

# Tool Integrations for Monitoring Solutions and Associated Performance Analysis

**Alexandru COCIORVA\***

*Bucharest University of Economic Studies, Bucharest, Romania*

*\*Corresponding author, cociorvaalexandru@yahoo.co.uk*

**Nicoleta ONOFREI**

*Bucharest University of Economic Studies, Bucharest, Romania*

*onofreinicoleta.94@gmail.com*

**Alexandru-Lucian VÎLCEA**

*Bucharest University of Economic Studies, Bucharest, Romania*

*lucian.vilcea@ie.ase.ro*

**Abstract.** *Today's Application Performance Monitoring evolution is in a direct relationship with fast growing e-commerce demands and associated technology prerequisites. To fulfill the high-demanding range of market expectations, tools need to be more efficient and robust, considering newly developed algorithms that can perform optimal calculations in a correct and fast way. The technological upgrade that the e-business world is facing is very often translated into new possibilities of supervising the resources for planning accordingly and setting new thresholds in various e-commerce sectors. This means that Application Performance Monitoring must be more concise, more results oriented and more determined in managing big sets of data along with high transaction ranges. This optic is not only related to administrative purposes, but it also means a paradigm change in the world of resource allocation and digitization, in the context of a fast-changing information society. The newly proposed Application Performance Monitoring approach includes tools that can handle multiple integrations, Application Programming Interface calls with external parties and a wide range of transaction mechanisms that can be implemented in a fast and efficient way. Hence, this article proposes different models for monitoring mechanisms/ tools that can integrate functions, procedures and even applications with the purpose of highlighting efficient resource allocation structures and their potential beneficial role in e-commerce optimization sector. Various integration schemas are proposed to be taken into consideration when developing an Application Performance Monitoring tool for solving complex monitoring requests, along with some web performance analysis because of associated mechanisms implementation. As a result, this article proposes different Application Performance Monitoring integration mechanisms for handling e-commerce and in general e-business solutions high demanding and complex requests.*

**Keywords:** APM, API, monitoring, integrations, Big Data, IoT, Cloud, automations.

## Introduction

E-business world is highly demanding in an increasing and vast area of digitization, in the context of an already established information society. To keep track on all the technological changes and adapt to e-commerce prerequisites, involving multiple parallel transactions and a high volume of data, tools must be enhanced with robust and optimal algorithms that can perform efficient calculations. This aspect is also translated to APM (Application Performance Monitoring) tools, since application monitoring is, nowadays, an important part of e-business enhancement, in terms of efficiently handling resources. As a paradigm of today's technological context, resource

optimization, automations and adaptive mechanisms are no longer an optional item for progress but a must have to keep up with the changing e-commerce environment. For evaluating this kind of mechanisms and efficiently supervising resources to keep e-business tools up and running, integrated application performance monitoring mechanisms are necessary. These integrations can be made with various add-ons, third party applications and API (Application Programming Interface) calls, but for this system to be available and functional is necessary to have a primary robust monitoring tool structure that can keep functional the rest of the integrated items. Integrations can be done with third party monitoring applications as well, like Zabbix, Nagios, Cacti, True Sight, while an analysis must be done for the primary monitoring applications for integration feasibility, considering the e-business context and technological demands.

What this paper proposes, is a range of models for which monitoring mechanisms can be implemented along with their solutions, while their actual customization and integration can be done based on various business cases. Hence, the paper will contribute to analyzing the range of possibilities for monitoring mechanisms implementation, for complex e-business solutions, in the context of Big Data, IoT (Internet of Things), 5G evolution and a rapidly increasing of digitization as a strong prerequisite of information society.

The proposed methodology includes two experimental methods, one through direct experience and one through observance, both of which rely on dedicated UAT and production virtual machines. A main difference between the two is related to dedicated application usage, in the second case (experience through observance) and involvement of already created application structure. Both methods can use performance tracking scripts for highlighting aspects related to infrastructure and business transaction logic.

Going through literature review as a synopsis and afterwards the development of descriptive elements for the proposed models, this paper concentrates on relevant monitoring aspects in the world of e-business, through complex mechanism descriptions and a suggestive integrations ecosystem. Hence, this paper desires to introduce elements of novelty regarding monitoring mechanism implementations for e-business solutions, fact which is also underlined in the conclusions section.

## **Literature review**

### ***Technological evolution in monitoring mechanisms & related tools***

As a starting point in the monitoring field (APM) early developments show that basic command lines were created for PC (Personal Computer) resource monitoring/ supervision purposes, in the 1980s. This framework expanded later, in the 1990s when more complex scripts (like Crontab for Linux) were created to serve the same purpose as highlighted earlier. The transition from script/ command line-based monitoring to actual tool monitoring came later in the second part of the '90s when Task Manager was introduced as a resource monitoring and administration functionality, in Windows NT 4.0 operating system. Since then, this functionality became more and more complex, offering the possibility to view resource allocation per CPU (Central Processing Unit Performance) and highlight other useful sections like: App History, Startup, Users, Detail (showing each program executable with its PID (Process Identifier) and Status) and Services status, every function with its dedicated group of administration and monitoring operations. This was going to represent just a start in the long and complex journey which will later become an extensive field, the APM. Starting with year 2000, a lot of dedicated tools were created in the purpose of serving monitoring/ supervision purpose, not only for PC related resources but also for different application function aspects, with time, extending the possibility to capture resource monitoring for an entire network

of computers/ servers and applications. This moment was going to represent the main prerequisite for what we nowadays call APM in its complex and vast defined area, including ETL (Extract, Transform, Load) mechanisms, API integrations, Cloud resource monitoring and administration, AI/ML (Artificial Intelligence/Machine Learning) implemented algorithms, going much beyond a regular monitoring tool and what it would have represented in its initial concept.

Latest types of monitoring tools include API integrations for serving different functionalities of monitoring for a better administration of the monitored e-business application, for example an e-business application that has incorporated an SAP module, for which a third-party application or system would need to be contacted. Hence through API calls/integrations various monitoring functionalities can be made available and even if there's no external module incorporated into the main e-business application, the main monitoring tool could call other monitoring functions and even tools for bringing new supervision and administration aspects if it's the case of a complex e-business application. API calls and third-party integrations could be made available not only for monitoring purposes but also for transaction/ transfer processes as part of a response-based monitoring implemented mechanism, including proactive and reactive responses.

