

Design und Implementierung einer Certification Authority als Enterprise Application für FlexiTRUST

15. Oktober 2002

Markus Winkler & Lutz Feldgen

Diplomarbeit an der
Technischen Universität Darmstadt
Fachgebiet Theoretische Informatik
Kryptographie und Computeralgebra
Prof. Dr. Johannes Buchmann
Alexander Wiesmaier

Markus Winkler, Ludwigsplatz 8 a, 64283 Darmstadt, markus.winkler@gmxpro.de

Lutz Feldgen, Ludwigsplatz 8 a, 64283 Darmstadt, lfeldgen@gmx.de

Inhaltsverzeichnis

1	Einleitung	6
1.1	Über dieses Handbuch	6
1.2	Struktur dieses Handbuches	6
2	Die FlexiTRUST-CA	7
2.1	Systemüberblick	7
2.2	Zielumgebung	7
2.2.1	Hardware	7
2.2.2	Software	7
2.3	Verwendete Werkzeuge	7
3	Spezifikation der Anwendungsfälle und Klassen	8
3.1	Spezifikation der Anwendungsfälle JWS	8
3.1.1	Anwendungsfälle der Domain Bench	8
3.1.2	Anwendungsfälle der Domain Stock	8
3.1.3	Anwendungsfälle der Domain Worker	8
3.2	Spezifikation der Anwendungsfälle CA	8
3.2.1	Anwendungsfälle der Domain Bench	8
3.2.2	Anwendungsfälle der Domain Configuration	8
3.2.3	Anwendungsfälle der Domain Entrance	8
3.2.4	Anwendungsfälle der Domain Exit	8
3.2.5	Anwendungsfälle der Domain Request	8
3.2.6	Anwendungsfälle der Domain Stock	8

<i>INHALTSVERZEICHNIS</i>	3
4 Systemarchitektur	9
4.1 Struktur Java 2 Enterprise Edition	9
4.2 Struktur FlexiTRUST-JWS	9
4.2.1 Architektur EJB-Tier Bench	9
4.2.2 Architektur EJB-Tier Stock	9
4.2.3 Architektur EJB-Tier Worker	9
4.3 Struktur FlexiTRUST-CA	9
4.3.1 Architektur EJB-Tier Bench	9
4.3.2 Architektur EJB-Tier Configuration	9
4.3.3 Architektur EJB-Tier Entrance	9
4.3.4 Architektur EJB-Tier Exit	9
4.3.5 Architektur EJB-Tier Request	9
4.3.6 Architektur EJB-Tier Stock	9
5 Umsetzung der fachlichen Anforderungen	10
5.1 Clustering	10
5.2 Farming	10
5.3 Error-Logging	10
5.4 Schnittstellen zu externen Systemen	10
5.4.1 Dateischnittstelle	10
5.4.2 Hardwareschnittstelle	10
5.5 Konfiguration und Management	10
5.6 Datenbank	10
5.6.1 Modell	10
6 Beschreibung der Pakete	17
7 Installation des Systems	18
7.1 Systemarchitektur	18
7.2 Datenbank Server	18
7.2.1 Systemvoraussetzungen	18

<i>INHALTSVERZEICHNIS</i>	4
7.2.2 Vorbereitende Schritte	19
7.2.3 Installation der Anwendung	19
7.2.4 Update der Anwendung	19
7.2.5 Deinstallation	19
7.3 Applikation Server	19
7.3.1 Systemvoraussetzungen	19
7.3.2 Vorbereitende Schritte	20
7.3.3 Installation der Anwendung	21
7.3.4 Update der Anwendung	23
7.3.5 Deinstallation	23
7.3.6 Verzeichnisstruktur	23
7.4 Pkcs7Server	24
7.4.1 Systemvoraussetzungen	24
7.4.2 Vorbereitende Schritte	24
7.4.3 Installation der Anwendung	24
7.4.4 Update der Anwendung	25
7.4.5 Deinstallation	25
8 Betrieb des Systems	26
8.1 System überwachen	26
8.1.1 Datenbankserver	26
8.1.2 Applikationsserver	26
8.1.3 Pkcs7Server	26

Abbildungsverzeichnis

Kapitel 1

Einleitung

MW

1.1 Über dieses Handbuch

Das vorliegende Dokument beschreibt auf DV-technischer Ebene den Ansatz für die Umsetzung der Funktionalitäten der FlexiTRUST Certification Authority (FlexiTRUST-CA). Weiterhin werden die Voraussetzungen für den Betrieb der FlexiTRUST-CA und die für die Installation der verschiedenen Komponenten notwendigen Maßnahmen erleutert.

