Smart Video Services Based on Edge Computing with Multiple Cameras

Nikita Bazhenov, Dmitry Korzun
Institute of Mathematics and Information Technology,
Petrozavodsk State University (PetrSU)
Petrozavodsk, Russia
{bazhenov, dkorzun}@cs.karelia.ru

***Abstract*—Many existing video surveillance solutions are based on sending information between camera and remote server. In this paper, we study video services that are being built in an edge-centric Internet of Things (IoT) environment. Multiple cameras are available. Each video stream is subject to create simple events. Complex events can also be created based on the combination of simple events and context from IoT environment, forming the semantics for video analytics. We introduce microservice-based on step-by-step technology to show how software can be designed and developed for various application areas using multiple cameras. We extend video services that were discussed in our previous works using the developed technology. Our pilot services for smart video monitoring are in the following areas: Industrial Internet of Things (monitoring of production equipment), mHealth (monitoring of a patient in everyday life), Smart Room (monitoring of collaborative work participants).**

I. Introduction

The Internet of Things (IoT) technology supports development of smart environments, where the key element if service [1]. Smart video service provides the video surveillance and visual interactivity with the user [2]. Service provides analytical information about the object under monitoring [3].

We consider a video surveillance problem in respect to development of video services for various application domains. In this paper, we study smart video services development for the problem domains of mHealth, Smart Room, Industrial IoT (IIoT), continuing our work [4], [5], [6], [7]. To develop such services, a novel technology is constructed. We consider the basic development steps as well as models and tools to implement the steps.

Our study is focused on the following research and development problems, which reflect requirements from IoT-enabled smart environments as well as requirements to advanced data analytics and service delivery to the user.

- Service needs integration of video data from multiple cameras. It leads to the data fusion and semantics problems [8].
- Video data are processed using various hardware elements, also including local and mobile resources. It leads to the challenges of edge-centric computing [9].
- Service supports transformation of a camera into a smart IoT object (e.g., service makes the impression that the camera recognizes human movement). It leads to the intellectualization problem of IoT-enabled environments [10].
- Software agent makes a camera smart. If the camera capacity is low then the data analytics task is delegated to other resources nearby [7].
- Smart activity is reduced to event detection and evaluation in multiple video data sources. It leads to creating a shared information space that collects the semantics [1].

The introduced technology is used to construct virtual representation of the object under monitoring. As a result, services make impression that the object provides analytical information about itself (“telling the own story). In particular, such services are needed for controlling built-in equipment in building, for monitoring the people presence in a room, for evaluating human movement activity at home, and for many other practical problems.

The rest of the paper is organized as follows. Section II the studied class of smart video services and its application domains. Section III introduces smart video technology, which supports the key development steps for smart video service. Finally, Section IV concludes the paper.

II. Related work

We study an advanced class of video services deployable in edge-centric IoT environment. The key properties of the studied service class are microservices [11] for operating with multiple data streams,

Microservice architecture is used as an efficient way to separate components within a single application [12] describes the microservice as the shortest way to deliver services to the user. Existing service delivery solutions are inadequate because they include a large number of operations and calculations. Including, the described subject area too narrow or vice versa too wide. Thus, services are not delivered to customers properly when it needs to be done as soon as possible. Distinctive a feature of using microservices is the delivery of services using small stand-alone modules connected to a common service system and performing certain tasks. Microservice architecture is an alternative approach to structuring applications. In order to increase scalability and flexibility, the whole application is shared to independent components (modules).

In [13] authors discuss the use of smart contracts for the IoT areas. The main goal of the authors is to present solutions based on Smart Contracts that allow improving security and information management, identifying new opportunities and problems, and providing general recommendations, including safety recommendations. It is supposed to use smart contracts and blockchain technologies that create the potential for a viable solutions. The main functionalities of this technology are: Push-Pull (downloading and receiving new changes directly to the blockchain), Publish-Subscribe (publication / subscription scheme in an interactive model), Event-based (operations generated based on methods used by users). Main advantages for Smart Contracts are openness, invariability, provision of information on smart contracts is available at any time.

