AI Agents for Fun & Profit: Hands-On Workshop

From Zero to Deployed in One Day 🙏🏽

Date & Time: 31st July, 09:30

Location: **Space4**

Who Am I?

Stanley Mwangi

- Pioneer in generative AI → published the world's first AI-human poetry anthology using ChatGPT
  - Poetic Genesis: An Anthology of 52 AI Crafted Poems
- Former junior ML engineer at Syd
- Co-founder of AceTheRound
  - AI interview coaching for finance students

Opening Engagement

- *Icebreakers*
  - "What's the last thing you asked an LLM to do for you?"
  - "If you could automate one 30-minute task today, what would it be?"
- *Goal:* Surface everyday pain-points that agents could tackle.

Workshop Outcomes

By the end you will:

1. *Build* a LangGraph ReAct agent with memory
2. *Integrate* tools e.g. Exa search for more dependable search
3. *Deploy* the agent + Streamlit UI to Render via Docker Compose
4. *Refactor* the codebase to solve a business problem of your choice (if time allows)

Agenda

From	To	Description	Modality
9:30	09:45	Quick networking and informal discussions	Plenary
09:45	10:15	*Introduction:* The Agentic Opportunity	Slides + Q&A
10:15	10:30	*Concepts*: Agent architecture & Think-Act-Observe	Slides + Q&A
10:30	11:30	*Lab 1*: Build a ReAct agent with a search tool & memory	Pair-programming
11:30	11:45	Break	N/A
11:45	12:45	*Lab 2*: Add more tools	Pair-programming
12:45	13:45	Lunch	N/A
13:45	14:30	*Lab 3:* Containerise with Docker + deploy on render	Pair-programming
14:30	15:15	Lab 4: Choose your own biz-problem + architect approach	Pair-programming
15:15	15:30	Break	N/A
15:30	16:45	Lab 5: Solve your biz-problem	Mini-exercise
16:45	17:15	Final presentations	Reflection
17:15	17:30	Wrap-up & resource pack	Discussion

The Agentic Opportunity

Why Agents Change Everything

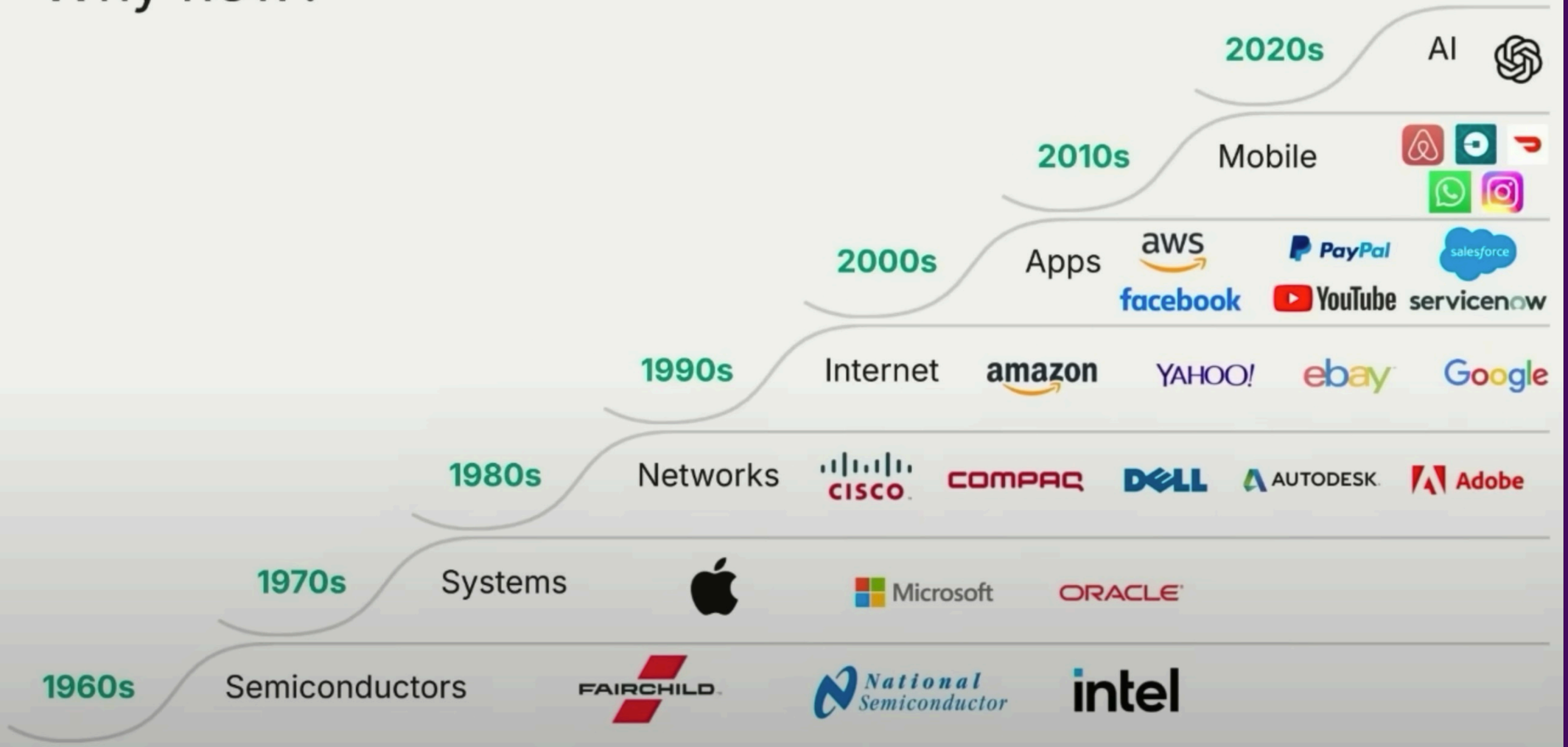
From Assistants → Agents → Agent Economy