Es richtet sich an Systemadministratoren, die die CA installieren und administrieren. Vorausgesetzt werden Kenntnisse in UNIX- und/oder Windows-Systemadministration und Administration von MySQL Datenbanken.

1.2 Struktur dieses Handbuches

Kapitel	Inhalt
2	Ein Überblick über das System der FlexiTRUST-CA.
3	Anwendungsfälle untergliedert nach Arbeitsabläufen.
4	Beschreibung der Systemarchitektur.
5	Umsetzung der fachlichen Anforderungen.
7	Installation und Administration des Systems.
8	Betrieb des Systems.

Kapitel 2

Die FlexiTRUST-CA

MW

2.1 Systemüberblick

2.2 Zielumgebung

2.2.1 Hardware

2.2.2 Software

2.3 Verwendete Werkzeuge

Wesentliche Werkzeuge im Rahmen des Designs und der Entwicklung sind:

- Together/J Version 6.0 für die Geschäftsklassenmodellierung und als Front-End zum Debugging,
- Java Development Kit 1.4.0
- Emacs/JDE für die Source-Code-Erstellung und als Front-End für die Kompilierung und Versionsverwaltung
- CVS für die Versionsverwaltung

Kapitel 3

Spezifikation der Anwendungsfälle und Klassen

3.1 Spezifikation der Anwendungsfälle JWS

3.1.1 Anwendungsfälle der Domain Bench

3.1.2 Anwendungsfälle der Domain Stock

3.1.3 Anwendungsfälle der Domain Worker

3.2 Spezifikation der Anwendungsfälle CA

3.2.1 Anwendungsfälle der Domain Bench

3.2.2 Anwendungsfälle der Domain Configuration

3.2.3 Anwendungsfälle der Domain Entrance

3.2.4 Anwendungsfälle der Domain Exit

3.2.5 Anwendungsfälle der Domain Request

3.2.6 Anwendungsfälle der Domain Stock

Kapitel 4

Systemarchitektur

4.1 Struktur Java 2 Enterprise Edition

4.2 Struktur FlexiTRUST-JWS

4.2.1 Architektur EJB-Tier Bench

4.2.2 Architektur EJB-Tier Stock

4.2.3 Architektur EJB-Tier Worker

4.3 Struktur FlexiTRUST-CA

4.3.1 Architektur EJB-Tier Bench

4.3.2 Architektur EJB-Tier Configuration

4.3.3 Architektur EJB-Tier Entrance

4.3.4 Architektur EJB-Tier Exit

4.3.5 Architektur EJB-Tier Request

4.3.6 Architektur EJB-Tier Stock

Kapitel 5

Umsetzung der fachlichen Anforderungen

5.1 Clustering

5.2 Farming

5.3 Error-Logging

5.4 Schnittstellen zu externen Systemen

5.4.1 Dateischnittstelle

5.4.2 Hardwareschnittstelle

5.5 Konfiguration und Management

5.6 Datenbank

LF

5.6.1 Modell

LOGIN AND PWD

usr: flexiTrust; pwd: flexiTrust

– issuer table

```
DROP TABLE IF EXISTS issuers;
CREATE TABLE issuers (
issuerPK INT NOT NULL AUTO_INCREMENT,
issuerDN VARCHAR(255) NOT NULL,
nextCertId BIGINT NOT NULL,
issuerPrivKey BLOB,
certificate BLOB,
keyPass VARCHAR(16),
sigAlg VARCHAR(16),
secProv VARCHAR(255),
certChain BLOB,
PRIMARY KEY (issuerPK),
UNIQUE (issuerDN)
);
```

