UNIVERSITY OF ŽILINA
Faculty of Management Science and Informatics

BRAIN:IT

PROJECT

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

Linked to work package KPB1 – Design of the virt. solution

Linked to output V1, the result of the analysis of available solutions

Funded by the European Union
NextGenerationEU

RECOVERY AND RESILIENCE PLAN

GOVERNMENT OFFICE OF THE SLOVAK REPUBLIC

Contents

1 Cloud computing 3
Reference model 3
Cloud entities 3
Cloud consumer 3
Cloud provider 4
Cloud auditor 4
Cloud broker 4
Cloud carrier 4
Interactions between entities 4
Essential characteristics of CC 5
Service model: Software as a Service (SaaS) 5
Who are the consumers? 6
What does the consumer get? 6
How are charges calculated? 6
Relationship to the layered model 6
Examples of use 6
Advantages 7
Service model: Platform as a Service (PaaS) 7
Who are the consumers? 8
What does the consumer get? 8
How are charges calculated? 8
Relationship to the layered model 8
Examples of use 9
Advantages 9
Service model: Infrastructure as a Service (IaaS) 9
Who are the consumers? 9
What does the consumer get? 9
How are charges calculated? 9
Relationship to the layered model 9
Logical architecture of the IaaS cloud model 10
Advantages 11
Deployment models 11
On-site private cloud 12
Outsourced private cloud 13
On-site community cloud 13
Outsourced community cloud 14
Public cloud 15
Hybrid cloud 16

1 CLOUD COMPUTING

Document [7] defines cloud computing (CC) as a system model in which the provider enables the consumer to have convenient, ubiquitous, on-demand access to shared configurable computing resources (networks, servers, storage, applications and services). These computing resources can be provisioned or released with minimal intervention by the provider.

Reference model

Document [8] defines the reference model of the cloud computing system.

This model depicts the main entities, their purpose and their function in the system.

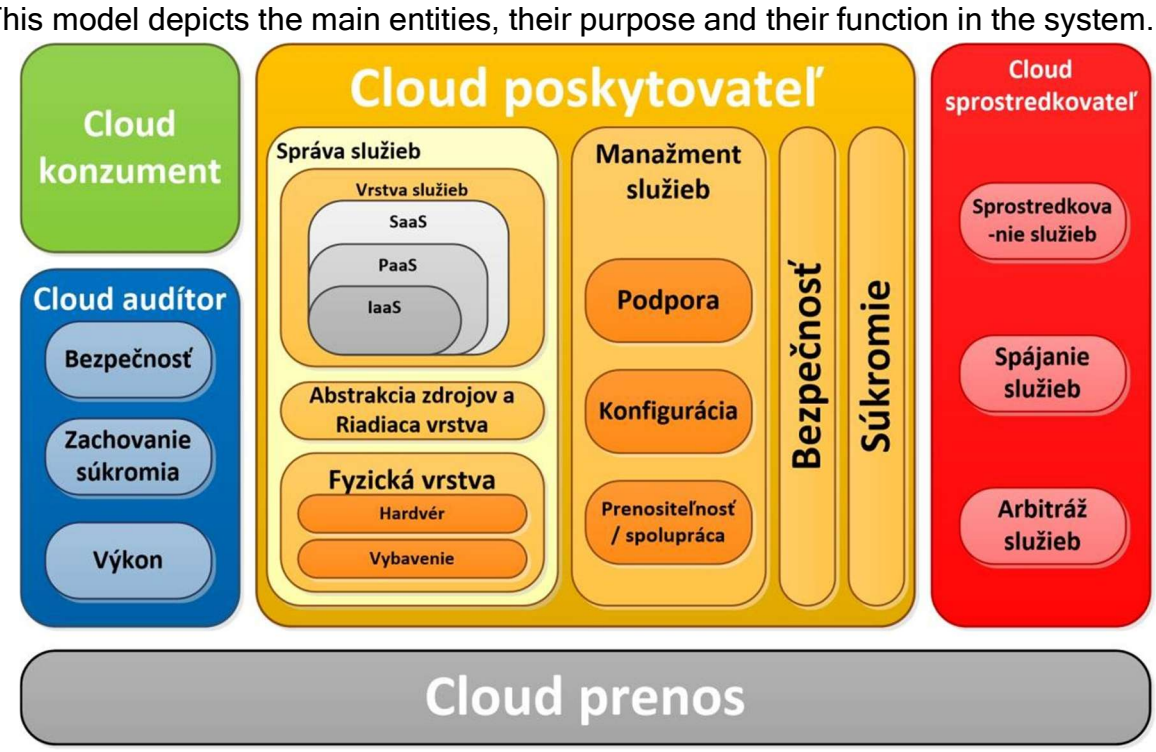


Fig. 5 - Reference model

Fig. 5 presents all five entities: *the cloud consumer, cloud provider, cloud carrier, cloud auditor* and *cloud broker*. Each entity participates in a transaction or process and/or performs some role in CC. The reference model further consists of five essential characteristics, three service models (Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS)) and four deployment models.

Cloud entities

Cloud consumer

A person or organization that maintains a relationship with, and uses the services of, a cloud provider. The consumer chooses from the provider's services, submits requests for a specific service and then uses it. The service may be charged for on the basis of a Service-level-agreement (SLA).

