VOL. 11, NO. 19, OCTOBER 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

DESIGNING ARCHITECTURE OF SOFTWARE FRAMEWORK FOR BUILDING SECURITY INFRASTRUCTURE OF GLOBAL DISTRIBUTED COMPUTING SYSTEMS

Pavel Sergeyevich Ptitsyn[1], Dmitry Vladimirovich Radko[2] and Oleg Victorovich Lankin[3]
[1]Research Institute of Semiconductor Engineering, JSC, Leninsky Prospekt, 160a, Voronezh, Russian Federation
[2]Voronezh Innovation and Technology Center, LLC, Leninsky Prospekt, 160a, Voronezh, Russian Federation
[3]Voronezh State University of Engineering Technologies, Revolyutsii Prospekt, Voronezh, Russian Federation
E-Mail: pptitsyn@mail.ru

ABSTRACT

The implementation of global distributed information systems is built on various technical platforms and design approaches. There are many problems of ensuring a high level of information security of these systems, because they operate critical or confidential data, and the components of these systems are located in different physical sites. The existing distributed information systems as a rule do not meet the current challenges in the field of ensuring a high level of information security. In addition, the actual questions about the integration of these systems with corporate information systems. The aim of this work is to provide flexible architecture approach for building security software framework of global distributed computing systems. Given security framework provides access control management, identity management, user session management, monitoring and auditing, management of data sources, programming interfaces. The design includes application architecture, describing the major functional subsystems, system architecture, describing the software and hardware that implement security infrastructure, data architecture, describing the main business entities and relational data model.

Keywords: distributed computing systems, security framework, security infrastructure.

1. INTRODUCTION

Distributed information systems allow creating a geographically distributed computing infrastructure that brings together diverse resources and the ability to implement multiple access to these resources.

Currently the most prevalent approach to the implementation of distributed information systems is built on cluster and grid solutions, connected by a network of loosely coupled heterogeneous computers [1, 2, 3].

The basic elements for providing the functions of information security distributed systems are the following [4, 5]:

- Authentication - process of identity participant interaction. In traditional systems, the authentication process provides the protection of the server. In the global distributed systems, however, in order to protect themselves from intruders, it is equally important to check the authenticity of the server.
- Authorization - determines the acceptability of the system requested operation. The global distributed system for the adoption of similar solutions, such mechanisms must act based on the rules established for each resource.
- Confidentiality and data integrity - transmitted or stored data should be protected by adequate mechanisms to prevent illegitimate access. In some cases, it is necessary to ensure complete isolation of a specific data set from illegitimate users.
- Monitoring and auditing. To create large-scale distributed structures, organizations require mechanisms that monitor and calculate the amount of resources used. Accounting mechanisms also ensure that all parties comply with the agreement on the use of resources. The audit of the transaction information makes it possible to identify the source of danger or security breach.

The information security solutions of distributed systems must interact successfully with existing local security infrastructure. However, the diversity of local solutions significantly complicates creation of integrated security systems. It should be noted that one of the possible ways to address the difficulties of this kind is the use of common standardized approaches to the implementation of local solutions to ensure information security of distributed systems [6].

The aim of this work is to design architecture approaches for development of software framework for building security infrastructure of global distributed computing systems. Given security framework will be optimized for operation across a wide range of distributed systems, including obsolete. This security framework will implement the basic functions of information security based on improved methods and algorithms, and the use of open standards for building distributed systems such as SOA, OGSA, and WSRF. In addition, this security framework will provide integration infrastructure with corporate information systems, and will have advanced tools for the development, support and administration of security subsystems.

2. METHODS

The developed approach for security provision of distributed computing systems is based on the principles of centralized access to distributed resources by allocating dedicated security servers [7, 8], which will provide the following functions:

- Connection, authentication, authorization of end users and application services based on complex approach including flexible mechanisms of adjustment and adaptation algorithms of secure access functions.
- Configurable authentication and authorization.
- Authentication and authorization of web services using SOAP, REST, WSS and OGSA standards.
- Control of the lifetime and session management of the users and the resources.
- Messaging security using signed hashes in order to avoid substitution requests and content of transmitted data.
- Flexible programming interface, which exposes secure functions based on open SOA architecture.
- Graphical user interface to configure and support security servers.

The principal scheme of the centralized security server and related distributed computing systems is represented on Figure-1.

