



Welcome to

DESIGNCON[®] 2026

WHERE THE CHIP MEETS THE BOARD

DesignCon Hybrid IBIS Summit
Santa Clara, CA
February 27, 2026



Multi-Agent AI Systems for PCB & Package Design and Validation

Alex Manukovsky, Intel

Joshua Nutzati, Intel

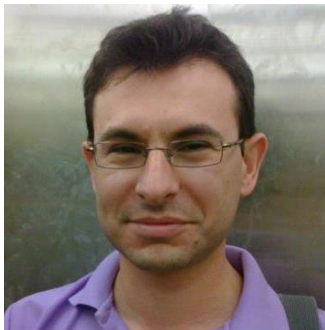
Yuriy Shlepnev, Simberian Inc

Shweta Pujari, MathWorks

Barry Katz, MathWorks



SPEAKERS



Alex Manukovsky, Intel

Principal Engineer Signal and Power Integrity, Intel

alex.manukovsky@intel.com | intel.com |

[linkedin.com/in/manukovsky](https://www.linkedin.com/in/manukovsky)



Joshua Nutzati, Intel

Software and EDA Engineer, Intel

joshua.nutzati.fontaine@intel.com | intel.com

[linkedin.com/in/joshua-nutzati-fontaine-a3bbb3110/](https://www.linkedin.com/in/joshua-nutzati-fontaine-a3bbb3110/)



In This Session

1

Basic Concepts

2

Towards LLM friendly Models and Documentation

3

Multi-agent system You can Replicate

4

Amazing Demo

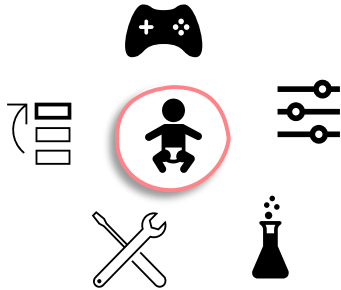


Personal Productivity

Human Centric

AI Assistance

LLM based tools
Nonessential functionality
Optimizers
Advisors
Data fetching
Data Summary

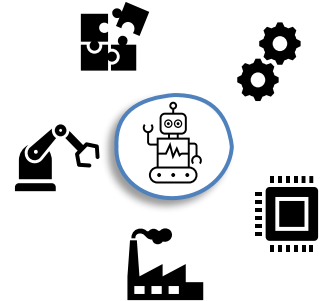


Enterprise Agentic System

Agent Centric

Enterprise Agentic System

AI Agents -Handling the core functionality of the business
Humans- set the goals and boundaries of the desired outcome




In focus

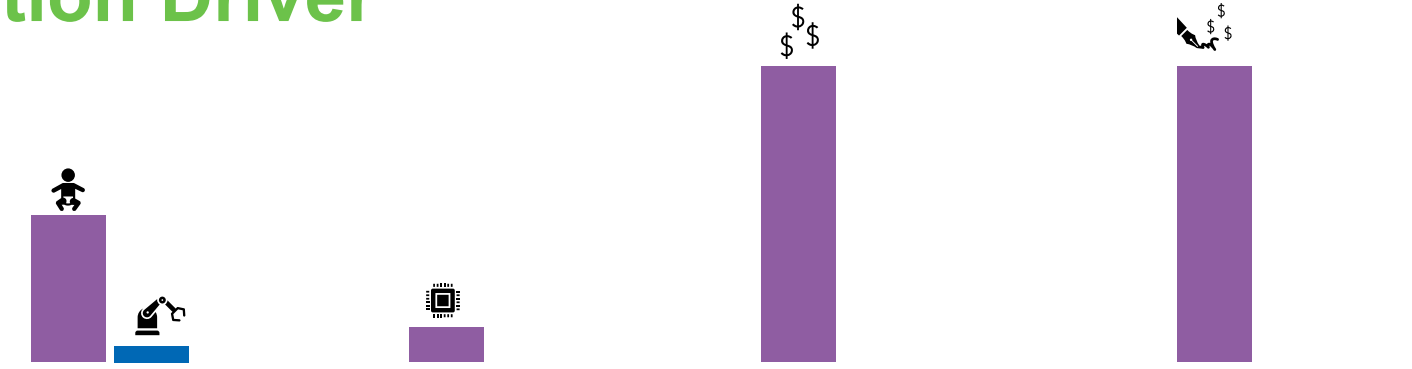
Transformation



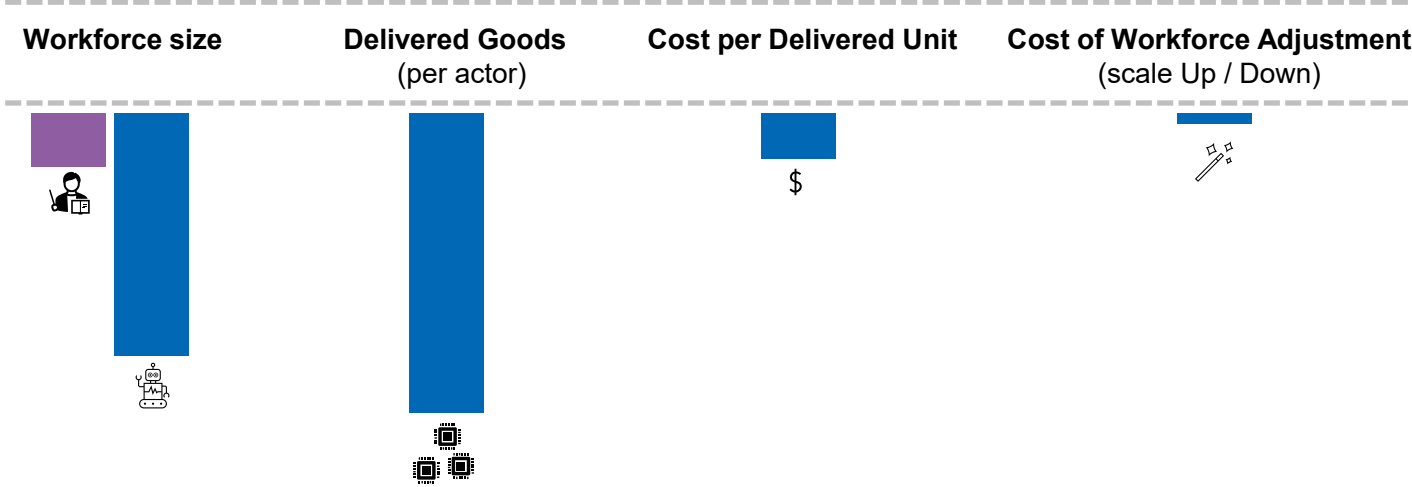
Transformation Driver



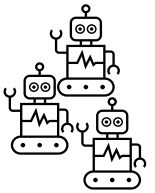
Human Centric




Agent Centric



This is about AI agents for:



- Helping your kids with Homework
- Summarizing documents
- Writing emails
- Searching the web



Personal Productivity

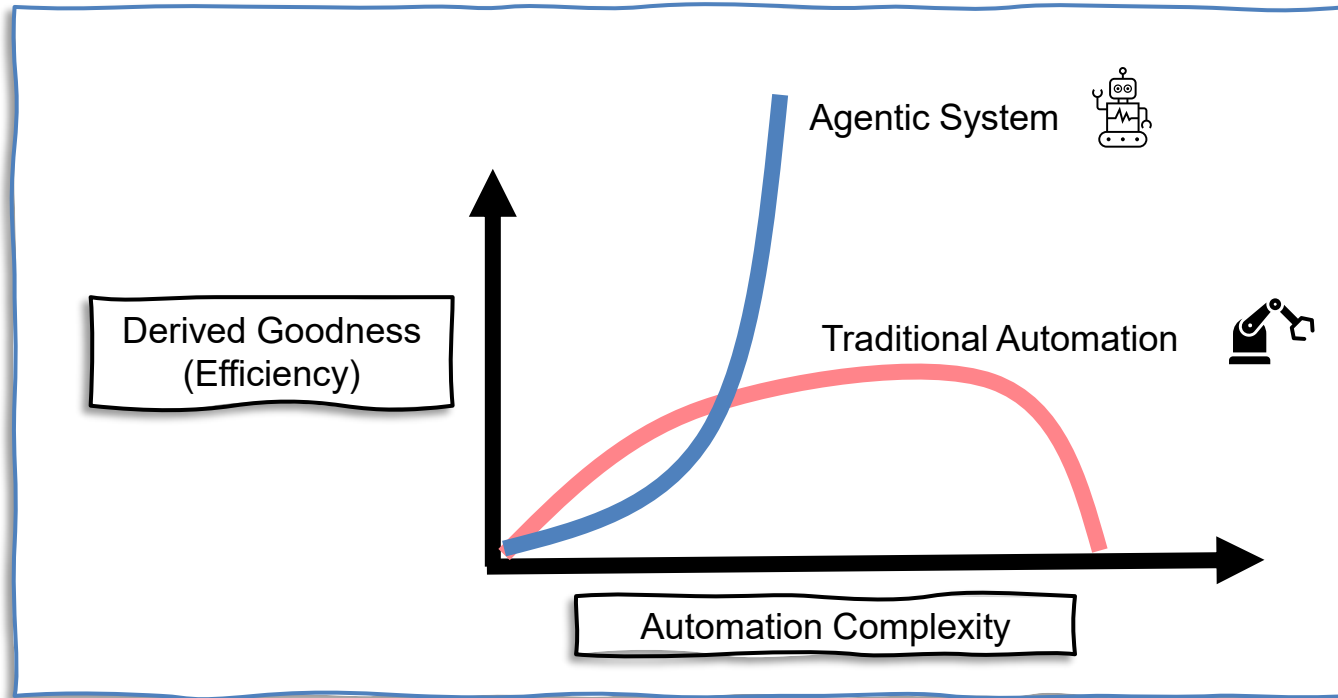
- **Enterprise Grade(AI agents)**
- Predictable
- Reliable
- Secure
- Results Confidence & Trust (Production level)
- Expandable & Tracible flows
- Standard Compliant Process Execution



