Remote Management Interface

Stationäres eHealth-Kartenterminal
ORGA 6141 online und ORGA Neo
ab Firmware V3.9.2



Inhaltsverzeichnis

Version History 5
Tabellenverzeichnis 6
Herausgeber / Editor 6
Copyrights 6
Open Source Software 6

1. Einleitung und Architektur 7
1.1 Hinweis zur Absicherung der Schnittstelle 7

2. Genereller Ablauf 8
2.1 Request -> Response 8
2.2 Subscription -> Notification, Cancellation 9
2.3 Verbindungsaufbau, Login und Keepalive 9
2.4 Firmware Update 10
2.5 Remote SMC-B PIN 11
2.6 Setzen und Vererben von Passwörtern 11
2.7 Remote Pairing 12

3. JSON-Objekte, Service Module und Methoden 13
3.1 Service übergreifende Beschreibung 13
3.1.1 Generische Beschreibung der verwendeten JSON-Objekte 13
3.1.1.1 Request 13
3.1.1.2 Response 14
3.1.1.3 Exception 14
3.1.1.4 Subscription 15
3.1.1.5 Cancellation 15
3.1.1.6 Notification 15
3.1.2 Implementierte Services und ihre Methoden 15
3.1.3 Implementierte Datentypen und deren Wertebereiche 17
3.1.4 Allgemeine Fehlermeldungen / Exceptions 18
3.2 Protokoll Version und Versionsinformationen der Services – Service API 18
3.2.1 Methoden, Parameter und Response Objekte 18
3.2.1.1 getVersionInfo - request 18
3.2.1.2 getVersionInfo - response 19
3.2.1.3 Fehlermeldungen / Exceptions 19
3.3 Authentication und Keepalive – Service: Auth 20
3.3.1 Methoden, Parameter und Response Objekte 20
3.3.1.1 basicAuth - request 20
3.3.1.2 basicAuth - response 21
3.3.1.3 keepAlive - request 21
3.3.1.4 keepAlive - response 22
3.3.1.5 close - request 22
3.3.1.6 close - response 22
3.3.1.7 basicAuthSetCredentials - request 23
3.3.1.8 basicAuthSetCredentials - response 23
3.3.2 Fehlermeldungen / Exceptions 24
3.4 Verwalten der Pairing-Blöcke – Service „Pairing“ 24
3.4.1 Methoden, Parameter und Response Objekte 24
3.4.1.1 getInfo - request 24
3.4.1.2 getInfo - response 24
3.4.1.3 deleteBlock - request 26
3.4.1.4 deleteBlock - response 26
3.4.1.5 deletePublicKey - request 26
3.4.1.6 deletePublicKey - response 27

3.4.2 Fehlermeldungen / Exceptions ... 27
3.5 Ausführen von Systemfunktionen – Services „System“ ... 27
3.5.1 Methoden, Parameter und Response Objekte ... 28
3.5.1.1 reboot - request ... 28
3.5.1.2 reboot - response ... 28
3.5.2 Fehlermeldungen / Exceptions ... 28
3.6 Verändern und Abfragen von Einstellungen - Service: Settings ... 28
3.6.1 Methoden, Parameter und Response Objekte ... 29
3.6.1.1 getProperties - request ... 29
3.6.1.2 getProperties - response ... 29
3.6.1.3 setProperties - request ... 30
3.6.1.4 setProperties - response ... 30
3.6.2 Fehlermeldungen / Exceptions ... 31
3.7 Firmware Update – Service: Update ... 31
3.7.1 Methoden, Parameter und Response Objekte ... 31
3.7.1.1 beginTransmission - request ... 31
3.7.1.2 beginTransmission - response ... 31
3.7.1.3 transmit - request ... 32
3.7.1.4 transmit - response ... 32
3.7.1.5 endTransmission - request ... 32
3.7.1.6 endTransmission - response ... 33
3.7.1.7 abort - request ... 35
3.7.1.8 abort - response ... 35
3.7.1.9 install - request ... 35
3.7.1.10 install - response ... 36
3.7.2 Fehlermeldungen / Exceptions ... 36
3.8 Remote SMC-B PIN Eingabe – Service: Smartcard ... 36
3.8.1 Methoden, Parameter und Response Objekte ... 37
3.8.1.1 getCardInfo - request ... 37
3.8.1.2 getCardInfo - response ... 38
3.8.1.3 pinVerificationTopic - subscription ... 39
3.8.1.4 pinVerificationTopic - subscription - response ... 40
3.8.1.5 pinVerificationTopic - subscription - cancellation ... 40
3.8.1.6 pinVerificationTopic – cancellation - response ... 41
3.8.1.7 pinVerificationEvent - notification ... 41
3.8.1.8 verifyPin - request ... 42
3.8.1.9 verifyPin - response ... 42
3.8.1.10 abortPinVerification - request ... 43
3.8.1.11 abortPinVerification - response ... 43
3.8.2 Fehlermeldungen / Exceptions ... 44
3.9 Remote Pairing – Service: Dialog ... 45
3.9.1 Methoden, Parameter und Response Objekte ... 45
3.9.1.1 dialogRequestTopic - subscription ... 45
3.9.1.2 dialogRequestTopic - subscription - response ... 45
3.9.1.3 dialogRequestTopic - subscription - cancellation ... 46
3.9.1.4 dialogRequestTopic – cancellation - response ... 46
3.9.1.5 dialogPromptEvent - notification ... 46
3.9.1.6 submit - request ... 48
3.9.1.7 submit - response ... 48
3.9.2 Fehlermeldungen / Exceptions ... 49

4. Settings - Parameter 50
4.1 LAN & NTP 50
4.2 VPN 52
4.3 SICCT Parameter 55
4.4 Netzwerkstatus / Verbindungsstatus 56
4.5 Benutzer Interface 57
4.6 Firmware Update per TFTP 58
4.7 Remote Management Interface / Remote Admin Passwort 58
4.8 Remote SMC-B PIN und Remote Pairing Feature 59
4.9 Selbstauskunft 60
4.10 Kartenslots 61
4.11 Geräte Karte gSMC-KT 62
4.12 Betriebsdaten / Statistik 63

5. Web-Applikation 66
5.1 Anbindung über eine VPN-Verbindung 66
5.2 Benutzerrollen & Berechtigungen 66
5.3 Webbrowser basierte Konfiguration 67
5.4 Web-Browser 70
5.5 Web-Applikation 71
5.5.1 Grundsätzlicher Aufbau 72
5.5.2 Grundsätzliche Handhabung 73

6. Referenzen 75
6.1 Spezifikationen 75

Version History

Version:	Änderungshistorie:	Autor:	Datum:
1.0.0	Erste offizielle Version	TSI/JBA	05.07.2024
1.0.1	Wertebereiche der PropertyIDs „Message Read Timeout“ in Kapitel 4 aktualisiert.	JBA	17.07.2024
1.0.2	Wertebereiche in den Abschnitten VPN und Netzwerkstatus in Kapitel 4 ergänzt und Formatierung ergänzt.	JBA	25.07.2024
1.0.3	Berechtigung der PropertyID „vendor_url_productInfo“ der FW3.9.0 angepasst	JBA	26.07.2024
1.0.4	Anpassungen bzgl. „net_vpn_client_name“ Rechte des Benutzers „Anonymous“ eingeschränkt	THE	11.02.2025
1.0.5	Beschreibungen für den Dialog Service ergänzt net_vpn_ipAddrSicctUnicast hinzugefügt	TSI/THE	21.02.2025
1.0.6	Aktualisierung Kapitel 5 – Aufnahme Hinweise zum WireGuard VPN	JBA	03.04.2025
1.0.7	Beschreibung zu sicct_ipAddrUnicast hinzugefügt Neuer Parameter „channelId“ in der pinInfo Struktur des Smardcard Service	THE/TSI	14.07.2025
1.0.8	In Kapitel 4.11 – „Geräte Karte gSMC-KT“ einen Hinweis ergänzt.	TSI	30.07.2025
1.0.9	In Kapitel 5 – Web-Applikation – Aktualisierung der Bilder	JBA	10.09.2025

Tabelle 1: Änderungshistorie

Tabellenverzeichnis

Tabelle 1: Änderungshistorie 5
Tabelle 2: Spezifikationen 75

Herausgeber / Editor

Worldline Healthcare GmbH
Konrad-Zuse-Ring 1
24220 Flintbek
WEEE DE 32266764

Tel.: 04347 90 11 111
E-Mail: kontakt.whc@worldline.com
Internet: www.worldline.com/de/healthcare

Copyrights

Copyright© 2023/2024/2025
Worldline Healthcare GmbH - Alle Rechte vorbehalten.
Alle Produkte oder Dienstleistungen, die in diesem Dokument genannt werden, sind Marken, Dienstleistungsmarken, eingetragene Marken oder eingetragene Dienstleistungsmarken der entsprechenden Eigentümer.

Kein Teil dieser Dokumentation darf ohne schriftliche Genehmigung der Worldline Healthcare GmbH kopiert, gesendet, übertragen, elektronisch gespeichert oder in eine andere Sprache übersetzt werden.

Die Worldline Healthcare GmbH behält sich das Recht auf die Änderung von Funktionen, Eigenschaften und technischen Angaben zu jeder Zeit und ohne vorherige Benachrichtigung vor.

Open Source Software

Eine Liste der verwendeten Open Source Software für die Firmware des Kartenterminals als auch für die Web-Applikation zum Remote Management Interface befindet sich als Download in der Web-Applikation im Reiter INFO unter dem Abschnitt SOFTWARE INFORMATIONEN.

1. Einleitung und Architektur

Das ORGA 6141 online / ORGA Neo bietet ab der Firmware Version V3.9.0 ergänzend zur lokalen auch eine Remote Management Schnittstelle (RMI) an. Über diese Schnittstelle kann entweder mit Hilfe der über das Kartenterminal ausgelieferten Web-Applikation (basierend auf HTML, CSS und JavaScript) in einem Webbrowser oder einer eigenen Remote Management Client Applikation (Terminal Management System, TMS) kommuniziert werden.



Die Grundidee hinter dem Design der Remote Management Schnittstelle ist ein typensicheres Remote-Procedure-Call Interface (RPC), wobei das Protokoll, welches für die Verbindung genutzt wird, das Secure WebSocket Protokoll (WSS) nach [RFC_6455] ist. Hierbei wird alles in Textform übertragen (Opcode: 0x01), der Subprotokoll-Name wird dabei nicht ausgewertet.

Die bereitgestellten Services, Methoden und deren Parameter werden im weiteren Verlauf des Dokumentes detailliert beschrieben. Eine Auflistung aller an der Schnittstelle verfügbaren Datenfelder (für den Service Settings) befindet sich im Anhang.

1.1 Hinweis zur Absicherung der Schnittstelle

Der gesamte Datenverkehr mit dem RMI erfolgt verschlüsselt. Der WebSocket Server des Kartenterminals ist hierzu ausschließlich auf dem TCP-Port 443 empfangsbereit und akzeptiert nur über das TLS-Protokoll 1.2 etablierte Verbindung vom Client. Hierbei weist sich der Server entweder mit einem im Terminal generierten ECC-Authentifizierungszertifikat auf Basis von NIST-Kurven oder mit dem ECC-Zertifikat einer gSMC-KT G2.1 auf Basis von Brainpool-Kurven gegenüber dem Client aus.

Der Client entscheidet in der TLS-Handshake-Phase, welches Authentifizierungszertifikat er bevorzugt. Das Terminal unterstützt für diesen Anwendungsfall, die TLS-Extension „Server Name Indication“ (SNI) aus [RFC_6066] Kapitel 3 JSON-Objekte, Service Module und Methoden. Soll das Authentifizierungs-zertifikat der gSMC-KT G2.1 verwendet werden, so muss der Client die Server Name Indication Extension unterstützen und als host_name die ICCSN der gSMC-KT präsentieren. Nur dann wird sich das Terminal mit dem ECC-Zertifikat der gSMC-KT authentisieren. In allen anderen Fällen, wird das auf

dem Terminal generierte Zertifikat dem Client präsentiert. Der Client hingegen muss sich nicht mit einem Zertifikat authentisieren.

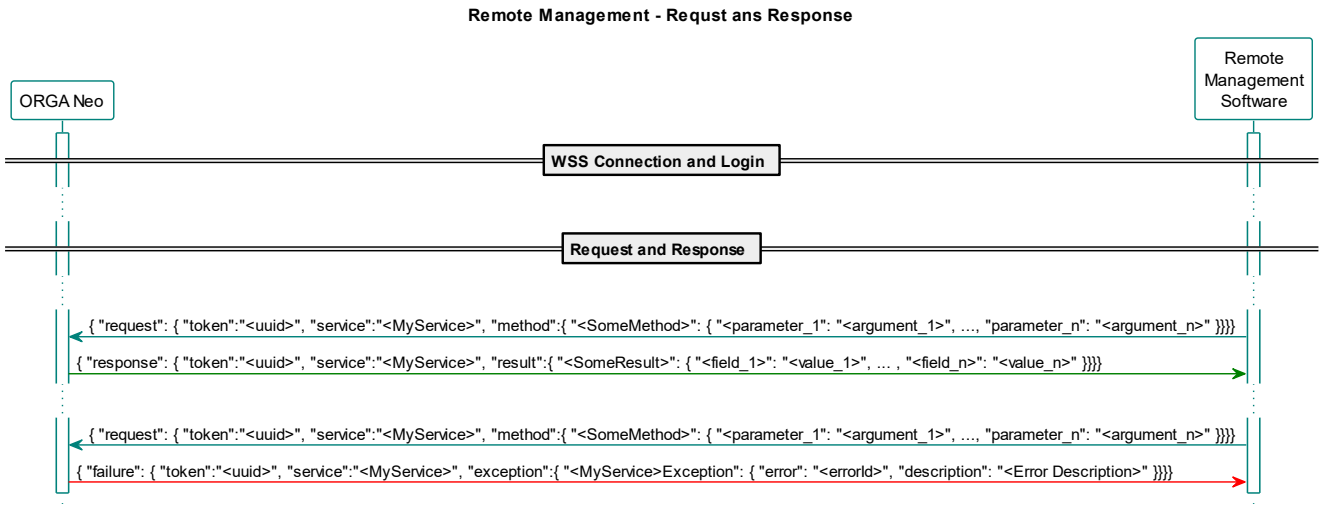
Ein weiterer notwendiger Schritt, um Informationen über die RMI-Schnittstelle zu erhalten, ist das Etablieren einer Session. Hierzu muss sich ein im System hinterlegter Benutzer mit seinen Zugangsdaten einloggen.
Je nach gewählter Benutzerrolle existieren unterschiedliche Berechtigungen. Eine so etablierte Session muss mittels Keepalive-Nachrichten aktiv gehalten werden. Findet während einer konfigurierbaren Zeit (siehe rmi_timeout im Kapitel 4.7 Remote Management Interface) kein Datenaustausch über diese Schnittstelle statt, so beendet das Terminal die etablierte RMI-Session automatisch.

2. Genereller Ablauf

In diesem Kapitel soll mit Hilfe von Sequenzen-Diagrammen der prinzipielle Ablauf der Kommunikation über die Remote Management Schnittstelle verdeutlicht werden. Zunächst abstrakt, um die Struktur der verwendeten JSON-Objekte (JavaScript Object Notation) vorzustellen und im weiteren Verlauf etwas konkreter anhand implementierter Services. Meldungen, die nicht dem JSON-Schema entsprechen, werden abgelehnt und die Verbindung wird geschlossen.

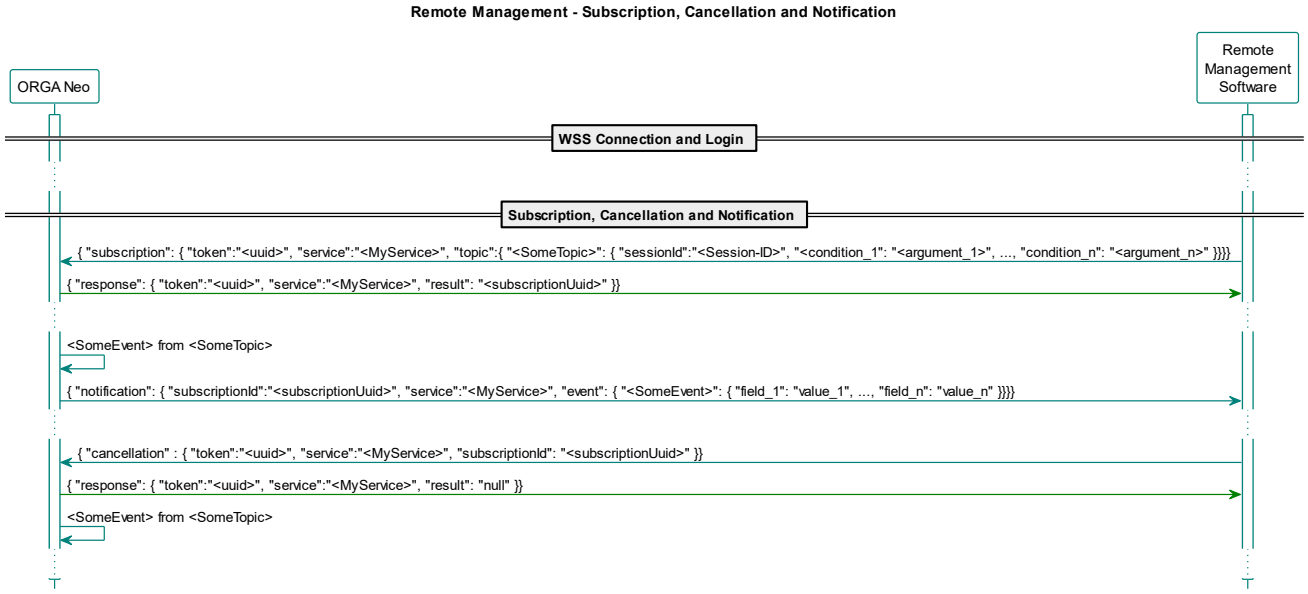
2.1 Request -> Response

Auch wenn der konkrete Ablauf des Verbindungsaufbaus, des Logins und der Keepalive in einem späteren Kapitel erfolgen, so ist dieser Ablauf essenziell und eine Voraussetzung für den generellen Ablauf und daher in jedem Sequenzen-Diagramm als Referenz enthalten. Das Remote Management Interface beruht auf dem klassischen Anfrage- und Antwort-Prozess, bei dem der Client eine Anfrage sendet, die von der Remote Management Software beantwortet wird.

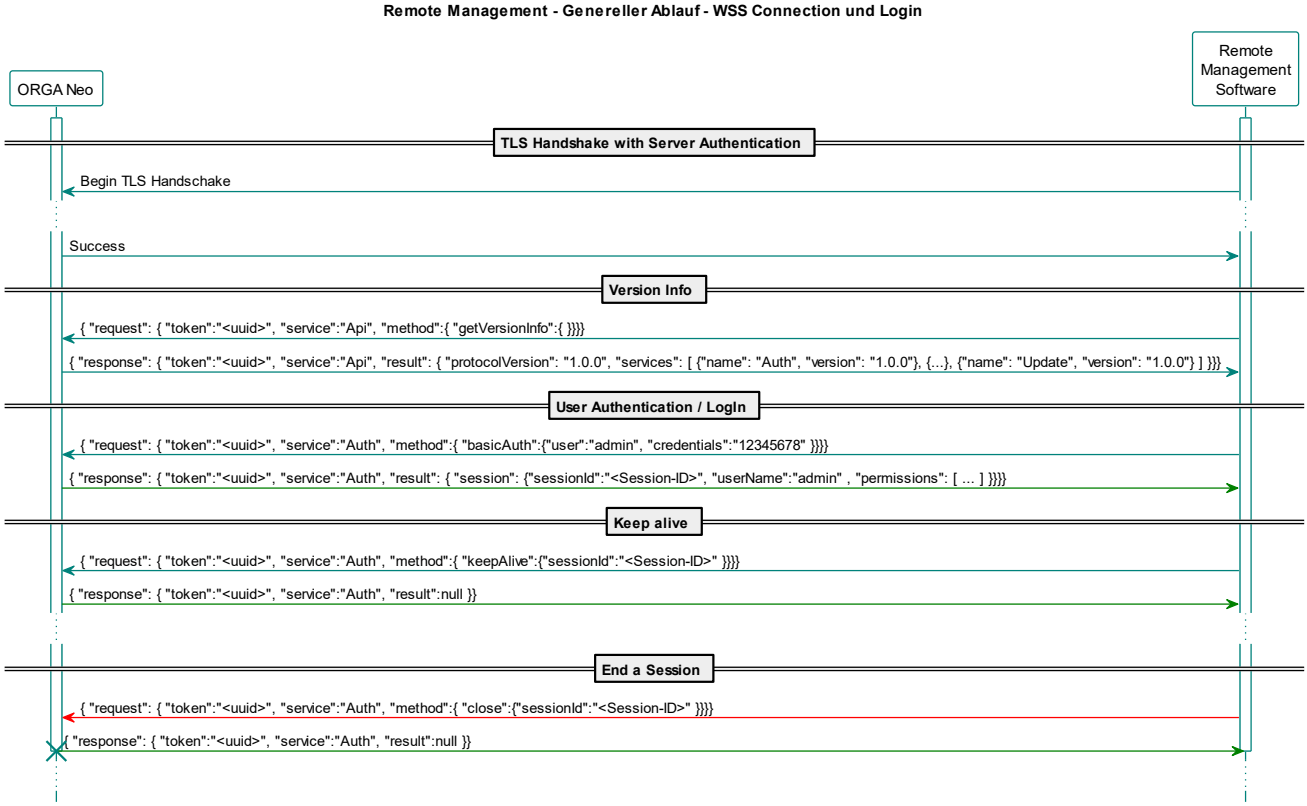


2.2 Subscription -> Notification, Cancellation

Während jeder Anmeldung wird eine Session mit einer universellen Kennung (UUID) eröffnet, die diese Session eindeutig identifiziert. Während dieser Session können so lange Events stattfinden, bis die Session storniert oder regulär beendet wird.



