Deep Science Publishing
https://doi.org/10.70593/978-93-49307-44-5_11

# Chapter 11: Connecting enterprise systems through unified data layers, APIs, and intelligent agent-based platforms

## 11.1. Introduction

Due to the widespread proliferation of knowledge and data circulated on the Internet, quite a few devices are equipped with sensors, which generate data from a number of sources, including but not limited to social networks, cellular networks, geocode satellites, WiFi location identifiers, hardware diagnostics, vehicle movement records, device internet addresses, and intelligent building systems. This voluminous data is often referred to as Big Data, implying they carry a great potential, motivational value, and knowledge capacity indispensable for our upcoming generations. High value is hidden in this data, thus extracting such knowledge is essential for data-driven development, enhancement, automated and intelligent activities. However, classic tools and strategies for data gathering, processing, and distribution are proven to be inefficient, as the built frameworks cannot handle incoming data in real-time or provide useful responses relevant to the data. Hence, the notion of systems is built to obtain intelligent agents that automatically search, gather, reason, analyze, interpret, and filter the needed data. The ultimate goal of this work is to integrate big data and deploy pools of intelligent agents in a cloud-based environment. Tools capable of acquiring real-time data from any number of diverse, heterogeneous, and autonomous sources are essential.

The challenges are simplified with some constraints: First, the objective of the system is to provide the data- and task-oriented workflow for collecting and integrating data from a wide range of diverse services. Secondly, the user is separated from the actual data providers by the abstract type system and agents that operate on it. The presented system is based on a loosely-coupled modular agent-based structure. The main reason for employing this approach is clear separation of concerns. The system has to support both artful statistics and basic data properties. Furthermore, there is no constraint on the produced results, so the intelligence cannot be inferred and is kept in a hidden manner. Such topics as user generic queries, resulting from the exploration process, as well as ignorance-based interactions model, cannot be expressed explicitly in a single SQL-like query. As a consequence, the basic functional specifications for agents managing the input data stream, and user-interacting agents being less autarchic, are quite similar. On the other hand, much of the intelligence needed to provide the required usability and access transparency has to be put into the agents and their cooperation.

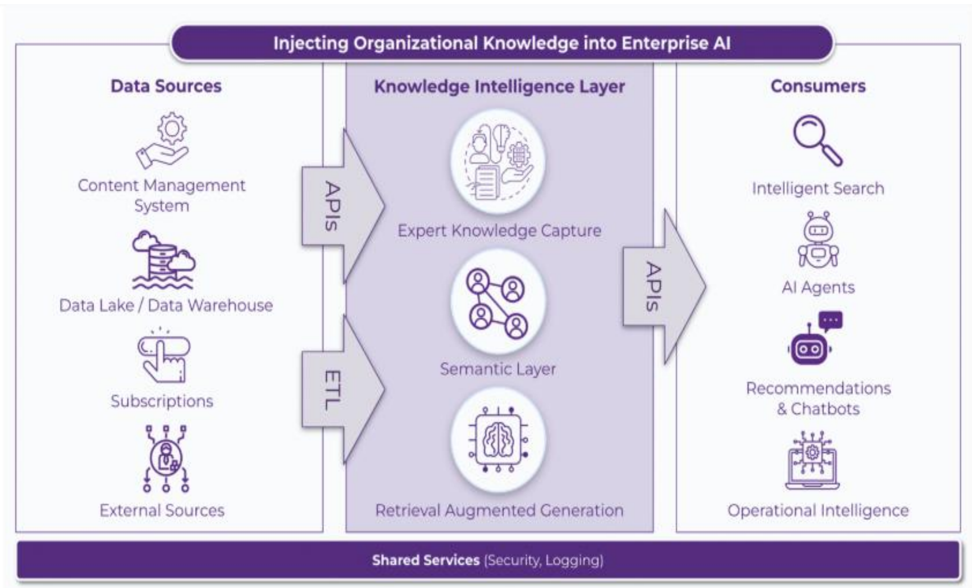


**Fig 11.1:** Enterprise AI Architecture Series

### 11.1.1. Background and Significance

Interoperability, in a broad sense, is the ability to communicate, understand and use data and information, independently of the source and the format. Typical examples of systems for which interoperability is crucial include the internet of things, e-health systems, national railways networks, and automotive modes. This notion has been one of the most important but least resolved challenges in information systems. It concerns the validity of the message sent from one system (the origin) to the message being interpreted and used in another system (the destination). It can be seen from the utterly different algorithms implied in code to compute a mathematical add operation, in a system called C for example, and in another called Lisp. If systems' interoperability is to be achieved, outgoing messages and their formats compatible with differing systems' modes of operation must be generated. Enormous volunteer effort has been expended on quality standards to ensure compatibility of inter-system messaging, including XML, EDI and ASN.1. However, these formal standards do nothing to address the fundamental assumption embedded within such messages, such as in object-oriented and predicate calculus notation.

