AU2026

THE DESIGN & MAKE CONFERENCE

September 15–17, 2026 | Las Vegas | Digital

From Data to Decisions:

Accelerating Semiconductor Fab Planning with AI-Powered BIM Automation in Revit

Yoonkyeong Seon · Samsung Electronics

Tommy Lee · Samsung Electronics

Wonho Cho · Gachon University

2026-09-16

Opening — from data to decisions: last year's promise, running this year.

Who We Are

Samsung Electronics × Gachon University

Yoonkyeong Seon

Design Productivity Engineer

Samsung Electronics

Specializes in semiconductor facility design optimization. Uses AI and Generative Design to optimize room layouts and routing, reducing material cost and improving project management efficiency.

Tommy Lee

System Engineer

Samsung Electronics

Specializes in digital systems and construction data integration, and in applying AI and automation to enhance construction processes.

Wonho Cho

Generative Designer

Gachon University

Specializes in BIM automation for semiconductor facilities — from existing-fab data extraction to generative layouts and verified local-AI review.

Three of us — two Samsung engineers and a generative designer. One pipeline, from the existing fab to the next one.

Introduction and Executive Summary

#AU2026

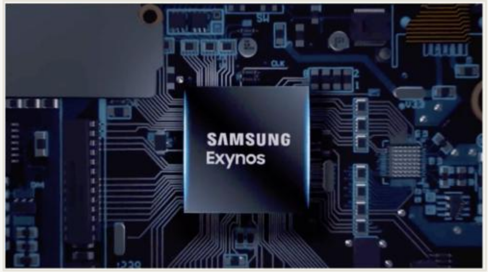
Samsung Semiconductor

Device Solution division, its FABs and its position in the market

FABS

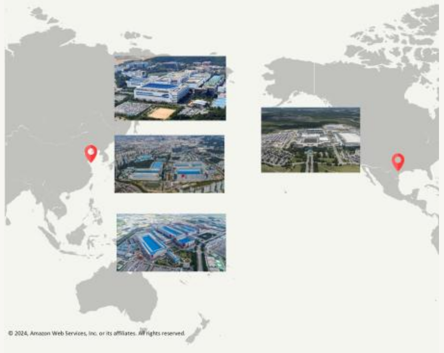
Device Solution (DS) division has been producing top tier semiconductor products for electronic devices (such as smart phones, home appliances, and data centers) since 1983

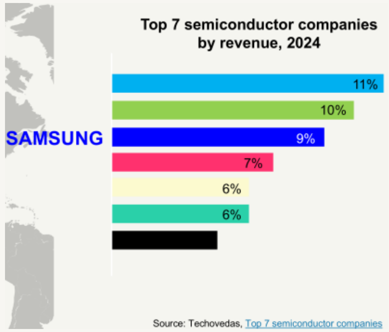
PRODUCTS



V-NAND memory · Exynos SoC

MARKET POSITION





With large-scale semiconductor production in multiple FABs, Samsung has been acting as a major chip supplier in the semiconductor industry

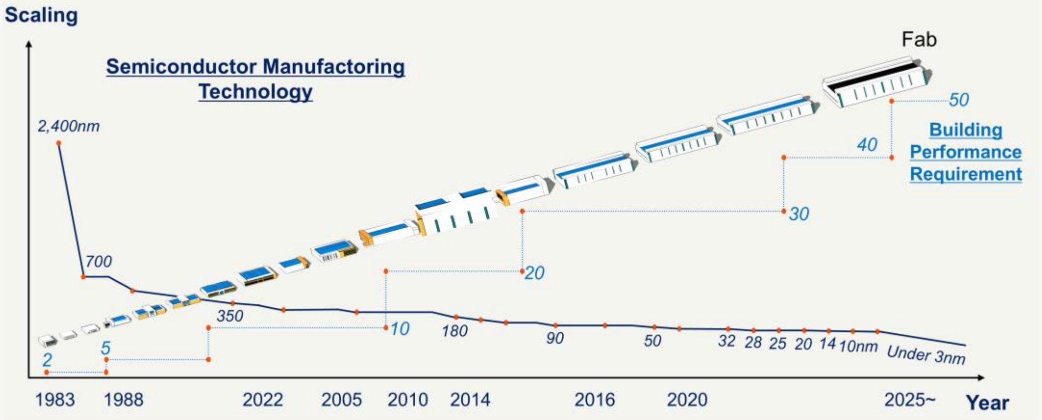
FAB construction characteristics

Why a FAB is a hard building — and how the requirement has moved

COMPLEXITY

Burj Khalifa, Dubai · DDP, Seoul · Samsung FAB, Korea

TECHNOLOGY GENERATION



Manufacturing technology node vs. building performance requirement

Design and building of FABs is significantly complex, as it is for other large-scale construction project types (such as high-rise buildings or irregularly shaped buildings)

The size and requirements of FABs have been changed with the development of technology

Samsung's FAB project and its flow

The constraints it works under, the tools applied, and the delivery sequence

CHALLENGES

Fast-track & Frequent Change

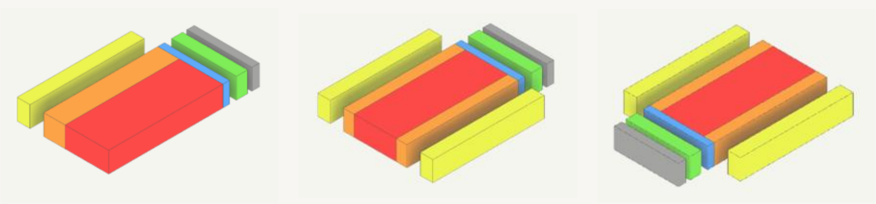
Concurrent design and
construction phases

Security

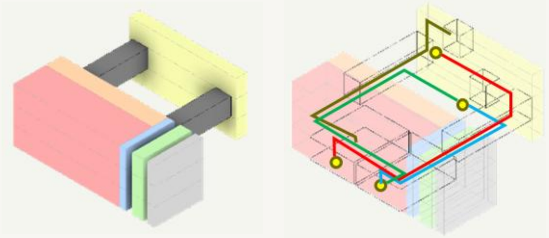
Operated as a national security project,
necessitating strigent security measures

TOOLS APPLIED

Generative Design



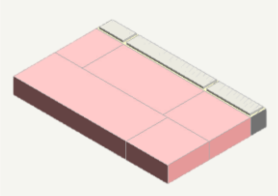
Autodesk Tandem



DELIVERY FLOW

1 GD Layout Creation

Automatically generate facility layouts based on design constraints
Explore multiple alternatives efficiently
Layout sketch → GD algorithm diagram

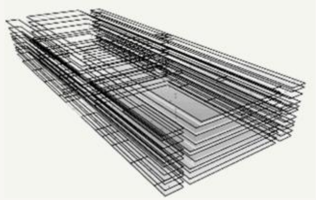


2 Model Generation

Automated routing of pipes, ducts, and cables
BIM model generation with minimized clashes
Auto-routed piping/duct 3D model snapshot

3 Assets Management

Connect BIM model into Tandem Digital Twin
Manage assets with properties, maintenance history, and operational data
Ensure continuity from design to operations



4 Review & AFD

Conduct a comprehensive review using Digital Twin insights
Proceed with AFD (Approved for Design) drawings for execution

Executive summary and why planning must evolve

The objective of the work, and the four drivers behind it

EXECUTIVE SUMMARY

01. Research Objective

Develop and validate a generative design framework for optimizing complex semiconductor facility layouts using Dynamo and Revit.

02. Industry Significance

Semiconductor fabs represent some of the most complex, expensive facilities in the world, where even small layout improvements yield substantial cost savings.

03. Methodology

Rule-based generation of layout alternatives for Fab, Complex building, and connecting Overbridge using parametric design and performance-based optimization

04. Expected Contributions

Systematic approach to facility layout optimization, cost reduction through utility distance minimization, and foundation for operational digital twins.

