THE OPEN RULEBOOK

ADCR

The AI Decision Check Record

What an AI decision record must prove under EU law

Version	0.2.0
Status	development
Date	11 November 2026
Editor	**ComplyEdge**
Licence	Text: Community Specification License 1.0. Schema, test vectors and verifier: Apache-2.0.
Stable URL	https://complyedge.io/rulebook/0.2.0/spec/
Cite as	`ADCR 0.2.0, classes <list>`

Not a certification, not endorsed by any regulator or standards body, and not a harmonised standard.

ADCR, the AI Decision Check Record: what an AI decision record must prove under EU law

Version 0.2.0, stability: development. Schema URL: `https://complyedge.io/rulebook/0.2.0`. Changes from 0.1.0 are listed in Annex D.

Editor: ComplyEdge (first implementation). **Licence:** the text of this document is licensed under the Community Specification License 1.0; the schema, the test vectors and the verifier are licensed under Apache-2.0.

Name: ADCR is the working name of this Rulebook (section 11 records how it was chosen). The Rulebook is the editor's open specification of the record; the **ComplyEdge** rule set (the Rego rules) is a separate artifact, and this Rulebook calls any such collection of rules a *rule set*.

Status note: a version is called *stable* only after one implementation by people other than the editor has passed the public test vectors. Until then every rule here may change at a minor version.

Contents

0. In one page 4

1. Scope, terms and keywords 5
- 1.1 Scope 5
- 1.2 Terms 5
- 1.3 Keywords 5
- 1.4 Conformance classes 6

2. The record (REC) 7
- 2.1 Form 7
- 2.2 Actors 7
- 2.3 Content commitments and pointers 8
- 2.4 The decision 8
- 2.5 Retention 9

3. The rule and the law (LAW) 10

4. Chain and anchor (CHN, ANC) 11
- 4.1 Write-time chain 11
- 4.2 Qualified anchor 11

5. Commitments and pseudonyms 13

6. Oversight, rights and notices (OVS, RGT) 14
- 6.1 Human oversight 14
- 6.2 Decision mode and rights 14

7. What is excluded by name 16

8. Export and verification (EXP) 17

9. Reading the record: the judge's crosswalk 19
- 9.1 Clause crosswalk 19
- 9.2 Properties judges test 21
- 9.3 Procurement and attestation sentences the classes answer 21

10. Profiles, vectors and conformance claims 23

11. What this version does not settle 25

Annex A. Record schema 26

Annex B. Interpretation guide (informative) 27

Annex C. Mapping to the editor's first implementation (informative) 28
Annex D. Changes from 0.1.0 (informative) 30

0. In one page

An AI decision record is the evidence that an AI system's decision was checked against a rule, by whom, when, and under which legal provision, kept in a way a third party can verify later without seeing the content that was decided on and without being told who the people were.

This Rulebook says what such a record MUST contain and MUST prove. It is written for five readers: the auditor who needs criteria "different practitioners" can apply the same way, the certification body that needs a schema and a retention period in writing, the data protection officer who must approve a log that can touch employees and customers, the standards committee member who needs clauses already tested and tied to the law (section 9 maps each rule to the law and the audit standards; GOVERNANCE.md sets out the route to a formal standard), and the engineer who has to produce the record from a gateway, a guardrail or an agent runtime.

Five things set it apart from every logging format the editor surveyed (ten IETF individual drafts, sixteen vendor and platform schemas, and the OpenTelemetry, OCSF and OWASP agent-control conventions, surveyed October 2026):

1. **Each decision is bound to a checked legal provision.** The record names the rule that fired, the content digest of the approved rule set it came from, and the provision as a European Legislation Identifier (ELI) web address. "Checked" means the provision is taken from a versioned, public rule set with an approval record, never typed in by the caller.
2. **Content-free, retrievable, pseudonymous.** The record keeps a keyed commitment of each text it decided on and a pointer to where that text is held at its source. Every person is a keyed pseudonym carrying a key version, never a direct identifier. Pseudonymous is not anonymous: a keyed pseudonym is still personal data under Regulation (EU) 2016/679 for anyone who holds the key or other means to link it (EDPB Guidelines 01/2025 on pseudonymisation), and the record is handled as personal data.
3. **Rows are proven, not files.** A sequence number and the previous record's hash are written in the same transaction as the record. The chain head is timestamped by an eIDAS-qualified provider on a bounded cadence. An offline verifier recomputes everything from the export file and reports each class of failure by name.
4. **The decision is recorded as the law sees it.** Beside the engine's verdict, the record carries the decision mode (solely automated, decisive, advisory), the legal basis or GDPR Article 22(2) exception, whether notice was given, and the timeline of any contest.
5. **The judge's kit ships with it.** A clause-by-clause crosswalk to the AI Act, the GDPR, the sector acts and the audit standards, public golden vectors with the expected verifier verdict per failure class, and an interpretation guide, so the record can be read by someone who has never seen the system that wrote it.

What this Rulebook does not do: it does not classify an AI system as high-risk, it does not say which rules an organisation must apply, and it does not replace the logs the AI system itself generates. It says what the record of a decision check must show, and how to prove it was not altered.

1. Scope, terms and keywords

1.1 Scope

This Rulebook applies to a **decision record**: the record written when a text, a request, an output or an action produced or consumed by an AI system is checked against one or more rules and a verdict is reached, by a machine, by a person, or by both.

It covers records written by: a request-path checker (gateway, guardrail, policy engine), an agent runtime recording tool calls and dispositions, a human-oversight interface recording a person's action on the system's proposal, and an export process that packages records for an auditor, a certifier, a supervisory authority or a court.

It does not cover: model training logs, infrastructure logs (access control, capacity, network), or the content of the texts themselves.

1.2 Terms

- **Record.** One decision record, one JSON object conforming to the schema in Annex A.
- **Producer.** The system that writes records. The producer holds the account key.
- **Account.** The tenant or controller on whose behalf records are written. One key per account.
- **Deciding system.** The AI system whose request, output or action was checked. Not the checker.
- **Rule.** A machine-evaluable condition with a stable identifier, a legal provision and an approval record.
- **Rule set.** The versioned collection of rules loaded when a decision was made, identified by a content digest.
- **Provision.** A unit of law at article, paragraph and point level, identified by an ELI web address.
- **Commitment.** A keyed one-way value computed over a text under the account key, so that the text can later be shown to match the record without the record revealing it.
- **Pseudonym.** A keyed one-way value computed over an identifier of a person, so that the same person maps to the same value within one account and to an unrelated value in any other account.
- **Chain.** The per-account sequence of records in which each record carries the hash of the previous one.
- **Anchor.** A qualified electronic timestamp (Regulation (EU) No 910/2014, Article 41(2)) issued over a statement of the chain head.
- **Export.** A file holding records, chain fields, anchors and the verification material.
- **Verifier.** Software that checks an export offline and reports findings.
- **Judge.** Any reader who evaluates records against criteria: an auditor, a certification body, a data protection officer, a supervisory authority, a court.

1.3 Keywords

The key words MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY and OPTIONAL in this document are to be interpreted as described in BCP 14 (RFC 2119, RFC 8174) when, and only when, they appear in all capitals.

1.4 Conformance classes

A producer, an export or a verifier claims conformance to one or more named classes. Each class is a set of rules identified below by its prefix.

Class	Prefix	What it proves
Record	REC	the record carries every required field, content-free
Law	LAW	each decision is bound to a checked provision
Chain	CHN	records are sequenced and linked at write time
Anchor	ANC	the chain head is dated by a qualified third party
Oversight	OVS	human actions on the system's proposal are recorded
Rights	RGT	the fields the person's rights depend on are recorded
Export	EXP	the export can be verified offline and names its own gaps

A conformance claim names the Rulebook version, the classes, the vector set run, the date and the implementation commit (section 10).

