KoPS 3.1
Handbuch

Stand:	Release
Version:	2.2.7
SW-Version:	3.1.*

Erstellt von: eHealth Experts GmbH
Emil-Figge-Str. 85
4427 Dortmund

Inhaltsverzeichnis

Einleitung

Um Primärsystem-Herstellern zu ermöglichen, ihr Primärsystem ohne physischen Konnektor, Kartenterminals, Karten oder Online-Anbindung zu überprüfen, bietet die gematik den Konnektorsimulator für Primärsysteme (KoPS) an. Er wurde mit dem Ziel entwickelt, die Schnittstellen des Anwendungskonnektors zum Primärsystem virtuell und spezifikationskonform abzubilden.

KoPS richtet sich damit in erster Linie an die Hersteller von Primärsystemen und dient ihnen als Hilfestellung bei der Anpassung ihrer Produkte für die Integration in die *Telematikinfrastruktur* (TI). Die Nutzer haben mit KoPS die Möglichkeit, einfach und bequem die Funktion eines Konnektors spezifikationskonform zu simulieren und dadurch die Kompatibilität ihrer Software zu testen. KoPS dient darum in erster Linie dazu, dem Nutzer eine Simulationsumgebung für eine nichtlineare Auseinandersetzung zum Verhalten von Konnektor und Primärsystem zur Verfügung zu stellen.

Als weitergehende Hilfestellung bietet KoPS auch ein vorkonfiguriertes Set von linearen Testfällen an, welche Schritt-für-Schritt durchgeführt werden können. Jeder einzelne Testschritt erlaubt eine technische Validierung der zu erwartenden Reaktionen des zu testenden Primärsystems. Zusätzlich kann durch den Nutzer eine manuelle Validierung in Hinblick auf die Erwartungshaltung an das Primärsystem erfolgen. Im Anschluss an die Testfälle werden durch KoPS zusammenfassende Testreports erstellt, mit denen die spezifikationskonforme Funktionsweise des Primärsystems gegenüber der *gematik* dokumentiert werden kann.

Die derzeitige Ausbaustufe von KoPS 3.1 (Version 3.1.*) bildet alle notwendigen Schnittstellen für die Fachanwendungen *Versichertenstammdatenmanagement* (VSDM), *Notfalldaten-Management (*NFDM), *Elektronischer Medikationsplan* (eMP) und *Datenmanagement zur Prüfung der Arzneimitteltherapiesicherheit (*AMTS), *sichere Kommunikation im Gesundheitswesen* (KIM), *Basisdienste QES*, *tokenbasierte Authentisierung* (TBAuth) sowie *elektronische Patientenakte* (ePA) mit dem Stand *Online-Produktivbetrieb* (Release 3.1.3) an.

Mehrbenutzerszenario

Die Software KoPS 3.1 wird, vergleichbar mit der Software eines physischen Konnektors, als Serveranwendung ausgeführt. Der Zugriff auf die Anwendung erfolgt über eine separate Web-Oberfläche; Hier kann der Nutzer sowohl die Konfiguration von KoPS 3.1 anpassen als auch die TI simulieren. Die Nutzung dieser Serveranwendung kann auf zwei unterschiedliche Arten erfolgen:

- als Installation auf einem netzwerkfähigen Computer mit der Möglichkeit zum Zugriff durch verschiedene Nutzer von verschiedenen Arbeitsplätzen,

- als lokale Installation zum alleinigen Gebrauch an einem Einzelplatz-PC.

Obwohl KoPS 3.1 eine Serveranwendung ist, kann unter Umständen ein dezentraler Gebrauch durch mehrere Nutzer – wie bei einem echten Konnektor – Nebeneffekte verursachen: Verschiedene Nutzer derselben Installation könnten sich gegenseitig beeinflussen, denn Konfigurations- und Zustandsänderungen wirken sich immer global (also auf alle Nutzer) über die gesamte Anwendung aus.

Dieser Effekt lässt sich für die Nutzer dadurch umgehen, dass pro Instanz ein separates Set an Testdaten verwendet wird. Alternativ kann KoPS auch problemlos als Einzelarbeitsplatzanwendung betrieben werden – mit beliebig viele Instanzen auf verschiedenen Rechnern beziehungsweise auf verschiedenen Ports.

Nachrichtenanalyse

KoPS 3.1 bietet dem Nutzer die Möglichkeit, Nachrichten auf der Anwendungsebene zu analysieren. Es können *SOAP-Nachrichten*, *CETP-Eventnachrichten* und *DVD/SDS-Anfragen* einfach und bequem nachvollzogen werden. Probleme, die auf Netzwerkebene beziehungsweise beim TLS-Verbindungsaufbau auftreten, müssen jedoch durch separate Anwendungen (beispielsweise *Tcpdump*, *Wireshark* oder vergleichbare) analysiert werden.

Die Zertifikatsprüfung im Rahmen des TLS-Verbindungsaufbau wird in KoPS 3.1 explizit geloggt. Dadurch können Fehler in diesem Zusammenhang auch einfach in den Log-Dateien von KoPS 3.1 nachvollzogen werden.

Abgrenzung zu einem physischen Konnektor

Bei der Anwendung KoPS 3.1 handelt es sich um eine Simulation der Schnittstellen eines Anwendungskonnektors zum Primärsystem für die spezifischen Fachanwendungen VSDM, Notfalldaten-Management (NFDM), Elektronischer Medikationsplan/Arzneimitteltherapiesicherheit (eMP/AMTS), sichere Kommunikation im Gesundheitswesen (KIM), Basisdienste QES (QES), tokenbaiserte Authentisierung (TBAuth) sowie elektronische Patientenakte (ePA) in dem Stand von OPB Release 3.1.2. Darüber hinaus vorhandene Funktionen und Konfigurationsmöglichkeiten eines physischen Konnektors stehen nicht im Fokus der Anwendung und werden dementsprechend nicht mit abgebildet.

Aufgrund dieser Tatsache ergeben sich Vereinfachungen bei der Konfiguration bezüglich des Umfangs und der Komplexität. Sämtliche Einstellung bezüglich der Telematikinfrastruktur entfallen in der Anwendung. Das Infomodell wurde im Zuge dessen soweit vereinfacht, dass es nur die Aspekte abbildet, die sich auf die Außenschnittstellen zum Primärsystem auswirken.

Das Infomodell von KoPS 3.1

Das Infomodell eines Konnektors prägt sich an den Außenschnittstellen vor allem durch den Aufrufkontext aus. Daher beschränken sich die Konfigurationsmöglichkeiten bezüglich des Infomodels in KoPS 3.1 hauptsächlich auf diesen Aspekt. *Mandanten*, *Clientsysteme*, *Arbeitsplätze* und *Kartenterminals* können allerdings in KoPS 3.1 nicht alle einzeln konfiguriert und einander zugeordnet werden. Stattdessen können mehrere *Aufrufkontexte* konfiguriert werden, die jeweils über die Kombination aus *Mandanten-ID*, *Clientsystem-ID* und *Arbeitsplatz-ID* identifiziert werden. Es kann mehrere Aufrufkontexte geben, mit demselben Mandanten beziehungsweise Clientsystem oder Arbeitsplatz. Nur das Tripel aus diesen drei Werten muss immer eindeutig sein.

Pro Aufrufkontext können zugehörige Terminals verwaltet werden. Die Terminalverwaltung umfasst jedoch nicht das Pairing und Zuordnen von Terminals. Sofern ein Terminal in einem Aufrufkontext vorhanden ist, gilt es automatisch als gepairt und zugleich auch diesem Aufrufkontext zugeordnet. Die Konfiguration der Eigenschaften eines Terminals erlaubt es festzulegen, ob es sich um ein Remote-Terminal und oder ein physisches Terminal handelt und ob es aktuell verbunden ist.

Ein Terminal an mehreren Arbeitsplätzen

Während ein Terminal in der Praxis mehreren Arbeitsplätzen zugewiesen werden kann, erfolgt in KoPS 3.1 eine solche direkte Zuordnung nicht – hier werden die Terminals unterhalb eines Aufrufkontextes verwaltet. Solange der Betrachtungsgegenstand eines Tests nur ein Arbeitsplatz bzw. ein Aufrufkontext ist, spielt diese Einschränkung keine Rolle. Um dennoch indirekt simulieren zu können, dass ein Terminal gleichzeitig mehreren Aufrufkontexten zugeordnet ist, erfolgt innerhalb von KoPS 3.1 eine Synchronisation mehrerer Terminals mit der gleichen *Terminal-ID*. Wird also in einem Aufrufkontext in ein Karten-Terminal eine Karte gesteckt, wird diese Karte auch in allen anderen Karten-Terminals mit derselben Terminal-ID unter anderen Aufrufkontexten gesteckt.

Verwaltung von SMC-Bs

Eine Registrierung und Zuordnung von SMC-Bs zu den einzelnen Mandanten/Aufrufkontexten ist in KoPS 3.1 nicht notwendig. Alle angelegten SMC-Bs sind immer auch allen existierenden Mandanten zugeordnet. Um dennoch die Reaktion des Primärsystems testen zu können, wenn eine dem Mandanten nicht zugeordnet SMC-B genutzt wird, gibt es an der Karte einen Parameter. Über diesen Parameter kann angegeben werden, dass die Karte keinem Mandanten zugeordnet ist. Wenn diese Karte genutzt wird, erhält man eine Fehlernachricht mit dem entsprechenden Fehlercode.

Zugangsdaten für die TLS Verbindungen

Gemäß Konnektor-Spezifikation sind die Zugangsdaten (Benutzername und Passwort für Basic Authentication; die Zertifikate für die Client-Authentifizierung) den Clientsystemen zugeordnet. So ist es auch in KoPS 3.1 umgesetzt. Für alle Clientsystem-IDs, die in der Verwaltung der Aufrufkontexte angelegt wurden, können separate Zugangsdaten definiert werden. Darüber hinaus können Standardzugangsdaten festgelegt werden. Diese gelten für alle Clientsysteme, für die keine speziellen Zugangsdaten festgelegt wurden.

Ereignisse

KoPS 3.1 löst alle Ereignisse, die durch die Zustandsänderung an der KoPS-Oberfläche entstehen, spezifikationskonform aus. So sorgt das Stecken einer Karte in KoPS 3.1 auch dafür, dass ein entsprechendes *CardInsert*-Ereignis ausgelöst wird. Genauso sorgt das Trennen der virtuellen Verbindung zur TI auch dafür, dass die entsprechenden Ereignisse ausgelöst werden.

Allerdings lassen sich nicht alle Ereignisse, die der Konnektor werfen kann, durch Interaktionen an der Oberfläche indirekt auslösen (zum Beispiel *EC_Time_Sync_Pending_Warning*). Um die Reaktion des Primärsystems dennoch auch auf solche, Anwendungsfall unabhängige Ereignisse testen zu können, bietet KoPS 3.1 die Möglichkeit alle spezifizierten Events zusätzlich auch manuell auslösen zu können.

Bei der Registrierung von Ereignissen wird in KoPS 3.1 keine Filterfunktion für die Ereignisnachrichten unterstützt.

Installation

Systemvoraussetzungen

Die Verwendung von KoPS 3.1 stellt folgende Mindestvoraussetzungen an das System des Nutzers:

- mindestens 2GB freier Arbeitsspeicher
- ein *Java 11* fähiges Betriebssystem
- *Java 11 64-Bit*
- *Firefox ESR* (Version 53 oder höher)
    - Textkodierung *Unicode*
    - Zoomfaktor *100%* (für eine optimale Nutzung)
- Mindestfenstergröße *1024 × 768px*
  Für eine optimale Nutzung von KoPS 3.1 empfiehlt sich die Einhaltung der oben genannten Mindestmaße. KoPS 3.1 kann auch bei kleineren Bildschirmgrößen verwendet werden, jedoch wird das Interface in diesem Fall in der Größe nicht weiter minimiert, sondern gescrollt.

> **!** Es ist prinzipiell möglich, die Web-Oberfläche von KoPS 3.1 auch mit anderen aktuellen Browsern, wie *Chrome*, *Edge* und *Safari* zu verwenden, jedoch wurden diese nicht vollständig auf die Kompatibilität zu KoPS 3.1 getestet. Eine Nutzung von KoPS 3.1 in Kombination mit diesen Browsern geschieht somit ohne Gewährleistung.

Distribution

Um KoPS 3.1 nutzen zu können, muss die Software zunächst installiert werden, dafür steht folgende Distribution bereit:

1. *KoPS 3.1*

Diese Distribution beinhaltet die KoPS 3.1-Laufzeitumgebung sowie alle benötigten Konfigurationsdateien. Diese Distribution ist auf allen Betriebssystemen lauffähig, welche die Systemvoraussetzungen erfüllen. Für die Nutzung dieser Distribution müssen auf dem System die folgenden zusätzlichen Komponenten vorab installiert werden:

- Java 11

Für die Installation von Java 11 folgen sie bitte der Installationsanleitung des Herstellers.

Installation

Nach Auswahl der geeigneten Distribution muss diese in das gewünschte Installationsverzeichnis entpackt werden. Dies kann zum Beispiel unter Windows der folgende Pfad sein:

```
C:\KoPS3.1\
```

> **!** Für den Start von KoPS 3.1 wird eine Lizenzdatei benötigt. Diese muss vor dem Start der Anwendung in das Installationsverzeichnis kopiert werden. Der Name der Lizenzdatei muss dem Schema *kops*.lic* entsprechen.

> **!** Nach der Installation einer neuen Version von KoPS sollte der Browser-Cache geleert werden, um eine korrekte Darstellung der Benutzeroberfläche zu gewährleisten.

Starten des Programms

Das Installationsverzeichnis beinhaltet ein Batch-Script zum Start der KoPS 3.1-Laufzeitumgebung. Unter Windows wird für den Start die Datei *start.bat*, unter Linux/Unix die Datei *start.sh* verwendet.

Beispiel für Windows:

```
C:\KoPS3.1\start.bat
```

Standardmäßig ist die Web-Oberfläche über alle Netzwerkadressen des Rechners mit dem Port *8080* erreichbar. Soll die Oberfläche nur über eine bestimme Netzwerkadresse beziehungsweise über einen anderen Port erreichbar sein, kann dies über Kommandozeile-Parameter angegeben werden.

