

Technology Reference Manual

How the system finds records, checks evidence,
and builds Land Files

7 Presentation

6 Agents

5 Services

4 Knowledge

3 Records

2 Entities

1 Accounts

0 Sources

Version 1.1 · October 15, 2026

For landowners, historians, board members, and IT professionals

Version and Snapshot Notice	5
What This Book Is and Is Not.....	6
How to Read This Book	7
Visual Conventions and Core Vocabulary	8
Part I: The Big Picture	9
1 What YourLandHistory Is and the Land File	10
1.1 A walk through a sample parcel	10
1.2 What owners can add	12
1.3 One body of knowledge, many ways to present it	13
1.4 A living file, not a one-time report	14
1.5 Who it's for	14
1.6 Where YourLandHistory sits.....	14
1.7 The promise: no unbuilt promises	14
2 The Building: Eight Layers Numbered 0 to 7	16
2.1 Why layers.....	16
2.2 A floor-by-floor tour	17
2.3 Following one request through the floors.....	18
2.4 One important note on the databases.....	19
3 Seven Rules the System Keeps	20
3.1 The rules in action	21
3.2 House style	21
Part II: How a Land File Is Made	23
4 Finding the Land Across Six Survey Systems	24
4.1 The owner's view.....	24
4.2 Four ways in	25
4.3 Six ways America described land.....	26
4.4 People as dated roles.....	26
5 The Research Plan, Polite Fetching and Record Extraction	28
5.1 The owner's view.....	28
5.2 The checklist.....	28
5.3 Polite fetching.....	29
5.4 What each fetcher retrieves.....	30
5.5 When a source changes or fails	31
5.6 Holdings: stored once, traced forever.....	31
5.7 Reading the old survey notes	31

6	Evidence Cards, Verification States and Honest Gaps.....	33
6.1	The owner's view.....	33
6.2	Anatomy of an evidence card	33
6.3	Where cards live today, and where they're going	35
6.4	The ladder of trust.....	35
6.5	When records disagree	35
6.6	Honest gaps	36
7	Writing, Checking and Sharing the Story	37
7.1	The owner's view	37
7.2	The eight story chapters	37
7.3	The writer	38
7.4	The independent checker (the red pencil)	38
7.5	Editor-locked sections	39
7.6	The reader filter and the house style	39
7.7	Staying in step with the evidence	39
7.8	Today's report is the first presentation	39
7.9	Sharing.....	40
	Part III: What Holds It Up.....	42
8	Where Things Run and the Two-Database Transition	43
8.1	Two programs at the edge	43
8.2	The two-engine reality	44
8.3	How data will move.....	44
8.4	A report build from start to finish.....	44
8.5	Manuals as code	45
9	The Three Data Classes, Storage and Privacy	47
9.1	Three classes, three rooms	47
9.2	License versus serving.....	48
9.3	What happens to a member's upload today	48
9.4	More file types, document reading, and the redaction rule (D4-v2)	49
9.5	If the project ever ends.....	50
9.6	AI providers and training.....	50
9.7	What changes with member sign-in	51
10	Accounts, Authentication and the Single API Door	53
10.1	One door.....	53
10.2	Authentication today	54
10.3	A review catch on October 15, 2026	54

10.4 Authentication planned.....	55
10.5 The agent interface in more detail	55
10.6 Security at a glance	57
11 Spend Guardrails, Safety and Release Operations	59
11.1 Spending breakers	59
11.2 Other safety locks	60
11.3 How a change reaches the live site	60
11.4 What the independent reviewer looks for	61
11.5 Hard rules for operators.....	62
Part IV: People, Agents and Key Decisions	64
12 Who Does What: Humans and Agents	65
12.1 Build and governance roles.....	65
12.2 Models at work inside each Land File	66
12.3 People in the loop	67
13 Why We Made These Decisions	68
13.1 Decisions we may revisit	69
Part V: How the Technology Will Change Over Time.....	71
14 The Road Ahead	72
14.1 An honest premise	72
14.2 Phase 1: Near term (finishing the core and opening member accounts)	73
14.3 Phase 2: Next year (partners, the Reference Desk, and developer access)	73
14.4 Phase 3: Longer term (national scale and changing AI).....	74
14.5 What a partner can build on today.....	74
14.6 How we'll handle changing AI models	74
14.7 Risks, and how we'd respond	75
Appendix A: Route and Interface Directory.....	77
Appendix B: Database Reference	80
Appendix C: State Coverage and Source Inventory	84
Appendix D: Operations Runbook	85
Appendix E: Decision Digests (ADRs 0001–0006).....	87
Appendix F: Worked Examples.....	90
Appendix G: Glossary and Component Index.....	92
Appendix H: Current Team Roster and Operating Mode (as of October 15, 2026).....	94
Appendix I: Questions Readers Often Ask.....	96
Appendix J: Keeping This Manual Current	98
Colophon.....	100

Version and Snapshot Notice

Manual version: 1.1. **Date:** October 15, 2026. **Describes:** the YourLandHistory.org website, Edition 2.0, as built on that date. **Code baseline:** the repository `dave-schooltrustlands/yourlandhistory`, main branch at commit `f12738c`.

The web edition of this manual is at yourlandhistory.org/about/technology. The Word and PDF editions are dated snapshots of it. The system changes week by week. When the live site, its capability ledger, and this book disagree, the live ledger wins. The system is built so that menus and manuals report only what is actually running, and this book follows the same rule.

What This Book Is and Is Not

This book explains how YourLandHistory works behind the screen: where its records come from, how they are checked, how a Land File is written, where information is stored, who can see it, how changes reach the live site, and where the technology is headed.

It is not a how-to for landowners. For that, read the *Quick Start Guide* and the state guides, such as the *Oregon Guide*. It is not a legal document either. The site's *Terms* and *Privacy* pages govern the legal relationship between the project and its users. Above all, a Land File is a historical research account. It is not a title-insurance policy, a boundary survey, or a legal opinion about who owns what.

How to Read This Book

The book is written for three kinds of readers at once. Pick your path:

- **The 20-minute tour (board members, historians, partners).** Read Part I (Chapters 1–3), look through the fifteen figures, and read Chapter 14, The Road Ahead. That’s enough to understand the overall flow and to see that every step has a check.
- **The founder’s path.** Read all the main text, the *Why This Choice* notes, Part IV (people and decisions), and Part V (the road ahead).
- **The operator and auditor path (IT professionals).** Read everything, including the shaded *For IT: Under the Hood* boxes, Part III, and the appendices, which hold the route directory, the database reference, the runbook, and the decision digests.

Nothing you need to understand the overall flow appears only in an IT box. Those boxes add exact names, routes, tables, and file paths for the people who will operate or audit the system.

Visual Conventions and Core Vocabulary

Status badges. Each chapter ends with a status strip. The badges mean:

Badge	Meaning
[BUILT]	Running today.
[PARTIAL]	Some of it runs today; the strip says what is done and what isn't.
[PENDING]	Small, known follow-up work that's under way.
[PLANNED]	Designed or intended but not yet built. Never described as live.

Diagram borders. In the figures, a **solid green border** means built, a **dashed border with a yellow fill** means partial or pending, and a **dotted border with a gray fill** means planned. A **double red border** marks a checkpoint, a place where the system stops something that fails a rule. The shapes stay distinct when printed in black and white.

Seven words used throughout:

- **Land File:** the living, source-cited history of one landowner's land.
- **Source:** a place records come from, such as a federal archive, a county office, or a state agency.
- **Holding:** one copy of a record the library has fetched and stored, with a note of where it came from.
- **Reading:** what someone, a person or a machine, read off a record, such as a bearing tree in a surveyor's field notes.
- **Evidence card:** one fact, with its source, a link to find it, its dates, and how far it reaches (this land, or only nearby).
- **Finding:** a fact saved with its source and a verification status.
- **ADR:** an Architecture Decision Record, a short written record of a major design decision, its reasons, and its trade-offs. They are cited as ADR-0001 through ADR-0006 (Chapter 13 and Appendix E).
- **Private paper:** anything a member adds about their own land, such as a management plan, deed, survey, map, letter, family story, or photograph. Today it arrives as an image or scan; other file types are planned. It belongs to that member.

Part I: The Big Picture

1 What YourLandHistory Is and the Land File

In short. YourLandHistory.org is the public website of America's School Trust Library. A landowner picks a state, finds their land, and gets a Land File: a history of that land told from today backward, where every sentence points to a real record. Owners can add whatever they have about their land, from management plans and deeds to letters and family stories. Today the Land File is presented as one standard report. The layers underneath are built so the same knowledge can later be presented in many other ways. The site says only what it can actually do today.

1.1 A walk through a sample parcel

Imagine the owner of a sample tree-farm parcel in Oregon. They open yourlandhistory.org and are asked one question first: *Where is your land?* Oregon and Utah are open now. Washington is next, with a target of early 2027, then North Carolina, aiming for mid-2027. The rest of the states and DC come after that.

Next, the owner finds the land. They can type a street address or a county tax account number, pick a county map and tax lot, or paste a legal description from a deed. Our sample farm is made up of several tax lots, so the owner puts them all in the *multi-lot tray*, and the site groups the lots that touch into one piece of land.

Then the system goes to work, the way a research librarian would. It works out which public records exist for that land, fetches the ones it is allowed to fetch, reads them, and turns each fact into an evidence card. A writer program drafts the story from those cards, and a separate checker program removes any sentence the cards don't support.

While the system works, the owner can add their own material. Our sample owner has a 2004 forest management plan, a copy of the deed from the family that sold them the land, a hand-drawn map of old skid roads, and a letter from the previous owner about planting the north slope. Today they photograph or scan those pages and add them to the Land File with a short caption each. Later, when the planned document reader and other file types arrive, they could upload the plan as a PDF and record the family story in their own words.

What the owner gets back is a **Land File**, presented today as one standard report:

- **The story in brief:** a short summary at the top.

- **Eight story chapters** that run from today backward:
 1. the land today and over the last fifty years;
 2. the chain of deeds and title back to the first private owners;
 3. the township's school-trust sections;
 4. the first owners, through claims and patents;
 5. the original government surveyors;
 6. the people who lived on the land before the surveyors came;
 7. the land itself: rock, soil, elevation, and water;
 8. a custom research plan for what to look up next.
- **Reading levels** for kids, adults, and PhDs, plus a choice of Brief or Full depth.
- **Evidence cards** behind every sentence, with links to the original records.
- **Honest gaps:** what wasn't found, and where to ask for more.
- **The owner's own material**, kept private to that Land File: whatever they have about the land (see "What owners can add" below).
- **A read-only Story link** for sharing with family. It hides the owner's name and never shows their private material.

That report is the first way of presenting the Land File, not the last. The same checked knowledge about the sample farm could later become a land-focused view of its soils, trees, plants, animals, and birds, or a people-focused history of the families who held it, including the previous owner's letter if the owner chooses to include it. It could also be printed as a book, published as a website, or narrated as a video. Those views and formats are planned; the section "One body of knowledge, many ways to present it" below explains why the system is built for them.

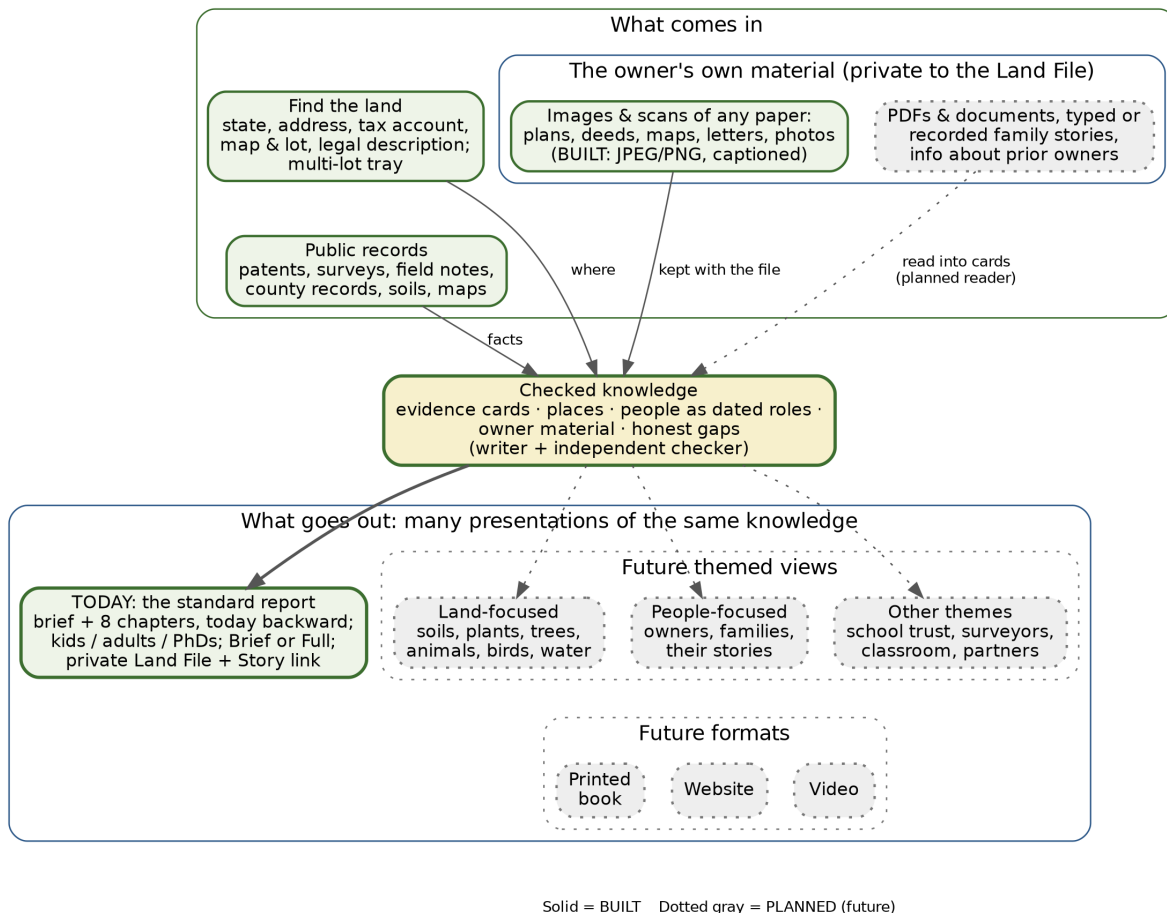


Figure 1.1: What goes into a Land File, and the many ways it can come out. Today's standard report is built; themed views and other formats are future.

1.2 What owners can add

The public record is only half of a land's history. The other half sits in desk drawers, barn offices, and family albums. The design invites owners to add **anything they have about their property**, for example:

- forest management plans, farm and conservation plans, timber cruises, and consultants' reports;
- deeds, title papers, surveys, plats, and easements;
- maps, old aerial photographs, and sketches;
- letters, diaries, and receipts;
- information about prior owners and neighbors;
- family stories and memories, and photographs of the land, its buildings, and its people.

Here is the honest line between today and the plan. **Today** the upload accepts images: JPEG or PNG images up to 8 megabytes each, up to 24 per Land File, each with an optional caption, who is in it or who wrote it, and when. Any paper can come in today as a photograph or scan, so a page of a management plan, a deed, a letter, or a map can be added now and appears in the owner's private Land File with its caption. **Planned** are other file types, such as PDF documents, word-processing files, and typed or recorded family stories, and a document reader that can read the text inside an upload and offer its facts as evidence cards (Chapters 9 and 14). Until that reader arrives, an owner's material sits beside the record-based story but doesn't yet feed it.

1.3 One body of knowledge, many ways to present it

The Land File the owner receives today is a **simple, standardized report**: the brief and the eight story chapters, at the reading level and depth the owner picks. That is the first presentation, not the only one the system is built for.

Everything under the website (the evidence cards, the people recorded as dated roles, the places, the records and the owner's own material) is stored as structured knowledge, not as finished pages. Because the knowledge is kept apart from the way it is shown, the same knowledge can be **presented in many ways**. Possibilities the design is built to support include:

- **A land-focused view:** the soils, rocks, water, plants, trees, animals, and birds of this place, and how they have changed.
- **A people-focused view:** the owners and families who lived here, their stories, and the history of the land seen through the people who worked it.
- **Other themed views:** a school-trust history, a surveyors' story, a forestry or farming history, a classroom unit, a walking tour, a story for a family reunion, or a view a partner organization designs for its members.
- **Other formats:** a printed report or book, a website, a video, and whatever forms come next.

These views and formats are **planned** (Chapter 14). The point for readers of this manual is that they don't require a new system. They are new presentations of the same checked knowledge, held to the same rules: every sentence traces to evidence, and private material stays private unless its owner chooses otherwise.

1.4 A living file, not a one-time report

A Land File is *revisable*. It can grow when the owner adds material, when a volunteer checks a reading, or when a new source comes online for that county. That isn't surveillance; the system doesn't watch anyone's land. It just means the history never stays frozen at a draft that later proves incomplete.

1.5 Who it's for

YourLandHistory is built first for **landowners**: families, tree farmers, ranchers, and homeowners who want to know the story of the ground they live on and care for. It also serves **historians and genealogists**, who want sources they can check. It serves **teachers**, who want local history their students can touch, and **woodland and watershed groups**, whose members share a landscape. Over time it aims to serve **partner organizations**, such as historical societies, landowner associations, and school-trust advocates, that want to offer Land Files to their members (Chapter 14). The design choices in this book, from the source rule to the private storage room, come from what these readers need in order to trust what they read.

1.6 Where YourLandHistory sits

YourLandHistory is one part of **America's School Trust Library** (ASTL), whose home is **schooltrusts.org**. A companion showcase, *The School Sections*, explains the lands Congress granted to the states to support public schools. A good deal of the history this system tells starts with those grants and with the government surveys that laid them out.

1.7 The promise: no unbuilt promises

The site keeps a strict rule: **say only what is live**. Its menus, manuals, and state guides draw their claims from two master lists kept with the code. The *capability ledger* says what each part of the site actually does today. The *state source registry* says which records can be reached in each state. Anything not yet working is labeled as planned, and this manual follows the same rule.

Why This Choice. Landowners, historians, and partners need to trust what the site tells them. Promising features that don't exist would spend that trust. A ledger checked against the live site keeps the promises honest. (See Chapter 3, Rule 5.)

For IT: Under the Hood. The ledgers are ledger/capabilities_ledger.json (each capability has an id, a screen, and a status such as live, live_demo_backend_planned, or in_progress) and ledger/states_config.json (sources per state, each with a mode). The eight story chapters are listed under CAP-S5-CHRONOLOGICAL-SPINE. The public site is the Cloudflare Worker yourlandhistory-web, which serves yourlandhistory.org, yourlandhistory.com, and their www hosts.

Status strip, Chapter 1

Capability	Status
State-first front door (Oregon and Utah open)	[BUILT]
Four ways to find land; multi-lot tray	[BUILT]
Today-backward story, reading levels, Brief and Full	[BUILT]
Read-only Story link that hides the owner's name	[BUILT]
Owner uploads (images and scans of any paper, with captions), private to the Land File	[BUILT]
Other upload types (PDF and other documents, typed or recorded stories)	[PLANNED]
Reading of uploaded documents (deeds, surveys)	[PARTIAL] (a demonstration screen is live; the full reader is planned)
One standard report presentation	[BUILT]
Themed views (land, people, and more) and other formats (print book, website, video)	[PLANNED]
Washington (target early 2027), North Carolina (target mid-2027), other states	[PLANNED]

New terms: Land File, multi-lot tray, story chapters, Story link, capability ledger, presentation.

2 The Building: Eight Layers Numbered 0 to 7

In short. The system is built in eight layers, like the floors of a library. Raw records arrive at the loading dock (Layer 0), and the public reads the finished work in the public rooms (Layer 7). Each floor has one job, and requests pass between floors through set doors rather than shortcuts.

