13. August 2014 Studie · Digitale Identität · Mobile Sicherheit · 63 Min. Lesezeit

Alternative Zweifaktorauthentifizierung

Archiviert. Dieser Inhalt wird nicht mehr aktualisiert. Er bleibt zur Information verfügbar.

Quelle https://technology2.a-sit.at/veroeffentlichungen/alternative-zweifaktorauthentifizierung/

Zweifaktorauthentifizierung ist ein essentieller Mechanismus, der den sicheren Zugriff auf entfernte Dienste ermöglicht. Die österreichische Bürgerkarte ist nur eines von vielen Beispielen, das auf dem Grundkonzept der Zweifaktorauthentifizierung beruht. Die meisten Methoden der Zweifaktorauthentifizierung wurden für klassische Endnutzergeräte wie Desktop-PCs oder Laptops entwickelt. Chipkartenbasierte Ansätze oder auch das SMS–TAN-Verfahren sind Beispiele dafür. In den letzten Jahren konnten jedoch mobile Endnutzergeräte entscheidend an Bedeutung gewinnen. Da sich diese Geräte in Handhabung, Sicherheitsmerkmalen und Funktionalität mitunter signifikant von klassischen Endnutzergeräten unterscheiden, können etablierte Methoden der Zweifaktorauthentifizierung auf diesen Geräten nicht oder nur bedingt eingesetzt werden.

Um sich dem Thema Zweifaktorauthentifizierung unter diesem Gesichtspunkt zu nähern und mögliche Varianten der sicheren Remote-Authentifizierung unter Verwendung mobiler Endnutzergeräte zu analysieren, wurde von A-SIT eine Studie erstellt. Dazu wurden im Wesentlichen folgende Tätigkeiten durchgeführt, die sich im Inhalt der Studie widerspiegeln.

- Es wurde zunächst ein allgemeiner Überblick über die grundlegenden Konzepte hinter dem Konzept der Zweifaktorauthentifizierung erarbeitet.
- Danach wurden anhand eines abstrakten Modells generische Anforderungen an Methoden der Zweifaktorauthentifizierung auf mobilen Geräten definiert.
- Mit Hilfe der definierten Anforderungen wurden bestehende Ansätze systematisch evaluiert.
- Aus den erhaltenen Ergebnissen der Evaluierung wurde eine geeignete Lösung erarbeitet.
- Die prinzipielle Anwendbarkeit dieser Lösung wurde evaluiert, indem diese auf den konkreten Anwendungsfall serverbasierter Signaturlösungen angewendet wurde.
- Die praktische Umsetzbarkeit der Lösung wurde schließlich anhand einer prototypischen Implementierung nachgewiesen.

Insgesamt zeigt diese Studie damit, dass moderne mobile Endnutzergeräte trotz zu berücksichtigender Limitierungen Möglichkeiten bieten, alternative Methoden der Zweifaktorauthentifizierung umzusetzen. Die Studie zeigt außerdem, dass diese Methoden auch in bestehende Applikationen integriert werden können und somit deren Sicherheit auch im Falle eines Zugriffs über mobile Endnutzergeräte gewährleistet werden kann.

1. Einleitung

Der anhaltende Trend, Dienste über das Internet zur Verfügung zu stellen und zu nutzen, macht die Verfügbarkeit sicherer und verlässlicher Methoden zur Remote-Authentifizierung von Benutzerinnen und Benutzern notwendig. Da einfache Authentifizierungsmethoden basierend auf Benutzername und Passwort generell als unsicher angesehen werden müssen, haben sich für sicherheitskritische Anwendungen in den letzten Jahrzehnten mehrfaktorbasierte Authentifizierungsmethoden durchgesetzt. Diese kombinieren mindestens zwei der Authentifizierungsfaktoren Wissen, Besitz und Eigenschaft und bieten insgesamt ein wesentlich höheres Sicherheitsniveau als Methoden, die lediglich auf einem dieser Faktoren beruhen. Prominente Anwendungsbeispiele aus dem täglichen Leben, die auf Mehrfaktorauthentifizierung beruhen, sind die österreichische Bürgerkarte [1] oder auch SMS-TAN-Verfahren diverser E-Banking-Lösungen [2]. Insgesamt kommen mehrfaktorbasierte Authentifizierungsmethoden vor allem dort zum Einsatz, wo neben einer herkömmlichen Authentifizierung der Benutzerin oder des Benutzers auch die Autorisierung einer Transaktion notwendig ist. Beispiele hierfür sind etwa E-Banking-Lösungen oder auch serverbasierte Signaturlösungen.

Bestehende und etablierte Authentifizierungsmethoden wurden zumeist für klassische Endnutzergeräte entworfen und entwickelt. Dazu gehören beispielsweise Desktop-PCs oder Laptops. Ein Paradebeispiel für eine derartige Authentifizierungsmethode, die speziell auf die Eigenschaften klassischer Endnutzergeräte zugeschnitten wurde, ist das SMS-TAN-Verfahren. Bei diesem wird ein Einmalpasswort – eine sogenannte Transaktionsnummer (TAN) – per SMS an das Mobiltelefon der Benutzerin bzw. des Benutzers gesendet. Auf diese Weise wird der Besitz des Mobiltelefons verifiziert. Neben dem Mobiltelefon ist die Verwendung eines zweiten Endnutzergeräts vorgesehen, über das weitere Zugangsdaten und unter anderem auch die empfangene TAN – meist über einen Web-Browser – eingegeben werden müssen. Die Sicherheit dieses Systems beruht unter anderem auf der Verwendung zweier getrennter Endnutzergeräte.

In den letzten Jahren wurden klassische Endnutzergeräte zunehmend von mobilen Geräten wie Smartphones oder Tablet-Computer abgelöst. Diese unterscheiden sich von klassischen Endnutzergeräten jedoch in vielerlei Hinsicht. Dies bringt einerseits neue Möglichkeiten in der Nutzung von Diensten, birgt mitunter aber auch neue Gefahren. Auch hier kann das SMS-TAN-Verfahren als plakatives Beispiel dienen. Durch den stets wachsenden Funktionsumfang moderner Smartphones sind diese theoretisch in der Lage, das SMS-TAN-Verfahren, das ursprünglich für zwei getrennte Endnutzergeräte entworfen wurde, auf ein und demselben Gerät zu implementieren. Dies wird dadurch möglich, da Smartphones sowohl über den dafür nötigen Web-Browser verfügen und darüber hinaus in der Lage sind, SMS-Nachrichten und damit TANs zu empfangen. Durch die Zusammenführung der Funktionalität auf einem einzelnen Endnutzergerät wird jedoch ein zentrales Sicherheitsmerkmal dieses Verfahrens umgangen. Zusätzlich ergeben sich auf Smartphones durch Schadsoftware potentiell zusätzliche Bedrohungen für die Sicherheit von über SMS übertragenen TANs. Im Falle der ausschließlichen Verwendung eines Smartphones für den Zugriff auf eine sicherheitskritische Applikation ist das SMS-TAN-Verfahren in seiner derzeitigen Form aus

sicherheitstechnischer Sicht daher nicht anwendbar. Gleichzeitig bieten moderne mobile Endnutzergeräte jedoch eine Vielzahl neuer Technologien, die alternative Möglichkeiten der sicheren mehrfaktorbasierten Authentifizierung ermöglichen.

Diese Studie hat zum Ziel, Möglichkeiten der mehrfaktorbasierten Benutzerauthentifizierung auf mobilen Endnutzergeräten systematisch zu analysieren und eine geeignete Lösung zu erarbeiten. Diese Lösung soll eine sichere mehrfaktorbasierte Authentifizierung an einer zentralen Applikation ermöglichen. Grundannahme ist dabei, dass ausschließlich ein einzelnes mobiles Endnutzergerät für den Zugriff auf diese zentrale Applikation verwendet wird und dennoch ein entsprechendes Sicherheitsniveau erreicht wird. Zur systematischen Erarbeitung einer geeigneten Lösung werden anhand eines abstrakten Modells zunächst generische Anforderungen definiert. Diese Anforderungen werden in weiterer Folge verwendet, um bestehende Ansätze zur Umsetzung von Mehrfaktorauthentifizierung auf mobilen Endgeräten zu evaluieren. Basierend auf den erhaltenen Evaluierungsergebnissen wird schließlich eine geeignete Lösung erarbeitet.

Die erarbeitete Lösung wird bewusst abstrakt gehalten und lediglich als allgemeines Modell definiert. Dadurch wird gewährleistet, dass diese Lösung auf verschiedene konkrete Anwendungsbereiche anwendbar ist. Die Anwendbarkeit der erarbeiteten Lösung wird über zwei aufeinanderfolgende Schritte evaluiert. In einem ersten Schritt wird die abstrakte Lösung auf ein konkretes Anwendungsszenario – eine serverbasierte Signaturlösung – angewendet. Im darauffolgenden zweiten Schritt wird die Lösung zudem anhand einer prototypischen Implementierung in Bezug auf ihre Umsetzbarkeit evaluiert.

2. Grundlagen der Mehrfaktorauthentifizierung

Herkömmliche Authentifizierungsmethoden beruhen zumeist auf dem Authentifizierungsfaktor Wissen. Um sich erfolgreich zu authentifizieren, muss die Benutzerin bzw. der Benutzer nachweisen, ein gemeinsames Geheimnis zu kennen. In der Regel handelt es sich dabei um ein alphanumerisches Passwort oder um eine numerische PIN. Passwortbasierte Authentifizierungsmethoden wurden in den letzten Jahrzehnten in zahlreiche Anwendungen integriert und finden sich heutzutage in vielen webbasierten Internetdiensten oder auch bei der Benutzerauthentifizierung von Betriebssystemen.

Alphanumerische Passwörter erfreuen sich großer Beliebtheit, weisen jedoch auch einige Nachteile auf. Als größtes Problem erweist sich dabei meist der Trade-Off zwischen Sicherheit und Benutzerfreundlichkeit. Einfache Passwörter können von Benutzerinnen und Benutzern in der Regel einfach gemerkt werden, gleichzeitig jedoch auch von Angreiferinnen und Angreifern ohne großen Aufwand erraten werden. Komplexe Passwörter bieten einen besseren Schutz gegen Angriffe, sind jedoch auch schwieriger zu merken. Dies verleitet Benutzerinnen und Benutzer dazu, sich komplexe Passwörter zu notieren, was wiederum ein Sicherheitsrisiko darstellt.

Im Bewusstsein der zahlreichen Schwächen alphanumerischer Passwörter wurden in den letzten Jahren einige Alternativen entwickelt, um Authentifizierungsmethoden basierend auf dem Authentifizierungsfaktor Wissen zu ermöglichen. Vor allem im wissenschaftlichen Bereich konnten Authentifizierungsmethoden basierend auf graphischen Passwörter hier in den letzten Jahren mögliche Alternativen aufzeigen. Diese Methoden beruhen prinzipiell auf der Idee, dass sich Menschen Bilder und in Bildern kodierte Informationen leichter merken als alphanumerische Zeichenfolgen. Ein früher Überblick über Authentifizierungsmethoden, die auf graphischen Passwörtern beruhen, wurde bereits im Jahr 2005 von Suo et al. [3] gegeben. Seither

wurden zahlreiche weitere Methoden vorgestellt. Beispiele sind Authentifizierungsmethoden, die von Almulhem [4] oder von Khan et al. [5] vorgestellt wurden. In der Praxis konnten sich graphische Passwörter bisher jedoch nur sehr begrenzt durchsetzen.

In Ermangelung einer geeigneten sichereren Alternative zu alphanumerischen Passwörtern, verfolgen sicherheitskritische Applikationen zumeist einen anderen Ansatz, um die Sicherheit von Authentifizierungsmethoden zu erhöhen. Anstatt den Authentifizierungsfaktor Wissen zu ersetzen bzw. diesen anders als durch alphanumerische Passwörter abzudecken, wird der Faktor Wissen mit einem zweiten Faktor kombiniert. Dadurch ergibt sich eine sogenannte Mehrfaktorauthentifizierung. Um einen erfolgreichen Angriff auf eine derartige Authentifizierungsmethode durchzuführen, muss die Angreiferin bzw. der Angreifer mehrere Authentifizierungsfaktoren kompromittieren. Das richtige Erraten eines Passworts ist nicht mehr ausreichend für eine erfolgreiche Attacke. Mehrfaktorauthentifizierung bietet daher in der Regel ein höheres Maß an Sicherheit.

2.1. Authentifizierungsfaktoren

Prinzipiell kommen drei Faktoren in Frage, die im Zuge einer Authentifizierung kombiniert werden können. Die Wahl der Faktoren, die zur Anwendung kommen sollen, obliegt der jeweiligen Authentifizierungsmethode. In den folgenden Unterabschnitten werden die drei möglichen Faktoren beschrieben und deren Vor- und Nachteile herausgearbeitet.

2.1.1. Faktor Wissen

Der Faktor Wissen wurde bereits in der Einleitung zu diesem Abschnitt näher erläutert. Hierbei handelt es sich um ein Geheimnis, das die Benutzerin bzw. der Benutzer kennt. Wie erwähnt, wird dieser Authentifizierungsfaktor zumeist über geheime alphanumerische Passwörter, in seltenen Fällen auch über graphische Passwörter abgedeckt.

Der größte Vorteil des Faktors Wissen ist dessen einfache Umsetzbarkeit. Passwortbasierte Authentifizierungsmethoden können einfach implementiert und in beliebe Anwendungen integriert werden. Dies hat auch zur anhaltenden Popularität passwortbasierter Authentifizierungsmethoden beigetragen, welche sich in der Vielzahl an Anwendungen widerspiegelt, die auf alphanumerische Passwörter vertrauen.

Der größte Nachteil des Authentifizierungsfaktors Wissen ist der bekannte und bereits kurz erwähnte Trade-Off zwischen Sicherheit und Benutzerfreundlichkeit. Während die Sicherheit mit einer steigenden Komplexität von Passwörtern zunimmt, reduzieren komplexe Passwörter die Benutzerfreundlichkeit erheblich. Die Praxis zeigt, dass Benutzerinnen und Benutzer daher dazu tendieren, einfache Passwörter zu verwenden, bzw. ein und dasselbe Passwort für die Authentifizierung an verschiedenen Diensten zu verwenden. Während die Wahl zu einfacher Passwörter durch die Definition entsprechender Passwort-Policies verhindert werden kann, ist der Wiederverwendung von Passwörtern an verschiedenen Diensten technisch kaum beizukommen. Jedoch birgt auch die Definition zu strikter Passwort-Policies Risiken, da Benutzerinnen und Benutzer dazu tendieren, komplexe Passwörter an potentiell unsicheren Orten zu notieren.

Insgesamt kann festgehalten werden, dass der Authentifizierungsfaktor Wissen im Allgemeinen und alphanumerische Passwörter im Speziellen keinen ausreichenden Schutz für eine Authentifizierung an sicherheitskritischen Anwendungen ermöglichen. Hauptgrund dafür ist der Trade-Off zwischen Sicherheit

und Benutzerfreundlichkeit, der sich durch die Notwendigkeit komplexer Passwörter ergibt.

2.1.2. Faktor Besitz

Der Authentifizierungsfaktor Besitz wird durch eine Hardware-Komponente abgedeckt. Besitzbasierte Authentifizierungsmethoden kommen unter anderem bei Zutrittsschutzmechanismen zur Anwendung. Beispielsweise ermöglichen in vielen Fällen Mitarbeiterkarten den Zutritt zu Gebäuden und Räumlichkeiten. In diesen Fällen kann jede bzw. jeder, die bzw. der im Besitz der Mitarbeiterkarte ist, Zutritt zu geschützten Bereichen erlangen.

Der Faktor Besitz alleine stellt daher in der Regel ebenfalls keinen ausreichenden Schutz für sicherheitskritische Systeme dar, da benötigte Hardware-Komponenten theoretisch entwendet werden können. In der Regel kommt der Authentifizierungsfaktor Besitz daher meist kombiniert mit dem Faktor Wissen zur Anwendung. Ein Beispiel dafür ist die Bankomatkarte. Um mit dieser beispielsweise Bargeld zu beheben, muss die Kundin bzw. der Kunde im Besitz der Karte sein. Zusätzlich sind die Funktionen der Karte über eine geheime PIN geschützt, die nur der Besitzerin bzw. dem Besitzer der Karte bekannt ist. Das Wissen um diese PIN muss im Zuge der Verwendung der Bankomatkarte nachgewiesen werden. Damit sind zur Verwendung der Bankomatkarte sowohl der Authentifizierungsfaktor Besitz, als auch der Faktor Wissen notwendig. Die Bankomatkarte ist damit ein Paradebeispiel für die Umsetzung einer Mehrfaktorauthentifizierung.

Unabhängig davon, ob der Faktor Besitz alleine oder in Kombination mit dem Faktor Wissen verwendet wird, ergeben sich für Authentifizierungsverfahren, die auf diesem Faktor beruhen, einige Herausforderungen. So muss die Benutzerin bzw. der Benutzer in jedem Fall über eine geeignete Hardware-Komponente verfügen, die den Faktor Besitz entsprechend abdeckt. Je nach Ausformung dieser Komponente können zudem zusätzliche Hard- und Software-Komponenten notwendig sein. Beispielsweise bedingt die Verwendung von Chipkarten geeignete Chipkartenlesegeräte, über die auf die Chipkarte und deren Funktionalität zugegriffen werden kann. Die Notwendigkeit derartiger Geräte kann eine Hürde für Benutzerinnen und Benutzer darstellen. Insgesamt kann festgehalten werden, dass Authentifizierungsmethoden, die auf dem Faktor Besitz beruhen, schwieriger in Applikationen zu integrieren sind als etwa rein passwortbasierte Methoden.

2.1.3. Faktor Eigenschaft

Als Alternative bzw. als Ergänzung zu den Authentifizierungsfaktoren Wissen und Besitz bietet sich der Faktor Eigenschaft an. Darunter versteht man eine eindeutige Eigenschaft einer Benutzerin oder eines Benutzer, die diese bzw. diesen eindeutig identifiziert. In den meisten Fällen wird der Authentifizierungsfaktor Eigenschaft durch biometrische Verfahren abgedeckt. Beispiele sind Fingerabdrucksensoren oder Iris-Scanner, die den Zutritt zu Gebäuden und Räumlichkeiten oder den Zugriff auf elektronische Geräte erlauben. Biometrische Verfahren haben unter anderem durch deren Integration in Smartphones in letzter Zeit einen neuen Aufschwung erfahren. Beispielsweise ermöglichen aktuelle Versionen des mobilen Betriebssystems Google Android die Verwendung von Gesichtserkennung als Alternative zu passwortbasierten Zugriffsschutzmechanismen [6]. Auch Apple iOS bietet durch Integration eines Fingerabdrucksensors Unterstützung für biometrische Zugriffsschutzmechanismen [7]. Die Sicherheit dieser Mechanismen ist freilich beschränkt, wie durch aktuelle Analysen gezeigt wurde [8][9].

Zu potentiellen Schwächen konkreter Implementierungen müssen für den Authentifizierungsfaktor Eigenschaft im Generellen und für biometrische Ansätze im Speziellen noch weitere Aspekte in Betracht gezogen werden. So ist zu beachten, dass Eigenschaften, die für Authentifizierungsmethoden herangezogen werden, oft nur schwer geheim zu halten sind. Beispielsweise ist es nahezu unmöglich, im täglichen Leben die eigenen Fingerabdrücke geheim zu halten, da diese bei jeder Berührung mit glatten Oberflächen hinterlassen werden. Ein weiterer konzeptioneller Nachteil des Authentifizierungsfaktors Eigenschaft ist dessen Nicht-Widerrufbarkeit. Während zum Beispiel kompromittierte Passwörter einfach widerrufen und durch neue sichere Passwörter ersetzt werden können, ist dies bei biometrischen Merkmalen in der Regel nicht möglich.

Möglichkeiten und Nachteile biometrischer Verfahren wurden unter anderem von Bhattacharyya et al. [10] diskutiert. Insgesamt kann festgehalten werden, dass biometrische Verfahren eine interessante Alternative darstellen, in der Praxis deren Anwendung jedoch oft schwierig und mit zahlreichen Nachteilen verbunden ist.

2.2. Mehrfaktorauthentifizierung auf mobilen Geräten

Mobile Endnutzergeräte haben in den letzten Jahren zunehmend an Popularität gewonnen. Alleine im Jahr 2013 wurden weltweit 1,8 Milliarden Mobiltelefone verkauft [11]. Ausgelöst wurde dieser Trend vor allem durch die Einführung von Smartphones im Jahr 2007. Während klassische Mobiltelefone in Bezug auf deren Funktionalität relativ beschränkt waren, boten bereits Smartphones der ersten Generationen vielerlei Funktionen und Features. Vor allem deren App-basierte Softwareverwaltung, die eine einfache Erweiterung der Funktionalität erlaubte, unterschied diese Smartphones von ihren Vorgängern. Heute lösen Smartphones und verwandte mobile Endnutzergeräte nicht nur klassische Mobiltelefone, sondern auch andere klassische Endnutzergeräte wie Desktop-PCs oder Laptops zunehmend ab. Vor allem für die reine Konsumierung von Inhalten und für einfache Tätigkeiten wie das Verfassen und Lesen von E-Mails kommen zunehmend mobile Endnutzergeräte zum Einsatz.

Ihr verbreiteter Einsatz in verschiedenen Anwendungsszenarien macht auch die Implementierung sicherer Authentifizierungsmethoden auf mobilen Geräten notwendig. Nur wenn derartige Methoden auf diesen Geräten verfügbar und anwendbar sind, können mobile Endnutzergeräte für den Zugriff auf sicherheitskritische Daten verwendet werden. Aus der in Abschnitt 2.1 gegebenen Beschreibung möglicher Authentifizierungsfaktoren wird bereits klar, dass eine Verwendung biometrischer Ansätze auf aktuellen mobilen Endnutzergeräten kaum in Frage kommt. Derzeit auf diesen Geräten verfügbare Technologien sind für einen Einsatz in sicherheitskritischen Anwendungen noch nicht geeignet. Bei der Betrachtung möglicher mehrfaktorbasierter Authentifizierungsmethoden für mobile Endnutzergeräte wird der Fokus daher zunächst auf die Faktoren Wissen und Besitz gelegt. Damit werden auch die erwähnten konzeptionellen – von der jeweiligen Umsetzung unabhängigen – Nachteile biometrischer Verfahren umgangen.

Da der Faktor Eigenschaft nicht näher betrachtet wird, muss für die Entwicklung geeigneter mehrfaktorbasierter Authentifizierungsmethoden der Fokus auf die beiden Faktoren Wissen und Besitz gelegt werden. Interessanterweise wurden mobile Endgeräte früh in die Umsetzung mehrfaktorbasierter Authentifizierungsmethoden miteinbezogen. Konkret wurden diese Geräte verwendet, um den Faktor Besitz abzudecken. Als Beispiel kann hier das SMS-TAN-Verfahren genannt werden, welches unter anderem bei verschiedenen E-Banking-Lösungen oder auch bei der Österreichischen Handy-Signatur zur Anwendung kommt. Bei diesem Verfahren wird davon ausgegangen, dass die Benutzerin bzw. der Benutzer

zwei getrennte Endnutzergeräte verwendet. In der Regel ist dies ein Desktop-PC oder Laptop und ein Mobiltelefon. Beide Geräte sind notwendig, um sich bei einem entfernten Service zu authentifizieren. Dazu werden über den Web-Browser am klassischen Endnutzergerät zunächst Telefonnummer und Passwort eingegeben und an das entfernte Service gesendet. Dadurch wird der Authentifizierungsfaktor Wissen abgedeckt. Anschließend wird ein Einmalpasswort (TAN) per SMS an das Mobiltelefon geschickt. Das erhaltene Einmalpasswort muss von der Benutzerin bzw. vom Benutzer wiederum über den Web-Browser am klassischen Endnutzergerät eingegeben und an das zentrale Service gesendet werden. Durch Eingabe des Einmalpassworts weist die Benutzerin bzw. der Benutzer nach, im Besitz des Mobiltelefons zu sein. Damit wird über dieses Einmalpasswort der Authentifizierungsfaktor Besitz abgedeckt.

