

ZEKNOP Data Interoperability Standard: Asset Portability and Privacy-by-Design Identity Framework

Defensive Publication v1.1

Authors: Vladislav Urbánek

On behalf of: Authors with intent to transfer to ZEKNOP Stichting upon its formation

Date: 2026-01-27

Previous Version: v1.0 - DOI [10.5281/zenodo.17666290](https://doi.org/10.5281/zenodo.17666290)

DOI: [To be assigned upon publication]

License: Creative Commons Attribution 4.0 International (CC BY 4.0)

Contact: info@zeknop.org

Executive Summary

Data about individuals are now distributed across hundreds of institutions and services. Users repeatedly provide the same information—identity documents, addresses, financial details—to each service independently, losing control over where their data resides and how it is used, creating significant privacy risks and systemic inefficiencies.

This fragmentation manifests in two fundamental problems:

- **Over-collection:** Services collect more data than necessary (e.g., platforms requesting passport copies when they only need age confirmation)
- **Repeated verification:** Users re-enter the same information at every institution, with no portability between services

A pressing example is age verification: regulations like [Australia's Online Safety Act](#) (16+) and [France's Social Media Ban](#) (15+) require platforms to verify users' ages, but current methods force collection of highly sensitive personal data—creating new privacy risks while failing to solve the underlying problem efficiently.

The Solution: ZEKNOP (Zero-Knowledge Interoperable Open Protocol) addresses these challenges. It is an open, privacy-by-design standard that enables:

- **Verification of user claims** (age, KYC status, PEP status) without accessing underlying personal data
- **Seamless data portability** of financial assets, liabilities, and insurance policies between institutions
- **Granular identity management** with 9 distinct identity blocks, each shareable independently

The protocol's power is immediately demonstrated by its ability to solve the age verification challenge: a social network can verify a user is "over 16" (Australia) or "over 15" (France) without ever seeing their date of birth—receiving only a cryptographically signed "yes" or "no" from a trusted institution.

How It Works: The protocol leverages existing, highly-regulated institutions that users already trust—such as banks or government identity providers—to act as verifiers and data custodians. Users authenticate with their existing accounts; institutions issue only the minimal attestations or data needed. No new identity databases are created, and personal data flows only to authorised parties with explicit consent.

Why It Matters: ZEKNOP provides a practical, immediately deployable solution:

- **For regulators:** Enables compliance with privacy-preserving verification (age verification, KYC) while respecting data protection principles.
- **For platforms:** Reduces liability from storing sensitive personal data; simplifies compliance.
- **For users:** Protects privacy; enables seamless data portability; uses existing trusted relationships.
- **For the ecosystem:** Open standard prevents vendor lock-in; transparent protocol enables independent audit.

This publication establishes the open standard in the public domain, enabling ecosystem-wide adoption while preventing third-party patents on these fundamental interoperability concepts.

Abstract (Technical Summary)

ZEKNOP Data Interoperability Standard is an open protocol for **standardised asset portability and privacy-preserving identity management**. The standard addresses critical inefficiencies in transferring assets, liabilities, policies, and identity data between institutions, advisors, and service providers, while embedding privacy-by-design principles to minimise data exposure.

Core Components:

- 1. Asset & Liability Portability Standard** - Standardised data models (JSON Schema) for assets (14+ types including real estate, securities, commodities, cryptocurrencies), liabilities (11+ types), and insurance policies (12+ types), enabling automated transfer via OAuth 2.0 consent flows.
- 2. Granular Identity Framework** - Privacy-by-design identity data structure with verification levels (pseudonymous → official) and minimal disclosure patterns, allowing services to request only necessary identity attributes.
- 3. Consent Management** - OAuth 2.0 authorisation with granular scopes enabling precise control over what data is shared with which consumer.

Key Innovations Disclosed:

- **9 granular identity blocks** linked by universal user identifiers: pseudonymous, contact, address, financial, KYC, documents, biometric, membership, and digital life.
- **Derived identity claims** (e.g., "isPEP: false", "isOver15: true", "isOver16: true") without exposing underlying documents or birth dates
- **Privacy-by-design patterns**: zero-knowledge flows (eshops receive email only, carriers receive shipping address only), banks receive KYC status without passport copies, social networks verify age without birth dates
- **External type registries**: Extensible hierarchical naming for assets, liabilities, and policies (e.g., `eu.banking.current_account`, `us.retirement.401k`, `au.superannuation.accumulation`) enabling jurisdictional variations and new financial instruments
- **Jurisdictional data models**: Core fields (mandatory) plus jurisdiction-specific extensions, enabling cross-border data portability with fallback types
- **Standardised financial data models** enabling cross-institutional interoperability (14+ asset types, 11+ liability types, 12+ policy types)
- **Data migration right**: Users can transfer complete datasets between custodians, old custodians invalidate migrated data to reduce exposure risk
- **Audit trail** for consent and data access events with GDPR compliance

Scope: This publication establishes prior art for the **data model structures, API patterns, privacy principles, and data portability workflows** that form the open interoperability layer. Specific identity verification mechanisms, cryptographic implementations, and system architectures are **intentionally excluded** and subject to separate technical protocols (see Section 11: Scope of Defensive Publication).

1. Introduction

1.1 Problem Statement

Modern digital services suffer from critical interoperability and privacy failures, creating friction for users and significant risks for businesses. This is acutely evident in two key areas: financial data management and identity verification.

The Identity Verification Dilemma:

Digital services require trustworthy user verification, but current methods force users to over-share sensitive data. A pressing example of this tension is **age verification**. Governments worldwide are mandating that online platforms verify users' ages to protect minors from harmful content. Australia's **Online Safety Act** (16+) and France's **Social Media Ban** (15+) require designated platforms to take reasonable steps to verify users are of appropriate age. Similar regulations exist or are emerging in the UK, EU, and other jurisdictions.

Current age verification methods create a problematic trade-off:

METHOD	PRIVACY RISK	PRACTICAL ISSUES
ID document upload	Creates centralised identity databases vulnerable to breach	Users reluctant to share government IDs with platforms
Facial recognition	Biometric data collection; surveillance concerns	Accuracy issues; excludes users without webcams
Credit card verification	Financial data exposure	Excludes unbanked users; age-inappropriate for minors' debit cards
Third-party KYC providers	Data shared with additional parties; new data silos	Cost; user friction; trust issues

The result: Platforms face an impossible choice between collecting sensitive personal data (creating privacy risks and compliance burdens) or accepting self-declaration (failing to comply with regulations and protect minors).

Financial Data Silos:

Beyond identity verification, significant friction exists in financial data management:

- Clients switching financial advisors must manually re-enter complete asset inventories (1-2 hours).
- AML/KYC questionnaires are repeated at every institution (30+ minutes each).
- Manual data transfer via PDF/Excel exports is slow and error-prone.
- Institutions face a heavy compliance burden, unable to easily verify data provenance or freshness.

Identity Data Over-Disclosure:

- E-commerce platforms collect full residential addresses when only a shipping address is needed for delivery.
- Banks receive copies of passport/ID documents when only the verification status ("KYC complete") is required.
- Users have no granular control over which specific identity attributes are shared for a given transaction.

Quantified Impact:

- Financial advisor onboarding: **5-10 hours** of manual work per client.
- Data breach risk: Over-collection of sensitive identity documents dramatically increases the surface area for attack and potential liability.
- Compliance overhead: **€50-200** per client per institution in manual processing and secure data storage.

1.2 Vision

ZEKNOP Data Interoperability Standard provides:

- 1. Standardised Data Models** - JSON Schema definitions for financial assets, liabilities, policies, and identity attributes, enabling machine-readable interoperability.
- 2. Granular Consent Mechanism** - OAuth 2.0 authorisation allowing users to grant precise scopes (e.g., `assets:read`, `identity:kyc:status` but NOT `identity:documents:full`).
- 3. Privacy-by-Design Patterns** - Derived claims, minimal disclosure, and purpose-specific data sharing embedded in protocol design.
- 4. Audit Trail** - Every data access logged with timestamp, consumer, scope, and record count for accountability.
- 5. Open Governance** - Standard managed by independent foundation (Stichting), ensuring no vendor lock-in.

Design Philosophy: Separate **data model standardisation** (public domain, this document) from **verification/authentication mechanisms** (separate protocols, implementation-agnostic).

1.3 Use Cases

Flagship Use Case: Privacy-Preserving Age Verification As a primary example of the protocol in action, consider a social media platform that needs to comply with regulations like the Australian Online Safety Act (16+) or the French Social Media Ban (15+). Instead of collecting passports or dates of birth, the platform requests the specific scope `identity:age:over16` or `identity:age:over15`. The user authenticates with their bank or government identity provider (e.g., myGovID in Australia,

FranceConnect in France), which cryptographically attests `{ "isOver16": true }` or `{ "isOver15": true }` without ever sharing the user's date of birth or other personal data with the platform. See [Section 8](#) for detailed implementation.

Financial Advisory: Client switches advisors → OAuth consent for `assets:read`, `liabilities:read`, `policies:read`, `identity:kyc:status` → Advisor receives complete portfolio data in 5 minutes instead of weeks.

Insurance Brokerage: Broker requests `policies:read`, `identity:basic` → Accesses existing insurance contracts and contact info → Provides recommendations without re-entering data.

E-Commerce Checkout: Eshop requests `identity:contact:email`, `identity:userId` → Receives email for order confirmation + user identifier → Shipping address shared DIRECTLY with carrier via separate OAuth flow → Eshop never has customer's address.

Banking KYC: Bank requests `identity:kyc:status` → Receives `{isPEP: false, amlRiskLevel: "low", verifiedAt: "2025-11-18"}` → Completes onboarding without requesting passport copies or proof-of-address documents.

