Australian Government

Consumer Data Right

Data Holder Technical Guidance Material 4.0.2

Conformance Test Suite

Table of Contents

1 Document Control

Document Version	1.0
Document Status	Approved
Issued Date	03 November 2022
Owner	ACCC

1.1 Test Plan Revision History

Test Plan Version	CDS Standards Version	Issued Date	Description of Changes
4.0.0	1.16.0	15 August 2022	This document covers the guidance related to the Data Holder (DH) Test Plan (TP) version 4.0. The Test Plan has a number of changes from the previous DH TP 3.4.1 summarised below: • No new scenarios • This test plan has been built as a sector agnostic test plan to suit the needs of current and future sectors. Updated to conform with CDS standards 1.16.0 & FAPI 1.0 Phase 2. Rationalised scenarios and the following have been removed or merged: a. Discovery Document validation b. Reactive Software Product c. Register PUT GET d. Amending Account for an Existing Consent Scenario with PAR e. Consent Software Statement Assertion with Sector Identifier URI

Test Plan Version	CDS Standards Version	Issued Date	Description of Changes
4.0.1	1.19.0	21 September 2022	This document covers the guidance related to the Data Holder (DH) Test Plan (TP) version 4.0.1. This test plan has been updated to conform with the latest version of Consumer Data Standards version 1.19.0. Additionally it has been enhanced to support requests to all active versions of the following Register APIs: • Get Data Recipients - V1 , V2 and V3 • Get Data Recipients Statuses - V1 and V2 • Get Software Product Statuses V1 and V2
4.0.2	1.19.0	03 November 2022	This test plan includes bug fixes for the following scenarios: 2. Concurrent Consent 7. Amending Existing Consent

2 Overview

2.1 Document Purpose

The purpose of this document is to provide technical information about the Consumer Data Right (CDR) Conformance Test Suite (CTS) Data Holder Test Plan. It will provide an in-depth understanding to Data Holder (DH) brands on:

- the scope of the CTS
- the purpose of each CTS scenario
- what is being tested to ensure technical conformance
- pass and fail conditions for each CTS scenario
- how to react correctly to valid and invalid request.

2.2 Background

The CTS is a final checkpoint for participants of key elements of a participant's solution before activation in the ecosystem. The primary focus of the CTS is to provide the ACCC as the CDR Registrar, performing its function to maintain the security, integrity, and stability of the Register, with a level of confidence that:

- a participant has delivered to the security standard required for CDR
- a participant is able to share consumer data in the CDR ecosystem without significant disruption
- key capabilities have been built unless an exemption has been granted.

The CTS is designed to verify a limited subset of standards alignment against security profile and consent components as well as other high-risk areas. The CTS will continue to evolve and further test scenarios will be added as ecosystem requirements change. The execution of CTS is not a one-time event for a participant, it is expected that active participants will complete the new test scenarios as standards are updated or their software evolves.

The CTS is **not designed** to:

- test the internal workings and validations of a Data Holder brand or data recipient software products
- test compliance to **all** CDR Rules (the Rules) and CDS
- be a sandbox or assisted development tool. It will not help participants design and build a product that conforms to the CDS. Before undertaking the CTS, participants require a production-ready data holder brand or data recipient software product that is built in accordance with the CDS.

For the steps you need to complete before you can use the CTS please consult the On-boarding for data holders page on the CDR website.

2.3 CTS Scope for Data Holders

The CTS interacts with the Data Holder's brand and assesses the technical competency in conforming to the CDS. To achieve this, the CTS simulates the Register and an Accredited Data

Recipient (ADR), testing that the DH brand can safely interact with a data recipient in the system.

In Scope	Out of Scope
The CTS will conduct a series of tests to determine the technical competency of the DH and whether they can conform to the CDS. • Dynamic Client Registration (DCR) • Interaction with the Register • Consent • Common APIs (Get Customer) • Register status • Software Status • Consent withdrawal • Token revocation	The CTS does not test the internal workings and validations of DH brands. • How consent is managed within a DH's brand • How a DH's brand correctly handles certain consent flow attack vectors • How a DH removes consent and consumer data in their brand • How a DH's brand conforms to FAPI 1.0 standards
Note: The CTS includes only those endpoints that are detailed in this document.	

2.4 Technical Considerations

2.4.1 Stateless Protocol

The CTS has been designed to be “stateless” between scenarios but “stateful” within a test scenario so that participants can run test scenarios in any order and run each test scenario as many times as required to complete the certification process. A consequence of this stateless behavior is that participants who are engaged in the CTS testing process cannot reliably assume that the results of one test will be carried forward to a subsequent test. For example, a test designed for a DH may require them to interact with a simulated ADR, including steps to determine the status of the ADR or the ADR’s Software Product. However, the DH shouldn’t assume that the simulated ADR will still exist when a subsequent test is run and nor should the DH rely on the cached value/s for that ADR.

Due to the stateless nature of CTS, participants will encounter an influx of participant metadata records when performing the necessary polling of the CTS Simulated Register during testing. Therefore, the participant may choose to implement a process to regularly purge test data (e.g. after each scenario / at the end of each testing day / at the completion of CTS testing) to return their solution to a known, baseline state. Participant metadata should not be removed whilst executing a CTS scenario (i.e. between test steps) as it may result in errors.

3 DH CTS Scenarios and Tests

This section captures the following scenarios and all their associated steps that can be part of a DH Test Plan:

1. Dynamic Client Registration

2. Concurrent Consent

3a. Get Software Product Status Register Polling

3b. Get Data Recipients Register Polling

4. Ensure Infosec Endpoints Using MTLS

5. Ensure Holder of Key for Resource Requests

6. Ensure Client Assertion Data In Token Request

7. Amending Existing Consent

8. Removed Software Product

9. Data Holder Initiated Revocation

10. Data Recipient Initiated Revocation

11. Data Recipient Initiated Token Revocation

Note: If your brand checks the ADR register status by polling the register GetSoftwareProductStatus then the Get Software Product Status Register Polling will be assigned. Alternatively, if your brand checks the ADR register status by polling the register GetDataRecipients then the Get Data Recipients Register Polling scenario will be assigned. The appropriate scenario is determined during the enrolment process and assigned accordingly.

3.1 CTS Entry Criteria

The CTS simulates both the Register and an ADR that your brand interacts with. You should enrol in the CTS when your brand is ready for production release or close to being ready. After receiving your enrolment confirmation, you can start your CTS tests. You are advised to take the CTS tests in the order they are listed in this document.

Before you start:

1. Apply the test certificate to your brand.
2. You must have a valid account on the CDR Participant Portal.
3. You must be registered as a CTS tester as part of the CTS enrolment.

3.2 CTS Exit Criteria

1. You must execute the tests as selected in your enrolment form.
2. This test can be run multiple times during the test run. The result of the last attempt of the test will be included in the test run report for the CTS outcome assessment.
3. You are required to provide test results for all scenarios on your test plan, or to provide justification on why the test/s is not relevant. Australian Competition and Consumer

Commission (ACCC) can give special consideration on whether to grant a CTS Pass status even if you fail a test.

4. Submit the test result via the DH User Interface (UI) after finishing the tests. You must inform the On-boarding Officers when you submit your test results via email, so that an On-boarding Officer can start assessing your results.

4 Dynamic Client Registration Scenario

4.1 Purpose

The ability for a Participant DH to validate they correctly register a software product from the CTS simulated ADR and ensure that the Data Holder registration process can respond to a PUT and GET DCR request, for a given Client ID.

This scenario forms the starting point for CTS Testing.

4.2 Scenario Conditions

Not Applicable.

4.3 Endpoints

See also Endpoints used in Data Holder Scenarios.

Endpoint	Description	Method
OpenID Provider Configuration End Point	CTS Simulated ADR requests the discovery document from the Data Holder via the Discovery Endpoint	GET
Dynamic Client Registration	CTS Simulated ADR sends a DCR request to the Data Holder via the Registration Endpoint	POST
Get JWKS (Register)	Participant DH requests the JWKS from the CTS Simulated Register via the Register's Get JWKS Endpoint	GET
Get JWKS (ADR)	Participant DH requests the JWKS from the CTS Simulated ADR via the JWKS Endpoint	GET
Redirect URI	Participant DH calls the CTS Simulated ADR Redirect Uri Endpoint to signin	GET
Get Data Recipient Status	Participant DH requests the data recipient status from the CTS Simulated Register via the Get Data Recipient Status Endpoint	GET

Get Software Product Status	Participant DH requests the software product status from the CTS Simulated Register via the Get Software Product Status Endpoint	GET
Get Data Recipients	Participant DH requests the data recipients from the CTS Simulated Register via the Get data recipients Endpoint	GET
Update Data Recipient Registration	CTS Simulated ADR sends an update Client Registration request for a given Client ID via the Update Data Recipient Registration endpoint.	PUT
Get oAuth Client Registration	CTS Simulated ADR sends a get Client Registration request for a given Client ID via the Get oAuth Client Registration endpoint.	GET

Link to specifications

https://consumerdatastandardsaustralia.github.io/standards-archives/standards-1.18.0/#dcr-apis

4.4 Scenario Results

Pass

You have passed the scenario test when you **can**:

- receive a valid GET request to your Discovery Endpoint (provided during enrolment) and send a valid Discovery Document JSON response
- receive a valid DCR request from the CTS Simulated ADR, successfully register the CTS Simulated ADR software product, and supply a valid response
- can process both GET and PUT requests to the registration Endpoint.

Fail

You have failed the scenario test when you **cannot**:

- receive a valid GET request to your Discovery Endpoint (provided during enrolment).
- receive a valid DCR request from the CTS Simulated ADR, register the CTS Simulated ADR software product and supply a valid response
- process both GET and PUT requests to the registration Endpoint.

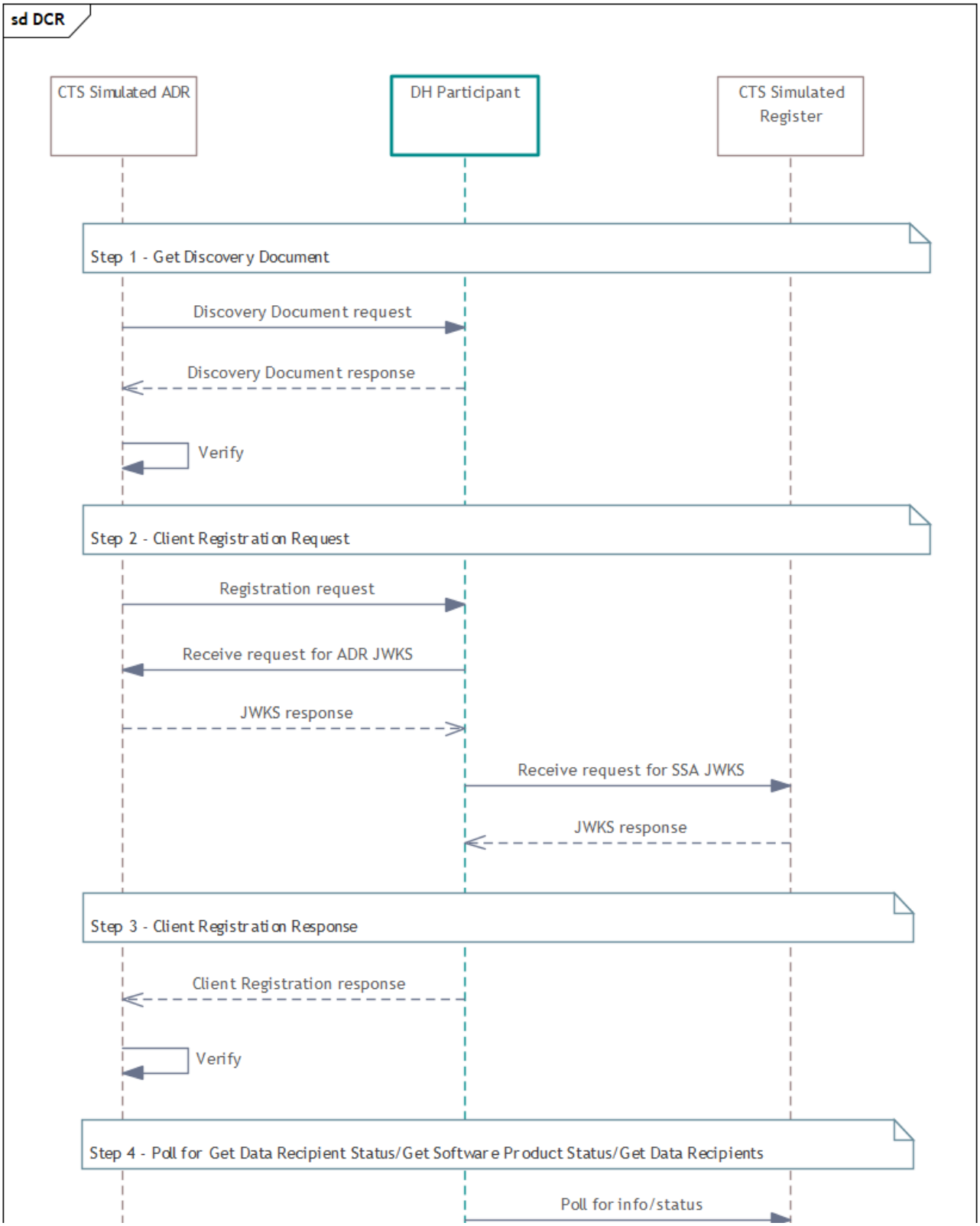
4.5 Scenario High-Level Test Steps

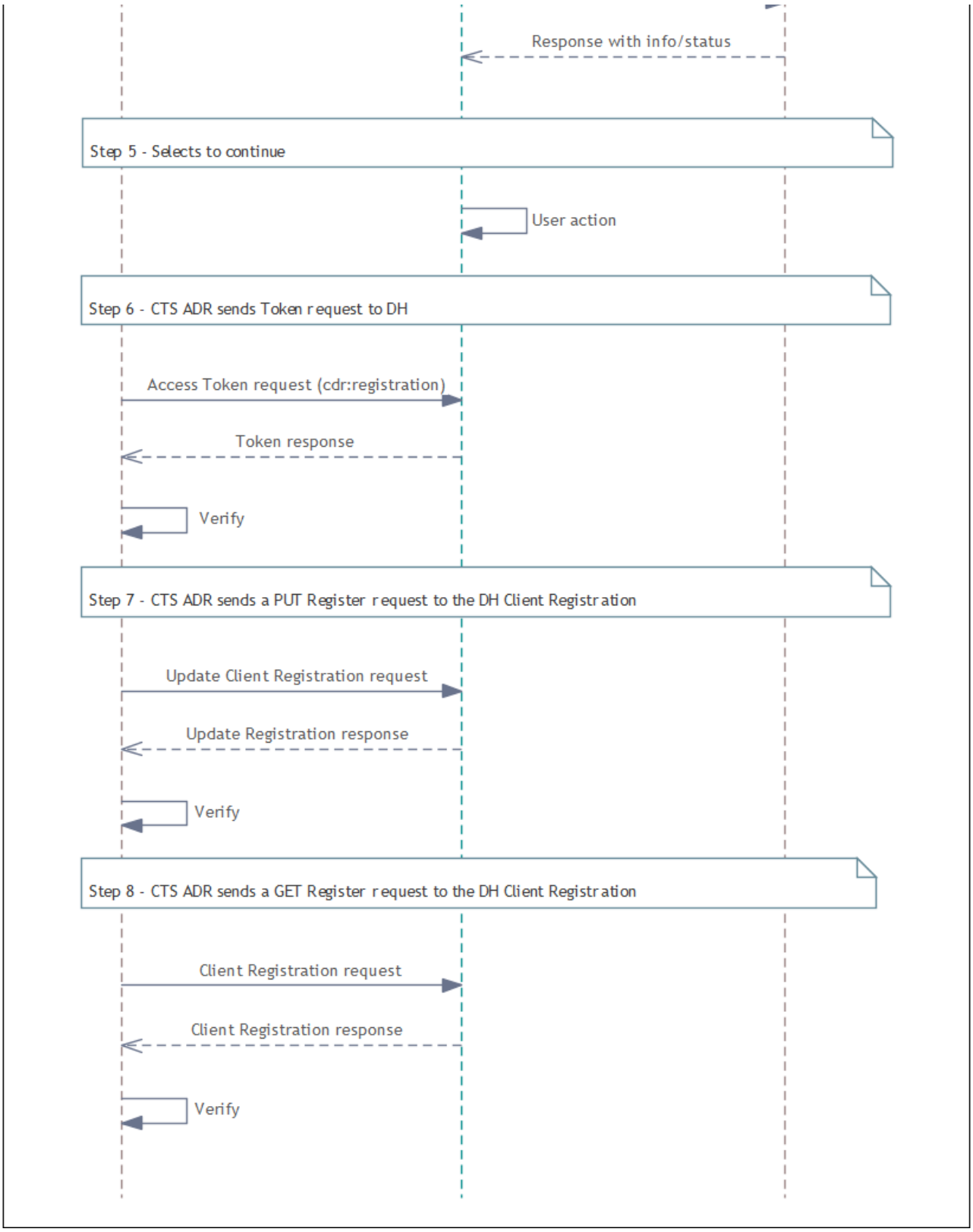
1. **CTS ADR requests a Discovery Document from the Participant DH**

a. CTS Simulated ADR sends a Discovery Document request to the Participant DH via the OpenID Provider Configuration Endpoint.
b. Participant DH returns a response with their Discovery Document.
c. CTS verifies the Discovery Document *(DCR scenario only).*

2. **CTS ADR sends a DCR request to the Participant DH**
   a. CTS Simulated ADR sends a DCR request to the Participant DH.
   b. Participant DH receives and verifies the CTS Simulated ADR DCR request.
      i. Participant DH calls the CTS Simulated ADR JWKS Endpoint (using the jwks_uri from the SSA).
      ii. CTS Simulated ADR returns a response with an ADR JWKS.
      iii. Participant DH calls the CTS Simulated Register SSA JWKS Endpoint to verify the SSA signature using the Register public keys.
      iv. CTS Simulated Register returns a response with an SSA JWKS.
3. **DH responds to the CTS ADR DCR request**
   a. Participant DH registers the Software Product and returns a response to the CTS Simulated ADR.
   b. CTS verifies the response.
4. **DH polls the register to Get Data Recipient status**
   a. Participant DH sends a data recipient status request to the CTS Simulated Register via the Get Data Recipient Statuses Endpoint.
   b. CTS Simulated Register returns a valid response and/or Participant DH polls the register to Get Software Product Status.
   c. Participant DH sends a 'software product status' request to the CTS Simulated Register via the Get Software Product Status Endpoint.
   d. CTS Simulated Register returns a valid response and/or Participant DH polls the register to Get Data Recipients.
   e. Participant DH sends a 'data recipients' request to the CTS Simulated Register via the Get Data Recipients Endpoint.
   f. CTS Simulated Register returns a valid response.
5. **DH selects to continue through the User Interface (UI) after at least one of the above Register APIs is polled**
6. **CTS ADR sends Token request to DH**
   a. CTS Simulated ADR sends a POST Token request to the Participant DH via the Token Endpoint using ClientCredentials grant type.
   b. Participant DH validates the CTS request and returns a response.
   c. CTS verifies the response.
7. **CTS ADR sends PUT Register a request via the Update Data Recipient Registration Endpoint**
   a. CTS Simulated ADR sends a PUT Registration Request containing the bearer token from step 4, with an updated SSA to reflect the changes, to the Participant DH via the Update Data Recipient Registration Endpoint.
   b. Participant DH validates the CTS request and returns a response.
   c. CTS verifies the response.
8. **CTS ADR sends GET Register a request via the Get oAuth Client Registration Endpoint**
   a. CTS Simulated ADR sends a GET Register request with the bearer token via the DH Get oAuth Client Registration Endpoint.

b. Participant DH validates the CTS request and returns a response.
c. CTS verifies the response equates to the PUT request that was sent to the DH DCR Register Endpoint.