Beispiel für Windows:

```
C:\KoPS3.1\start.bat <host> <port>
```

Nach dem Start der KoPS 3.1 Laufzeitumgebung erfolgt der Zugriff auf die Web-Oberfläche via Browser. Dazu muss im Browser die folgende URL eingegeben werden:

```
http://localhost:8080/KoPS/web
```

beziehungsweise

```
http://<host>:<port>/KoPS/web
```

Inbetriebnahme

Nach dem erfolgreichen Start der Anwendung kann die Web-Oberfläche im Browser aufgerufen werden. Die Web-Oberfläche von KoPS 3.1 besteht grundsätzlich aus drei Bereichen, der Kopfzeile, dem Hauptbereich und dem Seitenmenü. Zusätzlich wird KoPS 3.1 in zwei Einsatzbereiche unterteilt, die beiden Betriebsmodi *Test*- und *Admin-Modus*. Ein Umschalten zwischen den verschiedenen Anwendungen (VSDM, eMP/AMTS, ePA, NFDM, KIM, QES, TBAuth) erfolgt über ein Auswahlmenü in der Kopfzeile.

Abhängig von der eingesetzten Lizenz stehen für den Nutzer ggf. nicht alle Anwendungen zur Verfügung.

Wechsel zwischen den Anwendungen

Das Bedienkonzept von KoPS ist für die verschiedenen Anwendungen so aufgebaut, dass die GUI für jede simulierte Anwendung eine eigene Ansicht bereitstellt. Dies dient der Übersichtlichkeit und der Reduktion der Komplexität der Anwendungen für den Nutzer und schafft zusätzlich die Möglichkeit, vom Primärsystem nicht unterstützte freiwillige Anwendungen (z. B. die Anwendung NFDM für Zahnärzte) nicht in einer gemeinsamen Oberfläche darstellen zu müssen. Für die jeweilige Ansicht werden die anwendungsspezifische Funktionalität, die anwendungsspezifischen Testdaten und der anwendungsspezifische Testfallkatalog angezeigt. Die Funktionalitäten, Testdaten und Testfallkataloge der anderen Anwendungen werden ausgeblendet.

Ein Umschalten zwischen den Anwendungen erfolgt über ein Auswahlmenü. Bei dem Wechsel der Anwendungen werden jeweils die Testdaten und Testkataloge der gewählten Anwendung geladen. Temporär im Arbeitsspeicher vorgehaltene Testdaten der vorherigen Anwendung sowie der bestehende Aufrufkontext werden dabei verworfen.

Betriebsmodi

Die Konnektor-Simulation KoPS 3.1 verfügt über zwei Betriebsmodi. Der jeweils für den Nutzer aktive Modus wird über dem Seitenmenü eingeblendet und zusätzlich durch ein eindeutiges Farbschema kenntlich gemacht. Beim Start von KoPS 3.1 wird standardmäßig der Test-Modus der Anwendung VSDM mit dem Menü-Eintrag *Aufrufkontext* geladen. Der Nutzer kann zwischen den beiden Betriebsmodi nach Belieben wechseln.

Admin-Modus

Befindet sich KoPS 3.1 im Admin-Modus, verwendet KoPS 3.1 ein blaues Farbschema.

Wünscht der Nutzer die vordefinierten Testdaten zu bearbeiten, muss dieser hierzu in den Admin-Modus wechseln.

Test-Modus

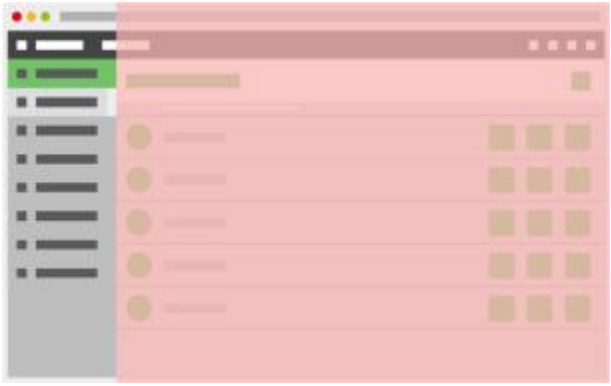
Der Test-Modus dagegen verwendet ein grünes Farbschema, um eine schnelle visuelle Differenzierung zu gewährleisten.

Ziel des Test-Modus ist es, dem Nutzer die Möglichkeit zu geben, verschiedene Prozesse rund um einen spezifikationskonformen Konnektor nachzustellen.

Kopfzeile (Header)

Über das Auswahlmenü in der Kopfzeile kann ein Wechsel zwischen den Anwendungen durchgeführt werden. Ebenfalls in der Kopfzeile befinden sich die Schaltflächen *Live-Protokoll*, *Konnektorstatus*, *Verbindungsstatus* und *Systeminformation*.

Seitenmenü



Für den schnellen Zugriff auf alle relevanten Funktionen verfügt KoPS 3.1 über ein links angeordnetes Seitenmenü. Dieses kann über das Menü-Icon ein- und ausgeklappt werden. Beim Start von KoPS 3.1 befindet sich das Seitenmenü im ausgeklappten Zustand.

Zum Wechsel zwischen den beiden Betriebsmodi wird die Schaltfläche *Modus wechseln* im Seitenmenü verwendet.

Folgende Menü-Einträge sind Bestandteil der beiden Betriebsmodi:

	Test	Admin	Beschreibung
Allgemeine Anteile			
Test-Modus	✓		Wechsel zum Admin-Modus
Admin-Modus		✓	Wechsel zum Test-Modus
Aufrufkontext	✓ *	✓ **	* Konfigurationen der verfügbaren Aufrufkontexte. **Erstellen und Verwalten von Aufrufkontexten.
Praxis		✓	Erstellen und Verwalten von Praxiskarten (SMC-B, HBA, ...).
Abonnenten	✓		Übersicht der am Ereignisdienst angemeldeten Abonnenten.
Ereignisse	✓		Auslösen vordefinierter Ereignisse.
Testlabor	✓		Durchführung von Testfällen.

Authentifizierung	✓	✓	Verwaltung der Zugangsdaten für die Client-Authentifizierung.
Einstellungen	✓	✓	Konfigurationen des Konnektors (Verbindung und Dienstverzeichnis), des Ereignisdienstes und der Anzeigeeinstellungen (Live-Protokoll, Aktualisierung).
Protokolle	✓	✓	Zugriffmöglichkeit auf die Protokolldateien (Basis-Log, SOAP-Log, Schnittstellen-Log), die während der Nutzung des Programmes generiert wurden.
Datensicherung		✓	Erstellung von Datensicherung und Wiederherstellung einer vorherigen Konfiguration.
Anwendungsspezifische Anteile VSDM			
Versicherte		✓	Erstellen und Verwalten von Versichertenkarten (eGK, KVK, ...).
Anwendungsspezifische Anteile NFDM			
eGK NFDM		✓	Erstellen und Verwalten von eGKs für die Fachanwendung NFDM
Anwendungsspezifische Anteile eMP/AMTS			
eGK eMP/AMTS		✓	Erstellen und Verwalten von eGKs für die Fachanwendung eMP/AMTS
Anwendungsspezifische Anteile QES			

Praxis/ QES		✓	Erstellen und Verwalten von Praxiskarten für die Fachanwendung QES
Anwendungsspezifische Anteile KIM			
Praxis / KIM		✓	Erstellen und Verwalten von Praxiskarten für die Fachanwendung KIM
Postfächer	✓**	✓*	* Erstellen und Verwalten von E-Mails in POP3-Postfächern ** Anzeige der Inhalte von POP3 und SMTP-Postfächern
VZD-Einträge	✓		Übersicht der Inhalte des Verzeichnisdienstes
Anwendungsspezifische Anteile TBAuth			
TBAuth	✓		Einstellungen zur Fachanwendung TBAuth
Anwendungsspezifische Anteile ePA			
eGK/ePA		✓	Erstellen und Verwalten von eGKs für die Fachanwendung ePA und Verwaltung von Einstellungen zur Patientenakte

Hauptfenster

Der Hauptbereich von KoPS 3.1 befindet sich im rechten Bereich des Browserfensters.

Das Hauptfenster von KoPS 3.1 unterteilt sich in zwei Bereiche: der Titelleiste und dem Inhaltsbereich. Innerhalb dieses Bereiches werden alle wichtigen Aktionen der Software durchgeführt.

Filter

Unterhalb der Titelleiste wird bei Listenansichten ein Filter eingeblendet, der die Filterung der Einträge erlaubt. Hinter dem Eingabefeld des Filters wird die Gesamtzahl der Einträge beziehungsweise im gefilterten Zustand die Anzahl der dargestellten Einträge angezeigt. Sobald eine Filterung vorgenommen wird, werden Einträge entsprechend der Eingabe reduziert.

Die Filterung erfolgt über das *Label* und die *Beschreibung*. Die angegebenen Filterwerte werden als *reguläre Ausdrücke* für die Filterung ausgewertet. Ohne Sonderzeichen verhält sich die Filterung wie eine normale Textsuche. Unter Verwendung von Sonderzeichen (*?*+{}^$()*) können darüber hinaus auch komplexere Filterungen vorgenommen werden. Dabei werden Sonderzeichen innerhalb des Sucheintrages immer als regulärer Ausdruck gewertet. Um diese Zeichen auch in einer einfachen Textsuche verwenden zu können, muss diesen immer ein Backslash \ vorangestellt werden.

Beispiele für mögliche Filterwerte und deren Bedeutung:

- *eGK*
  Es werden alle Einträge mit dem Wert „eGK“ gefiltert.
- *eGK* \(
  Es werden alle Einträge mit dem Wert „eGK (“ gefiltert.
- *eGK|PIN*
  Es werden alle Einträge mit dem Wert „eGK“ oder dem Wert „PIN“ gefiltert.
- *(?=.*eGK)(?=.*PIN)*
  Es werden alle Einträge mit dem Wert „eGK“ und dem Wert „PIN“ gefiltert.

- *\d*
  Es werden alle Einträge gefiltert, die eine Nummer enthalten.
- *(Prüfung).*(nachweis)*
  Es werden alle Einträge gefiltert, die als Wert „Prüfung“ folgend von einer beliebigen Anzahl von Zeichen und abschließend dem Wert „…nachweis“ enthalten.
- *(Prüfung).?(nachweis)*
  Es werden alle Einträge gefiltert, die als Wert „Prüfung“ folgend von einem beliebigen Zeichen und abschließend dem Wert „…nachweis“ enthalten.

Live-Protokoll

Zur Kontrolle der korrekten Funktionsweise der Konnektor-Simulation kann über die Schaltfläche *Live-Protokoll* jederzeit die zentrale Log-Datei von KoPS 3.1 eingesehen werden. Das Live-Protokoll steht dabei sowohl im Test-Modus als auch im Admin-Modus zur Verfügung.

Die Darstellung erfolgt über einen nichtmodalen Dialog. Die Position und Größe dieses Dialogs kann vom Nutzer geändert werden. Wird das Browserfenster so sehr verkleinert, dass der Dialog nicht mehr vollständig dargestellt werden kann, kann es beim Vergrößern des Browserfensters zu Darstellungsproblemen kommen. Sollte das Live-Protokoll außerhalb des sichtbaren Bereichs dargestellt werden, kann das Darstellungsproblem dadurch behoben werden, indem das Live-Protokoll über entsprechende Schaltflächen neu eingeblendet oder die Seite gänzlich neu geladen wird.

Das Live-Protokoll verfügt über eine Titel-Leiste mit verschiedenen Aktionen, die die Darstellung des Fensters beeinflussen:

Icon	Funktion
	Live-Protokoll auf die Größe der Titel-Leiste reduzieren.
	Live-Protokoll ausklappen.

	Live-Protokoll in linke untere Ecke minimieren.
	Live-Protokoll wiederherstellen.
	Live-Protokoll maximieren.
	Live-Protokoll schließen.

Die Ausgabe des Live-Protokolls lässt zusätzlich eine Reduzierung der Ausgabe in Bezug auf die vier Kategorien *Ereignisse*, *Aufrufe*, *Fehler* und *Hinweise* zu. Sie erfolgt über die An- oder Abwahl der jeweiligen Checkboxen im oberen Menüband des Fensters. Zusätzlich besteht die Möglichkeit, die Log-Ausgabe über die Angabe eines Freitextfilters einzuschränken. Über die Checkbox *automatisches Scrollen* kann das automatische Scrollen bei Aktualisierungen aus- oder eingeschaltet werden.

Icon	Typ	Beschreibung
	Anfrage	Empfangener SOAP-Request.
	Antwort	Versendete SOAP-Response.
	Dienstverzeichnis	Anbruf des Dienstverzeichnisdienstes.
	Ereignis	Versendetes Ereignis.
	Ereignis	Ausgelöstes aber nicht versandtes Ereignis.
	Fehler	Versendeter SOAP-Fault
	Testfall	Ausführungsschritt eines Testfalls

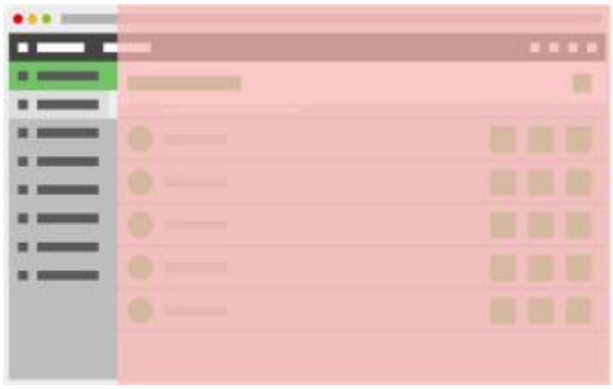
Wiederkehrende Schaltflächen

KoPS 3.1 verwendet für die Schaltflächen des Programms eigens erarbeitete Icons.

Schaltflächen in der Kopfzeile

Icon	Beschreibung
	Öffnet und schließt das Seitenmenü.
	Öffnet das Live-Protokoll.
	Konfiguration vom Konnektor (ein/aus) und Verbindungsstatus (online/offline).
	Anzeige der Systeminformationen

Schaltflächen im Seitenmenü



Icon	Beschreibung
	Wechselt zwischen Admin- und Test-Modus.

