Unabhängige Treuhandstelle
UNIVERSITÄTSMEDIZIN GREIFSWALD

THS Schnittstellen-Spezifikation [Allgemeiner Teil]

Hinweis: Work in Progress!
Dieses Dokument stellt keinen finalen Stand dar und wird stetig ergänzt, Bestehendes wird möglichst beibehalten. Die Inhalte dieses Dokumentes und somit der Schnittstelle sind nicht mit einer vollständigen projektspezifischen Implementierung gleichzusetzen. Referenzimplementierungen oder verbindliche Annahmen sollten unbedingt in Abstimmung mit den Autoren dieses Dokumentes erfolgen.

1 Allgemeines

1.1 Ziel

Das Ziel dieses Dokumentes ist eine detaillierte Beschreibung des Konzeptes und der technischen Gegebenheiten der Schnittstelle der Treuhandstelle. Das Konzept stellt zunächst einen generischen Ansatz dar, der durch eine konkrete Implementation technisch umgesetzt werden kann. Die technische Implementation, basierend auf dem generischen Konzept, wird anschließend detailliert beschrieben.

1.2 Erläuterungen

Das Konzept der Schnittstelle ist an das ID-Management „Mainzelliste“ der Universität Mainz angelehnt. Durch die Mainzelliste wurden bereits grundlegende generische Sicherheitsmechanismen, Abläufe und Datenformate entwickelt. Diese können für jede Funktion der Treuhandstelle eingesetzt werden und bieten somit ein solides Grundkonzept der Schnittstelle.

1.3 Kompatibilität mit Mainzelliste

Das Kommunikationskonzept und dessen REST-Implementierung sind mit der Mainzelliste kompatibel. Da aber die Mainzelliste ein reines ID-Management ist und die THS der Universätsmedizin Greifswald weitere Funktionalitäten unterstützt, unterscheiden sich die beiden Schnittstellen im Funktionsumfang. Aber auch bei inhaltlich gleichen Funktionen sind teilweise zusätzliche Parameter notwendig.

2 Konzept

Im Folgenden wird ein allgemeines Konzept erläutert, welches den Ablauf der Kommunikation und die Funktionen der Treuhandstelle in einer abstrakten Form beschreibt.

2.1 Grundlegendes Konzept

Jede Komponente, die mit der Treuhandstelle kommuniziert, muss sich zunächst authentifizieren. Authentifizierte Komponenten sind durch zugeordnete Rollen (siehe Kapitel 2.2) zur Nutzung bestimmter Funktionalitäten autorisiert. Die Nutzung einer Funktionalität ist in einem in Kapitel 2.3 beschriebenen Ablauf integriert.

2.2 Rollen

Rollen definieren Aufgaben die Systemkomponenten / Software oder Anwendungsteile übernehmen können. Mit einer Rolle sind konkrete Kommunikationsabläufe verbunden, in denen Nachrichten für konkrete Funktionen wie z.B. „Patient anlegen“ kommuniziert werden. Folgende Rollen wurden definiert THS-Service, Receiver und Provider.

2.2.1 THS-Service

Diese Rolle bildet die Funktionalitäten z.B. „Patient anlegen“ einer Treuhandstelle ab. Die anderen Rollen können diese Funktionen des THS-Services nutzen.

2.2.2 Receiver

Diese Rolle übernimmt eine Komponente, die Daten vom THS-Service erhalten soll, z.B. ein Pseudonym oder Einwilligungsinformationen. Der Receiver initiiert, durch die Wahl der der Funktion, den gesamten Ablauf mit dem THS-Service und erhält nach dem Funktionsaufruf die angeforderten Daten.

2.2.3 Provider

Da der Receiver die Funktion der Treuhandstelle vorgibt, liefert der Provider dem THS-Service die Daten, die zum erfolgreichen Ausführen der Funktion benötigt werden.

