A CONTRIBUTION TO THE MULTI VECTOR INDUSTRY CHALLENGE · MODELS 2026

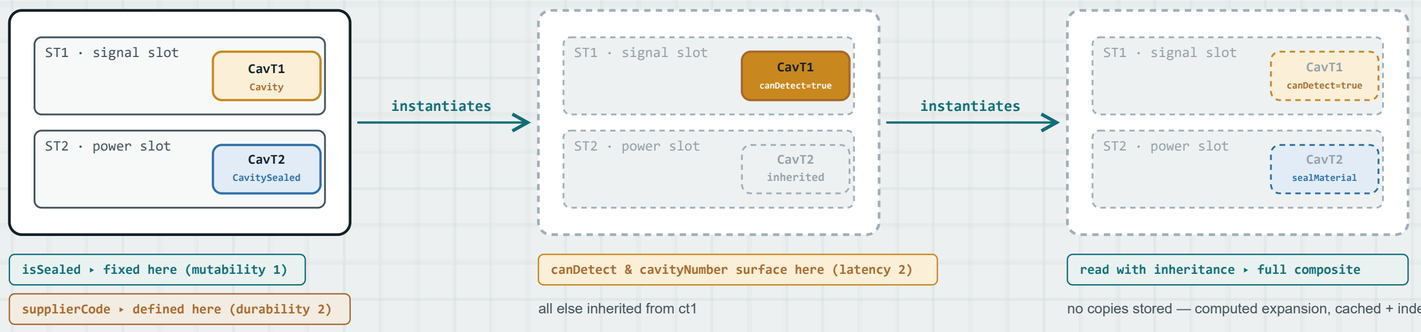
The emulation machinery disappears.

Native deep instantiation for automotive connector catalogs on the Metada no-code platform.

Petr Smolík · Metada · petr.smolik@metada.com RBSM / Meta4 · interpreted at runtime Live: **https://demo.metada.com/multi/vector-challenge/**

PREEvision emulates three connector levels inside a two-level architecture — Type-Object for the structured typeOf, a vertical Type-Square for typed composition. On a platform where instantiability is a declared capability of every object, both patterns don't get re-modeled more elegantly — they dissolve.

Type → Prototype → Instance · one concept per entity · one relationship · three dials per feature — latency (appear) · mutability (freeze) · durability (persist)



ONE Connector CONCEPT · POTENCY 3 · THREE INSTANTIATION DEPTHS OF ONE rbsm:instantiates CHAIN

Detecting cavity (amber) is the plain Cavity variant; sealed cavity (blue) is CavitySealed — selected once at the type, inherited below. Solid = local decision · dashed = inherited / projected.

One housing, three depths. Each level stores only its own decisions; everything below is computed expansion — derived, cached, search-indexed. The sealed-cavity variant, chosen once at the type, is inherited all the way down. Illustration, not a tool rendering; the tool's live output is in panel 04 (a diagram view for Meta4 models is on the roadmap).

00 THE PLATFORM UNDERNEATH

Two metamodels, stacked: **RBSM** (linguistic) under **Meta4** (ontological), both interpreted at runtime.

Deep instantiation isn't application code — it's an engine primitive.

`No-code` The challenge is **data** in the interpreted zone. Its editors, API, tests, and the live instance form are **interpreted from the model at runtime** — nothing is generated or compiled.

`Meta4 · interpreted` The ontological metamodel turns the model into **Concepts, level-aware schemas, validation, and the UI** — read from storage and rendered as dynamic forms, no view code.

`RBSM · engine` The linguistic meta-metamodel supplies **instantiates · extends · potency · durability · mutability · latency** — the multi-level substrate every model inherits, written once in the engine.

So the three connector depths above aren't something this submission coded — they're what the platform already does.

01 THE CHALLENGE

Connector **types** in a catalog, **prototypes** placed in subsystems, **instances** installed in vehicles — each level a structured **typeOf** of the one above, and the relation must reach the **parts** too: slots, and their cavities, level by level.

Features are introduced at a level and projected downward: **isSealed** fixed at the type, **canDetect** first valued at the prototype, **sealMaterial** only on the sealed variant. The introduction level is itself declared — **`rbsm:latency`** keeps a feature dormant until its depth is reached.

02 EMULATION → NATIVE

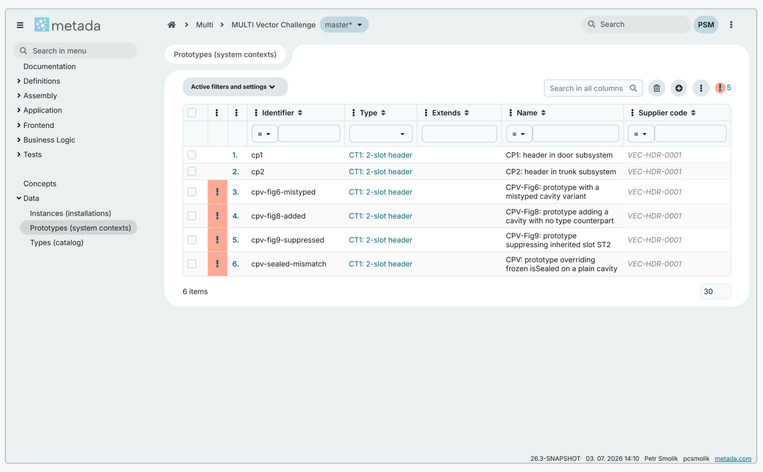
PREEVISION — EMULATED	METADA — NATIVE
3 M2 classes per entity + Type-Object links	one concept each: `Connector`, `Slot`, `Cavity`
type / instance associations + constraint lattice	**`rbsm:instantiates`** — linguistic, transitively materialized
vertical Type-Square (SlotType / Proto / …)	embedded `PropertyComplex` parts; **part typing derived** = `(owner typeOf) ∘ (rbs:id)`
M2 variant taxonomy (Cavity vs CavitySealed)	**`rbsm:extends`** + per-feature suppression
gray attributes / feature projection	**`instantiateValue`** flag + **`latency`** / **`mutability`** dials — `project · appear · freeze`

