

A small red horizontal bar is located to the left of the main title.

Revolutionizing Verification With GenAI Powered Automation

A Paradigm Shift Towards Agentic Workflows

Dr Charles Alpert, AI Fellow

Verification Futures 2025

This presentation contains forward-looking statements based on current expectations or beliefs, as well as a number of preliminary assumptions about future events that are subject to factors and uncertainties that could cause our actual results to differ materially from those expectations or results described in the forward-looking statements. You are cautioned not to put undue reliance on these forward-looking statements, which are not a guarantee of future performance and are subject to a number of risks, uncertainties and other factors, many of which are outside Cadence's control.

Agentic AI: Continuing Evolution of EDA

EDA's 40-Year History

Improving Productivity Through Design Abstraction

Manual
Layout

Automated
Transistor-Level
Layout

Automated
Cell-Based
Place and Route

RTL
Synthesis

High-Level
Synthesis

Excellence in Simulation and Optimization

NUMERICAL SOLVERS

BINARY DECISION DIAGRAMS

SAT SOLVERS

SYMBOLIC MODEL CHECKERS

SMT SOLVERS

BOUNDED MODEL CHECKER

ADAPTIVE MESHING

MATRIX SOLVERS

GRAPH THEORY AND TOPOLOGY

DISTRIBUTED PROCESSING

NON-LINEAR SOLVERS

COMPUTATIONAL GEOMETRY

cāden̄ce.ai

Natural Language

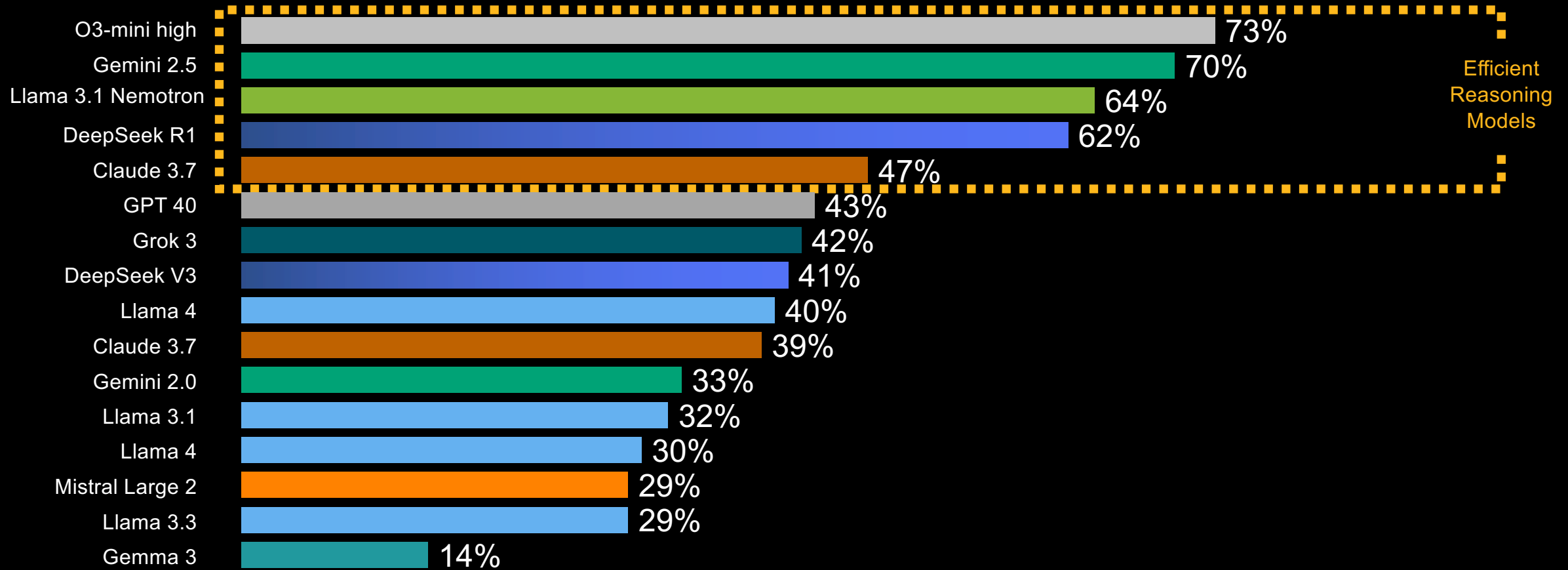


Cadence
Agentic AI

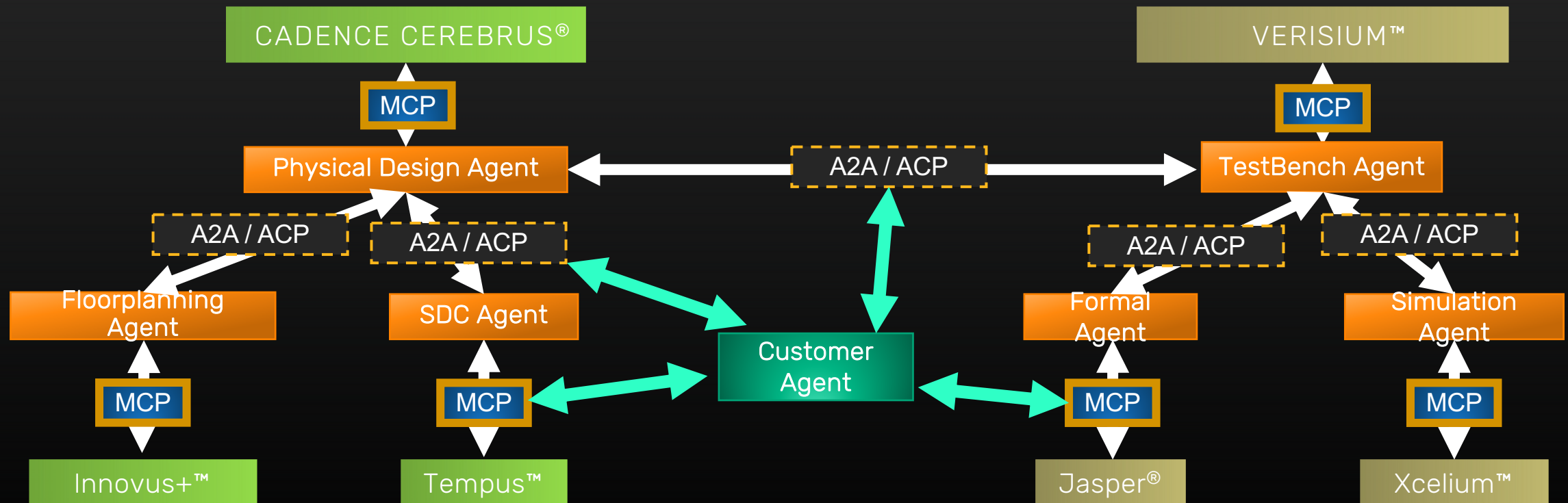
Optimization AI

The Emergence of Reasoning Models

