IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 7 March 2025, accepted 22 April 2025, date of publication 24 April 2025, date of current version 2 May 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3564133

RESEARCH ARTICLE

Implementing Generative AI Into ERP Software

SIAR SARFERAZ, (Member, IEEE)
SAP SE, Research and Development Department, 69190 Walldorf, Germany
e-mail: siar.sarferaz@sap.com

ABSTRACT Enterprise Resource Planning (ERP) systems digitize all business processes within companies in order to enhance automation and optimize efficiency. These solutions integrate data and processes across multiple functions such as sales, marketing, finance, supply chain, manufacturing, services, procurement, and human resources, serving as a central repository of information for numerous organizations. ERP systems typically encompass tens of thousands of business processes and manage data across thousands of tables, creating significant opportunities for the integration of Generative Artificial Intelligence (AI) for increasing process automatization and optimization. Nonetheless, embedding Generative AI into ERP solutions is a complex task due to the intricate nature of these systems, which consist of hundreds of millions of lines of code and cater to a wide array of industry-specific and regional requirements. Consequently, the key research question addressed in this paper is: How to systematically develop and operate Generative AI business applications in ERP systems? This article aims to answer this question by conducting a use case analysis, deriving business requirements, designing and implementing a solution framework, and evaluating its effectiveness through real-world ERP use cases.

INDEX TERMS Enterprise resource planning, ERP, artificial intelligence, AI, generative AI, enterprise AI, business AI, business applications, software integration, AI development, AI operations.

I. INTRODUCTION

In today's rapidly evolving digital landscape, the integration of advanced technologies into enterprise systems is not just beneficial but essential for maintaining competitive edge and operational efficiency [1]. One of the most transformative technologies to emerge recently is Generative Artificial Intelligence (AI), which has demonstrated remarkable potential in fields ranging from natural language processing to creative content generation. This paper delves into the implementation of Generative AI within Enterprise Resource Planning (ERP) software, exploring the synergies that can drive unprecedented enhancements in business processes. ERP systems serve as the backbone of modern enterprises, orchestrating a multitude of business functions including finance, human resources, supply chain management, and customer relationship management. Traditionally, ERP systems have been designed to streamline these functions through automation based on rule-based techniques where developers predefine the instructions. However, the integration of Generative AI represents a significant leap forward, augmenting ERP capabilities with intelligent automation based on self-learning techniques where machines derive the instructions from data.

Potential benefits of Generative AI for ERP are numerous and can significantly improve the user experience, streamline content creation, and enhance developer productivity. One of the key benefits is the improvement of the software and service experience for customers. By enabling interactions with the software using natural language, users can more easily navigate the system and access the functionality they need. Automation in customer support can lead to quicker resolution of issues and improved satisfaction levels. Conversational retrieval of information allows users to obtain the data they need more efficiently, making the entire user experience more enjoyable and productive. Another benefit is the assistance provided in content creation and knowledge management. Generative AI can generate or improve various types of content, such as marketing and sales prospectus, making it easier for businesses to communicate their value proposition to their customers. Additionally, the models can help in summarizing ERP documents and data, enabling users

The associate editor coordinating the review of this manuscript and approving it for publication was Anandakumar Haldorai.

to quickly understand the key points and make informed decisions. Lastly, Generative AI can increase the speed and effectiveness of developers working with ERP. With features such as code generation from natural language and code auto-completion, developers can work more efficiently and reduce the time it takes to bring new features or improvements to market. Automated generation of documentation also ensures that customers have access to accurate and up-to-date information, further streamlining the development process.

By leveraging the power of these Generative AI models, non-technical users as typical ERP users are, can now solve their business tasks simply by describing them in natural language. This opens up a wide possibility for businesses and individuals alike, empowering them to harness the capabilities of AI without requiring extensive technical and data science expertise. Large language models (LLM), like OpenAI's GPT series, are designed to be broadly applicable to a multitude of tasks without the need for re-training by consumers. By employing these models, businesses can streamline their processes, enhance their decision-making, and automate repetitive tasks, ultimately driving growth and innovation. The adaptability of general-purpose foundation models is another key advantage. These models can be fine-tuned to perform specific tasks in a relatively short amount of time, allowing organizations to quickly deploy AI solutions tailored to their needs.

However, implementing Generative AI within the ERP domain poses significant challenges. It requires not only an in-depth expertise in Generative AI methodologies but also a thorough understanding of the technology, programming model, and business processes inherent to ERP systems. Effective integration of Generative AI into ERP systems demands a user-centric approach that acknowledges the everyday realities of business users. These clerks and specialists focus on specific business process steps without AI or data science expertise. The Generative AI functionality must therefore seamlessly embed within their familiar workflows, providing assistance at precisely the right moment and context without disrupting established procedures. This contextual awareness must extend to understanding the nuances of different business roles and their specific needs, ensuring that a procurement specialist receives different Generative AI support than a financial analyst. Enterprise readiness of Generative AI technologies presents another significant hurdle. For ERP integration, these technologies must for example satisfy legal compliance across various jurisdictions, maintain strict data privacy and protection standards, support localization across different regions and languages, provide the extensibility and configuration capabilities that enterprise software demands. The systematic implementation across diverse business processes compounds these challenges. ERP systems contain hundreds of thousands of business processes, each with its own logic and requirements. Any Generative AI solution must scale across this vast landscape. This demands an architectural approach that identifies common patterns while accommodating legitimate variations. The solution must also perform consistently across different industries, company sizes, and regions - the hallmark challenge of standard software versus custom development. To address these challenges, we will begin by reviewing existing literature on the application of Generative AI in ERP contexts, followed by explaining our research methodology. We will identify and address identified gaps through our proposed solution architecture. Lastly, we will evaluate the outcomes of applying our approach to real-world scenarios.

II. LITERATURE REVIEW

We carried out the procedure [2] of systematic literature review to find, assess, and interpret the scientific papers relevant to our research question: How to systematically develop Generative AI business applications in ERP systems? To explore this question, we composed the search term "(Generative AI OR Generative Artificial Intelligence) AND (ERP OR Enterprise Resource Planning) AND (Development OR Implementation) AND (Operations OR Lifecycle Management)". We chose the databases ACM Digital Library, AIS Electronic Library, DBLP Computer Science Bibliography, IEEE Xplore/IEEE-IET Library, SpringerLink, Google Scholar, Wiley Online Library, and ScienceDirect, based on criteria such as relevance of content and accessibility of licenses. We applied the search phrase on the selected databases. However, as we often found nearly no results, we simplified the search term by removing some of the AND conditions and determined 751 papers. After reviewing the titles, abstracts, and full texts of the publications we identified, we narrowed it down to eight papers for discussion in this section.

The paper [3] introduces a taxonomy for foundation model (FM) based systems through the lens of software architecture, highlighting their role as fundamental building blocks for future AI systems. The taxonomy categorizes FM-based systems into three areas: pretraining and adaptation, architecture design, and responsible-AI-by-design, aiming to provide guidance for architects in designing such systems. The paper also acknowledges challenges such as responsible AI issues and the complexity of integrating FMs into software systems, addressing these with potential solutions like reinforcement learning from human feedback and adversarial testing. The strength of the paper lies in its comprehensive approach to systematically categorizing foundation model-based system designs, providing clarity in a relatively unexplored area of software architecture. However, its focus is broad and theoretical, lacking specific practical applications or case studies that demonstrate the taxonomy in actionable scenarios. The paper does not contribute directly to the topic of implementing Generative AI into ERP software because it does not address ERP-specific challenges, use cases, or integration strategies. Its theoretical framework and taxonomy are too general to offer practical insights or implementation guidance for ERP systems.

The paper [4] examines the disruptive role of Generative AI in transforming digital enterprise platforms from product to innovation platforms. By focusing on Salesforce, the study employs a single case-study research design and a grounded theory approach to identify the benefits, challenges, and implications Generative AI presents for platform capabilities, architecture, and governance. The research aims to develop a theory on integrating Generative AI in enterprise platforms by theorizing actionable steps for platform owners to leverage its advantages while mitigating challenges. One of the strengths of the paper is its thorough methodological approach, combining interviews and case studies to gather diverse insights, which offers a comprehensive view of Generative AI's ecosystem implications. However, the paper's limitation is its focus on a single platform, which may restrict the generalizability of its findings to other enterprise platform ecosystems. Despite its exploration of Generative AI's transformative potential at a higher level, the paper does not specifically address how to integrate Generative AI into ERP software, rendering it less relevant for the direct implementation of Generative AI in ERP systems. It offers broad insights into digital platform transformation but lacks applicable strategies or technical details pertinent to ERP-specific applications of Generative AI.