Schaltflächen in der Titelleiste

Icon	Beschreibung
	Fügt der Liste einen neuen Eintrag hinzu.
	Lädt einen gematik Testbericht als ZIP-Archiv herunter.
	Lädt eine Testfallübersicht als ZIP-Archiv herunter.

Schaltflächen im Inhaltsbereich

Icon	Beschreibung
Allgemeine Schaltflächen	
	Übernimmt die vorgenommenen Änderungen.
	Verwirft die vorgenommenen Änderungen.
	Lädt die aktuelle Konfigurationsdatei (.puppetbox) herunter.
	Stellt den Auslieferungszustand der Konfiguration wieder her.
	Einspielen einer externen Konfigurationsdatei (.puppetbox).
Schaltflächen der Listeneinträge	
	Der Eintrag ist gesperrt, er kann weder bearbeitet noch gelöscht werden.
Kontextspezifische Auswahlmöglichkeiten für Listeneinträge	
	Wechselt zur Terminalansicht eines Eintrages.
	Überschreibt das Standardverhalten eines Aufrufkontextes.

	Deaktiviert einen Eintrag.
	Aktiviert einen Eintrag.
	Trennt die Terminal-Verbindung.
	Stellt die Terminal -Verbindung wieder her.
	Öffnet die Detailmaske im Admin-Modus für eine Karte.
	Entfernt eine zuvor gesteckte Karte.
	Löscht einen bestehenden Eintrag aus der Liste.
	Lädt eine PDF-Datei herunter.
	Lädt eine Log-Datei herunter.
	Startet einen Testlauf.

Schaltflächen in den Detailmasken des Inhaltsbereiches

Icon	Beschreibung
	Schließt die Detailmaske ohne Änderungen.
	Speichert vorgenommene Änderungen in der Detailmaske.
	Klont einen bestehenden Eintrag.
	Bietet einen Hilfetext via Tooltip zu einem spezifischen Parameter an.
	Auswahl einer Datei.
	Generiert ein Zertifikat.
	Download des ausgewählten Eintrages.
	Löschen des ausgewählten Eintrages.

Menü-Einträge

Nachfolgend werden die einzelnen Menü-Einträge von KoPS 3.1 und die dahinter verknüpften Konfigurations- bzw. Analysefunktionen genauer beschrieben. Dabei wird zwischen allgemeinen Menüeinträgen in den verschiedenen Modi und anwendungsspezifischen Menüeinträgen unterschieden.

Admin-Modus

Der Admin-Modus dient der Konfiguration der Testdaten, die später im Test-Modus verwendet werden sollen. Bei diesen handelt es sich um die allgemeinen Anteile wie *Aufrufkontexte*, die *Praxiskarten* und die *Authentifizierungsdaten* sowie die anwendungsspezifischen Anteile *Versicherte*, *eGK NFDM*, *eGK eMP/AMTS, eGK ePA, Praxiskarten QES, Praxiskarten KIM und Postfächer KIM.*

Aufrufkontext

Über den Menü-Eintrag *Aufrufkontexte* im Admin-Modus gelangt man zu einer Listenansicht der im Hauptbereich konfigurierten Aufrufkontexte. Es können neue angelegt bzw. bestehende angepasst und gelöscht werden. Pro Aufrufkontext wird das Tripel aus *Mandanten-ID*, *Clientsystem-ID* und *Arbeitsplatz-ID* angegeben. Pro Aufrufkontext können die diesem Aufrufkontext zugeordneten Terminals verwaltet werden.

Praxis

Über den Menü-Eintrag *Praxis* gelangt der Nutzer zu der Verwaltung der Praxiskarten (*SMC-B*, *HBA* und *vHBA*). Für jede Karte kann sowohl ihr fachlicher Inhalt als auch ihr Verhalten festgelegt werden. Eine Beschreibung der einzelnen Parameter kann auf der Web-Oberfläche jeweils über den Hilfe-Button angezeigt werden.

Authentifizierung

Über den Menü-Eintrag *Authentifizierung* gelangt der Nutzer zu der Verwaltung der Zugangsdaten für die Client-Authentifizierung. Pro Clientsystem-ID, die in einem der konfigurierten Aufrufkontexte angelegt wurde, wird hier automatisch ein entsprechender Eintrag erstellt. Darüber hinaus gibt es *Standard-Authentifizierungs-Zugangsdaten*. Diese werden immer für die Clientsysteme verwendet, solange sie nicht explizit überschrieben worden sind. Die Zugangsdaten sind dann relevant, wenn als Authentifizierungsmechanismus *TLS* mit *Basic Authentification* oder *TLS* mit *Zertifikatsprüfung* verwendet wird.

Für die Nutzung der *Basic Authentification* muss ein Benutzername und ein Passwort hinterlegt werden. Für die *Zertifikatsprüfung* muss hingegen entweder ein Truststore hochgeladen werden oder ein Zertifikat durch KoPS 3.1 generiert werden, welches das Primärsystem dann ebenfalls verwendet.

> **!** Das Format der Keystores bzw. Truststores ist gemäß Konnektor Spezifikation PKCS12. Die Passwörter (Datei und Schlüssel) müssen für KoPS 3.1 immer *123456* sein.

Datensicherung

Im Menü-Eintrag *Datensicherung* kann der Nutzer den aktuellen Zustand seiner Konfiguration (Testdaten, Testfälle und Einstellungen) von KoPS 3.1 sichern, wiederherstellen oder eine neue Konfiguration einspielen. Um den aktuellen Zustand von KoPS 3.1 zu sichern, muss der Nutzer die Backup-Datei (.puppetbox) seiner aktuellen Konfiguration herunterladen. Bei Bedarf kann er diese Datei über die entsprechende Funktion wieder in KoPS 3.1 einspielen. Zudem ist es möglich, fremde Konfigurationen einzuspielen. In diesem Fall werden die eigenen Testdaten überschrieben.

Versicherte

Über den für die Anwendung VSDM spezifischen Menü-Eintrag *Versicherte* gelangt der Nutzer zu der Verwaltung der Versichertenkarten (eGK und KVK). Für jede Karte kann sowohl ihr fachlicher Inhalt als auch ihr Verhalten festgelegt werden. Für die Inhalte gibt es Eingabefelder, die eine vereinfachte Bearbeitung der Versichertenstammdaten (VSD) erlauben. Darüber hinaus ist es auch möglich, generierte XML-Daten für die persönlichen Versichertendaten (PD), die allgemeinen Versicherungsdaten (VD), die geschützten Versichertendaten (GVD) und den Prüfungsnachweis zu überschreiben und somit auch Datensätze zu erzeugen, die nicht über die eigentlichen Konfigurationsparameter von KoPS 3.1 hergestellt werden können. Eine Beschreibung der einzelnen Parameter kann auf der Web-Oberfläche jeweils über den Hilfe-Button angezeigt werden.

eGK NFDM

Über den für die Anwendung NFDM spezifischen Menü-Eintrag *eGK NFDM* gelangt der Nutzer zu der Verwaltung der eGKs für das Notfalldatenmangement. Für Jede Karte kann sowohl ihr fachlicher Inhalt als auch ihr Verhalten festgelegt werden. Für die Inhalte gibt es Eingabefelder, die eine vereinfachte Bearbeitung der Notfalldaten (NFD) und persönlichen Erklärungen (DPE) erlauben.

In dem Reiter XML ist es möglich, über eine Formular- oder eine XML-Ansicht die Inhalte des Notfalldatensatzes und des Datensatzes persönliche Erklärung einzusehen oder zu verändern. Dabei wird in der Formularansicht die Struktur des XML-Datensatzes als Baumstruktur angezeigt. Je nach Ebene können die Elemente des XML-Datensatzes bearbeitet werden. Über ein Kontextmenü in

der Darstellung der Baumstruktur können zusätzlichen Elemente eingefügt werden bzw. existierende Element gelöscht werden. Vor dem Speichern muss ausgewählt werden, ob die Daten aus der Formularansicht oder der XML-Ansicht in die Testdaten übernommen werden sollen.

eGK eMP/AMTS

Über den für die Anwendung eMP/AMTS spezifischen Menü-Eintrag *eGK eMP/AMTS* gelangt der Nutzer zu der Verwaltung der eGKS für die Anwendung eMP/AMTS. Für jede Karte kann sowohl ihr fachlicher Inhalt als auch ihr Verhalten festgelegt werden. Für die Inhalte gibt es Eingabefelder, die eine vereinfachte Bearbeitung der eMP-Daten und der Einwilligung erlauben.

In dem Reiter XML ist es möglich, über eine Formular- oder eine XML-Ansicht die Inhalte des Medikationsplanes und der Einwilligung einzusehen oder zu verändern. Dabei wird in der Formularansicht die Struktur des XML-Datensatzes als Baumstruktur angezeigt. Je nach Ebene können die Elemente des XML-Datensatzes bearbeitet werden. Über ein Kontextmenü in der Darstellung der Baumstruktur können zusätzlichen Elemente eingefügt werden bzw. existierende Element gelöscht werden. Vor dem Speichern muss ausgewählt werden, ob die Daten aus der Formularansicht oder der XML-Ansicht in die Testdaten übernommen werden sollen.

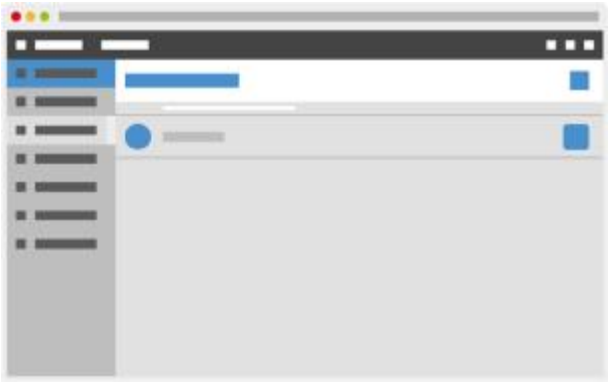
eGK/ePA

Über den für die Anwendung ePA spezifischen Menü-Eintrag *eGK/ePA* gelangt der Nutzer zu der Verwaltung der eGKs und zugehörigen Patientenakten für die Anwendung ePA. Für jede Karte kann sowohl ihr fachlicher Inhalt als auch ihr Verhalten festgelegt werden.

In dem Reiter ePA ist es möglich, Grundeinstellungen zum Aktenkonto des jeweiligen Versicherten vorzunehmen. Dazu gehören Daten zum Versicherten wie Vorname, Name, Geschlecht, Geburtsdatum und Versichertennummer sowie Daten

zum Aktenkonto wie Status des Aktenkontos, Zugriffsberechtigung (Art, Dauer) und die Home-CommunityID.

Praxiskarten QES

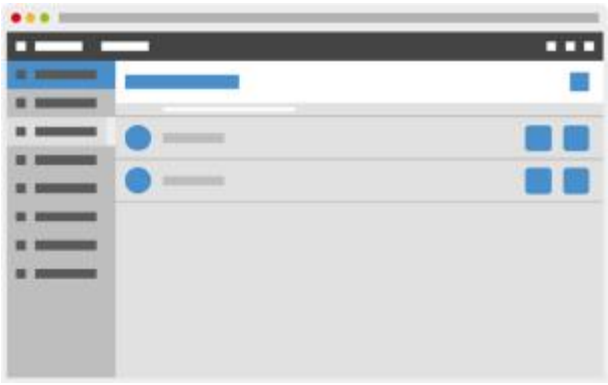


Über den für die Anwendung QES spezifischen Menü-Eintrag *Praxis / QES* gelangt der Nutzer zu der Verwaltung der Praxiskarten für die Anwendung QES. Für jede Karte kann sowohl ihr fachlicher Inhalt als auch ihr Verhalten festgelegt werden.

In dem Reiter QES ist es möglich, die Grundeinstellungen zur QES und das Antwortverhalten der kartenbasierten Operationen (*SignDocument, DecryptDocument, EncryptDocument, ExternalAuthenticate, ReadCardCertificate, VerifyCertificate*) für die Basisdienste QES festzulegen.

Beim Verschlüsselungsdienst sind die Antworten der Operationen DecryptDocument und Encrypt Document statisch in KoPS hinterlegt. Die zugehörigen entschlüsselten Dokumente können über die Schaltfläche *Dateien speichern* aus KoPS heruntergeladen werden.

Praxiskarten KIM/VZD



Über den für die Anwendung KIM spezifischen Menü-Eintrag *Praxis / VZD* gelangt der Nutzer zu der Verwaltung der Praxiskarten für die Anwendung KIM und die zugehörigen Verzeichnisdiensteinträge. Für jede Karte kann sowohl ihr fachlicher Inhalt als auch ihr Verhalten festgelegt werden.

In dem Reiter POP3 ist es möglich, einen POP3-Account für einen Benutzer anzulegen und zu verwalten, sowie das Antwortverhalten für den POP3-Account festzulegen. Analog kann über den Reiter SMTP der zugehörige SMTP-Account angelegt und verwaltet werden. Der Benutzername muss dabei gemäß der Bildungsregel für SMTP

und POP3 Benutzernamen [gemILF_PS# Tab_ILF_PS_Bildungsregel_SMTP-POP3_Benutzername] der E-Mail-Adresse des Benutzers entsprechen. Die Server-URI (Domain-Adresse) kann in KoPS frei gewählt werden (z.B. mail.kim.telematik.de). Abhängig vom Benutzername und der Server-URI erzeugt KoPS die entsprechenden Postfächer für den Benutzer.

In dem Reiter VZD werden die zugehörigen Einträge im Verzeichnisdienst für den Benutzer hinterlegt. Dabei muss mindestens eine E-Mail-Adresse des Benutzers gemäß dem Benutzernamen (siehe oben) als KIM Adresse hinterlegt werden, damit das Primärsystem nach der zugehörigen E-Mail-Adresse per LDAP im VZD suchen kann.
Über den Eintrag KIM-Version kann zwischen den KIM-Versionen 1.0 und 1.5 für das Verhalten der E-Mail Kommunikation eingestellt werden. Falls keine KIM-Version hinterlegt ist, wird standardmäßig die KIM-Version 1.0 angenommen.
Mit der KIM-Version 1.5 ist die Verarbeitung von Anhängen > 25 MB möglich.