– admin table

```
DROP TABLE IF EXISTS admins;
CREATE TABLE admins (
adminPK INT NOT NULL AUTO_INCREMENT,
adminDN VARCHAR(255) NOT NULL,
nextCertId BIGINT NOT NULL,
adminPrivKey BLOB,
certificate BLOB,
keyPass VARCHAR(16),
sigAlg VARCHAR(16),
secProv VARCHAR(255),
certChain BLOB,
PRIMARY KEY (adminPK),
UNIQUE (adminDN)
```

```
);  
- user table  
DROP TABLE IF EXISTS users;  
CREATE TABLE users (  
userPK INT NOT NULL AUTO_INCREMENT,  
userDN VARCHAR(255),  
PRIMARY KEY (userPK),  
UNIQUE (userDN)  
);  
- certificate table  
DROP TABLE IF EXISTS certificates;  
CREATE TABLE certificates (  
certificatePK VARCHAR(255) NOT NULL,  
adminPK INT NOT NULL,  
requestPK VARCHAR(255) NOT NULL,  
certId BIGINT NOT NULL,  
certificate BLOB NOT NULL,  
userPK INT NOT NULL,  
validFrom BIGINT NOT NULL,  
validUntil BIGINT NOT NULL,  
revocPass VARCHAR(16) NOT NULL,  
certRole INT NOT NULL,  
PRIMARY KEY (adminPK, certId),  
FOREIGN KEY (adminPK) REFERENCES admins(adminPK),  
FOREIGN KEY (requestPK) REFERENCES requests(requestPK),  
FOREIGN KEY (userDN) REFERENCES users(userDN),  
UNIQUE (certificatePK)  
);  
- crlEntry table
```

```
DROP TABLE IF EXISTS crlEntries;
CREATE TABLE crlEntries (
  adminPK INT NOT NULL AUTO_INCREMENT,
  certId BIGINT NOT NULL,
  crlEntry BLOB NOT NULL,
  validUntil BIGINT NOT NULL,
  FOREIGN KEY(adminPK) REFERENCES admins(adminPK),
  FOREIGN KEY(certId) REFERENCES certificates(certId),
  PRIMARY KEY(adminPK, certId)
);
-- privateKey table
DROP TABLE IF EXISTS privKeys;
CREATE TABLE privKeys (
  adminPK INT NOT NULL,
  certId BIGINT NOT NULL,
  userPrivKey BLOB NOT NULL,
  keyPass VARCHAR(16) NOT NULL,
  PRIMARY KEY(adminPK, certId)
);
-- card table
DROP TABLE IF EXISTS cards;
CREATE TABLE cards (
  adminPK INT NOT NULL,
  certId BIGINT NOT NULL,
  cardInfo BLOB NOT NULL,
  FOREIGN KEY(adminPK) REFERENCES admins(adminPK),
  FOREIGN KEY(certId) REFERENCES certificates(certId),
  PRIMARY KEY(adminPK, certId)
);
```

– request table

```
DROP TABLE IF EXISTS requests;
CREATE TABLE requests (
requestPK VARCHAR(255) NOT NULL,
state INT NOT NULL,
creationDate BIGINT NOT NULL,
retirementDate BIGINT NOT NULL,
type INT NOT NULL,
sigDat BLOB NOT NULL,
nextBench VARCHAR(16) NOT NULL,
PRIMARY KEY (requestPK)
);
```

– x509CrtRequest specific table

```
DROP TABLE IF EXISTS x509CrtRequests;
CREATE TABLE x509CrtRequests (
requestPK VARCHAR(255) NOT NULL,
x509Certificate BLOB NOT NULL,
revocPass VARCHAR(16) NOT NULL,
privKey BLOB NOT NULL,
keyPass VARCHAR(16),
PRIMARY KEY (requestPK),
FOREIGN KEY (requestPK) REFERENCES requests
);
```

– x509CrlRequest specific table

```
DROP TABLE IF EXISTS x509CrlRequests;
CREATE TABLE x509CrlRequests (
requestPK VARCHAR(255) NOT NULL,
issuer BLOB NOT NULL,
crlEntry BLOB NOT NULL,
```

```
revocPass VARCHAR(16),  
PRIMARY KEY (requestPK),  
FOREIGN KEY (requestPK) REFERENCES requests  
);
```