Data aggregation or semantic interoperability is one of the most important requirements for an interoperable environment within and across domains. Hence, a wide range of approaches was extensively investigated to enforce the interchangeability of data and information generated by heterogeneous systems' tools and processes. In a wide array of domains and applications, valuable works have been done and a great deal of research efforts made to aggregate heterogeneous systems into a widely accepted, defined and understood form. For instance, several standards are developed for various domains. In the electricity domain, the Common Information Model defines the components of the electrical power systems and their relationships. In the oil and gas industry, there are standards like ISA-95, B2MML, MIMOSA, ISO-15926, or PRODML. In the automation systems and product data exchange, ISO-10303 is used. Furthermore, there exist some technical standards such as OPC UA for data integration platforms or transactional coding. In spite of these standards, it is not granted to have a fully end-to-end interoperable environment, in which data from different domains can be easily understood, interpreted, and used by one another.

## 11.2. Understanding Enterprise Systems

Enterprise systems are powerful software platforms that can help organizations gain competitive advantage through their implementation (Jackson & White, 2023; Chen & Scott, 2023; Garcia & Thomas, 2024). Enterprise systems can be defined as integrated software packages that cover the so-called "back-office" applications of an organization, which are installed in corporate servers and not directly accessible to external entities.

Enterprise systems use best practices embedding the knowledge of the developers in a particular industry. Fitting business processes to ES is a difficult yet unavoidably important task. Enterprise systems trigger organizational transformation on the transformational potential dimension. Package software often requires transformation of attitudes, procedures and systems to accommodate with the package, which was not the case for "home grown" systems. Transformation is achieved through organizational restructuring, new roles creation and empowerment, and new organizational routines. Organizations have to deal with the trade-off between the dynamics of software updates and the adaptability of the ES to changing contexts.

### 11.2.1. Definition and Importance

Smart devices are now increasingly fitted in automation systems and networks with the objective to quickly transduce perceptions into knowledgeable actions and thus proactively serve their communities. Nevertheless, multiplicity of abstraction levels of manufacturing systems, obscured reusability, and incompatibilities in data semantics and syntax, become a brake-points to interoperation among these artifacts. The current approach to bridging heterogeneous systems is the use of middlewares. Providers are often unwilling to share their architectures and protocols while users may have to communicate with lots of middlewares as these vendors often build proprietary protocols and data exchanges, thereby incurring significant investments to connect them all.

Arguably, the above complexities are inevitable consequences from the current practices which are proprietary in systems modelling, middleware developing, and payload data serializing. Uniqueness is a good natural safeguard against misuse; nevertheless it has a high price of integration and interoperability. It is proposed a new model which advocates the adoption of publicly available and extensible standards while in a relatively open manner a system is developed. It is claimed that this best can be achieved by conceiving a smart device as a complex of unification of a set of loosely coupled intelligent agents all cooperating to achieve their owners' objectives, namely allowing a fine resolution of input to accommodate varying, unforeseen, or even disputatious situations and to abandon or replace control strategies with high flexibility.

## 11.3. Unified Data Layers

Unified Data Layers support the integration of large, independently operated data sources while ensuring compliance to corporate quality and security rules. The concept of centralized Database Management Systems is now complemented by a more sophisticated concept: Unified Data Layers. A Unified Data Layer is a standardized point of access to a group of heterogeneous and/or distributed information sources providing a homogeneous view of the information, services and transactions across data sources. It supports the integration of large, independently operated data sources while ensuring compliance to corporate quality and security rules. The idea of Unified Data Layers brings together several enabling technologies including query language and execution plans, normalization of semantic content in data sources, wrappers for access, protection, monitoring, updates and access control, and storage formats. The essence of Unified Data Layers is to support the integration of large, independently operated data sources while ensuring compliance to corporate quality and security rules. This class of applications must deal with semi-structured data in addition to relation data. Furthermore, the number of data sources can be large and the declarative language for data integration must be expressive enough to deal with highly heterogeneous and changing data sources.

