



Austin LangChain AIMUG

Welcome to the Austin LangChain AIMUG - where Texas innovation meets AI exploration! Join our community of tech enthusiasts, developers, and AI practitioners as we learn together in an open, supportive environment.

Today's Agenda

1 Intros

Get to know your fellow AI enthusiasts

2 LangChain / LangGraph 1.0 Alpha Update

Colin shares insights from Harrison's team on directions and naming steps

3 Talks & Demos

Sal on Claude Code & custom agents,

Collier on deep agents,

James demos 'Rosie the Robot'

4 Wrap Up & After Party

Continue the conversation upstairs at The Tavern

Interviews thanks to James



Start Time: 6:20 PM

AIMUG.ORG – Docs

1

Visit AIMUG.org

Our digital home for all resources, events, and community updates

2

Explore Learn Section

Curated tutorials, guides, and educational materials for all skill levels

3

Check July 2025 Resources

Latest documentation, code samples, and community contributions

Our documentation is community-driven and constantly evolving, just like our **Texas-sized** ambitions for AI development.

The screenshot shows the AIMUG.org website's documentation section for July 2025. The top navigation bar includes links for Learn, Blog, Events, Community, and Support. The main content area features a sidebar with sections for "July 2025 Session Recording", "Documentation Sections", "News & Updates", and "Lightning Talks". The "Documentation Sections" section is currently active, displaying a list of topics such as "Main Event", "News & Updates", and "Key Highlights". The "News & Updates" section lists recent news items like "AIMUG Monthly Membership Program Launch" and "LangChain Ambassadors Meeting Updates". The "Lightning Talks" section lists presentations on topics like "Toolhouse Fastlane Worker" and "Cloudflare Containers". A GitHub icon is visible in the top right corner.



AIMUG – Member Supported – Founding Members

Founding Members

- Cameron Rohn
- Jeff Linwood
- Robert (Muzz)
- Ryan Booth
- Christopher Ewing
- John Van Lowe
- Nate Little
- Sal Castoro
- Colin Best
- Joseph Fluckiger
- Orlando Kalossakas
- Scott Askinosie
- Colin McNamara
- Kaiwalya Joshi
- R Sarasa
- Wendy Schorn
- Collier King
- Karim Lalani
- Riccardo Pirruccio
- Zack Angelo
- Hutch Enbach
- Miguel Lucero
- Roberto Fuentes

Our community thrives thanks to the passion and dedication of our founding members who share a vision for democratizing AI knowledge in the heart of Texas:



Corporate Underwriters



Our initiatives are made possible through the generous support of organizations that believe in the power of community-driven AI education and innovation:

Center for Government and Civil Services

Supporting public sector AI applications and education

LangChain AI

Providing technical resources and framework support

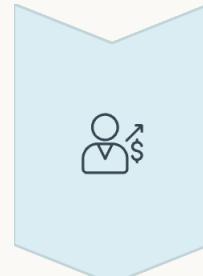
Always Cool Brands / AI

Championing ethical AI development across industries



Toolhouse

Hi, I'm Colin McNamara



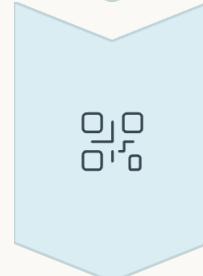
Entrepreneur

Co-founder of Always Cool Brands | AI Using AI to take Poison out of our Food Supply ([LangChain](#) / [LangGraph](#) User)



Community Builder

Founding Member of Austin LangChain AIMUG, committed to growing our Texas AI ecosystem



50% Developer

Consulting Engineer, Cloud Builder, Casual Coder with a passion for practical AI solutions

[Public Good](#): Helping our community succeed by bringing Texas hospitality to the AI world

Connect with me on [LinkedIn](#)



Who We Are



1400+

Members Strong

Connecting Austin's tech, creative, and business leaders passionate about AI, bringing together Texas innovation with cutting-edge technology



100%

Learning in the Open

Exploring AI through the lens of LangChain, pulling the ecosystem together with Texas-sized ambition and collaborative spirit

24/7

Supportive Community

A well-balanced Discord that's "useful, but not overwhelming" - reflecting our commitment to southern hospitality in the tech world



Learning in the Open – "Be Cool to Each Other" 🤝

Our community values reflect the best of Texas: friendliness, openness, and a genuine desire to help each other succeed.



Respect & Inclusivity

Our only rule is simple: "Be cool to each other, don't be gross." Texas hospitality meets tech community values.



AI Enthusiasts of All Levels Welcome

We celebrate diverse perspectives and unique paths in AI, whether you're a beginner or expert.

Where and When We Meet



Monthly Showcases, Labs & Mixer

Location: ACC RGC3000

Community members and vendors share projects, demos, and hands-on learning opportunities in a collaborative Texas atmosphere

Weekly Virtual Syncs

Online: Tuesdays at 5PM

Workshopping, problem-solving, and AI discussions from the comfort of your home or office

1

2

3

Hacky Hours (ad hoc)

Location: Rotating Local Bars

Informal happy hours at local spots; meet, discuss, and relax with fellow AI enthusiasts over Texas craft beers

WTF is LangChain?