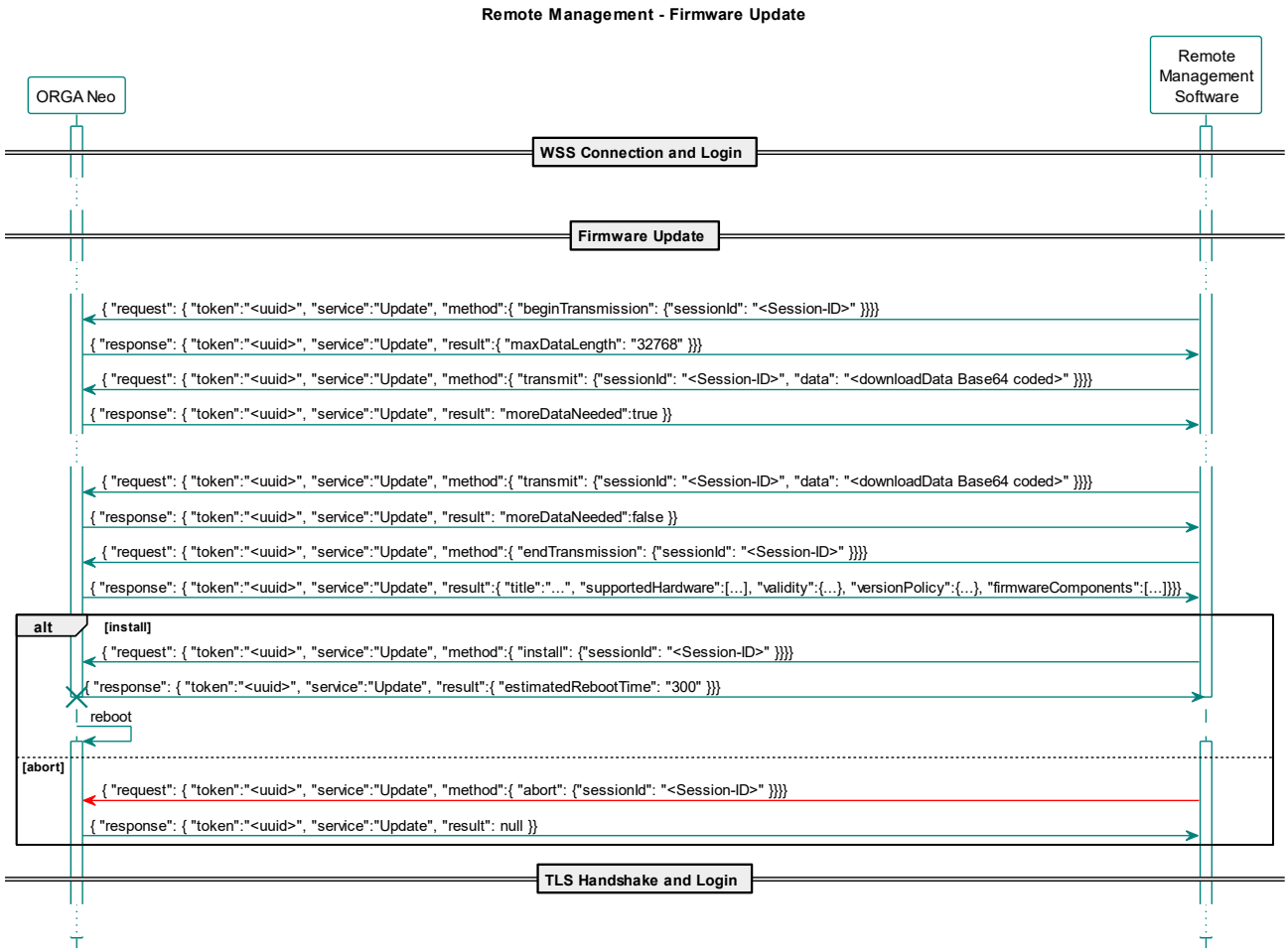
2.3 Verbindungsaufbau, Login und Keepalive



Jede Session wird mit einem TLS-Protokoll verschlüsselt. Im TLS-Handshake findet dafür zunächst ein sicherer Schlüsselaustausch sowie eine Authentifizierung zwischen dem Server und dem Client statt, gefolgt vom eigentlichen Login-Prozess (User Authentication). Mit den im Handshake ausgehandelten symmetrischen Schlüsseln wird eine sichere Datenübertragung gewährleistet, die mit spezifischen Keepalive-Paketen weiter abgesichert ist. Wenn der Client es unterlässt, Keepalive-Paket zu senden oder generell Anfragepakete ausbleiben, so wird in einem vorgegebenen Timeout die Session automatisch beendet.

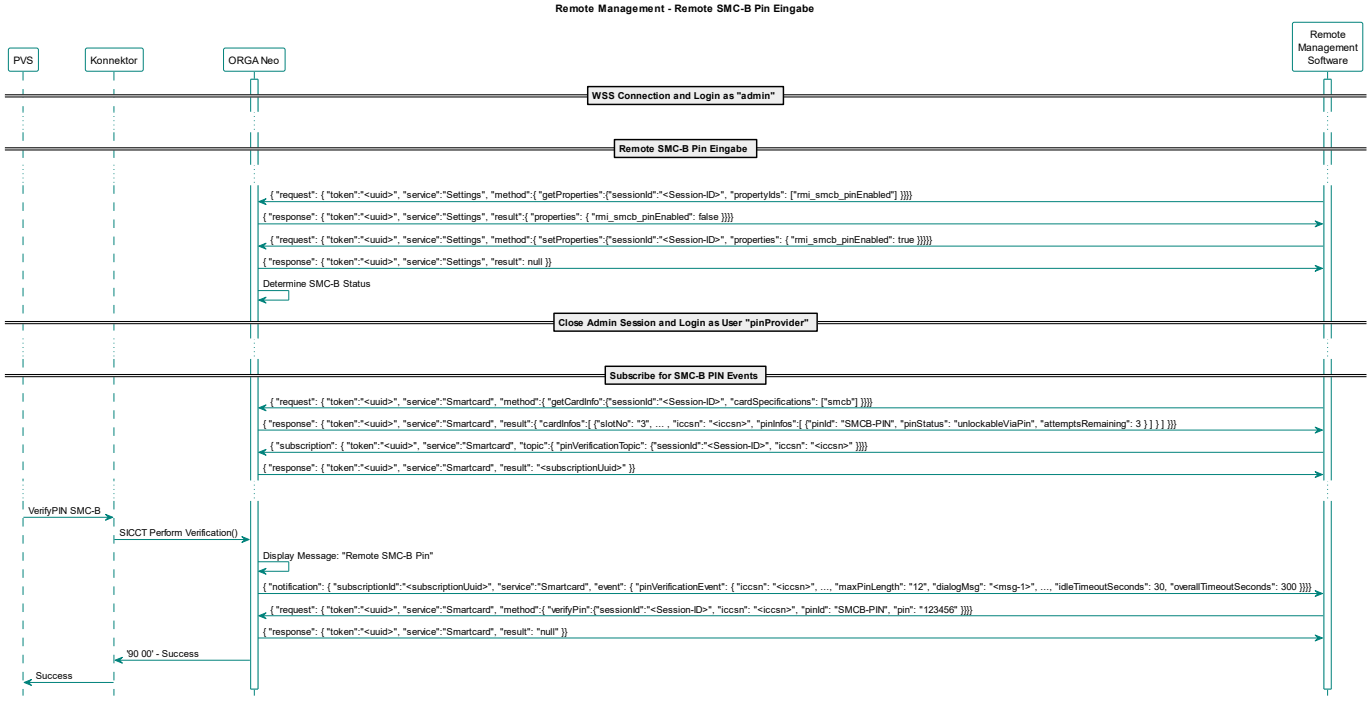
2.4 Firmware Update

Das Firmware Update über die Remote Management Schnittstelle erfolgt durch Übertragung Base64 codierten Paketen mit einer maximalen Größe von 32 KByte. Nach dem Ende der Übertragung erhält man eine Zusammenfassung, welche Firmwarebestandteile sich durch das Update verändern würden. An dieser Stelle kann man den Update-Prozess noch abbrechen oder durch das „install“ Kommando den Update Prozess abschließen. Das Terminal startet nach dem Installieren selbständig neu.



2.5 Remote SMC-B PIN

Das Setzen einer SMC-B PIN über die Remote Management Schnittstelle kann von zwei Benutzerrollen durchgeführt werden, zum einen über die Administrator-Rolle als auch über die PIN-Provider Rolle (pinProvider). Das Kartenterminal kann bis zu 3 SMC-B PINs verwalten, die auch alle über die Remote Management Schnittstelle neu gesetzt werden können. Der erste Teil des Sequenzdiagramms beschreibt die Statusermittlung einer SMC-B durch den Administrator, der zweite Teil des Sequenzdiagramms das Setzen einer neuen SMC-B PIN mit Hilfe der PIN-Provider Rolle.



2.6 Setzen und Vererben von Passwörtern

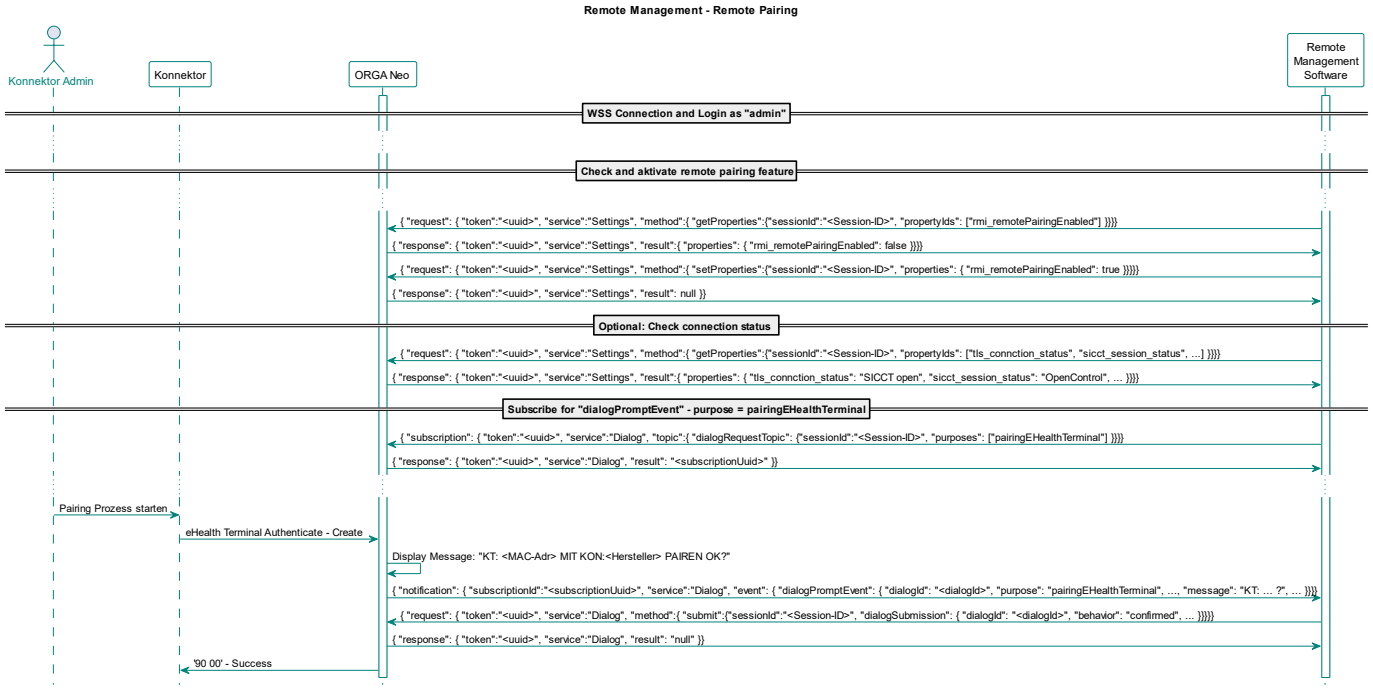
Gemäß dem Auth Service, der in Kapitel 3.3 beschrieben ist, sind für die RMI-Schnittstelle 3 zusätzliche Benutzerrollen eingeführt worden, von denen die folgenden 2 Rollen mit je einem Passwort geschützt sind:

a. Remote Admin (admin)
b. PinProvider

Beim ersten Update eines in der TI installierten Kartenterminals mit der Firmware 3.8.2 (oder niedriger) auf die Firmware 3.9.0 (oder höher) wird das Passwort des Geräte Admins **einmalig** auf den **Remote Admin** und den **PinProvider** übertragen/vererbt. Damit haben Sie die Möglichkeit, die Web-Applikation zu starten oder das Terminal remote direkt über die Schnittstelle anzusprechen.
Über die Schnittstelle selbst oder den Service-Bereich der Web-Applikation können diese Passwörter auch individuell geändert werden. Die Änderung des **Admin-Passwortes für das Gerätes** ist nur am Kartenterminal selbst möglich. Die Benutzerrolle **Anonymous** benötigt kein Passwort.

2.7 Remote Pairing

Die Bestätigung der Pairing-Anfrage des Konnektors kann über das Remote Management Interface erfolgen. Hierzu muss das Feature zunächst generell aktiviert werden und der RMI-Administrator muss sich für diesen Dialog Event registrieren. Sind diese Voraussetzungen erfüllt und der Konnektor Administrator löst über die Management Oberfläche des Konnektors den Pairing Prozess aus, so wird neben der lokalen Displaymeldung gleichzeitig eine Meldung an die Remote Management Software für das Terminal gesendet. Sowohl lokal, als auch Remote kann der Dialog nun bestätigt werden.



3. JSON-Objekte, Service Module und Methoden

3.1 Service übergreifende Beschreibung

Der Datenaustausch über die Remote Management Schnittstelle erfolgt über JSON-Objekte [RFC_4627]. Jede Anfrage an den WebSocket Server enthält dabei Felder und Methoden (die wiederum Parameter enthalten können) und jede Antwort des WebSocket Servers enthält ebenfalls Felder und Rückgabewerte der jeweils aufgerufenen Methode. Fehlerindikationen werden mittels eines Fehler Objektes und einer enthaltenen Exception angezeigt.

Folgende generischen JSON-Objekte werden derzeit unterstützt:

Objekt Type	Anfrage / Respone	Beschreibung
„request“	Anfrage	Generelle Anfrage des Websocket Client an den Server
„response“	Response	Antwort des Websocket Servers an den Client – Gut-Fall
„failure“	Response	Antwort des Websocket Servers an den Client – Fehler-Fall
"subscription"	Anfrage	Anfrage des Websocket Client an den Server. Registrieren auf ein bestimmtes Ereignis
"cancellation"	Anfrage	Anfrage des Websocket Client an den Server. De-Registrieren eines bestimmten Ereignisses.
"notification"	Response	Antwort des Websocket Servers an den Client – Ereignis ist eingetreten.

Jede Anfrage an den WebSocket Server, die ein gültiges und unterstütztes JSON-Objekt enthält, wird mit einem Response-Objekt beantwortet. Nicht unterstützte JSON-Objekte oder Anfragen in einem anderen Format werden kommentarlos verworfen.

Jedes JSON-Objekt enthält mindestens die nachfolgend beschriebenen Felder:

Feld	Format	Beschreibung
„token“	UUID	Eine vom Client zu generierende UUID [RFC_4122] die vom Server in die Responsedaten gespiegelt wird, um eine Zuordnung zwischen Anfrage und Response zu gewährleisten. Der Token muss für jede neue Anfrage einzigartig sein. Sollte der Server bei der Verarbeitung einer Anfrage mit identischem Token, zur Laufzeit der Bearbeitung einer vorherigen Anfrage, eine Kollision feststellen, so wird diese Anfrage mit einer Fehlermeldung abgewiesen.
„service“	String	Zeichenkette, die einen der unterstützten Services benennt (siehe Kapitel 3.1.2 Implementierte Services und ihre Methoden)

3.1.1 Generische Beschreibung der verwendeten JSON-Objekte

3.1.1.1 Request

```
{
	"request": {
		"token":"4194e6cc-76f4-441b-b21f-a3760e3c9040",
		"service":"MyService",
		"method": {
			"<SomeMethod>": {
				"sessionId":"01234567-89ab-cdef-0123-456789abcdef"
				"parameter1":"argument1",
				"parameter2":"argument2",
				"parameter3":"argument3",
				...
			}
		}
	}
}
```

3.1.1.2 Response

```
{
	"response": {
		"token":"4194e6cc-76f4-441b-b21f-a3760e3c9040",
		"service":"MyService",
		"result": {
			"<SomeResult>": {
				"field1":"value1",
				"field2":"value2",
				"field3":"value3",
				...
			}
		}
	}
}
```

3.1.1.3 Exception

```
{
	"failure": {
		"token":"4194e6cc-76f4-441b-b21f-a3760e3c9040",
		"service":"MyService",
		"exception": {
			"serviceException": {
				"error":"errorId",
				"description":"Error Description"
			}
		}
	}
}

{
	"failure": {
		"token": "4194e6cc-76f4-441b-b21f-a3760e3c9040",
		"service": "MyService",
		"exception": {
			"invalidPropertiesException": {
				"description": "Invalid Properties",
				"invalidProperties": [
					{
						"propertyId": "some_property",
						"invalidValue": "some_invalidValue",
						"issue": "Invalid value for property"
					},
					{
						"propertyId": "another_property",
						"invalidValue": "another_invalidValue”,
						“issue”: “Invalid value for property”
					},
					...
				]
			}
		}
	}
}
```

3.1.1.4 Subscription

```
{
	"subscription": {
		"token":"9ba91f5e-fc36-4722-8777-ed6ab45e99c7",
		"service":"MyService",
		"topic": {
			"<SomeTopic>": {
				"sessionId":"01234567-89ab-cdef-0123-456789abcdef"
				"condition1": "argument1",
				"condition2": "argument2",
				"condition3": "argument3",
				...
			}
		}
	}
}
```

3.1.1.5 Cancellation

```
{
	"cancellation": {
		"token":"99b68a34-100f-42e5-bb99-858b6c12a2b1",
		"service":"MyService",
		"subscriptionId": "e8db1139-dde1-4d2a-91dd-4c5726370b6f"
	}
}
```

3.1.1.6 Notification

```
{
	"notification": {
		"subscriptionId": "e8db1139-dde1-4d2a-91dd-4c5726370b6f",
		"service":"MyService",
		"event": {
			"<SomeEvent>": {
				"field1":"value1",
				"field2":"value2",
				"field3":"value3",
				...
			}
		}
	}
}
```

3.1.2 Implementierte Services und ihre Methoden

Service	Object-Type	Methoden / Topic	Beschreibung
„Api“			**Service für die Abfrage der Versionsinformationen des RMI-APIs und der implementierten Services**
	Request	„getVersionInfo“	Abfrage der Protokollversion und der Versionsinformationen der implementierten Services. (Neben der Methode „basicAuth“ des Services „Auth“ die einzige Methode, die auch ohne ein Login bzw. ohne eine Session-ID gesendet werden darf)
„Auth“			Service für Login, Logout und Keepalive Service für alle Benutzerrollen: admin, pinProvider, anonymous

