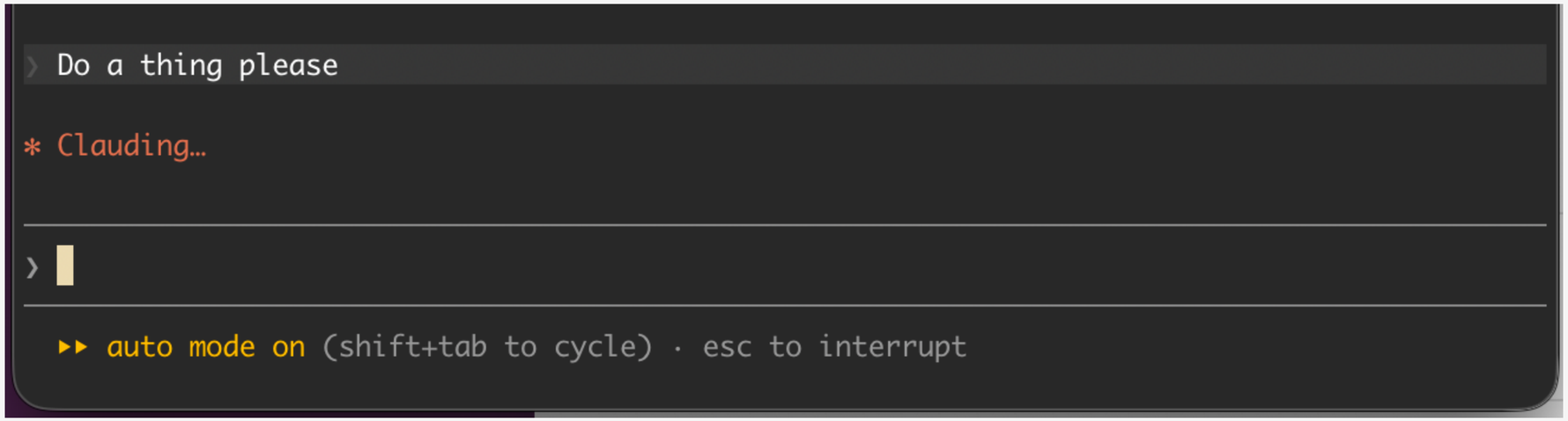
Planning next moves ...

Maude Lemaire / LDx3 London / 2026

How many of you are Cursor users?

Tell me about time to first token.

Planning next moves


› Do a thing please
* Clauding…
›
▸▸ auto mode on (shift+tab to cycle) · esc to interrupt

What *is* “Planning next moves”?

Time to first token (TTFT) is the delay from when an inference request is accepted until the model starts emitting the first output token.

Time to first token (TTFT) is the latency users feel most acutely.

