STUNIR Code Emission Contract (v0)

This document specifies the **deterministic IR → code emission** interface used by STUNIR.
It is intended to make emission:
- deterministic (byte-for-byte where applicable),
- hermetic (no network, no ambient host data),
- checkable by digest binding (receipts), not by trusting re-execution.

1) Determinism contract (normative)

Emitters MUST obey:
- Encoding: UTF-8
- Newlines: LF ( )
- No timestamps in emitted source or receipts (except via an explicit, controlled `emit_epoch` input)
- No environment-dependent behavior (locale/timezone/randomness) influencing output bytes
- All iteration order MUST be defined by the IR order (the IR is responsible for canonical ordering)

1.1) `emit_epoch` (normative)

- `emit_epoch` is an explicit input used to control any time-derived metadata (e.g., output file mtimes, receipt metadata).
- Emitters MUST NOT read the current time.
- The effective `emit_epoch` MUST be sourced from either:
- the IR (if the IR carries an epoch field), or
- an explicit CLI argument.
- If no `emit_epoch` can be determined, strict emitters MUST fail.

2) IR contract

The IR MUST validate against:
- `schemas/stunir_ir_v1.schema.json`

Additionally (not expressible in JSON Schema):
- IR arrays that represent sets MUST already be in canonical order.
- Unknown fields MUST be rejected by strict emitters.

3) Template Pack contract

A template pack for language `{lang}` lives at:
- `templates/{lang}/`

It MUST contain:
- `templates/{lang}/TEMPLATE_PACK.json` (machine-readable pack descriptor)
- `templates/{lang}/templates_manifest.tsv` (digest binding for all files that influence emission)

3.1 TEMPLATE_PACK.json

`TEMPLATE_PACK.json` MUST validate against:
- `schemas/stunir_template_pack_v1.schema.json`

3.2 templates_manifest.tsv (normative)

Format per non-empty, non-comment line:

- `SHA256_HEX<TAB>REL_PATH`

Where:
- `SHA256_HEX` is 64 lowercase hex characters (SHA-256 digest of the referenced file bytes).
- `REL_PATH` is a **safe relative path** (forward slashes) rooted at `templates/{lang}/`.

Rules:
- File encoding: UTF-8
- Newlines: LF only
- Ordering: the file MUST be sorted lexicographically by `REL_PATH` under `LC_ALL=C`.
- Rationale: ordering-by-path ensures deterministic output and matches `pack_manifest.tsv` ordering.
- `REL_PATH` MUST be a **safe relative path**:
- MUST NOT be absolute
- MUST NOT contain `..` segments
- MUST NOT contain NUL
- SHOULD NOT begin with `-`
- Scope:
- The manifest scope is **all files that influence output bytes** (templates, pack descriptor, any deterministic helper data).
- The manifest MUST include `TEMPLATE_PACK.json`.
- The manifest MUST NOT include `templates_manifest.tsv` itself (avoids fixed-point/self-hash ambiguity).

4) Emit receipt

Emitters SHOULD write an emission receipt at a stable location (project-specific), with schema:
- `contracts/receipt_emit_v1.schema.json`

Minimum required bindings:
- IR digest (`ir_sha256`)
- Template pack digest (`template_pack_sha256`)
- Emitter identity evidence (id/path/hash or version-text hash)
- Output file list (path + sha256)

Design principle: verification should not require re-running the emitter; it should be possible to verify by digest binding.