Rollen- und Rechtekonzept

Inhaltsverzeichnis

1. Ziele....1
2. Konzeption zur Realisierung durch Access Control List und im Management-Interface....2
2.1. Ansatz....2
2.2. Safety oder Security....2
2.3. User- und Rechteverwaltung....2
2.4. Definierte Rechte....3
2.5. Vordefinierte Rollen....4
2.6. Probleme....5
3. Konzept zur Realisierung durch eingeschränkten Zugriff auf das Management-Interface....8
3.1. Security-Modell....8
3.2. Schrittweise Umsetzung....8

1. Ziele

Häufiger Kundenwunsch ist, dass unterschiedliche Usergruppen in unterschiedlichem Umfang auf die Funktionen des opsi-Servers zugreifen können. Zum Beispiel sollen User

- Informationen nur sehen, aber nicht bearbeiten können (typisch Helpdesk-Anwender)
- eventuell zusätzlich wenige ausgewählte Funktionen auslösen können (Nachricht an Benutzer, Reboot)
- generell Daten für bestimmte Depots bearbeiten können (lokale Administratoren)

Von daher ist anzustreben, dass entsprechende Rollen definiert und im opsi-Kontext realisiert werden.

Grundsätzlich wäre wünschenswert, dass Rollenmodelle in alle Mechanismen der opsi-Datenhaltung und -Präsentation Einzug halten. Ein entsprechender Ansatz wird in Abschnitt 2 diskutiert.

Wegen des enormen Aufwandes, der zur Realisierung dieses Ansatzes erforderlich ist und – wie noch zu klären wäre – auch zu einem schwierigen Laufzeitverhalten führen kann, wird in Abschnitt 3 eine Konzeption dargestellt, die die Modellierung des gewünschten Verhaltens durch eine Kombination von Zugriffsregelungen auf Systemebene und in der Bedienoberfläche gewährleisten.

2. Konzeption zur Realisierung durch Access Control List und im Management-Interface

2.1. Ansatz

Opsi verfügt über eine 'Access Control List' die Möglichkeit, die Ausführungsrechte auf Methoden auf bestimmte Gruppen zu beschränken. Diese Funktionalität lässt aber noch Wünsche offen:

- Rechteeinschränkungen welche nicht auf einen kompletten Methodenaufruf abbildbar sind, sondern nur den Zugriff auf eine Teilmenge des Ergebnisses eines Methodenaufrufs zulassen (z.B. nur eine bestimmte Clientgruppe)
- Abbildung der Rechte im Managementinterface. Also das Deaktivieren von Controls und Menü-Punkten, welche zu Aktionen führen, die über die Access Control List des Webservice verboten sind und somit zu einem für den user evtl. verwirrenden Fehlermeldung führen
- Einfache Editierbarkeit von Rechten und Rollen im Rahmen einer User- und Rechteverwaltung im Managementinterface. Die hier gesetzten Rechte sollen gleichermaßen in der serverseitigen Access Control List wie im Managementinterface umgesetzt werden.

Durch die Entwicklung einer entsprechenden opsi-Erweiterung sollen diese Wünsche umgesetzt werden.

2.2. Safety oder Security

In einigen Fällen ist es sicherlich ausreichend, ein Recht nur im Managementinterface einzuschränken um versehentliche Fehloperationen (client löschen per rechtem Mausklick) zu unterbinden. Ein so eingeschränkter user hätte trotzdem das Recht über den Webservice einen Client zu löschen wenn er das vorsätzlich möchte. Um wirklich sicherzustellen , das ein in seinen Rechten eingeschränkter user die ihm verbotenen Aktionen mit seinen Rechten nicht ausführen darf, muss dies sowohl auf der Ebene der Access Control List des Webservice als auch im Managementinterface abgebildet werden.

Unsere weiteren Überlegungen basieren auf einer echten Rechteeinschränkung auch im Webservice.

