Saroj Niraula

Modular AI Framework for Small Software Companies

Metropolia University of Applied Sciences
Master of Engineering
Master’s Degree Programme in Information Technology
Master’s Thesis
17 October 2025

Abstract

Author(s):	Saroj Niraula
Title:	Modular AI Framework for Small Software Companies
Number of Pages:	73 pages + 2 appendices
Date:	17 October 2025
Degree:	Master of Engineering
Degree Programme:	Master's Degree Programme in Information Technology
Specialisation option:	Networking and Services
Instructor(s):	Aarne Klemetti, Supervisor

This thesis presents the conceptual design of a Modular AI Framework intended to help small and medium-sized software companies adopt artificial intelligence in a structured and sustainable way. The study applies a design science research approach to identify adoption barriers, define framework requirements, and propose a three-layer architecture consisting of Module, Integration, and Management layers. Each layer supports modularity, interoperability, and maintainability while remaining lightweight enough for small development teams.

The framework was conceptually evaluated through illustrative case scenarios, such as a natural language processing service and a model management module. These examples demonstrated how containerization, standardized APIs, and clear module specifications can reduce integration effort and long-term maintenance costs.

The results suggest that modularity enables SMEs to adopt AI incrementally, reuse components across projects, and avoid vendor lock-in. Although no full implementation was developed, the conceptual evaluation confirms the framework's potential to improve accessibility and sustainability of AI solutions for smaller organizations. The study concludes that modular architecture provides a pathway for SMEs to participate in AI innovation while maintaining technical and regulatory flexibility.

Keywords: modular AI, framework design, SMEs, interoperability, conceptual design

The originality of this thesis has been checked using Turnitin Originality Check service.

Contents

1 Introduction

While artificial intelligence (AI) has grown significantly, the pattern to how software products are designed and delivered has also changed accordingly. Large companies have enough resources to hire a dedicated AI teams and often have a supportive infrastructure, but small and medium-sized enterprises (SMEs) do not have enough resources, and they struggle to adopt AI. There could be a numerous reason for it. Limited budgets, and non-uniform technology stacks could be some that can be named. Despite having these challenges, SMEs are still expected to meet customer expectations for smarter and data-driven services.

SMEs might have tried to adopt to AI usage, but they are mostly based on solving a single problem with a single solution or, they probably integrate it in the systems in such a way that a lot needs to be changed to fix a simple bug in the integration or, they directly use third-party APIs. This might solve the immediate problem but with the use of third-party API, there is a risk of being in vendor lock-in. To solve all these problems, modular software architecture principles can be applied while integrating AI. If we could successfully package AI functionality into independent modules which have clear interfaces, SMEs can easily reuse those components and adapt to different programming environments. This help SMEs to depend on a single provider.

1.1 Research Problem

Using AI can give advantages to many SMEs over their competitors but there are practical barriers that SMEs face while trying to adapt to AI usage. Some of those barriers are that their systems are built with different technologies, they do not have technical skills that is required or they also risk getting locked into one vendor. There, however, are many existing frameworks and commercial AI services but, unfortunately, they are designed with larger enterprises in mind. This makes the services very costly and complex for SMEs. Because of this, there is a need for integration of AI in a sustainable and reusable way in small teams.

1.2 Objectives

The main objective of this thesis is to propose the conceptual design of a Modular AI Framework that addresses the specific needs of small software companies. The framework should:

- Provide modular components that can be reused across multiple projects.
- Remain lightweight enough for small teams with limited resources.
- Support interoperability across different programming languages and platforms.
- Reduce integration time and maintenance overhead compared to ad hoc solutions.

1.3 Research Questions

This thesis is guided by the following research questions:

- What architectural requirements must a modular AI framework meet to be effective for SMEs?
- How can AI functionality be packaged into modules that are interoperable across different technology stacks?
- Does a modular framework reduce integration time and improve maintainability compared to traditional approaches in SME projects?

2 Literature Review

2.1 Modularity in AI Systems

Modularity means separating modules into specific small sections which can be reused, maintained independently, and be recreated. It is a common practice in software development to use modularity. When we apply modularity to AI systems, we can break down bigger and more complex AI capabilities into smaller and reusable units. These units can then be easily integrated in different environments of different technology stacks. This design principle mainly focuses on SMEs because SMEs usually do not have enough resources to manage monolithic architectures.

Recent studies demonstrate how modularity can significantly reduce the barriers to AI adoption. For example, the AI Music Artist Toolkit (AIMAT) provides a modular, containerized environment that simplifies experimentation with generative music models by encapsulating complex dependencies in Docker containers and providing standardized interfaces via Open Sound Control (OSC) [1]. Although developed for creative practice, AIMAT illustrates how modular containerization can lower technical barriers for non-expert users, a principle equally applicable to SMEs seeking accessible AI integration.

Another example is AI4U framework, which introduces modular AI application design principles that emphasize reusable components, clear architectural guidelines, and interoperability [2]. The framework shows how modularity can enable rapid prototyping. At the same time, it also makes sure that flexibility is maintained, and change can be pursued as the business requires.

If we look at both works together, we can see how important modularity is in AI system. Modularity not only simplifies the deployment but also extends the usability of AI more than just specialized teams. So, overall, the usage of modularity reduces the integration overhead, cuts off technical debts, and increases the longevity of AI investments.

Theoretical Foundations of Modularity

The principle of modularity has been a cornerstone of software architecture for several decades. Early work established that systems should be organized based on information hiding, where each module conceals its internal logic while exposing only necessary interfaces [3]. This approach enables developers to modify or replace modules without affecting the entire system. Later research expanded on this notion, framing modularity as both a technical and economic mechanism for managing complexity, innovation, and system evolution [4].

These fundamental ideas have been the core of AI systems. Modularity allows the organization to separate concerns and keep data pipelines, algorithms, and interfaces to their own distinct components. They are interconnected to each other. Modularity, thus, enables different teams or individuals to contribute to the modules while keeping the system integrity intact.

Modular Architectures in Modern AI Development

In present day AI engineering, modularity can be seen in architectures such as MLOps pipelines and microservices-based deployments. Frameworks like TensorFlow Extended (TFX), MLFlow, and Kubeflow apply modular principles to the AI lifecycle. These frameworks divide data ingestion, feature engineering, training, and deployment into separate and distinct components. Each stage acts as an independent module which can be scaled, replaced, or reused across projects depending upon the requirements of integration.

This design philosophy supports continuous integration and continuous deployment (CI/CD) practices. Continuous integration and continuous deployment are essential for maintaining quick movement in dynamic environments. It also encourages the creation of reusable model components which can then be combined to form composite AI solutions. For SMEs, this modular structure reduces technical risk as well as implementation cost by allowing them to adopt the modules that are relevant to their needs.

Microservices and Containerization in AI Systems

Nowadays companies have adapted to the use of microservices architecture. This has supported the use of modularity in AI systems. Microservices means physically separating components across services. Using microservices, each module acts as its own deployable service. The services communicate through APIs. This design helps integrating services regardless of the technological stacks they use. This has made easier for scaling AI components.

There are different tools available that help small teams to access this approach, such as, Docker and Kubernetes. It is easier to separate concerns between development and production environments by using such containers. This particularly helps SMEs, because they often lack extensive DevOps. The pressure to acquire reliability and reproducibility in AI deployments is still there.

Challenges and Trade-offs in Modular AI Design

It is true that modular architectures offer significant benefits. But while offering benefits they also introduce design and operational trade-offs. When the number of modules increases, it becomes more complex to maintain consistent interfaces and ensuring version compatibility becomes difficult. To manage inter-module communication, it requires standardized data schemes, error handling, and documentation. This helps prevent fragmentation.

When communication between the modules is done using API, it is possible that we face latency. This can create problems for real-time applications. Not only that, to adapt containerization, it requires operational maturity. SMEs might initially lack such maturity. To meet design challenges, it is crucial that we create a balance between modular flexibility and operational simplicity. So, it is ideal that while implementing modularity, we do not create complexity in operation.

Summary

There are pros and cons of having modularity in AI systems. The pros are that they are easier to maintain, they work across multiple technology stacks, their use is flexible, reusing becomes possible. These all add up to the sustainable AI adoption in SMEs. The cons are that it can create performance challenges, it requires new coordination. But if we compare the pros to the cons, the pros far outweigh the cons.

So, it is safe to say that modularity is not only a technical method but also a practical way for SMEs to join the AI field in a flexible and scalable way.

2.2 SME Adoption of AI

When we look at the global software industry, there are several small and medium-sized enterprises (SMEs). But there have been problems for such companies to adopt artificial intelligence (AI). Large organizations have dedicated research teams. They can facilitate many sectors with huge investments. But SMEs often lack this. On top of that, having different technology stacks and a limited expertise also put SMEs in a very difficult position. This can cause problem for SMEs to adopt AI, and the integration becomes slower. They often must rely on third-party for AI integration, which increases the risk of them being locked into a single vendor.

Some case studies show that when small and medium-sized companies use AI in the right way, the results can be positive. Lewis [5] talks about four small businesses that used AI to make their work smoother and improve how they deal with customers. They were able to do this without having big data science terms. In the same way, ActivDev [6] gives examples of small companies that use AI for things like predictive maintenance, automation, and customer analytics. These examples show that simple and lightweight AI solutions can already give a lot of value to SMEs when they are designed to be easy to use and modular.

New AI platforms are also making things easier for small companies. For example, Hugging Face [7] gives access to many ready-made models that anyone can use without building big systems. This helps small teams to try out AI ideas faster and cheaper. In a similar way, ThriveAI [8] is an AI tool mentioned by Business Insider that helps product managers in their daily work. From these, it is easier to see how small companies can also get help from AI tools. The development of such tools and platforms further shows that AI is not just for larger companies. It can also be used by smaller teams who want practical ways to use AI in their own projects.

At the same time, big companies like IBM also talk about the importance of choosing the right AI framework. In their report [9], IBM explains that small companies should think carefully about how flexible and scalable a system is. They should also think how much it will cost them in the long run. This connects well with what McKinsey [10] found in their global AI survey. They noticed that while AI use is growing everywhere, small companies are still more careful and usually start with small, specific projects instead of large, complex projects.

Collectively, these sources illustrate both the opportunities and limitations of AI adoption in SMEs. The evidence suggests that SMEs benefit most from modular, flexible, and pre-integrated solutions that minimize technical expertise requirements. This reinforces the need for frameworks such as the one proposed in this thesis, that provide reusable, interoperable AI modules designed with SME constraints in mind.

Barriers to AI Adoption in SMEs

Even though many small companies already see the benefits of AI, there are still a lot of problems that make it hard for them to use it properly. The most common problem is the lack of technical skills. Many SMEs don’t have data scientists or machine learning experts who can design and manage AI models [11]. Because of that, they often depend on ready-made tolls or pre-trained models, which are easy to use but limit how much control they have.

