

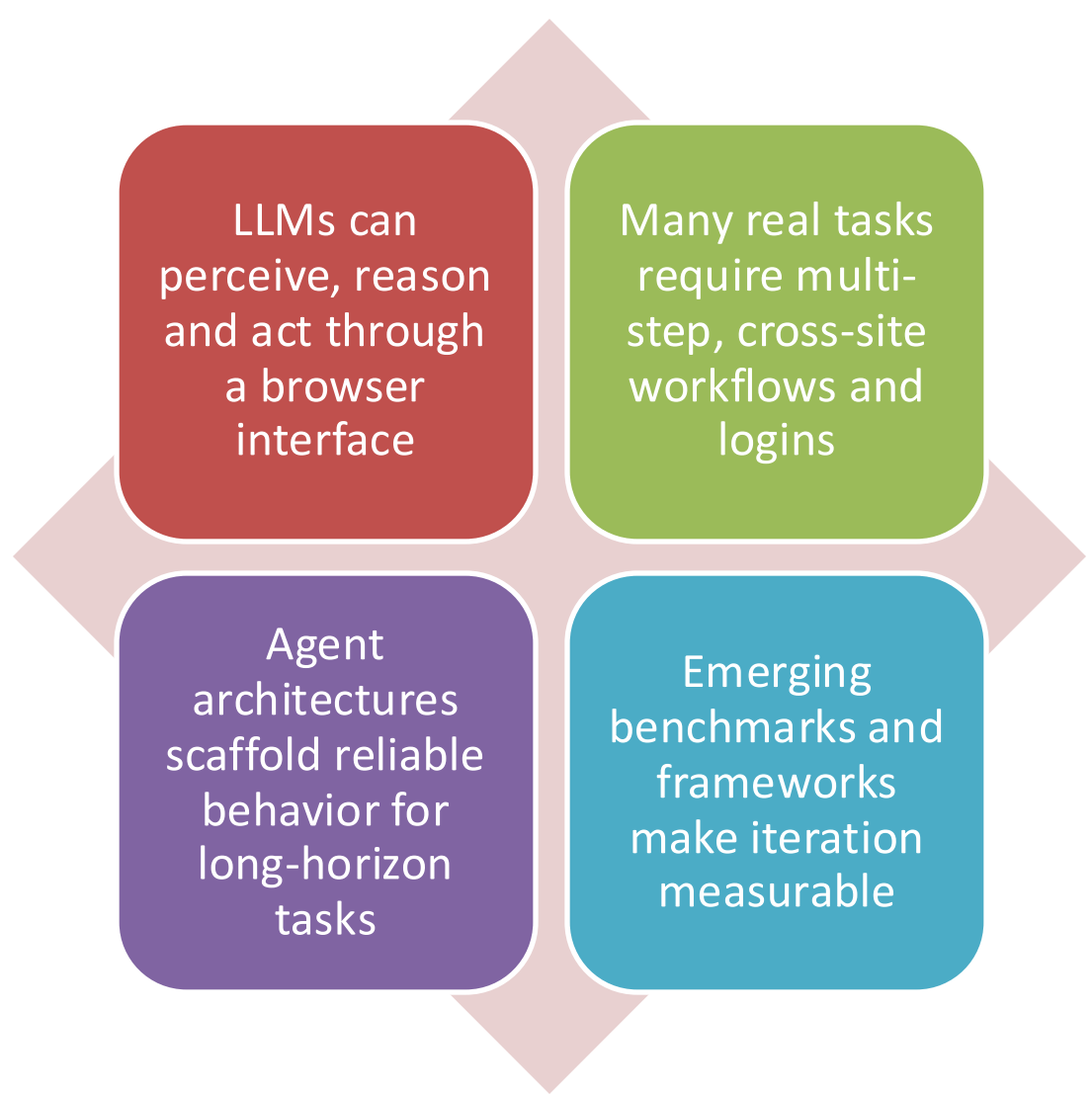


Web Agents – Architectures & Capabilities

Agenda

- What is a Web Agent?
- Architectural Building Blocks
- Planning & Execution Patterns
- Frameworks & Stacks
- Benchmarks & Evaluation
- Design Trade-offs + Case Study
- Preview of Security & Safety