2.3. User- und Rechteverwaltung

- Es werden zunächst 'Rechte' definiert welche fein granuliert bestimmte Zugriffe beschreiben.

- Weiterhin werden 'Rollen' eingeführt. Eine 'Rolle' ist ein Set aus Rechten, das benötigt wird um bestimmte Adminstrative aufgaben ausführen zu können. Es wird eine kleine Anzahl vordefinierter 'Rollen' geben (Fulladmin, UHD,...) sowie die Möglichkeit eigene Rollen zu definieren.
- Den 'Rollen' können dann usernamen zu gewiesen werden, welche damit die in den Rollen definierten Rechte erhalten. Die user selber sind Systemuser, nur deren Namen werden zur Rechteverwaltung verwendet.
- Für die Verwaltung der 'Rechte', 'Rollen' und user wird eine eigene Datenhaltung entwickelt. Diese wird nur für das MySQL-Backend implementiert.

2.4. Definierte Rechte

Das folgende Rechtekonzept folgt typischen Arbeitsvorgängen bei der Administration wie sie auch im opsi-Managementinterface abgebildet sind. Im Rahmen von zukünftigen opsi-Erweiterungen ist zu erwarten das neue Rechte entstehen und entsprechend mit der Erweiterung implementiert werden.

Zur Implementierung vorgesehene 'Rechte':

- useradmin
  Darf user anlegen, löschen, rechte zuweisen
- depots-visible [liste] / all
  Darf depots und deren Clients sehen
- clientgroupview [list] / all
  Darf nur clients der angegebenen Gruppen sehen
- productgroupview [list] / all
  Darf nur produkte der angegebenen Gruppen sehen
- clientmanager
  Darf clients erstellen, löschen, umbenennen, zwischen depots verschieben
- clientgroupmanager
  Darf ClientGruppen anlegen, löschen, und Mitglieder hinzufügen und entfernen
- productgroupmanager
  Darf Produktgruppen anlegen, löschen, und Mitglieder hinzufügen und entfernen
- sendmessage
  Darf Nachrichten an clients versenden

- pushinstall
  Darf on_demand Ereignisse auslösen
- client_control
  darf Clients rebooten, herunterfahren
- wol
  Darf Wake_on_lan schicken
- localboot_product_at_client_change
  Darf für clients action requests, properties und states ändern
- netboot_product_at_client_change
  Darf für clients action requests, properties und states ändern
- server_host_parameter_create
  Darf Server Hostparameter anlegen
- server_host_parameter_change
  Darf Server Hostparameter ändern, löschen
- client_host_parameter_change
  Darf client Hostparameter ändern, zurücksetzen
- hardview
  darf HW-Audit sehen
- softview
  darf SW-Audit sehen
- logview
  darf Logs sehen
- depotpropertymanager
  Darf Eigenschaften der sichtbaren Depots ändern
- licenseview
  Darf lizenzmanagement öffnen und schauen
- licensework
  Darf lizenzmanagement öffnen und ändern

2.5. Vordefinierte Rollen

Beispiele für mögliche vordefinierte Rollen:

	Rolle		
Recht	admin	UHD	Depot-admin
useradmin	+		
Depots-visible	all	all	list
clientgroupview	all	all	all
productgroupview	all	all	all
clientmanager	+		
clientgroupmanager	+	+	+
productgroupmanager	+	+	+
sendmessage	+	+	+
pushinstall	+	+	+
client_control	+		
wol	+	+	+
localboot_product_at_client_change	+	+	+
netboot_product_at_client_change	+	+	+
server_host_parameter_create	+	+	+
server_host_parameter_change	+		
client_host_parameter_change	+	+	+
hardview	+	+	+
softview	+	+	+
logview	+	+	+
depotpropertymanager	+	+	+
licenseview	+	+	+
licensework	+		

2.6. Probleme

Eine Diskussion von verschiedenen Anwendungsszenarien hat gezeigt, das bei eine eingeschränkten Sicht auf die Objekte (depots-visible, clientgroupview, productgroupview) es Referenzen der sichtbaren Objekte auf nicht sichtbare Objekte geben kann. Dies kann zu Problemen führen. Hierzu ein paar Beispiele:

