AI MADE EZ · LEVEL 5 SUMMARY

Design AI systems

8 lessons · about 56 minutes in all · free

01 Set menu or free chef? **02** Build a standard outlet **03** The kitchen system **04** A brigade of chefs
05 Double shifts **06** Locks and keys **07** Head-office checks **08** Draw the floor plan

01 Set menu or free chef?

- A set menu is a workflow.
- A free chef is an agent.
- Every cook is an augmented LLM.

Augmented LLM = one call plus tools, search, memory
Prompt chaining = steps in a row, each uses the last
Routing = sort first, send to a specialist
Parallelization = at once, then combine or vote
Orchestrator-workers = a lead decides the subtasks at run time

Say it: “We pick the simplest pattern that works. Agents only when steps can’t be planned.”

02 Build a standard outlet.

- An MCP server offers three things.
- The wiring is called the transport.
- Labels are how Claude picks a tool.

Server offers tools, resources, prompts
Transport = the wiring
stdio = local, the app starts it
Streamable HTTP = a web server many reach
OAuth = sign in once; the app gets a limited key

Say it: “An MCP server is a standard outlet: few clear tools, the right wiring, small scopes.”

03 The kitchen system.

- The harness is everything around the model.
- Build it, or rent it.
- Tools run in a sandbox.

Harness = loop, tools, permissions, logs around the model
Agent SDK = Claude Code’s harness in your app
Managed Agents = Anthropic hosts it (beta)
Session = one run, in its own sandbox
Sandbox = walled-off computer where tools run

Say it: “The model decides. The harness runs tools in a sandbox, asks first and caps spend.”

04 A brigade of chefs.

- A brigade is a kitchen’s crew, run by a lead agent.
- Clean counters are the point.
- A brigade eats tokens.

Multi-agent system = a lead plus helper agents
Lead agent = plans, hands out, checks, combines
Delegation = the handoff, with a clear brief
Context isolation = each helper has its own counter
Synthesis = combining reports into one answer

Say it: “A brigade pays off only when the work splits into pieces the lead can check.”

06 Locks and keys.

- Least privilege: only the keys the job needs.
- Anything Claude reads can carry orders.
- Lock the doors that can't be undone.

Least privilege = only the keys the job needs
Untrusted content = anything Claude reads can carry orders
Lethal trifecta = data + untrusted content + a way out; cut one
Network allowlist = only listed hosts
Vault = secrets the sandbox never sees

Say it: "Fewest keys, everything it reads is untrusted, and a gate on what can't be undone."

05 Double shifts.

- Context engineering is keeping the counter clean.
- A crowded counter blurs.
- Long jobs need notes.

Context engineering = smallest set of useful tokens
Context rot = recall slips as context grows
Compaction = summarize the old shift
Context editing = clear old tool results
Memory tool = notes in a folder your app stores

Say it: "We clear old results, compact near the limit, and leave notes for the next shift."

08 Draw the floor plan.

- Sketching is done. Now defend it.
- Every choice is a trade-off.
- Plan for the bad night.

Design doc = the plan you defend; job and test first
Trade-off = what you gave up, written down
Cost model = tokens per task × price × volume
Failure mode = one way it goes wrong
Blast radius = how far damage spreads

Say it: "Here's the plan: the job, its test, the cost math, what can fail, and who approves."

07 Head-office checks.

- You can't inspect what you didn't record.
- Audit logs answer "who changed that?"
- A person stays on the floor.

Observability = you can see what happened
Trace = the step-by-step record of one run
Audit log = who did what, when
Human oversight = someone reviews, and can switch it off
RSP = Anthropic's voluntary safety rulebook

Say it: "We trace every run, log who did what, and someone can switch it off."

Prompts to try

Change the parts in [brackets] to fit you. One a day is plenty.

01 · Set menu or free chef?