Cloud provider

A person or organization responsible for making a service available to interested parties. The provider manages the computing infrastructure required for providing the services, implements the software that provides the services and creates the infrastructure required for delivering the services to consumers over the internet. It provides its services on the basis of an SLA.

Cloud auditor

An entity that can perform an independent examination of a service for the purpose of objective assessment. Audits are performed to verify conformance to standards. The examination may concern security, privacy, performance, etc.

Cloud broker

An entity that manages the use, performance and delivery of services and negotiates the relationship between the consumer and the provider. The consumer can thus request a service from the broker instead of making a direct request to the provider. The entity helps the consumer manage access to the provider.

The broker provides services in three categories:

- **Service intermediation:** improves a given service by enhancing some of its properties and providing added value to them for the consumer. This may involve access control to services, security enhancement, performance assessment, etc.
- **Service aggregation:** combines and integrates multiple services into one or more new services. It provides data integration and ensures its secure movement between consumers and multiple providers.
- **Service arbitrage:** similar to service aggregation. The broker has the option of selecting services from different providers. In this way it is possible, for example, to select the best of several identical services according to a rating score.

Cloud carrier

An intermediary that provides connectivity and transport of services between the consumer and the provider via networks, telecommunication devices and others.

The consumer can thus use services via devices on the internet such as PCs, phones, etc.

Interactions between entities

Fig. 6 depicts the interactions between the individual entities:

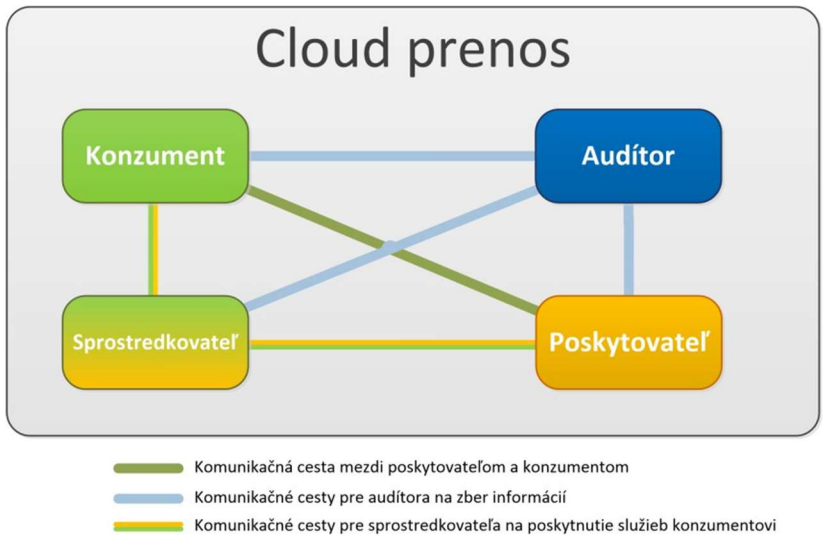


Fig. 6 - Interactions between entities

Basic characteristics of CC

Document [7] defines the essential characteristics of CC in more detail:

- **On-demand self-service:** the consumer can automatically be provided with the computing resources it needs without human interaction with the service provider.
- **High network availability:** resources are available over the network and can be accessed from various standard devices such as PCs, mobile phones and tablets.
- **Resource pooling:** the provider's computing power is shared among multiple consumers. Physical and virtual resources are dynamically assigned and reassigned according to their requirements. Thus the consumer does not know the exact location where the resources assigned to them are situated (at a higher level of abstraction, the location can be narrowed down, e.g. to a country or a data centre). Examples of such resources are storage space, CPU, memory and network bandwidth.
- **Rapid elasticity:** resources are elastically allocated and released, in some cases automatically, in order to adapt rapidly to the corresponding demands. To the consumer, the resources thus appear to be unlimited and can be used in any quantity at any time.
- **Measured service:** The cloud system automatically controls and optimizes resources by implementing metering at some level of abstraction appropriate to the given type of monitoring (e.g. storage space or active user accounts). Resource usage can be monitored, controlled and summarized in reports, thereby providing transparency for both the provider and the consumer of the given service.

Service delivery model: Software as a Service (SaaS)

The service for the consumer consists of applications accessible over the internet, provided by the provider. They are accessible from various consumer devices, e.g. via a web browser or a programming interface.

Who are the consumers?

- Organizations that provide their employees with access to typical office applications (spreadsheet, e-mail, etc.).
- End users who use the provided software directly.
- Application administrators who configure the applications for end customers.

What does the consumer get?

The right to use a specific application on demand and data management, such as data backup and sharing between consumers.

How are fees calculated?

They are usually based on the number of users, the time the application is used, the number of uses, launches or processed records; the bandwidth used and the amount of data stored.