Hence, a complex monitoring tool could incorporate dedicated ETL functions for proactive and reactive corrections applied to the main monitored e-business application. In this way, the monitoring mechanism not only becomes a supervision tool but could also become a correction base framework, for small to medium issues, especially for critical e-business applications. The issue correction threshold depends on the complexity and criticality of the e-business application/ system and on the technological barriers which are often represented by version incompatibility, servers, API architectures, security gateways and other protocols. Hence communication and security protocols are very important in defining the relationship between monitoring mechanisms, third party applications, API integrations and the actual monitored e-business application. It is very important to understand all relationships that are formed within the mentioned system to know how to apply a complex monitoring mechanism correctly and efficiently.

Integrations are widely used in monitoring mechanism frameworks, hence ITOps (IT Operations) tools need support from monitoring solutions for enabling the delivery of quality IT service. Incident response tools (event management/ notifications) can be integrated with ITSM platforms, event management or data analytic tools as proved by Centreon platform, to take as one relevant example (Chauvel, 2022). Studies on the topic of monitoring systems show that main focus was on standalone applications, avoiding the complex ecosystem integration, in terms of research and development (Oborski, 2014). Hence our proposed models rely on integrating multiple frameworks with complex monitoring solutions for creating significant value in the e-business solutions world. This framework which involves complex architectures, on both monitoring mechanism and monitored applications sides, involve large amounts of processed data, hence Big Data concept is introduced and numerous techniques for handling this aspect are necessary to be investigated once integrations are done (Bamberger, 2016).

Nowadays, more complex integrations involving functions from Microsoft APIs (including 365 cloud platform tools calls) for different e-business use cases along with complex resource supervision and administration features constitute the framework of complex monitoring mechanisms and tools. Latest developments show that complex monitoring features and mechanisms are mostly applied on IoT, 5G and cloud-related e-business applications, with a continuous growth in Big Data, Cloud platforms and AI/ML areas. One of the biggest advantages of cloud computing is related to cost reduction by freeing up some of the IT (Information

Technology) administrative time (Gupta et al., 2013). This aspect creates the premises for concentrating on complex e-business tasks and their associated monitoring operations.

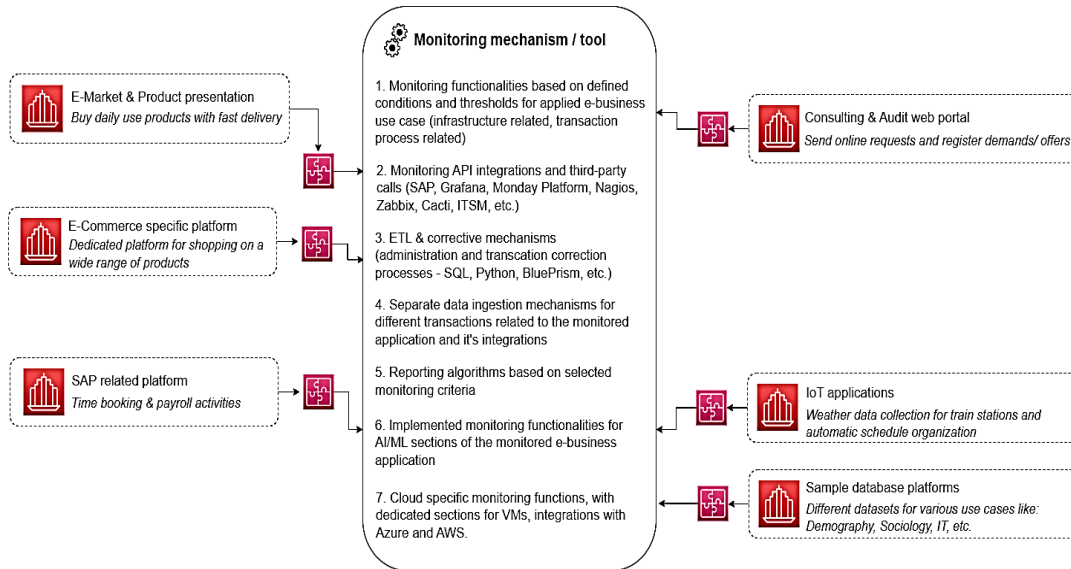
With the advancements in artificial intelligence and machine learning, IT operations teams were able to use such technologies in day-to-day work. The emergence of AIOps (Artificial Intelligence for IT Operations) monitoring system allowed the analysis of a various set of parameters in a more efficient way, by allowing the system to scan and detect infrastructure issues, such as an overload of a certain component (RAM – Random Access Memory, CPU, storage), as well as application issues, such as high exception frequencies, abnormal number of requests or lack of responsiveness. These systems usually provide ways to easily set complex alert policies, create and manage incidents and stop potential issues before they have the chance to become catastrophic.

Since the IoT available literature is very difficult to find, also related econometric research, experimental approach in finding and justifying some real-world use cases remain an optimal decision, fact which will be highlighted in this paper in the methodology section (Sarbu, 2022). Nowadays there is an increasing demand for research on the strategic use of AI to obtain competitive advantages (Borges et al., 2021). Hence, this framework could be enhanced through AI related monitoring and correction mechanisms, offering a substantial increase in competitiveness for different business cases. AI and cloud development bring also technical and business challenges in many organizations, hence for addressing different requirements from Industry and Academia, a structured framework would be necessary to provide for different business needs, also recommendations for best practices which can be adapted in different use cases (Chang et al., 2013). This aspect would also help in creating suitable contexts for enhanced monitoring solutions implementation, in a more efficient way.

### ***Descriptive elements related to the creation and implementation of the models***

It is of high importance to start the analysis of the proposed monitoring mechanism models by highlighting the e-business contexts in relationship with technological aspects, such as latest technology frameworks and boundaries related to them when it comes to the actual implementation in real life scenarios.

A suggestive diagram, that could illustrate the multitude of use cases for complex monitoring mechanisms/ tools and their integrations is described in Figure 1.



**Figure 1. Integration between different systems and the monitoring mechanisms**

Source: Authors' own research results/contribution.

