



Demo: MobileClaw: A Notification-Driven, Human-Gated Mobile LLM Agent

Boyuan Yang¹, Alexander Du^{2*}, Ozzie Hassel^{1*}, Jash Uganandan^{1*}, Wei Gao¹

¹ University of Pittsburgh ² North Allegheny Senior High School * Equal contribution

MobiCom 2026

GitHub

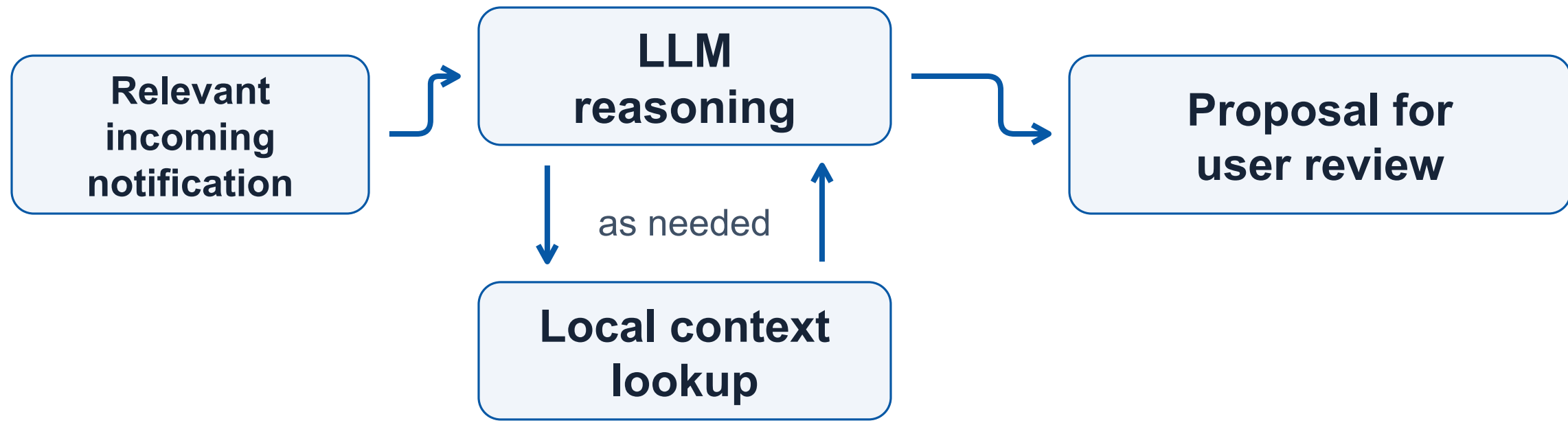


github.com/pittis/MobileClaw

Motivation & Existing Work

From notification to informed proposal

Useful assistance starts with relevant notifications. It retrieves local context as needed before proposing a response or action for review.