Money is another big issue. Large companies can spend a lot of money on AI. But small companies usually must be careful with their budgets. Things like data collection, computing power, and skilled staff can be too expensive [12]. This makes it risky for SMEs to invest in AI when they are not sure about the return.

Another challenge is the mixed technology setup that many small companies have. They might use old systems together with new cloud tools, and connecting AI into that mix can be difficult. Without standard frameworks, the work easily gets duplicated or tied to one specific vendor, which makes it harder to scale.

Finally, there are also cultural and strategic issues. Some companies are still not sure what AI can really do for them, or they think it’s something only big companies need. Studies show that even if the technology is ready, AI projects often fail because management is not fully committed or there is no clear plan in place [13].

Enablers and Support Mechanisms

Even with the challenges stated above, there are now programs and tools that try to help small companies use AI. The European Union has started projects like the Digital Europe Programme and the AI Act, which aim to support responsible and safe AI use [14]. These programs are providing funding as well as guidance for companies that want to build AI systems in a trustworthy way.

Many European countries have also made their own national plans to help smaller companies. For example, Finland’s AI 4.0 Program and Germany’s Mittelstand-Digital initiative give support to small companies through funding, training, and partnerships with research groups. Looking at these efforts, it can be seen how governments, schools, and businesses can work together to make AI easier for SMEs to adopt.

Open-source communities are another big help. There are different platforms which make ready-to-use AI models. Example of such platforms are Hugging Face, MLFlow, and TensorFlow Hub. They also have clear instructions which are available for free. With these, even small teams can start experimenting with AI without needing to hire a full data science team. Because of this open and shared way of developing AI, smaller companies are getting access to the same tools that big ones use. This has helped in making innovation more equal across companies.

Role of Modularity in Overcoming Barriers

Modularity can fix a lot of the problems small and medium-sized enterprises (SMEs) have when they try to bring AI into their work. Instead of trying to build one big system all at once, they can just start small. They can build one part at a time. This makes the while process easier to handle and a lot less risky. Not only that, but it also saves the company a big budget which would otherwise be needed to hire a big technical team.

It also helps when a company uses many different tools or programming languages. If the AI is built as small parts that can talk to each other through simple APIs, then it doesn't matter what language or which platform each part uses. The company can just pick what works best for them without getting stuck with a single provider.

As the company grows, they can keep adding new parts or replace old ones without touching the rest. It's much easier to update that way. They don't have to start everything from zero again.

It's also better for teamwork, when each part is clearly explained and separate from other parts, people can focus on their own piece. There is no need to mix up the puzzle pieces and start scratching heads. Developers, data people, and managers can all work in their own areas without breaking each other's work. It keeps things smooth and saves a lot of time.

Summary

Looking at the literature, we can see what stops SMEs from adopting AI. It is mainly the combination of technical, financial and cultural challenges. However, we also saw that there are many factors that support or inspire SMEs to adopt AI. Government initiatives, open-source communities, and standardized frameworks are some that can be named. It is obvious that the software developing community is evolving and the adoption of AI has been crucial. In such a scenario, modularity can be seen as a tool that can offer technical and strategic solutions, specifically to SMEs who can then have more flexibility, reusability, and scalability to participate in AI-driven innovation.

So overall, modularity just makes things simpler. It gives small companies a way to use AI without a need to rebuild everything from scratch. They don't need to depend on one big system. It also keeps things flexible. Since it can be done in parts, it is easy to update, and easier for people in different teams to work together. It obviously needs some planning at first, but in the long run, it saves both time and money. Since we are talking a lot about budgets and how SMEs might have limited budget, this kind of structure can really make AI adoption possible for them.

2.3 Ethical and Regulatory Considerations

Adoption of AI is not only about technology or data models. There are also ethical and legal questions that can't be ignored. For small and medium-sized enterprises (SMEs), these things can be even more confusing than the technical ones. Big companies usually have full-time lawyers, compliance officers, and data protection experts who take care of all this. But SMEs probably don't. They have limited people, limited time, and often no one whose job is to read and interpret all the regulations.

AI systems can sometimes make or suggest decisions that affect real people. This might include customers, employees, or even suppliers. When AI system makes that choice, it is important to explain why AI made that choice. If the reasoning is hidden inside a "black box", then even the developers might not fully know what's going on. This is where transparency and accountability come in [15]. For larger organizations, this often means adding explainability tools or human review. But for SMEs, that's not easy. They might use prebuilt AI solutions without full control or access to how the model works.

Another challenge is related to data privacy and protection. There is a General Data Protection Regulation (GDPR) in the European Union which sets rules about how personal data can be collected, stored and used [16]. These rules must be followed. They are not optional. But many SMEs simply don't have the legal or technical knowledge to make sure their systems are following these rules. They often depend on third-party providers for that. They most probably assume that compliance comes "included" with the AI service. This creates another form of dependency and increases the risk of vendor lock-in [17].

Then there's a problem of bias. If the data used to train AI systems is biased, the outcome will be biased too [18]. For example, a model trained mostly on data from one region might not perform well for users in another region. A big company can afford to audit and retrain its models, but a small company may not even notice the issue until customers start complaining about it. Ethical design principles therefore need to be part of the AI tools themselves, not something that SMEs have to figure out separately.

The researchers have raised another point which is about accountability and liability. When an AI makes a mistake, who takes the blame? The user? The developer? The company that sold the AI? This is still being discussed at many levels, including the European AI Act [18]. These regulations are trying to define clear responsibilities, but many SMEs don't even know what the AI Act means for them yet. That uncertainty itself can slow down the adoption of AI.

So, ethics and regulation are not extra topics to think about later. They are part of the foundation. A framework that helps SMEs adopt AI should make these things easier. It should not be adding more forms to fill out, instead it should be building ethical and legal awareness into the design itself. When compliance and transparency are part of the system from the start, SMEs can use AI more confidently, without feeling lost or dependent on bigger players.

Regulatory Landscape for SMEs

In Europe, the rules about AI keep changing as it becomes more updated. The EU is trying to make sure that using AI is safe and fair for all. But while doing this, it gets more complicated for small companies to follow. The main law that is in force is called the AI Act. The AI Act goes together with GDPR. GDPR is about privacy. It already controls how data can be collected and used. The AI Act is more about how AI itself is made and used in real life.

The AI Act splits AI systems into four types: prohibited, high-risk, limited-risk, and minimal-risk [18]. For example, if an AI is used to make medical or financial decisions, then it is categorized as high-risk. If the AI is used for recommending products or helping to automate small tasks, then it is categorized as limited-risk. Minimal-risk means it's not a big deal, like a spam filter or something simple.

For SMEs, this whole thing is hard to follow. They don’t have lawyers or big compliance teams. Most of them are doing small things like chatbots or automation tools, which fall into the limited-risk side. But the moment they touch healthcare, education, or money-related stuff, the requirements go up like crazy. It’s not just more rules; it’s more time and more cost.

The EU says they’re helping. They have some websites, guides, and what they call “regulatory sandboxes”, where companies can test their AI safely before using it for real. It’s a good idea, but most small teams don’t even have time to figure out how to join one.

That’s why it makes more sense if the AI tools and frameworks already follow these rules. Then SMEs don’t need to worry about all the legal details. They can just build and use AI knowing the basics are already taken care of. That’s what will help them, not another 200-page document full of legal words nobody has time to read.

Ethical Dimensions of AI Adoption

When it comes to AI, the discussion is not only about laws or compliance. There is also the question of what is fair and right when using it. These things fall under ethics. For small and medium-sized companies, this part is just as important as the technical side, may be even more. Because for them, trust means everything. If customers lose trust, the business is gone.

The biggest problem that AI has is, it is bias. AI learns from whatever data we give it. So, if that data is one-sided, then AI will also be one-sided [16]. It’s like reins to the horses. For example, if a company trains its AI mostly with data from one place or one type of people, then it won’t work well for others. It might even give unfair results without anyone noticing at first. Big companies can fix that; they have people and tools for it. But small ones, they usually notice it only when someone points it out or complains.

Another thing is transparency. When AI gives some result, there should be a way to tell why it said that. If it rejects a loan, or chooses one person over another, there must be a reason people can understand [17]. Otherwise, it just feels like guessing, and that's when people stop trusting it.

Accountability is another point. If something goes wrong, who should we hold responsible? Is the developer who mad the model responsible? Or is the company that uses it responsible? Or is it the one that sold the tool responsible? Right now, it's still not always clear. That's why it's better to think about responsibility early, before the AI is used.

Privacy is also part of ethics. AI usually needs a lot of data, and that data often includes personal information [18]. The line between using data to improve systems and invading someone's privacy can be very thin. Companies, especially smaller ones, must be careful not to cross that line, even by accident.

There are tools that can help with this. For example, open frameworks like IBM's Fairness 360 or Google's Model Card Toolkit can check models for bias or unfair behaviour without costing much. Using such tools can help small teams keep their AI fair and open.

Ethics in AI doesn't have to be some huge extra topic. It is most of the times about being clear, fair, and not hiding how things work. The best way for SMEs to handle it is to make it part of the design from the very beginning, not something to patch later.

Compliance Challenges for SMEs

The rules around AI are meant to make things safe and fair. But for small companies, following them is not easy. Most of the time, it's not because they don't want to, but because they just don't have enough people, money, or time to deal with it.

Small companies usually don’t have lawyers or anyone who knows how to read long legal texts. The terms used in laws like GDPR, or the AI Act are often confusing. They sound clear on paper but don’t really tell what to do in real life. For example, when it says that an AI system should be “transparent” or “accountable”, it’s not always clear what that means for a small software team that just wants to build something simple.

Keeping records is another headache. The set of rules that are laid down, expect companies to document where their data is coming from, how the AI models are making decisions, and what do they use to check for bias. For small teams, that is a lot of extra work. They already have enough work just to keep the system running.

And then there’s the cost. Some AI tools fall into the “high-risk” category and might need an outside check or certification. That is expensive. Most small teams simply cannot afford it. So, they either skip it or take the risk and hope everything works fine.

To help with this, the EU has started things like the Digital Innovation Hubs. They give small companies a chance to ask questions, get training, and even test their AI safely before releasing it. It indeed is a good idea. But it still does not remove the fact that most of the SMEs are busy just trying to survive.

That is why it is easier if the tools and frameworks they try to use, already follow these rules. Then the small teams would not need to study every detail. They could just focus on building their product, knowing that the compliance part is already handled in the background for them.