WHY IT MUST EVOLVE

01. Rapid Technological Advancement

Process nodes have moved from 2,400nm to under 3nm, and each generation changes the facility's size, utility load and cleanroom requirements

02. Massive Capital Investment

With fab construction costs exceeding $20 billion, even small layout efficiency improvements can yield hundreds of millions in savings and improve ROI

03. Operational Complexity

Modern fabs feature thousands of integrated systems, utilities, and cleanroom zones that must be optimally arranged while maintaining strict environmental controls

04. Time-to-Market Pressure

Compressed design and construction schedules demand rapid design iteration and evaluation to maintain competitive advantage in a fast-moving market.

What is Generative Design?

Definition, the paradigm shift, its benefit in AEC, and the Dynamo application

DEFINITION · PARADIGM · BENEFITS · APPLICATION

01. Definition

Generative Design is a computational design methodology that uses algorithms to explore a vast solution space based on defined goals, constraints, and parameters to produce optimized design outcomes

02. Paradigm Shift

Unlike traditional design where a designer creates one solution at a time, generative design simultaneously explores thousands of possibilities that satisfy design criteria and constraints

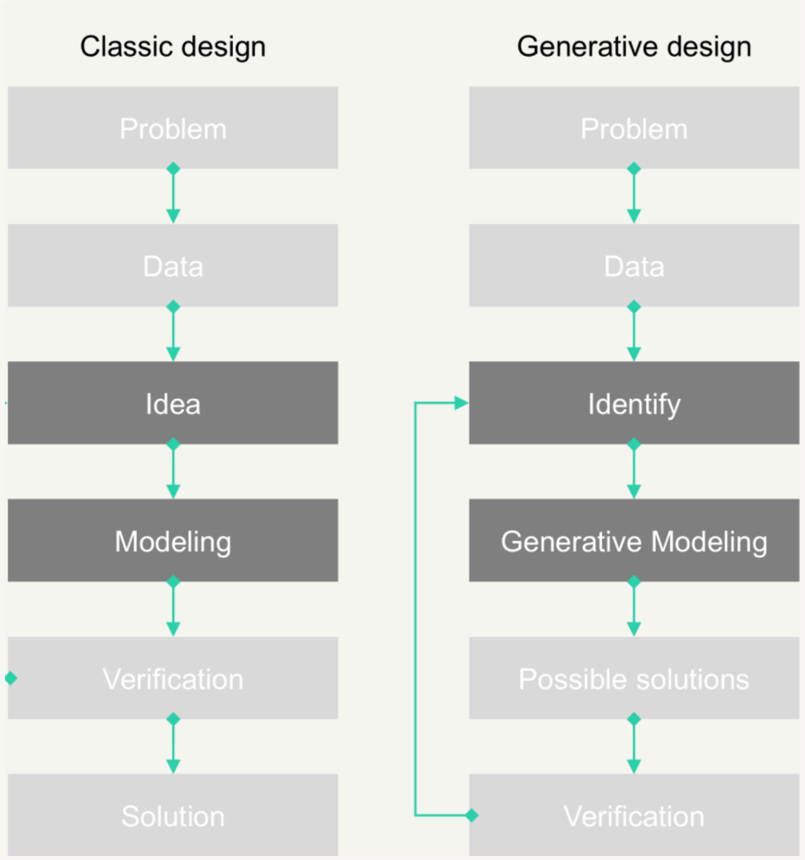
03. Key Benefits in AEC

Rapidly generates diverse solutions, optimizes for multiple objectives (space efficiency, structural performance, cost), automates repetitive tasks, and supports data-driven decision making

04. Dynamo Application

Dynamo's visual programming interface enables AEC professionals to create generative design workflows without extensive coding knowledge, making computational design accessible

CLASSIC vs GENERATIVE



Agenda

one loop, four acts — one contract runs through all four

1. **Build it**
   extract · match · refine → one library
2. **Let it plan**
   GD explores on rules the library wrote
3. **Read it**
   viewer + verified local AI · on-premise
4. **Bring it back**
   same contract → Revit → new vs existing

Build it, let it plan, read it, bring it back — one contract runs through all four.

The answers are already on your servers.

generations of fab BIM — and three reasons none of it is planning the next fab

Heavy

50+ models per discipline —
minutes to open just one

Scattered

rooms, equipment, lengths —
locked in separate RVTs

Renamed

the same room, a different
name in every generation

$20B+

per fab. A few percent of quantity and cost is worth hundreds of millions.

Thousands of rooms, tens of thousands of units, 30+ piping systems — and all of it moves.

The answers are already on your servers — heavy, scattered, renamed, and a few percent of a $20B fab is worth hundreds of millions.

One loop. Every discipline rides it.

six stages · four disciplines — layouts are conceptual, geometry anonymized for security

	1 Existing Fab BIM	2 Structure & Refine	3 Plan New · GD	4 Review · Viewer+AI	5 Execute in Revit	6 Compare
Room	rooms · levels · zones	**room DB + name history**	room program rules	room tree · filters	**room masses + auto views**	room delta ledger
Architecture & Structure	walls · ceilings floors · columns framing	merged geometry refs	grid + span limits	context display	**DirectShape context**	footprint check
MEP · Routing	30+ systems · lengths	**lengths per room · system**	utility demand rules	route toggles · length queries	rack proxy + risers	**Δ length per system**
Equipment	placed units	**5-tier join · counts**	capacity rules	EQ views · count queries	instance placement	**4/5 auto-place · Δ units**

↺ **the comparison feeds back — the archive grows; nothing is hand-carried between stages**

One loop, six stages — and every discipline rides it; nothing is hand-carried between stages.

Build it.