2. Data Model Standard

2.1 Financial Data Types

ZEKNOP standardises financial data into three categories, each with multiple subtypes:

2.1.1 Assets (14+ Types)

JSON Schema: `schemas/asset.json`

Asset Type Registry:

Asset types are defined in an **external, extensible registry** to accommodate:

- Jurisdictional differences (EU vs. US vs. Asia vs. Australia banking products)
- New financial instruments (DeFi, tokenised assets)
- Regional specifics (e.g., Germany's "Bausparvertrag", UK's "ISA", US "401k", Australia's "Superannuation")

Registry Structure: Hierarchical naming using reverse domain notation:

```
eu.banking.current_account      # European current account
us.retirement.401k            # US 401(k) retirement account
de.savings.bausparvertrag      # German building savings contract
au.superannuation.accumulation  # Australian superannuation (accumulation phase)
au.superannuation.pension      # Australian superannuation (pension phase)
global.crypto.bitcoin          # Cryptocurrency (jurisdiction-agnostic)
```

Governance: ZEKNOP Technical Committee maintains the registry. New asset types are added via RFC process. Backward compatibility guaranteed (deprecated types never removed).

```
{
  "assetId": "uuid",
  "assetType": "string", // Reference to external registry
  "assetTypeRegistry": "https://zeknop.org/registries/asset-types/v1",
  "jurisdiction": "ISO 3166-1 alpha-2 (e.g., DE, US, EU, AU)",
  "institutionName": "string",
  "accountNumber": "string (encrypted or masked)",
  "balance": "number",
  "currency": "ISO 4217",
  "acquisitionDate": "ISO 8601 date",
  "lastUpdated": "ISO 8601 datetime",
  "valuation": {
    "currentValue": "number",
    "valuationDate": "ISO 8601 date",
    "valuationMethod": "market | book | appraisal"
  }
}
```

Example - Bank Account:

```
{
  "assetId": "550e8400-e29b-41d4-a716-446655440000",
  "assetType": "eu.banking.current_account",
  "assetTypeRegistry": "https://zeknop.org/registries/asset-types/v1",
  "jurisdiction": "EU",
  "institutionName": "Example Bank",
  "accountNumber": "*****1234",
  "balance": 50000,
  "currency": "EUR",
  "acquisitionDate": "2020-01-15",
  "lastUpdated": "2025-11-19T10:00:00Z"
}
```

Core Asset Categories (v1 Registry):

- `eu.banking.*` - European banking products
- `eu.investment.*` - Investment products (mutual funds, ETFs, stocks)
- `eu.real_estate.*` - Real estate holdings
- `us.retirement.*` - US retirement accounts (401k, IRA, Roth IRA)
- `au.superannuation.*` - Australian superannuation funds
- `au.banking.*` - Australian banking products
- `global.crypto.*` - Cryptocurrency assets (Bitcoin, Ethereum, etc.)
- `[jurisdiction].[category].[specific]` - Extensible hierarchy

2.1.2 Liabilities (11+ Types)

JSON Schema: `schemas/liability.json`

Liability Type Registry: Follows the same external registry pattern as assets, using hierarchical naming:

```
eu.mortgage.residential      # European residential mortgage
us.loan.student_federal     # US federal student loan
eu.credit.consumer          # European consumer credit
au.mortgage.residential     # Australian residential mortgage
au.loan.hecs                 # Australian HECS/HELP student loan
```

```
{
  "liabilityId": "uuid",
  "liabilityType": "string", // Reference to external registry
  "liabilityTypeRegistry": "https://zeknop.org/registries/liability-types/v1",
  "jurisdiction": "ISO 3166-1 alpha-2",
  "institutionName": "string",
  "accountNumber": "string",
  "principalAmount": "number",
  "outstandingBalance": "number",
  "interestRate": "number",
  "maturityDate": "ISO 8601 date",
  "monthlyPayment": "number",
  "currency": "ISO 4217"
}
```

Example - Mortgage:

```
{
  "liabilityId": "660e8400-e29b-41d4-a716-446655440001",
  "liabilityType": "eu.mortgage.residential",
  "liabilityTypeRegistry": "https://zeknop.org/registries/liability-types/v1",
  "jurisdiction": "EU",
  "institutionName": "Example Bank",
  "accountNumber": "*****5678",
  "principalAmount": 300000,
  "outstandingBalance": 250000,
  "interestRate": 3.5,
  "maturityDate": "2045-12-31",
  "monthlyPayment": 1500,
  "currency": "EUR"
}
```

2.1.3 Insurance Policies (12+ Types)

JSON Schema: `schemas/policy.json`

Policy Type Registry: Follows the same external registry pattern as assets and liabilities:

```
eu.insurance.life           # European life insurance
us.insurance.health_private # US private health insurance
eu.insurance.property       # European property insurance
au.insurance.life           # Australian life insurance
au.insurance.income_protection # Australian income protection insurance
```

```
{
  "policyId": "uuid",
  "policyType": "string", // Reference to external registry
  "policyTypeRegistry": "https://zeknop.org/registries/policy-types/v1",
  "jurisdiction": "ISO 3166-1 alpha-2",
  "insurerName": "string",
  "policyNumber": "string",
  "coverageAmount": "number",
  "premiumAmount": "number",
  "premiumFrequency": "monthly | quarterly | annual",
  "effectiveDate": "ISO 8601 date",
  "expiryDate": "ISO 8601 date",
  "beneficiaries": [...]
}
```

Example - Life Insurance:

```
{
  "policyId": "770e8400-e29b-41d4-a716-446655440002",
  "policyType": "eu.insurance.life",
  "policyTypeRegistry": "https://zeknop.org/registries/policy-types/v1",
  "jurisdiction": "EU",
  "insurerName": "Example Insurance Co.",
  "policyNumber": "POL-123456",
  "coverageAmount": 500000,
  "premiumAmount": 1200,
  "premiumFrequency": "annual",
  "effectiveDate": "2020-01-01",
  "expiryDate": "2050-01-01",
  "beneficiaries": ["John Doe", "Jane Doe"]
}
```

Reference: Complete JSON Schema definitions available in [specs/data-portability/schemas/](https://zeknop.org/specs/data-portability/schemas/)

2.2 Identity Data Framework

ZEKNOP defines identity data as **granular blocks** linked by a universal user identifier, enabling privacy-by-design minimal disclosure.

2.2.1 Universal User Identifier

Every user has a persistent **userId** (UUID v4) that links identity blocks across systems:

```
{
  "userId": "550e8400-e29b-41d4-a716-446655440000"
}
```

Properties:

- Stable across all services (portable identifier)
- Pseudonymous (no PII in the identifier itself)
- Used to link blocks: pseudonymous, contact, address, financial, KYC
- OAuth **sub** claim in access tokens

Important: This standard defines the **userId** as a linking mechanism. How users obtain, verify, or upgrade their **userId** verification level is defined by separate identity verification protocols outside the scope of this defensive publication.

2.2.2 Identity Blocks

JSON Schema: `schemas/identity-blocks.json`

Pseudonymous Block

Minimal public identity, no real-world linkage:

```
{
  "userId": "uuid",
  "blockType": "pseudonymous",
  "nickname": "string",
  "publicKey": "string (optional, for cryptographic operations)",
  "createdAt": "ISO 8601 datetime"
}
```

Contact Block

Communication channels without revealing full identity:

```
{
  "userId": "uuid",
  "blockType": "contact",
  "email": "string",
  "emailVerified": "boolean",
  "phoneNumber": "E.164 format",
  "phoneVerified": "boolean",
  "preferredLanguage": "ISO 639-1"
}
```

Address Block

Physical addresses with type differentiation:

```
{
  "userId": "uuid",
  "blockType": "address",
  "addressType": "residential | billing | shipping | business",
  "address": {
    "street": "string",
    "city": "string",
    "postalCode": "string",
    "country": "ISO 3166-1 alpha-2",
    "isPrimary": "boolean"
  },
  "verifiedAt": "ISO 8601 datetime (optional)"
}
```

Privacy Pattern: User may have multiple addresses. E-commerce does NOT receive addresses (shipping address goes directly to carrier via separate OAuth flow). Only delivery carriers receive `addressType: "shipping"` via scope `identity:address:shipping`. Residential address remains private.

Financial Block

Banking and payment information:

```
{
  "userId": "uuid",
  "blockType": "financial",
  "bankAccount": {
    "iban": "string (encrypted or tokenised)",
    "bic": "string"
  },
  "taxId": "string (encrypted)",
  "taxResidency": "ISO 3166-1 alpha-2"
}
```

KYC Block (Derived Claims)

Key Innovation: Provides verification status and derived claims WITHOUT exposing underlying documents.

```
{
  "userId": "uuid",
  "blockType": "kyc",
  "verificationLevel": "pseudonymous | basic | extended | official | kyc_verified",

  // Derived claims (no underlying documents)
  "isPEP": "boolean",
  "amlRiskLevel": "low | medium | high",
  "sanctionsCheckPassed": "boolean",
  "kycCompletedAt": "ISO 8601 datetime",
  "kycExpiresAt": "ISO 8601 datetime",

  // Optional: Full documents (separate high-trust consent)
  "documents": {
    "scope": "identity:kyc:documents", // Requires separate consent
    "nationalIdReference": "encrypted reference",
    "proofOfAddressReference": "encrypted reference",
    "pepDeclarationReference": "encrypted reference"
  }
}
```

Verification Levels:

LEVEL	DESCRIPTION	USE CASE
pseudonymous	Nickname only, no real-world identity	Forums, gaming, public comments
basic	Email + phone verified	Newsletter subscriptions, low-risk services
extended	Name, date of birth, address verified	E-commerce, mid-tier services
official	Government-issued ID verified	Financial services, legal contracts
kyc_verified	Full KYC/AML compliance	Banking, investment, high-value transactions

Privacy-by-Design: Banks request `identity:kyc:status` scope and receive `{isPEP: false, amlRiskLevel: "low"}` — sufficient for compliance WITHOUT storing passport copies.

Documents Block

Official identity documents with derived claims pattern:

```
{
  "userId": "uuid",
  "blockType": "documents",
  "documents": [
    {
      "documentType": "passport | nationalId | drivingLicense | birthCertificate |
                      marriageCertificate | residencePermit | workPermit",
      "documentNumber": "string (encrypted)",
      "issuingCountry": "ISO 3166-1 alpha-2",
      "issuingAuthority": "string",
      "issuedAt": "ISO 8601 date",
      "expiresAt": "ISO 8601 date",
      "verificationLevel": "self_declared | ocr_verified | official_verified",

      // Reference to encrypted document (not the document itself!)
      "documentReference": "encrypted_storage_reference",

      // Derived claims (public, no document exposure)
      "isValid": "boolean",
      "isExpired": "boolean",
      "daysUntilExpiry": "number"
    }
  ]
}
```

Privacy Pattern - Age Verification:

```
// Social network requests scope: identity:age:over18
// Receives: { "isOver18": true, "verifiedAt": "2025-11-18" }
// Date of birth NOT disclosed, passport NOT disclosed
// Sufficient for age-gated content compliance
```

Privacy Pattern - Document Validity:

```
// Car rental requests: identity:documents:drivingLicense:validity
// Receives: { "isValid": true, "expiresAt": "2028-05-15" }
// License number, address on license NOT disclosed
```

OAuth Scopes:

- `identity:documents:passport:validity` (just validity status)
- `identity:documents:drivingLicense:full` (full document - high trust only)
- `identity:age:over18` , `identity:age:over21` , `identity:age:over16` (derived claims)

Biometric Block

Biometric data and photographs from official identity documents:

CRITICAL PRIVACY NOTICE: Biometric data is highly sensitive under GDPR Article 9 (special category data) and Australian Privacy Act. This block requires explicit consent and is subject to strictest security measures.

```
{
  "userId": "uuid",
  "blockType": "biometric",

  "photograph": {
    "photoReference": "encrypted_storage_reference", // Reference, NOT the photo itself
    "sourceDocument": "passport | nationalId | drivingLicense",
    "capturedAt": "ISO 8601 datetime",
    "expiresAt": "ISO 8601 datetime", // Document expiration
    "photoHash": "SHA-256 hash (for integrity verification)"
  },

  "fingerprint": {
    "fingerprintReference": "encrypted_storage_reference", // Some national IDs include fingerprints
    "sourceDocument": "nationalId",
    "capturedAt": "ISO 8601 datetime"
  },

  // Derived claims (safe to share)
  "hasPhotograph": "boolean",
  "hasFingerprint": "boolean",
  "photographVerified": "boolean"
}
```

Privacy-by-Design Patterns:**Pattern 1: Photo Verification Status**

```
// Service requests: identity:biometric:photo:verified
// Receives: { "photographVerified": true, "verifiedAt": "2025-11-19" }
// Does NOT receive: actual photo, photo reference
```

Pattern 2: Photo Reference (High Trust Only)

```
// KYC provider requests: identity:biometric:photo:reference
// Receives: { "photoReference": "encrypted_ref", "sourceDocument": "passport" }
// Provider can retrieve photo via separate API call (logged in audit trail)
// Requires Certified or Secure+ certification
```

GDPR Art. 9 and Australian Privacy Act Compliance:

Biometric data processing requires:

- Explicit consent (not implied, checkbox required)
- Specific purpose (identity verification, NOT profiling)
- Encrypted storage (references encrypted, photos encrypted at rest)
- Right to deletion (user can delete biometric data anytime)
- Minimal disclosure (derived claims preferred over raw data)

OAuth Scopes:

```
identity:biometric:photo:verified      # Verification status only
identity:biometric:photo:reference     # Reference (Certified+ only)
identity:biometric:fingerprint:verified
identity:biometric:fingerprint:reference
```

Security Requirements:

1. Encrypted References:

- Photo and fingerprint references MUST be encrypted at rest
- References point to separate secure storage (not inline data)
- Access logged in audit trail with timestamp and consumer

2. Expiration Policy:

- Biometric data inherits document expiration
- User notified 90 days before document expiration
- Expired biometric data automatically invalidated

3. Breach Response:

- If biometric data breached: immediate user notification
- User MUST re-verify identity with new document capture

Implementation Note:

This block stores **references to biometric data from official documents only** (passport photos, national ID fingerprints). Advanced authentication mechanisms are NOT part of this defensive publication and are subject to separate technical specifications.

