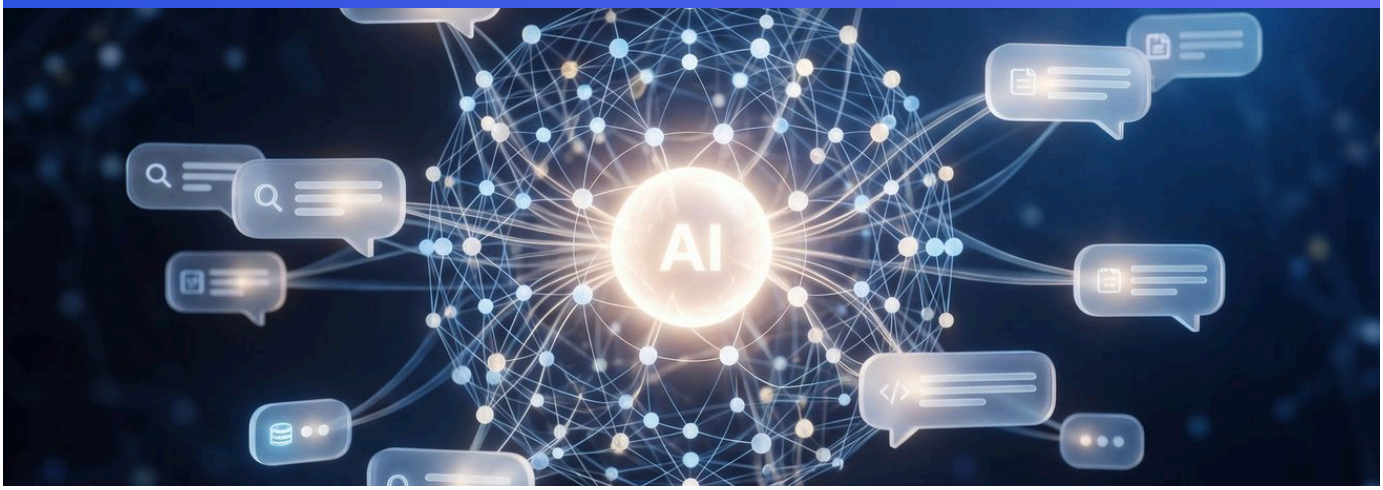
BURDENOFF · PRODUCT BROCHURE

Botlit

The enterprise agentic AI platform — build agents you can see, version, and govern.



Overview

Botlit is an enterprise-grade agentic AI platform built on the Burdenoff Workspaces supergraph. Teams use it to build, govern, and ship AI agents that run on their own model providers, answer from their own knowledge, speak on the channels people already use, and live entirely inside their workspace's security model — with versioning, per-execution cost audit, and gateway-enforced RBAC built in.

The control plane fits together as a chain: user-facing bot apps front the specialist agents behind them; those agents are grounded in knowledge bases via managed RAG; they bind to LLM providers of your choice; and they deploy out to channels — web chat, Slack, Teams, Discord, WhatsApp, and webhooks. Every app, agent, knowledge base, and conversation lives inside a governed, multi-tenant workspace with platform RBAC, activity audit, billing, notifications, i18n, and file management already in place.

Unlike point-solution chatbot builders, Botlit is a module of a full workspace platform; unlike raw agent frameworks, it ships a complete management surface — versioning, templates, a package store, execution logs with token and cost metrics, policy definitions, and an embedded FluidGrids visual workflow builder. The management plane and execution record-keeping are real today; the next wave wires retrieval grounding, tool-calling, and policy enforcement into the live agent loop.

Trust by construction, not by faith

Every GraphQL field carries a gateway-enforced RBAC directive, every agent run writes an execution record with token and cost metrics, and every agent and app is versioned with publish, changelog, and rollback. You can see, version, and govern everything an agent does.

Bring your own model, your own knowledge

Register any of 11 LLM provider types with vaulted keys and connection tests, and ground agents in managed knowledge bases built from your own documents. You stay in control of your models, contracts, and data.

A complete management plane, not a framework

Apps, agents, versioning, providers, knowledge, channels, policies, a package store, and execution telemetry — surfaced in one governed UI inside a multi-tenant workspace, so non-ML teams can run an agent workforce without standing up infrastructure.

Meet people where they already are

Connectors and routes deploy agents to Slack, Teams, Discord, WhatsApp, and webhooks. Inbound messages acknowledge instantly while agents respond in the background, and every conversation lands in one shared team inbox.

Key capabilities

1 Bot apps & agents with version control

Full CRUD, lifecycle status, and visibility for user-facing bot apps and the specialist agents behind them. Give each agent a system prompt and a model binding, publish versions with changelogs (DRAFT to PUBLISHED to HISTORICAL), and roll back when a prompt change regresses behavior. Agent capability flags cover RAG, MCP tools, A2A, reasoning, multimodal, code execution, web browsing, and memory.