LangChain is a framework for developing applications powered by language models, enabling:

Context-Aware Applications

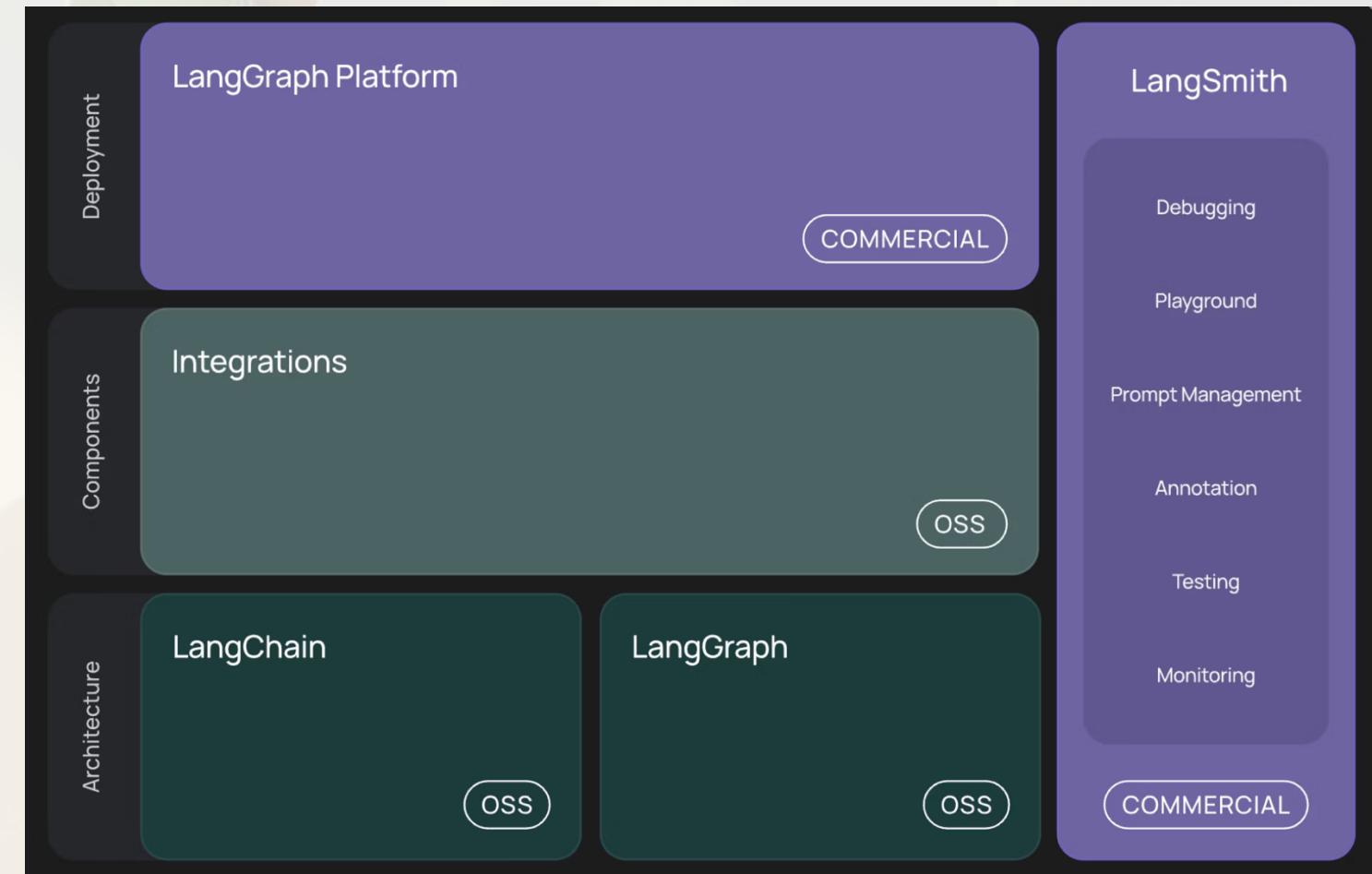
Connect LLMs to external data sources and systems

Reasoning Capabilities

Chain multiple LLM calls together for complex logic

Agent-Based Systems

Create autonomous AI assistants that interact with tools



Just like Texas prides itself on building connections, LangChain connects AI models to make them more powerful and practical.

Talks & Demos!

LangChain/LangGraph Update – Colin

Insights into LangChain/LangGraph 1.0 Alpha, including discussions and updates from Harrison's team.

Cloud Code & Custom Agents – Sal

Dive into how Claude Code, custom agents, and subagents work together to build powerful AI applications.

Deep Agents – Collier

An in-depth presentation exploring the concepts and applications of deep agents.

"Rosie the Robot" Demo – James

A 15-minute live demonstration of a desktop cleanup project, incorporating mind chain topics and practical AI solutions.

"Streamlit Dashboard – Jeff Linwood

A quick overview of how to build dashboards in streamlit that incorporate LangChain



Our mascots are ready for a variety of exciting lightning talks and demos!

LangChain/LangGraph 1.0 Alpha Update

Colin McNamara



Langchain

Colin shares the latest developments and strategic insights directly from Harrison's team at LangChain, shaping the future of LLM applications.

LangGraph 1.0α & LangChain 1.0α – the new defaults for building agentic systems

Alpha releases for **LangGraph** and **LangChain** in **Python & JS** are here, targeting **late October** for the official 1.0 release. This represents a fundamental shift toward shrinking surface area and hardening the runtime to ship faster with more reliability.

[Alpha notice:](#) docs and content are evolving rapidly. Treat this as **try now, production-gate later** while we finalize the 1.0 release.