In [14] authors consider the organization of services in corporate information systems. A heterogeneous distribution of information is demonstrated due to the strong connection between the application and the system. The solution cited by the authors was generated on the basis of the following problems: inefficient code, delayed delivery project, code repetition, lack of documentation and productivity among developers software. The authors proposed an application called Enterprise Service Governance (ESG), which is designed to fix these problems. With this application, you can detect and register various services, search for an interface to search for service metadata that allows you to manage developing services throughout the life cycle and providing dependency analysis and reporting for service-oriented architectures.

Microservice architectures have a limited service taxonomy [15], when it comes to classifying service types, it basically consists of only two service types. Functional services are services that support specific business operations or functions, while infrastructure services support non-functional tasks, such as authentication, authorization, audit, registration, and monitoring. This is an important difference in the architecture of microservices, because infrastructure services are not exposed to the outside world, but rather are considered as private shared services, available only inside other services. Access to functional services is provided from the outside and, as a rule, is not transferred to other services.

Event-driven approach [16] allows to descry services construction from various angles. Object-oriented modeling allows you to describe entities with occurring phenomena in the form of flowcharts with a specific set of characteristics and properties. The use of such a methodology allows you to track the various phases at the stages of creating a software product. The event-oriented approach allows you to better define and describe the phenomena occurring within the system. Moreover, it allows to specifically describe the requirements at all stages of product design for a clear understanding by the developer of the minimum development costs required. In the case when there are a lot of video cameras and computing components, and the calculations are non-trivial and multi-layered, this approach will help solve this problem.

Context-based mechanisms [17] components are used in video analytics to separate image analysis into several phases. As a result, the algorithm yields a result on whether the image belongs to any class (for example, a person / not a person). The use of such segmentation allows you to more accurately determine the objects in the image depending on the context. Moreover, such sequences of images are conveniently stored in

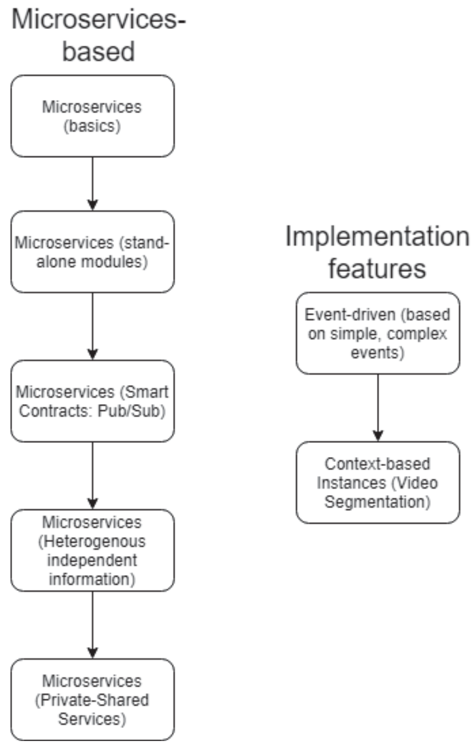

Fig. 1. Microserviced-based Escalating Architecture

a database, as well as their analysis at the stage of processing raw images.

Modern mobile healthcare systems [18] provide a wide range of services that organize the treatment of patients and have high accuracy in diagnosing diseases with low cost. mHealth must be properly designed and built to provide patients and doctors with mobile applications with high safety and treatment efficiency using appropriate algorithms and techniques. Also smartphones can be used not only to monitor the patient's heart rhythms, but also to monitor his condition using the video camera [4] . The camcorder can determine the person's state by recognizing his actions during the day, recognizing his temperature and emotions.

Microservices-based Escalating Architecture with main features is shown in Fig. 1. We want to use microservices and its architecture to create technology for designers and developers of software that would help to unify the creation of software. Thus, the basic architecture of microservices will be expanded due to independent modules, pub / sub technology, the possibility of using heterogeneous independent information, private-shared services.