Existing approaches and their limits

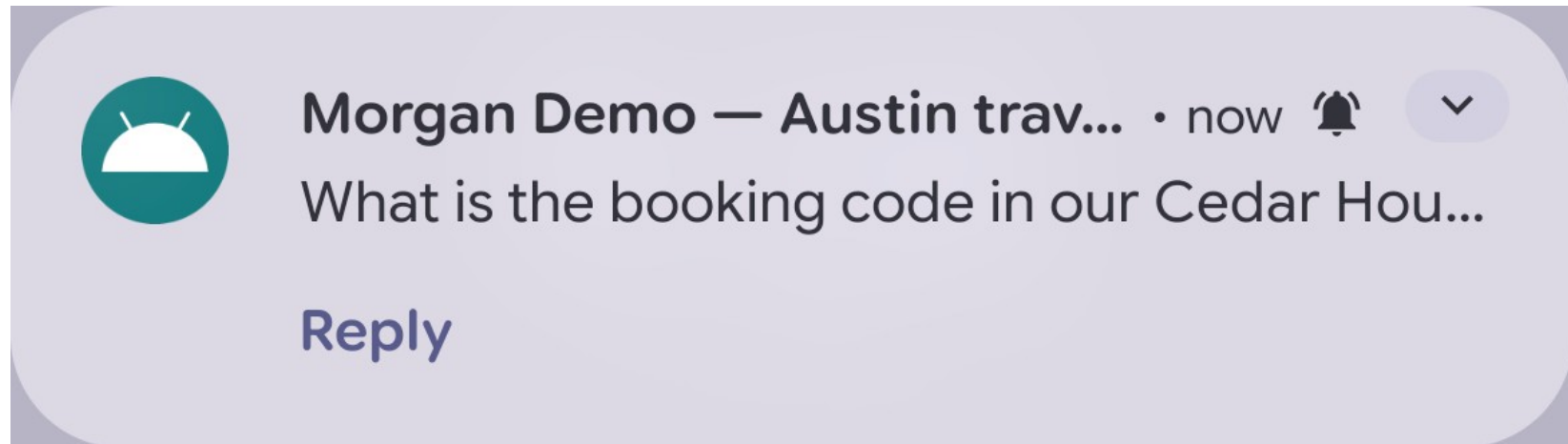
Cloud workspaces — Meta Muse

The agent workspace lives in a cloud VM; device context must be made available through integrations.

Gemini Android screen automation

Gemini offers Connected Apps and screen automation. Its screen-automation path operates app interfaces in an isolated window.

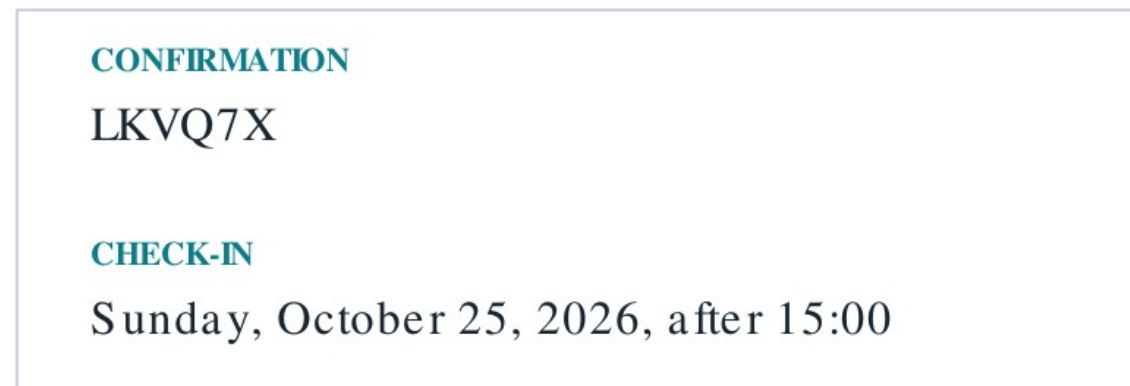
Why local context matters



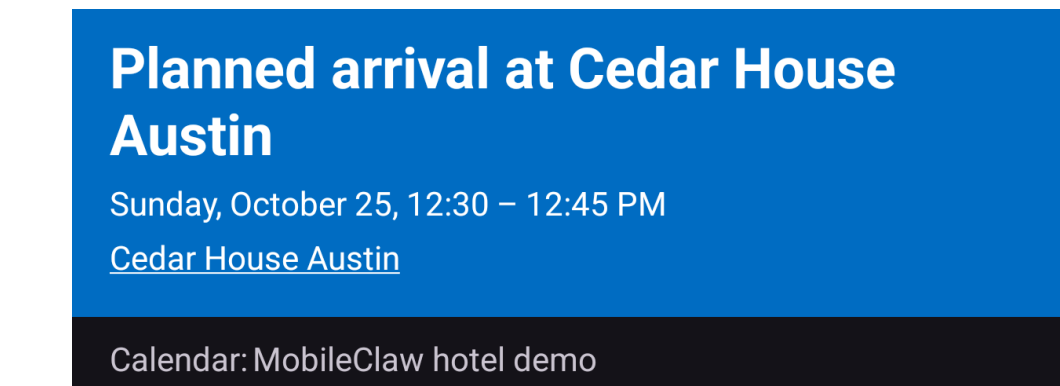
Message: “What is the booking code in our Cedar House PDF? Does our calendar show arrival before check-in?”

The reply needs booking details from the local PDF and arrival time from the calendar.