Da das SMS-TAN-Verfahren von einer Verwendung zweier getrennter Endnutzergeräte ausgeht, ist dessen Verwendung auf einem einzigen mobilen Gerät nicht vorgesehen. Obwohl ein modernes Smartphones in der Lage wäre, die Aufgaben beider Endnutzergeräte – d.h. Bereitstellen eines Web-Browsers und Empfang einer SMS – zu implementieren, widerspricht die Realisierung beider Komponenten auf einem einzigen Gerät dem zugrundeliegenden Sicherheitskonzept des SMS-TAN-Verfahrens.

Unglücklicherweise sind auch andere Varianten den Faktor Besitz auf einem Smartphone oder verwandtem mobilen Endnutzergerät zu implementieren begrenzt. So ist die Verwendung von Chipkarten, die auf herkömmlichen Endnutzergeräten häufig zur Implementierung dieses Faktors zur Anwendung kommen, schwierig. Mit Produkten wie dem iAuthenticate [TM] Smart Card Reader [12] für das Apple iPhone existieren zwar Lösungen, die eine Verwendung von Chipkarten auf Smartphones prinzipiell erlauben, diese haben jedoch bisher kaum Verbreitung gefunden und zielen vor allem auf einen Nischenmarkt ab. Eine andere Variante den Faktor Besitz auf einem Smartphone abzudecken ergibt sich durch die SIM (Subscriber Identity Module). Da ein Zugriff auf die SIM jedoch in den meisten Fällen nur über den jeweiligen Mobilfunkanbieter möglich ist, ergeben sich für SIMbasierte Authentifizierungsmethoden zusätzliche Einschränkungen.

Zusammenfassend kann festgehalten werden, dass die Implementierung des Authentifizierungsfaktors Besitz auf mobilen Endnutzergeräten eine bedeutende Herausforderung darstellt. Zusätzliche Bedeutung kommt der geeigneten Umsetzung dieses Faktors zu, da aufgrund der Schwächen biometrischer Methoden der Faktor Besitz neben dem Faktor Wissen die einzig verbleibende Alternative ist. Während existierende auf mobile Endnutzergeräte beruhende Authentifizierungsmethoden wie das SMS-TAN-Verfahren nicht ohne weiteres für eine ausschließliche Verwendung auf mobilen Endnutzergeräten geeignet sind, bieten moderne Smartphones eine Reihe von Features und Funktionen, die die Entwicklung alternativer Authentifizierungsmethoden erlauben. Anforderungen an diese Methoden werden im folgenden Abschnitt systematisch erarbeitet.

3. Anforderungen

Die Umsetzung von Mehrfaktorauthentifizierung auf mobilen Endnutzergeräten bedingt die Integration der Authentifizierungsfaktoren Wissen und Besitz. Der Faktor Eigenschaft und damit die Implementierung biometrischer Methoden wird hier aus den genannten Gründen nicht näher betrachtet. Der Authentifizierungsfaktor Wissen lässt sich auch auf mobilen Endnutzergeräten relativ einfach umsetzen. Hierfür können numerische, alphanumerische oder auch graphische Ansätze verfolgt werden. Aus Umsetzungssicht stellt die geeignete Implementierung des Authentifizierungsfaktors Wissen in jedem Fall keine bedeutende Hürde dar.

Als herausfordernder erweist sich die Implementierung des Authentifizierungsfaktors Besitz. Um diesen Faktor geeignet abzudecken, müssen von möglichen Umsetzungen diverse Anforderungen erfüllt werden. Um diese systematisch zu erarbeiten, wurde ein abstraktes Modell für Implementierungen des Authentifizierungsfaktors Besitz entwickelt. Dieses ist in Abbildung 1 dargestellt. Das abstrakte Modell wurde unter folgenden Annahmen entwickelt:

- Eine serverbasierte zentrale Applikation muss für die Durchführung einer sicherheitskritischen Transaktion eine entfernte Benutzerin bzw. einen entfernten Benutzer sicher und verlässlich authentifizieren. Durch die erfolgreiche Authentifizierung der Benutzerin bzw. des Benutzers wird die Transaktion autorisiert.
- Zur Authentifizierung der Benutzerin bzw. des Benutzers vertraut die zentrale Applikation auf die Authentifizierungsfaktoren Wissen und Besitz. Nur der Faktor Besitz wird durch das Modell abgebildet. Der zweite notwendige Faktor Wissen ist in der Regel trivial zu implementieren und wird durch das Modell daher nicht abgebildet.
- Die Benutzerin bzw. der Benutzer greift auf die Applikation ausschließlich über ein einzelnes mobiles Endnutzergerät zu.

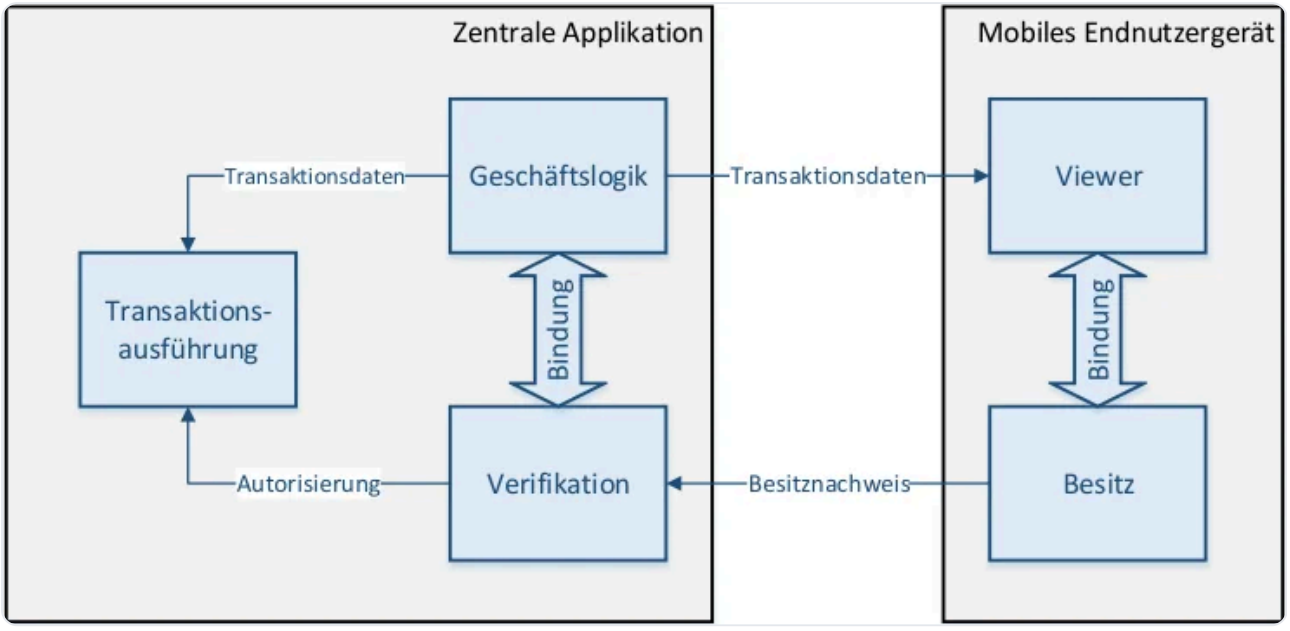


Abbildung 1. Abstraktes Modell zur Implementierung des Authentifizierungsfaktors Besitz. Für die Zentrale Applikation selbst wurden drei, für das Mobile Endnutzergerät zwei Subkomponenten definiert. Die Subkomponente Geschäftslogik der Zentralen Applikation implementiert im Wesentlichen deren Funktionalität mit Ausnahme der sicherheitskritischen Transaktion, die in eine eigene zentrale Subkomponente namens Transaktionsausführung ausgelagert wurde. Die Geschäftslogik definiert unter anderem auch die Transaktionsdaten, die im Zuge der Transaktion bearbeitet werden sollen. Diese Daten werden einerseits an die für die Transaktionsausführung verantwortliche zentrale Komponente, als auch an das Mobile Endnutzergerät gesendet.

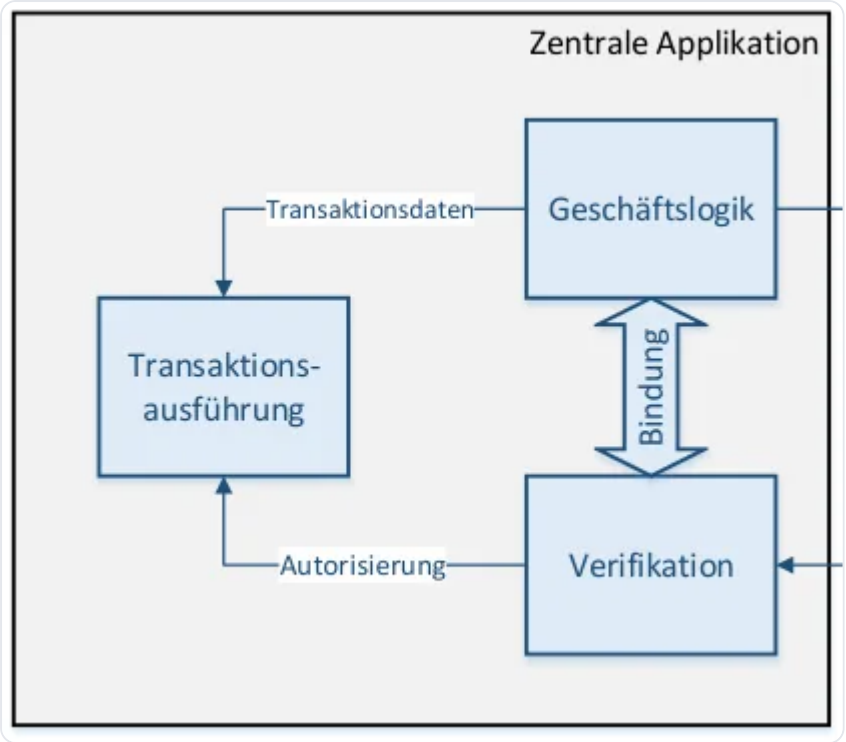
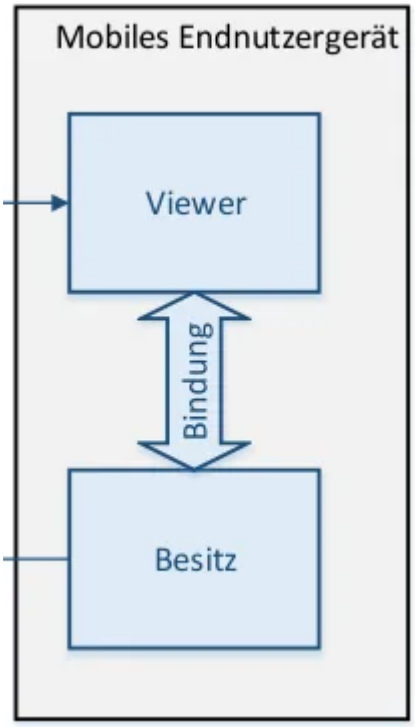


Abbildung 1: zeigt ein Modell aller Komponenten, die unter diesen Annahmen für die Umsetzung des Authentifizierungsfaktors Besitz notwendig sind. Als ersten Schritt identifiziert Abbildung 1 dazu die beiden Hauptkomponenten, die in die Umsetzung des Authentifizierungsfaktors Besitz und damit in den Authentifizierungsprozess involviert sind. Dies ist auf der einen Seite die Zentrale Applikation, an der sich die Benutzerin bzw. der Benutzer authentifizieren muss. Auf der anderen Seite ist dies das Mobile Endnutzergerät, das von der Benutzerin bzw. vom Benutzer für den Zugriff auf die Zentrale Applikation und für den Nachweis des Faktors Besitz verwendet wird.



Am *Mobilen Endnutzergerät* werden die erhaltenen Transaktionsdaten in einer geeigneten *Viewer* Komponente dargestellt. Dies gibt der Benutzerin bzw. dem Benutzer die Möglichkeit, die Daten einzusehen und zu kontrollieren. Um die Authentifizierung abzuschließen und die Transaktion zu autorisieren, muss ein entsprechender Besitznachweis an die *Zentrale Applikation* gesendet werden. Dieser entspringt der am *Mobilen Endgerät* vorhandenen Komponente *Besitz* . Diese Komponente wurde bewusst abstrakt gehalten, da in der Praxis unterschiedliche Möglichkeiten bestehen, diese Komponente umzusetzen und den Faktor Besitz zu implementieren. In jedem Fall obliegt es der Benutzerin bzw. dem Benutzer, unter Verwendung der Komponente *Besitz* einen entsprechenden Besitznachweis an die *Zentrale Applikation* zu senden und damit die Authentifizierung abzuschließen. Dabei ist von besonderer Relevanz, dass durch die Benutzerin bzw. den Benutzer sichergestellt werden kann, dass der gesendete Besitznachweis tatsächlich nur für die Bearbeitung der angezeigten Transaktionsdaten verwendet wird. Eine entsprechende Bindung zwischen den in der Komponente *Viewer* dargestellten Transaktionsdaten und dem über die Komponente *Besitz* gesendeten Besitznachweis ist daher unbedingt notwendig. Diese Notwendigkeit ist ebenfalls durch das in Abbildung 1 gezeigte Modell abgebildet.

Auf Seiten der *Zentralen Applikation* nimmt die Subkomponente *Verifikation* den gesendeten Besitznachweis entgegen. Aufgabe der Subkomponente *Verifikation* ist es, den erhaltenen Besitznachweis zu überprüfen. Dies inkludiert die Überprüfung, ob der erhaltene Nachweis auch eindeutig den an das mobile Endgerät übermittelten Transaktionsdaten zugeordnet ist. Die entsprechend notwendige Bindung zwischen den von der *Geschäftslogik* definierten Transaktionsdaten und dem erhaltenen Besitznachweis ist durch das in Abbildung 1 gezeigte Modell ebenfalls abgebildet.

Nach erfolgreicher Prüfung des Besitznachweises durch die Subkomponente *Verifikation* autorisiert diese schließlich die *Transaktionsausführung* in der dafür zuständigen Komponente. Dazu verwendet diese die von der *Geschäftslogik* zu Beginn bereitgestellten Transaktionsdaten.

Das in Abbildung 1 gezeigte Modell wurde bewusst auf einem abstrakten Level gehalten, ohne auf Spezifika eines speziellen Anwendungsfalls einzugehen. Mögliche Anwendungsszenarien sind beispielsweise E-Banking-Lösungen oder auch serverbasierte Signaturlösungen. Bei ersteren wäre die in Abbildung 1 gezeigte *Transaktionsausführung* beispielsweise die Durchführung einer Überweisung. Im Rahmen einer serverbasierten Signaturlösung wäre die *Transaktionsausführung* der Signaturerstellungsprozess. Durch die abstrakte Charakteristik des gewählten Modells ist dieses für beliebige Anwendungsszenarien gültig.

Gerade wegen seiner abstrakten Natur kann das in Abbildung 1 dargestellte Modell für die Herleitung von Anforderungen an Umsetzungen des Authentifizierungsfaktors Besitz herangezogen werden, da das Modell alle relevanten Komponenten und Beziehungen zwischen diesen Komponenten identifiziert. Dementsprechend können von diesem Modell die folgenden Anforderungen abgeleitet werden:

- **Anforderung 1: Implementierung auf einem einzigen mobilen Endnutzergerät:** Aus Abbildung 1 wird ersichtlich, dass das *Mobile Endnutzergerät* die einzige Schnittstelle der Benutzerin bzw. des Benutzers zur *Zentralen Applikation* ist. Dementsprechend müssen alle Funktionen, die zur Implementierung des Authentifizierungsfaktors Besitz notwendig sind, auf diesem Gerät umgesetzt werden. Die Verwendung eines zusätzlichen Geräts ist nicht möglich.

- **Anforderung 2: Eindeutige Bindung des Besitznachweises an die Transaktion:** Der von der Benutzerin bzw. vom Benutzer erbrachte Besitznachweis muss an die jeweilige Transaktion gebunden sein. Diese Bindung muss sowohl von der Benutzerin bzw. vom Benutzer, als auch von der

Zentralen Applikation überprüfbar sein.

- **Anforderung 3: Sichere Übermittlung von Transaktionsdaten:** Transaktionsdaten müssen von der *Zentralen Applikation* an die lokale *Viewer* -Komponente am *Mobilen Endnutzergerät* sicher übertragen werden. Das heißt, die Vertraulichkeit und Integrität dieser Daten muss im Zuge der Übermittlung sichergestellt werden.
- **Anforderung 4: Sichere Übermittlung von Besitznachweisen:** Besitznachweise, die im Zuge der Authentifizierung zwischen *Mobilem Endnutzergerät* und *Zentraler Applikation* übermittelt werden, müssen im Zuge der Übertragung geeignet geschützt werden.

4. Analyse und Evaluierung bestehender Ansätze

In den letzten Jahren wurden zahlreiche zweifaktorbasierte Authentifizierungsmethoden für mobile Endnutzergeräte vorgestellt. In diesem Abschnitt sollen diese Methoden analysiert und gegen die in Abschnitt 3 definierten Anforderungen evaluiert werden. Auf diese Weise werden jene Methoden identifiziert, die für eine Implementierung des Authentifizierungsfaktors Besitz unter den in Abschnitt 3 getroffenen Annahmen geeignet sind. Verschiedene dafür in Frage kommende Ansätze, Methoden und Lösungen werden in den folgenden Unterabschnitten behandelt.

4.1. Statische Besitznachweise

Die Verwendung einer personalisierten Smartphone-App ist wahrscheinlich der nächstliegende Ansatz, um Besitznachweise auf mobilen Endnutzergeräten zu realisieren. Eine personalisierte Smartphone-App ist eindeutig an eine bestimmte Benutzerin bzw. einen bestimmten Benutzer und an ein bestimmtes mobiles Endnutzergerät gebunden. Dadurch ist diese App auch in der Lage, statische Besitznachweise zu generieren, die für diese spezielle Instanz der App eindeutig sind. Damit sind personalisierte Smartphone-Apps prinzipiell in der Lage, den Authentifizierungsfaktor Besitz zu implementieren.

Der größte Nachteil dieses Ansatzes ist die statische Charakteristik verwendeter Besitznachweise. Obwohl generierte Besitznachweise für jede App-Instanz und damit für jede Benutzerin und jeden Benutzer einzigartig sind, wird doch von einer konkreten Instanz für jede Transaktion stets derselbe Besitznachweis verwendet. Dies ermöglicht die Durchführung von Replay-Attacken. Gelingt es einer Angreiferin oder einem Angreifer, einen Besitznachweis abzufangen, kann dieser für beliebige zukünftige Transaktionen wiederverwendet werden.

Ein weiterer Nachteil dieses Ansatzes ist ebenfalls durch die statische Natur der verwendeten Besitznachweise bedingt. Da sich Besitznachweise einzelner Transaktionen nicht unterscheiden, ist eine eindeutige Zuordnung von Besitznachweisen zu Transaktionen nicht möglich. Weder die Benutzerin bzw. der Benutzer, noch die *Zentrale Applikation*, die die Authentifizierung verlangt, sind in der Lage, Besitznachweise einer konkreten Transaktion zuzuordnen. Dies ist auch in Abbildung 2 dargestellt, welche die definierten Anforderungen diesem konkreten Ansatz gegenüberstellt.

Anforderungen	Statische Besitznachweise
Implementierung auf einem einzigen mobilen Endnutzergerät	Erfüllt
Eindeutige Bindung des Besitznachweises an die Transaktion	Nicht erfüllt
Sichere Übermittlung von Transaktionsdaten	Erfüllt
Sichere Übermittlung von Besitznachweisen	Erfüllt

Abbildung 2. Evaluierungsergebnisse statischer Besitznachweise. Aus Abbildung 2 wird auch ersichtlich, dass statische Besitznachweise in der Lage sind, alle anderen Anforderungen zu erfüllen. Konkret lässt sich dieser Ansatz auf aktuellen mobilen Endnutzergeräten implementieren, da dafür lediglich die Installation und Verwendung einer personalisierten App notwendig ist. Dies ist auf aktuellen mobilen Endnutzergeräten möglich. Des Weiteren sind derartige Apps in der Lage, über erprobte Protokolle wie SSL/TLS sichere Verbindungen zu entfernten Systemen aufzubauen. Dementsprechend können auch Transaktionsdaten und Besitznachweise sicher übermittelt werden.

Insgesamt sind statische Besitznachweise jedoch nicht in der Lage, alle definierten Anforderungen zu erfüllen. Erschwerend kommt hinzu, dass statische Besitznachweise außerdem anfällig gegenüber Replay-Attacken sind. Es ist daher wenig verwunderlich, dass dieser Ansatz in der Praxis selten implementiert wird.

4.2. Zeitbasierte Einmalpasswörter

Eine geeignete Gegenmaßnahme gegen Replay-Attacken ist die Verwendung dynamischer Ansätze. Bei diesen ist ein Besitznachweis lediglich für eine Verwendung gültig. Wird ein Besitznachweis abgehört, kann er daher nicht mehr für nachfolgende Transaktionen verwendet werden.

Ein gängiger Ansatz, um derartige dynamische Ansätze zu implementieren, sind zeitbasierte Einmalpasswörter. Diesem Ansatz folgend, sendet die Benutzerin bzw. der Benutzer im Zuge der Authentifizierung ein Einmalpasswort an die *Zentrale Applikation*, die die Authentifizierung durchführen möchte. Das Einmalpasswort wird in der Regel von einem Hardware-Token generiert, der zuvor an die Benutzerin bzw. den Benutzer ausgegeben wurde. Dieses Token generiert in konstanten Intervallen von der aktuellen Zeit abhängige Einmalpasswörter. Das aktuellste Passwort wird über ein Display angezeigt und kann im Zuge von Authentifizierungsprozessen verwendet werden. Durch Verwendung geeigneter kryptographischer Methoden können gültige Einmalpasswörter nur vom Token selbst generiert werden. Der Nachweis des Wissens um das aktuelle Passwort weist damit implizit auch den Besitz des Tokens nach.

Damit die *Zentrale Applikation* in der Lage ist, übermittelte Einmalpasswörter zu überprüfen, muss das an die Benutzerin bzw. den Benutzer ausgegebene Token mit der *Zentralen Applikation* gekoppelt sein. Der dafür nötige Pairing-Prozess wird in der Regel bei Ausgabe des Tokens, in jedem Fall jedoch vor dessen Verwendung durchgeführt.

