Analysis: 0PN Interop as Universal Digital Identity Interoperability Layer

Author: Mark Lizar v1 9-2-26

Executive Summary

0PN Interop functions as universal digital identity interoperability infrastructure by providing the missing transparency layer that enables trust decisions across all identity protocols, frameworks, and jurisdictions.

Core Mechanism: Controller-ID first architecture creates protocol-agnostic transparency disclosure that works **before** any identity protocol executes—making 0PN Interop a **pre-protocol interoperability layer** rather than another competing identity protocol.

Universal Property: Every digital identity interaction requires a controller (verifier, relying party, service provider). 0PN Interop standardizes **controller transparency disclosure** independent of which identity protocol or credential format follows.

Why Digital Identity Needs Universal Interoperability

Current Fragmentation Problem

Multiple competing protocols:

- OpenID Connect (authentication)
- OAuth 2.0 (authorization)
- SAML (enterprise SSO)
- W3C Verifiable Credentials (credential format)
- ISO/IEC 18013-5 mDL (mobile driver's license)
- DIDComm (DID-based messaging)
- FIDO/WebAuthn (authentication)
- OpenID4VP (verifiable presentation)
- UMA 2.0 (user-managed authorization)

Each protocol solves part of the problem:

- Authentication (who is this individual?)
- Authorization (what can they access?)
- Credential presentation (what claims can they prove?)
- Key management (how do they control keys?)

None solve the fundamental trust question:

"Do I trust this controller enough to present my identity/credential to them?"

Missing Layer: Controller Transparency

All identity protocols assume:

1. Individual encounters controller (verifier/RP/service)
2. Controller requests credential/authentication

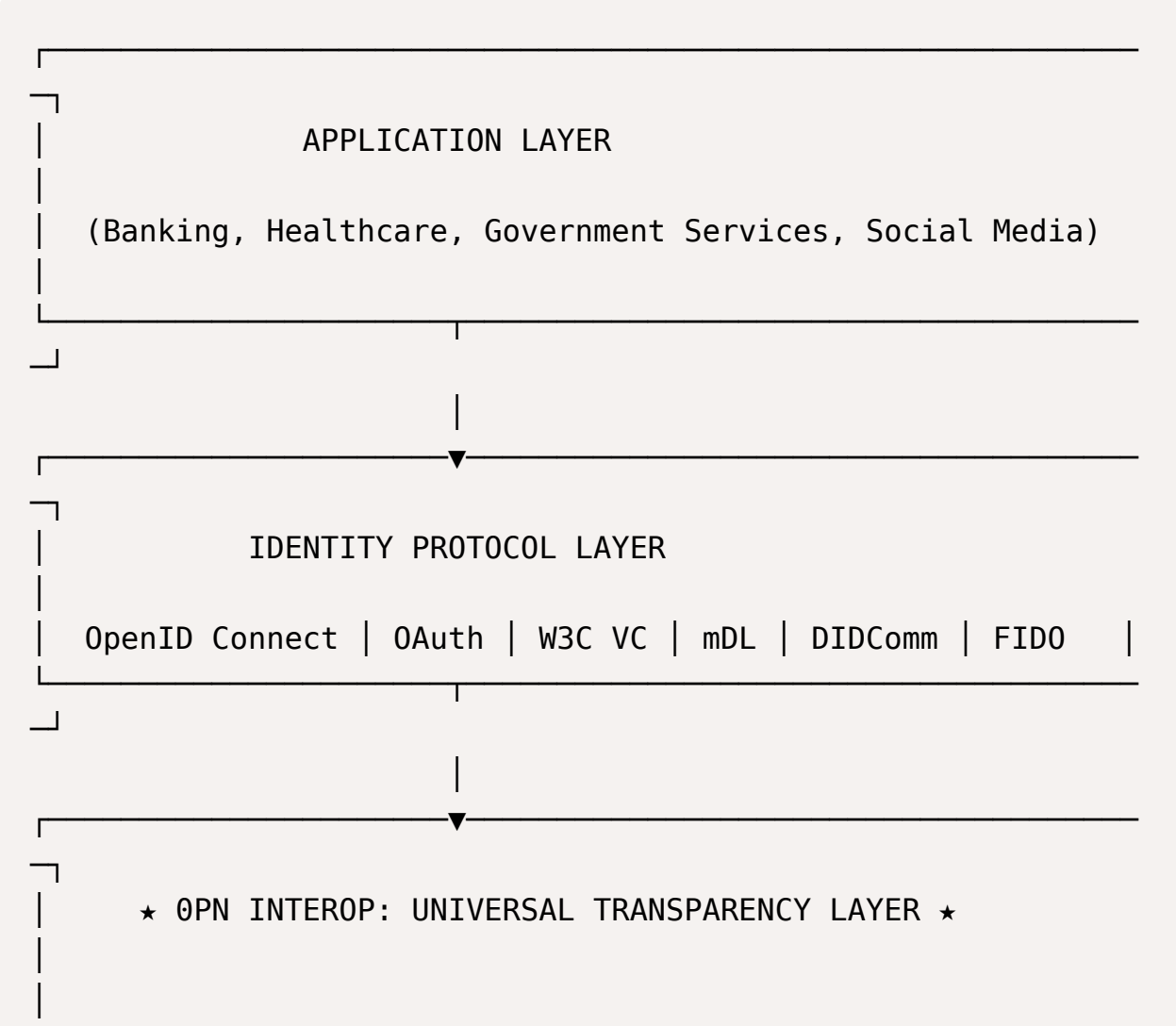
3. Individual presents identity/credential
4. Protocol executes

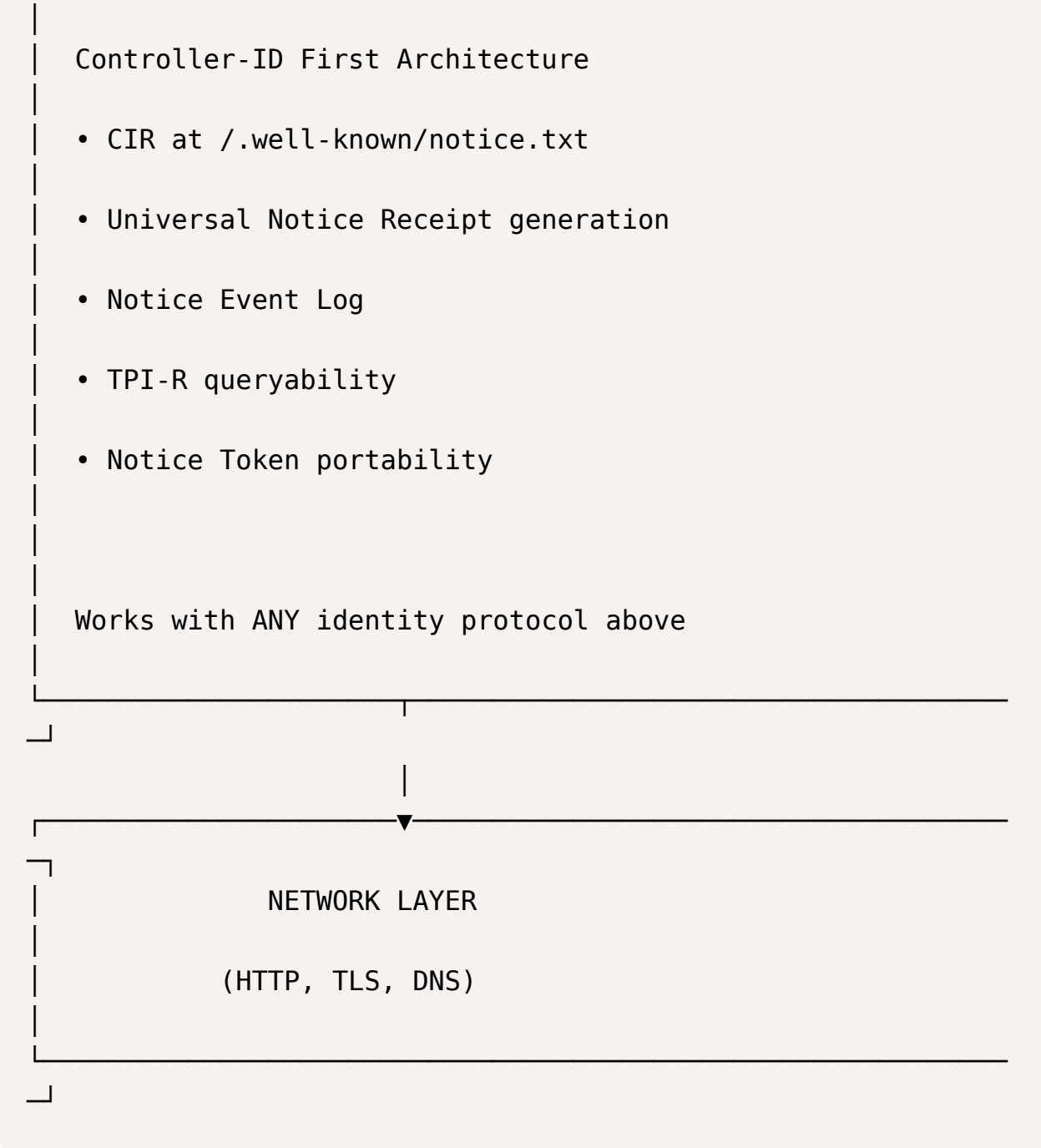
Problem: Individual cannot evaluate trustworthiness of controller **before** identity disclosure.