Dynamic Client Registration Scenario Sequence Diagram

5 Concurrent Consent Scenario

5.1 Purpose

The ability for a Participant DH to verify the Consent flow with the CTS Simulated ADR by granting two consent arrangements for a single ADR/Consumer pairing (multiple CDR Arrangement IDs). The Participant DH receives a call from the CTS Simulated ADR to the Get Customer endpoint with the access token as part of the consent.

5.1.1 Business Context

Concurrent consent allows the same ADR software product to establish more than one active consent with a Participant DH on the consumer's behalf. This is in support of the CDR's intention to allow consents with different use cases or purposes to be established under one ADR application so that ADRs can correctly observe the Data Minimisation principle. This is achieved technically through the establishment of separate CDR Arrangement IDs for each individual consent.

5.2 Scenario Conditions

Not Applicable.

5.3 Endpoints

See also Endpoints used in Data Holder Scenarios.

Endpoint	Description	Method
Authorize	CTS Simulated ADR requests with the Participant DH via the Authorize Endpoint	GET
Token	CTS Simulated ADR exchanges their code for a Token from the Participant DH via the Token Endpoint CTS Simulated ADR exchanges their Refresh Token for an Access Token from the Participant DH via the Token Endpoint	POST
Introspection	CTS Simulated ADR sends an Introspection request to the Participant DH Token Introspection Endpoint to retrieve information about a token	POST
Get Customer	CTS Simulated ADR sends a request to the Participant DH's Get Customer Endpoint	GET

Pushed Authorisation	CTS Simulated ADR sends a Pushed Authorisation Request object for request_uri to the Participant DH via Pushed Authorisation Endpoint	POST

Link to specifications

https://consumerdatastandardsaustralia.github.io/standards-archives/standards-1.18.0/#introduction

5.4 Scenario Results

Pass

You have passed the Concurrent Consent scenario test when you **can**:

- Authorize 2 consents on behalf of a single consumer
- Return an authorization and token response, for each consent
- Return a valid response payload from the common API endpoint.

Fail

You have failed the Concurrent Consent scenario test when you **cannot**:

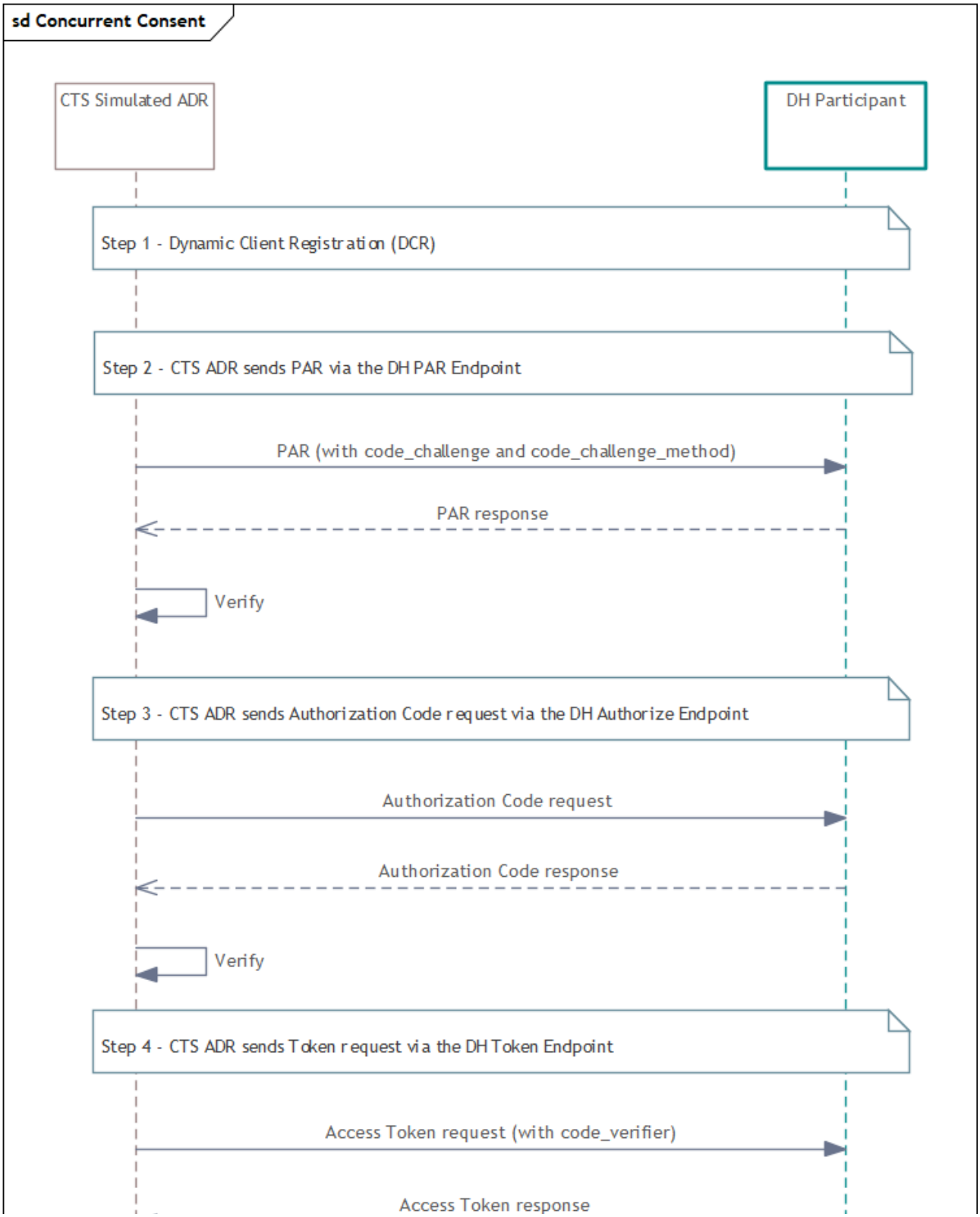
- Authorize 2 consents on behalf of a consumer
- Return multiple authorization responses and token responses (Access and Refresh Token)
- Ensure that with each consent established, a new CDR Arrangement ID is created
- Redirect back to the CTS Simulated ADR
- Return a valid response payload from the common API endpoint.

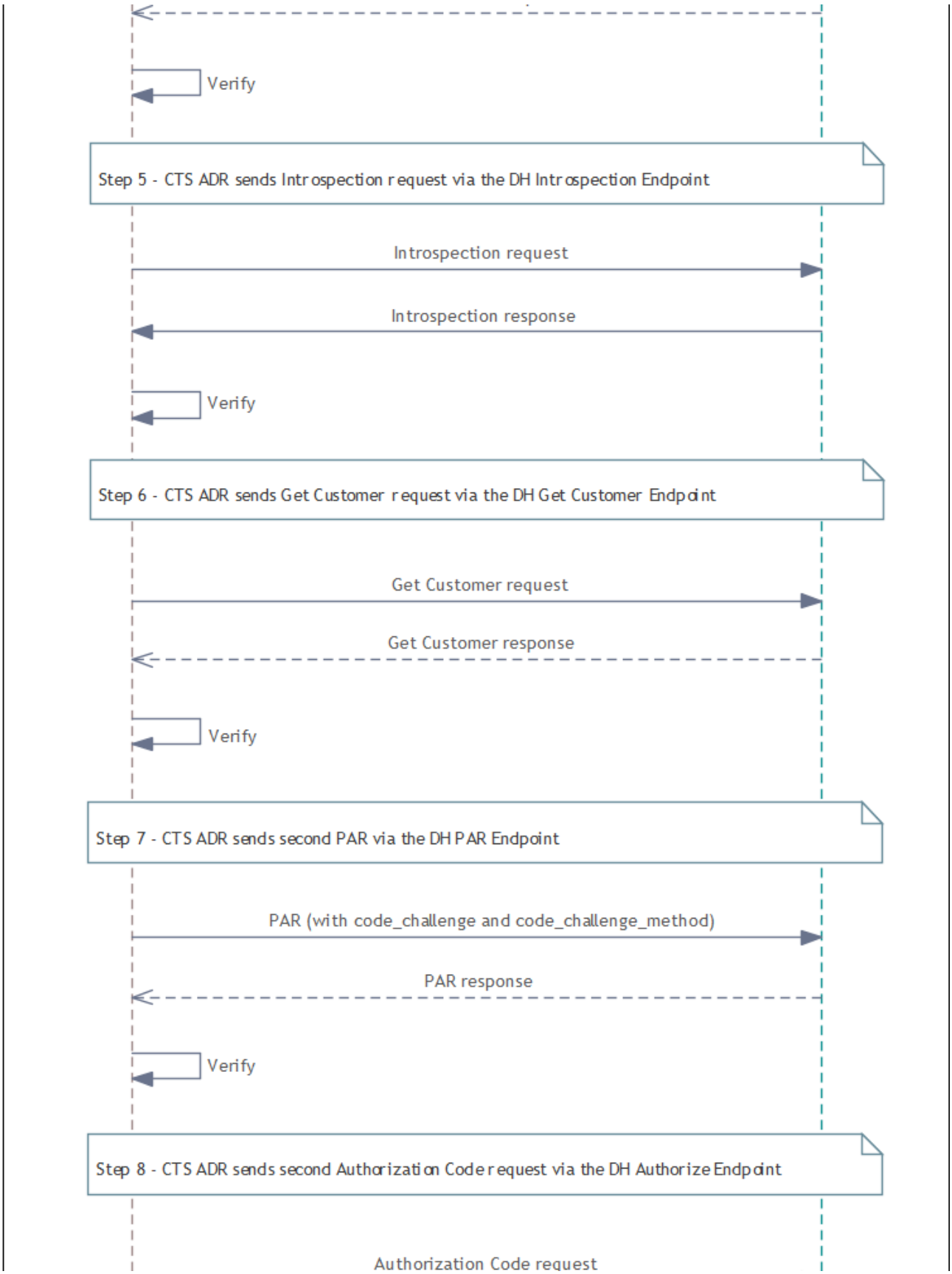
5.5 Scenario High Level Test Steps

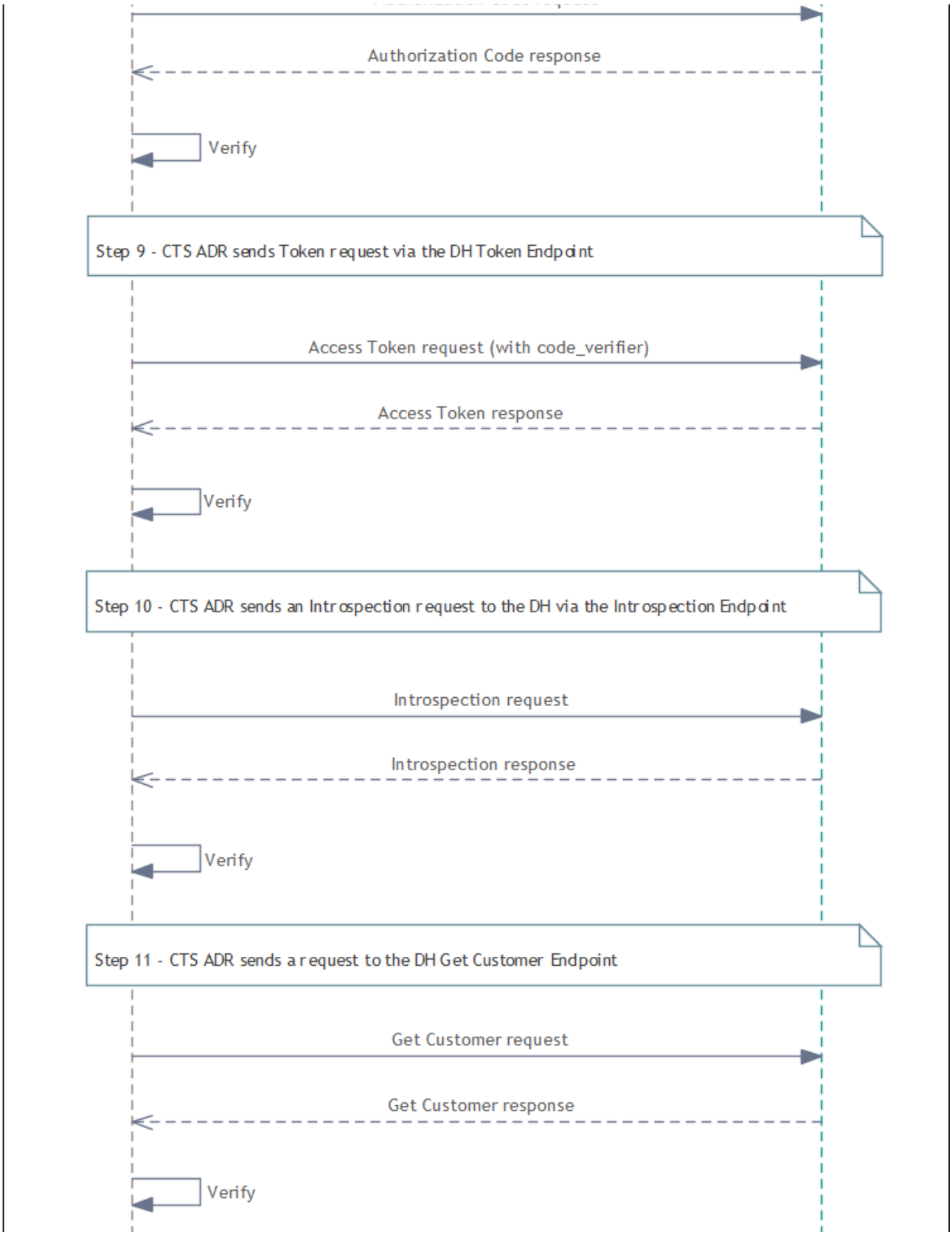
1. Dynamic Client Registration (DCR) *(Steps 1 - 4 from DCR scenario)*
2. **CTS ADR sends Pushed Authorization Request (PAR) via the DH Pushed Authorisation Endpoint**
    a. CTS Simulated ADR sends a PAR with the request object to the Participant DH via the Pushed Authorisation Endpoint.
    b. Participant DH validates the CTS Simulated ADR PAR, and responds with request_uri.
    c. CTS verifies that in the Participant DH PAR response that request_uri is contained in the response.
3. **CTS ADR requests authorization via the DH Authorize Endpoint**
    a. CTS Simulated ADR sends a request to the Participant DH via the Authorize Endpoint.
    b. Participant DH validates the CTS Simulated ADR Authorize request, verifying that the Simulated ADR Software product is registered with the Participant DH and responds via the Redirect URI with Code, State and encrypted and signed ID Token.
    c. CTS verifies the Participant DH Authorize response.
4. **CTS ADR sends Token request via the DH Token Endpoint**
    a. CTS Simulated ADR sends a Token request to the Participant DH via the Token Endpoint, exchanging their code for a Token.

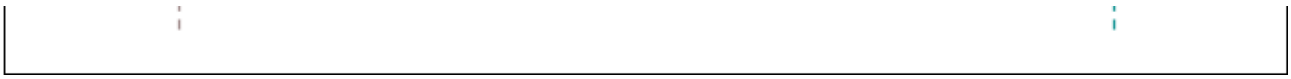
   b. Participant DH validates the CTS Simulated ADR Token request and returns a response with a cdr_arrangement_id, access token, and refresh token (for ongoing consents only).
   c. CTS verifies the Participant DH Token response.
5. **CTS ADR sends Introspection request via the DH Introspection Endpoint**
   a. CTS Simulated ADR sends an Introspection request to the Participant DH Token Introspection Endpoint.
   b. Participant DH validates the CTS Simulated ADR Introspection request and returns a valid response.
   c. CTS verifies the Participant DH Token Introspection response.
6. **CTS ADR sends request to the DH Get Customer Endpoint**
   a. CTS Simulated ADR sends a request, using the Participant DH issued Access Token, to the Participant DH via the Get Customer Endpoint.
   b. Participant DH validates the CTS Simulated ADR request and returns a response with the mock Customer payload.
   c. CTS verifies the Participant DH Get Customer response.
7. **CTS ADR sends a second PAR via the DH Pushed Authorisation Endpoint**
   a. CTS Simulated ADR sends a second PAR with the request object to the Participant DH via the Pushed Authorisation Endpoint.
   b. Participant DH validates the CTS Simulated ADR second PAR and responds with request_uri.
   c. CTS verifies that in the Participant DH second PAR response that request_uri is contained in the response.
8. **CTS ADR sends a second Authorization Code request via the DH Authorize Endpoint**
   a. CTS Simulated ADR sends a request to the Participant DH via the Authorize Endpoint for the same user.
   b. Participant DH validates the CTS Simulated ADR Authorize request, verifying that the CTS Simulated ADR Software product is registered with the Participant DH and responds via the Redirect URI with Code, State and encrypted and signed ID Token.
   c. CTS verifies the Participant DH response.
9. **CTS ADR sends Token request via the DH Token Endpoint**
   a. CTS Simulated ADR sends a Token request to the Participant DH via the Token Endpoint, exchanging their code for a Token.
   b. Participant DH validates the CTS Simulated ADR Token request and returns a response with a cdr_arrangement_id , access token, and refresh token (for ongoing consents only).
10. **CTS ADR sends Introspection request via DH Introspection Endpoint**
    a. CTS Simulated ADR sends an Introspection request to the Participant DH Token Introspection Endpoint.
    b. Participant DH validates the CTS Simulated ADR Introspection request and returns a valid response.
    c. CTS verifies the Participant DH Token Introspection response.
11. **CTS ADR sends Get Customer request via the DH Get Customer Endpoint**
    a. CTS Simulated ADR sends a request, using the Participant DH issued Access Token, to the Participant DH via the Get Customer Endpoint.

b. Participant DH validates the CTS Simulated ADR request and returns a response with the mock Customer payload.
c. CTS verifies the Participant DH Get Customer response.




Verify
Step 5 - CTS ADR sends Introspection request via the DH Introspection Endpoint
Introspection request
Introspection response
Verify
Step 6 - CTS ADR sends Get Customer request via the DH Get Customer Endpoint
Get Customer request
Get Customer response
Verify
Step 7 - CTS ADR sends second PAR via the DH PAR Endpoint
PAR (with code_challenge and code_challenge_method)
PAR response
Verify
Step 8 - CTS ADR sends second Authorization Code request via the DH Authorize Endpoint
Authorization Code request


Authorization Code response
Verify
Step 9 - CTS ADR sends Token request via the DH Token Endpoint
Access Token request (with code_verifier)
Access Token response
Verify
Step 10 - CTS ADR sends an Introspection request to the DH via the Introspection Endpoint
Introspection request
Introspection response
Verify
Step 11 - CTS ADR sends a request to the DH Get Customer Endpoint
Get Customer request
Get Customer response
Verify



Concurrent Consent Scenario Sequence Diagram

6 Get Software Product Status Register Polling Scenario

6.1 Purpose

The ability for a Participant DH to validate they correctly poll CTS Register every 5 minutes for a Participant ADR's Software Product Status changes. To achieve this, the Participant DH can poll the Get Software Products Status API to retrieve the current status and cache this for use during requests for Consumer Data.

6.2 Scenario Conditions

Not Applicable.

6.3 Endpoints

The table below lists endpoints specific to this scenario. See also Endpoints used in Data Holder Scenarios.