2.1 Why layers

A layered design keeps each job in one place. The website never reaches straight into the stacks; it asks at the service window. A report-writing program never edits the catalog directly. That makes the system easier to check and easier to repair. It is also safer, because every request passes the same locks.

The layers are numbered from **zero** at the bottom, because Layer 0 is the outside world: the archives and offices the library depends on but doesn't own.

7 Presentation Public Rooms	yourlandhistory.org, manuals (BUILT); partner portals (PLANNED)
6 Agents Library Staff	writer, checker, survey-note reader (BUILT); Reference Desk, outside-agent keys (PLANNED)
5 Services (API) Service Window	/v1 REST + /mcp + /api/v2 on a Cloudflare Worker
4 Knowledge Reading Room	state registry, capability ledger, findings, guides
3 Records Stacks	content-addressed store built; 3 class buckets filling (PARTIAL)
2 Entities Catalog Room	places, people as dated roles: schema built, data beyond OR & UT PARTIAL
1 Accounts & Tenancy Front Desk	tables + RLS built; secret link keys live; member sign-in PLANNED
0 Sources Loading Dock	BLM GLO, CadNSDI, counties, USGS, NRCS, archives

Figure 2.1: The eight-layer architecture.

2.2 A floor-by-floor tour

Layer 0: Sources (the loading dock). The outside record keepers: the Bureau of Land Management’s General Land Office records (land patents, survey plats, and surveyors’ field notes), the national survey-grid dataset called CadNSDI, county assessors, recorders, and surveyors, state archives, the U.S. Geological Survey (USGS), the Natural Resources Conservation Service (NRCS, for soils), the National Oceanic and Atmospheric Administration (NOAA, for climate), the Census Bureau, and the Library of Congress. The library visits them politely. It sends no more than one request per second to any one website, follows each site’s posted rules for automated visitors, and records each source’s terms of use before keeping a copy.

Layer 1: Accounts and tenancy (the front desk). Who someone is and what they may see: users, organizations, memberships, roles, API keys (stored only in scrambled form), and an append-only audit log. The rule is that an individual landowner owns their private Land File by default, and an organization, such as a woodland association, can sponsor access for its members without seeing their files. The tables and the database’s ownership rules are built. Member sign-in through these tables is planned (Chapter 10). Today a private Land File opens only with its own secret link key (Chapter 9).

Layer 2: Entities (the catalog room). The *things* history is about. *Places* are parcels described in any of the six American survey traditions. *People* are recorded as dated roles tied to a source (“patentee in 1871,” “deputy surveyor in 1854”) rather than as a single “owner” field. Organizations, record repositories, and statutes live here too.

Layer 3: Records (the stacks). The documents themselves: fetched public records and members’ own material (anything an owner adds, from management plans to family letters), with the text read from them and notes on where each came from. Each file is stored once, under a digital fingerprint computed from its contents. The permanent design sorts them into three storage rooms (public, private, and donated), described in Chapter 9.

Layer 4: Knowledge (the reading room). What the library knows: the registry of sources state by state, the capability ledger, the findings, the catalog of books, and the state research guides. Reference text lives in Markdown and JSON files kept with the code. Findings and live registry data live in databases.

Layer 5: Services (the service window). The one door that requests pass through: a single small program at the network edge that offers a standard web application programming interface (API) and an interface for AI agents (Chapter 10). Before a paid AI call, today’s report path checks a kill switch and a monthly spending guard. A per-Land-File

budget reservation is built in the permanent database and will guard every paid call once the move to that database is complete (Chapter 11).

Layer 6: Agents (the library staff). The AI workers: the report writer, the independent checker, a machine reader for old survey notes, the researcher, and the planned Reference Desk (“Ask About Your Land”). The design rule is that agents use the same service door and audit trail as any other client, with no back doors to the database. Today the writer, checker, and survey-note reader run inside the backend’s report pipeline. Giving outside agents their own limited keys is planned.

Layer 7: Presentation (the public rooms). The website, the planned partner portals, and the manuals and guides. This floor holds no business rules; it only shows what the lower floors produce. Today it shows a Land File as one standard report. Because the knowledge below is kept separate from how it is shown, this floor can grow many presentations of the same knowledge: land-focused and people-focused views, other themed views, and formats such as a printed book, a website, or a video (Chapters 1, 7, and 14). This manual is itself a Layer 7 product, written in Markdown (see the Colophon).

2.3 Following one request through the floors

To see how the floors work together, follow a single request. The sample owner taps a sentence in Chapter 5 of their Land File to see its evidence.

1. **Layer 7.** The browser asks the website for the card behind that sentence. The website has no data of its own, so it passes the request along.
2. **Layer 5.** The request crosses the private line to the backend service, which checks that the request carries the Land File’s secret link key. Without the key, a private file opens nothing.
3. **Layer 4.** The backend looks up the card in the report’s knowledge: the claim, its scope, its verification status, and the holding it came from.
4. **Layer 3.** The card points to a holding, here a scanned field-note page, stored once under its fingerprint, along with its provenance.
5. **Layer 2.** The holding is tied to a place (the section corner) and to a person in a dated role (the deputy surveyor who wrote the note).
6. **Layer 0.** The card’s locator link points back to the original page at the Bureau of Land Management, so the owner can check it with their own eyes.
7. **Layer 1** is the gate that every request passes. Today that gate is the secret link key and, for data routes, the administrator key. With member sign-in, it becomes the member’s own account.

Nothing in that journey skipped a floor, and nothing reached past the service door to the data directly. That discipline is what lets a new floor-level component, such as a partner portal on Layer 7 or a new AI model on Layer 6, be added without weakening the others.

2.4 One important note on the databases

Today the live site keeps its working report data, including its evidence cards and written sections, in a fast edge database (Cloudflare D1). The permanent system of record is built: a full Postgres database with map geometry (Supabase Postgres with PostGIS). The move into it is under way. Chapter 8 explains this two-engine reality, and the rest of the book is careful to say which engine does what.

Why This Choice. Numbering the floors and giving each one a single job means a new partner portal, a new state, or a new AI model can be added on one floor without rebuilding the others. (ADR-0001, ADR-0004.)

For IT: Under the Hood. Layer 7 is the Worker `yourlandhistory-web` (static assets through the `ASSETS` binding). Layer 5 is the Worker `parcel-v2-backend`, serving `/v1`, `/mcp`, and `/api/v2`. The working report store is the D1 database `parcel-v2-db`. Layers 1, 2, and 4 have their target home in the Supabase project `astl-core` (Postgres with PostGIS, `pgvector`, `pgcrypto`, `uuid-oss`, `pg_stat_statements`, and `Vault`; `UUIDv7` keys). Layer 3 is content-addressed storage keyed by SHA-256, with three R2 buckets for the permanent classes. Layer 0 politeness is in `backend/src/fetch/polite.js` and `robots.js`.

Status strip, Chapter 2

Capability	Status
Layers 0, 4, 5, and 7 in daily use	[BUILT]
Layer 1 tables and ownership rules	[BUILT]; member sign-in [PLANNED]
Layer 2 schema for six survey systems	[BUILT]; data beyond the open states [PARTIAL]
Layer 3 content-addressed storage	[BUILT]; moving into the three class rooms [PARTIAL]
Layer 6 writer, checker, survey-note reader	[BUILT]; Reference Desk backend and outside-agent keys [PLANNED]
Partner portals	[PLANNED]

New terms: layer, source, entity, holding, service door, agent.

3 Seven Rules the System Keeps

In short. Seven rules run through every part of the system. Together they make sure the history it tells is traceable, honest about its limits, respectful of privacy, and checked by someone other than its author.

These rules are written into the project’s governing document, `AGENTS.md`. Where possible they are also written into the software, so they can’t be forgotten.

Rule 1: No source, no sentence. Every sentence in a Land File must cite a record, the page or region within it, and the record’s review status, whether that is unreviewed, checker-passed, or person-verified. Citing a source is required; having a person confirm it is the highest level of trust, not a precondition. The report checker drops uncited sentences today, and the permanent database refuses to save a finding without a source. (*Chapters 6 and 7.*)

Rule 2: Nearby is not here. A record about the neighborhood, the township, or the county is never written as if it happened *on this land*. The checker removes sentences that blur that line. (*Chapter 7.*)

Rule 3: A person’s reading beats a machine’s. When a volunteer has read a surveyor’s handwriting, that reading stands. A machine reading is labeled “*Reading needs checking*” and can never overwrite a person’s. (*Chapter 6.*)

Rule 4: Members own their private material; public records stay public. A member’s own material belongs to that member. It is kept private to their Land File, is not used to train AI models (only AI services whose terms forbid training on what we send are used), and can be erased. Public government records are licensed for open reuse under the Creative Commons Attribution license (CC BY 4.0) where the source’s terms allow. Direct public downloads and archive mirrors are planned. (*Chapter 9.*)

Rule 5: Say only what is live. Menus, manuals, and guides draw from the capability ledger and the state source registry. Planned features are labeled as planned. (*Chapter 1.*)

Rule 6: Mechanical breakers, not human gates. Instead of a person approving each expensive step, automatic switches limit spending and can stop all AI calls at once. Work stays fast without risking a runaway bill. (*Chapter 11.*)

Rule 7: The builder never grades its own work. The agent or model that builds a feature, guide, or manual doesn't approve its own release. A different reviewer checks it. (*Chapters 11 and 12.*)

3.1 The rules in action

Rules are easiest to understand through examples. Here is each one at work on the sample farm.

- **Rule 1.** The writer drafts “The farm’s oldest deed dates from 1871.” No card in the chapter carries the year 1871, so the checker drops the sentence before anyone reads it.
- **Rule 2.** A county history says settlers burned the valley’s prairies each fall. The story may say that about *the valley*. It may not say it happened *on this farm* unless a record places it there.
- **Rule 3.** The machine reads a bearing tree as “oak, 14 inches.” A volunteer reads the same line as “oak, 16 inches.” The volunteer’s reading is the one shown, and the machine’s stays filed with its label.
- **Rule 4.** The owner uploads a photo of the old homestead site, and a scan of a page from their forest management plan. Hidden GPS locations are removed, both stay inside the owner’s Land File, and neither appears on the Story link the owner shares with cousins.
- **Rule 5.** Washington is coming next, but until its sources are wired and checked, the site lists it as coming, not open, and no guide describes its records as available.
- **Rule 6.** A busy month runs up AI costs. When the monthly guard’s limit is reached, new AI calls are refused automatically, and reports wait. Nobody had to be on watch.
- **Rule 7.** The agent that drafted this manual didn’t approve it. Four different reviewers checked it against the code, and their fixes were applied before publication.

3.2 House style

The house style supports these rules. Reports use US customary units only: feet, chains, links, acres, and miles. (A *chain* is 66 feet. A *link* is one-hundredth of a chain, or 7.92 inches. These are the units the original surveyors used.) Reports are written in American English, and Native history is written as factual history, drawn from treaties, survey records, and scholarship.

Why This Choice. Historians judge a history by its footnotes. These rules give every Land File the footnotes a careful historian would demand, and they make the checks automatic instead of relying on someone to remember.

For IT: Under the Hood. Rule 1 is enforced at report time by the sentence checker in backend `/src/report/writer.js`, and in the Postgres system of record by the check constraint `chk_finding_has_source` on findings, which requires a `holding_id` or a `paper_id`. Rule 5 is the ledger pair under `ledger/`. Rule 6 is ADR-0006 (the LLM kill switch, the monthly guard, and `reserve_ai_budget`). Rule 7 is “cross-model separation of duties” in `AGENTS.md §1`.

Status strip, Chapter 3

Rule	Where enforced today	Status
1 No source, no sentence	Report checker; Postgres constraint for the permanent store	[BUILT]
2 Nearby is not here	Report checker	[BUILT]
3 Person beats machine	Readings store and register	[BUILT]
4 Member owns private material	Secret-link access, image cleaning, Forget	[BUILT]; member sign-in and document-text redaction on the server [PLANNED]
5 Say only what is live	Ledgers	[BUILT]
6 Mechanical breakers	Kill switch, monthly guard	[BUILT]; per-file and per-organization budgets in live traffic [PLANNED]
7 Builder never grades itself	Release process	[BUILT]

New terms: chain, link, separation of duties.

Part II: How a Land File Is Made

Part II follows one sample tree-farm parcel from the moment its owner types an address to the moment they share a finished story. It borrows a second picture from the land-records world: the **title-abstract examiner**. An examiner works from a checklist, keeps a chain of custody for every document, and has a second examiner at a separate desk check the work with a red pencil. (A Land File, though, is a historical account, not a title policy.)

4 Finding the Land Across Six Survey Systems

In short. The owner describes their land in whatever way they know it. The system turns that description into one or more tax lots, groups the lots that touch, and works out how the land was first surveyed, so it knows which old records to look for.

4.1 The owner's view

Our sample owner doesn't know the township, range, and section of their tree farm, and they don't need to. They know their address, and they have last year's property-tax statement on the kitchen table. They choose Oregon, type the address, and the site shows the county's tax-lot outline over a map, with the lot's map number and account number beside it.

The farm is actually three tax lots: a large wooded lot and two smaller lots along the road, which were bought at different times. The owner taps each one into the **multi-lot tray**. The site notices that all three share boundaries and offers to treat them as one piece of land with one story. A fourth lot the family owns across the valley doesn't touch the others, so the site keeps it separate. It could get a Land File of its own.

Behind the scenes, the site has now done three things. It has found the county records for each lot. It has drawn an outline around the whole holding. And it has asked which original survey that outline falls in. In Oregon, the answer is the federal rectangular survey that began at the Willamette Meridian in 1851, along with the Donation Land Claims that early settlers filed before and during that survey. That one fact decides which old records exist: township plats, surveyors' field notes, claim files, and land patents.

If one of the lots had been inside a town or a filed subdivision, the site would have noticed that too. A house lot in town shares most of its history with its neighbors, so the story would lean on the neighborhood: the subdivision plat, fire-insurance maps, and the house itself. Where no automatic source exists for those records yet, the site points to where to look.

4.2 Four ways in

The intake screens accept four kinds of description:

1. **Street address:** the easiest. It is matched to a point by the Census Bureau's address service, then to the county's tax lot. The address itself is never saved in the public holdings.
2. **Tax account number:** from a property-tax statement, looked up in the county's assessor data where the county publishes it.
3. **County map and tax lot:** the assessor's map number and lot number.
4. **Legal description:** the words in a deed, such as a township, range, and section, or a claim number.

County lookups run through an **adapter**, one per county or statewide layer. Each adapter answers the same questions: find by map and lot, by account, by point, by drawn outline, and find which lots touch a given lot. That uniform design lets the start page, the multi-lot tray, and the report builder treat every county the same way. A county without an adapter falls back to the national sources, and the site says so plainly.

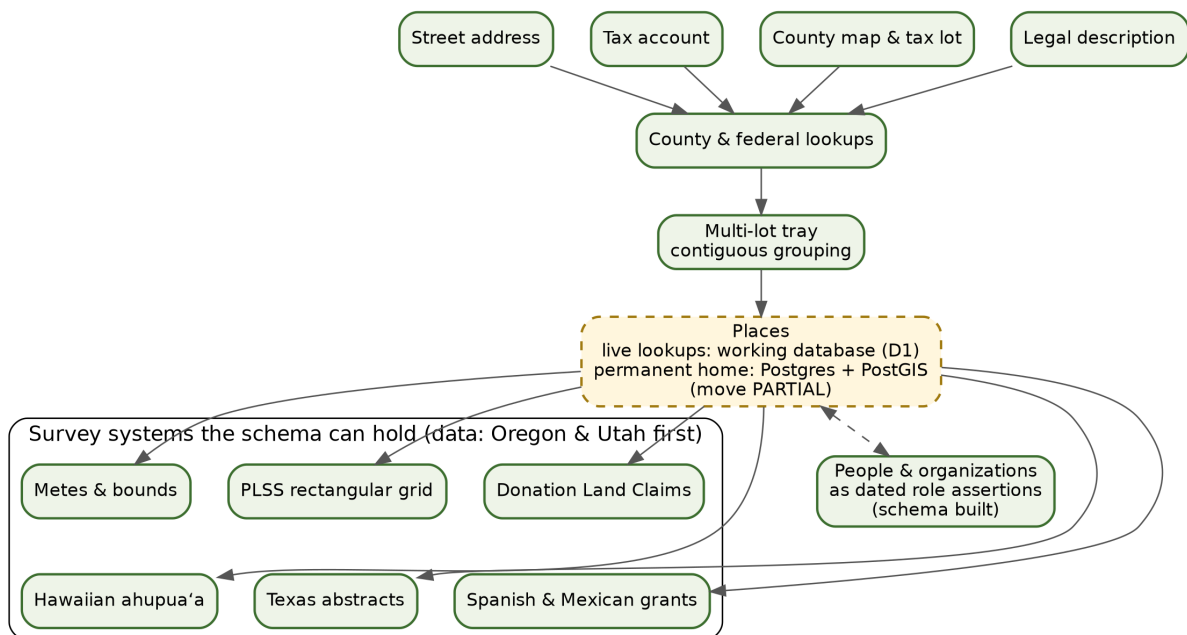


Figure 4.1: Parcel intake and the six US survey systems.

4.3 Six ways America described land

American land wasn't all surveyed the same way, so the permanent database is designed to hold all six major traditions:

- **The Public Land Survey System (PLSS):** the rectangular grid of townships, ranges, and one-square-mile sections (640 acres each) that covers most of the West and Midwest.
- **Donation Land Claims:** irregular claims granted under the 1850 Donation Land Claim Act to early settlers in the Oregon Territory, which included present-day Oregon and Washington. Each claim is identified by its number.
- **Metes and bounds:** boundaries described by distances and directions from landmarks, as in the original colonies and several eastern states.
- **Texas abstracts:** Texas's own system of surveys and abstract numbers.
- **Spanish and Mexican grants:** land grants made before U.S. sovereignty in the Southwest and California.
- **Hawaiian ahupuaʻa:** traditional land divisions running from the mountains to the sea, recorded in the Great Māhele of 1848.

Being able to *describe* a place in a system is not the same as having *data* for it. The schema supports all six today, and ten states are registered across them. Live parcel lookups currently run through the county adapters and the working report database (D1). The permanent places table in Postgres has its geometry and survey fields built and is seeded lightly while the move continues.

4.4 People as dated roles

The design doesn't store "the owner" as a single field that gets overwritten. It records people and organizations as **dated role assertions**: *this person, with the name spelled as the record spells it, was the patentee in this year, according to this record*. A parcel's history then becomes a series of sourced claims that can't erase each other, which is exactly what a historian needs. Each person also carries a living-status label based on the record dates, so living people can be treated with more care than historical ones.

Why This Choice. A single owner field forgets the past. Dated roles keep every claim with its source, so conflicting records can sit side by side until someone works out which is right. (ADR-0001.)

For IT: Under the Hood. The lookup routes on the backend are `/api/v2/lookup/address`, `/lookup/taxlot`, `/lookup/lots`, and `/lookup/town`. Address and town lookups are in `backend/src/fetchers/finder.js`. The county adapters are in `backend/src/adapters/: arcgis.js` (one configuration per county, with the same five lookups), `polk.js` and `benton.js` (tax lots plus assessor records), `owrd.js` (the Oregon Water Resources Department's statewide tax-lot outline layer, outlines and lot IDs only), and `index.js` (the registry). Grouping and town plats are in `backend/src/pipeline/town.js` and `aliquot.js`. In Postgres, `places` holds `place_key`, `survey_system`, PLSS fields, `is_school_trust_section`, PostGIS geom with validity dates, and UUIDv7 keys; `land_file_places` links places to Land Files; and `role_assertions` holds `person_name_as_written`, `role_type`, `assertion_date`, and `living_or_presumed_dead`.