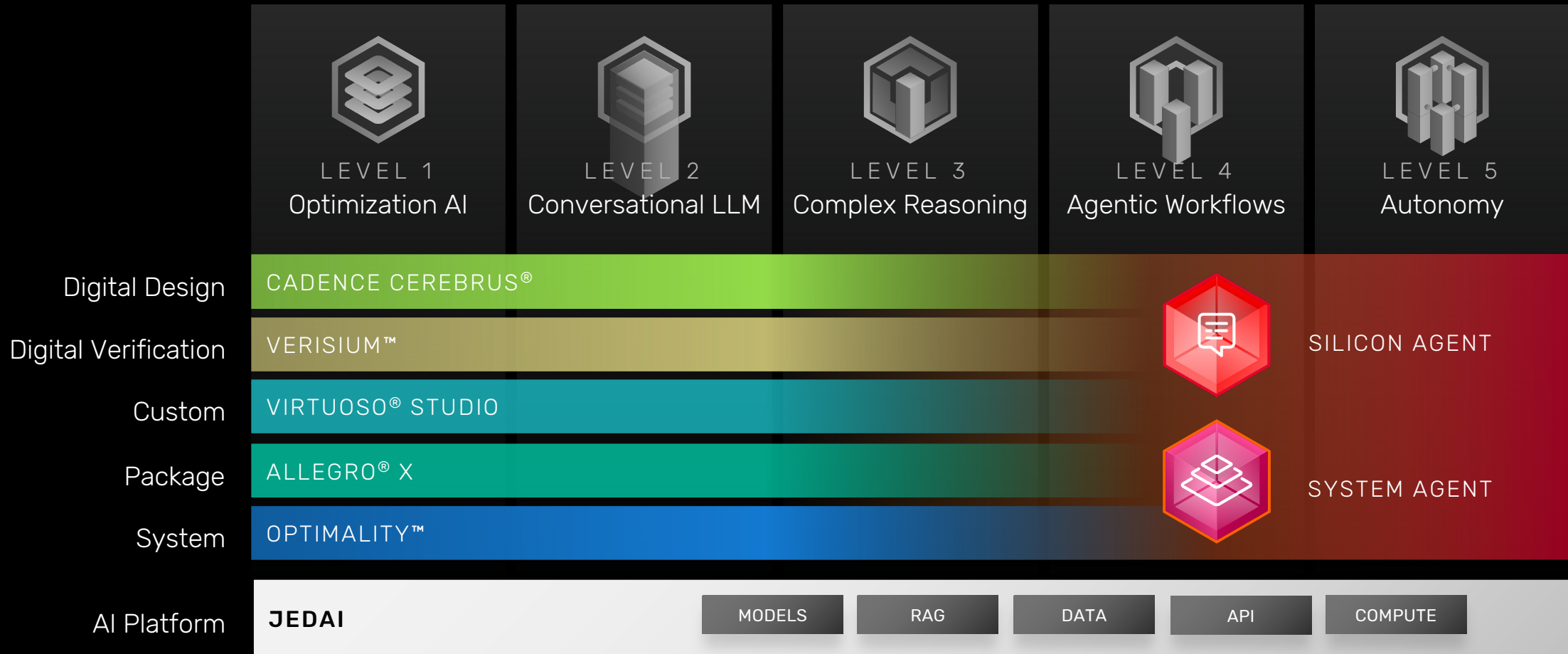
LiveCodeBench (Coding) Results



New Protocols Are Enabling Plug-and-Play Agentic Systems



The Journey to Autonomous Design



Optimization AI in Digital Verification: Verisium SimAI



LEVEL 1
Optimization AI

cādence.ai

Chips-to-systems intelligent design acceleration

AGENTIC AI

Cadence®

OPTIMIZATION AGENTS

Verisium™ SimAI

Debug and Verification

10x

Verification productivity

AI-driven automation and analysis for digital chip verification

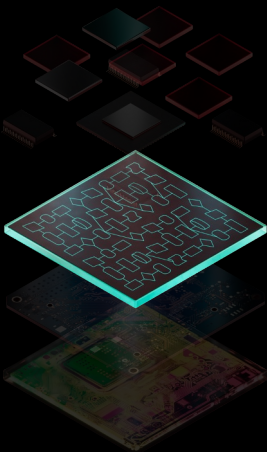
AI-ENHANCED DESIGN ENGINES

Jasper™

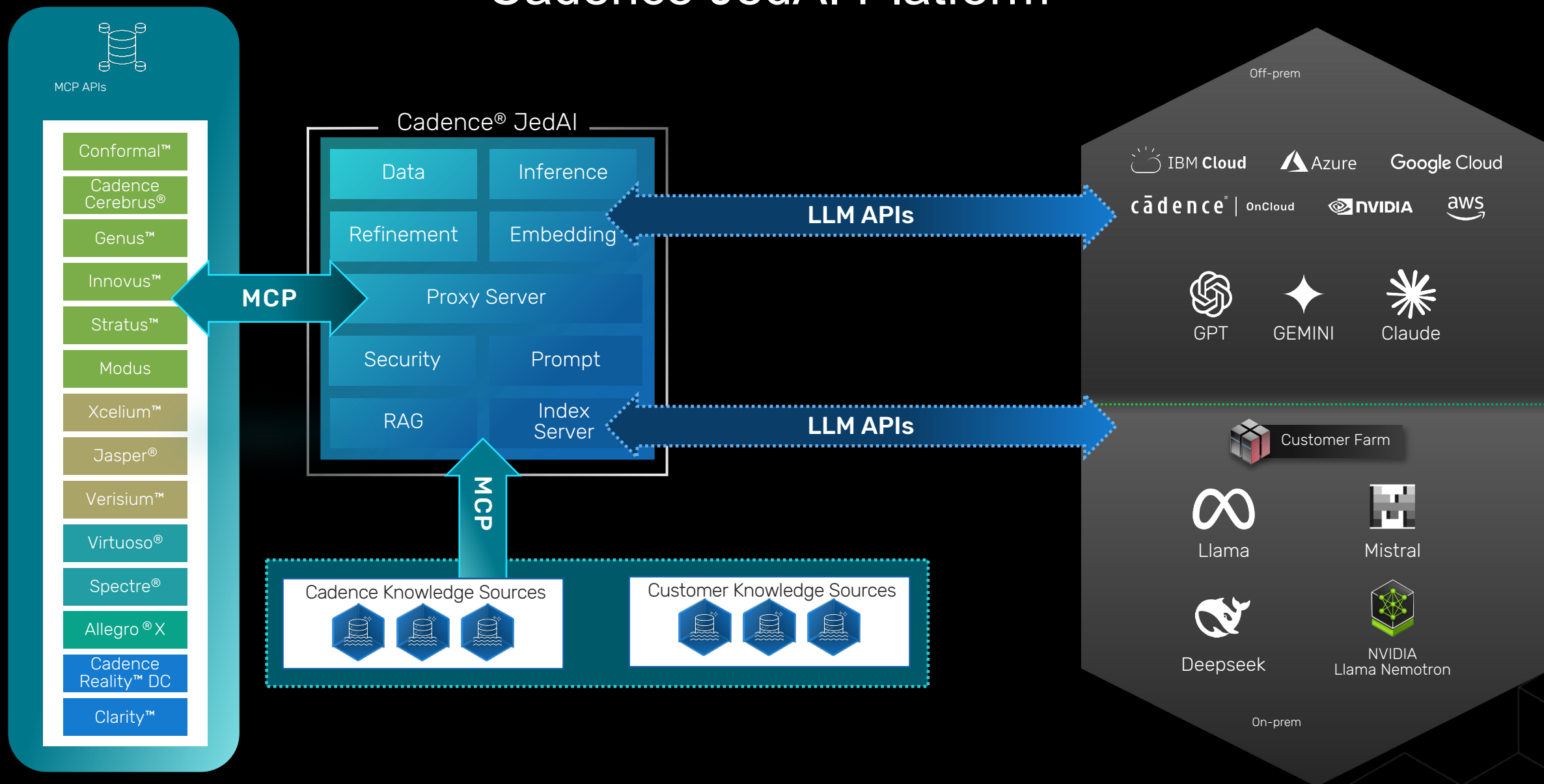
Xcelium™

Design Type	Speedup
Modem IP	20X
Mobile SoC	7X
AI IP	8X
AI IP	3X
Memory IP	5X
Base Station	18X
NOC IP	5X
Automotive SoC	5X
CPU IP	3X
Network IP	4X
CPU IP	9X

