

LEADPIPE

THE 2026 FIELD GUIDE

The data foundation for agentic AI.

How to give AI agents the identity, activity, and intent they need to act on real buyers – and the contract, interfaces, and guardrails that keep them right.



Inside this guide

01 Software that acts needs a different kind of data.

Dashboards were read by people who filled in the gaps. Agents read every field literally — and act on it at machine speed.

02 Four layers of business context.

Who someone is, what they did, what they are researching, and which company they belong to are four different facts. Keep them separate and an agent can reason; blend them and it can only guess.

03 Identity: the key everything else hangs on.

Before an agent can reason about a buyer, it needs to know which buyer. Identity turns an anonymous session into a person and gives every other fact something to attach to.

04 Activity: what happened on your properties.

First-party behavior is the most relevant evidence an agent can cite. Give it typed counters and bounded summaries, not raw clickstreams.

05 Intent: what the market is researching.

Your site sees part of a buying journey. Orbit adds the rest: people researching your category across the web, including people who have never visited you.

06 The context contract: data an agent can't misread.

The interface between your data and your model is a contract. Write it down: what each field means, what unknown looks like, where it came from, and how fresh it is.

07 Interfaces: REST, SDK, MCP, and webhooks.

The same data, four ways in. Choose by who is driving: your backend, your application code, an agent at runtime, or an event.

08 Reference architectures.

Five patterns cover most of what teams build. Each names who does what: Leadpipe supplies facts, your system owns logic, an agent drafts, and a person approves where it matters.

09 Guardrails that scale with autonomy.

The more an agent can do, the more the data foundation has to define what it may do. Build controls into the pipeline, not just the prompt.

10 Evaluate, then expand.

Judge the foundation by whether the next decision gets better — on your own traffic, in the destination you actually use.

A Field reference

Canonical field names, the layer each belongs to, and how to read it.

B Questions and resources

What teams ask first, and where to build next.

Reading time

21 minutes

Updated

October 2026

Online edition

leadpipe.com/agentive-data-guide

Software that acts needs a different kind of data.

Dashboards were read by people who filled in the gaps. Agents read every field literally – and act on it at machine speed.

For fifteen years, go-to-market data was built for a person looking at a screen. A rep saw a company name next to a pricing-page visit and applied judgment: that one is a customer, that one is a competitor, that one is worth a call. The data could be loose because the reader was not.

Agents remove that reader. A research assistant, an AI SDR, a routing workflow, or a copilot inside your own product consumes the payload directly, decides, and moves to the next record. Whatever the data implies, the agent will do – thousands of times a day.

A model supplies reasoning. The data foundation supplies the facts it reasons about. When the facts are vague, the agent is confidently wrong at scale.

That changes what “good data” means. Coverage still matters, but so does shape: whether a field is typed, whether a missing value is distinguishable from a zero, whether observation is separated from inference, and whether the agent can tell how fresh a fact is.

THE SAME RECORD, BUILT FOR TWO DIFFERENT READERS		
	BUILT FOR A DASHBOARD	BUILT FOR AN AGENT
Identity	A name and a logo on a card	A stable key – a SHA-256 email hash – that joins identity, activity, and intent across systems
Missing data	Blank cells a person skims past	Explicit <code>null</code> values and empty arrays the agent is told never to fill
Activity	A chart of page views	Typed counters such as <code>pricingPageViews</code> and bounded session summaries with timestamps
Intent	A “hot account” badge	Topic matches with scores, per-run snapshots, and a history the agent can cite
Freshness	“Updated recently”	<code>firstSeenAt</code> , <code>lastSeenAt</code> , a frozen <code>asOf</code> , and a run date for every snapshot
Action	A person decides	Your application decides what the agent may do, and who approves it

This guide is about building the right-hand column: a data foundation an agent can stand on. It covers the four layers of business context, how to hand them to a model without losing meaning, which interface fits which system, five

reference architectures, the guardrails that keep autonomy safe, and how to evaluate the whole system on your own traffic.

NOTE

Who this guide is for

Engineers building agents, RevOps and growth teams wiring AI into the funnel, product teams embedding visitor intelligence in their own platform, and agencies running it for clients. Every chapter states the decision before the implementation, so the code is optional reading.

Four layers of business context.

02

Who someone is, what they did, what they are researching, and which company they belong to are four different facts. Keep them separate and an agent can reason; blend them and it can only guess.

L1	Identity Who is this?	LEADPIPE Identification	email emailHash jobTitle seniority
L2	Activity What did they do on our properties?	LEADPIPE Data API + session summaries	sessions pricingPageViews landingPage lastSeenAt
L3	Intent What are they researching across the web?	LEADPIPE Orbit	topics minScore topic overlap runs
L4	Company Which organization – and does it fit?	LEADPIPE Enrichment + IP to company	companyName industry companyEmployeeCount
Interfaces & controls REST API · TypeScript SDK · MCP server · Webhooks · Suppression · Page targeting			

Joined by one key – the SHA-256 email hash – and kept as separate facts all the way to the model.

