IEEE Region 4
Serving IEEE Members in all or parts of Illinois, Indiana, Iowa, Michigan, Minnesota Nebraska, North and South Dakota, Ohio, and Wisconsin

IEEE Region 4

2024 – Issue 04

Contents

Director's Column

A Heartfelt Thank You as I Conclude My Term as IEEE Region 4 Director

As I near the end of my term as IEEE Region 4 Director, I want to take a moment to reflect on the remarkable journey we've shared together and extend my deepest gratitude to everyone who contributed to making this experience so rewarding. This role has been a tremendous honor, and I am proud of the progress we've made as a community. However, none of this would have been possible without the support, commitment, and collaboration of countless volunteers.

Over the past two years, we have achieved a number of significant milestones in Region 4. From expanding our member services to promoting cutting-edge technical initiatives, every accomplishment has been the result of collective effort. I'm incredibly proud of what we've achieved, particularly in fostering innovation, increasing engagement among our sections and chapters, and supporting the professional growth of our members.

Some highlights include:

- New and rejuvenated Affinity Groups - IEEE Southern Minnesota Section, Women in Engineering Affinity Group, IEEE Twin Cities Section, Women in Engineering Affinity Group
- New Chapters - IEEE Central Illinois Section Computer Society Chapter, IEEE Twin Cities Section Sensors Council Chapter, IEEE Twin Cities Section Engineering in Medicine and Biology Society Chapter, IEEE Engineering in Medicine and Biology Society Student Branch Chapter at the Mayo Clinic Graduate School, IEEE Microwave Theory and Technology Student Branch Chapter at the University of Wisconsin-Lacrosse, IEEE Systems, Man, and Cybernetics Society Student Branch Chapter at the Purdue University-West Lafayette, and finally the Southeastern Michigan Magnetics Society Chapter.
- New Student Branch - IEEE Student Branch at Fox Valley Technical College, IEEE Student Branch at University of Wisconsin-Lacrosse
- EIT continues to be a success and we brought back our Student and Young Professional Conference Nexus, along with other events across the Region
- Hosted quarterly Region Senior Elevation Events having 226 members advance to Senior Member year to date in 2024 and 198 in 2023
- We announced a new website and increased our social media presence, reaching new members and old

None of these successes would have been possible without the incredible work of our volunteers, the leadership of the Section Chairs, and the participation of our dedicated members across the Region.

The true heart of IEEE Region 4 lies in its volunteers and leaders. I am deeply thankful for the unwavering dedication and support that the many chairs, vice-chairs, and committee members have shown. You have all played pivotal roles in ensuring that Region 4 remains a vibrant, dynamic community. Thank You!

To our Section Chairs, thank you for your leadership in steering your local communities. Your hard work and passion are the foundation of IEEE's success in our Region. I also want to acknowledge the invaluable contributions of the Region 4 leadership team, which has been instrumental in executing our vision.

I'd also like to express my heartfelt appreciation to the members of IEEE Region 4. Whether you are a student member just beginning your journey or a senior member with decades of experience, you are the reason why our Region continues to thrive. Your enthusiasm, dedication, and commitment to advancing technology for humanity are truly inspiring. It has been an honor to serve you, and I look forward to continuing to work alongside many of you in the future.

While my term as Director is coming to an end, I am confident that the strong foundation we've built will continue to support and guide the Region into the future. We are at the forefront of numerous technological advancements, and I have no doubt that Region 4 will continue to play a leading role in shaping the future of engineering and technology.

I am excited to pass the baton to my successor, who will undoubtedly lead with the same passion and vision that we have cultivated together. I am certain that the Region will continue to grow, innovate, and make a lasting impact on the world.

In closing, I want to reiterate how grateful I am for the opportunity to serve as your IEEE Region 4 Director. It has been an incredibly rewarding experience, and I will forever cherish the relationships we've built and the work we've accomplished. Thank you for your trust, your support, and your commitment to advancing the IEEE mission.

As always, thank you for all you do for IEEE!

Respectfully submitted,
Vickie Ozburn, IEEE Region 4 Director (2023-2024)

Editorial Corner

In this issue:

Presenting the 2024 Q4 and EOY edition! This has been by far the largest edition ever produced and almost resembles our flagship IEEE publications – due to the numerous technical articles contributed by many of the IEEE members. Thank you to all and keep them coming!

Among some of the technical content rich and detailed-oriented highlights are the articles by Gaurav Shekhar on AI and software development, as well as an interesting one on EV charging case study in an urban city setting by Sid Bennet. Other technical articles to note are: API gateways by Ikram Mohammed, AI and Cloud trends by Ayisha Tabbassum, GenAI and Cybersecurity by V. Mandela; AI and Healthcare Risks by Vijay Viradia; Insurance policy migrations in the IT field and methods of data retrieval using GenAI by a repeat author – Shamila Chandariah (note these are two articles in this edition, she had contributed in a past issue to the R4 newsletter as well); finally a tech note by Dipankar Saha on an open source project Apache Iceberg. We hope that will inspire more budding tech authors to come forward.

Hearty congratulations are due to all the recently elected fellows. Time and bandwidth permitting – we will try to feature them in future editions of the newsletter, asking them to describe their work and share a little bit about themselves.

Congratulations to our member who got recognized by the British Computer Society: Jeevan Sreerama.
And don't forget to check out the CFP (Call for Papers) for our very own R4 EIT 2025 conference!

This issue has already become very lengthy and we try to keep it just long enough to allow for readers attention and so a few articles had to be postponed to the next edition.

Previous editions in this series may be found on the Region 4 website. Click on the "Newsletter" button in the top left column. Comments, newsletter submissions, articles of interest and suggestions may be sent via email to the editor: sharan.kalwani@ieee.org

Microsoft Word format is preferred but we can work with ODT as well. Where possible use the Arial font in point size of 10 and single spacing. Images can be in either *JPEG, GIF, PNG or similar formats.*

We try to complete the newsletter layout a week before publication, to allow time for review and corrections. If you have an article or notice, please submit it as early as possible. *We publish once every quarter.*

The newsletter relies on the contributions of our members and officers, so please do not be shy. If you have something that should be shared with the rest of the region, we want to give you that opportunity. The next deadline will be the middle of February 2025 (around the Valentine's Day).

Sharan Kalwani,
Editor, Region 4 Newsletter and Enthusiastic IEEE volunteer
Chair, IEEE Southeastern Michigan Section (2021-2025)

Call for Volunteers

Thank you for being an IEEE member and a member of IEEE Region 4. As a Member of IEEE you automatically become a member of your local IEEE Section, this allows you to share technical, professional, and personal interest with others in the worldwide member community of IEEE.

Are you looking for a way to get more involved within your local IEEE Section or Region 4? **If so**, **We want you!** Do you want to help guide programs or project ideas or maybe take part in a micro volunteering activity? So you may ask what is micro volunteering.

Micro Volunteering: Making a Difference in a Matter of Minutes.
Micro-volunteering describes a volunteer, or team of volunteers, completing small tasks that make up a larger project. These short, infrequent volunteer opportunities are often called "microvolunteering," which allows people to volunteer for specific tasks that can be completed in a short window of time. We want to make volunteering for IEEE fun and easy.

One of the objectives for Region 4 is to recruit and provide leadership and volunteering opportunities to our members. In order to accomplish this, we will send in regular intervals a Form to seek Volunteering and Leadership interest for our members.

Please let us know and we'll be happy to help out and find a spot just for you. We request you to please fill out the following form to express your interest:
https://docs.google.com/forms/d/19k46v6NsE1TwwR4Bky4MgNvdIKRN46LJ9x2x3pOuioM

AI & SW Development

Gaurav Shekhar
Vice President- Sr Group Application Manager
https://www.linkedin.com/in/gaurav-shekhar-engineer/

The Impact of AI and Automation on Software Development: A Deep Dive

Abstract: This is due to the fact that the technological growth, most especially in the artificial intelligence and automation system has influenced a number of fields, among them being software development. Also, in this paper, the author looks at the advancement of AI and Automation in software engineering and discusses the effect of the two key concepts in enhancing the development processes, efficiency and quality of code, as seen in the sections below. In this part, the tools and techniques involved in ASD are described, the benefits and issues are explored, and the different roles of developers are also described, especially in the context of ASD. AI's impact at different phases of the software development life cycle, such as requirement analysis, design, coding, testing, and implementation, is analyzed. The applicability of the AI tools, examples including machine learning models and automated code generation tools, are also discussed in considerable detail. This study is divided into six sections: There is the research proposal including such sections as the definition of the problem, literature review, methodology, results, and discussion with the conclusion. The introduction only gives the background on AI, automation and their applicability to the development of software. A literature review also presents a historical perspective of the integration of AI in software engineering and major work and developments. The methodology highlights the methods which were employed in order to collect the necessary information and knowledge. In the result and discussion section, this study provides the outcome of the research. It measures the benefits of using AI in terms of coding efficiency, reliability in software, and cost-effectiveness as well. Last but not least, the conclusion explains the opportunities and threats that underlie the AI revolution to refashion the software development paradigm.

Keywords: Artificial Intelligence, Automation, Software Development, Machine Learning, Code Generation, Automated Testing, DevOps.

1. Introduction

It is interesting to examine how software development has changed within the last few decades from a process that relied more on manpower and manual coding to a process that incorporated automation and artificial intelligence (AI). This was tedious and lasted for some time, especially with lots of coding and debugging being done manually, which introduced a lot of errors. [1-3] Automation tools came into the picture as another revolution, which brought concepts to solve the mundane problem of mere coding where certain functions like compilation and testing were cumbersome. Over time, artificial intelligence became a revolutionary concept, which introduced better features into the software-making process. Today there are available tools which may generate small code snippets, detect even the most complex bugs and fix them, suggest means for increasing work performance and even help manage large projects. This move towards an AI-intelligent approach to development is changing the approach and flow of development while making it faster, more accurate and much more efficient.

1.1. Importance of AI in Modern Software Development

The use of Artificial Intelligence (AI) in today's software development environment has ensured various advantages that improve efficiency, precision, and creativity. Here is an in-depth look at its significance:

- **Enhanced Coding Efficiency:** Specifically, AI tools enhance the coding speed because most integrated tasks are repetitive and can be automated or have templates created for them. For example, modern code companions such as GitHub Copilot and OpenAI Codex are capable of generating code driven by simple natural language descriptions or code stubs that are not yet full-fledged, thereby saving the developer's time for completing repetitive tasks. What is more, this automation accelerates development while helping developers concentrate on deeper, more creative decision-making processes related to software design and architecture. Simple coding tasks reduce the burden of coders and contribute to the progress of the project to be delivered faster.

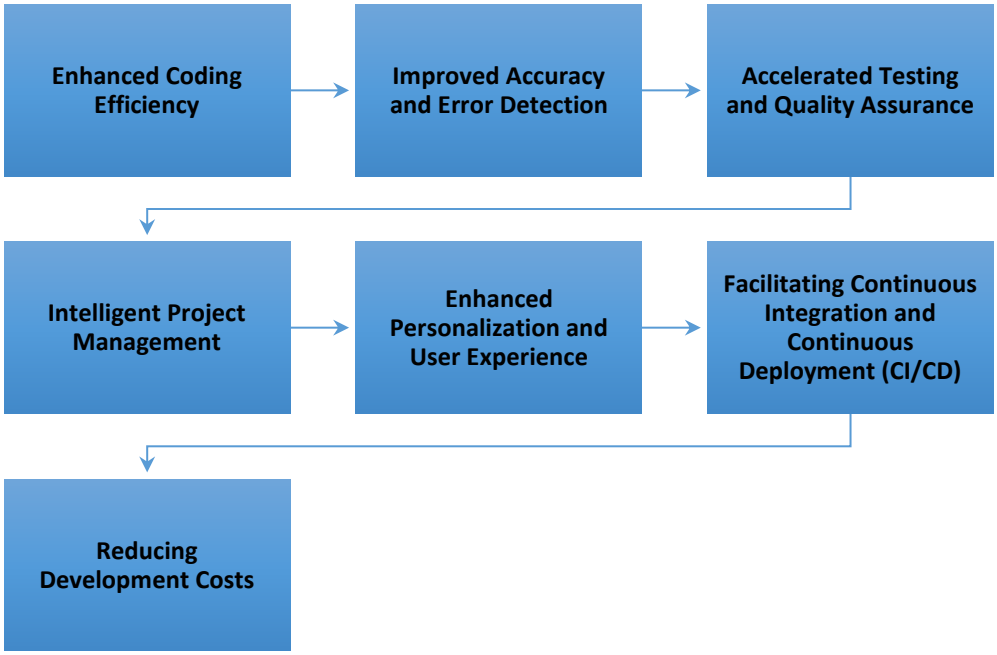


Figure 1: Importance of AI in Modern Software Development

- **Improved Accuracy and Error Detection**: Arguably, this is one of the most important benefits that have been provided by AI to software development, specifically in that it provides means to enhance the correctness of subsequent processes as well as error checking. Services such as DeepCode and CodeGuru employ an artificial intelligence algorithm that will analyze the line of code and detect areas of the code that might contain errors or even areas that consist of flaws that human eyes cannot observe. Inflammations with such tools lower the risk of key mistakes being pushed to online production, giving software more reliability and quality.
- **Accelerated Testing and Quality Assurance**: A highly effective area that relies on software testing and is time-consuming as well as requires a lot of effort is also an excellent candidate for the usage of AI. Then there are AI-based testing tools like TestComplete and Tricentis Tosca that can create and run tests with a small or even no contribution of a human element. AI can use historical test data to come up with edge cases and then optimize for the weakness hence thorough testing. This not only helps in speeding up the process which is the testing phase of any software, but also increases the efficiency of various tests that are done, which in blends gives good and more accurate testing results, thereby giving more efficient and bug-free software.
- **Intelligent Project Management:** Two more highly significant application areas of AI are predictive analysis and task automation in project management. Appropriate project management tools powered with artificial intelligence have the capability to predict a project's duration and potential hazards and even allocate resources according to the project's specifications and historical data. These predicted capabilities enable decision-making, in this case by the project managers, to optimize workflow and overall project performance.
- **Enhanced Personalization and User Experience**: Mobile applications can be made more specific to the users' needs by integrating AI into the development of software applications. By evaluating the users' activity and preferences, AI can propose new options, changes to the interface, or content relevant to the individual user. This capability improves user satisfaction by aligning software with regard to the user preference, making the software respond to the user's desire, and making the user more engaged.
- **Facilitating Continuous Integration and Continuous Deployment (CI/CD)**: Today, CI/CD practices are crucial in the field of software development in order to keep code of high quality and shorten deployment time. Artificial intelligence helps to automate the CI/CD pipeline by incorporating the build, testing as well as deployment procedures. AI tools can look at code changes, run builds automatically and can also release updates using the least human interactions. This automation guarantees that new functions and patches are deployed swiftly and stably and promotes the usage of agile development paradigms.
- **Reducing Development Costs**: In one way or the other, various techniques of software development are made cheaper through the use of Artificial Intelligence. Implementing AI in the process also optimizes the amount of time and effort that is active in a work, thereby providing shorter time periods for the execution of the projects. Furthermore, the identification of bugs at an early stage and the optimization recommendations also help in preventing a large number of late fixes and enhancements hence reducing the total cost throughout the life cycle.

1.2. Evolution of AI and Automation

Artificial Intelligence (AI) and automation can be described as the advancement that has improved technology and industries over time. This section discusses the historical perspective and history of AI and automation with emphasis on its evolution from a concept to a reality in software development.

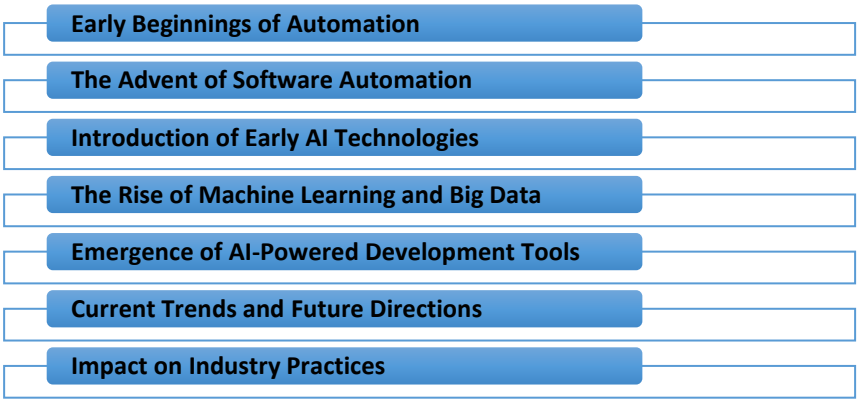


Figure 2: Evolution of AI and Automation

- **Early Beginnings of Automation:** Automation can be dated back to the early part of the twentieth century with the friction between mechanical automation and early calculating machines. Process automation was initiated by assembling line innovation, which capability extended to software advancement. Within computing, the start of the innovation in the first electronic computers established in the years around the 1940s and 1950s, like ENIACs and UNIVACs, became the basic technology for the later developments in software and automation.

- **The Advent of Software Automation:** The next significant advancement was in the period of 1960 and 1970, and was commonly termed software automation as the first compilers were invented together with early programming languages. Compilers enabled the mechanization of translation of higher-level languages into low-level languages, therefore minimizing the amount of coding work. During this period, Integrated Development Environments (IDEs) appeared, and the first version control systems were also introduced which provided auto mechanics of the software.

- Introduction of Early AI Technologies: The late seventies and the eighties that continued up to the nineties saw further advances in AI, whereby the first expert systems and the initial uses of machine learning were developed. MYCIN and DENDRAL are examples of expert systems, which were created to mimic real-life experts in certain field and to offer useful inputs and recommendations. With the algorithms for machine learning and pattern recognition society created the basis for the progress of AI. During this period, the first generations of automatic testing tools and continuous integration systems appeared, which opened new opportunities in the developers' workflows.

- The Rise of Machine Learning and Big Data: Machine learning and big data, which emerged in the early 2000s, are founded on the enhancement of computational abilities and media storage. Concepts of Artificial intelligence and Machine learning evolved as the FLT systems were capable of understanding and learning from large data sets over a period of time. The application meant that AI systems could draw more reliable conclusions based on big data available at their disposal. Thus, at this stage, initial penetration of AI technologies into software development took place with the help of tools for the automated analysis of code, bug detection, as well as optimization of the program's performance.

- Emergence of AI-Powered Development Tools: The use of Artificial Intelligence tools in software development grow on a steep slope in the 2010s. GitHub Copilot, OpenAI Codex and DeepCode serve as examples of the use of AI in code automation, bug detection and code review. All these tools use NLP, neural networks and deep learning to improve elements of the SDLC process such as software requirements definition. The help of AI also developed other testing capabilities. These included consistent machine learning and even prognosis capabilities and the ability to create test cases automatically.

- Current Trends and Future Directions: In subsequent years there has been improvement in the advanced use of AI and automation where the approach advances to adopt the use of AI in DevOps and CI/CD. Today's AI technologies provide developers with intelligent code completion, infrastructure manipulation, and monitoring with intelligent alerts. The future of AI in software development is to attend even higher levels of skill, including autonomous coding assistants AI project management and other more profound depths of integrating deep machine applications with the blockchain and quantum computing technology.

- Impact on Industry Practices: The advancements in the technology of AI and automation have changed the practice in industries especially how the software is developed, tested, and maintained. The application of automation has helped in cutting costs, reducing manual labor, and speeding up development cycles. AI has helped in improving the quality of software through such aspects as intelligent insights and even predictions helping in improving the development processes. Due to constant enhancement in the technology of AI and automation, the development of software faces expanded challenges in the future.

2. Literature Survey

2.1. Historical Evolution of Automation in Software Development

There have been advancements made in the area of automation in software development and these have been defined below. The process of evolution started in the year 1950s when the development of compilers emerged, which helped to translate standard languages to machine language, making the programmer's task easier and fast. [4-9] It is important to notice the development of such an early tool, which laid the groundwork for future tools. Continuous integration tools like Jenkins came into use in the 1990s, and Selenium for testing added more capability to the software development process. Jenkins changed the means by which developers incorporated code changes, and Selenium, on the other hand, prepared the means for higher levels of AI with its functional testing of web applications. However, early automation was not as intelligent or flexible as what current high-end AI incorporate into the automation process. Modern tools that incorporate AI Malware learn from these technologies as they present improvements in the ability to write code, evaluate it as well as deploy it.

