ChatGPT

ClearSide Product Specification (v0.1)

Introduction

ClearSide is an AI-powered structured reasoning and debate engine designed to help users think clearly about complex, high-stakes questions. Rather than producing a single answer or recommendation, ClearSide slows thinking down by generating steel-man arguments **for** and **against** a proposition, moderating the disagreement, and surfacing the assumptions and uncertainties that drive disagreement. It is a thinking support system, not an opinion generator.

Why ClearSide

Modern decision-making is overwhelmed by opinions but underserved in structured reasoning. Most AI tools tend to collapse complex questions into a single answer, hide assumptions and value judgments, overstate certainty, or reward rhetorical dominance rather than clarity. This leads to poorly understood trade-offs, false confidence in AI outputs, shallow debate and decision fatigue. ClearSide closes this gap by orchestrating adversarial reasoning and making the internal logic transparent.

Success Metric (North Star)

> **Did the user understand the issue better than when they started?**
> ClearSide measures success by clarity and understanding, not clicks or final answers.

Target Audience and Use Cases

ClearSide serves people and teams who need to examine complex decisions with structured reasoning:

- **Civic & Policy Analysis:** AI regulation, data center moratoria, climate policy, housing, healthcare and other public policy debates.
- **Professional & Technical Decision Support:** Architecture trade-offs, build vs buy, risk vs speed, migration decisions and other engineering or business choices.
- **Personal Decisions:** Career moves, major purchases, lifestyle choices and other personal trade-offs.

In all cases, ClearSide acts as a thinking partner rather than an advice engine. It reveals the trade-offs and competing values that underlie different positions.

Unique Value Proposition

ClearSide differs from generic pros-and-cons generators or chatbots in several ways:

- **Dual adversarial agents:** separate AI agents create the strongest possible arguments for and against the proposition, preventing one-sided outputs.

- **Steel-man enforcement:** agents must assume competent critics and cannot set up straw-man arguments or trivial objections.
- **Moderator synthesis:** a neutral moderator agent identifies areas of agreement, core disagreements, assumption conflicts, evidence gaps and decision hinges without choosing a winner.
- **Transparent assumptions:** each side explicitly lists assumptions and uncertainties, allowing users to question or re-weigh them.
- **User challenge loop:** after reading the debate, users can challenge specific assumptions or ask for stronger counterarguments, and the system replies inline.

ClearSide positions AI as a cognitive partner that surfaces reasoning rather than substituting for human judgment.

Core Components & Features

1. **Topic Intake**
   The user enters a question or claim and optional context (geography, timeframe, domain, stakeholders). The orchestrator agent normalizes this into a neutral proposition.

2. **Pro Advocate (Agent 1)**
   Generates the strongest possible case **for** the proposition. Arguments are organized by category (economic, ethical, technical, social, etc.) and must explicitly state assumptions, evidence types and confidence levels. No rebuttals to the con side are allowed.

3. **Con Advocate (Agent 2)**
   Generates the strongest possible case **against** the proposition under the same constraints. Assumptions and uncertainties are explicit, and trivial objections or emotional dismissals are forbidden.

4. **Moderator / Synthesis Agent (Agent 3)**
   Receives the pro and con outputs and synthesizes without taking a side. It surfaces areas of agreement, core disagreements (with root causes), assumption conflicts, evidence gaps and decision hinges. The moderator never recommends an action or picks a winner.

5. **User Challenge & Cross-Examination (Agent 4 – optional)**
   In the MVP this is a simple challenge function. Users can question an assumption, ask for a stronger counterargument or inquire what evidence would change the outcome. The challenge agent analyzes the targeted element and responds with dependencies and classification (factual, uncertain, values-dependent). Future versions could add a cross-examiner that actively stresses both sides.

6. **Output Formats**

7. **Quick View:** concise bullet lists for pros and cons with confidence and uncertainty flags.
8. **Structured Report:** an essay-style breakdown with executive summaries, detailed arguments by category, assumption lists and moderator synthesis.

9. **Debate Script:** formatted like a debate with opening statements, rebuttals, cross-examination and closing arguments.
   All outputs separate facts from inference, preserve uncertainty and explicitly list assumptions.

10. **MVP Features**

11. Single-question input form with optional context.
12. Pro advocate and con advocate generation.
13. Moderator synthesis.
14. Structured output in Markdown and JSON.