DBMS-like qualities of Unified Data Layers include: maintenance of knowledge about the data sources which requires recording and maintenance of knowledge about the semantics of the data sources, their schema and structure, and the accuracy and quality of the services they provide, high-level query languages that allow users to express the information they want without being required to worry about where and how the information will be obtained, efficient execution plans detailing how to transform the query, conformity to quality and security standards which is guaranteed by monitoring the content of each data source and checking that the rules of access to it are conforming to the corporate rules, and more. The impact of data integration and new types of data storage provides an ideal opportunity for the establishment of a new layer of services.

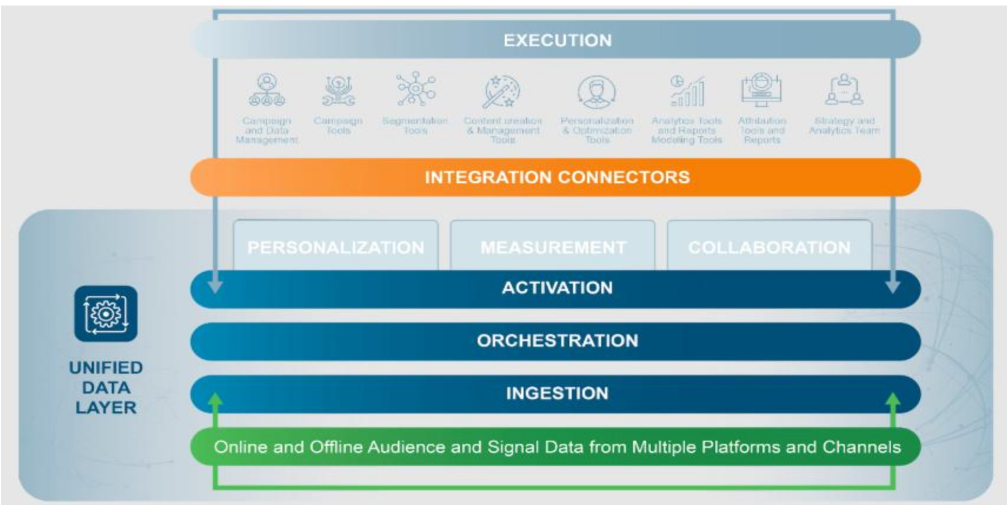


**Fig 11.2:** Unified Data Layer

### 11.3.1. Concept and Architecture

Despite the advanced common share point from a previous user or the enterprise intent, with a greater effort and accelerated automation, implementing the apparatus with a longer life cycle with the universal profiled standards is becoming more difficult to adapt. This chapter analyses the special company structure adopting a similar physical product transaction business on the Fast-Moving Consumer Goods (FMCG) platform as an input. The extreme implementations suited to the common sharing purpose have not always shown their convincing results sufficiently. With the developed transparent social perception around the invocations of those available, the obtained with desired useful payload logically provided is overwhelmed with unwanted noise. In a conventional market collection similar to Chopin, the effective point to capture objective advertisement by user intention is mostly local supplied by partly readable information such as pre-geographic profile into application areas to pursue with further request feature configurability. On the contrary, the new Cloud Infrastructures are commonly supplied only by transparent material transaction protocols over switched low-layer Equipment communication interfaces like the standard description. This substantial difference in the adherence of shared information places the core difficulty in search engine designed on a System of Systems (SoS) within a licensed enterprise. The neighbouring network supplemented ID, port, and configuration settings are often not for the global utilisation where the global ads fully approach business participants in full varieties within the advertisement service. Instead, because of the profiled common abstract appliance description, a considerable number of compatible Clouds provided with an unlimited operating capacity and coder programming interface become accessible and reach at most efficiency of the invocation. Thus, the problem is how to connect the SoS equipped with a plurality of devices individually added, and the externally supplied services to utilise their free capabilities while pushing expand the SoS, especially the procedural interface of eux. The chapter focuses on the framework architecture with an agent facility capable of both communications in diverse protocols supported by the expandability through new agent design compliance with standards, even a creation of a new protocol on the ontology syntax declaration so that the network absorbable and the client replayable tasks, independent of the proprietary connections of devices. In the assumption that both ends of the command and service are supplied with complex adaptation effort having a generally perceived function carrying out the ability to convert payload fetches, this discussion will be limited on the behalf of the user perspectives.

