Three regulations, modelled by lawyers, turned into executable law

Decision services that can be executed, tested rule by rule, and published as machine-readable law — run three times, for Amsterdam, SZW and Den Haag.

3	34	179	286	0
government bodies	decisions published	rules, one test case each	test cases against the live engine	failures today

THE PROBLEM THIS CLOSES

An export that opens in a modeller is not a model a computer can execute.

All three delivered exports failed to deploy or failed to answer. Every one now runs, is tested rule by rule, and is published as linked data with its legal basis attached.

TWO CAVEATS THAT STAY ATTACHED

The 179 rules are the **published** models. Two of the three grew during derivation (Den Haag 44 → 55, SZW 17 → 25) — so this is not a count of what legal analysis delivered.

"0 failures" means every case passes against the live engine **today**. It does not mean the models are legally correct: **four decisions await the responsible body, plus one flagged assumption** — listed on slide 06.

Six stages execute. Three need a person. One sends you back.

Execution · Human decides

01
RECEIVE
A DMN export arrives from the body that did the legal analysis. Nothing is assumed about it.

02
SURVEY
Does it deploy? Does it answer? What does the validator say? Does the answer look right?

03 · HUMAN
DECIDE
The derivation strategy. Needs a domain expert — the temptation is to make it run by quietly changing what it decides.

04 · HUMAN
DERIVE
Produce the patched model. Every difference from the original is listed in the changelog.

05
DEPLOY
Push to the decision engine. It versions rather than replaces, so a deployment is additive and reversible.

06
TEST
One case per rule, each routed to the decision whose rule it exercises. Run live against the engine.

07
PUBLISH
Attach to a public service in the CPSV Editor, export Turtle, validate against the SHACL shapes.

08 · HUMAN
INSPECT
Read the published file. The stage most likely to be skipped — and four findings appeared only here.

09
RECORD
Three documents per pass: changelog, validation write-up, and a runner anyone can execute.

08 → 06 · what publishing reveals sends you back to the suite

One workflow absorbed three very different models

Every number measured from the repository. Arrows read "before → after" derivation.

AMSTERDAM — WIDE
25 shallow decisions on income schemes. The problem was volume and repetition.

SZW — DEEP
Two decisions: one emits 20 *bijstandsnorm* amounts, the other picks one per *peildatum*. A chained lookup over time.

DEN HAAG — STRUCTURAL
Written against an object model no decision engine can navigate. The problem was translation.

MEASURED	Amsterdam	SZW	Den Haag
Source tool	iKnow export	Camunda Modeler	Camunda, over an object model
Decisions	**25**	**2**	**9 → 7**
Rules	**99**	**17 → 25**	**44 → 55**
Declared inputs	52	2	2 → 26
Test cases · decisions covered	**100** · 25 of 25	**121** · 2 of 2	**65** · 7 of 7
Legal links in the source	48 sources / 99 links	none	11 sources / 12 links
Cell-level legal grounding	1 rule, 6 cells (proof of concept)	**80 cells**	none available
Published as linked data	Digital-Twin-Inkomensregelingen.ttl	Normenbrief—-Informatie-voor-gemeenten.ttl	Aanvraag-LevensOnderhoud-ALO.ttl

A defect catalogue that grows across the passes

Amsterdam discovered most of it. By Den Haag the same checks were run up front. Fifteen recurring defect families are now looked for before a model is even deployed.

THE C-LEVEL SUMMARY

Every one of these is invisible until the model is executed — and several are invisible even then.

That is what the test suite is for.

NOT A VALIDATION TOOL

The validator passed all three exports as "valid" while none of them could run correctly. Passing validation is a weak signal, not a tick.

DEFECT	EFFECT	FOUND IN
Unreachable approval rules	**The model could not grant entitlement at all**	Den Haag
Amounts copied into the wrong cell	Wrong benefit paid	SZW (2 half-years)
Outputs no input could reach	7 of 20 amounts unobtainable	SZW
Two inputs mapped to one output	Two situations answered identically	SZW
Malformed conditions	Silently resolve to the wrong thing	Amsterdam (56)
Output column with a label but no name	Blank, unlogged error — hides behind any other defect	Amsterdam, Den Haag (6 of 9)
Empty input expression	**Invalid model that works** — fails later when something unrelated changes	SZW, Den Haag
Value type contradicting its own cells	Right answer, wrong type, no warning	SZW
Object-model navigation	No decision engine can follow it	Den Haag (3 decisions)
Empty placeholder rules	Match everything not caught earlier, return nothing	Den Haag (3)
Wildcard default under the wrong hit policy	Two rules match where one must	Amsterdam

Plus: multi-word bare names, missing history setting, unescaped **&** in a URL (48), **n/a** used as a condition (10).

