

Spine Component Seed

Spine Evaluation Edition 0.7.4 | Component Seed form

© 2026 Big Intelligence AI, LLC. You may copy this form and fill it in to use with Spine.

What this form is for. A Component Seed describes one thing you procure or build as a unit (an ECU, a sensor, an actuator, a harness, a software module) in your own words: what it must do, what it connects to, what could go wrong, and who is responsible for what. Your AI assistant adds it to the Spine model, and Spine's **Component Seed Quality Report** then shows which of the thirteen design categories below the model covers.

How to use it.

1. Fill it in, replacing each _____. Plain language is fine.
2. Save it as a Markdown (`.md`) file in your Spine project folder, for example `seeds/component-seed-<name>.md` , next to the Feature Seed of the feature it serves.
3. Ask your AI assistant to build the model from both files.

A worked example of both forms, for an electric power steering system, ships with Spine.

Who fills this in. The person who owns the component — often a supplier engineer, sometimes a component owner on the OEM side. **You should not need a functional-safety or cybersecurity specialist next to you to complete this form.** Where a category asks about safety or security, it asks what this component touches and what could go wrong, in your words — that is an `answered` disposition however short. Reserve `not_applicable` for categories that genuinely do not apply to this component, and say why.

Section 1 — Component Identity

Component Name *(required)*

What is it? *(required)* — one or two sentences: the physical or logical thing this seed describes (an ECU, a sensor, an actuator, a wiring harness, a software module procured as a unit). Not what it does yet — that is the `functional` category in Section 6.

Who supplies it? *(required)* — in-house design, or a named Tier-1/Tier-2 supplier. If a supplier, name them; if not yet selected, say "TBD" and note when a decision is expected.

Target Vehicle / Product Line *(required)*

Which feature(s) does this component serve? *(required)* — a component seed almost always exists in service of a feature seed. Name the feature(s) it is procured or designed for, if known.

Section 2 — Component ASIL

Read this section before the design categories — the answer here changes which of them Spine treats as required. A QM component is not asked for a safety case it does not need; an ASIL D component cannot skip safety.

- **Is an ASIL already allocated to this component?** yes | no
- **If yes, allocated ASIL:** QM | A | B | C | D
- **Allocated by:** *(which vehicle-level analysis, and by whom — e.g. "OEM vehicle-level HARA, dated 2026-06, ref HARA-114")*

● **An ASIL arriving here is *allocated*, not *derived*.** A component seed often carries an ASIL already decided by an upstream vehicle-level analysis that Spine did not perform. The model records that allocation as an **input from your analysis** — never as a determination Spine made. If a later analysis disagrees with the allocated ASIL, that disagreement is raised for a person to resolve; the value recorded here is never quietly overwritten.

If no ASIL has been allocated yet, answer "no" and do not guess one. Describe what you know about exposure — what this component touches, what could go wrong — in the `safety` and `cybersecurity` categories instead. With no ASIL, Spine reports those categories as "unknown", not as "not needed", so answer them whenever there is any exposure.

Section 3 — Design Constraints

Include connectivity protocols, compute platform, packaging/mounting constraints, and any standard architecture decisions your organization mandates (e.g. CAN-FD backbone, a required MCU family, a required connector series). This is background for the `non_functional` and `interface` categories in Section 6, not a category answer on its own.

Section 4 — Existing Component Specification

If an OEM component specification already exists, you do not need to re-key it. Reference it here and answer only the categories it does not cover. Every one of the thirteen categories in Section 6 still needs a disposition — a reference to the existing specification is a legitimate `answered` disposition for a category **when the referenced document genuinely covers that category**. Point at the section or page, not just the document name, so the answer can be checked.

Existing component specification (if any):

___ *(document name/ID, and which sections cover which categories in Section 6)*

Section 5 — Existing Documentation

Any other artifacts this seed draws on — a datasheet, a prior-program FMEA, a supplier capability statement. Each item is optional; the thirteen categories in Section 6 are not.

Known field issues / failure history (if this component or a close predecessor has shipped before):

Existing certifications:

Section 6 — Design Categories

Thirteen categories. Every one must carry a **disposition**, answered or not_applicable. There is no third option: a blank category is a gap, not a light answer.

Which categories Spine treats as required (your Section 2 ASIL decides):

- **Always:** functional, non_functional, interface, context_boundary, environmental, regulatory, vv_criteria, robustness.
- **At ASIL A or above:** safety, cybersecurity, diagnostic, responsibility_split. At QM they are not required; with no ASIL known, Spine reports them as "unknown".
- **Recommended:** variants.

not_applicable needs a reason — one line is enough, e.g. *"not applicable — this component has no external interface, because it is potted and wired point-to-point with no addressable protocol."* At QM, say why the exposure is absent ("this component moves nothing and is not reachable from any external bus"), never just "because it is QM". In this edition, the Component Seed Quality Report counts only the requirements in the model, so a required category you mark not_applicable is still listed as having none. Your stated reason is the record of why.