ORDER #001 copy · paste · send

```
I’m planning an AI project: answer
customer emails. What matters most:
low cost. I don’t code.
1. Which fits best: one Claude call,
a workflow (prompt chaining,
routing, parallelization,
orchestrator-workers or evaluator-
optimizer) or an agent? Why, in 3
lines.
2. The steps, as a short numbered
list.
3. The simpler option you rejected,
and what would make us switch to an
agent.
4. One way it could fail, and how
we’d notice.
```

02 · Build a standard outlet.

ORDER #002 copy · paste · send

```
Design an MCP server for our booking
system. Who plugs in: just my team.
I don’t code; a developer builds it.
1. MCP server, or a plain tool in
one app? Why?
2. A few tools (namespaced name,
job, read or write), plus resources
and prompts.
3. Transport (stdio or Streamable
HTTP) and sign-in (OAuth scopes),
and why.
4. Two tests before anyone uses it.
Unsure a feature exists? Say so;
name the page
(modelcontextprotocol.io).
```

03 · The kitchen system.

ORDER #003 copy · paste · send

```
Plan the agent harness for: a weekly
competitor report. Hosting:
Anthropic hosts it (Managed Agents).
I don’t code.
1. Is that hosting right for this
job, or would a scheduled workflow
do? Why?
2. Tools it needs, which run alone,
which wait for a yes, which built-
ins go off.
3. What the sandbox can and can’t
reach.
4. A spend cap and a quick stop.
Unsure a feature exists? Say so;
name the page
(platform.claude.com/docs).
```

04 · A brigade of chefs.

ORDER #004 copy · paste · send

```
Job: research 12 competitors.
Proposed team: a lead + 3 helpers. I
don’t code.
1. Does it split into independent
pieces? Why or why not?
2. If yes: each helper’s brief,
tools, model and output.
3. Rough cost, 1 agent vs this team
(Anthropic: agents use about 4x a
chat’s tokens, multi-agent about
15x).
4. How the lead catches a wrong
helper.
Recommend the smallest team that
works. Unsure of a number? Say so.
```

05 · Double shifts.

06 · Locks and keys.

ORDER #005 copy · paste · send

```
An AI agent for a long job: move 500
help articles to a new site, running
a few hours. I don't code. Write a
context plan:
1. What it loads first, what it
fetches later.
2. When to compact or clear old tool
results; what must never be lost.
3. Its note files, with a short
template each.
4. How a fresh session picks up.
Mark API features vs. good habits.
Unsure? Say so.
```

ORDER #006 copy · paste · send

```
Security review for an inbox
assistant. It touches: Reads email,
Can send email. I don't code.
1. Lethal trifecta (private data,
untrusted content, a way out): which
leg can we cut?
2. Least privilege: only tools and
sites it needs.
3. Where secrets live, out of its
sight.
4. What waits for a person.
5. What we log; how we stop it fast.
For each: the lock, why, owner.
Unsure a setting exists? Say so.
```

07 · Head-office checks.

ORDER #007 copy · paste · send

```
Governance plan for a customer
support agent. Sign-off: our team
lead. I don't code.
1. Traces: what each run records;
who sees it.
2. Audit logs: which actions we
keep.
3. Oversight: who reviews samples,
how often, who can stop it.
4. Private data: what it sees; which
plan fits; does it train on chats?
5. Evals: real cases re-tested after
each change.
6. An 8-item monthly checklist.
Flag legal questions. Unsure? Say
so.
```

08 · Draw the floor plan.

ORDER #008 copy · paste · send

```
Use case: answer customer emails.
Hardest limit: tight budget. I don't
code.
Ask up to 3 questions first; label
guesses. Then a design doc:
1. The job and its test.
2. Pattern, the simpler one
rejected, trade-offs.
3. Cost: per task × volume, with and
without a batch, math shown.
4. Top 3 failures, blast radius, how
we'd notice.
5. Who approves what.
6. A 6-week pilot: which team,
weekly measures, the go/no-go rule.
```

> You can draw the plan for a real AI system and defend every choice.

Made by Alexandra Vargas · not affiliated with Anthropic