January // Reliability

February 14th to May 15th

Claude Code Operational

90 days ago — 99.14 % uptime — Today

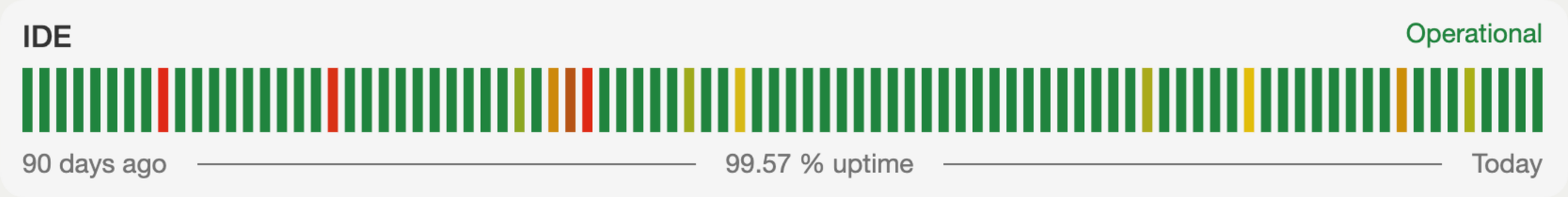
Copilot

90 days ago — 99.86 % uptime — Today

Normal

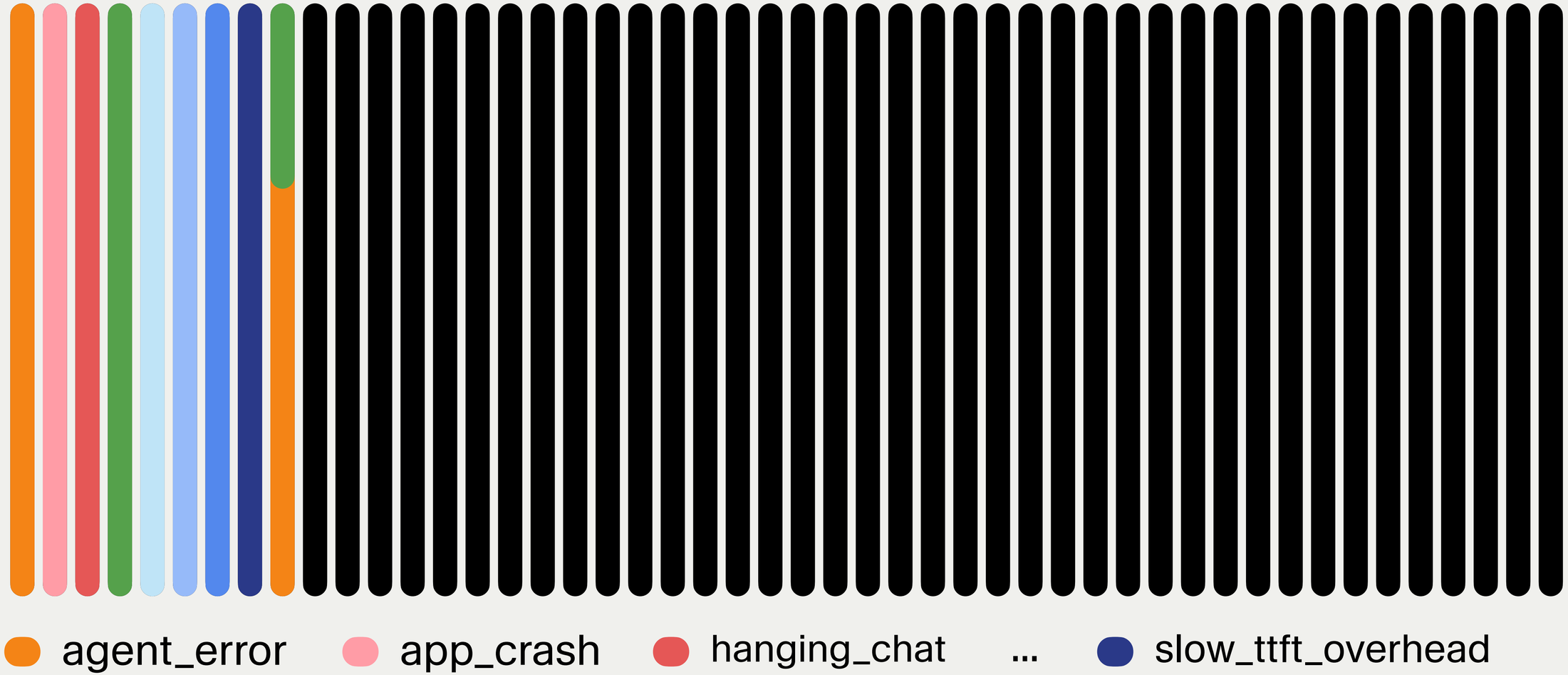
Codex 5 components 99.98% uptime

February 14th to May 15th

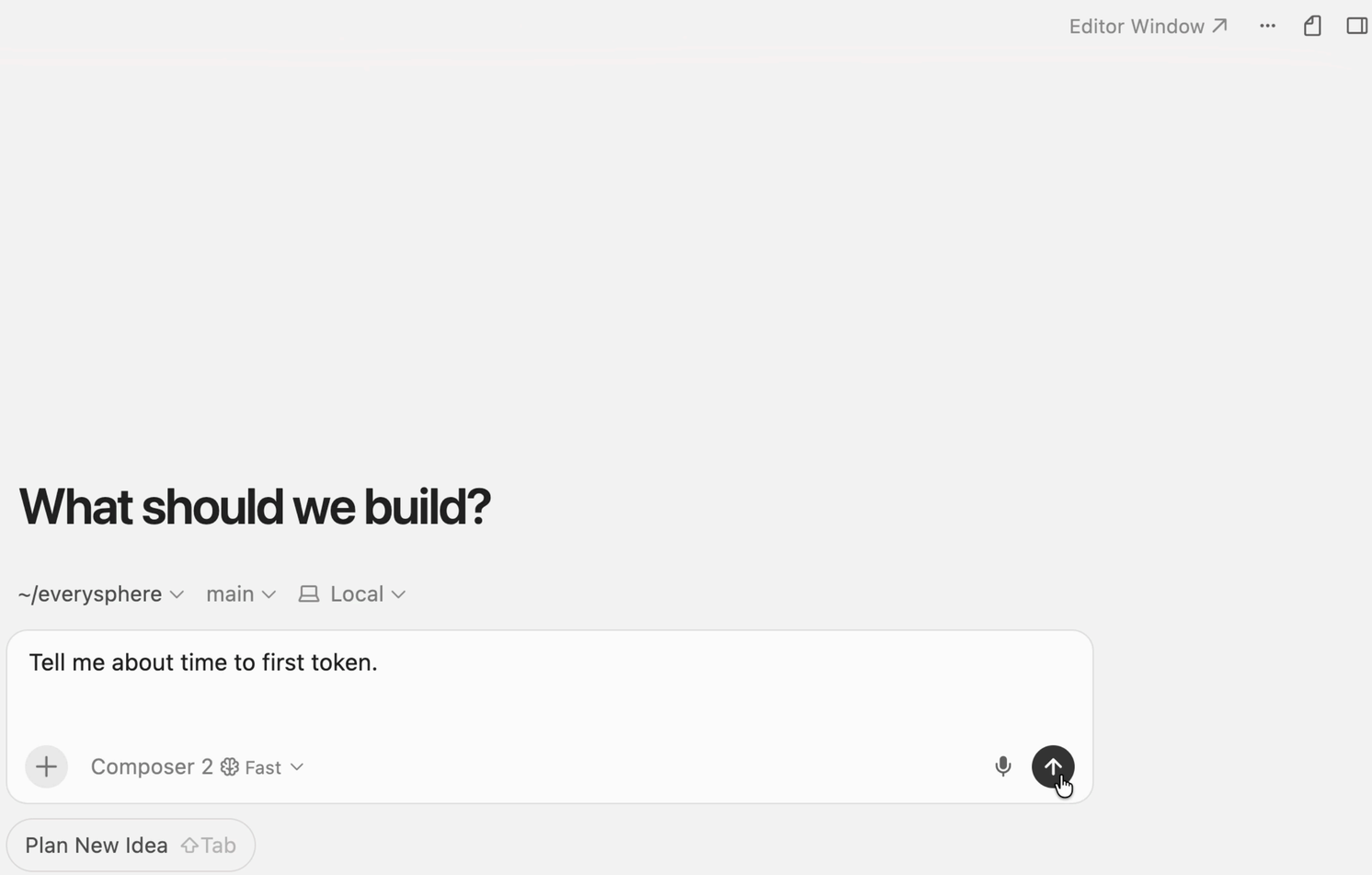


February // Degraded sessions

Degraded Sessions



Let’s do a little experiment

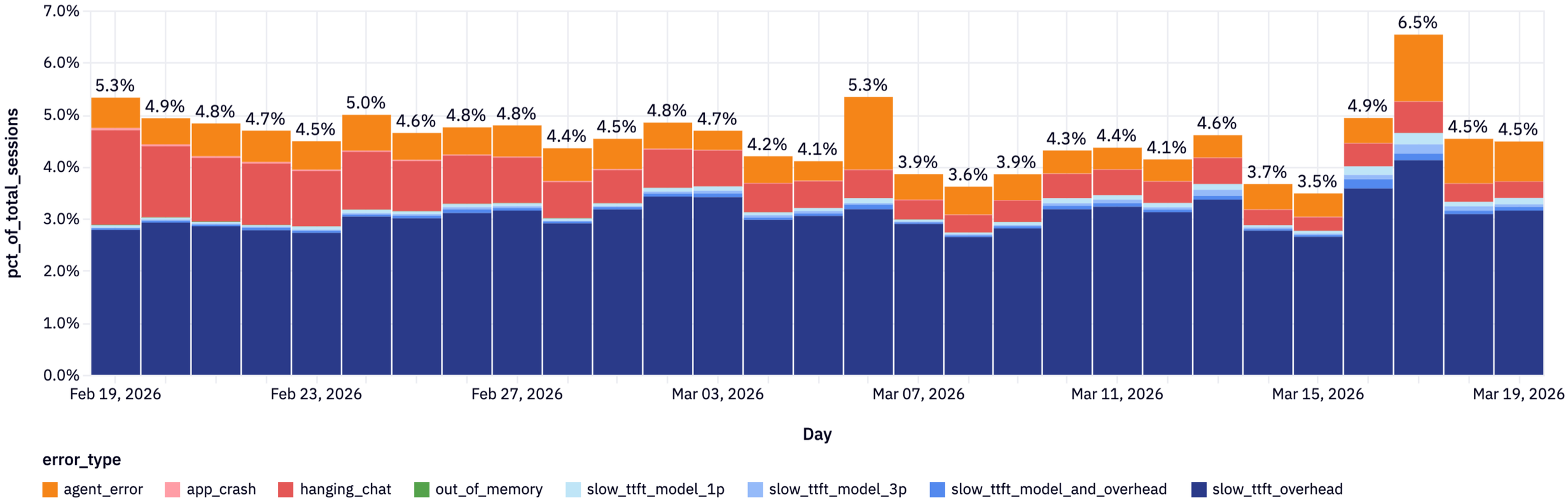

Editor Window
What should we build?
~/everysphere
main
Local
Tell me about time to first token.
Composer 2
Fast
Plan New Idea
Tab

“slow” is 40s

Degraded Sessions

% of Sessions Degraded by Error Type (Weighted)

dt before 2026-03-20



March & April // Only 1% degraded sessions by June

slow_ttft_overhead 🐰

“Tell me a punny joke about Cursor’s TTFT.”

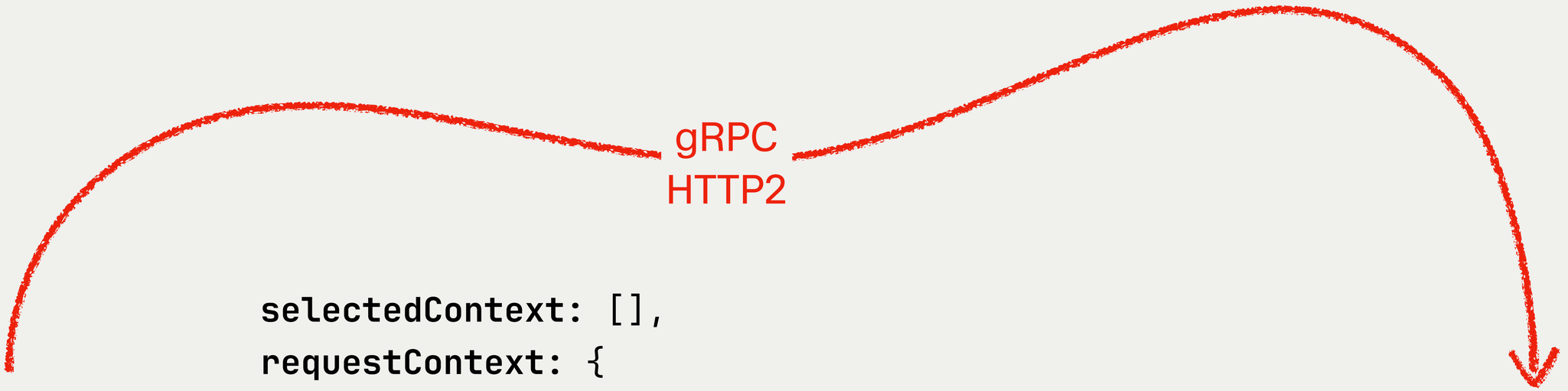
client

“Tell me a punny joke about Cursor’s TTFT.”

client

- **Selected context:** @mentions: files, folders, images, terminals, diffs, rules
- **Request context:** workspace and session, MCP tools, git repos, file contents
- **Conversation state** and **new action**: full prior conversation *and* your new message