Toward Responsible and Modular AI Governance

For small companies, the real challenge is to build things fast but still stay within the rules. The best way to do that is to think about responsibility and compliance while building, not after. If the system itself already takes care of these things, it saves a lot of trouble later.

In modular design, this can be built in quite easily. Simple things like keeping logs, access control, or automatic reports can already be part of the same setup. If the framework could just handle those by default, then the team does not need to further worry about them every time they try to make something new. It will just come as the part of the process.

This idea fits modularity well. Each module can take care of its own checks. For example, one module can record what data it used or how it decided, and the management layer can collect that in one place. That makes the whole system easier to follow and more transparent.

When the laws change, it is also easier to update. We just fix the part that needs it, not the whole system. For small teams, that makes a big difference.

So, responsible AI is not something separate or extra. It’s just about building things in a way that’s clear, traceable, and easy to fix when rules or needs change.

Summary

The ethical and regulatory side of AI is getting bigger and more complex. For small companies, it’s a mix of trying to follow the rules and still build something useful. Modularity helps here. If the framework already takes care of things like transparency and compliance, small teams don’t need to make it a separate task. It just happens in the background while they work.

GDPR and the AI Act are not only about the rules. They’re about making AI fair and reliable. For small companies, doing things the right way builds trust too. When the design already includes ethics and compliance in it, it becomes much easier to stay fair and be ready when the changes are coming.

2.4 Cross-Domain Perspectives on Modularity

The idea of modularity is not limited to AI or software engineering. It's used in many other areas too, like education, creative work, healthcare, and manufacturing. Looking at these examples helps us to understand how modularity makes complex systems easier to handle and how it could help small companies with AI adoption.

In education, Maček Blažeka [19] shows how modular course design works well with generative AI. For teachers, it is easier to divide lessons into smaller parts. It helps them update and adjust for different students. When combined with AI, lessons can be more personal and easier to maintain. Even though this is about education, the same idea applies elsewhere. Modularity helps non-experts to deal with complex systems by keeping things flexible and clear.

The same can be seen in creative work. The AIMAT framework [1] lets musicians use advanced AI models without needing to be programmers. It hides the complicated parts and gives artists easy tools to create with. This shows that modularity is not only about technology; it's also about giving people access to it. For small companies, this is a useful comparison. Just like teachers and musicians, small software teams also don't need to know every technical detail. They just need to make sure that the modularity handles all the complexity for them.

These cross-domain examples prove that modularity works because modularity separates complexity into simpler, smaller and manageable parts. Table 1 below summarizes examples of modular frameworks developed across different domains and highlights the current lack of SME-focused approaches.

Table 1. Comparison of modularity across domains.

Domain	**Modular Unit Example**	**Barrier Reduced**	**User Benefit**
Education	AI-assisted lesson modules	Complexity of AI lesson design	Personalized, up-to-date teaching tools
Creative arts	Dockerized AI music models (AIMAT)	Technical dependencies	Access to advanced generative models
SMEs	Reusable AI modules in frameworks	Integration & maintenance effort	Faster deployment, reduced vendor lock-in

Modularity in Healthcare Systems

Healthcare gives one of the clearest examples of modular design. Hospitals and research groups use modular information systems that keep patient records, diagnostics, and analytics as separate parts [19]. This setup allows safe data sharing and helps meet privacy laws like GDPR and HIPAA.

AI-based diagnostic tools can be added as separate modules which can be connected through secure APIs. New medical models can be installed or removed without touching the main system. This keeps it both safe and compliant.

Modularity in Industrial and Manufacturing Systems

Industrial automation has long used modular design. Concepts like Industry 4.0 depend on modular machine-control and analytics components [20]. AI-enabled manufacturing lines often consist of modular sensors and controllers that feed data to machine-learning modules for quality control or predictive maintenance.

For SMEs, modular industrial design reduces entry barriers. Instead of full digital transformation, small manufacturers can introduce targeted AI modules such as energy optimization or defect detection while maintaining compatibility with existing systems.

Modularity in Education and Knowledge Systems

Educational technology also applies modularity through learning-management systems that separate content, delivery, and analytics. This structure enables the combination of diverse educational resources and tools. Generative-AI assistants integrated as modules can automate grading or personalize content without redesigning entire curricula. Such modular integration supports transparency and keeps educators in control of pedagogy.

Modularity in Robotics and IoT Systems

Robotics and IoT are built around the same modular idea. Robots have separate modules for controlling, sensing, and tasks that talk to each other through standard interfaces [22]. Frameworks like ROS let developers change or replace one function without breaking the rest.

IoT systems use modular sensors, gateways, and analytics services as well. Small companies can add AI-based sensors which can help them take care of maintenance, energy tracking, or logistics. This will help them cut huge costs and at the same time their setup remains intact.

Summary

When we look at all these areas, we can easily see that the modularity just works. In hospitals, in factories, in schools, in robots, it basically doesn’t matter where. The idea of it remains the same. When we break things into smaller parts, it becomes easier to handle. When there are smaller parts, it is easier to fix them, and it is easier to change them as well. People can utilize their resources and their focus on their own parts. They simply do not need to worry that the whole system will be messed up when they are just concentrating on their own parts. That's what keeps things running smoothly and flexible.

For small companies, this is kind of the whole point. They don't have time or people to build huge systems. They need something that can grow little by little. If one thing breaks, they can fix that and move on. No panic, no full rebuild. That's how modular AI should be for them: simple, practical, and something that doesn't demand a full team to manage. It's not about some big theory. It's just about making it possible for small teams to use AI in a way that fits how they actually work.

2.5 Industry Trends and Tools

AI has been moving fast. The tools around it keeps changing too. For SMEs, it's not easy to stay updated, but it's also getting easier to try things because of new platforms and community tools. There's a lot happening in the industry, some of it makes AI adoption simpler, some of it just adds more confusion. But the overall direction is clear. AI is becoming more accessible, even for smaller teams.

Global surveys consistently report increasing AI adoption rates. According to McKinsey's State of AI survey [10], more than half of organizations have implemented at least one AI capability, with natural language processing, computer vision, and virtual agents among the most widely deployed. However, the report also notes that smaller companies often limit themselves to experimental or narrowly scoped projects, reflecting their cautious approach to resource allocation and risk management. The adoption gap between larger enterprises and SMEs is illustrated in Figure 1, which shows that larger organisations continue to lead in the implementation of AI technologies [10].

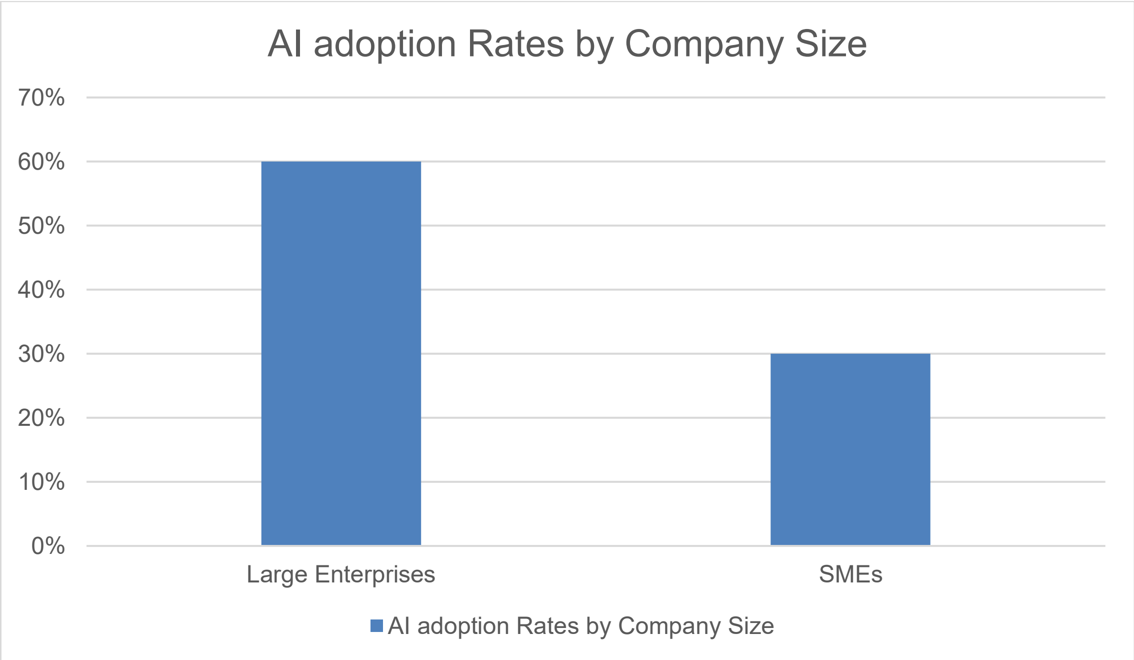

Figure 1. AI adoption rates by company size (based on McKinsey, 2025)

There are also new platforms that have made things easier for small teams. Hugging Face [7] is one of the examples. It basically gives access to ready-made models that everyone can try, and we don’t need to build anything from scratch. Because of that it is easier to test ideas and see for ourselves what works. Another example is ThriveAI [8], which was made to help product managers in their daily work. It’s a small tool but it shows how AI can be used in simple ways too. These kinds of tools prove that AI isn’t just for big companies. Even small teams can start using it when it’s made easy enough.

Big companies like IBM have also talked about the importance of choosing the right kind of AI framework. In one of their reports [9], they point out that small companies should think carefully about flexibility, scalability, and cost before picking any system. McKinsey [10] also found the same thing in their global AI survey. They noticed that while AI use is growing everywhere, small companies are still more careful and usually start with smaller, specific projects instead of going for big ones.

When we look at these trends, we can clearly see that the current AI ecosystem has both opportunities and threats. Open-source platforms and specialized startups have clearly been able to provide the access that’s needed, but it is still challenging for SMEs to integrate these systems and maintain for a long time. The modular framework, that this thesis argues upon addresses these gaps. A modular framework combines accessibility to the open-source platforms and provides reliability to the SMEs.

Open-Source Ecosystems and Democratization of AI

The growth of open-source AI tolls has changed how companies build and use artificial intelligence. Platforms like Hugging Face, TensorFlow, and PyTorch have made models, libraries, and tutorials available to everyone [7,23]. For small companies, this has been a big help. By using these they can test and use AI, and they don’t need to start from scratch. So, without the need of building a model from scratch, they can take an existing one, adjust it a bit, and see if it fits their need. This open and shared way of building AI has made development faster. It also helps small teams and large teams to look more alike.

Open platforms also make it quicker to try new ideas. With ready-made tools, small teams can build something simple and check if it works. They can connect models with APIs, do small changes, and test them in their own way. Tools like Docker or Streamlit make it easy to put things together without much setup. It’s the same basic idea of modularity. We pick the parts we need and use them. For small companies, that makes AI feel more practical and less like a huge technical job.