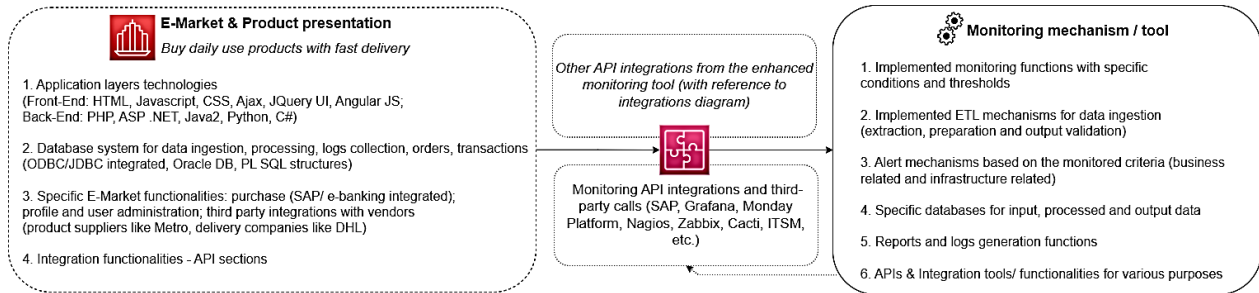
The diagram not only highlights the multitude of business use-cases from the real world but also presents the complexity of several options and scenarios in which monitoring mechanisms could be implemented for optimizing resources. It is important to mention that above presented diagram shows relationships between different types of platforms and their most appropriate type of integration, considering individual research and experience that we collected from e-business area, hence the other integrations within the monitoring mechanism could be very useful for each monitored platform.

Taking into consideration the degree of applicability of the monitoring mechanisms it's important to elaborate on each use case, as follows:

1. *E-Market & Product presentation platforms (daily use products with fast delivery)* – they represent important e-business applications in which processes are regarded as valuable resources, hence financial transactions need a complex and robust monitoring system which can integrate APIs like: SAP transactions system, Grafana, Nagios, Zabbix, Cacti etc. These integrations have an added value in capturing complex event logs for each process, not only business transactions but also infrastructure related processes (server connectivity, URL – Uniform Resource Locator availability, RAM, CPU etc.). Other integrations like Cloud services, ETL, automation and correction mechanisms could be applied in this use case, but the most important aspects, after our research regarding e-markets platform monitoring, are related to integrations with complex monitoring tools, SAP (System Analysis Program Development) system, banking systems and ITSM (IT Service Management) platform.

Automatic correction mechanisms in enhanced monitoring tools are more likely to be applied to small e-business applications, because they have reduced risk in causing transaction failures or interoperability issues. While this hypothesis is probed on the presented scenario, it is important to mention that the aim is to assure this kind of monitoring and correction mechanisms to complex e-business solutions, in the future. Another important aspect would be the integration of a web service for knowledge flow, as an integrative system that supports e-marketplaces, suggesting an integrated process that could enhance even monitoring operations (Iyer et al., 2003). However, this integration

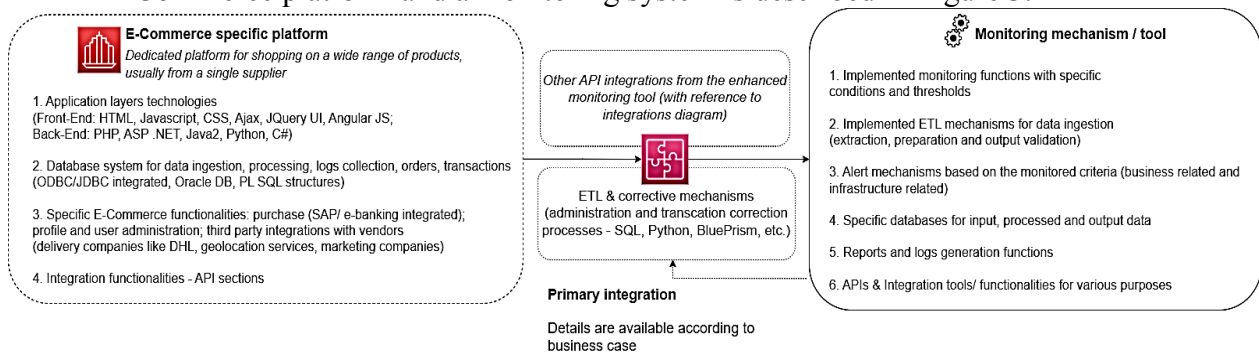
should not be a primary entity in our opinion, because it could be assimilated by a specific knowledge module that can be integrated through designed integration functionalities (API sections). A possible integration model between monitoring tools and such systems is described in Figure 2.



**Figure 2. Integration of E-Market Product Presentation Platform with the monitoring system**

Source: Authors' own research results/contribution.

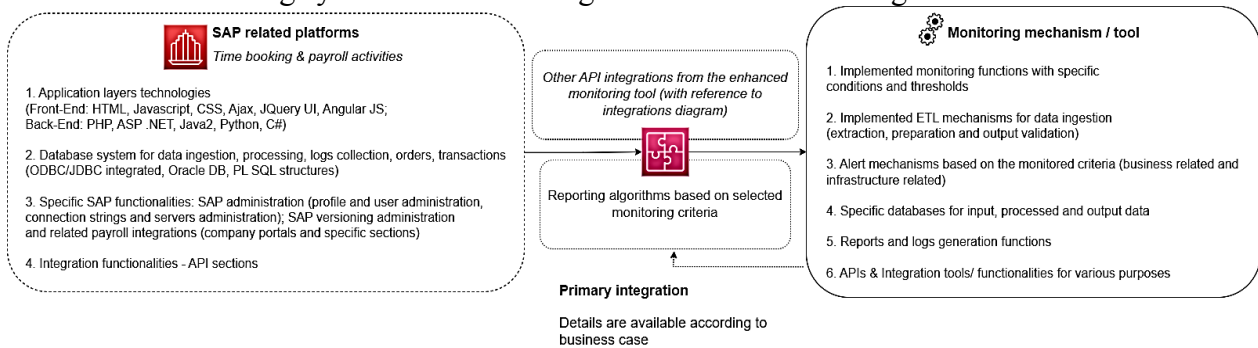
2. *E-Commerce specific platform (dedicated platform for shopping on a wide range of products from a single seller/ vendor)* – In this case the range of transactions and associated processes is smaller, meaning that enhanced monitoring mechanisms are easier to implement as well as integrations that could server different purposes like: applied transaction corrections through automations, alert mechanisms that could generate tasks in a dedicated platform, ETL mechanisms for different use cases related to the e-commerce platform (generating reports for an external party, communicating quantities and costs to an interconnected organization etc.). Although integrations are not easy to apply, taking into consideration barriers such as technology versions incompatibility or lack of appropriate infrastructure resources and even law regulation barriers, most e-commerce platforms already have or desire to have implemented an enhanced monitoring mechanism, since resource allocations becomes a more difficult problem day by day. Hence, these e-commerce entities would naturally desire enhanced monitoring mechanisms that could integrate a wide range of other applications and functions for serving mainly functionality and administration purposes. In this situation, automatic correction mechanisms as part of ETLs, could have an important positive impact such as: near live transactions correction with a high rate of success, profiles administration based on some criteria which can be defined on the go, automated alerting mechanisms directly to suppliers and all involved stakeholders, based on the nature of the registered issue etc. A proposal for integrating an E-Commerce platform and a monitoring system is described in Figure 3.



