

APRIL 2025

VIRTUALISATION AND CLOUD SECURITY

Barbara Krašovec

TABLE of contents

01. Architectural Concepts and Design

02. Threat landscape and security challenges

03. Cloud Infrastructure Security

04. Cloud data security

05. Cloud application security

Virtualisation

Virtualisation architecture is the abstraction of physical resources, hypervisor sits on top of physical hardware and abstracts physical resources.

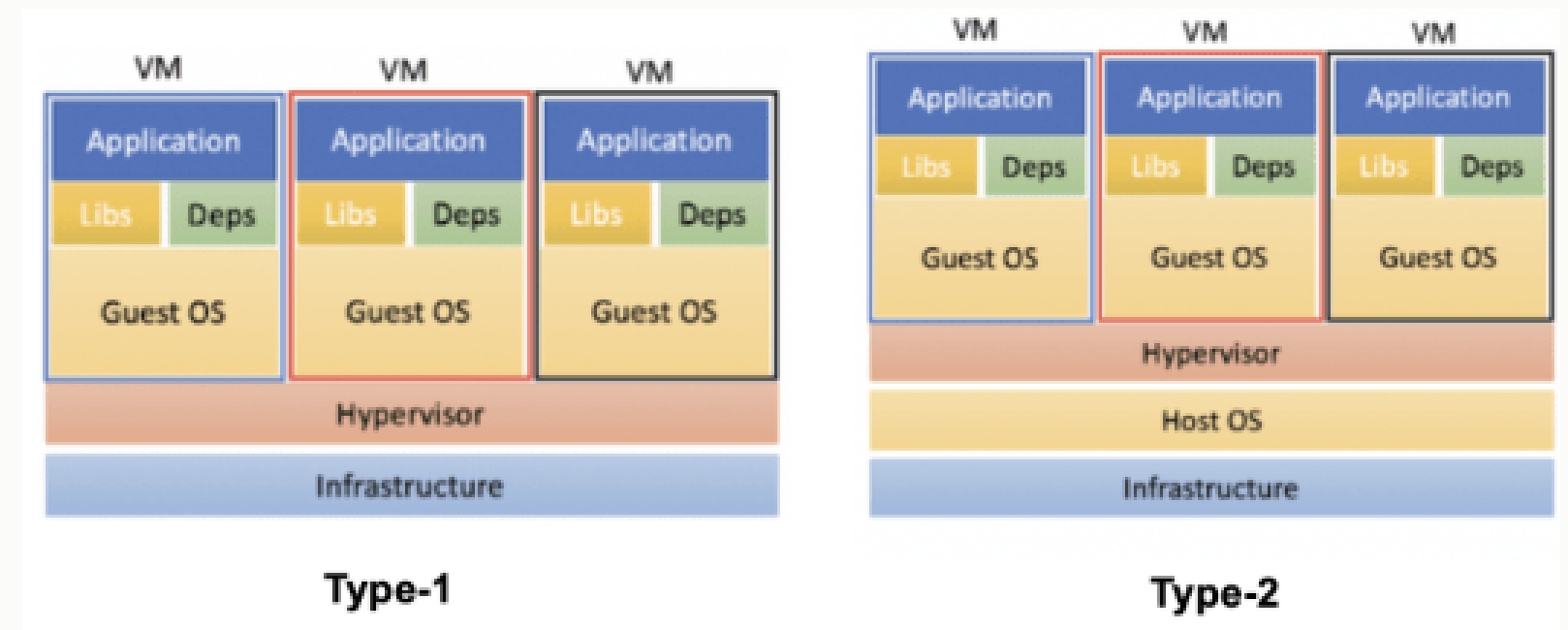
WHY VIRTUALISING?

- efficient usage of resources,
- multiple OSs on the same system,
- isolation of services,
- lower operating costs (compared to using physical machine for each service),
- flexibility,
- fast (re)deployment,

Hypervisor is the OS providing the virtualisation environment

Hypervisor

- Hypervisor is the main component in virtualisation,
- the hypervisor controls access between virtual guests and host hardware,
- once a hypervisor is compromised, the attacker owns everything,
- access control and monitoring of virtual administrators is critical,
- there is no way to completely isolate one OS from another (no physical separation as in physical networks) - direct memory access, resource sharing



Hypervisor security

Same security recommendations as for any other host:

- keep your software updated,
- remove/mask services that you don't need,
- if possible use the same hardware for all hosts (easier to follow vulnerabilities from just one vendor, use secure boot),
- apply HW firmware updates before OS installation,
- restrict access to hypervisors and monitor it (do you really need a public IP?),
- restrict access, use restrictive SSH access configuration,
- use SELinux (put a hypervisor in another security context),
- audit and use firewall, VPN, security monitoring.

VM security

- follow OS hardening guidelines (STIG and CIS controls),
- harden automatic installation and configuration (e.g. Ansible),
- remove services and packages that are not needed,
- provide security monitoring (e.g. Prometheus),
- track versions of OS, software, users with access, images,
- use asset management: keep track of what is (still) needed, if not delete,
- implement vulnerability management.

VM attacks

- hypervisor breakout (**VM escape**) - exploit of software vulnerability, rare (but CVE-2008-0923, CVE-2021-22960, CVE-2022-20779, CVE-2023-20867),
- **VM hyper jumping** by exploiting the host/hypervisor (e.g. CVE-2020-8554),
- **DoS** (e.g. Dirty Cow CVE-2016-5195),
- **side-channel attacks** (eg. Spectre, Meltdown CVE-2017-5715, CVE-2018-3646) - an unintended side effect of running code on the physical system breaks encryption - this is the cloud provider's responsibility,
- **misconfigurations** (e.g. network firewall rules exclude localhost),
- **code execution**,
- non-updated hypervisors with **outdated** packages,
- **hyperjacking** - by exploiting a vulnerability in the hypervisor the attack targets VMs

VM images

- Don't store credentials, certificates or any sensitive data in the image,
- harden the software in the image,
- sign images,
- don't use third-party images (use trusted sources),
- scan images for vulnerabilities,
- keep them updated
- track images (asset management).