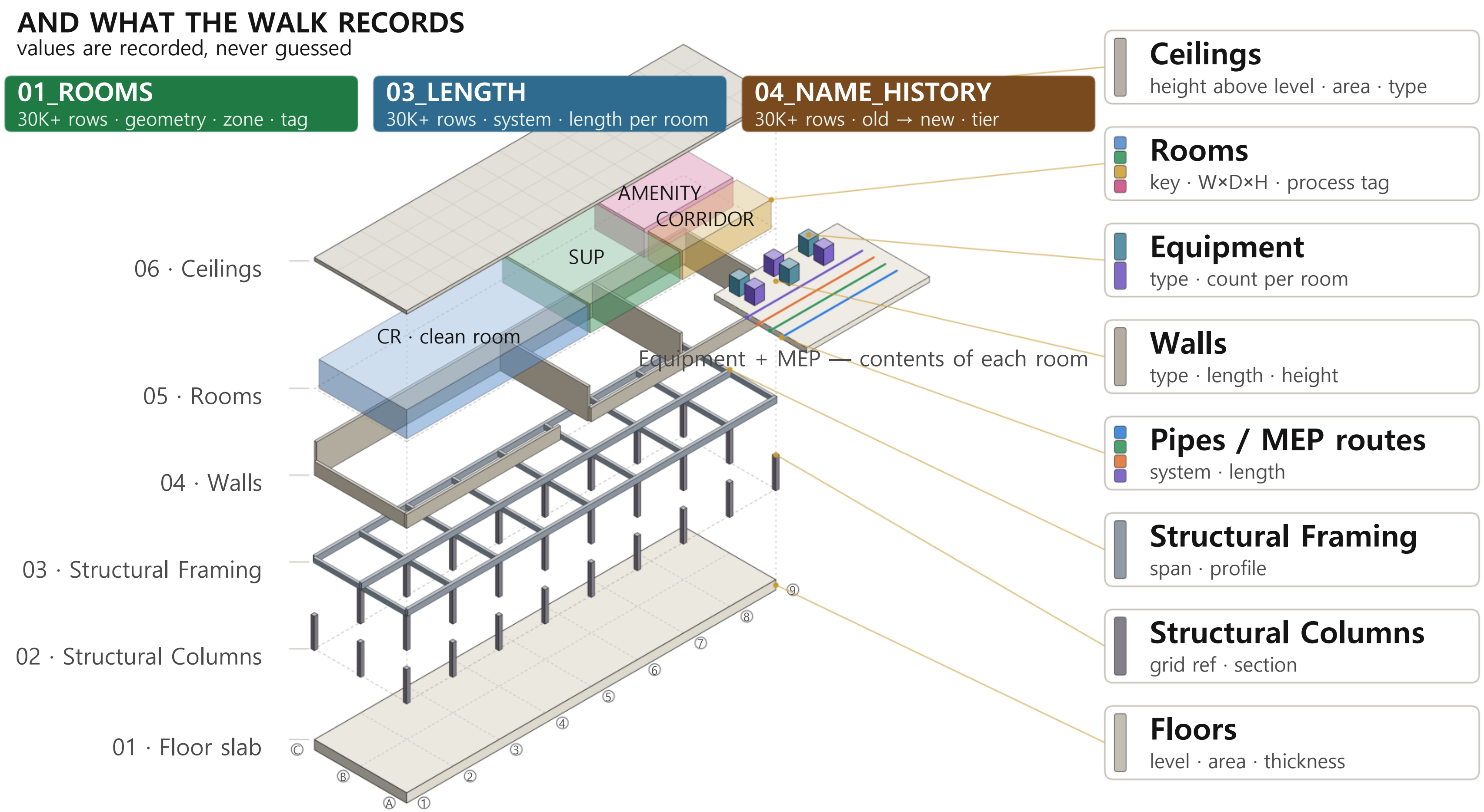
ACT 1

#AU2026

Act one — build the library out of existing fab BIM.

What the add-in reads — and records.

eight Revit categories → fields → rows · values are recorded, never guessed



schematic bay · column grid 12 m — fab-like proportions, not project coordinates · card fields = extractor schema per category

Eight Revit categories, field by field — and the rows they land in.

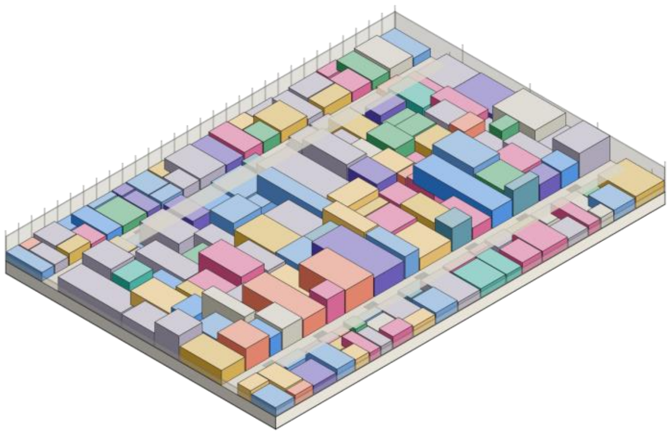
Rooms in their structure.

two levels, one bay — zoning · column grid · walls · equipment · routing

1 · what the archive holds

3RD FL. — production level

rooms as built · one level of seven



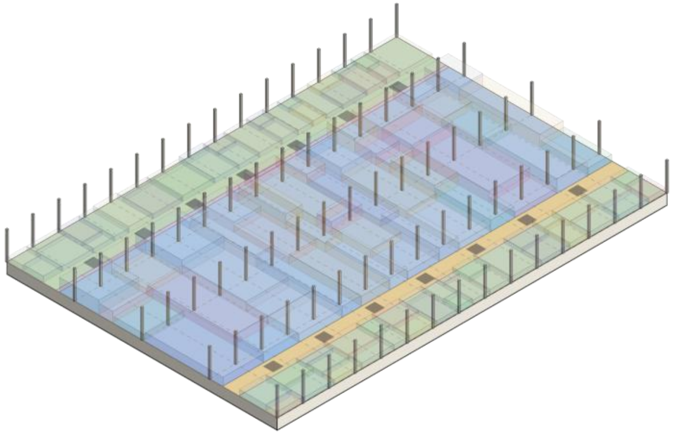
- clean room
- support
- circulation
- transport
- amenity · storage · safety
- other

colour = process zone tag where the room carries one, else category

2 · how the floor is zoned

1ST FL. — support level

rooms ghosted · fab zoning + 80 columns

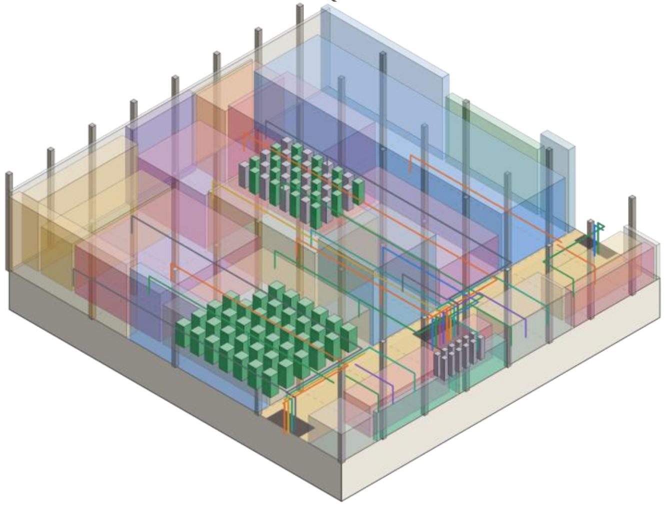


- SUB — support band
- SHAFT-B — drop shafts
- MAIN — production band
- SHAFT-A — air ducts

3 · what the later acts touch

ONE BAY — everything layered

30 rooms · 92 EQ units · 71 routes



- rooms (ghosted)
- columns · grid
- perimeter wall
- equipment units
- routing bundles

Two levels doing different jobs, and one bay with everything layered on.

column grid is an assumption (not in the export) — flagged in plate notes

What the archive holds

four disciplines · one join key

Architecture	Structure	MEP · Routing	Equipment
rooms · levels · zones	frames · slabs · openings	30+ systems · lengths	tens of thousands of units
room polygons, level datum, program tags, name history	grids, spans, floor datums, shaft openings	system codes, per-room lengths, shaft targets, riser rules	types, counts per room, placement grids, capacities

Dozens to 150+ RVT models per discipline · three fab generations · tens of GB each
Nobody reviews that by opening files. One key makes it one database: { level | room name }

{ level | name } = join key

Four disciplines, dozens of models each — one join key makes a database.

Five ways to recognize a room.

three generations of renaming, absorbed

MATCHING LADDER

band width = rows resolved at that tier, counted from 04_NAME_HISTORY

	Tier	Rows	Share	Rule	Example
1	EXACT	808	46%	identical key	MECH RM-2 → MECH RM-2
2	HISTORY	331	19%	old instance name, chained	ANALYSIS LAB-2-3 → ANALYSIS LAB-2
3	NORMALIZED	246	14%	separator / suffix noise stripped	WET-PROCESS-RM (B) → WET PROCESS RM
4	BRIDGE	147	8%	cross-project alias	CHILLER RM → MECH RM-3
5	FAMILY	154	9%	generic family → specific member	CONTROL 10x → CONTROL RM
6	UNRESOLVED	67	4%	no rule matched — listed for a human	?? → STORE

Σ 1,753 rows resolved = 1,753 in 04_NAME_HISTORY

every row lands in exactly one bucket

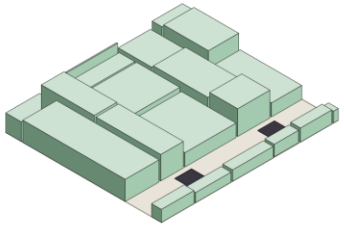
EXCLUSIVE ACCOUNTING

01_ROOMS splits with nothing left over — and the overlap is swept, not claimed

1,753

■ rooms ■ shaft voids ■ zone markers

1,753 + 98 + 14 = 1,865 rows in 01_ROOMS



rooms — generated and counted

shaft voids stay empty — never solid

SHAFT-A 49 · SHAFT-B 49 · 14 voids on this level

measured overlap
0.000 m²

room × void area sweep over 163 rooms and 14 voids (3RD FL.)