Membership Block

Loyalty programs, subscriptions, tickets, vouchers:

```
{
  "userId": "uuid",
  "blockType": "membership",
  "memberships": [
    {
      "membershipType": "loyaltyCard | subscription | ticket | coupon | voucher |
                        membershipCard | pass",
      "provider": "string", // "Tesco Clubcard", "Netflix", "Prague Public Transport"
      "membershipId": "string",
      "tier": "string (optional)", // "Gold", "Premium", "VIP"
      "pointsBalance": "number (optional)",
      "validFrom": "ISO 8601 date",
      "validUntil": "ISO 8601 date (optional)",
      "status": "active | expired | suspended | pending",

      // For digital tickets/passes
      "barcode": "string (optional)", // QR code, barcode for scanning
      "barcodeFormat": "qr | ean13 | code128",

      "benefits": ["string"], // ["10% discount", "Free shipping", "Priority boarding"]
      "metadata": {
        "autoRenew": "boolean",
        "nextBillingDate": "ISO 8601 date (optional)"
      }
    }
  ]
}
```

Use Cases:

- Loyalty Card Consolidation: All loyalty programs in one place, share with partner (e.g., aggregator app)
- Public Transport Tickets: Ticket inspector scans QR code, validates via API (no personal data exposed)
- Subscription Management: User grants access to subscription tracker, revokes anytime
- Event Tickets: Transfer ticket to friend via consent (revoke original, grant to recipient)

OAuth Scopes:

```
identity:membership:loyaltyCards
identity:membership:subscriptions
identity:membership:tickets
```

Digital Life Block

Social media profiles, gaming accounts, avatars, online reputation:

```
{
  "userId": "uuid",
  "blockType": "digitalLife",
  "profiles": [
    {
      "profileType": "socialMedia | gaming | content | avatar | forum | metaverse",
      "platform": "string", // "Bluesky", "Mastodon", "Steam", "Twitch", "Discord"
      "platformUserId": "string",
      "handle": "string", // "@username"
      "displayName": "string",
      "profileUrl": "url",
      "avatarUrl": "url (optional)",
      "verifiedAccount": "boolean",
      "accountCreatedAt": "ISO 8601 date",

      // Reputation and social proof
      "followerCount": "number (optional)",
      "reputationScore": "number (optional)", // Platform-specific
      "achievementsBadges": ["string"],
      "level": "number (optional)", // Gaming level, forum rank

      // Privacy controls
      "visibility": "public | friends | private",
      "ageVerified": "boolean"
    }
  ]
}
```

Use Cases:**Age Verification for Social Networks:**

```
// Social network requests: identity:age:over18
// Receives: { "isOver18": true, "verifiedAt": "2025-11-18" }
// User can access age-restricted content
// Birth date, government ID NOT disclosed to social network
```

Gaming Profile Portability:

```
// New gaming platform requests: identity:digitalLife:gaming
// Receives: Steam profile with achievements, level, reputation
// Platform can offer personalised onboarding, cross-platform rewards
```

Social Proof for Services:

```
// Marketplace requests: identity:digitalLife:social:reputation
// Receives: { "twitterVerified": true, "followerCount": 15000 }
// Increases trust for buyer/seller, reduces fraud
```

Metaverse Avatar Portability:

```
// Virtual world requests: identity:digitalLife:avatar
// Receives: Avatar appearance, customisation, purchased items
// User maintains consistent identity across virtual worlds
```

OAuth Scopes:

```
identity:digitalLife:social
identity:digitalLife:gaming
identity:digitalLife:avatar
```

2.2.3 Person Profile (Consolidated View)

JSON Schema: `schemas/person-profile.json`

Optional consolidated view linking multiple blocks (used when consumer has broad consent):

```
{
  "userId": "uuid",
  "verificationLevel": "official",

  "pseudonymousData": { "nickname": "..." },
  "contactData": { "email": "...", "phone": "..." },
  "addresses": [
    { "addressType": "residential", "address": {...} },
    { "addressType": "shipping", "address": {...} }
  ],
  "financialData": { "taxId": "...", "taxResidency": "CZ" },
  "kycData": {
    "isPEP": false,
    "amlRiskLevel": "low",
    "kycCompletedAt": "2025-11-18T10:00:00Z"
  },
  "documentsData": [
    { "documentType": "passport", "isValid": true, "expiresAt": "2030-01-01" }
  ],
  "memberships": [
    { "membershipType": "loyaltyCard", "provider": "Tesco", "pointsBalance": 1500 }
  ],
  "digitalLifeProfiles": [
    { "profileType": "gaming", "platform": "Steam", "handle": "..." }
  ]
}
```

Access Control: Consumer receives ONLY blocks for which they have consent (OAuth scopes).

Summary of Identity Blocks:

- 9 granular blocks enabling privacy-by-design minimal disclosure
- Each block accessible via separate OAuth scopes
- Derived claims prevent document over-collection
- Universal `userId` links blocks (implementation-agnostic)

2.3 Jurisdictional Data Models

Design Principle: Financial and regulatory systems vary significantly across jurisdictions. ZEKNOP accommodates this through jurisdiction-aware data models without fragmenting the core standard.

2.3.1 Jurisdiction Identifier

Every data custodian and consumer declares their primary jurisdiction:

```
{
  "custodianId": "example-bank-de",
  "jurisdiction": "DE", // ISO 3166-1 alpha-2
  "regulatoryFrameworks": ["EU-GDPR", "EU-PSD2", "DE-BaFin"],
  "supportedJurisdictions": ["DE", "AT", "CH"] // Multi-country support
}
```


2.3.2 Data Model Extensions by Jurisdiction

Core Model (Jurisdiction-Agnostic): All implementations MUST support core fields (balance, currency, accountNumber).

Jurisdiction-Specific Extensions: Implementations MAY add jurisdiction-specific fields via `extensions` object:

```
{
  "assetId": "uuid",
  "assetType": "de.savings.bausparvertrag",
  "jurisdiction": "DE",

  // Core fields (mandatory)
  "balance": 25000,
  "currency": "EUR",

  // Jurisdiction-specific fields (optional)
  "extensions": {
    "de": {
      "bausparvertrag": {
        "targetAmount": 50000,
        "savingsPhase": "active",
        "loanPhase": "pending",
        "wohnungsbauprämie": 512 // German housing subsidy
      }
    }
  }
}
```

2.3.3 Cross-Jurisdictional Data Transfer

Scenario: User migrates from German bank to Canadian bank.

Challenge: Canadian bank may not understand `de.savings.bausparvertrag`.

Solution: Fallback type for interoperability:

```
{
  "assetType": "de.savings.bausparvertrag",
  "fallbackType": "eu.savings.savings_account", // Generic equivalent
  "jurisdiction": "DE",
  "targetJurisdiction": "CA",

  // Core fields understood by all jurisdictions
  "balance": 25000,
  "currency": "EUR",

  // Germany-specific fields preserved but optional
  "extensions": { "de": {...} },

  // Migration notes for manual review
  "migrationNotes": "German Bausparkasse contract - no direct Canadian equivalent. Consider converting to"
}
```

2.3.4 Regulatory Compliance by Jurisdiction

JURISDICTION	REGULATORY FRAMEWORK	ZEKNOP COMPLIANCE
EU	GDPR, PSD2, eIDAS	Core design principle
DE	BaFin, German Banking Act	Supported via extensions
AU	Online Safety Act, Privacy Act (APPs), AML/CTF Act	Core design principle; age verification flagship use case
FR	GDPR, Social Media Ban (<15)	Supported (Zero-Knowledge Age Verification)
US	GLBA, FCRA, state laws	Planned (registry extensible)
CA	PIPEDA, FINTRAC	Planned (registry extensible)
UK	UK GDPR, FCA rules, Online Safety Act	Supported (post-Brexit extensions)

2.3.5 Governance of Jurisdictional Extensions

RFC Process for New Jurisdictions:

1. Jurisdiction authority or community member proposes extension
2. Technical Committee reviews regulatory requirements
3. Schema extensions drafted, validated against local regulations
4. Public comment period (30 days)
5. Adoption vote, addition to registry

This approach ensures ZEKNOP remains globally applicable while respecting regional regulatory differences.

3. Privacy-by-Design Principles

3.1 Minimal Disclosure Patterns

ZEKNOP embeds privacy-by-design at the protocol level, not as optional feature.

Pattern 1: Derived Claims Without Underlying Data

Problem: Bank needs to know if client is Politically Exposed Person (PEP) for AML compliance.

Traditional Approach: Client provides PEP declaration document → Bank stores copy → GDPR risk, over-collection.

ZEKNOP Approach:

```
// Bank requests scope: identity:kyc:status
// Bank receives:
{
  "userId": "uuid",
  "isPEP": false, // ← Boolean, no document
  "kycCompletedAt": "2025-11-18T10:00:00Z"
}
// PEP declaration document stays with KYC provider
// Bank compliant, no document storage
```

Additional Examples:

Age Verification (Adult Content):

```
// Service requests: identity:age:over18
// Receives: { "isOver18": true }
// Date of birth NOT disclosed
```

Age Verification (Australia - Online Safety Act):

```
// Social platform requests: identity:age:over16
// Receives: { "isOver16": true, "verifiedAt": "2026-01-27" }
// Date of birth NOT disclosed
// Platform complies with Online Safety Act without collecting PII
```

Credit Scoring:

```
// Lender requests: identity:credit:score
// Receives: { "creditScore": 750, "scoredAt": "2025-11-18" }
// Full credit report NOT disclosed
```

Pattern 2: Purpose-Specific Address Disclosure (Zero-Knowledge)

Problem: E-commerce platforms traditionally collect complete customer profiles (name, email, phone, billing address, shipping address) even though each party needs only specific data (eshop needs email for order confirmation, carrier needs shipping address for delivery).

Traditional Approach: Eshop form requests: name, email, phone, billing address, shipping address → All stored in eshop database → Single breach exposes complete customer profiles.

