



Agentic AI Foundation

AGNTCon + MCPCon Japan

Where the agentic stack is being built.

Build Agent Assembly Line

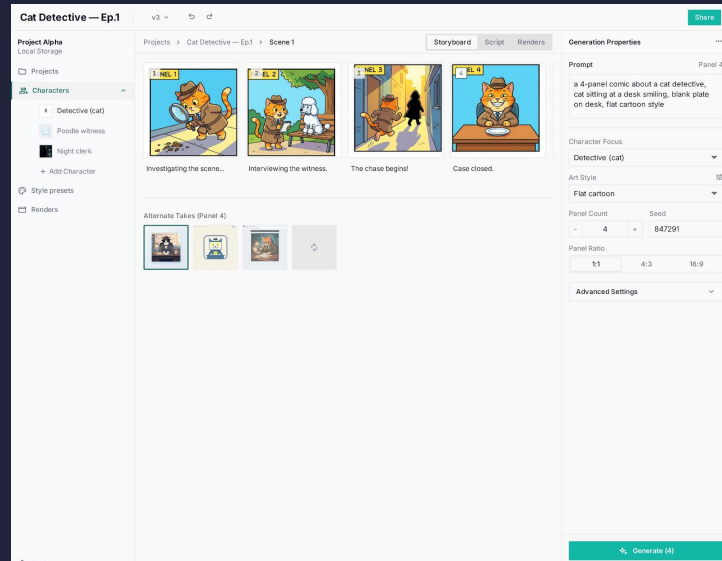
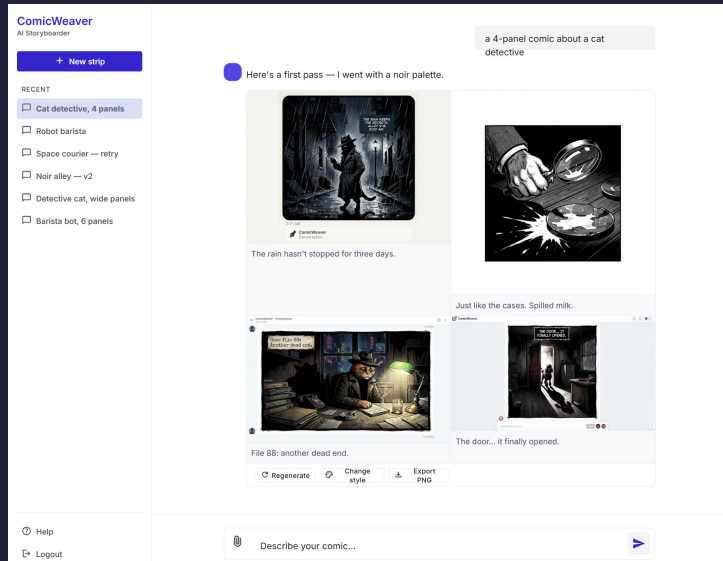
Addressing same input, but different outputs

KAN WANG @ Raytone

AGNTCon + MCPCon Japan · September, 2026

We run same prompt twice

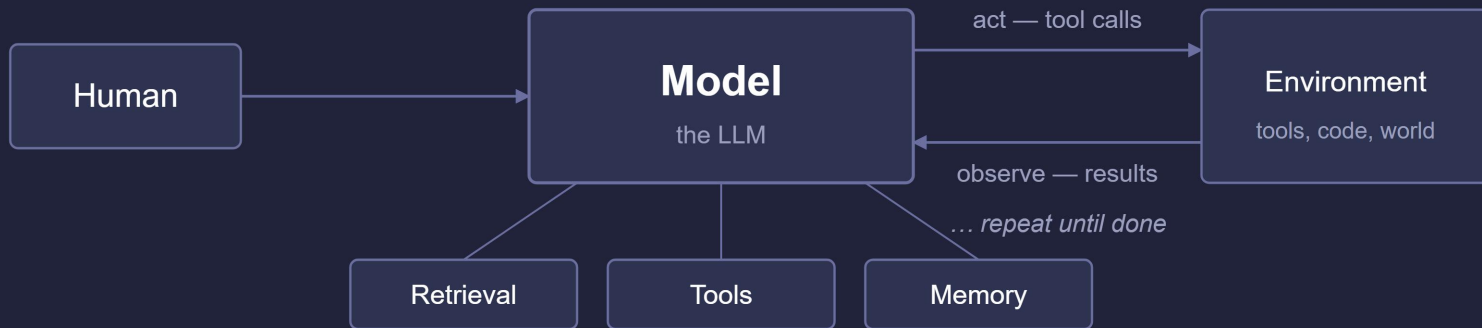
with same model, same harness.



Two tasks are all completed, but the results are totally different.