15. Basic challenge actions (question an assumption, ask for stronger counterargument, etc.).

16. **Future Features (Phase 2+)**

17. Citations toggle to label sources or confidence with references.
18. Value-weighting sliders so users can re-weight economic, ethical, social and technical factors.
19. Persona debates (economist vs ethicist vs environmentalist).
20. Saved debates and shareable links.
21. Account system and persistence.

Design Principles & Guardrails

- **Single-responsibility agents:** each agent has one job; no agent is allowed to win the argument.
- **No final answers:** ClearSide never decides or recommends. It preserves disagreement and uncertainty.
- **Explicit assumptions:** every argument lists its premises so users can challenge them.
- **Evidence classification:** arguments distinguish between facts, projections, analogies and value judgments.
- **Uncertainty preservation:** no false certainty; unknowns are surfaced as decision hinges.
- **User autonomy:** the user can challenge any element and explore deeper reasoning.
- **Neutral tone:** professional, non-chatty language; clarity over rhetoric.

These guardrails ensure ClearSide remains a reasoning support system rather than a persuasion or recommendation engine.

Agent Architecture

ClearSide is built on a multi-agent architecture where each agent outputs a specific section of a shared JSON schema. Agents do not see each other's work until the appropriate stage, preventing premature convergence.

Agents and Roles

Agent	Role	Mission	Hard Rules
Orchestrator	Normalization & Routing	Normalize the user question, extract context (geography, timeframe, domain) and dispatch tasks to pro and con advocates.	Cannot add arguments or make judgments. Must output only the `proposition` section of the schema.
Pro Advocate	Steel-man FOR	Construct the strongest possible case **for** the proposition. Organize arguments by category; list assumptions, evidence type and confidence; include uncertainties.	No straw-man arguments. No rebuttals. Cannot soften or hedge. Must output only the `pro` section.
Con Advocate	Steel-man AGAINST	Construct the strongest possible case **against** the proposition under the same rigor.	No trivial objections. No emotional dismissals. Treat the pro position as intelligent and serious. Output only the `con` section.
Moderator	Synthesis & Critique	Evaluate pro and con arguments without choosing a side. Identify areas of agreement, core disagreements, assumption conflicts, evidence gaps and decision hinges.	Cannot pick a winner or recommend a course of action. Must output only the `moderator` section.
Challenge Agent (optional)	Incremental Stress Test	Respond to user-initiated challenges by analyzing a specific assumption or claim. Provide dependency analysis, historical context and classify the challenge as factual, uncertain or values-dependent.	Do not regenerate the full debate. Output only the relevant element of the `challenges` section.

Interaction Flow (MVP)

1. User submits a question or claim with optional context.
2. **Orchestrator** normalizes the proposition and passes it to pro and con advocates.
3. **Pro Advocate** and **Con Advocate** independently generate their sections.
4. **Moderator** receives both sections, synthesizes, and outputs the `moderator` section.
5. The system assembles the final JSON object and renders the output in the UI.
6. The user can challenge assumptions or ask for deeper reasoning. The **Challenge Agent** handles the request and appends a response in the `challenges` section.

Agents never see more than necessary; for example, the con advocate cannot see the pro advocate's arguments before writing its own. This fosters robust adversarial reasoning.

JSON Output Schema (v1)