The big picture in one slide

LangChain 1.0α

A **single, focused agent** (`create_agent` / `createAgent`)
built on LangGraph

LangGraph 1.0α

Promoted with **no breaking changes**; durable
execution, streaming, HTML, and time travel

LangChain-Core

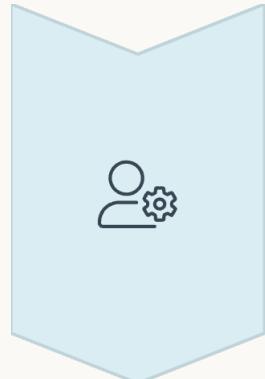
Introduces **standard message content** via
.content_blocks / `contentBlocks` (typed, provider-
agnostic, **back-compatible**)

Unified Docs

Unified docs site across Python & JS for streamlined
developer experience

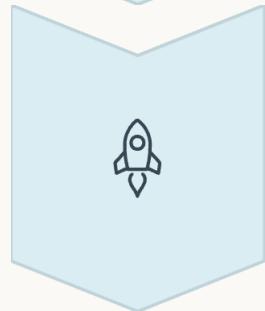
This is the mental model shift: LangChain = entry + patterns; LangGraph = runtime & control.

Roles & mental model (builder-friendly)



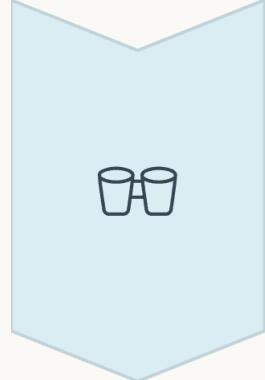
LangGraph = runtime/orchestrator

State graphs, **checkpoints** (threads), **HITL** (interrupt),
streaming, **time travel**



LangChain = getting started & patterns

The **agent** interface + **standard content blocks**



Key relationship

The LangChain agent **runs on LangGraph** → 10-line start,
production-grade runtime



LangChain 1.0α: what actually changed

One agent abstraction

`create_agent/createAgent` (ReAct-style loop) now **on LangGraph**

Standard content

`.content_blocks / contentBlocks` unify **reasoning, citations, tool calls, multimodal; backwards-compatible**

Slimmer surface + compatibility

Legacy chains/agents move to **langchain-legacy**

Retrievers are **under discussion**; the package is **significantly slimmed down** with a focus on agents. This emphasizes fewer choices, better defaults, and a clear migration path.





LangGraph 1.0α: what actually changed

01

Conservative v1

Mostly **deprecation cleanup**; core runtime unchanged

02

Runtime guarantees

Deterministic concurrency (Pregel/BSP), **loops, parallelism**

03

Built-in capabilities

HITL via interrupt, **checkpointing/threads, streaming modes, time travel**

Why this matters: **stability + observability + control**. The conservative approach ensures production systems can upgrade confidently.

Streaming modes (design your UX)



messages

Token stream + model metadata (great for chat feel)



updates

State deltas (progress/events for dashboards)



values

Full state snapshots (visualize evolution)

