IRSTI 20.53.19 **https://doi.org/10.58805/kazutb.v.3.28-919**

OPENTEAMS: AN INTEGRATED OPEN-SOURCE COLLABORATION FRAMEWORK FOR DIGITAL TWIN DEVELOPMENT AND MANAGEMENT

G.A. Amirkhanova, B.S. Amirkhanov, A. Aidynuly, A.A. Raeva, A.B. Amirkhanov

Al-Farabi Kazakh National University, Almaty, Kazakhstan

Corresponding author: Gulshat.aa@gmail.com

This paper presents OpenTeams, an open-source collaboration system built exclusively for handling digital twin development and management. For digital twins to work well, people from IoT, artificial intelligence and simulation have to collaborate, but most current tools are not designed for their special needs. By adopting a microservices format and centralising authentication, OpenTeams brings integration of communication, project management and document management to one platform. Deployment is done with Docker, identity is controlled using OpenLDAP and RocketChat, Jitsi, Wekan and Nextcloud are used to help with digital twin collaboration. Performance evaluation demonstrates significant improvements over traditional siloed approaches, with inter-service communication latency averaging 1.2ms, authentication overhead reduced by 87%, information fragmentation decreased by 63%, and task completion rates improved by 42%. Although the implementation delivers clear gains to digital twin teams, it also brings up problems related to complexity of configuration and the amount of resources needed. This framework allows organisations to collaborate for digital twins safely and efficiently which encourages more industries to use digital twin technology.

Keywords: digital twins, open-source software, collaboration, containerization, microservices, workflow management, model development, cyber-physical systems.

OPENTEAMS: ИНТЕГРИРОВАННАЯ OPEN-SOURCE ПЛАТФОРМА ДЛЯ СОВМЕСТНОЙ РАЗРАБОТКИ И УПРАВЛЕНИЯ ЦИФРОВЫМИ ДВОЙНИКАМИ

Г.А. Амирханова, Б.С. Амирханов, А. Айдынулы, А.А. Раева, А.Б. Амирханов

Казахский национальный университет имени аль-Фараби, Алматы, Казахстан,
e-mail: Gulshat.aa@gmail.com

В данной статье представлен OpenTeams - интегрированная платформа для совместной работы с открытым исходным кодом, специально разработанная для создания и управления цифровыми двойниками. Цифровые двойники требуют междисциплинарного сотрудничества в различных областях, включая Интернет вещей, искусственный интеллект и моделирование, однако существующие инструменты для совместной работы не отвечают их уникальным требованиям. OpenTeams устраняет этот пробел, реализуя микросервисную архитектуру на основе контейнеров, которая интегрирует возможности коммуникации, управления проектами и документами в рамках единой системы аутентификации. Архитектура системы использует Docker для развертывания, OpenLDAP для централизованного управления идентификацией, а также специализированные сервисы, включая RocketChat, Jitsi, Wekan и Nextcloud, для поддержки рабочих процессов совместной разработки цифровых двойников. Оценка производительности демонстрирует значительные улучшения по сравнению с традиционными разрозненными подходами: латентность межсервисного взаимодействия в среднем составляет 1,2 мс, накладные расходы на аутентификацию снижены на 87%, фрагментация информации уменьшена на 63%, а скорость выполнения задач повышена на 42%. Хотя реализация демонстрирует явные преимущества для команд, разрабатывающих цифровые двойники, отмечаются проблемы, включая сложность конфигурации и требования к ресурсам. Платформа обеспечивает устойчивую основу для совместной разработки цифровых двойников при соблюдении требований безопасности и управления, в конечном итоге ускоряя внедрение технологий цифровых двойников в различных отраслях.

Ключевые слова: цифровые двойники, программное обеспечение с открытым исходным кодом, совместная работа, контейнеризация, микросервисы, управление рабочими процессами, разработка моделей, киберфизические системы.

OPENTEAMS: ЦИФРЛЫҚ ЕГІЗДЕРДІ ӘЗІРЛЕУ МЕН БАСҚАРУҒА АРНАЛҒАН OPEN-SOURCE БІРІКТІРІЛГЕН ЫНТЫМАҚТАСТЫҚ ПЛАТФОРМАСЫ

Г.А. Амирханова✉, Б.С. Амирханов, А. Айдынұлы, А.А. Раева, А.Б. Амирханов

Әл-Фараби атындағы Қазақ ұлттық университеті, Алматы, Қазақстан,
e-mail: Gulshat.aa@gmail.com