Status strip, Chapter 4

Capability	Status
Four intake modes, multi-lot tray, grouping of touching lots	[BUILT] (in the working report database)
Town-lot and plat detection	[BUILT] (in Oregon counties with adapters)
Permanent schema for six survey systems and dated roles	[BUILT]
Place data in Postgres beyond the open states	[PARTIAL]

New terms: PLSS, Donation Land Claim, metes and bounds, adapter, role assertion.

5 The Research Plan, Polite Fetching and Record Extraction

In short. For each parcel, the system builds a research checklist from the state source registry. It fetches what it is allowed to fetch, politely and within each source's rules, and stores each record once with a note of where it came from. Then it reads the details out of the old survey notes.

5.1 The owner's view

Once the sample farm's outline is known, the owner sees a short "looking" screen while the system works. Behind that screen it is running a checklist much like a title examiner's. Which township is this? Who surveyed it, and when? Which claims and patents touch these sections? What do the soil and geology maps say? Is there a water right on file? Which treaty covered this ground? Which offices hold the deeds that aren't online?

For most of these, the system can go and get the record itself. The township's original plat comes from the Bureau of Land Management, as do the surveyors' field notes from the 1850s and the land patents that turned public land into private property. The soil map unit and the geologic unit come from the national services, as do the old USGS topographic maps of the area and the treaty boundary. For others, such as county deed books from the early 1900s, the system can't fetch them, but it knows which office holds them, and the research plan says how to ask.

A few minutes later, the owner's Land File has a shelf of holdings behind it: the plat, perhaps a dozen field-note pages, two patents, a soil report, a geology summary, a water-right record, several historical maps, and a treaty reference. Each one has a fingerprint, a source, a date, and a link back to where it lives.

5.2 The checklist

Every state has an entry in the **state source registry**, which lists each known source and how it can be reached. Each source gets one of five modes:

- **Automatic:** the site fetches and reads the record itself.
- **Follow link:** a checked public link the owner opens directly.

- **Ask office:** a county, state, or federal office, with whatever contact details, hours, and fees have been checked for it.
- **Found, not wired:** a public data service the team has confirmed exists, but the code to use it isn't written yet.
- **Not verified:** the source or its terms are still being checked, and the site never promises it.

From this, the system makes a **research plan** for the parcel: the steps it does itself, plus the steps the owner can take, such as which office to write to and what to ask for.

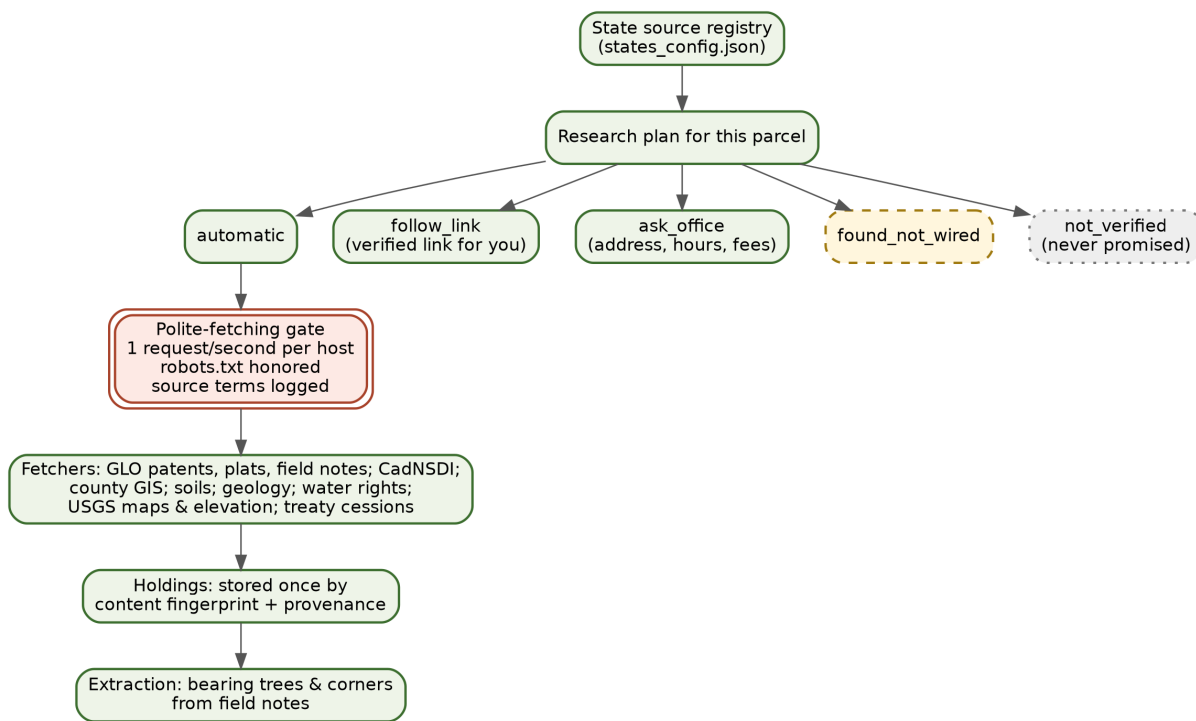


Figure 5.1: Research plan modes and the polite-fetching gate.

5.3 Polite fetching

Every automatic source passes through a single **polite-fetching gate**. It sends no more than one request per second to any one host. It reads and obeys each site's `robots.txt` file, the posted rules for automated visitors. It records the source's terms of use before keeping a copy. And it identifies itself honestly. Public archives are shared resources, and the library intends to be a good guest. Where a service's rules forbid automated queries, the system doesn't query it. One national geology service, for example, is used from a one-time download instead of its blocked point service.

5.4 What each fetcher retrieves

The fetchers are small modules, one per source:

- **General Land Office records (GLO).** From the BLM's national GLO Records site: land patents, survey plats, and field notes for the thirty public-land states. Patents name the patentee, the date, the authority (a cash sale, a homestead, a donation claim), and the land described. The system fetches them when a user's report needs them.
- **BLM Oregon land records.** The BLM Oregon State Office's township survey index for the Willamette Meridian, which covers Oregon and Washington: plats, field-note volumes and page numbers, and the historical index. This is how the system finds the exact field-note pages for the sample farm's section lines.
- **The PLSS survey grid (CadNSDI).** The BLM's national survey-grid map service. Given a point, it returns the township. Given a township, it returns every section and claim inside it. The point a user searched for is treated as private and is not filed in the holdings.
- **County sources.** County-level public data, such as climate summaries, agricultural statistics, and Census figures. Each is filed against the county that holds the township's center.
- **Polk County map.** Polk County, Oregon's public tax-lot layer, used under the county's terms, which encourage derivative products but prohibit reselling the data.
- **Soils (SDA).** The Natural Resources Conservation Service's Soil Data Access service: soil map units, their names, and their properties, filed per section.
- **State geology (SGMC).** The USGS State Geologic Map Compilation, downloaded once and cut to each township's geologic units.
- **Oregon geology (DOGAMI).** The Oregon Department of Geology and Mineral Industries' statewide geologic compilation: faults, folds, and map units, filed per section or claim.
- **Place names (GNIS).** The USGS Geographic Names Information System: named creeks, buttes, schools, and communities around the township.
- **Historical topographic maps (TNM).** The USGS National Map's historical topographic quadrangles, filed per township.
- **Treaty and cession lands (Royce).** Boundaries of tribal land cessions mapped from Royce's 1899 compilation, queried at the township's center.
- **Finder.** The address and town lookups described in Chapter 4.

5.5 When a source changes or fails

Public data services change without warning: a county moves its map server, a federal site redesigns its search, a service goes down for maintenance. The system is built to fail honestly. A fetch that fails is recorded as a failure, not silently skipped. The research plan shows the step as not yet done, and the Land File's gap box says what couldn't be reached and where to look in the meantime. Holdings already fetched keep working, because they are stored copies. A diagnostic route reports which source hosts the backend can currently reach, so an operator can tell a broken adapter from a temporary outage. When a source changes for good, its adapter is updated and the registry's "last checked" date moves forward. Until then, the source's mode can be dropped back to "follow link" or "ask office," so the site never promises what it can't deliver.

5.6 Holdings: stored once, traced forever

Each fetched record becomes a **holding**. It is stored once, under a digital fingerprint computed from its contents (a SHA-256 content hash). Two identical files always have the same fingerprint, so if two Land Files need the same plat, they share one copy, and the fingerprint proves the copy hasn't changed since it was fetched. With each holding the system records its **provenance**: the source, the item's identifier at that source, when it was fetched, under what terms, and which parser read it. If someone later questions a fact, its record can be traced back to the original.

5.7 Reading the old survey notes

When a federal surveyor set a section corner in the 1850s, he marked nearby trees as witnesses. For each one he recorded the species, the trunk's diameter in inches, its compass bearing from the corner, and its distance in links. A typical line reads something like "*a fir 24 ins dia bears N 45 E 112 lks dist.*" These **bearing trees** are some of the most vivid evidence of what the land looked like before settlement. Was it oak savanna, fir forest, or open prairie? Where were the creeks and the trails?

The system reads bearing trees out of the field notes in a pluggable step with two parts:

- A **parser** turns a transcribed line into structured fields: species, diameter, bearing (quadrant, degrees, and direction), and distance. It converts links to feet for display, so 112 links shows as about 74 feet.
- **Readers** supply the transcribed text. A **person** can type what they see on the scanned page, or a **machine reader**, an AI model that reads page images, can make a first pass.

Person readings and machine readings are stored separately and labeled with their method. A machine reading is shown as “*Reading needs checking*” and never replaces a person’s (Rule 3). The register of a corner’s trees is built from people’s readings wherever a person has keyed that corner. Machine readings appear only where nobody has read the page by hand yet, and then always with their label.

Why This Choice. Fetching once, storing by fingerprint, and logging the terms keeps costs low, keeps the archives friendly, and gives every fact an unbroken trail back to its source. (ADR-0003.)

For IT: Under the Hood. Modes are defined under `status_tags` in `ledger/states_config.json`. National sources (CADNSDI, GLO, USGS_TNM_3DEP_EPQS, USDA_NRCS_SDA, NOAA_NCEI, US_CENSUS_TIGER, ROYCE_KAPPLER) carry host, coverage, refresh interval, and storage class. Planning is in `backend/src/pipeline/plan.js`; the fetch gate is in `backend/src/fetch/polite.js` and `robots.js`; the fetchers are in `backend/src/fetchers/` (glo, blmor, cadnsdi, county, polkmap, sda, sgmc, dogami, gnis, tnm, royce, finder). Holdings go through `backend/src/kb/holdings.js` (getOrFetch by place, source, item type, and item key) and `blobstore.js`. Bearing trees are in `backend/src/extract/trees.js` (parseTreeText plus pluggable readers); readings carry method (person or llm) and, for machine readings, the model name and verification: 'needs-checking'. Admin reading tools are under `/api/v2/admin/readings` and `/api/v2/admin/machine-read`.

Status strip, Chapter 5

Capability	Status
Five source modes and a per-parcel research plan	[BUILT]
Polite-fetching gate	[BUILT]
The fetchers listed above	[BUILT]
Holdings stored once by fingerprint, with provenance	[BUILT]; moving holdings into the three class rooms [PARTIAL]
Person and machine bearing-tree readings	[BUILT]
“Found, not wired” sources	[PENDING] (adapter code, source by source)

New terms: research plan, source mode, polite fetching, holding, provenance, bearing tree, parser.

6 Evidence Cards, Verification States and Honest Gaps

In short. Every fact becomes an evidence card that says where it came from and how far it reaches. Facts climb a ladder of trust, from unreviewed to checker-passed to person-verified. Whatever the system couldn't find, it says so plainly.

6.1 The owner's view

Open any chapter of the sample farm's Land File and tap a sentence. A card slides out. Say the sentence is about a big fir at the northwest corner. The card says the fact comes from the deputy surveyor's field notes for that township, page such-and-such, written in March 1854. It gives a link to the scanned page at the Bureau of Land Management, and it says the fact concerns a corner of *this* land. If a volunteer has checked the reading against the scan, the card says *person-verified*. If only a machine has read it, the card says *Reading needs checking*.

Tap a different sentence, about a wagon road through the valley, and the card looks different. Its source is a county history, and its scope is marked *region*. The story can say that a road ran through the valley. It can't say the road crossed this farm.

Scroll to the end of the chapter and you'll find a short box titled with three plain headings: what was found, what was not found, and where more might be. For the sample farm it might say that no early deed for the smaller roadside lot was found online, that the county clerk holds the deed books for those years, and what the clerk charges for a copy. That box is as much a part of the history as the sentences above it.

6.2 Anatomy of an evidence card

An evidence card holds:

- **The claim:** one fact, in plain words.
- **The source holding:** the record it came from.
- **Locator links:** one link for each place the original can be found.
- **Dates:** when the event happened, and when the record was made.
- **Scope:** whether the fact is about *this land* or only about the region nearby.

- **Verification:** where the fact stands on the ladder of trust.

Scope matters because a township survey or a county history may describe the area without proving anything about one particular parcel. The card says which it is, and the writing step honors that (Rule 2). Numbers on cards are shown in the surveyors' own units with feet alongside, for example "59.59 chains (3,933 feet)."

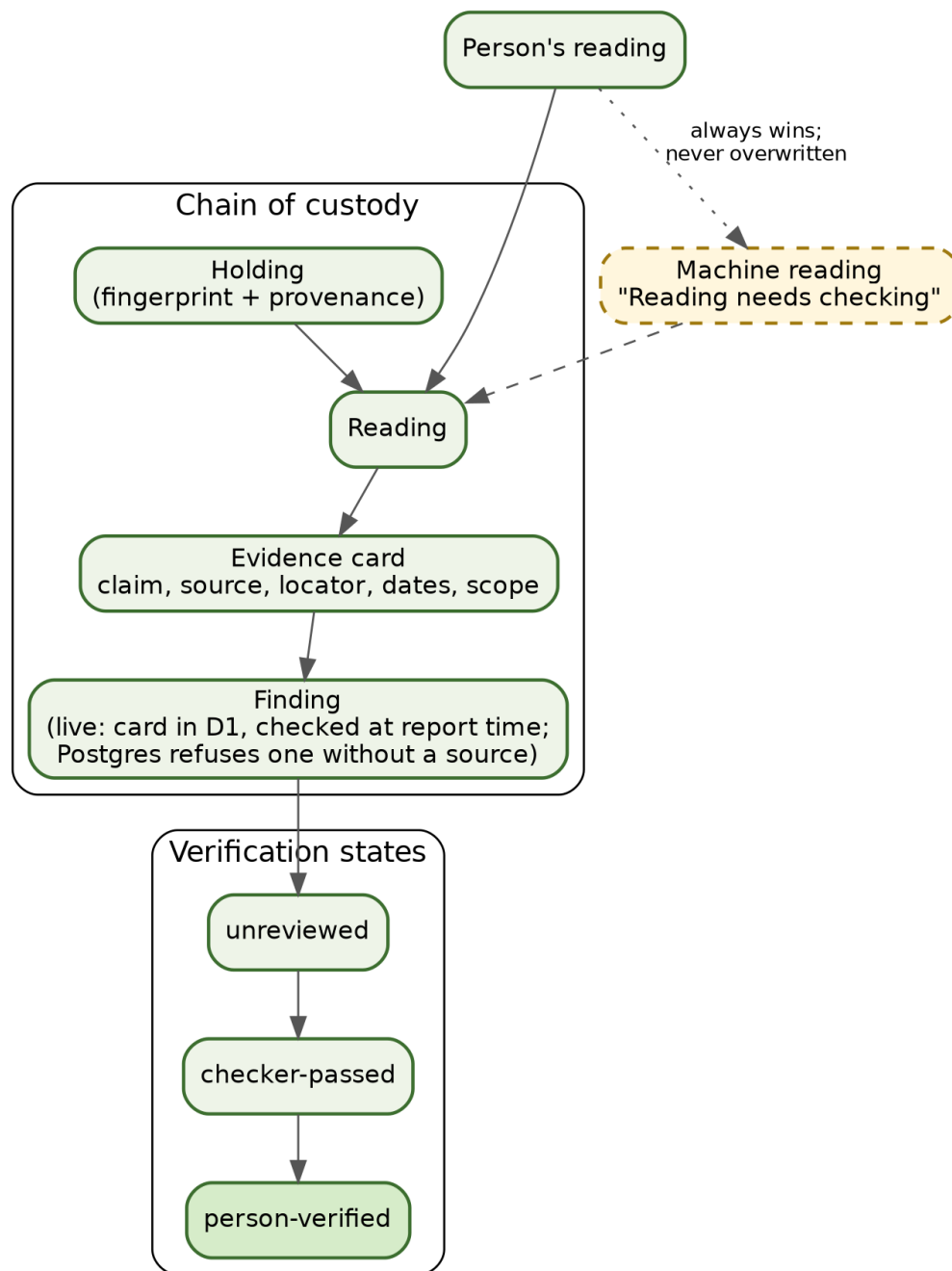


Figure 6.1: Chain of custody and verification states.

6.3 Where cards live today, and where they're going

Today the live site keeps evidence cards with each report in its working database (D1). Rule 1 is enforced at report time by the checker described in Chapter 7: a sentence that cites no card, or the wrong card, is dropped.

In the permanent Postgres system of record, a card is saved as a **finding**. There, a built-in database rule, a check constraint, refuses any finding that lacks a source, whether that source is a holding or a member's paper. A finding also carries a supporting quotation, the model and prompt version that produced it (if a machine did), a link to any finding it conflicts with, and who verified it and when. As the working data moves into Postgres, that database rule takes over the job of making unsourced findings impossible.

6.4 The ladder of trust

Each finding sits on one rung of a three-step ladder:

1. **Unreviewed:** captured with a source, but not yet checked.
2. **Checker-passed:** an automated check has confirmed that the claim matches its cited source and stays within that source's scope.
3. **Person-verified:** a person has compared it against the original record and confirmed it.

This ladder applies to *facts*. It is separate from the *sentence* check in Chapter 7, which makes sure the story's sentences match the cards. Only person verification confirms a fact against the original archive.

6.5 When records disagree

Old records often disagree. Two surveys may give different distances for the same line, an index may misspell a name that the patent spells correctly, or a county history may give a date the original record contradicts. The system doesn't hide disagreements, and it doesn't pick a winner silently. Each claim keeps its own card and source. In the permanent database, a finding can point to another finding it conflicts with. The story says plainly when the records differ ("the index gives 1871; the patent itself is dated 1872") and leaves the conflicting cards side by side, so a reader or a volunteer can settle the question against the originals. Where a person has settled it, the person-verified reading stands.

6.6 Honest gaps

A good researcher reports what they *didn't* find. Each Land File lists open questions: records not found, where more might be, and which office to ask. The system keeps checked contact details for many county offices and for tribal cultural-resource and historic-preservation offices, and it shows the date each was checked. The bearing-tree register is honest in the same way: a corner where the surveyor recorded no trees is listed as having none, not quietly dropped.

Why This Choice. Telling a reader what's missing earns more trust than a seamless story that hides its gaps, and it gives the owner a real next step. (*Rules 1 and 2.*)

For IT: Under the Hood. In the working database, cards carry `locators[]`, dates, scope, and verification labels. The bearing-tree register card records `data.register_from`, which is 'person', 'keyed corners', 'person', or 'machine', and lists machine entries separately in `data.machine_entries`, each with `verification: 'needs-checking'`. In Postgres, `findings` has `claim_text`, `holding_id`, `paper_id`, `locator`, `supporting_quote`, `conflicts_with_finding_id`, `produced_by_model`, `prompt_version`, `review_status`, `verified_by`, `verified_at`, and an embedding column for planned search. The constraint `chk_finding_has_source` and `FORCE ROW LEVEL SECURITY` apply. Review states are `unreviewed`, `checker_passed`, and `person_verified`. Open questions come from `/api/v2/open-questions`, and tribal-office contacts from `/api/v2/kb/tribal-offices` (`backend/src/data/tribal_offices.js`).

