RFI Response: AI Tools to Streamline Public-Sector Procedures

Recoding America Fund

POLICY ENGINE

Max Ghenis, CEO
max@policyengine.org
policyengine.org

1. About PolicyEngine

PolicyEngine builds open-source infrastructure for "computable policy"—translating tax and benefit statutes and regulations into executable code. Today, our rules engine calculates how laws affect households: eligibility, benefits, taxes, and marginal rates across all 50 states. API customers use this to help applicants navigate benefits and identify over $1 billion in unclaimed support.

The same infrastructure that powers these calculations can be extended to diagnose procedural burden. Because we compile legal text into executable logic with dependency tracking and cross-state coverage, we have a foundation for identifying **conflicting, duplicative, and burdensome provisions**.

Our track record:

- **3,000+ rules, 9,000+ parameters** across federal and state tax-benefit systems, covering statutes, regulations, and agency guidance
- **1,800+ legal citations** linking every calculation to authoritative sources (U.S. Code, CFR, state statutes, agency manuals)
- **All 50 states** covered for income taxes, SNAP, Medicaid, EITC, and dozens of other programs
- **Policy analysis users:** Joint Economic Committee, Niskanen Center, UK Cabinet Office/HM Treasury, New York State Legislature
- **NSF POSE Phase I awardee** for open-source ecosystem development
- **Validated against:** NBER TAXSIM (MOU with NBER; Dan Feenberg serves as advisor) and the Atlanta Fed Policy Rules Database (MOU with Atlanta Fed)
- **Public-facing:** Our tools are already available to the general public at policyengine.org, not just government staff

Sample outputs:

Live policy calculator · Policy research and analysis · Open-source codebase with every rule traceable to legal citations

2. Our Perspective on RAF's Concepts

The Core Insight: Law and Regulation are Code

The RFI seeks tools that "scan a state's statutory codes for sources of burden." Our experience suggests that **text scanning alone is insufficient**. Statutes and regulations are inherently computational—they define inputs, conditions, and outputs. To truly identify burden, you must **compile the legal text into executable logic** and analyze that logic mathematically.

When a regulation says "if income exceeds 130% of the federal poverty level, benefits shall be reduced by 30 cents for each dollar above that threshold," it is describing an algorithm. By making this algorithm explicit—compiling both statutes and regulations into code—we gain analytical capabilities that text scanning cannot provide.

What our infrastructure provides today:

- **Executable rules with dependency graphs:** Every calculation traces through a network of variables and parameters, making it possible to identify circular references, redundant inputs, and logical conflicts
- **Cross-state coverage:** We encode the same programs (SNAP, Medicaid, EITC, etc.) across all 50 states, enabling direct comparison of how different states implement the same federal requirements
- **Citation traceability:** Every rule links to its authoritative source, so we can trace from a calculated outcome back to specific statutory or regulatory text
- **Real-world friction data:** Our API customers—MyFriendBen, Amplifi, Student Basic Needs Coalition, Mirza, and Starlight—surface where applicants actually get stuck, which we can trace back to specific provisions

> **What this enables for RAF's goals:** By analyzing our dependency graphs and cross-state data, we can identify where regulations add steps not required by statute, where states impose more burdensome procedures than peer states for the same program, and where multiple provisions require the same underlying information.

How We Address Each RAF Concept

Concepts 1 & 2: Diagnostic Tools for Statutes, Regulations, and Guidance — We currently use AI (Claude and GPT-5) to accelerate encoding of new rules, with human experts validating the output. For RAF's diagnostic use case, we would extend this pipeline to systematically ingest a state's statutory and administrative codes, building a comprehensive dependency graph that can surface conflicts, duplications, and gaps between regulations and their authorizing statutes.

Concept 3: Rewriting Tools — Our platform generates human-readable explanations of policy rules for applicants (visible at policyengine.org). This same capability can be extended to rewrite regulatory text in plain language while preserving legal requirements.

Concept 4: Models Trained on Procedural Burden — Our cross-state coverage already enables comparison: we can show how State X's SNAP eligibility process differs from other states. Our API customers—MyFriendBen ($800M+ in unclaimed benefits identified in Colorado), Amplifi ($185M in California), Student Basic Needs Coalition, Mirza, and Starlight—provide real-world data on where applicants encounter friction, which we can trace back to specific provisions.

Concept 5: Support Tools for Regulatory Cleanup — Cross-state comparison is core to what we do: "How does State X's SNAP eligibility compare to other states?" is a question we answer today. Extending this to draft model legislation or regulatory amendments is a natural next step, using AI to generate text that achieves the same policy goal with fewer procedural steps.

3. Technical Architecture

Our core stack consists of policyengine-core, derived from OpenFisca—the rules-as-code framework created and used by the French government. We use Claude and GPT-5 for statute/regulation parsing with human-in-the-loop validation.

Challenge	Our Approach
Context window limits for long documents	Chunking with citation-aware boundaries; retrieval-augmented generation
Verifying correct parsing of statutory/regulatory text	Test-driven validation with known outcomes; human expert review; automated comparison against TAXSIM and Atlanta Fed PRD
Demonstrating success to human reviewers	Every output includes source citations; diff views showing before/after; quantitative complexity metrics (decision branches, input counts)

Data needs from states: Access to statutory and administrative codes (typically public), agency guidance documents and policy manuals, and historical data on application processing (for validation).

4. Partnership Model and Open Source Commitment

PolicyEngine is fully aligned with RAF's vision:

- **Open source:** All tools released under AGPL or MIT licenses. Any state can fork, adapt, and deploy at no cost. See our GitHub.
- **Low-cost scaling:** Tools developed for one state are immediately available to others.
- **Public availability:** policyengine.org is already public-facing, not restricted to government staff.
- **Co-development:** We view states as partners, not customers. We need state input to validate outputs against administrative reality.

Path to impact: While PolicyEngine focuses on the statutory/regulatory layer, our API customers operate at the applicant interface. MyFriendBen, Amplifi, and others deploy our rules engine for benefits screening—together identifying over $1 billion in unclaimed benefits. This creates a feedback loop: they surface friction from real applicant behavior, which we trace to specific provisions, enabling targeted regulatory reform.

5. Indicative Cost and Timeline

Single State Deployment

Option	Scope	Timeline	Cost
Focused Pilot	3-5 priority domains (e.g., Medicaid eligibility, SNAP, housing)	12 months	$350,000
Comprehensive	All major tax and benefit programs	18 months	$600,000-750,000

Scaling to Additional States

Because ~60-80% of policy logic is federal or follows common patterns, subsequent states cost significantly less:

Scenario	Cost	Timeline
State 2	$150,000-200,000	6-8 months
State 3+	$100,000-150,000	4-6 months
At scale (10+ states)	$75,000-100,000	3-4 months

6. Why PolicyEngine

We've already done this. This isn't a research project—it's an extension of production infrastructure that governments already use. We are **cross-state by default**, enabling immediate benchmarking. We have **continuous validation** against TAXSIM and Atlanta Fed. Our AI is **bidirectional**—generating code from legal text and plain-language explanations from code. We have a **government track record** (Joint Economic Committee, UK Cabinet Office) and understand the compliance-to-outcomes shift that RAF seeks.

Max Ghenis, CEO
PolicyEngine (a fiscally sponsored project of the PSL Foundation)
max@policyengine.org · policyengine.org