Security-Referenzimplementierung des VWS vernetzt-Testbeds

V0.10

Technical Report

7. Mai 2021

Universität Magdeburg
Fakultät für Elektrotechnik und Informationstechnik
Institut für Automatisierungstechnik
Postfach 4120, D-39016 Magdeburg
Germany

Table of content

1 Abkürzungen

AAS	Asset Administration Shell
CP(P)S	Cyber-physikalisches (Produktions-)System
I4.0	Industrie 4.0
Id	Identifikator, Identifier
(I)IoT	(Industrial) Internet of Things
ISO	International Organization for Standardization
OSI-Modell	Open Systems Interconnection model
VWS	Verwaltungsschale

2 Relevante Dokumente

[1]Plattform Industrie 4.0: Diskussionspapier: Sicherer Downloadservice. 25.09.2020.

[2] VDI 2193 Blatt 1: Sprache für I4.0-Komponenten - Struktur von Nachrichten

[3] FIPA ACL Message Structure Specification (http://www.fipa.org/specs/fipa00061/SC00061G.html)

[4] AASiD Part 1: The exchange of information between partners in the value chain of Industrie 4.0 (Version 2.0.1)

[5] AASiD Part 2: Interoperability at Runtime – Exchanging Information via Application Programming Interfaces (V1.0RC01 for Review)

3 Ziel

Das Dokument beschreibt die Nutzung der Security-Referenzimplementierung des VWS vernetzt-Testbeds. Diese dient der praktischen Erprobung der Security-Konzepte für VWS gemäß des Diskussionspapiers "Sicherer Downloadservice" der Plattform Industrie 4.0 [1]. Während das Diskussionspapier das Szenario eines Datei-Downloads zur Veranschaulichung verwendet, wurden die Konzepte im Rahmen der hier beschriebenen Referenzimplementierung auf die Interaktion von VWS des Typs 3 mittels I4.0-Sprache übertragen. Dazu wurde ein einfaches semantisches Protokoll genutzt (*PingPong*), nach welchem VWS nach erfolgter Authentifizierung miteinander interagieren. Die Festlegungen des Diskussionspapiers [1] können so vollständig und ohne unnötig hohen Entwicklungsaufwand für das veranschaulichende Szenario getestet werden. Das *PingPong*-Protokoll kann durch beliebig komplexe und praktisch relevantere semantische Protokolle ersetzt werden. Die Referenzimplementierung ist eine VWS gemäß der Protokollrolle *pong*.

Die Abbildung 1 zeigt das PingPong-Szenario bei zunächst fehlender Authentifizierung. Das gesamte Szenario umfasst über das reine *PingPong*-Protokoll hinaus zusätzlich die Interaktion, um die Authentifizierung herbeizuführen. Dazu gehört ein weiterer Teilnehmer, der die Funktion des Authentifizierungs-Servers erfüllt. Diese zusätzlichen Aspekte beschreibt das Diskussionspapier "Sicherer Downloadservice" [1].