0PN Interop Solution: Insert controller transparency disclosure **before step 2**.

How 0PN Interop Provides Universal Interoperability

Layer Model





Key Insight: 0PN Interop sits **below** identity protocols, providing universal transparency disclosure that works regardless of which protocol executes next.

Universal Interoperability Mechanisms

1. Protocol-Agnostic Controller Disclosure

Standard Discovery Mechanism

Every controller publishes CIR at standard location:

```
https://example.com/.well-known/notice.txt
```

CIR Structure (Protocol-Agnostic):

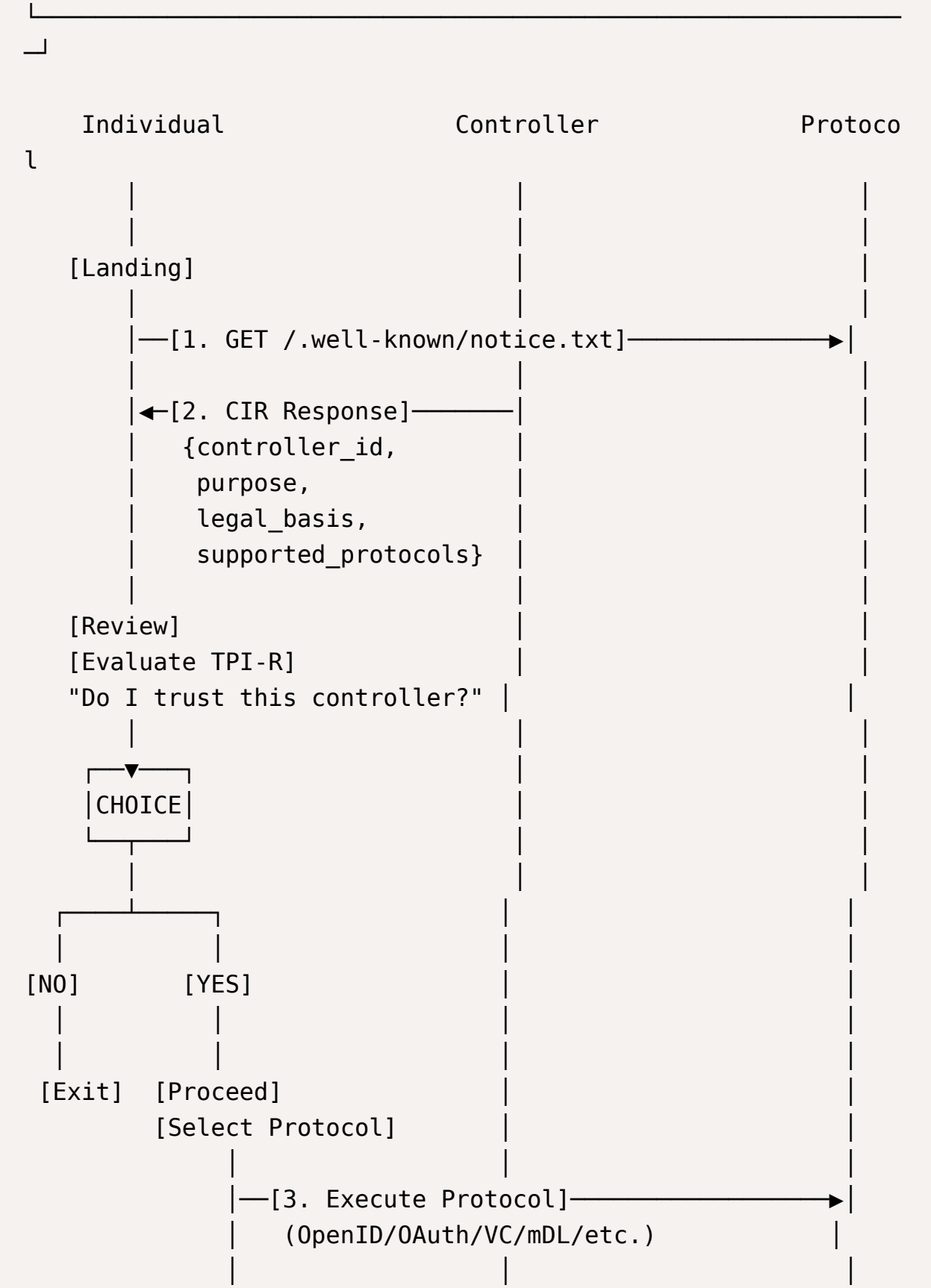
```
{
  "controller_id": "example.com",
  "legal_name": "Example Corp Ltd.",
  "jurisdiction": ["US", "EU"],
  "purpose": ["Authentication", "Service Delivery"],
  "legal_basis": ["Consent", "Contract"],
  "privacy_access_point": "https://example.com/privacy",
  "data_protection_officer": "dpo@example.com",
  "supported_protocols": [
    "openid-connect",
    "oauth2",
    "w3c-vc",
    "iso18013-5-mdl"
  ]
}
```

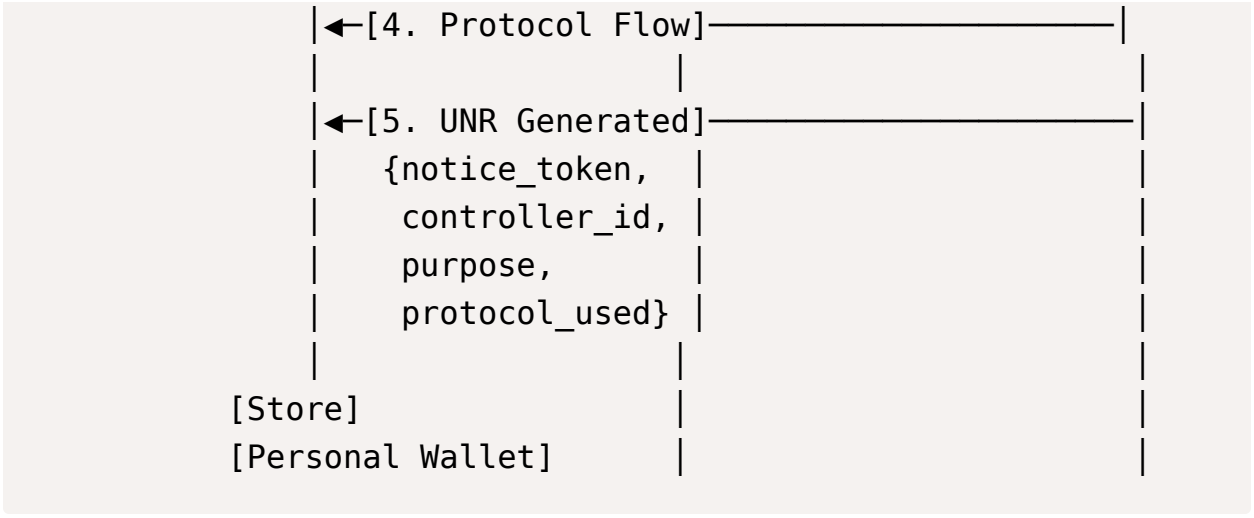
Universal Property: Discoverable via standard HTTP GET—no protocol-specific discovery required.

Integration Pattern: Pre-Protocol Disclosure

Any identity protocol can integrate:

```
┌──────────────────────────────────────────────────────────────────┐
┐
│                    UNIVERSAL PATTERN
│
```


Individual
Controller
Protocol
[Landing]
[1. GET /.well-known/notice.txt]
[2. CIR Response]
{controller_id,
purpose,
legal_basis,
supported_protocols}
[Review]
[Evaluate TPI-R]
"Do I trust this controller?"
CHOICE
[NO]
[YES]
[Exit]
[Proceed]
[Select Protocol]
[3. Execute Protocol]
(OpenID/OAuth/VC/mDL/etc.)



Result: Individual evaluates controller transparency **before** any identity protocol executes.

