



# TokenControl V2 Implementation

white paper  
for internal / expert use only  
Version 0.12  
april 08, 2002

Alle Rechte vorbehalten. Die Informationen in diesem Dokument sind für das Personal der Secaron und der Firmen gedacht, die in diesem Dokument als Auftraggeber oder Zielgruppe benannt sind. Da sich in diesem Dokument vertrauliche oder geheime Informationen befinden können, darf es ohne schriftliche Genehmigung der secaron AG nicht an andere Personengruppen als die oben genannten kopiert, elektronisch weitergeleitet oder auf irgendeine andere Art weiterverteilt werden.

Falls Sie dieses Dokument fälschlicherweise erhalten haben, kontaktieren Sie bitte umgehend die Fa. secaron AG, Ludwigstr.55, 85399 Hallbergmoos, Germany,  
Telefon: +49 (0)811 9594-100

## 1 Dokument-Daten

### 1.1 Verantwortungen

Verfasser: Bernhard Weber  
Eigentümer: SyTrust  
Freigabe:

### 1.2 Historie

Version 0.1:	06.03.02	B. Weber
Version 0.11:	09.03.02	B. Weber
Version 0.12:	08.04.02	B. Weber

## 2 Integration eines Zertifikats-Speichers

### 2.1 Einleitung

Die Basis für eine sichere elektronische Kommunikation innerhalb einer Firma, bzw. auch zwischen verschiedenen Firmen ist eine Publik-Key-Infrastruktur (PKI). Mit ihr werden Authentizität, Integrität und Vertraulichkeit erst möglich. PKI-Techniken basieren auf digitalen Signaturen, den sogenannten Zertifikaten. Die wiederum können sehr unterschiedlich eingesetzt werden. Je nach Abhängigkeit, welches Keysigning (= „Verwendungszweck“) in dem Zertifikat vorhanden ist, kann mit dem Zertifikat verschlüsselt, signiert oder eine sonstige privilegierte Tätigkeit durchgeführt werden.

Leider gibt es **keinen eindeutigen** Standard für Zertifikate. Derzeit kristallisieren sich folgende „Standards“ heraus: X.509 und PGP.

PGP ist im Gegensatz zu X.509 kein öffentlicher Standard und deshalb auch wesentlich kritischer zu betrachten. Im nachfolgenden werden nur noch Zertifikate auf Basis X.509 betrachtet.

### 2.2 Wie sieht ein Zertifikat aus?

Jedes Zertifikat beinhaltet mindestens einen öffentlichen Schlüssel. Der Eigentümer hat zu dem entsprechenden Zertifikat noch den privaten Schlüssel. Die Schlüssel sind in der Regel mindestens 128 Bit (= 16 Byte) groß. Insgesamt nimmt das Zertifikat jedoch deutlich mehr Platz ein, da neben den Schlüsseln noch viele zusätzlichen Informationen abgespeichert werden müssen.

Als da wären:

- Aussteller des Zertifikates
- Gültigkeit (von, bis)
- Keysigning
- ....

In Summe ist ein einfaches X.509V3-Zertifikat ca. 1 KB groß. Komplexe Zertifikate können ein Vielfaches der Größe annehmen. Aufbewahrt werden X509-Zertifikate entweder in SmartCards, oder in PKCS#12-Dateien.

## 3 Verwendung eines Zertifikates

Meistens genügt es, die vertrauenswürdigen User-Zertifikate in den Zertifikatsstorage des entsprechenden Betriebssystems / Browsers zu laden. Analog für die CA-Zertifikate. Die Anwendung sucht sich in der Regel das passende Zertifikat und verwendet es. Zertifikate stellen aus der Sicht der Geheimhaltung erst mal kein Problem dar. Lediglich der private Schlüssel (für das eigene Zertifikat) muss besonders verwahrt werden. Die Daten in diesem privatem Schlüssel sind vergleichbar mit der eigenen Identität bzw. mit der eigenen Unterschrift. Dieser geheime Schlüssel ist zwar nochmals durch eine PassPhrase geschützt, trotzdem sind hier Angriffe denkbar.

## 4 Warum TokenControl ?

TokenControl stellt - speziell für mittlere und große Unternehmen - die ideale Komponente zum Speichern und Abrufen von Zertifikaten dar.

