They Might Say € MVP PDM (Lincoln-only)

Listening to History through AI

They Might Say € MVP Product Definition & Technical Build Spec (Standalone)

Last updated: Aug 18, 2025 | Owner: Tim Walsh | Audience: Technical Associate (outsourced), In•house Research & Production

1) Purpose & One•liner

Purpose. Build a shippable MVP for *They Might Say*: rigorously sourced, conversation•driven episodes between Tara (archivist•narrator) and historical personas, designed to evoke **awe** at human accomplishment while naming costs€ ,**Listening to History through AI**.*ƒ*

One•liner. An audio•first show with a primary•sourced Retrieval•Augmented Generation (RAG) stack that produces verifiable, non•moralizing conversations and episode notes with citations.

2) MVP Scope

In scope (MVP):

- One pilot episode: **Abraham Lincoln** (28„38 minutes). - Transcript with inline citations and a links appendix. - 12+ AC•01 Anecdote Cards per episode (see Appendix A schema). - Working RAG pipeline: ingest … chunk … embed … retrieve … ground … generate. - Admin console for source management, anecdote tagging, and retrieval QA. - Exporters: Markdown + PDF episode notes; JSONL citation bundle.

Out of scope (MVP):

- Video avatars, deepfake voices, advanced sound design. - Public search UI; multi•tenant SaaS. - Complex licensing ingestion (paid academic journals).

3) Success Criteria (Acceptance)

- **A1. Traceability & Credibility:** 100% of assertions in generated outputs carry **†1 highly credible citation** (Trust Tier 1„3; see ‡12) with click-through from transcript … snippet … source. If a paragraph relies on a single high-cred source, it is **explicitly marked** as **[single-source]** in the UI and footnoted in exports. - **A2. Retrieval Quality:** precision@5 † 0.75 on a Lincoln-only 100•prompt evaluation set; zero uncited tokens emitted. - **A3. Anecdotes:** † 12 AC•01 cards; each with a primary/near-primary source plus a critical/dissenting source **when available** (if none, card is marked [single-source]). - **A4. Conversational Experience:** Listener beta (n†50): †70% report ,**felt like a human conversation** (extemporaneous), not a reading.*ƒ* - **A5. Delivery:** One pilot (Lincoln), notes PDF, JSONL citations, and admin console shipped. - **A6. Language Policy Compliance:** 100% of persona lines pass the **Language & Register** rules (clear, modern English; **no contemporary slang or out•of•period idioms**); 0 critical linter flags in final exports.

4) Users & Roles

- **Host/Research Lead (in•house):** curates sources, defines prompts,

approves anecdotes. - **Producer/Editor (in•house):** assembles audio; ensures tone & legal. - **Technical Associate (outsourced):** builds data pipeline, APIs, admin app, CI/CD. - **Legal/Fact•check (in•house):** reviews claims, flags contentious areas.

5) User Stories (Top•priority)

1. As a Host, I can upload a primary source (PDF/EPUB/TXT/HTML) and see it chunked, embedded, and searchable within 2 minutes. 2. As a Researcher, I can create an **Anecdote Card (AC•01)** with Who/When/Where/What/Why + dual citations and mark reliability. 3. As Tara, I can answer prompts using only retrieved, cited snippets; any uncited token is blocked. 4. As Legal, I can export a citations JSONL bundle with source IDs, page ranges, and links. 5. As Producer, I can export episode notes (MD/PDF) with a Legacy Ledger: 3 benefits/3 harms.

5.5) Operational Flow & Studio Mode (MVP)

Goal: TA builds personas & tooling; Host creates the episode with live, conversational interactions.

Swimlanes - **Tech Associate (TA):** Build/maintain RAG, Admin, Studio Mode, Episode Builder; enforce guardrails. - **Host:** Outline, prompt, converse in Studio, pin beats, assemble episode; approve notes. - **Production:** Fact-check spot pass; record host VO; mix.