In den letzten Jahren wurden zahlreiche Produkte vorgestellt, die den Ansatz zeitbasierter Einmalpasswörter verfolgen. Beispiele dafür sind die Lösungen SecurID [13], Safe Word 2008 [14] oder DIGIPASS [15]. All diese Lösungen verwenden ein dediziertes Hardware-Token, das mit einer *Zentralen Applikation* gekoppelt werden kann und für Benutzerinnen und Benutzer zeitbasierte Einmalpasswörter generiert.

In letzter Zeit wurden neben diesen Lösungen auch Produkte vorgestellt, die auf die Verwendung dedizierter Hardware-Token verzichten und deren Funktionalität stattdessen mithilfe einer Smartphone-App implementieren. Anstelle des Hardware-Tokens wird diese App mit der *Zentralen Applikation* gekoppelt und generiert in regelmäßigen Intervallen neue zeitbasierte Einmalpasswörter. Ein Beispiel für eine Lösung, die auf der Verwendung einer entsprechenden Smartphone-App beruht, ist Google Authenticator [17]. Die Lösung ist auf verschiedenen Plattformen verfügbar. Einmalpasswörter werden basierend auf Standards der OATH-Initiative [18] generiert. Im speziellen Fall zeitbasierter Einmalpasswörter werden diese nach RFC 6238 [19] berechnet.

Aufgrund der Tatsache, dass Verfahren, die auf der Verwendung zeitbasierter Einmalpasswörter beruhen, für jede Transaktion ein neues Passwort und damit einen neuen Besitznachweis generieren, wirken diese der Durchführung von Replay-Attacken entgegen. Trotzdem sind auch diese Verfahren nicht in der Lage, eine eindeutige Bindung zwischen der aktuellen Transaktion und dem jeweiligen Besitznachweis herzustellen. Obwohl verwendete Besitznachweise nur für einen bestimmten Zeitraum gültig sind, sind diese nicht spezifisch für eine spezielle Transaktion. Konkret kann die *Zentrale Applikation* nicht verifizieren, ob ein erhaltenes Einmalpasswort tatsächlich für die aktuelle Transaktion vorgesehen war. Dies ist auch in Abbildung 3 dargestellt.

Abbildung 3 zeigt auch, dass zeitbasierte Einmalpasswörter ähnlich wie auch statische Besitznachweise in der Lage sind, die restlichen drei definierten Anforderungen zu erfüllen. Bereits verfügbare auf Smartphone-Apps beruhende Lösungen zeigen, dass Verfahren, die auf zeitbasierten Einmalpasswörtern beruhen, auch auf aktuellen mobilen Endnutzergeräten umgesetzt werden können. Eine sichere Kommunikation zwischen Smartphone-Apps und *Zentralen Applikationen* kann wie erwähnt über erprobte Protokolle wie SSL/TLS erreicht werden. Dadurch kann auch eine sichere Übermittlung von Transaktionsdaten und Besitznachweisen, d.h. Einmalpasswörtern, gewährleistet werden.

Anforderungen	**Zeitbasierte Einmalpasswörter**
Implementierung auf einem einzigen mobilen Endnutzergerät	Erfüllt
Eindeutige Bindung des Besitznachweises an die Transaktion	Nicht erfüllt
Sichere Übermittlung von Transaktionsdaten	Erfüllt
Sichere Übermittlung von Besitznachweisen	Erfüllt

Abbildung 3. Evaluierungsergebnisse zeitbasierter Einmalpasswörter. Als einziger Nachteil zeitbasierter Einmalpasswörter verbleibt somit die nicht vorhandene Bindung zwischen dem Besitznachweis und der aktuellen Transaktion. Dadurch ist auch dieses Verfahren für Anwendungen unter den in Abschnitt 3 definierten Annahmen nicht geeignet.

4.3. Eventbasierte Einmalpasswörter

Eventbasierte Einmalpasswörter sind ihren zeitbasierten Pendants konzeptionell sehr ähnlich. Auch hier werden Besitznachweise über Einmalpasswörter implementiert. Durch den Nachweis, das Einmalpasswort zu kennen, wird implizit auch der Besitz des Tokens, das für die Generierung dieses Passworts notwendig ist, nachgewiesen.

Eventbasierte Einmalpasswörter unterscheiden sich von zeitbasierten Einmalpasswörtern jedoch in der Art ihrer Erstellung. Während für zeitbasierte Einmalpasswörter die aktuelle Zeit in deren Berechnung eingeht, basiert die Generierung von eventbasierten Einmalpasswörtern auf einem bestimmten Ereignis. Im einfachsten Fall dient ein Zähler als Basis, der nach jeder Passwortgenerierung inkrementiert wird. Dieser Zähler muss zwischen *Zentraler Applikation* und lokalem Token synchronisiert sein, um eine zentrale Verifikation von generierten Einmalpasswörtern zu ermöglichen.

Die Verwendung eines Zählers zur Berechnung von Einmalpasswörtern ist beispielsweise in RFC 4226 [16] spezifiziert. Dieser Standard wurde beispielsweise durch Google Authenticator [17] oder auch durch das Barada Projekt [20] implementiert. Letzteres bietet eine Smartphone-App für Google Android, über welche Einmalpasswörter gemäß RFC 4226 erstellt werden können.

Aufgrund der konzeptionellen Ähnlichkeit zu zeitbasierten Einmalpasswörtern, ergeben sich für eventbasierte Einmalpasswörter ähnliche Überlegungen in Bezug auf die Erfüllung der in Abschnitt 3 definierten Anforderungen. Dies ist in Abbildung 4 dargestellt.

Anforderungen	Eventbasierte Einmalpasswörter
Implementierung auf einem einzigen mobilen Endnutzergerät	Erfüllt
Eindeutige Bindung des Besitznachweises an die Transaktion	Nicht erfüllt
Sichere Übermittlung von Transaktionsdaten	Erfüllt
Sichere Übermittlung von Besitznachweisen	Erfüllt

Abbildung 4. Evaluierungsergebnisse eventbasierter Einmalpasswörter. Auch für eventbasierte Einmalpasswörter kann die nicht vorhandene Bindung zwischen der aktuellen Transaktion und dem verwendeten Besitznachweis, d.h. dem Einmalpasswort, als größter Nachteil identifiziert werden. Auch dieser Ansatz ist daher unter den in Abschnitt 3 getroffenen Annahmen nicht geeignet.

4.4. SMS-TAN-Verfahren

Das prinzipielle Konzept des SMS-TAN-Verfahrens wurde bereits in Abschnitt 2.2 rudimentär beschrieben. Dort wurde auch festgestellt, dass dieses Verfahren prinzipiell die Verwendung von zwei getrennten Endnutzergeräten vorsieht und daher für Lösungen, die auf einem einzigen mobilen Endnutzergerät verwendet werden sollen, nicht geeignet ist. Der Vollständigkeit halber soll dieses Verfahren dennoch auch hier betrachtet und gegen die in Abschnitt 3 definierten Anforderungen evaluiert werden.

Beim SMS-TAN-Verfahren wird der Faktor Besitz über die SIM der Benutzerin bzw. des Benutzers abgedeckt. Der Besitz der SIM wird überprüft, indem ein Einmalpasswort – eine sogenannte Transaktionsnummer oder TAN – an die SIM gesendet wird. Durch Bekanntgabe der erhaltenen TAN weist die Benutzerin bzw. der Benutzer nach, im Besitz der SIM zu sein. Die TAN fungiert in diesem Fall also als Besitznachweis.

Obwohl auch beim SMS-TAN-Verfahren Einmalpasswörter zum Einsatz kommen, ergibt sich dennoch ein interessanter konzeptioneller Unterschied zu den bereits diskutierten Verfahren, die auf zeitbasierten oder eventbasierten Einmalpasswörtern beruhen. Zeit- oder eventbasierte Einmalpasswörter werden

unabhängig voneinander in der *Zentralen Applikation* und bei der Benutzerin bzw. beim Benutzer generiert. Dementsprechend inkludiert der Vorgang des Besitznachweises lediglich einen Kommunikationsschritt, nämlich die Übermittlung des lokal generierten Einmalpassworts an die *Zentrale Applikation*, die die Authentifizierung verlangt. Im Gegensatz dazu wird beim SMS-TAN-Verfahren das Einmalpasswort, d.h. die TAN, ausschließlich von der *Zentralen Applikation* generiert. Dementsprechend inkludiert der Vorgang des Besitznachweises auch zwei aufeinanderfolgende Kommunikationsschritte. Die TAN wird zunächst von der *Zentralen Applikation* an das *Mobile Endnutzergerät* übertragen. Im Anschluss wird die TAN unverändert an die *Zentrale Applikation* retourniert.

Die ausschließlich zentrale Generierung des Einmalpassworts macht auf der einen Seite einen zusätzlichen Kommunikationsschritt notwendig, bringt auf der anderen Seite jedoch auch einen signifikanten konzeptionellen Vorteil. Da der Besitznachweis von der *Zentralen Applikation* generiert wird, kann diese den Nachweis eindeutig einer Transaktion zuordnen. Durch die Implementierung geeigneter Methoden kann außerdem sichergestellt werden, dass auch die Benutzerin bzw. der Benutzer in der Lage ist, diese Zuordnung zu verifizieren. Aktuelle Anwendungen, die auf das SMS-TAN-Verfahren zurückgreifen, verwenden dazu beispielsweise einen zusätzlichen Referenzwert. Dieser wird gleichzeitig mit der TAN generiert und zusammen mit dieser an die Benutzerin bzw. den Benutzer gesendet. Derselbe Referenzwert wird auch in der lokalen *Viewer* -Komponente zusammen mit den Transaktionsdaten angezeigt. Durch Vergleich der beiden über verschiedene Kanäle empfangenen Referenzwerte kann die Benutzerin bzw. der Benutzer überprüfen, ob die erhaltene TAN den angezeigten Transaktionsdaten zugeordnet ist. Damit ist das SMS-TAN-Verfahren im Gegensatz zu allen bisher diskutierten Methoden in der Lage, eine entsprechende Bindung zwischen der jeweiligen Transaktion und dem verwendeten Besitznachweis verifizierbar sicherzustellen.

Die ausschließlich zentrale Erstellung der TAN wirkt sich jedoch auf die Erfüllung anderer in Abschnitt 3 definierter Anforderungen negativ aus. Konkret betrifft dies die Forderung nach einer sicheren Übermittlung von Besitznachweisen. Das SMS-TAN-Verfahren verlangt, dass die zentral generierte TAN per SMS an das *Mobile Endnutzergerät* übertragen wird. Dies ist notwendig, da nur auf diese Weise der Besitz der SIM verifiziert werden kann. Allerdings ist die SMS-Technologie bekannt dafür, keine sichere Datenübermittlung gewährleisten zu können. Im Speziellen gilt dies für moderne Smartphones, auf denen eingehende SMS-Nachrichten je nach Plattform relativ einfach kompromittiert werden können. Da die Vertraulichkeit versendeter TANs für die Sicherheit des Verfahrens notwendig ist, kann das SMS-TAN-Verfahren die Forderung nach einer sicheren Übertragung von Besitznachweisen nicht erfüllen. Da das SMS-TAN Verfahren in der Regel nur einen Authentifizierungsfaktor abdeckt, ist die potentielle Angreifbarkeit von via SMS übermittelter Daten nicht von entscheidender Bedeutung. Bei herkömmlichen E-Banking-Lösungen, die auf dem SMS-TAN-Verfahren beruhen, ist beispielsweise für die Autorisierung einer Transaktion auch neben der TAN auch eine authentifizierte Browser-Session notwendig. Durch Kompromittierung der TAN alleine kann eine Angreiferin bzw. ein Angreifer die Sicherheit des Gesamtsystems daher nicht untergraben. Kritischer ist die Sachlage, wenn auch die Browser-Session am selben Gerät vorhanden ist, über das auch die TAN empfangen wird. Dieser Fall ist jedoch üblicherweise durch entsprechende Nutzungsrichtlinien ausgeschlossen.

Anforderungen	SMS-TAN-Verfahren
Implementierung auf einem einzigen mobilen Endnutzergerät	Erfüllt
Eindeutige Bindung des Besitznachweises an die Transaktion	Erfüllt
Sichere Übermittlung von Transaktionsdaten	Erfüllt
Sichere Übermittlung von Besitznachweisen	Nicht erfüllt

Abbildung 5. Evaluierungsergebnisse des SMS-TAN-Verfahrens. Abbildung 5 zeigt zusammenfassend, welche der definierten Anforderungen durch das SMS-TAN-Verfahren erfüllt werden können. Aus Abbildung 5 wird ersichtlich, dass SMS-TAN-Verfahren keine sichere Übermittlung von Besitznachweisen garantieren können, jedoch in der Lage sind, alle anderen Anforderungen zu erfüllen. So lassen sich derartige Verfahren – obwohl ursprünglich für zwei getrennte Endnutzergeräte konzipiert – prinzipiell auch auf einem einzelnen Gerät implementieren, sofern dieses in der Lage ist, SMS-Nachrichten zu empfangen und als Client für die Übermittlung erhaltener TANs an die *Zentrale Applikation* zu dienen. Dies trifft auf moderne Smartphones in der Regel zu, womit die Anforderung nach Umsetzbarkeit auf einem einzelnen mobilen Endnutzergerät aus technischer Sicht erfüllt ist. Da Smartphones mit *Zentralen Applikationen* durch Verwendung geeigneter Kommunikationsprotokolle sicher kommunizieren können, ist auch die Forderung nach einer sicheren Übermittlung von Transaktionsdaten erfüllt. Für die Übermittlung von Besitznachnachweisen gilt dies nicht, da für diese eine Verwendung der potentiell unsicheren SMS-Technologie notwendig ist. Insgesamt ist daher auch das SMS-TAN-Verfahren für eine Verwendung unter den in Abschnitt 3 getroffenen Annahmen nicht geeignet.

4.5. Challenge-Response-Ansätze

Challenge-Response-Ansätze sind aus konzeptioneller Sicht dem SMS-TAN-Verfahren ähnlich. Dies betrifft vor allem die Anzahl der für die Erbringung eines Besitznachweises nötigen Kommunikationsschritte. Wie beim SMS-TAN-Verfahren sind auch bei Challenge-Response-Ansätzen hierfür zwei aufeinanderfolgende Kommunikationsschritte notwendig. Um einen Besitznachweis zu erhalten, generiert die *Zentrale Applikation*, die die Benutzerin bzw. den Benutzer authentifizieren möchte, zunächst eine Challenge. Je nach eingesetztem Verfahren kann es sich dabei zum Beispiel um eine Zufallszahl handeln. Wichtig dabei ist, dass für jede Transaktion eine neue eindeutige Challenge generiert wird.

Im ersten Kommunikationsschritt wird die zentral generierte Challenge an das *Mobile Endnutzergerät* übermittelt. Dort wird aus der erhaltenen Challenge und in der Regel unter Verwendung einer geeigneten kryptographischen Methode eine Response berechnet. Die Berechnung dieser Response beruht auf der Verwendung eines geheimen kryptographischen Schlüssels. Dieser Schlüssel ist an ein bestimmtes Gerät gebunden und deckt damit im Wesentlichen den Authentifizierungsfaktor Besitz ab. Dazu muss gewährleistet sein, dass eine korrekte Response ohne diesen Schlüssel nicht berechenbar ist. Die Anforderungen kann durch Wahl geeigneter kryptographischer Methoden erfüllt werden. Die so berechnete Response repräsentiert schließlich den Besitznachweis und wird in einem zweiten Kommunikationsschritt an die *Zentrale Applikation* übertragen.

Trotz ihrer teilweisen Ähnlichkeit zum SMS-TAN-Verfahren unterscheiden sich Challenge-Response-Ansätze in einigen relevanten Punkten.

- Obwohl sowohl das SMS-TAN-Verfahren als auch Challenge-Response-Ansätze auf zwei aufeinanderfolgenden Kommunikationsschritten beruhen, unterscheiden sich die beiden Varianten in der Art der übermittelten Daten. Beim SMS-TAN-Verfahren werden dieselben Daten, d.h. die TAN, sowohl von der *Zentralen Applikation* zum *Mobilen Endnutzergerät*, als auch von diesem wieder zurück an die *Zentrale Applikation* übertragen. Beim Challenge-Response-Verfahren unterscheiden sich die in den beiden Kommunikationsschritten übermittelten Daten. Da der Besitznachweis bei Challenge-Response-Ansätzen ausschließlich durch die Response repräsentiert wird, ist die Geheimhaltung der von der *Zentralen Applikation* an das *Mobile Endnutzergerät* übertragenen Challenge für die Sicherheit nicht von entscheidender Bedeutung.

- Beim SMS-TAN Verfahren wird der Authentifizierungsfaktor Besitz durch die SIM der Benutzerin bzw. des Benutzers implementiert. Damit ist auch die Übermittlung der TAN an das *Mobile Endnutzergerät* an die SMS-Technologie gebunden. Bei Challenge-Response Ansätzen wird der Authentifizierungsfaktor Besitz hingegen über einen an ein bestimmtes Gerät gebundenen kryptographischen Schlüssel abgedeckt. Dadurch ist die Verwendung der SMS-Technologie zur Miteinbeziehung der SIM nicht mehr zwingend notwendig, wodurch sich mehr Umsetzungsmöglichkeiten ergeben.

Durch die Notwendigkeit, kryptographische Methoden zur Berechnung der Response zu implementieren, waren Challenge-Response-Ansätze auf klassischen Mobiltelefonen schwer bis gar nicht umsetzbar. Moderne Smartphones und verwandte mobile Endnutzergeräte bieten diesbezüglich mehr Flexibilität und ermöglichen so die Implementierung entsprechender Lösungen. Dies führte in den letzten Jahren zur Entwicklung zahlreicher Lösungen, die sichere Authentifizierungslösungen basierend auf Challenge-Response-Verfahren anbieten.

Prinzipiell können solche Lösungen in zwei Kategorien klassifiziert werden. Als Klassifizierungskriterium dient dabei die Art und Weise der Umsetzung der Berechnung der Response und der Speicherung des dafür notwendigen kryptographischen Schlüssels. Die Speicherung kryptographischen Schlüsselmaterials und die Umsetzung kryptographischer Methoden können entweder in Software oder in Hardware erfolgen. Dementsprechend können Umsetzungen basierend auf Challenge-Response-Ansätzen für mobile Endnutzergeräte in softwarebasierte und hardwarebasierte Lösungen unterteilt werden. Für beide Kategorien wurden in den letzten Jahren entsprechende Implementierungen vorgestellt.

Für softwarebasierte Lösungen wurde in den letzten Jahren unter anderem auf die QR-Code-Technologie zurückgegriffen. Google experimentierte in diese Richtung im Rahmen des Forschungsprojekts Google Sesame [21]. Eine andere QR-basierte Lösung, die in den letzten Jahren vorgestellt wurde, ist SQRL [22]. SQRL steht für Secure Quick Reliable Login und hat zum Ziel, Benutzerauthentifizierungen an Websites sicherer und benutzerfreundlicher zu gestalten. SQRL sieht dazu die Verwendung einer Smartphone-App vor. Diese implementiert im Wesentlichen ein Challenge-Response-Verfahren, über das sich Benutzerinnen und Benutzer an Websites anmelden können.

Hardwarebasierte Lösungen speichern geheimes Schlüsselmaterial in sicheren Hardware-Elementen und nutzen diese auch, um benötigte kryptographische Operationen zu implementieren. Damit weisen diese Lösungen im Vergleich zu rein softwarebasierten Ansätzen einen höheren Grad an Sicherheit auf. Gleichzeitig sind hardwarebasierte Lösungen jedoch auch schwieriger umzusetzen, da diese höhere Anforderungen an das mobile Endnutzergerät stellen. Konkret erweist sich in der Praxis vor allem die

Realisierung des sicheren Hardware-Elements oft als schwierig. Ein konkretes Beispiel für eine hardwarebasierte Lösung ist U2F, das von der FIDO Alliance vorgestellt wurde [23]. U2F macht die Verwendung eines Hardware-Dongles am – nicht notwendigerweise mobilen – Endnutzergerät notwendig. Die U2F-Spezifikationen sind jedoch abstrakt genug gehalten, um die Verwendung unterschiedlicher Technologien zur Umsetzung dieses Dongles zu unterstützen.

Unabhängig von der Art der Umsetzung sind Challenge-Response-Ansätze in der Lage, alle in Abschnitt 3 definierten Anforderungen zu erfüllen. Zahlreiche existierende Lösungen zeigen, dass diese Ansätze auf modernen mobilen Endnutzergeräten prinzipiell umsetzbar sind. Durch den bereits vom SMS-TAN-Verfahren bekannten Ansatz, die Basis für den Besitznachweis, d.h. im konkreten Fall die Challenge, durch die *Zentrale Applikation* berechnen zu lassen, ist auch eine eindeutige Bindung zwischen der aktuellen Transaktion und dem generierten Besitznachweis gegeben. Die *Zentrale Applikation* kann die Challenge eindeutig der aktuellen Transaktion zuordnen. Da die Response ebenfalls eindeutig für die jeweilige Challenge ist, kann auch die Response, d.h. der Besitznachweis, eindeutig der aktuellen Transaktion zugeordnet werden. Unter der legitimen Annahme, dass eine Smartphone-App unter Verwendung erprobter Protokolle und Technologien in der Lage ist, Daten sicher mit der *Zentralen Applikation* auszutauschen, ist auch die Forderung nach einer sicheren Übermittlung von Transaktionsdaten und Besitznachweisen erfüllt. In Bezug auf die sichere Übertragung von Besitznachweisen ergeben sich für Challenge-Response-Ansätze im Vergleich zum SMS-TAN-Verfahren im Speziellen zwei Vorteile. Erstens ist bei Challenge-Response-Ansätzen die Vertraulichkeit der übermittelten Challenge nicht von Bedeutung. Auch wenn es einer Angreiferin oder einem Angreifer gelingt, die Challenge zu kompromittieren, ist für sie oder ihn die Berechnung einer gültigen Response ohne Wissen des geheimen kryptographischen Schlüssels nicht möglich. Die Forderung nach einer sicheren Übermittlung von Besitznachweisen bezieht sich bei Challenge-Response-Ansätzen daher ausschließlich auf die Übertragung der Response an die *Zentrale Applikation* . Zweitens bedingen Challenge-Response-Ansätze im Gegensatz zum SMS-TAN-Verfahren nicht zwingendermaßen die Verwendung der potentiell unsicheren SMS-Technologie zur Übertragung von Daten. Optional kann die SMS-Technologie trotzdem für die Übertragung der Challenge herangezogen werden, da die Vertraulichkeit der Challenge ohnehin nicht von zentraler Bedeutung ist.

Anforderungen	**Challenge-Response-Ansätze**
Implementierung auf einem einzigen mobilen Endnutzergerät	Erfüllt
Eindeutige Bindung des Besitznachweises an die Transaktion	Erfüllt
Sichere Übermittlung von Transaktionsdaten	Erfüllt
Sichere Übermittlung von Besitznachweisen	Erfüllt

Abbildung 6. Evaluierungsergebnisse von Challenge-Response-Ansätzen. Insgesamt kann festgehalten werden, dass Challenge-Response-Ansätze in der Lage sind, alle in Abschnitt 3 definierten Anforderungen zu erfüllen. Dies ist auch in Abbildung 6 dargestellt. Damit sind diese Ansätze anderen Methoden vorzuziehen, die zumindest eine der Anforderungen nicht erfüllen können. Aus dieser Erkenntnis kann auch das zentrale Ergebnis der durchgeführten Analyse und Evaluierung bestehender Ansätze zur

Implementierung des Authentifizierungsfaktors Besitz auf mobilen Endnutzergeräten abgeleitet werden: Challenge-Response-Ansätze stellen den bestgeeigneten Ansatz da, um sichere Mehrfaktorauthentifizierung auf mobilen Endnutzergeräten umzusetzen.

