

Chapter 1: Architecting AI-Based Systems for Scalable and Adaptive Digital Ecosystems

1.1. Introduction to AI-Based Systems

A great variety of systems are being developed using Artificial Intelligence (AI) technologies. In the context of intelligent Digital Ecosystems, the capacity of the AI-based systems to adapt dynamically is highly desirable, even in domains where scalability is an important requirement. Specifically, AI models are currently applied for the automatic control of three elements of a Digital Ecosystem environment: the integration of services from different providers, services that offer the same functionality or capability, and the exchange of data and information between the components of the ecosystem. This approach contributes to satisfying scalability requirements in Digital Ecosystems.

To face the requirements of scalability and adaptability in the control of smart Digital Ecosystems, an architectural framework that facilitates the integration of different AI-based control modules was developed. Based on it, it is possible to incorporate machine learning models, online training methods, and different data management alternatives. The services that perform the roles of Data Collector and Data Manager allow the implementation of an adequate operation of the system, collecting, storing, and providing the required data and information for the training and execution processes of any AI-based service that needs to be used. This architecture is intended to support the development of intelligent systems that require adaptability and scalability features in the context of Digital Ecosystems.

1.1.1. Overview of AI-Driven Systems

Artificial intelligence (AI) encompasses methods to implement human-like intelligent behaviors in machines, such as perception, reasoning, learning, and natural language processing. AI-driven systems are designed to build autonomous systems for human use

with varying degrees of autonomy. Achieving scalability and adaptability is crucial for AI-based systems to function within any digital ecosystem. As digital ecosystems evolve and businesses grow, user requests increase accordingly. Without well-designed infrastructures capable of scaling resources in response to growing demands, the performance of AI-based systems will inevitably degrade. Moreover, architects must address not only scalability but also adaptability to ensure that these systems can adjust readily to changes in the digital ecosystem and continue to meet user requirements. Adaptive AI-based systems process learning dynamically, introducing better concepts and knowledge to other systems operating within the digital ecosystem.

The construction of AI-based systems hinges on an architectural style selected to best meet given organizational requirements. Microservice architecture is often deemed the most suitable for developing scalable software products. Event-driven architecture is likewise employed to support scalability and adaptability in dynamic business environments. In this architectural style, distinct components generate and consume events asynchronously. The emergence of serverless computing has transformed applications into functions that execute on-demand, eschewing the traditional model of always-on, monolithic services and yielding ample capacity for dynamic resource scaling. Beyond these architectures, data management remains pivotal. The prowess of an AI-based system derives largely from the quantity and quality of data collected from source systems, along with the methods used for efficient storage and processing. Machine learning models—spanning supervised, unsupervised, and reinforcement learning—play an instrumental role in shaping the machine’s learning capabilities.

1.2. Understanding Digital Ecosystems

Similar to natural ecosystems, the co-evolution and interplay of species within digital ecosystems drive agile adaptation and novelty in complex, changing environments. Digital ecosystems can therefore be defined as distributed adaptive open socio-technical systems with properties of self-organisation, scalability, and sustainability, inspired by natural ecosystems [1,2,3]. The ecosystem-orientated architecture of Digital Business Ecosystems supports the spontaneous evolution of a “healthy” balance between supply and demand at the level of business sectors, regions, nations, and continents, supporting the evolution of business sectors, regions, nations, and continents. To enable the interplay of business sectors, the modelling of the Digital Business Ecosystem architecture must conserve real-world locality of supply and demand.

The proposed Generic Ecosystem, an abstraction of essential structures and functions of the biological ecosystems, is defined before its use as an environment for Digital Ecosystems. Generic Ecosystem Properties, modelling the variation, selection, and retention processes of any adaptive system, are identified. These define the self-

replicating agent population dynamics, evolutionary processes, and spatial interactions that underpin the formation of complex hierarchical organisations, localised populations, and the continuous adaptation of the Digital Ecosystem. The Service-Oriented Architecture and Distributed Evolutionary Computing are then introduced, before their use in a Digital Business Ecosystem is detailed.

1.2.1. Exploring the Components of Digital Ecosystems

A key source of inspiration for the Aufbau of a Digital Ecosystem is natural ecosystems, specifically their robust, self-organising properties, which allow complex organisations to emerge without a centralised or hierarchical organisation. These properties lead to the high-order organisation of a system. Natural ecosystems also provide a level of assurance about the stability, resilience, and sustainability of the energy flow. Energy cycles provide a base for the survival of the species in the ecosystem. Similarly, the information cycle for Digital Ecosystems ensures the survival of the business of the Digital Ecosystem.