	Request	„basicAuth“	Login, Aufbau einer RMI-Session
	Request	„keepAlive“	Keepalive Nachricht
	Request	„close“	Logout, beenden der RMI-Session
	Request	„basicAuthSetCredential“	Ändern von PINs bzw. Passwörtern für den jeweiligen Benutzer
„Settings“			**Setzen und Abfragen der Terminalkonfiguration, sowie Abfragen der Terminalselbstauskunft, bestimmter Statusinformationen, Statistikdaten (Daten, die über die Remote Management Schnittstelle nicht gesetzt werden können)** Service für alle Benutzerrollen: admin, pinProvider, anonymous, wobei das Setzen von Konfigurationsparametern dem admin vorbehalten ist
	Request	„getProperties“	Abfrage eines oder mehrerer Konfigurationsparameter
	Request	„setProperties“	Setzen eines oder mehrerer Konfigurationsparameter
„Pairing“			**Abfragen und Löschen der Pairing-Informationen** Service für die Benutzerrolle admin
	Request	„getInfo“	Abfragen der Pairing-Informationen
	Request	„deleteBlock“	Löschen eines oder mehrerer Pairing-Blöcke
	Request	„deletePublicKey“	Löschen eines oder mehrerer Public Keys aus einem Pairing-Block
„System“			**Ausführen von Systemfunktionen** Service für die Benutzerrolle admin
	Request	„reboot“	Neustart des Terminals
„Update“			**Firmware Update** Service für die Benutzerrolle admin
	Request	„beginTransmission“	Initiieren einer Update-Session
	Request	„transmit“	Übertragen der Update-Pakete
	Request	„endTransmission“	Beenden der Übertragung. Liefert Detailinformationen zum anstehenden Firmware Update.
	Request	„abort“	Abbrechen des Update-Prozesses
	Request	„install“	Installation des Updates und Terminal-Neustart
„Smartcard“			**Remote SMC-B PIN Feature** Service für die Benutzerrolle admin und pinProvider
	Request	„getCardInfo“	Statusinformationen der gesteckten SMC-B Karten ermitteln (Slot Nummer, Formfaktor, ICCSN, PIN-Status)
	Subscription	„pinVerificationTopic“	Registrierung als Interessent an SMC-B PIN-Ereignissen
	Cancellation	-	Kein Interesse mehr an SMC-B PIN-Ereignissen
	Notification	„pinVerificationEvent“	Eine SMC-B PIN-Eingabe wurde soeben vom Konnektor (bzw. PVS) angestoßen.
	Request	„verifyPin“	Übertragen der SMC-B PIN
	Request	„abortPinVerification“	Abbrechen einer laufenden remote SMC-B PIN-Eingabe
„Dialog“			**Remote Pairing Feature** Service für die Benutzerrolle admin
	Subscription	"dialogRequestTopic"	Registrierung als Interessent an Dialog-Ereignissen (Pairing-Anfragen)
	Cancellation		Kein Interesse mehr an Dialog-Ereignissen (Pairing-Anfragen)
	Notification	"dialogPromptEvent"	Ein Dialog (Pairing Anfrage) wurde soeben vom Konnektor angestoßen.
	Request	"submit"	Beantworten der Dialog Anfrage

3.1.3 Implementierte Datentypen und deren Wertebereiche

Folgende JSON-Basis-Datentypen werden für diese Remote Management Schnittstelle verwendet:

Daten-Type	Format / Beispiel	Beschreibung
Number	`{ "id": 1 }`	Eine JSON-Zahl kann nach [RFC_4627] das Gleitkommaformat unterstützen. Für die hier beschriebene Schnittstelle ist aber in der Regel ein positiver Integerwert im Bereich von 0 – 65535 gemeint (oder je nach Parameter noch weiter eingeschränkt -> siehe Spalte „Wertebereich“ im Anhang).
String	`{ "name": "RemoteSmcbPin" }`	Ein JSON-String kann nach [RFC_4627] eine Folge aus Unicode-Zeichen sein, die in Hochkommata **“…“** eingeschlossen werden, wobei ein Backslash (\) als Maskierungszeichen dient. Da viele der hier einstellbaren und abfragbaren Parameter auch direkt über die Management-Oberfläche (Terminal-Menü) einstellbar sind oder über die SICCT-Schnittstelle veränderbar sind, ist der erlaubte Zeichenvorrat meist weiter eingeschränkt. (Für die SICCT-Schnittstelle beispielsweise auf den Zeichensatz ISO 646-DE (DIN 66003))
Boolean	`{ "net_lan_dhcpEnabled" : false }`	Boolesche Werte haben den Wertebereich **true** und **false** und werden nicht in Anführungszeichen gesetzt.

Folgende JSON-Objecttypen werden für diese Remote Management Schnittstelle verwendet:

Opject-Type	Format / Beispiel	Beschreibung
Null	`"response": { …, "result":null }`	Wenn ein angefragter Parameter kein (oder noch keinen) Wert hat, kann der Wert null zurückgegeben werden. Beispiel wären bestimmte Statusinformationen, die bis dato noch nicht erfüllt sind. Oder aber als Antwort auf einen Methodenaufruf, der keinen Rückgabewert besitzt (void). Meist wird so der Erfolgsfall signalisiert.
Object	`"request": { "token": "…", "service": "…", "method": { "…": { } } }`	Der Objektdatentyp ist ein Satz aus Namens- oder Wertepaaren, die zwischen **{ }** (geschweiften Klammern) eingefügt werden. Die Schlüssel müssen Strings sein und sollten einzeln durch ein Komma getrennt werden.
Array	`"services": [ { "name": "Auth", "version": "1.0.0" }, { "name": "Settings", "version": "1.0.0" } ]`	Ein Array-Datentyp ist eine geordnete Sammlung von Werten, die zwischen [ ] (eckigen Klammern) eingefügt werden. In JSON müssen Array-Werte vom Typ **String, Number, Object, Array, Boolean** oder **Null** verwendet werden.

Weitere in diesem Dokument verwendeten Einschränkungen und Formatvorgaben:

Daten-Type	Wertebereiche	Beschreibung
String	“psk“|“mschap“|“pubkey“|“eap_tls“	Eine Enum-Datentype wird im JSON-Format als Basis-Datentype String repräsentiert. Dieser darf aber nur die in der Spalte Wertebereich und durch das griechische **Iota groß** Zeichen „ **|** “ getrennten Zeichenketten enthalten.
String	Version-String	Versionsinformation im Format “MMM.mmm.ppp“. Wobei MMM die Major-, mmm die Minor-Number und ppp den Patchlevel im Bereich von 0-255 darstellt. Beispiel: “1.0.0“
String	IP-v4-Addr	Gültige IPv4 Adresse nach [RFC_791]. Beispiel: “192.168.1.1“

Die spezifischen Wertebereiche jedes Datentyps befinden sich im Anhang in der Spalte „Wertebereich“.

3.1.4 Allgemeine Fehlermeldungen / Exceptions

Die nachfolgend beschriebenen Fehlermeldungen können Service übergreifend auftreten.

Exception Fehlertypen	Fehlermeldung	Beschreibung
authException	„invalidSessionId“	„Invalid Session ID“
rpcException	„timeout“	„Timeout“
rpcException	„duplicateToken“	„Duplicate Token“
rpcException	„invalidService“	„Invalid Service“
rpcException	„invalidMethod“	„Invalid Method“
rpcException	„invalidTopic“	„Invalid Topic“
rpcException	„invalidObject“	„Invalid Object“

3.2 Protokoll Version und Versionsinformationen der Services – Service API

3.2.1 Methoden, Parameter und Response Objekte

3.2.1.1 getVersionInfo - request

```
{
	"request": {
		"token":   "76ed90aa-7668-4740-94e2-f2ed6e538b89",
		"service": "Api",
		"method": {
			"getVersionInfo": {

			}
		}
	}
}
```

Methode	Parameter	Beschreibung
getVersionInfo		
	-	Leere Parameterliste

3.2.1.2 getVersionInfo - response


```
{
	"response": {
		"token": "76ed90aa-7668-4740-94e2-f2ed6e538b89",
		"service":"Api",
		"result": {
			"protocolVersion": "1.0.0",
			"serviceVersions": [
				{
					"name": "Auth",
					"version": "1.0.0"
				}, {
					"name": "Settings",
					"version": "1.0.0"
				}, {
					"name": "Pairing",
					"version": "1.0.0"
				}, {
					"name": "System",
					"version": "1.0.0"
				, {
					"name": "Update",
					"version": "1.0.0"
				}, {
					"name": "Smartcard",
					"version": "1.1.0"
				}, {
					"name": "Dialog",
					"version": "1.0.0"
				}
			]
		}
	}
}
```


Result Objekt	Parameter / Array Elemente	Beschreibung
„protocolVersion“		Versionsinformation des Remote Management Protokolls („MMM.mmm.ppp“)
„serviceVersions“		Array mit den Versionsinformationen pro implementierten Service
	„name“	Name des implementierten Service
	„version“	Versionsinformation des Services („MMM.mmm.ppp“)

3.2.1.3 Fehlermeldungen / Exceptions

Die nachfolgend beschriebenen Fehlermeldungen können spezifisch für den Service „API“ auftreten.

Exception Fehlertypen	Fehlermeldung	Beschreibung
authException	permissionDenied	„Permission denied“

3.3 Authentication und Keepalive – Service: Auth

Dieser Service realisiert die Funktionen zum Log-In bzw. den Aufbau einer RMI-Session, die damit verbundene Benutzer- und Rechteverwaltung, sowie Funktionen, um die Remote-Management Session am Leben zu erhalten und zu beenden. Folgende Benutzerrollen mit den dazugehörigen Rechten sind implementiert:

Benutzerrolle	Rechte	Beschreibung
anonymous	Die detaillierte Auflistung dieser Leseberechtigung befindet sich im Anhang in der Spalte „Anonymous“	Leseberechtigung für ausgewählte Systeminformationen.
pinProvider		Benutzer mit sehr eingeschränkten Rechten wie die Benutzerrolle anonymous, jedoch zusätzlich berechtigt für die Ausführung des Smartcard Services für den Anwendungsfall „Remote SMC-B PIN Eingabe“
admin		Lese- und Schreibberechtigung für alle über das RMI freigegebenen Konfigurationseinträge, Abfrage sämtlicher Systeminformationen, Firmware Update und Ausführen von Systemfunktionen wie Neustart.
sicctAdmin		Administrator der SICCT-Protokoll-Schnittstelle zum Konnektor. Über die RMI-Schnittstelle kann das Passwort (bzw. die PIN) dieser Benutzerrolle geändert werden - es ist jedoch nicht möglich sich mit diesem Benutzernamen an der RMI-Schnittstelle anzumelden.

Jeder Benutzer kann bis zu 3 RMI-Sessions öffnen. Der Versuch eine weitere Session zu öffnen, führt zu einer entsprechenden Fehlermeldung (Exception). Die Gesamtanzahl der möglichen offenen Sessions beträgt somit 9 (jeweils 3 Sessions für die Benutzer anonymous, pinProvider, und admin).

3.3.1 Methoden, Parameter und Response Objekte

3.3.1.1 basicAuth - request

Beispiel:

```
{
	"request": {
		"token":"11111111-1111-1111-1111-111111111111",
		"service":"Auth",
		"method": {
			"basicAuth": {
				"user":"admin",
				"credentials":"12345678"
			}
		}
	}
}
```

Methode	Parameter	Beschreibung
basicAuth		
	„user“	"anonymous" - für eine anonyme Session "pinProvider" - User-Berechtigung "admin" - Administratorberechtigung
	„credentials“	PIN oder Passwort des jeweiligen Benutzers. "" - für eine anonyme Session

3.3.1.2 basicAuth - response

Beispiel:

```
{
	"response": {
		"token":"11111111-1111-1111-1111-111111111111",
		"service":"Auth",
		"result": {
			"session": {
				"id":"01234567-89ab-cdef-0123-456789abcdef",
				"userName":"admin",
				"permissions": [
					"readSysInfo",
					"readStatusInfo",
					"readDefaultSettings",
					"writeDefaultSettings",
					"readClassifiedSettings",
					"writeClassifiedSettings",
					"sicctDialog"
				]
			}
		}
	}
}
```

Result Objekt	Parameter	Beschreibung
„session“		
	„id“	Session-ID
	„userName“	"anonymous" - für eine anonyme Session "pinProvider" - User-Berechtigung "admin" - Administratorberechtigung
	„permissions“	Lister der Berechtigungen für den jeweiligen Benutzer

3.3.1.3 keepAlive - request

Beispiel:

```
{
	"request": {
		"token":"22222222-2222-2222-2222-222222222222",
		"service":"Auth",
		"method": {
			"keepAlive": {
				"sessionId":"01234567-89ab-cdef-0123-456789abcdef"
			}
		}
	}
}
```

Methode	Parameter	Beschreibung
keepAlive		
	„sessionId“	Session-ID

3.3.1.4 keepAlive - response

Beispiel:

```
{
	"response": {
		"token":"22222222-2222-2222-2222-222222222222",
		"service":"Auth",
		"result":null
	}
}
```

Result Objekt	Parameter	Beschreibung
null	-	Positive Bestätigung des Empfangs der KeepAlive-Nachricht

3.3.1.5 close - request

Beispiel:

```
{
	"request": {
		"token":"33333333-3333-3333-3333-333333333333",
		"service":"Auth",
		"method": {
			"close": {
				"sessionId":"01234567-89ab-cdef-0123-456789abcdef"
			}
		}
	}
}
```

Methode	Parameter	Beschreibung
close		
	„sessionId“	Session-ID

3.3.1.6 close - response

Beispiel:

```
{
	"response": {
		"token":"33333333-3333-3333-3333-333333333333",
		"service":"Auth",
		"result":null
	}
}
```

Result Objekt	Parameter	Beschreibung
null	-	Positive Bestätigung. Session ist geschlossen.

3.3.1.7 basicAuthSetCredentials - request

Beispiel:

```
{
	"request": {
		"token":"33333333-3333-3333-3333-333333333333",
		"service":"Auth",
		"method": {
			"basicAuthSetCredentials": {
				"sessionId":"01234567-89ab-cdef-0123-456789abcdef"
				"user":"admin",
				"currentCredential":"12345678"
				"newCredential":"87654321"
			}
		}
	}
}
```

Methode	Parameter	Beschreibung
basicAuthSetCredentials		
	„sessionId“	Session-ID
	„user“	"pinProvider" - User-Berechtigung "admin" - Administratorberechtigung für das RMI „sicctAdmin“ - SICCT-Admin-Session Berechtigung
	„currentCredential“	PIN oder Passwort des jeweiligen Benutzers.
	„newCredential“	Neue PIN bzw. neues Passwort des jeweiligen Benutzers.

3.3.1.8 basicAuthSetCredentials - response

Beispiel:

```
{
	"response": {
		"token":"33333333-3333-3333-3333-333333333333",
		"service":"Auth",
		"result":null
	}
}
```

Result Objekt	Parameter	Beschreibung
null	-	Positive Bestätigung. PIN / Passwort Änderung ist erfolgreich.

3.3.2 Fehlermeldungen / Exceptions

Die nachfolgend beschriebenen Fehlermeldungen können spezifisch für den Service „Auth“ auftreten:

Exception Fehlertypen	Fehlermeldung	Beschreibung
authException	„authenticationFailed"	„Authentication failed"
authException	„guardTimeActive“	„Guard time active“
authException	„invalidSessionId“	„Invalid Session Id“
authException	„invalidUser“	„User invalid or unknown“
authException	„permissionDenied“	„Permission denied“
authException	„setCredentialFailed“	„Setting of credential failed“
authException	„maximumSessionsPerUserExceeded“	"Maximum sessions per user exceeded"

3.4 Verwalten der Pairing-Blöcke – Service „Pairing“

3.4.1 Methoden, Parameter und Response Objekte

3.4.1.1 getInfo - request

```
{
	"request": {
		"token": "acb6e306-101b-42f2-b1d5-ef90c0b8e313",
		"service": "Pairing",
		"method": {
			"getInfo": {
				"sessionId":"01234567-89ab-cdef-0123-456789abcdef"
			}
		}
	}
}
```

Methode	Parameter	Beschreibung
getInfo		
	„sessionId“	Session-ID

3.4.1.2 getInfo - response

Beispiel - 2 Pairing-Blöcke vorhanden. Block 1 enthält 2 Public Keys, Block 2 enthält 1 Public Key:

```
{
	"response": {
		"token": "acb6e306-101b-42f2-b1d5-ef90c0b8e313",
		"service":"Pairing",
		"result": {
			"blocks": [
				{
					"id": 1,
					"name": "pairingblock-1",
					"publicKeys": [
						{
							"id": 1,
							"name": "public-key-1: (RSA)",
							"type": "RSA",
							"key":
"0123456789ABCDEF...FEDCBA9876"
						}, {
							"id": 2,
							"name": "public-key-2: (ECC)",
							"type": "ECC",
```

```
                                        "key":
"FEDCBA9876543210...0123456789"
                                    }
                                ]
                            }, {
                                "id": 2,
                                "name": "pairingblock-2",
                                "publicKeys": [
                                    {
                                        "id": 1,
                                        "name": "public-key-1: (ECC)",
                                        "type": "ECC",
                                        "key":
"FFEEDDCCBBAA9988...7766554433"
                                    }
                                ]
                            }
                        ]
                }
        }
}
```


Beispiel - Keine Pairing-Blöcke vorhanden:


```
{
        "response": {
                "token":   "acb6e306-101b-42f2-b1d5-ef90c0b8e313",
                "service": "Pairing",
                "result": {
                        "blocks": []
                }
        }
}
```


Result Objekt	Parameter / Array Elemente	Array Elemente	Beschreibung
„blocks“			Array mit den Informationen zu den gespeicherten Pairing-Blöcken - kann 0-3 Elemente enthalten.
	„id“		Laufende Nummer des jeweiligen Pairing-Blocks
	„name“		Gespeicherter Name des Pairing-Blocks - meist „`pairingblock-x`“
	„publicKeys“		Array mit den Informationen zu den gespeicherten Public Keys in dem jeweiligen Pairing-Block - kann 0-3 Elemente enthalten.
		„id“	Laufende Nummer des jeweiligen Public Keys innerhalb des jeweiligen Pairing-Blocks
		„name“	Gespeicherter Name des Public Keys
		„type“	Type des Public Keys [„RSA“ oder „ECC“]
		„key“	Hexadezimale Repräsentation des Public Keys

3.4.1.3 deleteBlock - request

Beispiel - Alle Pairing-Blöcke löschen:

```
{
	"request": {
		"token": "de337864-1277-434a-94c0-209d03c5655b",
		"service": "Pairing",
		"method": {
			"deleteBlock": {
				"sessionId": "01234567-89ab-cdef-0123-456789abcdef",
				"blockIds":  [1, 2, 3]
			}
		}
	}
}
```

Methode	Parameter	Beschreibung
deleteBlock		
	„sessionId“	Session-ID
	„blockIds“	Array mit den zu löschenden Pairing-Block IDs - bis zu 3 IDs können übergeben werden

3.4.1.4 deleteBlock - response

Beispiel – Pairing-Blöcke erfolgreich gelöscht:

```
{
	"response": {
		"token":   "de337864-1277-434a-94c0-209d03c5655b",
		"service": "Pairing",
		"result":  null
	}
}
```

Result Objekt	Parameter	Beschreibung
null	-	Positive Bestätigung. Pairing-Block / Pairing-Blöcke wurden gelöscht.

3.4.1.5 deletePublicKey - request

Beispiel - 2 Public Keys aus Pairing-Block 1 löschen:

```
{
	"request": {
		"token": "a121fa90-f5fc-47d4-8651-9b85877a38d3",
		"service": "Pairing",
		"method": {
			"deletePublicKey": {
				"sessionId":    "01234567-89ab-cdef-0123-
456789abcdef",
				"blockId":      1,
				"publicKeyIds": [1,2]
			}
		}
	}
}
```

Methode	Parameter	Beschreibung
deletePublicKey		
	„sessionId“	Session-ID
	„blockId“	Pairing-Block ID, aus dem der / die Public Keys gelöscht werden soll.
	„publicKeyIds“	Array mit den zu löschenden Public Key IDs – bis zu 3 IDs können übergeben werden,

3.4.1.6 deletePublicKey - response

Beispiel - Public Keys erfolgreich gelöscht:

```
{
      "response": {
            "token":   "a121fa90-f5fc-47d4-8651-9b85877a38d3",
            "service": "Pairing",
            "result":  null
      }
}
```

Result Objekt	Parameter	Beschreibung
null	-	Positive Bestätigung. Public Key / Public Keys wurden aus dem Pairing-Block gelöscht.

3.4.2 Fehlermeldungen / Exceptions

Die nachfolgend beschriebenen Fehlermeldungen können spezifisch für den Service „Pairing“ auftreten.

Exception Fehlertypen	Fehlermeldung	Beschreibung
authException	permissionDenied	„Permission denied“
pairingException	pairingBlockIdsEmpty	"Pairing: 'blockIds' are empty"
pairingException	pairingBlockIdNotAvailable	"Pairing: one or more of 'blockIds' are not available"
pairingException	pairingBlockIdOutOfRange	"Pairing: blockId(s) are out of range"
pairingException	pairingDeleteBlockFailed	"Pairing: delete of pairing block failed"
pairingException	pairingPublicKeyIdsEmpty	"Pairing: 'publicKeyIds' are empty"
pairingException	pairingPublicKeyIdNotAvailable	"Pairing: one or more of 'publicKeyIds' are not available"
pairingException	pairingPublicKeyIdOutOfRange	"Pairing:pPublicKeyId(s) are out of range"
pairingException	pairingDeletePublicKeyFailed	"Pairing: delete of public key failed"
pairingException	pairingSavingOfDataFailed	"Pairing: saving of pairing-data failed"
pairingException	paringGetInfoFailed	“Paring: get info of pairing-data failed”

3.5 Ausführen von Systemfunktionen – Services „System“

Dieser Service ist für das Ausführen von Systemfunktionen implementiert. Derzeit kann nur ein Neustart des Terminals durchgeführt werden. Dies ist erforderlich, wenn insbesondere die Netzwerkkonfiguration geändert wurde oder die SICCT-Parameter angepasst wurden, um die übertragende Änderung wirksam werden zu lassen.

