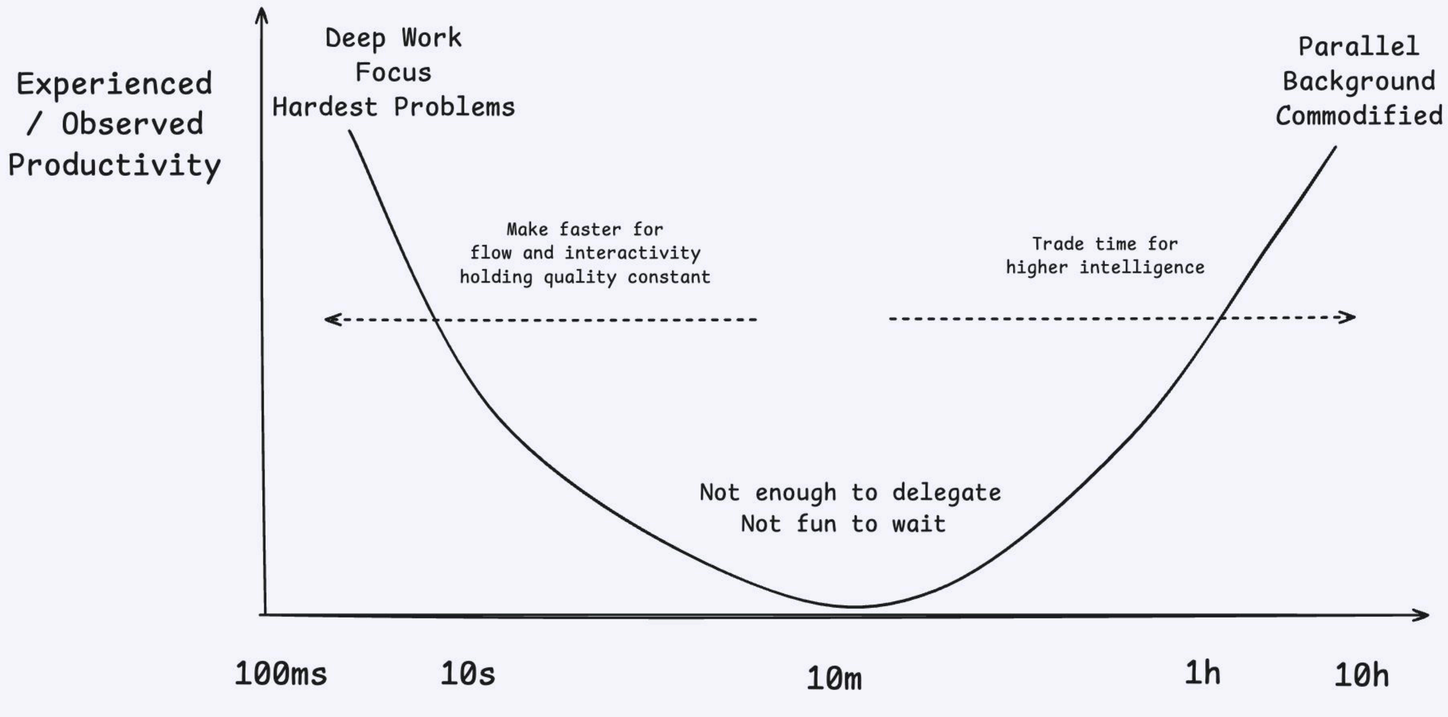
Autonomy is All You Need

Michele Catasta

President & Head of AI

replit

The Semi-Async Valley of Death



What about non-technical users?


swyx
@swyx

Cognition
@cognition

Supervised vs Autonomous Coding Agents

Supervised Agents

Autonomous Agents

Agent 3

Tesla FSD / Supervised Coding Agents
Meant to be used with a *technically sophisticated* human-in-the-loop

Requires technical input from the user
(Driver's License still needed)

Waymo / Autonomous Coding Agents
Empower *every knowledge worker* in the world to create software

No technical input from the user
(No Driver's License required)

From Copilots to Autonomous Agents

<1 minute feedback loop, Constant supervision

- **Completions**
- **Assistant**
- **v1 – ReAct**
- **v2 – Tool Calling**
- **v3 – Autonomous Agents**

Hours of independent work

Code completion / assistant require deep technical expertise

v1/v2 Agents were **not sufficiently autonomous** for wide adoption by knowledge workers ("valley of death")

Can we build fully autonomous Agents?

Scoped Autonomy

Autonomy is **conflated with long run-times, or a loss of control**

In reality, the autonomy given to Agents can be given a **very specific scope**

For instance, Replit Agent 3 makes technical decisions autonomously:

- could lead to long gaps between user interactions, if the **scope is broad**
  → because Agent can complete its work without constant check-ins

But an Agent can be **both autonomous and fast on a narrow scope**!