Five ways to recognize a room — three generations of renaming, absorbed.

Let it plan.

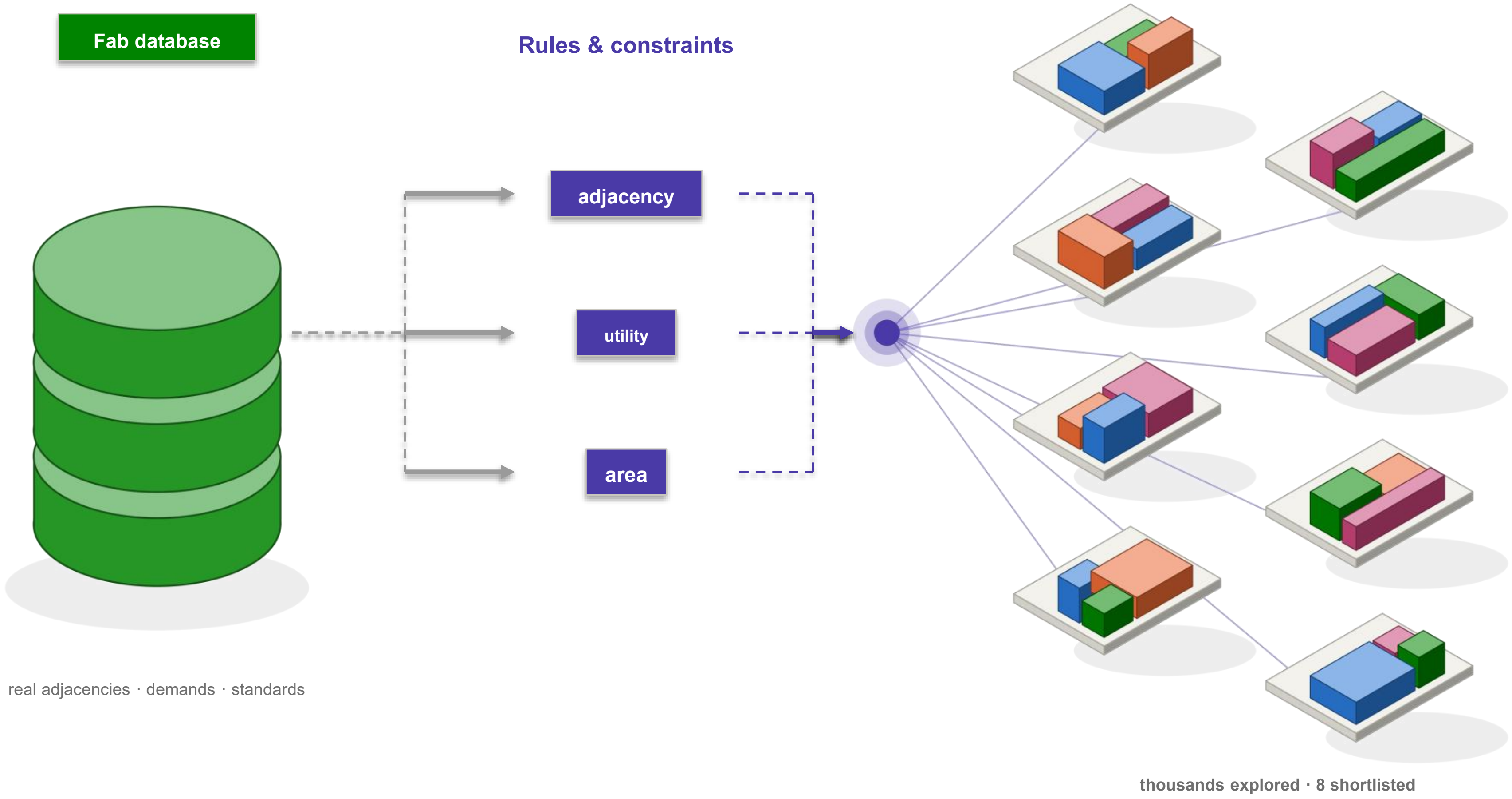
ACT 2

#AU2026

Act two — the library writes the rules; GD explores.

The archive writes the rules.

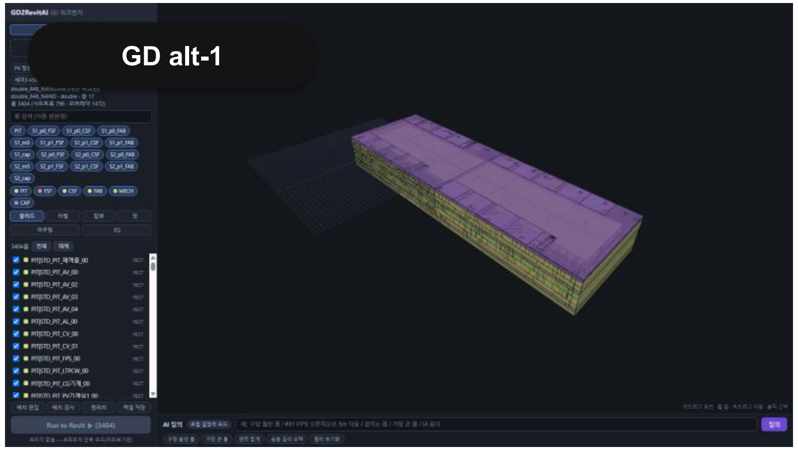
last year's framework — now fed by real data

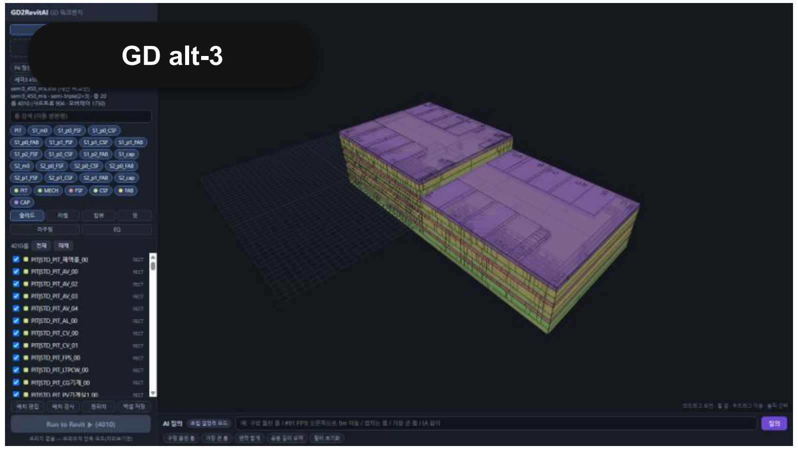


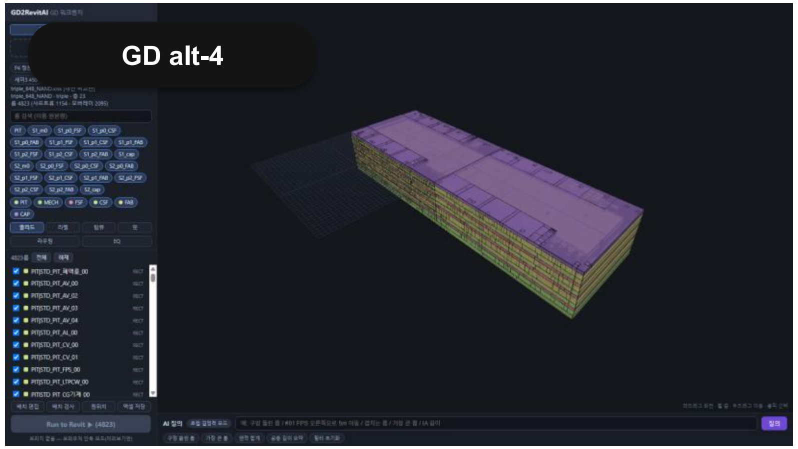
The archive writes the rules — last year's GD framework, now fed real data.

