AI Politician

An evidence-constrained political-reasoning system for Australia

Personal project · Mason Tatafu · Python · SQLAlchemy/SQLite · Claude API · FastAPI · Jinja2 · pytest

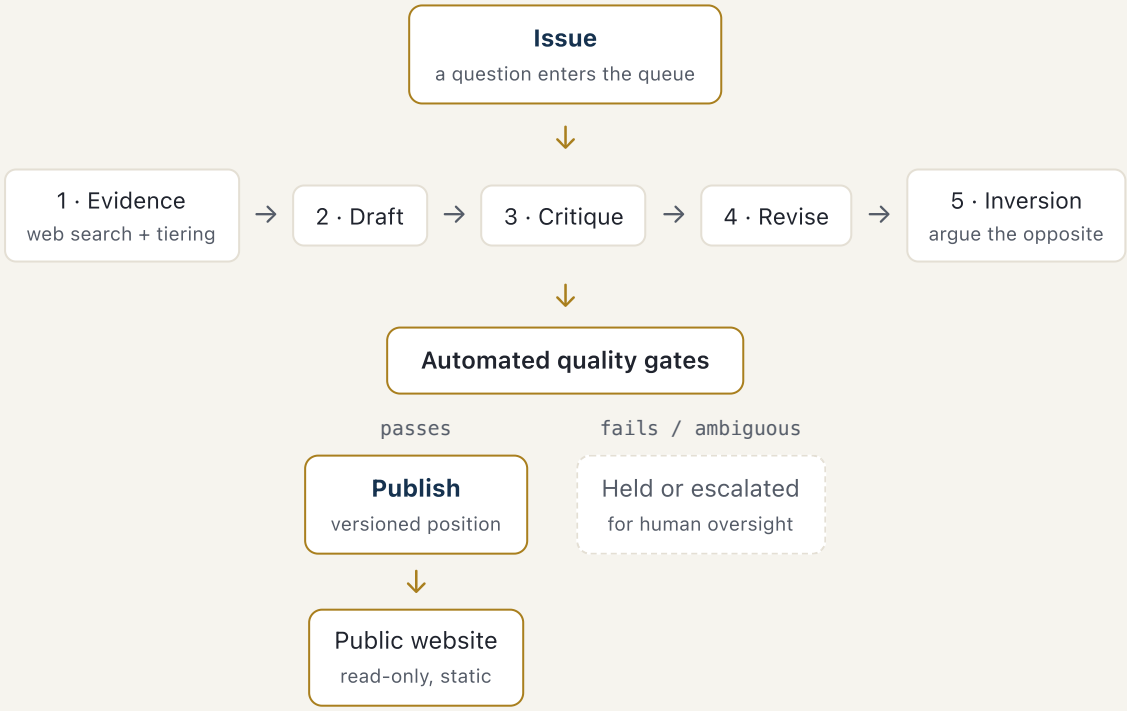
01 The problem, and the idea

Real politicians break promises, hide who they answer to, and leave the public unable to tell what's true. This project is the opposite: a system that **cannot lie about its reasons, cannot be swayed by money, and cannot change its mind without telling you why.**

You give it a real question — "*Why does Australia sell uranium to India but not use nuclear power at home?*" — and it researches the question against real sources, writes an evidence-based answer, and publishes it with the full reasoning, the sources, and a confidence level attached. The public sees a read-only website; there is no chatbot and nothing to submit.

02 How a question becomes a published position

Every question runs through a fixed pipeline, and automated quality gates decide whether the result is trustworthy enough to publish. Anything that fails is held or escalated for a human. Every step is written to a tamper-proof log.



Every step is written to a tamper-evident audit log (SHA-256 hash chain).

Figure 1 — The deliberation pipeline and its publish / hold / escalate gates.

03 What keeps it honest

The point of the whole system is trust, so everything that would otherwise rely on taking its word for it is made checkable:

It shows its work

Every answer links to real sources and keeps the full record of how it was written.

It fact-checks its own numbers

Before publishing, it re-reads its sources and blocks anything it can't verify.

It measures its own fairness

It scores whether it treats each party by the same standard, and publishes the score.

It argues the other side

Every answer is re-argued from the opposite view to catch bias.

It notices when it's out of date

Old answers are flagged for review as their facts age.

It can be challenged

Anyone can file a correction, which forces a revised version — the old one stays on record.

It keeps score of itself

Its predictions are tracked, so you can later check whether it was right.

It can't be secretly edited

Everything is in a tamper-proof log with a published fingerprint that changes if history is altered.

04 Under the hood

Each stage of the pipeline is a call to a large language model whose output is **validated against a JSON schema**, so the system works with clean structured data instead of unpredictable prose. A position only publishes if it clears every automated gate: enough credible, primary-anchored, corroborated sources; a **substantiation check** that re-fetches each source and confirms the cited figures actually appear in it; a bias scan; and a cross-position consistency check. The distinctive engineering is the self-checking layer on top:

COMPONENT	WHAT IT DOES
`symmetry`	Scores even-handedness across the party assessments using statistical dispersion, and publishes the number.
`freshness`	Gives each position a review horizon by claim type and re-queues stale ones — self-correction against time.
`substantiate`	Re-reads cited sources and blocks any figure it can't find in them.
`corrections`	Right-of-reply: a filed challenge forces a revised, superseding version.
`calibration`	Registers falsifiable, time-bound claims and measures whether confidence tracks reality.
`audit + attest`	SHA-256 hash-chained log; a published fingerprint makes tampering detectable from outside.

05 Data model

A normalised relational database built around **versioning**: a policy area keeps a full history of position versions, each tied to the values framework it was reasoned from, the sources it cites, and the verbatim reasoning log. Nothing is overwritten — a revision adds a new version and supersedes the old one.

RELATIONSHIPS `issues` → `policies` → `position_versions` (a policy's full version history). Each position version is reasoned from a `constitution_version`, cites many `sources` (via `position_sources`), records its `reasoning_logs`, and registers `predictions`. PK primary key · FK foreign key · UK unique key. Core of the 12-table schema.

Figure 2 — The versioned relational schema (core tables).

06 Outcomes

59	12	29	68
PUBLISHED POSITIONS	DATABASE TABLES	MODULES · ~5,400 LOC	AUTOMATED TESTS

A complete, self-auditing public record: the full pipeline, autonomous quality gates, bias and fairness measurement, evidence-freshness tracking, a corrections channel, a calibration ledger, and a tamper-evident audit log with external attestation — all built and passing tests.

Skills demonstrated: `Python` `relational data modelling (SQL)` `SQLAlchemy ORM` `REST APIs (FastAPI)` `LLM application & pipeline design` `structured outputs / schema validation` `data-integrity engineering (hash chaining)` `automated testing (pytest)` `static site generation` `Git`

AI Politician — an AI political-reasoning system that produces sourced analysis and then holds itself accountable: it fact-checks its own sources, measures its own bias, and keeps a record no one can quietly edit.