Most agent projects that disappoint collapse these layers into one number. “Lead score 87” gives a model nothing it can explain, verify, or challenge. Four distinct layers give it something to reason with: this person (identity) works at a company inside our ICP (company), viewed pricing twice this week (activity), and has matched our category topics in three consecutive Orbit runs (intent).

Each layer answers a different question

LAYER	ANSWERS	EVIDENCE	CHANGES WHEN
Identity	Who visited?	A resolved person and contact fields	People change roles or addresses
Activity	What did they do here?	Sessions, pages, timing, campaigns	Every visit
Intent	What are they researching elsewhere?	Topic matches and scores per run	Each daily audience run
Company	Which account, and does it fit?	Firmographics and IP-to-company context	Headcount, funding, or ownership changes

Why the separation matters to an agent

- **Explainability.** An agent that cites “viewed /pricing twice; matched *data warehouse migration* in the last two runs” can be audited. An agent that cites a blended score cannot.
- **Independent unknowns.** Identity can be absent while company context is present. Intent can exist for someone who has never visited. Each layer needs its own empty state.
- **Different permissions.** A work email for sales follow-up, a hashed email for ad audiences, and company context for routing are different decisions with different owners.
- **Different freshness.** A visit from ten minutes ago and a topic match from last week’s run should not carry equal weight in a prompt.

VERIFIED PROFILES	INTENT SIGNALS	WEBSITES MONITORED	INTENT TOPICS
280M+	60B+	5M+	20,810

Leadpipe-published figures. Definitions: <https://www.leadpipe.com/resources/our-data/>

IN PRACTICE

A compound signal beats a bigger list

The strongest record in most workflows is the overlap: someone researching your category across the web who also visits your site. Orbit exposes that overlap directly (chapter 05), so you do not have to build the join yourself.

Identity: the key everything else hangs on.

Before an agent can reason about a buyer, it needs to know which buyer. Identity turns an anonymous session into a person and gives every other fact something to attach to.

Leadpipe Identification resolves eligible website visits into people and companies by matching deterministically against its own identity graph. The output is a person record – name, work and personal emails, phone, role, seniority, department, and employer – with on-site activity attached.

Deterministic identity is what makes autonomy safe

A probabilistic guess such as “someone from Acme, probably in marketing” is a useful hint for a human. For an agent it is a liability, because the agent will write to a real person on the strength of that guess. Deterministic identity means the person is resolved from a match rather than inferred, so the agent’s first fact is a fact.

Use the hash as your join key

Every visitor record carries the normalized primary `email` and its hashes: `emailHash` (SHA-256, also in `emailHashes.sha256`), `emailHashes.sha1`, and `emailHashes.md5`. The lowercase SHA-256 value ties the layers together – it is the `visitorId` for session summaries and the `hashedEmail` for Orbit topic history. Pass the hash between systems; pass the address only where a workflow actually needs to contact someone.

Identity and company fields in a visitor record (illustrative values)

```
{
  "email": "jordan.lee@northwind.example",
  "emailHash": "5f2b...c91e",
  "emailHashes": { "sha256": "5f2b...c91e", "sha1": "...", "md5": "..." },
  "businessEmails": ["jordan.lee@northwind.example"],
  "personalEmails": [],
  "phones": ["+1 555 0100"],
  "phonesDnc": false,
  "jobTitle": "Director of Data Engineering",
  "seniority": "Director",
  "department": "Engineering",
  "companyName": "Northwind Analytics",
  "industry": "Software Development",
  "companyEmployeeCount": 420,
  "companyTotalRevenue": null
}
```

Aliases resolve to one person

People use more than one address. The Data API represents identity aliases as one person, so an agent does not greet the same buyer twice under two emails. Hashes apply to the normalized primary `email`, not to every address in `emails` — when you join on a hash, join on that one.

Depth of record, by geography

Person-level identification is strongest on eligible US traffic. For traffic everywhere, `IP to company` returns stored company context for IPv4 and IPv6 addresses, so a visit from London or Singapore can still route to the right account. EU and UK traffic defaults to company level; person-level identification there requires affirmative consent. Design the agent to accept both depths of record and to act appropriately on each.

GUARDRAIL

Identity is not permission

A resolved identity tells you who visited. It does not by itself establish that a given use — email, calling, ad targeting — is permitted. Check `phonesDnc` before any call workflow and review each channel with the team that owns it (chapter 09).

Activity: what happened on your properties.