Postfächer (KIM)

Über den für die Anwendung KIM spezifischen Menü-Eintrag *Postfächer* gelangt der Nutzer zu der Übersicht der Postfächer für die Anwendung KIM. In der Übersicht werden die angelegten POP3-und SMTP-Accounts dargestellt.

Für die POP3-Accounts können E-Mails zum Abruf durch das E-Mail-Modul des Clientsystems angelegt und verwaltet werden. Zusätzlich können Anhänge an die E-Mails hinterlegt werden.
Falls die KIM-Version 1.konfiguriert ist, können Anhänge >25 MB verarbeitet werden.
Über das Feld KIM-Dienstkennung kann eine spezifische Dienstkennung für die E-Mail aus einer Auswahlliste hinterlegt werden.

Test-Modus

Der Test-Modus enthält sämtliche Menü-Einträge, die für Zustandsänderungen der simulierten Umgebung relevant sind. So können hier beispielsweise Karten gesteckt und gezogen werden, Verbindungen getrennt, Fehler provoziert, Zeitverhalten manipuliert und Ereignisse ausgelöst werden. Darüber hinaus können in diesem Modus die linearen Testfälle ausgeführt werden.

Neben den allgemeinen Menüeinträgen, die für alle Fachanwendungen gültig sind, gibt es im Test-Modus zusätzlich bei KIM die anwendungsspezifischen Menüeinträge *Postfächer* und *VZD-Einträge* und bei TBAuth den anwendungsspezifischen Menüeintrag *TBAuth*.

Aufrufkontext

Über den Menü-Eintrag *Aufrufkontexte* im Test-Modus gelangt man zu einer Listenansicht im Hauptbereich. Bei den dort gelisteten Einträgen handelt es sich um die im Admin-Modus konfigurierten Aufrufkontexte. Pro Aufrufkontext kann in die jeweilige *Live-Terminal-Ansicht* dieses Kontextes gewechselt werden. Dazu muss auf das Terminal-Icon eines Eintrages geklickt werden.

In der *Live-Terminal-Ansicht* hat der Nutzer die Möglichkeit, in angezeigten Terminals (analog zu einem physischen Terminal) Karten zu stecken oder zu entfernen. Dazu stehen dem Nutzer die vordefinierten Versicherten- und Praxiskarten der jeweiligen Anwendung zur Auswahl.

Das Stecken der Karte erfolgt per Drag'n'Drop. Im rechten Bereich kann eine Karte ausgewählt und auf einen der freien grauen Slots im linken Bereich gezogen werden. Dadurch wird diese Karte in den jeweiligen Slot gesteckt. Soll die Karte aus dem Terminal gezogen werden, kann hierfür die entsprechende Schaltfläche des Terminals verwendet werden.

In der Live-Terminal-Ansicht kann auch die physikalische Trennung der Verbindung vom Kartenterminal zum Konnektor simuliert werden.

Pro Aufrufkontext kann zu der Verwaltung des *Übersteuerten Antwortverhaltens* gewechselt werden. Diese Übersteuerung gilt entsprechend nur für den aktuellen Aufrufkontext. Der Nutzer hat die Möglichkeit, das Antwortverhalten für verschiedene Methoden zu manipulieren. So kann eingestellt werden, ob ein SOAP-Fehler oder eine technisch manipulierte Antwort statt der eigentlichen Antwort zurückgegeben werden soll. Die Antwort lässt sich dabei auch in Gänze durch eine benutzerdefinierte Antwort ersetzen. Zusätzlich kann die Ausführung einer Methode auch verzögert werden. Eine solche Verzögerung kann vor oder nach der fachlichen Bearbeitung der Methode erfolgen.

In der Listenansicht der Aufrufkontexte können diese mit einem Klick deaktiviert oder aktiviert werden. Ein deaktivierter Aufrufkontext ist über die Konnektor-Schnittstelle nicht nutzbar. Somit können Aufrufkontexte einfach deaktiviert werden, ohne die zugrundeliegende Konfiguration entfernen zu müssen.

Abonnenten

Unter dem Menü-Eintrag *Abonnenten* können alle Abonnenten des Ereignisdienstes eingesehen werden, die sich über die Ereignisdienst-Schnittstelle des Konnektors bei KoPS 3.1 registriert haben. Pro Abonnent wird angezeigt, mit welchen Parametern sich dieser registriert hat. Zusätzlich ist es möglich, bestehende Abonnenten zu entfernen. Somit lässt sich zum Beispiel eine Situation simulieren, in welcher der Konnektor aufgrund von Netzwerk-Problemen dreimal Ereignisse nicht erfolgreich abschicken konnte und daher den Abonnenten aus seiner Liste entfernt hat.

Ereignisse

Unter dem Menü-Eintrag *Ereignisse* hat der Nutzer die Möglichkeit, alle spezifizierten Ereignisse manuell auszulösen. Für jedes Ereignis sind verschiedene Konfigurationsparameter möglich (siehe Konnektor-Spezifikation).

Die Werte eines zuvor ausgelösten Ereignisses werden gespeichert und sind beim nächsten Auswählen des Ereignisses vorausgewählt.

Testlabor

Unter dem Menü-Eintrag *Testlabor* kann der Nutzer die vordefinierten linearen Testfälle der jeweiligen Anwendung durchführen. Das Testlabor enthält eine Übersicht der vorhandenen *Testsets* (z.B. des *gematik Testfallkatalogs*). Durch einen Klick auf das Testset werden dessen Details angezeigt. Hier besteht die Möglichkeit eine Testfallübersicht über die im Set enthaltenen Testfälle und deren jeweiligen Status zu erhalten. Die Testfallübersicht kann als PDF heruntergeladen werden. Das PDF enthält die Anzahl der erfolgreichen, der fehlerhaften und der nicht unterstützten sowie der nicht durchgeführten Testfälle und die Einzelergebnisse der Testfälle im Testset. Alternativ kann dazu auch eine Excel-Liste heruntergeladen werden, die neben der Zusammenfassung auch die Definitionen der Testfälle enthält. Um alle Testfallergebnis PDFs und die Testfallübersicht als Paket herunter zu laden, bietet KoPS 3.1 eine ZIP-Datei an. Mit einem Klick auf das rechtsbündig angezeigte Icon wird die Übersicht der im Set enthaltenen Testfälle angezeigt.

Handelt es sich um einen gematik Testfallkatalog, enthält der Detailbereich des Testsets einen Abschnitt für den gematik Testbericht. Hier können die Werte *Primärsystem Name*, *Primärsystem Version*, *Aktenzeichen* der Leistungserbringerorganisation, *gematik ZLS* und der *Name des Testers* gepflegt werden. Diese Werte werden bei der Generierung des Testberichts und der Testfallergebnis-PDFs verwendet. Für die Generierung des Testberichts müssen mindestens Primärsystem Name, Primärsystem Version und der Name des Testers angegeben sein. Der offizielle *Testbericht* kann als reines PDF oder als Paket mit allen Testfallergebnis-PDFs in Form einer ZIP-Datei heruntergeladen werden. Zusätzlich können im Detailbereich des Testsets auch alle bisherigen Testfallergebnisse des Testsets komplett zurückgesetzt werden.

In der Übersicht der Testfälle besteht die Möglichkeit, durch einen Klick auf einen Testfall dessen Details anzuzeigen. Zu diesen Details gehören der Name des Testfalls, eine Kurzbeschreibung, sowie eine ausführliche Beschreibung. Handelt es sich um einen als optional markierten Testfall (von einer SOLL- oder KANN-Anforderung abgeleitet) im Testset, kann der Nutzer diesen als durch sein

Primärsystem nicht unterstützt kennzeichnen. Diese Kennzeichnung erfolgt über eine Checkbox „Testfall wird durch das Primärsystem nicht unterstützt.“ in der Detailansicht des jeweiligen Testfalls. Wurde der Testfall bereits durchgeführt, wird zusätzlich der Zeitpunkt der letzten Testausführung angezeigt. Im Menüband unterhalb der Schaltfläche „Modus wechseln“ kann durch einen Klick auf das jeweilige Icon direkt eine Testfallübersicht als auch der Testbericht (nur innerhalb des gematik-Testfallkatalogs) als ZIP-Datei heruntergeladen werden.

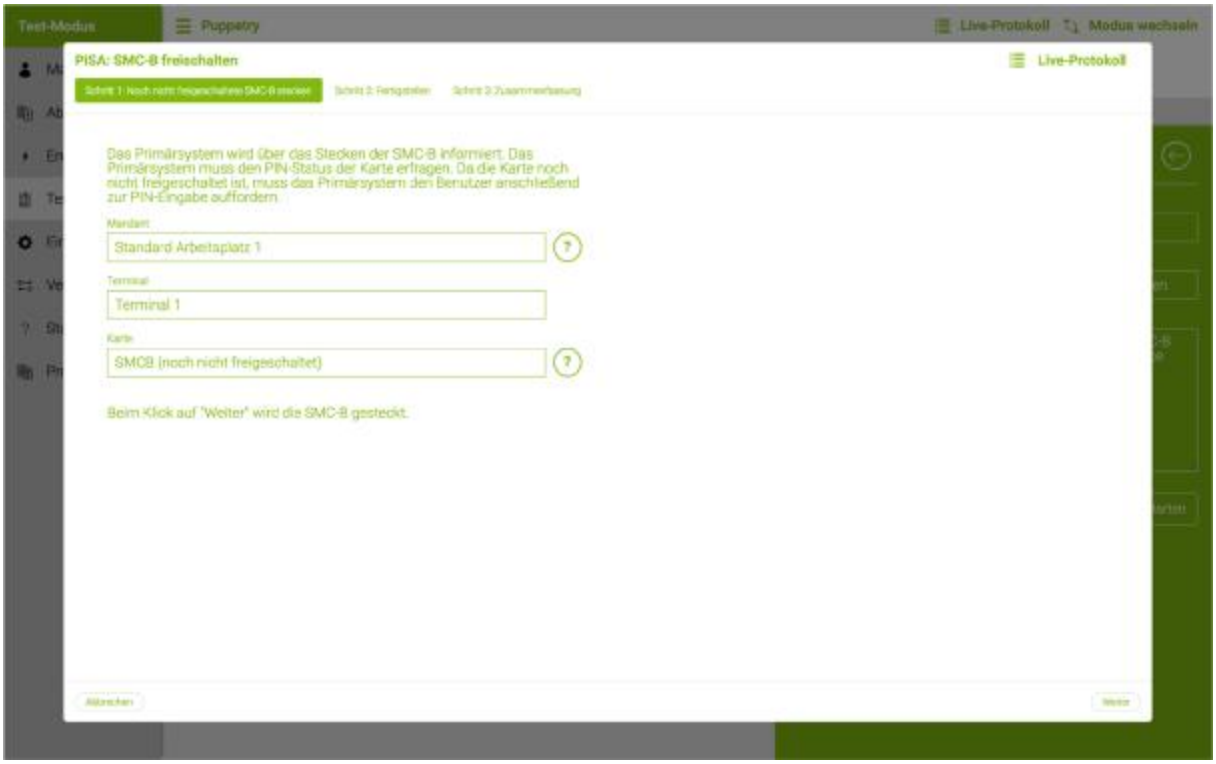
Für jeden durchgeführten Testfall kann das Testfallergebnis und die dazugehörigen Log-Dateien des letzten Testdurchlaufes abgerufen werden. Zusätzlich ist es möglich, eine Zusammenfassung über alle Testfälle als PDF-Bericht herunterzuladen. In diesem Bericht ist jeder Testfall einzeln mit seinem letzten Ergebnis aufgeführt.

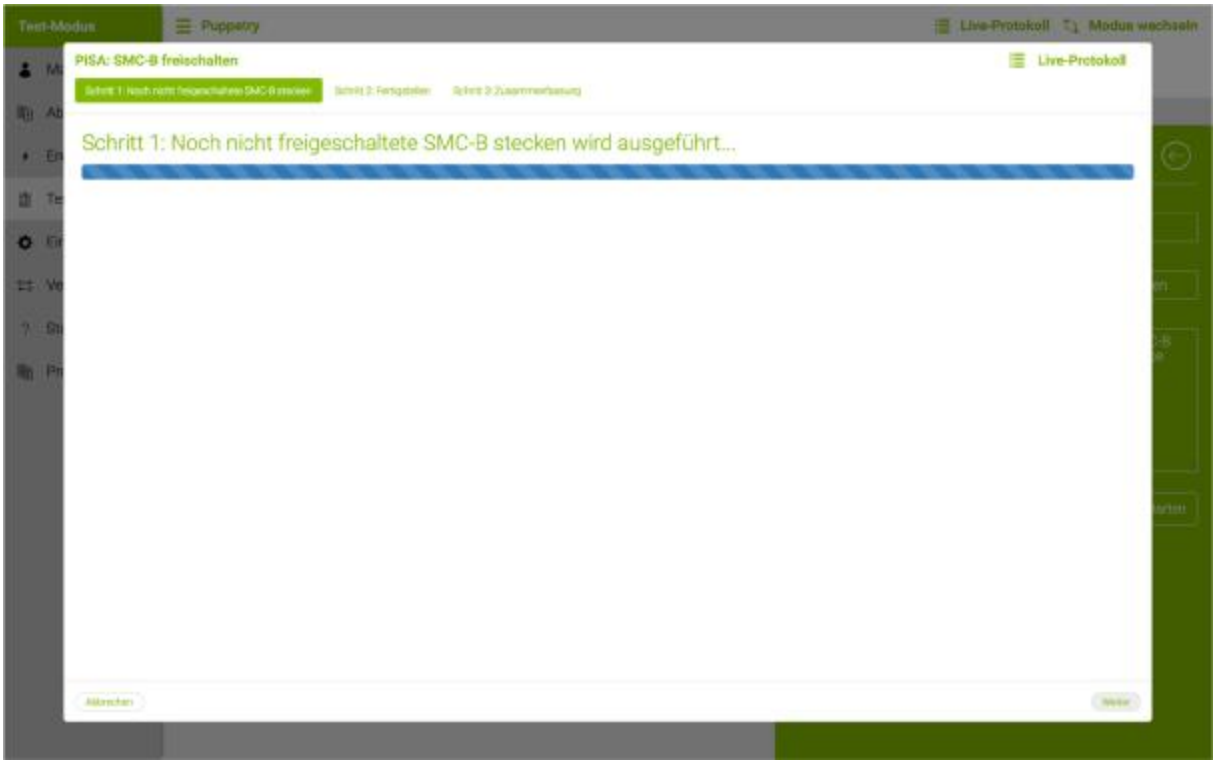
Nach dem Start eines Testfalls wird dem Nutzer ein Pop-Up angezeigt, welches ihn Schritt-für-Schritt durch den Testfall führt. Es wird immer der Name des Testfalls (die Kurzbeschreibung kann via Tooltip auf dem Namen eingeblendet werden) und der aktuelle Testfallschritt angezeigt. Die Anzahl der Schritte ist vom jeweiligem Testfall abhängig. Nach Abschluss des Testfalls wird ein Testfallergebnis erstellt. Dieses Testfallergebnis umfasst im PDF eingebettet auch sämtliche Log-Dateien (diese können in einem entsprechenden PDF-Reader eingesehen werden).