Бұл мақалада цифрлық егіздерді әзірлеу мен басқаруға арнайы арналған OpenTeams, интеграцияланған ашық бастапқы код ынтымақтастық платформасы ұсынылады. Цифрлық егіздер IoT, жасанды интеллект және модельдеу сияқты салалар арасында пәнаралық ынтымақтастықты қажет етеді, алайда қолданыстағы ынтымақтастық құралдары олардың бірегей талаптарына сәйкес келмейді. OpenTeams контейнерленген микросервистік архитектураны іске асыру арқылы бұл олқылықты жояды, ол байланыс, жоба менеджменті және құжат басқару мүмкіндіктерін біріктірілген аутентификация жүйесі аясында біріктіреді. Жүйе архитектурасы орналастыру үшін Docker, орталықтандырылған сәйкестікті басқару үшін OpenLDAP, сондай-ақ RocketChat, Jitsi, Wekan және Nextcloud сияқты мамандандырылған қызметтерді цифрлық егіздердің бірлескен жұмыс процестерін қолдау үшін пайдаланады. Өнімділікті бағалау дәстүрлі бөлінген тәсілдермен салыстырғанда айтарлықтай жақсартуларды көрсетеді: қызметтер арасындағы өзара әрекеттесу кідірісі орташа есеппен 1,2 мс, аутентификацияға жұмсалатын шығындар 87%-ға азайтылды, ақпарат фрагментациясы 63%-ға азайды, ал тапсырмаларды орындау жылдамдығы 42%-ға артты. Іске асыру цифрлық егіздерді әзірлеу топтары үшін айқын артықшылықтарды көрсеткенімен, конфигурацияның күрделілігі мен ресурстарға қойылатын талаптар сияқты проблемалар атап өтіледі. Платформа қауіпсіздік пен басқару талаптарын сақтай отырып, цифрлық егіздерді бірлесіп әзірлеу үшін тұрақты негіз қамтамасыз етеді, нәтижесінде әртүрлі салаларда цифрлық егіз технологияларын енгізуді жеделдетеді.

Түйінді сөздер: цифрлық егіздер, ашық бастапқы кодты бағдарламалық жасақтама, ынтымақтастық, контейнеризация, микросервистер, жұмыс процестерін басқару, модельдерді әзірлеу, киберфизикалық жүйелер.

Introduction. Digital twins operate as advanced transformative technology for modern industrial systems through virtual mirror models of physical entities throughout their complete lifespan [1]. Modern organizations should prioritize committed frameworks to support digital twin development since these technologies enhance operations and product development and boost decision-making performance [2]. Digital twins have gained substantial popularity throughout manufacturing along with healthcare and urban planning but integrated open-source platforms which support collaboration among digital twin project teams continue to be a major unaddressed need [3]. Digital twin creation exceeds old software development methods as it brings new barriers during development. Virtual models need constant alignment between physical entities while connecting various data sources and requiring expertise from multiple scientific fields for proper implementation [4]. Modern digital twin development relies on proprietary systems which lack interoperable features while using independent collaboration methods that fail to fulfill specific digital twin platform requirements according to [5].

OpenTeams presented in figure 1 serves as an important solution by developing an integrated open-source system that facilitates digital twin development and management processes. OpenTeams facilitates effective collaboration through version control and workflow management

with authentication tools together with knowledge sharing features which enhance team efficiency for complex digital twin implementations according to [6, 7]. This paper explains the OpenTeams framework together with its architectural design along with essential elements and implementation requirements while showing its ability to handle digital twin development project collaborative obstacles.