Phases - **Phase 0 € Character Packs (TA):** Deliver **Tara v1.0** and **Lincoln v1.0** with persona configs, prompts, exemplars, and corpus. Seed †12 AC•01 anecdotes. - **Phase 1 € Host Pre•Pro (Host):** Draft outline (Cold Open … Scope … Toolbox … Costs … Counterfactual … Takeaways … Source Spotlight). Build a **Prompt Deck** (10„20 prompts tagged by segment) + select 6„10 AC•01. - **Phase 2 € Rehearsal (TA+Host):** In **Studio Mode**, test prompts. TA tunes retrieval and persona constraints. Save **Studio Preset** (Lincoln v1.0 / Tara v1.0). - **Phase 3 € Recording (Host):** In **Studio Mode (Recording)**, ask prompts. Persona answers conversationally with per•paragraph citations. Pin great responses as **Beats**. Insert Tara cutaways and mark Read•Aloud excerpts. - **Phase 4 € Assembly (Host):** Use **Episode Builder** to order Beats into segments; auto•generate Notes + Legacy Ledger. - **Phase 5 € Fact•check (Production):** Open **Citations View**; verify high•cred citations (Tier 1„3). Single•source paragraphs kept if footnoted. - **Phase 6 € Voice & Mix (Production):** Record host VO (and optional TTS stems); mix; export.

Studio Mode € Functional Requirements - Prompt input, Answer pane (conversational style), Evidence pane (snippets with page/folio, trust tier). - Validator: **min 1 high•cred citation per paragraph**; if exactly one … badge **[single-source]** + optional 1•line scope note. - Controls: Regenerate (Shorter/More method/More cost/Narrow/With dissent); **Pin to Beat**; **Insert Tara Cutaway**; **Mark Read•Aloud**.

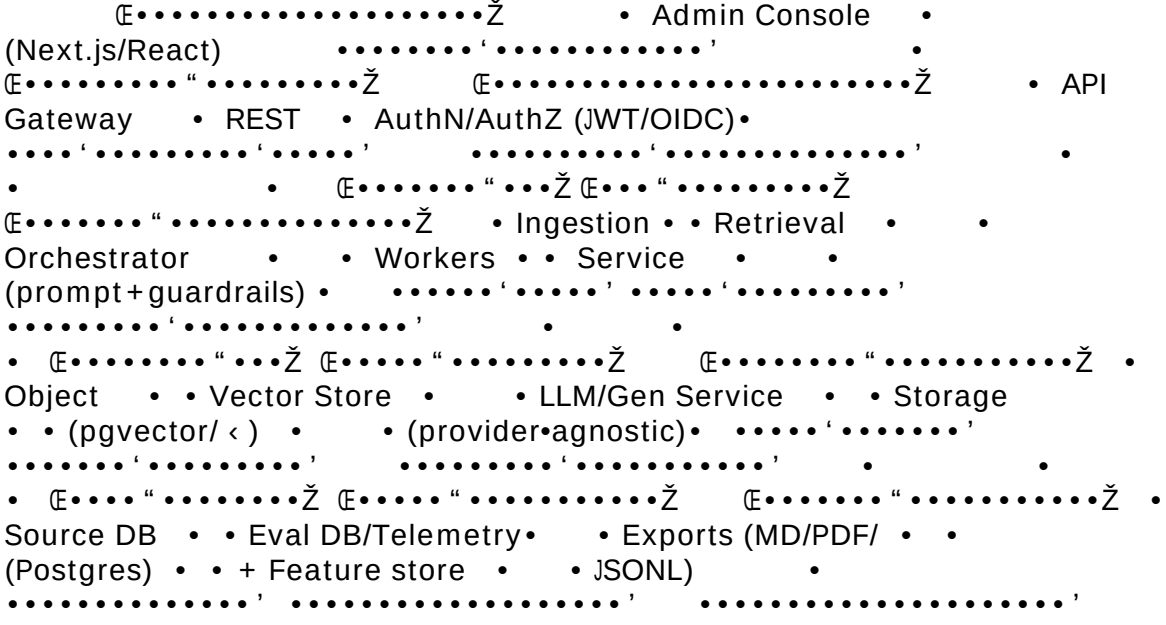
Episode Builder € Functional Requirements - Drag•drop Beats into segments; convert to Host Script; auto•build **Legacy Ledger** (3 benefits/3 harms, cited). - Export: Transcript (MD/PDF), Citations JSONL, Beatboard JSON.

