Why Unemployment Insurance Systems Belong in the Cloud

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Contents

Introduction

3.3 Million File Unemployment Claims as Economy Comes Apart

- The New York Times, March 26, 2020

The numbers tell the story. In February 2020, about 210,000 Americans filed initial unemployment claims every week. One month later, as the COVID-19 pandemic took hold, that number skyrocketed to 6,648,000—an increase of more than 30 times in just one month. COVID-19 drove unemployment to levels unseen since the Great Depression and did more damage to the economy in three months than the 2008 financial crisis did in two years.

In this time of need, too many unemployment insurance (UI) systems failed the people they were designed to help and few, if any, UI programs came out of the pandemic unscathed. The good news is states can—and are—working to fix this. This paper offers our point of view on how states can take action now to prepare for the next economic crisis.

But first, it's important to understand what happened. Why did everything fall apart, and perhaps more interestingly, why did it all fall apart in the same way, state by state, system by system? In big states and small, rich states and poor, four things went wrong. They built on each other, creating a self-reinforcing cycle that kept people from getting the help they needed and kept agencies feeling helpless to respond.

First, systems crashed

Most UI systems, whether built in the 70s or the 2010s, simply were not built to scale—to handle a sharp increase in people filing claims. Built before cloud technology was widely used, they ran on legacy infrastructure purchased outright and maintained by state employees or vendors. But infrastructure is a risky buy. To not overpay, you need to purchase based on how much load you *think* you'll have, not how much load you'll *actually* have.

During the pandemic, these inelastic infrastructures were incapable of keeping up with the surges of traffic coming to their websites. There just wasn't enough compute power to go around. As a result, websites slowed or crashed, applications couldn't be submitted, and frustrated claimants had to turn to call centers for help.

Then, call centers got overwhelmed

With no functioning websites to rely on, claimants picked up the phone. But the scalability problem also applies to staffing. You pay for the amount of call agents you *think* you'll need, with no way of knowing what's coming. And scaling people is even harder than scaling tech. You can buy new hardware in a pinch, but new humans need recruiting, training, and support. The struggle is multiplied when the systems and business processes those people rely on to do their jobs are antiquated too—manual processes, clunky software, poor analytics.

During the pandemic, these call centers collapsed. Claimants were left on hold for hours, if they got through at all. In states that threw bodies at the problem, residents found themselves talking to someone who didn't understand their problems and didn't know how to help. Even when public-facing systems came back online, call center volume stayed high. And because most UI systems are designed for compliance, not experience, first-time filers had no way of knowing whether their claims were accepted, where they stood in the adjudication process, or when they might see their payments.

Then, UI became news

When people couldn't get through on the phone, they turned to the press and social media. Journalists and think tanks started to highlight UI's cascading problems. Facebook groups mobilized with thousands of followers demanding help. By May 2020, the Economic Policy Institute and Brookings were speaking authoritatively about what went wrong; outlets like Vox, The New York Times, and NBC were diving deep into programmatic failures. Fast Company spelled out the impact: a man leaving quarantine to fax his pay stubs at a Staples, a governor pleading for anyone who knew COBOL (a programming language launched in 1959), an agency's rule where only people with last names between A and H could call in on Mondays. As people read the news and waited weeks and months without payment, trust in UI began to falter.

And finally, the bad guys took advantage

Recognizing the scale of the crisis, legislators rushed to expand existing programs or create new ones to help the unemployed get the money they needed as quickly as possible. But these new programs were often built on a shaky foundation—legacy

applications that were not designed to accommodate these changes quickly. Agencies couldn't fix legacy systems, but they could prop them up to keep the pages live, and make sure people could still submit their applications.

But a propped-up system is still a broken one, and these systems became ripe targets for fraud. Across the world, criminals gathered in private forums and Telegram channels to discuss how to bleed cash from these systems. They stole identities. They used bots to submit bulk claims. They sold guides to overcoming rickety security systems. The U.S. Department of Labor's Office of the Inspector General estimated that at least $163 billion in UI payments were improperly paid—more than the annual state budget of everyone but New York and California.

What's next: Tenets for building the future

Even though COVID's full impact was unforeseeable, UI system failure was predictable. Before the pandemic, at least 12 states still had COBOL systems. Most ran in data centers, self- or vendor-owned rooms of towering mainframes. Few, if any, were built with the applicant's experience in mind.

