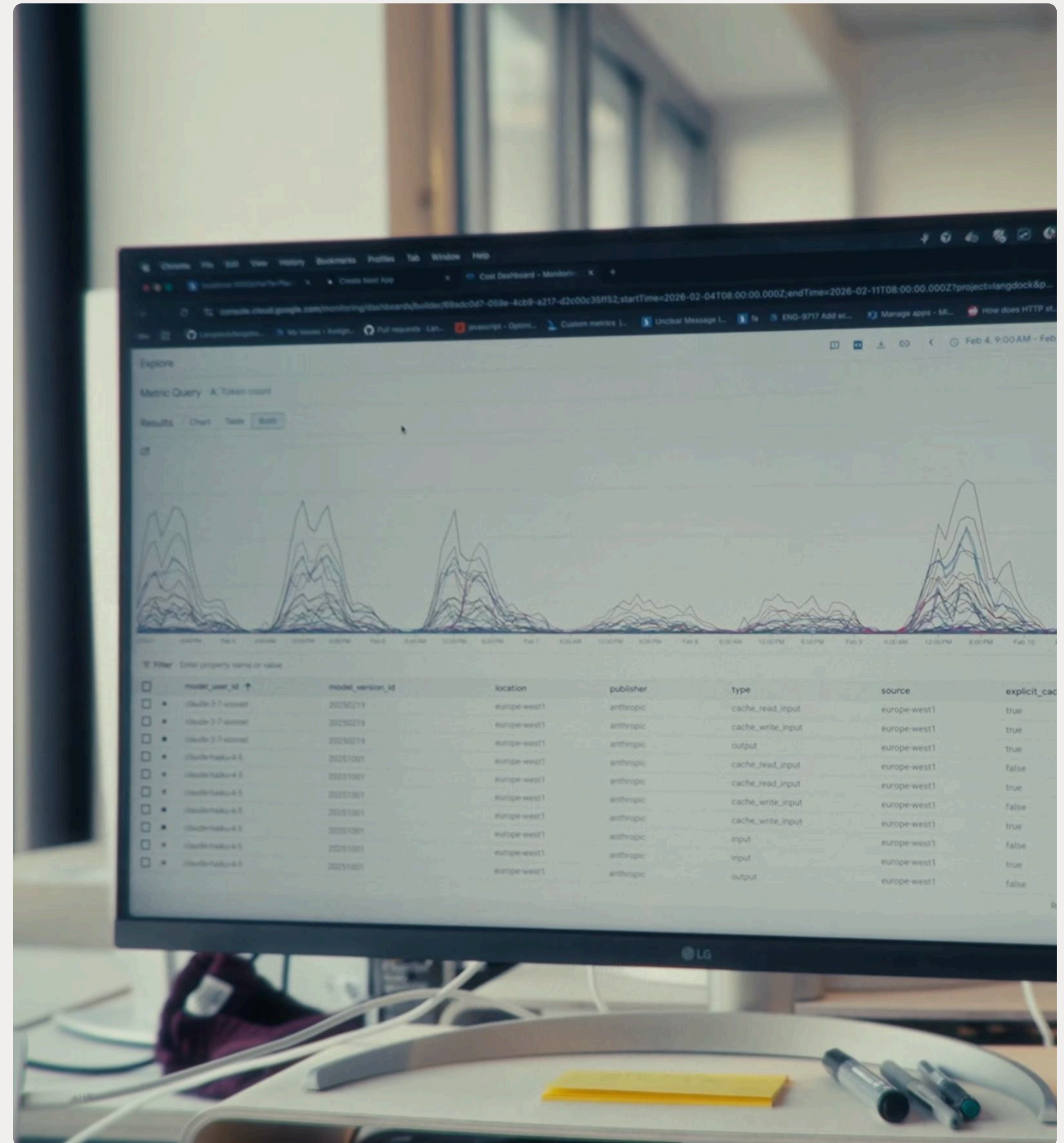


Cache Money

How Prompt Caching Cut Our LLM Bills in Half

Balázs Csomor
Engineering Lead @ Langdock





What is Langdock?