---

2. Cross-Protocol Notice Receipt Generation

Universal Notice Receipt (UNR) Structure

Protocol-agnostic bilateral proof:

```
{
  "schema_version": "27560-UNIVERSAL-NOTICE-2025-1.0",
  "receipt_id": "rcpt_xyz789",
  "controller_id": "example.com",
  "pii_principal_id": "did:example:123",
  "issued_at": "2026-02-09T18:35:00Z",
  "purpose": ["Authentication", "Service Delivery"],
  "legal_basis": ["Consent"],
  "identity_protocol_used": "openid-connect",
  "protocol_transaction_id": "oidc_txn_abc456",
  "credential_types_presented": ["email", "name"],
  "notice_token": "nt_secure_xyz789",
  "bilateral_proof": {
    "individual_receipt_hash": "sha256_individual",
    "controller_shadow_receipt_hash": "sha256_controller",
    "synchronized_timestamp": "2026-02-09T18:35:00Z"
```

```
  }
}
```

Key Properties:

- **Protocol-agnostic:** Works with any identity protocol via `identity_protocol_used` field
- **Bilateral:** Both individual and controller receive synchronized receipts
- **Portable:** `notice_token` works across protocols and controllers
- **Auditable:** `bilateral_proof` enables third-party verification

Protocol-Specific Extensions

OpenID Connect UNR:

```
{
  "identity_protocol_used": "openid-connect",
  "protocol_metadata": {
    "id_token_hash": "sha256_id_token",
    "nonce": "oidc_nonce_xyz",
    "acr": "urn:mace:incommon:iap:silver",
    "amr": ["pwd", "mfa"]
  }
}
```

W3C Verifiable Presentation UNR:

```
{
  "identity_protocol_used": "w3c-vp",
  "protocol_metadata": {
    "presentation_hash": "sha256_vp",
    "credential_types": ["VerifiableCredential", "EmailCreden
tial"],
    "presentation_submission": {
      "id": "submission_123",
      "definition_id": "def_456"
```

```
    }
  }
}
```

ISO/IEC 18013-5 mDL UNR:

```
{
  "identity_protocol_used": "iso18013-5-mdl",
  "protocol_metadata": {
    "device_engagement": "mdl_engagement_xyz",
    "doc_type": "org.iso.18013.5.1.mDL",
    "data_elements_disclosed": ["family_name", "given_name",
"birth_date"],
    "mdl_version": "1.0"
  }
}
```

Universal Property: UNR structure extends naturally to any protocol while maintaining core transparency fields.

---

3. Cross-Border Interoperability via Convention 108+

Treaty-Based Mutual Recognition

Convention 108+ provides legal framework for cross-border transparency recognition:

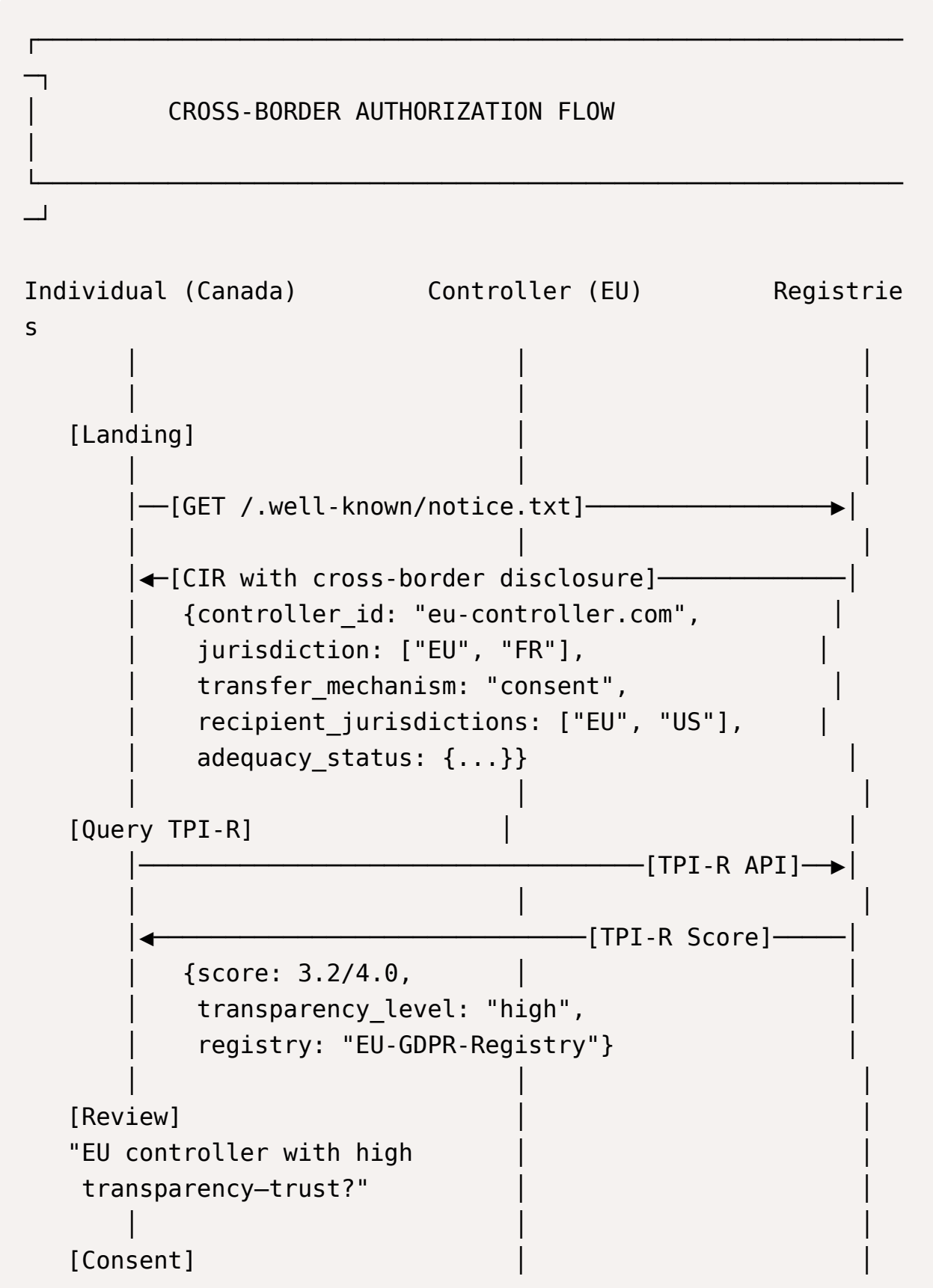
Article 8.2 (Transparency): Controller identification, purpose, legal basis disclosed before collection

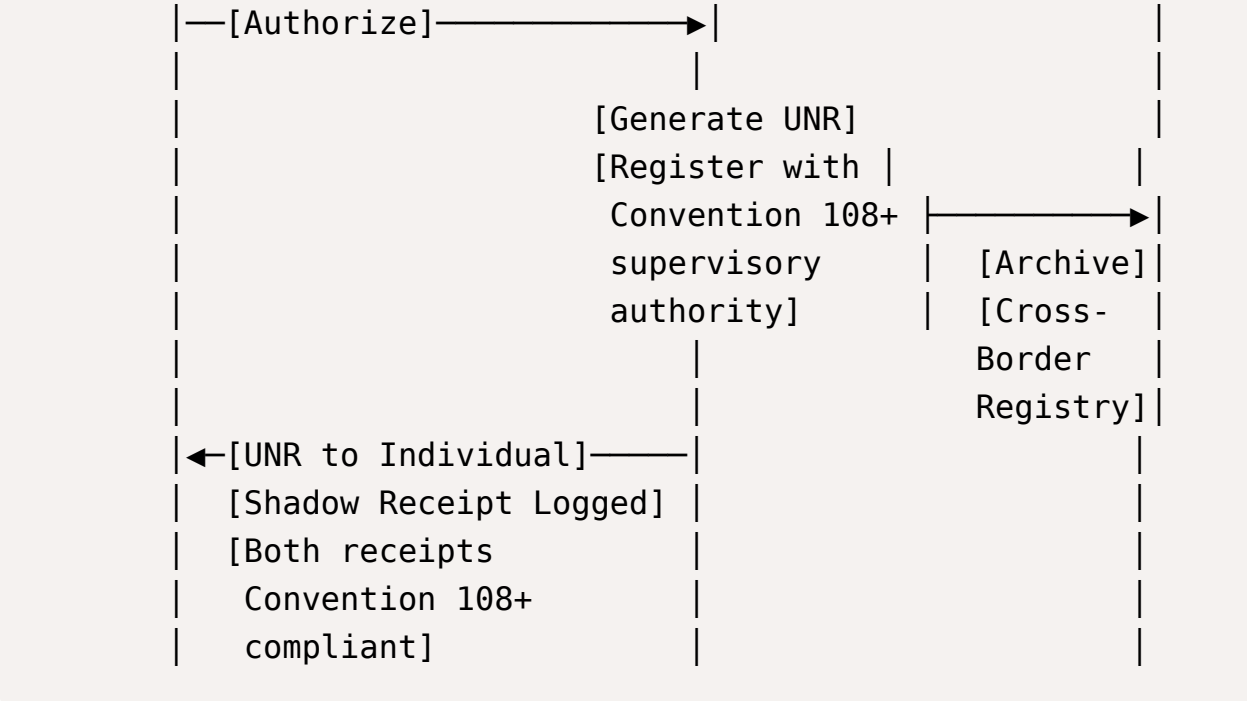
Article 14 (Cross-Border Flows): Data flows permitted when transparency maintained

