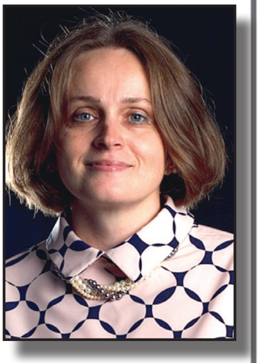
DOI: https://doi.org/10.30932/1992-3252-2021-19-1-18-46

Building Architecture of Intelligent Control System for Urban Rail Transit System

Victor M. ALEXEEV

Leonid A. BARANOV

Maxim A. KULAGIN



Valentina G. SIDORENKO

Alexeev, Victor M., *Russian University of Transport, Moscow, Russia.*
Baranov, Leonid A., *Russian University of Transport, Moscow, Russia.*
Kulagin, Maxim A., *Railway Research Institute, Moscow, Russia.*
Sidorenko, Valentina G., *Russian University of Transport, Moscow, Russia*.*

ABSTRACT

The increase in the volume of passenger transportation in megalopolises and large urban agglomerations is efficiently provided by the integration of urban public transit systems and city railways. Traffic management under those conditions requires creating intelligent centralised multi-level traffic control systems that implement the required indicators of quality, comfort, and traffic safety regarding passenger transportation. Besides, modern control systems contribute to traction power saving, are foundation and integral part of the systems of digitalisation of urban transit and the cities. Building systems solving the traffic planning and control tasks is implemented using algorithms based on the methods of artificial intelligence, principles of hierarchically structured centralised systems, opportunities provided by Big Data technology. Under those conditions it is necessary to consider growing requirements towards software as well as theoretical design and practical implementation of network organisation.

This article discusses designing architecture and shaping requirements for developed applications and their integration with databases to create a centralised intelligent control system for the urban rail transit system (CICS URTS). The article proposes the original architecture of the network, routing of information flows and software of CICS URTS. The routing design is based on a fully connected network. This allows to significantly increase the network bandwidth and meet the requirements regarding information protection, since information flows are formed based on the same type of protocols, which prevents emergence of covert transmission channels. The implementation of the core using full connectivity allows, according to the tags of information flows, to pre-form the routes for exchange of information between servers and applications deployed in CICS URTS. The use of encrypted tags of information flows makes it much more difficult to carry out attacks and organise collection of information about the network structure.

Platforms for development of intelligent control systems (ICS), which include CICS URTS, high computing power, data storage capacity and new frameworks are becoming more available for researchers and developers and allow rapid development of ICS. The article discusses the issues of interaction of applications with databases through a combination of several approaches used in the field of Big Data, substantiates combination of Internet of Things (IoT) methodology and microservice architecture. This combination will make it possible to single out business processes in the system and form streaming data processing requiring operational analysis by a human, which is shown by relevant examples.

Thus, the objective of the article is to formalise the principles of organising data exchange between CICS URTS and automated control systems (ACS) of railway companies (in our case, using the example of JSC Russian Railways), URTS services providers, and city government bodies, implement the developed approaches into the architecture of CICS URTS and formalise principles of organisation of microservice architecture of CICS URTS software. The main research methods are graph theory, Big Data and IoT methods.

Keywords: *urban rail transit, intelligent control, microservice architecture, network architecture, Big Data, software architecture, urban rail transit.*

*Information about the authors:
Alexeev, Victor M. – D.Sc. (Eng), Professor at the Department of Information Control and Protection of Russian University of Transport, Moscow, Russia, alekseevvm@rambler.ru.
Baranov, Leonid A. – D.Sc. (Eng), Professor, Head of the Department of Information Control and Protection of Russian University of Transport, Moscow, Russia, Baranov.miit@gmail.com.
Kulagin, Maxim A. – Deputy Head of the Technological Information Systems' Development Unit of Railway Research Institute (JSC VNIIZhT), Moscow, Russia, kulagin.maxim@vniizht.ru.
Sidorenko, Valentina G. – D.Sc. (Eng), Professor at the Department of Information Control and Protection of Russian University of Transport, Moscow, Russia, ✉ Valenfalk@mail.ru.

Article received 20.01.2021, accepted 26.02.2021.

For the original Russian text of the article please see p. 18.

BACKGROUND

The growth in the volume of passenger transportation in large urban agglomerations and megalopolises is effectively ensured by integration of public transit and urban railway lines. Traffic control in these conditions requires creation of intelligent centralised multi-level control systems that implement the specified indicators of quality, comfort, and safety of passenger transportation. Modern control systems additionally solve the problems of saving energy for traction of rolling stock, are the foundation and an integral part of digitalisation systems of urban transit and the cities.

The construction of systems that solve the problems of planning and traffic control is implemented via algorithms using artificial intelligence methods, the principles of hierarchically structured centralised systems, and the opportunities offered by Big Data technology. In these conditions, it is necessary to consider the increased requirements not only towards software, but also towards theoretical and practical solutions for network organisation. First, it should be noted that ICS process a much larger amount of information, since they use recognition methods to control access to infrastructure objects: this is background face recognition of personnel, intrusion detection, and an increased volume of technological video and audio communication, as well as of data on the parameters of many objects. Safety control systems record all staff negotiations. In this regard, the volumes of transmitted information significantly increase. If for audio communication, to ensure quality of transmitted information, it is sufficient to use information compression technologies in channels with an information transfer rate of up to 64 kB/s, then to support video communication, at least ten times better quality is required. Models of suburban railway traffic control within cities (for example, at Moscow Central Diameters (MCD) and Moscow Central Circle (MCC)) [1], on the metro and high-speed tram lines [2]) significant amounts of information from control systems monitoring access to infrastructure facilities are used, since it is necessary to quickly manage to stop the rolling stock in case of an emergency [3]. These systems are built using pattern recognition models, and, therefore, this leads to an increase in the volume of transmitted traffic via both wireless and optical information transmission channels. In this regard, the volume of stored information is calculated in petabytes. It should be noted that not only the volumes of transmitted information have changed, but also the essence of external and internal attacks on information resources [4–7] has changed accordingly. If earlier the attack supposed to interfere the functioning of a system, today the emphasis is on covert collection of information and transmission of information in a covert way.

Building a network using standard methods [8; 9], which were recommended and used in practice when implementing systems of a new class, such as ICS, cannot be applied. As a rule, networks are built according to a standard scheme with two or three central switches, to which peripheral switches are connected that connect various objects. These objects include servers, applications, databases, and users (dispatchers, programmers, administrators of various subsystems: security subsystems, LTE networks, databases, etc.). There are only two routes for organising information flows with three switches. This does not allow differentiating information transmission traffic, even when using visualisation in a local network. As a result, implementation of information flows focused on the use of direct transmission channels from objects (controllers) of a control system to objects (servers) is practically impossible. Meanwhile, ICS requires aggregation of a large amount of data and a high data transfer rate, which implies the use of optical interfaces for communication with information storage bases. In this regard, construction of ICS network core with many central switches opens up new possibilities for increasing speed, the volume of transmitted information, organising interaction of applications and databases, as well as providing information security.

It is known that power of a network is determined by the connections between nodes. The number of links B is determined as follows: $B = N (N - 1) / 2,$ where N is the number of nodes. The number of routes M in this case is equal to $M = B - 1.$ This allows channel transmission of information from a large number of objects (objects of automation and telemechanics, microprocessor centralisation (MPC), track circuits at stations and railway hauls, switch turnouts), from which it is necessary to collect information, and to transfer it to the database and process it with the appropriate application [10]. The use of fully connected nodes in the core of the network makes it possible to organise information flows for delivering information through direct channels using optical communication lines, which increases speed and volume of transmitted information tenfold. It should be noted that differentiation of flows by features makes it possible to make transmission over covert channels as difficult as possible, since the number of features that can be used to transfer information by permutations is limited, since practically a single protocol is used in each flow. In organisation of the network, connection of information flows at individual nodes is extremely rare, which is a necessary condition for a high transmission speed.

