Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

Microservices In Context: Internet of Things -Infrastructure And Architecture

Zaheen Ahmad
Dept of Computer Science, University of Engineering and Technology, Mardan, Pakistan
zaheen.ahmad@uetmardan.edu.pk

Muhammad Danyal
Dept of Computer Science, University of Engineering and Technology, Mardan, Pakistan
muhammad.danyal@uetmardan.edu.pk

Sadaat Ahmed (Correspondence Author)
Department of Computer Science, FATA University, F-R Kohat, Pakistan
sadaatdost@gmail.com

Sadiq Shah
Department of Computer Science, FATA University, F-R Kohat, Pakistan
sadiqshah@fu.edu.pk

Misbah Daud
Department of Computer Science, Shaheed Benazir Bhutto Women University Peshawar, Pakistan
Misbahdaud@sbbwu.edu.pk

ABSTRACT

Enterprise systems are commonly developed using monolithic software architectures. However, this approach is often not well suited for Internet of Things (IoT) environments due to the large number and diversity of connected devices. As IoT systems continue to grow, managing scalability, flexibility, and frequent updates becomes increasingly challenging. This study explores the role of microservices architecture (MSA) in the context of IoT systems. A conceptual architecture is discussed to identify the main components and relationships involved in IoT environments while considering key characteristics of microservices. The conceptual model highlights important aspects related to the integration of microservices in IoT infrastructures. Based on this understanding, a reference architecture is outlined to support the development of modular and distributed IoT applications.

Keywords: Microservices Architecture (MSA), Internet of Things (IoT), Conceptual Model (CM) and Reference Architec-tures (RA)

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

I. INTRODUCTION

The internet has revolutionized our approach to business and entertainment, offering a standardized means for people to connect from anywhere and at any time. Entering a new era of seamless mobility and expanded coverage, the Internet's rapid growth holds the potential to eliminate barriers to truly seamless interaction.

The vision of IoT extends the internet into the real world, connecting everyday objects to the virtual world - grounded on steady advances in microelectronics, communications, and information technology. Due to the fact of their diminishing size, falling price and declining energy consumption in recent years, processors and communication modules are being increasingly integrated into everyday objects [1].

Most enterprise systems are built with a monolithic way of software architecture. A unified monolithic architecture framework is not a very suitable and realistic solution due to the quantity and diversity of IoT devices connected to the internet [2]. The continuous development, frequent updates, and changes require the whole application to be redeployed, which is a large impact and high risk for updates and deployment build-up between releases [3].

The large code bases as a single deployable unit, making it cumbersome and difficult to change, test, and deploy, typically in monthly deployment cycles. Therefore, MSA addresses these issues by separating the application into multiple deployable units e.g., service components [4].

II. LITERATURE REVIEW

This literature review aims to identify and analyze existing published research and information within the research area. The objective is to explore the intersection of MSA and IoT, examining how both paradigms are utilized in conjunction.

B. Internet of Things

In the domain of ICT people are equipped with fast-speed internet access. Therefore, the interface between the physical and digital world requires the capacity and convergence of at least three domains, that are, technologies (nano-electronics, sensors, actuators, embedded systems, software, cloud com-puting), communication, and the intelligence to sense and act upon it [5]. Referring to the simplistic equation, IoT is composed of sensors, actuators and embedded controllers [6]. In the realm of IoT, devices are compact units equipped with their processor, operating system, and sensors. They incorporate logic to interact with streamlined APIs, essentially functioning as microservices [7]. The microservices architec-ture suits the design and development of the IoT application architecture. It is defined as "the minimal independent process that interacts via messaging" and "a distributed application where all its modules are microservices" [8]

C. Microservices Architecture

Exploring improved methodologies for system development, embracing emerging technologies and anticipating the trans-formative impact of advancements to create IT systems that satisfy both customers

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

and developers alike. The customers of Microservices Architecture are the organizations, developers, and architects. Microservices Architecture significantly divides enterprise IoT systems into a scalable, logical and granular set of functional domains often in a domain-driven design context [9]. Moreover, existing components can be extracted to reduce the size of monolithic applications as microservices are created smaller and easier to maintain code [10].

D. Microservices Architecture Reference Model

The systematic literature review [9], introduced a reference model characterizing based on an architecture-centric classification of microservices. It helps to demonstrate current research at a conceptual level to identify trends and research directions. However, there is currently no secondary study to consolidate the research based on the characterization of a framework. However, the use of frameworks can ease the development of individual services [11].

Design Principles

The objective of this section is to understand microservice design principles and constraints to be considered while de-veloping microservices.

- Isolation
- Share-Nothing Architecture
- Small and Focused
- Loose Coupling
- Language-Neutral
- Bounded Context
- The Need for an API Gateway

Microservice Architecture Technologies

Some of the basic technologies that lead to the microservices architecture are listed below [11].

- Domain-driven design
- Continuous delivery
- Ports and adapters pattern
- Machine-to-machine communication
- Virtualization platforms
- Choreography over orchestration
- Container
- Docker

Cross-Cutting Concerns

It is to embrace an approach to microservices adoption that cuts across the architecture, organization, culture, processes, and tools; which enables to improve their delivery speed, flexibility, autonomy, and developer morale. Hootsuite has created a goal-oriented tool set for microservices development and has empowered the iterative style which is a match for the microservices way [12].

