Australian Government

Consumer Data Right

Data Holder Technical Guidance Material 5.2.0

Conformance Test Suite

Table of Contents

1 Document Control

Document Version	1.0
Document Status	Approved
Issued Date	10 Apr 2025
Owner	ACCC

1.1 Test Plan Revision History

Test Plan Version	CDS Standards Version	Issued Date	Description of Changes
5.2.0	1.33.0	10 Apr 2025	The new version of Data Holder (DH) test plan (TP) 5.2.0 is created to conform with version 1.33.0 of the Consumer Data Standards, notably the retirement of OIDC Hybrid flow. As part of the changes the test plan will: • Validate data holders to ensure that the OIDC hybrid flow is not supported. Please refer ***Technical Considerations*** section on this page for more information about the implemented changes due to retirement of OIDC hybrid flow in CTS.

Test Plan Version	CDS Standards Version	Issued Date	Description of Changes
5.1.0	1.31.0	05 Sep 2024	The new version of Data Holder (DH) test plan (TP) 5.1.0 includes the following changes: • CTS was sending all scopes except for Client Registration as part of the PAR Requests that it sends in the DH plans. In 5.1.0, CTS will now send only sector specific scopes - so if the DH is banking, CTS will only send the relevant mandatory, customer and banking scopes and not send energy scopes anymore and vice versa. • Additionally, CTS will use a different set of JWKS supporting PS256 for interaction with the mock register instead of using the JWKS generated for the DH. Previously, the same JWKS was used for the register and the data holder which meant that if the data holder only chooses to support ES256 for signing but the register is configured to expect PS256 the test plan would fail which is something that we encountered when testing. ◦ Further, CTS has also improved validation in Ensure Infosec Endpoints using MTLS scenario for missing client certificate test step.

Test Plan Version	CDS Standards Version	Issued Date	Description of Changes
5.0.1	1.30.0	13 June 2024	This test plan has been updated to conform with version 1.30.0 of the Consumer Data Standards. As part of these changes the test plan has: • CTS Simulated Register has been updated to improve consistency across ACCC CDR implementations. ◦ Introduction of an additional scenario group 'Ensure SSA Validation' to ensure that the DH is validating the SSA signature in the DCR request.
5.0.0	1.27.0	30 Nov 2023	For the Data Holder test plan 5.0.0 the existing 12 large scenarios which had repetitive duplicated steps, have been condensed into a single testing scenario, consisting of smaller grouped steps to reduce redundancy and simplify the testing process. Multiple client registrations have been replaced with a single shared client registration for the test plan.
4.3.0	1.24.0	15 June 2023	The Data Holder test plan 4.3.0 updates the 'Ensure Holder of Key for Resource Requests' scenario to be sector agnostic. This test plan will allow data holders to demonstrate conformance with version 1.24.0 of the Consumer Data Standards. This test plan will use Common APIs instead of sector specific resource APIs. The CTS Simulated ADR will now call the Data Holder's Get Customer endpoint instead of the Get Accounts & Get Energy Accounts endpoints.

Test Plan Version	CDS Standards Version	Issued Date	Description of Changes
4.2.1	1.23.0	15 May 2023	This test plan version has been updated to conform with version 1.23.0 of the Consumer Data Standards. As part of these changes the Data Holder test plan has: • Removed id_token encryption parameters from Dynamic Client Registration request for Authorisation Code Flow ◦ Updated validation of the ID Token during authorisation
4.2.0	1.22.0	20 April 2023	The following API versions have been retired and are no longer supported by CTS: • Get Software Product Statuses V1 • Get Data Recipient Statuses V1 • Get Data Recipients V1 & V2
4.1.0	1.21.0	16 February 2023	This test plan has been updated to conform with version 1.21 of the Consumer Data Standards, notable the FAPI 1.0 Phase 3 obligations. As part of these changes Data Holder test plan has: • Added conformance testing for the Authorisation Code Flow (ACF) • Added support for JARM • Removed id_token encryption in Authorisation Code Flow (ACF) • Removed conformance testing for OIDC Hybrid Flow
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 Holders (DH) 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 requests.

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 the 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 competencies will be added as ecosystem requirements change. The execution of the CTS is not a one-time event for a participant, it is expected that active participants will complete the new test competencies 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 the 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 scenarios 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 scenarios 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

Retirement of OIDC Hybrid flow

As part of CDS v1.33.0, the proposal is to retire the OIDC Hybrid Flow after which CTS will ensure that ONLY Authorization Code Flow is supported by Data Holders from 12th May 2025. *Therefore, a new Data Holder Test Plan v5.2.0 is introduced in CTS to ensure that the data holders do not support OIDC hybrid flow.*

As part of these changes, the new test plan has:

- Modified the validation of the **Discovery Document response** received from a data holder to
  - Ignore `id_token_encryption_alg_values_supported` and `id_token_encryption_enc_values_supported` fields.
  - Change validations for `response_types_supported` field to ensure that the value does not include "code id_token".
- Modified the validations of the **Client Registration Response** received from a data holder for **Create, Update and Get Client Registration** requests to
  - Ensure that the response must not either contain `id_token_encrypted_response_alg` and `id_token_encrypted_response_enc` fields or the fields must have 'blank' or 'null' value". It is acceptable for these fields to be completely absent in the response.
  - Change validations for `response_types` field to ensure that the value does not include "code id_token".

3 Test Scenario

This section captures the following scenario groups and all their associated steps that can be part of a DH test plan:

1. Discovery Document
2. Ensure Infosec Endpoints Using MTLS
3. Ensure SSA Validation
4. Create Client Registration
5. First Consent
6. Holder Of Key Resource Requests
7. Second Consent
8. Data Recipient Initiated Arrangement Revocation
9. Amending Existing Consent
10. Data Recipient Initiated Token Revocation
11. Data Holder Initiated Arrangement Revocation
12. Ensure Client Assertion Data in Requests
13. Retrieve and Update Client Registration
14. Removed Software Product

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.

Before you start:

1. Apply the CTS certificates to your brand.
2. If necessary, configure your software product to access CTS. Infrastructure changes, such as firewall rules or IP allowlisting, may need to be configured.
3. You must have a valid account on the CDR Participant Portal.
4. You must be registered as a CTS tester as part of the CTS enrolment.

Consult the on-boarding guide and the CTS Connection Datasheet for more information on the steps and actions.

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 Analyst when you submit your test results via email, so that an On-boarding Officer can start assessing your results.

3.3 Register Status Endpoints

1. After the Discovery Document scenario group, you must make a valid request to at least one of these three Register status endpoints, available in the "Endpoints used in Data Holder Scenarios" section, to continue the scenario execution.
    a. Get Data Recipient Statuses
    b. Get Software Product Statuses
    c. Get Data Recipients
2. Once you have made a valid request to one of the three endpoints, you can select the "Continue" button when you are ready for the scenario to continue execution.
3. These three endpoints remain available throughout the scenario execution.

3.4 Discovery Document

3.4.1 Purpose

The ability for a Participant DH to demonstrate that they can return a valid Open ID Discovery Document (OIDD).

3.4.2 Scenario Conditions

Not Applicable.

3.4.3 Endpoints

See also Endpoints used in Data Holder Test Scenario

Endpoint	Description	Method
OpenID Provider Configuration End Point	The CTS Simulated ADR requests the Discovery Document from the Data Holder via the Discovery endpoint	GET

Link to specifications

https://consumerdatastandardsaustralia.github.io/standards/#register-apis

3.4.4 Scenario Results

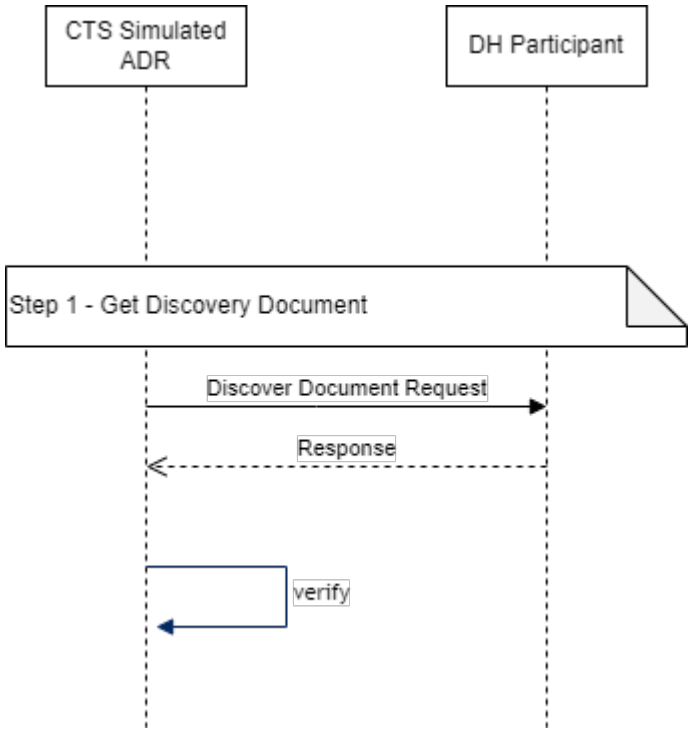
Pass

You have passed the Discovery Document scenario when you **can**:

- Receive a valid GET request to your Discovery endpoint (provided during enrolment) and send a valid Discovery Document JSON response

3.4.5 Scenario High-Level Test Steps

1. **CTS Simulated 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.



3.5 Ensure Infosec Endpoints Using MTLS

3.5.1 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.

3.5.2 Scenario Conditions

Not Applicable.

3.5.3 Endpoints

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

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/#dcr-apis

3.5.4 Scenario Results

Pass

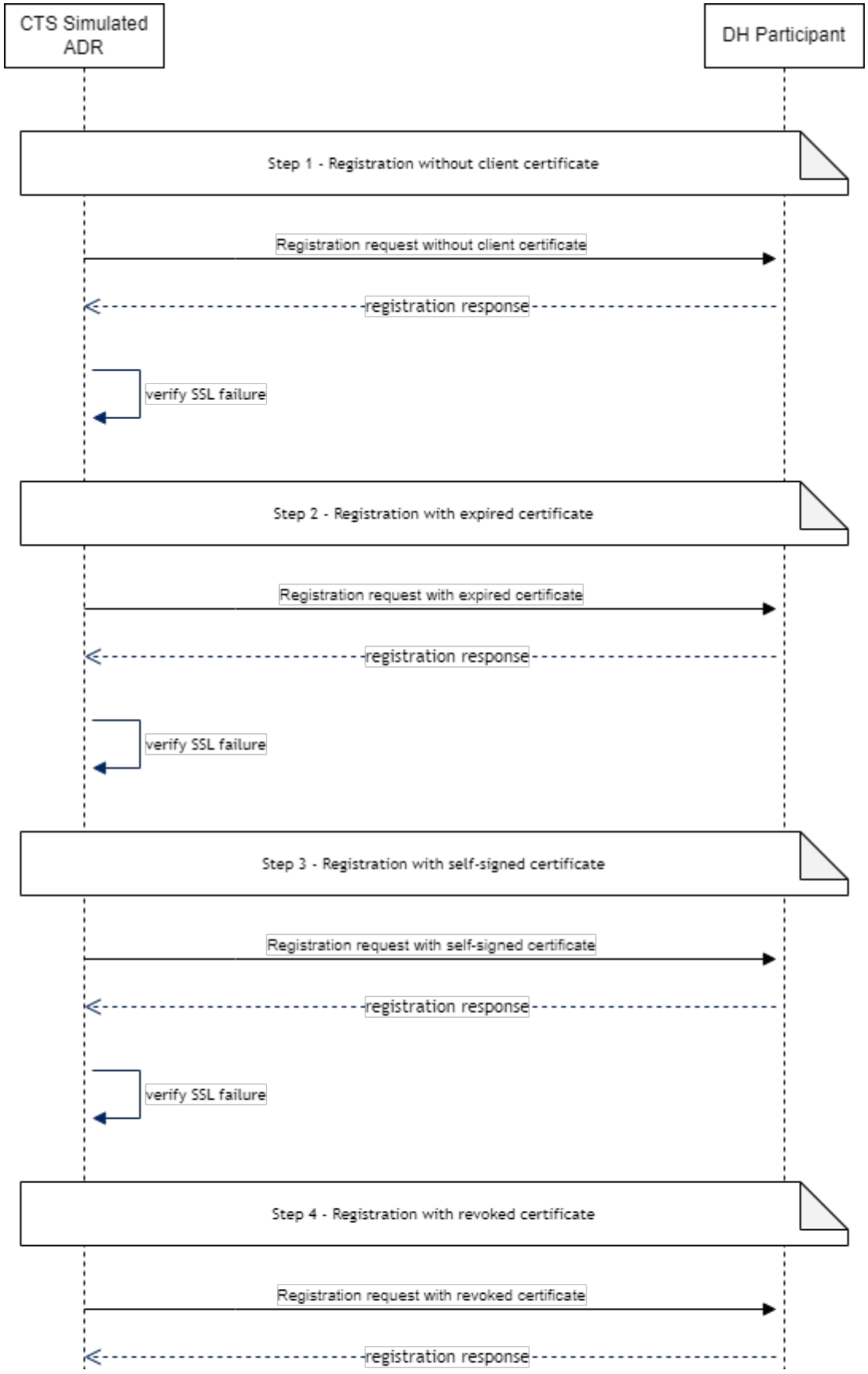
You have passed the scenario when the registration request is sent with an invalid/missing client certificate and the response is an appropriate error that indicates invalidation of the certificate.

