AI & Agentic AI (AI 101)

A plain-language foundation — what these tools actually are, how they fit together, and how to choose.

CORTEXA CONSULTING

Intelligence, applied.

hello@cortexa-consulting.ai · www.cortexa-consulting.ai

WHAT WE'LL COVER

Four ideas, in order

What AI really is
01

A very well-read assistant with one talent: prediction.

What makes it agentic
02

The same brain, given hands, memory, and a goal.

The dependency stack
03

How hardware, models, and software hold each other up.

Key terms, plainly
04

The working vocabulary, each in one line.

A CORTEXA BRIEFING COMPANION

This reference (AI 101) accompanies **Cortexa Briefing** — Cortexa Consulting's plain-language orientation to applied AI for marketing and technology teams.

A brilliant intern who read almost everything

Picture an intern who has read nearly the entire internet — every book and manual — **but has no hands, no eyes on the world, and forgets everything when the chat ends.**

Its one real talent is **shockingly good guessing**: given some text, it predicts what should come next. Writing, coding, summarizing — all of it is that single trick at enormous scale.

THE ONE-SENTENCE VERSION

An AI model is a **prediction engine**. Everything it appears to "know how to do" is that prediction skill, applied.

your text

the model

predicts the next piece

Building the brain vs. using it

Training

the years of school

- Where the model learns, by finding patterns across enormous amounts of text.
- Slow, done once, and extraordinarily expensive — this is the part that needs warehouses of chips.
- You are almost never doing this yourself.

Inference

asking a question

- Where the trained model answers — the part you touch every time you use AI.
- Fast and comparatively cheap, but it happens millions of times a day, so cost adds up.
- This is what you pay for when you "run" a model.

Give the brain a body

A chatbot responds

A brain in a jar. You ask a question, it gives an answer, and the exchange is over. It cannot do anything in the world on its own.

An agent acts

Give that same brain hands **(tools)**, a notepad **(memory)**, and a goal instead of a question. Now "handle my inbox" becomes: read, decide, draft, send, check, and try again until it's done.

The loop that makes it "agentic"

1

Goal

Start with an outcome, not a single question.

2

Plan

Break the goal into steps.

3

Act

Use tools: search, code, send.

4

Check

Did that step actually work?

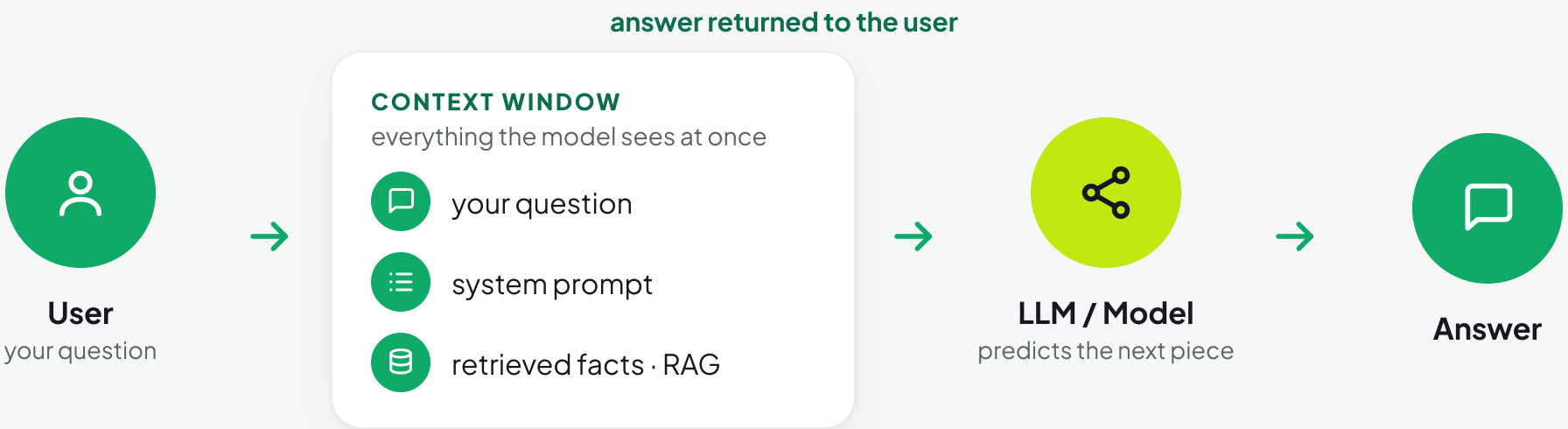
5

Adjust

Fix and repeat until the goal is met.

