Leistungsbeschreibung

TRAVIC-EBICS-Kernel V3.0

Dokumentversion:	1
Status:	Abgenommen
Datum:	30.05.2018

Versionsführung für Dokument **TRAVIC-EBICS-Kernel V3.0 Leistungsbeschreibung V1.docx**

Name	Datum	Dokument-version	Bemerkungen
Michael Lembcke	11.05.2018	1	Initialversion zur Programmversion 3.0

Inhaltsverzeichnis

1 Einleitung

Der EBICS-Standard wurde in der ursprünglichen Form für den sicheren Datenaustausch im Zahlungsverkehr zwischen Firmenkunden und Banken vom deutschen ZKA (heute DK) definiert und wird heute von der in Brüssel ansässigen EBICS-Gesellschaft gepflegt und weiterentwickelt. Der EBICS-Standard basiert auf zeitgemäßer IP- und XML-Technologie. Er wird bisher in Europa in unterschiedlichen Ausprägungen genutzt.

Der EBICS-Standard ist in Bankrechnern und in Kundenprodukten bzw. Kundenanwendungen entsprechend umgesetzt. Für die Kundenseite ist es sinnvoll, die Implementierung der Kommunikation über EBICS in einem Kommunikationsbaustein zu kapseln, der in das Kundenprodukt eingebunden wird. Der EBICS-Kernel ist genau dieser Kommunikationsbaustein.

Die grundlegende Aufgabe des EBICS-Kernels ist die Kommunikation mit den bankseitigen Servern. Zudem dient der EBICS-Kernel dazu, die Nutzdaten auf- und nachzubereiten und die kryptografischen Funktionen für die Sicherheit bereitzustellen.

Wesentliche Aufgaben des EBICS-Kernels sind das Senden und Empfangen von EBICS-Nachrichten. Die notwendigen Informationen werden vom Kundenprodukt bereitgestellt und zur Übertragung an einen bankseitigen EBICS-Server beauftragt. Umgekehrt werden die EBICS-Nachrichten von einem bankseitigen EBICS-Server angenommen, entschlüsselt und dem Kundenprodukt bereitgestellt. Zudem stellt der EBICS-Kernel kryptografische Funktionen für die Sicherheit bereit. Zur Kryptografie gehören die kryptografischen Prozesse wie z. B: Schlüsselgenerierung oder Signieren und Verifizieren von Nachrichten.

Der EBICS-Kernel V3.0 für Kundenprodukte basiert auf dem Stand der EBICS-Spezifikation, Version 2.5 (Schema H004), der Vorgänger-Version 2.4 (Schema H003) und der Version 3.0 (Schema H005). Darüber hinaus unterstützt der EBICS-Kernel das Schema H000 für die Auftragsart HEV. Der EBICS-Kernel entspricht der DK-Spezifikation der EBICS-Anbindung inklusive aller EBICS-Transaktionen, deren Implementierung auf Bankseite optional ist.

2 Technische Rahmenbedingungen

In diesem Kapitel sind die technischen Rahmenbedingungen und die Schnittstellen des EBICS-Kernels beschrieben.

- Systemtechnik

  Der EBICS-Kernel ist in Java realisiert. Nach außen werden zwei Schnittstellen zur Verfügung gestellt:

  - Eine Java-Schnittstelle, für deren Nutzung die Installation einer Standard Java-Laufzeitumgebung (J2SE 1.7) erforderlich ist.
  - Eine C-Schnittstelle mit einer DLL, die von einem C-Programm aus aufgerufen werden kann. Die Installation einer Standard-Java-Laufzeitumgebung ist für die C-Schnittstelle nicht notwendig. Die Kernel-DLL erfordert jedoch weitere DLLs und andere Dateien, die zusammen mit der Kernel-DLL installiert werden müssen. Die C-Schnittstelle wird nur für die MS Windows-Betriebssysteme angeboten.

- Multiuserfähigkeit

  Der Kernel kann grundsätzlich in einem Multiusersystem eingesetzt werden. Einschränkungen betreffen z. B. den Einsatz von Kryptografie-Hardware wie Chipkartenlesern.

