

Die neue Schnittstelle der deutschen Kreditwirtschaft für chipkartengestützte Anwendungen am PC

- Migration der Schnittstelle für GeldKarte-Zahlungen im Internet -

Medi Samsami [medi@samsami.name]

1 Einleitung

Das vorliegende Dokument ist ein Auszug aus der Diplomarbeit mit dem oben genannten Titel. Die Arbeit wurde bei der Firma SRC Security Research & Consulting GmbH verfasst und am 1. Juni 2004 beim Fachbereich Angewandte Informatik der University of Applied Sciences Bonn-Rhein/Sieg eingereicht. Die Diplomarbeit befasst sich mit der neuen Schnittstelle der deutschen Kreditwirtschaft für chipkartengestützte Anwendungen am PC. Im Mittelpunkt steht die Migration der Schnittstelle für GeldKarte-Zahlungen im Internet auf das neue Schnittstellenkonzept.

1.1 Ausgangslage

Gegenwärtig ist die erste realisierte Nutzungsmöglichkeit für Chipkarten der deutschen Kreditwirtschaft (ZKA-Chipkarten) am PC die Anwendung zum Bezahlen mit der GeldKarte im Internet. Der Nutzer benötigt für die Abwicklung eines Bezahlvorgangs einen internetfähigen PC, eine ZKA-Chipkarte mit der Anwendung "GeldKarte" sowie einen vom Zentralen Kreditausschuss (ZKA)¹ zugelassenen Chipkartenleser. Die Chipkarte führt im Zusammenspiel mit dem Kartenleser die sicherheitskritischen Teile des Transaktionsablaufs aus. Solche Chipkartenleser werden als Internet-Kundenterminals (IKT)² oder Kundenterminals (KT) bezeichnet.

Der Bezahlvorgang wird über ein Applet abgewickelt. Das Applet greift für die Durchführung der Transaktion über ein API (Application Programming Interface)³, also eine definierte Schnittstelle, auf eine Terminalanwendung (TA) auf dem PC des Kunden zu. Die TA führt nicht sicherheitskritische Bestandteile eines Transaktionsablaufs aus und realisiert die An-

¹ Im Zentralen Kreditausschuss (ZKA) sind seit 1932 die fünf Spitzenverbände der deutschen Kreditwirtschaft (Bundesverband der Deutschen Volksbanken und Raiffeisenbanken e. V., Bundesverband deutscher Banken e. V., Bundesverband Öffentlicher Banken Deutschlands e. V., Deutscher Sparkassen- und Giroverband e. V. und Verband deutscher Hypothekenbanken e. V.) zusammengeschlossen. Der ZKA versteht sich als Interessenvertretung der kreditwirtschaftlichen Spitzenverbände.

² Mit Display und Tastatur ausgestatteter Kartenleser, der eine sichere Umgebung für den Zugriff auf die Karte darstellt.

³ Ein API (Application Programming Interface) für ein Kundenterminal, im Folgenden vereinfacht KT-API genannt, ist eine festgelegte Sammlung von Funktionen für den Zugriff auf eine Terminalanwendung (TA) und die zugehörige Kundenterminalanwendung (KTA).

bindung des KT. Die im KT implementierten Funktionen zur Durchführung des Bezahlvorganges werden logisch in einer sog. Kundenterminalanwendung (KTA) zusammengefasst. In Abbildung 1 ist die bisherige Architektur für das Bezahlen mit der GeldKarte dargestellt.

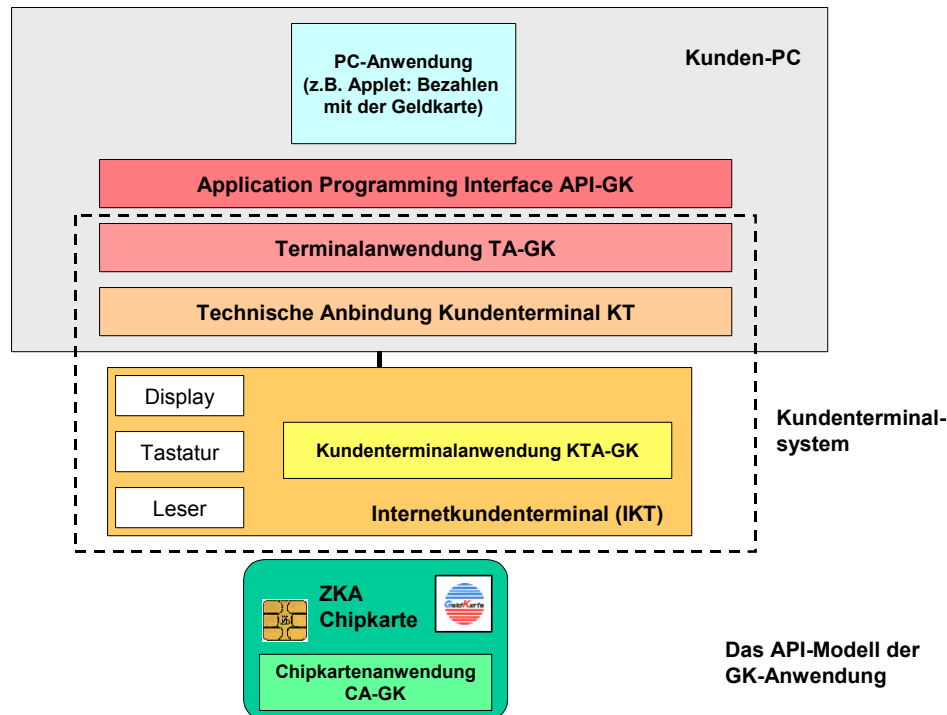


Abbildung 1: Bisherige Architektur am Beispiel „Bezahlen mit der GeldKarte“

Neue Dienste

Das Anwendungsspektrum der ZKA-Chipkarten wird zur Zeit stark erweitert. Dies betrifft insbesondere im PC- bzw. Internetumfeld einsetzbare Anwendungen. Möglich sind neben GeldKarte-Zahlungen die Nutzung

- der GeldKarte-Ladefunktion im Internet,
- der Signaturfunktion durch PC-Anwendungen,
- von EMV⁴-Bezahlverfahren im Internet,
- von EMV-Anwendungen zur Erzeugung von Transaktionsnummern (TAN) sowie

⁴ EMV steht für Europay, MasterCard und Visa und bezeichnet ein Standard für kartengestützte Debit- und Kredit-Zahlungsverfahren. EMV-Chipkarten werden zur Kundenauthentisierung im Rahmen von 3D-Secure bzw. SecureCode-Zahlungen eingesetzt. 3D-Secure bzw. SecureCode ist ein von Visa bzw. MasterCard entwickeltes Protokoll für Internetzahlungen.

- der Nutzung der Marktplatz- und Fahrschein-Anwendung im Internet.

Die genannten und alle weiteren Funktionen, die sich mit Hilfe von ZKA-Chipkarten im Heimbereich umsetzen lassen, werden im Folgenden auch als *Dienste* bezeichnet.

Die benötigte Architektur für die Umsetzung eines beliebigen Dienstes entspricht im Wesentlichen dem in der Abbildung 1 dargestellten Modell für GeldKarte-Zahlungen.

Da der Zugriff statt durch ein Applet ggf. auch durch eine lokal installierte Anwendung erfolgen kann (z.B. bei der Nutzung der Signaturfunktion durch E-Mail-Programme), wird die aufrufende Anwendung, unabhängig von ihrer Ausprägung, im Folgenden als PC-Anwendung (PCA) bezeichnet.

1.2 Das neue Schnittstellenkonzept

Vor dem Hintergrund der geplanten Erweiterung des Nutzungsumfangs ist das eingangs beschriebene Schnittstellenkonzept zu überdenken. Neben der existierenden GeldKarte-Anwendung führt die Ergänzung neuer Terminalanwendungen (Bibliotheken) auf der Basis der bisherigen Architektur zu folgenden Problemen:

- **Versionierung und Migration:** Es ist nicht möglich, mehrere Versionen einer Schnittstelle parallel zu betreiben. Bei der Installation einer weiteren Bibliothek wird die alte überschrieben. Es ist davon auszugehen, dass im Laufe der Zeit der Bedarf nach Konsolidierung einer Schnittstelle wächst, Zusatzfunktionen werden entwickelt und mögliche Fehler müssen behoben werden. Außerdem findet eine Umstellung der Systeme nicht zeitgleich statt. Ein Beispiel: Während Händler A sein Shop-System (PC-Anwendung) schon an die neue Schnittstelle angepasst hat, betreibt Händler B immer noch das alte System. Damit Kunde XY bei Händler A und B bezahlen kann, muss er beide Versionen auf seinem PC installieren. Es müssen also Mechanismen zur Migration vorgesehen werden.
- **Mehrere Chipkartenleser an einem PC:** Jeweils nur ein Leser kann denselben Dienst unterstützen, da die TA über den Namen der Bibliothek angesprochen wird. Bei der Installation eines zweiten Lesers werden die entsprechenden TA des ersten Lesers im Allgemeinen überschrieben. Das Betreiben mehrerer Leser an einem PC wird in der Zukunft häufiger auftreten: Z.B. erhält Kunde XY von seiner Bank A für Homebanking einen Leser, der HBCI und EMV-Anwendungen unterstützt, von seiner anderen Bank B erhält er für das Bezahlen im Internet einen multifunktionalen Leser, der ebenfalls EMV unterstützt. Für diesen Fall wird ein Schnittstellenkonzept benötigt, das die Auswahl aus mehreren installierten Lesern ermöglicht.

Gegenwärtig wird im Auftrag der deutschen Kreditwirtschaft an der Konsolidierung des Schnittstellenkonzepts gearbeitet. Die endgültige Verabschiedung ist für Ende 2004 geplant, die Umsetzung für Ende 2005. Das neue Konzept sieht die Etablierung einer zweiten Schnittstellenschicht oberhalb der TA vor. Diese Schicht wird wiederum in Form einer dy-

namischen Bibliothek realisiert und hat die Funktionsweise eines Wrapper-API⁵. Im Folgenden wird dieses Wrapper-API als **SCAMPI** (Smart Card Access Module Programming Interface⁶) bezeichnet. Das neue Konzept sieht drei Gruppen von API-Funktionen vor:

- **Administrationsfunktionen:** Die TA lassen sich beim SCAMPI mittels der Administrationsfunktionen `registerService()` und `deregisterService()` registrieren bzw. deregistrieren. Dabei kann jeder Dienst mehrfach, in mehreren Schnittstellenversionen und für mehrere Leser registriert werden.
- **Funktionen für das Transaktionsmanagement:** Das SCAMPI und alle TA enthalten die Funktionen `info()`, `open()`, `init()`, `fini()`, `close()` und `alive()` für das Transaktionsmanagement.
- **Ausführungsfunktion:** Sämtliche dienstspezifischen Funktionen werden mittels `execute()` ausgeführt. Dazu werden für jeden Dienst eigene Kommandos in Form von Bytelisten definiert.

Abbildung 2 zeigt die Architektur des SCAMPI-Schnittstellenkonzeptes. Dabei wird als Beispielszenario von einem PC mit zwei KT ausgegangen, die teilweise die gleichen Dienste unterstützen.

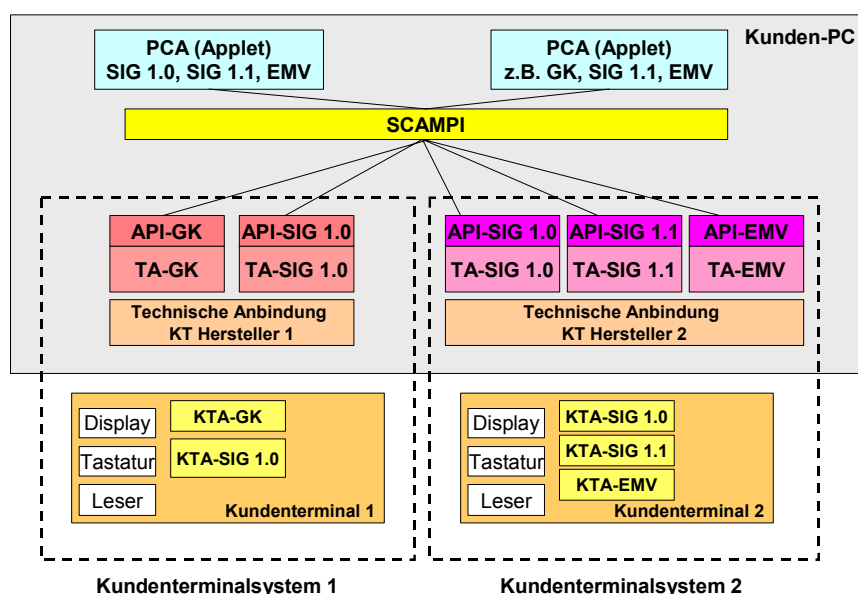


Abbildung 2: Architektur mit übergreifender Schnittstelle

⁵Ein Wrapper-API (von engl. wrapper = Hülle) stellt eine Schnittschicht ohne eigene Verarbeitungsfunktionalität dar, die darunter liegende Schichten verwaltet und deren Verarbeitungsfunktionen aufrufenden Komponenten zur Verfügung stellt.