ZEKNOP Approach (Zero-Knowledge):

```

Step 1: Eshop Authorisation
Eshop requests scopes:
- identity:contact:email
- identity:userId

Eshop receives:
- Email (for order confirmation)
- User identifier (for carrier handoff)
- DOES NOT receive shipping address

Step 2: Carrier Authorisation (Separate OAuth Flow)
User authorises carrier directly:
- Carrier requests: identity:address:shipping
- User grants consent to carrier
- Carrier receives: shipping address

Step 3: Order Coordination
Eshop → Carrier API:
- Sends: orderId + userId (reference only)
- Carrier validates userId against OAuth token
- Carrier delivers to address from OAuth (not from eshop)

User's residential address:
- NEVER disclosed to eshop
- Shipping address disclosed ONLY to carrier
- Eshop acts as authorisation intermediary, NOT data holder

```

Result:

- Attack surface minimised: Eshop breach does NOT expose any addresses
- Zero-knowledge: Eshop has email, Carrier has address, neither has both
- Privacy maximum: Residential address never leaves data custodian

Pattern 3: Financial Advisor Granular Access

Problem: Advisor needs portfolio overview, not full bank access.

Traditional Approach: Client provides bank statements (PDF) → Advisor sees ALL transactions, balances, counterparties.

ZEKNOP Approach:

```

// Client grants scopes:
// - assets:read:summary
// - liabilities:read:summary
// - policies:read

// Advisor receives:
[
  { "assetType": "bankAccount", "balance": 50000, "currency": "EUR" },
  { "assetType": "mutualFund", "balance": 120000, "currency": "EUR" },
  { "liabilityType": "mortgage", "outstandingBalance": 200000 }
]

// Advisor does NOT receive:
// - Individual transactions
// - Counterparty names
// - Account numbers (only masked)

```

Advanced: If advisor needs transaction access, client grants additional scope `assets:transactions:read` with time-bound consent (expires after 90 days).

3.2 Granular OAuth Scopes

Scope Naming Convention: `resource:block:action`

Identity Scopes (Core):

```
identity:pseudonymous      # Nickname only
identity:contact           # Email + phone
identity:address:residential # Home address
identity:address:shipping  # Delivery address
identity:address:billing   # Payment address
identity:financial         # Bank account, tax ID
identity:kyc:status        # Derived claims (isPEP, riskLevel)
identity:kyc:documents     # Full documents (high trust only)
```

Identity Scopes (Age Verification - Derived Claims):

```
identity:age:over13        # Age verification (13+, COPPA compliance)
identity:age:over15        # Age verification (15+, French Social Media Ban)
identity:age:over16        # Age verification (16+, Australian Online Safety Act)
identity:age:over18        # Age verification (18+, adult content)
identity:age:over21        # Age verification (21+, US alcohol)
identity:age:over25        # Age verification (25+, car rental)
```

Identity Scopes (Documents - Derived Claims):

```
identity:documents:passport:validity # Validity status only
identity:documents:passport:full      # Full passport data (high trust)
identity:documents:drivingLicense:validity # License validity
identity:documents:drivingLicense:full  # Full license data
identity:documents:nationalId:validity  # ID validity
```

Identity Scopes (Biometric - Document References):

```
identity:biometric:photo:verified      # Photo verification status only
identity:biometric:photo:reference      # Photo reference (Certified+ only)
identity:biometric:fingerprint:verified # Fingerprint verification status
identity:biometric:fingerprint:reference # Fingerprint reference (Certified+ only)
```

Identity Scopes (Membership & Loyalty):

```
identity:membership:loyaltyCards # All loyalty programs
identity:membership:tickets      # Event tickets, transport passes
identity:membership:subscriptions # Streaming, software subscriptions
identity:membership:coupons      # Vouchers, discounts
```

Identity Scopes (Digital Life):

```
identity:digitalLife:social      # Social media profiles
identity:digitalLife:gaming     # Gaming accounts, achievements
identity:digitalLife:avatar     # Metaverse avatars, virtual items
identity:digitalLife:reputation # Social proof, follower counts
```

Financial Data Scopes:

```
assets:read      # All assets
assets:read:summary # Balances only, no transactions
assets:transactions:read # Full transaction history
liabilities:read # All liabilities
policies:read    # Insurance policies
policies:documents:read # Policy documents (PDFs)
```

Consent Workflow:

1. Consumer (financial advisor, eshop, bank) requests specific scopes
2. User presented with clear consent screen: "Advisor X requests access to: Assets (summary), Liabilities, Insurance Policies"
3. User approves → OAuth access token issued with granted scopes
4. Consumer API requests limited to granted scopes
5. Consent logged in audit trail

3.3 Time-Bound and Revocable Consent**Expiration:**

```
{
  "consentId": "uuid",
  "userId": "uuid",
  "consumerId": "advisor-company-abc",
  "scopes": ["assets:read", "liabilities:read"],
  "grantedAt": "2025-11-18T10:00:00Z",
  "expiresAt": "2026-11-18T10:00:00Z", // 1 year
  "status": "active"
}
```

Revocation:

- User can revoke consent at any time via data custodian UI
- Consumer's access tokens immediately invalidated
- Revocation logged in audit trail

3.4 Architectural Principle: Zero-Knowledge Data Flow

Core Rule: Only data custodians (entities storing and managing user data) may hold multiple identity blocks simultaneously. All other consumers (services, apps, businesses) must receive ONLY the minimal data required for their specific function.

Rationale:

- Prevents creation of new "honeypots" (attractive targets for attackers)
- Ensures breach of one consumer does NOT expose full identity
- Maintains granular consent enforcement at architectural level

Example - E-Commerce Flow:

```
User → Eshop (grants: identity:contact:email)
  |
  └─ Carrier (grants: identity:address:shipping, separate authorisation)
```

Result:

- Eshop knows: email + orderID
- Carrier knows: shipping address + orderID
- Neither party has BOTH email + address
- Link via orderID, but no data aggregation

Example - Financial Advisory:

```
User → Advisor (grants: assets:read:summary, identity:kyc:status)
```

Advisor receives:

- Asset balances (no transaction details)
- KYC status (isPEP: false, amlRiskLevel: low)

Advisor does NOT receive:

- Full name + address + assets simultaneously
- Government ID documents
- Bank account numbers (only masked references)

Exception: Data custodians are explicitly allowed to hold multiple blocks because they are subject to strictest security/compliance requirements (SOC 2, ISO 27001), implement data portability rights, and users explicitly choose them as trusted parties.

Full Specification: See RFC-0002: Zero-Knowledge Architecture Patterns (to be published)

3.5 Consent Templates and Jurisdictional Restrictions

Problem: Users should not need to understand OAuth scopes to grant appropriate consent. Misconfiguration leads to poor UX and security risks.

Solution: ZEKNOP defines consent templates - pre-approved scope bundles for common use cases, validated by jurisdiction.

Example Template:

```
{
  "templateId": "ecommerce.checkout.eu",
  "name": "E-Commerce Checkout (EU)",
  "jurisdiction": "EU",

  "allowedScopes": [
    "identity:contact:email",
    "identity:userId"
  ],

  "prohibitedScopes": [
    "identity:address:residential",
    "identity:address:shipping", // Must go to carrier directly
    "identity:documents:*",
    "identity:biometric:*",
    "identity:financial:*"
  ],

  "regulatoryBasis": ["GDPR Art. 5(1)(c) - data minimisation"]
}
```


Example Template - Australian Age Verification:

```
{
  "templateId": "age_verification.au",
  "name": "Age Verification (Australia)",
  "jurisdiction": "AU",

  "allowedScopes": [
    "identity:age:over16"
  ],

  "prohibitedScopes": [
    "identity:documents:*",
    "identity:biometric:*",
    "identity:address:*",
    "identity:contact:*"
  ],

  "regulatoryBasis": ["Online Safety Act - age verification", "Privacy Act (APPs) - data minimisation"]
}
```

Jurisdictional Restrictions:

Different jurisdictions impose different restrictions on data collection, enforced at protocol level:

CONSUMER TYPE	JURISDICTION	PROHIBITED SCOPES	RATIONALE
E-Commerce	EU, US, CA, AU	identity:documents:* , identity:biometric:*	No legitimate interest in government IDs for shopping
E-Commerce	All	identity:address:residential	Privacy: home address not needed for delivery
Social Media	EU, AU	identity:biometric:* (except age verification)	GDPR Art. 9 / Privacy Act: Biometric data only for specific purposes
Social Media	AU	All scopes except identity:age:over16	Online Safety Act: only age verification required, not identity
Financial Advisor	EU	identity:documents:passport:full	No legal basis for storing passport copies

Enforcement:

```
IF (consumer requests prohibited scope) THEN
  RETURN error: {
    "error": "invalid_scope",
    "error_description": "Scope 'identity:biometric:face:view' is
                          prohibited for consumer type 'ecommerce'
                          in jurisdiction 'EU' under GDPR Art. 9."
  }
}
```

User Consent Screen (Simplified):

```
"ExampleShop.com wants to complete your order.  
  
▫ Email address (for order confirmation)  
▫ Order reference (for delivery tracking)  
  
▫ Shipping address will be shared directly with the carrier  
  (not with ExampleShop.com)  
  
[Approve] [Customise] [Deny]"
```

Full Specification: See RFC-0003: Consent Template Registry (to be published)

4. API Specification

4.1 OAuth 2.0 Authorisation Flow

ZEKNOP uses **OAuth 2.0 with PKCE** (Proof Key for Code Exchange) for secure authorisation.

Roles:

- Resource Owner: User (data subject)
- Client: Consumer (financial advisor, bank, eshop)
- Authorisation Server: ZEKNOP-compliant identity provider
- Resource Server: Data custodian (current financial institution, KYC provider)

Flow:

1. Client → Authorisation Request
GET /oauth/authorize?
 response_type=code&
 client_id=advisor-company-abc&
 redirect_uri=https://advisor.com/callback&
 scope=assets:read liabilities:read identity:kyc:status&
 state=random_state&
 code_challenge=sha256(verifier)&
 code_challenge_method=S256
2. User → Consent Screen
"Advisor Company ABC requests:
- Read your assets (summary)
- Read your liabilities
- Access your KYC verification status

[Approve] [Deny]"
3. User approves → Authorisation Code
Redirect: https://advisor.com/callback?code=AUTH_CODE&state=random_state
4. Client → Token Request
POST /oauth/token
{
 "grant_type": "authorization_code",
 "code": "AUTH_CODE",
 "redirect_uri": "https://advisor.com/callback",
 "client_id": "advisor-company-abc",
 "code_verifier": "original_verifier"
}
5. Authorisation Server → Access Token
{
 "access_token": "eyJhbGciOiJSUzI1NiIs...",
 "token_type": "Bearer",
 "expires_in": 3600,
 "refresh_token": "tGzv3J0kF0XG5Qx2TlKWIA",
 "scope": "assets:read liabilities:read identity:kyc:status"
}
6. Client → API Request
GET /v1/assets
Authorization: Bearer eyJhbGciOiJSUzI1NiIs...
7. Resource Server → Data Response
[
 { "assetType": "bankAccount", "balance": 50000, "currency": "EUR" },
 { "assetType": "mutualFund", "balance": 120000, "currency": "EUR" }
]

4.2 REST API Endpoints

Base URL: `https://api.data-custodian.example/v1`

Authentication: Bearer token (OAuth 2.0 access token)

