PROJECT

Multi-tenant cybersecurity operations centre designed as an open cloud service with machine learning elements

Linked to work package KPB1 - Design of the virtualization solution

Linked to output V2: list of defined criteria;

Funded by the European Union
NextGenerationEU

RECOVERY AND RESILIENCE PLAN

GOVERNMENT OFFICE OF THE SLOVAK REPUBLIC

Contents

1 Introduction 3
2 Evaluation methodology 3
2.1 Evaluation criteria for the selection of an open-source solution 3
2.2 Number of evaluation criteria 3
2.3 Conditions for the selection of candidate solutions 4
2.4 Categories of evaluation criteria 4
2.5 Weighting of criteria and their significance 5
2.6 Examples from similar projects and comparisons 6
2.7 Summary 7
3 Definition of comparison criteria for the selection of a cloud platform 7
3.1 Functional criteria 7
3.1.1 K1: Support for IaaS functionality (weight: 5) 7
3.1.2 K2: Modularity and architecture (weight: 4) 7
3.1.3 K3: Support for orchestration and automation (weight: 4) 8
3.2 Technical criteria 8
3.2.1 K4: Performance and scalability (weight: 5) 8
3.2.2 K5: Reliability and stability (weight: 4) 9
3.2.3 K6: Compliance with standards and interoperability (weight: 4) 9
3.3 Operational criteria 10
3.3.1 K7: Complexity of deployment and administration (weight: 3) 10
3.3.2 K8: Support, availability of documentation and community (weight: 4) 10
3.4 Security criteria 11
3.4.1 K9: Authentication and access control (weight: 5) 11
3.4.2 K10: Tenant isolation and data protection (weight: 5) 11
3.4.3 K11: Network/monitoring integration (weight 3) 11
3.5 Evaluation methodology 12
3.6 Design of the evaluation matrix 13
4 Design of the evaluation table for the selection of a cloud platform 13
Conclusion 14
List of sources 14

1 INTRODUCTION

A multi-tenant cybersecurity operations centre, designed as an open cloud service with machine learning elements, requires a careful selection of the technology platform. Since the project includes the design of a comprehensive virtualization platform and its intelligent monitoring, one of the first steps is to define comparison criteria for the selection of a suitable solution. The core tasks of work package KPB1 include precisely the definition of criteria, the analysis of available solutions and the selection of an open-source platform. Carefully set criteria will make it possible to compare the alternatives objectively and select the solution that best meets the project requirements. This eliminates random or emotional decision-making – a quantified evaluation helps to remove subjective guesswork from the decision-making process.

The selection of comparison criteria is critical, because it determines which aspects of the solution are important. In the context of a cloud platform for a security centre, a broad spectrum of factors needs to be taken into account, ranging from the functional side (services offered, integration), through technical parameters (performance, scalability, reliability) and operational aspects (costs, support, administration), to security requirements (security features, compliance with standards). According to technological recommendations and practice, the key criteria when selecting a cloud solution include, for example, security, reliability, performance, scalability and price. Defining these criteria and determining their significance (weights) will ensure that the subsequent evaluation of the available platforms is systematic and defensible.

2 EVALUATION METHODOLOGY

2.1 EVALUATION CRITERIA FOR THE SELECTION OF AN OPEN-SOURCE SOLUTION

When comparing open-source solutions for a private cloud in an academic or research environment, it is important to choose an appropriate number of evaluation criteria, divide them into clear categories and assign them weights according to their importance. The criteria must cover all relevant areas (functional, technical, operational, security, etc.), but at the same time there must not be too many of them, so that their evaluation is realistically manageable within a pilot project. Below we present the recommended range of criteria, a proposed categorization, the approach to weighting and examples from similar projects

