# Talk to Text Canada — Technical Brief for Developers

*\*\*Purpose:\*\* Give any incoming developer everything needed to understand the system, run it locally, deploy to WHC, and safely ship improvements without breaking core business rules.*

## 1) Product snapshot (what it MUST do)

* Services: \*\*AI-only\*\*, \*\*Human-only\*\*, \*\*Hybrid (Human+AI)\*\*, \*\*LegalScript Studio\*\* (legal document production).
* Upload → \*\*auto-transcribe with Speechmatics\*\* → client edits in dashboard → \*\*download\*\*.
* \*\*Download rules:\*\* AI/standard = \*\*locked PDF\*\* (non‑editable). LegalScript = \*\*editable .docx\*\*.
* \*\*Trial:\*\* up to \*\*3 uploads or 3 hours\*\* via coupon system under `InitialTrial/`.
* \*\*Payments:\*\* Stripe subscriptions + webhooks update client access.
* Legal clients can upload templates or use an \*\*Ontario Court Forms\*\* library.
* Branding: \*\*Navy `#003366`\*\* primary + light/dark lavender accents; logo on every page.

## 2) Tech stack & major components

* \*\*Backend:\*\* Flask (Python 3.11+). Primary entry: `app/webhook\_app.py` (merged version).
* \*\*ASR:\*\* Speechmatics (REST). Wrapper lives in `app/speechmatics\_handler.py` & `transcription/`.
* \*\*Docs:\*\* DOCX generation/merge, DOCX→PDF, \*\*PDF locking\*\* utility.
* \*\*Frontend:\*\* Flask/Jinja templates + SCSS compiled to `app/static`.
* \*\*Payments:\*\* Stripe (Checkout + Webhooks).
* \*\*Hosting:\*\* WHC (Web Hosting Canada) via cPanel \*\*Python App (Passenger/WGSI)\*\*.

## 3) Environment & secrets (do not commit)

`.env` lives in `config/.env` and is loaded by `dotenv` and `utils.init\_env()`:

* `SPEECHMATICS\_API\_KEY`, `SPEECHMATICS\_USER\_ID`, `SPEECHMATICS\_BASE\_URL`
* `FLASK\_SECRET\_KEY`
* `STRIPE\_SECRET\_KEY`, `STRIPE\_WEBHOOK\_SECRET`
* `ADMIN\_USERNAME`, `ADMIN\_PASSWORD`
* `CLIENT\_FILES\_DIR` (absolute or project‑relative base path for client data)

*\*\*Note:\*\* Any path used for uploads/finals must be \*\*writable\*\* on WHC.*

## 4) Data model (conceptual)

\*\*Client\*\* `{client\_id, subscription\_status, metadata}`

\*\*Project\*\* `{project\_id, client\_id, project\_name, intake\_data(JSON), status}`

\*\*Filesystem per project\*\*

client\_files/<client\_id>/projects/<project\_id>\_<project\_name>/

  uploaded\_content/

  transcripts/

  previews/

  final/                # FINAL OUTPUTS HERE

  intake/               # per-project JSON

  legal\_templates/      # optional client-provided

\*\*Canon file names:\*\*

* AI: `final/FinalDocument.pdf` (locked)
* LegalScript: `final/FinalDocument.docx`

## 5) Core routes & flows

### Upload & ASR

* `GET|POST /upload` → expects `client\_id`, `project\_name`, `file`.
* Saves to `uploaded\_content/` → calls `transcribe\_audio(client\_id, project\_name, path)`.
* Returns JSON `{status, transcript\_path}` on success.

### Editing

* `transcript\_editor.html` (AI clients): edit/format + tools; persists back to project.

### Downloads (strict rules)

* `GET /download/pdf/<client\_id>/<project\_id>`
* If missing, builds \*\*locked PDF\*\* from latest DOCX; \*\*disables editing\*\*.
* `GET /download/docx/<client\_id>/<project\_id>`
* LegalScript deliverables only (kept editable).

### Trial

* Views in `InitialTrial/templates/` (`upload.html`, `download.html`, `initial\_trial\_landing.html`).
* Logic in `InitialTrial/Modules/coupon\_handler.py` enforcing \*\*3 uploads / 3 hours\*\*.

### Intake & project control

* \*\*Intake CRUD\*\* per project (route present/expected): load/edit/save JSON used in merges.
* \*\*Project switcher\*\* in dashboard to move between existing projects.

### Stripe

* Checkout link creation (automation WIP).
* Webhooks call `update\_client\_meta\_payment`/`update\_client\_config\_billing` to toggle access.

### Policy pages

* Privacy Policy, Terms, NDA — render + allow \*\*.docx\*\* downloads.

