# Oracle Key Vault

# HSM und Wallet / Keystore Repository

**Heinz-Wilhelm Fabry**
**ORACLE Deutschland B.V. & Co. KG**
**München**

**Schlüsselwörter**
Hardware Security Modul, HSM, Wallet, Keystore, Passwörter, Repository

**Einleitung**
Oracle Key Vault (OKV) ist ein neues Technologieprodukt, das im August zum Einsatz freigegeben wurde. Es handelt sich dabei um ein Hardware Security Modul (HSM), das ausserdem als Repository für das Speichern von Schlüsseln, Passwörtern und Dateien, die Schlüssel und Passwörter enthalten, genutzt wird.

Oracle Produkte speichern Passwörter und Schlüssel zum Beispiel für Oracle Advanced Security Transparent Data Encryption (TDE), *external password stores* und andere Funktionen in Dateien, die Wallets (Versionen 10 und 11) oder Keystores (Version 12) genannt werden. Für den Bereich TDE gibt es auch seit der Datenbankversion 11.2 die Möglichkeit, statt der Wallets / Keystores HSMs einzusetzen.

HSMs sind zum Beispiel sinnvoll, wenn man befürchten muss, dass Wallets unbefugt kopiert werden, wenn Daten und dazugehörige Wallets aufgrund gesetzlicher oder anderer Bestimmungen auf unterschiedlichen Rechnern gespeichert werden müssen oder wenn Schlüssel zur Konsolidierung oder zur gemeinsamen Nutzung zentral gespeichert werden sollen.

Ein 'Wallet Repository‘ ist zum Beispiel sinnvoll, wenn nicht ausgeschlossen werden kann, dass versehentlich lokale und nicht gesicherte Wallets gelöscht werden. Die Repository Lösung erhält ausserdem die Unabhängigkeit der betroffenen Datenbank: Bei Verlust der Verbindung zwischen Datenbank und OKV bleibt die Datenbank voll funktionsfähig, weil nur das lokale Wallet für den Betrieb benötigt wird.

Beide Anwendungsfälle - Wallet Repository und HSM - sollen hier skizziert werden.

**Installation und Konfiguration**
OKV wird in Form eines ISO Image als Software Appliance geliefert. Das heisst, dass die Kundin / der Kunde einen X86-64bit Rechner mit mindestens 4GB RAM und 500GB Speicherplatz zur ausschliesslichen Nutzung für OKV zur Verfügung stellt. Auf diesem Rechner installiert er / sie OKV. Dabei werden das Betriebssystem Oracle Linux, eine Oracle Datenbank Enterprise Edition einschliesslich TDE, Oracle Database Vault und Virtual Private Database (VPD) sowie eine graphische Benutzeroberfläche, die Oracle Key Vault Management Console, installiert. Die Console wird über einen Browser bedient, der für Installation und Konfiguration - einschliesslich Tastaturbelegung - auf die englische Sprache eingestellt sein sollte. Das gesamte System ist gehärtet.

Der Installationsprozess erwartet nur zwei Eingaben: Angaben zur IP Adresse (IP Adresse, Mask und sofern nötig Gateway) für den OKV und ein Einmalpasswort, das OKV als Passphrase bezeichnet und mit dem man sich erstmalig in der Console einloggen kann.

Nach der Installation dürfen die Komponenten des OKV in keiner Weise verändert, sondern nur in festgelegtem Ausmass konfiguriert werden. Die Konfiguration umfasst zunächst die Verteilung von Aufgaben und Funktionen. Sie können auf eine Person oder auf mehrere Personen verteilt werden. Dies geschieht über die Console.