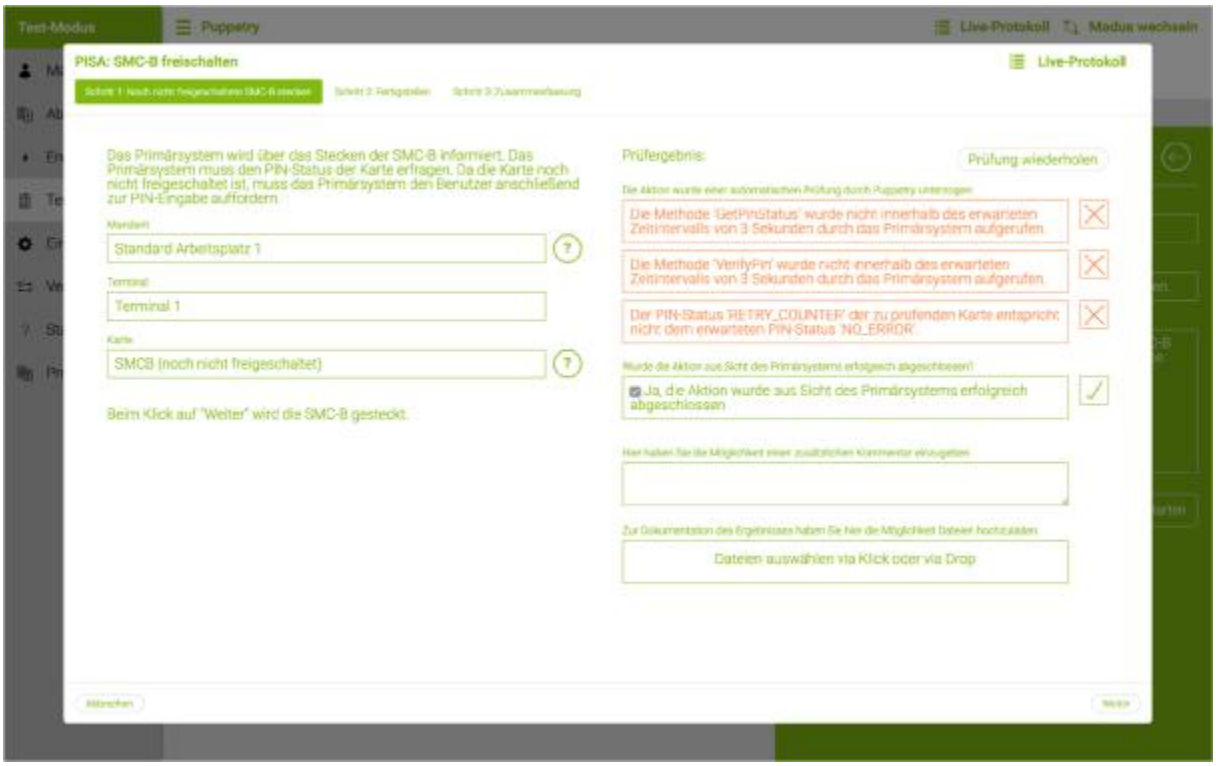
Zu Beginn eines Testfalls wird automatisch ein initialer Zustand hergestellt. Das heißt, es werden alle Karten gezogen bzw. die für den Test relevanten Karten gesteckt und ggf. Einstellungen wie der zu verwendende Authentifizierungsmechanismus angepasst. Nach Beendigung des Testfalls wird wieder der Zustand vor der Ausführung des Testfalls hergestellt. Also die in dem Testfall benutzten Karten werden gezogen und die vor der Testfallausführung steckenden Karten werden wieder gesteckt. Auch die im Rahmen des Testfalls angepassten Einstellungen werden zurückgesetzt (inkl. ggf. notwendigem Neustart von KoPS 3.1).

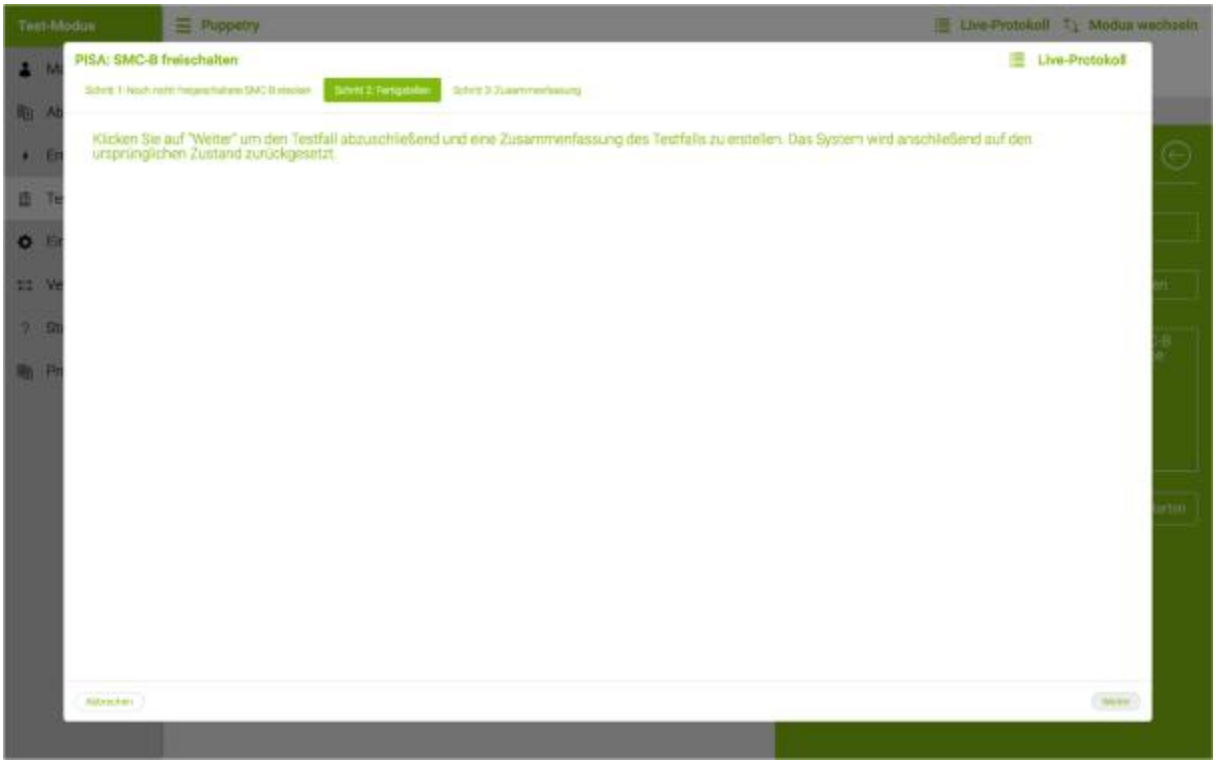
Bei einem Testfall kann es zu einer technischen Validierung durch KoPS 3.1 kommen (beispielsweise, ob eine bestimmte Methode vom Primärsystem aufgerufen wurde). In der Regel ist es möglich, eine Validierung erneut anzustoßen, sofern die Methode nicht innerhalb des dafür vorgesehenen Zeitfensters vom Primärsystem aufgerufen wurde. Neben der technischen Validierung kann der Nutzer aufgefordert werden, eine Überprüfung des Verhaltens am Primärsystem vorzunehmen. In diesem Fall bietet KoPS 3.1 bei dem zugehörigen Eintrag eine Checkbox an, über die die Beobachtung zum erwarteten Verhalten am Primärsystem aktiv bestätigt werden muss.

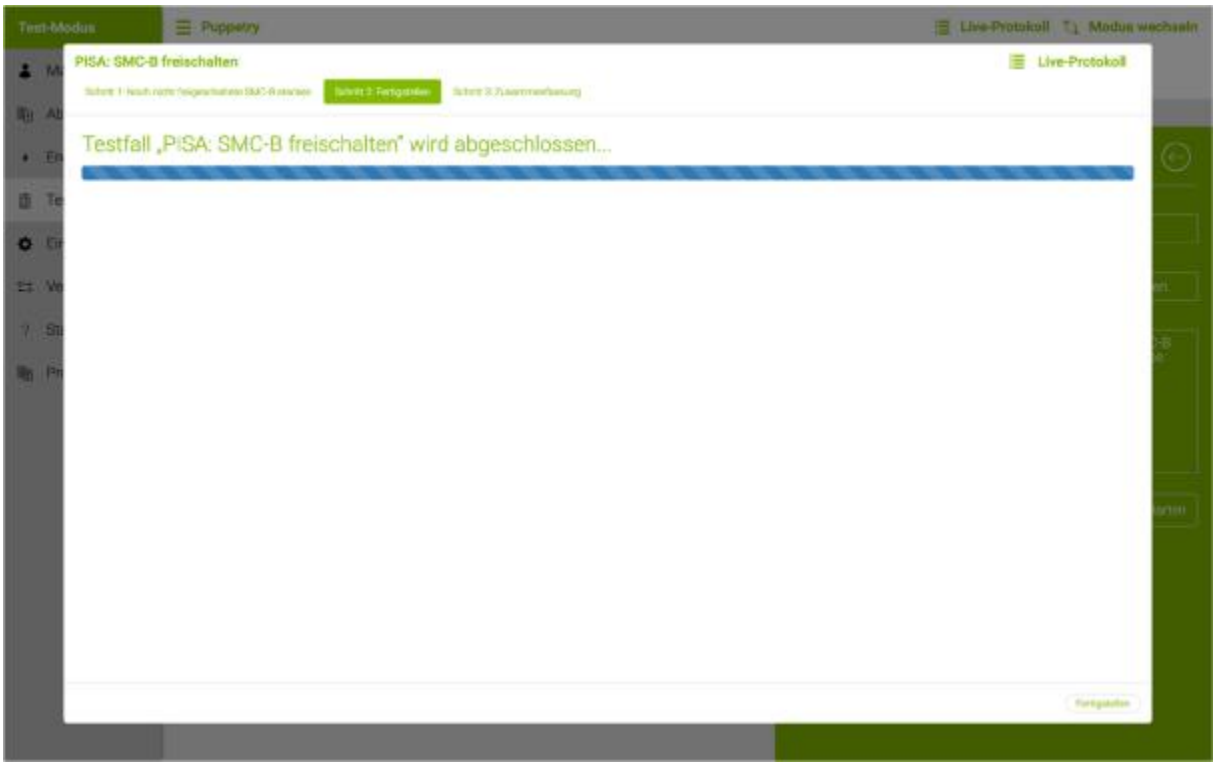
Die nachfolgenden Bilder zeigen einen beispielhaften Ablauf eines Testfalls:

