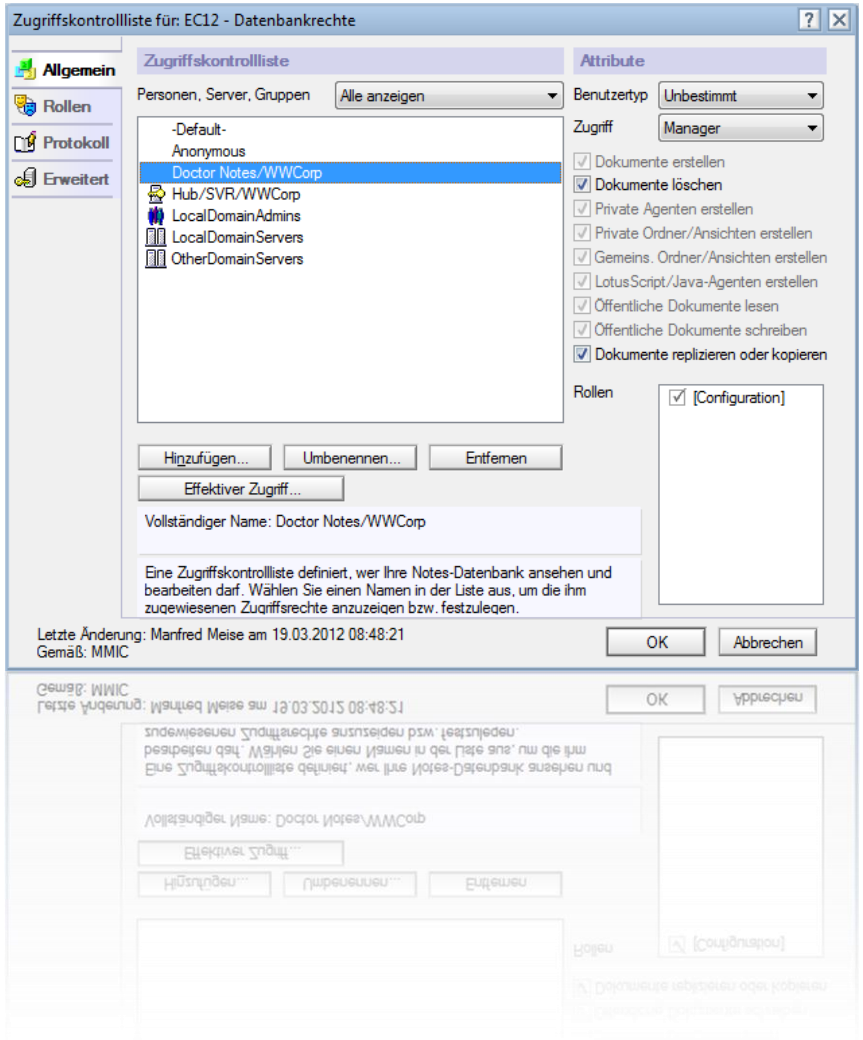
Das Notes/Domino Sicherheitsmodell



- Authentifiziert den Benutzer / Server
- Prüft Nutzungsberechtigung des spez. Servers (j/n)
- Prüft Nutzungsberechtigung spez. Unterverzeichnisse des Servers (j/n)
- Datenbank-ACL ordnet Benutzer in 7 das Ebenen Rechteschema ein

Datenbankzugriffsliste (ACL)

- Rechteliste pro Datenbank (in Datenbank gespeichert)
- legt Berechtigung für Benutzer, Server, Gruppen für diese Datenbank gemäß 7 Rechteebenen fest
- anwendungsspezifische Rollendefinitionen möglich (Auswirkung durch den Entwickler implementiert) – Namen, Auswirkung, Zuweisung erfordern firmenspezifische Dokumentation
- Benutzer können
  - anonym sein (i.d.R. Browser-Zugriff)
  - authentifiziert sein (i.d.R. Lotus Notes Client oder auch Browser-Zugriff)
- Max. Browserzugriff separat definierbar
- Standardisierungen dringend empfohlen
- Protokollierung aller Änderungen (für die Lebensdauer der Datenbank) in der Datenbank



ACL Berechtigungen

	Zugriffsebene	Benutzerzugriff	Serverzugriff
	Kein Zugriff	➢kein Zugriff	➢kein Zugriff
	Einlieferer	➢Dok. erstellen	➢keine Replikation
	Leser	➢Dok. lesen **1	➢Replikation lesend ➢min. Zugriff für Repl.
	Autor	➢Dok. Lesen **1 ➢Dok. erstellen **2 ➢eigene Dok. bearbeiten **3	➢kann neue Dok. replizieren, nicht jedoch bestehende akt. ➢Hinweis: Nicht zu empfehlen
	Editor	➢Dok. Lesen **1 ➢Dok. erstellen **2 ➢alle Dok. bearbeiten	➢kann alle neuen und geänderten Dok. replizieren
	Entwickler	➢wie Editor ➢ + Optik und Funktion der Anwendung beeinflussen	➢wie Editor ➢repliziert Gestaltungsänderungen
	Manager	➢wie Entwickler ➢+ Datenbankrechte setzen ➢+ Datenbank löschen	➢vollständige Replikation von Daten, Gestaltung und ACL

**1 sofern keine dokumentbasierte Leserbeschränkungen existieren
**2 sofern entsprechendes Zugriffsattribut gesetzt ist
**3 erfordert die Verwendung von Autoren Feldern
**4 Löschberechtigungen durch separates Attribut für alle Zugriffsebene

Zugriffsattribute der ACL

Zugriffsebene	Dokumente erstellen	Dokumente löschen	Private Agenten erstellen	Private Ordner/Ansichten erstellen	Gemeinsame Ordner/Ansichten erstellen	LotusScript/Java Agenten erstellen	ÖffentlichDokumente lesen	Öffentliche Dokumente schreiben	Dokumente replizieren oder kopieren
Kein Zugriff									
Einlieferer									
Leser							X		
Autor							X		
Editor	X						X	X	
Entwickler	X		X	X	X		X	X	
Manager	X		X	X	X	X	X	X	

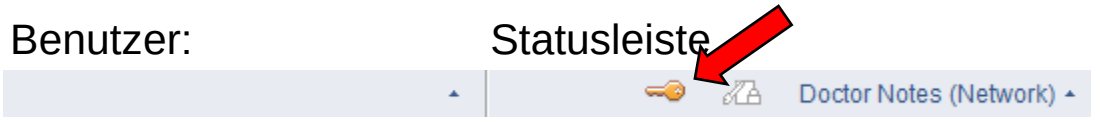
	Selektierbar durch Manager
	Selektion nicht möglich (nicht selektiert)
X	Selektion nicht möglich (selektiert)

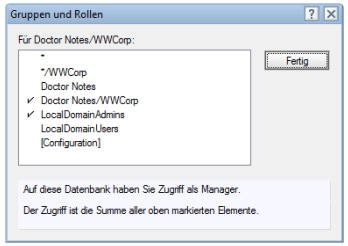
ACL Auswertungen

- Auswertung stets bei Öffnen der Datenbank (Recht bleibt erhalten, solange Session offen)
- Auswertereihenfolgen
  - Anonymous
  - Benutzerspezifische Festlegung
  - Gruppendefinition(en)
  - -Default-
- Berechtigung
  - Kumulation bei Mitgliedschaft in mehreren Gruppen
  - Explizite Definition bei benutzerspezifischer Angabe
- Rollen und Attribute
  - Kumulation über alle zutreffenden Auswertungen und Gruppen
- Konsistente ACL
  - Erforderlich?!?
- Max. Internetnamens- und Kennwortzugriff
  - abweichende Berechtigung eines Benutzers über Browser-Zugriffe

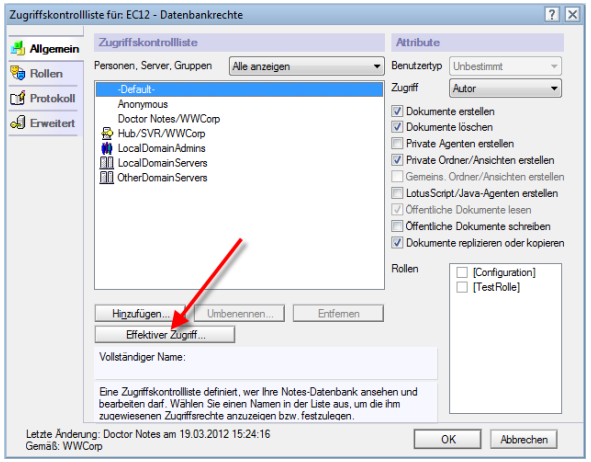
Effektive Berechtigung

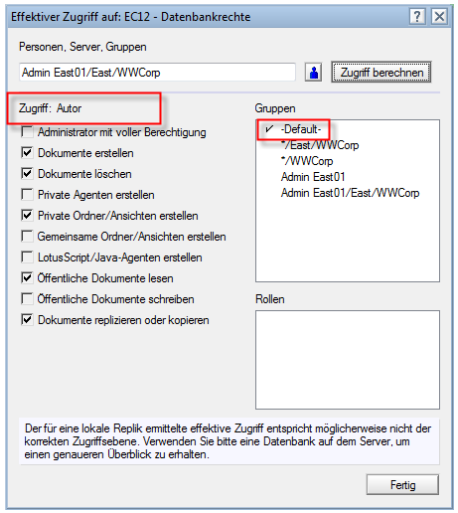
- ACL Dialog ist primär zur Pflege gedacht (Soll-Zustand)
- Bei Berücksichtigung von Gruppen, Schachtelungen, Caching etc. kann der Ist-Zustand der Berechtigung von Soll-Zustand abweichen
- Effektive Berechtigung stellt fest, welches Recht aktuell gilt:
  - Benutzer: Statusleiste





