Oracle Transparent Data Encryption im Compliance Umfeld

Thomas Knauber , Atos Worldline GmbH

13/11/2011

Atos Worldline

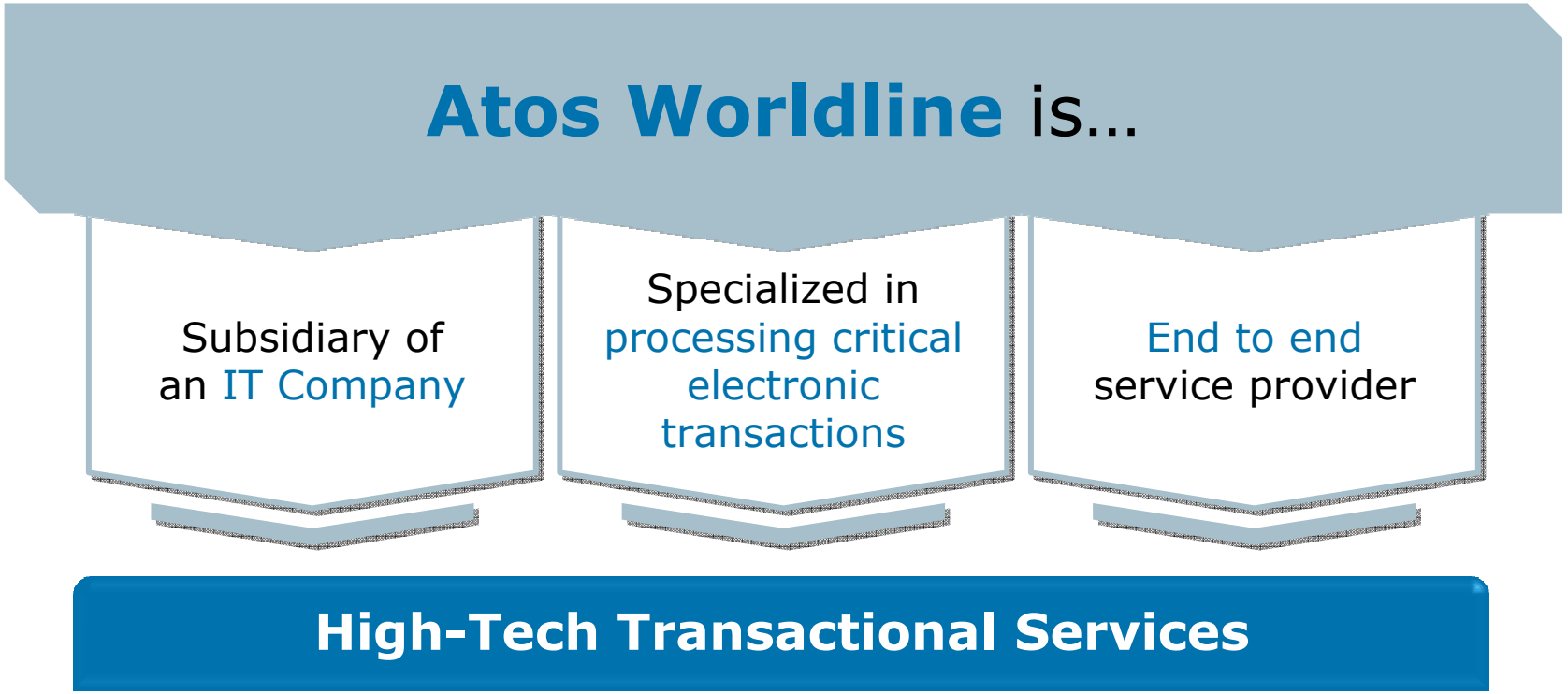
Agenda

13/11/2011
Thomas Knauber

- **Atos Worldline – Firmenpräsentation**
- **PCI – Übersicht**
- **Transparent Data Encryption - Übersicht**
- **TDE – Key-Management**
- **Nachträgliche Einführung von TDE**
- **Betriebliche Aspekte von TDE**
- **Fazit**
- **Q & A**