The paper [5] provides a comprehensive review of the applications of Generative AI within the context of 6G networks, emphasizing its role in addressing challenges related to network complexity, data traffic, and personalized services. It systematically analyzes existing literature and potential uses of Generative AI to optimize network functions, enhance security, and offer personalized media content. Furthermore, the paper suggests directions for future research, focusing on model scalability, ethical considerations, and integration with existing and future network frameworks. The strengths of the paper lie in its methodical approach to identify research gaps for the integration of Generative AI with 6G. However, the paper's weakness is its narrow focus on 6G network management, which limits its applicability to other areas where Generative AI could be beneficial. The paper does not contribute to the topic of implementing Generative AI into ERP software because it is exclusively focused on the telecommunications and 6G networks domain, providing no insights, methodologies, or frameworks applicable to ERP systems. It lacks any discussion or exploration of enterprise applications, ERP functionalities.

The paper [6] explores the integration of Natural Language Processing (NLP) and Generative AI within ERP systems to enhance fraud prevention, focusing on areas like corruption, asset misappropriation, and financial statement fraud. It proposes a detailed roadmap for developing and integrating APIs with these technologies to address identified vulnerabilities, emphasizing the potential of such integration in advancing anti-fraud measures. The study encourages further empirical research to validate proposed solutions in practical settings, underlining the interdisciplinary approach involving ERP systems, fraud detection, NLP, and Generative AI. The paper's strengths include a clear identification of ERP vulnerabilities through the fraud tree taxonomy and providing a comprehensive integration roadmap for the accounting and finance professions to adopt AI-driven fraud prevention technologies. However, its weaknesses lie in the lack of empirical validation of the proposed solutions and the absence of concrete case studies demonstrating the real-world applicability of the integration approach. While the paper provides valuable insights into fraud prevention within ERP systems, it does not contribute directly to the topic of implementing Generative AI into ERP software for broader purposes beyond fraud prevention. Its focus remains limited to fraud-related applications rather than exploring the full spectrum of Generative AI capabilities within ERP systems.

The paper [7] explores the integration of Generative AI models like ChatGPT and Bard into intelligent manufacturing systems, showcasing their potential to revolutionize various aspects of the manufacturing process. By leveraging natural language processing capabilities, these models enhance communication, predictive maintenance, and supply chain management, fostering a more adaptive and responsive manufacturing environment. The study underscores the impact of AI on improving efficiency, productivity, and competitiveness in the manufacturing industry. Strengths of the paper include its comprehensive examination of AI's role in multiple areas of manufacturing, and its systematic approach to exploring current literature and trends. However, weaknesses lie in its lack of specific focus on ERP systems and the challenges of integrating Generative AI into existing manufacturing IT infrastructures. The paper contributes less to the topic of implementing Generative AI into ERP software because it primarily concentrates on intelligent manufacturing applications rather than ERP applications in general. It neglects the unique requirements and challenges of integrating AI into ERP systems, making it less relevant for discussions centered around ERP and Generative AI intersection.

The paper [8] proposes a comprehensive architecture leveraging Generative AI to optimize accounts receivable processes across industries, aiming to enhance efficiency, accuracy, and overall effectiveness. It addresses challenges like customer segmentation, credit management, dispute verification, and email customization, with a focus on reducing the cash conversion cycle and improving cash flow. Techniques used include OpenAI's GPT, Langchain SQL Agent, and prompt engineering, resulting in enhanced operations and significant reductions in processing time and cash conversion cycles. A key strength of the paper is its detailed approach to leveraging Generative AI and related technologies to address specific operational challenges in accounts receivable management. However, a notable weakness is its heavy reliance on generative output, which can occasionally yield unintended or irrelevant responses, necessitating manual intervention and potentially undermining trust in the system. The paper contributes minor to the topic of implementing Generative AI into ERP software because its focus is narrowly on accounts receivable management and does not discuss broader ERP integration or how Generative AI can be embedded or utilized across a comprehensive ERP system.

The paper [9] discusses how Generative AI, particularly ChatGPT, can serve as an effective assistant in Human Resource Management (HRM) by aiding human resources professionals in both strategic and operational tasks. It offers guidelines for creating effective AI prompts and demonstrates their application across various HRM domains like diversity recruitment and strategic human resources management. The paper highlights the importance of trained professionals for maximizing the efficacy of AI tools and suggests a verification process to ensure the validity of AI-generated suggestions. The strength of the paper lies in its comprehensive approach to explaining the applications of Generative AI in HRM and in providing actionable guidelines for prompt creation. However, a significant weakness is the reliance on AI-generated case studies that may be fictitious, compromising the reliability of the recommendations without a robust verification process. While the paper provides valuable insights for using Generative AI within HRM, it doesn't contribute to the specific topic of integrating Generative AI into ERP software. This is because the focus remains exclusively on HRM functions and doesn't address the broader or technical implications of adapting AI capabilities within ERP systems.

The paper [10] explores the application of Generative AI in automating software testing processes specifically for ERP systems. It highlights the creation of a chatbot integrated with Generative AI to automatically generate test cases, scenarios, and data for effective software testing, aiming to address challenges faced due to the complexity of ERP systems. The study demonstrates the efficiency gains in software testing through automation but does not directly explore the implementation of Generative AI in broader ERP functionalities beyond testing. The paper provides a comprehensive approach to utilizing Generative AI for enhancing software testing processes, which can significantly improve efficiency and accuracy. However, it focuses narrowly on testing facets without addressing potential contributions of Generative AI to other ERP domains, limiting its applicability in broader ERP system functionalities.

In conclusion, although the papers obtained through systematic literature review focus on the use of Generative AI in business applications, they do not address the challenges involved in developing and operating Generative AI within ERP systems. Particularly, we could not find any ERP AI solutions for utilizing as foundation for our work respectively as comparison with our framework. This aligns with our observations over the past years, indicating that this area has yet not been extensively researched.

III. RESEARCH METHODOLOGY

After conducting a thorough literature review that did not reveal any major gaps and solutions in how Generative AI is implemented in ERP software, we decided to explore the issue more deeply. We did this by closely studying specific use cases [17], [18] of Generative AI within the ERP area. We followed these use cases from the initial analysis stage through development and into their operational phase. Through a repetitive process, we spotted gaps, suggested solutions, evaluated improvements in following use cases, and adjusted our solutions accordingly.

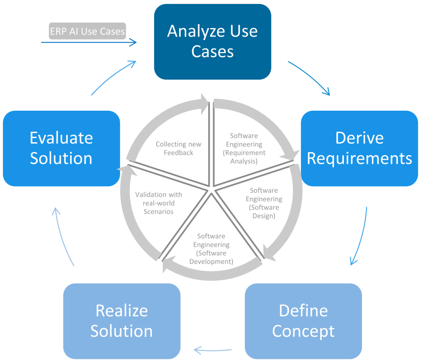

ERP AI Use Cases
Analyze Use Cases
Derive Requirements
Define Concept
Realize Solution
Evaluate Solution
Collecting new Feedback
Software Engineering (Requirement Analysis)
Software Engineering (Software Design)
Software Engineering (Software Development)
Validation with real-world Scenarios

FIGURE 1. Process model (outer circle) with applied methods (inner circle).

During our use case analysis, we focused on key questions such as whether Generative AI was the best approach to solve the underlying problem or if rule-based methods would work just as well. We also looked at what technical features were needed to successfully implement the Generative AI use cases. To make sure we comprehensively covered the topic, we examined use cases within core ERP business processes like bringing ideas to market, sourcing and paying for supplies, planning to fulfill orders, managing leads to cash flow, handling recruitment to retirement processes, acquiring and decommissioning assets, and overseeing governance and financial operations. From our analysis, we distilled business requirements and confirmed these with feedback from customers and domain experts. We found that the main gap was the absence of a framework to guide the development and operation of Generative AI use cases in the ERP context. To fill this gap, we proposed a framework aimed at standardizing and streamlining the implementation and operation of these use cases, which can help lower overall costs. To evaluate our framework, we created a practical implementation as proof of concept using SAP's ERP platform, one of the leading solutions in the market. We applied the framework to implement real-world Generative AI use cases, showing that our proposed solution is both practical and effective. Figure 1 provides an overview of our methodology, showing our iterative process model in the outer circle and the methods we applied in the inner circle.