Bug-free designs



Cadence JedAI Platform



The Deployment Dilemma



On-Premises



Cloud



LEVEL 3

Complex Reasoning

```
Python
line = re.search(r"Endpoint:\s+(.+)", b)
ie = re.search(r"Path Group:\s+(.+)", b)
ie = re.search(r"clock\s+(.+)\s+\(rise edge\)", b)
ie = re.search(r"slack\s+(VIOLATED)\s+([-+]\d\.\d)", b)
ie = re.search(r"data ar", b)

= endpoint_line.group(1).strip() if endpoint_line else
= (group_line.group(1).strip() if group_line else
= next((g for g in (clock_line.group(1) if clock_line
clock_line.group(2) if clock_line

Verilog
parameter RESET_VAL = 1'b0
) (
input wire clk_dst,
input wire rst_n, // async active-low reset
input wire async_in, // asynchronous input
output wire sync_out, // synchronized to clk_dst
);
reg s1, s2;
always @(posedge clk_dst or negedge rst_n) begin
    if (!rst_n) begin
        s1 <= RESET_VAL;
    end
end

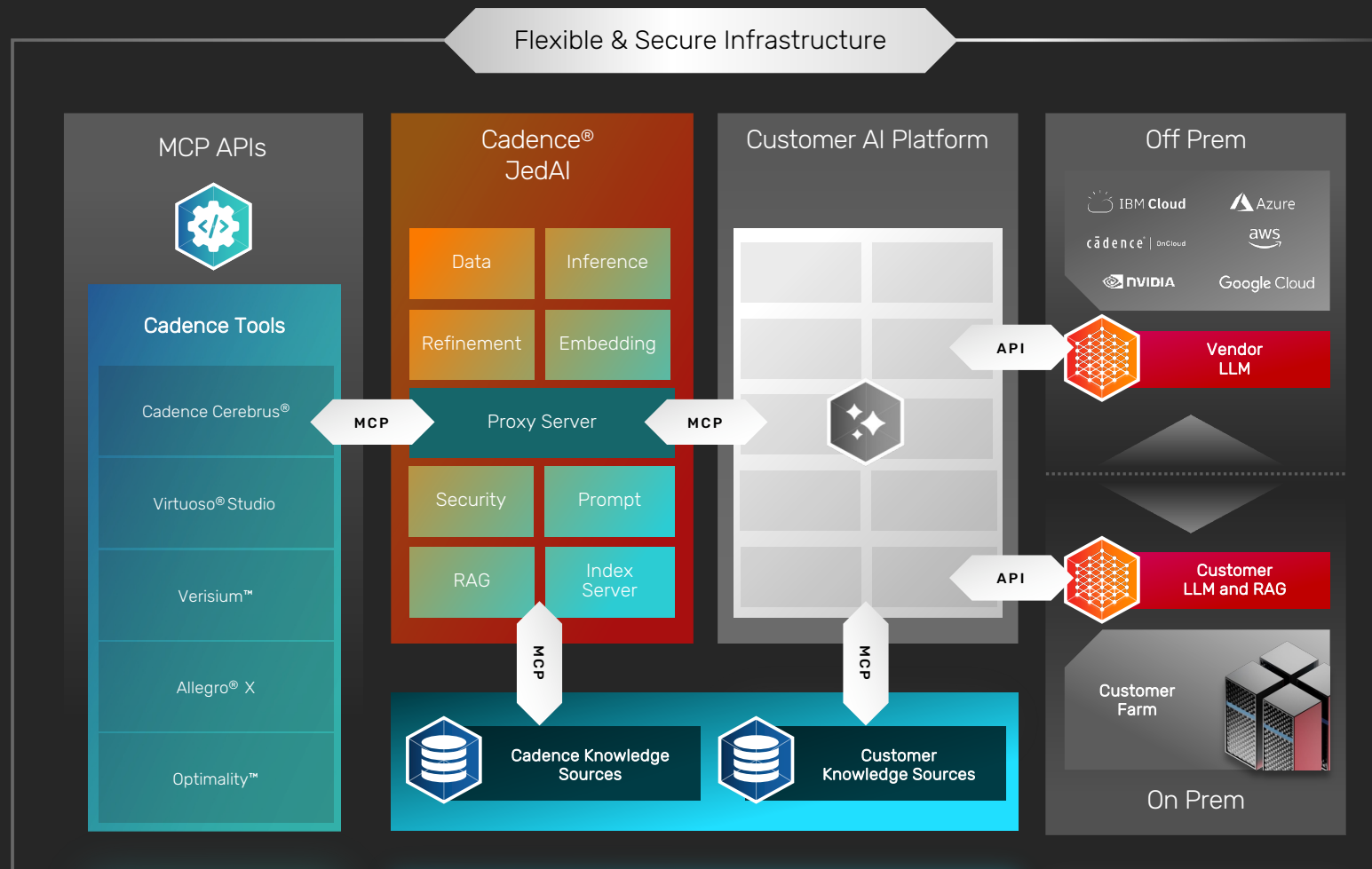
when(inst->prop->w w = inst->prop->w->value)
when(inst->prop->l l = inst->prop->l->value)
; fallbacks: check master CDF
props = cdfGetCellCDF(master)
when(props
    unless(w w = cdfFindParamByName(props "w")
    unless(l l = cdfFindParamByName(props "l")
)
; If still nil, skip
whenf w. 22.1

TCL
read_libs [list $::env(STD_CELL_LIBS) $::env(MEM_LIBS)]
read_lef [list $::env(TECH_LEF) $::env(STD_LEF) $::env(MEM_LEF)]
read_verilog ./rtl/$design.v
link_design $design

# --- Clocks & basic EDC ---
create_clock -name CORE_CLK -period 2.000 [get_parts clock]
set_clock_uncertainty 0.05 [get_clocks CORE_CLK]
set_input_delay 0.2 -clock CORE_CLK [all_inputs]
set_output_delay 0.2 -clock CORE_CLK [all_outputs]
```