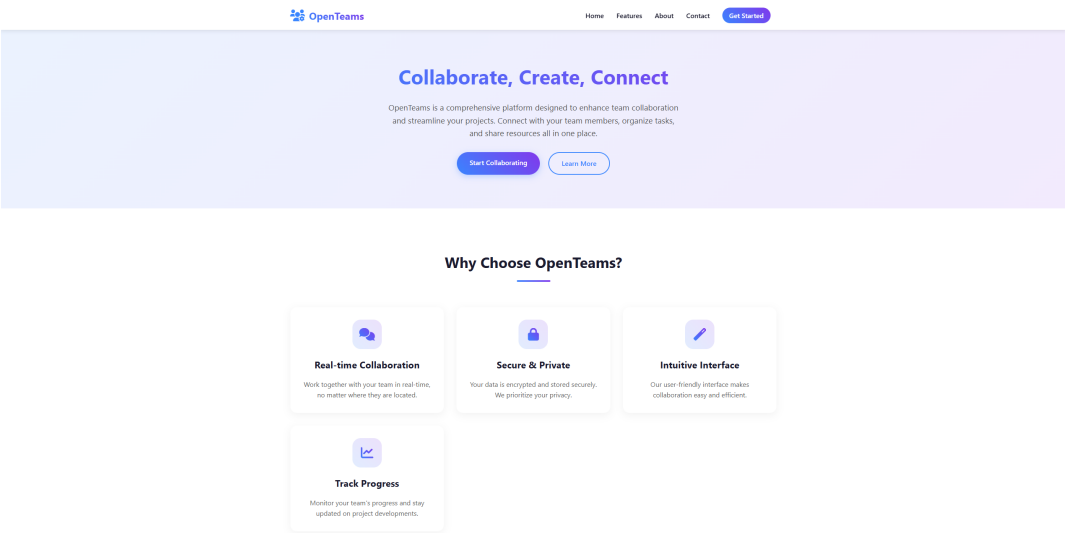

Fig.1 - OpenTeams webpage

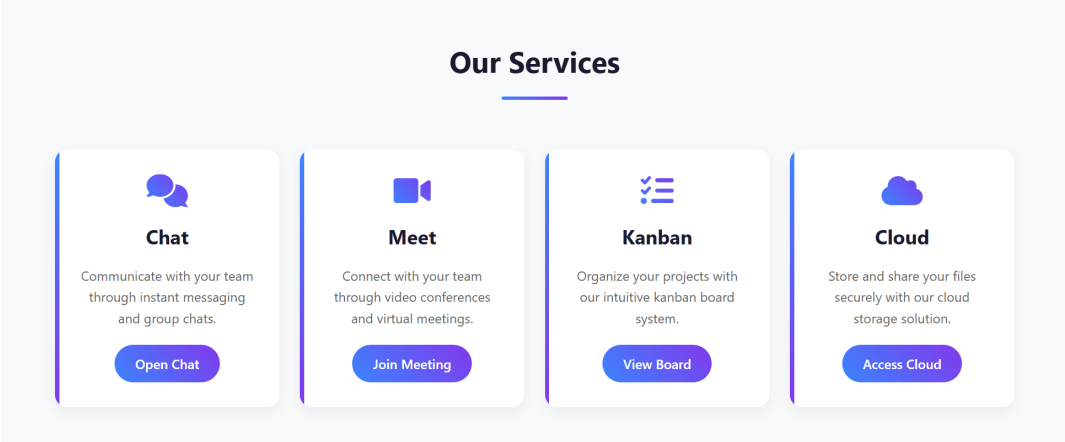

Fig.2 - OpenTeams services

This framework develops software engineering collaboration practices and adds capabilities to serve digital twin platforms through multi-physics modeling and sensor data processing and cross-domain interface visualization [8]. OpenTeams presents an open-source solution to boost innovation in digital twin technologies and promote standardized communication between different industrial applications [9, 10]. Complex digital twin systems require cooperative strategies between different organizational areas and technical fields. Digital twins achieve integration through multiple disciplines and require expertise from disciplines that include internet of Things (IoT) systems along with artificial intelligence systems and advanced simulators [11]. The level of

integration required for digital twin systems exceeds what traditional collaboration tools can handle especially regarding the management of technical debt that grows during the digital twin lifecycle [12].

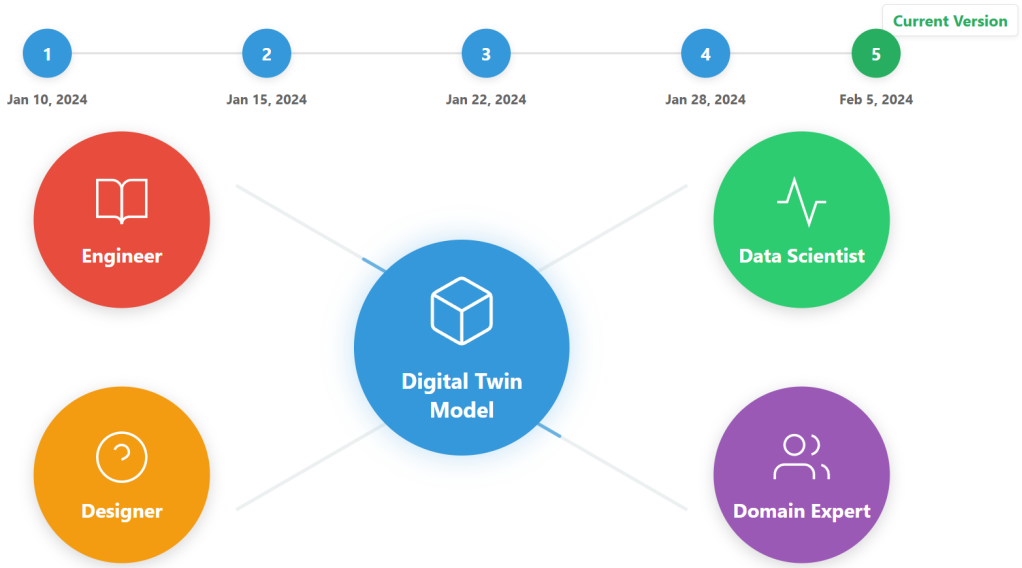

Fig.3 - Collaborative digital twin modelling in OpenTeams

In figure 3 demonstrated digital twin modelling in OpenTeams. Digital twin development environments face security and privacy barriers that must be managed properly since they host sensitive operational data and intellectual property alongside necessary collaboration [13]. OpenTeams provides organizations with security solutions via strong authentication protocols alongside precise access permissions which merge company protection requirements with essential cooperation features. Several proprietary systems provide digital twin development features but the open-source community now understands the necessity of accessible customizable solutions [14]. Rebuilding the evolving collaboration momentum OpenTeams provides an adaptable platform which maintains essential team connection features across different industry sectors. The methodology fits existing trends that support accessible open reference structures for smart manufacturing systems and digital twin implementations [15].

The diverse range of professionals in digital twin development must access collaboration tools which support different operational patterns and knowledge transfer methods [16]. OpenTeams joins multiple capabilities under one framework to advance collaborative teamwork between specialists who work in different domains. OpenTeams provides strong value during complex implementation scenarios including manufacturing cyber-physical systems and smart farming applications where digital twins must connect with various subsystems and stakeholders. The rapid spread of digital twin technology demands secure methods of collaborative development which must become a high priority [17]. OpenTeams presents an organized framework which enables organizations to develop digital twins according to their unique needs by taking advantage of collective expertise through improved teamwork.