To address the identified gaps, we utilized well-known software engineering methods [11]. For deriving requirements from these gaps, we applied established techniques [12] in requirement analysis. To convert these requirements into solution concepts, we used common practices [13] related to software design. Then, to implement these concepts, we followed standard software development procedures [13]. These methodologies are well-established and widely accepted in computer science, so there's no need to go into further detail here.

To ensure our findings and conclusions are verifiable, we employed the design science research methodology [14], which offers a solid framework for validating both processes and results. Additionally, in line with the taxonomy of information systems theory [15], our work is based on the design and action domain, which provides the theoretical base for our efforts in this paper. Together, these methodologies enhance the rigor and reliability of our research and findings.

IV. IDENTIFIED REQUIREMENTS

The Generative AI use cases which we analyzed were originated from the ERP core business processes idea to market, source to pay, plan to fulfill, lead to cash, recruit to retire, acquire to decommission, governance and finance. Thus, all segments of ERP modules were considered. As mentioned in the methodology section we applied requirement engineering techniques to drove the following functional requirements from the investigated Generative AI use cases:

1. Content Generation: Generative AI can help in content creation within ERP systems by automating the production of various text-based materials. This includes generating articles for internal communications, crafting job descriptions that accurately reflect role requirements, developing tailored interview questions, creating media content for marketing purposes, drafting emails, and producing detailed product descriptions. Generative AI can generate coherent, contextually appropriate content based on provided themes or keywords, significantly reducing the time and effort required for these tasks.

2. Question-Answering: This use case involves developing intelligent systems capable of understanding and responding to user queries within the ERP environment. By analyzing vast amounts of structured and unstructured data, including product documentation, financial reports, and operational guidelines, the Generative AI can provide accurate, context-aware answers to user questions. This capability can greatly enhance user support, reduce the load on human resources, and improve overall system usability.

3. Conversation Agents: Implementing Generative AI within ERP systems can provide users with interactive, intelligent assistance. These agents, in the form of chatbots or virtual assistants, can guide users through complex processes, explain and help resolve errors, offer customer support, and recommend actions based on the user's context and historical data. This can significantly improve user experience, reduce training time, and increase overall system efficiency.

4. Text Summarization: Generative AI can be employed to create concise, informative summaries of various text-based elements within ERP systems. This includes condensing lengthy instructions, dashboard data, error messages, or configuration information into easily digestible formats. By presenting key information in a summarized form, users can quickly grasp essential concepts and make informed decisions, enhancing productivity and reducing information overload.

5. Translation: In today's globalized business environment, ERP systems often need to support multiple languages. Generative AI can facilitate real-time, context-aware translation of text across different languages, maintaining the original meaning and nuances. This capability is crucial for supporting regional requirements, ensuring effective communication across global teams, and enabling seamless operation of ERP systems in multilingual environments.

6. Sentiment Analysis: By analyzing the tone and emotional content of text data, Generative AI can provide valuable insights into customer sentiment, employee feedback, and market perceptions. This capability can be applied to various text sources within ERP systems, such as customer reviews, support tickets, social media mentions, and internal communications. The resulting sentiment classifications (positive, negative, or neutral) can inform decision-making processes, drive customer satisfaction initiatives, and guide strategic planning.

7. Text Classification: Generative AI can automatically categorize incoming text data into predefined groups, streamlining various ERP processes. For instance, it can sort support tickets by urgency or department, classify resumes based on job requirements, categorize incoming invoices by vendor, or segment leads from various sources like emails, social media, and web forms. This automated classification can significantly improve data organization, routing, and processing efficiency within ERP systems.

8. Code Generation: For ERP system customization and development, Generative AI can assist in code generation. Given natural language descriptions of desired functionality, the Generative AI can produce code snippets or even complete modules in various programming languages. Additionally, it can provide intelligent code completion suggestions, enhancing developer productivity and reducing the likelihood of coding errors.

9. Data Augmentation: In the context of ERP systems, Generative AI can create synthetic data samples to enhance machine learning model training. This is particularly useful for scenarios where real-world data may be limited or sensitive. The AI can generate realistic, varied data sets that reflect the complexities of business operations, improving the performance and generalization capabilities of predictive models within the ERP system. Additionally, it can generate diverse test cases for evaluating code quality and system robustness.

10. Information Extraction: Generative AI can be employed to automatically extract structured information from unstructured text within ERP systems. This includes identifying and extracting key data points such as names, dates, addresses, financial figures, and other relevant information from documents like invoices, delivery notes, or sales orders. This capability can significantly reduce manual data entry, improve data accuracy, and accelerate document processing workflows.

11. Personalization: By analyzing user behavior, preferences, and historical data within the ERP system, Generative AI can create personalized experiences for individual users. This may include generating tailored content, providing personalized recommendations for actions or decisions, or customizing user interfaces. In the context of ERP, this could manifest as personalized training materials, customized dashboards, or AI-driven recommendations for sourcing, inventory management, or financial planning.

12. Creative Applications: Generative AI can support creative processes within ERP-related business functions. This includes assisting with brainstorming sessions, generating product names or advertising slogans, and even creating visual or audio content for new product ideas. By providing AI-generated creative inputs, businesses can stimulate innovation, enhance marketing efforts, and explore new product development avenues more efficiently.

Additionally, we identified the following technical requirements for realizing and operating Generative AI applications in context of ERP software:

[REQ-01] Vendor Diversity: It is required to be open for diverse Generative AI model vendors so that ERP applications have the choice and not locked into a specific model. This requirement emphasizes the importance of maintaining openness and flexibility in the selection of Generative AI model vendors within ERP applications. It's a crucial consideration that has significant implications for the long-term viability, performance, and cost-effectiveness of AI-enhanced ERP systems.

[REQ-02] Built-in Generative AI: This requirement emphasizes the importance of integrating Generative AI capabilities directly into the ERP system's business processes. The AI features should be contextually embedded, ensuring they are available to the right users, at the appropriate points in workflows, and at the optimal times. This integration should feel natural and intuitive, enhancing user productivity without disrupting established processes.

[REQ-03] Standardized Development: To facilitate efficient development and maintenance, the ERP system should provide a uniform programming model for Generative AI features. This standardization allows developers to work with different AI technologies using a consistent set of tools, APIs, and methodologies. This approach reduces the learning curve, improves code maintainability, and enables easier integration of new AI technologies as they emerge.

[REQ-04] Standardized Operations: Similar to the development standardization, this requirement calls for a uniform approach to operating Generative AI features within the ERP system. Regardless of the underlying AI technology, customers should interact with a consistent interface for managing AI functionalities. This standardization simplifies system administration, reduces training needs, and ensures a coherent user experience across different AI-powered features.

[REQ-05] Model Adoption: This requirement addresses the need for tools and mechanisms to adapt and fine-tune Generative AI models for specific ERP use cases. This includes facilities for creating effective prompts, incorporating domain-specific embeddings, and adjusting model parameters. These capabilities allow organizations to tailor the Generative AI's performance to their unique business contexts and requirements.

[REQ-06] Legal Compliance: Given the sensitive nature of data processed in ERP systems, this requirement is crucial. The integration of Generative AI must adhere to data privacy and protection regulations. This includes implementing robust consent management, controls for automated decision-making, comprehensive logging of data access, and mechanisms for legal auditing. Ensuring compliance protects both the organization and its customers from legal and reputational risks.

[REQ-07] Optimized Lookup: To ensure rapid performance of Generative AI features, especially those relying on embeddings, the ERP system should support efficient vector search engines. This optimization is critical for maintaining responsive AI interactions within the ERP environment, particularly when dealing with large volumes of data or complex queries.

[REQ-08] Validation: This requirement calls for mechanisms to validate both the inputs to and outputs from Generative AI models. Validation should cover aspects such as business correctness, ensuring non-discriminatory responses, and filtering out inappropriate content like hate speech. This safeguard is essential for maintaining the integrity and reliability of AI-generated content within the ERP system.