2. The record (REC)

2.1 Form

REC-01. A record MUST be a JSON object. Its canonical form for hashing MUST be the JSON Canonicalization Scheme (RFC 8785). The only JSON numbers a record may carry are integers within ± (2^53 - 1); a larger integer, and any non-integer value (a confidence, a rate, an amount), MUST be written as a string, so that every implementation hashes the same bytes.

REC-02. A record MUST carry `rulebook` (the schema URL of the version it conforms to, section 0) and `record_version` (the producer's own schema version). A reader MUST be able to tell from `rulebook` alone which rules to apply. A record keeps the `rulebook` it was written under and is never rewritten to name a later version, so an export under this version MAY carry records that name `https://complyedge.io/rulebook/0.1.0`; each record is checked against the rules of the version it names.

REC-03. A record MUST carry `record_id`, unique within the account. A UUID (RFC 9562) version 4 or 7 is RECOMMENDED.

REC-04. A record MUST carry `occurred_at`, an RFC 3339 timestamp in UTC with at least millisecond precision, taken from a clock synchronised to a declared reference. The reference and its accuracy MUST be stated once, in the export envelope (`clock_source`), not per record. For anomalies and incidents the record MUST also carry `detected_at` when detection and occurrence differ.

REC-05. A record MUST carry `event_type`, one of: `decision`, `oversight`, `contest`, `notice`, `fallback`, `logging_failure`, `opt_out`, `incident`, `classification`, `oversight_evaluation`, `legal_hold`. A producer MAY register further values in its profile (section 10); a reader MUST treat an unknown value as `decision`.

2.2 Actors

REC-06. A record MUST carry `account_id` and `producer_id`. It MUST carry `deploying_party` and SHOULD carry `providing_party` when they differ, as stable identifiers of the organisation roles in Regulation (EU) 2024/1689, Article 3 (deployer, provider).

REC-07. A record MUST carry `deciding_system`, an object with `id` and `version`, identifying the AI system whose request, output or action was checked, as the deployer names it in its own inventory. A record about the account's chain rather than one AI system (a `logging_failure` whose `failure_kind` is `anchor_failed` or `chain_failed`, and a `legal_hold`) has no deciding system and does not carry the field. When the checked system is a model served by a provider, `version` MUST be the model identifier the provider exposes. The checker's own engine and model, if any, go in `checker`, a separate object with `id`, `version` and `path`, and MUST NOT be reported as the deciding system.

REC-08. Every identifier of a natural person in a record (user, reviewer, verifier, approver, data subject reference) MUST be a pseudonym (section 5) and MUST carry the account key version in `key_version`. A role or job title MAY be stored in clear and MUST name a role, never a person. A record MUST NOT carry an email address, a login name, a national identifier or any other direct identifier of a natural person. When the decision affects an identified or identifiable natural person, the record MUST carry `subject`, that person's pseudonym. A decision record that carries `subject` is a *decision about a natural person*; that is the trigger for LAW-07, RGT-01, RGT-02 and RGT-04.

> **REC-09.** When the person whose data was processed is an employee of the deploying party, the record MUST carry `subject_is_employee: true` and MAY carry `works_agreement_ref`, a reference to the agreement under which such records are permitted.

2.3 Content commitments and pointers

> **REC-10.** A record MUST NOT carry the text that was checked, any excerpt of it, any tool-call argument or result, any retrieved document, any system instruction, any prompt variable, or any free-text explanation that reproduces the content. Section 7 lists the attributes of common telemetry conventions that are excluded by name.

> **REC-11.** The record MUST carry `commitments`, an array with one commitment object for each text that was checked: `alg` (section 5), `value`, `key_version`, and `direction` (`input`, `output`, `action`). A record about a request-output pair MUST carry both commitments and a `correlation_id` shared by the input record and the output record of the same exchange.

> **REC-12.** For each commitment the record SHOULD carry `source_ref`, an opaque locator that the account can resolve to the text at its source, so that a person's access request (Regulation (EU) 2016/679, Article 15(1) and (3)), a biometric match record (Regulation (EU) 2024/1689, Article 12(3)(c)) or an authority's request for the logs (Article 21(2)) can be answered. The locator MUST be a URI (RFC 3986) of at most 255 characters with no whitespace, and MUST NOT reveal content.

> **REC-13.** The record MUST carry `data_categories`, the categories of personal data the decision used, as labels (an empty array when it used none), and `special_category: true` when any of them falls under Regulation (EU) 2016/679, Article 9(1). When Regulation (EU) 2024/1689, Article 4a applies (special-category data processed for bias detection), the record MUST carry `art4a_basis_ref`.

2.4 The decision

> **REC-14.** A record of `event_type: decision` MUST carry `verdict`, one of `allow`, `deny`, `modify`, `ask`, `defer`, and `action`, the effect applied (`passed`, `blocked`, `flagged`, `modified`, `escalated`). A producer MUST NOT infer a verdict from a provider's native result; it MUST map it and keep the native value in `native_result`.

> **REC-15.** A record MUST carry `rules_fired`, an array (possibly empty) of rule references (section 3), and `rule_set`, the digest and label of the rule set loaded when the decision was made.

> **REC-16.** A record SHOULD carry `reason_codes`, an array of codes from the rule set's published code list, and MAY carry `confidence`, a decimal string in [0, 1] (REC-01), when a probabilistic component contributed. A reason code is a code: letters, digits and `_ . : / -`, no spaces. A record MUST NOT carry free-text reasoning that reproduces content.

> **REC-17.** A record MUST carry `exemptions`, an object with `claimed` and `rejected` arrays, each entry naming the exemption, its provision reference and, when rejected, a reason code. The assessment that a system is not high-risk (Regulation (EU) 2024/1689, Article 6(3)) is itself recorded as an `event_type: classification` record carrying `classification_result` (`high_risk`, `not_high_risk`) and `assessment_ref`, a reference to the documented assessment (Article 6(4)).

> **REC-18.** When a check was skipped, degraded or replaced by a fallback, the record MUST carry `fallback`, an object with `kind` (`skipped`, `degraded`, `fallback_engine`, `timeout`) and `reason_code`. A check that was skipped as a whole MAY instead be recorded as an `event_type: fallback` record carrying the same

> object. A producer MUST write an `event_type: logging_failure` record, carrying `failure_kind` (`write_failed`, `chain_failed`, `anchor_failed`) and `detected_at`, when it detects that recording failed, and an `event_type: opt_out` record when a lawful user opt-out suppressed recording.

2.5 Retention

> **REC-19.** A record MUST carry `retention`, an object with `class` (a label defined in the producer's profile) and `erase_after` (RFC 3339). Erasure or destruction of records MUST be an authorised and documented process. A legal hold is placed after the records it covers have been chained, so it MUST be recorded as its own record, with `event_type: legal_hold` and `hold` (`id`; `action`, `placed` or `lifted`; `scope`, with `from_seq` and, when the hold has an end, `to_seq`; and MAY carry `reason_code`), and never as a field on the records it covers. A record within the scope of a hold that was placed and not lifted MUST NOT be destroyed.

> **REC-20.** The retention period MUST be at least six months for records of high-risk AI systems within the meaning of Regulation (EU) 2024/1689 (Articles 19(1) and 26(6)), unless Union or national law provides otherwise, and MUST be the period the account's data protection assessment justifies for everything else. The period in force MUST be published in the producer's profile.

3. The rule and the law (LAW)

> LAW-01. Every rule reference in `rules_fired` MUST carry `rule_id`, a stable identifier that is never reassigned to a different provision, and `rule_version`.