Era	*Capability*	*Example*
Assistants	Single-prompt helpers	ChatGPT Feb 2023
Agents	ReAct cycles, tool use	Today's LangGraph labs
Agent Swarms	Multi-agent collaboration	MCP / inter-LLM calls
Agent Economy	Persistent, transacting actors	Vertical AI firms (Harvey, Abridge)

So What? — $ Trillions in Play

- AI services *start* at a base > $400 B—already larger than software was pre-cloud.
- Both *software and services* profit pools are under attack; agents march from **tool → co-pilot → autopilot → outcome-as-a-service**.
- Waves are *additive & faster*: compute, data, talent, networks are all in place.

Why now?


2020s
AI
2010s
Mobile
2000s
Apps
aws
PayPal
salesforce
facebook
YouTube
servicenow
1990s
Internet
amazon
YAHOO!
ebay
Google
1980s
Networks
CISCO
COMPAQ
DELL
AUTODESK
Adobe
1970s
Systems
Microsoft
ORACLE
1960s
Semiconductors
FAIRCHILD
National Semiconductor
intel

Why Now? — Distribution Physics Are Zero-Friction

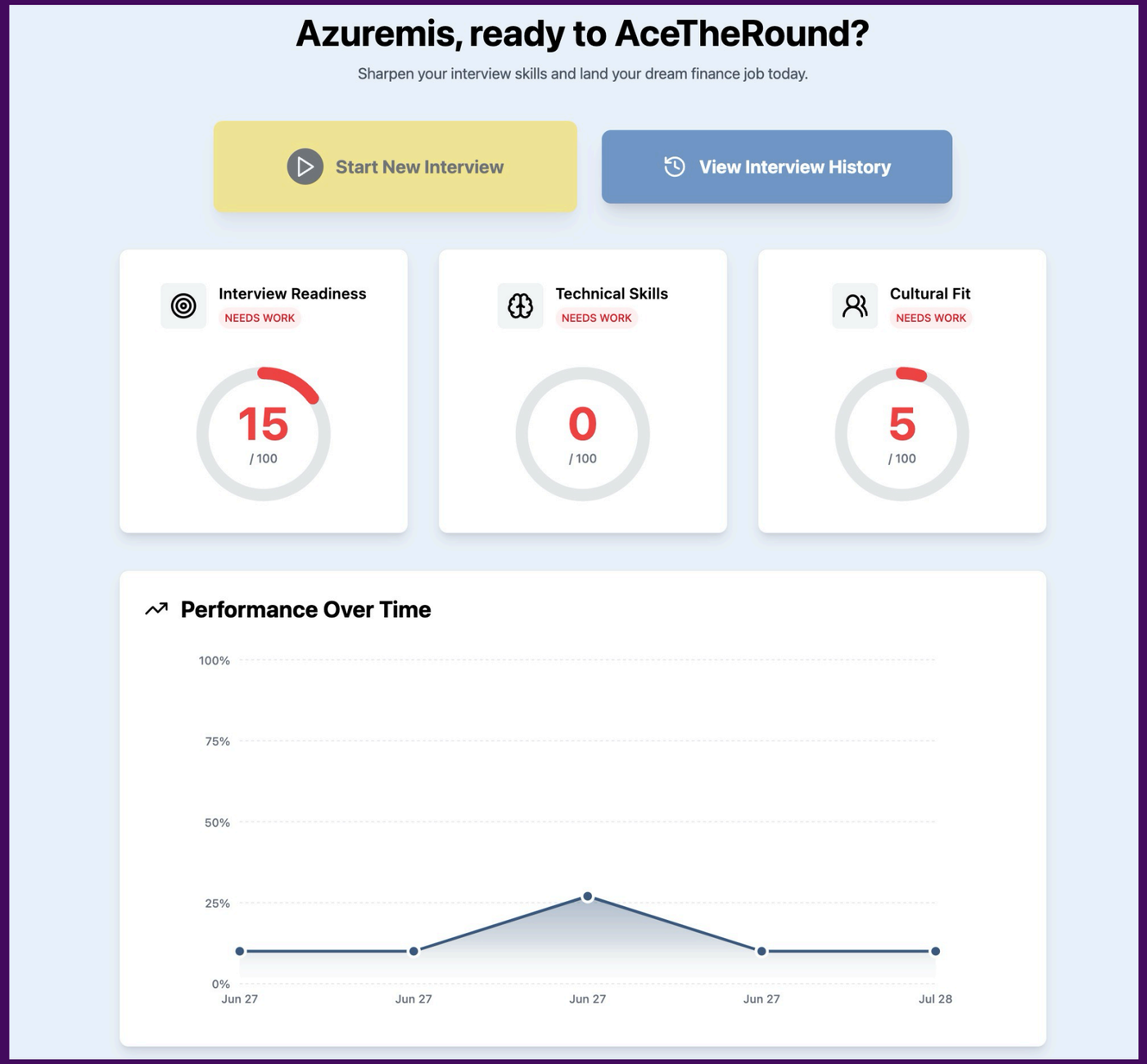
1. *Attention:* 1.8 B users on Reddit + X spread AI memes overnight.
2. *Desire:* Everyone has a personal "ChatGPT moment"; demand is **pull**, not **push**.
3. *Access:* 5.6 B people online; APIs one-click deploy.

Rails are laid, the starting gun fired—there are no barriers to adoption.

Where Will Value Accrue?

The Application / Outcome Layer

- Foundation models *reach down* into shallow apps e.g. summarisation, rewriting.
- Start-ups win by going *customer-back*:
  - Vertical-specific workflows e.g. finance interview tool
  - Function-specific "painkiller" agents e.g. hard to get interview practice
  - Human-in-the-loop complexity
- *Data flywheels* must tie to **business** metrics—not vanity usage.


Azuremis, ready to AceTheRound?
Sharpen your interview skills and land your dream finance job today.
Start New Interview
View Interview History
Interview Readiness
NEEDS WORK
15
/ 100
Technical Skills
NEEDS WORK
0
/ 100
Cultural Fit
NEEDS WORK
5
/ 100
Performance Over Time
100%
75%
50%
25%
0%
Jun 27
Jun 27
Jun 27
Jun 27
Jul 28

Winning Moats Across the Value Chain

1. *Opinionated UX* – ship the **answer**, not a raw GPT prompt.
2. *Trust & Security* – compliance, guardrails, vibe > feature list.
3. *Proprietary Data Flywheel* – usage data improves model, improves result, improves lock-in.