Relationship to the layered model

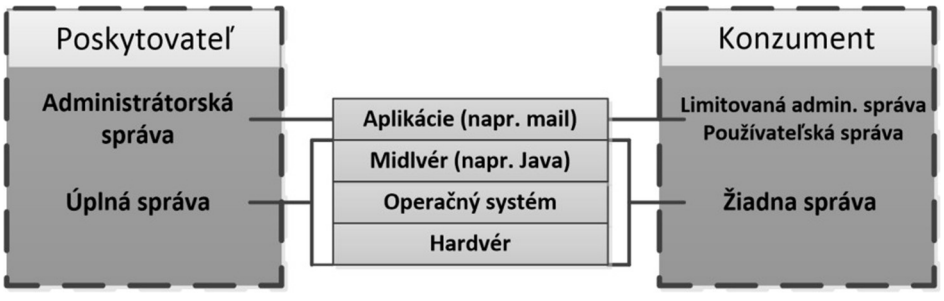


Fig. 7 - SaaS - relationship to the layered model

The consumer pays for access to the application but neither manages nor controls the lower layers of the model (network, servers, operating system, storage space, nor some settings of the application used):

In the SaaS model, the consumer has access to the resources allocated to them by the provider's SaaS application. For example, e-mail, where the consumer can create, send and store mail. If the application allows it, the consumer has the option of limited administrative control (creating accounts for other consumers, an overview of activities, etc.).

The provider is the administrator with all rights to the SaaS application. Its task is the configuration, updating and provision of the required services.

Examples of use

Examples of suitable use:

- **business activities -** connecting businesses with suppliers or customers (mail, invoicing, inventory management),

- **collaboration -** applications for team collaboration (screen and document sharing; calendar, online gaming),
- **office productivity -** applications such as text editors, spreadsheets, presentation programs and databases (they extend the programs with collaboration capabilities),
- **software tools -** applications for security management, applications for web development, etc.

Examples of unsuitable use:

- **real-time software -** applications for aviation, control of industrial robots, etc. (data transfer via the Cloud cannot guarantee a sufficiently fast response),
- **transfer of huge amounts of data -** applications in medicine or physics, where huge amounts of data are generated, which may not be advantageous to transfer via the Cloud,
- **critical software -** applications where an error may mean damage to property or loss of life (response problem).

Advantages

- **Simplicity -** the consumer's web browser is efficient enough to display the requested data, so there is no need for additional applications. Therefore, no software installation is required, and only a minimum of application-related data is thus located on the consumer's PC.
- **Efficient use of licences -** a single licence can be used by multiple consumers simultaneously. This is made possible by the fact that the software is on the provider's side and the provider includes the consumer's use of licensed products in the price of providing the services.
- **Centralized management and data -** from the consumer's point of view, all data appear to be centralized, since they are located at the provider (in reality they are decentralized for reasons of redundancy, etc.). Logical centralization guarantees the consumer protection against data loss and security, and enables backups to be made, etc.
- **The provider assumes responsibility for the platform -** the consumer does not participate in managing the lower layers.
- **Financial savings -** the provider supplies all the hardware, power, equipment maintenance, etc., which should ultimately cost the consumer less money (even taking the fees into account) than if it had to provide all this itself.

Service delivery model: Platform as a Service (PaaS)

The service for the consumer is the possibility of placing their own software in the cloud environment or the possibility of using programming languages, libraries and tools provided by the provider.

Who are the consumers?

- Application developers.
- Application testers who test the application in various environments.
- Programmers who deploy new applications (or their updates) to the cloud environment.
- Administrators who configure, tune and monitor application performance.
- End users who use the application (for them, access to the application is the same as in the SaaS model).

What does the consumer get?

The possibility of using the provider's tools and resources for running, developing, testing, deploying and administering applications.

How are fees calculated?

They are usually based on the number of users, their type (e.g. developer/end user), storage size, computing resources used, bandwidth used by the platform and the length of time it is used.

Relationship to the layered model

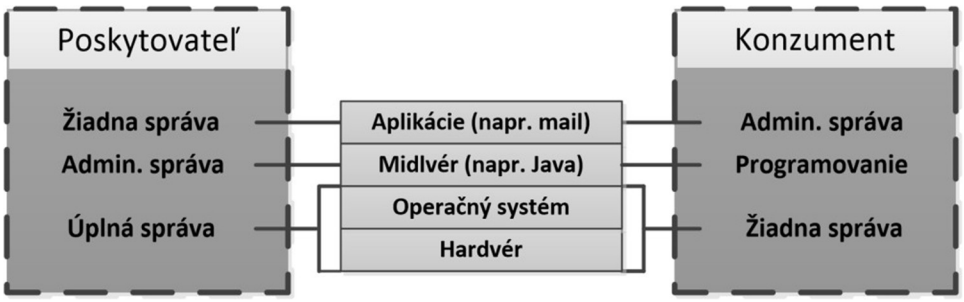


Fig. 8 - PaaS - relationship to the layered model

The consumer neither manages nor controls the lower layers of the model (network, servers, operating system, storage space), but has the ability to control and configure applications.

In the PaaS model, the provider manages the lowest layers of the model (operating system, hardware, networking within the LAN and between data centres). It provides the consumer with an environment for running and programming its applications, as well as CPU performance, memory, storage space, databases, network connectivity, etc. The provider determines the programming model, i.e. the conditions under which the consumer will be able to program its own applications (for the purpose of charging). The consumer has full control over its own applications.

The PaaS model provides space for the development, running and administration of software that is intended for a large number of consumers, processes large amounts of data and will potentially be reachable from any location via the internet.

PaaS is very similar to a traditional computer system. Unlike it, applications in the PaaS model can use as much computing power as they need, process large amounts of data, be deployed almost instantly and serve a large number of users.