Fig 1. 1 : Digital Ecosystem Inspired by Natural Ecosystems

A natural ecosystem is composed of living (biotic) and nonliving (abiotic) components. Community is a term for the living component, which consists of populations of different species interacting with each other. Similarly, the abiotic environment is composed of things such as the atmosphere and the physical resources of any particular place, for example, water and minerals [2-4]. A community, together with the abiotic environment of a place, makes up an ecosystem. Such biosystems are of great concern for the survival of the species. Similarly, companies need the support of digital and physical resources for their survival in any environment. Therefore, the support of the digital and physical resources to the community of companies through Digital Ecosystems ensures the survival of the community. The following is an abstraction of the communities of the natural ecosystems and of the communities of the businesses in any digital environment. Communities are composed of populations, and each population consists of individuals. The collection of populations makes up a community. The community of businesses makes up the business ecosystem.

1.3. Scalability in AI Systems

The preceding discussion has suggested that while platform-type AI systems are aptly scaled for rapid demand growth, those requiring mass personalization tend to underperform when faced with exponential growth in demand for new behaviors. Yet the notion of scalability is twofold; on the one hand, it refers to scaling out with growing demand (as automatic spatial scale rescaling provides). On the other hand, there is also scaling up of services, i.e., adding new intelligent services and behaviors to the system. The Architecture of Intelligence in an AI-based system determines its capability concerning the second form of scalability. For instance, intelligent behavior is more scalable in birds than in ants. Humans are, in turn, more capable of conceiving novel intelligent behavior and designing intelligent artifacts (leading to an accelerated development of Machine Intelligence). Similar conclusions hold when comparing intelligence across different areas of the neural network within an individual.

The four basic challenges with designing self-aware AI-based systems—scalability, openness, distribution,* self-capabilities, and adaptability—are independent of architecture but differ in magnitude. In this context, the Architecture of Intelligence in an AI-based system can be defined as the underlying architecture, way of designing, and mechanism governing the learning process through which it evolves its intelligent behavior. An attempt at defining a generative model for artificial Intelligence that captures different styles of learning in Biology and Machine Intelligence goes beyond the focus of this paper. Nonetheless, the concept of an Architecture of Intelligence can guide research in AI systems towards designing platform B-systems that exhibit these self-* properties.

1.3.1. Defining Scalability

While many definitions of scalable systems exist within computer science and IT, standardized interpretations or widely agreed-upon meanings remain scarce. In a purely computational context, scalability denotes the capacity of a system, network, or process to manage a growing amount of work or its potential to be enlarged to accommodate such growth. Bearing this in mind, an AI-based system requiring extra computational power to achieve a target quality of service upon an increase in workload is scalable if it is capable of acquiring those additional resources—perhaps leveraging Dynamic Resource Allocation within a cloud ecosystem—when necessary. Specific behaviours such as Idle Removal and Supplantation are also pertinent. Scalability extends beyond merely scaling upwards in size and can also involve scaling downwards. A system that obtains more computational resources to handle growing workloads is characterized as Scaling Up, whereas scaling its workload downward. It is described as Scaling Down.

Within simple operating systems like Windows, a single user is usually not permitted to open more than one instance of a program—severe word processing does not allow two Word documents to be open simultaneously—and similar constraints often apply to specific programs. Complex systems, however, should be able to handle increased workloads and, if necessary, distribute jobs across various computational units. When faced with escalating workloads, the system should be capable of processing tasks in parallel to meet escalating Quality of Service demands. This ability constitutes Service Scalability. An AI-based system that allocates tasks across the devices available in the environment can thus be termed a Service-Scalable AI-based System.

1.3.2. Challenges in Scaling AI

The section of Interest centers on specific architectural challenges that arise when deploying AI-based systems in highly distributed digital ecosystem platforms. It showcases the pragmatic factors that often inhibit the realization of these systems' full potential, instead creating unanticipated limitations that entail complex adaptation and evolution operations. Interest is therefore concentrated on real-life AI capabilities rather than the realm of theoretical and proof-of-concept research. Although these issues partially pertain to scaling, different perspectives are adopted in distinct operational contexts [1-5].

The advent of AI services in highly distributed settings, such as Smart Cities, unveils distinctive requirements that remain underexplored in scientific literature and commercial offerings. Some of these gaps arise from the proprietary nature of the underlying technologies and the innovative character of these ecosystems. The

integration of AI services within the urban infrastructure introduces ever-new demands and constraints, challenging existing paradigms and spurring new research directions.

1.3.3. Techniques for Achieving Scalability

The previous section introduced Artificial Intelligence (AI) concepts that have been proven useful for developers composing highly scalable systems. The current section discusses techniques for attaining adaptability, self-learning, and self-organization in highly scalable systems.