Product

- Enterprise-ready
- Security first
- Deployable anywhere
- Model agnostic

Company

- 150k+ users
- 7,000+ customers
- \$30M ARR
- Team of 50

The screenshot displays the Langdock interface. On the left is a sidebar with a navigation menu for 'Langdock Team'. The main chat area shows a conversation with an AI agent. At the top right of the chat area, a system message reads: 'Google Calendar add a 2 hour blocker before my talk today'. The chat history includes:

- AI: I'll find your talk today first.
- User: Found events >
- AI: Your talk starts at 13:55. I'll add a 2-hour blocker from 11:55 to 13:55.
- User: Created event >
- AI: Done. Added a 2-hour blocker "Blocker – Prep for Talk" from 11:55 to 13:55, right before your "Balázs – Talk @ Main Stage" at 13:55.

At the bottom of the chat area, there is a text input field with the placeholder 'Ask anything, @ for context and skills'. To the right of the input field are icons for a plus sign, a tools icon, and the text 'Tools'. Further right are icons for a refresh button, the model name 'Opus 4.7', a microphone icon, and an upward arrow icon.

The sidebar on the left contains the following items:

- Langdock Team
- New chat
- Search chats
- Inbox
- Library
- Agents
- Skills
- Workflows
- Prompts
- Integrations
- Agents >
- Projects
- New project
- Today
- Google Calendar blocker (highlighted)
- Presentation notes cleanup
- QR code generator
- Raw md grammar check
- Empty document
- Last 7 days >
- Last 30 days >
- Older than 30 days >
- Get started >
- 76% done • Almost there!