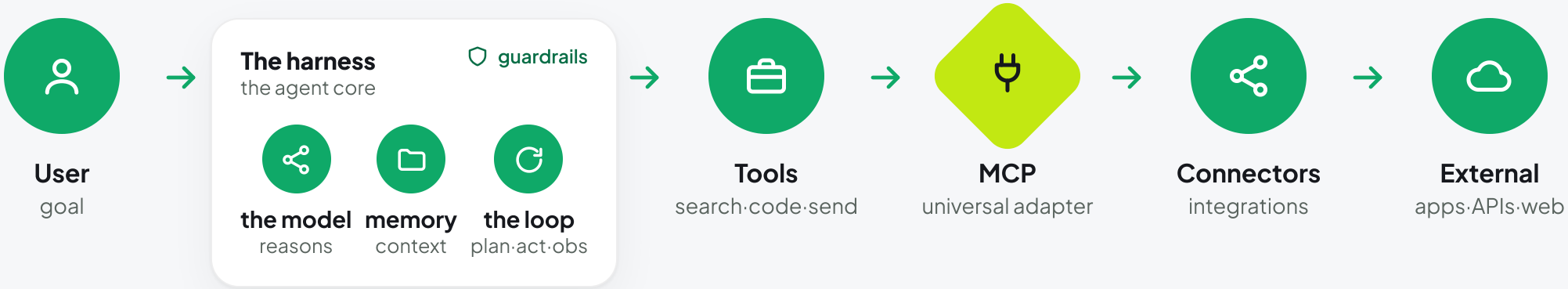
repeats on its own until the goal is met

Simple AI: one straight pass



Assemble the context — your question, the system prompt, and any facts pulled in by **RAG** — then the model predicts an answer. No tools, no actions, no loop.

Agentic AI: a loop that uses tools

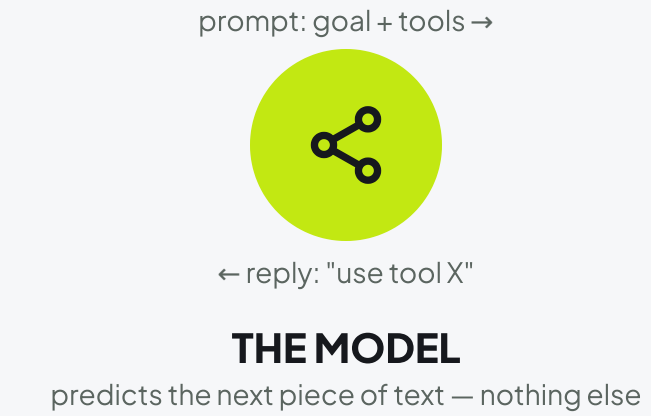


The model never touches your systems directly. It calls **tools**, standardized by **MCP**, which reach real apps and data through **connectors** — then results loop back until the goal is met.

The model thinks; the harness acts

THE HARNESS
the code wrapped around the model

- builds the prompt the model sees
- reads the model's reply
- runs the tools & connectors
- keeps memory across steps
- enforces guardrails
- retries and loops until done



On its own, a model only predicts text. The **harness** is the code around it that builds the prompt, runs the tools, keeps memory, and loops — turning predictions into actions.

The model is the engine; the harness is the rest of the car that turns raw power into getting somewhere.

IDEA 03 · HOW IT ALL HOLDS TOGETHER

The dependency stack

Each layer rests on the one below. Change the bottom and everything above moves.

Application — The chat app, agent, or copilot people actually touch.

Serving software — Runs the model efficiently and handles many requests.