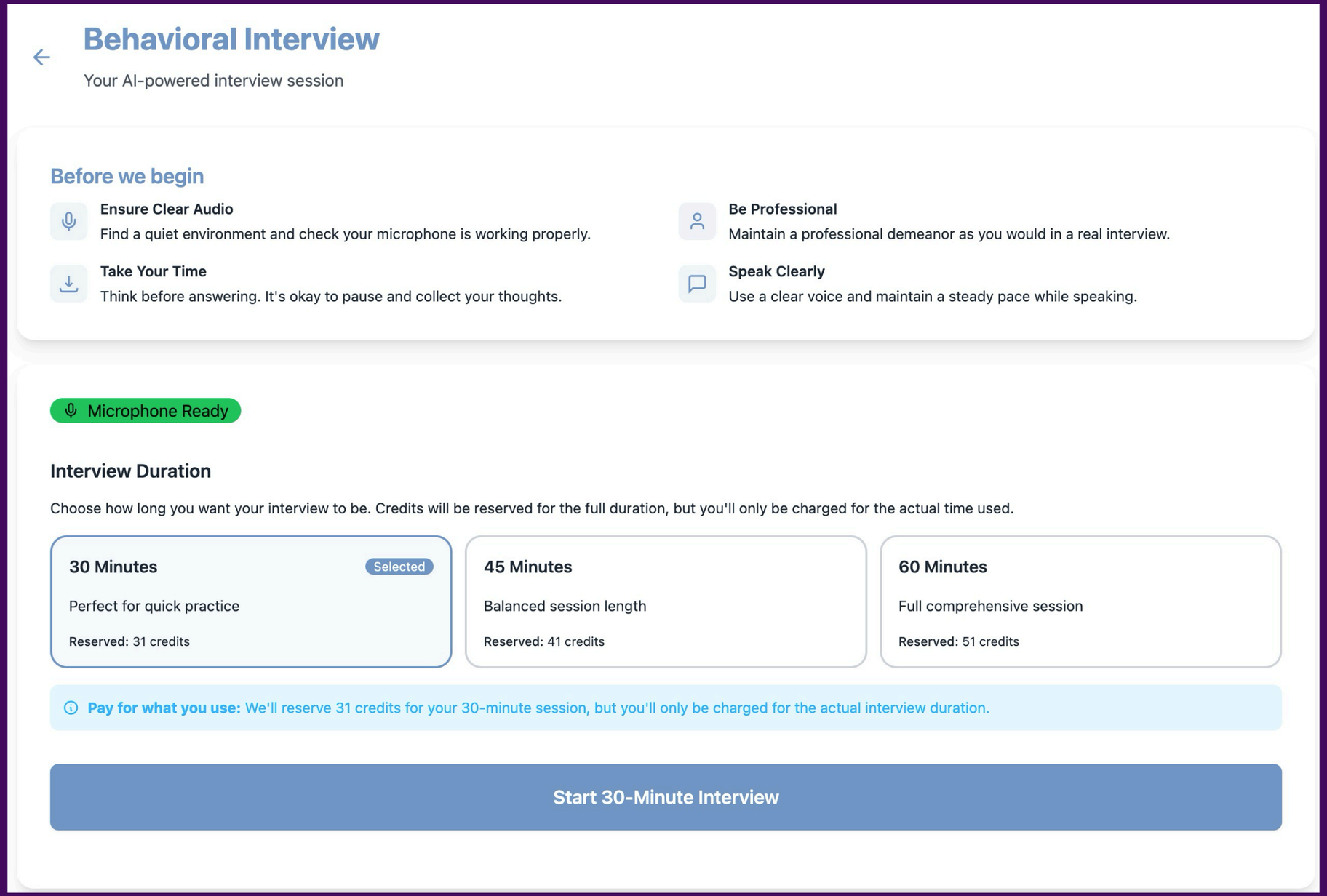
Tech Gaps to an Agent Economy

Challenge	*Why It Matters*
Persistent Identity	Agent must remember itself **and** you.
Open Protocols	TCP/IP moment for agent-to-agent value transfer (e.g. MCP).
Robust Security	No face-to-face trust → new cottage industry.