Für kleine Firmen macht eine PKI nur eingeschränkt Sinn. Die Aufbau- und Unterhaltskosten sind in einer Größenordnung, die sich nur schwer wieder amortisieren. Hier wird es vermutlich einfacher und günstiger sein, für alle Mitarbeiter die Zertifikate durch externe Unternehmen (z.B. VeriSign) versorgen zu lassen.

Anders bei mittleren und großen Unternehmen. Hier schlagen die einzelnen Zertifikate in Summe mit einem erheblichen Betrag zu Buche. Auch wird der logistische/ administrative Aufwand - bei Vergabe an ein externes Unternehmen – unverhältnismäßig hoch. In diesem Fall amortisiert sich eine eigene Lösung relativ schnell.

Wenn innerhalb einer Firma die Entscheidung zugunsten einer internen PKI getroffen wurde (ein externer Dienstleister kommt viel zu teuer), kommen bald konkrete Fragen auf die Tagesordnung:

- „Wo“ wird „Welcher“ Schutzlevel benötigt ?
- Wie sollen Zertifikat beschaffen sein ? (Hardware, Software, beides ?)
- Baut man eine eigene CA auf, oder kauft man sich in einem TrustCenter ein ?
- Wie viele Zertifikate werden pro Mitarbeiter benötigt ?
- Wie technisch versiert sind die Mitarbeiter ?
- Wie wird verfahren, wenn die PassPhrase vergessen wurde ? Bei plötzlichem Tod des Mitarbeiters ? Bei Ausscheiden des Mitarbeiters ?
- Wie wird verfahren bei Verlust, oder Beschädigung einer SmartCard ? Wie lange dauert es, bis eine neue SmartCard zur Verfügung steht ? (Kann mein Mitarbeiter in der Zeit weiterarbeiten ?)
- Was kostet das „Ausrollen“ der Zertifikate / evtl. Leser ?
- Wie lange dauert das Ausrollen ?

Schnell kommt man zu folgenden Schlussfolgerung:

- In großen Betrieben kommt ein externer Dienstleister (z.B. TrustCenter) sehr teuer.
- In den seltensten Fällen sind SmartCards für die Mitarbeiter nötig. (SoftToken reichen aus)
- Die Token (egal ob SmartCard, oder SoftToken) müssen auf einfache Weise vom Mitarbeiter verwendet werden können.
- Die Token (egal ob SmartCard, oder SoftToken) müssen auf einfache Weise durch die Firma gehandhabt werden können.

TokenControl wurde geschaffen, um genau für die o.a. Probleme eine Lösung bereitzustellen.

Die wichtigsten Features auf einem Blick:

- Token-Administration with Policies
- SmartCard-Replacement
- SmartCard-Backup
- Token-History
- Supported storage-types: File, DB, LDAP
- Encrypted Storage

## 5 Einsatzszenarien

Jede Firma hat unterschiedliche Anforderungen an Skalierbarkeit, Hochverfügbarkeit und Hardware. Um dieser Situation gerecht zu werden, wurden exemplarisch verschiedene Einsatzszenarios aufgezeigt.