The **user maintains control** over aspects of the project that matter to them

Autonomy is not a vanity metric

Tasks have natural complexity (a minimum, ***irreducible*** amount of work)

Agents constantly **plan → implement → test → loop,** which requires performing work over coherent long trajectories

- Goal: **Maximize the irreducible runtime of Agent**

Especially at Replit:

- Users are nontechnical, so Agent must be a trusted technical collaborator
- Abstract away as much of the pain of software creation as possible
- Users control the project but **trust Agent with any technical decision**

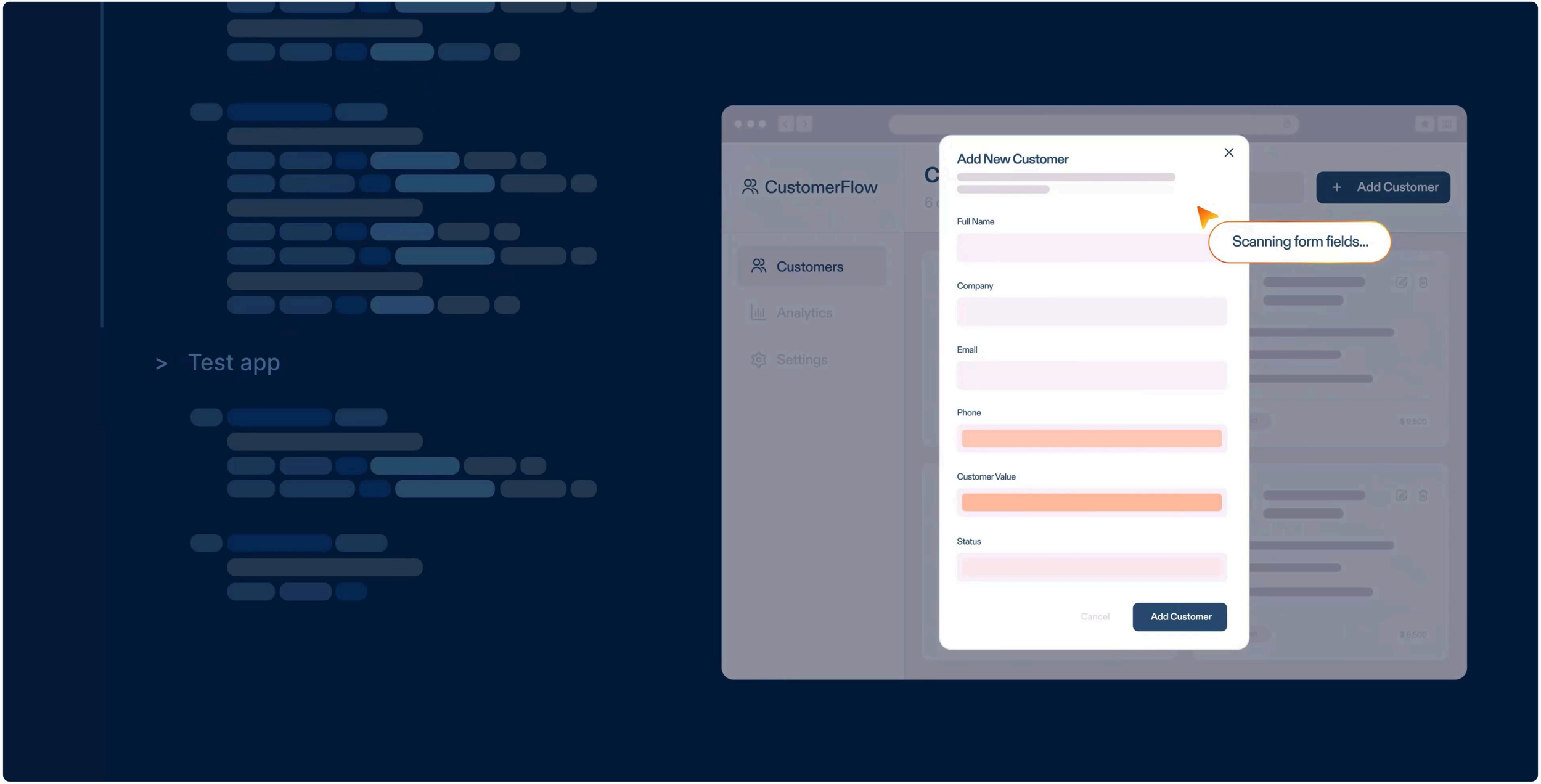
Pillars of Autonomy

Multi-hour autonomous work

Frontier capabilities:
baseline IQ

Verification:
local correctness,
9s of reliability (avoid compounding errors)