First-party behavior is the most relevant evidence an agent can cite. Give it typed counters and bounded summaries, not raw clickstreams.

Identity says who. Activity says what they did – and on your own site, that is the closest thing to stated interest a buyer gives you before they fill out a form. Leadpipe returns activity in the same canonical visitor record as identity, so an agent receives both together.

CATEGORY	FIELDS	WHAT AN AGENT CAN DO WITH IT
Engagement	<code>sessions</code> , <code>pageviews</code>	Separate a single bounce from a multi-session evaluation
High-intent pages	<code>pricingPageViews</code> , <code>demoPageViews</code>	Prioritize, and cite the specific behavior in a brief
Timing	<code>firstSeenAt</code> , <code>lastSeenAt</code>	Weight recency and avoid acting on stale visits
Journey	<code>landingPage</code> , <code>visitedPages</code> , <code>referrer</code> , <code>referrerDomain</code>	Explain what brought them and what they read
Campaigns	<code>utmSource</code> , <code>utmMedium</code> , <code>utmCampaign</code> , <code>utmTerm</code> , <code>utmContent</code>	Match follow-up to the message they clicked
Ad clicks	<code>gclid</code> , <code>fbclid</code> , <code>bingClickId</code> , <code>linkedinClickId</code>	Close the loop with ad platforms
Your labels	<code>customAttribution</code>	Carry client, campaign, or tenant IDs through the pipeline

Privacy-bounded session summaries

Agents rarely need a full clickstream, and you rarely want to give them one. `GET`

`/v1/data/visitors/{visitorId}/sessions` returns one bounded record per session – start and end time, duration, page-view count, and path-only entry and exit pages – without detailed events, full URLs, query strings, or fragments. It is the right default for a model's context window: enough to reason about engagement, nothing it does not need.

One session summary

```
{
  "sessionId": "ses_1vR8zPqY2dV5...",
  "version": "sv_8sH2mN6qR1vY...",
  "startTime": "2026-08-13T14:00:00.000Z",
  "endTime": "2026-08-13T14:14:00.000Z",
  "durationSeconds": 840,
  "pageViews": 4,
  "entryPage": "/pricing",
  "exitPage": "/thank-you",
  "updatedAt": "2026-08-13T14:15:00.000Z"
}
```

Three details that matter for agents

- **version** is a **change detector**. It changes whenever any field in the summary changes. Store it and re-run the agent only when it moves – cheaper and more consistent than reprocessing on a timer.
- **asOf** **freezes the window**. The first page of a traversal fixes **meta.asOf**, so a relative window such as **30d** does not shift while you paginate. Include **asOf** in the agent’s context so it knows what “recent” meant.
- **meta.coverage.partial** is a **fact, not noise**. When it is **true**, the population does not fully cover the requested window. Carry that qualification into any report or summary an agent writes.

Carry your own context through the pixel

Append query parameters to the pixel script – a partner customer ID, a campaign, a region – and Leadpipe returns them in **customAttribution** on every visitor record and webhook delivery. For a multi-tenant agent, this is how a visitor arrives already labeled with the tenant it belongs to.

Pixel with custom attribution

```
<script
  src="https://leadpipe.aws53.cloud/p/YOUR-PIXEL-ID.js?partnerCustomerId=customer_123&campaign=spring-2026"
  async
></script>
```

IN PRACTICE

Scope collection before you scope prompts

Page targeting (**includedPaths** or **excludedPaths** on the pixel) limits where collection runs at all. Excluding careers pages, support docs, or logged-in app routes keeps irrelevant activity out of every downstream prompt.

Intent: what the market is researching.

Your site sees part of a buying journey. Orbit adds the rest: people researching your category across the web, including people who have never visited you.

Most research happens before a buyer reaches your site — on publications, communities, documentation, and comparison pages. **Orbit** is Leadpipe’s person-level intent product: a monitored network of 5M+ websites organized into 20,810 topics, with active audiences refreshed daily. Instead of an account-level “surge” flag, it returns people, the topics they matched, and how strongly.

How Orbit models intent

CONCEPT	WHAT IT IS	WHY IT MATTERS TO AN AGENT
Topic	A subject people research, from a catalog you can browse, search, or derive from any website	The vocabulary of your market — and of your prompts
Intent score	Strength of a person’s match on the selected topics; filter with <code>minScore</code>	A threshold you can tune and explain
Topic overlap	How many of the selected topics a person must match	Higher overlap means a narrower, usually stronger audience
ICP filters	Company, role, job title, contactability, geography, and demographics	Fit before reach
Audience	A saved definition: topics, thresholds, and filters	Reusable, reviewable targeting logic
Run	A materialized snapshot of an audience at a point in time	The unit an agent should cite: “in the run on 21 October”

From topic to people

1. **Find topics.** Analyze a website, browse the catalog, or search it. Topic discovery is public, so an agent can explore your category before it has an API key.
2. **Preview.** Estimate audience size and inspect masked sample rows without saving anything.
3. **Save the audience** as `draft` or `active`. Active audiences refresh through scheduled daily runs.
4. **Run.** Queue a manual run with an `Idempotency-Key`, or let the daily refresh produce the next snapshot.
5. **Read.** Page through hydrated results — ordered by intent score, then topic overlap — check fill-rate stats, or export a CSV.