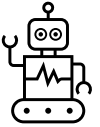
**Core Business
for Enterprise**



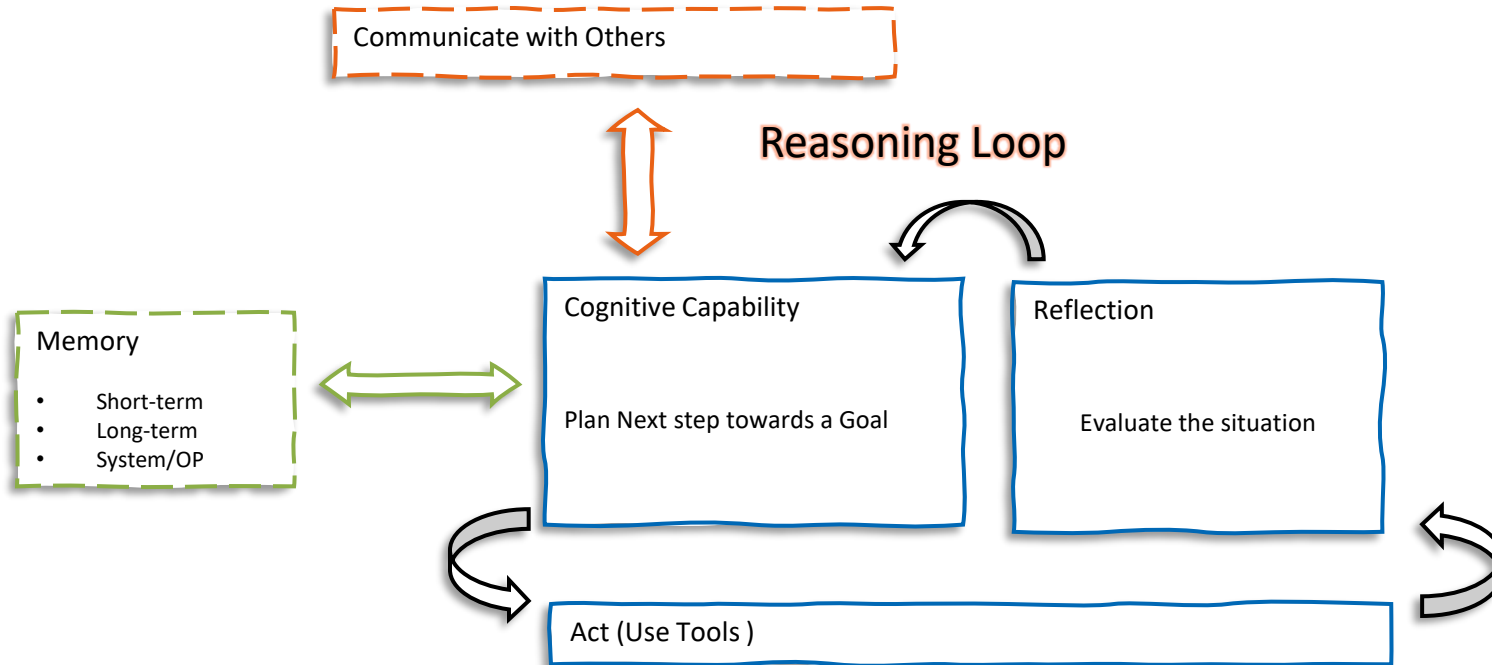
Automation Efficiency



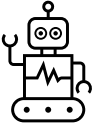
Agent Blueprint



- Has a Goal
- Autonomous
- Flexible
- Tool Use



Predictable Path



- Has a Goal
- Autonomous
- Flexible
- Tool Use

Communicate with Others

Reasoning Loop

Memory

- Short-term
- Long-term
- System/OP

Cognitive Capability

Plan Next step towards a Goal

Work according to expected Flow

Reflection

Evaluate the situation

Skills

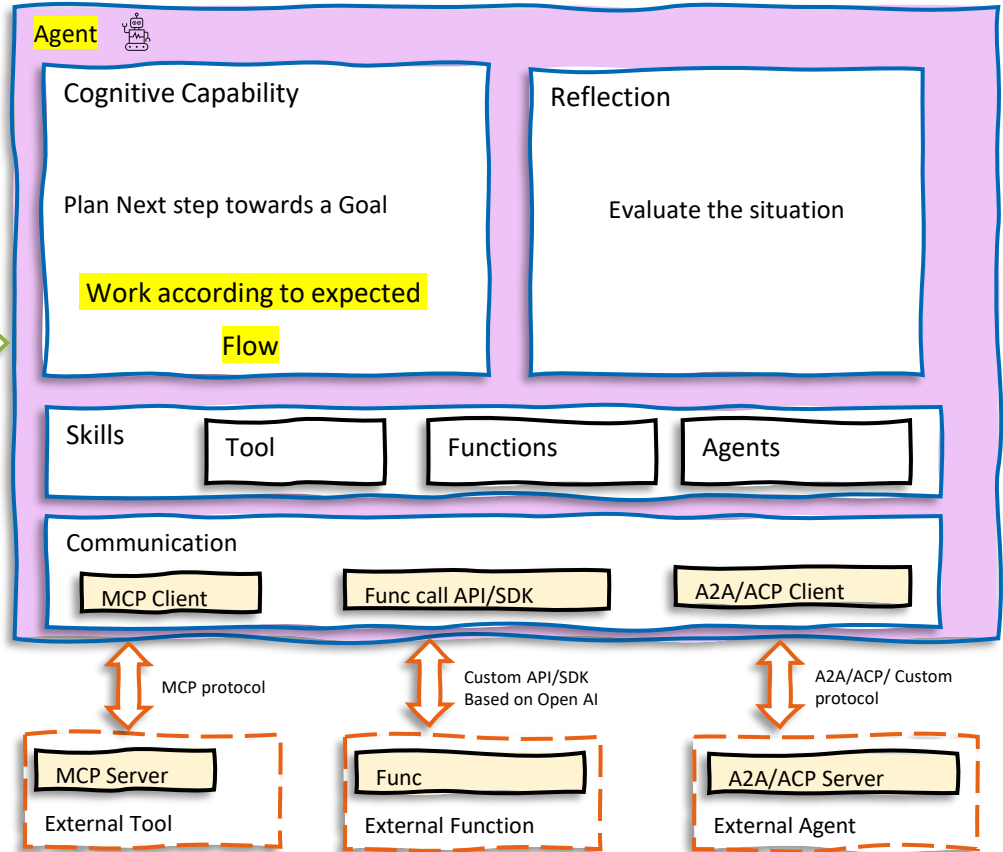
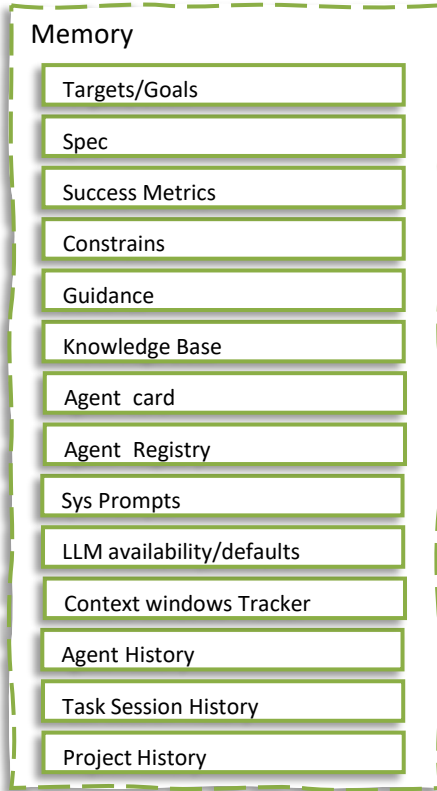
Tool