Virtualisation security recommendations

- don't use default credentials,
- don't mix production and development VMs on the same hypervisor
- use different networks or security groups for production and development,
- define different risk profiles and place service with similar risk on the same host,
- use different credentials for production and development VMs,
- monitor all VMs (production, testing, development),
- shut down VMs that you don't need,
- always update offline VMs before putting them back online,
- maintain inventory of VMs,
- check for open ports, default passwords, unpatched software (nmap, Metasploit, OpenVAS, Nessus) - check also <https://github.com/dev-sec/puppet-os-hardening>

Virtualisation vs cloud

- **virtualisation is a technology:** that allows creating multiple environments from a single, physical hardware system, it is the ability to emulate hardware using software
- **cloud is an environment:** it can include bare metal, virtualisation, or container software

Running services in the cloud

Consider the benefits of running services in the cloud.

- What are your risks?
- What are your responsibilities?
- Which domains are under your control, and which in the hands of the cloud provider?
- Where will you store your data and how will you transfer it, and use it?
- Are there any regulations about storing the data in the cloud?

Cloud implementation models

There are 3 main types of as-a-Service solutions:

- IaaS - infrastructure as a service (VMs)
- PaaS - platform as a service (DB, k8s)
- SaaS - software as a service (applications)

Also:

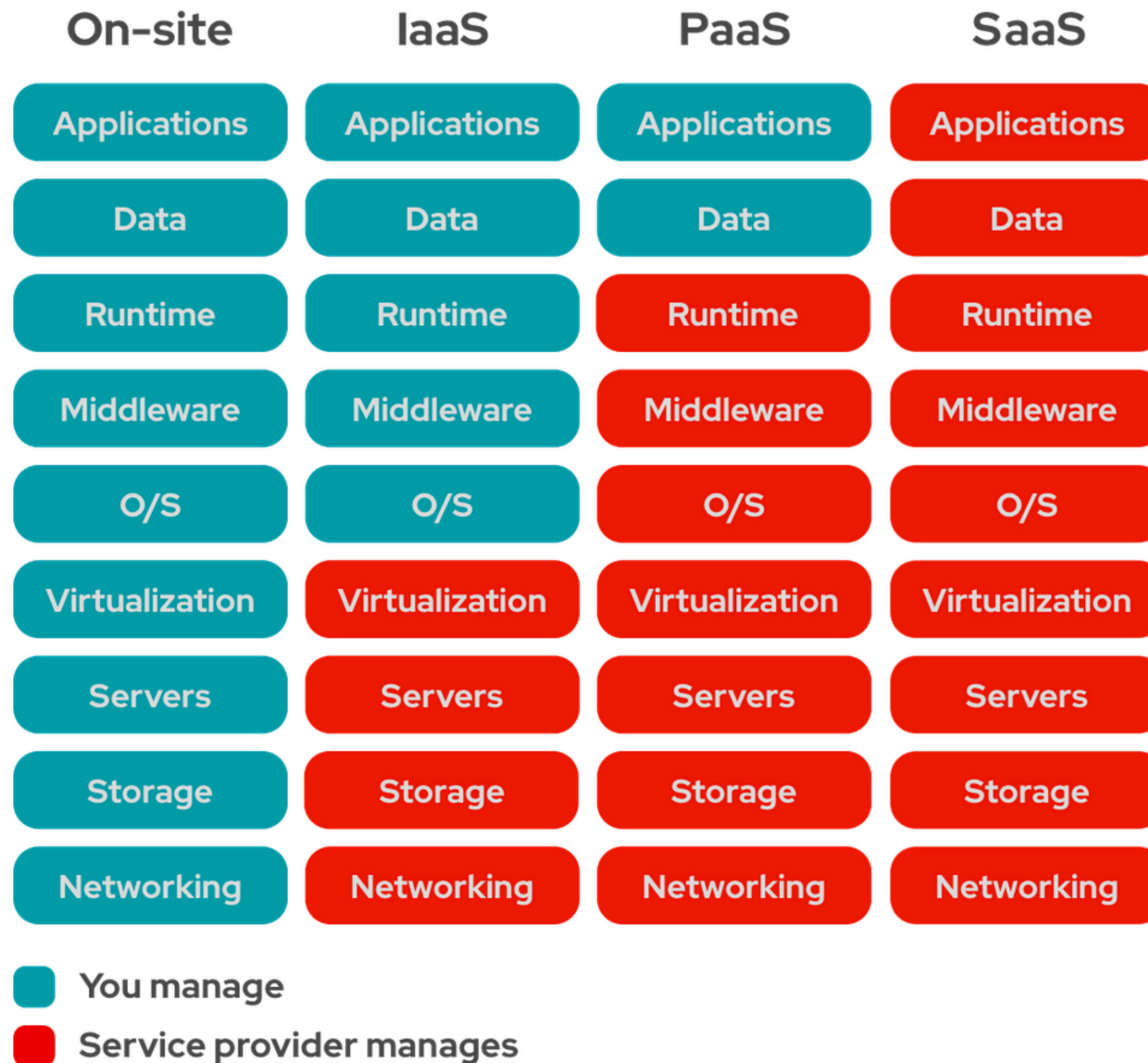
CaaS = Container as a Service

FaaS = Function as a Service

NaaS = Network as a Service

DSaaS = Data Storage as a Service

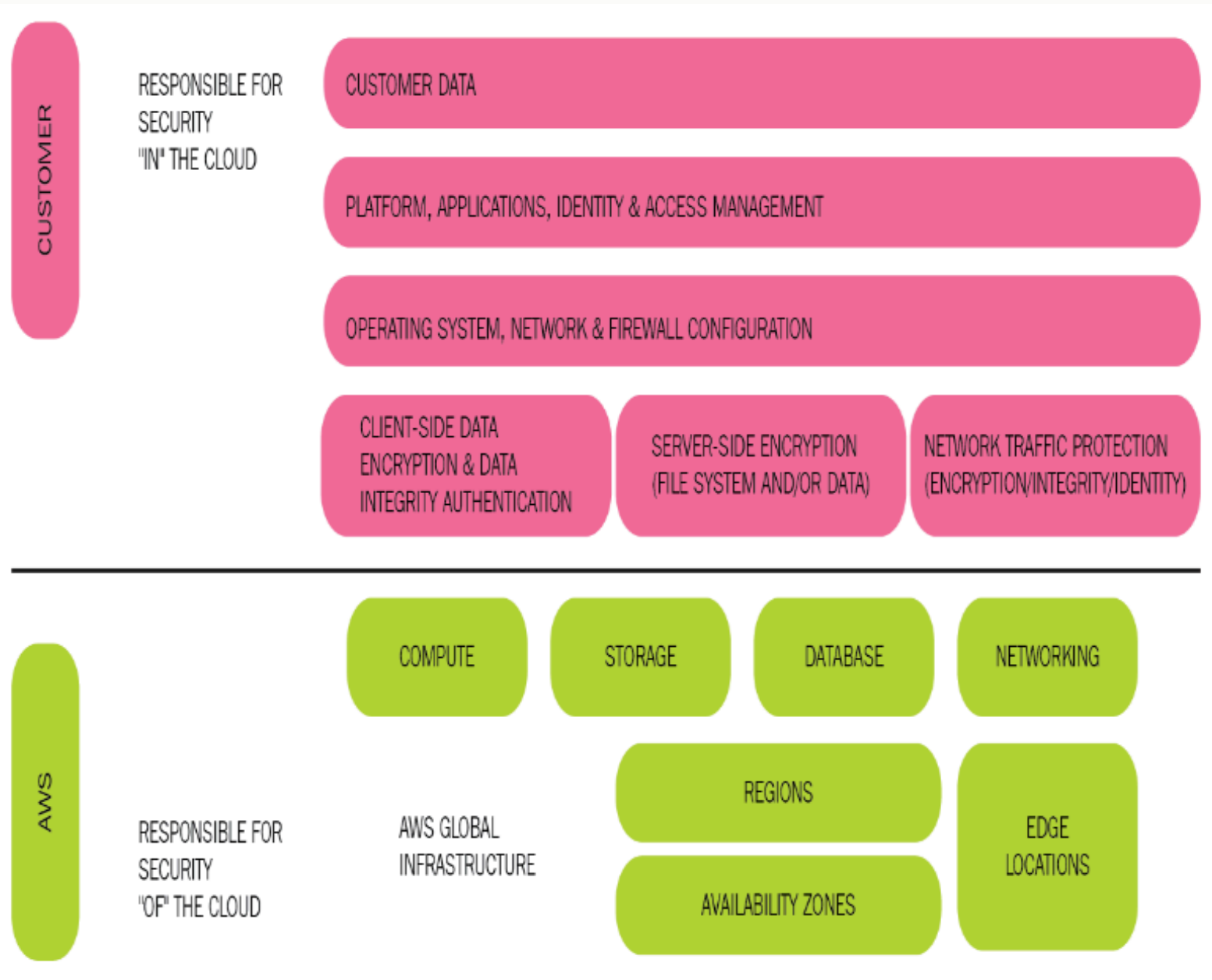
Cloud implementation models



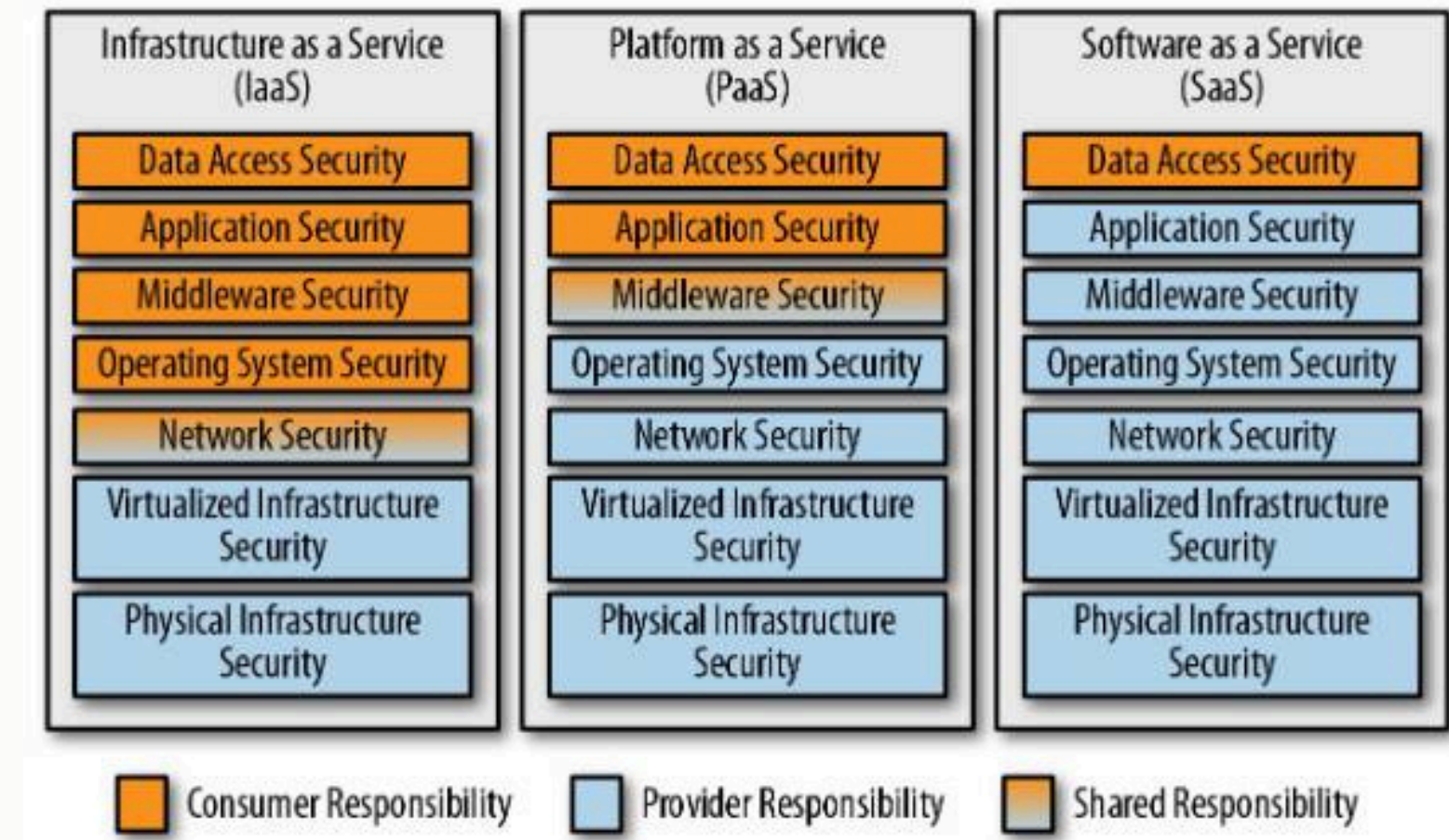
Security, responsibilities and maintenance are shared between the cloud provider and the customer.