The ClearSide schema is the contract between AI agents, the UI, and any future API. It preserves assumptions, separates facts from inference, and maintains uncertainty. The schema is versioned (`schema_version: "1.0.0"`) to allow backward-compatible evolution. A simplified overview is below (keys are illustrative):

```
{
  "meta": {
    "schema_version": "1.0.0",
    "generated_at": "2025-01-20T18:42:00Z",
    "model_info": { "provider": "openai", "model": "gpt-5.2" },
    "confidence_level": "medium",
    "notes": "No citations requested"
  },
  "proposition": {
    "raw_input":
"The US should impose a temporary moratorium on new large-scale AI data center
construction.",
    "normalized_question": "Should the United States impose a temporary
moratorium on new large-scale AI data center construction?",
    "context": { "geography": "United States", "timeframe": "2025–2030",
"domain": "AI governance and public policy" }
  },
  "pro": {
    "executive_summary": [],
    "arguments": [ { "category": "", "claim": "", "explanation": "",
"evidence_type": "", "confidence": "" } ],
    "assumptions": [],
    "uncertainties": []
  },
  "con": {
    "executive_summary": [],
    "arguments": [ { "category": "", "claim": "", "explanation": "",
"evidence_type": "", "confidence": "" } ],
    "assumptions": [],
    "uncertainties": []
  },
  "moderator": {
    "areas_of_agreement": [],
    "core_disagreements": [ { "topic": "", "description": "" } ],
    "assumption_conflicts": [ { "pro_assumption": "", "con_assumption": "" } ],
    "evidence_gaps": [],
```

```
    "decision_hinges": []
  },
  "challenges": {
    "available_actions": [ "question_assumption", "stronger_counterargument",
"evidence_that_changes_outcome" ],
    "responses": [ { "action": "", "target": "", "response": { "analysis": [],
"historical_context": [], "classification": "" } } ]
  }
}
```

Schema Guarantees

- **Explicit assumptions and uncertainties** remain visible; the system never collapses them.
- **Evidence type and confidence** fields classify claims as facts, inference, historical analogy or projection.
- **UI-safe and diff-friendly**: the keys and structure remain stable across versions, enabling regression testing of reasoning quality.

Prompt Contracts & Enforcement

ClearSide uses strict prompt templates that map directly to sections of the JSON schema. Models are never asked to answer the question; they are asked to populate their assigned section of the schema. Enforcement rules include:

- **Global system prompt:** all agents are instructed that they are part of a structured reasoning system, must output valid JSON only, cannot provide final answers and must make uncertainty explicit.
- **Orchestrator prompt:** normalize the user input and output only the `proposition` object.
- **Pro Advocate prompt:** populate the `pro` section by constructing the strongest case for the proposition. Steel-man reasoning; no hedging; explicit assumptions and uncertainties. Do not reference or rebut the con side.
- **Con Advocate prompt:** populate the `con` section under mirror constraints.
- **Moderator prompt:** populate the `moderator` section. Identify agreements, disagreements, assumption conflicts, evidence gaps and decision hinges. Do not pick a winner or recommend action.
- **Challenge prompt:** given a user challenge (e.g., question an assumption), analyze only the targeted element, identify dependencies, provide historical context and classify the nature of the challenge. Output only the `challenges.responses` element.

Enforcement Strategy

- Use `response_format: json_schema` or equivalent to ensure valid JSON.
- Reject outputs with missing keys, extra fields or non-JSON text.
- Log schema violations for debugging and quality control.
- Version and test prompts to prevent drift and ensure reasoning quality over time.

MVP User Experience (UX)

The MVP is intentionally minimal: a single screen with no onboarding or accounts. Users input a question, optional context and generate the pro/con debate. The output appears below on the same page.

Screen Layout

- **Header:** displays the ClearSide name and tagline “Think both sides. Decide with clarity.”
- **Question Input:** a large text area where the user types their question or claim (e.g., “Should the U.S. impose a temporary moratorium on new AI data centers?”).
- **Optional Context:** a collapsible text area for specifying geography, timeframe or domain (e.g., “United States, 2025–2030, public policy & AI governance”).
- **Generate Button:** labeled “Generate Pro & Con Debate.” When clicked, it triggers the debate generation and shows a loading state (“Analyzing both sides…”).
- **Structured Reasoning Output:** appears below a divider once ready. It includes three columns: **FOR** (green), **AGAINST** (red) and **Moderator Synthesis** (gray) with clear headings and bullet lists. The synthesis section highlights areas of agreement, core disagreements and decision hinges.
- **Challenge Actions:** a small panel with buttons like “Question an assumption,” “Ask for stronger counterargument,” and “What evidence would change this?” Clicking one reveals an inline response from the Challenge Agent.

Design Goals

- No scrolling until output appears, so the user sees input and output together.
- Neutral, serious tone — more like a policy tool than a chatbot.
- Clean separation of pro vs con vs moderator sections.
- Clear call-to-action for challenging assumptions.

Explicit Non-Goals

- No user accounts, personalization or collaboration.
- No chat bubbles or conversational UI.
- No gamified or social features.
- The focus is on clarity and trust, not engagement loops.