> LAW-02. Every rule reference MUST carry `provision`, an object with:

- `eli`: the European Legislation Identifier of the provision as enacted, in the Official Journal form `http://data.europa.eu/eli/<typedoc>/<year>/<number>/art_<N>/par_<P>/pnt_<x>/oj`, using the real `typedoc` (`reg`, `reg_del`, `reg_impl`, `dir`, `dir_del`, `dir_impl`, `dec`), the subdivision tokens `art_`, `par_`, `pnt_` as far as the provision goes, and `anx_<roman>` in lower-case Roman numerals for an annex (`anx_iii`). The ELI is an identifier: the resolver returns the whole act and anchors only to paragraph and annex level, so a reader MUST NOT rely on it as a deep link.
- `consolidated_eli` (RECOMMENDED): the ELI of the act as amended on the date the rule set was approved, at act level, in the form `http://data.europa.eu/eli/<typedoc>/<year>/<number>/<YYYY-MM-DD>`, with no subdivision and no `/oj`. The two fields answer two questions: what was enacted, and what the law said when the rule was approved.
- `label`: the human-readable form (for example `Regulation (EU) 2024/1689, Article 5(1)(c)`).

For a provision in national law the `eli` MUST be the national ELI where the Member State publishes one at the level cited, and otherwise `eli` is omitted and `url` carries the official publication address of the provision (for example a section page of the German federal law portal, since no German ELI resolver exists at the time of writing; Spain and France publish ELIs at act level only). `label` MUST name the instrument and provision in the Member State's own form.

> LAW-03. Every rule reference MUST carry `citation`, the text of the provision as quoted or closely paraphrased in the rule set, so that a reader without network access can read what the rule enforced.

> LAW-04. `rule_set` MUST carry `digest` (SHA-256 over the canonical content of every rule file loaded, excluding tests), `label` (the human version), and `index_url`, where each `rule_id` resolves to its rule page. A rule page MUST show the provision, the citation, the condition, the approval status and the name of the approver for an approved rule.

> LAW-05. A provision in a record is **checked** only when the rule set it came from is public, versioned, content-digested and carries a per-rule approval record. A producer MUST NOT accept a provision supplied by the caller at decision time as a checked provision; it MAY record it separately as `caller_asserted_provision` and a reader MUST NOT treat it as checked.

> LAW-06. When no rule fired, `rules_fired` is empty and the record MUST still carry `rule_set`, so the reader knows which rules were evaluated and found no violation.

> LAW-07. For each decision about a natural person the record MUST carry `legal_basis`, an object naming the basis of the processing (`provision`, as in LAW-02) and, where Regulation (EU) 2016/679, Article 22 applies, the exception in Article 22(2)(a), (b) or (c) relied on.

> LAW-08. The record MUST carry `purpose_ref`, a reference to the intended purpose of the deciding system as documented by the deployer, and, when the system is listed in Annex III of Regulation (EU) 2024/1689, `annex_iii_point`.

4. Chain and anchor (CHN, ANC)

4.1 Write-time chain

CHN-01. Each record MUST carry `chain`, an object with `seq` (a positive integer, dense per account, starting at 1), `prev` (the `hash` of the record with `seq - 1`, or the empty string for `seq` 1), `hash`, and `alg`.

CHN-02. `chain.hash` MUST be computed as `SHA-256( prev || canonical(record minus the chain object) )`, where `prev` is the previous hash as its lower-case hexadecimal string in UTF-8 (empty for `seq` 1), `canonical` is RFC 8785 over the record exactly as it will be exported, and `||` is byte concatenation. The hash is written as 64 lower-case hexadecimal characters. `chain.alg` MUST name this construction (`sha256-chain-1` for this Rulebook version).

CHN-03. `seq`, `prev` and `hash` MUST be assigned by the single writer in the same transaction that persists the record, against a per-account head that carries the newest `seq` and `hash`. A record that cannot be chained MUST NOT be written as unchained; the failure MUST be recorded per REC-18 (`failure_kind: chain_failed`) once writing is possible again.

CHN-04. A record MUST NOT be backdated: `occurred_at` of a record with `seq` N MUST NOT be earlier than `occurred_at` of the record with `seq` N-1 by more than the clock tolerance declared in the profile.

CHN-05. A producer that also signs records or chain heads SHOULD do so with a signature the account can verify independently of the producer, and MUST declare the algorithm and key reference in the export envelope. An origin signature is RECOMMENDED for conformance to class Chain. A producer that claims the presumption of integrity and correct origin of Regulation (EU) No 910/2014, Article 35(2) MUST apply a qualified electronic seal, since that presumption attaches only to qualified electronic seals. The presumption of Article 45k(2) attaches only to qualified electronic ledgers; a producer's own chain is not one, and this Rulebook claims no presumption for it.

4.2 Qualified anchor

ANC-01. On a bounded cadence declared in the profile, and at least once per day on which the head moved, the producer MUST obtain a qualified electronic timestamp (Regulation (EU) No 910/2014, Article 41(2); Implementing Regulation (EU) 2025/1929) over the SHA-256 digest of the **anchor statement**: the canonical JSON `{"account_id": ..., "chain_alg": ..., "chain_hash": ..., "chain_seq": N}`. The statement MUST carry no record content, no commitment and no pseudonym.

ANC-02. The timestamp token MUST be an RFC 3161 TimeStampToken (with RFC 5816 certificate identifiers), issued by a provider listed as qualified for time stamping on an EU Member State trusted list at the time of issue, with declared accuracy of one second or better to UTC (ETSI EN 319 421, TIS-7.7.2-03). The export carries it in `token_b64` as the provider returned it: the TimeStampResp, or the TimeStampToken alone. The bytes MUST NOT be re-encoded or re-ordered before storage or export. A verifier checks the token in this order and reports the first step that fails (EXP-04): *parse* (the bytes are a TimeStampResp or TimeStampToken whose content is a TSTInfo); *status* (the response was granted, the signing certificate was valid at the token's genTime and is for time stamping, and any declared accuracy is one second or better); *imprint* (the message imprint is the SHA-256 digest of the anchor statement); *signature* (the CMS signature verifies with the signing certificate, at genTime); *trust* (the signing certificate chains, at genTime, to a trust anchor the reader pinned from a trusted list). A verifier that cannot run a step reports that it did not check it, never that the token failed.

> **ANC-03.** The producer MUST store the statement, the token, the provider's URL and the time of storage in a write-once store with a retention at least as long as the longest retention of the records the anchor covers, and MUST keep the provider's certificate chain so the token can be verified after the provider's certificate expires.

> **ANC-04.** For retention periods beyond the validity of the signing algorithm or certificate, the producer SHOULD augment the evidence with a new qualified timestamp over the old token and the statement (ETSI TS 119 511, section 7.15; evidence records per RFC 4998 or RFC 6283).

> **ANC-05.** A failed anchor MUST NOT block decisions or writes. It MUST be recorded as an `event_type: logging_failure` record with `failure_kind: anchor_failed`, alarmed, and retried on the next cadence.

5. Commitments and pseudonyms

CMT-01. A commitment MUST be a keyed one-way function under the account key over the commitment domain tag (CMT-02) followed by the UTF-8 bytes of the text. HMAC-SHA-256 is REQUIRED for conformance to this version (`alg: hmac-sha256-1`). An unkeyed hash, or a hash salted with a value stored beside it, MUST NOT be presented as a commitment.

CMT-02. A pseudonym MUST be a keyed one-way function over the canonical form of the identifier under the account key, with a domain separator that differs from the commitment domain, so that a pseudonym can never be mistaken for a commitment of the same bytes. Both inputs MUST carry a domain tag, neither tag a prefix of the other, and the producer MUST state both tags in its profile (`commitment_domain`, `pseudonym_domain`). The prefix `p1:` and 32 hexadecimal characters is the form used by this version.

CMT-03. The account key MUST have at least 256 bits of entropy, MUST be stored only in encrypted form at rest, MUST be decrypted only in memory bound to the account, and MUST NOT be returned by any interface. The producer MUST log every execution of the pseudonymisation function's reverse mapping, if one exists, and SHOULD rate-limit it.

CMT-04. Every commitment and every pseudonym MUST carry `key_version`. Rotation MUST be performed by applying a second keyed one-way function to the existing values, never by reconstituting the original text or identifier.