Achieving Adaptability in Highly Scalable Systems. Milojevic [2016] identifies the distinct properties of adaptable and non-adaptable architectures. An architecture is considered adaptable when it is context-aware, can self-configure, self-heal, and self-protect, and possesses self-development capabilities. These properties enable the architecture to self-design, self-organize, self-manage, self-contain faults, and self-recover. Within the context of Cloud Computing and highly scalable systems, the term adaptability can be refined, directed, and limited to three basic properties: the ability to self-configure (i.e., react properly to changes in the context, such as a sudden growth or peak in resource-demanding tasks), the ability to self-heal (i.e., recover internally from disrupted services and dependencies), and the ability to self-optimize (i.e., monitor operational metrics to optimize the use of available resources).

1.4. Adaptability in AI Systems

A central area to be considered when architecting AI-based systems is the adaptability aspects. Adaptability in these systems comprises four properties: tolerance, elasticity, scalability, and dynamics. Tolerance can be achieved by introducing redundancy in AI-based systems [6-8]. Elasticity indicates the ability to maintain service quality by adding or removing redundant resources when necessary, considering the defined service class of the client (SLA) requirements. Scalability is the ability to retain operation despite changes in resource consumption or demands on processing capacity. Dynamics refers to the reaction to disturbances generated by the environment, system, or even provoked by the system.

1.4.1. Concept of Adaptability

Due to volatile requirements and infrastructure in a constantly changing environment, system adaptability becomes a major concern for any Digital Decision Support System (DDSS). Adaptive DDSSs respond correctly to changes in their environment. The

environment of such a system could be any change that can influence the system, such as a business process change or an emergent user request.

UML can be used for illustrating adaptive system behavior because of its capability to achieve broad recognition and interoperability among various analysis and design tools when combined with the right set of mechanisms. These mechanisms incorporate the concepts of roles, scenarios, generic relationships, and frameworks. They provide a lightweight support and do not require changes in the UML metamodel.

1.4.2. Factors Affecting Adaptability

Adaptability focuses on how fast a system can respond to changes and can be thought of as the derivative of responsiveness. Factors influencing adaptability and resilience include scalability, degree of mobility, and the presence of self-awareness. High scalability—encompassing bandwidth and latency—provides additional capacity for replication and reallocation of components, enabling timely responses to environmental requirements [2,4,6]. The degree of mobility supported in a digital ecosystem affects its ability to rapidly position components closer to points of interest. Systems with more mobility support are capable of greater responsiveness and adaptability. Self-awareness adds supplementary functionalities and perspectives to a system, resulting in enhanced adaptability and resilience.

1.4.3. Strategies for Enhancing Adaptability

Adaptability is an important characteristic of an active digital ecosystem that enables it to react quickly to business continuity failures by scaling up or down to meet operational demands. It is necessary to enrich the proposed ecosystem with new adapted versions of the AI artefacts. It is recommended that the current ecosystem be enhanced such that the Planning phase of the MAPE-K model of the adaptive control loop supports various strategies for adaptability.

The strategies for adaptability should include:

- Horizontal scaling of AI artefact containers when the active digital ecosystem requires that more copies of an AI artefact must operate simultaneously.
- Vertical scaling of AI artefact containers when the active digital ecosystem requires that an AI artefact must allocate the resources of its container dynamically.
- Adapted versions of current AI artefacts should ensure that the active digital ecosystem operates with new versions that, ideally, improve services.



Fig 1 . 2 : Adaptability in Digital Ecosystems

1.5. Architectural Frameworks for AI Systems

The CoCoSim Arguments Model proposes a Meta-model, a Finite Domain Linear Temporal Logic formula generator, and a satisfiability checker. The Meta-model allows the representation of structured natural-language argument descriptions. An example illustrates how a LINQ-generated test requires the Properties of several other tests. The corresponding Finite Domain Linear Temporal Logic formulas are deduced from the Meta-model description.

Achieving the great promises made by AI requires sharing the benefits of AI with as many people as possible. Ensuring that AI is accessible to ordinary people is fundamental—whether it is in education, taking leadership positions, or running their businesses. Increasing the number of human beings who can benefit from AI is rapidly becoming a moral imperative. A society in which the benefits of AI are shared by only

a few is one that will lead to a great deal of discontent. In a world in which progress is accelerated through AI, access to education and enterprise becomes even more important than ever.

