

From Experts to Everyone

AI-Augmented Approach to PCB Design Verification

"The Wright brothers' first flight lasted just 12 seconds."

Build, write, and evolve DRC rules from a single line of natural language – bringing verification capability from a handful of experts to every engineer

Hyunjae Kim
Sr. Application Engineer



Remember? Three Years Ago, in This Same Room

2023 → 2026

Back in 2023, at this very forum, I proposed: **"Let's solve the expert bottleneck with verification automation."**



Background

- How do you verify your design before NPI fab?

Design to guidelines & hope for the best

Hire external Consultants to analyze

Wait for review by an Internal expert

JK ED&C SIEMENS

Background

- Collaboration of specialized engineers in each field during product development.
 - H/W Engineer
 - PCB Designer
 - Simulation Engineer
- Lack of expertise in each field
 - Mutual supplementation
 - Reuse of validated designs
 - Automation of repetitive tasks
- Specialized software for each field
 - Software is not integrated and limited to specific fields
 - Difficulties in learning and using multiple tools

Tools are complex, incomplete, and disparate

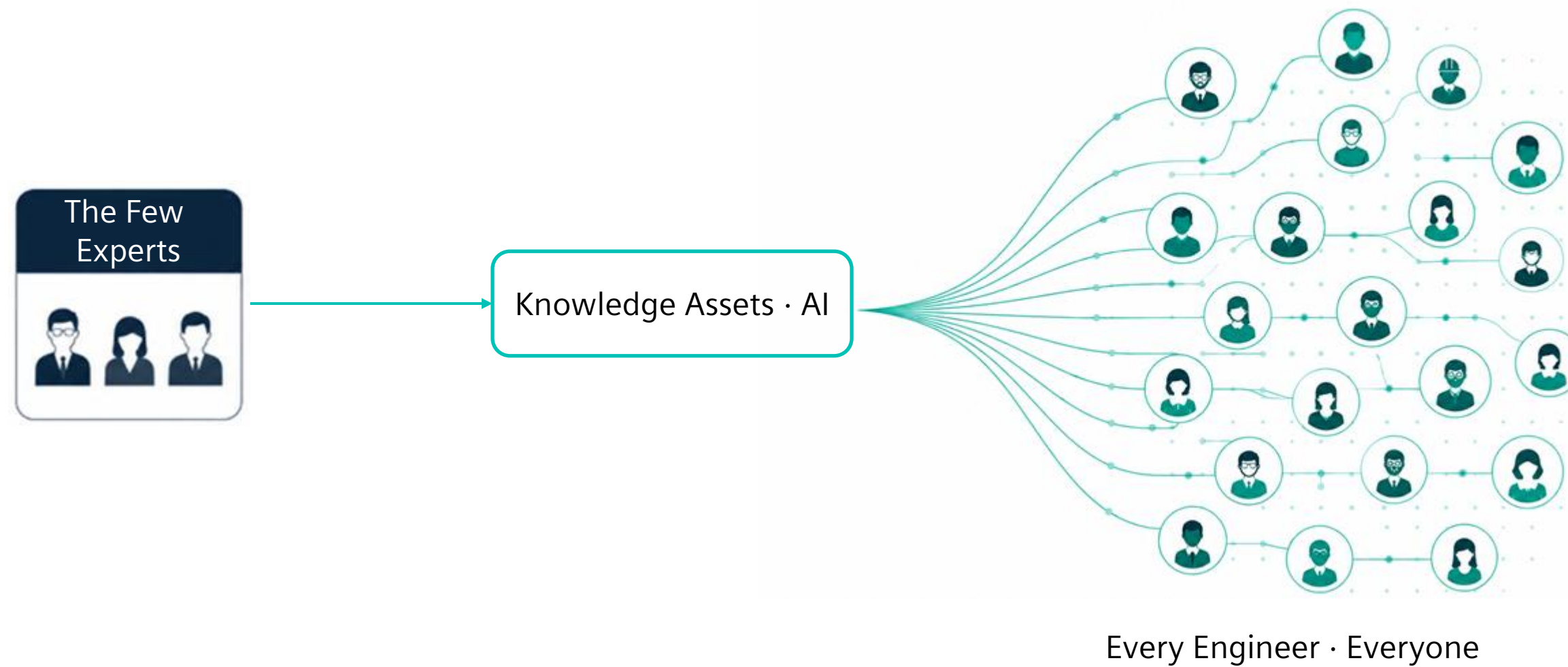
Domain specialists are scarce

JK ED&C SIEMENS

What was a trend three years ago has become today's standard practice.
The trend for 2026 begins – here, today, in this room.

DRC Authoring, No Longer Experts-Only

THE SHIFT



The key is that the structure of **"who creates it"** changes — from the few to everyone.

Today, I'll show you how that structure actually works in practice.

What We'll Cover Today

AGENDA

01

The Problem

Traditional DRC rule development and the expert bottleneck

02

Phase1 • Knowledge Assetization

Structuring knowledge — API index · patterns · knowledge

03

Phase2 • AI

Natural language → generation · testing · explanation · reporting

04

Field Cases — our deepest dive

Crop · Same-Net Cross · onboarding existing rules

05

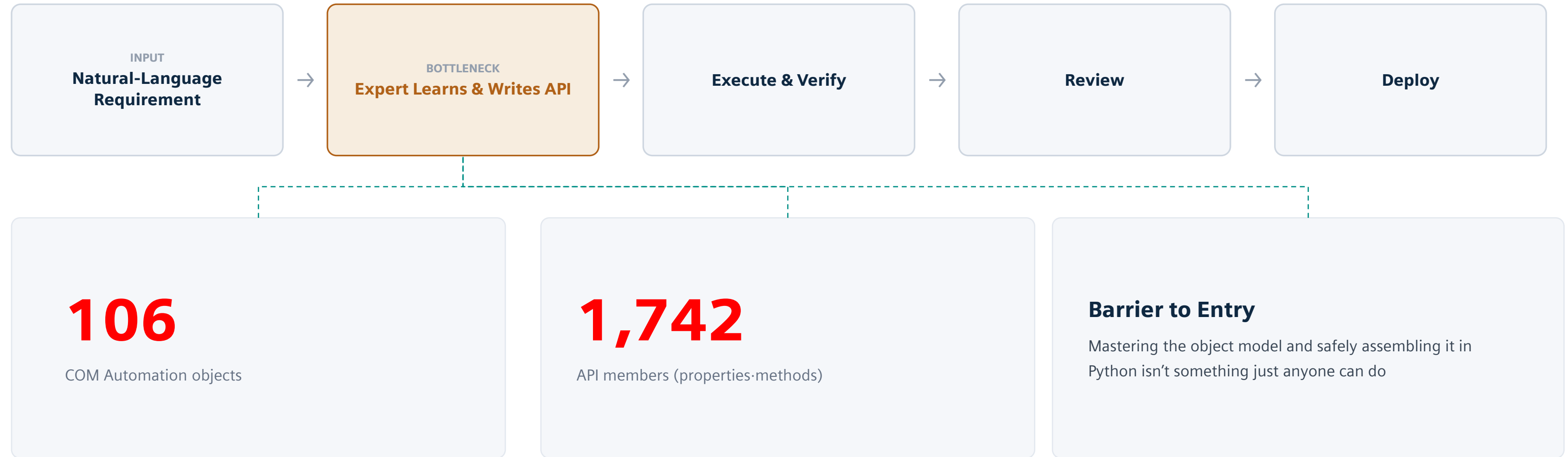
Evolution & Future

Evolution by industry & business unit, and redefining the expert

The Traditional DRC Rule Development Flow

PROBLEM 01

Every time a new requirement comes up, an expert has to dig through the COM API and hand-assemble the rule.



So Where Does It Hurt?

PROBLEM 02

A

Schedule Bottleneck

Every new rule means waiting in line for an expert's availability.

B

Locked-In Roles

Engineers are stuck at "execute&review." They can't create.

C

Quality Variance

Edge-case and exception handling swings wildly depending on who wrote it.

"How do we share this capability with everyone?"

First, Turn the Expert's Mind into a Structured Asset

STAGE 1 · Assetization

01

AOM Local Index

106 API objects · 1,742 members, turned into a searchable index. No more humans digging through docs.

02

Pattern Library

Dimensions · traces · spacing · vias · BGA · components... verification scripts by category.

03

Knowledge Base

Gotchas · errors · patterns, accumulated on a per-object basis..

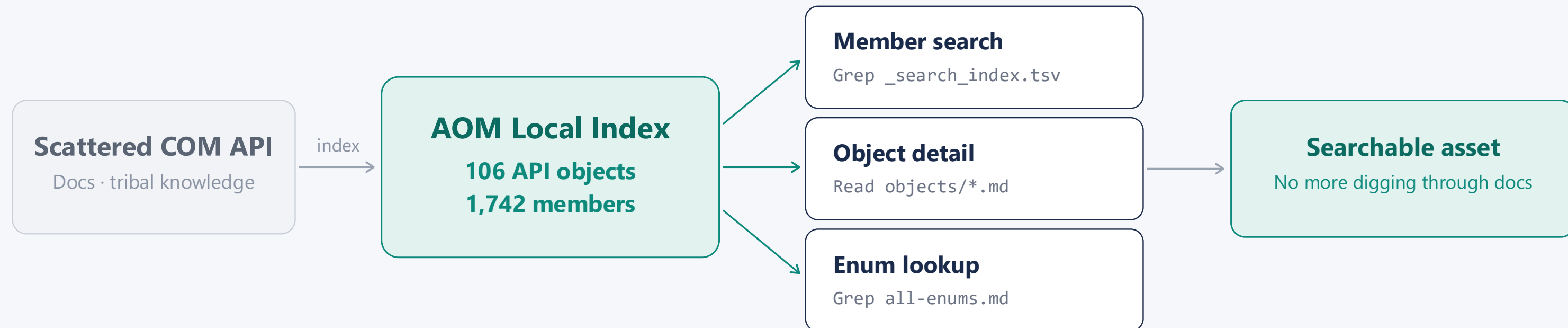
First, Turn the Expert's Mind into a Structured Asset

STAGE 1 · Assetization

01

AOM Local Index

106 API objects · 1,742 members, turned into a searchable index. No more humans digging through docs.



First, Turn the Expert's Mind into a Structured Asset

STAGE 1 · Assetization

02

Pattern Library

Dimensions · traces · spacing · vias · BGA · components... verification scripts by category.