2.2. AI and Code Generation

The concept of generation of code through the use of artificial intelligence can be dated back to the 1990s. However, the early generation approaches were simple and required careful guidance from the human programmer. Code generators of that period were rather narrow tools that could be used only in single sectors and which are different from the intelligent systems of the present day. This is because in the recent past with the incorporation of AI on the code generation tools, the progress has been immense. For instance, GitHub Copilot and OpenAI Codex bring another level of AI-driven code generation where the AI is fully capable of generating larger sequences of code from the input given by the user. These tools incorporate such features as smart context-sensitive code completion, which employs the use of machine learning to learn the intent of the programmer and write the code to be typed in with increased efficiency, thus reducing the instances of coding that have to be done manually. It is possible to discuss that the transition from such concrete domain generators to smart systems can be considered the key achievement in generating the code automatically.

2.3. AI in Software Testing

Software testing is one of the phases of software development that requires a lot of effort; however, significant improvements have been achieved in the application of automation. Earlier automation testing tools like Test Complete and Tricentis Tosca helped minimize testing efforts that were performed manually. The above tools helped in automating most testing activities which helped in improving testing activities and making them more effective. Modern software testing has continued to be changed by the incorporation of AI over the last several years. The use of machine learning methods in handling source code patterns for the creation of test cases and when it went beyond automation incorporated the ability to predict. It is now possible for the deep learning algorithms to feed the developers with extra details regarding the possible loophole or edge cases that a developer may not have a probabilistic vision of hence improving the testing effectiveness and the general software quality. This evolution is a great improvement in the level of testing in software development and enhancement.

2.4. Machine Learning Models for Bug Detection

Nowadays, machine learning models have shifted to the center of the activities that are aimed at detecting and eliminating the problems associated with code issues at the initial stage of product development. Software like DeepCode and CodeGuru use the data and the sophisticated techniques of machine learning to shed light on areas of code that are most possibly buggy. Often, these tools are applied to massive databases of code changes and defect patterns, which in turn offer recommendations to the developers dealing with the problems at hand, thus preventing further development of these problems. This predictive capability not only reduces the debugging time but also the quality of the developed software because it allows developers to fix vulnerabilities before they can be exploited. It is important to note that the use of

machine learning in bug detection, comes as a much better approach than the conventional one in that it provides a much better and timely outcome in identifying code quality.

2.5. The Rise of DevOps Automation

DevOps practices, which emphasize continuous integration, continuous deployment (CI/CD), and ongoing monitoring, have been greatly enhanced by automation. The integration of AI into DevOps has optimized complex workflows, minimized human errors, and accelerated deployment processes. Tools such as Ansible and Puppet, along with AI-driven platforms like Harness, play a critical role in managing infrastructure, ensuring smooth software releases, and handling rollback strategies. AI's ability to automate and streamline these processes has led to faster, more reliable deployments, contributing to the overall efficiency of DevOps practices. The rise of AI in DevOps reflects a broader trend towards greater automation and intelligence in software development workflows.

2.6. Key Challenges in Implementing AI and Automation

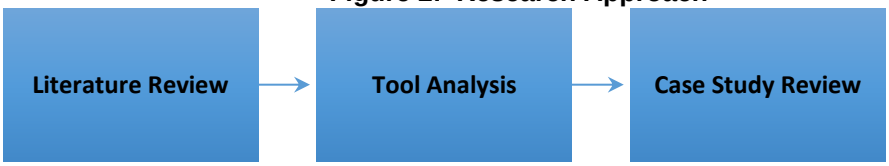
Nevertheless, there are a number of challenges that still arise in AI and automation in software development. A major challenge relates to the relative newness of this technology to most developers, who may not have the necessary experience or knowledge of the tools available and their implications in development, hence the challenges in adopting them. Also, the cost of implementing such AI technologies is high, a factor that does not augur well with SMEs, who might not afford the high costs needed to implement the technologies. AI models also have issues to do with bias and accuracy whereby the model will have biased data fed into the program hence a wrong code or suggestion. These issues, therefore, suggest rebutted approaches that will help in addressing the challenge, such as the training programs, costs, data quality and model accuracy which must be an ongoing process to ensure they improve continually.

3. Methodology

3.1. Research Approach

This study adopts a qualitative research approach [10-14], drawing on three key methods to explore the integration of AI and automation in software development:

Figure 2: Research Approach



- **Literature Review:** In particular, the literature review involved the evaluation of existing information sources which includes journal articles, conference papers, and white papers. However, such sources gave a sufficient explanation regarding the outlook, development and issues faced in employing the paradigms of AI and automation in designing software. Analyzing the literature review of theory and the recent research work available in the literature, one came to know how AI has transformed from automation tools to smart and efficient tools which help the developers in various phases of the Software Development Life Cycle (SDLC). Some of the issues that were raised in this review and which are to date include; cost of implementation, bias from the implemented AI model and proficiency in AI technologies in software engineering.
- **Tool Analysis:** This research has involved a critical assessment of Artificial Intelligence based software development tools. We chose several popular tools, including GitHub's GitHub Copilot, DeepCode AI, and the automation tool Ansible and read through their documentation and specifications, as well as their application scenarios. Official sites and developers' feedback were also considered to evaluate the ability of all the tools in code generation automation, bug detection, and other DevOps tasks. This took a closer look at how these tools work, what AI technologies underpin them (e. g., machine learning, NLP or rule-based automation), and the benefits that developers can expect. Also, there emerged how the evaluation showed how the use of AI saves time in coding through the suggestion of optimized code and efficiency in testing.
- **Case Study Review:** The review of the case study focused more on real-life Artificial Intelligence and Automation adopted by organizations such as Microsoft, Google as well as IBM. These companies have been defining the use of AI for the development of software applications since they have adopted the use of tools and frameworks that incorporate AI in the development process. Studying these cases allowed us to obtain practical experience in implementing the advantages and drawbacks of using AI in LSEWE projects. Microsoft has recently released GitHub Copilot, which demonstrated its efficiency, and, in the case of Google, the corporation applies machine learning for code optimization and drastic increase of deployment speed with zero impact on code reliability. These case studies also showed how, within DevOps, AI has been applied in the CI/CD pipeline.

3.2. Data Collection

The data collection for this study relied on three main sources, each contributing a unique perspective on the impact of AI and automation in software development:

Figure 3: Data Collection

- **Academic Journals:** Articles were used as the primary source mainly with that purpose in mind because they provided a theoretical foundation on the subject of Interest under consideration – Artificial Intelligence and automation of software engineering. It is so only to a certain extent since the amount of papers till May 2023 helped in building background to understand the change that had taken place in the practices having links to AI-integrated SDLC. Transactions on Software Engineering, Journal of Software Engineering and Computing surveys published articles having the enhancement in algorithms using Artificial Intelligence, the problem of incorporating artificial intelligence and artificial intelligence in the construction of code, testing and bug detection. These sources were helpful in putting limits on the fresh trends and finding out how much more is required to be studied on some issues that are still undiscovered and, therefore, the academic backgrounds of this study.
- ***Industry Reports:*** The analysis of the different publications like reports and white papers of leading tech companies like Microsoft, Google, IBM, etc., was helpful to anchor the study in real-world applications. These are some of the leading companies, which adopted artificial intelligence technology and have put AI tools into practice in their development processes. Their reports offered a rich source of information on how AI and automation are disrupting and revolutionizing software development, DevOps automation, intelligent bug detection and test automation, among many others. Success stories in those reports described examples of how the deployment of AI translates into tangible improvements in productivity, code quality, and time to market. Moreover, these documents provide information on implementing AI issues and solutions that have been encountered; among them, there were scalability problems, the incorporation of AI tools into existing ones, and human supervision concerns.
- ***Tools Documentation*:** The official technical papers of famous AI-based software development tools, including GitHub Copilot, OpenAI Codex, DeepCode, and Amazon CodeGuru. Another important source of this study was the manuals and documents of the above-said tools. These documents offered technical specifications on the use of each tool and the architecture of the tool along with the AI methodology used in it, which includes machine learning, NLP and rule-based automation. From the analysis of these documents, we were able to get familiar with the strengths and weaknesses of each tool, how it worked in different programming languages, users' feedback, and the existing issues or bugs. Having outlined general descriptions of these tools and their relevance to improving the efficiency of software development, this detailed technical analysis provided a valuable basis for comparing them.

3.3. Tools and Techniques

In this work, several AI tools across the SDLC were also assessed. Here are the tools that signify progressing in AI technologies, solving problems like code generation, bug detection, and the DevOps process. [15-18] below, we provide detailed insights into three key tools: GitHub Copilot, DeepCode, and Ansible.

3.3.1. GitHub Copilot

GitHub Copilot is a cutting-edge assignment writing assistance tool that utilizes Artificial Intelligence that GitHub created with the contributions of OpenAI. The tool uses machine learning and Natural Language Processing (NLP) in order to help the developers receive code suggestions based on the natural language input in real time. Copilot is tightly embedded into software development tools such as Visual Studio Code, which enables developers to code much faster by providing means for automating mundane, repetitive tasks and writing less boilerplate code. Due to its capability to provide context-sensitive code fragments, it can be most helpful for upgrading efficiency in the early phase of programming. Copilot works with a wide range of programming languages and offers code completions right in the line so that the developers can improve their work as fast as possible.

- **AI Techniques Used**: GitHub Copilot masks itself behind the NLP and machine learning algorithms with learning based on billions of lines of open-source code.

- **Key Features**: The features added include context-aware suggestions, the support of multiple languages in-code, inline autocompletion, and autocompletion of entire functions or classes from a text area comment or hash-prompt.

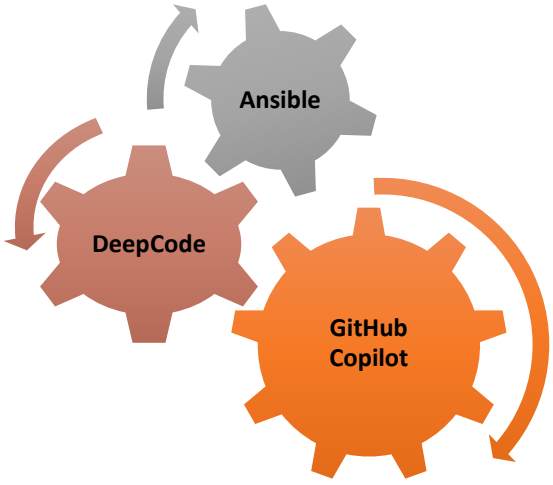


Figure 5: Tools and Techniques

3.3.2. DeepCode

It is an AI tool used specifically for code analysis and bug detection with the name DeepCode. It deploys machine learning for code base search for defects analysis of datasets to detect security vulnerabilities in the code base. Besides, DeepCode is not a typical static analysis tool whereby code is analyzed conventionally or based on the experience of human developers as well as other conventional methods; instead, it incorporates Artificial Intelligence that helps it to learn from millions of open-source projects and is therefore capable of detecting subtle problems. DeepCode is used to refine the quality and the security of code, and thus, it is more useful to the teams in the testing and debugging of the code at the SDLC.

- **AI Techniques Used**: The most important technology used by DeepCode is machine learning used for code pattern recognition and for searching for anomalous code. These models are updated based on open repositories and the performance is very dynamic.
- **Key Features**: This is; automated bug detection that helps to identify weak links in the code, active scanning for security vulnerabilities and AI suggestions when it comes to improving the code.

3.3.3. Ansible

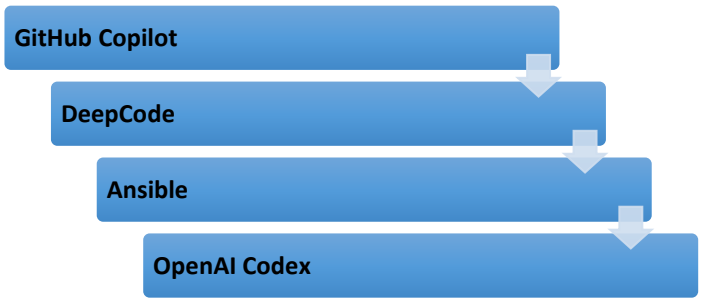
Ansible can be described as an automation tool that is primarily structured to address IT environment management and application deployment and simplification of repetitive DevOps activities. However, as is the case with many of the advanced tools and applications we mentioned in this guide, Ansible does not actively use AI technology like that of GitHub Copilot or DeepCode. However, it serves as the backbone of most corporations for rule-based automation of heavy workloads. A simple statement lexicon defines Ansible to describe the tasks in the organization to ensure that it automates everything ranging from server instantiation to continuous delivery pipelines. There is no doubt that this tool is quite popular in the DevOps domain and is used for managing Infrastructure as Code (IaC) and for achieving automation in the deployment process.

- **AI Techniques Used**: Nevertheless, the software does not incorporate top-notch artificial intelligence strategies, although it applies rule-based automation, which can help avoid mistakes and organize the work well. This makes it a truly useful tool to deploy large scale and complex deployments on an automated basis.
- **Key Features**: It has IaC, automation of various difficult tasks in DevOps, continuous delivery, and plenty of modules that help in various configurations of IT solutions and business processes.

3.4. Detailed Comparison of AI Tools

Comparison of the various AI tools makes it easier to understand the capabilities of the tools, AI methods employed, and the benefits derived. The following table summarizes these aspects for GitHub Copilot, DeepCode, Ansible, and OpenAICodex:

Figure 6: Detailed Comparison of AI Tools



3.4.1. GitHub Copilot

- **Functionality**: GitHub Copilot is intended to augment development by providing code completions as well as code relevant to the context of a conversation. It works well with code editors, as it will give context-sensitive suggestions about the code that one is typing and makes coding easier and faster. Basically, it is used to minimize the amount of time one spends while coding basic or repetitive scripts and to assist in the generation of prototypes or templates.
- **AI Techniques** Used: GitHub Copilot uses Natural Language Processing and Machine Learning techniques. These approaches help the tool to parse the task descriptions in natural language and provide the code snippets by training on a large number of programming samples.
- **Key Advantages**: That means the main advantage that one can extract from GitHub Copilot includes eliminating the amount of boilerplate code, thereby making the process of development faster and more efficient. Also, it has built-in support for multiple languages to enable developers to write in different programming languages, making it a universal tool meant for different coding systems.

3.4.2. DeepCode

- **Functionality**: DeepCode is a company which mainly focuses on analyzing the code in search of bugs and security flaws. To the best of my understanding, what I know is that it relies on machine learning to analyze the code and come up with probable suggestions on potential defects. This ensures that there is a possibility of correcting mistakes before they reach a concerning level; hence is a good measure of ensuring high quality of code as well as quality of security.
- **AI Techniques Used**: As for DeepCode, it is noted that the system is based on machine learning and pattern recognition algorithms. One of the benefits it sustains is that it can also identify patterns that a human will not easily notice due to the large database of code it trains on. This helps the tool extract information on how to optimize the code and any security risks that need to be made known to the developer.
- **Key Advantages**: The main advantage of DeepCode is that DeepCode is good at identifying vulnerabilities and bugs in the code and, as a result, enhancing code quality. The self-jeopardizing results show that automated analysis minimizes human-intensive work, such as code reviews, in addition to increasing the overall code reliability of the software systems.

3.4.3. Ansible

- **Functionality**: The role of Ansible is to bring as much IT infrastructure and application management, deployment and operation automation as possible. It is a declarative language where automation tasks are defined, making it easy to work with complex environments as well as deployments. This makes it a very important tool in DevOps practices as well as infrastructure as code (IaC).
- **AI Techniques Used**: While many applications use AI technologies, it should be noted that Ansible mostly employs a rule-based automation approach. Its automation features are based on procedural algorithms and setting that guarantee the program's performance in various settings.
- **Key Advantages**: The benefits of Ansible are its general usage in automation of the deployment tasks, organizing the infrastructure and serving as a simple-to-configure solution for DevOps tools usage. The primary benefit of Arbortext is that it adopts a rule-based style of interaction that can produce very dependable and constant deployments.

3.4.4. OpenAI Codex

- **Functionality**: OpenAI Codex performs well in the generation of difficult codes and tackling of problems. It enhances the elements of artificial intelligence to solve complex algorithms and also employs a deep learning algorithm to offer suggestions. Codex can write code in terms of fragments, and also full functions or individual programs from a detailed text description in natural language.
- **AI Techniques Used**: From the previous discussion, we are able to deduce that OpenAI Codex optimizes Natural Language Processing (NLP) and machine learning. It stands on the shoulders of deep learning models for coding problems and comes up with code completion services that reflect the coding specifications as articulated in NLP statements.
- **Key Advantages**: The key strengths of OpenAI Codex include complex algorithm processing as well as offering suggestions based on deep learning. Due to this, it forms one of the best solutions for handling compact yet complex programming utilities and enhancing the development process.

4. Results and Discussion

The following section focuses on the results identified from the research on tools leading to AI in software development. It outlines gains in efficiency, the advancements made to product reliability and difficulties associated with the implementation of these tools in organizations.

4.1. Impact on Software Development Productivity

From the article's scientific analysis, it is clear that the application of artificial intelligence boosted productivity in software development. These tools can automate some of the specific stages of the development process, thereby freeing up the developers to engage in more difficult or creative processes. Here is a detailed look at how different AI tools impact productivity:

- **GitHub Copilot:** GitHub Copilot is among the shining examples of how AI can help to increase the pace of coding. Copilot makes developers save time by automatically generating the boilerplate code and thus frees them from the tedious task of coding. This automation not only increases the rate at which development of the project can take place but also minimizes the chances of which are often accompanied by mistakes done through coding. The opportunity to generate the code is useful for developers, especially if the majority of the writing is done by Copilot, whose tips can help achieve a faster result and thus free up time for what is more complex or creative.
- **Automated Testing:** Automated testing tools are also worth mentioning here as they have greatly enhanced productivity as well. These tools perform tests that would have otherwise required so much time and effort to complete on the normal paper. With ad hoc testing, developers and the QA teams can guarantee better coverage and faster feedback on coded changes, especially if the tests are automated. Automated testing gets problems earlier in the developmental cycle so that less time is expended on manual testing while more is accomplished overall.
- **DeepCode:** First, the information shows that DeepCode generates the AI-identified bugs, which saves the developer's time when debugging. The current flow and vector static analysis tools can only detect simple mistakes, while DeepCode's models are built to detect a wider range of problems. With better and quicker bug detection, DeepCode reduces time spent on constructing problems and empowers developers with early identification time so that they will not have to rely solely on manual code reviews and fixes.

Table 1: Time Savings with AI Tools

Tool	Task	Time Saved (%)	Description
GitHub Copilot	Code Generation	40%	Automates boilerplate code, speeding up development and reducing manual coding efforts.
Automated Testing	Test Execution	50%	Saves time by automating repetitive testing tasks, allowing

			for more comprehensive testing with less manual intervention.
DeepCode	Bug Detection	30%	Enhances productivity by identifying bugs more quickly than traditional methods, reducing the time spent on manual debugging.

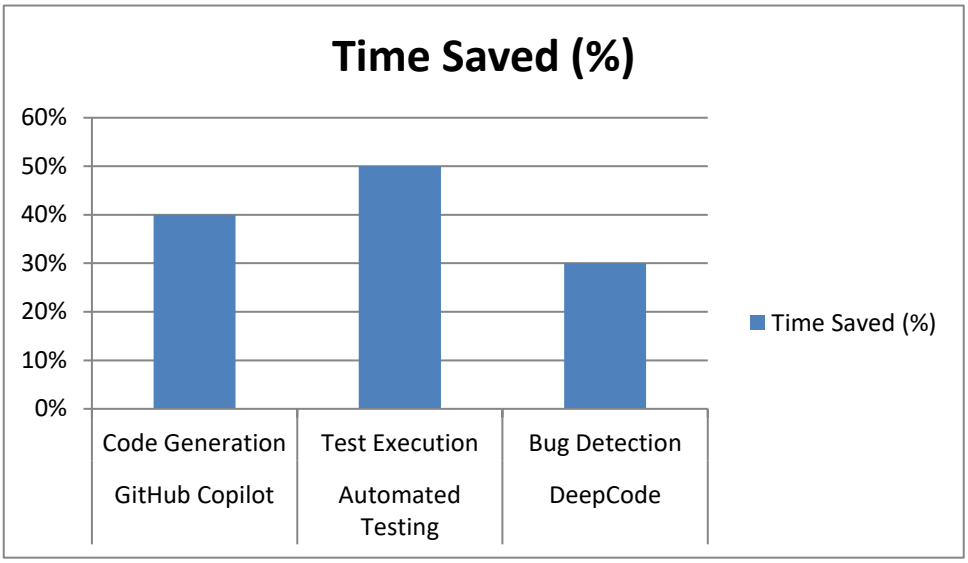


Figure 1: Time Savings with AI Tools

4.2. Quality and Reliability

AI tools are of great importance in the quality and reliability of the software as it is evident through the reduction of bugs and better code analysis. Of all these tools, DeepCode has been quick to show higher results, especially in defect detection, hence improving software quality.