1.5.1. Microservices Architecture

New-generation digital platforms are characterized by their high degree of scalability, agility, cost-effective resilience to failures, and dynamic nature. These requirements point to the adoption of cloud-native solutions, which generally rely on microservice-based ecosystems for their architecture and deployment. Services comprise an application design pattern built around discrete aspects of a business function that interact through well-defined interfaces and protocols. This approach to service orientation enables the partitioning of complex applications into small independent services and hence provides a degree of modularity and scalability that is not possible when creating monolithic applications. For these reasons, microservices have become the prevalent architectural choice for the evolution of existing service ecosystems and the implementation of new-generation commercial services. The microservices architecture style promotes the design of complex applications as suites of independently deployable services that are loosely coupled and can be implemented in any programming language and evolve independently. These capabilities enable continuous delivery and deployment of complex applications and help in the alignment with business objectives through the autonomous organization around business capabilities. Thus, microservice-based software architectures reduce the gap between software development and marketing teams.

1.5.2. Event-Driven Architecture

Cloud-native systems rely heavily on event-driven system architectures that promote loose coupling between services and enable the development of highly scalable and adaptive systems. At the heart of event-driven cloud-native systems lie event brokers, which deliver messages among event producers and consumers via publish-subscribe mechanisms. They provide asynchronous messaging that allows decoupling in time, space, and synchronization, and enable powerful semantic features such as topic-based and type-based filtering of events.

Event brokers are distinguished by architectural choices such as the topology of their broker networks, the associated routing strategies, the delivery semantics, and the size of an event backlog. Some designs feature a decoupled topology with distributed and loosely coupled brokers disseminating events among each other. This topology supports a many-to-many communication semantics that complements the publish-subscribe

functions of the individual brokers. However, decoupled topologies also pose additional design challenges. In particular, routing strategies are required to efficiently map subscriptions onto producers and consumers across the broker network. Provide state-of-the-art knowledge regarding the design of scalable and adaptive event-driven cloud-native systems.

1.5.3. Serverless Architecture

Serverless architecture is defined as the execution model of cloud computing in which the cloud provider dynamically manages the allocation and provisioning of servers. The name is a misnomer, as servers are still used by the cloud provider to execute code. Serverless code runs in containers that are ephemeral and event-triggered. Application code is usually packaged and deployed as one or more functions, as in popular Function-as-a-Service (FaaS) offerings, or as one or more containers, as in Backend-as-a-Service (BaaS) offerings [7-9].

Schlag et al. argue that the computational model underlying serverless functions abstracts out many low-level concerns that have traditionally been the responsibility of software architects, projecting the complications of concurrency, scaling, fault-tolerance, and security to the architects of serverless applications. Consequently, it pays to take a closer look at these underlying concerns in order to better understand what they imply for architectures of serverless applications.

1.6. Data Management in AI Systems

Intelligent systems require the management of three forms of data: the model, the input data, and the product of the intelligent behavior. In the Intelligent Localization System case, the learning model (m), the snapshot to which new data is aligned (s), and the transformed point cloud (T). The enterprise of data management involves the allocation of these forms of data to resource groups. Essentially, m was associated with the cloud, given its compute resource requirements, s was allocated to the local network, given its association with a physical location, and T was colocated with the service requesting the data, given the utility of the information for business processes.

By revisiting the four digital ecosystem functions through the lens of data management, it becomes evident that they are essential capabilities that foster the evolution of structures that support the implementation of various service allocators. These structures, carefully crafted to optimize the performance of each function, manage resource allocation for corresponding combinations of m , s , and T . Within the Intelligent Localization System, several functions regulate the handling of the three data forms:

Resource Management (m) governs the learning model; Data Management (s), the snapshot; Load Management (T), the transformed point cloud; and Adaptability Management (m, s, T) oversees all three. Depending on the service allocator employed, a single data-centred function or a complementary combination addresses allocation requirements.

1.6.1. Data Collection Techniques

The development of AI-powered digital ecosystems, such as supply network resilience experts and crowd-sourced disaster management tools, benefits greatly from incorporating structured and disparate data and knowledge sources. Data acquisition techniques may leverage natural language processing, web scraping, and semantic technologies to integrate machine-interpretable semantics. Public APIs can also provide additional information. Suitable styles and patterns that enable loose coupling and scalability with such external data sources are crucial.

For example, supply network data and knowledge are often distributed across disparate sources and processes. The SnD resilience expert architecture aims to leverage ML techniques for correlations in production and supply network data, such as data from various network tiers and external information like adverse weather conditions.

