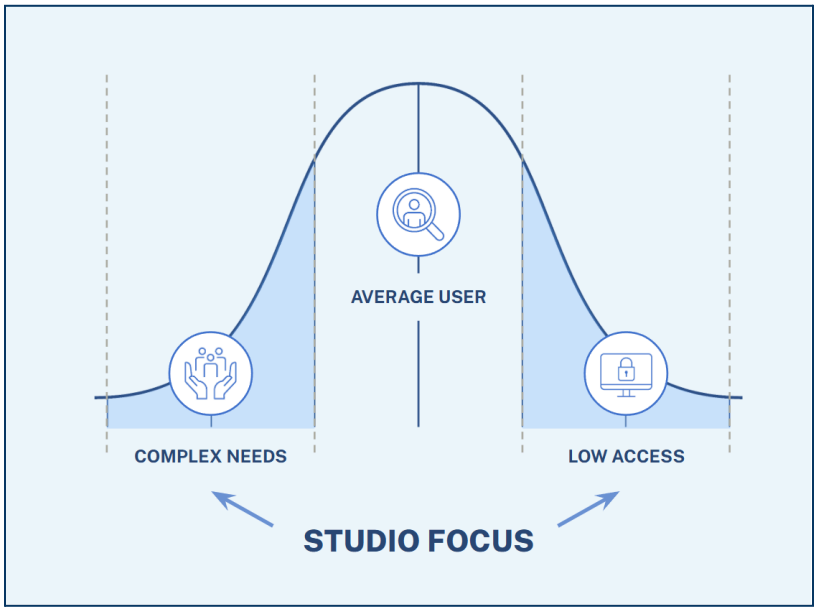
Document extraction to accelerate application processing

Summary of product exploration and proof-of-concept development

October 2024 - May 2025

About the Public Benefits Studio

The Public Benefits Studio is a domain-focused product accelerator within GSA's Technology Transformation Services (TTS), founded to incubate and scale shared technology infrastructure that makes government more efficient and effective. The Studio focuses on public benefits as a domain because designing for extremes, such as the high-stakes, high-volume nature of benefits delivery, pressure-tests products to work well for everyone. This borrows from Stanford and IDEO's concept of "designing for extremes," where solving problems for people who face the most complex challenges results in solutions that serve everyone else, too.



The Studio's design approach: focusing on users with complex needs and low access to ensure solutions are inclusive, resilient, and broadly applicable by testing against the most challenging use cases.

In 2023, the Studio launched its first shared service, Notify.gov, a bulk one-way text messaging platform. Building on that success, in 2024, the team evaluated a short list of new products and zeroed in on better document submission, addressing chronic pain points around how both the public and government staff submit, process, and extract data from critical benefit-application documents.

The Challenge

State and federal benefits programs today face three intertwined challenges that make document submission both a user pain point and an administrative quagmire:

1. **Rising Mobile Submissions, Growing Back-End Burden**
   Nearly one-third of Americans earning under $30K rely solely on mobile phones for internet access, and since 2019 mobile-responsiveness in benefits portals has jumped 25%. Yet while front-end uploaders have become more flexible, accepting PDFs, photos, even crumpled or handwritten forms, these unstructured files simply shift work downstream. Administrators now spend inordinate hours classifying, verifying, and keying in data from documents like W-2s, 1099s, DD214s, and others into data-management systems, delaying benefit determinations and diverting staff from higher-value tasks.

2. **Manual Work & Error Risk**
   Because forms arrive in inconsistent formats, staff must decipher legibility issues, parse multi-part names, handle non-English characters, and wrestle with poor image quality. This manual extraction process is slow, error-prone, and mentally taxing, leading to higher operational costs, approval delays, and mounting frustration for both processors and applicants.

3. **Procurement & Technology Gaps**
   Although commercial OCR exists, they are expensive, difficult to integrate into existing workflows, and as such, agencies typically procure through large, multi-year modernization projects that leave tools outdated before launch. Off-the-shelf solutions often demand heavy customization and external expertise — locking agencies into costly contracts and stymieing rapid iteration.

Potential Cost Savings for automating data extraction.

- **Processing could be 3x faster.** Automation significantly reduces document processing times according to a USDA RPA case study.

- **50 million hours[1] of staff time could be saved.** If we consider the total number of hours saved in the human services industry alone.

- **Potential to reduce manual error rate from 31%** - which is currently the rate of manual document flows leading to inaccuracy (ABYY).

To break this cycle, government agencies need a secure, scalable document-processing solution that can:

- **Automatically classify** commonly requested documents as part of application processes
- **Accurately extract** key data fields — even from low-quality images
- **Continuously improve** through model training and feedback loops
- **Work alongside** existing agency case-management workflows and systems including, integrating when possible.
- **Deploy safely** within a compliant government cloud environment

At the same time, teams must balance speed, ongoing costs, and long-term maintainability, including carefully weighing commercial versus open-source OCR options, so that upgrades stay current, affordable, and under agency control.