**Figure 3. Integration of E-Commerce Platform with the monitoring system**

Source: Authors' own research results/contribution.

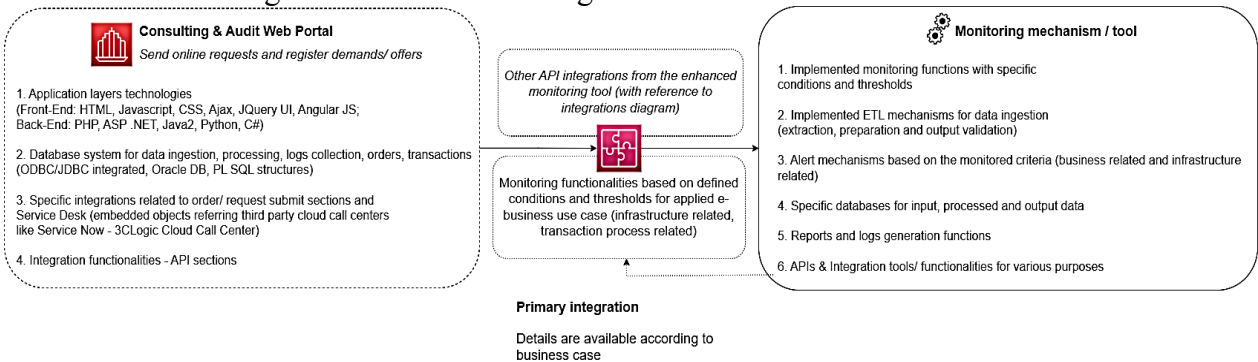
3. *SAP related platforms (time booking and payroll activities/ operations)* – For this kind of platforms it is very important, not only to monitor existent resources but also to correlate with reporting integrations linked to different parties or vendors, hence input data can be validated by the enhanced monitoring mechanism to be parsed to other systems associated to different organizations. The validation operation must be done before sending data to third parties for which the e-business case applies. All the other presented integrations are important (like Cloud services, Nagios, Cacti, Zabbix monitoring tools) but we consider that, after resource monitoring, reporting is also very important, being part of the monitoring process, since some of the generated reports could be used as activity and transaction logs for correcting problematic functionalities of the platform. The correction operations can be done through an enhanced ETL and automatic process correction mechanism, which could be an integration for the enhanced monitoring tool. SAP platforms and monitoring systems could be integrated as described in Figure 4.



**Figure 4. Integration of SAP Related Platforms with the monitoring system**

Source: Authors' own research results/contribution.

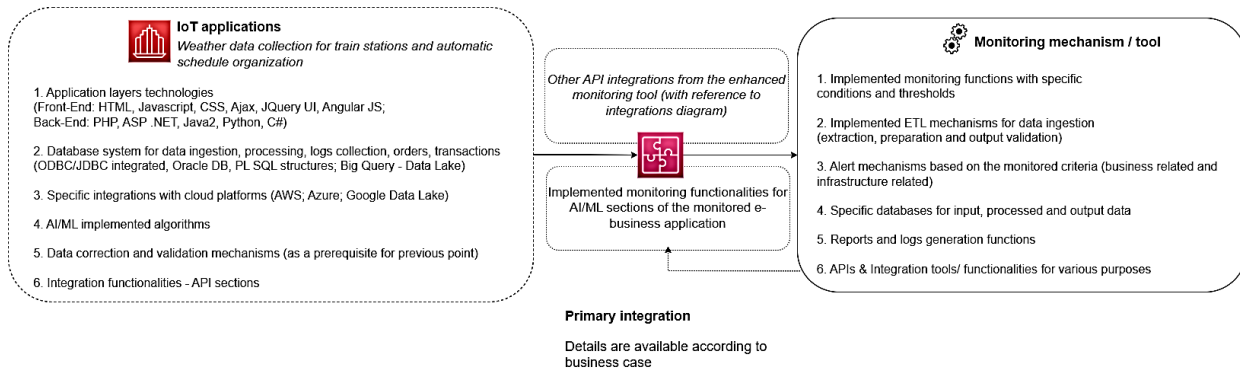
4. *Consulting & Audit Web Portal (Send online requests and register demands/ offers)* – For this kind of platforms, monitoring is represented by functionalities based on defined conditions and thresholds for applied e-business use cases (infrastructure related, transaction process related), although other integrations could be very useful for this scenario. In this case, complexity doesn't necessarily represent a characteristic for an associated monitoring mechanism, hence there wouldn't be an imperative need for including many integrations, as vital could be basic infrastructure and business-related processes supervision along with reporting functionalities, option that could offer important synthetic highlights to stakeholders and even clients. Web portals and monitoring tools could be integrated as described in Figure 5.



**Figure 5. Integration of Web Portals with the monitoring system**

Source: Authors' own research results/contribution.