Level 3 - Code Generation



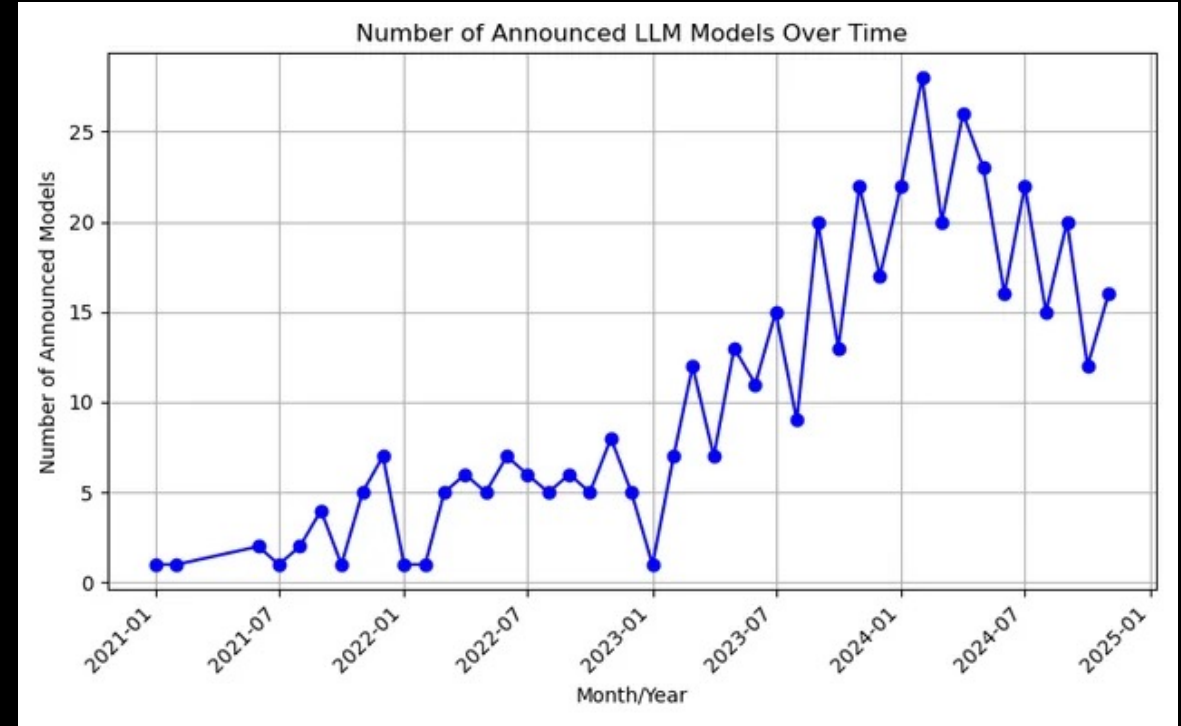
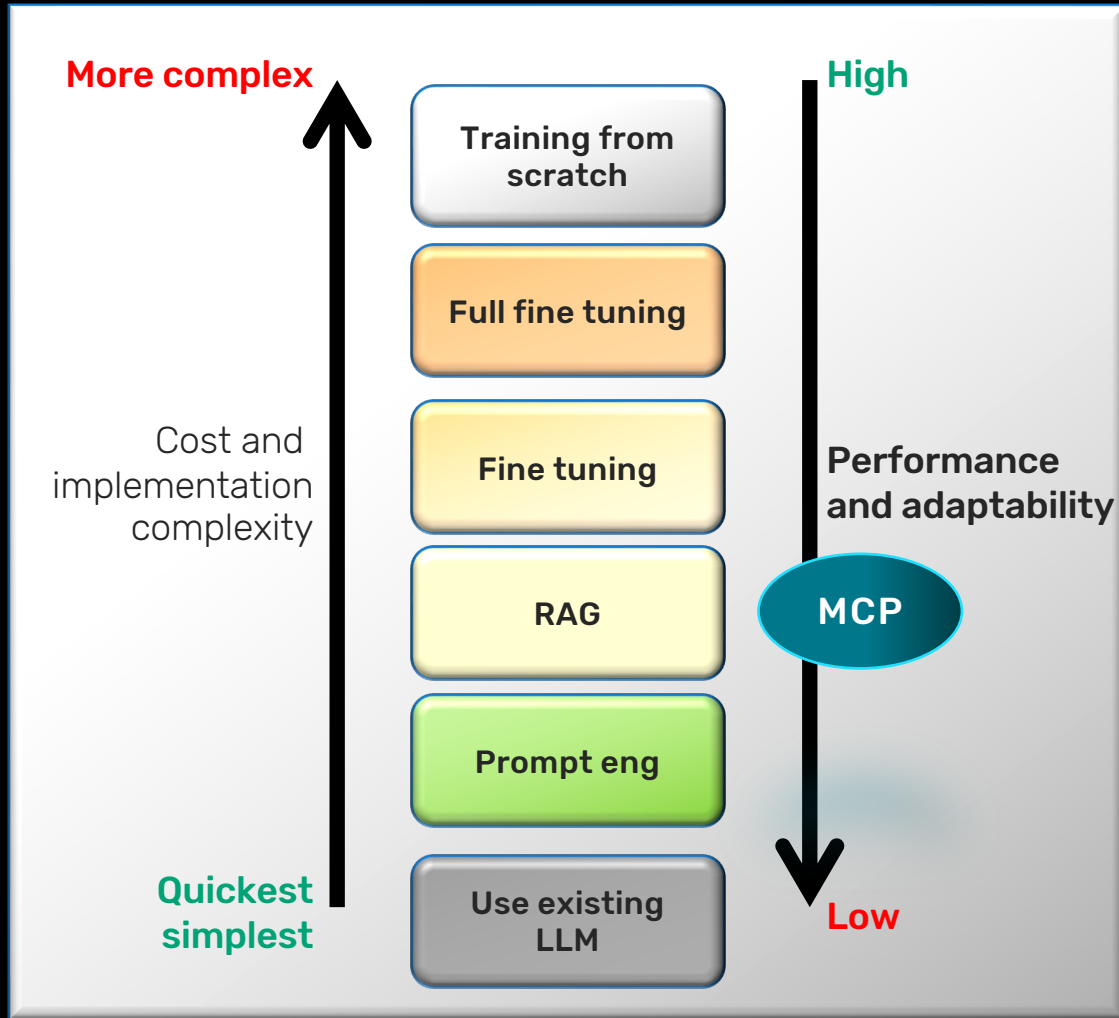
vDebug AI Assistant

The screenshot displays the Verisium Debug AI Assistant interface, which is a tool for debugging and analyzing hardware designs. The interface is divided into several sections, each with a specific function:

- UVM and Low power knowledge:** This section provides information about UVM (Universal Verification Methodology) and low power design techniques.
- Search code:** A search bar at the top right allows users to quickly find code snippets or variables within the design.
- Full hierarchy and design knowledge:** The left sidebar shows a hierarchical view of the design, including components like `cdn_multi_transfer` and `cdn_serial_transfer`.
- Full access to Smartlog DB messages:** The central pane displays Smartlog messages, including UVM reports and error messages. For example, it shows a mismatch between expected and received data: `UVM_ERROR - Mismatch between cdn_multi_transfer - received data : 0x89c930df expected data : 0x7636cf20`.
- Explain and perform Driver Tracing:** The right pane shows the source code of the design, with annotations and tracing information. It includes code snippets like `data_out <= #1 cha_read == 1 ? cha_data :` and `parity_out <= #1 cha_read == 1 ? cha_parity :`.
- Summarize signal functionality:** This feature provides a high-level summary of the signal's behavior and its role in the design.
- Inspect code for potential issues:** The interface highlights potential issues or errors in the code, such as the mismatch mentioned in the Smartlog messages.
- Natural Language complex expressions:** The AI Assistant can understand and respond to natural language queries, making it easier for users to interact with the tool.
- Send to waveform:** A button at the bottom left allows users to send selected data to the waveform view for further analysis.
- Highlight values:** The waveform view at the bottom displays signal values over time, with specific values highlighted for emphasis.
- Help docs and settings:** A bottom bar provides access to help documentation and configuration settings.
- Full access to waveform DB values:** The waveform view provides a detailed view of the signal values, including their timing and relationships.

The interface also includes a chat prompt area on the right, where users can enter queries like "explain the functionality of the @visible_code". The chat area also displays a list of suggested prompts, such as "Compare the @smartlog.selected_r", "Find the declaration of function read_proc", and "Find all the times where arbiter.gnt0 had an x value in it".

The Agentic AI Investment Dilemma



SOURCE: Reddit, Dec 2024.