Endpoint	Description	Method
Get Software Product Status	Participant DH requests the Software Product Status from the CTS Simulated Register via the Get Software Product Status Endpoint	GET

Link to specifications

https://consumerdatastandardsaustralia.github.io/standards-archives/standards-1.18.0/#get-software-products-statuses

6.4 Scenario Results

Pass

You have passed the CTS Simulated Register Polling tests when you call the Software Product Status endpoint twice within 5 minutes.

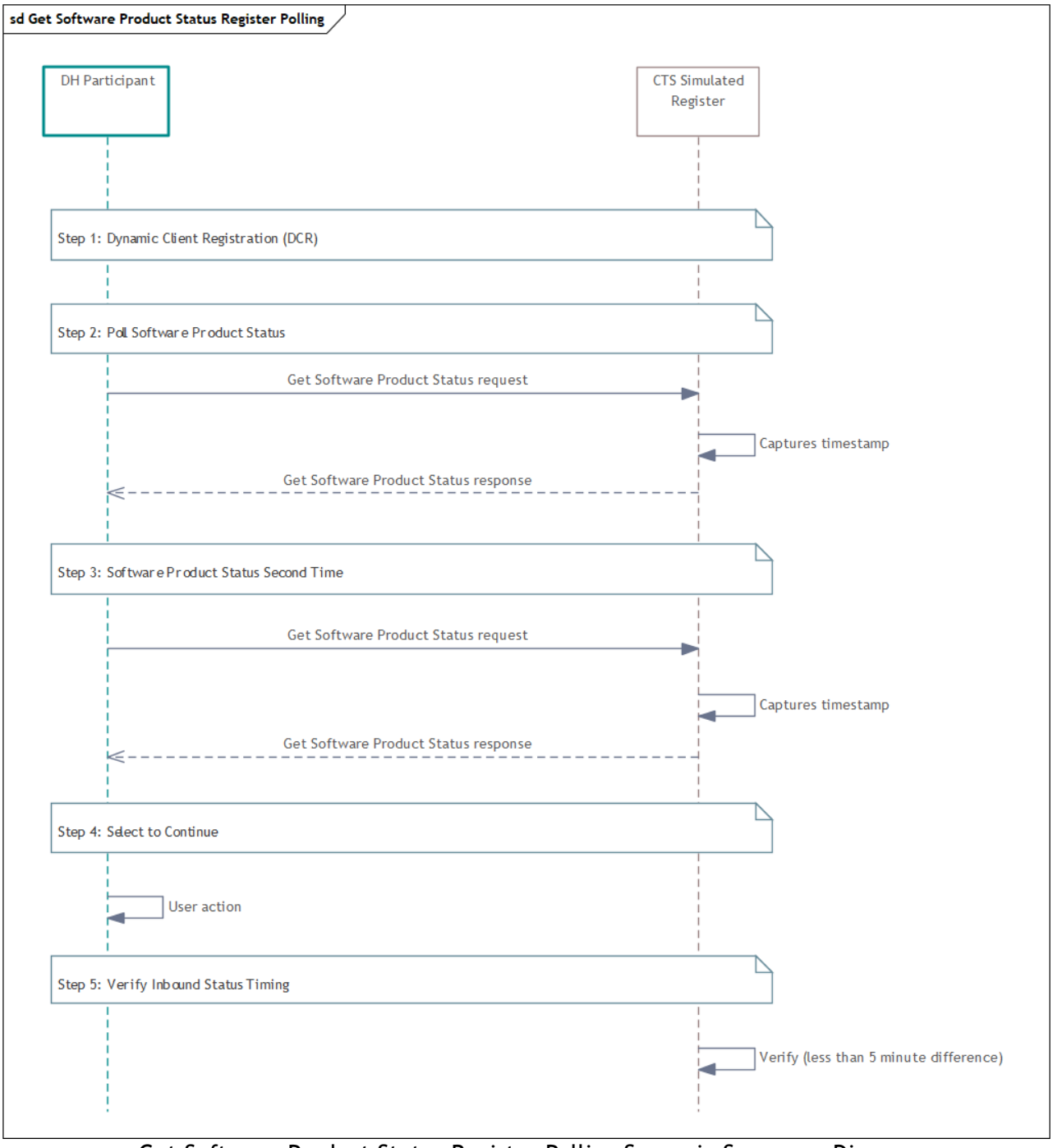
Fail

You have failed the CTS Simulated Register Polling tests when you **do not** call the Software Product Status endpoint twice within 5 minutes.

6.5 Scenario High Level Test Steps

1. Dynamic Client Registration (DCR) *(Steps 1, 2, 3, 4 and 5 from DCR scenario)*
2. **DH polls the Register to Get Software Product Status**
    a. Participant DH sends a 'Software Product Status' request to the CTS Simulated Register via the Get Software Products Status endpoint.
    b. CTS Simulated Register returns a valid response.

   c. CTS captures a timestamp of the first polling request.
3. **DH polls the Register a second time to Get Software Product Status**
   a. Participant DH sends a ‘Software Product Status’ request to the CTS Simulated Register via the Get Software Products Status endpoint.
   b. CTS Simulated Register returns a valid response.
   c. CTS captures a timestamp of the second polling request.
4. **DH selects to continue the execution of the Scenario through the UI**
5. **CTS verifies inbound status request time difference**
   a. CTS compares the timestamps of the two polling requests.
   b. CTS verifies a second status request was received in less than 5 minutes.



Get Software Product Status Register Polling Scenario Sequence Diagram

7 Get Data Recipients Register Polling Scenario

7.1 Purpose

The ability for a Participant DH to validate they correctly poll CTS Register every 5 minutes for the list of registered data recipients. To achieve this, the Participant DH can poll the Get Data Recipients API to retrieve the current list of registered Accredited Data Recipients and cache this for use during requests for Consumer Data.

7.2 Scenario Conditions

Not Applicable.

7.3 Endpoints

The table below lists endpoints specific to this scenario. See also Endpoints used in Data Holder Scenarios.

Endpoint	Description	Method
Get Data Recipients	Participant DH requests the Data Recipients from the CTS Simulated Register via the Get Data Recipients Endpoint	GET

Link to specifications

https://consumerdatastandardsaustralia.github.io/standards-archives/standards-1.18.0/#get-data-recipients

7.4 Scenario Results

Pass

You have passed the CTS Simulated Register Polling tests when you call the Get Data Recipients endpoint twice within 5 minutes.

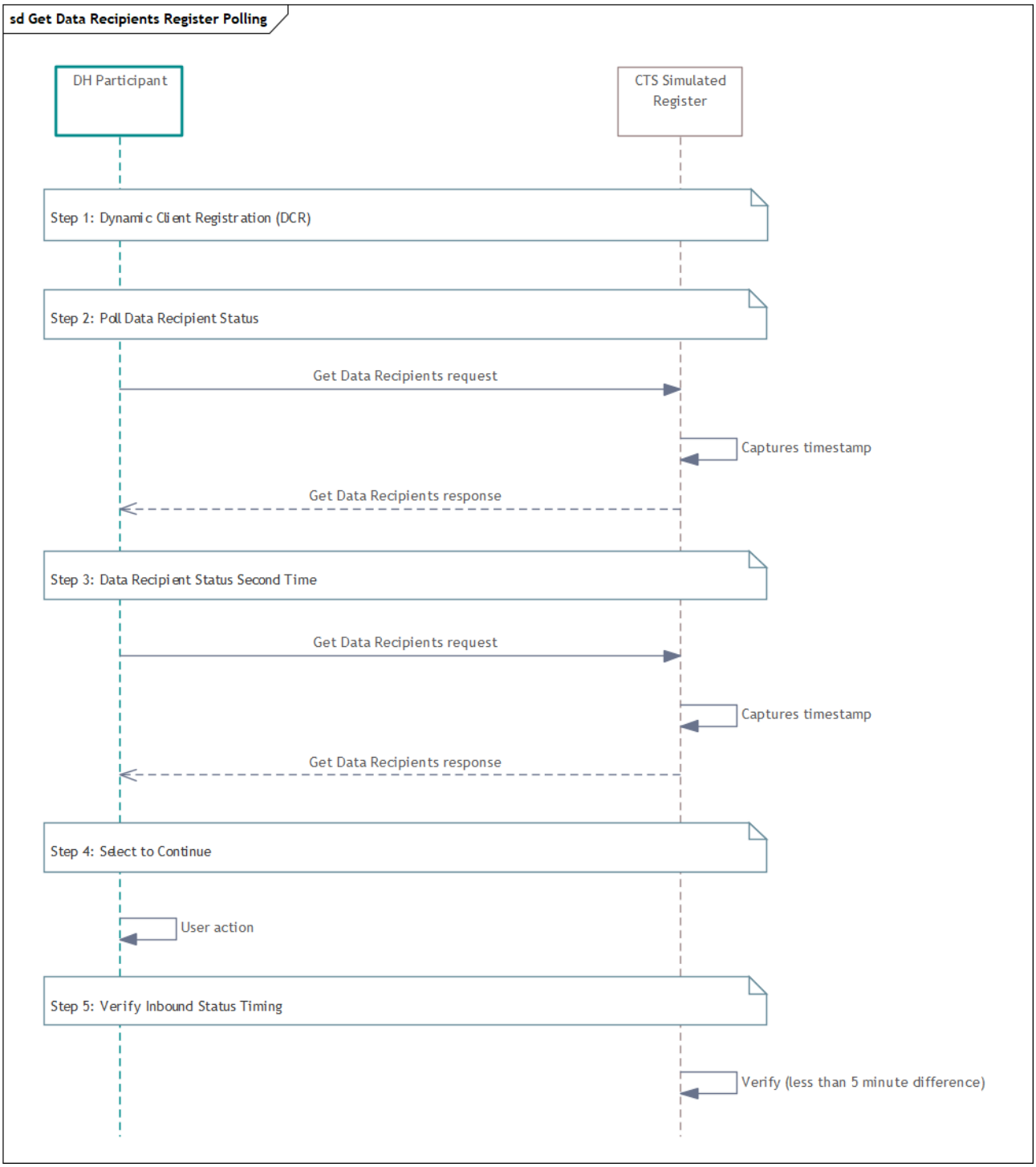
Fail

You have failed the CTS Simulated Register Polling tests when you **do not** call the Get Data Recipients endpoint twice within 5 minutes.

7.5 Scenario High Level Test Steps

1. Dynamic Client Registration (DCR) *(Steps 1, 2, 3, 4 and 5 from DCR scenario)*
2. **DH polls the Register to Get Data Recipients**
    a. Participant DH sends a 'Data Recipients' request to the CTS Simulated Register via the Get Data Recipients Endpoint.

   b. CTS Simulated Register returns a valid response.
   c. CTS captures a timestamp of the first polling request.
3. **DH polls the Register a second time to Get Data Recipients**
   a. Participant DH sends a ‘Data Recipients’ request to the CTS Simulated Register via the Get Data Recipients Endpoint.
   b. CTS Simulated Register returns a valid response.
   c. CTS captures a timestamp of the second polling request.
4. **DH selects to continue the execution of the Scenario through the UI**
5. **CTS verifies inbound status request time difference**
   a. CTS compares the timestamps of the two polling requests.
   b. CTS verifies a second status request was received in less than 5 minutes.



Get Data Recipients Register Polling Scenario Sequence Diagram

8 Ensure Infosec Endpoints Using MTLS Scenario

8.1 Purpose

The ability for a Participant DH to validate that they correctly handle an invalid client certificate received from a Participant ADR. This CTS scenario includes:

- registration request sent without a CDR client certificate
- registration request sent with an expired CDR client certificate
- registration request sent with a non-CDR (self-signed) client certificate
- registration request sent with a revoked CDR client certificate.

8.2 Scenario Conditions

Not Applicable.

8.3 Endpoints

The table below lists endpoints specific to this scenario. See also Endpoints used in Data Holder Scenarios.

Endpoint	Description	Method
Dynamic Client Registration	CTS Simulated ADR sends a DCR request to the Participant DH via the Registration Endpoint	POST

Link to specifications

https://consumerdatastandardsaustralia.github.io/standards-archives/standards-1.18.0/#dcr-apis

8.4 Scenario Results

Pass

You have passed the scenario when the registration request is sent with an invalid/missing client certificate and the response is an appropriate error that indicates invalidation of the certificate.

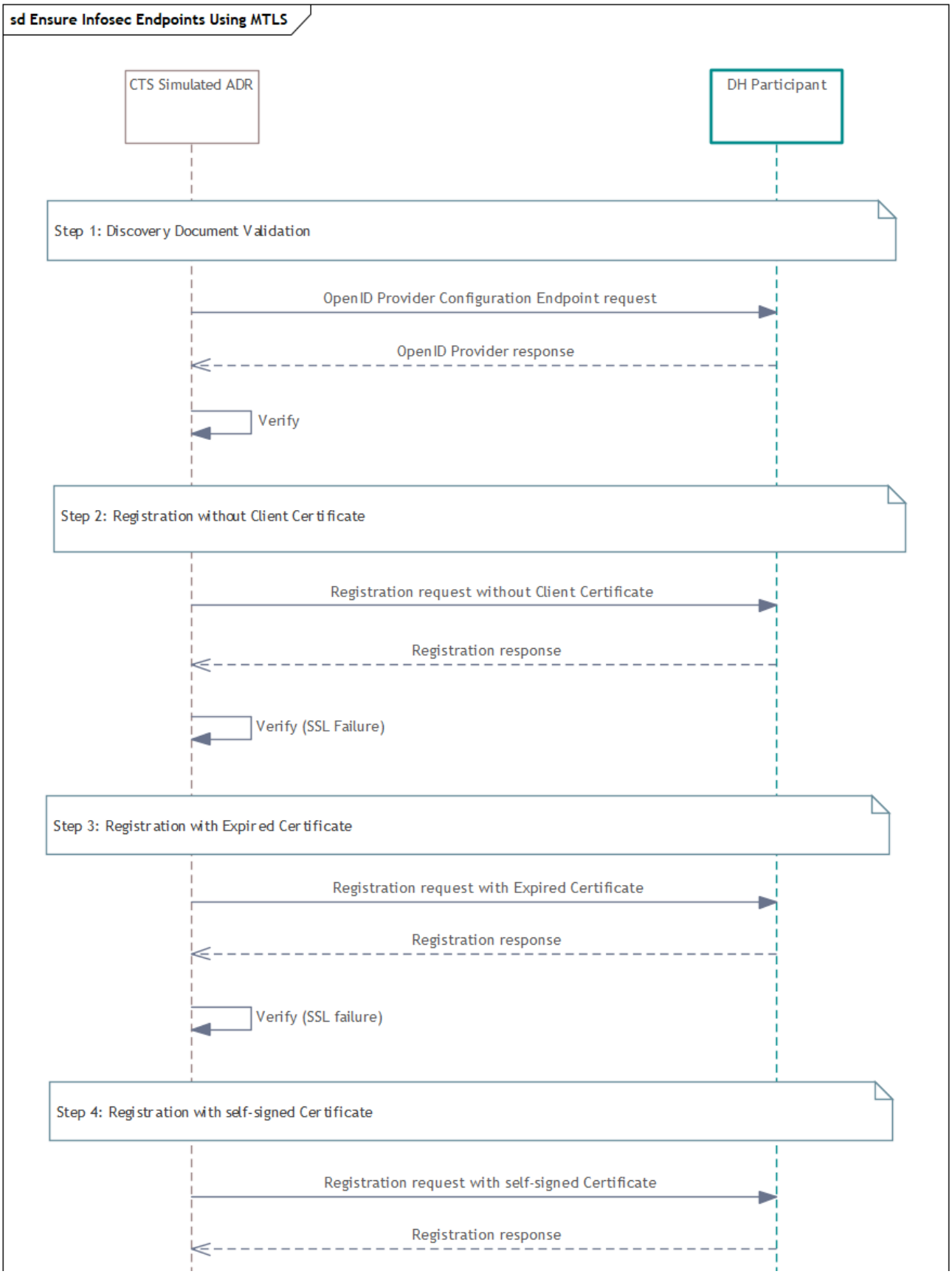
Fail

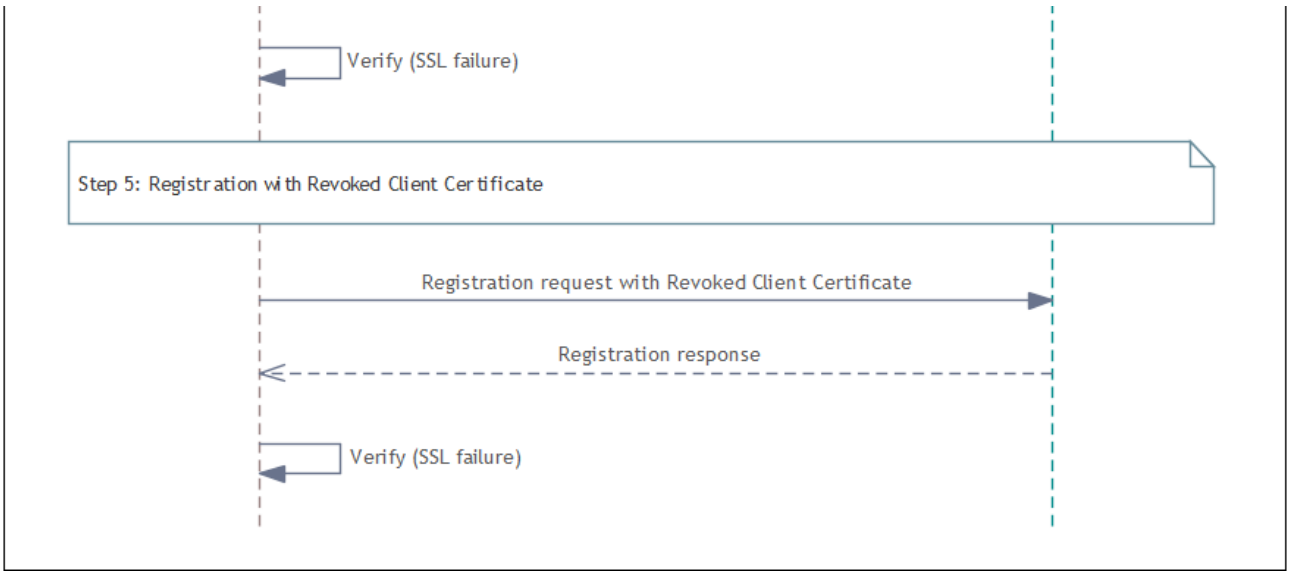
You have failed the scenario when the registration request is sent with an invalid/missing client certificate and the response is **not** an appropriate error.