- Decomposition Techniques

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

- Deployment of Microservices
- Service Discovery and Registry

E. IoT and Microservices Architectural Style

The Microservices Architectural Style has got hype in cloud and enterprise business applications. It should be noted that the scalability of applications makes a prerequisite for the efficient use of IoT resources [13]. The SOA is already a solution in the IoT but is still not sufficient to achieve interoperability [11].

Aspects and Contexts

Hence, both industry and academia present partial solutions. The design and development challenge is the sustainable adop-tion of microservices to distributed application/infrastructure architecture due to the novel aspects/facets of microservices [14]. The key aspects of microservice architecture and IoT are conceptually quite relevant [15]. The MSA and IoT share similar attributes.

- Componentization via Services
- Organize teams around business capabilities
- Smart endpoints and Dumb Pipes
- Decentralized Governance and Data Management
- Design for Failure
- Evolutionary Design

Gapes

Essentially, such a gap is caused by the technology-driven research [16]. Indeed, current technologies make the IoT concept feasible but do not fit well with the scalability and efficiency requirements. It has been noted that there are wide gaps between the current industry level and academia. The available studies regarding the research were limited in providing a temporal overview [17].

Challenges

The internet introduces several new challenges [18]. The wide spectrum of services, applications and IoT ecosystem poses several questions. Thus, lack of standardization still requires proper addressing might have an impact on the paradigm evolution [19]. Designing software systems with microser-vices architecture and implementation in practice might be challenging [20]. One of the most crucial problems is the interoperable data exchange in large-scale heterogeneous IoT network elements. However, using microservices simplifies the design and implementation of individual services, but increases the complexity of distributed systems.

F. Related Work

The research gives a brief overview of patterns and best practices emerging from the microservice approach for IoT – how can they be adopted in several IoT devices to create applications?

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

The researchers have described the benefits of applying MSA to build a Smart City IoT platform for a variety of relevant applications [20]. The Sensor HUB framework is a collection of different technologies to collect the sensor data, transmit, process, analyze and support the IoT-related development [21]. The research [22] investigates the design and development contexts of IT service systems and applications. Further, it claims that the systematic relationships among the contexts can be exploited as design parameters or patterns. The work [23] has presented three cloud microservices alongside mobile and IoT usage examples.

III. METHODOLOGY

To utilize microservices architecture for the Internet of Things, there is a need for a development framework, offering modelling languages for vocabulary, an architecture language for describing application functionality, and a deployment language for deployment features and implementation tech-nologies. Appropriate tool-kits and frameworks play a useful and important role for complex event processing e.g., formu-lating machine-readable rules for determining the triggered sequences of events activity or process [24].

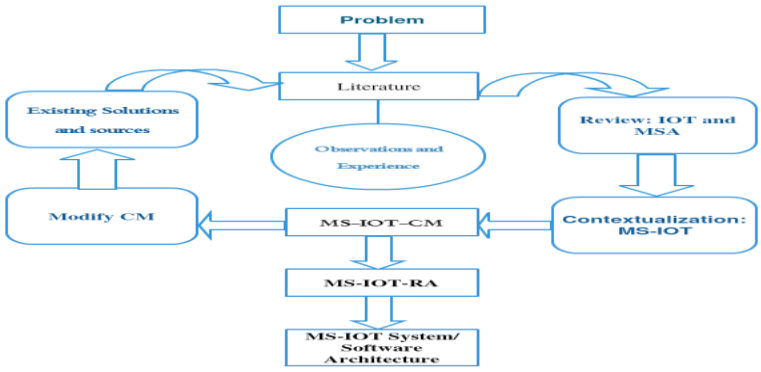

Fig. 1. The overall Research Design

A. Architecture Development Framework

The overall methodological development process is used to derive MS-IoT Reference Architecture and MS-IoT Sys-tem/Software Architecture starting from conceptualization firmly grounded on contextual analysis. The contextual anal-ysis provides a list of important IoT-related factors and MSA design constraints, patterns, technologies and crosscutting con-cerns in microservice provision on the Internet from things. Thereby, it makes it easier to see the systematic relationships among the contexts for the design and presentation of a preliminary conceptual model.

Conceptual models, also known as conceptual schema, are easier to understand than real software due to their higher level of abstraction [25]. Further, if they are to be used in a Model-Driven Development (MDD) framework to generate code automatically [26]. Thus, conceptual schemas are created during the requirements engineering stage and form the basis for the design of reference

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

architecture.

The reference architecture is selected because of the possibility of evaluating their contribution to the design of concrete architectures [27]. The reference architecture is used as a stan-dardization tool that guarantees the interoperability between systems and their components specifications of concrete Micro services IoT architecture design down to its implementation.

B. Description, Instantiating and Evaluating the Reference Architecture

In particular, the description of reference architecture and its instances is based on the combination of the IEEE 1471-2000 recommended practice [28]. The reference architecture is then instantiated to create and extend new software architectures for classes of systems [29]. The paper [30] aims to provide a concept for future IoT architecture, including a definition, a review of developments, a list of key requirements and technical design for possible implementation. The paper [31] presents an approach to describing a reference architecture and instantiating it into software architecture by making im-plementation decisions and evaluating it concerning quality attributes.

The domain analysis and software engineering practices of the requirement elicitation process [32] and RUP 4+1 [33] is a well-established software engineering process. To construct the specific set of views shall use the Unified Modeling Language, which is a de facto standard to design the diagrams of the architectural views.