55+ Signatory Jurisdictions: Including Commonwealth countries, EU member states, Latin American countries

Cross-Border UNR Pattern

Individual in Canada authorizes EU controller:

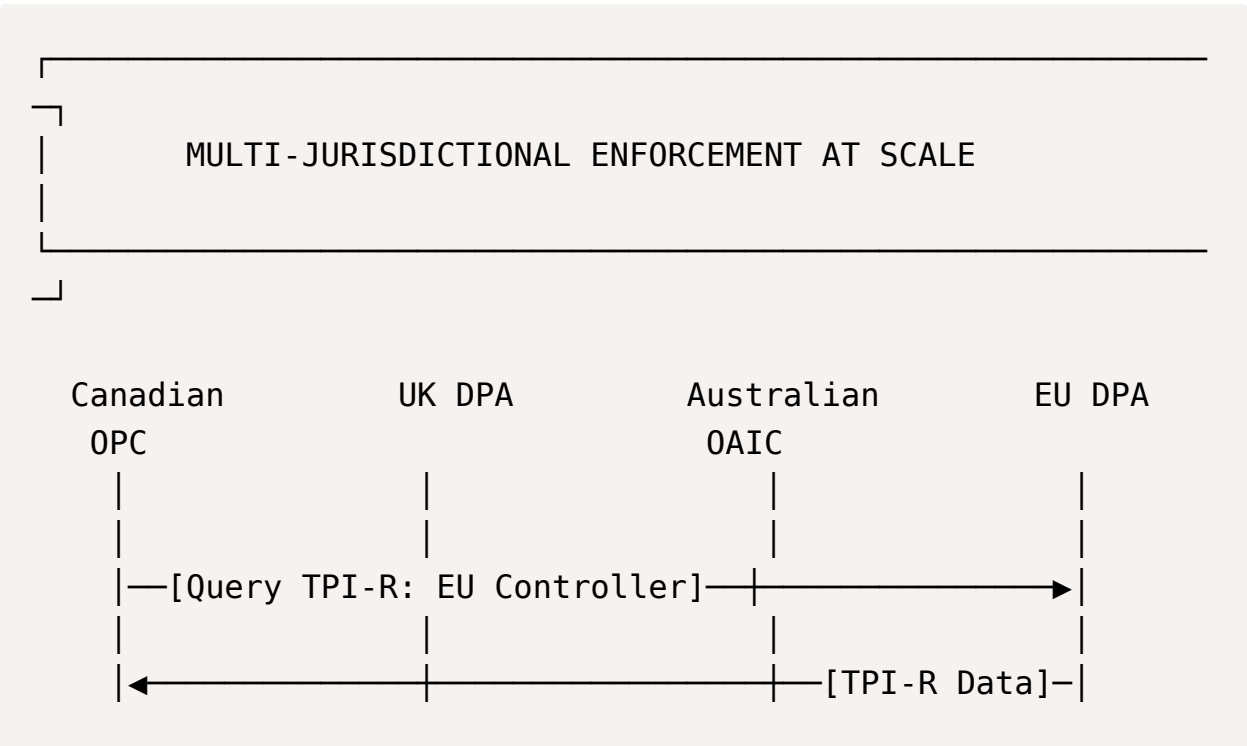

CROSS-BORDER AUTHORIZATION FLOW
Individual (Canada)
Controller (EU)
Registries
[Landing]
[GET /.well-known/notice.txt]
[CIR with cross-border disclosure]
{controller_id: "eu-controller.com",
jurisdiction: ["EU", "FR"],
transfer_mechanism: "consent",
recipient_jurisdictions: ["EU", "US"],
adequacy_status: {...}}
[Query TPI-R]
[TPI-R API]
[TPI-R Score]
{score: 3.2/4.0,
transparency_level: "high",
registry: "EU-GDPR-Registry"}
[Review]
"EU controller with high transparency—trust?"
[Consent]

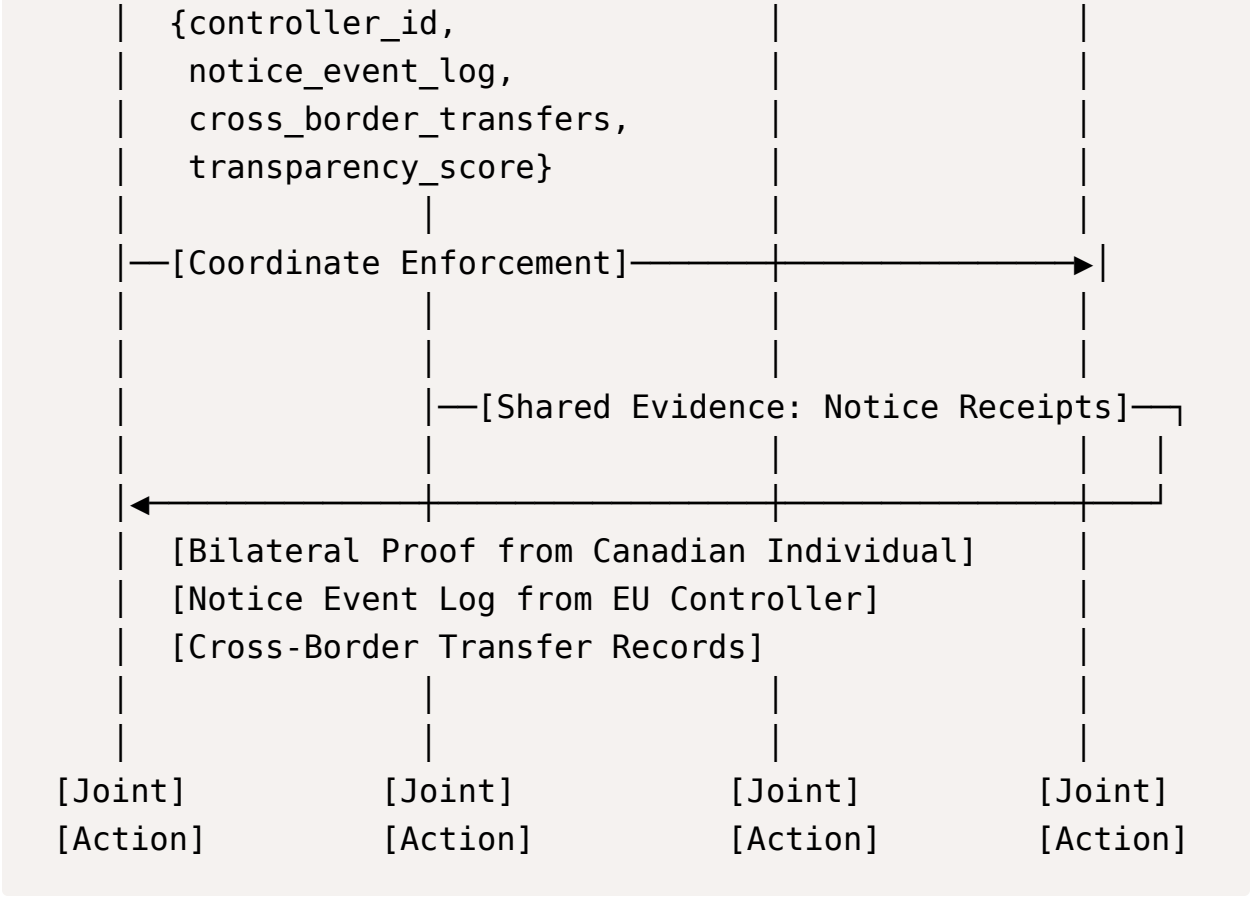


Universal Property: Convention 108+ provides treaty-based legal framework enabling Notice Receipt mutual recognition across 55+ jurisdictions.

Multi-Jurisdictional Enforcement

TPI-R Queryability Across Borders:





Universal Property: TPI-R infrastructure enables regulators across jurisdictions to coordinate enforcement using shared transparency evidence.