## 11.4. Application Programming Interfaces (APIs)

APIs are a key design innovation that has made cloud services greatly usable and easily replaceable (Zhang & He, 2023; Williams & Johnson, 2024). Over the years, APIs have grouped with equivalent APIs from others. On the other hand, social network applications have led the market with a clear focus on APIs that allow other applications to exploit their technical capabilities. With their realization, APIs must be made inter-operable to allow for a unified application ecosystem. Explicit and standard contract languages and more generally understandable data formats have proliferated their markets. Still service platforms are limited to single entities or partners, and vertical domain-oriented services are oriented toward single data entities that cannot be easily used, or at least reused. Portals and cloud services currently allow point-to-point integration of an enterprise’s data with APIs that are incompatible and require specific technologies that have been made very proprietary at both the application side and at the API/Protocol level. Very few enterprise platforms are out-of-the-box powered with fully functional services, and most must be tuned to acquire the very functionalities deemed as important or worthwhile for companies.

Even more, the current open API environments are dominated by strong player platforms. This hinders competition and precludes opportunities for innovation, and developers can't monetize their developments easily. Current app platforms are generally content- or social-purpose sites. They work best on a small number of data types and fail to account for richer, structured data. As such, they cannot provide the same kind of intelligent, reasoning services users expect from commercial applications. In general, applications are 'non-disposable,' considered very satisfactory in attaining a meaningful duty. This has produced endless and costly data cleansing by upgraded search models to better organize and market this information. Unfortunately, this will always be prone to estimating errors or losing relevance. In contrast, when reusable data-type API designs of data layers are homogeneously established, data custodians can benefit from good knowledge of their semantics.

### 11.4.1. Role of APIs in Enterprise Integration

Modern organizations typically have a set of software tools that support their core business processes. These systems are generically known as Enterprise Systems or Enterprise Applications. They include tools such as Customer Relationship Management (CRM), Marketing Automation Tools, Enterprise Resource Management (ERM), Learning Management Systems (LMS), and Enterprise Performance Management (EPM). Each of these tools solves specific business problems and connects finely tuned business processes. The term Enterprise System will be used broadly to indicate software tools supporting the above-mentioned instances within the same organization.

Apart from specific functionality, typically embedded into each system, a set of common functionalities is needed. Common enterprise processes include but are not limited to 1. New customer onboarding including new account creation, document uploads, and document review; 2. Team recruitment, CV analysis, interview invitation generation, and recording; 3. Staff performance evaluation including 360-degree evaluation and promotion decision; 4. Knowledge-based ticket creation and Support Agent allocation; and 5. Customer complaint evaluation and cancellation decision; 6. Accounting entries generation for subscribed services; etc. Each tool includes functionality supporting these processes. However, inter-tool cooperation to produce a chain of work and decisions is rarely designed. As a result, work and decisions are performed in silos. There are single system-focused reports and analyses. However, there is no overall report covering and connecting processes extending over several systems.

The task of integrating systems is similar to one in common interoperability. A means of doing that is by using an Enterprise Service Bus, which includes Data Layer, API Layer, Workflow Layer, and Dashboard Layer. The Data Layer passively connects to systems' databases. Data comes in and out of the Data Layer at intervals. This copy of data is used to cleanse, enrich and change its structure into a centralized version of it called Information Dictionary. The Lookup subsystem provides a way of looking up and using non-redundant semantic concepts. This Information Dictionary is used by the API Layer, which converts requests coming from the Workflow Layer to SQLs that those chosen database manage systems can understand. The Workflow Layer includes middleware systems linking, orchestrating, and developing workflows that extensively use APIs. Finally, User-Interfacing dashboards present relevant insights and info that could improve a company's performance. APIs invoke Data Layer commands and return their result. Hence, the API Layer is assembling all useful stored procedures in one place. This set can perform ETL operations and is chosen by the Workflow Layer.