Open-source tools help a lot, but they also bring some problems. They give everyone access, but then the responsibility is on the user. Small teams must think about data security, about if the model is even working right, and about what happens when something breaks. Ther’s no support line to call. It’s all on them. So, it is obvious that open tools make it easier to start, but to maintain it is a challenge. That is the reason why small companies need to look for tools that already handle privacy and transparency. That way, they don’t need to deal with issues later.

Rise of Domain-Specific AI Tools

Beyond general-purpose frameworks, the AI industry has seen the emergence of domain-specific tools tailored to particular business processes. Examples include Jasper for marketing content, Synthesia for video production, and ChatGPT Enterprise for conversational automation. These tools illustrate a growing trend toward specialization, where modular AI components are optimized for narrow but high-value use cases [24].

For small companies, these kinds of tools are a good way to start using AI. They remove the need for complex setup and make things faster to deploy. But the problem is that they also limit what we can do. Most of these tools are built as closed systems. They don’t necessarily always work well with other software.

And there isn’t much we can do to change how they work. After a while we get stuck to one provider. And that’s not very ideal for small teams which basically need to remain flexible.

A modular setup can fix that. It gives small companies the chance to use these focused tools when they want to but keep everything under their own control. They can connect different parts, move things around, or replace what they don’t need anymore. It keeps them free to grow and change without starting again every time.

AI as a Service (AIaaS) and Cloud Integration

Cloud-based AI, often called AI-as-a-Service (AIaaS), has become another big way for companies to use artificial intelligence. Providers like Amazon, Microsoft, and Google offer ready AI tools for things such as language understanding, image recognition, and predictions [25]. Small companies can use these without buying any special hardware or setting up large systems. They just use what they need and pay for that. It’s a fast way to try AI and see real results without heavy investment.

Even though cloud services make things easier, they also bring some problems. The costs can go up fast because you pay for what you use. And sometimes it’s hard to predict how much that will be. There are also questions about privacy since the data goes through an external provider. It can be tricky for small companies. They need the flexibility and performance that cloud services give but at the same time, they don’t always have full control over where their data is stored or how their data is used.

A modular framework can make cloud use easier for small companies. It gives them a way to mix local and cloud-based parts depending on what works best for them. Some modules can run on the cloud, while others can stay local for safety or cost reasons. This setup keeps them flexible. They can still use powerful cloud tools but stay in control of their data and be very well within their budget.

Emerging Trends: Generative AI and Agentic Systems

The year 2024 marked an inflection point for generative AI, with large language models (LLMs) becoming central to both consumer and enterprise applications. Generative systems are now being extended into autonomous or semi-autonomous agentic architectures, capable of reasoning, planning, and executing complex workflows [26].

For small companies, these developments bring new chances to automate work, manage information, and interact with customers in smarter ways. But at the same time, generative models also make it more important to have good structure and control. They need proper design and governance so that the systems stay reliable and safe to use. Integrating them within a modular framework enables controlled use of LLMs while maintaining boundaries for data flow, monitoring, and ethical oversight.

The shift toward agentic systems suggests that future AI solutions will not be isolated components but networks of interoperable modules capable of collaboration. This further reinforces the relevance of modular AI frameworks that can manage distributed intelligence through standardized communication layers.

Summary

It’s not very difficult to see what’s happening in the AI world. Things certainly are moving very fast. There are number of tools available, there are number of platforms available, there are more examples available. And they are only heading one direction, and that’s upward. For small companies, this is a good time to explore. They no longer need big budgets or long projects to get started. There are many open-source models, ready-made APIs, and small AI services that make it possible to check the ideas right away. Most of the difficult job is done somewhere else. What is really necessary at this point is to see how they can connect to what’s actually available.

Even after having all these options, it is still not simple. Tools keep changing all the time, and small teams can easily get lost between updates, pricing, and setup. It’s also not always clear how to make different parts to talk to each other, especially when every provider does things in their own way. That exactly is somewhere where the risk comes in. It is very easy to start, but it isn’t very easy to keep it working over the course of time. And once we depend on one system too much, it will become more difficult and messier to change it later.

That’s why a modular approach makes sense here. Instead of building one big thing which totally depends upon one provider, small companies can build many small parts which fit together. Small companies can choose what they need, connect it, and if they see something better, they can replace it later. It keeps them flexible and in control. In the middle of all these new tools and trends, this is probably the best way forward. It is not about chasing everything new, but about building in a way that can adapt when things change.

2.6 Identified Gaps in the Literature

Even with all the progress in modular AI design, industry adoption, and better rules around AI, there are still a few clear gaps in what we know.

Limited SME-Focused Modular Frameworks

Most modular AI frameworks so far have been made for specific areas like music [1] or education [2]. These show that modularity can make AI easier to use, but they don't really focus on small and medium-sized software companies. There isn't much research that looks at how modular AI could fit into the kind of projects small companies handle every day. Because of that, there's still a gap between what's discussed in theory and what can actually be used in real projects by smaller teams.

Lack of Unified Architectural Standards

Although industry platforms like Hugging Face [7] and specialized tools like ThriveAI [8] have increased accessibility, these are largely designed for general use or narrow verticals. They do not provide a consistent architectural framework that SMEs can adopt across multiple projects. This limits reusability and increases the risk of fragmented adoption strategies.

Integration of Compliance and Ethics in Framework Design

Regulatory and ethical discussions have advanced significantly [15-18], but literature offering concrete design principles that embed compliance into SME-friendly frameworks is limited. SMEs often lack dedicated compliance resources, suggesting the need for frameworks that integrate data protection and ethical safeguards at the architectural level.

Lack of Actionable Technical Guidance

Reports like McKinsey [10] make it clear that small companies are still behind bigger ones when it comes to using AI. These reports show the problem very well, but they stop here. They don’t really show how small companies can fix or what kind of system they could use. What’s missing is a simple, practical framework that small teams can actually build on. Something that’s bot realistic for their technical level and still follows the main rules and standards. That’s the gap this thesis aims to look at.

Synthesis and Research Motivation

From everything discussed in this chapter, one thing is clear, i.e. modularity is not new, but it still hasn’t really reached small companies. Most of the work that’s done so far only talks about modular design in other areas. It doesn’t talk about how small software companies could actually use it in their own projects. This leaves a gap between what is already known in theory and what can be implied in practice.

The missing part is a single framework that brings these things together. Something that covers the technical side, includes data protection and ethics, and still works in real situations. That kind of setup would let small companies bring AI into their work without making things too heavy or complicated.

3 Methodology

3.1 Research Design

This thesis follows the design science research approach. The main goal is to build something useful, i.e. a conceptual modular AI framework that small software companies can actually use. The focus is on creating something practical rather than only analysing what already exists. The idea here is not only to look at existing systems, but to build something new from what has already been learned in the earlier chapters. Design science works well for this kind of study because it is about creating and shaping a solution for a real problem. It connects the understanding from research with an outcome that can be used and improved in practice.

The research process consists of three main stages:

Problem Identification

The main problems were first found by going through earlier studies and by talking with developers from small software companies. From both, it was clear that small teams face the same kind of issues again and again. They usually don’t have enough people or enough time. Their tools often don’t work well together. And they are too much dependent on a single vendor. Because of all these things, it was hard for small companies to bring AI into their work in a practical way. So, before building anything, the first step was to understand these problems properly and make sure the framework focuses on fixing them.

Artifact Design and Development

After the problems were clear, the next step was to design the actual framework. The idea was to make it modular and simple enough that small teams could use it in different situations. It follows basic software engineering ideas like modularity, interoperability, and easy maintenance. The framework includes possible parts, like a natural language processing (NLP) module and a model management part that handles AI models and their updates. The goal is to make something that helps small companies reuse what they already have and build on it, instead of starting from zero each time.

Evaluation

The framework is evaluated conceptually through small case-style examples that show how it could work in real projects. The evaluation looks at things like integration time, developer experience, and maintainability compared to less organized AI adoption methods. The research also takes insights from SME developer feedback and connects them with what was seen in the literature. Combining both perspectives gives a more complete view of how modular AI could actually help small teams in practice.

3.2 Data Collection

To make sure that the proposed framework is useful for small software companies, this study used data from different sources. The information came from both existing research and direct input from developers working in small teams. The goal was to make the framework realistic and something that connects to real work instead of staying only theoretical.

Literature Review

The base of this framework comes from what was already studied in earlier research, that we discussed in Chapter 2. Academic papers, industry reports, and regulatory documents helped to understand how modularity works, what problems small companies face with AI, and what tools are currently used. These findings shaped the design requirements and main ideas of the framework.

Developer Discussions

A few informal talks were held with developers working in small companies. These were not formal interviews but open discussions about what problems they face when trying to use AI. Many of them mentioned things like weak infrastructure, not enough expertise, and worries about maintenance once systems go live. These talks gave practical insights that helped align the framework with what really happens in small teams instead of only what is found in theory.

Case Study Projects

For testing the framework in a basic way, two small project examples were used as case study scenarios. These represent common cases where small companies want to add AI but don’t have a lot of time or budget. The cases look at things such as how long integration might take, how errors can be managed, and what kind of feedback or results developers might see based on reports from both literature and practice.

By combining these three sources, i.e. the literature, the developer insights, and the case projects, the research keeps a balance between theory and real use. This mix helps to keep the framework both grounded and flexible so that it fits the real needs of small software companies.

3.3 Framework Design Process

The design of modular AI Framework followed established principles of modular software architecture, adapted to the constraints and needs of small and medium-sized enterprises. The objective was to create a framework that is lightweight, reusable, and interoperable across different technology stacks while minimizing integration effort.

The design process consisted of three stages:

Requirement Definition

The framework requirements came from both the earlier studies and the talks with developers. The goal from the start was to make something that small companies can actually use without too much setup or cost. It needed to be practical, not heavy. Based on that, a few main points were decided.

- Modularity: The AI parts should be made as small pieces which then gives the possibility of using it again in differnt projects.
- Interoperability: The parts thus made should be able to work across different programming languages and platforms like .NET or Node.js. This gives possibility to small teams to use what they already know.
- Maintainability: Each of the part that is made should be easy to change, fix, or update. And doing that should not break the rest of the system.
- Scalability: The system should be small when we start but it still needs to be able to grow later when we need it to.
- Compliance: The design that we make should make it easy to include privacy and ethical checks, so that the companies can just follow basic rules and not care much about other GDPR rules.

These points became the main guide for the framework. They helped keep the design focused on what small teams actually need, not on adding more complexity.

Architectural Model

The framework is planned around a simple layered setup. The idea here is to keep every part separate but at the same time make them work together without any problem. Each layer takes care of it own job and when put together they form the full structure.