CMT-05. Destroying the account key MUST render every commitment and pseudonym under it unlinkable to any text or person. The producer MUST record the destruction (`key_shredded_at`) on the account, never on a record, and MUST disclose in its profile every copy of records and keys and the time after which a destruction is complete across those copies.

CMT-06. A producer MUST offer a verification interface that takes a text and a `record_id` and answers whether the text matches the commitment, without storing or logging the text.

6. Oversight, rights and notices (OVS, RGT)

6.1 Human oversight

OVS-01. A human action on the output of an AI system MUST be recorded as an `event_type: oversight` record carrying `proposal_ref` (the `record_id` of the decision record holding the system's proposal), `oversight_action`, one of `accepted`, `disregarded`, `overridden`, `reversed`, `not_used` (the person decided without the system), `intervened`, `stopped` (the system was brought to a safe state), `verified`, `reviewer` (a pseudonym), `reviewer_role`, `authority_ref` (the reference under which the reviewer holds the competence, training and authority to change the decision), `review_opened_at`, `occurred_at`, `inputs_ref` (a reference to what the reviewer could see, including anything the affected person submitted), `reason_code` (from the published code list, never prose), and `outcome_ref` (the record of the final decision when it differs from the proposal; REQUIRED when `oversight_action` is `overridden` or `reversed`). The action names mirror the four intervention functions of Regulation (EU) 2024/1689, Article 14(4)(d) and (e): disregard, override, reverse, and interrupt to a safe state.

OVS-02. The system's proposal and the human's final action MUST be two records linked by `proposal_ref`, never one record overwritten: a decision record MUST NOT carry `proposal_ref`, `oversight_action`, `reviewer`, `review_opened_at` or `outcome_ref`, and `proposal_ref` MUST name a decision record. A reader MUST be able to reconstruct the pair (what the system proposed, what the person finally did) from the two records without any content. When the proposal is not in the export, the verifier reports that the pair could not be checked.

OVS-03. Where Regulation (EU) 2024/1689, Article 14(5) applies, the decision record holding the identification carries `annex_iii_point` beginning `1(a)`, and a decision record that acts on it carries `identification_ref`, the `record_id` of the identification. Before any such acting record with `action` other than `passed` is written, two `oversight` records with `oversight_action: verified`, `proposal_ref` naming the identification and distinct `reviewer` pseudonyms MUST precede it in the chain.

OVS-04. Where a decision category is reserved to a human by law (for example Directive (EU) 2024/2831, Article 10(5)), the decision record MUST carry `decision_mode: human` and `human_decider`, a pseudonym with `role` and `authority_ref`.

OVS-05. Per-reviewer override rates are oversight-evaluation data, not performance data. An `event_type: oversight_evaluation` record carries `period`, counts per `oversight_action` aggregated across reviewers, `evaluator_role` and `representatives_informed: true|false` (Directive (EU) 2024/2831, Article 10(1) and (4)). Per-reviewer counts, when kept, are carried in `reviewer_counts` (an array of `reviewer` and `action_counts`), and `action_counts` keys MUST be `oversight_action` values. An export whose envelope `export_purpose` is not `oversight_evaluation` MUST NOT carry `reviewer_counts`, and a producer MUST NOT expose per-reviewer rates through any interface other than the evaluation role's.

6.2 Decision mode and rights

RGT-01. Every decision record about a natural person MUST carry `decision_mode`, one of `solely_automated`, `decisive` (a human relies strongly on the output), `advisory`, `human`. The value MUST be set by the deployer's documented process, not inferred by the checker.

RGT-02. Every decision record about a natural person MUST carry `notice`, an object with `given` (boolean), `given_at` and `notice_ref`, recording that the person was informed that they are subject to the use of the AI system (Regulation (EU) 2024/1689, Article 26(11); Directive (EU) 2023/2225, Article 18(9), where it applies; Directive (EU) 2024/2831, Article 9(1) and (3)), and, for employees, `workers_informed_at` (Article 26(7)).

RGT-03. Any request by the person for human intervention, to express their view, for an explanation, to contest the decision, or to correct personal data the decision used MUST be recorded as an `event_type: contest` record carrying `decision_ref`, `request_kind` (`human_intervention`, `express_view`, `explanation`, `contest`, `review`, `correction`) and `requested_at`. `correction` is a request to correct inaccurate personal data used in the decision (Regulation (EU) 2016/679, Article 16); the record carries neither the data nor the correction, only the request and its answer. Records are never overwritten, so the answer is a further `contest` record for the same `decision_ref` that also carries `responded_at`, `reviewer` (a pseudonym), `review_outcome` (`upheld`, `rectified`, `withdrawn`), `statement_ref` (a reference to the written statement of reasons, never its text) and, when the outcome is `rectified`, `rectified_at`.

RGT-04. Every decision record about a natural person MUST carry `explanation_inputs`, an object captured at decision time, under the pinned `deciding_system.version`, that lets a later explanation be produced without the content: `logic_ref` (the rule set digest and the rule ids, or the model identifier, version and configuration digest), `procedure_ref` (the versioned description of the procedure and principles applied, held outside the record), `main_factors` (an ordered array of factor codes with an importance bucket, never values), `data_categories` (REC-13), `risks_ref` (the documented statement of the risks, significance and effects of the processing for this system version), `main_elements` (outcome code, effect code, effective date), and, where the system can compute it, `sensitivity` (per factor, the direction and the smallest change bucket that would have changed the result). The inputs MUST be captured at decision time because the explanation owed is of the result actually produced, not of a later recomputation.

RGT-05. When information that would normally be disclosed to the person is withheld on trade-secret or third-party grounds, the record MUST carry `withheld: true` and `escrow_ref`, a reference under which the withheld information can be produced to a supervisory authority or a court.

RGT-06. A record of `event_type: incident` MUST carry `incident_kind` (`risk`, `substantial_modification`, `serious_incident`, `suspension`), `detected_at`, `reported_at` and `report_ref`, and the deciding system's `id` and `version` frozen at the time of the incident.

RGT-07. The issue of an explanation, a statement of reasons or a rights notice to the person MUST be recorded as an `event_type: notice` record with `notice_kind` (`use_of_ai`, `rights`, `explanation`, `statement_of_reasons`, `rejection`), `decision_ref`, `channel_ref`, `language`, `issued_at` and `statement_ref` (a reference, never the text).

7. What is excluded by name

A producer that also emits telemetry under the OpenTelemetry GenAI semantic conventions or the OCSF AI operation profile MUST NOT copy the following attributes into a record, because they carry content:

- OpenTelemetry: `gen_ai.input.messages`, `gen_ai.output.messages`, `gen_ai.system_instructions`, `gen_ai.tool.call.arguments`, `gen_ai.tool.call.result`, `gen_ai.retrieval.documents`, `gen_ai.retrieval.query.text`, `gen_ai.memory.query.text`, `gen_ai.memory.records`, `gen_ai.evaluation.explanation`, `gen_ai.prompt.variable`, `gen_ai.guardrail.content.input.value`, `gen_ai.guardrail.content.output.value`.
- OCSF: `message_context.prompt_text`, `message_context.response_text`.
- OpenInference: `input.value`, `output.value`, `message.content`, `document.content`.

A producer MAY carry a commitment (section 5) in place of any of them.

The following MAY be carried because they do not reveal content: `gen_ai.conversation.id`, `gen_ai.response.id`, `gen_ai.agent.id`, `gen_ai.agent.version`, `gen_ai.request.model`, `gen_ai.response.model`, `gen_ai.guardrail.policy.id`, `gen_ai.guardrail.policy.version`, `gen_ai.guardrail.policy.rule.id`, OCSF `ai_model`, `ai_agent`, `delegation.uid`, `ai_status.permission_result`, and any `*_fingerprint` attribute.

