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 statutes, regulations, and administrative guidance into executable code that can be analyzed for burden, inconsistency, and complexity. Our platform already does what the RFI describes: we scan legal text, compile it into executable logic, and surface **outdated, conflicting, and duplicative provisions**—across both statutory and regulatory layers.

Our track record:

- **9,000+ encoded rules** 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, TANF, EITC, and dozens of other programs
- **Government users:** Joint Economic Committee, 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
- **Public-facing:** Our tools are already available to the general public at policyengine.org, not just government staff

Sample outputs:

Live policy calculator · Cross-state comparison tools · 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:

- **Conflicting and contradictory provisions** where regulations contradict their parent statutes, or where multiple regulations define the same term differently

- **Gaps between regulations and statute** where agency interpretations add requirements not present in law, or fail to implement statutory mandates
- **Outdated provisions** such as dollar thresholds not indexed for inflation since 1990
- **Duplicative requirements** where multiple regulations collect the same information
- **Excessively burdensome interpretations** where regulations impose unusually high fees or unnecessary steps compared to other states
- **Procedural burdens not obvious in text** surfaced through our partner ecosystem—MyFriendBen and others deploy our rules engine in applicant-facing tools, creating a feedback loop that traces real-world friction back to specific regulatory provisions

> This transforms burden identification from a subjective reading exercise into a **quantitative diagnostic**. We measure complexity in decision branches and input requirements, enabling objective cross-state comparison.

How We Address Each RAF Concept

Concepts 1 & 2: Diagnostic Tools for Statutes, Regulations, and Guidance — Our AI pipeline ingests statutory codes, administrative codes, and agency guidance documents. We use Claude and GPT-5 to parse legal text and generate executable Python code, with human experts validating edge cases. Once compiled, our engine automatically flags: outdated provisions, conflicting rules, duplicative requirements, reporting/meeting mandates, wet signature requirements, and—critically—**gaps between regulations and their authorizing statutes**.

Concept 3: Rewriting Tools — Our platform already generates human-readable explanations from code. We can extend this to rewrite regulations in plain language while preserving legal requirements, flag passages where original text is vague or ambiguous, and generate applicant-facing guidance that accurately reflects the rules.

Concept 4: Models Trained on Procedural Burden — Because we encode rules across all 50 states, we can flag regulations that interpret statutes in unusually costly ways. If State X requires three in-person visits for a license that other states issue online, our cross-state analysis surfaces this. Our partner applications (MyFriendBen, Amplifi, Student Basic Needs Coalition) surface procedural burdens identifiable in practice—the approval matrices, unwritten practices, and friction points that don't appear in legal text but shape how rules are actually applied.

Concept 5: Support Tools for Regulatory Cleanup — We already enable cross-state analysis: "How does your SNAP asset test compare to the 10 least-burdensome states?" is a question we answer today. When we identify a burdensome provision, we can draft model statutory language to streamline it, alternative regulatory text achieving the same goal with fewer steps, and comprehensive reform packages addressing multiple related burdens.

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, we partner with organizations at the applicant interface. MyFriendBen deploys our rules engine for benefits screening. 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, TANF, occupational licensing)	12 months	$350,000
Comprehensive	All major benefit programs + licensing/permitting regimes	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 Tuscaly sponsored project of the PSL Foundation)
max@policyengine.org · policyengine.org