Thus, using the above properties will provide the following properties: enhanced scalability, better fault isolation, localized complexity using microservices architecture and more detailed description of modules, enhanced service flexibility, simplified debugging and maintenance, app coordination between developers and end users, flexible development of software at the design stage thanks to event-driven management. Advantages designated by the authors in the above articles can be used in our developments. The main features in creating technology will be: focus on an expanded microservice architecture using various devices (and the possibility of using low-power com-

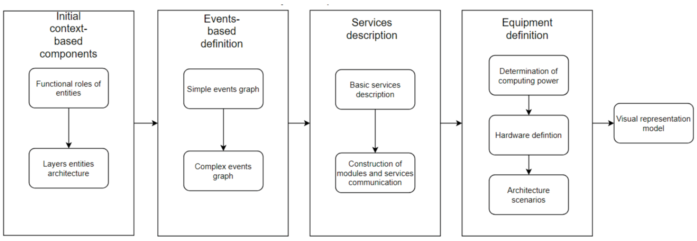

Fig. 2. Technology construction

puters, for example, microcomputers), tracking and generation of simple and complex events, context-sensitive instances.

III. Smart Video Technology

The complexity of developing video services in the IoT environment lies in the complexity of the general scheme of video and devices, which is difficult to understand, the dependence of some services on others (both at its level and on the links with neighboring levels), incoming large amounts of video data, heavy for fast analysis and interpretation.

Thus, there is a need to create a universal solution (technology) that would help resolve these difficulties. Attraction of microservice architecture will help organize simplicity of understanding and implementation of the general concept of technology, the independence of each of the modules and the autonomy of services, as well as the individual processing of various video data at each level.

Technology is a set of processes (steps) and methods for creating software. In this case it will be the sequence of the proposed steps that make smart video cameras out of ordinary video cameras, that is, video cameras that not only transmit the video stream to the Internet, but that can understand and analyze the current state in the video image of the camera, and independently interact with other system components (server, database) and have independent separate components that interact with each other. The final stage of the technology (result) will be video services - solutions that provide certain functionality in the form of an environment for interacting with the system (user interface or command line). Proposed technology steps will be described further.

Technology consists of the next steps as shown in Fig. 2:

- Initial initialization of computational entities, definition of hierarchy and structure of transmitted information (context-based);
- Definition of spatio-temporal components, discovery of relationships and attributes between components (events management);
- Definition of end services (services) and software modules defining them, definition of users by the system and services (video micro services, tree structure);
- Analysis and description of computing capabilities, equipment selection;
- Building a representative service-oriented model;

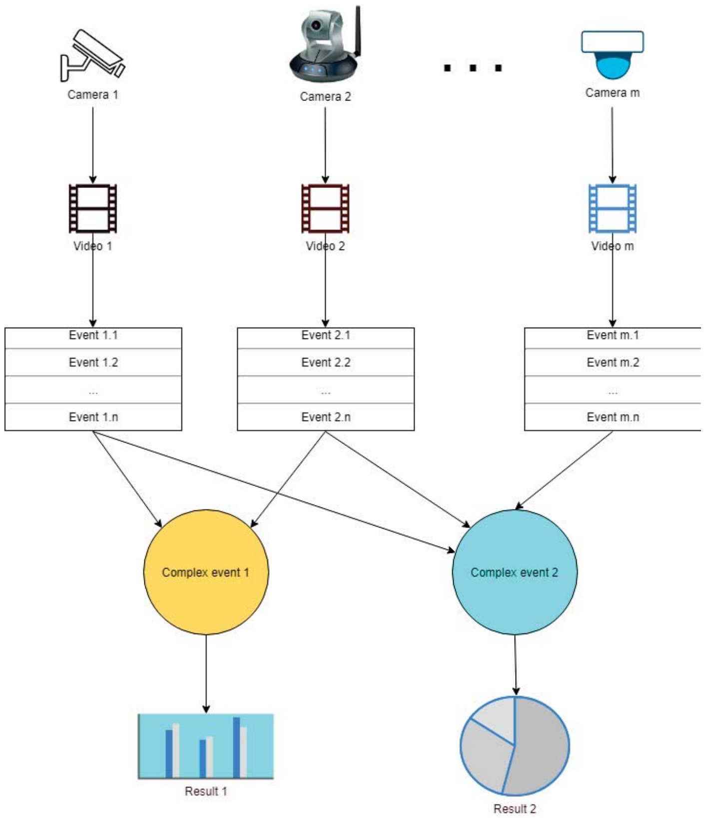

Fig. 3. Event model Based on Video Streams

Basic event	Algorithm	Implementation	Hardware requirements
Basic event 1..n	Required theoretical algorithms for basic event	Implementation for algorithm using programming language	What equipment can stably calculate algorithms
Complex event 1..n	Complex event building algorithms	Implementation/representation for algorithm using programming/modeling language	What equipment can be used in total