No brand section here. Brand perception ("will it protect my family?", "is it protected from theft?") belongs to the *feature* this component serves, on its Feature Seed. A component inherits it rather than carrying its own.

functional — What should it do?

- **Disposition:** answered | not_applicable
- **If not_applicable, why:**
- **Answer:**

non_functional — How well, how fast, how often?

See Section 3 for any standing platform/protocol constraints that bound this answer.

- **Disposition:** answered | not_applicable
- **If not_applicable, why:**

- **Answer:**

interface — What does it exchange with, outside itself?

◆ **This category carries more weight here than on a feature seed.** A component handed to a Tier-1 lives or dies on its interface contract — get this wrong and the supplier builds against the wrong pins, signals, or protocol version. List every physical and logical interface: connectors, pins, buses (CAN/LIN/Ethernet/etc.), signal names or a reference to a signal database, power in, and any interface to a human (a control, an indicator). For each, say who owns the interface definition — you, the supplier, or a shared spec both parties reference — and where that definition lives. If an interface control document (ICD) already exists, reference it here rather than re-listing every signal.

- **Disposition:** answered | not_applicable
- **If not_applicable, why:**
- **Answer:**

context_boundary — Where does it operate, and what is explicitly not yours?

State what this component owns versus what a neighboring component or the host feature owns. Example: "this component detects the fault condition; raising the driver-facing warning is the head unit's job, not this component's."

- **Disposition:** answered | not_applicable
- **If not_applicable, why:**
- **Answer:**

safety — Could this hurt someone, and how?

Describe the exposure in your own words — for example "commands physical motion near people" or "controls a load-bearing mechanical latch". **Do not supply an ASIL here.** Section 2 records the ASIL if one has already been *allocated* to you by an upstream analysis; this field is for exposure in plain language.

- **Disposition:** answered | not_applicable
- **If not_applicable, why:**
- **Answer:**

cybersecurity — What would an attacker want from it?

Describe the exposure in your own words — for example "accepts commands over a bus reachable from outside the vehicle" or "stores a credential". **Do not supply a CAL.** The analysis determines that, and a low result still traces back to what you wrote here.

- **Disposition:** answered | not_applicable
- **If not_applicable, why:**
- **Answer:**

environmental — Where does it live — heat, water, vibration, dirt?

Mounting location, ingress-protection expectations, temperature range, vibration/shock exposure. A reference to a standard environmental class (e.g. "engine-bay class per our platform spec") is a complete answer.

- **Disposition:** answered | not_applicable
- **If not_applicable, why:**
- **Answer:**

diagnostic — When it breaks, how does anyone find out?

What fault conditions should this component detect and report, and how (a DTC, a status signal, a visible indicator)? A short answer is enough — a later failure analysis may add to the list.

- **Disposition:** answered | not_applicable
- **If not_applicable, why:**
- **Answer:**

regulatory — What rules apply, or which markets is it sold in?

- **Disposition:** answered | not_applicable
- **If not_applicable, why:**
- **Answer:**

vv_criteria — How will you know it works?

What test, inspection, or acceptance criterion would tell you this component does what the functional category says it should?

- **Disposition:** answered | not_applicable
- **If not_applicable, why:**
- **Answer:**

responsibility_split — Who builds which part, and who is accountable?

◆ **This category carries more weight here than on a feature seed.** A component handed to a Tier-1 needs an explicit Development Interface Agreement (DIA — the ISO 26262 agreement assigning each safety activity to you or the supplier) or a RASIC (Responsible/Accountable/Support/ Informed/Consulted) split, not an assumption that "the supplier handles it." State, at minimum:

- Who owns the requirements for this component (you, or delegated to the supplier)?
- Who designs it, who verifies it, and who has final sign-off authority?
- If a supplier is building it, what do you retain approval rights over (e.g. the FMEA, the safety case, any design change)?

A one-line table is enough if the split is simple; a reference to a supplier quality agreement (SQA/APQP) is a complete answer when it covers this.

- **Disposition:** answered | not_applicable
- **If not_applicable, why:**
- **Answer:**

variants — Does it differ by market, trim, or model?

- **Disposition:** answered | not_applicable
- **If not_applicable, why:**
- **Answer:**

robustness — Does it keep its promises when something is interrupted?

What must this component remember across a power cycle, how it recovers from an interrupted command, its behaviour when its link to the rest of the system is degraded or absent, and what it may legitimately forget. A short answer is enough — reference a platform-level robustness spec if one covers it.

- **Disposition:** answered | not_applicable
- **If not_applicable, why:**
- **Answer:**