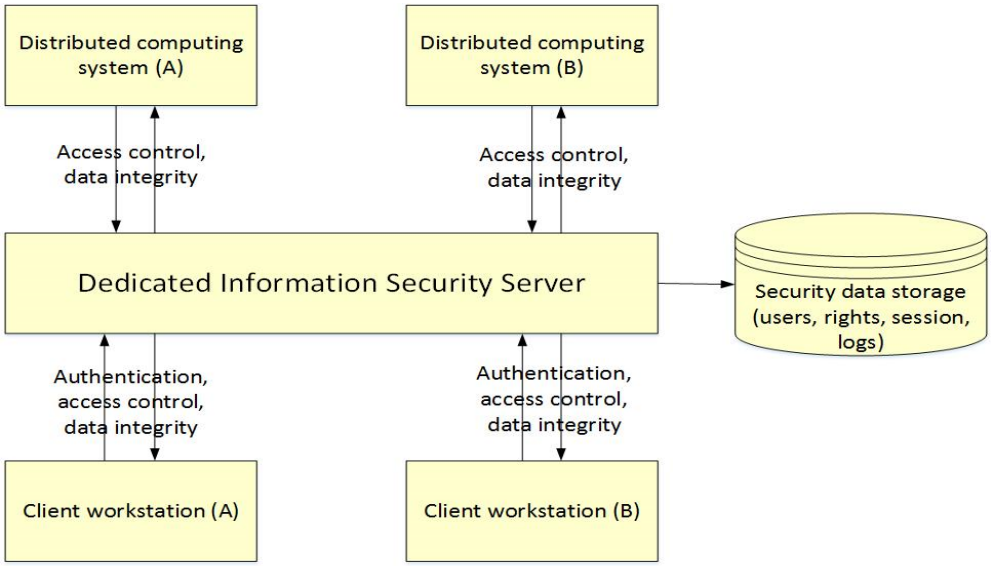

Figure-1. The principal scheme of working the centralized security server.

A distinctive feature of the implemented approach to the security server will support the following schemes the connection and the authentication on the server security side [9]:

- Basic authentication user scheme - given scheme provides mechanism, which establishes connection, creates session, and checks user credentials without the usage of additional data protection algorithms. In this case, it is necessary to provide more efforts of the defense at the level of data transmission channels, such as SSL, TLS.
- Authentication scheme using data encryption algorithms - given scheme provides mechanism, which establishes connection, creates session, and checks user credentials using the data encryption algorithms. In this case, the data exchange is performed by an encrypted channel using asymmetric encryption algorithms.
- Combined authentication scheme – given scheme provides mechanism, which establishes connection, creates session, and checks user credentials using the data encryption algorithms, and the formation of a unique code that contains algorithms for user authentication and encryption key generation algorithms based on the user data. In this case, the client received special generated assembly, which provided the data exchange by secure channel. This approach provides additional level of security because it is necessary the significant time for decompilation and modification of this assembly that is not comparable to the connection session lifetime.

The basis of the developed security software framework would be on the following main architectural and technological design principles of complex distributed information systems [10, 11]:

- The system provides information interaction with different categories of users, information resources and systems.
- The system built on modular principles, provides flexible configuration and extension of the functional and technical capabilities of the system.
- The system uses a centralized distributed database that contains related user data, as well as a description of the business logic to process the related user data.
- The system supports distributed data access and distributed business transactions in cluster, grid and cloud environment.
- The system implementation is built on three-tier architecture, which includes data layer, business logic layer and user interface layer. On all layers of implementation will ensure the transfer of security context.
- The system has a flexible configuration (user interface, business logic, data forms) and is open, which allows for integration cooperation with a wide range of information systems.
- The modules of the system will be developed using the methodologies of designing distributed and scalable applications (OGSA, SOA, MDA, ESB, MVC), as well as open standards (XML, SOAP, WSDL, REST, Web Services).
- Table-1 presents the main advantages of the proposed architectural and technological approaches for the development of security framework of distributed computing systems [12, 13].

Table-1. The main advantages of the proposed architectural and technological approaches for the development of security framework of distributed computing systems.