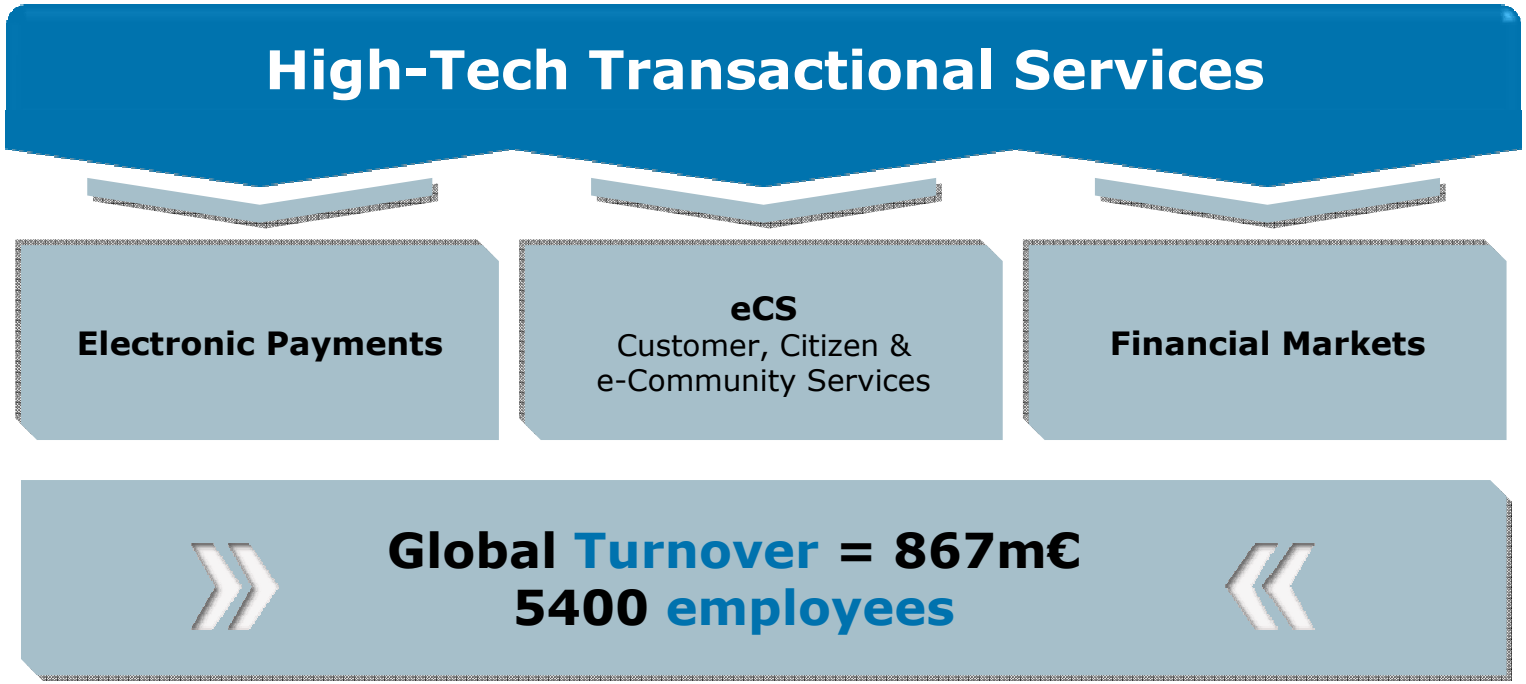
Atos Worldline

13/11/2011
Thomas Knauber



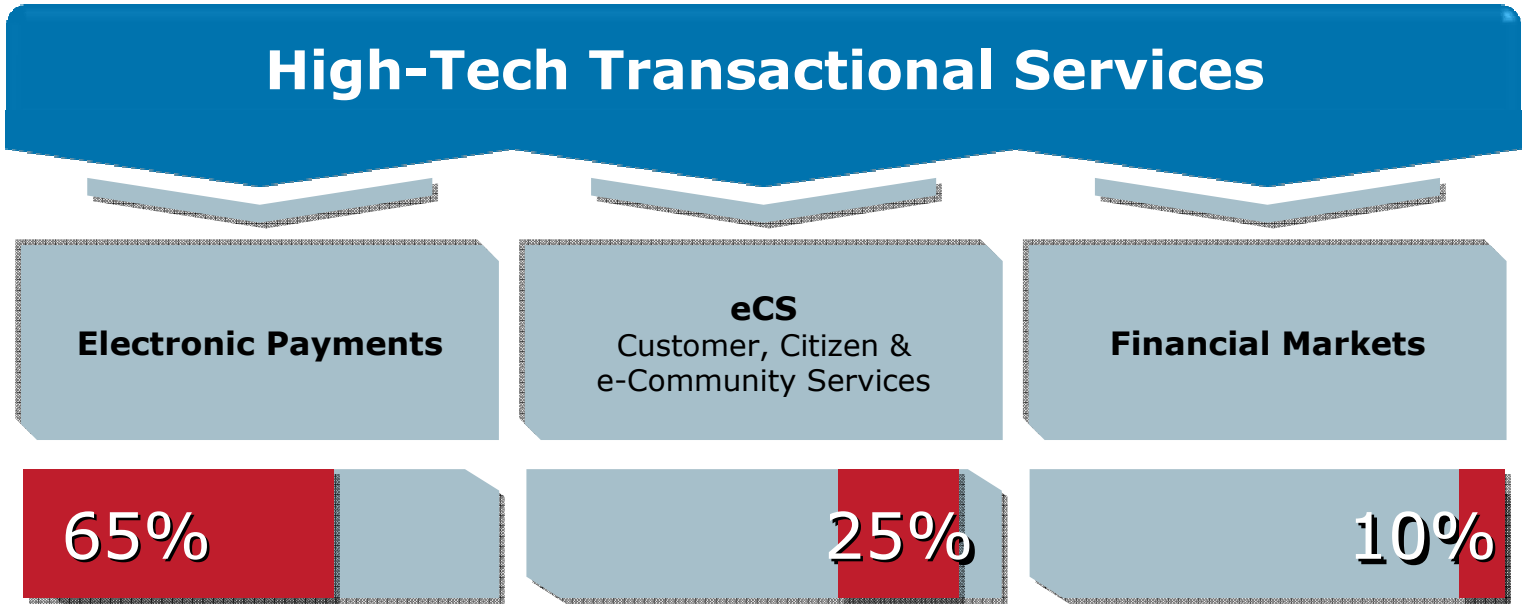
Atos Worldline

13/11/2011
Thomas Knauber



Atos Worldline

13/11/2011
Thomas Knauber



Atos Worldline

13/11/2011
Thomas Knauber

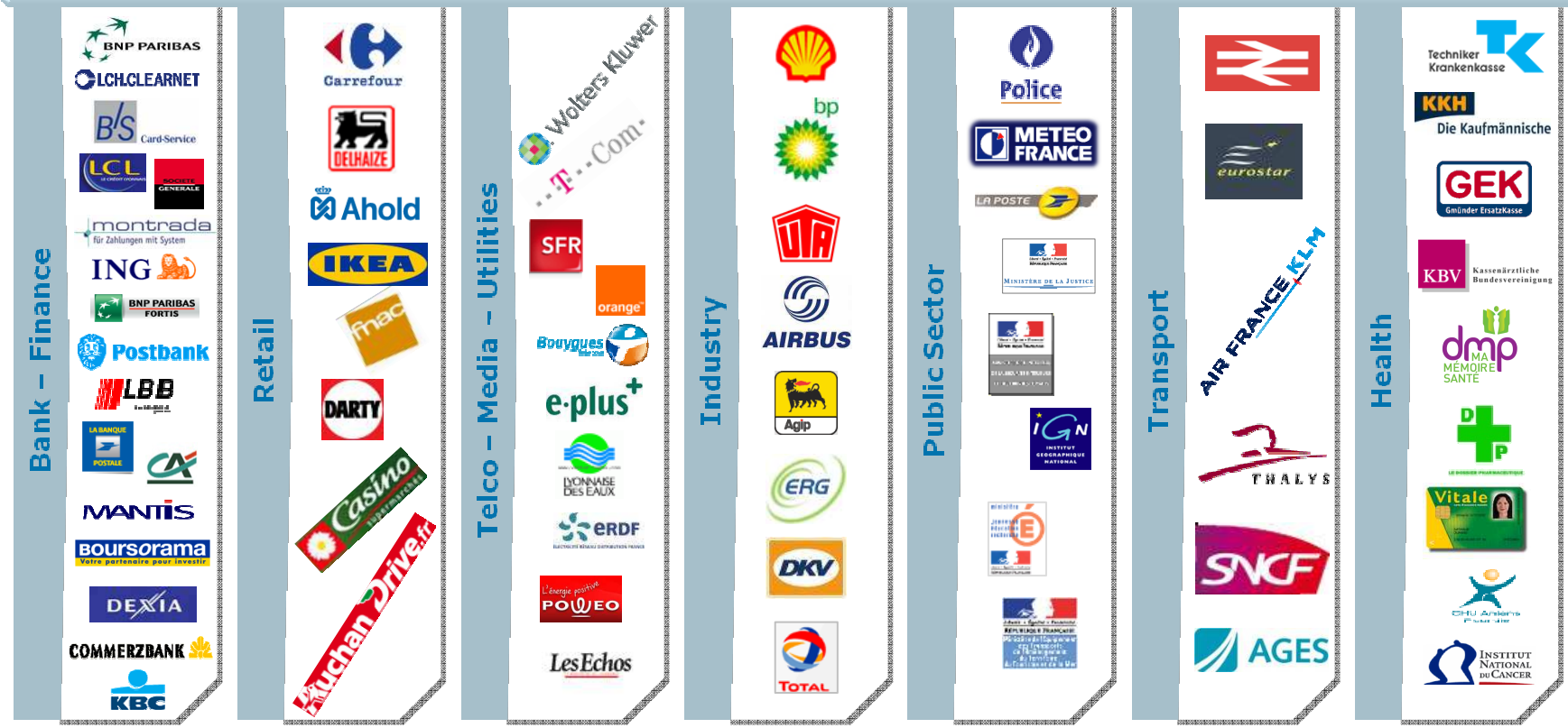
High-Tech Transactional Services

Electronic Payments	eCS Customer, Citizen & e-Community Services	Financial Markets
28 million credit cards and debit cards 415 million remote payment* 2.2 billion acquiring transactions* 570 000 terminals	60 million mailboxes 150 billion e-mails* +1 billion SMS* 38 million loyalty cards 2 billion calls (IVR & Contact Center)* 144 billion internet pages viewed* +1 billion e-documents*	280 million trades € 450 billion assets under management

Atos Worldline

13/11/2011
Thomas Knauber

They trust Atos Worldline to **efficiently manage** their **critical electronic transactions**



Payment Card Industry Data Security Standard PCI DSS

Übersicht

Payment Card Industry Data Security Standard

13/11/2011
Thomas Knauber

- **PCI ist ein Security Standard der Payment Card Industry**
- **PCI ist ein umfassendes Regelwerk um die Sicherheit von Payment Karten zu verbessern.**
- **Ursprünglich gab es 5 verschiedene Standards (z.B. von VISA, MasterCard etc.) die 2004 zu einem einheitlichen Standard zusammengefasst wurden.**
- **Zur Zt. verlangen Visa und MasterCard von allen Service Providern die Kartendaten verarbeiten und speichern, dass sie die PCI Anforderungen erfüllen.**
