Exploring a new way to make eligibility rules easier to implement

By Ed Mullen

Published on October 16, 2018

public benefits

Here at 18F, we've worked with many talented and dedicated civil servants who deliver critical services to residents across the country through programs including but not limited to Medicaid, SNAP, National School Lunch Program, and Social Security Disability Insurance/Supplemental Security Income, among others.

These programs are managed at the federal level, but administered at the state agency level, and often executed at the local level. Residents have to be deemed eligible for these services in order to receive benefits. These programs change and adapt over time. The challenge is that when federal agencies issue a policy change, say income eligibility guidelines, that policy gets communicated down to the states as text on the Federal Register or via PDF, and it's up to each state to:

1. Operationally define what that change means for their state's program operations
2. Make a change to the underlying IT system that powers that program to implement that change in policy and operations

This translation of federal policy into many state systems, with many technology partners in between, creates opportunities for implementation errors and represents significant duplicated effort, since most states (and territories, and sometimes each county and tribal organization) manage their own separate IT system(s) to run these programs.

When this happens...

Given all this duplicative effort and the corresponding risk, we got to thinking about opportunities for improvement.

Exploring a different approach

In learning more about this problem, we thought to ourselves...

What if a federal agency coded their program's eligibility criteria into a single, central *web service* that states could use to help determine eligibility?

When someone applies to a program, their data is evaluated against eligibility criteria by a rules engine the state operates. Most states operate their own.

A web service is an application or data source that's accessible over the internet. Web services communicate with other computer systems, not directly with users. They function behind-the-scenes to allow separate systems to work together.

In this world, an eligibility rules service could receive an applicant's anonymized data, weigh it against the rules, and send an immediate response to the requestor about whether the applicant meets the criteria. States wouldn't need to build out the logic over and over. They would just gather the applicant data, send it to the central eligibility rules service, and use the responses to *help* make eligibility determinations. All the data could be anonymous, and none of it would need to be stored by the rules service.

(One important note: we don't intend for this sort of eligibility rules service to *make determinations*. That is done at the program level. This service could help conduct the deterministic evaluation while program staff would be left to make the determinations themselves.)

In this world, states would simply need to help configure their state-specific scenarios and connect to the service, rather than building and managing an entire, separate eligibility rules engine themselves. Community-based organizations and service providers could use this same web service to help their clients make decisions about applying. Oversight bodies and the public could view the open source code repository and audit the criteria, knowing exactly what the system does. Policy makers could use the logic to test program changes and make evidence-based decisions.

What if each eligibility-based federal program took this same approach, and **there was an entire open ecosystem with trusted, open, official, auditable, executable eligibility policy,** being used by states, non-profits, and other public organizations to help serve the public?

Testing this idea

At 18F, we like to break big problems into smaller ones and then work to validate or invalidate our assumptions. We identify the area we need to learn the most about and start investigating.

In the last few months, we explored the technical feasibility of building out a single eligibility rules service. And through this work, **we've learned that our "what ifs" are not so far out of reach.** We found that building this sort of centralized eligibility rules service *is* feasible. The logic can accommodate real-world policy variations between states to deliver state-specific criteria. It can be built entirely with free, open source software, and can avoid the overhead of adopting complex rules engines. We worked through a number of technical questions that allowed us to establish a solid starting point with less risk for future efforts. Also, we found that the staff responsible for these systems are interested in exploring ways to mitigate risk.

How do we know?

We've explored this concept from a technical angle, and it's not just possible, it's pretty darn simple.

To understand the challenges of interpreting complex, varied eligibility policy and turning it into an API-based web service that multiple parties could access, we built a prototype eligibility rules service for a sample federal program. This allowed us to test our hypothesis that we could accommodate this kind of variability in the rules service. Turns out we could.

We started by combing through a sample federal program's regulations and policies as well as the state-level policy, and translated those policies and regulations into clear statements that could form the basis of the rules at the core of our web service. We checked our interpretations with policy experts at the federal and state level along the way. Then we began building our web service. Development took a bit more than a month.

Our work is visible in this GitHub repository. The web service can be tested directly via the API. Since a web service is a computer-to-computer system, and normally wouldn't be visible to a casual user, we built a test form that can be used to send data to the web service.

We've evaluated this concept with federal and state partners and it seems to resonate.

Our conversations with people who run or oversee eligibility systems at the state and federal levels found the proposition of a centralized, web-based rules service compelling.

The lessons are broadly applicable. If you set policy for a program that individuals apply to, and that policy is implemented by different government entities, be they states or counties or some other body, this applies to you.

Digging deeper

Of course, there are still unknowns. We need to figure out what the integration with various state systems would look like, where the division of responsibilities between state systems and the rules service should be, what sort of eligibility responses state programs would need, and a number of other questions. But the impact could be huge.

Over the next several months, we'll be collaborating with a new federal partner and one or two states to get clarity on some of the remaining unknowns. We'll track this work through our GitHub repository as we have so far.

If you're interested in learning more, implementing a similar effort, or even if you just generally have feedback or questions for us about this idea, please reach out to us at eligibility-apis-initiative@gsa.gov! We need your expertise and help to make this the most effective resource that it can be.

A number of people have contributed ideas to this subject, including Alex Pandel, Shawnique Muller, Catherine Devlin, and Tony Garvan.

Icons by Nick Kinling, Adrien Coquet, Iconic, Jayson Lane, anbileru adaleru, Eucalyp, Wireform, Arthur Slain, Chameleon Design, Nook Fullopticon, Storm Icons, and Rflor from *Noun Project*.

Follow 18F
18F on GitHub
18F on Twitter
18F on LinkedIn
RSS feed

‹ Previous post
Implementing rules without a rules engine

Next post ›
Two exercises for improving design research through reflective practice

Modular contracting and working in the open

Working in the open is a key component of building trust between governments and vendor partners. Read about how the State of Alaska is using openness and code sharing to foster greater trust between government project teams and vendor teams as part of a large legacy system overhaul.

Implementing rules without a rules engine

If you're building a rules-based system, don't assume that you need a separate business *rules engine* product. Rules can be implemented more easily and with less overhead by cross-functional teams working to describe the rules and policy directly in code using a general purpose programming language like Python, Ruby, etc.

Catching up with the TANF Data Portal project

Around 800,000 low-income American families receive cash assistance through Temporary Assistance for Needy Families (TANF) each month. 18F and the Administration for Children & Families' Office of Family Assistance partnered on building a new data portal for TANF. We caught up with Office of Family Assistance leaders to see how their agency is continuing with the work.