– initKeyRequest specific table

```
DROP TABLE IF EXISTS initKeyRequests;  
CREATE TABLE initKeyRequests (  
requestPK VARCHAR(255) NOT NULL,  
issuerDN VARCHAR(255),  
nextCertId BIGINT,  
issuerPrivKey BLOB,  
certificate BLOB,  
keyPass VARCHAR(16),  
sigAlg VARCHAR(16),  
revocPass VARCHAR(16),  
secProv VARCHAR(255),  
certChain BLOB,  
PRIMARY KEY (requestPK),  
FOREIGN KEY (requestPK) REFERENCES requests  
);
```

– issuerRequest specific table

```
DROP TABLE IF EXISTS issuerRequests;  
CREATE TABLE issuerRequests (  
requestPK VARCHAR(255) NOT NULL,  
issuerDN VARCHAR(255),  
nextCertId BIGINT,  
issuerPrivKey BLOB,  
certificate BLOB,  
keyPass VARCHAR(16),
```

```
sigAlg VARCHAR(16),
revocPass VARCHAR(16),
secProv VARCHAR(255),
certChain BLOB,
PRIMARY KEY (requestPK),
FOREIGN KEY (requestPK) REFERENCES requests
);
– adminRequest specific table
DROP TABLE IF EXISTS adminRequests;
CREATE TABLE adminRequests (
requestPK VARCHAR(255) NOT NULL,
adminDN VARCHAR(255),
nextCertId BIGINT,
adminPrivKey BLOB,
certificate BLOB,
keyPass VARCHAR(16),
sigAlg VARCHAR(16),
revocPass VARCHAR(16),
secProv VARCHAR(255),
certChain BLOB,
PRIMARY KEY (requestPK),
FOREIGN KEY (requestPK) REFERENCES requests
);
```

Kapitel 6

Beschreibung der Pakete

Kapitel 7

Installation des Systems

MW

7.1 Systemarchitektur

7.2 Datenbank Server

7.2.1 Systemvoraussetzungen

Systemvoraussetzungen Hardware

Die Voraussetzungen beziehen sich auf einen Clusternode

- Pentiumclass System mit > 500 MHz
- 256MB Ram
- 50 MB HD

Systemvoraussetzungen Software

- Betriebssystem Linux (Kernel 2.4), Solaris 7 oder Microsoft Windows 2000.
- DBMS MySQL 3.23.52 oder höher
- JDK 1.4.0
- Ant 1.5

7.2.2 Vorbereitende Schritte

Es muss eine Datenbank mit dem Namen

caDB

angelegt werden.

Weiterhin ist ein Benutzer <flexiTrust> mit Passwort <flexiTrust> für diese Datenbank anzulegen. Dieser Benutzer muss von allen im Cluster vorhandenen Nodes Zugriff auf die Datenbank haben¹.

7.2.3 Installation der Anwendung

7.2.4 Update der Anwendung

7.2.5 Deinstallation

7.3 Applikation Server

7.3.1 Systemvoraussetzungen

Systemvoraussetzungen Hardware

Die Voraussetzungen beziehen sich auf einen Clusternode

- Pentiumclass System mit > 500 MHz
- 256MB Ram
- 50 MB HD

Systemvoraussetzungen Software

- Betriebssystem Linux (Kernel 2.4), Solaris 7 oder Microsoft Windows 2000.
- Applicationserver: JBoss 3.0.2 oder höher

¹Im momentanen Entwicklungsstand der FlexiTRUST-CA ist weiterhin ein anonymer Benutzer ohne Passwort mit allen Berechtigungen und Zugriff von allen Hosts angelegt werden. Dieses wird in der endgültigen Version der CA nicht mehr notwendig sein, bzw. der Benutzer muss aufgrund der Sicherheitsrichtlinien entfernt werden.

- DBMS MySQL 3.23.52 oder höher
- JDK 1.4.0
- Ant 1.5

7.3.2 Vorbereitende Schritte

Installation von JBoss Application Server

Die Installation muss gemäß „JBoss 3.0 getting started“ (siehe <http://prdownloads.sourceforge.net/jboss/JBoss.3.0QuickStart.Draft3.pdf> Kapitel 2) erfolgen.