---

4. Portable Authorization via Notice Tokens

Cross-Protocol Portability

Notice Token as Universal Authorization Credential:

Problem: Current identity protocols create protocol-specific tokens:

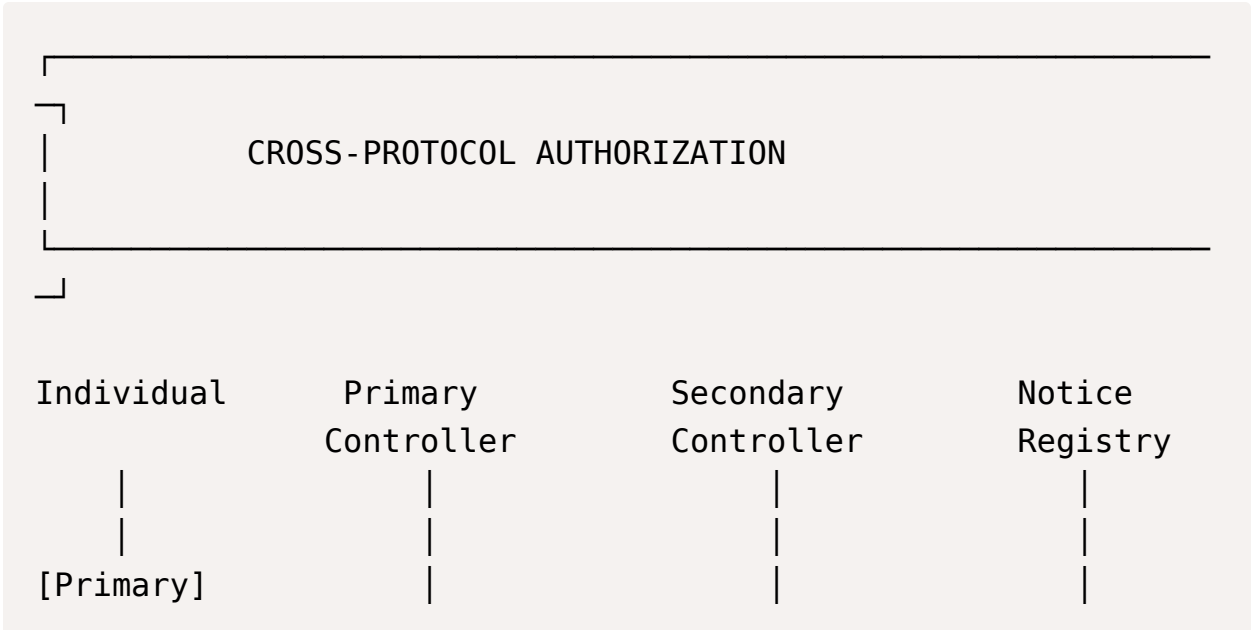
- OAuth: access_token, refresh_token (not portable across OAuth providers)
- OpenID: id_token (not portable, RP-specific)
- W3C VC: Verifiable Presentations (not authorization tokens)
- mDL: Session-specific, not portable

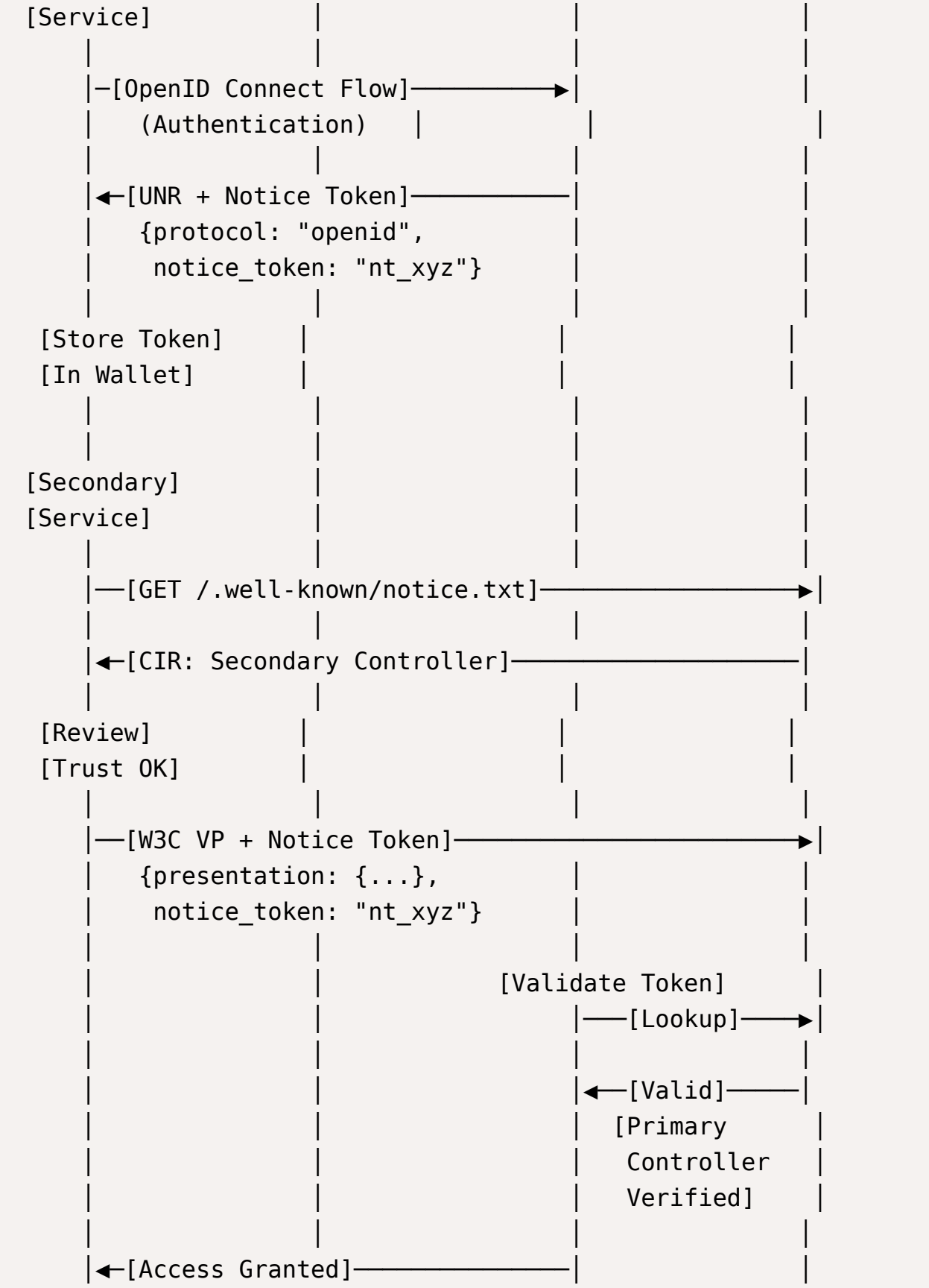
Solution: Notice Token provides protocol-agnostic portable authorization:

```
{
  "notice_token": "nt_universal_xyz789",
  "issued_by": "original-controller.com",
  "bound_to_principal": "did:example:123",
  "authorized_purposes": ["Service Delivery", "Analytics"],
  "legal_basis": ["Consent"],
  "valid_from": "2026-02-09T18:35:00Z",
  "expires_at": "2027-02-09T18:35:00Z",
  "portable_to": ["partner-controller.com"],
  "revocation_endpoint": "https://original-controller.com/.we
ll-known/revocation",
  "cryptographic_binding": {
    "algorithm": "ES256",
    "public_key": "...",
    "signature": "..."
  }
}
```

Cross-Protocol Authorization Flow

Individual authorizes primary controller via OpenID Connect, then uses Notice Token with secondary controller via W3C VC:




[Service]
[OpenID Connect Flow]
(Authentication)
[UNR + Notice Token]
{protocol: "openid",
notice_token: "nt_xyz"}
[Store Token]
[In Wallet]
[Secondary]
[Service]
[GET /.well-known/notice.txt]
[CIR: Secondary Controller]
[Review]
[Trust OK]
[W3C VP + Notice Token]
{presentation: {...},
notice_token: "nt_xyz"}
[Validate Token]
[Lookup]
[Valid]
[Primary
Controller
Verified]
[Access Granted]

```
    |   [New UNR Generated]           |                |
    |   {protocol: "w3c-vp",          |                |
    |    linked_to: "nt_xyz"}         |                |
```

Universal Property: Notice Token enables authorization to flow across protocols without re-authenticating or re-presenting credentials.