Fig. 4. Building IIoT Service for Employee Location Tracking Using Multiple Cameras Based on Smart CCTV Technology

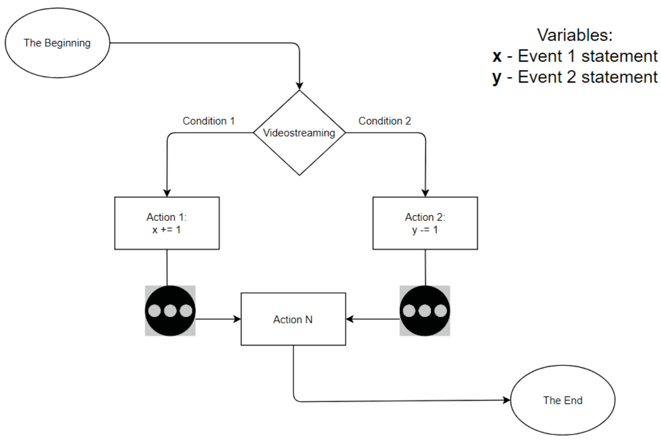

Fig. 5. Building Services Based on Smart CCTV Technology

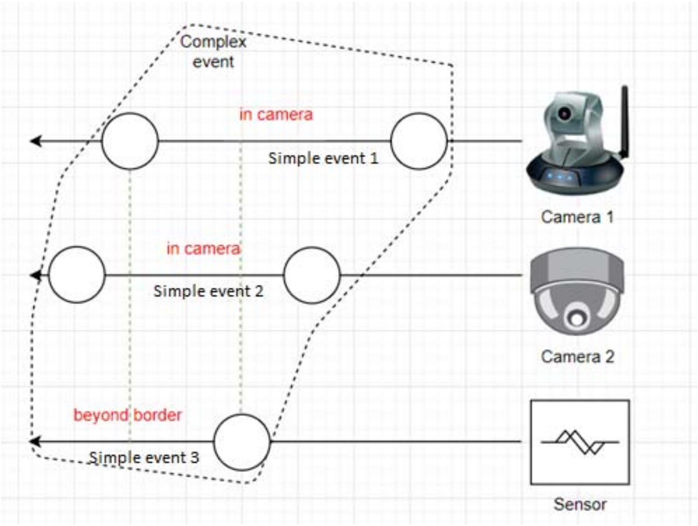

Fig. 6. A space-time graph of an event

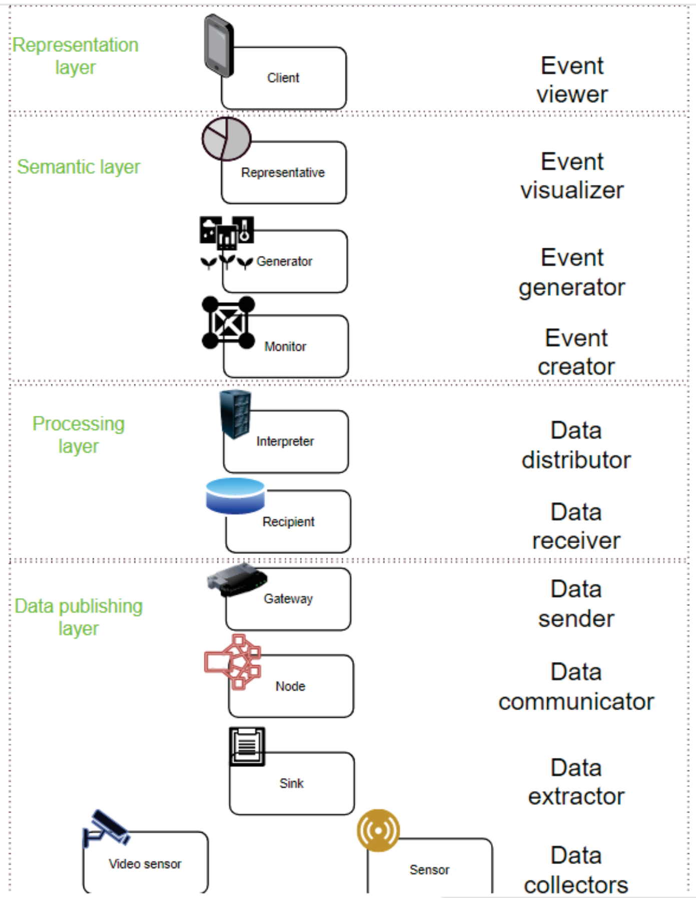

Fig. 7. Layers entities architecture

One of the main underlying architectures was the description of the functional roles of entities based on IoT and video services. Now we want to improve this by visualizing the basic interaction of functional roles in the form of architecture. If in the previous work the emphasis was placed on the description of roles, their activities and how they can be represented, here we want to show how entities will interact with each other at different levels of data transfer.

Then it will be shown how several different video streams that track simple events interact with each other and generate complex events. Complex events can be compiled on the basis of several simple ones received from video cameras and / or sensors. Complex events are generated at the semantic layer level using the complex event generation module. The generation module can be implemented on a remote or local server, depending on the implementation.

Next, we show the architecture of the implementation of modules, which will show how the video processing modules can be implemented in the system and how they interact with each other. The architecture consists of basic configuration and connection modules for video cameras, as well as modules for tracking and generating simple and complex events.

After that, a visual model of complex events will be compiled based on microservices using industrial monitoring video services as an example. This visual model shows which equipment is used to process data, which devices organize the calculations at their processing levels.

Step 1. Event model in Fig. 3. At this step, it is necessary to describe the events that will be ”monitored” based on the video streams received from the video surveillance cameras. The following rules are proposed as basic rules: each video camera generates an individual video stream, which can contain many key elements that make up a simple event. At the request of the developer, the underlying event may vary depending on time, space and