The reference architecture can be prototyped with relatively cheap hardware and software to enable prototypical imple-mentation of the system and software architecture [34]. The instantiations that are needed to transform the actual system are based on the Reference Architecture utilizing a prototype for a specific use case [35]. This prototype is a showcase of how added-value microservices can be seamlessly created on top of the proposed architecture framework.

IV. EXPLOITING MSA IN THE CONTEXT OF THE IOT

The development methodological process adopted to derive MS-IoT Architecture shows how the architectural descriptions and design specifications of the MS-IoT Reference Archi-tecture are guided by its MS-IoT conceptual model firmly grounded on contextual analysis. Indeed it helps in the iden-tification of the common denominators that facilitate cross-disciplinary research work continuation of the target domains.

A. MS-IoT Architecture Development Framework

We discuss a framework for understanding software ar-chitecture via architectural styles, revealing how styles can be used to guide the architectural design of IoT systems and software application development. Micro-service-oriented development is a complicated process. The methodology can be implemented as a concrete architecture development frame-work characterized by the concept of the notion of a service component.

The framework is constructed to support the analysis of reference architectures in terms of relationships between their context, goals, and architecture design/specification. It pro-vides a set of modelling languages and techniques at different phases of microservices IoT applications.

Contextual Analysis

Some common design features of microservices suit the context of a choice for developing

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

IoT applications. Furthermore, we describe MSA in the context of software architectural style with common and relevant aspects of the IoT- analyzing the IoT and Software architectural styles/patterns.

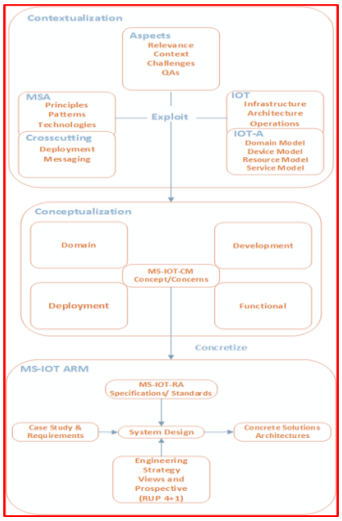

Fig. 2. MS-IoT Architecture Development Framework

Conceptualization

The research [41] focuses on domain ontologies that specify conceptualization specific to a domain. In the field of infor-mation systems viewing domains in a particular way is called the conceptual model [25].

Conceptualization aims to answer specific focus questions through vigorous literature. Thus, the relevance of the con-cepts and suitability from things as well as internet-oriented aspects of the IoT infrastructure, software architecture and microservice architectural style constraints are formulated in the focused research question.

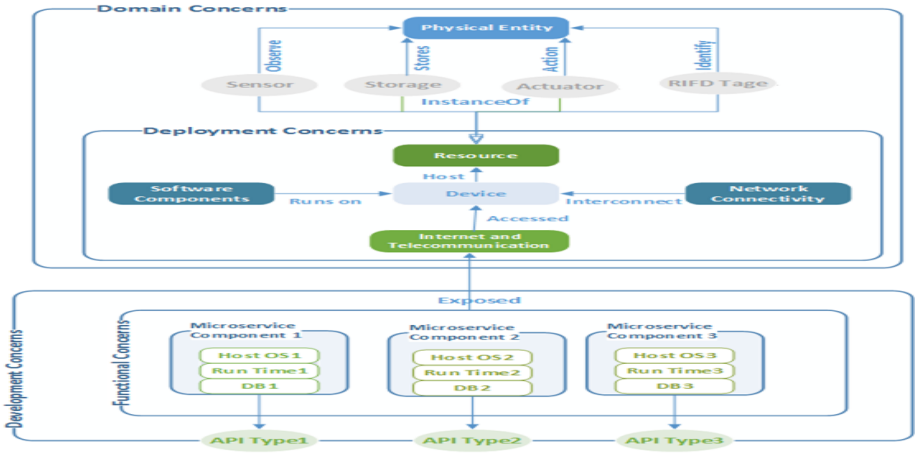

Fig. 3. High-Level MSA and IoT Conceptual Model

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

MS-IoT ARM

This is an approach towards developing the reference architecture. In previous research [36, 37], the IoT-A reference models could be interpreted as the basis for describing reference architecture. As an alternative, we used a conceptual model instead to establish terms and concepts about the content and concerns of architectural descriptions and design of the reference architecture. The MS-IoT ARM consists of interconnected parts i.e., the MS-IoT Conceptual Model (CM), MS-IoT Reference Architecture and Guidance to define the overall process that should be used when deriving a concrete MS-IoT System/Software architecture. In particular, the guidance part proposes a large set of tactics and design choices that can be associated with the perspectives, i.e., qualities of the system the designer wants to meet. These mentioned aspects influence the design of the reference architecture and specifications of MS-IoT architectures.

MS-IoT Reference Architecture

We propose an MS-IoT reference architecture (RA) shown in Figure 4, which in turn is based on deployment and development concerns and concepts introduced in a high-level conceptual model. The reference architecture is a richer and more concrete model considering the integration of hetero-geneous network-agnostic connectivity like WAN, and LAN communication models for constrained and unconstrained de-vice architectures.

Such reference architecture can be the reference for building compliant views and perspectives on different architectural aspects, which provide a common foundation on specific requirements of the concrete micro service IoT infrastructure and architecture via the cloud on IP-enabled smart object solutions. Concrete software architectures are designed based on required functionalities and architecture qualities reflected in a specific context and business goals of the system.