8.5 Scenario High Level Test Steps

1. **Discovery Document Validation**

   a. CTS Simulated ADR sends a OpenID Provider Configuration Endpoint request to the Participant DH.
   b. Participant DH validates the CTS Simulated ADR second PAR and responds with OpenID Provider.
   c. CTS verifies that the OpenID Provider is contained in the response.
2. **CTS ADR sends registration request via the DH Registration Endpoint without CDR Client Certificate**
   a. CTS Simulated ADR sends a registration request without a CDR Client Certificate to the Participant DH.
   b. Participant DH validates the CTS Simulated ADR request and responds with error "`400 No required SSL certificate was sent`".
3. **CTS ADR sends registration request via the DH Registration Endpoint with an expired CDR Client Certificate**
   a. CTS Simulated ADR sends a registration request with an expired CDR Client Certificate to the Participant DH.
   b. Participant DH validates the CTS Simulated ADR request and responds with error "`400 The SSL certificate error`".
4. **CTS ADR sends registration request via the DH Registration Endpoint with a non-CDR (self-signed) Client Certificate**
   a. CTS Simulated ADR sends a registration request with a non-CDR (self-signed) Client Certificate to the Participant DH.
   b. Participant DH validates the CTS Simulated ADR request and responds with error "`400 The SSL certificate error`".
5. **CTS ADR sends registration request via the DH Registration Endpoint with a revoked CDR Client Certificate**
   a. CTS Simulated ADR sends a registration request with a revoked CDR Client Certificate to the Participant DH.
   b. Participant DH validates the CTS Simulated ADR request and responds with error "`invalid_software_statement`".


sd Ensure Infosec Endpoints Using MTLS
CTS Simulated ADR
DH Participant
Step 1: Discovery Document Validation
OpenID Provider Configuration Endpoint request
OpenID Provider response
Verify
Step 2: Registration without Client Certificate
Registration request without Client Certificate
Registration response
Verify (SSL Failure)
Step 3: Registration with Expired Certificate
Registration request with Expired Certificate
Registration response
Verify (SSL failure)
Step 4: Registration with self-signed Certificate
Registration request with self-signed Certificate
Registration response



Ensure Infosec Endpoints Using MTLS Scenario Sequence Diagram

9 Ensure Holder of Key for Resource Requests Scenario

9.1 Purpose

The ability for a Participant DH to validate their compliance to the Holder of Key (HoK) mechanism when accessing MTLS Secured APIs.

9.2 Scenario Conditions

The Get Customer API request is made with a CDR certificate that is not bound to the access token used in the request.

9.3 Endpoints

The table below lists endpoints specific to this scenario. See also Endpoints used in Data Holder Scenarios.

Endpoint	Description	Method
Get Customer	CTS Simulated ADR sends a request to the Participant DH's Get Customer Endpoint	GET
Authorize	CTS Simulated ADR requests with the Participant DH via the authorize Endpoint	GET
Token	CTS Simulated ADR exchanges their code for a Token from the Participant DH via the Token Endpoint	POST
Get Data Recipient Status	Participant DH requests the data recipient status from the CTS Simulated Register via the Get Data Recipient Status Endpoint	GET
Get Data Recipients	Participant DH requests the data recipients from the CTS Simulated Register via the Get Data Recipients Endpoint	GET
Get Software Product Status	Participant DH requests the software product status from the CTS Simulated Register via the Get Software Product Status Endpoint	GET

Arrangement Revocation **DR to DH**	CTS Simulated ADR sends a request, using their CDR Arrangement ID, to the Participant DH to withdraw arrangement consent	POST
Pushed Authorisation	CTS Simulated ADR sends request object for request_uri to Participant DH via Participant DH's Pushed Authorisation Endpoint	POST

Link to specifications

https://consumerdatastandardsaustralia.github.io/standards-archives/standards-1.18.0/#get-customer

9.4 Scenario Results

Pass

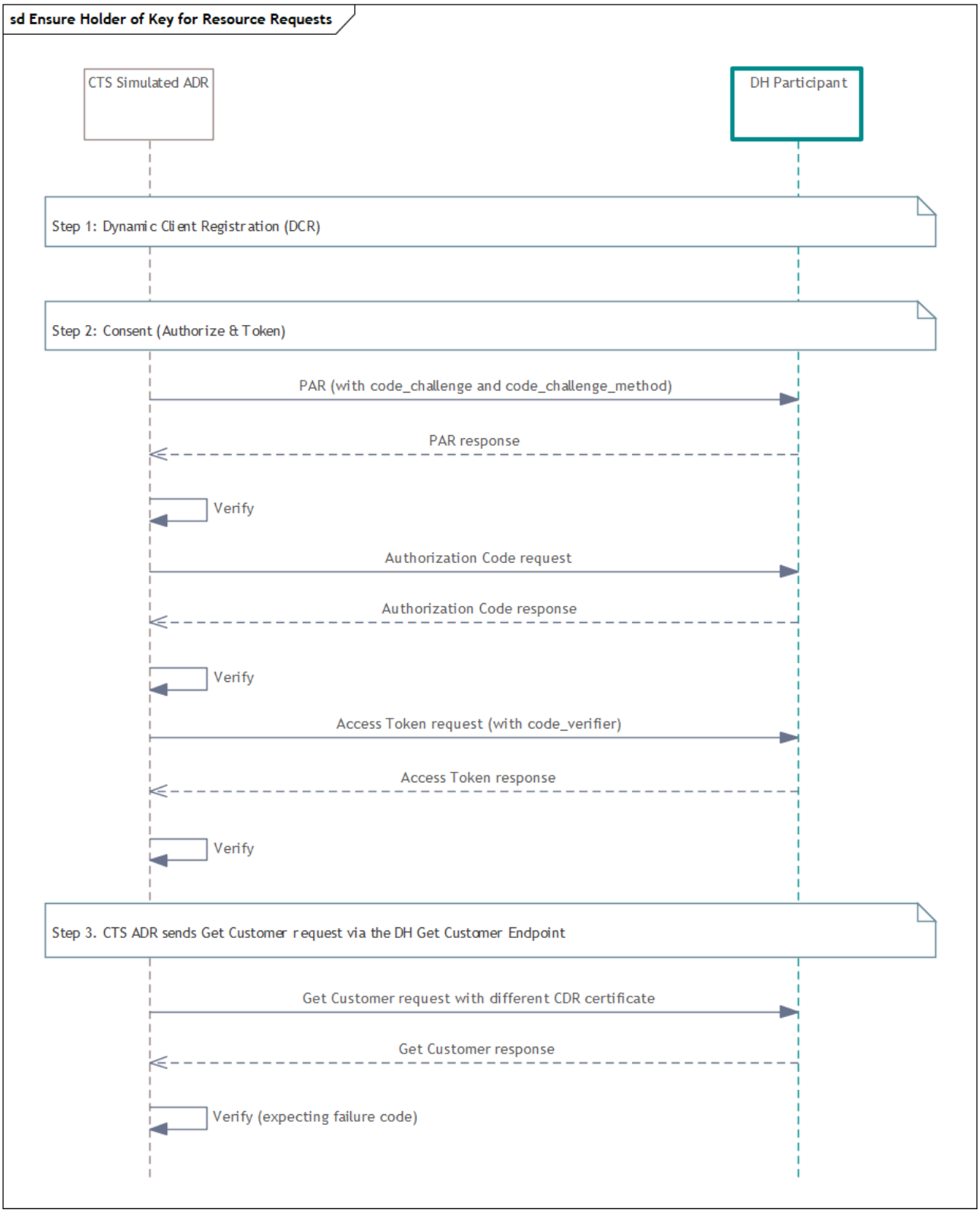
You have passed the scenario when the Get Customer request is sent with a CDR certificate that is not bound to the access token in the request and the response is an error code.

Fail

You have failed the scenario when the Get Customer request is sent with a CDR certificate that is not bound to the access token in the request and the response returns the request of the accounts and/or it **does not** return an error code.

9.5 Scenario High Level Test Steps

1. Dynamic Client Registration (DCR) *(Steps 1-4 from DCR scenario)*
2. Consent (Authorise & Token) *(Steps 2, 3 & 4 from Concurrent Consent scenario)*
3. **CTS ADR sends Get Customer request via the DH Get Customer Endpoint**
    a. CTS Simulated ADR sends a request to the Participant DH Get Customer Endpoint using the Participant DH issued Access Token and a different CDR certificate than the one used to request the access token.
    b. Participant DH validates the CTS Simulated ADR request and returns an error response of error `"invalid_token"` with Http status code of "`Unauthorised`".
    c. CTS verifies the Participant DH Get Customer Response.



Ensure Holder of Key for Resource Requests Scenario Sequence Diagram

10 Ensure Client Assertion Data in Token Request Scenario

10.1 Purpose

The ability for a Participant DH to demonstrate that they respond with appropriate bad request error to several poorly formed access token requests from CTS Simulated ADR.

10.2 Scenario Conditions

Not Applicable.

10.3 Endpoints

The table below lists endpoints specific to this scenario. See also Endpoints used in Data Holder Scenarios.

Endpoint	Description	Method
Get Customer	CTS Simulated ADR sends a request to the Participant DH's Get Customer Endpoint	GET
Authorize	CTS Simulated ADR requests with the Participant DH via the authorize Endpoint	GET
Token	CTS Simulated ADR exchanges their code for a Token from the Participant DH via the Token Endpoint	POST
Get Data Recipient Status	Participant DH requests the data recipient status from the CTS Simulated Register via the Get Data Recipient Status Endpoint	GET
Get Data Recipients	Participant DH requests the data recipients from the CTS Simulated Register via the Get Data Recipients Endpoint	GET
Get Software Product Status	Participant DH requests the software product status from the CTS Simulated Register via the Get Software Product Status Endpoint	GET

Pushed Authorisation	CTS Simulated ADR sends request object for request_uri to Participant DH via Participant DH's Pushed Authorisation Endpoint	POST

Link to specification

https://consumerdatastandardsaustralia.github.io/standards-archives/standards-1.18.0/#request-object

https://openid.net/specs/openid-connect-core-1_0.html#HybridFlowAuth

https://consumerdatastandardsaustralia.github.io/standards-archives/standards-1.18.0/#tokens

https://openid.net/specs/openid-connect-core-1_0.html#HybridIDToken

https://openid.net/specs/openid-connect-core-1_0.html#TokenEndpoint

10.4 Scenario Results

Pass

You have passed the Client Assertion Data in Token Request scenario when you handle each poorly formed request correctly.

Fail

You have failed the Client Assertion Data in Token Request scenario when you **do not** handle each poorly formed request test correctly.

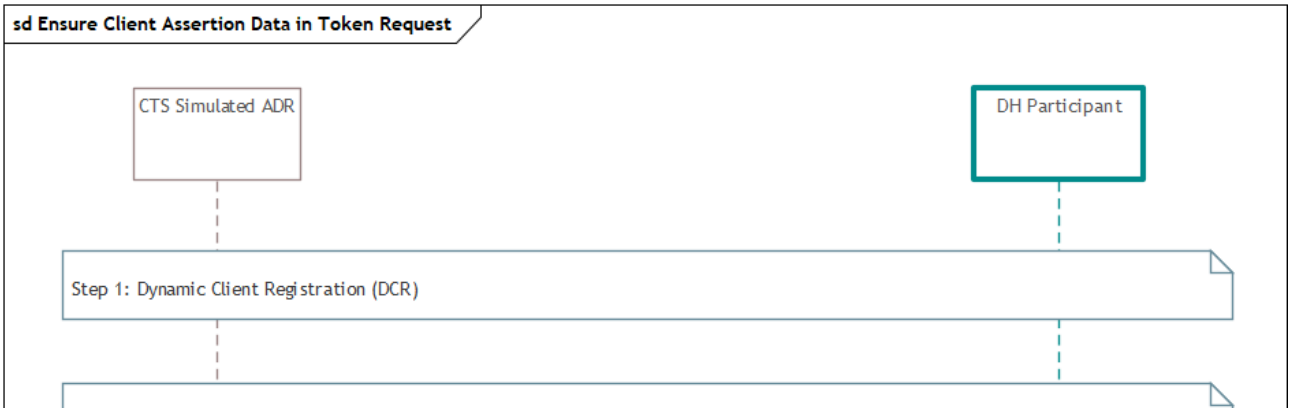
10.5 Scenario High Level Test Steps

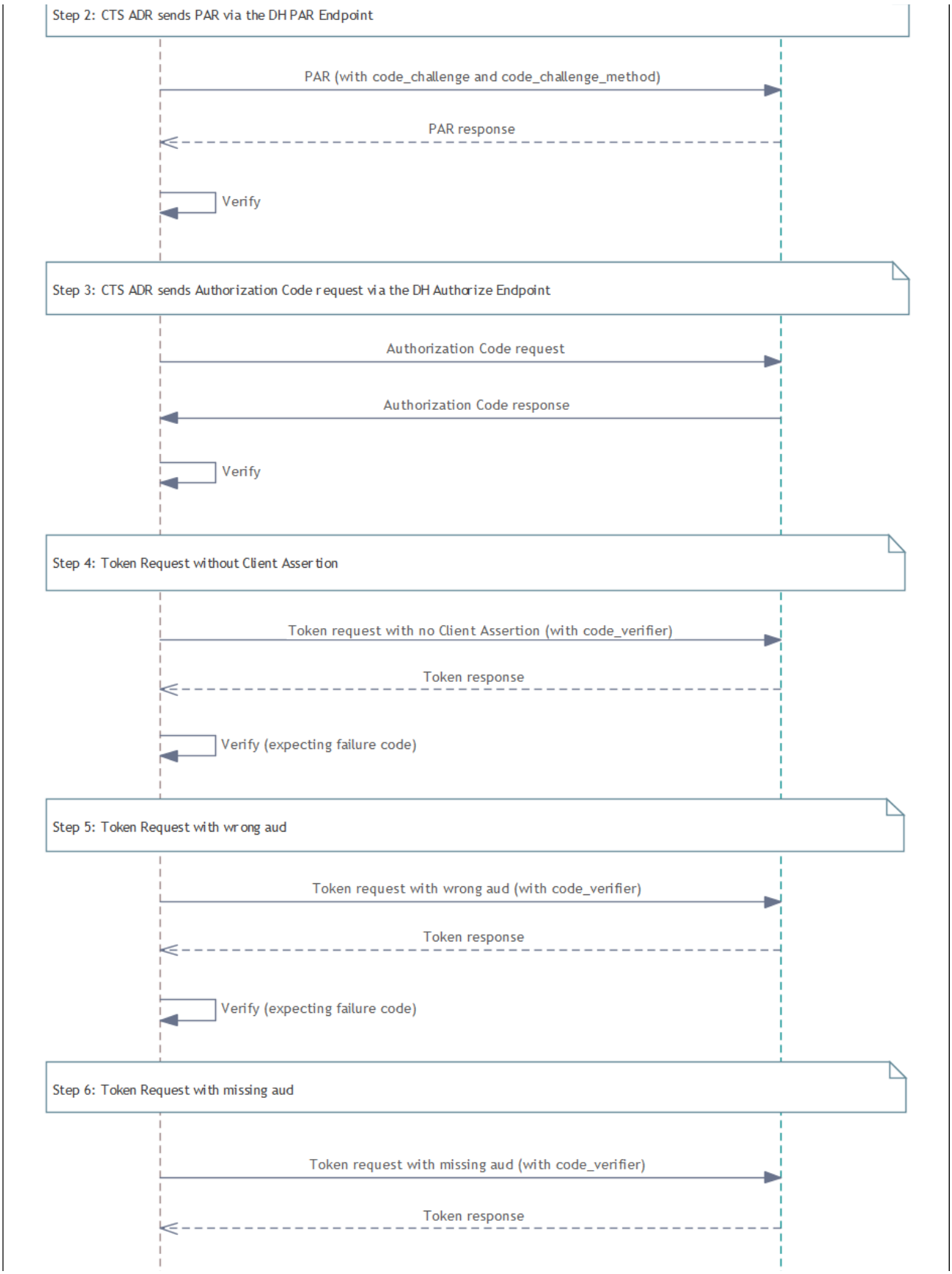
1. Dynamic Client Registration (DCR) *(Steps 1-4 from DCR scenario)*
2. **CTS ADR sends Pushed Authorization Request (PAR) via the DH Pushed Authorisation endpoint**
    a. CTS ADR sends a Pushed Authorization Request with the request object to the DH via the Pushed Authorization endpoint
    b. DH validates the CTS ADR Pushed Authorization Request (PAR), and responds with request_uri
    c. CTS verifies that in the DH PAR response that request_uri is contained in the response.
3. **CTS ADR requests authorisation via the DH Authorise Endpoint**
    a. CTS Simulated ADR sends an Authorisation request to the DH via the Authorise endpoint.
    b. DH validates the CTS Simulated ADR Authorisation request, verifying that the ADR Software Product is registered with the DH and responds via the Redirect URI with Authorisation Code, State, and encrypted and signed ID Token.
    c. CTS verifies the DH Authorisation response.
4. **CTS ADR sends Token request via the DH Token Endpoint**

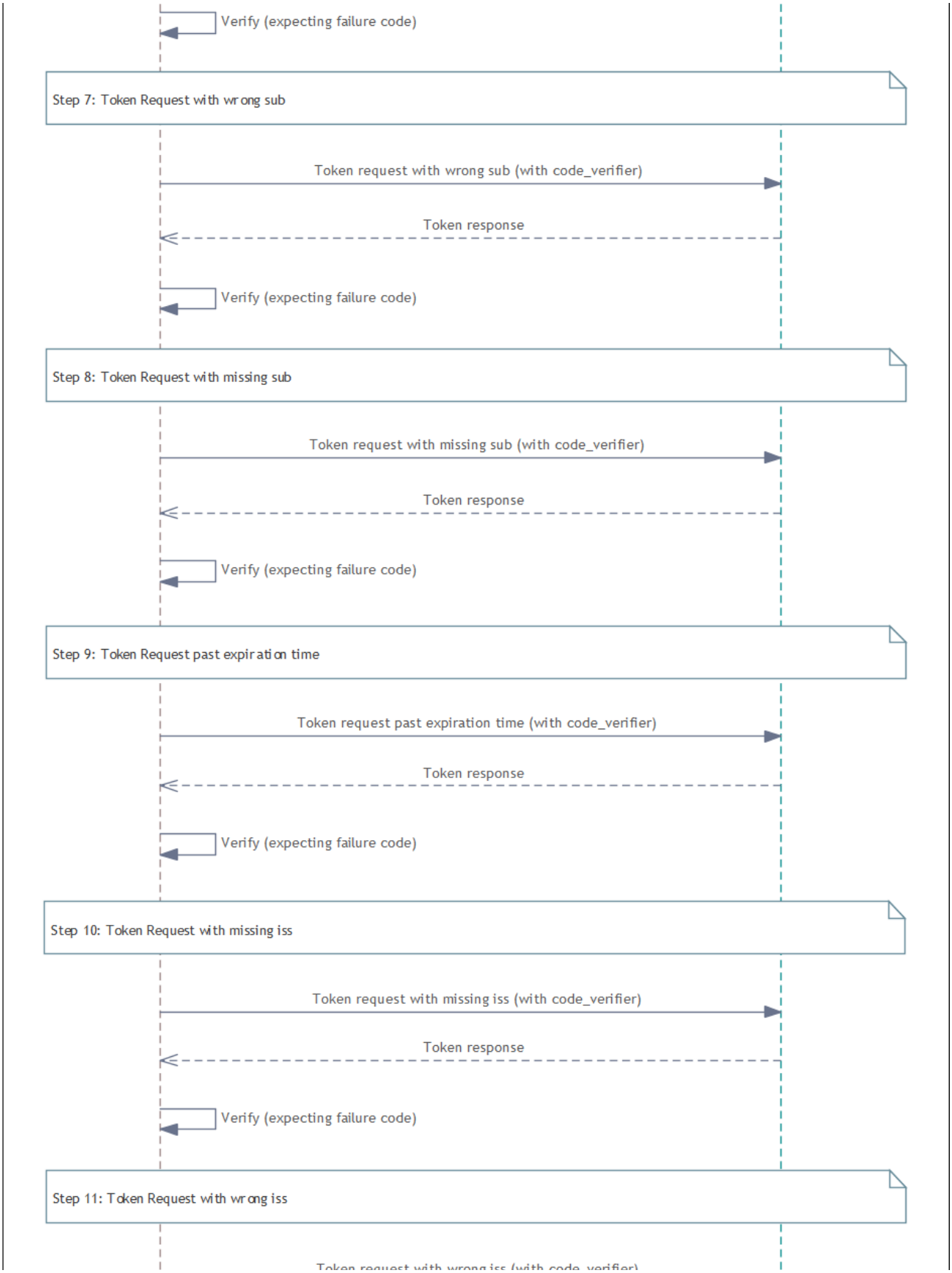
   a. CTS Simulated ADR sends a Token request **without a client assertion** to the DH via the Token Endpoint, exchanging their Authorisation code for a Token.
   b. DH validates the CTS ADR Token request and returns a response.
   c. CTS verifies the DH Token response with error "`invalid_client`" [bad request].
5. **CTS ADR sends Token request via the DH Token Endpoint**
   a. CTS Simulated ADR sends a Token request with a **wrong 'aud'** in client assertion to the DH via the Token Endpoint, exchanging their Authorisation code for a Token.
   b. DH validates the CTS ADR Token request and returns a response.
   c. CTS verifies the DH Token response with error "`invalid_client`" [bad request].
6. **CTS ADR sends Token request via the DH Token Endpoint**
   a. CTS Simulated ADR sends a Token request **missing 'aud'** in client assertion to the DH via the Token Endpoint, exchanging their Authorisation code for a Token.
   b. DH validates the CTS ADR Token request and returns a response.
   c. CTS verifies the DH Token response with error "`invalid_client`" [bad request].
7. **CTS ADR sends Token request via the DH Token Endpoint**
   a. CTS Simulated ADR sends a Token request **wrong 'sub'** in client assertion to the DH via the Token Endpoint, exchanging their Authorisation code for a Token.
   b. DH validates the CTS ADR Token request and returns a response.
   c. CTS verifies the DH Token response with error "`invalid_client`" [bad request].
8. **CTS ADR sends Token request via the DH Token Endpoint**
   a. CTS Simulated ADR sends a Token request with a **missing 'sub'** in client assertion to the DH via the Token Endpoint, exchanging their Authorisation code for a Token.
   b. DH validates the CTS ADR Token request and returns a response.
   c. CTS verifies the DH Token response with error "`invalid_client`" [bad request].
9. **CTS ADR sends Token request via the DH Token Endpoint**
   a. CTS Simulated ADR sends a Token request with **an expired** client assertion to the DH via the Token Endpoint, exchanging their Authorisation code for a Token.
   b. DH validates the CTS ADR Token request and returns a response.
   c. CTS verifies the DH Token response with error "`invalid_client`" [bad request].
10. **CTS ADR sends Token request via the DH Token Endpoint**
   a. CTS Simulated ADR sends a Token request with a **missing 'iss'** in client assertion to the DH via the Token Endpoint, exchanging their Authorisation code for a Token.
   b. DH validates the CTS ADR Token request and returns a response.
   c. CTS verifies the DH Token response with error "`invalid_client`" [bad request].
11. **CTS ADR sends Token request via the DH Token Endpoint**
   a. CTS Simulated ADR sends a Token request with a **wrong 'iss'** in client assertion to the DH via the Token Endpoint, exchanging their Authorisation code for a Token.
   b. DH validates the CTS ADR Token request and returns a response.
   c. CTS verifies the DH Token response with error "`invalid_client`" [bad request].
12. **CTS ADR sends Token request via the DH Token Endpoint**
   a. CTS Simulated ADR sends a Token request with a **missing 'alg'** in client assertion to the DH via the Token Endpoint, exchanging their Authorisation code for a Token.