Status strip, Chapter 6

Capability	Status
Evidence cards with locators, dates, scope (working database)	[BUILT]
Postgres findings with the “must have a source” rule	[BUILT] in the schema; live cards still in D1 [PARTIAL]
Three-step ladder of trust	[BUILT]
Open questions and office contacts	[BUILT]
Volunteer verification at scale	[PARTIAL] (volunteer transcription is under way on sample registers)

New terms: evidence card, scope, finding, verification state, open question.

7 Writing, Checking and Sharing the Story

In short. A writer program turns the cards into readable prose, one section at a time. A separate checker removes any sentence that cites the wrong card, invents a number, or puts a nearby record “on this land.” Sections written by historians are locked, so the writer can never change them. The owner can share a read-only version that hides their name.

7.1 The owner’s view

When the sample farm’s records are in, the owner sees the chapters of today’s standard report fill in one after another. Each chapter opens with a few plain paragraphs. A small control lets them switch the reading level: **Kids** for a grandchild, **Adults** for most readers, or **PhDs** for the historian in the family. A second control switches between **Brief** and **Full**.

Here is how one fact from the Sullivan tree farm near Pedee reads at each level.

- **Kids:** “When surveyors first walked this land in 1854, they marked a corner by a giant fir tree. It was two feet across. That’s wider than a kitchen table!”
- **Adults:** “In March 1854 a deputy surveyor set the corner at the northwest of this land and marked a fir 24 inches in diameter, about 74 feet to the northeast, as its witness.”
- **PhDs:** “The 1854 field notes record the corner as witnessed by a fir, 24 inches in diameter, bearing N 45° E at 112 links (about 74 feet). That is consistent with the closed conifer stand the same notes describe along the north line.”

Every one of those sentences carries the same card. Change the level and the words change, but the evidence stays the same.

7.2 The eight story chapters

Each Land File has eight story chapters, ordered from today backward so the reader starts in the familiar present and walks into the past:

1. **This land today, and the last fifty years:** current use, the assessor’s record, recent aerial and map history.
2. **The title and deed bridge:** the chain of private ownership back to the first patent, where records allow.

3. **The township and its school-trust sections:** the sections granted to the state for schools, and what became of them.
4. **The first owners:** donation claims, cash entries, homesteads, and patents.
5. **The surveyors:** who surveyed the township and when, and what they saw: bearing trees, creeks, trails, prairies, and houses.
6. **Before the surveyors:** the Native peoples who lived on and used the land, drawn from treaties, survey records, and scholarship.
7. **The land itself:** rock, soil, elevation, water, and floods.
8. **Your research plan:** what to look up next, and where.

At the top sits “**The story in brief,**” a short summary written after the chapters.

7.3 The writer

The **writer** is a small, fast AI model. It writes one section at a time, plus the brief. It sees only that section’s evidence cards, and only under display names. A township key becomes plain words such as “a township in the Willamette Meridian,” and internal card IDs, accession numbers, book and file numbers, and source codes are taken out before it sees them. It must attach card references to every sentence it writes.

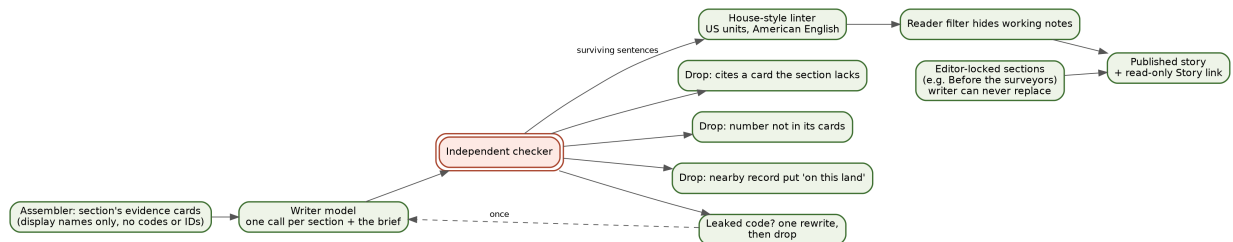


Figure 7.1: Writing and checking the story.

7.4 The independent checker (the red pencil)

A separate **checker** reads every sentence the writer produced and drops any sentence that:

- **cites a card the section doesn’t have;**
- **states a number that isn’t in the cards it cites; or**
- **puts a regional or nearby record “on this land.”**

If a sentence still contains an internal code, it goes back to the writer once to be rewritten. If the rewrite also fails, the sentence is dropped. The sentences that survive go on to the reader filter and the style linter. If nothing in a section survives, the section shows a short note instead of made-up text.

The checker confirms that sentences match the cards. It doesn't re-examine the cards against the original archives. That is the job of person verification (Chapter 6).

7.5 Editor-locked sections

Some sections are too important to hand to a machine. The best example is “**Before the surveyors,**” the account of the Native peoples who lived on and used the land. For the sample, it was drafted from treaties, the survey records, and published scholarship, including Bob Zybach's 1999 Oregon State University master's thesis on the Soap Creek Valley and the stories recorded in 1913–14. It was reviewed by editors and then **locked**. A locked section is stored exactly as approved, at each reading level. The writer never writes “Before the surveyors” and can never replace any locked text. Scholarship like the thesis is cited and quoted, never redistributed as a file.

7.6 The reader filter and the house style

Before anything is shown to the public, a **reader filter** hides working notes, such as lines marked as gaps for the editors or notes addressed to the builders. The stored copy is unchanged, and the system records what it hid. A **house-style linter** enforces US customary units (feet, chains, links, acres, and miles) and American English. Surveyors' names are shown in full only where a record gives them in full. Otherwise they appear as the index spells them, with a note saying so.

7.7 Staying in step with the evidence

Written text belongs to the specific build of evidence it came from. When the evidence is rebuilt, the old text is hidden until it's rewritten, so stale prose never sits on top of new facts. When a build finishes, all sections are written in parallel, and then the brief.

7.8 Today's report is the first presentation

The eight chapters, the brief, the levels, and the depths make up today's **standard report**. Every Land File uses the same structure, which keeps it simple to read, check, and compare.

The writer, the checker, and the evidence cards don't depend on that structure. The writer is handed a set of cards and a section to write; the checker tests each sentence against its cards.

The same machinery can therefore write other presentations from the same knowledge (planned):

- a **land view** organized around soils, plants, trees, animals, birds, and water, drawing on the land and survey cards;
- a **people view** organized around the owners, families, surveyors, and neighbors, with their stories and the owner's own family material where the owner allows it;
- other **themed views** chosen by an owner, a teacher, or a partner organization;
- different **formats**: a printed book, a website, a narrated video, or a shorter piece for sharing.

Whatever the view or format, the same rules hold. Every sentence must trace to a card, the checker still runs, and private material appears only where its owner chooses.

7.9 Sharing

The owner can share a **read-only Story link**. It removes the owner's name, never includes private material such as the owner's uploads, and can't be used to change anything.

Why This Choice. Fast AI writing makes Land Files affordable. The independent checker and the editor-locked sections make them trustworthy. Neither would be enough without the other. (Rules 1, 2, and 7.)

For IT: Under the Hood. The modules are `backend/src/report/assemble.js`, `writer.js`, and `texts.js` (levels `adults`, `kids`, `phds`, and `all`). The writer is called through `backend/src/lib/llm.js`, which applies the kill switch and spend guard. On-demand writing is `POST /api/v2/admin/write` and locked text is `POST /api/v2/admin/section-text` with `locked: true`, both requiring the admin bearer key. Rewrites are logged as `writer:<slug>:rewrite`. The reader filter records what it hid in `hidden_from_readers`. Setting the Worker variable `WRITER=off`, or turning the LLM kill switch off, stops the writer. Each section carries `text_brief` for the Brief depth when an author has filed one. Text rows carry `format: plain | markdown`.

Status strip, Chapter 7

Capability	Status
Eight story chapters, three levels, two depths, and the brief	[BUILT]
Writer with display names only	[BUILT]

Capability	Status
Independent sentence checker and one-rewrite rule	[BUILT]
Editor-locked sections	[BUILT]
Reader filter and house-style linter	[BUILT]
Read-only Story link	[BUILT]
Themed views (land, people, others) and formats (print book, website, video) from the same cards	[PLANNED]

New terms: writer, checker, editor-locked section, reader filter, linter.

Part III: What Holds It Up

Snapshot of the live infrastructure on October 15, 2026. Part III names the actual programs, databases, and storage rooms running today. It's the part of the book most likely to change, so check the live ledgers for the current migration status. It uses plain, literal terms rather than metaphors.

8 Where Things Run and the Two-Database Transition

In short. The site runs as two small programs at the network edge, one for the website and one for the backend, joined by a private line. Working report data lives today in a fast edge database. A full Postgres database with map geometry is built as the permanent system of record, and the move into it is partway done.

8.1 Two programs at the edge

Both the website and the backend run as **Cloudflare Workers**: small programs that run in Cloudflare’s data centers around the world, close to each visitor. There are no servers to patch, and capacity grows automatically with traffic.

- **The web Worker** (`yourlandhistory-web`) serves `yourlandhistory.org`: the pages, images, videos, books, and guides. It holds no business rules.
- **The backend Worker** (`parcel-v2-backend`; the name is left over from an earlier stage of the project) does the work: lookups, research plans, fetching, report building, writing and checking, and the API.

The web Worker talks to the backend over a **service binding**, a private, direct line between two Workers that never crosses the public internet. Each Worker has a **staging twin**, with `-dev` added to its name, which is hidden from search engines and used for testing before anything goes live. The staging twins talk only to each other, never to production.

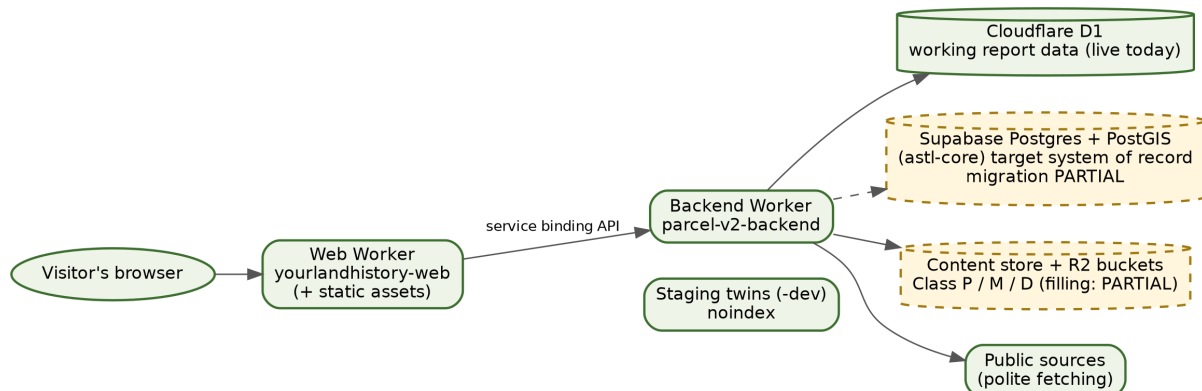


Figure 8.1: Where things run, and the two-database transition.

8.2 The two-engine reality

Two databases are in play right now, and it matters to be clear about which does what.

- **Cloudflare D1** (`parcel-v2-db`) is the working report database. Everything the live site does with reports today reads from and writes to D1: properties, lots and trays, evidence cards, readings, written sections, uploads metadata, report link keys, settings, and AI usage logs. **[BUILT]**
- **Supabase Postgres with PostGIS** (`project astl-core`) is the permanent system of record. It is built, with 16 tables, row-level security switched on for every one of them, map geometry, the pgvector extension installed for future search by meaning, and the budget circuit-breaker function. It holds the foundation schema, the state and source registries, and seed data. Moving the working data over from D1 is under way. **[PARTIAL]**

Why move? D1 is fast and simple, and it serves the live site well. But the long-term library needs real map geometry across six survey systems, row-by-row ownership rules for many members and organizations, rich relationships between people, places, and records, and a standard database that any competent IT team can operate, back up, and audit. Postgres with PostGIS does all of that. When the move is complete, D1 will remain only as a short-lived cache at the edge (ADR-0001).

8.3 How data will move

The intended approach is gradual and reversible: new permanent records (places, holdings, findings, story versions) are written to Postgres first while the live report path keeps reading from D1, and then the live path switches over one area at a time, each switch checked on staging before it reaches production. Nothing is deleted from D1 until the Postgres copy has been verified. This manual will be updated as each area moves.

8.4 A report build from start to finish

Here is the order of events when the sample owner presses “make my Land File,” with roughly where each step happens:

1. **The website** sends the chosen lots to the backend over the private line.
2. **The backend** creates the report record in the working database, issues the report’s secret link key (keeping only its fingerprint), and starts a build job.
3. **The planner** reads the state source registry and lays out the steps.

4. **The fetchers** run through the polite-fetching gate. New records become holdings in the content-addressed store, and already-held records are reused.
5. **The extractors** read structured details, such as bearing trees, corners, soil units, and patent rows, into cards, each labeled with its method and verification.
6. **The assembler** sorts cards into the eight story chapters and the evidence panels.
7. **The writer and checker** write each chapter in parallel, and then the brief. Every AI call passes the kill switch and the monthly guard and is logged with its cost.
8. **The report** is ready. The owner's browser holds the link key, and the build is recorded so later rebuilds can tell which text belongs to which evidence.

The report page shows progress while the build runs. If a written section isn't ready when the owner opens the report, the next visit carries the writing forward, a couple of sections at a time.

8.5 Manuals as code

The project's manuals and state guides are written in Markdown and kept in the same repository as the code. A build script uses Pandoc to turn each one into a Word document, a Large Print Word document, and a PDF. So far it builds the Quick Start Guide and the Oregon Guide. This manual was produced the same way, from Markdown, through Pandoc, into Word, Large Print Word, PDF, and the web page. One source feeds every format, so the formats can't drift apart (ADR-0005).

Why This Choice. Running at the edge keeps the site fast and inexpensive anywhere in the country. Moving to Postgres gives the library a durable, standard database that any competent IT team can operate. (ADR-0001, ADR-0005.)

For IT: Under the Hood. Production `yourlandhistory-web` serves the `yourlandhistory.org` and `yourlandhistory.com` custom domains and their `www` hosts. Its static assets are in `web/public` (bound as `ASSETS`), and its service binding API points to `parcel-v2-backend`. Staging is `yourlandhistory-web-dev`, bound to `parcel-v2-backend-dev`. The backend's D1 binding is `DB` (`parcel-v2-db`, or `parcel-v2-db-dev` on staging), with migrations in `backend/migrations` (0001–0012). Knowledge-base migrations 0019–0022 are applied on staging only, and production gets them through `ops/PROMOTE_KB.sh`. The Postgres extensions are `pgcrypto`, `uuid-oss`, `postgis`, `vector`, `pg_stat_statements`, and `supabase_vault`, and the schema is `supabase/migrations/0001_astl_foundation.sql`. The backend reaches Supabase over HTTPS (PostgREST) with a server-only key. Docs are built with `node scripts/build-docs.mjs`, from `content/` into `dist/docs/`.

Status strip, Chapter 8

Capability	Status
Web and backend Workers, staging twins, service binding	[BUILT]
D1 working report database	[BUILT]
Postgres system of record (schema, row-level security, functions)	[BUILT]
Moving working data from D1 to Postgres	[PARTIAL]
Search by meaning (pgvector installed)	[PLANNED]
Markdown-to-Pandoc document pipeline	[BUILT]

New terms: Cloudflare Worker, service binding, staging, D1, Postgres, PostGIS.

9 The Three Data Classes, Storage and Privacy

In short. The design sorts every stored file into one of three storage rooms: public records, members' private material, or donated heritage. Members can add anything they have about their land. Today it arrives as images (photographs and scans of papers), stripped of hidden location data, kept private to the Land File's secret link, and erasable. Other file types and reading the text inside documents are planned.

9.1 Three classes, three rooms

The permanent design classifies each file when it arrives and stores it in its own Cloudflare R2 bucket, a digital storage room:

- **Class P, public holdings.** Government records fetched for a user, where the source's terms allow keeping a copy and the people named are historical, plus state guides and public sample reports. Licensed **CC BY 4.0** with attribution to America's School Trust Library, except where a county's or source's terms restrict it.
- **Class M, member private material.** Anything an owner adds about their land: management and forestry or farm plans, reports, deeds, title policies, surveys, tax statements, maps, letters, information about prior owners, family stories, photographs, private notes, and, in time, Reference Desk conversations. Today these arrive as images and scans; other file types are planned. Owned by the individual member, kept private, never shown on Story links, and erasable.
- **Class D, donated public heritage.** Papers, stories, and checked transcriptions that members choose to give to the public under an open contribution license.

All three storage rooms exist and are connected to the backend. Moving the working collection into them is part of the move described in Chapter 8. Today the working holdings and uploads sit in the backend's content-addressed store.

	Class P Public holdings	Class M Member private	Class D Donated heritage
What	Government records, guides, sample reports	Anything owners add: plans, deeds, maps, letters, stories, photos (images & scans today; other file types planned)	Papers & transcriptions given to the public
Owner	The public commons	The individual member	Released by the donor
License	CC BY 4.0	Private; never used for model training	Open contribution license
Bucket	astl-class-p-public	astl-class-m-private	astl-class-d-donated
On Story links?	Yes	Never	When donated for public use
Served publicly today	Not yet (planned)	Never; secret link only	Not yet (planned)
Mirrors	Zenodo, Internet Archive, OSU Libraries (planned)	None	Planned, with public serving

Figure 9.1: The three data classes.

9.2 License versus serving

A license says what people *may* do with something. Serving is whether the files are actually published for download. The Class P license is set. Direct public downloads from the storage rooms aren't switched on yet, and the planned quarterly mirrors to Zenodo, the Internet Archive, and Oregon State University Libraries haven't started. For now, public records reach people through Land Files and through links back to the original sources. Class D follows the same path: it will be served and mirrored once public serving opens.

9.3 What happens to a member's upload today

The intent is that an owner can add anything they have about the land: a forest management plan, a farm plan, a consultant's report, a deed, a survey, a map, a letter, notes about prior owners, a family story, or a photograph of an old barn or a boundary marker.

Today the live upload accepts **images**: JPEG or PNG files up to 8 megabytes each, up to 24 per Land File. That covers photographs and also scans or phone pictures of any paper, so a page of a management plan or a deed can be added now. Each upload can carry a short caption, who is in it or who wrote it, and when. When an upload arrives:

1. The backend checks its size and type.
2. It **strips the hidden information** inside the image file, including the GPS location where it was taken. The date it was taken is kept.
3. It stores the cleaned file, linked only to that Land File, marked as the owner's own material and not public.

4. The upload is shown only to someone who opens that Land File with its **secret link key**. The system keeps only a fingerprint of that key, not the key itself, so a Land File's web address alone opens nothing. The public sample reports are the only exceptions.
5. The owner can delete any upload, or ask the system to **forget** the whole Land File, which erases its uploads, plans, versions, and records.

9.4 More file types, document reading, and the redaction rule (D4-v2)

Two steps are planned. The first is accepting more kinds of files, such as PDF documents, word-processing files, and typed or recorded family stories. The second is **reading** what owners upload, so that facts in a management plan, deed, survey, or tax statement can become evidence cards in the owner's private Land File. The Edition 2.0 site shows how it will work on its "Add your land papers" screen. That screen runs the redaction check in the visitor's browser as a live demonstration. The rule it demonstrates, called D4-v2, is deliberately narrow:

- **Images:** remove the GPS location hidden in the image. This is done on the server today.
- **Text:** redact Social Security numbers, driver's license numbers, bank routing numbers, and valid payment-card numbers before any AI model reads the text.
- **Keep:** land identifiers a history needs, such as tax account numbers, title-policy numbers, and loan numbers.
- **Never** lock the owner out of seeing their own page.

Enforcing the text redaction on the server, on every path where a model reads a document, comes with the full document reader, which is planned alongside member sign-in.