Examples of use

The PaaS model environment can be used to develop applications providing the services mentioned for the SaaS model.

Advantages

The consumer is freed from the obligation to select, install, maintain or control the platform components. It has many advantages in common with the SaaS model (*simplicity, centralized management and data, responsibility for the platform is assumed by the provider, financial savings*).

Service delivery model: Infrastructure as a Service (IaaS)

The service for the consumer is the provision of computing resources, storage space, networking and other fundamental resources so that it can deploy and run arbitrary software in the cloud system (including the operating system and applications).

Who are the consumers?

System administrators.

What does the consumer get?

Access to a virtual computer, storage space available over the network, network elements such as a firewall, and the ability to configure services.

How are fees calculated?

Usually CPU usage/hour, stored data/hour, consumed bandwidth, use of network infrastructure (e.g. IP addresses)/hour and value-added services (e.g. monitoring, automatic scalability, etc.).

Relationship to the layered model

In the IaaS model, the operating system layer is divided into two parts. A lower (and more privileged) layer is added, in which a Virtual Machine Monitor (VMM), called a hypervisor, runs. It is used to coordinate multiple Virtual Machines (VMs), where a VM appears to the consumer as a physical computer that can be configured over the network. The operating system installed in a VM is called the guest operating system.

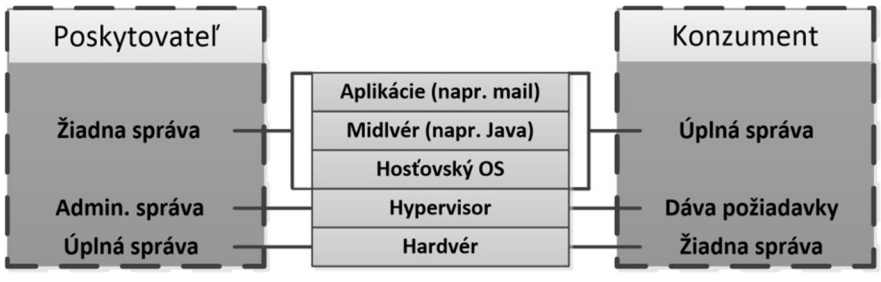


Fig. 9 - IaaS - relationship to the layered model

The provider controls the lowest layers of the model (physical hardware and hypervisor). The consumer can submit a request to create a new VM. This request is granted only if it complies with the provider's policy. Through the hypervisor, the provider can provide the consumer with virtual network elements so that the consumer can create and configure their own virtual network within the provider's infrastructure. However, the provider continues to remain responsible for managing, updating and configuring the infrastructure for the purposes of security and reliability.

In the IaaS model, the consumer has full control over the guest operating system in each VM and over all the layers above it.

Logical architecture of the IaaS cloud model

Companies develop proprietary solutions and do not provide detailed technical information about their systems and algorithms. Fig. 10 is based on information from open-source solutions (e.g. Eucalyptus).