  - Datenbankverwalter: ACL-Dialog





Effektive Berechtigung: Beispiele

Admin East01/East/WWCorp ist mit Mitglied von „Gruppe 1“ und „Gruppe 2“

- Szenario1:

Eintrag	Zugriffsebene	Rolle
Gruppe 1	Editor	A
Gruppe 2	Leser	B

  - **Ergebnis:** Benutzer erhält „Editor“ Recht und die Rollen „A“ und „B“

- Szenario 2:

Eintrag	Zugriffsebene	Rolle
Gruppe 1	Editor	A
Gruppe 2	Leser	B
Admin East/East/WWCorp	Autor	

  - **Ergebnis:** Benutzer erhält „Autor“ Recht und keine Rollen

ACL Pflege

- ACL Pflege von Datenbanken in der Produktionsumgebung ist Aufgabe des Systemadministrators
- Vorgaben, Empfehlungen, Dokumentation der Auswirkungen auf Anwendungsverhalten ist Aufgabe des Datenbankentwicklers
- Pflege der Mitglieder von Rechtegruppen kann durch Systemadministration, Helpdesk oder auch Fachabteilung erfolgen

Best Practice - Gruppen

- Einheitliches Namensschema
- Namensschema sollte Typen berücksichtigen
  - Sonderzeichen zu Beginn des Namens
  - Zielanwendung als Teil des Namens
  - Berechtigungsebene als Teil des Namens
- Typisierung verwenden
  - Mail only
  - Access Only
- ACL Benutzertypen setzen
- Administrative Gruppen
  - LocalDomainServers – mehrere Zwecke – Benutzertyp „Servergruppe“
  - LocalDomainAdmins – mehrere Zwecke – Benutzertyp „Personengruppe“
- Deny Access Groups
- Serververwaltungsgruppen zur Steuerung auf Serverebene
  - Datenbanken anlegen etc.

ACL Prüfung / Pflege auf Rollen - DBOpen

```
'-------------------------------------------
' Specify and Verify Roles in ACL
'-------------------------------------------
If UBound(strRequireRoles) >= 0 Then
        ' Get list of missing or obsolete roles
        vRolesCurrent           = db.acl.Roles
        vRolesMissing           = FullTrim(ArrayReplace (strRequireRoles, vRolesCurrent, ""))
        vRolesObsolete          = FullTrim(ArrayReplace (vRolesCurrent, strRequireRoles, ""))

        If (vRolesMissing(0) <> "") Or (vRolesObsolete(0) <> "") Then
                'Prepare resulting message
                If vRolesMissing(0) <> "" Then
                        strErrorMsg = "Fehlende ACL-Rollen in Datenbank:" & _
                        Chr(10) & Join (vRolesMissing, Chr(10)) & Chr(10)
                End If

                If (vRolesObsolete(0) <> "") Then
                        If strErrorMsg <> "" Then strErrorMsg = strErrorMsg & Chr(10)
                        strErrorMsg = "Überflüssige ACL-Rollen in Datenbank:" & _
                        Chr(10) & Join (vRolesObsolete, Chr(10))  & Chr(10)
                End If

                'Roles are missing in ACL of this database
                If db.CurrentAccessLevel = ACLLEVEL_MANAGER Then
                        'We can add missing roles
                        Call setRoles (db, ArrayUnique(strRequireRoles))
                        MessageBox strErrorMsg & "ACL wurden automatisch angepasst",
MB_ICONEXCLAMATION,""

                Else
                        'Notify administrator to add missing roles
                        MessageBox strErrorMsg & "Bitte den Systemadministrator
benachrichtigen", MB_ICONEXCLAMATION,""

                End If
        End If
End If
```

Auswertung der ACL

- für jede servergestützte Datenbank
- lokale Replikate von servergestützten Datenbanken
  - Benutzer erhalten ein ähnliches Verhalten wie für das servergestützt Replikat
  - lokales Replikat kann gelöscht werden, auch wenn man kein Manager ist
  - Gruppenmitgliedschaften werden bei jeder Replikation ausgewertet, um lokales Verhalten zu aktualisieren
- „konsistente ACL“ führt zur ACL Auswertung auch bei lokalen Datenbanken / Kopien
- Typisierung der ACL Einträge erhöhen Sicherheit (erfordern allerdings saubere Zuordnung und „reinrassige“ Gruppen)
  - „man kann typisieren – muss es aber nicht“
  - „wenn man typisiert, muss man allerdings richtig typisieren“

Persönliche Ansichten / Ordner

- Gespeichert in Desktop
  - Roaming seit 8.5.1
- Gespeichert in Datenbank
  - Erfordern entsprechende ACL Attribute
  - Bei fehlenden ACL Attributen werden diese in Desktop gespeichert

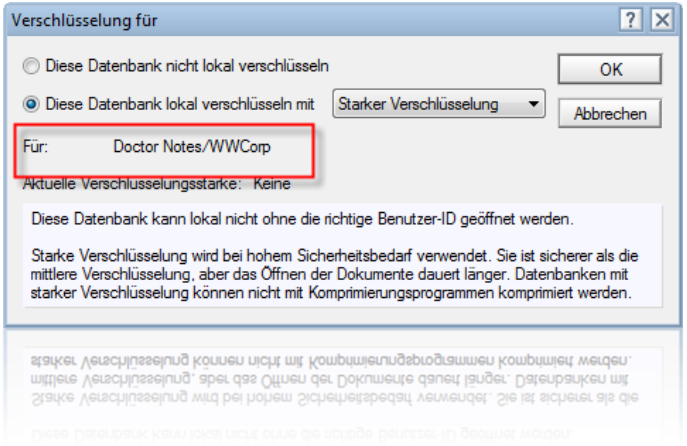
Programmatische Rechteprüfung-/pflege

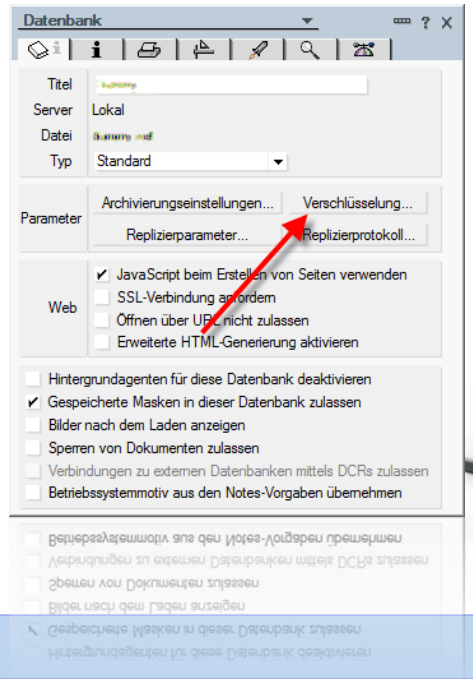
- @UserNamesList
  - Ermittelt alle Varianten des Namens, der Gruppenmitgliedschaften und Rollen des aktuellen Benutzers in der aktuellen Datenbank
- @UserAccess(server : file ; [accessPrivilege])
  - Ermittelt die aktuelle Zugriffsebene oder ausgewählte Attribute des aktuellen Benutzers in der aktuellen Datenbank
- @UserRoles
  - Ermittelt die Rollen der aktuellen Datenbank

- NotesACL
  - Merkmale und Funktionen der Datenbank-ACL
- NotesACLEntry
  - Merkmale und Funktionen zu einem spezifischen Eintrag

Max. Schutz lokaler Datenbanken

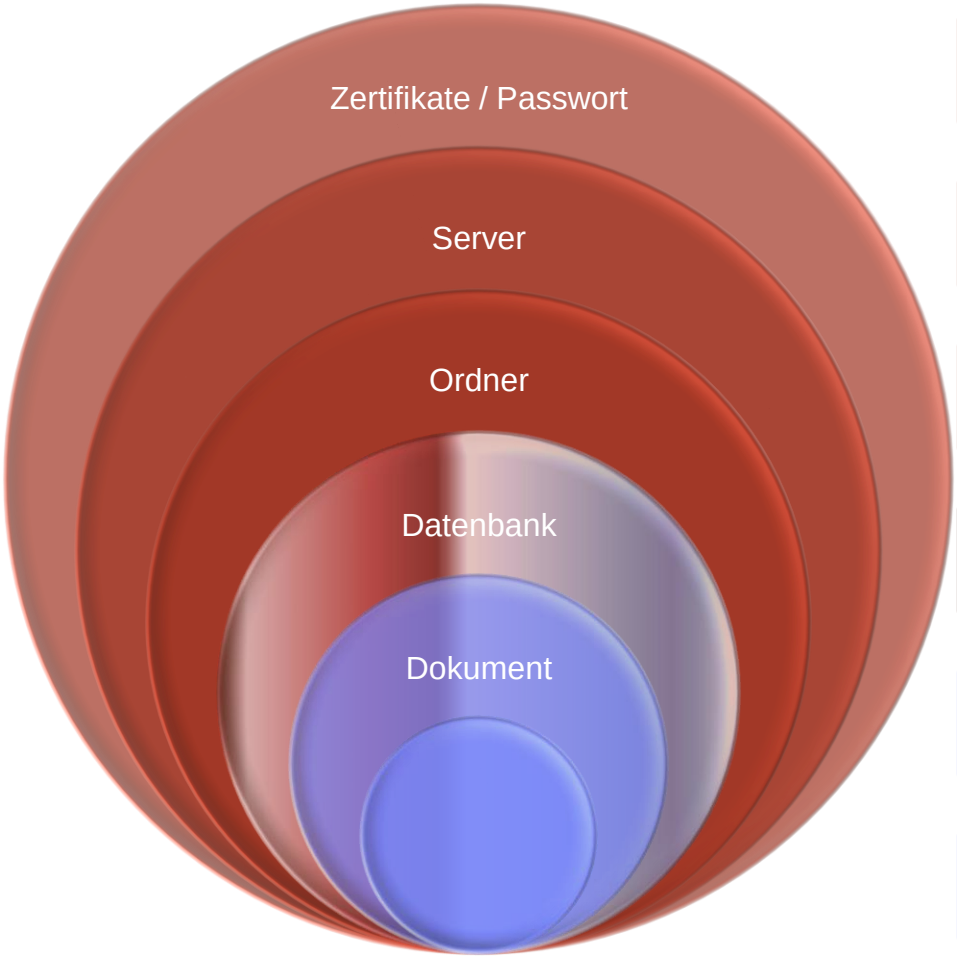
- Lokale Datenbanken können mit der ID **eines** Benutzers verschlüsselt werden (erhöhter Schutz vor unbefugter Nutzung/Diebstahl)
- Einrichtung bei Erstellung, neuer Kopie, neuer Replik
- Nachträglich änderbar
- **ACHTUNG**: Daten gehen unwiderruflich verloren, wenn Notes-ID verloren wird