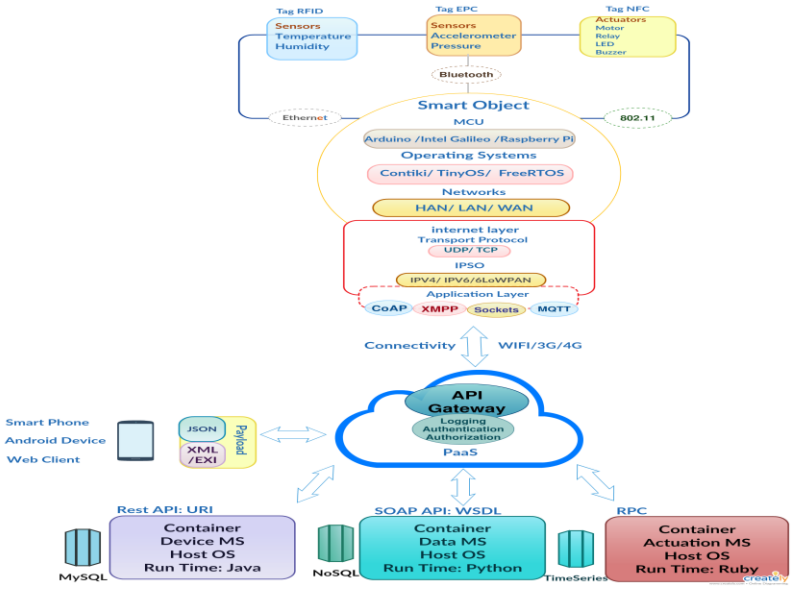

Fig. 4. MS-IoT Reference Architecture

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

V. EVALUATION

We demonstrate how all the aspects of the MS-IoT reference architecture can be instantiated for the derivation of software architecture reference implementation, according to software enterprise engineering frameworks. The MS-IoT reference architecture will be validated through instantiation for the MS IoT architecture. Software architectures can be designed and evaluated to fulfil specific criteria and quality attributes [31]. Further, Rapid Application Development (RAD) and Agile development could be useful to be utilized for implementation. A case study approach could be used to highlight the core and auxiliary principles of MSA applied to IoT for a set of real projects. The research work applies micro-service architecture to develop the evolvable system. Which can be beneficial for small to medium software projects [38]. The evaluation of usability and guidelines to derive concrete architectural solutions are open issues. The evaluation gives assurance of fulfilling quality requirements. Evaluation criteria help to select the optimal architecture evaluation method. The evaluation methods can be classified as software metrics-based, scenario-based, and attribute-based approaches. The evaluation methods are based on several parameters such as the method's goal, evaluation approach, quality attributes and stakeholder involvement, method's application, and tool support for software development. The different proposed ap-proaches are: experience-based, scenario-based, mathematical modeling, and simulation-based. Among these scenario-based approaches is quite mature to evaluate software architectures [39]. The scenario-based software architecture design and as-sessment method has its major strengths in the way evaluations are addressed [40]. IoT offers ecological systems using environmental parameters collected by a large number of sensors and automatic alarms for forests, glaciers, volcanoes, and seismic and factory mon-itoring [42]. Global warming raises the average temperature causing several significant problems on Earth. Water flooding is the most considerable problem of climate change. However, monitoring rivers via WSNs needs an event-driven flood monitoring system [43].

A. Illustrating the Case Study

Flood has become one of the major problems in most of the countries around the world [44]. Every year, it causes lives and damages to infrastructure, and agricultural production and severely affects local economic development [45]. There must be applications that offer a lot of services. The case study presented in the paper [44] uses remote sensing data in the operative process of early warning, mitigation of flood disasters and receiving notification in case of floods in the area.

Microservices Flood Forecasting System (MFFS)

From the Himalayas of Tibet, the River Indus flows crossing India through Pakistan before falling into the Arabian Sea covering 3,200km. Another important aspect is the possibility of extending the set of available services with new components that could exploit the data collected by the system making the decision based on flood forecasting and weather prediction methods and alarms [46].

Flood emergency management may offer many services in a complex scenario and large system that may rely on the weather, hydrology, remote sensing, surveying and mapping to work and bind into effective flood forecasting systems.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

Requirement Elicitation

There are a lot of requirements, but its key requirements are organized by practising the requirement elicitation process with different phases to identify the right requirements from the flood forecasting and anticipation of early alarms.

- Functional Requirements

Functional requirements often get the most focus in a devel-opment project. The flood monitoring system needs to support real-time and reliable data acquisitions for the operative pro-cess of early warning in case of flood disasters.

→ Sensing real-time hydrological and meteorological stream flows and levels.
→ Covering large geographic region information.
→ Detecting and predicting river floods.
→ Triggering timely warnings or notifications.
→ Configuring remote device management

- Stakeholders and Concerns

Users of the climate forecast include federal, state, and local agencies, as well as, public communities. The real-time information will be available to users via multiple wireless communication systems without geographical and weather constraints on the web.

→ The system stakeholders, and the concerns that they may have regarding any possible aspect of the IoT system.
→ The definition of the viewpoints that will address the stakeholders' concerns, and prescribe the methodology used and the contents of corresponding views.
→ The views are the most important or architecturally significant modelling elements.

End-users and their Roles

A user interacts with an end-user application e.g., visuals of sensed information, hearing an alert message, etc. Users access the flood data and events of stream flow and level estimates to make timely alerts and notifications via internet applications.