⁶Die Terminalanwendungen werden in dieser Terminologie als "Smart Card Access Modules" bezeichnet.

Jede Transaktion wird also über dieselbe generische Schnittstelle abgewickelt. SCAMPI reicht die zur Abwicklung der Transaktion verwendeten Kommandos an die verwendete TA zur Ausführung weiter. Damit eine TA in SCAMPI integriert werden kann, muss ihr API den SCAMPI-Konventionen entsprechen.

Die Auswahl des Dienstes erfolgt immer durch die aufrufende PC-Anwendung. Darüber hinaus kann die PC-Anwendung die Verwendung eines bestimmten Lesers und einer bestimmten Version des Dienstes anfordern. Falls der Leser durch die von der PC-Anwendung übergebenen Parameter und die verfügbaren Leser nicht eindeutig festgelegt ist, wird der Nutzer vom SCAMPI zur Auswahl aufgefordert.

Die Verwendung dieses generischen Ansatzes erlaubt

- die Registrierung von TA (Bibliotheken) zur Laufzeit,
- die Definition neuer Dienste ohne Anpassung der allgemeinen Schnittstelle,
- die Änderung der Kommandos eines Dienstes ohne Einfluss auf die C-Schnittstelle und dadurch die Unterstützung verschiedener Versionen eines Dienstes (z.B. durch Registrierung mehrerer Bibliotheken für einen Dienst),
- die Verwaltung mehrerer Leser mit sich überschneidendem Einsatzspektrum durch Registrierung mehrerer Bibliotheken für einen Dienst.

1.3 Problemstellung

Das neue Schnittstellenkonzept betrifft Anwendungen, die sich in unterschiedlichen Entwicklungsstadien befinden. Während die meisten neueren Anwendungen (z. B. Fahrschein, EMV) noch im Konzeptionsstadium sind, wird die Anwendung zum Bezahlen mit der GeldKarte im Internet bereits eingesetzt. Das Kundenterminalsystem und die PC-Anwendung müssen dementsprechend auf das neue Konzept migriert werden.

In dem vorliegenden Dokument wird eine Strategie für die Migration von dem alten auf das neue Konzept entwickelt, da die Realisierung der PC-Anwendungen und der Komponenten des Kundenterminalsystems nicht unbedingt zeitgleich erfolgt: Es kann sein, dass beispielsweise neue TA bestimmter KT-Hersteller bereits auf dem Markt etabliert sind, einige Händler die neue Bezahlsoftware aber noch nicht in ihre Systeme integriert haben. Genau so gut ist es möglich, dass ein KT-Hersteller die neue Software erst spät oder gar nicht zur Verfügung stellt. Dabei sind sowohl die Kunden- als auch die Händlerinteressen zu berücksichtigen:

- Der Händler möchte so viele Kunden wie möglich unterstützen, auf Dauer aber möglichst nur eine Lösung einsetzen (geringerer Aufwand bei der Systempflege).

- Der Kunde möchte bei möglichst vielen Händlern bezahlen und dabei immer den vollen Funktionsumfang des neuen Systems nutzen können, z.B. Bezahlen eines Fahrscheins⁷ (vgl. Abschnitt 3.2).

1.4 Aufbau der Arbeit

Die vorliegende Arbeit umfasst neben der Einführung drei Abschnitte:

Abschnitt 2 befasst sich mit dem alten Konzept zum „Bezahlen mit der GeldKarte im Internet“. Nach einem kurzen Überblick über die Entwicklung der GeldKarte im Internet wird eine Einführung in das GeldKarte-System gegeben, dabei findet eine Gegenüberstellung zwischen der GeldKarte-Zahlung am *Point of Sale (POS)* und im Internet statt. Hierbei wird die Problematik deutlich, die durch den Einsatz der GeldKarte im Internet entsteht. Nachfolgend werden die Sicherheitsanforderungen an das System und das *Kundenterminal* als Mittel zu deren Erfüllung dargestellt. Anschließend wird das Konzept des bisherigen GeldKarte-API vorgestellt.

Abschnitt 3 befasst sich mit dem neuen Konzept zum „Bezahlen mit der GeldKarte im Internet“. Hierbei werden die Gründe für eine Überarbeitung des alten Konzepts deutlich. Im weiteren Verlauf werden das neue KT-Konzept und das SCAMPI vorgestellt. In Abschnitt 4 werden die möglichen Konzepte zur Migration betrachtet.

2 Das alte Schnittstellenkonzept

Zum besseren Verständnis der angesprochenen Problematik ist in Abschnitt 2.1 kurz der historische Ablauf der Entwicklung dargestellt, beginnend bei den Anfängen des GeldKarte-Systems im Internet bis hin zu dem neuen Konzept.

Die aktuell eingesetzte Architektur für GeldKarte-Zahlungen im Internet wird in den Dokumenten [KT GK] und [KT API] beschrieben. Abschnitt 2.2 gibt einen Überblick über dieses System.

In Abschnitt 2.3 werden die Sicherheitsanforderungen und das *Kundenterminal (KT)* als Mittel zu deren Erfüllung dargestellt. Die anwendungsspezifische Software des KT, die sogenannte *Kundenterminalanwendung (KTA)*, wird im Abschnitt 2.4 erläutert.

Schließlich wird im Abschnitt 2.5 das Konzept des bisherigen API der Anwendung zum „Bezahlen mit der GeldKarte“ vorgestellt.

⁷ Das Bezahlen eines elektronischen Fahrscheins steht in diesem Abschnitt stellvertretend für die Nutzung mehrerer Chipkartenanwendungen innerhalb einer Transaktion.

2.1 Einsatz der GeldKarte im Internet

Die Anwendung zum "Bezahlen mit der GeldKarte" ist die erste im Internet eingesetzte Anwendung der deutschen Kreditwirtschaft. Abbildung 2 stellt die Entwicklung der „GeldKarte im Internet“ dar.

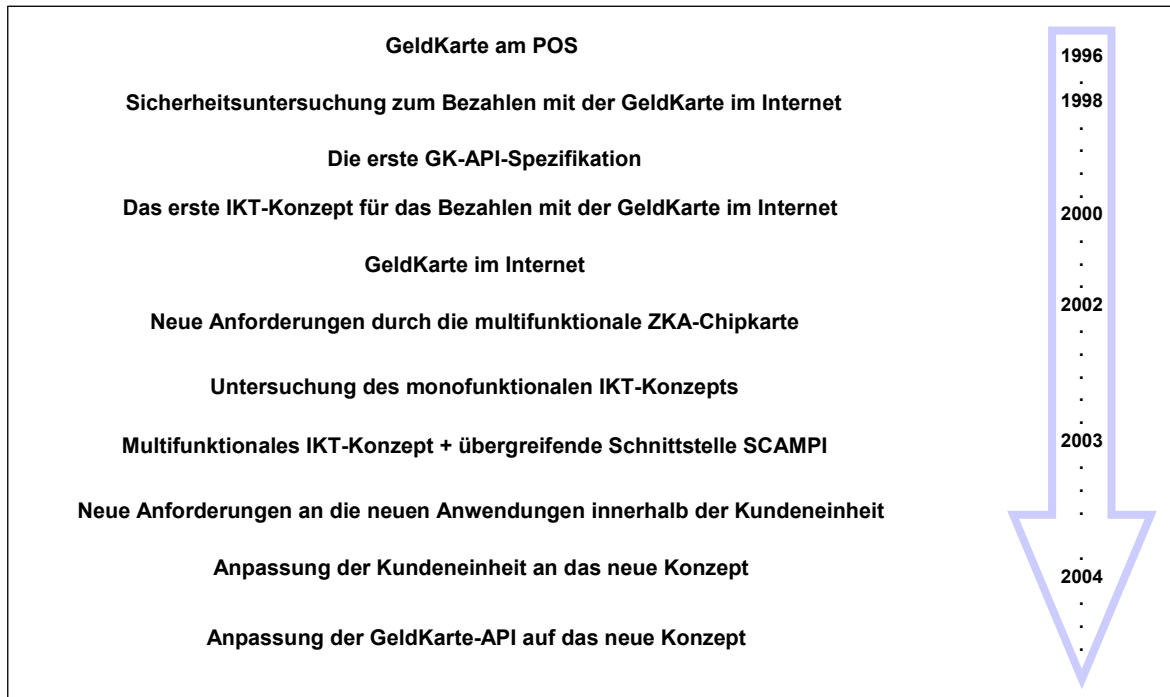


Abbildung 3: Entwicklung der „GeldKarte im Internet“

Die Grundkomponenten für eine GeldKarte-Zahlung im Internet sind auf Kundenseite die GeldKarte, ein PC, ein Chipkartenleser und die Bezahlsoftware.

PC und Internet stellen ein unsicheres Umfeld dar. Um trotzdem die Sicherheitsanforderungen des GeldKarte-Systems erfüllen zu können, wird ein spezieller Chipkartenleser eingesetzt, der als sichere Hardwarekomponente angesehen werden kann, das sogenannte **(Internet-) Kundenterminal (IKT)** (vgl. Abschnitt 2.3).

In der folgenden Tabelle findet eine Gegenüberstellung von GeldKarte-Zahlungen am **Point of Sale⁸ (POS)** und im Internet statt. Die dargestellten Unterschiede spielen eine wesentliche Rolle bei den Betrachtungen der folgenden Abschnitte.

GeldKarte am POS	GeldKarte im Internet
Kunde und Händler befinden sich räumlich an einem Ort, z.B. Bezahlen in einer Kantine	Kunde und Händler sind räumlich getrennt. Der Kunde sitzt an seinem PC und der Händler ist ein Online-Shop
Kunde und Händler können sich gegenseitig persönlich von ihrer Authentizität überzeugen.	Auf Grund der räumlichen Trennung zwischen Kunde und Händler ist eine persönliche gegenseitige Authentisierung nicht möglich.
Der Bezahlvorgang und die Auslieferung der gekauften Ware erfolgen unmittelbar.	Der Bezahlvorgang und die Auslieferung der gekauften Ware erfolgen zeitlich getrennt.
Das Terminal am POS ist ein in sich geschlossenes System. Für einen Angreifer ist es fast unmöglich, in das System einzudringen.	Das Terminal ist ein verteiltes System. Kundenterminalsystem und Händlerkomponente sind räumlich getrennt (vgl. Abschnitt 2.3).

2.2 Systemüberblick

Das Bezahlen mit einer GeldKarte am POS wird offline und ohne PIN-Prüfung an einem **Händlerterminal** (vgl. [HSys]) durchgeführt. Das Händlerterminal (bestehend aus dem **Akzeptanz-⁹**, **Kassenschnitt-¹⁰** und **Einreichungsterminal¹¹**) ist eine sichere Komponente, die in Kommunikation mit der GeldKarte tritt. Die Integrität des Transaktionsablaufs wird durch ein **Sicherheitsmodul** garantiert. Dieses Modul wird in Form einer Chipkarte, der **Händlerkarte**, realisiert. Die Händlerkarte dient dazu, die Kommunikation zur GeldKarte abzusichern und Zahlungsdatensätze zu akkumulieren, die vom Händler zur Abrechnung mit der kartenausgebenden Bank eingereicht werden können.

Alle sicherheitsrelevanten Funktionen, wie die Echtheitsprüfungen von GeldKarte und Händlerkarte sowie die Integritätssicherung der Kommunikation zwischen beiden Kompo-

⁸ Der Ort, an dem ein bestimmtes Gut oder eine bestimmte Dienstleistung verkauft wird.

⁹ Durchführung von Chipkartentransaktion, Erzeugung von Umsatzdatensätzen.

¹⁰ Erzeugung von Summensätzen über die Umsatzdaten

¹¹ Übertragung der Summensätze und Umsatzdaten an die Evidenzzentrale. Die Evidenzzentrale führt die Verrechnung zwischen Händlerkonto und Schattenkonto der GeldKarte durch.

renten werden von der GeldKarte bzw. von der Händlerkarte abgedeckt. Die hierzu erforderlichen kryptographischen Schlüssel und weitere sicherheitsrelevante Daten sind ebenfalls in diesen beiden Karten sicher gespeichert.