6) Editorial Rails (for engineers to enforce)

- **Awe ˆ approval.** Output templates always include **Impact / Method / Cost / Myth**. - **Conversational, extemporaneous voice.** Personas **paraphrase** and speak naturally; direct quotes are used sparingly (‰10% of spoken lines; ‰60 words each) and flagged for Source Spotlight. Avoid ‚wooden*ƒ* recitations. - **Language & Register.** All historical personas speak **clearly understandable modern English** unless otherwise specified. **Never** use contemporary slang, memes, or idioms that post•date the figure. Keep archaic diction minimal; when a period term is essential, **immediately paraphrase** it in plain English. No ‚performative accents*ƒ* or faux period filler. - **Citations are not spoken** by personas; they appear in transcript/notes and the Evidence pane. - **AI disclosure.** Tara opens with a one•line disclosure (‚Išm an AI•constructed narrator ‹ *ƒ*) and the show notes repeat this. - **Single•source policy.** One highly credible source (Tier 1„3) is sufficient per paragraph; mark **[single-source]** when used without corroboration. - **No euphemisms for harm.** Red•team rules flag sanitized phrasing. - **Segment rail enforced**: Cold Open ... Scope ... Toolbox ... Costs ... Counterfactual ... Takeaways ... Source Spotlight.

7) System Architecture (MVP)

7.1 High•level diagram (ASCII)



7.2 Suggested stack (pluggable)

- **Backend/API:** Python (FastAPI) or Node (NestJS). *Pick one; examples below assume FastAPI.* - **Vector store:** Postgres + **pgvector** (preferred for control & cost) or managed (e.g., Pinecone/OpenSearch Vector) behind a repository interface. - **Object storage:** GCS (preferred) or S3 for raw files & exports. - **DB:** Postgres 14+ (Cloud SQL or RDS). Separate schemas: `sources`, `retrieval`, `eval`. - **Embedding model:** Provider•agnostic adapter (supports local OSS models and hosted). Config via env var. - **LLM:** Provider•agnostic (completion & JSON•mode support). Must support function/tool•calling or structured outputs. - **Admin App:** Next.js 14 (React), Tailwind, shadcn/ui. Auth via Google OIDC. - **Infra:**

Docker + Terraform; CI/CD via GitHub Actions to Cloud Run (GCP) or ECS Fargate (AWS). - **Observability:** OpenTelemetry … Grafana/Prometheus; log retention 30„90 days.

7.3 Environments

- **Local:** docker•compose; seeded with a small Lincoln sample set. - **Staging:** Cloud project with restricted secrets; test keys only. - **Prod:** Separate project; least•privilege service accounts; VPC egress controls.

8) Data Model (core tables)

8.1 `source`

- `id` (uuid), `title`, `author`, `year`, `type`, `provenance_url`, `license`, `trust_tier` (1„4), `notes`, `sha256`, `created_at`.

8.2 `document`

- `id`, `source_id` (fk), `path` (object storage URI), `pages_json` (optional OCR map), `ingest_meta` (jsonb).

8.3 `chunk`

- `id`, `document_id` (fk), `loc` (page/folio + line range), `text`, `tags` (Impact/Method/Cost/Myth + Era/Locale/Actor/DocType), `embedding` (vector), `quality_score` (float), `created_at`.

8.4 `anecdote`

- `id`, `title`, `who`, `when_date`, `where`, `what_happened`, `why_it_matters`, `primary_citation_id` (fk: `citation`), `critical_citation_id` (fk, nullable), `single_source` (bool), `reliability_notes`, `read_aloud_excerpt`, `status` (draft/approved), `tags`.

8.5 `citation`

- `id`, `chunk_id` (fk), `source_loc` (page/folio), `quote_span` (char idx), `url` (if web), `trust_tier` (1„4), `confidence` (float).