Dokumentensicherheit

Das Notes/Domino Sicherheitsmodell

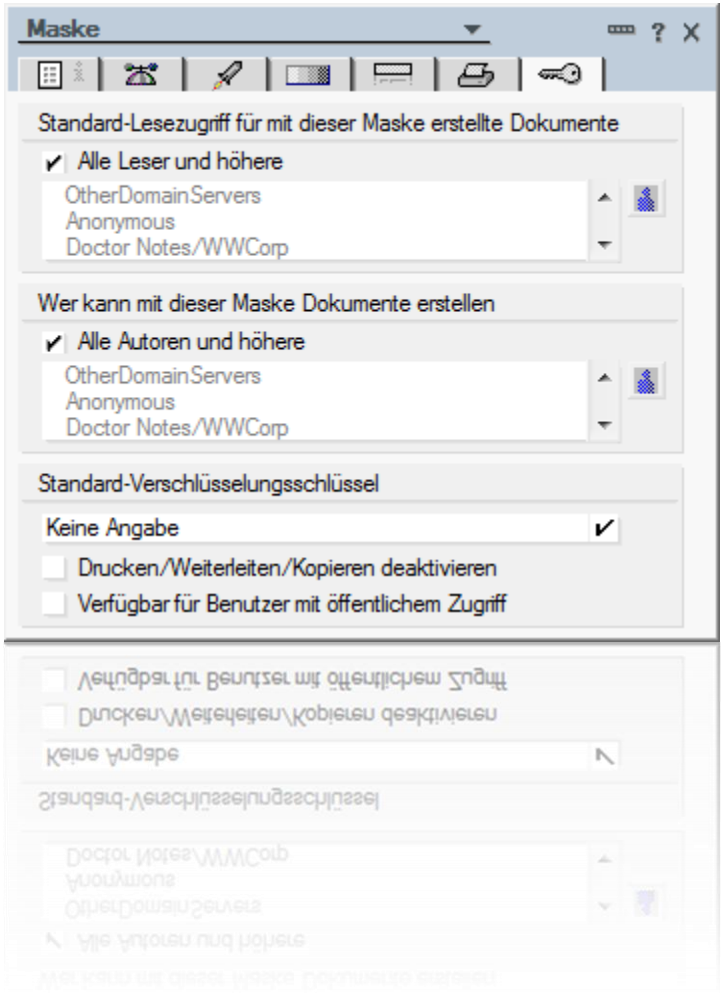


- Authentifiziert den Benutzer / Server
- Prüft Nutzungsberechtigung des spez. Servers (j/n)
- Prüft Nutzungsberechtigung spez. Unterverzeichnisse des Servers (j/n)
- Datenbank-ACL ordnet Benutzer in 7 das Ebenen Rechteschema ein
- Leser-/Autorennamen für Dokumentenberechtigungen

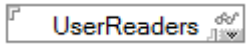
Dokumente nach Typ verrechten

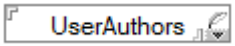
- Maskenvorgabe sieht vor, dass alle Benutzer Dokumente mit der Maske erstellen, und lesen können
- Eingeschränkte Definitionen erstellen zur Laufzeit
  - $Readers

  Felder
- wenig flexibel (keine Anwendungslogik hinterlegbar)
- Definitionsmöglichkeit für „öffentliche Dokumente“
  - $PublicUserAccess

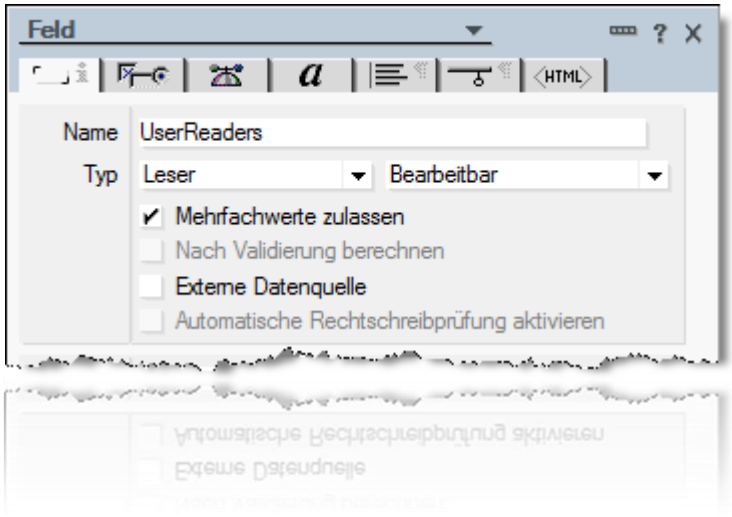


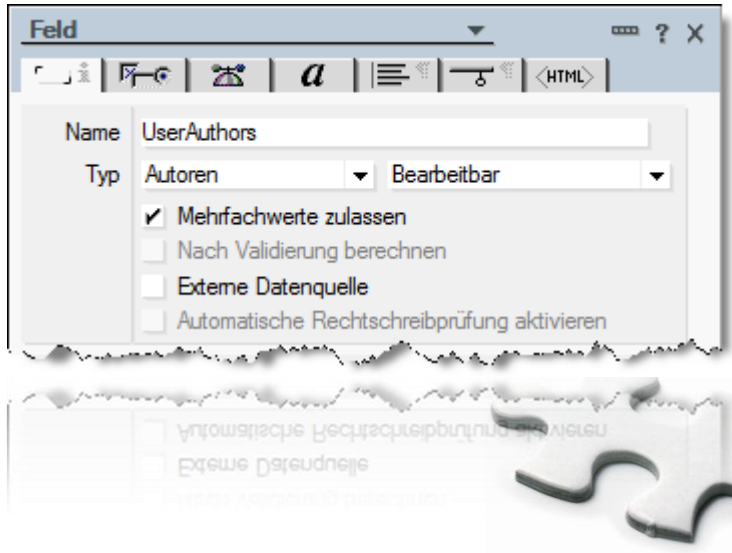
Besondere Feldtypen zur Rechtesteuerung des Dokumentes (optional)

- Lesernamen 

  - steuern Sichtbarkeit des Dokumentes
    - Hierarchische Namen von Benutzern/Servern (kanonisch!!!)
    - „Wildcard-Notation“ möglich
    - Gruppen
    - Rollen
  - automatisch gepflegt durch AdminP
- Autorennamen 

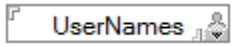
  - regeln die Bearbeitungsrechte am Dokument
    - Hierarchische Namen von Benutzern/Servern (kanonisch!!!)
    - „Wildcard-Notation“ möglich
    - Gruppen
    - Rollen
  - automatisch gepflegt durch AdminP
  - wirkt nur bei ACL Berechtigung „Autor“



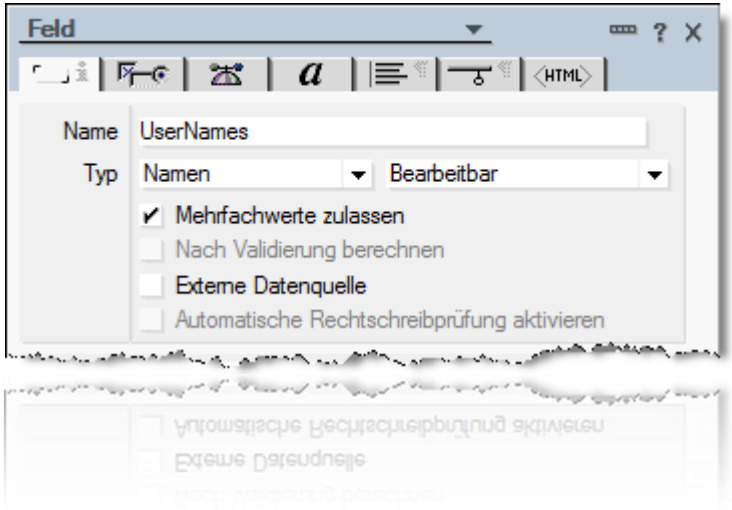


Besondere Feldtypen ohne Rechtesteuerung

- Namensfelder



  - Steuern keine Dokumentberechtigung
  - automatisch gepflegt durch AdminP
  - Erweiterte Auswahlmöglichkeit (gegenüber Textfeldern) bei Werteingaben