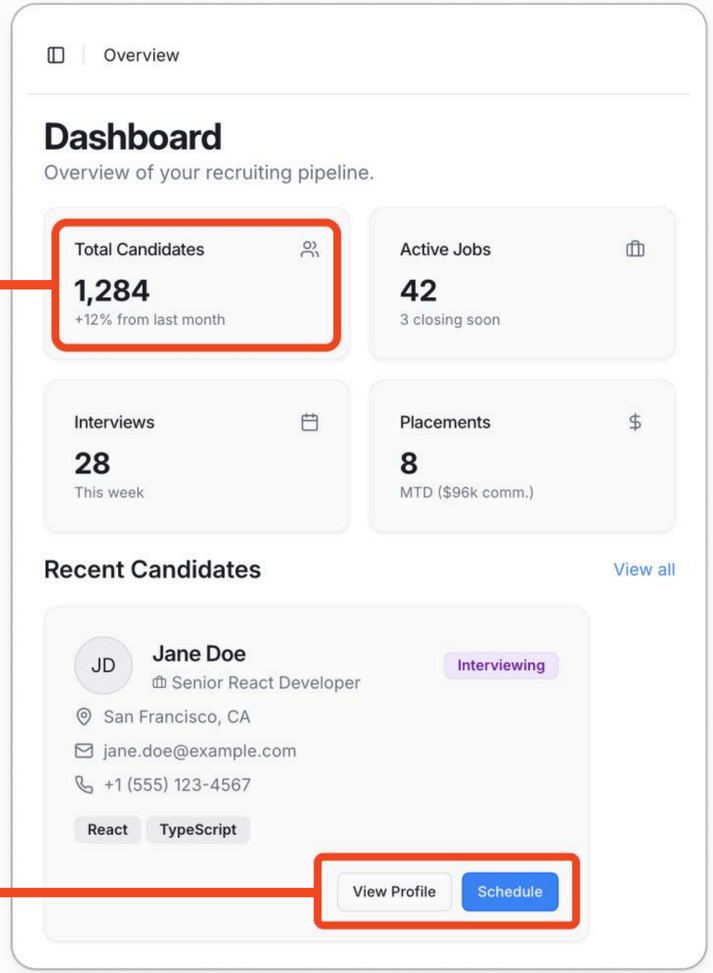
Context Management:
global coherence,
goal+task management


> Test app
CustomerFlow
Customers
Analytics
Settings
+ Add Customer
Add New Customer
Full Name
Company
Email
Phone
Customer Value
Status
Cancel
Add Customer
Scanning form fields...

Testing fixes broken and hallucinated features

Mock Data with no DB connection

Button handlers not hooked up



Without testing, Agents build “painted doors”

More than 30% of individual features are broken

Almost every app has at least one broken feature or painted door

Feedback without Human Supervision

Agents must gather the feedback they need from their environment

- Non-technical users cannot provide the needed technical feedback
- They can only perform QA testing, which is often tedious *(bad UX)*

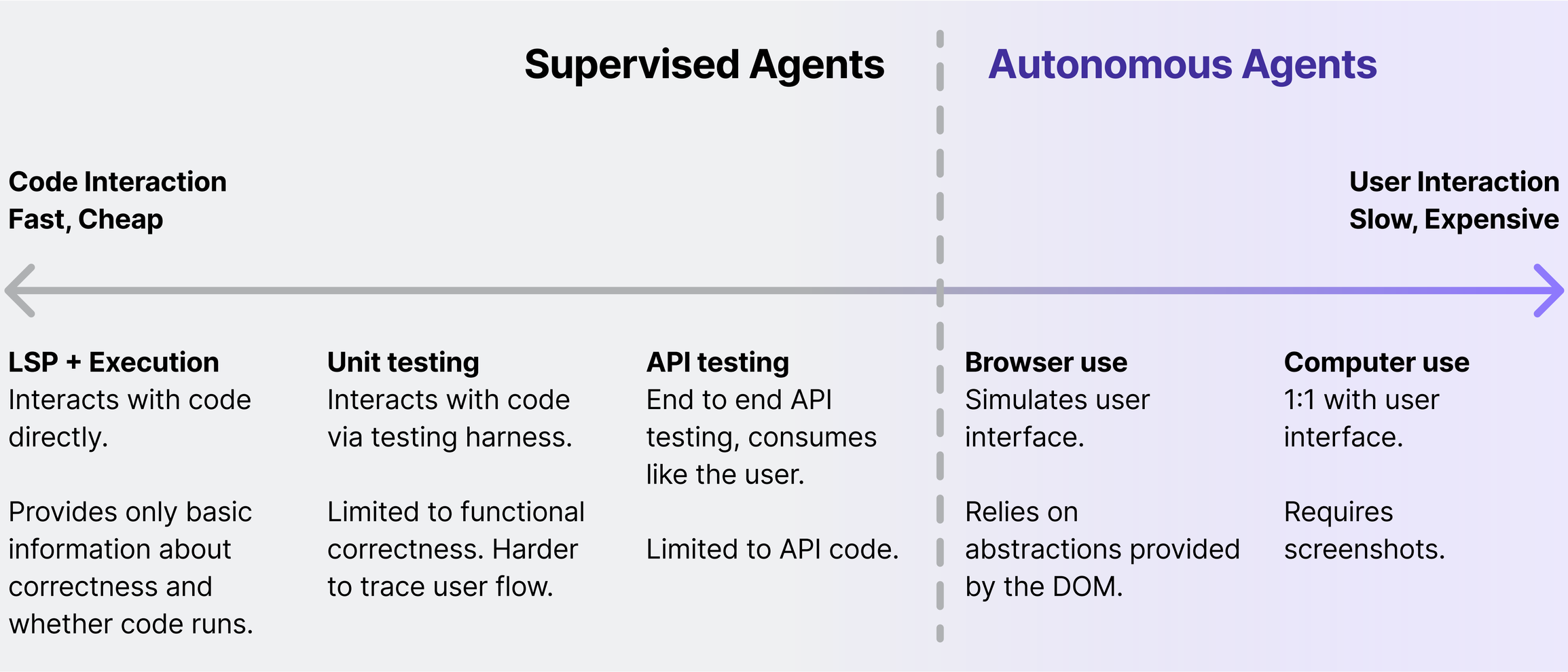
Why autonomous testing?