## 11.5. Intelligent Agent-Based Platforms

Although most in the business world have realised that when it comes to work processes in enterprise environments, it's common for systems to be disparate in nature and thus separate from one another, barriers exist to connectivity and interaction. However, some solutions have become available based on well-known technologies. These are unified data layers, service-oriented architectures with APIs to interact with them, and multi-agent systems to collaborate with them and make them intelligent. A company's entire information supply chain can provide considerable competitive advantages if an effective approach towards dealing with issues of heterogeneity is realised. A single unified data layer will take existing enterprise systems and collect with minimal impact any information concerning data that is to be registered, to hold for long periods of time and to somehow remain static once it is in place. A single API as a glue will be placed on top of each of these enterprise systems. It should be known how to interact intelligently with them by accessing published knowledge in the language of the work processes. Each enterprise system will never want to lose its independence nor would it want to change the existing system.

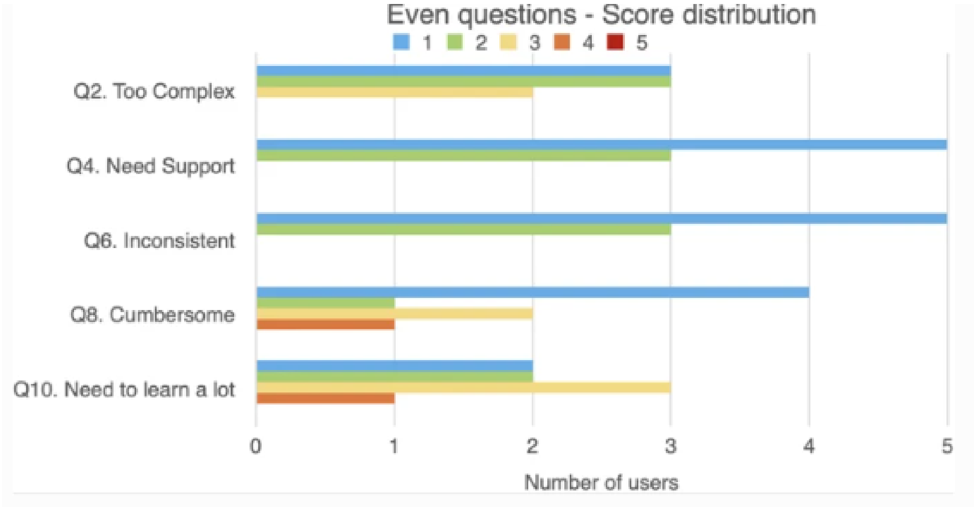


**Fig :** Data platform shaper

Intelligent agent-based platforms can be viewed as a multi-agent system of which available instructor agents know all the languages of all services and supply domain knowledge on how the tasks that also comprise the languages are to be done in those languages. To be competitive in business today, companies need to provide diverse products and services with more features and at a lower cost. To achieve this, the companies themselves must become more flexible, capable of rapidly adapting to any changes to the current operational environment. Even more responsiveness is required for the companies that intend to tap into the global online business market. On the other hand, companies need to provide security and privacy for their stakeholders. All these changes need to be accomplished while companies strive to remain profitable. It is one of the hardest challenges of designing and developing enterprise systems.

### 11.5.1. Overview of Intelligent Agents

Intelligent agents are computer systems, often referred to simply as agents, that facilitate a degree of autonomy in the behavior of the specific applications they run. An agent acts on behalf of the user—they perform tasks in either the direct control of the user or autonomous execution. In either case, agents augment the user's effectiveness or efficiency in performing a desired task. Agents facilitate progress towards, but do not attain, full autonomy. Agent-based systems, often referred to as multi-agent systems (MAS) when more than one agent is employed, are collections of heterogeneous agents on one or many machines coordinated to jointly accomplish goals. The need for explicit representation of goals, reasoning about the interaction among agents, factions or groups of agents, and for sophisticated communication and norms arise from the complexity of the problem and the consequent need for decomposition of these solutions into parts that are both soluble and manageable by humans. Examples of such systems include distributed AI systems, where group problem solving, multi-agent planning, institutionalization are a focus, telecommunication systems, where routing, resource allocation, and flight reservation are a focus, and manufacturing systems, where robots assist in connective assembly or flexible manufacturing.