Standardfelder mit Rechtebezug

Name	Bedeutung	Wann/wie gesetzt
$UpdatedBy	Liste aller Bearbeiter	automatisch
$Anonymous	Anonyme Dokumente (ohne „UpdatedBy“)	Maskenattribut
$Readers	Leseberechtigung für Dokument	Maskenattribut
$PublicAccess	Kennzeichnung eines Dokumentes als „öffentliches Dokument“	Maskenattribut
$KeepPrivate	Verhindert Kopieren, Drucken	Maskenattribut
Sign	„1“ erzwingt Signatur	- Anwendung -
Encrypt	„1“ erzwingt Verschlüsselung	- Anwendung -
SecretEncryptionKeys	Namen des/der Schlüssel zum Verschlüsseln	- Anwendung -
PublicEncryptionKeys	Namen des/der Schlüssel zum Verschlüsseln	- Anwendung -

Dokumentenberechtigungen und ACL

ACL	Benutzer in vorhandenem Leserfeld	Benutzer in vorhandenem Autorenfeld	Lesen möglich?	Bearbeiten möglich?
Leser	X		JA	NEIN
Autor	X		JA	NEIN
Autor		X	JA	JA
Editor			NEIN	NEIN
Editor	X		JA	JA
Manager			NEIN	NEIN
Manager	X		JA	JA

Live-Demonstration

Verhalten in Sonderfällen

- Lesernamen
  - dürfen leer sein – keine Wirksamkeit
  - dürfen mehrfach bestehen – Kumulation der Leserberechtigten
- Autorennamen
  - dürfen **NICHT** leer sein – sonst keine Autorenzuordnung
  - mindestens Editor Berechtigung zur Bearbeitung erforderlich
  - Autorennamensfelder dürfen mehrfach bestehen – Kumulation der Bearbeitungsberechtigten
- Server respektieren sowohl Lesernamen als auch Autorennamensfelder
  - **Tipp:** Stets ein Autorennamensfeld in jedes Dokument aufnehmen, dass Server- und Administratorenberechtigung sicherstellt
- öffentliche Dokumente
  - lesen (seltener Fall)
    - Benutzer können diese lesen, auch wenn „kein Zugriff“ gewährt wurde
    - Wenn UI Behandlung gewünscht, müssen **alle** involvierten Gestaltungselemente ebenfalls für öffentlichen Zugriff geöffnet werden
  - schreiben (häufig falsch gesetzt)
    - bei eingeschränkten Zugriff jedoch mit Attribut „öffentliche Dokumente erstellen“, wird erst beim Versuch zu speichern geprüft/entschieden, ob dieses erlaubt/möglich ist – ganz schlechte Idee!

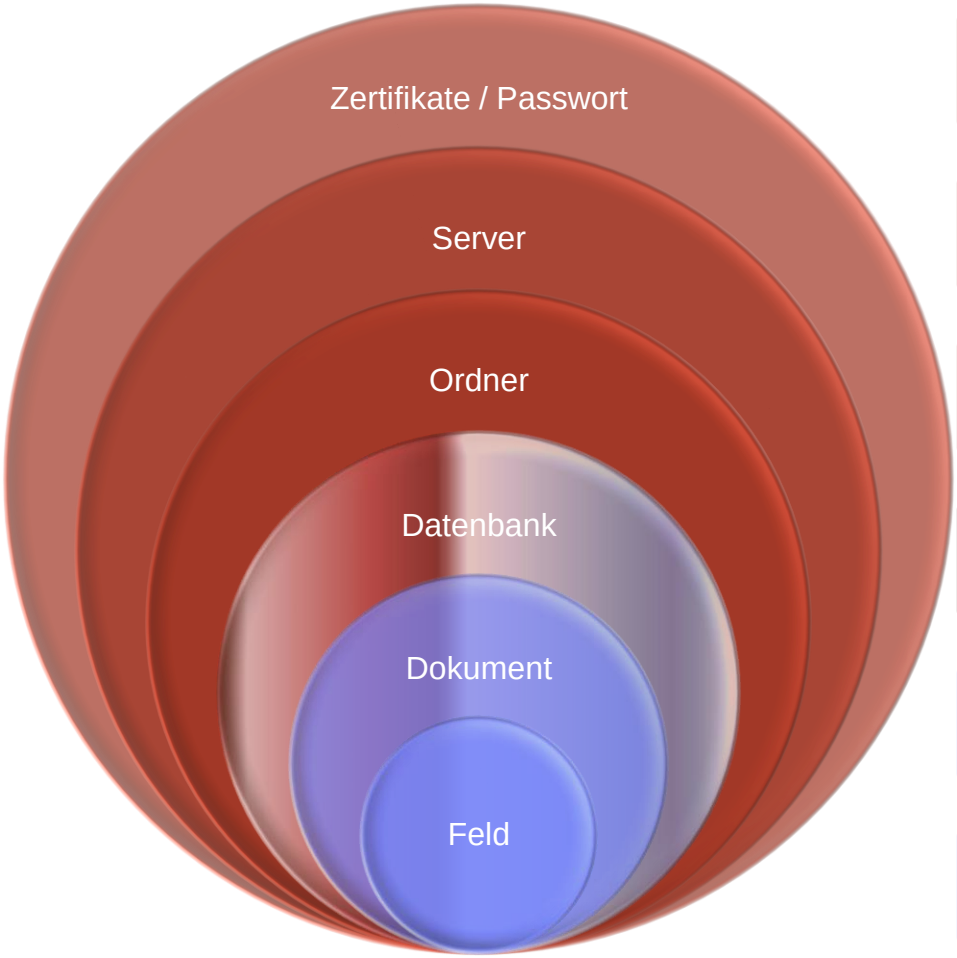
Programmatische Behandlung von Dokumentrechten

- @Name([CANONICALIZE]; NextAuthor)
  - stets kanonische Schreibweise sicherstellen
- Rollennamen stets mit „[„ und „]“ angeben

- document.Authors
  - ermittelt, wer das Dokument bisher bearbeitet hat
- item.isAuthors
  - ermittelt/setzt das Feldattribut „Autoren“
- item.isReaders
  - ermittelt/setzt das Feldattribut „Leser“
- item.isNames
  - ermittelt/setzt das Feldattribut „Namen“

Feldsicherheit

Das Notes/Domino Sicherheitsmodell



Authentifiziert den Benutzer / Server

Prüft Nutzungsberechtigung des spez. Servers (j/n)

Prüft Nutzungsberechtigung spez. Unterverzeichnisse des Servers (j/n)

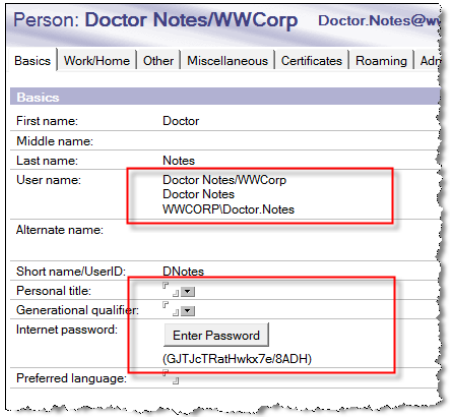
Datenbank-ACL ordnet Benutzer in 7 das Ebenen Rechteschema ein

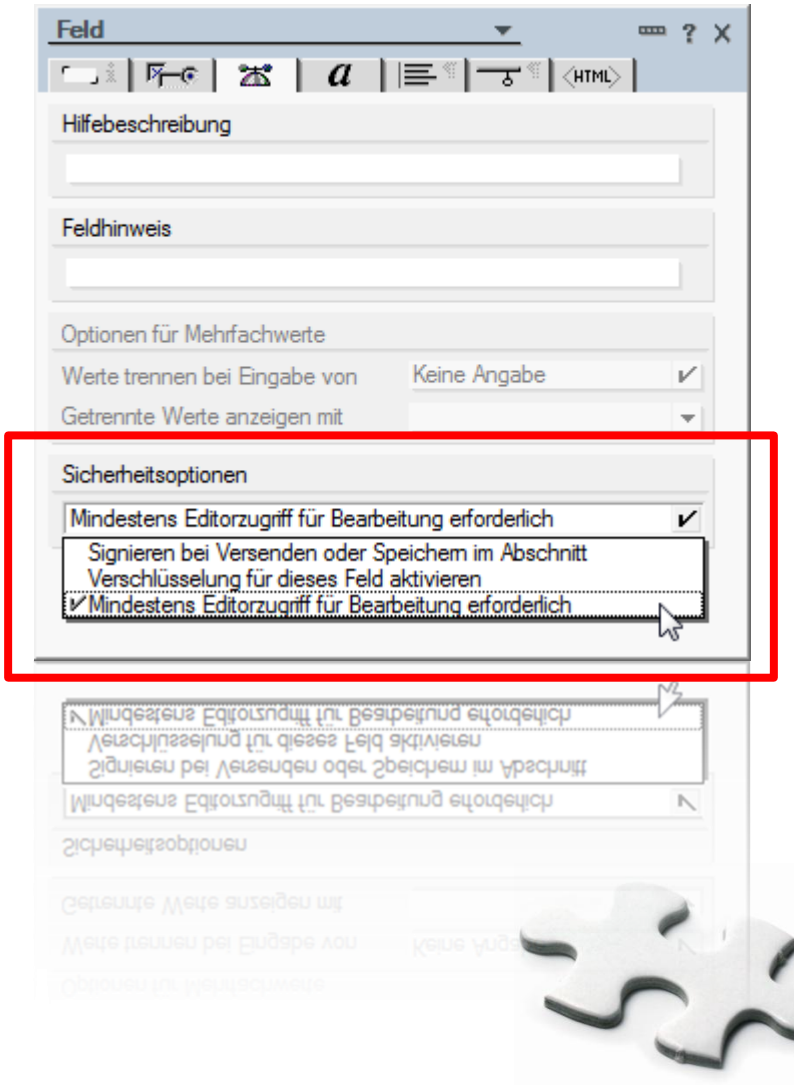
Leser-/Autorennamen für Dokumentenberechtigungen