Architecture features	Functional capabilities
Loosely-coupled	▪ It enables organizations quickly rebuild their business processes in accordance with the changes, due to the weak links between parts of the organization and production processes (services). ▪ Reducing cost of the implementation due to the high degree of re-use of existing services.
Flexibility	▪ It allows the use of all investment previously invested in information technology. ▪ Avoids redesign and re-testing of existing systems.
Modularity	▪ It provides scalable design, implementation and support. ▪ With increasing functionality of the system, its complexity is not increased. ▪ The possibility of gradual replacement of software and hardware.
Standardized	▪ It allows avoiding software lock-in. ▪ Reducing complexity and fragmentation from the use of closed products.
Multi-tiered	▪ The ability to handle large volumes of data. ▪ High performance of data processing. ▪ Reliable protection against unauthorized access to data. ▪ High reliability of data storage. ▪ The consistency of data by setting rules of integrity. ▪ Ability to work with distributed databases and replication. ▪ High scalability, the ability to increase the number of system users.

The proposed architecture approach for building information security framework of distributed computing systems is represented on Figure-2.

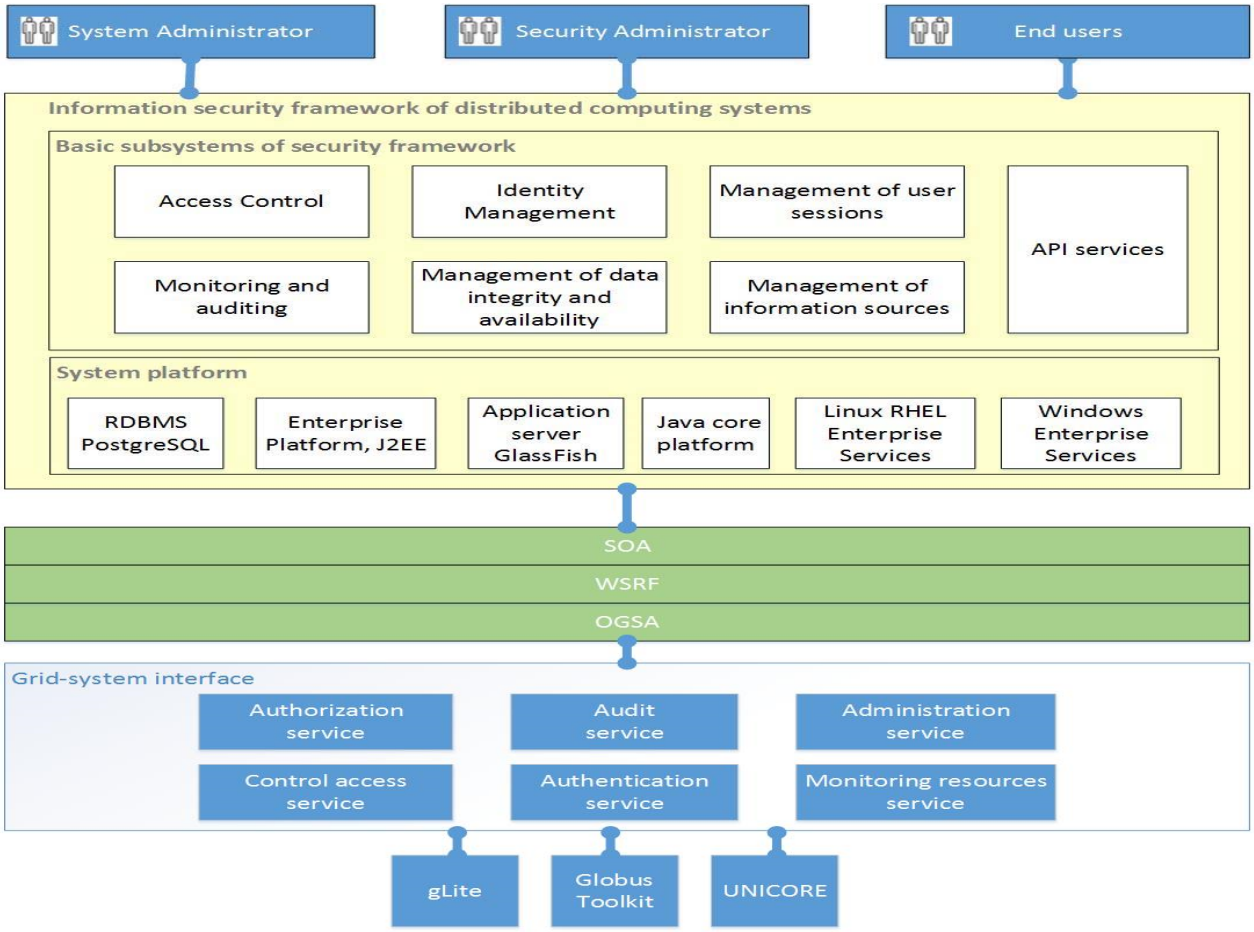

Figure-2. The architecture approach for building information security of distributed computing systems.