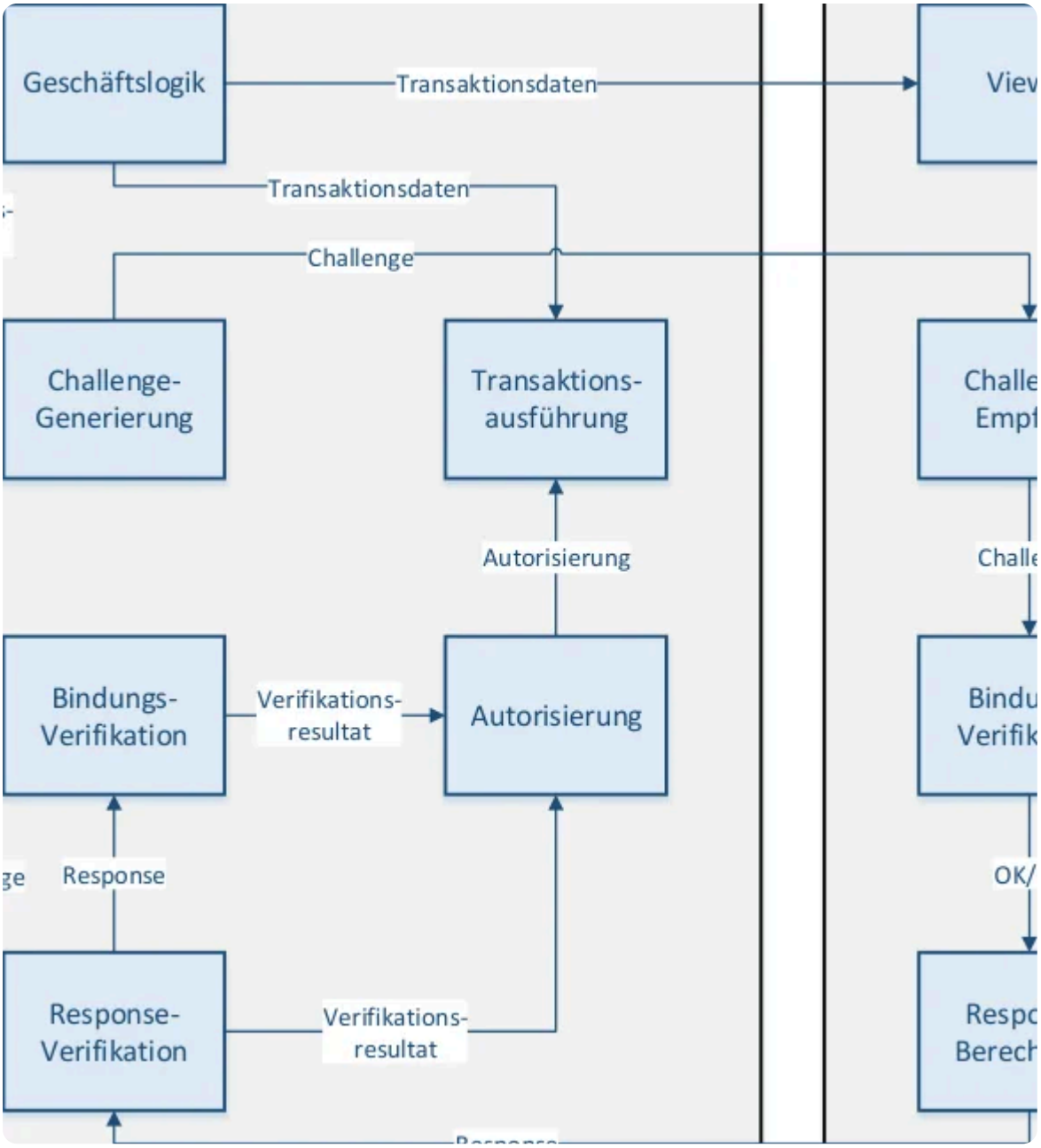
5. Lösung

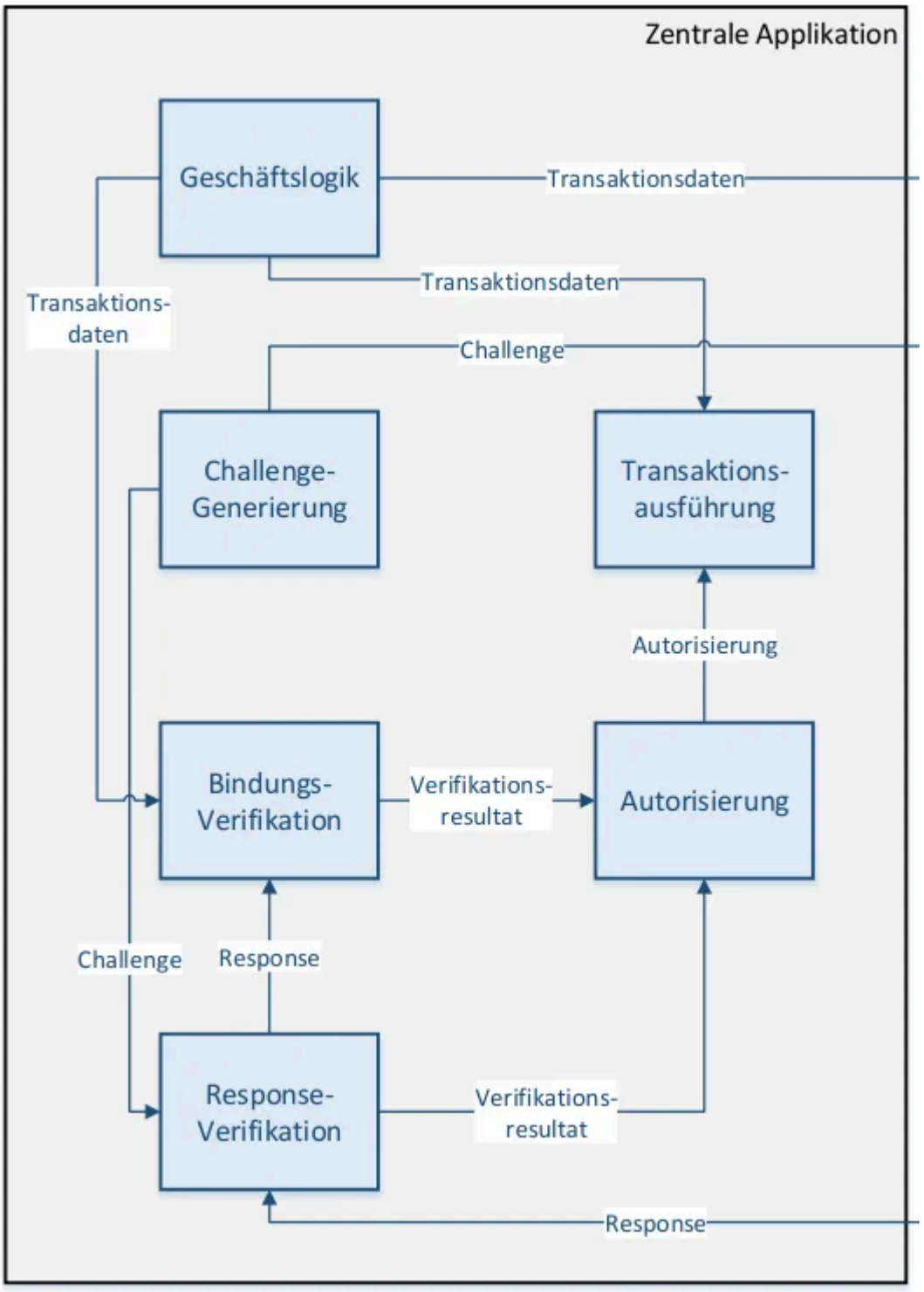
Die durchgeführten Analyse und Evaluierungen zeigten, dass unter den in Abschnitt 3 getroffenen Annahmen Challenge-Response-Ansätze am besten geeignet sind, sichere Mehrfaktorauthentifizierung unter Miteinbeziehung des Faktors Besitz zu implementieren. Diese Erkenntnis wird in diesem Abschnitt genutzt, um das in Abschnitt 3 definierte abstrakte Modell weiter zu verfeinern. In einem zweiten Schritt wird das verfeinerte Modell dann auf ein konkretes Anwendungsszenario angewendet. Dadurch wird gezeigt, wie das Modell im Rahmen einer konkreten Anwendung verwendet werden kann, um einen geeigneten Authentifizierungsmechanismus zu modellieren und schließlich umzusetzen.

5.1. Verfeinertes Modell

In Abschnitt 3 dieser Studie wurde ein Modell definiert, das eine abstrakte Umsetzung des Authentifizierungsfaktors Besitz zeigt. Dem Modell wurde die Annahme zu Grunde gelegt, dass eine serverbasierte zentrale Applikation für die Durchführung einer sicherheitskritischen Transaktion eine Benutzerin bzw. einen Benutzer sicher und verlässlich authentifizieren muss und dazu auf den Authentifizierungsfaktor Besitz zurückgreift. Des Weiteren wurde angenommen, dass die Benutzerin bzw. der Benutzer auf die Applikation ausschließlich über ihr bzw. sein mobiles Endnutzergerät zugreift und dieses auch verwendet, um einen entsprechenden Besitznachweis zu erbringen.

Dieses sehr abstrakt gehaltene Modell kann nun weiter verfeinert werden. Dafür werden die Ergebnisse der durchgeführten Evaluierung unterschiedlicher Methoden zur Bereitstellung von Besitznachweisen auf mobilen Endgeräten herangezogen. Entsprechend diesen Ergebnissen wird das abstrakte Modell dahingehend verfeinert, dass dieses einen Challenge-Response-Ansatz abbildet. Dies ergibt schlussendlich das in Abbildung 7 dargestellte verfeinerte Modell zur Implementierung des Authentifizierungsfaktors Besitz.


Geschäftslogik
Transaktionsdaten
Viev
Transaktionsdaten
Challenge
Challenge-
Generierung
Transaktions-
ausführung
Challe
Empf
Autorisierung
Challe
Bindungs-
Verifikation
Verifikations-
resultat
Autorisierung
Bindu
Verifik
ge
Response
OK/
Response-
Verifikation
Verifikations-
resultat
Respo
Berech


Zentrale Applikation
Geschäftslogik
Transaktionsdaten
Transaktionsdaten
Transaktions-
daten
Challenge
Challenge-
Generierung
Transaktions-
ausführung
Autorisierung
Bindungs-
Verifikation
Verifikations-
resultat
Autorisierung
Challenge
Response
Response-
Verifikation
Verifikations-
resultat
Response

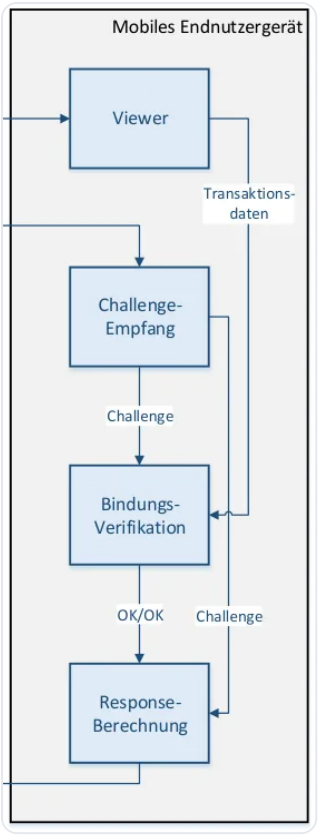


Abbildung 7. Verfeinertes Modell zur Implementierung des Authentifizierungsfaktors Besitz. Das verfeinerte Modell definiert mit der Zentralen Applikation und dem Mobilen Endnutzergerät dieselben Hauptkomponenten wie das zugrundeliegende abstrakte Modell. Auch die Subkomponenten Geschäftslogik und Transaktionsausführung der zentralen Komponente, sowie die Subkomponente Viewer des Mobilen Endnutzergeräts wurden unverändert übernommen. Die beiden Subkomponenten Besitz und Verifikation des abstrakten Modells wurden im verfeinerten Modell hingegen ersetzt und detaillierter ausgeführt. Entsprechend der in Abbildung 7 dargestellten Architektur gliedert sich der Besitznachweis im Rahmen eines Authentifizierungsprozesses in die folgenden Schritte:

ktions ten hallen	**Col2**
hallen ktions ten	en
hall	hall

ransa da hallen	**Col2**
hall	hall
hall	en

1. Die Subkomponente *Geschäftslogik* generiert Transaktionsdaten, deren Verarbeitung in der Subkomponente *Transaktionsausführung* einer Autorisierung und damit einer Authentifizierung der Benutzerin oder des Benutzers bedarf.
2. Dazu sendet die Subkomponente *Geschäftslogik* die generierten Transaktionsdaten an die *Viewer* -Komponente des *Mobilen Endnutzergeräts* .

3. Gleichzeitig generiert die Subkomponente *Challenge-Generierung* eine zufällige Challenge und sendet diese an das *Mobile Endnutzergerät*, wo diese von der Subkomponente *Challenge-Empfang* entgegengenommen wird.

4. Die lokale Subkomponente *Bindungs-Verifikation* des *Mobilen Endnutzergeräts* erhält von der *Viewer* -Komponente die Transaktionsdaten und von der Subkomponente *Challenge- Empfang* die aktuelle Challenge. Mit diesen Daten überprüft die Subkomponente *Bindungs- Verifikation* die korrekte Bindung zwischen angezeigten Transaktionsdaten und erhaltener Challenge.

5. Ist diese Überprüfung positiv, erstellt die Subkomponente *Response-Berechnung* aus der erhaltenen Challenge die zugehörige Response. Dazu wird die Challenge von der lokalen Subkomponente *Challenge-Empfang* bezogen.

6. Die berechnete Response wird an die Subkomponente *Response-Verifikation* der *Zentralen Applikation* gesendet. Diese verifiziert die erhaltene Response. Das Ergebnis der Überprüfung wird an die Subkomponente *Autorisierung* weitergeleitet.

7. Zusätzlich leitet die Subkomponente *Response-Verifikation* die erhaltene Response an die zentrale Subkomponente *Bindungs-Verifikation* weiter. Zusätzlich werden auch die von der *Geschäftslogik* generierten Transaktionsdaten an diese Subkomponente übertragen. Damit ist die Subkomponente *Bindungs-Verifikation* in der Lage, die Bindung zwischen aktuellen Transaktionsdaten und erhaltener Response zu überprüfen. Das Ergebnis dieser Überprüfung wird ebenfalls an die Subkomponente *Autorisierung* weitergeleitet.

8. Die Subkomponente *Autorisierung* analysiert die beiden erhaltenen Verifikationsergebnisse. Sind beide positiv, wird die *Transaktionsausführung* in der entsprechenden zentralen Subkomponente autorisiert.

5.2. Mapping auf konkretes Anwendungsszenario

Das in Abbildung 7 dargestellte verfeinerte Modell spezifiziert zwar die Verfolgung eines Challenge-Response-Ansatzes zur Implementierung des Authentifizierungsfaktors Besitz, ist aber trotzdem allgemein gehalten, sodass dieses für beliebige Anwendungsszenarien anwendbar bleibt. In diesem Abschnitt wird das erarbeitete verfeinerte Modell nun auf ein konkretes Anwendungsszenario angewendet. Damit wird gezeigt, dass das Model auch für konkrete Anwendungen geeignet ist.

Konkret wird in diesem Abschnitt gezeigt, wie das in Abbildung 7 gezeigte Modell verwendet werden kann, um serverbasierte Signaturlösungen auch auf mobilen Endnutzergeräten verwendbar zu machen. Aktuelle Signaturlösungen wie die österreichische Handy-Signatur weisen diesbezüglich Limitierungen auf, da diese eine Verwendung zweier getrennter Endnutzergeräte vorsehen. Eine Verwendung dieser Lösungen auf einem einzelnen Endnutzergerät ist daher in der Regel nicht möglich. Durch Integration einer geeigneten alternativen Authentifizierungsmethode kann dieser Limitierung potentiell entgegengewirkt werden.

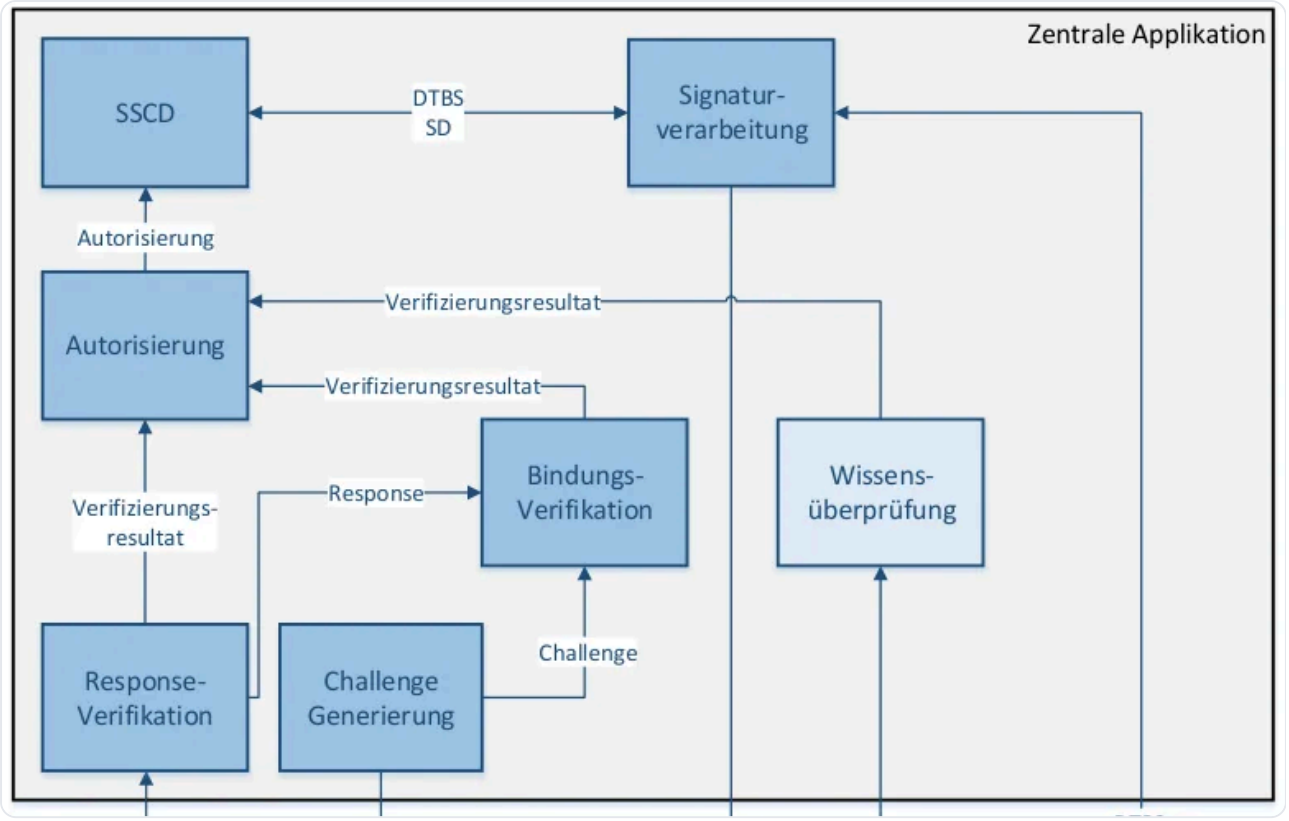
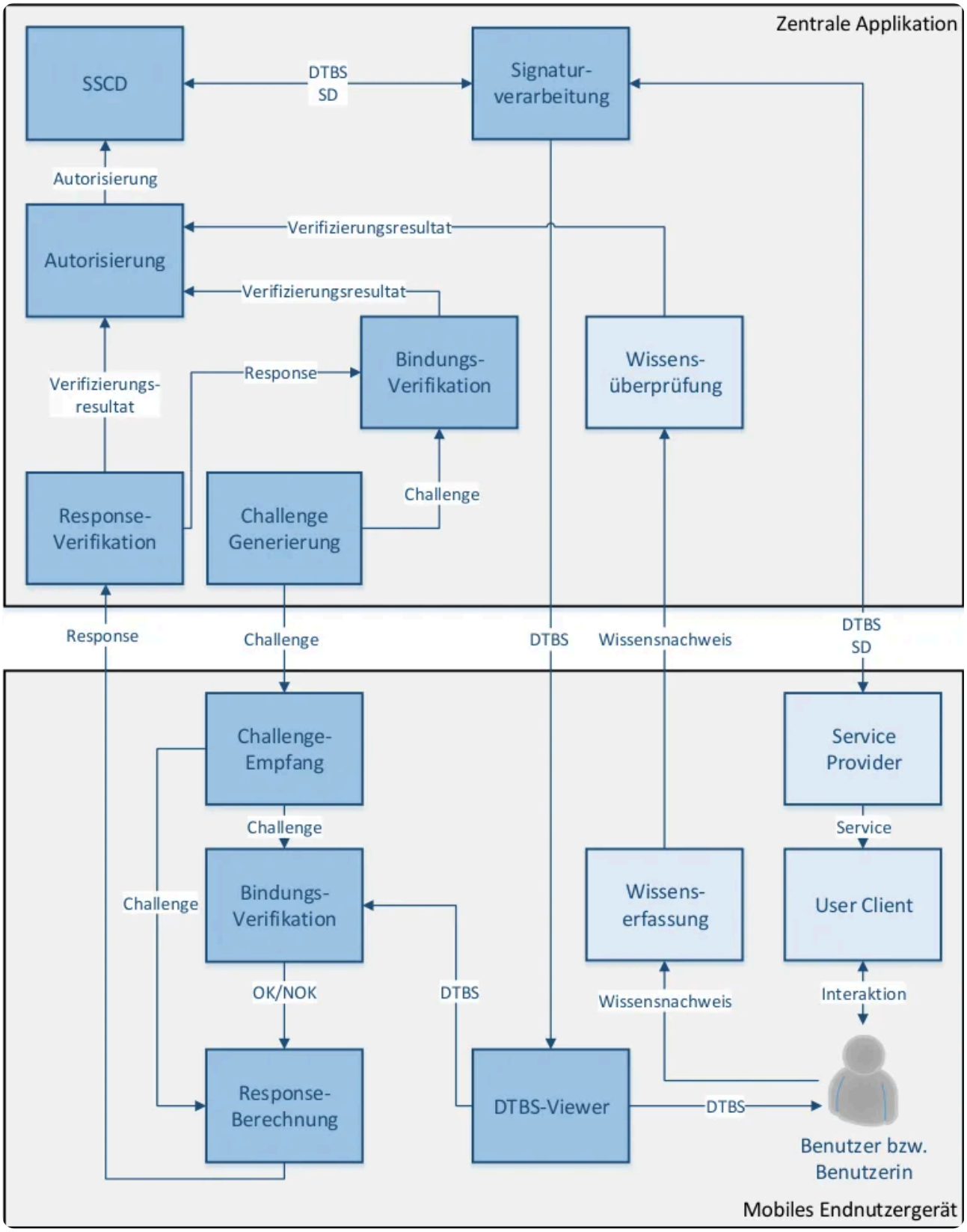
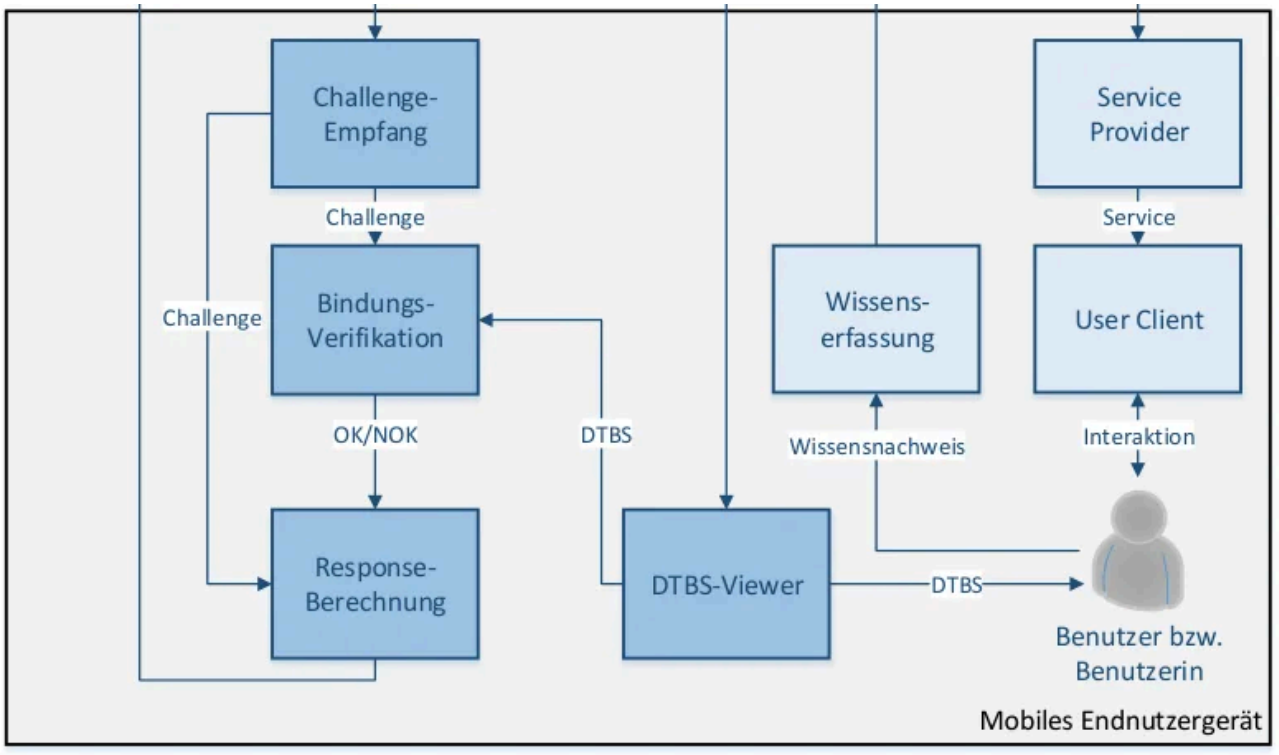