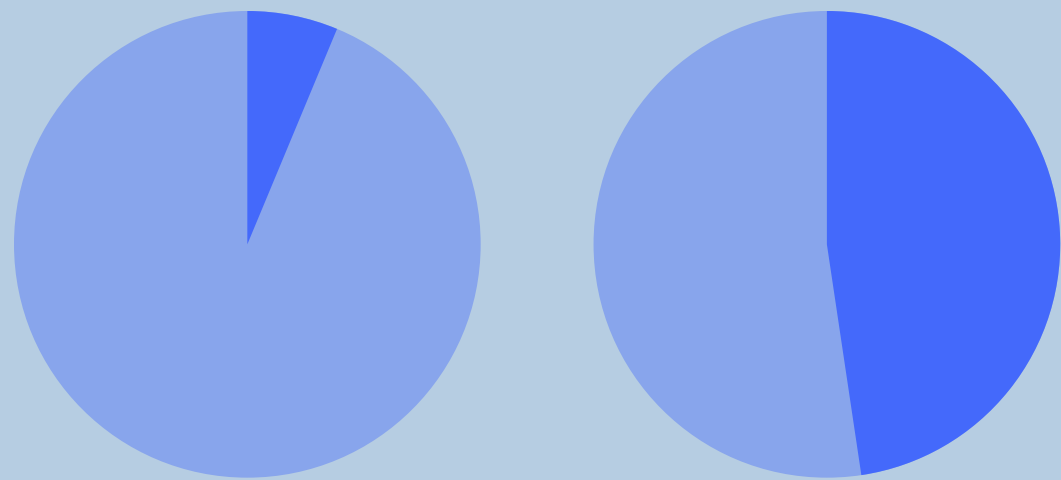


Rising LLM costs

Costs grew faster than the number of users, for two reasons

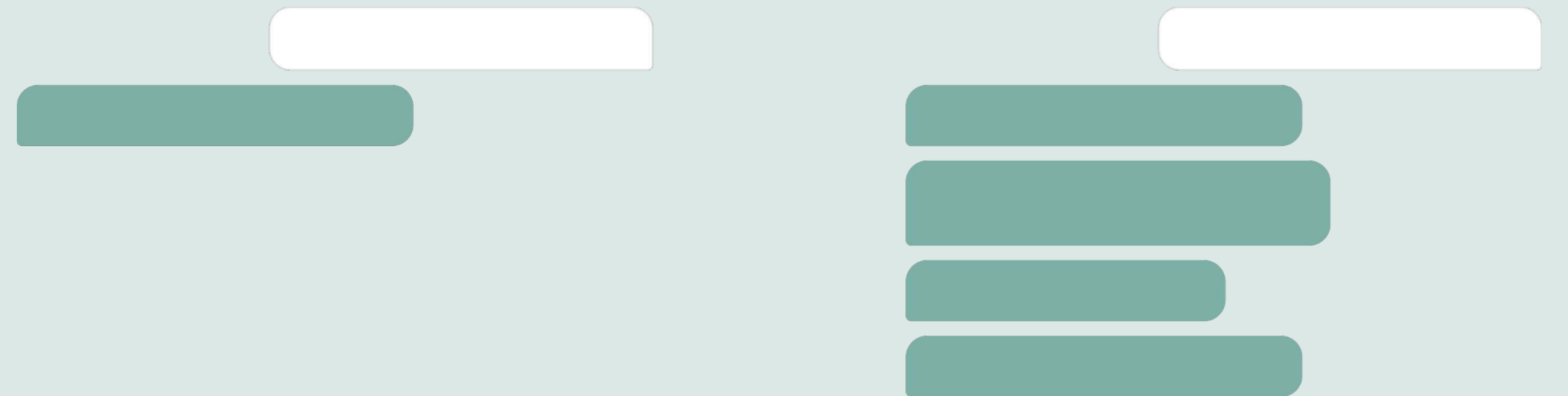
Model shift

More expensive models became more popular among Langdock users



Agentic workloads

More tool calls result in more API calls and more tokens used





What can you do about LLM costs?

...when your core business is providing access to LLMs?

~~Raise prices~~

~~Add stricter usage limits~~

~~Remove expensive models~~

~~Limit tool calling~~

~~Prompt compression~~

~~Smarter context window~~

Optimize prompt caching

Prompt caching

The basics

A provider-side feature that lets an LLM API reuse the prefix of a prompt across requests, making responses faster and cheaper.

Sending the same prompt twice is ~90% cheaper on the second request.

Request

```
1 {
2   "model": "gpt-5.2",
3   "input": [
4     {
5       "role": "user",
6       "content": "Please translate this text to English: [...]"
7     }
8   ]
9 }
```

Response #1

```
1 {
2   "usage": {
3     "input_tokens": 1455,
4     "input_tokens_details": {
5       "cached_tokens": 0
6     },
7     "output_tokens": 102,
8     "output_tokens_details": {
9       "reasoning_tokens": 0
10    },
11    "total_tokens": 1557
12  },
13  ...
14 }
```

Response #2

```
1 {
2   "usage": {
3     "input_tokens": 1455,
4     "input_tokens_details": {
5       "cached_tokens": 1280
6     },
7     "output_tokens": 108,
8     "output_tokens_details": {
9       "reasoning_tokens": 0
10    },
11    "total_tokens": 1563
12  },
13  ...
14 }
```



Prompt caching

In conversational systems

The powerful part is the prefixable nature of conversational systems.

A follow-up message can use the cache from the the previous messages.

Input tokens	1536
Cached tokens	0
Output tokens	1236

Please translate this text to English: [...]

Sure, here is a translation of the text you provided: [...]

Input tokens	2774
Cached tokens	2560
Output tokens	1246

Please translate this text to English: [...]

Sure, here is a translation of the text you provided: [...]

No, make it way less formal

Of course, here is a less formal translation of the text you provided: [...]



Prompt caching In agentic systems

LLMs in agentic systems output a sequence of assistant messages and tool calls without user input.

In a well-designed system, steps inside turns are cheap.

