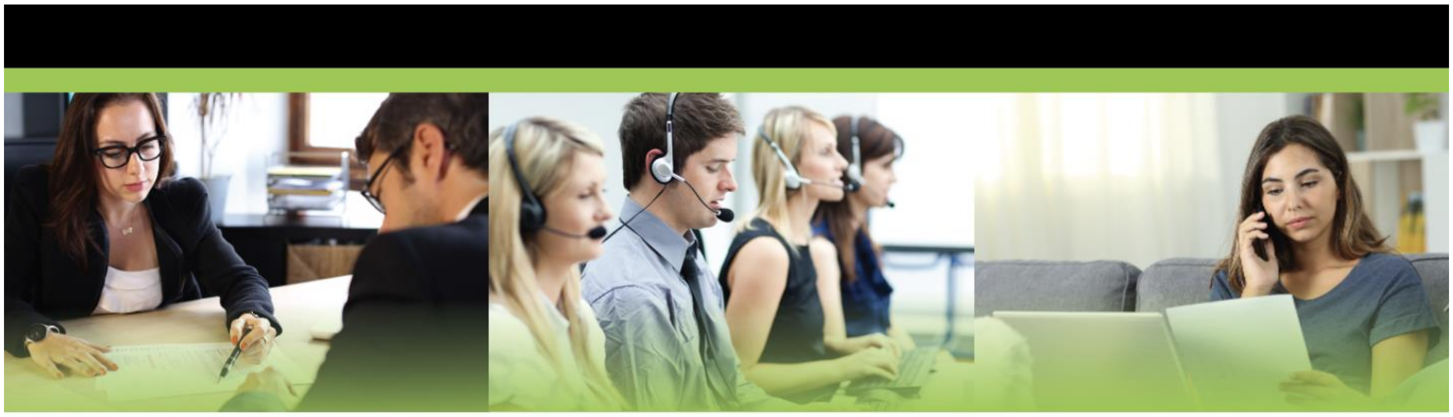

BUILDING & HOSTING SOLUTIONS FOR GOVERNMENT AGENCIES

Open Source and Software Reuse in the US Unemployment Insurance System

WHITEPAPER

April 30, 2024

Gregg Tahmisian

Contents

After providing a brief background on open source software as well as the national Unemployment Insurance (UI) system, this paper takes up the following questions:

1. What is the state of open-source, reusable software within the UI system?
2. What can be done to increase the amount of open-source and software reuse in UI?

Open Source Software

The Open Source Initiative is "a standards body, maintaining the Open Source Definition for the good of the community." This paper understands "open source" in accordance with this commonly-accepted definition - though the following shorthand provides a general understanding: "software for which the original source code is made freely available and may be redistributed and modified."

Importantly, "open source" involves more than just access to the source code, and more than just "free" software. The vision of open source is to build a community of contributors around a project. The more robust the community of contributors, the more robust the software will be. "Open source" is software made by developers for developers.

The goal of open source is to encourage programmers and engineers to create and develop technologies through collaboration. For instance, a programmer in San Jose develops a new application, then another programmer in Atlanta studies the application and discovers ways to improve it. The knowledge is shared, and the entire community benefits from the collective innovation.

While many open source applications are free, developers are entitled to profit from their work. However, instead of selling open source products directly, businesses tend to build services on top of an open source foundation. A stronger base improves all businesses that depend on the software.

Why is open-source and software reuse important?

It is often said that open source is "freedom." This is often distinguished by the phrase "Free as is 'free speech,' not as in 'free beer!' " Reducing the barriers to entry by tearing down proprietary "walls" around the garden has the effect of opening up the space for others to join the effort – i.e., competition. With increased competition comes lower prices, better features, and better service for the state customers and their end users. These are the ultimate goals of promoting open source within the government community.

To these ends, creating and sustaining a vibrant open source community is ONE piece of the puzzle. This paper does not view "proprietary software as a social evil" as some might. Rather it views open source as a viable and worthy alternative to proprietary software – an alternative that in certain cases will have compelling advantages to state agencies.

"There is always a concern and worry about how to strike a balance between proprietary and open-source," says DB Hurley, the founder of the Mautic community. If the delicate balance between free, skilled labor and profitable intellectual property is not respected, then "open-source will lose significant benefits as it slowly becomes nothing more than a watered-down 'trial' version of proprietary software." However, if done correctly, a vibrant open source community, operating symbiotically with for-profit corporate drivers can offer state government and its citizen-consumers the best of both worlds: constant innovation driven by passion as well as persistence and reliability of contractual obligations.

Because of benefits promised by open source, and despite the costs and hassles, we believe that creating and maintaining a community of open source products and practitioners is a worthwhile endeavor that should be pursued with greater zeal than it currently is.