Motivation: Why Web Agents, Why Now



Definition

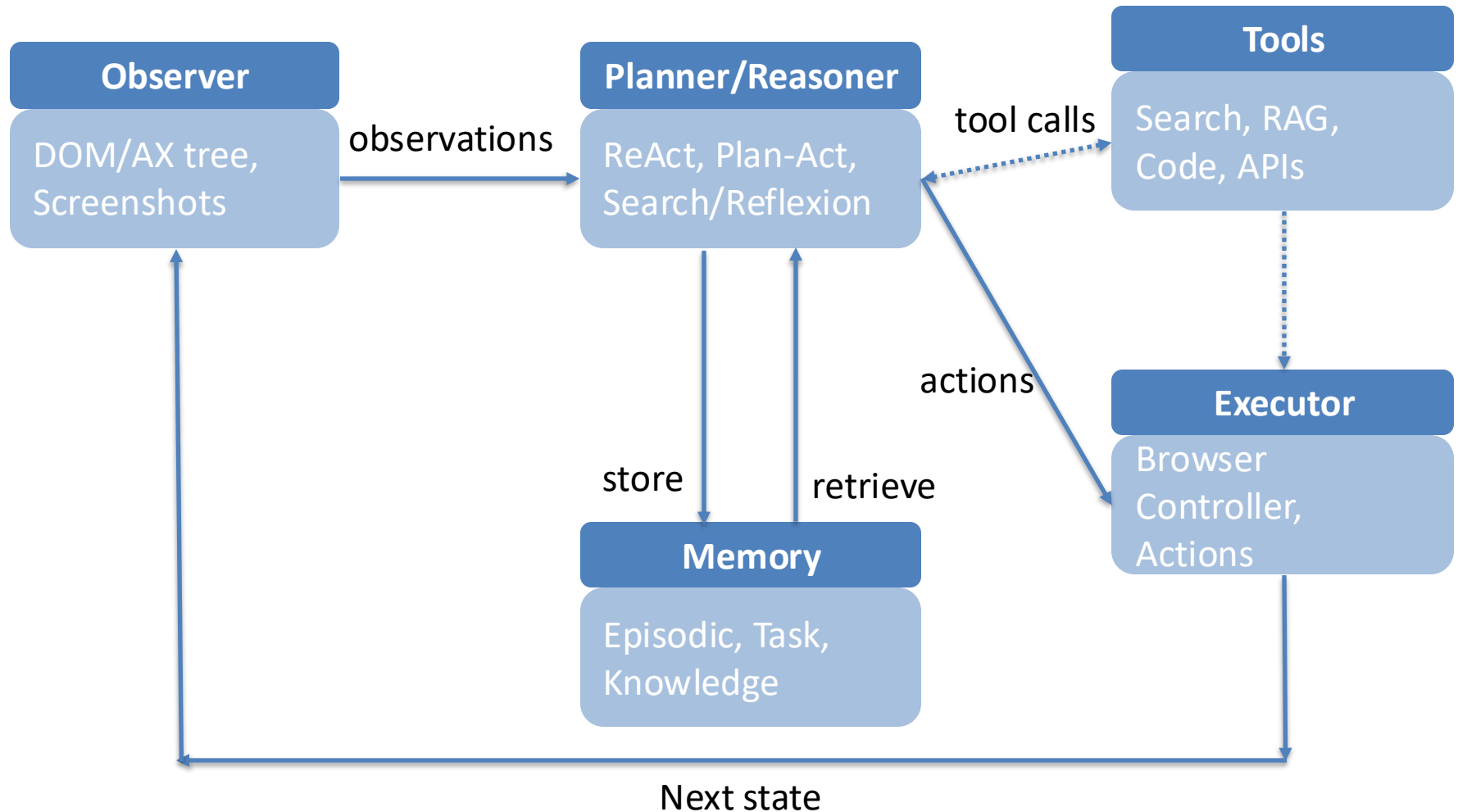
- A web agent is an autonomous system that uses a browser to achieve user goals
 - Observes: DOM, accessibility tree, screenshots, network signals
 - Acts: navigate, click, type, select, upload, execute scripts
 - Thinks: plans subgoals, tracks state, decides when to ask for help

A pile of colorful LEGO bricks in yellow, green, blue, and red, scattered on a white surface. The bricks are of various sizes and shapes, including 1x2, 1x3, and 2x4 bricks, some with studs on top and some without.

ARCHITECTURAL BUILDING BLOCKS

High-level reference architecture

Reference Architecture (Conceptual)



Observation Modalities

DOM tree (semantic structure, selectors)

Accessibility tree (role/label signal; robust to layout)

Vision (screenshots; OCR; multimodal grounding)

Network/console (performance, errors, requests)

Planning Paradigms

ReAct: interleave reasoning with actions

Plan-then-Act: dedicated planner emits subgoals for executor

Search-based: Tree-of-Thoughts / graph search over action sequences

Self-critique (Reflexion) and retry strategies

Execution Layer

Action space: `navigate(url)`, `click(selector)`, `type(text)`, `select`, `upload`

Control backends: Playwright, Selenium, custom browser wrappers

Non-determinism: dynamic DOMs, popups, auth gates; need retries & guards

Observability: structured logs, screenshots, DOM diffs

Tools & Integrations

Search & RAG for context injection

Code execution for transformation/scraping

External APIs (calendar, email, databases)

Structured tool schemas to contain untrusted input

Memory

Scratchpad (short-term chain-of-thought artifacts)

Episodic task memory (state machine, checklist, history)

Long-term knowledge (vector DBs, graphs)

Summarization and distillation to manage cost/latency

Multi-Agent Topologies

Manager–worker decomposition

Specialist agents (retriever, coder, navigator) with message bus

Graph-based control flow (LangGraph) and shared state

Human-in-the-loop escalation as a first-class path

An aerial photograph of a large, intricate green maze. The maze is composed of many narrow, winding paths that form a complex pattern of concentric and intersecting lines. The paths are filled with a dense, vibrant green vegetation, likely hedges or low-lying plants. Scattered throughout the maze are numerous small, rounded, light green bushes or trees. In the lower right quadrant of the image, a small, simple wooden building with a dark roof is visible, partially surrounded by the maze's paths. The overall scene is a lush, green landscape with a high degree of geometric complexity.