3.5.5 Scenario High Level Test Steps

1. **CTS Simulated 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 either 'an http status code of >=400 or <=599' OR 'no response is received OR response indicates invalidation of the certificate'.
2. **CTS Simulated 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 either 'an http status code of >=400 or <=599' OR 'no response is received OR response indicates invalidation of the certificate'.
3. **CTS Simulated 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 either 'an http status code of >=400 or <=599' OR 'no response is received OR response indicates invalidation of the certificate'.

4. **CTS Simulated 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 either 'an http status code of >=400 or <=599' OR 'no response is received OR response indicates invalidation of the certificate'.


CTS Simulated ADR
DH Participant
Step 1 - Registration without client certificate
Registration request without client certificate
registration response
verify SSL failure
Step 2 - Registration with expired certificate
Registration request with expired certificate
registration response
verify SSL failure
Step 3 - Registration with self-signed certificate
Registration request with self-signed certificate
registration response
verify SSL failure
Step 4 - Registration with revoked certificate
Registration request with revoked certificate
registration response

3.6 Ensure SSA Validation

3.6.1 Purpose

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

- Registration request sent with invalid SSA

3.6.2 Scenario Conditions

Not Applicable.

3.6.3 Endpoints

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

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/#register-apis

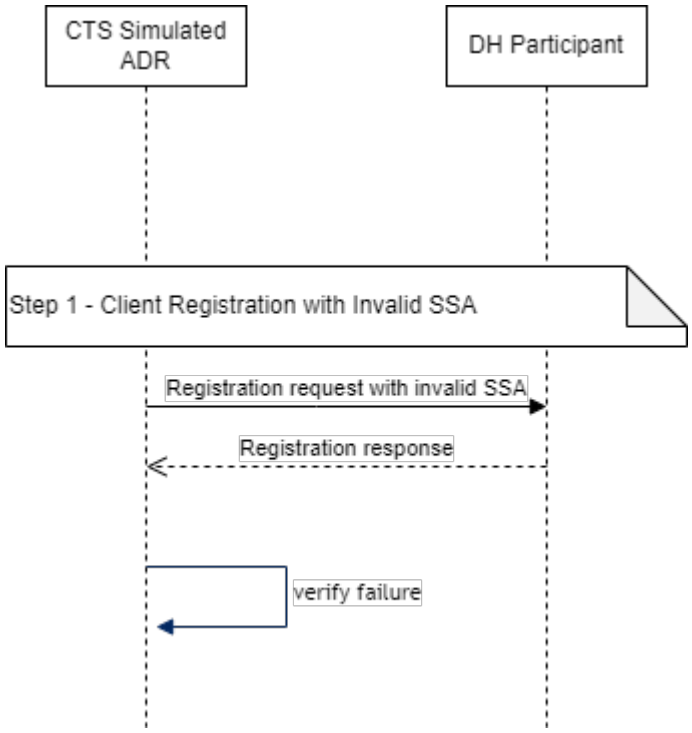
3.6.4 Scenario Results

Pass

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

3.6.5 Scenario High Level Test Steps

1. **CTS Simulated ADR sends registration request via the DH Registration endpoint with invalid SSA**
    a. CTS Simulated ADR sends a registration request with invalid SSA to the Participant DH.
    b. Participant DH validates the CTS Simulated ADR request and responds with error "`400 invalid_software_statement"` or `"400 unapproved_software_statement"` " containing a Response Body that aligns to the schema defined here for RegistrationError.



3.7 Create Client Registration

3.7.1 Purpose

The ability for a Participant DH to demonstrate that they can register the CTS simulated ADR's software product.

3.7.2 Scenario Conditions

Not Applicable.

3.7.3 Endpoints

See also Endpoints used in Data Holder Test Scenario

Endpoint	Description	Method
OpenID Provider Configuration End Point	The CTS Simulated ADR requests the Discovery Document from the Data Holder via the Discovery endpoint	GET
Dynamic Client Registration	The 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 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

Link to specifications

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

3.7.4 Scenario Results

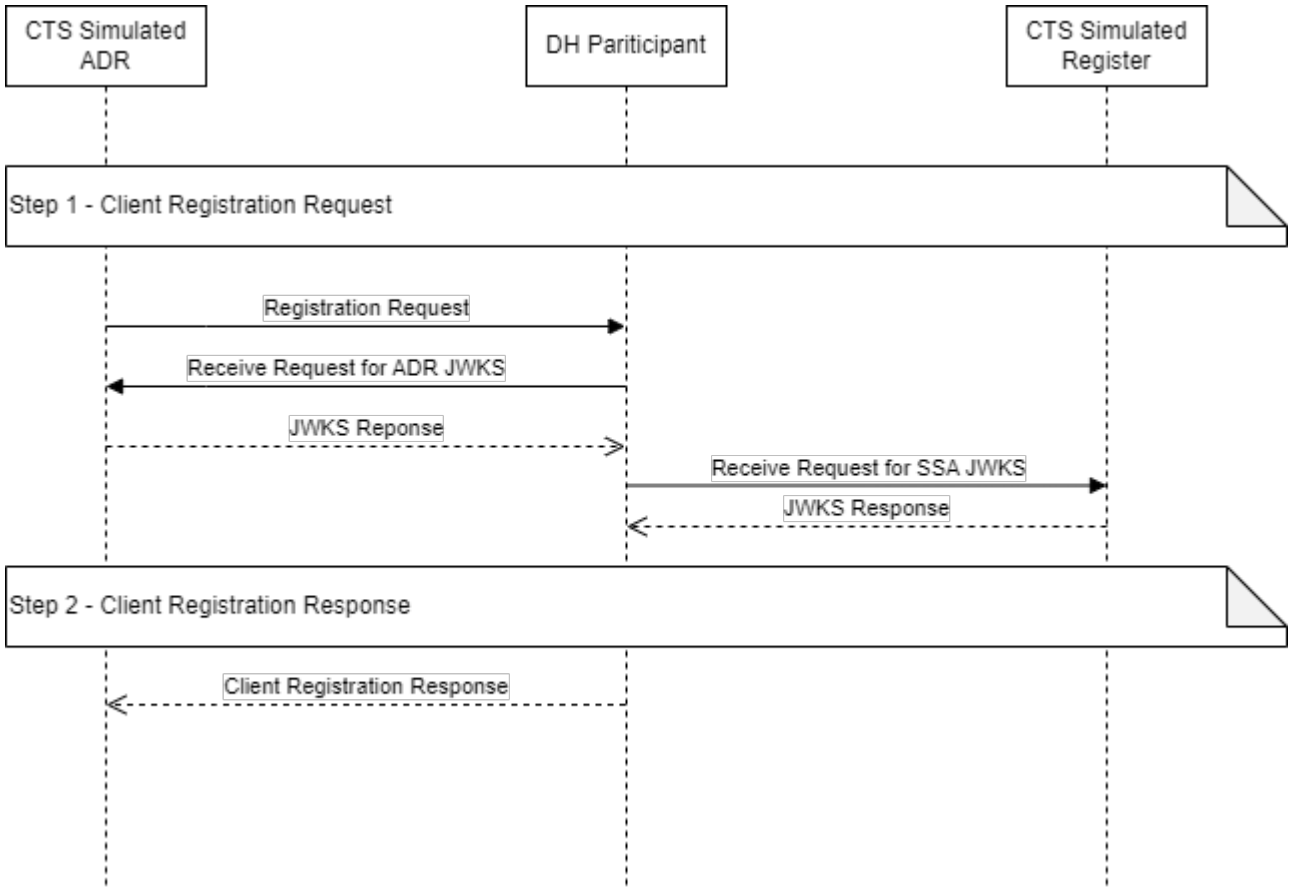
Pass

You have passed the scenario when you can receive a valid DCR request from the CTS Simulated ADR, successfully register the CTS Simulated ADR software product, and supply a valid response.

3.7.5 Scenario High-Level Test Steps

1. **CTS Simulated 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 Register SSA JWKS endpoint to verify the SSA signature using the Register public keys.
        ii. CTS Simulated Register returns a response with an SSA JWKS (No scenario group associated).
        iii. Participant DH calls the CTS Simulated ADR JWKS endpoint (using the jwks_uri from the SSA).
        iv. CTS Simulated ADR returns a response with an ADR JWKS.
2. **DH responds to the CTS Simulated ADR DCR request**

a. Participant DH registers the software product and returns a response to the CTS Simulated ADR.



3.8 First Consent

3.8.1 Purpose

The ability for a Participant DH to verify the consent flow with the CTS Simulated ADR. 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.

3.8.2 Scenario Conditions

Not Applicable.

3.8.3 Endpoints

See also Endpoints used in Data Holder Test Scenario

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
Authorisation	CTS Simulated ADR requests authorisation with the Participant DH via the Authorisation 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

Link to specifications

https://consumerdatastandardsaustralia.github.io/standards/#introduction

3.8.4 Scenario Results

Pass

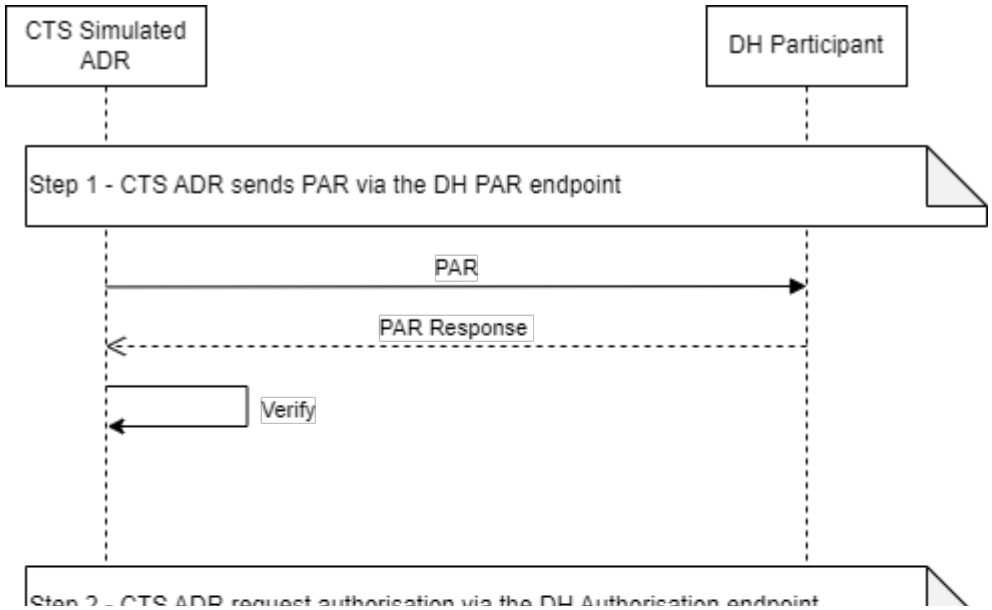
You have passed the First Consent scenario when you **can**:

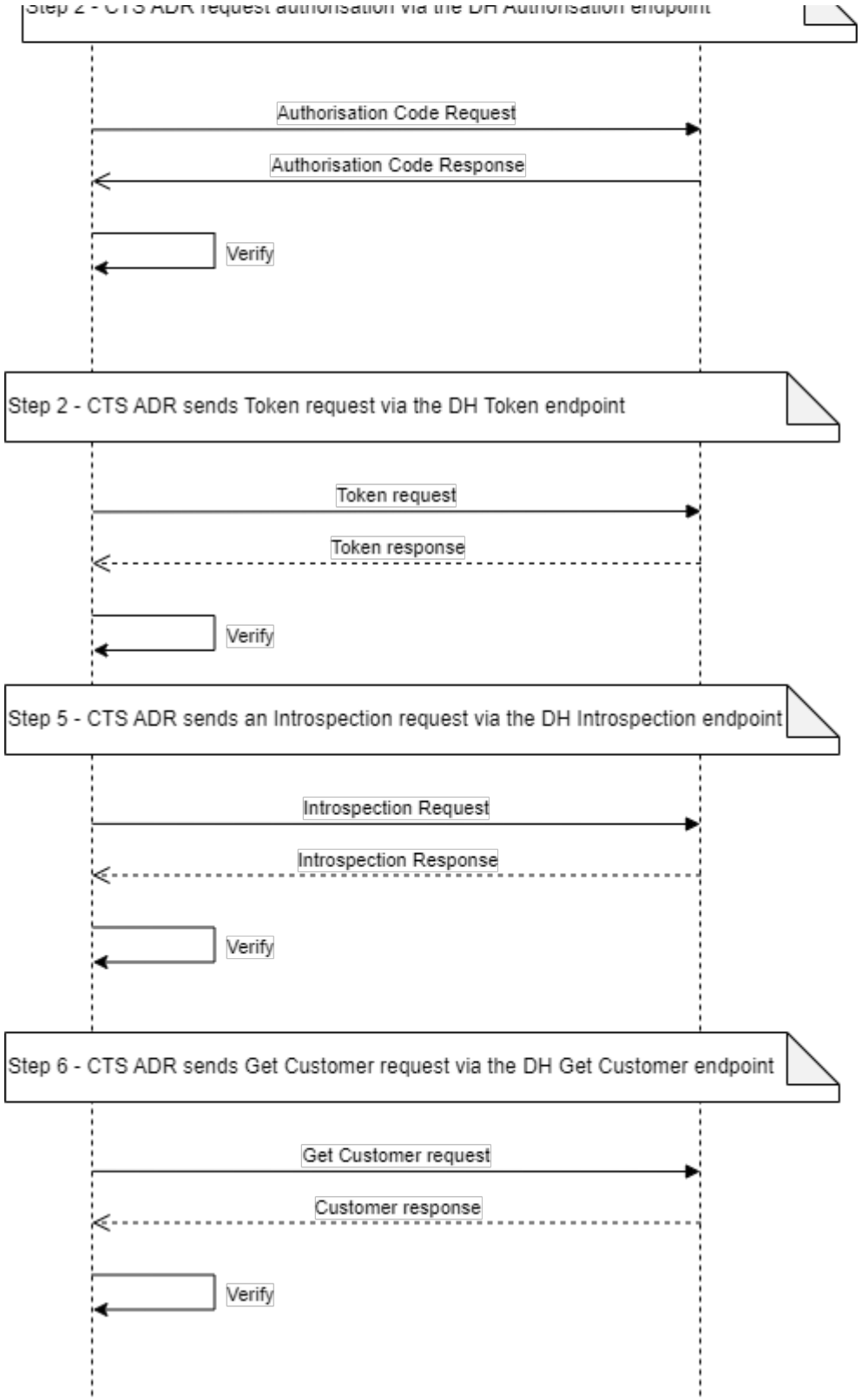
- Authorise a consent on behalf of a single consumer
- Ensure that with consent established, a CDR Arrangement ID is created
- Redirect back to the CTS Simulated ADR
- Return an authorisation and token response, for the consent
- Return a valid response payload from the Get Customer endpoint.

3.8.5 Scenario High Level Test Steps

1. **CTS Simulated ADR sends a Pushed Authorisation 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 the request_uri in the Participant DH response.
2. **CTS Simulated ADR requests authorisation via the DH Authorisation endpoint**
   a. CTS Simulated ADR sends a request to the Participant DH via the Authorisation endpoint.
   b. Participant DH validates the CTS Simulated ADR authorisation request, verifying that the CTS Simulated ADR software product is registered with the Participant DH and responds via the Redirect URI with code and state.
   c. CTS verifies the Participant DH authorisation response.
3. **CTS Simulated ADR sends a 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, an ID Token and Refresh Token (for ongoing consents only).
   c. CTS verifies the Participant DH Token response.
4. **CTS Simulated 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.
5. **CTS Simulated ADR sends a request to the DH Get Customer endpoint**
   a. CTS Simulated ADR sends a customer 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.




Authorisation Code Request
Authorisation Code Response
Verify
Step 2 - CTS ADR sends Token request via the DH Token endpoint
Token request
Token response
Verify
Step 5 - CTS ADR sends an 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
Customer response
Verify

3.9 Holder Of Key Resource Requests

3.9.1 Purpose

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

3.9.2 Scenario Conditions

A request is made to the Get Customer endpoint with a CDR certificate that is not bound to the Access Token used in the request.

3.9.3 Endpoints

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

Endpoint	Description	Method
Get Customer	CTS Simulated ADR sends a request to the Participant DH's Get Customer Endpoint	GET

Link to specifications

https://consumerdatastandardsaustralia.github.io/standards/#common-apis

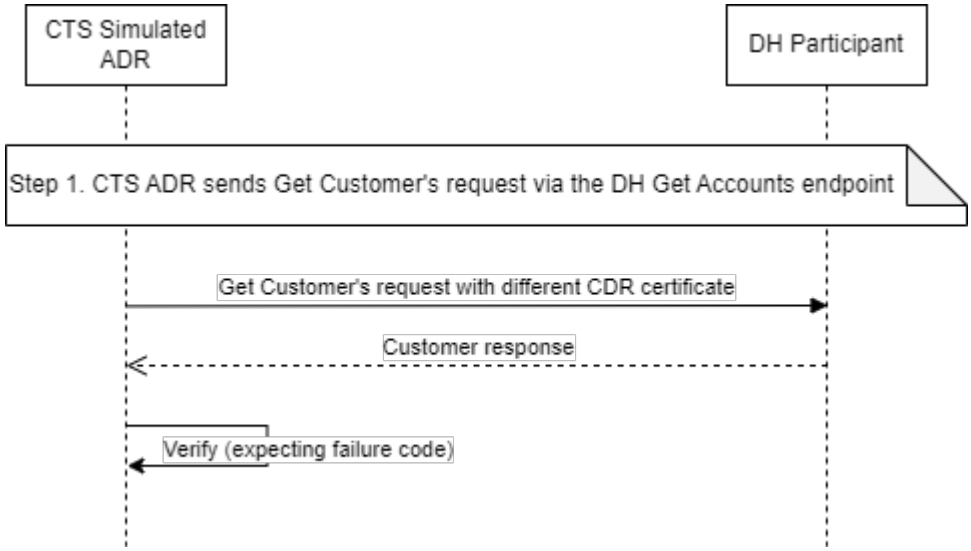
3.9.4 Scenario Results

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.

3.9.5 Scenario High Level Test Steps

1. **CTS Simulated ADR sends a 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.



3.10 Second Consent

3.10.1 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.

3.10.1.1 Business Context

Second 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.

3.10.2 Scenario Conditions

Not Applicable.

3.10.3 Endpoints

See also Endpoints used in Data Holder Test Scenario

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
Authorisation	CTS Simulated ADR requests authorisation with the Participant DH via the Authorisation 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
Get Customer	CTS Simulated ADR sends a request to the Participant DH's Get Customer endpoint	GET

Link to specifications

https://consumerdatastandardsaustralia.github.io/standards/#introduction

3.10.4 Scenario Results

Pass

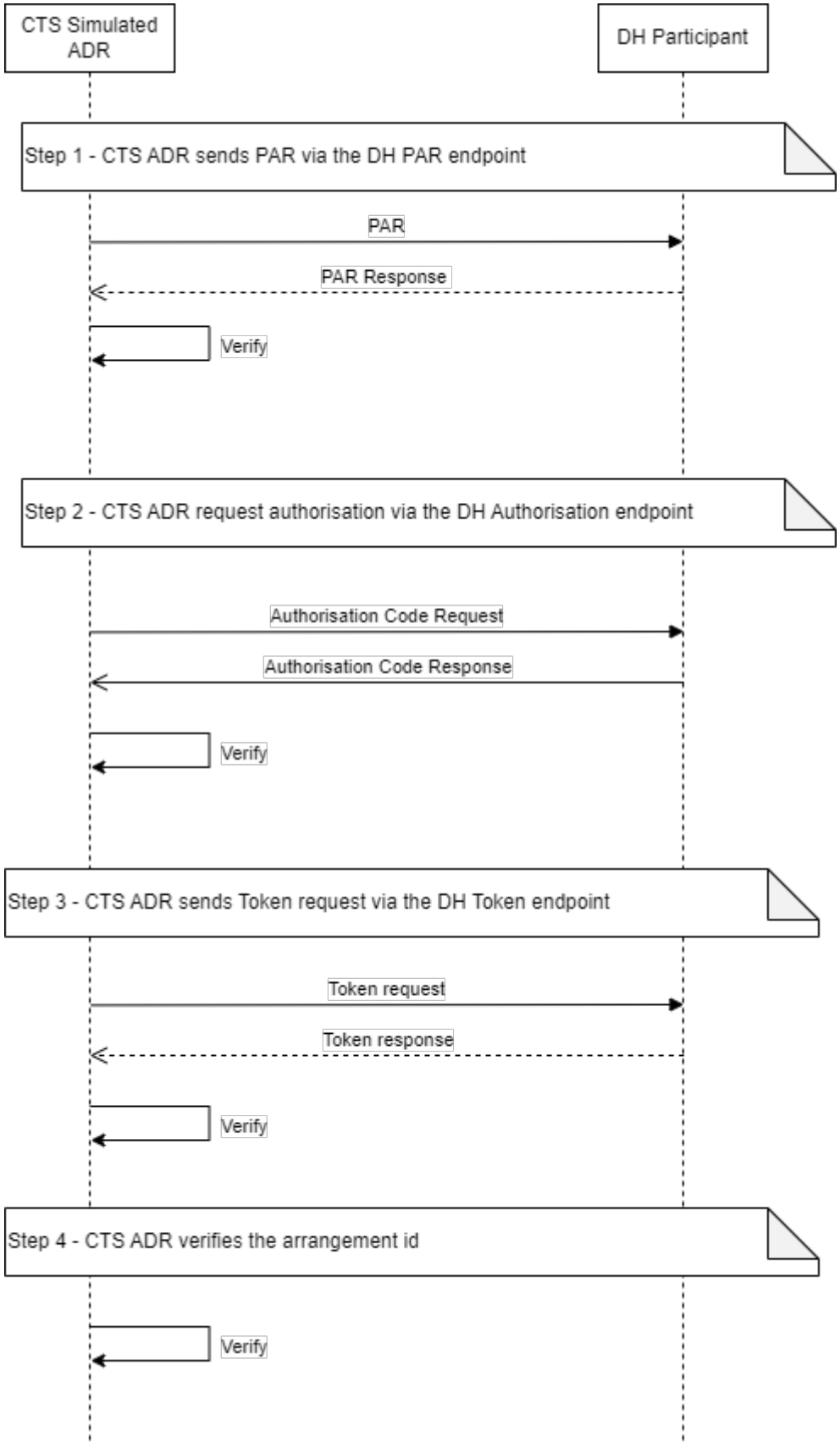
You have passed the Second Consent scenario when you **can**:

- Authorise a second consent on behalf of a consumer
- Redirect back to the CTS Simulated ADR
- Return an authorisation response and a token response (Access and Refresh Token)
- Ensure that with the second consent established, a new CDR Arrangement ID is created

3.10.5 Scenario High Level Test Steps

1. **CTS Simulated ADR sends a 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 second PAR and responds with request_uri.
    c. CTS verifies in the Participant DH's PAR response, that request_uri is contained in the response.
2. **CTS Simulated ADR sends an authorisation code request via the DH Authorisation endpoint**
    a. CTS Simulated ADR sends a request to the Participant DH via the Authorisation endpoint for the same user.

   b. Participant DH validates the CTS Simulated ADR authorisation request, verifying that the CTS Simulated ADR software product is registered with the Participant DH and responds via the Redirect URI with code and state.
   c. CTS verifies the Participant DH response.
3. **CTS Simulated 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, an ID Token and Refresh Token (for ongoing consents only).
4. **CTS Simulated ADR verifies the arrangement id**
   a. CTS Simulated ADR verifies that a new arrangement id has been generated for the second consent.


CTS Simulated ADR
DH Participant
Step 1 - CTS ADR sends PAR via the DH PAR endpoint
PAR
PAR Response
Verify
Step 2 - CTS ADR request authorisation via the DH Authorisation endpoint
Authorisation Code Request
Authorisation Code Response
Verify
Step 3 - CTS ADR sends Token request via the DH Token endpoint
Token request
Token response
Verify
Step 4 - CTS ADR verifies the arrangement id
Verify

3.11 Data Recipient Initiated Arrangement Revocation

3.11.1 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.

3.11.2 Scenario Conditions

A CDR Arrangement Id was issued to the CTS Simulated ADR by the Participant DH.