The proposed architecture contains the following basic subsystems [14, 15]:

- Access Control - subsystem manages user access to distributed information systems and resources, including the functions of the user registration, account management, the delegation of user rights.
- Identity Management - subsystem provides authentication mechanisms are built on open standards, and the provision of appropriate rights to access the resources.
- Management of user sessions - subsystem provides mechanisms for managing user data in the distributed IT- environment.
- Monitoring and auditing - subsystem provides a mechanism for collecting and analyzing information on users activity and conditions of IT-infrastructure.
- Management of data integrity and availability - subsystem provides mechanisms to protect the security infrastructure from unauthorized attempts to modify the data, and attempts to disable the security of infrastructure.
- Management of information sources - subsystem provides secure integration mechanisms of interaction with the external distributed information systems and data sources.
- API services - subsystem ensures the implementation of the application programming interfaces for access to the security infrastructure from external applications.

The proposed architecture approach supports the following information security features [16, 17]:

- Single sign-on (SSO) authentication is built on open standards such as SSO SAML, OpenID and OAuth.
- PIN-code authentication.
- Support for the OGSA architecture, including services, which responsible for authentication,

authorization, access control data, monitoring and administration.

- Support for the information security context of the enterprise platforms, including SAP ERP, Microsoft Dynamics, 1C.
- Support for access to repositories of unstructured data such as Apache Hadoop using Web-service oriented OData protocol.
- Support for legally significant document workflow in accordance with ISO 27000 standards.
- Ability to operate as active load-balancing clusters.
- Enhanced delegation of user rights, including functions to write business logic using script languages.
- Flexible customization capabilities and adaptation using the designer report, which provides custom business entities, rules, role models, and tools for constructing custom UI forms.
- Flexibility to build and publish security audit reports.

The infrastructure for building secure distributed computing systems is represented on Figure-3.

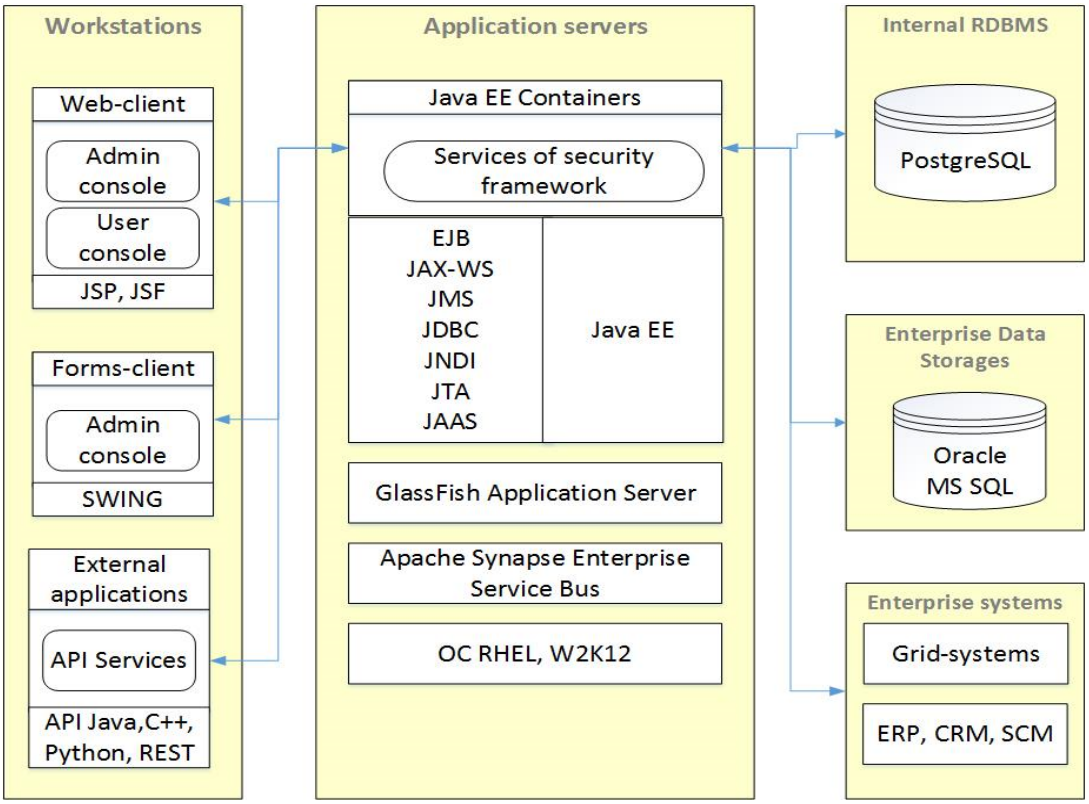

Figure-3. The security infrastructure for building global distributed computing systems.