[REQ-09] Response Time: To maintain user engagement and system usability, this requirement specifies performance targets for Generative AI interactions. The defined response times (typically 150ms for instant feedback, 1000ms for simple interactions, and 3000ms for complex tasks) ensure that AI features remain responsive and do not impede workflow efficiency. Meeting these targets may involve optimizations in model selection, hardware allocation, and network infrastructure.

[REQ-10] Configuration: This requirement addresses the need for flexible configuration of Generative AI features. Users should be able to adjust parameters such as token limits, temperature settings, and hardware resource allocation. Importantly, these configurations should persist through system updates and upgrades, ensuring consistent operation and preventing unexpected changes to Generative AI behavior.

[REQ-11] Extensibility: To accommodate diverse business needs, the ERP system should provide mechanisms for extending and customizing Generative AI applications. This extensibility allows organizations to tailor Generative AI features to their specific requirements. Like configurations,

these extensions should be preserved during system updates, maintaining the integrity of customized functionality.

[REQ-12] Localization: This requirement ensures that Generative AI features can operate effectively across different languages and regional contexts. Implementation of functional localizations is crucial for global organizations, enabling Generative AI capabilities to provide consistent performance and accuracy regardless of the user's language or location.

V. SOLUTION ARCHITECTURE

This section outlines concepts to address the requirements discussed previously. We utilized established software engineering techniques for software design, including Unified Modeling Language (UML), to translate requirements into software design, as detailed in the methodology section.

Insights from current use cases indicate that Generative AI models, such as large language models (LLMs), have significant potential across various applications, albeit with possible adaptation needed for optimal performance in particular tasks or domains. To facilitate this, we can apply techniques such as prompt engineering and embeddings. Prompt engineering involves formulating specific tasks or queries in natural language, guiding the Generative AI model to produce more accurate and relevant responses. By designing prompts carefully, we can effectively direct the model to prioritize the necessary aspects of a task, thus enhancing its performance. Embeddings leverage external knowledge to significantly enhance a model's ability to incorporate domain-specific knowledge. They encode information in a format that the Generative AI model can accurately process. By integrating domain-specific embeddings (e.g., ideal code examples, product documentation), we enrich the model's comprehension of a domain and equip it with valuable references for generating more precise and context-relevant outputs. Although fine-tuning through training could further optimize Generative AI models, however, this paper will not delve into data science topics. This is also true for data science techniques for prompt engineering.

Employing these well-established techniques allows us to adapt Generative AI models to a broad array of tasks and domains, maximizing their capabilities to meet our functional demands. These model adoption methods represent a novel aspect in application development and are thus incorporated into the proposed solution architecture as demonstrated in Figure 2.

The ERP system in Figure 2 contains the Generative AI Application which consumes capabilities of the Generative AI model like content generation or text summarization. The DevOps Framework standardizes the development and operational tasks of Generative AI Applications, encompassing model configuration and deployment. The primary design-time artifact is the Intelligent Scenario, encapsulating all necessary development entities for a Generative AI use case. The Prompt Executor employs pre-defined Prompt Templates for specific use cases during the design phase and populates parameters with concrete values during runtime. The following text depicts an illustrative Prompt Template for an internal job description use case:

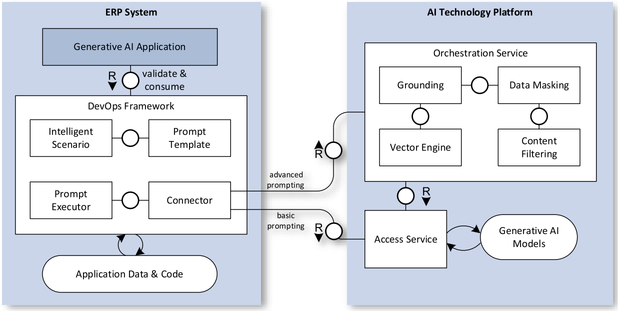


FIGURE 2. **Solution architecture.**

- You are an assistant designed to generate appealing job descriptions for an international company named ***[company_name]****.*

- Users will input structured data for a job position. You should generate an html-formatted job description.

- Avoid bias based on physical appearance, ethnicity, or race. Replace inappropriate language with inclusive language, politely refuse results, if that is not possible.

- Provide the response in ***[language]****.*

- Generate an internal job description for ***[job title]****. The candidate shall have* ***[Skill-01]****,* ***[Skill-02]*** *and* ***[Skill-03]****.*

- Hiring Manager is ***[manager]*** *and recruiter is* ***[recruiter]****.*

- Location is ***[location]*** *and start of work is* ***[start-date]****.*

The above template includes placeholders indicated by square brackets. These Prompt Templates are defined and stored during design, while the Generative AI Application supplies parameter values during runtime when the Generative AI feature is utilized. The interaction relies on an inference API provided by the Prompt Executor, which applies text functionality and substitutes parameters with specific values, sending the populated prompt through the Connector to the Generative AI model, and relaying the response as text to the Generative AI Application. Such straightforward use cases we refer to as basic prompting. Decoupling Prompt Templates from application code and storing them in the ERP system optimizes lifecycle management. For instance, Prompt Templates can be updated to newer Generative AI model versions without altering the application code.

To alleviate stress on the transactional ERP system and maintain acceptable response times for business processes, Generative AI model services with substantial hardware needs are deployed alongside the ERP system on an AI Technology Platform. These platforms are offered by companies like Amazon, Google, Microsoft, SAP, and startups. Depending on use case requirements, embeddings may need to be appended to the prompt. These could be straightforward examples (e.g., exemplary job descriptions) or intricate contexts (e.g., PDF documents containing job descriptions for various categories) utilizing Vector Engines and Grounding. Vector Engines optimize Generative AI model performance by processing data's semantic representations (vectors), facilitating efficient information retrieval. Unlike traditional keyword-focused search engines, vector search employs mathematical methods to represent words, phrases, sentences, or documents in a high-dimensional space. Semantically similar entries have analogous vector representations, making Vector Engines particularly useful for information retrieval tasks, such as document retrieval, where comprehending the query's context and meaning is crucial for discovering pertinent results. These results are incorporated into the prompt to improve Generative AI model responses. Grounding ensures AI model outputs remain contextually relevant and anchored in the specific data domains of the ERP system. Prompts may contain confidential information (e.g., manager and recruiter in the job description example) not intended to be shared with Generative AI models. To meet this requirement, sensitive data is masked before transmitted to the external Generative AI model, using techniques like anonymization or pseudonymization. The returned response is de-masked (e.g., pseudonymization parameters replaced with original values) and sent to the application. Content Filtering evaluates requests to and responses from Generative AI to prevent inappropriate content generation (e.g., illegal, hateful, or violent). With advanced prompting we refer to use cases incorporating one or multiple features of the Orchestration Service. The Access Service oversees connectivity to various Generative AI models and directs application requests accordingly, invoked directly in basic prompting or via the Orchestration Service for advanced prompting. The interplay of the building blocks of the solution architecture are shown in Figure 3 for the advanced prompting scenario.

The Generative AI Application calls the Prompt Executor with concrete values for the underlying Prompt Template. The Prompt Executor retrieves the Prompt Template, instantiates it with parameter values, and calls the Orchestration Service. This service includes embeddings in the prompt, applies data masking and input filtering before handing the prompt to the Access Service, which calls the Generative AI model and returns the response. The Orchestration Service then applies output filtering, de-masking, and provides the response back to the Prompt Executor, which sends it to the Generative AI Application. A business validation (e.g., check product codes) is performed before the results are incorporated into ERP business processes and user interfaces. This methodology allows Generative AI applications to trigger and coordinate multiple Generative AI model calls to tackle complex tasks.

The solution architecture facilitates the provisioning of Generative AI models allowing prompting and embeddings by the Generative AI Application. Therefore, all features provided by Generative AI models can be the utilized so that functional requirements 1-12 from previous section can be fulfilled. In the next section we will introduce a concrete realization of the solution architecture to illustrate conceptual details. Nevertheless, we consider already the technical requirements form the previous section which are typically ensured by the solution architecture.

