MISP

INTELLIGENCE SHARING

MISP Standard

Formats, shared vocabularies and interoperable intelligence

CIRCL / Team MISP Project

www.misp-standard.org

Standards reviewed: 3 November 2026

Why a family of MISP standards?

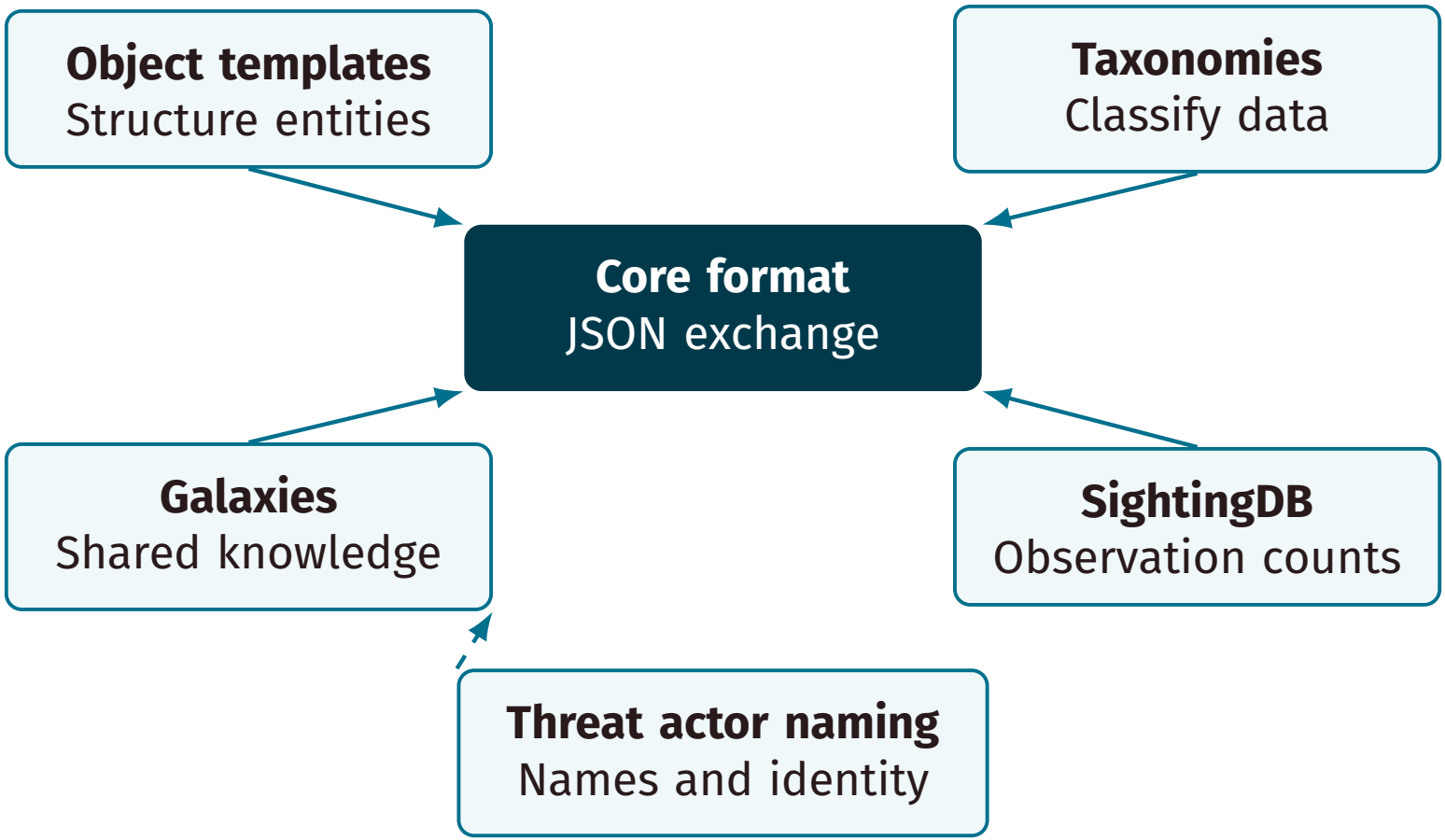
- Exchange intelligence with a shared structure and meaning across MISP instances, other platforms and security tools.
- Reuse JSON formats and public vocabularies for CTI, fraud, vulnerability information, DFIR and other intelligence domains.
- Separate the **exchange format**, the **templates and vocabularies**, and the **software implementing them**.

After this lesson

Choose the right specification, explain how its parts fit together, and check the revision and implementation before integrating.

MISP published standards and specification links

How the published standards fit together



Arrows describe roles, not a mandatory processing pipeline.