Dieser Service ist der Benutzerrolle „admin“ vorbehalten.

3.5.1 Methoden, Parameter und Response Objekte

3.5.1.1 reboot - request

```
{
	"request": {
		"token":   "4152ef88-2436-49a2-b4ca-22d3b7f951ee",
		"service": "System",
		"method": {
			"reboot": {
				"sessionId": "dd3284ac-8560-4f29-a1b3-1dfa7eda5fdd"
			}
		}
	}
}
```

Methode	Parameter	Beschreibung
reboot		
	„sessionId“	Session-ID

3.5.1.2 reboot - response

Beispiel – Terminal wird in Kürze neu gestartet:

```
{
	"response": {
		"token":   "4152ef88-2436-49a2-b4ca-22d3b7f951ee",
		"service": "System",
		"result":  null
	}
}
```

Result Objekt	Parameter	Beschreibung
null	-	Positive Bestätigung. Terminal wird in Kürze neu gestartet.

3.5.2 Fehlermeldungen / Exceptions

Die nachfolgend beschriebenen Fehlermeldungen können spezifisch für den Service „System“ auftreten.

Exception Fehlertypen	Fehlermeldung	Beschreibung
authException	permissionDenied	„Permission denied“

3.6 Verändern und Abfragen von Einstellungen - Service: Settings

Dieser Service ist für das Setzen und Abfragen der Terminalkonfiguration (Parameter, die sowohl gelesen als auch verändert werden können), sowie das Abfragen der Terminalselbstauskunft, bestimmter Statusinformationen und Statistikdaten (Parameter, die über die Remote Management Schnittstelle nicht verändert und nur gelesen werden können).

Alle Parameter werden mit den notwendigen Zusatzinformationen in Kapitel 4 Settings - Parameter tabellarisch beschrieben.

3.6.1 Methoden, Parameter und Response Objekte

3.6.1.1 getProperties - request

```
{
	"request": {
		"token": "4a13cfe1-2971-4508-ab06-b4e725004595",
		"service": "Settings",
		"method": {
			"getProperties": {
				"sessionId": "d6ba27f9-c051-4a28-884f-97c6d80c126e",
				"propertyIds": [
					"net_dns",
					"net_hostname",
					"net_lan_net_lan_dhcpEnabled",
					"net_lan_dhcpOptEnabled",
					"net_lan_ipAddr"
				]
			}
		}
	}
}
```

Methode	Parameter	Beschreibung
getProperties		
	„sessionId“	Session-ID
	„propertyIds“	Array von Properties, die abgefragt werden sollen (siehe Kapitel 4 für eine Auflistung aller möglichen Property IDs)

3.6.1.2 getProperties - response

```
{
	"response": {
		"token": "4a13cfe1-2971-4508-ab06-b4e725004595",
		"service": "Settings",
		"result": {
			"properties": {
				"net_dns"                : "0.0.0.0",
				"net_hostname"           : "ORGA6100-00300000010301",
				"net_lan_dhcpEnabled"    : false,
				"net_lan_dhcpOptEnabled" : false,
				"net_lan_ipAddr"         : "192.168.1.1"
			}
		}
	}
}
```

Result Objekt	Parameter	Beschreibung
properties	-	Liste der angefragten Properties als Key Value Paare (siehe Kapitel 4 für eine Auflistung aller möglichen Property IDs)

3.6.1.3 setProperties - request

```
{
	"request": {
		"token":"22222222-2222-2222-2222-222222222222",
		"service":"Settings",
		"method":{
			"setProperties": {
				"sessionId":"01234567-89ab-cdef-0123-456789abcdef",
				"properties": {
					"sicct_keepAliveIntervalSec"       : 10,
					"sicct_keepAliveTimeoutSec"        : 120,
					"sicct_blockReadTimeoutSec"        : 5,
					"sicct_messageReadTimeoutSec"      : 5,
					"sicct_maxErrors"                  : 5,
					"sicct_tls_acceptTimeoutSec"       : 20,
					"sicct_tls_version"                : 12,
					"sicct_tls_caList"                 : "pu",
					"sicct_announcementIntervalSec"    : 5,
					"sicct_adminSessionEnabled"        : true,
					"sicct_cmd_setStatusEnabled"       : true,
					"sicct_cmd_downloadEnabled"        : true,
				}
			}
		}
	}
}
```

Methode	Parameter	Beschreibung
setProperties		
	„sessionId“	Session-ID
	„properties“	Liste der zu setzenden Properties als Key : Value Paare (siehe Kapitel 4 für eine Auflistung aller möglichen Property IDs)

3.6.1.4 setProperties - response

```
{
	"response": {
		"token":"22222222-2222-2222-2222-222222222222",
		"service":"Settings",
		"result":null
	}
}
```

Result Objekt	Parameter	Beschreibung
null	-	Positive Bestätigung. Änderungen wurden erfolgreich übernommen. **Achtung:** Bestimmte Properties erfordern einen Neustart des Terminals!

3.6.2 Fehlermeldungen / Exceptions

Die nachfolgend beschriebenen Fehlermeldungen können spezifisch für den Service „Settings“ auftreten.

Exception Fehlertypen	Fehlermeldung	Beschreibung
authException	permissionDenied	„Permission denied“
invalidPropertiesException	„invalidProperties“:[…]	„Invalid Properties“ / „Invalid value for property

3.7 Firmware Update – Service: Update

3.7.1 Methoden, Parameter und Response Objekte

3.7.1.1 beginTransmission - request

```
{
	"request": {
		"token":   "90743dcf-6a0b-42ef-a647-fba569201119",
		"service": "Update",
		"method": {
			"beginTransmission": {
				"sessionId": "01234567-89ab-cdef-0123-456789abcdef"
			}
		}
	}
}
```

Methode	Parameter	Beschreibung
beginTransmission		
	„sessionId“	Session-ID

3.7.1.2 beginTransmission - response

```
{
	"response": {
		"token":   "90743dcf-6a0b-42ef-a647-fba569201119",
		"service": "Update",
		"result": {
			"maxDataLength": 32768
		}
	}
}
```

Result Objekt	Parameter	Beschreibung
„maxDataLength“	-	Maximale Länge der Binärdaten, die in einer transmit-Message übertragen werden können. **Hinweis:** Die Base 64codierten Daten können durch die Codierung entsprechend länger sein.

3.7.1.3 transmit - request

```
{
	"request": {
		"token":   "76400c03-e981-4beb-a1a8-144475638b03",
		"service": "Update",
		"method": {
			"transmit": {
				"sessionId": "01234567-89ab-cdef-0123-456789abcdef",
				"data":      "<downloadData Base64 coded>"
			}
		}
	}
}
```

Methode	Parameter	Beschreibung
transmit		
	„sessionId“	Session-ID
	„data“	Die Base 64codierten Firmware Image Daten.

3.7.1.4 transmit - response

```
{
	"response": {
		"token":   "76400c03-e981-4beb-a1a8-144475638b03",
		"service": "Update",
		"result": {
			"moreDataNeeded": true
		}
	}
}
```

Result Objekt	Parameter	Beschreibung
„moreDataNeeded“	-	Indikation, ob das vollständige Firmware Image bereits empfangen wurde: true – Firmware Image noch nicht vollständig false - Firmware Image vollständig empfangen

3.7.1.5 endTransmission - request

```
{
	"request": {
		"token":   "0e14fabc-7eca-42a4-b00f-b1840600f553",
		"service": "Update",
		"method": {
			"endTransmission": {
				"sessionId": "01234567-89ab-cdef-0123-456789abcdef"
			}
		}
	}
}
```

Methode	Parameter	Beschreibung
endTransmission		
	„sessionId“	Session-ID

3.7.1.6 endTransmission - response


```
{
      "response": {
            "token":   "0e14fabc-7eca-42a4-b00f-b1840600f553",
            "service": "Update",
            "result": {
                  "title":"Orga6141 Operating System 3.9.0",
                  "supportedHardware":[
                        "ORGA 6141 Online"
                  ],
                  "validity":{
                        "size"         :true,
                        "format"       :true,
                        "checksum"     :true,
                        "signature"    :true,
                        "compatibility":true,
                        "effectivity"  :true,
                        "version"      :true
                  },
                  "versionPolicy":{
                        "currentVersionWhitelistId":"19",
                        "currentVersionWhitelist"  :["3.8.2"],
                        "updateVersionWhitelistId" :"21",
                        "updateVersionWhitelist"   :["3.9.0"],
                        "willInstall":true
                  },
                  "firmwareComponents":[
                        {
                              "name":"Loader",
                              "currentVersion:":"8.2.0",
                              "updateVersion":"8.2.0",
                              "willInstall":false
                        },{
                              "name":"Kernel",
                              "currentVersion:":"5.4.191",
                              "updateVersion":"5.4.236",
                              "willInstall":true
                        },{
                              "name":"RootFS",
                              "currentVersion:":"3.8.2",
                              "updateVersion":"3.9.0",
                              "willInstall":true
                        },{
                              "name":"CI",
                              "currentVersion:":"1.0.0",
                              "updateVersion":"1.0.0",
                              "willInstall":true
                        },{
                              "name":"TSL-PU",
                              "currentVersion:":"1.5.0",
                              "updateVersion":"1.5.0",
                              "willInstall":true
                        },{
                              "name":"TSL-RU",
                              "currentVersion:":"1.12.0",
                              "updateVersion":"1.12.0",
                              "willInstall":true
```

```
                        },{
                              "name":"TSL-LU",
                              "currentVersion:":"1.20.0",
                              "updateVersion":"1.20.0",
                              "willInstall":true
                        }
                  ]

            }
      }
}
```

Result Objekt	Parameter	Beschreibung
„titel“		String, der das Firmware Update Image beschreibt
„suportedHardware“		Array von Strings, die die unterstützten Terminalvarianten beschreiben.
„validity“		Liste von Prüfungen, die das Firmware Image erfolgreich bestanden hat:
	„size“	Größen Überprüfung
	„format“	Formatprüfung
	„checksum“	Checksummenprüfung
	„signature“	Signaturprüfung
	„compatibility“	Kompatibilitätsüberprüfung
	„effectivity“	Mindestens eine Firmware Komponente wird installiert.
	„version“	Versionsprüfung
„versionPolicy“		Informationen zum Update der Firmwaregruppen Liste
	„currentVersionWhitelistId“	Nummer (ID) der derzeit im Terminal vorhandenen Firmwaregruppen Liste
	„currentVersionWhitelist“	Liste der Firmware Versionen, für die ein Update zulässig ist, die derzeit im Terminal gespeichert ist
	„updateVersionWhitelistId“	Im Firmware Update Image enthaltene Nummer (ID) der Firmwaregruppen Liste
	„updateVersionWhitelist“	Liste der Firmware Versionen, für die ein Update zulässig ist, die im Firmware Update Image enthalten ist
	„willInstall“	Indikation, ob die Firmwaregruppen Liste aktualisiert wird: true – neue Firmwaregruppen Liste wird installiert false – es bleibt bei der im Terminal vorhandenen Firmwaregruppen Liste
„firmwareComponents“		Array von Informationen zu dem im Update Image und im Terminal vorhandenen Firmware Komponenten, deren Version und welche Komponenten aktualisiert werden würden.
	„name“	Name der Firmware Komponente
	„currentVersion“	Im Terminal vorhandene Version der Komponente
	„updateVersion“	Im Update Image enthaltene Version der Firmware Komponente
	„willInstall“	Wird diese Komponente aktualisiert? true – die Firmware Komponente würde aus dem Update Image entnommen und installiert. false – die bereits im Terminal vorhandene Komponente wird nicht aktualisiert

3.7.1.7 abort - request

```
{
	"request": {
		"token":   "f55b5e05-a030-4551-4f20-9daacd6085b7",
		"service": "Update",
		"method": {
			"abort": {
				"sessionId": "01234567-89ab-cdef-0123-456789abcdef"
			}
		}
	}
}
```

Methode	Parameter	Beschreibung
abort		
	„sessionId“	Session-ID

3.7.1.8 abort - response

```
{
	"response": {
		"token":   "f55b5e05-a030-4551-4f20-9daacd6085b7",
		"service": "Update",
		"result": null
	}
}
```

Result Objekt	Parameter	Beschreibung
null	-	Positive Bestätigung. Update erfolgreich abgebrochen.

3.7.1.9 install - request

```
{
	"request": {
		"token":   "9db7f55b-5145-204f-30a0-5e066085daac",
		"service": "Update",
		"method": {
			"install": {
				"sessionId": "01234567-89ab-cdef-0123-456789abcdef "
			}
		}
	}
}
```

Methode	Parameter	Beschreibung
install		
	„sessionId“	Session-ID

3.7.1.10 install - response

```
{
      "response": {
            "token":   "9db7f55b-5145-204f-30a0-5e066085daac",
            "service": "Update",
            "result": {
                  "estimatedRebootSeconds": 300
            }
      }
}
```

Result Objekt	Parameter	Beschreibung
„aRebootSeconds“	-	Positive Bestätigung. Update wird installiert, Terminal startet im Anschluss neu und benötigt ca. die angegebene Zeit in Sekunden für diesen Vorgang.

3.7.2 Fehlermeldungen / Exceptions

Die nachfolgend beschriebenen Fehlermeldungen können spezifisch für den Service „Update“ auftreten.

Exception Fehlertypen	Fehlermeldung	Beschreibung
authException	permissionDenied	„Permission denied“
updateException	updateInitFailed	„UpdateInit failed“
updateException	updateAppendFailed	„UpdateAppend failed“
updateException	updateAppendDataToLong	„UpdateAppend data to long“
updateException	updateFinishFailed	„UpdateFinish failed“

3.8 Remote SMC-B PIN Eingabe – Service: Smartcard

Mit Hilfe des Remote SMC-B PIN Features ist es möglich eine eigentlich lokale PIN-Eingabe über das Remote Management Interface durchzuführen. Dieses Feature ist aufgrund aktueller gematik Anforderungen auf SMC-B Karten begrenzt. Es ist nicht möglich für eine eGK oder einen eHBA die PIN-Eingabe über das RMI durchzuführen.

Voraussetzung für die Remote SMC-B PIN-Eingabe ist das generelle Aktivieren der Funktion, welche im Auslieferungszustand deaktiviert ist. Dies kann der RMI-Administrator über den Service "Settings" mit Hilfe des Properties "rmi_smcb_pinEnabled" erreichen (siehe auch Kapitel 2.5 Remote SMC-B PIN) oder über die Remote Management Web-Applikation (siehe Kapitel 5 Web-Applikation) auf der Registerkarte „Service“ im Abschnitt „PRAXISKARTE /SMC-B“ – Remote SMC-B PIN = Ein.

Terminalintern werden nach Aktivierung des Features alle gesteckten Karten auf das Vorhandensein von einer oder mehrerer SMC-B Karten untersucht und deren PIN-Status ermittelt. Dieser Status kann dann mit Hilfe des "Smartcard" Services und der Methode "getCardInfo" abgefragt werden. Neben dem PIN-Status wird auch die Seriennummer (ICCSN) der gesteckten SMC-B Karten zurückgeliefert. Diese wird benötigt, um sich als Interessent der PIN-Verification einer bestimmten SMC-B zu registrieren ("subscription"). Erst nach dieser Registrierung ("subscription") werden die vom PVS bzw. vom

Konnektor getriggerten PIN-Eingaben an das Remote Management Interface delegiert, in dem ein "pinVerificationEvent" in einer "notification" Message an den registrierten Interessenten gesendet wird.

Dieser hat nun die Möglichkeit (innerhalb der bereitgestellten Timeout-Zeiten), die entsprechende SMC-B PIN in einer "request" Message an das Terminal zu senden, um die SMC-B freizuschalten.

Der Erfolg oder der Misserfolg dieser Aktion wird über die entsprechende Antwortnachricht ("response" oder "failure") mitgeteilt.

Folgende Randbedingungen sind zu beachten:

- Es können zwar mehrere RMI-Sessions existieren aber nur eine Subscription pro SMC-B
- Eine Subscription persistiert nicht. Folgende Aktionen führen zum Beenden der Subscription:
  - Senden der entsprechenden "cancellation" Message
  - Beenden der RMI-Sessions
  - Neustart des Terminals
- Die Benutzerrolle pinProvider wurde eingeführt, um die Remote SMC-B PIN-Eingabe auch ohne eine offene Admin-Session nutzen zu können. Das generelle Aktivieren dieses Features ist jedoch der Administrator-Rolle vorbehalten.

3.8.1 Methoden, Parameter und Response Objekte

3.8.1.1 getCardInfo - request

```
{
      "request": {
            "token": "84a5059a-1c3b-4cb3-8d2c-c036c48ccd26",
            "service": "Smartcard",
            "method": {
                  "getCardInfo": {
                        "sessionId":"01234567-89ab-cdef-0123-456789abcdef",
                        "cardSpecifications": ["smcb"]
                  }
            }
      }
}
```

Methode	Parameter	Beschreibung
getCardInfo		
	„sessionId“	Session-ID
	„cardSpecification“	Der Parameter `cardSpecifications` muss immer `["smcb"]` sein.

3.8.1.2 getCardInfo - response


```
{
	"response": {
		"token": "84a5059a-1c3b-4cb3-8d2c-c036c48ccd26",
		"service": "Smardcard",
		"result": {
			"cardInfos": [
				{
					"slotNo":    2,
					"formfactor": "fullSized",
					"cardSpecifications": ["smcb"],
					"iccsn":     "80276883580000008788",
					"pinInfos": [
						{
							"channelId": 0
							"pinId": "SMCB-PIN",
							"pinStatus": "verifyable",
							"attemptsRemaining": 3
						},
						{
							"channelId": 1
							"pinId": "SMCB-PIN",
							"pinStatus": "verified"
						}
					]
				},
				{
					"slotNo":    3,
					"formfactor": "simSized",
					"cardSpecifications": ["smcb"]
					"iccsn":     "80276883580000001669",
					"pinInfos": [
						{
							"channelId": 0
							"pinId": "SMCB-PIN",
							"pinStatus": "verified",
						}
					]
				}
			]
		}
	}
}
```

Result-Objekt	Parameter / Array Elemente	Array Elemente	Beschreibung
„cardInfos“			Array mit 0-3 Elementen, das Informationen liefert in welchem Slot SMC-B Karten stecken und welchen PIN-Status sie haben.
	„slotNo“		Nummer des Kartenslots in dem die SMC-B steckt
	„formfactor“		String der den Formfaktor der Karte beschreibt: „fullSized“ oder „simSized“
	„cardSpecification“		Der Parameter `cardSpecifications` ist derzeit immer `["smcb"]`.
	„iccsn“		Seriennummer (ICCSN) der SMC-B
	„pinInfos“		Array von PIN-Info Objekten. Enthält für die SMC-B nur ein Element.
		„pinId“	String der das PIN-Objekt beschreibt. Derzeit immer „SMCB-PIN“.
		„channelId“	Der logische Kanal der Karte, auf dem der PIN-Status ermittelt wurde.
		„pinStatus“	Der Parameter `pinStatus` kann folgende Werte annehmen: • "disabled" PIN deaktiviert, PIN-Verifikation nicht erforderlich • "emptyPIN" PIN muss noch gesetzt werden (Leer-PIN) • "transportPIN" PIN muss noch mit Hilfe der Transport-PIN gesetzt werden • "blocked" Karte gesperrt. PIN kann nicht zurückgesetzt werden • "verifyable" PIN-Eingabe erforderlich. • "verified" PIN wurde erfolgreich eingegeben. Karte ist freigeschaltet. • "unknown" PIM-Status kann nicht ermittelt werden.
		„attemptsRemaining“	Der Parameter ist optional und nur bei den nachfolgenden PIN-Status Informationen mit enthalten: • "verifyable" • "blocked"

3.8.1.3 pinVerificationTopic - subscription

```
{
      "subscription": {
            "token": "d2f24d46-f86b-40de-903f-132befad72c4",
            "service": "Smartcard",
            "topic": {
                  "pinVerificationTopic": {
                        "sessionId": "01234567-89ab-cdef-0123-456789abcdef",
                        "iccsn":     "80276883580000008788"
                  }
            }
      }
}
```

Subscription-Topic	Parameter	Beschreibung
pinVerificationTopic		
	„sessionId“	Session-ID
	„iccsn“	Seriennummer (ICCSN) der SMC-B für die die PIN-Eingabe am Remote Management Interface erfolgen soll.

3.8.1.4 pinVerificationTopic - subscription - response

```
{
	"response": {
		"token":   "d2f24d46-f86b-40de-903f-132befad72c4",
		"service": "Smartcard",
		"result":  "e25e86c2-81be-4a07-ac77-db39dc764b2f"
	}
}
```

Result Objekt	Parameter	Beschreibung
„String“	-	Die `subscriptionId`, die für das Beenden der Subscription (cancellation) benötigt wird und in der gesendeten Notification enthalten ist.