Six alternatives, one click apart.

six GD layouts · eight loadable datasets — the metrics never move







Six GD alternatives, one click apart — the metrics never move.

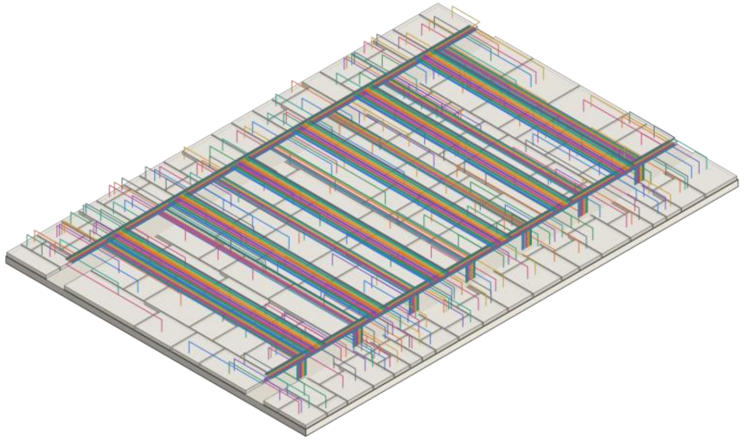
Routing and equipment, computed by rule.

routing · equipment · and the two connected

1 · ROUTING, COMPUTED BY RULE

unit → riser → rack → corridor spine → nearest shaft

300+ routes · 60+ trunk lines · rule-assigned where no join



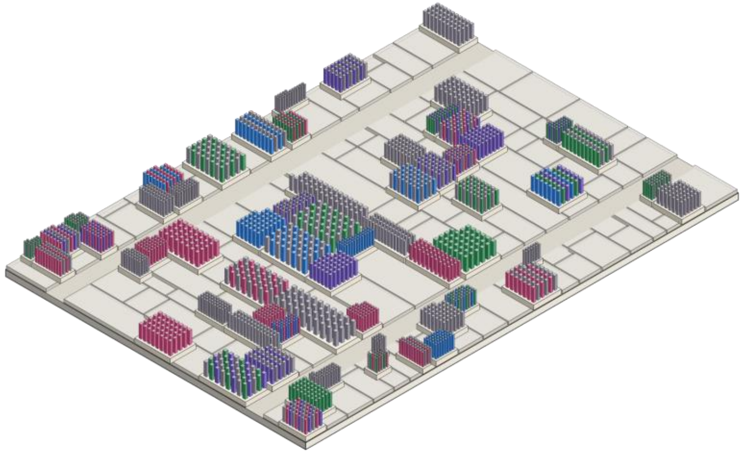
SYS-A SYS-B SYS-C SYS-D
SYS-E SYS-F SYS-G SYS-H

8 systems · 0.35 m lanes (display-exaggerated)

2 · EQUIPMENT, PLACED BY RULE

grid per room · cap 40

60+ placed rooms · 2K+ units



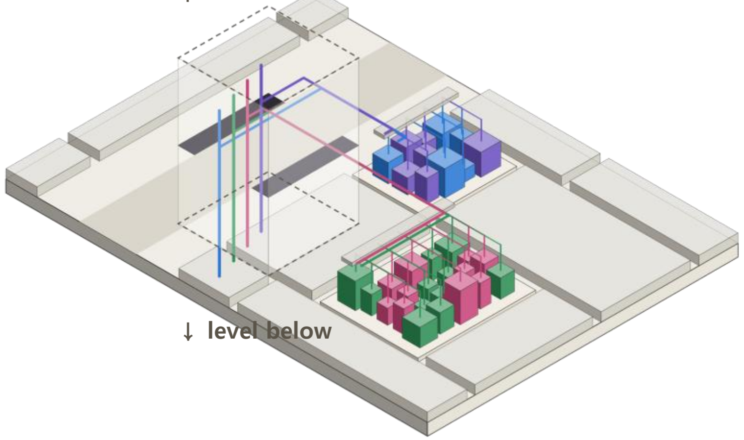
SYS-A SYS-B SYS-C SYS-D
ETC

top-4 codes + rest — anonymized

3 · ONE BAY, EVERY UNIT CONNECTED

every unit wired · the shaft goes up and down

two rooms · 30+ units · one riser each · 4 systems in the shaft



① unit ② riser ③ rack ④ spine ⑤ shaft ↕

3RD FL. · conceptual · rack drawn raised · not as-built

GD_EXPORT_DEMO.xlsx · 3RD FL. · conceptual layout · rule reconstruction, not as-built

Routing and equipment, computed by rule — and connected.

Read it.

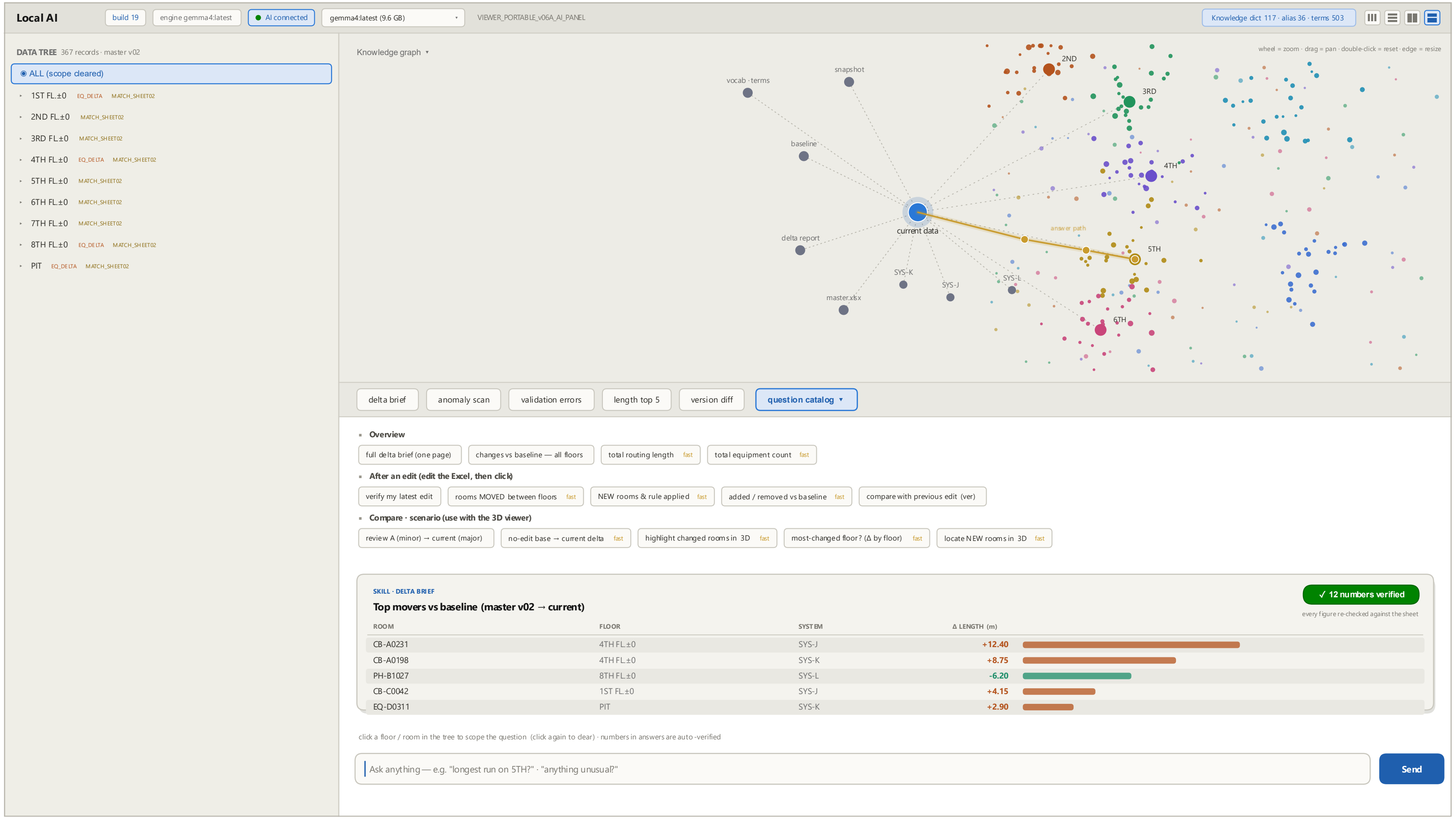
ACT 3

#AU2026

Act three — read the result: viewer and a verified local AI.

