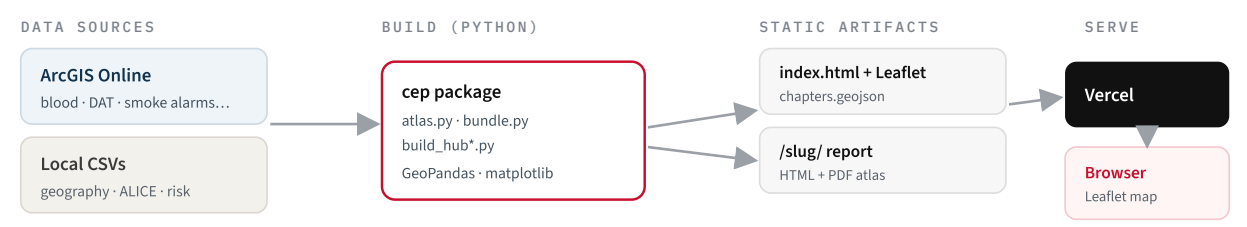


What You Built & How It Works

A static, Python-built web atlas — not an ArcGIS JavaScript app. · chapters.jbf.com · 2026-06-21

You built a **fast, free, fully-owned web map** plus **editorial chapter reports**. Python pulls data (some from ArcGIS Online) **at build time** and bakes it into static files; the browser just loads them. No Esri client license, no sign-in wall, no server — it loads instantly and costs almost nothing to run.

The pipeline



Key point: ArcGIS is only the **data well**, queried by Python with a token **during the build**. The live site never calls ArcGIS — viewers need no Esri API and no sign-in.

The stack

LAYER	TECHNOLOGY	ROLE
Hub map	Leaflet + CARTO tiles + GeoJSON	Interactive U.S. chapter map (client-side, vanilla JS)
Reports	Python — GeoPandas + matplotlib	Per-chapter editorial HTML + PDF atlas (static)
Build	Python — cep package + scripts	Pulls data, bakes JSON/HTML/PDF; no runtime server
Data	ArcGIS Online (token) + local CSVs	Mission metrics, geography, demographics, ALICE, risk
Host	Vercel (static)	chapters.jbf.com — instant, free-tier, no backend

Key files

- scripts/build_hub.py — builds the Leaflet hub (index.html)
- scripts/build_hub_data.py — chapter polygons + four-color division/region coloring
- src/cep/render/atlas.py — per-chapter report (HTML + PDF)
- src/cep/bundle.py — assembles a chapter's data bundle (AGOL + CSV, joined by FIPS)
- scripts/design_check.py — design-standard conformance gate

The pattern (how you do this)

- **Build static, not dynamic.** Python does the heavy work once; the web serves plain files.
- **ArcGIS at build time, never on the client.** Pull with a token, bake the result — fast + no auth wall.
- **Join by FIPS, display the name.** County data keyed by code; names are labels only.
- **Leaflet over the ArcGIS JS API** for public maps — lighter, free, fully yours.
- **One design system** (franken-app-standards): Inter + IBM Plex Mono, gray/white/red, enforced by a conformance gate.