Hotel confirmation PDF

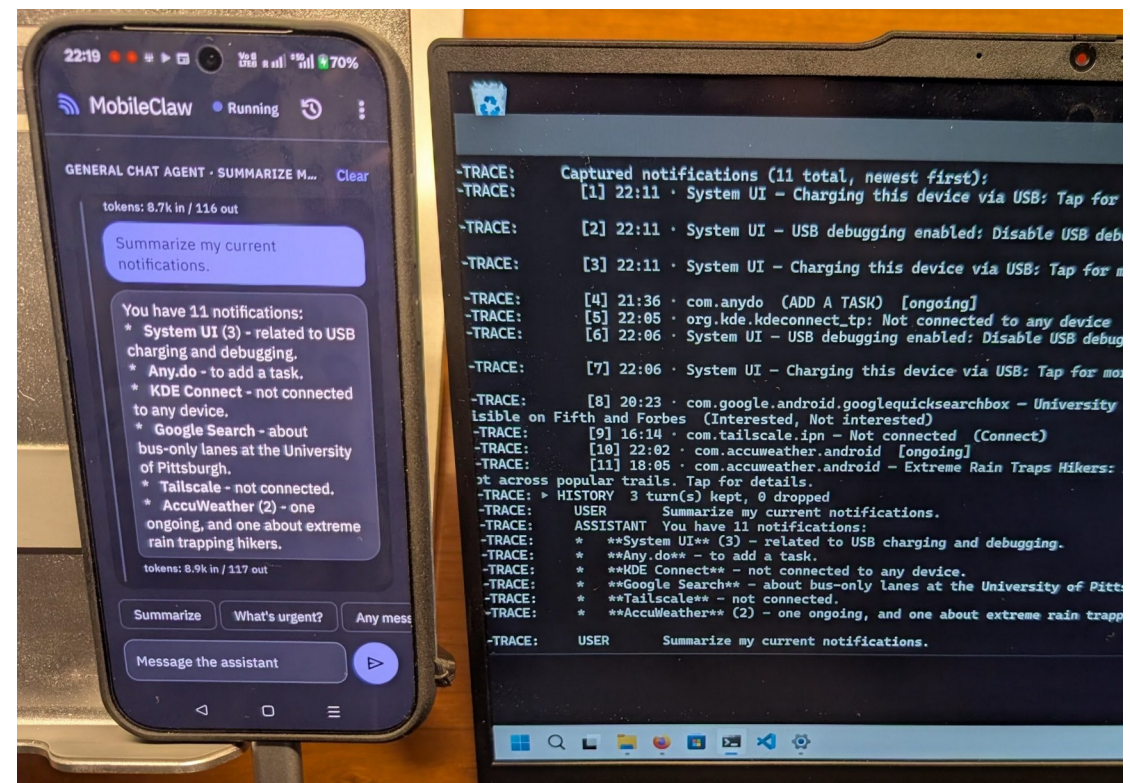


Calendar event



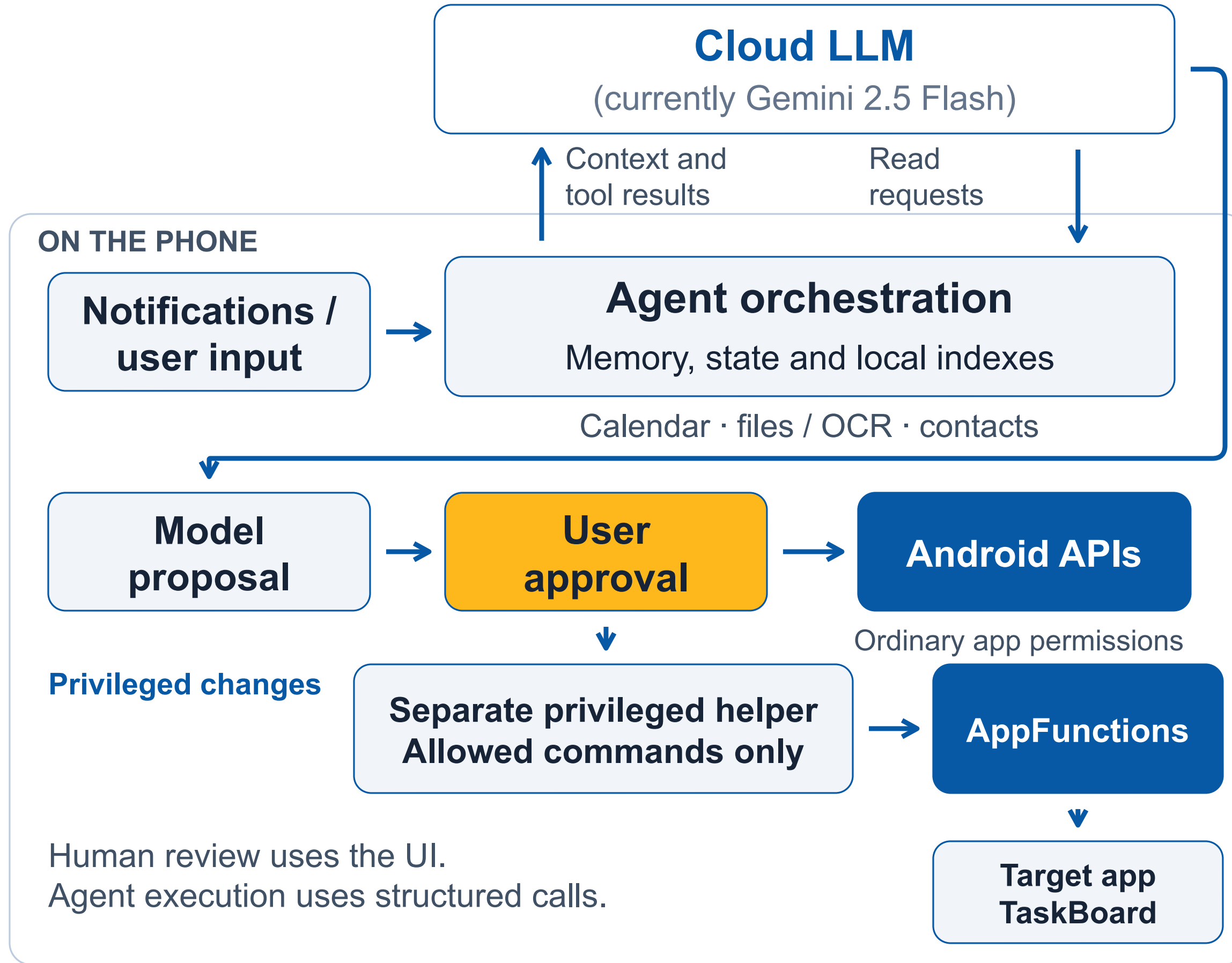
Prototype on a real phone

MobileClaw runs on the phone. The laptop displays its execution trace.



System Overview & Contributions

Local workspace, structured execution