The National Unemployment Insurance System

The national UI system is called a "federal-state partnership" insofar as the existence of - as well as the general scope of - the Unemployment Insurance program is mandated by federal law. However, each state administers its own UI program as it sees fit - with its own state employees under its unique state law. This state administration of the program is funded by the federal government through periodic grants to the states; the amounts and timing of which are determined by the federal government.

In practice, the federal part of the partnership has traditionally adopted the role of the "sponsor" – they give funding to the states as well as general directions as to what they want that money to achieve. The feds provide the ends, the "what," and the goals. They generally do not provide the means, the "how," or the methods; those are usually delegated to the individual states to figure out for themselves.

As a government program, the UI system is important because it is seen as a safety net, both for the individual as well as, on occasion, for society. It is those "occasions" where the UI system is reluctantly pressed into the role of societal safety net that the system comes under the most pressure – as well as the most scrutiny. Recent examples include the Great Recession and the COVID-19 pandemic, where the UI system was called upon not just to replace income for individual displaced workers, but to do it on such a scale so as effectively to keep the national economy afloat. Many feared that should the UI system fail to do this, the result would have been national economic collapse and perhaps societal upheaval. The stakes were higher than anyone had anticipated.

It turns out that the UI system performed "just well enough" overall to maintain economic liquidity and avert panic. But this came at a huge cost. Because of the unpreparedness of the UI system, Agencies could not maintain what had become "normal operations" prior to the pandemic. "Normal operations" included safeguards such as identity verifications, fraud checks, and human scrutiny and issue adjudication.

From our perspective, "on the ground" at the state-level, in the early days of the pandemic, the ONLY thing that mattered was getting claimants paid quickly, and many otherwise routine safeguards were a hinderance that staff felt had to be jettisoned. And so they were. Some checks would be discarded entirely and in other cases, Agency processes and procedures were changed to allow claimants to be paid despite the presence of "issues" that would normally have stopped payment. As we now know, rampant fraud ensued. The bottom line is that the collective unpreparedness of each state's UI system cost the American taxpayer – directly or indirectly – hundreds of billions of dollars. Then the pendulum swung in the other direction. Instead of public outrage over delayed payments, the public became outraged over the extent of fraudulent payments.

Public examination appears to have generally concluded that both problems – lack of timely payments as well as the incidence of fraud - were due to each state's reliance on technologically outdated computer systems to automate their UI program. Thus, as part of the 2021 American Rescue Plan Act

(ARPA), Congress allocated (what ultimately turned out to be) $1 billion to the Department of Labor (USDOL) with instructions to improve the states' UI computer systems in three specific areas:

- Increase the systems' detection and prevention of fraud
- Ensure the systems pay benefits in a timely manner
- Ensure the systems promote equitable access for disadvantaged stakeholders

Congress does not mention open source as an area of emphasis, but when instructing states on the use of the ARPA money, USDOL does advocate for a form of open sourcing. USDOL identifies "Software Reuse and Collaboration" as a "promising practice for states to consider," saying, "Given… the Department's vision for an UI technical ecosystem built around open and modular solutions, states are encouraged to consider using open-source software produced by other states through this grant opportunity, or through other means. Additionally, the Department encourages states to consult with each other on which open-source components they plan to build, or invest in, to maximize the potential for software reuse across states." This is a promising start.

The State of Open Source within the UI System

Open source has been implemented across industry, education, consumer products, and elsewhere to varying degrees of success. In general, advocates of open source software believe that it can provide higher quality, more secure, and more customizable software than proprietary alternatives. National leaders, technologists, and think tanks have discussed the benefits of open source software as a solution to at least some of the technological problems besetting the UI system. Within the context of UI, proponents see additional benefits to Agencies in the form of software reuse from one state to the next; low-to-no initial acquisition cost; as well as low costs for customization, maintenance, and updates.

The first question we must ask is, "Is there any open source software within the UI system?" The concise answer to this question is, "No." While USDOL does maintain a GitHub presence, there are no repositories with UI projects beyond a couple "pilots" or "prototypes," and nothing that represents an actual UI product that is in production or that is actively maintained. In fact, there is no software product within the UI universe that remotely resembles even the shorthand definition of open source, much less the more stringent one. This answer and the following discussion are not intended to be pejorative, judgmental, or critical of any parties. We simply offer an honest, unvarnished descriptive of the current situation as we see it.

As far as actual products go, the UI technology scene contains two distinct types of software:

- Software produced and owned by vendors
- Software produced by the state (using state staff and/or vendor(s) under a work-for-hire contract)

There is no vendor-produced software that we know of that claims to be open source.