ICS network model is built based on the discretionary method [11], which means the unambiguous binding of the subject to the object. In practice, this means that the subject is assigned with personal objects in the network.

The *objective* of the research is to implement methods of intelligent protection of information flows by organizing temporary trusted routes based on tags and using a fully connected ICS network core, as well as interaction of applications of an intelligent control system based on the principles of microservice software architecture.

Based on this objective, the article discusses the issues of organising the network structure, organising temporary trusted routes for integration of subsystems of intelligent control of urban transit, as well as microservice architecture aimed at implementing program interaction of subsystems in a corporate network within an intelligent control system. To solve the set tasks, the *methods* of systems analysis, graph theory and computer science are used.

RESULTS

Structure of ICS network

In ICS, moving objects are controlled via 4*G* (*LTE*) wireless network [10; 11]. *LTE* network transfers heterogeneous traffic: voice, video, and data. In this regard, the use of *MPLS* multiservice network technology is a necessary and justified condition. We describe the discretionary model of *MPLS* network through the Cartesian product of objects and subjects:

$$(O_i^{MPLS} \cdot O_j^{MPLS}) \cdot S_k,\ i \neq j,\ k = 1,\ m,$$

where S_k are subjects, personnel;

m is number of subjects;

$\left(O_i^{MPLS} \cdot O_j^{MPLS}\right)$ – Cartesian product of objects in *MPLS* network.

MPLS network operates using specialised hardware, in the operating system of which special packet processing functions are implemented. A tag is added to the packet that contains the *IP* address prefix, ingress port, and egress port on *PE* (provider router) and *P* (*MPLS* network core router) routers. Routes are pre-recorded in the routing table. A 33-bit tag is processed quickly, therefore delivery speed increases. On *PE* routers, the tag is removed, and then the packets go to *CE* router (client router) and to the local network.

Let's consider operation of an application deployed at O_i^{MPLS} in *MPLS*[1] network [4–6]. The application Pr_l on the object O_i^{MPLS} is initialised by the subject and sends over the *Ethernet* interface an *Ethernet* packet containing the source and destination *MAC* addresses, as well as *DATA* area where data of the protocol used by the application Pr_l is stored. In *DATA* area of the *Ethernet*-packet there is an extension that contains the identifier N_{Pr_r} of the information flow generated by the application Pr_l. The identifier represents an encrypted record of the *MAC* address of the object O_i^{MPLS} and the number of the application operating in the network, as well as the parameter t_0 – start point of the flow transmission and t_{live} – life or duration parameter of the information flow. The purpose of parameters t_0 and t_{live} will be explained below. To form the code cipher, we use the symmetric key *kl*, which is predefined for the application and the server:

$$(MAC_{O_i^{MPLS}}, N_{\mathrm{Pr}_r}, t_0, t_{live})_{E_{kl}},$$

where E_{kl} – encryption E with the secret key *kl*.

Next comes the standard *MPLS* processing procedure. The packet arrives at *CE* router, which sends it to *PE* router, where a tag is affixed to the packet, and by the *IP* address prefix, it is directed to the egress *PE* router, where the tag is removed.

[1] Methodological document «Information protection measures in state information systems» (approved by the Federal Service for Technical and Export Control on February 11, 2014). [Electronic resource]: https://www.garant.ru/products/ipo/prime/doc/70491518/. Last accessed 17.02.2021.

Table 1

Algorithm for route formation

Identifier	Core object	Input port	Output port
$(MAC_{O_i^{MPLS}}, N_{\Pr_r}, t_0, t_{live})_{E_{kl}}$	O_j^{ISC}	e_i, i = 1, n	$e_k \times [Z], k = 1, n, k \neq i$
$(MAC_{O_{i+k}^{MPLS}}, N_{\Pr_{r+k}}, t_0, t_{live})_{E_{kl}}$	O_{j+k}^{ISC}	e_i, i = 1, n	$e_k \times [Z], k = 1, n, k \neq i$
The table is filled in accordance with the structural diagram of the network			

When passing through *MPLS* network, the original source *MAC* address is replaced, but it is retained in the extended *DATA* area.

After removing the tag, the packet with data arrives at the *Ethernet* interface of the switches of the nodes of the network core, where in the routing table for:

$$(MAC_{O_i^{MPLS}}, N_{\Pr_r}, t_0, t_{live})_{E_{kl}}$$

possible options for movement along the route are pre-registered. The routing is organised by a specialised server: Object Security Monitor (OSM).

Objects of *MPLS* network O_i^{PE}, connected to the outputs of *PE* routers and connected with a part of the objects of the core of ICS network O_j^{ISC}, can be described through the Cartesian product:

$$O_i^{PE} \cdot O_j^{ISC}, i \neq j .$$

A packet with the identifier

$$(MAC_{O_i^{MPLS}}, N_{\Pr_r}, t_0, t_{live})_{E_{kl}}$$

may appear on any of the ports of *MPLS* network objects $O_i^{PE.}$

It is possible to implement two options for routing information flows.

First option. OSM server, knowing all *MAC* addresses and numbers assigned to applications N_{Pr_r}, generates routes through the ICS core network. We describe the route through the starting and ending points:

$$M = M(O_j^{ISC}, O_k^{ISC}) ,$$

where *j* and *k* are the starting and ending points of the route.

To choose the shortest route, we will apply Dijkstra's algorithm with the parameter *C* = 1 and the matrix *[Z]* for controlling the occupancy rate of the node in the previously compiled route. The route is selected with the minimum distance to the final object:

$$d(M) = \sum_{i=1}^{n} C(O_j^{ISC}, O_k^{ISC}) ,$$

where $M(O_j^{ISC}, O_k^{ISC})$ is an arbitrary route from the initial *j*-th to the final *k*-th point. The delivery end point (application server, database, or other objects) is not determined by the *IP* address in *DATA* area, but by its *MAC* address (predefined), $C(O_j^{ISC}, O_k^{ISC})$ is distance from the initial *j*-th point to the final *k*-th one. The implementation of the routing algorithm is presented in Table 1.

In the description of the model for selecting the output port, the node occupancy control parameter is used as matrix [*Z*], the rows of which correspond to the ports of the object, and the columns – to the objects. At the intersection of columns and rows, 0 is put if the port is occupied, otherwise – 1. If the port is already occupied in the route, then when choosing a route for a set of objects working with another application, another free port of the object is selected.

Let us consider the purpose of the parameters of the start time t_0 and of the duration of the existence of the information flow t_{live}. The start time can be pre-agreed in the application, the beginning and duration of the information flow transmission can be determined [7; 12; 13]. This information is sent by the application to OSM server and is used by it to analyse the flow identifier, for example, when polling the sensors-controllers of the state of infrastructure elements, moving objects, objects of automation systems, etc. At the same time, there can be a single request from the application server to the controller object, where the start time and duration of the response flow are also written. Likewise, the server application informs the object security application. The timing parameters of the information flow for the security system are important to determine the lifetime of the route in the core of ICS network. The parameters t_0 and t_{live} are set by the application, determined from the necessary needs based on the knowledge of the values of the control parameters that are used in the application. At the end of $_{tlive}$ time, the route is cancelled. This means that any attack using a *MAC* address inside the network will not be possible because the trusted route is temporary and has been parsed.