ID scheme

G-NNN Gotcha

E-NNN Error

P-NNN Pattern

Entry schema

AOM · Related object

Severity · critical / important / basic

Context · Problem · Solution

Example · python snippet

Title · date (YYYY-MM-DD)

register

pattern.md

Summary table

by-aom/{Cat}.md

Detailed patterns

_aom_index.tsv

Grep lookup

Grep `"^{AOM}\t" _aom_index.tsv` — find patterns instantly by object name

First, Turn the Expert's Mind into a Structured Asset

STAGE 1 · Assetization

03

Knowledge Base

Gotchas · errors · patterns, accumulated on a per-object basis..

knowledge/

_aom_index.tsv

Grep-optimized search index

gotchas / errors / patterns .md

Summary tables

by-AOM/{Category}.md

Core · Geometry · ViolationRule · NetTrace
ComponentPin · BoardLayer · Via · Collection ...

scripts/rebuild_index.py

by-AOM/*.md → _aom_index.tsv

Organized per AOM object

READ · before using an AOM

Grep "**^{Object}\t**" _aom_index.tsv
→ Read by-AOM/{Category}.md

WRITE · after a fix or finding

1. Add detail → by-AOM/{Category}.md

2. Add row → summary table

3. Run rebuild_index.py

Step 3 regenerates the search index — the loop that keeps knowledge queryable

First, Turn the Expert's Mind into a Structured Asset

STAGE 1 · Assetization

01

AOM Local Index

02

Pattern Library

03

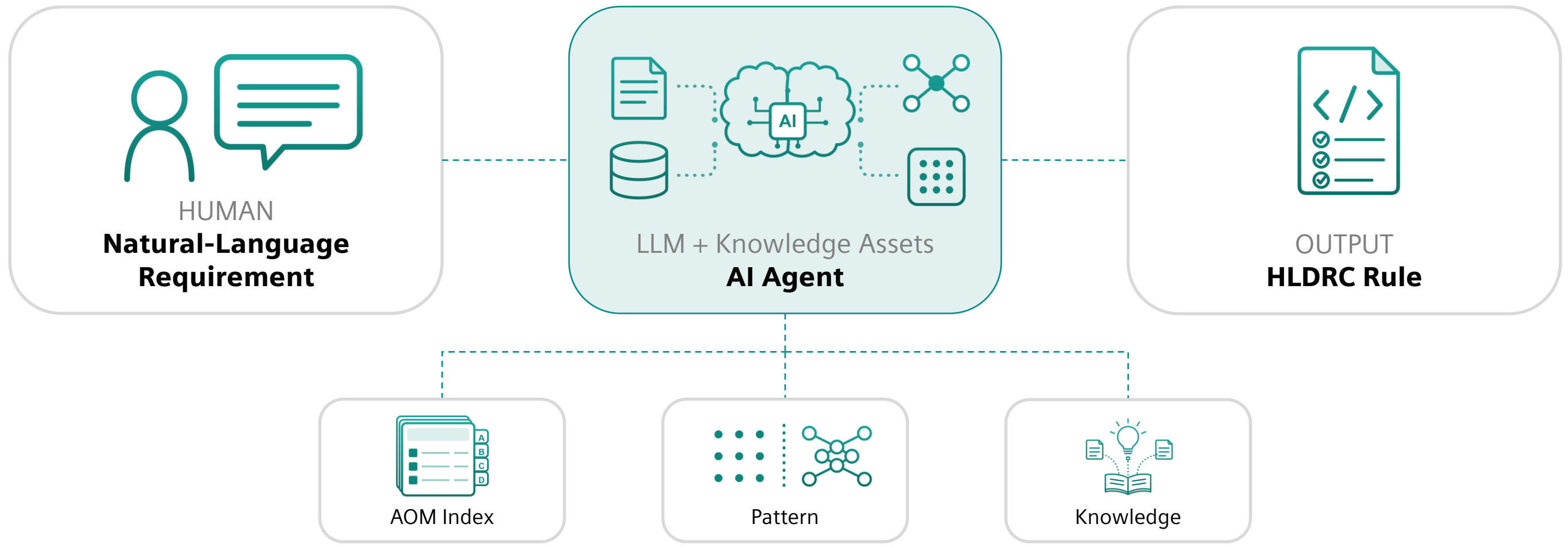
Knowledge Base



...but up to here, a **human still has to assemble it**. One more step to go.

Natural Language Goes In, a Verification Rule Comes Out

STAGE 2 · AI



Expected Benefits — 4 of them, demonstrated in real-world cases

- Faster Turnaround
- No Expert Bottleneck
- Barrier Removed
- Rule Quality & Accuracy

It doesn't stop at generation — **testing · explanation · debugging · reporting** all flow as one continuous process.

One Workflow, Six Skills

STAGE 2 · Tools

/hldrc-rule

Natural Language →

Rule generation · editing · review

/hldrc-api-lookup

AOM index-based API lookup

/hldrc-test-rule

Live COM execution · auto-fix · report

/hldrc-debug

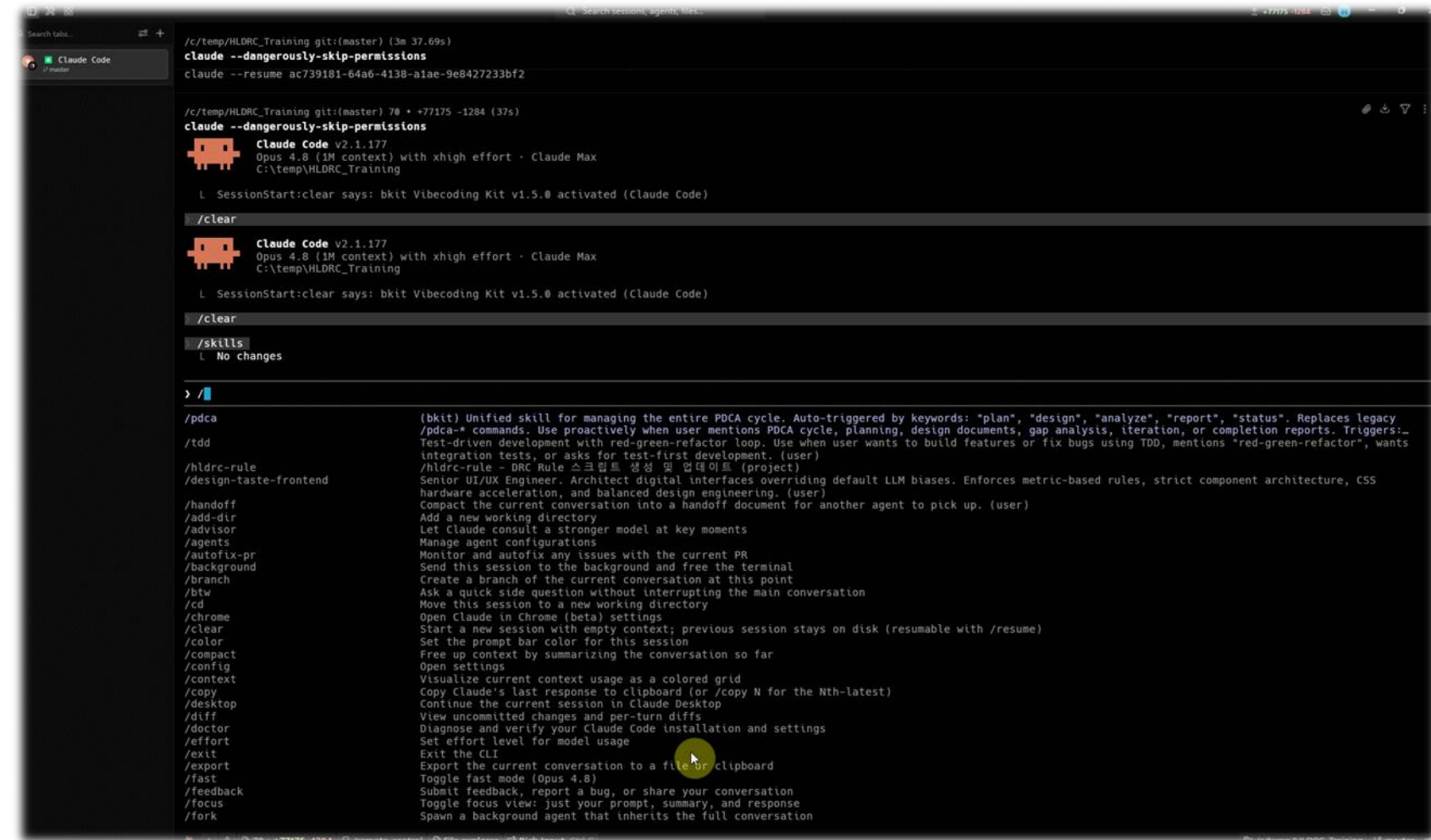
Error diagnosis · debugging (knowledge)