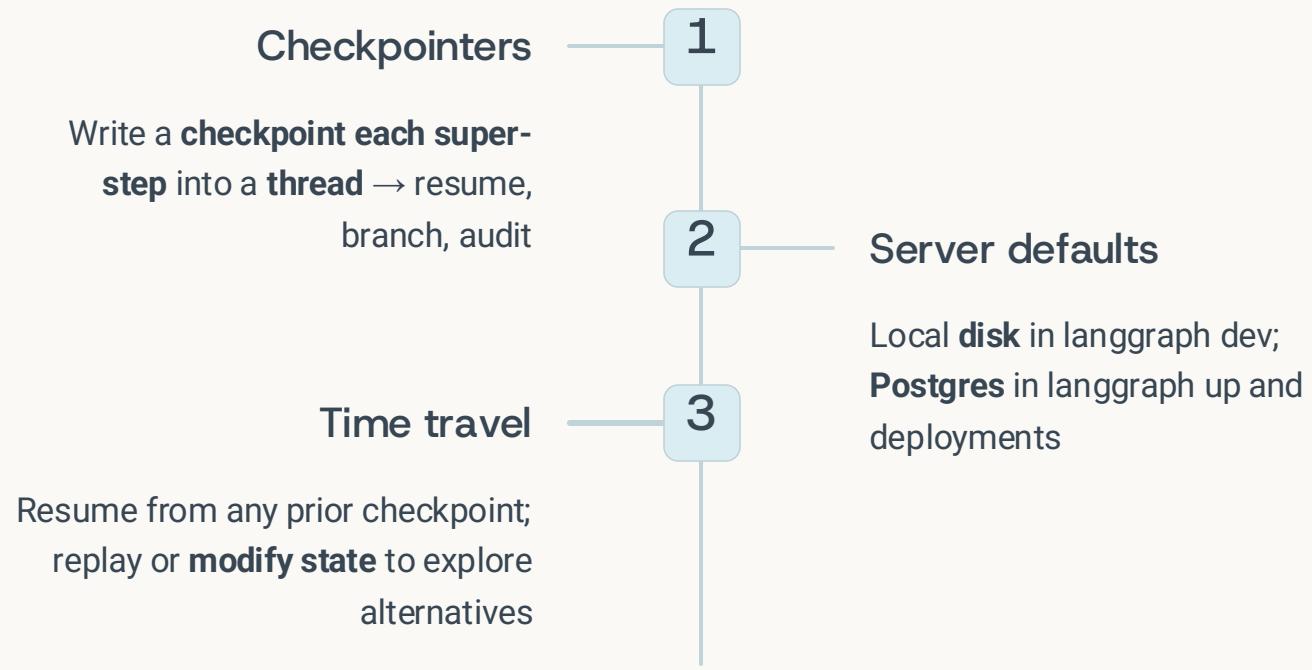


custom / debug

Arbitrary signals / traces when you need them

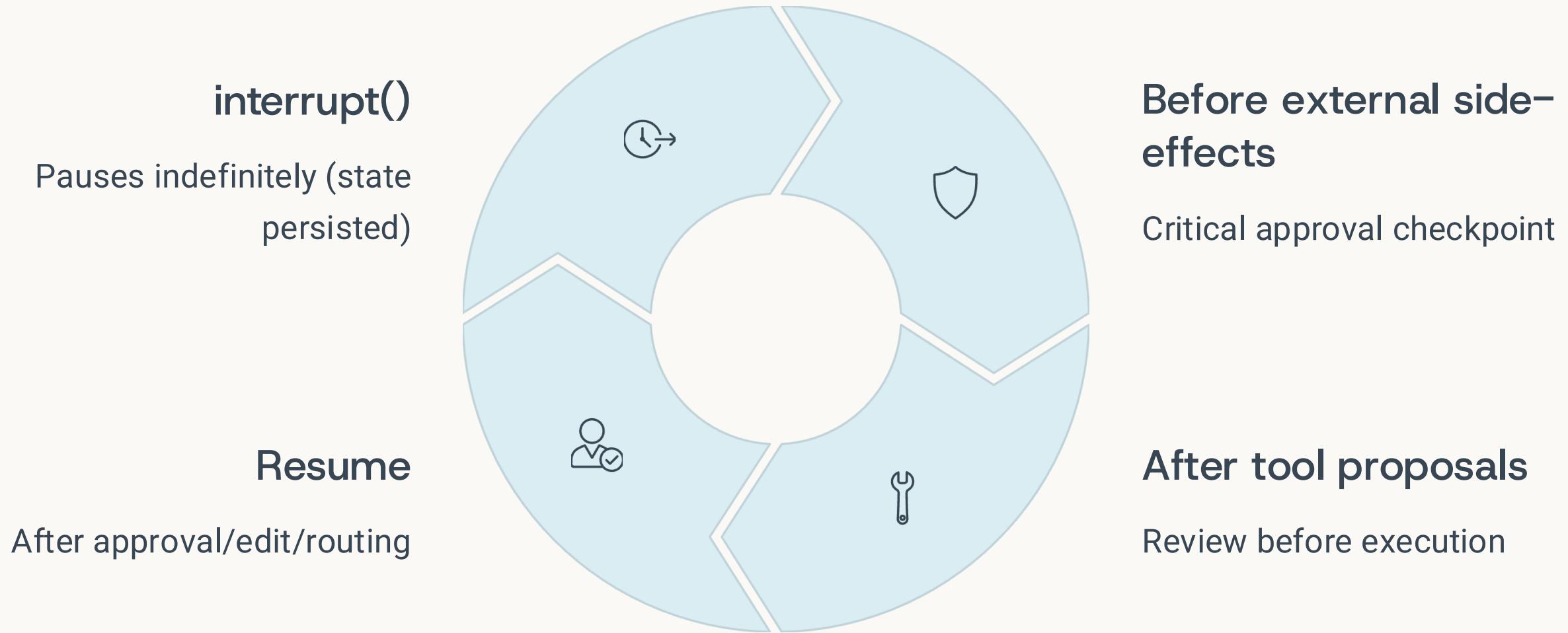
Pick the mode(s) that reduce perceived latency and increase user trust in your agentic systems.

Persistence & time travel (what builders must plan)



ⓘ Treat threads/checkpoints like your production safety net and memory backbone.

Human-in-the-loop (HITL) patterns



HITL + checkpoints lets you add enterprise controls without brittle hacks. Design approval gates for **policy gates** and **high-risk actions**.

Standard content blocks (why they matter)



1

One typed view across providers

OpenAI, Anthropic, Responses API, etc.; normalizes **reasoning, citations, tools, multimodal**

2

Zero breakage

Computed lazily from existing .content

3

Practical win

Fewer provider branches, consistent UIs, easier model swaps

"This removes half the glue code when you change models."

Migration & compatibility quick hits

3.10

Python minimum

3.9 dropped in v1

20

Node minimum

Several legacy subpath exports removed



Pin alpha versions; read the Python/JS v1 release notes before upgrading

Legacy path: langchain-legacy keeps older chains/agents working as you adopt v1. Use `createAgent` and `ToolNode` for new JavaScript projects.

What to build (concept checklists)



Default path

Start with the **LangChain agent** (it already rides LangGraph)



Runtime design

Choose **streaming mode(s)**, define **thread ids**, pick a **checkpointer** (SQLite/PG/Redis) per environment



HITL

Mark **interrupt points** before risky effects; design resume UX



Content

Adopt **content blocks** in your renderers/logging

Concrete, immediate steps your team can do tonight—no live coding needed.

Proof that the runtime is production-grade



Used in production

Teams at **Uber**, **LinkedIn**, **Klarna** and others are using LangGraph in production environments. Design choices (Pregel/BSP + checkpoints) reflect what real agent systems need at scale.

The goal is confidence to evaluate and adopt these frameworks in your own production systems.