Turn 1

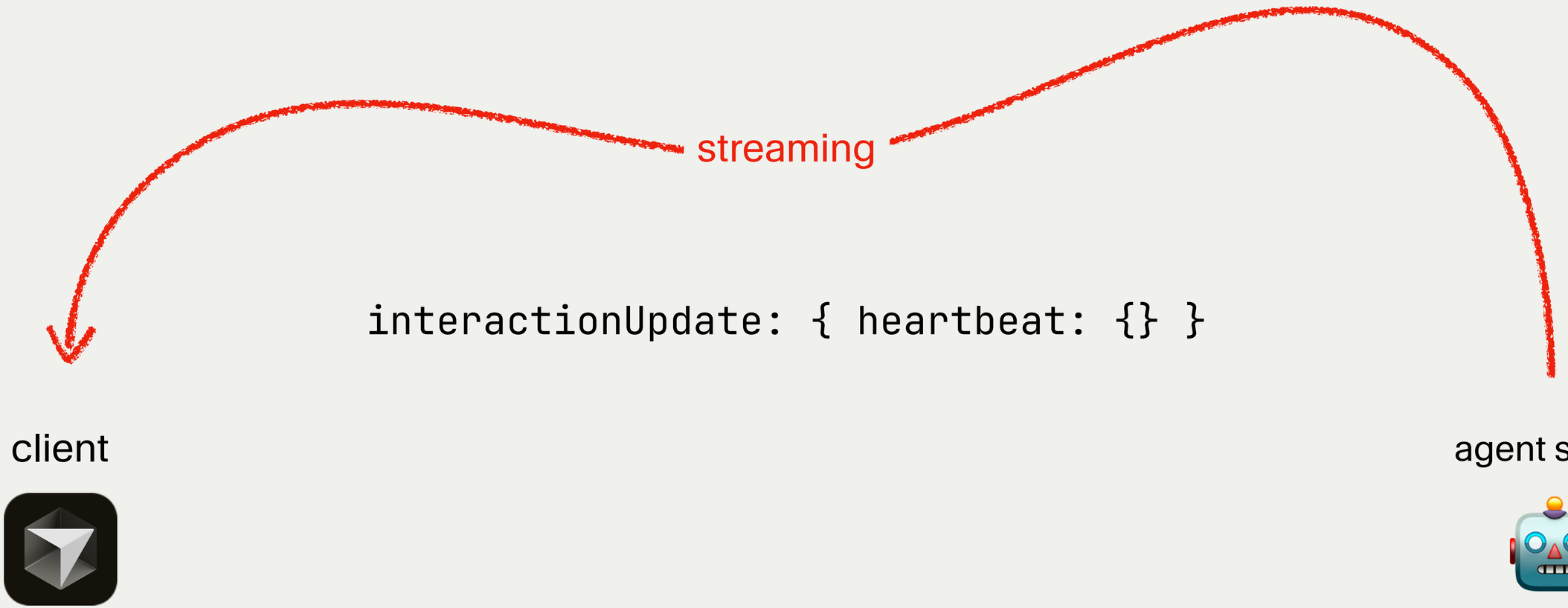


client

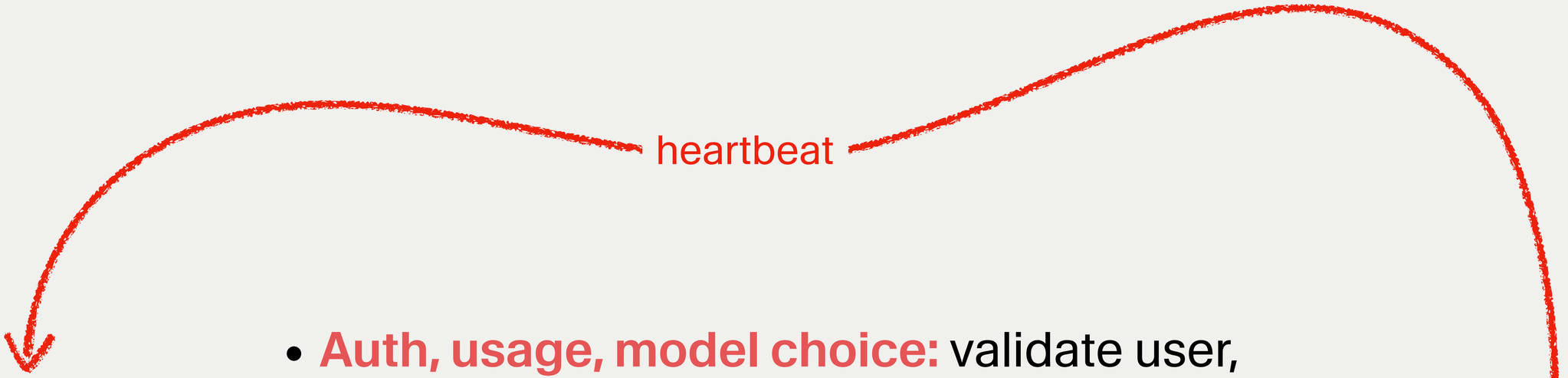
agent server

```
selectedContext: [],
requestContext: {
    env: { os: “darwin 25.5.0”, … },
    gitRepos: { root: “/Users/maude/repo”, … },
    agentSkills: [ { name: “unslop”, … } ],
}
conversationState: [],
action: { text: “Tell me a punny joke about Cursor’s TTFT” }
modelDetails: { modelName: “composer-2.5-fast” }
```

Turn 1


streaming
interactionUpdate: { heartbeat: {} }
client
agent server

Turn 1

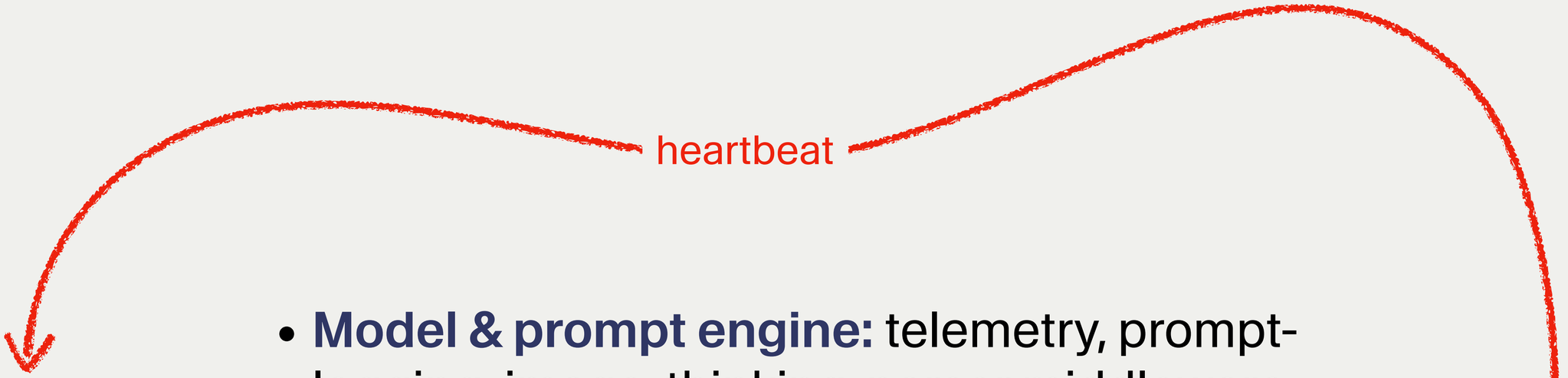


client

- **Auth, usage, model choice:** validate user, check quota, pick the model.
- **Split stream:**
  - remote execution
  - remote kv
  - ...

agent server

Turn 1

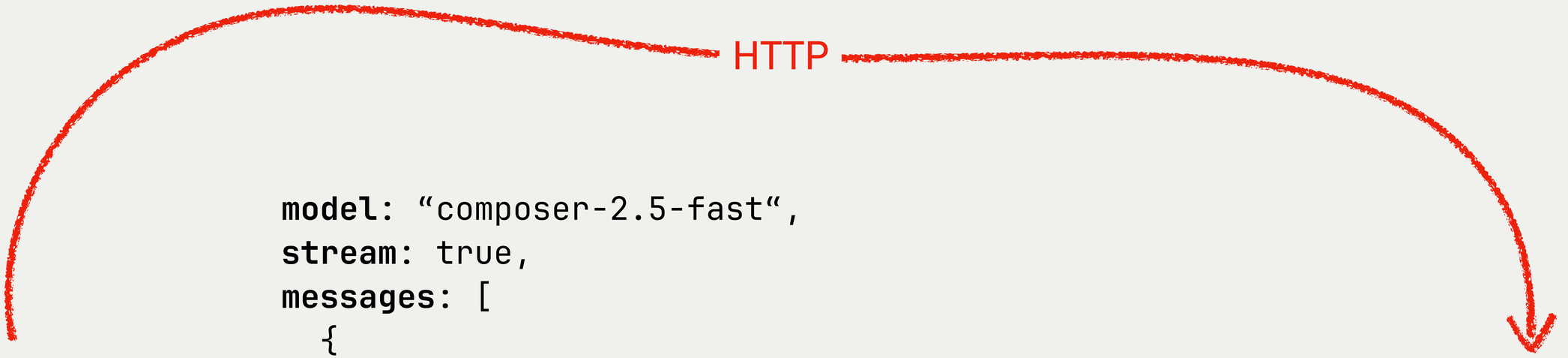


client

agent server

- **Model & prompt engine:** telemetry, prompt-logging, image, thinking-parser middleware
- **Build prompt:** assemble the complete system prompt from the request context (rules, skills, env), and user message
- **Call out to inference proxy**