Functions

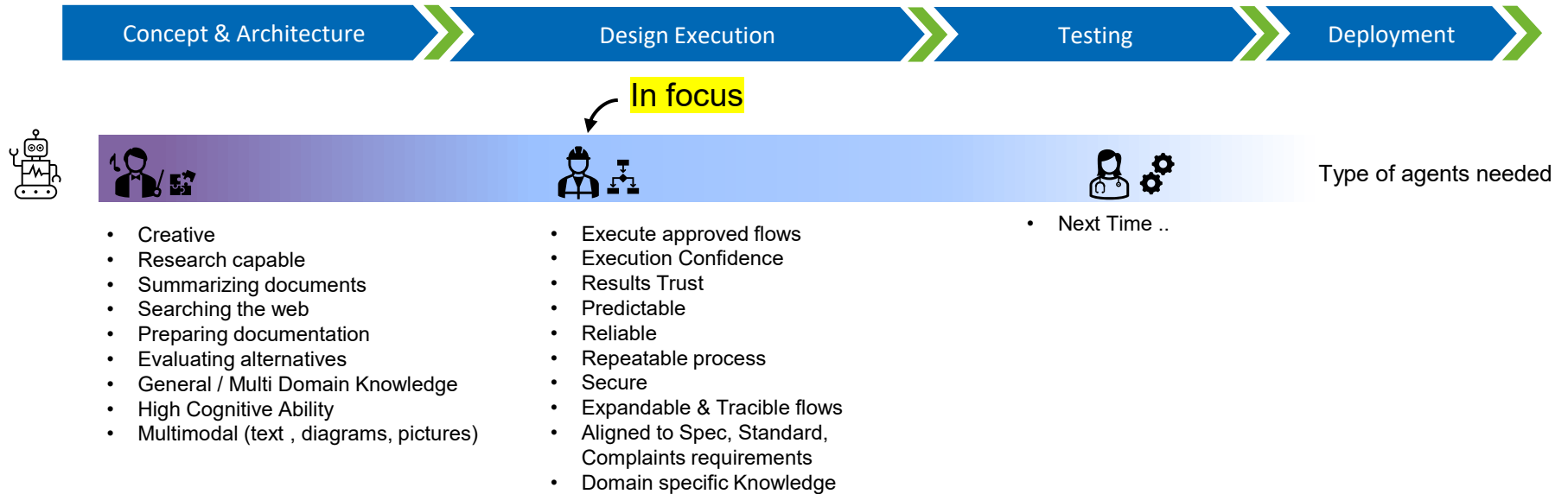
Agents



Concept Arch

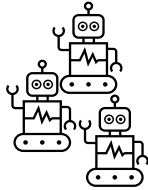


AI Agents and Product Design Life Cycle

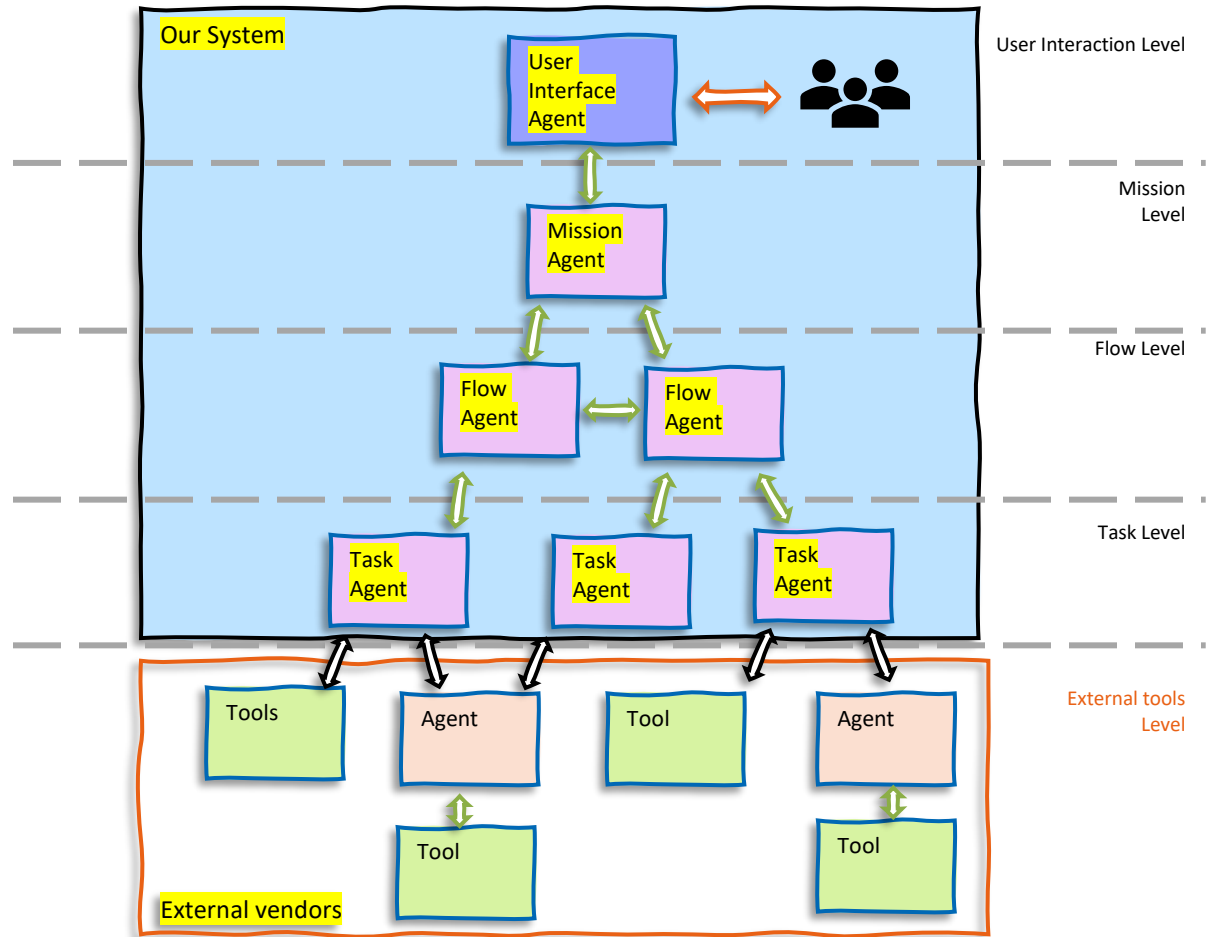


Agentic System

We only (almost) know how to manage teams of Engineers

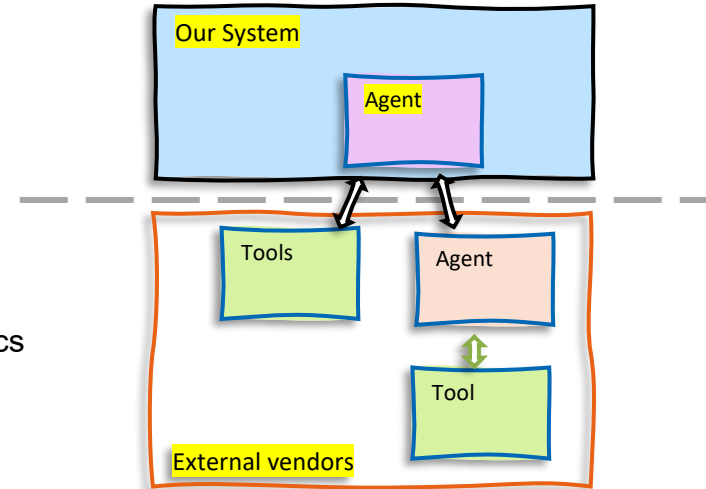


Mimic Engineering Teams
Resemble Corporate structure



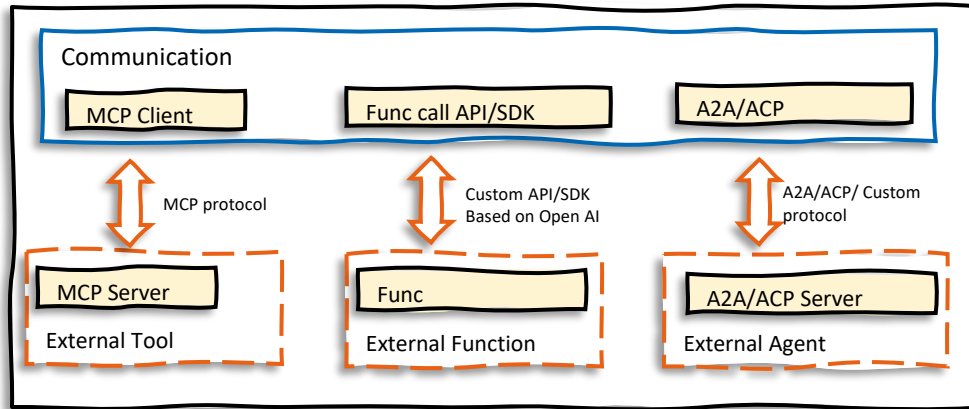
Tool Usage- In Agentic World

- Operating tools properly is a burden/Imposed cost on a User
- The tool Vendor is expected to handle proper operation and settings
- Tool user to describe the high-level goals , constraints and success metrics



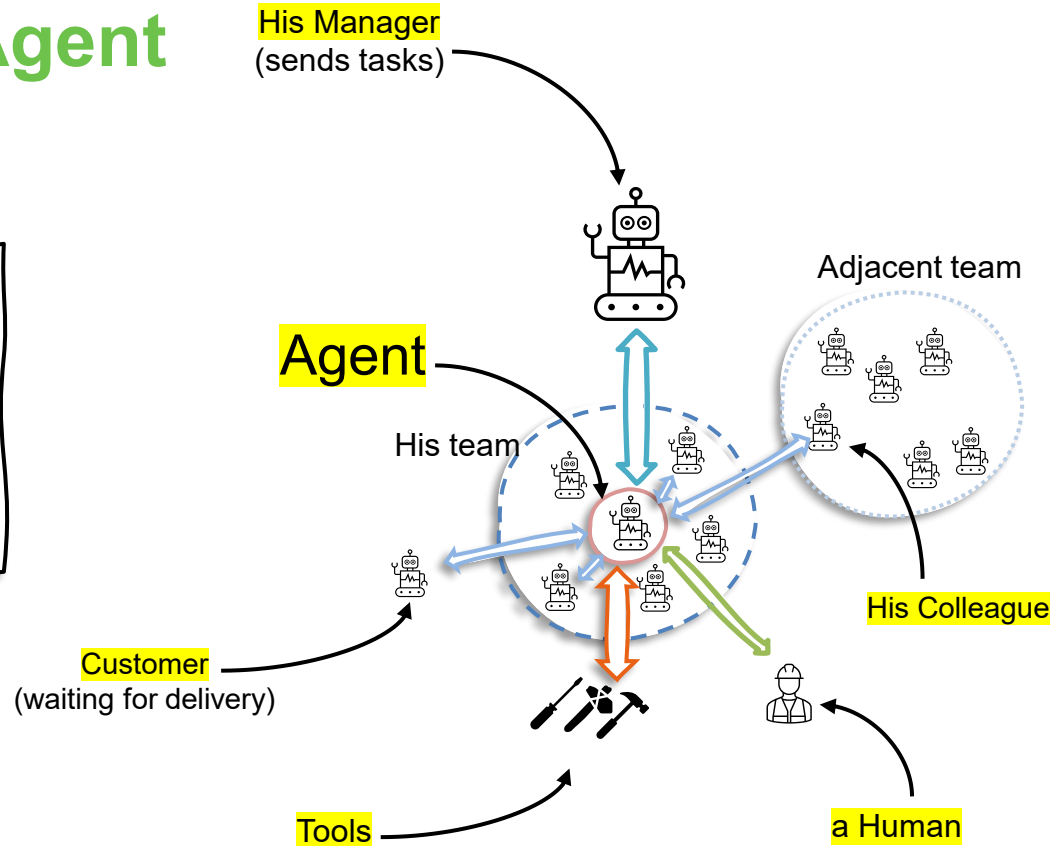