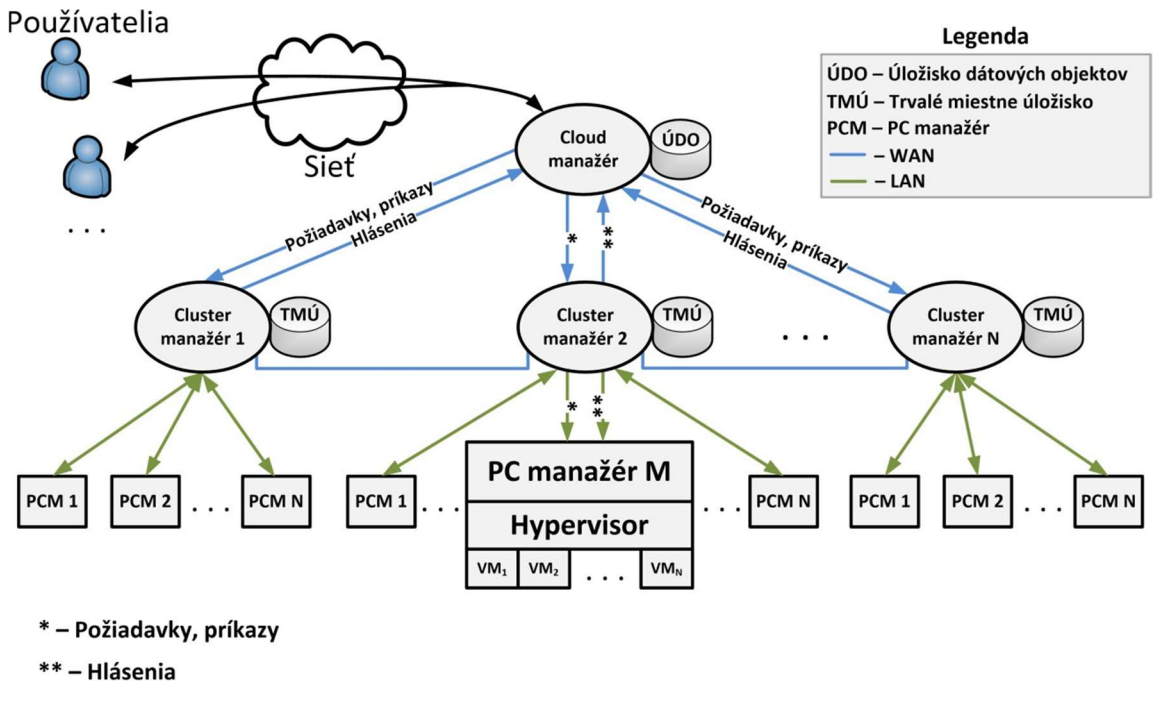


Fig. 10 - Logical diagram of the IaaS model

Advantages

- **Full control of computing resources through administrative access to VMs -** the consumer accesses resources through standard network protocols, which are encrypted. Access to resources takes three forms:
  - the consumer creates a request to create a VM or to store data on a cloud server,
  - The consumer with administrative access to the VM uses administrative commands,
  - any user can use the services of a VM via network services if the consumer has previously allowed them to do so.
- **Flexible and efficient rental of computing hardware -** in the IaaS model, the computing resources are administrative access to VMs, network bandwidth, storage space and IP addresses (monitoring services, firewalls, databases, etc. also belong here). In this way, consumers are provided with raw hardware performance and the ability to rapidly launch a large number of VMs or other resources. This makes it possible to quickly create a large network containing many VMs, which means that resources for solving large problems are created without the need to physically procure all the necessary hardware.
- **Portability, interoperability with legacy applications -** since the consumer can install an operating system of its choice in a VM, interoperability between legacy applications and applications in the IaaS model must be guaranteed. IaaS supports the portability (migration) of VMs.

Deployment models

For all deployment models, NIST has created a general concept of boundaries in order to separate the individual models from one another. These boundaries are described in document [9]. Fig. 11 depicts the key component - the *security perimeter*:

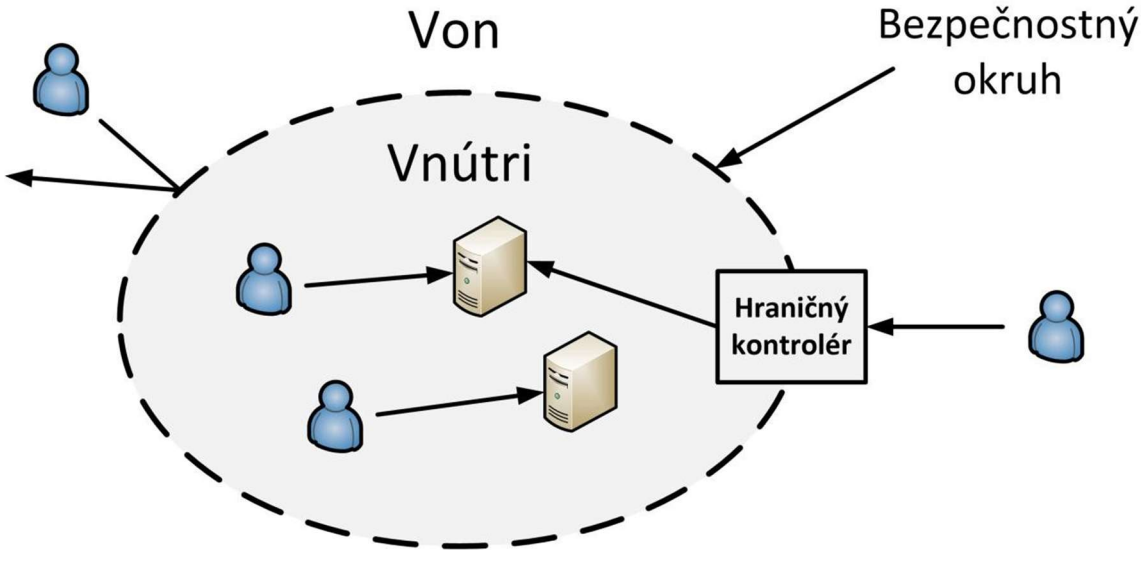


Fig. 11 - Security perimeter

The security perimeter is an access boundary:

- entities located inside can use all resources within the perimeter,
- entities located outside can use resources within the perimeter only if they are permitted to do so by the boundary controller (usually a firewall or VPN), which incorporates the access policy.

The security perimeter cannot be said to be, for example, merely a firewall. The concept of security is more general. It may be, for example, the separation of privilege levels within software or the boundary between an application and the operating system. The perimeter on its own is not an adequate protection mechanism; however, it is an important component in building security.

Cloud computing consists of four deployment models:

- **Private cloud -** the cloud infrastructure is available exclusively to a single organization comprising multiple customers. It may be owned and managed by the organization, a third party or a combination of them. They exist in on-site and outsourced variants.
- **Community cloud -** the cloud infrastructure is available exclusively to customers from a community of organizations that share interests. It may be owned and managed by one or more organizations of the community, a third party or a combination of them. They exist in on-site and outsourced variants.[4].
- **Public cloud -** the cloud infrastructure is intended for use by the general public. It may be owned and managed by a business, an academic or government organization, or a combination of them.
- **Hybrid cloud -** a cloud infrastructure consisting of two or more distinct infrastructures (private, community or public cloud) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application compatibility.