Es werden eingerichtet

- ein Key Administrator, der zum Beispiel den Zugriff auf Wallets und Schlüssel kontrolliert und weitere Key Administratoren einrichten kann
- ein System Administrator, der zum Beispiel zuständig ist für das Starten und Stoppen des OKV, für Backups / Recovery, das Einrichten von Hochverfügbarkeit und von Endpoints und Benutzern sowie für das Einrichten weiterer Administratoren
- ein Audit Manager, der den Audit Trail des OKV verwaltet und weitere Audit Manager einrichten kann
- ein Root User, der sich bei Bedarf, zum Beispiel zum Ausführen festgelegter Skripte oder im Recoveryfall, als Benutzer *root* einloggen kann
- ein Support User, der sich bei Bedarf mit SSH anmelden kann

Ausserdem wird ein Notfallpasswort für den Recoveryfall oder für ein Einloggen nach Verlust des Systemadministratorpassworts festgelegt.

**Einsatz als Repository**
OKV kann zur Zeit als Backup Repository für folgende Dateitypen verwendet werden

- Oracle Wallets und Keystores
- Java Keystores (JKS)
- Java Cryptography Extension Keystores (JCEKS)
- SSH Key Files und
- Kerberos Keytabs

Dateien dieser Typen werden aktuell nur von Rechnern akzeptiert, die auf den Betriebssystemen Linux x86-64 (Versionen 5 und 6) sowie Solaris (Versionen 10 und 11) betrieben werden. Während Oracle Wallets und Keystores von diesen Systemen gelesen und in sogenannten *virtual wallets* gespeichert werden, erfolgt das Speichern zum Beispiel von Kerberos Keytabs als LOBs.

Von OKV unterstützte Rechner werden Endpoints genannt. Diese Endpoints werden durch sogenannte Endpoint Administratoren verwaltet. Endpoint Administratoren sind im Fall von Datenbankservern in der Regel die zuständigen Datenbankadministratoren.

Während es für Testumgebungen durchaus möglich ist, dass Endpoint Administratoren Endpoints selbständig einrichten (OKV Terminologie *self-enrollment*), werden Endpoints in Produktivumgebungen normalerweise gemeinsam von OKV System Administrator und Endpoint Administrator eingerichtet (OKV Terminologie *administrator-initiated enrollment*). Dabei übernimmt der OKV System Administrator die Führungsrolle. Er vergibt einen Namen für den neuen Endpoint, legt Typ und Betriebssystem des Endpoints fest und erzeugt schliesslich ein sogenanntes Token, das er später dem Endpoint Administrator übermittelt. Der folgende Screenshot zeigt unter anderem, dass ein Endpoint namens JAHRESKONFERENZ eingerichtet wurde sowie das zugehörige Token.