Das GeldKarte-System und seine Protokolle sind so konzipiert, dass durch eine kryptographische end-to-end Sicherung (Integritätsprüfung durch **Message Authentication Codes, MAC**) zwischen GeldKarte und Händlerkarte die Kommunikation zwischen diesen vollkommen transparent und ohne zusätzliche Sicherheitsmechanismen erfolgen kann. Es ist also nicht notwendig, dass sich die GeldKarte und die Händlerkarte während der Abwicklung eines Bezahlvorganges am gleichen Ort befinden. Damit eröffnet sich die prinzipielle Möglichkeit, die GeldKarte für Bezahlvorgänge im Internet einzusetzen.

Technisch wird das Bezahlen mit der GeldKarte über das Internet durch ein **verteiltes Händlersystem** umgesetzt, in welchem die GeldKarte und die Händlerkarte räumlich getrennt sind und via Internet miteinander kommunizieren. Dabei zerfällt das Händlerterminal in eine **Kundeneinheit** und eine **Händlereinheit**.

Die Funktionen Kassenschnitt- und Einreichungsterminal werden vollständig von der Händlereinheit übernommen. Die Funktion Akzeptanzterminal wird von beiden Komponenten gemeinsam wahrgenommen.

Aus dieser Sicht kommen der Komponente Kundeneinheit des verteilten Händlersystems die folgenden Aufgaben zu:

- Bereitstellung einer Schnittstelle (Bezahlsoftware und Kundenterminal) für den Kunden,
- Kommunikationsvermittlung zwischen Händlereinheit und Chipkarte,
- ggf. Datenspeicherung (im Sinne einer Protokollierung).

Das verteilte Händlersystem bestehend aus der Kundeneinheit und Händlereinheit, lässt sich wie folgt darstellen.

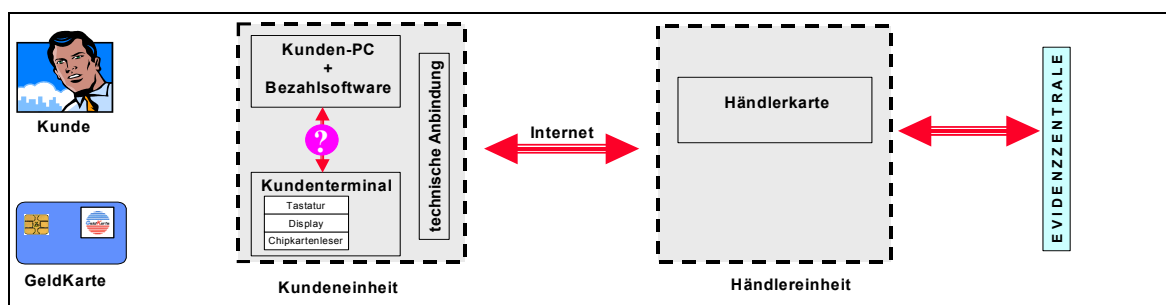


Abbildung 4: Das verteilte Händlersystem

2.3 Das Kundenterminal als sicherheitstechnisches Instrument

Die Aufteilung des Händlersystems in zwei räumlich getrennte Komponenten, die unsichere Umgebung des Kunden-PC und das unsichere Verbindungsmedium Internet, erfordern spezielle Maßnahmen zur Absicherung, um den "Kriterien für eine Bewertung von chipgestützten Zahlungssystemen" [Krit] gerecht zu werden.

In diesem Abschnitt wird erläutert, wie der Einsatz eines Kundenterminals (KT) zu einer Absicherung des Systems führt und ein der POS-Umgebung vergleichbares Sicherheitsniveau erreicht werden kann. Dabei werden insbesondere die Aufgaben des KT deutlich.

Zunächst werden die Sicherheitsrisiken dargestellt, die unter der Annahme bestehen, dass ein einfacher Chipkartenleser - ohne Display, Tastatur und interne Verarbeitungslogik - eingesetzt wird. Auf dieser Grundlage soll der benötigte Funktionsumfang des KT erläutert werden.

Die Risiken lassen sich wie folgt beschreiben:

- **Allgemeine Sicherheitsrisiken:** Eine Bezahlsoftware führt innerhalb des PC Operationen aus und kommuniziert mit der Außenwelt. Sie könnte durch Viren oder Trojaner manipuliert werden.
- **Vertraulichkeit der Daten:** Auch wenn sich eine Kartenummer nicht einer bestimmten Person zuordnen lässt, ist aufgrund der Internetdialoge in Verbindung mit weiteren Daten (z.B. IP-Adressen, URL) feststellbar, welche Beträge zwischen Instanzen transferiert werden (Für die Sicherheit des Verfahrens ist es eigentlich nicht notwendig, die Vertraulichkeit der Daten abzusichern, dies wird aber aus Datenschutzgründen¹² gefordert).
- **Integrität der Daten:** Teile der Transaktionsdaten (Händleridentität, Kundenidentität, Beträge) zwischen Händlern und Kunden sind vor Manipulationen nicht geschützt.
- **Ablaufsicherung:** Die korrekte Abwicklung der Schritte des Transaktionsprotokolls kann auf Grund fehlender Sicherheit im Kunden-PC mit einem einfachen Leser nicht sichergestellt werden

Da die Vertraulichkeit der via Internet transportierten Daten alleine aus Gründen des Datenschutzes gefordert wird und keinen Einfluss auf die Sicherheit des Verfahrens selbst hat, erreicht man mit einer Absicherung der Internetverbindung durch PC-Komponenten (z.B. SSL) ein hinreichend hohes Sicherheitsniveau. Es bleiben diejenigen Risiken, die die Sicherheit des Verfahrens direkt betreffen.

Folgende Anforderungen werden gestellt:

- Für den Kunden muss feststellbar sein, welchen Händler er bezahlt und welcher Betrag aus seiner GeldKarte abgebucht wird.

¹² Der Begriff Datenschutz umfasst alle technischen und rechtlichen Maßnahmen, die dazu dienen, das Grundrecht der informationellen Selbstbestimmung als Freiheitsrecht des Bürgers zu sichern. (s. Bundesdatenschutzgesetz, [BSDG]).

- Bei einer Internetzahlung muss der zu zahlende Betrag dem Kunden authentisch angezeigt und vom Kunden bestätigt werden.
- Der Kunde muss in der Lage sein, den Händler authentisch zu identifizieren.
- Der Transaktionsablauf muss gegen Manipulationen geschützt werden.

Dabei ist der erste Punkt für die Absicherung des Kunden von Bedeutung. Darüber hinaus würde das Verfahren insgesamt durch diesbezügliche Mängel kompromittiert. Eine Transaktion, bei welcher der Kunde nicht nachweislich Betrag und Händleridentität bestätigt hat, wird bestreitbar.

Die genannten Anforderungen werden durch den Einsatz eines Kundenterminals wie folgt erfüllt:

Ablaufsicherung

Das Kundenterminal (KT) übernimmt die sichere Ablaufsteuerung während der Kommunikation mit der GeldKarte. Das KT stellt insbesondere sicher, dass das zentrale Chipkartenkommando ABBUCHEN ausschließlich von der KT-Anwendung zum „Bezahlen mit der GeldKarte“ und nicht im Transparentmodus (von einer PC-Anwendung) an die Chipkarte geschickt wird¹³. Die Ablaufsteuerung sorgt darüber hinaus dafür, dass die Kommandos der Transaktion nur in der spezifizierten Reihenfolge und nur bei erfolgreicher Ausführung aller vorherigen Schritte ausgeführt werden können.

Prüfung der Händleridentität

Für den Nachweis der Händleridentität werden mittels asymmetrischer Kryptographie (RSA) Zertifikate verwendet. In das Zertifikat gehen der Händlername und die Händlerkartenummer ein. Während der Name durch den Kunden geprüft und bestätigt werden kann, geht die Nummer der Händlerkarte in die Transaktion ein. Das KT prüft das Zertifikat, zeigt dem Kunden den Händlernamen zur Bestätigung an und akzeptiert erst danach die Durchführung einer Transaktion, und zwar ausschließlich mit der bestätigten Händlerkarte.

Das KT muss somit RSA-Zertifikate prüfen und über entsprechende Mechanismen zur Verwaltung öffentlicher Schlüssel verfügen. Für das Anzeigen und Bestätigen des Händlernamens muss das KT über ein Display und eine Bestätigungstaste¹⁴ verfügen. Die Ab-

¹³ Das gilt ebenso für die entsprechenden Kommandos anderer Zahlungsanwendungen (Firewall-Funktionalität).

¹⁴ im Hinblick auf Multifunktionalität des KT wird zusätzlich eine numerische Tastatur gefordert.

laufsteuerung des KT muss sicherstellen, dass die Transaktion erst nach der Bestätigung des Kunden und nur mit der bestätigten Händlerkarte fortgesetzt wird.

Prüfung des Abbuchungsbetrages

Der vom Händlersystem übermittelte Abbuchungsbetrag wird durch das KT angezeigt. Die Transaktion kann erst nach Bestätigung des Kunden und ausschließlich über den angegebenen Betrag (bzw. bei inkrementellen Abbuchungen bis zur Höhe des Betrages) ausgeführt werden. Für das Anzeigen und Bestätigen des Betrages werden wiederum ein Display und eine Bestätigungstaste benötigt. Die Ablaufsicherung des KT sorgt dafür, dass die Abbuchung erst nach der Bestätigung durch den Kunden und nur über den bestätigten Betrag erfolgt.

2.4 Kundenterminalanwendungen (KTA)

Kundenterminals waren von Anfang an als multifunktionale Geräte ausgelegt. In diesem Sinne wird die Anwendung zum Bezahlen mit der GeldKarte innerhalb des KT als eine **Kundenterminalanwendung (KT-Anwendung, KTA)** neben anderen realisiert. Da die oben genannten Sicherheitsmechanismen anwendungsspezifisch sind, werden sie durch die KTA umgesetzt. Eine Ausnahme bildet dabei die Sperrung der transparenten Ausführung von zentralen Kommandos der Zahlungsanwendungen. Diese Kommandos werden durch die globale Anwendungsverwaltung des KT abgewiesen, falls sie nicht von den entsprechenden Anwendungen innerhalb des KT abgesetzt werden. Die Schnittstelle der KTA zur aufrufenden Komponente wird durch sogenannte **KT-Kommandos** gebildet. Durch den ZKA werden ausschließlich die Datenfelder der KT-Kommandos festgelegt. Die Kommandos selbst sind spezifisch für jeden KT-Hersteller. Die Art und Weise der Übermittlung an das KT ist i.A. abhängig von der verwendeten Hardware (Treiber).

2.5 GK-API

Für den Zugriff auf das KT durch die Bezahlsoftware wird eine definierte, von der Hardware des KT unabhängige Schnittstelle benötigt. Um auch eine weitgehende Unabhängigkeit von der Plattform des Kunden-PC zu erreichen, entschied man sich für die Verwendung einer **dynamischen Bibliothek** mit einem in **ANSI-C** realisierten **API (Application Program-**

ming Interface)¹⁵. Die Schicht innerhalb der dynamischen Bibliothek unterhalb des API, welche die Funktionen des Kundenterminals aufruft, wird als **Terminalanwendung (TA)** bezeichnet.

Aus Sicht der Bezahlsoftware kapselt das **GK-API** die Funktionen des KT in einer einfach zu handhabenden Schnittstelle. Für die Nutzung des KT muss die Bezahlsoftware lediglich die GK-API-Bibliothek (über ihren Namen) laden und die definierten Funktionen des API aufrufen. Die Bezahlsoftware kann damit insbesondere unabhängig von den verwendeten Protokollen und Treibern zur Kommunikation mit dem KT entwickelt werden¹⁶.

Funktionen des API

Die wesentlichen Verarbeitungsfunktionen des API werden in Aufrufe von KT-Kommandos umgesetzt. Die folgende Tabelle gibt einen Überblick:

KT-Kommandos	API-Funktionen
READ CARD DATA	<code>gk_read_card_data()</code>
READ CARD DATA SCHNELL	<code>gk_read_card_data_schnell()</code>
ABBUCHEN EINLEITEN	<code>gk_abbuchen_einleiten()</code>
ABBUCHEN EINLEITEN SCHNELL	<code>gk_abbuchen_einleiten_schnell()</code>
ABBUCHEN	<code>gk_abbuchen()</code>
ABBUCHEN_IE	<code>gk_fini()</code>
ABBUCHEN_IE SCHNELL	<code>gk_abbuchen_ie()</code>
ABBUCHEN_IW	<code>gk_abbuchen_ie_schnell()</code>
ABBUCHEN_IW SCHNELL	<code>gk_abbuchen_iw()</code>
FINI	<code>gk_abbuchen_iw_schnell()</code>
FINI_I	<code>gk_fini_i()</code>

¹⁵ Die Verwendung der Programmiersprache C bietet den Vorteil eines festgelegten Binärformates, das die Verwendung aus verschiedenen, nicht zusammen mit dem API erstellten Programmen (Compiler, ...) möglich macht. Dynamische Bibliotheken können aus Java-Applets als "Native Code" betriebssystemunabhängig geladen werden. Dabei werden die Spezifika des Betriebssystems durch das JRE (Java Runtime Environment) gekapselt.