Looking in to agent: model calling tools in a loop

it is widely agreed that agents processing tasks by looping tool calls.



Adapted from Anthropic, "Building Effective Agents" (2024)

Whatever that task is, agents consist of same components.

Where the agentic stack is being built.

Agent components

Every task processed by same components

Model

For example: DeepSeek, GLM, Qwen

Interface

Message Rendering, Streaming

State & Memory

Session State, Long-term Memory, Retrieving

Tool Integration

MCP, Sandbox, Search, Database Driver

Environment

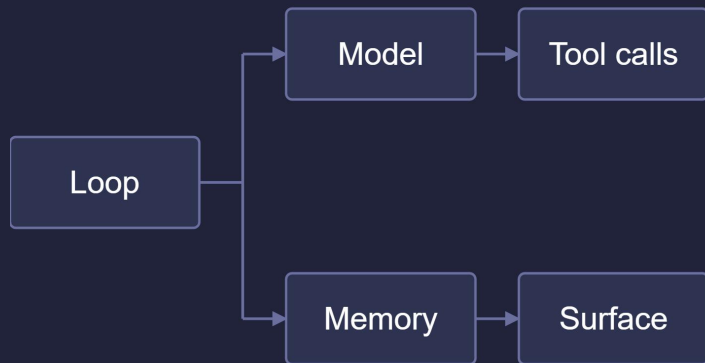
Harness

Loop, Tool-call,
Context Management

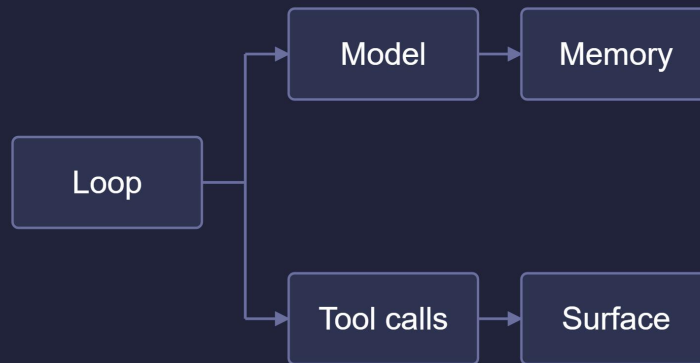
All Agents have similar components

Two agents: customer support, coding

Support agent



Coding agent



≈

Where the agentic stack is being built.

Where the agentic stack is being built.

Two ways to make things.

Craftsmanship

Directly hand-made

Output: Unique crafts

Each output is one of a kind

Unit cost cannot be lowered

Morden Factory Production

Produced with mold

Output: standardized product

Each output is quality-checked with tolerance standard

As output volume increases, unit cost deceases

Vibe coding is software craftsmanship

Craftsmanship

From ideas to crafts

If you can design, you can make it

Every output is unique

Limit on production volume

VIBE CODING

→ From ideas to running apps

→ If you can describe it, you can make it

→ Unique code is generated each time

→ Cannot massively produce agents

Factory Production, what software industry is lack of



What is the next stage?

Where the agentic stack is being built.

Software Industry Never Had Factories

Other disciplines had long established standardised production method

Machinery

Standard parts
Clear tolerance standard

Electronics

Data sheets,
Reference design

Construction

Building codes

Software

Only package manager—
Installable, but what about
“well used”?



Same Input, Different Results

Gap 1 — Stability

Same prompt produces different apps

If you run a prompt twice, it generates different codes, produces different agents

Zero commitment

No consistency and stability, each new version is unpredictable

Cannot know which version is the best

No way to know the version difference

Reusability is the minimum benchmark of a system

Where the agentic stack is being built.

Quality-check becomes optional

Gap 2 — Quality

Quality depends on who conduct the task

Same tool, different producer, different quality.

Testing relies entirely on self-discipline.

Test or not, what to test, not mandatory

No quality benchmark

Not like factory to impose quality benchmark on each product

The key is not if there is a test. The key is testing is not mandatory

Where the agentic stack is being built.

Each agent is independent

Gap 3 — High maintenance

**Experience of fixing A
cannot transfer to B**

experience intransferrable

No knowledge deposition

Every setp starts from zero

**One change leads to
regenerating the whole**

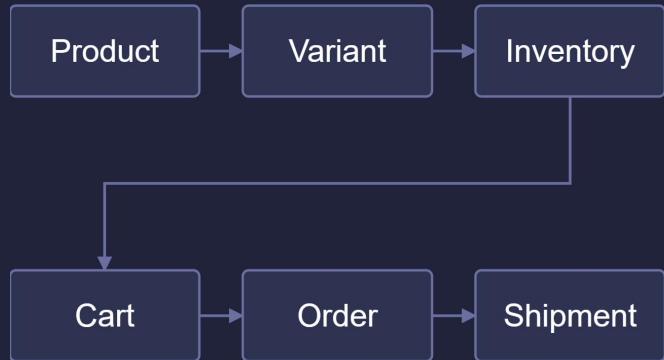
when regenerating, it brings unstability

10 agents mean 10 maintenance jobs

Where the agentic stack is being built.