### 5.1 Szenario: Testversion

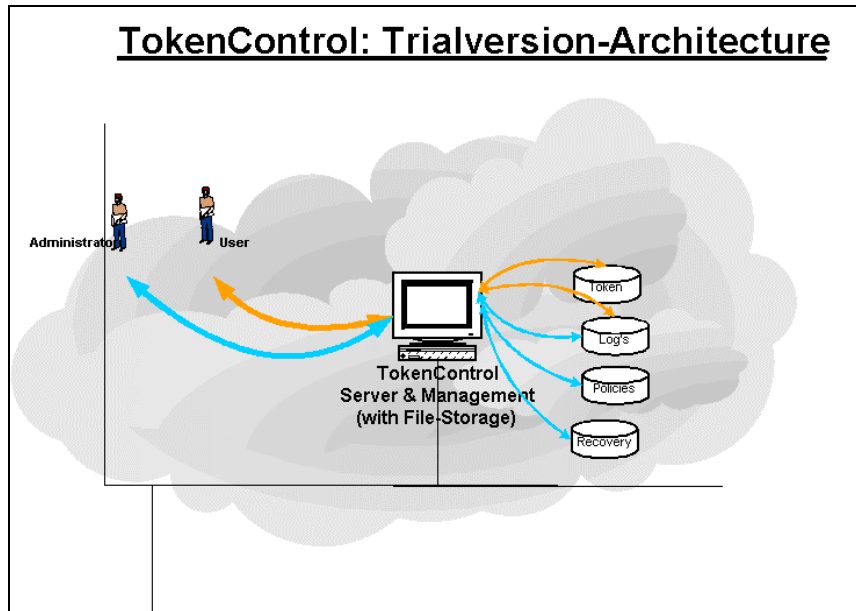


Abbildung 1 TC-Trial-Architecture

In der Trial-Version kann der komplette TokenControl auf einer Maschine (SUN Solaris, PC) installiert und in Betrieb genommen werden. Es wäre auch noch denkbar, auf diesem Rechner auch noch eine Datenbank (mySQL, Oracle ) zu installieren, jedoch sollte dabei bedacht werden, dass Datenbanken i.d.R. sehr viele Ressourcen verbrauchen. Außerdem kommt es bei der Trial-Version mehr auf eine schnelle und unkomplizierte Inbetriebnahme an, was eindeutig dafür spricht, die Datenhaltung im Filesystem vorzunehmen.

Auch die Policies werden bei der Trial-Version so eingestellt, dass alle Benutzergruppen (Benutzer gleichermaßen wie Administratoren) diesen einen Rechner verwenden können. Solange sich die Datenmengen in Grenzen halten (<1000), sollte diese Variante der Datenhaltung (Filesystem) nur unwesentlich langsamer sein.

## 5.2 Szenario: "Small" Business Solution

Für mittelständische Betriebe, die nicht sehr viele Mitarbeiter (bis ca. 1.000 Bildschirm-arbeitsplätze) – jedoch ein gesteigertes Bedürfnis an Sicherheit – haben, kann TokenControl bei geringem Kostenaufwand, dennoch fast professionellen Ansprüchen genügen:

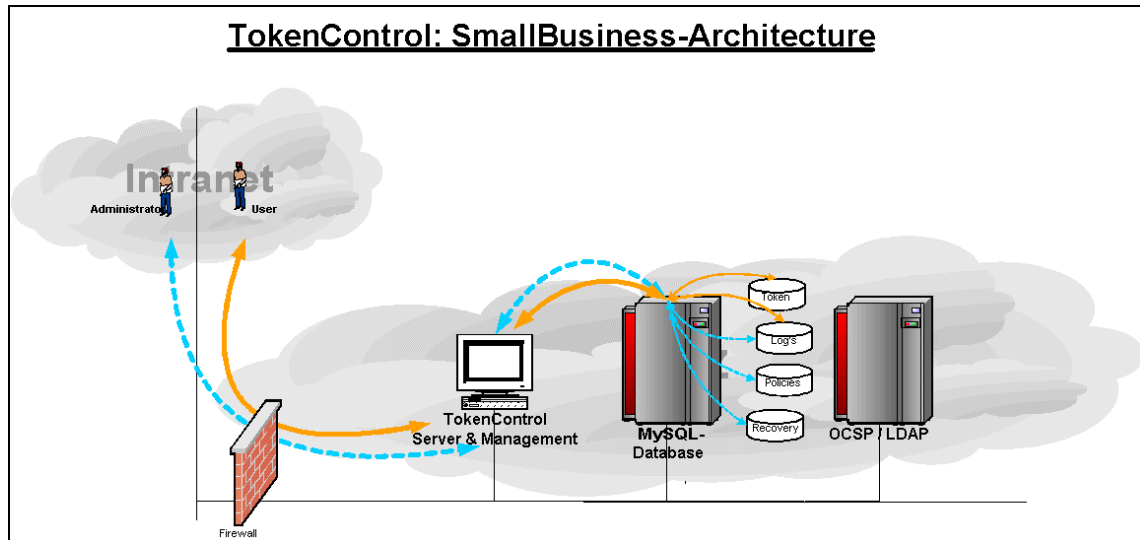


Abbildung 2 TC-SBS-Architecture

I.d.R. haben Betriebe in dieser Größenordnung eine Firewall mit einer kleinen DMZ für die nötigsten Dienste / Anwendungen.