Eight judgements no pipeline can make for you

Three stages need a human; one needs a domain expert who can say whether an answer is right.

DECISION	THE REAL CHOICE	WHAT WAS CHOSEN, AND WHY
Which file is the deliverable	Patch the original, or derive a new one	**Derive.** The original stays as the auditable record of what legal analysis delivered.
A wrong amount	Correct it, or leave and report	**Report.** Corrected only once a second published source confirmed it.
A missing reason code	Invent one, or leave the rule untestable	Invent, and **flag it as the only value in the model with no source**.
An unreachable output	Extend the model, or record the gap	Extended where the vocabulary was mechanical (SZW); recorded where it was policy (Den Haag).
A rule that matches nothing	Fix the logic, or record it	**Record.** What a mixed household is entitled to is policy, not transcription.
Disconnected decisions	Wire them in, or leave them	Leave. Wiring them in would decide when a *hardheidsclausule* applies.
Unknown facts	A second boolean, or a third value	A **third value** — so the answer can say *which* fact is missing.
Legal grounding	Ground against inferred citations, or don't ground	Ground only where published sources exist (SZW). Den Haag has none, so no grounding.

"We fix what is provably broken. We report what is arguably wrong."

A model that runs but decides something nobody authorised is worse than one that does not run. The pipeline reports questions to the responsible body — it does not answer them.

Four decisions awaiting the responsible body, plus one flagged assumption

Each is a question for a government body about its own regulation. A number without a list is not usable — so here is the list.

01 — SZW OPEN

Which article is the basis for *JO / JOK*?

Art. 20 **lid 1 c** rather than lid 2 c? Both carry the lid 2 c amount in all five *termijnen*. The amount is confirmed by a second source — the citation is not, nor whether an 18–20-year-old with an AOW-age partner and no child gets the *with children* norm.

02 — DEN HAAG OPEN

What is the correct *reden code*?

The source has a literal **???** for the ongeoorloofd-onbetaald-verlof rule. **"04"** was adopted so it could be tested — the only value in any of the three models with no source.

03 — DEN HAAG OPEN

What is a mixed household entitled to?

One person refused, the other entitled. Today the model matches no rule and returns nothing. Any answer adds a rule; **which** one is policy, not transcription.

04 — DEN HAAG OPEN

Should the three standalone decisions be wired in?

TekortSchieten, *DringendeReden*, *HardheidsclausuleToepassen* — nothing requires them today. Connecting them decides **when a hardship clause applies**, so they were left standalone and reported.

ALL FOUR BELONG TO TWO OF THE THREE PASSES

Amsterdam contributes none.

Its 99 rules were legally coherent once the technical defects were fixed. Den Haag's had genuine policy holes — a household it could not decide, three decisions connected to nothing. That is **a broken export** versus **an unfinished regulation**, and only executing the model reveals which you have.

PLUS ONE FLAGGED ASSUMPTION

Not blocking — still unanswered

Den Haag's unknown-*voorliggende-voorzieningen* rule reported an *informatiebehoefte* naming a different fact than its own column. Changed — and flagged in case the text was right and the column wrong.

THREE LANES, THREE ADDRESSEES

4+1	The regulation	The government body — the four above
2	The standard	CPRMV spec owner — identifier stability, ruleType
1	The tooling	Us — legal-source layer never reaches the file

Amsterdam adds an observation, not a question: 2 of its 99 legal links do not resolve against the annotations file.

IDEA ONE

“Unknown” is a third value, not a second flag

ROOD	ORANJE	GROEN
Refused	Information needed	Entitled

Den Haag’s source model tracked, for each of six facts per person, whether it was true, false, **or not yet established** — and when a fact was missing it said *which one*. An earlier attempt collapsed all six into a single “information incomplete” flag.

COLLAPSED	RESTORED
“Something is missing.”	“We need the residence permit.”

Both versions run. Both are “correct”. Collapsing discarded the model’s most useful answer while appearing to simplify. Restoring it turned **1 rule back into 6**, per person.