8. Export and verification (EXP)

EXP-01. An export MUST be one JSON document with an `envelope` and a `records` array. The envelope MUST carry: `rulebook`, `account_id`, `producer_id`, `period` (`from`, `to`), `clock_source` (`reference`, `tolerance_ms`), `chain_summary` (`alg`, `first_seq`, `last_seq`, `chained_rows`, `pre_chain_rows`), `anchors` (each: the statement, the token in base64, the provider URL, the time stored), `anchor_coverage` (`newest_anchored_seq`, `unanchored_rows`), `retention_profile_url`, `signatures` (when CHN-05 applies; each with `alg`, `key_ref` and `value`), and `interpretation_guide_url`, and MAY carry `export_purpose`. `chain_summary` and `anchor_coverage` MUST state what the records and anchors in the export actually give, and every record's `account_id` MUST be the envelope's.

EXP-02. The export MUST label every record written before the chain existed with `chain_status: pre_chain` and MUST NOT present such a record as verified.

EXP-03. The verifier MUST work offline from the export, the hash and canonicalisation identifiers, the anchor tokens, the timestamp providers' certificate chains and a pinned trusted-list root or its retrieval date. It MUST NOT need the producer, the account key or network access.

EXP-04. The verifier MUST report a verdict (`verified`, `failed`, `unverified`, or `refused` when a required pinned input such as the trust anchors was not supplied and nothing was evaluated) and a findings array, one finding per occurrence, each with `finding`, `seq`, `record_id` and `detail`. `unverified` means every finding is informational (`pre_chain`, `unanchored_tail`, `no_anchor`, `reference_unresolved`) or a step the verifier could not run (`anchor_token_not_checked`); any other finding makes the verdict `failed`. The finding names in this version are:

Finding	Meaning
`hash_mismatch`	a record's stored hash is not the hash of its content and `prev` (an edit)
`sequence_gap`	`seq` N present, N+1 absent, N+2 present (a removed record)
`sequence_duplicate`	two records share a `seq` (a forged insertion)
`link_break`	`prev` is not the previous record's hash (reorder, insertion or rewrite)
`unchained_after_chain_start`	a record without chain fields dated after the first chained record
`pre_chain`	a record written before the chain existed; labelled, never verified
`unanchored_tail`	records newer than the newest anchor; not a failure, not yet dated by a third party
`truncated_tail`	the newest anchor attests a `seq` the export does not reach (records removed after the anchor)
`no_anchor`	the export carries no anchor
`anchor_mismatch`	an anchor's hash is not the export's hash at that `seq`
`anchor_token_parse`, `anchor_token_status`, `anchor_token_imprint`, `anchor_token_signature`, `anchor_token_trust`	the timestamp token fails at that step (ANC-02)
`anchor_token_not_checked`	the verifier could not run the signature or trust step (for example, no usable OpenSSL); not a failure, never `verified`
`backdated`	CHN-04 violated
`content_present`	a record carries an attribute excluded by section 7

Finding	Meaning
`person_in_clear`	a record carries a direct identifier of a natural person
`provision_unchecked`	a rule reference lacks `rule_set` digest, `eli` or approval, or a `caller_asserted_provision` was presented as checked
`required_field_missing`	a REQUIRED field of a claimed class is absent
`invalid_value`	a field holds a value outside its closed set or its form (a malformed timestamp, a floating-point number, an unkeyed commitment, a free-text reason code), a value contradicts the export (an `account_id` or summary that differs, a duplicate `record_id`, a `rule_id` naming two provisions), or a decision record carries oversight fields
`reference_unresolved`	a `proposal_ref`, `outcome_ref`, `decision_ref` or `identification_ref` names a record that is not in the export; the link cannot be checked
`dual_verification_missing`	OVS-03 violated
`reviewer_rates_exposed`	OVS-05 violated
`profile_mismatch`	the export disagrees with the producer's profile: a `producer_id` or `clock_source` that differs, a retention class the profile does not define, an `event_type` or reason code it does not list (PRF-01)

> **EXP-05.** An export for a supervisory authority or a court MAY add the texts behind named commitments as a separate, access-controlled attachment; the export itself MUST stay content-free, and the attachment MUST be verifiable against the commitments with the verification interface (CMT-06).

9. Reading the record: the judge's crosswalk

Each row says which clause a field rests on and which audit criterion it answers. The crosswalk is normative for the mapping and informative for the law: where a reader's own reading of a clause differs, the clause governs.

9.1 Clause crosswalk

Field or rule	Regulation (EU) 2024/1689 (AI Act)	Regulation (EU) 2016/679 (GDPR) and case law	Sector and national law	Standards and judges
record exists per event (REC-01..05)	Art 12(1), 12(2)(a)-(c); Recital 71	Art 5(2)	DORA RTS 2024/1774 Art 12(2)(c); NIS2 IR 2024/2690 Annex 3.2.3	NIST SP 800-53 AU-2, AU-12; ISO/IEC 42001 A.6.2.8
`occurred_at`, `detected_at`, clock source (REC-04)	Art 12(3)(a) (start and end of each use; remote biometric identification only)		DORA RTS Art 12(2)(f), 23(4)(a)-(b); NIS2 IR 3.2.6; RTS 25 (Delegated Regulation (EU) 2017/574) Arts 1-4	NIST AU-8; ETSI EN 319 421 TIS-7.7.2-03; ISO/IEC 27001 A.8.17
`deciding_system` (REC-07)	Art 12(2)(a), 73(6), Annex IV 1(a), 1(c)		PLD (Directive (EU) 2024/2853) Arts 9-10	BSI AIC4 RE-02; BaFin guidance 18.12.2025 (Modellversionen)
pseudonyms, `key_version` (REC-08, section 5)		Art 25(1), 32(1)(a); EDPB Guidelines 01/2025 paras 88-91, 107	BetrVG §87(1) Nr. 6 (DE)	CNIL délibération 2021-122 (minimise personal data in logs)
`subject_is_employee` (REC-09)	Art 26(7)		BetrVG §87(1) Nr. 6; BAG 1 ABR 7/15	
commitments (REC-10..11, section 5)	Art 12(3)(c) (biometric)	Art 5(1)(c), 25, 32; EDPB 01/2025 fn 39; EDPB 02/2025 §3.2		IIA AI Auditing Framework (prompts and responses)
`source_ref` (REC-12)	Art 12(3)(c), 21(2)	Art 15(1), 15(3); CJEU C-203/22 para 63	MiFID II Art 16(7) (communications held at source)	ISO/IEC 27037 (acquisition, chain of custody)
`data_categories`, `special_category`, `art4a_basis_ref` (REC-13)	Art 4a (as amended by Reg (EU) 2026/1744)	Art 15(1)(b), 22(4), 30(1)(c)	PWD 2024/2831 Art 7	
`verdict`, `action`, `native_result` (REC-14)				ISAE 3402 para 28 (deviations); OWASP ACS §6.1 dispositions
`rules_fired`, `rule_set` (REC-15, LAW-01..06)	Art 13(3)(b)(iv), 86(1)	Art 15(1)(h); C-203/22 operative point 1 ("procedure and principles actually applied")		ISAE 3402 para 25(a) (by whom or by what means); IDW PS 861 (only approved changes reach production); ISAE 3000 A45-A46 (suitable criteria)
`provision.eli` (LAW-02)				Council conclusions on the ELI (2012/C 325/02 and successors)
`exemptions`, `classification` record (REC-17)	Art 6(3); Commission draft guidelines on high-risk classification (19 May 2026) para 116	Art 22(2)		