Neben dem Intranetserver und div. Datenbankanwendungen könnte in dieser DMZ auch ein OCSP-Responder mit LDAP-Verzeichnis und TokenControl stehen.

Um die Kosten nicht unnötig in die Höhe zu treiben (Nicht alle Firmen haben den „Zwang“, eine Oracle-DB zu betreiben), kann in dieser Situation ein MySQL-Server verwendet werden. Administrator und Anwender benutzen verschiedene Ports und können so durch die Firewall ordentlich „getrennt“ werden.

Ausfallsicherheit: Auch wenn das Budget kein ColdStandby-Gerät für TokenControl vorsieht, kann man Ausfallzeiten sehr gering halten, indem man das Installationspaket so „konfiguriert“, dass die Installation ohne jegliche Useraktion durchläuft (vom einlegen und wechseln der CD mal abgesehen) – die Konfiguration über die Datenbank macht dies möglich!

### 5.3 Business Solutions

Eine Nummer größer zeigt sich das folgende Szenario. Es ist gedacht für größere Firmen mit ca. 1.000 – 10.000 Bildschirmarbeitsplätzen.

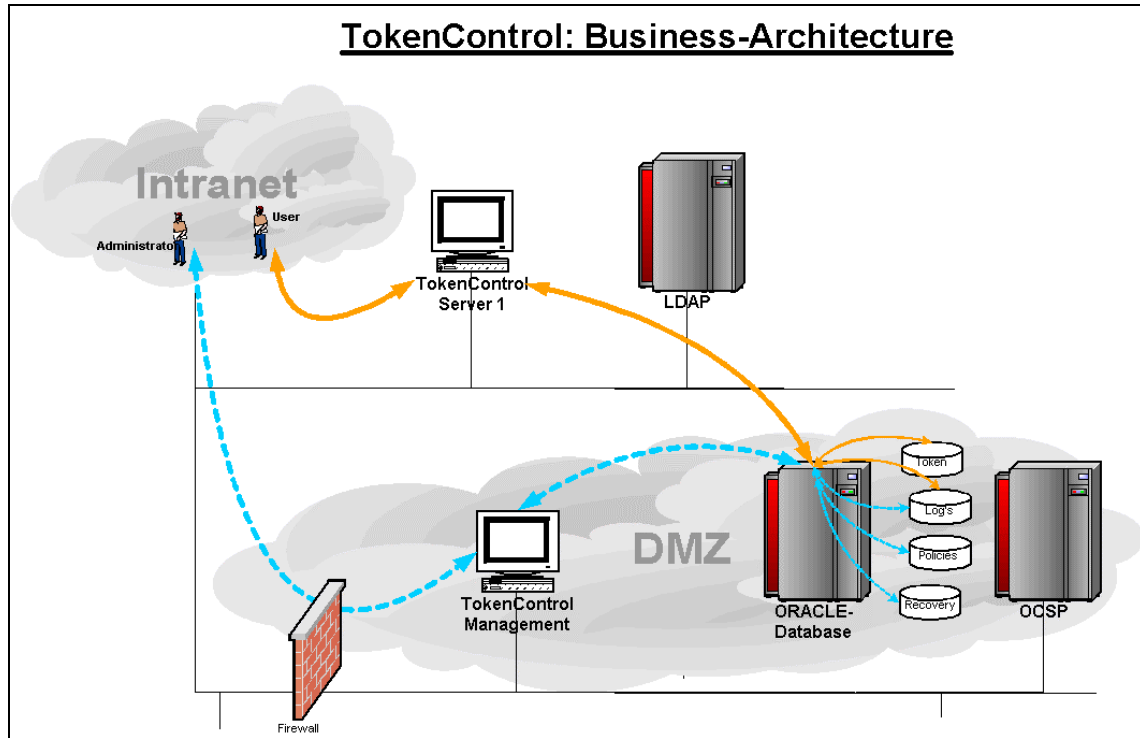


Abbildung 3 Business-Architecture

Der TokenControl wird auf zwei Maschinen aufgeteilt. Dies erfolgt überwiegend nicht aus Performance-Gründen, sondern aus der konsequenten Haltung, dass ein User keine Möglichkeit haben sollte, eine „Administrator-Maschine“ zu kontaktieren. Außerdem wird so sichergestellt, dass z.B. ein Bulk-Load (Lade 10000 Zertifikate ins TokenControl) auf die Administrator-Maschine, den Regelbetrieb nicht stört (weil entkoppelt). Als Datenbank wird in diesem Modell ORACLE verwendet.