Turn 1



agent server

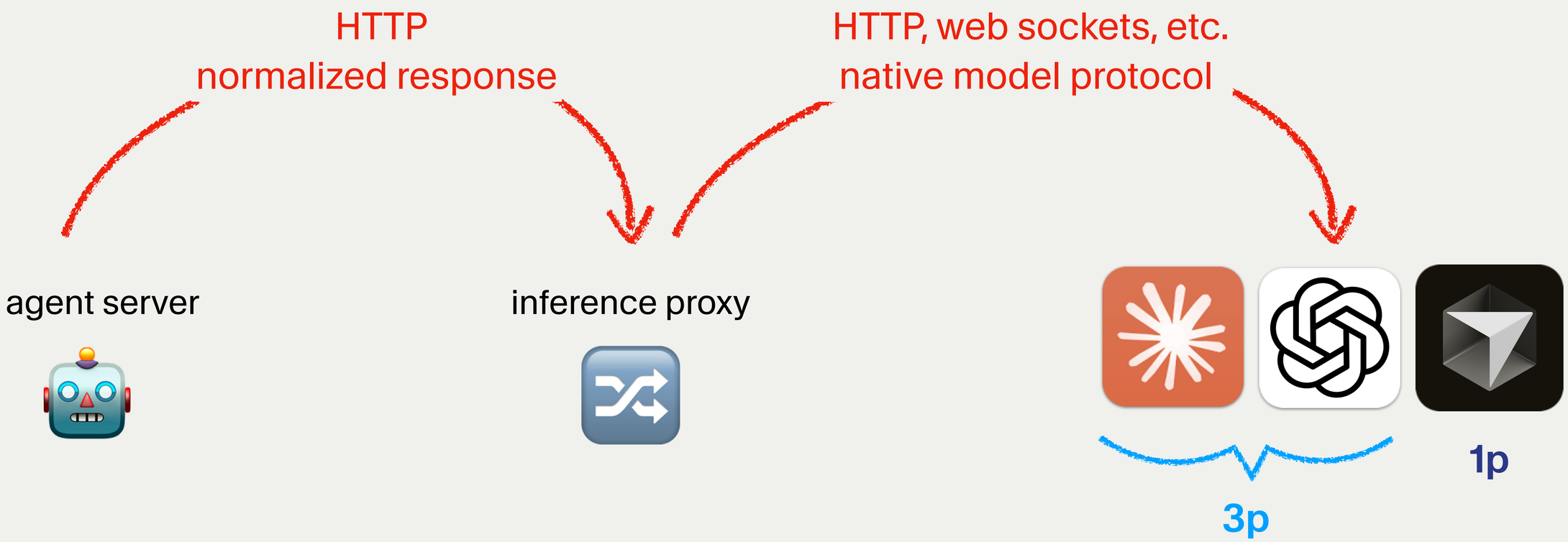
🤖

inference proxy

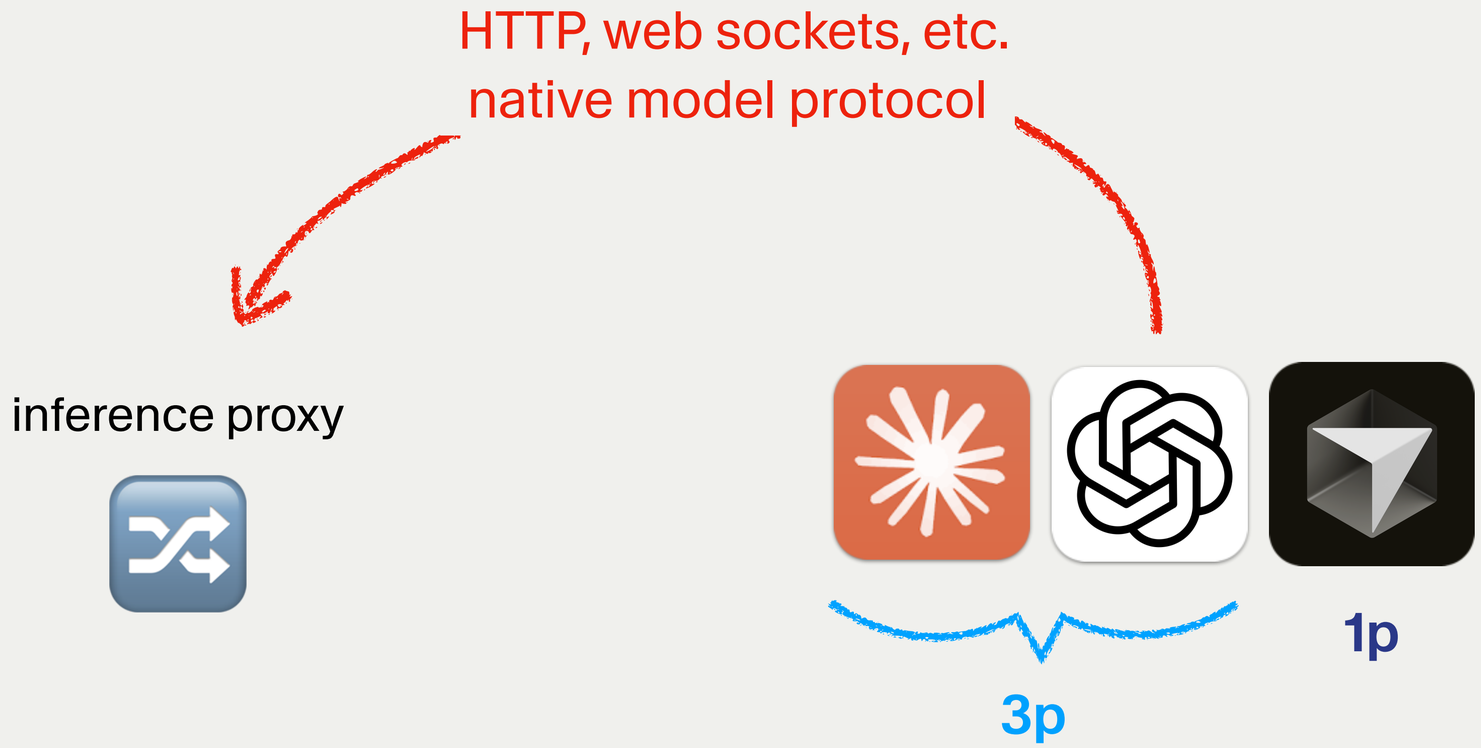
🔀

```
model: “composer-2.5-fast“,
stream: true,
messages: [
  {
    role: "system",
    content: "<the big Cursor agent system prompt:
    tools, rules, env, user info…>"
  },
  {
    role: "user",
    content: "tell me a punny joke about Cursor's TTFT”
  }
],
tools: [ /* read_file, shell, MCP tools, … */ ],
max_tokens: 8192,
```

Turn 1


HTTP
normalized response
HTTP, web sockets, etc.
native model protocol
agent server
inference proxy
3p
1p

Turn 1


HTTP, web sockets, etc.
native model protocol
inference proxy
1p
3p

Turn 1

streaming

```
data: {"type":"delta","delta":{"text":"Why "}}
data: {"type":"delta","delta":{"text":"did the "}}
data: {"type":"delta","delta":{"text":"Cursor agent "}}
data: {“type":"delta","delta":{"text":"refuse to race? "}}
…
data: {"type":"message_stop"}
```

agent server

inference proxy

Turn 1

streaming

first token! 🎉

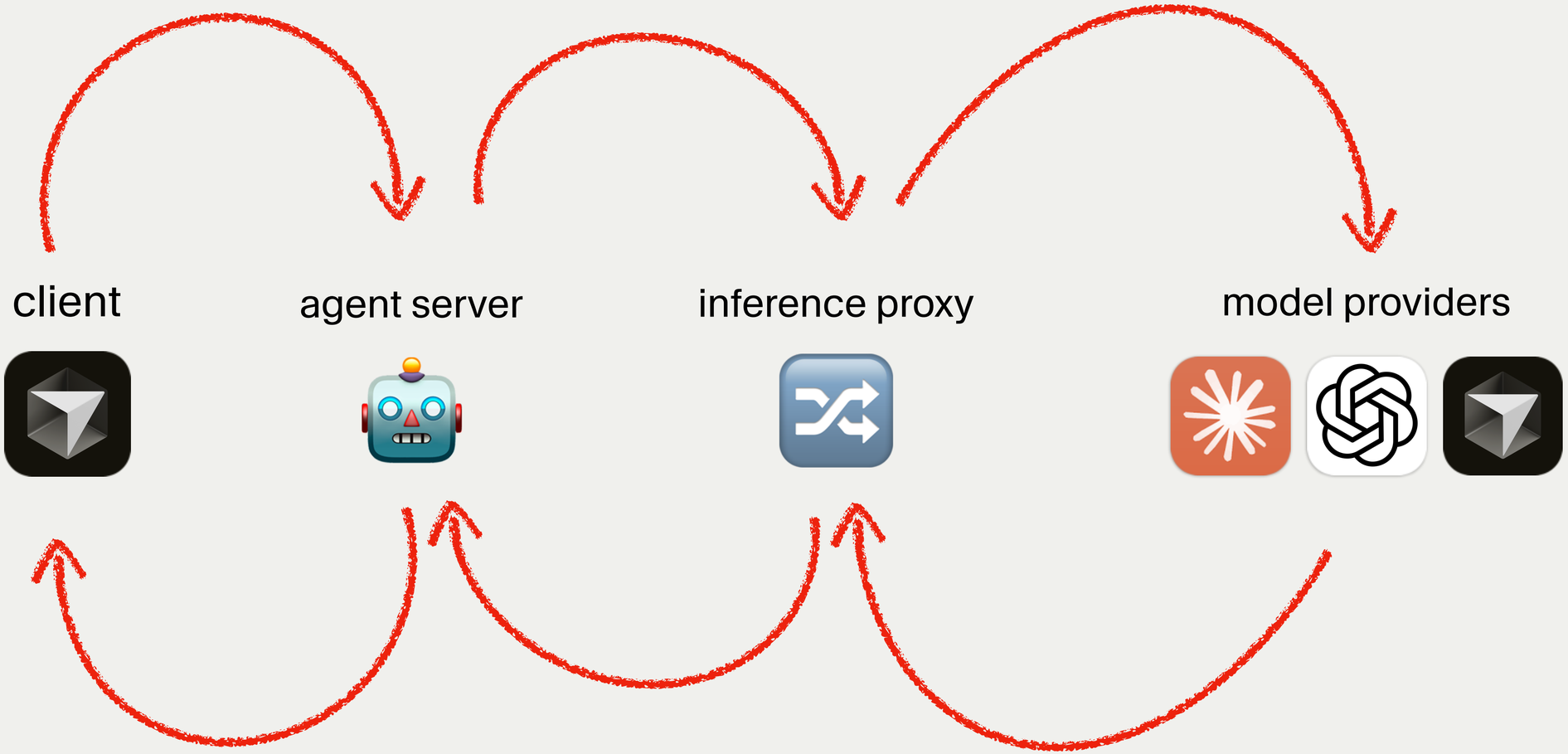
```
data: {"type":"delta","delta":{"text":"Why "}}
data: {"type":"delta","delta":{"text":"did the "}}
data: {"type":"delta","delta":{"text":"Cursor agent "}}
data: {“type":"delta","delta":{"text":"refuse to race? "}}
```