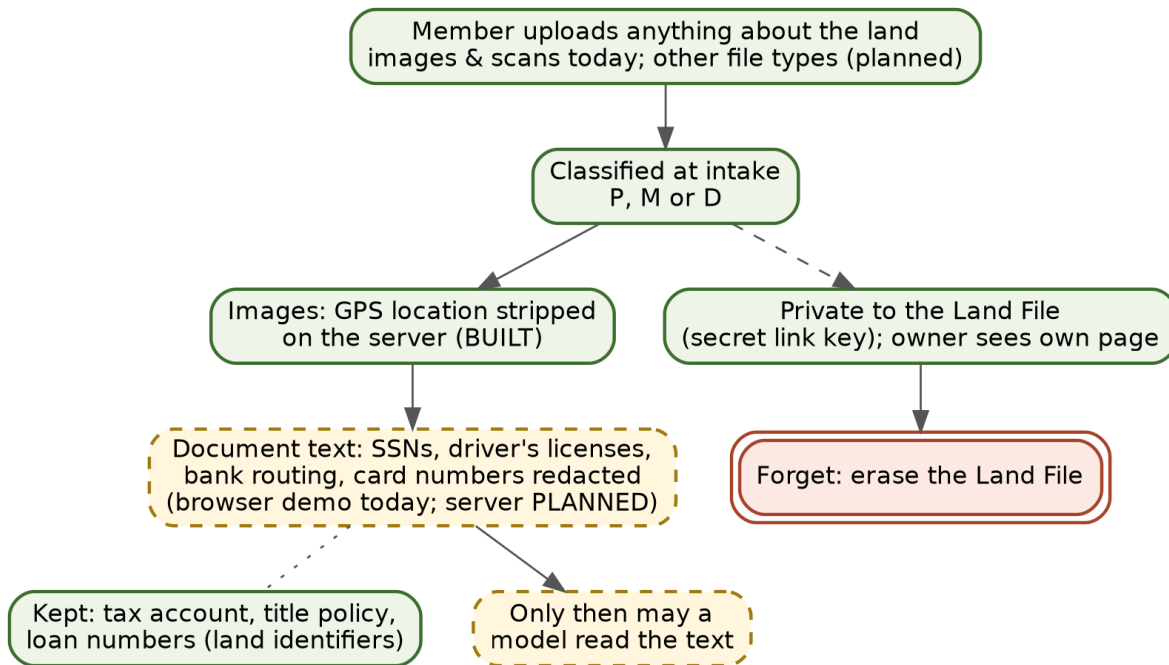


Figure 9.2: Intake redaction and privacy.

9.5 If the project ever ends

The About page makes promises about the long term, and the architecture is built to keep them. Owners can download what is in their files. The project commits to at least twelve months' notice before any closing or transfer. Private files are never sold or handed over without the owner's consent. Material that owners choose to publish carries an open license and is meant to be deposited with a named successor archive. Content-addressed storage, a standard Postgres database, open licenses on public records, and planned public mirrors all make those promises practical to keep.

9.6 AI providers and training

The rule for any private material is that it goes only to paid AI services whose terms forbid training on what we send. The site's code keeps a short list of allowed providers and refuses requests to any model not on it. The Privacy page names the current provider and its retention terms.

9.7 What changes with member sign-in

Today's protections rest on the secret link key, image cleaning, Story links that hide names, and erasure. With member sign-in, each Land File and paper is tied to a signed-in owner, the database's row-level security filters every request to that owner, private files are handed out only through short-lived signed links, and owners get self-service export, a named successor, and **Forget me** for their whole account. Cloudflare R2 encrypts stored data at rest by default.

Why This Choice. Members will trust the library with family papers only if the papers stay theirs. Public records grow in value when everyone can reuse them. Keeping the two in separate rooms with separate rules serves both. (ADR-0002, ADR-0003.)

For IT: Under the Hood. The buckets and bindings are `astl-class-p-public` (`R2_CLASS_P_PUBLIC`), `astl-class-m-private` (`R2_CLASS_M_PRIVATE`), and `astl-class-d-donated` (`R2_CLASS_D_DONATED`). As of October 15, 2026 all three have `r2.dev` access off and no custom domains. Photo uploads go to `POST /api/v2/property/:id/uploads` (multipart field photo, 8 MB limit) and are cleaned by `cleanImage()` in `backend/src/lib/imgclean.js`, then stored in the blob store under `uploads/<property>/<upload>`. The uploads table records `permission: 'owner'` and `public_ok: 0`. Report link keys are stored only as SHA-256 hashes (`properties.token_hash`, migration 0011), enforced when `REPORT_TOKENS=on`, with `PUBLIC_REPORT_IDS` listing the public samples. Erasure is `POST /api/v2/property/:id/forget` → `backend/src/report/forget.js`. The D4-v2 browser check is in `web/public/argon/`. In Postgres, papers carries `pii_redacted`, `exif_stripped`, `reader_model`, and `rights_holder`, with `FORCE ROW LEVEL SECURITY`.

Status strip, Chapter 9

Capability	Status
Three R2 storage rooms	[BUILT]; moving the collection in [PARTIAL]
Class P license (CC BY 4.0)	[BUILT]
Public downloads and archive mirrors	[PLANNED]
Image uploads (photographs and scans of any paper) with GPS stripped, private to the secret link	[BUILT]
Other file types (PDF, documents, typed or recorded stories)	[PLANNED]
Forget a Land File	[BUILT]
D4-v2 text redaction	[PARTIAL] (live demonstration in the browser; server enforcement planned)

Capability	Status
Document reader for deeds, surveys, statements	[PLANNED]
Member sign-in, signed links, self-service export and successor	[PLANNED]

New terms: R2 bucket, Class P, M, and D, CC BY 4.0, secret link key, redaction, Forget.

10 Accounts, Authentication and the Single API Door

In short. Every request goes through one door, the backend service. Today, the sensitive parts of that door are locked with a long administrator key, and each private Land File opens only with its own secret link key. Member sign-in, per-member database rules, and limited keys for partners and agents are planned.

10.1 One door

The backend offers three entrances, all in the same program:

- **/v1:** a standard web API described by an OpenAPI 3.1 contract, a machine-readable menu of every route that tools can read automatically.
- **/mcp:** an interface for AI agents using the Model Context Protocol (MCP), so an agent can discover and call the same functions. It offers seven tools.
- **/api/v2:** the older routes the current website uses for lookups, reports, and uploads.

Website screens, future partner portals, and AI agents all come through this one door. Behind it, the same queries, permissions, and cost controls apply. There are no back doors into the database (ADR-0004).

No back doors into the database

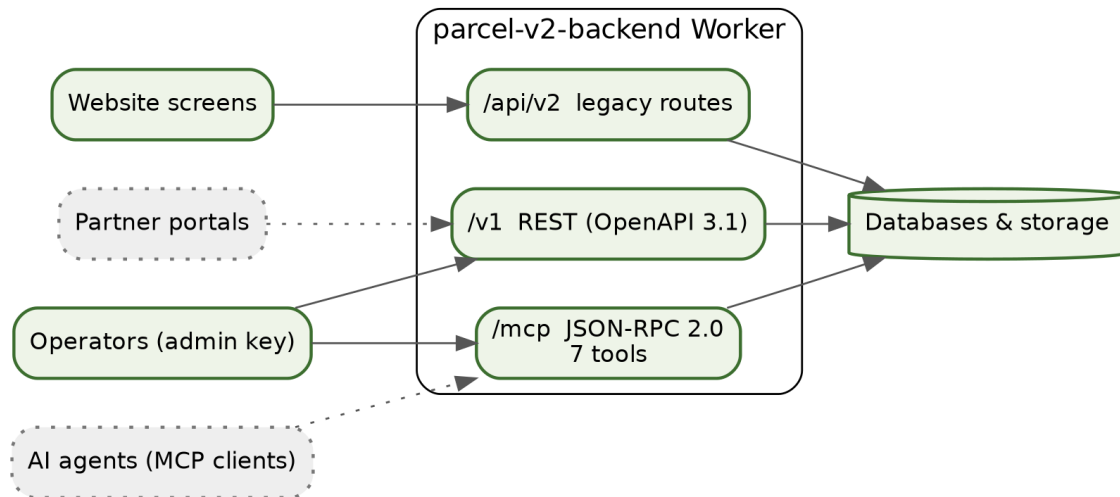


Figure 10.1: The single API door.

10.2 Authentication today

- **Public pages and report lookups** are open, as a public library should be. They are protected by the spending breakers in Chapter 11, and private reports are protected by their secret link keys (Chapter 9).
- **The /v1 data routes and /mcp** require a long administrator key, compared in a way that doesn't leak timing clues. Only basic information routes (the API's description, its health, and its contract) are open.
- **Admin routes** on /api/v2 (writing, locking text, loading holdings, switching AI off) also require an administrator key.
- **Secrets** are kept only in Cloudflare's secret storage for each Worker, never in the code repository.
- **The backend connects to the permanent database** with a server-only key that never leaves the backend.

10.3 A review catch on October 15, 2026

On October 15, 2026, the new /v1 and /mcp routes were briefly reachable without a login check. The coordinator found this in review and closed it the same day by requiring the administrator key on every data route. The fix was deployed to production and committed to the repository, so future deploys build from the corrected code. This is the review process doing its job: a problem found and closed on the day it appeared.

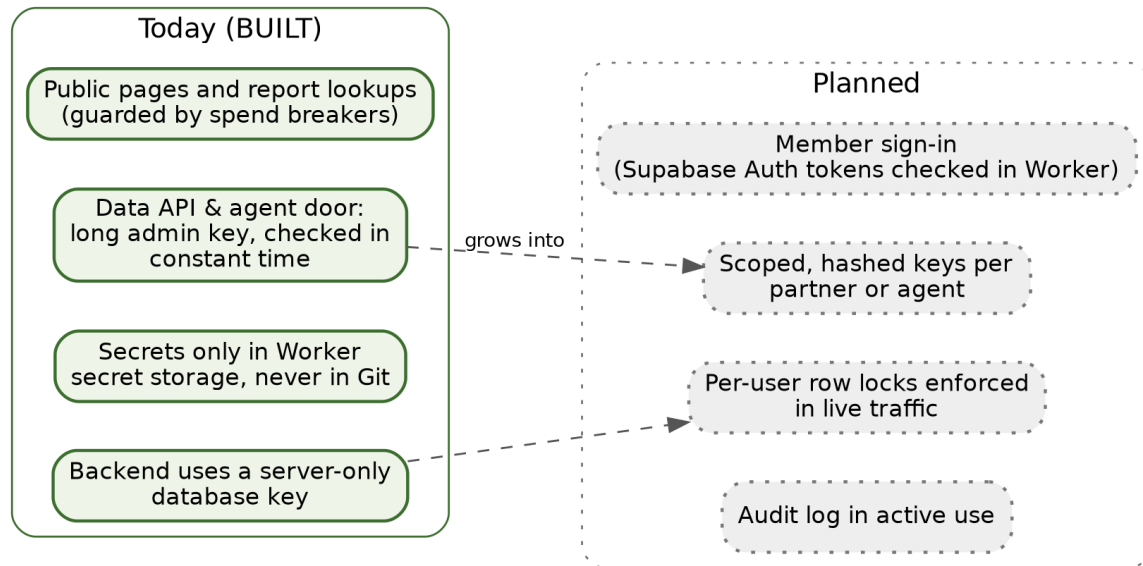


Figure 10.2: Authentication today versus planned.

10.4 Authentication planned

- **Member sign-in.** Members sign in through Supabase Auth, and the backend checks each member's sign-in token on every request.
- **Per-member row filtering in live traffic.** The database's row-level security, already switched on for every table and forced on Land Files, papers, and findings, filters each request to the signed-in member's own rows. Today the backend uses a server-only key that bypasses these rules, so per-member filtering begins when sign-in arrives.
- **Limited keys.** Partners and agents get their own keys, stored only in scrambled (hashed) form, each limited to what that client needs and to a monthly spending limit. Keys can be revoked and set to expire.
- **An active audit log.** An append-only record of who did what, including which agent acted and what it cost, in daily use.

10.5 The agent interface in more detail

For an IT reader or a future partner, here is how the agent interface works today.

The envelope. /mcp speaks JSON-RPC 2.0. A client sends a POST with a JSON body containing `jsonrpc: "2.0"`, an `id`, a `method`, and `params`. The methods are `initialize` (it returns the server's name, version, and capabilities), `ping`, `tools/list` (it returns the seven tools and

their input schemas), and `tools/call` (it runs one tool with `name` and `arguments`). Each call is independent. The server keeps no session, so any number of copies can answer requests. Results come back as `result.content` containing the tool's JSON output. Errors use standard JSON-RPC error objects, and a request without a valid administrator key gets an HTTP 401 before any tool runs.

The seven tools.

Tool	Inputs	Returns
<code>list_states</code>	optional <code>status</code> (open, partial, coming_soon)	States from the registry, with survey system, meridians, and status
<code>find_land</code>	any of <code>place_key</code> , <code>state_code</code> , <code>county_name</code> , <code>plss_township</code> , <code>plss_range</code> , <code>plss_section</code> , <code>is_school_trust_section</code>	Matching places
<code>get_place</code>	<code>id</code> or <code>place_key</code>	One place with its survey fields and geometry summary
<code>list_holdings</code>	optional <code>place_id</code> , <code>source_id</code> , <code>sha256_hash</code>	Public holdings with their fingerprints and provenance
<code>get_land_file</code>	<code>id</code>	Land File metadata and its cited findings
<code>get_findings</code>	optional <code>land_file_id</code> , <code>place_id</code> , <code>review_status</code>	Source-cited claims with review status
<code>reserve_ai_budget</code>	<code>land_file_id</code> , <code>estimated_cost_usd</code>	Whether the reservation fits under the per-file and per-organization ceilings

The REST twin. Each tool has a matching `/v1` route (see Appendix A), described in `/v1/openapi.json`, so the same functions can be called from any ordinary web client or a generated software development kit (SDK). Both doors read from the permanent Postgres database. While the move from D1 continues, they show the Postgres collection, not the live report data.

Status codes. 200 for success, 400 for a malformed request, 401 for a missing or wrong key, 404 for an unknown route or record, and 5xx if the database can't be reached. Every response carries headers that tell search engines not to index it.

10.6 Security at a glance

Concern	Today	Planned
Who can open a private Land File	Holder of its secret link key (fingerprint stored, never the key)	Signed-in owner, plus explicit shares
Who can call the data API and agent interface	Holder of the administrator key	Partners and agents with limited, hashed, revocable keys
Who can change report text or switch AI off	Holder of the admin key	Same, with every action in the audit log
Where secrets live	Cloudflare Worker secret storage only	Same
What the database enforces	Row-level security on every table (no ordinary access without policies); a source required for every finding	Per-member policies in live traffic
What protects owner uploads	Hidden GPS removed; private to the Land File; never on Story links	Signed short-lived links
What protects document text	Redaction rule demonstrated in the browser	Server-side redaction before any model reads it
How problems are caught	Staging first, independent review, smoke tests	Plus automatic staging deploys and active audit review

Why This Choice. One door means one set of locks to get right and one audit trail, and that same door can later serve the website, partners, and AI agents equally. (ADR-0004.)

For IT: Under the Hood. The door is implemented in `backend/src/api/v1_mcp.js` (`handleV1AndMcp()`) and tested in `backend/test/v1_mcp.test.mjs`. The open routes are `GET /v1`, `/v1/health`, `/v1/openapi.json`, and `CORS OPTIONS`. Every other `/v1` route, and `GET` or `POST /mcp`, requires `Authorization: Bearer <administrator key>`, compared in constant time, with a key of at least 32 characters. The administrator key and the backend's database key are Worker secrets, set with `npx wrangler secret put`. Admin `/api/v2/admin/*` routes require their own admin bearer key. The planned design verifies Supabase Auth JWTs in the Worker and uses the public anon key together with the member's JWT, so row-level security filters per member, while budget reservation stays server-only. The Layer 1 tables are `users`, `orgs`, `org_memberships`, `api_keys` (`key_prefix`, `key_hash`, `scopes`, `monthly_limit_usd`, `expires_at`, `revoked_at`), and `audit_log` (`actor_user_id`, `actor_agent`, `action`, `target_type`, `target_id`, `cost_usd`).

Status strip, Chapter 10

Capability	Status
Single door: /v1, /mcp, /api/v2	[BUILT]
Administrator-key lock on data routes and /mcp	[BUILT]
Secret link keys for private Land Files	[BUILT]
Secrets only in Worker secret storage	[BUILT]
Row-level security switched on in the schema	[BUILT]
Member sign-in and per-member row filtering in live traffic	[PLANNED]
Limited hashed keys for partners and agents; active audit log	[PLANNED]

New terms: API, OpenAPI, MCP, JSON-RPC, administrator key, row-level security, audit log.

11 Spend Guardrails, Safety and Release Operations

In short. Automatic breakers keep AI spending within bounds and can stop it instantly. Every change goes to a hidden staging copy first, gets an independent check, and only then goes live, under firm rules about databases and secrets.

11.1 Spending breakers

AI calls cost money, so the system protects itself mechanically instead of asking a person to approve each call (ADR-0006). On today's live report path, every AI call goes through one gate that runs these checks in order:

1. **The kill switch.** One setting turns all AI calls off. It can be set as a Worker setting or flipped by an administrator, and when it's off, every call is refused.
2. **The monthly spending guard.** Once the month's AI spending reaches its limit, calls are refused.
3. **An optional per-report cap.** It's available but switched off by the owner's choice. The monthly guard stays on.
4. **Logging.** Every call, and every refusal, is logged with its token counts and estimated cost.

The permanent database adds a fifth breaker, the **budget reservation**. Before a paid call, it reserves the estimated cost against two ceilings: \$15 per Land File and \$300 per month per organization by default. If either ceiling would be exceeded, the reservation is refused before any money is spent. This breaker is built and can be called through the API today. It will guard the live report path once the move to Postgres is complete and organization accounts exist.

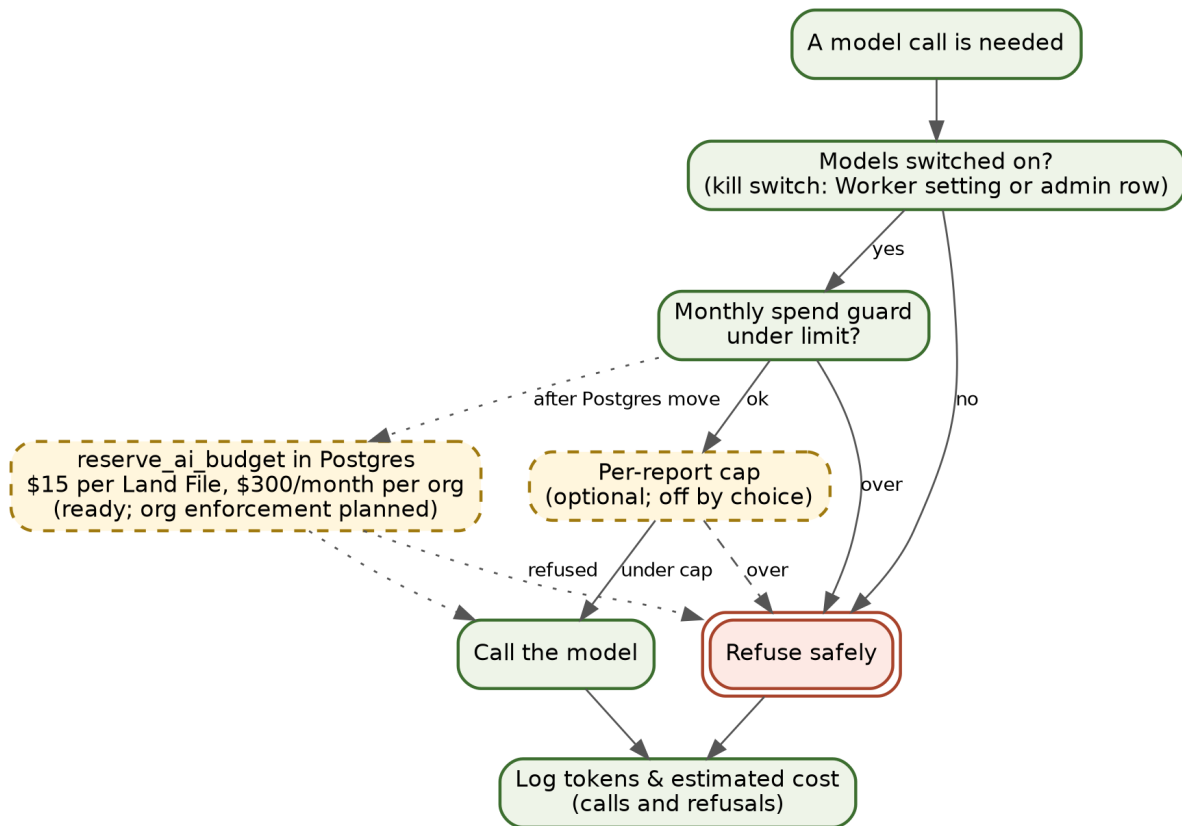


Figure 11.1: Spending and safety breakers.