→ Break the feedback bottleneck *(no more waiting on human response)*

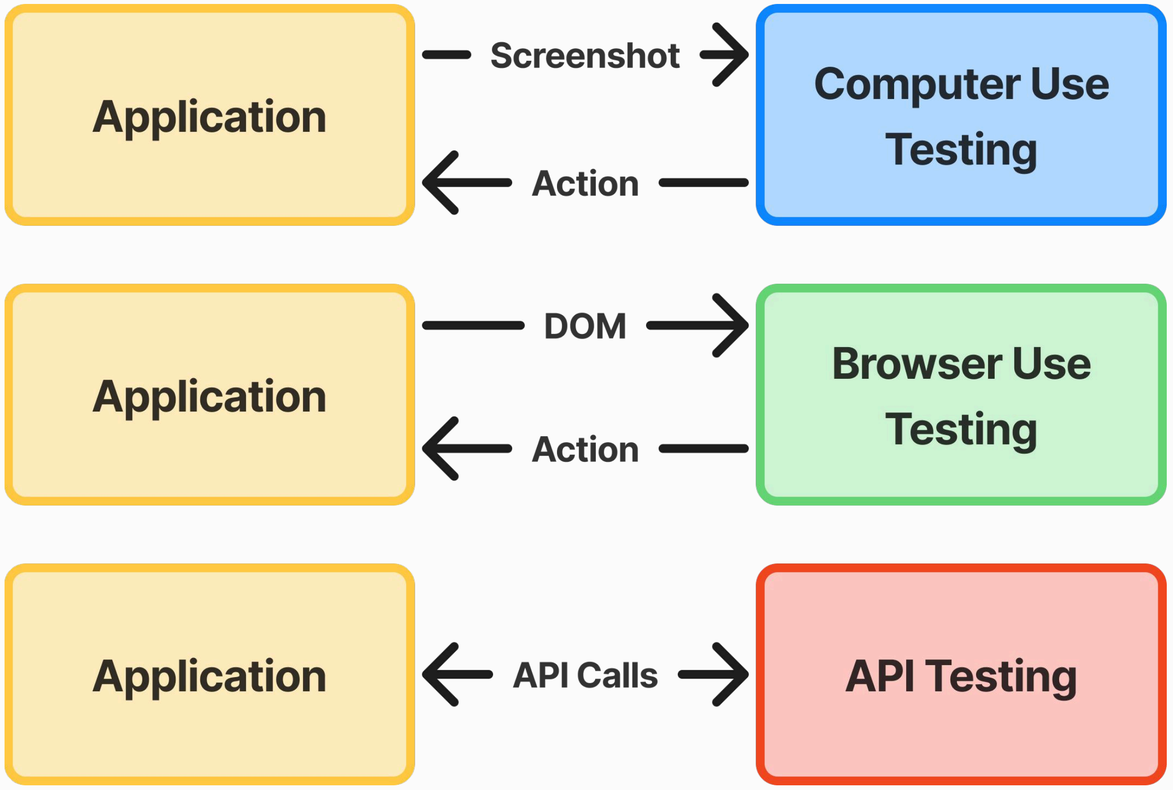
→ Prevent the accumulation of small errors

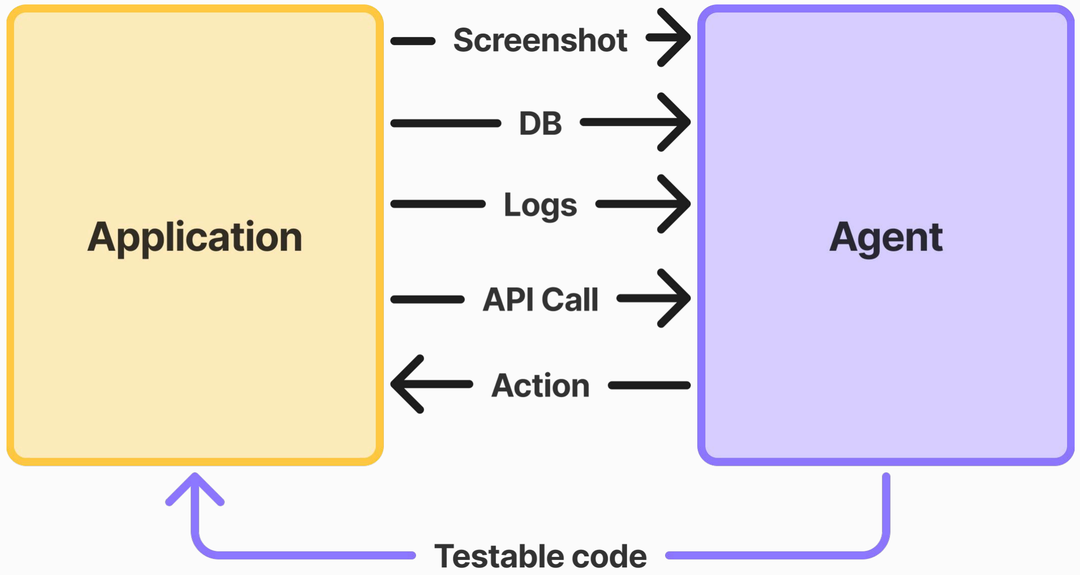
→ Overcome the laziness of frontier models *(verify task completions)*