System design

1. API execution without GUI automation

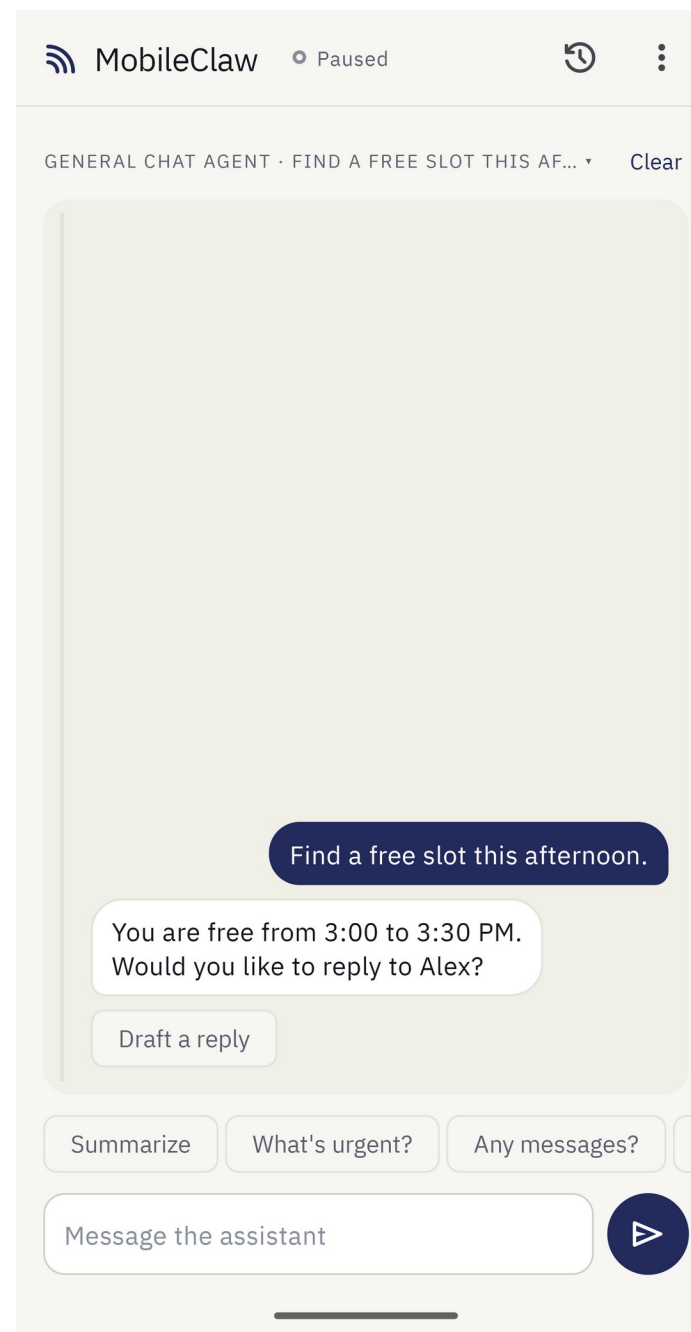
11 skills group 24 tools. Android APIs and AppFunctions execute workflows without screen parsing or simulated taps. AppFunctions coverage depends on target apps exposing functions. Gemini also previews it.