Build an Orbit audience with the TypeScript SDK

```
import { Leadpipe } from "@leadpipe/client";

const client = new Leadpipe({ apiKey: process.env.LEADPIPE_API_KEY });

// 1. Find topics that describe your market
const topics = await client.intent.topics.search({ q: "data warehouse", limit: 5 });
const topicIds = topics.data.map((topic) => topic.topicId);

// 2. Save an audience with fit filters, then activate it
const audience = await client.intent.audiences.create({
  name: "Warehouse migration researchers",
  config: { topicIds, minScore: 70, filters: { hasBusinessEmail: true } },
});
await client.intent.audiences.update(audience.data.id, { status: "active" });

// 3. Wait for the snapshot, then read the strongest matches
await client.intent.audiences.waitForReady(audience.data.id, { timeoutMs: 600_000 });
const results = await client.intent.audiences.results(audience.data.id, { limit: 25 });
```

Where intent meets activity

The compound signal – researching your category and visiting your site – is available as its own set of operations, deduplicated by email hash:

QUESTION	OPERATION
Which people in our latest audiences visited our site?	GET /v1/intent/audiences/website-visitors
Which members of one audience visited?	GET /v1/intent/audiences/{id}/website-visitors
Which topics has this visitor matched over time?	GET /v1/intent/audiences/website-visitors/{hashedEmail}/topic-history
What changed since the previous run?	GET /v1/intent/audiences/{id}/dashboard
How has the audience moved across runs?	GET /v1/intent/audiences/{id}/trends

The run-over-run dashboard is especially useful for agents. People added, people removed, and intent and topic movement are exactly the “what is new since yesterday” an agent needs to write a useful morning brief.

GUARDRAIL

Intent is a signal, not a verdict

A topic score describes research behavior. It is not proof that someone is ready to buy, and an agent should never phrase it that way. Have the agent reference the topic, not the observation: “teams evaluating warehouse migrations often ask...” rather than “we saw you researching...”.

The context contract: data an agent can't misread.

The interface between your data and your model is a contract. Write it down: what each field means, what unknown looks like, where it came from, and how fresh it is.

Many “hallucinations” in go-to-market agents are contract failures, not model failures. The model was handed a record with a blank title and asked for a personalized email, so it invented a title. It was handed a score with no explanation, so it explained it. The fix is rarely a better prompt. It is a better envelope.

Seven rules for agent-ready context

1. **Separate facts by layer.** Identity, activity, intent, and company go in distinct objects. Never merge them into one score before the model sees them.
2. **Make unknown explicit.** Leadpipe returns empty arrays and `null` rather than placeholders. Preserve that, list the missing fields, and instruct the model never to infer them.
3. **Separate observation from inference.** Facts come from Leadpipe; conclusions come from your agent. Store them in different places so a summary cannot be laundered back into a fact.
4. **Timestamp everything.** `lastSeenAt`, the traversal `asOf`, and the Orbit run date. “Recent” is meaningless to a model without a clock.
5. **Keep provenance.** Record which operation, pixel, segment, or run produced each fact, so every claim can be traced.
6. **Send only what the job needs.** A routing agent needs company and activity, not phone numbers. Fewer fields means less risk and lower token spend.
7. **Use canonical names.** Build on `companyName`, `companyEmployeeCount`, `phonesDnc`, and the other canonical fields. Legacy aliases exist for backward compatibility, not for new builds.

An agent-ready envelope

Here is one way to package a Leadpipe visitor for a model. It is an application-side shape you assemble from API or webhook data — not a raw Leadpipe response — and it makes every rule above visible to the model.

context.json – assembled by your application

```
{
  "subject": { "emailHash": "5f2b...c91e", "asOf": "2026-10-21T09:00:00Z" },
  "facts": {
    "identity": {
      "name": "Jordan Lee",
      "jobTitle": "Director of Data Engineering",
      "seniority": "Director"
    },
    "company": {
      "companyName": "Northwind Analytics",
      "industry": "Software Development",
      "companyEmployeeCount": 420
    },
    "activity": {
      "sessions": 3,
      "pricingPageViews": 2,
      "lastSeenAt": "2026-10-20T16:42:00Z",
      "landingPage": "/product/orbit",
      "utmCampaign": "warehouse-migration"
    },
    "intent": {
      "audience": "Warehouse migration researchers",
      "runDate": "2026-10-21",
      "topics": ["Data Warehouse Migration", "Cloud Data Platforms"]
    }
  },
  "unknown": ["phones", "companyTotalRevenue"],
  "provenance": {
    "identity": "GET /v1/data?email=...",
    "activity": "GET /v1/data/visitors/{visitorId}/sessions",
    "intent": "Orbit audience run, 2026-10-21"
  },
  "policy": { "allowedAction": "draft", "channel": "email", "approver": "account owner" }
}
```