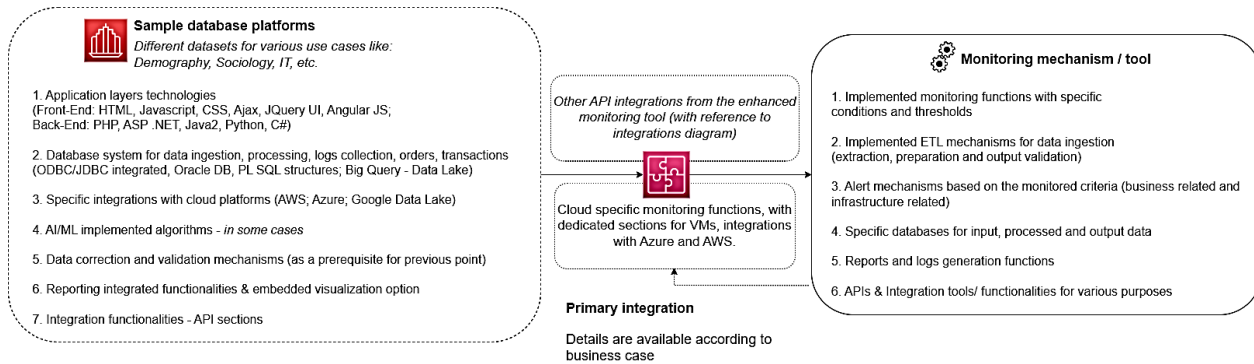
5. *IoT Applications/ Platforms (as example, weather data collection for train stations and automatic schedule organization based on provided input data) – For this business case IoT specific monitoring functionalities are required along with AI/ML algorithm supervision and automatic valid data detection for training datasets. This aspect is feasible for complex platforms connected with 5G, Big Data and cloud environments, hence VM (Virtual Machine) infrastructure and Big Query Google platform monitoring would be required. There could be numerous use cases for this scenario because the area is vast and constantly growing in a fast-changing e-business world. We could say that this scenario is a point of interest and development for latest technology implementations, being in a constant R&D (Research & Development) status. IoT applications can be monitored as described in Figure 6.*



**Figure 6. Integration of IoT Applications and Platforms with the monitoring system**

Source: Authors' own research results/contribution.

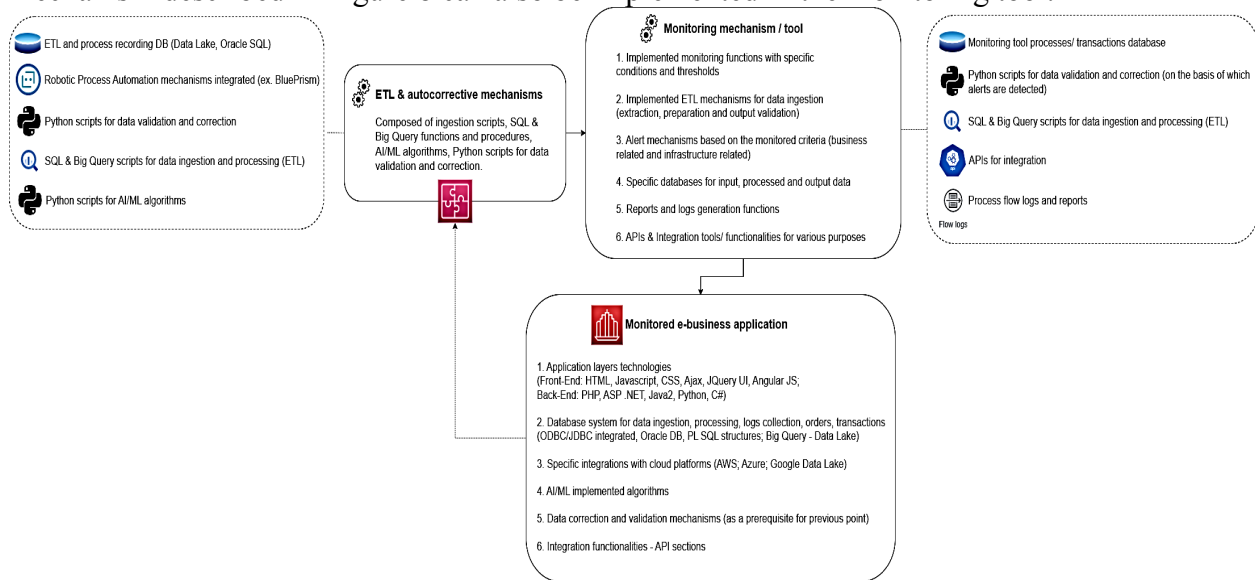
6. *Sample database platforms (different data sets for various use cases like demography, sociology, IT etc.) – This type of platform requires special monitoring with accent on the database side, hence cloud specific monitoring functions with dedicated sections for VMs, integrations with Azure and AWS (Amazon Web Services) would be included as integrations in the primary monitoring tool/ mechanism. Like in the above presented cases, all the other integrations could be included but this aspect would differ based on the business case and context. Since large data samples require Big Data and big storage capabilities, cloud environments are necessary, and they are offering the basis for sample data platforms. A database platform can be monitored as described in Figure 7.*



**Figure 7. Integration of Database Platforms with the monitoring system**

Source: Authors' own research results/contribution.

It is important to mention that all the presented business use cases and platforms can be interconnected, therefore all integrations which were presented in the first diagram could interrelate and create an operational framework for the enhanced monitoring mechanism/ tool. As an addition to the existent monitoring, lots of other technologies and platforms for various functionalities can be added as integrations for serving the business context and creating an optimal framework for resources supervision, such as OpenStack, Apache HTTP Server, Next Cloud, AWS related (AWS CloudWatch, AWS EC2, AWS RDS, AWS S3), Cloud Foundry, Ignite, Alerta, AppDynamics etc., depending on the business context and monitored e-business application use case. Since we mentioned that ETL and autocorrective mechanisms are significantly important as integrations in the enhanced monitoring process, for all the use cases (not only the ones chosen as examples), the mechanism described in Figure 8 can also be implemented in the monitoring tool.



**Figure 8. Autocorrective mechanism for monitoring tools**

Source: Authors' own research results/contribution.

Based on above presented mechanism, ETL and autocorrection integrated can be implemented considering business requirements for the monitored application, technological barriers and capabilities, other aspects related to e-business context and distributed applications interconnection.

Also, it is important to mention that above presented mechanism could be applied, for a start, not only in the case of a small or medium enterprise, but also for a big enterprise since small or medium sized process correction scenarios are also applicable here. It is more likely to apply this kind of monitoring and correction mechanism to small sized business use cases to significantly reduce the risk of process interruption or even major business impact. It could be useful, as a future research point, to study the implications of such a corrective mechanism in CRM (Customer Relationship Management) systems, using AI in decision making and not only (Saura et al., 2021).

## Methodology

### *Experimental method used for testing the model*

To test the enhanced monitoring solution with ETL and automated corrective implemented mechanism the following activities have been performed and can be validated in simulation

environments, through the experimental method: different UAT (User Acceptance Testing) environments use case testing through multiple scripts; monitoring tools relationship with different applications using different orchestration workflows.