Our Approach

Problem Framing & Validation Research

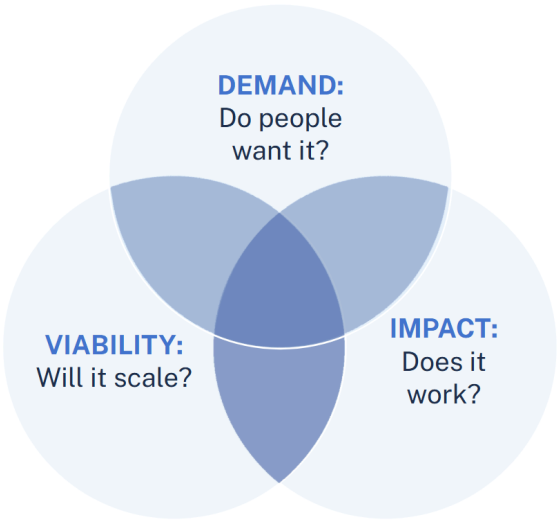
We adopted a mulit-phase, research-driven methodology to ensure we fully understood the problem before building a solution. In the Problem Framing & Validation Research phase, which spanned three months, we kicked off with rapid, low-fidelity mock-ups of various submission and processing workflows. By presenting these sketches to agency staff and document-solution experts, we gathered direct feedback on what features were most desirable and how users would realistically interact with the tool.

---

[1] Number of hours saved based on the following calculation: *(s*h)*(w)*(e) = (.30*8)*(48)*(433000)

- s = % time saved by using automated processing tools (https://fns-prod.azureedge.us/sites/default/files/resource-files/snap-rpa-ct-infographic.pdf)
- h = hours spent on manual document processing a week (https://www.abbyy.com/company/news/amidst-the-great-resignation-52-percents-of-employees-believe-ai-skills-will-make-their-job-easier-according-to-abbyy-survey/)
- w = working weeks in a year
- e = estimate of the total number of employees in human services jobs (https://www.bls.gov/ooh/community-and-social-service/social-and-human-service-assistants.htm)

Simultaneously, we conducted a comprehensive landscape scan and market analysis, mapping out existing vendor offerings and pinpointing critical gaps — especially the lack of modular, government-friendly OCR and extraction services. From these activities, we narrowed in on the hypothesis that document data extraction solution has the potential to deliver quick, high-impact wins by automating repetitive data-entry work.



To decide whether to proceed with a proof-of-concept, we applied TTS's "Desirable-Viable-Feasible" framework. We confirmed strong demand among state agencies and benefits experts, a compelling return on investment through staff-hour savings, technical feasibility using mature OCR (AWS Textract), and broad applicability across programs.

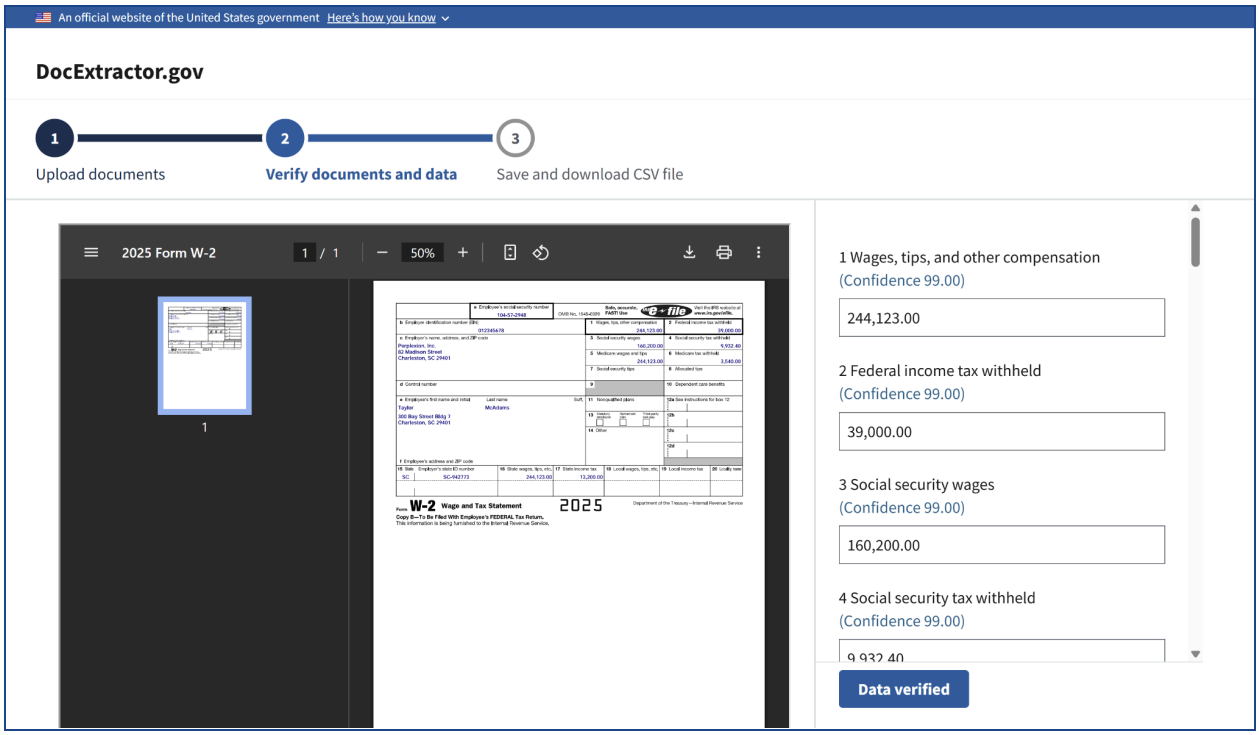
Armed with these validated insights and clear criteria, we transitioned into the implementation phase to rapidly prototyping a document-extractor PoC that could be tested, refined, and ultimately scaled.