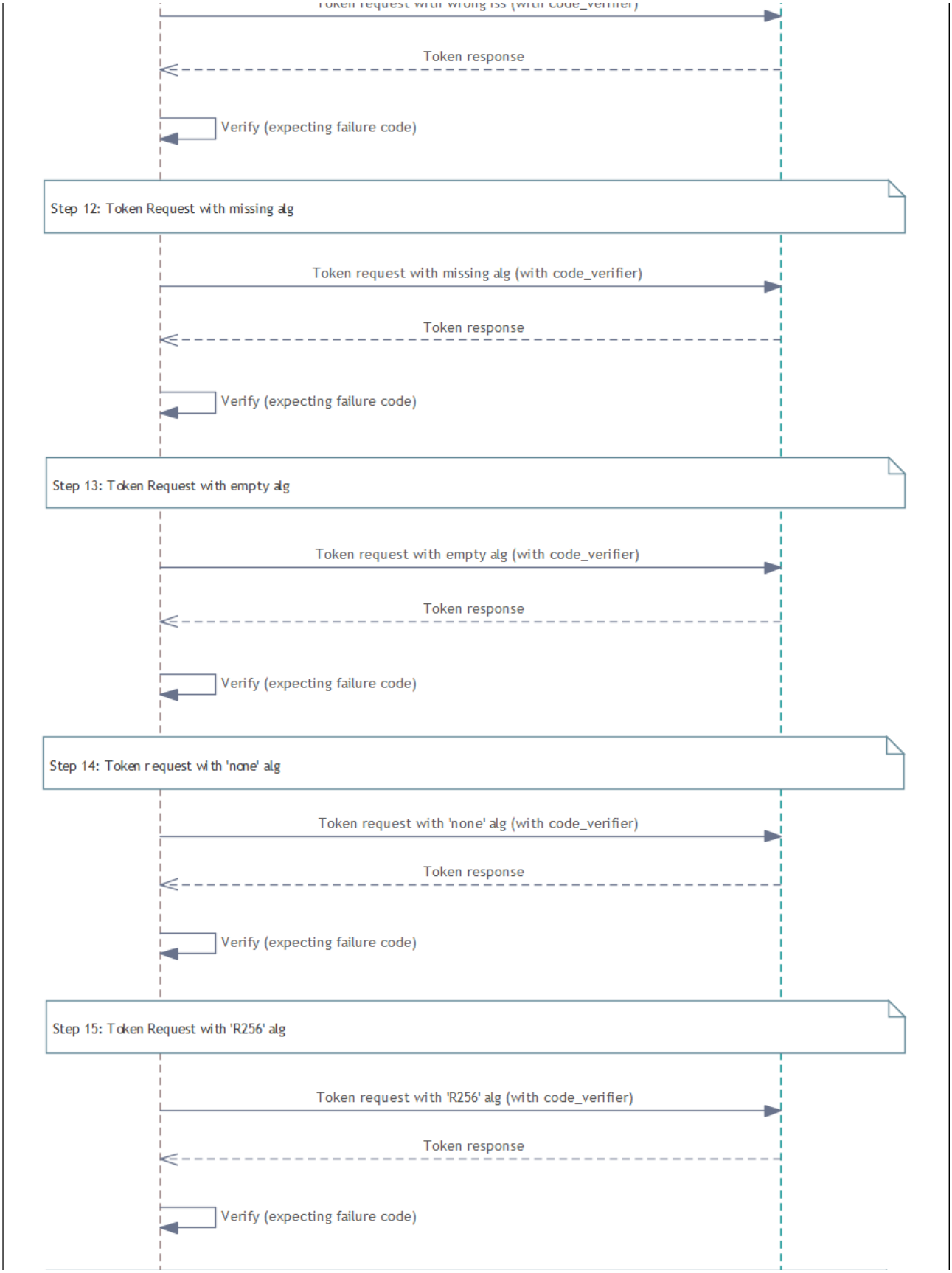
b. DH validates the CTS ADR Token request and returns a response.
c. CTS verifies the DH Token response with error "`invalid_client`" [bad request].

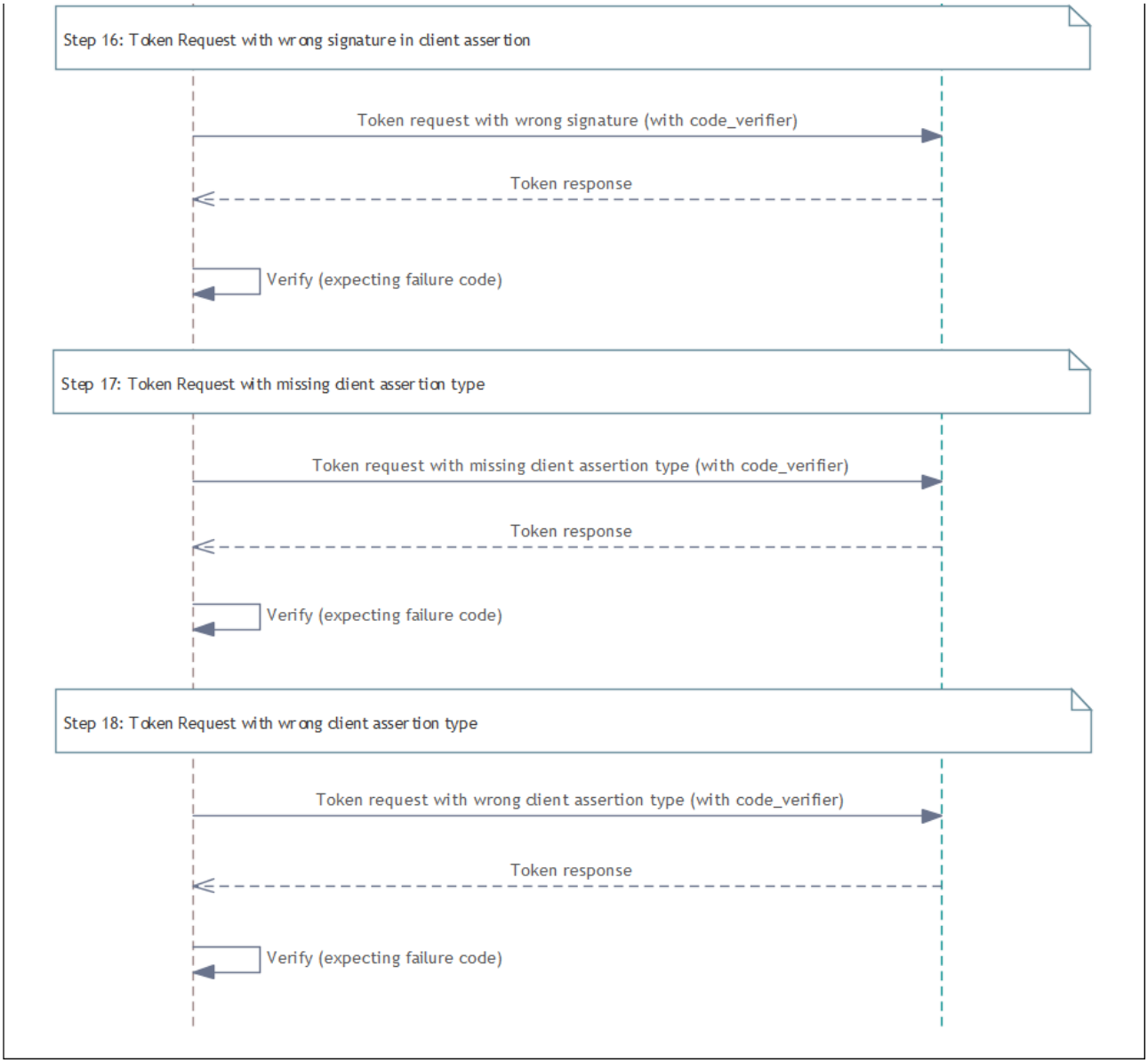
13. **CTS ADR sends Token request via the DH Token Endpoint**
    a. CTS Simulated ADR sends a Token request with an **empty 'alg'** in client assertion to the DH via the Token Endpoint, exchanging their Authorisation code for a Token.
    b. DH validates the CTS ADR Token request and returns a response.
    c. CTS verifies the DH Token response with error "`invalid_client`" [bad request].
14. **CTS ADR sends Token request via the DH Token Endpoint**
    a. CTS Simulated ADR sends a Token request with a **'none' 'alg'** in client assertion to the DH via the Token Endpoint, exchanging their Authorisation code for a Token.
    b. DH validates the CTS ADR Token request and returns a response.
    c. CTS verifies the DH Token response with error "`invalid_client`" [bad request].
15. **CTS ADR sends Token request via the DH Token Endpoint**
    a. CTS Simulated ADR sends a Token request with a **'RS256' 'alg'** in client assertion to the DH via the Token Endpoint, exchanging their Authorisation code for a Token.
    b. DH validates the CTS ADR Token request and returns a response.
    c. CTS verifies the DH Token response with error "`invalid_client`" [bad request].
16. **CTS ADR sends Token request via the DH Token Endpoint**
    a. CTS Simulated ADR sends a Token request with a **wrong signature** in client assertion to the DH via the Token Endpoint, exchanging their Authorisation code for a Token.
    b. DH validates the CTS ADR Token request and returns a response.
    c. CTS verifies the DH Token response with error "`invalid_client`" [bad request].
17. **CTS ADR sends Token request via the DH Token Endpoint**
    a. CTS Simulated ADR sends a Token request with a **missing 'client_assertion_type'** to the DH via the Token Endpoint, exchanging their Authorisation code for a Token.
    b. DH validates the CTS ADR Token request and returns a response.
    c. CTS verifies the DH Token response with error "`invalid_client`" [bad request].
18. **CTS ADR sends Token request via the DH Token Endpoint**
    a. CTS Simulated ADR sends a Token request with a **wrong 'client_assertion_type'** to the DH via the Token Endpoint, exchanging their Authorisation code for a Token.
    b. DH validates the CTS ADR Token request and returns a response.
    c. CTS verifies the DH Token response with error "`invalid_client`" [bad request].




Step 2: CTS ADR sends PAR via the DH PAR Endpoint
PAR (with code_challenge and code_challenge_method)
PAR response
Verify
Step 3: CTS ADR sends Authorization Code request via the DH Authorize Endpoint
Authorization Code request
Authorization Code response
Verify
Step 4: Token Request without Client Assertion
Token request with no Client Assertion (with code_verifier)
Token response
Verify (expecting failure code)
Step 5: Token Request with wrong aud
Token request with wrong aud (with code_verifier)
Token response
Verify (expecting failure code)
Step 6: Token Request with missing aud
Token request with missing aud (with code_verifier)
Token response


Verify (expecting failure code)
Step 7: Token Request with wrong sub
Token request with wrong sub (with code_verifier)
Token response
Verify (expecting failure code)
Step 8: Token Request with missing sub
Token request with missing sub (with code_verifier)
Token response
Verify (expecting failure code)
Step 9: Token Request past expiration time
Token request past expiration time (with code_verifier)
Token response
Verify (expecting failure code)
Step 10: Token Request with missing iss
Token request with missing iss (with code_verifier)
Token response
Verify (expecting failure code)
Step 11: Token Request with wrong iss


Token request with wrong iss (with code_verifier)
Token response
Verify (expecting failure code)
Step 12: Token Request with missing alg
Token request with missing alg (with code_verifier)
Token response
Verify (expecting failure code)
Step 13: Token Request with empty alg
Token request with empty alg (with code_verifier)
Token response
Verify (expecting failure code)
Step 14: Token request with 'none' alg
Token request with 'none' alg (with code_verifier)
Token response
Verify (expecting failure code)
Step 15: Token Request with 'R256' alg
Token request with 'R256' alg (with code_verifier)
Token response
Verify (expecting failure code)



Ensure Client Assertion Data in Token Request Scenario Sequence Diagram

11 Amending Existing Consent Scenario

11.1 Purpose

The ability of a Participant DH to replace an existing consent arrangement and correctly respond to a token request with an invalid refresh token.

11.2 Scenario Conditions

Not Applicable.

11.3 Endpoints

The table below lists endpoints specific to this scenario. See also Endpoints used in Data Holder Scenarios.

Endpoint	Description	Method
Pushed Authorisation	CTS Simulated ADR sends a Pushed Authorisation Request object for request_uri to the Participant DH via Pushed Authorisation Endpoint	POST
Get Customer	CTS Simulated ADR sends a request to the Participant DH's Get Customer Endpoint	GET
Authorize	CTS Simulated ADR requests with the Participant DH via the authorize Endpoint	GET
Token	CTS Simulated ADR exchanges their code for a Token from the Participant DH via the Token Endpoint	POST

Link to specifications

https://consumerdatastandardsaustralia.github.io/standards-archives/standards-1.18.0/#introduction

11.4 Scenario Results

Pass

You have passed the PAR flow when you **can**:

- Return a valid response when the CTS Simulated ADR sends you a PAR.
- Return a payload when you receive a call from the CTS Simulated ADR to the Customer API using the established consent.
- Return a response with an error code when the CTS Simulated ADR calls your Token endpoint with a Refresh Token from the initial consent arrangement.