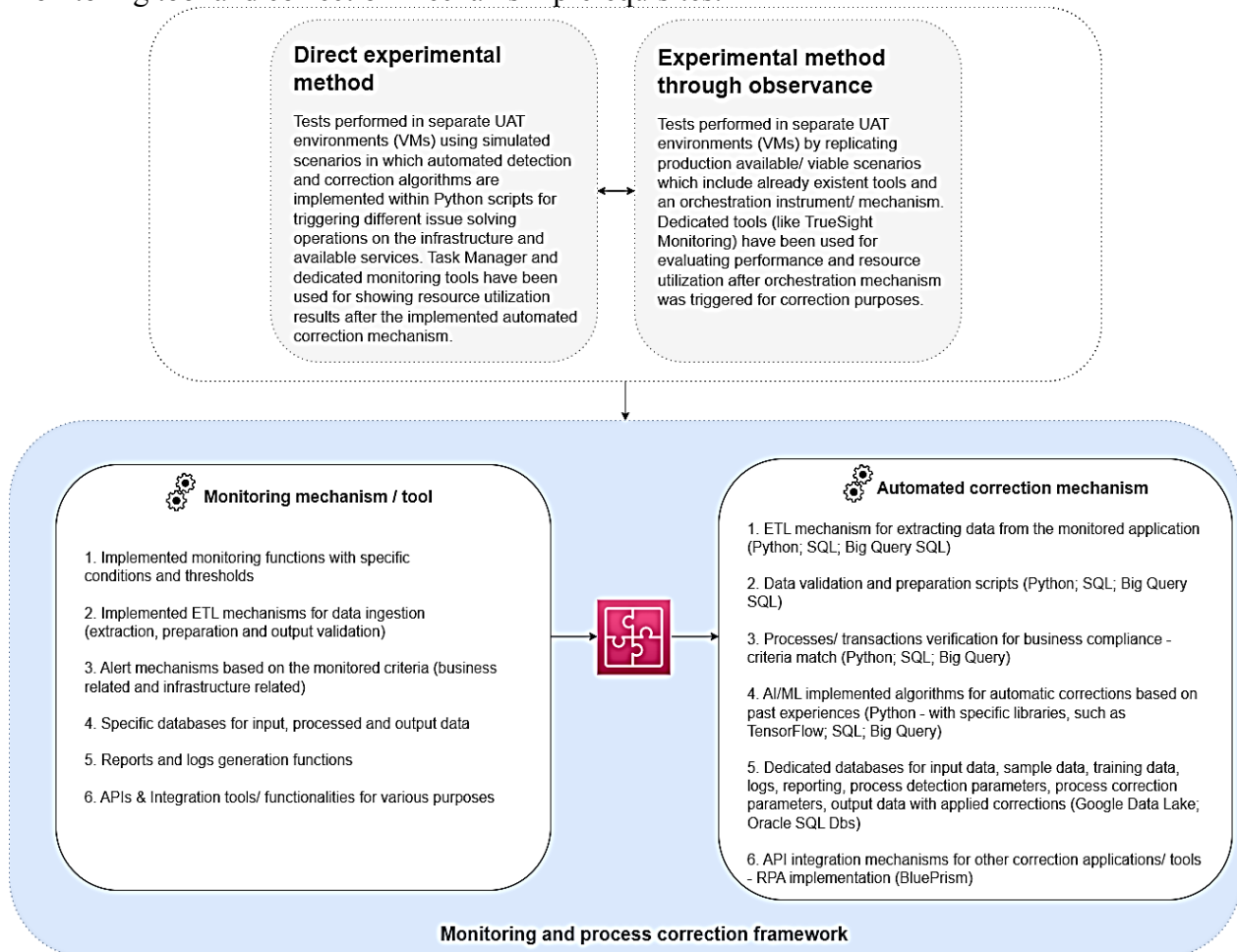
Since the multitude of e-business use cases is constantly growing it is difficult to make exact evaluations on the implementation of complex systems, application related, or monitoring tool related. In this situation, the main experimental activities have been performed in UAT environments for small business use cases as well as observing production related aspects in which monitoring tools, integrations and relevant applications are involved.

In the case of UAT environment testing the first relevant scenario was to develop and implement some Python scripts that detect when a Windows or an application service is down and automatically restart it, invoking associated service bat scripts from services.msc. This is only a test prerequisite for what automatic correction mechanisms represent in their extensive form. The next step was to enhance the existent Python scripts with more complex structures/ algorithms to perform more difficult operations like modifying a record from a database where specific business conditions were not met, perform servers restart, generate diagnose reports based on error logs and even correct application parameters in case some functionalities were broken. Since an enhanced monitoring system would involve complex structures, some monitoring tools like Zabbix and Nagios were introduced in the testing phase to communicate with the Python scripts for the monitoring and automatic correction to take place on another level. Hence, from the activity logs of the used tools we could have observed how the Python correction script is working and what are the efficiency parameters. Based on these parameters, script additions can be performed, also enhancing the existent Python scripts until CPU resource utilization is not optimal anymore. Testing each kind of integration in this manner provided us a holistic image on how integrations work for different monitoring tools, on different infrastructures, since we have chosen as UAT environments VMs with different configurations. This would offer a relevant image to the customer in terms of costs, effort, and technological barriers, related to his business use case. To verify UAT environments stability, Cloud VM administration functions from Microsoft Azure were used for Windows related tests, while other use cases were tested on Linux RedHat operating system, using Crontab scripts. The results showed a performance increase in Linux test environments, compared to Windows OS (Operating System), but from monitoring engineers' perspective Windows would have a significant relevancy increase, since UI (User Interface) operating systems are easier to operate and debug. Hence, on OS utilization type we could have seen that, from experience, the opinions are divided in an approximate range of 50% for Windows and 50% for Linux, in terms of monitoring.