2 App-to-agent routing

A bot app is the front door; agents are the specialists. AppAgent links connect an app to its agents with a routing strategy — single, round-robin, intelligent, or custom. Link management ships today; intent-based intelligent dispatch at runtime is on the roadmap.

3 Live chat & execution

The chatWithAgent mutation resolves the agent's model and provider credentials, calls the LLM, records an Execution, and stores both sides of the exchange in the platform conversations service. Today's runtime is single-turn; multi-turn history replay, RAG context injection, and live MCP tool-calling are on the roadmap.

4 Knowledge bases & managed RAG

Build managed retrieval corpora: documents are chunked with one of five strategies (fixed, recursive, semantic, markdown, code), embedded (default text-embedding-3-small) into a per-KB Milvus vector collection, and validated with semantic search. KB lifecycle runs EMPTY to INDEXING to READY, with index rebuild. Runtime grounding inside the live chat answer is on the near-term roadmap.

5 Multi-provider LLM management

Provider and model registries for 11 provider types — OpenAI, Anthropic, Google, Cohere, Mistral, Groq, Together, HuggingFace, Azure OpenAI, AWS Bedrock, and custom endpoints — with connection testing and per-model parameters. The execution path uses an OpenAI-compatible client today; native protocol adapters for Anthropic, Bedrock, and Google are on the roadmap.

6 Credential routing & vaulting

At execution time keys resolve in priority order: workflow-injected key, then platform integrations connection (OAuth/token vault via the internal gateway), then stored credential, then environment fallback — with a resolution cache to keep latency down. Keys are never stored in agent definitions; they live in per-environment Azure Key Vault.

7 Channels & message flow

Connector and route management for seven channel types (web widget, Slack, Teams, Discord, WhatsApp, webhook, custom). Outbound messages are stored and delivered in parallel; inbound processing is fire-and-forget so the platform webhook never waits on LLM latency. All bot conversations land in the shared platform conversations inbox alongside human messaging.

8 MCP integration

A Model Context Protocol connection registry with status tracking, connection testing, and tool/resource discovery and refresh. Discovered tools are not yet invoked inside the live chat loop; agent-side tool execution lands with the runtime upgrade, which the Python agent stack already demonstrates.

9 A2A agent-to-agent interoperability

An open-protocol A2A registry of known agents with explicit trust levels (unverified, verified-publisher, internal), plus connections and prioritized message send/acknowledge with delivery status and a message-received subscription. This is the protocol and data layer today; live cross-agent delegation in production conversations is on the roadmap.

10 Prompt templates, skills & policies

Versionable, variable-parameterized prompt templates ({{variable}} rendering, duplication, slug lookup) standardize agent instructions. Skills are reusable, distributable capability definitions. Policies span seven guardrail types — content, tool, data, response, budget, rate-limit, retention — with tool-risk levels. Policy definitions ship today; runtime enforcement in the execution path is on the roadmap.

11 Executions, analytics & monitoring

Every agent, app, and workflow run is recorded with status transitions (pending, running, completed, failed, cancelled, timeout), input/output, and metrics — total/input/output tokens, cost, tool calls, and RAG chunks — plus per-agent and per-app stats, a live execution subscription, and the Botlit dashboard. A platform-wide analytics module is available in the app shell.

12 Templates, package store & buckets

Curated app/agent blueprints with one-click instantiation, plus a package store distributing ten kinds (app, agent, MCP pack, prompt template, skill, retrieval profile, policy bundle, channel template, A2A agent, KB template) as LINKED installs that track upstream or FORKED copies you own. Buckets give hierarchical folders for organizing the fleet.

13 Embedded FluidGrids workflow orchestration

The FluidGrids visual workflow builder is embedded directly in Botlit at /botlit/workflows — the same engine used across the Burdenoff platform — so agents compose into larger automations. Workflow credential adapters can inject LLM keys into the chat execution path.

How it works

1 Build

Build agents from a template or from scratch. Give each a system prompt and a model binding, link apps to agents with a routing strategy, then publish a version with a changelog — and roll back a bad change. Organize the fleet with buckets and visibility controls.

2 Ground

Turn your documents into managed RAG corpora: chunk with the strategy you choose, embed into a per-knowledge-base vector collection, and prove retrieval quality with semantic search on real queries before anything goes live. Reuse retrieval profiles and share them as store packages.

3 Deploy

Bind your workspace to Slack, Teams, Discord, WhatsApp, and webhooks with connectors, then map routes to bot apps. Inbound webhooks acknowledge instantly while agent runs happen in the background; both sides of every exchange land in the shared conversations inbox.

4 Govern & observe

Every GraphQL field is gated by an RBAC directive enforced at the gateway edge, and every run writes an execution record with token and cost metrics. Define policies across seven types, track spend per run, and keep full version history with one-click rollback.

Who it's for

Digital agencies & consultants

Build and operate branded AI assistants for many clients from one platform, each walled off in its own workspace, and resell agent packages through the store.

Enterprise organizations (1000+)

Governed, auditable agent deployment at scale — RBAC, policies, a multi-provider model strategy, and private knowledge across the org.

