System Architecture & Catalog

Purpose

End-to-end Digital Public Infrastructure (DPI) for capturing actions and omissions as verifiable events, enforcing policy, and publishing tamper-evident proofs with selective disclosure. Built for Canada federal pilots and scalable worldwide.

Component Diagram (PlantUML source)

```
@startuml
skinparam componentStyle rectangle
actor User
actor SourceSystem as SS
package "Ingress" {
  [Collector API]
  [Connector Runners]
}
package "Processing" {
  [Pseudonymization Service]
  [Policy Engine (OPA)]
  [Schema Registry]
  [Merkle/Proof Builder]
}
package "Data Plane" {
  database "PostgreSQL (Aurora)" as PG
  node "Object Store (S3)" as S3
  [Search Index (OpenSearch)]
}
package "Access" {
  [Admin Console (GCWeb, EN/FR)]
  [Public Proof Portal]
  [VC Issuer/Verifier]
}
package "Ops & Assurance" {
  [CICE Agent]
  [Observability (Logs/Metrics/Traces)]
  [Anchoring (OTS)]
  [Transparency Export (Trillian)]
}
SS --> [Connector Runners]
[Connector Runners] --> [Collector API]
[Collector API] --> [Pseudonymization Service]
[Pseudonymization Service] --> [Policy Engine (OPA)]
[Policy Engine (OPA)] --> PG
[Policy Engine (OPA)] --> S3
[Policy Engine (OPA)] --> [Search Index (OpenSearch)]
[Collector API] ..> [Schema Registry] : validate
[Merkle/Proof Builder] --> S3 : proofs (WORM)
[Admin Console (GCWeb, EN/FR)] --> PG
[Admin Console (GCWeb, EN/FR)] --> [Search Index (OpenSearch)]
[Public Proof Portal] --> S3 : verify proofs
[VC Issuer/Verifier] --> PG
[VC Issuer/Verifier] --> S3
[CICE Agent] --> [Collector API] : control attestations
[Anchoring (OTS)] --> S3 : merkle roots
[Transparency Export (Trillian)] --> S3 : roots
User --> [Admin Console (GCWeb, EN/FR)]
User --> [Public Proof Portal]
@enduml
```

Event → Proof → Verification Sequence (PlantUML source)

```
@startuml
actor Connector
participant Collector
```

```
participant Pseudo as "Pseudonymization"
participant OPA as "Policy (OPA)"
participant Store as "Event Store"
participant Proof as "Proof Builder"
participant WORM as "WORM (S3 Object Lock)"
participant Portal as "Public Verifier"
Connector -> Collector: POST /ingest (event)
Collector -> Pseudo: tokenize identifiers
Collector -> OPA: authorize + retention class
OPA -> Store: append event (hash-chain)
Proof -> Store: read accepted events (hourly)
Proof -> Proof: build Merkle tree
Proof -> WORM: publish {root, inclusion proofs}
Portal -> WORM: fetch proof bundle
Portal -> Portal: verify inclusion
@enduml
```

AWS Canada Deployment Topology (summary)

Region ca-central-1 (optional ca-west-1 DR), ECS Fargate/EKS on private subnets, ALB+WAF edge, Aurora PostgreSQL (PITR), S3 (Object Lock) for proofs/logs, optional OpenSearch, SSO via SSC, WebAuthn MFA, SCIM, CloudWatch + OpenTelemetry, SCP deny non-CA, mandatory encryption.

Component & API Catalog (summary)

Services: Collector API; Connector Runners; Pseudonymization; Policy Engine (OPA); Merkle/Proof Builder; Admin Console (GCWeb EN/FR); Public Proof Portal; VC Issuer/Verifier; CICE Agent; Anchoring (OpenTimestamps); Transparency Export (Trillian).

Key APIs: POST /ingest; GET /schemas/{version}; POST /tokenize; POST /policy/decide; GET /proofs/{period}; POST /vc/issue; POST /vc/verify.

Data Stores: PostgreSQL (event, proof_batch, access_log, policy); S3 (payload, proofs WORM, logs); OpenSearch (tokenized attributes).

Policies: ABAC deny-by-default; retention with legal holds; Object Lock retention mapping.

Observability: ingest/backlog/proof-lag/OPA-denials/KMS-errors dashboards; SLO burn; chain-break alerts; WORM failures; anomalous connector alerts.