Beim vorliegenden Dokument handelt es sich um einen Entwurf der gematik in Vorbereitung auf zukünftige normative Festlegungen als Grundlage entsprechender Zulassungs- und Bestätigungsverfahren. Die gematik veröffentlicht diesen Entwurf mit dem Ziel, dass sich Interessierte bereits frühzeitig einen Überblick über die mögliche Weiterentwicklung der Telematikinfrastruktur verschaffen können. Die gematik übernimmt keine Gewähr für die Aktualität, Richtigkeit und Vollständigkeit dieses Entwurfes und behält sich das Recht vor, ohne vorherige Ankündigung Änderungen oder Ergänzungen vorzunehmen oder von den Regelungen insgesamt bzw. teilweise Abstand zu nehmen.

Änderungsbedarf:

Der Basisdienst TBAuth fordert im Passive Requestor Profile bei den Operationen signin und signout eine Bestätigung des Nutzers, damit für ihn ersichtlich ist, dass gerade eine Identitätsbestätigung ausgestellt wird. Hintergrund ist, dass jeder Server, zu dem eine Verbindung per Browser aufgebaut wird, eine beliebige Identitätsbestätigung erzeugen lassen kann, ohne, dass der Nutzer etwas davon bemerkt.

Diese zusätzliche Bestätigung behindert jedoch die Abläufe des nutzenden Projekts und soll entfallen.

Stattdessen wird nun darauf hingewiesen, dass der aufrufende Dienst dafür sorgen muss, dass für den Dienst erkennbar ist, dass eine Identitätsbestätigung im Rahmen seiner Session ausgestellt wurde.

Folgende Konzepte und Spezifkationen werden dahingehend angepasst:

gemKPT_Arch_TIP

gemKPT_Arch_TIP_Anhang_E_Anwendungsfälle

gemSpec_Kon_TBAuth

Änderungen in gemKPT_Arch_TIP

5.5.1.4.2 I_IDP_Auth_Passive_Client (Provided)

⌦ **TIP1-A_6776 Schnittstelle I_IDP_Auth_Passive_Client**

Die Schnittstelle I_IDP_Auth_Passive_Client MUSS alle zugehörigen logischen Operationen implementieren.⌫

⌦ **TIP1-A_6040 Logische Operation I_IDP_Auth_Passive_Client::signin**

Die Schnittstelle I_IDP_Auth_Passive_Client MUSS die logische Operation signin implementieren.

Tabelle 1: Operation signin

I_IDP_Auth_Passive_Client					Berechtigung: CS
signin	Parameter				V, I, A
	In	wfresh	TimeInformation	IM101	M/H/H
	In	wct	TimeInformation	IM101	M/H/H

	In	wtrealm	Identifier	IM101	M/H/H
	In	wreply	URI	IM304	M/H/H
	In	wctx	SessionContext	IM101	M/H/H
	Out	wresult	IdentityAssertion	IM426	M/M/M
	Out	wctx	SessionContext	IM101	M/H/H

Diese Schnittstelle ermöglicht das Ausstellen von Identitätsbestätigungen (*wresult*) basierend auf WS-Trust, SAML2.0 Assertions, und WS Federation 1.2 für Passive Requestor Profile .

Zur Sicherstellung einer zeitlichen Konsistenz übergibt der Aufrufer seine aktuelle Zeit (*wct*). Sollte diese von der lokalen Zeit über mehr als eine Minute abweichen, wird die Ausstellung einer Identitätsbestätigung verweigert.

Die Festlegung der Gültigkeitsdauer (*wfresh*) und die Bindung an den aufrufenden Dienst (*wtrealm*) werden durch den Anwender festgelegt und in die Identitätsbestätigung integriert. Sollte keine Gültigkeitsdauer übergeben werden, so wird eine Gültigkeit von 3 Stunden gesetzt.

Um die durch diese Operation erstellten Identitätsbestätigungen von denen eines lokalen IDPs unterscheiden zu können, tragen die Identitätsbestätigungen im Feld Aussteller (Issuer) die Bezeichnung „IDP der TI-Plattform“.

Die in einer Identitätsbestätigung enthaltenen Identitätsinformationen werden ausschließlich aus dem Zertifikat der kryptografischen Identität ID.HCI.OSIG der im Browser konfigurierten SMC-B übernommen und mit derselben Identität signiert.
Sollte keine SMC-B im Browser konfiguriert sein, fragt die Operation ab, welche SMC-B verwendet werden soll und speichert die Auswahl im Browser für nachfolgende Aufrufe.

Die Gültigkeit von Identitätsbestätigungen ist auf den anfragenden Dienst beschränkt (*wtrealm*). Somit wird verhindert, dass sich Dienste Identitätsbestätigungen ausstellen lassen können, die zum Zugriff auf andere Dienste berechtigen. Damit liegt es in der Verantwortung des anfragenden Dienstes, den Parameter (*wtrealm*) so zu wählen, dass er eindeutig erkennen kann, dass eine Identitätsbestätigung durch ihn erstellt wurde.

Die Operation legt einen Session-Cookie für die erzeugte Identitäts-Session im Browser an.