Integration with Major Identity Frameworks

OpenID Connect Integration

Pre-Authentication Controller Disclosure

Standard OpenID Flow:

```
1. Individual → RP: Click "Sign in with OpenID"
2. RP → OP: Redirect to authorization endpoint
3. Individual → OP: Authenticate
4. OP → RP: Return id_token
```

0PN Interop Enhanced Flow:

```
0. Individual → RP: Land on site
1. RP → Individual: Serve CIR at /.well-known/notice.txt
2. Individual: Review RP transparency via TPI-R
3. Individual: Decide to proceed (or exit)
4. Individual → RP: Click "Sign in with OpenID"
5. RP → OP: Redirect to authorization endpoint
6. Individual → OP: Authenticate
7. OP → Individual: Generate UNR for OP authentication
8. OP → RP: Return id_token
9. RP → Individual: Generate UNR for RP service delivery
10. Individual: Store both UNRs (OP + RP) with Notice Tokens
```

Benefits:

- Individual evaluates **both** OP and RP transparency before authentication
- Bilateral proof of notice for both OP and RP relationships
- Notice Tokens enable portable authorization across RPs
- TPI-R enables regulatory oversight of OpenID ecosystem

OpenID Provider as Transparent Controller

OpenID Provider publishes CIR:

```
{
  "controller_id": "openid-provider.com",
  "legal_name": "OpenID Provider Inc.",
  "role": "Identity Provider",
  "purpose": ["Authentication", "Identity Verification"],
  "data_collected": ["email", "name", "authentication_logs"],
  "retention": "3 years",
  "third_party_sharing": ["relying-parties-list.json"],
  "supported_protocols": ["openid-connect", "oauth2"],
  "tpi_r_endpoint": "https://openid-provider.com/.well-known/
tpi-r"
}
```

Relying Party publishes CIR:

```
{
  "controller_id": "relying-party.com",
  "legal_name": "Relying Party Services Ltd.",
  "role": "Service Provider",
  "purpose": ["Service Delivery", "Customer Support"],
  "authentication_providers": ["openid-provider.com"],
  "data_collected": ["email", "name", "usage_logs"],
  "retention": "2 years",
  "supported_protocols": ["openid-connect"],
  "tpi_r_endpoint": "https://relying-party.com/.well-known/tp
```

```
i-r"
}
```

Result: Individual can evaluate transparency of **entire authentication chain** before proceeding.

W3C Verifiable Credentials Integration

Verifier Transparency Disclosure

Problem with Current W3C VC Flow:

```
1. Verifier requests presentation
2. Holder presents credential
3. Verifier processes credential
```

Issue: Holder presents credential **before** knowing verifier identity or purpose.

0PN Interop Solution:

```
1. Verifier publishes CIR at /.well-known/notice.txt
2. Holder reviews verifier transparency via TPI-R
3. Holder decides whether to present credential
4. If yes: Verifier requests presentation
5. Holder presents credential
6. UNR generated documenting presentation
7. Notice Token enables future presentations to same verifier
```

Verifier CIR Example

```
{
  "controller_id": "verifier.example.com",
  "legal_name": "Example Verifier Service",
  "role": "Verifier",
  "purpose": ["Age Verification", "Identity Verification"],
  "legal_basis": ["Legitimate Interest", "Legal Obligation"],
```

```
  "presentation_definition": {
    "id": "age-verification-over-18",
    "input_descriptors": [{
      "id": "age_credential",
      "constraints": {
        "fields": [{
          "path": ["$.credentialSubject.birthdate"],
          "filter": {"type": "date", "maximum": "2008-02-09"}
        }]
      }
    }]
  },
  "data_retention": "Verification result only, no credential
storage",
  "third_party_sharing": "None",
  "supported_protocols": ["openid4vp", "w3c-vp"],
  "tpi_r_endpoint": "https://verifier.example.com/.well-know
n/tpi-r"
}
```

Benefits:

- Holder evaluates verifier purpose **before** presenting credential
- Selective disclosure decision informed by verifier transparency
- UNR provides bilateral proof of presentation
- Notice Token enables zero-knowledge subsequent presentations

---

ISO/IEC 18013-5 mDL Integration

Reader Transparency Disclosure

Current mDL Flow (ISO 18013-5):

```
1. Reader requests data elements
2. Holder device displays request
```

```
3. Holder approves/denies
4. mDL transmitted
```

Problem: Reader identity often opaque; holder doesn't know who is requesting or why.

0PN Interop Enhanced mDL Flow:

```
1. Reader publishes CIR at /.well-known/notice.txt
2. Reader QR code includes CIR URL or embedded CIR
3. Holder device fetches CIR automatically
4. Holder device displays:
   - Reader identity (from CIR)
   - Purpose (from CIR)
   - Legal basis (from CIR)
   - TPI-R score (if available)
   - Requested data elements (from mDL request)
5. Holder reviews and decides
6. If approved: mDL data elements transmitted
7. UNR generated documenting transaction
8. Notice Token stored for subsequent transactions with same
reader
```

Reader CIR Example (Airport Security)

```
{
  "controller_id": "airport-security.example.gov",
  "legal_name": "Example Airport Security Authority",
  "role": "Government Verifier",
  "purpose": ["Aviation Security", "Identity Verification"],
  "legal_basis": ["Legal Obligation"],
  "authority": "Transportation Security Act 2023",
  "data_elements_requested": [
    "family_name",
    "given_name",
    "birth_date",
```

```
    "document_number",
    "portrait"
  ],
  "data_retention": "24 hours (security logs only)",
  "third_party_sharing": "Law enforcement (warrant only)",
  "supported_protocols": ["iso18013-5-mdl"],
  "device_engagement_method": "QR code",
  "tpi_r_endpoint": "https://airport-security.example.gov/.we
ll-known/tpi-r"
}
```

Benefits:

- Holder knows **who** is requesting mDL data and **why**
- Legal basis disclosed (vital for government verifiers)
- TPI-R enables oversight of reader transparency practices
- Notice Token enables rapid subsequent presentations (e.g., return flight)

DIDComm Integration

DID Document + CIR Alignment

DID Document (Identity Layer):

```
{
  "@context": "https://www.w3.org/ns/did/v1",
  "id": "did:example:123",
  "verificationMethod": [...],
  "authentication": [...],
  "service": [
    {
      "id": "did:example:123#transparency",
      "type": "ControllerTransparency",
      "serviceEndpoint": "https://example.com/.well-known/not
ice.txt"
```

```
    }
  ]
}
```

CIR (Controller Transparency Layer):

```
{
  "controller_id": "did:example:123",
  "controller_type": "Individual",
  "purpose": ["Peer-to-Peer Communication"],
  "legal_basis": ["Legitimate Interest"],
  "supported_protocols": ["didcomm-v2"],
  "message_retention": "End-to-end encrypted, no server stora
ge"
}
```

Integration: DID service endpoint references CIR, enabling DIDComm agents to query controller transparency before establishing connection.

Universal Interoperability Properties

Property 1: Protocol-Agnostic Discovery

Every controller, regardless of identity protocol, publishes CIR at standard location:

`https://<controller-domain>/.well-known/notice.txt`

Discovery is universal:

- No protocol-specific metadata endpoints
- No proprietary discovery mechanisms
- Standard HTTP GET—works everywhere

Property 2: Legal Basis Agnostic

0PN Interop works across all 6 Convention 108+ Article 5 legal bases:

1. **Consent:** Notice Token provides portable consent credential
2. **Contract:** CIR discloses contractual purpose and terms
3. **Legal Obligation:** CIR references legal authority
4. **Vital Interest:** CIR discloses emergency processing purpose
5. **Public Interest:** CIR references public task authority
6. **Legitimate Interest:** CIR discloses legitimate interest and balancing test

Result: Universal interoperability across regulatory contexts.

Property 3: Jurisdiction-Agnostic

Convention 108+ provides treaty-based framework across 55+ jurisdictions:

- Commonwealth countries (UK, Canada, Australia, etc.)
- EU member states (via GDPR alignment)
- Latin American countries
- African countries

CIR supports multi-jurisdictional disclosure:

```
{
  "jurisdiction": ["US", "EU", "CA"],
  "applicable_laws": [
    {"jurisdiction": "US", "law": "CCPA"},
    {"jurisdiction": "EU", "law": "GDPR"},
    {"jurisdiction": "CA", "law": "PIPEDA"}
  ]
}
```

Property 4: Cryptographically Verifiable

Level 3 ANCR provides cryptographic verification:

- Blind notarization (zero-knowledge proof)
- Merkle tree anchoring

- Non-repudiation via digital signatures
- Long-term proof validity

Works across all identity protocols:

- OpenID id_token + ANCR
- W3C VC presentation + ANCR
- mDL transaction + ANCR
- OAuth token + ANCR

Property 5: Portable Authorization

Notice Token provides cross-protocol portability:

Single authorization flows across multiple protocols:

```
Individual authorizes Controller A via OpenID Connect
  ↓
Receives Notice Token
  ↓
Uses Notice Token with Controller A via OAuth API
  ↓
Uses Notice Token to authorize Controller A → Controller B da
ta sharing
  ↓
Controller B validates Notice Token via TPI-R
  ↓
Authorization honored without re-authentication
```

Universal Digital Identity Interop Use Cases

Use Case 1: Cross-Protocol Healthcare Journey

Scenario: Individual uses multiple identity protocols across healthcare journey

Journey:

1. **Hospital Website (OpenID Connect):**
    - Hospital publishes CIR at `hospital.example.com/.well-known/notice.txt`
    - Individual authenticates via OpenID Connect
    - UNR generated with Notice Token
2. **Pharmacy App (mDL):**
    - Pharmacy publishes CIR with mDL support
    - Individual presents mDL with Notice Token from hospital
    - Pharmacy validates Notice Token, confirms hospital authorization
    - New UNR generated linking hospital authorization
3. **Lab Results Portal (W3C VC):**
    - Lab publishes CIR with W3C VC support
    - Individual presents VerifiableCredential with Notice Token
    - Lab validates authorization chain (hospital → pharmacy → lab)
    - Bilateral proof across all three controllers

Result: Single authorization flow across three protocols (OpenID, mDL, W3C VC) with maintained transparency.