/hldrc-explain

Analyze existing rules · Mermaid flowchart

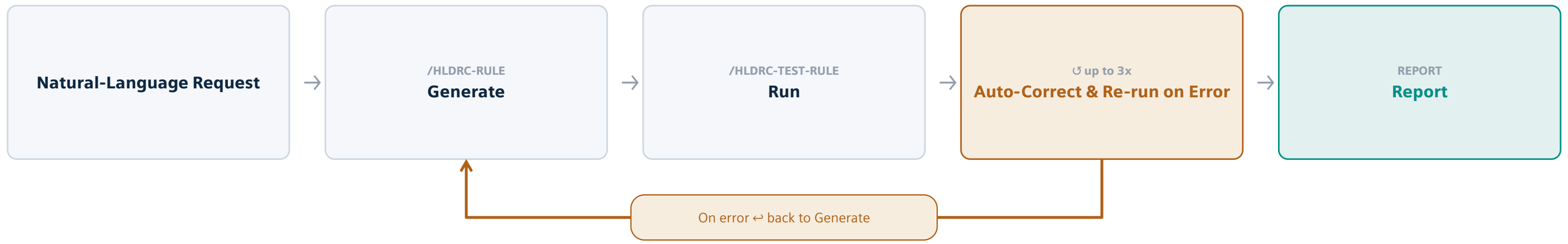
/hldrc-pattern

Pattern library search · recommendation



Generate → Run → Self-Correct → Report

STAGE 2 · END-TO-END



Why It Matters:

- Not “build and done” – **validated by running on a real design**
- Even on failure, it **fixes itself** using knowledge
- Results remain as a **shareable report**
- The **expert sign-off gate** before deployment stays in place

Three Real-World Cases

REAL CASES

04

How the system solves the exact points where engineers actually **got stuck**.

All real cases, real deliverables – work that **even experts had to dig through the API for, every time**, now compressed into natural language.

UC1 · Large-Scale Design Crop

UC2 · Same-Net Cross

UC3 · Explaining Existing Rules

Large-Scale Design — Crop / Rule Area Selection

No Expert Bottleneck

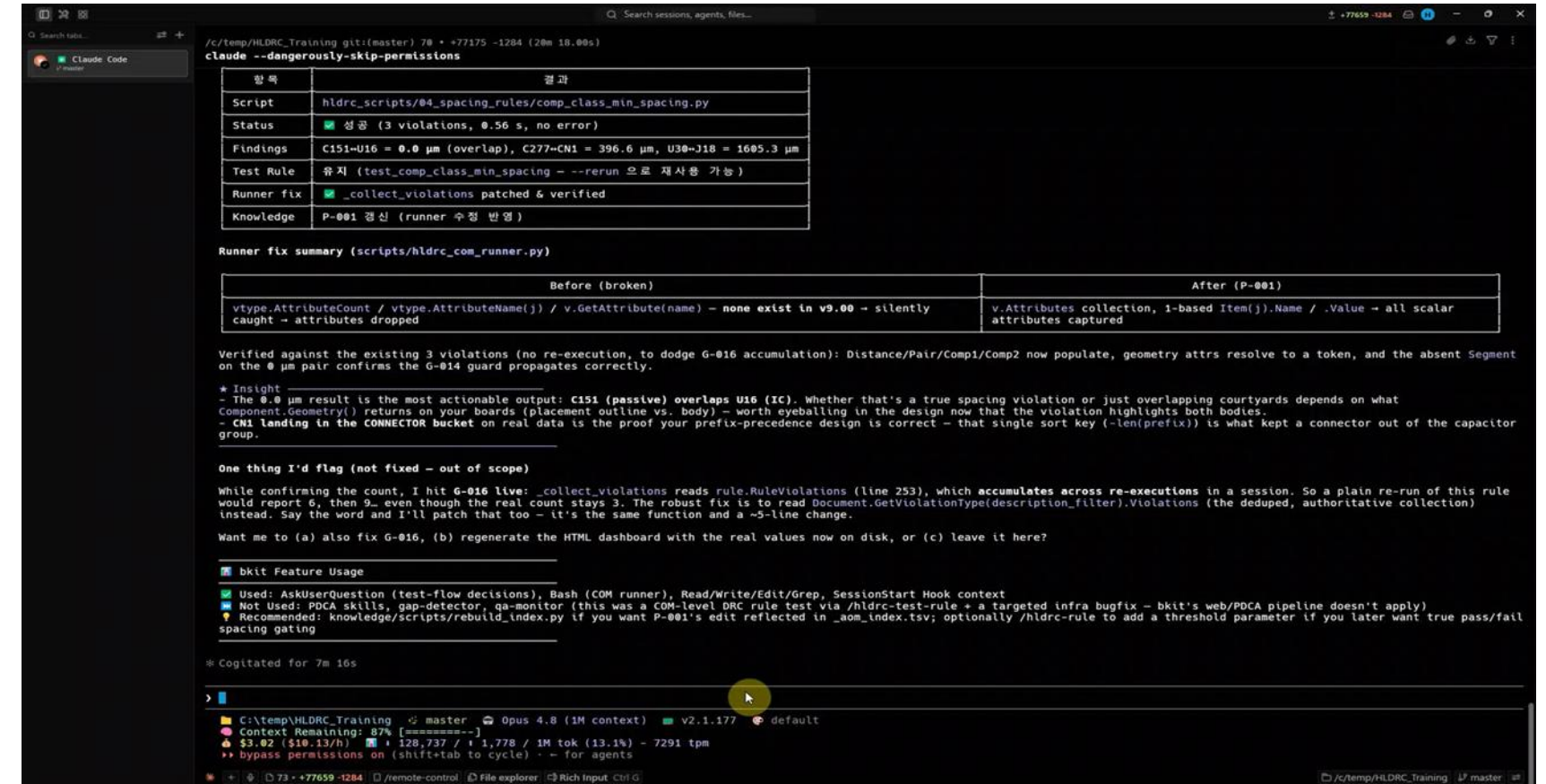
Entry Barrier Removed

The Sticking Point

“I want to crop just part of a huge design, or define a Rule Area to verify – but I don’t know how to handle it”

Here’s How AI Does It

- Expand selected objects by a **Margin**
- Auto-generate the **boundary**
- Save just the cropped region with **SaveAreaCrop**