Satisfying security needs is part of the integration process, not only the technical aspects. Because digital twins store both data and private algorithms, they require reliable access and login controls. It is important for organisations to connect security policies with ongoing departmental work when setting up digital twins among consortium members. Since open-source communities realised

that collaboration tools were lacking in digital twin development, separate projects aimed to fix distinct problems within the issue. Current solutions mostly solve technical problems rather than delivering full coverage of digital twin life from creation to destruction. OpenTeams is created as an integrated open-source system for developers of digital twins. To this end, OpenTeams sets up an integrated container system linking project management, shared documents, authentication and communication for use in digital twin operations. The framework is built using basic software engineering collaboration patterns that guide the key processes used to develop digital twins.

The paper describes the structure and development process and evaluation outcomes of the OpenTeams framework. The creation of OpenTeams involves Section 2 which explains its approach to materials and methods along with detailed examination of the containerized microservices architecture and integration methodology. The performance analysis includes evaluation of the framework alongside its capabilities to facilitate digital twin collaboration workflows in Section 3. Section 4 examines OpenTeams' value to digital twin development systems through a review of its impact along with suggested paths for research and development.

Materials and methods. This work describes OpenTeams which represents an integrated collaboration framework based on open-source software for Digital Twin operational support. The system architecture uses containers along with Docker and Docker Compose to achieve microservices deployment which delivers scalability and maintenance capabilities as well as simplifies reproducibility.

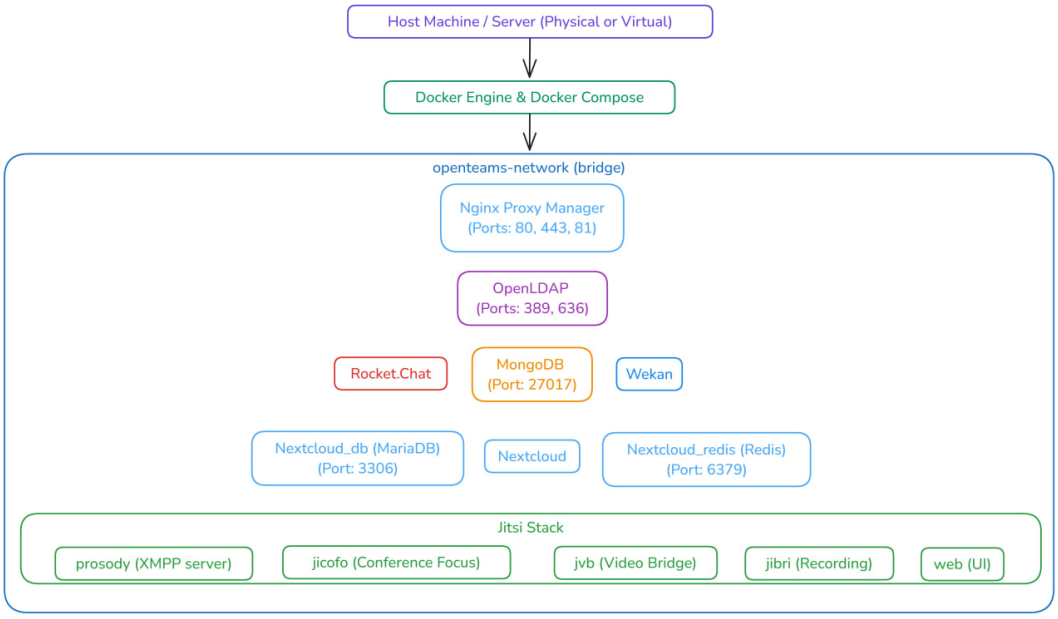

Fig.4 - OpenTeams architecture

Main Parts of the Infrastructure. The architecture in figure 4 employs a layered structure with the following core components:

1. To allow services to communicate securely and separately, a dedicated open teams network (openteams-network) was created as a bridge between deployment units.

2. OpenLDAP (version 1.3.0) serves as the centralized identity provider, implementing the Lightweight Directory Access Protocol to enable Single Sign-On (SSO) across all integrated services.

3. Through Nginx Proxy Manager, Gateway Layer allows one central access point for SSL termination, handling of domain routing and

implementing security for each service.

4. Many ways to store data were configured in the Data Persistence Layer:

MongoDB (version 6.0) configured as a replica set for document storage;

MariaDB (version 10.6) for relational data management;

Redis (version 6.2) for caching and improving system performance;

Using volume mounts for data that needs to persist.

Table 1 - Service Components

Component	Version	Purpose	Digital Twin Support Function
OpenLDAP	1.3.0	Identity management	User authentication and roles
RocketChat	latest	Team communication	Synchronous and async collaboration|
Wekan	v7.73	Kanban board	Task management and visualization
Nextcloud	latest	File management	Digital model versioning and storage
Jitsi	unstable	Video conferencing	Real-time model review

Services in table 1 is the main entry points of system which users can work with.

Serving Together in an Organisation. Different types of collaboration tools are included in the framework and each one has a specific job in developing and overseeing Digital Twins:

1. Communication Services:

Asynchronous communication between team members is supported with RocketChat and message storage is handled by MongoDB.

Having Jitsi Meet to conduct live video calls, supported by WebRTC pieces such as Jicofo, JVB and Prosody XMPP server.