Add a 2 hour blocker before my talk today

I'll find your talk today first

Tool call
Find events in calendar

Tool call
Create event in calendar

Done. Added a 2-hour blocker from 11:55 to 13:55.



Prompt caching

Breaking the cache

Any change in the prefix will result in a cache key miss and the entire request will be processed at full price.

Request #1

```
1 {
2   "model": "gpt-5.2",
3   "input": [
4     {
5       "role": "user",
6       "content": "Please translate this text to English: [...]"
7     }
8   ]
9 }
```

Response #1

```
1 {
2   "usage": {
3     "input_tokens": 1455,
4     "input_tokens_details": {
5       "cached_tokens": 0
6     },
7     "output_tokens": 102,
8     "output_tokens_details": {
9       "reasoning_tokens": 0
10    },
11    "total_tokens": 1557
12  },
13  ...
14 }
```

Prompt caching

Breaking the cache

Any change in the prefix will result in a cache key miss and the entire request will be processed at full price.

Request #2

```
1 {
2   "model": "gpt-5.2",
3   "input": [
4     {
5       "role": "user",
6       "content": "Can you please translate this text to English:"
7     }
8   ]
9 }
```

Response #1

```
1 {
2   "usage": {
3     "input_tokens": 1455,
4     "input_tokens_details": {
5       "cached_tokens": 0
6     },
7     "output_tokens": 102,
8     "output_tokens_details": {
9       "reasoning_tokens": 0
10    },
11    "total_tokens": 1557
12  },
13  ...
14 }
```

Response #2

```
1 {
2   "usage": {
3     "input_tokens": 1455,
4     "input_tokens_details": {
5       "cached_tokens": 0
6     },
7     "output_tokens": 102,
8     "output_tokens_details": {
9       "reasoning_tokens": 0
10    },
11    "total_tokens": 1557
12  },
13  ...
14 }
```



The *DO*s and *DON'T*s of prompt caching

**By the time costs are a problem, you may
have already made a few mistakes.**



Turn on prompt caching

And override some basic configurations

OpenAI

Enabled by default

- Extend prompt cache retention to 24 hours as default is 5-10 minutes
`prompt_cache_retention`
- Use cache key as default is first 256 tokens
`prompt_cache_key`

Anthropic

Disabled by default

- Two modes
 - Automatic caching
 - Explicit cache breakpoints
- TTL
 - 5 minutes - 1.25x price
 - 1 hour - 2x price




Don't let tool definitions change

The order of information sent to the LLM is

1. Tool definitions
2. System prompt
3. Messages

Any change in tool definitions busts your entire cache for that conversation.

Hello :)

 Hello! What are you working on today, and how can I help?



Where's the Applied AI Conf?



 Web search



 Auto 





Don't put dynamic content in the system prompt

Don't add information that changes mid-conversation into the system prompt.

Use hidden assistant-user message pairs or tool calls instead instead.

Examples:

- Current timestamp
- Available documents

You are a helpful support assistant, helping users resolve issues.

The current time is:

Thu May 28 2026 11:01:46 GMT+0200 (Central European Summer Time)

...

Write me a project summary for Q1

Here's a general overview of your Q1 project...

No, use our internal template

I don't have access to your templates. Could you paste it here?

Here's the template. Now rewrite it...



Don't put dynamic content in the system prompt

Don't add information that changes mid-conversation into the system prompt.

Use hidden assistant-user message pairs or tool calls instead instead.

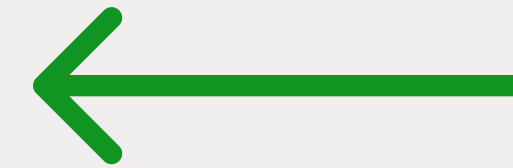
Examples:

- Current timestamp
- Available documents

You are a helpful support assistant, helping users resolve issues.

...

Tool call
Get current time



Write me a project summary for Q1

Here's a general overview of your Q1 project...

No, use our internal template

I don't have access to your templates.
Could you paste it here?