2.2 NUMBER OF EVALUATION CRITERIA

For the purposes of an objective comparison, we recommend choosing a **moderate number of criteria**, typically in the range of **approximately 5 to 15**. The literature shows a wide range of approaches – there are very detailed comparisons with a hundred detailed criteria (divided into several categories) [1], while on the other hand some studies made do with a few key criteria (for example ~5–10) covering the most important areas [2]. Approximately **8–12 criteria represent a practical compromise**: they cover the essential functions and properties of a cloud platform, yet a team in an academic pilot project can still evaluate them reliably in the time available. With a smaller number of criteria there is a risk that some aspects of the solutions will remain uncovered, whereas with too large a number (dozens of criteria) the evaluation becomes too complex and time-consuming. It is therefore advisable first to identify all potential factors and

then narrow them down to the most important and non-overlapping criteria (i.e. remove duplicate or marginal factors) [3].

2.3 CONDITIONS FOR THE SELECTION OF CANDIDATE SOLUTIONS

When selecting the candidate cloud platforms for comparison in the next phase of the project, the following basic selection conditions were defined:

- The platform must be **open-source**, available under a free licence and without any obligation to commit to a specific commercial vendor,
- It must be **deployable on-premise**, with the possibility of operation in the environment of an academic institution or an internal data centre.

The aim is to find solutions that provide IaaS (Infrastructure as a Service) functionality, enable a multi-tenant environment and are sufficiently robust for security operations, ideally with open source code or at least with a flexible licensing policy for the academic sphere.

2.4 CATEGORIES OF EVALUATION CRITERIA

To make the comparison clear, it is advisable to divide the criteria into thematic categories. In the context of IaaS platforms for a private cloud (especially with regard to academic/research use and cybersecurity), the following categories are often used:

- **Functional criteria**: They evaluate the **functionality and features** of the platform. This includes, for example, support for various hypervisors and virtualization technologies, the types of supported storage and networks, the available APIs and integrations, support for multi-tenancy (separation of multiple projects/users), orchestration and automation, system scalability and advanced features (e.g. autoscaling, monitoring). The aim is to find out what **services and options** each platform offers and to what extent they meet the needs of the project. These criteria answer the question of **what the platform can provide** in terms of functionality.
- **Technical criteria:** They examine the **technical architecture and requirements** of the solution. These include criteria such as the overall system architecture (modularity, components), compatibility with the existing infrastructure and supported hardware, supported operating systems and databases, performance and efficiency (e.g. how many resources the system consumes, what the limits are on the number of VMs, network speed, etc.), scalability (how well the system grows with increasing load), robustness and reliability (resilience to failures, availability), and interoperability and compatibility with standards. We can also include here information on the technologies on which the platform is built (e.g. the programming language used, the database backend), as this may affect, for example, the knowledge requirements for staff [2]. These criteria evaluate the technical prerequisites of the platform and answer the question of **how the platform works and what demands it places.**
- **Operational (organizational) criteria**: They focus on the **operation, administration and support of the platform** in a real environment. This includes the complexity of installation and configuration, ease of maintenance, quality of documentation and availability of learning resources, size and activity of the user community, available support (community or commercial) and its quality, frequency of updates and of the development cycle, maturity of the project (whether it is a proven technology or a novelty), or licensing requirements (all the solutions mentioned are open-source, but there may be optional paid support services). In an academic environment, where the team of administrators may be small, criteria such as **ease of deployment and**

automation are important (so that the system does not require disproportionate administration effort). We can also include operating costs here – although the software is open-source, the indirect cost must be considered: the necessary hardware, administration time, staff training, and possibly support. Operational criteria thus evaluate the practical usability of the platform in specific conditions (they answer the question of **how demanding it is to deploy and operate the platform**).