In the past, software production has very few reusable components

Retail



≠

Travel



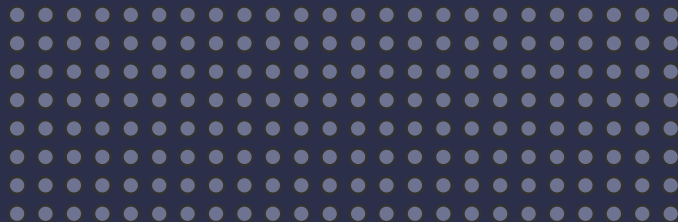
Once the scenarios and data models change, everything is different.

But, agent is the first software that can be produced by standard assembly line.

Model is of a new role

Coder: Build everything from zero

Architecture, Directory, Database Choice, Error Processing, Session State Storage



Tens of thousands decisions to make



Assembler: follow assembling list

Arrange components, look in to imports errors



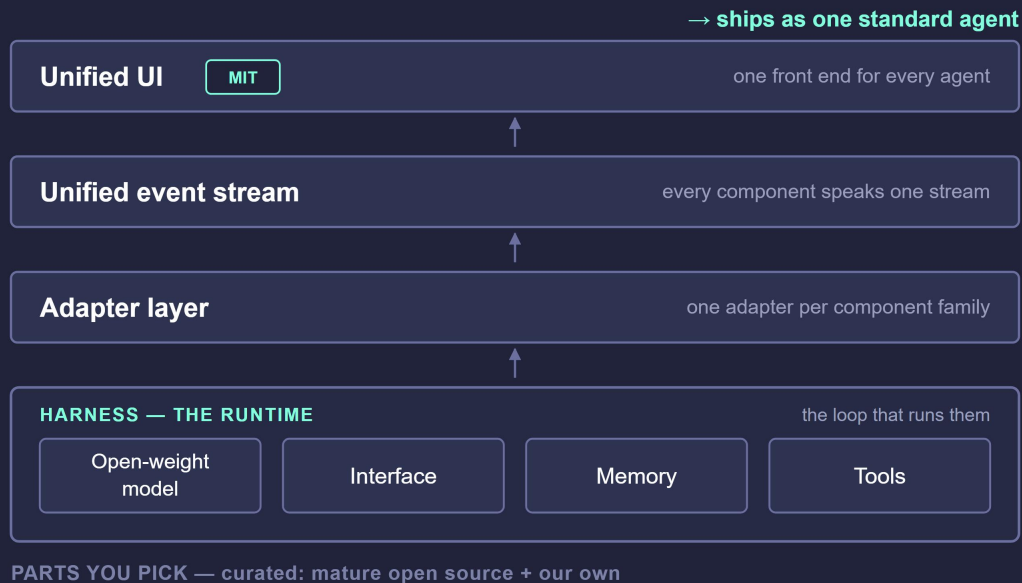
Decision volume decreases to dozens

Requirements on models to make their own decisions decreases dramatically

Each decision is a possible divergence point

Our New Stack

With our new stack, agents can be stably assembled with standard components

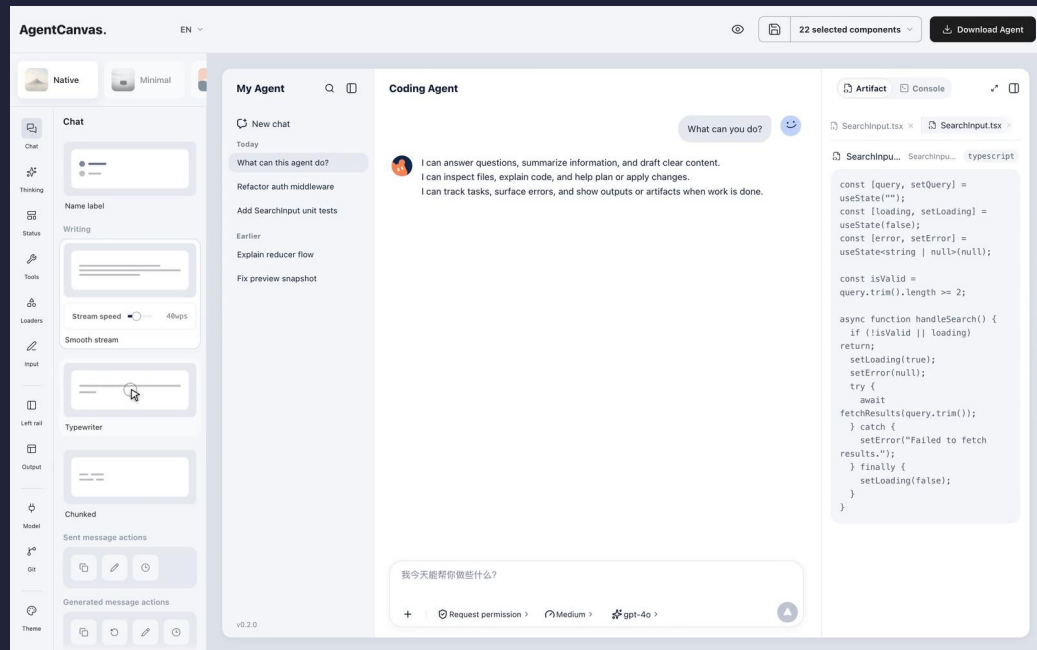


No need for users to concern about RTE

Where the agentic stack is being built.

AgentCanvas where you can assemble your agents

AgentCanvas—the interface layer has been opensourced (MIT protocol)



github.com/raytone-lab/agentcanvas

MIT · TypeScript

Components Selection

Interface, Thinking process, Tools, Approval, theme

UI Generation

A real scaffold that you can deploy

Each component is an interchangeable part

Where the agentic stack is being built.

How much token spent on coding task for an agent

4.17M