Agentic Communication

- Agent operating tools Designed for Humans → MCP / Direct SDK calls
- Agent asking the tool vendor's Agent to operate a tool → A2A (“formal Communication”)
- Agent Communicating with peers → Custom Protocol

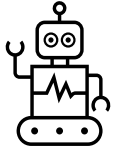


Social Circle of an AI Agent

- We only (almost) know how to handle Engineers
- Mimic Office Communication
- Flexible (Need to know basis)



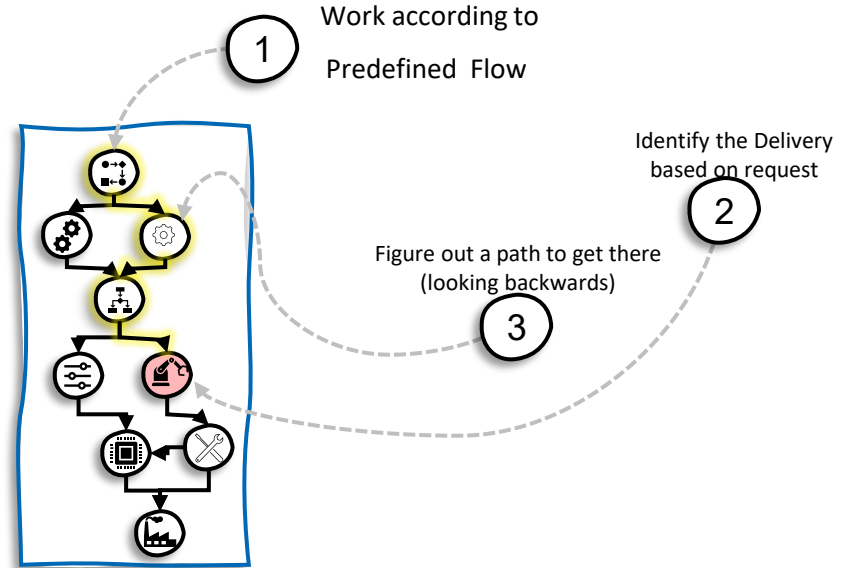
Execution Confidence



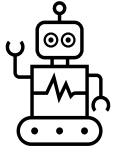
An Agent We Can Trust

Define Deterministic Flows According to:


- Best Know Methodes
- Industry Standards
- Verified practices
- Nested Hierarchy – Agentic Cals