IN PRACTICE

Put the contract in the system prompt, too

Tell the model in plain words what each object means: “ **facts** were observed by Leadpipe. **unknown** lists fields with no value – never guess them. Only **policy** decides which action is allowed.” Models follow a contract they can read.

Interfaces: REST, SDK, MCP, and webhooks.

The same data, four ways in. Choose by who is driving: your backend, your application code, an agent at runtime, or an event.

INTERFACE	BEST WHEN	DRIVEN BY	COVERS
REST API	Your backend pulls visitors, sessions, segments, Orbit runs, or company lookups	Your code	Identification, Orbit, IP to company, pixels, suppression, account
TypeScript SDK	You are building Orbit discovery and audiences into a TypeScript application	Your code	Orbit topics and audiences, with polling helpers
MCP server	An agent in Claude, Cursor, Codex, or another MCP client should explore and operate	The agent, at runtime	Topics, audiences, results, visitor data, pixels, account
Webhooks	Work should start the moment a qualifying visitor is identified	An event	Identified-visitor deliveries by segment, pixel, and trigger

MCP: give an agent the foundation directly

The [official MCP server](#) wraps the public Leadpipe APIs in a local server that MCP-compatible clients use over stdio. Install it with one command and your own API key.

Claude Desktop or Cursor – mcpServers configuration

```
{
  "mcpServers": {
    "leadpipe": {
      "command": "npx",
      "args": ["-y", "@leadpipe/mcp"],
      "env": { "LEADPIPE_API_KEY": "sk_..." }
    }
  }
}
```

Codex CLI

```
codex mcp add leadpipe --env LEADPIPE_API_KEY=sk_... -- npx -y @leadpipe/mcp
codex mcp list
```

WHAT THE AGENT CAN DO	
GROUP	TOOLS
Topic discovery	<code>list_topics</code> , <code>search_topics</code> , <code>get_topic_facets</code> , <code>get_topic_trend</code> , <code>compare_topics</code> , <code>get_topic_movers</code> , <code>analyze_website_topics</code>
Audience builder	<code>get_audience_filters</code> , <code>preview_audience</code> , <code>query_audience</code> , <code>list_audiences</code> , <code>get_audience</code> , <code>create_audience</code> , <code>update_audience</code> , <code>delete_audience</code>
Audience results	<code>get_audience_status</code> , <code>get_audience_results</code> , <code>list_audience_runs</code> , <code>get_audience_stats</code> , <code>export_audience</code>
Visitor data	<code>query_visitor_data</code>
Pixels	<code>list_pixels</code> , <code>create_pixel</code> , <code>update_pixel</code>
Account	<code>get_account_status</code>

The server also ships guided prompts – `discover-audience-topics`, `operate-saved-audience`, and `investigate-visitor-data` – plus documentation resources (`leadpipe://docs/auth`, `leadpipe://docs/workflows`, `leadpipe://docs/data-api`) the agent can read before it acts.

GUARDRAIL

Scope what the agent may change

The MCP server can create and update pixels and audiences, and delete audiences. If an agent is meant to research rather than operate, allow only read tools in your client or require approval for every write.

REST: the full surface from your backend

Every API uses the base URL `https://api.aws53.cloud` and an organization key in the `X-API-Key` header. Keys belong in a backend or secret manager – never in browser code or a pixel URL. Machine-readable OpenAPI specifications are published for [identification](#) and [intent](#).

Pull identified visitors from your backend

```
# Newest identified visitors, 50 per page
curl "https://api.aws53.cloud/v1/data?page=1" \
  -H "X-API-Key: $LEADPIPE_API_KEY"

# One person, with their session journey and summary
curl "https://api.aws53.cloud/v1/data?email=jordan.lee@northwind.example" \
  -H "X-API-Key: $LEADPIPE_API_KEY"

# Only visitors that match a saved segment
curl "https://api.aws53.cloud/v1/data/query" \
  -H "X-API-Key: $LEADPIPE_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{"segmentId": "YOUR_SEGMENT_ID", "page": 1, "limit": 50}'
```