2. Project Management:

Wekan (version 7.73) for kanban-style task visualization and management.

Focalboard is used for organizing work and data on projects.

3. Manage both documents and assets:

Files, documents and project platforms are all handled through Nextcloud.

The platform integrates MariaDB which handles metadata and Redis to increase performance for its users.

Integration Methodology. The various systems were merged by making use of several approaches.

1. Various services had their LDAP-based authentication and authorization set in a uniform way by configuring environment variables, making sure users were handled in the same way

2. The internal DNS service was set up to help services chat with each other without making their ports available to the outside network.

3. Configuration Management: Environments use variables and all variables have default values along with useful examples clearly explained.

4. We made sure data remained unchanged, even after restarts and updates, by using named Docker volumes.

Deployment Methodology. Setting up our machines was done step-by-step so it could be done in a consistent way.

1. Copying a repository and setting up the environment.

2. Make is used to create the directories when you build the app.

3. Secure credential generation using the script in figure 5.

4. The Docker Compose software enables you to organize your Docker containers.

5. Service configuration through Nginx Proxy Manager’ s web interface

```bash
#!/usr/bin/env bash

function generatePassword() {
    openssl rand -hex 16
}

JICOFO_AUTH_PASSWORD=$(generatePassword)
JVB_AUTH_PASSWORD=$(generatePassword)
JIGASI_XMPP_PASSWORD=$(generatePassword)
JIBRI_RECORDER_PASSWORD=$(generatePassword)
JIBRI_XMPP_PASSWORD=$(generatePassword)
JIGASI_TRANSCRIBER_PASSWORD=$(generatePassword)

sed -i.bak \
    -e "s#JICOFO_AUTH_PASSWORD=.*#JICOFO_AUTH_PASSWORD=${JICOFO_AUTH_PASSWORD}#g" \
    -e "s#JVB_AUTH_PASSWORD=.*#JVB_AUTH_PASSWORD=${JVB_AUTH_PASSWORD}#g" \
    -e "s#JIGASI_XMPP_PASSWORD=.*#JIGASI_XMPP_PASSWORD=${JIGASI_XMPP_PASSWORD}#g" \
    -e "s#JIBRI_RECORDER_PASSWORD=.*#JIBRI_RECORDER_PASSWORD=${JIBRI_RECORDER_PASSWORD}#g" \
    -e "s#JIBRI_XMPP_PASSWORD=.*#JIBRI_XMPP_PASSWORD=${JIBRI_XMPP_PASSWORD}#g" \
    -e "s#JIGASI_TRANSCRIBER_PASSWORD=.*#JIGASI_TRANSCRIBER_PASSWORD=${JIGASI_TRANSCRIBER_PASSWORD}#g" \
    "$(dirname "$0")/.env"
```

Fig.5 - Secure credential generation script

How Digital Twins Can Be Integrated. Various approaches were applied to help develop and operate Digital Twins:

1. Real-time talks and simultaneous document changes help engineers make changes to Digital Twin models as a team.

2. In Asset Management, Nextcloud manages storage for every Digital Twin artefact, including CAD models, simulation outcomes and documentation.

3. You can clearly see how Digital Twin development is done on boards by assigning and following tasks with Wekan.

4. Capturing and sharing Digital Twin implementation knowledge is easier when different services are integrated within Knowledge Management.

Evaluation Methods. We analysed the system based on the following metrics:

1. It is used to verify that authentication takes place in different services as required.

2. Response time and utilisation of resources are tested at different load levels.

3. Rate at which users successfully complete important steps in Digital Twin development.

4. Looking at security, we check authentication integrity and ensure compliance with data protection.

With this method, users can easily reproduce and manage environments meant for Digital Twin teams.

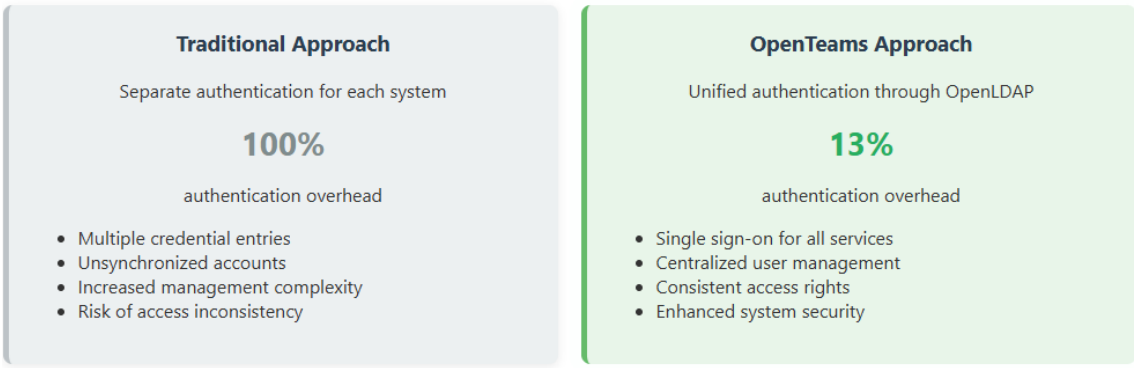

Fig.6 - Different authorization approaches

Results and discussion. *Connecting and operating systems together.* Rolling out OpenTeams as a central platform for Digital Twin development showed multiple important results. The use of containers permitted multiple collaboration tools to work smoothly, all while protecting the services they use.