context. The number of basic events from one video stream can be unlimited. Several video streams contain an n-number of basic events. Several basic events make up a complex event (based on 1..n video streams). A complex event can include several simple consecutive or simultaneous base events. Also, basic events can be independent and observed at different time periods with different durations, but make one complex. Complex events represent the final result, which can be presented in the form of a graph, chart, text. The result should be clear and representative to the user.

Step 2. Event recognition algorithms in Fig. 4. The developer must build a table of basic (simple) events and complex events (a chain of basic events). Depending on the underlying event, it is necessary to choose a theoretical algorithm by which this event must be implemented. The technology allows the developer to choose one of the theoretical algorithms that he will use depending on his needs. Algorithms differ in execution time and quality of the final results. So, some algorithms may show low accuracy of calculations, but at the same time run on low-power devices. As an implementation, a library is proposed, with which you can implement a theoretical algorithm in one of the programming languages. Hardware requirements vary depending on the chosen implementation. Many algorithms and their implementations have different configurations with parameters that differ in terms of the resources used.

Step 3. Block diagram of event recognition microservices in Fig. 5. The basic idea of this step is a consistent description of the actions and processes taking place inside the event. The proposed technology offers the use of flowcharts, however, the developer can choose the description of events according to his preference: these can be, for example, visual models or uml diagrams. The proposed description should include the beginning and end of the process. A separate event can be described (then the base point will be the beginning of the event, and the end point will be the final event), and the whole algorithm with several events as a whole. This description may include several subcircuits in which an event is defined. The flowchart should reflect the algorithm of sequential actions, which in one way or another will lead to the execution of the algorithm and its end.