→ Meteorologist: Generate flood forecast information and also capture real-time stream gauge data for deciding emergencies. Hazard analysts use the output of the model to determine the extent and severity of flooding.
→ Government Agency: Must have access to the flood con-ditions, notifications and geographic location information.
→ National Agency: Authorities receive timely warnings from the system of flood conditions and decide to no-tify the public regarding flood intensity with geographic location information.
→ The Network Operators: The device management capabilities for registering and configuring device services over a network.

B. The RUP '4 + 1' View Model

This model is to demonstrate its effectiveness in how the reference architecture can be instantiated.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

The RUP '4 + 1' View Model [33] is a well-established software engineering process of a specific set of views. First, we describe a scenario which can be a set of use cases.

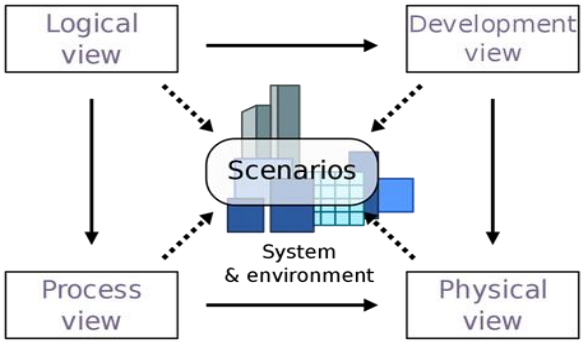

Fig. 5. The RUP '4 + 1' View Model

Scenarios View

The scenarios view focuses on illustrating and validating the designed architecture for very concrete use cases and application scenarios. Scenarios are a list of events which help elicit, validate, and document requirements. A scenario is a sequence of actions that illustrates behaviour, an interaction or the execution of a use-case instance.

Physical View

Looking at the system architecture deployment, topology, and communication from a system engineer's perspective we use a plethora of sensors and actuators in interacting with the environment.

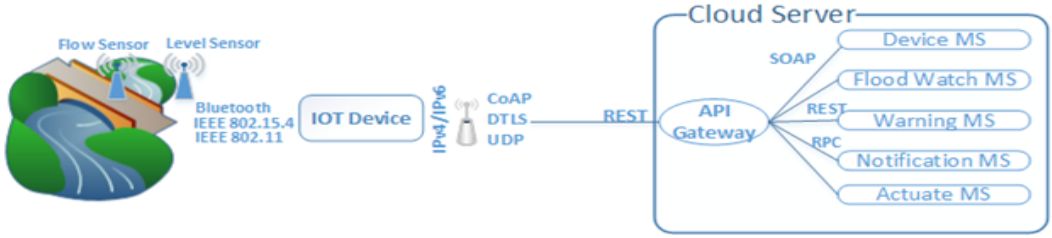

Fig. 6. Microservices Flood Forecasting System Architecture.

Logical View

The logical view is mandatory in the architectural context that describes objects and their interactions through remote wireless sensors in the application solution specified.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

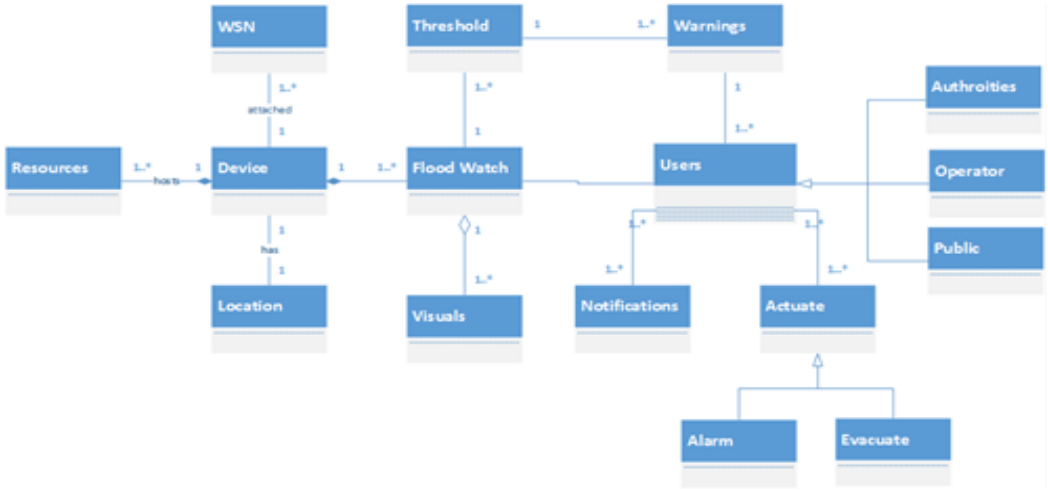

Fig. 7. Initial Object Model of Flood Forecasting System.

Process View

The process view focuses on the runtime behaviour of the system that the system processes and communicates. The process view deals with the dynamic aspects of the system. It illustrates the process decomposition and communication of the system in an activity diagram.