- **Die Erfüllung der PCI Anforderungen wird in einem jährlichen Audit überprüft.**

PCI DSS Anforderungen

Regulierungsziel	PCI DSS Anforderung
Aufbau und Pflege eines Secure Networks	1. Installation und Pflege einer Firewall
	2. keine Verwendung von Default Passwörtern
Schutz von Kartendaten	3. Schutz der gespeicherten Kartendaten
	4. Verschlüsselte Übetragung in Rechnernetzen
Aufbau eines Gefährdungsmanagements	5. Einsatz und regelmäßiges Update von Virenschutzsoftware
	6. Entwicklung sicherer Systeme und Anwendungen
Implemetierung starker Zugangskontroll-Maßnahmen	7. Zugriff auf Kartendaten nur bei Notwendigkeit
	8. Zuweisung einer eindeutigen ID an jede Person
	9. Beschränkung des physikalischen Zugangs zu Daten
Regelmäßige Netzwerk Tests und Monitoring	10. Protokollieren und Prüfen aller Zugriffe auf Kartendaten.
	11. Regelm. Test der Security Systeme und Prozesse
Aufbau einer Security Policy	12. Einführen und Einhalten von Richtlinien in Bezug auf Informationssicherheit

PCI DSS
Anforderungen an Datenbanken

- **PCI ist ein eher allgemein formuliertes Regelwerk. Konkrete Anweisungen zur Verwendung bestimmter technischer Lösung wie z.B. von Oracle Features sind nicht enthalten.**

- **Interessant ist hier für Datenbanken die Anforderung :**

  “3. Schutz der gespeicherten Kartendaten”

- **Die PCI-Auditoren leiten daraus ab, dass alle gespeicherten Kartendaten davor geschützt werden müssen mit Tools wie z.B. Hex-Editoren ausgelesen zu werden.**

- **Das bedeutet für Oracle Datenbanken, dass dort gespeicherte Kartendaten verschlüsselt gespeichert werden müssen.**

