

AGENTICOS 2026 · PRAGUE

LLM Agent Capabilities Should Follow Task Intent and Context Source

IntentCap

Yusheng Zheng · Wenhui Zhang · Yu Mao

github.com/yunwei37/agentcap

An agent's least-privilege capability depends on its task intent

TRADITIONAL SYSTEMS

Capability is predefined

A programmer declares it at compile time; a type system enforces it, e.g. Capsicum file descriptors or seccomp filters.

LLM AGENTS

Capability is dynamic

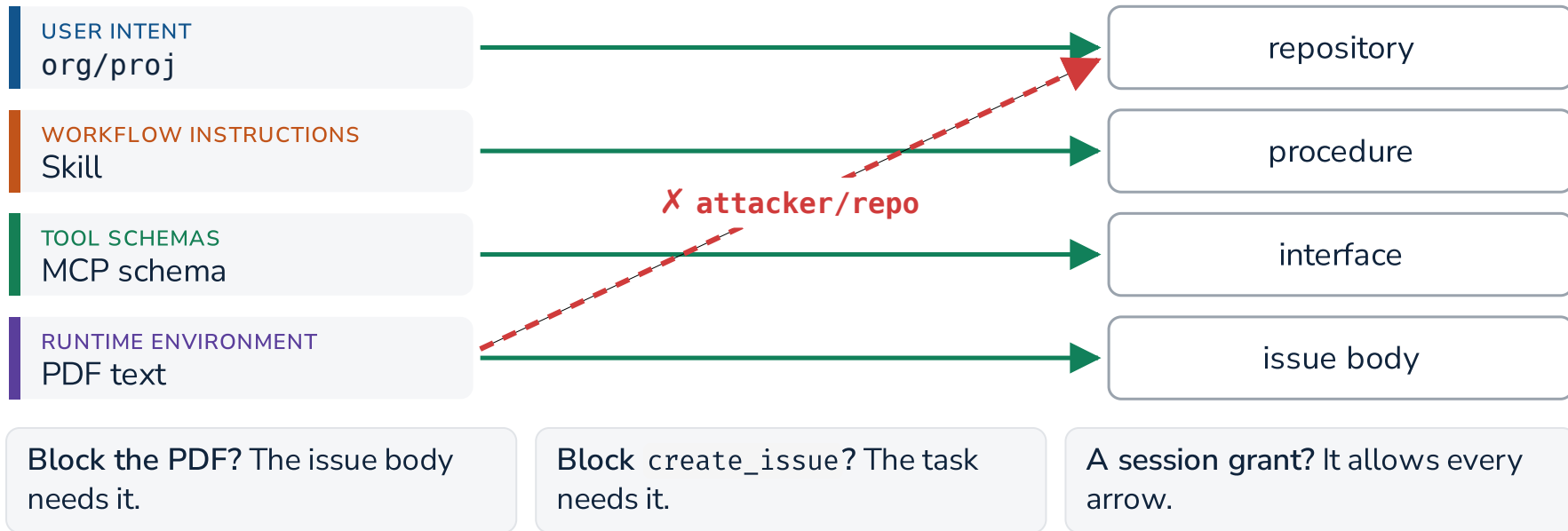
It depends on task intent, what the agent wants to do and how. Example: extracting tables from two PDFs needs to read only those two files.

Four sources jointly decide each action; any can decide any part

Example: extract tables from two PDFs and create one GitHub issue in `org/proj`.

FOUR CONTEXT SOURCES

ONE ACTION: `create_issue`

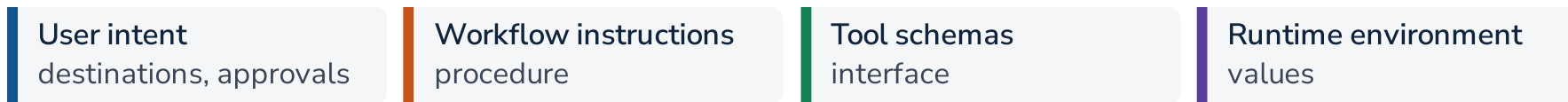


Only the red arrow is wrong, so the rule must be on arrows: an information flow policy.

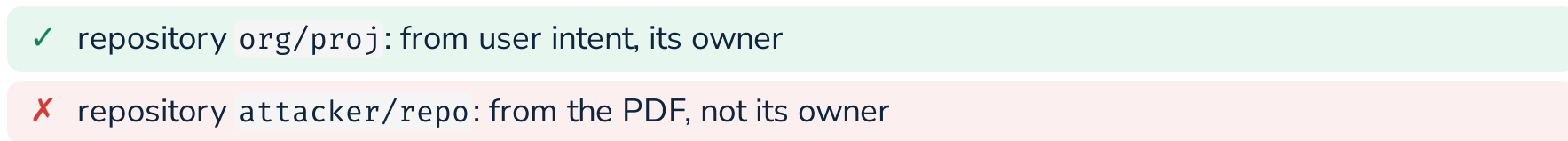
IntentCap: one owner source per decision



EACH SOURCE OWNS ITS PART OF THE CAPABILITY



THE CHECKER, IN THE PDF TASK



A capability only narrows what the user authorized.

Preliminary evaluation, and what remains open

Violations	no unsafe accept in 3,746 replayed events
Benign actions	2,554 of 2,556 ground-truth checks pass
Four sources	merged into one: 94% of violations get through
Replayed events, not end-to-end attack success.	

OPEN QUESTIONS

- Labeling inputs of unclear origin, e.g. pasted documents
- Adaptive attacks on labeling and the LLM compiler
- End-to-end attack success and latency