2. A phone-resident workspace

Orchestration, state and resource lookup stay on the phone. The cloud model receives task context and tool results, not a copy of the whole workspace.

3. Controlled privileged execution

Privileged changes require user approval. A separate helper rechecks allowed commands before execution. Ordinary Android API calls use app permissions.



App UI Screenshot

Conclusion

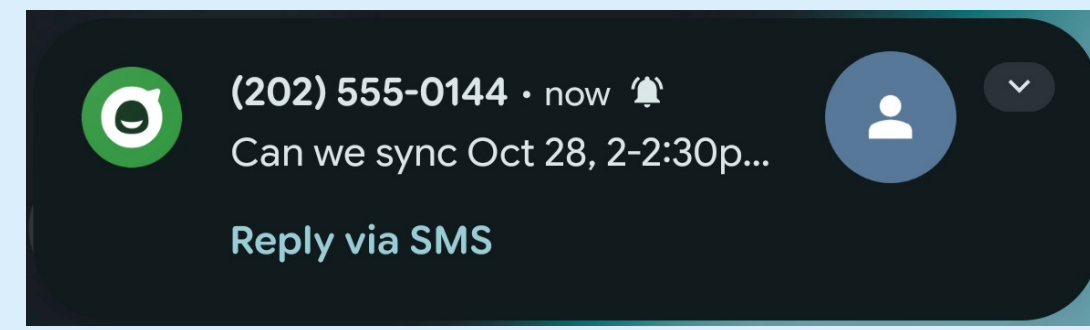
MobileClaw connects phone events and local context to app actions through structured APIs. The demonstrations show automatic lookup, proposals for user review, and approved cross-app execution.

Showcases & Skills

Schedule coordination

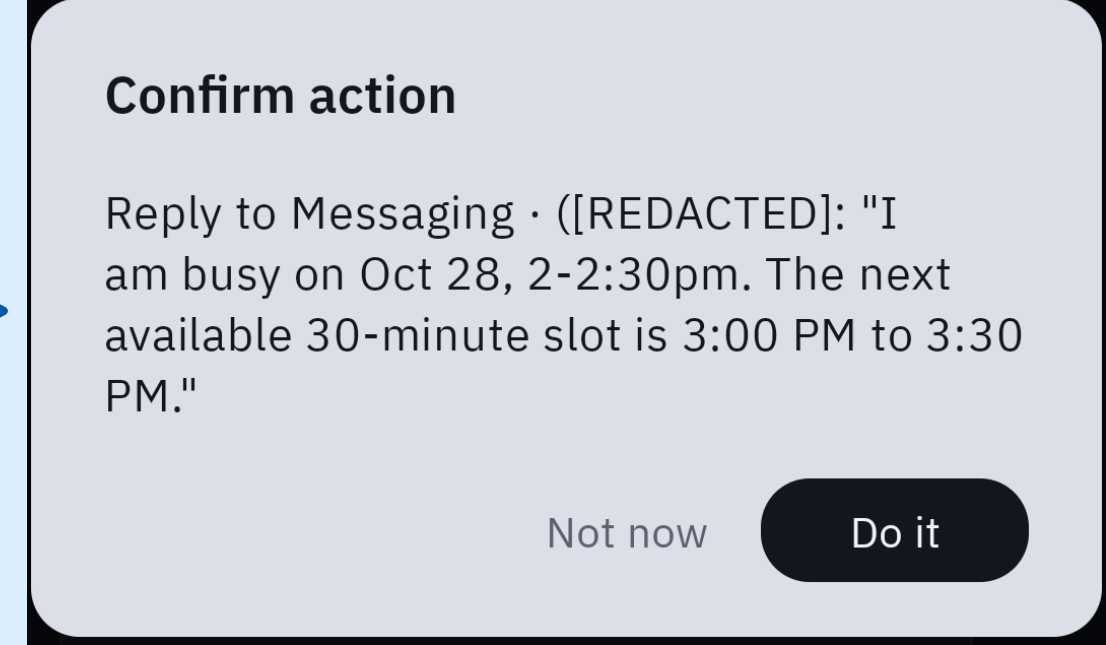
On request, reads calendar availability through Android APIs and revises the reply proposal.