Schnittstellenbeschreibung

- Programmschnittstelle (API)

  Der EBICS-Kernel erhält alle Informationen über die API bzw. stellt dem aufrufenden System alle Informationen über die API bereit.

- Konfigurationsdaten/Stammdaten

  Der EBICS-Kernel hat keine eigenen Konfigurations- und Stammdaten, die für die Nutzung der externen Schnittstelle erforderlich sind. Der Kernel wird beim Initiieren einer EBICS-Transaktion über seine API konfiguriert.

- EBICS-Version

  Die Schnittstelle enthält einen Parameter für die jeweils zu benutzende EBICS-Version.

- bankfachliche Nutzdaten

  Die reinen bankfachlichen Informationen (z. B. Zahlungsaufträge und Kontoauszüge) sind für den EBICS-Kernel transparent, d. h. sie werden unverändert an den bankseitigen EBICS-Server übergeben und vom Kernel nicht interpretiert. Diese Daten werden zwischen Kernel und Kundenprodukt alternativ über die API oder über Dateien ausgetauscht.

- sonstige Nutzdaten

  Die sonstigen Nutzdaten umfassen

  - elektronische Unterschriften
  - organisatorische Informationen (Bankparameter, Kunden- und Teilnehmerinformationen, abholbare Aufträge)
  - Informationen zur Verteilten Elektronischen Unterschrift
  - Schlüsselinformationen (Schlüssel des Kunden, Schlüssel der Bank)

  Die hier aufgeführten sonstigen Nutzdaten werden grundsätzlich nur über die API als Objekt (Java) bzw. Struktur (C) mit der Anwendung ausgetauscht.

  Ausnahme: Wenn für die HVT-Transaktion (VEU-Transaktionsdetails abrufen) die komplette Auftragsdatei abgeholt wird, kann diese auch als Datei an das Kundenprodukt übergeben werden.

- Interner Kernel-Zustand

  Der Kernel hält keinen internen Zustand persistent in einer Datenbank oder im Dateisystem vor.

  Für die Unterstützung von Restart und Recovery müssen einige Informationen über den internen Zustand des Kernels persistent vorgehalten werden. Diese Informationen speichert der Kernel im Kundensystem über entsprechende Callback-Funktionen.

- Rückmeldungen des EBICS-Kernels

  Der EBICS-Kernel liefert deutsche oder englische Rückmeldungstexte. Die zu verwendende Sprache wird über die API konfiguriert.

- Identifikationen

  Die Kunden-ID, die Teilnehmer-ID und der Host-Name können eine maximale Länge von 35 Zeichen haben.

3 Auftragsmanagement

Die Verbindungsverwaltung erfolgt über den EBICS-Kernel. Alle nicht behebbaren Fehlermeldungen teilt der EBICS-Kernel dem aufrufenden Programm mit.

Für die Kommunikation mit den Banksystemen stellt der EBICS-Kernel grundlegende Funktionalitäten bzw. Teilfunktionalitäten zur Verfügung. Diese Funktionalitäten werden implizit durch den EBICS-Kernel erbracht:

- Dialog mit der Bank führen
- Unterstützung der serverseitigen Vergabe von Auftragsnummern
- Nutzdaten komprimieren und dekomprimieren
- Base64-Kodierung und -Dekodierung
- Nutzdaten verschlüsseln und entschlüsseln (Verfahren E002)
- Nutzdaten segmentieren und desegmentieren
- Authentifizierungsunterschrift erstellen und verifizieren (Verfahren X002)
- XML-Datenstrom erstellen bzw. ankommenden XML-Datenstrom parsen
- Restart und Recovery

Der EBICS-Kernel bietet Funktionen, um EBICS-Transaktionen auszuführen. Das sind im Wesentlichen der allgemeine Upload und Download von Daten.