Flagship Demo Debate

The flagship example below demonstrates the ClearSide output for the proposition **“Should the United States impose a temporary moratorium on new large-scale AI data centers to allow democratic institutions to catch up?”** It serves as the quality benchmark and regression test for agent behavior. Future improvements should only modify this example if reasoning quality improves.

Pro Advocate (FOR the Moratorium)

Executive Summary

- AI infrastructure is scaling faster than democratic governance.

- Data centers lock in technological trajectories before public consent is established.
- Temporary pauses are historically valid tools to prevent irreversible harm.
- The societal costs of unchecked AI scaling may outweigh short-term economic gains.
- A moratorium buys time for regulation, standards and public understanding.

Core Arguments

1. **Democratic & Governance Concerns**
2. AI capabilities advance faster than laws, courts and regulators can adapt.
3. Infrastructure build-out locks in technological trajectories before society consents.

4. Decisions affecting national security, labor markets and civil liberties are being made by private actors first.
   Assumptions: democratic legitimacy requires informed public participation; infrastructure scale constrains future policy options.

5. **Risk of Irreversible Harm**

6. Once built, data centers create sunk costs and political inertia, incentivizing continued use even if harms emerge.

7. AI systems may cause long-term societal effects (labor displacement, surveillance normalization) that are difficult to reverse.
   Assumptions: some AI harms are non-linear and visible only after deployment; precaution is justified under high uncertainty.

8. **Concentration of Power**

9. Large AI data centers disproportionately benefit a small number of tech firms and capital-rich regions.

10. This risks oligopolistic control, regulatory capture and reduced innovation diversity.
    Assumptions: infrastructure scale correlates with market dominance; market forces alone will not ensure equitable outcomes.

11. **Environmental & Resource Strain**

12. AI data centers demand massive electricity and water resources. Communities often lack meaningful input before construction.
13. Environmental impacts are externalized to local populations.
    Assumptions: current energy grids are already stressed; environmental review processes lag technological pace.

Known Weak Points (FOR)

- Economic opportunity costs are real.
- Other countries may not pause, creating strategic disadvantage.
- Enforcement boundaries could be difficult to define.

Con Advocate (AGAINST the Moratorium)

Executive Summary

- A moratorium would slow innovation and weaken U.S. global competitiveness.
- Infrastructure pauses do not stop AI development — they shift it elsewhere.
- Economic, security and scientific costs may outweigh speculative harms.
- Regulation can evolve alongside deployment.
- History shows that pauses often entrench incumbents rather than protect the public.

Core Arguments

1. **Global Competitiveness & National Security**
2. AI is a strategic technology; compute capacity correlates with geopolitical influence.
3. Slowing domestic infrastructure benefits foreign competitors and reduces U.S. leverage in standards-setting.
   Assumptions: AI leadership correlates with national power; other nations will not adopt similar pauses.
4. **Innovation & Economic Growth**
5. Data centers enable startups, research institutions and cloud access for smaller players.
6. A moratorium may favor existing large firms with overseas capacity while reducing job creation and local tax revenue.
   Assumptions: infrastructure growth broadly benefits innovation ecosystems; economic spillovers are net-positive.
7. **Regulatory Adaptation Is Possible Without Pausing**
8. Many technologies (internet, aviation, biotech) evolved alongside regulation.
9. Governance does not require a freeze — it requires iteration.
10. Pauses risk becoming indefinite due to political incentives.
    Assumptions: institutions can adapt quickly enough; incremental regulation is effective.
11. **Slippery Precedent**
12. Pausing infrastructure sets a precedent for politicizing technological build-out and arbitrary freezes in other sectors.
13. Uncertainty discourages investment beyond AI.
    Assumptions: investors and builders respond strongly to regulatory uncertainty; long-term trust in U.S. markets matters.

Known Weak Points (AGAINST)

- Optimistic assumptions about regulatory speed.

- Underestimation of cumulative environmental impact.
- Assumes benefits distribute broadly rather than concentrating.

Moderator Synthesis (No Verdict)

Areas of Genuine Agreement

- AI is transformative and high-impact.
- Governance currently lags technological change.
- Infrastructure decisions have long-term consequences.
- Environmental and resource impacts are non-trivial.