Feldverschlüsselung erfordert phys. Besitz von Schlüsseln

Bearbeitung nur mit Editor-Recht

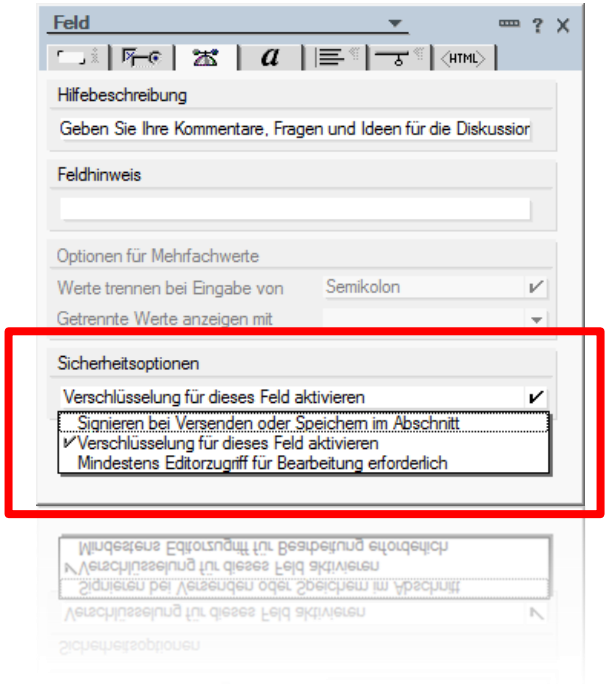
- Beschränkte Bearbeitung von Feldern einrichten
  - Sperrung der Veränderung durch Benutzer (Autor-Berechtigung)
  - Bearbeitung durch Anwendungsverantwortliche / Administratoren (min. Editor-Berechtigung)
- **Beispiel:** Personendokument im Domino Directory





Verschlüsselung beliebiger Felder

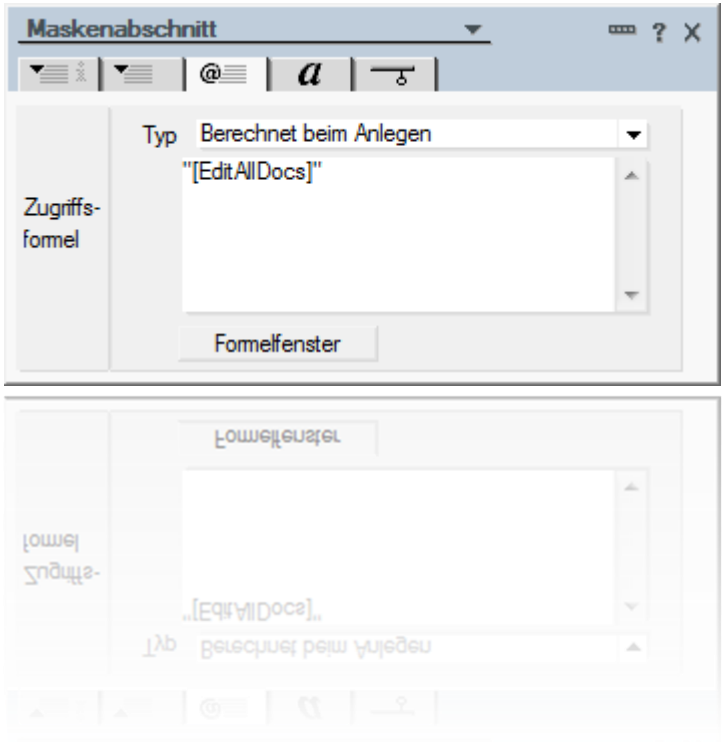
- Verschlüsselung verhindert den Zugriff durch Administratoren (z.B. Vollzugriff) und erfordert Besitz des Schlüssels
- Verschlüsselung der Felder mit Public Keys (aus Domino Directory) oder Dokumentenschlüssel aus der Notes-ID des Benutzers
  - Meistens nur ein Feld pro Maske
  - Bei mehreren Feldern werden alle mit dem gleichen Schlüssel bearbeitet



- Schlüsselauswahl
  - Maskenattribut
  - Reserviertes Feld „SecretEncryptionKeys“
    - erlaubt Berechnung des zu verwendenden Schlüssels durch Formel zur Laufzeit

Zugriffskontrollierter Abschnitt

- Horizontaler Bereich einer Maske, der Teile der Maske einschließt
- Steuert durch zusätzliche (berechenbare Formel) den/die Bearbeiter dieses Bereiches
- Felder, welche signiert werden, können so den „Genehmiger" eines Dokumentes festhalten



Live-Demonstration

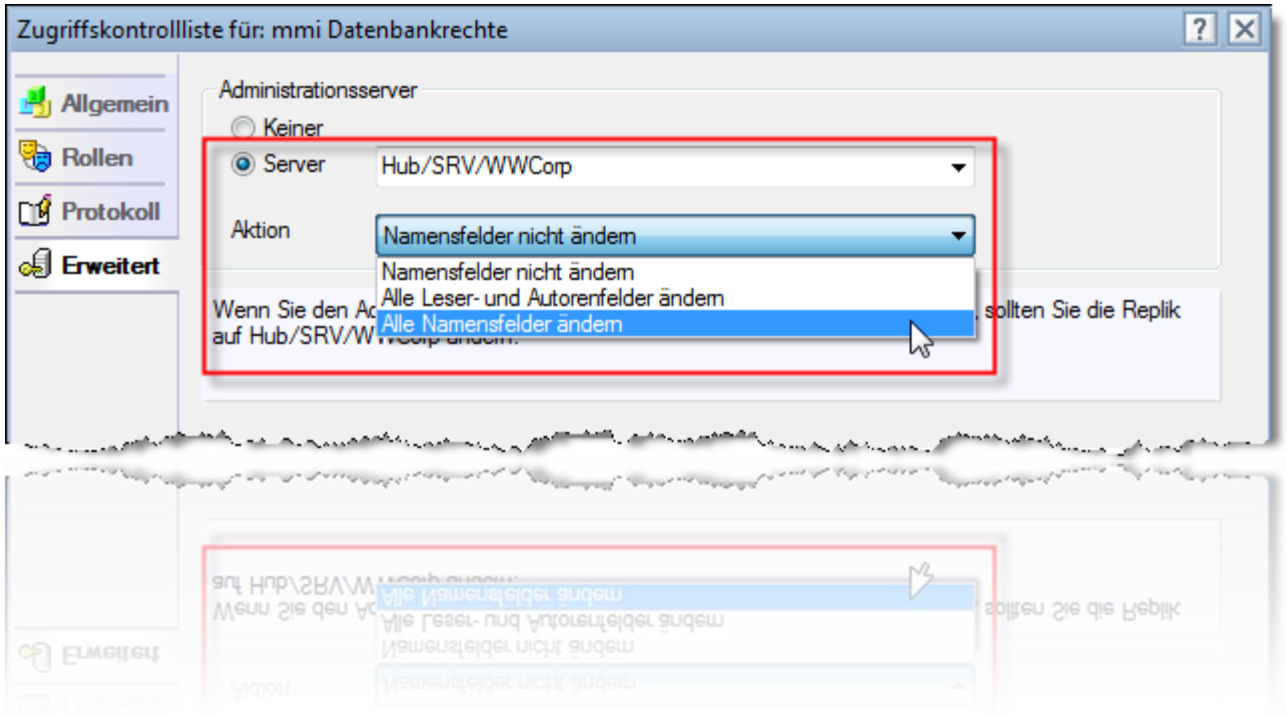
Programmatische Behandlung

- notesDocument.sign
  - signiert das Dokument
- notesDocument.isSigned
  - prüft, ob Dokument signiert ist
- notesDocument.encrypt
  - verschlüsselt das Dokument
- notesDocument.isEncrypted
  - prüft, ob das Dokument verschlüsselt ist