1.6.2. Data Storage Solutions

Data Storage Solutions

The continuous growth of data volume in current and future information systems places increasingly heavy demands on information storage. Adaptive systems need more resource expansion to accommodate more learners, organizational users, and learners' historical interaction data. When performing large-scale distributed collaborative computing, the system requires massive operational data for model parameter updates. Data storage—designed in a distributed manner with increased storage and query efficiency—must scale accordingly to meet changing quantities. B/s construct services involve data storage and analyzing learners' content on digital learning platforms. Additionally, data storage is fundamental for supporting the adaptive data of self-regulating adaptive systems [5,7,8].

Technologies related to self-regulating adaptive system data storage include both an operational data store for collaborative computing in data analysis and data mining, and a digital learning platform data store for storing learners' digital content. In the first case, large-scale distributed collaborative computing requires massive user operational data for model training; distributed data management, replication, modification, and deletion

present challenges. In the second, learners require sufficient digital content for intelligent analysis and data mining, while server and database architectures must efficiently adapt to complex business requirements. An adaptive system can harness a B/S architecture to provide users with operational services, positing services defined by intelligent classification of large-scale content and subsequent services for disseminating knowledge or skills. However, with continuous user growth, adaptive storage for digital learning platforms demands enhanced storage and query capacity.



Fig 1 . 3 : Adaptive Data Storage for Self-Regulating Systems

1.6.3. Data Processing Frameworks

Data processing frameworks integrate various types of data processing engines with data storage and messaging systems to form a processing-oriented cohesive framework. Each processing engine specializes in a particular processing pattern optimized for specific workloads and use cases. To address the running costs of the overall data processing, cloud providers have diversified their cloud offerings with different services—

homogeneous services, heterogeneous services, and emerging lightweight services. Services in data store systems also tend to support multiple data access patterns in a scalable and elastic manner. These efforts guarantee the cost-efficiency of both the cloud platform and the hosted Data Engineering (DE) applications. They can be divided into cluster resource managers and job schedulers, which apply to all workloads, and service-specific resource managers and job schedulers, designed for specific engines, support multi-engine workloads, or target emerging Kubernetes platforms.

While existing scheduling solutions leverage the knowledge of job structures for performance optimization, they have limited consideration of the underlying heterogeneous cloud infrastructures or diverse workload characteristics. The variation in cost—when using services of varying capabilities and runtimes—is usually a key factor in the execution of batch-oriented DE jobs. Runtime estimation is a vital part of dataflow scheduling. Dataflow jobs can be broken down into tasks of different structures, and heuristic or dynamic programming methods can be employed to calculate the shortest runtime. Runtime prediction also plays a central role in data engineering on shared infrastructure, where runtime impacts the performance and cost of executions [5,10-11]. Architectures that support multiple types of batch workloads and architectures focusing on optimizations tailored to MapReduce batch processing have been proposed. "Job-aware" cluster managers use job-level information to schedule tasks, improving performance and efficiency. Resource management on emerging lightweight services, such as Apache Spark on AWS Fargate, has been explored.

1.7. Machine Learning Models in Digital Ecosystems

Digital Ecosystems are integrated technology environments that act as distributed engines of intelligent services within dynamic, scalable, and adaptive complex adaptive systems. Empirical evidence based on the Scaled Agile Framework (SAFe®) suggests that AI-driven engineering can improve business integration, business agility, software quality, inspection efficiency, and predictability, thereby accelerating the transformation into a digital business. The key question is: Which AI techniques, services, and capabilities achieve the greatest gain at different stages of business transformation?

Machine learning models constitute automated systems for predicting the values of variables from the available data. These predictive systems can recommend business decisions by using data to train a model based on experience. Specifically, a trained model can be used to estimate the values of the target property to support the decision-making process. Classification explores the relationships between a category value of a predetermined attribute in a data set and the values of other variables to predict the category for future instances. Consequently, machine learning models properly trained can be used to estimate the values of performance metrics, predict risk factors, and

evaluate other measures related to testing. These models cover the subproblems of categorization, ranking, and rating.

1.7.1. Supervised Learning Models

Supervised learning models are the most prevalently utilized algorithms today in many fields, showcasing impressive results with very large datasets. They are implemented using feedforward neural networks that are trained using backpropagation on a dataset comprising many examples. During each training iteration, the network receives an image and provides a class score for each possible class. The prediction is then compared to the image label, and an error is calculated against the ground truth. This difference is propagated back through the network to adjust the weights, with the goal of minimizing the error. This training procedure is inspired by the real-time adjustment of the animal brain's synaptic structures during life. Over time, the network becomes an expert on the dataset it has been trained on.

However, this approach has a drawback: the larger the dataset is, the longer the training time. Similar to a car engine, the network's performance depends on the learning rate parameter. Before training, network weights must be initialized, as indicated by their name. During operation, a lower learning rate can increase accuracy but reduce the network's ability to learn new categories. This constraint is also present in the animal brain; mature brains excel at pattern recognition but struggle with acquiring new knowledge, thereby affecting scalability.