Six entries in the MISP published standards catalog

Revision snapshot: 3 November 2026

Specification	MISP site	Latest IETF submission
Core	draft-20	-20; expired
Object template	draft-06	-06; expired
Taxonomy	draft-11	-12; expired
Galaxy	draft-08	-08; expired

- Naming guidance: site labels **draft-00**, dated 6 August 2026. SightingDB: site text dated **13 April 2020**.
- Draft revisions, library versions and MISP software releases are separate version sequences.
- Internet-Drafts are work in progress; submission does not establish IETF endorsement or RFC status.

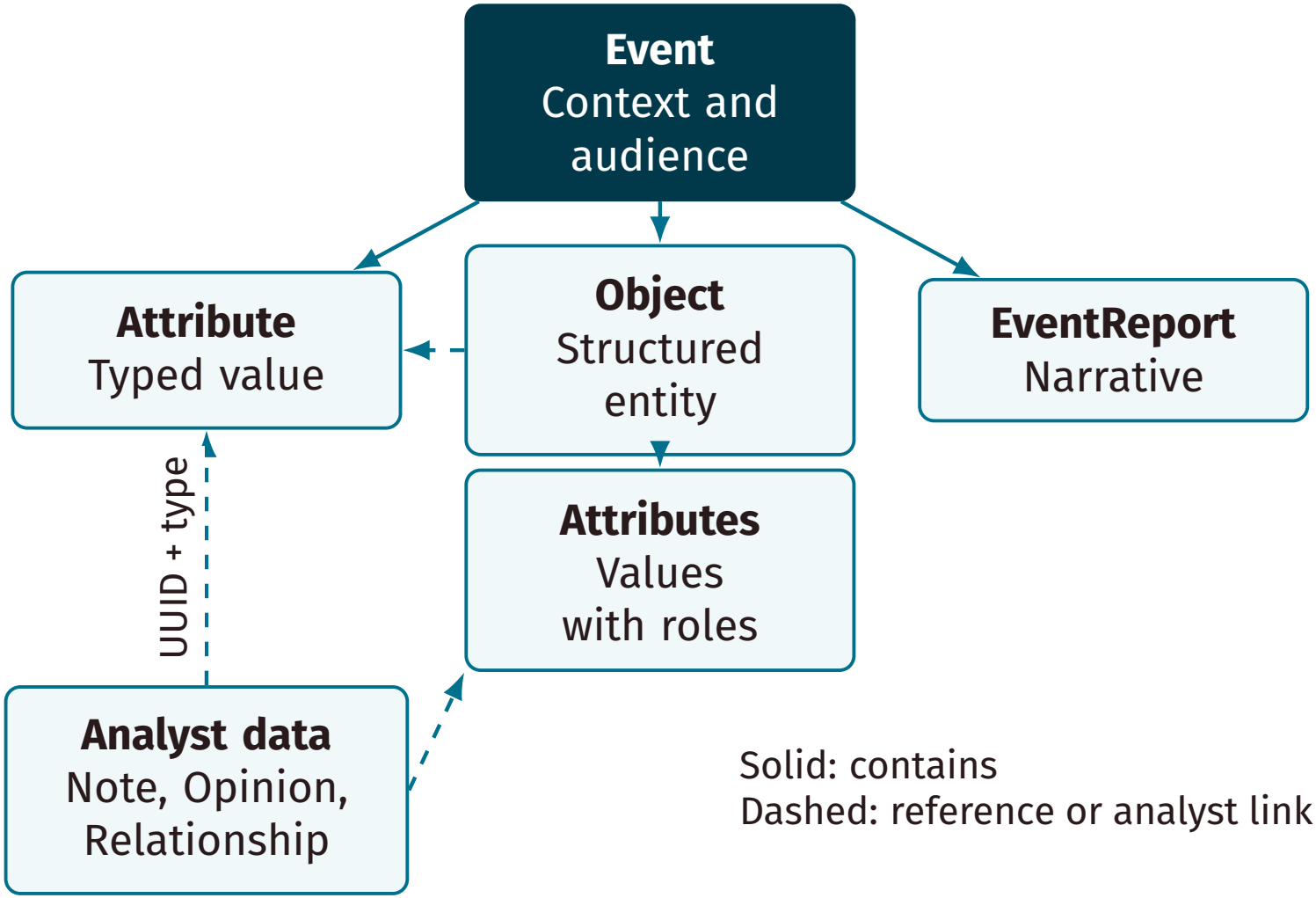
Published documents; IETF revisions above link to Datatracker

Core format: more than a list of indicators

- **Event**: context, organisations, publication state and distribution, with attributes and objects.
- **Attribute**: a typed value and its context. **Object**: related attributes describing one entity; references connect entities.
- **EventReport**, tags, galaxies and sightings add narrative, classification, knowledge and observations.
- **Analyst data**: Note, Opinion and Relationship add analysis linked to identified data, including events, objects and attributes.