Transparent Data Encryption
TDE

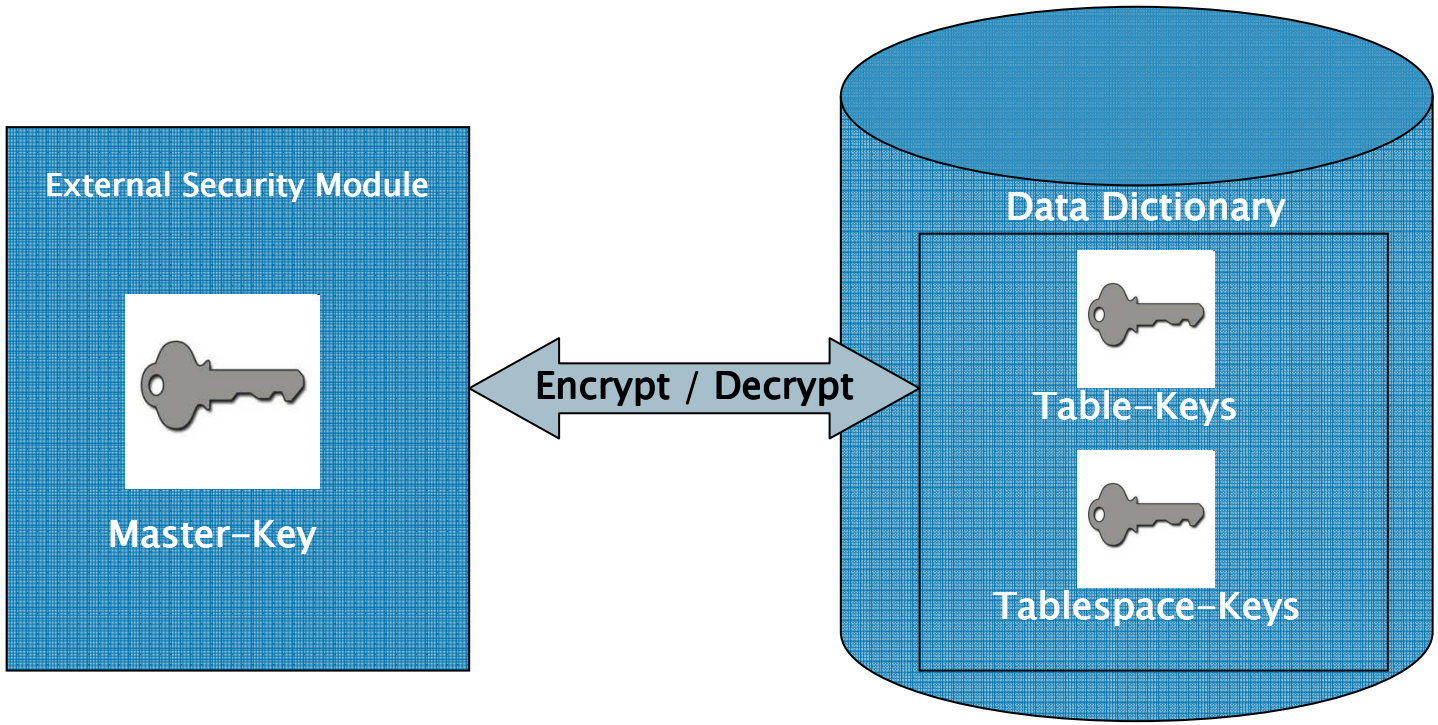
Übersicht

Oracle Transparent Data Encryption TDE

- **Die geforderte Verschlüsselung kann z.B. durch Verschlüsselung in der Anwendung sichergestellt werden. D.h. die Daten werden bereits verschlüsselt in die Tabellen geschrieben und können nur von der Anwendung entschlüsselt werden.**
- **Bei den meisten von Atos selbst entwickelten Lösungen ist dies der Fall.**
- **Bei allen anderen Datenbanken in denen Kartendaten gespeichert werden, muss auf eine externe Lösung zurückgegriffen werden.**
- **Oracle bietet dafür das Feature Transparent Data Encryption – TDE**
- **TDE ist Bestandteil der Advanced Security Option und verschlüsselt Daten beim Schreiben in die Datafiles.**
- **TDE ist *keine* Security-Gesamtlösung, sondern kann nur ein Baustein eines Gesamtkonzepts sein.**

Oracle TDE
Übersicht

- **TDE realisiert Verschlüsselung mittels Keys und Verschlüsselungs-Algorithmen.**
- **Es wird eine 2-Schicht-Architektur bestehend aus einem Master-Key und Tabellen- bzw. Tablespace Keys verwendet.**



Oracle TDE
Übersicht

- **Der Master-Key wird dazu benutzt die Tabellen- bzw. Tablespace-Keys zu verschlüsseln.**
- **Der Master-Key wird in einem sogenannten external Security Module gespeichert.**
- **Dies kann eine sogenannte Wallet oder auch ein Hardware Security Modul (HSM) sein.**
- **Der Master Key wird bei der initialen Einrichtung von TDE generiert.**
- **Es können einzelne Columns oder ganze Tablespaces mit TDE verschlüsselt werden.**

TDE

Key Management

Oracle TDE
Key Speicherung

- **Bzgl. des Master-Keys hat es einige Veränderungen von Oracle 10.2 über 11.1 bis zu 11.2 gegeben.**
- **In 10.2 gibt es nur den Master Key für die Column Encryption**
- **Seit 11.1 gibt es auch Tablespace Encryption. Bei der initialen Einrichtung wird dafür ein *separater* Master Key in der Wallet gespeichert.**
- **Mit 11.2 wird nun ein Unified Master Key eingeführt.**
- **Bei einer Re-Key Operation (Erzeugung eines neuen Schlüssels) werden evtl. vorhandene separate Schlüssel für Tablespace und Columns zusammengeführt.**
- **Beim initialen Aufsetzen von TDE unter 11.2 wird direkt ein Unified Master-Key erzeugt.**