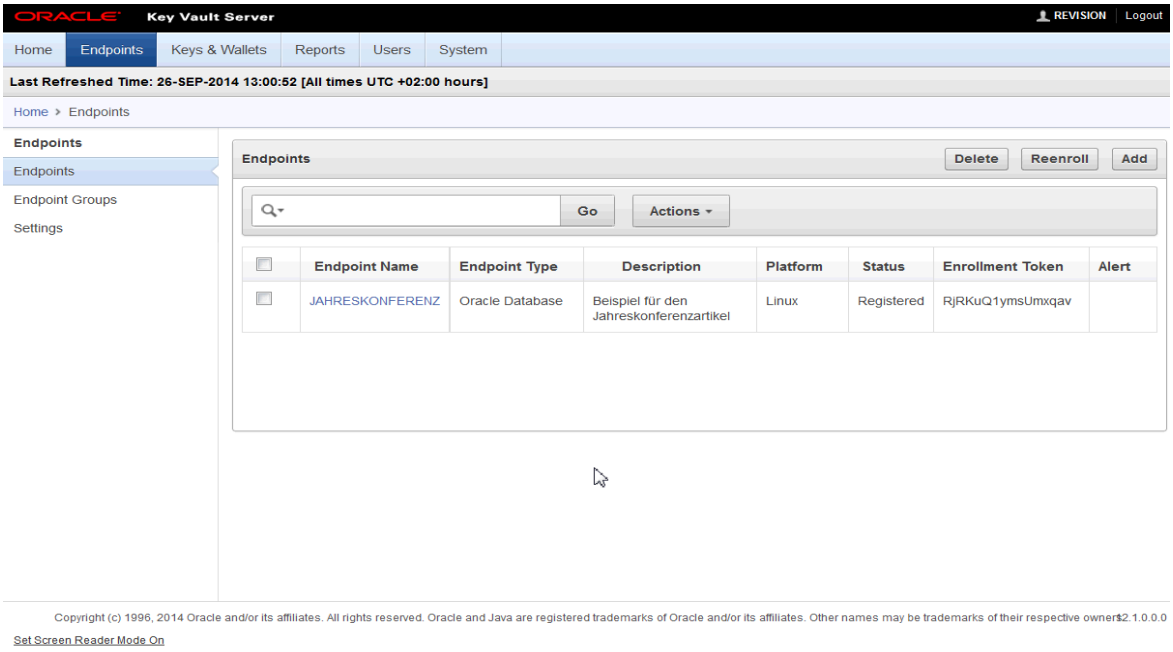


Abb. 1: Eingerichteter Endpoint

Nachdem der Endpoint eingerichtet ist, erzeugt ein Benutzer mit der Rolle des Key Administrators ein virtuelles Wallet (*virtual wallet*). In diesem virtuellen Wallet werden die Informationen aus den Wallet / Keystore Dateien gespeichert, die lokal auf den Rechnern liegen. Im Beispiel für diese Jahreskonferenz liegen die Rollen des System Administrators und des Key Administrator in einer Person namens REVISION. Das folgende Bild zeigt das eingerichtete virtuelle Wallet JKwallet und die dafür eingerichteten Zugriffsberechtigungen. Der OKV Administrator teilt dem Endpoint Administrator gemeinsam mit dem Token auch den Namen des virtuellen Wallets mit.