8.6 `persona_pack`

- `id`, `name` (e.g., Tara v1.0), `role` (tara|figure), `version`, `config_json` (style, banned euphemisms, disclosure, **language_policy**), `created_at`.

`config_json.language_policy` example: { "register": "modern-plain", "ban_modern_slang": true, "ban_out_of_period_idioms": true, "allow_archaic_terms": "minimal", "require_immediate_paraphrase": true }

8.7 `episode` `episode`

- `id`, `title`, `status` (draft/ready/published), `created_at`.

8.8 `beat`

- `id`, `episode_id` (fk), `segment` (cold_open|scope|toolbox|costs|counterfactual|takeaways|spotlight), `prompt_text`, `response_text`, `citations` (jsonb array), `single_source` (bool), `trust_tiers` (int[]), `confidence` (int 1„5), `created_at`.

8.9 `prompt_template`

- `id`, `name`, `role` (Tara/persona/system), `body`, `schema_uri`,

`version`.

8.10 `eval_case`

- `id`, `input`, `expected_citations[]`, `assertions[]`, `results_json`, `p_at_5`, `pass_fail`.

9) Ingestion Pipeline

Inputs: PDF, EPUB, TXT, HTML, CSV (metadata), images with OCR.

Steps:

1. **Upload** via Admin … object storage; compute `sha256` & MIME sniff. 2. **Normalization**: pdfminer/tesseract for OCR; epub…html…text; strip headers/footers; preserve page anchors. 3. **Chunking**: rule•based (semantic/length hybrid). Target 600„1200 chars; overlap 80„120 chars. Store `loc` anchors. 4. **Tagging**: rule•based + optional ML classifier to assign Impact/Method/Cost/Myth + metadata tags. 5. **Embedding**: configurable model; store in `embedding` (vector) + quality score. 6. **Indexing**: upsert into `chunk` with cross•refs; link to `source` + `document`. 7. **QA gate**: spot•check 5 random chunks; evaluate OCR confidence; flag low quality.

Re•ingest idempotency: refuse upload if `sha256` seen unless `force=true`.

10) Retrieval & Generation

10.1 Query flow

1. Build a **retrieval query** from host prompt + episode context (persona, era, tags). 2. Hybrid search: BM25 + vector; rerank top•50 … top•8. 3. **Citations constraint:** if fewer than 3 distinct sources, return ‚insufficient grounding.ƒ 4. Construct a **grounding bundle** (snippets + metadata) … Orchestrator. 5. Orchestrator runs **structured prompts** to produce: - **Cold open** (‰220 words) with at least 1 primary citation. - **Scope/Toolbox/Costs/Counterfactual/Takeaways/Spotlight** sections with per•section citations. 6. Output validator rejects any section lacking †2 citations or containing ‚uncited claims.ƒ

10.2 Prompt scaffolds (snippets)

- **System (Tara):** ‚You are Tara, archivist•narrator ‹ ”awe without absolutionŠ‹ never write a claim without citing SourceID[page]. Use neutral verbs. Include ”Legacy LedgerŠ.ƒ - **Validator:** JSON schema requiring `citations[]` per `paragraph_id`.

11) Admin Console (MVP features)

- Upload & ingest monitor (progress bars; OCR %, chunk count). - Search (filters: Tag, Era, Figure, Source Type; trust tier badges). - **Studio Mode** (Rehearsal & Recording): prompt input, answer & evidence panes, validator, pin to **Beats**, Tara cutaways, Read•Aloud markers. - **Language & Register controls:** toggle "modern•plain" vs. stricter period flavor; built•in **slang/idiom/anachronism linter** with inline warnings and quick•fix prompts ("rephrase in plain English; remove out•of•period idioms"). - **Episode Builder:** drag•drop Beats into segments; Host Script view; Legacy Ledger auto•calc; one•click exports. - **Anecdote Card editor**

with AC•01 fields, validation, status. - Retrieval QA: run saved prompts, inspect top•k, label relevance, export eval set. - Export center: MD, PDF, JSONL citations; episode kit ZIP. - Users & permissions: Admin, Researcher, Viewer.