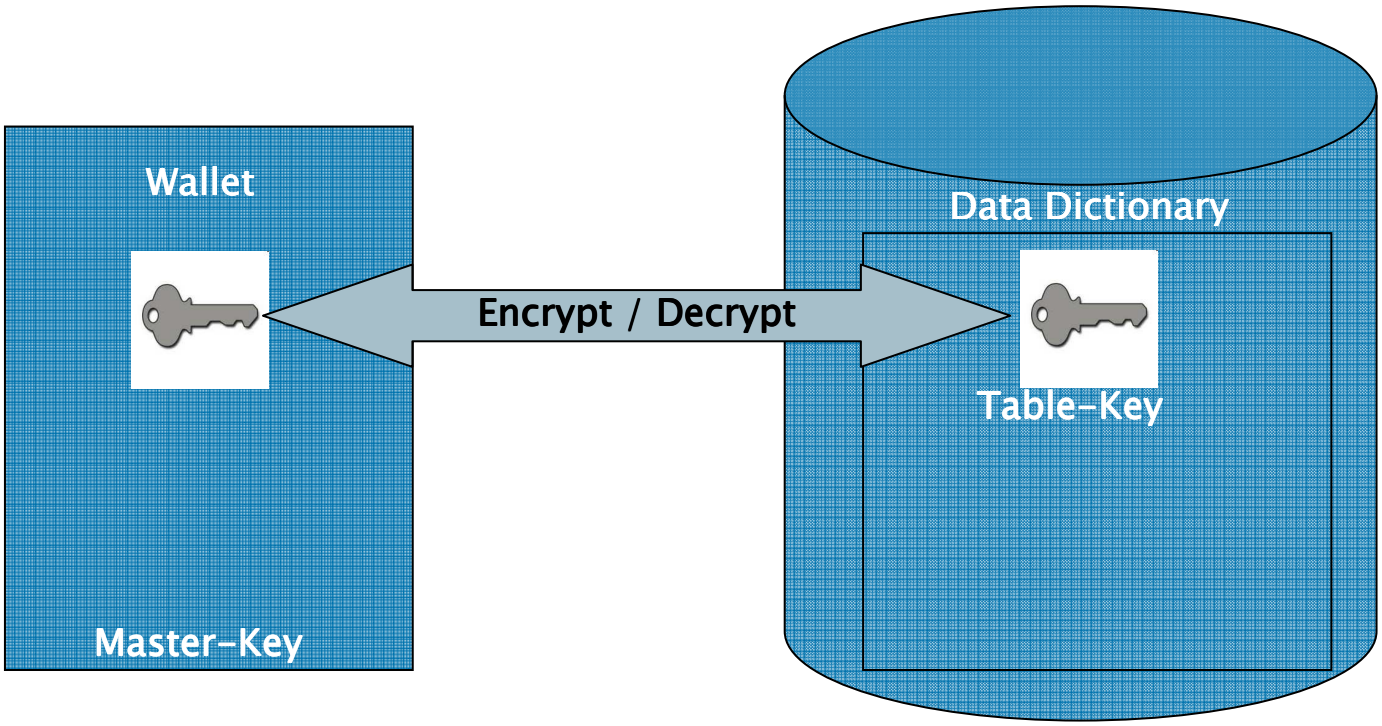
Oracle TDE
Key Speicherung

- **Der Master-Key ist nicht mit dem Wallet-Passwort zu verwechseln.**

- **Der Masterkey wird automatisch generiert, das Wallet Passwort schützt nur den Zugriff auf die Wallet.**