- **DeepCode's Impact on Bug Detection**: The algorithms used by DeepCode to build and develop its product have helped enhance the detection of defects as opposed to ordinary techniques. By using machine learning and pattern recognition DeepCode can provide a more comprehensive analysis and find flaws, which are not identified by the standard tools for static analysis of the code. In a real-life scenario, DeepCode has been seen to outperform conventional detectors by a percentage point of thirty percent and, therefore, indicates its efficiency in identifying latent defects and availing better code quality.
- **Quality Improvement through AI-Driven Tools:** Useful tools such as DeepCode not only enhance the ability to identify bugs but also enhance the quality of created programs. These tools assist the developers in detecting faults during the development cycle, thus helping them solve these problems as they are still simple to solve. They practice a bug prevention strategy as a way of minimizing the number of bugs that make it to the product being released into the market. Moreover, the sources that are based on AI can suggest improvements to the code that is used, which makes the code more clean and maintained. Such continuous feedback is useful to guide the programmers into sticking with the best practices as well as coding standards, hence improving the quality of the resultant software.

Table 2: Bug Detection Effectiveness

Tool	Detection Improvement (%)	Comparison to Traditional Tools
DeepCode	30%	Superior in detecting hidden bugs and defects
Traditional	-	Baseline detection effectiveness

4.3. Challenges in AI Integration

However, there are several threats which may affect the application of AI tools in software development even though there are numerous benefits as highlighted in section two. Sensitive identification of these challenges is important in enabling developers to get the most out of the AI technology in the different developmental workflows.

- **Developer Resistance:** The problems that arise when transitioning to AI tools include developer pushback against using them. This resistance can be due to a lack of prior exposure to the new technology, doubt about the success of the use of the technology or fear of disruption of the protocol. Some developers may be wary of using AI tools in executing important tasks or think that such tools pose a threat of taking over their functions. This sort of resistance may well delay the process of adopting these innovations and reduce the benefits that can be derived from AI tools. To overcome this challenge, proper strategies, including training programs, demonstration of how the specific tools work, and incorporating developer's feedback that will help in improving the specific tools are important.
- **Financial Hurdles:** One disadvantage associated with the use of AI tools is that a considerable capital outlay may be needed to acquire and implement the tools. Monthly costs can also be the familiar purchase or subscription fees, but also regular and unplanned maintenance and updates training costs of personnel. Some of these expenses can be difficult to handle, especially for small organizations or organizations that have little funding. Further, such AI implementations require upgrades/modifications on the existing system in terms of infrastructure as well as compatibility issues. To overcome the financial challenges, it is necessary to plan the expenses, look for the less costly options, and show that the usage of AI systems will result in more effective utilization of time and money in future.
- **Data Quality Issues:** The strength of AI tools comes with the kind of data fed into the tools and the range of data available. Using low-quality or biased data may bring about erroneous conclusions and, therefore, undermine AI recommendations. That is, if an AI model is trained with inadequate or biased data, the generated recommendations can be very much off from the recommended use cases, which can then have implications for code quality or bug finding. It is, therefore, very important to make sure that the training data used for AI tools are very inclusive and of high quality. This may require spending on data acquisition tools, data cleansing, and, now and then, data validation processes.

Table 3: Challenges in AI Integration

Challenge	Description	Impact
Developer Resistance	Unfamiliarity and reluctance to adopt new tools	Slows down adoption and reduces the effectiveness
Financial Hurdles	High costs of acquisition and maintenance	Increases implementation costs
Data Quality Issues	Dependency on the quality of training data	Affects accuracy and reliability

5. Conclusion

Artificial intelligence and automation are looked at as revolutionary tools that drastically change the existing software development paradigm and provide significant advantages at any phase of the software design and development process. Combined with AI tools, there has been a fascinating potential to improve code generation procedures, improve the testing process, and increase the performance of DevOps. For example, GitHub Copilot greatly assists in coding by automating simple tasks as well as writing basic templates of code so that developers can focus more on specific and innovative regions of coding and development. In the same way, automated testing tools and platforms like DeepCode enhance the quality and reliability of software since it facilitate the identification of bugs and offers more insights into the code's weaknesses. These enhancements not only minimize the time taken to develop software products but also improve the quality of such products.

However the process of attaining the use of AI for generality in the software development process comes with several difficulties. One of the major challenges for the implementation of the strategy is the lack of support from developers since they may not trust new technologies or are not familiar with them. That is why, at its extreme, resistance can block the integration process and reduce the efficiency of AI tools. Therefore, to overcome this challenge, institutions need to develop extensive training structures and incorporate practical experience with AI technologies into their practices, as well as to show the extent of the positive effects of such technologies. One can also eliminate barriers related to conflict and gain consensus by using feedback from the developers involved during the early stages of the adoption process.

Time limitation is another major challenge and financial restriction is another major issue. Challenges like the cost of acquiring the tools, the costs of implementing and maintaining the tools and the cost of integrating them into the organization can be expensive, especially for organizations that are not well endowed. Due to the focus on higher initial outlays and constant costs associated with updates and maintenance, actual benefits from the introduction of AI

instruments should be searched for in the increase in productivity and numerous advantages associated with their use in the long term. These financial problems need to be solved in a proper manner, and organizations need to find ways to minimize costs and show how these tools increase efficiency and productivity resulting in faster time-to-market. However, most AI tools' usefulness is highly dependent on the quality of data that are fed into the AI for training purposes. Compilation of low-quality or even procured biased information may cause the effectuation of wrong outcomes and reduced effectiveness of AI-generated solutions. To solve this problem, there should be a continuation of special attention being paid to data quality for training the AI model so that the information being used is vast and diverse. It is also important to note that other aspects, such as periodic checking on the sources of data and refreshing of the database are also critical on the issue of reliability of the AI tools.

In particular, further incorporation of AI in software engineering has been projected to rise in the future, and future integrated tools will be smarter and more responsive. Machine learning will soon improve, and natural language processing and other AI technologies will improve, so there will be better tools that are more helpful. Some of the challenges today may well be solved by these future developments, which are expected to enhance the greatness of the AI tools in terms of affordability, viability and credibility.

References:

1. Dijkstra, E. W. (1968). The structure of the "THE"-multiprogramming system. CACM, 11(5), 341-346.
2. Fowler, M. (2012). Patterns of enterprise application architecture. Addison-Wesley.
3. Chomsky, N. (2014). The minimalist program. MIT Press.
4. Beizer, B. (2003). Software testing techniques. dreamtech Press.
5. Myers, G. J. (2006). The art of software testing. John Wiley & Sons.
6. Humble, J., & Farley, D. (2010). Continuous delivery: reliable software releases through build, test, and deployment automation. Pearson Education.
7. Duvall, P. M., Matyas, S., & Glover, A. (2007). Continuous integration: improving software quality and reducing risk. Pearson Education.
8. Bourbakis, N. G. (Ed.). (1998). Artificial intelligence and automation (Vol. 3). World Scientific.
9. Maruping, L. M., & Matook, S. (2020). The evolution of software development orchestration: current state and an agenda for future research. European Journal of Information Systems, 29(5), 443-457.
10. Malhotra, R., Bahl, L., Sehgal, S., & Priya, P. (2017, March). Empirical comparison of machine learning algorithms for bug prediction in open source software. In *2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC)* (pp. 40-45). IEEE.
11. Mohammad, S. M. (2018). Streamlining DevOps automation for Cloud applications. International Journal of Creative Research Thoughts (IJCRT), ISSN, 2320-2882.
12. Shaw, J., Rudzicz, F., Jamieson, T., & Goldfarb, A. (2019). Artificial intelligence and the implementation challenge. Journal of medical Internet research, 21(7), e13659.
13. Bera, K., Schalper, K. A., Rimm, D. L., Velcheti, V., & Madabhushi, A. (2019). Artificial intelligence in digital pathology—new tools for diagnosis and precision oncology. Nature reviews Clinical oncology, 16(11), 703-715.
14. de Barros Sampaio, S. C., Barros, E. A., de Aquino, G. S., e Silva, M. J. C., & de Lemos Meira, S. R. (2010, August). A review of productivity factors and strategies on software development. In 2010, fifth International Conference on software engineering advances (pp. 196-204). IEEE.
15. Ahmed, A., Ahmad, S., Ehsan, N., Mirza, E., & Sarwar, S. Z. (2010, June). Agile software development: Impact on productivity and quality. In 2010 IEEE International Conference on Management of Innovation & Technology (pp. 287-291). IEEE.
16. Lavazza, L., Morasca, S., & Tosi, D. (2018). An empirical study on the factors affecting software development productivity. E-Informatica Software Engineering Journal, 12(1), 27-49.
17. Sudhakar, G., Farooq, A., & Patnaik, S. (2012). Measuring productivity of software development teams. Serbian Journal of Management, 7(1), 65-75.
18. Macarthy, R. W., & Bass, J. M. (2020, August). An empirical taxonomy of DevOps in practice. In 2020 46th euromicro conference on software engineering and advanced applications (seaa) (pp. 221-228). IEEE.
19. Hourani, H., Hammad, A., & Lafi, M. (2019, April). The impact of artificial intelligence on software testing. In 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT) (pp. 565-570). IEEE.

20. Khaliq, Z., Farooq, S. U., & Khan, D. A. (2022). Artificial intelligence in software testing: Impact, problems, challenges and prospect. arXiv preprint arXiv:2201.05371.

Author Bio:

Gaurav Shekhar is working as Vice President - Technology in the 5th largest bank in USA. With over a 18+ years in the software realm, he has honed a passion for melding the art of Managing/Architecting and coding with the precision of cloud technologies, data science, and machine learning. He is not just an Architect or Engineer; but a strategist who bridges the gap between complex technical processes and tangible business outcomes.

Throughout his journey, he has steered multifaceted projects from ideation to fruition, consistently exceeding both timelines and expectations. His penchant for innovation is evident in the architectures envisioned and the culture of standardization and automation that he has championed. He is skilled in directing technical projects from start to end, preparing and executing strategic plans and control structures for projects, and ensuring successful completion within time. He has been recognized for innovative solutions, architecture vision, and inner-sourcing/open-source standardization and automation. Navigating the entire lifecycle of software development, from requirement analysis to maintenance, is second nature to him. No industry is foreign, and no challenge is too daunting to him.

EV Charging in a Condo

So… You Want to Install Electric Vehicle Charging in Your Condominium… A Case Study in Chicago

Sid Bennett, Consulting Inventor, sidbennett82@gmail.com, October 6, 2024

FEAR OF RUNNING OUT

Disclaimer

This report is based on an actual “EV Make-Ready” project in the City of Chicago and is based on available information at the time of writing and on the author’s experience and opinions. The information contained herein is subject to change as the environment is dynamic and subject to global events, politics, technology innovations and other factors. Each project requires a separate analysis and design by qualified consultants, contractors and engineers.

The author disclaims any liability for any personal injury, property, or other damages of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly, or indirectly resulting from the use of or reliance on this document and makes no guarantee or warranty as to the accuracy, completeness or usefulness of any information presented herein.

AI Use Restriction

Any use of the information or drawings in this paper by any form of artificial intelligence is not authorized. The paper may be freely distributed in its entirety, including this statement.

Preface

This paper has been prepared to share the insights, documents and other information associated with an “EV Make-Ready” Infrastructure project nearing completion at a 26-year-old mid-rise condominium in Chicago with an enclosed garage. The information is believed to be factually correct but is not intended as either engineering or legal advice. Rather, it may be a starting point for a condo board considering such a project and needs to include the engagement of qualified personnel to plan and execute the work, as each condominium has unique physical and legal attributes and financial aspects that may impact the project plan. The Summary of our project may be read without specialized knowledge and is followed by a narrative of the considerations that led to the project plan which could be helpful starting points for both the engineering and programmatic decisions that may be needed to initiate such a project. As you are all aware, free advice is worth every penny you paid for it.

Summary
The Whitney is a mid-rise building on Dearborn Parkway in the Gold Coast neighborhood of Chicago, completed in 1997, comprised of 83 units of varying sizes and having 136 limited-common-element (LCE) controlled-access parking spaces in three distinct indoor areas. The project concept seems most applicable to situations (use cases) where there are more than 6 parking spaces whose occupancy is restricted to individual owners and not easily transferrable independently of the living unit.

Before any significant effort was made to further the project a simple survey of the owners was conducted. A one-month response rate of about 30% percent indicated that there was substantial interest in the project and answered some questions regarding the potential EV adoption rates, but the most significant data was garnered from one question: "How many miles did you drive your car in the last year?" It is conventional wisdom that 80% of the re-charging of private cars will be residential, and as the yearly milage includes trips, this would be an overestimate of the average energy demand. The demand for our building, which is in the heart of Chicago, was the equivalent of about 10 miles per day per car! For suburban buildings the demand would be greater, and a study done for the City of Toronto can be used to adapt this design.

Do the user survey. It eliminates speculation and enables concentration on the actual solution.
The essential attributes of our approach to making the entire garage "EV Ready" are:

Feature	Benefit	Comment
Plug and Play charging at every parking space	No need to manage your charging individually	All monitoring and control functions performed without user interaction
All cars re-charged overnight for average daily usage	Efficient use of building power resources	We all need sleep : -)
Flat charge of $25/month/space (starts when charger is installed)	Eliminate costs and mark-up for managed billing (actual average electrical costs about $15/month/car)	You can choose any charging network you want for on-the-road charging. No network needed for local charging.
208vac junction box installed at each parking space "EV Ready"	Install EV charger only when you need it	EV adoption will be gradual, but infrastructure needs to anticipate need without rebuilding
Groups of 6 chargers on a 60-ampere branch circuit	Substantially reduce infrastructure cost	Power sharing makes for efficient system design

Adoption of EVs (including BEV, PHEV-plug-in-hybrid) is an evolutionary process, so that there is expected to be a gradual increase in the need for **E**lectric **V**ehicle **S**ervice **E**quipment (EVSE)—also known as a vehicle "charger"—over an extended period: perhaps 10 or more years. Which parking spaces will be electrified and when this would occur is effectively unknown at the outset and as it is dependent on individual owner decisions. Except for very recently constructed buildings, there is no provision for electrical power at individual parking spaces.

Anecdotally there is a belief that the utility power capacity to supply the EV charging load may be inadequate in existing buildings and require substantial infrastructure work, cost and planning time. We have found a way.

In practice, residential EVSE are of the Level 2 class (L2), which typically provide up to about 10 kilowatts (kW) of power to each charging connector from the building common element power panel (typically used for elevator, air conditioning, circulating pumps, and the like). Infrastructure design is governed by the National Electrical Code, which is an evolutionary standard that is generally adopted by local or state governments as the basis for issuing construction permits and inspection of installations. The most relevant aspect of the code is that the circuits cannot supply more than 80% of

the current rating of the protective circuit breaker, and that the other elements of the system be similarly sized. (A 60-ampere circuit breaker can supply 48 amperes or 9.6 kW of power).

L2 chargers are mainly intended to be used in residential properties for overnight charging (sometimes called “destination charging” when used in publicly accessible locations such as hotels and shopping malls) and may not have any capability of being networked or otherwise managed, depending on the manufacturer. So, if each L2 charger of the 136 potential EVSE of our project were to be connected individually to the building common power supply, the electrical code would require a capability of a maximum demand power demand of (10kW x 136 spaces/0.8) = 1.7 Megawatts (1700kW)! Very few building power panels indeed can supply such an additional load.

More recently, EVSEs from a several manufacturers have become available that are “networkable” and can be coordinated to dynamically share the power amongst a group of EVSEs so that the total current on a common electrical circuit does not exceed a specified maximum (e.g., 48 amperes). In recent years, the electrical code has slowly been amended to clarify the requirements for such installations, since previous interpretations tended to be inconsistent and unduly restrictive.

The number of EVSE that can be usefully operated in a networked fashion depends on the use case and has a significant impact on the infrastructure cost. This requires an engineering tradeoff between several constraints:
Each electric vehicle needs to be charged during an overnight charging session to at least replace the energy expended in an average day of driving;

When the power is being shared amongst a plurality of vehicles, the minimum current supplied should not be less than about 8 amperes; and the location of the chargers of each group of networked chargers should permit communication between the individual chargers or a networking connection to effectively manage a group of chargers automatically and in real time.

Table 1 shows a comparison of the required utility power supply capacity for EVSE groupings of various sizes. As described herein, a configuration of six (6) L2 EVSE, each group of 6 connected to a 60-ampere circuit breaker, achieves substantial infrastructure cost savings including the reduction of the maximum demand load for 136 parking spaces to (1.7MW/6) = 283kW.

This configuration will reliably re-charge the vehicle overnight.

Optimization Study (150 spaces)

# EVSE per group	# of EVSE Groups, N (branch circuits)	Minimum power per charger, P (kW)	Total Power Capacity Required N xP (kW)	8 hour charging Capability (miles)	D (Total Wiring length), ft
1	150	10	1500	240	11250
2	75	5	750	120	6414
3	50	3.3	500	79	5100
4	38	2.5	380	60	4389
5	30	2.0	300	48	3870
6	25	1.7	250	41	3583
7	22	1.4	220	34	3380
8	19	1.2	190	30	3320
9	17	1.1	170	26	3093
10	15	1.0	150	24	3015

EVSE Maximum Power: 10kW
Branch Circuit Breaker Rating: 60 amperes (48 amps maximum current)
Operating Voltage: 208VAC, single phase
1kWh of stored power results in a 3-mile driving distance

Table 1. Total installed electrical capacity and minimum overnight individual vehicle charging performance as a function of the number of vehicles in a power sharing group.

The study also included an example installation to provide an estimate of the amount of electrical conduit required for each configuration as this is a significant component of the project cost. We determined that a system comprising groups of 6 EVSE on a single 60- ampere branch circuit was the most advantageous from a cost and management perspective.

The EV-Ready charging system for our use case may be considered to be comprised of 3 integrated subsystems: the "central infrastructure" comprising an interface with the utility power, conversion of utility voltages to 208VAC and distribution to a number of panel boards (circuit breaker boxes) located to facilitate further distribution of power to individual parking spaces; the "local distribution" network of branch circuits extending from the panel boards to the parking spaces and having the ability to connect easily to a subsequently installed EVSE; and, the installed EVSE and any communications capability needed to manage the power sharing for each group of EVSEs.

Perhaps most daunting hurdle is the provision that is often found in the declaration document of a condominium requiring that a capital improvement cost of a project of substantial size be approved by a 2/3 positive ballot of all of the owners.

As the feasibility of the project is strongly influenced by the installation cost of the EV charging system infrastructure, we attempted to independently estimate the cost of each subsystem and major components. EVSE manufacturers generally do not offer infrastructure construction services but do work with local electrical contractors. At the time we began serious work on the project, there were precious few of them in Chicago and most were frustrated by the tedious business of dealing with condo boards and management companies. For the design which we discuss in detail here, we attempted to solicit bids for a system specification, but no one was interested in bidding. We then had further discussions with some of the contractors to address the central infrastructure and the branch circuit distribution portions and made some progress in obtaining budgetary estimates and one-page proposals. From this we were able to refine our cost model and better understand the perceived issues of the potential contractors.

During this process, we engaged a licensed professional electrical engineer (PE) to prepare a one-line electrical drawing of a system meeting our needs, from which a narrative statement of work (SOW) and specification was written. At this juncture, we could estimate the project cost and were able to solicit substantially compliant bids from two contractors, subject to further negotiation once the project was approved by the owners.

Our proposed project budget was $275,000 for the central infrastructure and a distribution network for 136 parking spaces in three distinct physical areas of the building ($2000/space). This does not include the cost of the EVSEs and connection to the local junction box at the parking space (about $1200 each). Included additional expenses were expected to be incurred for an "Owners Representative" to coordinate the project with building management, to provide an independent source of advice on potential change orders and schedule adjustments and to provide periodic reports on progress to the board; legal expenses were expected for contract review and development of a license agreement to control access to the system.

So, for a cost of about $2000 per parking space, the entire parking garage would be "EV Ready". Only when an owner required an EVSE at a parking space would one be installed and connected to the pre-positioned infrastructure. We also estimated the additional cost of a Wi-Fi system in the garage, if needed for power sharing management. But the question remained, who pays for it? And this question may need to be resolved individually for each condominium project depending on the financial condition of the association and the opinions of the owners.

Our proposal to the owners was that the condo association pay for the central infrastructure as it is an upgrade essential to modernize the building, that the condo association be reimbursed for a proportionate share of the branch circuit distribution network cost by the individual owners only when they chose to have management install an EVSE at a specific parking space, and that the individual owner pays for cost of the EVSE and connection to the infrastructure at the time of its installation.

Two Zoom presentation sessions explained the project, its need, its scope, the proposed funding approach, and some of the practical aspects of the implementation and operation. In the summer of 2023, a ballot of the owners was conducted, and the required 2/3 approval was achieved by the time of the September 2023 annual meeting. The main contract was let in December 2023 and the on-site construction began in June 2024 with completion scheduled for October 2024, including some procurement delays related to the continuing supply-chain problems in the electrical industry.