- **Security criteria**: In the cybersecurity environment, this is a key category. It assesses the **security features and mechanisms** of the platform. It includes the method of authentication and authorization (support for integration with LDAP/AD, fine-grained access control, support for multi-factor authentication), isolation between tenants and virtual machines (e.g. network isolation, resource separation), support for encryption (of data in storage, of communications), the availability of audit logs and security monitoring, the ability to configure security policies (firewalls, security groups), and compliance with industry standards or best practices. For example, whether the platform makes it possible to fulfil the controls under standards such as ISO/IEC 27001 in the cloud context. In an academic SOC-as-a-Service project, it is also important how the platform supports the integration of security tools (IDS/IPS, SIEM systems) and whether it includes functionalities such as role-based access control (RBAC) for segregation of duties. Security criteria evaluate **how well the platform is protected** and what it offers for securing the managed systems and data.

The categories above can be adapted to specific needs. For example, some methodologies list the **support category** separately (i.e. community, documentation and support as a separate area), or include financial or legislative aspects. For the purposes of a research/pilot project in the academic sphere, however, the four areas listed above (functional, technical, operational, security) typically cover the main criteria. It is important that each category has at least several (e.g. 2–5) specific criteria on the basis of which the given aspect can be evaluated. This produces a balanced **comparison matrix**, which can also be presented in the form of a table – the rows represent the criteria (grouped by category) and the columns the individual platforms being compared.

2.5 WEIGHTING OF CRITERIA AND THEIR SIGNIFICANCE

In **multi-criteria decision-making**, it is advisable to assign **weights** to the individual criteria according to their relative importance for the given project. A simple scale is recommended, and we will also use one, for example **1 to 5**, where 1 means low importance and 5 key, critical importance. **The significance of the weight** lies in the fact that criteria with a higher weight will have a greater impact on the overall evaluation of the platform.

The procedure will be that we **score each platform** on each criterion (for example with 1–5 points or verbally: excellent, average, weak), and then **multiply these values by the weight** of the given criterion. This gives us a **weighted score** for each platform. After summing the weighted scores across all criteria, we obtain the overall ranking of the platforms. Such an approach ensures that a platform that excels in critically important areas achieves a better overall evaluation than one that perhaps has more weaker "nice-to-have" features. Weights thus help to **formalize decision-making** – for example, they make it possible to create a "security-oriented" vs. "performance-oriented" scenario and the like. In an academic environment, a more formal method can also be used, for example the analytic hierarchy process (AHP), which determines the weights consistently by means of pairwise comparison of criteria [4]. In most cases, however, simple weighting on a 1–5 scale is sufficient. **It is important to justify the assigned weights** – e.g. if security has a weight of 5, to state explicitly that this is a project focused on cybersecurity, where data protection is paramount.

2.6 EXAMPLES FROM SIMILAR PROJECTS AND COMPARISONS

There are several **studies and projects** that have dealt with the comparison of open-source cloud platforms, from which inspiration can be drawn for the choice of criteria and the evaluation methodology:

- **Academic comparisons:** An extensive study conducted [5] proposed as many as **100 detailed criteria** for comparing open-source IaaS platforms, dividing them into **6 main categories**: virtualization, storage, network, management, security and support [1]. This work emphasizes the need for a comprehensive view – for example, that in addition to the basic IaaS features, security and the level of support must also be evaluated. Another academic approach used a **narrower selection of ~9 criteria**, among which it included, for example, the system architecture, the programming language used, the supported types of cloud deployments and hypervisors, the user interface, licensing, **robustness**, **interoperability**, **security** and compatibility. Other authors focused only on certain aspects – for example, Steinmetz et al. compared OpenStack and Eucalyptus exclusively in terms of **performance** (instance start-up times, network throughput, etc.) [2]. These examples show that the choice of criteria depends on the focus of the project – whether it is a comprehensive comparison of all features or an in-depth analysis of a specific area.
- **Diploma and research projects:** Academic works often use a combination of quantitative and qualitative criteria. For example, recent research at the University of Twente developed a framework for evaluating IaaS providers using **6 main criteria** (Availability, Performance, Scalability, Reliability, Security, Cost-effectiveness), each of which also had sub-criteria mapped to specific metrics (e.g. in the performance category they monitored CPU performance, disk I/O, network latency, etc.). Although this particular example focused on public clouds, the methodology is instructive – they used weights for criteria and scenarios so that the resulting evaluation reflected the priorities (e.g. a security-sensitive vs. a cost-sensitive scenario) [4]. A similar approach can be applied when selecting a private cloud platform: define the main areas and several metrics for each. Another diploma thesis or pilot project may have, for example, criteria such as *support for containerization* or *integration with specific security tools*, if relevant – what is important is that their number remains manageable and that each criterion can actually be measured or assessed.
- **Industry recommendations:** In industry practice, various checklists and best practices exist for selecting a cloud solution. For example, comparative articles from cloud consulting firms often advise evaluating **scalability vs. robustness vs. support**, etc., and recommend **adapting the category weights** to the needs of the organization. There are also open methodologies such as OSS-PAL or QSOS for evaluating open-source software, which emphasize the necessity of evaluating not only functionality but also **community quality and security**. When selecting IaaS for SOC-as-a-Service, industry standards (e.g. CSA – Cloud Security Alliance checklists) would suggest including **security controls** and **risk management** as criteria. Finally, standards such as ISO/IEC 27017 (security controls for the cloud) or reference architectures (e.g. NIST for the cloud) can serve as a source for creating security and operational criteria, so that the chosen platform meets the required level of security.

