# Applying YANG Provenance Signatures in CBOR Objects

-

# IETF Hackathon

IETF 125

14–15 March Shenzen 2026

Shenzen, China

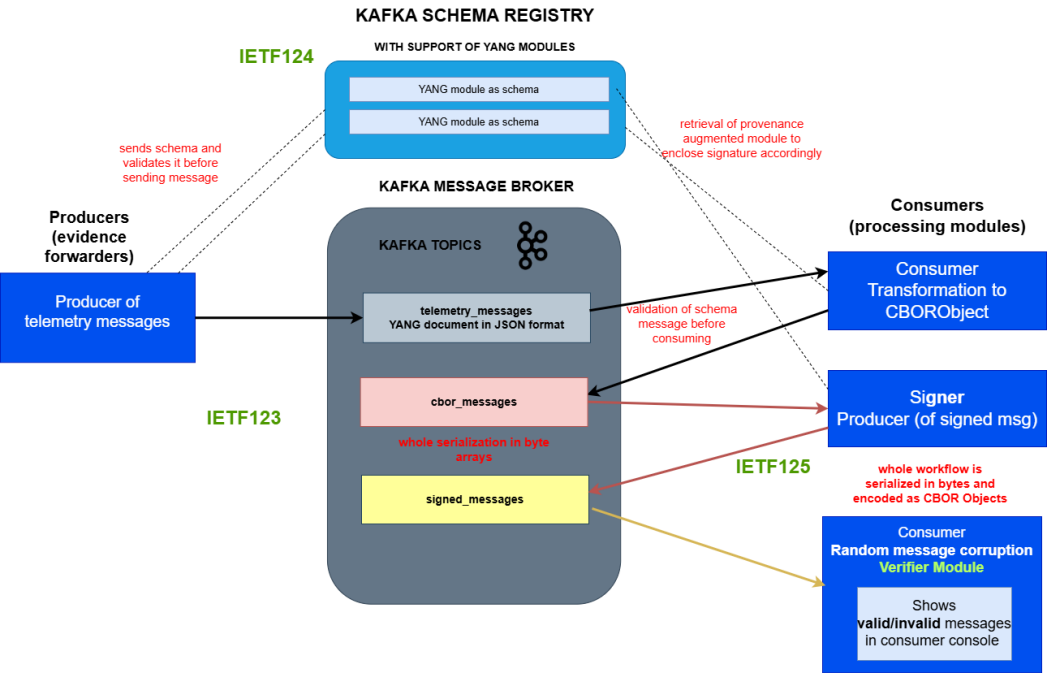
# Prov·e·nance | ˈprävən(ə)ns |

- More specifically, *data provenance*
  - A documented trail accounting for the origin of a piece of data and where it has moved from to where it is presently
- Assurance of the origin and integrity of YANG datasets
  - Whenever the dataset is used beyond an original online flow
    - Use of data intermediaries, such as data lakes
    - AI/ML training and validation
    - Audit trails, including forensics evidence
- Initial use cases
  - Device configuration integrity
  - Telemetry and monitoring data source and integrity
  - Network-wide service orchestration assurance
- Procedures defined in draft-ietf-opsawg-yang-provenance

# What We Brought

- Following previous IETF123 and IETF124 hackathons demo:
  1. End-to-end workflow in a Kafka message broker: data ingestion, serialization, formatting, and Signer/Verifier modules
  2. YANG schema validation prior to data processing
  3. New workflow, operating on byte serialization and CBOR object management (binary only, no JSON)

# Kafka Integration Details




1. The **producer** sends data to a topic, validating its YANG schema beforehand against the YANG schema registry (draft-ietf-nmop-yang-message-broker-integration-09 - An Architecture for YANG-Push to Message Broker Integration)

2. The first consumer validates the schema message before consuming it. The Yang Document is **transformed into a CBOR Object,** and from this point onward all message serialization is performed in binary.

3. The second consumer **signs** the given CBOR data according to the provenance YANG module augmentation retrieved from the YANG Kafka schema registry and forwards it to *signed_messages* topic.

3. The **verifier module** consumes from *signed_messages*, to simulate **tampering**, some messages are randomly altered before being cryptographically **verified,** and any differences are displayed in the console.

# What We Achieved Here

- Demonstrated RI evolution, aligned with the latest draft updates
- Enhanced Kafka integration, building on the IETF 123 & 124 demonstrations
- Presented a complete workflow using byte array serialization and CBOR Object management, covering signing, enclosing, and validating procedures
  - Adopting CBOR data formats and binary encoding reduces message size, enables faster processing, and improves extensibility
- Continued progress in convergence with YANG Push and Kafka Schema Registry
- Next Hackathon → provide a multi-signing implementation