Im Fehlerfall, dass eine Subscription nicht möglich ist, weil Remote SMC-B PIN deaktiviert ist:

```
{
	"failure": {
		"token":"d2f24d46-f86b-40de-903f-132befad72c4",
		"service":"Smartcard",
		"exception": {
			"subscriptionException": {
				"description":	"Remote SMC-B PIN not enabled."
			}
		}
	}
}
```

3.8.1.5 pinVerificationTopic - subscription - cancellation

```
{
	"cancellation": {
		"token": "4cdf29da-5b45-478f-af4f-421106fcad45",
		"service": "Smartcard",
		"subscriptionId": "e25e86c2-81be-4a07-ac77-db39dc764b2f"
	}
}
```

Cancellation	Parameter	Beschreibung
	„subscriptionId“	Parameter, der als Antwort der Subscription bezogen wurde.

3.8.1.6 pinVerificationTopic – cancellation - response

```
{
      "response": {
            "token":   "4cdf29da-5b45-478f-af4f-421106fcad45",
            "service": "Smartcard",
            "result":   null
      }
}
```

Result Objekt	Parameter	Beschreibung
null	-	Positive Bestätigung. Subscription erfolgreich beendet..

3.8.1.7 pinVerificationEvent - notification

```
{
      "notification": {
            "subscriptionId": "e25e86c2-81be-4a07-ac77-db39dc764b2f",
            "service": "Smartcard",
            "event": {
                  "pinVerificationEvent": {
                        "iccsn":         "80276883580000008788",
                        "pinId": "SMCB-PIN",
                        "minPinLength": 5,
                        "maxPinLength": 12,
                        "dialogMsg":     "Bitte Geheimzahl eingeben",
                        "pinPrompt":     "INPUT: ",
                        "successMsg":    "Aktion erfolgreich",
                        "abortMsg":      "Abbruch",
                        "idleTimeoutSeconds": 30,
                        "overallTimeoutSeconds": 300
                  }
            }
      }
}
```

Notification-Event	Parameter	Beschreibung
	„subscriptionId“	Parameter, der als Antwort der Subscription bezogen wurde.
pinVerificationEvent		Liste an Informationen, die für den PIN-Eingabedialog verwendet werden können. Kommen zum großen Teil vom Konnektor.
	„iccsn“	Seriennummer (ICCSN) der SMC-B für die die PIN-Eingabe am Remote Management Interface erfolgen soll.
	„pinId“	String der das PIN-Objekt beschreibt. Derzeit immer „SMCB-PIN“.
	„minPinLength“	Minimale Länge der PIN
	„maxPinLength“	Maximale Länge der PIN
	„dialogMsg“	Dialog Message für die PIN-Eingabe
	„pinPrompt“	PIN-Prompt. Maximal 10 Zeichen. Kann vor der Eingabeaufforderung angezeigt werden. (<PIN-Prompt> : *********)
	„successMsg“	Dialog Message, die im Erfolgsfall angezeigt werden kann.

	„abortMsg“	Dialog Message die bei Abbruch der PIN-Eingabe angezeigt werden kann.
	„idleTimeoutSeconds“	Zeit bis zur Eingabe des 1. Zeichens und zwischen der Eingabe der Zeichen in Sekunden.
	„overallTimeoutSecomds“	Gesamtzeit, die die PIN-Eingabe höchstens dauern darf, bevor ein Abbruch (Timeout) erfolgt – ebenfalls in Sekunden.

3.8.1.8 verifyPin - request

```
{
	"request": {
		"token": "63100ab9-fe5f-4607-8ce1-f0ddfb536bf9",
		"service": "Smartcard",
		"method": {
			"verifyPin": {
				"sessionId": "01234567-89ab-cdef-0123-456789abcdef",
				"iccsn":     "80276883580000008788",
				"pinId":     "SMCB-PIN",
				"pin":       "123456"
			}
		}
	}
}
```

Methode	Parameter	Beschreibung
verifyPin		
	„sessionId“	Session-ID
	„iccsn“	Seriennummer (ICCSN) der SMC-B an die die PIN-Eingabe übergeben werden soll.
	„pinId“	String der das PIN-Objekt beschreibt. Derzeit immer „SMCB-PIN“.
	„pin“	PIN der SMC-B

3.8.1.9 verifyPin - response

Beispiel - PIN ist korrekt - PIN-Eingabe erfolgreich:

```
{
	"response": {
		"token": "63100ab9-fe5f-4607-8ce1-f0ddfb536bf9",
		"service": "Smartcard",
		"result": null
	}
}
```

Result Objekt	Parameter	Beschreibung
null	-	Positive Bestätigung. PIN-Eingabe war erfolgreich.

Beispiel - PIN ist falsch - PIN-Eingabe nicht erfolgreich:

```
{
	"failure": {
		"token":"63100ab9-fe5f-4607-8ce1-f0ddfb536bf9",
		"service":"Smartcard",
		"exception": {
			"pinException": {
				"error":			"pinRejected",
				"attemptsRemaining": 2
				"description":		"Invalid PIN",
			}
		}
	}
}
```

3.8.1.10 abortPinVerification - request

```
{
	"request": {
		"token": "48206424-dc8b-4413-b8b6-279a77e9e07a",
		"service": "Smartcard",
		"method": {
			"abortPinVerification": {
				"sessionId": "01234567-89ab-cdef-0123-456789abcdef",
				"iccsn":	"80276883580000008788",
				"pinId":	"SMCB-PIN"
			}
		}
	}
}
```

Methode	Parameter	Beschreibung
abortPinVerification		
	„sessionId“	Session-ID
	„iccsn“	Seriennummer (ICCSN) der SMC-B für die die PIN-Eingabe abgebrochen werden soll.
	„pinId“	String der das PIN-Objekt beschreibt. Derzeit immer „SMCB-PIN“.

3.8.1.11 abortPinVerification - response

Beispiel - Pin-Eingabe erfolgreich abgebrochen:

```
{
	"response": {
		"token": "48206424-dc8b-4413-b8b6-279a77e9e07a",
		"service": "Smartcard",
		"result": null
	}
}
```

Result Objekt	Parameter	Beschreibung
null	-	Positive Bestätigung. PIN-Eingabe wurde abgebrochen.

Beispiel - PIN-Eingabe konnte nicht abgebrochen werden (wurde ggf. bereits abgebrochen z.B. durch ein Timeout):

```
{
	"failure": {
		"token":"48206424-dc8b-4413-b8b6-279a77e9e07a",
		"service":"Smartcard",
		"exception": {
			"unexpectedRequestException": {
				"description": "Unexpected request"
			}
		}
	}
}
```

3.8.2 Fehlermeldungen / Exceptions

Die nachfolgend beschriebenen Fehlermeldungen können spezifisch für den Service „Smartcard“ auftreten:

Exception Fehlertypen	Fehlermeldung	Beschreibung
authException	permissionDenied	"Permission denied"
unexpectedRequestException		„Unexpected request“
unexpectedRequestException		„Invalid slot“
unexpectedRequestException		„No verification running“
unexpectedRequestException		„No subscription found“
pinException	pinRejected	
	pinBlocked	
	pinNotUsable	
	pinNotFound	
	pinInvalidLength	
	pinInvalidCharacter	
	pinSecurityStatus	
	unknown	
cardCommunicationException	-	"Card I/O Error"
cardCommunicationException	-	„No TLS with Konnektor“
cardCommunicationException	-	„No card inserted“
cardCommunicationException	-	„Invalid card inserted“
invalidArgumentException	-	"Invalid Argument"
invalidIccsnException	-	"Invalid ICCSN"
accessDeniedException	-	"Access denied"
timeoutException	-	"Timeout"
subscriptionException	-	„Remote SMC-B PIN not enabled“
subscriptionException	-	"A subscription already exists for this card"

3.9 Remote Pairing – Service: Dialog

3.9.1 Methoden, Parameter und Response Objekte

3.9.1.1 dialogRequestTopic - subscription

```
{
	"subscription": {
		"token": "699b8096-45ba-45be-9c54-bcc96920fe85",
		"service": "Dialog",
		"topic": {
			"dialogRequestTopic": {
				"sessionId": "1a4a6c7f-53e4-40d4-bc4a-9a64d3068105",
				"purposes": ["pairingEHealthTerminal"]
			}
		}
	}
}
```

Subscription-Topic	Parameter	Beschreibung
dialogRequestTopic		
	„sessionId“	Session-ID
	„purposes“	Dialog Message, für die sich registriert werden soll. Derzeit wird hier nur "pairingEHealthTerminal" unterstützt.

3.9.1.2 dialogRequestTopic - subscription - response

Im positiven Fall erhält man im „result“ eine subscriptionId zurück, die man benötigt um die Registrierung rückgängig zu machen („cancellation“) und die als Parameter einer „notification“ mit angegeben wird.

```
{
	"response": {
		"token":   "d2f24d46-f86b-40de-903f-132befad72c4",
		"service": "Dialog",
		"result":  "e25e86c2-81be-4a07-ac77-db39dc764b2f"
	}
}
```

Result Objekt	Parameter	Beschreibung
„String“	-	Die subscriptionId, die für das Beenden der Subscription (cancellation) benötigt wird und in der gesendeten Notification enthalten ist.

Im Fehlerfall wird eine Exception generiert und zurückgemeldet. Hier am Beispiel dargestellt, dass eine Subscription nicht möglich ist, weil Remote-Pairing deaktiviert ist:

```
{
	"failure": {
		"token":"d2f24d46-f86b-40de-903f-132befad72c4",
		"service":"Dialog",
		"exception": {
			"subscriptionException": {
```

```
                        "description":       "Remote Pairing not enabled."
                  }
            }
      }
}
```

3.9.1.3 dialogRequestTopic - subscription - cancellation

```
{
      "cancellation": {
            "token": "4cdf29da-5b45-478f-af4f-421106fcad45",
            "service": "Dialog",
            "subscriptionId": "e25e86c2-81be-4a07-ac77-db39dc764b2f"
      }
}
```

Cancellation	Parameter	Beschreibung
	„subscriptionId“	Parameter subscriptionId, die als Antwort der subscription im result Parameter bezogen wurde.

3.9.1.4 dialogRequestTopic – cancellation - response

```
{
      "response": {
            "token":   "4cdf29da-5b45-478f-af4f-421106fcad45",
            "service": "Dialog",
            "result":   null
      }
}
```

Result Objekt	Parameter	Beschreibung
null	-	Positive Bestätigung. Subscription erfolgreich beendet..

3.9.1.5 dialogPromptEvent - notification

Benachrichtigung, wenn der Konnektor die Pairing Anfrage zum Terminal sendet:

```
{
      "notification": {
            "subscriptionId": "e25e86c2-81be-4a07-ac77-db39dc764b2f",
            "service": "Dialog",
            "event": {
                  "dialogPromptEvent": {
                        "dialogId": "d1a1061d-4444-4444-4444-cccccccccccc",
                        "purpose": "pairingEHealthTerminal",
                        "severity": "inquiry",
                        "priority": 0,
                        "title": "Pairing",
                        "message": "KT:000DF8:16CD2 MIT KON:ICW-Kon_1 PAIREN",
                        "confirmationButton": "ok",
                        "rejectionButton": "cancel",
                        "submitTimeoutSeconds": 300
                  }
            }
      }
}
```

Notification-Event	Parameter	Beschreibung
	„subscriptionId“	Parameter, der als Antwort der Subscription bezogen wurde.
dialogPromptEvent		Liste an Informationen, die für den Remote Pairing-Eingabedialog verwendet werden können. Kommen zum Teil vom Konnektor.
	„dialogId“	Eindeutige Dialog ID. Kann zukünftig zum Differenzieren von Dialog-Messages verwendet werden.
	„purpose“	Für welchen Zweck ist die Dialog Message vorgesehen. Derzeit kann hier nur „pairingEHealthTerminal“ zurückkommen.
	„severity“	Art der Dialogmessage. Für den Remote Pairing Dialog wird hier immer „inquiry“ zurückgeben. Für zukünftige Dialoge könnten aber auch „info“, „warning“ oder „error“ enthalten sein.
	„priority“	Derzeit immer 0.
	„titel“	Potentieller Titel für die Dialog Message.
	„message“	Dialogmessage vom Konnektor.
	"confirmationButton"	Potentielle Klasse des Bestätigungs-Buttons – derzeit immer „ok“. Zukünftig können weitere Werte aus dem Wertebereich: „noButton“, „yes“, „apply“, „accept“, „agree“, „proceed“, „submit“, „close“ hinzukommen.
	"rejectionButton"	Potentielle Klasse des Buttons zum Abbrechen des Dialoges – derzeit immer „cancel“. Zukünftig können weitere Werte aus dem Wertebereich: „noButton“, „no“, „reject“ hinzukommen.
	„submitTimeoutSeconds“	In dieser Zeit muss der Dialog mit OK bestätigt oder mit Cancel abgebrochen werden.

Benachrichtigung, wenn der Dialog an der lokalen Schnittstelle bestätigt oder abgebrochen wurde oder ein Timeout aufgetreten ist:

```
{
	"notification": {
		"subscriptionId": "e25e86c2-81be-4a07-ac77-db39dc764b2f",
		"service": "Dialog",
		"event": {
			"dialogCloseEvent": {
				"dialogId": "d1a1061d-4444-4444-4444-cccccccccccc"
			}
		}
	}
}
```

Notification-Event	Parameter	Beschreibung
	„subscriptionId“	Parameter, der als Antwort der Subscription bezogen wurde.
dialogCloseEvent		Dialog wurde beendet.
	„dialogId“	Eindeutige Dialog ID. Kann zukünftig zum Differenzieren von Dialog-Messages verwendet werden.

3.9.1.6 submit - request

Beispiel: Pairing Anfrage wurde mit OK bestätigt:

```
{
	"request": {
		"token": "84206424-dc8b-4413-b8b6-279a77e9e0a7",
		"service": "Dialog",
		"method": {
			"submit": {
				"sessionId": "d1f91582-38f8-44ab-84d6-6b442677c3f6",
				"dialogSubmission": {
					"dialogId": "d1a1061d-4444-4444-4444-cccccccccccc",
					"behavior": "confirmed",
				}
			}
		}
	}
}
```

Methode	Parameter	Beschreibung
submit		
	„sessionId“	Session-ID
	"dialogSubmission"	Antwort auf die Dialoganfrage
	"dialogId"	Eindeutige Dialog ID aus der notification.
	"behavior"	Antwort auf die Dialoganfrage – welcher Knopf wurde gedrückt? Mögliche Werte sind hierbei: „confirmed“, „rejected“ oder „ignored“ (Timeout).

3.9.1.7 submit - response

Beispiel – Dialog Antwort erfolgreich verarbeitet:

```
{
	"response": {
		"token": "84206424-dc8b-4413-b8b6-279a77e9e0a7",
		"service": "Dialog",
		"result": null
	}
}
```

Result Objekt	Parameter	Beschreibung
null	-	Positive Bestätigung. Dialog Antwort erfolgreich verarbeitet.

Beispiel – Dialog Antwort konnte nicht verarbeitet werden (wurde ggf. bereits abgebrochen z.B. durch ein Timeout):

```
{
	"failure": {
		"token": "84206424-dc8b-4413-b8b6-279a77e9e0a7",
		"service":"Dialog",
		"exception": {
			"unexpectedRequestException": {
				"description": "Unexpected request"
			}
		}
	}
}
```

3.9.2 Fehlermeldungen / Exceptions

Die nachfolgend beschriebenen Fehlermeldungen können spezifisch für den Service „Dialog“ auftreten:

Exception Fehlertypen	Fehlermeldung	Beschreibung
authException	permissionDenied	"Permission denied"
unexpectedRequestException		„Unexpected request“
unexpectedRequestException		„No subscription found“
invalidArgumentException	-	"Invalid Argument"
accessDeniedException	-	"Access denied"
timeoutException	-	"Timeout"
subscriptionException	-	„Remote Pairing not enabled“

4. Settings - Parameter

In diesem Kapitel werden alle an dieser Schnittstelle verfügbaren Parameter, sortiert in funktionale Gruppen, aufgelistet. Diese Parameter haben unterschiedliche Datentypen und Wertebereiche. In den Spalten Admin und Anonymous / PIN-Provider ist für die jeweilige Benutzerrolle festgelegt, ob der Parameter verändert (W – Write) oder nur ausgelesen werden kann (R – Read). Die Spalte Neustart gibt bei veränderbaren Parametern darüber Auskunft, ob ein Neustart (Y - Yes) des Terminals für das Wirksamwerden der Änderung notwendig ist. VPN-Parametern werden hier sonderbehandelt, siehe Fußnoten. Bei dem Eintrag (N – No) wird eine Änderung – ohne Neustart - sofort übernommen. Ein (-) gibt an, dass ein Parameter Read-Only ist.

4.1 LAN & NTP

LAN-Parameter (Basis Daten)

Beschreibung	ID	Typ	Wertebereich	Admin	Anonymous/ PinProvider	Neustart
SICCT Terminal Name	sys_terminalName	String	min. 1, max. 32 Zeichen Erlaubte Zeichen: [a-z], [A-Z], [0-9] und [-]	R/W	R	Y
Host Name	net_hostname	String	min. 1, max. 32 Zeichen Erlaubte Zeichen: [a-z], [A-Z], [0-9] und [-], wobei [-] nicht als letztes Zeichen stehen darf	R/W	R	Y
Sprache einstellen (de/en/fr)	sys_locale	String	de, en, fr	R/W	R	Y
DHCP einschalten	net_lan_dhcpEnabled	Boolean	true/false	R/W	R	Y
DHCP erweiterte Optionen	net_lan_dhcpOptEnabled	Boolean	true/false	R/W	R	Y
IP Adresse (aktuell: DHCP oder statisch)	net_lan_ipAddr	String	IP-v4-Addr	R	R	-
Subnet Mask (aktuell: DHCP oder statisch)	net_lan_subnetMask	String	IP-v4-Addr	R	R	-
Gateway Adresse (aktuell: DHCP oder statisch)	net_lan_gatewayIpAddr	String	IP-v4-Addr	R	R	-
IP Adresse (statisch)	net_lan_ipAddrStatic	String	IP-v4-Addr	R/W	R	Y
Subnet Mask (statisch)	net_lan_subnetMaskStatic	String	IP-v4-Addr	R/W	R	Y
Gateway Adresse (statisch)	net_lan_gatewayIpAddrStatic	String	IP-v4-Addr	R/W	R	Y
IP Adresse (DHCP)	net_lan_ipAddrDhcp	String	IP-v4-Addr	R	R	-

Beschreibung	ID	Typ	Wertebereich	Admin	Anonymous/ PinProvider	Neustart
Subnet Mask (DHCP)	net_lan_subnetMaskDhcp	String	IP-v4-Addr	R	R	-
Gateway Adresse (DHCP)	net_lan_gatewayIpAddrDhcp	String	IP-v4-Addr	R	R	-
Broadcast Adresse	net_lan_ipAddrBroadcast	String	IP-v4-Addr	R	R	-
DNS Adresse	net_dns	String	IP-v4-Addr	R/W	R	Y
IP Adresse (SICCT Announcement)	sicct_ipAddr	String	IP-v4-Addr	R	R	-
SICCT TCP Port	sicct_tcpPort	Number	1-65535	R/W	R	Y
SICCT UDP Port	sicct_udpPort	Number	1-65535	R/W	R	Y
Unicast Adresse (optional) Kann für das zielgerichtete Versenden von Service-Announcement, z.B. in VPN-Netzen, benutzt werden. Ersetzt die ID “net_vpn_ipAddrSicctUnicast“	sicct_ipAddrUnicast	String	IP-v4-Addr	R/W	R	Y

NTP-Client

Beschreibung	ID	Typ	Wertebereich	Admin	Anonymous/ PinProvider	Neustart
NTP Client einschalten	sys_ntp_enabled	Boolean	True/false	R/W	R	Y
NTP Server IP Adresse	sys_ntp_serverIpAddr	String	IP-v4-Addr	R/W	R	Y
Angabe der Zeitquelle	sys_timeSource	String	[SysDefault, RTC, NTP, Cert)	-	-	-
NTP Zeitzone	sys_locale_timeZone	String	MEZ-1MEZ-2, M3.5.0,M10.5.0	R/W	R	Y

4.2 VPN

VPN-Parameter 1 / 3

Beschreibung	ID	Typ	Wertebereich	Admin	Anonym/ PinProv.	Neustart
VPN Client - Aktivierungsstatus	net_vpn_client_enabled	Boolean	true/false	R/W	R	Y[1]
VPN Server - Gateway Adresse	net_vpn_server_gateway	String	IP-v4-Addr, gültige Domain (max. 255)	R/W	R	Y[1]
VPN Server - CA-Zertifikat Öffentlicher Schlüssel	net_vpn_server_caCertificate	String (PEM)	während Import vpnCaCertSubject, vpnCaCertIssuer, vpnCaCertSn, vpnCaCertCxd ermitteln und setzen	W	-	Y[1]
VPN Server - Hash Wert CA-Zertifikat (Fingerprint)	net_vpn_server_caCertificateHash	String	0 oder 95 Zeichen	R	-	-
VPN Client - Authentifizierung Methode	net_vpn_client_authMode	String	None| EapMsChapV2| PubKey| Psk| EapTls	R/W	R	Y[1]
VPN Client - Benutzer Kennung VPN Zugang	net_vpn_client_userId	String	min.0, max.32 (Leerstring=Löschen)	R/W	-	Y[1]
VPN Client - Benutzer Passwort VPN Zugang oder PKCS 12 Container	net_vpn_client_passwd	String	min.0, max.32	W	-	Y[1]
VPN Client - PreSharedKey IPsec Client	net_vpn_client_preSharedKey	String	min.0, max.32	W	-	Y[1]
VPN Client - Privater Schlüssel	net_vpn_client_privateKey	String (PEM)	während Import vpnAuthPublicKeyHash berechnen und setzen	W	-	Y[1]
VPN Client - Hash Wert Privater Schlüssel (Fingerprint)	net_vpn_client_privateKeyHash	String	0 oder 95 Zeichen	R	-	-
VPN Client - Zertifikat Öffentlicher Schlüssel	net_vpn_client_certificate	String (PEM)	während Import vpnClientCertIssuer, vpnClientCertSubject, vpnClientCertCn, vpnClientCertSn, vpnClientCertCxd ermitteln und setzen	W	-	Y[1]

[1] Bei Veränderung von VPN-Parametern wird nach dem Abspeichern ein Geräte-Neustart empfohlen. Technisch notwendig ist nur das Setzen des Parameters VPN Client – Aktivierungsstatus.