Prototyping Phase and Technical Development

As part of the prototyping phase, we aimed to answer a central question:

> ***How might we convert uploaded documents (PDFs, images) into machine-readable data to reduce the manual data-entry burden for benefits administrators?***

To address this, we developed a web-based PoC that ingests various file formats, including PDF, JPEG, and PNG, and extracts key fields using a combination of OCR and NLP technologies. The extracted data is presented in editable fields displayed next to the document image, enabling quick human review. Users can then export the cleaned data in CSV or JSON formats for easy integration into existing data-management systems.



DocExtractor.gov prototype interface showing the simple three-step process for document upload, data verification, and CSV export, designed for ease of use and fast integration into agency workflows.

We began with baseline accuracy testing using real-world document samples, including W-2s, 1099s, and DD214s (Military discharge papers), documents that were cited in research as key to unemployment applications. These were submitted in diverse formats such as PDF downloads, mobile phone photos (both crumpled and clean), and screenshots. This helped us establish the extraction engine's baseline performance and deepen our understanding of user pain points. Comparative evaluations of OCR tools, Tesseract, PaddleOCR, and EasyOCR, revealed that AWS Textract was the most accurate and adaptable for structured data, making it our preferred engine for the PoC.

To improve classification and processing efficiency, we implemented a keyword-and position-based logic to automatically classify form types. This logic achieved 100% accuracy in test scenarios and, along with backend optimizations such as Lambda pre-warming and SQS tuning, increased processing speed by 33%. We modularized the codebase to support robust unit testing, and added authentication to ensure security standards.

As development progressed, we trained the system with a diverse dataset, including edge-case names and non-English characters. This training led to a 14% accuracy improvement across development rounds. Refinements to the extraction logic using Textract queries helped reduce data clutter and improve field-level accuracy. To further support operational readiness, we refactored our backend architecture for modularity,

implemented CI/CD workflows with GitHub Actions, and deployed the application into GSA's AWS environment.

UI enhancements were also made to improve perceived speed and user trust, ensuring that the user experience remained seamless alongside technical advances. This iterative and technically rigorous approach ensured our PoC delivered both high accuracy and operational reliability, core requirements for reducing administrative overhead in document processing.

Tech Stack

- Backend development.
    - Python.
    - uv project and package manager.
    - Ruff linting and formatting.
    - Pytest.
    - Clean Architecture for flexibility and module testing.
- Front-end development.
    - JavaScript.
    - React.
    - USWDS design system.
    - Parcel bundler.
    - Prettier.
    - ESlint
- Pre-commit.
- Cloud.
    - AWS (GSA environment).
    - Textract for OCR (with custom queries and training adapters).
    - CloudFront.
    - API Gateway.
    - Lambda.
    - SQS.
    - DynamoDB.
    - Secrets Manager.
    - S3.
    - Terraform for IaC.
- CI/CD.
    - GitHub Actions.
    - Dependabot for dependency updates.
- Authentication: SSO-based login with optional Login.gov integration (in planning).

Outcomes

Across three rounds of testing with both real and synthetic documents, we achieved an average accuracy improvement of 14%. Our system delivered 100% classification accuracy for 1099 forms and 99% for W-2s. Performance optimizations, including reduced Lambda cold starts and improved queue handling, resulted in a 33% faster processing time. The application was securely deployed to the cloud with automated release pipelines, meeting operational compliance requirements. Additionally, user trust was enhanced through improved UI feedback, including clearly displayed classification labels and responsive loading indicators.

Future Potential

The success of this PoC has opened several promising paths:

- **Agency-facing deployment and low-code integration:**
    - Once production ready, this tool could be deployed to government agencies directly, enabling them to improve internal document processing.
    - To ease implementation and extend usability, there is potential to build a complementary low-code interface that would allow agency staff to map extracted data fields to their existing workflows without extensive technical overhead.
- **Expanded document coverage:** With more training data, the extractor can handle a wider range of documents, including multi-page or handwritten forms.
- **User feedback loops:** Human reviewers can flag anomalies or train the tool further, creating a continuous cycle of improvement.
- **ATO readiness:** While formal Authority to Operate (ATO) is not currently prioritized, the groundwork has been laid for a production-grade deployment when needed.