- Module Layer: This is where the main AI parts are found. It can include things like NLP, classification, or entity recognition. Each part works on its own and provides a clear API so other systems can use it easily.
- Integration Layer: This layer handles how everything is connected. The primary use of it is to manage communication through things like REST APIs or RPC calls. The whole point of this is to make sure that the modules communicate with each other even if they are not built in the same programming language.
- Management Layer: This layer takes care of setup, logging, and monitoring. The use of it is to make sure that the modules can be deployed, updated, or replaced without having to touch the rest of the system.

This setup keeps everything loosely connected but still organized under one structure. It's simple enough for small teams to use, but flexible enough to grow whenever needed.

Module Specification

To show how the framework can work in real use, a few example module designs are described. These examples are not full implementations but simple outlines that show what kind of structure small companies could follow.

- Input and Output Schemas: Each module should use clear JSON-based input and output formats. This keeps everything consistent and makes it easier to connect modules, no matter what language or platform is used.
- Deployment Packaging: The modules can be packed in containers such as Docker. Packing it in Docker helps to keep all the dependencies together and it also helps to avoid problems when we have to move from one system to another system.

- Documentation and Metadata: Each module should include version numbers. They should also include a short list of dependencies. It is also a good idea to include simple usage notes. This will help new developers to read through and understand how it actually works and how does it connect with other parts of the system.

The design always focuses on keeping things simple. The goal is not to build something complicated but build something that small companies can use without needing heavy infrastructure or special AI knowledge.

3.4 Conceptual Implementation

The modular AI framework in this thesis was not built as a full working system, but it was tested in concept through a few practical examples. These examples were used to show how the framework could work in small software companies while staying simple and very light to use.

Two use cases were used to show how this could look in practice: one for a natural language processing (NLP) service used for text classification, and another for model management that handles trained AI models. In both cases, the same layered structure was followed, i.e. module, integration, and management layers. This made it clear how each part can be separated, connected through APIs, and managed without affecting other parts of the system.

Container tools like Docker were also considered during the design. The use of Docker usually keeps everything packaged properly and because of that it is easier to move between different setups. The input and output were planned in JSON format. Having it consistently using JSON format can give an advantage of working across different programming languages like C# or React, and across different environments like .NET or Node.js.

These cases worked as a basic prototype which showed how the framework can stay modular, flexible, and easy to maintain. Even though it was not a real system running in a real production environment, these examples still gave a clear picture of how it could be used in a real production environment of a small software company. It also helped in preparing for the evaluation stage that followed.

3.5 Evaluation

The evaluation of the proposed modular AI Framework was carried out at a conceptual level, since no full-scale implementation was developed. The purpose of the evaluation was to determine whether the framework design meets the requirements identified in the research problem and objectives, and to assess its potential value for small and medium-sized software companies.

The evaluation followed three complementary approaches:

Case Study Scenarios

Two small project examples were used to see how the framework could work in practice. One focused on a natural language processing (NLP) service for text classification, and the other on a model management service for handling trained AI models. These examples helped to show that the framework actually supports modularity, reusability, and easy maintenance in real project settings.

Criteria-Based Assessment

The framework was also check for a few points that came up during the study of literature and from talks with developers. These points were modularity, interoperability, maintainability, scalability, and compliance. Each one was looked at separately to see if the framework really covers it. The idea was to make sure the design can handle the usual problems that small companies face, like not having enough resources, systems that don’t fit together, or being stuck with one vendor. The check was not about numbers or performance, but about whether the structure itself makes sense and solves these basic issues.

Literature and Practitioner Feedback

The framework was compared with similar modular setups discussed in research and seen in industry use. Short discussions with developers from small companies also helped confirm that the design choices make sense. They especially agreed on the importance of keeping modules lightweight, making sure they work across platforms, and using containerization for deployment.

Since this evaluation was conceptual, it did not include direct measurements or performance testing. The aim was to confirm that the design itself is realistic and that the main ideas can work for small teams in practice. The results of this evaluation form the base for the next chapter, where the final framework and its applications are described.

4 Results

4.1 Framework Overview

The modular AI framework made in this study is built to stay simple and easy to use for small software companies. The main idea of building this framework was to make something that can be reused and doesn't take too much effort to set up. It follows a few key ideas, which are: modularity, interoperability, maintainability, scalability, and compliance. These are the things small teams need if they want to use AI without making things too heavy or complicated.

The framework has three main layers, and each layer has its own clear job.

Module Layer

This is where the actual AI work happens. It includes things like NLP, text classification, or model management. Each of these is built as a separate part with simple input and output rules. That makes it easy to use them again in other projects. Keeping these parts separate also helps small teams take one step at a time. They can add new features slowly instead of trying to do everything together at once.

Integration Layer

This part is what connects everything. It is the part which handles how modules talk to each other and other systems. It uses basic methods like REST APIs or message queues. Using these basic methods makes sure that it doesn't matter which programming language is used. It can be .NET or Node.js or something else. That's very important because most of the small companies already have mixed tools and systems. And this framework should fit right into their system without changing everything that is already there.

Management Layer

This layer is the part that keeps the system steady. This layer takes care of logging, setup, and updates. If there is a need to change a module or if a module needs to be replaced, then it can be done easily without breaking other parts of the module. This setup also keeps things light, so small companies don’t need a big DevOps team or expensive infrastructure to run it.

Altogether, these three layers keep things separate but still connected. Each part mentioned above does its own job. And because of that it is easier for small teams to change or update whatever they need to change without having to touch anything else. That is the main idea which makes this whole framework very practical and flexible to use.

4.2 Module Specifications

There are a variety of modules that are supported by this framework so that, they encapsulate AI capabilities and use them as independent and reusable components. Each of the modules thus created, follow standardized specification format so that we could make sure that it runs with different programming languages and across different projects. The specification format includes input/output schemas, packaging, and interoperability features.

Specification Format

Each module is described using the following attributes:

- Purpose: Functional description of the module.
- Inputs: Data required for the module to operate.
- Outputs: Results or predictions generated by the module.
- Packaging: Deployment format, including containerization.
- Interoperability: Supported integration methods (e.g., REST API, RPC).

Example Modules

Module	Purpose	Inputs	Outputs	Packaging	Interoperability
NLP Service	Performs text classification and entity recognition for SME applications.	Raw text (JSON schemas)	Classified labels, extracted entities	Docker container	REST API, JSON I/O
Model Management Service	Handles storage, versioning, and deploymen	Model artifacts, metadata	Model registry entry,	Docker container	REST API, gRPC endpoints

Module	Purpose	Inputs	Outputs	Packagin g	Interoperabili ty
	t of trained AI models.		deployme nt status		

These examples show how we can develop, deploy and maintain modules independently. They can work across multiple platforms and use different programming languages yet be standardized through API calls and lightweight containerized environments. We can introduce additional modules without having to break through the previous ones or without changing the existing functionality. This way the framework remains modular and scalable.

4.3 Case Study Scenarios

To show how the proposed Modular AI Framework could be applied in practice, two conceptual case study scenarios were developed. These scenarios represent common situations in which small software companies might want to integrate AI functionality, but they do not have the resources to do so.

Scenario A: Text Classification in a Customer Support Tool

A small company developing a customer support platform requires the ability to automatically classify incoming messages into categories such as technical issue, billing inquiry, or general feedback.

Using the framework, the company integrates the NLP Service module. The message text is passed to the module via a JSON-based schema. After this, the classification output is returned as structured labels. Since the module is containerized, it can be deployed on the company's existing cloud infrastructure without significant overhead. The Integration Layer's REST API then feeds the classification results directly into the customer support workflow. This then helps the system to sort tickets automatically. And all this process results in faster responses and a better experience for the user.

Scenario B: Model Management for Predictive Analytics

Another small company operates a software product that provides predictive analytics for inventory management. The company develops multiple machine learning models over time. And each of them is trained on different datasets. Without proper management, it will not be easy to switch between models which makes it error-prone and difficult to maintain.

By adopting the Model Management Service, the company registers its trained models along with metadata such as version, accuracy, and date of training. The service, exposed through a REST API, allows the system to retrieve the most recent or highest-performing model at runtime. With the use of containerization, it is easy to ensure that dependencies are encapsulated. Not only that, but it also ensures that the Management Layer provides monitoring and logging features so that model usage can be tracked. With this approach, the complexity of integration is reduced, and the production system can be improved without being disrupted.

These scenarios very well explain how the framework's modularity enables SMEs to adopt AI capabilities. Instead of building a giant block in the system, companies can add or replace individual modules whenever they need them. This not only ensures flexibility, but it will also be easy to maintain them for longer term.

4.4 Key Observations

The conceptual implementation and evaluation of the Modular AI Framework highlight several important observations regarding its sustainability for small and medium-sized software companies.

Modularity Enables Reuse

Here, we separate each AI function into a separate module. This allows companies to reuse what they build. A team can create a module for a specific job and then use it again in a completely different project. Because of this, the initial development cost is spread out and there is no need of a huge budget to get started.

Interoperability Supports Heterogeneous Environments

This part solves a major headache for small companies. REST APIs and JSON can be used as connectors. So, the framework makes sure modules can talk to each other using these REST APIs. It doesn’t matter if the system is using different environments or different programming language, the modules can still communicate with each other through these REST APIs. This is perfect for SMEs because SMEs often have a mix of old and new systems and they need everything to work together.

Maintainability Reduces Long-Term Risk

When the system is separated clearly between Module, Integration, and the Management Layers, it becomes very easy to look after it. We can update or replace a single module, and we don’t need to worry about breaking the rest of the application. This eventually will reduce long-term technical debt and avoids the high costs of being locked into a single vendor. It also saves SMEs from dealing with the outdated parts.

Lightweight Design Fits SME Constraints

The framework is built to be light, using containers and simple APIs. This is key for small teams who don’t have big servers or specialized experts. It means they can actually try out AI and run experiments without needing a large DevOps team or data scientists.

Limitations of Conceptual Evaluation

Of course, this was a design on paper, not a full build. Because of that, it was not possible to measure real performance numbers like how accurate or fast it is under heavy use. The next step would be to test things in real life situations with real pilot projects in small companies.

Taken together, these points show that the framework tackles the main problems that SMEs face while trying to adopt AI. This framework gives them a flexible starting point which can grow and be tested in real life situations.

5 Discussion

5.1 Linking Results to Research Questions

This section interprets the results of the study in light of the research questions defined in Chapter 1. The purpose is to demonstrate how the proposed Modular AI Framework addresses the key issues identified for small and medium-sized software companies and to position the results within the broader research context.

Research Question 1

What architectural requirements must a modular AI framework meet to be effective for SMEs?