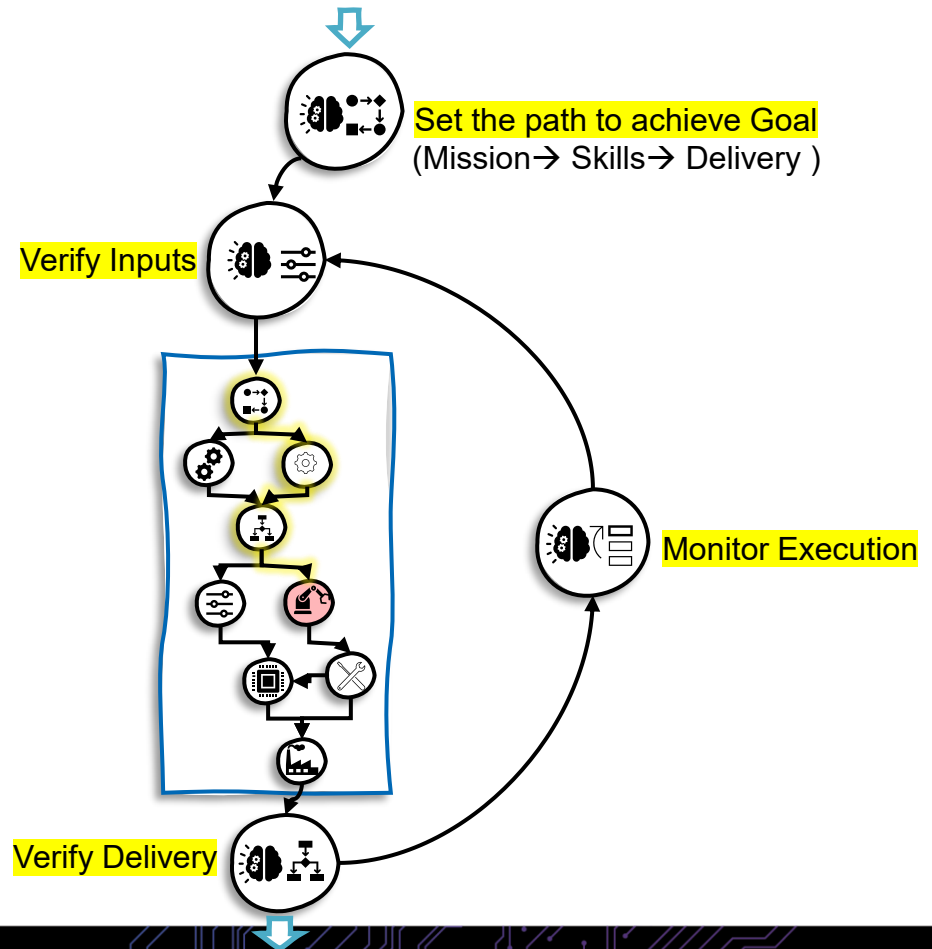


Predictable Autonomy

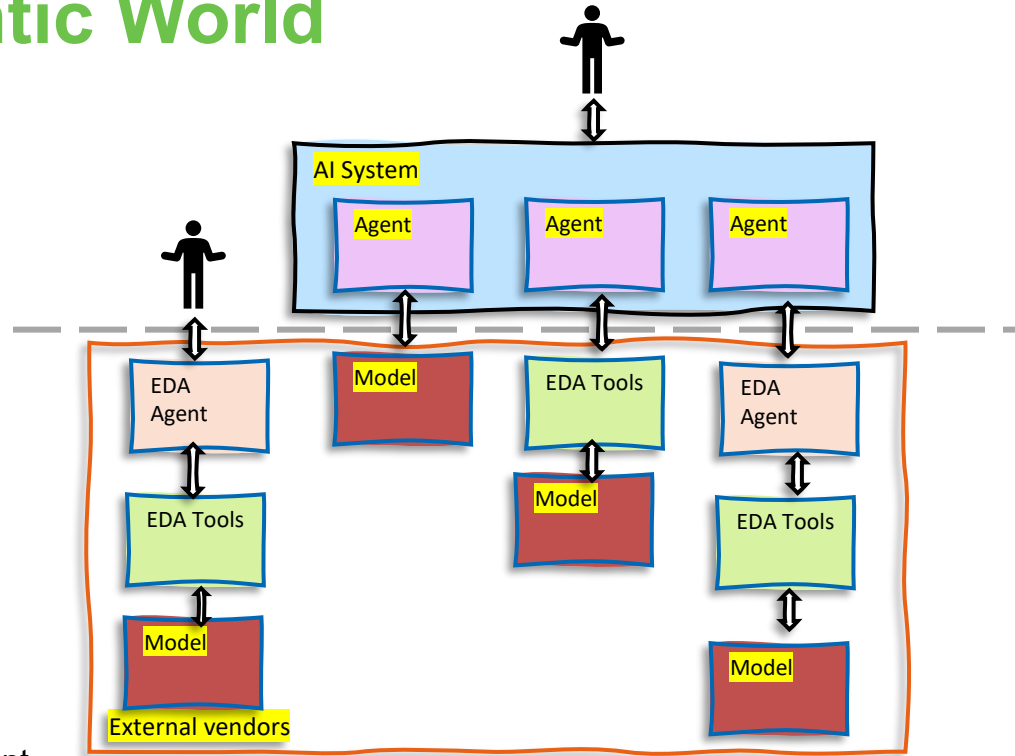
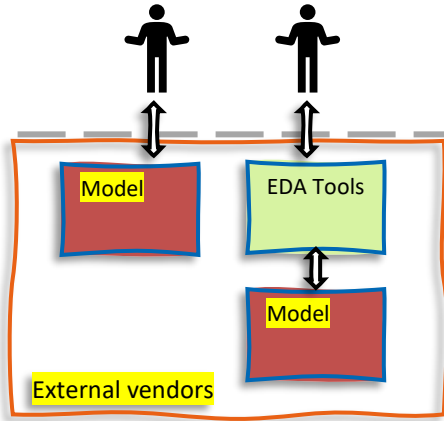


An Agent We Can Trust

- Deterministic Flows
- Add Reasoning loop
- Cognitive Capability (LLM )
- Orchestration
- Error handling
- Quality control



Models Usage- In Agentic World



- Most of the model usage is thru an LLM
- Engineers avoid direct usage , prefer thru an LLM/Agent
- Most of EDA tools Agentic



Towards LLM friendly Models and Documentation

LLM Friendly Format

Plaintext

```
[Model]      DDR5_DQ
Model_type   I/O
...
[Voltage Range]  1.1V      NA      NA
[Pullup]
```

	Voltage	I(typ)	I(min)	I(max)
	-1.10	-40.0m	-38.0m	-42.0m
	-1.00	-38.0m	-36.0m	-40.0m



LLM Friendly Format

JSON

```
{
  "model": {
    "name": "DDR5_DQ",
    "type": "I/O",
    "voltage_range": {
      "typ": 1.1,
      "min": null,
      "max": null
    },
    "pullup": {
      "columns": ["voltage", "current_typ", "current_min", "current_max"],
      "data": [
        [-1.10, -0.040, -0.038, -0.042],
        [-1.00, -0.038, -0.036, -0.040]
      ]
    }
  }
}
```



World First PCB/Package Design Agent

- Trust Results confidence (Production level)
- Capable in following approved flows
- Flexible
- Local /Cloud or Hybrid Deployment
- LLM Agnostic
- Easy to Transition (Adaptation)
- Scalable (multiagent)
- Cost Sensible to operate for large scale deployments
- Secure by design ,access / permission/Isolation (encapsulated capabilities)



Key Concept

Taskmaster Framework

Connects all the ingredients
Parent class for all agents

Contains Methods for:

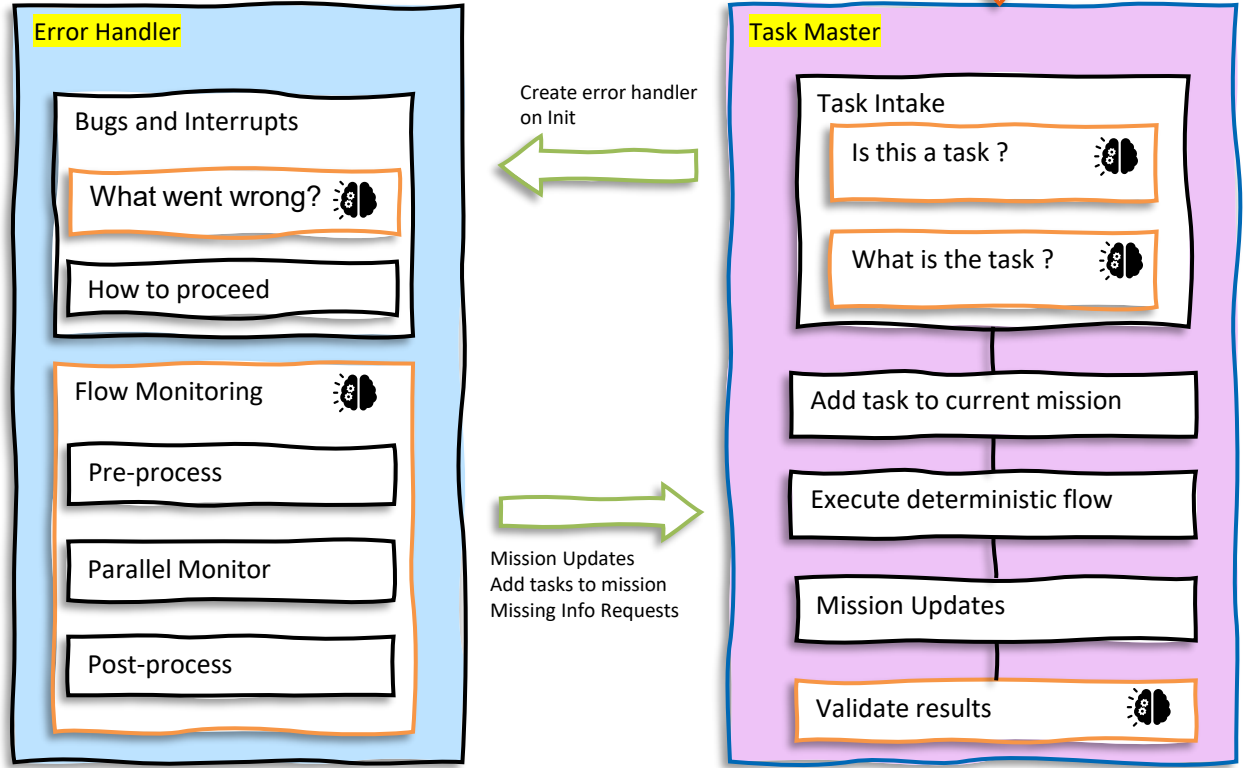
- Managing the state machine
- Performing missions
- Initiating child agents
- Running functions



Mode Of Operation

Operate in parallel

Communicate with Caller
(missing info /fails/updates/etc.)