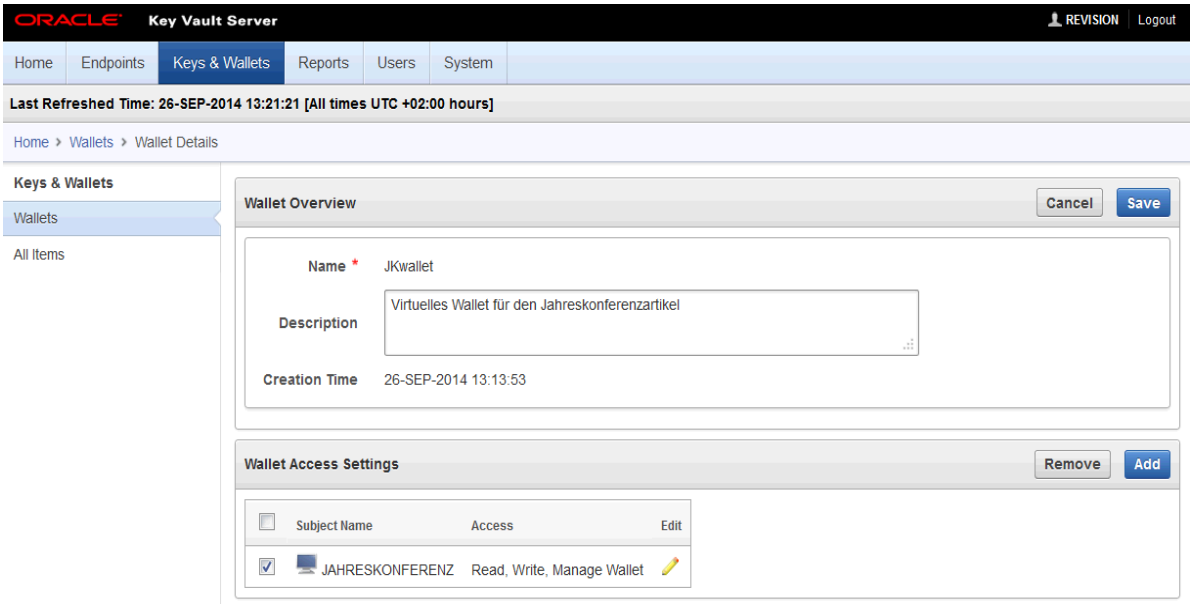


Abb. 2: Eingerichtetes virtuelles Wallet

Der DBA ruft dann im Browser die OKV Konsole auf, logged sich allerdings nicht ein, sondern klickt am Fuss der Seite auf einen Link mit dem Namen *Endpoint Enrollment and Software Download*. Auf der Seite, die sich dann öffnet, wird das Token eingegeben. Ausserdem werden auch hier die Angaben zum Typ (Database) und Betriebssystem (Linux) gemacht. Nach dem Anklicken von *Submit Token* und *Enroll* wird die Datei *okvclient.jar* zum Speichern angeboten. *okvclient.jar* enthält das Hilfsprogramm *okvuti*l, das für das Zusammenspiel zwischen Datenbank und OKV benötigt wird, ein TLS Zertifikat zur Verschlüsselung der Kommunikation zwischen Datenbank und OKV sowie weitere Komponenten (siehe unten).

Nach dem Herunterladen wird die Datei *okvclient.jar* auf dem Datenbankserver in einem beliebigen Verzeichnis entpackt. Benötigt wird dazu mindestens JDK 1.5. Ausserdem ist die Variable JAVA_HOME zu setzen, und auch der Pfad zum Java Executable sollte explizit gesetzt sein.

Während des Entpackens wird ein Passwort angegeben. Für das Beispiel hier lautet es "DOAG.nov2014" (ohne Hochkommata). Die Eingabe erfolgt verdeckt, aber das Passwort wird später noch gebraucht und deshalb hier genannt.

```
[oracle@hwf Desktop]$ java -jar okvclient.jar -d
                            /oracle/app/oracle/product/okvutil -v
Detected JAVA_HOME: /usr/lib/jvm/java-1.6.0-openjdk-
                            1.6.0.0.x86_64/jre
Enter new Key Vault endpoint password (RETURN for auto-login):
Confirm new Key Vault endpoint password:
Oracle Key Vault endpoint software installed successfully.
```

Nun kann ein vorhandenes Wallet nach OKV geladen werden. Als Beispiel dient ein TDE Wallet, dessen Inhalt mit folgendem Befehl in das virtuelle Wallet geladen wird.

```
./okvutil upload
-l "/oracle/app/oracle/product/11.2.0/dbhome_1/network/admin"
-t wallet -g "JKwallet"
Enter source wallet password:
Enter Oracle Key Vault endpoint password:
Upload succeeded
```

Zur Erläuterung: Der Aufruf erfolgt hier aus dem Verzeichnis *bin* des Verzeichnisses, in dem die Datei *okvclient.jar* oben ausgepackt wurde. Nach der Option *-l* wird das Verzeichnis angegeben, in dem das Wallet gespeichert ist, nach der Option *-t* der Typ des Wallets, hier ein Oracle Wallet. Nach der Option *-g* wird angegeben in welchem virtuellen Wallet gespeichert werden soll, hier in dem zuvor angelegten *JKwallet*. Nach der Abfrage des Wallet- und des OKV Endpoint Passwortes (wurde oben im Befehl *java -jar* angegeben), beginnt das Laden.

