International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 08 | Aug 2026 www.irjet.net p-ISSN: 2395-0072

Privacy-Preserving Multi-Agent Architecture for Accessible Digital Form Processing

Kolli Bhavya Sri[1], Dr. Chandu Delhipolice[2]

[1]Student, [2]Professor, Dept. of Computer Science and Engineering, Priyadharshini Institute of Technology and Science, Chinthalapudi, Guntur, Andhra Pradesh, India

---

Abstract - *Digital services have been widely adopted in government, financial services, healthcare, education and other fields, so digital form completion is becoming a routine service task. But elderly users, visually impaired users, and users with limited digital literacy will be encountered to complete the digital forms. The personal information including ID, address, bank account, phone and other information will make privacy concern. Facing above issues, this paper presents a modular multi-agent AI framework for accessible and privacy-aware digital form completion. Different agents are responsible for detecting form-fields, detecting and masking personal information, validating form field, providing accessibility, large language model (LLM) based user assistance, and the orchestrating layer coordinates activities of these agents. The prototype system was built using Python and Flask, with document-processing and OCR, speech accessibility, and an LLM assistant based on Google Gemini, deployed. At testing, using a simple admin form containing 8 target field, the system detected 8 fields, with 7 fields initially has values, initial completion rate was 87.5%. The missing Address field was then processed to complete the form. The Privacy Agent successfully detected and masked the tested national ID information. The validation and accessibility functions worked as part of the workflow. This work demonstrated the feasibility to combine privacy masking, accessibility, validation, LLM assistant for digital form completion as a modular multi-agent system.*

Key Words: Multi-Agent Systems, Human-Computer Interaction (HCI), Web Accessibility Guidelines (WCAG 2.2), Natural Language Processing (NLP), Digital Inclusion, Zero-Trust Privacy, PII Masking.

1. INTRODUCTION

As the use of digital services both within government and the financial, health and education sectors increases, individuals have become more accustomed to the completion of online forms. Despite the reduction of paper usage and ease of recording information associated with digital forms they may still cause difficulties for the elderly, the visually impaired or the digitally-literate.

1.1 Accessibility Challenges

Most on-line forms are not designed for the needs of users with a visual or physical disability. If the forms are formatted formats, contain graphics or include features such as interactive items, they may become impossible for some adaptive equipment such as a screen reader to work with. Users who will need help in completing on-line forms will be used.

1.2 Cognitive load and interaction with the user

When a user interacts with a computer system the amount of cognitive load which the user will experience must be taken in to account. If the cognitive load is high, then the number of errors the user will make will also be high. Therefore, the presentation of data also needs to be taken in to account from a usability perspective.

When filling an online form, the user has to establish a meaning for some of the fields of the form, know what data has to be included in the fields and enter the date into the form as it should be entered. This can mean an increase of cognitive load (CITADELLI, 2000), mainly when the form has a lot of fields or when the user is used to the digital interface. Guided interaction and online voice assistance can help decrease this.

1.3. Privacy and Security.

Concerns from users and other stakeholders in regards to system security and privacy as well.

The other forms that have been entered: a security number; a telephone number; a passport number; information about personal finances. 2Introduce the possibility of information being accidentally published if the data is not protected while it's being entered onto the form. Sensitive data should be designated and protected when they 're being handled.

The conventional monolithic applications carry out different forms of processing on one system. These applications are difficult to maintain due to its complex install and change process. In multi agent approach this job can be scattered into smaller, specific jobs which performed by using different components with process/orchestration engine.

In this paper, I introduce a modular multi-agent system of the accessible private form filling. In this modular multi-agent system, the detection of form field, detection & masking PII, validation, accessibility and AI help are

performed respectively by specialized agents, coordinated by an overall orchestrator. The modular system facilitates as the individual development and extension of each component, and the coordinated form filling process.

2. RELATED WORK

2.1 Extracting Meaningful Data from PDFs

In order to realize our analysis and retractions of modern document layout and fields we need the state-of–the-art OCR engine of today that the freely available PyMuPDF and Tesseract engines are. The bottom line from above crucial point is that state-of-the-art OCR engines give us generally a maximum available extraction of the structure of the text in our PDF files. But this not immediately mean that we can easily able to build up a correlation between the fields and use them in the document context in order to cluster into" phone numbers" or" e-mails" and then" heal" them with respect to available data.