client

agent server

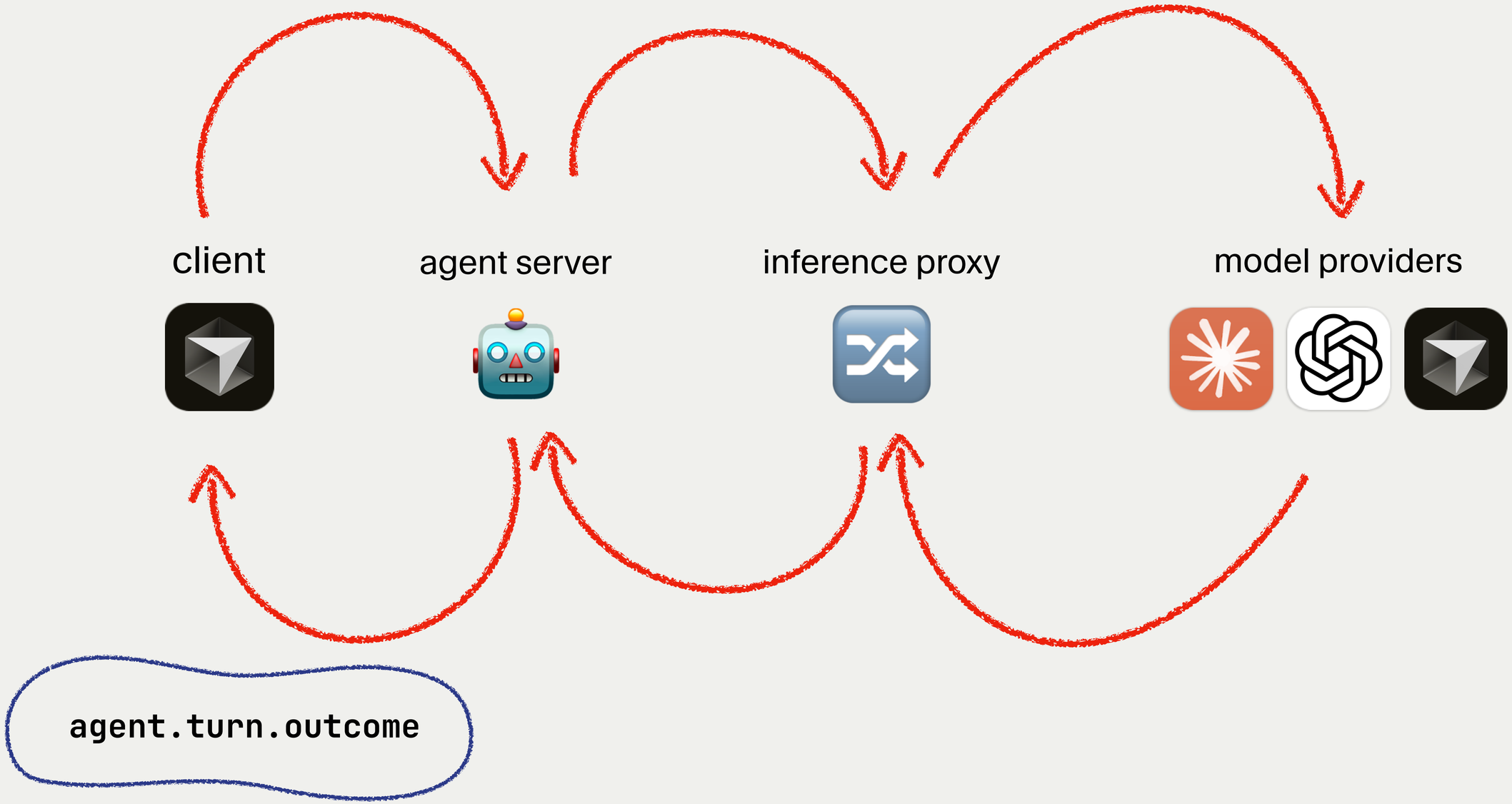
🤖

Turn 1

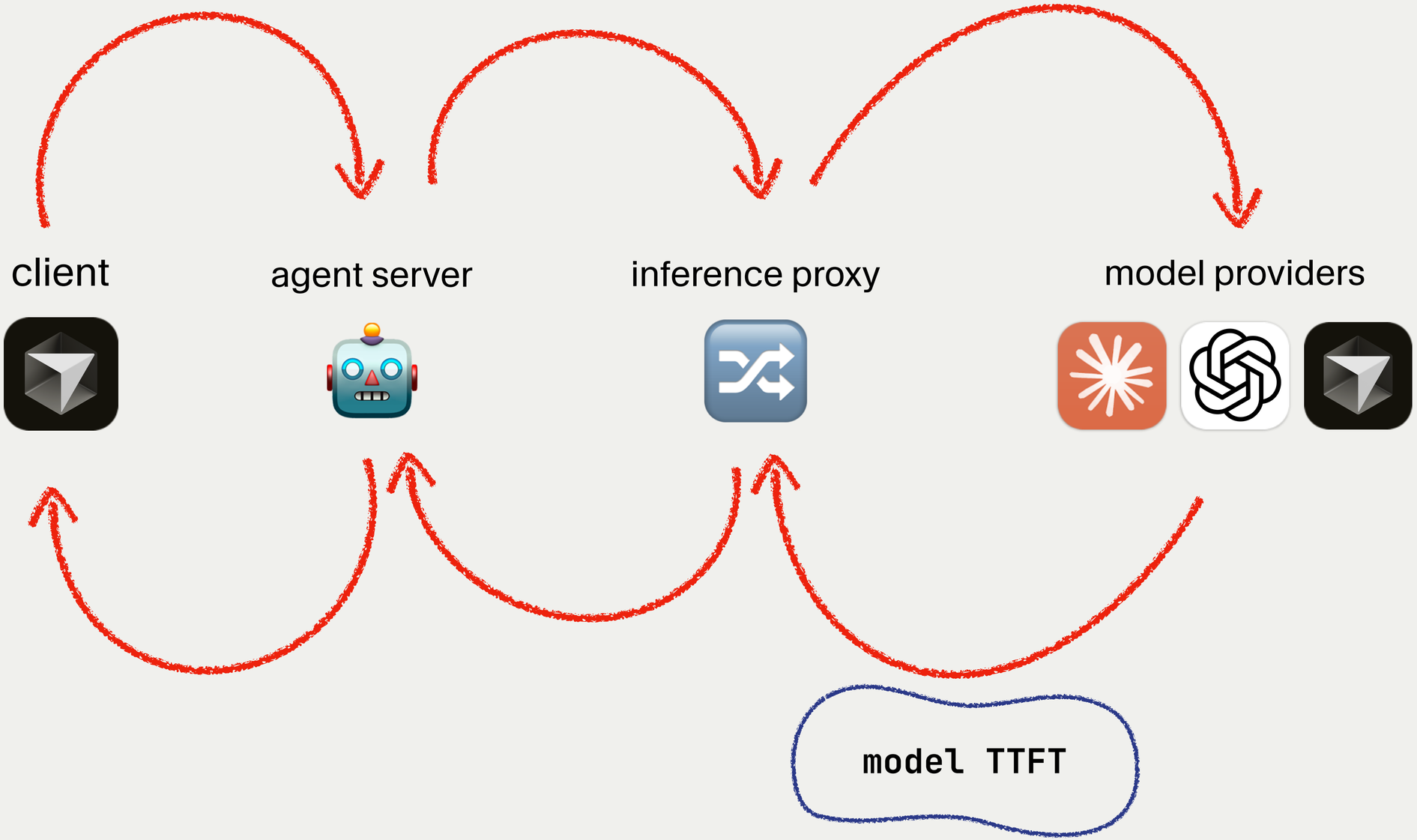


Wrinkle #1 // Missing telemetry

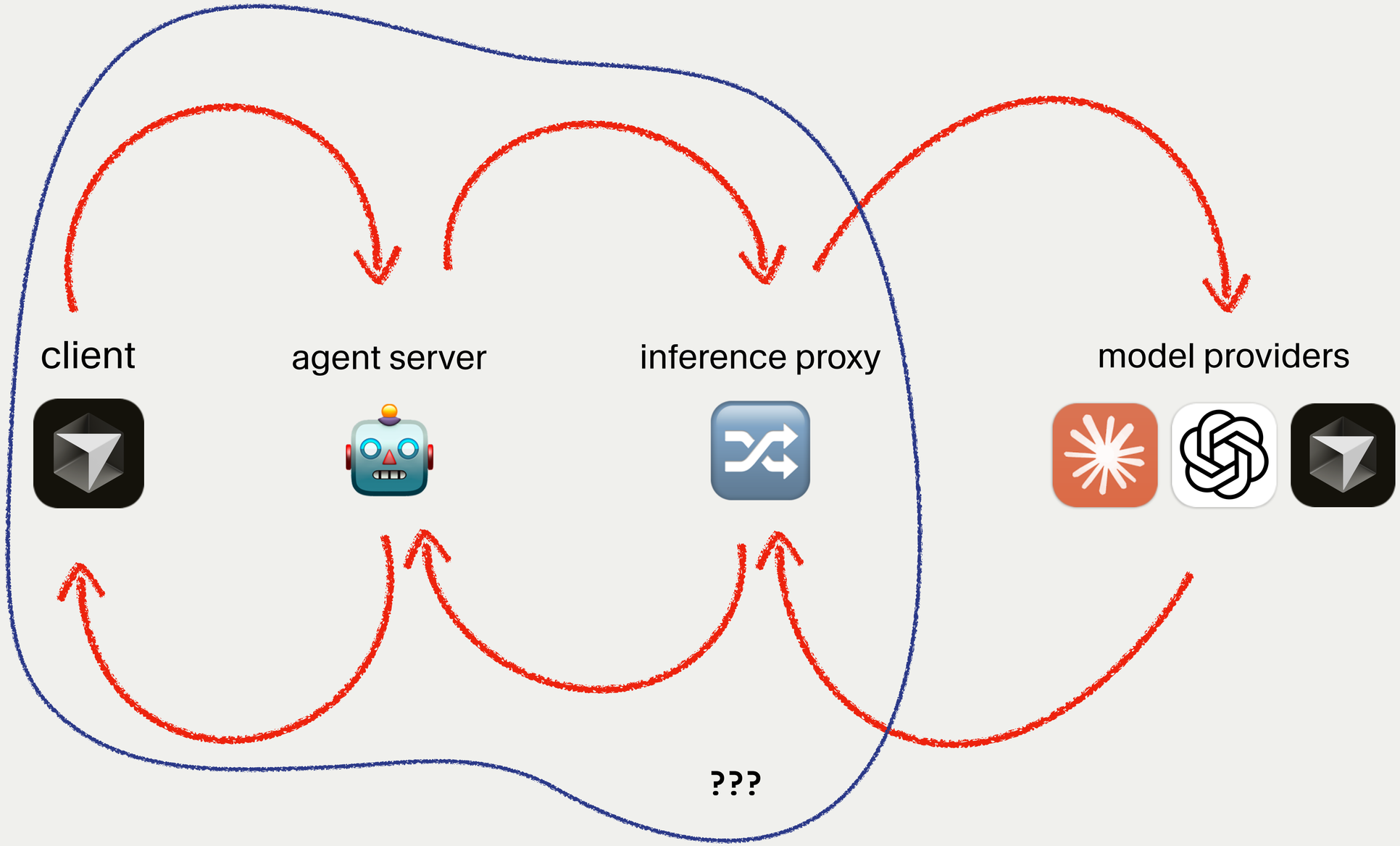
Turn 1

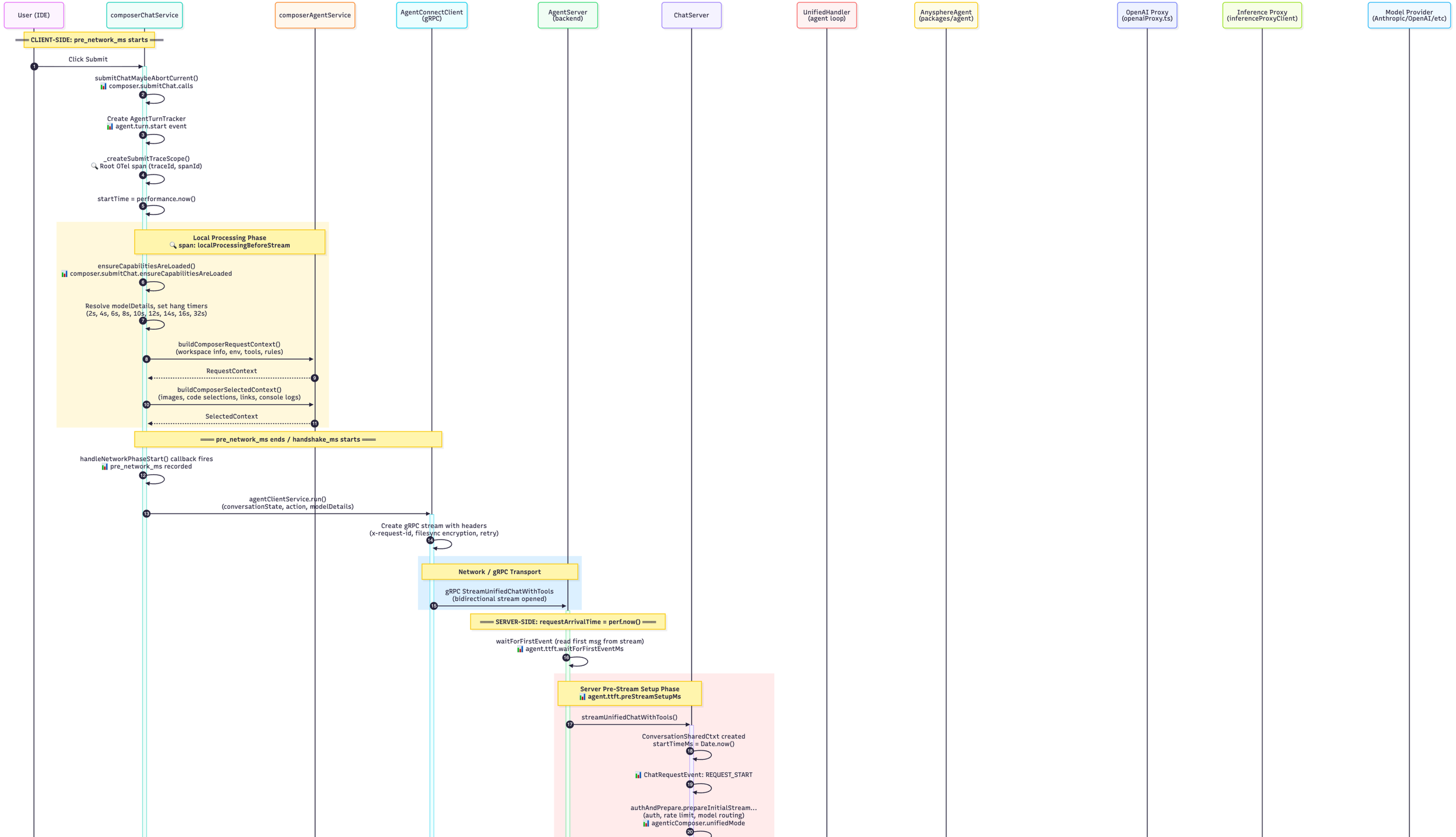

client
agent server
inference proxy
model providers
agent.turn.outcome

Turn 1


client
agent server
inference proxy
model providers
model TTFT

Turn 1




User (IDE)
composerChatService
composerAgentService
AgentConnectClient (gRPC)
AgentServer (backend)
ChatServer
UnifiedHandler (agent loop)
AnysphereAgent (packages/agent)
OpenAI Proxy (openaiProxy.ts)
Inference Proxy (inferenceProxyClient)
Model Provider (Anthropic/OpenAI/etc)
CLIENT-SIDE: pre_network_ms starts
Click Submit
1
submitChatMaybeAbortCurrent()
composer.submitChat.calls
2
Create AgentTurnTracker
agent.turn.start event
3
_createSubmitTraceScope()
Root OTel span (traceId, spanId)
4
startTime = performance.now()
5
Local Processing Phase
span: localProcessingBeforeStream
ensureCapabilitiesAreLoaded()
composer.submitChat.ensureCapabilitiesAreLoaded
6
Resolve modelDetails, set hang timers
(2s, 4s, 6s, 8s, 10s, 12s, 14s, 16s, 32s)
7
buildComposerRequestContext()
(workspace info, env, tools, rules)
8
RequestContext
9
buildComposerSelectedContext()
(images, code selections, links, console logs)
10
SelectedContext
11
pre_network_ms ends / handshake_ms starts
handleNetworkPhaseStart() callback fires
pre_network_ms recorded
12
agentClientService.run()
(conversationState, action, modelDetails)
13
Create gRPC stream with headers
(x-request-id, filesync encryption, retry)
14
Network / gRPC Transport
gRPC StreamUnifiedChatWithTools