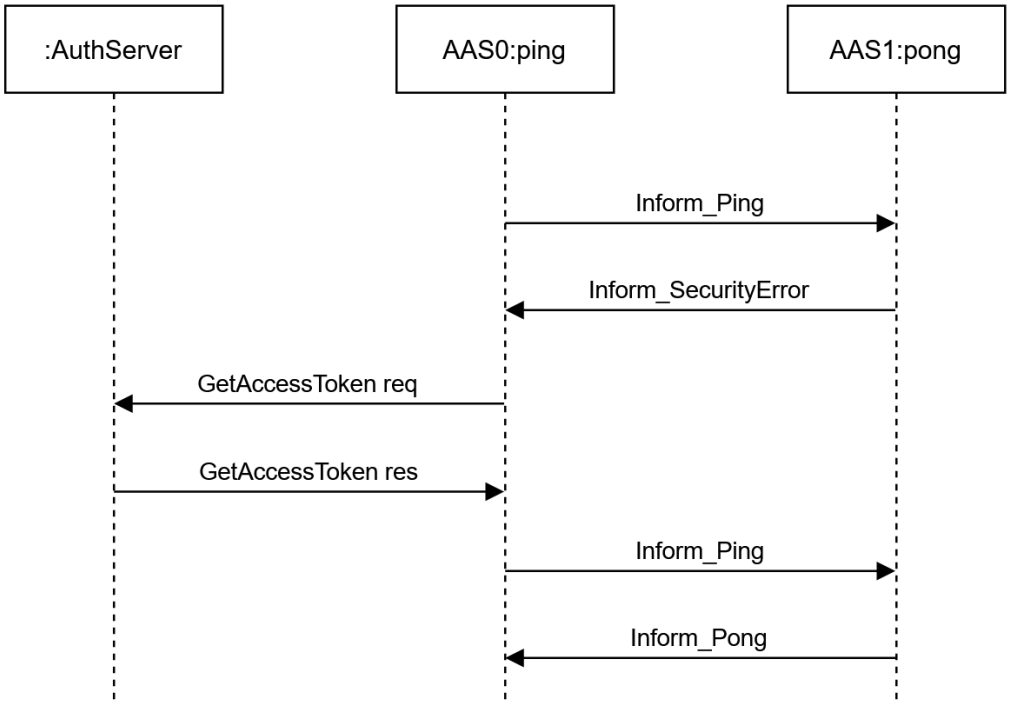

Abbildung 1: PingPong bei zunächst fehlender Authentifizierung

4 PingPong-Szenario

Das Szenario nutzt ein simples PingPong-Protokoll, um die Verwendung der Authentifizierung-Mechanismen zu veranschaulichen. Das semantische Protokoll *PingPong* ist definiert durch seine Nachrichtentypen sowie deren Verwendung in den protokollspezifischen Nachrichtensequenzen.

Es ist durch die Id *www.admin-shell.io/interaction/pingpong* identifizierbar. Beteiligte einer Interaktion gemäß dieses Protokolls sind einer der Rollen *ping* und *pong* zugeordnet. Die Referenzimplementierung ist eine VWS gemäß der Protokollrolle *pong*.

Die nachfolgenden Tabellen definieren in verkürzter Form die Interaktionselemente der jeweiligen Nachrichtentypen gemäß I4.0-Sprache. Zu jedem der Nachrichtentypen liegt diesem Dokument eine beispielhafte, in JSON serialisierte Nachricht gemäß I4.0-Sprache bei.

4.1 Nachrichtentypen

4.1.1 Inform_Ping

Zweck: Dient VWS dazu, eine andere VWS zur Antwort mit einer Nachricht vom Typ *Inform_Pong* aufzufordern.

Rolle des Senders: ping

Rolle des Empfängers: pong

Nachrichtentyp der erwarteten Antwort: Inform_Pong

Beispiel: Inform_Ping.json

Tabelle 1: Interaktionselemente des Nachrichtentyps „Inform_Ping“

Interaction element	Description	Value
Message	Content of the message	ping
AccessToken	Token received from the Authentifizierung server necessary to interact with an AAS of the role *pong*	<AccessToken>

4.1.2 Inform_Pong

Zweck: Dient VWS dazu, auf eine Nachricht einer anderen VWS vom Typ *Inform_Ping* zu reagieren.

Rolle des Senders: pong

Rolle des Empfängers: ping

Nachrichtentyp der erwarteten Antwort: keine

Beispiel: Inform_Pong.json

Tabelle 2: Interaktionselemente des Nachrichtentyps „Inform_Pong“

Interaction element	Description	Value
Message	Content of the message	pong

4.1.3 Inform_SecurityError

Zweck: Dient VWS dazu, auf eine nicht erfolgreiche Authentifizierung hinzuweisen.