2.2 Privacy Preservation across Document Parsing Pipelines

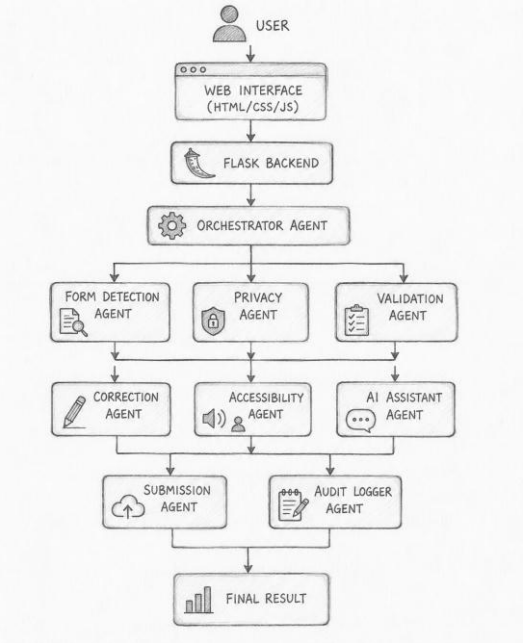
In the real practice implementations of the concern of the privacy, the entire masking policy itself would be defined in the level of the input stream either through the most primitive Regex rules engines or through the new NER models which may be able to would be able to prompt and cleanse the data stream of any sensitive personally identifiable information (Phone numbers, e-mails etc.). Both the end to end approaches are quite limited in its scope and usability while working in the real world scenarios as each would depend on the accuracy of the models which by the very nature of science and its variation quite subjective in the end or is heavily limited by document formatting with template only data, where the false-negative rate might be too prohibitively high for a lot of implementations. The future trends in the privacy preservation would be to specify dynamically the documents masking networks which would lead to paste and tamper the sensitive information within the documents even before they are sent to any downstream processing capability (Cloud OCR engines, LLMs etc.)

2.3 Enhancing Digital Accessibility in Web Apps

Web Content Accessibility Guidelines (WCAG 2.2) describes the best optimized design for web apps & sites can fit to the visually-audio physically-cognitively disabled user's accessible demands. But to best now, the available ingredients for accessible desktops are mainly TTS/STT (all-in-one or totally-ready-to-use for speech-to-text and text-to-speech), high contrast views, ultra-advanced ARIA roles, more 1 or 2-very limited solutions and just several products had empowered such the naturally smart Accessibility Manager: to automatically analyze the user's data knowledge & store-record the strange doc-part, then flexibly modify its layout & experience accordingly.

3. PROPOSED SYSTEM ARCHITECTURE

The system is created to run on multi-agent architecture, which is implemented on Python Flask back-end with a PyMuPDF for document extraction, multiple agents on regex and heuristics to prevent privacy issues and the Google Gemini API as natural language assistant.



3.1 Overview of the Proposed System Agent Function

3.1.1 User Interaction & Ingestion: User Input User enters the target document (Image/PDF) onto the web using the Web UI.

3.1.2. Orchestrator Agent, at its output, a payload, instead of 1st scenario, that have been created in the 1st step and it is set to be provided to successor agents.

3.1.3. Form Detect Agent: Payload is received by Form Detect Agent. It examines the raw text and finds out the form name. Let kind of form, then it constructs the key value pairs.

3.1.4. Privacy Agent: Following the original payload comparison with the many pattern matchers, the Privacy Agent camouflages it prior to the cloud API.

3.1.5. Validation & correction: Validation Agent validates the payload on the rules defined in above steps. And Correction Agent do the correction of tokens / OCR related problems.

3.1.6. Accessibility Agent: The Accessibility Agent calculates the UI usability value and starts the available

accessibility tools (TTS, STT, contrast, font-size); and answers the questions if needed. using the context of cleaned document, The AI Assistant Agent provides general help using the cleaned context.

3.1.7. Audit & Completion: The Audit Logger Agent collects the anonymized audit trail from the Audit Server and the Submission Agent uses it to check whether the job has been finished.

4. MULTI-AGENT SYSTEM DESIGN & AGENT ROLES

The division of system functions among agents helps to avoid their functional overlap and ensures isolation at the implementation level.

4.1. Orchestrator Agent

The following entity, which controls the system state, carries data between agents, is responsible for the corrected errors and the exits from the processing by saving the data to leak toward other parts of the system.

4.2. Form Detection Agent

This agent processes the form structure and data, by using the library PyMuPDF. This agent retrieves the text-bots of the form, get the label and the user data, find the document type and store everything in normalized JSON objects.

4.3. Privacy Agent

The Privacy Agent as the name indicated that it is to handle the privacy concerned issue. The privacy related functions are to operate on target data base to write-down the private data [14]. And it will block the private data in the result of the query.

4.4. Validation Agent

This agent contains several validation rules and is used to evaluate the correctness of the data, the presence of required fields, the validity of domain names for email addresses, and phone number formats. This agent is used to report validation issues and provide repair suggestions.

4.5. Accessibility Agent

This agent is responsible for ensuring that the application is compatible with the underlying platforms and used to provide defaults for things like native text-to-speech, speech-to-text input, large-scale font sizes, dark mode views, and other accessibility-related options.

4.6. Correction Agent

This agent applies rules to correct common optical character recognition errors based on context and proximity before validation.

4.7. AI Assistant Agent

This agent is designed to answer questions and provide additional context to users when asking for information using the Google Gemini large language model, while operating on the same privacy constraints as the rest of the system.

4.8. Submission Agent

This agent checks whether all fields are completed and validated and private data is masked before allowing the submission to happen.

4.9. Audit Logger Agent

This agent logs the system's core features and operations, such as event timing information and error details, without private information.

5. EXPERIMENTAL RESULTS & EVALUATION

The multi-agent system was tested on an initial web application and an experimental administrator form with Eight 8 target fields for locating the required data, Name, Email, Phone number, Date of Birth, National Identity Number, PAN number, Address and Gender.