(bidirectional stream opened)
15
SERVER-SIDE: requestArrivalTime = perf.now()
waitForFirstEvent (read first msg from stream)
agent.ttft.waitForFirstEventMs
16
Server Pre-Stream Setup Phase
agent.ttft.preStreamSetupMs
streamUnifiedChatWithTools()
17
ConversationSharedCtxt created
startTimeMs = Date.now()
18
ChatRequestEvent: REQUEST_START
19
authAndPrepare.prepareInitialStream...
(auth, rate limit, model routing)
agenticComposer.unifiedMode
20

Wrinkle #2 // Considering the p99

Wrinkle #2 // Considering the p99

Search results (100+) Best matches

✨April Goals for Server Side TTFT
Degraded Sessions: Faster TTFT • Maude Lemaire • Edited 7h ago

Open TTFT Questions
Maude Lemaire • Edited Apr 17

⚡ Degraded Sessions: Faster TTFT
Maude Lemaire • Edited Apr 21

🌍 TTFT & Geography: Nomadic Machine Analysis
Maude Lemaire • Edited Mar 20
across regions shows far users are consistently 1.3-1.7x slower at p50 across every stratum. The ~2s of additional client overhead at p50 for far users lives in the client-...

JWT Auth for TTFT Most viewed
Degraded Sessions: Faster TTFT • Maude Lemaire • Edited Apr 21

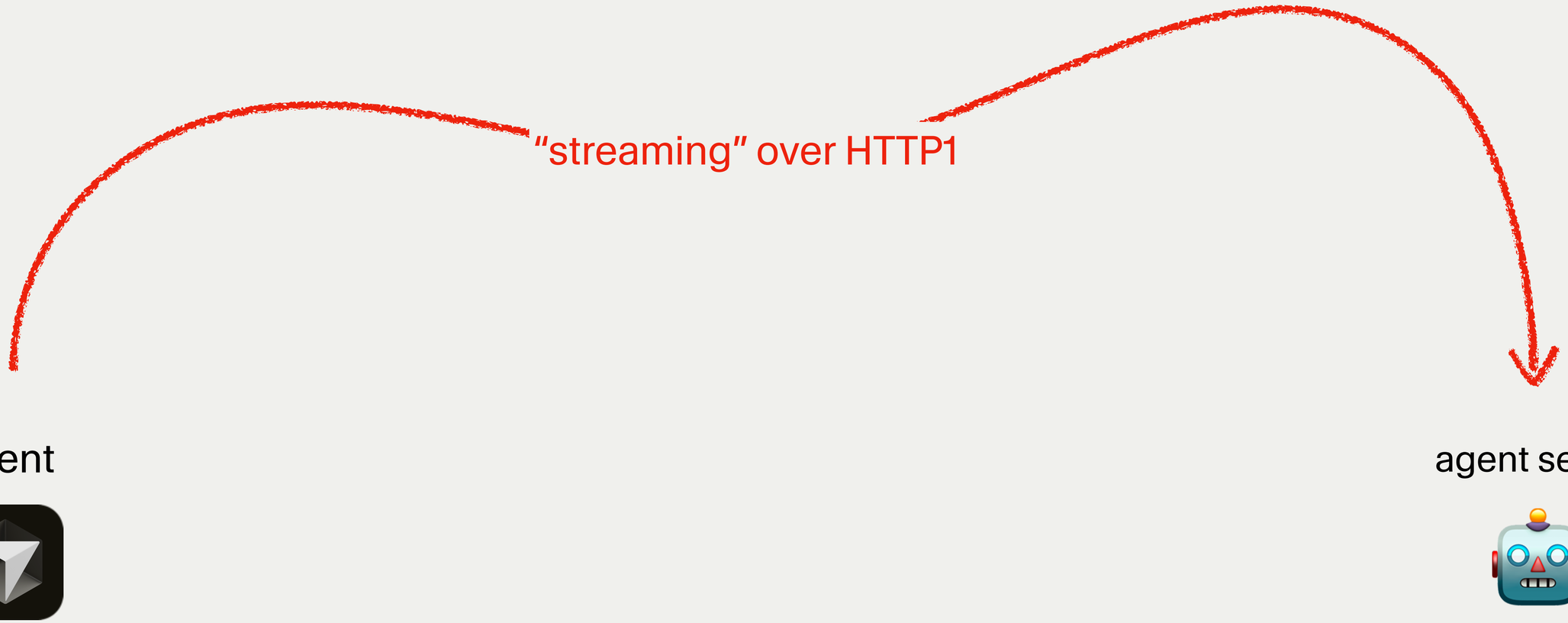
🎯 MRI SLOs (Draft)
Maude the Code Toad / MRI Thoughts • Maude Lemaire • Edited May 26
Why. TTFT alerts are the #1 after-hours page in the during March/April (Baseten neb1/fld10/fld11/uw2/primary TTFT pages every single shift). Latency is the dimension

Annotated Complete TTFT Flow
Maude Lemaire • Edited Mar 13
TTFT flow from the client, to the server, to inference proxy, all the way back to the client. It's annotated with all of the metrics and events we fire along the way to...