What is your responsibility? If you can touch it, it is your responsibility.

Cloud models from security perspective



Source: Prashant Priyam, Cloud Security Automation, Packt



Source: Chris Dotson, Practical Cloud Security, O'Reilly

Security challenges in the cloud

- shared responsibility
- complex environment, high chance of misconfiguration, human error
- limited control
- insufficient IAM
- data loss (due to outages, accidental deletion)
- insecure APIs
- evolving threat landscape

- **for the customer:** no longer access to the hypervisor or hardware (physical, host security), cannot control which customers host on the same host and how well they protect their VMs; users are often in the role of sysadmin
- **for cloud providers:** complex network designs and no control over the state of VMs and services running on them

Concepts of cloud security

Some mechanisms:

- **Cloud IAM:** access and identity management (RBAC)
- **Encryption**
- **Cloud Security Posture Management (CSPM)** - identifying and mitigating risks associated with misconfigurations, compliance violations (eg. CloudCheckrm, Snyk, Prisma Cloud, CloudGuard, Wiz)
 - configuration management
 - active security monitoring
 - compliance checks
- **Data Loss Prevention (DLP)** - unauthorised sharing, data loss and leakage prevention (eg. Cloudlock, OpenDLP, MyDLP, Github DLP etc.)

- Cloud infrastructure/
platform security
- Cloud data security
- Cloud application security
- Cloud operations

Common threats in the cloud

- cyber attacks: DoS, spoofing, man-in-the-middle,
- escalation of privileges, unauthorized access,
- hijacking accounts,
- misconfigurations of cloud resources,
- internal/external threats,
- malware,
- data breaches and unauthorised access,
- data loss,
- unpatched software and exploited vulnerabilities,
- inadequate encryption practices/implementation,
- insecure interfaces/APIs,
- external data sharing and data transfers,
- insufficient technical skills,
- VM escape,
- leaked credentials (published in git),



- misconfiguration and inadequate change control,
- lack of cloud security architecture and strategy,
- insufficient key management,
- limited cloud usage visibility.

How to prevent them?

To gain unauthorized access, attackers need to:

- gain access to VM or
- gain access to the host from the VM

- update your software regularly,
- IAM configuration crucial! Apply MFA also for privileged accounts,
- limit access to the cloud services to a few locations,
- use encryption,
- apply network security,
- check VM for open ports, restrict access by enabling firewall,
- regular backups and disaster recovery planning.

Cloud data storage

Cloud has multiple data stores:

- **object storage,**
 - **block storage** and
 - **file storage.**
- An object store is like a valet parking: you give a car to a valet, he parks and gives you a ticket, and you don't care where the car is parked (files as objects, the application manages them, not caring about where they stay and how big they are)
 - Block storage as traditional hard disks (FC, iSCSI)
 - File storage presents itself as a filesystem (NFS, CephFS)

One is a fact: data will move between different physical nodes

Cloud data lifecycle



Source:CCSP Workbook

CREATE data: internally or externally, define the level of sensitivity, if created externally, encrypt it before sending it to the cloud.

STORE data: store to db, object store, as files on the system and secure (ACLs, encryption, auditing, logging access, redundancy, backup)

USE data: access + processing data + view + modify. Data has to be decrypted. Log and audit these actions. Use DLP.

SHARE data: data is used by others, configure access rules, security no longer in our domain.

ARCHIVE: move data to long-term storage

DESTROY: data is removed by Cloud provider, have a policy in place, who and when can delete it?

Data security strategies

- **Restrict access** to data,
- use **MFA**,
- use **tokenisation** - replaces a piece of sensitive data with a randomly generated token
- use **encryption**: protect data at rest, in transit, and in use,
 - key management includes creating, distributing, storing, making recovery and revoking keys,
 - where encryption keys are stored can affect the overall risk of the data,
- obscuring data in the cloud by **masking, obfuscation, anonymization**,
- **monitor** data-related activities, detect unusual events,
- to better collect, manage, analyze, and display log data, use SIEM.