On-site private cloud

Fig. 12 presents a simple view of an on-site private cloud:

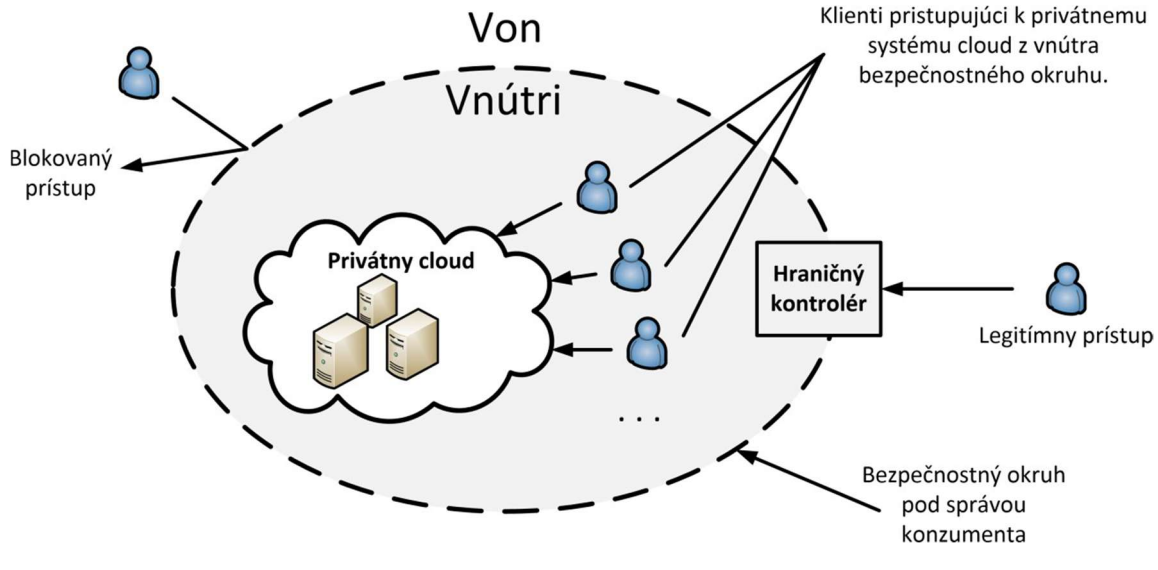


Fig. 12 - On-site private cloud

The security perimeter encompasses the consumer's local resources and the resources of the private cloud system. A private cloud may be centralized at a single consumer site or at multiple consumer sites. In this deployment, it is up to the consumer whether it implements a security perimeter. If so, the perimeter allows it to control access to its private resources.

Outsourced private cloud

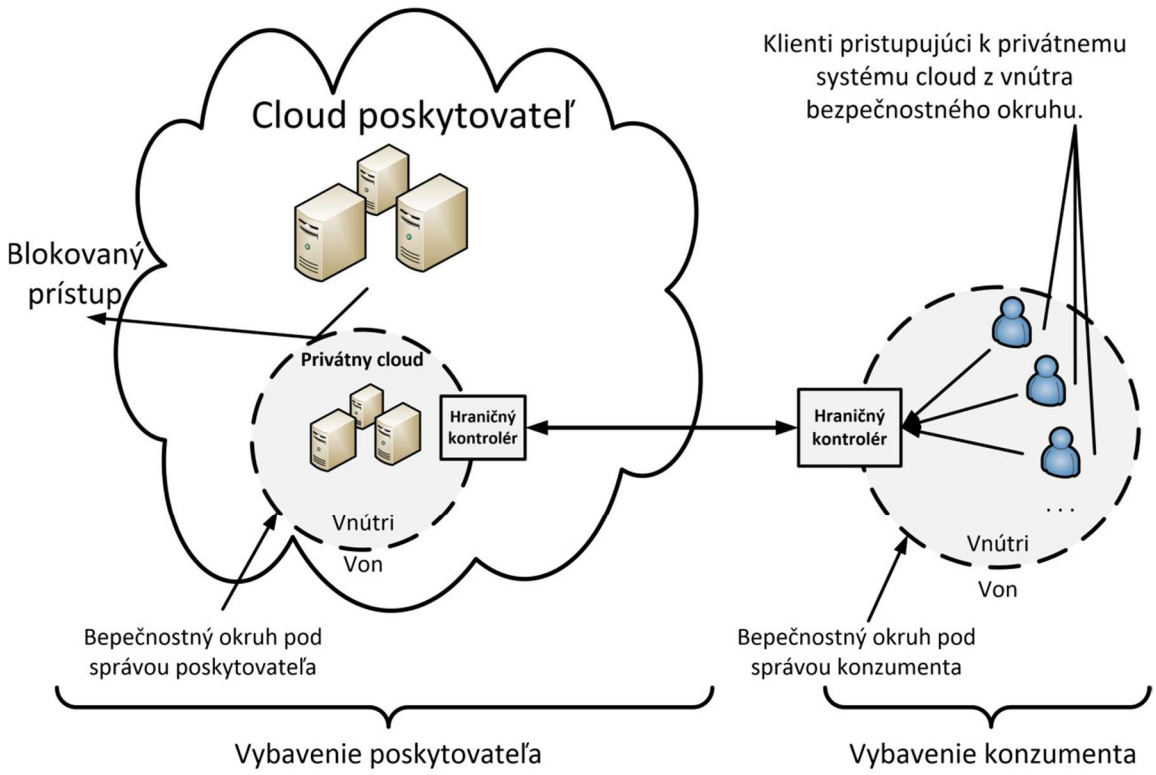


Fig. 13 - Outsourced private cloud

Fig. 13 presents a simple view of an outsourced private cloud:

This deployment contains two security perimeters, one implemented by the consumer and the other by the provider. Both are connected by a secured link. The security of the entire deployment and the availability of services are thus dependent on the strength and availability of both security perimeters. The provider assumes responsibility for preventing the resources of the private cloud system from being mixed with other resources in the cloud system that are outside its security perimeter. Various mechanisms such as VLANs, VPNs, separate network segments, etc. are used for this purpose.