Abbildung 8. Modell einer serverbasierten Signaturlösung. Abbildung 8 zeigt ein Modell einer serverbasierten Signaturlösung, die auf einer geeigneten Zweifaktorauthentifizierung beruht. Konkret verwendet diese Lösung die Authentifizierungsfaktoren Wissen und Besitz. Der Faktor Besitz wird dabei über einen Challenge-Response-Ansatz abgedeckt. Dementsprechend integriert das in Abbildung 8 gezeigte Modell der Signaturlösung auch das in Abbildung 7 dargestellte verfeinerte Modell. Dessen Komponenten sind in Abbildung 8 aus Gründen der Übersichtlichkeit farblich hervorgehoben. Dabei ist zu beachten, dass einige Komponenten entsprechend dem konkreten Anwendungsfall umbenannt wurden. So wird die Komponente Transaktionsausführung in Abbildung 8 durch die Komponente SSCD (Secure Signature Creation Device) repräsentiert, da die kritische Transaktion im Rahmen einer Signaturlösung der Signaturerstellungsprozess ist, der – zumindest bei Lösungen, die die Erstellung qualifizierter Signaturen ermöglichen – in einem SSCD stattfindet. Entsprechend wurde auch die generische Komponente Geschäftslogik in die konkrete Komponente Signaturverarbeitung übergeführt. Auch am Mobilen Endnutzergerät wurde eine Komponente näher spezifiziert. Hier wurde die generische Komponente Viewer in die spezifischere Komponente DTBS-Viewer übergeführt. Dadurch wird die Tatsache berücksichtigt, dass es sich bei Transaktionsdaten im Rahmen von Signaturlösungen um die zu signierenden Daten, d.h. Data-To-Be-Signed (DTBS), handelt. Diese werden vor der Autorisierung des Signaturerstellungsprozesses der Benutzerin bzw. dem Benutzer über die lokale Viewer -Komponente angezeigt.


Zentrale Applikation
SSCD
DTBS
SD
Signatur-
verarbeitung
Autorisierung
Autorisierung
Verifizierungsresultat
Verifizierungsresultat
Bindungs-
Verifikation
Wissens-
überprüfung
Verifizierungs-
resultat
Response
Response-
Verifikation
Challenge
Generierung
Challenge
Response
Challenge
DTBS
Wissensnachweis
DTBS
SD
Challenge-
Empfang
Service
Provider
Challenge
Service
Challenge
Bindungs-
Verifikation
Wissens-
erfassung
User Client
OK/NOK
DTBS
Wissensnachweis
Interaktion
Response-
Berechnung
DTBS-Viewer
DTBS
Benutzer bzw.
Benutzerin
Mobiles Endnutzergerät



Neben den bereits von dem in Abbildung 7 gezeigten verfeinerten Modell bekannten Komponenten enthält das in Abbildung 8 gezeigte Modell einer kompletten Signaturlösung noch weitere Komponenten. So sind auch jene Komponenten angeführt, über die der Authentifizierungsfaktor Wissen abgedeckt wird. Konkret sind dies die lokale Komponente *Wissenserfassung* und die zentrale Komponente *Wissensüberprüfung* . Erstere ermöglicht der Benutzerin bzw. dem Benutzer einen Wissensnachweis zu erbringen. Dies kann beispielsweise ein geheimes Passwort sein. Die Komponente *Wissenserfassung* leitet diesen Wissensnachweis zur Überprüfung an die *Zentrale Applikation*, konkret an die Komponente *Wissensüberprüfung* weiter. Das Resultat der Überprüfung des bereitgestellten Wissensnachweises geht in die Autorisierung der Transaktion, d.h. des Signaturerstellungsprozesses, ein.

Neben Komponenten zur Implementierung des Authentifizierungsfaktors Wissen definiert das in Abbildung 8 gezeigte Modell auch noch die Komponenten *Service Provider* und *User Client* . Die Komponente *Service Provider* implementiert einen Dienst, d.h. ein Service, der von der Benutzerin bzw. vom Benutzer konsumiert werden kann. Darüber hinaus ist der *Service Provider* jene Komponente, die einen Signaturerstellungsprozess in der *Zentralen Applikation* anstößt. Dementsprechend definiert diese Komponente einerseits die zu signierenden Daten (DTBS) und stellt andererseits auch den Empfänger der erstellten Signatur (Signed Data – SD) dar. Damit implementiert diese Komponente auch die Funktion der Relying Party, die üblicherweise als Empfänger erstellter Signaturen fungiert. Aus Gründen der Übersichtlichkeit wurden die Funktionen der Definition der DTBS und des Empfangs der SD in einer Komponente, nämlich der Komponente *Service Provider*, kombiniert. Obwohl als integrale Komponente des *Mobilen Endnutzergeräts* dargestellt, kann der *Service Provider* theoretisch auch in einer eigenen Domain, beispielsweise als Web-Applikation, umgesetzt werden. Die Lokation des *Service Providers* hat jedoch nur geringe Auswirkungen auf den Signaturerstellungsprozess an sich. Aus Gründen der Übersichtlichkeit werden diese Varianten durch das Modell in Abbildung 8 daher nicht abgebildet.

Neben der Komponente *Service Provider* definiert das in Abbildung 8 gezeigte Modell auch noch die Komponente *User Client* . Diese fungiert als Bindeglied zwischen dem *Service Provider* und der Benutzerin bzw. dem Benutzer und ermöglicht einen Zugriff auf Dienste, die vom *Service Provider* angeboten werden. Diese Komponente ist vor allem dann von Bedeutung, wenn der *Service Provider* in einer externen Domain implementiert ist. Ist der *Service Provider* beispielsweise als Web-Applikation umgesetzt, wird die lokale Komponente *User Client* in der Regel durch einen Web Browser implementiert.

Entsprechend dem in Abbildung 8 dargestellten Modell, besteht ein typischer Signaturerstellungsprozess aus den folgenden Schritten:

1. Die Benutzerin bzw. der Benutzer interagiert mit dem *User Client* um einen vom *Service Provider* angebotenen Dienst zu konsumieren.

2. Im Zuge der Konsumierung dieses Dienstes benötigt der *Service Provider* von der Benutzerin bzw. vom Benutzer eine elektronische Signatur. Dazu sendet er einen Signaturerstellungs-Request an die *Zentrale Applikation*, konkret an die Komponente *Signaturverarbeitung*, die dafür ein entsprechendes Interface zur Verfügung stellt. Dieser Request enthält unter anderem die zu signierenden Daten (DTBS).

3. Bevor die Signaturerstellung im *SSCD* durchgeführt wird, muss die Benutzerin bzw. der Benutzer authentifiziert werden. Erst eine erfolgreiche Authentifizierung autorisiert den Signaturerstellungsprozess in der *Zentralen Applikation* . Um die Benutzerin bzw. den Benutzer zu authentifizieren, fordert die *Zentrale Applikation* zunächst einen Wissensnachweis von der Benutzerin bzw. vom Benutzer ein. Dieser Wissensnachweis wird von der Benutzerin bzw. vom Benutzer über die Komponente *Wissenserfassung* bereitgestellt und an die zentrale Komponente *Wissensüberprüfung* übergeben. Die Komponente *Wissensüberprüfung* verifiziert den erhaltenen Wissensnachweis und leitet das Resultat der Verifikation an die Komponente *Autorisierung* weiter.

4. Der nachfolgende Authentifizierungsschritt deckt den Faktor Besitz ab. Dazu generiert die Subkomponente *Challenge-Generierung* eine zufällige Challenge und sendet diese an das *Mobile Endnutzergerät*, wo diese von der Subkomponente *Challenge-Empfang* entgegengenommen wird.

5. Gleichzeitig sendet die Komponente *Signaturverarbeitung* die über den Signatur-Request erhaltenen DTBS an die lokale Komponente *DTBS-Viewer*, über die die Benutzerin bzw. der Benutzer diese Daten einsehen und kontrollieren kann.

6. Die lokale Subkomponente *Bindungs-Verifikation* des *Mobilen Endnutzergeräts* erhält von der *Viewer* -Komponente die DTBS und von der Subkomponente *Challenge-Empfang* die aktuelle Challenge. Mit diesen Daten überprüft die Subkomponente *Bindungs-Verifikation* die korrekte Bindung zwischen angezeigten DTBS und erhaltener Challenge.

7. Ist diese Überprüfung positiv, erstellt die Subkomponente *Response-Berechnung* aus der erhaltenen Challenge die zugehörige Response. Dazu wird die Challenge von der lokalen Subkomponente *Challenge-Empfang* bezogen.

8. Die berechnete Response wird an die Subkomponente *Response-Verifikation* der *Zentralen Applikation* gesendet. Diese verifiziert die erhaltene Response. Das Ergebnis der Überprüfung wird an die Subkomponente *Autorisierung* weitergeleitet.

9. Zusätzlich leitet die Subkomponente *Response-Verifikation* die erhaltene Response an die zentrale Subkomponente *Bindungs-Verifikation* weiter. Zusätzlich werden auch die DTBS an diese Subkomponente übertragen. Damit ist die Subkomponente *Bindungs-Verifikation* in der Lage, die Bindung zwischen aktuellen DTBS und erhaltener Response zu überprüfen. Das Ergebnis dieser Überprüfung wird ebenfalls an die Subkomponente *Autorisierung* weitergeleitet.

10. Die Subkomponente *Autorisierung* analysiert alle erhaltenen Verifikationsergebnisse. Sind alle positiv, wird die Signaturerstellung im *SSCD* autorisiert.

11. Das Resultat der Signaturerstellung (SD) wird vom *SSCD* an die Komponente *Signaturverarbeitung* übermittelt.

12. Die Komponente *Signaturverarbeitung* erstellt aus den vom *SSCD* erhaltenen SD eine entsprechende Signaturerstellungs-Response. Diese Response wird dem *Service Provider* als Antwort auf den Signaturerstellungs-Request retourniert.

Das in Abbildung 8 gezeigt Modell einer serverbasierten Signatur-Lösung zeigt, dass der erarbeitete Ansatz zur Implementierung des Authentifizierungsfaktors Besitz einfach auf konkrete Anwendungsbereiche angewendet werden kann. Um dessen Umsetzbarkeit auch in der Praxis zu evaluieren, wurde der erarbeitete Ansatz außerdem im Rahmen eines Demonstrators prototypisch implementiert. Diese prototypische Implementierung wird im folgenden Abschnitt näher beschrieben.

6. Demonstrator

In Abschnitt 5.2 wurde bereits gezeigt, dass die erarbeitete Lösung zur Implementierung des Authentifizierungsfaktors Besitz trotz ihres abstrakten Charakters sehr wohl auf konkrete Anwendungsszenarien anwendbar ist. Dies wurde anhand einer serverbasierten Signaturlösung beispielhaft gezeigt. In diesem Abschnitt soll die Evaluierung der erarbeiteten Lösung nun abgeschlossen werden, indem diese über einen Demonstrator prototypisch umgesetzt wird.

Ausgangspunkt für die Erstellung eines Demonstrators ist das in Abbildung 8 gezeigte abstrakte Modell einer serverbasierten Signaturlösung. Dieses Modell wird in diesem Abschnitt zunächst weiter konkretisiert und in ein funktionales Modell übergeführt. Dazu werden diverse Designentscheidungen getroffen, um konkrete Umsetzungen der einzelnen durch das Modell definierten Komponenten festzulegen. Vom resultierenden funktionalen Modell werden in weiterer Folge Architektur und Prozessfluss des Demonstrators abgeleitet. Dessen Umsetzung wird schließlich über Screenshots veranschaulicht.

6.1. Designentscheidungen

Um das in Abbildung 8 dargestellte Modell einer serverbasierten Signaturlösung weiter zu konkretisieren und in ein funktionales Modell überzuführen, müssen einige Designentscheidungen getroffen werden. Diese definieren und spezifizieren die konkrete Umsetzung einzelner durch das zugrundeliegende Modell definierter Komponenten. Die für die Entwicklung des funktionalen Modells getroffenen Designentscheidungen werden in den folgenden Unterabschnitten erläutert.

6.1.1. Umsetzung von Wissensnachweisen

Obwohl bei der Erstellung des in Abbildung 8 gezeigten Modells der Fokus klar auf die Implementierung des Authentifizierungsfaktors Besitz gelegt wurde, muss für die Umsetzung einer Mehrfaktorauthentifizierung auch ein zweiter Authentifizierungsfaktor berücksichtigt werden. Abbildung 8 zeigt, dass im erarbeiteten Modell dafür der Faktor Wissen gewählt wurde. Zur Entwicklung eines funktionalen Modells muss die Implementierung dieses Faktors weiter konkretisiert werden.

Verschiedene Möglichkeiten, den Authentifizierungsfaktor Wissen umzusetzen, wurden bereits in Abschnitt 2.1.1 erläutert. Neben herkömmlichen alphanumerischen Passwörtern wurden in letzter Zeit vor allem auch Ansätze basierend auf graphischen Passwörtern als Alternative vorgestellt. Da sich diese bisher jedoch nicht auf breiter Basis durchsetzen konnten und ebenso wie alphanumerische Passwörter diverse Nachteile aufweisen, wurde entschieden, für die Erstellung des funktionalen Modells und damit in weiterer Folge auch für die Umsetzung des Demonstrators auf alphanumerische Passwörter zurückzugreifen, um den Authentifizierungsfaktor Wissen zu implementieren.

Unter Berücksichtigung dieser Designentscheidung können die Komponenten *Wissenserfassung* und *Wissensüberprüfung* des in Abbildung 8 gezeigten Modells weiter konkretisiert werden. Die Aufgabe der Komponente *Wissenserfassung* ist im Wesentlichen die Abfrage eines geheimen alphanumerischen Passworts von der Benutzerin bzw. vom Benutzer. Dementsprechend kann diese Komponente durch die Komponente *Passworteingabe* ersetzt werden. Durch die Festlegung auf alphanumerische Passwörter zur Implementierung des Authentifizierungsfaktors Wissen kann auch die zentrale Komponente *Wissensüberprüfung* entsprechend konkretisiert werden. Aufgabe dieser Komponente ist es, das von der Benutzerin bzw. vom Benutzer eingegebene Passwort zu verifizieren. Dementsprechend kann diese Komponente durch die Komponente *Passwortüberprüfung* ersetzt werden.

6.1.2. Umsetzung und Übertragung der Challenge

Die Umsetzung der Challenge und der Komponenten, die unmittelbar in deren Übertragung und Verarbeitung involviert sind, hängt zu einem großen Teil von der Technologie ab, die für die Übermittlung der Challenge zwischen *Zentraler Applikation* und *Mobilem Endnutzergerät* genutzt wird. Für diese Übertragung kommen prinzipiell zwei Möglichkeiten in Frage. Die Challenge kann entweder über eine Internetverbindung übertragen werden. Alternativ kann auch das mobile Netzwerk genutzt und die Challenge beispielsweise über SMS übertragen werden. Andere auf mobilen Endnutzergeräten verfügbare Kommunikationstechnologien wie NFC oder Bluetooth sind primär für kurze Reichweiten und Peer-to-Peer-Kommunikation konzipiert und damit für eine Übertragung der Challenge über potentiell lange Distanzen nicht geeignet.

Unter Berücksichtigung der beiden verfügbaren Alternativen wurde für die Entwicklung des Demonstrators entschieden, auf SMS-Technologie zur Übertragung von Challenges zurückzugreifen. Konkret werden Challenges durch den Demonstrator in Form von TANs, die über SMS an das *Mobile Endnutzergerät* übertragen werden, implementiert. Das mag auf den ersten Blick überraschend sein, da für das SMS-TAN-Verfahren in Abschnitt 4.4 einige Nachteile identifiziert wurden. Allerdings gibt es einen relevanten Unterschied zwischen dem SMS-TAN-Verfahren und dem durch den Demonstrator verfolgten Ansatz. Beim SMS-TAN-Verfahren ist die Vertraulichkeit der über SMS übermittelten Daten, d.h. der TAN, von zentraler Bedeutung, da diese den Besitznachweis repräsentiert. Dies ist in dem durch den Demonstrator verfolgten

Challenge-Response-Ansatz nicht der Fall, da hier die Challenge lediglich zur Berechnung einer Response herangezogen wird. Kenntnis der Challenge alleine ermöglicht einer Angreiferin oder einem Angreifer noch nicht, einen gültigen Besitznachweis zu erbringen.

Während die Verwendung von SMS-TANs als Challenges also aus sicherheitstechnischer Sicht nicht nachteilig ist, ergeben sich durch diesen Ansatz darüber hinaus einige Vorteile. So wird durch Verwendung der SMS-Technologie und des mobilen Netzwerks ein zusätzlicher Kommunikationskanal in das Gesamtsystem eingebracht. Zweitens sind Benutzerinnen und Benutzer die Verwendung von SMS-TANs aufgrund der langen Tradition von herkömmlichen SMS-TAN-Verfahren gewohnt. Eine Verwendung von SMS-TANs zur Implementierung eines Challenge-Response-Ansatzes erleichtert somit den Umstieg auf dieses neue Verfahren, da bekannte Technologien und Mechanismen genutzt werden.

Unter Berücksichtigung der getroffenen Designentscheidung, Challenges über SMS-TANs zu implementieren, können einige der durch das in Abbildung 8 gezeigte Modell definierte Komponenten näher spezifiziert werden. Konkret kann die zentrale Komponente *Challenge- Generierung* in die Komponente *TAN-Generierung* übergeführt werden. Entsprechend kann auch die lokale Komponente *Challenge-Empfang* durch die Komponente *SMS-Empfang* ersetzt werden.

6.1.3. Umsetzung der lokalen Bindungs-Verifikation

Die Überprüfung der Bindung zwischen Transaktionsdaten, d.h. DTBS im konkreten Fall von Signaturlösungen, und erbrachten Besitznachweisen ist eine zentrale Funktion sowohl in der *Zentralen Applikation* als auch am *Mobilen Endnutzergerät* . Am *Mobilen Endnutzergerät* muss dazu konkret die Bindung der erhaltenen Challenge, d.h. der SMS-TAN, an die angezeigten DTBS sichergestellt werden. Diese Funktionalität kann entweder über eine technische Komponente oder aber manuell durch die Benutzerin bzw. den Benutzer umgesetzt werden.

Bei einer technischen Umsetzung muss die TAN direkt aus den DTBS abgeleitet werden, um eine automatische Verifikation der Bindung zwischen DTBS und TAN zu ermöglichen. Soll die notwendige Bindung hingegen von der Benutzerin bzw. vom Benutzer manuell überprüft werden, muss ein geeigneter Mechanismus implementiert werden, der diese Überprüfung erlaubt. Ein derartiger Mechanismus kommt beispielsweise beim SMS-TAN-Verfahren zum Einsatz. Hier wird zusätzlich zur TAN ein sogenannter Referenzwert generiert. Dieser wird zusammen mit der TAN an das *Mobile Endnutzergerät* übertragen und gleichzeitig auch mit den Transaktionsdaten angezeigt. Dadurch ist es der Benutzerin bzw. dem Benutzer möglich, durch Vergleich der angezeigten Referenzwerte die Bindung zwischen TAN und Transaktionsdaten zu überprüfen.

Aus Gründen der Einfachheit wurde entschieden, für die Umsetzung des Demonstrators auf eine manuelle Überprüfung der Bindung zwischen DTBS und erhaltener Challenge zu vertrauen. Dementsprechend müssen einige Komponenten des in Abbildung 8 gezeigten Modells adaptiert werden, um die Verwendung des notwendigen zusätzlichen Referenzwerts entsprechend abzubilden. Beispielsweise muss die Komponente *TAN-Generierung* in der Lage sein, zusätzlich zur TAN auch geeignete Referenzwerte zu generieren.

6.1.4. Erstellung der Response

Die entscheidende Komponente in Bezug auf die Erstellung von Besitznachweisen ist die lokale Komponente *Response-Berechnung* . Diese berechnet aus der erhaltenen Challenge unter Verwendung eines benutzerspezifischen Geheimnisses eine eindeutige Response, die schließlich den Besitznachweis repräsentiert. Challenge-Response-Ansätze verwenden dazu üblicherweise geeignete kryptographische Methoden. Entsprechend wird das benutzerspezifische Geheimnis über einen geheimen kryptographischen Schlüssel repräsentiert.