¹⁶ Die vom KT-Treiber benutzten Protokolle zur Kommunikation mit dem KT werden in dieser Arbeit nicht behandelt. Kommunikationsverfahren wie PC/SC, CT-API, OCF sind für die Schnittstelle selbst nicht relevant und daher nicht Gegenstand dieser Arbeit.

Zusätzlich gibt es die Funktionen `gk_api_init()` und `gk_api_close()` zum Verbindungsaufbau und -abbau.

Für die Input- und Output-Parameter der GK-API-Funktionen werden C-Strukturen festgelegt. Die Bezahlsoftware ruft die vom GK-API zur Verfügung gestellten Funktionen auf und wertet die Rückgabedaten aus.

Ein Leser, eine Karte

Insbesondere ist in den Daten, die im GK-API übergeben werden, keine Adressierung eines bestimmten KT vorgesehen. Die Bezahlsoftware geht davon aus, dass für jede Transaktion durch das GK-API ein eindeutig bestimmter KT-Treiber und eine eindeutig bestimmte GeldKarte angesprochen werden. Das API wird durch den Hersteller des verwendeten KT implementiert.

3 Das neue Schnittstellenkonzept

Das alte Schnittstellenkonzept bedurfte aus einer Reihe von Gründen der Überarbeitung. Diese Gründe sollen zunächst dargestellt werden. Auf diese Grundlage wird anschließend das neue Konzept entwickelt.

Das Einsatzspektrum der ZKA-Chipkarte im Internet wurde inzwischen stark erweitert. Es sollte eine gemeinsame Schnittstelle für alle Anwendungen geben. In Abschnitt 3.1 werden die Anwendungen kurz dargestellt. Jede der Anwendungen stellt spezifische Anforderungen an diese Schnittstelle.

Neu ist insbesondere die Anforderung, verschiedene Anwendungen innerhalb **einer** Transaktion verwenden zu können. Dieser Aspekt wird in Abschnitt 3.2 betrachtet. Wie die GeldKarte zum Bezahlen eines elektronischen Fahrscheins eingesetzt werden kann, wird als Szenario dargestellt.

Neben der Erweiterung des Einsatzspektrums ist es das Ziel des neuen Konzeptes, folgende Mängel des Systems zu beseitigen:

- Nach dem alten Konzept war der Einsatz mehrerer Versionen einer API-Bibliothek nicht möglich (vgl. Abschnitt 3.3).
- Der Betrieb mehrerer Leser an einem PC war ebenfalls nicht möglich (vgl. Abschnitt 3.4).

Anschließend wird im Abschnitt 3.5 die neue übergreifende Schicht „SCAMPI“ vorgestellt.

3.1 Die neuen Anwendungen

Das erweiterte Einsatzspektrum der ZKA-Chipkarte im Internet, das insbesondere eine Folge der Zunahme der Anwendungen auf der ZKA-Chipkarte ist, führt zum Anstieg der Anforderungen speziell an die API-Schnittstelle.

- **Laden der GeldKarte im Internet:** Gegenwärtig existiert kein realisiertes Verfahren zum „Laden der GeldKarte“ im Internet. Um kontoungebundene wie kontogebundene GeldKarten im Internet laden zu können, wird zur Zeit die Konzeption als "Laden gegen andere Zahlungsmittel" angestrebt (vgl. [KT Laden]).
- **EMV:** EMV-Chipkarten können im Internet zur Authentikation des Karteninhabers eingesetzt werden. Dazu werden durch Chipkarten erzeugte Kryptogramme verwendet, die sowohl für Internet-Bezahltransaktionen geeignet sind, als auch für die Erzeugung von Transaktionsnummern (TAN) (vgl. [KT EMV]).
- **Signaturanwendung:** Die Signaturanwendung stellt insofern eine Besonderheit dar, als sie i.A. nicht durch Internetanwendungen genutzt wird, sondern durch lokal auf einem PC installierte Anwendungen. Die Signaturanwendung wird beispielsweise zur Absicherung von E-Mails oder einer Internetverbindung (SSL) genutzt. Der Zugriff auf die Anwendung erfolgt dann also durch Mail-Programme und Webbrowser (vgl. [KT SIG]).
- **Elektronische Fahrscheine:** Zur Zeit wird eine Spezifikation für den Einsatz des ZKA-Fahrscheins im Internet erstellt. Die Spezifikation beschreibt das Anlegen eines Fahrscheins auf die ZKA-Chipkarte (vgl. [KT FS]).
- **Marktplatz-Anwendungen:** Die Marktplatz-Anwendung auf ZKA-Chipkarten dient dem Speichern von Bonusberechtigungen, Gutscheinen und Ausweisen. Insbesondere wird der ZKA-Marktplatz zum Speichern des Jugendschutzmerkmals verwendet, dem auch für die Anwendung im Internet große Bedeutung beigemessen wird (vgl. [KT MPL]).

3.2 Das KT-Konzept

Ein KT kann mehrere KT-Anwendungen (KTA) unterstützen. Dazu wird eine Anwendungsübergreifende Schicht im KT benötigt, im Folgenden als **Anwendungsverwaltung** (engl. **Application Manager**) bezeichnet. Die Gestaltung dieser Schicht war nach dem alten Konzept vollständig den Leserherstellern überlassen. Das neue Konzept [KT Basis] macht einige Festlegungen zur Funktionsweise der Anwendungsverwaltung.

Jede KTA besteht aus einer Sammlung von KT-Kommandos, einer anwendungsspezifischen „Firewall“ und einem Zustandsautomaten. Darüber hinaus existiert in der Anwendungsverwaltung zusätzlich ein übergreifender Zustandsautomat, der für die Ablaufsicherung und Steuerung der voneinander abhängigen KTA zuständig ist.

Der Zustandsautomat einer Anwendung ist im neuen KT-Konzept von zentraler Bedeutung. Unterschieden werden die globalen Zustände **aktiv**, **passiv** und **wartend**. Jede KTA unter-

stützt mindestens die Zustände *aktiv* und *passiv*. Im KT kann immer nur eine Anwendung den Zustand *aktiv* haben.

Im Ausgangszustand sind alle Anwendungen im KT *passiv*. Die Anwendungsverwaltung erlaubt für eine *passive* Anwendung ausschließlich die Ausführung eines Initialisierungskommandos (anwendungsspezifisch festgelegt, z.B. INIT GK BEZAHLEN für die Anwendung zum Bezahlen mit der GeldKarte).

Nach der Initialisierung ist die KTA zunächst *aktiv*, d. h. bereit für die Verarbeitung weiterer applikationsspezifischer KT- und Transparentkommandos. Nach der Terminierung einer Anwendung (Übergang in den Zustand *passiv*) werden alle flüchtig gespeicherten Daten der KTA gelöscht.

In bestimmten Fällen kann es erforderlich sein, dass Anwendungszustand und gespeicherte Daten z. B. beim Wechsel der KTA erhalten bleiben. Hierfür wird die Zustandsbezeichnung *wartend* verwendet. Für jede KTA muss separat festgelegt werden, bei welchen Ereignissen sie in den passiven Zustand zurückkehrt bzw. in den *wartenden* Zustand übergeht.

Zur Verwaltung der *aktiven*, *passiven* und *wartenden* KTA verwendet das KT das Konzept von **Aktivierungs-IDs**. Eine Aktivierungs-ID (gelegentlich auch durch den engl. Begriff **Activation-ID** bezeichnet) ist ein vier Byte langer numerischer Wert, der als eindeutige Referenz auf eine *aktive* oder *wartende* Anwendung im KT dient.

Jedes KT-Kommando wird im KT zunächst von der Anwendungsverwaltung geprüft. Die Anwendungsverwaltung ordnet Kommandos an Hand der Aktivierungs-ID einer *aktiven* / *wartenden* Anwendung zu. Alle Kommandos, außer den Initialisierungskommandos, die der Aktivierung von Anwendungen dienen, müssen in den Kommandodaten eine Aktivierungs-ID enthalten.

KT-Anwendung zum „Bezahlen mit der GeldKarte“ (KTA-GK)

Die KTA zum „Bezahlen mit der GeldKarte“ unterstützt den Zustand *wartend* nach erfolgter Zahlung bzw. vor einem eventuell durchzuführenden Rückbuchen (Storno). Damit kann die Anwendung z.B. für das „Bezahlen eines elektronischen Fahrscheins“ eingesetzt werden. Nach erfolgter Zahlung, bei der Aktivierung der KTA-Fahrschein (KTA-FS) geht die KTA-GK in den Zustand *wartend* über. Falls das Anlegen des Fahrscheins scheitert, kann die KTA-GK reaktiviert werden und ein Rückbuchen des Zahlungsbetrags erfolgen.

Im Folgenden wird erläutert, wie die Anwendung zum Bezahlen eines elektronischen Fahrscheins eingesetzt werden kann. Elektronische Fahrscheine dienen hierbei stellvertretend als Beispiel für alle Waren oder Dienstleistungen, bei deren Erwerb ein Zugriff auf weitere Chipanwendungen neben der GeldKarte notwendig ist.

Initialisierung von KTA-GK

Zunächst wird von einem KT im Grundzustand ausgegangen, d.h. alle KTA sind 'passiv' (vgl. Abbildung 5).

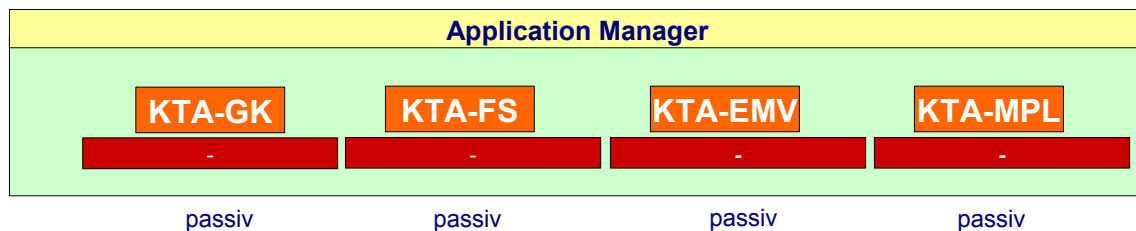


Abbildung 5: Sämtliche KTA im passiven Zustand

Die erste Funktion, die über das GK-API an das KT abgesetzt wird ist die Funktion `open()`. Sie stellt die technische Verbindung (z.B. PC/SC) zum KT her und hat keinerlei Auswirkung auf den Zustandsautomaten des KT und der KTA.

Die nächste aufgerufene Funktion des GK-API ist `init()`, die auf der KT-Ebene die KT-Funktion **INIT GK BEZAHLEN** ausführt. Der Application Manager initiiert einen Zustandswechsel der KTA-GK von *passiv* nach *aktiv*. Falls zu diesem Zeitpunkt eine andere Anwendung *aktiv* ist, wird sie entsprechend ihrer Spezifikation in den Zustand *passiv* oder *wartend* versetzt. Alle Anwendungen, die den Zustand *wartend* haben, werden gemäß ihrer Spezifikation in den Zustand *passiv* versetzt oder verbleiben im Zustand *wartend*. Bei der Aktivierung von KTA-GK erzeugt und speichert der Application Manager eine neue Aktivierungs-ID. Das Verfahren zur Erzeugung der Aktivierungs-ID muss dergestalt sein, dass eine Aktivierungs-ID (solange der Kunden-PC eingeschaltet ist) für einen möglichst großen Zeitraum – auch über einen Reset des KT hinaus – eindeutig ist. Der Einfachheit halber wird hier davon ausgegangen, dass die KTA-GK als erste aktivierte Anwendung im betrachteten Ablauf die Aktivierungs-ID „01“ bekommt.