OPERATION	USE IT FOR
<code>GET /v1/data</code>	Identified visitors, newest activity first
<code>GET /v1/data?email=...</code>	One person plus their session journey and summary
<code>GET /v1/data/visitors/{visitorId}/sessions</code>	Privacy-bounded session summaries
<code>POST /v1/data/query</code>	Visitors matching a saved segment; reads existing data without resolution credits
<code>GET /v1/data/company?ip=...</code>	Company context for an IPv4 or IPv6 address
<code>/v1/intent/topics/...</code>	Orbit topic discovery, analysis, and saved topic lists
<code>/v1/intent/audiences/...</code>	Orbit audiences, runs, results, stats, exports, and visitor overlap

Webhooks: event-driven agents

When the agent should start the moment someone is identified, use a webhook. Each one has an HTTPS destination, a segment, a pixel, and a trigger: **First Match** delivers once per qualifying visitor; **Every Update** keeps delivering as the record changes. Start with First Match on a narrow segment, make the receiver idempotent by upserting on the email hash, and watch delivery success before widening the segment.

NOTE

Envelopes differ

API responses, webhook deliveries, and Orbit run results share canonical visitor fields but use different wrappers. Build each parser from its own reference or a saved test payload.

Documentation your agent can read

Point a coding agent at the machine-readable entry points before it writes integration code: the [documentation index](#), Leadpipe's [agents.txt](#) operating brief, and the [site index](#).

Reference architectures.

Five patterns cover most of what teams build. Each names who does what: Leadpipe supplies facts, your system owns logic, an agent drafts, and a person approves where it matters.

Real-time visitor brief

A high-fit person is on your site, and sales should know why — now.

01 LEADPIPE

A First Match webhook fires for the “ICP + pricing page” segment.

02 YOUR SYSTEM

The receiver upserts on `emailHash` and fetches session summaries.

03 LEADPIPE

Topic history for the same hash adds off-site research context.

04 AGENT

Writes a five-line brief: who, company fit, what they read, what they research, a suggested angle.

05 PERSON

The account owner reviews it in Slack or the CRM and chooses the next step.

GUARDRAIL The agent drafts; a person sends. Suppress customers and open opportunities before the webhook fires.

In-market prospecting

Find people researching your category who have not visited yet.

01 LEADPIPE

An active Orbit audience refreshes on its daily run.

02 YOUR SYSTEM

Reads the latest completed run, ordered by intent score.

03 AGENT

Checks each person against your ICP notes and drafts a first touch that references the topic.

04 PERSON

Approves the batch; your outbound tool sends from its warmed domains.

GUARDRAIL Cap daily volume, honor unsubscribes and `phonesDnc`, and keep the run date with every record.

Intent + visit priority queue

Rank the people showing demand both on and off your site.

01 LEADPIPE

Returns audience members who also visited your site.

02 YOUR SYSTEM

Joins recent activity and ranks by recency, pricing views, and demo views.

03 AGENT

Explains each rank in one sentence, citing the facts behind it.

04 PERSON

Works the queue from the top.

GUARDRAIL Rank on facts you can show. If the agent cannot cite why a record ranks high, it should not rank high.

Embedded intelligence for platforms

Your product offers visitor intelligence to your own customers.

01 YOUR SYSTEM

Creates a pixel per customer through the API and appends `partnerCustomerId`.

02 LEADPIPE

Returns that label in `customAttribution` on records and webhooks.

03 YOUR SYSTEM

Routes each record to its tenant and never mixes tenants in one prompt.

04 AGENT

Runs inside one tenant's context, with only that tenant's data.

GUARDRAIL Tenant isolation lives in your code. Keep the organization key server-side.

Conversational RevOps analyst

A teammate asks questions of live data in Claude, Cursor, or Codex.

01 PERSON

"Which pixels are paused, and how did last week's identified traffic compare?"

02 AGENT

Calls `get_account_status`, `list_pixels`, and `query_visitor_data` through MCP.

03 LEADPIPE

Returns live account data under that user's own key.

04 AGENT

Answers with numbers and states what it could not see.

GUARDRAIL Read-only by default. Pixel and audience changes need explicit approval.

IN PRACTICE

Start with one pattern

Pick the pattern closest to a decision your team already makes every day, ship it in draft mode, and add a second only once the first is trusted. A stable workflow beats five demos.

Guardrails that scale with autonomy.

The more an agent can do, the more the data foundation has to define what it may do.
Build controls into the pipeline, not just the prompt.