Financial Data Endpoints

```
GET /assets:
  summary: List all assets for which consent granted
  security: [oauth2: [assets:read]]
  responses:
    200:
      content:
        application/json:
          schema:
            type: array
            items: { $ref: '#/components/schemas/Asset' }

GET /assets/{assetId}:
  summary: Get specific asset detail
  parameters:
    - name: assetId
      in: path
      required: true
  responses:
    200: { schema: Asset }
    403: Forbidden (scope insufficient)

GET /liabilities:
  security: [oauth2: [liabilities:read]]

GET /policies:
  security: [oauth2: [policies:read]]
```

Identity Data Endpoints

```

GET /identity/profile:
  summary: Get consolidated identity profile
  security: [oauth2: [identity:contact, identity:address:*.]]
  responses:
    200:
      content:
        application/json:
          schema: { $ref: '#/components/schemas/PersonProfile' }
          # Returns ONLY blocks for which consumer has scope

GET /identity/blocks/contact:
  security: [oauth2: [identity:contact]]

GET /identity/blocks/address:
  summary: Get addresses (filtered by scope)
  security: [oauth2: [identity:address:residential OR identity:address:shipping]]
  parameters:
    - name: addressType
      in: query
      schema:
        enum: [residential, billing, shipping, business]
      # Consumer receives ONLY address types matching their scopes

GET /identity/blocks/kyc:
  security: [oauth2: [identity:kyc:status]]
  responses:
    200:
      content:
        application/json:
          example:
            {
              "userId": "uuid",
              "verificationLevel": "kyc_verified",
              "isPEP": false,
              "amlRiskLevel": "low",
              "kycCompletedAt": "2025-11-18T10:00:00Z"
            }

GET /identity/claims/age:
  summary: Get age verification claims
  security: [oauth2: [identity:age:over16 OR identity:age:over18 OR identity:age:over21]]
  responses:
    200:
      content:
        application/json:
          example:
            {
              "userId": "uuid",
              "isOver16": true,
              "isOver18": true,
              "isOver21": false,
              "verifiedAt": "2026-01-27T10:00:00Z"
            }
          # Returns ONLY age claims for which consumer has scope

```

Consent Management Endpoints

```
GET /consents:
  summary: List active consents (user view)
  security: [oauth2: [consents:read]]
  responses:
    200:
      content:
        application/json:
          schema:
            type: array
            items:
              type: object
              properties:
                consentId: string
                consumerId: string
                consumerName: string
                scopes: array
                grantedAt: datetime
                expiresAt: datetime
                status: enum [active, revoked, expired]

DELETE /consents/{consentId}:
  summary: Revoke consent
  security: [oauth2: [consents:manage]]
  responses:
    204: No Content (consent revoked)
```

Complete OpenAPI 3.0 Specification: <specs/data-portability/openapi.yaml>

4.3 Data Migration and Portability

User Right: Users can migrate their complete dataset from one data custodian to another at any time, exercising their fundamental data portability right.

Migration Workflow

1. User initiates migration to new custodian
↓
2. User authenticates with current custodian (source)
User authenticates with new custodian (target)
↓
3. Source custodian exports complete dataset:
 - All identity blocks (pseudonymous, contact, address, financial, KYC, documents, membership, digitalLi
 - All financial data (assets, liabilities, policies)
 - All audit trail (consent history, access logs)
 - All active consents (consumer authorisations)↓
4. Target custodian imports and validates dataset:
 - JSON Schema validation passes
 - Data integrity checks complete
 - User confirms import successful↓
5. Source custodian marks data as "migrated":
 - status: "migrated"
 - migratedTo: "target-custodian-id"
 - migratedAt: "2025-11-18T10:00:00Z"
 - All local data encrypted/archived (compliance retention)
 - Active data access DISABLED↓
6. Active consents transferred or re-authorised:
 - Option A: Consents automatically transferred (target custodian notifies consumers)
 - Option B: Consents revoked, consumers request new authorisation from target custodian↓
7. Source custodian issues migration certificate:
 - Cryptographic proof of data transfer
 - Audit trail entry
 - User receives confirmation

API Endpoint: Data Export

```
POST /v1/export:
summary: Initiate full data export for migration
security: [oauth2: [data:export]]
requestBody:
  content:
    application/json:
      schema:
        properties:
          targetCustodian:
            type: string
            description: Target custodian identifier
          exportFormat:
            type: string
            enum: [zeknop_json, encrypted_archive]
          includeAuditTrail:
            type: boolean
            default: true
responses:
  202:
    description: Export initiated
    content:
      application/json:
        schema:
          properties:
            exportId: string
            status: enum [pending, processing, completed, failed]
            estimatedCompletion: datetime
  200:
    description: Export completed
    content:
      application/json:
        schema:
          properties:
            identityBlocks: array
            financialData: array
            auditTrail: array
            activeConsents: array
            exportedAt: datetime
            dataHash: string # SHA-256 integrity hash
```

API Endpoint: Data Import

```

POST /v1/import:
  summary: Import dataset from previous custodian
  security: [oauth2: [data:import]]
  requestBody:
    content:
      application/json:
        schema:
          properties:
            exportedData: object # Complete export from source custodian
            sourceCustodian: string
            migrationCertificate: string # Cryptographic proof from source
  responses:
    201:
      description: Import successful
      content:
        application/json:
          schema:
            properties:
              importId: string
              recordsImported: number
              validationsPassed: boolean
              importedAt: datetime

```

Data Invalidation Policy

Source Custodian Responsibilities:

After successful migration confirmation:

1. Immediate Actions:

- Mark all user data: `status = "migrated"`
- Revoke all active OAuth access tokens (consumers lose access)
- Disable API endpoints for this user (returns 410 Gone)
- Log migration event in audit trail

2. Data Retention (Compliance):

- Retain encrypted archive per regulatory requirements (typically 5-10 years)
- Archive NOT accessible via API (only for regulatory audit)
- Periodic retention review (delete after retention period expires)

3. Consumer Notification:

- Notify all consumers with active consents: "User has migrated to [target custodian]"
- Provide target custodian contact info (if user consents)
- Consumers can request new authorisation from target custodian

Security Rationale:

- **Risk Reduction:** Minimises data exposure surface (fewer custodians = fewer breach targets)
- **User Control:** User decides which custodian they trust most
- **GDPR Compliance:** Right to data portability (Art. 20), right to erasure (Art. 17)
- **Privacy Act Compliance:** Australian Privacy Principle 12 (access to personal information)

Migration Frequency: Users can migrate unlimited times. Best practice: Custodians should NOT charge fees for migration (anti-competitive, violates data portability spirit).

4.4 Pagination and Filtering

Large Datasets:

```
GET /assets?page=1&pageSize=50&assetType=bankAccount
```

Incremental Sync:

```
GET /assets?updatedAtSince=2025-11-17T00:00:00Z
```

Response Format:

```
{
  "data": [...],
  "pagination": {
    "page": 1,
    "pageSize": 50,
    "totalPages": 5,
    "totalRecords": 243
  }
}
```

5. Audit Trail and Compliance

5.1 Audit Event Types

Every significant action is logged:

Consent Events:

- `consent.created` - User granted consent to consumer
- `consent.revoked` - User revoked consent
- `consent.expired` - Consent expired (automatic)

Data Access Events:

- `data.access` - Consumer accessed data via API
- `data.export` - User initiated full data export (GDPR right)

Identity Events:

- `identity.verification_upgraded` - User upgraded verification level (e.g., basic → official)

5.2 Audit Log Structure

```
{
  "eventId": "uuid",
  "timestamp": "2025-11-18T10:15:30.123Z",
  "eventType": "data.access",

  "userId": "uuid", // Subject
  "consumerId": "advisor-company-abc", // Accessor

  "details": {
    "endpoint": "/v1/assets",
    "method": "GET",
    "scopes": ["assets:read"],
    "recordsReturned": 12,
    "ipAddress": "198.51.100.42 (hashed or anonymised)",
    "userAgent": "FinancialAdvisorApp/2.1.0"
  },

  "compliance": {
    "consentValid": true,
    "consentId": "uuid",
    "legalBasis": "consent (GDPR Art. 6(1)(a))"
  }
}
```

5.3 GDPR Compliance

GDPR ARTICLE	ZEKNOP IMPLEMENTATION
Art. 6 - Lawfulness	Explicit consent via OAuth 2.0 authorisation
Art. 7 - Consent conditions	Clear consent screens, granular scopes, easy revocation
Art. 15 - Right of access	User can query <code>/consents</code> and <code>/audit-log</code> to see all access
Art. 16 - Right to rectification	User can update identity blocks via data custodian UI
Art. 17 - Right to erasure	User can delete account → all consents revoked, data purged
Art. 20 - Data portability	Core purpose of this standard!
Art. 25 - Privacy by design	Derived claims, minimal disclosure, granular scopes
Art. 30 - Records of processing	Audit trail provides required record
Art. 32 - Security	TLS 1.3, OAuth 2.0, encrypted storage (implementation-specific)

5.4 Australian Privacy Act Compliance

AUSTRALIAN PRIVACY PRINCIPLE	ZEKNOP IMPLEMENTATION
APP 1 - Open and transparent management	Protocol specification is public; audit trail available to users
APP 3 - Collection of solicited information	Only collects data necessary for declared purpose (consent scopes)
APP 5 - Notification of collection	Clear consent screens explain what data is collected and why
APP 6 - Use or disclosure	Data used only for consented purposes; audit trail tracks all access
APP 8 - Cross-border disclosure	Jurisdictional controls on data transfer
APP 11 - Security	Encrypted storage, access controls, audit logging
APP 12 - Access to personal information	Users can view all their data via API
APP 13 - Correction	Users can update identity blocks via data custodian UI

6. Identity Verification (High-Level Framework)

6.1 Verification Levels

ZEKNOP defines **verification levels** as business concept, agnostic to underlying verification technology:

LEVEL	REQUIREMENTS	ASSURANCE
pseudonymous	Self-asserted nickname	None (anonymous)
basic	Email + phone verified (OTP)	Contact reachability
extended	Name, DOB, address verified (documents)	Real-world identity linkage
official	Government-issued ID verified (passport, national ID)	High assurance, government linkage
kyc_verified	Full KYC/AML process including PEP check, sanctions screening	Regulatory compliance (financial services)

6.2 Verification Mechanisms (Out of Scope)

This standard defines **WHAT** data is verified, **NOT HOW** verification occurs.

How users obtain verified identity blocks is subject to separate identity verification protocols, which may include but are not limited to:

- Traditional KYC providers (e.g., bank-issued credentials)
- Government-issued digital identity (eIDAS 2.0, myGovID in Australia, national digital ID programs)
- Document verification services (OCR + fraud detection)
- Other verification methods (details in separate technical specifications)

Implementation Agnostic: Data custodians and consumers accepting ZEKNOP identity blocks do NOT need to know the underlying verification mechanism. They trust the **verificationLevel** and **kycCompletedAt** attestation provided by the identity provider.

Separation of Concerns:

- This standard (PUBLIC): Data structure, verification levels, OAuth scopes
- Verification protocols (SEPARATE): Authentication methods, cryptographic proofs, system architectures

6.3 Identity Provider Certification (Future)

ZEKNOP Foundation (upon establishment) will define **certification criteria** for identity providers:

- **Basic Certification:** Implements pseudonymous, basic, extended levels
- **KYC Certification:** Implements official and kyc_verified levels with audit trail
- **Advanced Certification:** Additional features (biometrics, document verification, fraud detection)

Note: Certification criteria focus on data format compliance and security audits, not on specific verification technologies.