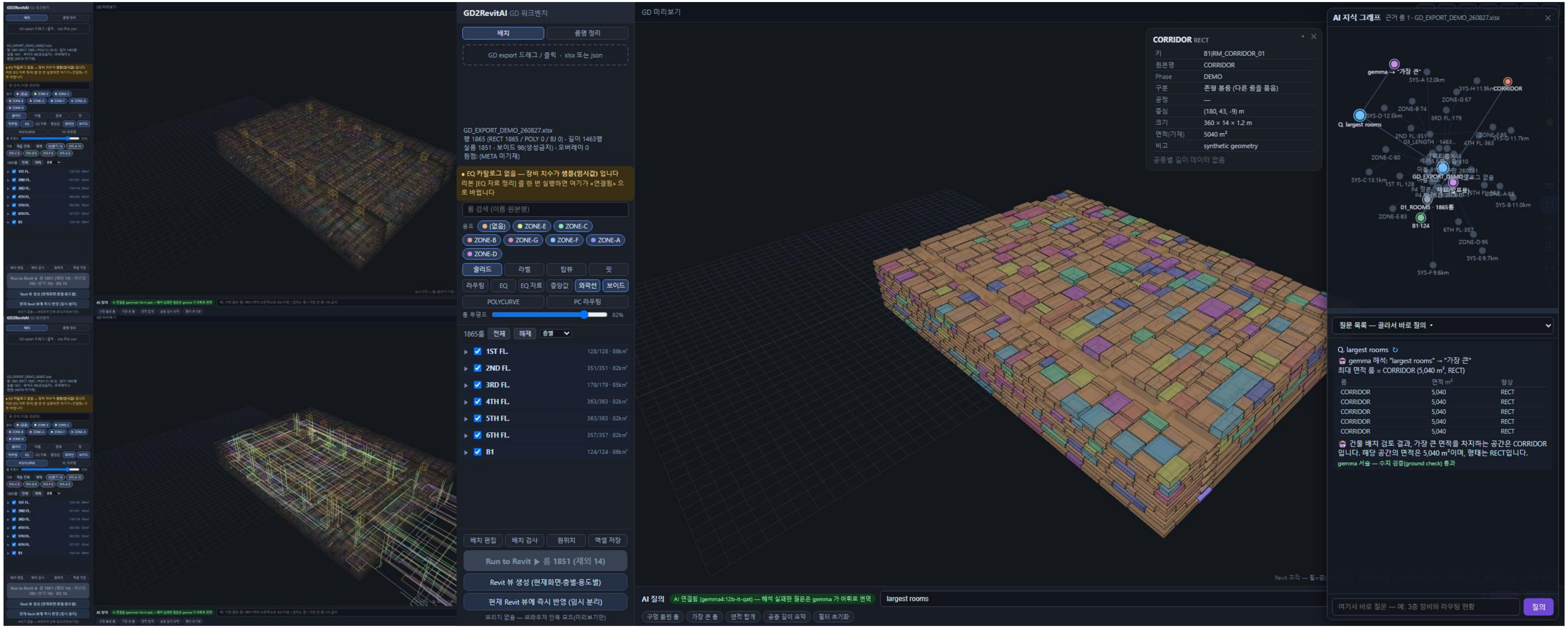
The panel, piece by piece.

every element reconstructed — the real screen follows



Real tool. Concept model.

live capture of the shipped add-in — the model is synthetic



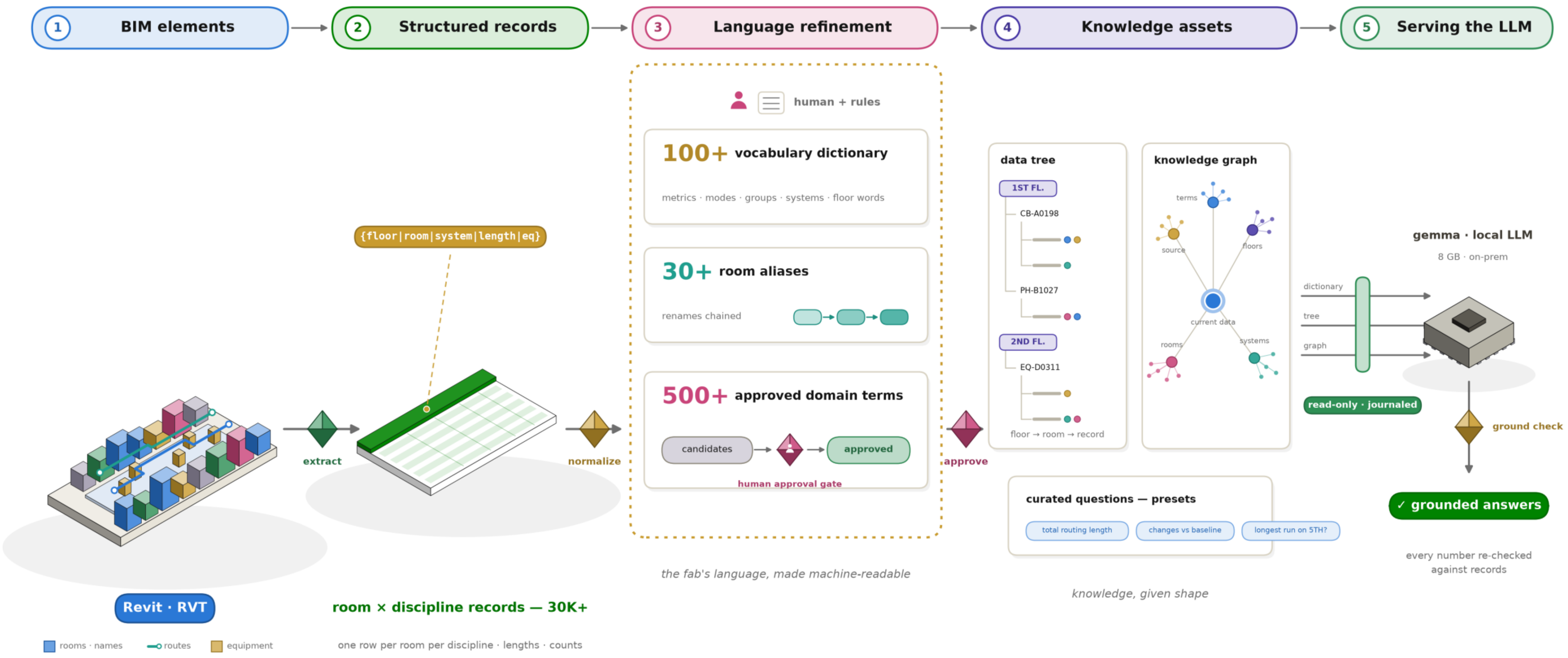
Live capture of the shipped add-in. Room names, systems and quantities are generalized — the fab data never leaves Samsung.

The graph thinks in stages.

a replay of the real access trace — not an animation

How BIM becomes AI knowledge.