Besonderheit

In der Regel werden Receiver und Provider nicht von der gleichen Komponente eingenommen. Die Pseudonyme des Receivers sind dem Provider nicht bekannt und umgekehrt. Der Vermittler zwischen diesen Rollen ist der THS-Service. Im Standardfall ist auch eine Kommunikation zwischen den Rollen Provider und Receiver implementiert bzw. diese Kommunikation ist der „Auslöser“ für die Interaktion mit dem THS-Service. Die Kommunikation zwischen Provider und Receiver (sekundäre Kommunikation) wird hier nicht näher erläutert sondern nur angedeutet, um einen beispielhaften Überblick zu geben.

2.3 Interaktion der Rollen (Kommunikationsabläufe)

Im Folgenden wird die Interaktion der Rollen näher erläutert. Es existiert ein allgemeingültiger Ablauf, der als Variante 1 definiert ist. Prinzipiell ist für jede Funktion der Treuhandstelle diese Variante möglich. Ist die Variante 1 nicht passend, ist eine Alternative Variante 2 beschrieben, die je nach Funktion ebenfalls möglich sein kann.

Begriffserläuterungen

Session → Eine Session ist ein Datenobjekt, welches Informationen zum Receiver beinhaltet. Sie dient als eindeutige Identifikation des Receivers. Ebenso beinhaltet eine Session die Funktionen, die der Receiver angefordert hat. Eine Session hat eine eindeutige ID und besitzt eine Gültigkeit.

Token → Ein Token ist eine Art Ticket und bildet eine Funktion der Treuhandstelle ab. Ein Token ist immer einer Session zugeordnet. Das Token wird durch den Receiver angefordert und durch den Provider eingelöst. Anhand der Session und des Tokens können Receiver und Provider zusammen im THS-Service eindeutig identifiziert und deren Daten verarbeitet werden. Ein Token hat eine eindeutige ID und besitzt eine temporäre Gültigkeit. Es kann, sofern es kein Token für eine Stapelverarbeitungsfunktionalität ist, nur einmal eingelöst werden.

Request → Ein Request ist eine Anfrage zur Datenübertragung zwischen Rollen.

Response → Ein Response ist eine Antwort einer Rolle auf einen Request.

Callback → Der Callback dient der Übermittlung von Daten an den Receiver. Der Receiver definiert eine Callback-URL, an die die Daten vom THS-Service gesendet werden.

Redirect → Der Redirect ist eine URL des Receivers, auf die der Provider vom THS-Service weitergeleitet werden kann, wenn das Token vom Provider eingelöst wurde.

2.3.1 Variante 1 (KV1)

In folgender Abbildung 1 ist eine vereinfachte Darstellung der Variante 1 erläutert am Beispiel „Senden von MDATs von Provider an Receiver mit Austausch des PSN über die Treuhandstelle“. Diese Variante definiert die Kommunikation mit 3 Rollen. Die folgenden Abbildungen beschränken sich zunächst nur auf die Kommunikation, die auch in der Spezifikation erläutert wird. Dies ist die primäre Kommunikation. Weitere Kommunikation zwischen Receiver und Provider (z.B. der Austausch der Token-ID) wird als sekundäre Kommunikation bezeichnet.