Nicht nur die Meldung *Upload succeeded* zeigt, dass die Aktion erfolgreich war, sondern auch der Blick auf die relevante Seite in OKV:

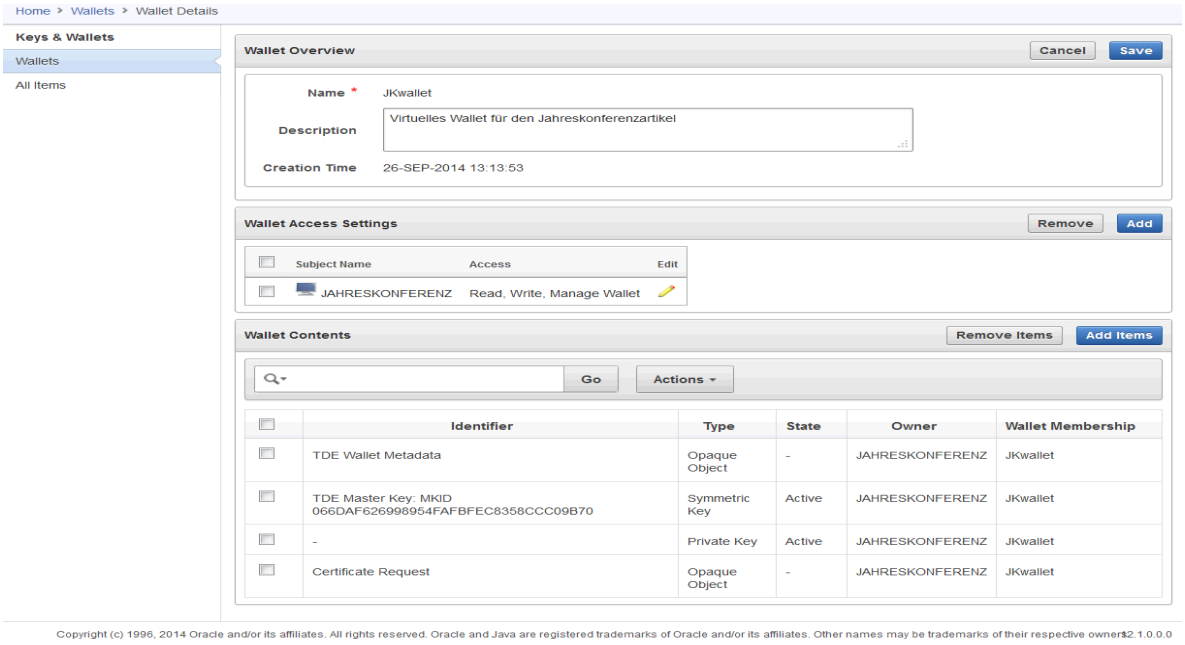


Abb. 3: Gespeichertes Wallet

Aus diesem virtuellen Wallet würde sich, falls das Originalwallet verloren geht, ein neues, absolut identisches Wallet herunterladen lassen.

**Einsatz als HSM für TDE**

Der Einsatz von OKV als HSM wird in der OKV Terminologie als *direct connection* bezeichnet. Dazu wird mindestens eine Oracle Datenbank der Version 11.2 benötigt. Ausgangsbasis kann eine Datenbank sein, die noch nicht für TDE konfiguriert wurde, aber auch die Migration von Wallet / Keystore zu OKV wird unterstützt. Hier wird anhand einer Datenbank Version 11.2.0.4 die Variante für eine Datenbank beschrieben, die zuvor noch nicht für TDE konfiguriert wurde.

Auch wenn man OKV als HSM nutzt, muss die dazu vorgesehene Datenbank als Endpoint eingerichtet werden. Die Vorgehensweise wurde oben beschrieben. Das Einrichten eines virtuellen Wallets entfällt, aber das Herunterladen und Entpacken der Datei *okvclient.jar* erfolgt genau wie oben. Für dieses Beispiel wurden beide virtuellen Maschinen, die, in der OKV installiert wurde, und die, in der die Datenbank läuft, zurückgesetzt. Dann wird der Endpoint wie oben neu angelegt, *okvclient.jar* heruntergeladen und wie oben ausgepackt.