Control what is collected and processed

CONTROL	WHAT IT DOES	USE IT FOR
Page targeting	<code>includedPaths</code> or <code>excludedPaths</code> on a pixel limits where collection runs	Keep app, careers, and support routes out
Suppression rules	Matched visitors consume no credits and trigger no webhooks or integrations	Customers, competitors, staff, and bad-fit industries
List replacement	Replaces the whole suppression list in one transaction	Sync from your CRM or another source of truth on a schedule
Industry discovery	Lists the industry values actually present in your visitor data	Suppress industries by values that really occur
Segments	Saved rules that scope webhooks and queries	Hand agents only qualified slices of traffic

Suppression accepts up to 25,000 rules per upload, and replacing the list from your CRM each night keeps “never contact our customers” true without anyone remembering to update it. Suppression governs future processing; it is not a deletion request and does not recall records already delivered.

Decide what the agent may do

AUTONOMY	THE AGENT	A PERSON	FITS
Inform	Summarizes, ranks, and explains	Decides and acts	Research briefs, morning digests, analysis
Draft	Prepares messages, CRM updates, and audiences	Approves each action	First-touch outreach, record updates
Act within limits	Executes pre-approved actions under caps	Reviews samples and exceptions	Routing, internal alerts, audience refreshes

Start anything customer-facing at Draft. Move a workflow up a level only when its acceptance rate, measured in chapter 10, shows that people rarely change what the agent proposes.

Operate safely

- **Keys.** Organization-scoped and server-side. Regenerating a key invalidates the previous one, so update every backend that uses it. Never send an `sk_` key as a Bearer token.
- **Rate limits.** Read the `RateLimit-Limit`, `RateLimit-Remaining`, and `RateLimit-Reset` headers and honor `Retry-After` on a `429`. Do not reuse one endpoint's allowance for another.
- **Retries.** Bound retries on reads. Orbit run creation accepts an `Idempotency-Key`. Do not automatically retry company lookups after an uncertain response – each company result uses allowance.
- **Logs.** Record method, path, status, and request ID. Strip keys and personal data from anything shared.
- **Contact rules.** A populated contact field is not proof of deliverability or permission. Honor `phonesDnc`, unsubscribes, and each channel's policy.

Geography and privacy review

Person-level identification is strongest on eligible US traffic. EU and UK traffic defaults to company level, and person-level identification there requires affirmative consent. Give your privacy reviewer the actual installation scope, contact fields, and destination workflow, and request Leadpipe's current privacy documentation and terms. Review email, advertising, and sales uses separately with the teams that own them.

Healthcare and other regulated industries

Healthcare technology vendors, suppliers, and service organizations use Leadpipe for commercial work: account research, business development, and marketing to the organizations they sell to. The data is business context – professionals and the companies they work for, what they read on your site, and the topics they research. It is not a clinical record, and it should never be joined with patient data or used in a clinical workflow.

- Define the commercial relationship the workflow serves before choosing data.
- Exclude patient-facing pages and portals with page targeting.
- Keep the agent's outputs inside sales and marketing systems.
- Review the workflow with your compliance team before launch.

IN PRACTICE

In practice

A healthcare technology company routes identified visitors from its provider-facing product pages to an agent that prepares account briefs for business development, while patient-facing pages are excluded from collection entirely.

Evaluate, then expand.

Judge the foundation by whether the next decision gets better – on your own traffic, in the destination you actually use.

Results from someone else's traffic say little about yours. What matters is whether, for your ICP and your workflow, the records are actionable and the people who own the next step accept what the agent produces.

Measure the data

METRIC	CALCULATION
Work-email coverage	Records with a usable work email ÷ records reviewed
Phone coverage	Records with a phone value ÷ records reviewed
ICP fit	Records matching your ICP ÷ records reviewed
Actionable records	Records meeting both your fit and required-field criteria

Review a defined sample with the same pixel, period, and filters. For Orbit, run stats report field fill rates and the intent-score distribution for each completed snapshot.

Measure the agent

CHECK	PASSES WHEN
Unknown handling	Given a record with a <code>null</code> job title, the agent never invents one
Grounding	Every claim in a brief traces to a field in <code>facts</code>
Duplicate delivery	The same webhook payload twice produces one CRM record
Partial coverage	Reports carry the <code>partial</code> qualification whenever it is present
Acceptance rate	Most drafts are approved without edits by the person who owns the action
Time to action	Minutes from identification to a reviewed next step keep falling

A 30-day rollout

WEEK	FOCUS	DONE WHEN
1	Collect. Install the pixel, set page targeting, and upload suppression for customers, competitors, and staff.	Identified visitors appear and known noise does not
2	Define. Save one ICP segment, build one Orbit audience, and choose one destination.	A sample of records passes the data scorecard
3	Draft. Turn on one architecture in Draft mode and review every output. Tune the contract and prompt.	The agent passes the unknown-handling and grounding checks
4	Decide. Score data and agent metrics with the people who use the output.	You expand, adjust, or stop on evidence

IN PRACTICE

Start on a trial

Run the first two weeks on a free trial: 7 days or 500 profiles, whichever comes first, with no card. Check [pricing](#) for current plans before you scale.