VPN-Parameter 2 / 3

Beschreibung	ID	Typ	Wertebereich	Admin	Anonym/ PinProv.	Neustart
Unicast Adresse bei aktiven VPN (optional) **Wird in späteren Releases nicht mehr unterstützen und ist durch sicct_ipAddrUnicast ersetzt worden.**	net_vpn_ipAddrSicctUnicast	String	IP-v4-Addr	R/W	R	Y

VPN-Parameter 3 / 3

Beschreibung	ID	Typ	Wertebereich	Admin	Anonym./ PinProv.	Neustart
VPN Client - Hash Wert Zertifikat (FP)	net_vpn_client_certificateHash	String	0 oder 95 Zeichen	R	-	-
VPN Client - Hash Wert Zertifikat Public Key (Fingerprint)	net_vpn_client_certificatePubKeyHash	String	0-9, A-F, ':'	R	-	-
VPN Client - Konfiguration PKCS 12 Container	net_vpn_pkcs12	BIN	variable Länge (max. 4k)	W	-	Y[2]
VPN Client - Passwort PKCS 12 Container	net_vpn_pkcs12_passwd	String	min.0, max.32 (Leerstring=Löschen) Alphanumerische Zeichen mit Umlauten und folgenden Sonderzeichen: öÖäÄüÜß!#$%&()*+./\:;<=>?@-	W	-	Y[2]
VPN Client - Konfiguration	net_vpn_client_configuration	String	Text-File, max. 4k (Leerstring=Löschen)	R/W	-	Y[2]
VPN Client - Certificate Signing Request file	net_vpn_client_csr	String (PEM)	Textdatei, base64 Zeichen, Zeilenumbruch Textdatei im PEM-Format	R	-	-
VPN Client - Dead-per-detection Delay (DPD)	net_vpn_client_dpdDelaySeconds	Number	1-65535	R/W	R	Y[2]
VPN Client – Library Name	net_vpn_client_name	String	HW 1.2.0: "strongswan" HW 2.0.0: "strongswan" oder "wireguard"	R/W	-	Y[2]
IP Adresse VPN	net_vpn_ipAddr	String	IP-v4-Addr	R	R	-
VPN Status / Kurzinfo	net_vpn_shortInfo	String	max.4k Zeichen	R	R	-
VPN Status / Information	net_vpn_connected	Boolean	true/false	R	R	-
VPN Server CA Zertifikat Aussteller	net_vpn_server_caCertificateIssuer	String	gemäß ISO/IEC 9594-8 (X.509)	R	R	-
VPN Server CA Zertifikat Name	net_vpn_server_caCertificateSubject	String	gemäß ISO/IEC 9594-8 (X.509)	R	R	-
VPN Server CA Zertifikat Serien Nr	net_vpn_server_caCertificateSerial	String	gemäß ISO/IEC 9594-8 (X.509)	R	R	-
VPN Server CA Zertifikat Ablauf Datum	net_vpn_server_caCertificateCxd	String	dd.mm.yyyy HH:MM	R	R	-
Anzahl der importierten CA-Zertifikate	net_vpn_server_caCertificateCounter	Number	0-65535	R	R	-
VPN Client Zertifikat Aussteller	net_vpn_client_certificateIssuer	String	gemäß ISO/IEC 9594-8 (X.509)	R	R	-
VPN Client Zertifikat Name	net_vpn_client_certificateSubject	String	gemäß ISO/IEC 9594-8 (X.509)	R	R	-
VPN Client Zertifikat Common Name	net_vpn_client_certificateCommonName	String	gemäß ISO/IEC 9594-8 (X.509)	R	R	-
VPN Client Zertifikat Serien Nr	net_vpn_client_certificateSerial	String	gemäß ISO/IEC 9594-8 (X.509)	R	R	-
VPN Client Zertifikat Ablauf Datum	net_vpn_client_certificateCxd	String	dd.mm.yyyy HH:MM	R	R	-

[2] Bei Veränderung von VPN-Parametern wird nach dem Abspeichern ein Geräte-Neustart empfohlen. Technisch notwendig ist nur das Setzen des Parameters VPN Client – Aktivierungsstatus.

4.3 SICCT Parameter

Beschreibung	ID	Typ	Wertebereich	Admin	Anonym./ PinProv.	Neustart
Keep Alive						
Keepalive Intervall [s]	sicct_keepAliveIntervalSec	Number	1-10	R/W	R	Y
Keepalive Timeout [s]	sicct_keepAliveTimeoutSec	Number	120-300	R/W	R	Y
Protokoll						
Block Read Timeout [s]	sicct_blockReadTimeoutSec	Number	5-60	R/W	R	Y
Message Read Timeout [s]	sicct_messageReadTimeoutSec	Number	5-600	R/W	R	Y
Max. Protokollfehler	sicct_maxErrors	Number	5-60	R/W	R	Y
SSL Accept Timeout [s]	sicct_tls_acceptTimeoutSec	Number	1-30	R/W	R	Y
TLS Einstellung						
TLS Version auswählen	sicct_tls_version	String	TlsVersion_1_2	R/W	R	Y
CA Liste auswählen	sicct_tls_caList	String	pu|ru|tu|lu	R/W	R	Y
Announcement						
Announcement Intervall [s]	sicct_announcementIntervalSec	Number	0-3000	R/W	R	Y
Pairings						
Dialog Timeout [s]	sicct_pairing_submitTimeoutSec	Number	30-600	R/W	R	N
Admin Session / Rechte						
Admin Session	sicct_adminSessionEnabled	Boolean	true/false	R/W	R	N
Set Status	sicct_access_setStatusEnabled	Boolean	true/false	R/W	R	N
Download	sicct_access_downloadEnabled	Boolean	true/false	R/W	R	N

4.4 Netzwerkstatus / Verbindungsstatus

Beschreibung	ID	Typ	Wertebereich	Admin	Anonym./ PinProv.	Neustart
TSL Status	tls_connction_status	String	"TLS closed"| "TLS down HOST"| "TLS connected"| "SICCT open"	R	R	-
SICCT Session Status	sicct_session_status	String	Closed| OpenAdmin| OpenControl| OpenDownload	R	R	-
SICCT CI	sicct_commandPermission	String	Undefined| RestrictedInterpreter| RestrictedInterpreterWithAuth| Unrestricted	R	R	-
DHCP Client Status	net_lan_dhcpClientInfo	String	"-"| PREINIT| BOUND| RENEW| REBIND| REBOOT	R	R	-
Aktive Pairing Block Nummer	sicct_pairing_active_blockNumber	Number	0-3	R	-	-
Aktiver SICCT Public Key	sicct_pairing_active_pubKeyModulus	String	0-9, A-F (variable Länge)	R	-	-
Nummer des Public Keys Aktiver Pairing Block	sicct_pairing_active_pubKeyNumber	Number	0-3	R	-	-
Admin PIN - Temporärer Fehlerzähler	sys_localAdmin_pin_errCount	Number	0-65535	R	-	-
Admin PIN - Temporäre Rest-Sperrzeit in Sekunden	sys_localAdmin_pin_lockDuration	Number	0-86400	R	-	-
Session PIN - Temporärer Fehlerzähler	sicct_sessionAdmin_pin_errCount	Number	0-65535	R	-	-
Session PIN - Temporäre Rest-Sperrzeit in Sekunden	sicct_sessionAdmin_pin_lockDuration	Number	0-86400	R	-	-
TLS Client IP Adresse	tls_client_ipAddr	String	IP-v4-Addr	R	R	-
TLS Client Authenticate Subject	tls_client_authSubject	String	gemäß ISO/IEC 9594-8 (X.509)	R	R	-
TLS Client Authenticate Issuer	tls_client_authIssuer	String	gemäß ISO/IEC 9594-8 (X.509)	R	R	-
TLS Client Auth.-Zertifikat Fingerprint (SHA256)	tls_client_authFingerprint	String	0 oder 64 Zeichen (Hex-String)	R	R	-
TLS Client Authenticate Algorithmus	tls_client_pubKeyAlgo	String	"EC" oder "RSA"	R	R	-
TLS Client Authenticate EC Curve	tls_client_ecGroup	String	Name der EC-Kurve des Client-Zertifikats. z.B. "brainpoolP256r1"	R	R	-

Beschreibung	ID	Typ	Wertebereich	Admin	Anonym./ PinProv.	Neustart
TLS Server Authenticate Subject	tls_server_authSubject	String	gemäß ISO/IEC 9594-8 (X.509)	R	R	-
TLS Server Authenticate Issuer	tls_server_authIssuer	String	gemäß ISO/IEC 9594-8 (X.509)	R	R	-
TLS Server Auth.-Zertifikat Fingerprint (SHA256)	tls_server_authFingerprint	String	0 oder 64 Zeichen (Hex-String)	R	R	-
TLS Server Authenticate Algorithmus	tls_server_pubKeyAlgo	String	"EC" oder "RSA"	R	R	-
TLS Server Authenticate EC Curve	tls_server_ecGroup	String	Name der EC-Kurve des Server- Zertifikats. z.B. "brainpoolP256r1"	R	R	-
TLS Cipher Suite SICCT aktiv	tls_cipherSuite	String	Name der verwendete TLS-Ciphersuite z.B. "ECDHE-ECDSA-AES256-CGM-SHA384"	R	R	-

4.5 Benutzer Interface

Beschreibung	ID	Typ	Wertebereich	Admin	Anonym./ PinProv.	Neustart
Display						
Freitext auf Ruhebildschirm	gui_idleMessage	String	max. 46 Alphanum. Zeichen mit Umlauten (inkl. Leerzeich.), sowie ß und Sonderzeichen: !?#$&*_/-+ sowie '.' ',' ';' ':'	R/W	R	N
gSMC-KT Warnmeldung einschalten	gui_smkt_expirationWarningEnabled	Boolean	true/false	R/W	R	N
Töne						
Tasten Klick	sound_keyClickEnabled	Boolean	true/false	R/W	R	N
Akustische Signale	sound_acousticSignalingEnabled	Boolean	true/false	R/W	R	N
Start Jingle	sound_jingleEnabled	Boolean	true/false	R/W	R	N
Lautstärke	sound_volume	Number	0-100	R/W	R	N
Lautstärke PIN Schutz	sound_pinNoiseVolume	Number	0-100	R/W	R	N
Kiosk Modus						
Kiosk Modus einschalten	gui_kioskModeEnabled	Boolean	true/false	R/W	R	N

4.6 Firmware Update per TFTP

Beschreibung	ID	Typ	Wertebereich	Admin	Anonym./ PinProv.	Neustart
Datei Name	update_fileName	String	min. 1, max. 32 Zeichen	R/W	R	N
TFTP Server IP Adresse	update_serverIpAddr	String	IP-v4-Addr	R/W	R	N
TFTP Polling Status	update_tftp_poll_status	Boolean	true/false	R/W	R	N
TFTP Poll Timing	update_tftp_poll_window	String	noWait| wait_15_Sec| wait_255_Sec| wait_4096_Sec	R/W	R	N

4.7 Remote Management Interface / Remote Admin Passwort

Beschreibung	ID	Typ	Wertebereich	Admin	Anonym./ PinProv.	Neustart
Remote Management (RMI)						
RMI Aktivierung Ein/Aus	rmi_sessionEnabled	Boolean	true/false	R/W	R	Y
RMI Timeout in Sekunden (Default ist 30 Sekunden)	rmi_timeout	Number	10-600	R/W	R	Y
RMI Server Zertifikat	rmi_server_certificate	String (PEM)	0-4000	R/W	R	Y
RMI Server Zertifikat Signing Request	rmi_server_csr	String (PEM)	0-1000 Textdatei, base64 Zeichen, Zeilenumbruch Textdatei im PEM-Format	R	-	-
RMI Server Zertifikat Default Länderkennung (C)	net_cert_def_country	String	"DE" / 2	R	R	-
RMI Server Zertifikat Default Bundesland (ST)	net_cert_def_state	String	"Schleswig-Holstein" / 1-64	R	R	-
RMI Server Zertifikat Default Stadt (L)	net_cert_def_city	String	"Flintbek" / 1-64	R	R	-
RMI Server Zertifikat Defalt Organisation (O)	net_cert_def_org	String	"Worldline Healthcare GmbH" / 1-64	R	R	-
RMI Server Zertifikat Default Name (CN)	net_cert_def_name	String	"ORGA6100- <Serien Nummer>" / 1-64	R	R	-
RMI Server Zertifikat Default eMail Adresse (emailAddress)	net_cert_def_email	String	kontakt.whc@worldline.com /1-64	R	R	-
RMI Server Zertifikat Länderkennung (C)	rmi_server_new_cert_country	String	0-2	R/W	R	N

Beschreibung	ID	Typ	Wertebereich	Admin	Anonym./ PinProv.	Neustart
Remote Management (RMI)						
RMI Server Zertifikat Bundesland (ST)	rmi_server_new_cert_state	String	0-64	R/W	R	N
RMI Server Zertifikat Stadt (L)	rmi_server_new_cert_city	String	0-64	R/W	R	N
RMI Server Zertifikat Organisation (O)	rmi_server_new_cert_org	String	0-64	R/W	R	N
RMI Server Zertifikat Name (CN)	rmi_server_new_cert_name	String	1-64	R/W	R	N
RMI Server Zertifikat eMail Adresse (emailAddress)	rmi_server_new_cert_email	String	0-64	R/W	R	N
RMI Server Zertifikat Start Zeitpunkt	rmi_server_new_cert_notb	String	"DD.MM.JJJJ" / 10	R/W	R	N
RMI Server Zertifikat Ablaufdatum	rmi_server_new_cert_nota	String	"DD.MM.JJJJ" / 10	R/W	R	N

4.8 Remote SMC-B PIN und Remote Pairing Feature

Beschreibung	ID	Typ	Wertebereich	Admin	Anonym./ PinProv.	Neustart
Remote SMC-B PIN einschalten	rmi_smcb_pinEnabled	Boolean	true/false	R/W	R	N
Remote Pairing einschalten	rmi_pairingEHealthTerminal_enabled	Boolean	true/false	R/W	R	N

4.9 Selbstauskunft

Beschreibung	ID	Typ	Wertebereich	Admin	Anonym./ PinProv.	Neustart
Serien Nummer	vendor_serialNumber	String	14-character	R	R	-
Firmware Version	sys_firmwareVersion	String	Version-String "MMM.mmm.PPP"	R	R	-
Firmware Datum	sys_firmwareBuildDate	String	ISO-Date "JJJJ-MM-TT"	R	R	-
Firmware Gruppe (ID)	update_allowedVersionSetId	Number	00001-99999	R	R	-
Firmware Gruppe	update_allowedVersionSet	String	Kommaseparierter Versions-String	R	R	-
Hardware Version	vendor_hardwareVersion	String	Version-String "MMM.mmm.PPP"	R	R	-
Hersteller ID	vendor_deviceManufacturerId	String	max 5-stellig "INGHC"	R	R	-
Produkt Kürzel	vendor_deviceModelName	String	"ORGA6100"	R	R	-
Produkt Name	vendor_productName	String	"ORGA 6141 online"	R	R	-
Version der CI-Partition	vendor_ciVersion	String	Version-String "MMM.mmm.PPP"	R	-	-
Produkt Webseite	vendor_url_productInfo	String	html-Link	R	-	-
Produkt Typ Version	vendor_productVersion	String	Version-String "MMM.mmm.PPP"	R	R	-
TLS - Listen						
Version TSL - PU	tsl_puVersion	String	Version-String "MMM.mmm.PPP"	R	R	-
Version TSL - RU	tsl_ruVersion	String	Version-String "MMM.mmm.PPP"	R	R	-
Version TSL - TU	tsl_tuVersion	String	Version-String "MMM.mmm.PPP"	R	R	-
Version TSL - LU	tsl_luVersion	String	Version-String "MMM.mmm.PPP"	R	R	-
Version TSL - SU	tsl_suVersion	String	Version-String "MMM.mmm.PPP"	R	R	-
LAN MAC Addresse	net_lan_macAddr	String	Hex "00:0D:F8:XX:XX:XX"	R	R	-
Update ID						
Aktuelle TFTP Update ID	update_tftp_poll_id	String	max. 32 Zeichen	R	R	-
TFTP Update Error Code	update_tftp_poll_idErr	Number	0-65535	R	R	-

4.10 Kartenslots

Beschreibung	ID	Typ	Wertebereich	Admin	Anonym./ PinProv.	Neustart
Status Kartenslots						
Slot 1 Icon-Status	card_slot1_status	Number	1-2	R	R	-
Slot 1 Kartentyp	card_slot1_cardType	Number	0-7	R	R	-
Slot 2 Icon-Status	card_slot2_status	Number	1-2	R	R	-
Slot 2 Kartentyp	card_slot2_cardType	Number	0-7	R	R	-
Slot 3 Icon-Status	card_slot3_status	Number	1-2	R	R	-
Slot 3 Kartentyp	card_slot3_cardType	Number	0-7	R	R	-
Slot 4 Icon-Status	card_slot4_status	Number	1-2	R	R	-
Slot 4 Kartentyp	card_slot4_cardType	Number	0-7	R	R	-

Hinweis

Die Zahlenkombinationen in den Wertebereichen geben Auskunft über folgende Zustände an den Kartenslots:

Für die Slot ID: card_slotX_status:	Kartentype: card_slotX_cardType:
(1) = "Karte gesteckt"	(0) = "unbekannt"
(2) = "Kartenslot leer bzw. keine Karte gesteckt"	(1) = "Prozessor-Karte"
	(2) = "Speicher-Karte"
	(3) = "I2C-Bus Speicher-Karte" [3]
	(4) = "2-Draht Speicher-Karte" [3]
	(5) = "3-Draht Speicher-Karte" [3]
	(6) = "NFC-Karte" [3]
	(7) = "USB-Karte" [3]

[3] Dieser Kartentype wird in der Firmware Version V3.9.1 noch nicht angezeigt und ist für zukünftige Releases vorbereitet.

4.11 Geräte Karte gSMC-KT

Beschreibung	ID	Typ	Wertebereich	Admin	Anonym./ PinProv.	Neustart
Serien Nr.	card_smkt_iccsn	String	Card ICCSN	R	R	-
Version Nr.	card_smkt_version	String	"vMM.mm.PP"	R	R	-
Slot Nr.	card_smkt_slotNum	Number	3 oder 4	R	R	-
Zertifikat Typ	card_smkt_autType	String	"RSA" oder "EC"	R	R	-
Aktivierung	card_smkt_autCed	String	"TT.MM.JJJJ"	R	R	-
Gültigkeit bis	card_smkt_autCxd	String	TT.MM.JJJJ	R	R	-
Zertifikat Typ2	card_smkt_aut2Type	String	"RSA" oder "EC"	R	R	-
Aktivierung	card_smkt_aut2Ced	String	TT.MM.JJJJ	R	R	-
Gültigkeit bis	card_smkt_aut2Cxd	String	TT.MM.JJJJ	R	R	-
CVC Zertifikat Typ	card_smkt_rpsType	String	"RSA" oder "EC"	R	R	-
Aktivierung	card_smkt_rpsCed	String	TT.MM.JJJJ	R	R	-
Gültigkeit bis	card_smkt_rpsCxd	String	TT.MM.JJJJ	R	R	-

Sollte keine gSMC-KT im Terminal vorhanden sein, so wird für die Parameter vom Typ String ein “-“ zurückgegeben und für die Slot Nr. wird der Wert 0 per Default gesetzt.