5.1. Form Field Detection and Completion

The Form Detection Agent was able to locate all the eight (8) fields against the test payload. Seven (7) of these fields were occupied (excluding the Address field which was not populated / null). This resulted in a beginning completion metric of 87.5%.

TABLE 1: Form Detection and Completion Results

Evaluation Parameter	Observed Result
Total Fields Detected	8
Initially Completed Fields	7
Initially Missing Fields	1
Initial Completion Rate	87.5%
Missing Field Target	Address
Missing Information Completed	Yes
Final Form Status	FILLED

5.2. PII Detection and Masking

The Privacy Agent was also tested on its ability to hide privacy fields that could may have private data, such as the identification of the field that is storing the credential (national identity) and hiding it so the entire value is not exposed.

TABLE 2: PII Detection and Masking Results

Parameter	Observed Result
Sensitive Field Tested	National Identity Credential
Detection Status	Successful
Sensitive Data Classification	National Identification Number
Masking Execution	Successful
Masked Representation	[Redacted] / XXXX XXXX XXXX
Sensitive Information Exposed	None (Fully Anonymized)

5.3. Validation Agent Evaluation

The Validation agent functioned successfully for the fields that it applied to (email, phone and DOB).

TABLE 3: Validation Agent Performance

Parameter	Result
Validation Agent Execution	Completed
Field Correctness Verification	Supported
Structured Fields Processed	Email, Phone, Date of Birth
Quantitative Benchmark	Functional Baseline Confirmed

5.4. Accessibility Evaluation

The Accessibility Agent has also confirmed that the form can be filled out and submitted "hands free" providing there are dynamic front-end controls in place such as: reading the form (text-to-speech/TTS), stopping speech.

TABLE 4: Accessibility Functional Results

Accessibility Feature	Integration Status
Read Form (TTS)	Available / Integrated
Stop Speech	Available / Integrated
Voice-Assisted Input (STT)	Available / Integrated
Form Navigation Support	Successfully Integrated
Overall Agent Status	Completed

5.5. Multi-Agent Pipeline Evaluation

There are 9 agent run in the system. The system, which is made possible by the agent's modularity, succeeded in creating a pipeline of operations to be run upon the uploaded form. The operations were detecting form structure, detect form privacy and mask nonpublic info, validate form, support accessibility, create an AI assistant Chabot agent to respond to end-user questions at runtime.

TABLE 5: Multi-Agent Functional Evaluation

Agent	Primary Function	Execution Status
Orchestrator Agent	Coordinate the multi-agent workflow	Completed
Form Detection Agent	Extract form fields and create structured data	Completed
Privacy Agent	Detect and mask sensitive information	Completed
Validation Agent	Validate supported field formats	Completed
Correction Agent	Handle correction-related processing	Integrated
Accessibility Agent	Provide TTS, STT, and interaction support	Completed
AI Assistant Agent	Provide context-aware user assistance	Integrated
Submission Agent	Handle the final form-processing stage	Integrated

6. DISCUSSION AND ADVANTAGES

The multi-agent framework provides distinct advantages over traditional monolithic web applications:

6.1. Decoupled Architecture: Allows various tasks to be separated into various agents, with the effect that they can cope with without affecting each other. It can also enjoy the advantage of being able to upgrade smoothly and efficiently.

6.2. Strict Privacy Boundaries: User privacy is by design preserved through local execution layers that do PII extract and anonymize before calling to cloud APIs or at storing log files.

6.3. Inclusive Human-Computer Interaction: Automated TTS/STT voice controls integrated with multiple dynamic ARIA UI modes directly supplement WCAG 2.2 standards, massively reducing mental and subsequent physical work for all encountered consumers!

7. CONCLUSION AND FUTURE WORK

In this work, we proposed an agent-based AI system to achieve easier, safer, more private digital form filling. The proposed system would combine the deterministic local parser and privacy filter with the power of LLM and UI tools so that safe and effective automation is achievable. Future development activities can extend to multimodal OCR of complex hand-written forms, multilingual voice intersections, cryptographic, zero-knowledge audit logging.

REFERENCES

[1] W3C, "Web Content Accessibility Guidelines (WCAG) 2.2," W3C Recommendation, 2023.
[2] M. Wooldridge, An Introduction to Multi Agent Systems, 2nd ed., Wiley, 2009.
[3] P. Viola and M. Jones, "Robust real-time face detection," International Journal of Computer Vision, vol. 57, no. 2, pp. 137–154, 2004.
[4] R. Smith, "An overview of the Tesseract OCR engine," in Proc. ICDAR, 2007
[5] Google Cloud, "Gemini: A Family of Highly Capable Multimodal Models," arXiv preprint arXiv:2312.11805, 2023.
[6] A. Vaswani et al., "Attention is all you need," in Proc. NeurIPS, 2017.
[7] ISO/IEC 27001, "Information security management systems Requirements," ISO, 2022.
[8] M.N.Alanazi, "Privacy-preserving machine learning: Challenges and solutions," IEEE Access, vol. 10, 2022.