FRAMEWORKS & STACKS

Ecosystem overview

Framework Landscape



LangGraph: stateful, graph-native control for agents



LangChain: tool-using agents and chains



AutoGen: multi-agent conversational workflows



BrowserGym / WebArena: wrappers for browser control & evaluation

Case Study: Graph-Native Control (LangGraph)

- Nodes: tools, policies, model calls; Edges: conditional control flow
- Global state object drives determinism & recovery
- Easy to add approval checkpoints and retries

BENCHMARKS & EVALUATION

How we measure
progress

Key Benchmarks (Web)

WebArena — open web tasks across shopping, forums, wikis, maps

VisualWebArena — multimodal (vision+text) on real websites

WorkArena++ — enterprise workflows and authenticated tasks

WebLINX — conversational navigation and instruction-following

Metrics & Methodology

Task success rate and normalized score per domain

Step efficiency and error taxonomy (where runs fail)

Human eval for acceptability and safety checks

Ablations: observation modalities, planning strategy, guardrails

Baselines & Trends

AgentOccam: simple yet strong baseline on WebArena

Refusal-trained models may still fail as browser agents

Vision grounding helps when sites are highly dynamic



DESIGN TRADE- OFFS & CASE STUDY

Design Trade-offs (Illustrative)

Dimension	Option A	Option B	Trade-off
Observation	DOM-only	Screenshot+OCR	A: precise selectors; B: robust to dynamic UIs
Planning	ReAct	Plan-then-Act	A: simple; B: better for long-horizon tasks
Control	Synchronous	Async / multi-tab	Simplicity vs. throughput & complexity
Memory	Minimal	Rich episodic	Cost/latency vs. fewer failure loops
Safety	Free-form	Policy/approvals	Friction vs. reduced risk

Case Study
Template (Fill
with Your
Implementation)

Architecture diagram with
modules and data flow

Prompting strategy (system +
tool schemas)

State & memory design (what
is persisted)

Observability (tracing) and
evaluation protocol

Security posture (approvals,
allowlists, credentials)

Preview: Security & Safety (Next Lecture)



Agents ingest untrusted web content and can be steered (prompt/indirect injection)



High-privilege sessions widen impact of mistakes



Secure-by-design: least privilege, explicit approvals, structured tooling

Summary & Takeaways

Web agents: perception → planning → tools/memory → execution loop

Frameworks help structure control flow and state

Benchmarks provide an external check on progress

Design for reliability and observability from day one

References (1/2)

- WWW 2024 Tutorial — Large Language Model Powered Agents in the Web: <https://www2024.thewebconf.org/docs/tutorial-slides/large-language-model-powered-agents.pdf>
- EMNLP 2024 Tutorial — Language Agents (Introduction): <https://language-agent-tutorial.github.io/slides/I-Introduction.pdf>
- CS224n (2024) — Lecture 14: Reasoning & Agents: <https://web.stanford.edu/class/cs224n/slides/cs224n-spr2024-lecture14-agents-shikhar.pdf>
- Angel Xuan Chang — LLM Agents: <https://angelxuanchang.github.io/nlp-class/assets/lecture-slides/L20-LLM-agents.pdf>
- Cisco Live 2025 — AI Agents 101: <https://www.ciscolive.com/c/dam/r/ciscolive/global-event/docs/2025/pdf/BRKETI-1008.pdf>
- CiscoU-3005 — LangGraph: <https://www.ciscolive.com/c/dam/r/ciscolive/global-event/docs/2025/pdf/CISCOU-3005.pdf>

References (2/2)

- WebArena (ICLR 2024):
https://proceedings.iclr.cc/paper_files/paper/2024/file/4410c0711e9154a7a2d26f9b3816d1ef-Paper-Conference.pdf
- VisualWebArena (OpenReview):
<https://openreview.net/pdf/02cbc0b8d8d5948797bf942089564e83c84b4120.pdf>
- WorkArena++ (NeurIPS 2024 slides): <https://neurips.cc/media/neurips-2024/Slides/97713.pdf>
- WebLINX: <https://language-agent-tutorial.github.io/slides/III-Applications-Data-Evaluation.pdf>
- AgentOccam (Amazon Science):
<https://assets.amazon.science/e8/9b/35bdbcb9448da1083ec5710b7c75/agentoccam-a-simple-yet-strong-baseline-for-llm-based-web-agents.pdf>