Second option. The routing model is based on the use of local assignment of *MAC* addresses of the switch by its ports by the administrator. The tag about local control of the *MAC* address is put at the end, and the *MAC* address looks like: xxxx.xxxx.xxxxx.xxxx.xxxx.xx01. The route table in the second approach is built based o *MAC*-addresses, port, $m_{l\mathrm{tag}}$. The tag m_l is assigned based on the destination *IP* address in the *Ethernet DATA* packet area. In this case, *IEEE 802.1q* protocol is assigned in the network, which contains a byte with a m_l tag, a byte with a protocol type and an indicator that the network uses *IEEE* 802.1*q* protocol packets. The route table compares the tag assigned in advance by OSM server and the tag in *IEEE* 802.1*q* protocol, as a result, the packet is directed to the corresponding port of the switch. When the delivery endpoint is reached, the information tag is removed, and the data packet arrives at the server. It is to note that the use of this technology coincides with network virtualisation and, when fully connected, allows us to isolate information flows. It is to mention that it is one of the conditions for implementation of the network and information interaction for the critical facilities of the Russian Federation.

Let us consider a model of interaction between applications (various control tasks of ICS), which are controlled by dispatchers of URTS, administrators of *LTE* networks, networks of subsystems of URTS (metro, suburban railway transport within cities, high-speed trams):

$$(O_i^{ISC}\bullet O_j^{ISC})_{coreISC}\bullet(U_i\left(O_i^{pr}\bullet S_j^{ds}\right)+U_i\left(O_i^{adm}\bullet S_j^{adm}\right)+$$
$$+U_i\left(O_i^{db}\bullet S_j^{db}\right)+$$
$$\left[U_i\left(O_i^{MPLS}\bullet O_j^{MPLS}\right)\bullet S_j^{sp}\right]_{rail}+\left[U_i\left(O_i^{MPLS}\bullet O_j^{MPLS}\right)\bullet S_j^{sp}\right]_{metro}+$$
$$+\left[U_i\left(O_i^{MPLS}\bullet O_j^{MPLS}\right)\bullet S_j^{sp}\right]_{tram}+O^{id}+O^{usc}\right),i\neq j,$$

where $(O_i^{ISC}\bullet O_j^{ISC})_{coreISC}$ – core of ICS network;

$U_i\left(O_i^{pr}\bullet S_j^{ds}\right)$ is centralised traffic control systems by types of URTS: metro, suburban railway transit within cities, high-speed tram;

$U_i\left(O_i^{adm}\bullet S_j^{adm}\right)$ – objects associated with administrators of ICS system (system, network);

$U_i\left(O_i^{db}\bullet S_j^{db}\right)$ – objects associated with administrators of databases;

S_j^{sp} – service personnel;

$[U_i\left(O_i^{MPLS}\bullet O_j^{MPLS}\right)\bullet S_j^{sp}]_{rail}$ – a network of a centralised control system for electric train traffic of suburban railway transport within cities (CCSETT), which unites the centre for situational control of electric train traffic of suburban railway transport within cities and integrated subsystems of upper functional level of CCSETT;

$[U_i\left(O_i^{MPLS}\bullet O_j^{MPLS}\right)\bullet S_j^{sp}]_{metro}$ – a network of a centralised control system of metro train traffic (CCSMTT), which unites the centre for situation control of metro train traffic and integrated subsystems of upper functional level of CCSMTT;

$[U_i\left(O_i^{MPLS}\bullet O_j^{MPLS}\right)\bullet S_j^{sp}]_{tram}$ – a network of a centralised control system of high-speed tram traffic (CCSHSTT), which unites the centre for situational control of high-speed tram traffic and integrated subsystems of upper functional level of CCSHSTT;

O^{id} – a network of interdepartmental electronic document management,

O^{usc} – a network of city centre for off-street transport control.

The model also includes an isolated OSM security network[2, 3]. It contains a separate *GE* interface for managing core routes through routing tables. The discretionary model of the OSM server network is as follows:

$$O_{GE}^{SecO}\bullet[\left(O_i^{ISC}\bullet O_j^{ISC}\right)_{GEcoreISC}+\cup_i(O_i^{pr})_{GE}+\cup_i(O_i^{db})_{GE}+$$
$$+\cup_i(O_i^{MPLS})_{GErail}+\cup_i(O_i^{MPLS})_{GEmetro}+\cup_i\left(O_i^{MPLS}\right)_{GEtram}\right],$$

[2] Order of the Federal Service for Technical and Export Control of the Russian Federation of December 25, 2017 No. 239 On approval of requirements for ensuring security of significant objects of critical information infrastructure of the Russian Federation (as amended by Orders of the FSTEC of Russia dated August 9, 2018 No. 138, dated 26 March 2019 No. 60, dated February 20, 2020 No. 35). [Electronic resource]: https://fstec.ru/tekhnicheskaya-zashchita-informatsii/obespechenie-bezopasnosti-kriticheskoj-informatsionnoj-infrastruktury/288-prikazy/1592-prikaz-fstek-rossii-ot-2517-degnabrya-239. Last accessed 17.02.2021.

[3] GOST R ISO 14813-1-2011. National Standard of the Russian Federation «Intelligent Transport Systems. Scheme of building the architecture of intelligent transport systems. Part 1. Service domains in the field of intelligent transport systems, service groups and services». [Electronic resource]: https://docs.cntd.ru/document/1200086739. Last accessed 17.02.2021.

where O_{GE}^{SecO} are objects of the OSM server, applications, databases, *MPLS* network in an isolated OSM security network, connected via the *GE* interface.

A separate physical controlled (monitored) network interface for each external telecommunication service should be used in ICS network.

The model provides for an exchange with government agencies and business organisations (JSC Russian Railways, Moscow Department of Transport, etc.) through the system of interdepartmental document management[4, 5].

The structure of the centralised intelligent control system of the urban rail transport system (CICS URTS) is shown in the picture below (Pic. 1). List of information resources – applications for solving management problems comprises:

- an intelligent system for forecasting and analysing the passenger traffic of URTS;
- an intelligent system for forecasting, planning and analysing operation of URTS;
- an intelligent system for forecasting, planning and analysing the work of vehicle operators of URTS;
- an intelligent system for forecasting, planning and analysing the work of dispatchers of URTS;
- an intelligent system for forecasting, planning and analysing works on technical maintenance of URTS infrastructure;
- an intelligent system for forecasting, planning and analysing operation of vehicles of URTS;
- an intelligent system for control of traffic and vehicle safety system of URTS;
- on-board devices for unmanned vehicle control of URTS;
- situational centre.

The technical implementation of ICS network is based on the use of optical switches with a built-in control system through the *GE* port of OSM network. The ICS network core is physically located in the rack where the mesh network is deployed. Communication between *MPLS* networks of centralised traffic control systems by types of URTS is carried out on the basis of establishing connections between *PE* routers (unmarking) of the *MPLS* network and forwarding to a *CE* router connected to the core switches via optical cables with an exchange rate of up to 8 Gb/s.

The exchange rate in the core is 16 Gb/s with peripherals (personal computers of dispatchers, network, and application administrators, as well as CCS computers) up to 8–16 Gb/s. Database storages are formed according to the types of transport and types of management tasks to be solved for a specific type of transport. The exchange rate between objects of the network core and drives is 16 Gb/s. The storage capacity for each type of transport is: suburban rail transport within cities – up to 20 petabytes (PB), metro – up to 20 PB, high-speed tram – up to 3 PB.

According to security requirements, access to the Internet is prohibited for critical facilities, which include city transport[6, 7, 8].