Anti-patterns to avoid



Single function agents

No structure → no checkpoints/HITL. Prefer **graphs**



Token-only streaming

Users need **progress** → add updates mode



Ephemeral memory

In prod → add real **checkpoint + threads**



Provider-specific parsing

Hard-coded → use **content blocks**

These are the most common "gotchas" we see in code reviews. Avoid these patterns to build more maintainable agentic systems.

Docs & learning path (what's improved)



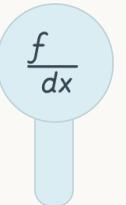
Unified OSS docs site

Python & JS together for streamlined learning



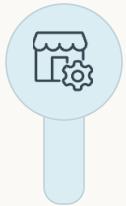
Dedicated guides

Streaming, persistence, HTL, time travel



Integration docs prioritized

Contrib guide and YouTube series organization



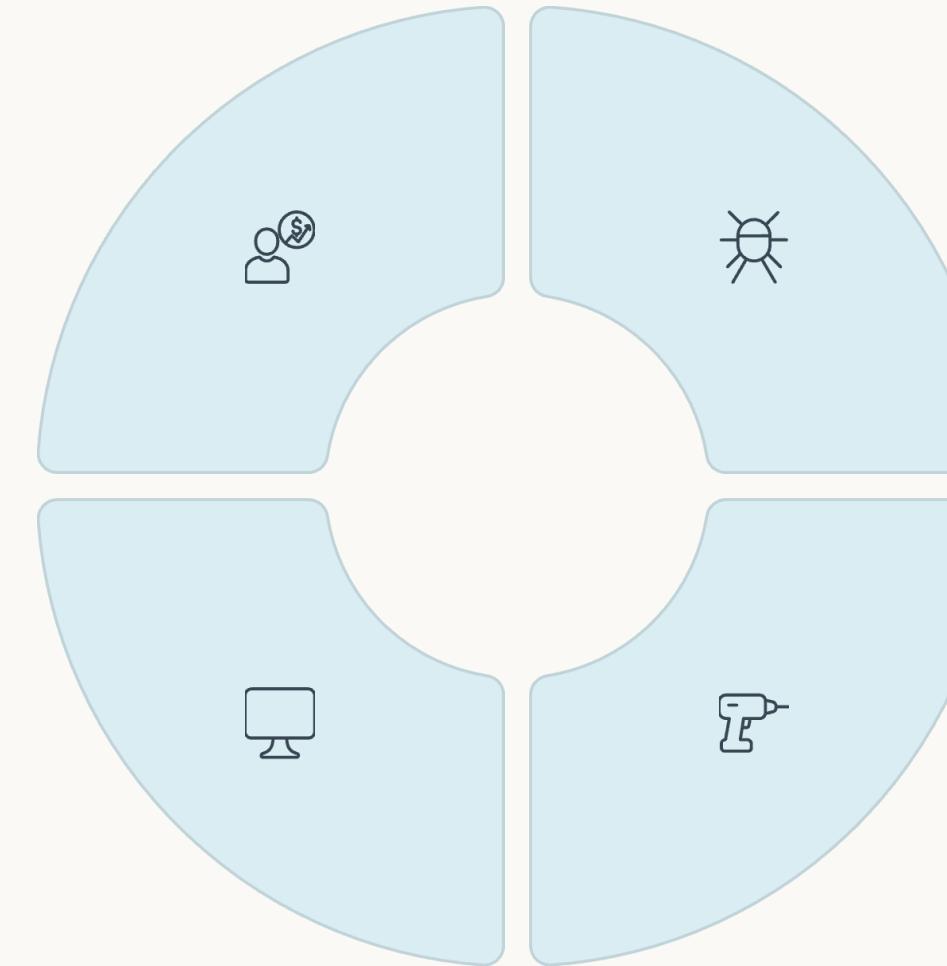
Notebook → enterprise templates

Production-ready patterns and examples



Insights & analytics (forward-looking)

Usage patterns
New **Insights** clusters common behaviors



Failure modes
Identify and analyze system failures

Future monitoring
Hooks for advanced observability

Drill-downs
Deep investigation capabilities

ⓘ Currently in **beta** behind a feature flag

Useful for observing agent behavior across threads and checkpoints in production environments.