Der Upload umfasst das Senden von Nutzdaten, d. h. von Auftragsinformationen, Elektronischen Unterschriften und kryptografischen Schlüsseln an einen bankseitigen EBICS-Server. Das Kundensystem stellt dem EBICS-Kernel die Nutzdaten wahlweise über die API, d. h. im Funktionsaufruf, oder in einer Datei bereit. Ein Upload-Auftrag kann Daten für die Vorabprüfung eines Auftrags durch die Bank enthalten. Im Rahmen der Funktionen zum Upload kann der EBICS-Kernel bankfachliche Elektronische Unterschriften erstellen und an das Kundenprodukt zurückliefern.

Der Download unterstützt das Abholen von Nutzdaten von einem bankseitigen EBICS-Server. Die Nutzdaten stellt der EBICS-Kernel dem aufrufenden Kundensystem wahlweise über die API oder in einer Datei bereit.

Mit EBICS 2.5 transportieren alle VEU-Auftragsarten die Datei-Attribute, die in den Auftragsarten FUL und FDL bereits für den französischen Markt genutzt werden. Damit ist die VEU auch im französischen EBICS-Profil umsetzbar.

Der Kernel bietet außerdem die folgenden spezialisierten Upload-Aufträge:

FUL französische Auftragsart für den Upload mit Formatparameter sowie Start- und Ende-Datum

H3K alle öffentlichen Schlüssel (Signatur-, Authentifikations- und Verschlüsselungsschlüssel) eines Teilnehmers und optional ihre Zertifikate für die Initialisierung senden

HCA öffentlichen Authentifikations- und Verschlüsselungsschlüssel aktualisieren

HCS öffentlichen bankfachlichen Schlüssel und öffentlichen Authentifikations- und Verschlüsselungsschlüssel eines Teilnehmers aktualisieren

HIA neu erzeugten öffentlichen Authentifikations- und Verschlüsselungsschlüssel eines Teilnehmers senden und Informationen, die zur Erstellung des Initialisierungsbriefs benötigt werden, an das Kundensystem geben

HSA FTAM-Teilnehmer initialisieren

HVE bankfachliche Signatur der VEU-Prozessverarbeitung hinzufügen

HVS VEU-Storno

INI öffentlichen bankfachlichen Schlüssel eines Teilnehmers senden und Informationen, die zur Erstellung des Initialisierungsbriefs benötigt werden, an das Kundensystem geben

PUB öffentlichen bankfachlichen Schlüssel eines Teilnehmers senden

Der Kernel bietet des Weiteren die folgenden spezialisierten Download-Aufträge:

FDL französische Auftragsart für den Download mit Formatparameter sowie Start- und Ende-Datum

HAA abrufbare Auftragsarten abholen

HAC Kundenprotokoll im XML-Format abholen

HEV Liste der vom EBICS-Server unterstützten Schemata abholen

HKD Kunden- und Teilnehmerinformationen des Kunden abrufen

HPB Download der bankfachlichen Schlüssel

HTD Kunden- und Teilnehmerinformationen des Teilnehmers abrufen

HVD VEU-Status abrufen

HVT VEU-Transaktionsdetails abrufen

HVU VEU-Übersicht abholen

HVZ VEU-Übersicht mit Zusatzinformationen abholen, neu mit Kennzeichen `isCredit` (Angabe, ob es sich um eine Lastschrift oder eine Gutschrift handelt)

Mit EBICS 3.0 unterstützt der EBICS-Kernel zusätzlich die Business Transfer Parameter (BTF) BTU (für Upload) und BTD (für Download). Diese lösen mittelfristig im Sinne der EBICS-Harmonisierung die operativen Auftragsarten und die Fileformat-Parameter (FUL und FDL) ab.

4 Kryptografie

Der EBICS-Kernel unterstützt die drei Sicherheitsverfahren, die von EBICS vorgesehen sind:

- Bankfachliche Elektronische Unterschrift mit den Sicherheitsverfahren A005 und A006
- Verschlüsselung mit dem Sicherheitsverfahren E002
- Autorisierung von Nachrichten mit dem Sicherheitsverfahren X002

Der EBICS-Kernel unterstützt als Sicherheitsmedien die Chipkarte sowie Kryptografie in Software. Zugriffe auf kryptografische Schlüssel, die auf Wechselspeichermedien (USB-Stick etc.) liegen, werden vom EBICS-Kernel unterstützt. Weiterhin können kryptografische Schlüssel von RDH2-Disketten in Form eines Byte-Arrays an den EBICS-Kernel übergeben werden.