2.7 SUMMARY

For our project, we will select approximately 5–15 of the most relevant criteria, divided into meaningful categories (e.g. functional, technical, operational and security). We will assign each criterion a weight according to its significance for the project, so that the resulting comparison reflects your priorities. This approach will make it possible to create a clear **tabular comparison** of the platforms (OpenStack, CloudStack, OpenNebula, Proxmox, or others), in which the strengths and weaknesses of the individual solutions will be clearly visible. The procedure is in line with best practices from the literature – from very detailed academic analyses to practical recommendations – and is **adapted to the context of a research project** focused on building SOC-as-a-Service. Such a comparison should significantly facilitate the **decision** as to which platform best meets the stated requirements.

3 ESTABLISHING COMPARISON CRITERIA FOR THE SELECTION OF A CLOUD PLATFORM

The chapter presents the categories and specific criteria that were proposed for comparing the candidate cloud platforms in the project. For each criterion, a justification of its significance is given, together with an assigned weight on a scale of 1–5 (where 5 denotes the highest importance). The evaluation methodology is then described – the method of scoring the alternatives against the criteria, the use of weights in calculating the overall scores and the decision-making procedure based on the results. The document also includes a proposal for an evaluation matrix that clearly presents the criteria, their weights and the point scores. In the conclusion, we summarize the benefits of the proposed evaluation system and show how its application will enable the selection of a suitable platform for the given project.

3.1 FUNCTIONAL CRITERIA

3.1.1 K1: Support for IaaS functionality (weight: 5)

Justification: The project requires full IaaS functionality – management of virtual machines (VMs), disks, networks, images, snapshots and separate resources. The platform must support the creation and management of various types of instances, allow resources to be allocated according to tenants' needs and at the same time offer a robust API for programmatic access. The ability to provide complete IaaS functionality is key to ensuring the flexible and reliable operation of SOC as a service.

Scoring recommendation:

- 5 points: Support for complete IaaS functions (VMs, networks, disks, snapshots), strong APIs, confirmed references.
- 3 points: Most of the basic IaaS functions available, API with certain limitations.
- 1 point: Limited IaaS functionality, weak or no API support.

3.1.2 K2: Modularity and architecture (weight: 4)

Justification: The modularity of the architecture fundamentally affects the extensibility and scalability of the solution. A platform that is divided into separate components (e.g. Nova – compute, Neutron – networking, Cinder – block storage) allows modules to be deployed, tuned and replaced independently as needed. It is also advantageous if the platform supports a modern microservices architecture, plugins, containerization and orchestration layers.

Higher modularity also reduces the risk of an outage of the entire system in the event of a failure of a single service.