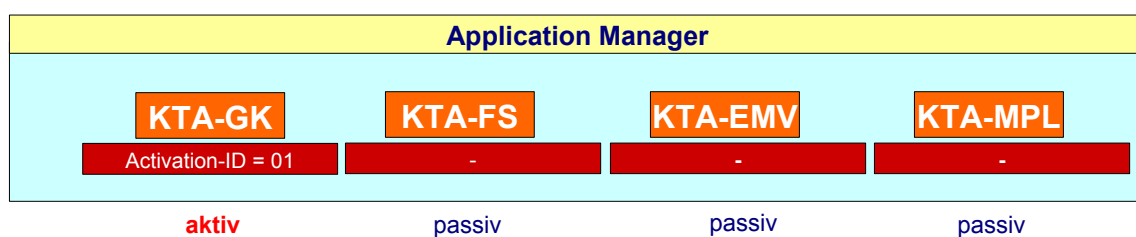


Abbildung 6: KTA-GK im aktiven Zustand mit der Activation-ID = 01

Bezahlen eines Fahrscheins

Die KT-Kommandos ABBUCHEN_EINLEITEN, ABBUCHEN und ABSCHLUSS werden gemäß ihrer Spezifikation durchgeführt. Damit ist die Zahlung erfolgt. Um den Fahrschein anzulegen, muss die KTA-FS aktiviert werden. Damit eine Stornierung der Zahlung möglich ist (falls das Anlegen des Fahrscheins scheitert), muss die Initialisierung der KTA-GK dabei erhalten bleiben. Dies wird durch den Zustand *wartend* realisiert. Die Activation-ID „01“ bleibt dabei gültig. Die KTA-FS wird aktiviert und bekommt vom Application Manager die Activation-ID „02“.

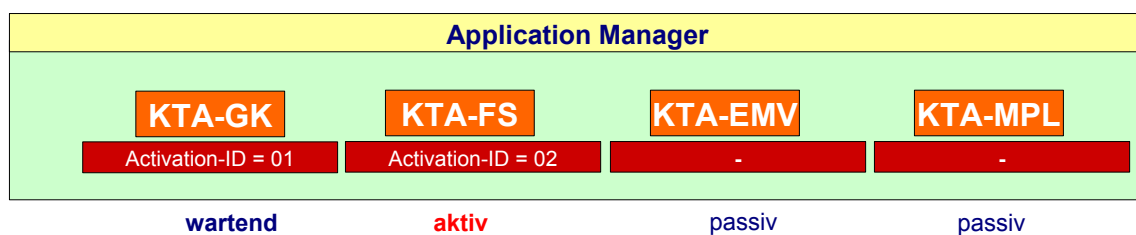


Abbildung 7: KTA-GK im Zustand 'wartend' und KTA-FS im Zustand 'aktiv'

Der Fahrschein wird gemäß der Spezifikation von KTA-FS angelegt. Falls das Anlegen scheitert und ein Rückbuchen der GeldKarte-Zahlung ausgeführt werden muss, wird die KTA-GK reaktiviert. Dies erfolgt durch erneuten Aufruf des KT-Kommandos „ABSCHLUSS“ der KTA-GK, bei dessen Ausführung nun ein Rückbuchen ausgeführt wird.

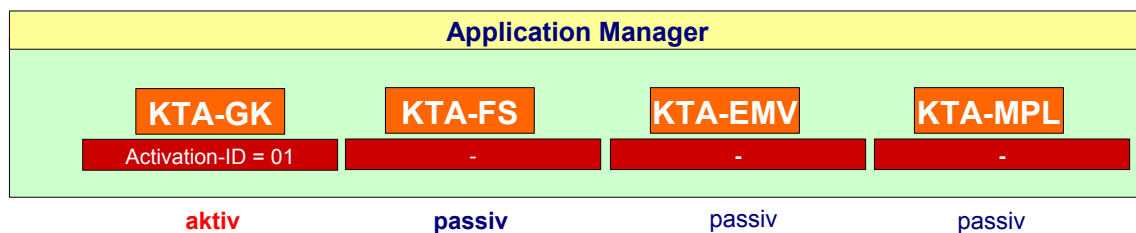


Abbildung 8: KTA-GK wieder im Zustand 'aktiv'

Mit der Ausführung von FINI GK BEZAHLEN geht die KTA-GK in den *passiven* Zustand über.

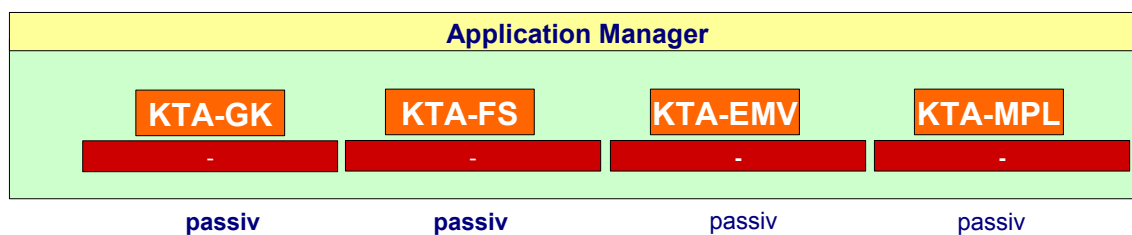


Abbildung 9: Sämtliche KTA wieder im Ruhezustand

3.3 Versionierung und Migration

Nach dem alten Konzept konnte nur eine einzige API-Bibliothek (gk-api.dll) auf dem Kunden-PC existieren. Es war weder konzeptionell noch technisch vorgesehen, mehrere Versionen des GK-API zu unterhalten. Diese Problematik ist schon bei der ersten Konzeption der Anwendung „Laden der GeldKarte im Internet“ deutlich geworden.

Da man nicht von einer fort dauernden Stabilität der Schnittstelle ausgehen kann, muss ein Mechanismus zur Migration von einer älteren auf eine neuere Schnittstellenversion vorgesehen werden. Insbesondere wird ein einheitliches Verfahren zur Nummerierung von Versionen benötigt.

Das neue Konzept erlaubt den Einsatz von mehreren Bibliotheken. Es ist also möglich, für jede Version einer TA eine eigene Bibliothek zu realisieren, z.B. gk-api210.dll oder gk-api202.dll, usw.

3.4 Mehrere Leser an einem PC

Nach dem alten Konzept kann jeweils nur ein Leser dieselbe Anwendung unterstützen, da die TA also die API-Bibliothek über den Namen der Bibliothek angesprochen wird. Bei der Installation eines zweiten Lesers werden die entsprechenden TA des ersten Lesers im Allgemeinen überschrieben.

Mit dem steigenden Funktionsumfang und einer stärkeren Diversifizierung der Lesereigenschaften wird das Szenario von mehreren installierten Lesern an einem PC wahrscheinlicher. Beispielsweise kann neben einem monofunktionalen mobilen Leser für EMV-Anwendungen (vgl. Abschnitt 3.1) ein multifunktionales KT betrieben werden, das ebenfalls EMV-Anwendungen unterstützt. Für diesen Fall wird ein Schnittstellenkonzept benötigt, das die Auswahl aus mehreren installierten Lesern ermöglicht.

Die Unterstützung mehrerer Leser kann durch Registrierung je einer TA pro Leser erreicht werden.

3.5 SCAMPI – Eine gemeinsame Schnittstelle für alle Anwendungen

Um die in den Abschnitten 3.1 bis 3.4 genannten Anforderungen zu erfüllen, entschied man sich für eine übergreifende Komponente. Diese Komponente wird nach dem Vorbild der ersten GK-API-Spezifikation ebenfalls als dynamische Bibliothek realisiert und im Folgenden auch als Wrapper-API oder **SCAMPI (Smart Card Access Module Programming Interface)** bezeichnet. Ein Wrapper-API (engl. *wrapper* = Hülle) stellt eine Schnittschicht ohne eigene Verarbeitungsfunktionalität dar, die darunter liegende Schichten verwaltet und deren Verarbeitungsfunktionen den aufrufenden Komponenten zur Verfügung stellt. Der Zugriff der PC-Anwendung auf alle TA erfolgt also über dieses gemeinsame API. Abbildung 10 zeigt die Architektur des SCAMPI-Schnittstellenkonzeptes.

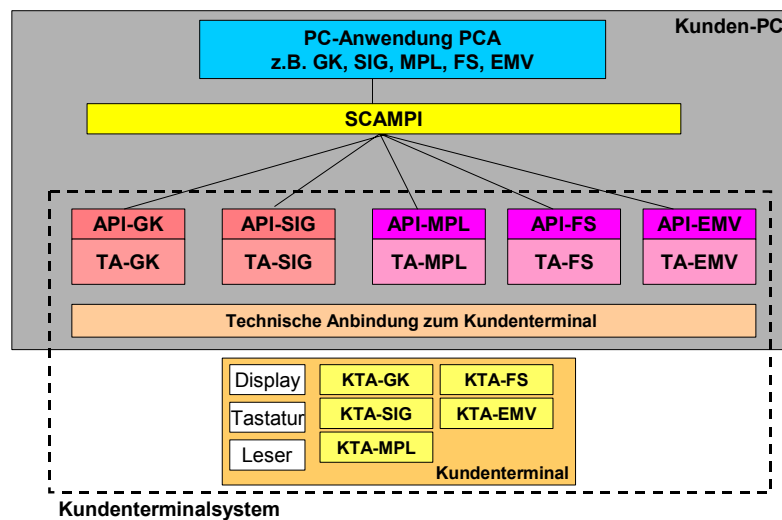


Abbildung 10: Architektur der Kundeneinheit mit übergreifender Schnittstelle

Die SCAMPI-Bibliothek muss auf dem PC des Kunden unter dem gemäß JNI-Konventionen (Java Native Interface) aus dem Namensrumpf "**scampi**" gebildeten Namen auf dem PC des Nutzers installiert sein¹⁷. Der Name ist damit Betriebssystemabhängig (etwa `scampi.dll` auf Windows- und `libscampi.so` auf Linux/Unix-Systemen). Das Laden der Bibliothek ist damit

¹⁷ Für den Zugriff aus Java-Anwendungen, enthält das Wrapper-API die C-Komponente eines Java Native Interface (JNI), das mit beliebigen Java-Laufzeitumgebungen (Version ≥ 1.1) verwendet werden kann. Die C-Komponente des JNI besteht aus C-Funktionen, deren Signatur (Funktionsname und Parameter) den JNI-Konventionen entspricht. Da der Zugriff auf alle TA über das Wrapper-API erfolgt, ist es nicht notwendig, die TA selbst mit einem Java Native Interface auszustatten.

aus einem Java-Applet betriebssystemunabhängig über den Namen möglich (Verwendung des Namens ohne Präfix und Suffix).

Damit eine TA über das SCAMPI angesprochen werden kann, muss sie zunächst im SCAMPI registriert werden. Dabei kann jeder Dienst¹⁸ ("Service") mehrfach, in mehreren Schnittstellenversionen und für mehrere Leser registriert werden. Bei der Registrierung wird dem SCAMPI mitgeteilt, welchen Dienst die TA erfüllt und welchen Namen die dynamische Bibliothek der TA hat.

Jede Transaktion wird also über dieselbe generische Schnittstelle abgewickelt. Die zur Abwicklung der Transaktion verwendeten Kommandos werden vom SCAMPI an die verwendete TA zur Ausführung weitergereicht. In Abbildung 11 wird die Funktionsweise des SCAMPI dargestellt.

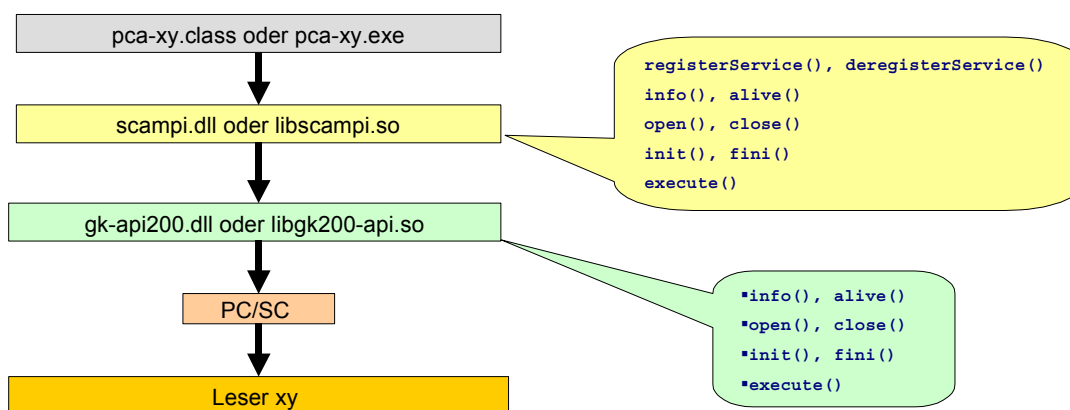


Abbildung 11: Funktionsweise des SCAMPI

Mittels der Administrationsfunktionen `registerService()` und `deregisterService()` können TA beim SCAMPI registriert bzw. de-registriert werden.

Das SCAMPI und alle TA enthalten für das Transaktionsmanagement die Funktionen `info()`, `open()`, `init()`, `fini()`, `close()` und `alive()`.