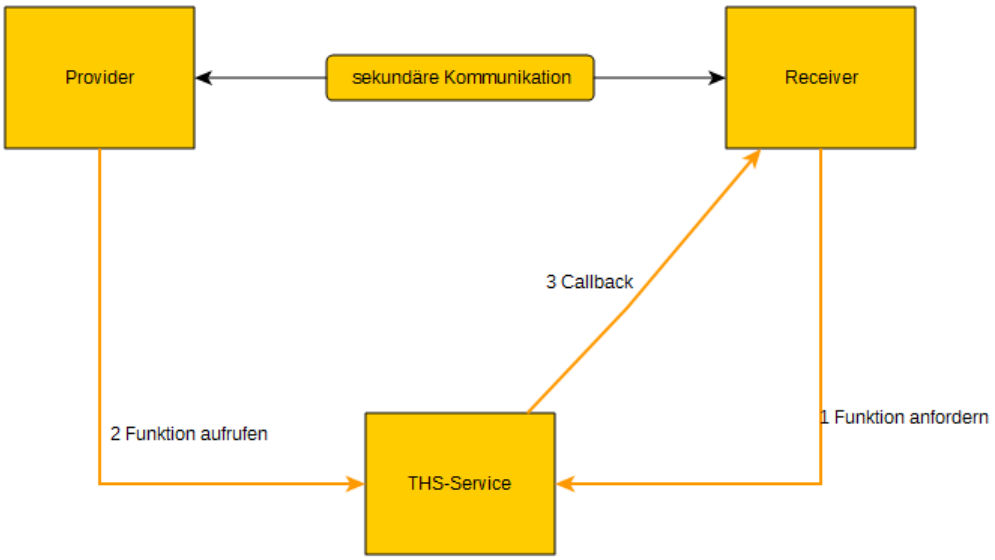


Abbildung 1: vereinfachte primäre Kommunikation zwischen den Rollen in Variante 1

Konventionen für nachfolgende Tabellen und Diagramme

Im Folgenden werden Funktionen und Nachrichten näher beschrieben. Für eine einheitliche und standardisierte Bezeichnung wurden folgende Konventionen festgelegt:

- Funktionen beginnen mit „F“, gefolgt von einem „_“ und dem Kürzel der Funktion, z.B. „F_AP“ für die Funktion „addPatient“.
- Die allgemeingültigen und funktionsunspezifischen Nachrichten im generischen Konzept beginnen mit einem „N“ gefolgt von einem „_“ und dem Kürzel, z.B. „N_TC“ für Nachricht „Token-Call“.

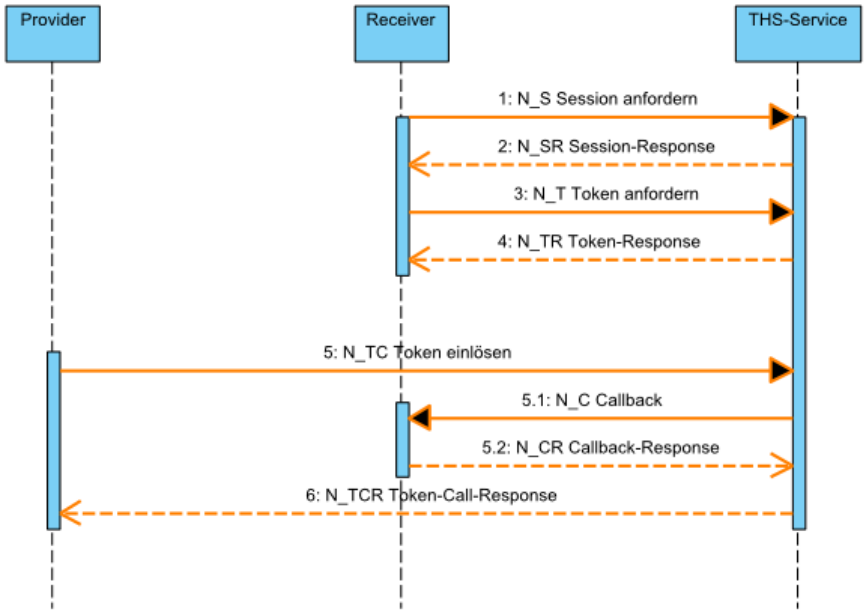


Abbildung 2 zeigt den Ablauf in einem Sequenzdiagramm und folgende Tabelle beschreibt die Nachrichten der primären Kommunikation im Detail.