Field or rule	Regulation (EU) 2024/1689 (AI Act)	Regulation (EU) 2016/679 (GDPR) and case law	Sector and national law	Standards and judges
`fallback`, `logging_failure`, `opt_out` (REC-18)			DORA RTS Art 12(2)(e); NIS2 IR 3.2.3(k); CRA 2024/2847 Annex I Part I (2)(l)	TÜV AUSTRIA Trusted AI catalogue (fallback activations); ISAE 3402 para 28
`retention` (REC-19..20)	Art 19(1), 26(6)	Art 5(1)(e), 17(3)(b), 17(3)(e), 30(1)(f)	MiFID II Art 16(7) (5-7 years); RTS 6 (Delegated Regulation (EU) 2017/589) Art 28(3) (5 years, high-frequency trading order records); DORA RTS Art 12(2)(a)	AESIA Guía 12 §5.3 (destruction authorised and documented; legal hold); BSI AIC4 RE-02 (3 months minimum); NIST AU-11
`legal_basis` (LAW-07)		Art 22(2)(a)-(c), 5(2)		
`purpose_ref`, `annex_iii_point` (LAW-08)	Art 12(2), 26(11), 86(1), Annex III	Art 30(1)(b)		
chain (CHN-01..04)		Art 5(1)(f), 32(1)(b)	DORA RTS Art 12(2)(d), 23(3); NIS2 IR 3.2.5; eIDAS Art 45k(2) (presumption for qualified electronic ledgers only; a producer's own chain is not one)	NIST AU-9(3); ISAE 3000 A148, A150; BaFin (lückenlos, gegen Manipulation geschützt); ISO 15489-1 (integrity)
origin seal or signature (CHN-05)			eIDAS Art 35(2) (qualified electronic seal); Art 45l(1)(b) and IR 2025/2531 REQ-7.5-03 (requirements for qualified electronic ledgers)	NIST AU-10; IDW PS 861 (Authentizität, Verbindlichkeit)
anchor (ANC-01..05)			eIDAS Art 41(2); IR 2025/1929; ETSI EN 319 421, EN 319 422	ISAE 3000 A149 (consistency across sources); ETSI TS 119 511 §7.15, §9.2
oversight records (OVS-01..04)	Art 12(3)(d), 14(4)(d)-(e), 14(5), 26(2)	Art 22(3); WP251rev.01 p.21 (meaningful, authority and competence)	PWD Art 10(2), 10(5)	AESIA Guía 06 (register the proposal and the final action); BSI AIC4 RE-04 (authorised subjects correct output)
`oversight_evaluation` records (OVS-05)	Art 14(4)(b) (automation bias), 26(2)	WP251rev.01 p.21 (record the degree of human involvement)	PWD Art 10(1), 10(4); Dutch DPA guidance on meaningful human intervention (2025) p.18-19	
`notice` records with `notice_kind` (RGT-07)	Art 86(1)	Art 12(1), 13-14	CCD Art 18(8)-(9); PWD Art 11(1)	
`decision_mode` (RGT-01)	Art 86(1) (role of the AI system)	Art 22(1); CJEU C-634/21 (draws strongly)	CCD Art 18(8)-(9); PWD Art 9(1)(b)(ii)	
`notice` (RGT-02)	Art 26(7), 26(11)	Art 13-14	CCD Art 18(9) (on rejection after an automated assessment), Art 13 (personalised offers); PWD Art 9(1), 9(3)	
`contest` records (RGT-03)		Art 16 (`correction`), 22(3); C-634/21 para 66	CCD Art 18(8)(c); PWD Art 11(2)-(3) (two weeks); Colorado SB 26-189, C.R.S. 6-1-1705(1)(a) (correction, human review and reconsideration)	
`explanation_inputs` (RGT-04)	Art 86(1), 13(3)(b)(iv)	Art 15(1)(h); C-203/22 paras 59, 61-62	CCD Recital 56 (main variables, the logic and risks); PWD Art 9(1)(b)(iii)	

Field or rule	Regulation (EU) 2024/1689 (AI Act)	Regulation (EU) 2016/679 (GDPR) and case law	Sector and national law	Standards and judges
`withheld`, `escrow_ref` (RGT-05)		Art 15(4); C-203/22 operative point 2		
incident records (RGT-06)	Art 12(2)(a), 26(5), 73, Art 3(49)		DORA RTS Art 23(4)	NIST AI RMF MANAGE 4.3
exclusions (section 7)		Art 5(1)(c)		EDPB Support Pool LLM report (minimise logging)
export, verifier (EXP-01..05)	Art 21(2), 13(3)(f)	Art 15(3)	KI-MIG §11(2) (DE, remote access by API)	AESIA Guías 13, 15 (format, retention, how to interpret the logs); BSI AIC4 BC-04; RFC 9943 §7.1 (verifiable without online access)

9.2 Properties judges test

Property	Who asks	Rules that deliver it
contemporaneous	ISAE 3000 A148	CHN-03, CHN-04
gap-free over the period	ISAE 3000 A150; BaFin	CHN-01, EXP-04 (`sequence_gap`, `truncated_tail`)
tamper-evident at write time	BaFin; DORA RTS; CNIL	CHN-02, EXP-04
dated by a third party	eIDAS Art 41(2); ISAE A149	ANC-01..03
model-pinned	BSI AIC4 RE-02; BaFin	REC-07
failures included	ISAE 3402 para 28; TÜV AUSTRIA	REC-18
schema and retention published	BSI AIC4 BC-04; AESIA Guías 13, 15; AI Act Art 13(3)(f)	REC-02, REC-19..20, section 10
proposal and final action linked	AESIA Guía 06; WP251	OVS-01..02
criteria usable by different practitioners	ISAE 3000 A45-A46	every MUST; vector coverage per MUST in `conformance/COVERAGE.md`
verifiable without content or key	GDPR Art 5(1)(c)	sections 5, 7, EXP-03

9.3 Procurement and attestation sentences the classes answer

Public buyers and auditors write their logging requirement in a sentence, not a schema. Each row names the sentence as published and the class that answers it.

Sentence (as published)	Source	Answered by
"technically allow for the automatic recording of events ('logs') over the lifetime of the AI System"; traceability "appropriate to the Intended Purpose"; events that may "result in the AI System presenting a risk … or lead to a Substantial Modification"; logs kept "for the duration of the Agreement" then handed over; optional "access to the logs … on a real time basis"; mechanisms to "collect, store and interpret the logs"	European Commission, Model Contractual Clauses for AI (high-risk), Article 5 and Annex C	Record (REC-01..05, REC-18), Rights (RGT-06), Export (EXP-01..04), the profile (PRF-01)
"Protokollierung, Auditierung sowie … Change- und Exit-Prozesse"	Federal AI platform notice, Germany (TED 688455-2025)	Record, Chain, Export
"Governance mit Zugriffsverwaltung, Modellauswahl, Logging, Reporting"	State AI platform notice, Saxony (January 2026)	Record (REC-07), Export

Sentence (as published)	Source	Answered by
"type of request, processing times including time stamps and metadata on the user requesting the AI service", kept "at least three months", described "in the system description"	BSI AIC4, criterion RE-02 and BC-04	Record (REC-04, REC-08, REC-19), the profile (PRF-01)
`trace_id`, `span_id`, `start_time`, `end_time`, `processing_activity_id`, an encrypted `data_subject_id` with its type; pseudonymisation recommended; no employee identity in the log line	Logius, Logboek Dataverwerkingen 1.0 (Netherlands)	Record (REC-03, REC-04, REC-08, REC-11 `correlation_id`), Law (LAW-08 `purpose_ref`)
logs secured against "sletning, modifikation eller fabrikation" so they can serve as "bevis ved domstolene"; exact timestamps from a stratum-2 source; separable fields with correlation keys	Danish Agency for Digital Government, eID gateway logging policy 1.1	Chain, Anchor, Record (REC-04)
"hvornår (datostempel), hvem (aktør) har gjort hvad (hændelse), og i givet fald på vegne af hvilken part"	Danish Agency for Digital Government, Digital Post log data rules 1.0 (2026)	Record (REC-04, REC-06..08, REC-14)
"No logging of fallback activations" as a typical audit finding	TÜV AUSTRIA Trusted AI audit catalogue (2025)	Record (REC-18)

Under ISAE 3000 and the audit standards built on it (IDW PS 861, BSI AIC4), the audited organisation selects the criteria and names them in its system description; the auditor opines on their suitability. Attestations cite the criteria catalogue by its official title and version. This Rulebook is written to be named that way: a system description cites `ADCR <version>, classes <list>` and the profile URL, and the auditor tests the records against the vectors.