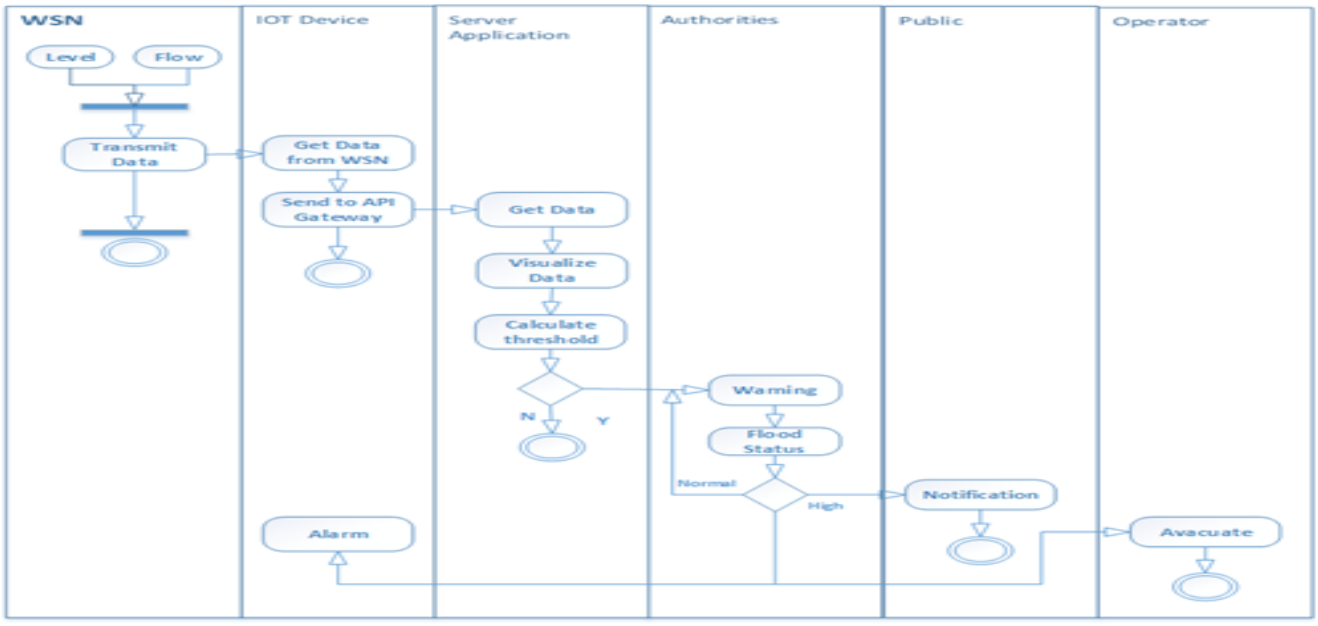

Fig. 8 Activity Flow Diagram of Flood Forecasting System

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

Development View

The Development View focuses on the static organization of the software in its development environment, e.g., component diagrams, and package diagrams. The flood management system is decomposed and developed into independently deployable applications exposed via restful HTTP APIs for device registration and configuration.

FloodWatch, Warning and Notification Microservices

This section describes building the IoT flood monitoring with MSA for a network of devices, to provide the implementation of the key elements i.e., IP-based device Flood Smart Meter. The actual water levels raw data is accessed and visualized via horizontal FloodWatch indicator as an HTTP RESTful microservice with the GIS to locate the appropriate location of the flood situation. It could be the base for receiving timely Warnings, Notifications and Actuation (Alert/Alarm), and are the use-cases/processes that each can be functionally decomposed into separate microservices.

- End-users and their Roles

The FloodWatch microservice can be implemented to provide businesses as in Fig 9. A Flood Smart Meter is an internet-connected device that broadcasts real-time flood-level data. Much of this information contains device location, flood flow, depth, and rainfall-measured values representation for visuals and threshold endpoints.

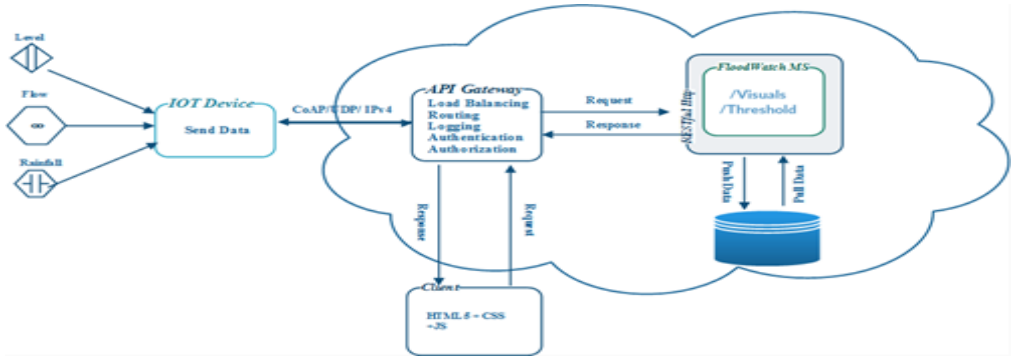

Fig. 9. FloodWatch, Warning, Notification MS Implementation

- Warning Microservice

A wireless sensor network sends weather information analyzed with decision tree techniques to detect floods. Warnings are generated when the FloodWatch application detects flood data,

available on the Web layer using REST APIs in (NodeJS, and AngularJS).

When it receives a timely warning based on threshold param-eters and events, it triggers to generate Notification and sound Actuate alarms. Emergency responders and authorities initiate a warning process with the intent to take appropriate action to reduce casualties and losses.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

TABLE I
Flood areas warnings and alerts

Description	Parameters	Endpoint
All flood warnings and alerts	county, lat=y, long=x	{root}/id/floods
All flood areas to which warnings and alerts	search=x, lat=y, long=x, dist=r	{root}/id/flood areas

Notification Microservice

The Notification microservice can be instantiated on the internet via GSM, Android/IOS apps, Radio networks, TV networks etc. The suggested system has email notifications (using SMTP protocol), Push notifications (web/mobile apps) and SMS alerts using GSM service or external app integration (Social media notifications). The Apache Thrift pushes notifi-cations to the client. Live Notifications (Keep everyone, and everything, in real-time with live notifications across HTTP, SMS, FCM, and APNS2).