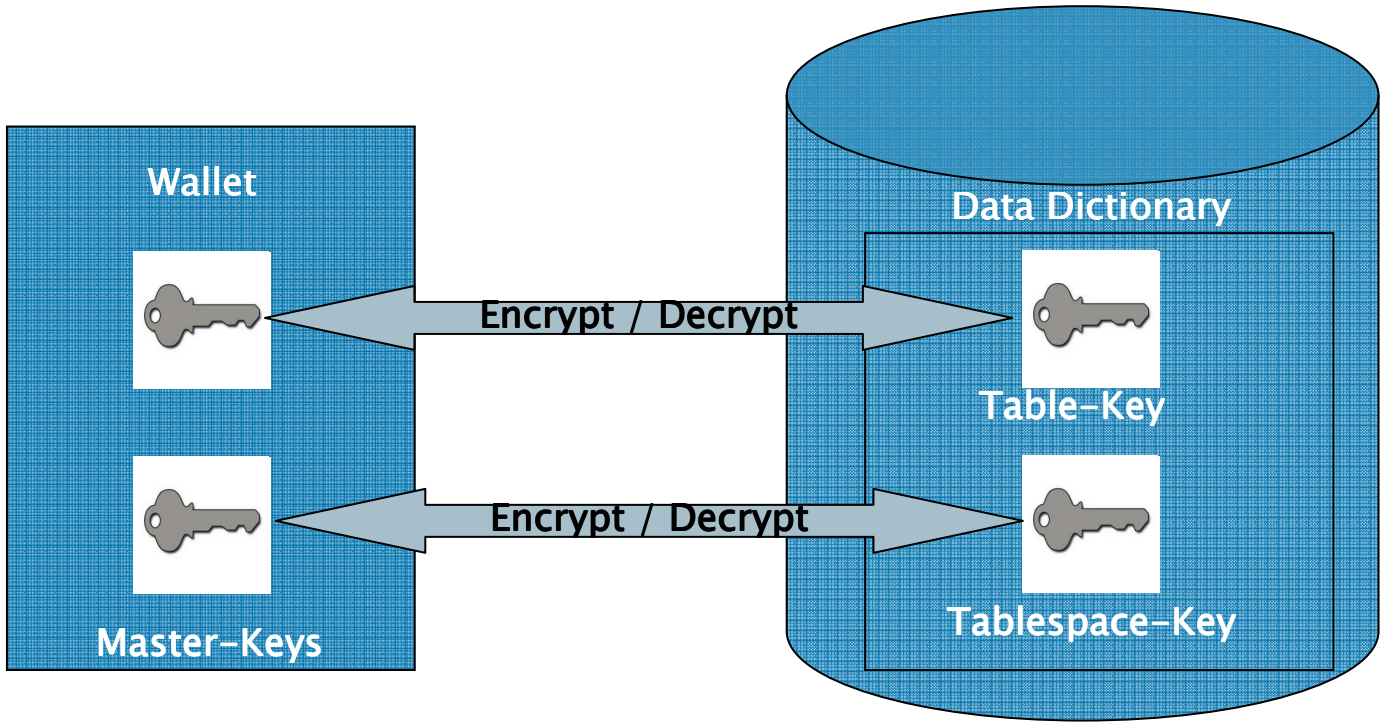
Oracle TDE
Key Speicherung

Oracle 10gR2



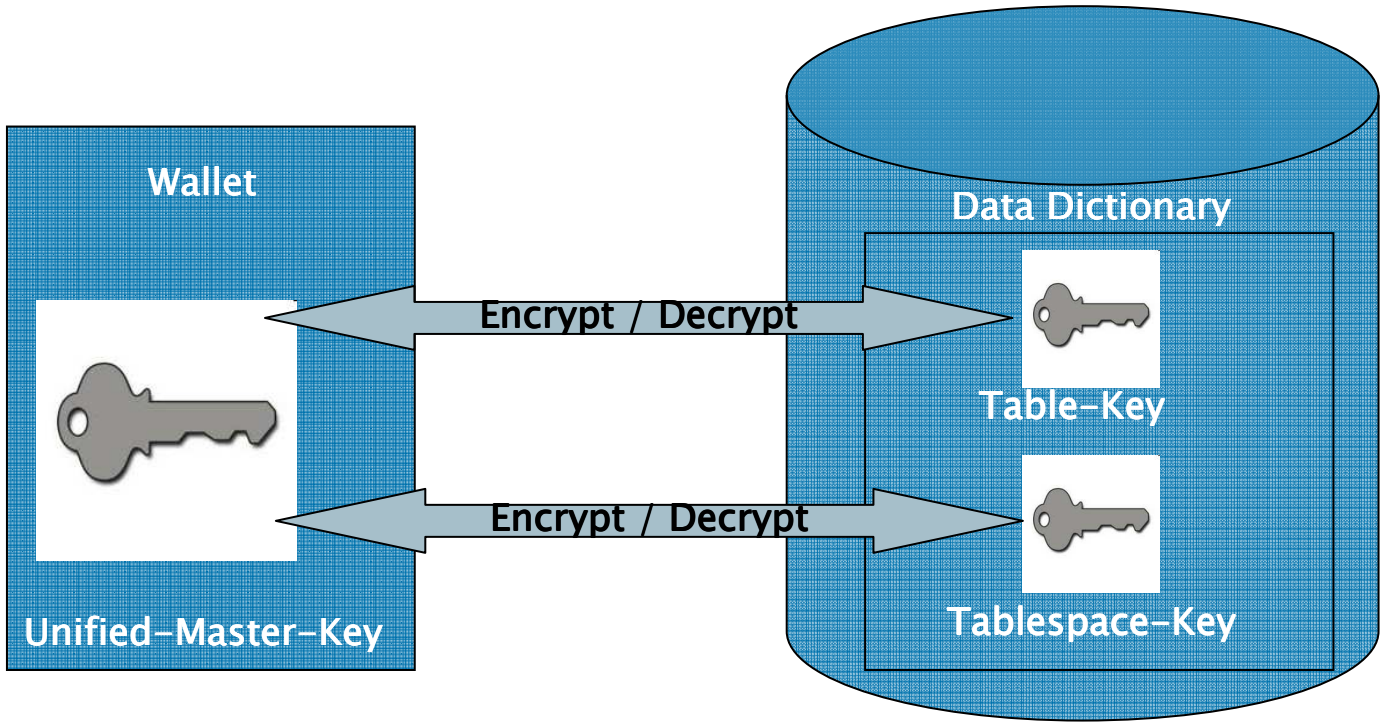
Oracle TDE
Key Speicherung

Oracle 11gR1



Oracle TDE
Key Speicherung

Oracle 11gR2



Oracle TDE
Re-Key-Operationen

- **Unter einer Re-Key-Operation versteht man die Erzeugung eines neuen Schlüssels.**
- **PCI schreibt inzwischen eine jährliche Re-Key-Operation für die Master-Keys vor.**
- **Das Erzeugen eines neuen Master-Keys ist unproblematisch, da er nur der Verschlüsselung der anderen Keys dient.**
- **Ein Re-Key der Table Keys ist zwar seit 10.2 möglich, aber es müsste jeder Datensatz in der verschlüsselten Tabelle verändert werden.**
- **Ein Re-Key der Tablespace Keys ist bis jetzt nicht möglich, aber auch von PCI nicht gefordert. Eine solche Anforderung würde die Verwendung von Tablespace Encryption derzeit ausschließen.**

Oracle TDE
Re-Key-Operationen

Re-Key Optionen				
	Column Encryption		Tablespace Encryption	
	Master Key	**Table Keys**	**Master Key**	**Tablespace Keys**
Oracle 10gR2	Ja	Ja	n/a	n/a
Oracle 11gR1	Ja	Ja	Nein	Nein
Oracle 11gR2	Ja	Ja	Ja	Nein

Unified Master Key

Oracle TDE
Column Encryption

- **Column Encryption verschlüsselt eine oder mehrere Spalten einer Tabelle.**

- **Erzeugung mit :**

```
CREATE TABLE table_name ( column_name column_type ENCRYPT,....);
ALTER TABLE table_name MODIFY ( column_name column_type ENCRYPT,...);
```

- **Alter Table... verschlüsselt nur die Daten die danach eingefügt werden.**

- **Man muss selbst wissen welche Tabellen und Spalten sensible Daten enthalten.**

- **Mögliche Verschlüsselungs-Algorithmen sind :**
  - **3DES168**
  - **AES128 (Default)**
  - **AES192**
  - **AES256**

Oracle TDE
Tablespace Encryption

- **Tablespace Encryption verschlüsselt alle Daten in einem Tablespace.**
- **Erzeugung mit :**

```
CREATE TABLESPACE securespace DATAFILE '/home/user/oradata/secure01.dbf' SIZE
150M ENCRYPTION USING '3DES168' DEFAULT STORAGE(ENCRYPT);
```

- **Existierende Tablespaces können nicht nachträglich verschlüsselt werden.**
- **Mögliche Verschlüsselungs-Algorithmen sind :**
  - **3DES168**
  - **AES128 (Default)**
  - **AES192**
  - **AES256**

Oracle TDE
Welche Option ist die Richtige ?

- **Bei Column Encryption muss man alle zu verschlüsselnden Spalten selbst identifizieren. Das ist z.B. bei kryptischen Column-Namen nicht einfach.**
- **Software-Hersteller können nicht immer alle Spalten benennen, die sensible Daten enthalten.**
- **Verschlüsselte Spalten bringen Limitierungen bzgl. der Verwendung von Indices mit sich :**
  - **Es dürfen nur B-Tree Indices darauf verwendet werden**
  - **Es können nur Spalten indiziert werden, die mit "No Salt" verschlüsselt wurden.**
  - **Range Scans sind nicht möglich ("between Operations")**
  - **Man muss vor der Verschlüsselung wissen mit welchen SQL-Statements auf die Spalten zugegriffen wird.**
- **Eine enge Abstimmung mit dem Software-Development ist bei Column-Encryption unverzichtbar.**

Oracle TDE
Welche Option ist die Richtige ?