Die Funktionen für den Upload und den Download laufen hinsichtlich der Kryptografie mehrschrittig zwischen dem Kundensystem und dem EBICS-Kernel ab:

- Das Kundensystem ruft eine Funktion zum Upload bzw. Download auf. Der Funktionsaufruf enthält u. a. Callback-Funktionen für die Kryptografie.
- Der EBICS-Kernel ruft für alle kryptografischen Funktionen, für die ein privater Schlüssel des Teilnehmers benötigt wird, einen Callback auf.
- Das Kundensystem ermittelt den ggf. erforderlichen privaten Schlüssel, wenn der Schlüssel auf einem Speichermedium (USB-Stick etc.) gespeichert ist.
- Das Kundensystem kann für die auszuführende kryptografische Funktion (Signieren, Entschlüsseln) eine entsprechende Funktion des EBICS-Kernels aufrufen. Der EBICS-Kernel stellt zum Signieren und Entschlüsseln unterschiedliche Funktionen für die Kryptografie mit Chipkarte und für die Kryptografie in Software zur Verfügung.
- Das Kundensystem gibt das Ergebnis der kryptografischen Funktion als Ergebnis des Callback an den EBICS-Kernel.

Der EBICS-Kernel stellt Funktionen für eine Schlüsselmigration bereit. Kryptografische Schlüssel, die in der Datenhaltung eines Kundensystems liegen, können in das Datenformat, das vom EBICS-Kernel unterstützt wird, migriert werden.

4.1 Kryptografie Chipkarte

Für die Verwendung von Chipkarten stehen Funktionen für folgende Zwecke zur Verfügung:

- Chipkarten-ID (CID) auslesen
- Daten signieren

- Daten entschlüsseln
- öffentliche Schlüssel der Chipkarte auslesen
- PIN ändern

Für die Funktionen zur E002-Verschlüsselung und X002-Signatur unterstützt der EBICS-Kernel Mehrschritttransaktionen: Diese Funktionen können mehrfach aufgerufen werden, ohne dass bei den Folgeaufrufen eine PIN-Eingabe erforderlich ist.

Für den Zugriff auf Chipkartenleser wird der Zugriff per PC/SC-Schnittstelle empfohlen. PC/SC ist ein internationaler Standard und für viele Chipkartenleser verfügbar.

Die bisher verwendete CT-API-Schnittstelle – mit der dafür notwendigen Konfigurationsdatei `hbcikrnl.ini` im Windows-Verzeichnis – wird weiter unterstützt, jedoch mittelfristig entfallen, da nur noch wenige Hersteller auf diesem Standard aufsetzen.

Zusätzlich zu den Standard-Chipkartenlesern unterstützt der EBICS-Kernel den Secoder S von Reiner SCT als so genannten installationslosen Chipkartenleser. Bei diesem Gerät muss kein Treiber installiert werden.

4.2 Kryptografie Software

Der EBICS-Kernel unterstützt folgende Funktionen für die Kryptografie in Software:

- Schlüssel generieren

  Schlüssel für die Elektronische Unterschrift (A005, A006), für die Verschlüsselung (E002) und für die Authentifikation (X002) generieren. Der EBICS-Kernel gibt ein generiertes Schlüsselpaar zur Speicherung an das Kundensystem zurück.

- A005/A006-Signatur erstellen

  Signatur mit dem privaten Schlüssel eines Teilnehmers erstellen

- E002-Verschlüsselung

  Transaktionsschlüssel mit privatem Verschlüsselungsschlüssel entschlüsseln

- X002-Signatur

  Authentifikationssignatur mit privatem Authentifikationsschlüssel erstellen

4.3 Zertifikate

Die öffentlichen kryptografischen Schlüssel können in X.509-Zertifikaten von einem EBICS-Client an einen EBICS-Server übertragen werden. Hierfür werden Methoden zur Befüllung von Zertifikatstrukturen angeboten. Das Benut-

zerzertifikat für den A00x-Schlüssel steht dabei stets an erster Position in der Kette der Zertifikate.