3.11.3 Endpoints

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

Endpoint	Description	Method
Arrangement Revocation **ADR to DH**	CTS Simulated ADR sends a request, using their CDR Arrangement Id, to the DH to withdraw arrangement consent	POST

Link to specifications

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

3.11.4 Scenario Results

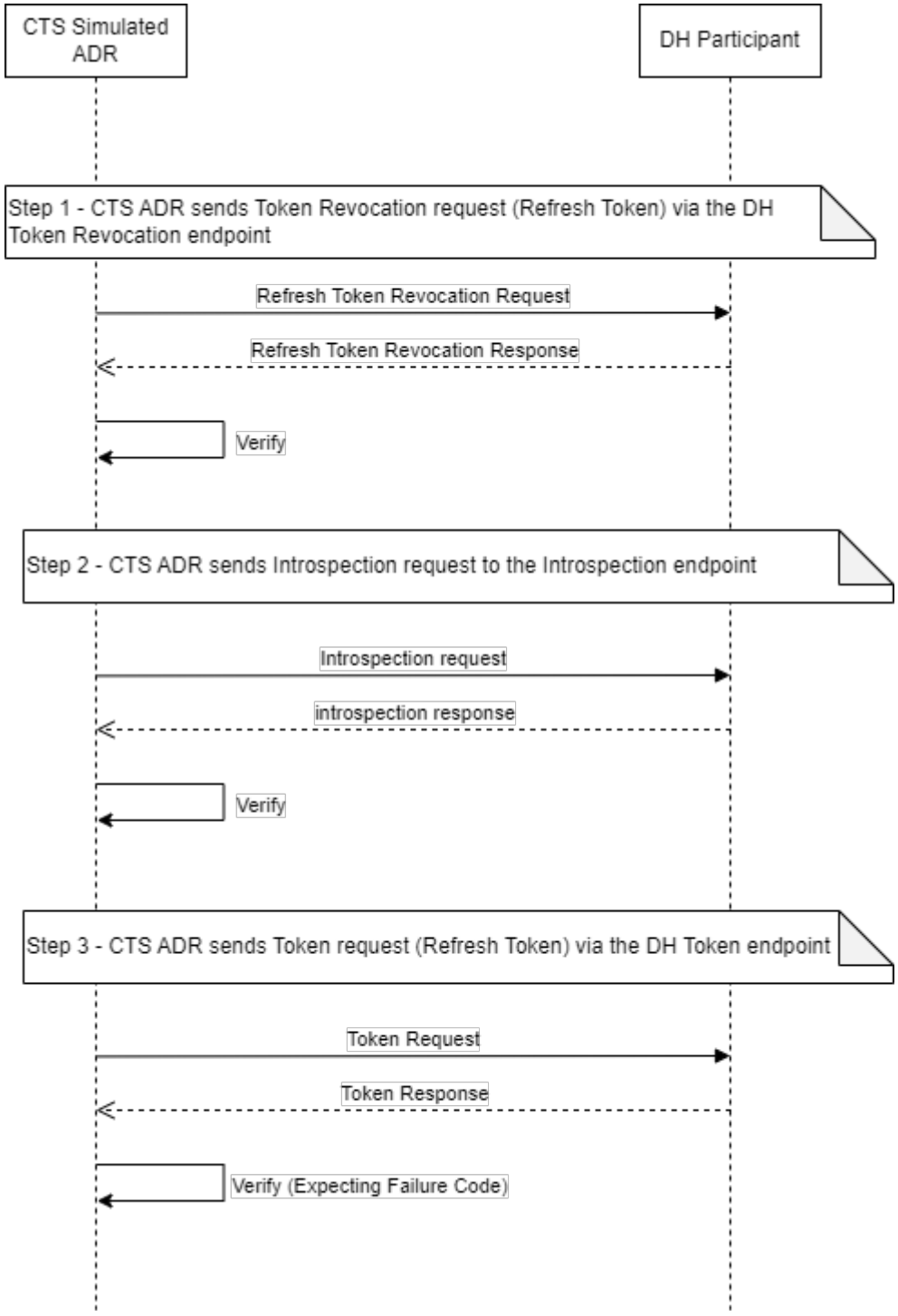
Pass

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

3.11.5 Scenario High Level Test Steps

1. **CTS Simulated ADR sends an arrangement revocation request to the DH's revocation endpoint**
    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.
2. **CTS Simulated 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.
3. **CTS Simulated 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
   c. CTS verifies the Participant DH token response and expects an error response from the DH



3.12 Amending Existing Consent

3.12.1 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.

3.12.2 Scenario Conditions

Not Applicable.

3.12.3 Endpoints

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

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

Link to specifications

https://consumerdatastandardsaustralia.github.io/standards/#introduction

3.12.4 Scenario Results

Pass

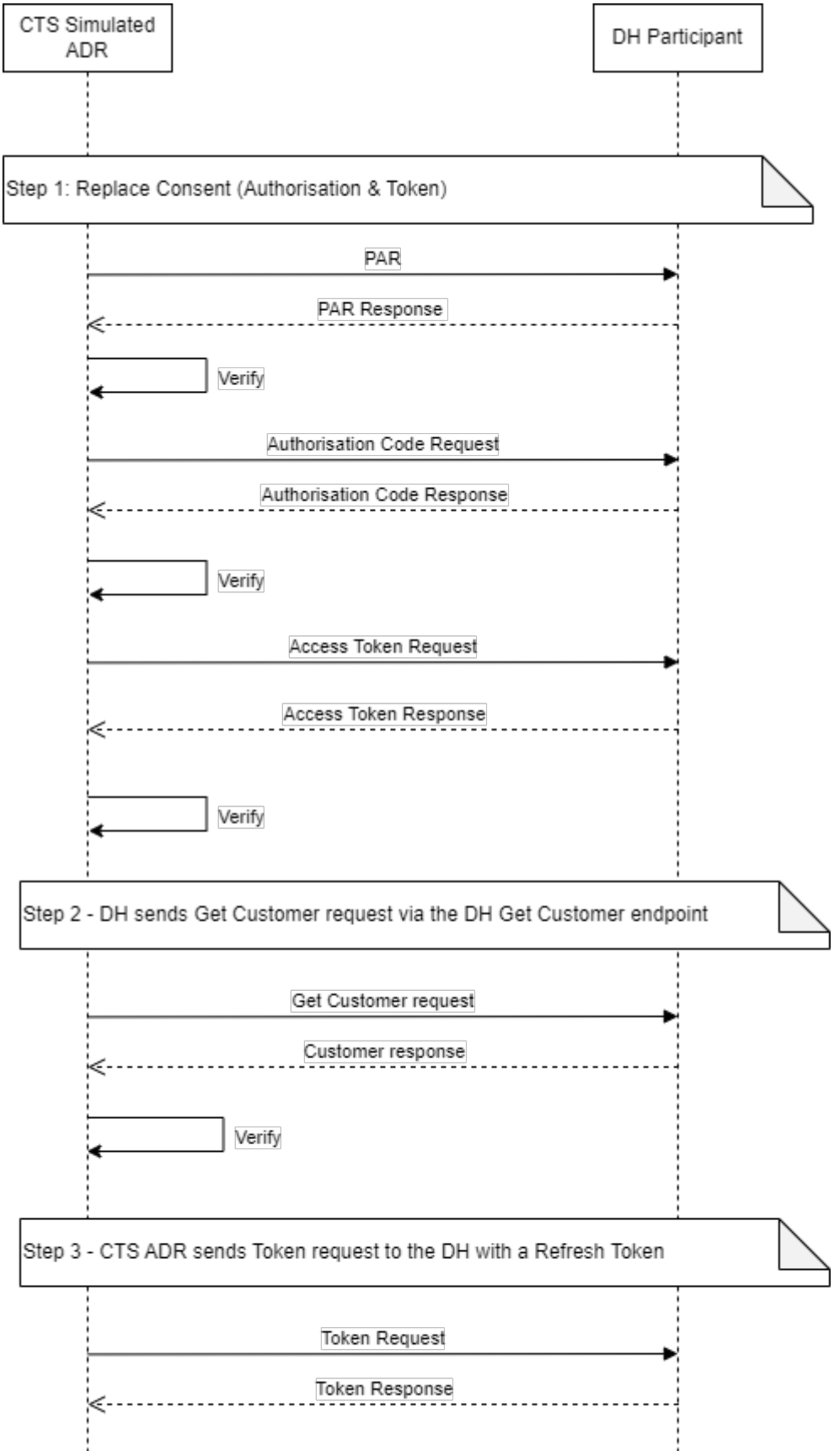
You have passed the PAR flow when you **can**:

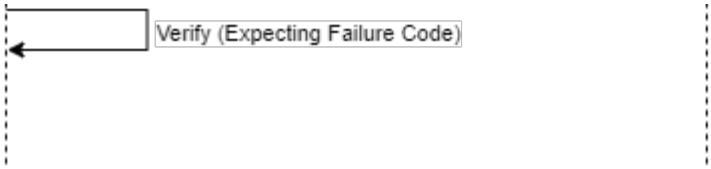
- Return a valid response when the CTS Simulated ADR sends you a PAR.
- Return a payload when you receive a call from the CTS Simulated ADR to the Customer endpoint using the established consent.
- Return a response with an error code when the CTS Simulated ADR calls your Token endpoint with a Refresh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

3.12.5 Scenario High Level Test Steps

1. **CTS Simulated ADR sends a PAR via the DH Pushed Authorisation endpoint**
    a. CTS Simulated ADR sends a PAR (this time containing the cdr arrangement id from First Consent) with the request object to the Participant DH via the Pushed Authorisation endpoint.

   b. Participant DH validates the CTS Simulated ADR third PAR and responds with request_uri.
   c. CTS verifies in the Participant DH's PAR response, that request_uri is contained in the response.
2. **CTS Simulated ADR sends an authorisation code request via the DH Authorisation endpoint**
   a. CTS Simulated ADR sends a request to the Participant DH via the Authorisation endpoint for the same user.
   b. Participant DH validates the CTS Simulated ADR authorisation request, verifying that the CTS Simulated ADR software product is registered with the Participant DH and responds via the Redirect URI with code and state.
   c. CTS verifies the Participant DH response.
3. **CTS Simulated 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, an ID Token and Refresh Token (for ongoing consents only).
4. **CTS Simulated 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.
5. **CTS Simulated 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 First 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 and expects an error response from the DH


CTS Simulated ADR
DH Participant
Step 1: Replace Consent (Authorisation & Token)
PAR
PAR Response
Verify
Authorisation Code Request
Authorisation Code Response
Verify
Access Token Request
Access Token Response
Verify
Step 2 - DH sends Get Customer request via the DH Get Customer endpoint
Get Customer request
Customer response
Verify
Step 3 - CTS ADR sends Token request to the DH with a Refresh Token
Token Request
Token Response



3.13 Data Recipient Initiated Token Revocation

3.13.1 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 (ADR to DH).

3.13.2 Scenario Conditions

Amended consent is revoked.