The core of CICS URTS unites the following centres into a single hub: City centre for off-street transport control, connected with the centres of situational control by types of urban rail transit: suburban rail transit within cities, metro, high-speed tram, as well as integrated subsystems of the upper functional

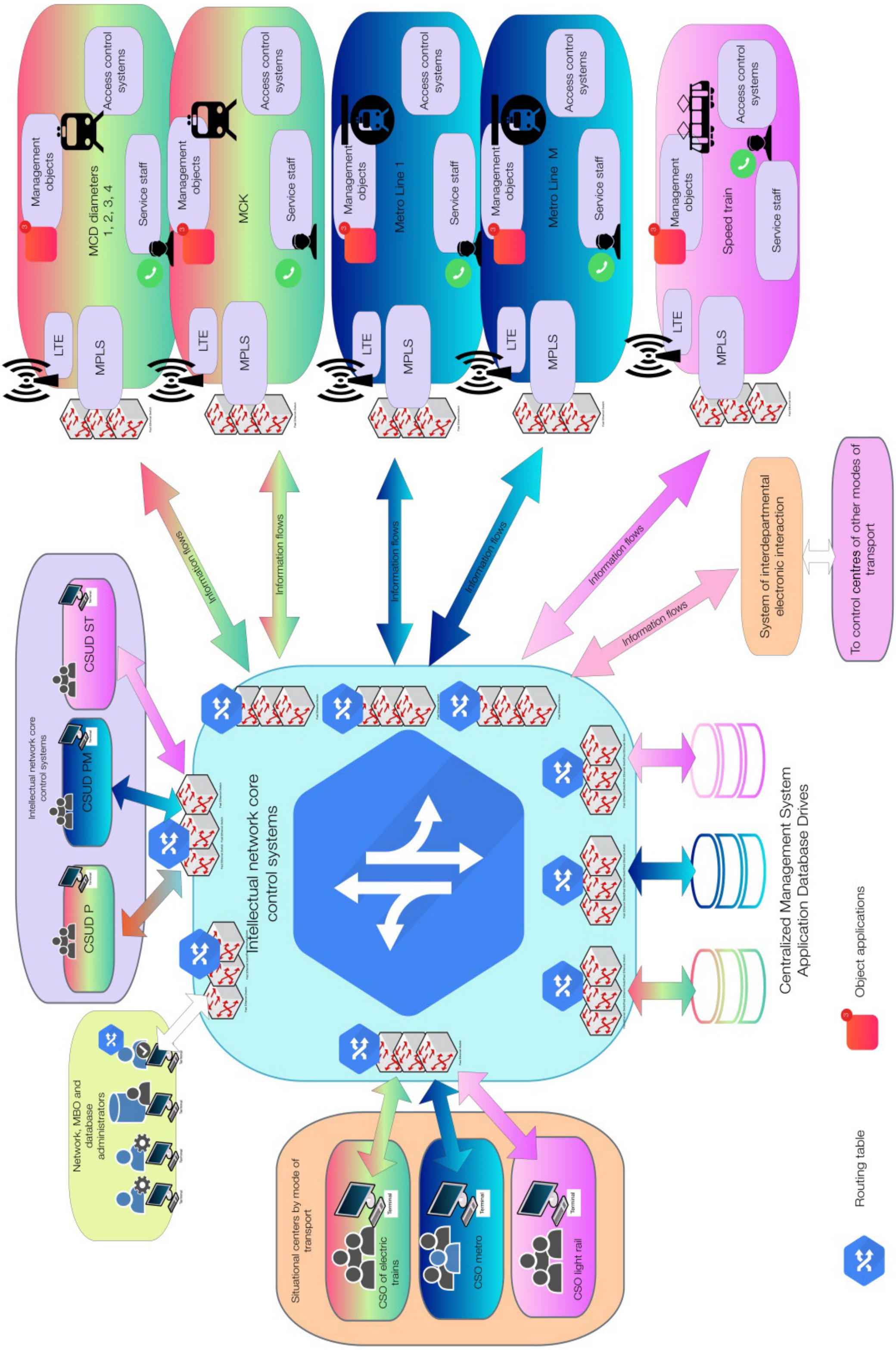
[4] Methodological document «Information protection measures in state information systems» (approved by the Federal Service for Technical and Export Control on February 11, 2014). [Electronic resource]: https://www.garant.ru/products/ipo/prime/doc/70491518/. Last accessed 17.02.2021.

[5] Order of the Federal Service for Technical and Export Control of the Russian Federation of February 11, 2013 No. 17 «On approval of the Requirements for protection of information that does not constitute a state secret contained in state information systems» (as amended on April 27, 2020, version effective from 1 January 2021). [Electronic resource]: https://docs.cntd.ru/document/499002630. Last accessed 17.02.2021.

[6] Methodological document «Information protection measures in state information systems» (approved by the Federal Service for Technical and Export Control on February 11, 2014). [Electronic resource]: https://www.garant.ru/products/ipo/prime/doc/70491518/. Last accessed 17.02.2021.

[7] Order of the Federal Service for Technical and Export Control of the Russian Federation of December 25, 2017 No. 239 On approval of requirements for ensuring security of significant objects of critical information infrastructure of the Russian Federation (as amended by Orders of the FSTEC of Russia dated August 9, 2018 No. 138, dated 26 March 2019 No. 60, dated February 20, 2020 No. 35). [Electronic resource]: https://fstec.ru/tekhnicheskaya-zashchita-informatsii/obespechenie-bezopasnosti-kriticheskoj-informatsionnoj-infrastruktury/288-prikazy/1592-prikaz-fstek-rossii-ot-2517-degnabrya-239. Last accessed 17.02.2021.

[8] GOST R ISO 14813-1-2011. National Standard of the Russian Federation «Intelligent Transport Systems. Scheme of building the architecture of intelligent transport systems. Part 1. Service domains in the field of intelligent transport systems, service groups and services». [Electronic resource]: https://docs.cntd.ru/document/1200086739. Last accessed 17.02.2021.



Pic. 1. The architecture of CICS URTS (compiled by the authors).

level of centralized traffic control means of suburban rail transport within cities, metro, and high-speed trams.

The ICS core provides the necessary resources for each centre: dispatching pools by types of urban transport separately for each metro line, a section of suburban railway transit within cities (diameter or ring) and a section of a high-speed tram; database; servers; access to corporate *MPLS* networks deployed on metro lines, sections of suburban railway transit within cities (diameter or ring) and sections of high-speed tram routes.

Administrators of databases, applications of top-level control systems by modes of transport and administrators of the *MPLS* network and *4G (LTE)* cellular communication, as well as security administrators are connected to the core of CICS URTS. The core of CICS URTS provides organisation of isolated large-volume information flows between application servers of control systems, through the use of optical interfaces and implementation of a fully connected network directed to the objects of the lower level of control systems.

Let us consider operation of the network for control of electric train traffic of suburban rail transit within cities. The basic devices that ensure safe movement are MPC systems that control switches, signals, track circuits, crossings and other devices within the station, automatic blocking systems (AB) that monitor the state of the track circuit in the control mode, power supply systems, and infrastructure control systems (track, roadbed, rails). The high-speed train control system aggregates information from these systems on the servers of radio block centres to transmit it to the on-board control systems on rolling stock (using a secure protocol) via the 4G (*LTE*) wireless communication system, where speed limit values are calculated to ensure the braking distance.

To ensure the security requirements of the MPC system, dispatch centralisation (DC), AB manage objects using their own servers using special protocols. To ensure the mode of secure exchange of information between the servers of the radio blocking centre, a *MySQL* buffer database has been created in CCSETT. Information from the database is sent through the *MPLS* network to the core of the network of CICS URTS to the servers of the upper functional level of CCSETT.

The subsystems of the upper functional level of CCSETT use data coming from the infrastructure control system through the *MPLS* network. Infrastructure control systems collect information from track elements to determine speed limits due to the condition of the track superstructure, repair work, and the absence of illegal entry of objects, animals, people. To ensure high-speed delivery of control information from the subsystem of the upper functional level of CCSETT to the on-board control systems on rolling stock, a route is formed between the traffic control server of rolling stock and the controller on rolling stock in the core of the central control system of URTS. For management packets, priority is set to at least 5 (*QoS*). Packet delay time is not more than 20 μs.