Taking into consideration experimental observance results, we could conclude that, from a relevant example enhanced monitoring mechanism with ETL and autocorrection integration work efficiently for small sized use cases. In the example BMC TrueSight Monitoring was chosen for sharing statistics and reports on ITSM and MySupport & Services platforms. Since ITSM & MySupport & Services platforms can be integrated with TrueSight Orchestration and share monitoring reports on both TrueSight Monitoring and TrueSight Orchestration (as individual application reports) we chose to develop and implement an orchestration workflow that could automatically restart some ITSM related services and observe the generated monitoring logs in both applications. The results were approximately the same and increased efficiency was registered. After modifying the TrueSight Orchestration workflow to make additional operations like resource optimization and profile administration tasks, the monitoring results were still efficient but registered a significant increase in CPU. Hence, the first drafted conclusion indicated

that correction flow optimization is necessary in most cases when VM or the specific server infrastructure is not enhanced due to costs or to choose between infrastructure enhancement and correction flow optimization. In many situations correction flow optimization is chosen because infrastructure enhancement (adding RAM & CPU) is costly, and the customer would likely end in the situation when even after infrastructure enhancement flow optimization is still necessary, due to data utilization increase. In this case, business stakeholders chose to optimize the flows from the beginning, considering scenarios when increased data sets will overload the applications. This is the reason why, in many cases, Big Query, Data Lake, Cloud VMs are used from the beginning, using complex optimized Python scripts that have the role of efficiently handling application and infrastructure resources. One of the biggest challenges, now, from our point of view, is to create optimal scripts for correction and automation, that can make the relationship between business processes and infrastructure processes by automatically adding more resources on a VM when, for example, a database causes server performance issues or application stability issues. In this situation dynamic scripts can be used and even scripts with implemented algorithms that can learn from experience and adapt timely to changing conditions and context. This is the actual challenge of AI/ML landscape and the method in which adaptive mechanisms can be implemented.

The following diagram highlights used experimental methods in conjunction with monitoring tool and correction mechanism prerequisites:



**Figure 9. Experimental methods in conjunction with monitoring tool and correction mechanisms**

Source: Authors' own research results/contribution.

It is important to specify that direct experimental method along with experimental method through observance are interconnected since in the testing process, both methods benefit from each other's prerequisites; in most cases, production tool output data cannot be verified without developing specific use case scripts and UAT environment tests cannot be accurate without dedicated monitoring tools.

Another important step in performing necessary tests is related to specific process tracking scripts creation (mainly VB – Visual Basic and Python scripts) and implementation, fact which allows the possibility to analyze better how the monitoring mechanism/ application is behaving in terms of performance parameters and functionalities validation. The used scripts could track the following aspects: used services availability, integration functionality through API calls, CPU usage, links availability, ETL parameters, input and output data validity, database performance, etc. Some other basic verifications can be performed during UAT, such as database connectivity and transaction logs monitoring, URL availability in different testing stages, for various web operations as well as data and visualization refresh rates.

Tool integrations, especially when we're talking about monitoring mechanisms, imperatively involve a high security layer, which can protect sensitive data, along with business transaction algorithms. Hence the security tests performed here are related to the inclusion of Two Factor authentication and latest SSL, TLS security protocols implementation in the communication layers between the actual monitoring mechanism and integrations, using also enhanced API security Gateways.

## Results and discussions

In terms of benefits, the following points can be highlighted for implementing the described models:

1. They offer synthetic views for viable e-business cases, helping the implementation for various integrations within monitoring mechanisms, especially in complex business and technical scenarios.
2. A contribution to the monitoring process enhancement is observed, especially through ETL and automatic correction mechanisms, also helping the integration of other APIs, related to the main purpose of the monitored application. It also offers an overall image on the types of integrations that are most appropriate for specific business and e-business use cases.
3. Experimental results show that automatic correction mechanisms could be used as an enhanced monitoring tool integration in all business cases, but success rate is more likely to be grown in small and medium sized business cases, because of risks that could be involved in the processing part. Hence complex business context along with vast associated platforms could face multiple risks in implementing corrective mechanisms, although one of the aims is to introduce this kind of mechanisms in the future. Complex business/ e-business cases are the ones that need mostly complex corrective automated mechanisms for optimizing business processes and other transactions due to high volume of data, which is to be registered, mainly due to 5G and IoT near future implementations.
4. Presented diagrams could offer a holistic image on applied integrations for enhanced monitoring solutions, providing e-business use case suggestions for start-up enterprises in a complex digital world. This fact could trigger ideas related to applied monitoring mechanisms for e-business solutions based on different scenarios and contexts.
5. Through different experimental methods, applied automated correction mechanisms and ETLs are proven effectiveness especially for integrated AI/ML applications; in this case

correction processes could be part of the monitored applications and not separate entities as integrations from the enhanced monitoring system. The idea to separate integrations and correction mechanisms from the actual monitored applications and introduce them as parts from an enhanced monitoring tool is related to cost, efficiency and to avoid complex application structures that could encounter various processing issues and face security risks in real-world scenarios.

6. The presented scenarios could involve lots of integrations and create a multitude of possibilities for enhanced monitoring mechanisms and techniques. The models presented in this paper are only a starting point in what could represent a very complex framework for monitoring mechanisms that could be applied on a large scale, for all sorts of distributed applications, in an efficient manner and even for technology areas that are currently under development.
7. As IT infrastructure becomes less relevant for Cloud consumers, they can concentrate on core their business, taking into consideration best practices when receiving public Cloud services (Nieuwenhuis et al., 2018). This aspect means that consumers will be more focused on services and integrations that could leverage the e-business framework, taking it to the next dimension in which AI/ML plays a definitive role. This is the point in which automated monitoring and corrective mechanisms could precociously step in, taking into consideration that still, lots of e-business use cases are currently adapting to this kind of technology.

The disadvantages of using proposed enhanced monitoring mechanisms with automatic correction integrations could be as follows:

1. Some of the most important disadvantages in using very complex integrated structures as part of monitoring tools are related to high cost and even some technological barriers, since some integrations could be very risky in terms of security and deployment.
2. Automated corrective mechanisms are only applicable to a certain extent, in few scenarios from the real e-business world, taking into consideration that efficiency is proven only at a high scale only for AI/ML developed applications or pilot projects. Hence, there is yet a lot to improve in this area and intensive research is currently ongoing.
3. Enhanced monitoring mechanisms could face barriers like technological incompatibility, Big Data structures which are not correctly and completely defined, alerting mechanisms that aren't consistent in terms of conditions and set thresholds etc.
4. There could be an ethical aspect related to monitoring mechanisms with automatic correction integrations, meaning that businesses may seek automation for short-term financial gain while ignoring greater macro effects, especially in the relationships between involved business parties or stakeholders (Wright and Schultz, 2018).