Here's the template. Now rewrite it...

Sure, here's another attempt...



Don't put dynamic content in the system prompt

Don't add information that changes mid-conversation into the system prompt.

Use hidden assistant-user message pairs or tool calls instead instead.

Examples:

- Current timestamp
- Available documents

You are a helpful support assistant, helping users resolve issues.

...

Hello :)

Hi there! What can I help you with?



Don't put dynamic content in the system prompt

Don't add information that changes mid-conversation into the system prompt.

Use hidden assistant-user message pairs or tool calls instead instead.

Examples:

- Current timestamp
- Available documents

You are a helpful support assistant, helping users resolve issues.

You have access to the following documents:
- presentation.pdf

...

Hello :)

Hi there! What can I help you with?

 presentation.pdf

Can you translate this document?

Sure! I will go ahead and translate the document you uploaded...



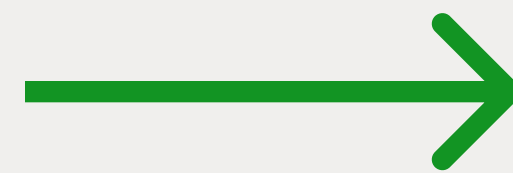
Don't put dynamic content in the system prompt

Don't add information that changes mid-conversation into the system prompt.

Use hidden assistant-user message pairs or tool calls instead.

Examples:

- Current timestamp
- Available documents



You are a helpful support assistant, helping users resolve issues.
...


Hello :)

Hi there! What can I help you with?



[HINT] I uploaded the following document: presentation.pdf, you have now access to it.

[ACK] The user has uploaded a document.

 presentation.pdf
Can you translate this document?

Sure! I will go ahead and translate the document you uploaded...

Make serializations deterministic

Semantically identical tool definitions, tool results, serialized JSON objects can break the cache.

Always sort JSON/YAML keys consistently (e.g. by alphabetical order).

```
1 {
2   type: "function",
3   name: "get_weather",
4   description: "Get current temperature",
5   parameters: {
6     type: "object",
7     properties: {
8       location: {
9         type: "string",
10        description: "City and country",
11      },
12      unit: {
13        type: "string",
14        description: "Temperature unit",
15        enum: ["celsius", "fahrenheit"],
16      },
17    },
18    required: ["location"]
19  }
20 }
```

```
1 {
2   name: "get_weather",
3   type: "function",
4   description: "Get current temperature",
5   parameters: {
6     type: "object",
7     properties: {
8       location: {
9         type: "string",
10        description: "City and country",
11      },
12      unit: {
13        type: "string",
14        description: "Temperature unit",
15        enum: ["celsius", "fahrenheit"],
16      },
17    },
18    required: ["location"]
19  }
20 }
```



Don't use a rolling context window

Dropping oldest messages that don't fit into the context window will break the cache on every message.

Use summarization, compaction or another context window management strategy instead.