12) Security, Legal, & Ethics

- **Auth:** OIDC (Google). Roles via JWT claims; backend checks RBAC per route. - **PII:** none expected; still encrypt secrets; no public uploads. - **Licensing:** MVP uses public•domain or licensed sources only; store `license` per `source`. - **AI disclosure:** A short spoken and written disclaimer identifies personas as AI•constructed; disclosure text stored in `persona_pack`. - **Trust tiers:** `source.trust_tier` (1=primary/near•primary; 2=peer•reviewed/university press; 3=high•quality reference; 4=other). Only Tiers 1„3 satisfy single•source rule. - **Language linter:** automated check for contemporary slang, memes, and out•of•period idioms; blocks export on critical flags unless Host overrides with a scope note. - **Red•team checks:** profanity/violence euphemism detection; contested claims prefer dissenting source; if unavailable, mark **[single-source]** + scope note. - **Audit log:** store prompt, retrieved chunk IDs, trust tiers, linter flags, and outputs with hash.

13) Observability & QA

- **Metrics:** ingestion latency, chunk error rate, retrieval P@k, % paragraphs with †1 Tier•1„3 citation, % paragraphs marked [single-source], **slang/anachronism flags per 1k tokens (target: 0)**, hallucination rate (0 allowed), token cost. - **Tracing:** spans for ingest, embed, retrieve, generate. - **Eval harness:** nightly run on Lincoln eval set; compute P@1/3/5 and MRR; alert on regression.

14) Deployment & DevOps

14.1 Repos

- `tms-api` (FastAPI) - `tms-admin` (Next.js) - `tms-workers` (Ingestion) - `tms-infra` (Terraform)

14.2 Environment variables (sample)

APP_ENV=staging DB_URL=postgresql+psycopg2://user:pass@host:5432/tms VECTOR_DB_URL=postgresql://user:pass@host:5432/tms EMBEDDINGS_PROVIDER=openai|local|vertex|cohere EMBEDDINGS_MODEL=text-embedding-xxx LLM_PROVIDER=openai|local|vertex|anthropic|cohere LLM_MODEL=name-or-endpoint OBJECT_BUCKET=gs://tms-sources-staging GOOGLE_CLIENT_ID=... GOOGLE_CLIENT_SECRET=... JWT_SIGNING_KEY=... MAX_CONTEXT_TOKENS=12000 CHUNK_SIZE=1000 CHUNK_OVERLAP=100

14.3 Local dev (docker•compose)

- Services: `api`, `admin`, `worker`, `postgres`, `minio` (S3•compatible), `otel•collector`. - Seed script loads a handful of Lincoln letters.

14.4 Cloud (GCP reference)

- Cloud Run services for `api`, `worker`, `admin`. - Cloud SQL (Postgres + pgvector extension). - Cloud Storage buckets: `tms-sources`, `tms-exports`. - Secret Manager for keys; Workload Identity; restricted egress.

(AWS alt: ECS Fargate, RDS Postgres + pgvector, S3, Secrets Manager.)

15) API (selected endpoints)

Auth - `POST /auth/callback` (OIDC) … JWT.

Sources & Ingest - `POST /sources` (multipart: file + metadata) - `GET /sources/{id}` - `POST /ingest/{source_id}` … async job id - `GET /ingest/{job_id}` … status

Search & Retrieval - `POST /search` {q, filters, k} … {chunks[]} - `POST /retrieve` {prompt, persona, filters} … {bundle}

Studio Mode - `POST /studio/ask` {prompt, persona, preset_id} … {paragraphs:[{text, citations, single_source, trust_tiers}]} - `POST /beats` {episode_id, segment, prompt_text, response_payload} … {beat_id} - `GET /beats?episode_id=…`

Episodes - `POST /episodes` {title} - `PATCH /episodes/{id}` {status} - `POST /export/episode` {episode_id, format}

Anecdotes - `POST /anecdotes` (AC•01 payload) - `GET /anecdotes?status=approved`

Eval - `POST /eval/run` … metrics

16) Prompting & Templates (snippets)

