Google - ADK

file structure

- project_dir
  - agent_dir
    - __init__.py
    - .env
    - agent.py

① Imports — os, dotenv

② google-ADK imports

```
google.adk.agents → Agent
google.adk.runners → InMemoryRunner
google.adk.tools → google_search
google.genai → types
```

③ Initialize root_agent

```
root_agent = Agent(
        name = "__",
        model = "gemini-__"
        description = "__",
        instruction = "__",
        tools = [ ],
)
```

④ Initialize runner (optional)

```
runner = InMemoryRunner(agent = root_agent)
```

⑤ Way to run this adk project

```
adk web
adk run project_dir
```

Day - 1b (Agent Architectures) multi_agent_system

1. Imports for api_key setup
2. Setup api_key & Vertex ai
3. google-adk import

Section-01

- SequentialAgent
- ParallelAgent
- LoopAgent
- Agent

google.adk.agents

AgentTool
FunctionTool
google_search

→ google.adk.tools

Section-2

1) Why multi-agent systems?

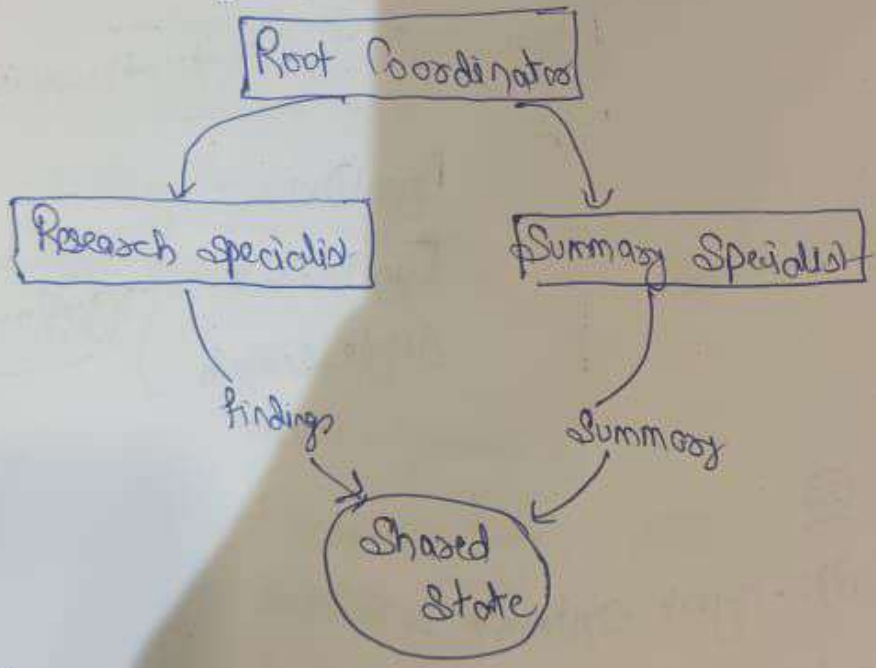
- Here we try to build a team of simple, specialized agents that collaborate, just like a real-world team.
- Each agent has a one clear job
  eg- one agent → for research
  another agent → only writes
- easier to build, test, maintain, reliable

④ Architecture

a) Single agent -

One Agent does Everything X Monolithic Agent

b) Multi-Agent team:



④ Code

- Create (research_agent) & (summarizer_agent) structure will be same as root_agent
- Create root_agent