Die spezifischen Verarbeitungsfunktionen einer TA werden über die Funktion `execute()` abgewickelt. Dazu definiert jeder Dienst eigene Kommandos in Form von Bytelisten, die mittels `execute()` an die TA zur Ausführung übergeben werden. Diese Kommandos werden im Folgenden auch als **API-Kommandos** bezeichnet. Die folgende Tabelle gibt einen Überblick über die Funktionen der SCAMPI-Bibliothek.

¹⁸ Als Dienst wird der Einsatzzweck einer TA-Bibliothek bezeichnet. Die Bibliothek „gk-api200.dll“ implementiert beispielsweise den Dienst „Bezahlen mit der GeldKarte“.

SCAMPI-Funktion	Beschreibung
<code>registerService()</code>	Registriert eine neue TA-Bibliothek
<code>deregisterService()</code>	De-registriert eine TA-Bibliothek
<code>info()</code>	Gibt Informationen über die registrierten TA-Bibliotheken
<code>alive()</code>	Überprüft die Gültigkeit eines Karten-Handles
<code>open()</code>	Lädt eine TA und Etabliert die Verbindung zum einem Leser
<code>close()</code>	Entlädt eine TA und Schließt die Verbindung zum Leser
<code>init()</code>	Initialisiert eine TA bzw. KTA
<code>fini()</code>	Schließt eine Transaktion ab
<code>execute()</code>	Führt TA-spezifische Funktionen (API-Kommandos) aus

Falls mehrere Versionen eines Dienstes bzw. Implementierungen verschiedener Hersteller registriert sind, kann eine Auswahl des Dienstes über die aufrufende PC-Anwendung erfolgen. Hierzu kann die PC-Anwendung die Verwendung eines bestimmten Lesers und / oder einer bestimmten Schnittstellenversion anfordern. Für die Ermittlung der möglichen Parameter kann die Funktion `info()` verwendet werden.

Falls der Leser durch die Parameter der Funktion `open()` nicht eindeutig bestimmt ist, fordert das Wrapper-API den Nutzer über einen Dialog zur Auswahl eines Lesers auf. Der Mechanismus erlaubt es, dass die PC-Anwendung beim Aufruf von `open()` lediglich den Dienst auswählt und die Auswahl der TA allein dem Wrapper-API überlässt. Im Allgemeinen wird eine PC-Anwendung eine bestimmte Schnittstellenversion des angeforderten Dienstes erwarten und daher beim Aufruf von `open()` neben dem Dienst auch die Schnittstellenversion angeben.

Die Verwendung dieses generischen Ansatzes erlaubt

- die Registrierung von TA zur Laufzeit,
- die Definition beliebiger neuer Dienste,
- die Änderung der Kommandos eines Dienstes ohne Einfluss auf die C-Schnittstelle, dadurch
- die Unterstützung verschiedener Versionen eines Dienstes durch dasselbe Wrapper-API mittels Registrierung mehrerer Bibliotheken für einen Dienst sowie
- die Unterstützung verschiedener Leser durch Registrierung mehrerer Bibliotheken für einen Dienst.

Zustandsautomaten für Terminalanwendungen

Die oben für KTA definierten Zustände *aktiv*, *passiv* und *wartend* werden auch für TA verwendet. Eine TA hat dabei immer den Zustand, den die zugehörige KTA nach dem zuletzt ausgeführten Zugriff hatte. Der Zustand der TA muss nach jeder Ausführung eines Zugriffs auf die KTA aktualisiert werden. Der Zustand der TA spielt insbesondere dann eine Rolle, wenn die Bibliothek der TA durch das SCAMPI entladen werden soll. Eine TA darf nur im *passiven* Zustand entladen werden, da die flüchtig gespeicherten Daten (insbesondere die Aktivierungs-ID der zugehörigen KTA) der TA andernfalls für weitere Zugriffe erhalten bleiben muss.

Die Funktion `fini()` muss durch jede TA so implementiert werden, dass die TA durch Ausführung der Funktion in den Zustand *passiv* übergeht. Nach Ausführung von `fini()` darf die Bibliothek also in jedem Fall entladen werden.

4 Migrationsstrategie

In diesem Abschnitt werden Vorschläge für die mögliche Vorgehensweise zur Migration vom alten auf das neue Schnittstellenkonzept gemacht.

Von der Migration sind als Nutzer des Verfahrens zwei Parteien mit verschiedenen Interessen (vgl. Abschnitt 1.3) betroffen:

- Händlerinteressen:
 - Der Händler möchte so viele Kunden wie möglich unterstützen.
 - Auf Dauer möchte er möglichst nur eine Lösung einsetzen (geringerer Aufwand bei der Systempflege).
- Kundeninteressen:
 - Der Kunde möchte bei möglichst vielen Händlern bezahlen können.
 - Er möchte dabei den vollen Funktionsumfang des neuen Systems nutzen können, z.B. Bezahlen eines Fahrscheins¹⁹ (vgl. Abschnitt 3.2).

Die Realisierung des neuen KT-Systems auf Kundenseite und der Bezahlsoftware auf der Händlerseite erfolgt nicht unbedingt zeitgleich. Es kann sein, dass beispielsweise neue TA bestimmter KT-Hersteller bereits auf dem Markt etabliert sind, einige Händler die neue Bezahlsoftware aber noch nicht in ihre Systeme integriert haben. Genauso gut ist es möglich, dass ein KT-Hersteller die neue Software erst spät oder gar nicht zur Verfügung stellt.

Im nachfolgenden Abschnitt werden vier Modelle für eine parallele Unterstützung der alten und der neuen Schnittstelle dargestellt und bewertet. Nach der Veranschaulichung der Rollen der Beteiligten bei der Migration im Abschnitt 4.2 wird anschließend im Abschnitt 4.3 auf der Basis der im Abschnitt 4.1 unternommenen Bewertungen ein Gesamtvergleich der Modelle vorgenommen und eine Strategie für die Migration vorgeschlagen.

4.1 Die Modelle

Es müssen Strategien entwickelt werden, um neben den neuen Komponenten zumindest in der Übergangszeit weiterhin auch alte Komponenten einsetzen zu können. Die Interessen der Händler und Kunden müssen berücksichtigt werden.

Dieser Abschnitt schlägt vier Modelle für eine parallele Unterstützung der alten und der neuen Schnittstelle vor. Diese Modelle werden dargestellt, diskutiert und bewertet.

¹⁹ Das Bezahlen eines elektronischen Fahrscheins steht in diesem Abschnitt stellvertretend für die Nutzung mehrerer Chipkartenanwendungen innerhalb einer Transaktion.

4.1.1 Modell A: Parallele Installation beider API-Bibliotheken

Da die bisher verwendeten API und das SCAMPI über die Namen der implementierenden Bibliotheken angesprochen werden und es keine Überschneidungen bei der Namensgebung gibt, ist eine parallele Installation der alten und der neuen Schnittstellenbibliotheken leicht möglich. So kann ein Dienst sowohl über die alte, als auch über die neue Schnittstelle angesprochen werden.

Durch die Erhaltung des alten Systems können die bisherigen Komponenten, wie die PCA und das GK-API weiterhin benutzt werden. Das neue System wird parallel zum alten betrieben.

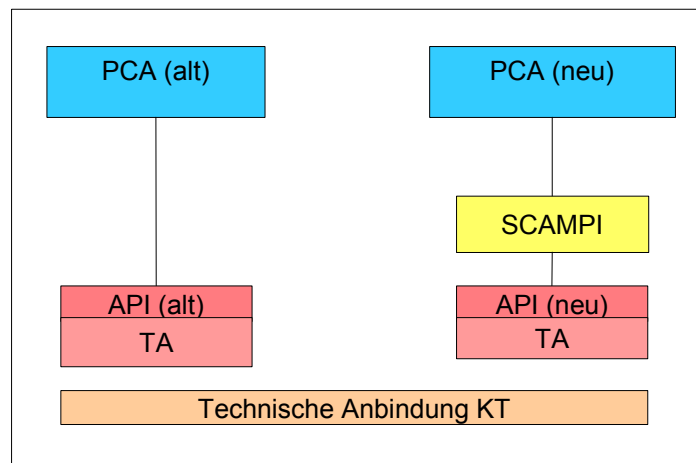


Abbildung 12: Parallele Unterstützung des alten und des neuen API

Vor- und Nachteile

Für die parallele Realisierung beider Konzepte werden sowohl die alte KTA benötigt als auch die neue. Diese Möglichkeit ist prinzipiell durch die Multifunktionalität des KT gegeben. Der volle Funktionsumfang („Bezahlen von Fahrscheinen“, Unterstützung mehrerer Leser) kann jedoch nur über das neue API genutzt werden.

Aufwand

Die Terminalhersteller müssen KT mit Unterstützung beider KTA realisieren.

4.1.2 Modell B: Altes API als Wrapper des neuen

Bei dieser Strategie greift das alte API nicht direkt auf die technische Schnittstelle zum KT zu, sondern über das neue API. Aufrufe von Funktionen des alten API müssen also in Aufrufe von Funktionen des neuen API umgesetzt werden. Ein- und Ausgabeparameter müssen entsprechend konvertiert bzw. ergänzt werden

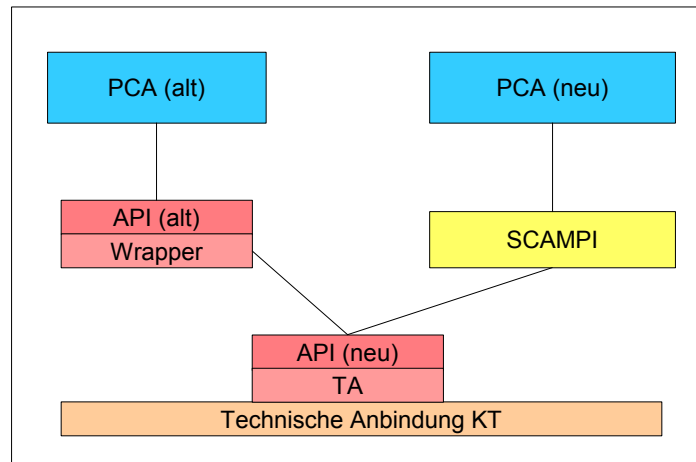


Abbildung 13: Altes API als Wrapper des neuen

Vor- und Nachteile

Das alte Konzept kann eingesetzt werden, ohne dass die alte KTA weiter benötigt wird. Das Bezahlen von Fahrscheinen ist mit dem neuen API möglich. Es ist davon auszugehen, dass eine Bezahlsoftware, die das alte API nutzt, keine Bezahltransaktion zum Anlegen eines elektronischen Fahrscheins durchführt.

Weitere Funktionen des SCAMPI-Systems, wie die Verwaltung mehrere Leser, können über die alte Schnittstelle nicht genutzt werden.

Aufwand

Die alte Schnittstelle muss als Wrapper des neuen API realisiert werden. Die Realisierung ist im Wesentlichen herstellerunabhängig mit der Einschränkung, dass der Name der zu verwendenden neuen TA-Bibliothek herstellerabhängig ist.

4.1.3 Modell C: Altes API als Wrapper des SCAMPI

In dieser Variante greift das alte API über SCAMPI auf das neue GK-API zu. Die Funktionsaufrufe des alten API werden in Funktionsaufrufe des SCAMPI umgesetzt.

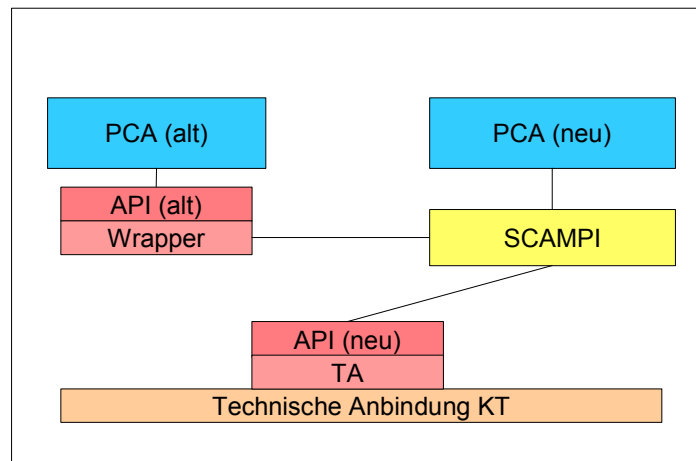


Abbildung 14: Altes API als Wrapper des SCAMPI

Vor- und Nachteile

Diese Variante hat zwei besondere Vorteile:

- Die SCAMPI-Funktionalität ist über das alte API verfügbar. Sind mehrere Leser auf dem Rechner installiert und das alte API greift auf das SCAMPI zu, so wird der Kunde durch das SCAMPI zur Auswahl des Lesers angefordert.
- Darüber hinaus kann das alte API herstellerunabhängig realisiert werden.