Prinzipiell können sowohl symmetrische als auch asymmetrische kryptographische Methoden verwendet werden, um aus erhaltenen Challenges entsprechende Responses zu erstellen. Aufgrund ihrer Vorteile speziell in Bezug auf notwendige Schlüsselaustauschmechanismen wurde entschieden, für den Demonstrator auf asymmetrische Verfahren zurückzugreifen. Konkret werden erhaltene Challenges, d.h. SMS-TANs, durch die Komponente *Response-Berechnung* signiert, um entsprechende Responses zu erzeugen. Dementsprechend kann die generische Komponente *Response-Berechnung* durch die Komponente *TAN-Signator* ersetzt werden.

Auf Seiten der *Zentralen Applikation* ist entsprechend dem in Abbildung 8 dargestellten Modell die Komponente *Response-Verifikation* für die Überprüfung erhaltener Responses verantwortlich. Unter Berücksichtigung der Entscheidung, Responses über signierte TANs zu implementieren, kann diese Komponente weiter konkretisiert und durch die spezifischere Komponente *Response-Verifikation* ersetzt werden.

6.2. Funktionales Modell

Mit Hilfe der getroffenen Designentscheidungen kann aus dem in Abbildung 8 gezeigten Modell einer serverbasierten Signaturlösung ein vollständiges funktionales Modell erstellt werden. Dieses ist in Abbildung 9 dargestellt. Das funktionale Modell verfeinert und konkretisiert die Umsetzung einzelner Komponenten des zugrundeliegenden Modells entsprechend der getroffenen Designentscheidungen. Komponenten, die im Gegensatz zu dem in Abbildung 8 gezeigten Modell adaptiert wurden, sind in Abbildung 9 farblich hervorgehoben. Dies betrifft folgende Komponenten:

- **Passworteingabe** : Diese Komponente ersetzt die abstrakte Komponente *Wissenserfassung* und berücksichtigt die Entscheidung, den Faktor Wissen über Passwörter abzudecken.
- **Passwortüberprüfung:** Diese Komponente ersetzt die abstrakte Komponente *Wissensüberprüfung* und berücksichtigt die Entscheidung, den Faktor Wissen über Passwörter abzudecken.
- **TAN-Generierung:** Diese Komponente ersetzt die abstrakte Komponente *Challenge- Generierung* und berücksichtigt die Entscheidung, Challenges durch SMS-TANs zu implementieren.
- **SMS-Empfang:** Diese Komponente ersetzt die abstrakte Komponente *Challenge-Empfang* und berücksichtigt die Entscheidung, Challenges und auch Referenzwerte über SMS-Technologie zu übermitteln.
- **TAN-Vergleich:** Diese Komponente ersetzt die abstrakte Komponente *Bindungs-Verifikation* und berücksichtigt die Entscheidung, Challenges durch SMS-TANs zu implementieren.

- **DTBS-Provider:** Diese Komponente wurde aufgrund der Notwendigkeit eines zusätzlichen Referenzwerts neu hinzugefügt. Die Komponente *DTBS-Provider* kombiniert den Referenzwert der aktuellen Transaktion mit den zugehörigen DTBS.
- **DTBS-Viewer:** Die Komponente *DTBS-Viewer* wurde dahingehend erweitert, dass diese nun auch in der Lage ist, Referenzwerte zusammen mit DTBS anzuzeigen.
- **TAN-Signator:** Diese Komponente ersetzt die abstrakte Komponente *Response- Berechnung* und berücksichtigt die Entscheidung, Responses durch Signieren der erhaltenen SMS-TAN zu erzeugen.
- **Signatur-Verifikation:** Diese Komponente ersetzt die abstrakte Komponente *Response- Verifikation* und berücksichtigt die Entscheidung, Responses durch Signieren der erhaltenen SMS-TAN zu erzeugen.

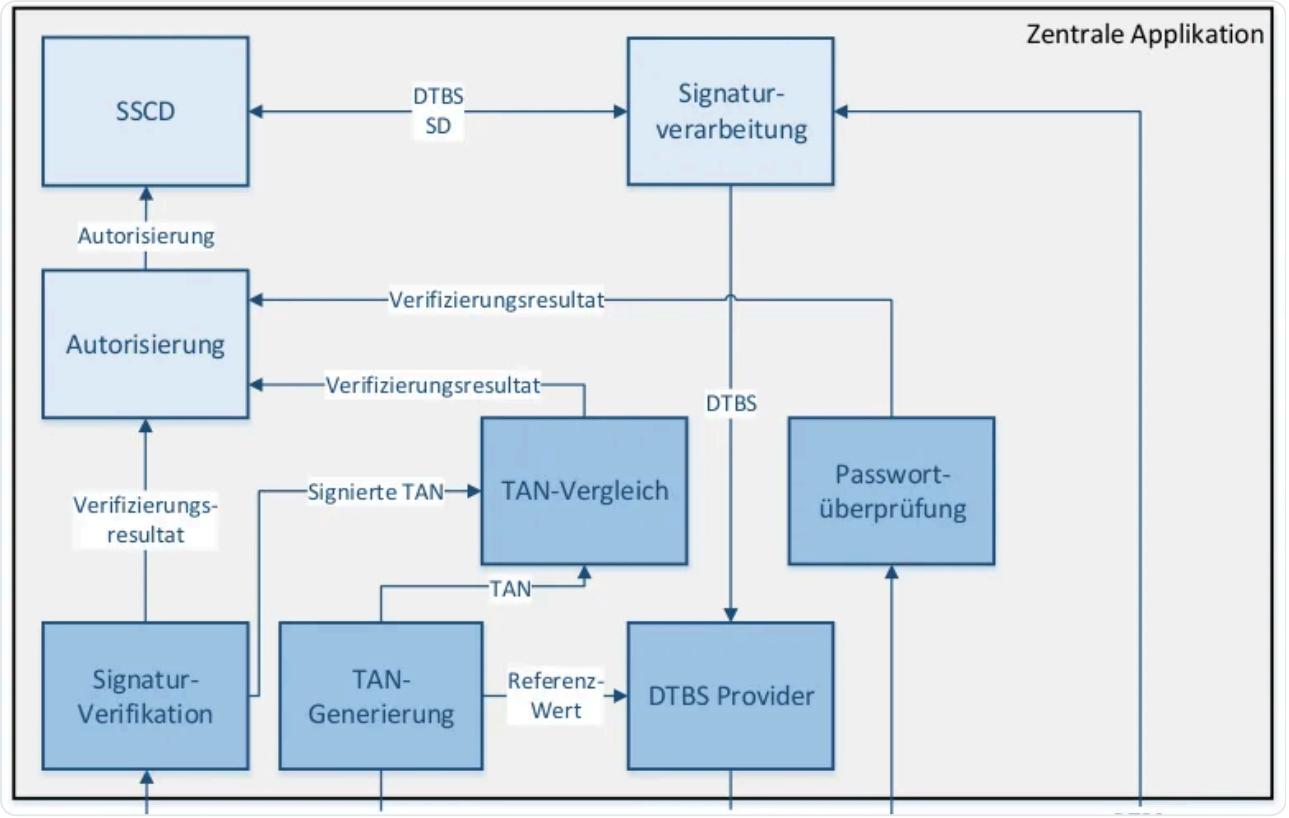
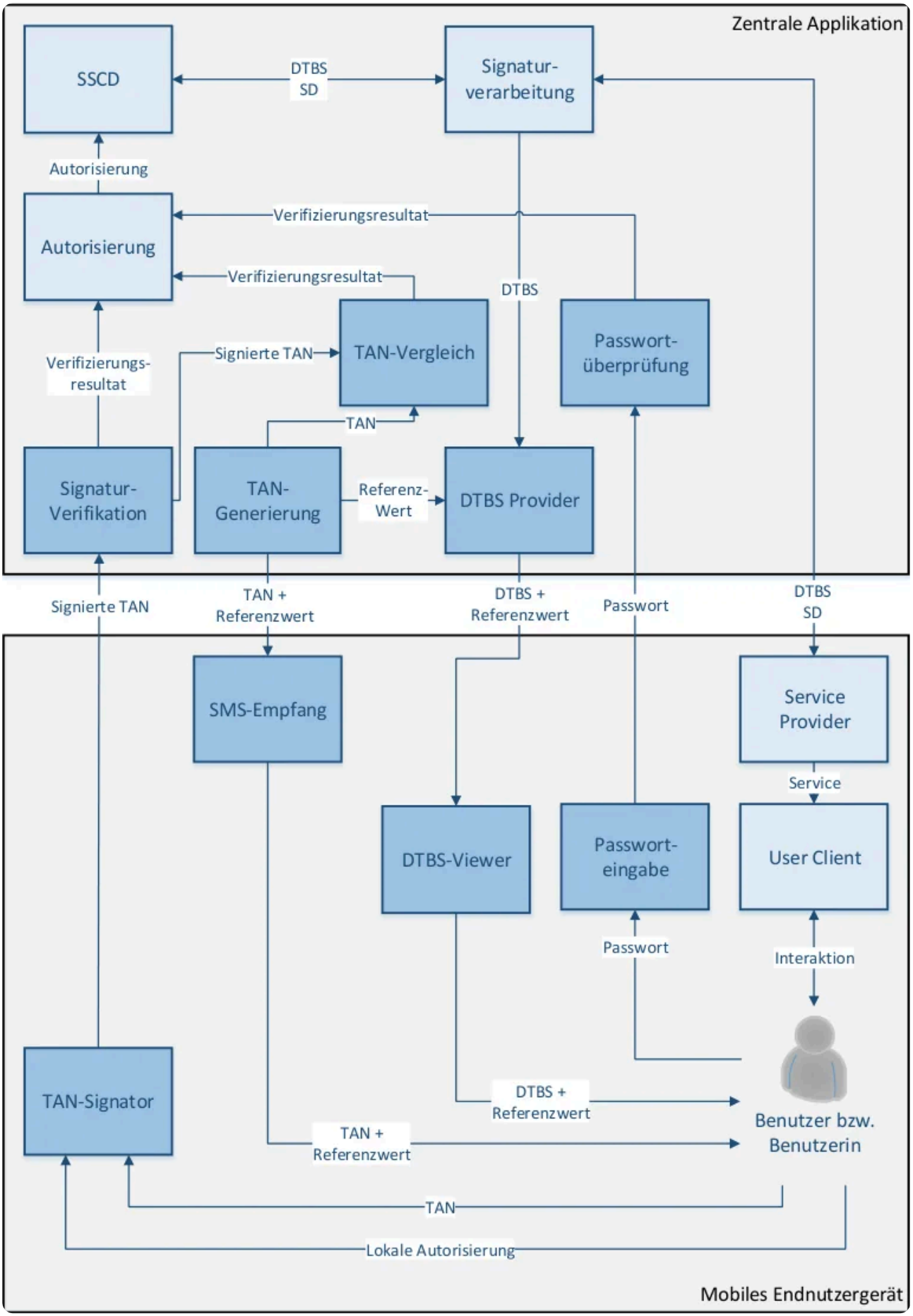
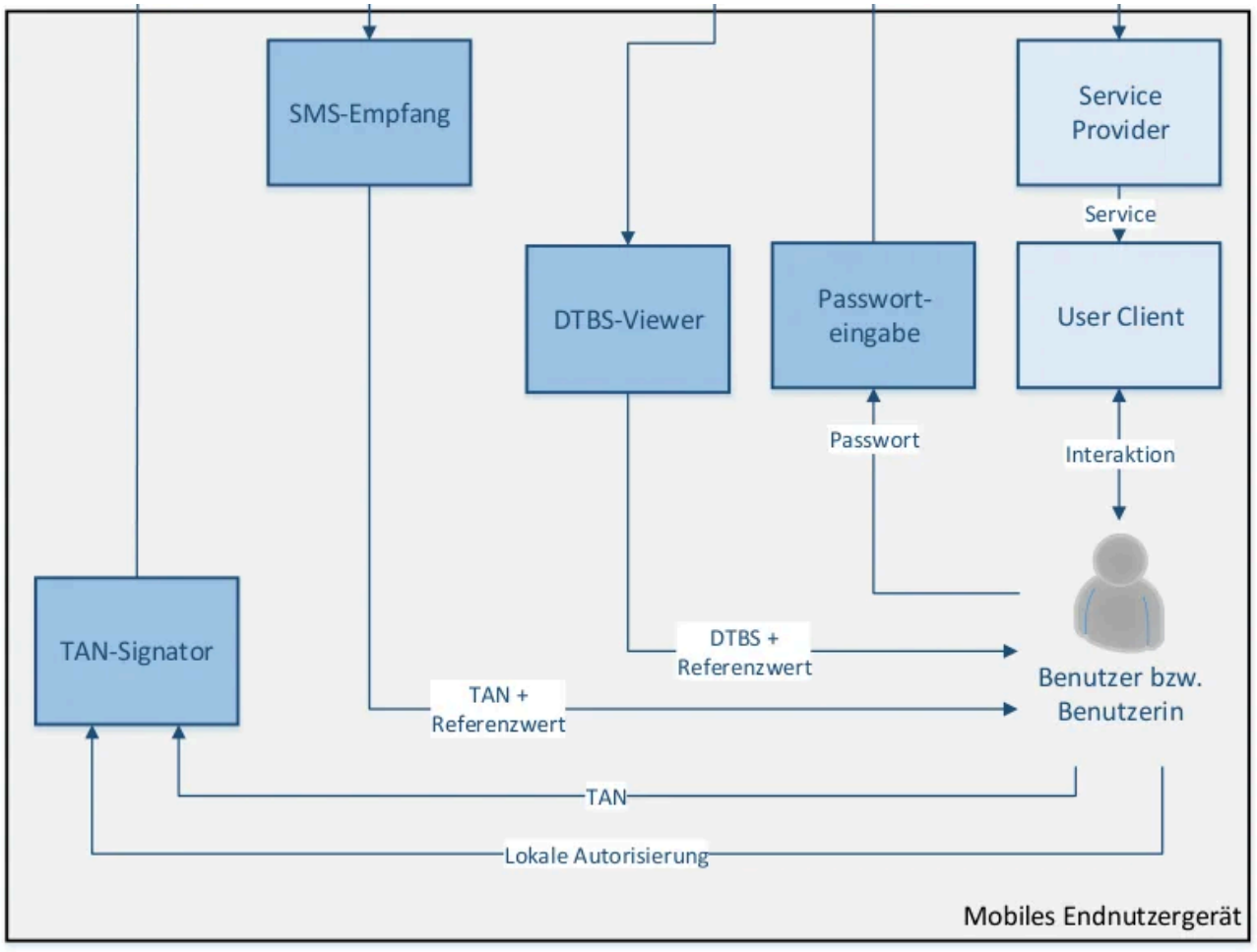


Abbildung 9. Funktionales Modell des Demonstrators.


Zentrale Applikation
SSCD
DTBS SD
Signatur-verarbeitung
Autorisierung
Autorisierung
Verifizierungsresultat
Verifizierungsresultat
DTBS
Verifizierungs-resultat
Signierte TAN
TAN-Vergleich
Passwort-überprüfung
TAN
Signatur-Verifikation
TAN-Generierung
Referenz-Wert
DTBS Provider
Signierte TAN
TAN + Referenzwert
DTBS + Referenzwert
Passwort
DTBS SD
SMS-Empfang
Service Provider
Service
DTBS-Viewer
Passwort-eingabe
User Client
Passwort
Interaktion
TAN-Signator
DTBS + Referenzwert
TAN + Referenzwert
Benutzer bzw. Benutzerin
TAN
Lokale Autorisierung
Mobiles Endnutzergerät



6.3. Architektur

Das in Abbildung 9 gezeigte funktionale Modell spezifiziert die Funktionen der einzelnen Komponenten, definiert jedoch nicht deren konkrete Umsetzung unter Verwendung aktuell auf mobilen Endnutzergeräten verfügbaren Technologien. Um das in Abbildung 9 dargestellte funktionale Modell weiter in Richtung einer konkreten Implementierung zu entwickeln, wurde aus dem funktionalen Modell eine konkrete Architektur abgeleitet. Dafür wurden aktuell verfügbare Technologien analysiert und entsprechend berücksichtigt. Die resultierende Architektur der serverbasierten Signaturlösung für mobile Endnutzergeräte, die durch den entwickelten Demonstrator implementiert wurde, ist in Abbildung 10 dargestellt.

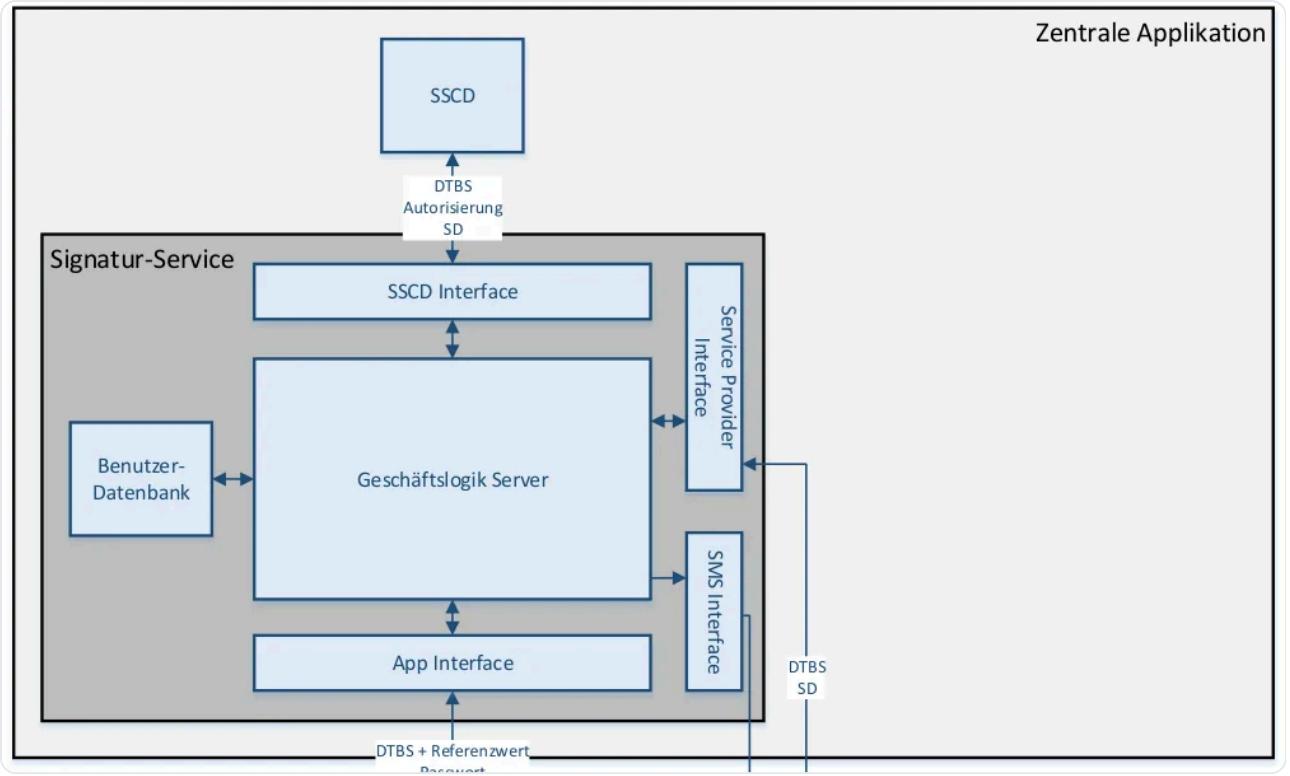
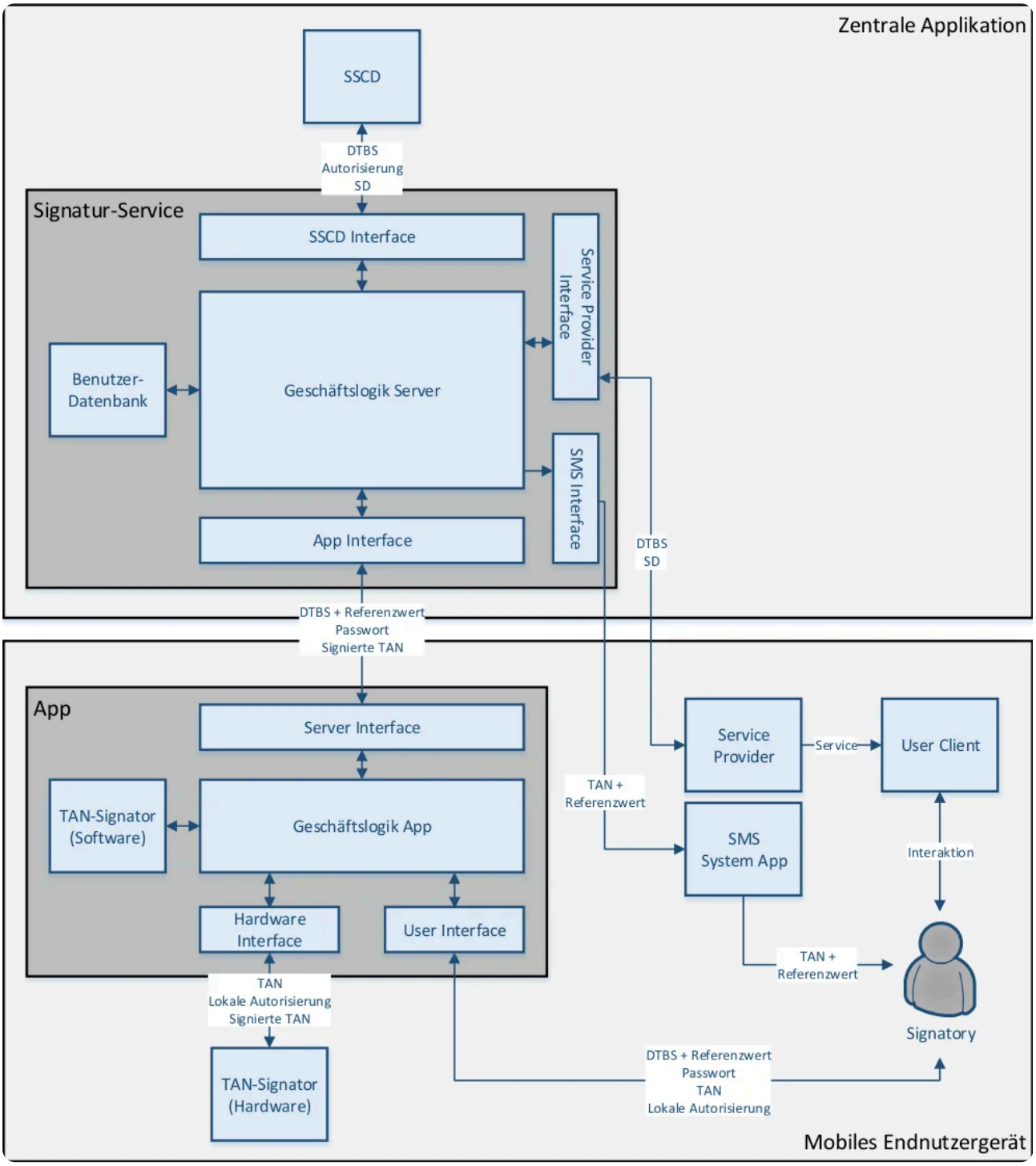
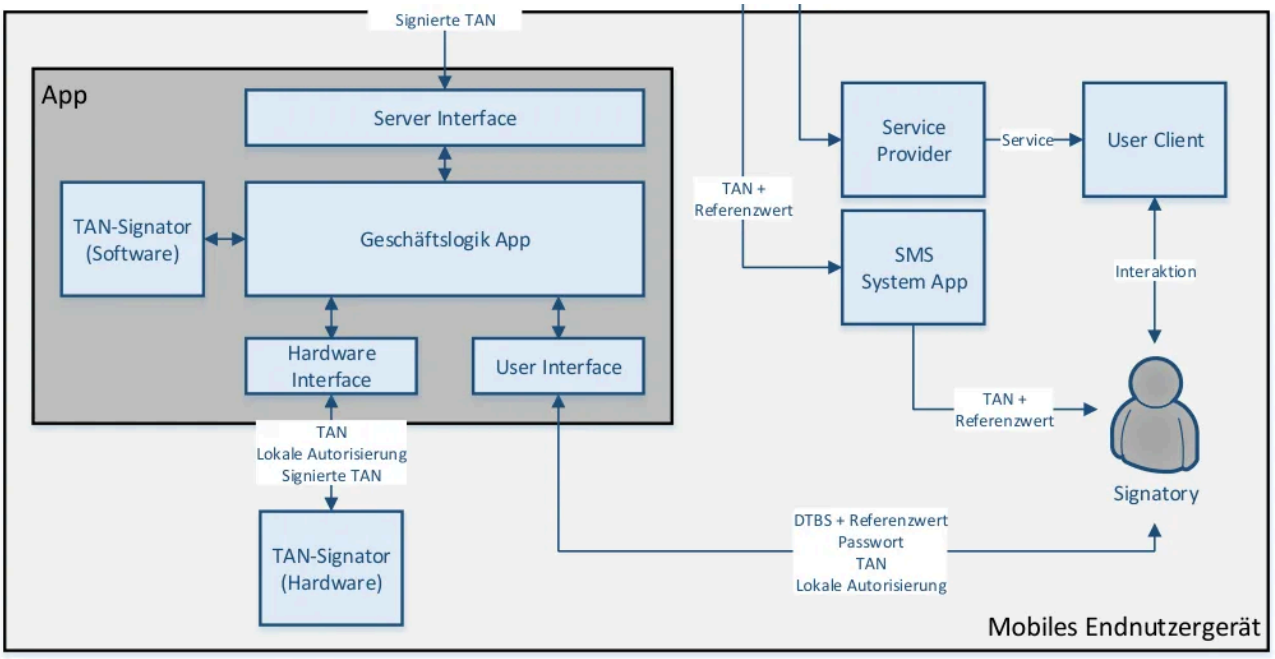