Rolle des Senders: pong

Rolle des Empfängers: ping

Nachrichtentyp der erwarteten Antwort: keine

Beispiel: Inform_SecurityError.json

Tabelle 3: Interaktionselemente des Nachrichtentyps „Inform_SecurityError“

Interaction element	Description	Value
ErrorName	Short name of the error	InvalidToken or MissingToken
ErrorCode	Code of the error	E001 or E002
ErrorDescription	Description of the error	Access could not be granted because of invalid (E001) or missing (E002) AccessToken. Address the aasx-IdentityServer4 to get a valid one (GET https://admin-shell-io.com:50001/.well-known/openid-configuration).

4.2 Zustandsmaschine der Rolle Ping

Das Szenario kombiniert das PingPong-Protokoll mit dem Protokoll für die Authentifizierung. Daher ist für dieses spezifische Szenario eine zusätzliche Zustandsmaschine nötig, die die zugehörigen Protokollrollen, die die Interaktionspartner einnehmen können, im Sinne des Szenarios koordiniert (Abbildung 2, aus Sicht der Ping-Rolle).

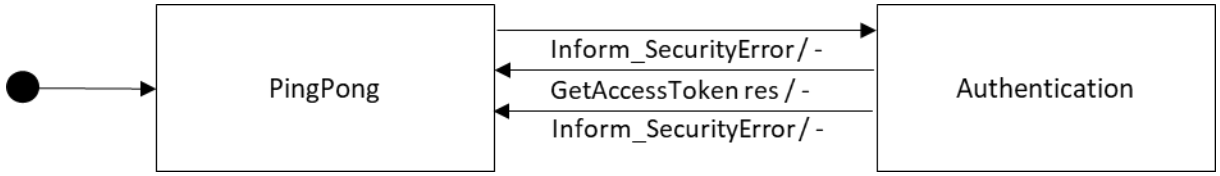

Abbildung 2: Koordinierende Zustandsmaschine

Abbildung 3 zeigt die vollständige Zustandsmaschine in grafischer Form, in der die enthaltenen Sub-Zustandsmaschinen aufgelöst sind, aus Sicht der Ping-Rolle. Tabelle 4 zeigt diese Zustandsmaschine in tabellarischer Form.