4.4 Signatur-Token 3SKey*

Der EBICS-Kernel unterstützt optional die Verwendung von Signatur-Token des Typs 3SKey von SWIFT. Diese Art von Signatur-Token greift nach dem Einstecken in den Rechner auf den lokalen Zertifikatsspeicher des Windows-Systems zu.

Für die Java-Version des EBICS-Kernels wird eine Adapterkomponente zur Nutzung des 3SKey-Tokens bereitgestellt.

Bei den unterstützten Windows-Versionen muss der Lizenznehmer den EBICS-Kernel selbst an den lokalen Zertifikatsspeicher des Windows-Systems anpassen.

4.5 Sicherheitsinformationen

Sicherheitsinformationen wie die PIN der Chipkarte fordert der EBICS-Kernel zur Laufzeit über Callback-Funktionen an.

* optional lizenzierbar

5 Weitere Funktionalitäten

Der EBICS-Kernel bietet weiterhin Funktionalitäten zur Verwaltung von Zertifikaten, unterstützt HTTP-Proxies und erstellt konfigurierbar Trace- und Debug-Ausgaben.

5.1 Zertifikatsverwaltung

Zur Kommunikation per HTTPS identifiziert sich der Bankrechner über ein Zertifikat. Damit die HTTPS-Verbindung zustande kommen kann, muss der Kernel das Zertifikat verifizieren können. Zu diesem Zweck benutzt der Kernel einen Zertifikatsspeicher.

Bei der Java-Schnittstelle handelt es sich um die Datei `cacerts` der Java-Installation. Bei der C-Schnittstelle wird diese Datei mit ausgeliefert. Zur Verwaltung dieses Zertifikatsspeichers bietet der EBICS-Kernel die folgenden Funktionen an:

- Zertifikatskette vom Server lesen
- Zertifikat in den Zertifikatsspeicher einfügen
- Server-Zertifikatskette gegen den Zertifikatsspeicher prüfen
- Zertifikat aus dem Zertifikatsspeicher löschen
- Passwort ändern

5.2 Proxy-Server

Der EBICS-Kernel unterstützt die IP-Kommunikation über HTTP-Proxies. Für die C-Schnittstelle werden außerdem explizit SOCKS-Proxies unterstützt. Im Java-Kernel können SOCKS-Proxies durch die in Java vorgesehene Konfiguration über System-Properties verwendet werden. Der Java-Kernel unterstützt den NTLM-Proxy.

Die Proxies können mit oder ohne Authentifizierung benutzt werden.

5.3 Logging

Zur Fehleranalyse kann der EBICS-Kernel Trace- und Debug-Ausgaben erstellen. Der Trace-Level, d. h. der Umfang der Trace-Ausgaben ist einstellbar. Die Konfiguration und die Ausgabe der Trace-Informationen erfolgt abhängig von der Systemumgebung (C, Java).

5.4 Streaming*

Die Schnittstelle für die Dateiübergabe zwischen der nutzenden Anwendung und dem EBICS-Kernel zur Übertragung oder zum Empfang von Auftragsdateien bzw. Bereitstellungsdateien unterstützt Streams.

Für das Streaming bietet der EBICS-Kernel einen `InputStreamDataSource` und einen `OutputStreamDataStore`.

Diese Streaming-Funktion steht optional per Lizenz zur Verfügung.