11.2 Other safety locks

- **Hidden staging.** Staging sites and the backend tell search engines not to index them.
- **Polite fetching** protects the public archives (Chapter 5).
- **Separation of duties.** The builder never approves its own release (Rule 7).
- **Say only what is live.** The ledgers keep menus and manuals honest (Rule 5).
- **Documents are data, not instructions.** Text inside an uploaded or fetched document is treated as material to read, never as a command. A sentence in a document can't open a file, send a message, or change a permission.

11.3 How a change reaches the live site

1. **The build and infrastructure agent makes the change** in the repository and runs the tests.
2. **It deploys to staging**, the hidden twin. Today this is a manual deploy command. The automatic “deploy staging on every push” workflow is being restored.

3. **The independent reviewer checks it.** A reviewer who didn't build the change does a quick check and smoke test on staging: the main pages load, the changed feature works, and nothing private shows.
4. **The coordinator or the build agent promotes it to production.** During the current testing period, the owner has given standing authorization to promote tested changes without asking each time (see Appendix H). Staging and the independent check still come first.
5. **The shared STATUS.md file is updated** at the end of every task, so every agent and person can see the exact live state.

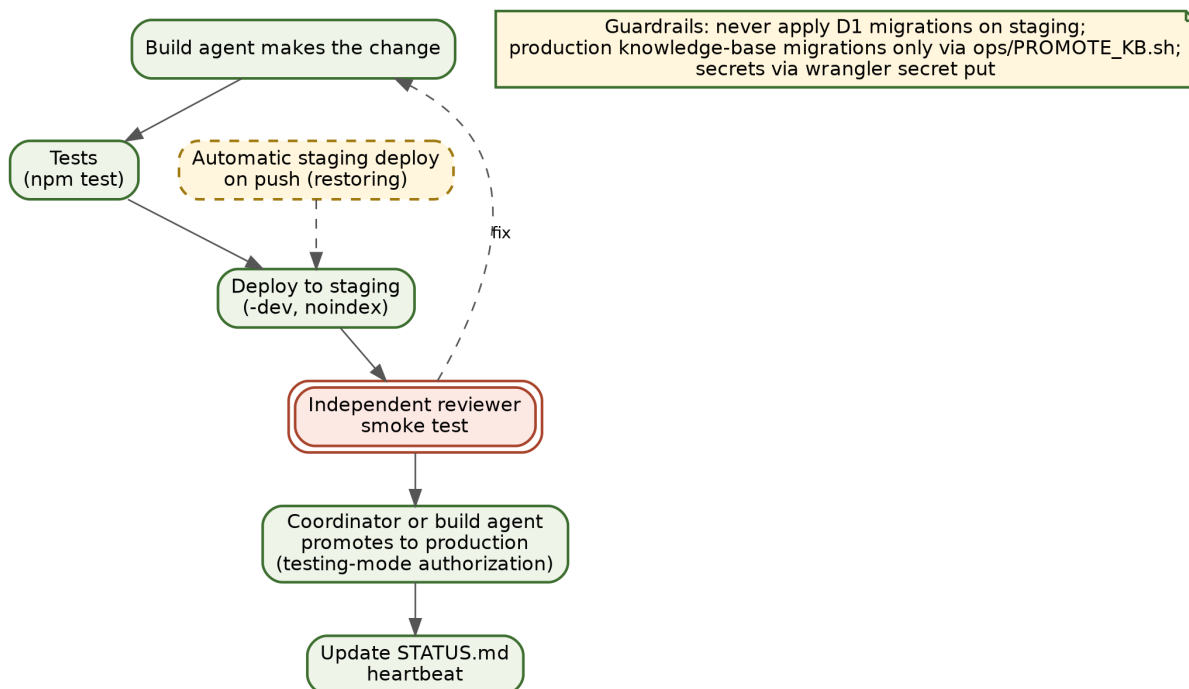


Figure 11.2: Release and promotion flow.

11.4 What the independent reviewer looks for

A smoke test is short, but it isn't casual. For each change, the reviewer works through four questions.

1. **Does it load?** The home page, the About pages, videos, the user manual, and the start page all open on staging.
2. **Does it do what it says?** The changed feature behaves as described in the change note, including on a phone-sized screen.
3. **Did it break anything nearby?** Pages that share code with the change still work, and the backend's health check still reports the database reachable.

4. **Is anything exposed that shouldn't be?** No key value or private file appears on a changed page, and a data route without a key still answers “not authorized.”

If any answer is no, the change goes back to the build agent with a note, and the cycle starts again on staging. Production never sees it.

11.5 Hard rules for operators

- Never run the D1 “apply migrations” command on staging. Staging’s knowledge-base changes were applied by hand, and re-running them would damage it.
- Production knowledge-base database changes go only through the promotion script, ops/PROMOTE_KB.sh.
- Set secrets only with wrangler secret put. Never commit them, print them, or paste them into chat.
- If something goes wrong in production, roll back to the previous Worker version first and investigate afterward (Appendix D).

Why This Choice. Mechanical breakers let the team move fast during testing without risking a runaway bill, and a staging copy plus an independent reviewer catch mistakes before the public sees them. (ADR-0006.)

For IT: Under the Hood. The gate is backend/src/lib/llm.js. The kill switch is the Worker variable LLM_ENABLED='off' or the settings row llm_enabled='off', set with POST /api/v2/admin/llm and the admin bearer key. Usage goes to the D1 table llm_usage (tokens and cents, per month, job, and property, ok flag, and a note for refusals). The per-report cap is LLM_REPORT_CAP_CENTS (unset). The writer switch is WRITER=off. The Postgres breaker is public.reserve_ai_budget(p_land_file_id uuid, p_estimated_cost_usd numeric), which atomically checks and increments land_files.budget_spent_usd and orgs.monthly_budget_spent_usd. It is exposed as POST /v1/budget/reserve and the MCP tool reserve_ai_budget. Tests run with npm test in backend/. Staging deploys with npx wrangler deploy --env dev and production with npx wrangler deploy. The CI workflow source is ci/staging.yml, being restored to .github/workflows/. Responses carry X-Robots-Tag: noindex, nofollow, and the backend’s /robots.txt disallows everything.

Status strip, Chapter 11

Capability	Status
Kill switch, monthly guard, logging (live path)	[BUILT]

Capability	Status
Per-report cap	[BUILT] (switched off by choice)
Postgres budget reservation	[BUILT]; on the live path and per organization [PLANNED]
Staging-first release with an independent check	[BUILT]
Automatic staging deploy on push	[PENDING]
Production knowledge-base migrations via script	[PENDING] (not yet run)

New terms: kill switch, circuit-breaker, staging, smoke test, promotion, rollback, STATUS.md.

Part IV: People, Agents and Key Decisions

12 Who Does What: Humans and Agents

In short. People make the decisions. Specialist reasoning models handle deep research and substantive writing, coordinator agents orchestrate workflows and mechanical verification, and the build agent owns the code, presentation layer, and canonical facts. None grades its own work.

This chapter describes **roles**. The current names of the agents and models filling those roles, which will change over time, are in Appendix H.

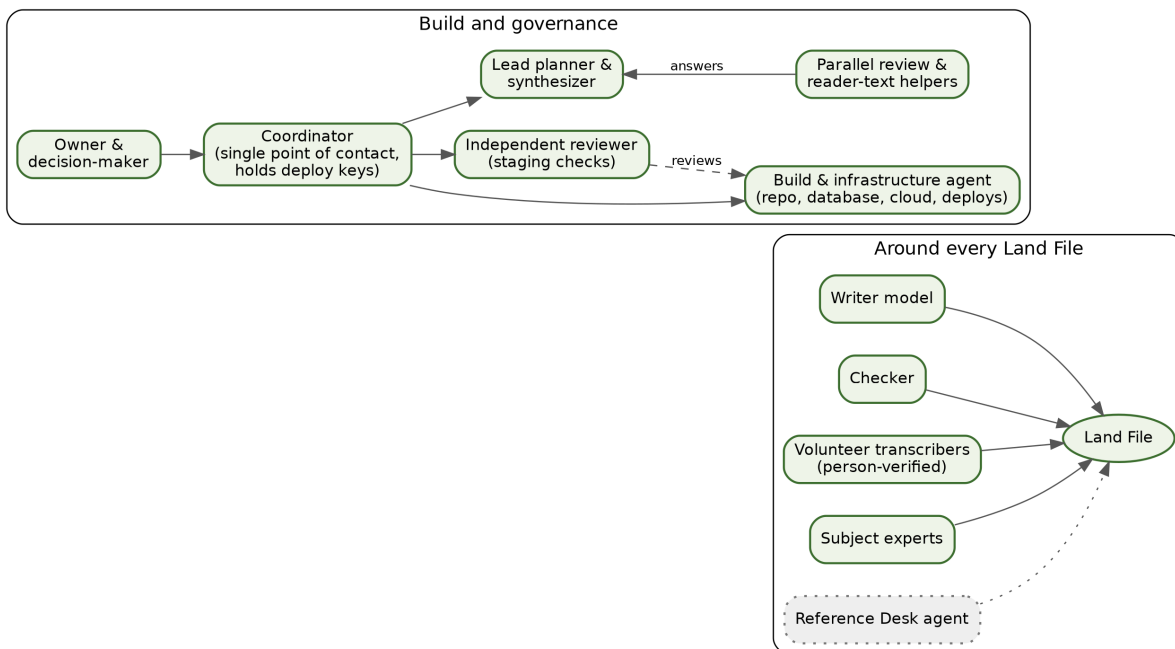


Figure 12.1: Who does what.

12.1 Build and governance roles

- **Owner and decision-maker.** Sets direction and makes final decisions. The owner deals with one point of contact rather than with each agent.
- **Coordinator.** The owner's single point of contact. It turns requests into work, picks the right model for each job, runs the parallel rounds, sends build requests to the build agent, checks the results, and delivers them. The coordinator and its crew do not write or research themselves: primary research goes to ChatGPT 6 Astra, Claude Fable

(never route gating/security/procedure to Fable), and Argon (Gemini 4); primary writing goes to Claude Opus 5.5 (routed through Farm Fable) and Argon (Gemini 4). ChatGPT and Grok also serve on single-model tasks, fan-outs, and red-team checks.

- **Build and infrastructure agent.** Owns the code repository, the backend, the databases, the cloud configuration, and deployments, and keeps the shared STATUS.md file up to date. It also does all the rendering at the presentation layer and owns the facts and every place they appear (web pages, reports, this manual), so a fact changed once updates everywhere.
- **Independent reviewer.** Checks changes on staging and runs smoke tests, and never reviews work it built itself. (This agent used to build and deploy the site itself. Since October 15, 2026 it only reviews, which is what makes it independent of the build agent.)
- **Lead planner and synthesizer.** Writes plans and master plans, and merges the answers from parallel review rounds into one result.
- **Parallel review and reader-text helpers.** For important questions, several different AI models answer the same question independently and send their answers to the lead planner. Different models catch blind spots any single model would miss. One helper also drafts reader-level text, which editors review and lock.

A typical project runs like this manual did. The owner asks the coordinator for something. The coordinator gathers the facts and sends a brief to the lead planner and the helpers. The helpers answer the lead planner, and the lead planner merges their answers into a plan. The work is done against that plan, and a second round of independent reviews checks it. The coordinator applies the fixes and delivers the result to the owner.

12.2 Models at work inside each Land File

- **Section writer:** the automated software pipeline's internal drafting agent (Layer 6) that drafts the story from evidence cards (Chapter 7), distinct from the team's operational writing farm.
- **Independent checker:** the rules that drop unsupported sentences. The checker is deterministic code, not a second model, so it can't be talked out of its rules.
- **Survey-note reader:** a model that makes a first reading of field-note page images, always labeled "Reading needs checking."
- **Document reader:** *planned*. It will read members' deeds and surveys after redaction.
- **Reference Desk agent:** *planned*. It will answer "Ask About Your Land" questions from cited sources. A demonstration screen is live today.

Outside the report pipeline, the project’s video and outreach work also uses speech-to-text tools for recorded interviews and text-to-speech tools for narration. Those tools don’t touch Land Files.

12.3 People in the loop

- **Volunteer transcribers** read surveyors’ handwriting and produce the person-verified readings that outrank any machine.
- **Subject experts and historians** supply the scholarship behind editor-locked sections. For example, Bob Zybach’s 1999 Oregon State University master’s thesis on the Soap Creek Valley is cited and quoted (never redistributed as a file), and his expertise informs the planned Reference Desk.
- **Landowners** add whatever they have about their land, from management plans and deeds to letters, family stories, and photographs, and they tell the project where its records are wrong.

Why This Choice. Clear roles make it obvious who is accountable for each step. Separating the builder from the reviewer, and using several different models for important questions, keeps mistakes from slipping through unchallenged. (*Rule 7, ADR-0006.*)

For IT: Under the Hood. Separation of duties is in `AGENTS.md` §1, and the release process in `AGENTS.md` §4. The writer model and its guards are in `backend/src/report/writer.js` and `backend/src/lib/llm.js`. Machine readings carry `method: 'llm'` and the model name. The PostgreSQL `audit_log` has an `actor_agent` column, so agent actions can be attributed once the log is in active use.

Status strip, Chapter 12

Role	Status
Owner, coordinator, build agent, reviewer, lead planner, helpers	[BUILT] (in daily use)
Section writer, checker, survey-note reader	[BUILT]
Volunteer transcription	[PARTIAL]
Document reader; Reference Desk agent	[PLANNED] (Reference Desk demonstration screen live)

New terms: coordinator, reviewer, lead planner, fan-out review.

13 Why We Made These Decisions

In short. Six recorded decisions shape the system. Each names the question faced, the choice made, why it matters, and what it costs.

The project records major choices as **Architecture Decision Records (ADRs)** in `docs/decisions/`. Here they are in plain language. Fuller digests are in Appendix E.

ADR-0001: A real map database as the system of record. *Question:* Where should the library's permanent data live? *Choice:* Supabase Postgres with PostGIS (map geometry) and pgvector (similarity search), with the edge database (D1) kept only as a fast cache once the move is done. *Why it matters:* real geometry across six survey systems, row-by-row ownership for many tenants, and a standard database any IT team knows. *Trade-off:* two databases during the transition. **Status:** schema [BUILT]; move [PARTIAL].

ADR-0002: Landowners own their private material. *Question:* Who owns a member's uploads when an organization sponsors them? *Choice:* the individual, always, by default. A sponsoring organization provides branding and a pooled AI budget, not automatic visibility into members' files. The policy gives owners export, a named successor, and erasure, plus narrow redaction before any model reads a document. *Why it matters:* families will share papers only if the papers stay theirs. *Trade-off:* organizations see less by default. **Status:** secret-link privacy, photo cleaning, and Forget [BUILT]; member sign-in, self-service export and successor, and server-side text redaction [PLANNED].

ADR-0003: Public records are shared openly. *Question:* What may the public do with records the library fetches? *Choice:* CC BY 4.0 with attribution where source terms allow, rights flags checked for each source, and quarterly mirrors to Zenodo, the Internet Archive, and Oregon State University (OSU) Libraries, so the record survives independently of any one company. *Why it matters:* the public commons should be permanent and reusable. *Trade-off:* rights must be tracked source by source. **Status:** license and storage [BUILT]; public serving and mirrors [PLANNED].

ADR-0004: One door for everyone. *Question:* How should the website, partners, and AI agents reach the data? *Choice:* one service with a standard REST API (`/v1`) and an agent interface (`/mcp`), backed by the same queries, permissions, and cost controls. *Why it matters:* one set of locks, one audit trail, no back doors. *Trade-off:* everything has to be designed as an API first. **Status:** [BUILT], locked to administrators; partner and agent keys [PLANNED].

ADR-0005: Manuals as code. *Question:* How do we keep manuals true to the live site? *Choice:* Markdown in the repository, compiled by Pandoc into Word, Large Print Word, and PDF, and checked against the ledgers. *Why it matters:* one source, every format, no drift. *Trade-off:* writers work in plain text. **Status:** [BUILT] for the Quick Start and Oregon guides; this manual uses the same method.

ADR-0006: Mechanical breakers instead of human gates. *Question:* How do we move fast without runaway costs or self-graded work? *Choice:* agents act directly in staging and tests, spending is guarded by automatic breakers, and an independent checker plus database rules verify every claim. *Why it matters:* speed with safety. *Trade-off:* the breakers must be set correctly. **Status:** kill switch and monthly guard [BUILT]; Postgres reservation [BUILT]; on the live path and per organization [PLANNED].

13.1 Decisions we may revisit

A recorded decision isn't a promise never to change course. It's a promise to change course openly. Three decisions are the likeliest to be revisited as the system grows:

- **How much stays at the edge (ADR-0001).** Once the move to Postgres is done, the team will measure whether a small edge cache is still worth keeping, or whether Postgres alone is fast enough.
- **Spending ceilings (ADR-0006).** The \$15-per-file and \$300-per-organization defaults were set before real usage data existed. They will be tuned, and partners may choose their own.
- **What “open” means for Class P (ADR-0003).** Some counties' terms restrict republishing. As more states open, the rights flags may grow more detailed, and some public records may be linked rather than mirrored.

Any change of this kind gets a new numbered ADR that explains what changed and why. The old one stays on file.

Why This Choice. Writing decisions down, with the reasons and the trade-offs, lets a new board member, partner, or engineer see *why* the system is shaped as it is, and lets the team revisit a decision honestly when circumstances change.

For IT: Under the Hood. The ADR files are docs/decisions/0001-postgres-postgis-over-d1.md, 0002-individual-ownership-class-m-files.md, 0003-cc-by-class-p-public-holdings.md, 0004-single-openapi-and-mcp-door.md, 0005-markdown-to-pandoc-docs.md, and 0006-mechanical-spend-circuit-breakers.md. The rights flags in source_registry are right_retrieve, right_retain, right_reproduce, right_publish, and right_paid_use.

Status strip, Chapter 13

Decision	Status
ADR-0001 Postgres system of record	[PARTIAL]
ADR-0002 Members own private material	[PARTIAL]
ADR-0003 Open public records	[PARTIAL]
ADR-0004 One door	[BUILT] (administrators only)
ADR-0005 Manuals as code	[BUILT]
ADR-0006 Mechanical breakers	[PARTIAL]

New terms: ADR, system of record, rights flags.

Part V: How the Technology Will Change Over Time

14 The Road Ahead

In short. Here is where the technology is headed, in three phases. Dates are targets, not promises. Software and AI change quickly, and the plans will change with them.

14.1 An honest premise

No one can predict the future of software or AI with certainty. What follows is our best current plan. The first phase is mostly finishing work already under way. The later phases depend on funding, partners, volunteers, and how the technology itself changes. Partners should build on what Chapters 1–13 mark as **[BUILT]** and treat everything here as direction, not commitment.