Spectrum of Code Verification



Autonomous App Testing

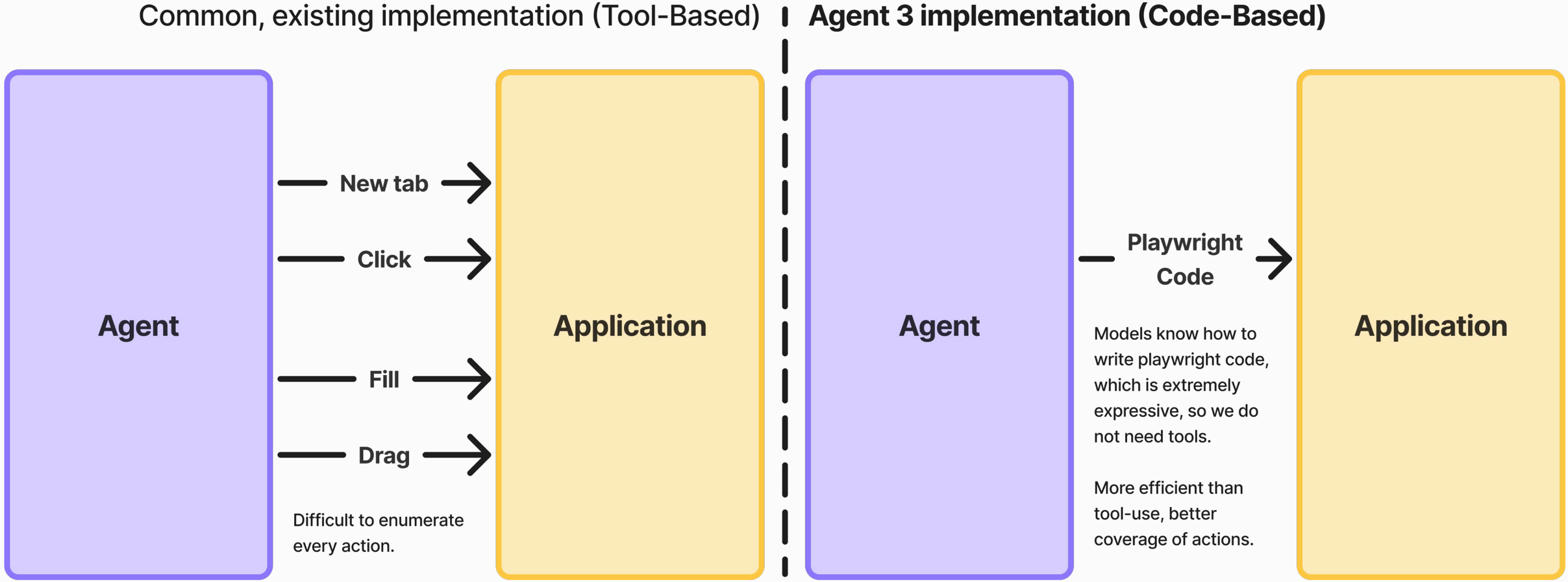




Replit Agent 3

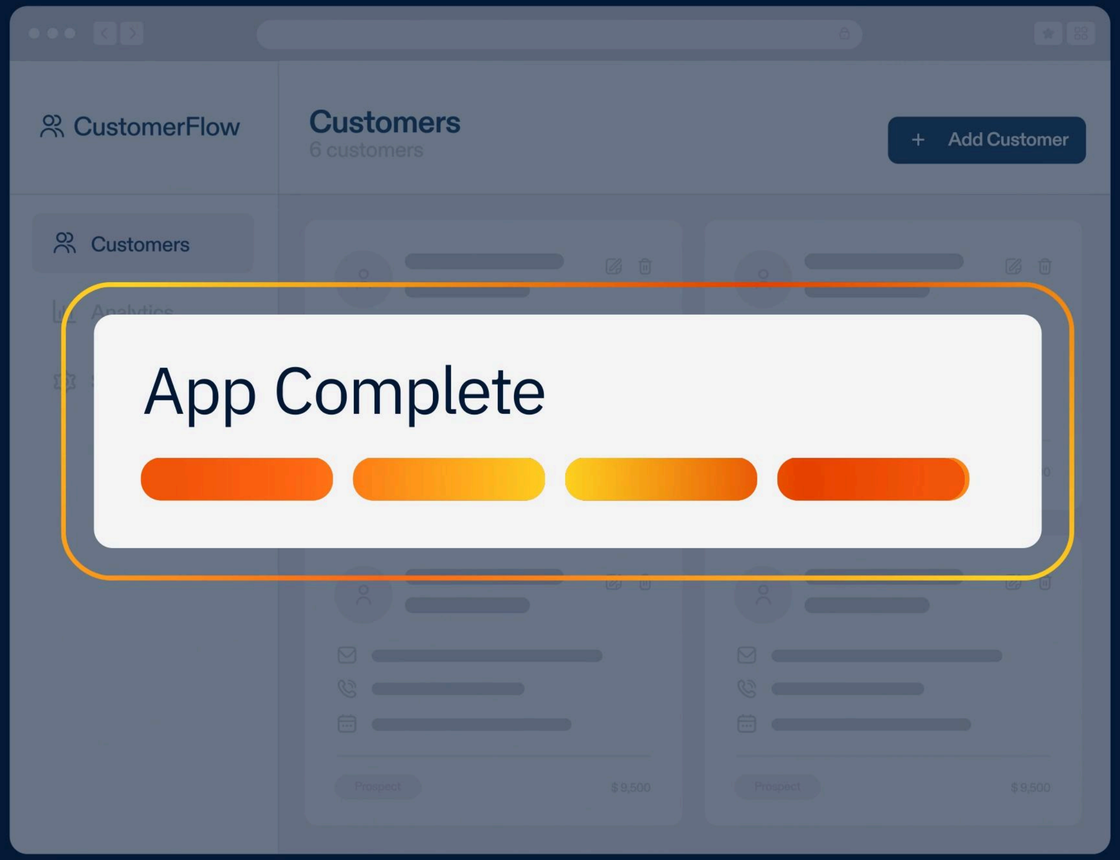
- Generates applications amenable to testing
- Testing integrates information from multiple sources to generate more useful feedback
- Cost effective w/ Computer Use only as a fallback

Tools vs Code for Browser Use



Time worked: 282 minutes

> Work done: 1436 actions



Context Management and Coherent Trajectories

Long-context models are **not needed** for coherent long trajectories

- Most tasks can be accomplished within 200k tokens