For development of software framework for building security infrastructure of global distributed computing systems the following technical solutions based on open-source software will be used:

- Core Java programming language and Java Platform Enterprise Edition 7.1, which provide the building of high-performance, scalable, and reliable applications.
- High performance GlassFish Application Server 3.1, which supports operating in cluster modes.
- Apache Synapse Enterprise Service Bus 2.1, which provides flexible ways to integrate enterprise applications using SOA approach.
- The data storage will be implemented using PostgreSQL 9.4 database configured for operation in cluster mode.
- Server operating system including Linux RHEL 2.4 or Microsoft Windows Server 2012.

- Programming languages such as C ++, Java, Python for the implementation of API libraries.
- NetBeans IDE 8 software development platform.

The security software framework of global distributed computing systems will be based on the following application services of Java Platform Enterprise Edition [18, 19]:

- JSP (Java Server Pages) technology provides API for building graphical user web-interface.
- JSF (Java Server Faces) technology provides implementation of the MVC design pattern for building graphical user web-interface.
- SWING technology allows creation of window graphical user interface.
- JAX-WS (Java API for XML Web Services) technology provides creation of Web services in accordance with the specifications SOAP \ XML-RPC.
- JAX-RS (Java API for RESTful Web Services) technology provides creation of Web services in accordance with REST specification.
- JMS (Java Message Service) technology provides a mechanism for the exchange of messages and provides guaranteed message delivery.
- EJB (Enterprise Java Beans) technology provides server-side components that implement business logic.
- JTA (Java Transaction API) technology provides a mechanism for managing distributed business transactions.
- JDBC (Java Database Connectivity) technology provides API for accessing various data sources.
- JNDI (Java Naming and Directory Interface) technology provides API for accessing LDAP directory services.
- JAAS (Java Authentication and Authorization Service) technology provides support for information security infrastructure of Java-based applications.
- JACC (Java Authorization Contract for Containers) technology provides support for interfaces of authorization services.

The common data model [20] of security software framework of global distributed computing systems is represented on Figure-4.

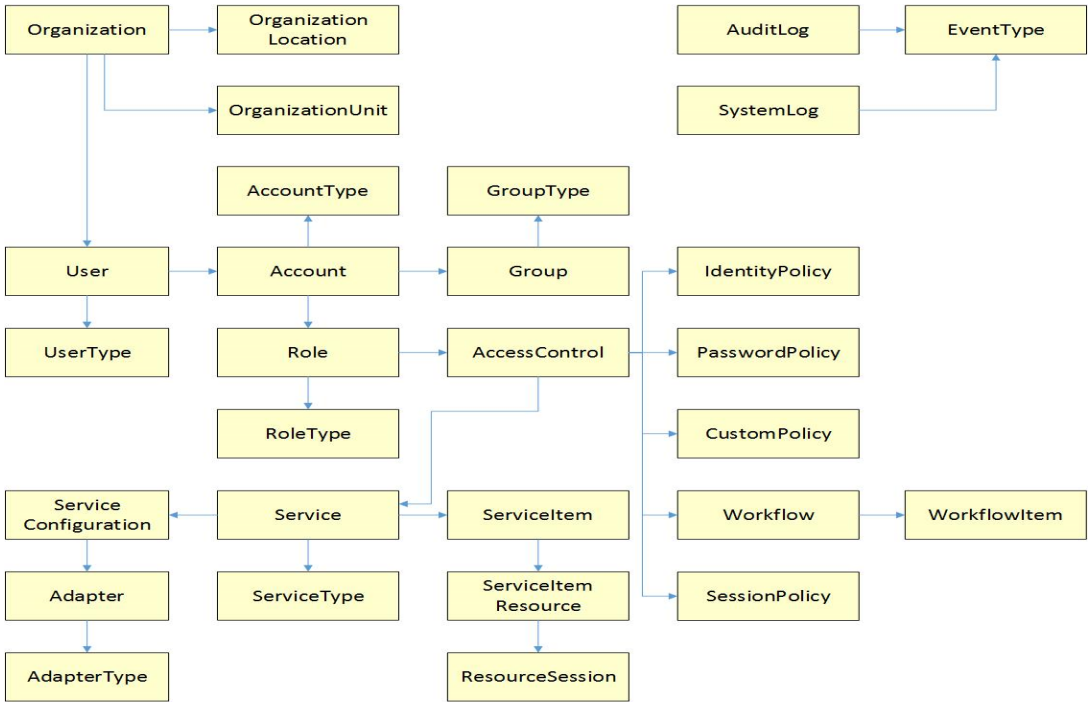

Figure-4. The common data model of security software framework of global distributed computing systems.