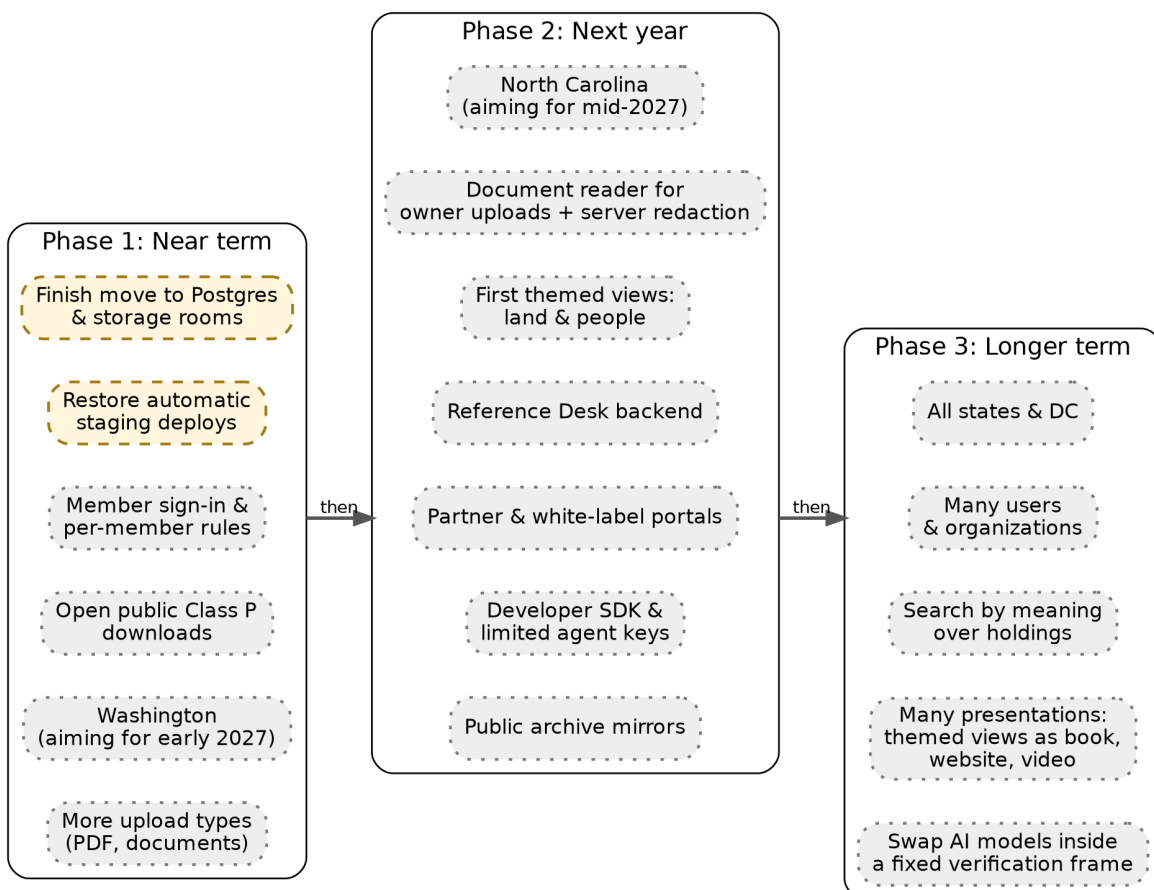


Figure 14.1: The road ahead.

14.2 Phase 1: Near term (finishing the core and opening member accounts)

- **Finish the move to Postgres.** Bring the working report data from the edge database into the permanent system of record, and move the collection into the three storage rooms.
- **Restore automatic staging deploys,** so every change pushed to the repository lands on staging by itself.
- **Turn on member sign-in.** Members sign in, and the database's row-level security filters every request to that member's own Land Files and papers.
- **Open public access to Class P records,** so public holdings can be downloaded directly under their open license.
- **Washington,** aiming for early 2027.
- **More ways to add material.** Accept more upload types than images, such as PDF documents and word-processing files, so owners can add management plans, reports, deeds, and family papers in the form they have them.

14.3 Phase 2: Next year (partners, the Reference Desk, and developer access)

- **North Carolina,** aiming for mid-2027.
- **A document reader** for whatever owners upload, such as management plans, deeds, surveys, tax statements, and letters, with D4-v2 redaction enforced on the server. Facts it reads become evidence cards in the owner's private Land File.
- **Family stories in the owner's words.** Typed or recorded stories about the land and its people, kept private unless the owner chooses to share or donate them.
- **The first themed views.** A land-focused view (soils, plants, trees, animals, birds, and water) and a people-focused view (owners, families, and their stories), written from the same checked cards as today's report.
- **A working Reference Desk.** "Ask About Your Land" moves from a demonstration screen to a conversational research helper that answers from cited sources and follows the same rules as the writer.
- **Partner and white-label portals.** Historical societies, woodland owners' associations, title researchers, and schools could sponsor Land Files for their members under their own branding, with a monthly AI budget per organization. Members still own their own papers.
- **Developer access.** A published software development kit (SDK) and limited keys for outside AI agents to use the /v1 API and the /mcp interface, with every call audited.

- **Public archive mirrors.** Quarterly copies of public holdings go to Zenodo, the Internet Archive, and Oregon State University Libraries.

14.4 Phase 3: Longer term (national scale and changing AI)

- **All states and DC**, building on the six survey systems the database already understands, by adding state registries and county adapters.
- **Many more users and organizations**, using edge caching, fingerprint-keyed storage that never stores the same file twice, and per-organization budgets and audit.
- **Many presentations of each Land File.** More themed views chosen by owners, teachers, and partners, delivered in the format that suits the reader: a printed book, a website, a narrated video, or forms not yet invented. Each is a new presentation of the same checked knowledge, not a new system.
- **Search by meaning** across holdings, built on the pgvector extension already installed in Postgres.

14.5 What a partner can build on today

For an organization considering working with America's School Trust Library, here is the practical line between today and tomorrow.

- **You can rely on these today:** the public website for Oregon and Utah, the Land File and its checks, the state guides and books, the Story link, and the published decision records and this manual.
- **Ask us about these:** read-only access to the /v1 API and the /mcp interface for research collaborations, using an administrator-issued key under supervision. These doors exist today, but they aren't yet open for self-service use.
- **Plan for these but don't depend on them yet:** member accounts, partner portals and branding, more upload types, themed views and formats, limited self-service keys, an SDK, the Reference Desk, document reading, and archive mirrors.

We'd rather under-promise and deliver. If any of the planned items matters to your plans, talk to us. Partners help decide what comes first.

14.6 How we'll handle changing AI models

AI models will keep getting faster, cheaper, and more capable, and old ones will be retired. The architecture treats them as **replaceable workers** on Layer 6. When a better model arrives, we can swap the writer, a reader, or the Reference Desk model behind the same interfaces. What stays fixed is the frame around them: evidence cards with sources, the

database rule that refuses unsourced findings, the checker’s rules, editor-locked history, and the people who verify readings. That frame is what makes a Land File trustworthy, whichever model writes the sentences.

14.7 Risks, and how we’d respond

A candid roadmap names its risks:

- **A key AI provider changes its terms or prices.** The writer and readers can be swapped behind the same interfaces, and private material goes only to providers whose terms forbid training. If none qualifies for a task, that task waits.
- **A major source closes or blocks automated access.** The source drops to “follow link” or “ask office,” the stored holdings remain, and the gap box says so.
- **Funding is slower than hoped.** Phase 1 finishes the core either way. Later phases slow down rather than cut corners, and the trust frame isn’t traded for speed.
- **Volunteer verification doesn’t keep up.** Machine readings stay labeled until a person checks them. The story stays honest about what is checked and what isn’t.
- **A security mistake.** Staging, independent review, administrator keys, row-level security, and fast rollback limit the damage, and as on October 15, 2026, problems are fixed in the open and recorded.

Why This Choice. Building the trust frame first, and treating models as replaceable, means the library can take advantage of better AI without having to re-earn its readers’ trust each time.

For IT: Under the Hood. The roadmap items map to existing scaffolding. Postgres tables and functions are in `supabase/migrations/0001_astl_foundation.sql`. Org budgets are `orgs.monthly_budget_limit_usd`. Limited keys are `api_keys`. Branding is `orgs.custom_domain` and `branding_config`. Search by meaning will use `findings.embedding` (pgvector). The SDK can be generated from `/v1/openapi.json`. The staging workflow is `ci/staging.yml`. None of these are live features until the ledger says so.

Status strip, Chapter 14

Item	Status
Postgres and storage move	[PARTIAL]
Automatic staging deploys	[PENDING]
Member sign-in, public Class P access, Washington	[PLANNED] (Phase 1 targets)

Item	Status
North Carolina, document reader, Reference Desk, partner portals, SDK and agent keys, mirrors	[PLANNED] (Phase 2 targets)
All states, national scale, search by meaning	[PLANNED] (Phase 3 direction)

Appendix A: Route and Interface Directory

A.1 Web routes (yourlandhistory-web).

Route	Purpose	Status
/	Home: the state-first front door	[BUILT]
/start, /find, /state, /state/ confirm, /lots, /map	Find and confirm land; multi-lot tray	[BUILT]
/land-file, /my-land, / examples	Land File entry, the owner's land, worked examples	[BUILT]
/oregon, /utah, /coverage, / catalog	State pages, coverage by county, the catalog of record types	[BUILT]
/books-videos, /books, / videos, /guides/, /howto, / manual	Books, videos, state guides, quick start, user manual	[BUILT]
/school-sections/	The School Sections showcase	[BUILT]
/about, /about/technology, / how-it-works, /plans, / privacy, /terms, /text-size	About pages (including this manual) and policy pages	[BUILT]
/argon/*	Edition 2.0 interactive screens, including the Add Papers demonstration	[BUILT]
/sign-in	Sign-in screen for secret-key access	[BUILT]; member accounts through Supabase Auth [PLANNED]
/api/*	Passed through to the backend over the service binding	[BUILT]

A.2 /v1 (OpenAPI 3.1) on parcel-v2-backend.

Method and route	Purpose	Main parameters	Access
GET /v1	API index	none	Open
GET /v1/health	Service and database connectivity	none	Open
GET /v1/openapi.json	The OpenAPI 3.1 contract	none	Open
GET /v1/states	State registry	status	Administrator key
GET /v1/ states/:state	One state	two-letter code	Administrator key

Method and route	Purpose	Main parameters	Access
GET /v1/places	List places	state, county, PLSS fields	Administrator key
POST /v1/places/find	Find places	JSON body like find_l and	Administrator key
GET /v1/places/:id	One place	UUID	Administrator key
GET /v1/holdings	Public holdings with provenance	place_id, source_id, sha256_hash	Administrator key
GET /v1/land-files	Land File list	none	Administrator key
GET /v1/land-files/:id	One Land File, with findings	UUID	Administrator key
GET /v1/findings	Source-cited claims	land_file_id, place_id, review_status	Administrator key
POST /v1/budget/reserve	Budget reservation	land_file_id, estimated_cost_usd	Administrator key

A.3 /mcp (JSON-RPC 2.0, stateless, administrator key). The methods are initialize, ping, tools/list, and tools/call. The tools are list_states, find_land, get_place, list_holdings, get_land_file, get_findings, and reserve_ai_budget, with their inputs as in Chapter 10. An example call:

```
POST /mcp
Authorization: Bearer <administrator key>
Content-Type: application/json

{"jsonrpc": "2.0", "id": 1, "method": "tools/call",
 "params": {"name": "list_states", "arguments": {"status": "open"}}
```

A.4 /api/v2 (legacy routes used by the website).

Route	Purpose	Access
GET /api/v2/health	Health	Open
GET /api/v2/registry, /catalog, /coverage	Source registry, record-type catalog, county coverage	Open
GET /api/v2/holdings	Holdings index	Open
/api/v2/lookup/address, /taxlot, /lots, /town	Find land	Open
POST /api/v2/property; GET /api/v2/property/:id	Build a report; read a report	Open; private reports need their secret link key
/api/v2/property/:id/uploads	List, add, or remove the owner's uploads (images today)	Secret link key

Route	Purpose	Access
POST /api/v2/property/:id/forget	Erase a Land File	Secret link key
GET /api/v2/open-questions	Open questions	Open
GET /api/v2/kb/tribal-offices, /kb/zybach	Office contacts; cited scholarship index	Open
GET /api/v2/diag/reach	Network reachability check	Open
/api/v2/admin/write, /section-text, /holdings, /llm, /readings, /evidence, /machine-read, /survey-dates	Administration	Admin key

Appendix B: Database Reference

B.1 Supabase Postgres (astl-core). All 16 tables have row-level security switched on. FORCE ROW LEVEL SECURITY, which applies the rules even to the table's owner, is set on `land_files`, `papers`, and `findings`. A table with row-level security and no policy refuses all ordinary access, so until member sign-in adds per-member policies, only the backend's server-only key can read or write these tables. Primary keys are time-ordered UUIDs from `uuidv7()`, except where noted.

Layer	Table	Purpose	Key columns
1	<code>orgs</code>	Sponsoring organizations	<code>slug</code> , <code>name</code> , <code>custom_domain</code> , <code>branding_config</code> , <code>org_owns_member_files</code> (default false), <code>monthly_budget_limit_usd</code> (\$300 default), <code>monthly_budget_spent_usd</code>
1	<code>users</code>	Members	<code>email</code> , <code>display_name</code> , <code>successor_name</code> , <code>successor_email</code>
1	<code>org_memberships</code>	Who belongs to which organization	<code>org_id</code> → <code>orgs</code> , <code>user_id</code> → <code>users</code> , <code>role</code>
1	<code>api_keys</code>	Limited keys (hashed)	<code>org_id</code> , <code>user_id</code> , <code>key_prefix</code> , <code>key_hash</code> , <code>scopes</code> , <code>monthly_limit_usd</code> , <code>monthly_spent_usd</code> , <code>expires_at</code> , <code>revoked_at</code>
1	<code>audit_log</code>	Append-only record of actions	<code>org_id</code> , <code>actor_user_id</code> , <code>actor_agent</code> , <code>action</code> , <code>target_type</code> , <code>target_id</code> , <code>cost_usd</code> , <code>metadata</code>
0/4	<code>states_registry</code>	States and survey systems (key: <code>state_code</code>)	<code>state_name</code> , <code>status</code> , <code>primary_survey_system</code> , <code>meridian_summary</code> , <code>capabilities</code> , <code>bbox</code> , <code>last_verified_date</code>

Layer	Table	Purpose	Key columns
0/4	source_registry	Sources and their rights	state_code, county_fips, office_name, source_url, allows_owner_name_search, right_retrieve, right_retain, right_reproduce, right_publish, right_paid_use, terms_notes, last_checked_at, parser_version
2	places	Parcels and survey units	place_key, state_code, county_name, survey_system, PLSS fields, is_school_trust_section, external_identifiers, geom (PostGIS), geom_date_valid_from/_to
2	role_assertions	People as dated roles	land_file_id, place_id, holding_id, paper_id, person_name_as_written, role_type, assertion_date, living_or_presumed_dead
3/5	land_files	Land Files	org_id, owner_user_id, secret_claim_token_hash, title, state_code, primary_place_id, data_class, encryption_key_ref, budget_limit_usd (\$15 default), budget_spent_usd, deleted_at
3/5	land_file_places	Places in a Land File (composite key)	land_file_id, place_id, relationship_note
3	holdings	Public and donated records	data_class, source_id, place_id, r2_object_key, sha256_hash, mime_type, title, document_date, extracted_text, license, fetched_at

Layer	Table	Purpose	Key columns
3	papers	Member material	land_file_id, org_id, owner_user_id, data_class, origin, r2_object_key, sha256_hash, original_filename, mime_type, pii_redacted, exif_stripped, extracted_text, reader_model, reader_prompt_version, rights_holder
4	findings	Source-cited claims	land_file_id, place_id, claim_text, holding_id, paper_id, locator, supporting_quote, conflicts_with_finding_id, produced_by_model, prompt_version, review_status, verified_by, verified_at, embedding; constraint chk_finding_has_source
5	research_plan_steps	Steps of a research plan	step_number, step_kind, target_office, legal_description_scope, estimated_fee_usd, user_consented, status, instructions_or_draft_letter
5	story_versions	Published editions of a story	edition_number, reading_level, markdown_body, change_notes, writer_model, checker_model, source_manifest_hashes, is_published_class_d

Functions. `uuidv7()` generates time-ordered keys. `reserve_ai_budget(p_land_file_id, p_estimated_cost_usd)` atomically checks and increments the per-file and per-organization spending.

B.2 Where the live data is today.

Data	Lives in
Live reports: properties, lots and trays, evidence cards, readings, written sections, uploads metadata, report link-key fingerprints, settings, AI usage	Cloudflare D1 parcel-v2-db (migrations 0001–0012; knowledge-base 0019–0022 on staging only)
Fetches records and owner uploads (file contents)	The backend's content-addressed store
Foundation schema, state and source registries, seed data	Supabase Postgres astl-core
Permanent class storage	R2 astl-class-p-public, astl-class-m-private, astl-class-d-donated (connected; being filled)

Appendix C: State Coverage and Source Inventory

C.1 Registered states (states_registry).

State	Status	Survey system
Oregon	Open	PLSS (Willamette Meridian) and Donation Land Claims
Utah	Open	PLSS (Salt Lake Meridian)
Washington	Coming (target early 2027)	PLSS (Willamette Meridian) and Donation Land Claims
Idaho	Registered, coming later	PLSS (Boise Meridian)
California	Registered, coming later	PLSS (Mount Diablo and others); Spanish and Mexican grants
New Mexico	Registered, coming later	PLSS (New Mexico Principal Meridian); Spanish and Mexican grants
Illinois	Registered, coming later	PLSS (2nd and 3rd Principal Meridians)
Texas	Registered, coming later	Texas abstracts and sections
Virginia	Registered, coming later	Metes and bounds
Hawaii	Registered, coming later	Ahupuaʻa (Great Māhele)

North Carolina is listed in the source ledger with a target of mid-2027.

C.2 National sources in the source ledger. BLM CadNSDI and GLO records are fetched automatically, as are USGS National Map products, USDA NRCS soils, and the Royce/Kappler treaty cessions. NOAA NCEI climate data and US Census data are used through the county fetcher. These entries record host, coverage, refresh interval, storage class, and the date each was checked.

C.3 Fetchers and adapters. Fetchers: glo, blmor, cadnsdi, county, polkmap, sda, sgmc, dogami, gnis, tnm, royce, finder. County adapters: arcgis (generic), polk, benton, owrdr (statewide outlines in the 13 Oregon counties that allowed public viewing on October 12, 2026).

C.4 Source modes. automatic, follow_link, ask_office, found_not_wired, not_verified.

Appendix D: Operations Runbook

This runbook is for operators. All commands assume a clone of `dave-schooltrustlands/yourlandhistory` and Wrangler authenticated with a Cloudflare token that is allowed to deploy Workers. Never paste a key value into a command line that is logged, a ticket, or a chat.

D.1 Everyday change

1. **Test (from backend/).** `cd backend && npm test`, then `cd ..` to return to the repository root.
2. **Build docs (from the repository root), if content changed.** `node scripts/build-docs.mjs`. The outputs land in `dist/docs/`.
3. **Deploy the backend to staging (from backend/).** `npx wrangler deploy --env dev`.
4. **Deploy the website to staging (from web/).** `npx wrangler deploy --env dev`. The web build step regenerates the book pages first.
5. **Smoke test staging.** The test passes when all of these hold:
 - the staging home page, `/about`, `/videos/`, `/manual`, and `/start` return 200 (after redirects);
 - the changed feature works as described;
 - `GET /v1/health` on the staging backend reports the database reachable;
 - a `/v1` data route without a key returns 401;
 - no private material or key appears on any changed page.
6. **Independent review.** The independent reviewer signs off on staging.
7. **Promote to production.** Run `npx wrangler deploy` (no `--env`) in `backend/` and/or `web/`. Under the testing-mode authorization (Appendix H), no further approval is needed.
8. **Verify production.** Repeat the step 5 checks on `yourlandhistory.org` and the production backend.
9. **Record.** Update and commit `STATUS.md` with the new Worker version IDs.