1.7.2. Unsupervised Learning Models

Unsupervised learning or pattern discovery includes raw or unstructured data, such as clustering, where the training vectors have no class labels and the algorithms train a model to group the training vectors into clusters. The output is either a linear projection or a mapping of each training vector to a cluster. Clustering is one of the most used unsupervised learning methods. Unlike classification, the clusters that are represented as groups of patterns may only be used for descriptive purposes without any subsequent interpretation or labeling. An often used goal is dimensionality reduction, such as in Kohonen self-organizing maps or cluster graph trees. The two main reasons for using unsupervised learning are dealing with data stored in very large volumes or multidimensionality or variable spaces, and the patterns found by unsupervised learning methods may provide knowledge about the data and its associations that was not known before.

Techniques used in unsupervised learning include competitive learning, adaptive resonance theory, Hamerly K-means algorithm, self-organizing maps, and simple clustering methods based on Voronoi diagrams. Autoassociative neural networks can be considered a hybrid since they optimize an error function during training, but training vectors have identical input and output components. They are used for clustering due to their dimensionality reduction capabilities in hidden layers, as in the context of signature verification. At the same time, ANNs can perform unsupervised learning, although the training of the systems is mainly associated with their use in pattern recognition.

1.7.3. Reinforcement Learning Models

Digital ecosystems leverage cooperation and competition to ensure the growth of the service platform and ensure the ecosystem stays in balance by using resources efficiently. In highly dynamic service ecosystems, the future state and strategy of other agents are unknown or difficult to predict. Therefore, agents must have cognitive abilities to continuously adapt their behaviors to current situations—to maintain the balance between cooperation and competition—thus ensuring smooth execution of services and platform development. In an SGFE modeling framework, the service ecosystem is modeled as a game. Based on state observations, an agent determines its strategy by learning an approximate value function, which reflects the contributions of its behaviors toward expected future profits [1,5,7].

Reinforcement-learning (RL) approaches are promising for scheme design and decision-making in complex and dynamic service ecosystems. By constructing reward functions, the objectives of agents are translated to the reinforcement signals, enabling the associations of current behaviors with the profits gained in a long-term process. Time-varying environment states and the unknown strategies of other agents can be reflected in the observations used by the learning algorithm. Therefore, agents with RL capabilities can make adaptively cooperative–competitive decisions, based on their real-time interests, and are capable of naturally handling the exploration–exploitation conflict.

1.8. Future Trends in AI-Based Systems

The next development stage of smart cyber-physical ecosystem–oriented systems will integrate on a deep semantic level AI methods and tools for smart services in various orders and levels with smart ecosystems. In these newly defined ecosystems, AI models work with human-centric and community-related data to provide the required smart services for the community. Shortly, the structure of an AI-based smart digital ecosystem–oriented system will be an ecosystem of ecosystems because it will integrate

entities such as IoT, big data, cloud, edge, fog, and so on, from other ecosystems. A significant trend will develop because multiple systems will integrate the AI methods and tools to provide smart services in a complete, efficient, and effective decision-making process for the community involved. Even though many new smart ecosystems will arise for specific domains, at present, these systems lack the provided AI methods and tools.

The AI ecosystem services will create smart services oriented to community progress and management. Each AI smart service will employ specific sensors from the cyber-physical ecosystem and the community and will be characterized by a certain periodicity and geographical area, all of which will be defined by the AI models associated with the service. A related and very important trend for a smart community is AI ecosystem services. AI is one of the key technologies for the smart society, smart city, and smart country of the future. The implementation of AI methods tackles community problems such as climate change, e-health during the pandemic crisis, or a deteriorated economy.