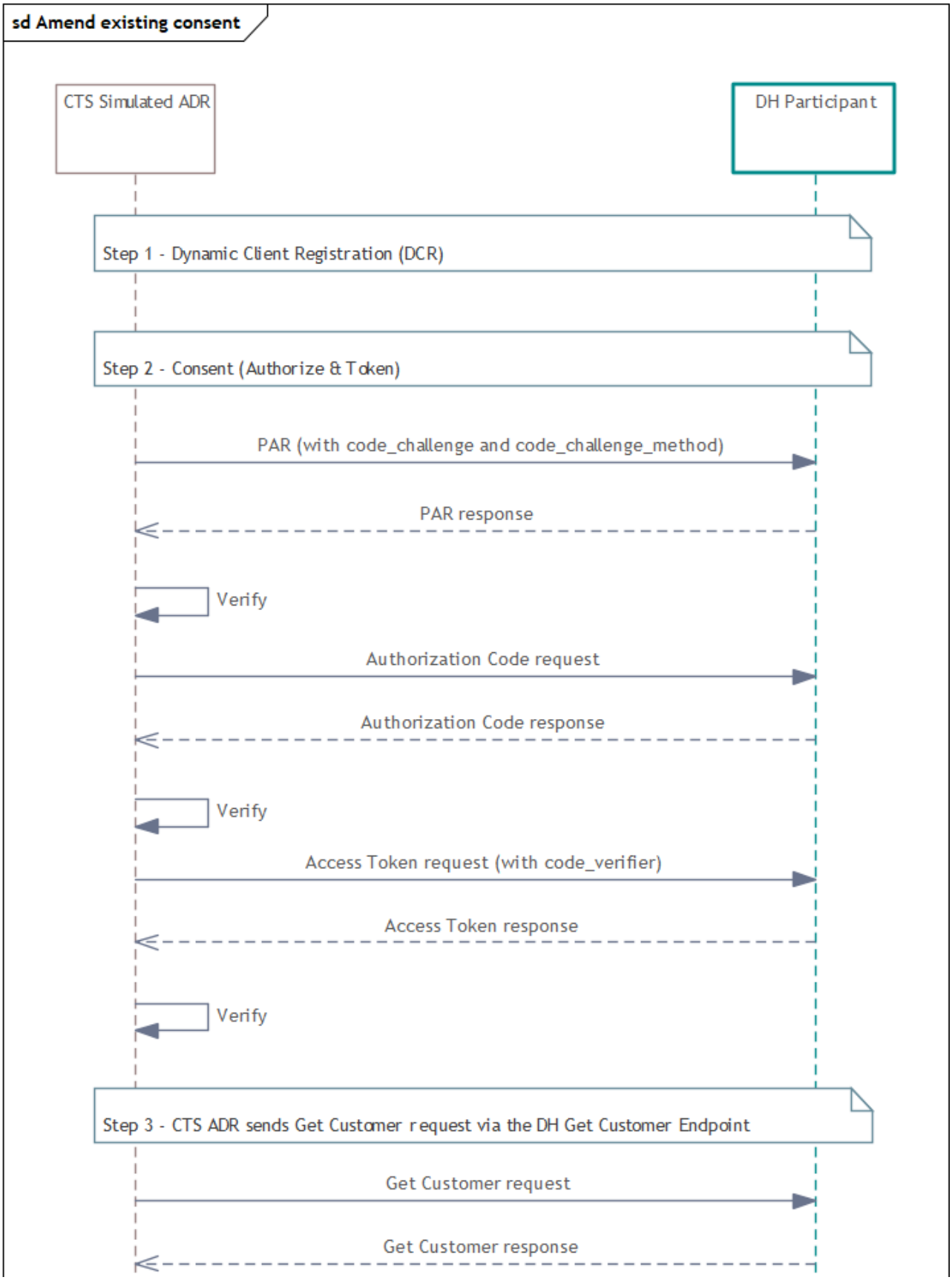
Fail

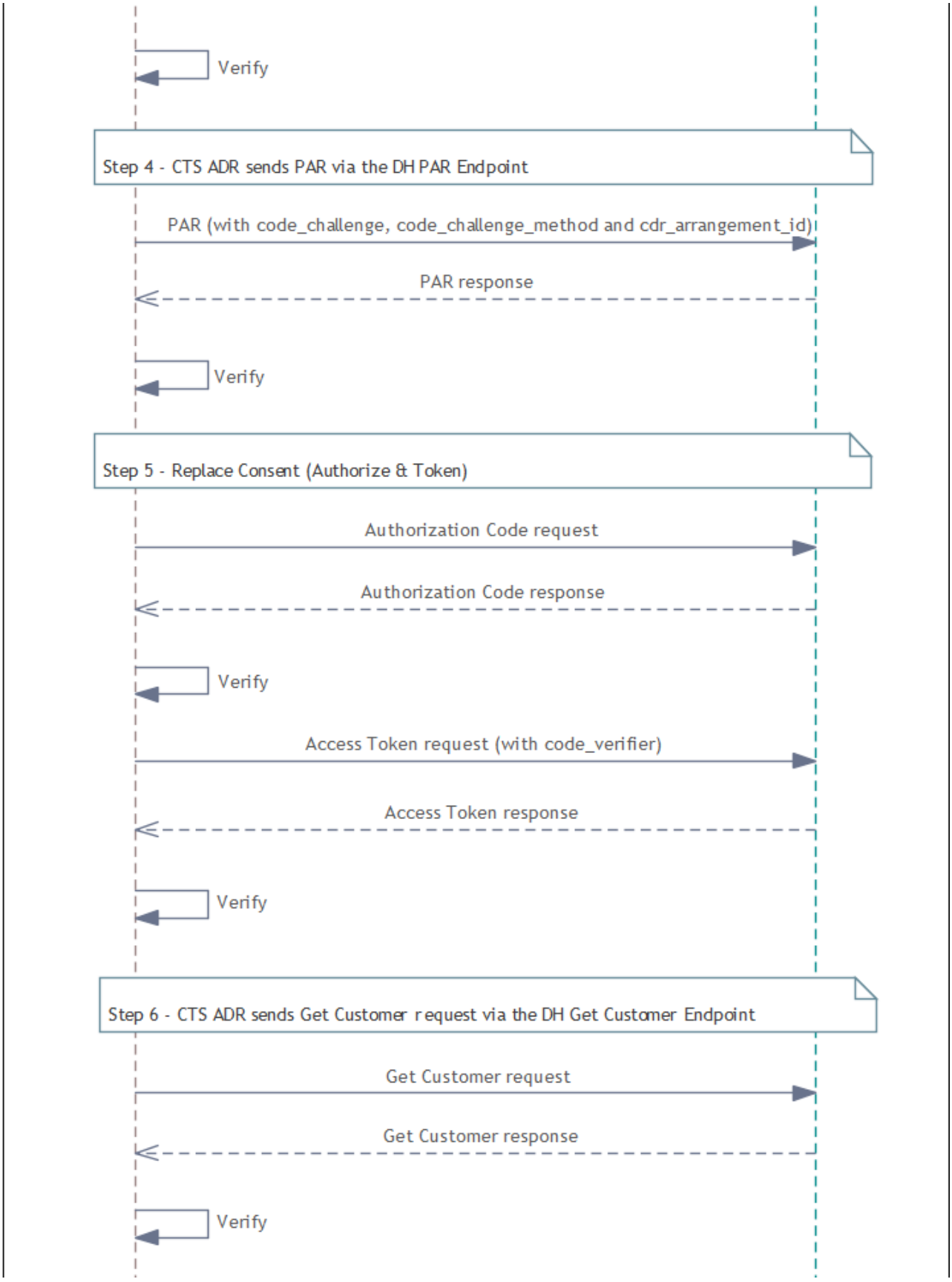
You have failed the PAR flow when you **cannot**:

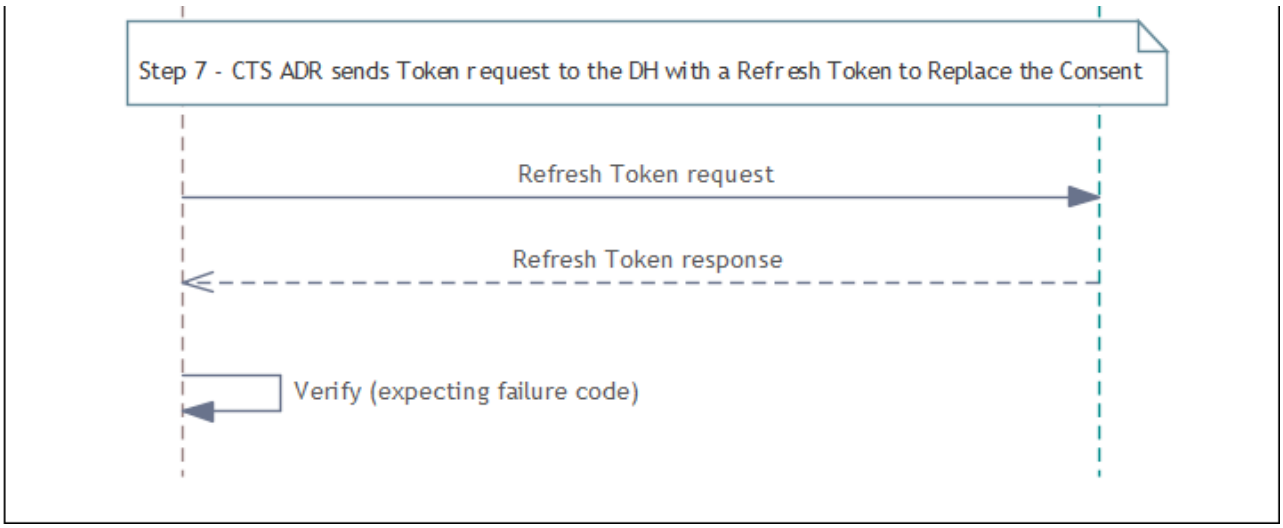
- Return a valid response when the CTS Simulated ADR sends you a PAR.
- Return a payload when you receive a call from the CTS Simulated ADR to the Customer API using the established consent.
- Return a response with an error code when the CTS Simulated ADR calls your Token endpoint with a Refresh Token from the initial consent arrangement in an attempt to exchange for a new access token.

11.5 Scenario High Level Test Steps

1. Dynamic Client Registration (DCR) *(Steps 1- 4 from DCR scenario)*
2. Consent (Authorise & Token) *(Steps 2 - 4 from Concurrent Consent scenario)*
3. **CTS ADR sends request via the DH Get Customer Endpoint**
    a. CTS Simulated ADR sends a request, using the Participant DH issued Access Token, to the Participant DH via the Get Customer Endpoint.
    b. Participant DH validates the CTS Simulated ADR request and returns a response with the Get Customer payload.
    c. CTS verifies the Participant DH Get Customer response.
4. **CTS ADR sends a second Pushed Authorization Request (PAR) via the DH Pushed Authorisation Endpoint**
    a. CTS Simulated ADR sends a PAR, with Client Authentication and an additional claim for cdr_arragement_id from the initial consent, to the Participant DH via the Pushed Authorisation Endpoint.
    b. Participant DH validates the PAR.
    c. Participant DH returns a response to the PAR with a 'RequestURI' and 'ExpiresIn'.
    d. CTS verifies the Participant DH PAR response.
5. Replace consent (Authorise & Token) *(Steps 3 - 4 from Concurrent Consent scenario)*
6. **CTS ADR sends a request to the DH Get Customer Endpoint**
    a. CTS Simulated ADR sends a request, using the Participant DH issued Access Token, to the Participant DH via the Get Customer Endpoint.
    b. Participant DH validates the CTS Simulated ADR request and returns a response with the Customer payload.
    c. CTS verifies the Participant DH Get Customer Response.
7. **CTS ADR sends a Token request to the DH with a Refresh Token to replace the Consent**
    a. CTS Simulated ADR sends a Token request to the Participant DH via the Token Endpoint, exchanging their Refresh Token of the original consent request for an Access Token.
    b. Participant DH validates the CTS Simulated ADR Refresh Token request and returns a response.
    c. CTS verifies the Participant DH Token response [bad request].


sd Amend existing consent
CTS Simulated ADR
DH Participant
Step 1 - Dynamic Client Registration (DCR)
Step 2 - Consent (Authorize & Token)
PAR (with code_challenge and code_challenge_method)
PAR response
Verify
Authorization Code request
Authorization Code response
Verify
Access Token request (with code_verifier)
Access Token response
Verify
Step 3 - CTS ADR sends Get Customer request via the DH Get Customer Endpoint
Get Customer request
Get Customer response


Verify
Step 4 - CTS ADR sends PAR via the DH PAR Endpoint
PAR (with code_challenge, code_challenge_method and cdr_arrangement_id)
PAR response
Verify
Step 5 - Replace Consent (Authorize & Token)
Authorization Code request
Authorization Code response
Verify
Access Token request (with code_verifier)
Access Token response
Verify
Step 6 - CTS ADR sends Get Customer request via the DH Get Customer Endpoint
Get Customer request
Get Customer response
Verify



Amending Existing Consent Scenario Sequence Diagram

12 Removed Software Product Scenario

12.1 Purpose

This scenario tests that a Participant DH fulfills its responsibilities when an ADR's Software Product status changes to `Removed`.

12.1.1 Business Context

The ACCC Registrar can change the status of a software product independently of the ADR accreditation status. Therefore, DHs are required to be able to react to Software Statuses changes within 5 minutes of the change occurring on the CDR Register.

As per the CDS, when a Software Product changes to `Removed`, a DH should:

1. Must not continue to disclose CDR Data,
2. Must not continue to facilitate consent authorisation,
3. Must invalidate consents, and
4. Cleanup Registrations.

This test scenario seeks to validate that a Participant DH is checking the status of an ADR's Software Product prior to the disclosure of data. For example, if a request is received, where the ADR software product has a status of 'Removed', then data is not disclosed by the DH.

For more information on software, statuses refer to https://consumerdatastandardsaustralia.github.io/standards-archives/standards-1.18.0/#participant-statuses

12.2 Scenario Conditions

Not Applicable.

12.3 Endpoints

The table below lists endpoints specific to this scenario. See also Endpoints used in Data Holder Scenarios.

Endpoint	Description	Method
Get Customer	CTS Simulated ADR sends a request to the Participant DH's Get Customer Endpoint	GET
Get Data Recipient Status	Participant DH requests the data recipient status from the CTS Simulated Register via the Get Data Recipient Status Endpoint	GET

Get Data Recipients	Participant DH requests the data recipients from the CTS Simulated Register via the Get Data Recipients Endpoint	GET
Get Software Product Status	Participant DH requests the software product status from the CTS Simulated Register via the Get Software Product Status Endpoint	GET
Arrangement Revocation **DR to DH**	CTS Simulated ADR sends a request, using their CDR Arrangement ID, to the Participant DH to withdraw arrangement consent	POST
Pushed Authorisation	CTS Simulated ADR sends a Pushed Authorisation Request object for request_uri to the Participant DH via Pushed Authorisation Endpoint	POST
Authorize	CTS Simulated ADR requests with the Participant DH via the Authorize Endpoint	GET
Token	CTS Simulated ADR exchanges their code for a Token from the Participant DH via the Token Endpoint	POST

Link to specifications

https://consumerdatastandardsaustralia.github.io/standards-archives/standards-1.18.0/#introduction

12.4 Scenario Results

Pass

You have passed the Register Status tests when an ADR Software Product status is 'Removed' and you:

- do not disclose CDR data
- do not facilitate consent authorisation
- do not facilitate consent withdrawal.

Fail

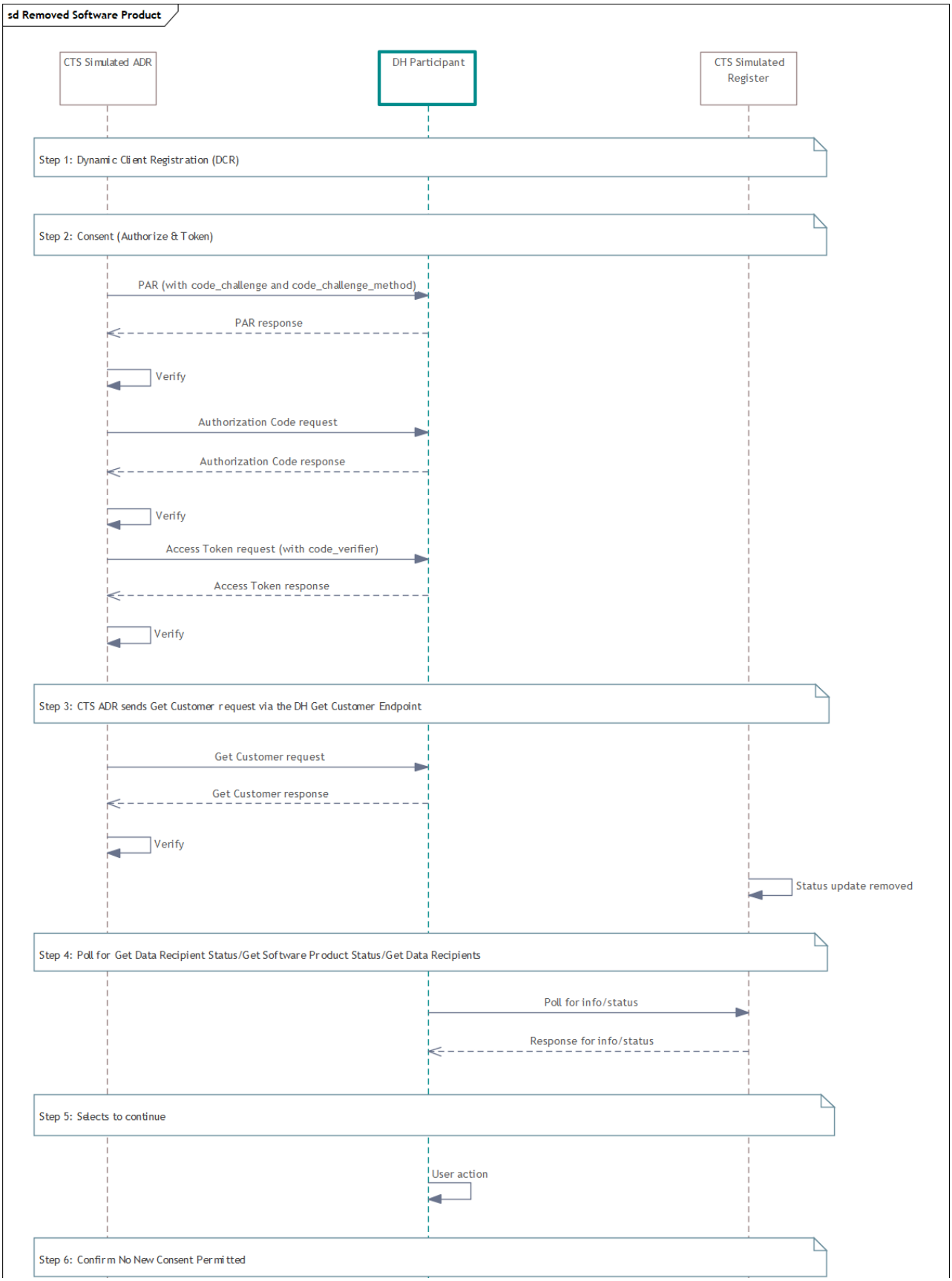
You have failed the Register Status tests when an ADR Software Product status is 'Removed' and you:

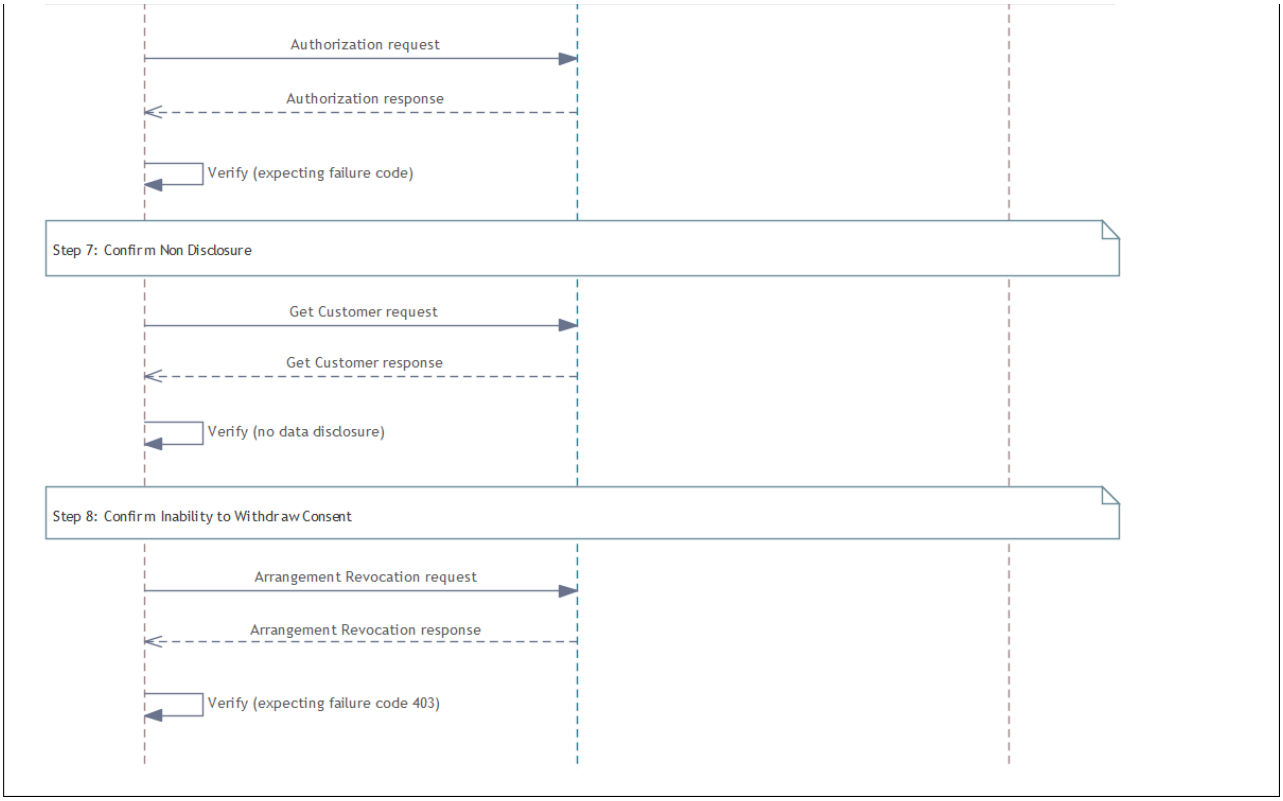
- disclose CDR data
- facilitate consent authorisation

- facilitate consent withdrawal.