Model — The brain. Its size is measured in parameters.

Hardware (GPUs) — The engines that do the actual math. Nothing thinks without them.

Runs through every layer

Your data — the ingredients the model reasons over.

Tools — the gadgets an agent uses to act (search, code, send).

Capability is bought with silicon.

IDEA 03 · THE STRATEGIC INSIGHT

The choices are chained together

A decision at one layer quietly commits you to the rest. Two common starting points show how the chain runs:

Optimize for capability

Reach for a top-tier **hosted** model

↓

…used as a service, over the internet

↓

…so your data is processed off your premises

↓

…and you pay per use, rising with volume

Optimize for control

Run a model on infrastructure **you govern**

↓

…which favors open models you can host yourself

↓

…with a larger up-front investment to stand up

↓

…and a modest trade-off in raw capability

There's no "best" — only the right-sized fit for the data, the budget, and the team.

Key terms, plainly

- **LLM**
  Large Language Model — the “intern” brain that predicts text.
- **Parameters**
  The model's adjustable knobs; more usually means more capable.
- **Context window**
  The model's desk space — how much it can see at once.
- **System prompt**
  The standing instructions that set the model's role and rules before you say anything.
- **Foundation model**
  The big, general-purpose base that others build on.
- **Tokens**
  The LEGO bricks of text models read and write in. You're billed per token.
- **Prompt**
  The instruction you give it.
- **Multimodal**
  Handles images, audio, or video too — not just text.

Key terms, plainly

- **API**
  The ordering window — how software talks to a model it doesn't own.
- **RAG**
  Handing it your files right before it answers, so it speaks from your data.
- **Vector database**
  Where those numbers live, so the right facts surface in an instant.
- **Latency**
  How long you wait for an answer.
- **Fine-tuning**
  A short specialty course so it's better at your specific job.
- **Embeddings**
  Turning text into numbers so a computer can find what's most relevant.
- **Temperature**
  The creativity dial — low for precise and repeatable, high for varied.
- **Cost per token**
  You pay per token in and out; agents loop and call the model many times.

Key terms, plainly

Agent
A model given tools, memory, and a goal — so it can act, not just answer.

Harness
The code around the model — runs the loop, calls tools, keeps memory, enforces guardrails.

Human-in-the-loop
A person reviews or approves the agent's work before it acts.

Quantization
Compressing a model to run on smaller, cheaper hardware.

MCP
A universal adapter for plugging tools and data into an agent.

Guardrails
Rules and limits that keep an AI from doing the wrong thing — off-limits topics or sign-off.

GPU / TPU
The specialized chips that do the heavy math.

Distillation
Training a smaller, cheaper model to mimic a larger one — most of the skill, less of the cost.

Key terms, plainly

- **Hallucination**
  When it confidently makes something up. The main reliability risk.
- **Prompt injection**
  A hidden instruction smuggled into content the AI reads, trying to hijack it — the key agent security risk.
- **Reasoning models**
  Models that work step-by-step before answering — better at hard problems, but slower.
- **Evals**
  How you test whether an AI system works — scoring answers against known-good results.
- **Data privacy**
  Whether your data is stored or used to train the vendor's model. Always confirm.
- **Grounding**
  Tying answers to real sources so they can be checked — the antidote to hallucination.
- **Jagged capability**
  AI can be brilliant at one task and fail at a trivial one — test before you trust.
- **Chain-of-thought**
  Having the model show its working, step by step, which improves hard answers.
- **Knowledge cutoff**
  The date training stops; it can't know newer events unless you give it search or files.

THE CORTEXA LENS

How we read all of this (Cortexa AI)

Augment, not replace

AI makes people faster and sharper. People stay at the forefront of every solution.

What's real, not hype

We separate genuine capability from marketing — grounded, evidence-led, every time.

Right-sized to the need

The best solution a team can actually own beats the biggest, flashiest one.

CORTEXA
CONSULTING

Intelligence, applied.

SEE CLEARLY. ACT INTELLIGENTLY. BUILD WHAT'S NEXT.

www.cortexa-consulting.ai · hello@cortexa-consulting.ai