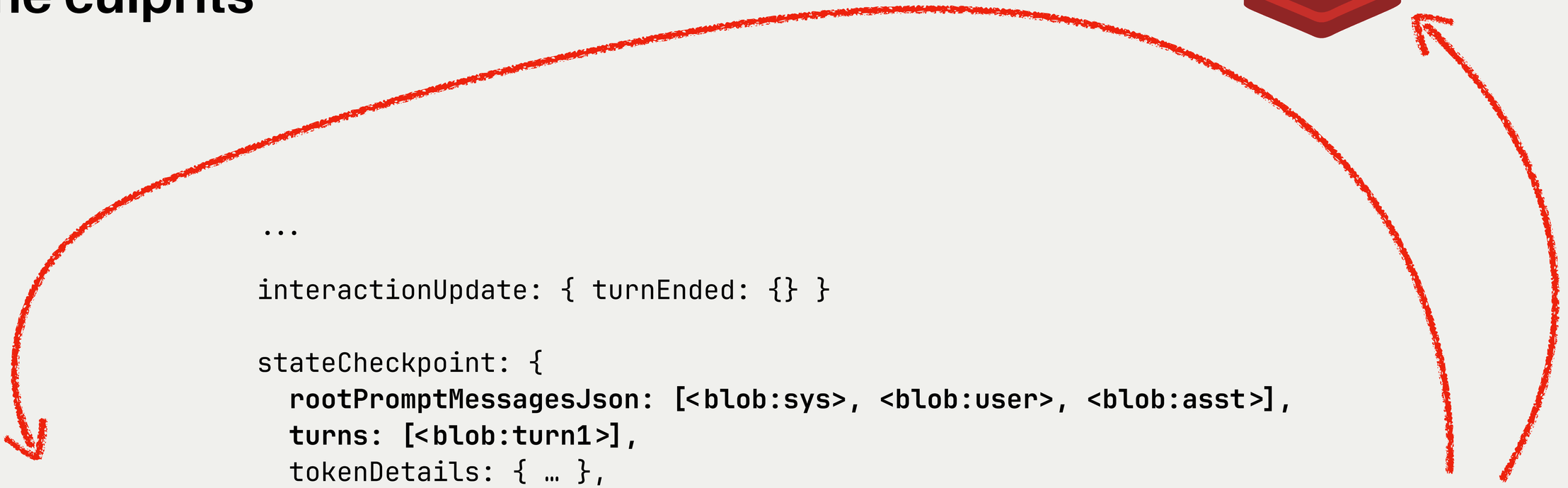
The culprits

Returning user over **HTTP/1** with **blobs out of cache**

The culprits



The culprits



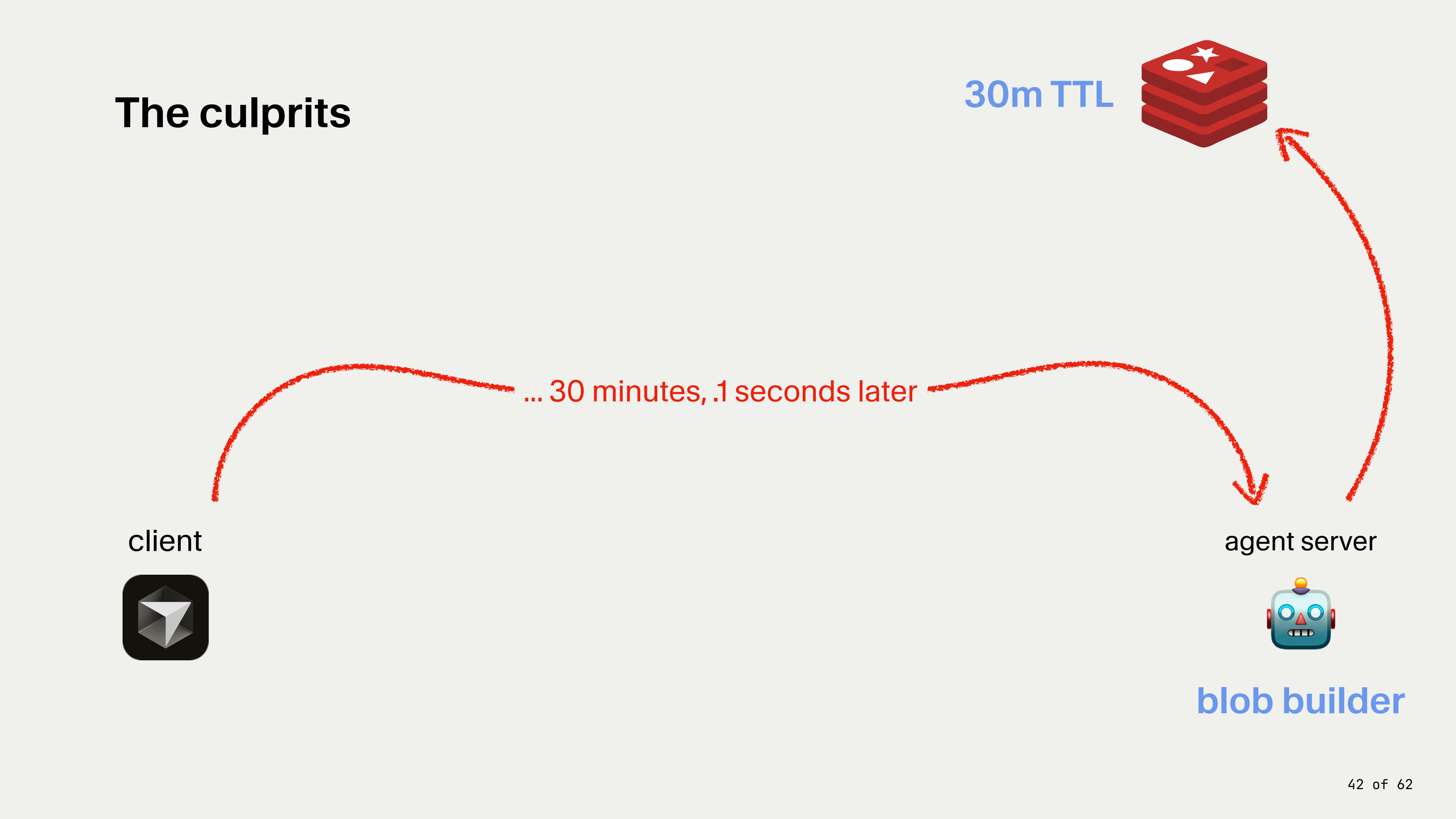
```
...

interactionUpdate: { turnEnded: {} }

stateCheckpoint: {
  rootPromptMessagesJson: [<blob:sys>, <blob:user>, <blob:asst>],
  turns: [<blob:turn1>],
  tokenDetails: { … },
}
```