12.5 Scenario High-Level Test Steps

1. Dynamic Client Registration (DCR) *(Steps 1-4 from DCR scenario)*
2. Consent (Authorise & Token) *(Steps 2, 3 & 4 from Concurrent Consent scenario)*
3. **CTS ADR sends Get Customer request via the DH Get Customer Endpoint**
    a. CTS simulated ADR sends a request, using the DH issued Access Token, to the DH via the Get Customer Endpoint.
    b. DH validates the CTS simulated ADR request and returns a response.
    c. CTS verifies the DH Get Customer API response payload.
    d. CTS changes the ADR Software Product status in the CTS simulated Register from 'Active' to 'Removed'.
4. **DH polls the CTS Register to Get data recipient status**
    a. Participant DH sends a 'data recipient status' request to the CTS Simulated Register via the Get Data Recipient Statuses Endpoint.
    b. CTS Simulated Register returns a valid response and/or
    c. Participant DH sends a 'software product status' request to the CTS Simulated Register via the Get Software Product Statuses Endpoint.
    d. CTS Simulated Register returns a valid response and/or
    e. Participant DH sends a 'data recipients' request to the CTS Simulated Register via the Get Data Recipients Endpoint.
    f. CTS Simulated Register returns a valid response.
5. **DH selects to continue through the UI after at least 1 of the above Register APIs is polled**
6. **CTS confirms the inability to facilitate an authorisation**
    a. The CTS simulated ADR sends an Authorization Code Request to the DH Authorize Endpoint. Unlike the Authorization request in Step 2, this is done as a backchannel request from CTS to the DH Authorize Endpoint, with CTS impersonating a user agent. (Note - this is the only way CTS can verify that the error response is returned to the customer, rather than the CTS Simulated ADR).
    b. Participant DH validates the request and returns a failure code response. [HTTP Status code 403 - *for more information please see the technical note under the diagram below*].
    c. CTS verifies the response.
7. **CTS confirms non-disclosure of CDR data**
    a. CTS simulated ADR calls the DH Get Customer Endpoint to confirm that Participant DH does not disclose CDR data.
8. **CTS confirms inability to withdraw consent**
    a. CTS simulated ADR sends an Arrangement Revocation request to the DH Arrangements Revocation Endpoint.
    b. DH validates the request and returns a failure code response (HTTP Status code 403 or 422 – for more information please see the technical note below).
    c. CTS verifies the response.


sd Removed Software Product
CTS Simulated ADR
DH Participant
CTS Simulated Register
Step 1: Dynamic Client Registration (DCR)
Step 2: Consent (Authorize & Token)
PAR (with code_challenge and code_challenge_method)
PAR response
Verify
Authorization Code request
Authorization Code response
Verify
Access Token request (with code_verifier)
Access Token response
Verify
Step 3: CTS ADR sends Get Customer request via the DH Get Customer Endpoint
Get Customer request
Get Customer response
Verify
Status update removed
Step 4: Poll for Get Data Recipient Status/Get Software Product Status/Get Data Recipients
Poll for info/status
Response for info/status
Step 5: Selects to continue
User action
Step 6: Confirm No New Consent Permitted



Removed Software Product Scenario Sequence Diagram

12.5.1 Technical note

CTS will validate that the participant responses conform to CDS standards which would include validation of HTTP Status Codes, Error schema, and the Standard Error code itself to ensure that the Error code correlates to the specific failure condition.

The following CDS HTTP status codes and URNs will result in a pass:

HTTP status code	URN
403	urn:au-cds:error:cds-all:Authorisation/AdrStatusNotActive
403	urn:au-cds:error:cds-all:Authorisation/RevokedConsent
403	urn:au-cds:error:cds-all:Authorisation/InvalidConsent
422	urn:au-cds:error:cds-all:Authorisation/InvalidArrangement

13 Data Holder Initiated Revocation Scenario

13.1 Purpose

The ability for a Participant DH to validate that when a customer withdraws their consent from the Participant DH, the Participant DH handles the arrangement revocation correctly. To verify this, the participant DH will send an arrangement revocation to the CTS Simulated ADR.

13.2 Scenario Conditions

A cdr_arrangement_Id was issued to the CTS ADR by the DH.

13.3 Endpoints

The table below lists endpoints specific to this scenario. See also Endpoints used in Data Holder Scenarios.

Endpoint	Description	Method
Arrangement Revocation **DH to DR**	Participant DH sends a request, using their CDR arrangement ID, to the CTS Simulated ADR to withdraw an arrangement consent	POST
Pushed Authorisation	CTS Simulated ADR sends a Pushed Authorisation Request object for request_uri to the Participant DH via Pushed Authorisation Endpoint	POST

Link to specifications

https://consumerdatastandardsaustralia.github.io/standards-archives/standards-1.18.0/#introduction

13.4 Scenario Results

Pass

You have passed the withdrawal of consent flow when you call the CTS Simulated ADR Arrangement Revocation Endpoint and receive a success code response.

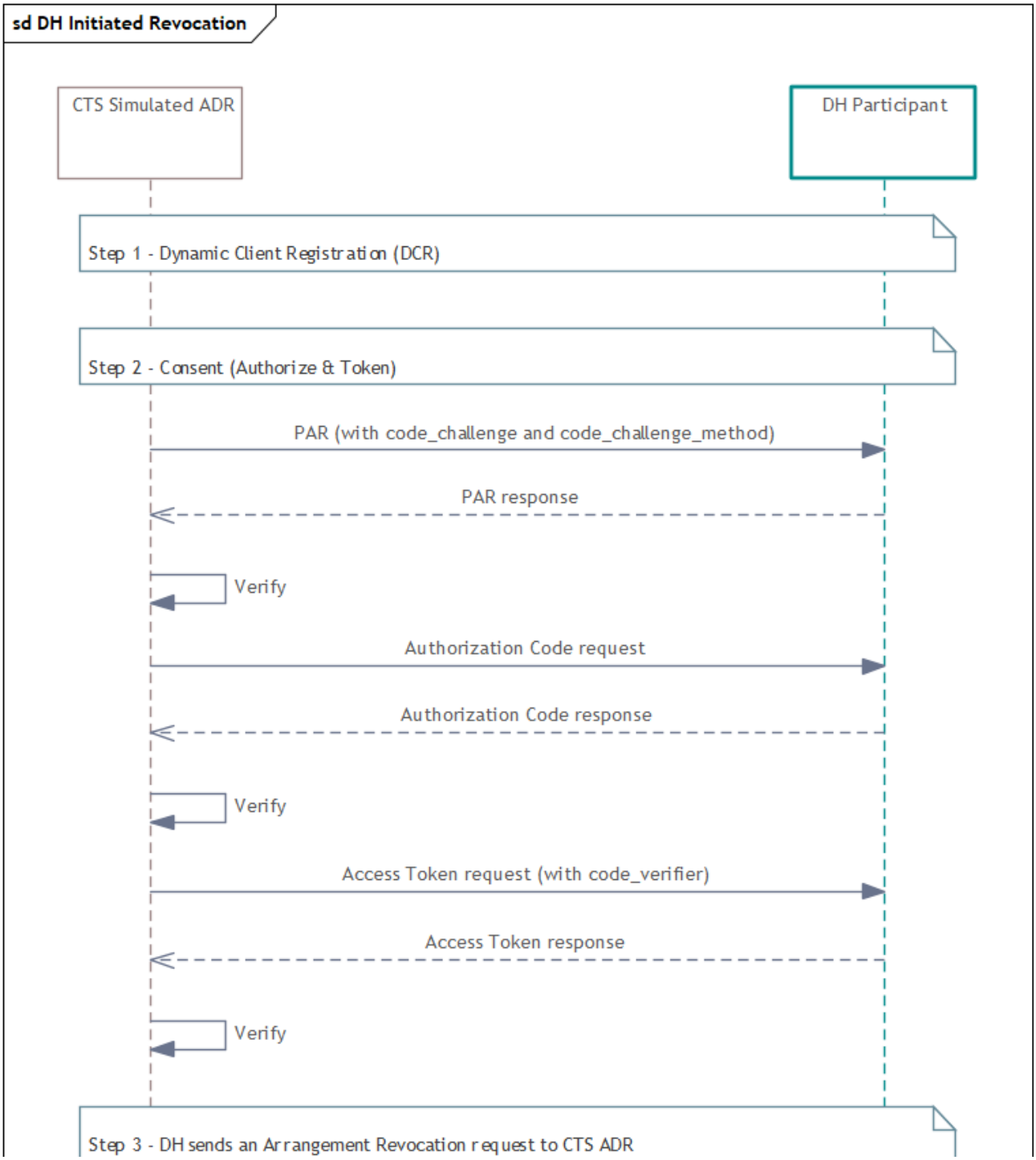
Fail

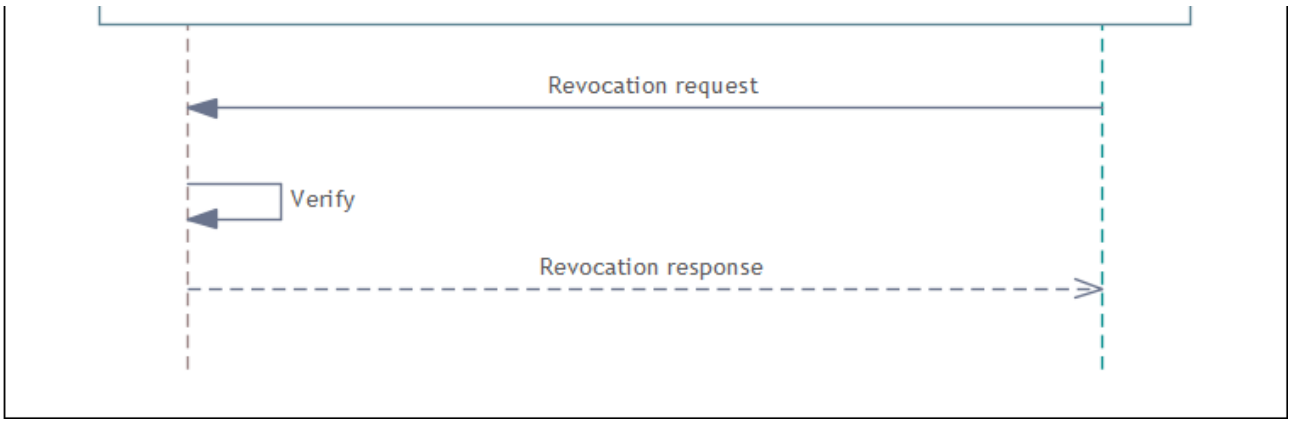
You have failed the withdrawal of consent flow when you call the CTS Simulated ADR Arrangement Revocation Endpoint and **do not** receive a success code response.

13.5 Scenario High Level Test Steps

1. Dynamic Client Registration (DCR) *(Steps 1-4 from DCR scenario)*

2. Consent (Authorise & Token) *(Steps 2, 3 & 4 from Concurrent Consent scenario)*
3. **DH sends an Arrangement Revocation request to the CTS ADR**
    a. Participant DH sends an Arrangement Revocation request with a cdr_arrangement_jwt containing the cdr_ arrangement_id generated by the initial consent arrangement.
    b. CTS Simulated ADR validates the request, invalidates the consent arrangement, and returns a success code response.





Data Holder Initiated Revocation Scenario Sequence Diagram

14 Data Recipient Initiated Revocation Scenario

14.1 Purpose

The ability for a Participant DH to validate that when a customer withdraws their consent from the Participant ADR, the Participant DH handles the arrangement revocation correctly. To confirm this, the participant DH will receive an arrangement revocation from the CTS Simulated ADR.

14.2 Scenario Conditions

A cdr_arrangement_id was issued to the CTS Simulated ADR by the Participant DH.

14.3 Endpoints

The table below lists endpoints specific to this scenario. See also Endpoints used in Data Holder Scenarios.

Endpoint	Description	Method
Arrangement Revocation **DR to DH**	CTS Simulated ADR sends a request, using their CDR Arrangement ID, to the DH to withdraw arrangement consent	POST
Pushed Authorisation	CTS Simulated ADR sends a Pushed Authorisation Request object for request_uri to the Participant DH via Pushed Authorisation Endpoint	POST

Link to specifications

https://consumerdatastandardsaustralia.github.io/standards-archives/standards-1.18.0/#security-endpoints

14.4 Scenario Results

Pass

You have passed the withdrawal of consent flow when the CTS Simulated ADR calls your Arrangement Revocation Endpoint and receives a success code response.

Fail

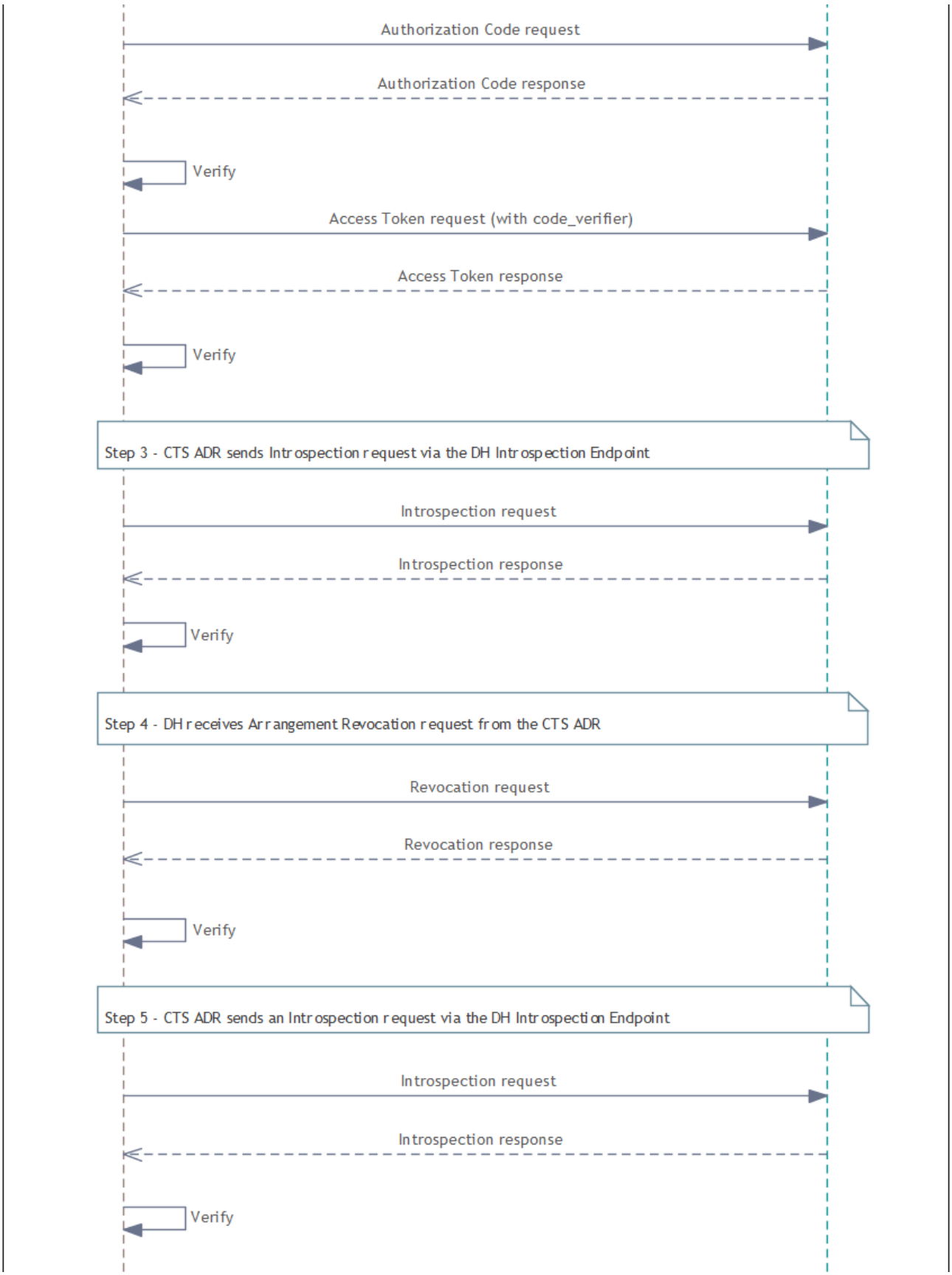
You have failed the withdrawal of consent flow when the CTS Simulated ADR calls your Arrangement Revocation Endpoint and **does not** receive a success code response.

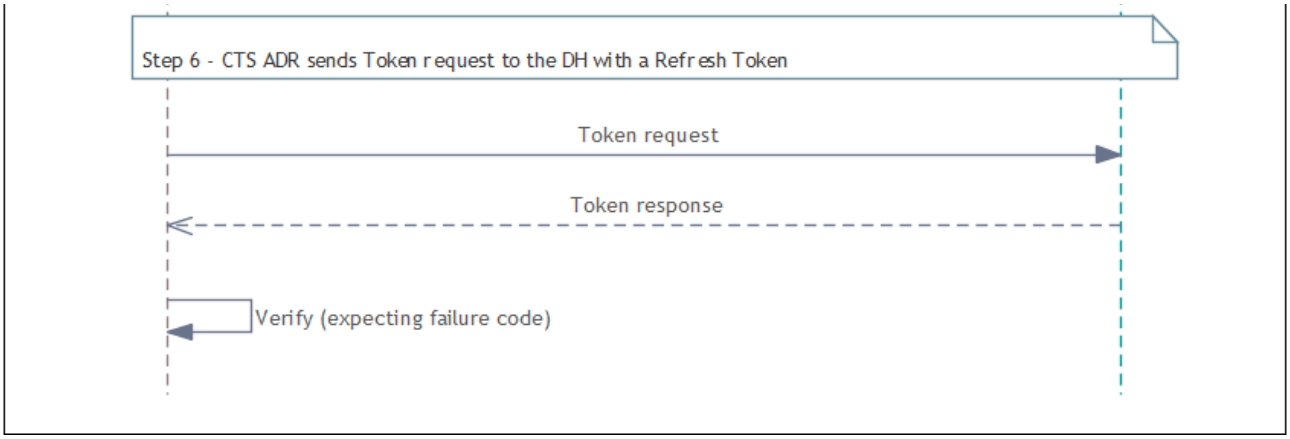
14.5 Scenario High Level Test Steps

1. Dynamic Client Registration (DCR) *(Steps 1-4 from DCR scenario)*

2. Consent (Authorise & Token) *(Steps 2, 3 & 4 from Concurrent Consent Scenario)*
3. **CTS ADR sends an Introspection request via the DH Introspection Endpoint**
    a. CTS Simulated ADR sends an Introspection request to the Participant DH Token Introspection endpoint.
    b. Participant DH validates the CTS Simulated ADR Introspection request and returns a response.
    c. CTS verifies the Participant DH Token Introspection response.
4. **DH receives Arrangement Revocation request from the CTS ADR**
    a. CTS Simulated ADR sends an Arrangement Revocation request to the Participant DH's CDR Arrangement Revocation Endpoint (registered uri).
    b. Participant DH validates the request and returns a success code response.
    c. CTS Simulated ADR verifies the Participant DH Arrangement Revocation response code.
5. **CTS ADR sends Introspection request via the DH Introspection Endpoint**
    a. CTS Simulated ADR sends an Introspection request to the Participant DH Token Introspection endpoint.
    b. Participant DH validates the CTS Simulated ADR Introspection request and returns a response.
    c. CTS verifies the Participant DH Token Introspection response.
6. **CTS ADR sends Token request to the DH with a Refresh Token**
    a. CTS Simulated ADR sends a Token request to the Participant DH via the Token Endpoint, exchanging their Refresh Token for an Access Token.
    b. Participant DH validates the CTS Simulated ADR Refresh Token request and returns a response

CTS verifies the Participant DH Token response [bad request].


