

FluidGrids

Draw your process once. Trust it to run forever.



Overview

FluidGrids is a visual workflow automation and orchestration platform. You describe what should happen — "when a Stripe payment lands, enrich the customer in HubSpot, post to Slack, and push the numbers to a dashboard" — by dragging nodes onto a canvas and wiring them together. FluidGrids turns that picture into a versioned, executable workflow that runs in the background, with per-node logs, typed errors, retries, and a complete run history.

Build on a drag-and-drop React Flow canvas, then let FluidGrids run it reliably in the background on horizontally-scaled workers driven by Redis Streams. Connect 60+ services — Slack, GitHub, Stripe, Salesforce, OpenAI, Anthropic, Google Workspace and more — plus sixteen built-in flow-control nodes. And because the entire builder ships as an embeddable React SDK, SaaS product teams can offer native, white-label workflow automation inside their own product in days instead of quarters.

FluidGrids is workspace-native: every workflow, run, connection, and credential belongs to a workspace, so multi-tenancy, RBAC, billing, audit, and notifications are built in from day one — inherited from the Burdenoff platform rather than bolted on. Most automation tools force a choice between simple-but-shallow and powerful-but-developer-only; FluidGrids sits in the middle and adds a third dimension no competitor leads with: it is a product and an embeddable component at the same time.

One canvas for operators and engineers

The graph you draw is the graph that runs. A drag-and-drop builder readable by the operations manager and the engineer alike — no choosing between approachable and powerful.

Execution you can defend

Versioned definitions, runs pinned to their exact version, per-node state, typed errors, and surgical single-node retries. Auditors and SREs get the same story, with no black boxes.

Embed-first, not an afterthought

The same builder that powers the FluidGrids app ships as @burdenoff/fluidgrids-fe-sdk. Ship white-label automation inside your own SaaS without building an engine, a canvas, or a connector catalog.

Enterprise plumbing on day one

Multi-tenancy, RBAC, billing, usage metering, audit, notifications, and i18n are inherited from the Burdenoff platform — capabilities most automation startups spend years bolting on.

| Key capabilities

- 1 Visual drag-and-drop builder**

A React Flow canvas with node palette, edge wiring, drop-a-node-on-an-edge insertion, and auto-layout. Flow validation runs before save to catch dangling nodes and bad connections.
- 2 Manifest-driven node configuration**

Every connector renders its own configuration form from its manifest as an Adaptive Card — so adding a connector never requires frontend changes, and builders self-serve.
- 3 60+ connector nodes**

Versioned, manifest-driven packages across messaging, CRM, dev tools, AI/LLM, payments, databases, Google Workspace, marketing, and cloud — each with typed operations and credential-adapter bindings.
- 4 Sixteen built-in flow-control nodes**

If, switch, loop, merge (five modes), wait, set, filter, aggregate, transform, validate, code (JS/Python), http, log, error, noop, and sub-workflow — executed natively by the workers with no network hop.
- 5 Triggers for every "when"**

API, webhook (unique URL per registration with logged, individually retryable deliveries), cron schedule, and a chat trigger with a built-in chat UI for conversational workflows. Polling and platform-event triggers are phased in.
- 6 Background execution engine**

Workflows run on horizontally-scaled, stateless workers consuming Redis Streams with competing-consumer semantics, so adding replicas adds throughput linearly. At-least-once delivery with idempotent run handling.
- 7 Per-node observability**

Live, color-coded logs on the canvas during execution; an executions list and detail view with per-node drill-down into inputs, outputs, errors, and execution time. Worker health endpoints and Prometheus metrics included.
- 8 Surgical recovery**

Typed errors distinguish a bad credential from a downstream 500 from a validation error. Cancel a run, resume a paused run, or retry a single failed node with its original inputs — no full re-run.
- 9 Versioning and lifecycle**

Every structural save can mint an immutable version under a stable workflowKey. Publish, activate (which arms schedules), deactivate, duplicate, and restore any previous version into a fresh draft — rollback without losing history.
- 10 Connections and secure credentials**

OAuth or API-key connections managed through the integrations layer and selected inside node config. Nodes reference connections by id via credential adapters; secrets are resolved server-side at run time and never serialized into a workflow.
- 11 AI sub-node pattern**

AI nodes compose: an AI Agent node accepts sub-node adapters — chat models (OpenAI/Anthropic/Ollama), memory (buffer/Redis), and tools (calculator, code, HTTP, even other workflows). An agent with tools is just another node on the canvas.
- 12 Datasink pipelines to live dashboards**

Append the BigConsole DataSink node to any workflow to push keyed output into BigConsole. Schedule-triggered extracts plus in-flow transforms turn any workflow into an ETL pipeline behind a live dashboard — ETL without an ETL team.

| How it works

- 1 Draw and save**

Build on the React Flow canvas — drag a trigger, connector, and internal nodes, then wire them. Saving serializes nodes and edges and mints a versioned WorkflowDefinition.
- 2 Trigger**

An API call, webhook, schedule tick, or chat message creates a WorkflowRun and publishes a job message onto Redis Streams.

3 Execute in the background

A worker picks up the job, loads the pinned definition version, and walks the graph node by node — internal nodes run in-process; connector nodes call their own service endpoint with credentials resolved through adapters; edge mappings transform data between nodes.

4 Fan out and complete

Parallel branches run concurrently and reconverge at merge nodes. When every reachable node settles, the run is marked completed (or failed with a typed error), and events fire for notifications and usage metering.

5 Observe and recover

Watch per-node logs in the executions view. When a node fails, open the detail, fix the config, and retry that single node — or restore a previous version into a draft and re-activate; schedules re-arm automatically.