Memory Cards(Examples)

System/OP Memory

Agent Card

Description of agent and capabilities :

- Agent configurations
- Skill card location
- Preset prompts
- Properties and Variables
- Context and LLM Cache
- State

System/OP Memory

Skill Card

(Excel)

A table with the skills and functions contained in the agent's flow: definitions for the state machine and all supplementary info for the LLM to identify tasks

Skill – is a capability exposed to outside

Function – Internal flow operation, not exposed outside the Agent or to the LLM

Short-term Memory

Mission Card

(MATLAB struct)

Mision Log

Updated dynamically throughout the run from mission updates.

Current task

Task priority

State machine operation (overbites /state changes)

Update values for the Agent setup

**Functions, other agents, and the error handler can all provide input for mission updates, which are then validated before being applied to the mission card.*



Memory Cards (Examples)

System/OP Memory

State Card

(Table- when operational)
(Excell for backup)

**State of all nodes and functions
of the agent**

**Excell Used for loading and saving
agent state*

System/OP Memory

Error Handler Card

(Excell)

**A table with the skills and
functions that the error handler
agent has in its flow**

Used for Flow monitoring or
Agentic Debug

**Creates separation from main flows*

Long-term Memory

Project Card

(Struct when operational)
(Excell for backup)

**All required context for defining
the environment in which the
agent is operating**

All of the broader context
, definitions and data necessary for
running the current project

**Excell used for loading and saving
project data*



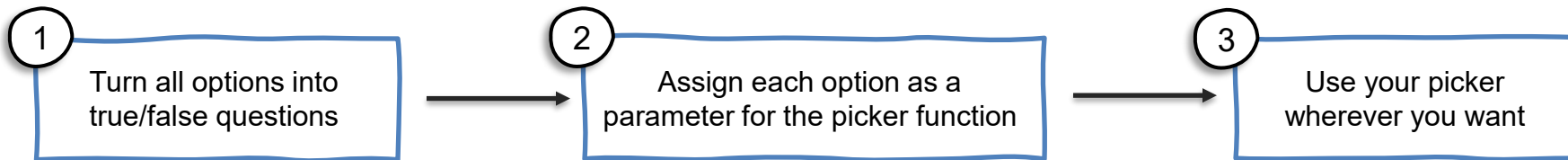
Picker Agent (Class)



- How do we reduce hallucinations in LLM managed decision making?
- How can we make a dynamically updatable structure for using tool calling with our agents?
- How can we ensure that the correct context is being leveraged in agentic operations.

Parameter Name	Type	Description
Parameter_1	Boolean	Return 1 if X, and 0 otherwise
Parameter_2	Boolean	Return 1 if Y, and 0 otherwise
...
Parameter_n	Boolean	Return 1 if Q, and 0 otherwise

It's Pickers all the way down!



Picker Agent Examples



QueryType_Picker - is the last message a:

1. NormalQuestion: Conversation or question, discussion that does not require performing action
2. FunctionCall: Request to perform an action, as might be achieved by [CONTEXT]

Function_Picker - is the required function:

1. Function_A: Does [QUERY] relate to running of [DESCRIPTION_A], [TAGS_A]
2. Function_B: Does [QUERY] relate to running of [DESCRIPTION_B], [TAGS_B]

Error_Picker - is the exception of type:

1. MissingInputs: Did the function call require a variable that did not exist within the workspace
2. FileNotFound: Was the function call referencing a file that was not on path



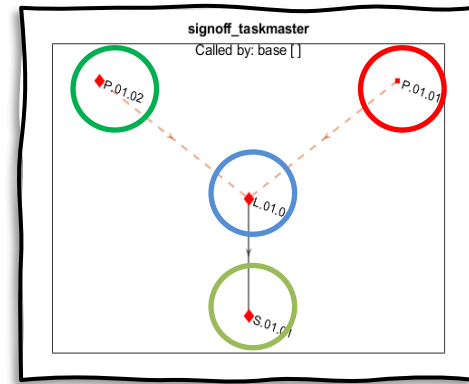
Taskmaster Agent (Class)



1. Get a request to perform a task
2. Determine how the Mission correlate to His Skills and capabilities
3. Determine the goal target state - deliverables
4. Identify the path to achieve the goal according to the approved/supported workflows
5. Execute and monitor the flow
6. Validate results
7. Pass up results or errors



Signoff -Taskmaster Skill Card



Unique Identifier used for creating nodes and mapping dependencies

Indicated If this step is Agentic, or normal script? Skill, or an intermediate function.

ID	Name	Type	Description	Tags	Dependencies
P.01.02	call_ProjectStart	Agent Skill	Call an agent to start the flow	Project, Start, Agent, Inputs	
P.01.01	start_project	Function	Function for starting the flow	Project, Start, Inputs	
L.01.01	call_SimbExtraction	Agent Skill	Call an agent to perform Simbeor extraction of a board	Extraction, Simbeor, Board, Layout, Nets	P.01
S.01.01	call_SparamReport	Agent Skill	Call an agent that can analyze an SnP analysis file for selected s-parameters, output presentation based upon definitions in the file	S param, snp, Scattering Parameter, report, agent, AI, LLM	L.01.01

The name of the step, which corresponds to a specific script

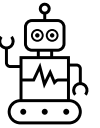
To ensure both the user and the LLM comprehend the function of this node.

Tags To help LLM identify relevant Skills

Which nodes need to be executed prior to this node?



In This Demo-World First Multi-Agent system for SI/PI



We will demonstrate

- Fully operational Multi-Agent system
- Production Level Confidence

Use case :

- Signal Integrity analysis for Post Layout extraction for PCIe channels on PCB for Signoff
- IBIS –AMI Simulation and PAM4 eye analysis

We going to :

1. Define the mission
2. The agents will run the analysis
3. Generate report



SI PCB Signoff with Multi-Agent system



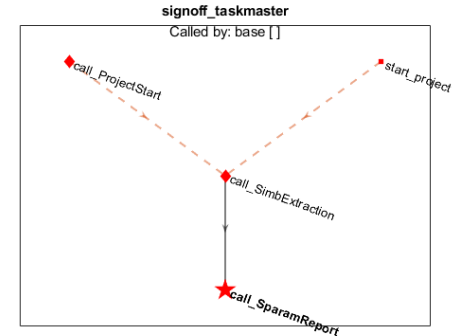
1 Intake

1. Identify what Skill relates to task
2. Check dependencies for that Skill
3. Find starting Node
4. Pull task list from plan

2 Define Flow

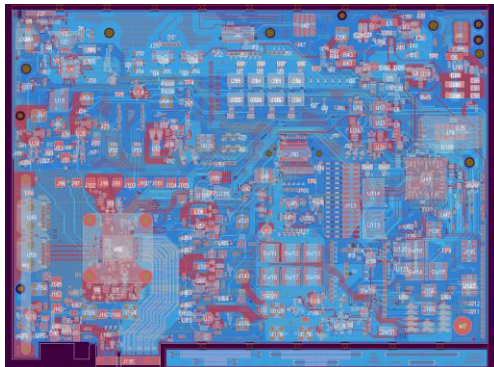
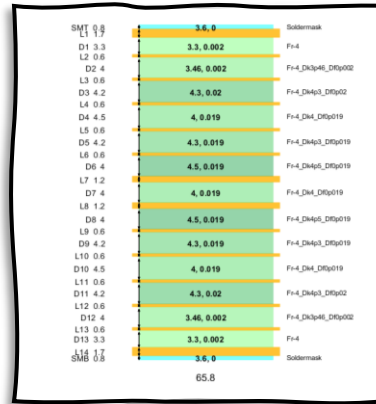
1. Collect the Required Inputs
 1. Project name and locations
 2. Board file
 3. Stackup
 4. Nets of interest
 5. Identify the Interface and compliance specifications
 6. Identify the Analysis type
2. Run the Layout analysis with Simbeor
 1. Write the Instructions for the Simbeor tools
 2. Open Simbeor
 3. Run Simbeor Instructions
3. Run the IBIS-AMI analysis with MATLAB
 1. Write the Instructions for the Simbeor tools
 2. Open Simbeor
 3. Run Simbeor Instructions
4. Prepare a report
 1. Get a correct report template
 2. Run Compliance analysis for results (analyze results vs spec)
 3. Generate to PowerPoint