Environmentvariablen

Folgende Environmentvariablen müssen gesetzt sein:

- HOME (Bsp.: /home/username),
- JAVA_HOME (Bsp.: /usr/local/bin/java),
- JBOSS_HOME (Bsp.: ~/JBoss),
- FLEXITRUST_HOME (Bsp.: ~/FlexiPKI/FlexiTRUST)

FlexiTRUST-CA

Eine Distributionsversion von FlexiTRUST-CA kann über

```
ant dist
```

im Verzeichnis `ca/install` erzeugt werden. Durch diesen Vorgang werden zwei gepackte Tar-Archive im Verzeichnis `ca/dist` erstellt, die fertig konfigurierte JBoss-Server beinhalten. Das Archiv `FlexiTRUST-Server.tar.gz` stellt den Ein- und Ausgang der CA dar, das Archiv `FlexiTRUST-CA.tar.gz` beinhaltet das EJB-Tier. Die erzeugten Konfigurationen sind auf Clustering ausgelegt. Das bedeutet, dass im Cluster ein Application-Server FlexiTRUST-Server mit `p7In`, `p7Out` und `p7Err` läuft und auf beliebig vielem anderen Nodes jeweils ein FlexiTRUST-CA.

Sourcecodeversion von FlexiTRUST-CA

Die Quelltexte der FlexiTRUST-CA befinden sich im CVS-Repository

```
cd $FLEXITRUST_HOME
export CVSROOT=":ext:<username>@cdc-ultra1.cdc.informatik.tu--darmstadt.de:/cdc/Flexi
export CVS_RSH="ssh"
cvs co -r ejb all
cvs co -r ejb ca
cvs co -r ejb jws
```

Übersetzt und deployed werden kann der Server-Teil durch ein

```
ant deploy-server
```

im Verzeichnis ca/src. Das EJB-Tier durch ein

```
ant deploy-ca-ejbs
```

im gleichen Verzeichnis.

7.3.3 Installation der Anwendung

Konfiguration der Umgebung

Im Verzeichnis `$FLEXITRUST_HOME/ca/personal-props` ist die Datei `ant.properties` wie folgt anzupassen:

```
all.home = <Verzeichnisname von all>
ca.home = <Verzeichnisname von ca>
jws.home = <Verzeichnisname von jws>
appserver.deploy = server/FlexiTRUST/deploy
mysql.ip = <IP des DBMS>
mysql.port = <Listenport der DBMS>
partition.name = <Partitionsname der FlexiTRUST-Partition>
```

Anlegen einer initialen Datenbank

Im Verzeichnis `$FLEXITRUST_HOME/ca/src` ist ein

```
ant db-install
```

auszuführen. Hierdurch wird das Datenbankschema der caDB erzeugt.

Konfiguration von JBoss

Im Verzeichnis `$FLEXITRUST_HOME/ca/src` befindet sich ein Ant-Buildfile für die automatische Konfiguration des Application-Servers. Sie wird durch

```
ant jboss-install
```

ausgelöst.

Start des Application-Servers

Der Application-Server mit FlexiTRUST-Konfiguration wird über den Aufruf eines Startskripts im Verzeichnis `$JBASS_HOME/bin` gestartet.

```
./startFlexiTRUST.sh
```

Deployment der FlexiTRUST-CA

Hierbei werden zwei Fälle unterschieden:

1. FlexiTRUST im Cluster und
2. FlexiTRUST ohne Clustering

Soll die CA innerhalb eines Clusters betrieben werden ist im Verzeichnis `$FLEXITRUST_HOME/ca/src` ein

```
ant farm-ca-ejb
```

auszuführen. Hierdurch werden alle Packages die für den Ablauf der CA notwendig sind erzeugt und dem Application Server für das Hot-Deployment und Farming übergeben.

Das Betreiben der CA ohne Clustering erfolge durch ein

```
ant deploy-ca-ejb
```

im gleichen Verzeichnis.