4.12 Betriebsdaten / Statistik

Beschreibung	ID	Typ	Wertebereich	Admin	Anonym./ PinProv.	Neustart
Betriebsstunden gesamt	sys_uptime_durationTotal	String		R	R	-
Betriebsstunden seit Neustart	sys_uptime_durationSinceBoot	String		R	R	-
Betriebsstunden seit FW-Update	sys_uptime_durationSinceFwUpdate	String		R	R	-
Neustarts (Gesamtanzahl)	sys_boot_countTotal	Number	0-65535	R	R	-
Kaltstart (Spannung aus/ein)	sys_boot_countCold	Number	0-65535	R	R	-
Warmstarts (Gesamtanzahl)	sys_boot_countWarm	Number	0-65535	R	R	-
Warmstarts seit FW-Update	sys_boot_countFwUpdate	Number	0-65535	R	R	-
Warmstarts durch Menü initiiert	sys_boot_countMenue	Number	0-65535	R	R	-
Warmstarts durch Watchdog	sys_boot_countWatchdog	Number	0-65535	R	R	-
Anzahl Starts der SICCT-Applikation	sicct_daemon_startsTotal	Number	0-65535	R	R	-
Verbindungsabbrüche durch Terminal Gesamtanzahl	sicct_connDisconn_byTerm_countTotal	Number	0-65535	R	R	-
Verbindungsabbrüche durch Terminal seit Neustart	sicct_connDisconn_byTerm_countSinceBoot	Number	0-65535	R	R	-
Verbindungsabbrüche durch Konnektor Gesamtanzahl	sicct_connDisconn_byHost_countTotal	Number	0-65535	R	R	-
Verbindungsabbrüche durch Konnektor seit Neustart	sicct_connDisconn_byHost_countSinceBoot	Number	0-65535	R	R	-
Admin-PIN Anzahl erfolgreicher Eingaben	sys_localAdmin_pin_okTotal	Number	0-65535	R	-	-
Admin-PIN Anzahl falscher Eingaben	sys_localAdmin_pin_wrongTotal	Number	0-65535	R	-	-
Admin PIN Anzahl aufgetr. Sperrungen auf Zeit	sys_localAdmin_pin_locksTotal	Number	0-65535	R	-	-
Admin PIN temporärer Fehlerzähler	sys_localAdmin_pin_errCount	Number	0-65535	R	-	-
Admin PIN temporäre Rest-Sperrzeit in Sekunden	sys_localAdmin_pin_lockDuration	Number	0-86400	R	-	-
Session-PIN Anzahl erfolgreicher Eingaben	sicct_sessionAdmin_pin_okTotal	Number	0-65535	R	-	-
Session-PIN Anzahl falscher Eingaben	sicct_sessionAdmin_pin_wrongTotal	Number	0-65535	R	-	-
Session PIN - Anzahl aufgetr. Sperrungen auf Zeit	sicct_sessionAdmin_pin_locksTotal	Number	0-65535	R	-	-
Session PIN temporärer Fehlerzähler	sicct_sessionAdmin_pin_errCount	Number	0-65535	R	-	-
Session PIN - temporäre Rest-Sperrzeit in Sekunden	sicct_sessionAdmin_pin_lockDuration	Number	0-86400	R	.	-
SICCT Daemon - Anzahl Starts seit letztem Boot	sicct_daemon_startsSinceBoot	Number	1-65535	R	R	-
Anzahl Steckzyklen Slot 1	card_slot1_plugCycles	Number	0-65535	R	R	-
Anzahl Steckzyklen Slot 2	card_slot2_plugCycles	Number	0-65535	R	R	-
Anzahl Steckzyklen Slot 3	card_slot3_plugCycles	Number	0-65535	R	R	-
Anzahl Steckzyklen Slot 4	card_slot4_plugCycles	Number	0-65535	R	R	-

Beschreibung	ID	Typ	Wertebereich	Admin	Anonym./ PinProv.	Neustart
Slot 1 Anzahl erfolgreicher Kartenaktivierungen	card_slot1_activationSuccesful	Number	0-65535	R	R	-
Slot 1 Anzahl unlesbarer Karten	card_slot1_activationFailed	Number	0-65535	R	R	-
Slot 1 Anzahl Memorykarten	card_slot1_activationTypeMem	Number	0-65535	R	R	-
Slot 1 Anzahl Prozessorkarten	card_slot1_activationTypeProc	Number	0-65535	R	R	-
Slot 2 Anzahl erfolgreicher Kartenaktivierungen	card_slot2_activationSuccesful	Number	0-65535	R	R	-
Slot 2 Anzahl unlesbarer Karten	card_slot2_activationFailed	Number	0-65535	R	R	-
Slot 2 Anzahl Memorykarten	card_slot2_activationTypeMem	Number	0-65535	R	R	-
Slot 2 Anzahl Prozessorkarten	card_slot2_activationTypeProc	Number	0-65535	R	R	-
Slot 3 Anzahl erfolgreicher Kartenaktivierungen	card_slot3_activationSuccesful	Number	0-65535	R	R	-
Slot 3 Anzahl unlesbarer Karten	card_slot3_activationFailed	Number	0-65535	R	R	-
Slot 3 Anzahl Memorykarten	card_slot3_activationTypeMem	Number	0-65535	R	R	-
Slot 3 Anzahl Prozessorkarten	card_slot3_activationTypeProc	Number	0-65535	R	R	-
Slot 4 Anzahl erfolgreicher Kartenaktivierungen	card_slot4_activationSuccesful	Number	0-65535	R	R	-
Slot 4 Anzahl unlesbarer Karten	card_slot4_activationFailed	Number	0-65535	R	R	-
Slot 4 Anzahl Memorykarten	card_slot4_activationTypeMem	Number	0-65535	R	R	-
Slot 4 Anzahl Prozessorkarten	card_slot4_activationTypeProc	Number	0-65535	R	R	-
Slot 1 Protokollfehler Gesamtzahl	card_slot1_t1Error	Number	0-65535	R	R	-
Slot 1 Protokollfehler seit Neustart	card_slot1_t1ErrorSinceBoot	Number	0-65535	R	R	-
Slot 1 interner Fehlerzähler	card_slot1_fixNcn6001	Number	0-65535	R	R	-
Slot 1 interner Fehlerzähler seit letztem Boot	card_slot1_fixNcn6001SinceBoot	Number	0-65535	R	R	-
Slot 2 Protokollfehler Gesamtzahl	card_slot2_t1Error	Number	0-65535	R	R	-
Slot 2 Protokollfehler seit Neustart	card_slot2_t1ErrorSinceBoot	Number	0-65535	R	R	-
Slot 2 interner Fehlerzähler	card_slot2_fixNcn6001	Number	0-65535	R	R	-
Slot 2 interner Fehlerzähler seit letztem Boot	card_slot2_fixNcn6001SinceBoot	Number	0-65535	R	R	-
Slot 3 Protokollfehler Gesamtzahl	card_slot3_t1Error	Number	0-65535	R	R	-
Slot 3 Protokollfehler seit Neustart	card_slot3_t1ErrorSinceBoot	Number	0-65535	R	R	-
Slot 3 interner Fehlerzähler	card_slot3_fixNcn6001	Number	0-65535	R	R	-
Slot 3 interner Fehlerzähler seit letztem Boot	card_slot3_fixNcn6001SinceBoot	Number	0-65535	R	R	-

Beschreibung	ID	Typ	Wertebereich	Admin	Anonym./ PinProv.	Neustart
Slot 4 Protokollfehler Gesamtzahl	card_slot4_t1Error	Number	0-65535	R	R	-
Slot 4 Protokollfehler seit Neustart	card_slot4_t1ErrorSinceBoot	Number	0-65535	R	R	-
Slot 4 interner Fehlerzähler	card_slot4_fixNcn6001	Number	0-65535	R	R	-
Slot 4 interner Fehlerzähler seit letztem Boot	card_slot4_fixNcn6001SinceBoot	Number	0-65535	R	R	-
Gesamtanzahl Verb. Aufbauten	net_vpn_up_countTotal	Number	0-65535	R	R	-
Seit Neustart Anzahl Verb. Aufbauten	net_vpn_up_countSinceBoot	Number	0-65535	R	R	-
Gesamtanzahl Verb. Abbauten	net_vpn_down_countTotal	Number	0-65535	R	R	-
Seit Neustart Anzahl Verb. Abbauten	net_vpn_down_countSinceBoot	Number	0-65535	R	R	-
Gesamtanzahl Verb. Re-Trigger	net_vpn_restart_countTotal	Number	0-65535	R	R	-
Seit Neustart Anzahl Verb. Re-Trigger	net_vpn_restart_countSinceBoot	Number	0-65535	R	R	-
Total Anzahl der PHY-Restarts	net_phy_restartTotal	Number	0-65535	R	R	-
Anzahl Admin PIN Sperren total	rmi_admin_pin_locksTotal	Number	INT32	R	-	-
Anzahl Admin PINs ok total	rmi_admin_pin_okTotal	Number	INT32	R	-	-
Anzahl Admin PINs nok total	rmi_admin_pin_wrongTotal	Number	INT32	R	-	-
PIN Provider PIN temporärer Fehlerzähler	rmi_pinProvider_pin_errCount	Number	0-65535	R	-	-
PIN Provider PIN - temporäre Rest-Sperrzeit in Sekunden	rmi_pinProvider_pin_lockDuration	Number	0-65535	R	-	-
Anzahl PIN Provider PIN Sperren total	rmi_pinProvider_pin_locksTotal	Number	INT32	R	-	-
Anzahl PIN Provider PINs ok total	rmi_pinProvider_pin_okTotal	Number	INT32	R	-	-
Anzahl PIN Provider PINs nok total	rmi_pinProvider_pin_wrongTotal	Number	INT32	R	-	-

5. Web-Applikation

Das ORGA 6141 online / ORGA Neo bietet ab der Firmware Version V3.9.0 die Konfiguration der Geräte auch über eine mitgelieferte Web-Applikation an, die auf HTML, CSS (Cascading Style Sheets) und JavaScript basiert. Diese kann aus dem lokalen Netzwerk als auch über ein externes Netzwerk/Internet gestartet werden.

Mit dieser Web-Applikation ist der Administrator als auch allgemein der Leistungserbringer in der Lage, ohne zusätzliche Programmieraufwände das Kartenterminal aus der Ferne zu monitoren und zu administrieren. Zur Nutzung dieser Web-Applikation ist ledig ein plattformunabhängiges Device (PC, Notebook, Tablet, eingeschränkt auch ein Handy) notwendig, welches über ein Netzwerk mit dem Kartenterminal verbunden ist und eine entsprechende Zugangsberechtigung zur Web-Applikation hat, die Sie i.d.R. vom Geräte-Administrator bekommen.

5.1 Anbindung über VPN inkl. Portweiterleitung

Grundsätzlich kann die Web-Applikation des Terminals aus externen Netzwerken heraus auch über eine VPN-Verbindung erreicht werden, die auch das Administrieren des Kartenterminals über diesen VPN-Tunnel ermöglicht. In der Regel wird diese Verbindung über ein VPN-Gateway ermöglicht, wenn das Terminal eine VPN-Verbindung bereits zu einem TI-Gateway oder direkt zu einem Konnektor aufgebaut hat. Ab der Firmware 3.9.1 verfügt das Kartenterminal auch über eine Portweiterleitung, um sicher-zustellen, dass die RMI-Schnittstelle als auch die Web-Applikation erreichbar bleiben, auch wenn von außen über andere Ports als der interne Port 443 angefragt wird. Zusätzlich kann ein lokaler Admin vor Ort sich auch noch über das lokale Netzwerk direkte mit der Web-Applikation des Kartenterminals verbinden.

Ab der Firmware 3.9.1 unterstützt das ORGA Neo (Hardwareversion 2.0.0) zwei VPN-Protokolle:

- IPSec/IKEv2
- WireGuard

Für das IPSec/IKEv2 Protokoll stehen vier verschiedene Authentifizierungsmethoden zur Verfügung. Zur Konfiguration konsultieren Sie unser **VPN-Tutorial**, welches Ihnen auf unserer Webseite zum Download bereitsteht. Für die Konfiguration des VPN-Gateways kontaktieren Sie bitte ihren Netzwerk-Administrator.

5.2 Benutzerrollen & Berechtigungen

Die Web-Applikation unterscheidet 4 Benutzergruppen mit Ihren Rollen und Berechtigungen, wobei nur mit den ersten drei ein Login an der RMI-Schnittstelle möglich ist:

1. **Anonymous**
   Die Rolle des Anonymous ist etabliert worden, um Leseberechtigungen für ausgewählte Systeminformationen bereitzustellen, ohne sich mit Zugangsdaten einloggen zu müssen.

Bei der Web-Applikation betrifft das die Informationen, die auf dem Startbildschirm zur Verfügung gestellt werden.

2. **Remote Admin**
   Der Remote Admin hat Lese- und Schreibberechtigung für alle über die Web-Applikation freigegebenen Konfigurationseinträge, die Abfrage sämtlicher Systeminformationen, das Ausführen des Firmware Updates sowie das Ausführen von Systemfunktionen wie Neustart.

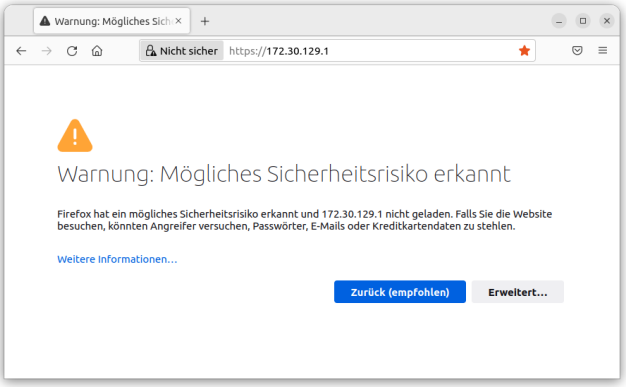
3. **pinProvider**
   Der pinProvider hat ausschließlich Rechte zur Ausführung / Freischaltung des Anwendungsfalls „Remote SMC-B PIN Eingabe“.
   Diese Rolle ermöglicht es einem Admin, einem weiteren Benutzer einen Fernzugriff ausschließlich auf die Freischaltung der SMC-B PIN zu geben, ohne dass dieser Benutzer weitere Parameter des Kartenterminals verändern kann. Dieser hat jedoch Leserechte auf die allermeisten Parameter.

4. **sicctAdmin**
   Hierbei handelt es sich um den Admin für die SICCT-Protokoll-Schnittstelle (Verbindung zum Konnektor). Über die RMI-Schnittstelle kann das Passwort (bzw. die PIN) dieses Benutzers geändert werden.

Die einzelnen Berechtigungen je Parameter finden Sie in Kapitel 4 Settings - Parameter.

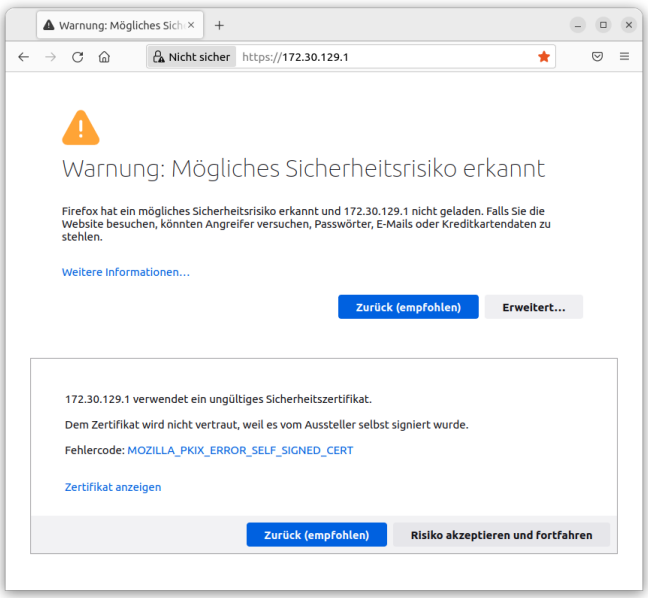
5.3 Webbrowser basierte Konfiguration

Grundsätzlich kann die Web-Applikation plattformunabhängig in verschiedenen Browsern gestartet werden. Die Browser müssen jedoch ein selbstsigniertes Zertifikat zum Ausführen der Webseite akzeptieren. Eine Option ist hier der Firefox Browser, anhand dem die nächsten Schritte beschrieben werden:
Starten Sie den Firefox Browser und geben in der Befehlszeile die von Ihnen vergebene IP-Adresse für das Kartenterminal ein und öffnen diese mit ENTER.
In unserem Beispiel hat das Kartenterminal die **IP-Adresse 172.30.129.1** erhalten.
Bitte geben sie immer **"https://<IP>"** vor der IP-Adresse an. Es wird dann folgende Seite angezeigt:



Grund für diese Warnmeldung ist, dass das Kartenterminal (der Web-Server) nicht über ein öffentlich signiertes Zertifikat verfügt, sondern mit einem selbst signierten Zertifikat arbeitet.

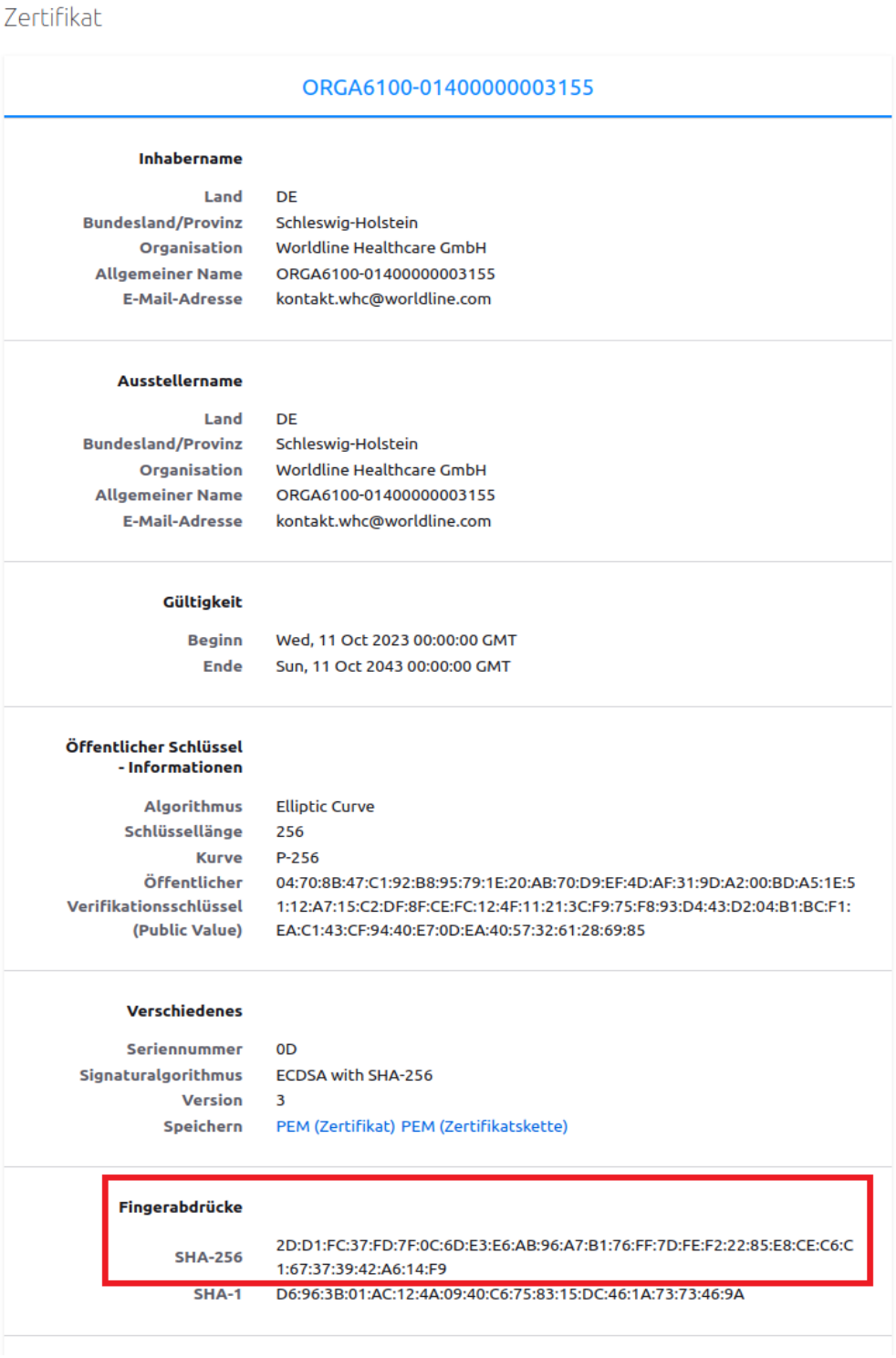
Da die Software aus einer vertrauenswürdigen Quelle stammt, klicken Sie auf den **Erweitert…**-Button. Es erscheint folgende Erklärung:



Hier haben Sie jetzt die Möglichkeit, sich das Zertifikat anzeigen zu lassen und den Fingerprint des Zertifikats mit dem hinterlegten Fingerprint dieses Zertifikats im Kartenterminal zu vergleichen.