Möglichst geringe Ausfallzeiten können in diesem Modell durch HotStandby- / oder ColdStandby-Komponenten realisiert werden, die – zumindest was TokenControl betrifft – keine zusätzlichen Lizenzgebühren / Wartungsgebühren zur Folge haben. Da sämtliche Daten (Token, Konfigurationen, etc.) alle auf der Datenbank liegen, kann ein „Ersatz-TokenControl“ mit **sehr geringem Aufwand** in Betrieb genommen werden.

## 5.4 Enterprise Solutions

Für Großbetriebe / Konzerne sind Hochverfügbarkeit und Skalierbarkeit sehr wichtige Eigenschaften. Die Leistung muss durch Zukauf weiterer Komponenten im Idealfall linear anwachsen. Hochverfügbar heißt in diesem fall weder ColdStandby noch HotStandby, sondern vielmehr eine doppelte Auslegung aller Komponenten, die alle permanent in Betrieb sind. Im Fehlerfall wird dann lediglich die Performance abfallen, jedoch ein lückenloser Betrieb über das verbleibende Gerät möglich sein.

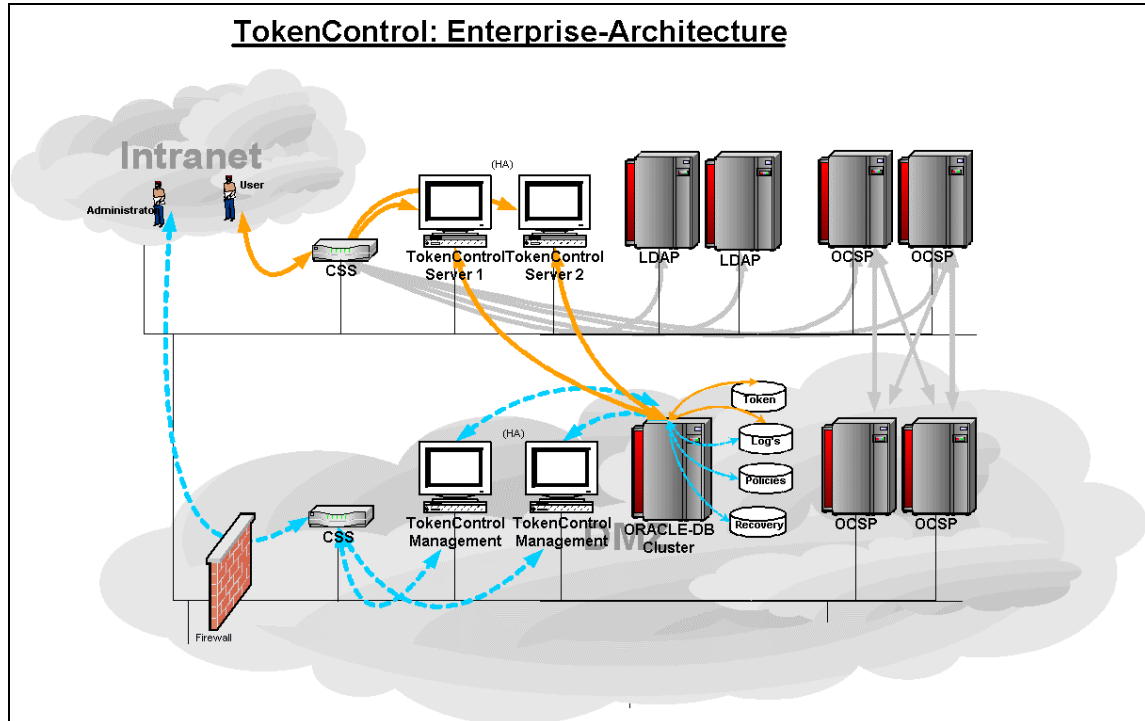


Abbildung 4 Enterprise Architecture

Alle unkritischen Komponenten stehen hierbei im direkten Zugriff des internen Netzes. Der „normale user“ baut eine Verbindung zum CSS auf. Der fungiert als Lastverteiler (natürlich auch doppelt ausgelegt) und baut wiederum eine Verbindung zu einem der TokenControl-Server auf. Der Request wird dort angenommen. Zur Abarbeitung muss der TC-Server eine Verbindung – durch die Firewall hindurch – zur Datenbank aufnehmen. Anders beim Administrator: Der geht direkt durch die Firewall auf die doppelt ausgelegte TokenControl-Management-Station.