The fields an agent sees most.



Canonical names, the layer each belongs to, and how to read it. The complete schema lives in the API reference at docs.leadpipe.com.

FIELD	LAYER	HOW TO READ IT
<code>email</code>	Identity	Normalized primary email; the source of the hashes
<code>emailHash</code>	Identity	Lowercase SHA-256 of <code>email</code> ; the cross-system join key
<code>emails / businessEmails / personalEmails</code>	Identity	Arrays; empty when no values are available
<code>phones</code>	Identity	Phone numbers; no phone hashes are generated
<code>phonesDnc</code>	Identity	Do-not-call status: <code>true</code> , <code>false</code> , or <code>null</code> when unavailable
<code>jobTitle / seniority / department</code>	Identity	Role context for qualification and routing
<code>companyName / industry</code>	Company	Employer and industry; industry also drives suppression rules
<code>companyEmployeeCount / companyTotalRevenue</code>	Company	Size signals; nullable
<code>sessions / pageviews</code>	Activity	Engagement counters
<code>pricingPageViews / demoPageViews</code>	Activity	High-intent page counters
<code>firstSeenAt / lastSeenAt</code>	Activity	Recency boundaries for the visitor
<code>landingPage / visitedPages / referrer</code>	Activity	Journey context
<code>utm* / gclid / fbclid / bingClickId / linkedinClickId</code>	Activity	Campaign and ad-click attribution
<code>customAttribution</code>	Activity	Your own labels, passed through the pixel URL
<code>meta.asOf</code>	Contract	Frozen traversal boundary; include it in agent context
<code>meta.coverage.partial</code>	Contract	<code>true</code> when the window is not fully covered
<code>Session version</code>	Contract	Changes when a session summary changes; use for change detection
<code>Orbit run</code>	Intent	A dated snapshot of an audience; the unit to cite
<code>minScore / topic overlap</code>	Intent	Thresholds that make an audience narrower and stronger

Questions teams ask first.

What is a data foundation for agentic AI?

It is the layer of facts an AI agent reasons over before it acts: who a buyer is (identity), what they did on your properties (activity), what they research across the web (intent), and which company they belong to – delivered in a shape the agent cannot misread, through interfaces it can call, under rules that define what it may do.

Why not give the agent a contact database?

A contact database knows who exists. It does not know who is evaluating you right now. Identity resolved from your own site, combined with person-level intent from Orbit, tells an agent who to prioritize and why, with facts it can cite.

Should I use MCP, the SDK, or the REST API?

Use MCP when an agent in Claude, Cursor, Codex, or another MCP client should explore and operate at runtime. Use the REST API or the TypeScript SDK when your own code controls the workflow, and webhooks when work should start the moment a visitor is identified. Many teams prototype with MCP and run production on the API.

How do I stop an agent from inventing data?

Give it an explicit contract: facts separated by layer, a list of unknown fields it must never fill, timestamps, and provenance. Then test it on records with missing fields and fail any output that contains a claim you cannot trace to a fact.

Does Leadpipe work outside the United States?

Yes. IP to company returns company context for IPv4 and IPv6 addresses, so international traffic can route to the right account. Person-level identification is strongest on eligible US traffic; EU and UK traffic defaults to company level, and person-level identification there requires affirmative consent.

Can healthcare organizations use this data?

For commercial workflows, yes: account research, business development, and marketing to organizations. The data is business context, not a clinical record, and it should never be combined with patient data.

How do I try it?

Start a free trial – 7 days or 500 profiles, whichever comes first, with no card – install the pixel, and follow the 30-day rollout in chapter 10.

Build with Leadpipe

Documentation

Guides, API reference, and the Orbit API

docs.leadpipe.com/

Orbit API

Topics, audiences, runs, overlap

docs.leadpipe.com/orbit-api/introduction

TypeScript SDK

@leadpipe/client on GitHub

github.com/leadpipe-com/sdk

Our data

What the identity graph and intent network contain

www.leadpipe.com/resources/our-data/

API reference

Identification, pixels, suppression, account

docs.leadpipe.com/api-reference/introduction

MCP server

@leadpipe/mcp on GitHub

github.com/leadpipe-com/mcp

agents.txt

The operating brief for AI agents

www.leadpipe.com/agents.txt

Pricing

Plans, credits, and the free trial

www.leadpipe.com/pricing/

LEADPIPE

Build on real context.

Identity, activity, and intent your agents can act on – through the REST API, the TypeScript SDK, the MCP server, and webhooks.

Start free

7 days or 500 profiles, whichever comes first. No card.
dashboard.leadpipe.com/auth/signup

Talk to our team

cal.com/team/leadpipe/discovery

Read the documentation

docs.leadpipe.com