Table-2 presents the entities description of common data model of security software framework of global distributed computing systems.

Table-2. Entities description of common data model of security software framework.

Entity name	Entity description
Organization	The entity describes the virtual organization.
OrganizationLocation	The entity describes the location of the virtual organization.
OrganizationUnit	The entity describes the division of the virtual organization.
User	The entity describes the user profile.
UserType	The entity describes the type of the user.
Account	The entity describes the user account.
AccountType	The entity describes the type of the account.
Role	The entity describes the user role.
RoleType	The entity describes the type of the user role.
Group	The entity describes the user group.
GroupType	The entity describes the type of the user group.
AccessControl	The entity describes the access levels of users to the distributed resources.
IdentityPolicy	The entity describes the user identity policy.
PasswordPolicy	The entity describes the user password policy.
CustomPolicy	The entity describes the user custom policy.
SessionPolicy	The entity describes the session management policy.
Workflow	The entity describes the workflow of control access policy to the distributed resources.
WorkflowItem	The entity describes the workflow item of control access policy to the distributed resources.
Service	The entity describes the distributed computing system (exposed service).
ServiceType	The entity describes the type of distributed computing system.
ServiceItem	The entity describes the item of distributed computing system.
ServiceItemResource	The entity describes the resource of distributed computing system.
ResourceSession	The entity describes the session states of the distributed resources.
ServiceConfiguration	The entity describes the configuration of distributed computing system.
Adapter	The entity describes the integration adapter, which provides access to services of distributed computing system.
AdapterType	The entity describes the integration adapter type.

For accessing to the data models will be used the following design patterns [21. 22]:

- Template Row Data Gateway provides a gateway to a single record in the data source.
- Template Active Record provides storage of business logic for accessing the object of data entity.
- Template Table Data Gateway provides a gateway between the data in the application and the database. Gateway object table contains all SQL queries to access a single table or view: fetch, update, insert, delete (CRUD). The rest of the code to interact with the database refers to the methods of the object lock.
- Template Data Mapper provides a programming layer that separates the object and the database. Its duty is to transfer data between them and isolate them from each other. When using the Data Mapper objects do not require knowledge on the existence of the database.
- Template Identity Map provides a single object for downloading, storing data about an object in the matching map.

- Template Association Table Mapping ensures that the correspondence in the form of tables with foreign keys to the tables, which are incorporated by this correspondence.
- Template Domain Model provides a domain object model, combining data and behavior.
- Template Data Transfer Object implements the object that sends data between processes to reduce the number of method calls.

3. RESULTS AND DISCUSSIONS

Based on proposed architectural approaches a security software framework of global distributed computing systems was developed, which consists of the following subsystems and its core features.

3.1 Access control

Access control provides managing of user access to distributed information systems and resources, including the functions of the user registration, user account management, the delegation of user rights.

The function of user registration includes the following features:

- Setting up registration templates for different categories of users and organizations.
- Setting up mechanisms to activate the user account.
- Protection from automatic registration of users.
- The function of user account management includes the following features:
- Blocking of the accounts with the possibility of automatic release within the specified time or event.
- Completing password management policies (checking password strength and its compliance with set rules, restrictions).
- Automated granting users a predefined set of rights in information services related to their official duties, based on configurable business roles.
- Automatic change set of user rights when transferring to another position.
- Role-based access control to the functions of application services within a virtual organization.
- Ability to review user access authority at regular intervals.

The function of delegation user rights includes the following features:

- Setting up a policy delegating of users, processes, devices.
- Single registry service for delegating rights.
- Rule-based scenarios using scripting languages.
- Sharing policies in the XML-format.
- Delegation based on using proxy certificates and temporary tokens.

3.2 Identity management

Identity Management provides authentication mechanisms are built on open standards, and the provision of appropriate rights to access the resources.

The function of authentication includes the following methods:

- Authentication based on reusable passwords.
- Authentication based on one-time passwords.
- PIN-code authentication.
- Strong authentication based on symmetric algorithm.
- Strong authentication based on asymmetric algorithms.