Aufwand

Die alte Schnittstelle muss als Wrapper des neuen API realisiert werden. Die Realisierung ist herstellerunabhängig.

4.1.4 Modell D: Neues API als Wrapper des alten

Das neue API greift (als Wrapper) über das alte API auf das KT zu. Die Funktionsaufrufe des neuen API werden in Funktionsaufrufe des alten umgesetzt.

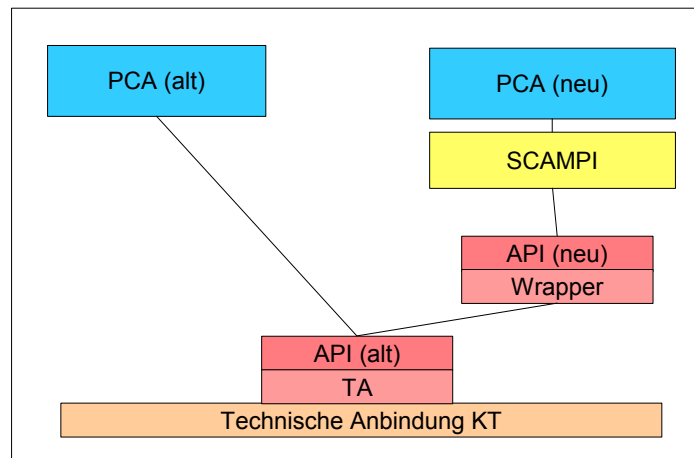


Abbildung 15: Neues API als Wrapper des alten

Vor- und Nachteile

Das alte API greift als letzte Instanz auf das KT zu und dadurch wird eine alte KTA eingesetzt, daher ist das „Bezahlen eines Fahrscheins“ (vgl. Abschnitt 3.2) nicht möglich.

Über das SCAMPI kann auf beliebig viele KT zugegriffen werden, die das alte oder das neue API unterstützen²⁰. Eine Auswahl erfolgt ggf. durch das SCAMPI-System.

Aufwand

Das neue GK-API wird als Wrapper des alten realisiert. Die Realisierung des Wrappers kann herstellerunabhängig erfolgen.

4.2 Die Rollen bei der Migration

An der technische Durchführung der Migration auf das neue System sind mehrere Seiten beteiligt: die Kundenbank als Dienstleister der Kunden, die KT-Hersteller als Zulieferer und die Händler als Betreiber der Bezahlsoftware.

Die Spezifikation des SCAMPI unterliegt der Verantwortlichkeit des ZKA. Im Rahmen der Migration, insbesondere bei der Realisierung von Systemkomponenten, tritt der ZKA nicht notwendigerweise in Erscheinung.

²⁰ Falls mehrere KT mit dem alten API registriert werden sollen, müssen die alten API-Bibliotheken unterschiedlich benannt werden (und nicht alle „gkapi.dll“)

Die KT-Hersteller sind zumindest Lieferanten für die benötigte Hardware und Hardware-abhängige Software-Komponenten. Herstellerunabhängige Komponenten können durch KT-Hersteller, Banken oder Dritte realisiert / bereitgestellt werden. Dies wird in der im Folgenden dargestellten Rollenverteilung berücksichtigt. Die Aufgaben der Beteiligten sind im Einzelnen:

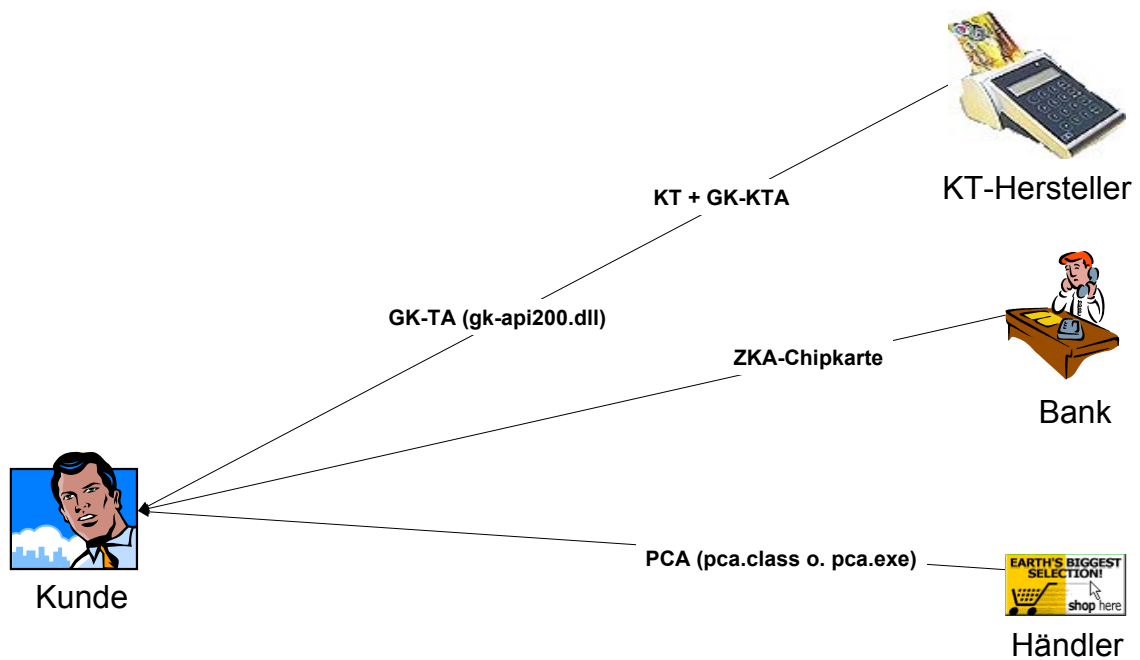


Abbildung 16: Die Rollen bei der Migration

Händler

Der Händler stellt die PC-Anwendung (PCA) bereit, dabei muss er sowohl die alte, als auch die neue PCA zur Verfügung stellen, wenn gleichzeitig alte und neue KT-Systeme unterstützt werden sollen.

KT-Hersteller

Der KT-Hersteller ist für die Realisierung des KT, der TA und der KTA zuständig. Er stellt die API-Bibliothek (z.B. „gkl-api200.dll“) zur Verfügung.

Der KT-Hersteller kann darüber hinaus

- den Kunden durch die Bank (indirekt) oder durch eigene Vertriebswege (direkt) mit KT und Software ausstatten.
- die SCAMPI-Bibliothek realisieren.
- andere herstellerunabhängige Komponenten (Wrapper-Bibliotheken) realisieren.

Bank

Die Bank gibt die ZKA-Chipkarte aus.

Die Bank kann darüberhinaus

- als Zwischenhändler den Kunden mit KT und Software ausstatten.
- die SCAMPI-Bibliothek realisieren.
- herstellerunabhängige Komponenten (Wrapper-Bibliotheken) realisieren.

Kunde

Der Kunde hat hier keine aktive Rolle. Er erhält von seiner Bank die ZKA-Chipkarte. Vom Händler wird er mit der PCA ausgestattet.

Der Kunde kann von der Bank und / oder von KT-Hersteller mit den benötigten Betriebsmitteln (KT, Software) ausgestattet werden.

4.3 Bewertung der Modelle

Alle oben beschriebenen Modelle sind geeignet, Bezahltransaktionen durchzuführen, unabhängig davon, ob die Bezahlsoftware des Händlers das alte und/oder das neue Schnittstellenkonzept unterstützt.

In der folgenden Tabelle werden die Eigenschaften der Modelle zusammenfassend dargestellt:

Merkmale	Modelle							
	A		B		C		D	
	alt	neu	alt	neu	alt	neu	alt	neu
SCAMPI	-	✓	-	✓	✓		-	✓
TA	✓	✓	W	✓	W	✓	✓	W
KT / KTA	neue und alte KTA		Nur neue KTA		nur neue KTA		nur alte KTA	
Charakteristik	Installation der alten und der neuen API-Bibliothek		Die alte PCA greift über das alte API (hier als Wrapper) auf das neue API zu		Das alte API (hier als Wrapper) greift auf das SCAMPI zu		Das neue API (hier als Wrapper) greift auf das alte API zu	
Bezahlen von FS möglich	-	✓	-	✓	-	✓	-	☹
Auswahl aus mehreren Lesern möglich	-	✓		✓	✓ F	✓	-	☹
Pro	Nutzung beider Systeme in vollem Funktionsumfang		Nutzung beider Systeme in vollem Funktionsumfang		Nutzung beider Systeme in vollem Funktionsumfang. Feature: Mit dem alten System ist dank SCAMPI eine Auswahl des Lesers möglich.		Die Nutzung des alten Systems mit Einsatz eines alten KT ist möglich	
Kontra	Die Realisierung eines multifunktionalen KT ist für den KT-Hersteller aufwändig und unwirtschaftlich		Die alte TA muss als Wrapper realisiert werden		Die alte TA muss als Wrapper realisiert werden		Die neue TA muss als Wrapper realisiert werden. Bezahlen von FS ist über das neue API nicht möglich	
Aufwand	●●●		●	-	●	-	-	●
Fazit	Die neue KTA muss realisiert werden. Die Modelle B und C zeigen, dass eine weitere parallele Nutzung der alten KTA nicht sinnvoll ist		Die alte TA wird als Wrapper eingesetzt, dadurch ist sowohl das alte als auch das neue System im vollen Umfang nutzbar (wie in Modell A)		Dieses Modell hat den gleichen Nutzwert und Aufwand wie das Modell B, jedoch mit einem Zusatznutzen: Auch über das alte API ist dank SCAMPI eine Auswahl des Lesers möglich		Das Bezahlen eines FS oder die Auswahl eines Lesers ist zwar nicht möglich, der Kunde kann aber sein altes KT mit dem neuen System nutzen	

Legende

- W** Realisierung des TA als Wrapper ohne GK-spezifische Funktionalität
F Zusatznutzen (Feature)

Bei der Bewertung der Modelle müssen zwei verschiedene Szenarien berücksichtigt werden:

1. Ein Kunde setzt ein KT ein, das die neue KTA unterstützt.
2. Ein Kunde setzt ein altes KT ein, das die neue KTA (noch) nicht unterstützt.

Modell D geht davon aus, dass das KT nur die alte KTA unterstützt. Dieses Modell ist dann sinnvoll, wenn der Kunde ein altes KT (ohne erneuerte Software) einsetzt. Die benötigten herstellerunabhängigen Ergänzungen der Softwareausstattung des Kunden können beispielsweise durch die kartenausgebende Bank bereitgestellt werden. Damit kann der Kunde das neue Verfahren auch dann nutzen, wenn der Hersteller des verwendeten KT keine aktualisierte Software bereitstellt.

Die anderen Modelle gehen davon aus, dass das KT die neue KTA unterstützt. Hierbei bietet Modell C den Zusatznutzen, dass eine Leserauswahl über das SCAMPI-System auch bei Verwendung des alten Schnittstellenkonzeptes möglich ist. Die Modelle A und B bieten gegenüber dem Modell C keinerlei Vorteile. Die dynamische Bibliothek, die das alte GK-API als Wrapper des SCAMPI implementiert, kann standardmäßig zusammen mit dem SCAMPI-System installiert werden.