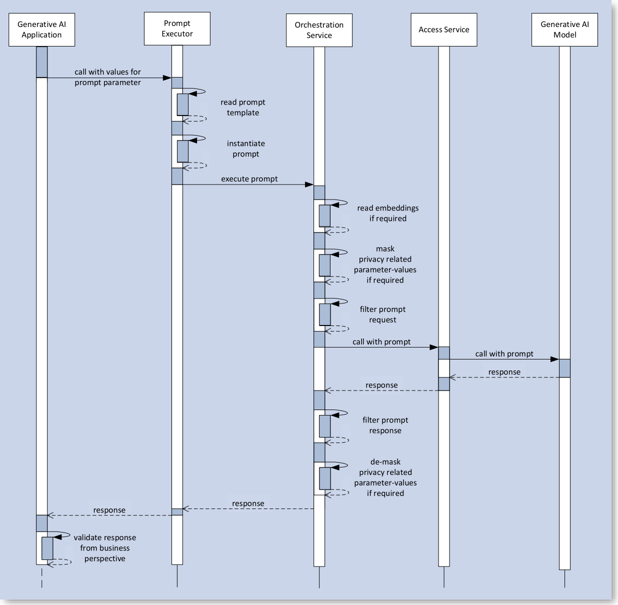

Generative AI Application
Prompt Executor
Orchestration Service
Access Service
Generative AI Model
call with values for prompt parameter
read prompt template
instantiate prompt
execute prompt
read embeddings if required
mask privacy related parameter-values if required
filter prompt request
call with prompt
call with prompt
response
response
filter prompt response
de-mask privacy related parameter-values if required
response
response
validate response from business perspective

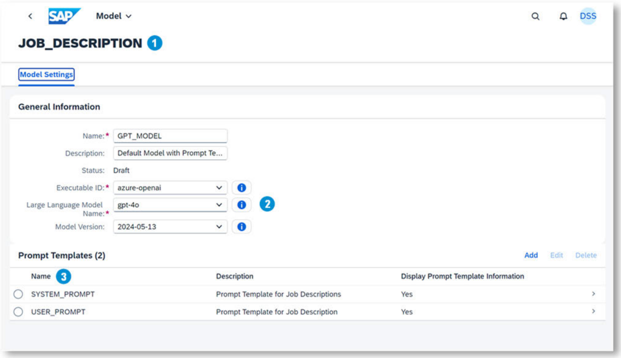
FIGURE 3. Processing advanced prompting.

Vendor diversity [REQ-01] is achieved by allowing the Generative AI Application to define Prompt Templates for various Generative AI models, connecting to the model via the Access Service at runtime. Built-in Generative AI [REQ-02] is facilitated through the Prompt Executor's inference API, integrating generative capabilities deep into ERP business processes and interfaces. The Intelligent Scenario, along with the Prompt Template for decoupling code from prompts and a standardized inference API, streamlines the programming model for all Generative AI Applications [REQ-03]. The DevOps Framework abstracts underlying Generative AI models, providing standardized operational steps like deployment and model activation [REQ-04]. The architecture supports prompting, embeddings, and model adaptation [REQ-05]. ERP systems provide concepts and tools [16] for ensuring data privacy and data protection regulation, consent management, automated decision making, read access logging and legal auditing for narrow AI models which can be entirely reused for Generative AI models too [REQ-06]. Specific is avoiding inadequate (e.g., racial or discriminating content) requests and responses which is mitigated with input and output Content Filtering. Furthermore, the exchange of personal and sensitive data with the Generative AI model is counter measured with foreseen Data Masking functionality (e.g., manager and recruiter are masked in the Prompt Template Hiring Manager is [manager] and recruiter is [recruiter]). Incorporating Grounding and Vector Engine satisfies the technical requirement for optimized lookup [REQ-07]. Validation [REQ-08] employs a two-phase approach: Content Filtering covers security (e.g., SQL injection) and legal (e.g., hate speech) aspects, while the Generative AI Application performs additional business validation (e.g., correct invoice type). For Generative AI models supporting streaming, output is generated in segments and processed through the Orchestration Service, aiding applications needing real-time model interaction for better response times [REQ-09]. Configuration settings, like streaming, token count, and temperature, are managed via the DevOps Framework, allowing adjustments based on specific ERP customer needs [REQ-10]. Extensibility [REQ-11] is facilitated by pre-defined enhancement points as parameters in the Prompt Template for customer use (e.g., The candidate shall have [Skill-01] and [Skill-02][Enhancement Point].; the parameter ‘‘Enhancement Point’’ can be used by customers to extend the Prompt Template with additional sentences). For localization [REQ-12], ERP systems provide solutions that can be leveraged [16]. Specific is the translation of the responses provided by the Generative AI models. In simple cases this could be resolved within the Prompt Template (e.g., provide the response in [language]; the parameter language correspondence to the login language of the user) but may require using the translation service of the underlying ERP system if the translation quality is inadequate.

VI. EVALUATION

To demonstrate the practical feasibility of the proposed solution architecture for implementing Generative AI applications in context of ERP, we depict a concrete framework based on these concepts. A practical implementation needs to be built on an existing ERP platform. We chose SAP ERP for this purpose due to its widespread use and market leadership.

Given the expansive functionality of the framework integrating Generative AI into ERP systems, we selectively highlight in this section key components. Specifically, we focus on the Intelligent Scenario, the Prompt Template, and the Orchestration Service, as depicted in Figure 2.



JOB_DESCRIPTION

General Information

Name:* GPT_MODEL
Description: Default Model with Prompt Te...
Status: Draft
Executable ID:* azure-openai
Large Language Model Name:* gpt-4o
Model Version: 2024-05-13

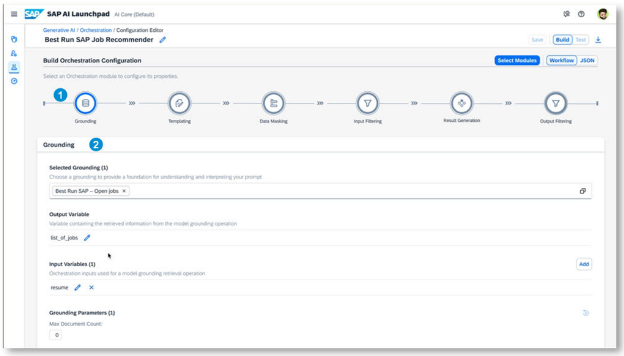
Prompt Templates (2)

	Name	Description	Display Prompt Template Information
○	SYSTEM_PROMPT	Prompt Template for Job Descriptions	Yes
○	USER_PROMPT	Prompt Template for Job Description	Yes

FIGURE 4. Intelligent scenario.

Once the prompt engineering phase is concluded, we can initiate the development of a Generative AI application. This process begins with defining the Intelligent Scenario, as shown in Figure 4 (section 1). This scenario incorporates the Prompt Templates, which are the outcome of the prompt engineering phase, illustrated in Figure 4 (section 3). Additionally, the Intelligent Scenario includes important metadata, such as the specific Generative AI model being utilized and its corresponding version, as illustrated in Figure 4 (section 2). By establishing a clear understanding of these foundational elements, we can efficiently proceed with the application development.

In situations that require advanced prompting, the Orchestration Service comes into play. This service enables development with a suite of functions frequently necessary for Generative AI applications within ERP systems. In a basic scenario, various modules available in the Orchestration Service can be linked together to form a pipeline. This pipeline can then be executed with a single API call. Within this pipeline, the output from one module serves as the input for the subsequent module. The sequence in which these modules are executed is centrally managed within the Orchestration Service. Nevertheless, users have the flexibility to configure the specifics for each module and to exclude optional modules by providing an orchestration configuration in JSON format in the request body. This ability to customize the orchestration flow allows for adaptability and precision in handling various AI-driven tasks.



Best Run SAP Job Recommender

Build Orchestration Configuration

Select an Orchestration module to configure its properties.