Scoring recommendation:

- **5 points:** High modularity, microservices, independently deployable components, clear plugin integration.
- **3 points:** Modularity is partial; some components are mutually dependent.
- **1 point:** Monolithic architecture, limited options for modifying or replacing components.

3.1.3 K3: Support for orchestration and automation (weight: 4)

Rationale: The platform should support tools such as Heat (OpenStack orchestration), Terraform (infrastructure as code), Ansible (automation) or Kubernetes Operator for the efficient management and deployment of complex infrastructure scenarios. This capability will enable rapid scaling of services, automated roll-out of environments, repeatable testing and recovery from outages. In a SOC environment, where requirements and network topology change frequently, orchestration is essential to ensure flexibility, sustainability and operational efficiency.

Scoring recommendation:

- 5 points: Excellent support (Heat, Terraform, Ansible, Kubernetes), robust and flexible automation.
- 3 points: Basic orchestration available, partial limitations in tool support.
- 1 point: Very limited or no automation, weak integration with popular tools.

3.2 TECHNICAL CRITERIA

3.2.1 K4: Performance and scalability (weight: 5)

Justification: The evaluation considers how efficiently the solution can utilize resources, what limits it has in terms of the number of VMs, containers or tenants, and which automatic scaling mechanisms it supports. Latency and overall performance in data processing are also important, especially in real time for the purposes of security event monitoring. The solution must be able to grow with the number of VMs, tenants and services. Horizontal and vertical scalability are evaluated. Under the conditions of the project, we need a platform that can grow together with the number of SOC service clients and the volume of data processed.

This criterion is therefore assigned a **weight of 5** – maximum importance, since insufficient performance or limited scaling would represent a fundamental risk.

Scoring recommendation:

- 5 points: Excellent scalability, proven ability to handle hundreds of VMs, horizontal and vertical scaling.
- 3 points: Adequate scalability with certain limitations (tens of VMs without a significant drop in performance).
- 1 point: Poor performance, significant scaling limits even with a low number of VMs.

3.2.2 K5: Reliability and stability (weight: 4)

Justification: We assess the maturity of the platform, its stability in production operation and the availability of mechanisms for highly available deployment. The criterion includes the level of testing and debugging of the software and the possibility of deploying components redundantly (HA – High Availability). In a security centre environment, it is important to minimize downtime and ensure continuous availability of the service. When evaluating cloud services, it is recommended to thoroughly examine past performance, customer satisfaction and the handling of unplanned outages. We also compare SLAs (Service Level Agreements) if available – although in the case of an open-source platform these are rather internal guarantees, availability of support, etc.

Since reliability directly affects the quality of the SOC service provided, this criterion has a **weight of 4** (higher importance).

Scoring recommendation:

- 5 points: No serious outages, reliable production references, support for HA deployment.
- 3 points: Occasional minor outages, partial HA support.
- 1 point: Frequent or significant outages, no HA support.

3.2.3 K6: Compliance with standards and interoperability (weight: 4)

Justification: From a technical point of view, it is significant whether the platform supports open standards and common technologies, which determines its interoperability with other systems. The solution should be compatible with popular hypervisors, network protocols, storage technologies and management interfaces (e.g. the OpenStack API is the de facto standard for open-source clouds). Broad support for standards also facilitates integration and minimizes the risk of proprietary limitations. We also assess the availability of certifications or references – for example, whether the platform is based on proven open-source projects supported by a large community. As cloud provider evaluation practices indicate, adherence to industry standards and certifications (e.g. ISO 27001 in security) is a significant selection criterion. In our project, interoperability will make it possible in future to interconnect multiple cloud environments and integrate further components.

It is also important that the platform contains tools for implementing the required measures (e.g. logging, encryption, access controls) and that the community/vendors provide guidance on regulatory compliance. Regulatory compliance will affect the acceptability of the solution in production – for example, for public administration data the platform should support deployment in compliance with the personal data protection law (GDPR). Given the nature of the planned SOC solution (which may process clients' sensitive security data), we consider this criterion to be very significant and assign it a weight of 4.