Mindset Shifts for Builders

- *Stochastic Thinking:* Goodbye determinism, hello probability.
- *Management over Coding:* Orchestrate fleets of agents vs. writing every line.
- *Leverage > Certainty:* More output, less predictability—embrace risk management.

High leverage with lower certainty is the new normal.


Behavioral Interview
Your AI-powered interview session
Before we begin
Ensure Clear Audio
Find a quiet environment and check your microphone is working properly.
Be Professional
Maintain a professional demeanor as you would in a real interview.
Take Your Time
Think before answering. It's okay to pause and collect your thoughts.
Speak Clearly
Use a clear voice and maintain a steady pace while speaking.
Microphone Ready
Interview Duration
Choose how long you want your interview to be. Credits will be reserved for the full duration, but you'll only be charged for the actual time used.
30 Minutes
Selected
Perfect for quick practice
Reserved: 31 credits
45 Minutes
Balanced session length
Reserved: 41 credits
60 Minutes
Full comprehensive session
Reserved: 51 credits
Pay for what you use: We'll reserve 31 credits for your 30-minute session, but you'll only be charged for the actual interview duration.
Start 30-Minute Interview

Predictions for 2025

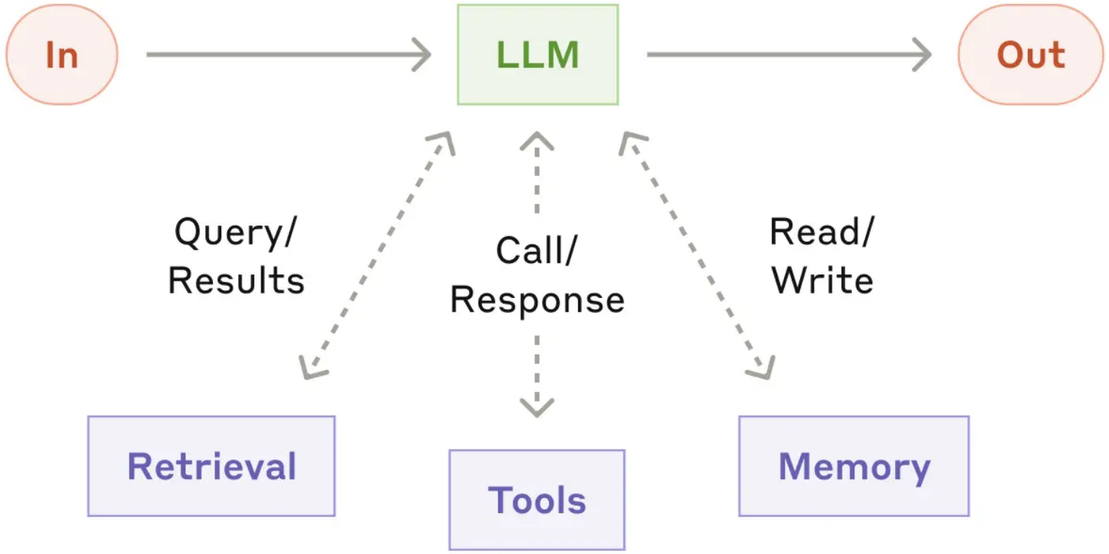
1. *Vertical agents outperform top humans* in security, DevOps, networking.
2. *Coding abundance* explodes—taste becomes the scarce asset.
3. *One-person unicorn* feasible as processes merge into agent networks.
4. *Run-Like-Heck Era:* Nature hates a vacuum; velocity beats perfection.

Call to Action

- *Build vertically-deep agents* that own outcomes.
- *Invest in trust, not just tokens.*
- *Optimize for speed*—the market suction is deafening.
- Let's create the agent economy together!