Many of these problems were pervasive a little more than a decade earlier, when the 2008 Great Recession spiked unemployment to 20 percent in some states. Today, these problems still exist, masked by historically low levels of unemployment. We can, and should, solve these problems now before the next crisis hits. And thanks to the American Rescue Plan Act, billions of federal dollars are available for states to build modern, best-in-class UI systems. Now is the moment to do something big.

Based on AWS's work with UI agencies during the pandemic, we've developed a point of view on what modern UI systems should look like and how states can get there. We'll make the case for a new vision for UI benefits programs, then provide a more technical roadmap to realize that vision. We assume a cloud-first approach, but cloud is an enabler for delivering better systems, not a solution in and of itself.

At Amazon Web Services (AWS), we build our work around tenets. We find it helps us stay organized around themes while still remaining flexible enough to bend when circumstances change. We propose five tenets to anchor a vision of a modern UI benefits system.

Tenet 1: Future systems are usable

Historically, UI systems are designed to ensure compliance with policy and rules, not to optimize the claimant or worker experience. Whether it's confusing legalistic language, a lack of transparency into the process, or simply no easy way to get information on a claim's status, traditional UI systems rarely leave claimants feeling supported or heard. As Scott Jensen, the former director of the Rhode Island Department of Labor and Training put it, "If someone is applying for unemployment insurance benefits, they are already having a bad day. Nothing about the process of applying should make them feel worse."

Future systems should look to benchmark usability against the usable technology we use every day. We carry smartphones in our pockets and wear smartwatches on our wrists; clever applications wake us in the morning, organize our days, and deliver our pizza at night. The gap between public and private sector technology is even more galling because we know just how good things can be. If we can book a car ride or order a pair of shoes in minutes, why is it so hard to ask the government for help?

Your UI system will be more usable when you:

Practice user-centered design in your work

What makes private sector solutions so elegant is their commitment to user-centered design. They are customer-obsessed, investing real time and money in understanding the user experience, then working to make that experience as simple as possible. At AWS, we work backward from the customer problem to build simple and effective solutions around user needs.

User-centered design also means optimizing for the channels claimants want to use. Most systems rely on internet portals and contact centers or interactive voice response (IVR) to engage with claimants. But increasingly, claimants prefer to do business on their phones through mobile apps and text messaging.

Use artificial intelligence (AI) and machine learning (ML) to understand claimant needs and improve customer experience

Calls made to your contact centers are likely recorded for quality control. Each call is its own story: a person with a problem, a process for solving it, an ultimate resolution (or lack thereof). These calls are a treasure trove of information about what claimants and jobseekers want and need.

Historically, this data has been difficult to mine because there's just so much of it. Hundreds of agents might take thousands of calls per day. And although individual calls that go particularly wrong may be discussed, it's still difficult to look into what beneficiaries are asking for across the board. Today, however, AI/ML can help you derive critical insights to improve system usability and truly understand your customer's needs and pain points. AI/ML can help you follow the sentiment and trends of customer conversations in real time to identify crucial feedback.

You can also track the agent compliance of customer conversations in your contact center to ensure standard greetings and sign-offs are used, help train agents, and replicate successful interactions. Supervisors can conduct fast full-text search on all transcripts to quickly troubleshoot customer issues. Using real-time analytics powered by ML, you can also get alerted to issues during live customer calls and deliver coaching to agents while calls are in progress, improving customer satisfaction.

Be proactive communicators

Customers get frustrated, and rightly so, when they simply don't know where their application sits. The UI adjudication process can be complicated and lengthy, but letting claimants know the status of their claim and setting reasonable expectations around timeframes can help ease their anxiety. Future systems shouldn't wait for a claimant to ask about their claim. They should proactively guide claimants through the process and notify them about a change in status or a call to action, using the channel—either text, phone, or email—that the claimant prefers. As a baseline, citizens expect to be able to access their applications 24/7 and receive updates on their phones. Providing this access improves customer experience, reduces call volume, and streamlines workflow during business hours.

Tenet 2: Future systems are elastic

The infrastructure supporting UI systems must be elastic. It needs to be able to scale up instantly to accommodate a surge in demand, and just as importantly, scale down as the economy recovers and claim volumes ebb. Because UI funding is tied to claims volume, these systems must align costs with usage, so agencies aren't stuck paying for capacity that's no longer needed.

The cloud is born to be elastic. The AWS Cloud provides infrastructure and advanced services that can scale up immediately when needed and scale down when demand

ebbs. And because these services are available on a pay-as-you-go basis, costs are aligned with usage.