Schritt	Beschreibung	Nachricht
1	Receiver initiiert eine Session beim THS-Service und übergibt Parameter	N_S (Session)
2	THS-Service gibt die SessionID an den Receiver zurück (Response)	N_SR (Session-Response)
3	Receiver initiiert Token beim THS-Service übergibt die Funktion und funktionsspezifische Parameter	N_T (Token)
4	THS-Service gibt TokenID und weitere Informationen des Tokens an den	N_TR (Token-Response)

	Receiver zurück (Response)	
5	Provider löst das Token ein und übergibt die funktionsspezifischen Daten an den THS-Service	N_TC (Token-Call)
5.1	THS-Service übermittelt die angeforderten Daten an den Receiver zurück (Callback)	N_C (Callback)
5.2	Receiver bestätigt den erfolgreichen Erhalt der Daten (Callback-Response)	N_CR (Callback-Response)
6.a	THS-Service gibt funktionsspezifische Daten an den Provider zurück (Response)	N_TCR (Token-Call-Response)
6.b	THS-Service gibt funktionsspezifische Daten an den Provider zurück (Redirect)	N_R (Redirect)

Die Nachricht 6 unterscheidet 2 Typen. Je nachdem, ob der Provider z.B. ein Browser ist und eine User-Interaktion benötigt (6.b) oder der Provider ein System ist, welches automatisiert die Daten überträgt (6.a). Es wird nur ein Typ je Aufruf verwendet. Nähere Erläuterungen hierzu sind in Kapitel 2.5.2 zu finden.

Variante 1 inklusive sekundärer Kommunikation

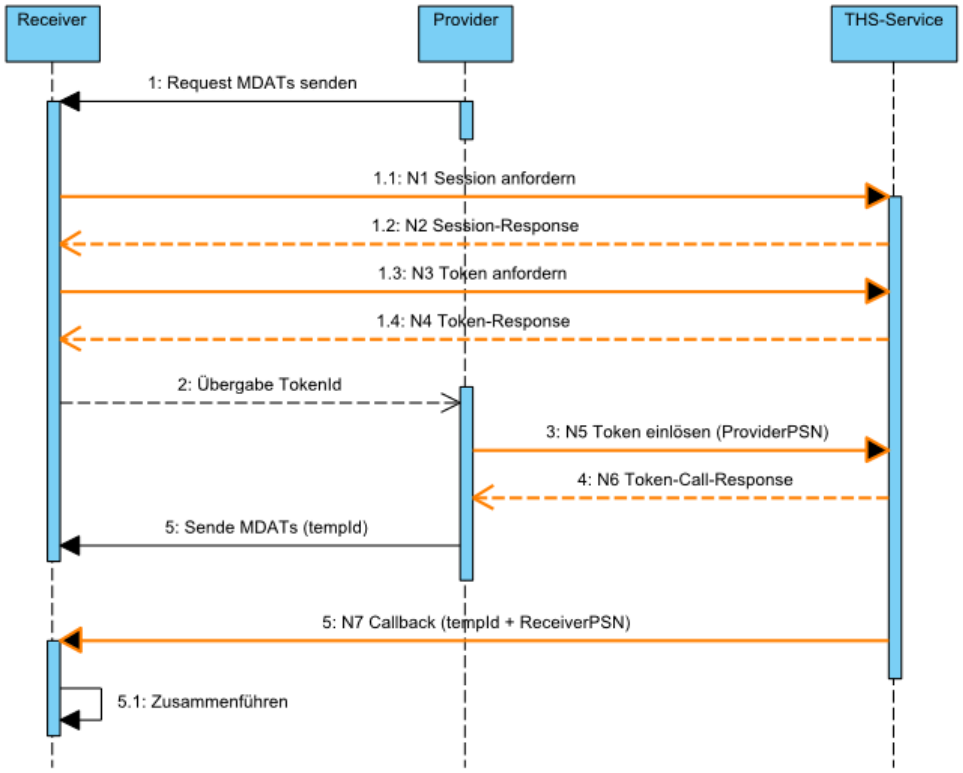


Abbildung 3: Erweitertes Sequenzdiagramm der Variante 1 mit sekundärer Kommunikation zwischen Receiver und Provider. Nur die farbigen Kommunikationsschritte werden in dieser Spezifikation beschrieben.