Authorization Code request
Authorization Code response
Verify
Access Token request (with code_verifier)
Access Token response
Verify
Step 3 - CTS ADR sends Introspection request via the DH Introspection Endpoint
Introspection request
Introspection response
Verify
Step 4 - DH receives Arrangement Revocation request from the CTS ADR
Revocation request
Revocation response
Verify
Step 5 - CTS ADR sends an Introspection request via the DH Introspection Endpoint
Introspection request
Introspection response
Verify



Data Recipient Initiated Revocation Scenario Sequence Diagram

15 Data Recipient Initiated Token Revocation Scenario

15.1 Purpose

The ability for a Participant DH to validate the correct treatment of the withdrawal of a token verifying that a Participant DH can receive a token revocation request from the CTS Simulated ADR (DR to DH).

15.2 Scenario Conditions

Not Applicable.

15.3 Endpoints

The table below lists endpoints specific to this scenario. See also Endpoints used in Data Holder Scenarios.

Endpoint	Description	Method
Token Revocation **DR to DH**	CTS Simulated ADR sends a Token Revocation request, using their token, to the Participant DH to withdraw a token.	POST

Link to specifications

https://consumerdatastandardsaustralia.github.io/standards-archives/standards-1.18.0/#security-endpoints

15.4 Scenario Results

Pass

You have passed the DR initiated token revocation test when you **can**:

- Receive a Token Revocation request from the CTS Simulated ADR to the Participant DH Revocation Endpoint with an Access/Refresh Token
- Validate the revocation request and return a success code (200 OK) response
- Return an error response (bad request) for the Get Customer request after the access token is revoked
- Return a response of (bad request) for the Refresh token request after the token has been revoked.

Fail

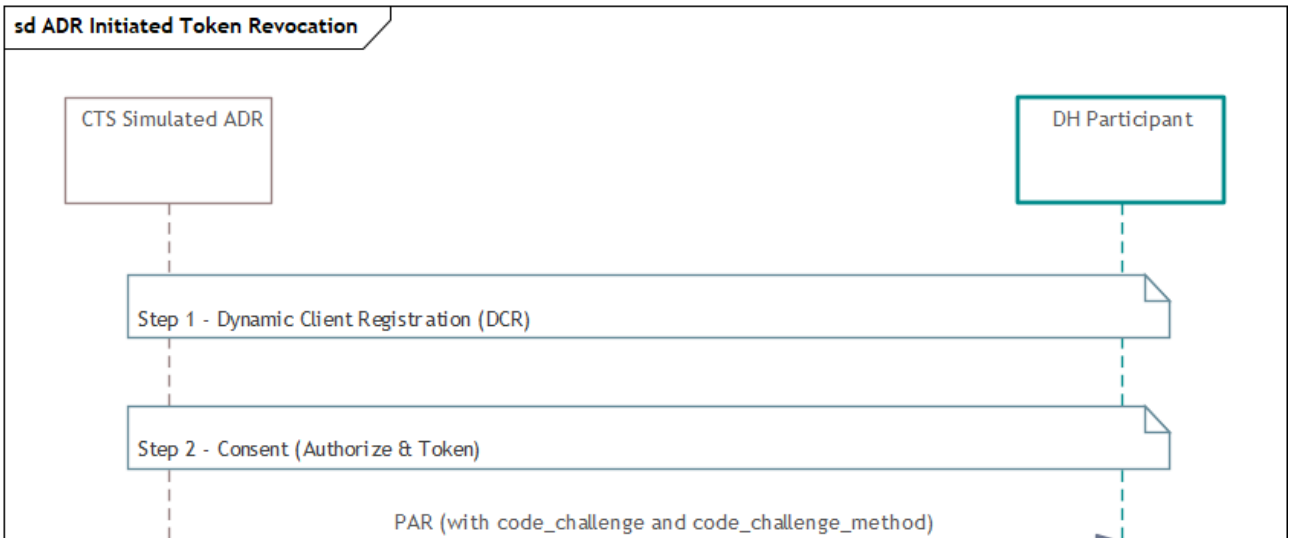
You have failed the DR initiated token revocation test when you **cannot**:

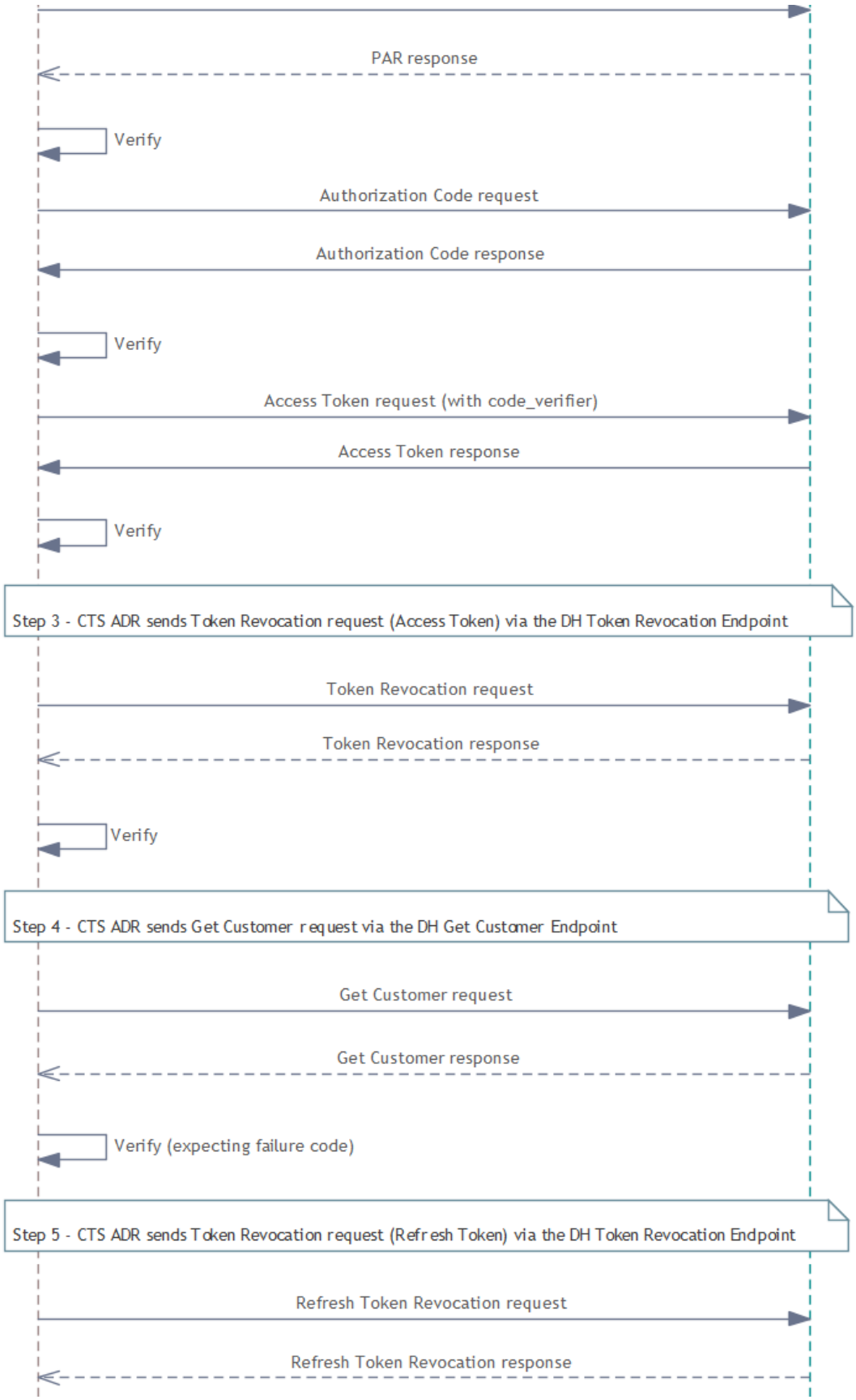
- Receive a Token Revocation request from the CTS Simulated ADR to the Participant DH Revocation Endpoint with an Access/Refresh Token

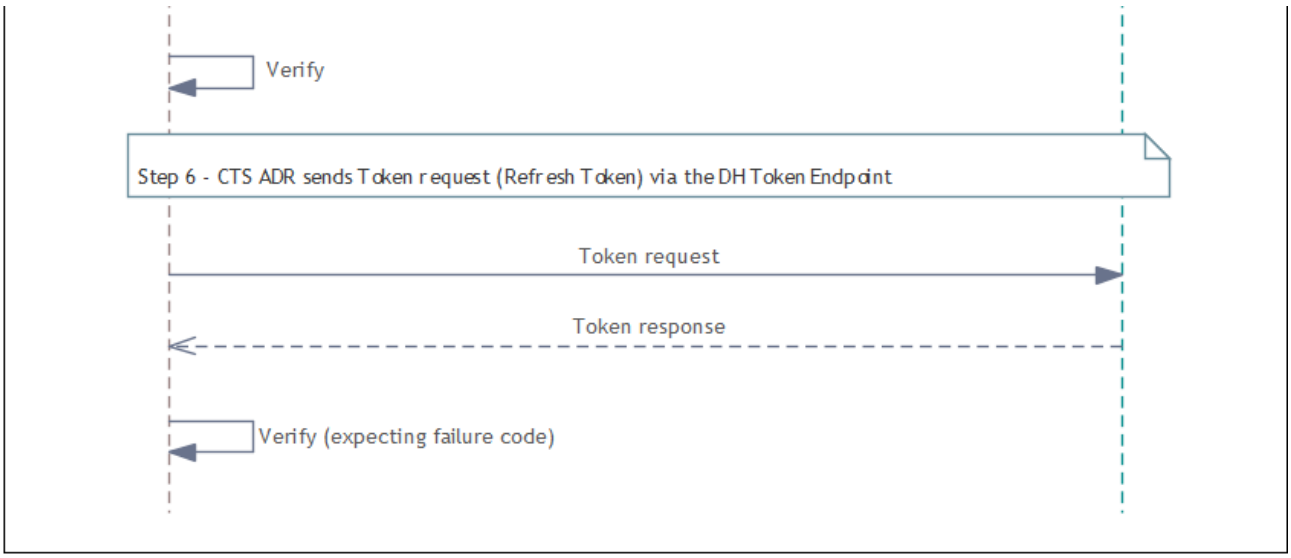
- Validate the revocation request and return a success code (200 OK) response
- Return an error response (bad request) for the Get Customer request after the access token is revoked
- Return a response of (bad request) the Refresh token request after the token has been revoked.

15.5 Scenario High Level Test Steps

1. Dynamic Client Registration (DCR) *(Steps 1-4 from DCR scenario)*
2. Consent (Authorise & Token) *(Steps 2, 3 & 4 from Concurrent Consent scenario)*
3. **CTS ADR sends Token Revocation Request (Access Token) via the DH Tokem Revocation Endpoint**
    a. CTS Simulated ADR sends a Token Revocation request with the Access Token via the DH Token Revocation Endpoint.
    b. Participant DH validates the CTS Simulated ADR request and returns a response.
    c. CTS Simulated ADR verifies the response.
4. **CTS ADR sends request via the DH Get Customer Endpoint**
    a. CTS Simulated ADR sends a request, using the Participant DH issued Access Token, to the Participant DH via the Get Customer API Endpoint.
    b. Participant DH validates the CTS Simulated ADR request and returns a response.
    c. CTS verifies the Participant DH Get Customer Response [bad request].
5. **CTS ADR sends Token Revocation Request (Refresh Token) via the DH Token Revocation Endpoint**
    a. CTS Simulated ADR sends a Token Revocation request with the Refresh Token via the DH Token Revocation Endpoint.
    b. Participant DH validates the CTS Simulated ADR request and returns a response.
    c. CTS Simulated ADR verifies the response.
6. **CTS ADR sends Token Request (Refresh Token) via the DH Token Endpoint**
    a. CTS Simulated ADR sends a Token request with the Refresh Token to the Participant DH via the Token Endpoint.
    b. CTS Simulated ADR verifies the response [bad request].




PAR response
Verify
Authorization Code request
Authorization Code response
Verify
Access Token request (with code_verifier)
Access Token response
Verify
Step 3 - CTS ADR sends Token Revocation request (Access Token) via the DH Token Revocation Endpoint
Token Revocation request
Token Revocation response
Verify
Step 4 - CTS ADR sends Get Customer request via the DH Get Customer Endpoint
Get Customer request
Get Customer response
Verify (expecting failure code)
Step 5 - CTS ADR sends Token Revocation request (Refresh Token) via the DH Token Revocation Endpoint
Refresh Token Revocation request
Refresh Token Revocation response



Data Recipient Initiated Token Revocation Scenario Sequence Diagram

16 Endpoints used in Data Holder Scenarios

Function	Endpoint	Hosted By	Description	Example URL
Discovery	**OpenID Provider Configuration End Point**	Data Holder	CTS Simulated ADR requests the discovery document from the Participant Data Holder via the Discovery Endpoint	/.well-known/openid-configuration
DCR	**Register Data Recipient oAuth Client**	Data Holder	CTS Simulated ADR sends a DCR request to the Data Holder via the Registration Endpoint	/register
	Get JWKS	Register	Participant DH requests the JWKS from the CTS Simulated Register via the JWKS Endpoint	/cts/{conformanceId}/register/cdr-register/v1/jwks
	Get JWKS	Data Recipient	Participant DH requests the JWKS from the CTS Simulated ADR via the JWKS Endpoint	/cts/{conformanceId-guid}/dr/jwks
	Redirect URI	Data Recipient	Participant DH calls the CTS Simulated ADR Redirect Uri Endpoint to signin	/cts/{conformanceId-guid}/dr/signin
	Get Data Recipient Statuses	Register	Participant DH requests the ADR status from the CTS Simulated Register via the Get Data Recipient Status Endpoint	/cts/{conformanceId}/register/cdr-register/v1/{industry}/data-recipients/status
	Get Software Product Statuses	Register	Participant DH requests the software product status from the CTS Simulated Register via the Get Software Product Status Endpoint	/cts/{conformanceId}/register/cdr-register/v1/{industry}/data-recipients/brands/software-products/status

	Get Data Recipients	Register	Participant DH requests the data recipients from the CTS Simulated Register via the Get data recipients Endpoint	/cts/{conformanceId}/register/cdr-register/v1/{industry}/data-recipients
Consent	**Authorize**	Data Holder	CTS Simulated ADR requests authorisation with the Participant DH via the Authorize Endpoint	/authorize
	Token	Data Holder	CTS Simulated ADR exchanges their code for a Token from the Participant DH via the Token Endpoint CTS Simulated ADR exchanges their Refresh Token for an Access Token from the Participant DH via the Token Endpoint	/token
	Introspection	Data Holder	CTS Simulated ADR sends an Introspection request to the Participant DH Token Introspection Endpoint to retrieve information about a token	/token/introspection
	Push Authorisation	Data Holder	CTS Simulated ADR sends a Pushed Authorisation Request object for request_uri to the Participant DH via Pushed Authorisation Endpoint	/par
	Get Customer	Data Holder	CTS Simulated ADR sends a request to the Participant DH's Get Customer Endpoint	/common/customer

Revocation	**Arrangement Revocation**	Data Recipient	Participant DH sends a request, using their CDR Arrangement ID, to the CTS Simulated ADR to withdraw Arrangement Consent	/cts/{conformanceId-guid}/dr/arrangements/revoke
	Arrangement Revocation	Data Holder	CTS Simulated ADR makes an Arrangement Revocation request to the Participant DH Revocation Endpoint (registered uri)	/arrangements/revoke
	Token Revocation	Data Holder	CTS Simulated ADR makes a Token Revocation request to the Participant DH Token Revocation Endpoint (registered uri)	/revocation

17 CTS Glossary

This section provides a list of CTS specific terms and their meanings.

Term	Meaning
Accredited Data Recipient (ADR)	An Accredited Data Recipient (ADR) is a system entity that is accredited to collect CDR data from Participant Data Holders through authorised Software Products. A Data Recipient **MUST** be accredited in order to participate in the CDR Federation. Accreditation rules for Data Recipients are beyond the scope of this artefact. The process of accreditation is managed by the CDR Registrar. For the purposes of the CTS a single accredited organisation is represented via the Register as a single Data Recipient and **MAY** be represented by multiple separate Software Products to support multiple applications or services.
Authenticate / Authentication	When a consumer verifies themselves with a Participant DH For more information see: https://consumerdatastandardsaustralia.github.io/standards-archives/standards-1.18.0/#authentication-flows
Authorize / Authorization	A consumer confirming to the disclosure of their CDR data from a Participant DH For more information see: https://openid.net/specs/openid-connect-core-1_0.html#Overview
Brand	A Participant DH's system that is designed to interact with a Participant ADRs software product.
CDR	Consumer Data Right
CDS	Consumer Data Standards

Consent	Used to refer to when a consumer agrees to share their CDR data with a Participant ADR for a specific purpose (i.e. collect and use); technically distinguished from the final affirmative action (i.e. authorise) in the consent flow. Consent is also used as a term in consumer-facing interactions to refer to data sharing arrangements. Consent requirements will be communicated between the Participant ADR and Participant DH via the authorisation request object. The primary mechanism for capturing consent will be scopes and claims under Open ID connect. Other patterns for the establishment of consent may be considered in the future, including the incorporation of fine-grained consent for specific use cases. For more information see: https://consumerdatastandardsaustralia.github.io/standards-archives/standards-1.18.0/#consent
CTS	Conformance Test Suite
CTS Simulated ADR	The Simulated Data Recipient built within CTS. Used to test a Participant DH's brand during on-boarding.
CTS system	The components of the CTS which a Participant ADR or a Participant DH will interact with during conformance testing.
CTS Simulated Register	The Register is a central point of discovery for both Data Holders and Data Recipients. CTS has replicated the Register which is referred to as the CTS Simulated Register for the purpose of testing Participant ADR software products and Participant DH brands during on-boarding.
Data Holder (DH)	The Participant Data Holder is a system entity that authenticates a consumer (Customer, resource owner or user), as part of an authorisation process initiated by a Participant Accredited Data Recipient, and issues an authorisation for that Participant ADR to access the Customer's data via published APIs. For the purposes of CTS a single designated organisation **MAY** be represented via the CDR Register as multiple separate Data Holders to support multiple brands or market identities.

Revoke / revocation	When a consumer stops a data sharing arrangement (i.e. consent/authorisation). This can occur via an ADR or a DH. A Participant DH and Participant ADR **MUST** implement a CDR Arrangement Revocation Endpoint as described in the Consumer Data Standards Security Endpoints. The CDR Arrangement Revocation Endpoint is used to revoke an existing sharing arrangement. The Participant DH **MUST** implement a Token Revocation Endpoint as described in section 2 of [RFC7009]. The revocation end point serves as a revocation mechanism that allows an ADR to invalidate its tokens as required to allow for token clean up. Revocation of refresh tokens and access tokens **MUST** be supported. For more information see: • https://consumerdatastandardsaustralia.github.io/standards-archives/standards-1.18.0/#security-endpoints • https://tools.ietf.org/html/rfc7009#section-2
Software Product	A software product developed by a Participant ADR is a system entity that is authorised by a Data Holder to access consumer resources (APIs). It is designed to interact with a Participant DH brand to facilitate consent and request consumer data.
Test run	A single instance of end to end testing that a participant will complete, resulting in a report for the CDR Register (the Register) to consider in allowing the participant to be active on the Register.
TP	Test Plan
Withdrawal	See **revocation**.

18 Brand vs Conformance ID Infographic

The below diagram illustrates how each brand has its own Conformance ID / testing workflow.