The upper functional level of CCSMTT is built for each metro line. The network of CICS URTS control performs the following functions: connects the servers of the upper functional level of CCSMTT of each line, on the one hand, and on-board control systems on rolling stock (the lower level of CCSMTT), infrastructure access control systems (control of appearance of foreign objects, people and other obstacles: landslides, groundwater, etc.), the Ministry of Emergency Situations and fire alarm systems, systems for diagnosing the technical condition of automation and telemechanics devices (automatic locomotive signalling systems with automatic speed control (ALS-ARS) and DC, preventing speed excess and ensuring that the train stops at emergencies), rolling stock, power supply systems on the other hand.

To implement the upper functional level of CCSMTT on metro lines, a multi-service *MPLS* network has been deployed, the task of which is to integrate a 4G (*LTE*) wireless information transmission network for communication of on-board control systems on rolling stock and technological communication systems for service personnel, integration via optical channels with access control systems, traffic safety systems ALS-ARS, DC, technical diagnostics systems for power supply devices, automation and telemechanics located on the metro line.

To ensure the mode of priority delivery of information from the applications of the servers of the upper functional level of CCSMTT, the isolated route mode (due to the full connectivity of the network objects) is implemented in ICS network core with a speed of 16 Mb/s (via an optical cable) and *QoS* mode in *MPLS* with a priority of at least 5, which ensures delivery of information from control servers to on-board control systems on rolling stock with a delay of no more than 20 μs.

Algorithms for maintaining the rolling stock of the metro and their parameters are determined at the upper functional level of CCSMTT, considering the current technical condition of the infrastructure and rolling stock, as well as the restraints formed by traffic safety systems.

Let's consider operation of the network for control of high-speed tram traffic. The connection between the on-board control systems on the high-speed tram and the upper functional level of CCSHSTT is carried out over *LTE* network through the corporate *MPLS* network. The location of the tram is determined based on a high-precision coordinate network and a digital track model, which provide high-speed tram positioning accuracy. The assessment of the state of the track is carried out using video cameras installed along the lines of movement. Based on the information received, speed limits or a signal to stop the train are generated. The required speed of information delivery from the upper functional level of CCSHSTT is ensured by setting priorities for control packets and an isolated route in the network core of CICS URTS.

ICS application-level structure

Let's consider the features of implementation of applications, their management, and their interaction with databases.

When creating CICS URTS, it is necessary to consider several bottlenecks in terms of storage of automation and interaction with external systems:

• storage and processing of information, which is necessary for functioning of CICS URTS. This information is collected in a large number of third-party automated systems. The existing automated systems use various technologies, interfaces, architecture;

• individual implementation of information exchange between automated control systems (ACS) of third-party developers;

• reliability of data stored in various ACS is not always ensured by appropriate technologies and procedures for logical data control;

• available ACS provide only operational data storage, long-term accumulation of information during the entire life cycle of control objects is not performed, which makes it difficult or impossible to effectively use predictive analytics tools, machine learning, etc.

To eliminate these restrictions and create CICS URTS, the following tasks should be solved:

• determination of the range of digital technologies that allow accumulating and analysing significant amounts of information from vehicles and infrastructure facilities in order to provide a flexible approach to the technical operation of URTS, technological equipment and engineering structures;

• development of data schemes, ensuring their storage in the structure of the distributed data warehouse of CICS URTS;

• creation of a distributed data warehouse for CICS URTS using big data technologies and a distributed register intended for joint use of CICS URTS and external data providers;

• creation of a mechanism for entering data into a distributed storage based on a unified *API*-interface on the initiative of adjacent automated control systems upon the appearance of data in them required by CICS URTS;

• creation of the possibility of entering data from on-board systems in real time.

• creation of functionality:

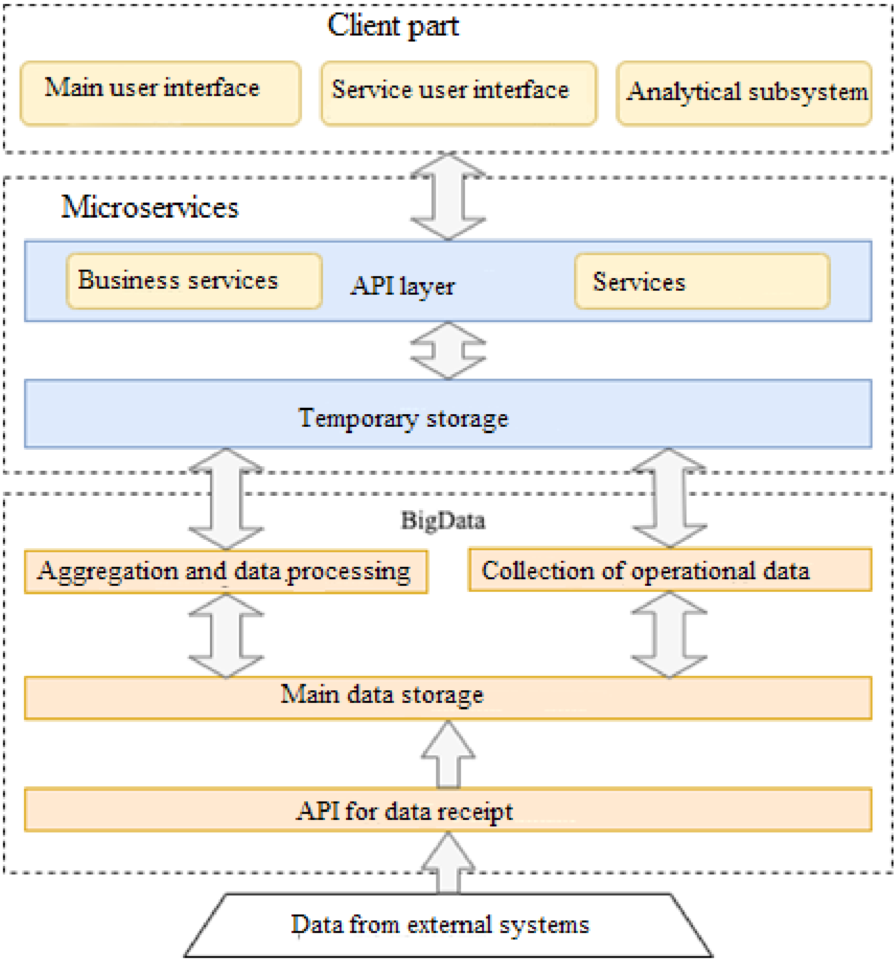
– logical control of data entering CICS URTS;

– management of conflicts and errors detected when data is received;

– visualisation of data stored in a distributed storage;

– ensuring the possibility of early identification of risks in management of maintaining the technical readiness of human and technical resources of URTS through adoption of modern analytical methods of statistical processing and machine learning;

– implementation of information services in the interests of consumers of information about URTS.



Pic. 2. The architecture of the system for collecting and processing a large amount of data for the implementation of CICS URTS (compiled by the authors).

Architecture of CICS URTS software

The authors have developed an architecture and set requirements for construction of a heterogeneous system for collecting and processing a large array of information coming from various sources, considering certain tasks.

Two architectures were taken as a basis for building a system for working with big data: *lambda* and microservice (MS).