7. Use Case Workflows

7.1 Financial Advisor Client Onboarding

Before ZEKNOP:

1. Client fills AML questionnaire: 30 minutes
2. Client manually lists assets: 1-2 hours
3. Client provides bank statements (PDF): 30 minutes scanning/emailing
4. Advisor manually enters data into CRM: 2-4 hours
5. Total: 5-10 hours, high error rate

With ZEKNOP:

```

1. Client visits advisor's onboarding portal
↓
2. Portal redirects to client's data custodian (current bank)
   OAuth flow initiates with scopes:
   - assets:read
   - liabilities:read
   - policies:read
   - identity:kyc:status
   ↓
3. Client reviews consent screen:
   "Advisor Company ABC requests:
   - Your assets (summary)
   - Your liabilities
   - Your insurance policies
   - Your KYC verification status
   [Approve] [Deny]"
   ↓
4. Client approves → API automatically transfers data
   ↓
5. Advisor's system receives:
   - Complete asset list with current valuations
   - All liabilities with outstanding balances
   - Insurance policies with coverage details
   - KYC status: { isPEP: false, amlRiskLevel: "low" }
   ↓
6. Advisor CRM auto-populates → Client profile ready
   ↓
Total: 5 minutes, 100% accuracy

```

Privacy Preserved:

- Advisor does NOT receive full bank statements (only summaries)
- Advisor does NOT receive passport copies (only KYC status)
- Client retains granular control via consent

7.2 E-Commerce Privacy-Preserving Checkout

Before ZEKNOP:

Eshop collects:

- Full name
- Email
- Phone
- Billing address (full)
- Shipping address (full)
- Payment card details

→ All stored in eshop database

→ Breach exposes residential addresses, payment info

With ZEKNOP (Zero-Knowledge):

1. Customer clicks "Checkout with ZEKNOP"
 - ↓
2. Eshop requests scopes:
 - identity:contact:email
 - identity:userId
 - ↓
3. Customer approves → Eshop receives:


```
{
  "email": "customer@example.com",
  "userId": "550e8400-e29b-41d4-a716-446655440000"
}
```

 - ↓
4. Eshop creates order:


```
{
  "orderId": "ORD-12345",
  "userId": "550e8400-...",
  "email": "customer@example.com",
  "items": [...]
}
```

 - ↓
5. Eshop redirects to carrier authorisation:


```
"Authorise [Carrier Name] to receive your shipping address"
```

 - ↓
6. Customer authorises carrier (separate OAuth):
 - Carrier requests: identity:address:shipping
 - Customer approves
 - Carrier receives shipping address (DIRECTLY from custodian)
 - ↓
7. Carrier validates delivery:
 - Eshop → Carrier API: { "orderId": "ORD-12345", "userId": "550e8400-..." }
 - Carrier validates userId matches OAuth token
 - Carrier delivers to address from OAuth (not from eshop)
 - ↓
8. Payment processed via separate payment gateway

Result:

- Eshop NEVER has shipping address (only email + userId)
- Carrier has shipping address but NO email or customer identity
- Zero-knowledge: No single party has complete customer profile
- Eshop database breach does NOT expose any addresses
- Residential address NEVER disclosed to anyone

7.3 Bank KYC Without Document Storage

Before ZEKNOP:

Bank onboarding:

1. Client provides passport copy (PDF/scan)
 2. Client provides proof of address (utility bill)
 3. Client fills PEP declaration form
 4. Bank stores all documents in compliance database
 5. Bank performs manual AML checks
- Bank now custodian of sensitive documents (GDPR risk, storage cost)

With ZEKNOP:

1. Client applies for bank account
 - ↓
 2. Bank requests scope: identity:kyc:status
 - ↓
 3. Client approves → Bank receives:


```
{
  "userId": "uuid",
  "verificationLevel": "kyc_verified",
  "isPEP": false,
  "amlRiskLevel": "low",
  "sanctionsCheckPassed": true,
  "kycCompletedAt": "2025-11-18T10:00:00Z",
  "kycExpiresAt": "2026-11-18T10:00:00Z"
}
```

 - ↓
 4. Bank verifies:
 - KYC completed by certified provider
 - PEP status: negative
 - AML risk: low
 - Sanctions: clear
 - ↓
 5. Bank approves account opening
 - ↓
- Bank does NOT store:
- Passport copies (remain with KYC provider)
 - Proof of address documents
 - PEP declaration forms
- Reduced GDPR risk, lower storage costs, faster onboarding

If Regulatory Audit Required:

- Regulator requests: Bank provides consent audit trail + KYC provider reference
- KYC provider (as data custodian) provides documents to regulator
- Bank never needs to store sensitive documents

7.4 Event Ticket Security (Anti-Fraud)

Problem: Electronic ticket fraud - users sell tickets multiple times or use counterfeit copies.

Traditional Electronic Tickets:**Issues:**

- PDF tickets can be copied/forwarded multiple times
- No way to verify authenticity at venue
- Secondary market fraud (fake tickets sold)
- Ticket bots buy en masse, resell at inflated prices

With ZEKNOP Membership Block:

```

1. Event organiser issues ticket as membership item
{
  "membershipType": "ticket",
  "provider": "Concert Venue XYZ",
  "barcode": "unique_qr_code",
  "validFrom": "2025-12-01T19:00:00Z",
  "validUntil": "2025-12-01T23:00:00Z",
  "status": "active"
}
↓
2. Original buyer receives ticket in their ZEKNOP identity
↓
3. Buyer wants to transfer to friend:
- Buyer REVOKES own consent to ticket
- System marks original ticket: status = "transferred"
- Friend creates new consent (via OAuth flow)
- Friend receives ticket with NEW barcode
↓
4. At venue entrance:
- Scanner reads QR code
- Validates via API: Is ticket active? Is holder authorised?
- Checks: status = "active", validUntil > now, barcode unique
↓
5. Entry granted, ticket marked as "used"

```

Security Benefits:

- No double-spending: Once transferred, original ticket invalidated
- Cryptographically verifiable: API validates ticket ownership in real-time
- Audit trail: Complete history (issued → transferred → used)
- Anti-scalping option: Organiser can set transfer limits (e.g., max 1 transfer)
- Fraud prevention: Counterfeit tickets fail API validation

Privacy Preserved:

- Venue scanner does NOT receive buyer's full identity
- Scope granted: `identity:membership:tickets:validation` (just ticket status)
- Personal data (name, address, email) remains private

Use Cases:

- Concert tickets (prevent scalping, enable legitimate transfers)
- Sports events (season ticket transfers)
- Public transport passes (validate without collecting PII)
- Conference badges (verify attendee, no personal data exposure)

8. Flagship Use Case: Privacy-Preserving Age Verification

8.1 The Problem

Governments worldwide are mandating age verification to protect minors online. Australia's [Online Safety Act](#) requires designated platforms to take reasonable steps to verify users are of appropriate age (16+ for social media). France's [Social Media Regulation Act \(2023\)](#) prohibits social networks from registering users under 15 without verified parental consent, with enforcement mechanisms under development. Similar requirements exist or are emerging in the UK, EU Digital Services Act, and other jurisdictions.

Current age verification methods create a problematic trade-off:

Traditional Approaches and Their Flaws:

APPROACH	PRIVACY RISK	PRACTICAL ISSUES
ID Document Upload	Creates centralised databases of government IDs vulnerable to breach	Users reluctant to share passports with social networks
Facial Recognition	Biometric data collection; surveillance infrastructure	Accuracy issues; excludes users without cameras
Credit Card Verification	Financial data exposure	Excludes users without credit cards; doesn't verify age accurately
Third-Party KYC Providers	Identity data shared with additional parties	Creates new data silos; user friction

The Core Problem: All these methods require platforms to collect and store sensitive personal data, creating "honeypots" for attackers and undermining user privacy.

8.2 ZEKNOP Solution: Verify the Claim, Not the Identity

Key Insight: Platforms need to know if users meet an age threshold. They do NOT need to know users' dates of birth, names, or other identity details.

ZEKNOP enables trusted institutions to **attest to specific claims** without revealing underlying data.

8.3 How It Works

ZEKNOP AGE VERIFICATION FLOW

- STEP 1: Platform requests age verification
Platform → User: "Please verify you are over 16"
OAuth scope requested: identity:age:over16
- STEP 2: User selects trusted institution
User chooses from:
- Bank (CommBank, ANZ, Westpac, NAB, etc.)
- Government ID provider (myGovID in Australia)
- Other certified identity providers
- STEP 3: User authenticates with institution
User logs in using existing credentials
(No new account required - uses existing bank login)
- STEP 4: Institution generates signed attestation
Institution computes locally: "User is over 16: TRUE"
Institution signs attestation cryptographically
Date of birth NEVER leaves institution's systems
- STEP 5: Platform receives and verifies attestation
Platform receives:
{
 "isOver16": true,
 "verifiedAt": "2026-01-27T10:00:00Z",
 "issuer": "institution_id",
 "signature": "cryptographic_signature"
}
Platform verifies signature against institution's key
- STEP 6: Access granted
User accesses platform
No date of birth stored
No name stored
No government ID stored

8.4 Data Minimisation in Practice

ENTITY	DATA RECEIVED	DATA STORED
Platform	Boolean claim (<code>isOver16: true</code>) + signature	Verification status (user X is age-verified)
Institution	Verification request metadata	Audit log (user requested age verification)
User	Attestation	Under user's control

Critical Point: Date of birth, full name, and other identity attributes **never** leave the institution's existing systems.

8.5 Age Threshold Flexibility

Different jurisdictions and contexts require different age thresholds. ZEKNOP supports multiple age verification scopes:

SCOPE	AGE	USE CASE
<code>identity:age:over13</code>	13+	COPPA compliance (US), children's services
<code>identity:age:over15</code>	15+	French Social Media Ban, platforms requiring 15+
<code>identity:age:over16</code>	16+	Australian Online Safety Act, EU Digital Services Act
<code>identity:age:over18</code>	18+	Adult content, gambling, alcohol (most jurisdictions)
<code>identity:age:over21</code>	21+	US alcohol, certain services
<code>identity:age:over25</code>	25+	Car rental, some insurance products

Platforms request only the scope they need. A social network in Australia requests `identity:age:over16` ; a gambling site requests `identity:age:over18` .

8.6 Technical Implementation

Attestation Format:

```
{
  "claims": {
    "isOver16": true
  },
  "subject": {
    "userId": "platform_specific_pseudonym"
  },
  "issuer": {
    "id": "institution_id",
    "name": "Example Bank",
    "type": "bank"
  },
  "issuedAt": "2026-01-27T10:00:00Z",
  "expiresAt": "2026-01-28T10:00:00Z",
  "nonce": "platform_provided_nonce",
  "signature": "cryptographic_signature"
}
```

Verification Process:

- 1. Platform verifies signature against institution's published public key
- 2. Platform checks attestation has not expired
- 3. Platform verifies nonce matches expected value (prevents replay)
- 4. Platform accepts claim value

Security Standards:

- Signature scheme: Standard cryptographic signatures (RSA, ECDSA, or EdDSA)
- Hash function: SHA-256 for integrity
- Transport security: TLS 1.3 minimum
- Key management: HSM recommended for institutions

8.7 Privacy Guarantees

Unlinkability: Each verification uses platform-specific pseudonyms. Institutions cannot track which platforms users verify for.