When we looked at what small companies actually need from an AI framework, there were few key requirements that kept coming up.

The architecture had to be modular so they could start small, interoperable to work with their mixed technology setups, maintainable so they wouldn't get stuck with technical debt, scalable for future growth, and compliant with regulations. Our three-layer design tackles each of these directly.

For modularity, we made sure each AI function lives in its own containerized modules, like having separate building blocks instead of one solid piece. For interoperability, we used standard JSON schemas and REST APIs as a common language that works across .NET, Node.js, or whatever stack a company uses. The maintainability comes from how we separated the layers, i.e. we can update one module without breaking everything else.

Even though we designed this for small teams today, we made sure the architecture wouldn't block future scaling if they grow. And for compliance, we built the management layer to easily add monitoring and audit features when regulations like the AI Act come into effect. Together, these choices match what we heard from both the literature and developers about what SMEs actually need.

Research Question 2

How can AI functionality be packaged into modules that are interoperable across different technology stacks?

The key to making AI modules work across different systems lies in three simple but important choices, i.e. clear input/output definitions, standard ways to communicate, and consistent packaging.

First, it was necessary to make sure that every module speaks a universal language. That's why use of JSON for all data coming in and going out is considered. Whether a team is using C# in .NET or Javascript in Node.js, every modern language can easily handle JSON structure. This solves the basic data problem.

But data alone isn't enough. We need a way to deliver it. That's where the standard REST API comes in. It acts like a universal plug. It doesn't matter whether there is a .NET application or a Python script or a Node.js application, anything can call it. The module doesn't care who is on the other end as long as they use the same HTTP protocol.

The final piece was making the module itself portable. By packing it into a Docker container, we bundle all its specific dependencies. We carry the exact version; we carry the exact library files and everything else it needs to run. This makes sure that it will behave the same way on a developer's Windows machine or on a cloud server running Linux.

When we put all these together, there is a powerful effect. A small company can build one module, like the text classifier, and then reuse it across completely different projects. The module becomes a true black box, i.e. reliable, portable, and most importantly, reusable without extra cost.

Research Question 3

Does a modular framework reduce integration time and improve maintainability compared to traditional approaches in SME projects?

Looking at our case studies, the answer seems to be yes. The modular approach appears to make integration faster and long-term upkeep much simpler.

In the customer support scenario, the company didn’t have to build a text classification system from the scratch. They also avoided getting locked into a specific vendor’s API. Instead, they just connected their system to the pre-built NLP module. It was more about plugging in a ready-made component than building something new.

The same logic applied to the predictive analytics case. Managing different model versions is usually a messy, manual job. But with the Model Management Service, that process became standardized. The company could easily track, version, and switch between models without worrying about breaking their live system.

There were no stopwatch tests on integration time, so there aren’t exact numbers to be given out. But the concept clearly shows that this approach avoids a lot of the custom, one-time code that creates “technical debt”. Using this framework, we are not working with a tangled mess that only developer can understand, but we are working with defined components that can be updated or replaced individually. This actually matches what we see in other studies, i.e., a modular structure leads to fewer surprises and lower maintenance costs for small teams.

Summary of Research Questions and Findings

Research Question	**Key Findings from Results**	**Supporting Evidence**
What architectural requirements must a modular AI framework meet to be effective for SMEs?	Framework structured into Module, Integration, and Management Layers; addresses modularity, interoperability, maintainability, scalability, and compliance.	Framework overview (Section 4.1); evaluation criteria (Section 3.5); literature alignment with modular principles.
How can AI functionality be packaged into modules that are interoperable across different technology stacks?	Modules packaged with Docker containers; standardized APIs (REST/gRPC); JSON-based input/output schemas ensure cross-platform use.	Module specifications (Section 4.2); case study scenarios (Section 4.3); comparison with Hugging Face and AI4U frameworks (Section 2.1, 2.5)
Does a modular framework reduce integration time and improve maintainability compared to traditional approaches in SME projects?	Conceptual evaluation suggests reduced integration effort and easier model management; modules can be replaced independently.	Case Study A (text classification) and Case Study B (model management) in Section 4.3; Key Observations (Section 4.4).

In short, the framework that is proposed in this study directly answers the main questions we started with. Even though this is a theoretical model and not a tested product, it clearly shows how a modular approach can make AI adoption a much more realistic goal for small and medium-sized software companies.

5.2 Comparison with Existing Approaches

The main idea of developing modular AI framework is to solve specific problems for SMES, i.e. how to use AI without getting locked into complex or expensive systems. To see how it actually works we need to compare what's already out there. So, we are making a comparison between what's already out there in both research and the industry and how it differs from what is proposed in this thesis.

Academic Frameworks

In academic research, there are modular AI frameworks like AIMAT and AI4U. These are great for testing new ideas in a lab setting, but they aren't really built for a small company's reality. They usually don't think about limited budgets, small teams, or the need for simple deployment.

Our approach is different because we focused on being lightweight and practical from the start. We used technologies like Docker and REST APIs that small teams can actually work with, without needing a dedicated DevOps expert. We also planned for compliance form the beginning, building a place in the management layer for rules like GDPR and the AI Act, which most research prototypes ignore.

Industry Platforms

On the commercial side, platforms like Hugging Face offer amazing AI models that can be used via an API. But their kind of modularity is about swapping models. They do not give us a whole architecture to build on. It is almost like they are selling pre-made bricks, but not the tools to build our own houses.

Then there are other heavy-duty platforms like TensorFlow Extended (TFX), Azure Machine Learning, or AWS SageMaker. These are quite powerful, but they can be too complex or too expensive for a small company. They usually tie us to a monthly subscription and a single cloud provider, which creates a risk of vendor lock-in.

Our framework tries to find a middle ground. It’s not just a collection of models, and it’s not a massive enterprise platform. It’s a vendor-neutral structure that lets a small company host and control its own AI pieces, mixing and matching what they need without depending on a single provider.

Comparative Summary

Aspect	**Academic Frameworks (e.g., AIMAT, AI4U)**	**Industry Platforms (e.g., Hugging Face, Azure ML, SageMaker)**	**Proposed Modular AI Framework**
Primary Goal	Research and experimentation	Production-scale AI services	Practical SME adoption of modular AI
Infrastructure Requirement	Moderate to high	High (cloud-based)	Low (containerized and lightweight)
Modularity level	Functional modules	Service-based, often vendor-specific	Architectural and technology-agnostic
Interoperability	Limited to specific tech stacks	High within platform, low outside	Cross-platform via REST/JSON/gRPC
Compliance Considerations	Rarely addressed	Partial (depends on vendor)	Embedded at management layer
Cost & Accessibility	Free/open but complex	Subscription-based	Open and self-managed

This comparison illustrates that while existing frameworks and platforms contribute valuable insights, none fully address the specific context of SME-scale AI adoption. The proposed framework offers a balance between academic rigor and industrial practicality; it retains the modular design of research prototypes while providing autonomy and the ability to deploy, absent in commercial solutions.

Positioning the Proposed Framework

The results of this comparison indicate that the proposed framework occupies a unique middle ground, i.e., it is more structured and adaptable than isolated industry APIs but more accessible than large-scale enterprise platforms. It also advances the academic discourse by demonstrating that modular design principles can be applied effectively even when organizational and financial resources are limited.

From a strategic perspective, this positions the framework as a reference architecture for SMEs aiming to develop or extend AI functionalities using open and maintainable principles. Its flexibility suggests potential applicability beyond SMEs as well, including startups, educational institutions, and public sector projects that share similar constraints.

5.3 Strengths of the Proposed Framework

The modular framework that's proposed in this thesis clearly gives advantages for a small team. These aren't just technical wins, instead, they are practical benefits that match how SMEs actually work.

Modularity and Reusability

The core idea of breaking everything into smaller modules turned out to be one of the biggest strengths of this framework. For a small company, it means that they don't have to pour whatever they have into one big AI project. They can start with a single and useful module, like the text classifier. If it works, then they can build another one later. And the best part is that any module they create isn't thrown away after one project. They can take that same text classifier and plug it into a different product next year, which basically spreads the initial development cost across multiple uses.

Interoperability Across Technology Stacks

A second major strength is the framework's capacity for cross-platform interoperability. We know that small companies rarely have a uniform tech stack, so it is made sure that this framework handles that messiness. By insisting on standard connectors like REST APIs and JSON, it was made so that a module doesn't care if it's being called from a .NET backend or a Node.js service. It simply works. This helps to remove one of the biggest headaches for small teams, which is, trying to make different technologies talk to each other without writing a new custom bridge every single time.

Maintainability and Flexibility

The separation of concerns between the module, integration, and management layers enhances maintainability. If a better model comes out or a module needs an update, we can just swap it out. We don't have to touch the rest of the application. This is how we avoid "technical debt", where a system becomes so tangled that no one wants to change it. It also means that companies are not trapped by one vendor’s toolset, they can always replace a part without rebuilding everything.

Lightweight and Cost-Effective Design

We deliberately avoided anything that needed heavy infrastructure. Using Docker containers and simple APIs means a team can run this on a spare server or a cheap cloud account. They don’t need to hire a DevOps expert or buy expensive licenses just to try an idea. This low cost to start is probably the most important thing for getting SMEs to actually experiment with AI instead of just talking about it.

Compliance and Governance Readiness

Dealing with regulations like GDPR or the upcoming AI Act is a major worry for small companies that can’t afford compliance teams. The framework tries to simplify this. The management layer acts as a central choke point for everything the modules do. This wasn’t just an accident, it means that when new regulations come out, a company doesn’t have to search through every part of their system to add compliance features. They have one obvious place to put things like monitoring or access rules. It’s basically about avoiding a future where compliance becomes impossibly expensive because it’s scattered everywhere.

Strategic Scalability

The framework is designed to grow with a company, not just handle more data. A startup might begin with a single module for processing customer feedback. For small company, the real test isn’t just whether the technology can scale, but whether their investment can last. Since the modules are separate pieces, adding a new AI function for a new product doesn’t mean starting from zero. The main framework stays as it is. They’re just slotting in another piece. This changes how they see AI. It stops being a temporary cost for a single project only, instead it starts being a foundation they can keep building on, year after year.

The main point of all these strengths is that they work together for one goal. They make AI practical for a small company. The framework lets them start with what they can afford today. Then it lets them grow without throwing away their first investment. This is the thing that makes this solution unique. It builds a bridge between a simple start and a complete AI system. It does it one step at a time.

5.4 Limitations of the Proposed Framework

Every project has its limits and so does this project. It is important to be clear about what this study could not do.