References



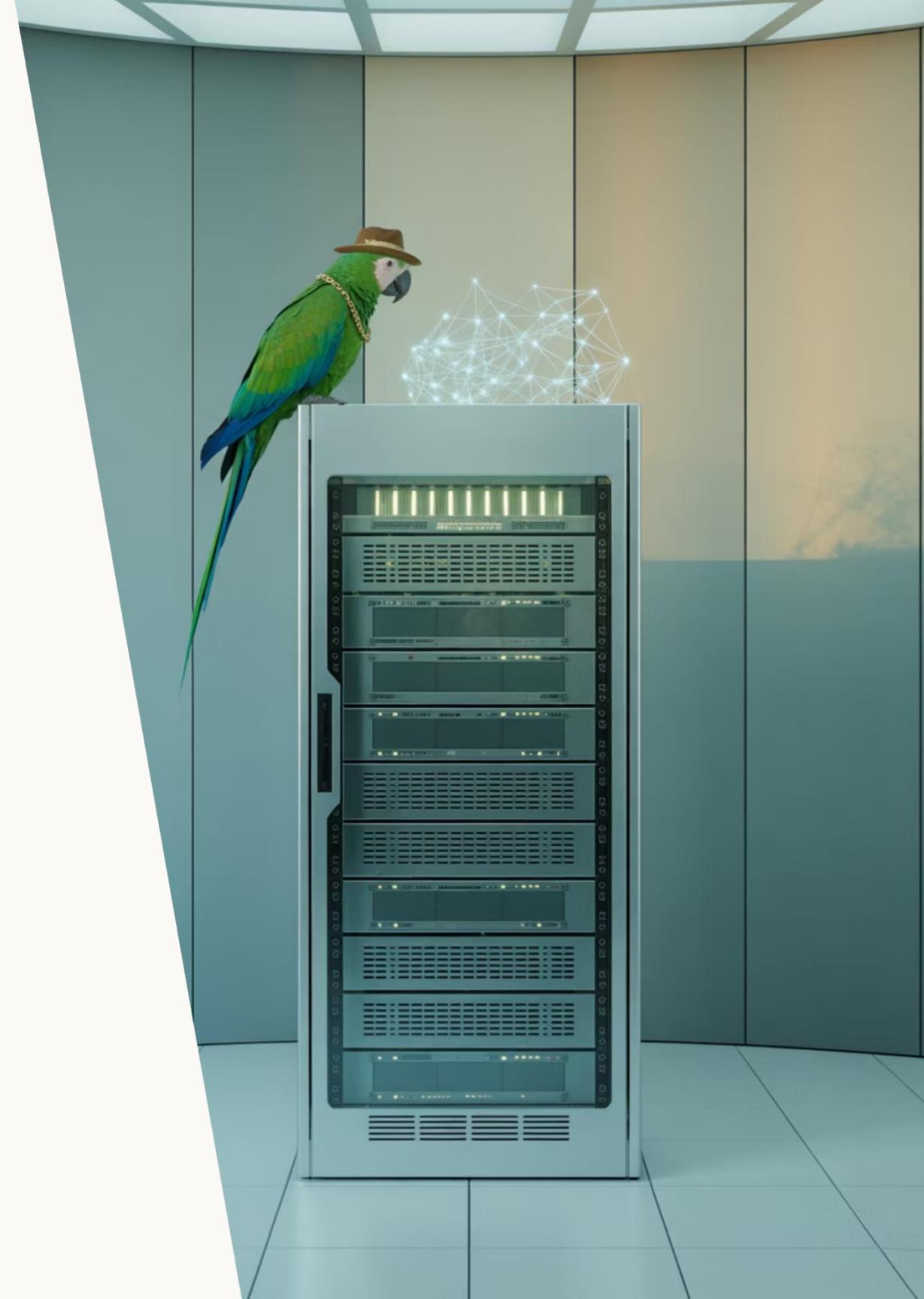
Key Resources:

- [Alpha announcement & timeline](#) (late Oct target; agent built on LangGraph; legacy pkg; unified docs)
- [LangGraph design \(Pregel/BSP\)](#)
- [Streaming modes](#) (values/updates/messages/custom/debug)
- [HITL / interrupt](#)
- [Persistence/threads & time travel](#)
- [Content blocks explainer](#)
- [JS v1 notes](#) (Node ≥20, createAgent, ToolNode)
- [Python v1 notes](#) (Python ≥3.10, legacy, .content_blocks)
- [Production usage](#) (Uber, LinkedIn, Klarna)

Presented by Sal Castoro

Claude Code & Custom Agents

Building intelligent systems requires seamless orchestration between backend logic and dynamic AI capabilities. Let's dive into how Claude Code empowers custom agents and their subagents to deliver robust, scalable solutions.

