# Sequential subagents: fresh context, bounded work & handoffs

Key takeaways

1. Fresh context buys independence only when evidence and role are bounded.
2. Delegate four fields: objective, output, tools, and boundaries.
3. Hand off durable artifacts between roles, not retold chat transcripts.

# Common misunderstandings of sequential subagents

**Which claim would you defend right now? Choose before the evidence.**

"Fresh context improves quality even when the new agent repeats the same role."

"A role label such as 'reviewer' is enough to define a useful delegation."

"Chat summaries are the safest way to hand state to the next agent."

Keep your choice. Look for the mechanism or counterexample that would make it fail.

# You already run one sequential handoff

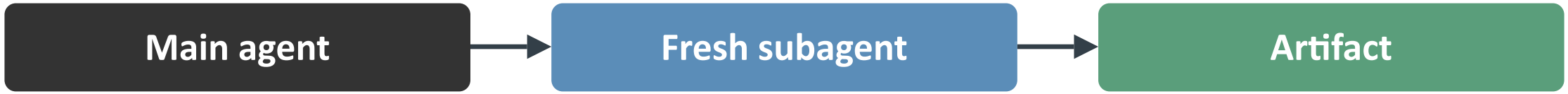
Week 2’s cross-model review is the two-stage case; today you design the pipeline

| Cross-model review (Week 2) | Sequential subagents (today) |
|---|---|
| The other model challenges one finished change | Any fresh context takes any bounded role |
| The driver assembles the diff snapshot for you | You choose the evidence each stage receives |
| Findings arrive in a fixed protocol | You define each handoff artifact and its acceptance check |
| Three rounds, then the protocol escalates | You set the budget and stop rule in every dispatch |

Same mechanism — separated context and bounded evidence — now under your design control.

# The subagent gets evidence, not history

A second agent is valuable when separation changes the judgment




Builds the change with full project context.

Its history includes every compromise it made.

Receives a bounded question and selected evidence.

No need to defend the first agent's reasoning.

Returns findings that survive the conversation.

A file, patch, test, or cited report.

Sequential does not mean redundant: each agent should change the evidence or the verdict.

# Concrete move: ask for an independent review

The same request, asked two ways, produces different evidence

**Same context: self-review**

```
# The builder sees its own explanation.
"Review the change you just made.
 Did you miss anything?"

# Prior choices remain in context;
# agreement is the easy continuation.
```

**Fresh context: artifact review**

```
git diff --staged > /tmp/change.diff
claude -p "$(cat REVIEW_PROMPT.md)
  Read /tmp/change.diff.
  Return only actionable findings."

# The diff, not the builder's story,
# is the reviewer's starting point.
```

The new context is not automatically smarter. It is useful because it is independent: no sunk cost, no prior approval, and no need to preserve a narrative.

source: anthropic.com/engineering/multi-agent-research-system

# Two dispatch surfaces in Claude Code

In-session subagents isolate; headless runs compose into pipelines

**In-session: spawn a subagent**

```
"Launch a subagent to find every
 caller of parse_mbox() and return
 file:line plus one-line context."

# Fresh context: your transcript
#   stays out of the subagent.
# Its tool calls stay out of yours;
#   only the final report returns.
# Named roles: .claude/agents/*.md
```

**Headless: one-shot claude -p**

```
claude -p "$(cat tasks/research.md)" \
  > .handoffs/evidence.md

# Composes with shell, make, cron.
# Artifacts land on disk, where a
#   script can check them.
# Codex and Gemini one-shot modes
#   fill the same pipeline role.
```

Both give the worker a fresh, bounded context. Use in-session subagents while you steer interactively; use headless runs when code owns the control flow.

# Fresh context buys three kinds of separation

Judgment, attention, and role are distinct reasons to isolate

## Judgment

The reviewer did not choose the implementation, so criticism does not contradict its own work.

**Use for design review, security review, and random challenges.**

## Attention

Only the task-relevant files and constraints enter the context; old tool logs stop competing.

**Use when the main session is long or a question is narrow.**

## Role

A researcher searches; a builder edits; a reviewer attacks assumptions. The prompt names the lens.

**Use when a specialized output is easy to verify.**

Isolation is an information design choice, not a magic quality multiplier.

source: anthropic.com/engineering/multi-agent-research-system

# A delegation prompt is a four-field contract

Objective, output, tools, boundaries — including a budget and stop rule

**Objective • output • tools • boundaries**

```
objective: Find the cause of the stale schedule PDF.
output: |
  Return: cause, file:line evidence, and one minimal fix.
  Do not edit files.
tools:
  - rg and file reads inside web/ and slides/
  - make -n for dependency inspection
boundaries:
  - Do not search the entire home directory.
  - Do not redesign the build system.
  - Stop after one falsifiable cause or 12 minutes.
```

The contract tells the subagent what success looks like and what it must leave alone. A role name alone (“researcher”) supplies neither.

source: anthropic.com/engineering/multi-agent-research-system

# Specificity divides labor; verbosity does not

Anthropic’s production prompt names all four contract fields

**Vague delegation**

```
"Research subagents and tell me
 what matters. Be thorough."

# Which question?
# Which sources?
# What deliverable?
# When should it stop?
```