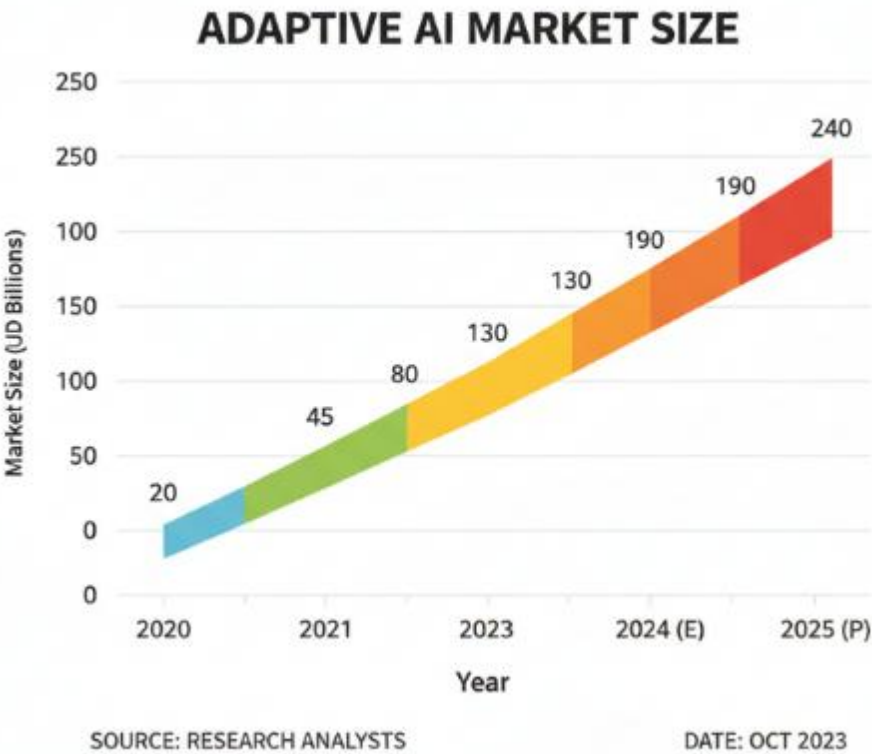


Fig 1 . 4 : Adaptive AI Market Size

1.8.1. Emerging Technologies

From Brain Coevolution to Digital Ecosystems

Processes from biology and natural ecosystems can also be applied to computing and software architecture. Ecosystems have inspired applied digital ecosystems, because they are robust, self-organising, and scalable architectures that can automatically solve complex, dynamic problems. These properties come from the underlying organisations of ecosystems—acting to provide complex solutions for the survival of the species within the ecosystem—that can be used to solve complex, dynamic problems in artificial systems. Moreover, biological systems are characterised by being energy-efficient because they use natural energy sources such as photosynthesis, internal body heat, and natural movement within the environment, as well as requiring little maintenance over thousands of years of natural evolutionary development.

Processes from biology and natural ecosystems can also be applied to computing and software architecture. Ecosystems have inspired applied digital ecosystems, because they are robust, self-organising, and scalable architectures that can automatically solve complex, dynamic problems [9,11-12]. These properties come from the underlying organisations of ecosystems—acting to provide complex solutions for the survival of the species within the ecosystem—that can be used to solve complex, dynamic problems in artificial systems. Moreover, biological systems are characterised by being energy-efficient because they use natural energy sources such as photosynthesis, internal body heat, and natural movement within the environment, as well as requiring little maintenance over thousands of years of natural evolutionary development.

1.8.2. Predictions for AI Evolution

One way to look at the progress of AI is to look at the timeline of the milestones and estimate prediction curves. Whether that is Moore’s law, plotting the number of elements per integrated circuit over time, or the number of operations per second of the most powerful computer at a given year, they are good at revealing general trends. Other models focus on larger datasets and more complex quantum computers.

Although the MIT AI professor Eugene Charniak stated in 2019 that “six of the ten following predictions will be fulfilled in the next 10 years,” this list cannot be regarded as a reliable or scientific forecast. It does serve as mental preparation for the upcoming AI changes, nonetheless.

1.9. Conclusion

Successfully operating AI-based systems within dynamic ecosystems comprises real-time analysis, flexible adaptation, and designing novel ecosystem components. Continuous adaptation addresses ecosystem evolution and unexpected system situations, enabled by AI learning and self-adaptation techniques. Human-in-the-loop supports expert knowledge integration, while marketplace-oriented mechanisms consider the vendor perspective in dynamic ecosystem environments.

The Digital Business Ecosystem (DBE) fosters Digital Product Innovation through the Digital Product Innovation Ecosystem (DPIE) sub-ecosystem, implementing open collaboration and integration functionalities. The AI Solutions Marketplace sub-ecosystem supports Diversified AI Supply through Collaboration and Competition among AI actors. Finally, the AI-based Digital Enterprise sub-ecosystem enables Digital Enterprise Growth by connecting AI Supply and Demand. These sub-ecosystems collectively contribute to diversified digitalization within enterprises and respond to multiple industrial demands for intelligent digital transformation.

1.9.1. Final Thoughts and Key Takeaways

Adaptability and scalability are essential characteristics of AI-based digital ecosystem architectures. The emerging trends in AI-based digital ecosystems cater to diverse application scenarios, exhibiting a high degree of adaptability and dynamic, scalable attributes. An LLM-CoT approach is envisioned to effectively construct such systems, where the Chain of Thought methodology features dynamic branching, enabling the ecosystem framework to evolve organically according to application requirements. The ecosystem's dynamic, scalable model incorporates three perspectives: assignment scalability, resource scalability, and architecture scalability, which correspond to three levels of AI Coach judgements—task-oriented, resource-oriented, and actor-oriented, respectively.

