Gestión y Emisión de la Credencial

Emisión de una Nueva Credencial

Este es el proceso completo, paso a paso, que ocurre desde que el ciudadano inicia la solicitud hasta que recibe su credencial.

Paso 1: Inicio de Sesión y Autenticación Fuerte

El ciudadano necesita probar su identidad al sistema de emisión. Esto se logra mediante el flujo de Código de Autorización de OIDC.

1. **Acción del Usuario**: El usuario presiona "Solicitar Licencia Digital" en la wallet `Soy Yo RD` .
2. **Llamada Inicial (Frontend)**: La wallet `Soy Yo RD` genera e invoca una llamada de autorización a `Cuenta Única`. No es una API REST directa, sino una redirección del navegador en la app (o un Custom Chrome Tab / ASWebAuthenticationSession).
   Ejemplo de URL de Autorización (GET):
   ```
   https://auth.cuentaunica.gob.do/auth?
     response_type=code
     &client_id=SOYYO_RD_WALLET_CLIENT_ID
     &scope=openid profile cedula
     &redirect_uri=soyyord://callback
     &state=STATE_STRING_ALEATORIO
     &nonce=NONCE_STRING_ALEATORIO
     &code_challenge=S256_CHALLENGE
     &code_challenge_method=S256
   ```
3. **Autenticación del Usuario**: El usuario se autentica en la interfaz web de `Cuenta Única` (usuario, contraseña, 2FA).
4. **Redirección con Código**: `Cuenta Única` redirige de vuelta a la app `Soy Yo RD` a través de la `redirect_uri` con un código de autorización.
5. **Intercambio del Código por Tokens (Backend-a-Backend)**: La wallet envía el `code` a su propio backend de soporte (o lo hace directamente si es una app confidencial), el cual realiza una llamada **API REST (POST)** al endpoint de token de `Cuenta Única`. Esta

comunicación es segura (backend a backend) y evita exponer los tokens en el lado del cliente.
Llamada a la API de Token (POST):

- **Endpoint:** `https://auth.cuentaunica.gob.do/token`
- **Headers:** `Content-Type: application/x-www-form-urlencoded`
- **Body:**

```
grant_type=authorization_code
&code=CODIGO_RECIBIDO
&redirect_uri=soyyord://callback
&client_id=SOYYO_RD_WALLET_CLIENT_ID
&client_secret=CLIENT_SECRET_DE_LA_APP
&code_verifier=VERIFIER_ORIGINAL
```

6. **Respuesta de la API de Token**: `Cuenta Úunica` responde con un objeto JSON que contiene los tokens de acceso e identidad.
Formato de Respuesta (JSON):

```
{
  "access_token": "eyJhbGciOiJSUzI1NiIsImtpZCI6...",
  "token_type": "Bearer",
  "expires_in": 3600,
  "id_token": "eyJhbGciOiJSUzI1NiIsImtpZCI6..."
}
```

El `id_token` es un **JSON Web Token (JWT)** que, al ser decodificado, contiene la información verificada del ciudadano, como su número de Cédula.

Paso 2: Solicitud Formal de la Credencial

Con la prueba de identidad (el `id_token` o `access_token`), la wallet ahora puede solicitar la credencial al servidor de emisión del INTRANT.

1. **Llamada a la API de Emisión (POST)**: La wallet `Soy Yo RD` realiza una llamada HTTPS a la API de `Inji-Certify`.
   - **Endpoint:** `https://api.intrant.gob.do/v1/issue/driving-license`
   - **Headers:** `Authorization: Bearer <access_token_de_cuenta_unica>`
   - **Body (JSON):**

```
{
  "holderDid": "did:key:z6Mkt...H73tias"
}
```

   - **Información Relevante:**

- El `access_token` en el header autoriza la operación y permite a `Inji-Certify` identificar al solicitante (consultando el endpoint `/userinfo` de Cuenta Única o validando el `id_token`).
- El `holderDid` es el Identificador Descentralizado del ciudadano, generado por la wallet. Será el `id` del sujeto (`credentialSubject`) en la VC.

Paso 3: Verificación Interna y Construcción de la VC

El servidor de emisión ahora tiene todo lo necesario para procesar la solicitud.