Questions, debates, wild predictions?

Agent Architecture



Layers & Responsibilities

Layer	*Purpose*	*Example Tech*
UI / UX	Capture intent, show progress	Streamlit, React
Controller	Routes requests to…	FastAPI, Flask
Agent Core	Plans, iterates (TAO loop)	LangGraph node
Tool Layer	Interfaces to reality	Exa Search, SQL, Bash
Memory / State	Stores context across turns	Redis, PGVector
Observability	Logs, traces, evals	Grafana, Honeycomb

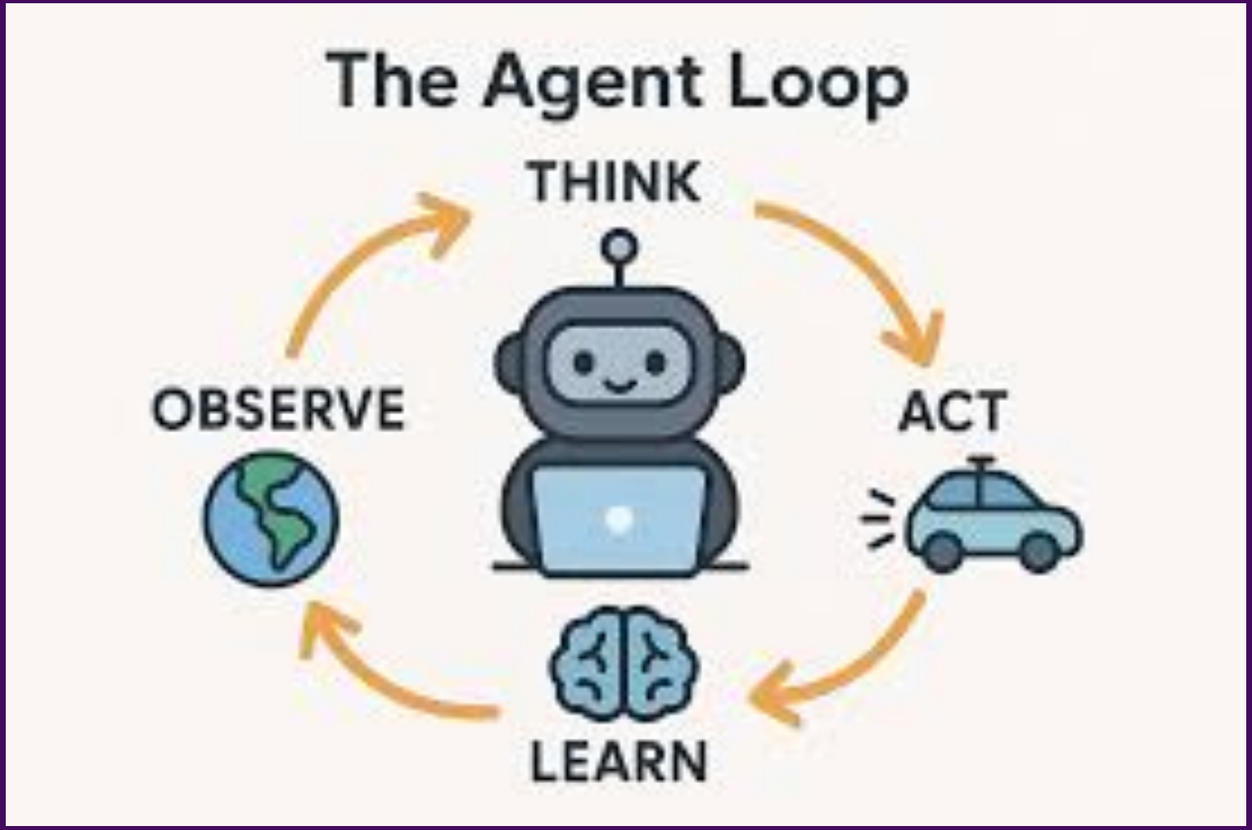
The Think → Act → Observe Loop (TAO)