Grounding

Selected Grounding (1)
Choose a grounding to provide a foundation for understanding and interpreting your prompt
Best Run SAP – Open jobs ×

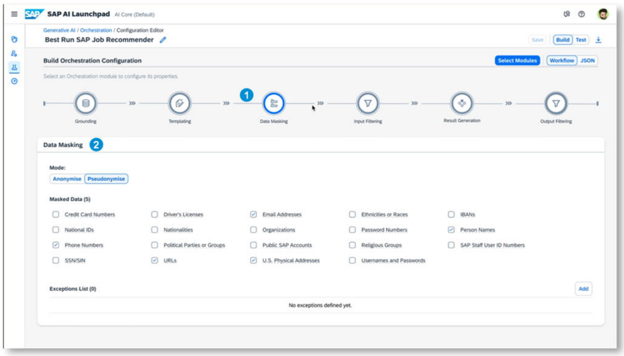
Output Variable
Variable containing the retrieved information from the model grounding operation
list_of_jobs

Input Variables (1)
Orchestration inputs used for a model grounding retrieval operation
resume

Grounding Parameters (1)
Max Document Count:
0

FIGURE 5. Grounding.

The Grounding module as shown in Figure 5 (section 1) of the Orchestration Service plays a crucial role in enhancing Generative AI processes by integrating external, contextually relevant, domain-specific, or real-time data as illustrated in Figure 5 (section 2). This integration enriches the natural language processing capabilities of pre-trained models, which typically rely on general datasets. The Grounding module is specifically designed to manage data-related tasks such as grounding and retrieval, and it utilizes vector databases to achieve this. The pipeline API is routed through the runtime of the Orchestration Service and incorporates vector stores to facilitate specialized data retrieval through these databases. It effectively grounds the retrieval process by using external and contextually relevant data. By combining Generative AI capabilities with the ability to leverage real-time and precise data, Grounding significantly enhances decision-making and business operations. This is particularly beneficial for developing AI-driven solutions tailored to specific business needs.



Best Run SAP Job Recommender

Build Orchestration Configuration

Select an Orchestration module to configure its properties.

Data Masking

Mode:
Anonymise [✓ Pseudonymise]

Masked Data (5)

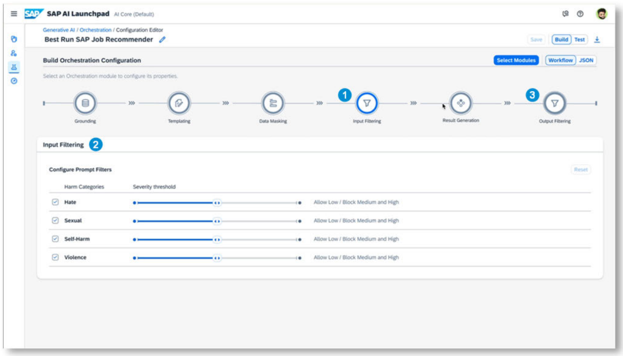
☐ Credit Card Numbers	☐ Driver's Licenses	☑ Email Addresses	☐ Ethnicities or Races	☐ IBANs
☐ National IDs	☐ Nationalities	☐ Organizations	☐ Password Numbers	☑ Person Names
☑ Phone Numbers	☐ Political Parties or Groups	☐ Public SAP Accounts	☐ Religious Groups	☐ SAP Staff User ID Numbers
☐ SSN/SIN	☑ URLs	☑ U.S. Physical Addresses	☐ Usernames and Passwords	

Exceptions List (0)

No exceptions defined yet.

FIGURE 6. Masking.

The Data Masking module, which is an optional component as shown in Figure 6 (section 1), serves an important function of safeguarding privacy. This module anonymizes or pseudonymizes personally identifiable information within the input for selected entities as referred in Figure 6 (section 2). In the anonymization process, personally identifiable information from specific categories is replaced with a masked entity placeholder, ensuring that the data cannot be reverted to its original form since no information about the original data is retained. Conversely, pseudonymization involves replacing such sensitive information with a masked entity ID placeholder. Unlike anonymized data, pseudonymized data retains the capability to be reverted, or unmasked, in the response. This distinction highlights the flexibility provided by the Data Masking module to accommodate varying levels of data privacy and security requirements, based on the specific needs of the ERP application.



Best Run SAP Job Recommender

Build Orchestration Configuration

Select an Orchestration module to configure its properties.

Input Filtering

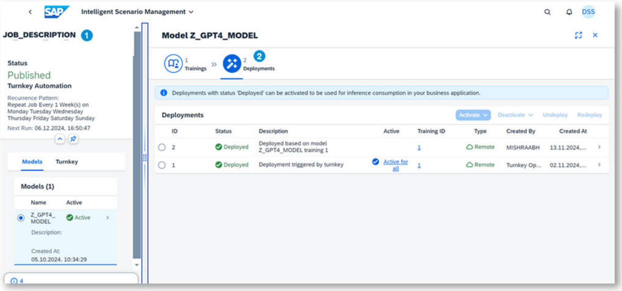
Configure Prompt Filters

	Harm Categories	Severity threshold
☑	Hate	Allow Low / Block Medium and High
☑	Sexual	Allow Low / Block Medium and High
☑	Self-Harm	Allow Low / Block Medium and High
☑	Violence	Allow Low / Block Medium and High

FIGURE 7. Content filtering.

The Input Filtering module allows users to determine the type of content that is sent to the Generative AI model as illustrated in Figure 7 (section 1). This module includes a content safety classification service that identifies four specific content categories as outlined in Figure 7 (section 2): Hate, Violence, Sexual, and Self-Harm. Hate and fairness harms refer to any content that targets or uses discriminatory language against individuals or groups based on attributes such as race, ethnicity, nationality, gender identity, sexual orientation, religion, physical appearance, body size, disability status, or involves harassment and bullying. The Sexual category covers language related to anatomical organs and genitals, romantic relationships, and sexual acts, including vulgar content, prostitution, nudity, pornography, abuse, child exploitation, and forced sexual acts. The Violence category deals with language describing physical harm, damage, or intent to kill, including discussions about weapons, bullying, stalking, and violent extremism. Finally, the Self-Harm category includes language that pertains to actions intended to harm oneself, such as those related to eating disorders or bullying.

It's important to note that a text can belong to multiple categories (e.g., a text may be classified as both Hate and Violence). Each identified category is presented with a severity level rating of 0, 2, 4, or 6, with higher values indicating more severe content. Similarly, the Output Filtering module allows users to decide the type of content received from the Generative AI model as shown in Figure 7 (section 3). Notably, some Generative AI models already have embedded content filters included in their standard features, meaning that some content may be filtered out regardless of the specific settings configured through the Output Filtering module. This dual-layer filtering process ensures that both input to and output from the AI models are carefully monitored and maintained within acceptable safety standards.



JOB_DESCRIPTION

Status
Published
Turnkey Automation
Recurrence Pattern:
Repeat Job Every 1 Week(s) on
Monday Tuesday Wednesday
Thursday Friday Saturday Sunday
Next Run: 06.12.2024, 16:50:47

Models (1)

Name	Active
● Z_GPT4_MODEL Description: Created At: 05.10.2024, 10:34:29	Active

Model Z_GPT4_MODEL

Deployments with status 'Deployed' can be activated to be used for inference consumption in your business application.

Deployments

	ID	Status	Description	Active	Training ID	Type	Created By	Created At
○	2	Deployed	Deployed based on model Z_GPT4_MODEL training 1		1	Remote	MISHRAABH	13.11.2024,...
○	1	Deployed	Deployment triggered by turnkey	Active for all	1	Remote	Turnkey Op...	02.11.2024,...

FIGURE 8. Deployment.

Once the development and testing phases are successfully completed, the Generative AI application is ready to be released for operational use. The DevOps Framework plays a crucial role in this process, providing customers with the necessary tools and functionalities to effectively manage the Generative AI application as shown in Figure 8. These tools allow users to perform for an Intelligent Scenario (section 1) tasks such as adjusting the configuration parameters of the Generative AI model, fine-tuning the settings for Content Filtering, and deploying (section 2) as well as activating the Generative AI capability along with the corresponding model. Once these activation steps are complete, the Generative

AI application becomes fully operational and accessible for end-users. In the following section, we will present an example to illustrate how such a Generative AI application can be implemented and utilized based on the suggested framework.

To evaluate the proposed solution architecture for implementing Generative AI into ERP software, a framework was introduced previously. This proved that the theoretical concept can be practically implemented. The next step is to show that this framework is applicable for developing Generative AI applications in context of ERP. To do this, we depict an example of a Generative AI application related to ERP tax jurisdictions and subsequently discuss additional use cases.