Nach Abschluss der Operation wird ein HTTP-Redirect auf *wreply* ausgeführt und dabei die erzeugte Identitätsbestätigung (*wresult*) und der gelieferte SessionContext (*wctx*) übergeben.

Verfügbarkeit: NA, Nichtabstreitbarkeit: M

⌫

⌦ TIP1-A_6041 Logische Operation I_IDP_Auth_Passive_Client::signout

Die Schnittstelle I_IDP_Auth_Passive_Client MUSS die logische Operation signout implementieren.

Tabelle 2: Operation signout

I_IDP_Auth_Passive_Client					Berechtigung: CS
signout	Parameter				V, I, A
	In	wreply	URI	IM304	M/H/H
	In	SessionId	Session	IM410	M/H/H

Diese Schnittstelle ermöglicht das Abmelden basierend auf WS-Trust, SAML2.0 Assertions, und WS Federation 1.2 für Passive Requestor Profile. Mit Aufruf dieser Operation loggen sich Anwender ~~nach einer separat abgefragten Bestätigung~~ aus und die Identitäts-Session (*SessionId*) und der zugehörige Session-Cookie im Browser werden gelöscht.
Nach Abschluss der Operation wird ein HTTP-Redirect auf *wreply* ausgeführt.

Verfügbarkeit: NA, Nichtabstreitbarkeit: NA

⌫

Änderungen in gemKPT_Arch_TIP_Anhang_E_Anwendungsfälle

14.5 TIP_TBAUTH_UC_005 Abmeldung vom Dienst

Tabelle 3: TIP_TBAUTH_UC_005 Abmeldung vom Dienst

Szenario:	
Beteiligte Akteure	Dienst, Webbrowser, Konnektor
Beschreibung	Der Anwender meldet sich vom Dienst ab und seine Identitätsbestätigung wird aus dem Browser-Cache gelöscht.
Pre-Condition	Im Rahmen von TIP_TBAUTH_UC_004 hat sich der Anwender an einem Dienst angemeldet und der Konnektor hat eine Identitätsbestätigung ausgestellt, die zum Durchführungszeitpunkt noch aktuell ist.
Ablauf	1. Der Anwender meldet sich vom Dienst ab, der diesen an den lokalen Konnektor weiterleitet. 2. Der Webbrowser ruft am Konnektor die Operation I_IDP_Auth_Passive_Client:: signout auf. ~~3. Der Konnektor lässt sich die Abmeldung vom Anwender bestätigen.~~ 4. Der Konnektor löscht den zur Sitzung zugehörigen Sitzungs-Cookie im Webbrowser.
Alternativen	
Post-Condition	• Der im Webbrowsers gespeicherte Sitzungs-Cookie wurde gelöscht. • Der Konnektor hat temporär gespeicherte Informationen über die Sitzung verworfen.

Änderungen in gemSpec_Kon_TBAuth

4.2.1 Operation signIn

~~⌦ **TIP1-A_7025 Bestätigung des SignIn**~~

~~Der Basisdienst TBAuth MUSS beim Aufruf der Operation signIn eine Bestätigung des Anmeldevorgangs durch den Benutzer abfragen. ⌫~~

[…]

4.2.2 Operation signOut

[…]

~~⌦ **TIP1-A_6861 Operation signOut: Bestätigung durch Benutzer**~~

~~Der Basisdienst TBAuth MUSS an der Schnittstelle I_IDP_Auth_Passive_Client bei Aufruf der Operation signOut im Webbrowser eine Bestätigung durch den Benutzer abfragen, ob dieser sich abmelden will. Bei Abbruch zeigt der BD-TBAuth dem Benutzer einen entsprechenden Hinweis an. ⌫~~

Sitzungsinformationen könnten die Identitätsbestätigung oder eine Referenz auf diese enthalten, sind aber letzten Endes implementierungsabhängig. Je nach Implementierung des Basisdienstes, könnte dieser Sitzungsinformationen temporär speichern oder auch in einem Session-Cookie im Webbrowser des Nutzers ablegen.

⌦ **TIP1-A_6862 Annullierung ~~nach Bestätigung~~ durch signOut**

Der Basisdienst TBAuth MUSS an der Schnittstelle I_IDP_Auth_Passive_Client bei Aufruf der Operation signOut ~~und nach Bestätigung durch den Nutzer~~ die zugehörige Identitätsbestätigung annullieren, indem es ggf. zugehörige Sitzungs-Cookies im Webbrowser des Benutzers löscht, ggf. temporär gespeicherte Sitzungsinformationen innerhalb des Basisdienstes löscht und für diesen Nutzer zuvor über I_IDP_Auth_Passive_Client ausgestellte Identitätsbestätigungen (über I_IDP_Auth_Active_Client::renew_Identity_Assertion) nicht mehr erneuert. ⌫

~~⌦ **TIP1-A_6863 Schutz vor CSRF**~~

~~Der Basisdienst TBAuth MUSS an der Schnittstelle I_IDP_Auth_Passive_Client sicherstellen, dass es Identitätsbestätigungen nur nach Bestätigung des Benutzers annulliert und hierbei Cross-Site Request Forgery (CSRF)-Angriffe verhindert werden. ⌫~~

[…]