D.2 Database changes

- **D1, staging.** Never run `wrangler d1 migrations apply` on staging. Its knowledge-base migrations were applied by hand.
- **D1, production knowledge-base migrations.** Run only `ops/PROMOTE_KB.sh`, which applies the reviewed knowledge-base migrations in order.
- **Postgres.** Add a new numbered file under `supabase/migrations/`, review it, and apply it to `astl-core` with the Supabase tools. Never edit an applied migration.

D.3 Secrets

Set a secret with `npx wrangler secret put NAME` (add `--env dev` for staging) and type the value at the prompt. List secret names with `npx wrangler secret list`; values are never shown. To rotate a key, set the new value, verify, then revoke the old one wherever it was issued.

D.4 Emergency stop for AI spending

Either set the Worker variable `LLM_ENABLED` to `off` and redeploy, or switch it off at runtime (send `true` to switch it back on):

```
curl -X POST https://<backend host>/api/v2/admin/llm \
-H "Authorization: Bearer $ADMIN_TOKEN" \
-H "Content-Type: application/json" \
-d '{"enabled":false}'
```

`GET /api/v2/admin/llm`, with the same header, reports the switch and the month's spending. To stop only report writing, set `WRITER` to `off`.

D.5 Rollback and recovery

- **A Worker release went wrong.** Run `npx wrangler deployments list` in the affected folder, then `npx wrangler rollback <version-id>` to return to the previous version. Investigate afterward, then fix forward through staging.
- **D1 data.** D1 keeps point-in-time history. `npx wrangler d1 time-travel info parcel-v2-db` shows the restore points, and `npx wrangler d1 time-travel restore parcel-v2-db --timestamp=<time>` restores. Restore on a copy first where you can.
- **Postgres.** Supabase keeps daily backups for the project, and point-in-time recovery is available on the paid plan. Restore from the Supabase dashboard.
- **Stored files.** Holdings are content-addressed, so a missing public holding can be re-fetched from its source. Member uploads can't be re-fetched, which is why deletions of member material happen only on the owner's request.

Appendix E: Decision Digests (ADRs 0001–0006)

E.1 ADR-0001: Supabase Postgres + PostGIS + pgvector over Cloudflare D1 (docs/decisions/0001-postgres-postgis-over-d1.md)

- *Context and question.* The first versions of the site kept everything in D1, a small edge database. A national library needs real geometry, multi-tenant ownership, and standard tooling.
- *Decision.* Managed Postgres with PostGIS and pgvector on Supabase (astl-core) becomes the single system of record for Layers 1–5, reached from Workers over HTTPS and pooled connections. D1 may remain only as a short-lived edge cache.
- *Mechanism.* 16 tables with row-level security, UUIDv7 keys, PostGIS geometry with validity dates, and an embedding column for future search.
- *Why the owner and board care.* Data that will outlive any one vendor’s edge product, in a database any IT professional can run.
- *Trade-offs.* A period of running two databases, and the work of moving data carefully.
- *Status.* Schema [BUILT]; working data still in D1 [PARTIAL].

E.2 ADR-0002: Individual ownership of Class M files (docs/decisions/0002-individual-ownership-class-m-files.md)

- *Context and question.* Organizations may sponsor members. Who then owns a member’s papers?
- *Decision.* The individual owner, by default. A sponsor provides branding and a pooled AI budget, not visibility. Row-level security restricts Class M records to their owner, or to an explicit, disclosed share. Story links never show Class M. Owners can export, name a successor, or erase. Narrow D4-v2 redaction applies before any model reads text.
- *Mechanism.* owner_user_id on land_files and papers; org_owns_member_files defaults to false; FORCE ROW LEVEL SECURITY; pii_redacted and exif_stripped flags; the successor_* columns on users.
- *Why the owner and board care.* Families will trust the library with their papers only if those papers remain theirs.
- *Trade-offs.* Sponsors see less, and support work is harder without access to members’ files.

- *Status.* Today, privacy rests on secret link keys, server-side image cleaning, and Forget [BUILT]. Text redaction is a browser demonstration [PARTIAL]. Member sign-in, per-member row filtering, self-service export, and successor settings are [PLANNED].

E.3 ADR-0003: CC BY 4.0 for Class P public holdings (docs/decisions/0003-cc-by-class-p-public-holdings.md)

- *Context and question.* What may the public do with records the library fetches and curates?
- *Decision.* Class P holdings, state guides, and transcriptions are released under CC BY 4.0 with attribution to ASTL, excluding records whose source terms forbid publishing. Connectors honor `robots.txt`, throttle to one request per second per host, and check five rights flags before caching. Unencumbered holdings are mirrored quarterly to Zenodo, the Internet Archive, and Oregon State University Libraries.
- *Mechanism.* The `source_registry` rights flags, the `astl-class-p-public` bucket, and content hashes.
- *Why the owner and board care.* A public commons that survives independently of any one company or grant.
- *Trade-offs.* Rights must be researched and recorded for every source.
- *Status.* License, polite fetching, rights flags, and bucket [BUILT]; public serving and mirrors [PLANNED].

E.4 ADR-0004: A single OpenAPI and MCP service door (docs/decisions/0004-single-openapi-and-mcp-door.md)

- *Context and question.* The website, future partners, and AI agents all need data. Should each get its own path?
- *Decision.* One service boundary on the backend Worker with two protocol surfaces, `/v1` (REST, OpenAPI 3.1) and `/mcp` (JSON-RPC 2.0), backed by the same queries, permissions, and cost controls.
- *Mechanism.* `backend/src/api/v1_mcp.js`, the routes in Appendix A, and seven MCP tools.
- *Why the owner and board care.* One set of locks to secure, one audit trail, and no back doors.
- *Trade-offs.* Every feature must be designed as an API first.
- *Status.* [BUILT], locked to an administrator key; partner and agent keys [PLANNED].

E.5 ADR-0005: Docs as code, Markdown compiled by Pandoc (docs/decisions/0005-markdown-to-pandoc-docs.md)

- *Context and question.* Manuals drift away from software. How do we keep them true?

- *Decision.* Canonical Markdown in the repository, checked against the ledgers, compiled by one script into Word, Large Print Word (14-point), and PDF, and linted for US units and American English.
- *Mechanism.* `content/`, `scripts/build-docs.mjs`, and `dist/docs/`.
- *Why the owner and board care.* Readers always get a manual that matches the site, in the format they prefer.
- *Trade-offs.* Writers work in plain text, and layout is set by templates.
- *Status.* [BUILT] for the Quick Start and Oregon guides; this manual uses the same method.

E.6 ADR-0006: Mechanical spend circuit-breakers over human gates (`docs/decisions/0006-mechanical-spend-circuit-breakers.md`)

- *Context and question.* Waiting for human approval at every step slows testing. Running agents unchecked risks runaway cost and self-graded work.
- *Decision.* Agents act directly in staging and tests. Before any paid AI call, a transactional Postgres function reserves budget against per-file and per-organization ceilings. The model that writes never grades its own output. An independent checker and database rules verify claims.
- *Mechanism.* `reserve_ai_budget`, the LLM kill switch, the monthly guard, `chk_finding_has_source`, and the report checker.
- *Why the owner and board care.* Speed during testing, with a hard ceiling on cost and an independent check on quality.
- *Trade-offs.* The breakers have to be tuned correctly.
- *Status.* Kill switch, monthly guard, and checker on the live path [BUILT]; `reserve_ai_budget` [BUILT], on the live path and per organization [PLANNED].

Appendix F: Worked Examples

These examples follow the system's real formats and data models, drawn from the Sullivan tree farm near Pedee (T9S R6W Sec 33, Polk County, OR, 386.4 acres) and pipeline test cases.

F.1 A bearing tree, from field note to verified reading

Raw field note (transcribed): "Set post for cor. to secs. ... from which a fir 24 ins dia bears N 45 E 112 lks dist."

Machine reading (structured):

```
{ "corner": "C4", "species": "fir", "diameter_in": 24,
  "bearing": "N 45 E", "distance_links": 112, "distance_feet": 74,
  "method": "llm", "verification": "needs-checking" }
```

As shown: "Fir, 24 inches, N 45° E, 112 links (about 74 feet). *Reading needs checking.*"

Person verification. A volunteer opens the scanned page beside the reading. The volunteer reads 112 links and confirms it, and the reading becomes `"method": "person"`, *person-verified*. Had the volunteer read 142 links, the person's reading would be stored and used, and the machine reading would stay in the machine list, still labeled.

F.2 A sentence the checker drops

Cards in the section: k1, a county history describing a wagon road through the valley in 1852, with scope `region`; and k2, a township plat (1855) showing a trail along a creek in the section, with scope `region`.

Writer draft: "In 1852 a wagon road crossed this land [k1]."

Checker. Every card the sentence cites is about the region, not this land, and the sentence says "this land." **Dropped:** *places a regional or nearby record on this land.*

A sentence that passes: "A wagon road ran through the valley by 1852 [k1]." It is regional and says so.

When the original would pass. If a card were an 1852 field note or plat showing the road crossing this parcel's own boundaries (scope `this-land`), the original sentence would pass, and the year 1852 would have to appear on that card.

F.3 An upload and the redaction rule

Today (an image). An owner uploads a phone picture of the first page of their forest management plan, captioned “2004 management plan, page 1.” The file carries the GPS location where it was taken. The backend strips it, keeps the date taken, stores the cleaned image linked only to that Land File, and reports how many bytes of hidden data were removed. The image appears only to someone holding that Land File’s secret link key. It never appears on the Story link.

The planned document path (demonstrated on the Add Papers screen). A scanned tax statement contains a tax account number, a Social Security number, and a bank routing number. After redaction, the text a model would see reads:

```
Account: 12345 (kept)    SSN: [REDACTED]    Routing: [REDACTED]
```

The tax account number stays because the history needs it. The owner can still see their original page.

Appendix G: Glossary and Component Index

Plain term	Technical term	Chapter
Small program at the network edge	Cloudflare Worker	8
Private line between two Workers	Service binding	8
Hidden test copy of the site	Staging (-dev)	8, 11
Working report database	Cloudflare D1	8
Permanent system of record	Supabase Postgres + PostGIS	8
Digital storage room	R2 bucket	9
Digital fingerprint of a file	SHA-256 content hash	5
Where a record came from	Provenance	5
Secret key that opens a private Land File	Report link key (hashed)	9
Database rules that limit rows to their owner	Row-level security (RLS)	10
Spending switch that trips before money is spent	Transactional circuit-breaker	11
Master lists of what is live	<code>capabilities_ledger.json</code> , <code>states_config.json</code>	1
Machine-readable API menu	OpenAPI 3.1	10
Agent interface	Model Context Protocol (MCP), JSON-RPC 2.0	10
Recorded design decision	Architecture Decision Record (ADR)	13
Surveyors' units	Chain (66 feet); link (7.92 inches)	3
Trees marking a survey corner	Bearing (witness) trees	5
Anything an owner adds about their land (plans, deeds, maps, letters, stories, photographs)	Owner upload; Class M material	1, 9
One way of showing the same knowledge (today's standard report; planned themed views and formats)	Presentation (Layer 7)	1, 2, 7, 14

Component index.

- *Workers*: `yourlandhistory-web`, `yourlandhistory-web-dev`, `parcel-v2-backend`, `parcel-v2-backend-dev`.
- *Databases*: `parcel-v2-db` and `parcel-v2-db-dev` (D1); `astl-core` (Supabase).
- *Buckets*: `astl-class-p-public`, `astl-class-m-private`, `astl-class-d-donated`.
- *Backend modules*: `api/v1_mcp.js` and `api/lots.js`; `fetch/polite.js` and `fetch/robots.js`; `fetchers/*`; `adapters/*`; `pipeline/` (`aliquot`, `build`, `native`, `plan`, `recent`, `shelf`, `town`, `units`); `report/` (`assemble`, `writer`, `texts`, `forget`); `kb/` (`blobstore`, `holdings`); `extract/trees.js`; `lib/` (`llm`, `imgclean`, `geo`, `units`, and others); `data/` (`catalog`, `registry`, `tribal offices`, and others).
- *Scripts*: `scripts/build-docs.mjs`, `web/scripts/build-books.mjs`, `ops/PROMOTE_KB.sh`.
- *Ledgers*: `ledger/capabilities_ledger.json`, `ledger/states_config.json`.
- *Governance*: `AGENTS.md`, `STATUS.md`, `docs/decisions/`.

Appendix H: Current Team Roster and Operating Mode (as of October 15, 2026)

Role (Chapter 12)	Currently filled by	Operational Scope & Authority
Owner and decision-maker	Dave Sullivan	Strategic direction, legal/business posture, sole sign-off on sent communications, direct voice drafting.
Chief of Staff & Coordinator	Chaney (Grok Bot)	Owner's single contact; docket triage, un-spun packet preparation, fan-out orchestration, mechanical verification, draft staging. Zero substantive prose or research.
Specialist Bot Crew	Herald, Advocate, Librarian, Social, Bunny Writer, Masthead, Sullishak	Domain docket coordination, source intake, mechanical citation checks, draft staging in review queues.
Build, Presentation & Facts Engine	Antigravity (Google's agent environment)	100% of code, edge APIs, database schemas, presentation-layer rendering across all surfaces (Web, Video, Print), style systems, and canonical fact registry.
Independent reviewer	Builder (formerly build-and-deploy)	Staging review, automated smoke testing, and pre-production quality gatekeeper. Independent of build agent.
Primary Research Farm	ChatGPT 6 Astra, Claude Fable, Argon (Gemini 4)	Deep primary records, archival wisdom, cross-document reconciliation. <i>Fable is restricted: never route gating, security, or procedural tasks to Fable.</i>
Primary Writing Farm	Claude Opus 5.5, Argon (Gemini 4)	Outbound prose, persuasive advocacy, book chapters, technical manuals. <i>Opus 5.5 routes through the existing Farm Fable helper.</i>
Auxiliary & Red-Team Farm	ChatGPT (General), Grok	Simple tasks get 1 model; most tasks get best model or fan-out with merged results; adversarial cross-checks.

Role (Chapter 12)	Currently filled by	Operational Scope & Authority
Product Report Pipeline (Layer 6)	Internal pipeline models	Automated section writer and survey-note reader inside each Land File (named in backend/src/report/writer.js).

Model Routing Guide & Team Rules. Detailed routing policies, prompt boundaries, and fan-out procedures are maintained canonically in docs/MODEL_ROUTING_GUIDE.md and ledger/team_roster.json in the primary repository (dave-schooltrustlands/yourlandhistory).

Testing-mode authorization. On October 15, 2026, the owner gave standing authorization, for the current testing period, for the coordinator and the build agent to promote tested changes to production without asking each time. Staging deployment and an independent smoke test still come first. The owner can revoke the authorization at any time. It is recorded in AGENTS.md.

Appendix I: Questions Readers Often Ask

For landowners, historians, and board members

Is a Land File a legal document? No. It's a historical research account with sources. For boundaries or title, consult a licensed surveyor, a title company, or an attorney.

Can the AI make things up? The writer can draft a wrong sentence, which is why the checker exists. A sentence must cite a card from its own section, can't contain a number its cards don't have, and can't place a regional record on your land. If it fails, it's dropped. The most sensitive history is written by people and locked.

What can I add to my Land File? Anything you have about your land: management or farm plans, reports, deeds, surveys, maps, letters, notes about prior owners, family stories, and photographs. Today the upload takes images, so photograph or scan the pages. Other file types are planned.

Will the Land File always look like this? Today it is one standard report. The same checked knowledge is built to be presented in other ways later, such as land-focused or people-focused views, and as a printed book, a website, or a video (planned).

Who can see my Land File? Only someone with its secret link key. The public samples are the only exceptions. If you share a Story link, it hides your name and leaves out your photographs.

Will my uploads train an AI? No. Private material goes only to paid AI services whose terms forbid training on what we send.

Can I delete everything? Yes. You can delete any upload, or ask the system to forget a whole Land File. Self-service export and account-wide "Forget me" arrive with member sign-in.

Why only Oregon and Utah? Each state's records live in different places under different rules. A state opens only when its sources are wired, checked, and politely reachable. Washington is the next target.

How is Native history handled? As factual history, from treaties, survey records, and scholarship, written or reviewed by people and locked so the AI writer can't alter it. Tribal cultural-resource and historic-preservation offices are listed with checked contact details.

For IT professionals

Where is the source of truth today? For live reports, D1 (`parcel-v2-db`). For the permanent schema, registries, and future collection, Postgres (`astl-core`). Appendix B lists where each kind of data lives.

How are secrets handled? As Cloudflare Worker secrets, set with `wrangler secret put`. They are never in Git, never in logs, and never in this manual.

What stops runaway AI spending? On the live path: the kill switch, the monthly guard, and per-call logging. In Postgres: transactional per-file and per-organization reservations, ready for the live path after the move.

How do I get API access? Today, only with an administrator key, issued for supervised collaborations. Limited self-service keys are planned (Chapter 14).

Is the code open? The repository is private during testing. The About page records the project's commitment to an open architecture, and the decision records in this manual describe the design in full.

How do I recover from a bad deploy? `wrangler rollback` to the previous Worker version. Then fix forward through staging (Appendix D).

Appendix J: Keeping This Manual Current

This manual is meant to stay true, so its upkeep is part of the release process.

- **Who updates it.** The build and infrastructure agent (Antigravity) owns compiling and rendering the manual across all formats (Word, Large Print, PDF, Web), running the automated section-shrinkage verifier on every rebuild, and drafting updates when features ship. The coordinator reviews them, and the independent reviewer (Builder) verifies staging.
- **When.** Whenever a ledger entry changes status, such as a [PLANNED] item becoming [BUILT]. Also whenever a route, table, or Worker named here changes, and in any case at least once each quarter.
- **How.** Edit the Markdown source, re-render any changed diagram from its Graphviz source, rebuild the Word, Large Print, PDF, and web editions, and publish them together with a new version and date line.
- **What never goes in.** Key values or secrets of any kind, private member uploads, and promises about features that aren't live.
- **How to report an error.** Write to the project contact listed on the About page. Corrections are logged in the next version's notice.

Version history

Version	Date	Notes
1.2	October 15, 2026	Team roles and model routing updated: coordinator and specialist crew coordinate only; research farm adds ChatGPT 6 Astra alongside Fable and Argon (gating/security guardrail on Fable); writing farm with Opus 5.5 (via Farm Fable) and Argon; Antigravity presentation ownership across web, video, and audience-specific print formats; Builder independent review gate. Removed owner property privacy restrictions; real owner examples restored.

Version	Date	Notes
1.1	October 15, 2026	Owner uploads described as anything an owner has about the land (plans, reports, deeds, surveys, maps, letters, prior owners, family stories, photographs), with today's image-only upload kept distinct from planned file types. Presentation made a theme: today's standard report as the first of many planned views and formats. Figure 1.1 redrawn; Figures 9.1, 9.2 and 14.1 updated.
1.0	October 15, 2026	First edition. Outline from a four-model planning round; draft checked against the code; fifty numbered review fixes applied.

Colophon

This manual was written in Markdown. Its fifteen diagrams were drawn as Graphviz source files and rendered to SVG and PNG. It was compiled with Pandoc into Word, Large Print Word, and HTML, and the PDF was printed from the HTML, following the same docs-as-code approach the system uses for its own manuals (ADR-0005). Its outline came from a planning round across four AI models led by the lead planner. The draft was checked against the code and reviewed in a second independent round, and fifty numbered fixes were applied before publication.