The Assignment Scalability (AS) model dynamically assigns and decomposes tasks based on the LLM Coaches' assessment of dependencies among AI Agents and tasks. The Resource Scalability (RS) model determines resource allocation for each task group and AI Agent according to environmental demands and resource availability. The Architecture Scalability (AtS) model constructs the AI Agent ecosystem architecture by enabling dynamic incremental and decremental adjustments in response to the constantly changing application environment. Moving forward, one principal challenge of the LLM-CoT approach for AI-based digital ecosystem architecting involves managing the recursive composability among system actors, thus preserving the adaptability and scalability of the constructed ecosystems.

References

- [1] Kalisetty, S. Leveraging Cloud Computing and Big Data Analytics for Resilient Supply Chain Optimization in Retail and Manufacturing: A Framework for Disruption Management.
- [2] Challa, K., Sriram, H. K., & Gadi, A. L. (2025). Leveraging AI, ML, and Gen AI in Automotive and Financial Services: Data-driven Approaches to Insurance, Payments, Identity Protection, and Sustainable Innovation.
- [3] AI-Powered Fraud Detection Systems in Professional and Contractors Insurance Claims. (2024). IJIREEICE, 12(12). <https://doi.org/10.17148/ijireeice.2024.121206>
- [4] Mahesh Recharla , Sai Teja Nuka. (2025). Translational Approaches To Commercializing Neurodegenerative Therapies: Bridging Laboratory Research With Clinical Practice. South Eastern European Journal of Public Health, 121–144. <https://doi.org/10.70135/seejph.vi.6488>
- [5] AI-Based Testing Frameworks for Next-Generation Semiconductor Devices. (2025). MSW Management Journal, 34(2), 1272-1294.
- [6] Mashetty, S., Challa, S. R., ADUSUPALLI, B., Singireddy, J., & Paleti, S. (2024). Intelligent Technologies for Modern Financial Ecosystems: Transforming Housing Finance, Risk Management, and Advisory Services Through Advanced Analytics and Secure Cloud Solutions. Risk Management, and Advisory Services Through Advanced Analytics and Secure Cloud Solutions (December 12, 2024).
- [7] Balaji Adusupalli. (2025). Integrated Financial Ecosystems: AI-Driven Innovations in Taxation, Insurance, Mortgage Analytics, and Community Investment Through Cloud, Big Data, and Advanced Data Engineering. Journal of Information Systems Engineering and Management, 10(36s), 1103–1117. <https://doi.org/10.52783/jisem.v10i36s.6709>
- [8] Kishore Challa. (2025). AI and Cloud-Driven Transformation in Finance, Insurance, and the Automotive Ecosystem: A Multi-Sectoral Framework for Credit Risk, Mobility Services, and Consumer Protection. Journal of Information Systems Engineering and Management, 10(36s), 1084–1102. <https://doi.org/10.52783/jisem.v10i36s.6706>
- [9] Koppolu, H. K. R., Munnangi, A. S. M., Nayeem, S. M., Ravulapalli, L. T., & Mukkamalla, B. R. (2025). AI-Aided Prioritisation with Physics-Based Validation: MD/MM-PBSA of Antiviral Binding in SARS-CoV-2 and Monkeypox. Journal of Marketing & Social Research, 2, 223-235.
- [10] Mashetty, S., Malempati, M., Paleti, S., Adusupalli, B., & Singireddy, J. (2025). A Multidisciplinary Framework for AI and Data-Driven Transformation in Taxation, Insurance, Mortgage Financing, and Financial Advisory: Integrating Cloud Computing, Deep Learning, and Agentic AI for Community-Centric Economic Development. Insurance, Mortgage Financing, and Financial Advisory: Integrating Cloud Computing, Deep Learning, and Agentic AI for Community-Centric Economic Development (March 10, 2025).
- [11] Singireddy, S. (2025, May). AI-Driven Comprehensive Insurance and AAA Membership Benefits Overview. In 2025 2nd International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMKMATE) (pp. 1-13). IEEE.
- [12] Kummari, D. N., Challa, S. R., Pamisetty, V., Motamary, S., & Meda, R. (2025). Unifying Temporal Reasoning and Agentic Machine Learning: A Framework for Proactive Fault Detection in Dynamic, Data-Intensive Environments. Metallurgical and Materials Engineering, 31(4), 552-568.