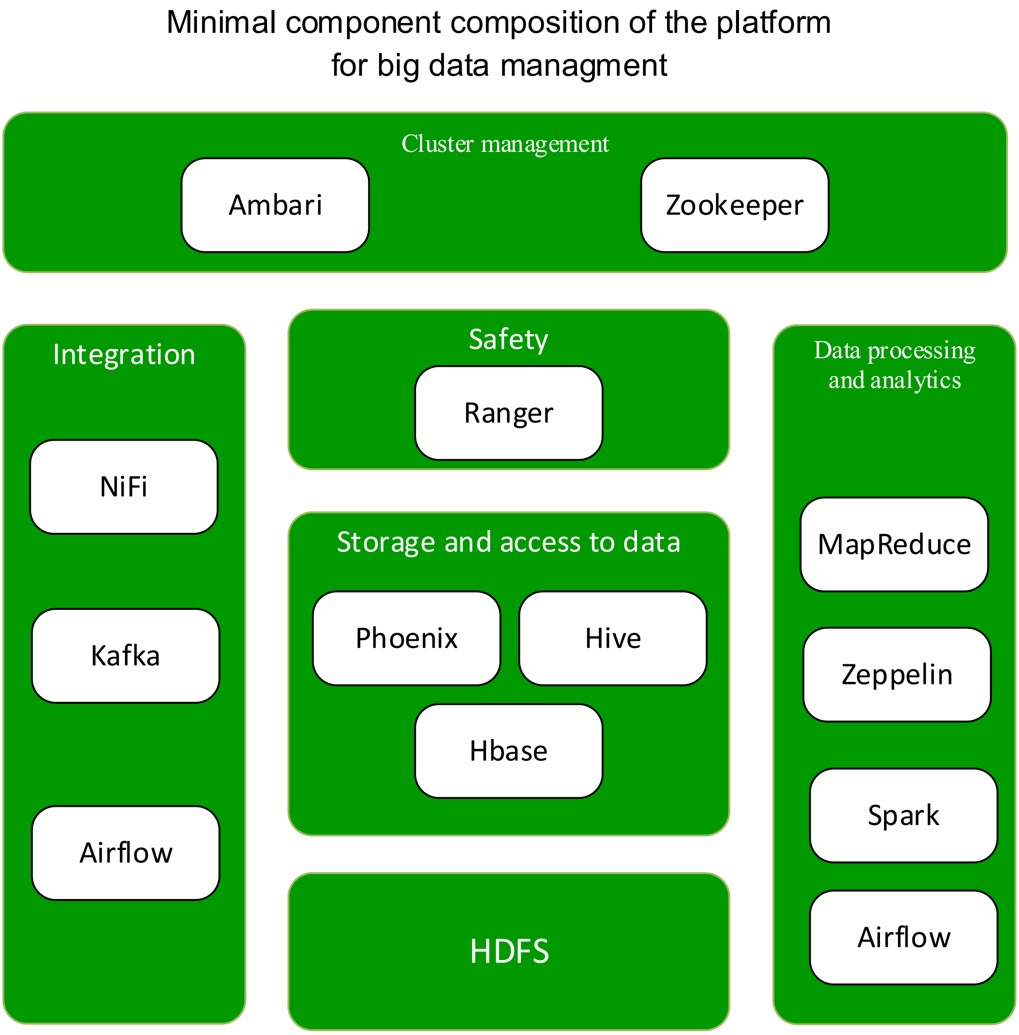
The architecture consists of three main blocks (Pic. 2):

- client part – services used by external clients;
- microservices – services that process requests from the client and interact with temporary storage;
- *BigData* – main storage with large data volumes that require significant resources for use and maintenance [14–18].

In the client part of the system, methods of interaction are implemented through the main and service user interface. The main user interface includes reporting forms for various business areas of the system. The service user interface includes subsystems for monitoring the general state of the system and provides direct access through the *web* interface to data for solving intelligent tasks.

The microservice part [19] consists of layers, each of which has its own logical level:

- *API* layer works directly with requests coming from the client part and requests the necessary data from the temporary storage. The *API* layer includes service and business modules.
- temporary storage – data storage with calculation results, operational data and user data. Technologies used: *MongoDB* [20] and *CouchDB* [21].


Minimal component composition of the platform for big data managment
Cluster management
Ambari
Zookeeper
Integration
NiFi
Kafka
Airflow
Safety
Ranger
Storage and access to data
Phoenix
Hive
Hbase
HDFS
Data processing and analytics
MapReduce
Zeppelin
Spark
Airflow

Pic. 3. Program composition of CICS URTS (compiled by the authors).

Big Data part consists of layers, each of which has its own logical level:

- aggregation and data processing – the service calculates features and forecasts based on the data received in the main storage;
- collection of operational data – an *API* service for transferring operational data to a temporary storage;
- main data storage – distributed file storage built using *HBase* [22] and *HDFS* [23] technologies;
- *API* for receiving data – a service for receiving data from external systems, built using *NiFi* technology [24].

The software environment can include (Pic. 3):

- *Unix* like operating system;
- *Ambari* – deployment and administration of *Hadoop* cluster [25];
- *Zookeeper* – a service that provides distributed synchronization of namespace configuration information for a group of applications used [26];
- *Ranger* – a module that provides central management of access control to files, folders, databases, tables [27];
- *NiFi* – a unified interface for collecting information from external systems [24].
- *Kafka* – queue manager [28];
- *Airflow* – a library for planning and monitoring workflows [29];
- *Phoenix* – declarative *SQL* syntax applied to the *HBase* database [30];
- *HBase*, *Hive* – distributed storage and synchronization of big data [25];

- *MapReduce* – a distributed computing technology used for parallel computing on large (up to several petabytes) datasets in computer clusters [31];
- *Zeppelin* – visualization and analytics of big data [32];
- *Spark* – a framework for distributed batch and stream processing of unstructured and semi-structured data [33];
- *HDFS* – distributed storage of big data files [25].

The system should provide the ability to store data from URTS during the entire life cycle of both the object itself and related objects. Data are not deleted and accumulated, forced cleaning is not provided. The possibility of data archiving is allowed.

The response speed of CICS URTS should be:

- for navigation operations on the screen forms of the system – no more than 5 seconds;
- for operations of generating reports – no more than 5 minutes.

The system should ensure that the following changes are applied:

- addition of new enterprises – users (functional customers) of CICS URTS;
- change of data sets coming from URTS.

The system must have open interfaces for development (modernization) and integration.

The system should provide the ability to gradually build up its functionality, including:

- replacement of hardware and system software, subject to compliance with the requirements specified in the documentation for standard and supplied software;
- expanding the functionality of the system in terms of development of new modules and subsystems, as well as integration with the developed automated and information systems.

Requirements for the modes of operation of the system are determined for the main (standard) mode of operation, in which the equipment, which constitutes a set of technical means, works properly, and the system, basic and application software functions properly; limited functionality mode (emergency mode); schedule of routine maintenance.

Reduced functionality mode is characterized by the failure of one or more software and/or hardware components. At the same time, the limited operability of the system for fulfilling the functional purpose remains. This can be caused by a communication failure or hardware and software failures, which cannot be compensated for by means of redundancy.

The routine maintenance mode is intended for carrying out repair and maintenance works and software modernization. In this mode, temporary (planned) unavailability of the productive system for users is allowed. The regulations for the transition from one mode of operation to another, as well as instructions for the personnel of the system for working in each of the above modes and for actions during the transition from one mode to another should be described in the operational documentation.

The requirements for diagnosing the system are that there should be tools for diagnosing operability of the main processes of the system and identifying events that are important from the point of view of functioning of the information system. This toolkit should provide the necessary monitoring of functioning of the information system by maintaining specialised registers (log files).

Complex monitoring of functioning of the information system is carried out by a combination of the following methods:

- standard monitoring of the information system – monitoring hardware resources, monitoring operability of processes and services;
- monitoring the availability of URL-links;
- monitoring of records in the log files of the information system, which have a standardized and documented structure;
- monitoring the content of the service tables of the information systems database;
- monitoring of messages received from the information system via the SNMP protocol to the monitoring servers, with an IP address set parametrically;
- other diagnostic methods agreed with the accompanying personnel and administrators of the monitoring system.

In case of emergencies or software errors, diagnostic tools should be able to store the complete set of information necessary to identify the problem.

The function of issuing reports on the system operation should be provided.