Fehlerverhalten

Aus Sicht des THS-Services folgt auf einem Request immer ein Response. Tritt im THS-Service ein Fehler auf, wird ein entsprechender Fehlercode als Response zurückgegeben. Eine Ausnahme bildet der Aufruf des Callbacks. Löst der Provider ein Token ein, wird dieses im THS-Service bearbeitet und der Callback wird aufgerufen. Es können bei der Abarbeitung eines Tokens folgende Ereignisse eintreten:

1. Token kann nicht verarbeitet werden,

2. Callback-URL ist nicht erreichbar oder im Callback-Response wurde ein Fehler zurückgegeben.

In beiden Fällen wird im Token-Call-Response an den Provider ein Fehlercode zurückgegeben. Der Receiver würde in keinem der Fälle einen Fehler erhalten, aber auch keine gültigen Daten. In der Regel müsste nun der Provider dem Receiver den Abbruch des Vorgangs mitteilen.

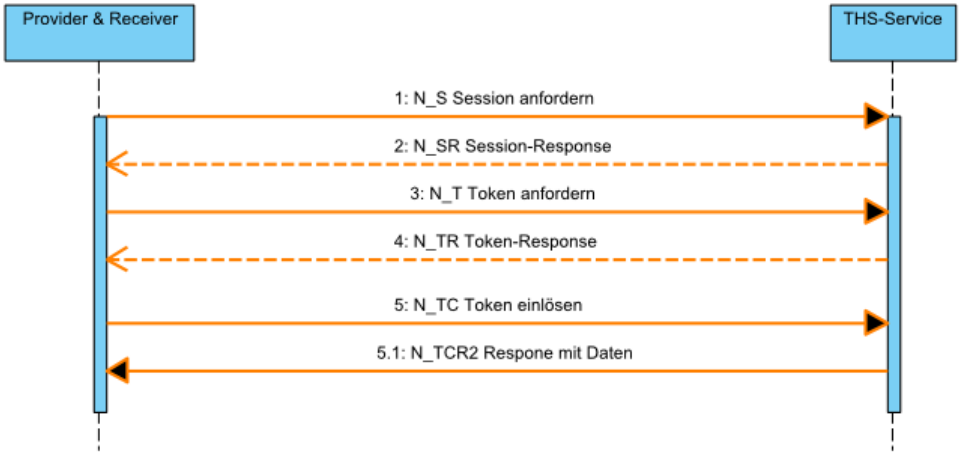
Tritt einer der beschriebenen Fehler auf, wird das Token ungültig und kann nicht wiederverwendet werden.

2.3.2 Alternative Variante 2 „Eine Komponente ist gleichzeitig Provider und Receiver“ (KV2)

Beschreibung

In einigen Anwendungsfällen kommuniziert nur eine Komponente mit dem THS-Service, z.B. bei dem Abfragen von Einwilligungen. In diesem Fall ist die Variante 1 und die Übermittlung der Ergebnisse per Callback nicht zwingend notwendig. Für die einfachere Umsetzung wurde Variante 2 konzipiert. Ist der Receiver auch gleichzeitig Provider kann das Ergebnis der Funktion direkt an den Provider im Response übermittelt werden.

Ablauf



Folgende Abbildung 4 zeigt den Ablauf in einem Sequenzdiagramm.

Schritt	Beschreibung	Nachricht
1	Receiver initiiert eine Session beim THS-Service und übergibt Parameter	N_S (Session)
2	THS-Service gibt die SessionID an den Receiver zurück (Response)	N_SR (Session-Response)
3	Receiver initiiert Token beim THS-Service übergibt die Funktion und funktionsspezifische Parameter	N_T (Token)
4	THS-Service gibt TokenID und weitere Informationen des Tokens an den Receiver zurück (Response)	N_TR (Token-Response)
5	Provider löst das Token ein und übergibt die funktionsspezifischen Daten an den THS-Service	N_TC (Token-Call)
5.1	THS-Service übermittelt die angeforderten Daten an den Provider (und Receiver) zurück	N_TCR2 (Token-Call-Response Variante 2)

Fehlerverhalten

Tritt ein Fehler auf, dann wird ein entsprechender Fehlercode im Response zurückgegeben. Da hierbei weder Callback- noch Redirect-URLs durch den THS-Service genutzt werden, entspricht das Fehlerverhalten der Standardprozedur.