I. *Think*

- Model reflects on goal & current state
- Produces _plan or next action_

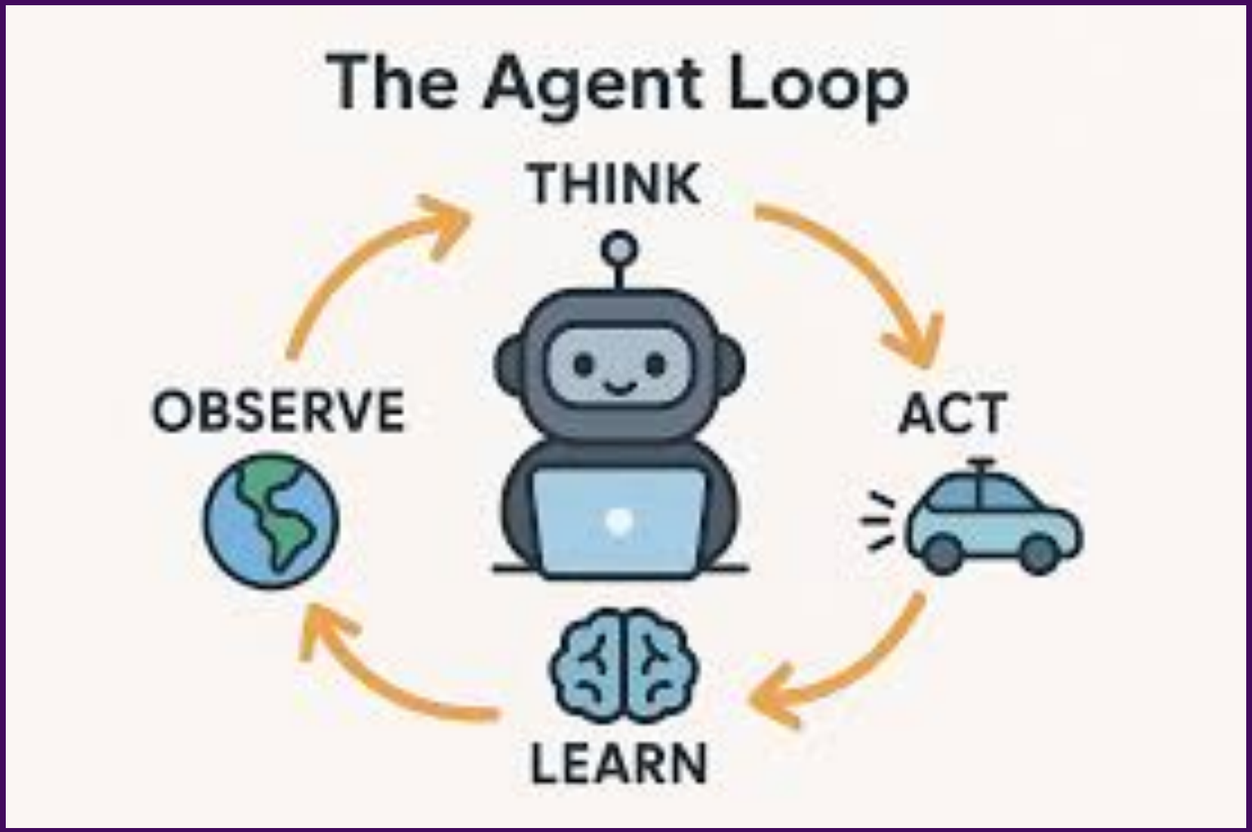
I. *Act*

- Executes tool, code, or sub-plan
- Emits side-effect (API call, file edit, etc.)



I. *Observe*

- Reads tool result / env feedback
- Appends to memory; loop repeats



Building Block: Augmented LLM

- *Retrieval-Augmented Generation (RAG)*
- *Tool Calling* (structured JSON)
- *Short-Term Memory* in context
- Optional *Long-Term Memory* (vector store)

Workflow ① – Prompt Chaining

Goal: Decompose task → sequential steps for higher accuracy.

```
Outline → Check Outline → Draft → Proof
```

- Programmable *gates* after each step
- Easy to debug; latency grows linearly

Use When: Steps are predictable & well-defined (marketing copy, doc drafting).