Das Einsatzszenario für die OCSP-Responder sei nur der Vollständigkeit halber erwähnt.

## 6 Implementierung

Bei der Entwicklung von TokenControl wurden höchste Ansprüche an die

- Modularität
- Stabilität und
- Sicherheit

gelegt.

### 6.1 Systemstruktur

Konsequente Modularität macht das Weiterentwickeln einfacher, da durch das Modul-Konzept die Aufsetzpunkte für neue Features definiert sind. Gleichfalls können auch die „individuellen Wünsche“ des Kunden i.d.R. wesentlich einfacher realisiert werden.

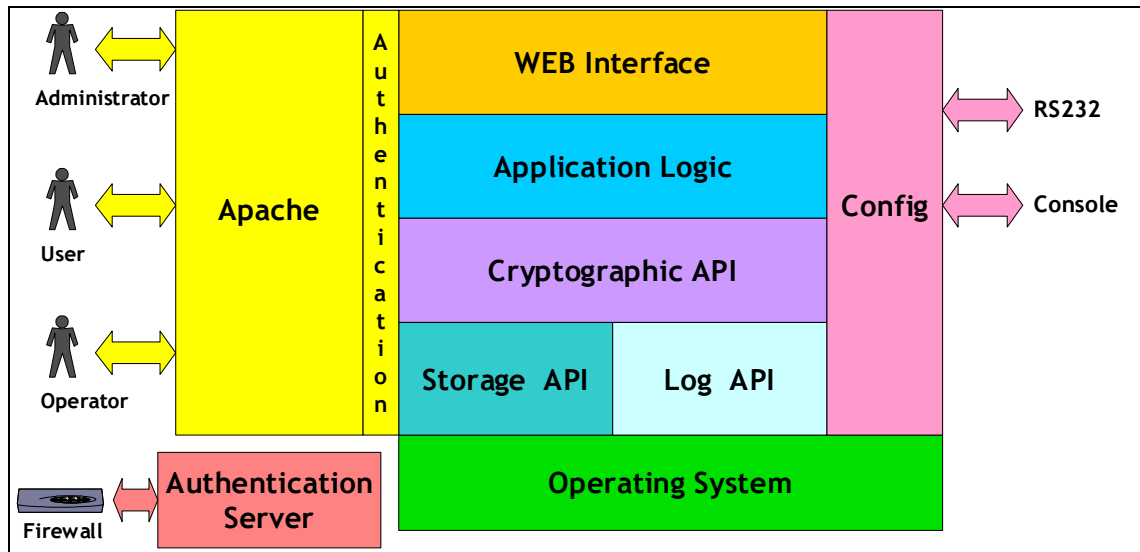


Abbildung 5 Module in TokenControl

Das Modulkonzept setzt sich auch in der Implementierung fort. Sowohl die Installation, als auch die Konfiguration ist höchst modular ausgelegt. Dies geht so weit, dass z.B. die komplette Konfiguration von einer CD gelesen werden kann (unattended installation). Wird dies nicht in diesem Umfang gewünscht, können beliebige Konfigurationsparameter ausgenommen - und „on the fly“ vom Administrator eingefordert werden.

Auch bei der Wahl des „Storages“ macht sich die Modularität bemerkbar. So stehen bei TokenControl z.Zt. 3 verschiedene Storage-Varianten zur Verfügung. Neben ORACLE und mySQL können alle Daten auch in ein FileSystem geschrieben werden. Auch ein LDAP-Modul, kann in kürze verwendet werden.