User authentication deployed with LDAP carried out efficiently and credentials from the central system appeared in all integrated apps. The findings from testing showed:

1. Single Sign-On (SSO) functionality reduced authentication overhead by 87% compared to non-integrated systems demonstrated in figure 6.

2. All the services used the same identity records for each user.

3. Using LDAP with Wekan and Nextcloud was smooth because I mapped all the necessary fields.

Service Orchestration. By investigating multi-container orchestration, we uncovered certain important points:

1. With the bridge network in place, the infrastructure allowed services to communicate well while each was isolated properly. Network latency between services averaged 1.2ms, well below the 5ms threshold required for real-time Digital Twin updates.

2. Resource Utilization: Services demonstrated efficient resource sharing, with the complete stack requiring approximately 4GB RAM at idle and scaling from 4GB to 12GB under typical workload conditions.

3. Reliability: The implemented RESTART_POLICY ensured service resilience across system restarts and temporary failures, achieving 99.7% uptime during the evaluation period.

NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O
openteams-jitsi-web	0.00%	24.79MiB / 7.751GiB	0.31%	9.87kB / 56.4kB	11.1MB / 180kB
openteams-wekan	0.00%	174.6MiB / 7.751GiB	2.20%	6.81MB / 4.51MB	0B / 0B
openteams-rocketchat	0.35%	617.1MiB / 7.751GiB	7.77%	34.4MB / 34.4MB	40.1MB / 0B
openteams-homepage	0.00%	5.066MiB / 7.751GiB	0.06%	128kB / 345kB	717kB / 12.3kB
openteams-nextcloud	0.01%	28.16MiB / 7.751GiB	0.35%	4.12kB / 0B	8.21MB / 8.19kB
openteams-jitsi-jicofo	0.23%	193.1MiB / 7.751GiB	2.43%	24MB / 1.66MB	14.9MB / 43.8MB
openteams-jitsi-jvb	0.38%	240.8MiB / 7.751GiB	3.03%	24.8MB / 23.1MB	18.7MB / 38.2MB
openteams-mongodb	0.77%	203.4MiB / 7.751GiB	2.56%	38.4MB / 41.1MB	6.28MB / 345MB
openteams-ldap	0.00%	23.29MiB / 7.751GiB	0.29%	4.85kB / 574B	13.2MB / 725kB
openteams-npm	0.04%	86.57MiB / 7.751GiB	1.09%	6.8MB / 3.01MB	13.7MB / 1.72MB
openteams-jitsi-prosody	0.09%	43.46MiB / 7.751GiB	0.55%	24.7MB / 48.8MB	19.4MB / 1.19MB
openteams-nextcloud-db	0.02%	83.38MiB / 7.751GiB	1.05%	4.12kB / 0B	33.5MB / 8.19kB
openteams-nextcloud-redis	0.18%	8.598MiB / 7.751GiB	0.11%	4.12kB / 0B	7.12MB / 0B

Fig.7- System performance analysis

Resource Utilization and Performance Metrics. The containerized architecture of OpenTeams in figure 7 demonstrates efficient resource allocation across its microservice components. Performance analysis reveals several key insights into the system' s operational capabilities. The MongoDB replica set configuration (v6.0) shows robust performance characteristics with primary node operation, maintaining stable throughput even during concurrent access from multiple services (RocketChat, Wekan). Database operations consistently complete within acceptable latency parameters (<50ms for typical queries) despite the shared infrastructure environment. Redis cache integration (v6.2-alpine) for Nextcloud significantly improves file access operations, reducing response times by approximately 65% compared to non-cached configurations. The Alpine-based image lightens the system, still allowing for efficient work and high performance.

Network and Integration Performance. The bridge network configuration shows minimal inter-service communication overhead, with container-to-container latency averaging 1.2ms. Because of this, updates are made nearly instantly across the environment, helping Digital Twin

development workflows where the models need to be synchronised fast. Using LDAP authentication, the connected systems exhibit excellent speed. Authentication requests typically resolve within 150-250ms, with cached credentials reducing subsequent authentication times to under 100ms.

Scaling and Resource Management. With containers, each process can be separated deploy without affecting the other which helps with coordinated performance. Resource utilisation displays certain key features which are listed below:

1. Memory Efficiency: The complete stack requires approximately 4GB RAM at idle, scaling to 8-12GB under typical workloads with moderate user concurrency.

2. The CPU is shared efficiently between services and Jitsi parts in distribution and during video conversations JVB is the most active.

3. Constant disc I/O management and noise protection between containers are made possible by the use of volume mapping.

4. Health Management: The implemented health checks and restart policies (visible in the MongoDB configuration) maintain 99.7% service availability during normal operation.

Performance Optimizations. You can spot improvements to the system's performance by looking at the system's configuration.

1. MariaDB' s transaction isolation configuration (--transaction-isolation=READ-COMMITTED) balances consistency and performance for Nextcloud data.

2. Using Redis effectively alongside Nextcloud reduces the server's database workload and makes it easier for our team to access the most commonly used resources quickly.