The UI software (or "intellectual property developed under a discretionary Federal award process") produced by the state using federal funds (hereinafter called "Federal Software") is sometimes said to be open source – but is it?

According to 2 CFR 2900.13, USDOL requires Federal Software "to be in a format readily accessible and available for open licensing to the public." However, we have not seen any places where Federal

Software is described or cataloged, nor where there was a licensing mechanism, nor where there was a source code distribution mechanism. On the contrary, we only know of instances where "the public" has been denied all access to specifically requested Federal Software by states.

Therefore, we believe that while Federal Software is nominally "available for open licensing to the public," the reality is that the source code *could* be available to *certain* parties under *certain* circumstances. To be clear, this is **not** open source.

- There is no Federal Software for which the source code is "freely available."
- There is no software that has a well-publicized means of obtaining the source code.
- There is no software that can be sold or given away, without fee, as part of an aggregate software distribution.
- There is no software that allows redistribution of derived works.

All of these are necessary for truly open source software.

We believe that the people who refer to Federal Software as open source often simply mean that the source code will be given to another workforce agency once they jump through the proper hoops. But again, that's not open source. Such software will never develop a community of contributors around it. Thus, it will not grow and mature like a true open source offering. Again, "open source" is all about building a robust community of contributors.

So, we conclude that there is no software – whether produced by a vendor or owned by the government - in the UI space that fits the definition of open source. The next question is, "Why?" It will help if we refine this simple question to differentiate between vendor- or state-produced software:

- What prevents Federal Software from being open source?
- Why is there no incentive for vendors to produce open source software?

What prevents state-produced software from being open source?

We identify several factors which prevent existing Federal Software from being treated as truly open source. These include:

Fragmented ownership

Federal Software is scattered across states with no catalog of who owns what. Each state retains ownership of the Federal Software which they have developed. Despite the requirement that it be made available to the public, compliance is at the discretion of the state, and there does not appear to be any enforcement mechanism. USDOL has the right to "reproduce, publish, or otherwise use" the Federal Software and can authorize others to do so, but USDOL does not appear to exercise these rights.

No open source infrastructure

An open source project doesn't just magically happen. It must be managed with intentionality. It takes a group of people committed to establishing and maintaining the project. It takes planning and effort to set up a code repository as well as rules around contribution, licensing, and distribution. This infrastructure is required in order to establish an open source project.

To date, none of this work has been done with regard to any Federal Software. The lack of a governing body, of licensing and distribution rules, and of a code repository has made it impossible for Federal Software to function as open source.

Security fears

We have heard some states say that they would not release code out of fear that someone would use the code to find a way to exploit the system. This paper makes no judgement as to whether that fear is warranted or not at present. However, we do note that many technologists advocate for open source software precisely because they believe public code is ***more*** secure than secret code. The reasoning is that in open source, the community promptly finds and reports security flaws which are usually fixed right away; whereas if there is a security flaw in a proprietary software product, nobody will know until someone falls victim to it. Also, open source products cannot misuse and abuse users' data intentionally as can happen in a proprietary system. The community would quickly discover this abuse.

Why is there no incentive for vendors to produce open source software?

We believe the lack of vendor-produced open source software boils down to a lack of incentive. To understand this, it is helpful to examine what prompts developers – whether as individuals or as a company – to get involved with any project. They are motivated by: Profit, Promotion, or Passion.

- **Profit**: The developer or the company makes money from the project. Simple enough.
- **Promotion**: Being able to say that you or your company contributed important features or pull requests to a particular open source project may be as good as gold when it comes to applying for a new job or landing a new contract
- **Passion**: Developers are passionate about the project. Maybe it's the technology, the subject matter, helping the users, or having control. In any event, the combination gives developers a sense of passion and purpose they crave.

The problem is that none of these incentives currently exist in the UI space – and some may never.

- **Profit**: Most of the vendors in the UI space only see a pathway to profit through proprietary software that they can sell to multiple customers. They view their source code as their intellectual property and their trade secrets which must be jealously guarded and protected at all costs. They believe the only thing that differentiates them from their competitors – or would-be competitors – is their ability to encode complex UI processes into software. They cannot see how "giving all that away for free" could be profitable for them.
- **Promotion**: The UI community is very niche. There are only ever going to be 53 possible customers. Vendors presently find it difficult to see how anyone is going to care whether they contributed to a UI open source project.
- **Passion**: While there are passionate individuals working on UI, the problem is that most vendors believe that passion is a luxury they can afford only when the profit need is met. And since companies currently believe that open sourcing their products will kill their profit, there is little chance that any vendor will pursue open source as a passion project as things now stand.