Scoring recommendation:

- **5 points:** Full compatibility with open standards, certifications (ISO 27001, GDPR compliant), active interoperability.
- **3 points:** Partial compatibility and interoperability, some standards are not fully covered.
- **1 point:** Limited or no compatibility, interoperability problems.

3.3 OPERATIONAL CRITERIA

3.3.1 K7: Complexity of deployment and management (weight: 3)

Justification: This criterion assesses the effort and resources that will need to be expended on initializing the platform and its subsequent management. It covers the complexity of installation (the number and intricacy of the steps involved in deploying the entire environment), the requirements for staff expertise, the time needed for full commissioning, as well as the clarity of maintenance (updates, upgrades, scaling). For example, some open-source platforms are highly configurable but may be complex to deploy, which requires an experienced team or the use of automation tools (e.g. Ansible scripts, etc.). From an operational point of view, we also evaluate whether the solution provides functionalities that facilitate management – e.g. self-healing, monitoring, simple cluster expansion, etc. The aim is to choose a platform whose operation will not be a disproportionate burden for the organization.

We give this criterion **a weight of 3** (medium importance), given that some additional effort with an open-source solution is acceptable as long as the other benefits outweigh it.

Scoring recommendation:

- 5 points: Simple deployment, clear documentation, minimal management requirements, strong automation.
- 3 points: Moderately more complex deployment, manageable with the available tools and documentation.
- 1 point: Complicated installation, demanding administration, insufficient documentation.

3.3.2 K8: Support, availability of documentation and community (weight: 4)

Justification: The level of support available for a given platform significantly affects its practical usability. For commercial solutions, we consider the quality of the vendor's technical support (24/7 availability, SLAs for incident resolution). For open-source solutions, the size and activity of the community are important – the number of contributors, the frequency of new releases and bug fixes, and the availability of documentation and a knowledge base. A strong community and support from large IT companies are a guarantee of the project's long-term sustainability. The existence of training, certifications or consulting services for the given platform is also evaluated. Experience shows that smaller providers may offer more personal support, while larger players have more extensive support resources, but often at an extra charge. Since in our case we are considering an open-source platform, the criterion focuses on community support and the ecosystem around it. The project's development activity and the frequency of updates are evaluated

We assign it **a weight of 4**, as the availability of expert support and active development are very important for the success of the project.

Scoring recommendation:

- **5 points:** Active community, rich documentation, regular updates and professional support.
- **3 points:** Average community, basic documentation, slower updates.
- **1 point:** Weak or non-existent community, outdated documentation, minimal support.

3.4 SECURITY CRITERIA

3.4.1 K9: Authentication and access control (weight: 5)

Justification: The platform should include the built-in security features needed to protect the infrastructure and data in the SOC environment. These include, for example, support for multi-level authentication and authorization of users, the possibility of data encryption (in storage, in transit), logging and auditing of events, and automated security policies. For an open-source solution, we assess which security modules it offers and whether it complies with modern security practices (regular security updates, hardening options).

Given the nature of the project (cybersecurity), this is a **critical** criterion with a weight of **5**. A platform that could not secure data and systems at an adequate level cannot be considered suitable.

Scoring recommendation:

- 5 points: Advanced multi-level authentication, robust access control, full support for logging and auditing.
- 3 points: Basic authentication and access control, limited auditing options.
- 1 point: Weak authentication security, minimal auditing options.

3.4.2 K10: Tenant isolation and data protection (weight: 5)

Justification: In a multi-tenant environment, it is essential that the platform guarantees strict isolation between individual customers (tenants) – their virtual machines, storage and networks must be separated so as to prevent unauthorized access or data leakage between tenants. This criterion evaluates the platform's architecture in terms of multi-layered security: separation of compute resources, network segmentation, access control mechanisms and the like. **The degree of data isolation between tenants is a fundamental aspect of multi-tenant systems**.