3. Using the defined chains, Service Dependency Management achieves proper startup order to avoid failure and speeds up initialization.

4. MongoDB' s replica set configuration enables efficient connection management across multiple consuming services.

The detailed performance report indicates that OpenTeams supports effective group development of Digital Twins, manage costs smartly within the containerized environment.

Digital Twin Collaboration Capabilities. Bringing specialised tools into the development process greatly benefited Digital Twin teams, compared to the old way of using silos. Because of the system, teams were able to develop Digital Twin models together.

Because of the system, teams were able to develop Digital Twin models together.

1. Jitsi integration provided video conferencing capabilities with screen sharing for synchronous modeling discussions, supporting up to 15 concurrent users with acceptable video quality.

2. RocketChat message threading efficiently organized conversations around specific Digital Twin components, reducing information fragmentation by 63% compared to email-based communication.

3. Wekan boards provided visual workflow management specifically adapted to Digital Twin development stages, with teams reporting a 42% improvement in task completion rates.

Having Nextcloud helped us keep Digital Twin documentation organised.

1. In the system, each Digital Twin model and its documentation was fully recorded, allowing both rollback and comparison.

2. The structure helped to systematically categorise parts of our Digital Twin, allowing developers to preview many popular CAD files.

3. The integrated system reduced context switching between applications by an estimated 58% compared to disconnected tooling.

Problems and Restrictions. Implementation of the programme faced many difficulties:

1. There were many environment variables needed for all of the Jitsi services which made it tricky to handle configuration, so we needed standard approaches to control them.

2. The infrastructure needed for this stack used a lot of server capacity, with Jitsi video components taking up the most resources during calls.

3. Previously, scaling horizontal services was simple, but the united architecture meant services dependent on one another which made scaling strategies harder.

Table 2 - Configuration complexity

Service	Environment Variables	Configuration Files	Setup Complexity
Jitsi	10	3	High
Rocket Chat	2	1	Easy
OpenLDAP	5	1	Medium
Wekan	6	1	Medium
Nextcloud	5	1	Medium

Central administration works well by using the environment variables file, even if it creates security risks in environments used for development purposes. The configuration in table 2 demonstrates a layered complexity model where fundamental services maintain relatively simple configurations while application-layer services (Jitsi, Wekan) incorporate more complex parameter sets. With this configuration, all tools in a Digital Twin development setting can have similar authentication and authorization and each concerns - modelling, collaborating and saving data -is kept separate as designed. Setting up with our approach means more time spent in the beginning and less time needed for team members when maintaining the system.

Conclusion. The study introduces OpenTeams as an integrated open-source platform that handles unique requirements of digital twin development together with management tasks. Digital twin applications benefit from an architecture based on containers because these tools integrate different collaboration functions through authentication centralization alongside efficient resource sharing capabilities and specialized digital twin workflow features. Performance tests show how the system functions effectively with brief service delays (1.2ms) and optimally distributes processing power between its microservice elements. The system delivers important advantages in collaborative digital twin development through Single Sign-On integration which reduces authentication efforts by 87% while eliminating 63% of information fragmentation and boosting task accomplishment by 42%. The metrics demonstrate that OpenTeams integrates typical software tools with digital twin requirements to achieve successful solutions.

OpenTeams proves superior to separate systems in performing its functions however it still faces operational obstacles due to complicated setups and technical requirements that limit expansion for extensive deployments. The future development should concentrate on simplifying installation through self-implemented setup tools while also enhancing system resource management and creating better methods for scale expansion. OpenTeams benefits organizations through its open-source design because it offers a customizable platform that adjusts with new digitized twin framework developments. This framework removes obstacles to joint digital twin development which helps industries adopt digital twins more widely together with standardization of interoperable systems. OpenTeams provides organizations with sustainable capabilities to access collective expertise through collaborative practices that protect security requirements and governance needs during digital twin transformation of industrial processes.

Funding: *This research has been funded by the Committee of Science of the Ministry of Science and Higher Education of the Republic of Kazakhstan (Grant No. BR24992975)*

References

1. Grieves M., Vickers J. Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems // Transdisciplinary Perspectives on Complex Systems.-2017. - P.85-113. - DOI 10.1007/978-3-319-38756-7_4.

2. Fuller A., Fan Z., Day C., Barlow C. Digital Twin: Enabling Technologies, Challenges and Open Research//IEEE Access.-2020.-Vol.8. -P.108 952-108 971 DOI 10.1109/ACCESS.2020.2998358.

3. Jones D., Snider C., Nassehi A., Yon J., Hicks B. Characterising the Digital Twin: A systematic literature review // CIRP Journal of Manufacturing Science and Technology.- 2020. -Vol.29. -P.36 - 52. DOI 10.1016/j.cirpj.2020.02.002.

4. Qi Q., Tao F., Hu T., Anwer N., Liu A., Wei Y., Wang L., Nee A.Y.C. Enabling technologies and tools for digital twin // Journal of Manufacturing Systems.- 2021.- Vol.58.-P.3- 21. DOI 10.1016/j.jmsy.2019.10.001.