1. **Validación de Identidad**: El backend de `Inji-Certify` valida el `access_token` con `Cuenta Única`.
2. **Consulta a la Base de Datos**: Utilizando el número de Cédula extraído del token, el sistema realiza una consulta a la base de datos interna del INTRANT para obtener los datos de la licencia del ciudadano.
   - **Mecanismo**: Conexión de base de datos segura (ej. sobre un túnel VPN) con una consulta tipo `SELECT * FROM licencias WHERE cedula = ?`.
3. **Construcción de la VC**: Si se encuentran datos válidos, `Inji-Certify` ensambla la Credencial Verificable.
   Formato de la VC (JSON-LD):

```
{
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://w3id.org/vdl/v1" // Contexto específico para licencias de conducir
  ],
  "id": "urn:uuid:3978344f-8596-4c3a-a978-8fcaba3903c5",
  "type": ["VerifiableCredential", "Iso18013DriversLicense"],
  "issuer": "did:web:intrant.gob.do", // DID del INTRANT
  "issuanceDate": "2025-10-26T14:23:14Z",
  "expirationDate": "2029-10-26T14:23:14Z",
  "credentialSubject": {
    "id": "did:key:z6Mkt...H73tias", // DID del ciudadano
    "familyName": "Perez",
    "givenName": "Juan",
    "birthDate": "1990-01-15",
    "issuingCountry": "DO",
    "drivingPrivileges": [
      {
        "vehicleCategory": "B",
        "issueDate": "2021-10-26",
        "expiryDate": "2025-10-26"
```

```
      }
    ]
  },
  "proof": { // Este bloque se añade después de firmar
    "type": "Ed25519Signature2020",
    "created": "2025-10-26T14:23:14Z",
    "verificationMethod": "did:web:intrant.gob.do#key-1",
    "proofPurpose": "assertionMethod",
    "proofValue": "z58D3...2k9fV" // La firma digital
  }
}
```

4. **Firma Criptográfica**: El servidor firma el objeto JSON-LD (excluyendo el bloque `proof`) utilizando la clave privada del INTRANT, almacenada en un **Hardware Security Module (HSM)** para máxima seguridad. El resultado es el `proofValue`.

Paso 4: Devolución de la Credencial

El servidor responde a la llamada API del Paso 2.

- **Respuesta de la API de Emisión (Código 200 OK):**
  - **Body (JSON):** El cuerpo de la respuesta es directamente la Credencial Verificable completa, en formato JSON-LD, tal como se describió arriba.

La wallet `Soy Yo RD` recibe este objeto JSON, valida la firma del emisor (resolviendo el `did:web:intrant.gob.do` para obtener la clave pública) y lo almacena de forma segura en el almacenamiento cifrado del dispositivo.

Gestión del Ciclo de Vida: Revocación

La revocación es crítica. No es suficiente con que una credencial tenga una firma válida; debe estar activa.

- **Mecanismo**: **Status List 2021**. El INTRANT mantiene un endpoint público que expone un bitmap (una larga cadena de bits, 0s y 1s). Cada credencial emitida está asociada a un índice en esta lista.
- **Acción de Revocación**: Cuando un oficial del INTRANT revoca una licencia en el sistema, la acción interna es una **llamada API (PUT o PATCH)** al servicio que gestiona la lista de estado para cambiar el bit en el índice correspondiente de `0` (válido) a `1` (revocado).
  - **Endpoint (interno):** `https://api.intrant.gob.do/v1/statuslist/main`
  - **Body (JSON):**

```
{
  "credentialId": "urn:uuid:3978344f-8596-4c3a-a978-8fcaba3903c5",
  "status": "revoked"
}
```

- **API Pública de la Lista de Estado**: Los verificadores acceden a esta lista a través de un endpoint público definido en la propia VC (propiedad `credentialStatus`).
  Llamada del Verificador (GET):
  - **Endpoint:** `https://api.intrant.gob.do/v1/status/1` (ejemplo)
  - **Respuesta:** Un objeto JSON que contiene el bitmap comprimido. El software del verificador lo usa para comprobar el estado de la credencial que está validando.

Esta arquitectura asegura un flujo de emisión robusto, seguro y estandarizado, donde cada componente se comunica a través de APIs bien definidas, utilizando formatos de datos interoperables como JSON, JWT y JSON-LD.

Revisión #2
Creado 2025-06-17 18:54:06 UTC por Tomás Familia
Actualizado 2025-06-17 19:01:21 UTC por Tomás Familia