~13MB

Cropped output design (.hldproj) – only the verification scope, separated out

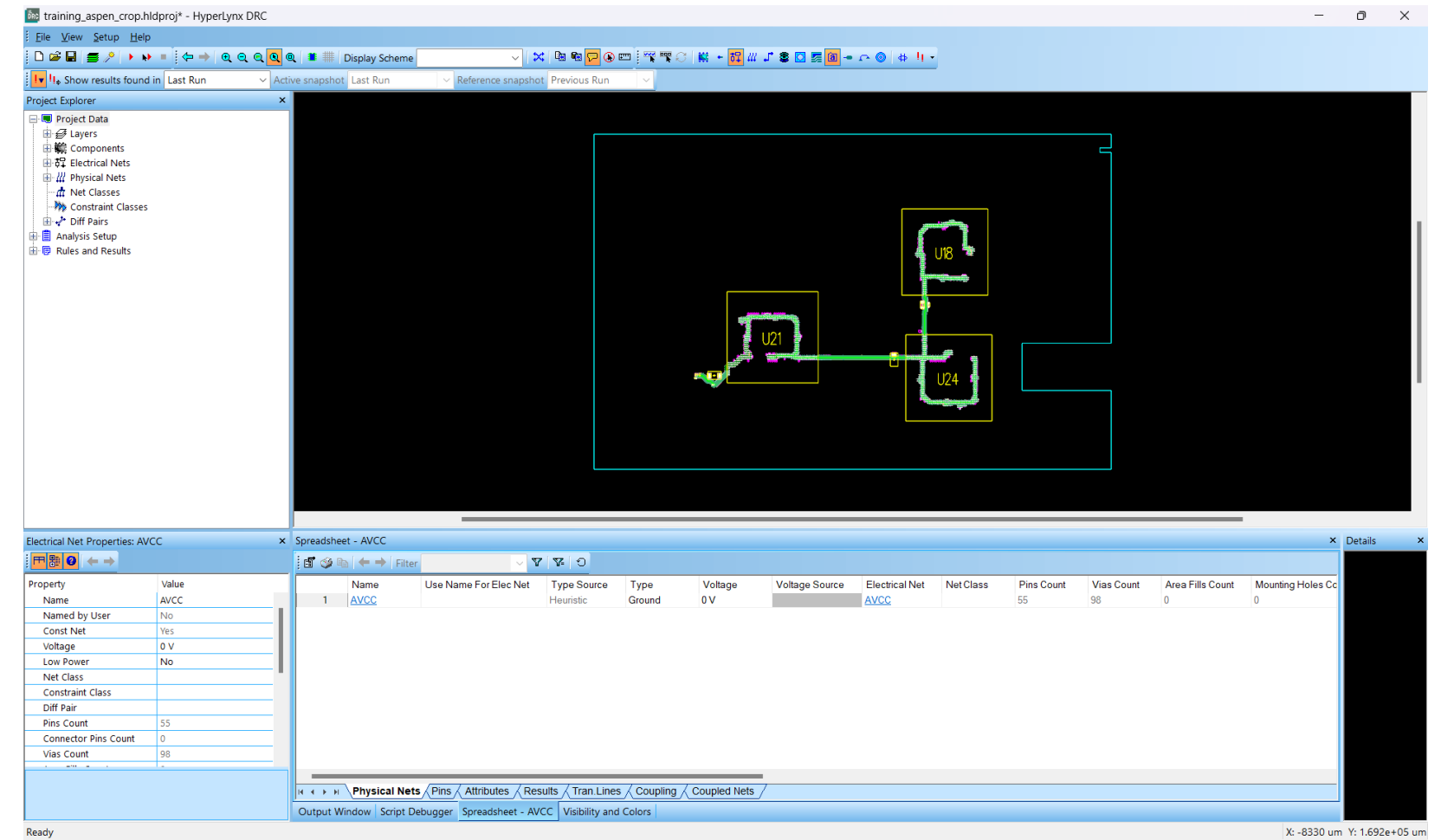
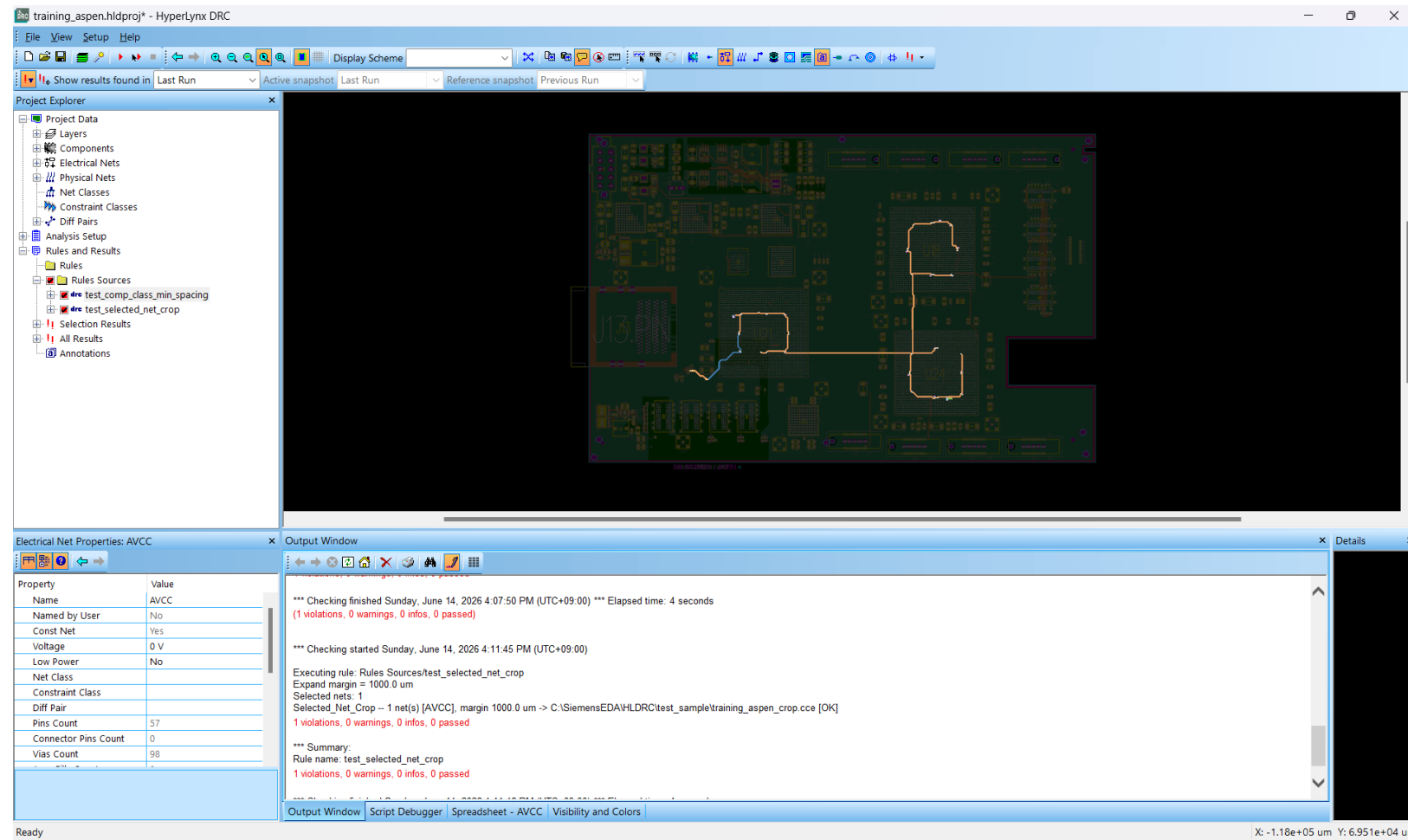
One Shot

Fully automated from selection to crop-save

Large-Scale Design — Crop / Rule Area Selection

No Expert Bottleneck

Entry Barrier Removed



Same-Net Clearance — Cross Method

Fater Turnaround

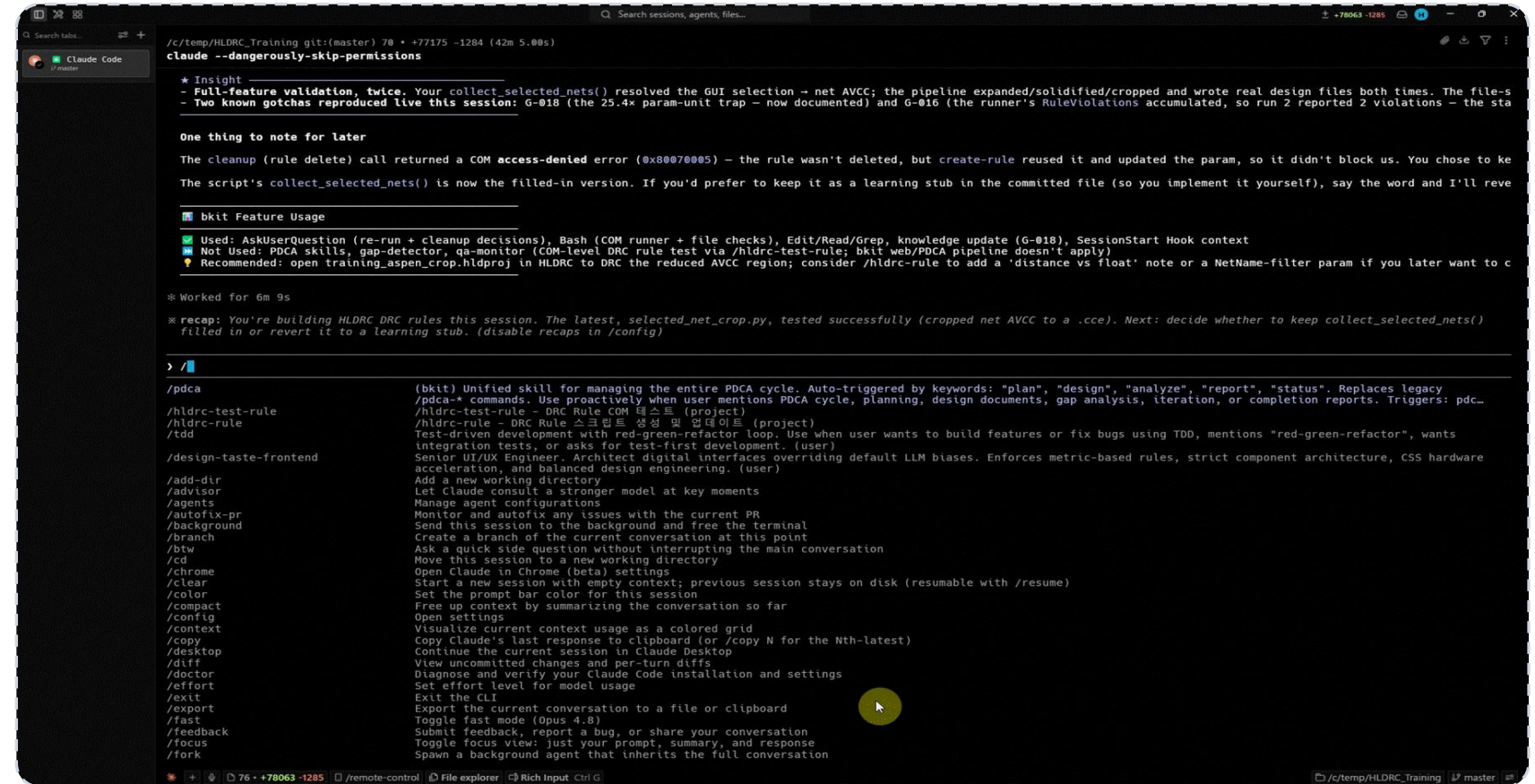
Rule Quality & Accuracy

The Sticking Point

“The Cross method seems useful for Same-Net Clearance checks, but I don’t know how to use it.”

Here’s How AI Does It

- Compute the intersection region with **GeomEngine.Cross()**
- Branch on intersection count + Net.Name comparison
- Auto-handle the empty-collection **None+Count double-check gotcha**



Cross()

The core geometry operation

0 Inquiries

Written with zero expert consultation

Same-Net Clearance — Cross Method

Faster Turnaround

Rule Quality & Accuracy

The Sticking Point

“The Cross method seems useful for Same-Net Clearance checks, but I don’t know how to use it.”