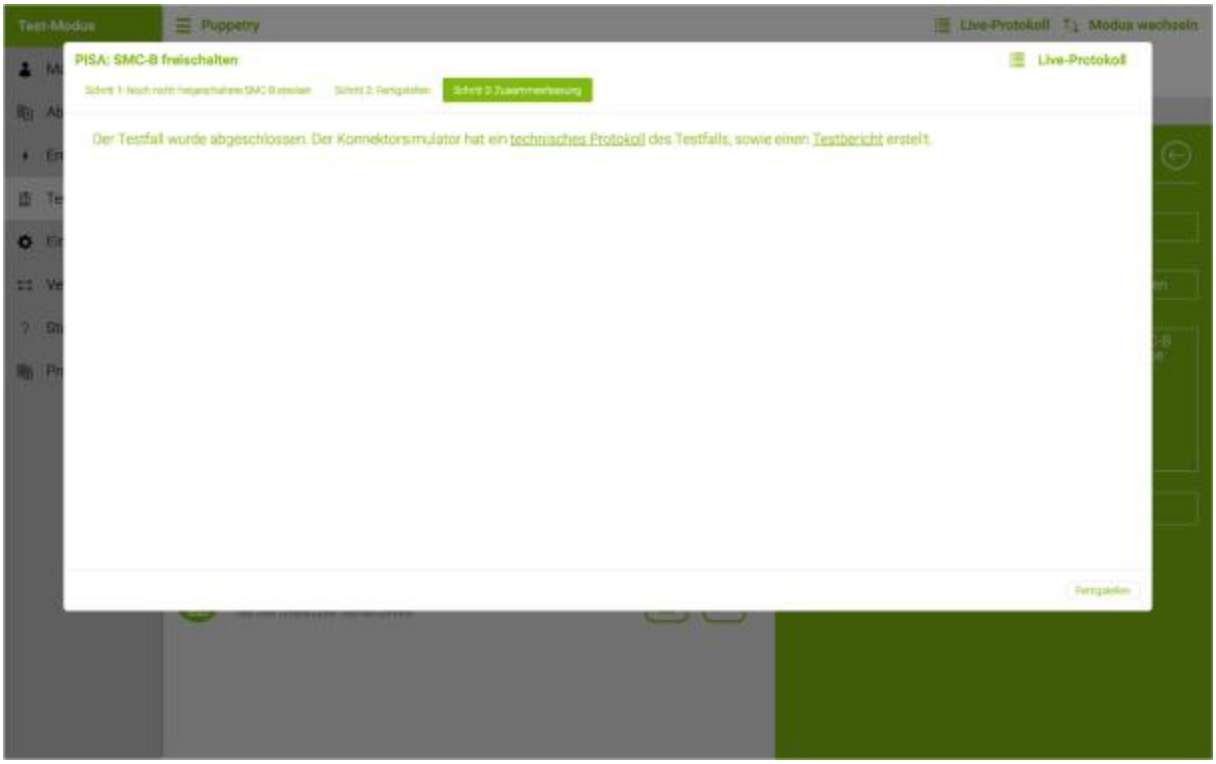





Test-Modus
Puppetry
Live-Protokoll
Modus wechseln
PISA: SMC-B freischalten
Live-Protokoll
Das Primärsystem wird über das Stecken der SMC-B informiert. Das Primärsystem muss den PIN-Status der Karte erfragen. Da die Karte noch nicht freigeschaltet ist, muss das Primärsystem den Benutzer anschließend zur PIN-Eingabe auffordern.
Mandant
Standard Arbeitsplatz 1
Terminal
Terminal 1
Karte
SMCB (noch nicht freigeschaltet)
Beim Klick auf "Weiter" wird die SMC-B gesteckt.
Prüfergebnis:
Prüfung wiederholen
Die Aktion wurde einer automatischen Prüfung durch Puppetry unterzogen:
Die Methode 'GetPinStatus' wurde nicht innerhalb des erwarteten Zeitintervalls von 3 Sekunden durch das Primärsystem aufgerufen.
Die Methode 'VerifyPin' wurde nicht innerhalb des erwarteten Zeitintervalls von 3 Sekunden durch das Primärsystem aufgerufen.
Der PIN-Status 'RETRY_COUNTER' der zu prüfenden Karte entspricht nicht dem erwarteten PIN-Status 'NO_ERROR'.
Wurde die Aktion aus Sicht des Primärsystems erfolgreich abgeschlossen?
Ja, die Aktion wurde aus Sicht des Primärsystems erfolgreich abgeschlossen
Hier haben Sie die Möglichkeit einen zusätzlichen Kommentar einzugeben.
Zur Dokumentation des Ergebnisses haben Sie hier die Möglichkeit Dateien hochzuladen.
Dateien auswählen via Klick oder via Drop


Test-Modus
Puppetry
Live-Protokoll
Modus wechseln
PISA: SMC-B freischalten
Live-Protokoll
Klicken Sie auf "Weiter" um den Testfall abzuschließend und eine Zusammenfassung des Testfalls zu erstellen. Das System wird anschließend auf den ursprünglichen Zustand zurückgesetzt.


Test-Modus
Puppetry
Live-Protokoll
Modus wechseln
PISA: SMC-B freischalten
Live-Protokoll
Testfall „PISA: SMC-B freischalten" wird abgeschlossen...


Test-Modus
Puppetry
Live-Protokoll
Modus wechseln
PISA: SMC-B freischalten
Live-Protokoll
Schritt 3: Zusammenfassung
Der Testfall wurde abgeschlossen. Der Konnektorsimulator hat ein technisches Protokoll des Testfalls, sowie einen Testbericht erstellt.

Postfächer (KIM)

Über den für die Anwendung KIM spezifischen Menü-Eintrag *Postfächer* im Test-Modus gelangt man zu einer Listenansicht im Hauptbereich. Bei den dort gelisteten Einträgen handelt es sich um die im Admin-Modus konfigurierten POP3- und SMTP-Postfächer der zugehörigen Praxiskarten.

Das E-Mail-Modul des Clientsystems kann über eine SMTP-Kommunikation E-Mails an KoPS senden. Der Konnektorsimulator stellt dem Clientsystem diese Operation über den TCP-Port 10465 ausschließlich über SMTPS bereit. KoPS implementiert hierbei das SMTP-Protokoll gemäß RFC 5321 und nimmt nach erfolgreicher Authentisierung eingehende E-Mails entgegen und stellt diese im zugehörigen SMTP-Postfach dar. Neben der eigentlichen E-Mail-Nachricht wird auch die Information gespeichert, ob das Clientsystem Lese- und/oder Zustellbestätigungen angefordert hat.

Das Clientsystem kann sich über SMTPS mittels der SASL-Mechanismen PLAIN und LOGIN am KoPS authentisieren. Hierfür müssen zuvor im Clientsystem entsprechende Anmeldedaten festgelegt werden. Diese Anmeldedaten müssen analog dem Schema des KIM Clientmoduls, welches in Abbildung 1 dargestellt ist, entsprechen.

Abbildung 1: Aufbau SMTP-Benutzername

Aus dem POP3-Postfach von KoPS können die hinterlegten E-Mails vom E-Mail-Modul des Clientsystems über POP3-Kommunikation abgerufen werden. Für diese Kommunikation stellt der KoPS dem Clientsystem ausschließlich eine POP3S-Schnittstelle auf dem TCP-Port 10995 bereit.
Hierbei wird das POP3-Protokoll implementiert und so dem Clientsystem die Möglichkeit gegeben, nach erfolgreicher Authentisierung, statisch hinterlegte E-Mail-Nachrichten abzurufen. Diese kön-

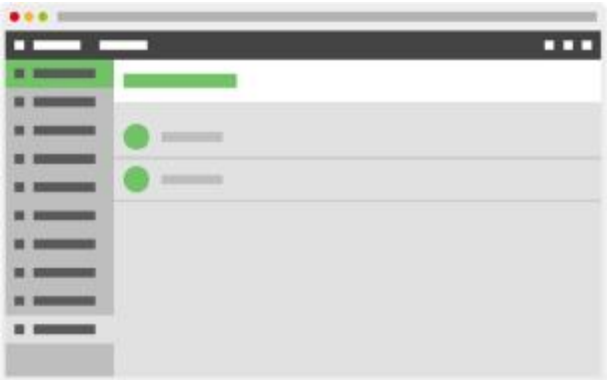
nen mit beliebigen Anhängen sowie beliebiger Größen im KoPS hinterlegt und so vom Clientsystem abgerufen werden. So können z. B. Empfangsbestätigungen, nicht entschlüsselte KIM-Nachrichten oder signierte eArztbriefe als Anhang im KoPS hinterlegt und vom Clientsystem abgerufen werden.

Darüber hinaus werden E-Mails, die über die SMTP-Schnittstelle an eine gültige E-Mail-Adresse aus dem VZD in KoPS gesendet werden, temporär im POP3-Postfach des jeweiligen Empfängers hinterlegt und können von dort abgerufen werden.

Das Clientsystem kann sich über POP3S mittels der USER/PASS und SASL-PLAIN-Mechanismen am KoPS authentisieren. Hierfür müssen im Clientsystem entsprechende Anmeldedaten festgelegt werden. Diese Anmeldedaten müssen analog dem Schema des KIM Clientmoduls, welches in Abbildung 2 dargestellt ist, entsprechen.

Abbildung 2: Aufbau POP3-Benutzername

VZD-Einträge (KIM)



Über den für die Anwendung KIM spezifischen Menü-Eintrag *VZD-Einträge* im Test-Modus gelangt man zu einer Übersicht aller VZD-Einträge in Listenansicht. Nach Auswahl einer Zeile, lassen sich die Inhalte des Eintrages im Detailbereich anzeigen.

TBAuth (TBAuth)

Über den für die Anwendung TBAuth spezifischen Menü-Eintrag *TBAuth* im Test-Modus kann man Einstellungen für den Aufruf und das Antwortverhalten der Fachanwendung TBAuth vornehmen.

Vom Modus und Anwendung unabhängige Menü-Einträge und Funktionen in der Kopfzeile

Neben den Menü-Einträgen, die nur im Admin-Modus beziehungsweise im Test-Modus zu sehen sind, gibt es Menü-Einträge, die dem Nutzer immer zur Verfügung stehen, da sie für beide Betriebsmodi und alle Anwendungen relevant sind. Darüber können die Funktionen in der Kopfzeile für alle Anwendungen und Betriebsmodi genutzt werden.

Einstellungen

Im Menü-Eintrag *Einstellungen* können Anpassungen bezüglich der Konnektor-Schnittstelle, des Ereignisdienstes und der Anzeigeeinstellung der Web-Oberfläche vorgenommen werden.

Für die *Konnektor-Schnittstelle* kann konfiguriert werden, über welches Interface und welchen HTTPS Port diese erreichbar ist. Gleiches gilt für die Dienste, jedoch kann hier zusätzlich eingestellt werden, über welchen Authentifizierungsmechanismus die Konnektor-Dienste erreichbar sind.

Für den *Dienstverzeichnisdienst* kann eingestellt werden, ob dieser weiterhin über HTTP erreichbar ist, während die Konnektor-Dienste nur über eine TLS-gesicherte Verbindung angebunden sind. Funktional kann für den Dienstverzeichnisdienst das Antwortverhalten gesteuert werden, es kann eine unterschiedliche Dienstversion oder eine angepasste Dienst-URL provoziert werden.

Für den *Ereignisdienst* des Primärsystems kann gleichfalls der Authentifizierungsmechanismus festgelegt, sowie für den Ereignisdienstclient das Kommunikationsverhalten verändert werden. Da das konkrete Kommunikationsverhalten von dem verwendeten Konnektor abhängt, kann somit der Umgang des Primärsystems mit verschiedenen möglichen Verhalten getestet werden. Um die Kommunikation des Ereignisdienstes unabhängig von der Nutzung des Systeminformationsdienstes (Subscribe) testen zu können, kann eine statische Ereignis-Endpunkt-Adresse angegeben werden. Ist diese Adresse eingerichtet, werden alle auftretenden Ereignisse – unabhängig von Abonnements – immer an diese Adresse geschickt. Um das Erneuern von Abonnements zu testen, kann die Gültigkeitsdauer eines Abonnements angepasst werden (Standard 25 Stunden). Zur Simulation von Netzwerkproblemen besteht die Möglichkeit zur Unterdrückung von Ereignis-Nachrichten an das Primärsystem.

Für die Web-Oberfläche kann zusätzlich noch eingestellt werden, in welchem Intervall das Live-Protokoll aktualisiert und in welchem Intervall die Live-Terminal-Anzeige aktualisiert wird (z.B. um über das programmatische Ziehen einer Karte informiert zu werden).

Protokolle

Unter dem Menü-Eintrag *Protokolle* können die zur Laufzeit durch KoPS 3.1 angelegten Log-Dateien *Basis* (enthält alle Log-Einträge der KoPS 3.1-Software), *SOAP* (enthält alle SOAP-Nachrichten) und *Schnittstelle* (entspricht den Einträgen im Live-Protokoll) heruntergeladen werden. Zu jedem Protokoll-Typ kann die aktuelle Log-Datei oder eine komprimierte Version, die zusätzlich alle vergangenen Log-Dateien beinhaltet, heruntergeladen werden.

Status

Unter dem Kopfzeilen Icon *Konnektorstatus* bzw. *Verbindungsstatus* kann der Nutzer den Status der Konnektor-Dienste und den Status der virtuellen Verbindung zur Telematikinfrastruktur ändern. Wird der Konnektor-Status auf *AUS* gestellt, stellt dies die Situation nach, in der ein physischer Konnektor ausgeschaltet beziehungsweise vom Strom genommen wird. Wird der Verbindungsstatus auf *OFFLINE* gestellt, stellt dies die Situation nach, in der bei einem physischen Konnektor die Kabelverbindung zur Telematikinfrastruktur unterbrochen wird.

Systeminfo

Unter dem Kopfzeilen Icon *Systeminfo* erhält der Nutzer genauere Informationen zur der aktuellen Installation von KoPS 3.1. Dazu gehört die Version der Software, der Konfiguration und die Lizenzinformationen.

Benutzerkonfiguration

Das Konfigurationsverzeichnis von KoPS 3.1 befindet sich im Unterordner *conf* des Installationsverzeichnisses – hier erfolgt in den spezifischen Unterverzeichnissen die Ablage von Testdaten- und Testfallkonfigurationen.
Bei den Testdaten wird zwischen anwendungsübergreifenden Testdaten und anwendungsspezifischen Testdaten unterschieden. Die anwendungsübergreifenden Testdaten (Aufrufkontext, Kartenterminals, Clientsystem, Praxiskarten) können von allen Anwendungen genutzt werden. Diese Testdaten werden standardmäßig im Unterordner Environment abgelegt.
Die anwendungsspezifischen Testdaten (z.B. VSDM, NFDM) werden in den jeweiligen Unterordnern der Anwendung mit einem spezifischen Dateinamen abgelegt (*testData_[Anwendung].xml*).
Die Testfallkataloge sind immer anwendungsspezifisch und werden ebenfalls standardmäßig unter den jeweiligen Unterordnern der Anwendung abgelegt und dem spezifischen Dateinamen abgelegt (*testCases_[Anwendung].xml)*.

Im Auslieferungszustand befindet sich im Verzeichnis *gematik* die geschützte Konfiguration, welche für die Bestätigung notwendig ist. Im Verzeichnis *user* werden die vom Nutzer mittels KoPS 3.1-Interface erstellten Testdaten abgelegt. Die Unterverzeichnisse enthalten die Testdaten und Testfallkonfigurationen für die jeweilige Anwendung.

Ein eigenes Unterverzeichnis mit einer *testCases_[Anwendung].xml* Datei wird im *Testlabor* jeweils als ein eigenständiges *Testset* angezeigt. Dabei ergibt sich der Name des *Testsets* aus dem Namen des Unterverzeichnisses.

Wenn verschiedene Testdatenkonfigurationsdateien verwendet werden, ist es wichtig, dass die gewählten technischen IDs der Konfigurationsobjekte über alle Konfigurationsdateien hinweg eindeutig sind. Sollten mehrere Testdatenkonfigurationsdateien vorhanden sein, welche die gleichen IDs verwenden, blockiert dies den Start von KoPS 3.1. Eine entsprechende Fehlermeldung wird dann beim Ausführen des Programmes angezeigt. Verzeichnisse oder Testdaten sollten darum niemals innerhalb des Verzeichnisses lediglich kopiert werden, sondern bedürfen auch immer einer Anpassung durch den Nutzer.