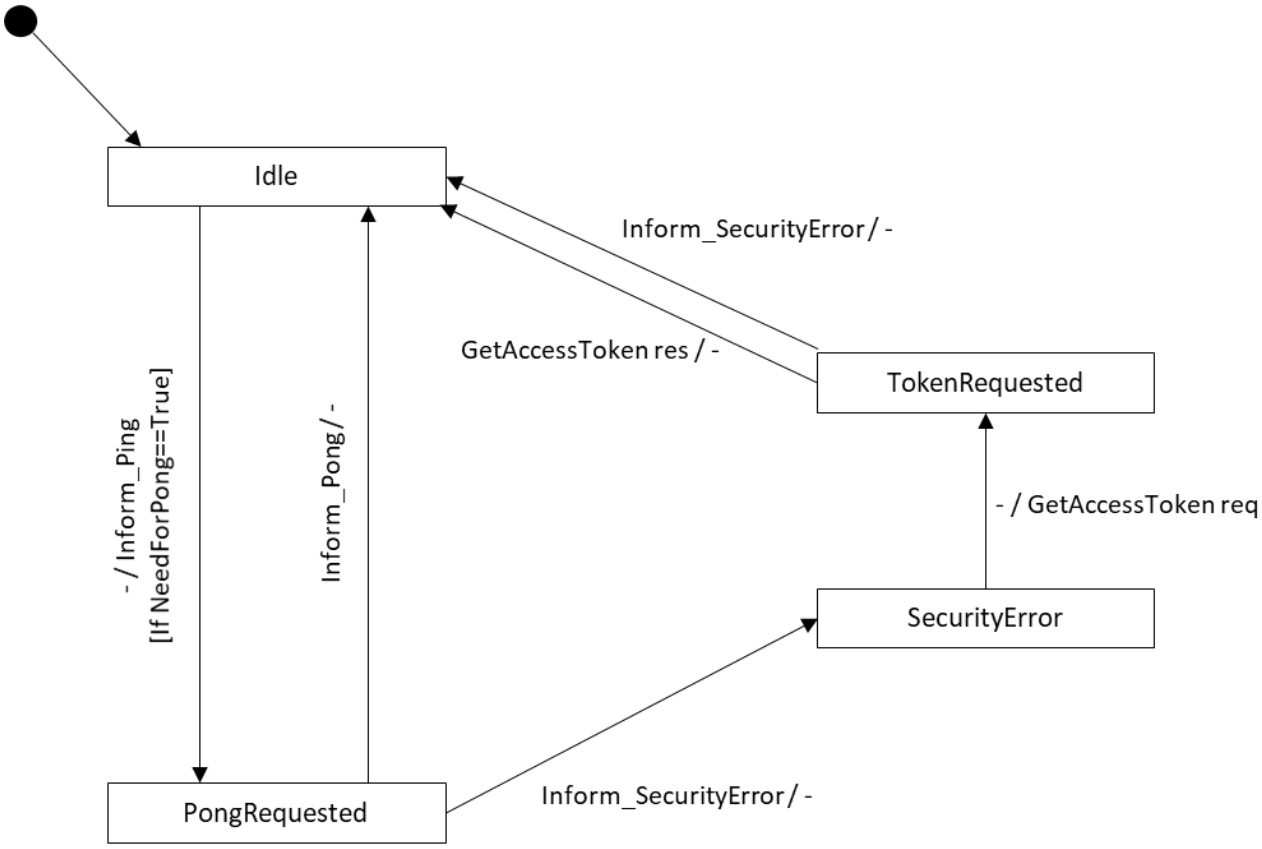

Abbildung 3: Zustandsmaschine mit aufgelösten Sub-Zustandsmaschinen aus Sicht der Ping-Rolle

Tabelle 4: Zustandsmaschine mit aufgelösten Sub-Zustandsmaschinen der Ping-Rolle

Source state	Destination state	Input	Condition	Output	Remarks
Idle	PongRequested	-	If NeedForPong==True	Inform_Ping	
PongRequested	Idle	Inform_Pong	-	-	
PongRequested	SecurityError	Inform_SecurityError	-	-	
SecurityError	TokenRequested	-	-	GetAccessToken req	
TokenRequested	Idle	Inform_SecurityError	-	-	
TokenRequested	Idle	GetAccessToken res	-	-	

4.3 Zustandsmaschine der Rolle Pong

Die Abbildung 4 zeigt die vollständige Zustandsmaschine in grafischer Form, in der die enthaltenen Sub-Zustandsmaschinen aufgelöst sind, aus Sicht der Ping-Rolle. Die Tabelle 5 zeigt diese Zustandsmaschine in tabellarischer Form.

