

## TECHNICALLY SPEAKING: AI-POWERED ENGINEERING INSIGHTS

Build AI systems that generate **publishable, decision-grade knowledge products** (e.g., briefs, case studies, comparative reports, opinion pieces) grounded in trusted E4C content and responsibly enhanced with open/public datasets.

### 🕒 What You'll Build

- A workflow that ingests E4C sources and selectively incorporates external datasets (e.g., Kaggle, open gov data) for quantitative context (trends, comparisons, scale).
- An AI-assisted drafting tool or pipeline that synthesizes evidence into a coherent narrative for real decision-makers.
- Traceability features (citations, source mapping) and fact-checking steps.

### 🕒 Must Include

- Clear sourcing and claim traceability (every claim can be traced to E4C content and/or documented data).
- Transparent data use (what, why, and how external data was used).
- Explicit assumptions, limitations, and uncertainty.
- Human-in-the-loop verification: AI supports interpretation and synthesis; it does not generate technical validation, policy prescriptions, or causal claims.

### 🕒 Deliverables

- A working demo (web app, notebook, or pipeline) that produces a finished, readable product.
- A sample output (e.g., brief/report) with full references.
- A short methods note explaining the data flow, guardrails, and fact-checking.

## Examples

A **regional policy brief** contextualizing E4C clean-cooking solutions with adoption and health-burden data.

A **comparative market report** linking E4C-covered energy technologies to electrification and income trends.

A **data-informed opinion piece** that uses public datasets to support (not assert) normative arguments.

## CHECK MY TECH: READINESS REVIEW WITH AI

Build tools that help innovators **assess their technology** by mapping their documentation onto E4C's Solutions Library technology assessment framework. Support innovators on submitting their product for inclusion in the Solutions Library.

### ● What You'll Build

- An assessment assistant that evaluates completeness, required metadata, and clarity (e.g., deployment model, distribution pathways, performance metrics, maintenance, servicing).
- Explainable scoring and feedback mapped to E4C criteria.
- Comparative cues (where relevant) to show how reported performance relates to peer technologies.

### ● Must Include

- Advisory outputs (support improvement; do not determine acceptance or technical validity).
- Transparent logic: every score/flag references the criterion used and the source of comparative data.
- Clear guardrails to prevent overstated claims or automated "pass/fail" decisions.

### ● Deliverables

- A working demo that accepts a submission (doc/form) and returns a readiness snapshot (what's complete, partial, missing).
- A structured feedback report with specific, actionable next steps.
- A brief methods note covering criteria mapping, data sources, and guardrails.

## Examples

#### Distribution & deployment gaps:

Flags missing distribution channels, maintenance responsibilities, or local servicing details for a water treatment technology.

#### Performance in context:

Highlights where a solar refrigeration solution's efficiency is below the median of similar Solutions Library entries and links to the metrics used.

#### Readiness snapshot:

Summarizes which E4C criteria are fully addressed, partially addressed, or missing—before submission.

### FROM CONSTRAINTS TO CONCEPTS: AI-AIDED DESIGN

Build AI tools that combine **creative exploration** with trusted **engineering knowledge** from E4C to support **early-stage design**—helping teams explore ideas, understand tradeoffs, and iterate faster **without replacing human judgment**.

#### 🕒 What You'll Build

- A design assistant that surfaces relevant precedents from E4C, benchmarks existing solutions, and makes real-world constraints (cost, materials, environment, deployment context, user needs) explicit and actionable.
- Interfaces that support compare/contrast across design options and explain tradeoffs.
- Human-in-the-loop flows that encourage better questions and consequence awareness as concepts evolve.

#### 🕒 Must Include

- Design reasoning tools (not publishable reports or automated evaluations).
- Explainability for suggestions and comparisons (where did guidance come from; what assumptions are in play).
- Clear scope boundaries—the goal is exploration, not final recommendations or validation.

#### 🕒 Deliverables

- A working demo (e.g., design canvas, exploration interface) showing tradeoff surfacing and knowledge grounding in E4C content.
- A short walkthrough of how precedents are retrieved, how comparisons are made, and how constraints are modeled.
- Example session(s) demonstrating an iterative design workflow.

### Examples

A **concept exploration tool** that surfaces relevant E4C precedents and flags likely cost or deployment tradeoffs as designs evolve.

A **design comparison interface** that clarifies how material / configuration choices perform in similar real-world contexts.

An **AI-assisted design canvas** prompting teams to consider maintenance, distribution, and user constraints early.