LEVEL 4
Agentic Workflows

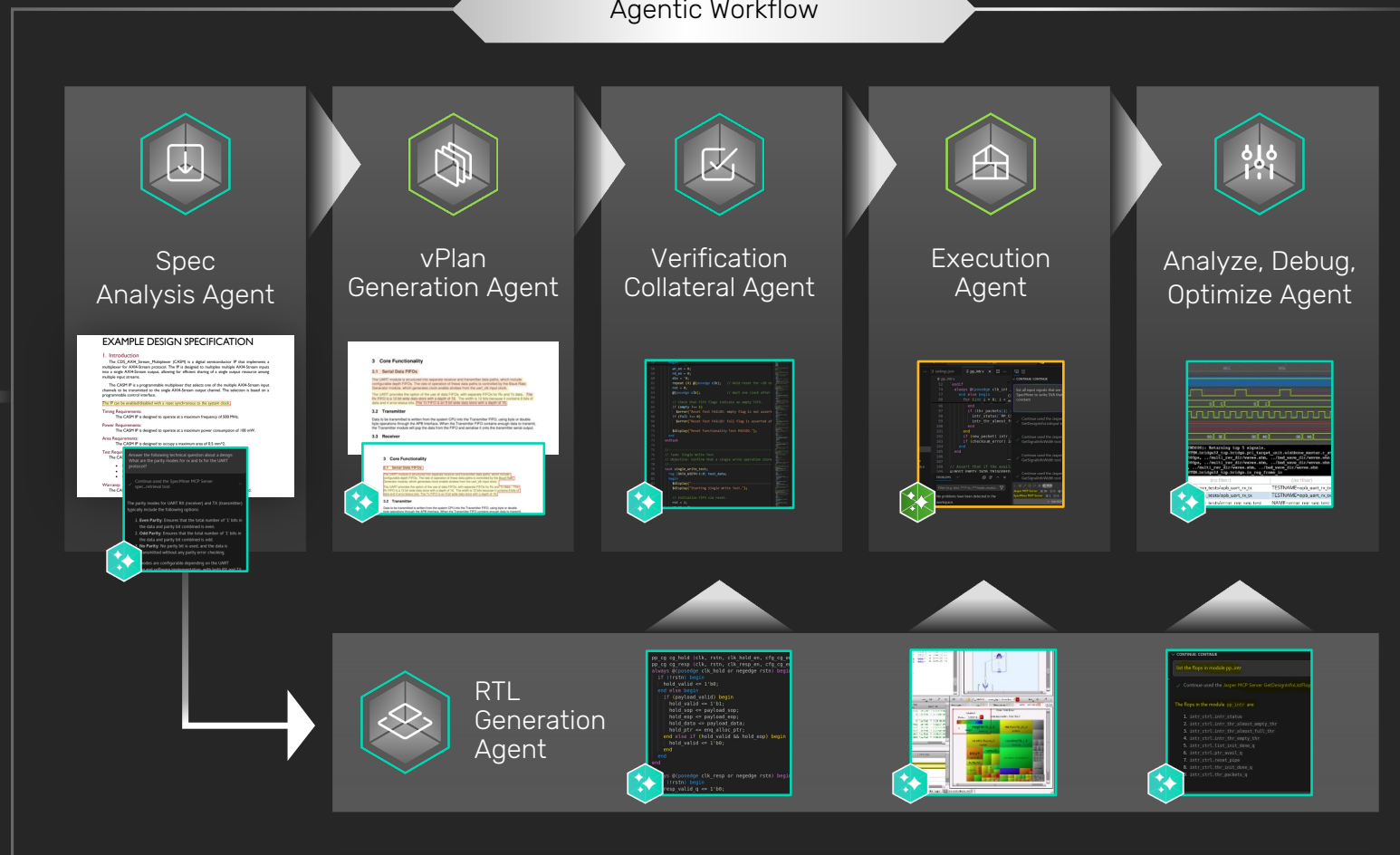
Level 4 - Frontend Silicon Agent



Frontend Silicon Agent

From Spec to RTL Signoff

Agentic Workflow



Meet Chipstack: the World's First AI Verification Engineer

An army of AI junior verification engineers — integrated, production-ready

Business Impact

10× Faster

Testbench and coverage generation

60-80% Automation

Eliminates manual DV engineering tasks

Production Ready

Deployed at Qualcomm, battle-tested at scale

Enterprise Security

On-prem or cloud, your IP stays secure



ChipStack AI Platform

Design
Understanding

Formal
Verification

Unit
Testing

UVM
Generation

AI Chat
Assistant

Coverage
Analysis

Agentic • Integrated • Automated

✓ FULL BATCH MODE CLI

Cadence Integration

Native integration with your tools



Xcelium

