RFI Response: AI Tools to Streamline Public-Sector Procedures

Recoding America Fund

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

About PolicyEngine

PolicyEngine builds open-source infrastructure for translating tax and benefit statutes and regulations into executable code. Our rules engine calculates how laws affect households—eligibility, benefits, taxes, and marginal rates—across all 50 states. API customers including MyFriendBen, Amplifi, and the Student Basic Needs Coalition use this to help applicants navigate benefits, together identifying over $1 billion in unclaimed support.

What we have built:

- **3,000+ rules, 9,000+ parameters** encoding federal and state tax-benefit programs (SNAP, Medicaid, EITC, income taxes, and dozens more)
- **All 50 states** with the same programs encoded consistently, enabling direct cross-state comparison
- **1,800+ legal citations** linking every calculation to U.S. Code, CFR, state statutes, and agency manuals
- **Dependency graphs** showing how each rule relates to others
- **policyengine-core**, derived from OpenFisca—the rules-as-code framework created and used by the French government

Validation and users:

- Validated against NBER TAXSIM (MOU; Dan Feenberg advises) and Atlanta Fed PRD (MOU)
- Used by Joint Economic Committee, Niskanen Center, UK Cabinet Office/HM Treasury, NY State Legislature
- NSF POSE Phase I awardee
- Public-facing at policyengine.org

Our Perspective on RAF's Concepts

The Core Insight

The RFI seeks tools that "scan a state's statutory codes for sources of burden." Text scanning can find keywords, but statutes and regulations are inherently computational—they define inputs, conditions, and outputs. A regulation that says "if income exceeds 130% of the federal poverty level, benefits shall be reduced by 30 cents for each dollar above that threshold" is describing an algorithm.

By compiling legal text into executable logic, you can analyze it mathematically: identify circular dependencies, find where two provisions require conflicting inputs, detect where regulations add requirements not present in authorizing statutes, and compare how different states implement the same federal program.

> **This is what we do for tax-benefit programs.** We haven't yet applied it to RAF's diagnostic use case—scanning for procedural burden like wet signature requirements or excessive approval steps—but our infrastructure provides a foundation.

How Our Infrastructure Applies to Each Concept

Concepts 1 & 2 (Diagnostic tools for statutes, regulations, and guidance): Our dependency graphs already trace how each rule connects to others. Extending this to flag conflicts, duplications, and gaps between regulations and authorizing statutes is architecturally feasible. We would need to expand our encoding scope beyond benefit calculations to include procedural requirements (signature requirements, reporting mandates, approval processes).

Concept 3 (Rewriting tools): Our platform generates plain-language explanations of how policies affect applicants. We have not built tools to rewrite regulatory text itself, but the underlying capability—translating between legal language and structured logic—is similar.

Concept 4 (Models trained on procedural burden): Our API customers (MyFriendBen, Amplifi, etc.) encounter real friction when helping applicants. We don't currently aggregate this systematically, but their experience could inform what provisions cause actual burden versus theoretical burden.

Concept 5 (Cross-state comparison and regulatory cleanup): **This is our core strength.** We encode the same programs across all 50 states. Today we can answer: "How does State X's SNAP eligibility compare to State Y's?" Extending this to generate model legislation or reform packages that simplify rules while preserving policy intent is a natural next step.

What We Would Deliver

For a pilot state, we would:

1. **Expand encoding scope** to include procedural requirements (approval processes, signature requirements, reporting mandates) beyond benefit calculations
2. **Build diagnostic queries** on our dependency graphs to surface conflicts, redundancies, and regulation-statute gaps
3. **Generate cross-state comparison reports** showing where the pilot state's procedures exceed peer states
4. **Trace findings to citations** so reviewers can verify each flagged provision against source text

Indicative Cost and Timeline

These estimates assume we're extending our existing 50-state infrastructure to RAF's diagnostic use case—work we haven't done before, but which builds on proven architecture.

Scope	Timeline	Cost
Pilot (2-3 programs, e.g., SNAP + Medicaid eligibility procedures)	6-9 months	$200,000-300,000
Expanded (5-7 programs including housing, childcare, workforce)	12-15 months	$400,000-500,000

Subsequent states cost less because federal rules and common patterns transfer:

Scenario	Cost	Timeline
State 2	$100,000-150,000	4-6 months
States 3+	$75,000-100,000	3-4 months

Partnership Model

PolicyEngine is fully aligned with RAF's vision:

- **Open source:** All tools under AGPL/MIT. Any state can fork and deploy. See github.com/PolicyEngine.
- **Low-cost scaling:** Work for one state immediately benefits others.
- **Public availability:** policyengine.org is already public-facing.
- **Co-development:** We need state input to validate what our tools surface against administrative reality.

Data needs from states: Statutory and administrative codes (typically public), agency guidance documents, and feedback on which flagged provisions are real burdens vs. necessary requirements.

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