Project Description and Design Considerations

Introduction

The above summary was intended to provide a broad overview of the project being executed, but there were many details to explore during the gestation period, and a discussion of some individual topics may aid others in determining if such a project is feasible for their condominium. One size clearly does not fit all, but there may be sufficient underlying similarities such that our studies may be a helpful starting point. Nothing can substitute for doing your own research, engaging consultants and talking to other buildings considering such a project. To the extent that we have provided examples of specifications, design computations and related documentation, none should be used without review and adaptation by competent individuals.

Design Considerations

Rules of Thumb

There are a few key factors to be considered in the design of the project and since some of the numerical values and terminology are often a subject of confusion, we present a short discussion of rules of thumb. In accordance with the electrical code, the maximum current that a circuit can supply is 80% of the circuit breaker rating. For a 60-ampere circuit breaker, the maximum current is 48 amperes (equivalent to 9.6 kW of power). To achieve this power level, the equipment must be hard-wired (directly connected: not a plug and socket) to the electrical circuit and have a 60 ampere circuit breaker.

Present generation EVs (BEVs and PHEVs) can travel about 3 miles per kWh (kilowatt-hour) of battery power. The maximum travel distance in miles is about 3 times the manufacturer's stated kWh capacity of the car battery but varies sufficiently, including with the ambient temperature, that you need to use the car's real-time estimate of remaining milage when you are driving.

The technical term for the wall-mounted unit having a power cable and connector to connect to a vehicle is "Electric Vehicle Service Equipment" (EVSE), but equipment such as the Wallbox Pulsar Plus, Tesla Gen3 Universal Wall Connector are commonly called "chargers". They are actually "control units" that receive an electrical signal from the car over the charging power cable indicating the amount of current that the car desires, and the control unit sends a signal to the car indicating the amount of current that the actual charger in the car can demand from the EVSE. Typically, the current requirement changes during the charging process. For the sake of functional clarity, the EVSE and "charger" are synonymous.

The L2 EVSE (connected to 208VAC) delivers up to 9.6 kW of power to the car through a cable and connector. Note that the power capability is less power capacity than the same EVSE connected to a home residential voltage of 240VAC (11.5 kW).

The output of the EVSE is described as either in amperes (amps) or power (kW); amps x 208volts = power in watts. Often when talking to an electrician, the term kVA (kilo-volt-amperes) is used. For practical purposes in this use case kVA and kW are the same numerical value.

A single EVSE connected to a single electric vehicle would transfer enough electrical power to increase the range of the vehicle by about (3 miles/kWh) x 10kW = 30 miles of range per hour of charging time.

Table 1, above, showed the minimum milage added when 6 EVSE are supplying power to connected vehicles from a single branch circuit. A typical personal EV spends at least an average of 8 hours per day parked in its parking space (41 miles). The vehicle could replenish the energy needed to go (30 x8) = 240 miles overnight if connected to a dedicated charger!

Use Cases

The applicability of our approach to other condominium situations depends on whether the use case has a similar or scalable relationship to the considerations that drove our design. City-center condominiums usually have enclosed garages and are comprised of deeded parking spaces, limited-common-element parking spaces (LCE) or assigned spaces. Some buildings have valet parking spaces as well. Only in the case where there is open parking or valet parking is it likely that complete groups of power-shared EVSE can be installed at once or in a planned manner. In our use case, the EVSE will need to be installed at the owner's individual space at the time the owner desires it.

An extended and uncertain deployment schedule is one of the significant problems faced in determining the economic feasibility of providing this amenity. Constructing something as a whole is obviously considerably cheaper than proceeding incrementally, but the up-front cost is often a stumbling block, since the bulk of the infrastructure installation may be underutilized for years. We have taken the intermediate approach where the entire garage is being made “EV Ready”, where the needed 208VAC power is wired to a junction box at each of the individual parking spaces. The final step is to install the EVSE at the parking space when it is desired.

Our use case is for 136 parking spaces; for other use cases, the initial design should be sized such that the eventual electrical demand is known to avoid later surprises.

Sizing the electrical demand and electrical wiring

From the above rules-of-thumb it is apparent that the average number of miles a day driven by the cars in the garage is a key factor in the design of the system. According to our survey of our owners, the average milage driven is 10 miles per day. Since the actual milage varies for each owner, this average is useful only in the aggregate and there is an uncertainty in the probability that each car will be fully charged overnight. We found a detailed study of this issue, done by a certified actuary for the case of Toronto, Canada, a city with environmental conditions similar to Chicago. A brief review of the paper entitled “Statistical Modelling of Load-Managed Charging for Electric Vehicles in Multi-Unit Residential Parking” (ref.1) will enable better appreciation of the factors that should be considered in sizing the capabilities of the system, including the variability of daily driving distances.

Our simulations show that, compared to dedicated circuits, shared circuits have the potential to dramatically reduce unused electrical capacity, while still providing excellent success rates for overnight charging in representative use cases. Roughly, a charging rate of between 1 and 2 kW will more than adequately recharge an average vehicle overnight for daily use. Due to the variability of daily usage, a design where each EVSE serviced a single vehicle, the capacity of each EVSE circuit would need to accommodate overnight full battery charges (at 48 amperes), whereas when the capacity of an EVSE circuit (say, 9.6 kW) is dynamically shared amongst a group of vehicles, the diversity of driving and charging patterns substantially reduces the average charging rate required.

Does that mean that 10 or more EVSE can be connected to a single 60 ampere circuit? In practice, no, as there are other considerations, particularly the minimum charging current of a car connected to an L2 charger, which is specified as 6 amperes (ref. 2). For conservatism in setting the lower current limit when EVSE are operated in a power-sharing mode, we used 6 EVSE as the standard group which will always supply a minimum of 8 amperes to each EVSE when all the EVSE are connected to charging vehicles.

How does this affect the installed capacity and the cost of the branch circuit distribution network? If the current of the branch circuit is not shared, then each branch circuit is rated at the circuit breaker value, and if the EVSE can supply 10kW of power, then that is the circuit capacity needed. For 6 EVSE, operating independently, this means a 60kW installed capacity, little of which will ever be used on average. But, when the 10kW is shared between chargers of a group, the installed capacity needed is only 10kW! (see Table 1, above) per group. This modular configuration can be replicated to give the results for your use case. Depending on the capacity of the utility connection to your building, just this aspect could determine the project economic feasibility.

The electrical wiring between the building central power switchboard and each EVSE is a significant expense since much of the installation cost is skilled labor. There is also a certain inequity in the cost due to the location of each of the parking spaces, depending on the length of the conduit run and whether the conduit needs to traverse concrete floors and walls. Much of this expense is for the portion of the conduit run that terminates at an electrical junction box in the vicinity of the group of 6 EVSE. The economics of this factor as a function of the number of EVSE in a group is shown in Table 1 (right-hand column), which is an extract from the detailed study for our project. The biggest improvements are for groupings of greater than 1 and less than 6 or 8, and they are significant.

Power Sharing Approaches

While the electrical code limits the maximum power supplied to an L2 EVSE to 48 amperes with a 60 ampere-rated branch circuit, there is also a provision that permits a group of EVSE units to be connected to such a branch circuit so long as the total current supplied to the group of EVSE units is AUTOMATICALLY limited to a total of 48 amperes, This is the basis for power sharing systems and is an essential consideration for most multifamily dwellings.

To do power sharing, the EVSE in the group need to communicate (wired or wirelessly) frequently to determine how much current should be supplied to each EVSE depending on the preset requirements and state-of-charge (SOC) of the

connected vehicles. This demand profile may be updated periodically to account for new connections, completed charging or changes in requested current by the individual EVSE. There are as many different approaches as there are EVSE manufacturers and most of them use proprietary software. Moreover, in practice, the same model EVSE needs to be used in each group for the devices to interoperate. Even though there is a de-facto standard for communications with an EVSE (ref. 3), this is mostly applicable to the public charging networks or when individually billing each owner for power consumption. Careful attention needs to be paid to the recurring operating costs associated with Internet (cloud)-based power sharing and management systems, particularly those which include individual energy consumption billing (see "How Much Does it Cost to Operate?") as these are recurring overhead expenses.

Hard-wired (data cable) communication between the EVSE is very reliable but, in Chicago, the communications cables need to be in metal conduit and may only be attractive when individual groups of 6 EVSE are installed together as part of a project. Otherwise, substantial costs may arise from continual reconfiguration as EVSE are added in a sparse manner.
Wi-Fi connectivity comes in various arrangements: connected from a garage-wide Wi-Fi system to a back-end server in the cloud over an internet connection, or over a cellular radio; or, local Wi-Fi sub-networks limited to each group (with a possibility of Internet connectivity for ancillary services). Almost all data communications can take place locally, except perhaps for billing, so the most reliable operation results when the bulk of the communications links are local to the garage or to the EVSE group.

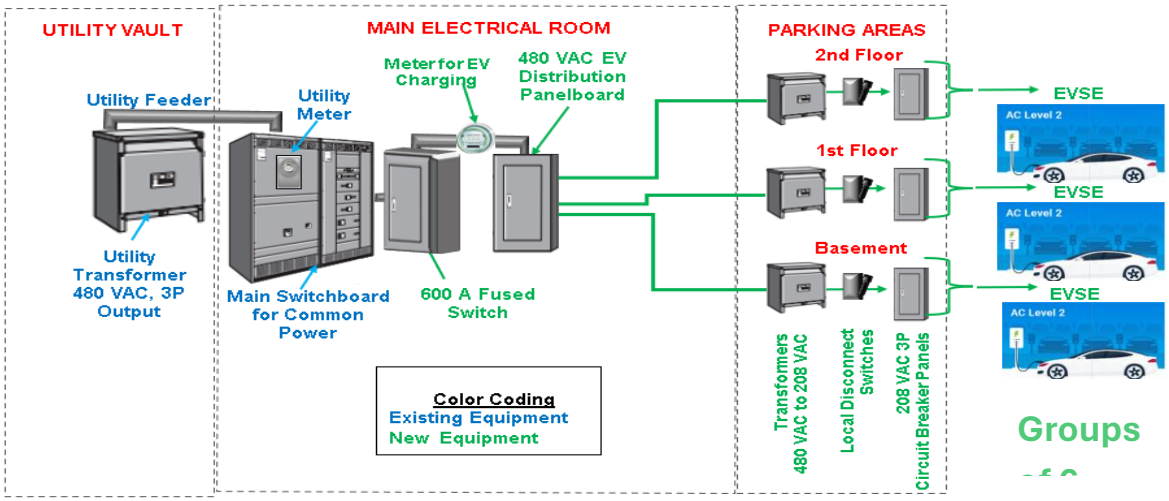
Some users report difficulty with the Wi-Fi approach with external (internet) connectivity as there is a diffuse allocation of responsibility to operators of the back-end cloud server, the Internet and the local Wi-Fi or cellular connection. It is often not possible to find a single entity to be willing to undertake the management of such a configuration.
There is at least one vendor offering a local-to-the-EVSE-group Wi-Fi connectivity solution for power sharing and this is what we have chosen. Should ancillary services (such as billing or time-of-use (TOU) electricity pricing) be needed in the future, external Wi-Fi connectivity can be installed.

Third-party vendors are beginning to offer billing and management services.

How Much Does It Cost to Operate and How Should We Charge Users?
What is the raw cost of electricity in your area? In Chicago, it is now about $0.15 per kWh (2024) and is not priced on a TOU basis. For our case where each user, on average, drives about 10 miles per day, needing 3.3 kWh to recharge, the total cost of electricity per month is 3.3 x 0.15 x 30= $14.85. Any other costs associated are additional. Initially we set a fixed charge of $25/month per space, leaving a reserve for incidental expenses such as repairs. Even with the potential disparity in usage by individual owners, this does not seem to have been an issue. In a suburban use case, this may lead to a different conclusion.

System Overview

Electrical Infrastructure
Our use case provides electrical power at each designated parking space sufficient to re-charge an electric vehicle overnight. Both BEV (battery electric vehicles) and PHEV (plug-in hybrid vehicles) are accommodated. The electrical infrastructure is sized to support multiple groups of up to 6 EVSE, which are located to permit wireless communication with and between the individual EVSE of a group. Each EVSE group is locally controlled to operate in a power-sharing mode where the 9.6 kW supply capacity of the group is automatically shared between the connected vehicles to perform the recharging function.



graphic credit: Tim Milburn

Fig. 1 is block diagram showing in outline of the entire system concept.

Power is received from the electrical utility company, typically at 480 VAC, three phase, delta connected energy, but the EVSE operates at 208VAC, single-phase. The conversion is performed by one or more transformers that are part of the new electrical infrastructure so that the appropriate voltage is distributed from local panel boards (circuit- breaker boxes) within the garage to the individual branch circuits, each single-phase circuit being protected by a 60-ampere circuit breaker.

Since our use case does not currently envisage cost recovery for individual vehicle power consumption, a single non-revenue power meter is provided at the input to the EV charging infrastructure to measure the total power consumption so that it may be distinguished from the remainder of the building common-element power consumption. This will be used to determine yearly adjustments in the monthly charge.

Each branch circuit can supply 48 amperes continuously; the branch circuit wiring connects from the local power panel to a junction box in proximity to each EVSE group, and to an individual junction box at each parking space. The actual routing and wire sizing of the branch circuit is based on electrical design principles to account for resistive voltage drops and current carrying capacity of the wiring. The infrastructure, extending from the electrical utility company to the individual parking spaces uses standardized components and well-established construction techniques.

In our installation, typical transformer sizes range from 75kVA to 112.5 kVA, and the transformers are located near to the circuit breaker panels to minimize electrical circuit losses.

Each garage presents a unique physical layout case and may be designed using an architectural plan-view drawing of the garage. A section of such a layout is shown in Fig. 2 and the EVSE groups are situated to achieve line-of-sight Wi-Fi communication paths between the EVSE of a group. Depending on the specific EVSE manufacturer and communication technique used there may be other considerations. But for our selected EVSE, the requirement is that at least one EVSE should have a line-of-sight view of all the other EVSEs of a group.

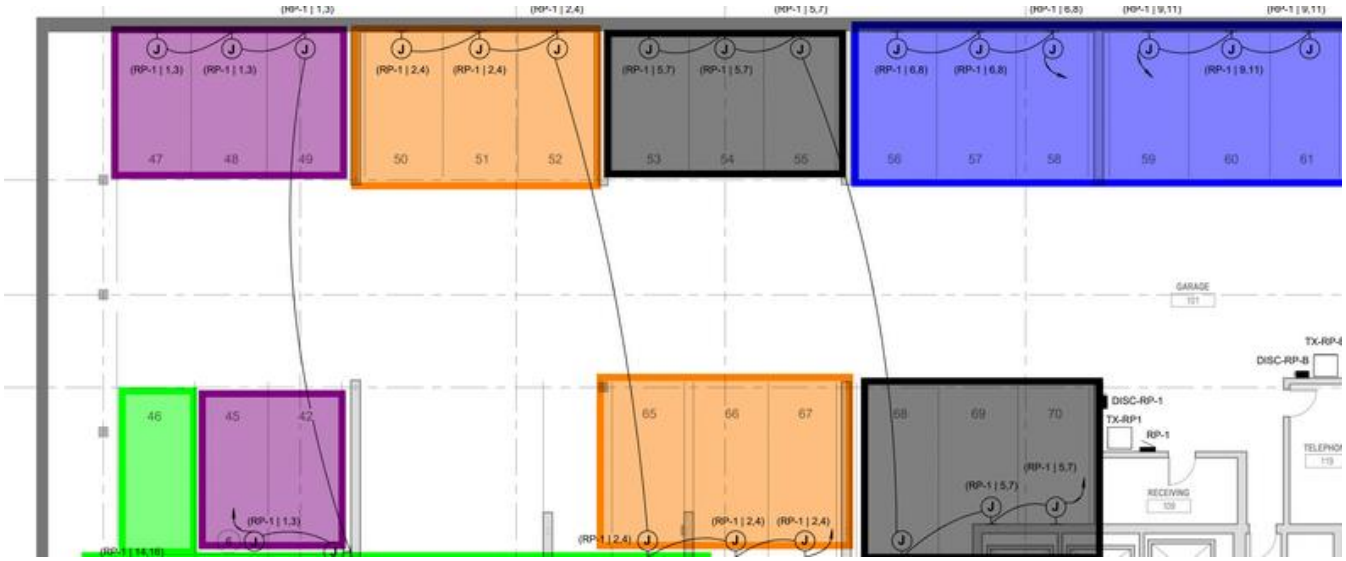


Fig. 2 Example of actual layout of several groups of EVSE chargers in garage.

At the outset our infrastructure will be underutilized and only about 10% of the parking spaces will have EVSE installed. Some of the groups will only have a few EVSE and some will have none. As the adoption of EVs proceeds, individual EVSE may be easily added to a group by an EV-qualified licensed electrician. The branch circuit breaker would be deenergized and an EVSE mounted at the parking spaces and hard-wired (connected without a plug and socket connection) to the electrical infrastructure. Once this is completed, the circuit is reenergized and the electrician "commissions" the group of EVSE (in our case, up to 6) for power sharing. This setting is in accordance with the manufacturer's instructions so that the group of EVSE operates in conformance with the electrical code. The user cannot modify the settings.

Selection of EVSE

At present there are a large number of manufacturers in the marketplace, each seeking to establish their product position. The standards for connection between the EVSE and the vehicle and the EVSE and the electrical infrastructure are now established technologies, and further improvements are likely to be backwards compatible with existing installations.

However, power sharing requires interoperability of the EVSE of different vendors products on a hardware software basis, so a power a sharing application may need to utilize the same vendor and product style for all the EVSE in a group. From a practical point of view, this means selecting a single EVSE model for all installations (at least at a group level).

There will be attrition in the marketplace and no selection of EVSE vendor is totally safe, but the modular nature of the EVSE groups may permit consolidation of existing chargers in groups.

Perhaps more problematical is the selection of the communications system as, despite industry standards, implementations are different, and the performance of each implementation is difficult to assess from documentation. From a system perspective, the fewer concatenated communications paths the better. So, the most reliable approach should be linking the EVSE group locally using a wired connection. Next would be a local-to-the-group Wi-Fi network. Approaches using external communications links and the Internet or cellular radio are likely to be less satisfactory. Before choosing an EVSE and communications approach, consultation with actual users regarding already installed approaches is highly desirable.

Future Proofing

Technology evolves and we should be concerned about obsolescence of design for the charging system. Our approach considers some of the factors that might be significant.

- Car Efficiency and Range
- Sufficiency of Utility Power
- EV Owner Usage Patterns
- Power Sharing and Payment for Use

At present, manufacturers are focused on increasing the range of the vehicle and reducing the cost. Focus on aerodynamic design and regenerative braking may further increase the range (miles/kWh) but there are natural limits to this. Providing that driving usage patterns remain similar to ICE (internal combustion engine) vehicles, the average power demand per vehicle on a daily basis will not meaningfully change.

The storage capacity of EV car batteries is increasing due to the substantial industry investment in research, and this would translate into an increased range. We expect driver charging behavior to evolve. Why charge your vehicle every night if your range is 350 miles and you only drive 10 miles per day? Less frequent charging does not reduce the average power consumption, but each less frequent charging session will require more power. Providing that the electrical infrastructure is initially properly sized, this is not likely to be a problem. In the worst case, owners will realize that they ought to recharge somewhat more often if they require a fully charged battery. PHEV owners will likely charge each night or so.

The performance of L2 chargers is bounded by the amount of power that can be supplied by a 60 ampere 208VAC circuit and as the actual charging technology is embodied in the car itself, any technological changes will be addressed by the vehicle manufacturer. But, the management of power sharing may very well become more sophisticated. Unless this is properly automated such improvements may not be very useful to the owners since the appeal of a residential system is simplicity. Just plug in the car.

Interface standards exist for the communications path and the vehicle connector standards have recently been resolved; NACS (aka Tesla connector, SAE J-3400) and the legacy SAE J-1772, and the utility and building infrastructure have been standardized for decades. There is a particular external communications standard, OCPP2.1 (ref. 3) that is likely to be adopted as a global standard; some L2 EVSE are already compatible, but it is not needed for local power-sharing control unless the power consumption of individual vehicles is billed at other than an average rate.

What we cannot effectively predict is how the utility power cost or the time-of-use (TOU) requirements will change in a particular geographical area. From a purely economic perspective, the electricity cost may rise to a level where a flat monthly fee is no longer seen to be equitable and individual billing may be needed. Some L2 chargers already have the technology to support this feature, but an external communication means (e.g., garage-wide Wi-Fi, cellular radio and Internet) may need to be added. The software may need to change, but many L2 chargers are capable of over-the-air software updates or already have the software needed. Scheduling electrical consumption for TOU pricing is something that the many existing external communications interfaces could do, and this may influence your choice of EVSE manufacturer.