Workflow ② – Routing

Goal: Classify input → send to specialised pathway.

Route	*Downstream Prompt / Model*
Billing	“Refund” template
Tech Support	Claude 3.5 Sonnet
FAQ	Cached answer

Use When: Heterogeneous requests need different prompts or models.

Workflow ③ – Parallelization

Sectioning

Split input → parallel sub-tasks → merge.

Voting

Same task *N* times → choose consensus.

- Cuts latency; boosts reliability through diversity
- Requires deterministic merge logic

Use When: Tasks are independent or benefit from multiple perspectives

Workflow ④ – Orchestrator + Workers

Central *orchestrator LLM*:

- Plans dynamic sub-tasks
- Delegates to *worker LLMs/tools*
- Synthesises results → final answer

Use When: Unknown number or type of subtasks (code refactor, research synthesis).

Workflow ⑤ – Evaluator + Optimizer

Loop: *Generate → Evaluate → Revise* until criteria met or budget spent.

- Needs clear score rubric
- Costs ↗; quality ↗↗

Use When: Quality bar is non-negotiable (literary translation, policy drafting).

Agents vs. Workflows

Aspect	*Workflow*	*Agent*
Control Path	Pre-coded	Model-decided
Latency / Cost	Lower	Higher
Flexibility	Limited	High
Best For	Repeatable tasks	Open-ended goals

- *Rule of Thumb:* Start simple (single call → workflow).
  - Level-up to agents only when decision-making flexibility pays for itself.

Lab 1 – Build a Minimal ReAct Agent

Objectives

- Create a LangGraph agent
- Add conversation memory
- Test via CLI

Lab 1 – Build a Minimal ReAct Agent

Key Steps

1. Go to the workshop page
2. Install prerequisites
3. Go to the repo url
4. Copy the repository url

Lab 1 – Build a Minimal ReAct Agent

Key Steps

1. Run **git clone -b feat/lab1-complete repo-url**
2. Make a .env file in the route and put your LLM key
3. Get the agent to run
    - ReAct Agent langgraph documentation
4. Join the group chat
    - yapli.chat
    - type in sabpg2

Lab 2 - Tool Integration

New Powers

Tool	Purpose
Exa Search	Web intelligence
Math Add	Quick arithmetic

Code Diffs - agent.py

```
from tools.search import exa_search
tools = [add_function, exa_search]
agent = create_react_agent(llm, tools, memory, verbose=True)
```

Test Prompt

“Who won the women's Euros in 2025 and what was the final score?”

Lab 3 – Deploy to the Cloud

Running your docker container on local

```
docker compose down                    # stop current containers
docker compose --build                 # build container with new configuration
```

Running your docker container on local

Then visit in your browser:

- Frontend: http://localhost:8501
- API: http://localhost:8000

Lab 3 – Deploy to the Cloud

Stack

- *FastAPI* ➜ /chat, /health on :8000
- *Streamlit* ➜ UI on :8501
- *Docker Compose* ➜ api + ui services
- *Render.com* ➜ one-click deploy to the internet

Success Criteria

- Health check returns ✅
- Public URL chats without errors

Resources

- Installing prerequisites
  - Docker
  - Make
    - Already in Unix by default: Linux, MacOs
    - Windows requires install

Resources

- OpenRouter
  - Free models
- Exa Search
- Render

Wrap-Up & Q&A

Key Takeaways

- ReAct + LangGraph = quick, transparent reasoning
- Tools turn LLMs into **doers**, not just talkers
- Containerisation makes cloud moves trivial

Questions? Ideas you're excited to try?

Connect

- *GitHub:* github.com/azuremis
- *LinkedIn:* linkedin.com/in/azuremis
- *Email:* azuremismachine@gmail.com

Let’s keep shipping useful agents!

Thank You!

You now have everything to build, deploy, and potentially profit from AI agents. Go create something awesome!