The function of SSO includes the following implementations: SSO Kerberos, SSO LDAP, SSO SAML, OpenID, and OAuth. Identity Management supports different categories of users, including end users, application services, system services, information resources, and data storage.

3.3 User session management

User Session Management provides mechanisms for managing user data in the distributed IT-environment, and it implements the following features:

- Managing session states and connections.
- Managing user data and resource data.
- Different schemes of caching user data.
- Store session states at the service or data storage.
- Support clusters and load balancing modes.
- Monitoring the status of user sessions.

3.4 Monitoring and auditing

Monitoring and auditing provides a mechanism for collecting and analyzing information on activity of users and IT-infrastructure conditions.

Registration of resource utilization includes the functions of collecting and recording data about using system and user resources, and has the following features:

- Collection and accounting of system resources (Hardware, OS, RDBMS).
- Collection and accounting of user resources (Applications, Services, Storages).
- Collection and accounting of resources for organizations and domains (Applications, Services, Storages).
- Reports of resource usage for defined period of time (day, week, month, year).
- Support different file formats for reports including XML, DOC, and PDF.
- Send reports to emails or cell phones.

Monitoring the availability of information resources provides a mechanism for analyzing and controlling data about the current state of system and user resources monitoring the availability has the following features:

- Providing a user interface for monitoring and visualization of resources to various categories of administrators.
- Providing configurable schemes for polling data of services, systems, virtual organizations, and users.
- Periodic assessment of system availability, including methods of stress testing.
- Sending alarm messages via e-mail and cell phones.
- Support working with archival data.

3.5 Management of data sources

Management of data sources provides secure integration mechanisms of interaction with the external distributed information systems and data sources.

Access to data sources provides a mechanism for secure access to data storages, including support for the following features:

- Databases, including Microsoft SQL Server, Oracle Database, MySQL, PostgreSQL.
- Repositories of unstructured data such as Apache Hadoop.
- Data source providers such as JDBC, OData.

Access to distributed information systems provides a mechanism to secure access to the services of information systems using the following integration adapters: FTP Adapter, Web Services SOAP Adapter, Web Services REST Adapter, WSRF Adapter, MQ Adapter.

Access to enterprise systems using specified security context including such enterprise solutions as Oracle OEBS, Microsoft Dynamics, SAP mySAP, 1C.

3.6 API services

API services provide the implementation of the application programming interfaces for access to the security framework infrastructure from the external applications. API services expose the following functionality:

- Authentication.
- Authorization.
- User registration.
- User account management.
- User rights management.
- User session management.
- Access to the user data and external resources.

API services provide interfaces for the following programming languages: Java, C++, Python as well as web services proxies for SOAP and REST standards.

3.7 Deployment and performance testing of security software framework

Table-3 presents the implementation of security software framework of global distributed computing systems tested on the environment [23].

Table-3. Test environment for testing security software framework of global distributed computing systems.

Server name	OS	RDBMS	Options
Application server	RHEL	PostgreSQL	Java EE, GlassFish AS
Transport server	RHEL	-	Java EE, GlassFish AS, Synapse ESB
Data processing server	W2K12	PostgreSQL	Java EE, GlassFish AS
Database server	RHEL	PostgreSQL	-
Web server	RHEL	-	Java EE, GlassFish AS

Table 4 presents the composition of server components of security infrastructure.

Table-4. The composition of server components of security infrastructure.

Options	Minimum option (basic level)			Optimum option (extended level)					Full option (enterprise level)						
Total amount of servers	1	2	3	1	2	3	4	5	1	2	3	4	5	6	7
Application servers	X			X	X				X	X					
Transport servers	X			X	X						X				
Data processing server		X				X	X					X	X		
Database servers	X			X	X						X				
Web server			X					X						X	X

Depending on the load each subsystem has the ability to scale according to the performance recommendations to the appropriate server-side. It is also possible to load redistribution and component duplication on different servers.

Minimum option (basic level) is for business operations with low frequency and small number of users (not recommended for loads over 1000 concurrent user sessions).

Optimum option (extended level) is for regular business operations (not recommended for loads over 3000 user sessions).

The full option (enterprise level) is for intensive use with a significant number of participants (not recommended for loads over 10,000 user sessions).

4. CONCLUSIONS

The result of this work is research of architectural approaches for building security software framework of global distributed computing systems. The design includes application architecture, describing the major functional subsystems, system architecture, describing the software and hardware that implement security infrastructure, data architecture, describing the main business entities and relational data model.