3 Flow Execution

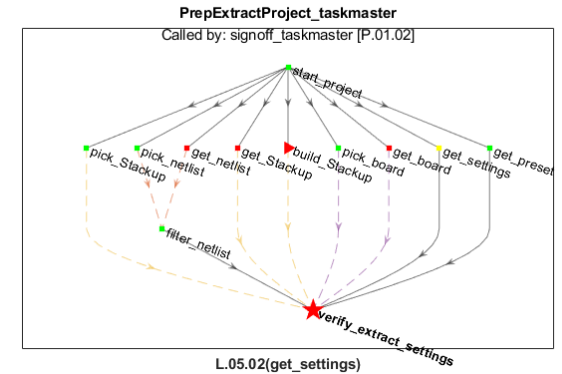
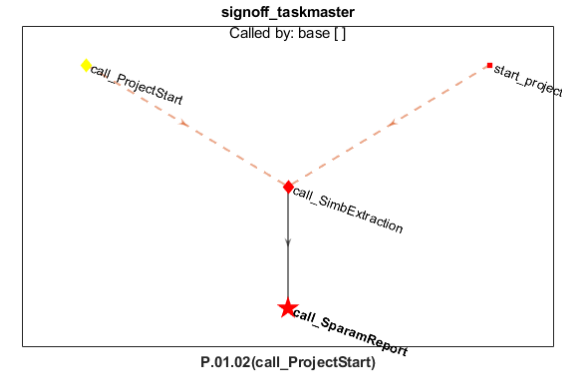


Step 1. Collect the Required Inputs

1. Project name and locations
2. Select Board file
3. Select Stackup
4. Select Nets of interest
5. Identify the Interface and compliance specifications
6. Identify the Analysis type

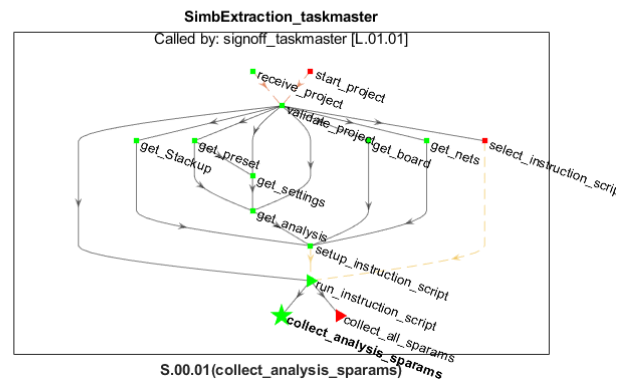
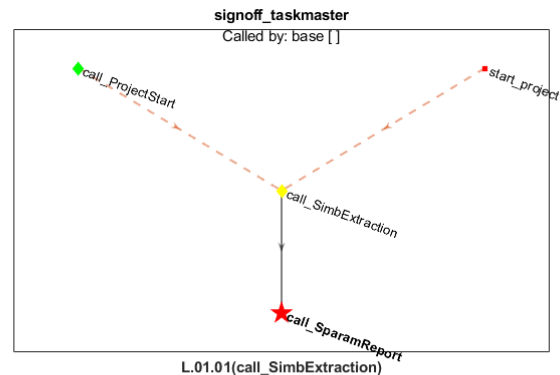
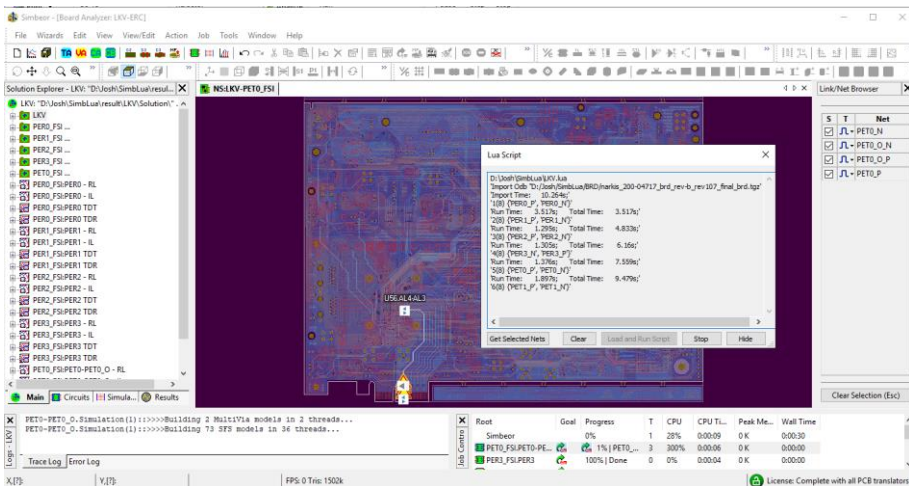


	A	B	C	D	E	F
1	Netname	DIFF	Netclass	Netgroup	EP1	EP2
2	PER0_P	P	PCle	RX	J195	U56
3	PER0_N	N	PCle	RX	J195	U56
4	PER1_P	P	PCle	RX	J195	U56
5	PER1_N	N	PCle	RX	J195	U56
6	PER2_P	P	PCle	RX	J195	U56
7	PER2_N	N	PCle	RX	J195	U56
8	PER3_N	P	PCle	RX	J195	U56
9	PER3_N	N	PCle	RX	J195	U56
10	PET0_P	P	PCle	TX	J195	U56
11	PET0_N	N	PCle	TX	J195	U56
12	PET1_P	P	PCle	TX	J195	U56
13	PET1_N	N	PCle	TX	J195	U56
14	PET2_P	P	PCle	TX	J195	U56
15	PET2_N	N	PCle	TX	J195	U56
16	PET3_N	P	PCle	TX	J195	U56
17	PET3_N	N	PCle	TX	J195	U56



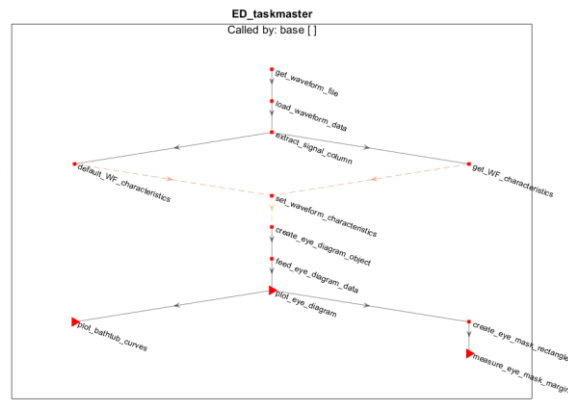
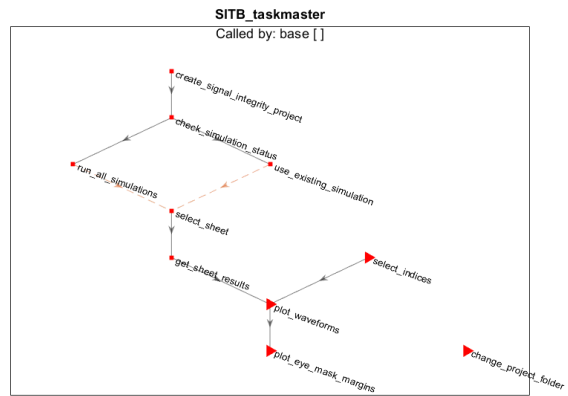
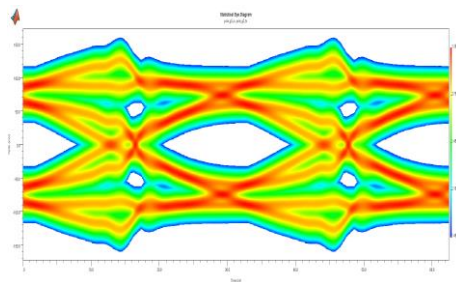
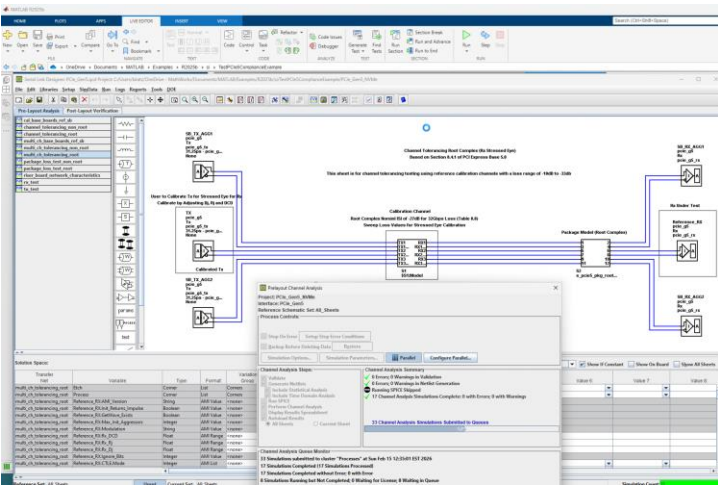
Step 2. Run the Layout Simulation with Simbeor