Conceptual Nature of the Framework

The most significant limitation is that the framework is a design on paper, not a live, running system. While the design follows good software principles, we couldn't run performance test. This means we don't have numbers to prove how fast, scalable, or robust it is in reality. The scenarios in Chapter 4 are helpful examples, but they are not data from a real implementation.

Limited Empirical Validation

Because this was a conceptual study, we couldn't do formal testing with users or collect performance metrics. The feedback from developers was useful, but it doesn't replace systematic testing. We can't say for sure how the framework would handle real-world challenges like heavy data loads or connecting to old company systems.

Scope Restricted to SMEs

The framework was designed keeping only small and medium-sized software companies in mind. This choice limits where the findings apply. Bigger companies with advanced tools might find it too simple. Very small startups might still find it too technical to set up. This was not even tested in other industries so it cannot be said that it would work along other industries like healthcare or manufacturing.

Dynamic and Evolving Technological Landscape

AI technology and regulations change very quickly. New tools and laws appear all the time. This means some design choices that were made in this study might become outdated. For example, new computing methods or standards could make our containerization approach less relevant in the future.

Resource and Time Constraints

This was a master's thesis project which was done within a limited timeframe and almost no budget. On top of that, there was no resources to build a real prototype or run pilot tests with companies. A large project would be needed to properly implement and validate the framework across different organizations.

Despite these limits, this conceptual work provides a solid starting point. It shows how modular design can help SMEs and creates a clear plan for future research to build and test a real version.

5.5 Practical Implications for SMEs

The Modular AI Framework proposed in this study is not only a theoretical construct but also a practical roadmap for small and medium-sized enterprises seeking to integrate artificial intelligence into their products and operations. The framework provides several implications for how SMEs can strategically and technically approach AI adoption while minimizing risk and maximizing sustainability.

Enabling Incremental AI Adoption

For a small company, the fear of a large, failed AI project is a major barrier. This framework tackles that by breaking the problem into smaller pieces. A team isn't forced to build a complete AI system. They can start with a single, useful module, something like a text classifier for customer emails. This first step is affordable and low risk. If that works, they can add another module later. This step-by-step approach lets them match their AI spending to their current budget and needs. It also makes it easy to test ideas on a small scale first and only expand them after seeing positive results.

Reducing Dependence on External Vendors

A common problem for small companies is getting locked into a specific AI vendor. This can happen with services that have complex pricing or only work on one cloud platform. The framework proposed in this study avoids this by being open and vendor-neutral. Since the modules are containerized, a company can choose to run them on their own servers or on any cloud provider they like. This gives them full control over their data and AI models, which cuts down long-term costs and is much better for data security, especially in industries with strict rules.

Enhancing Team Collaboration and Knowledge Retention

This standardized approach also streamlines collaboration with teams. Developers can concentrate on the technical implementation of modules, while data specialists focus on refining the AI models. This parallel work reduces dependencies and accelerates further development. On top of that, each module simplifies onboarding for new team members. This is mainly because it is self-contained in nature. Thus, instead of trying to understand a complex and interconnected system, they can learn one module at a time through its documentation dedicated to one module only.

Supporting Compliance and Ethical AI Practices

For the compliance part, the layering system we had, i.e. management layer acts as a single control point. This is very important for small companies that must follow regulations like the EU AI Act but who don’t have dedicated staffs to do so. Companies can just focus on this one layer instead of looking for compliance checks across the entire system. They can implement necessary monitoring and logging here, which makes meeting legal requirements for transparency and accountability much more manageable for a small team.

Encouraging Cross-Project Reuse and Cost Efficiency

Usually, AI projects start from zero every time. This means a lot of work gets repeated and resources are wasted. The modular framework tries to change this by encouraging reuse. A component built for one project can often be used in another project with very little change. Over time, a company can build its own library of AI modules. This library becomes a valuable asset. Reusing modules like this save money. Not only that, but it also helps get new projects started much faster.

Suggested Adoption Roadmap for SMEs

The following roadmap outlines a practical sequence through which SMEs could adopt the framework:

Phase	Objective	Key Activities	Expected Outcomes
Assessment	Identify business processes suitable for AI integration.	Conduct feasibility study, assess data availability, define scope.	Clear understanding of AI opportunities and readiness.
Pilot Implementation	Deploy one or two core modules (e.g., NLP or Model Management).	Install containerized modules, test interoperability, collect user feedback.	Functional proof-of-concept demonstrating framework feasibility.
Integration	Connect modules with existing business systems.	Implement REST APIs, integrate with databases, and refine workflows.	Seamless data flow between AI modules and core systems.
Scaling and Optimization	Expand the framework across multiple projects.	Add new modules, optimize model performance, and introduce monitoring.	Broader AI adoption across organizational units.
Governance and Maintenance	Embed compliance, documentation, and quality controls.	Introduce audit mechanisms and continuous improvement cycles.	Sustainable, compliant, and maintainable AI infrastructure.

This step-by-step plan shows that a company does not need to change everything at once or make a huge initial investment. They can treat AI adoption as a gradual process. It is more about learning and integrating new capabilities step by step and then optimizing as they go.

Long-Term Strategic Benefits

The framework helps SMEs in another important way, i.e. it builds their ability to compete over time. When a company uses this modular approach, they’re not just solving today’s problem. They’re creating a system that can easily adapt tomorrow. If the market changes or new AI tools appear, they can swap modules instead of rebuilding. The flexibility becomes a real business advantage. The approach also teaches better software habits like reusing code, writing documentation and testing thoroughly. These practices make their entire development process stronger, which helps with every project, not just AI.

In conclusion, the framework provides SMEs with a structured, realistic, and future-oriented pathway toward AI adoption. It bridges the gap between theoretical design and practical implementation. This enables smaller organizations to participate meaningfully in the evolving AI ecosystem.

5.6 Future Research Directions

The framework in this thesis offers a way to solve the main AI adoption problems for SMEs, but it is a conceptual model. This means that there are clear opportunities for further research on this framework. The next steps should focus on turning the design stated in this thesis into a working system. It should be tested with real companies. There is room for exploration on how new technologies and regulations fit into the structure.

Empirical Validation through Pilot Implementations

The most direct way to validate the framework is through a pilot implementation in a real small company. A live project would test its actual performance, scalability, and maintainability under daily operational pressures. This would provide crucial feedback on integration time and the overall developer experience. It would be important to collect specific metrics, such as time needed to deploy a module and the amount of computing resources it uses. This data would provide concrete proof of the modular approach’s benefits.

Comparative Evaluation with Alternative Frameworks

A structured comparison between this framework and other modular or service-oriented AI architectures could provide further insight. Future studies could benchmark testing using identical use cases to evaluate interoperability, latency, and cost across different approaches. Such comparative analysis would clarify the framework’s unique strengths and help refine its design principles.

Expansion of Module Library

Another important research direction is to create more types of modules. The current examples are text classification and model management, but this could be expanded. Future work could develop modules for other common needs, like recommendation systems, spotting anomalies in data, or computer vision tasks.

If these were built into an open-source library, then it would be a nice idea to create a shared resource for all SMEs. This will make it much easier for any small companies to find and use the AI components they need.

Integration with Emerging Technologies

Future work should also check newer technologies. Serverless computing is one option. Edge AI is another option. There is also federated learning. All these technologies can help in different ways. Serverless reduces server costs. Edge AI makes responses faster. Federated learning keeps data private. The research question is how to add them as modules. They must fit without making the framework more complicated. The end goal is to keep it simple for small and medium-sized enterprises.

Compliance Automation and Ethical Governance

As rules for AI get stricter, research should focus on building automatic compliance directly into the management layer. This means creating features that can keep audit trails, check for bias in models, and generate reports that explain how decisions are made. If we can develop standard, ready-to-use modules for this governance, it would help SMEs follow the law much more easily. It would also push them towards using AI in a more ethical and a transparent way from the very start.

Socio-Technical Factors and Organizational Readiness

Finally, future work must look at the human side of using this framework. The technical design is important, but it is only one piece. We need to understand how real teams in small companies actually work with these modules. This means studying what kind of training developers need to use the system effectively. It also means looking at whether management supports these kinds of technical changes. Another important area is how work processes need to adapt when introducing modular AI. The best way to understand these human factors would be through direct conversations with small business teams.

Interviews and workshops could reveal the real reasons why companies decide to adopt AI or decide against it. This research about people and organizations would complement the technical work done in this thesis.

In summary, the next goal for this research is to turn the conceptual framework into a tested, practical system. This would require collaboration between universities and software companies. Working together, they could build a collection of modular AI tools. This ecosystem would give small and medium-sized companies what they need to add AI to their products in a way that is both efficient and follows important rules for responsible AI.

6 Conclusion

The purpose of this thesis was to design a conceptual Modular AI Framework suitable for small and medium-sized software companies. What we wanted to solve were the big problems these companies keep facing when they try to use AI in their systems. There is the problem of not having enough technical people. There is the problem of tight budgets. And there is the problem of vendor lock-in that makes them dependent. Our approach was to use design science research. What we ended up with is a framework that stays lightweight. It makes sure different pieces can work together. It is built to be maintainable over time. And most importantly, it lets companies add AI features step by step, which is how small teams actually work.

6.1 Summary of the Study

The research started by looking at what was missing in existing studies about modular AI for SMEs. The literature review showed that there are many frameworks available. However, most of them are either research prototypes that cannot scale properly, or they are industrial solutions that depend too much on one company's infrastructure. This finding made it clear that a new kind of framework was needed. It had to be adaptable and open, while still being technically sound and easy to access.

For the methodology, the study used a conceptual design process. This process was guided by the main principles of software modularity and architectural abstraction. The framework was conceptualized into a three-layered structure. These three layers are: Module Layer, Integration Layer, and the Management Layer. When these layers work together, they give the necessary support for components to become independent. Not only that, but the layers also provide support for the modules to remain interoperable across different systems. On top of that, they support overall governance of the AI modules. To show how this structure would function in real situation, conceptual implementation scenarios were created and discussed. These scenarios used specific examples, such as an NLP Service for handling text and a Model Management Service for overseeing AI models. Finally, an evaluation of the framework was conducted to check its feasibility. This evaluation was based on important criteria that came from the literature review, and it also considered informal feedback that was gathered from developers during this research.

6.2 Key Findings

The findings from this study show that a modular approach offers clear advantages for small and medium-sized companies that want to start using AI. The framework that was developed lets these organizations begin with small, manageable projects. It allows them to test AI applications safely without major risk. As they grow more confident and their needs change, they can then expand their AI capabilities step by step. The design of the architecture itself promotes the reuse of components, makes the process of integration simpler, and encourages building compliance into the system from the start.