Weitere Hinweise

Umsetzung der Schnittstellen der Anwendung ePA

In KoPS 3.1 sind die Schnittstellen der ePA-Administration (PHRManagement) dynamisch umgesetzt und können über die entsprechenden Operationsaufrufe an der Schnittstelle von KoPS vom Primärsystem angesprochen werden. Dazu gehören folgenden Funktionalitäten der ePA-Administration:

- Aktenanbieter ermitteln (GetHomeCommunityId)
- Aktenkonto aktivieren (ActivateAccount)
- ad-hoc Berechtigung erteilen (RequestFacilityAuthorization)
- Liste der Berechtigungen abfragen (GetAuthorizationList)

Die Verwaltung der Zugriffsberechtigungen auf die Aktenkonten erfolgt statisch über die Stammdaten der eGK/ePA und ist nicht nach verschiedenen Leistungserbringern (SMC-Bs) differenziert.

In KoPS 3.1 sind die Schnittstellen des Dokumentenmanagements (PHRService) nur statisch über den Testfallkatalog zu testen, da keine Anbindung an eine gematik-konforme IHE-Akte besteht.

Alle Requests zu den Operationen des Patientenmanagements PHRService im Rahmen der "freien" Testung außerhalb des Testfallkataloges werden in KoPS 3.1 mit einem gematik SOAP-Fehler beantwortet. Dazu wird folgender Fehlercode verwendet:

- Fehlercode 7200 Lokalisierung des Aktensystems fehlgeschlagen

Glossar

Konnektor

Der Konnektor koordiniert und verschlüsselt die Kommunikation zwischen Clientsystem, eGK, HBA/SMC und zentraler Telematikinfrastruktur. Er stellt damit das Bindeglied zwischen diesen Komponenten auf Leistungserbringerseite bzw. eKiosk und Telematikinfrastruktur dar. Der Konnektor ist ein Produkttyp.

SMC-B (Institutionskarte)

Die Institutionskarte entspricht technisch weitgehend dem Heilberufsausweis (HBA), bezieht sich jedoch auf eine organisatorische Instanz des Gesundheitswesens (z.B. Praxis, Apotheke, Krankenhaus). Die Institutionskarte wird auch als Security Module Card Typ B (SMC-B) bezeichnet.

Terminal (Kartenterminal)

Migrationsfähige (entsprechen der aktuellen eHealth-Kartenterminal Spezifikation [gemSpec_KT]) Kartenlesegeräte, welche zusätzlich eine USB- bzw. V24-Schnittstelle unterstützen, sowie mit einem Upgrade ohne Austausch der Geräte zu einem vollwertigen LAN-fähigen „eHealth-KT“ aufgerüstet werden können. Kartenlesegeräte auf dieser Basis müssen an der V.24- und/oder USB-Schnittstelle mindestens den „Basis Command Set (BCS)“ unterstützen.

Primärsystem

Ein IT-System, das bei einem Leistungserbringer eingesetzt wird – z.B. eine Praxisverwaltungssoftware (PVS), ein Krankenhausinformationssystem (KIS) oder eine Apothekensoftware (AVS) – und sich unter dessen administrativer Hoheit befindet. Das Primärsystem ist kein Bestandteil der TI-Plattform.

Quelle: gematik Glossar
https://fachportal.gematik.de/fileadmin/user_upload/fachportal/files/Spezifikationen/Methodische_Festlegungen/gemGlossar_V400.pdf

Anhang 1: Unterstützte Signaturvarianten

KoPS 3.1 unterstützt alle Signaturvarianten gemäß [gemSpec_Kon#Tabelle TAB_KON_778] für den Einsatzbereich nonQES und QES, die an der Außenschnittstelle vom Konnektor angeboten werden.

Tabelle 1: Unterstützte Signaturvarianten von KoPS 3.1

Signaturverfahren	Signaturvariante	WAS wird signiert	WO wird die Signatur abgelegt
XAdES*	enveloped	gesamtes Input XML-Dokument (= Root-Element mit Subelementen)	Als direktes Child des Root-Elements
XAdES	enveloping	gesamtes Input XML-Dokument (= Root-Element mit Subelementen)	Im Dokument, das Root-Element umschließend
CAdES*	detached	gesamtes Binärdokument	außerhalb des Dokuments in der SignResponse
CAdES	enveloping	gesamtes Binär-Dokument	innerhalb des CMS-Dokuments
PAdES*	-	gesamtes PDF-Dokument	Im PDF-Dokument

*=Default Signaturverfahren

Die verschiedenen Signaturvarianten werden durch die Aufrufparameter der Operation SignDocument (nonQES und QES) `SignatureType, IncludeEContent, IncludeObject` und `SignaturePlacement` gesteuert. Dabei werden die nachfolgenden Aufrufparameter gemäß [gemSpec_Kon#Tabelle TAB_KON_065] unterstützt.

Tabelle 2: Unterstützte Aufrufparameter der Operation SignDocument

Aufrufparameter	Beschreibung
`CONN: CardHandle`	Identifiziert die zu verwendende Signaturkarte.
`CCTX: Context`	Aufrufkontext QES mit HBAx: MandantId, ClientSystemId, WorkplaceId, UserId verpflichtend Aufrufkontext nonQES mit SM-B: MandantId, ClientSystemId, WorkplaceId verpflichtend; UserId nicht ausgewertet
`TvMode`	Der Parameter wird im Konnektor nicht ausgewertet.

SIG: JobNumber	Die Nummer des Jobs, unter der der nächste Signaturvorgang gestartet wird. Parameter ist verpflichtend.
SIG: SignRequest	Ein SignRequest kapselt den Signaturauftrag für ein Dokument. Das verpflichtende XML-Attribut RequestID identifiziert einen SignRequest innerhalb eines Stapels von SignRequests eindeutig. Es dient der Zuordnung der SignResponse zum jeweiligen SignRequest.
SIG: OptionalInputs	Enthält optionale Eingangsparameter (angelehnt an dss:OptionalInputs gemäß [OASIS-DSS] Section 2.7).
SIG: Document	Dieses an das dss:Document Element aus [OASIS-DSS] Section 2.4.2 angelehnte Element enthält das zu signierende Dokument, wobei die Kindelemente CONN:Base64XML und dss:Base64Data auftreten können. Bei den als dss:Base64Data übergebenen Dokumenten werden folgende (Klassen von) MIME-Types unterschieden: · "application/pdf-a" – für PDF/A-Dokumente, · "text/plain", "text/plain; charset=iso-8859-15" oder "text/plain; charset=utf-8" – für Text-Dokumente, · "image/tiff" – für TIFF-Dokumente und · ein beliebiger anderer MIME-Type für nicht näher unterschiedene Binärdaten des spezifizierten Typs. Der MIME-Type „text/plain" wird interpretiert als „text/plain; charset=iso-8859-15". Das Element enthält ein Attribut ShortText Es muss für QES-Signaturen bei jedem Aufruf vom Clientsystem übergeben werden, für nonQES-Signaturen ist es optional. Über das Attribut RefURI kann gemäß [OASIS-DSS] (Abschnitt 2.4.1) ein zu signierender Teilbaum eines XML-Dokuments ausgewählt werden (nur bei QES NFDM).
SIG: Include RevocationInfo	Durch diesen verpflichtenden Schalter kann der Aufrufer die Einbettung von zum Zeitpunkt der Signaturerstellung vorliegenden Sperrinformationen anfordern. Es wird ausschließlich die zu erstellende Signatur betrachtet, d.h. es erfolgt keine Einbettung von Sperrinformationen für bereits enthaltene Signaturen. Für nicht-qualifizierte elektronische Signaturen (nonQES) wird diese Funktionalität nicht unterstützt.
dss: Signatur Type	Durch dieses in [OASIS-DSS] (Abschnitt 3.5.1) beschriebene Element kann der generelle Typ der zu erzeugenden Signaturen spezifiziert werden. Hierbei MÜSSEN folgende Signaturtypen unterstützt werden: • XML-Signatur

	Durch Übergabe der URI urn:ietf:rfc:3275 wird die Erstellung von XML-Signaturen gemäß [RFC3275], [XMLDSig] angestoßen. Das zu verwendende Profil ist XAdES-BES ([XAdES]). Die Rückgabe einer solchen Signatur erfolgt als `ds:Signature`-Element. • CMS-Signatur Durch Übergabe der URI urn:ietf:rfc:5652 wird eine CMS-Signatur gemäß [RFC5652] angestoßen. Das zu verwendende Profil ist CAdES-BES ([CAdES]). Die Signatur wird als `dss:Base64Signature` mit der oben genannten URI als `Type` zurückgeliefert. • S/MIME-Signatur Durch Übergabe der URI „urn:ietf:rfc:5751" wird eine S/MIME-Signatur gemäß [RFC5751] angestoßen. Die CMS-Signatur der übergebenen MIME-Nachricht erfolgt konform der Vorgaben zur CMS-Signatur. Das Rückgabedokument ist eine MIME-Nachricht vom Typ „application/pkcs7-mime" mit einer CMS-Struktur vom Typ_SignedData. • PDF-Signatur Durch Übergabe der URI http://uri.etsi.org/02778/3 wird die Erzeugung einer PAdES-Basic Signatur gemäß [PAdES-3] angestoßen, wobei das Dokument mit der integrierten Signatur als `dss:Base64Signature` mit der oben genannten URI als `Type` zurückgeliefert wird.
`SIG: Include EContent`	Durch dieses in [OASIS-DSS] (Abschnitt 3.5.7), definierte Element kann bei einer CMS-basierten Signatur das Einfügen des signierten Dokumentes in die Signatur angefordert werden.
`SIG: Include Object`	Dieses Element enthält zum Anfordern einer Enveloping XML Signatur ein dss:IncludeObject-Element gemäß [OASIS-DSS] (Abschnitt 3.5.6).
`dss: Signature Placement`	Durch dieses in [OASIS-DSS] (Abschnitt 3.5.8) definierte Element kann bei XML-basierten Signaturen gemäß [RFC3275] die Platzierung der Signatur im Dokument angegeben werden.
`dss: Return Updated Signature`	Durch dieses in [OASIS-DSS] (Abschnitt 4.5.8) definierte Element kann eine übergegebene XML- oder CMS-Signatur mit zusätzlichen Informationen und Signaturen (Parallel- und Gegensignaturen) versehen werden. Hierbei sind folgende Ausprägungen für das `Type`-Attribut vorgesehen:

	· http://ws.gematik.de/conn/sig/sigupdate/parallel Hierdurch wird eine Parallelsignatur zu einer bereits existierenden Signatur erzeugt und entsprechend zurückgeliefert. · http://ws.gematik.de/conn/sig/sigupdate/counter/ documentexcluding Hierdurch wird eine dokumentenexkludierende Gegensignatur für alle vorhandenen parallelen Signaturen erzeugt.
`sp: GenerateUnderSignaturePolicy`	Über dieses in [OASIS-SP], Kapitel 2.2.1.1.1.1 Optional Input <GenerateUnderSignaturePolicy>, definierte Element wird die erforderliche Singnaturrichlinie ausgewählt. Die im Element `sp:SignaturePolicyIdentifier` übergebene URI identifiziert die Signaturrichtlinie. Dieses Element wird nur für NFDM ausgewertet.

Die verschiedenen Varianten der Operation VerifyDocument werden bei enveloped Signaturen über das übergebene Dokument oder bei detached und enveloping Signaturen durch den Aufrufparameter `SignatureType` gesteuert. Dabei werden die nachfolgenden Aufrufparameter gemäß [gemSpec_Kon#Tabelle TAB_KON_066] unterstützt.

Tabelle 3: Unterstützte Aufrufparameter der Operation VerifyDocument

Aufrufparameter	Beschreibung
`CCTX: Context`	MandantId, ClientSystemId, WorkplaceId verpflichtend; UserId nicht ausgewertet
`TvMode`	Der Parameter wird im Konnektor nicht ausgewertet.
`SIG: Optional Inputs`	Enthält optionale Eingabeparameter (angelehnt an dss:OptionalInputs gemäß [OASIS-DSS] Section 2.7): Die zulässigen optionalen Eingabeparameter sind unten erläutert.
`SIG: Document`	Enthält im Fall der Prüfung von detached oder enveloped Signaturen das zur Signatur gehörende bzw. das diese umschließende Dokument (siehe [OASIS-DSS] Section 2.4.2 und oben).
`dss: SignatureObject`	Enthält die zu prüfende Signatur, wenn sie nicht im Dokument selbst eingebettet ist ([OASIS-DSS] Kapitel 4.1). Hierbei werden XML-Signaturen als `ds:Signature` Element und alle anderen Signaturen als `dss:Base64Signature` mit entsprechend gesetztem `Type`-Attribut (siehe `SignatureType`, Operationen SignDocument und ExternalAuthenticate) übergeben, wobei die nachfolgenden Werte unterstützt werden müssen: · CMS-Signatur urn:ietf:rfc:5652 · S/MIME-Signatur urn:ietf:rfc:5751

	· PDF-Signatur http://uri.etsi.org/02778/3 · PKCS#1-Signatur urn:ietf:rfc:3447
SIG: Include RevocationInfo	Durch diesen verpflichtenden Schalter kann der Aufrufer die Einbettung von zum Zeitpunkt der Signaturprüfung vorliegenden Sperrinformationen anfordern. Ist bereits eine Sperrinformation eingebettet, so wird die neue Sperrinformation zusätzlich eingebettet. Für in einer Gegensignatur enthaltene Signaturen erfolgt keine Einbettung von Sperrinformationen.
vr: Return VerificationReport	Durch dieses in [OASIS-VR] spezifizierte Element kann die Erstellung eines ausführlichen Prüfberichtes angefordert werden.

Beispiele SignDocument-Requests

Nachfolgend sind Beispiele für den unterschiedlichen Aufbau von SignDocument-Requests für die verschiedenen Signaturvarianten aufgeführt.