Management von Berechtigungen, Keys und Tokens

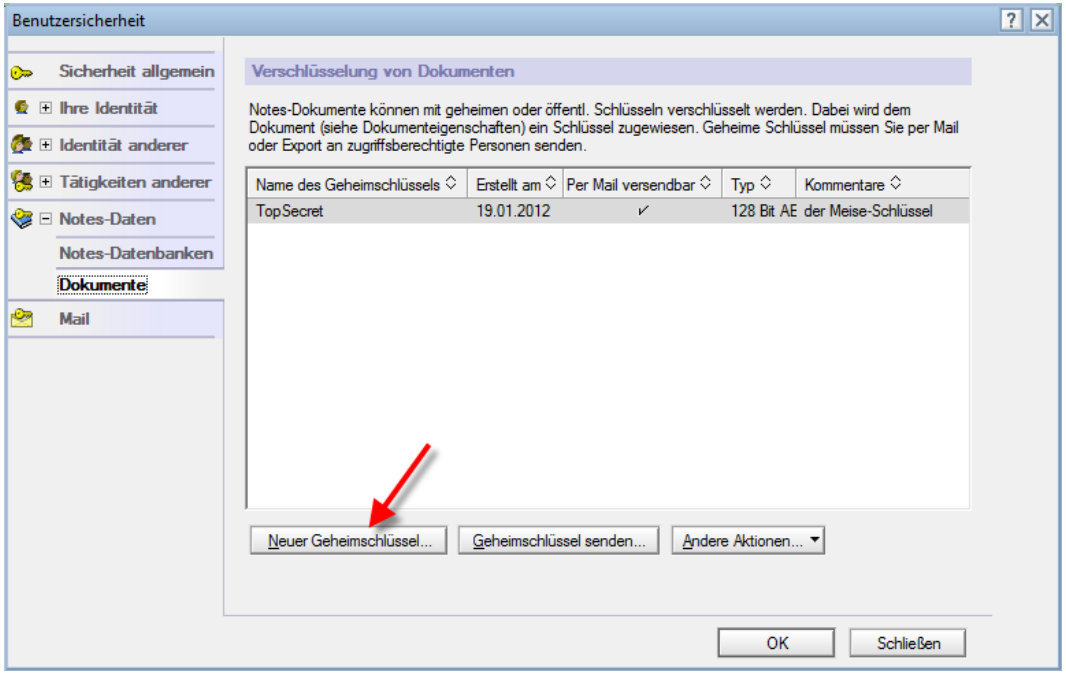
Administrationsprozeß

- Kann Datenbank-(Berechtigungs-)pflege übernehmen
  - Umbenennung / Löschung von Gruppen/Benutzern
- Erfordert Eintrag des Administrationsserver der Datenbank
  - Server muss diese Datenbank dann allerdings auch tragen – Hubserver?!
  - Pflegeregeln müssen der Administration vorgegeben werden



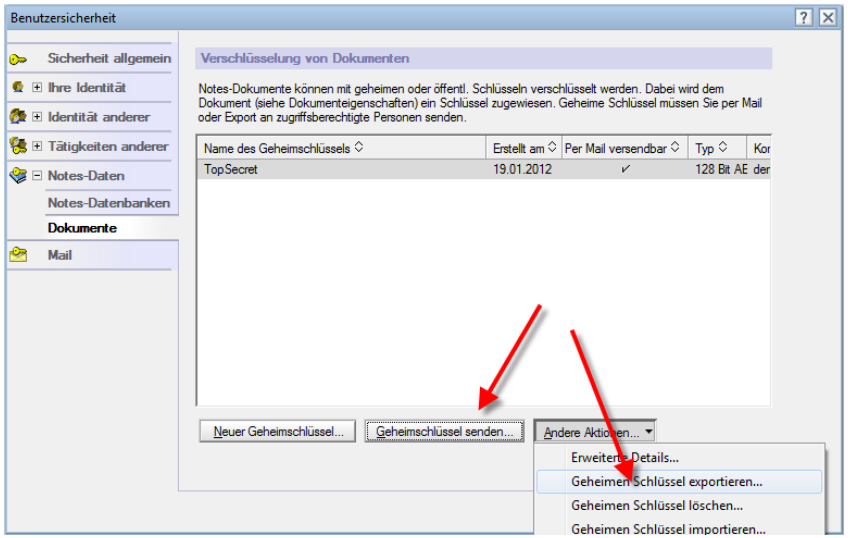
Dokumentenschlüssel: Erstellung

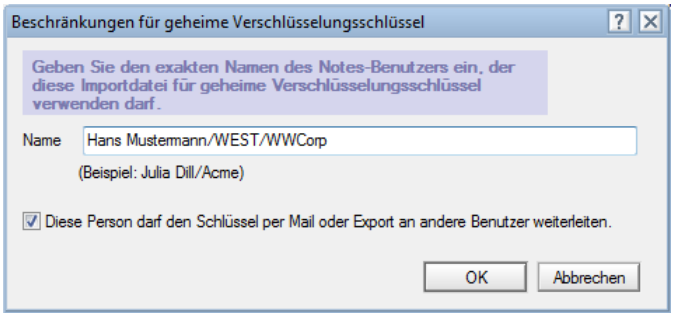
- **ACHTUNG**: Kann unwiderruflichen Datenverlust zur Folge haben, wenn Schlüssel verloren gehen – **nicht** in Produktionsumgebungen **testen** !
- Erstellung von Schlüsseln in Benutzer-IDs
  - separate Pseudo-User für die Verwaltung einsetzen
  - IDs mit Schlüsseln regelmäßig sichern !!! (Dateisystem oder besser ID Vault)
  - Benutzer (bzw. Notes-ID) mit Schlüsseln nie löschen (Schlüssel gehen verloren)

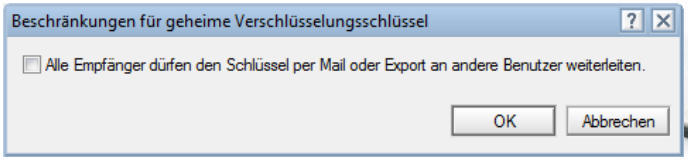


Dokumentenschlüssel: Verteilung

- Besitzer (Eigentümer) eines Schlüssels können diesen Exportieren (Dateiebene) oder per Email versenden
- Beim Export
  - kann/**sollte** festgelegt werden, ob/das die Empfänger diesen Schlüssel **NICHT** weitergeben können
- Beim Versand per Mail
  - kann/**sollte** festgelegt werden, ob/das die Empfänger diesen Schlüssel **NICHT** weitergeben können
  - kann/sollte der Versand des Schlüssels signiert/verschlüsselt erfolgen