10. Profiles, vectors and conformance claims

> PRF-01. A producer MUST publish a **profile** at a stable URL, carrying: the Rulebook version and classes claimed; its `producer_id`; `clock_source` and tolerance; the anchor cadence and the timestamp providers used; each retention class with its period in days and whether it holds records of high-risk AI systems (REC-20); every copy of records and keys and how many days after a key destruction each copy is gone (CMT-05); the `event_type` values and `reason_codes` it emits; the `rule_set` index URL; the interpretation guide; the commitment and pseudonym domain tags (CMT-02); and the URL of the verification interface (CMT-06). The profile MUST be a JSON document that validates against `schema/producer-profile.schema.json`. Given the profile, the verifier checks its fields and checks the export against it (finding `profile_mismatch`); what the profile states about the producer's processes remains the producer's attestation. A profile is the document AESIA Guía 13 and BSI AIC4 BC-04 ask for.

> PRF-02. The Rulebook publishes **test vectors**: for each conformance class, a set of exports with the expected verifier verdict and findings. The aim is at least one vector per finding name in EXP-04 and, per MUST, one positive and at least one negative case. In this version every finding name that describes an export has a vector (`anchor_token_not_checked` describes the verifier's environment instead), and 57 of the 62 rules that carry a MUST have a positive and a negative case; the other five are, wholly or in the part a MUST covers, about how the producer works or how a verifier runs, which neither an export nor a profile can show. `conformance/COVERAGE.md` lists every rule that carries a MUST with the cases that exercise it, or says that it is not yet vectored or is attested by the producer in its profile. A MUST without a vector still binds; the missing vector is a gap to close. A vector set is versioned with the Rulebook. A verifier claims conformance by passing every vector of the classes it claims.

The vector layout is `conformance/vectors/<suite>_v<N>/` holding `cases/<case>.json` (one export per case), `profiles/<case>.json` where a case checks an export against a producer profile, and `expected.json` (per case: `verdict`, `findings[]`, the trust anchors to pin and, where one applies, the profile), beside a shared `conformance/vectors/keys/` (the test timestamp authority's certificates, the pinned test trust anchor and the minted tokens; its private keys are not published), a `README.md` saying what each case proves and how to regenerate it, and a checker that imports only public libraries. Case names carry one failure axis each (`chain_gap_mid`, `chain_tail_truncated`, `anchor_token_untrusted_root`); axes combine only in an `interactions` suite. The verdict is a closed set: `verified`, `failed`, `unverified`, `refused` (a required pinned input, such as the trusted-list root, was not supplied). A runner emits per-suite pass, fail and skip counts in JSON and exits non-zero on any failure. Vectors are licensed Apache-2.0 so that a verifier embedded in a proprietary product can carry them.

> PRF-03. A **conformance claim** is a statement in the form: `Conforms to ADCR <version>, classes <list>, vector set <version>, run on <date>, implementation <identifier>`. The claim MUST be reproducible by a third party from the public vectors and the producer's profile.

> PRF-04. A version of this Rulebook is marked **stable** only when two conditions hold: one implementation by people other than the editor has passed the vectors for the Record, Law and Chain classes, and the vectors for every finding name in EXP-04 that describes an export exist. The second condition is met in this version; the first is not.

> PRF-05. Changes follow semantic versioning. A change that makes a previously conformant record non-conformant, removes a field or finding name, or changes a hash construction, is a major version. Field names in this Rulebook are chosen to be neutral and are not aligned to any draft standard whose text is

> not yet public; alignment to a published international standard will be offered as a mapping annex, not as a rename.

11. What this version does not settle

- Whether a qualified timestamp over the chain head extends the presumption of Regulation (EU) No 910/2014, Article 41(2) to every record beneath it, or only to the head value, is a question for counsel; this Rulebook claims only that the statement was timestamped.
- Whether destroying the account key satisfies Regulation (EU) 2016/679, Article 17, for records that remain, is for the controller's assessment with its supervisory authority; this Rulebook claims unlinkability as described by the EDPB (Guidelines 02/2025, section 3.2) with the caveats stated there.
- Field names are not aligned to ISO/IEC 24970 or prEN 18229-1, whose texts are not public at the time of writing. A mapping annex will follow publication.
- The Rulebook cites paid standards by title only.
- The name. "ADCR" was chosen in October 2026 after a collision check (web search, DNS, GitHub repository names) of nine candidates. It says what the record is: a record that an AI decision was checked against a rule. The rejected candidates collide inside this field or in EU legal vocabulary: "OADR" is OpenADR, an IEC standard and a registered mark held by a standards alliance; "ADER" is a foundation and an active machine-learning toolbox; "AIDR" is a humanitarian AI project; "ODR" and "ODRF" read as Online Dispute Resolution to EU lawyers; "ADRS" reads as the plural of architecture decision records to every engineer and names an AI research-systems project; "ADRP" is a US Army doctrine series; "VADR" is a bioinformatics tool and a US government assessment tool. ADCR's known collisions are an advertising-industry club in Russia and electronic part-number prefixes, both outside law, audit and AI. The EUIPO trade mark register was not reachable from the tooling used; a manual search for "ADCR" and "**ComplyEdge**" in classes 9, 42 and 45 is open. Judges cite a title and version, not a domain, so the Rulebook is served under the editor's own domain.
- Dates. Regulation (EU) 2026/1744 left Articles 12 and 19 of the AI Act unchanged, deferred the application of the Annex III high-risk obligations, and replaced the Article 72(3) template empowerment with Commission guidance and a voluntary template. This Rulebook therefore cites no Commission post-market monitoring template, and nothing in it is keyed to an application date: the record is the same whether the duty binds today or later. ISO/IEC 24970 (at final draft ballot) and prEN 18229-1 (enquiry draft under comment resolution) are cited as drafts with their stage, not as published standards.
- Peer formats surveyed in October 2026 (the ten IETF individual drafts, the sixteen vendor and platform schemas, and the OpenTelemetry generative-AI, OCSF 1.9.0 and OWASP Agent Control Standard 0.1.3 conventions of section 0) map the AI Act's Article 12, and none binds a decision to a checked provision or carries an ELI. Some carry the law in another form: one draft has a per-decision list of AI Act articles that the producer declares and that verifiers are told not to treat as checked, another has article flags, and a third has a "canonical URI of the law article". LAW-02 and LAW-05 are a differentiator of this Rulebook, not a point of convergence, and the mapping annex will say so.
- No Commission or EDPB guidance on Regulation (EU) 2024/1689, Article 86 existed at the time of writing. RGT-04 is built from the Court's reading of the GDPR explanation right and from the consumer-credit and platform-work directives; it will be revisited when guidance is published.

Annex A. Record schema

The JSON Schemas in `schema/decision-record.schema.json`, `schema/export.schema.json` and `schema/producer-profile.schema.json` state the structure that sections 2 to 6, 8 and 10 require of one record, of one export and of a producer profile. They cannot state the rules that span records (references between records, OVS-03, `rule_id` reuse, the chain and the anchors), the number form of REC-01, the checks on an anchor's timestamp token, the agreement between an export and its producer profile, or the rules inside one record that depend on other fields or on what a value means; the verifier checks those. The prose governs (GOVERNANCE.md): where the schema and the prose disagree, the schema is corrected.