Natürlich können alle Storage-Varianten mit/ohne Verschlüsselung verwendet werden und bieten zudem Transaktionssicherheit (!), obwohl dies bei mySQL und dem FileSystem eigentlich nicht unterstützt wird.

Je nach Komplexität des Umfeldes kann es durchaus Sinn machen, verschiedene Storages zur selben Zeit zu verwenden. So ist es durchaus denkbar, dass eine ORACLE-Datenbank für die

Speicherung der Zertifikate zum Tragen kommt, während die Konfigurationen in einem Filesystem niedergeschrieben werden sollen. Die Flexibilität ist bei TokenControl so groß, dass Sie unterschiedlichste Storage-Varianten miteinander kombinieren können.

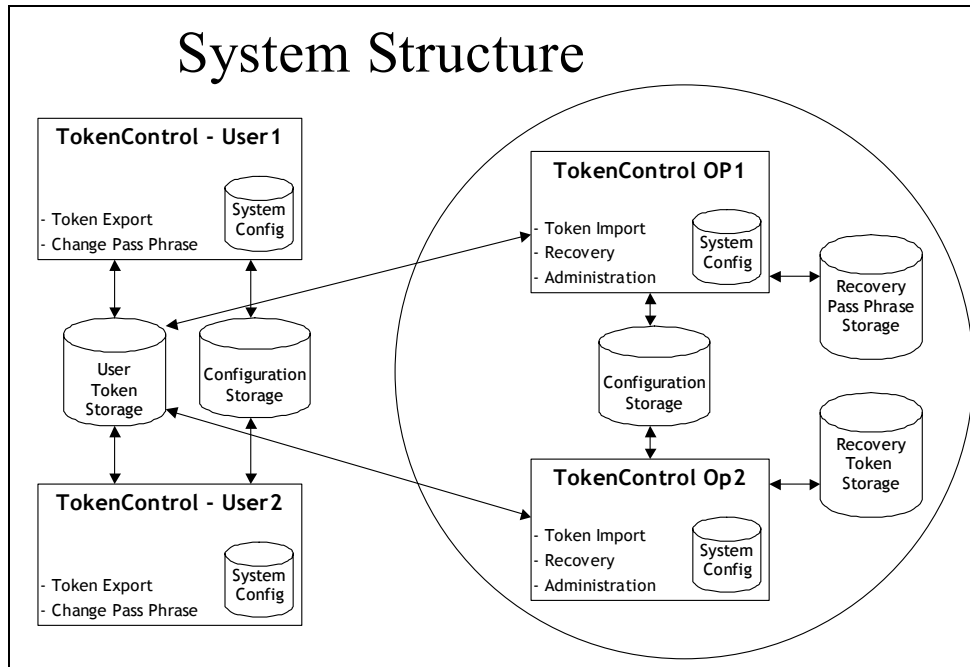


Abbildung 6 System-Structure (storages)

Sind die in Frage kommenden Storages ausgemacht, werden sie auf der Konfigurations-CD den einzelnen Datenbereichen zugewiesen. Ganz egal, wie viele weitere (lastteilende) TokenControls pro Firma eingesetzt werden, muss die Konfiguration des entsprechenden Storages nur einmal geschehen. Alle anderen TokenControls benötigen lediglich eine URL zu dem entsprechenden Storage.

Für folgende Datenbereiche muss ein Storage definiert werden:

- Token
- Recovery-Token
- Recovery-Passphrases
- Configuration
- TrustedStorages
- Logs

## 6.2 Policies

TokenControl unterliegt einem durchgängigen Policy-Konzept. Für sämtliche nachfolgende Aktionen müssen die Policies detailliert eingestellt werden:

- Group Management
- User Management
- Token Import Policy
- Token Export Policy
- Key Recovery Policy

- Authentication Policy
- Protocol Policy
- Access Control Policy

Bitte beachten Sie dabei, dass die Policies immer nur für Gruppen Anwendung finden können. d.h. zuerst müssen die User einer Gruppe zugewiesen werden. Als zweiter Schritt werden dann für diese Gruppe die einzelnen Policies eingestellt.