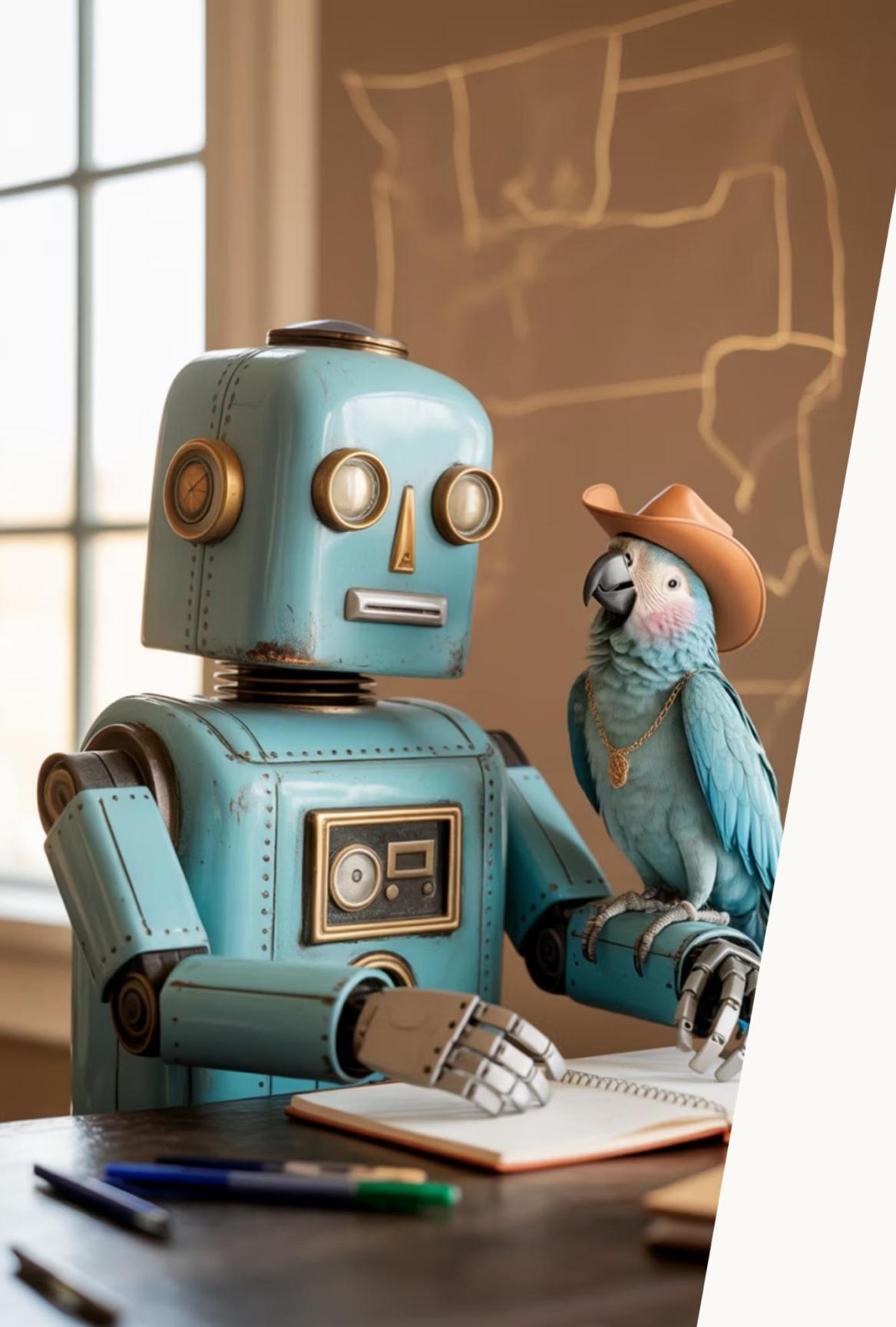




Presented by Collier King

Deep Agents: Practical Insights

Collier will share how deep learning is making AI agents smarter and more capable, allowing them to go beyond simple tasks and tackle complex problems with adaptive behavior.



Presented by James

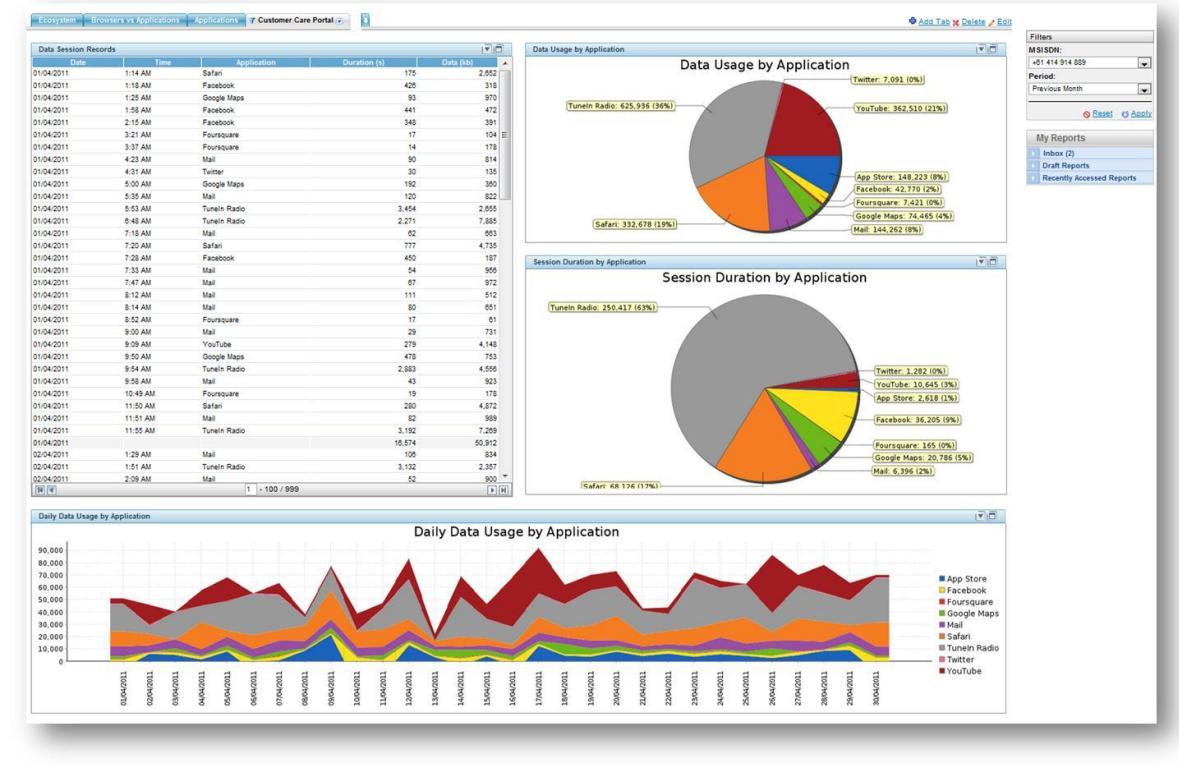
Rosie the Robot: Desktop Cleanup Demo!

Join James for a live demonstration of "Rosie the Robot"—a practical AI application designed for efficient desktop automation and digital workspace organization.

Presented by Jeff Linwood

Streamlit Dashboard:

Join Jeff for a quick intro to streamlit as a dashboard, with a quick intro of how to integrate LangChain into it.