3.13.3 Endpoints

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

Endpoint	Description	Method
Token Revocation **ADR 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/#security-endpoints

3.13.4 Scenario Results

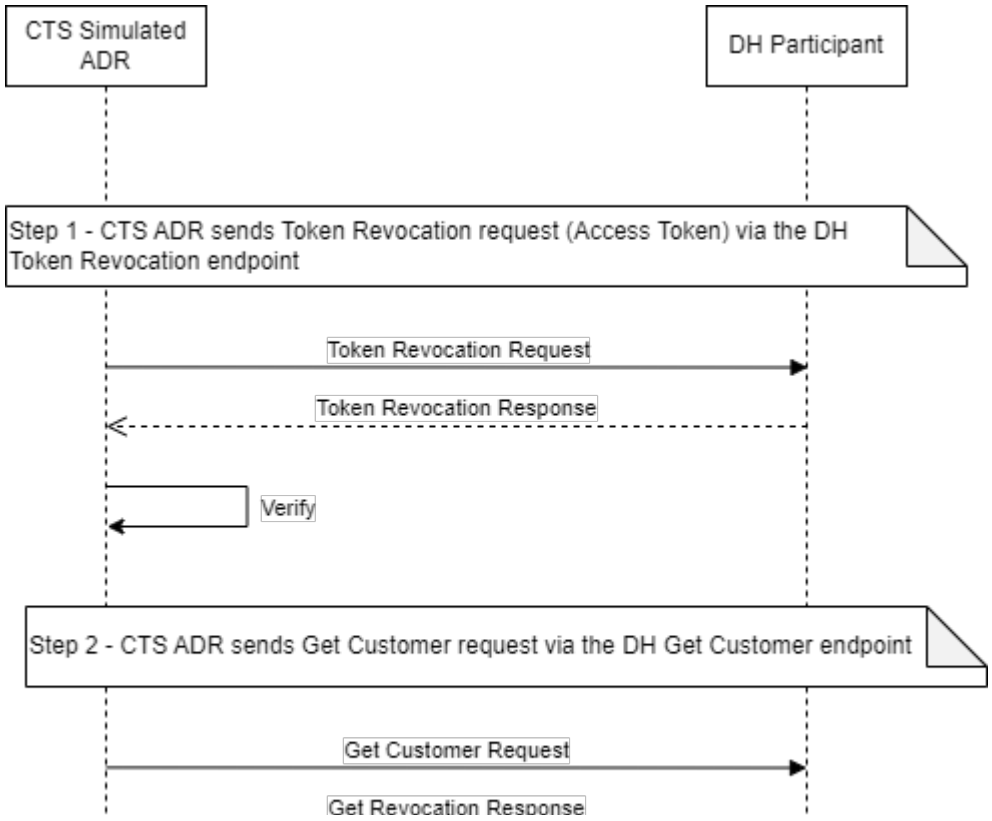
Pass

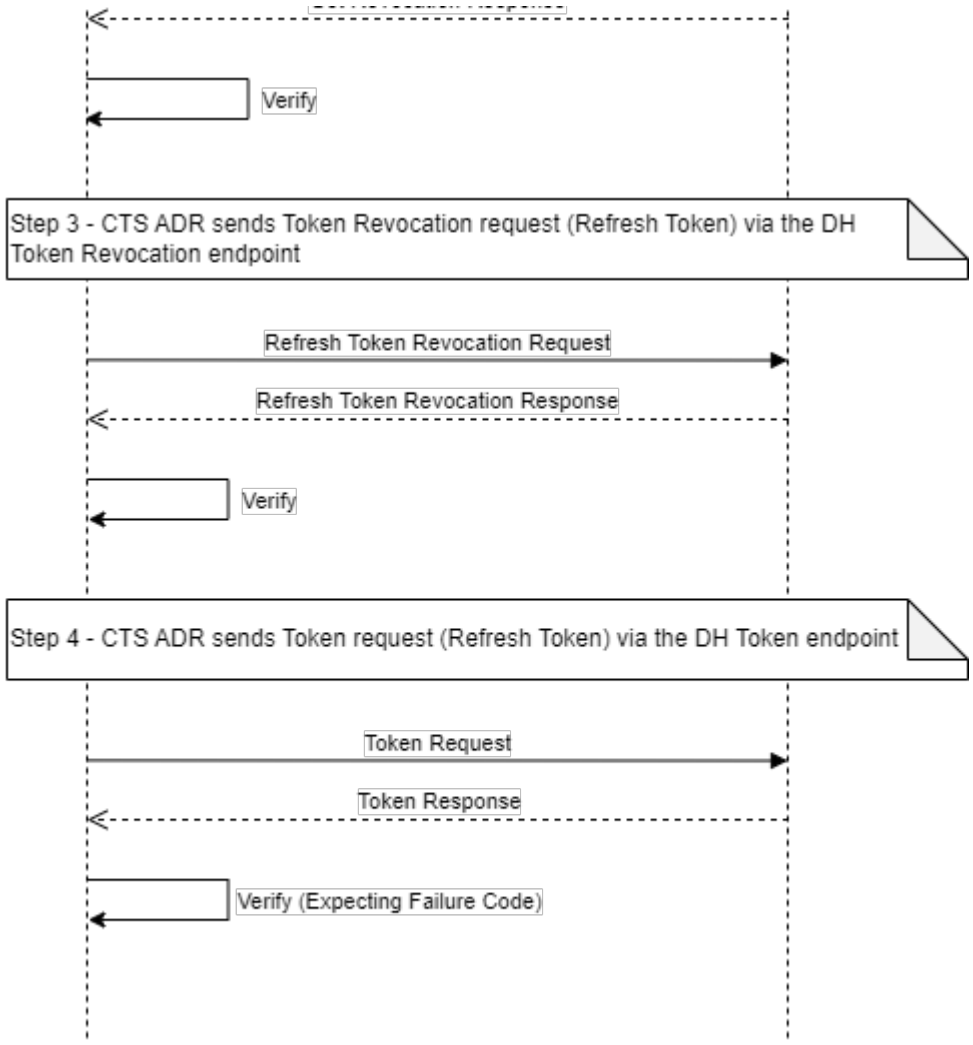
You have passed the ADR initiated token revocation test when you **can**:

- 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 for the Get Customer request after the Access Token is revoked.

3.13.5 Scenario High Level Test Steps

1. **CTS Simulated ADR sends a token revocation request (Access Token) via the DH Token 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.
2. **CTS Simulated ADR sends a 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.
    c. CTS verifies the Participant DH Get Customer Response and expects an error response from the DH
3. **CTS Simulated ADR sends a 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.
4. **CTS Simulated ADR sends a 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 and expects an error response from the DH





3.14 Data Holder Initiated Arrangement Revocation

3.14.1 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.

3.14.2 Scenario Conditions

A CDR Arrangement Id was issued to the CTS ADR by the DH.

3.14.3 Endpoints

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

Endpoint	Description	Method
Arrangement Revocation **DH to ADR**	Participant DH sends a request, using their CDR Arrangement Id, to the CTS Simulated ADR to withdraw an arrangement consent	POST

Link to specifications

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

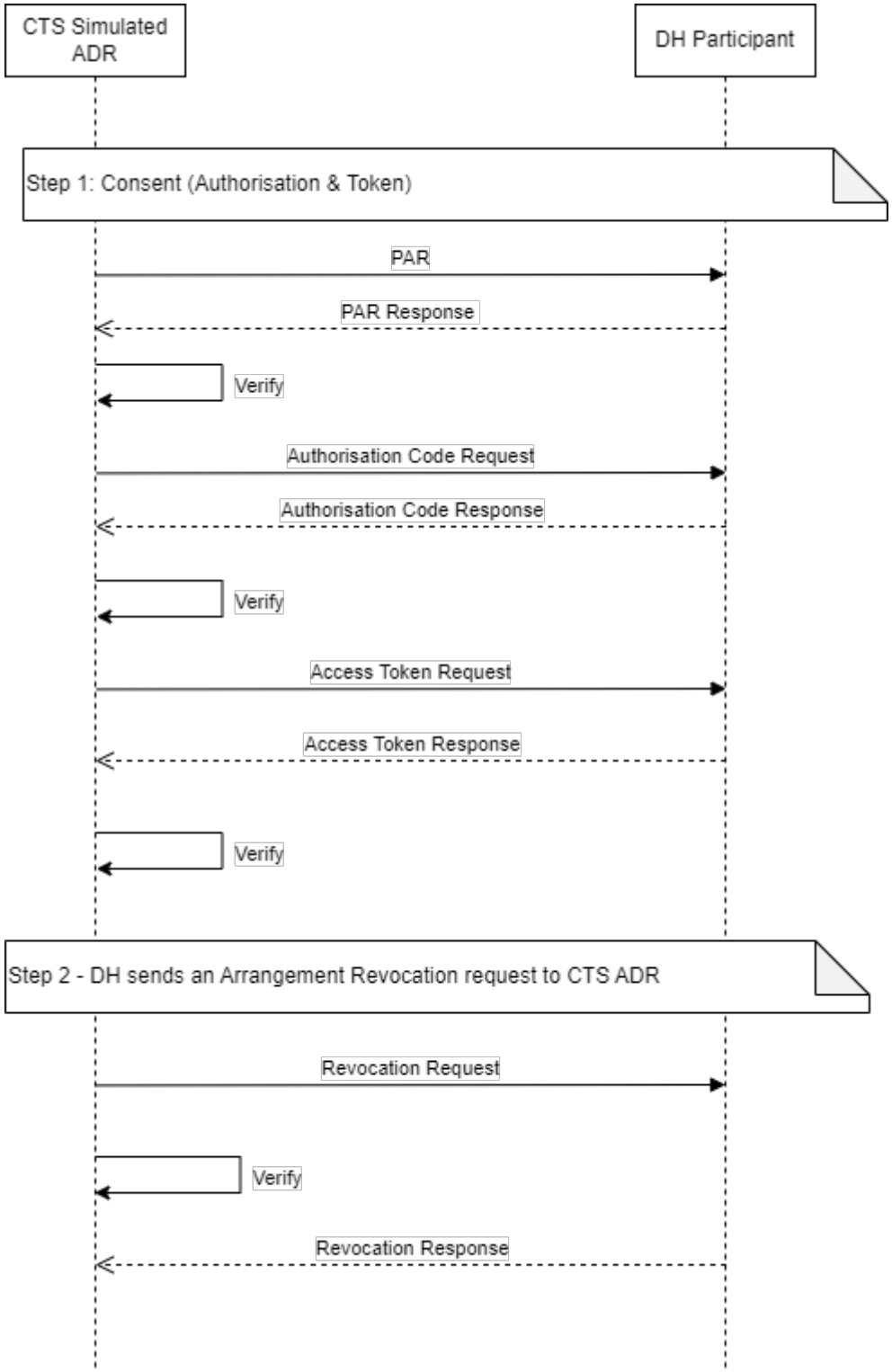
3.14.4 Scenario Results

Pass

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

3.14.5 Scenario High Level Test Steps

1. **CTS Simulated ADR sends a 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 fourth PAR and responds with request_uri.
    c. CTS verifies in the Participant DH's PAR response, that request_uri is contained in the response.
2. **CTS Simulated ADR sends an authorisation code request via the DH Authorisation endpoint**
    a. CTS Simulated ADR sends a request to the Participant DH via the Authorisation endpoint for the same user.
    b. Participant DH validates the CTS Simulated ADR authorisation request, verifying that the CTS Simulated ADR software product is registered with the Participant DH and responds via the Redirect URI with code and state.
    c. CTS verifies the Participant DH response.
3. **CTS Simulated 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, an ID Token and Refresh Token (for ongoing consents only).
4. **DH sends an arrangement revocation request to the CTS Simulated 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.


CTS Simulated ADR
DH Participant
Step 1: Consent (Authorisation & Token)
PAR
PAR Response
Verify
Authorisation Code Request
Authorisation Code Response
Verify
Access Token Request
Access Token Response
Verify
Step 2 - DH sends an Arrangement Revocation request to CTS ADR
Revocation Request
Verify
Revocation Response

3.15 Ensure Client Assertion Data in Requests

3.15.1 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 the CTS Simulated ADR.

3.15.2 Scenario Conditions

Not Applicable.

3.15.3 Endpoints

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

Endpoint	Description	Method
Authorisation	CTS Simulated ADR requests with the Participant DH via the Authorisation endpoint	GET
Token	CTS Simulated ADR exchanges their code for a Token from the Participant DH via the Token endpoint	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 specification

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

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

https://consumerdatastandardsaustralia.github.io/standards/#tokens

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

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

3.15.4 Scenario Results

Pass

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

3.15.5 Scenario High Level Test Steps

1. **CTS Simulated ADR sends a Pushed Authorisation Request (PAR) via the DH Pushed Authorisation endpoint**
    a. CTS Simulated ADR sends a Pushed Authorisation Request with the request object to the DH via the Pushed Authorisation endpoint
    b. DH validates the CTS Simulated ADR Pushed Authorisation Request (PAR), and responds with request_uri
    c. CTS verifies that in the DH PAR response that request_uri is contained in the response.
2. **CTS Simulated ADR requests authorisation via the DH Authorisation endpoint**
    a. CTS Simulated ADR sends an Authorisation request to the DH via the Authorisation 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 the Authorisation code and state.
    c. CTS verifies the DH Authorisation response.
3. **CTS Simulated ADR sends a token request without a client assertion**
    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 Simulated ADR token request and returns a response.
    c. CTS verifies the DH token response with the below errors:
    EITHER 'HTTP Status code of 400 or 401, and contain a response body of "invalid_client"
    OR HTTP Status code of 400 and contains a response body of "invalid_request".
4. **CTS Simulated ADR sends a token request with a 'aud' in the client assertion**
    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 Simulated ADR token request and returns a response.
    c. CTS verifies the DH token response with the below errors:
    EITHER 'HTTP Status code of 400 or 401, and contain a response body of "invalid_client"
    OR HTTP Status code of 400 and contains a response body of "invalid_request".
5. **CTS Simulated ADR sends a token request missing 'aud' in the client assertion**
    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 Simulated ADR token request and returns a response.
    c. CTS verifies the DH token response with the below errors:
    EITHER 'HTTP Status code of 400 or 401, and contain a response body of "invalid_client"
    OR HTTP Status code of 400 and contains a response body of "invalid_request".
6. **CTS Simulated ADR sends a token request wrong 'sub' in the client assertion**