Incoming request



Request: Oct 28, 2:00–2:30 PM.
Calendar: busy until 3:00 PM.

Proposed reply

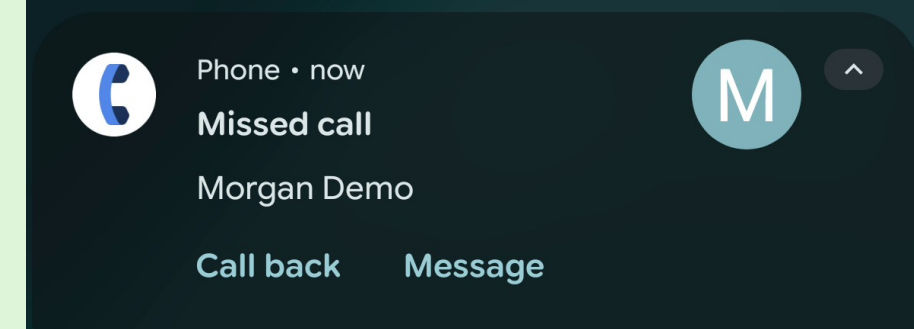


Skills: Notifications · Calendar

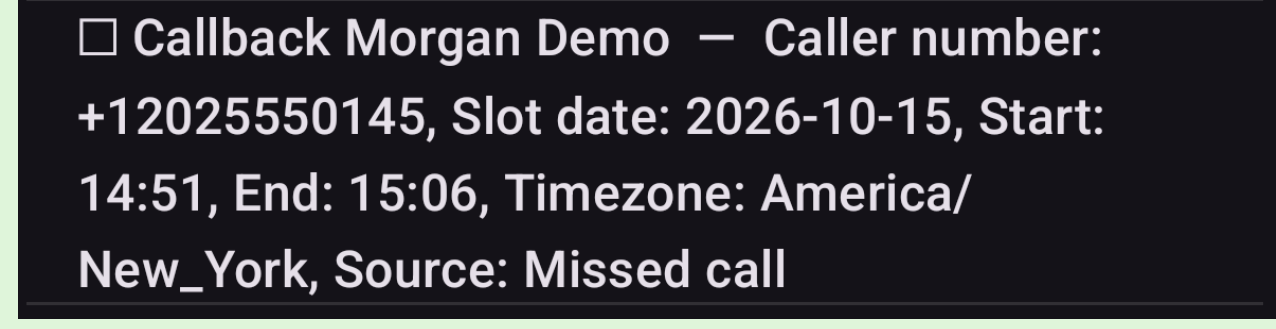
Missed call to callback task

On request, combines contact and calendar lookup, then creates a TaskBoard task after user approval.

