

# Product Blueprint: Wrought Iron (WI)

## The Military-Grade, Air-Gapped Data Foundry

**Version:** 1.0 (Final Architecture) **Classification:** Enterprise Offline DataOps Platform **Core Philosophy:** "Secure by Design." Zero egress, fully deterministic, auditable, and local.

## 1. Executive Summary

**Wrought Iron** is a comprehensive Command Line Interface (CLI) and Terminal User Interface (TUI) platform designed for the surgical management of sensitive data.

Unlike cloud-native tools (Snowflake, Databricks) that require data to leave the premise, Wrought Iron brings the power of a modern data science stack (`Pandas`, `Scikit-learn`, `Geopandas`) **to the data**, residing locally on secure hardware. It transforms raw, corrupted, or unstructured SQLite databases into pristine, gold-standard datasets through a 12-module operational pipeline.

### Target Audience:

- **Defense & Intelligence:** Handling classified personnel files.
- **Healthcare:** Managing patient records on offline servers.
- **Government:** Processing historic census or archival OCR data.

## 2. Technical Architecture

Wrought Iron is built on a modular Python architecture. It abstracts complex data engineering tasks into a streamlined Subject-Verb-Object syntax (`wi [module] [action]`).

### The Stack

- **Core Engine:** `Pandas` (In-memory analytics), `SQLite` (Storage), `Polars` (High-performance alternative).
- **Interface Layer:** `Typer` (CLI Parsing), `Rich` (Output Formatting), `Textual` (Interactive TUI).
- **Intelligence Layer:** `Scikit-learn` (ML/Imputation), `Rapidfuzz` (String Matching), `Microsoft Presidio` (PII Detection).
- **Geospatial Layer:** `Geopandas`, `Shapely`, `Leaflet` (via offline HTML export).
- **Security Layer:** `Cryptography` (Fernet), `ReportLab` (PDF Logs), `Hasulib`.

### The Dynamic Data Model (The "Backpack" System)

To ensure universality, Wrought Iron utilizes a **Schema-On-Read** approach:

1. **Connection:** Connects to any standard SQLite file.
  2. **Expansion:** Automatically detects columns containing JSON strings (the "Backpack").
  3. **Flattening:** Explodes JSON keys into virtual DataFrame columns on the fly, allowing users to query, graph, and clean fields that did not exist when the database was created.
- 

## 3. The 12-Module Command Matrix

This system provides **120+ specialized commands** across 12 distinct operational domains.

### I. Infrastructure & Access

#### 1. `wi connect` (The Gatekeeper)

*Manages the lifecycle of database files and secure connections.*

- **Key Capabilities:** File encryption at rest, merging distinct DBs, vacuuming storage.
- **Critical Commands:**
  - `wi connect merge [DB_A] [DB_B]` (Consolidate fragmented data)
  - `wi connect encrypt [PATH]` (Lock file with AES-256)

#### 2. `wi schema` (The Librarian)

*Deep inspection and structural management.*

- **Key Capabilities:** Schema drift detection, JSON flattening, column type casting.
- **Critical Commands:**
  - `wi schema inspect [TABLE]` (Full statistical profile of every column)
  - `wi schema detect-json [TABLE]` (Find hidden nested data)

#### 3. `wi query` (The Explorer)

*Flexible retrieval and filtering.*

- **Key Capabilities:** SQL direct access, Pandas boolean filtering, full-text search.
  - **Critical Commands:**
    - `wi query filter [TABLE] --where "confidence < 0.6"`
    - `wi query dups [TABLE] [COL]` (Instant duplicate detection)
- 

### II. Analytics & Visualization

#### 4. `wi aggregate` (The Analyst)

*Commercial-grade statistics engine.*

- **Key Capabilities:** Pivoting, skew/kurtosis analysis, complex group-bys.
- **Critical Commands:**

- `wi aggregate crosstab [TABLE] [COL_A] [COL_B]` (Matrix analysis)
- `wi aggregate moving-avg [TABLE] [COL]` (Trend analysis)

## 5. `wi plot` (The Visualizer)

*Terminal-native charting.*

- **Key Capabilities:** Rendering high-fidelity ASCII charts without a GUI.
- **Critical Commands:**

- `wi plot heatmap [TABLE] [COL_A] [COL_B]` (Density visualization)
- `wi plot matrix [TABLE]` (Scatter matrix for correlation)