CONCLUSIONS

As a result of the study, it was proposed to use organisation of secure information flows for CICS URTS based on tags and implementation of a fully connected core of the system network.

The full connectivity of the core allows increasing speed of information exchange, dividing flows between ICS applications by protocol types and creating temporary trusted information flows with a minimum set of features, making it difficult to organize secret information transmission channels.

Implementation of CICS URTS network using a fully connected core and optical channels allows connecting *MPLS* network of the upper-level control subsystems with CICS URTS core by a local fully connected segment, which increases safety and volumes of information transmitted from the lower-level peripheral equipment.

The software architecture proposed by the authors will make it possible to organize the efficient operation of CICS URTS from the point of view of the processes of storing, receiving, and processing data. Including, given the fact that the system at the software level includes a microservice architecture, it allows for ease of maintenance, development, and deployment. Microservice architecture has proven its worthiness in development of an integrated model of the metro line and creation on its basis of a wide range of control, planning, optimisation and training tools [34–36], as well as in creation of the «Trusted environment of the locomotive complex» system [37].

REFERENCES

1. Vakulenko, S. P., Romensky, D. Yu., Mnatsakanov, V. A., Dorokhov, A. V., Vlasov, D. N. Development of options for modernisation of Moscow monorail transit system [*Razrabotka variantov modernizatsii Moskovskoi moorelsovoi transportnoi sistemy*]. *Metro i tonneli*, 2020, Iss. 4, pp. 28–36. [Electronic resource]: https://hutor.info/wp-content/uploads/2020/12/2_5332278966877161853.pdf. Last accessed 14.02.2021.

2. Shevlyugin, M. V., Korolev, A. A., Golitsyna, A. E., Pletnev, D. S. Electric stock digital twin in a subway traction power system. *Russian Electrical Engineering*, 2019, Vol. 90, Iss. 9, pp. 647–652. DOI: 10.3103/S1068371219090098. Last accessed 14.02.2021.

3. Pavlovsky, A. A., Okhotnikov, A. L. Information transport situation [*Informatsionnaya transportnaya situatsiya*]. *Nauka i tekhnologii zheleznykh dorog*, 2018, Vol. 2, Iss. 6, pp. 16–24. [Electronic resource]: http://www.vniias.ru/images/img/online_journal/pdf/02_2018/2_2018.pdf. Last accessed 14.02.2021.

4. Atroshchenko V. A., Rudenko M. V., Dyachenko R. A., Bagdasaryan R. Kh. On the issue of assessing the reliability of information to prevent mitm attacks when transmitting classified information through open communication channels [*K voprosu otsenki dostovernosti informatsii dlya predotvrashcheniya mitm-ataki pri peredachi zakrytoi informatsii po otkrytym kanalam svyazi*]. *Sovremennie problemy nauki i obrazovaniya*, 2013, Iss. 3, pp. 82–82. [Electronic resource]: https://science-education.ru/pdf/2013/3/375.pdf. Last accessed 14.02.2021.

5. Sudhir, Udipi. The event data management problem: getting the most from network detection and response. *Network Security*, January 2021, Vol. 2021, Iss. 1, pp. 12–14. DOI: https://doi.org/10.1016/S1353-4858(21)00008-8.

6. Halpin, T., Morgan, T. Information Modelling and Relational Databases (Second Edition). The Morgan Kaufmann Series in Data Management Systems 2008, 976 p. DOI: 10.1016/B978-0-12-373568-3.X5001-2.

7. Yue, Zeng; Ye, Baoliu; Tang, Bin; Guo, Songtao; Qu, Zhihao. Scheduling coflows of multi-stage jobs under network resource constraints. *Computer Networks*, January 12, 2021, Vol. 184, pp. 107686. DOI: 10.1016/j.comnet.2020.107686.

8. Wenstrom, M. Managing Cisco Network Security First Edition. Moscow, Publishing house Williams, 2005, 768 p.

9. Kharitonova, E. V. Graphs and networks [*Grafy i seti*]. Ulyanovsk, UlGTU publ., 2006, 92 p. [Electronic resource]: https://www.studmed.ru/haritonova-ev-grafy-i-seti_9d47b8a399b.html. Last accessed 14.02.2021.

10. Kuzyukov, V. A., Novikov, V. G., Safronov, A. I. Microprocessor control systems for movement of trains in Moscow metro [*Mikroprotsessornie sistemy upravleniya dvizheniem poezdov v Moskovskom metropolitene*]. *Avtomatika na transporte*, 2020, Vol. 6, Iss. 3, pp. 268–293. [Electronic resource]: https://cyberleninka.ru/article/n/mikroprotsessornye-sistemy-upravleniya-dvizheniem-poezdov-v-moskovskom-metropolitene/pdf. Last accessed 14.02.2021.

11. Devyanin, P. N. Models of safety of computer systems [*Modeli bezopasnosti kompyuternykh sistem*]. Moscow, Goryachaya liniya-Telecom publ., 2018, 338 p.

12. Zhang, Shunliang; Zhu, Dali. Towards artificial intelligence enabled 6G: State of the art, challenges, and opportunities. *Computer Networks*, December 24, Vol. 183, pp. 107556. DOI: 10.1016/j.comnet.2020.107556.

13. Mei, Lifan; Hu, Runchen; Cao, Houwei; Liu, Yong; Han, Zifa; Li, Feng; Li, Jin. Realtime Mobile Bandwidth Prediction Using LSTM Neural Network. In: Choffnes D., Barcellos M. (eds) Passive and Active Measurement. PAM 2019. Lecture Notes in Computer Science, Vol. 11419. Springer, Cham. https://doi.org/10.1007/978-3-030-15986-3_3.

14. Kalipe, G. K., Behera, R. K. Big Data Architectures: A detailed and application-oriented review. *Int. Journal Innov. Technol. Explor. Eng.*, 2019, Vol. 8, pp. 2182–2190. [Electronic resource]: https://www.researchgate.net/profile/Rajat-Behera/publication/336915402_Big_Data_Architectures_A_detailed_and_application_oriented_review/links/5dba7a2e4585151435d62a79/Big-Data-Architectures-A-detailed-and-application-oriented-review.pdf. Last accessed 14.02.2021.

15. Thurner, T. Big Data Europe for Smart, Green and Integrated Transport [Electronic resource]: https://www.w3.org/community/bde-transport/files/2015/11/Big-Data-for-Smart-Green-and-Integrated-Transport-Workshop-Final-Report.pdf. Last accessed 14.02.2021.

16. Avdeeva, I. L. Analysis of foreign experience in the use of global technologies «Big Data» [*Analiz zarubezhnogo opyta ispolzovaniya globalnykh tekhnologii «Big Data»*]. *Internet journal «Naukovedenie»*, 2016, Vol. 8, Iss. 6, pp. 1–11. [Electronic resource]: http://naukovedenie.ru/PDF/13EVN616.pdf. Last accessed 14.02.2021.

17. Bik, R. Application of Big Data in transport planning [*Primenenie Big Data v transportnom planirovanii*]. [Electronic resource]: https://transport.mos.ru/common/upload/docs/1500293313_Moovit_Moscow_International_Transport_Expert_Council_R.pdf. Last accessed 14.02.2021.

18. Big Data technology in transport. How big data has become a valuable asset in transportation. Domestic

DBMS Tarantool in a big data analytics project [*Tekhnologiya Big Data na transporte. Kak na transporte bolshie Dannie prevratilis v tsenniy aktiv. Otechestvennaya SUBD Tarantool v proekte analitiki bolshikh dannykh*]. [Electronic resource]: https://tygeza.ru/tehnologiya-big-data-na-transporte-kak-na-transporte-bolshie-dannye.html. Last accessed 14.02.2021.