Since the electrical infrastructure should continue to be adequate to supply the power needed, the only problem is the eventual obsolescence of the EVSE, primarily from a maintenance viewpoint. As the EVSE are managed in groups of 6, any upgrade, if needed, can be done incrementally on a group basis. Certainly, there will be improvements in functionality and communications with the individual cars and this may create demand for additional features, but any such features would be an expansion of the use case and treated as such.

Management, Operation and Maintenance

EV charging in our condominium garage is viewed as an amenity and every effort made to simplify the management, operation and maintenance of the installed system. As described, the functionality of our system approach is directed to the plug-and-play overnight charging of individual vehicles. Apart from plugging the charger cable into the car, no further manual action is required. Unplug the car in the morning.

Selection of a single EVSE type simplifies the maintenance as any EVSE can be used as a spare. Billing a fixed fee per month rather than a detailed charge for power consumption reduces administrative costs and avoids the need for garage-wide Wi-Fi or external communications and third-party services but may not be acceptable in all use cases.

Updating the software of the chargers is usually done by an over-the-air process like on-line computer systems; but since the need for such updates should not be frequent, this may be done as part of a routine maintenance activity.

All of this may seem straightforward, but transforming the approach into practice required the creation of a license document and condo rules to control the evolution and operation of the system. Each use case presents a different balance of design and management issues, but it is often possible to gain some insight into the process by reviewing examples of practical solutions. Think the problem through with your condo management company and legal advisor.

Acknowledgements
Dillon Buschmann, PE, of AMS Industries, Inc., Woodridge, IL, did the electrical design and was the project manager for the installation. Tim Milburn of Green Ways 2Go, Chicago, IL, was our owner's representative.

References

1. (https://www.researchgate.net/publication/341763845_Statistical_Modelling_of_Load-Managed_Charging_for_Electric_Vehicles_in_Multi-Unit_Residential_Parking?enrichId=rgreq-3ceb99ae69da93e87fd561e77c1072e7-XXX&enrichSource=Y292ZXJQYWdlOzM0MTc2Mzg0NTtBUzo4OTY5MDcyNDEwODI4ODBAMTU5MDg1MDc1ODYzOA%3D%3D&el=1_x_2&_esc=publicationCoverPdf)
2. IEC 61851-1
3. OCCP2.1

Author:
Sid Bennett, LSM
Third Millenium Medal
Former Chair of IEEE Gyro and Accelerometer Panel, AESS
Former member of Board of Governors of AESS and IEEE Standards Boar

LMAG Update

IEEE Life members R4

Jim Riess, LSMIEEE, is the newly appointed Chair of the Region 4 Life Members Committee that has two primary goals, one to help form new Section Life Member Affinity Groups (LMAGs) and the second to help develop programs of interest for life members.

IEEE Life Membership is automatically bestowed upon an active IEEE member who has attained the age of 65 years and has been a member of IEEE for such a period that the sum of their age and their years of membership equals or exceeds 100 years.

Life Members Affinity Groups retain active IEEE associations, contribute to the social good in their communities, advance the professional interests of IEEE, and allow members to enjoy each other's company. IEEE Life Members are the least utilized members in the IEEE.

There are nearly **1800** life members in Region 4 in 2024. IEEE Life Membership is an official recognition of a strong and sustained commitment to IEEE. Life Members groups participate in educational excursions, work together to mentor students, provide needed participation in the IEEE Section and improve their communities. IEEE Life Members are technology influencers, pioneers, and valuable partners – sharing over one million years of experience with the next generation of innovators.

A Life Members Affinity Group can easily be established by first finding an organizer. The organizer (a Life Member) can then obtain petition signatures of at least six (6) IEEE Life Members who are members of the Section(s) involved and who indicate they will join the affinity group if established. The organizer becomes interim Chair pending election of a regular Chair at a later organization meeting. The petition to establish an IEEE Life Members Affinity Group must be submitted and approved by the IEEE Life Members Committee Chair, Region Director, and Section Chair.

Each Section is encouraged to form a Life Member Affinity Group (LMAG). Please let Jim Riess j.riess@ieee.org know by the end of the year if you have any interest, questions or need assistance in establishing an LMAG in your Section.

Member News

One of our own was awarded Fellowship of the British Computer Society!

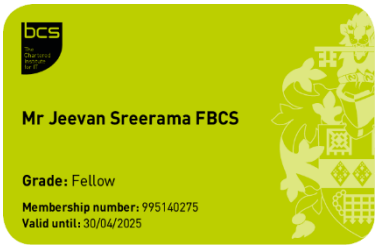
Jeevan Sreerama – FBCS, IEEE Senior Member, and AI Leader

Jeevan Sreerama, IEEE Senior Member, has made significant contributions to artificial intelligence, data science, and engineering over his 18-year career. Currently the Senior Data Scientist and Director of Artificial Intelligence at Soothsayer Analytics, USA, Jeevan has been instrumental in delivering innovative AI solutions across various industries, including healthcare, insurance, finance, retail, and manufacturing, enhancing efficiency and driving transformative results.

Jeevan holds a Master of Science in Information Technology, a Bachelor of Technology in Computer Science & Engineering, and a Post Graduate Certificate in Big Data Analytics and Optimization. His career began with roles such as Visiting Research Scholar at Carnegie Mellon University, Associate Mentor at IIIT Hyderabad, and Software Engineer at CA Technologies. He then served as Principal Data Scientist at INSOFE, where he was a leader, architect, consultant, and educator in AI and data science.

He has authored multiple scholarly articles focusing on AI-driven solutions for fraud detection, diagnostic accuracy, and customer insights, contributing to advancements in both academia and industry. His engineering contributions include projects like BetterBotAI, SmartDocAI, and AI-Powered Plant Efficiency, showcasing his expertise in addressing complex challenges through advanced AI methodologies. With deep proficiency in machine learning, deep learning, generative AI, NLP, computer vision, and MLSecOps, Jeevan continues to develop secure and scalable AI solutions that bridge academic innovation with real-world applications.

His recognition as a Fellow of the British Computer Society (FBCS) reflects his leadership and sustained impact on technology and AI innovation. He has also served as a judge for prestigious awards like the Globee® Awards and Stevie® Awards, evaluating innovations in technology, cybersecurity, and customer service. Additionally, he was honored with the title of Adjunct Lecturer at the University of Sydney for his collaboration with neurology PhD scholars, where he developed machine learning models to support groundbreaking research.



Jeevan's work exemplifies the integration of technical excellence, academic rigor, and industry relevance, solidifying his reputation as a thought leader and innovator in AI and technology.

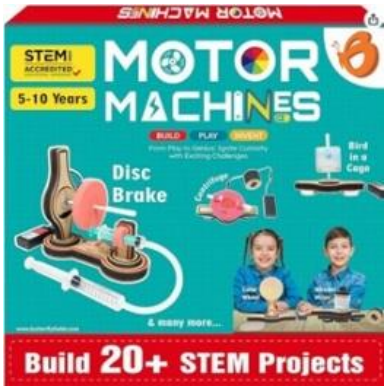
SKPL Report

Region 4 – Science Kits for Public Libraries program
Inspiring youth to consider Engineering careers

John Zulaski here to ask for your help with the 2025 "Call for SKPL Grant applications". The IEEE- *Region 4 Science Kits for Public Libraries (SKPL)* Grant program is again offering up to $2,000 in funding to public libraries located within Region 4 to enable them to build a circulating collection of science kits.

Do your part
Applications are now being accepted until **January 17, 2025**. Drop off a flyer at y**o**ur local library to make them aware of this Grant opportunity. To download the flyer, go to: https://r4.ieee.org/skpl/get-involved/ Public Libraries have a long tradition of building stronger communities by providing life-long learning for children and teens. Please take the opportunity to enrich the resources that your public library has to offer.

SKPL Impact
According to Joy Kyhn – Ravenna Public Library Director, "This is an amazing opportunity for our community. We are a rural, small, low-to-moderate income community and I try hard to provide opportunities that larger more funded libraries offer their patrons. This program has really impacted our community."



"The impact was tremendous in increasing our checkout numbers and also a new love for non-fiction books. It's harder to get the 10-year-olds and older to gravitate towards non-fiction books, but the science kits have done just that. Not only have the science kits benefited our general patron usage, but it has also helped us provide much needed equipment for our homeschooling families as well." The West family has checked out nearly all the science kits for their two girls. According to their mother, "The science kits in the library have granted us access to equipment we would have never been able to afford to purchase for our homeschool science lessons. Now the girls can get hands-on learning experiences!"

Orin K, age 12. Really liked the Motor Machines kit. This Science Kit has taught me about different types of motors." "I got to take this kit home to my Papa's house. He works on motors, and we worked on this kit together."

Want to know more? Go to **https://r4.ieee.org/skpl**

Help create more STEM teaching moments Donate

HTB Update

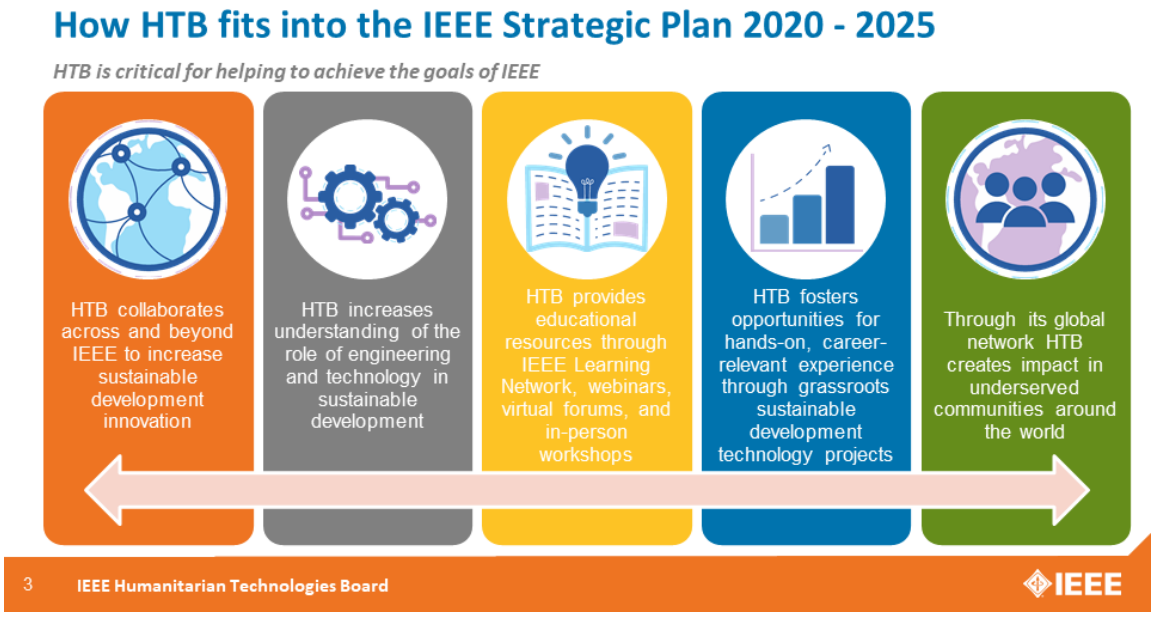
IEEE Humanitarian Technology Board Completes Second Year After 2022 Re-establishment

Bruce Howell, Region 4 Humanitarian Activities Coordinator

In November of 2022, the IEEE Board of Directors approved to re-establish the Humanitarian Activities Committee (HAC) as the IEEE Humanitarian Technologies Board (HTB). This newly re-established HTB is now finishing its second year under the leadership of the first Humanitarian Technologies Board Chair, Lwanga Herbert. During 2023-2024, the HTB completed several key objectives including authoring an Operations Manual in addition revamping the IEEE's Humanitarian web site, https://htb.ieee.org. Most of these HTB accomplishments are nicely summarized in the first ever Humanitarian Technologies Board Annual Report.

The new HTB operations manual includes Mission, Vision and Objectives statements as well as some key definitions. Possibly the most important definition is of Humanitarian Technology, which is stated as "Those IEEE programs and activities focused principally on applying science, engineering, and technology to satisfy the unaddressed social needs of specific communities which are not adequately served by existing government, commercial, or non-commercial services." Additionally, the IEEE HTB mission is "To support impactful and ethically informed volunteer-led initiatives, programs and projects, and mutually beneficial partnerships, as well as to inform policy formulation that harness technology and innovation to address societal challenges (including disaster recovery) in a responsive, effective, and sustainable way."

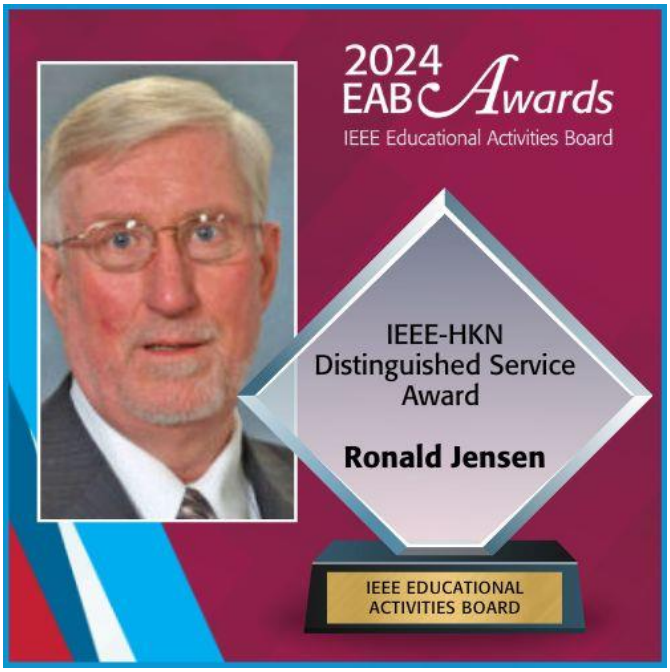
The elevation of HAC to the new board is commensurate with the growing numbers of SIGHT membership, project proposals, and funded teams, In addition to the 30% of all active IEEE members (and 60% of active IEEE student members) who indicate an interest in humanitarian programs at IEEE when they join the association or renew their annual membership. It also demonstrates the support of IEEE leaders, who have provided the structure to expand upon the significant achievements of HAC in its eleven years as an ad hoc committee and "standing" committee reporting to the IEEE Board of Directors. Humanitarian technology activities are important to IEEE members and are intrinsic to IEEE's identity and mission.



The predecessor to the HTB was the IEEE Humanitarian Activities Committee (HAC) which was first launched in 2016 as a committee of IEEE reporting to the IEEE Board of Directors (BoD). HAC was an outgrowth of the IEEE Humanitarian Ad Hoc Committee (HAHC) that was appointed at the end of 2011 by IEEE President Moshe Kam and finished in 2015 with the successful approval by the IEEE BoD of the HAC governance documents.

R4 EAB Award

IEEE Educational Activities Board (EAB) Awards recognize and honor major contributions in the development and implementation of educational programs and activities, either formal or otherwise, that relate directly to the discipline of electrical and electronics engineering. This includes, but not limited to, continuing education, pre-university guidance, accreditation, educational innovations, and private sector support of educational institutions or activities.



Congratulations to our very own Ronald Jenson for being awarded the HKN Distinguished Service Award. The Distinguished Service Award was initiated in 1971 to recognize those members who have devoted years of service to Eta Kappa Nu (or IEEE-HKN), resulting in significant benefits to all of the society's members. The award is based on lifetime contributions to Eta Kappa Nu (or IEEE-HKN) and is limited to one recipient each year.

IEC Update

Have an idea to share, requirement to engage with members in industries at IEEE Region 4 then why wait...? reach out now to your IEEE Region 4 one stop team of experts at r4-iec@ieee.org. We will make that collaboration possible and seamless for you. All it takes is one email to one ID!!

2024 Winners of R4 IEC Section Awards!!!

R4 IEC is pleased to acknowledge and recognize the work of the following sections for promoting industry themed in person and virtual events through their sections and local industry partners. **A big applause to these sections in being IEEE's shining lights showcasing the industries in our region!!**

- ✓ IEEE Northeastern Wisconsin Section
- ✓ IEEE Cedar Rapids Section
- ✓ IEEE Milwaukee Section
- ✓ IEEE Southeastern Michigan Section

IEEE Region 4 Industry Engagement Committee - Call for volunteers!!

Interested in leading the interactions and liaising between IEEE and fortune 500 companies, multiple micro, small and medium scale enterprises across industries in the mid-west? Then don't wait, join the R4 Industry Engagement Committee and be the crucial bridge connecting two critical cornerstones of engineering in the Midwest – IEEE & our industries. For more details regarding this excellent opportunity please contact: **r4-iec@ieee.org**

Nuclear Fusion Technologies new address - "America's Dairyland"

Yes, you read it right. ***Wisconsin*** has for years been the home ground for breakthrough technological advancements in the near impossible field of nuclear fusion technology. It's a trifecta effect pursued relentlessly over the years by coming together of academia, local industry and government agencies in the spirit of engage and excel. Curious to know more about what's this all about? Read on the below articles for an inside view of nuclear fusion technology and how it's evolving, shaping the future right at our own lovely IEEE Region 4-member state – Wisconsin!!!

- ✓ **How southern Wisconsin could become a nuclear fusion mecca**
  https://captimes.com/news/how-southern-wisconsin-could-become-a-nuclear-fusion-mecca/article_673a3740-8011-11ef-90b0-e79bb1399697.html
- ✓ **5 Reasons Wisconsin is Leading the Way for Fusion**
  https://www.shinefusion.com/blog/5-reasons-wisconsin-is-leading-the-way-for-fusion
- ✓ **UW-Madison one step closer to harnessing the power of the sun through fusion research**
  https://www.wpr.org/news/uw-madison-one-step-closer-to-harnessing-the-power-of-the-sun-through-fusion-research
- ✓ **Hungry for Energy, Amazon, Google and Microsoft Turn to Nuclear Power**
  https://www.nytimes.com/2024/10/16/business/energy-environment/amazon-google-microsoft-nuclear-energy.html

API Gateways

API Gateways: A Critical Component for API Success

In the rapidly evolving landscape of software development, APIs (Application Programming Interfaces) have become the backbone of modern applications. They enable different software systems to communicate and share data seamlessly. However, as the number of APIs grows, managing them efficiently becomes increasingly complex. This is where API gateways come into play. API gateways are a critical component for API success, providing a centralized entry point for managing, securing, and optimizing API traffic.

What is an API?

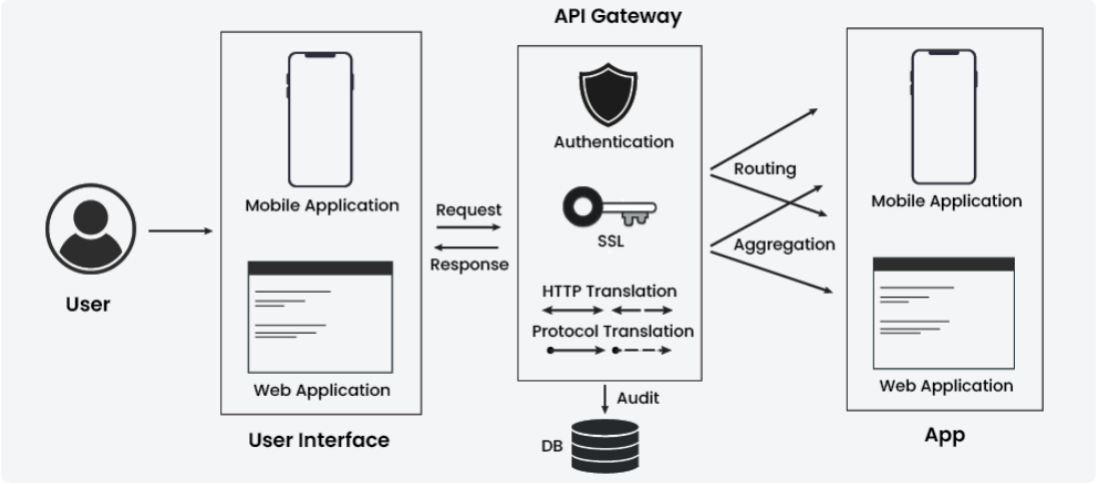
APIs, or Application Programming Interfaces, are essential tools in modern software development. They allow different software applications to communicate with each other, enabling the integration of various services and functionalities. APIs act as intermediaries that define the methods and data formats that applications can use to request and exchange information.

Why are APIs important?

1. **Interoperability**: APIs enable different systems and applications to work together, regardless of their underlying technologies. This interoperability is crucial for building complex, interconnected systems.
2. **Efficiency**: By providing predefined methods for accessing data and services, APIs streamline development processes. Developers can leverage existing APIs to add functionality to their applications without reinventing the wheel.
3. **Scalability**: APIs allow services to be scaled independently. For example, a mobile app can use an API to fetch data from a server, which can be scaled to handle increasing numbers of requests.
4. **Innovation**: APIs foster innovation by allowing developers to build on top of existing services. This modular approach encourages the creation of new applications and services that can interact seamlessly with others.