## 6. `wi report` (The Reporter)

*Portable, offline Business Intelligence.*

- **Key Capabilities:** Generating self-contained HTML dashboards and PDF briefings.
- **Critical Commands:**

- `wi report generate [TABLE]` (Sweetviz-style HTML dashboard)
- `wi report diff [SNAPSHOT_A] [SNAPSHOT_B]` (Before/After impact analysis)

## III. Wrangling & Intelligence

### 7. `wi clean` (The Wrangler)

*Algorithmic repair of broken data.*

- **Key Capabilities:** Cohort-based imputation, regex repair, text harmonization.
- **Critical Commands:**

- `wi clean impute-group [TABLE] [TARGET] --by [COHORT]` (Fixes corrupted source files)
- `wi clean fuzzy-match [TABLE]` (Resolves "J. Smith" vs "John Smith")

### 8. `wi ml` (The Scientist)

*Predictive modeling and classification.*

- **Key Capabilities:** Training classifiers, anomaly detection, KNN imputation.
- **Critical Commands:**

- `wi ml train-classifier [TABLE] [TARGET]` (Predicts missing categorical data)
- `wi ml detect-anomalies [TABLE]` (Isolation Forest outlier detection)

## 9. **wi geo** (The Geographer)

*Spatial intelligence system.*

- **Key Capabilities:** Coordinate normalization, catchment clustering, offline mapping.
  - **Critical Commands:**
    - `wi geo cluster [TABLE] --eps 0.5` (DBSCAN grouping)
    - `wi geo heatmap` (Terminal density map)
- 

## IV. Governance & Operations

### 10. **wi audit** (The Sentinel)

*Forensics, compliance, and tamper-proofing.*

- **Key Capabilities:** Cryptographic hashing, PII scanning, chain-of-custody logs.
- **Critical Commands:**
  - `wi audit hash-verify [TABLE]` (Ensures data integrity)
  - `wi audit export-cert` (Generates PDF compliance certificate)

### 11. **wi ops** (The Governor)

*Automation and pipelines.*

- **Key Capabilities:** Local cron scheduling, drift monitoring, pipeline execution.
- **Critical Commands:**
  - `wi ops schedule create [CMD] --cron "@daily"` (Automated night cleaning)
  - `wi ops drift-check [TABLE]` (Alerts if data distribution shifts)

### 12. **wi collab** (The Collaborator)

*Team synchronization without cloud sharing.*

- **Key Capabilities:** Sharing views, cleaning recipes, and annotations.
  - **Critical Commands:**
    - `wi collab config export` (Share cleaning parameters as JSON)
    - `wi collab recipe bundle` (Package entire project logic for transfer)
- 

## 4. The "Command Center" (TUI)

[Image of terminal user interface dashboard]

While the CLI is for automation, the **Command Center** (`wi interact`) is for exploration. It leverages **Textual** to provide a full application inside the terminal window.

- **Excel-Like Grid:** Navigate millions of rows with keyboard/mouse.
  - **Live Dashboard:** Real-time CPU/Disk metrics and cleaning progress.
  - **Visual Query Builder:** Checkboxes and sliders to filter data visually.
- 

## 5. Security & Compliance Strategy

---

Wrought Iron competes on security. It assumes the environment is hostile or compromised.

1. **Air-Gap Guarantee:** The codebase contains **zero** external network calls. No analytics phone home.  
No API dependencies.
  2. **At-Rest Encryption:** Supports AES-256 encryption for the SQLite file itself (`wi connect encrypt`).
  3. **Forensic Logging:** Every data mutation is logged to an immutable internal table and can be exported as a signed PDF.
  4. **PII Shielding:** Integrated PII scanner (`Microsoft Presidio`) proactively flags potential leaks in unstructured text fields.
- 

## 6. The "Golden Workflow"

---

This is the standard operating procedure for a Data Engineer using Wrought Iron:

1. **Ingest:** `wi connect file historic_data.db`
2. **Profile:** `wi schema inspect students` (Identify 40% missing birth years).
3. **Wrangle:** `wi clean impute-group students birth_year --by source_file` (Statistical fix).
4. **Enrich:** `wi geo normalize students suburb` (Add Lat/Lon).
5. **Dedupe:** `wi clean fuzzy-match students` (Merge identities).
6. **Verify:** `wi audit hash-create students` (Lock state).
7. **Report:** `wi report generate students --out briefing.html` (Send to Command).