VI. DISCUSSION AND FINDINGS

Understanding the purpose of communication between de-vices and applications is increasingly important as the number of connected devices grows. The significance of Microservices Architecture (MSA) lies in dividing functional actions onto scalable domains. Both IoT and MSA will require changes and adaptations in network architecture principles to address their unique conditions.

According to literature, Internet of Things (IoT) encompasses a wide range of technologies, including sensing, communication sub systems, data aggregation, pre-processing and service provisions. The IoT partially overlaps with the Internet of people and the Intranet or Extranet of Things. As a re-sult publications addressing IoT often also covers aspects of ubiquitous/ pervasive computing, internet communication and embedded device technology and applications [5].

This research significantly contributes to integrating microser-vices architectural styles into IoT solutions. The primary con-tribution is an extended approach that incorporates architec-tural meta-models based on specific constraints, technologies, and quality requirements for software architecture design and development within the field of Computer Science.

It will also attempt to set the foundation providing a overview of methodologies, processes, and tools for integrating meta-models. The analysis provides a list of important IoT-related ontologies for the orthogonal architecture approach in the holistic conceptualization of the problem domain. Conceptu-alization is the aim to answer specific focus questions through vigorous literature to explain a

Online ISSN 3007-3197
Print ISSN 3007-3189

phenomenon and create a conceptual model.
It will also attempt to set the foundation providing a temporal overview of methodologies, processes, and tools for integrat-ing meta-models. The analysis provides a list of important IoT-related ontologies for the orthogonal architecture approach in the holistic conceptualization of the problem domain. Con-ceptualization is the aim to answer specific focus questions through vigorous literature to explain a phenomenon and create a conceptual model.
Furthermore, it will derive a reference architecture, based on concepts and concerns introduced by the conceptual model. The specific technical contributions are relevant in software reference architectural support to exploit microservices to IoT solutions. Finally, it will contribute to instantiating and evaluating the proposed reference architecture into one or multiple concrete microservice IoT applications by specifying these infrastructures, integrated employing the reference decisions of different microservice architecture instances and service type implementation.

VII. CONCLUSION AND FUTURE WORK

We have provided the motivational context for this work, which aims to harness the potential of microservices for the Internet of Things (IoT). Our objective is to explore how previous research have supported IoT and microservices through architectural approaches. The microservices architec-tural approach organizes distributed applications as a suite of independent services. To create these applications, system architects must be highly educated and qualified researchers and developers across multiple disciplines, including advanced digital technology and classical engineering. This discipline is crucial for developing service contracts, which is a notable strength of this approach. Microservices also enable indepen-dent evolution by versioning the service APIs when there are incompatible API changes. This allows us to keep old versions of the interface operational while delivering new versions. Our future work involves proposing a novel taxonomy of MS-IoT architecture. This taxonomy will aim to exhaustively cover major technological recommendations for a framework that spans from sensors to applications. Additionally, selecting and applying predictive analytics algorithms involves building models from historical or current data to forecast future behaviour.

REFERENCES

[1] F. Mattern and C. Floerkemeier, “From the Internet of computers to the Internet of things,” in Lecture Notes in Computer Science, Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 242–259.
[2] D. NamIoT and M. S. Sneppe, “International Journal of Open Information Technologies,” ‘International Journal of Open Information Technologies’, ‘ISSN, vol. 2, 2014.

[3] Nginx.com.[Online].Available:https://www.nginx.com/wp content/uploads/2015/01/Building_ Microservices_ Nginx. pdf.[Accessed: 18-Mar-2024].
[4] M. Richards, Understanding Common Architecture Pat-terns and When to Use Them. O’Reilly Media, 2015.
[5] Www.internet-of-things.[Online]. Available:http://www.internet-of-things.[Accessed: 18-Mar-2024].