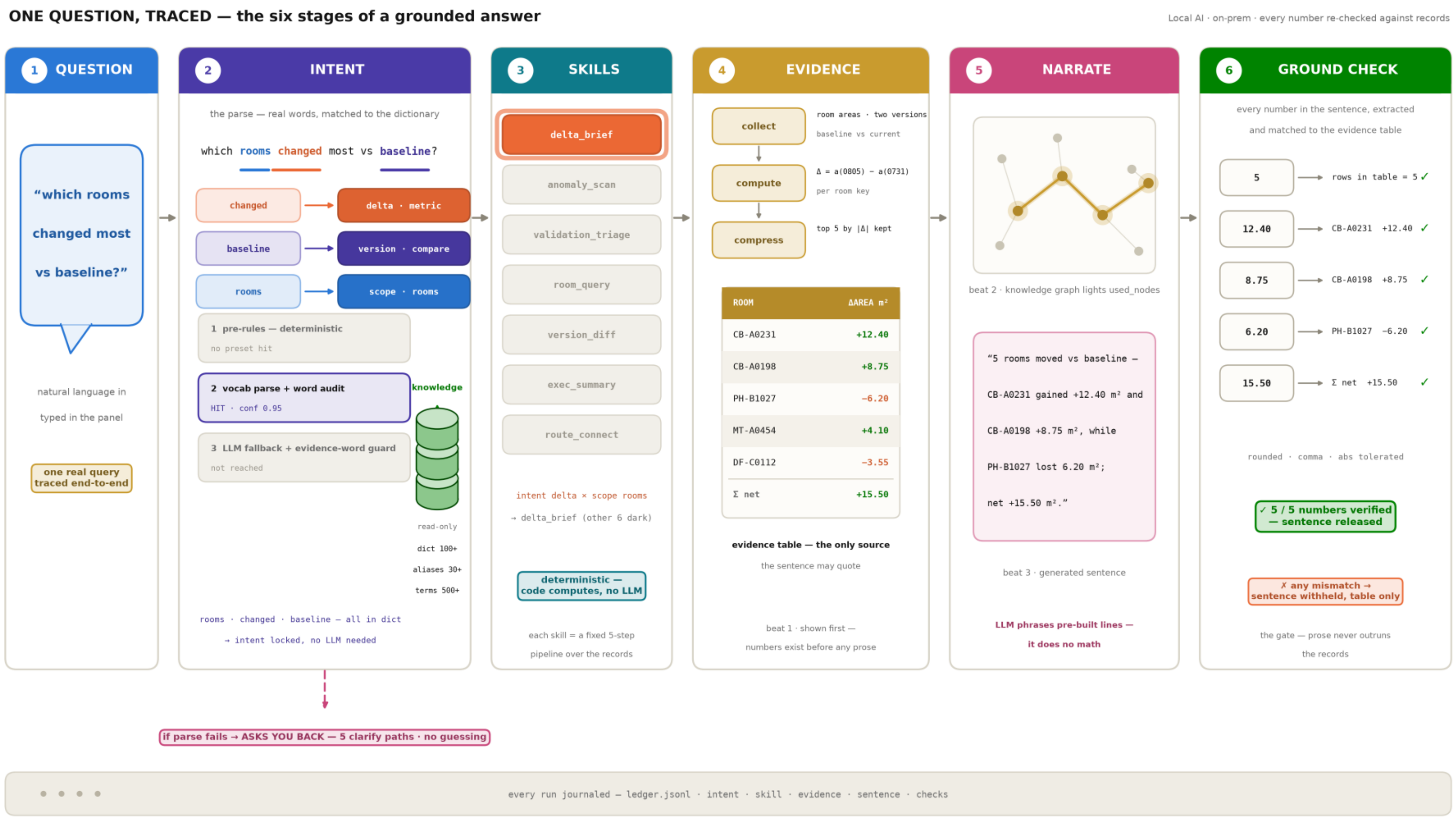
extracted · named · approved · served



BIM data doesn't become AI knowledge by magic — it is extracted, named, approved, and served. Every step inspectable.

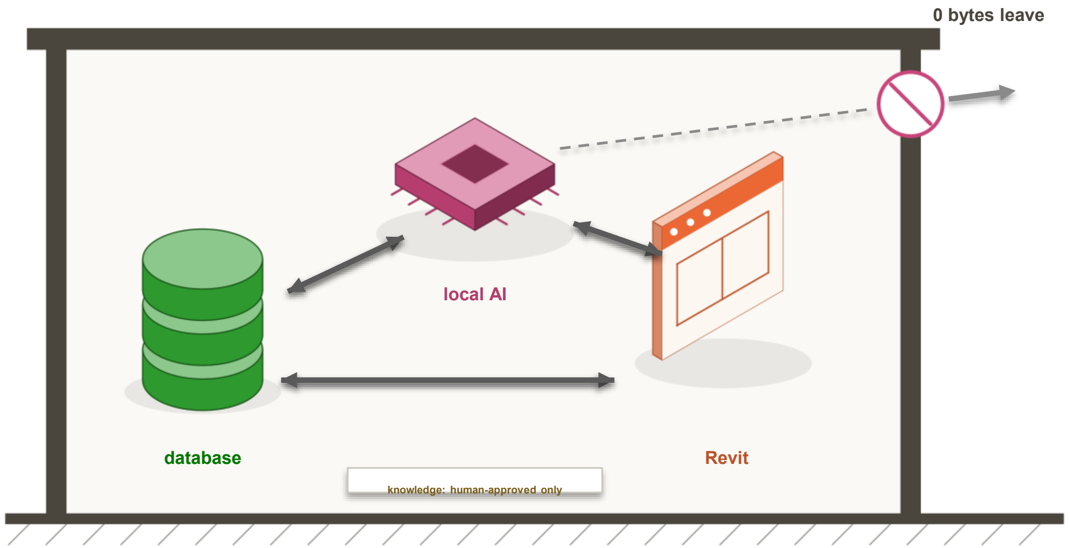
Inside the AI: six honest steps.

one real query, traced end to end



0 bytes

leave the building.
The AI only knows what
a human approved.



Zero bytes leave the building — the AI only knows what a human approved.

Bring it back.

ACT 4

#AU2026

Act four — GD's answer returns through the same contract, into Revit.

One contract, both ways.