Step 4. Building a space-time graph of an event in Fig. 6. Several sources of data generation (video cameras, sensors) receive data from the environment over time (that is, they change their state in space). In order to clearly demonstrate how several cameras and sensors interact with each other, depending on the context, it is necessary to build their dynamic states. The example shows how simple events from two cameras and one sensor generate one complex event (indicated on the graph by a dashed line on the time axis). The parameters for generating a complex event are set by the developer: for example, if the event occurs simultaneously for cameras and sensors (or occurs for only one component). In this example, the states of cameras and sensors are Boolean functions and are presented only in the form of 2 states. Such a representation can be complicated by different states of components that alternately change each other over time (for example, -1, 0, 1).

Step 5. Microservices and functional roles in Fig. 7. Entities that are intermediaries of information are on different layers of processing. The developer is invited to select data handlers from the above example and highlight the functional roles that these handlers must fulfill. The data publishing layer consists of data collectors: ”collectors” - processors that receive data from outside (for example, a video sensor or sensor), data extractors - software modules that extract data from sensors, nodes - data transmitters between other software entities, gates - data senders (for example, on the Internet - to another communication node). The processing layer consists of a data receiver (it can be a database or storage in the file system, both local and remote) and a data interpreter (server or broker that analyzes raw data from the database, choosing only the necessary data, for example, for the period , or with a certain frequency). The semantic layer consists of a monitor that monitors incoming (processed by the server) data and converts them into simple events. A generator based on many simple events builds complex events. Representative builds an interface from complex events that provides value from the user’s point of view. The presentation layer consists of clients who view events and interact with them. Such a scheme is extensible and can be supplemented with other functional roles, for example, in the case when events consist of several data layers and there is a need to create more complex analyzing structures.

Step 6. Hardware implementation model. This technology step is final, its result will be a designed model of all functioning devices, necessary computing modules and services, as well as actions and operations for interaction between components. This step is very demanding on all previous ones, since it completely depends on their complete, detailed and elaborated description. To build the model, it is necessary to connect the functional roles of the microservices described in the previous step 5 with the algorithms used and the minimum software from step 2 and the construction of the block diagram from step 3. The functional roles are represented by specific devices depending on the service implementation (for example, the Monitor will act specific module that performs calculations on the server).

The technology has the following advantages: peripherality and intelligence. Peripherality lies in a large number of different computing devices (laptop computers, mobile phones, laptops, microcomputers, and any devices capable of transmitting data using Bluetooth and Wi-Fi technologies. Intelligence is efficiency (that is, the calculations are made ”here and now” on the device a user who is in the system, including technology that allows the introduction of a voice assistant that will quickly analyze the information received from users and s necessary actions.

The proposed sequence of creating technology allows you to determine the necessary set of components that will be used in the system (video cameras, servers, databases, personalized mobile devices, microcomputers), the possibility of implementation in any complex system, the possibility of applying most technical video solutions. The main advantages of the technology (to describe): High performance, Low-performance devices (microcomputers), ubiquitous computing, Ambient Intelligence, Computational data heterogeneity, Semantic Web, Microservice architecture.

IV. Conclusion

This paper considered smart video services constructed using multiple edge devices. We discussed the development problem for such services, when services are constructed within IoT environments. The novel technology is introduced for development of smart video services in various application domains. We apply the microservice architecture, where a microservice is a kind of agent of knowledge processor to perform data analytics tasks. The technology supports multiple cameras such that a service is constructed based on events detected in multiple video data streams. The technology is experimented based on video services for the IIoT domain—digital monitoring of manufacturing processes.

Acknowledgment

This research is financially supported by the Ministry of Science and Higher Education of Russia within project no. 075-11-2019-088 in part of pilot implementation of IIoT services for smart manufacturing. This research is supported by RFBR (research project # 19-07-01027) in part of methods for IoT-enabled interaction and edge-centric computing. The work is implemented within the Government Program of Flagship University Development for Petrozavodsk State University (PetrSU) in 2017-2021.

References

[1] D. Korzun, E. Balandina, A. Kashevnik, S. Balandin, and F. Viola, *Ambient Intelligence Services in IoT Environments: Emerging Research and Opportunities*. IGI Global, 2019.