Data Minimisation: Only boolean claims are transmitted. No unnecessary data collection.

User Control: Users choose which institution to use and explicitly consent to each verification.

No Central Database: No entity accumulates identity data across verifications.

8.8 Comparison with Alternative Approaches

ASPECT	ID UPLOAD	FACIAL RECOGNITION	CREDIT CARD	ZEKNOP
Data Collected	Full ID document	Biometric template	Financial data	Boolean only
Breach Risk	High (centralised)	High (biometric)	Medium	Minimal
User Control	Low	Low	Low	High
Privacy Preserved	No	No	Partial	Yes
Accessibility	Requires ID	Requires camera	Excludes unbanked	Uses existing accounts

8.9 Regulatory Alignment

Australian Online Safety Act:

- ZEKNOP provides cryptographic proof of age threshold
- Privacy-preserving approach aligns with eSafety Commissioner guidance
- Accessible via existing bank accounts (85%+ coverage) or myGovID

Australian Privacy Act (APPs):

- APP 3 (Collection): Only collects boolean claim necessary for verification
- APP 6 (Use/Disclosure): No personal data to misuse
- APP 11 (Security): Minimal attack surface - no PII stored

French Social Media Regulation (Loi n° 2023-566):

- Requires age verification for social media registration (15+ threshold)
- Enforcement mechanisms under development
- ZEKNOP `identity:age:over15` scope designed for this requirement
- Privacy-preserving approach avoids centralised ID databases

GDPR (EU):

- Privacy by design (Art. 25)
- Data minimisation (Art. 5(1)(c))
- Purpose limitation (Art. 5(1)(b))

8.10 Implementation Considerations

Institution Readiness:

REGION	BANKS	GOVERNMENT ID	COVERAGE
Australia	Big 4 banks (CommBank, ANZ, Westpac, NAB)	myGovID	85%+ adults
France	Major banks (BNP Paribas, Crédit Agricole, Société Générale)	France Identité	70%+ adults
EU	PSD2-compliant banks	eIDAS-compatible national eIDs	Varies by country

- All institutions already have verified customer identities through existing KYC processes
- Integration requires standard API implementation (OAuth 2.0 + signing)
- No new identity infrastructure needed - leverages existing trust relationships

User Experience:

1. User clicks "Verify Age" on platform
2. User selects their bank from list
3. User redirected to bank login (familiar interface)
4. User approves: "Share age verification with [Platform]"
5. User returned to platform, verified

No new accounts, no document uploads, no facial scans.

9. Data Custodian Role and Implementation

9.1 What is a Data Custodian?

Definition: A Data Custodian is any entity that stores user data and implements the ZEKNOP Data Interoperability Standard to enable authorised data sharing.

Responsibilities:

1. Secure Storage: Store user data (financial data, identity blocks) with appropriate security measures
2. API Implementation: Implement ZEKNOP-compliant REST API (OAuth 2.0 authorisation, JSON Schema validation)
3. Consent Management: Respect user consent grants and revocations
4. Audit Logging: Record all data access events for accountability
5. Compliance: Meet GDPR, Privacy Act (Australia), PSD2, and other applicable regulations

Who Can Be a Data Custodian:

- Banks: Store financial assets, liabilities, transaction history
- Insurance Companies: Store policy data, claims history
- KYC/AML Providers: Store verified identity documents, verification status
- Financial Advisors: Store client portfolios (with client consent)
- Specialised Custody Services: Dedicated data portability providers
- Users Themselves: Self-hosted solutions for maximum control

9.2 Implementation Flexibility

The ZEKNOP protocol is architecture-agnostic.

The standard defines:

- WHAT data must be exposed (JSON schemas)

- HOW authorisation works (OAuth 2.0 scopes)
- WHEN access is allowed (consent validation)
- NOT the internal security architecture

How data custodians implement their internal storage, encryption, and access control mechanisms is out of scope of this publication.

This allows institutions to adopt ZEKNOP using their existing infrastructure.

9.3 Certification Program

Implementations can achieve three certification levels:

1. Compatible

Requirements:

- Implements core data models (JSON Schema validation passes)
- Implements OAuth 2.0 authorisation flow
- Passes conformance test suite (150+ test scenarios)
- Provides audit trail logging

Certification Method: Self-certified (automated test suite)

2. Certified

Requirements:

- All "Compatible" requirements
- Third-party security audit passed
- Penetration testing completed
- Applicable privacy legislation compliance documented (GDPR, Australian Privacy Act, CCPA, etc.)

Certification Method: Independent auditor evaluation

Recommendation: Production-ready for general use

3. Secure+ (Advanced Assurance)

Requirements:

- All "Certified" requirements
- SOC 2 Type II OR ISO 27001 certification
- Advanced security features documented
- High-availability guarantees (99.9%+ uptime)

Certification Method: Industry-standard compliance audits

Recommendation: High-security environments (banking, government, high-value assets)

10. Governance and Open Standard Development

10.1 Foundation Model

ZEKNOP will be governed by an independent **Stichting** (Dutch foundation):

Structure:

- **Technical Committee (TC):** Reviews and approves RFC proposals, manages standard versioning
- **Certification Body:** Evaluates implementation compliance, issues certifications
- **Community:** Open participation in RFC discussions, implementation feedback

Principles:

- **Vendor Neutrality:** No single commercial entity controls the standard
- **Transparency:** All RFC discussions and decisions public
- **Backward Compatibility:** Semantic versioning, deprecation policies, LTS support

10.2 RFC Process**Request for Comments (RFC) Workflow:**

1. Community member proposes change (new data type, API endpoint, scope)
2. Draft RFC published for 14-day public comment period
3. TC reviews comments, proposes amendments
4. TC votes on RFC adoption (majority required)
5. If approved: Spec updated, conformance tests added, implementations updated

Example RFCs:

- RFC-0001: Defensive Publication (this document)
- Future RFCs: Extensions to data types, additional privacy patterns, industry-specific integrations

10.3 Certification Program

Implementations can achieve three certification levels:

1. Compatible:

- Implements core data models (JSON Schema valid)
- Passes conformance tests (data format, API responses)
- Self-certified (automated test suite)

2. Certified:

- Security audit passed (third-party)
- OAuth 2.0 implementation reviewed
- Production-ready recommendation

3. Secure+:

- Advanced certifications: SOC 2 Type II, ISO 27001, FIPS 140-2
- High-assurance environments (banking, government)

11. Scope of Defensive Publication**11.1 Scope of This Defensive Publication**

This whitepaper establishes **prior art** for the following data model structures, API patterns, and privacy principles:

Asset & Liability Portability (PUBLIC DOMAIN):

- JSON Schema data models for assets (14+ types including real estate, securities, commodities, cryptocurrencies), liabilities (11+ types), insurance policies (12+ types)
- External type registries with hierarchical naming (e.g., `eu.banking.current_account` , `us.retirement.401k` , `au.superannuation.accumulation`)
- Jurisdictional data models: core fields (mandatory) plus jurisdiction-specific extensions
- Cross-jurisdictional data transfer with fallback types
- OpenAPI 3.0 specification with REST endpoints
- OAuth 2.0 consent flow with granular scopes
- Audit trail structure (consent events, data access logs)

- Pagination, filtering, incremental sync patterns
- Data migration API (export/import workflows)

Identity Data Framework (PUBLIC DOMAIN):

- Granular identity blocks (pseudonymous, contact, address, financial, KYC, documents, biometric, membership, digitalLife)
- Universal `userId` as linking mechanism across blocks
- Verification levels (pseudonymous → official → kyc_verified) as business concept
- Derived claims pattern (isPEP, amlRiskLevel, photographVerified, isOver13, isOver15, isOver16, isOver18, isOver21) without underlying documents
- Privacy-by-design principles: minimal disclosure, purpose-specific data sharing
- Biometric data references from official documents (passport photos, ID fingerprints)
- Age verification scopes and attestation format

Privacy-by-Design Patterns (PUBLIC DOMAIN):

- Derived claims without document storage (e.g., bank receives isPEP: false, not declaration)
- Zero-knowledge data flow (eshop receives email, carrier receives address, neither has both)
- Age verification via boolean attestation (platform receives isOver16: true, not date of birth)
- Time-bound and revocable consent
- Audit trail for accountability

Governance Model (PUBLIC DOMAIN):

- Foundation structure (Stichting), Technical Committee, RFC process
- Certification program (Compatible, Certified, Secure+)
- Open community participation

The information disclosed herein is **sufficient for skilled practitioners to implement interoperable data portability systems** using standard technologies (OAuth 2.0, JSON Schema, REST APIs), thereby preventing third-party patents on these data model structures and privacy patterns.

11.2 Declaration

This work creates a **public record** of the disclosed data model standards, API patterns, and privacy-by-design principles as of the publication date (2026-01-27).

These data structures and patterns are **contributed to the public domain** to enable ecosystem-wide interoperability and prevent vendor lock-in.

The authors and ZEKNOP Stichting (upon formation) reserve all rights regarding technical implementations and protocols not explicitly disclosed herein.

12. Security Considerations

12.1 Threat Model

OAuth 2.0 Authorisation Threats:

- Token theft: Mitigated by PKCE, short token expiration (1 hour), refresh token rotation

- Phishing/Impersonation: Mitigated by multiple layers:
 - **Consumer registration:** All data consumers must be registered and verified before requesting any scopes (unverified entities cannot participate in the protocol)
 - **Pre-configured scope limits:** Consumers can only request scopes they were approved for during registration (e.g., a social network cannot request financial data)
 - **Institution verification:** Data custodians verify consumer identity via registered redirect URIs, client certificates, or mutual TLS
 - **User awareness:** Clear consent screens showing verified consumer identity and requested scopes
- Man-in-the-middle: Mitigated by TLS 1.3 mandatory for all communications

Data Exposure Threats:

- Over-collection: Mitigated by granular scopes, consumers request minimal data
- Unauthorised access: Mitigated by scope validation at API server, audit trail
- Consent fatigue: Mitigated by clear consent screens, time-bound consents

Implementation-Specific Threats:

- Database breach: Data custodian must encrypt data at rest (implementation-specific)
- API server compromise: Rate limiting, WAF, intrusion detection (implementation-specific)
- Insider threats: Audit trail provides accountability, least-privilege access controls

12.2 Cryptographic Requirements

Transport Security:

- TLS 1.3 or TLS 1.2 (minimum) for all API communications
- Certificate pinning recommended for mobile apps
- HTTP Strict Transport Security (HSTS) headers

Data-at-Rest Encryption:

- AES-256-GCM or ChaCha20-Poly1305 for sensitive fields (nationalId, taxId, bankAccount)
- Key management: Implementation-specific (HSM, KMS, or other approaches)

Token Security:

- OAuth access tokens: JWT (RS256 or EdDSA signatures)
- Refresh tokens: Opaque, single-use, encrypted storage
- Token expiration: Access token 1 hour, refresh token 30 days (configurable)

12.3 Compliance Standards

Implementations SHOULD pursue:

Security Standards:

- SOC 2 Type II (security, availability, confidentiality)
- ISO 27001 (information security management)

Privacy Legislation (jurisdiction-dependent):

- GDPR (EU) - see Section 5.3
- Australian Privacy Act (AU) - see Section 5.4
- French Data Protection Act (FR) - Loi Informatique et Libertés, aligned with GDPR
- CCPA/CPRA (US California) - if serving California residents

Financial Regulations (if financial data involved):

- PSD2 (EU) - Payment Services Directive
- CDR (AU) - Consumer Data Right

- Open Banking (UK) - Open Banking Implementation Entity standards

Age Verification Regulations (if age verification implemented):

- Online Safety Act (AU) - see Section 8.9
 - Social Media Regulation Act (FR) - see Section 8.9
-

13. Limitations and Future Work

13.1 Current Scope Limitations

Financial Data Coverage:

- Currently defines 14 asset types, 11 liability types, 12 policy types
- Future: Real estate details, collectibles, intellectual property, business assets

Geographic Coverage:

- Data models optimised for EU (including CZ, FR), AU financial systems
- Future: Region-specific extensions (additional US types, UK post-Brexit, Asian markets, etc.)