- **Bei Tablespace-Verschlüsselung bleiben die Ausführungspläne gleich, da existierende Indices nach der Verschlüsselung weiterhin funktionieren.**
- **Bei beiden TDE-Optionen ist die nachträgliche Einführung in einem produktiven System nicht trivial.**
- **Es können nur neue Tablespaces verschlüsselt werden. D.h. bestehende Objekte müssen mit DataPump, CTAS oder bei partitionierten Tabellen mit** `'alter table ... MOVE partition'` **in den verschlüsselten Tablespace verschoben werden.**
- **Es kann besonders bei großen Tabellen zu erheblich Ausfallzeiten durch die Migration kommen.**
- **Bei produktiven Systemen bietet sich daher die Verwendung von DBMS_REDEFINITION an.**

Einführung von TDE bei bereits produktiven Datenbanken

Oracle TDE DBMS_REDEFINITION

- **DBMS_REDEFINITION ist ein Online Reorganisations-Package für Tabellen.**
- **Es eignet sich gut um vorhandene Tabellen mit minimaler Ausfallzeit nachträglich in die TDE-Verschlüsselung zu migrieren.**
- **Nach unserer Erfahrung funktioniert das stabil und gut, hat aber besonders bei großen Tabellen diverse Fallen.**
- **Das Package arbeitet im Hintergrund mit materialized Views und benötigt genügend Platz , da ja eine Kopie der Tabelle erstellt wird.**
- **Die Materialized View wird im Default Tablespace des Users angelegt -> Vorsicht Platzprobleme bei hoher Last !**
- **Wenn der Redefinition Vorgang abgebrochen werden muss, muss alles sauber mit** `DBMS_REDEFINITION.ABORT_REDEF_TABLE` **beendet werden, da sonst kein Restart möglich ist.**

Oracle TDE DBMS_REDEFINITION

- **Wir raten dringend dazu bei produktiven Datenbanken, besonders bei solchen mit größeren Datenmengen und/oder nur minimal möglicher Downtime den gesamten Redefinition Ablauf vorher auf einem realistischen Testsystem ausführlich zu Testen.**
- **Wenn man den gesamten Ablauf gründlich getestet hat, ist dbms_redefinition sehr gut geeignet die TDE –Einführung zu unterstützen.**
- **Vor solchen Operationen in der Produktion ist ein Backup Pflicht.**

Betriebliche Aspekte von TDE

Oracle TDE
Wallet Management

- **Auf den letzten Seiten ist deutlich geworden, dass der Wallet bei TDE eine zentrale Bedeutung zukommt.**
- **Zunächst stellt sich die Frage wo die Wallet gespeichert werden soll. Die Location der Wallet kann man in der Datei *sqlnet.ora* festlegen.**
- **Die Default Location ist :** `$ORACLE_BASE/admin/$ORACLE_SID/wallet`
- **Hier gab es einen Bug bis 10.2.0.3 . Die Default Location wurde nicht erkannt.**
- **Das Wallet Verzeichnis ist bei uns mit ‘chmod 000’ abgesichert. Der User Oracle hat dann über eine ACL entsprechende Zugriffsrechte.**

Oracle TDE
Wallet Management

- **Beispiel sqlnet.ora**

```
ENCRYPTION_WALLET_LOCATION=
             (SOURCE=(METHOD=FILE)(METHOD_DATA=
            (DIRECTORY=/u01/app/oracle/product/10.2.0/network/admin)))
```

- **Leider ist in der sqlnet.ora nur ein Eintrag möglich.**
- **Das führt zu Problemen, wenn mehrere Datenbanken auf dem Server laufen, da eine Wallet nur exclusiv für eine Datenbank gültig ist.**

Oracle TDE
Wallet Management

- **Sehr wichtig ist auch ein Backup der Wallet, denn ohne Wallet kommt man nach einem Crash nicht mehr an die verschlüsselten Daten.**
- **Die Wallet sollte zwar gleichzeitig mit der Datenbank gesichert werden, aber nie auf dem gleichen Sicherungsmedium gespeichert werden.**
- **Vor jeder Änderung der Wallet sollte diese auf jeden Fall gesichert werden.**
- **Laut Aussage des Oracle Supports gibt es keinerlei 'Hintertür' um ohne Wallet wieder an die verschlüsselten Daten zu kommen !**
- **Die Änderung des Wallet Passworts musste vor Version 11.1.0.7 über den Wallet Manager erfolgen. Bei höheren Versionen kann man das 'orapki' Utility verwenden.**

Oracle TDE
Betriebliche Aspekte

- **Man hat bei TDE die Möglichkeit ein sogenanntes Auto-Open-Wallet anzulegen, d.h. die Wallet wird bei einem Start der Datenbank automatisch geöffnet.**
- **Seit 11.1.0.7 gibt es die local Auto-Open-Wallet. Diese funktioniert nur auf dem Host auf dem sie angelegt wurde.**
- **Leider habe die PCI-Auditoren die Auto-Open-Wallet nicht zugelassen..**
- **Wir sind deshalb gezwungen ein Vier-Augen-Prinzip zu verwenden.**
  D.h. der DBA kennt das Wallet Passwort nicht und der Inhaber des Passwortes hat kein 'alter system' Recht. Nur beide zusammen können die Wallet öffen.

Oracle TDE
Betriebliche Aspekte

- **Dieses Vier-Augen-Prinzip führt aber leider zu Problemen bei einem Crash-Recovery.**
- **Hat man Tablespace-Encryption im Einsatz kommt es beim Recovery zu einer Fehlermeldung da verschlüsselte Tablespaces nicht recovert werden können bevor die Wallet geöffnet ist. Die Datenbank bleibt nach dem Start im Mount-Status.**
- **Aber wie kann man die Wallet mit dem Vier-Augen-Prinzip öffnen, wenn die Datenbank nicht geöffnet ist ? (Grant ist nicht möglich, wenn die DB nicht geöffnet ist)**
- **Dieser Sachverhalt macht den Einsatz von automatischen Cluster-Schwenks im PCI-Umfeld unmöglich da immer manuell eingegriffen werden muss.**