## Conclusions

A set of relevant conclusions can be drawn based on the presented research. Enhanced monitoring mechanisms along with automated correction mechanisms and ETL processes could leverage existent operations for any e-business application, especially in the Big Data, e-market, e-commerce, and financial fields. Complex cloud systems have the possibility to automatically update all software integrations within e-business applications framework, hence concentrating all resources on business processes (Bharadi et al., 2016). In this case, ETL and autocorrective mechanisms become even more desirable for resource optimization purposes, even as part of cloud integrated monitoring mechanisms. There are specific sets of integrations that are suitable for

different business use cases and scenarios, this paper is offering only a start in forming a complex framework that could be applied in numerous cases with different applicability. Integrated applications and monitoring mechanisms are interconnected to some extent, since in many cases monitoring functionalities are embedded directly into the main monitored structures and don't represent separate entities/ tools. Hence it is common to see applications having integrations with functionalities offered by third parties that are not usually associated with the field of activity of the application, but we tried to offer a set of guidelines through models that followed some observed patterns that could be useful for future use cases. As we live in an interconnected and interdependent world the e-world is a very good projection of business use cases met in real life, hence this aspect is met also in e-business applications, a reason why enhanced monitoring tools must contain robust structures and have relevant integrations for offering performance upgrade to monitored applications. There are numerous benefits for integrating different functionalities into e-business cloud base applications, but through this paper we concluded that, even more beneficial would be to have monitoring tools with integrated functions for different purposes, specifically ETL and automated corrective options. It is important to mention, that related to this aspect, third party integrations like secondary e-business applications across various operational metrics would require a specific level of excellence; hence it can be discerned that management must be sure that their information system can integrate process data both inside and outside the company (Sokiyna and Aqel, 2020). Also, multiple legal barriers can be encountered here, due to data privacy and protection, related to third party application integrations, especially when we include into discussion a monitoring and auto-corrective tool. It is important to highlight that automatic correction mechanisms along with monitoring solutions can be applied also in e-business use cases where RPA (Robotic Process Automation) tools with AI is involved, but for this is necessary to have well-defined, stabilized, and mature processes, like in strategic areas focused on customer tasks, increasing employee productivity, improving accuracy in categorizing, and routing processes, enhancing the analytical data analysis, etc. (Ribeiro et al., 2021) A part of this framework has been already presented in this paper, in e-commerce monitoring and associated integrations use cases.

Another aspect that could generate value in the context of integrating correction mechanisms into monitoring applications is related to IPCC (Incident, Problem, Change, Configuration) framework, hence once incidents of different types are recorded, necessary actions could be undertaken for correcting a transaction or a faulty administration of a user's section. Here our contribution refers to the creation of a specific IPCC framework which can be customized and adapted as per different business cases. Hence, automated correction mechanisms that implement actions like SQL (Structured Query Language) modifications for a transaction fix, through raised CRs (Change Requests), in relationship with PIRs (Problem Investigation Records) and incidents, could be of a high importance for the e-business solution. In this context, also AI / ML algorithms could be implemented for automatically recognizing patterns of issues and generating the proper solution that could be applied using a combination of diverse solutions applied for different business cases. In this context, it's important to mention the scalability of the monitored application, involved systems and, of course, the metrics that are taken into consideration, since a lot of use cases for the real e-business world are difficult or almost impossible to correct timely, especially through automated mechanisms. Hence, we'll take into consideration aspects that may be easily corrected for a start (transaction changes, profiles and pages administration, infrastructure related matters) for which the automated correction mechanism can apply timely system

corrections, while the difficult ones could be reported to dedicated teams for investigation through an automatic incident or PIR, raised into an ITSM platform.

As general outcome of the present research paper, there are some important differences between our study and previous works from the monitoring field, with focus on multiple integrated systems, integration capabilities, ETL and automated correction mechanisms. All the presented aspects were described taking into consideration IPCC framework, with accent on ITIL (IT Infrastructure Language) area elements.

## References

- Bamberger, M. (2016). Integrating big data into the monitoring and evaluation of development programmes. United Nations Global Pulse.
- Bharadi, V., Saswade, N., Zanzane, Y. (2016). Virtual machine monitoring in cloud computing. *7<sup>th</sup> International Conference on Communication, Computing and Virtualization 2016, Procedia Computer Science*, 79, 135-142
- Borges, F.S., A., Laurindo, J.B., F., Spinola, M.M., Goncalves, F., R., Mattos, A., C. (2021). The strategic use of artificial intelligence in the digital era: Systematic literature review and future research directions. *International Journal of Information Management*, 57, 102225.
- Chang, V., Walters, J., R., Wills, G. (2013). The development that leads to the Cloud Computing Business Framework. *International Journal of Information Management*, 33(3), 524-538.
- Chauvel, R. (2022). *Should we integrate IT monitoring with IT Operations tools?* Centreon. <https://www.centreon.com/should-we-integrate-it-monitoring-with-it-operations-tools/>
- Gupta, P., Seetharaman, A., Raj, J.R. (2013). The usage and adoption of cloud computing by small and medium business. *International Journal of Information Management*, 33(5), 861-874.
- Iyer, L., Salam, A.F., Singh, R. (2003). Web service for Knowledge Management in E-Marketplaces. *Indiana University Press e-Service Journal*, vol. 3, no. 1, 32-52.
- Nieuwenhuis, J.M., L., Ehrenhard, L., M., Prause, L. (2018). The shift to Cloud Computing: The impact of disruptive technology on the enterprise software business ecosystem. *Technological Forecasting & Social Change*, 129, 308-313.
- Oborski, P. (2014). Developments in integration of advanced monitoring systems. *The International Journal of Advanced Manufacturing Technology*, 75, 1613-1632.
- Ribeiro, J., Lima, R., Eckhardt, T., Paiva, S. (2021). Robotic Process Automation and Artificial Intelligence in Industry 4.0 – A Literature review. *Procedia Computer Science*, 181, 51-58.
- Sarbu, M. (2022). Which Factors Determine the Adoption of the Internet of Things? Impacts and Benefits. *Jahrbücher für Nationalökonomie und Statistik*, 242(1), 107-147.
- Saura, R., S., Ribeiro-Soriano, D., Palacios-Marques, D. (2021). Setting B2B digital marketing in artificial intelligence-based CRMs: A review and directions for future research. *Industrial Marketing Management*, 98, 161-178.
- Sokiyna, M., Aqel, M. (2020). The role of e-business applications software in driving operational excellence: Impact of departments collaboration using sustainable software. *Sustainable Computing: Informatics and Systems*, 28, 100445.
- Wright, A., S., Schultz, E., A. (2018). The rising tide of artificial intelligence and business automation: Developing an ethical framework. *Business Horizons*, 61, 823-832.