19. O'Connor, R. V., Elger, P., Clarke, P. M. Continuous software engineering – A microservices architecture perspective. *Journal of Software: Evolution and Process,* 2017, Vol. 29, Iss. 11, pp. e1866. [Electronic resource]: https://www.researchgate.net/profile/Rory-Oconnor-4/publication/316009873_Continuous_software_engineering-A_microservices_architecture_perspective/links/5a26bc4e4585155dd423eecc/Continuous-software-engineering-A-microservices-architecture-perspective.pdf. Last accessed 14.02.2021. DOI: 10.1002/smr.1866.

20. Boicea, A., Radulescu, F., Agapin, L. I. MongoDB vs Oracle-database comparison. 2012 3rd International Conference on Emerging Intelligent Data and Web Technologies. IEEE, 2012, pp. 330–335. [Electronic resource]: https://www.researchgate.net/profile/Alexandru-Boicea/publication/261040647_MongoDB_vs_Oracle_--_Database_Comparison/links/55c2132b08aebc967defd053/MongoDB-vs-Oracle--Database-Comparison.pdf. Last accessed 14.02.2021. DOI: 10.1109/EIDWT.2012.32.

21. CouchDB. Apache CouchDB. [Electronic resource]: https://couchdb. apache.org. Last accessed 14.02.2021.

22. Vora, M. N. Hadoop-HBase for large-scale data. Proceedings of 2011 International Conference on Computer Science and Network Technology. IEEE, 2011, Vol. 1, pp. 601–605. DOI: 10.1109/ICCSNT.2011.6182030.

23. Shvachko, K., Kuang, H., Radia, S., Chansler, R. The Hadoop Distributed File System. 2010 IEEE 26th symposium on mass storage systems and technologies (MSST). IEEE, 2010, pp. 1–10. DOI: https://doi.org/10.1109/MSST.2010.5496972.

24. Kim, S.-S., Lee, W.-R., Go, J.-H. A Study on Utilization of Spatial Information in Heterogeneous System Based on Apache NiFi. 2019 International Conference on Information and Communication Technology Convergence (ICTC). IEEE, 2019, pp. 1117–1119. DOI: 10.1109/ICTC46691.2019.8939734.

25. Venner, J., Wadkar, Sameer; Siddalingaiah, Madhu. Pro Apache Hadoop. Apress, Berkeley, CA, 2014, pp. 399–401. DOI: 10.1007/978-1-4302-4864-4_9.

26. Hunt, P., Konar, M., Junqueira, F. P., Reed, B. ZooKeeper: Wait-free Coordination for Internet-scale Systems. USENIX annual technical conference, 2010, Vol. 8, Iss. 9, pp. 1–14. [Electronic resource]: https://www.usenix.org/legacy/events/atc10/tech/full_papers/Hunt.pdf. Last accessed 14.02.2021.

27. Gupta, M., Patwa, F., Sandhu, R. An Attribute-Based Access Control Model for Secure Big Data Processing in Hadoop Ecosystem. Proceedings of the Third ACM Workshop on Attribute-Based Access Control, 2018, pp. 13–24. [Electronic resource]: https://www.researchgate.net/profile/Maanak-Gupta/publication/323785048_An_Attribute-Based_Access_Control_Model_for_Secure_Big_Data_Processing_in_Hadoop_Ecosystem/links/5ab0253b0f7e9b4897c1d52b/An-Attribute-Based-Access-Control-Model-for-Secure-Big-Data-Processing-in-Hadoop-Ecosystem.pdf. Last accessed 14.02.2021. DOI: 10.1145/3180457.3180463.

28. Wang, Guozhang; Koshy, Joel; Subramanian, Sriram; Paramasivam, Kartik; Zadeh, Mammad; Narkhede, Neha; Rao, Jun; Kreps, Jay; Stein, Joe. Building a replicated logging system with Apache Kafka. Proceedings of the VLDB Endowment, 2015, Vol. 8, Iss. 12, pp. 1654–1655. DOI: 10.14778/2824032.2824063.

29. Singh, P. Airflow. Learn PySpark. Apress, Berkeley, CA, 2019, pp. 67–84. [Electronic resource]: https://link.springer.com/content/pdf/10.1007%2F978-1-4842-4961-1.pdf. Last accessed 14.02.2021.

30. Akhtar, S., Magham, R. Using Phoenix. Pro Apache Phoenix. Apress, Berkeley, CA, 2017, pp. 15–35. [Electronic resource]: https://link.springer.com/content/pdf/10.1007%2F978-1-4842-2370-3.pdf. Last accessed 14.02.2021.

31. Condie, T., Conway, N., Alvaro, P., Hellerstein, J., Elmeleegy, K., Sears, R. MapReduce Online. Conference: Proceedings of the 7th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2010, San Jose, CA, USA, April 28–30, 2010, Vol. 10, Iss. 4, pp. 313–328. [Electronic resource]: https://www.usenix.org/legacy/events/nsdi10/tech/full_papers/condie.pdf. Last accessed 14.02.2021.

32. Cheng, Yanzhe; Liu, Fang; Jing, Shan; Xu, Weijia; Chau, Duen Horng. Building Big Data Processing and Visualization Pipeline through Apache Zeppelin. PEARC'18: Proceedings of the Practice and Experience on Advanced Research Computing, 2018, pp. 1–7. DOI: 10.1145/3219104.3229288.

33. Zaharia, M. [*et al*]. Apache spark: a unified engine for big data processing. Communications of the ACM 59.11, 2016, pp. 56–65. DOI: 10.1145/2934664.

34. Baranov, L. A. Balakina, E. P., Erofeev, E. V., Sidorenko, V. G. Multifunctional models of control systems [*Mnogofunktsionalnie modeli sistem upravleniya*]. *Izvestiya vysshykh uchebnykh zavedenii. Problemy poligrafii i izdatelskogo dela*, 2012, Iss. 2, pp. 79–82. [Electronic resource]: https://publications.hse.ru/mirror/pubs/share/folder/54otctuizr/direct/118351222.pdf. Last accessed 14.02.2021.

35. Sidorenko, V. G., Zhuo, M. A. Investigation of the possibility of applying genetic algorithms to solving problems of planning operation of metro electric rolling stock [*Issledovanie vozmozhnosti primeneniya geneticheskikh algoritmov k resheniyu zadach planirovaniya raboty elektropodvizhnogo sostava metropolitena*]. *Elektronika i elektrooborudovanie transporta*, 2017, Iss. 6, pp. 37–40. [Electronic resource]: https://publications.hse.ru/mirror/pubs/share/direct/213900236.pdf. Last accessed 14.02.2021.

36. Kulba, V. V., Kovalevsky, S. S., Kosyachenko, S. A., Kuznetsov, N. A. Methods of analysis and synthesis of modular information and control systems [*Metody analiza i sinteza modulnykh informatsionno-upravlyayushchikh sistem*]. Moscow, Fizmatlit publ., 2002, 800 p. [Electronic resource]: http://bookfi.net/book/1471957. Last accessed 14.02.2021.

37. Kharin, O. V., Yakimov, S. M., Kulagin, M. A., Gonik, M. M., Khludeev, M. A., Yaroshchuk, D. I. Automated system trusted environment of the locomotive complex (2019). Certificate of registration of the computer program RU 2020613754, 23.03.2020 [*Avtomatizirovannaya Sistema doverennaya sreda dlya lokomotivnogo kompleksa (2019). Svidetelstvo o registratsii programmy dlya EVM RU 2020613754, 23.03.2020*]. [Electronic resource]: https://www.elibrary.ru/item.asp?id = 42709956. Last accessed 14.02.2021.

Acknowledgments. The research was carried out with the financial support of the Russian Foundation for Basic Research, NTU Sirius, JSC Russian Railways and the Educational Foundation «Talent and Success» in the framework of scientific project No. 20-37-51001.