Key management

no

- **Challenges:**

- access to the keys
 - replication/backup of the keys
 - key storage
 - key lifecycle management
 - problematic integration of key store with other systems
- no keys = no data
 - keys should never leave trusted environment
 - can be internally/externally managed, or managed by a third party
 - public cloud providers offer software-based keystores, which do not comply with NIST requirements (physical security)

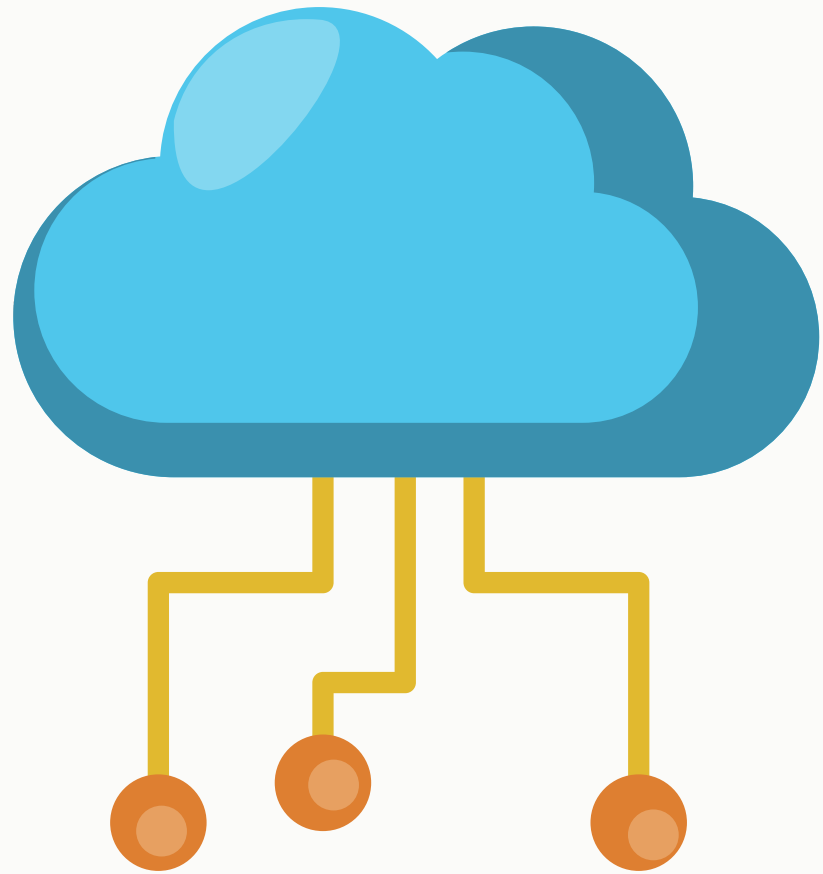
BEST PRACTICE:

- use a centralised key management solution (KMS)
- use strong access controls
- practice regular key rotation
- audit access to the keys
- backup/recovery plan

Cloud Infrastructure Security

Infrastructure security includes:

- network security:
 - rate limiting
 - bandwidth
 - filtering
 - routing
 - detecting security events
- physical/datacenter security
 - geographic considerations
 - natural disaster considerations
 - redundancy options (for electricity, network, cooling)



Private IaaS Cloud

Private cloud - IaaS

Security considerations

- access control: RBAC, MFA
- data encryption
- network security: firewall, IDS, IPS, VPN
- physical security (surveillance, environmental controls)
- regulatory and policy compliance
- active security monitoring
- backup and disaster recovery
- vulnerability management
- automation
- lack of skills and knowledge.

Private cloud - IaaS

Security considerations (2)

- network needs to be segregated: use management network for core cloud services, a data network for storage nodes, APIs network for APIs, an external network for VMs and an internal network for VM-to-VM communication within the same cloud,
- use role-based access control (define API policy per service),
- always keep an off-cloud backup,
- perform regular updates,
- encrypt the data at rest and in transfer,
- automate deployment and configuration of services,
- understand your security weaknesses,
- monitor your network,
- apply BIOS hardening and updates.

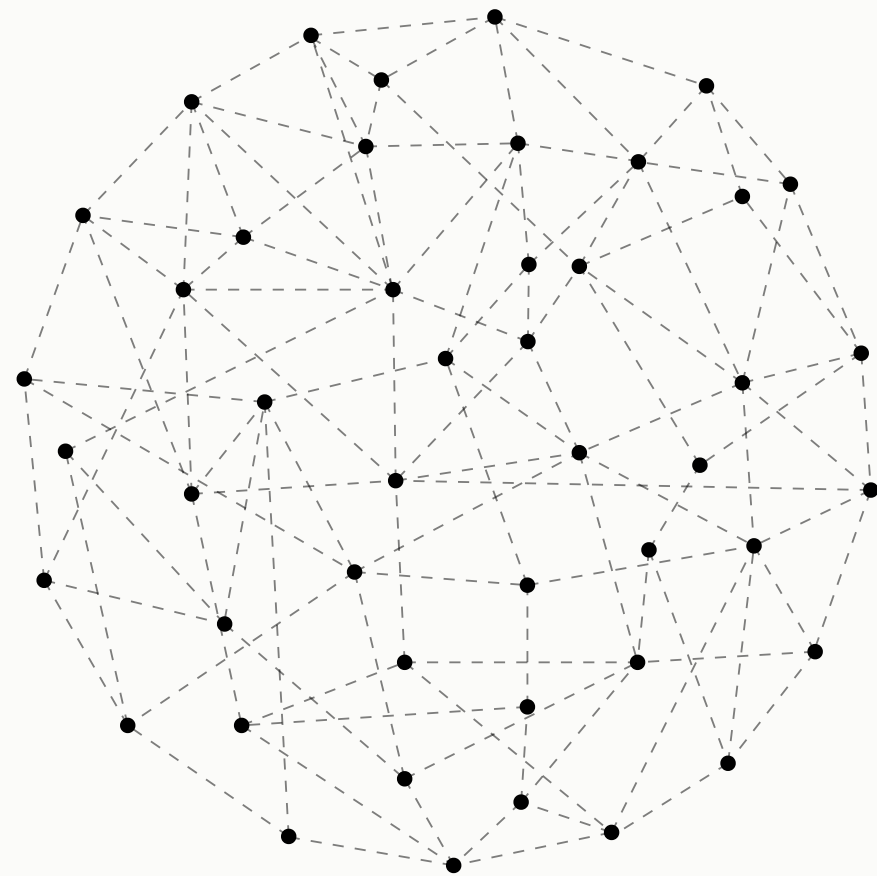
Private cloud - IaaS

Security considerations (3)

- restrict access to IPMI/ILO interfaces, keep them updated, disable default accounts,
- security groups are not the same as ACLs, set allowed traffic per VM and name the security groups in such a way that the rule will already explain what it does (e.g. SMTP_IMAP_ACCESS_TO_NETWORK_ABC),
- all inbound connections should use TLS, all service-to-service connections should use TLS,
- access to management nodes must be restricted (from certain IPs, by VPN, bastion host ..),
- use bastion or jumphost for SSH and RDP connections + VPN,
- outbound filtering with egress,
- use rate limiting on incoming connections.