Any weakness in isolation could lead to security incidents; we therefore assign this criterion **a weight of 5** (very high).

Scoring recommendation:

- **5 points:** High tenant isolation, complete resource segmentation, advanced data protection mechanisms.
- **3 points:** Partial isolation with some limitations, less robust protection mechanisms.
- **1 point:** Weak tenant isolation, risk of data leakage, weak access control.

3.4.3 K11: Network/monitoring integration (weight 3)

Justification: Justification: Evaluates the degree of integration with monitoring tools and security monitoring systems – in particular SIEM, IDS/IPS (e.g. Suricata, Snort), as well as metrics/logging tools such as Ceilometer, Grafana, Prometheus. It also takes into account the platform's ability to export security events, support for APIs/webhooks and the options for integration with external security components. The criterion complements the evaluation of authentication and isolation with the platform's ability to be part of an active security environment.

Scoring recommendation:

- 5 points: Full integration with tools (SIEM, IDS/IPS, Grafana, Prometheus), simple configuration.

- 3 points: Partial integration, some important tools are supported to a limited extent.
- 1 point: Very limited integration, problems with connecting tools and exporting events.

3.5 EVALUATION METHODOLOGY

Each solution will be scored on each criterion on a scale of **1 to 5**:

For an objective selection of the most suitable platform, we chose a **multi-criteria evaluation approach** based on scoring the individual alternatives against the established criteria and using a weighted sum. The evaluation methodology consists of the following steps:

1. **Defining the scoring scale:** Each criterion will be scored from 1 to 5 points, where 1 means that the solution meets the criterion very poorly or not at all, and 5 means excellent fulfilment of the criterion. This scale provides sufficient resolution and is easy to understand. For consistency of evaluation, verbal descriptions will also be established for the individual points (e.g. 1 – insufficient fulfilment, 3 – average fulfilment, 5 – excellent fulfilment). All evaluators must use the scale uniformly.
2. **Assigning weights to criteria:** Each criterion is assigned a **weight in the range 1–5** according to its relative importance (these weights are given above in the description of the criteria). The weight expresses how many times more significant the given criterion is compared with others. The most critical factors (e.g. security, key functions, costs) have a weight of 5, less important aspects a lower weight. We determined the weights by expert estimation based on the needs of the project; if necessary, they can be refined by pairwise comparison methods or by team consensus:
    - 5 = excellent fulfilment,
    - 4 = full fulfilment,
    - 3 = partial fulfilment,
    - 2 = limited fulfilment,
    - 1 = minimal or no fulfilment.
3. **Scoring the alternatives:** For each cloud platform alternative under consideration, all 11 criteria will be evaluated on a scale of 1–5. The evaluation will be based on the findings of the analysis (output V1) – e.g. the extent to which the given platform provides the required functions, what performance it has according to tests or references, how it meets the security requirements, etc. It may be a combination of measured data (e.g. from performance tests), information from documentation and expert judgement. Each assigned score will be justified so that the process is transparent.
4. **Calculating the weighted score:** Subsequently, for each alternative we calculate the overall **score** as the weighted sum of the points. I.e. the score for a criterion is multiplied by the weight of that criterion and these products are summed across all 11 criteria. The result is a single number – the overall evaluation of the alternative. This **weighted scoring model** is a proven tool for multi-criteria decision-making, making it possible to quantify the degree to which each option meets the overall requirements.
5. **Comparison and selection:** Alternatives are ranked according to the weighted score achieved. The highest score indicates the platform that best meets the established criteria and therefore represents the optimal choice. In the case of close results, a sensitivity analysis can be carried out – e.g. checking the effect of changes in the weights on the ranking, or examining in detail the differences in the key criteria. The

result of the process will be a recommendation of a specific platform (in the next output of the project, V3) with a justification based on the points achieved.