What is an API Gateway?

An API gateway acts as an intermediary between clients and backend services. It handles all incoming API requests, routes them to the appropriate services, and then returns the responses to the clients. Essentially, it serves as a single point of entry for all API interactions, simplifying the management of multiple APIs.

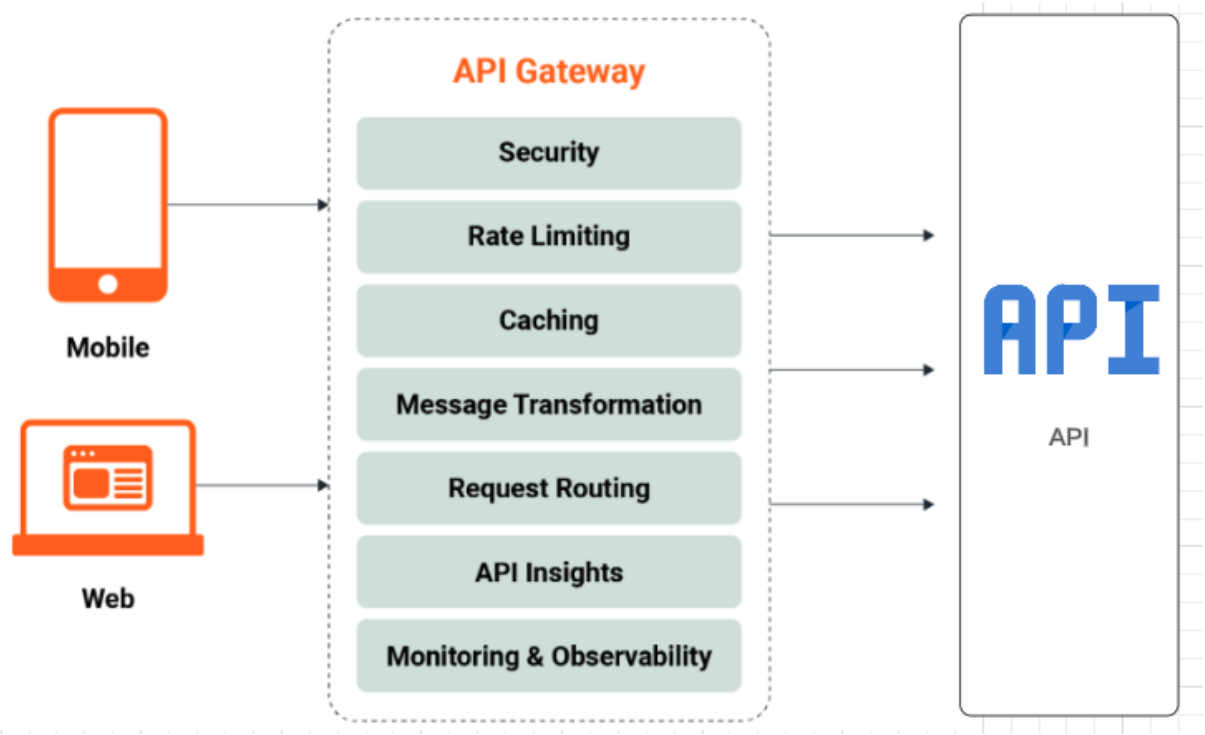


Working of API Gateway

Key Functions of an API Gateway

1. **Routing and Load Balancing:** API gateways intelligently route requests to the appropriate backend services. They can also distribute traffic evenly across multiple instances of a service, ensuring high availability and reliability.
2. **Security**: One of the most critical functions of an API gateway is to enforce security policies. This includes authentication, authorization, and rate limiting. By centralizing security, API gateways help protect backend services from unauthorized access and potential attacks.
3. **Protocol Translation**: API gateways can translate between different protocols, such as HTTP, HTTPS, WebSocket, and gRPC. This allows clients to communicate with backend services using their preferred protocols without requiring changes to the services themselves.

4. **Caching**: To improve performance and reduce latency, API gateways can cache responses from backend services. This is particularly useful for frequently accessed data, as it reduces the load on backend services and speeds up response times.
5. **Monitoring and Analytics**: API gateways provide valuable insights into API usage and performance. They can track metrics such as request counts, response times, and error rates, helping developers identify and address issues proactively.
6. **Transformation and Aggregation**: API gateways can modify requests and responses on the fly, such as adding headers, changing payload formats, or aggregating data from multiple services into a single response. This flexibility simplifies the integration of diverse services.



Benefits of Using an API Gateway

1. **Simplified API Management**: By centralizing API management, API gateways reduce the complexity of handling multiple APIs. This makes it easier to implement consistent policies and monitor API performance.
2. **Enhanced Security**: API gateways provide a unified layer of security, ensuring that all API requests are authenticated and authorized. This reduces the risk of security breaches and protects sensitive data.
3. **Improved Performance**: With features like caching and load balancing, API gateways enhance the performance and reliability of APIs. This leads to a better user experience and higher customer satisfaction.
4. **Scalability**: API gateways enable seamless scaling of backend services by distributing traffic and managing load. This ensures that APIs can handle increased demand without compromising performance.
5. **Flexibility**: API gateways support protocol translation and request/response transformation, making it easier to integrate diverse services and adapt to changing requirements.

Conclusion

In today's interconnected world, APIs are essential for enabling communication between different software systems. However, managing APIs effectively requires a robust solution that can handle security, performance, and scalability challenges. API gateways provide this solution, serving as a critical component for API success. By centralizing API management, enhancing security, and optimizing performance, API gateways empower organizations to build and maintain reliable, high-performing APIs that drive business growth.

As the demand for APIs continues to grow, the role of API gateways will become even more crucial. Organizations that leverage the power of API gateways will be better positioned to succeed in the competitive landscape of modern software development.

About the Author:

Ikram Ahamed Mohamed is a distinguished software engineer with 18 years of experience, specializing in building integration solutions via APIs. His expertise in creating seamless and efficient integration frameworks has made him a valuable asset in the tech industry. Ikram is an active member of prestigious organizations, including being an IEEE Senior Member. He has received numerous accolades, such as the Top API Voice badge on LinkedIn. Currently, Ikram works for Salesforce Inc. USA as a Manager and Integration Lead. In this role, he leads teams in developing and implementing cutting-edge integration solutions that drive business success. Ikram's contributions to the field of API integration are widely recognized. As a thought leader, he continues to share his knowledge and insights through publications and speaking engagements, inspiring the next generation of software engineers.

AI & Cloud Trends

Embracing AI and Multi-Cloud Trends: Unlocking New Frontiers in Digital Transformation
By Ayisha Tabbassum, Senior IEEE Member

The rise of Artificial Intelligence (AI) and multi-cloud strategies has marked a turning point in digital transformation, enabling organizations to innovate faster, scale efficiently, and optimize costs. As we close 2024, understanding these trends is essential for organizations aiming to stay competitive in an increasingly dynamic landscape.

AI Trends in 2024
1. Generative AI in Action: Generative AI models like GPT-4 and MidJourney have revolutionized industries ranging from content creation to drug discovery. Businesses are now integrating these models into workflows to enhance creativity, automate repetitive tasks, and improve decision-making.
2. Explainable AI (XAI): As AI becomes central to decision-making, there is a growing demand for transparency. Explainable AI frameworks allow stakeholders to understand, trust, and validate AI-driven outcomes, addressing ethical concerns and regulatory requirements.
3. AGI and its Implications: Advancements in Artificial General Intelligence (AGI) are pushing boundaries, enabling machines to perform tasks with human-like cognition. While AGI offers transformative potential, ethical and societal concerns must be addressed.
4. AIOps and MLOps: Automation in IT operations (AIOps) and machine learning operations (MLOps) is streamlining the deployment, management, and scaling of AI models. Platforms like Kubeflow and tools such as TensorBoard are pivotal in ensuring operational efficiency and transparency.

Multi-Cloud Trends in 2024
1. Hybrid and Multi-Cloud Adoption: Organizations are increasingly adopting hybrid and multi-cloud strategies to avoid vendor lock-in and leverage best-of-breed solutions from different providers. This approach also ensures higher availability and disaster recovery capabilities.
2. Unified Observability and Cost Management: Managing performance, security, and costs has become complex in multi-cloud environments. Unified observability platforms and FinOps tools like AWS Cost Explorer and Apptio are now critical for optimizing resource utilization.
3. AI-Powered Cloud Optimization: Cloud providers are leveraging AI to enhance workload optimization, resource scaling, and predictive maintenance. These advancements ensure that organizations achieve operational efficiency while reducing costs.

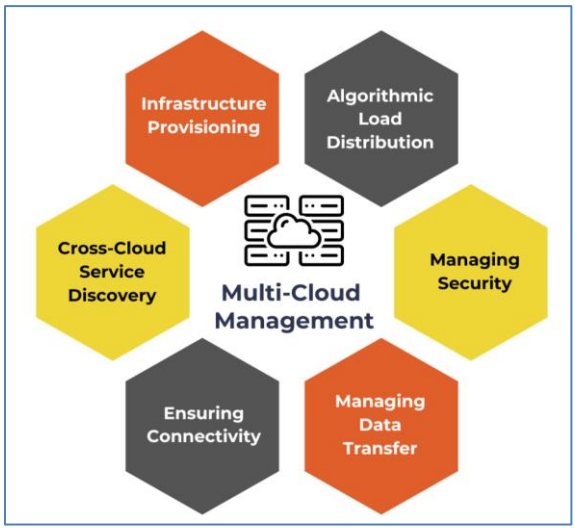


Figure : AI-Based Multi Cloud Management Aspects

Opportunities, Challenges, and Solutions

1. Opportunities
   Enhanced scalability and flexibility through multi-cloud and AI integration.
   Real-time insights and faster processing with Edge AI and AIOps.
   Streamlined workflows with platforms like Kubeflow for MLOps.

2. Challenges
   Complexity in managing multi-cloud environments and AI workloads.
   Ethical concerns with AGI and lack of transparency in AI decisions.
   High costs associated with AI and multi-cloud deployments.

3. Solutions
   Leveraging tools like Tensor Board for transparent model monitoring.
   Adopting robust governance frameworks for ethical AI usage.
   Employing FinOps strategies to optimize cloud spending and improve ROI.

Convergence of AI and Multi-Cloud

The integration of AI with multi-cloud platforms is a game changer. AI models hosted across cloud providers ensure high availability and improved performance, while multi-cloud strategies provide the scalability and flexibility needed for large-scale AI deployments. For instance, training AI models on datasets spread across AWS, Azure, and Google Cloud enables organizations to leverage each provider's unique capabilities. Meanwhile, AI-powered tools help optimize multi-cloud environments, ensuring seamless operations and cost efficiency.

The Road Ahead

As we move into 2025, the interplay between AI and multi-cloud will continue to drive innovation. Organizations must stay abreast of these trends, invest in upskilling their workforce, and adopt a proactive approach to technology adoption. The opportunities are immense, and those who embrace these advancements will undoubtedly lead the charge in the next wave of digital transformation.

Author Bio:

Ayisha Tabbassum is a seasoned Cloud Architect and thought leader with over a decade of experience in enterprise and multi-cloud architecture, infrastructure automation, and CI/CD application deployment. As the Founder and CEO of One Stop for Cloud, an NVIDIA-certified training partner, Ayisha has empowered professionals worldwide through hands-on training and mentorship in AI, cloud computing, and FinOps. With six patents, 40 research publications, and numerous accolades, she is a sought-after speaker and mentor, guiding aspiring cloud professionals toward achieving their career aspirations. Ayisha is also an active Senior IEEE member and a passionate advocate for diversity in technology.

New EMB Student Chapter

Cecelia Jankowski
Managing Director
Member and Geographic Activities
Phone + 1 732 562 5504
Fax + 1 732 867 9943
c.jankowski@ieee.org

Kate Fernandez

Rochester, MN
USA

Dear Kate Fernandez:

Congratulations! It is a pleasure to inform you that the requirements of the Member and Geographic Activities Board Operations Manual have been met and the IEEE Engineering in Medicine and Biology Society Student Branch Chapter at the Mayo Clinic Graduate School has been formed. The effective date of this Student Branch Chapter formation is 04 November 2024.

On behalf of the IEEE and its members, I would like to welcome your Branch Chapter to the student program. If you have any questions or need assistance, please do not hesitate to contact our Student Services department at:

Student Services
IEEE Member and Geographic Activities Department
445 Hoes Lane
Piscataway, NJ 08854

student-services@ieee.org, email
+1 732 562 5527, phone
+1 732 463 9359, fax

Sincerely,

Cecelia Jankowski

Cecelia Jankowski
Managing Director
Member and Geographic Activities

cc: V. Ozburn – Region 4 Director
J, Wolf – Region 4 Student Activities Chair
P. Sajda – Engineering in Medicine & Biology Society President
R. Laverello – Engineering in Medicine & Biology Society Vice President – MSA
N. Zimmerman – Executive Director Engineering in Medicine & Biology Society
N. Caballero – Engineering in Medicine & Biology Society Student AdCom Member
L. Zhong – Engineering in Medicine & Biology Society Chapter Development
C. White – Southern Minnesota Section Chair
L. Pramanik – Student Branch Counselor
A. Schreiber – Student Branch Chair
D. Holmes – Student Branch Chapter Advisor

GenAI & Cybersecurity

Generative AI as a Game-Changer in Cybersecurity

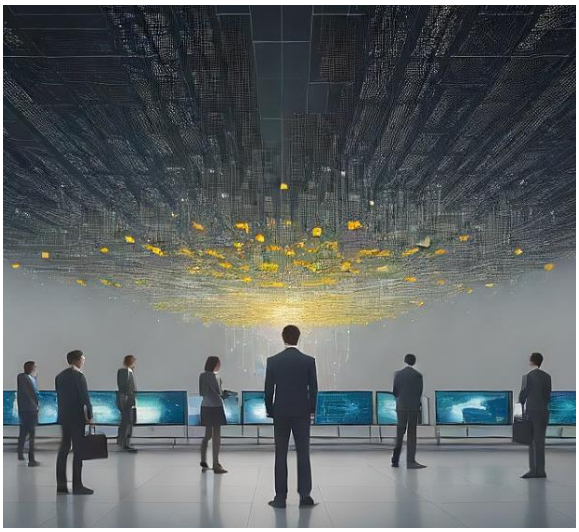
As digital ecosystems expand, safeguarding sensitive information has become a priority for organizations worldwide. Information leakage—the unintentional exposure of confidential data—severely threatens financial stability, reputation, and legal compliance. Generative AI has emerged as a transformative solution, offering proactive, scalable, and practical strategies to mitigate these risks. By identifying vulnerabilities, simulating threats, and deploying innovative prevention mechanisms, generative AI is redefining how organizations address information security challenges.

The Threat of Information Leakage in the Digital Era

Information leakage stems from multiple sources, including insider threats, phishing attacks, inadequate encryption, and misconfigured systems. The rapid digitization of processes and the increased use of data-sharing platforms have magnified the risk. A single instance of leakage can lead to substantial financial losses, legal repercussions, and diminished stakeholder trust. Traditional methods often need to be revised to address these multifaceted challenges, necessitating the adoption of advanced technologies like generative AI. Its ability to analyze large datasets, detect anomalies, and predict potential security breaches equips organizations with a proactive approach to information security, ensuring better detection of threats and preemptive action against potential risks.

Generative AI: Enhancing Information Security Frameworks

Generative AI leverages advanced models such as Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) to enhance information security. These models identify patterns, simulate scenarios, and generate synthetic data. By doing so, they offer organizations the ability to anticipate potential vulnerabilities and strengthen their defenses without exposing sensitive data. Generative AI can create realistic simulations of cyberattacks, enabling organizations to test their defenses against potential leakage scenarios. This approach helps identify weaknesses in existing systems and supports the development of more robust security measures. Additionally, AI-generated synthetic data replicates the statistical properties of real data without compromising privacy, making it ideal for testing and training purposes. This adaptability ensures that organizations remain one step ahead of malicious actors while adhering to data protection regulations.



Detecting Anomalous Behavior with Generative AI

An impactful application of generative AI in information security is its ability to detect system anomalies. By analyzing user behavior patterns, generative AI identifies deviations that signal suspicious activities. For instance, an employee accessing sensitive files outside typical working hours or downloading an unusual volume of data might indicate a potential threat. Generative AI flags such incidents and provides contextual insights, helping security teams determine whether the activity is benign or malicious. This capability significantly reduces false positives, enhancing operational efficiency and response times. In large organizations, where user behavior can vary widely, anomaly detection is critical to preventing information leakage. Over time, generative AI learns from emerging threats and adjusts its models, offering a dynamic and evolving layer of protection.

Addressing Insider Threats Through Behavioral Modeling

Insider threats, where individuals within an organization misuse their access to sensitive data, are among the most challenging to detect. Generative AI mitigates this risk by building detailed behavioral profiles for each user and tracking metrics such as access frequency, data usage patterns, and file interactions. These profiles establish a baseline for normal behavior, making it easier to spot deviations that might signal a threat. For example, frequent access to restricted

files or attempts to transfer data to external devices might trigger an alert. Reinforcement learning algorithms allow generative AI models to evolve alongside user roles and organizational changes, ensuring continuous protection against insider threats. This proactive approach minimizes the risk of information leakage originating within an organization, which is often a significant vulnerability.

Privacy Challenges and Generative AI Solutions

While generative AI provides powerful tools for preventing information leakage, its reliance on extensive datasets raises privacy concerns. Effective AI-driven security systems must balance the need for comprehensive data analysis with stringent privacy requirements. Generative AI addresses this challenge through federated learning, which allows AI models to train on data locally, ensuring that sensitive information never leaves its source. Only model updates are shared, minimizing exposure risks and supporting compliance with privacy regulations. Additionally, differential privacy techniques introduce noise into datasets, preserving the anonymity of individual data points while maintaining overall data utility. These privacy-preserving strategies enable organizations to harness the benefits of generative AI without compromising their compliance with data protection laws like GDPR and CCPA.

Leveraging Synthetic Data for Testing and Training

Synthetic data generated by AI replicates real-world scenarios without exposing actual sensitive information. This capability is precious for conducting security drills and penetration tests. Organizations can simulate phishing attacks, malware infiltrations, and other cyber threats to evaluate their defenses. By using synthetic data, organizations ensure compliance with data protection laws while preparing for potential security breaches. This approach fosters a culture of privacy, regulatory adherence, and proactive risk management, enabling organizations to develop and maintain robust defenses against evolving cyber threats.

Challenges and Future Directions in Generative AI for Security

Despite its potential, implementing generative AI in information security is challenging. The complexity of generative models requires specialized expertise and significant resources, which can hinder adoption among smaller organizations. Additionally, malicious actors' misuse of generative AI—such as creating realistic phishing content—poses new security threats. To address these challenges, the technology sector must collaborate on developing standardized frameworks and best practices for AI deployment. Integrating generative AI with blockchain technology could enhance security by providing immutable audit trails for data transactions. Blockchain's decentralized nature complements generative AI, enabling better traceability and accountability. As AI research progresses, efforts to improve the interpretability and transparency of generative models will help build trust in AI-driven security solutions.

Conclusion: Pioneering the Future of Information Security

Generative AI offers unparalleled opportunities to revolutionize information security. By leveraging its capabilities in anomaly detection, synthetic data generation, and adaptive behavioral analysis, organizations can adopt proactive strategies to prevent information leakage. While challenges remain, the evolution of generative AI and collaborative efforts to establish ethical frameworks will ensure its positive impact on global security. As digital ecosystems grow more complex, embracing generative AI as a cornerstone of information security is not just an option but a necessity. By doing so, organizations can protect their most valuable assets and build trust and resilience in the face of evolving cyber threats.

About Author:

Vishwanadham Mandala (PhD) is an IEEE Senior Member with 20 years of industry experience in Big Data, AI & ML, Data integration, and Data Architecture. He has a Bachelor's and Master's in CSE, a master's in data science, and a PhD in CSE. He has 16 patents in diverse areas, granted in India, the UK, and Germany. He has 36+ research papers published in Elsevier, Springer, MDPI, and other journals. He has been a reviewer for many organizations.

New Computer Chapter

Cecelia Jankowski
Managing Director
Member and Geographic Activities
Phone + 1 732 562 5504
Fax + 1 732 867 9943
c.jankowski@ieee.org

5 December 2024

Darryl Palmer
Chillicothe, IL 61523-9529
USA

Dear Darryl Palmer:

Congratulations! On behalf of the IEEE Member and Geographic Activities Vice President, Deepak Mathur, and IEEE Technical Activities Vice President, Manfred J Schindler, it is a pleasure to inform you that the requirements of the MGA Board Operations Manual have been met, and the IEEE Central Illinois Section Computer Society Chapter has been formed. The effective date of this chapter formation is 04 December 2024.