- Codebase itself maintains state *(including plan and tasks)*
- Agent writes information to a file before clearing memories

ANTHROP\C

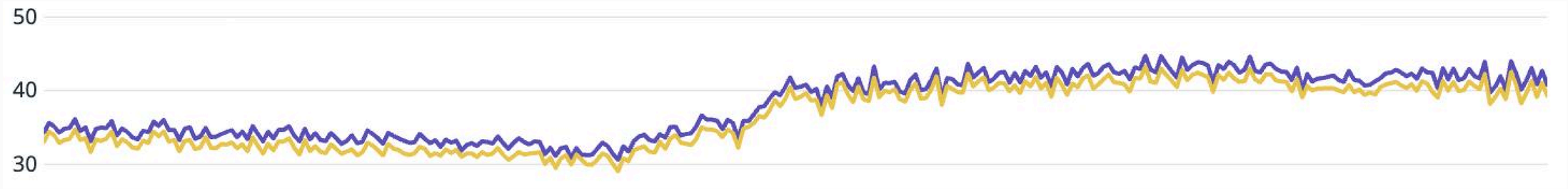
> Claude Sonnet 4.5 is state-of-the-art on the SWE-bench Verified evaluation, which measures real-world software coding abilities. Practically speaking, we've observed it maintaining focus for more than 30 hours on complex, multi-step tasks.

Context Management with Subagents

→ Subagents are invoked by the core loop with a task and fresh context

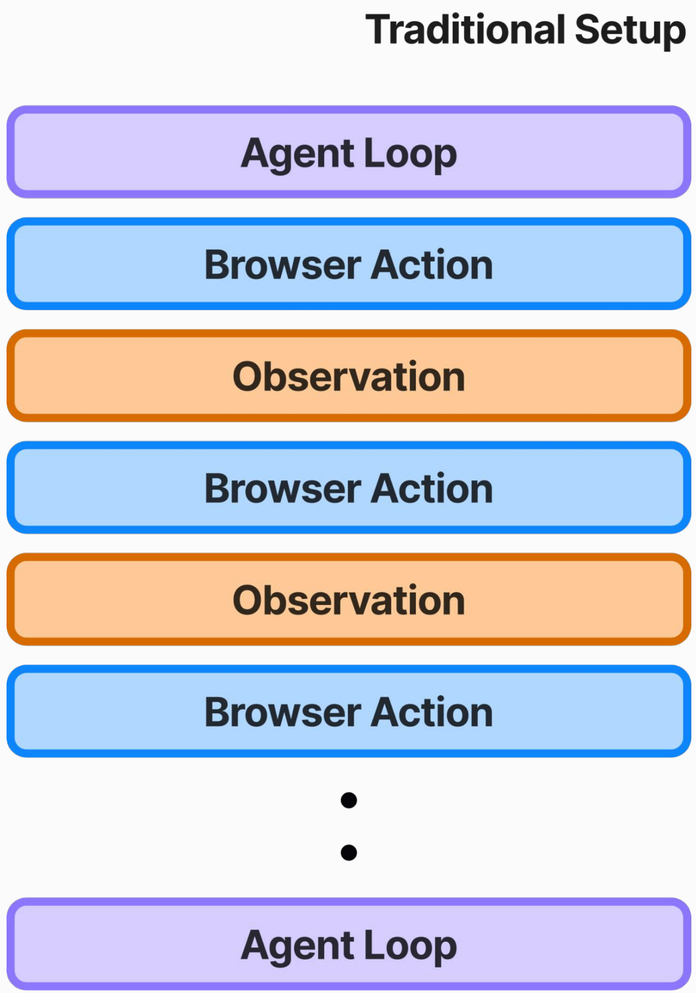
→ Separation of concerns **protects the core Agent's context window**