Core draft-20: format sections 2.2–2.11

Reading the core data model



Simplified relationships; not a JSON schema.

Core format: containment, object references and analyst data

Object templates: reusable blueprints

- A template defines the meaning and expected attributes of an object, such as a file, domain–IP association or account.
- Its **UUID and version** identify the blueprint. An object instance carries `template_uuid` and `template_version`.
- Elements define attribute types and roles; `required`, `requiredOneOf` and `multiple` express constraints.
- Object references use relationship vocabulary to link an object to another object or an attribute.

Template versus instance

A template describes the fields; an object holds the observed values.

Object template draft-o6

Public template and relationship library: MISP/misp-objects

Taxonomies: shared classification

- JSON vocabularies define a **namespace**, **predicates** and optional **values**, with descriptions and version metadata.
- Machine tags use `namespace:predicate="value"`; tags without a value also exist.
- UUIDs identify the taxonomy, predicates and values; preserve identity across updates.

A machine tag expresses an assessment

```
estimative-language:likelihood-probability=
"very-likely"
```

A classification tag supplies meaning. Distribution and sharing groups control MISP visibility.

Latest submitted taxonomy draft: -12

Public vocabulary library: MISP/misp-taxonomies

Galaxies: reusable knowledge and relationships

- A **galaxy** describes a collection; its **clusters** represent concepts such as threat actors, malware, tools or techniques.
- Cluster values have stable UUIDs, descriptions and metadata such as references, synonyms and external identifiers.
- `related` links clusters using `dest-uuid`, a relationship `type` and optional classification tags.
- Attach this knowledge to events or attributes while retaining references and uncertainty.

Two complementary roles

Taxonomy: classify the assessment. Galaxy cluster: identify and describe a reusable concept.

Galaxy draft-08

Public galaxy and cluster library: MISP/misp-galaxy

Threat actor naming: identity and provenance

- Search existing definitions; reuse matching UUIDs and retain supported aliases and historical relationships.
- Prefer unique, searchable ASCII names; keep actor names distinct from malware, tools and techniques.
- Record who allocated a vendor name. In cluster metadata, `name-attribution` entries use `name:producer-cluster-uuid`.
- Shared tooling or infrastructure alone does not establish that two actor definitions are equivalent.

```
"meta": {
  "name-attribution": [
    "TA511:cae79680-67a6-4411-903c-f824dbcc813f"
  ]
}
```

Naming draft-00: recommendations and vendor name attribution

Sightings and SightingDB serve different roles

Core Sighting

Attribute observations distinguish a sighting, a false-positive report and an expiration indication.

SightingDB query format

A separate format for a compatible connector: counts and observation times for a value in a namespace, plus tags, TTL and a consensus counter.

- SightingDB includes `first_seen`, `last_seen` and `count`; it also defines bulk requests.
- Its `consensus` counts namespaces containing a value. It is not a probability that the value is malicious.

Core format: Sighting

SightingDB specification: 13 April 2020

Supporting formats in the repository

- **Warninglists**: JSON lists of values that can flag potential false positives, such as common services or reserved addresses. A match calls for review.
- **Noticelists**: JSON notices about legal, privacy, policy or technical implications of using or sharing particular data.
- **Query format**: JSON search criteria and output selection for the REST interface. Check the deployed API documentation for supported parameters.

Scope and maturity

These are additional documents in `MISP/misp-rfc`. They are separate from the six entries in the published standards catalog; they are not all submitted IETF drafts.

MISP/misp-rfc: warninglist, noticelist and query format sources

Interoperability: preserve meaning as well as JSON

1. **Identify**: preserve UUIDs; keep instance-local numeric IDs separate from portable identity.
2. **Interpret**: retain types, object roles, template versions, relationships, timestamps and source context.
3. **Control sharing**: honour distribution and sharing-group constraints, including on attached analysis.
4. **Validate**: use relevant schemas and real round trips through the target implementation; inspect unsupported fields and conversion losses.

A useful integration question

Can the receiver reconstruct the context, identity and sharing intent of the intelligence it received?

Core semantics and JSON Schema; verify against deployed software

Contribute and keep the standards current

- Propose specification fixes in MISP/misp-rfc, editing the relevant `raw.md` source and providing examples.
- Regenerate derived XML, text and HTML using the document's build instructions; review the output.
- Contribute reusable content to the object, taxonomy or galaxy library that owns it.
- Check the published document *and* Datatracker before citing a latest revision or submission status.

Keep three references together

Specification revision, library revision and deployed software/API version.

Specification sources and build instructions

Published catalog; full review evidence in this deck's README