[2] K. Schoeffmann, M. A. Hudelist, and J. Huber, “Video interaction tools: A survey of recent work,” *ACM Computing Surveys (CSUR)*, vol. 48, no. 14, Sep 2015.

[3] Y. Chen, Y. Xie, Y. Hu, Y. Liu, and G. Shou, "Design and implementation of video analytics system based on edge computing," in *2018 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, 2018, pp. 130–137.

[4] A. Meigal, K. Prokhorov, N. Bazhenov, L. Gerasimova-Meigal, and D. Korzun, "Towards a personal at-home lab for motion video tracking in patients with Parkinson's disease," in *Proc. 21st Conf. Open Innovations Association FRUCT*, Nov. 2017, pp. 231–237.

[5] N. A. Bazhenov and D. G. Korzun, "Use of everyday mobile video cameras in IoT applications," in *Proc. 22nd Conf. Open Innovations Association FRUCT*, May 2018, pp. 344–350.

[6] N. Bazhenov, D. Korzun, and S. Balandin, "Smartphone-oriented development of video data based services," in *Proc. 23rd Conf. Open Innovations Association FRUCT*, Nov. 2019, pp. 62–68.

[7] N. Bazhenov and D. Korzun, "Event-driven video services for monitoring in edge-centric internet of things environments," in *Proc. 25th Conf. Open Innovations Association FRUCT*, Nov. 2019, pp. 47–56.

[8] A. D. W. Sumari and A. S. Ahmad, "Cognitive artificial intelligence: The fusion of artificial intelligence and information fusion," in *International Symposium on Electronics and Smart Devices (ISESD)*, Nov. 2016, pp. 1–6.

[9] H. Sun, Y. Yu, K. Sha, and B. Lou, "mVideo: edge computing based mobile video processing systems," *IEEE Access*, vol. 8, pp. 11 615–11 623, 2020.

[10] F. Viola, A. D'Elia, D. Korzun, I. Galov, A. Kashevnik, and S. Balandin, "The M3 architecture for smart spaces: Overview of semantic information broker implementations," in *Proc. of the 19th Conference of Open Innovations Association FRUCT*, S. Balandin and T. Tyutina, Eds. IEEE, Nov. 2016, pp. 264–272.

[11] H. Zhang, N. Yang, Z. Xu, B. Tang, and H. Ma, "Microservice based video cloud platform with performance-aware service path selection," in *Proc. 2018 IEEE Intl Conf. on Web Services (ICWS)*, Jul. 2018, pp. 306–309, 8456365.

[12] D. Nagothu, R. Xu, S. Y. Nikouei, and Y. Chen, "A microservice-enabled architecture for smart surveillance using blockchain technology," in *2018 IEEE International Smart Cities Conference (ISC2)*, Sep. 2018, pp. 1–4.

[13] N. Fotiou and G. C. Polyzos, "Smart contracts for the internet of things: Opportunities and challenges," in *2018 European Conference on Networks and Communications (EuCNC)*, Jun. 2018, pp. 256–260.

[14] A. G. Ozdogan, K. O. Akgun, and C. Kaya, "Service governance in service oriented architecture," in *2017 International Conference on Computer Science and Engineering (UBMK)*, Oct. 2017, pp. 195–200.

[15] M. Richards, "Microservices vs. service-oriented architecture," *OReilly Media, Inc.*, p. 55, 2016.

[16] W. Naiyapo and W. Jumpamule, "An event driven approach to create UML models," in *Proc. 22nd Intl Computer Science and Engineering Conference (ICSEC 2018)*, Nov. 2018, pp. 1–5, 8712621.

[17] J. Shao, X. Zhang, and Z. Cao, "Research on context-based instances selection of microservice," in *Proc. the 2nd Int'l Conf. on Computer Science and Application Engineering (CSAE '18)*. New York, NY, USA: ACM, 2018, pp. 1:1–1:5.

[18] A. P. S. Gagneja and K. K. Gagneja, "Mobile health (mhealth) technologies," in *2015 17th International Conference on E-health Networking, Application Services (HealthCom)*, Oct 2015, pp. 37–43.