Of course, the application and business processes should also be elastic. For a UI program, this means contact centers can scale up and down as needed without having to add expensive seat licenses or other fees. It means streamlining and automating business processes, so processes that work at low volume don't find human bottlenecks at high volume. For example, in many states, resetting a PIN was a manual process requiring human intervention before a claimant could submit a continuing claim. And because contact centers were already swamped with calls during the pandemic, this process couldn't scale and many claimants were essentially locked out of the system.

Tenet 3: Future systems are adaptable

Keeping a 40-year-old COBOL system afloat during the pandemic would have been difficult enough, but UI agencies had the added burden of standing up new programs, such as Pandemic Unemployment Assistance, in a matter of weeks. Because many UI systems weren't built to be adaptable, states and their partners had to be creative. In some cases, they took old system components out of mothballs and ran temporary, parallel systems. In other cases, they created new applications to accept new claims and processed them in batch overnight to not stress the legacy applications. Agency staffs were heroic in finding solutions, but relying on staff heroics isn't an effective long-term strategy. Rather, future UI systems should be built to be adaptable, recognizing that in an economic crisis existing program rules will change, and new programs may need to come online quickly. The AWS cloud positions you to respond to ambiguous policy landscapes and the evolving needs of your customers.

Develop and deploy using a modular or microservices architecture

Part of the challenge of even more recently deployed legacy UI systems is that they are what IT architects call tightly-coupled. These systems make it difficult to change one part of the system without risking a break in another part of a system. More loosely-coupled systems, using a modular or microservices architecture, allow you to change features and functions without having to worry about undermining the entire system. They also provide more flexible deployment options so faults can be identified and resolved before they impact your entire user population.

Tenet 4: Future systems are smart

During the pandemic, UI agencies struggled because they relied on highly manual business processes that were adequate at low volume but collapsed as claims volumes surged. Facing unprecedented adjudication workloads, agency staff were bogged down with important but routine tasks—like collecting data from different sources—where their expertise and judgment was not needed. At AWS, we call this sort of work undifferentiated heavy lifting—important work that should be automated, freeing workers to focus their time on higher value activities.

With AWS, agencies can leverage cloud native services as force multipliers to streamline operations. Smart chatbots on the web, over the phone, or through text messaging can answer FAQs and personalized claimant questions, such as requests for real-time claim status in multiple languages. Improved self-service options, in turn, divert traffic from the call center freeing highly skilled agents to work on more complex tasks.

Agencies can also use AI and ML capabilities to extract data from documents, flag data anomalies for adjudicators, and automate routine business processes – speeding up human review and approval. They can use machine learning models to identify potential fraudsters more efficiently and effectively, reducing the unnecessary friction placed on legitimate claimants. Lastly, they can leverage data lakes, advanced data analytics, and ML to gain high visibility into program operations, empowering program leaders to make data-driven decisions. Both real-time and batch analytics can provide insights into program operations metrics like backlogs and processing times, enabling improved benefit accuracy and timeliness.

Tenet 5: Future systems are secure

As we learned during the pandemic, UI systems can be ripe targets for bad actors, whether they are trying to extract sensitive data from these systems, use ransomware to hold the system hostage, or defraud the agency by submitting fraudulent claims. Moving to the cloud can help agencies address each of these threats more effectively.

Most systems improve their security posture when moving to AWS, because AWS invests substantially more in its security posture than any individual company or agency would be able to on their own. Importantly, this level of investment in cutting-edge security is a critical difference compared to so-called private cloud providers. With AWS's shared security model, AWS is responsible for security of the

cloud, while the customer is responsible for security in the cloud. And with an extensive set of services and tools to help maintain application security, AWS makes that simpler.

Ransomware is using software vulnerabilities to cripple a system until the agency pays a ransom. And it's an increasing threat to government agencies. One tool to combat the threat of ransomware is to have a robust disaster recovery (DR) strategy. That way, even if a primary system is exploited, the agency can switch almost seamlessly to another instance without losing time or data. The AWS Cloud provides lots of DR options to protect these mission-critical systems.

The early pandemic headlines about UI covered problems related to filing claims and receiving benefits. But more recently, the focus has been on widespread UI fraud. A combination of policy decisions and system limitations opened the door to unprecedented fraud, primarily from imposters filing false claims. In response, many states stopped making payments until they could bolt on identity verification and other fraud detection services. Even though those services were generally effective in reducing imposter fraud, they also made it more difficult for legitimate claimants. A modern UI system should integrate fraud detection and prevention into the application from inception, and leverage AI/ML to detect fraud more effectively.