Zusätzlich muss nun jedoch eine Library namens *liborapkcs.so* verfügbar gemacht werden, die den Zugriff auf PKCS#11 kompatible HSMs ermöglicht - und als solches soll OKV jetzt genutzt werden. Die Library ist ebenfalls Teil der Datei *okvclient.jar*. Da das System die Library in einem bestimmten Verzeichnis erwartet, stellt Oracle ein Skript namens *root.sh* zur Verfügung, das das Verzeichnis anlegt und die Library dort ablegt. Das Skript ist im Verzeichnis *bin* von *okvutil* zu finden und muss als Benutzer *root* ausgeführt werden.

```
[root@hwf bin]# ./root.sh
Creating directory: /opt/oracle/extapi/64/hsm/oracle/1.0.0/
Copying PKCS library to /opt/oracle/extapi/64/hsm/oracle/1.0.0/
Setting PKCS library file permissions
Installation successful.
```

Damit ist die Datenbank für das Arbeiten mit OKV vorbereitet. Es muss nur noch TDE eingerichtet werden. Dazu wird als erstes der Paramter ENCRYPTION_WALLET_LOCATION in die Datei *sqlnet.ora* eingefügt.

```
ENCRYPTION_WALLET_LOCATION=(SOURCE=(METHOD=HSM))
```

Wie gewohnt erzeugt dann das Statement ALTER SYSTEM den ersten Masterkey. Das Passwort, das diesem Befehl übergeben wird, ist das, das oben beim Entpacken von *okvutil* festgelegt wurde, nämlich "DOAG.nov2014". Das Tablespace wird angelegt und belegt damit, dass das Arbeiten mit dem OKV auch funktioniert.

```
SQL orcl> ALTER SYSTEM set encryption key IDENTIFIED BY
  2  "DOAG.nov2014";

System altered.

SQL orcl> CREATE TABLESPACE sicheristsicher
  2  DATAFILE '/home/oracle/dateiname.dbf' SIZE 10M
  3  ENCRYPTION USING 'AES256' DEFAULT STORAGE (ENCRYPT);

Tablespace created.

SQL orcl> SELECT tablespace_name, encrypted FROM dba_tablespaces
  2  WHERE tablespace_name like 'SICHER%';

TABLESPACE_NAME                ENCRYPTED
------------------------------ ---------
SICHERISTSICHER                YES
```

Auch der nächste Screenshot belegt, dass das Zusammenspiel der Systeme funktioniert und dass in OKV ein Masterkey angelegt wurde.