* optional lizenzierbar

6 Systemvoraussetzungen

Der EBICS-Kernel ist für die folgenden Betriebsumgebungen freigegeben:

- eines der aufgeführten Betriebssysteme
  - Windows Server 2012
  - Windows Server 2008
  - Windows
  - AIX, Version 7
  - Solaris 10 (SunOS 5.10) oder Solaris 11 (SunOS 5.11)
  - Distribution SuSE, SLES 11
  - Distribution RedHat Enterprise Linux 6
- Java-Laufzeitumgebung
  - J2SE 1.7
- JET-Software (nur bei Nutzung der C-Schnittstelle)

6.1 Unterstützte Chipkartenleser

Für den Zugriff auf Chipkarten werden folgende Chipkartenleser unterstützt:

- Reiner SCT cyberJack pinpad
- Reiner SCT cyberJack ecom
- Reiner SCT cyberJack ecom plus
- Reiner SCT cyberJack RFID standard
- Reiner SCT cyberJack RFID komfort
- Reiner SCT Secoder
- Reiner SCT cyberJack compact
- KOBIL KAAN Advanced
- KOBIL Tribank
- SCM SCR 335
- SCM SCR 3310
- Omnikey CardMan Mobile 4040 PCMCIA
- Omnikey CardMan Mobile 6121 USB
- Omnikey CardMan Trust 3621 USB
- Omnikey CardMan Desktop 3121
- Omnikey CardMan Desktop 3111

6.2 Unterstützte Chipkarten

Als zulässige Schlüsselmedien werden folgende Chipkarten mit Signaturanwendung unterstützt:

- Bankensignaturkarte mit Betriebssystem SECCOS 5.0
- Bankensignaturkarte mit Betriebssystem SECCOS 6.1
- Bankensignaturkarte mit Betriebssystem SECCOS 6.2

7 Erweiterungen zur vorherigen Version

Folgende Erweiterungen wurden für diese Version auf Basis des EBICS-Kernels V2.4 vorgenommen:

- Unterstützung der EBICS-Version 3.0

8 Dokumentation

Der EBICS-Kernel wird mit folgender Dokumentation in deutscher oder englischer Sprache ausgeliefert:

Dokument	Inhalt
Installationshandbuch EBICS-Kernel C	Das Handbuch beschreibt die Installation der C-Variante des EBICS-Kernels. Es richtet sich an Systementwickler, die mit dem Kernel eine EBICS-fähige Kundenanwendung für den Zahlungsverkehr mit dem Banksystem entwickeln wollen.
Installationshandbuch EBICS-Kernel Java	Dieses Handbuch beschreibt die Installation der Java-Variante des EBICS-Kernels. Es richtet sich an Systementwickler, die mit dem Kernel eine EBICS-fähige Kundenanwendung für den Zahlungsverkehr mit dem Banksystem entwickeln wollen.
Entwicklerhandbuch EBICS-Kernel Java	Dieses Entwicklerhandbuch beschreibt die Java-Schnittstellen des EBICS-Kernels. Es richtet sich an Systementwickler, die mit dem Kernel eine EBICS-fähige Kundenanwendung für den Zahlungsverkehr mit dem Banksystem entwickeln wollen. Das Handbuch besteht aus folgenden Kapiteln: ■ Release Notes ■ Systembeschreibung ■ Architektur: Pakete ■ Verarbeitung ■ Neuerungen mit EBICS 3.0 ■ Besonderheiten für EBICS in Frankreich ■ Programmierbeispiele ■ Quellcode-Dokumentation (Javadoc)

Dokument	Inhalt
Entwicklerhandbuch EBICS-Kernel C	Dieses Entwicklerhandbuch beschreibt die C-Schnittstellen des EBICS-Kernels. Es richtet sich an Systementwickler, die mit dem Kernel eine EBICS-fähige Kundenanwendung für den Zahlungsverkehr mit dem Banksystem entwickeln wollen. Das Handbuch besteht aus folgenden Kapiteln: ■ Release Notes ■ Systembeschreibung ■ Architektur: Bibliotheken und Header-Dateien ■ Verarbeitung ■ Neuerungen mit EBICS 3.0 ■ Besonderheiten für EBICS in Frankreich ■ Programmierbeispiele ■ Quellcode-Dokumentation (ccdoc)

Simrockstr. 4
53113 Bonn
Tel.: +49 228 4495-0
Fax: +49 228 4495-7555

E-Mail: info@siz.de
Internet: www.siz.de