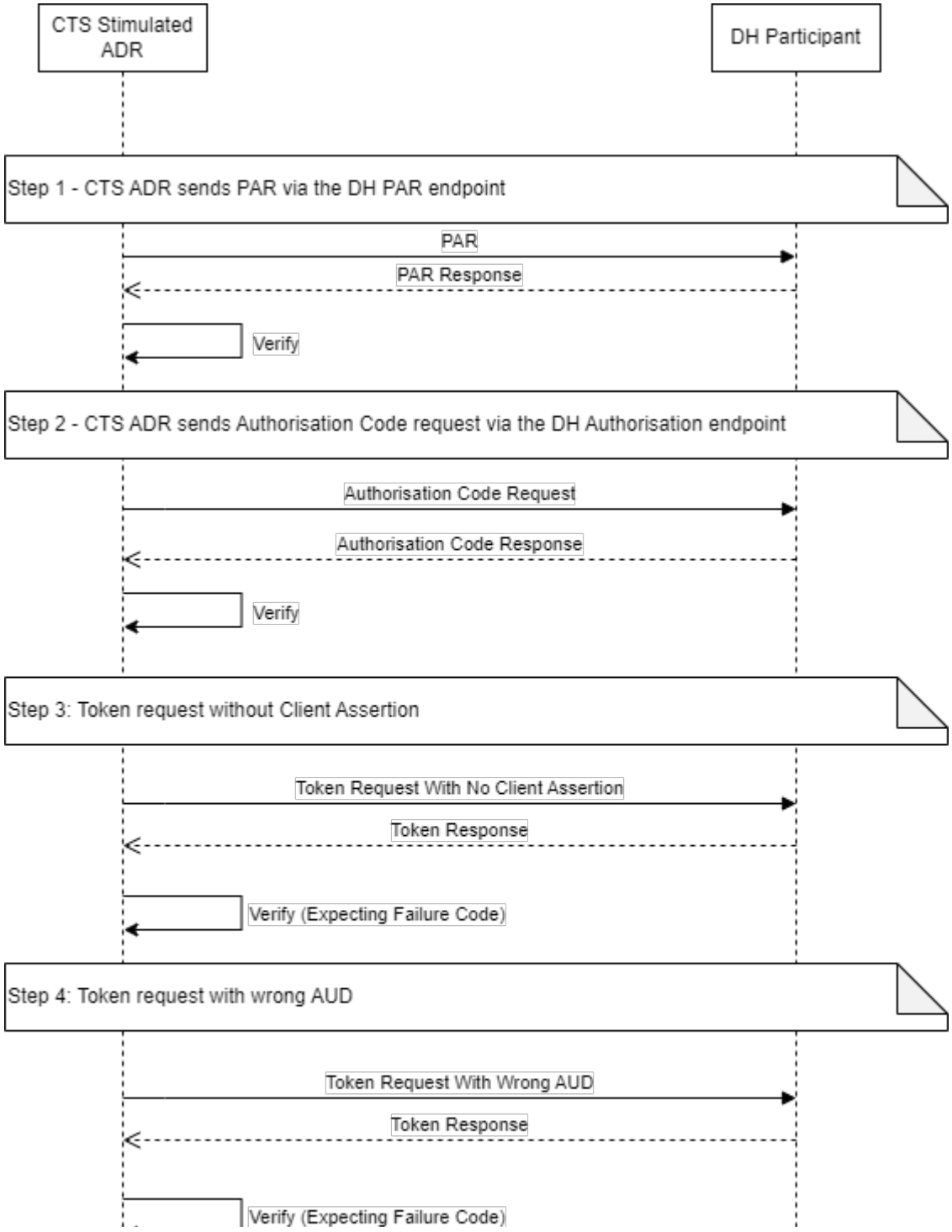
a. CTS Simulated ADR sends a token request **wrong 'sub'** in client assertion to the DH via the Token endpoint, exchanging their Authorisation code for a Token.
b. DH validates the CTS Simulated ADR Token request and returns a response.
c. CTS verifies the DH token response with the below errors:
EITHER 'HTTP Status code of 400 or 401, and contain a response body of "invalid_client"
OR HTTP Status code of 400 and contains a response body of "invalid_request".

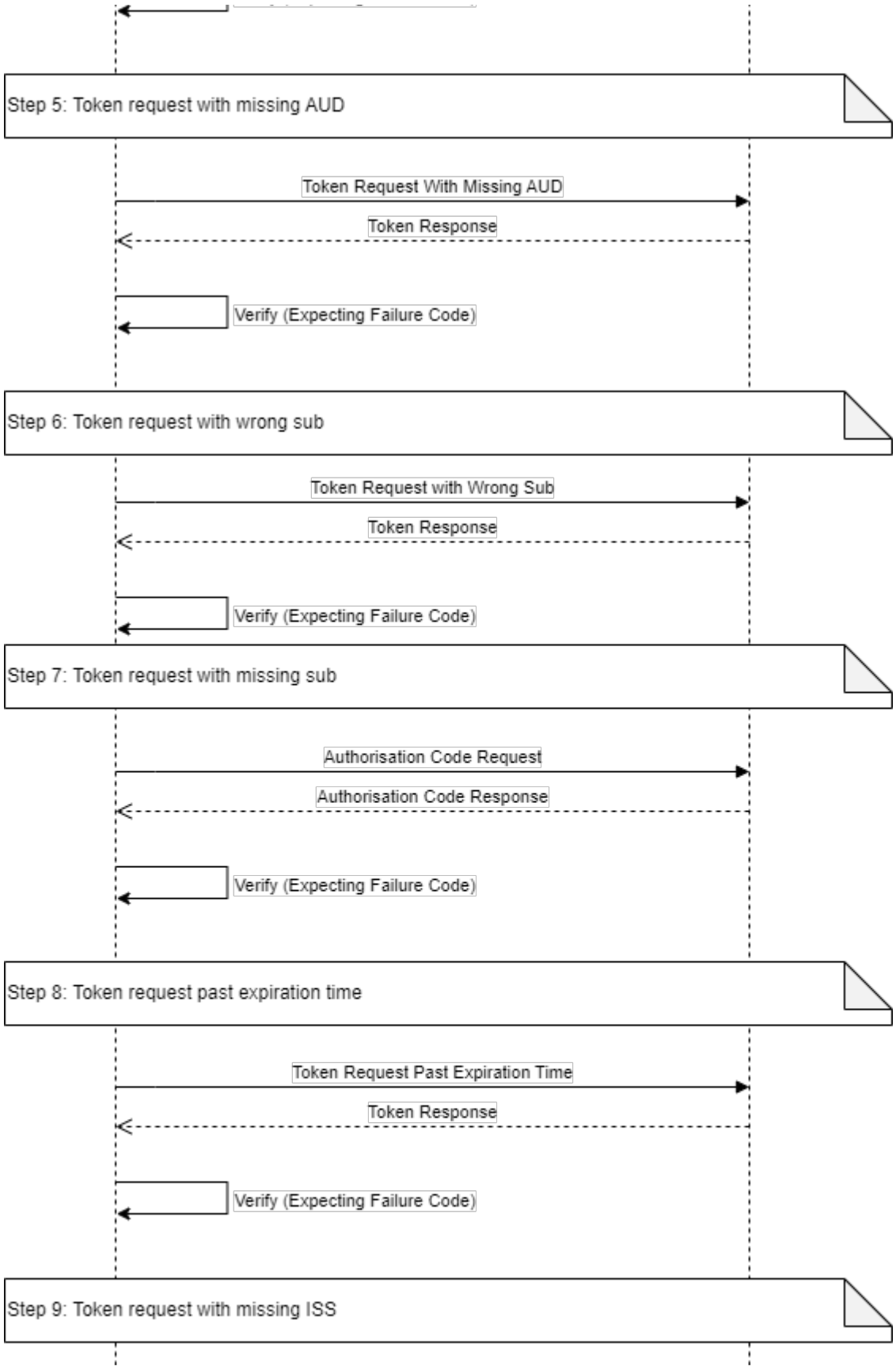
7. **CTS Simulated ADR sends a token request with a missing 'sub' in the client assertion**
   a. CTS Simulated ADR sends a token request with a **missing 'sub'** in client assertion to the DH via the Token endpoint, exchanging their Authorisation code for a Token.
   b. DH validates the CTS Simulated ADR token request and returns a response.
   c. CTS verifies the DH token response with the below errors:
   EITHER 'HTTP Status code of 400 or 401, and contain a response body of "invalid_client"
   OR HTTP Status code of 400 and contains a response body of "invalid_request".
8. **CTS Simulated ADR sends a token request with an expired client assertion**
   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 Simulated ADR token request and returns a response.
   c. CTS verifies the DH token response with the below errors:
   EITHER 'HTTP Status code of 400 or 401, and contain a response body of "invalid_client"
   OR HTTP Status code of 400 and contains a response body of "invalid_request".
9. **CTS Simulated ADR sends a token request with a missing 'iss' in the client assertion**
   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 Simulated ADR Token request and returns a response.
   c. CTS verifies the DH token response with the below errors:
   EITHER 'HTTP Status code of 400 or 401, and contain a response body of "invalid_client"
   OR HTTP Status code of 400 and contains a response body of "invalid_request".
10. **CTS Simulated ADR sends a token request with a wrong 'iss' in the client assertion**
    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 Simulated ADR token request and returns a response.
    c. CTS verifies the DH token response with the below errors:
    EITHER 'HTTP Status code of 400 or 401, and contain a response body of "invalid_client"
    OR HTTP Status code of 400 and contains a response body of "invalid_request".
11. **CTS Simulated ADR sends a token request with a missing 'alg' in the client assertion**
    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 Simulated ADR token request and returns a response.
    c. CTS verifies the DH token response with the below errors:
    EITHER 'HTTP Status code of 400 or 401, and contain a response body of

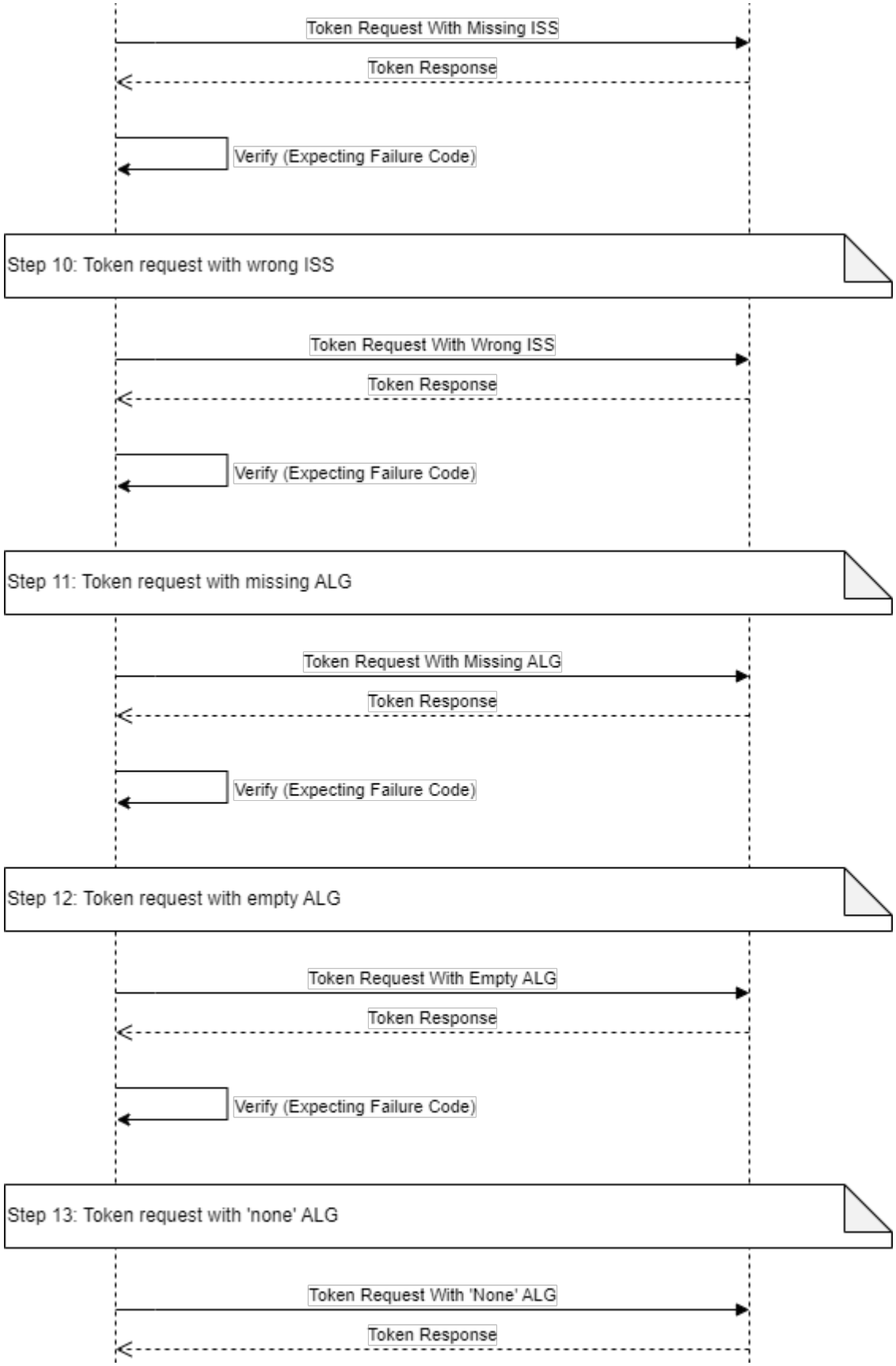
"invalid_client"
OR HTTP Status code of 400 and contains a response body of "invalid_request".