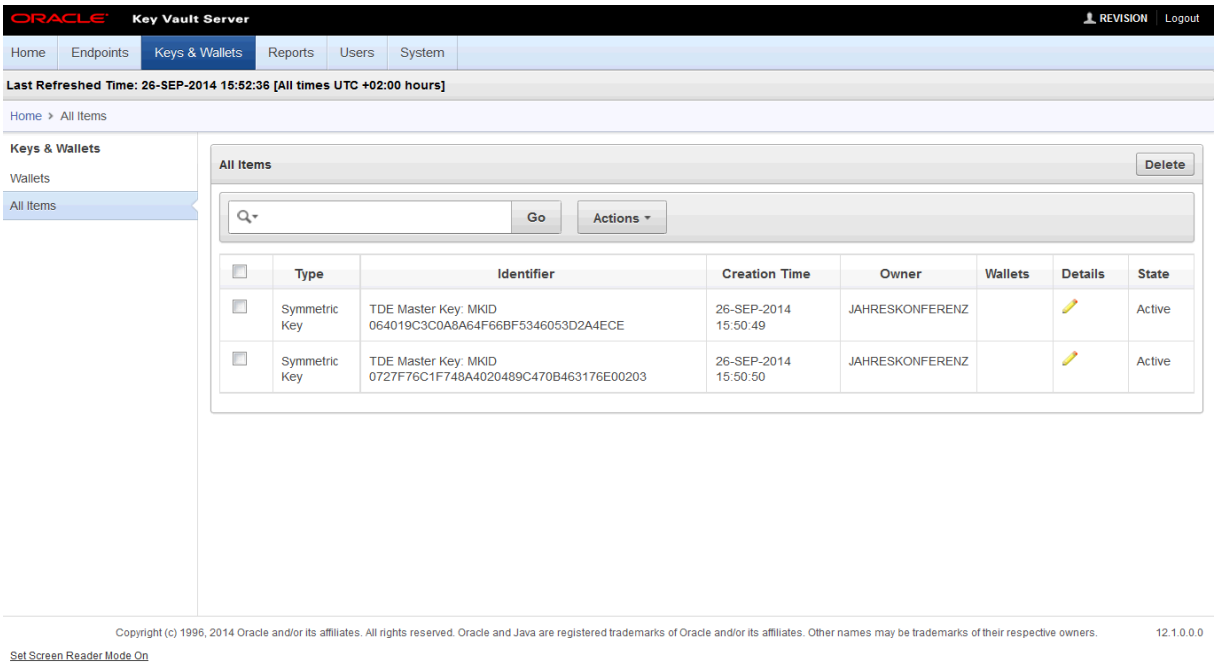


Abb. 4: In OKV für *direct connection* angelegte Masterkeys

**Hochverfügbarkeit und Backup**
Die Aspekte Hochverfügbarkeit und Backup / Recovery sind für den Einsatz von OKV sicherlich von entscheidender Bedeutung. Vor allem für den Einsatz im *direct connection* Szenario ist da zunächst die Frage nach der Verfügbarkeit zu stellen. Hier bietet OKV die Möglichkeit einer einfachen Spiegelung des gesamten OKV Servers. Eine produktive Nutzung von OKV wird ohne diese Spiegelung nicht empfohlen.

Zur Sicherung der gespeicherten Informationen verfügt OKV über eigene Backup Routinen. Sie erlauben sowohl ein Scheduling von Backups als auch 'spontane' Backups. Es handelt sich dabei um logische Backups, die verschlüsselt entweder lokal oder *remote* gespeichert werden (der *remote* Server muss dazu das SCP Protokoll unterstützen). Während lokal immer nur eine einzige Backupversion unterstützt wird und neue Backups alte überschreiben, können *remote* mehrere Versionen gespeichert werden. So wird nicht nur ein komplettes Restore des OKV unterstützt, sondern auch eine Art Point-in-Time Recovery oder 'Rollback' zu einem Zeitpunkt vor dem aktuellen Backup.

**Nachvollziehbarkeit und Berichte**
Lösungen im Security Umfeld sind nicht nur im Hinblick auf ihre praktische Verwendbarkeit zu bewerten, sondern auch im Hinblick darauf, was sie Administratoren, Auditoren und Sicherheitsbeauftragten an Informationsmöglichkeiten bieten.

Alle Informationen sind in OKV über die Console zugänglich. Es können immer nur die Informationen gesehen oder geändert werden, für die die gerade angemeldete Person aufgrund ihrer Rolle auch zugriffsberechtigt ist.

- Alle Informationen zu Schlüsseln und Endpoints können sichtbar gemacht werden, zum Beispiel Verwendung des Schlüssels, letzter Zugriff usw.
- Ein konfigurierbares Alert System macht darauf aufmerksam, wann Schlüssel erneuert werden müssen, wann Passwörter geändert werden müssen, wann Backups eventuell unabhängig von einem vorhandenen Schedule gemacht werden müssen und anderes.
- Ein eigenes Auditing macht sämtliche Aktionen innerhalb des OKV und die Zugriffe auf die Inhalte transparent.
- Vorgefertigte Berichte erlauben auch dem Management oder einem Auditor, sich einen Überblick über relevante Informationen zu verschaffen.

**Kontaktadresse:**
Heinz-Wilhelm Fabry
ORACLE Deutschland B.V. & Co. KG
Riesstr. 25
D-80992 München

Telefon: +49 (0) 89-14301534
E-Mail: heinz-wilhelm.fabry@oracle.com
Internet: www.oracle.com