Who it's for

Operations & automation leads

Mid-market teams replacing spreadsheet-and-email glue and brittle scripts with governed, observable automations they can read and trust.

SaaS product teams

Teams embedding white-label workflow automation in their own product via the SDK instead of building an engine, canvas, and connector catalog in-house.

Digital agencies & consultants

Builders operating client automations at scale — one workspace per client, a shared template library, and read-only embedded views handed back to clients.

Enterprise process owners

Finance, HR, customer-service, and IT teams orchestrating cross-system processes with RBAC, audit, and version rollback baked in.

Data & analytics teams

Teams using workflows as the ETL layer — scheduled extracts and in-flow transforms feeding BigConsole dashboards through datasinks.

Developers

Engineers extending the platform with custom connector nodes, sub-workflows, and code nodes when the visual palette isn't enough — every UI action is also a federated GraphQL operation.

Editions

Free

Trying it out and personal automations

- Limited workflows and monthly executions
- Full visual builder included
- Complete connector catalog and per-node observability

Starter

Small teams automating core processes

- More workflows and executions
- Custom nodes
- AI assistance allowance

Pro

Teams running business-critical automation

- Unlimited workflows and high concurrency
- Full GraphQL/REST API access
- Priority support

Enterprise

Scale, compliance, and embedding at volume

- Custom quotas and execution time
- SSO and dedicated support
- SDK / OEM white-label licensing

| Services we offer

Workflow automation consulting & process discovery — mapping manual and glue-code processes into a prioritized roadmap drawn on the canvas

Done-for-you automation implementation — we build, version, activate, and validate production workflows on your behalf

Integration engineering — wiring FluidGrids into your CRMs, payment processors, databases, messaging, and internal APIs

Custom node & connector development — versioned, manifest-driven connectors with Adaptive-Card config UIs through the validation pipeline

Embedded & white-label workflow engine — SDK integration, theming, auth/token patterns, and callbacks wired into your product

Data pipelines & live dashboards — scheduled extracts and in-flow transforms feeding BigConsole DataSinks

Migration from legacy automation — auditing Zapier/n8n/Make estates and re-architecting them as versioned FluidGrids workflows with parity testing

Managed automation & operations — monitoring runs, triaging failures, retrying broken nodes, and rolling back bad versions; plus training & enablement for your builders

| Why FluidGrids



Embed-first

The builder is a product and a component. OEM automation is a sales motion, not a special build — the app shell consumes the same SDK third parties embed, so there is one builder, not two.



Execution you can audit

Versioned definitions, runs pinned to their version, typed failures, and surgical retries. Editing a workflow never changes the meaning of an in-flight or historical run.



Workflows that feed dashboards and agents

Datasinks turn any workflow into a live dashboard pipeline for BigConsole, and the AI sub-node pattern makes output composable — not terminal.



Platform inheritance

Multi-tenancy, RBAC, billing, audit, usage metering, and notifications were running before the first workflow was drawn — every release ships enterprise-ready.



Priced on scale, not seats

Plans gate scale (workflows, executions, concurrency) while the builder, the full connector catalog, and per-node observability are available on every plan.

| By the numbers

60+

Connector nodes across messaging, CRM, dev tools, AI, payments, and databases

16

Built-in flow-control nodes executed natively by the workers

19

Trigger types — schedule, webhook, API, chat and more (phased)

5

Merge modes: append, combine, multiplex, keep-separate, choose-branch

1 npm package

@burdenoff/fluidgrids-fe-sdk embeds the entire builder in any React app

GraphQL-native

Every UI action is a federated GraphQL operation an API client can do too

| FAQ

Is FluidGrids no-code or for developers?

Both. Operators build visually on the canvas; engineers extend it with custom connector nodes, code nodes (JS/Python), sub-workflows, and full GraphQL/REST API access. It's one canvas readable by both.

Can I embed the workflow builder in my own product?

Yes — that's the headline differentiator. The @burdenoff/fluidgrids-fe-sdk package ships the same builder that powers the FluidGrids app. Embed a single WorkflowBuilder, a full FluidGridsRoot module, or bring your own Apollo client, with white-label theming, chrome control, and token-minted auth scoped to a workspace.

What happens when a workflow step fails?

The executor classifies failures with typed errors so you can tell a bad credential from a downstream 500 from a validation error. You can retry the single failed node with its original inputs, resume a paused run, or restore a previous version — no full re-run, and notifications fire on failure.

How does FluidGrids feed dashboards?

Append the BigConsole DataSink node to any workflow and it pushes keyed output into BigConsole, where parsers and widgets turn it into live KPIs, tables, and charts. A schedule-triggered workflow becomes a live dashboard feed — ETL without an ETL team.

How does FluidGrids handle security and credentials?

Auth and RBAC are enforced at the gateway edge; services never see raw tokens. Node configs reference connections by id, and secrets are resolved server-side at execution time — never serialized into a workflow. Code and edge mappings run in a locked-down sandbox screened against blocked identifiers.

Can I migrate from Zapier, n8n, or Make?

Yes. Our team audits your existing automations, re-architects them as versioned, observable FluidGrids workflows, migrates connections and credentials through secure adapters, and runs parity testing with a staged cutover so nothing silently breaks.

Start free at fluidgrids.ai, embed the builder with @burdenoff/fluidgrids-fe-sdk, or talk to our team about done-for-you automation and OEM licensing.

fluidgrids.ai · app.fluidgrids.ai

A Burdenoff Consultancy Services product · Proprietary