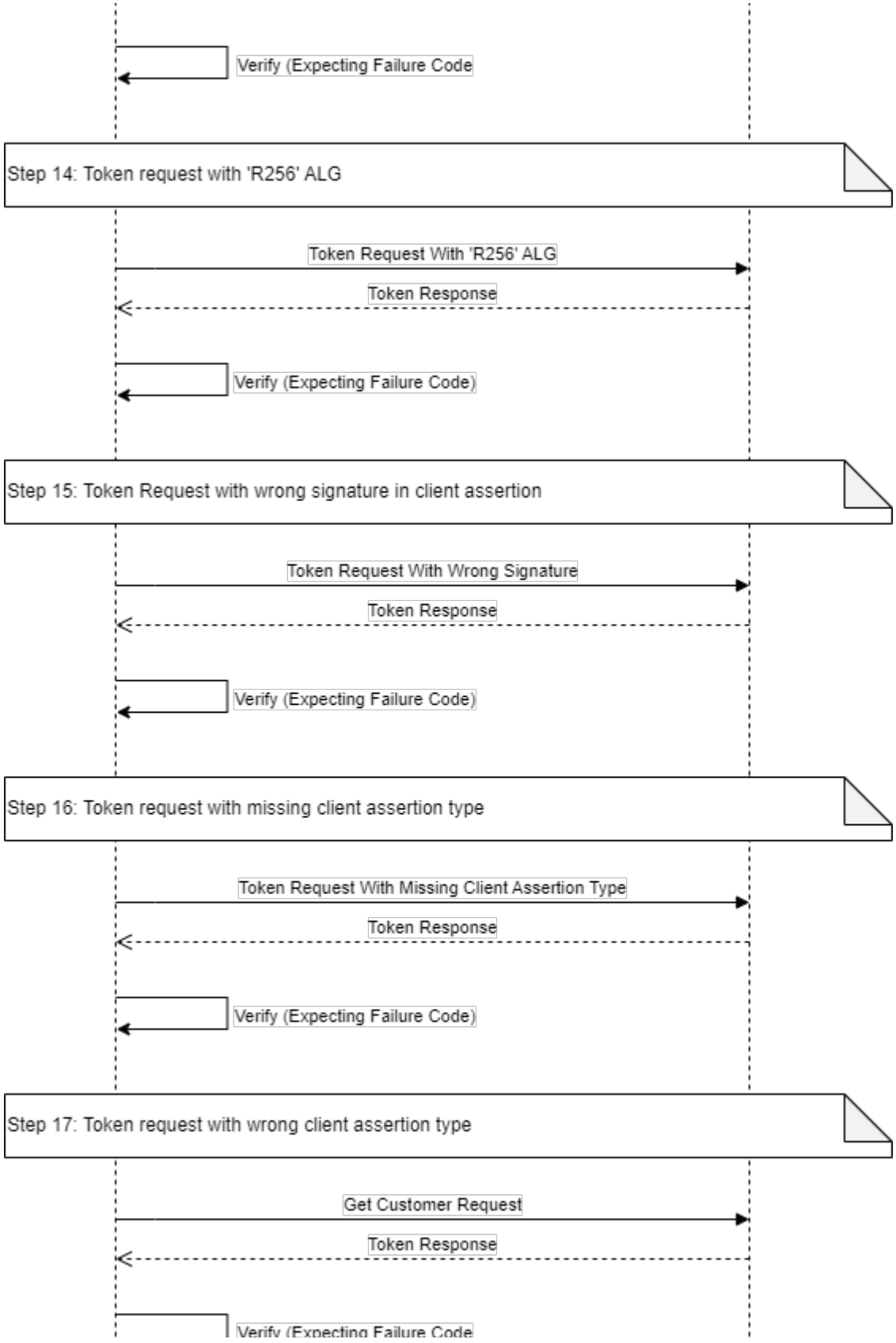
12. **CTS Simulated ADR sends a token request with an empty 'alg' in the client assertion**
    a. CTS Simulated ADR sends a token request with an **empty 'alg'** in client assertion to the DH via the Token endpoint, exchanging their Authorisation code for a Token.
    b. DH validates the CTS Simulated ADR token request and returns a response.
    c. CTS verifies the DH token response with the below errors:
    EITHER 'HTTP Status code of 400 or 401, and contain a response body of "invalid_client"
    OR HTTP Status code of 400 and contains a response body of "invalid_request".
13. **CTS Simulated ADR sends a token request with a 'alg' value of 'none' in the client assertion**
    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 Simulated ADR token request and returns a response.
    c. CTS verifies the DH token response with the below errors:
    EITHER 'HTTP Status code of 400 or 401, and contain a response body of "invalid_client"
    OR HTTP Status code of 400 and contains a response body of "invalid_request".
14. **CTS Simulated ADR sends a token request with a 'alg' value of 'RS256' in the client assertion**
    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 Simulated ADR token request and returns a response.
    c. CTS verifies the DH token response with the below errors:
    EITHER 'HTTP Status code of 400 or 401, and contain a response body of "invalid_client"
    OR HTTP Status code of 400 and contains a response body of "invalid_request".
15. **CTS Simulated ADR sends a token request with a wrong signature in the client assertion**
    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 Simulated ADR token request and returns a response.
    c. CTS verifies the DH token response with the below errors:
    EITHER 'HTTP Status code of 400 or 401, and contain a response body of "invalid_client"
    OR HTTP Status code of 400 and contains a response body of "invalid_request".
16. **CTS Simulated ADR sends a token request with a missing 'client_assertion_type'**
    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 Simulated ADR token request and returns a response.
    c. CTS verifies the DH token response with the below errors:
    EITHER 'HTTP Status code of 400 or 401, and contain a response body of "invalid_client"
    OR HTTP Status code of 400 and contains a response body of "invalid_request".
17. **CTS Simulated ADR sends a token request with a wrong 'client_assertion_type'**

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 Simulated ADR token request and returns a response.
c. CTS verifies the DH token response with the below errors:
   EITHER 'HTTP Status code of 400 or 401, and contain a response body of "invalid_client"
   OR HTTP Status code of 400 and contains a response body of "invalid_request".




Step 5: Token request with missing AUD
Token Request With Missing AUD
Token Response
Verify (Expecting Failure Code)
Step 6: Token request with wrong sub
Token Request with Wrong Sub
Token Response
Verify (Expecting Failure Code)
Step 7: Token request with missing sub
Authorisation Code Request
Authorisation Code Response
Verify (Expecting Failure Code)
Step 8: Token request past expiration time
Token Request Past Expiration Time
Token Response
Verify (Expecting Failure Code)
Step 9: Token request with missing ISS


Token Request With Missing ISS
Token Response
Verify (Expecting Failure Code)
Step 10: Token request with wrong ISS
Token Request With Wrong ISS
Token Response
Verify (Expecting Failure Code)
Step 11: Token request with missing ALG
Token Request With Missing ALG
Token Response
Verify (Expecting Failure Code)
Step 12: Token request with empty ALG
Token Request With Empty ALG
Token Response
Verify (Expecting Failure Code)
Step 13: Token request with 'none' ALG
Token Request With 'None' ALG
Token Response


Verify (Expecting Failure Code
Step 14: Token request with 'R256' ALG
Token Request With 'R256' ALG
Token Response
Verify (Expecting Failure Code)
Step 15: Token Request with wrong signature in client assertion
Token Request With Wrong Signature
Token Response
Verify (Expecting Failure Code)
Step 16: Token request with missing client assertion type
Token Request With Missing Client Assertion Type
Token Response
Verify (Expecting Failure Code)
Step 17: Token request with wrong client assertion type
Get Customer Request
Token Response
Verify (Expecting Failure Code

3.16 Retrieve and Update Client Registration

3.16.1 Purpose

The ability for a Participant DH to demonstrate that they update a registration of the CTS simulated ADR's software product. The CTS simulated ADR will attempt to modify and retrieve the registration with the Participant DH.

3.16.2 Scenario Conditions

A valid client registration exists that can be updated and retrieved.

3.16.3 Endpoints

See also Endpoints used in Data Holder Test Scenario

Endpoint	Description	Method
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/#dcr-apis

3.16.4 Scenario Results

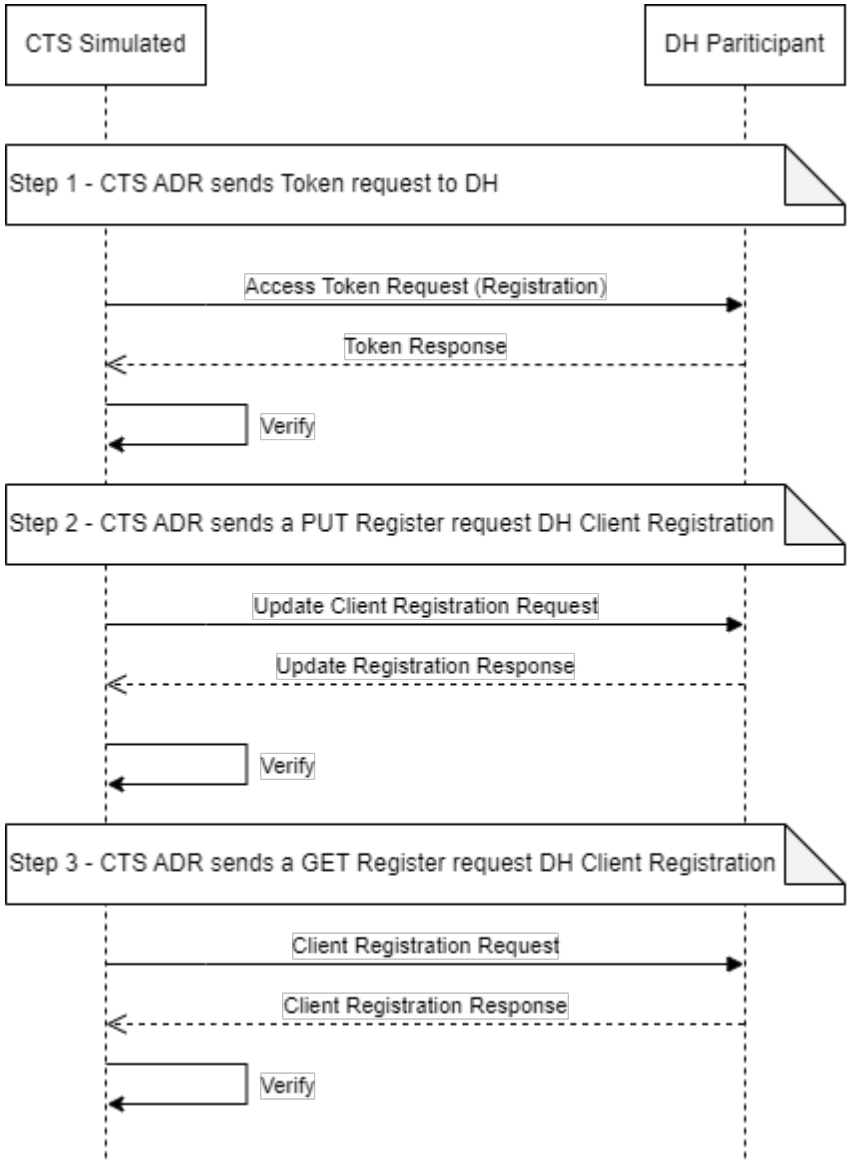
Pass

You have passed the scenario test when you can process both GET and PUT requests to the registration endpoint.

3.16.5 Scenario High-Level Test Steps

1. **CTS Simulated ADR sends a 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.
2. **CTS Simulated ADR sends a PUT register request via the Update Data Recipient Registration endpoint**
   a. CTS Simulated ADR sends a PUT registration request containing the Bearer Token from step 1, 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.
3. **CTS Simulated ADR sends a GET register request via the Get OAuth Client Registration endpoint**
   a. CTS Simulated ADR sends a GET register request containing the Bearer Token from step 1, to the Participant DH 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.