five sheets — the archive comes in through it, GD's answer goes out through it

EXISTING FAB BIM

150+ RVT · four disciplines · three generations

IN — the add-in reads →

THE CONTRACT

five sheets · one schema

- 01 ROOMS
- 02 ROOM_POLYGON
- 03 LENGTH
- 04 NAME_HISTORY
- 05 META

the same five sheets, both ways
any room shape · parsed in half a second

loads →

THE LIBRARY

one database · name history · rules

GD — SIX ALTERNATIVES

explored on rules the library wrote

OUT — the chosen plan exports →

builds →

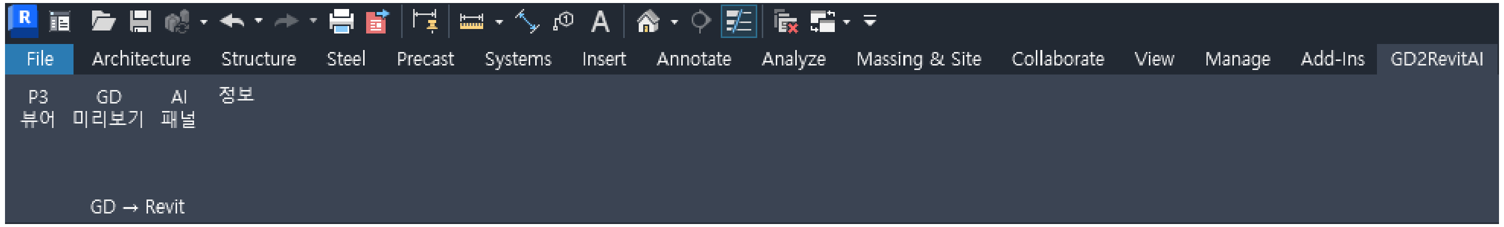
REVIT

rooms · masses · views · three ledgers

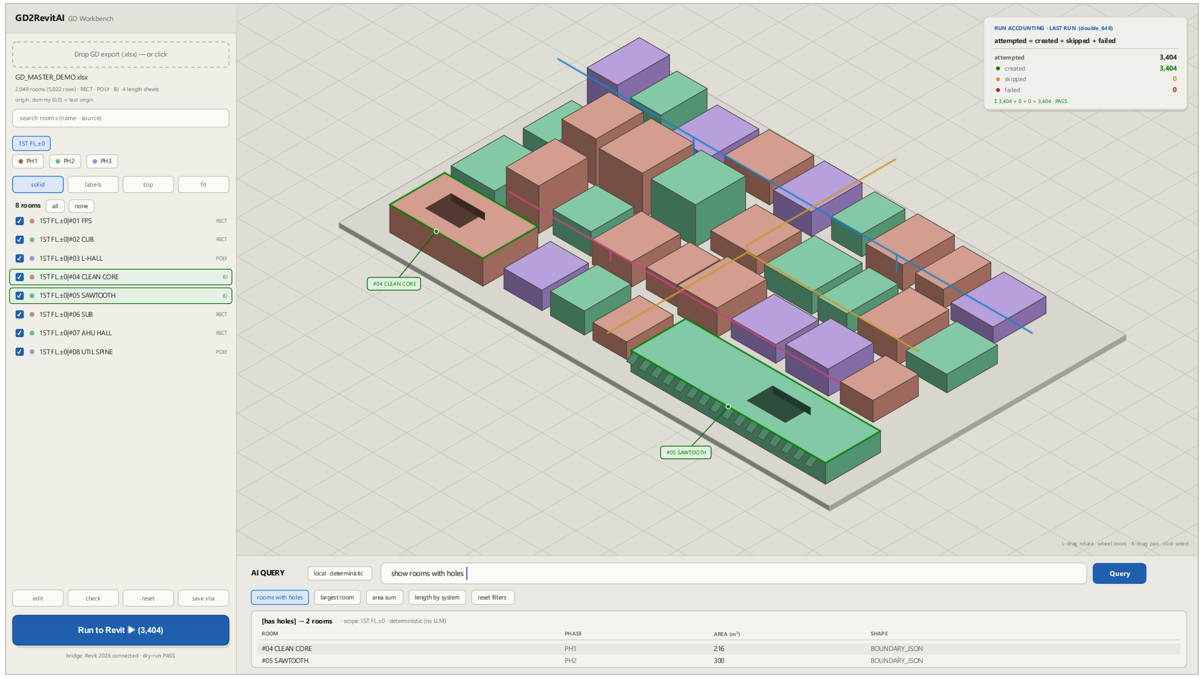
One contract, both ways — the archive comes in through it, GD's answer goes out through it.

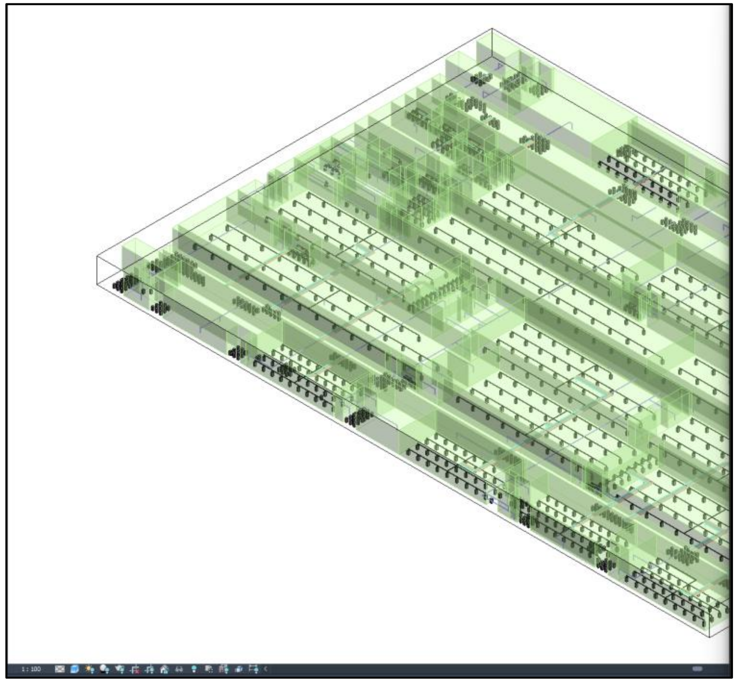
It lives inside Revit.

one tab, then one screen — list · viewport · run · accounting



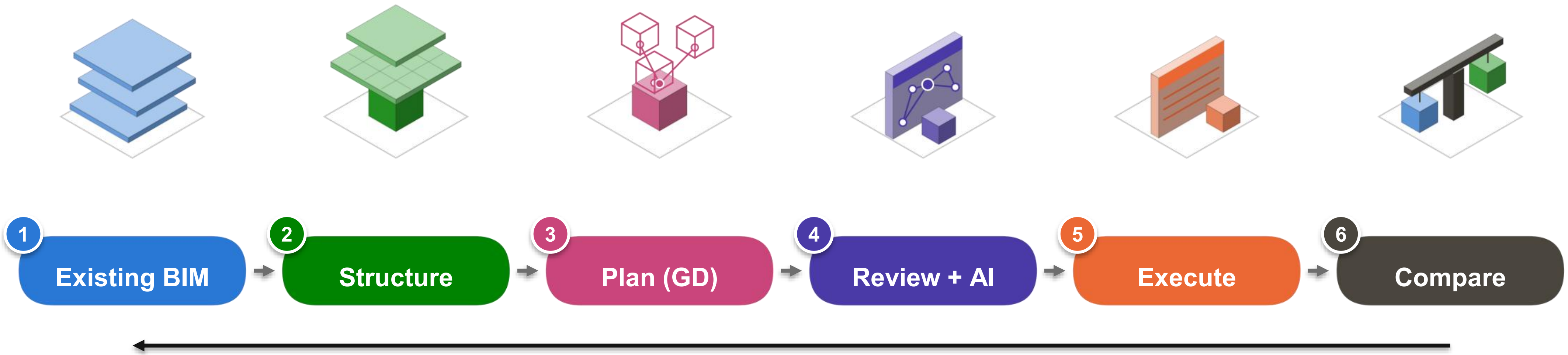
One tab — GD2RevitAI.
extract · preview · AI panel · generate
the shipped tab, Revit 2026





One tab inside Revit, then one screen — the list, the viewport, the run, and the accounting.

Key takeaways.



The archive — structured once — it compounds across every next fab

The new plan — GD explores on rules the archive wrote

the comparison feeds back — the archive grows

The review — seconds to query, verified answers, on-premise

The comparison — the last fab grades the next — in quantities you can price

Four takeaways — the archive compounds; review is verified; GD explores; comparison prices change.

Real today. Next, honestly.

what 'under development' means, precisely

RUNNING TODAY

- Existing-fab extraction → refined database
- GD alternatives under one contract
- Viewer + verified local AI review
- Revit generation · views · delta ledgers

NEXT, HONESTLY

- Cost axis — awaits a unit-price table, by design
- LLM tuning — synthetic data trains vision models today
- Undo — today, rollback = idempotent replace
- Unresolved rooms — explicit, human-assigned

What runs today, and what's next — honestly, by design.

Your archive is the AI's textbook. Revit and Dynamo are its hands.

From data to decisions — Factory Design & Construction on the Autodesk platform.

Your archive is the AI's textbook; Revit and Dynamo are its hands.

AU2026

THE DESIGN & MAKE CONFERENCE