Here’s How AI Does It

- Compute the intersection region with **GeomEngine.Cross()**
- Branch on intersection count + Net.Name comparison
- Auto-handle the empty-collection **None+Count double-check gotcha**

Cross Method



- Expanded Via Geometries
- Via Geometries

Same-Net Clearance — Cross Method

Faster Turnaround

Rule Quality & Accuracy

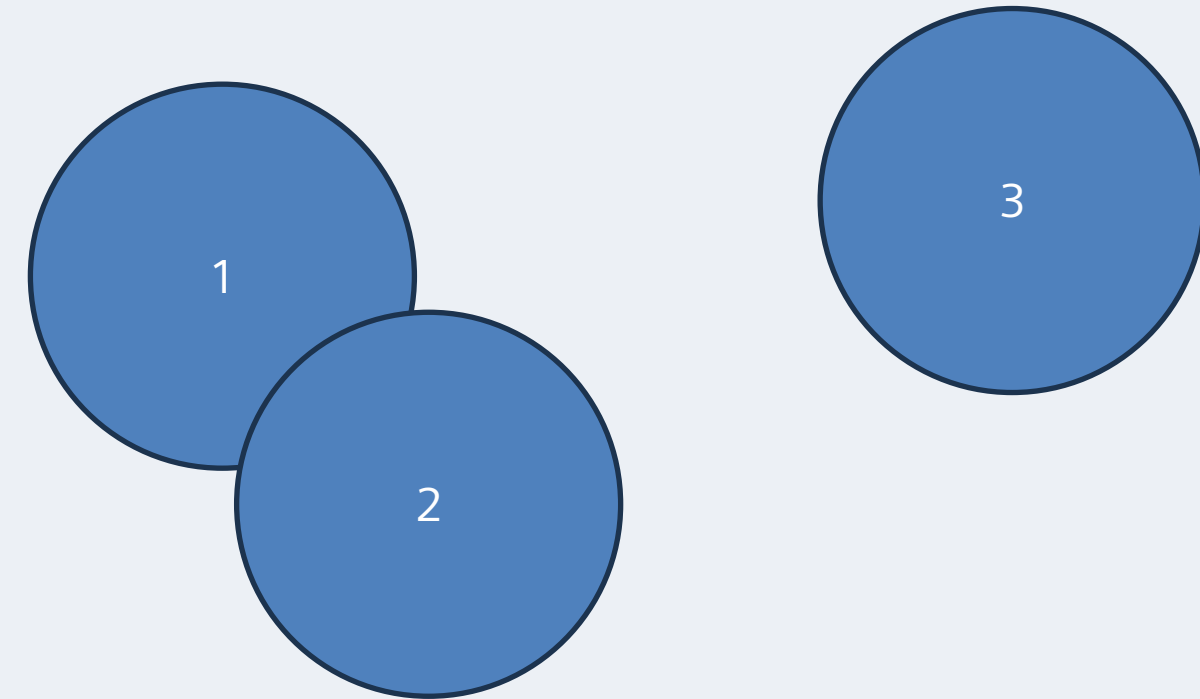
The Sticking Point

“The Cross method seems useful for Same-Net Clearance checks, but I don’t know how to use it.”

Here’s How AI Does It

- Compute the intersection region with **GeomEngine.Cross()**
- Branch on intersection count + Net.Name comparison
- Auto-handle the empty-collection **None+Count double-check gotcha**

Cross Method



- Expanded Via Geometries
- Via Geometries

Same-Net Clearance — Cross Method

Faster Turnaround

Rule Quality & Accuracy

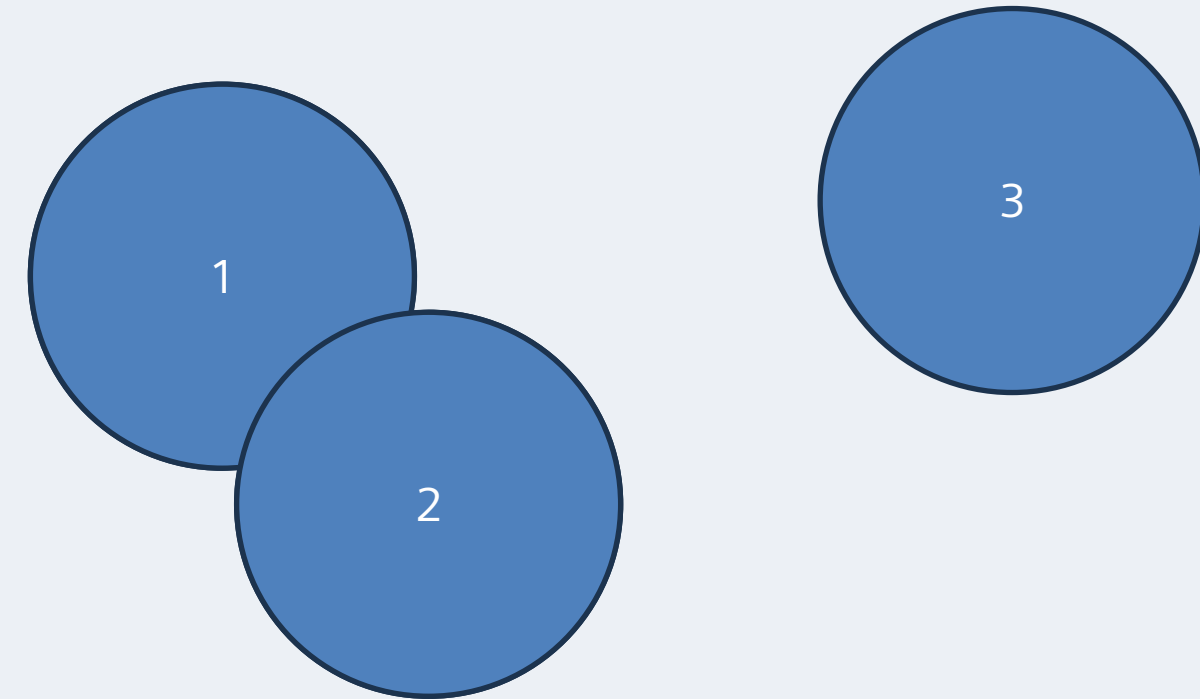
The Sticking Point

“The Cross method seems useful for Same-Net Clearance checks, but I don’t know how to use it.”

Here’s How AI Does It

- Compute the intersection region with **GeomEngine.Cross()**
- Branch on intersection count + Net.Name comparison
- Auto-handle the empty-collection **None+Count double-check gotcha**

Cross Method



- Expanded Via Geometries
- Via Geometries

Same-Net Clearance — Cross Method

Faster Turnaround

Rule Quality & Accuracy

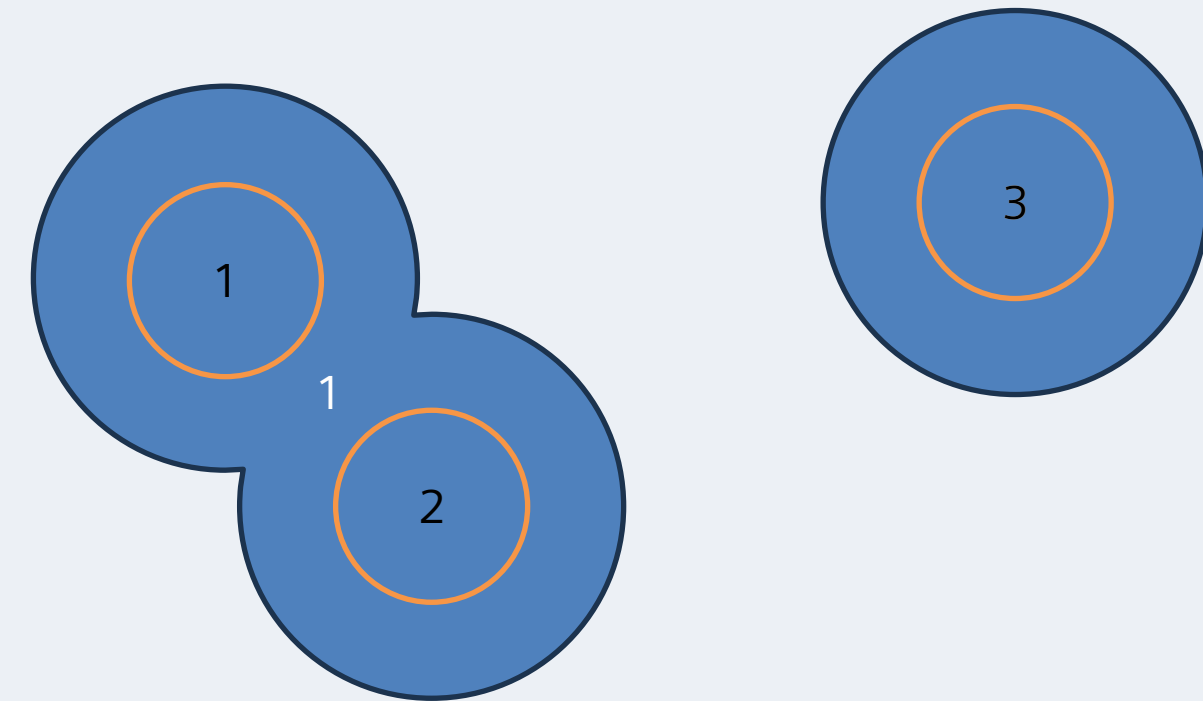
The Sticking Point

“The Cross method seems useful for Same-Net Clearance checks, but I don’t know how to use it.”