- **Conversational style primer (figures):** ‚Answer as if in an interview€ natural, concise, and reflective. **Paraphrase** sources; use first•person where historically appropriate. **Speak in clear, modern English. Do not use contemporary slang or idioms that are out of period for you.** If a period term is essential, state it, then immediately paraphrase in plain English. Avoid reading long quotations. If you must quote, keep it ‰60 words, and the quote will be moved to Source Spotlight.*ƒ* - **Tara system prompt:** archivist•narrator; ‚awe without absolution*ƒ*; no euphemisms for harm; include Legacy Ledger; one•line AI disclosure at open. - **Validator:** JSON schema requiring `citations[]` per `paragraph_id`, with at least **1** Tier•1„3 citation and `single_source` boolean; integrate **language_linter_flags** list. - **Cold•open template:** time/place precision, at least one high•cred citation. - **Toolbox of Genius:** 3 methods with evidentiary support; each includes snippet IDs. - **Costs & Contradictions:** name who paid (immediate/downstream/reputational). - **Legacy Ledger:** 3 quantified benefits + 3 harms (each cited).

17) Build Plan & Milestones (6„8 weeks)

W1: Infra skeleton, DB schema updates (trust tiers, episodes/beats, **language_policy**), local compose, minimal Admin auth. **W2:** Ingestion (PDF/TXT), OCR, chunking, embeddings, search API. **W3:** Retrieval orchestration, citation•constrained generation, **Studio Mode (alpha)** with rehearsal view, **language linter (baseline lexicon)**. **W4:** **Episode Builder** (beats … segments), anecdote editor, eval harness; Lincoln seed; precision@5 †0.6. **W5:** Conversational tuning, red•team checks, **linter tuning**, precision@5 †0.75; exports (MD/PDF/JSONL); Studio Preset save/load. **W6:** Beta test, polish, docs, handoff.

18) RACI (abbrev.)

- **Schema/DB:** Technical Associate (R), Tim (A), Research (C), Legal (I) - **Ingestion:** Technical Associate (R), Research (C) - **Prompts:** Research (R), Tim (A), Technical Associate (C) - **Eval:** Technical Associate (R), Research (C)

19) Risks & Mitigations

- **OCR quality on 19th•century prints** … dual OCR engines + manual QA queue. - **Hallucinations** … hard validator that rejects uncited spans. - **Licensing creep** … store `license` in source; ingest PD first; block uploads without license.

20) Deliverables (what you hand me)

1. Deployed staging + prod. 2. Admin Console with features listed in ‡11. 3. Seeded corpus (Lincoln) with †12 AC•01 anecdotes. 4. Pilot notes (MD/PDF) and citations JSONL. 5. DevOps: Terraform, CI/CD workflows, runbooks. 6. README with local dev instructions and .env.sample.

21) RFP & IP Language (paste into contract)

- **IP Ownership:** All code, prompts, datasets, embeddings, and outputs created under this SOW are **work•for•hire** and the exclusive property of Client (Tim Walsh). Contractor assigns all right, title, and interest worldwide to Client upon payment. - **Open•source:** Contractor may propose OSS; any copyleft licenses require prior written approval. Maintain a `THIRD_PARTY_LICENSES.md` file. - **Confidentiality:** All materials are confidential; no publicity without written consent. - **Deliverables Acceptance:** Acceptance tied to ‡3 criteria; payment milestones on W3/W6 delivery.

Appendix A € Anecdote Card (AC•01) Schema

Definition: A micro•narrative (‰250 words) directly involving the figure, or a tightly bound episode materially shifting our understanding of methods, impact, or costs. Impeccably sourced.