Gehen Sie dafür bitte in Ihrem Browser auf den Link **Zertifikat anzeigen.**

Sie erhalten dann folgende Informationen über das im Kartenterminal hinterlegt Zertifikat:



Zertifikat

ORGA6100-01400000003155

Inhabername

Land	DE
Bundesland/Provinz	Schleswig-Holstein
Organisation	Worldline Healthcare GmbH
Allgemeiner Name	ORGA6100-01400000003155
E-Mail-Adresse	kontakt.whc@worldline.com

Ausstellername

Land	DE
Bundesland/Provinz	Schleswig-Holstein
Organisation	Worldline Healthcare GmbH
Allgemeiner Name	ORGA6100-01400000003155
E-Mail-Adresse	kontakt.whc@worldline.com

Gültigkeit

Beginn	Wed, 11 Oct 2023 00:00:00 GMT
Ende	Sun, 11 Oct 2043 00:00:00 GMT

Öffentlicher Schlüssel - Informationen

Algorithmus	Elliptic Curve
Schlüssellänge	256
Kurve	P-256
Öffentlicher Verifikationsschlüssel (Public Value)	04:70:8B:47:C1:92:B8:95:79:1E:20:AB:70:D9:EF:4D:AF:31:9D:A2:00:BD:A5:1E:51:12:A7:15:C2:DF:8F:CE:FC:12:4F:11:21:3C:F9:75:F8:93:D4:43:D2:04:B1:BC:F1:EA:C1:43:CF:94:40:E7:0D:EA:40:57:32:61:28:69:85

Verschiedenes

Seriennummer	0D
Signaturalgorithmus	ECDSA with SHA-256
Version	3
Speichern	PEM (Zertifikat) PEM (Zertifikatskette)

Fingerabdrücke

SHA-256	2D:D1:FC:37:FD:7F:0C:6D:E3:E6:AB:96:A7:B1:76:FF:7D:FE:F2:22:85:E8:CE:C6:C1:67:37:39:42:A6:14:F9
SHA-1	D6:96:3B:01:AC:12:4A:09:40:C6:75:83:15:DC:46:1A:73:73:46:9A

Im oberen Bereich finden Sie allgemein wichtige Informationen über den Inhaber, den Aussteller, die Gültigkeit sowie Basisinformation über den Zertifikatstyp. Im unteren Bereich (roter Kasten) befinden sich zwei sogenannte Fingerabdrücke / Fingerprints dieses Zertifikats.

Der SHA-256 Fingerprint ist auch im Kartenterminal hinterlegt. Zur Anzeige gehen Sie bitte in das Menü des Kartenterminals unter **Einstellungen > Remote Management Interface > Zertifikatsfingerprint** und vergleichen diesen Fingerprint mit dem Fingerprint aus Ihrem Browser.

Im vorliegenden Beispielfall würde auf ihrem Kartenterminal folgende Information angezeigt:



Durch diesen Vergleich können Sie sicherstellen, dass die Verbindung zwischen ihrem Kartenterminal und ihrem PC/Notebook/Tablet vertrauenswürdig ist.

Wichtiger Hinweis
Es ist möglich, dass von Firmen gemanagte Rechner das Ausführen einer Webseite mit einem selbst signierten Zertifikat nicht akzeptieren und damit das Ausführen der Webseite unterbunden wird.
In diesem Fall wenden Sie sich bitte an ihren IT-Administrator und lassen sich die IP-Adresse des Kartenterminals als Ausnahme für ihren Rechner eintragen und/oder freigeben.

Bestätigen Sie nun abschließend den Button **Risiko akzeptieren und fortfahren**, um auf die Startseite der Web-Applikation zu gelangen.

5.4 Web-Browser

Um die Web-Applikation zu bedienen, können grundsätzlich diverse handelsübliche Web-Browser auf unterschiedlichen Betriebssystemen eingesetzt werden. Eine fehlerfreie Anzeige und Bedienung können dabei jedoch nicht garantiert werden. Im Folgenden finden Sie zur Orientierung Browser-Produkte mit Ihren Versionsständen, die von uns eingesetzt und geprüft wurden. Auch hierbei ist anzumerken, dass in seltenen Fällen auch höhere Versionsstände zu Anzeige- oder Bedienfehlern führen können.
Trotzdem empfehlen wir, immer die aktuelle Browser-Version zu verwenden:

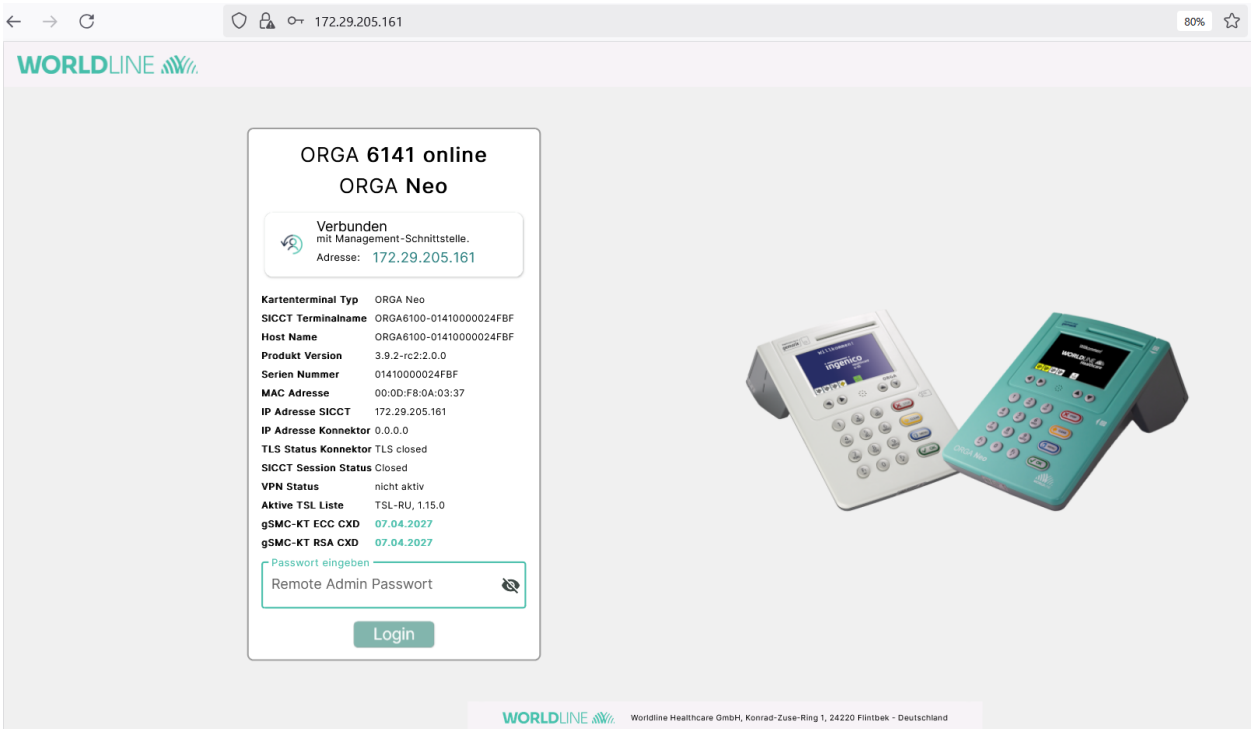
- Für Windows 10 – Systeme ab dem Funktionsupdate 22H2:
  - Microsoft Edge Version 121.0.2277.106 (64-Bit)
  - Firefox Version 122.0.1 (64-Bit)
  - Google Chrome Version 121.0.6167.161 (64-Bit)

- Für iOS-Systeme ab der Vers. 16.5:
  - Safari Version 18615.2.9.11.4

5.5 Web-Applikation

Auf dem Startbildschirm befinden sich erste Informationen zum angesprochenen Kartenterminal, die ohne Öffnen einer Admin Session zugänglich sind. Grundsätzlich kann die Firmware 3.9.0 und höher auf allen ORGA 6141 Online und ORGA Neo Geräten installiert und ausgeführt werden.

Damit ist es möglich, alle in der Telematikinfrastruktur (TI) installierten ORGA 6141 Online und ORGA Neo Geräte remotefähig zu machen und diese damit aus der Ferne zu administrieren sowie weitere Use Cases, wie das **SMC-B Remote PIN**-Verfahren oder auch das **Remote Pairing**-Verfahren, auszuführen.



Das Bild oben zeigt die Startseite der Web-Applikation mit den wichtigsten Terminaldaten, die bereits zur Verfügung gestellt werden, ohne dass Sie sich z.B. als RMI-Admin angemeldet haben.

Der **SICCT-Terminalname** und der **Host Name** haben initial den gleichen Namenseintrag, soweit das Geräte nicht individuell vorkonfiguriert wurde. Beide Namen können unter Einstellungen individuell verändert werden, so dass beide Namenseinträge von Bedeutung sein können. Daher wurden beide Parameter hier aufgenommen.

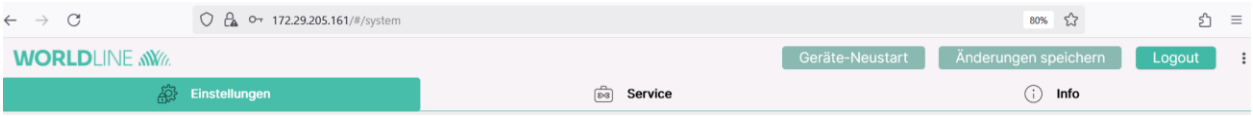
Die letzten beiden Positionen geben das Ablaufdatum des RSA und des ECC – Zertifikats (CXD: Certificate Expiration Date) der gesteckten gSMC-KT an. Die Anzeige bleibt bis zum 43. Tag vor Ablauf grün, vom 42. Tag an wird das Ablaufdatum ockerfarbig angezeigt. Ist das Ablaufdatum erreicht, wird es rot dargestellt.

Geben Sie jetzt ihre RMI Admin PIN ein, um eine RMI-Session zu eröffnen und auf die Hauptseite der Web-Applikation zu gelangen.

Diese Applikation ist sehr intuitiv aufgebaut und ermöglicht Ihnen die Einstellung und Anzeige fast aller Werte, die Sie bisher auch direkt am Kartenterminal sehen und einstellen konnten. Nur wenige Einstellungen, die sinnvollerweise nur direkt am Gerät vorzunehmen sind, wurden nicht in diese Web-Applikation übernommen.

5.5.1 Grundsätzlicher Aufbau

Diese Applikation ist in drei Hauptbereiche gegliedert und verfügt im oberen Bereich über drei Button. Der angewählte Bereich ist ab der FW 3.9.2 grün hinterlegt, die beiden jeweils inaktiven Bereiche sind grau hinterlegt:



1. **Einstellungen**
   Hier finden Sie alle Parameter, die Sie in der Regel verändern und an das Kartenterminal übertragen können, um das Terminal für den Betrieb zu konfigurieren (LAN, VPN, SICCT, uvm.).
2. **Service**
   Im Servicebereich finden Sie ausführbare Prozesse für das Terminal wie z.B. das **Firmware Update**, den **Geräte-Neustart,** den **Passwort-Änderungsservice,** das **Remote SMC-B PIN-Verfahren** sowie seit der FW 3.9.1 auch das **Remote Pairing Verfahren**.
3. **Info**
   Im Infobereich sind Geräte- und Statusinformationen zusammengefasst, die das Kartenterminal zur Anzeige bringen kann, die jedoch nicht vom Admin geändert werden können. Hier sind nun erstmalig auch alle **Betriebsdaten** übersichtlich angezeigt, die das Kartenterminal seit Installation der FW 3.8.0 über sich sammelt hat und eine wertvolle Informationsquelle für das Gerätemonitoring darstellt.
4. **Geräte-Neustart-Button**
   Der intelligente Geräte-Neustart-Button ist so lange inaktiv (leicht ausgegraut), bis vom System automatisch erkannt wird, dass ein Neustart notwendig ist. Wir empfehlen, diesen Neustart auch durchzuführen, auch wenn im Einzelfall (z.B. bei VPN-Einstellungen) das technisch nicht immer notwendig ist. Natürlich können Einstellungsänderungen auch gesammelt und dann abschließend mit einem Neustart aktiv gesetzt werden.
5. **Änderungen Speichern-Button**
   Der Änderungen Speichern-Button überträgt die Änderung im Datensatz in das Terminal. Einige Änderungen werden dabei sofort aktiv, andere benötigen einen Neustart. Der Neustart-Button ist so konfiguriert, dass er automatisch aktiv geschaltet wird, wenn ein Neustart erforderlich oder ratsam ist.

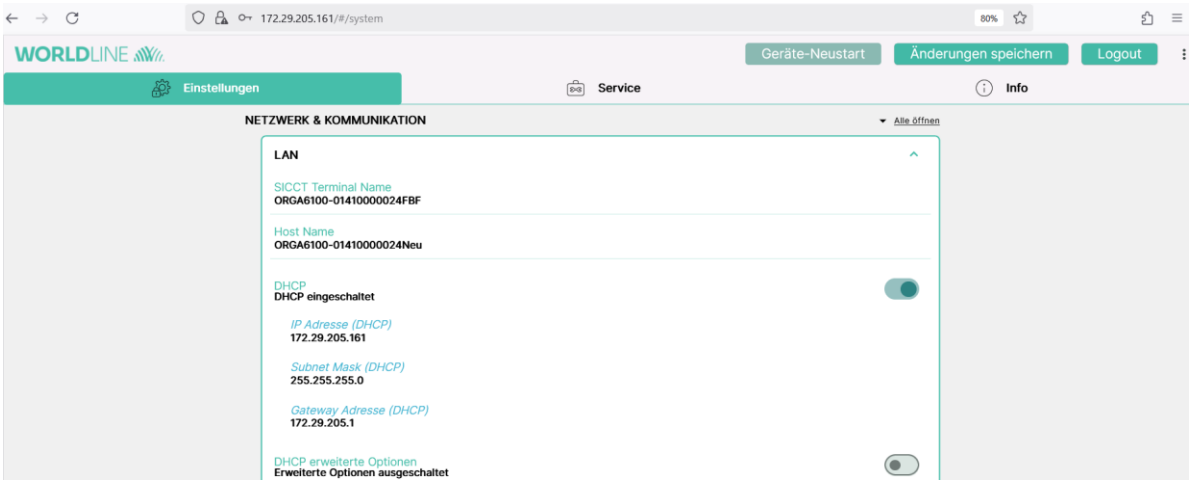
6. **Logout-Button**
   Mit dem Logout-Button loggen Sie sich aus der Web-Applikation wieder aus und gelangen automatisch wieder zur Eingabemaske

7. **Dreipunkt-Menü**
   Hier können Sie die Sitzungszeit in Minuten einstellen oder den Timer ganz unterbinden. Stellen Sie einen Timer ein, werden Sie nach Ablauf mit einem PopUp-Fenster gefragt, ob Sie die **Sitzung fortführen** möchten. Bestätigen Sie dieses PopUp-Fenster 30 min. lang nicht mit Ok, dann wird die Session automatisch beendet. Änderungen, die nicht gespeichert wurden, gehen verloren. Einzig in der Einstellung ∞ **/ unendlich** wird gar kein Timer gesetzt und die Session bleibt so lange offen, bis diese durch Bestätigung von **Sitzung beenden** oder Schließen des Browsers durch den Anwender geschlossen wird.

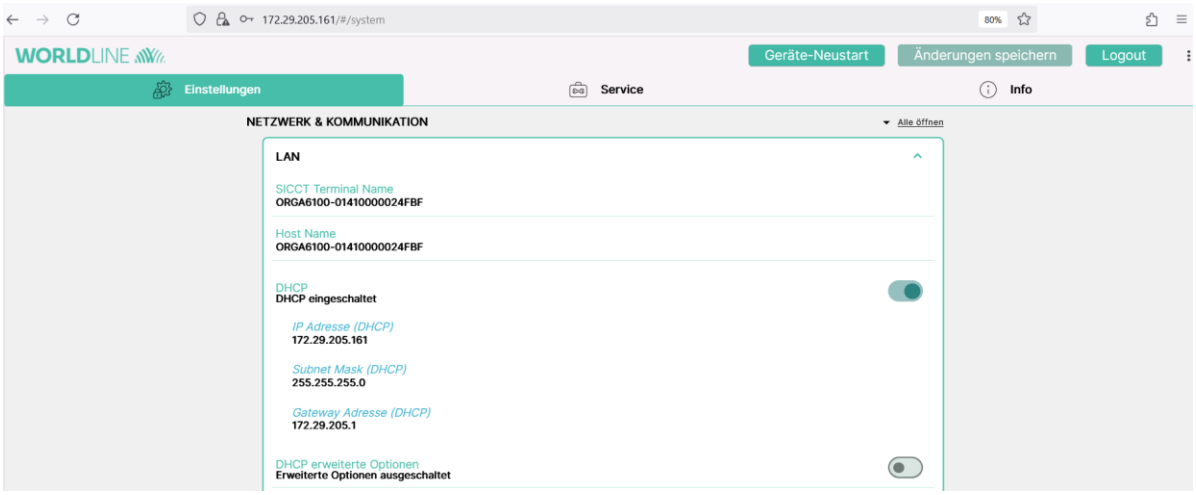
5.5.2 Grundsätzliche Handhabung

Die Themenbereiche sind sinnvoll gruppiert und befinden sich zur besseren Übersicht in einzelnen Ausklapp-Fenstern. Wählen Sie beispielhaft den Bereich **Einstellungen** und hier den Themenbereich **LAN** aus. Hier finden Sie folgenden grundsätzlichen Aufbau:

- **Schwarze Überschriften** öffnen und schließen das angeklickte Fenster. Zum Ausklappen des jeweiligen Themenbereiches reicht ein Klick in das jeweilige Feld, um die darin hinterlegten Parametern zur Anzeige zu bringen.
- **Grüne Überschriften** zeigen an, dass man diesen Wert ändern kann. Zum Ändern eines Parameters klicken Sie einfach auf seinen Namen oder Wert. Es erscheint eine Eingabe- und Änderungsmaske. Änderungen müssen mit **OK** bestätigt werden, das Schließen des Fensters kann über den Button **Cancel** erfolgen.
- **Türkisblaue Überschriften** zeigen an, dass man diese Felder nicht ändern kann. Zum Kopieren können Sie aber auf deren Wert klicken und bekommen hier auch eine Bestätigung, dass Sie den Werteinhalt in die Zwischenablage kopiert haben.
- **Ausgegraute Überschriften** sind temporär nicht aktiv, da diese - z.B. im VPN-Fenster - gerade nicht zur ausgewählten Methode zählen. Dadurch wird sichergestellt, dass immer nur die Felder gefüllt werden können, die bei der jeweiligen VPN-Methode auch benötigt werden.
- Außerdem finden Sie an verschiedenen Stellen **Schiebeschalter** , die Funktionen ein- bzw. ausschalten oder zusätzliche Konfigurationsparameter zur Ansicht bringen.
- Wurde eine Änderung vorgenommen und mit OK bestätigt, aktiviert sich der „**Änderungen speichern“**-Button und man kann diesen Wert mit einem Klick auf den Button in das Kartenterminal übernehmen. Sie können so viele Änderungen vornehmen, wie Sie möchten, müssen aber mindestens zum Schluss einmal den „**Änderungen speichern“**-Button ausführen, um die Änderungen in die Konfigurationstabelle des Gerätes zu übernehmen und diese damit im Kartenterminal zu speichern:



➢ Ist darüber hinaus für die Aktivierung dieser Änderung ein Neustart erforderlich, wir jetzt automatisch auch der **Geräte-Neustart**-Button aktiviert:



Ein **Geräte-Neustart** kann zusätzlich immer auch über den Reiter **Service** gestartet werden. Nach einem Geräte-Neustart wird empfohlen, den Browserinhalt einmal zu aktualisieren (F5-Taste), um sicher-zustellen, dass der angezeigte Inhalt auch dem neusten Stand entspricht.

6. Referenzen

6.1 Spezifikationen

Nr.	Titel
[RFC_791]	INTERNET PROTOCOL https://www.rfc-editor.org/rfc/rfc791
[RFC_793]	Transmission Control Protocol https://www.rfc-editor.org/rfc/rfc793
[RFC_4627]	The application/json Media Type for JavaScript Object Notation (JSON) https://www.rfc-editor.org/rfc/rfc4627
[RFC_4122]	A Universally Unique IDentifier (UUID) URN Namespace https://www.rfc-editor.org/rfc/rfc4122
[RFC_5246]	The Transport Layer Security (TLS) Protocol Version 1.2 https://www.rfc-editor.org/rfc/rfc5246
[RFC_6455]	The WebSocket Protocol https://www.rfc-editor.org/rfc/rfc6455
[RFC_6066]	Transport Layer Security (TLS) Extensions: Extension Definitions https://www.rfc-editor.org/rfc/rfc6066
[RFC_7230]	Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing https://www.rfc-editor.org/rfc/rfc7230
[gemProdT_KT]	Produkttypsteckbrief Prüfvorschrift eHealth-Kartenterminal, Produkttyp Version: 1.8.0-0, Version 1.0.0, 17.05.2022

Tabelle 2: Spezifikationen