Token consumption for an intelligent agent coding task

Average consumption of dialogue coding

3,390

30×

Fluctuations in the cost of the same task

And the most expensive one did not turn out better.

It is evident that "instability" and "skyrocketing bills" are essentially the same issue.

Bai et al., "How Do AI Agents Spend Your Money?" — Stanford Digital Economy Lab + Microsoft Research, arXiv:2604.22750

Where the agentic stack is being built.

One Action, Double Payoffs

Take decision making away from models,
and fix them into the component list



Reproducible

Same lists, produce same system

Output fluctuation eliminated

Saving tokens

Things that already exist do not need to be generated again

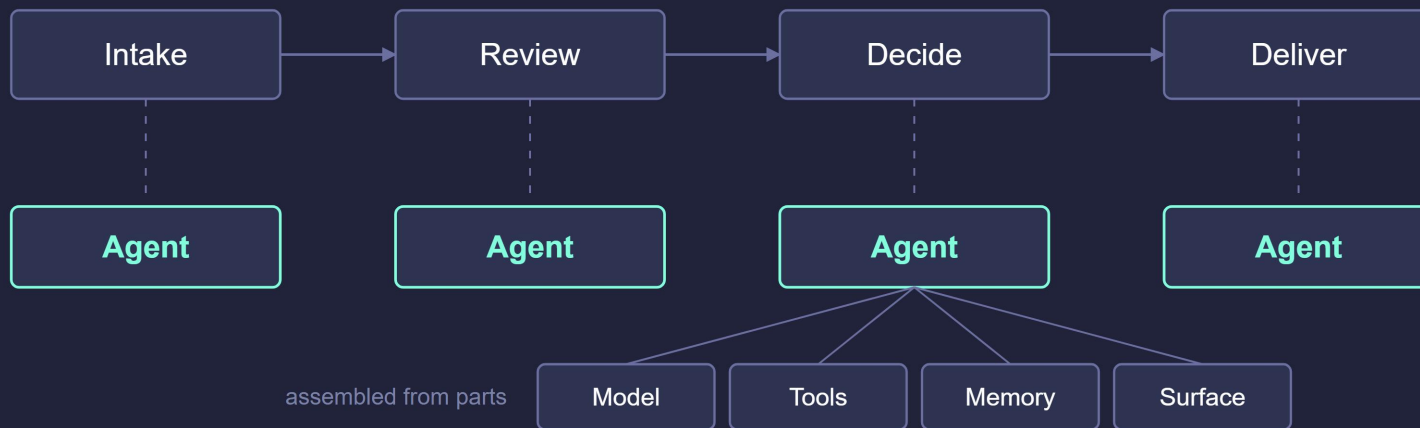
In the past, most of the expenses were spent rewriting existing code in open source.

Double payoffs by simply not to let model decide and choose.

Assembly Line Is Not Only for Making Agents

Today, the general practice of software industry is still heavily relying on customization. We believe that it eventually will become assembly lines.

AN INDUSTRY WORKFLOW



In the future, each node will be an agent rather than a function, and each agent will be assembled instead of being vibe coded.



Agentic AI Foundation

AGNTCon + MCPCon
Japan

Stop handcrafting. Start assembling.

Can it build me the same thing twice?

KAN WANG @ Raytone