- clientgroupview:
  - Gibt es eine Gruppe Abt5 und Abt5.1 (als Untergruppe von Abt5) und ist die Sicht auf Abt5.1 beschränkt, so muss das Managmentinterface Abt5 trotzdem sehen um den Gruppenbaum korrektdarstellen zu können und gleichzeitig wissen, das es auf diesem Objekt keine Schreibrechte hat.
  - Sehe ich Lizenzmanagment OEM-Lizenzen die nicht sichtbaren Clients zugeordnet sind. Wenn nicht, stimmt die Statistik nicht.
- productgroupview:
  - Ist in meiner Produktgruppe ein Produkt, welches eine

Abhängigkeit zu einem nicht sichtbaren Produkt hat, kann ich nicht sehen ob das abhängige Produkt nun mit auf 'setup' gestellt wird oder nicht.

Um im Managementinterface hier schwere Unklarheiten bei der Bedienung und ebenso Probleme beim internen Handling der Daten zu vermeiden, muss das Managementinterface wahrscheinlich alle (oder zumindest vielmehr) Daten sehen, als es bearbeiten darf. Dies bedingt natürlich eine aufwändige Unterscheidung zwischen Daten mit und ohne Schreibrecht im Managementinterface.

Diese Probleme und Lösungsmöglichkeiten wollen wir an einem möglichst wirlichkeitsnahem Szenario untersuchen und hieran allgemeingültige Lösungsstrategien entwickeln.

Tätigkeit	**h**	**€**
Konzeption		
DB-Design		
Scripte zur Erstellung/Update der DB (opsi-setup)		
Erweiterung MySQL-Backend und Servicemethoden		
Design und Implementierung der user/Rechteverwaltung im opsi-configed		
Anpassung opsi-configed zur Abbildung der Rechte bei Menüs und Controls		
Erweiterung der serverseitigen Access Control Lists zur Berücksichtigung der definierten Rechte		
Anpassung opsi-configed: Ist die Clientsicht auf bestimmte Gruppen beschränkt, so wird bei der Erstellung eines neuen Clients dieser in eine dieser Gruppen aufgenommen		
Anpassung opsi-configed: Ist die Clientsicht auf bestimmte Depots beschränkt, so wird bei der Erstellung eines neuen Clients dieser in eine dieser Depots aufgenommen		
Tests		
Dokumentation		

3. Konzept zur Realisierung durch eingeschränkten Zugriff auf das Management-Interface

3.1. Security-Modell

Die Regelung des Zugriffs erfolgt bei dieser Konzept durch folgenden Ansatz:

- User, die in ihren Rechten bzw. Möglichkeiten der opsi-Administration eingeschränkt sein sollen, werden *nicht* in die Gruppe opsiadmin aufgenommen.
- Sie können per SSH mit einem spezifischen Account auf den opsiserver zugreifen.
- Über diesen Account können Sie ausschließlich Instanzen des opsi-configed starten sowie diese über SSH-getunneltes VNC sehen und bedienen.
- Die opsi-configed-Instanzen, auf die sie Zugriffsrechte haben, sind so konfiguriert, dass nur die für den vorgesehenen User zulässigen Aktionen ausführbar sind.

3.2. Schrittweise Umsetzung

Die Umsetzung startet bei diesem Konzept mit der Realisierung eines Startverfahrens für den configed, das entsprechend dem Security-Modell arbeitet.

Schrittweise können dann Konfigurationen für den configed entwickelt werden, die die Rechte entsprechend Abschnitt 2.4 in einem von Kunden gewünschten Maß umsetzen. Dabei können entsprechende Erfahrungen gesammelt werden. Zugleich wird in der Präsentationsschicht vorbereitet, dass später die Konfigurationen durch im opsi-Server realisierte Rollenkonzepte vorgegeben werden.