## 6) Directory map (representative)

C:\DEV\TALKTOTEXTCANADA

│  README.md

│  requirements.txt

│

├─admin/

│   dashboard.html

│

├─app/

│  │  webhook\_app.py

│  │  client\_config.py

│  │  speechmatics\_handler.py

│  │  test\_flask.py

│  │  intake.json  # example schema

│  │  intake.ada   # example schema

│  │

│  ├─client\_files/

│  │   └─ <client\_id>/projects/<project\_id>\_<project\_name>/

│  │       ├─ uploaded\_content/

│  │       ├─ transcripts/

│  │       ├─ previews/

│  │       ├─ final/

│  │       └─ intake/

│  │

│  ├─templates/

│  │   index.html, welcome.html, upload.html, download.html,

│  │   transcript\_editor.html, about.html, about-founder.html (members-only),

│  │   thank\_you.html, admin pages, policy pages

│  │

│  └─static/images/

│

├─InitialTrial/

│  ├─ Modules/coupon\_handler.py

│  └─ templates/upload.html, download.html, initial\_trial\_landing.html

│

├─config/ .env

└─transcription/, utils/, generate\_pdf.py, ...

## 7) Branding & SCSS standards

* Primary: \*\*`#003366`\*\*; Accents: \*\*lavender (light/dark)\*\*.
* Keep \*\*American spelling\*\* in code identifiers; \*\*Canadian English\*\* in UI.
* Migrate Sass from `@import` → \*\*`@use`\*\*; define global tokens (colors/spacing/typography) and consume them everywhere.

## 8) What’s DONE vs TODO

\*\*DONE / in place\*\*

* Upload triggers Speechmatics transcription.
* Locked-PDF generation on demand (creates if missing) + route added.
* Trial system + coupon logic exists.
* Stripe integration present in prior `webhook\_app.py`; merged into current (needs validation).
* Members-only founder page; link from authenticated `welcome.html` pending.

\*\*TODO (highest impact first)\*\*

1. \*\*Finalize `webhook\_app.py` merge\*\* → remove duplicate routes, validate imports/paths, unify env loading.

2. \*\*Dashboard polish\*\*

* AI editor: templates, tools, autosave, better UX.
* LegalScript: template upload, “Finished Work” list, \*\*preview-before-download\*\*, subscription panel.

3. \*\*Intake CRUD\*\* (server + UI) and ensure merge pipeline uses intake data reliably.

4. \*\*Project switcher\*\* (UI + server queries) with safe access control per client.

5. \*\*Stripe\*\* → automate checkout link creation, verify webhook flows and metadata updates end-to-end.

6. \*\*Branding consistency\*\* on `about.html`, `pricing.html`, `index.html`, `thank\_you.html`.

7. \*\*SCSS\*\* → fix undefined variables, compile clean with `@use`.

8. \*\*WHC deployment\*\* → production `requirements.txt`, Passenger/WGSI config, runbook.

## 9) Developer quick-start (local)

1. Python 3.11+, create venv.

2. `pip install -r requirements.txt`.

3. Create `config/.env` with keys in §3.

4. Ensure `CLIENT\_FILES\_DIR` exists and is writable.

5. Run: `flask --app app/webhook\_app.py run` (or `python app/wsgi.py`).

6. Smoke test:

* Upload at `/upload`.
* Ensure transcript path returns.
* Check `/download/pdf/<client\_id>/<project\_id>` creates a locked PDF.
* Check `/download/docx/<client\_id>/<project\_id>` serves LegalScript doc.

## 10) WHC deployment checklist (Passenger)

* Create Python App in cPanel → point to \*\*WSGI entry\*\* (e.g., `app/wsgi.py`).
* Upload code \*\*without\*\* `venv/`; inside WHC virtualenv run `pip install -r requirements.txt`.
* Place `.env` in `config/` on server; verify absolute paths.
* Ensure `client\_files/` is writable; verify static file mapping.
* Configure Stripe Webhook URL to the deployed domain (HTTPS).

## 11) Acceptance criteria (MVP)

* End-to-end: upload → ASR → edit → download (locked PDF for AI; .docx for LegalScript).
* Trial strictly limits \*\*3 uploads / 3 hours\*\*.
* Stripe checkout + webhook flips subscription state; dashboard shows plan, status, expiry.
* Intake CRUD + project switcher fully functional.
* SCSS compiles clean; consistent branding and logo across all pages.
* All finals saved to `final/` with canonical names.

## 12) Security & compliance notes

* \*\*No emailing\*\* of transcripts; downloads are direct to the client device.
* \*\*PDF locking\*\* required for AI transcripts; LegalScript remains editable.
* Aim for \*\*Canadian data residency\*\* where practical; Speechmatics can be brought on‑prem later.
* Enforce auth on member‑only routes (about‑founder, dashboards, downloads).

## 13) Common pitfalls (“don’t break these”)

* Don’t change the \*\*download defaults\*\* (locked PDF vs editable DOCX).
* Don’t write outside `CLIENT\_FILES\_DIR` or break the folder naming.
* Don’t commit `.env` or secrets; don’t include `venv/` in deliverables.
* Don’t hardcode Stripe keys or product IDs in templates.

## 14) Glossary

* \*\*LegalScript Studio:\*\* Legal document creation workflow, editable outputs.
* \*\*Locked PDF:\*\* PDF with editing disabled, produced from DOCX when needed.
* \*\*Intake:\*\* Project-specific data used in merges/templates.

## 15) Contact

* \*\*Product Owner:\*\* Jennifer Ouellette — prefers direct, precise updates and fast iteration.