This Photo by Unknown Author is licensed under [CC BY-SA](#)

RGC-3000 and Tavern After Party

Join us after the meeting to continue conversations, network, and enjoy some Texas hospitality at The Tavern! Our green parrot mascot might even make an appearance.



Website

aimug.org – All things LangChain and AIMUG



Discord

[Join us](#) for ongoing conversations



GitHub

[Austin LangChain Repo](#) – Check out our projects



Meetup

[Austin LangChain Group](#)



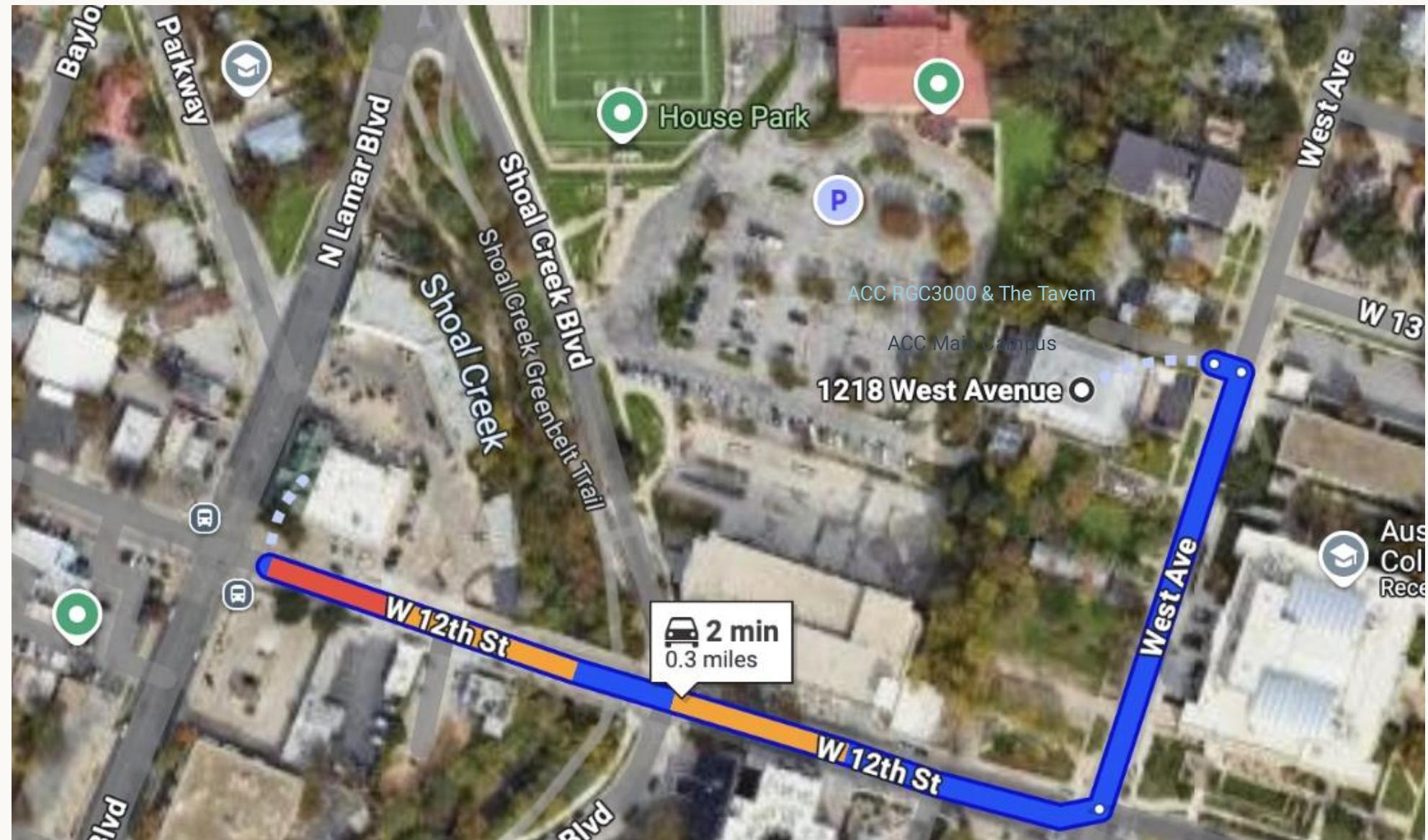
Twitter

[@AustinLangChain](#) – Latest updates



YouTube

[Austin LangChain Channel](#)



Finding US – ACC RGC-3000

The Austin Community College Rio Grande Campus (RGC) Building 3000 is our regular meeting location, offering modern facilities perfect for our tech-focused gatherings.

Easy Access

Centrally located in downtown Austin with parking available nearby

Tech-Ready Facilities

State-of-the-art presentation equipment and reliable Wi-Fi

Collaborative Spaces

Room arrangements perfect for both presentations and hands-on workshops

Look for Tex, our green parrot mascot with his cowboy hat and gold chain, greeting you at the entrance!



Next Event – October 1 ACC RGC3000

Wednesday, August 27th



Casual Networking

Join fellow AI enthusiasts for coffee, conversation, and coding in a relaxed Austin atmosphere



Bring Your Projects

Get feedback, troubleshoot, or collaborate on your latest LangChain experiments



All Welcome

From beginners to experts, our community embraces diverse perspectives and experience levels

"Be cool to each other, don't be gross." Our only rule reflects our commitment to Texas hospitality in the tech world.

Keep an eye out for Tex, our green parrot mascot with his cowboy hat and gold chain – he'll be there too!