On-site community cloud

Fig. 14 presents a simple view of an on-site community cloud consisting of a set of organizations:

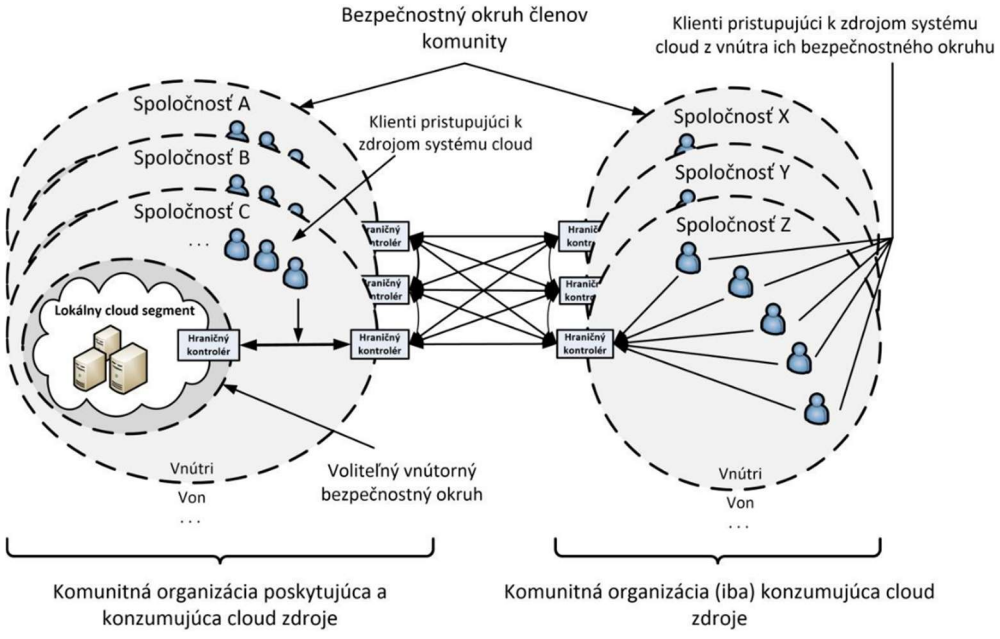


Fig. 14 - On-site community cloud

Each organization may provide CC services, consume them, or both. However, at least one organization must provide services for the deployment of the community cloud system to be functional. In the figure, the organization on the left provides services (and probably also consumes them) and the one on the right only consumes them. The connection between them is secured by a link between the two boundary controllers, which allows access to the security perimeter. It is also possible to implement an extra security perimeter that separates local resources from other local resources (in the figure).

Outsourced community cloud

Fig. 15 presents a simple view of an outsourced community cloud consisting of a set of organizations that consume cloud services:

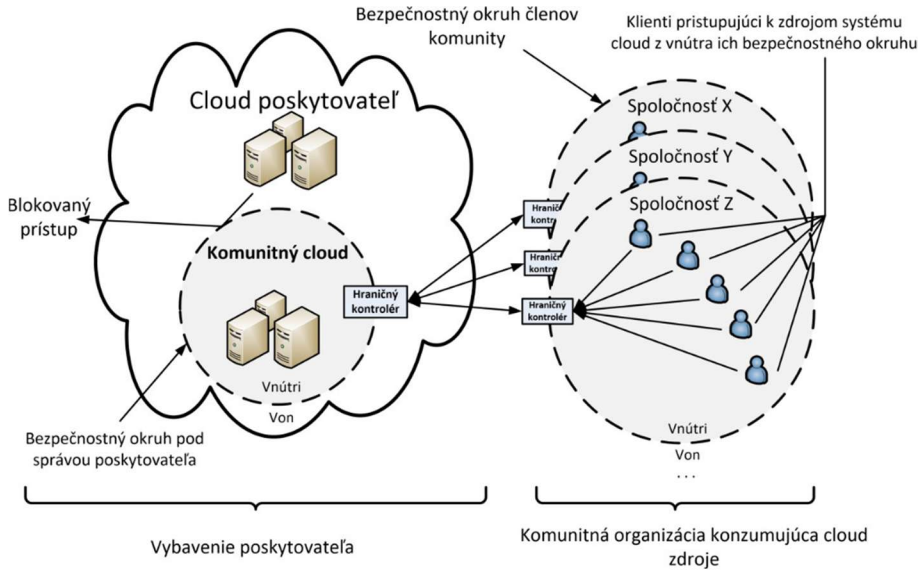


Fig. 15 - Outsourced community cloud

The deployment is very similar to the outsourced private cloud system. Server-side duties are the responsibility of the provider, which implements the security perimeter and prevents community resources from being mixed with other cloud resources that are outside the provider's security perimeter. An important difference is that the provider may create a sharing policy between the organizations in the community cloud system.