→ Significantly improves the number of memories per compression



Subagents in Replit Agent - before & after

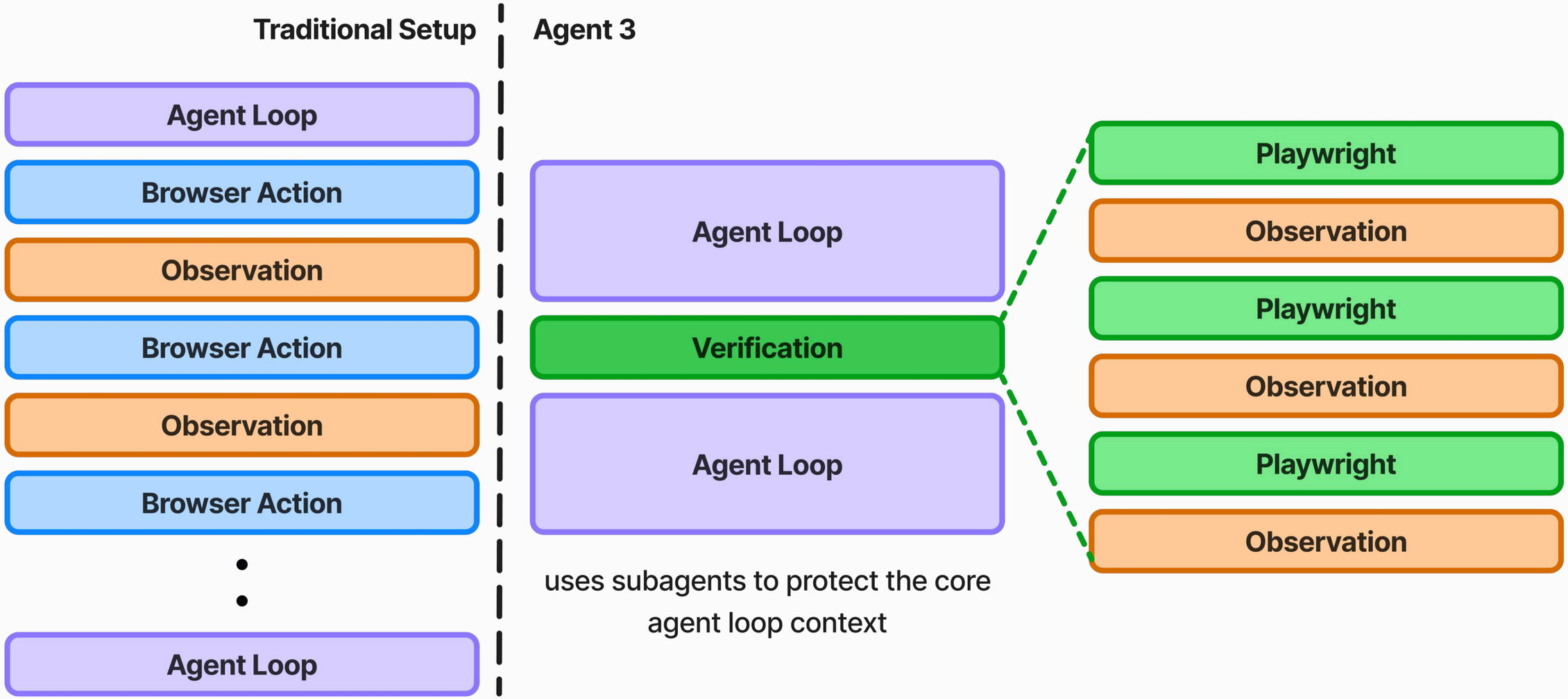
Testing as a Subagent



Traditional Browser interaction setup can lead to significant **context pollution**, causing:

- greater cost
- slower speed
- worse performance

Testing as a subagent




Parallelism

Trading compute for time with parallel agents

Parallel Agents spend extra compute in exchange for time:

- Cost to gather the **same context in multiple context windows**
- Cost to resolve **merge conflicts**

Enables features that would otherwise be too slow:

→ Continuous testing and code review

→ Async processes that inject information into the Agent's memory

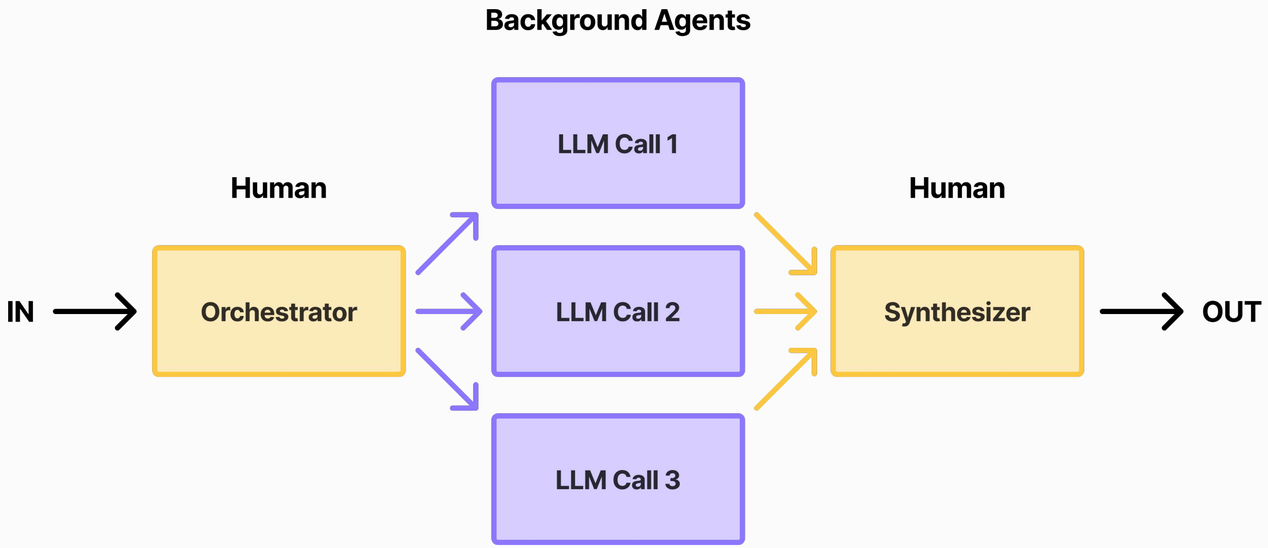
→ Sampling of multiple trajectories

Background Agents *(User as Orchestrator)*

Tasks are determined by the user, and each task is dispatched in its own thread.

Dispatch: An expert user determines tasks. Task decomposition is sub-optimal.

Merge: An expert user resolves merge conflicts. Automatic merge-conflict resolution is not solved in general.



Subagents *(Core Loop as Orchestrator)*

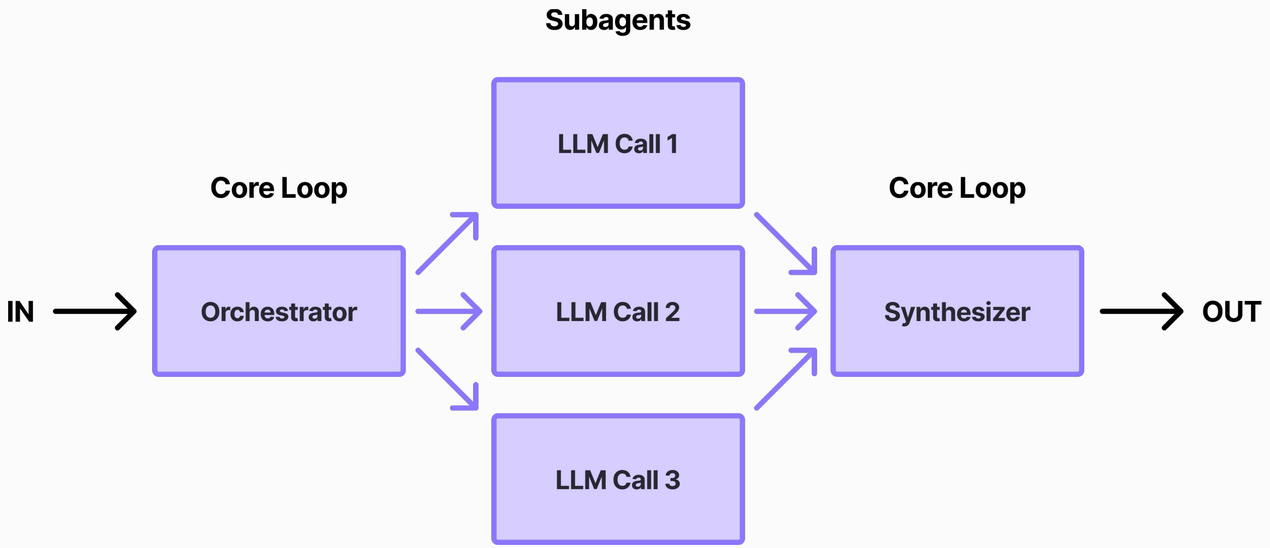
Tasks are determined by the core Agent loop.

Parallelism is decided on the fly, and mediated by the Agent's understanding of the problem and of its own capabilities.

Dispatch: The core loop spends tokens to reason about and dispatch tasks.

Merge: Merges are easier due to "merge conflict-aware" task decomposition.

Subagent can be specialized *(choice of model, prompt/persona)*



We are hiring!

https://replit.com/careers

Q&A

replit

Michele Catasta

President & Head of AI

Thanks to

Ryan Carelli

Zade Kaylani

Vaibhav Kumar

Peter Zhong