Abbildung 10. Architektur des Demonstrators. Aus Abbildung 10 geht hervor, dass die dem Demonstrator zugrundeliegende Architektur auf zwei Kernkomponenten beruht. Die Kernkomponente Signatur-Service implementiert im Wesentlichen die Funktionalität der Zentralen Applikation . Ausgenommen ist hier nur das SSCD, da dieses in der Regel in Hardware implementiert werden muss. Die zweite Kernkomponente App implementiert einen Großteil der Funktionalität, die am Mobilen Endnutzergerät umgesetzt werden muss. Ausgenommen sind hier die Komponenten Service Provider und User Client, die keine integralen Bestandteile der Signaturerstellungsapplikation selbst sind. Auch die Komponente SMS-Empfang wird außerhalb der Kernkomponente App implementiert. Dem liegt die Tatsache zugrunde, dass auf einigen mobilen Betriebssystemen Apps nicht in der Lage sind, SMS-Nachrichten zu empfangen. Diese Funktionalität wird daher über die SMS-System-App des Mobilen Endnutzergeräts abgedeckt. Einen Sonderfall nimmt die Komponente TAN-Signator ein. Diese wurde bereits im funktionalen Modell bewusst abstrakt gehalten, um verschiedene Umsetzungen zu ermöglichen. Um diese Flexibilität beizubehalten, berücksichtigt die in Abbildung 10 gezeigte Architektur sowohl eine softwarebasierte als auch eine hardwarebasierte Variante dieser Komponente. Die hardwarebasierte Variante muss naturgemäß ebenfalls außerhalb der softwarebasierten Kernkomponente App implementiert werden.


Zentrale Applikation
SSCD
DTBS
Autorisierung
SD
Signatur-Service
SSCD Interface
Service Provider Interface
Benutzer-Datenbank
Geschäftslogik Server
SMS Interface
App Interface
DTBS
SD
DTBS + Referenzwert
Passwort
Signierte TAN
App
Server Interface
TAN-Signator (Software)
Geschäftslogik App
Hardware Interface
User Interface
TAN
Lokale Autorisierung
Signierte TAN
TAN-Signator (Hardware)
Service Provider
Service
User Client
TAN + Referenzwert
SMS System App
Interaktion
TAN + Referenzwert
Signatory
DTBS + Referenzwert
Passwort
TAN
Lokale Autorisierung
Mobiles Endnutzergerät



Für beide Kernkomponenten spezifiziert die in Abbildung 10 gezeigte Architektur diverse Subkomponenten. Für die Kernkomponente *Signatur-Service* implementiert die Subkomponente *Geschäftslogik Server* einen Großteil der Funktionalität. Daneben existiert noch eine Reihe weiterer Subkomponenten, die im Wesentlichen geeignete Schnittstellen zu externen Komponenten wie dem *SSCD* oder der *App* am *Mobilen Endnutzergerät* implementieren. Außerdem enthält die Kernkomponente *Signatur-Service* eine *Benutzerdatenbank* . Diese ist notwendig, da handelsübliche HSMs, die typischerweise für die Umsetzung zentraler SSCDs verwendet werden, nur eine sehr begrenzte Anzahl an Schlüssel gleichzeitig speichern können. Aktuell nicht benötigte Schlüssel müssen daher entsprechend geschützt in der *Benutzerdatenbank* abgelegt werden. Eine Möglichkeit, die Sicherheit und Vertraulichkeit dieser Schlüssel jederzeit sicherzustellen wurde von Orthacker et al. [24] diskutiert.

Auch für die Kernkomponente *App* definiert die in Abbildung 10 gezeigte Architektur diverse Subkomponenten. Auch hier wird ein Großteil der Funktionalität durch die Kernkomponente *Geschäftslogik App* abgedeckt. Daneben kommen diverse Interface-Komponenten zur Anwendung, über die ein Datenaustausch mit externen Komponenten und der Benutzerin bzw. dem Benutzer ermöglicht wird.

Entsprechend der in Abbildung 10 gezeigte Architektur besteht der entwickelte Demonstrator aus einer zentralen Serverkomponente und einer lokalen Smartphone-App. Zusammen implementieren diese Komponenten eine Signaturlösung, die auf einem einzelnen *Mobilen Endnutzergerät* ohne Zuhilfenahme eines zusätzlichen Geräts verwendet werden kann. Der Prozessfluss eines typischen Signaturerstellungsprozesses wird im Folgenden näher beschrieben.

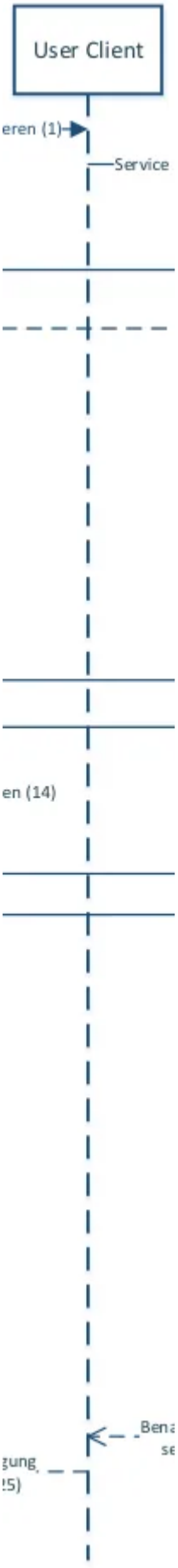
6.4. Prozessfluss

Der Prozessfluss eines typischen Signaturerstellungsprozesses ist in Abbildung 11 dargestellt. Dazu identifiziert Abbildung 11 die einzelnen Komponenten, die in diesen Prozess involviert sind. Diese Komponenten interagieren im Zuge eines Signaturerstellungsprozesses miteinander. Diese Interaktion wird ebenfalls durch Abbildung 11 abgebildet. Die einzelnen Kommunikationsschritte werden zudem nachfolgend beschrieben.

Zu beachten ist hier, dass der in Abbildung 11 dargestellte und nachfolgend beschriebene Prozessfluss keine Fehlerfälle abdeckt. Es wird hier aus Gründen der Übersichtlichkeit nur jener Pfad betrachtet, indem alle Komponenten wie vorgesehen funktionieren und sich auch die Benutzerin bzw. der Benutzer korrekt verhalten. Konkret werden beispielsweise jene Fälle nicht betrachtet, in denen durch die Benutzerin oder den Benutzer ein falsches Passwort oder eine falsche TAN eingegeben wird. Nichtsdestotrotz müssen auch diese Fälle berücksichtigt werden, wenn die in Abbildung 10 gezeigte Architektur in Rahmen einer konkreten Umsetzung realisiert wird.

Zu beachten ist auch, dass sich der in Abbildung 11 gezeigte Prozessfluss rein auf den Signaturerstellungsprozess bezieht. In allen Signaturlösungen muss diesem ein einmaliger Registrierungsprozess vorausgehen, im Zuge dessen u.a. die Signaturschlüssel der Benutzerin bzw. des Benutzers generiert und nötige Authentifizierungsdaten festgelegt werden. Da dieser Prozess nur einmalig durchlaufen werden muss, kann dieser auch durch nicht-technische Maßnahmen umgesetzt werden. Daher wird das Hauptaugenmerk an dieser Stelle auf den Signaturerstellungsprozess gelegt.


Benutzerin/
Benutzer
Service
n und Referenzwerte
Ben
an


User Client
eren (1)
Service
en (14)
Ben
se
gung
!5)


Service Provider
eren (1)
Signa
Passwo
Passwo
DTBS und Refe
TAN
[Lokale Autorisi
Signatur
gung,
4)


Signatur-
Service
st (2)
lern (4)
ben (5)
Schlüss
DT
d Referenzwert generi
t anzeigen (12)
d Referenzwert anzeig
(15)
reitstellen (16)
ierte TAN überprüfen
Signa
autc
Signatur
eren (23)


SSCD
Passwort anfordern (3)
asswort bereitstellen (
ort und
(7)
urschlüssel entschlüss
TAN und Refe
nd Referenzwert send
ierte TAN bereitstellen
llung
(20)
Signatur erstelle
eren (22)

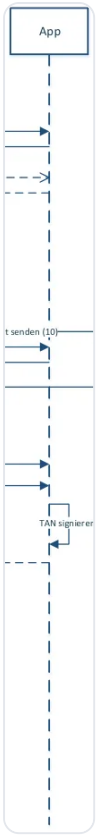


Abbildung 11. Prozessfluss eines typischen Signaturerstellungsprozesses. Unter der Annahme eines optimalen Ablaufs gliedert sich ein typischer Signaturerstellungsprozess in die folgenden Schritte:


SMS System
App

Service konsumieren (1)	Service konsumieren (1)	Signaturrequest (2)	Col4	nfordern (3)	0)
~~Service konsumieren (1)~~	~~Service konsumieren (1)~~	~~Signaturrequest (2)~~	~~Passwort a~~	~~Passwort a~~	~~Passwort a~~
~~Service konsumieren (1)~~		~~Passwort anfordern (4)~~			
renzwerte vergleichen (14)		~~Passwort eingeben (5)~~ TAN und Referenz			
renzwerte vergleichen (14)		~~Passwort eingeben (5)~~ TAN und Referenz	Schlüssel, Passwort und DTBS laden (7) ~~Passwort ber~~ Signaturschlüssel wert generieren (9)	~~ eitstellen (6)~~ entschlüsseln (8)	~~ eitstellen (6)~~ entschlüsseln (8)
renzwerte vergleichen (14)		~~Passwort eingeben (5)~~ TAN und Referenz	~~T~~	~~N und Referenzwert senden (~~ ~~ zwert senden (11)~~	~~N und Referenzwert senden (~~ ~~ zwert senden (11)~~
renzwerte vergleichen (14)		~~Passwort eingeben (5)~~ TAN und Referenz	~~DTBS und Referen~~	~~DTBS und Referen~~	~~DTBS und Referen~~
renzwerte vergleichen (14)	~~DT~~	~~S und Referenzwert anzeigen~~	~~ 12)~~		
renzwerte vergleichen (14)		~~TAN und Referenz~~	~~ wert anzeigen (13)~~	TA	N signieren (17)
		~~TAN eingeben (15)~~			
Benachrichtigung	~~[Lok~~ Benachrichtigung	~~ale Autorisierung bereitstellen~~ Signierte TAN ü	~~ 16)]~~	~~ 16)]~~	~~ 16)]~~
Benachrichtigung	~~[Lok~~ Benachrichtigung	~~ale Autorisierung bereitstellen~~ Signierte TAN ü	~~Signierte TAN b~~ Signaturerstellung autorisieren (20) Signa berprüfen (19)	~~ ereitstellen (18)~~ tur erstellen (21)	~~ ereitstellen (18)~~ tur erstellen (21)
Benachrichtigung	~~[Lok~~ Benachrichtigung	~~ale Autorisierung bereitstellen~~	~~Signatur retournieren (22)~~	~~Signatur retournieren~~	~~Signatur retournieren~~

Service konsumieren (1)	**Service konsumieren (1)**	**Signaturrequest (2)**	**Col4**	**nfordern (3)**	**0)**
		Signierte TAN ü		~~(22)~~	~~(22)~~
Benachrichtigung	~~[Lok~~ Benachrichtigung	~~Signatur retournieren (23)~~	~~Signatur retournieren (23)~~	~~Signatur retournieren (23)~~	~~Signatur retournieren (23)~~
Benachrichtigung	senden (24)	senden (24)	senden (24)	senden (24)	senden (24)
anzeigen (25)	anzeigen (25)	anzeigen (25)	anzeigen (25)	anzeigen (25)	anzeigen (25)

1. Die Benutzerin bzw. der Benutzer konsumiert über den *User Client* ein Service, das vom *Service Provider* bereitgestellt wird.
2. Der *Service Provider* benötigt eine Signatur von der Benutzerin bzw. vom Benutzer und schickt daher einen Signaturerstellungs-Request mit den DTBS an das zentrale *Signatur- Service* .
3. Das *Signatur-Service* fordert von der lokalen *App* das geheime Passwort der Benutzerin bzw. des Benutzers an.
4. Die *App* fordert die Benutzerin bzw. den Benutzer auf, das geheime Passwort einzugeben.
5. Die Benutzerin bzw. der Benutzer gibt das Passwort ein.
6. Die *App* sendet das Passwort an das *Signatur-Service* .
7. Das *Signatur-Service* holt den Schlüssel der Benutzerin bzw. des Benutzers aus der *Benutzerdatenbank*, und lädt diesen zusammen mit den DTBS in das *SSCD* .
8. Das *SSCD* bereitet die Signaturerstellung vor.
9. Das *Signatur-Service* generiert eine TAN und einen zugehörigen Referenzwert.
10. TAN und Referenzwert werden an die *SMS-System-App* des *Mobilen Endnutzergeräts* geschickt.
11. Das *Signatur-Service* schickt die DBTS zusammen mit dem Referenzwert an die *App* .
12. Die *App* stellt DTBS und Referenzwert für die Benutzerin bzw. den Benutzer dar.
13. Die *SMS-System-App* stellt TAN und Referenzwert für die Benutzerin bzw. den Benutzer dar.
14. Die Benutzerin bzw. der Benutzer vergleicht die beiden dargestellten Referenzwerte.
15. Die Benutzerin bzw. der Benutzer gibt die erhaltene TAN in der *App* ein.
16. Optional: Die *App* fordert von der Benutzerin bzw. dem Benutzer lokale Autorisierungsdaten an, die für eine Signatur der TAN notwendig sind.
17. Die *App* signiert die TAN.

18. Die *App* sendet die signierte TAN an das *Signatur-Service* .
19. Das *Signatur-Service* überprüft die signierte TAN.
20. Das *Signatur-Service* autorisiert die Signaturerstellung im *SSCD* .
21. Das *SSCD* erstellt die Signatur für die Benutzerin bzw. den Benutzer.
22. Das *SSCD* retourniert die erstellte Signatur an das *Signatur-Service* .
23. Das *Signatur-Service* übermittelt die Signatur an den *Service Provider* .
24. Der *Service Provider* informiert den *User Client* über den erfolgreichen Signaturerstellungsprozess.
25. Der *User Client* informiert die Benutzerin bzw. den Benutzer über den erfolgreichen Signaturerstellungsprozess.

6.5. Umsetzung

Basierend auf der in Abbildung 10 gezeigten Architektur und dem in Abbildung 11 illustrierten Prozessfluss wurde ein Demonstrator implementiert. Dieser repräsentiert im Wesentlichen eine serverbasierte Signaturlösung, die über mobile Endnutzergeräte verwendet werden kann. Durch diesen Demonstrator wird die im Rahmen dieser Studie erarbeitete Lösung einer alternativen Mehrfaktorauthentifizierung durch einen praktischen Einsatz evaluiert. Details des implementierten Demonstrators werden in diesem Abschnitt diskutiert.

6.5.1. Umsetzungsbasis

Als Ausgangsbasis für den entwickelten Demonstrator wurde die mobile Signaturlösung ServerBKU herangezogen. Diese wurde vom Institut für Angewandte Informationsverarbeitung und Kommunikationstechnologie (IAIK) [25] entwickelt, und entspricht in ihrer Funktionalität im Wesentlichen der österreichischen Handy-Signatur. Allerdings weist die ServerBKU eine höhere Flexibilität in Bezug auf verschiedene Deployment-Szenarien auf. Details zur ServerBKU wurden auch von Rath et al. [26] diskutiert.

Aufgrund ihrer konzeptionellen Ähnlichkeit zur österreichischen Handy-Signatur unterliegt jedoch auch die ServerBKU ähnlichen Limitierungen. Durch das von der ServerBKU verfolgte SMS-TAN-Verfahren ist die ServerBKU wie auch die österreichische Handy-Signatur prinzipiell für eine Verwendung mit zwei getrennten Endnutzergeräten ausgelegt. Damit eignet sich die ServerBKU jedoch ausgezeichnet zur Evaluierung der in dieser Studie erarbeiteten alternativen Authentifizierungsmethode. Durch Integration dieser Methode in die ServerBKU kann diese für eine Verwendung über ein einzelnes Endnutzergerät vorbereitet werden.

Die ServerBKU implementiert prinzipiell zwei relevante Anwendungsfälle: Registrierung und Signaturerstellung. Im Zuge der Registrierung wird eine sogenannte Mobile Bürgerkarte durch die Benutzerin bzw. den Benutzer erstellt. Diese entspricht im Wesentlichen einem virtuellen Signatur-Token. Im Zuge der Registrierung erstellte Mobile Bürgerkarten können im Zuge einer nachfolgenden Signaturerstellung beliebig oft verwendet werden. Diese beiden prinzipiellen Anwendungsfälle, die durch die

ServerBKU abgedeckt werden, werden in den folgenden Abschnitten näher erläutert. Für jeden Anwendungsfall wird außerdem gezeigt, wie dieser erweitert wurde, um die in dieser Studie erarbeitete alternative Zweifaktorauthentifizierung zu implementieren.

6.5.2. Registrierung

Im Zuge der Registrierung wird eine neue Mobile Bürgerkarte für die Benutzerin bzw. den Benutzer erstellt. Die ServerBKU bietet Benutzerinnen und Benutzern dafür eine webbasierte Schnittstelle an. Über ein einfaches Web-Formular können die für die Erstellung der Mobilen Bürgerkarte benötigten Daten eingegeben werden. Dazu zählen u.a. eine eindeutige ID, die Telefonnummer der Benutzerin bzw. des Benutzers und das dieser Mobilen Bürgerkarte zugeordnete geheime Passwort. Die angegebene Telefonnummer wird durch Zusendung eines Aktivierungscodes verifiziert. Dieser per SMS versendete Code muss zum Abschluss der Registrierung einer neuen Mobilen Bürgerkarte über das Web-Formular eingegeben werden.

Um die in dieser Studie erarbeitete alternative Authentifizierungsmethode zu integrieren, wurde der bestehende Registrierungsprozess erweitert. Gemäß der in Abbildung 10 gezeigten Architektur, sieht dieser Methode die Verwendung einer *App* vor. Hauptaufgabe dieser *App* ist es unter anderem, an das mobile Endnutzergerät gesendete TANs zu signieren. Damit das *Signatur-Service* in der Lage ist, die derart signierten TANs zu verifizieren, muss dieses über den entsprechenden öffentlichen Schlüssel verfügen. Da jede Instanz der *App* und damit jedes mobile Endnutzergerät über einen eigenen einzigartigen Schlüssel zum Signieren von TANs verfügt, muss ein geeigneter Pairing-Prozess implementiert werden, im Zuge dessen die jeweilige Instanz der *App* an die erstellte Mobile Bürgerkarte der Benutzerin bzw. des Benutzers gebunden wird. Im Zuge dieses Pairings wir dem *Signatur-Service* der öffentliche Schlüssel der *App* übermittelt. Nur so ist das *Signatur-Service* in der Lage, signierte TANs im Zuge von Signaturerstellungsprozessen zu verifizieren.

Dieses Pairing wurde im Zuge der Entwicklung des Demonstrators in den Registrierungsprozess der ServerBKU integriert. Dazu wurde der in Abbildung 12 gezeigte Dialog der webbasierten Schnittstelle der ServerBKU hinzugefügt. Über diesen Dialog wird der Benutzerin bzw. dem Benutzer ein zufälliger Aktivierungscode angezeigt. Alternativ wurde dieser Aktivierungscode auch als QR-Code kodiert und über den implementierten Dialog dargestellt.

Über den Aktivierungscode kann die Benutzerin bzw. der Benutzer ihre bzw. seine persönliche Instanz der *App* an die soeben erstellte Mobile Bürgerkarte binden. Die *App* wurde ebenfalls im Zuge der Erstellung des Demonstrators entwickelt. Als Zielplattform wurde Google Android gewählt, da dieses mobile Betriebssystem derzeit am weitesten verbreitet ist. Die implementierte *App* bietet unter anderem einen Dialog, über den die *App* an eine Mobile Bürgerkarte gebunden werden kann. Dieser Dialog ist in Abbildung 13 dargestellt und kann entweder manuell oder durch Scannen des angezeigten QR-Codes aufgerufen werden. Über den durch die *App* angezeigten Dialog kann die Benutzerin bzw. der Benutzer ihre bzw. seine Telefonnummer und den Aktivierungscode eingeben. Außerdem kann eine lokale PIN festgelegt werden, über die die Signaturfunktion der *App*, über die in weiterer Folgen TANs signiert werden, geschützt wird. Dazu implementiert die *App* ein eigenes Pin-Pad, um gegen Keyboard-Logger und ähnliche Schadsoftware immun zu sein. Dieses Pin-Pad ist in Abbildung 14 dargestellt. Nach Eingabe aller benötigten Daten führt die *App* den Pairing Prozess selbstständig und transparent für die Benutzerin bzw. den Benutzer aus. Dabei wird im Wesentlichen ein neues Schlüsselpaar generiert und der öffentliche Schlüssel gesichert an das

Signatur-Service übertragen, wo dieser für die eben erstellte Mobile Bürgerkarte registriert wird. Der erfolgreiche Abschluss des Pairing-Prozesses wird über die webbasierte Schnittstelle der ServerBKU angezeigt. Dies ist in Abbildung 16 dargestellt. Nach Abschluss des Pairing-Prozesses kann die eben erstellte Mobile Bürgerkarte zur Erstellung von Signaturen verwendet werden.