Public cloud

Fig. 16 presents a simple view of a public cloud:

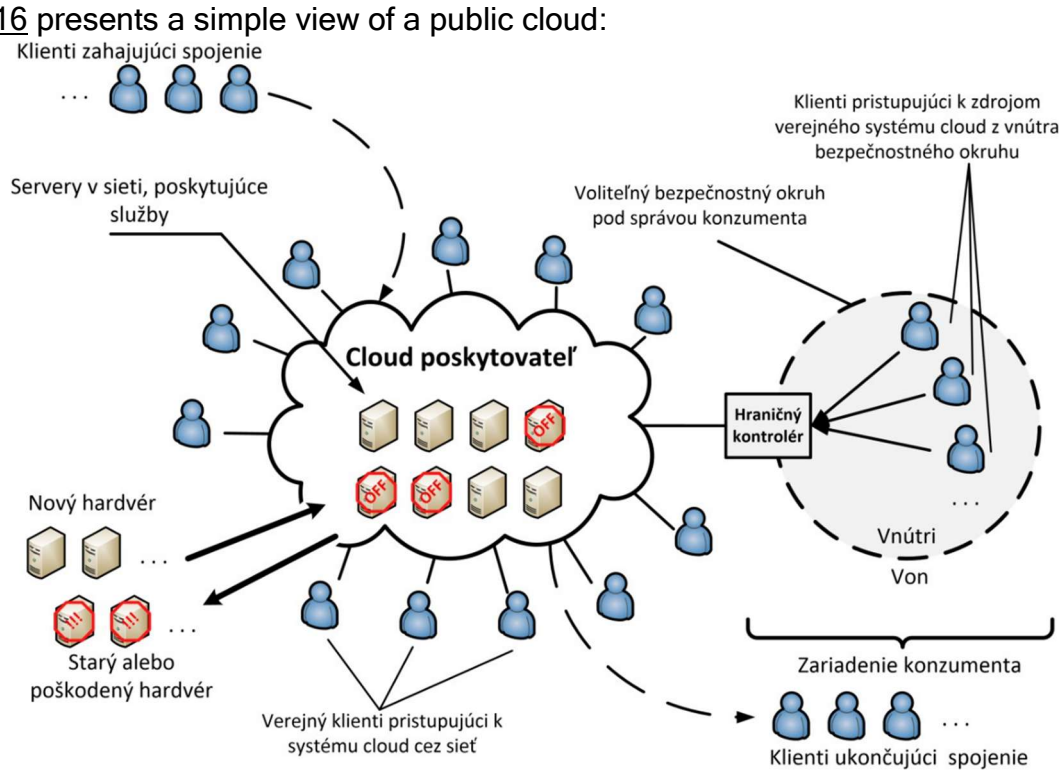


Fig. 16 - Public cloud

It is assumed that the provider has a large amount of resources (storage space, computing resources), that communication takes place over the public internet and that the cloud serves a large number of different consumers (or potential attackers).

Hybrid cloud

Fig. 17 presents a hybrid cloud:

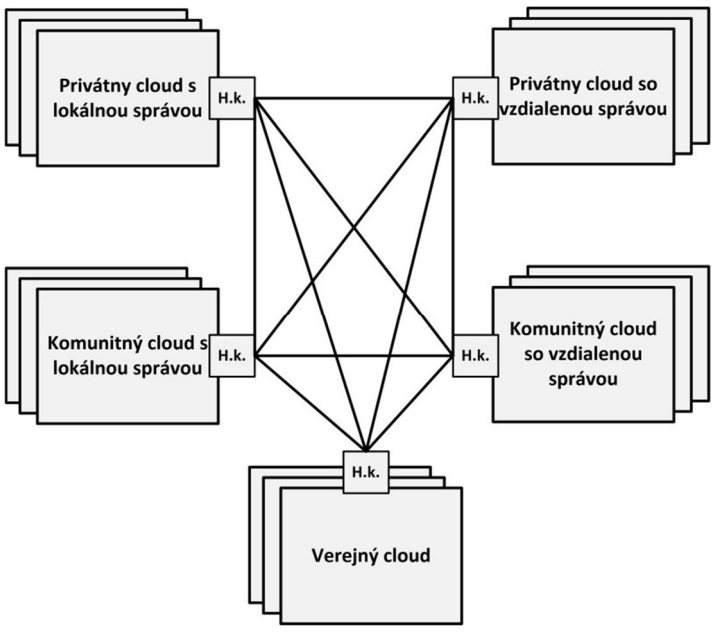


Fig. 17 - Hybrid cloud

A hybrid cloud is a composition of cloud systems in which each individual element is a different deployment. This figure presents the possibility of interconnecting all deployments (the figure omits the access point, network connectivity between them, the security policy, etc.). The possibilities for this deployment are very numerous. The hybrid cloud in the figure represents an extremely complex solution; however, simpler options also exist (e.g. when a company grows and needs to connect to another cloud system, use another cloud for data backup, etc.).