3.17 Removed Software Product

3.17.1 Purpose

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

3.17.1.1 Business Context

The CDR 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 status changes within 5 minutes of the change occurring on the CDR Register.

As per the CDS, when a software product status changes to `Removed`, a DH:

- Must not continue to disclose CDR Data
- Must not continue to facilitate consent authorization
- Must invalidate consents
- Must clean up 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/#participant-statuses

3.17.2 Scenario Conditions

Removed Software Product scenario is dependent on the consent created in Client Assertion scenario.

3.17.3 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 Statuses	Participant DH requests the data recipient status from the CTS Simulated Register via the Get Data Recipient Statuses 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 Statuses	Participant DH requests the software product status from the CTS Simulated Register via the Get Software Product Statuses endpoint	GET

Arrangement Revocation **ADR to DH**	CTS Simulated ADR sends a request, using their CDR Arrangement Id, to the Participant DH to withdraw Arrangement Consent	POST

Link to specifications

https://consumerdatastandardsaustralia.github.io/standards/#introduction

3.17.4 Scenario Results

Pass

You have passed the Register Status tests when an ADR software product status is `REMOVED` and you:

- Poll CTS Simulated Register status within 5 minutes of CTS changing the Software Product status to 'REMOVED'.
- Do not disclose CDR data.
- Do not facilitate consent withdrawal.

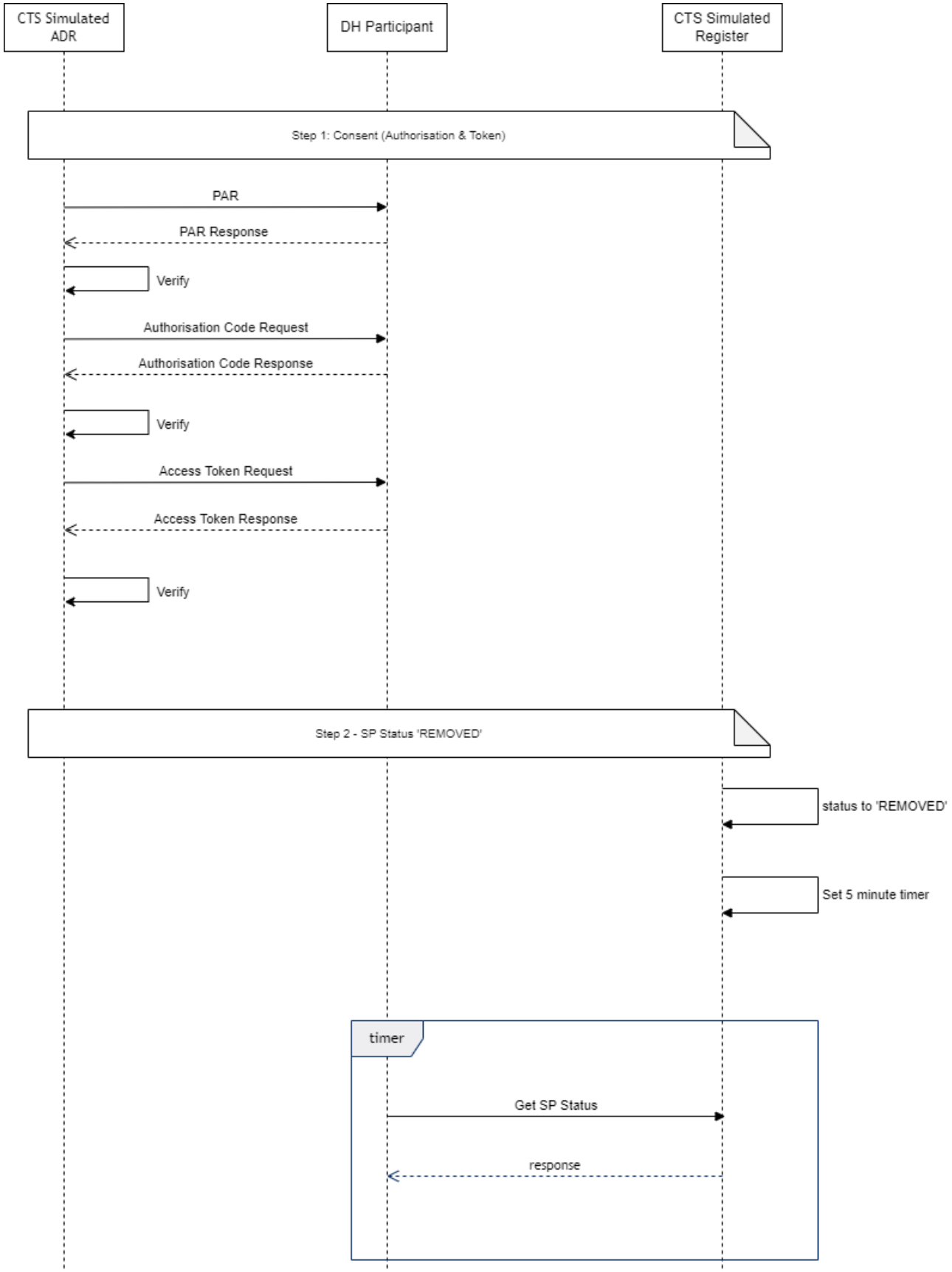
3.17.5 Scenario High-Level Test Steps

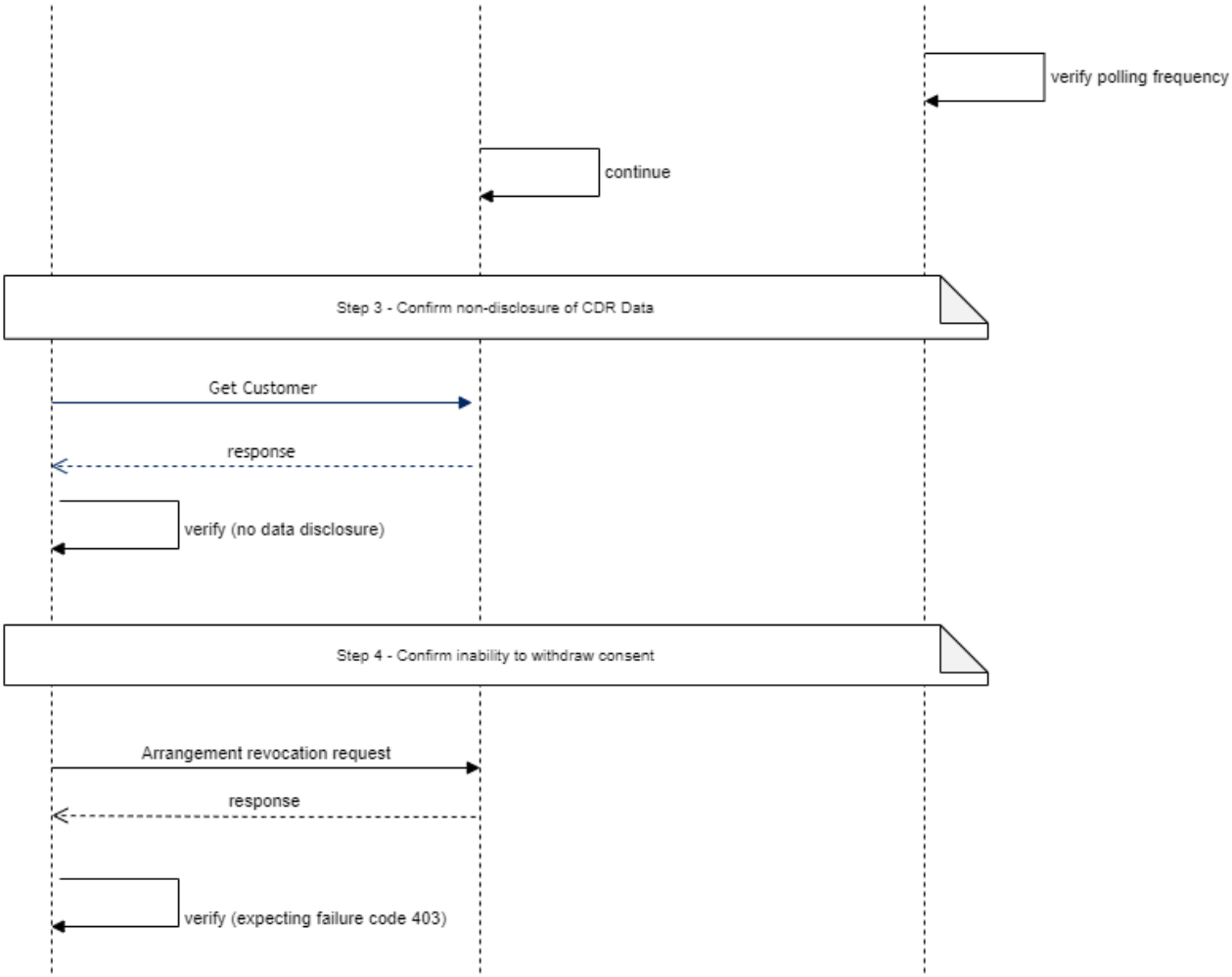
1. **CTS Simulated ADR sends a PAR via the DH Pushed Authorisation endpoint**
    a. CTS Simulated ADR sends a PAR with the request object to the Participant DH via the Pushed Authorisation endpoint.
    b. Participant DH validates the CTS Simulated ADR second PAR and responds with request_uri.
    c. CTS verifies in the Participant DH's PAR response, that request_uri is contained in the response.
2. **CTS Simulated ADR sends an authorisation code request via the DH Authorisation endpoint**
    a. CTS Simulated ADR sends a request to the Participant DH via the Authorisation endpoint for the same user.
    b. Participant DH validates the CTS Simulated ADR authorisation request, verifying that the CTS Simulated ADR software product is registered with the Participant DH and responds via the Redirect URI with code and state.
    c. CTS verifies the Participant DH response.
3. **CTS Simulated 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, an ID Token and Refresh Token (for ongoing consents only)
4. **CTS changes the ADR software product status in the CTS Simulated Register from 'ACTIVE' to 'REMOVED'**
    a. CTS awaits incoming requests (may take several minutes). After a Register Polling Request has been received, a continue button will appear. Please ensure to continue

the scenario before the 5 minute timer expires, else the step (and scenario), will fail.

b. Participant DH sends a 'software product status' request to the CTS Simulated Register via the Get Software Product Statuses endpoint.
c. CTS Simulated Register returns a valid response OR
d. Participant DH sends a 'data recipients' request to the CTS Simulated Register via the Get Data Recipients endpoint.
e. CTS Simulated Register returns a valid response.
f. Verify Register frequency polling.

5. **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.
   b. Participant DH validates the CTS Simulated ADR request and returns a response.
   c. CTS verifies the Participant DH Get Customer Response and expects an error response from the DH.
6. **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.


CTS Simulated ADR
DH Participant
CTS Simulated Register
Step 1: Consent (Authorisation & Token)
PAR
PAR Response
Verify
Authorisation Code Request
Authorisation Code Response
Verify
Access Token Request
Access Token Response
Verify
Step 2 - SP Status 'REMOVED'
status to 'REMOVED'
Set 5 minute timer
timer
Get SP Status
response



3.17.5.1 Technical note

The CTS will validate that the participant responses conform to the 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

4 Endpoints used in Data Holder Scenarios

A CTS Conformance ID is assigned to a Data Holder Brand upon enrolment in the CTS. This ID serves as a unique identifier while executing the CTS and must be included in the CTS Simulated Register and CTS Simulated ADR URLs to identify the Data Holder Brand. These URLs can be accessed through the domains `api.cts.cdr.gov.au` or `secure.api.cts.cdr.gov.au`.

The endpoints and URLs used in the Data Holder test plan are listed in the table below:

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 (ADR)	Data Recipient	Participant DH requests the JWKS from the CTS Simulated ADR via the JWKS endpoint	`/cts/{conformanceId-guid}/dr/jwks`
	Get JWKS (Register)	Register	Participant DH requests the JWKS from the CTS Simulated Register via the JWKS endpoint	`/cts/{conformanceId}/register/cdr-register/v1/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 Statuses 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 Statuses 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	**Authorisation**	Data Holder	CTS Simulated ADR requests authorisation with the Participant DH via the Authorisation 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 Revocation endpoint, 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`

5 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 their authorised Software Products. A Data Recipient **MUST** be accredited in order to participate in the CDR Eco-system. 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: Authentication Flows - Consumer Data Standards
Authorise / Authorisation	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: Consent – Consumer Data Standards
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. The 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 endpoints. 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-1.27.0/#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 (endpoints). 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 Registrar (the Registrar) to consider in allowing the participant to be active on the Register.
TP	Test Plan
Withdrawal	See **revocation**.

6 Brand vs Conformance ID Infographic

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