JSON Schema (abridged):

{ "$schema": "https://json-schema.org/draft/2020-12/schema", "type": "object", "required": ["title","who","when_date","where","what_happened","why_it_matters","primary_citation","critical_citation"], "properties": { "title": {"type":"string","maxLength":120}, "who": {"type":"string"}, "when_date": {"type":"string","format":"date"}, "where": {"type":"string"}, "what_happened": {"type":"string","maxLength":800}, "why_it_matters": {"type":"string","maxLength":400}, "read_aloud_excerpt": {"type":"string","maxLength":160}, "primary_citation": {"type":"object","required":["source_id","chunk_id","loc"],"properties":{"source_id":{"type":"string"},"chunk_id":{"type":"string"}, "loc":{"type":"string"}}}, "critical_citation": {"type":"object","required":["source_id","chunk_id","loc"],"properties":{"source_id":{"type":"string"},"chunk_id":{"type":"string"},"loc":{"type":"string"}}}, "reliability_notes": {"type":"string"}, "tags": {"type":"array","items":{"type":"string"}} } }

YAML Example:

title: Lincoln‰s Night at the Telegraph Office who: Abraham Lincoln when_date: 1864-08-10 where: Washington, D.C., War Dept. Telegraph Office what_happened: >- Lincoln spends late night hours reading casualty telegrams and revises a draft line ‹ why_it_matters: >- Shows executive poise under existential pressure; illustrates ‚telegraph cadence.*ƒ* read_aloud_excerpt: "I have not willingly planted a thorn in any man's bosom ‹ " primary_citation: source_id: src_lincoln_collected_works chunk_id: chk_00129 loc: vol7 p.211-212 critical_citation: source_id: src_press_correspondence_1864 chunk_id: chk_00987 loc: Aug 11 dispatch reliability_notes: Cabinet diary corroborates; time uncertain by •1 hr. tags: [Impact, Method, CivilWar, WashingtonDC]

Appendix B € Episode Export Templates

- **Markdown/PDF** with section headers and bracketed citations `[SourceID p.xx]`. - **Single•source footnote:** If a paragraph relies on one high•cred source, append `– Single•source statement (Tier X; edition info).` - **PDF styling:** single artifact image, dark cover, tagline: ‚Listening to History through AI.*ƒ* - **JSONL** per paragraph: `{text, citations:[...], single_source: true|false, trust_tiers:[...], confidence}`.

Appendix C € Evaluation Playbook

- Build 100 evaluation prompts (Lincoln-only). - Annotate expected relevant chunks (gold set). - Nightly job runs retrieval; compute P\@1/3/5 and MRR; send Slack alert on regression.

Appendix D € Red•Team Rules (Content)

- Flag euphemisms for atrocity; require direct naming (e.g., ‚enslaved people,*ƒ* not ‚servants*ƒ*). - Require dissenting source for reputational claims. - Block speculative psychoanalysis; label as inference if included.

Appendix E € Local Dev Quickstart

clone

mkdir tms && cd tms
(repos will be created as part of project setup)

bring up services

docker compose up -d
run DB migrations

alembic upgrade head # if Python
seed demo data

python scripts/seed_demo.py
open admin

open http://localhost:3000

Appendix F € Runbooks

- **Ingestion fails (OCR low):** re•run with high DPI; push to manual QA

queue. - **Retrieval weak (P@5 < .75):** adjust chunking, switch embedding model, add reranker. - **Hallucination detected:** inspect validator logs; raise citation threshold; retrain tagger. - **Language linter flags present:** open the Beat in Studio; click **Rephrase in plain English**; if the period term is necessary, add an immediate paraphrase; update persona `language_policy` or add term to allowed glossary if appropriate.

Appendix H € Quick Checklists

Host (before recording) - [ ] Episode outline & prompt deck ready - [ ] 6„10 AC•01 anecdotes selected - [ ] Studio Preset: Lincoln v1.0 / Tara v1.0 loaded

Tech Associate (before recording) - [ ] Retrieval P@5 † 0.75 on rehearsal prompts - [ ] Persona guardrails on (min 1 Tier•1„3 citation) - [ ] **Language linter** enabled; 0 critical flags on rehearsal set - [ ] Export templates validated (MD/PDF/JSONL)

After recording - [ ] Beats ordered into segments - [ ] Legacy Ledger populated with citations - [ ] Fact-check spot-pass 10/10 - [ ] Final exports delivered to editor