Private cloud network security

Here the network design is in the hands of the organisation. It needs to be carefully planned.

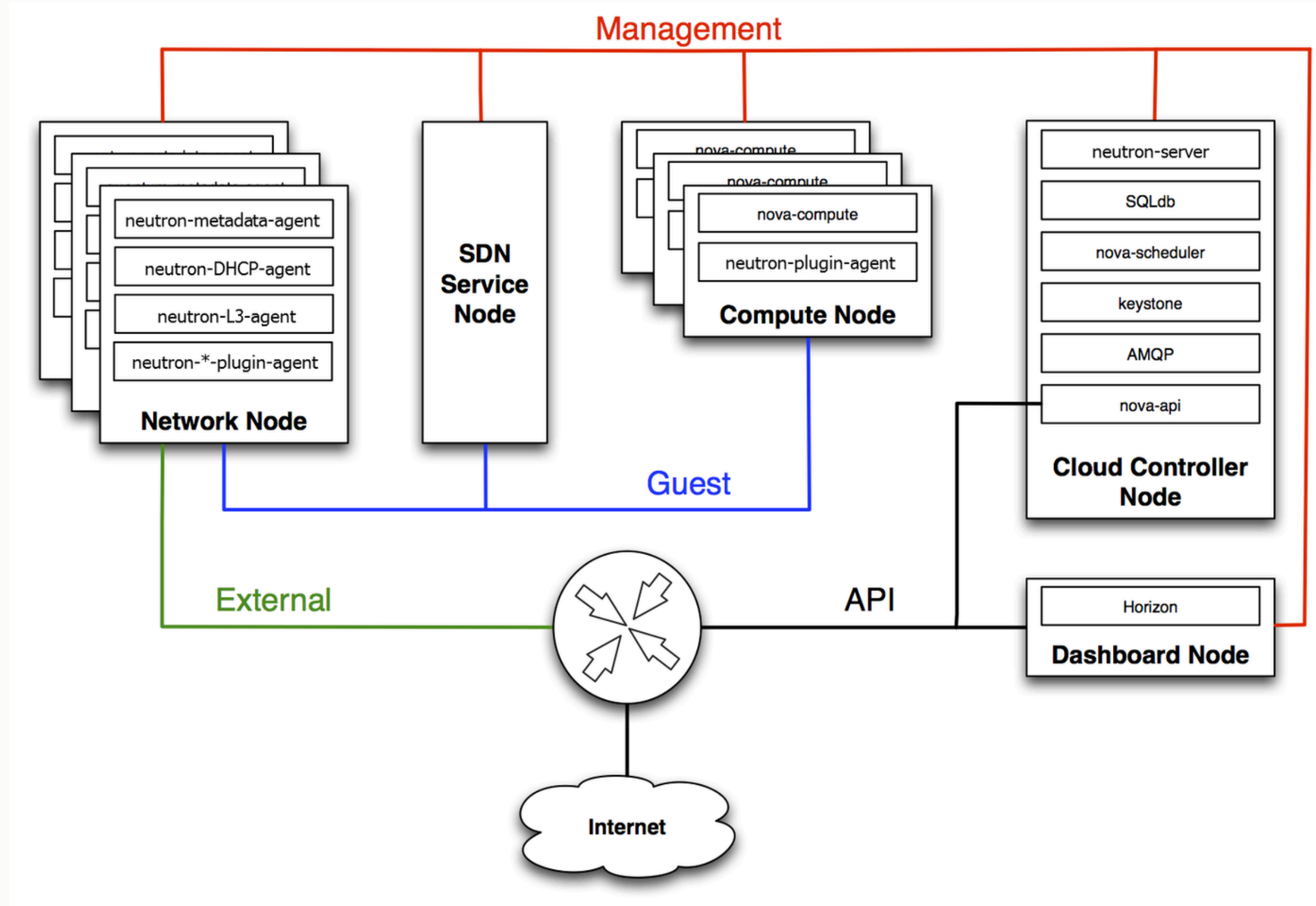


OPENSTACK NETWORK SEGMENTATION

OpenStack uses SDN, which complicates the design of the physical and virtual networks. There are typically 4 types of networks in OpenStack:

- **API network:** used to access APIs, accessible by anyone from the internet
- **Management network:** used for communication between the OpenStack components, traffic is typically not routed in or out of this network. (databases)
- **Guest/tenant network:** Used for VM data communication within the cloud deployment.
- **External/public network:** reachable from the Internet.

OpenStack network topology



Node provisioning

- **automated deployment** and **configuration management** systems are key (consistency, less chance for human errors, faster deployment, documented procedure)
- use a separate network for PXE,
- verify boot process (secure boot),
- **use multitenant network,**
- apply security measures already in the provisioning process.