Use Case 2: Cross-Border Employment

Scenario: Canadian individual employed by EU company, uses US cloud services

Actors:

- Individual: Canada
- Employer: EU (France)
- Payroll Service: US
- Cloud Storage: US

Flow:

1. **Employer (EU) Hires Individual (CA):**

- Employer publishes CIR with cross-border disclosure
- Individual reviews TPI-R (EU GDPR registry)
- Individual consents via OpenID Connect
- UNR generated with Convention 108+ Article 14 cross-border fields

2. **Employer Uses Payroll Service (US):**
   - Payroll service publishes CIR
   - Employer shares Notice Token with payroll service
   - Individual receives notice of third-party sharing
   - Individual reviews payroll service TPI-R
   - New UNR generated documenting data transfer CA→EU→US
3. **Individual Uses Cloud Storage (US):**
   - Cloud provider publishes CIR
   - Individual authenticates via W3C VC
   - Individual grants employment-related file storage authorization
   - Notice Token links employment authorization to storage

Result: Cross-border authorization chain (CA→EU→US) with regulatory visibility across three jurisdictions.

Use Case 3: Federated Identity with Transparency

Scenario: University (IdP) provides identity for library, learning management, email services

Traditional Federation Problem:

- Student authenticates via university IdP
- IdP releases attributes to service providers
- Student unaware of which services IdP shares data with
- No bilateral proof of authorization

0PN Interop Solution:

1. **University IdP publishes CIR:**

```
{
  "controller_id": "university.edu",
  "role": "Identity Provider",
  "federated_services": [
    "library.university.edu",
    "lms.university.edu",
    "email.university.edu"
  ],
  "attribute_release_policy_url": "https://university.edu/fed
eration-policy"
}
```

1. **Each Service Provider publishes CIR:**

```
{
  "controller_id": "library.university.edu",
  "role": "Service Provider",
  "identity_provider": "university.edu",
  "purpose": ["Library Access", "Resource Borrowing"],
  "attributes_received": ["student_id", "name", "email"]
}
```

1. **Student Authentication Flow:**
    - Student reviews university IdP CIR
    - Student reviews library service CIR
    - Student authenticates via SAML/OpenID
    - UNR generated documenting IdP→SP attribute release
    - Notice Token enables future library access without re-authentication
    - Notice Event Logs at both IdP and SP track authorization

Result: Transparent federation where student can query TPI-R for entire federation before authorizing.

Why 0PN Interop is "Universal"

1. Works with All Identity Protocols

Pre-protocol layer:

- Controller disclosure happens **before** any identity protocol executes
- Protocol-agnostic CIR structure
- UNR extends to any protocol via `identity_protocol_used` field

No protocol lock-in:

- Individual can use OpenID with Controller A
- Then use W3C VC with Controller B
- Then use mDL with Controller C
- All three controllers share CIR format and UNR generation

2. Works Across All Jurisdictions

Treaty-based foundation:

- Convention 108+ provides legal framework
- 55+ signatory jurisdictions
- Mutual recognition of transparency requirements

Multi-jurisdictional enforcement:

- TPI-R queryable across borders
- Notice Event Logs auditable by any supervisory authority
- Bilateral proof recognized internationally

3. Works Across All Legal Bases

Not consent-only:

- Consent: Notice Token provides portable consent credential
- Contract: CIR discloses contractual terms
- Legal Obligation: CIR references legal authority
- Legitimate Interest: CIR discloses balancing test
- Vital Interest: CIR discloses emergency purpose
- Public Interest: CIR references public task

4. Works at All Assurance Levels

Four-level progressive trust:

- **Level 1:** Self-assertion (low-risk, local services)
- **Level 2:** Registry verification (standard services)
- **Level 3:** Cryptographic signatures (high-value, cross-border)
- **Level 4:** Real-time active state (critical infrastructure)

Universal: Any identity protocol can adopt any TATA level.

5. Works for All Actors

Individuals:

- Query TPI-R before sharing identity/credentials
- Store Notice Receipts as bilateral proof
- Use Notice Tokens for portable authorization

Controllers:

- Publish CIR once, works with all identity protocols
- Generate UNRs regardless of protocol used
- Maintain Notice Event Logs for audit

Regulators:

- Query TPI-R across jurisdictions
- Access Notice Event Logs for investigation
- Validate bilateral proof in disputes

Identity Protocol Developers:

- Integrate CIR discovery at protocol initiation
- Generate UNRs at protocol completion
- Reference Notice Tokens in authorization flows

Technical Implementation: Universal Integration Pattern

Step 1: Pre-Protocol Discovery

Every identity protocol integration starts the same way:

```
// Universal pre-protocol discovery
async function discoverController(controllerDomain) {
  const cirUrl = `https://${controllerDomain}/.well-known/not
ice.txt`;
  const response = await fetch(cirUrl);
  const cir = await response.json();

  return cir;
}

// Query TPI-R (if available)
async function queryTransparency(controllerDomain) {
  const tpiUrl = `https://${controllerDomain}/.well-known/tpi
-r`;
  const response = await fetch(tpiUrl);
  const tpiScore = await response.json();
```

```
  return tpiScore;
}

// Universal trust decision
async function evaluateController(controllerDomain) {
  const cir = await discoverController(controllerDomain);
  const tpiScore = await queryTransparency(controllerDomain);

  // Display to individual
  showControllerInfo(cir, tpiScore);

  // Individual decides
  const userConsent = await promptUserDecision();

  return userConsent;
}
```

Step 2: Protocol-Specific Execution

After trust decision, execute chosen protocol:

```
// Universal pattern
async function executeIdentityProtocol(protocol, controllerDo
main) {
  // Step 1: Pre-protocol transparency
  const trustDecision = await evaluateController(controllerDo
main);

  if (!trustDecision) {
    return { status: 'declined', reason: 'trust_evaluation_fa
iled' };
  }

  // Step 2: Execute protocol
  let protocolResult;
```

```
  switch(protocol) {
    case 'openid-connect':
      protocolResult = await executeOpenIDConnect(controllerD
omain);
      break;
    case 'w3c-vp':
      protocolResult = await executeW3CVP(controllerDomain);
      break;
    case 'iso18013-5-mdl':
      protocolResult = await executeMDL(controllerDomain);
      break;
    case 'oauth2':
      protocolResult = await executeOAuth2(controllerDomain);
      break;
    default:
      throw new Error(`Unsupported protocol: ${protocol}`);
  }

  // Step 3: Generate UNR
  const unr = await generateUNR(controllerDomain, protocol, p
rotocolResult);

  // Step 4: Store Notice Token
  await storeNoticeToken(unr.notice_token);

  return { status: 'success', unr, protocolResult };
}
```

Step 3: Universal Notice Receipt Generation

Generate UNR regardless of protocol:

```
async function generateUNR(controllerDomain, protocol, protoc
olResult) {
```

```
  const unr = {
    schema_version: "27560-UNIVERSAL-NOTICE-2025-1.0",
    receipt_id: generateReceiptId(),
    controller_id: controllerDomain,
    pii_principal_id: getPrincipalId(),
    issued_at: new Date().toISOString(),
    purpose: protocolResult.purpose,
    legal_basis: protocolResult.legal_basis,
    identity_protocol_used: protocol,
    protocol_transaction_id: protocolResult.transaction_id,
    notice_token: generateNoticeToken(),
    bilateral_proof: {
      individual_receipt_hash: hashReceipt(unr),
      controller_shadow_receipt_hash: await getControllerRece
iptHash(controllerDomain, unr),
      synchronized_timestamp: new Date().toISOString()
    }
  };

  // Store locally
  await storeUNR(unr);

  // Submit to Notice Event Log
  await submitToNoticeEventLog(controllerDomain, unr);

  return unr;
}
```

Step 4: Cross-Protocol Portability

Use Notice Token across protocols:

```
async function authorizeWithNoticeToken(newControllerDomain,
noticeToken) {
  // Fetch new controller CIR
```

```
  const cir = await discoverController(newControllerDomain);

  // Validate notice token portability
  if (!cir.accepts_portable_tokens) {
    throw new Error('Controller does not accept portable Noti
ce Tokens');
  }

  // Submit notice token for validation
  const validation = await fetch(`https://${newControllerDoma
in}/.well-known/validate-token`, {
    method: 'POST',
    body: JSON.stringify({ notice_token: noticeToken })
  });

  if (validation.ok) {
    // Generate new UNR linking to original authorization
    const linkedUNR = await generateLinkedUNR(newControllerDo
main, noticeToken);
    return linkedUNR;
  } else {
    throw new Error('Notice Token validation failed');
  }
}
```

Strategic Benefits: Why Universal Interop Matters

For Individuals

Single transparency interface across all identity protocols:

- Same CIR format whether using OpenID, W3C VC, mDL, or OAuth
- Same TPI-R query mechanism across all controllers

- Same Notice Receipt format across all protocols
- Same Notice Token for portable authorization

Result: Consistent user experience regardless of underlying identity technology.