client

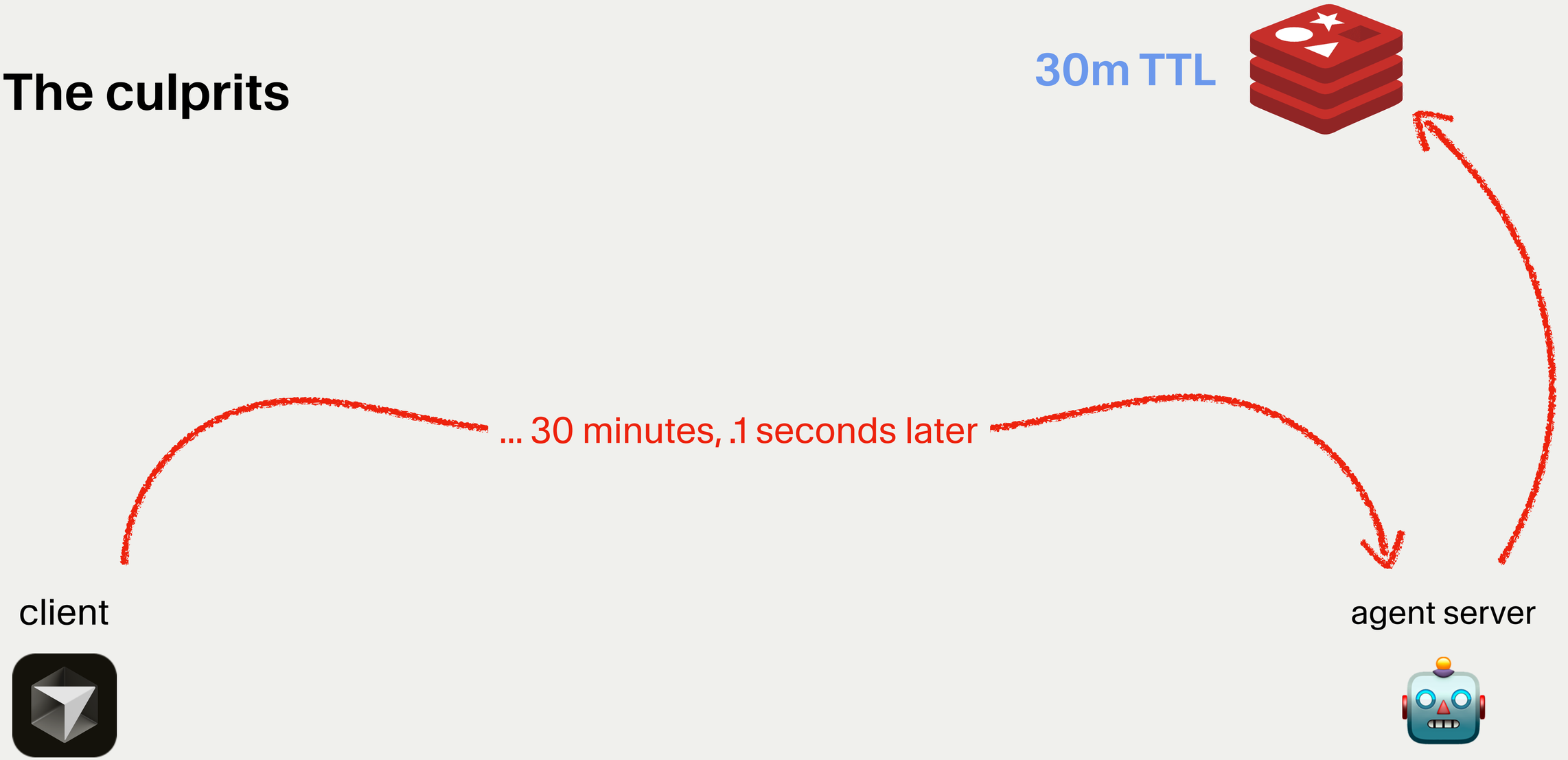
agent server

blob builder

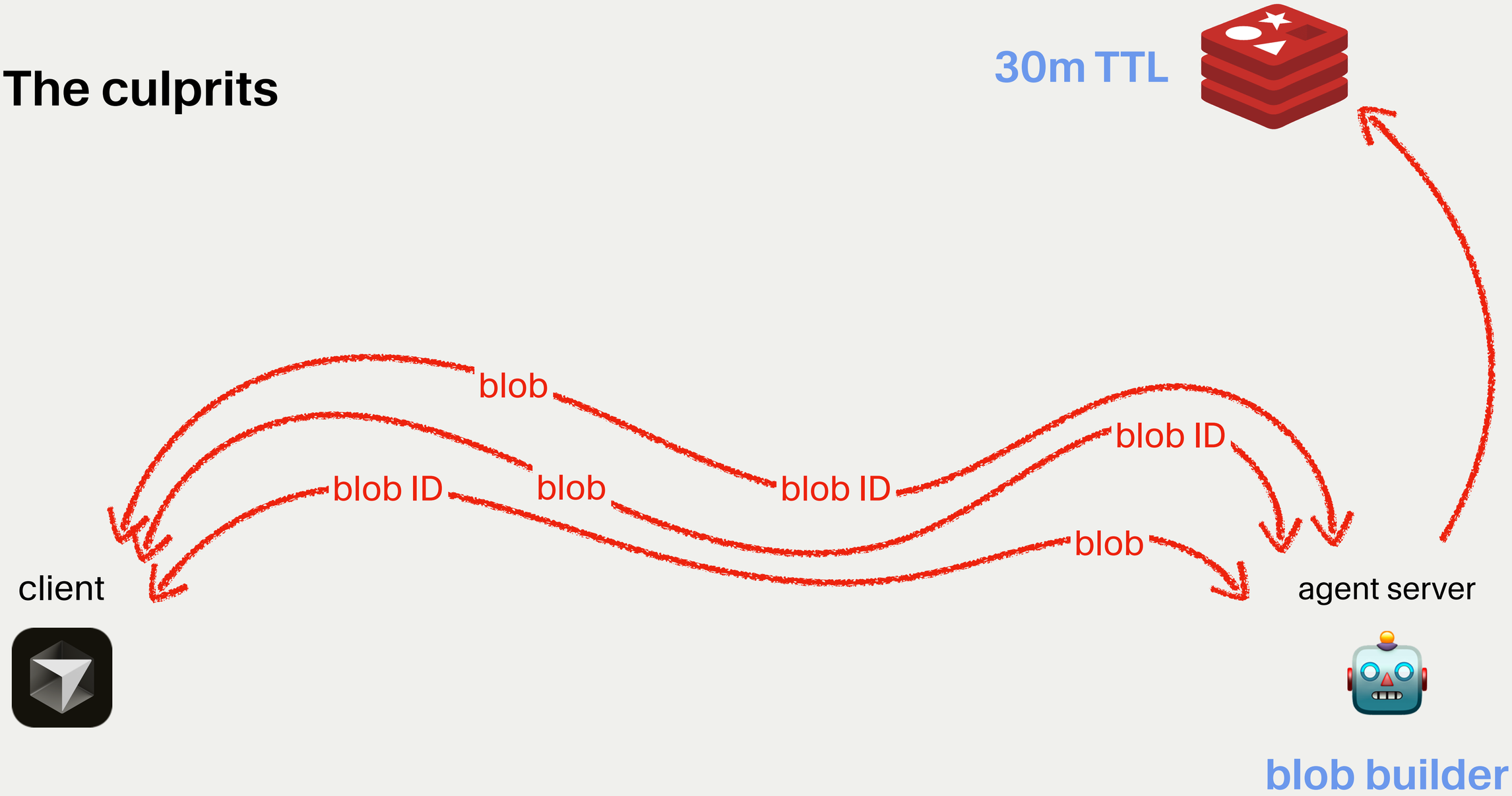
The culprits



The culprits



The culprits



Optimizations

Redis TTL

1. Increase Redis KV Blob Store TTL from 30 min to 2 hours

Status: ✅ Live and fully rolled out; dynamic config here.

The current 30-minute Redis blob store TTL creates a sharp latency cliff: turns arriving past the 30 min TTL see **8-22x higher state deserialization latency.**

Gap Bucket	N	p50 ms	p90 ms	p95 ms	p99 ms
≤ 5min	41.5M	**62**	**225**	395	1,582
5-30min	15.1M	**66**	**244**	469	2,221
30-45min	**1.2M**	**554**	**5,311**	**14,752**	**37,584**
45min-1hr	**640K**	**570**	**5,650**	**15,584**	**38,416**
1-1.5hr	714K	568	5,549	15,600	38,460
1.5-2hr	404K	580	6,078	16,244	39,006
2-4hr	670K	586	6,270	16,436	38,868
> 4hr	1.1M	597	5,689	15,915	38,983

Prefetched Blobs

Turn 2

client

agent server

```
selectedContext: [],
requestContext: {
    env: { os: “darwin 25.5.0”, … },
    gitRepos: { root: “/Users/maude/repo”, … },
    agentSkills: [ { name: “unslop”, … } ],
}
conversationState: [<blob:refs>],
action: { text: “Tell me a punny joke about Cursor’s TTFT” }
modelDetails: { modelName: “composer-2.5-fast” }
```

Turn 2

client

agent server

🤖

```
selectedContext: [],
requestContext: {
    env: { os: “darwin 25.5.0”, … },
    gitRepos: { root: “/Users/maude/repo”, … },
    agentSkills: [ { name: “unslop”, … } ],
}
conversationState: [<blob:refs>],
action: { text: “Tell me a punny joke about Cursor’s TTFT” }
modelDetails: { modelName: “composer-2.5-fast” }
prefetchedBlobs: [{id, bytes}],
```

Eliminating duplicate work on the client

Results

Sort of 😅

5 % to ~2.5 %

Wrinkle #3 // Lagging client releases

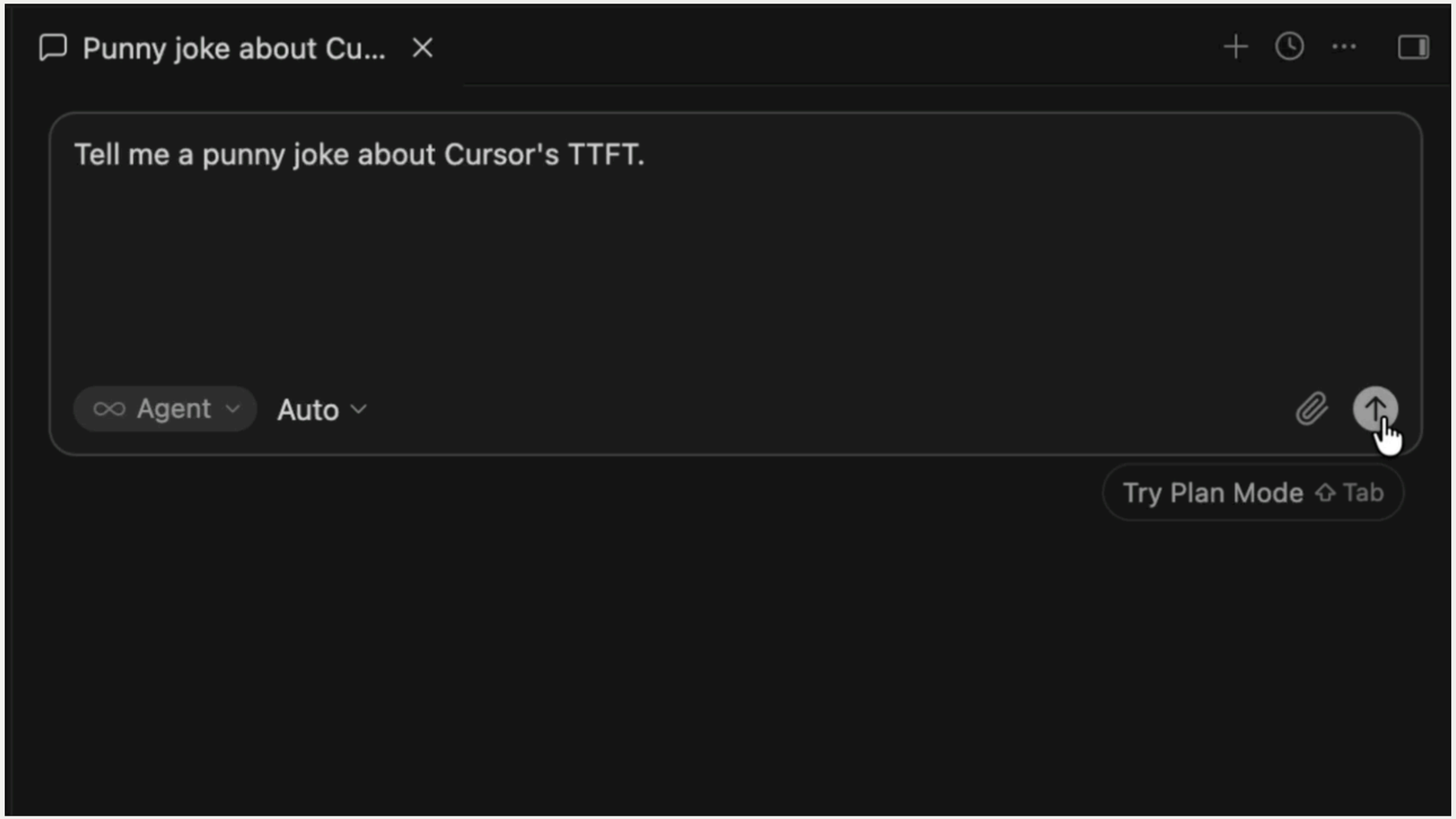
Wrinkle #4 // A big, slightly opaque composite metric

Wrinkle #4 // A big, slightly opaque composite metric

Wrinkle #4 // A big, slightly opaque composite metric

Goodhart’s Law

Goodhart’s Law


Punny joke about Cu...
Tell me a punny joke about Cursor's TTFT.
Agent
Auto
Try Plan Mode Tab

Planning next moves ...

cursor.com/careers