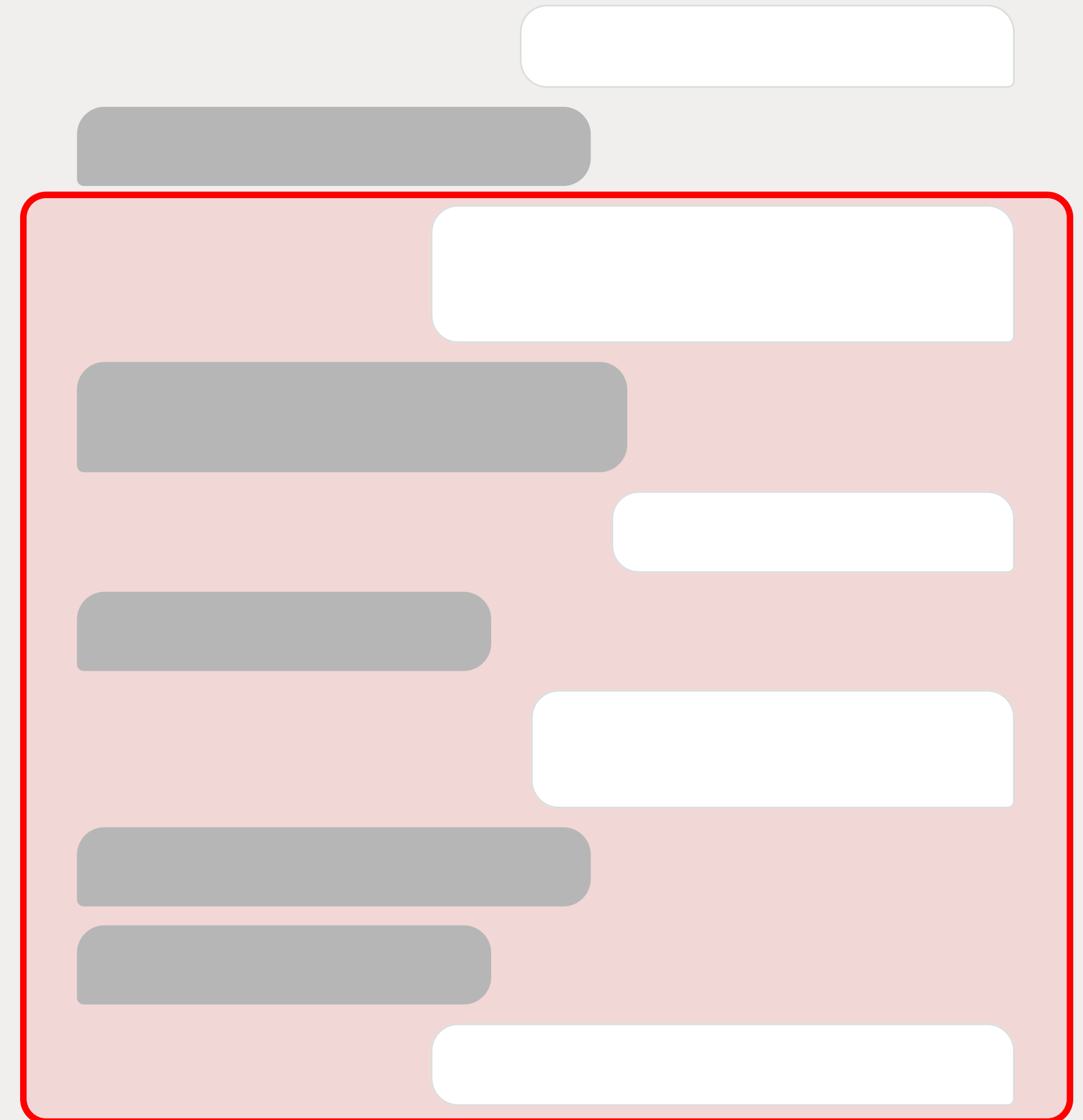




Don't use a rolling context window

Dropping oldest messages that don't fit into the context window will break the cache on every message.

Use summarization, compaction or another context window management strategy instead.





Don't use a rolling context window

Dropping oldest messages that don't fit into the context window will break the cache on every message.

Use summarization, compaction or another context window management strategy instead.





Build observability & developer tooling

Monitor cache hit rate (and costs) in production.

Build developer tooling and guardrails.