The case scenarios that were presented also demonstrated a key point, i.e. even complex AI functions can be deployed successfully without requiring companies to make large, upfront investments in new infrastructure. When we look at these results in relation to the original research question, they confirm that modularity does more than just solve technical problems. It also works as a strategic tool which enables SMEs to adopt AI in a sustainable way so that their business can grow in the long term.

6.3 Contributions of the Study

This thesis makes one primary contribution, i.e. it creates a conceptual modular AI architecture that fits the SME context. This was not done in isolation. The framework actually combines what we learn from academic research with what we see working in industry practice. What this does is bridge a clear gap that exists today. On one side we have experimental modular frameworks from research. On the other side we have commercial AI platforms from big companies. Our framework sits in the middle. It gives academic research a structured, technology-agnostic model that focuses on what SMEs need, i.e. accessibility, interoperability, and governance.

From a practical side, the study gives small and medium-sized companies a blueprint they can actually use. It shows them how to adopt AI bit by bit, not all at once. The modular structure means their small teams can reuse components they build. They can integrate these components more efficiently into their different projects. And they can maintain control over their own data and their own infrastructure. This combination is important. It shows that technical adaptability and organizational control can work together. This is how modular architectures can really help democratize AI development for small players.

6.4 Recommendations for Practice and Research

For the people working in companies, here is what the study suggests. Start by looking very carefully at what your business actually needs. It’s not necessary to do everything at once. Pick one or two AI modules that will solve your most urgent problems. It is also crucial to think about reusability from the very beginning. Good documentation is also very essential from the start. Do not forget about compliance requirements early on. If these steps are followed, then it can be made sure that the AI projects will keep working well for a long time.

For researchers who continue this work, there are several directions to think of. The framework needs proper testing in real environments. This empirical testing is very important. Another good approach is comparing this framework with other solutions. The module library should also be expanded. It needs to cover more AI capabilities than what we discussed here. The best way to do this research is through collaboration. Universities should work directly with small companies. These partnerships would validate the framework. They would also help refine its design using real feedback from actual users.

6.5 Final Reflection

This thesis shows that modularity gives SMEs a workable and scalable way to join the AI economy. The traditional problems with AI integration can be solved by using modular design principles. By traditional problems, it means high costs, technical complexity, and vendor lock-in. When organizations follow the approach suggested in this thesis, they can build innovation based on their own needs according to the resources that they have. They do not have to follow the same path as large companies with bigger budgets.

The modular AI framework that is proposed in this thesis is just a starting point. It is not a finished product. It is just a conceptual design at this point. But it does create a solid foundation for future work to build upon. The next phase should turn this architectural vision into practical solutions that companies can actually deploy. The ultimate goal here is to create tools that smaller companies can really use. If we keep continuing in this direction, we can help small enterprises do more than just survive. We can help them to compete effectively because AI is now becoming important in every industry. This would make AI innovation more accessible to everyone, not just large technology companies.

References

1 Browne E, Clemens M. AI Music Artist Toolkit (AIMAT): A modular environment for experimenting with AI in music. In: 2025 IEEE International Conference on Multimedia and Expo Workshops (ICMEW); 2025 Jun 30-Jul 4; Nantes, France. IEEE; 2025. p. 1-6. doi:10.1109/ICMEW68306.2025.11152154.

2 Woloszyn K, Turchan K, Rapala M, Piotrowski K. AI4U: Modular framework for AI application design. In: Lecture Notes in Networks and Systems. Cham: Springer; 2023. p. 56-65.

3 Parnas DL. On the Criteria to Be Used in Decomposing Systems into Modules. Communications of the ACM. 1972;15(12):1053-1058.

4 Baldwin CY, Clark KB. Design Rules: The Power of Modularity. MIT Press; 2000.

5 Lewis G. How four small businesses are getting a bang for their AI buck. Raconteur [Internet]. 2024 Feb 23 [cited 2025 Aug 12]. Available from: https://www.raconteur.net/technology/four-ai-case-studies.

6 ActivDev. Artificial intelligence for SMEs: Case studies and examples. 2025 Mar 10 [cited 2025 Aug 12]. Available from: https://www.activdev.com/en/artificial-intelligence-for-smes-case-studies-examples/.

7 Hugging Face. Hugging Face – The AI community building the future [Internet]. 2025 [cited 2025 Aug 13]. Available from: https://huggingface.co/.

8 Dhanuka I, Ngo B. ThriveAI: AI agent for product management. Business Insider. 2025 May 14 [cited 2025 Aug 13]. Available from: https://www.businessinsider.com/pitch-deck-ai-product-management-startup-thriveai-preseed-round-2025-5.

9 IBM. AI agent frameworks: Choosing the right foundation for your enterprise. IBM Think Blog. 2024 Dec 27 [cited 2025 Aug 13]. Available from: https://www.ibm.com/think/insights/top-ai-agent-frameworks.

10 McKinsey & Company. The state of AI: Global survey 2024. New York: McKinsey & Company; 2025 Mar.

11 Dwivedi YK, Hughes DL, Baabdullah AM, Ribeiro-Navarrete S, Giannakis M, Al-Debei MM, et al. Artificial Intelligence (AI): Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy. International Journal of Information Management, 2021;57:101994. doi:10.1016/j.ijinfomgt.2019.101994.

12 OECD. The Digital Transformation of SMEs. Paris: Organisation for Economic Co-operation and Development; 2021 [cited 2025 Aug 13]. Available from: https://doi.org/10.1787/bdb9256a-en.

13 European Commission. AI Watch: National Strategies on Artificial Intelligence – A European Perspective. Luxembourg: Publications Office of the European Union; 2023 [cited 2025 Aug 13]. Available from: https://data.europa.eu/doi/10.2760/38570.

14 European Commission. Digital Europe Programme 2023-2027 and Artificial Intelligence Act Overview. Brussels: European Commission; 2023 [cited 2025 Aug 13]. Available from: https://digital-strategy.ec.europa.eu/en/policies/digital-programme.

15 European Parliamentary Research Service. The impact of the General Data Protection Regulation (GDPR) on artificial intelligence [Internet]. Brussels: European Parliament; 2020 [cited 2025 Aug 14]. Available from: https://www.europarl.europa.eu/RegData/etudes/STUD/2020/641530/EPRS_STU(2020)641530_EN.pdf.

16 TechGDPR. The intersection of AI and ethics [Internet]. 2024 Sep 15 [cited 2025 Aug 14]. Available from: https://techgdpr.com/blog/intersection-of-ai-and-ethics/.

17 European Data Protection Board (EDPB). EDPB opinion on AI models: GDPR principles support responsible AI [Internet]. Brussels: EDPB; 2024 Dec 15 [cited 2025 Aug 14]. Available from: https://www.edpb.europa.eu/news/news/2024/edpb-opinion-ai-models-gdpr-principles-support-responsible-ai_en.

18 European Union. Artificial Intelligence Act – Summary [Internet]. Luxembourg: EUR-Lex; 2024 [cited 2025 Aug 14]. Available from: https://eur-lex.europa.eu/EN/legal-content/summary/artificial-intelligence-act.html.

19 Maček Blažeka K. Harnessing generative AI as an assistant in curriculum with modular approach implementation: A new frontier in teaching. In: *2025 MIPRO 48th ICT and Electronics Convention*; 2025 Jun 2–6; Opatija, Croatia. IEEE; 2025. p. 1–8. doi:10.1109/MIPRO65660.2025.11132006.

20 Riaño D, Real F, López-Vallverdú JA, Campana F, Ercolani S, Mecocci P, et al. An ontology-based personalization of health-care knowledge to support clinical decisions for chronically ill patients. Journal of Biomedical Informatics. 2012;45(3):429-446. doi:10.1016/j.jbi.2011.11.001.

21 Qin J, Liu Y, Grosvenor R. A categorical framework of manufacturing for Industry 4.0 and beyond. Procedia CIRP. 2016;52:173-178. doi:10.1016/j.procir.2016.08.002.

22 Quigley M, Gerkey BP, Smart W. Programming Robots with ROS: A Practical Introduction to the Robot Operating System. O'Reilly Media; 2022.

23 Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, et al. TensorFlow: A system for large-scale machine learning. In: Proceedings of the 12th Symposium on Operating Systems Design and Implementation (OSDI); 2016 Nov 2-4; Savannah, GA. USENIX Association; 2016. p. 265-283.

24 OpenAI. ChatGPT Enterprise: Secure AI for business [Internet]. 2024 Aug 30 [cited 2025 Aug 20]. Available from: https://openai.com/enterprise.

25 Amazon Web Services. Artificial Intelligence and Machine Learning on AWS [Internet]. 2025 [cited 2025 Aug 20]. Available from: https://aws.amazon.com/machine-learning/.

26 Shinn N, Labash B, Gopinath A. Reflexion: Language agents with verbal reinforcement learning. arXiv preprint arXiv:2303.11366. 2024. Available from: https://arxiv.org/abs/2303.11366.

Appendices

Conceptual Architecture of the Modular AI Framework

This appendix presents the conceptual architecture of the Modular AI Framework proposed in the thesis. The framework is organized into three primary layers, i.e. Module Layer, Integration Layer, and Management Layer, which together provide modularity, interoperability, and maintainability for small and medium-sized enterprises (SMEs).

- Module Layer: Contains independent AI capabilities such as natural-language processing, model management, or recommendation systems. Each module is reusable and self-contained.
- Integration Layer: Provides standardized communication between modules and external systems using REST APIs or gRPC endpoints.
- Management Layer: Handles configuration, logging, monitoring, and compliance, ensuring stable operation and facilitating updates or replacements.

This three-layered structure supports incremental AI adoption by allowing SMEs to deploy or replace individual modules without redesigning the entire system.

Example Module Specification (NLP Service)

This appendix provides a narrative example of how a single AI capability can be represented as a self-contained module within the Modular AI Framework. The example describes a Natural Language Processing (NLP) Service designed for text classification and entity recognition in SME applications.

The module can be summarized through the following characteristics:

- Purpose: Performs text classification and entity recognition to categorize and extract relevant information from unstructured text data.
- Inputs: Receives raw text in JSON format.
- Outputs: Returns structured results containing classified labels and extracted entities in JSON schema form.
- Packaging: Deployed as a Docker container that encapsulates all model dependencies and runtime libraries, ensuring portability across different environments.
- Interoperability: Communicates through standardized REST API endpoints and JSON-based input/output schemas, enabling integration with systems built on .NET, Node.js, or other technology stacks.
- Compliance Readiness: Incorporates audit logging, access control, and model-version tracking at the Management Layer to support GDPR and AI Act requirements.

This NLP service module illustrates how modular encapsulation supports reusability, interoperability, and governance alignment within the proposed framework. Similar module specifications can be developed for other components such as Model Management, Recommendation, or Anomaly Detection, according to organizational priorities.