Healthcare Data:

- Not currently addressed
- Future: FHIR integration, medical records portability

13.2 Adoption Challenges

Institution Resistance:

- Banks may resist enabling data export (competitive concern)
- Mitigation: Regulatory mandates (PSD2, data portability rights), customer demand

Standardisation Complexity:

- Financial data highly heterogeneous across institutions
- Mitigation: Extensible data models, versioning, backward compatibility

User Experience:

- OAuth consent screens can be complex for non-technical users
- Mitigation: Clear language, visual consent screens, pre-approved templates

13.3 Future Enhancements

The ZEKNOP standard is designed to be **extensible** to accommodate evolving requirements and additional use cases. The Technical Committee will evaluate community proposals for new data types, privacy patterns, and governance mechanisms through the RFC process.

Example Extension Areas (Non-Exhaustive):

- Additional asset classes and liability types via external type registry
- Cross-border data portability enhancements for new jurisdictions
- Real-time data synchronisation mechanisms
- Advanced cryptographic protocols as technology matures

Specific enhancements will be determined by community needs and regulatory requirements, published via the RFC process after appropriate review and public consultation.

14. Conclusion

ZEKNOP Data Interoperability Standard establishes an **open, privacy-by-design framework** for financial data portability and granular identity management. By standardising data models, API patterns, and privacy principles in the public domain, this specification enables ecosystem-wide interoperability while preventing vendor lock-in.

Key Contributions

- 1. Financial Data Standardisation:** JSON Schema models for 30+ asset/liability/policy types, enabling automated data transfer between institutions.
- 2. Granular Identity Framework:** Privacy-preserving identity blocks with derived claims, allowing services to request minimal data (e.g., isPEP status without documents, age verification without date of birth).
- 3. Privacy-by-Design Patterns:** Embedded at protocol level (minimal disclosure, purpose-specific data sharing), not optional features.
- 4. Age Verification Solution:** Demonstrates how to comply with regulations like Australia's Online Safety Act while preserving user privacy.
- 5. Open Governance:** Community-driven standard under independent foundation, ensuring no single vendor controls the ecosystem.

Defensive Publication Purpose

This whitepaper establishes **prior art** for data model structures, API patterns, and privacy principles, preventing third-party patents on these interoperability foundations while allowing authors to pursue intellectual property protection for specific technical implementations.

Call to Action

The ZEKNOP standard is designed as a **multi-stakeholder ecosystem**. We invite:

- **Australian Regulators and Platforms:** We invite Australian regulators, platforms, and financial institutions to engage in pilot programs and discuss implementation of this standard in the context of the Online Safety Act. ZEKNOP provides a privacy-preserving path to compliance.
- **Financial Institutions:** Pilot data portability implementations, join Technical Committee
- **Fintech Companies:** Build consumer applications using ZEKNOP APIs
- **Identity Providers:** Implement granular identity blocks, pursue certification
- **Developers:** Contribute to reference implementations, conformance tests
- **Regulators:** Provide feedback on compliance frameworks, adoption incentives
- **Researchers:** Review specifications, propose enhancements via RFC process

Contact: info@zeknop.org

Website: <https://zeknop.org>

Repository: [To be published with reference implementations]

15. References

ZEKNOP Specifications (CC BY 4.0)

Publication Status Note:

Full technical specifications (JSON Schemas, OpenAPI specs) are referenced throughout this document as implementation guidance. These specifications will be published in the ZEKNOP GitHub repository upon foundation establishment (Q1 2026) and licensed under CC BY 4.0.

Referenced file paths:

- `schemas/asset.json` - Asset data model

- [schemas/liability.json](#) - Liability data model
- [schemas/policy.json](#) - Insurance policy data model
- [schemas/identity-blocks.json](#) - Identity block schemas
- [schemas/person-profile.json](#) - Consolidated profile schema
- [specs/data-portability/openapi.yaml](#) - Complete REST API specification

Specifications:

1. Financial Data Portability Standard - Complete JSON Schema definitions
2. Identity Data Framework - Detailed block specifications
3. Integration Guide - End-to-end implementation examples
4. Test Scenarios - 150+ conformance tests

Standards and Protocols

5. OAuth 2.0 (RFC 6749): <https://datatracker.ietf.org/doc/html/rfc6749>
6. OAuth 2.0 PKCE (RFC 7636): <https://datatracker.ietf.org/doc/html/rfc7636>
7. OpenID Connect: <https://openid.net/connect/>
8. JSON Schema (Draft-07): <https://json-schema.org/draft-07/schema>
9. OpenAPI 3.0: <https://swagger.io/specification/>

Compliance and Privacy

10. GDPR (EU 2016/679): <https://gdpr-info.eu/>
11. eIDAS Regulation (EU 910/2014): <https://digital-strategy.ec.europa.eu/en/policies/eidas-regulation>
12. PSD2 (EU 2015/2366): Payment Services Directive
13. ISO 27001: Information Security Management
14. SOC 2: Service Organisation Control (AICPA)

Australian Regulatory Framework

15. Online Safety Act 2021 (as amended): <https://www.legislation.gov.au/Details/C2021A00076>
16. Privacy Act 1988 (Cth): <https://www.legislation.gov.au/Details/C2014C00076>
17. Australian Privacy Principles (APPs): <https://www.oaic.gov.au/privacy/australian-privacy-principles>
18. eSafety Commissioner - Age Verification: <https://www.esafety.gov.au/industry/age-verification>

Related Work

19. W3C Decentralised Identifiers (DID): <https://www.w3.org/TR/did-core/>
20. W3C Verifiable Credentials (VC): <https://www.w3.org/TR/vc-data-model/>
21. OpenBanking UK Standard: <https://www.openbanking.org.uk/>
22. FHIR (Healthcare): <https://www.hl7.org/fhir/>

16. Metadata for Zenodo Publication

Title: ZEKNOP Data Interoperability Standard: Asset Portability and Privacy-by-Design Identity Framework - Defensive Publication v1.1

Authors:

- Vladislav Urbánek (ORCID: [0009-0008-9183-5754](https://orcid.org/0009-0008-9183-5754))

Date: 2026-01-27

Keywords: asset portability, data interoperability, privacy by design, granular consent, OAuth 2.0, JSON Schema, identity management, derived claims, minimal disclosure, age verification, online safety, Australian Online Safety Act, open standard,

GDPR compliance, Privacy Act compliance, digital identity blocks, loyalty programs, metaverse, financial assets, real estate, cryptocurrencies, superannuation

Subjects: Computer Science - Cryptography and Security; Computer Science - Software Engineering; Economics - Financial Technology

License: CC BY 4.0 (<https://creativecommons.org/licenses/by/4.0/>)

Abstract: See Section "Abstract" above

Related Resources:

- Project Website: <https://zeknop.org> (planned)
- GitHub Repository: [To be published]
- Previous Version: DOI 10.5281/zenodo.17666290

Funding: [To be added if applicable]

Version: 1.1 (Defensive Publication - Asset Portability & Identity Framework with Age Verification Focus)

File Format: Markdown (.md) + PDF export

Document Type: Working Paper / Technical Specification

Language: English

Hash (SHA-256): [To be computed upon finalisation]

Changelog (v1.0 → v1.1)

Added

- **Executive Summary:** Enhanced to clearly articulate the problem-solution-value proposition for non-technical stakeholders
- **Section 8: Flagship Use Case - Privacy-Preserving Age Verification:** New dedicated section with detailed workflow, technical implementation, and regulatory alignment
- **Australian regulatory context:** Added Online Safety Act references throughout document
- **French regulatory context:** Added Social Media Regulation Act (Loi n° 2023-566) references for 15+ age verification
- **Australia in jurisdiction table:** Added row for AU with Online Safety Act, Privacy Act (APPs), AML/CTF Act
- **France in jurisdiction table:** Added row for FR with Social Media Regulation Act, GDPR
- **Australian Privacy Act compliance table:** New Section 5.4 mapping APPs to ZEKNOP implementation
- **Age verification scopes:** Added `identity:age:over13`, `identity:age:over15` (FR), `identity:age:over16` (AU) for jurisdictional flexibility
- **Australian financial product types:** Added `au.superannuation.*`, `au.banking.*`, `au.mortgage.*`, `au.loan.hecs` to registries
- **Australian references:** Added Online Safety Act, Privacy Act, APPs, eSafety Commissioner to references
- **French references:** Added Social Media Regulation Act, France Identité to references
- **Section 12.1 Security:** Enhanced phishing/impersonation mitigation with consumer registration, pre-configured scope limits, and institution verification
- **Section 12.3 Compliance Standards:** Restructured with categories (Security, Privacy, Financial, Age Verification) and added CDR (AU), Open Banking (UK), CCPA (US)

Changed

- **Executive Summary:** Completely restructured to begin with the fundamental problem of data fragmentation ("Data about individuals are now distributed across hundreds of institutions..."), addressing both privacy risks and systemic inefficiencies; age verification positioned as a pressing example, not the sole purpose
- **Problem Statement (Section 1.1):** Renamed "Age Verification Crisis" to "Identity Verification Dilemma" to reflect broader scope; age verification framed as "a pressing example"

- Use Cases (Section 1.3): Age verification marked as "Flagship Use Case" with framing "As a primary example of the protocol in action"
- Section 8.10 Implementation Considerations: Expanded with multi-jurisdictional table (AU, FR, EU) for institution readiness
- Section 9.3 Certification Program: Made jurisdiction-agnostic ("Applicable privacy legislation" instead of specific laws)
- Section 11.1 Scope: Updated derived claims list to include all age thresholds (isOver13, isOver15, isOver16, isOver18, isOver21)
- Section 13.1 Geographic Coverage: Updated to "EU (including CZ, FR), AU"
- Call to Action (Section 14): Added specific invitation to Australian regulators, platforms, and financial institutions

Fixed

- Identity blocks count: Verified as 9 blocks (consistent throughout document)
- Terminology consistency: Standardised on "ZEKNOP" for protocol, "ZEKNOP Data Interoperability Standard" for full name
- Framing balance: Refined to ensure age verification is presented as flagship demonstration, not sole purpose of the protocol

Removed

- `identity:digitalLife:ageVerification` scope: Removed as redundant with `identity:age:overXX` scopes

*Made with care for the open interoperability community
2025-2026 ZEKNOP Authors - Licensed under CC BY 4.0*