2.3.3 Variante 3 „THS-Service ist gleichzeitig auch Provider“ (KV3)

Für einige Anwendungsfälle ist es notwendig, dass der THS-Service selbst als Provider agiert, weil die THS Informationen z.B. durch andere Systeme erhalten hat und weiter kommunizieren muss. Ein Beispiel wäre die Änderung eines Consents oder gar ein kompletter Widerruf. Hierfür muss der Receiver beim THS-Service registriert/konfiguriert sein und bekommt anschließend die notwendigen Nachrichten (vom Provider) automatisch.

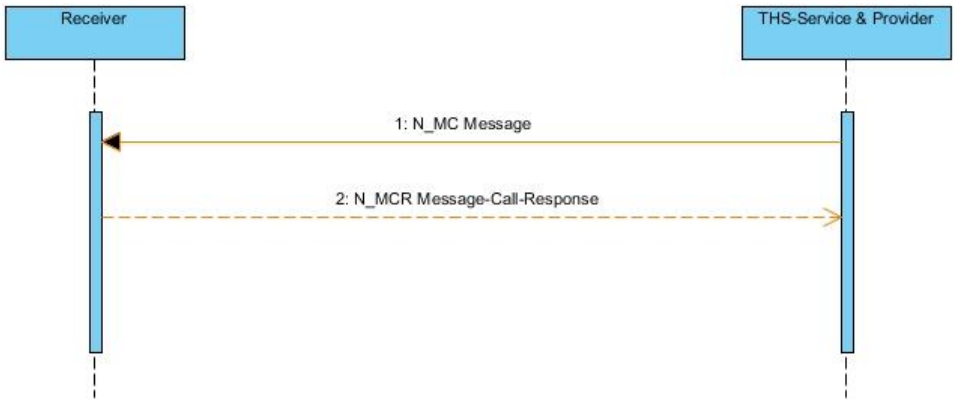


Abbildung 5: Sequenzdiagramm für Variante 3

Schritt	Beschreibung	Nachricht
1	THS-Service schickt eine Nachricht an den Provider.	N_MC (Message-Call)
2	Reciever quittiert die erhaltene Nachricht mit einem Status oder ErrorCode.	N_MCR (Message-Call-Response)

2.4 Notification Service

Die THS muss in bestimmten Anwendungsfällen Informationen an Receiver mitteilen, die sie durch anderen Systeme oder auch Personen erhalten und verarbeitet hat. Hierzu bietet die THS einen Notification Service an, dessen Funktionsweise im folgenden näher beschrieben wird.

2.4.1 Receiver Registrierung

Es ist vorgesehen, dass jeder Receiver, der über den Notification Service Nachrichten erhalten soll, beim THS-Service registriert sein muss. Diese Registrierung erfolgt nicht automatisiert. Die Receiver werden manuell konfiguriert, um zum einen die Vertrauenswürdigkeit sicherzustellen und zum anderen die Implementierung seitens des Receivers auf ein Minimum zu reduzieren.

Folgende Informationen vom Receiver werden für die Registrierung benötigt bzw. konfiguriert:

- eindeutige ID
- Empfangs-URL des Receivers + Aufruf-Methode (z.B. HTTP-POST/PUT)
- Nachrichtenformat
- Nachrichtentypen (siehe Messages in Kapitel 2.5)
- Patienten-Identifier-Typ

2.4.2 Kommunikation

Der Notification Service verwendet den Kommunikationsablauf Variante 3.