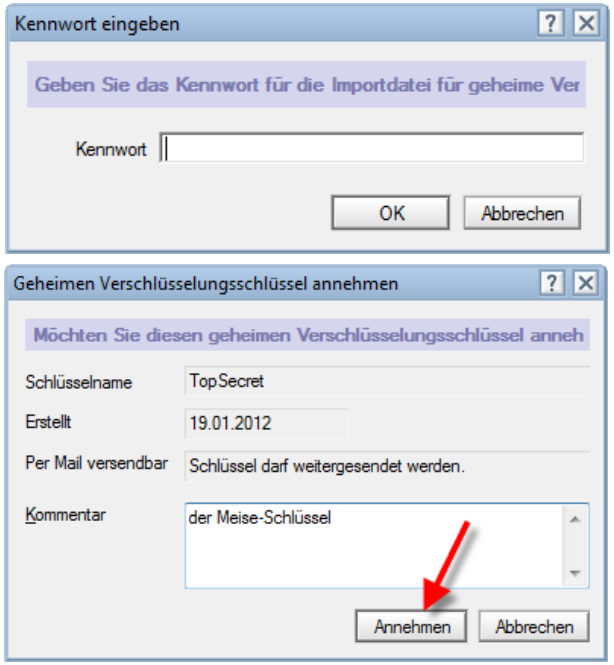


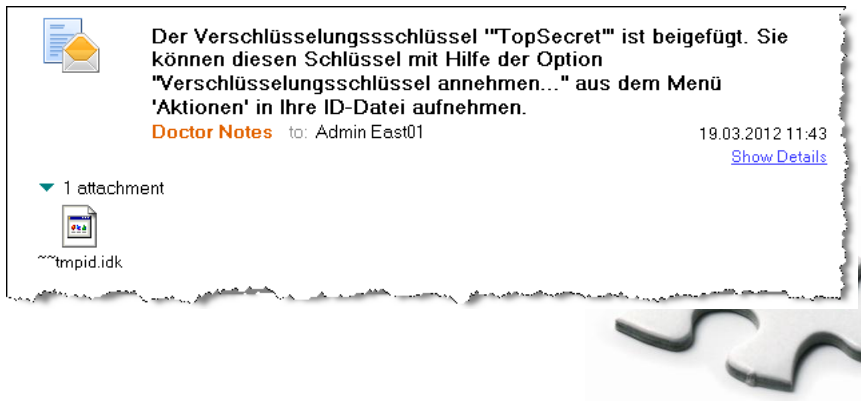




Dokumentenschlüssel: Annahme

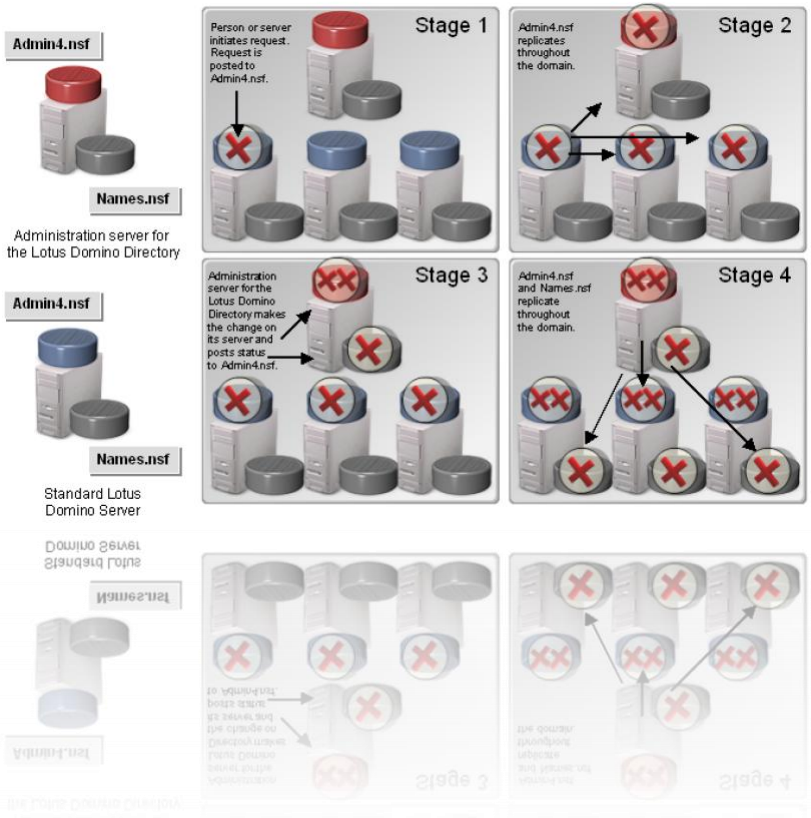
- Berechtigte Empfänger eines Schlüssels können diesen Importieren (Dateiebene) oder per Email annehmen
- Beim Import ist
  - die Datei mit dem Schlüssel auszuwählen
  - das Kennwort einzugeben
- Beim Empfang per Mail ist
  - das Dokument mit einer Anweisung im Betreff sowie Anhang versehen
  - Schlüsselübernahme erfolgt über separaten Menüpunkt „Aktionen – Verschlüsselungsschlüssel annehmen“
  - Bearbeitung/Umgang mit dem Anhang unmöglich





Administrationprozeß (AdminP)

- Servertask auf allen Servern
- Auftrags-/Quittungsprinzip über replizierende Anforderungsdatenbank (Admin4.nsf)
- Ausführungszeit kann schwer vorhergesagt werden
- kennt alle Stellen / kommt an alle Stellen heran, die Namen beinhalten und pflegt diese bei Umbenennung/Löschung von z.B. Personen oder Gruppen
- Anwendungen müssen Felder mit entsprechenden Feldtypen verwenden (Leser, Autoren, Namen), um den AdminP die zu pflegenden Felder offen zu legen
- Datenbank-ACL enthält Vorgaben, ob/wie diese Felder gepflegt werden sollen
  - fehlerhafte Einstellungen führen zum Verlust der automatischen Pflege – benötigen Anwendungscode zur „Reparatur“



Live-Demonstration

Rollen in Datenbanken managen

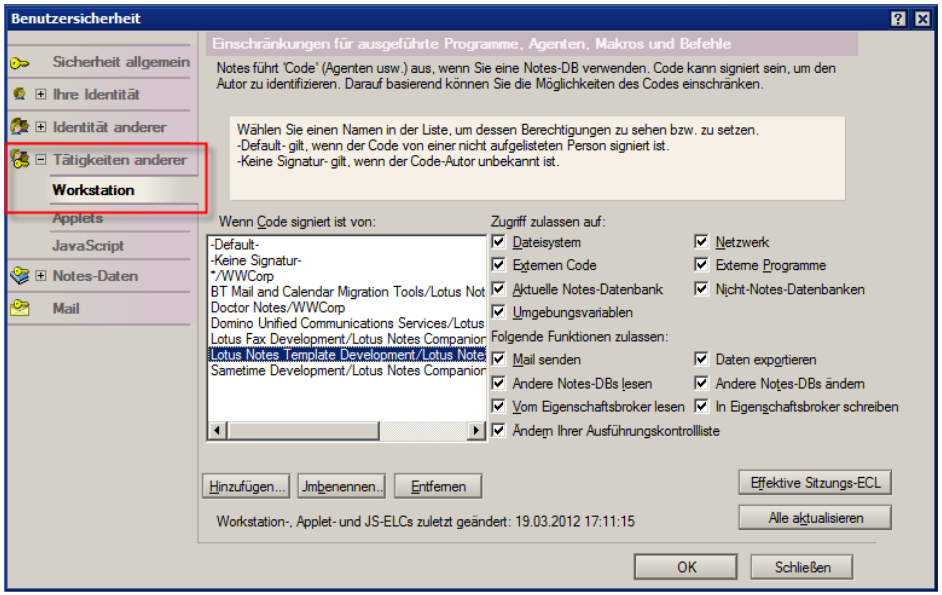
- Dokumentation der Rollen (Namen, Auswirkungen und Zuweisungsregeln) dokumentieren (z.B. im „About-Document“)
- Datenbank-Open-Script

ID Vault

- erfordert min R8.5 Server und R8.5 Clients
- servergestützte „Spezialdatenbank“
- sammelt alle Benutzer-IDs in diesem „Tresor“
- bei Benutzer-Löschungen können IDs dort vorgehalten (deaktiviert werden)
- Auditoren können gezielt einzelne Benutzer-IDs extrahieren und so in den Besitz von Schlüsseln gelangen, um Daten entschlüsseln zu können
- **ACHTUNG:** Freizügige Downloads von Notes.IDs können Sicherheit vollständig unterlaufen

Workstation ECL

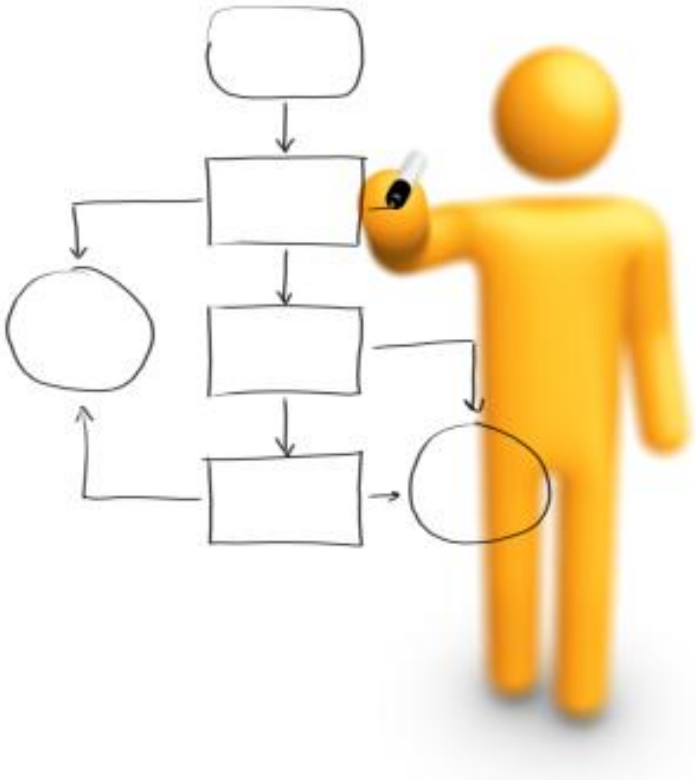
- Execution Control List (ECL
  - Schutz der Workstation vor aktivem Code
  - Java Applets
  - JavaScript
- Regelt die Ausführungsrechte aktiven Codes auf Basis von Signaturen
- Verteilung und Pflege über serverbasierte Richtlinien
- Erfordert bekannte Signaturstrategie von Anwendungen und striktes Application Lifecycle Management
- extrem sinnvoll, doch nachträgliche Einführung ist „stets ein Abenteuer“



Resümee und Ausblick

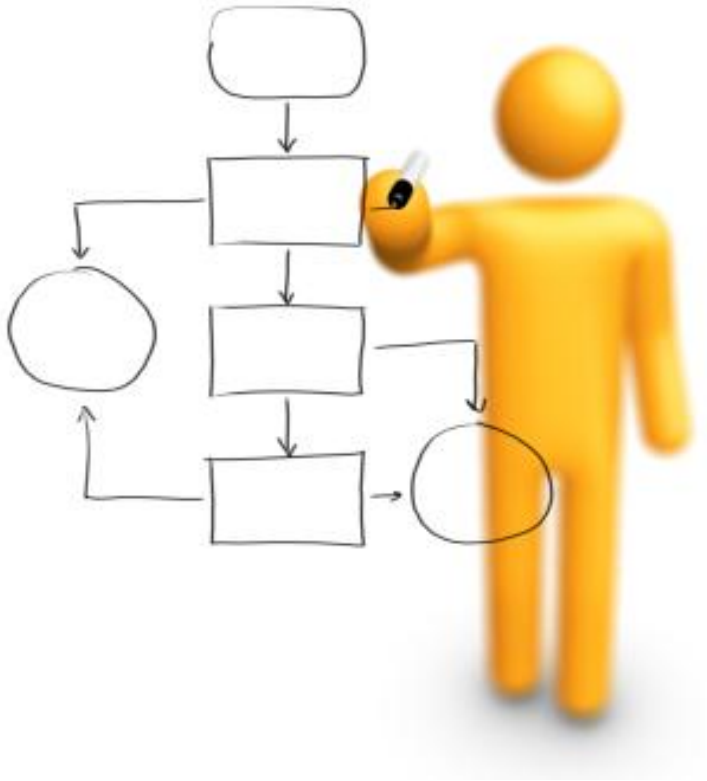
Datenbankrechte

- Keine Datenbank mit „-Default-" Manager betreiben
- „Anonymous" in jede ACL aufnehmen
  - Maximales Recht „Kein Zugriff" oder „Leser"
- ACLs auch in Templates pflegen
  - Schutz des Templates
  - Vorgabe der ACL für die Erstellung von Datenbanken aus der Schablone
- Agenten sollten nicht in Schablonen laufen
- Administration mit Vollzugriff vorsehen, um effiziente Pflege auch im Fehlerfall zu ermöglichen
- „öffentliche Dokumente lesen/schreiben" i.d.R deaktivieren – Sonderfälle
- Rollen dokumentierten und Erstellung automatisieren
- Einstellung für Administrationsserver (inkl. Regeln)
  - dokumentieren/setzen
  - regelmäßig prüfen (separate Werkzeuge erforderlich)