APIs

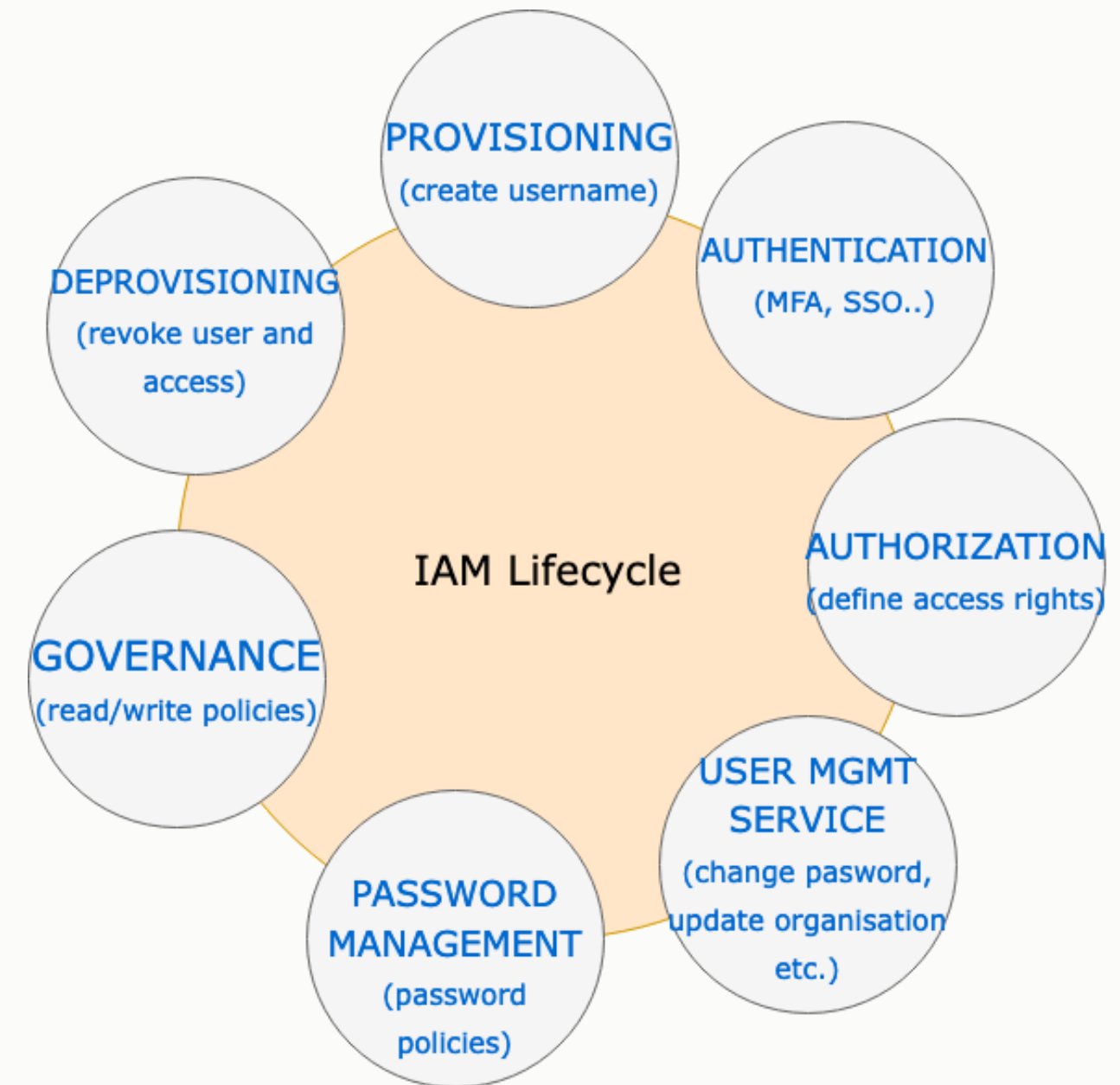
- APIs are the point of contact with the outside world, so a firewall as a security control is not enough
- **usernames, passwords and tokens should never appear in the URL, use proxy caching and logging, data encryption**
- isolate API endpoints on separate hosts
- use SELinux
- use host-based firewall rules
- use network ACLs and IDS
- use two-factor authentication where possible
- use a **reverse proxy** for REST API endpoints
- use **WAF** (Web application firewall to protect your APIs against common web exploits)
- OpenStack has private and public APIs, by default they are all public, but they don't have to be. Specify allowed API operations

Compute node security (Nova Compute)

- don't store credentials in plain text files (such as OS USERNAME, OS PASSWORD in OpenStack - better to use OpenStack CLI on a separate host),
- use tokens and limit their validity as much as possible (default in OpenStack is 1 hour, in older releases it was 24 hours),
- disable bash history,
- store all logs remotely,
- disable PCI passthrough,
- disable memory optimization as it uses memory page deduplication,
- use TLS for Spice/VNC sessions,
- don't keep VM logs that could contain any sensitive data from the customer.