1. Verify Inputs from previous agent
2. Collect correct instructions templates
3. Write the Instructions for the Simbeor tools
4. Open Simbeor
5. Run Simbeor Instructions



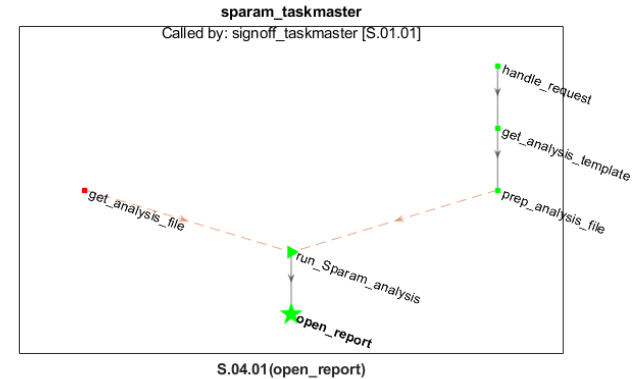
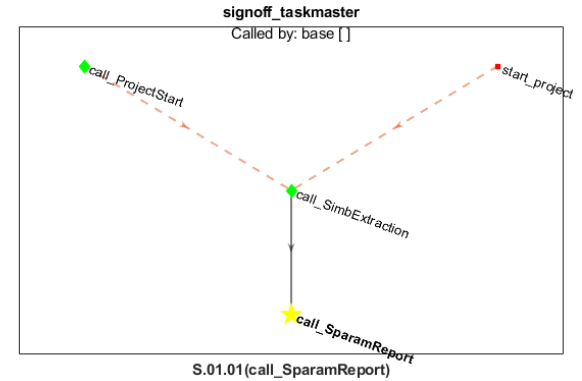
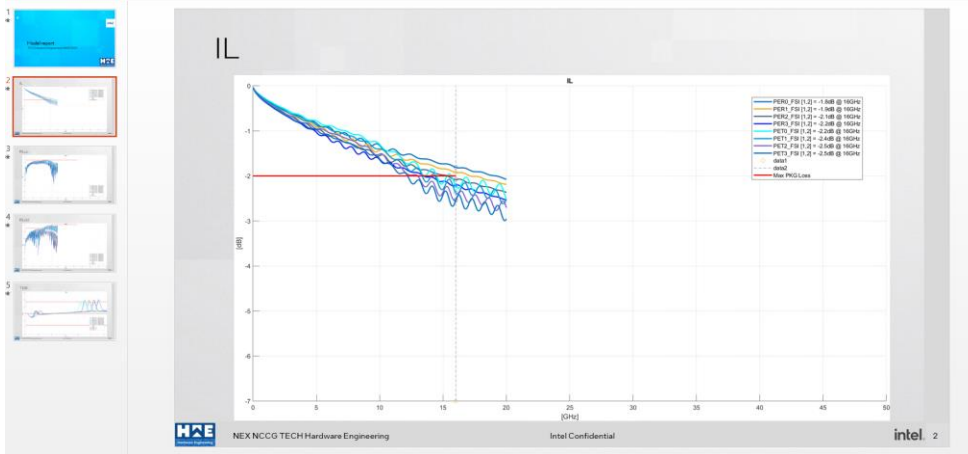
Step 3. Run IBIS-AMI Simulations with SI Toolbox

1. Verify Inputs from previous agent
2. Collect correct instructions templates
3. Write the Instructions for the SI Toolbox
4. Open SI Toolbox
5. Run SI Toolbox IBIS-AMI Simulations

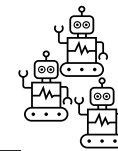


Step 4. Prepare a report

1. Collect results from all agents
2. Get a correct report template
3. Run Compliance analysis for results (analyze results vs spec)
4. Generate to PowerPoint

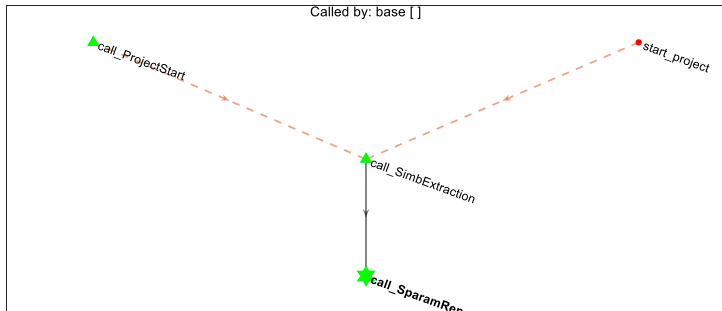


Multi-Agentic System



signoff_taskmaster

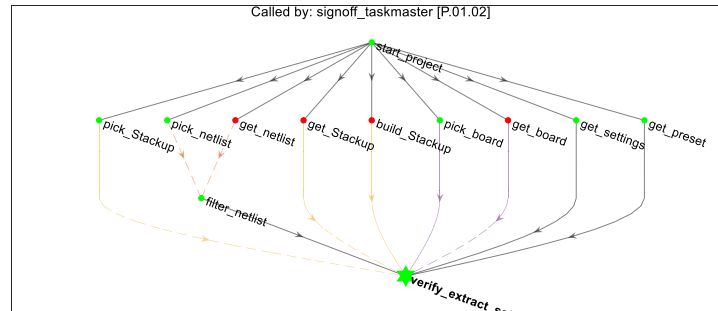
Called by: base []



S.01.01(call_SparamReport)

PrepExtractProject_taskmaster

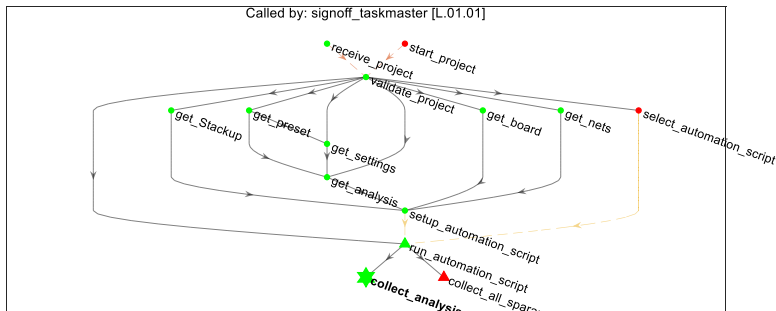
Called by: signoff_taskmaster [P.01.02]



L.06.01(verify_extract_settings)

SimbExtraction_taskmaster

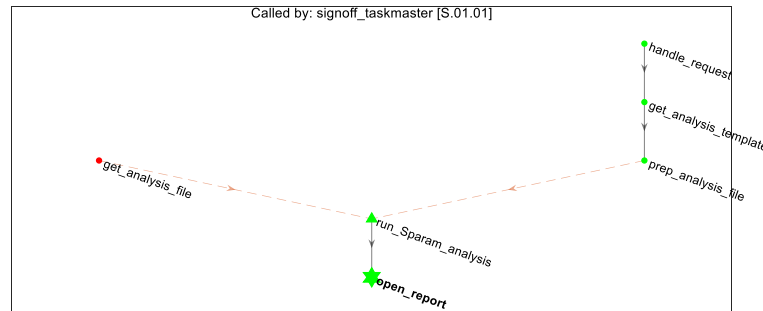
Called by: signoff_taskmaster [L.01.01]



S.00.01(collect_analysis_sparams)

sparam_taskmaster

Called by: signoff_taskmaster [S.01.01]



S.04.01(open_report)



Thank you!



QUESTIONS?