Missed call



Created task



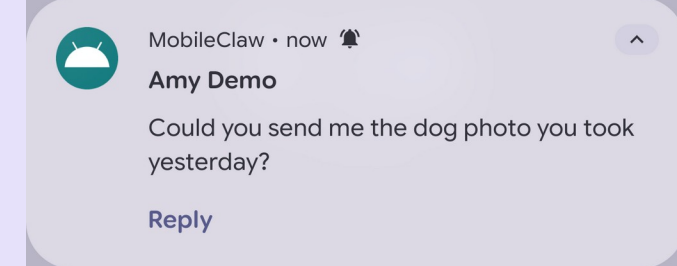
Structured execution: user approval → separate restricted helper → AppFunctions → TaskBoard

Skills: Notifications · Messaging · Calendar · AppFunctions

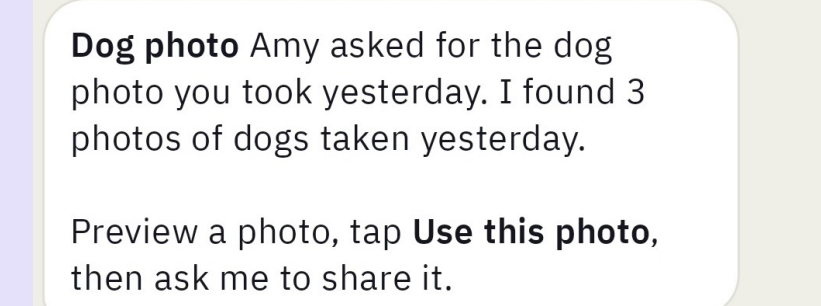
Photo search and sharing

Incoming request → automatic lookup by local index and capture date → candidates for user review.

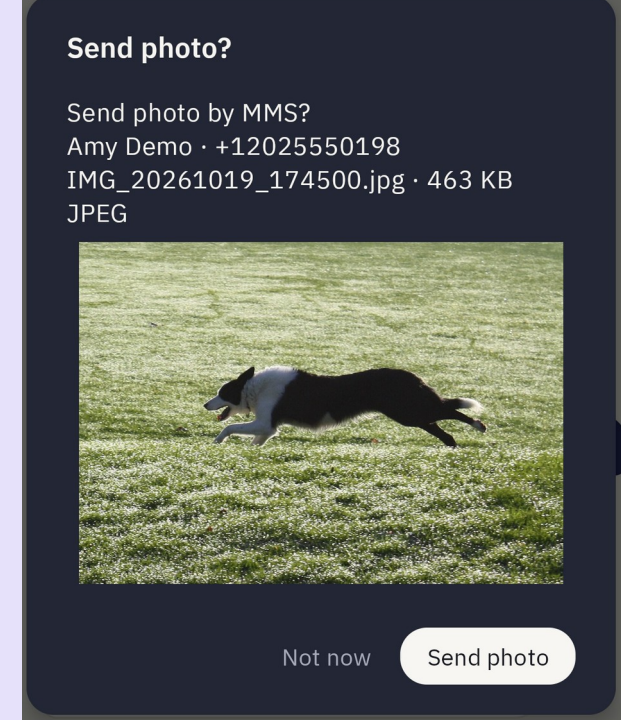
Incoming request



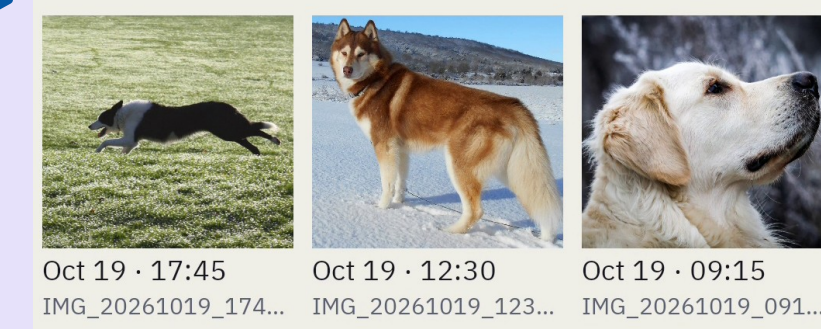
Automatic lookup



Share confirmation



“Could you send me the dog photo you took yesterday?”



User selects a photo and requests sharing

Skills: Notifications · File search · Messaging



The poster and paper are licensed under CC BY-NC-ND 4.0.
creativecommons.org/licenses/by-nc-nd/4.0/

Demo photos from Wikimedia Commons (separate licenses): Dietmar Rabich (Golden Retriever), CC BY-SA 4.0; MatthewJS (Emma the border collie), CC BY-SA 3.0; milanonegro (husky), CC0. Image pixels unchanged. Licenses: creativecommons.org/licenses/by-sa/4.0 and /by-sa/3.0