You may have better tenets for your unique circumstances. But having tenets, whichever they are, is a great way to get and stay aligned as you modernize your UI systems.

How to modernize your UI system in the cloud: A Technical Roadmap

There are several different approaches UI agencies can use to modernize their UI systems, with varying degrees of effectiveness. However, AWS believes the future of UI systems resides in the cloud. This can involve migrating existing obsolete UI systems into the cloud or building new modern UI systems on a cloud-native platform.

The cloud provides several important benefits which improve the capabilities of UI systems:

1. **Managed services.** You can shift undifferentiated heavy lifting away from your IT staff, allowing them to spend more time improving UI applications and enhancing business performance. Migrating to the cloud or modernizing in the cloud provides access to a wide variety of services and resources to support your UI applications.
2. **Security is top priority.** Cloud service providers (CSPs) make significant investments in physical and application security which far exceed what most organizations can achieve on their own. For AWS, providing our customers with a safe and secure environment is our number one priority.
3. **Pay only for what you use.** During periods of increased unemployment and higher claim loads, scale up your application as needed to provide uninterrupted service to your claimants and staff. During periods of lower unemployment, scale down to reduce operational costs. You no longer have to pay to maintain worst case capacity at all times.
4. **Infuse your application with AI/ML.** You can integrate services offering transcription, translation, text extraction, chatbots and more, even while your application is still running on premises. You don't have to complete a migration or modernization to start reaping these benefits.
5. **Do more with your data.** You can store your data securely in the cloud, at reduced cost and with greater durability. You can use data analysis and visualization tools to extract meaningful insights. You can archive older data for additional cost savings and utilize expiration policies to meet regulatory compliance requirements.

Modernization approaches

Migrating or re-architecting a large, obsolete UI system is a complex undertaking. There could be decades of code changes, often with little or no documentation to provide insight regarding the reason for or impact of the changes. There could also be decades of data, of varying quality, that needs to be migrated.

Embarking on a modernization journey can involve approaches such as:

- Enhancing your existing system (incremental modernization)
- Procuring a commercial off-the-shelf (COTS) product
- Developing a new system on an enterprise platform

- Developing a new system from scratch (custom development)

These approaches typically require lengthy and expensive projects that could take years before your users and claimants start to realize the benefits of your modernization efforts. There are a variety of ways to go about modernizing your legacy system. Some can even be combined to provide a better return on investment (ROI) in a shorter (and in some cases, much shorter) time frame.

In this section, we describe five popular alternatives for UI agencies considering modernization and/or replacement of their existing UI systems. These alternatives are:

- Around the edges
- Strangler fig pattern
- Lift and shift
- Language migration
- Emulation

Around the edges

To address short-term modernization needs, this approach adds new capabilities using modern technology wrapped around the edges of your existing system. You build integration points where needed, without over engineering. Using this approach, you can prioritize investment in changes to your existing system to reflect customer and user needs and space out changes over time to accommodate availability of budget and funding. In addition, you will quickly begin realizing the benefits of your modernization efforts and not have to wait until completion of a full system replacement. For example, with AWS, you can use Amazon Translate to translate email, voice, and text into a claimant's preferred language, reducing the chance of error due to misunderstanding.

Strangler fig pattern

This approach is rapidly gaining popularity. First, you identify components or capabilities to modernize within your existing system. Then, you build replacements for these components and strangle off the existing components. Your claimants and users benefit as soon as the new components are deployed to production. You can cut over gradually and allow customers to become familiar with the new user experience,

thus increasing the probability of user acceptance. You can also take advantage of modern web development practices.

Even though the strangler fig pattern isn't new, the cloud removed some of the difficulties with its implementation. For example, you can now use managed data replication tools to replicate and synchronize databases, both on premises and in the cloud. This makes it easier than ever to use your data in new ways.

Similar to the around-the-edges approach, your investment in changes to your existing system in the strangler fig pattern can be prioritized to reflect customer and user needs and spaced out over time to accommodate availability of budget and funding. In addition, you will quickly begin to realize the benefits of your modernization efforts and not have to wait until completion of a full system replacement.