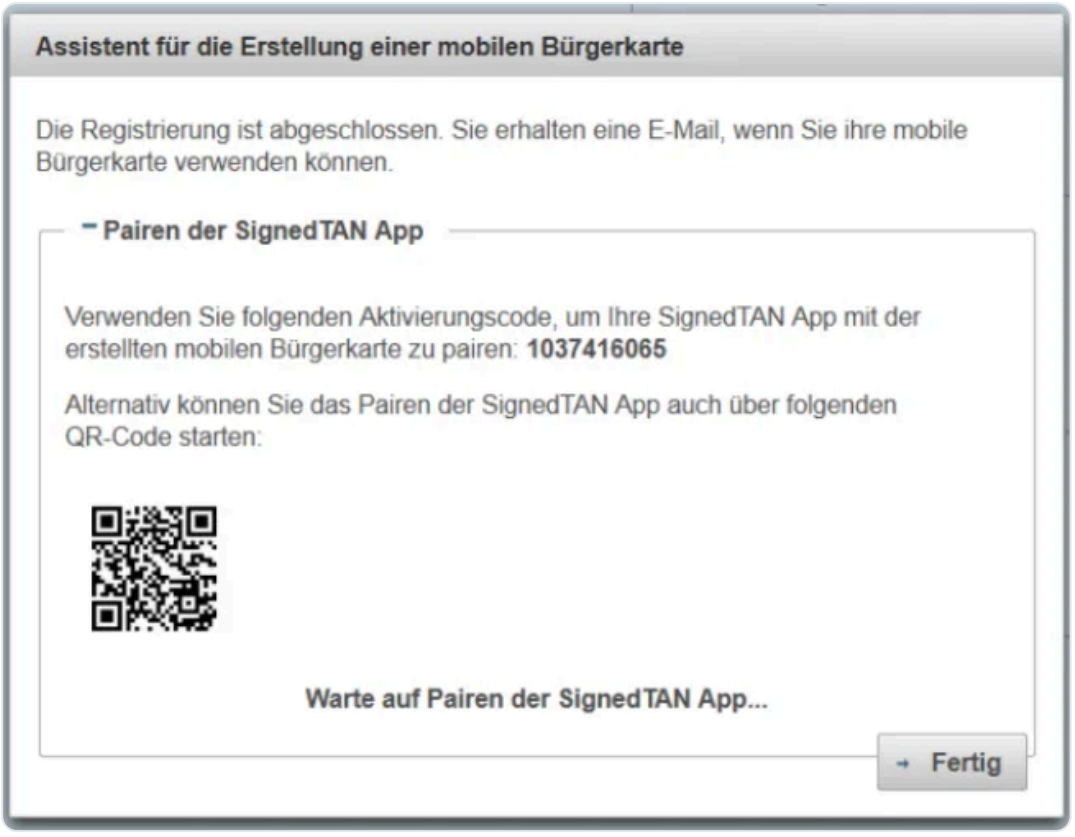


Abbildung 14. Pin-Pad der Smartphone-App. Abbildung 15. Durchführung des Pairings.

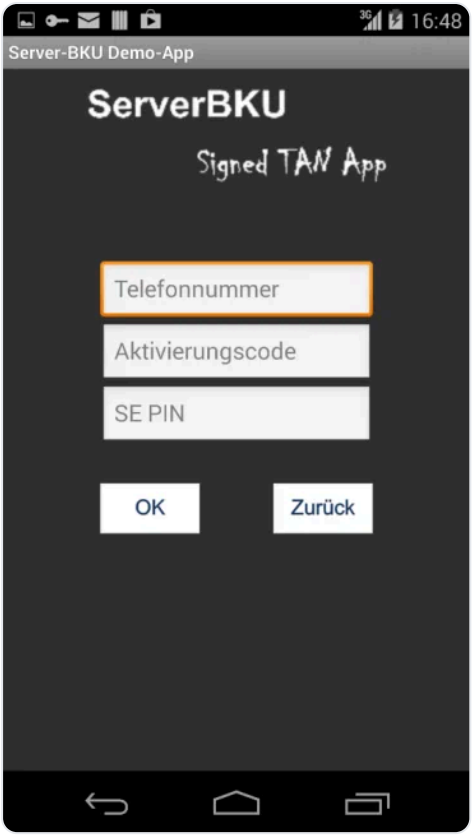


Abbildung 12. Implementierung des Pairing-Prozesses. Abbildung 13. Pairing-Dialog der Smartphone-App.

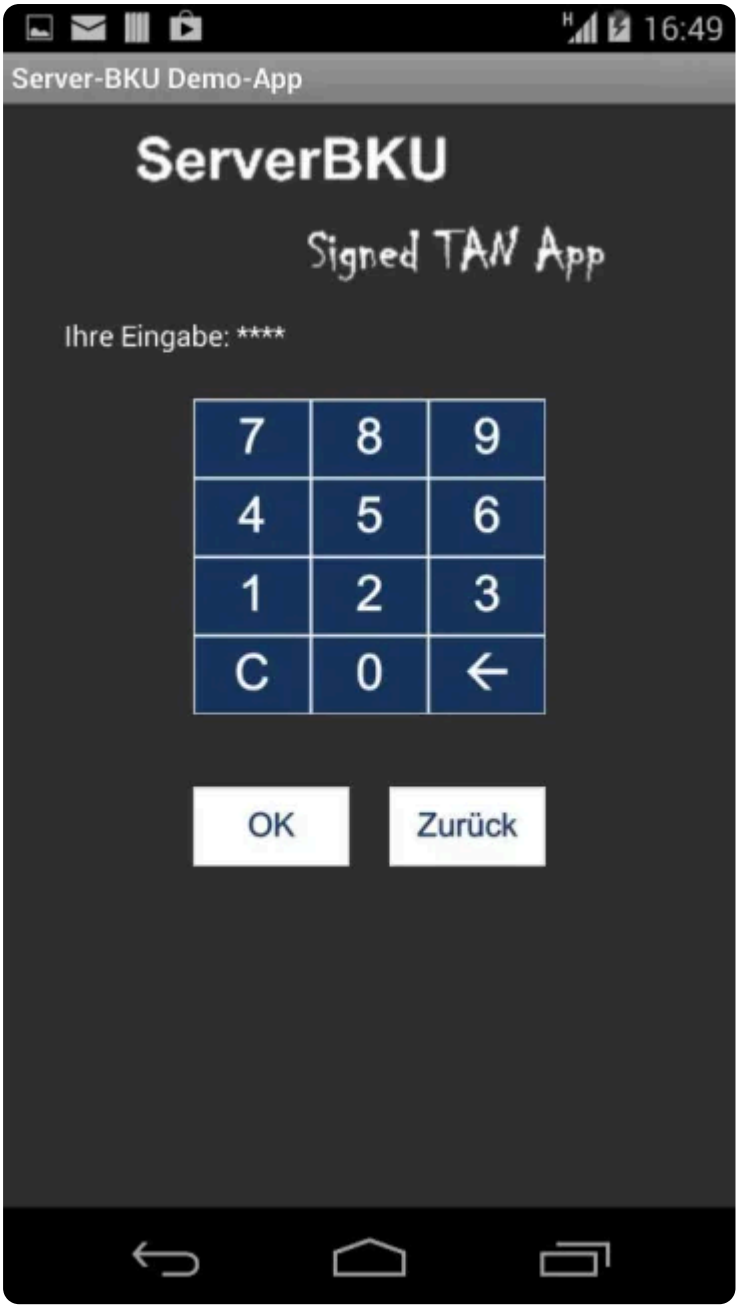
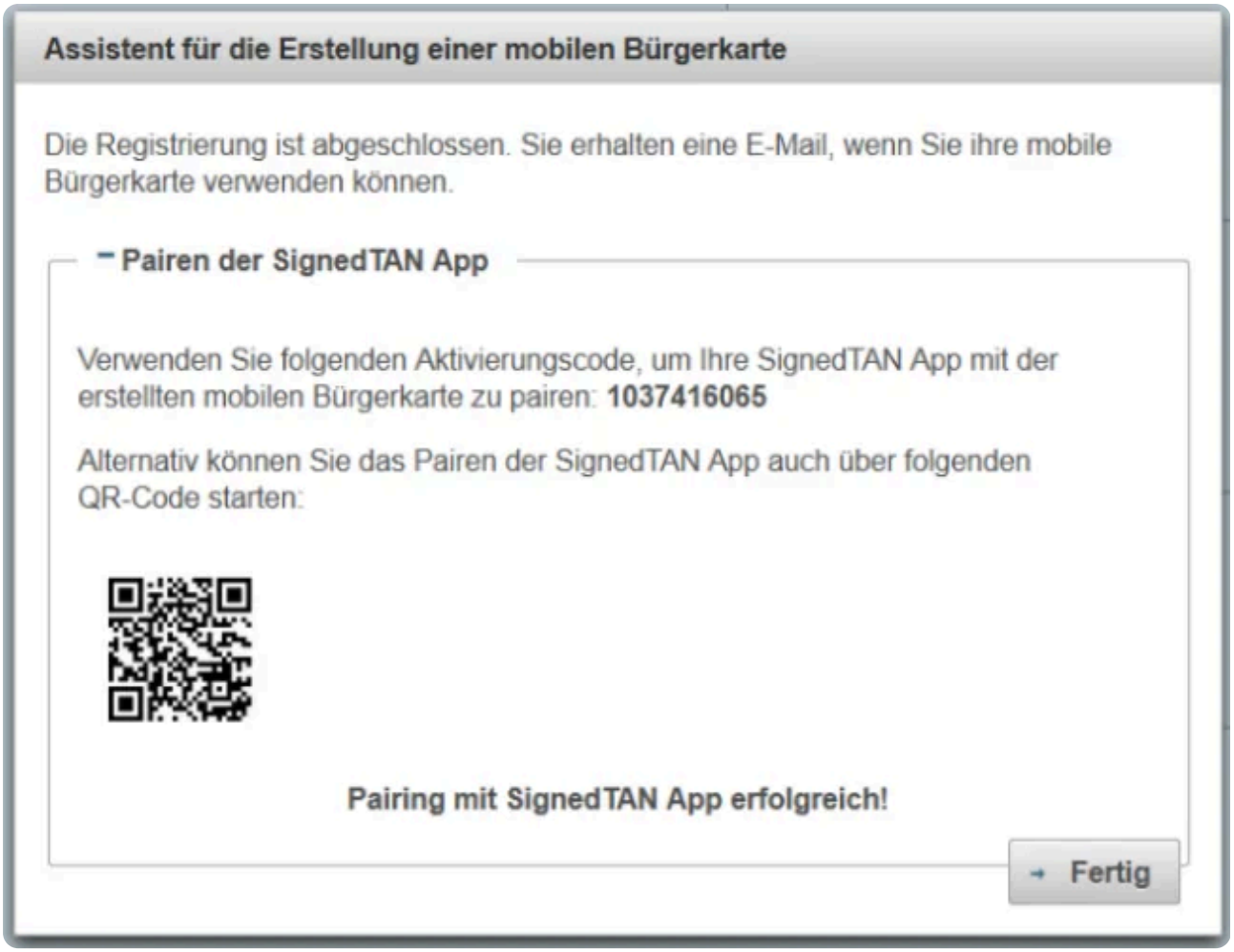

16:49
Server-BKU Demo-App
ServerBKU
Signed TAN App
Ihre Eingabe: ****
7
8
9
4
5
6
1
2
3
C
0
←
OK
Zurück



Abbildung 16. Erfolgreicher Abschluss des Pairing-Prozesses.



6.5.3. Signaturerstellung

Die einzelnen Schritte eines typischen Signaturerstellungsprozesses wurden in Abschnitt 6.4 definiert und erläutert. Im Folgenden wird gezeigt, wie ein Signaturerstellungsprozess über den entwickelten Demonstrator durchgeführt werden kann.

Zur Illustration der implementierten Funktionalität wurde neben der nötigen *App* selbst noch eine Demo-App entwickelt. Diese implementiert im Wesentlichen die Funktionalität der Komponente *Service Provider*, die auch durch die in Abbildung 10 dargestellte Architektur des berücksichtigt wurde. Gemäß der dieser Komponente zugewiesenen Rolle definiert der *Service Provider*, d.h. die Demo-App, die DTBS und übermittelt diese an die Signaturlösung. Am Ende des Signaturerstellungsprozesses fungiert die Demo-App zusätzlich als Zieldestination der erstellten Signatur.

Für den erstellten Demonstrator wurde die Demo-App so einfach wie möglich gehalten. Ihre Funktionalität beschränkt sich im Wesentlichen auf die Definition der DTBS, die Kommunikation mit der Signaturlösung und die Darstellung des Resultats des Signaturerstellungsprozesses. Das GUI der Demo-App, über das Benutzerinnen und Benutzer die DTBS festlegen können, ist in Abbildung 17 dargestellt. Nach Betätigen des Buttons *Signieren* werden die im Textfeld definierten Daten für die Signatur vorbereitet und an die Signaturlösung geschickt.

Vor Erstellung der Signatur wir die Benutzerin bzw. der Benutzer authentifiziert. Dazu werden von der zur Signaturlösung gehörenden *App* zunächst die eindeutige ID der zu verwendenden Mobilen Bürgerkarte und das zugehörige Passwort von der Benutzerin bzw. vom Benutzer abgefragt. Dies ist in Abbildung 18 dargestellt. Im Anschluss wird eine TAN an das mobile Endnutzergerät gesendet. Die Benutzerin bzw. der Benutzer tippen diese TAN wie in Abbildung 19 dargestellt in die *App* ein. Diese TAN wird dann durch die *App* signiert. Dazu wird von der Benutzerin bzw. vom Benutzer der im Zuge des Pairings definierte lokale Autorisierungscode abgefragt. Dessen Eingabe erfolgt in der *App* wiederum über das implementierte App-eigene Pin-Pad. Dies ist in Abbildung 20 illustriert.

Nach Eingabe des lokalen Autorisierungscodes wird die TAN durch die *App* signiert, an das zentrale *Signatur-Service* übermittelt und der Authentifizierungsprozess damit abgeschlossen. Damit wird der Signaturerstellungsprozess im *SSCD* autorisiert. Das Resultat dieses Prozesses wir am Ende wieder an die Demo-App übertragen und von dieser dargestellt. Dies ist in Abbildung 21 dargestellt.



Abbildung 17. GUI der Demo-App zur Definition der DTBS. Abbildung 18. Start des Authentifizierungsprozesses.



Abbildung 19. Eingabe der empfangenen TAN. Abbildung 20. Eingabe des lokalen Autorisierungscodes zur Signierung der TAN.


16:52
Server-BKU Demo-App
ServerBKU
Signed TAN App
Vergleichswert:
ZhoovQTWxQ
rjj9rr
OK
Zurück

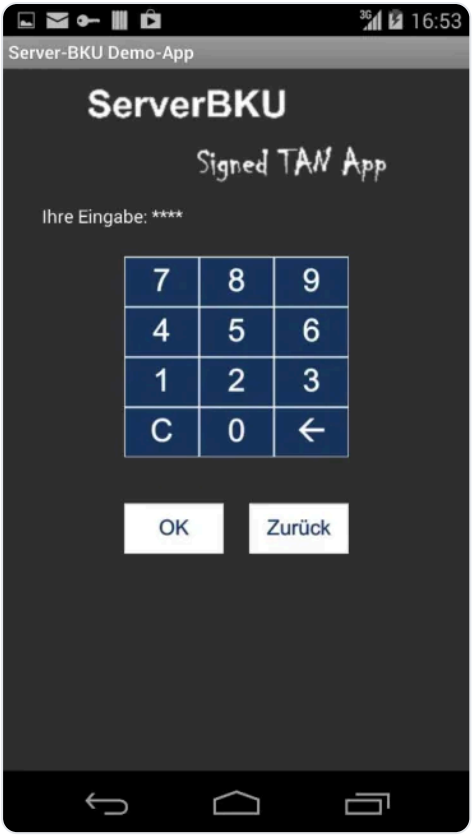


Abbildung 21. Darstellung des Resultats des Signaturerstellungsprozesses durch die Demo-App.



7. Fazit

Im Rahmen dieser Studie wurden unterschiedliche Ansätze betrachtet, Mehrfaktorauthentifizierung auf mobilen Endnutzergeräten umzusetzen. Dies ist vor allem deshalb sinnvoll, da mobile Endnutzergeräte in zunehmenden Maße klassische Geräte wie Desktop-PC oder Laptops ablösen. Dementsprechend werden mobile Endnutzergeräte auch verstärkt für den Zugriff auf sicherheitskritische Applikationen verwendet. E-Banking und E-Government sind dafür nur zwei von vielen Beispielen. Der Zugriff auf diese Applikationen muss über geeignete mehrfaktorbasierte Authentifizierungsmethoden abgesichert sein, um das benötigte Maß an Sicherheit zu gewährleisten.

Um eine geeignete Lösung für mobile Endnutzergeräte zu entwickeln, wurden zunächst unterschiedliche Ansätze systematisch anhand wohldefinierter Anforderungen analysiert. Basierend auf dem Ergebnis dieser Analyse wurde in weiterer Folge eine Authentifizierungslösung erarbeitet. Diese wurde bewusst abstrakt

gehalten, um für beliebige Anwendungsszenarien anwendbar zu sein. Die tatsächliche Anwendbarkeit der erarbeiteten Lösung wurde anhand eines spezifischen Szenarios, nämlich einer serverbasierten Signaturlösung, evaluiert. Zu guter Letzt wurde auch noch die praktische Umsetzbarkeit der entwickelten Lösung gezeigt, indem diese durch einen Demonstrator prototypisch umgesetzt wurde.

Die Resultate dieser Studien zeigen, dass sichere Mehrfaktorauthentifizierung auf mobilen Endnutzergeräten prinzipiell möglich ist. Zwar können viele der etablierten Methoden auf mobilen Geräten schwer bis gar nicht verwendet werden, da diese ursprünglich für eine Verwendung auf klassischen Endnutzergeräten konzipiert und entwickelt wurden. Jedoch bieten moderne mobile Geräte eine Vielzahl an neuen Möglichkeiten und zusätzlichen Technologien, die wiederum die Entwicklung neuer Authentifizierungsmethoden ermöglichen. Wie in dieser Studie anhand einer konkreten Umsetzung gezeigt wurde, können diese neuen Methoden auch in bestehende Applikationen integriert werden. Dadurch können diese Applikationen für eine sichere Verwendung über mobile Endnutzergeräte vorbereitet werden.

8. Referenzen

[1] Buergerkarte [2014]. Handy-Signatur und Bürgerkarte. http://www.buergerkarte.at/

[2] EasyBank [2014]. MobileTAN – was ist das? https://www.easybank.at/easy/Service_Links/e_banking_Information/44476/mobile-tan.html

[3] Suo, Xiaoyuan, Ying Zhu, and G S Owen [2005]. Graphical passwords: a survey. In Computer Security Applications Conference, 21st Annual, pages 10 pp.–472. ISSN 1063-9527. doi:10.1109/CSAC.2005.27

[4] Almulhem, Ahmad [2011]. A graphical password authentication system. In World Congress on Internet Security (WorldCIS-2011), London, UK, February 21, volume 23

[5] Khan, Wazir Zada, Mohammed Y Aalsalem, and Yang Xiang [2011]. A graphical password based system for small mobile devices. arXiv preprint arXiv:1110.3844

[6] Google [2011]. Introducing Android 4.0. http://www.android.com/about/ice-cream-sandwich/

[7] Apple [2013]. Apple iPhone 5S Features. http://www.apple.com/iphone-5s/features/

[8] Diaz, Jesus [2013]. iPhone 5S Fingerprint Security Can Be Easily Broken, Hackers Show. http://gizmodo.com/hackers-iphone-5s-fingerprint-security-is-not-secure-1367817697

[9] Silverman, Dwight [2011]. Android 4.0's facial recognition is cool, but don't trust it yet. http://blog.chron.com/techblog/2011/12/android-4-0s-facial-recognition-is-cool-but-dont-trust-it-yet/

[10] Bhattacharyya, Debnath, Rahul Ranjan, A Farkhod Alisherov, and Minkyu Choi [2009]. Biometric authentication: A review. International Journal of u-and e-Service, Science and Technology, 2(3), pages 13–28

[11] MobiThinking [2014]. Global mobile statistics 2014 Part A: Mobile subscribers; handset market share; mobile operators. http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats/a.

[12] IDENTIV [2014]. iAuthenticateTM Smart Card Reader. http://www.identive-group.com/en/products-and-solutions/identification-products/mobility-solutions/mobile-readers/iauthenticate-smart-card- reader

[13] EMC² [2014]. RSA SecurID. http://www.emc.com/security/rsa-securid.htm

[14] SafeNet [2014]. SafeNet Authentication Manager Express – OTP Generator. http://www.safenet-inc.com/multi-factor-authentication/authentication-management/safenet-authentication-manager-express-samx/

[15] Vasco [2014]. DIGIPASS. https://www.vasco.com/de/products/digipass/digipass_index.aspx

[16] Network Working Group [2005]. HOTP: An HMAC-Based One-Time Password Algorithm. http://tools.ietf.org/html/rfc4226.

[17] Google [2014]. Google Authenticator. http://code.google.com/p/google-authenticator/

[18] OATH [2014]. Security Hardware and Solution Provider for PKI token, One Time Password (OTP),Software Protection Dongle, Smart Card, Reader and Mobile Solution for Payment | OATH. http://www.openauthentication.org/

[19] Internet Engineering Task Force [2011]. TOTP: Time-Based One-Time Password Algorithm. http://tools.ietf.org/html/rfc6238

[20] Barada Project [2014]. Barada » Two-Factor Authentication For Android. http://barada.sourceforge.net/

[21] Bowling, Drew [2012]. Open Sesame: Google's Newest Security Log-In Uses QR Codes. http://www.webpronews.com/open-sesame-googles-newest-security-log-in-uses-qr-codes-2012-01

[22] SQRL [2014]. SQRL | Secure Quick Reliable Login. http://sqrl.pl/blog/

[23] FIDO Alliance [2014]. FIDO Alliance. http://fidoalliance.org/

[24] Orthacker, Clemens, Martin Centner, and Christian Kittl [2010]. Qualified Mobile Server Signature. In Proceedings of the 25th TC 11 International Information Security Conference SEC 2010

[25] IAIK [2014]. Institut für Angewandte Informationsverarbeitung und Kommunikationstechnologie. http://www.iaik.at

[26] Rath, Christof, Simon Roth, Manuel Schallar, and Thomas Zefferer [2014]. A Secure and Flexible Server-Based Mobile eID and e-Signature Solution. In The Eighth International Conference on Digital Society, pages 7–12