For Controllers

Implement once, works everywhere:

- Publish single CIR at `/.well-known/notice.txt`
- Generate UNRs using standard format
- Maintain single Notice Event Log
- Support multiple identity protocols without protocol-specific transparency implementations

Result: Reduced implementation cost, universal compliance.

For Identity Protocol Developers

Transparency layer abstraction:

- Focus on protocol-specific features (authentication, authorization, credential format)
- Delegate transparency disclosure to 0PN Interop layer
- Standard integration pattern across all protocols

Result: Faster protocol development, built-in transparency.

For Regulators

Protocol-agnostic oversight:

- Query TPI-R regardless of identity protocol used
- Access Notice Event Logs with standard format
- Validate bilateral proof across protocols
- Coordinate enforcement across jurisdictions

Result: Enforcement at scale without protocol-specific expertise.

For Digital Identity Ecosystem

Interoperability without centralization:

- No single identity protocol wins
- No central identity provider required
- No federated trust anchor needed
- Controllers and individuals choose protocols that fit their needs

Result: Competitive, innovative ecosystem with universal transparency baseline.

Comparison: 0PN Interop vs Other Interoperability Approaches

vs. Federated Identity (SAML, OpenID Connect)

Federated Identity:

- Requires IdP as central trust anchor
- Protocol-specific federation metadata
- Trust relationships established via bilateral agreements
- Individual visibility into federation limited

0PN Interop:

- No central trust anchor required
- Protocol-agnostic CIR discovery
- Trust relationships evaluated per-transaction via TPI-R
- Individual has full visibility via Notice Receipts

Result: 0PN Interop works **with** federation but doesn't require it.

vs. Self-Sovereign Identity (DIDs, VCs)

SSI:

- Individual controls identity and credentials
- No central IdP required
- Focus on credential format and verification
- Verifier transparency often implicit

0PN Interop:

- Individual controls authorization via Notice Tokens
- No central transparency authority required
- Focus on controller transparency and bilateral proof
- Verifier transparency explicit via CIR

Result: 0PN Interop **complements** SSI by adding controller transparency layer.

vs. Trust Frameworks (eIDAS, NIST 800-63)

Trust Frameworks:

- Define identity assurance levels
- Specify authentication mechanisms
- Establish accreditation processes
- Focus on identity verification strength

0PN Interop:

- Defines transparency assurance levels (TATA)
- Works with any authentication mechanism
- Progressive trust without accreditation requirement
- Focus on controller transparency and authorization proof

Result: 0PN Interop **complements** trust frameworks by adding transparency dimension.

vs. Wallet-Based Identity (EUDI Wallet, mDL)

Wallet-Based:

- Individual stores credentials in wallet
- Selective disclosure to verifiers
- Focus on privacy-preserving credential presentation
- Verifier identity often opaque

0PN Interop:

- Individual stores Notice Receipts in wallet (alongside credentials)
- Controller transparency enables informed selective disclosure
- Focus on privacy-enabling controller disclosure
- Verifier identity explicit via CIR

Result: 0PN Interop **enhances** wallet-based identity with controller transparency.

Critical Success Factor: Network Effects

Adoption Tipping Points

Phase 1: Early Adopters (0-5% of controllers)

- High-transparency organizations publish CIRs
- Privacy-conscious individuals query TPI-R
- Competitive differentiation for transparent controllers

Phase 2: Regulatory Push (5-15% of controllers)

- Convention 108+ enforcement accelerates
- Regulators require TPI-R queryability
- Non-compliant controllers face penalties

Phase 3: Critical Mass (15-30% of controllers)

- Digital identity wallets integrate CIR discovery by default
- Identity protocols add standard CIR support
- Non-adoption becomes competitive disadvantage

Phase 4: Universal Standard (30%+ of controllers)

- CIR expected by individuals
- Identity protocols require CIR for certification
- Universal interoperability achieved

Network Effect Drivers

More controllers → Better tools:

- Wallet providers build CIR discovery interfaces
- TPI-R aggregators emerge
- Notice Token management tools mature

More protocols → Higher compatibility:

- OpenID Foundation adds CIR support
- W3C VC specs reference controller transparency
- ISO mDL updates include CIR requirements

More regulators → Stronger enforcement:

- Convention 108+ coordination improves
- Cross-border investigation efficiency increases
- TPI-R becomes standard regulatory tool

More individuals → Greater demand:

- Privacy-conscious market segment grows
- Transparent controllers attract premium users
- Dark patterns lose effectiveness

Conclusion: Universal Digital Identity Interop

0PN Interop achieves universal digital identity interoperability by:

1. **Pre-Protocol Transparency Layer** — Controller disclosure happens before any identity protocol executes
2. **Protocol-Agnostic Standards** — CIR, UNR, Notice Token work with all identity protocols
3. **Legal Basis Agnostic** — Works across all 6 Convention 108+ Article 5 legal bases
4. **Jurisdiction-Agnostic** — Treaty-based framework enables cross-border interoperability
5. **Portable Authorization** — Notice Tokens flow across protocols and controllers
6. **Regulatory Capacity** — TPI-R enables enforcement at scale across jurisdictions

Result: Universal interoperability without requiring a single identity protocol, central IdP, or federated trust anchor.

Strategic Position: 0PN Interop is **not** another identity protocol competing for adoption. It is the **transparency substrate** that enables trust decisions across **all** identity protocols.

Tagline: *"Universal transparency infrastructure for universal digital identity interoperability."*

Next Steps for Standards Bodies

OpenID Foundation

Recommendation: Add CIR discovery to OpenID Connect and OpenID4VP specifications

Specific Actions:

- Define `.well-known/notice.txt` discovery in OpenID Discovery spec
- Add UNR generation to token endpoint response
- Include Notice Token in authorization flow
- Reference ISO/IEC 27560-1 for CIR structure

W3C Credentials Community Group

Recommendation: Reference controller transparency in Verifiable Credentials spec

Specific Actions:

- Add CIR URL to Verifier metadata
- Define UNR generation at presentation verification
- Include Notice Token in presentation submission
- Align with ISO/IEC 27560-1 transparency requirements

ISO/IEC JTC 1 SC 17 WG 10 (mDL)

Recommendation: Integrate controller transparency in ISO/IEC 18013-5 updates

Specific Actions:

- Add CIR to device engagement QR code
- Define reader transparency metadata
- Include UNR in mdoc transaction logs
- Reference ISO/IEC 27560-1 for bilateral proof

IETF

Recommendation: Standardize `.well-known/notice.txt` URI

Specific Actions:

- Submit Internet-Draft for controller transparency well-known URI
- Define CIR JSON schema
- Specify TPI-R query API
- Reference Convention 108+ transparency requirements

Kantara Initiative

Recommendation: Update UMA 2.0 with controller transparency

Specific Actions:

- Add CIR disclosure to Resource Server registration
- Include UNR generation at authorization grant
- Define Notice Token as portable authorization credential
- Align with ISO/IEC 27560-1 ANCR protocol

License & Attribution

Lizar, M. (2026). *Analysis: 0PN Interop as Universal Digital Identity Interoperability Layer*. 0PN Lab. Retrieved from [URL]

Licensing Questions: See 0PN Licensing FAQ

Document Status: Analysis v1.0

Date: February 9, 2026

Author: Mark Lizar, ISO/IEC 27560-1 Profile Editor

Contact: mark@transparencylab.ca

Related Resources:

- Introduction: 0PN Interop for Digital Identity and Personal Data Control
- ISO-IEC 27560 Standards Hub
- ISO/IEC 27560:2025-1 Universal Notice Receipt Profile v1.01