Lift and shift

This approach involves moving a copy of an existing application and data to cloud infrastructure with minimal or no redesign or modification. Applications utilizing Microsoft Windows or Linux-based operating systems can benefit in several ways from running in the cloud. First, using cloud-based capabilities like elastic load balancing and auto-scaling groups can enable these applications to scale dynamically, accommodating anticipated increases in demand or increases due to more disruptive events. During periods of lower activity, the application automatically scales down, reducing costs. The cloud also offers a wider variety of virtual servers, allowing you to tailor your infrastructure specifically to meet your application needs while minimizing unused capacity and cost.

Running in the cloud can also simplify your backup and disaster recovery strategies. You can schedule backups for your databases and file systems, storing them inexpensively in Amazon Simple Storage Service (Amazon S3) buckets. You can also utilize multiple availability zones and regions to meet or exceed your recovery time and recovery point objectives so you are always able to serve your constituents' needs, even in unforeseen circumstances.

Lastly, running your application in the cloud provides access to a host of additional services that can augment your existing capabilities. For example, you can use data warehousing and analytics tools to quickly gain insight from your data while paying only for the resources you use without lengthy procurement and configuration cycles.

Or, you can engage with your customers in new ways using SMS messaging and push notifications to improve the ease with which you provide services.

Language migration

This approach provides a compelling mechanism for moving on-premises COBOL applications into the cloud. Tools offering language migration typically take existing application code and convert it into a more modern language (like Java) and, in some cases, create APIs to expose desired functionality as web services. Legacy data stores are migrated as well, usually to relational databases. You also benefit from a more readily available pool of developers in the newer target language. However, you are still using essentially the same system, with its flaws. Language migration can therefore be a first step toward strangling off pieces of functionality, as described using the strangler fig pattern.

Emulation

This approach also provides a compelling mechanism for moving on-premises COBOL applications into the cloud. Emulation simulates the execution environment of your existing system, e.g., an IBM mainframe or an AS/400. You then lift your application more or less as-is into this new execution environment which is now running in the cloud. This still leaves you dependent on COBOL developers and operators, but may make it easier to begin to refactor portions of the system using the strangler fig pattern.

Comparison of modernization approaches

The table below summarizes and compares the approaches presented above and includes pros and cons for each approach with examples of when a particular approach might suit your needs.

Modernization approach	Pros	Cons	Ideal for an agency that...
Around the edges	- Adds new capabilities for staff and claimants - Low-risk approach with loose coupling - Can be used by both legacy and modernized systems	- Doesn't change behavior in your current system	... has an immediate need to solve a particular problem without making significant changes to their existing system.
Strangler fig pattern	- Incremental path to replacing system components in order of ease or importance - Offers the opportunity to take advantage of modern languages and cloud-native services	- You can't tackle everything at once, so it may take a while - Some things may be particularly difficult to strangle and will require an alternate approach	... wants to develop a custom replacement for their existing system while realizing benefits from new features deployed incrementally.
Lift and shift	- Can reduce the cost to run your application - Offers the opportunity to use cloud services alongside an existing application	- Some architectural changes may be required to fully realize cloud benefits - Still the same system with the same limitations	... has a system they're happy with and wants to benefit from the cloud's elasticity as well as take advantage of other cloud capabilities.

Modernization approach	Pros	Cons	Ideal for an agency that...
Language migration	- Increases system maintainability by converting code into a more modern language - Can usually be done relatively quickly	- Migrated code can be difficult to decipher, as it is still very closely tied to the legacy code and legacy platform - Legacy platform utilities may not have a clean migration option and require additional effort	... would like to move off of the legacy platform quickly and move to a more modern language. This may accelerate use of the strangler fig pattern.
Emulation	- Eliminates reliance on legacy hardware which may be difficult or expensive to maintain - Migrates data to the cloud, which may offer around-the-edges opportunities	- Requires the same knowledge to support the system with the added complexity of the emulation layer	... must quickly get off their aging on-premises hardware without the time to pursue any of the above options.

Summary

The right modernization approach for you depends largely on your unique circumstances. Some states may need a wholesale replacement of an antiquated UI system, while for other states it may make more sense to improve usability and add functionality while leaving the core system intact. Whichever approach is right for you, AWS can help. By providing an elastic, secure platform with over 200 advanced cloud native services to drive innovation, AWS should be the strategic choice to support your modernization efforts today – and tomorrow.

For more information about how AWS can support your state with UI modernization, please visit the AWS for Labor and Workforce main page and AWS Customer Enablement page. To learn how AWS can support your unique needs, contact your AWS account executive, or complete this form. AWS can help support the journey with in-person and online trainings, acceleration programs, or professional services support.