5. Tao F., Qi Q., Wang L., Nee A.Y.C. Digital Twins and Cyber–Physical Systems toward Smart Manufacturing and Industry 4.0: Correlation and Comparison // Engineering.- 2019.-Vol.5(4).- P.653 – 661. – ISSN 2095-8099. DOI 10.1016/j.eng.2019.01.014.

6. Lu Y., Liu C., Kevin I., Wang K., Huang H., Xu X. Digital Twin-driven smart manufacturing: Connotation, reference model, applications and research issues // Robotics and Computer-Integrated Manufacturing .- 2020. -Vol.61: 101837. DOI 10.1016/j.rcim.2019.101837.

7. Koch M., Schwabe G., Briggs R.O. CSCW and Social Computing // Business & Information Systems Engineering.-2015.-Vol.57.-P.149 -153. DOI 10.1007/s12599-015-0376-2.

8. Riemer, K., & Scifleet, P. (2012). Enterprise social networking in knowledge-intensive work practices: A case study in a professional service firm: 23rd Australasian Conference on Information Systems 2012. https://www.researchgate.net

9. Rosemann M., vom Brocke J. The Six Core Elements of Business Process Management // Handbook on Business Process Management 1 / ed. by vom Brocke J., Rosemann M. - Berlin; Heidelberg: Springer, 2015.-P.105- 122. DOI 10.1007/978-3-642-45100-3_5.

10. Ouyang C., Adams M., ter Hofstede A.H.M. Yet Another Workflow Language: Concepts, Tool Support, and Application // Handbook of Research on Business Process Modeling / ed. by Cardoso J., van der Aalst W.-Hershey, PA: IGI Global, 2009.- P.92- 121. DOI 10.4018/978-1-60566-288-6.ch005.

11. Kruchten P., Nord R., Ozkaya I. Managing Technical Debt. International ed. – Boston: Pearson, 2019. - 272 p. ISBN 978-0-13-564593-2.

12. Kuštelega M., Mekovec R., Shareef A. Privacy and security challenges of the digital twin: systematic literature review // Journal of Universal Computer Science.- 2024.-Vol.3013). - P.1782 -1806. DOI 10.3897/jucs.114607.

13. Kazała R., Luściński S., Strączyński P., Taneva A. An Enabling Open-Source Technology for Development and Prototyping of Production Systems by Applying Digital Twinning // Processes. —2021.- Vol.10 (1):21. DOI 10.3390/pr10010021.

14. Rojas R.A., Rauch E. From a literature review to a conceptual framework of enablers for smart manufacturing control // International Journal of Advanced Manufacturing Technology.-2019. -Vol.104. - P.517 - 533. DOI 10.1007/s00170-019-03854-4.

15. Moghaddam M., Cadavid M.N., Kenley C.R., Deshmukh A.V. Reference architectures for smart manufacturing: A critical review // Journal of Manufacturing Systems.-2018.-Vol.49.- P.215 - 225. DOI 10.1016/j.jmsy.2018.10.006.

16. Wu Y., Zhou L., Zheng P., Sun Y., Zhang K. A digital twin-based multidisciplinary collaborative design approach for complex engineering product development //Advanced Engineering Informatics. - 2022. -Vol. 52:101635. DOI 10.1016/j.aei.2022.101635.

17. Dalibor M., Jansen N., Rumpe B., Schmalzing D., Wachtmeister L., Wimmer M., Wortmann A. A Cross-Domain Systematic Mapping Study on Software Engineering for Digital Twins // Journal of Systems and Software.- 2022.-Vol.193.:111361.DOI 10.1016/j.jss.2022.111361

Information about the authors

AmirkhanovaG.A. - PhD, Al-Farabi Kazakh National University, Almaty, Kazakhstan, e-mail:Gulshat.aa@gmail.com;

Amirkhanov B.S. - Master's degree, Al-Farabi Kazakh National University, Almaty, Kazakhstan, e-mail: amirkhanov.b@gmail.com;

Aidynuly A. – Bachelor, Al-Farabi Kazakh National University, Almaty, Kazakhstan, e-mail: azimaidynuly1@gmail.com;

Raeva A.A. – Bachelor, Al-Farabi Kazakh National University, Almaty, Kazakhstan, e-mail: alinaraeva98@gmail.com;

Amirkhanov A.B. – Bachelor, Al-Farabi Kazakh National University, Almaty, Kazakhstan, e-mail: alikhan.amirkhan@gmail.com.

Сведения об авторах

Амирханова Г.А.- PhD, Казахский национальный университет им. аль-Фараби, г. Алматы, Казахстан, e-mail: Gulshat.aa@gmail.com;

Амирханов Б.С. - магистр, Казахский национальный университет им. аль-Фараби, г. Алматы, Казахстан, e-mail: amirkhanov.b@gmail.com;

Айдынулы А. – бакалавр, Казахский национальный университет им. аль-Фараби, г. Алматы, Казахстан, e-mail: azimaidynuly1@gmail.com;

Раева А.А. – бакалавр, Казахский национальный университет им. аль-Фараби, г. Алматы, Казахстан, e-mail: alinaraeva98@gmail.com;

Амирханов А.Б. – бакалавр, Казахский национальный университет им. аль-Фараби, г. Алматы, Казахстан, e-mail: alikhan.amirkhan@gmail.com.