Core Disagreements & Root Causes

Disagreement	Root Cause
Pause vs proceed	Risk tolerance vs competitiveness
Speed of regulation	Confidence in institutional adaptability
Global competition	Zero-sum vs cooperative worldview
Infrastructure lock-in	Reversibility assumptions

Assumption Conflicts

- *Precaution vs momentum:* Is slowing down safer than falling behind?
- *Adaptability:* Can democratic systems realistically keep pace?
- *Irreversibility:* Are AI harms easier to undo than infrastructure commitments?

Evidence Gaps

- Empirical data on long-term societal harm from AI at scale.
- Comparative outcomes of historical technology moratoria.
- Real-world measurements of AI data center environmental impact.
- Regulatory response timelines vs infrastructure deployment timelines.

Decision Hinges

The debate would materially shift if we knew:

1. Can AI governance demonstrably keep pace without pauses?
2. Are AI-driven harms reversible after large-scale deployment?
3. Would allied nations coordinate on similar limits?
4. How quickly can energy infrastructure sustainably scale?

This flagship debate serves as the canonical test case for ClearSide's reasoning quality.

Repository Structure & File Organization

When implemented as a project, ClearSide can be organized as follows:

```
clearside/
├── README.md                    # Overview, mission, docs index
├── docs/
│   ├── 01_product-vision.md     # Product vision and problem definition
│   ├── 02_flagship-demo.md      # Canonical demo debate and quality benchmark
│   ├── 03_agent-architecture.md # Detailed agent design and flow
│   ├── 04_json-schema.md        # Full JSON schema specification
│   ├── 05_prompt-contracts.md   # Prompt → schema mapping and enforcement
rules
│   ├── 06_mvp-ux.md             # MVP UI specification and wireframe notes
│   └── 07_iteration-log.md      # Version history and change rationale
└── .gitignore
```

The README acts as the entry point, explaining what ClearSide is (and isn't), and pointing to each document. The docs directory holds the canonical specifications and can evolve through versioned updates.

Implementation Considerations

- **Model orchestration:** ClearSide can be built with a multi-agent system orchestrating separate LLM calls (or distinct roles within a single LLM) using the JSON schema as contract. Each agent must operate independently, with the orchestrator controlling information flow and validating outputs.

- **Stateless vs persistent:** The MVP can be stateless; each session does not require persistence. Future versions might store debates and user preferences.

- **UI & frontend:** A simple web interface (React/Vue or static HTML with serverless functions) can render the input form and the structured output. It must enforce neutrality through design and clearly separate pro, con and moderator sections.

- **Validation & testing:** Automated tests should validate that agents adhere to the schema, produce diverse arguments, list assumptions and preserve uncertainty. Regression tests should compare new model outputs to the flagship demo to ensure reasoning quality does not degrade.

- **Scalability:** ClearSide is model-agnostic; multiple models or prompt variations can be tested behind the same interface. The contract ensures interchangeability.

Iteration & Governance

ClearSide's specifications are intended to evolve carefully. The **iteration log** records every significant change, with version numbers and rationale. Rules for changes include:

- Never overwrite intent silently; always record **why** a change was made.
- Preserve prior versions for reference; do not delete or overwrite historical reasoning.
- Increment schema version when adding or removing fields.
- Flag breaking changes and update all downstream dependencies (UI, API clients).
- Use the flagship demo as a benchmark; modifications should improve reasoning quality rather than simplify content.

Iteration Log (v0.1)

- Defined product vision.
- Built flagship demo debate.
- Locked agent roles and responsibilities.
- Defined initial JSON schema.
- Established prompt contracts and enforcement strategy.
- Specified single-screen MVP UX.
- Created repository structure and documentation outline.

Current Status & Next Steps

ClearSide is fully specified at the conceptual level but not yet implemented. All future work must preserve neutrality, explicit assumptions, preserved uncertainty and structured adversarial reasoning. The next steps could include:

- Building a proof-of-concept implementation of the orchestrator and agents using the specified prompts and schema.
- Developing the single-screen MVP UI based on the wireframe.
- Creating automated tests for schema compliance and reasoning quality.
- Planning for persistent storage and user accounts in later phases.
- Adding optional features such as citations, value weightings and persona debates.

ClearSide is no longer just an idea; it is a designed system ready for development and iteration.