Annex B. Interpretation guide (informative)

How to read one record, in order:

1. `rulebook` says which rules apply. `record_version` says which producer schema wrote it.
2. `event_type` says what happened. A `decision` is a check; an `oversight` is a person acting on a check; a `contest` is the person affected asking for something; `fallback`, `logging_failure` and `opt_out` say the check or the record was not normal.
3. `deciding_system` is the AI system that was checked. `checker` is the thing that checked it. They are never the same object.
4. `rules_fired` lists the rules that found something, each with the provision as an ELI web address and the provision's text. `rule_set.digest` is the exact rule set that ran; it resolves at `rule_set.index_url`. An empty `rules_fired` with a `rule_set` means the rules ran and found nothing.
5. `verdict` is what the checker concluded; `action` is what was done about it.
6. `commitments` prove which texts the record is about, without holding it. To check a text, use the producer's verification interface (CMT-06). `source_ref` says where the text is held by the account.
7. Every person is a pseudonym. The same person has the same pseudonym within one account. A role may be in clear.
8. `chain.seq` is the record's position; `chain.prev` ties it to the one before. A gap, a duplicate or a mismatch is a finding, named in EXP-04.
9. `retention.class` and `retention.erase_after` say how long the record lives. A `legal_hold` record says which range of the chain must not be destroyed, from when, and when the hold was lifted.
10. For a decision about a person: `decision_mode` says how much a human decided; `legal_basis` says why the processing was lawful; `notice` says whether they were told; `explanation_inputs` is what a later explanation is built from; a `contest` record says what they asked for and what happened.

Annex C. Mapping to the editor's first implementation (informative)

The editor's implementation writes ADCR records directly: one mapping, `record_version: complyedge-adcr-1`, turns each stored row into the record that the write-time chain hashes and the export serves, so a reader receives the bytes that were hashed. Attributes ADCR does not name are kept under `complyedge`, so the chain covers the whole row.

Rulebook field	ComplyEdge source
`rulebook`	`rulebook_version`, set when the row is written; a row written before it existed names 0.1.0
`record_id`, `account_id`, `deploying_party`	`event_id`, `tenant_id` (the account is the deployer)
`occurred_at`	`timestamp`, UTC with `Z`, milliseconds or finer
`deciding_system.id` / `.version`	`agent_id` / `deciding_system_version` (caller-supplied)
`checker.id` / `.version` / `.path` / `.model`	`complyedge`, `bundle_id`, `engine_path`, `model_used`, `confirm_model`
`verdict` / `action` / `native_result`	from `action` (`block`, `allow`, `flag`)
`rules_fired[]`	violations that carry rule metadata; each with `provision.eli` and `label` from the rule file's header
`reason_codes`, `complyedge.non_rule_findings`	engine findings that are not rules (REC-15, REC-16)
`rule_set.digest` / `.label` / `.index_url`	`bundle_digest`, `bundle_id`, the published rule set index
`exemptions.claimed` / `.rejected`	`exemptions`, `rejected_exemptions`
`fallback`	`engine_path: fallback_block`, `layer2_skipped`
`commitments[]`	`text_hash` (`hmac-sha256-1`, with a commitment domain tag), `pseudonym_key_version`, `direction`, `source_ref`
`user`, `session`, `subject`	`p1:` pseudonyms under the account key, with their own domain tag
`correlation_id`, `purpose_ref`, `data_categories`, `special_category`, `legal_basis`, `subject_is_employee`, `works_agreement_ref`	caller-supplied decision context, format-checked on input
`retention.class` / `.erase_after`	the account's retention setting at write time
`event_type: logging_failure`	rows that record a failed write or anchor
`event_type: oversight`, `proposal_ref`, `oversight_action`, `reviewer`, `reviewer_role`, `authority_ref`, `review_opened_at`, `inputs_ref`	a reviewer's final review of a flag in the dashboard (`accepted` or `disregarded`), written on the chain before the flag changes; `reviewer` is a `p1:` pseudonym of the signed-in person
`chain.seq` / `.prev` / `.hash` / `.alg`	`sha256-chain-1`, written at write time
anchors	envelope `anchors[]`, `anchor_coverage`

Measured on 15 October 2026 with the reference verifier (OpenSSL 3.6.4, `--profile`, `--trust-anchors`) on exports built through the implementation's real API write path:

- An export with a test-authority token, the trust anchors and the profile: `verified`, every record chained, no findings. Without trust anchors: `refused`. With an unsigned token: `failed` (`anchor_token_signature`). With one stored verdict edited: `failed` (`hash_mismatch`). A plan that keeps no anchors: `unverified` (`no_anchor`).

- Record, Law, Chain and Export require the decision context on every check in the export (`deciding_system`, `data_categories`, `purpose_ref`). The API accepts a check without it, decides and records it, and the reference verifier reports that record under REC-07 and REC-13. They are met only for accounts on one of the profile's retention classes; a custom period, or no expiry, is refused at the retention API.
- The default retention class is 184 days and is declared high-risk-capable under REC-20.
- Law: 136 of the 136 rules in the rule set are approved by a named reviewer, most through agent-delegated review, which each approval line states; exemption claims carry the rule's ELI or https provision when the rule is in the corpus.
- Anchor: met with the test timestamp authority only. A qualified TSA endpoint is configured, but the production hourly anchor job is not live yet, so no production export carries a qualified anchor. Production hourly anchoring stays off until that job is deployed and a live token is proven.
- Oversight: met for reviews made in the dashboard, under the same conditions as Record, Law, Chain and Export. An export holding a review made through the implementation's real review endpoint gives `verified`, every record chained, no findings; with `oversight` removed from the profile's event types, the same export gives `failed` (`profile_mismatch`). The implementation emits only `accepted` and `disregarded`, and the profile states OVS-03, OVS-04 and OVS-05 as not applicable.
- Rights: met on the live write path for decision records about a natural person on RGT-01, RGT-02, RGT-04 and RGT-05 when the caller supplies `subject` with `decision_mode`, `notice` and `explanation_inputs` (and `withheld`/`escrow_ref` when information is withheld). Contest, notice and incident event types are written via `POST /v1/audit/contest`, `/v1/audit/notice` and `/v1/audit/incident` (RGT-03, RGT-06, RGT-07). A lawful opt-out that suppressed recording is written via `POST /v1/audit/opt-out` (`event_type: opt_out`, REC-18). The producer profile lists those event types. Measured 16 October 2026: an export built through `log_audit_event` and `record_rights_event` and checked with the reference verifier carried no Rights findings (anchors may still be absent). The editor still claims no class: `classes` stays empty.
- Classification: implemented. `POST /v1/audit/classification` (Enterprise) writes a chained `classification` record with `classification_result` (`high_risk` or `not_high_risk`) and `assessment_ref` (REC-17); the profile lists `classification`.
- Fallback as its own event: implemented. A check that was skipped, degraded or replaced may be written as `POST /v1/audit/fallback` (Enterprise) with a `fallback` object (`kind`, `reason_code`) (REC-18); the same object may still appear on a decision record. The profile lists `fallback`.
- Legal holds: implemented. `POST /v1/audit/legal-hold` (Enterprise) writes a chained `legal_hold` record; the profile lists `legal_hold`; DynamoDB TTL is removed for covered sequence numbers until the hold is lifted.

The editor claims no class: the profile's `classes` list is empty.

Annex D. Changes from 0.1.0 (informative)

- RGT-03: a sixth `request_kind`, `correction`, for a request to correct personal data the decision used (Regulation (EU) 2016/679, Article 16; Colorado SB 26-189, C.R.S. 6-1-1705(1)(a)(I)).
- REC-02: a record keeps the version it was written under. An export under 0.2.0 may carry records that name 0.1.0, and each is checked against the rules of its own version. A record that names 0.1.0 and carries `correction` is invalid, because 0.1.0 has no such kind.
- The envelope of an export and the producer profile name 0.2.0.
- Vector set 0.2.0 adds cases for a correction request and its answer, a correction in a record that names 0.1.0, and an export that mixes 0.1.0 and 0.2.0 records.