Configuring ERP systems to manage United States tax jurisdictions presents several significant business challenges due to the complexity and diversity of tax laws in the region. In the United States, sales tax is not standardized at the federal level; instead, it is determined by a patchwork of rules across 45 states, the district of Columbia, Puerto Rico, and numerous local counties and cities. This results in more than 13,000 distinct sales tax jurisdictions. One of the primary challenges is managing the complexity and volume of these tax jurisdictions. Each one can have its own tax rates and rules, requiring ERP systems to be highly adaptable and configurable. This complexity is further compounded by the varied tax structures across jurisdictions. Some states have single statewide rates, while others permit local governments to impose additional taxes, creating intricate combinations of state, county, and city taxes. For example, New York City imposes a 4% state tax, a 4.5% city tax, and a 0.375% charge within the metropolitan commuter transportation district. Compliance is another significant challenge, as tax rates and regulations can change frequently. ERP systems must be continuously updated to reflect the latest tax laws across all those jurisdictions. This requires businesses to monitor legislative changes and update their systems promptly to maintain compliance. Granularity and specificity in tax applicability add another layer of difficulty. Taxes can vary not only by jurisdiction but down to specific locations or services. For instance, software-as-a-service subscriptions might be non-taxable at a venue in Orlando but subject to a 6% tax at the headquarters in Pennsylvania and an 8.875% tax rate in Manhattan. This necessitates detailed ERP configurations to ensure accurate tax calculations and reporting. Additionally, businesses often operate across multiple jurisdictions, requiring ERP systems to manage and report taxes accurately across different areas and rates. Integration of tax data with other business processes, such as sales, billing, and financial reporting, is essential, demanding robust and scalable ERP solutions. Lastly, the complexity of configuring and updating ERP systems requires knowledgeable users. Providing comprehensive training and ensuring a user-friendly system interface are crucial to effectively managing the intricate tax requirements. Overall, businesses must navigate these complexities to achieve compliance and operational efficiency in a highly fragmented tax environment.

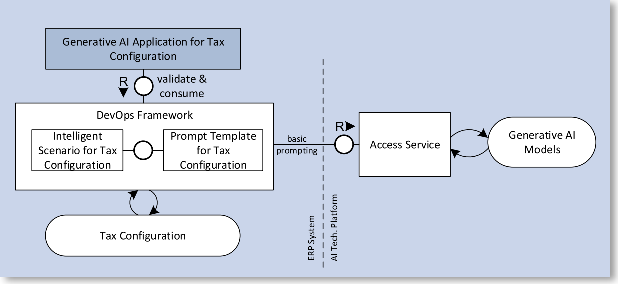
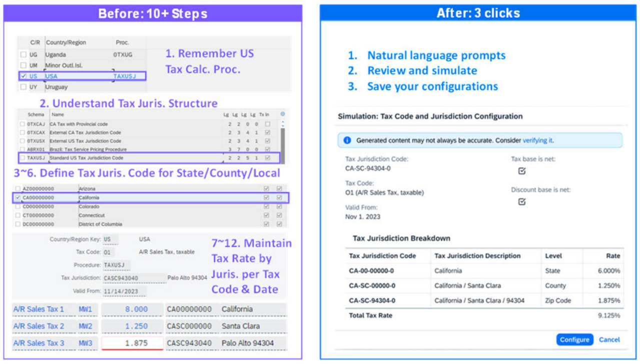


FIGURE 9. Utilizing the solution concept and framework for tax configuration.

As depicted in Figure 9 the proposed ERP solution for managing United Sates tax jurisdictions leverages Generative AI technology to simplify the complex process of tax configuration. At the heart of this solution is the Generative AI capability that autonomously identifies missing compliance configurations and generates proposals to fill these gaps. This significantly reduces the manual workload for tax accountants and minimizes the risk of errors in maintaining tax data. Users can interact with the Generative AI application using natural language prompts within the configuration simulator, which facilitates the input of address details and allows for the simulation and breakdown of tax jurisdictions. By entering a street address or ZIP code, the Generative AI application retrieves detailed jurisdictional information across multiple levels, including state, county, locality, and special tax levels. Users can review this information to ensure its accuracy and relevance to their needs. Once the Generative AI application provides the necessary data, users manually input the corresponding tax rates and set the validity periods for each tax level. The system mandates that updates be made in chronological order, which prevents configuration errors and ensures that changes align with existing tax schedules. This approach ensures that new entries do not result in overlapping periods, thereby preserving the integrity of the tax data. A key aspect of this solution is its verification protocol. Given that AI-generated content can occasionally be inaccurate, users are advised to verify the AI-generated address and tax information against authoritative sources, such as official publications from state tax departments. This step is crucial for ensuring that the ERP system remains compliant with current tax laws and regulations.

As shown in Figure 10 the AI-assisted ERP solution streamlines the management of United States tax jurisdictions, providing a user-friendly platform that enhances efficiency and ensures compliance with tax regulations. By combining Generative AI capabilities with rigorous verification processes, the solution enables tax professionals to navigate the complexities of tax configurations with greater ease and accuracy.


Before: 10+ Steps
1. Remember US Tax Calc. Proc.
2. Understand Tax Juris. Structure
3~6. Define Tax Juris. Code for State/County/Local
7~12. Maintain Tax Rate by Juris. per Tax Code & Date
After: 3 clicks

	C/R	Country/Region	Proc.
□	UG	Uganda	0TXUG
□	UM	Minor Outl.Isl.	
☑	US	USA	TAXUSJ
□	UY	Uruguay	

	Schema	Name	Lg	Lg	Lg	Lg	Tx In
□	0TXCAJ	CA Tax with Provincial code	2	2	0	0	□
□	0TXCAX	External CA Tax Jurisdiction Code	2	3	4	1	☑
□	0TXUSX	External US Tax Jurisdiction Code	2	3	4	1	☑
□	ABRX01	Brazil: Tax Service Pricing Procedure	3	7	0	0	☑
□	TAXUSJ	Standard US Tax Jurisdiction Code	2	2	5	1	☑

□	AZ00000000	Arizona	☑	☑
☑	CA00000000	California	☑	☑
□	CO00000000	Colorado	☑	☑
□	CT00000000	Connecticut	☑	☑
□	DC00000000	District of Columbia	☑	☑

Country/Region Key: US USA
Tax Code: 01 A/R Sales Tax, taxable
Procedure: TAXUSJ
Tax Jurisdiction: CASC943040 Palo Alto 94304
Valid From: 11/14/2023

A/R Sales Tax 1	MW1	8.000	CA00000000	California
A/R Sales Tax 2	MW2	1.250	CASC000000	Santa Clara
A/R Sales Tax 3	MW3	1.875	CASC943040	Palo Alto 94304

1. Natural language prompts
2. Review and simulate
3. Save your configurations

Simulation: Tax Code and Jurisdiction Configuration

Generated content may not always be accurate. Consider verifying it.

Tax Jurisdiction Code:
CA-SC-94304-0

Tax base is net: ☑

Tax Code:
O1 (A/R Sales Tax, taxable)

Discount base is net: ☑

Valid From:
Nov 1. 2023

Tax Jurisdiction Breakdown

Tax Jurisdiction Code	Tax Jurisdiction Description	Level	Rate
CA-00-00000-0	California	State	6.000%
CA-SC-00000-0	California / Santa Clara	County	1.250%
CA-SC-94304-0	California / Santa Clara / 94304	Zip Code	1.875%
Total Tax Rate			9.125%

FIGURE 10. User interaction with and without applying Generative AI.

In conclusion, the intuitive interface simplifies maintaining tax jurisdictions, making it accessible for users with varying levels of expertise. This eases the onboarding process for businesses entering or operating in the United States, ensuring they can quickly establish compliance with local tax laws. By integrating directly with tax authority websites, the system enhances accuracy and reliability, automatically updating configurations to reflect the latest legal requirements. Additionally, the Generative AI capability proactively identifies and generates proposals for any missing compliance configurations, reducing manual oversight and error. According to customer feedback the solution significantly reduces the effort required to set up sales and use tax configurations within the United States, achieving a 50-90% reduction in time and resources. This efficiency not only lowers operational costs but also accelerates the time-to-market for businesses, allowing them to focus more on strategic activities rather than administrative tasks. Overall, the integration of ERP and Generative AI in managing tax jurisdictions offers substantial competitive advantages through increased accuracy, speed, and compliance.