Beispiel 1: XaDES detached Signatur (QES NFDM)

```
...
   <v7:SignDocument>
     <v5:CardHandle>HBA-1</v5:CardHandle>
     <v2:Context>
       <v5:MandantId>Mandant1</v5:MandantId>
       <v5:ClientSystemId>ClientID1</v5:ClientSystemId>
       <v5:WorkplaceId>Workplace1</v5:WorkplaceId>
       <v5:UserId>User01</v5:UserId>
     </v2:Context>
     <v7:TvMode>NONE</v7:TvMode>
     <v7:JobNumber>JOB-006</v7:JobNumber>
     <v7:SignRequest RequestID="ID01">
       <v7:OptionalInputs>
         <urn:SignatureType>urn:ietf:rfc:3275</urn:SignatureType>
         <urn:SignaturePlacement WhichDocument="NFD_DOC_ID" CreateEnvelopedSignature="false">
           <urn:XPathFirstChildOf>/*[local-name()='NFD_Document']/*[local-name()='SignatureArzt']</urn:XPath-
FirstChildOf>
         </urn:SignaturePlacement>
         <urn1:GenerateUnderSignaturePolicy>
           <urn1:SignaturePolicyIdentifier>urn:gematik:fa:sak:nfdm:r1:v1</urn1:SignaturePolicyIdentifier>
         </urn1:GenerateUnderSignaturePolicy>
       </v7:OptionalInputs>
       <v7:Document ID="NFD_DOC_ID" RefURI="#ID1" ShortText="XaDES detached">
         <v5:Base64XML>[base64-codiertes XML-Dokument] </v5:Base64XML>
       </v7:Document>
       <v7:IncludeRevocationInfo>true</v7:IncludeRevocationInfo>
     </v7:SignRequest>
   </v7:SignDocument>
```

```
...
```

Beispiel 2: XaDES enveloped Signatur (direktes Child des Root-Elementes)

```
...
    <v7:SignDocument>
      <v5:CardHandle>HBA-1</v5:CardHandle>
      <v2:Context>
        <v5:MandantId>Mandant1</v5:MandantId>
        <v5:ClientSystemId>ClientID1</v5:ClientSystemId>
        <v5:WorkplaceId>Workplace1</v5:WorkplaceId>
        <v5:UserId>User01</v5:UserId>
      </v2:Context>
      <v7:TvMode>NONE</v7:TvMode>
      <v7:JobNumber>JOB-001</v7:JobNumber>
      <v7:SignRequest RequestID="ID_01">
        <v7:OptionalInputs>
          <urn:SignatureType>urn:ietf:rfc:3275</urn:SignatureType>
          <urn:SignaturePlacement WhichDocument="DOC_ID_01" CreateEnvelopedSignature="true" >
            <urn:XPathFirstChildOf>/*</urn:XPathFirstChildOf>
          </urn:SignaturePlacement>
        </v7:OptionalInputs>
        <v7:Document ID="DOC_ID_01" RefURI="" ShortText="XaDES enveloped Root">
          <v5:Base64XML>[base64-codiertes XML-Dokument]</v5:Base64XML>
        </v7:Document>
        <v7:IncludeRevocationInfo>true</v7:IncludeRevocationInfo>
      </v7:SignRequest>
    </v7:SignDocument>
...
```

Beispiel 3: XaDES enveloping Signatur

```
...
    <v7:SignDocument>
      <v5:CardHandle>HBA-1</v5:CardHandle>
      <v2:Context>
        <v5:MandantId>Mandant1</v5:MandantId>
        <v5:ClientSystemId>ClientID1</v5:ClientSystemId>
        <v5:WorkplaceId>Workplace1</v5:WorkplaceId>
        <v5:UserId>User01</v5:UserId>
      </v2:Context>
      <v7:TvMode>NONE</v7:TvMode>
      <v7:JobNumber>JOB-001</v7:JobNumber>
      <v7:SignRequest RequestID="ID_01">
        <v7:OptionalInputs>
          <urn:SignatureType>urn:ietf:rfc:3275</urn:SignatureType>
          <v7:IncludeObjects>
            <urn:IncludeObject WhichDocument="DOC_ID_01”/>
          </v7:IncludeObjects>
        </v7:OptionalInputs>
        <v7:Document ID="DOC_ID_01" RefURI="" ShortText="XaDES enveloping">
          <v5:Base64XML>[base64-codiertes XML-Dokument]</v5:Base64XML>
        </v7:Document>
        <v7:IncludeRevocationInfo>true</v7:IncludeRevocationInfo>
      </v7:SignRequest>
    </v7:SignDocument>
...
```

Beispiel 4: CaDES detached Signatur

```
...
    <v7:SignDocument>
      <v5:CardHandle>HBA-1 </v5:CardHandle>
      <v2:Context>
        <v5:MandantId>Mandant1</v5:MandantId>
        <v5:ClientSystemId>ClientID1< v5:ClientSystemId>
        <v5:WorkplaceId>Workplace1 </v5:WorkplaceId>
        <v5:UserId>User01</v5:UserId>
      </v2:Context>
      <v7:TvMode>NONE</v7:TvMode>
      <v7:JobNumber>JOB-001<v7:JobNumber>
      <v7:SignRequest RequestID="ID_01">
        <v7:OptionalInputs>
          <urn:SignatureType>urn:ietf:rfc:5652</urn:SignatureType>
        </v7:OptionalInputs>
        <v7:Document ShortText="CaDES detached">
          <urn:Base64Data MimeType="text-plain">[base64-codiertes CMS-Dokument]</urn:Base64Data>
        </v7:Document>
        <v7:IncludeRevocationInfo>true</v7:IncludeRevocationInfo>
      </v7:SignRequest>
    </v7:SignDocument>
...
```

Beispiel 5: CaDES enveloping Signatur

```
...
    <v7:SignDocument>
      <v5:CardHandle>HBA-1 </v5:CardHandle>
      <v2:Context>
        <v5:MandantId>Mandant1</v5:MandantId>
        <v5:ClientSystemId>ClientID1< v5:ClientSystemId>
        <v5:WorkplaceId>Workplace1 </v5:WorkplaceId>
        <v5:UserId>User01</v5:UserId>
      </v2:Context>
      <v7:TvMode>NONE</v7:TvMode>
      <v7:JobNumber>JOB-001<v7:JobNumber>
      <v7:SignRequest RequestID="ID_01">
        <v7:OptionalInputs>
          <urn:SignatureType>urn:ietf:rfc:5652</urn:SignatureType>
          <v7:IncludeEContent>true</v7:IncludeEContent>
        </v7:OptionalInputs>
        <v7:Document ShortText="CaDES enveloping">
          <urn:Base64Data MimeType="text/plain">[base-64 codiertes CMS-Dokument]</urn:Base64Data>
        </v7:Document>
        <v7:IncludeRevocationInfo>true</v7:IncludeRevocationInfo>
      </v7:SignRequest>
    </v7:SignDocument>
...
```

Beispiel 6: PaDES Signatur

```
...
    <v7:SignDocument>
      <v5:CardHandle>HBA-1 </v5:CardHandle>
      <v2:Context>
        <v5:MandantId>Mandant1</v5:MandantId>
        <v5:ClientSystemId>ClientID1< v5:ClientSystemId>
        <v5:WorkplaceId>Workplace1 </v5:WorkplaceId>
        <v5:UserId>User01</v5:UserId>
      </v2:Context>
      <v7:TvMode>NONE</v7:TvMode>
      <v7:JobNumber>JOB-001<v7:JobNumber>
      <v7:SignRequest RequestID="ID_01">
        <v7:OptionalInputs>
          <urn:SignatureType>http://uri.etsi.org/02778/3</urn:SignatureType>
        </v7:OptionalInputs>
        <v7:Document ShortText="PaDES">
          <urn:Base64Data MimeType="application/pdf-a">[base64-codiertes PDF]</urn:Base64Data>
        </v7:Document>
        <v7:IncludeRevocationInfo>true</v7:IncludeRevocationInfo>
      </v7:SignRequest>
    </v7:SignDocument>
...
```

Beispiele für VerifyDocument-Requests

Nachfolgend sind Beispiele für den unterschiedlichen Aufbau von VerifyDocument-Requests für die Standard-Signaturvarianten aufgeführt.

Beispiel 8: VerifyDocument XaDES enveloped

```
...
<v7:VerifyDocument>
      <v2:Context>
        <v5:MandantId>${#Project#MandantId}</v5:MandantId>
        <v5:ClientSystemId>${#Project#ClientSystemId}</v5:ClientSystemId>
        <v5:WorkplaceId>${#Project#WorkplaceId}</v5:WorkplaceId>
        <v5:UserId>${#Project#UserId}</v5:UserId>
      </v2:Context>
      <v7:TvMode>NONE</v7:TvMode>
      <v7:OptionalInputs>
      <urn1:ReturnVerificationReport>
          <urn1:IncludeVerifier>false</urn1:IncludeVerifier>
          <urn1:IncludeCertificateValues>true</urn1:IncludeCertificateValues>
          <urn1:IncludeRevocationValues>true</urn1:IncludeRevocationValues>
          <urn1:ExpandBinaryValues>false</urn1:ExpandBinaryValues>
          <urn1:ReportDetailLevel>urn:oasis:names:tc:dss:1.0:profiles:verificationreport:reportdetail:allDetails
           </urn1:ReportDetailLevel>
        </urn1:ReturnVerificationReport>
      </v7:OptionalInputs>
      <v7:Document ID="XMLDOC">
        <v5:Base64XML>[base64 codiertes XML-Dokument mit enveloped Signatur]</v5:Base64XML>
      </v7:Document>
      <v7:IncludeRevocationInfo>false</v7:IncludeRevocationInfo>
    </v7:VerifyDocument>
...
```

Beispiel 9: VerifyDocument CaDES detached

```
...
  <v7:VerifyDocument>
    <v2:Context>
      <v5:MandantId>${#Project#MandantId}</v5:MandantId>
      <v5:ClientSystemId>${#Project#ClientSystemId}</v5:ClientSystemId>
      <v5:WorkplaceId>${#Project#WorkplaceId}</v5:WorkplaceId>
      <v5:UserId>${#Project#UserId}</v5:UserId>
    </v2:Context>
    <v7:TvMode>NONE</v7:TvMode>
    <v7:OptionalInputs>
    <urn1:ReturnVerificationReport>
        <urn1:IncludeVerifier>false</urn1:IncludeVerifier>
        <urn1:IncludeCertificateValues>true</urn1:IncludeCertificateValues>
        <urn1:IncludeRevocationValues>true</urn1:IncludeRevocationValues>
        <urn1:ExpandBinaryValues>false</urn1:ExpandBinaryValues>
        <urn1:ReportDetailLevel>urn:oasis:names:tc:dss:1.0:profiles:verificationreport:reportdetail:allDetails
        </urn1:ReportDetailLevel>
      </urn1:ReturnVerificationReport>
    </v7:OptionalInputs>
       <v7:Document ID="ID01" ShortText="CMS">
       <urn:Base64Data MimeType="text/plain">[base64-codiertes CMS-Dokument]</urn:Base64Data>
    </v7:Document>
     <urn:SignatureObject>
      <urn:Base64Signature Type="urn:ietf:rfc:5652">[zugehörige Signatur]< urn:Base64Signature>
    </urn:SignatureObject>
    <v7:IncludeRevocationInfo>false</v7:IncludeRevocationInfo>
  </v7:VerifyDocument>
 </soapenv:Body>
</soapenv:Envelope>
...
```

Beispiel 10: VerifyDocument PaDES

```
...
   <v7:VerifyDocument>
     <v2:Context>
       <v5:MandantId>${#Project#MandantId}</v5:MandantId>
       <v5:ClientSystemId>${#Project#ClientSystemId}</v5:ClientSystemId>
       <v5:WorkplaceId>${#Project#WorkplaceId}</v5:WorkplaceId>
       <v5:UserId>${#Project#UserId}</v5:UserId>
     </v2:Context>
     <v7:TvMode>NONE</v7:TvMode>
     <v7:OptionalInputs>
     <urn1:ReturnVerificationReport>
         <urn1:IncludeVerifier>false</urn1:IncludeVerifier>
         <urn1:IncludeCertificateValues>true</urn1:IncludeCertificateValues>
         <urn1:IncludeRevocationValues>true</urn1:IncludeRevocationValues>
         <urn1:ExpandBinaryValues>false</urn1:ExpandBinaryValues>
         <urn1:ReportDetailLevel>urn:oasis:names:tc:dss:1.0:profiles:verificationreport:reportdetail:allDetails
         </urn1:ReportDetailLevel>
       </urn1:ReturnVerificationReport>
     </v7:OptionalInputs>
     <v7:Document ID="PDFDOC">
       <urn:Base64Data MimeType="application/pdf-a">[signiertes PDF-Dokument] </urn:Base64Data>
     </v7:Document>
     <v7:IncludeRevocationInfo>false</v7:IncludeRevocationInfo>
   </v7:VerifyDocument>
...
```

Beispiel 11: VerifyDocument S/MIME

```
...
   <v7:VerifyDocument>
     <v2:Context>
       <v5:MandantId>${#Project#MandantId}</v5:MandantId>
       <v5:ClientSystemId>${#Project#ClientSystemId}</v5:ClientSystemId>
       <v5:WorkplaceId>${#Project#WorkplaceId}</v5:WorkplaceId>
       <v5:UserId>${#Project#UserId}</v5:UserId>
     </v2:Context>
     <v7:TvMode>NONE</v7:TvMode>
     <v7:OptionalInputs>
     <urn1:ReturnVerificationReport>
         <urn1:IncludeVerifier>false</urn1:IncludeVerifier>
         <urn1:IncludeCertificateValues>true</urn1:IncludeCertificateValues>
         <urn1:IncludeRevocationValues>true</urn1:IncludeRevocationValues>
         <urn1:ExpandBinaryValues>false</urn1:ExpandBinaryValues>
         <urn1:ReportDetailLevel>urn:oasis:names:tc:dss:1.0:profiles:verificationreport:reportdetail:allDetails
         </urn1:ReportDetailLevel>
       </urn1:ReturnVerificationReport>
     </v7:OptionalInputs>
       <urn:SignatureObject>
         <urn:Base64Signature Type="urn:ietf:rfc:5751">">[signiertes S/MIME-Dokument]</urn:Base64Signature>
      </urn:SignatureObject>
     <v7:IncludeRevocationInfo>false</v7:IncludeRevocationInfo>
   </v7:VerifyDocument>
...
```