The procedure described ensures that the decision will be **transparent and defensible**. All inputs (weights, scores) will be documented. The approach is flexible – if necessary, new criteria can be added or the weights adjusted and the scores recalculated. In this way, the risk of a wrong choice is minimized and it is ensured that the selected solution will correspond to the defined needs of the project.

3.6 PROPOSED EVALUATION MATRIX

The following evaluation matrix (Table 1) serves to present the evaluation clearly. The rows represent the individual criteria (divided by category) with the assigned weight. The column "Point score (1–5)" will contain the points assigned to the alternative being evaluated – in the selection process, this table will be filled in for each platform assessed.

4 PROPOSED EVALUATION TABLE FOR THE SELECTION OF A CLOUD PLATFORM

Criterion (Kx)	Weight (1–5)	Score (1–5)
K1: Support for IaaS functionality	5	[1 - 5]
K2: Modularity and architecture	4	[1 - 5]
K3: Orchestration and automation	3	[1 - 5]
K4: Performance and scalability	5	[1 - 5]
K5: Reliability and stability	4	[1 - 5]
K6: Standards and interoperability	4	[1 - 5]
K7: Complexity of deployment and administration	3	[1 - 5]
K8: Documentation and community	4	[1 - 5]
K9: Authentication and access	5	[1 - 5]
K10: Tenant isolation and data protection	5	[1 - 5]
K11: Network / monitoring integration	3	[1 - 5]

The table also shows the proposed weights for the individual criteria (second column). When evaluating a specific platform, the score achieved, from 1 to 5, is entered in the third column. The weighted score is then calculated as the sum of the products of the weights and points in each row. Such a matrix will help to quickly identify the strengths and weaknesses of the evaluated solutions – for example, it shows which criteria have the highest weight and what score the alternative obtained in them. This will enable a transparent discussion of the results and possible fine-tuning of the evaluation. The matrix is designed to be universal; in the project, we plan to use it to compare several open-source platforms, with the expected output being the winning solution with the highest overall score.

CONCLUSION

The proposed system of comparison criteria provides a systematic framework for selecting a cloud platform for a multi-tenant SOC solution. In this document, we defined 11 key criteria in four categories – functional, technical, operational and security. Each criterion was justified with regard to the objectives and requirements of the project and was assigned a weight expressing its relative importance. This ensures that the evaluation will cover all relevant areas: from the platform's ability to provide the required functions, through its performance and reliability, costs and support, to the robustness of the built-in security mechanisms.

The criteria and weights compiled in this way will make it possible to objectively **compare alternative solutions**. By means of scoring and a weighted sum, we can quantify the degree to which the requirements are met for each platform assessed. The weights will ensure that the most important areas (e.g. security, critical functions, costs) have an appropriate impact on the resulting score. The selection process based on these criteria is transparent – there will be a justification for each mark, and the resulting score will clearly show which option best suits the project.

The proposed system of criteria and evaluation significantly increases the chances that we will choose a **suitable platform** for building a multi-tenant security centre. It enables us to identify the solution that best meets the functional requirements, will fit into the existing infrastructure, will be technically reliable and scalable, economically acceptable and at the same time will ensure a high level of cybersecurity. Such a platform will create a solid foundation for implementing the project's intentions and achieving its objectives.

LIST OF SOURCES

[1] I. Voras, B. Mihaljevič and M. Orlic, "Criteria for Evaluation of Open Source Cloud Computing Solutions," rev. *roceedings of the ITI 2011, 33rd International Conference on Information Technology Interfaces, Cavtat, Croatia, 2011*.

[2] F. Cocozza, "Cloud Management Platform Selection: A Case Study in a University Setting," 2015.

[3] Business analytics review, https://library.fiveable.me/business-analytics/unit-11/multi-criteria-decision-making/study-guide/Xgpbm7FKThBDW6Oe.

[4] G. Chalan, Developing Objective Comparison Evaluation, 2024.

[5] D. Greibler et al., "Private IaaS Clouds: A Comparative Analysis of OpenNebula, CloudStack and OpenStack," Conference: 2016 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP), 2016.