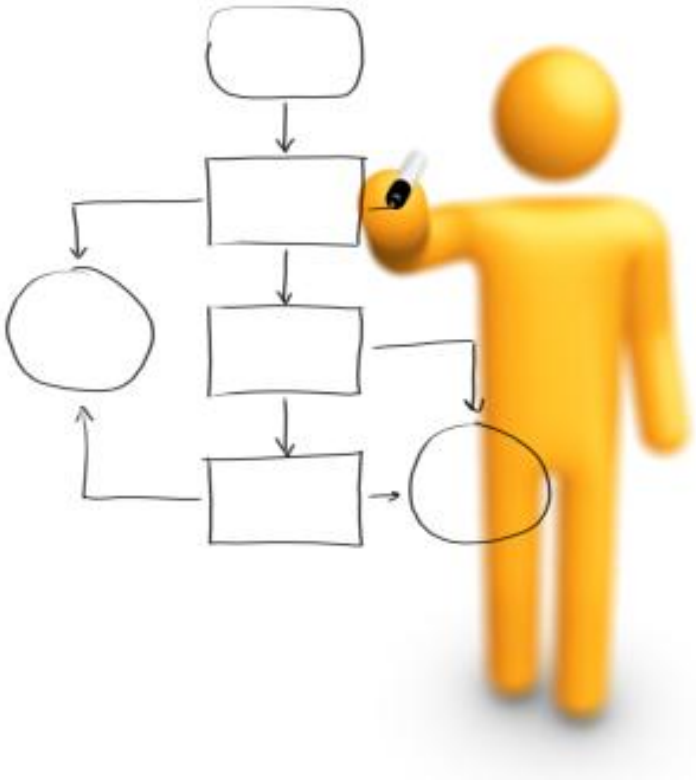
Dokumentenrechte

- Lesernamen / Autorennamen
  - Stets Autorennamenfelder zur Sicherstellung von Replikation und Administration
    - LocalDomainServers
    - LocalDomainAdmins
  - Feldnamen standardisieren
    - Erlaubt Erstellung einer Administrationsansicht zur Prüfung/Pflege
- Ansichtsrechte nicht mit Dokumentensicherheit verwechseln
- „Verbergen wenn" stellt keine Sicherheit dar (genauso wenig wie Maskenwechsel)
- Namensfelder einsetzen
  - Wenn Namen in Textfeldern aufgenommen werden und diese automatisch gepflegt werden sollen (muss nicht immer der Fall sein)



Sicherung von Domino Anwendungsdaten

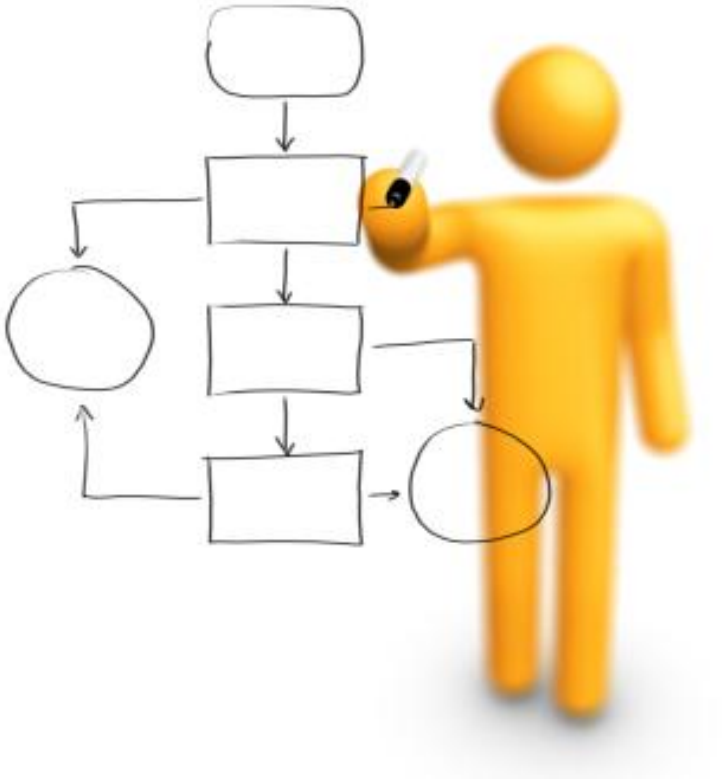
- Verschlüsselung lokaler Datenbanken
  - wertvoll für Notebooks (die keine weiteren Sicherheitsmechanismen einsetzen)
  - verhindert Datenzugang ohne Notes (O/S Ebene)
  - verhindert Datenzugang mit abweichender Notes-ID
  - **Wichtig:** Verfahren für vergessene Kennwörter implementieren!
- Serverdatenbanken **NICHT** verschlüsseln
  - keine (kaum) zusätzliche Sicherheit
  - nur sinnvoll, wenn auch der Server-ID mit Kennwort geschützt ist
  - Server sollten phys. und im Netzwerk geschützt sein!
- Verschlüsselung mit Hilfe von Dokumentenschlüssel
  - NUR einsetzen, wenn durchgängiges Schlüsselmanagementkonzept vorliegt
  - Sicherstellen, dass Schlüssel **NIE** verloren gehen, solange noch verschlüsselte Daten vorliegen
  - Daten sind ohne Schlüsselbesitz nicht zugänglich (keine Hintertür!!)



Nicht zu vergessen:

- Web Access
- Mobile Devices
- Portable Computers

- Traveler Companion zum Lesen verschlüsselter Mails erforderlich



Auf jeden Fall zu vermeiden sind …

- Server „zum Stöbern“ offen halten
- Fehlende Pflege von zielgerichteten Datenbank ACLs
- Vertrauen auf Starteigenschaften der Datenbanken, um Daten zu verbergen
- Ansichtsberechtigungen zum Schutz von unberechtigtem Schutz einsetzen
- Einsatz versteckter Ansichten, um vertrauliche Daten zu schützen
- Verwendung von „verbergen“ Optionen auf Masken
- Einsatz von Maskenformeln, statt Datenbanksicherheit
- Möglichkeit Hintergrundagenten über Browser zu starten
- Unerwünschte Datenbanksuchen zu ermöglichen
- zu häufiger Einsatz von „Editor“ (oder höheren) Berechtigungen
- Signatur von Datenbanken ohne vorherige Prüfung
- Signatur von Datenbanken stets mit der gleichen ID (nicht abgestuft)
- Anwendungsbezogene Lesernamen-Felder einzusetzen, ohne Berücksichtigung von Infrastrukturelementen und Vertreterkonzepten
- …

Sicherheit von Anwendungsdaten

- Vorsicht ist geboten beim Einsatz von Verschlüsselung
  - Verschlüsselte Daten können nicht in Ansichten dargestellt werden
  - Sicherstellen, dass jemand Sicherungskopien der Dokumentenschlüssel erstellt
  - Verhinderung, dass Empfänger von Dokumentenschlüsseln diese weitergeben können
- Dokumentenschlüssel nicht in Server.IDs aufnehmen
  - Benutzer können verschlüsselte Daten einsehen, wenn sie in den Besitz der Server-ID kommen
  - Im Falle einer externen Datenarchivierung wird ein separater Domino-Server erforderlich, in dessen Server-ID alle Schlüssel aufzunehmen sind. Durch zusätzliche Maßnahmen ist zu gewährleisten, dass niemand in der Besitz dieser Server-ID kommt und ggf. diese Server-ID durch Kennwort zu schützen ist

ID Management

- Namensschema festschreiben und dokumentieren
  - hierarisches Namensschema erleichtert Systemadministration
  - hierarisches Namensschema ermöglicht Nutzung in Domino Anwendungen
  - mindestens Server von Benutzern differenzieren
- ID-Vault
  - Storage and distribution
  - Password Reset
  - Auditor Function
- Umbenennungen nur mittels AdminP
- Gruppenmitgliedschaften
- Passwortmanagement
  - Initialkennwörter (Notes und Web)
  - Änderungen
  - SSO

Bitte nicht vergessen …

Noch Fragen offen geblieben?

http://www.mmi-consult.de
http://www.mmi-consult.de/faq
mailto:manfred.meise@mmi-consult.de

Quellennachweise

Quellenhinweise

Quellen:

- IBM Lotus Domino Administrator Hilfe
- IBM Schulungsunterlagen D8L75, D8L76, D8L77, D8L510
- mmi Best Practice sowie Tipps&Tricks
  `http://www.mmi-consult.de/faq`
- Bekannte Lotus Sicherheitslücken
  `http://www.ibm.com/developerworks/lotus/security`
- Developerworks: Designer Wiki
  `http://www-10.lotus.com/ldd/ddwiki.nsf`
- Developerworks: Notes und Domino Wiki
  `http://www-10.lotus.com/ldd/dominowiki.nsf`
- Wikipedia
  `http://de.wikipedia.org/wiki/IT-Sicherheit`

Werkzeuge:

- MartinScott: Noteman
- Ytria: scanEZ, aclEZ, etc.
- mmi: just:NSF