Here’s How AI Does It

- Compute the intersection region with **GeomEngine.Cross()**
- Branch on intersection count + Net.Name comparison
- Auto-handle the empty-collection **None+Count double-check gotcha**

Cross Method



- Expanded Via Geometries
- Via Geometries

Same-Net Clearance — Cross Method

Faster Turnaround

Rule Quality & Accuracy

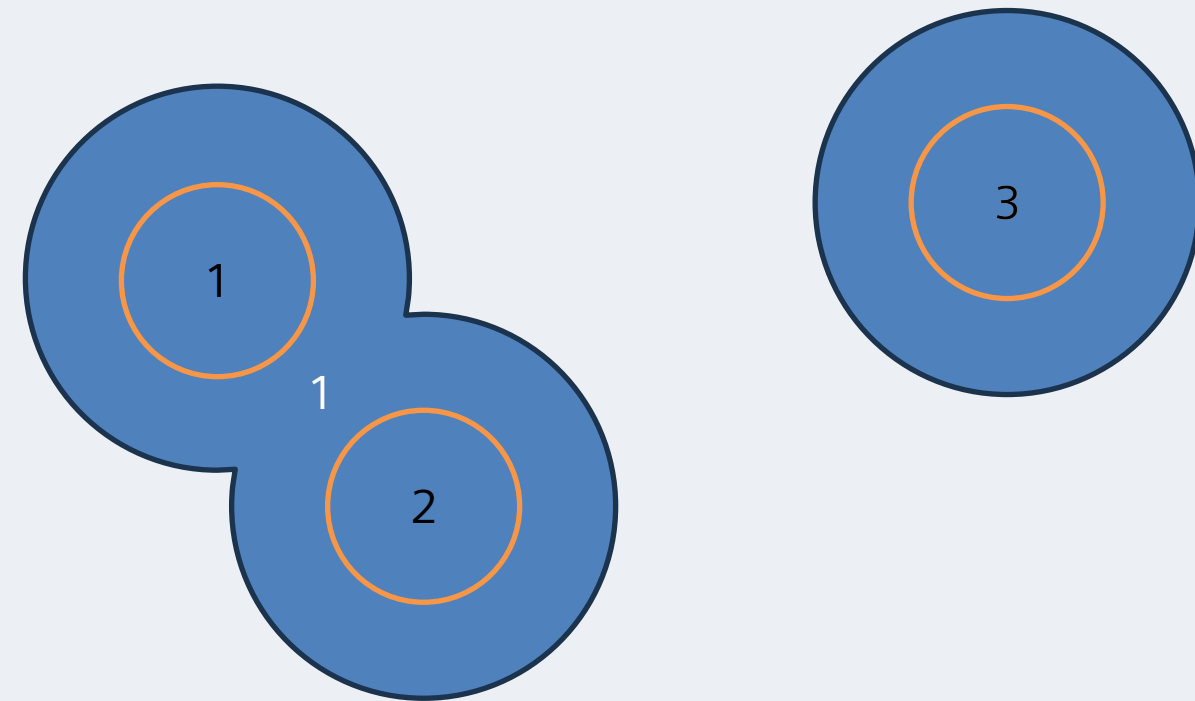
The Sticking Point

“The Cross method seems useful for Same-Net Clearance checks, but I don’t know how to use it.”

Here’s How AI Does It

- Compute the intersection region with **GeomEngine.Cross()**
- Branch on intersection count + Net.Name comparison
- Auto-handle the empty-collection **None+Count double-check gotcha**

Cross Method



	Array column index		
	0	1	2
Index of collection expanded via geometries	1	1	2
Index of collection via geometries	1	2	3

- Expanded Via Geometries
- Via Geometries

Same-Net Clearance — Cross Method

Faster Turnaround

Rule Quality & Accuracy

The Sticking Point

“The Cross method seems useful for Same-Net Clearance checks, but I don’t know how to use it.”

Here’s How AI Does It

- Compute the intersection region with **GeomEngine.Cross()**
- Branch on intersection count + Net.Name comparison
- Auto-handle the empty-collection **None+Count double-check gotcha**

Cross Method

Array column index	0	1	2
Index of collection expanded via geometries	1	1	2
Index of collection via geometries	1	2	3

Index of collection expanded via geometries	0	1	2
# of intersections	Not Used	2	1

- Expanded Via Geometries
- Via Geometries

Same-Net Clearance — Cross Method

Faster Turnaround

Rule Quality & Accuracy

The Sticking Point

“The Cross method seems useful for Same-Net Clearance checks, but I don’t know how to use it.”

Here’s How AI Does It

- Compute the intersection region with **GeomEngine.Cross()**
- Branch on intersection count + Net.Name comparison
- Auto-handle the empty-collection **None+Count double-check gotcha**

Cross Method

Array column index	0	1	2
Index of collection expanded via geometries	1	1	2
Index of collection via geometries	1	2	3

Index of collection expanded via geometries	0	1	2
# of intersections	Not Used	2	1

- Expanded Via Geometries
- Via Geometries

Same-Net Clearance — Cross Method

Faster Turnaround

Rule Quality & Accuracy

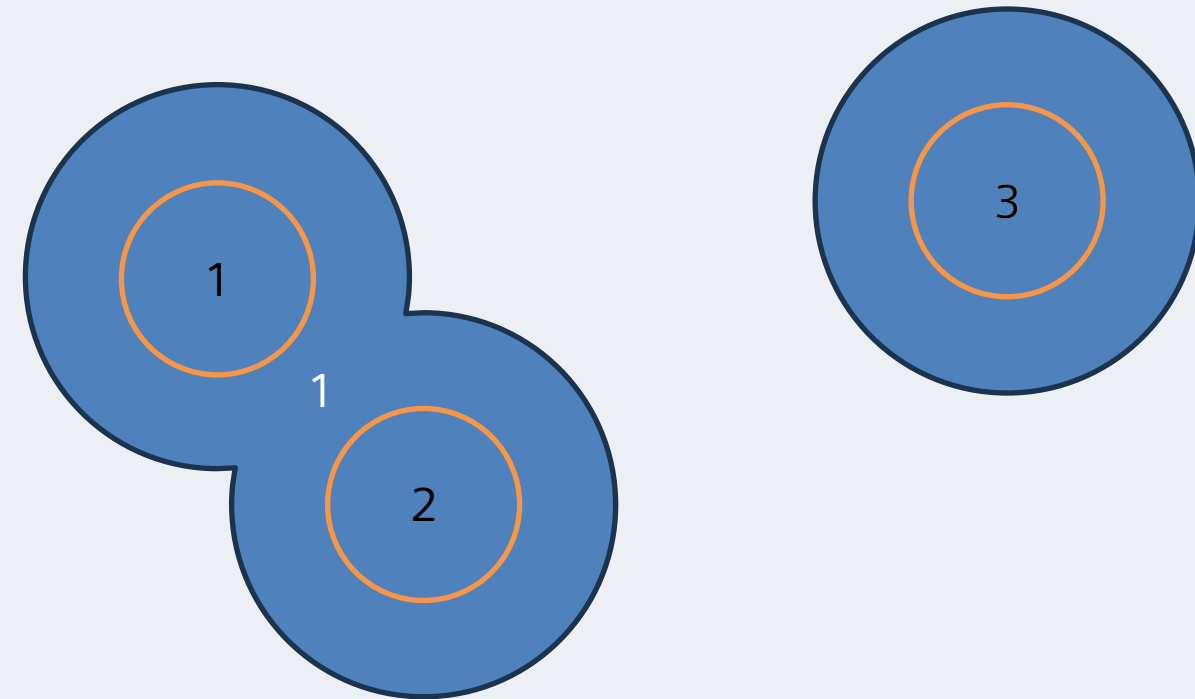
The Sticking Point

“The Cross method seems useful for Same-Net Clearance checks, but I don’t know how to use it.”

Here’s How AI Does It

- Compute the intersection region with **GeomEngine.Cross()**
- Branch on intersection count + Net.Name comparison
- Auto-handle the empty-collection **None+Count double-check gotcha**

Cross Method



	Array column index		
	0	1	2
Index of collection expanded via geometries	1	1	2
Index of collection via geometries	1	2	3

- Expanded Via Geometries
- Via Geometries

Same-Net Clearance — Cross Method

Faster Turnaround

Rule Quality & Accuracy

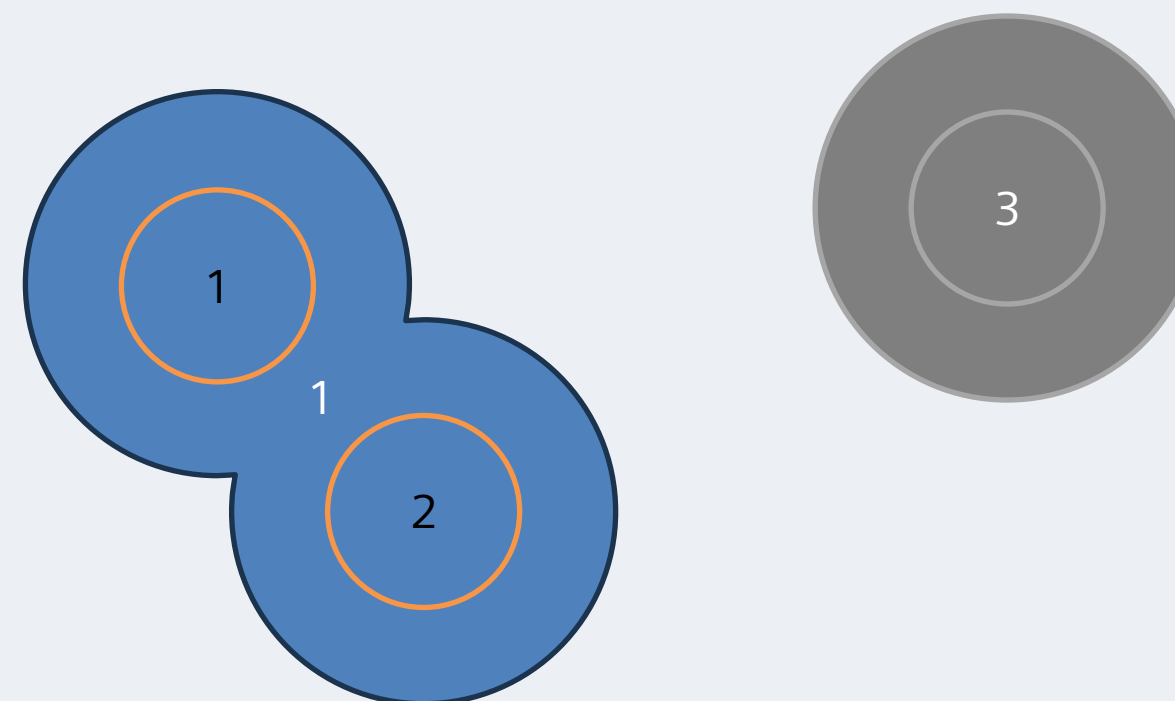
The Sticking Point

“The Cross method seems useful for Same-Net Clearance checks, but I don’t know how to use it.”