Glossar

Akzeptanzterminal	<p>Das Bezahlen mit einer GeldKarte erfolgt offline ohne PIN-Prüfung an einem Akzeptanzterminal. Hierbei wird im Rahmen eines Dialoges zwischen GeldKarte und Händlerkarte ein Betrag aus der GeldKarte abgebucht, und es werden für den Händler durch die Händlerkarte zertifizierte Umsatzdaten erzeugt, die verlustfrei im Akzeptanzterminal gespeichert werden. Das Akzeptanzterminal ist eine Komponente eines Händlersystems, das in [HSys], [HSysInc] und [Errata] spezifiziert ist.</p> <p>Zum Bezahlen mit einer GeldKarte im Internet wird das Akzeptanzterminal durch Kundeneinheit und Händler-einheit eines verteilten Händlersystems realisiert.</p>
Anwendungsspezifische Daten	Daten, die durch die KT-Kommandos einer Anwendung temporär im Kundenterminal bzw. durch API-Funktionen innerhalb der Terminalanwendung gespeichert werden.
API	Application Programming Interface, festgelegte Schnittstelle einer Anwendung. Ein API besteht i.A. aus einer Sammlung von Funktionen, die durch eine dynamische Bibliothek implementiert werden.
Applet	Dynamisch aus dem Internet auf den Kunden-PC geladenes und dort ausgeführtes Programm
BER-TLV	Basic Encoding Rules – Tag Length Value, Hierarchische Organisation von Daten. Jedes Datenfeld wird mit einem vorangestellten Tag (Etikett) und Längenfeld versehen. So entstehende <i>Datenobjekte</i> können beliebig geschachtelt werden.
C	Programmiersprache mit standardisiertem Binärformat, dadurch für die Realisierung definierter Schnittstellen durch dynamische Bibliotheken geeignet und allgemein verwendet (Verwendung aus verschiedenen, nicht zusammen mit dem API erstellten Programmen (verschiedene Compiler, ...) ist möglich)
Chipkartenanwendung	Anwendung auf der Chipkarte
CA	Chipkartenanwendung

Dienst	Die Funktion / der Einsatzzweck des aus einer Kartenanwendung, einer Terminalanwendung und einer KT-Anwendung bestehenden Systems
DLL	Dynamic Link Library, Bezeichnung für eine dynamische Bibliothek
Dynamische Bibliothek	Softwarebibliothek, die zur Laufzeit von aufrufenden Programm eingebunden wird. Die für das aufrufende Programm sichtbaren Funktionen der Bibliothek bilden ein API.
Einreichungsterminal	<p>Die Komponente des in [HSys], [HSysInc] und [Errata] spezifizierten Händlersystems, die Summensätze und Umsatzdaten nach einem festgelegten Format aufbereitet und an die zuständige Evidenzzentrale überträgt, wird als Einreichungsterminal bezeichnet.</p> <p>Zum Bezahlen mit einer GeldKarte im Internet wird das Einreichungsterminal durch die Händlereinheit eines verteilten Händlersystems realisiert.</p>
Firewall	hier eine Funktion des Kundenterminals bzw. der KT-Anwendungen: das KT / die KT-Anwendungen lassen nur die Verwendung bestimmter Transparentkommandos zu
Handle	Von Softwarekomponenten verwendete logische Referenz auf ein Objekt oder Gerät. Vom SCAMPI werden Handle des Typs 'unsigned int' für Chipkarten und Chipkartenleser verwendet.
Gesamtbetrag	<p>Der im Rahmen einer Internet-Zahlung insgesamt aus einer GeldKarte abgebuchte Betrag. Bei einer (schnellen) inkrementellen Internet-Zahlung, die der Spezifikation [HSysInc] genügt, wird der Gesamtbetrag als Summe von Teilbeträgen in sukzessiven Schritten aus der GeldKarte abgebucht. Es ist zulässig, dass hierbei nur ein Schritt ausgeführt wird.</p> <p>Bei einer Internet-Zahlung, deren Ablauf der Spezifikation [HSys] genügt, wird der Gesamtbetrag in einem Schritt aus der GeldKarte abgebucht.</p> <p>Falls der Gesamtbetrag in einem Schritt aus der GeldKarte abgebucht wird, ist der Gesamtbetrag identisch</p>

	mit dem ersten und einzigen Teilbetrag und wird kurz als Betrag oder Zahlungsbetrag bezeichnet.
Inkrementelle Internet-Zahlung	Eine inkrementelle Zahlung mittels GeldKarte in einem verteilten Händlersystem, deren Ablauf den Spezifikationen [HSysInc] und [Errata] genügt. Im Rahmen einer inkrementellen Internet-Zahlung werden aus der beteiligten GeldKarte in sukzessiven Schritten Teilbeträge abgebucht, deren Summe den abgebuchten Gesamtbetrag bildet. Die Abbuchung jedes Teilbetrags erfolgt nur nach der Bestätigung durch den Kunden über die Tastatur des Kundenterminals.
Internet	System aus weltweit miteinander vernetzten Computern. In diesem Dokument steht der Begriff stellvertretend für Kommunikationswege über offene Netze.
Internet-Händlersystem	Der vom Händler betriebene Teil des verteilten Händlersystems, bestehend aus der Bezahlsoftware auf dem Kunden-PC des Kunden und der Händlereinheit. Das Internet-Händlersystem ist als Einheit zulassungspflichtig.
Internet-Zahlung	Eine Zahlung mittels GeldKarte in einem verteilten Händlersystem, deren Ablauf den Spezifikationen [HSys] und [Errata] genügt. Im Rahmen einer Internet-Zahlung wird aus der beteiligten GeldKarte in einem Schritt ein Betrag abgebucht. Die Abbuchung erfolgt nur nach der Bestätigung durch den Kunden über die Tastatur des Kundenterminals.
Java Native Interface	System für den Zugriff von Java-Programmen auf native, d.h. systemspezifische Software, wie zum Beispiel dynamische Bibliotheken mit C-Schnittstelle. Kann umgekehrt auch für den Zugriff auf Java-Komponenten aus nativer Software verwendet werden
JNI	Java Native Interface
KT-Anwendung	Sammlung von KT-Kommandos, einer anwendungsspezifischen Firewall und i.A. einem Zustandsautomaten. Jedes KT-Kommando wird in Chipkartenkommandos der entsprechenden Chipkartenanwendung und KT-interne Verarbeitungsschritte umgesetzt.
KTA	KT-Anwendung

KT-Kommando	Ein Kommando, das durch ein Kundenterminal ausgeführt werden kann.
Kunde	In dem vorliegenden Dokument gewählte Bezeichnung für den Karteninhaber der GeldKarte und Bediener der Kundeneinheit bzw. des Kundenterminals.
Kundeneinheit	Die Kundeneinheit besteht aus dem Kundenterminalsystem und dem Kunden-PC des Kunden mit der darauf befindlichen Applikationssoftware.
Kunden-PC	Der Kunden-PC ist ein System aus Hardware und Software, das unter anderem die Verbindung über das Internet mit einem Hintergrundsystem darstellt. Mögliche Ausprägungen sind ein Standard-PC oder ein Mobiltelefon.
Kundenterminal	Das Kundenterminal ist ein System aus Hardware und Software, das zur Abwicklung von Transaktionen in der lokalen Umgebung des Kunden oder über offene Netze dient. Die Software und Hardware des Kundenterminals stellen Funktionalitäten bereit, die an der Schnittstelle des Kundenterminals mittels Kommandos (KT-Kommandos) abgerufen werden. Ein Kundenterminal muss die in diesem Dokument beschriebenen Anforderungen erfüllen.
Kundenterminalsystem	Das Kundenterminalsystem ist das vom Hersteller des Kundenterminals gelieferte Gesamtsystem, bestehend aus dem Kundenterminal mit der Kundenterminalsoftware, der physischen Anbindung an oder Integration in das Kunden-PC und der eventuell auf dem Kunden-PC erforderlichen Software zur Kommunikation mit dem Kunden-PC. Das Kundenterminalsystem ist als Einheit zulassungspflichtig.
Kassenschnittterminal	Zur Einreichung der Umsatzdaten von GeldKarte-Zahlungen bei der zuständigen Evidenzzentrale werden die Umsatzdaten und ein durch die Händlerkarte zertifizierter Summensatz zu den Umsatzdaten benötigt. Eine Komponente des Händlersystems muss die Funktion zur Erzeugung des Summensatzes und zur Reinitialisierung der Händlerkarte für die Summierung neuer Umsätze (Funktion Kassenschnitt) durchführen. Diese Komponente wird als Kassenschnittterminal bezeichnet.

PC-Anwendung / PCA	Anwendung auf dem PC des Kunden, welche die Chipkartentransaktion initiiert. Dabei kann es sich um eine fest installierte oder eine dynamisch über das Internet geladene Anwendung (Applet) handeln
POS	Point Of Sales, i.A. stationäres Bezahlterminal im Einzelhandel
SCAMPI	Smart Card Access Module Programming Interface, in diesem Dokument definierte gemeinsame Schnittstelle aller Terminalanwendungen gegenüber den aufrufenden PCA
Service	Dienst
Schnelle inkrementelle Internet-Zahlung	Eine inkrementelle Zahlung mittels GeldKarte in einem verteilten Händlersystem, deren Ablauf den Spezifikationen [HSysInc] und [Errata] genügt. Im Rahmen einer schnellen inkrementellen Internet-Zahlung werden aus der beteiligten GeldKarte in sukzessiven Schritten Teilbeträge abgebucht, deren Summe den abgebuchten Gesamtbetrag bildet. Bei der schnellen inkrementellen Internet-Zahlung muss nicht jeder Teilbetrag durch den Kunden bestätigt werden. Der Kunde legt stattdessen zu Beginn der Zahlung eine Obergrenze für den Gesamtbetrag und eine Obergrenze für die Teilbeträge fest.
Teilbetrag	<p>Der in einem Schritt einer Internet-Zahlung aus einer GeldKarte abgebuchte Betrag. Bei einer (schnellen) inkrementellen Internet-Zahlung, die der Spezifikation [HSysInc] genügt, werden Teilbeträge in sukzessiven Schritten aus der GeldKarte abgebucht. Hierbei wird mindestens ein Teilbetrag abgebucht.</p> <p>Bei einer Internet-Zahlung, deren Ablauf der Spezifikation [HSys] genügt, wird ein Teilbetrag in einem Schritt aus der GeldKarte abgebucht.</p> <p>Die Summe der Teilbeträge ergibt den Gesamtbetrag einer Internet-Zahlung. Falls der Gesamtbetrag in einem Schritt aus der GeldKarte abgebucht wird, ist der Gesamtbetrag identisch mit dem ersten und einzigen Teilbetrag und wird kurz als Betrag oder Zahlungsbetrag bezeichnet.</p>

Transparentkommando	Chipkartenkommando, das von der Terminalanwendung erzeugt und vom KT transparent an die Chipkarte weitergeleitet wird
Verteiltes Händlersystem	Ein Händlersystem zur Abwicklung von Internet-Zahlungen. Es gliedert sich in eine Kunden- und eine Händlereinheit und entspricht als Gesamtsystem den Spezifikationen [HSys] und [HSysInc] mit den Ergänzungen aus [Errata].
Wrapper	eigentlich 'Hülle', in der Elektronischen Datenverarbeitung allgemein für die Kapselung einer Schnittstelle (~nfunktion) durch eine zweite verwendet
XML	Extensible Markup Language, Standard für eine allgemeine Syntax zur Beschreibung hierarchischer Daten

Quellenverzeichnis

- [BSDG] Bundesdatenschutzgesetz, Bundesbeauftragten für den Datenschutz, <http://www.bfd.bund.de/informationen/BDSG.pdf>, 28.05.2004
- [Krit] Schnittstellenspezifikation für die ec-Karte mit Chip, Kriterien für die Bewertung und Konstruktion von chipkartengestützten Zahlungssystemen, Version 3.0, 02.04.1998
- [ISO 8825-1] Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER), 1995
- [KT GK] Schnittstellenspezifikation für die ec-Karte mit Chip, GeldKarte, Internetkundenterminal, Version 3.1, 11.02.2000
- [KT API] Schnittstellenspezifikation für die ec-Karte mit Chip, GeldKarte, GK-API zum Bezahlen mit der GeldKarte im Internet, Version 1.3, 27.01.2000
- [KT FS] Schnittstellenspezifikation für die ZKA-Chipkarte, Zusatzanwendungen, ÖPV-System, Internetkundenterminal, Version 1.2, 27.01.2004
- [KT MPL] Schnittstellenspezifikation für die ZKA-Chipkarte, Zusatzanwendungen, Marktplatz-System, Internetkundenterminal, Version 0.5, 06.02.2004
- [KT EMV] API for connected EMV card readers, Generic interface for CAP and other EMV token generating readers, Version 0.5, 18.12.2003
- [KT Laden] Schnittstellenspezifikation für die ZKA-Chipkarte, GeldKarte, Generisches Ladeterminal, Internet-Kundenterminal, Version 0.4, 18.12.2003
- [KT SIG] Schnittstellenspezifikation für die ZKA-Chipkarte, Spezifikation des Internet-Kundenterminals, für die Unterstützung der Signaturanwendung der ZKA-Chipkarte, Version 1.1, 19.09.2003
- [KT Basis] Schnittstellenspezifikation für die ZKA-Chipkarte, Allgemeine Grundlagen für Kundenterminals, Version 0.89, 01.10.2003
- [HSys] Schnittstellenspezifikation für die ec-Karte mit Chip, GeldKarte, Händlersysteme, Version 3.0, 02.04.1998
- [HSysInc] Schnittstellenspezifikation für die ec-Karte mit Chip, GeldKarte, Händlersysteme, Inkrementelles Abbuchen, Version 3.0, 12.05.1999

[Errata]

Schnittstellenspezifikation für die ZKA-Chipkarte, Terminalspezifikationen, Errata, Version 1.4, 24.02.2000