You have been recorded as the Chapter Chair. When an election has been held, please report the name and member number of the new Chapter Chair to the IEEE using the online officer reporting tool at https://officers.vtools.ieee.org/. Valuable information regarding IEEE Society Chapters can be found at http://www.ieee.org/societies_communities/geo_activities/chapters. If we can assist you in any way in the planning of the chapter activities, please let us know.

We extend our best wishes for the successful operation of this chapter.

Sincerely,

Cecelia Jankowski

Cecelia Jankowski
Managing Director
Member and Geographic Activities

cc: D. Mathur – Member and Geographic Activities Vice President
M. Schindler – Technical Activities Vice President
V. Ozburn – Region 4 Director
J. Athavale – Computer Society President
K. Boateng – Computer Society Vice President - Membership and Geographic Activities
D. Bondurant – Computer Society Geographic Activities Committee
B. Mayes – Central Illinois Section Chair
M. Bahar – Technical Activities Managing Director

New TC Sensors Council

Cecelia Jankowski
Managing Director
Member and Geographic Activities
Phone + 1 732 562 5504
Fax + 1 732 867 9943
c.jankowski@ieee.org

1 December 2024

Patricia Khashayar
Minneapolis, MN 55414-1917
USA

Dear Patricia Khashayar:

Congratulations! On behalf of the IEEE Member and Geographic Activities Vice President, Deepak Mathur, and IEEE Technical Activities Vice President, Manfred J Schindler, it is a pleasure to inform you that the requirements of the MGA Board Operations Manual have been met, and the IEEE Twin Cities Section Sensors Council Chapter has been formed. The effective date of this chapter formation is 14 November 2024.

You have been recorded as the Chapter Chair. When an election has been held, please report the name and member number of the new Chapter Chair to the IEEE using the online officer reporting tool at https://officers.vtools.ieee.org/. Valuable information regarding IEEE Society Chapters can be found at http://www.ieee.org/societies_communities/geo_activities/chapters. If we can assist you in any way in the planning of the chapter activities, please let us know.

We extend our best wishes for the successful operation of this chapter.

Sincerely,

Cecelia Jankowski

Cecelia Jankowski
Managing Director
Member and Geographic Activities

cc: D. Mathur – Member and Geographic Activities Vice President
M. Schindler – Technical Activities Vice President
V. Ozburn – Region 4 Director
D. Uttamchandani – Sensors Council President
A. Naumaan – Twin Cities Section Chair
M. Bahar – Technical Activities Managing Director

AI & Healthcare Risks

Navigating the Risks of AI in Healthcare: NIST AI 600–1
A Guide to Responsible Implementation and Risk Management

The adoption of AI in healthcare has the potential to enhance patient care, streamline administrative processes, and improve medical research. However, it also introduces unique risks that require a structured approach to risk management. The National Institute of Standards and Technology (NIST) provides a comprehensive framework for addressing these risks, specifically tailored for GAI applications. Healthcare organizations must navigate these risks carefully, as errors in GAI systems could lead to misdiagnoses, privacy violations, and unintended harm to patients.

NIST AI Risk Management Framework
The **NIST AI Risk Management Framework: Generative AI Profile 600-1** focuses on managing risks associated with Generative Artificial Intelligence (GAI), providing guidance for integrating AI trustworthiness and risk mitigation into the AI lifecycle. While "NIST 100-1" is part of a broader AI framework addressing global engagement on AI standards, outlining strategies for international cooperation on AI regulations and best practices; essentially, 600-1 is a specific focus on generative AI risks within the larger AI landscape covered by 100-1.

NIST 100-1:

- Deals with establishing international AI standards collaboration.
- Focuses on facilitating communication and alignment between different countries regarding AI regulations.
- Provides a roadmap for global engagement on AI governance.

NIST 600-1:

- Concentrates on generative AI risks.
- Provides guidance on mitigating potential dangers from advanced AI models like large language models.
- Considered a companion document to the broader NIST AI Risk Management Framework.

As healthcare continues to embrace AI technologies, the risks identified within this framework become increasingly relevant for ensuring the safe and responsible deployment of AI-driven solutions, particularly those utilizing Generative AI.

Unique Risks of GAI in Healthcare
GAI systems are designed to generate new content based on existing data, such as large language models (LLMs) for text generation or image generation models for medical imaging. While these systems hold immense promise, their use in healthcare is fraught with potential risks, particularly:

- **Data Privacy**: GAI systems often rely on vast datasets that may include sensitive health information. The use of such data without proper consent or safeguards can lead to privacy breaches.
- **Harmful Bias**: GAI models can inherit and amplify biases present in training data, leading to discriminatory outcomes. In healthcare, this could manifest in biased diagnostic recommendations, especially for underrepresented populations.
    - For example, an AI system trained primarily on data from one demographic may perform poorly for others, potentially leading to unequal care outcomes.
- **Misinformation and Disinformation**: GAI's ability to generate convincing yet false content can exacerbate misinformation in healthcare. For instance, incorrect medical advice generated by an AI model could spread through social media, potentially causing harm to individuals who act on it.
    - For example, a GAI system might inaccurately summarize patient medical history, leading to wrong treatment recommendations. Ensuring that AI outputs are accurate and reliable is critical, especially in applications that require medical expertise.

- **Intellectual Property (IP)**: GAI systems used in healthcare research could inadvertently violate intellectual property rights by generating content based on proprietary medical data or research without proper attribution. This could lead to legal challenges or the unethical use of research.

Governance and Risk Management Actions for Healthcare

The NIST framework outlines specific actions for managing GAI risks, which can be applied in healthcare settings to ensure that AI systems are trustworthy and safe. Key governance actions include:

- **Establishing Clear Governance Policies**: Healthcare organizations should create policies that align GAI development with legal and regulatory requirements, ensuring that privacy and intellectual property rights are respected. Transparent documentation of the training data, algorithms, and AI system outputs is crucial for maintaining accountability.
- **Pre-Deployment Testing**: Before deploying GAI systems in clinical settings, healthcare organizations should conduct rigorous testing to ensure the systems are free of harmful biases and capable of producing accurate, evidence-based outcomes. This testing should assess the system's performance across diverse demographic groups and in a variety of medical scenarios.
- **Continuous Monitoring and Incident Disclosure**: Once deployed, GAI systems must be continuously monitored for performance issues and potential risks. Healthcare providers should have incident disclosure protocols in place to address any failures promptly. This includes setting up feedback loops from healthcare professionals who interact with AI systems to detect errors or discrepancies in the AI's output.
- **Ensure Bias Mitigation**: Steps must be taken to identify and mitigate bias in AI models, including ensuring diverse data representation during model training and conducting regular audits of AI outputs to identify any emerging biases.
- **Transparency and Explainability**: Healthcare professionals need to understand how AI systems make decisions, especially when those decisions impact patient care. Implementing AI systems that are explainable and transparent allows clinicians to better trust AI recommendations and ensure they align with clinical guidelines.
- **Adopt Privacy-Enhancing Technologies**: Use advanced encryption, differential privacy, and secure multi-party computation techniques to protect patient data used in AI systems. Regular audits and compliance checks should be conducted to ensure data privacy is maintained.

Key Considerations for Healthcare AI Applications

Several areas in healthcare require special attention when using GAI systems:

- **Patient Safety**: AI systems must be robust and fail-safe, ensuring that errors in AI outputs do not lead to patient harm. For example, the misinterpretation of medical imaging or erroneous drug recommendations could have serious consequences. Proper safeguards and human oversight are critical.
- **Regulatory Compliance**: GAI systems used in healthcare must comply with regulatory standards such as HIPAA in the U.S. or GDPR in the EU. Organizations need to ensure that the data used to train AI systems is ethically sourced, anonymized where necessary, and processed in accordance with applicable laws.
- **Human-AI Interaction**: Healthcare professionals must be trained to effectively interact with GAI systems. Over-reliance on AI tools or poor configuration could lead to automation bias, where clinicians defer too much to AI recommendations without applying their professional judgment.
- **Ethical and Legal Implications**: The use of GAI in healthcare raises ethical and legal challenges, particularly regarding informed consent, data ownership, and the use of AI in decision-making. Ethical considerations must guide the development, deployment, and use of these systems to protect patients' rights.

Conclusion

Generative AI holds great potential for transforming healthcare, but it also presents significant risks that must be carefully managed. By implementing the risk management practices outlined in the NIST AI Risk Management Framework, healthcare organizations can mitigate these risks and harness the power of AI in a responsible and ethical manner.

Trustworthy AI systems will be key to advancing healthcare innovation while ensuring patient safety, data privacy, and equitable outcomes for all.

References:
NIST Trustworthy and Responsible AI NIST AI 600–1: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.600-1.pdf

NIST AI 6001, Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile Request for Comment Submitted by the Johns Hopkins Center for Health Security
https://centerforhealthsecurity.org/sites/default/files/2024-06/2024-06-02-jhchs-nist-ai-6001-rfc.pdf

Call for the responsible artificial intelligence in the healthcare: https://pmc.ncbi.nlm.nih.gov/articles/PMC11047988/

Ethical implications of AI and robotics in healthcare: A review: https://pmc.ncbi.nlm.nih.gov/articles/PMC10727550/

Ethics and governance of artificial intelligence for health: Guidance on large multi-modal models:
https://www.who.int/publications/i/item/9789240029200

Artificial intelligence in healthcare: transforming the practice of medicine:
https://www.sciencedirect.com/science/article/pii/S2514664524005277

Regulatory considerations on artificial intelligence for health: https://www.who.int/publications/i/item/9789240078871

About the Author

Vijay Viradia (PMP, SAFe, CSM)
Project Management, Healthcare & AI Specialist, Menomonee Falls, WI, USA. Mobile: +1 (262) 290-3160

LinkedIn | Blog | Website | Patent | Peer Reviews | Speaker

Vijay is a distinguished Digital Healthcare Transformation Leader, known for his unique blend of Healthcare IT, Cloud, AI, Data Analytics and Project Management expertise in delivering Large-Scale Healthcare IT solutions, and business acumen. He worked with Fortune 500 companies in delivering healthcare solutions. Vijay led the team of Architects, Business Analysts, Developers, and Operations staff for the implementation of the healthcare system.

Vijay has a strong interest in:

- Leveraging cloud, AI/ML, and microservices to transform healthcare systems.
- Revolutionize Patient Outcomes through Emerging Technologies.
- Health Equity, healthcare needs in rural areas, and enhancing access to healthcare for all individuals.
- Enhancing interoperability, data integrity, privacy, and security.
- Improving patient & provider experiences, quality of care, and healthcare operations.
- Improving healthcare workforce with training and guidance.
- Spreading awareness about Health insurance and government initiative towards better healthcare.

He can be contacted via email: vjv9982@gmail.com

Fox Valley Student Branch

Cecelia Jankowski
Managing Director
Member and Geographic Activities
Phone + 1 732 562 5504
Fax + 1 732 867 9943
c.jankowski@ieee.org

8 December 2024

Ethan Van Ornum
Waupaca, WI 54981-8526
USA

Dear Ethan Van Ornum:

Welcome to the IEEE Student Branch program! On behalf of the Member and Geographic Activities Board, we have approved your petition to form an IEEE Student Branch at Fox Valley Technical College.

Your Student Branch is located in Region 4 and your activities will be of interest to the volunteers listed below:

Vickie Ozburn, Region 4 Director
Jonathon Wolf, Region 4 Student Activities Chair
Stephen Messing, Region 4 Student Representative
Aurenice Oliveira, Northeastern Wisconsin Section Chair

Your Student Branch code is STB60100740 and your School Code is 60227782. Please be sure to use them on all correspondence and reporting forms. To ensure that the students are properly assigned to your Student Branch, they should join IEEE online at http://www.ieee.org/join and use the school search to find the school name Fox Valley Technical College.

On behalf of the IEEE and its members, I would like to welcome your Branch to the student program. If you have any questions or need assistance, please do not hesitate to contact our Student Services department at:

Student Services
IEEE Member and Geographic Activities Department
445 Hoes Lane
Piscataway, NJ 08854

student-services@ieee.org, email
+1 732 562 5527, phone
+1 732 463 9359, fax

Sincerely,

Cecelia Jankowski

Cecelia Jankowski
Managing Director
Member and Geographic Activities

cc:
V. Ozburn – Region 4 Director
J. Wolf – Region 4 Student Activities Chair
S. Messing – Region 4 Student Representative
A. Oliveira – Northeastern Wisconsin Section Chair
B. Bahraminejad – Student Branch Counselor

Congrats R4 Fellows!

On behalf of everyone in Region 4, we are pleased to share with you – all of the IEEE members who have been elevated to Fellow status, starting January 1 , 2025. Congratulations!

IEEE Fellows Elevated as of January 2025

R4 -Central USA

Central Illinois Section

Indranil Gupta

for contributions to reliable large-scale distributed systems

Maxim Raginsky

for contributions to information-theoretic analysis of stochastic systems in optimization and machine learning

Central Indiana Section

Dionysios Aliprantis

for contributions to rotating electric machine modeling and distributed energy resources

Chicago Section

Karen Livescu

for contributions to multi-view and pre-trained speech representation learning

Pai-yen Chen

for contributions to mesoscopic and multiscale electromagnetics for antenna and sensor applications

Zuyi Li

for contributions to functional microgrid design and microgrid cybersecurity analyses

Madison Section

Umit Ogras

for contributions to networks-on-chip for heterogeneous manycore architectures

Milwaukee Section

Craig Colopy

for contributions to design, development, and application of single-phase 32-step voltage regulators

Southeastern Michigan Section
Caisheng Wang *for contributions to modeling and control of distributed alternative energy systems and battery storage management*
Ece Yaprak *for leadership in engineering technology accreditation and education*
Jeffrey Fowlkes *for contributions to therapeutic ultrasound and the understanding of acoustic cavitation in medicine*
Lingjie Guo *for contributions to nanoimprint, scalable nanopatterning*
Zhen Xu *for development and clinical translation of non-invasive mechanical ultrasound ablation technology*
Twin Cities Section
Mingyi Hong *for contributions to optimization in signal processing, wireless communication and machine learning*

Policy Migration

Policy Remigration Unveiled: A Key Strategy for Underwriters to Ensure Accuracy and Compliance in Insurance

Author: Sharmila Devi Chandariah, Senior Technical Lead in Financial Services, SMIEEE, Policy Administration System & GenAI Expert

Introduction:

In the dynamic world of insurance, where adaptability and efficiency are crucial, the concept of policy remigration has gained significant attention. This process involves re-evaluating and transferring legacy policies that have already been migrated into modern Policy Administration Systems (PAS). While this may seem counterintuitive to some, underwriters are increasingly advocating for remigration due to several compelling reasons. Understanding the benefits of this practice can illuminate its importance in today's insurance landscape.

Why Underwriters Seek Policy Remigration:

Underwriters play a pivotal role in assessing risk and determining the terms of insurance policies. As they navigate the complexities of modern insurance practices, they often find that legacy policies despite having been migrated may not align perfectly with current underwriting standards or business objectives. This misalignment can stem from various factors, including changes in regulatory requirements, evolving market conditions, or advancements in technology that enhance data analytics capabilities.

Remigrating these policies allows underwriters to ensure that all legacy data is accurately represented and compliant with current standards. By revisiting previously migrated policies, underwriters can identify discrepancies or outdated information that could affect risk assessment and pricing strategies. This proactive approach not only enhances the accuracy of underwriting decisions but also helps maintain regulatory compliance, ultimately safeguarding the organization against potential liabilities.

Benefits of Policy Remigration in the Real World

The benefits of policy remigration extend beyond compliance and accuracy; they also encompass operational efficiency and improved customer service. By automating the remigration process, insurance companies can streamline their operations and reduce manual intervention, which often leads to errors and inconsistencies. Automation can significantly enhance productivity by allowing underwriters to focus on higher-value tasks rather than getting bogged down in administrative processes.

Moreover, remigrated policies can be integrated with advanced analytics tools that provide deeper insights into customer behavior and risk profiles. This integration empowers underwriters to make data-driven decisions, leading to more competitive pricing models and tailored insurance products that meet the evolving needs of customers.

Use Case: Automating Remigrations for Underwriters

Consider an insurance company that has recently migrated a substantial number of legacy policies into a new PAS. However, as market conditions change and new regulations are introduced, the need arises to reassess these policies for compliance and accuracy. By implementing an automated remigration solution, the company can efficiently extract relevant data from the existing PAS and reprocess it according to updated guidelines.

For example, if an underwriter identifies a specific segment of policies that require adjustments due to new regulatory standards, automation can facilitate a swift remigration process. The system can automatically flag these policies for review, extract necessary data, and reformat it according to the new requirements without extensive manual input. This not only saves time but also reduces the risk of human error during the remigration process.

Furthermore, automation allows for real-time tracking of policy changes and updates within the PAS. Underwriters can receive alerts regarding any discrepancies or issues that arise during remigration, enabling them to address concerns promptly. This level of oversight enhances operational transparency and fosters a culture of accountability within the organization.

Enhancing Data Quality Through Remigration

One of the critical advantages of policy remigration is the opportunity to enhance data quality within an organization's PAS. Legacy systems often contain outdated or inaccurate information that can lead to poor underwriting decisions and increased risk exposure. By remigrating these policies, insurers can cleanse their data sets, ensuring that only accurate and relevant information is retained.

For instance, during the remigration process, underwriters can conduct thorough reviews of policyholder information, claims history, and coverage details. This meticulous examination allows them to correct any inaccuracies or

inconsistencies before they become embedded in the new system. Improved data quality not only supports better underwriting practices but also enhances customer trust by ensuring that policyholders receive accurate coverage details and premium calculations.

Supporting Compliance Initiatives

In an era marked by stringent regulatory requirements, policy remigration serves as a crucial tool for supporting compliance initiatives within insurance organizations. As regulations evolve, insurers must ensure that their policies reflect current legal standards and industry best practices. By undertaking remigration efforts, underwriters can align legacy policies with new compliance mandates effectively.

For example, if new legislation requires specific disclosures or changes in coverage terms, remigrating affected policies allows insurers to update their offerings without starting from scratch. This adaptability not only ensures compliance but also minimizes disruption for policyholders who may otherwise face confusion during transitional periods.

Conclusion

Policy remigration is becoming an essential practice for underwriters in the insurance industry as they strive to maintain accuracy, compliance, and operational efficiency. By revisiting legacy policies that have already been migrated into modern systems, underwriters can ensure alignment with current standards while leveraging automation to streamline processes. As organizations continue to navigate an increasingly complex regulatory environment and evolving market demands, embracing policy remigration will be vital for maintaining competitive advantage. The ability to automate these processes not only enhances productivity but also empowers underwriters to focus on strategic decision-making that drives business growth.

Data Retrieval & GenAI

From Natural Language to SQL: How Generative AI is Transforming Data Retrieval

Author: Sharmila Devi Chandariah, Senior Technical Lead in Financial Services, SMIEEE, Policy Administration System & GenAI Expert

Introduction:

The emergence of Generative AI (GenAI) technologies has significantly transformed the landscape of data management, particularly in the realm of SQL query generation. The innovative text-to-SQL solution harnesses the power of natural language processing to enable users to generate complex SQL queries from simple, natural language inputs. This advancement not only streamlines data retrieval processes but also democratizes access to information across various industries, including finance, healthcare, and insurance.

The Importance of Data Accessibility

In today's data-driven world, accessing and analyzing information is crucial for informed decision-making. However, the traditional method of retrieving data often involves writing intricate SQL queries, a task that can be both time-consuming and error prone. This challenge is particularly pronounced for non-technical users who may lack the expertise needed to navigate complex database schemas. The text-to-SQL solution addresses this issue by allowing users to input questions in plain English, which are then translated into SQL queries, thus simplifying the interaction with databases.

Mechanism of the Text-to-SQL Solution

The text-to-SQL solution operates through two primary approaches: the Basic Approach and the Retrieval Augmented Generation (RAG) Approach.

Basic Approach:

In the Basic Approach, users interact directly with a Large Language Model (LLM) by entering a natural language query along with relevant database schema information. The LLM processes this input to generate the corresponding SQL query. The architecture of the text-to-SQL solution involves several key steps that facilitate the seamless generation of SQL queries from natural language inputs. Initially, the user types a question into a chatbot interface, providing a straightforward means of interaction. Following this, the system engages in prompt creation by combining the user input with relevant database schema details, which includes information about tables, fields, and their relationships. This combined input is then processed by a Large Language Model (LLM), which interprets the context and requirements to generate an appropriate SQL query. Finally, the generated SQL query is returned to the user through the chatbot interface, completing a streamlined process that enhances data accessibility and usability for non-technical users. This straightforward process is particularly effective for simple queries where the structure is predictable and does not require extensive contextual understanding.