Here’s How AI Does It

- Compute the intersection region with **GeomEngine.Cross()**
- Branch on intersection count + Net.Name comparison
- Auto-handle the empty-collection **None+Count double-check gotcha**

Cross Method



	Array column index		
	0	1	2
Index of collection expanded via geometries	1	1	2
Index of collection via geometries	1	2	3

- Expanded Via Geometries
- Via Geometries

Same-Net Clearance — Cross Method

Faster Turnaround

Rule Quality & Accuracy

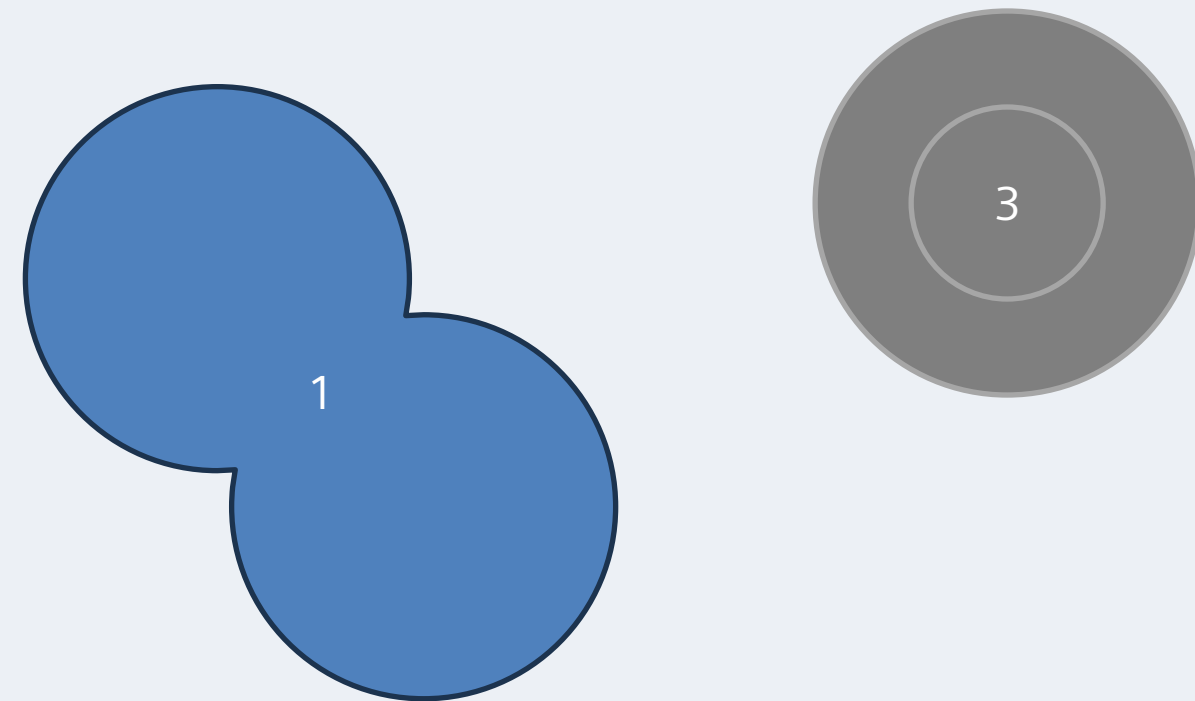
The Sticking Point

“The Cross method seems useful for Same-Net Clearance checks, but I don’t know how to use it.”

Here’s How AI Does It

- Compute the intersection region with **GeomEngine.Cross()**
- Branch on intersection count + Net.Name comparison
- Auto-handle the empty-collection **None+Count double-check gotcha**

Cross Method



	Array column index		
	0	1	2
Index of collection expanded via geometries	1	1	2
Index of collection via geometries	1	2	3

- Expanded Via Geometries
- Via Geometries

Same-Net Clearance — Cross Method

Faster Turnaround

Rule Quality & Accuracy

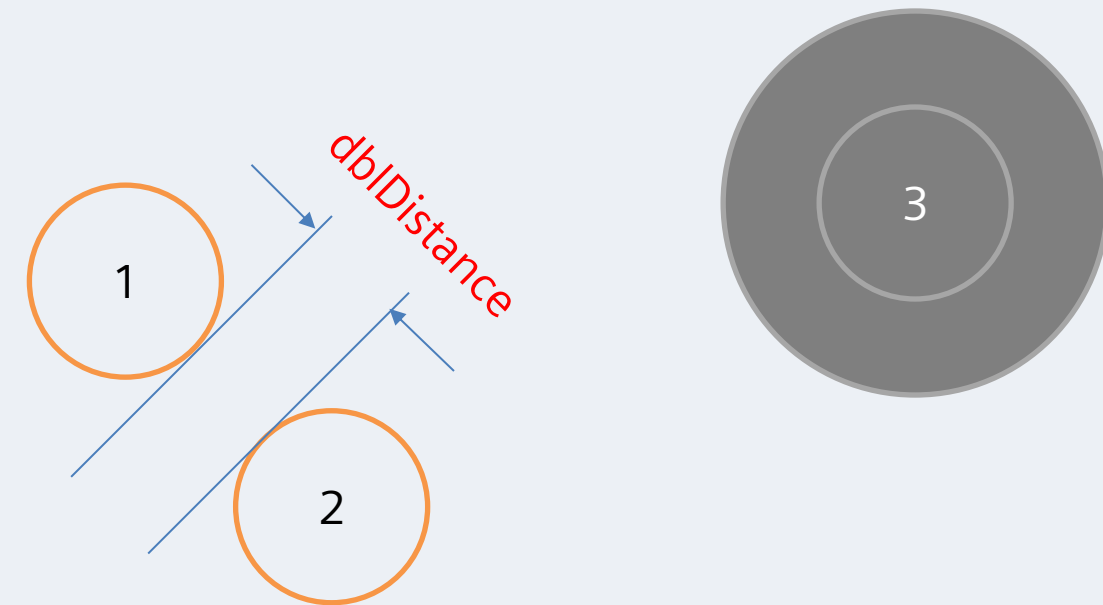
The Sticking Point

“The Cross method seems useful for Same-Net Clearance checks, but I don’t know how to use it.”

Here’s How AI Does It

- Compute the intersection region with **GeomEngine.Cross()**
- Branch on intersection count + Net.Name comparison
- Auto-handle the empty-collection **None+Count double-check gotcha**

Cross Method



	0	1	2
Array column index			
Index of collection expanded via geometries	1	1	2
Index of collection via geometries	1	2	3

- Expanded Via Geometries
- Via Geometries

Same-Net Clearance — Cross Method

Faster Turnaround

Rule Quality & Accuracy

The Sticking Point

“The Cross method seems useful for Same-Net Clearance checks, but I don’t know how to use it.”

Here’s How AI Does It

- Compute the intersection region with **GeomEngine.Cross()**
- Branch on intersection count + Net.Name comparison
- Auto-handle the empty-collection **None+Count double-check gotcha**

```

Claude Code
~/temp/HLDRG_Training gti(master) 76 • +77375 -1204 (426 5.00s)
claude --dangerously-skip-permissions

* Insight
- Full-feature validation, twice. Your collect_selected_nets() resolved the GUI selection - net AVCC; the pipeline expanded/solidified/cropped and wrote real design files both times. The files
Two known gotchas reproduced live this session: 0-010 (the 25.4x param-unit trap - now documented) and 0-016 (the runner's RuleViolations accumulated, so run 2 reported 2 violations - the sta

One thing to note for later
The cleanup (rule delete) call returned a COM access-denied error (@x80870005) - the rule wasn't deleted, but create-rule reused it and updated the param, so it didn't block us. You chose to ke
The script's collect_selected_nets() is now the filled-in version. If you'd prefer to keep it as a learning stub in the committed file (so you implement it yourself), say the word and I'll reve

bkit Feature Usage
Used: AskUserQuestion (re-run + cleanup decisions), Bash (COM runner + file checks), Edit/Read/Grab, Knowledge update (0-010), SessionStart Hook context
Not Used: PDCA skills, gap-detector, qa-monitor (COM-level DRC rule test via /hldrc-test-rule; bkit web/PDCA pipeline doesn't apply)
Recommended: open training_aspen_crop.hldproj in HLDRG to DRC the reduced AVCC region; consider /hldrc-rule to add a 'distance vs float' note or a NetName-filter param if you later want to c

* Worked for 6m 9s
* recap: You're building HLDRG DRC rules this session. The latest, selected_net_crop.py, tested successfully (cropped net AVCC to a .cce). Next: decide whether to keep collect_selected_nets()
filled in or revert it to a learning stub. (disable recaps in /config)

> /

/pdca (bkit) Unified skill for managing the entire PDCA cycle. Auto-triggered by keywords: "plan", "design", "analyze", "report", "status". Replaces legacy
/pdca+ commands. Use proactively when user mentions PDCA cycle, planning, design documents, gap analysis, iteration, or completion reports. Triggers: pdc-
/hldrc-test-rule - DRC Rule COM 테스트 (project)
/hldrc-rule - DRC Rule 스크립트 생성 및 업데이트 (project)
/tdd Test-driven development with red-green-refactor loop. Use when user wants to build features or fix bugs using TDD, mentions "red-green-refactor", wants
integration tests, or asks for test-first development. (user)
/design-taste-frontend Senior UI/UX Engineer. Architect digital interfaces overriding default LLM biases. Enforces metric-based rules, strict component architecture, CSS hardware
acceleration, and balanced design engineering. (user)
/add-dir Add a new working directory
/advisor Let Claude consult a stronger model at key moments
/agents Manage agent configurations
/autofix-pr Monitor and autofix any issues with the current PR
/background Send this session to the background and free the terminal
/branch Create a branch of the current conversation at this point
/bw Ask a quick side question without interrupting the main conversation
/cd Move this session to a new working directory
/chrome Open Claude in Chrome (beta) settings
/clear Start a new session with empty context; previous session stays on disk (resumable with /resume)
/color Set the prompt bar color for this session
/compact Free up context by summarizing the conversation so far
/config Open settings
/context Visualize current context usage as a colored grid
/copy Copy Claude's last response to clipboard (or /copy N for the Nth-latest)
/desktop Continue the current session in Claude Desktop
/diff View uncommitted changes and per-ture diffs
/doctor Diagnose and verify your Claude Code installation and settings
/effort Set effort level for model usage
/exit Exit the CLI
/export Export the current conversation to a file or clipboard
/fast Toggle fast mode (Opus 4.8)
/feedback Submit feedback, report a bug, or share your conversation
/focus Toggle focus view: just your prompt, summary, and response
/fork Spawn a background agent that inherits the full conversation
  