Execution Manager, VIPs



Jasper Formal

Property verification



vDebug

Debug and root cause



vManager

Verification management



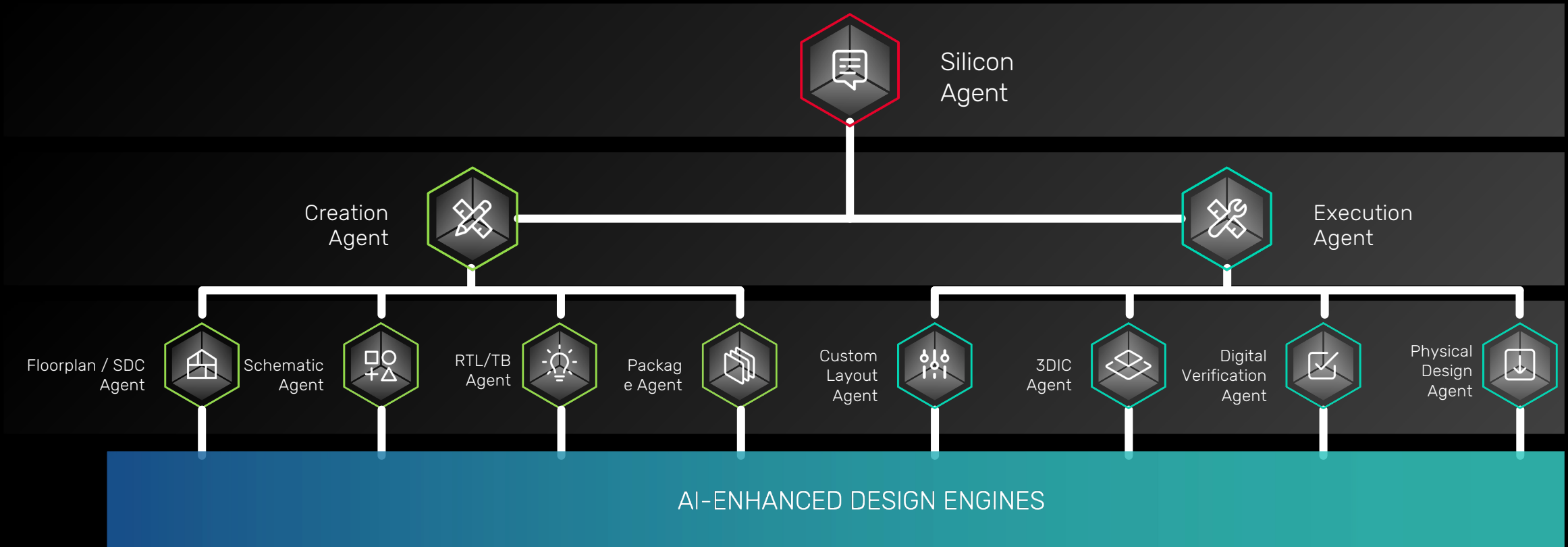
VWDB / UCDB

Waveform and coverage

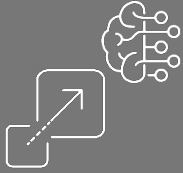
From Design Understanding to Signoff — Fully Automated, Deeply Integrated

Reduce verification time by 10× • Scale your team instantly • Deploy in weeks, not months

System of Agents

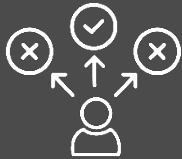


Challenges and Opportunities for Further Work



Scaling complexity

- What works for a FIFO will not scale to real examples
- New reasoning models help close the gap



Preventing auto-accept

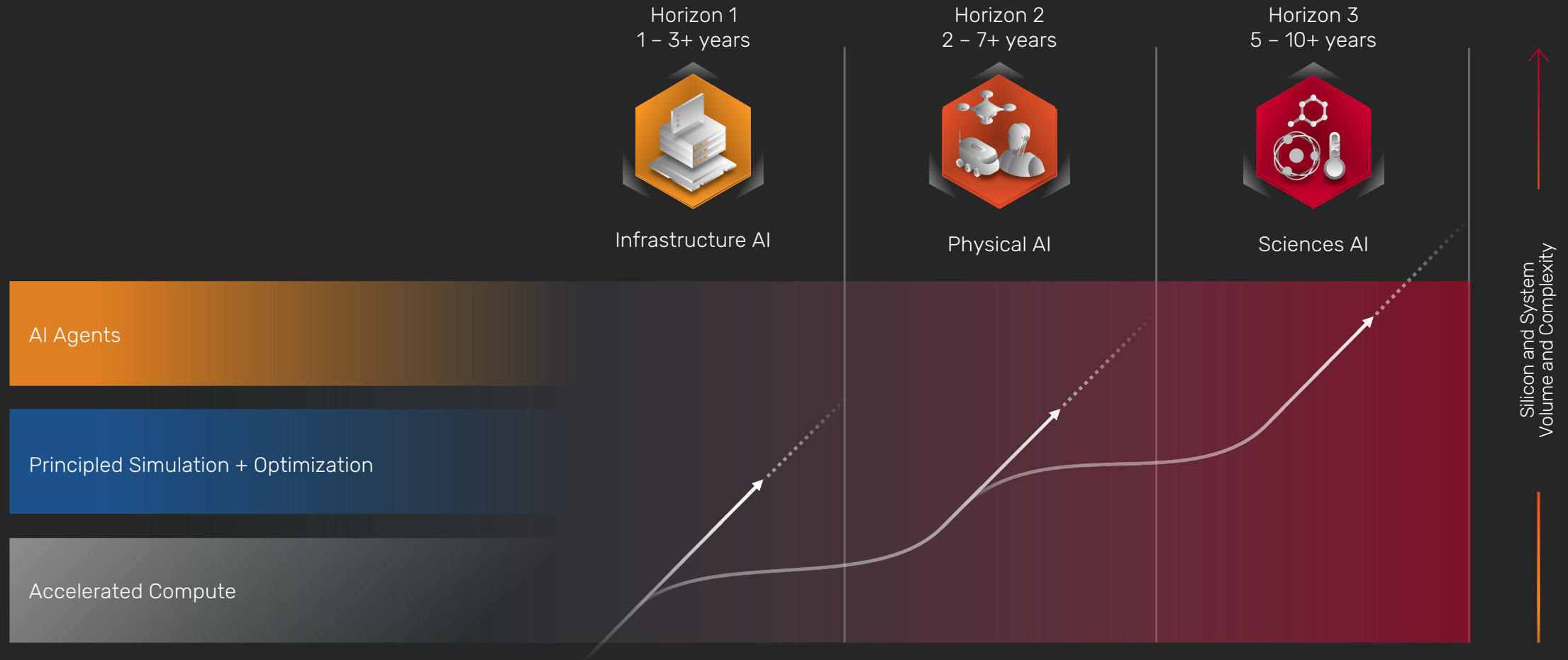
- Easy to just click accept on whatever the AI provides
- How to keep the user involved in an active review