**Bounded delegation**

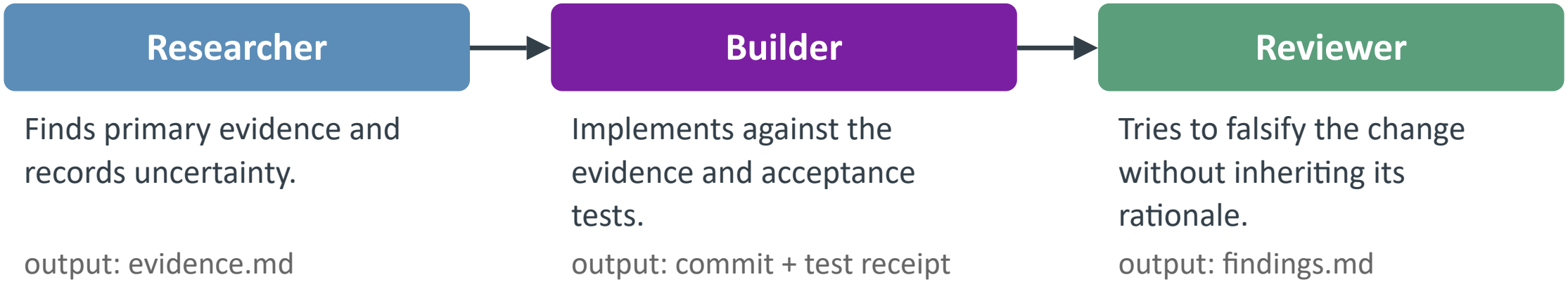
```
"From official Anthropic sources,
 find measured costs of agent teams.
 Return <= 5 claims, each with URL
 and exact supporting passage.
 No product survey; stop at 3 sources.
```

Anthropic reports that vague subagent requests produced duplicated searches and gaps; its production prompt names all four contract fields.

source: anthropic.com/engineering/multi-agent-research-system

# The useful sequence changes the kind of work

Researcher, builder, reviewer change the task, not just the context




Do not spawn three builders with three opinions when the task needs one implementation.

# Pass artifacts, not retellings

Files survive the conversation; a paraphrase decays at every hop

## Commands pass durable artifacts by path

```
claude -p "$(cat research.md)" > .handoffs/evidence.md
claude -p "Read evidence.md; implement SPEC.md."
git diff HEAD^ > .handoffs/change.diff
claude -p "Review change.diff; write findings.md.
```


**.handoffs/evidence.md → each model reads the original artifact**

**Pass file paths instead of copying a report through another model: this preserves fidelity and avoids paying to paraphrase it.**

source: anthropic.com/engineering/multi-agent-research-system

# A handoff artifact needs its own acceptance test

A receipt lets the consumer reject bad work before paying for more

**Machine-checkable receipt + human-readable artifact**

```
{
  "status": "complete",
  "claim": "Make omits the shared renderer dependency",
  "evidence": ["slides/Makefile:18", "review_deck_renderer.py:1"],
  "commands_run": ["make -n 085_orchestration_patterns-slides.pdf"],
  "open_questions": [],
  "artifact": ".handoffs/dependency-analysis.md"
}
```

- The consumer can reject a handoff missing evidence before doing more model work.
- A partial result is valid when status and open questions say exactly what remains.

# Match model and effort to uncertainty

The dispatch names the tier, the budget, and the partial-result stop rule

| Subtask | Model / effort | Budget and stop rule |
|---|---|---|
| Locate files, collect metrics | Fast model / low | 5–10 min; return paths + commands |
| Implement a bounded change | Coding model / medium | One coherent patch + tests |
| Challenge architecture | Strong reasoning / high | One verdict + top 3 risks |
| Mechanical formatting | Cheapest reliable / low | Fixed file set; zero redesign |

Put model, effort, budget, and the partial-result stop rule in the dispatch—not in hope.

# Instapoll

Instapoll · Review · Oct 15 A

Which task most benefits from a fresh sequential reviewer after an implementation agent finishes?

- A. Challenging assumptions in a bounded diff with explicit acceptance criteria
- B. Continuing the same edit while preserving every implementation rationale
- C. Repeating a deterministic formatter that already passed
- D. Copying the implementation agent’s final response into a report

# When does a second agent earn its tokens?

Delegate when an independent find is likely and a miss is expensive

**A decision rule, not a ritual**

```
expected_value = probability_of_independent_find * cost_of_miss
delegate = expected_value > review_cost

# Raise probability_of_independent_find with:
#   fresh evidence, a distinct role, and a falsifiable output.
# If the second agent sees the same story and does the same work,
# its probability of an independent find approaches zero.
```

- Worth it: security/design review, unfamiliar API research, random challenge, expensive mistake.
- Usually not: obvious one-line edit, deterministic formatter, duplicated implementation pass.

The second agent must add independence, specialization, or fresh capacity.

# You own the delegation contract

| Your Responsibility | Agent's Responsibility |
| --- | --- |
| Decide whether an independent verdict is worth its cost | Work inside the bounded role with a fresh context |
| Define the objective, acceptable output, and hard boundaries | Record commands, evidence, uncertainty, and unfinished work |
| Accept, reject, or reconcile conflicting artifacts | Return a durable artifact that the next role can verify |

Shared: The contract is the interface: both sides can detect when the handoff is incomplete

# Synthesis — separate judgment, preserve evidence

1. Fresh context earns an independent judgment only when the evidence and role are intentionally bounded.
2. Delegate with four fields: objective, output, tools, and boundaries—including a budget and stop rule.
3. Move durable artifacts between roles; use chat only to point at them and resolve decisions.

# write a delegation contract

10 min

- Before adopting a PDF library, your team must learn whether it preserves merged-cell boundaries and source-page coordinates.
- A research agent may read docs and run read-only examples. It may not edit the project or install persistent tools; it stops after 12 minutes.
- Write the contract: objective, output, allowed tools, and stop rule. Choose a model/effort level and one runnable acceptance check.
- Swap with a partner, mark one ambiguity or unbounded instruction, and revise. Deliverable: the revised contract and acceptance check.