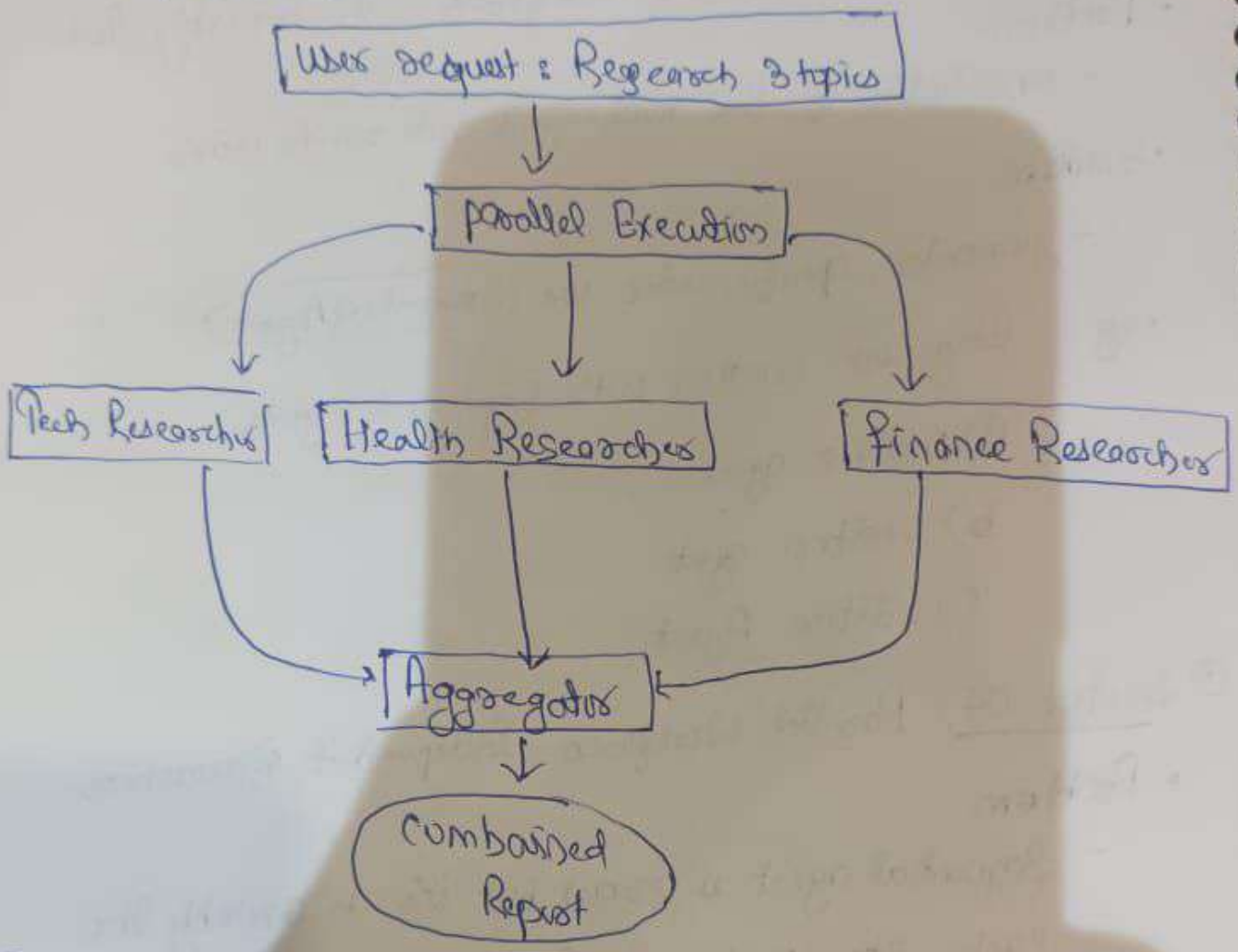
⑧ Section-03 : Sequential workflows - the assembly line

- Problem-
  - unreliable, if the multi-agent will run in order
- Solution-
  - guarantee, specific order, use (Sequential Agent)
- eg - Blog post creation with sequential agents
  a) outline agent
  b) writer agent
  c) Editor Agent

⑨ Section-04 : Parallel Workflows - Independent Researchers

- Problem
  - Sequential agent is great but it's an assembly line each step must wait for the prev one to finish
  - What if we have several tasks that are not dependent on each other?
- Solution
  - If tasks are not dependent on eachother, then use (parallel Agent)

⊙ Architecture



① Section-5 : Loop workflows - The refinement cycle

- Problem
  - One Shot Quality
    runs → start → to → finish (in single run)

◉ Architecture: Story writing & critique loop