RAG Approach:

The RAG (Retrieval Augmented Generation) Approach enhances the capabilities of the Basic Approach by integrating external knowledge sources to improve both accuracy and contextual relevance in SQL query generation. This method employs a more

sophisticated workflow that begins with pre-processing, where knowledge sources such as sample SQL queries and customization instruction files are embedded into a vector database. Following this, users enter their queries into a chatbot interface, like the Basic Approach. The next step involves schema retrieval and text embedding, where user queries are combined with relevant database schemas and converted into numerical vectors that represent their semantic meaning. A cosine similarity search is then performed to retrieve pertinent information from the vector database based on the user input. Finally, the Large Language Model (LLM) generates an SQL query using the augmented context derived from both user inputs and external knowledge sources. This approach is particularly beneficial for handling complex queries that involve multiple conditions or intricate relationships within large datasets, thereby significantly enhancing the overall effectiveness of data retrieval processes.

Benefits of Text-to-SQL Solutions

The implementation of text-to-SQL solutions offers numerous advantages that significantly enhance data accessibility and usability across various industries. One of the primary benefits is the democratization of data access, which allows non-technical users to query databases using natural language, thereby empowering a broader range of employees to engage with data without needing extensive SQL knowledge. This accessibility is complemented by improved efficiency, as automated SQL query generation saves time and reduces the manual effort required for data retrieval, enabling professionals to focus on higher-level analytical tasks. Additionally, these solutions contribute to error reduction; Large Language Models (LLMs) trained on extensive datasets produce syntactically correct and logically sound SQL queries, minimizing the potential for errors that could arise from manual query writing. Furthermore, the complexity of data retrieval is reduced, leading to enhanced productivity, as professionals can manage more tasks efficiently without requiring deep technical expertise. Finally, the consistent query generation provided by LLMs ensures standardized outputs that facilitate easier maintenance and reliability across various applications, ultimately driving better decision-making and operational effectiveness within organizations.

Choosing Between Approaches

When implementing a text-to-SQL solution, organizations must carefully evaluate their specific needs to determine whether to adopt the Basic or RAG (Retrieval Augmented Generation) approach. The first consideration is the complexity of queries; for straightforward queries, the Basic Approach is often sufficient, while intricate scenarios involving multiple tables or advanced joins are better suited for RAG. Another important factor is the user expertise level; if end-users are non-technical, RAG provides a more user-friendly interface that abstracts the complexities associated with SQL, making it easier for them to interact with the system. Additionally, data freshness is crucial; RAG's ability to connect with external databases ensures that users have access to real-time information, which is vital for many business operations where timely data is essential. Furthermore, organizations should consider performance metrics; RAG offers structured performance evaluations through metrics like Average Precision (AP) and Mean Reciprocal Rank (MRR), which are essential for continuous improvement in data retrieval processes. By weighing these factors query complexity, user expertise, data freshness, and performance metrics organizations can make informed decisions about which approach aligns best with their operational requirements and goals.

Use Case: Streamlining Claims Processing in Insurance

In the insurance industry, efficient data retrieval is critical for enhancing operational workflows especially in claims processing. A text-to-SQL solution powered by Generative AI can significantly streamline this process by enabling insurance professionals to query complex databases using natural language. For instance, consider an insurance claims manager who needs to retrieve detailed information about claims filed within the last year that exceed a certain monetary threshold associated with specific types of coverage like comprehensive or collision insurance. Traditionally, this task would require writing intricate SQL queries involving multiple tables claims details, policy information, and customer records making it time-consuming and prone to errors. With a text-to-SQL solution in place, the claims manager simply types a question such as "Show me all claims filed in the last year that exceed $10,000 linked to comprehensive coverage" into a chatbot interface. The system processes this input through either Basic or RAG approaches using LLMs trained on extensive datasets to generate an accurate SQL query automatically. Not only does this reduce manual effort significantly but it also empowers non-technical staff to access critical information quickly and efficiently while minimizing errors associated with manual query writing.

Conclusion

The text-to-SQL solution powered by Generative AI represents a significant leap forward in how organizations can interact with their data. By translating natural language queries into SQL commands, it not only enhances accessibility but also improves efficiency and accuracy in data retrieval processes. As industries continue to grapple with complex data environments, adopting such innovative solutions will be crucial for fostering informed decision-making and driving business success in an increasingly competitive landscape. In summary, whether through simple interactions or sophisticated contextual augmentations, text-to-SQL solutions are poised to redefine how organizations leverage their data assets making it more accessible and actionable than ever before.

Author Bio:

Sharmila Devi Chandariah is a Senior Technical Lead with over a decade of extensive experience in the fintech industry, particularly focusing on the banking and insurance sectors. As a Senior Member of IEEE, she has demonstrated exceptional leadership and technical expertise throughout her career. Her specialization includes developing various web applications and transforming Policy Administration Systems for the U.S. Property & Casualty Insurance. Leveraging her Guidewire Certified ACE credentials, Sharmila excels in delivering innovative solutions tailored to the unique needs of insurance clients, particularly through her expertise in Policy Administration. Sharmila's contributions to the field have been recognized with the Star Associate Award, awarded for her pivotal role in developing the Transient Schema for the Policy Migration tool. This solution significantly enhanced the speed and accuracy of data migration from legacy systems to

new Policy Administration Systems. Driven by a passion for integrating advanced technologies into insurance applications, Sharmila actively applies Machine Learning, Deep Learning, and Natural Language Processing (NLP) to tackle current business challenges. Her work in Generative AI has led to the development of various innovative solutions, including Test to SQL generation, knowledge management, and Splunk integration, resulting in considerable cost savings for her clients. This showcases her ability to merge technical prowess with practical business applications. In addition to her technical roles, Sharmila has recently been invited to serve on the advisory board of a startup focused on leveraging technology to enhance insurance processes. In this capacity, she will provide strategic insights and guidance on product development and market positioning. As the Innovation Lead at her current organization, Sharmila reviews and provides feedback on innovative solutions proposed by her peers, ensuring they align with client needs and can be effectively implemented. Her commitment to excellence and innovation positions her as a key person in advancing technology within the fintech landscape.

Disruptor in Data Engineering

By Dipankar Saha

Earlier this year, I came to know about Apache Iceberg and since then I got intrigued into this technology. This fall I wrote a review paper about Iceberg (link at the end of this article). During the Snowflake Build conference (held between November 12th through 14th, 2024), Iceberg was in the forefront. Not only this technology came up in the keynote talk of the event from Snowflake's chief product officer Christian Kleinerman, Snowflake actually had 4 dedicated sessions involving Iceberg. It was no surprise that the conference was heavy on AI but Iceberg made its own space in the conference and was the second most important thing discussed after AI. I thought of sharing about this timely and relevant technology with the IEEE community which is why I wrote this article.

1. What is it?

Apache Iceberg is an open source table format that has recently gained a lot of attention in the big data world. Managed under Apache Software Foundation, this project has fast become one of the industry favorites to solve some unique and complex big data problems. It is a disruptive and transformative technology which is redefining the landscape of large scale data management. Though this technology belongs in the realm of the data world, it provides unfamiliar capabilities compared to traditional data oriented technologies. As organizations continue to strive for more scalable, cost-effective, performant and reliable ways to handle data, Iceberg sits at the forefront of this to fulfill the expectation. Iceberg is a technology that is driving the goal of enterprise towards "zero copy".

It can be challenging to provide a proper definition of Iceberg to someone who has never encountered it before. While it most certainly is a technology about data, it is distinct from other traditional data-oriented technologies. Someone unfamiliar with Iceberg might ask, is it a database, is it a data warehouse, or is it some other form of database technology like the data lake of the big data world or is it the latest data buzzword in the industry "lakehouse"? The answer is it is neither of those.

Iceberg is actually just a "table format". It doesn't provide storage, it doesn't provide a compute engine either. Its domain of use cases are two fold. First, it allows several compute engines like Spark, Flink, Hive etc. to work simultaneously with the Iceberg tables. Secondly, together with a modern data processing engine, Iceberg helps in implementing the concept "data lakehouse", a hybrid big data technology that leverages the best out of data warehouse and data lake.

2. The advent of Open File Format

Before we delve deeper into Iceberg, we have to take a step back and discuss the data file formats leveraged by Iceberg under the hood. The data files in the context fall under a specific category called Open File Format. These Open File Format data types were invented almost a decade ago to support big data platforms. The motivation behind this open format was to support interoperability across multiple big data engines, avoid file copies, reduce storage cost and provide performance efficiency for data access. The popular Open File Formats are Parquet, ORC and Avro, out of them Parquet is leading the pack.

The Open File Format was invented to solve several problems for big data analytics use cases which were not solvable using the traditional file formats such as CSV, TEXT, JSON or XML.

- **Efficient storage and compression** - big data applications typically contain enormous volumes of data that require huge storage space. These file formats (Parquet, ORC) store data efficiently by implementing compression and encoding techniques to reduce infrastructure overhead as well as cost.

- **Improved query performance** - due to columnar storage architecture, queries often require retrieving data for a subset of the attributes, improving query performance to a great extent.
- **Interoperability across systems** - the file formats are of open standard and consumable by multiple big data engines like Spark, Hive, Impala etc. By storing the file only in one format (Parquet or ORC), multiple engines can operate on the data in parallel thus eliminating the need for storing data in multiple formats, reducing cost as well as storage space.

3. Problem Persist - here comes Open Table Format

Even though Open File Format such as Parquet/ORC solved a lot of problems in big data architecture as mentioned in the above section, there were additional problems that remained intractable. As the data files were stored as independent units, they didn't provide a consolidated tabular view. As a result, query engines were unable to determine which files corresponded to a table. There were some other major problems as well.

- The files do not allow change in schema.
- Time travel over data is not possible.
- Updates are not well supported. If updates are made to multiple files, they are not atomic, causing partial updates in case of failures which are difficult to rollback.

Open Table Format solves this problem by providing a metadata layer which is a set of files that contains information about the data files stored in Parquet/ORC/Avro etc., formats. The key features provided by Open Table Format are CRUD (Create-Read-Update-Delete) operations, ACID (Atomicity-Consistency-Isolation-Durability) transactions, schema evolution, time travel and significant performance improvement in read as well as write operation.

4. All about Iceberg

Iceberg is one of the most popular open table formats backed by several major players in modern data space, most notably Snowflake. This technology brings SQL behavior into the big data world which somewhat got lost in the past decade with the advent and evolution of data lake. Video streaming giant Netflix came up with Iceberg table format in 2017 to address their internal problem with big data management. Later in 2018 Netflix open sourced it to Apache Software Foundation. The project came out from incubator status in 2020. Since then, the software community is maintaining this project under Apache.

4.1 Brief overview of architecture and capabilities

Iceberg table format is a layered metadata architecture. It contains 3 layers [Figure 1] in the overall solution - a catalog layer, a metadata layer and finally the data layer which are the open file format Parquet, ORC or Avro files. The middle metadata layer is further broken down into 3 layers of files - metadata file, manifest-list file and manifest file.

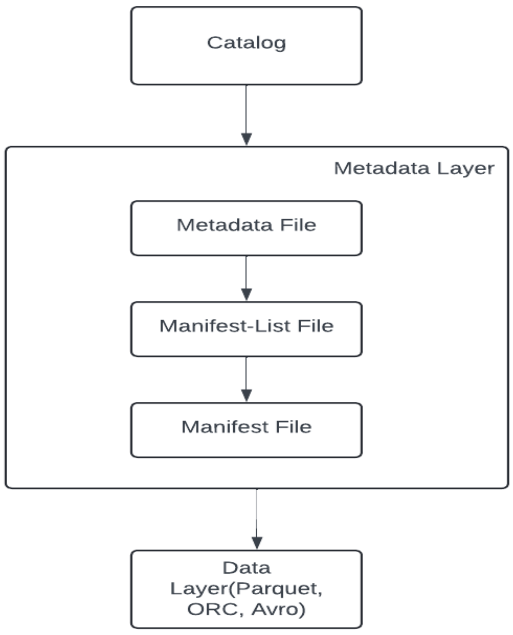


Figure-1 - Metadata architecture of Iceberg

For an in depth understanding of the architecture, please refer to the paper provided at the end of this article.

Iceberg as a table format provides several capabilities smoothing the way for increased adoption of this technology at a rapid pace in the industry:

- **ACID compliance** - data manipulations are atomic providing transaction capability
- **Hidden partitioning** - Iceberg manages partitioning internally by using data structure and metadata and users remain completely unaware of the partitioning scheme.
- **Partition Evolution** - if performance degrades over time, Iceberg changes the partitioning scheme on its own enabling partition evolution.
- **Schema evolution** - Schema evolution is straightforward in Iceberg as only metadata fields are updated. Standard operations such adding a column, dropping an existing column or renaming it are all possible in Iceberg using metadata file manipulation. The data files remain unchanged.
- **Time Travel** - Change in Iceberg causes creating a new version of metadata called snapshot. Old snapshot remains in the system for a while. This allows users to time travel over data using date range or version number of a snapshot.
- **Concurrency** - Iceberg allows concurrent reads and writes by multiple engines at the same time leveraging optimistic concurrency control. When there are multiple concurrent requests, Iceberg checks for conflicts at the file level, allowing multiple updates in a partition as long as there are no conflicts.

4.2 How about performance?

Iceberg has effectively addressed the performance limitations that traditional data lakes often encounter.

- **Read query performance**
  - Metadata & Partitioning - Search is one of the most essential use cases in database platforms. Iceberg supports low latency search of data in tables of size in petabytes. It achieves that through its hidden partitioning technique controlled by its metadata architecture.
  - File compaction - Data fragmentation and increasing number of data files are two key factors which if not managed efficiently will slow down execution of query over time in a database platform. Iceberg periodically runs a compaction job to merge small files into large ones or by merging delete files with data files. This helps in maintaining an optimal storage structure which in turn helps in fast query execution.
- **Write Query Performance** - Iceberg provides flexibility to implement two types of write strategy - copy-on-write (COW) and merge-on-read (MOR). Depending on the nature of the use cases - ready heavy system vs write heavy system, implementation can choose appropriate strategy to improve write query execution time by adopting MOR or compromise in write performance by adopting COW for read heavy systems.

4.3 Some more info on the ecosystem

Iceberg is supported through most of the common query engines such as Spark, Trino, Presto and data platforms such as Snowflake, Dremio lakehouse. Due to its flexible table format, high performance, ACID compliance and overall robust architecture, the adoption of this technology has been attractive to these platforms. All three query engines, Spark, Trino and Presto have native integration with Iceberg. Snowflake, the modern cloud data platform, allows Iceberg connectivity as external volume through its platform. Initially Snowflake supported 2 types of Iceberg implementation - native table and external tables, it has since then unified the two approaches and now offer a common solution of Iceberg through configuration. However for Iceberg integration with Snowflake, the data files must be in Parquet file format even though Iceberg in general as a technology, supports other file formats such as ORC and Avro.

Delta Lake and Apache Hudi are the other popular table formats apart from Iceberg. Delta Lake is backed by DataBricks, another key player in the modern cloud based data platform. It is hard to say with certainty which of the two table formats is the market leader, but Iceberg seems to be the industry favorite in the coming years.

Iceberg supports Git-like capabilities such as branching, tagging through integration with Project Nessie. Nessie in the big data world is synonymous with Git in source code repositories. Using the Nessie extension in Iceberg, the catalog table of Iceberg can be simultaneously updated by multiple users across different branches and later commit the changes from the individual branches to the main branch. This is a technique to accomplish multi-table transactions in Iceberg which is not natively supported otherwise. Because Nessie works like the Git version control system, it allows listing commit history and even cherry picking commits across branches if needed. This capability is not available in other table formats like Delta Lake and Hudi at this moment.

5. Industry Adoption

Iceberg is a disruptive technology which is being broadly adopted by top organizations and having tangible impacts across many sectors including finance, retail, healthcare and entertainment and even within the software industry itself [Table-1].

It is an already matured technology that is solving decade old complex problems of the big data world, in a stable and reliable manner.

Adoption	Use Cases
Adobe	Data integrity, version control and scalability of data platform
Netflix	Scalability and performance for streaming
Apple	Time travel for ML use cases, ACID for GDPR, improvement in batch reliability using Iceberg with Spark
Shopify	Scaling an interoperability of data across multiple engines in the organization
Pinterest	Cost reduction of infrastructure by cutting down cloud compute resources for recommendation, content delivery use cases
Snowflake	Allow customers to leverage their external storage having Iceberg data and provide rich capabilities and governance of Snowflake. Snowflake is the flag-bearer of this technology.

Table-1 - Adoption across industry

6. Looking ahead

In summary, by providing ACID, concurrency, interoperability, and performance for polyglot data types, Iceberg opens up data democratization and solves key issues of legacy data systems, thus establishing itself as a truly disruptive technology. It will be interesting to see how Iceberg evolves in the data space in the coming years both in terms of capability as well as acceleration in industry adoption.

7. Further References

Saha, Dipankar, Disruptor in Data Engineering - Comprehensive Review of Apache Iceberg (October 14, 2024). Available at SSRN: https://ssrn.com/abstract=4987315 or http://dx.doi.org/10.2139/ssrn.4987315

Author Bio:

Dipankar is a seasoned software professional having 20 years of work experience. For the past decade, he has been working in a technical leadership role in his current organization Royal Bank of Canada(RBC) US Wealth Management. He has been implementing impactful projects resolving customer challenges using cutting edge technologies. At present, he is producing solution architectures to innovate, solve business problems and foster growth within his organization. In his role as a Principal Solutions Architect, he collaborates with various stakeholders within and outside of his organization including but not limited to business, vendors, executives, technical leads, developers and testing team. His contribution has earned him numerous awards in his present and past organizations. The most noteworthy of them is the RBC Conference Award in 2021 which is the highest level of individual recognition for an employee at RBC. His areas of expertise are digital transformation, cloud computing, cyber security and data engineering. He is a mentor for the next generation of software engineers and plays this role within and outside of his organization. He is an IEEE Senior member and involved in various IEEE initiatives such as EPICS project proposal review, IEEE online course review etc.

EIT 2025 CFP!

ANNOUNCEMENT and CALL FOR PAPERS
2025 IEEE INTERNATIONAL CONFERENCE on ELECTRO/INFORMATION TECHNOLOGY
May 29-31, 2025
Valparaiso University
Valparaiso, IN 46383-6493 USA
http://www.eit-conference.org/eit2025

The 2025 IEEE International Conference on Electro/Information Technology (EIT) sponsored by IEEE Region 4 and hosted by Valparaiso University, is committed to advancing research and fostering innovation in the fields of electrical and computer engineering. Our mission is to provide a dynamic forum where academic researchers, industry professionals, and students discuss the latest developments, exchange ideas, and collaborate on emerging technologies. The conference aims to inspire innovation through rigorous presentations, enlightening workshops, and open dialog. Topics of interest include but are not limited to:

Robotics and Mechatronics
Intelligent Systems and Multi-agent Systems
Control Systems and System Identification
Reconfigurable and Embedded Systems
Power Systems and Power Electronics
Solid State, Consumer and Automotive Electronics
Electronic Design Automation
Biomedical Applications, Telemedicine
Biometrics and Bioinformatics
Nanotechnology
Micro Electromechanical Systems
Electric Vehicles
Wireless communications and Networking
Ad Hoc and Sensor Networks
Internet of Things
Artificial Intelligence and Machine Learning
Cybersecurity
Computer Vision
Signal/Image and Video Processing
Distributed Data Fusion and Mining
Cloud, Mobile, and Distributed Computing
Software Engineering & Middleware Architecture
Entrepreneurial Minded Learning
Engineering Education

Important dates:

- Submission of full papers: February 15, 2025
- Notification of acceptance: March 28, 2025
- Early registration deadline: April 11, 2025
- Final manuscript (PDF) due: May 2, 2025

For more information, ideas for organizing/chairing sessions, industry participation, tutorials, professional activities sessions, please contact: Dr. Sami Khorbotly or Dr. Hossein Mousavinezhad.

Note: Typical papers will be 4-6 pages, IEEE journal 2-column format, papers more than 6 pages will be charged a fee.

About R4 Social Media

R4 Media Sites

Region 4 Website
https://r4.ieee.org/

Each of the sites below may be accessed through the Website:

R4 Event Calendar
https://r4.ieee.org/events/month/

R4 Facebook Page
https://www.facebook.com/R4.IEEE/

R4 LinkedIn Page
https://www.linkedin.com/company/ieeer4/

R4 Twitter
https://twitter.com/IEEER4

R4 Instagram
https://www.instagram.com/ieeer4/

R4 YouTube Channel
https://www.youtube.com/@IEEER4

R4 Newsletter:
https://r4.ieee.org/our-newsletter/

R4 Committee Members:
https://r4.ieee.org/committees/