We illustrated the tax jurisdictions scenario; however, the framework has been successfully applied to numerous Generative AI use cases [17], [18] so that they have become all part of the SAP's ERP product. This demonstrates the framework's effectiveness and efficiency for implementing Generative AI applications. Ten thousand SAP customers are daily consuming those Generative AI capabilities. Their feedback is that they significantly have saved costs and optimized the ERP business processes. Additionally, the framework itself has also been incorporated into SAP's ERP product [19], [20], allowing customers and partners to develop their own Generative AI applications. This proves its functional correctness and real-world applicability.

The implementation of the framework serves as evidence that the concepts work effectively in real-world conditions. All ERP products are implemented with different programming languages. However, from theoretical computer science, we know that all programming languages are turing-complete [21], meaning they have equivalent expressive power. Since the framework was successfully implemented using SAP's ABAP language, we can conclude that the framework can be realized with programming languages of other ERP solutions and used for implementing Generative AI applications. This makes the framework broadly applicable and valid for different ERP products. This is particularly important as no similar ERP AI solution could be identified as discussed in the literature review.

VII. CONCLUSION

The integration of Generative AI into ERP systems represents a paradigm shift in how businesses manage and optimize their operations. As demonstrated in this study, the proposed framework for embedding Generative AI within ERP software not only enhances functional capabilities across various ERP modules but also brings forth significant efficiency and productivity gains. By addressing the myriad challenges inherent in ERP systems - such as diverse, complex codebases and the need for compliance with industry-specific and regional requirements - the framework provides a structured and practical approach for developers, users, and organizations. Through the systematic requirement analysis, solution design, and real-world evaluation presented in this paper, we have shown that generative models like large language models (LLMs) can be seamlessly and effectively integrated into ERP workflows. Key functionalities such as content generation, text summarization, and conversation agents empower both technical and non-technical users to accomplish tasks through natural language interfaces. This democratizes access to Generative AI capabilities and fosters innovative solutions to traditional ERP challenges. Moreover, the ability to tailor Generative AI models for specific applications through techniques like prompt engineering and embeddings enhances the precision and relevance of generated outputs, highlighting the flexibility and adaptability of our framework. The integration of robust development and operations practices ensures that these AI-enhanced ERP systems remain streamlined, scalable, and secure, thereby reducing the total cost of ownership and simplifying long-term management.

The practical implementation and evaluation using SAP ERP, and generalization to ERP systems due to turing-completeness, affirm the feasibility and effectiveness of the framework. The successful application to diverse use cases, including the configuration of tax jurisdictions, underscores the real-world benefits of integrating Generative AI into ERP systems. Our research has established a comprehensive, adaptable, and validated framework for incorporating Generative AI into ERP software. This framework not only addresses current limitations but also sets the stage for future advancements in intelligent enterprise solutions. As Generative AI technology continues to evolve, the principles and methodologies outlined in this paper will serve as a foundation for ongoing innovation, ensuring that ERP systems remain at the forefront of digital transformation.

REFERENCES

[1] M. Chui, E. Hazan, R. Roberts, A. Singla, K. Smaje, A. Sukharevsky, L. Yee, and R. Zemmel. (2023). *The Economic Potential of Generative AI*. [Online]. Available: https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/the-economic-potential-of-generative-ai-the-next-productivity-frontier#introduction

[2] B. Kitchenham and S. M. Chartes, "Guidelines for performing systematic literature reviews in software engineering," EBSE, Softw. Eng. Group School Comput. Sci. Math., Keele Univ., Keele, U.K., Tech. Rep. EBSE-2007-01, 2007.

[3] Q. Lu, L. Zhu, X. Xu, Y. Liu, Z. Xing, and J. Whittle, "A taxonomy of foundation model based systems through the lens of software architecture," in *Proc. IEEE/ACM 3rd Int. Conf. AI Eng.-Softw. Eng. AI*, Apr. 2024, pp. 1–6.

[4] H. Kazem and D. Safaei, "Generative artificial intelligence for enterprise platforms," in *Proc. 32nd Eur. Conf. Inf. Syst. (ECIS)*, Jun. 2024, pp. 13–19.

[5] H. Mahmoud, H. M. Elbadawy, T. Ismail, and D. Mi, "A comprehensive review of generative AI applications in 6G," in *Proc. 6th Novel Intell. Lead. Emerg. Sci. Conf. (NILES)*, Oct. 2024, pp. 593–596.

[6] A. Faccia, F. Manni, V. Pandey, and L. P. L. Cavaliere, "Blazing a new trail in ERP integration with NLP and generative AI through APIs: A fraud examination perspective," in *Proc. 8th Int. Conf. Inf. Syst. Eng.*, Dec. 2023, pp. 177–184.

[7] N. Rane, S. Choudhary, and J. Rane, "Intelligent manufacturing through generative artificial intelligence, such as ChatGPT or bard," Univ. Mumbai, Maharashtra, India, Tech. Rep. 4681747, Jan. 2024, doi: 10.2139/SSRN.4681747.

[8] P. Arora, L. Desu, A. Kumar, R. Kumar, and A. Marinescu, "Enhancing profitability through AI-optimized accounts receivable: Reducing cash conversion cycles," in *Proc. Int. Conf. Electr., Comput. Energy Technol. (ICECET)*, Jul. 2024, pp. 1–5.

[9] H. Aguinis, J. R. Beltran, and A. Cope, "How to use generative AI as a human resource management assistant," *Organizational Dyn.*, vol. 53, no. 1, Jan. 2024, Art. no. 101029.

[10] G. Anshumaan and D. Sharma, "Generative AI for software test modelling with a focus on ERP software," in *Proc. Int. Conf. Adv. Comput., Commun. Inf. Technol. (ICAICCIT)*, Nov. 2023, pp. 187–193.

[11] M. Broy and M. Kuhrmann, *Einführung in Die Softwaretechnik*. Berlin, Switzerland: Springer, 2021.

[12] J. Dick, E. Hul, and K. Jackson, *Requirements Engineering*. Cham, Switzerland: Springer, 2017.

[13] J. Dooley and V. Kazakova, *Software Development, Design, and Coding*. Cham, Switzerland: Springer, 2024.

[14] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *J. Manage. Inf. Syst.*, vol. 24, no. 3, pp. 45–77, Dec. 2007.

[15] Gregor, "The nature of theory in information systems," *MIS Quart.*, vol. 30, no. 3, pp. 611–642, 2006.

[16] S. Sarferaz, *Compendium on Enterprise Resource Planning*. Cham, Switzerland: Springer, 2022.

[17] SAP SE. Product Roadmap. *Artificial Intelligence*. Accessed: Feb. 1, 2025. [Online]. Available: https://roadmaps.sap.com/

[18] *SAP SE. Discovery Center, SAP Bus. AI Features*. Accessed: Feb. 1, 2025. [Online]. Available: https://discovery-center.cloud.sap/ai-catalog/

[19] SAP SE. Product Documentation. *Intelligent Scenario Lifecycle Management*. Accessed: Feb. 1, 2025. [Online]. Available: https://help.sap.com

[20] SAP SE. Product Documentation. *Generative AI Hub in SAP AI Core*. Accessed: Feb. 1, 2025. [Online]. Available: https://help.sap.com

[21] N. S. Yanofsky, *Theoretical Computer Science for the Working Category Theorist*. Cambridge, U.K.: Cambridge Univ. Press, 2022.

SIAR SARFERAZ (Member, IEEE) received the degree in computer science and philosophy and the Ph.D. degree in computer science. He is currently the Chief Software Architect of the Enterprise Resource Planning (ERP) Solution, SAP SE, working in the Research and Development Department, Walldorf, Germany. In this role, he drives the digital transformation by defining the solution architecture for the product and by providing concepts scaling for mission-critical business processes. He is the Lead Architect of artificial intelligence implementations for SAP's ERP product and is responsible for all conceptions of how to add intelligence to business processes. In the context of ERP software, he owns more than 30 patents and published numerous books. He began his career as a method Researcher with Siemens, before moving to SAP, where he has been for more than 20 years.