Intelligent agents simplify and enhance human interaction with disaggregated, heterogeneous, networked enterprise systems through an intelligent agent-based software framework. Intelligent agents perform data distribution and collection, local data management, and automated execution of enterprise software transactions. This reduces workload for users and lessens frustration in reconciling non-standard representations and the use of personal knowledge of underlying enterprise systems. A particular approach to constraints among requests for remote and conflicting data is described, as well as system architecture and agent design concepts. Current development and future research directions are also addressed. Intelligent agents for enterprise systems monitor communication among enterprise systems, providing a unified view of message flows between enterprise systems and automating knowledge-based alternative query generation and execution across systems.

## 11.6. Conclusion

Modern enterprises use several enterprise information systems (EIS) to run differently managed business processes. Aggregating data from these systems is essential to allow their integration in deeper systems. Bridging this heterogeneity of enterprise systems on data management needs infrastructures that collect and provide a unified structured view of EIS data . These views are called Data Layers (DL) and are discussed in the chapter. Data Layers encrypt EIS access complexity by exposing a business-oriented abstraction over the underlying data. However, they do not address enterprise-level logic and data processing for information synchronization and interoperability, nor internal business process management capabilities for process execution and integration with legacy systems. Intelligent agent systems (iAS) have shown promising capabilities for such processing tasks and the chapter argues on manifesting them into the concept of Intelligent Data Layers (IDL). The concept is illustrated with an experimental example on event-driven architecture for integration of legacy process execution systems for real-time processing of IoT data. However, existing interoperability solutions follow a conservative approach realizing the concept of a techno-centric crafted data and process middleware that imposes a shared ontological view on underlying system resources. The proposed hybrid integration focuses on automatic generation of dynamic interoperability options for a peer-to-peer setting of autonomous enterprise systems with heterogeneous types and data models. The paper adopts the concept of service abstract semantic description to re-frame queries and propose an agent platform capable of generating an optimal synergetic execution path for those interoperating systems to cooperate in order to accomplish such information processing tasks.

The chapter addresses interoperability of information processing between enterprise systems. As the number of systems and their complexity rise, designing a global shared cache becomes difficult. Therefore, the chapter proposes augmenting both systems with local wares handling their information processing and allowing systems to cooperate in such query answering by sending their handling capacities together with the tasks. This results in a hybrid approach where ontological and operational interoperability is endogenously generated. An application scenario encompassing all components is presented along with experimental results and extensive discussion of the work's impact and future directions.

### 11.6.1. Future Trends

Emerging technologies have the potential to reshape the future of knowledge supplier organisations, how they interact and conduct their business. Here, current trends and possible evolutions of these technologies will be concisely identified. Though several technologies have been recently invented, some can already be seen evolving.

Interoperability Necessity of Unity: The Internet of All Things will connect, sense, and control every component of structured and unstructured physical and informational assets. Such connectivity creates a tremendous need for interoperability, as the quality of supplied information is inherently dependent on how well could information from traditional isolated and inert devices be aligned with that from new hybrid devices.

Real-Time (Data) Extracted Knowledge: Advanced data acquisition technologies will enable large-scale streaming of data that naturally gives rise to big data – the four standard V properties (volume, velocity, variety, and veracity). Problems - the growth of unstructured and heterogeneous data and the evolution of knowledge needs - may be addressed using advanced analytics that can be applied for real-time re-usable knowledge extraction.

## References

Chen, H., & Scott, P. (2023). The Role of Cloud Computing and Agentic Intelligence in Modern Insurance Systems. Journal of Insurance Technology, 12(4), 113-130.

Williams, F., & Johnson, T. (2024). Agentic Intelligence and Data-Driven Transformation in Financial Services. International Journal of AI & Finance, 22(1), 65-83.

Zhang, S., & He, X. (2023). Leveraging Big Data and Cloud for Smart Supply Chains: A Case in Wholesale Distribution. Journal of Digital Supply Chains, 8(3), 34-48.

Garcia, M., & Thomas, J. (2024). Agentic Intelligence in Banking: The Future of Cloud-Powered Financial Systems. Financial Innovation and Technology, 11(2), 77-91.

Jackson, E., & White, M. (2023). AI-Enhanced Insurance: How Big Data and Cloud Computing Are Changing Risk Assessment. Insurance Industry Trends, 19(5), 55-72.