The main distinguishing characteristics of these architectural approaches for building information security infrastructure of distributed information systems are the following:

- The security infrastructure built on a modular principle, provides flexible configuration and expansion of the functional and technical capabilities.
- The security infrastructure uses a single centralized distributed database that contains all user data as well as a description of the business logic to process the given data.
- The security infrastructure supports distributed access to data and distributed business transaction that provides operating modes in the active cluster, grid and cloud systems.
- The security infrastructure was implemented based on 3-tier architecture that includes a data layer, business logic, user interface level.
- The developed modules used a technique of designing distributed and scalable applications (SOA, MDA, ESB, MVC), as well as open standards (XML, SOAP, WSDL, REST, Web Services).
- The security infrastructure has flexible settings (user interface, business logic, data forms) and is open, which allows for integration cooperation with a wide range of information systems.

ACKNOWLEDGEMENTS

The Ministry of Education and Science of the Russian Federation supported the work (Agreement #14.576.21.0078, unique identifier agreement RFMEFI57614X0078).

REFERENCES

[1] Foster I. and Kesselman C. 2004. The Grid 2: Blueprint for a New Computing Infrastructure (2nd ed.). San Francisco, CA: Morgan Kaufmann Publishers Inc.

[2] Foster I. 2001. The Anatomy of the Grid. Enabling Scalable Virtual Organizations. Supercomputer Applications. 15(3): 200-222.

[3] Foster I., Kesselman C., Tsudik G. and Tuecke S. 1998. Security architecture for computational grids, Proceedings of ACM conference on Computer and communications security (pp. 83-91). Washington, DC.

[4] Ptitsyn P. and Radko D. 2015. An analysis of technologies for building information security infrastructure of global distributed computing systems, Journal of Theoretical and Applied Information Technology. 82 (1): 1992-8645.

[5] Ian D. A. 2010. A security framework for distributed batch computing. Madison, WI: University of Wisconsin-Madison.

[6] Hee-Khiang N., Quoc-Thuan H., Bu-Sung L., Dudy L. and Yew-Soon O. 2005. Nanyang Campus Inter-organizational Grid Monitoring System, Proceedings of Grid Asia Workshop on Grid Computing and Applications, (pp. 118-127). Singapore: Nanyang Technological University.

[7] Leshak A. 2008. Analysis and design of information systems using UML 2.0. Moscow, Russia: Williams.

[8] Soloviev I. and Mayorov. A. 2009. Design of Information Systems. Moscow, Russia: Academic Project.

[9] Tulloch M. 2003. Microsoft Encyclopedia of Security. Redmond, WA: Microsoft Press.

[10] Purd J. 2007. Data Grids and Service-Oriented Architecture. Redwood Shores, CA: Oracle Corporation.

[11] GGF OGSA Security Workgroup. 2002. OGSA Security Roadmap. Chicago, IL: Argonne National Laboratory.

[12] Thomas E. 2005. Service-Oriented Architecture (SOA): Concepts, Technology, and Design. New York: Prentice Hall.

[13] Nicolai M. 2007. SOA in Practice. The Art of Distributed System Design. Sebastopol, CA: O'Reilly Media.

[14] Butt A. 2003. Grid-computing portals and security issues. New York: Academic Press.

[15] Neha M. 2014. Security issues in grid computing. International Journal on Computational Sciences and Applications. 4(1).

[16] Fowler M. 2012. Patterns of Enterprise Application Architecture. London, UK: Pearson.

[17] Hohmann L. 2003. Beyond Software Architecture: Creating and Sustaining Winning Solutions. Indianapolis, IN: Addison-Wesley Professional.

[18] Gupta A. 2013. Java EE 7 Essentials. Indianapolis. Sebastopol, CA: O'Reilly Media.

[19] Andreson J. and Andreson P. 2002. Enterprise JavaBeans Component Architecture: Designing and Coding Enterprise Applications. New York: Prentice Hall.

[20] Oppel A. 2010. Data Modeling: A Beginner's Guide. New York: McGraw-Hill Education.

[21] Hophe J. and Woolf B. 2003. Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Indianapolis, IN: Addison-Wesley Professional.

[22] Blaha M. 2010. Patterns of Data Modeling. Boca Raton, FL: CRC Press.

[23] Klemm B. and Smith W. 2013. Application Performance Testing: A Universal Performance Testing Methodology. Seattle, WA: Bernhard Klemm.