Oracle TDE
Betriebliche Aspekte


```
ALTER DATABASE OPEN
Beginning crash recovery of 1 threads
 parallel recovery started with 7 processes
Started redo scan
Completed redo scan
 6686 redo blocks read, 788 data blocks need recovery
Started redo application at
 Thread 1: logseq 580954, block 45311
Recovery of Online Redo Log: Thread 1 Group 3 Seq 580954 Reading mem 0
  Mem# 0: /db151/rlogs/m01/infoserp/infoserp_redo_G03_M01.dbf
  Mem# 1: /db171/rlogs/m02/infoserp/infoserp_redo_G03_M02.dbf
kcrf_decrypt_redokey: wallet is not opened..
Mon Dec 06 07:05:11 2010
kcbztek_get_tbskey: decrypting encrypted key for tablespace 0 without
opening the wallet
Mon Dec 06 07:05:11 2010
```

Oracle TDE
Betriebliche Aspekte


```
Completed: ALTER DATABASE   MOUNT
Mon Dec 06 11:30:41 2010
alter system set wallet open identified by *
Completed Successfully
   alter system set wallet open identified by *
alter database open
Beginning crash recovery of 1 threads
 parallel recovery started with 7 processes
Started redo scan
Completed redo scan
 1366 redo blocks read, 331 data blocks need recovery
Started redo application at
 Thread 1: logseq 580957, block 5816
Recovery of Online Redo Log: Thread 1 Group 1 Seq 580957 Reading
mem 0
```

Oracle TDE
Betriebliche Aspekte

- **Wir haben einen SR gestellt und einen Lösungsvorschlag erhalten :**
  - **Als Vorbereitung wird dem User, der das Wallet-Passwort kennt sysdba gegrantet.**
  - **Der DBA mountet die Datenbank**
  - **Die zweite Person meldet sich als sysdba über sqlnet an und öffnet die Wallet.**
  - **Der DBA öffnet die Datenbank.**
- **Leider kann man sysdba-Rechte nicht granten wenn die DB nicht offen ist.**
  - **Man sichert sich eine Kopie des Passwort-Files in dem der User sysdba-Rechte hat.**
  - **Ist die DB down wird das Passwortfile mit den Sysdba-Rechten restored.**
  - **Die DB wird gemountet und die zweite Person kann mit User-Passwort und Wallet-Passwort die Wallet öffen.**
  - **Die DB wird vom DBA geöffnet und das Sysdba-Recht entzogen.**

Oracle TDE
Betriebliche Aspekte

- **Diese Vorgehensweise funktioniert zwar, ist aber nur bei geplanten Aktionen wirklich praktikabel.**
- **Bei ungeplanten DB-Restarts sind zu viele Einzelschritte zwischen 2 Personen zu koordinieren. Das ist in der Produktion nicht realistisch.**
- **Es bleibt letztlich nur noch eine Option um PCI und die betrieblichen Anforderungen unter einen Hut zu bringen.**

Die Speicherung des Master Keys in einem Hardware Security Module

Oracle TDE
Betriebliche Aspekte

- **Bei Verwendung eines HSM werden die Keys für Column und Tablespace Verschlüsselung zum HSM geschickt und mit dem dort gespeicherten Master-Key ver- bzw. entschlüsselt.**

- **Der Master Key verlässt niemals das HSM.**

- **Die Keys für die Column-Encyption werden alledings nicht gecached.**
  D.h. bei jedem Zugriff müssen die Table-Keys erneut entschlüsselt werden.
  Die Tablespace-Keys werden nur beim Start der DB entschlüsselt und dann in der SGA gecached.

- **Seit 11.2.0.2 ist mit dem Patch 12626642 auch die Auto-Open Funktionalität mit HSM möglich.**

Oracle TDE
Betriebliche Aspekte

Database Version	Master Key für	in Wallet	in HSM
10gR2	Column Encryption	Ja	n/a
11gR1 11.1.0.6	Column Encryption	Ja	Ja
11gR1 11.1.0.6	Tablespace Encryption	Ja	Nein
11gR1 11.1.0.7	Column Encryption	Ja	Ja
11gR1 11.1.0.7	Tablespace Encryption	Ja	Ja (kein Re-Key)
11gR2	Unified Master Key	Ja	Ja

Oracle TDE
Fazit

- **Oracle TDE ist ein sinnvoller Baustein eines Security-Gesamtkonzepts**
- **TDE stellt den DBA vor eine Vielzahl von Entscheidungen bei Einführung und Betrieb, besonders wenn die Randbedingungen durch Compliance-Vorschriften zusätzlich verschärft werden.**
- **Man sollte sich zunächst genau überlegen welche TDE-Option man einsetzt. Wir sind zu dem Schluss gekommen, dass Tablespace-Encryption für uns praktikabler ist.**

  Column-Encryption wird empfohlen, wenn nur sehr wenige Spalten bei wenigen Tabellen verschlüsselt werden sollen (ca. weniger als 5 %) .
- **Wenn man sich für eine Option entschieden hat, gilt es die betrieblichen Bedürfnisse und die Compliance Anforderungen zu vereinen.**

  Dies kann z.B. dazu führen, dass auf eine bestimmte Oracle Version upgegraded werden muss, da einige TDE-Optionen (Re-Key etc.) nur mit bestimmten Versionen möglich sind

Oracle TDE
Fazit

- **Wir haben uns entschieden möglichst zeitnah Oracle 11gR2 mit Tablespace-Verschlüsselung bei allen PCI relevanten Datenanken einzusetzen.**

- **Aus betrieblichen Gründen wechseln wir von einer Wallet-Speicherung der Keys zum Einsatz von HSMs.**

- **Wenn Optionen und Betriebskonzepte festgelegt sind, ist entscheidend wie man bestehende Systeme nach TDE migriert.**
  Hier haben sich die klassischen Methoden (CTAS, alter table move..) bewährt, wenn die Verfügbarkeit und die Datenmengen es erlauben.

- **Bei höheren Verfügbarkeiten hat sich dbms_redefinition durchaus bewährt, aber gründliche Vorbereitungen und Tests sind hier unverzichtbar.**

TDE
Hilfreiche Dokumente

- **Master Note For Transparent Data Encryption (TDE) [ID 1228046.1]**
- **TDE FAQs**
  - http://www.oracle.com/technetwork/database/security/tde-faq-093689.html
- **TDE Oracle White Paper**
  - http://www.oracle.com/technetwork/database/focus-areas/security/twp-transparent-data-encryption-bes-130696.pdf

Oracle TDE
Q & A

13/11/2011

Thomas Knauber

Thank you

13/11/2011