03 STORE DECISIONS ONLY

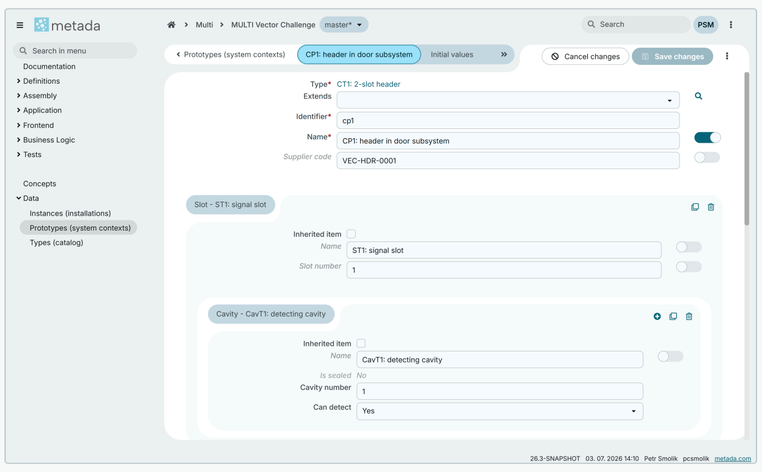
```
# ct1  (TYPE)  slots, variants, isSealed,
       supplierCode ← defined here
# cp1  (PROTO) canDetect=true,
       cavityNumber=1  (sparse)
# ci1  (INST)  name only
```

Reading `ci1` with inheritance returns the **full composite**: both slots and cavities, the per-part variant, the projected value, the mid-chain `supplierCode` — each tagged with its origin. Inherited values are **derived, cached & Lucene-indexed** — reads and filters never walk the chain; nothing is stored twice.

04 THE TOOL, RUNNING



The prototypes grid, **interpreted live** — no view code. Rows typed by CT1; **`supplierCode`** greyed = **inherited**; the four violation fixtures **flagged live** by the validator.



The `cp1` prototype, slot & nested cavity: **`canDetect`** & **`cavityNumber`** **surface and get their first values here**; **`isSealed`** already **frozen** — latency & mutability enforced by the editor.

05 FOUR MECHANISMS CARRY IT

Depth of concept-hood

Deep instantiation applied to the property-definition facility itself — how many levels stay "concepts" is a declared model value, not a fixed constant.

Appear, project, freeze, persist

Three dials count down one step per instantiation: **latency** says at which depth a feature surfaces, **mutability** how long its value stays editable, **durability** how long it exists at all. The **instantiateValue** flag projects the value down — the "gray attribute" is exactly these declarations.

Variants by subtyping

`CavitySealed extends Cavity` — adds **sealMaterial**, suppresses **canDetect**. No parallel class family per level.

Derived part typing

The expander deep-merges composite parts **by rbs:id** along the owner's instantiation edge. The part-level typeOf is computed — never stored.

06 INTEGRITY, THREE TIERS

CORRECT BY CONSTRUCTION

Unrefined part-typing congruence — the cross-owner mistypings of Figs. 6–7 are inexpressible.

ENGINE-ENFORCED

Schema closure, potency exhaustion (3→0), durability / mutability / latency countdowns, identity uniqueness.

EXPLICITLY VALIDATED

The genuinely violable conditions — bijection, admissibility, variant refinement — checked by `VC_ValidateStructure`.

07 VERIFIED — IT RUNS

- ✓ **`6 / 6 tests pass`** — valid chain + violation fixtures for Figs. 6, 8, 9, sealed admissibility, type-level guard.
- ✓ **`ci1 resolves live`** — full inherited composite; `instantiatesTransitively` Object→Connector→ct1→cp1.
- ✓ **`Interpreted UI renders`** the instance form — variant-specific fields exactly where they belong.
- ✓ **`Dials enforced live`** — `isSealed` frozen at cp1 & ci1 even when localized; `canDetect` dormant at the type, active from the prototype down.
- ✓ **`Level-aware schemas`** — closed at every depth, potency-aware definition rights.

08 SEE IT LIVE

`https://demo.metada.com/multi/vector-challenge/`

The **running** model — interpreted live, not generated. **Free registration** required; then browse the catalog → prototype → instance chain, the live grids, and the validator.

Honest scope: embedded parts aren't externally addressable repository objects — by design; the same concepts support a folder-composition encoding when external part references are needed.

✓ TAKEAWAY

The challenge measures the cost of emulating multi-levelness: triplicated classes, maintained typing links, a constraint lattice to keep them congruent.

Native deep instantiation reduces the same scenario to **one concept per entity, one relationship, three dials and a projection flag per feature, and a derived part-typing relation that makes the hardest congruence conditions unviolatable.** What stays explicit — the violable conditions and change-management guards — is a small, well-motivated rule set, which is exactly where it belongs.