Healthcare organizations

Member- and patient-facing assistants with strict data-retention and content policies. HIPAA posture is on the compliance roadmap.

Mid-market companies (100-1000)

Deflect support volume and automate internal helpdesk and HR Q&A — without standing up an ML team.

Technology startups & scale-ups

Ship an in-product AI assistant fast, with API-first access through the GraphQL gateway.

Editions

Free

Trying the platform

- Capped apps & agents with a starter execution quota
- A few knowledge bases
- Bring your own model keys

Pro

Running a real workload

- Generous app & agent caps and a high monthly token quota
- Many knowledge bases plus advanced RAG
- MCP servers and A2A collaboration

Enterprise

Governing at scale

- Unlimited apps, agents & knowledge bases
- Multi-agent orchestration
- SSO/SAML, custom branding, SLA, and custom quotas

Services we offer

- **Custom AI agent & bot development — bot apps and specialist agents, routing, versioning, and rollback**
- **Knowledge base & RAG setup — ingestion, chunking, embeddings, per-KB vector collections, and retrieval validation**
- **Channel integration — connectors and routes for web, Slack, Teams, Discord, WhatsApp, voice, and webhooks**
- **Conversational AI strategy & consulting — use-case discovery, agent architecture, and a shipped-vs-roadmap rollout plan**
- **Governance, policy & guardrails setup — RBAC, seven policy types, tool-risk levels, and audit-ready execution records**
- **Managed agents & bot operations — monitoring, prompt and routing tuning, safe versioning, and channel health**
- **Model routing, prompt engineering & evaluation — provider/model registration, versioned prompt templates, and eval loops**
- **Embedded & white-label agents — API-first GraphQL integration, per-client workspaces, and store packaging**
- **Training & enablement — role-based workshops, hands-on labs, runbooks, and office hours through first rollout**

Why Botlit

A module of a full workspace platform

Botlit inherits platform auth, RBAC, conversations, channels, integrations, activity audit, files, billing, notifications, and i18n — so agents are governed by the same identity and permission model as everything else the organization runs, not a siloed chatbot tool.

Honest about what ships today

Botlit is pre-launch and clearly labels what is available, partial, or planned. The management plane and execution telemetry are real now; runtime tool-calling, RAG grounding, live A2A delegation, and policy enforcement land in the open as roadmap — no hand-waving.

Bring-your-own-provider by design

Model access is workspace-scoped and BYO across 11 provider types with credential vaulting and connection tests — you keep your contracts, your keys, and your model strategy rather than being locked to one vendor.

A proven multi-agent runtime blueprint

The companion botlit-agent-stack (Google ADK + Agno + the open A2A protocol) demonstrates coordinator, team, and pipeline patterns and cross-framework agent delegation end-to-end — the blueprint the managed runtime is converging on.

By the numbers

11
LLM provider types (OpenAI, Anthropic, Google, Bedrock, Azure, and more)

21
Federated GraphQL API modules

7
Policy guardrail types

7
Channel connector types

10
Store package kinds

Per-token
Cost visibility on every run

FAQ

Do I bring my own model providers?

Yes. Botlit is bring-your-own-model by design. Register any of 11 provider types — OpenAI, Anthropic, Google, Cohere, Mistral, Groq, Together, HuggingFace, Azure OpenAI, AWS Bedrock, or a custom endpoint — with your keys, your contracts, and a connection test before any agent depends on them.

How do you keep agents trustworthy?

Trust is structural. Every GraphQL field carries a gateway-enforced RBAC directive, every agent run writes an execution record with token and cost metrics, and every agent and app is versioned with publish, changelogs, and rollback. You can see, version, and govern everything an agent does.

Which channels can I deploy to?

Connectors and routes cover Slack, Teams, Discord, WhatsApp, and webhooks today, with a web-widget channel type modeled. Inbound messages acknowledge instantly while the agent runs in the background, and every conversation lands in a shared team inbox.

Can agents answer from my company knowledge?

You can build managed knowledge bases today — ingest documents, chunk and embed them, and validate retrieval with semantic search. Injecting those retrieved chunks into the live chat answer (RAG grounding) is on the near-term roadmap; we label what ships and what is coming.

What about MCP tools and agent-to-agent collaboration?

The MCP connection registry and the A2A registry, connections, and messaging are in place as the protocol and data layer, with trust levels and tool discovery. Invoking MCP tools and delegating across agents inside live production conversations lands with the runtime upgrade — the Python agent stack already proves the patterns end-to-end.

Is Botlit available now?

Botlit is pre-launch. The management plane — apps, agents, versioning, providers, knowledge bases, channels, policies, the store, and execution telemetry — is real today, and we are opening to early-access teams through the waitlist. Runtime tool-calling, live A2A delegation, and policy enforcement land in the open as roadmap.

Ready to put AI agents to work? Join the waitlist for early access to Botlit at launch.

botlit.ai · app.botlit.ai