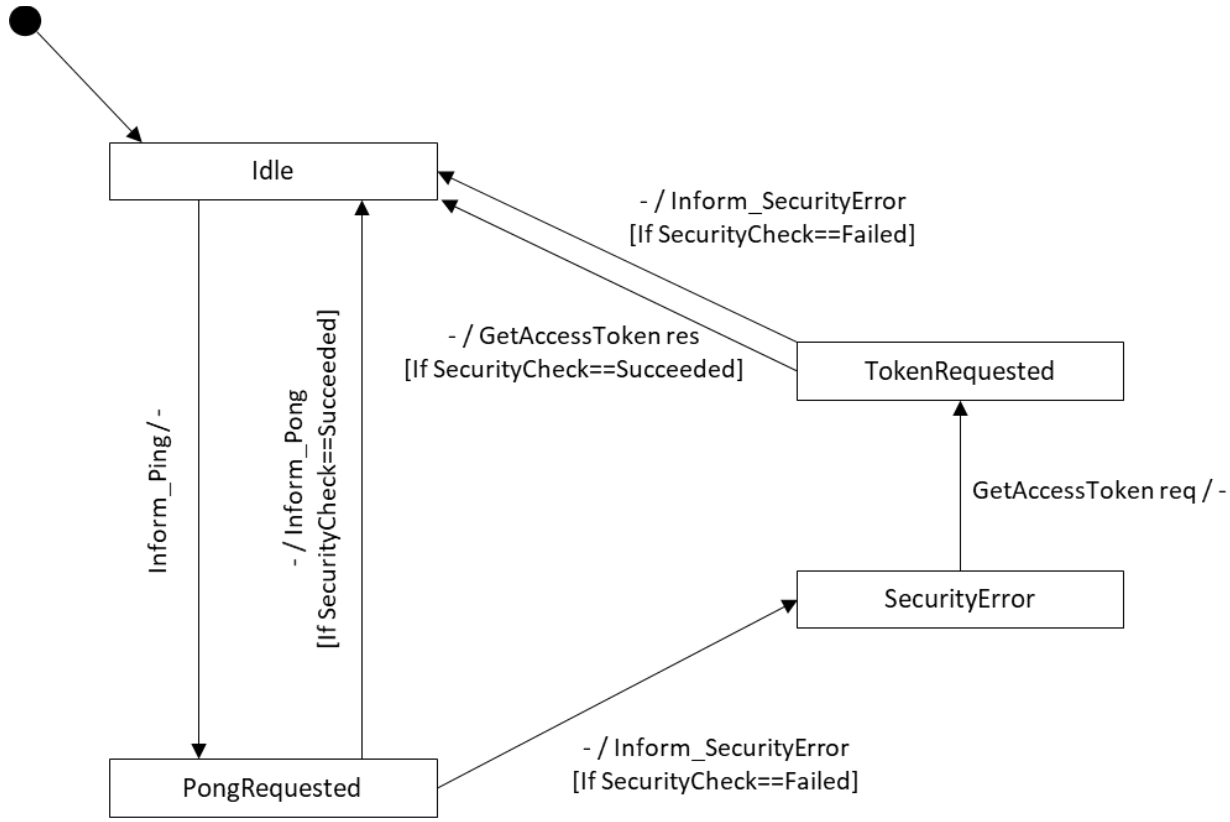

Abbildung 4: Zustandsmaschine mit aufgelösten Sub-Zustandsmaschinen aus Sicht der Pong-Rolle

Tabelle 5: Zustandsmaschine mit aufgelösten Sub-Zustandsmaschinen der Pong-Rolle

Source state	Destination state	Input	Condition	Output	Remarks
Idle	PongRequested	Inform_Ping	-	-	
PongRequested	Idle	-	If SecurityCheck==Succeeded	Inform_Pong	
PongRequested	SecurityError	-	If SecurityCheck==Failed	Inform_SecurityError	
SecurityError	TokenRequested	GetAccessToken req	-	-	
TokenRequested	Idle	-	If SecurityCheck==Succeeded	GetAccessToken res	
TokenRequested	Idle	-	If SecurityCheck==Failed	Inform_SecurityError	

5 Technische Details zur Nutzung der Referenzimplementierung

5.1 MQTT

Kommunikationsprotokoll: MQTT

Broker-Host: c222130.online-server.cloud

Broker-Port: 1883

Login-Name: industrie40

Passwort: %industrie:4.0

Downloadlink des CA-Zertifikats: https://www.dropbox.com/s/v8adyaf1jlw1rd2/ca.zip?dl=0

Subscribe-Topic der Referenzimplementierung: admin-shell.io/pong

Protokollrolle der Referenzimplementierung: *pong*

Test: Nutzung eines beliebigen MQTT-Clients und Senden modifizierter Beispielnachrichten an den Broker auf das Subscribe-Topic der Referenzimplementierung.

5.2 HTTP

Kommunikationsprotokoll: HTTP

Host: ovgu-pong.vws-vernetzt.de (TLS enabled)

Port: 8027

Protokollrolle der Referenzimplementierung: *pong*

Test: Nutzung eines beliebigen HTTP-Clients und Senden modifizierter Beispielnachrichten per POST-Request an die Registry-Referenzimplementierung.