Online ISSN 3007-3197
Print ISSN 3007-3189

[6] A. Mcewen and H. Cassimally, Designing the Internet of Things. John Wiley Sons, 2014.
[7] "IndaField," indaField, 20-Jan-2015. [Online]. Available: http://www.indafield.com/BizTalkNotes/2015/01/20/IoT-as-the-big-driver-of-microservices/, [Accessed: 18-Mar-2024].
[8] N. Alshuqayran, N. Ali, and R. Evans, "A systematic mapping study in microservice architecture," in 2016 IEEE 9th International Conference on Service-Oriented Computing and Applications (SOCA), 2016.
[9] C. Pahl and P. Jamshidi, "Microservices: A systematic mapping study," in Proceedings of the 6th International Conference on Cloud Computing and Services Science, 2016.
[10] A. Levcovitz, R. Terra, and M. T. Valente, "Towards a technique for extracting microservices from monolithic enterprise systems," arXiv [cs.SE], 2016.
[11] B. Butzin, F. Golatowski, and D. Timmermann, "Mi-croservices approach for the internet of things," in 2016 IEEE 21st International Conference on Emerging Tech-nologies and Factory Automation (ETFA), 2016.
[12] I. Nadareishvili, R. Mitra, M. Mclarty, and M. Amund-sen, Aligning Principles, Practices, and Culture. 2016.
[13] T. Vresk and I. Cavrak, "Architecture of an interoperable IoT platform based on microservices," in 2016 39th Inter-national Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2016.
[14] O. Zimmermann, "Microservices Tenets: Agile Approach to Service Development and Deployment," SummerSoC, 2016.
[15] Linkedin.com.[Online].Available: https://www.linkedin.com/pulse/microservices-containers - future-IoT-robert-brewin. [Accessed: 18-Mar-2024].
[16] Z. Pang, "Technologies and Architectures of the Internet-of-Things (IoT) for Health and Well-being," KTH, 2013.
[17] I. Ishaq et al., "IETF standardization in the field of the internet of things (IoT): A survey," J. Sens. Actuator Netw., vol. 2, no. 2, pp. 235–287, 2013.
[18] A. Krylovskiy, M. Jahn, and E. Patti, "Designing a smart city internet of things platform with microservice architecture," in 2015 3rd International Conference on Future Internet of Things and Cloud, 2015.
[19] L. Lengyel, P. Ekler, T. Ujj, T. Balogh, and H.Charaf, "SensorHUB: An IoT driver framework for supporting sensor networks and data analysis," Int. J. Distrib. Sens. Netw., vol. 11, no. 7, p. 454379, 2015.
[20] A. Dahanayake and B. Thalheim, "A Conceptual Model for IT Service Systems," Journal of Universal Computer Science, vol. 18, no. 17, 2012.
[21] P. Bak, R. Melamed, D. Moshkovich, Y. Nardi, H. Ship, and A. Yaeli, "Location and context-based microser-vices for mobile and internet of things workloads," in 2015 IEEE International Conference on Mobile Services, 2015.
[22] D. Bandyopadhyay and J. Sen, "Internet of things: Ap-plications and challenges in technology and standardiza-tion," Wirel. Pers. Commun., vol. 58, no. 1, pp. 49–69, 2011.
[23] A. Olive, Conceptual Modeling of Information Systems. Berlin Heidelberg: Springer-Verlag,

Online ISSN 3007-3197
Print ISSN 3007-3189

2007.
[24] D. A. Moncus, DESIGN AND IMPLEMENTATION OF A CONCEPTUAL MODELING ASSISTANT (CMA). 2010.
[25] S. Angelov, P. Grefen, and D. Greefhorst, "'a Classi-fication of Software Reference Architectures: Analyz-ing Their Success and Effectiveness," in Joint Working IEEE/IFIP Conference on Software Architecture Euro-pean Conference on Software Architecture, 2009.
[26] IEEE Recommended Practice for Architectural Descrip-tion of Software-Intensive Systems. IEEE, 2000.
[27] B. P. Gallagher, "'Using the Architecture Tradeoff Anal-ysis Method to Evaluate a Reference Architecture: A Case Study," in Technical Note: Architecture Tradeoff Analysis Initiative, 2000.
[28] . Uckelmann, M. Harrison, and F. Michahelles, "An architectural approach towards the future internet of things," in Architecting the Internet of Things, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 1–24.
[29] P. Avgeriou, Describing, Instantiating and Evaluating a Reference Architecture: A Case Study. 2017.
[30] T. C. Lethbridge and R. Laganiere, Object-Oriented Software Engineering' 'Practical Software Development using UML and Java. 2001.
[31] P. Kruchten, "Architectural Blueprints-The '4+1' View Model of Software Architecture," 1995.
[32] J. Cubo, A. Nieto, and E. Pimentel, "A cloud-based Internet of Things platform for ambient assisted living," Sensors (Basel), vol. 14, no. 8, pp. 14070–14105, 2014.
[33] M. A. Paramartha, Design and Instantiation of Refer-ence Architecture of Pluggable Service Platform in E-Commerce. 2014.
[34] IoT-a.eu. [Online].Available:http://www.IoTa.eu/public/ publicdocuments/copy_of_d1.2/at_ download/file. [Accessed: 18-Mar-2024].
[35] M. Bauer et al., "Deliverable D1.3 - Updated reference model for IoT v1," vol. 5, 2012.
[36] A. Patidar and U. Suman, "Bharati Vidyapeeth's Insti-tute of Computer Applications and Management (BVI-CAM)," in 2nd International Conference on -Computing for Sustainable Global Development, New Delhi, INDIA, 2015.
[37] P. Bengtsson, Software Architecture - Design and Eval-uation. Sweden, 1999.

[38] V. Sugumaran and V. C. Storey, "The Role of Domain Ontologies in Database Design: An Ontology Manage-ment and Conceptual Modeling Environment," ACM Transactions on Database Systems, vol. 31, no. 3, 2006.
[39] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang, "A vision of IoT: Applications, challenges, and opportunities with China perspective," IEEE Internet Things J., vol. 1, no. 4, pp. 349–359, 2014.
[40] N. El-Bendary, M. Fouad, R. Ramadan, S. Banerjee, and A. Hassanien, "Smart environmental monitoring using wireless sensor networks," in Wireless Sensor Networks, CRC Press, 2013, pp. 731–754.
[41] C. Sunkpho, Real-time flood monitoring and warning system. Thasala, 2010.
[42] Weebly.com.[Online].Available:http://thebritishgeograph

Online ISSN 3007-3197
Print ISSN 3007-3189

er.weebly.com/uploads/1/1/8/1/11812015/the_pakistani_floods_of_2010.pdf. [Accessed: 18-Mar-2024].

[43] M. Ancona, A. Dellacasa, G. Delzanno, A. L. Camera, and I. Relliniy, "An Internet of Things Vision of the Flood Monitoring Problem," in The Fifth International Conference on Ambient Computing, Applications, Ser-vices and Technologies, Genoa, Italy, 2015.