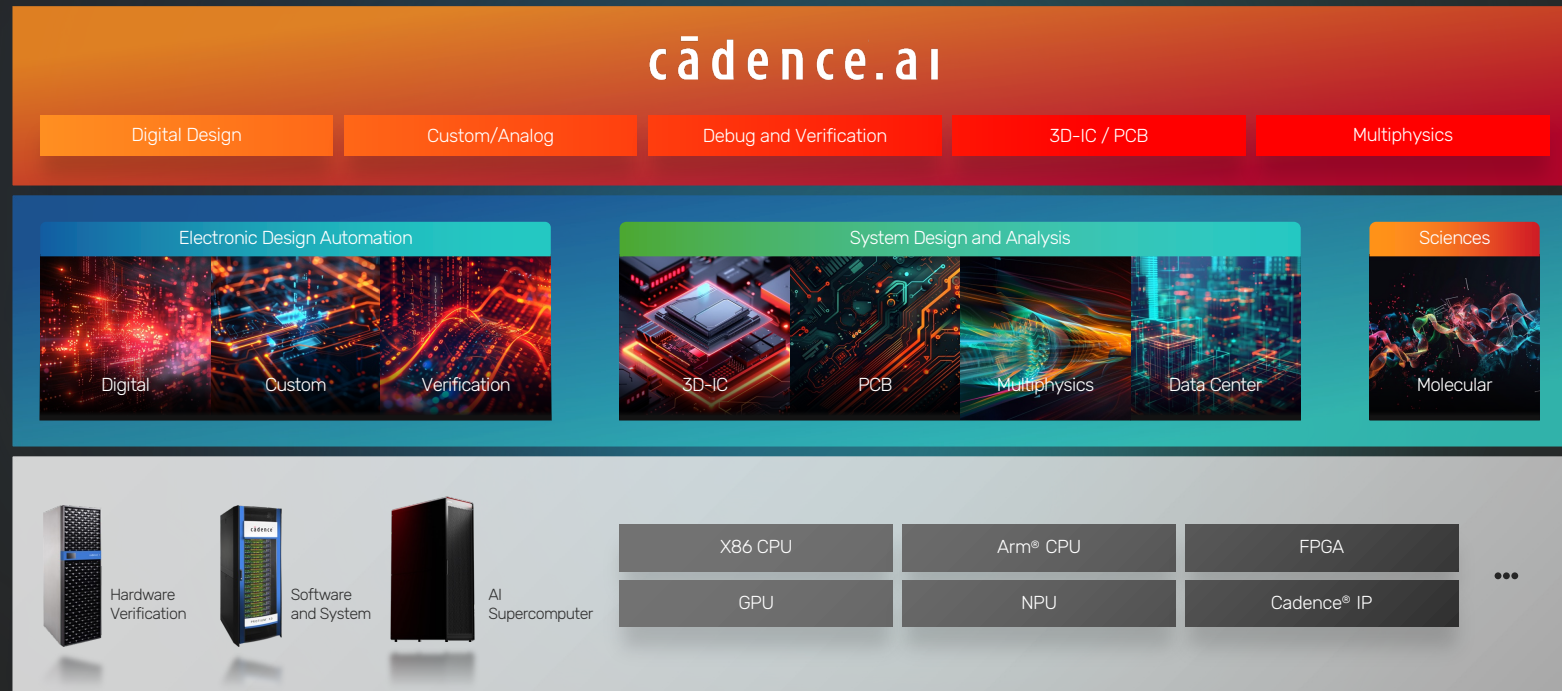
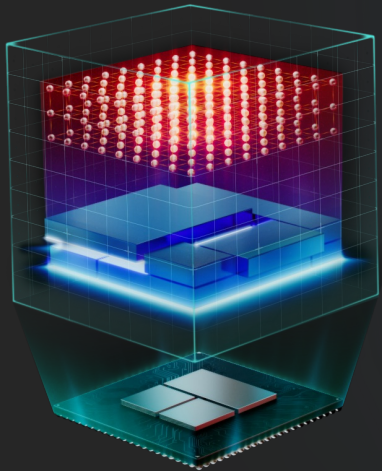
Automated evaluation is hard

- Good benchmarks available for SVA generation
- Much harder for testbench generation

Phases of AI Adoption



Design for AI and AI for Design



AI Agents

Principled
Simulation +
Optimization

Accelerated
Compute

AI Opportunities Are Endless



LEVEL 1
Optimization AI



LEVEL 2
Conversational LLM



LEVEL 3
Complex Reasoning

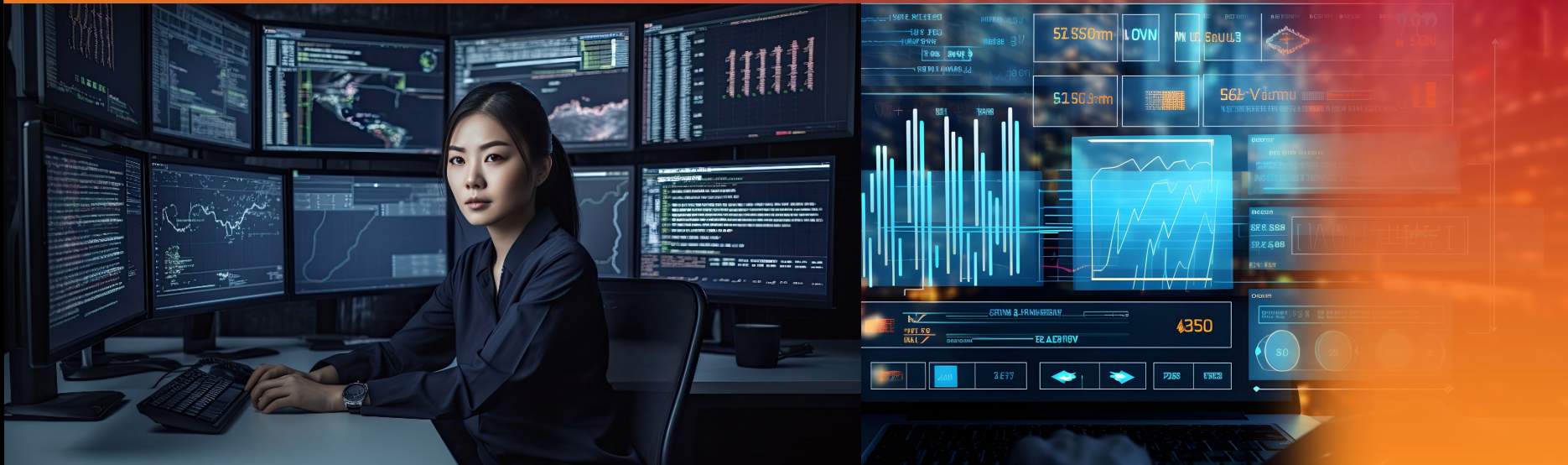


LEVEL 4
Agentic Workflows



LEVEL 5
Autonomy

The Journey to Autonomous Design



Most advanced chips are now designed with AI-assisted tools

Design for AI

cādence[®]

AI for Design