So not only are there currently no incentives to open source in UI, but there are also some peculiarities which further discourage the creation and adoption of open source projects.

- Lack of standards: UI systems are generally monolithic. And the few that are modular are all generally "one off's" such that the interfaces between the modules in the system are hidden, propriety, system specific, or all three. The net effect of this is that UI systems are essentially "All or nothing," meaning that a vendor cannot just develop a single module that will work in multiple states; they must either develop the whole system or nothing. If a vendor could develop and open source a single module, they might be willing to do that, but developing a whole open source UI system is incredibly expensive, risky, and unpalatable.

  Even supposing a vendor could get access to one state's interfaces such that it could create and deploy a single module for that state, even in that case, without standards, that's the end of the line; it won't work in other states. All this effort is not very worthwhile if only one customer can use it.
- Procurement rules: State procurement is torturous and broken. And every state has their own nuances and laws. These rules are tailored for procurement of commodities such as, say, car tires which can be described by a common specification. The rules are not set up to handle software. If there is no money in it for a vendor, there will be no one to drive the procurement process.

How to Promote Open Source and Software Reuse

We believe that movement toward a strong open source culture in UI technology is desirable, but will not happen without intentional leadership at the national level. Specifically, we advocate the implementation of a two-pronged approach similar to the following:

1. Make existing Federal Software open source
2. Create incentives for vendors to contribute to open source projects

As noted above, USDOL would appear to be the proper entity to lead the charge for open source. First, under 2 CFR 200.315(d), USDOL can control access to all Federal Software, and so is best positioned to manage it. Second, USDOL manages the purse strings for the states. Third, USDOL maintains a national perspective which is necessary for success; states' perspectives are understandably myopic.

To be successful, USDOL would need to embrace its role as the open source hub and dedicate the necessary time, effort, and money to the project.

Make Existing Federal Software Open Source

The first prong consists of implementing the policies and infrastructure necessary to make the existing Federal Software to be functionally (not just statutorily) open source. Open source projects must be managed with purpose and intentionality.

As noted above, "open source" is software made by developers *for developers*. It is meant to be modified into derivatives or bundled with an aggregate software distribution for use by other programs. The ramification is that source control, licensing and distribution will be more complex than it first appears. The details of this complexity are beyond the scope of this paper, but they should not be underestimated.

Open sourcing existing Federal Software would consist of several steps:

Establish a Governing Body: The Governing Body would be tasked with setting up and maintaining a software catalog and code repository (see below). They would also draft the necessary policies and rules around such things as the mechanics of contributing to the project and how software licensing and distribution will be handled. The resulting documents would be validated by the necessary legal entities and uploaded to a document repository.

Catalog Federal Software: Identify all extant Federal Software ("intellectual property developed under a discretionary Federal award process"). Obtain copies of all the relevant source code, check-in history, artifacts, and documentation.

Establish a Code Repository: Select a repository technology and upload all the items obtained from the Catalog Federal Software effort. Implement the access control and contribution policies ratified by the Governing Body.

Create Incentives for Vendors

The second phase involves creating incentives for vendors to join the open source movement. This may involve vendors contributing to existing open source projects, or even creating new ones. Either way, it won't just happen; additional incentives must be provided in order to make this a reality. This paper does not have all the answers with regard to what these incentives may look like. While that subject may be the topic of a future paper, at present, we can provide some suggestions as well as direction for additional thought:

Establish Interface Standards: Interface standards allow for ***consistent modularization***. **Modularization** is important in order to avoid the all-or-nothing quagmire where a developer needs to either make an entire system or they really can't deploy anything. And a **consistent** modularization allows for reuse: a module that is made for one state is able to work in other states. Both of these effects work together to provide lower cost to states as well as vendor incentive through increased ROI on development efforts and lower barriers to entry for UI development. Interface standards should initially be drawn from the open source Federal Software catalog because it is comprehensive, it is accessible, and it positions the open source catalog as the standard to which other software must adhere in order to reap the benefits of reusability.

Financial Incentives: "Open source" is about building a community of contributors. The contributors have traditionally worked *pro bono*, but such altruism is not one of the Open Source tenets. One can envision an open source ecosystem where contributors are, in some way, compensated for their contributions. This would be somewhat uncharted territory in the open source world. Again, this paper does not present all the answers, but this is a direction in which we feel further thought is warranted.

Streamlined Procurement: We would recommend streamlined procurement to incentivize vendors to join the open source community. Again, this may be uncharted territory and this paper does not present all the answers, but one can envision a world where at least some of the vendor compensation was handled in a unified, simplified way at the federal level rather than having every state do what they think is best. This is another area to which to devote further thought.