```

Cross()

The core geometry operation

0 Inquiries

Written with zero expert consultation

Instantly Understand Existing Rules — Onboarding & Handover

Anyone Can Do It

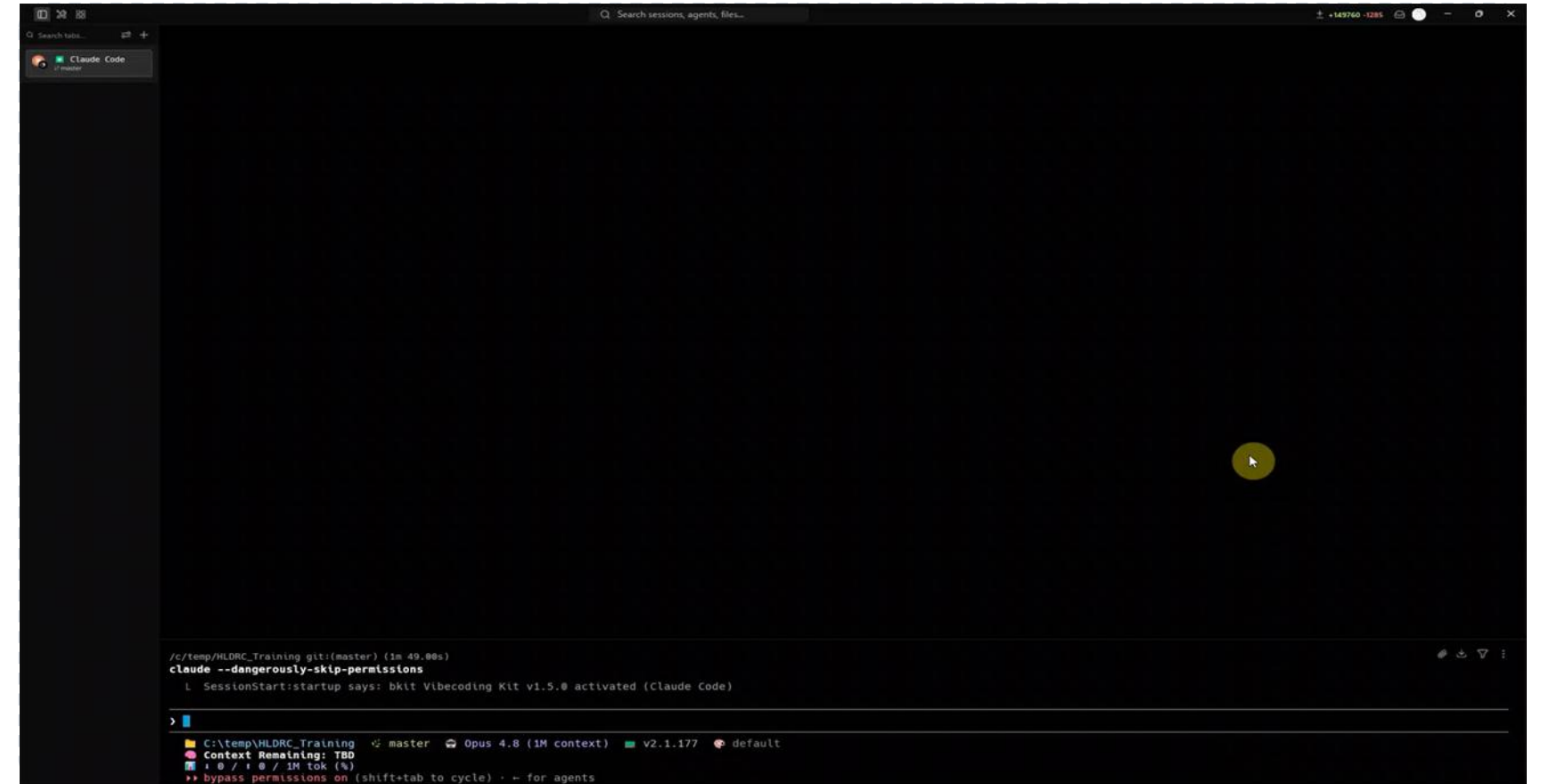
No Expert Bottleneck

The Sticking Point

“I inherited someone else’s complex rule, but I have no idea what or why it’s checking – so I can’t touch it.”

Here’s How AI Does It

- One-line summary + code-section breakdown via **/hldrc-explain**
- Trace & explain **every COM API used**, through the AOM index
- Auto-generate a **Mermaid flowchart** – the logic at a glance



1,742

Member API index for automatic tracing & explanation

Minutes

Grasp even a rule you’ve never seen – no expert handoff

Four Things Move at Once

IMPACT

Faster Turnaround

API exploration and trial-and-error disappear – the entire write-verify cycle gets shorter.

No Expert Bottleneck

Decoupled from the expert's schedule, which used to be the bottleneck. Build it the moment you need it.

Entry Barrier Removed

Author in natural language with no API knowledge – the role expands from “execute & review” to “create.”

Rule Quality & Accuracy

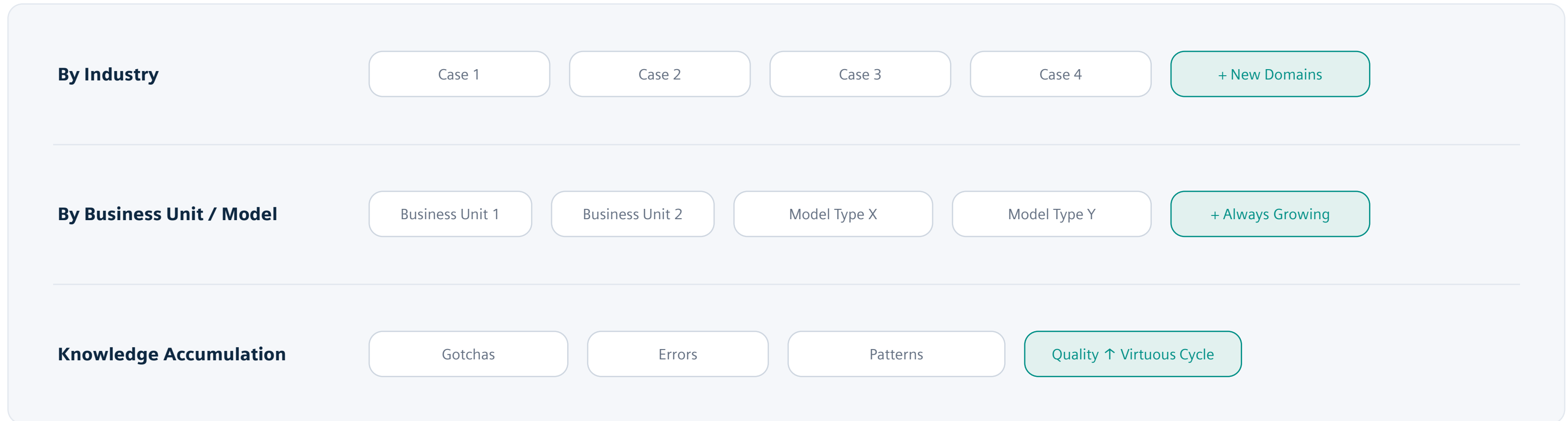
Avoids knowledge gotchas + verified by running on real designs – author-to-author variance shrinks.

The four things promised in Phase 2 – **confirmed, exactly as stated, across all three cases.**

Deployment Isn't the End – It's the Beginning

— EVOLVING

Knowledge doesn't stop at customer deployment. It keeps accumulating and evolving.



The more gotchas and patterns accumulate, the higher the **AI generation quality** climbs → a system that gets smarter the more you use it.

Experts Don't Disappear – Their Role Changes

REDEFINITION

BEFORE — Rule Author

- Hand-codes every request
- The cause of the schedule bottleneck
- Knowledge trapped in individuals

AFTER — Curator · Validator · Direction-Setter

- **Curates** knowledge assets & patterns
- **Validates & approves** AI deliverables
- **Sets the direction** on hard problems & standards

Every engineer's verification quality **expands**, and the expert becomes **leverage**.

This Is Just the Beginning

WHAT'S NEXT

01

Enterprise System Integration

Connect with PLM · constraint DBs · design standards

02

Platform Integration

Embed rule automation into verification pipelines / CI

03

The Full Verification Flow

Expand beyond DRC to the entire PCB verification process

“The age of aviation began with 12 seconds. For PCB verification, that 12 seconds is **now** — and we’ve **already taken off.**”

From Experts to Everyone

THANK YOU

Three Takeaways

- Natural language → rule, end-to-end, **actually works**
- The **three field challenges** that used to block you are solved
- Knowledge **keeps evolving**

Want to Learn More?

If you're curious about this approach (AI-assisted DRC verification), feel free to reach out by email – I'd be glad to help you one-on-one.

Hyunjae.kim@siemens.com