Model Cache Rate

Smooth

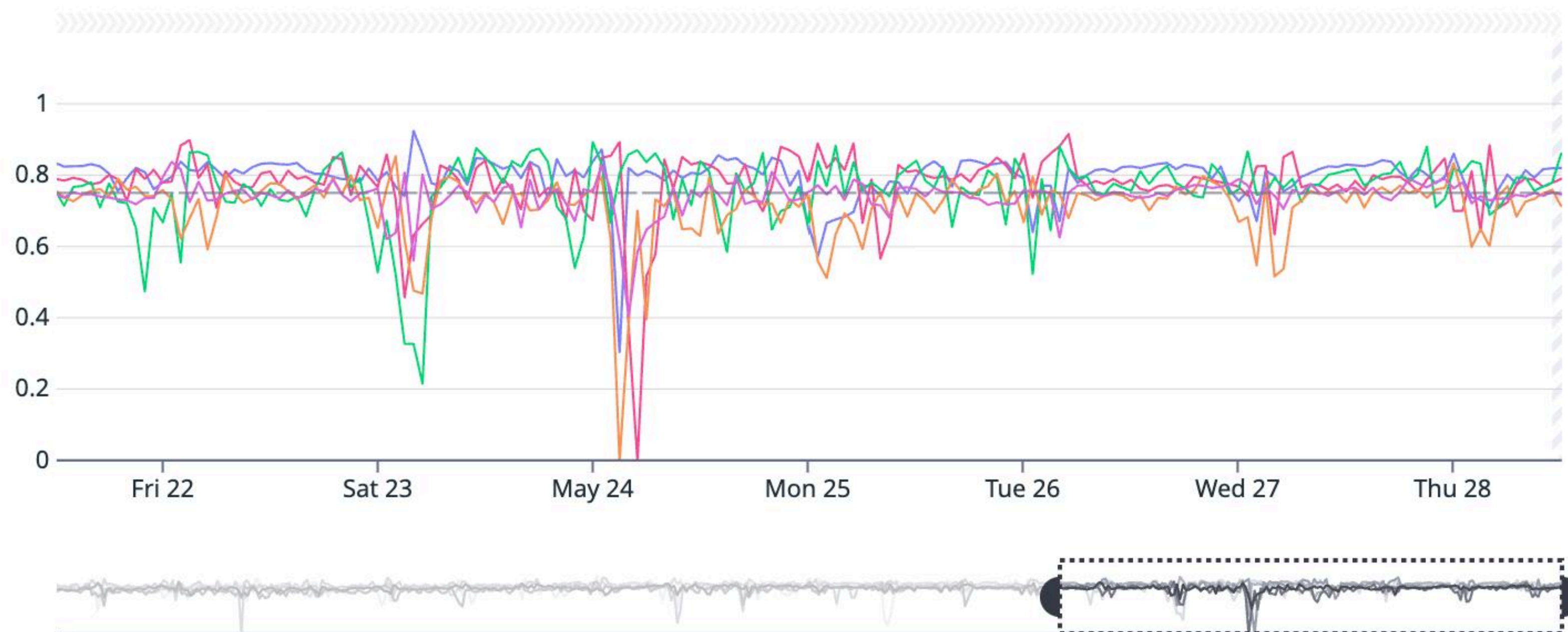
Trend Line

Forecast



Share

No findings



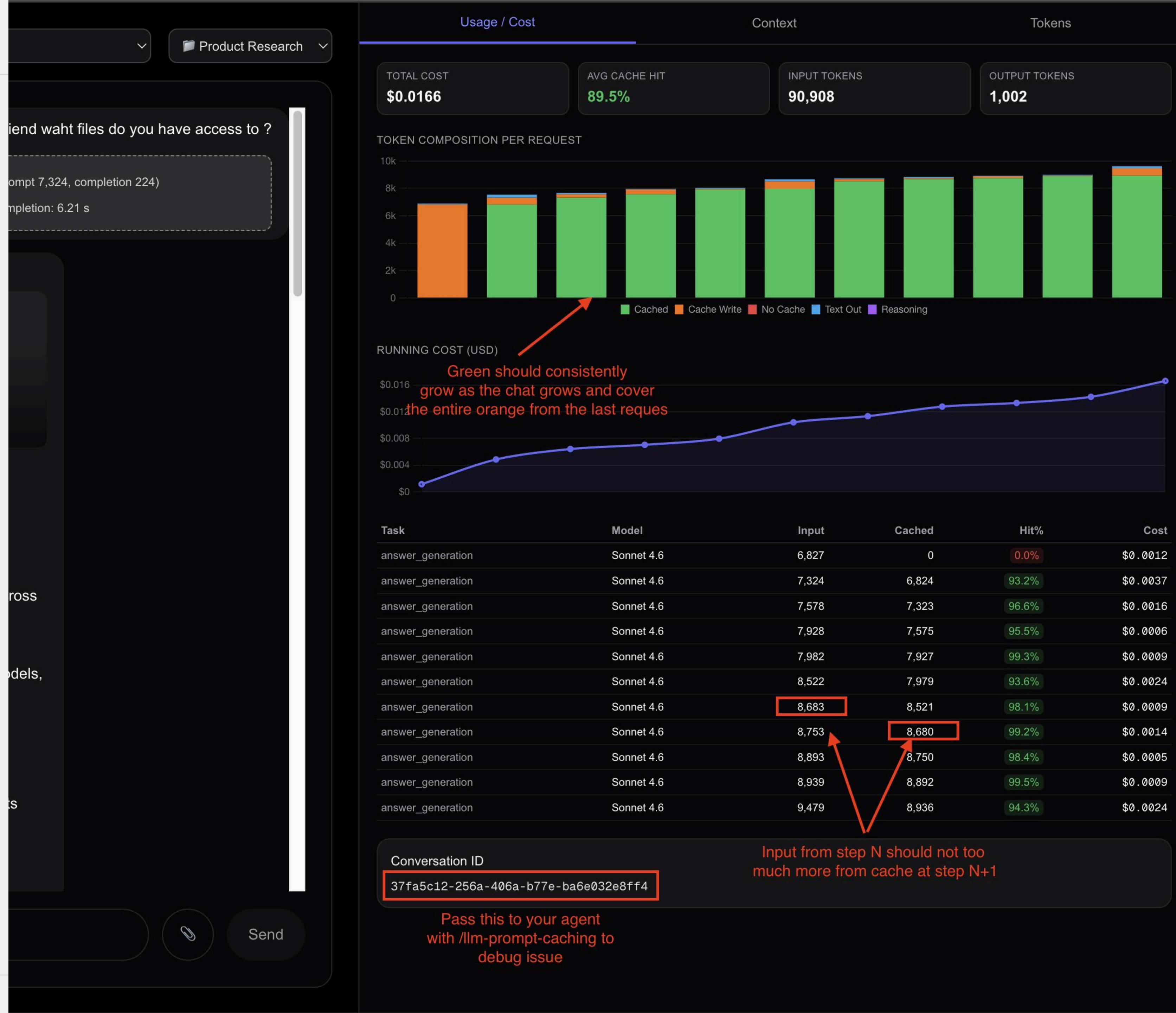
Filter series

	model in default_zero(sum:engine.llm_r...	FINDINGS	AVG	MIN	MAX	SUM	VALUE
	gpt-5.2		0.80	0.30	0.93	135	0.82
	claude-opus-4-6		0.78	0	0.92	131	0.79
	gpt-5.4		0.76	0.22	0.89	128	0.86
	claude-opus-4-7		0.71	0	0.85	121	0.71
	claude-sonnet-4-6		0.74	0.41	0.84	125	0.75

Build observability & developer tooling

Monitor cache hit rate (and costs) in production.

Build developer tooling and guardrails.





Three open questions

How do we educate the user?

Stale conversations are more expensive.

Should we always summarize them? Should the user know?

What's our long-term caching observability stack?

Currently a mix of

- Provider dashboards
- Datadog
- Production database.

Where does model provider tooling evolve?

OpenAI's 24-hour cache by default is new.

Anthropic's automatic prompt caching at is new.

What should you do next?

Still experimenting with LLMs?

Keep this in mind, don't optimize prematurely.

Already in production?

Go through the checklist.
Your margins (and your CV) will thank you.



Notes & prompt



We're hiring!
langdock.com/careers

**Make
something
people
want.**