Die FlexiTRUST-CA ist nun einsatzbereit.

7.3.4 Update der Anwendung

Update von JBoss

Ein Update des Application-Server wird analog zur Installation (vgl. Abschnitt 7.3.2) durchgeführt.

Anschliessend muss der Server analog zu Abschnitt 7.3.3 konfiguriert werden.

Update der FlexiTRUST-CA

Das Update der CA erfolgt durch ein CVS-Update der Module all, ca und jws.

Anschliessend müssen die aktualisierten Packages analog zu Abschnitt 7.3.3 deployed werden.

7.3.5 Deinstallation

Die Deinstallation erfolgt durch einfaches Löschen der Verzeichnisse `$JBOSS_HOME` und `$FLEXITRUST_HOME`.

7.3.6 Verzeichnisstruktur

Root-Verzeichnis von JBoss

```
$JBOSS_HOME
```

Start-Skript für die FlexiTRUST-Konfiguration

```
bin/startFlexiTrust.sh
```

FlexiTRUST-Konfiguration

```
server/flexiTrust/conf  
server/flexiTrust/deploy  
server/flexiTrust/farm  
server/flexiTrust/lib  
server/flexiTrust/log
```

7.4 Pkcs7Server

7.4.1 Systemvoraussetzungen

Analog zu 7.3.1.

7.4.2 Vorbereitende Schritte

Analog zu 7.3.2.

7.4.3 Installation der Anwendung

Analog zu 7.3.3. Nicht durchgeführt werden muss das Deployment der FlexiTRUST-CA.

Deployment des Pkcs7Servers

Im Verzeichnis `$FLEXITRUST_HOME/ca/src` ist ein

```
ant deploy-server
```

auszuführen. Hierdurch werden alle Packages die für den Betrieb des Pkcs7Servers notwendig sind erzeugt und dem Application Server für das Hot-Deployment übergeben.

Der Pkcs7Server ist nun einsatzbereit.

Variante 1: Deployment des Pkcs7Servers zusammen mit der FlexiTRUST-CA

Im Verzeichnis `$FLEXITRUST_HOME/ca/src` ist ein

```
ant farm
```

oder

```
ant deploy
```

auszuführen. In der Variante `farm` werden die `ca-lib.jar` und `ca-ejbs.ear` über `farming` an alle bekannten Nodes verteilt. Die Variante `deploy` erzeugt eine vollständige Konfiguration für nur einen Server ohne Clustering.

Variante 2: Installation und Konfiguration der FlexiTRUST-CA von Scratch

Im Verzeichnis `$FLEXITRUST_HOME/ca/src` ist ein

```
ant farm-scratch
```

oder

```
ant deploy-scratch
```

auszuführen. Hierbei wird das gesamte System konfiguriert und alle Komponenten der CA dem Application-Server für das Deployment übergeben. Einzige Voraussetzung ist eine elementare Installation des JBoss und MySQL.

7.4.4 Update der Anwendung

Analog zu 7.3.4.

7.4.5 Deinstallation

Analog zu 7.3.5.

Kapitel 8

Betrieb des Systems

8.1 System überwachen

8.1.1 Datenbankserver

8.1.2 Applikationsserver

8.1.3 Pkcs7Server

Abkürzungsverzeichnis

API	Application-Programming-Interface
BMP	Bean-Managed-Persistence
BO	Business-Object
CMP	Container-Managed-Persistence
CORBA	Common-Object-Request-Broker-Architecture
DAO	Data-Access-Object
EJB	Enterprise-Java-Bean
IOR	Interoperable-Object-Reference
J2EE	Java-2-Platform-Enterprise-Edition
JDBC	Java-Database-Connectivity
JDK	Java-Development-Kit
JMS	Java-Message-Service
JNDI	Java-Naming-and-Directory-Interface
JVM	Java-Virtual-Machine
MDB	Message-Driven-Bean
MOM	Message-Oriented-Middleware
ORB	Object-Request-Broker
PTP	Point-To-Point
Pub	Publish
RR	Request-Reply
SQL	Structured-Query-Language
Sub	Subscribe
XML	EXtensible-Markup-Language

Literaturverzeichnis