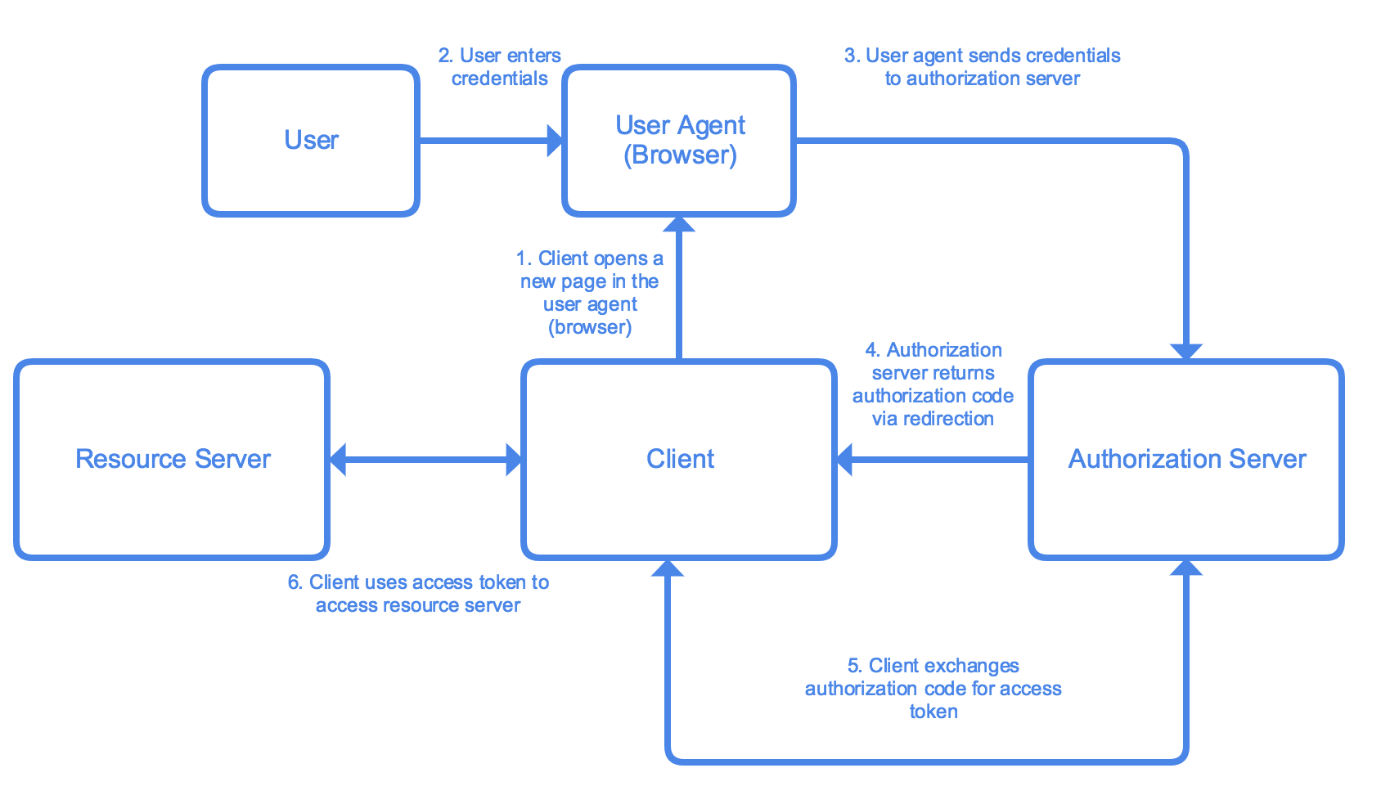
**Authorization data contract**

**Access**

* **API Endpoint**: **N/A (Will be available after creating a web server)**
* **Authentication**: OAuth 2.0 with Bearer token following the structure:



The Authorization Code flow includes the following steps:

1. The **client** prepares a link to the **authorization server** and opens the link for user in a **user agent** (browser). The link includes information that allows the authorization server to identify and respond to the client.
2. User enters their credentials on the new page.
3. Credentials are sent to **authorization server** via the **user agent** (browser).
4. The **authorization server** validates the credentials and redirects user back to the client with an authorization code.
5. The client talks with the **authorization server**, confirms its identify and exchanges the authorization code for an access token and optionally a refresh token.
6. The client uses the access token to access resources on the **resource server**.[[1]](#footnote-1)

* **Rate Limit**: 50 requests per minute to prevent abuse
* **Pagination**: Not applicable for token and authorization requests
* **Schema (using SDMX)**:

|  |
| --- |
| * **<structure>** * **<id>oauth\_data\_contract</id>** * **<name>OAuth Data Contract</name>** * **<attributes>** * **<attribute>** * **<id>client\_id</id>** * **<name>Client ID</name>** * **<type>string</type>** * **</attribute>** * **<attribute>** * **<id>client\_secret</id>** * **<name>Client Secret</name>** * **<type>string</type>** * **</attribute>** * **<attribute>** * **<id>redirect\_uri</id>** * **<name>Redirect URI</name>** * **<type>string</type>** * **</attribute>** * **<attribute>** * **<id>grant\_type</id>** * **<name>Grant Type</name>** * **<type>string</type>** * **</attribute>** * **<attribute>** * **<id>scope</id>** * **<name>Scope</name>** * **<type>string</type>** * **</attribute>** * **<attribute>** * **<id>access\_token</id>** * **<name>Access Token</name>** * **<type>string</type>** * **</attribute>** * **<attribute>** * **<id>refresh\_token</id>** * **<name>Refresh Token</name>** * **<type>string</type>** * **</attribute>** * **<attribute>** * **<id>token\_type</id>** * **<name>Token Type</name>** * **<type>string</type>** * **</attribute>** * **<attribute>** * **<id>expires\_in</id>** * **<name>Expires In</name>** * **<type>integer</type>** * **</attribute>** * **</attributes>** * **</structure>** |

**Structural Quality**

* **Missing Values**: No missing values allowed for **client\_id**, **client\_secret**, and **grant\_type**
* **Data Volume**: Depends on the number of OAuth clients and tokens; expected several records per month (City council employees)
* **Update Frequency**: Real-time as clients register and tokens are issued/revoked
* **Completeness**: 100% for all required attributes
* **Freshness**: Real-time updates as data changes

**Statistical Quality**

* **Anomaly Detection**: Real-time monitoring for unusual access patterns or brute force attempts
* **Outlier Threshold**: Not applicable

**Error Handling**

* **HTTP Status Codes**: Use standard codes (e.g., 200 OK, 400 Bad Request, 401 Unauthorized)
* **Error Messages**: Clear and descriptive error messages for invalid requests, unauthorized access, etc.

**Versioning**

* **API Version**: v1
* **Schema Version**: 1.0

**Support**

* **Documentation**: **N/A**
* **Contact**: **dus15@uclive.ac.nz**

**Service Level Agreement (SLA)**

* **Uptime**: 99.95%
* **Response Time**: Less than 300ms for 95% of requests

**Data Lineage**

* **Source**: Client registration forms, token requests from web and mobile applications
* **Transformations**: Encryption of sensitive data, normalization of scopes

**Data Governance**

* **Data Steward**: Dimitrii Ustinov
* **Retention Policy**: Client data retained indefinitely unless deletion is requested; tokens expire and are deleted according to policy

1. [auth-server-sample/GCF/code/auth.pug at master · michaelawyu/auth-server-sample (github.com)](https://github.com/michaelawyu/auth-server-sample/blob/master/GCF/code/auth.pug) [↑](#footnote-ref-1)