IDEA TWO

Publishing is a discovery step, not the last step

None of the findings below is visible from the model, or from a green test suite. Only from the published artefact — which is why stage 8 loops back to stage 6.

FINDING	HOW IT SURFACED
Published outputs were silently empty whenever evaluation returned no rows	Republishing Amsterdam
The generator overwrote a legal rule’s real identifier with its own web address — **12 places in one file**	Inspecting the SZW artefact (fixed)
The test runner treated a “false” answer as “no answer”	Mutation-testing our own runner (fixed)
The editor reads no part of DMN’s legal-source layer, so **23 provenance links** never reach the published file	Inspecting the Den Haag artefact (open)
An intermediate decision’s output — **the colour the entire model turns on** — was absent from the published vocabulary	Inspecting the Den Haag artefact after a second publish

Every test case had been routed through the top-level decision, so nothing ever asked the middle one directly. **The pipeline inspects itself** — three editor improvements and two defects came out of these three passes.

A green run is evidence, not reassurance

Four practices, in increasing order of how much they surprise people.

01 — COVERAGE

One case per rule

Not per feature, not per happy path. 100, 121 and 65 cases for 99, 25 and 55 rules.

02 — INDEPENDENCE

The suite does not trust the model

Each generator holds its own copy of the expected answers and emits nothing if that copy disagrees — a suite generated *from* the thing under test proves nothing.

03 — SELF-CHECK

The runner is tested too

Deliberately broken expectations confirm each is reported as a failure. Not theatre: it found a real defect in our own comparison logic.

04 — BOTH PATHS

Two verdict paths, both exercised

The command-line runner and the editor's own pass/fail logic read the expectations differently, so both run against the live engine.

WHAT COMES OUT, PER PASS

<original>.dmn	the export, untouched
<original>-patched.dmn	what actually runs
<published>.ttl	the linked-data service
CHANGELOG.md	what changed, when, and why
testCases/	the suite, the reasoning, and a runner anyone can execute

And live: a deployed, versioned decision on the engine, callable over HTTP with a documented request body.

TASK

Run it a fourth time

Amsterdam discovered the defect catalogue; by Den Haag the same checks were run up front. A fourth pass costs less than the third and returns the same artefacts — an executable model, a suite that is evidence, and a published service with its legal basis attached.

Not fully automated. Not finished at publish. Not a legal opinion — four decisions plus one flagged assumption stay open with the responsible bodies.

Three things only a standardisation body can settle

*Slide 06 split the open items into three lanes. These are the two in the **standard** lane, plus the one the other two keep producing.*

01 — IDENTIFIERS

Who owns the convention for naming a rule inside an article?

Legislation attaches at **three levels** — whole decision, whole rule, single cell — and none is authoritative. In one file the generator minted its own web address over a rule's real identifier **12 times**; 2 of Amsterdam's 99 links still do not resolve.

Deliver: persistent identifiers at rule granularity, lined up with **Juriconnect/BWB** and **ELI** rather than alongside them.
Support: a standards-management home for CPRMV — or a route to one.

02 — CONFORMANCE

Can conformant mean *it executes*, not just *it validates*?

The validator passed all three exports as valid. **None of them could run correctly** — 15 defect families deep. And 23 provenance links were lost at the DMN-to-profile boundary with the file still validating clean.

Deliver: a Dutch application profile of DMN whose test is *it deploys and it answers* — allowed subset, the defect catalogue as shared checks, one case per rule, legal source required.
Support: the intake; we contribute the catalogue and the cases.

03 — OUTCOMES

Must a decision service say *which* fact is missing?

"Information incomplete" and "we need the residence permit" both run and both pass structural validation. Six per-person facts collapse into one flag with no rule broken — so today the weaker model is conformant.

Deliver: a required third outcome — *information needed*, naming the fact — as an interoperability agreement, not a per-body choice.
Support: the motivation argument under Awb.

WHAT WE ARE NOT ASKING FOR

A study. We are asking for an owner and a conformance rule.

All three questions were produced by **running the models, not by reading them** — which is also the offer: a fourth pass, on a regulation you choose, as the evidence base.

And one gap we cannot close alone: findings land with three different addressees (slide 06), with **no agreed route back**.

WHAT WE BRING TO THE TABLE

179	rules published as executable, citable linked data
286	test cases, one per rule, running against a live engine
3	passes documented end to end — changelog, validation, runner

The defect catalogue on slide 04 is reusable as a conformance checklist by anyone adopting the profile.