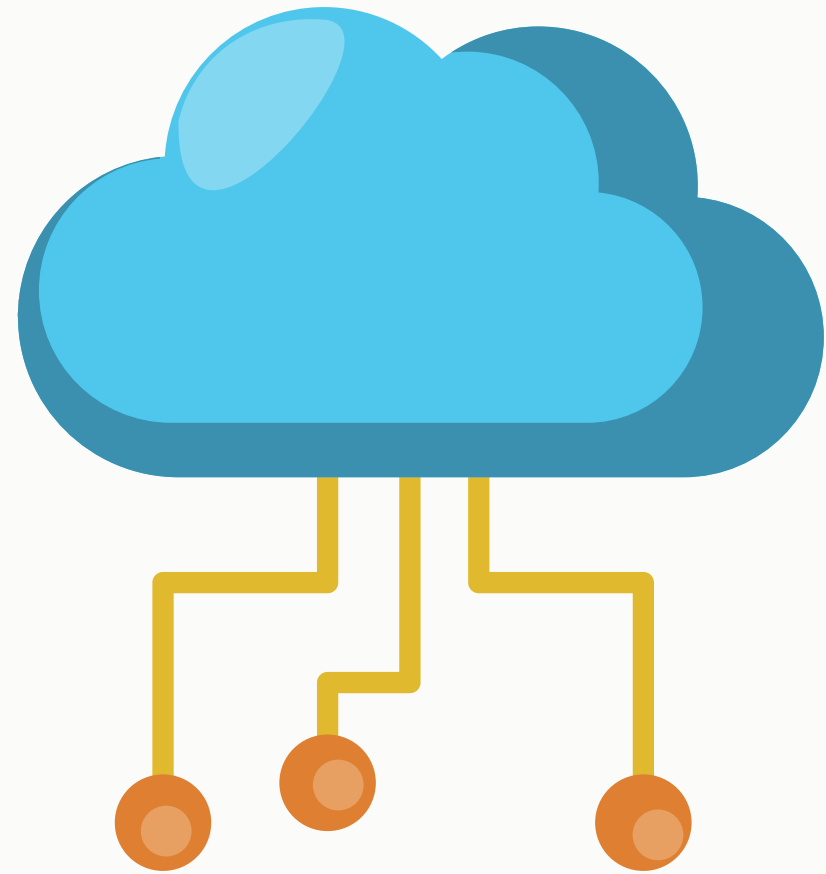
Authentication and access management

- authentication (identity) vs authorization (access)
- **traditional IT:** getting identity means obtaining email, VPN, and access to services
- **cloud:** deleting identity means not being able to login, how about access?
- services provide long-lived authentication tokens that exist even when identity is gone.



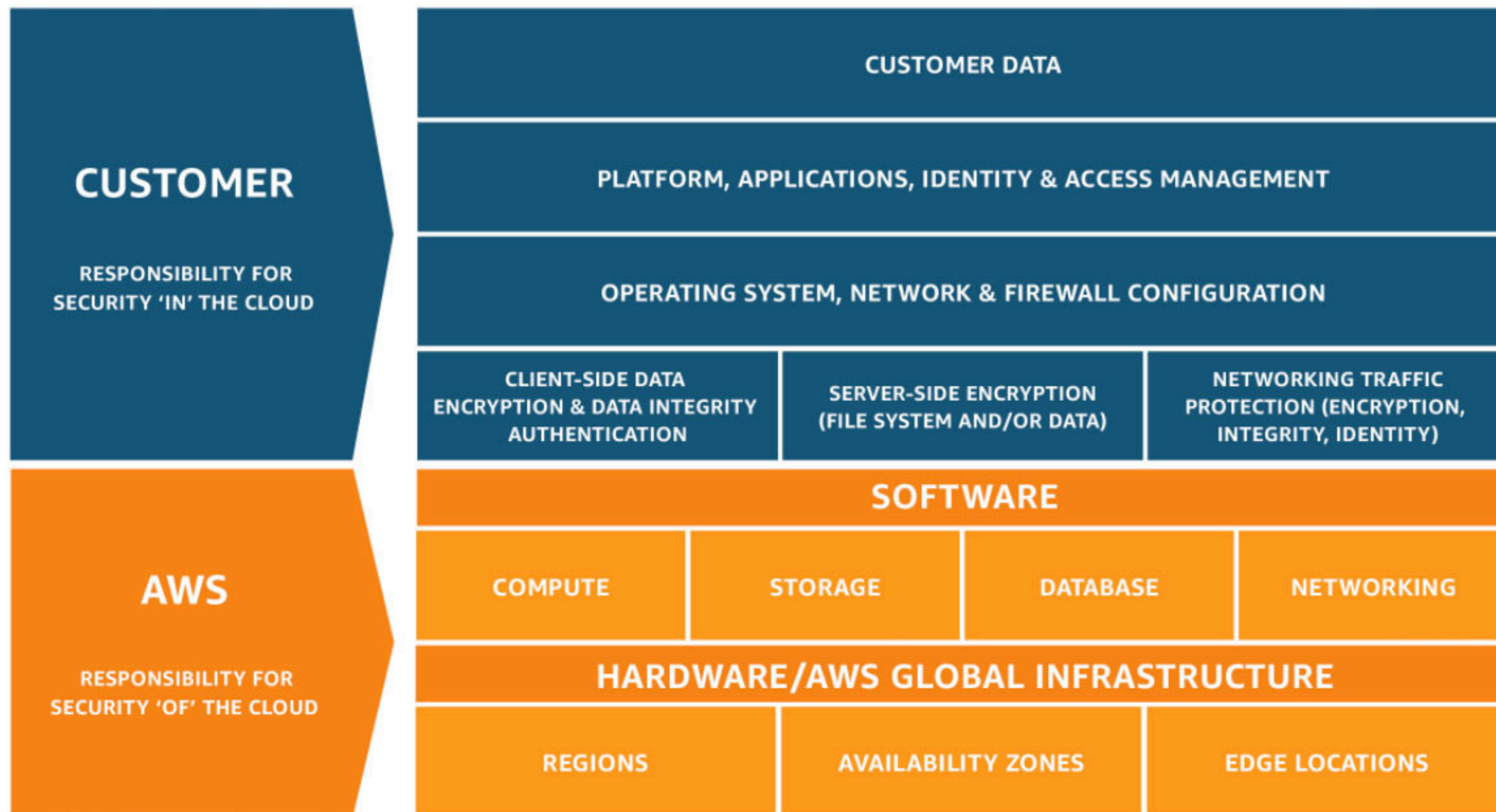
How to prevent common attacks?

- **Spoofting (faking someone's identity):** use SSH keys for authentication, TLS for communication, strong password policy, link Keystone with LDAP directory, MFA
- **Tampering (intentional change/corruption of data):** use digital signatures for data integrity (Glance supports image signing), mandatory access control (MAC) and role-based access control (RBAC) to protect services
- **Repudation (denial of responsibility for an activity by a user or system) :** central logging and auditing in place, SIEM, monitor networks of anomalies (IDS/IPS)
- **Data disclosure:** use encryption, MAC/RBAC (role-based access control)
- **DoS:** redundant services (HA), use quotas per domain/project/user, isolate services from direct access, use proxy to access services from DMZ, good network design
- **Escalation of privileges:** MFA, restrict API, monitor



Public cloud

Security is shared responsibility



Infrastructure security in public cloud

SECURITY CONSIDERATIONS

- **Regions** (physical location of the clusters)
- **Availability zones** (isolated areas within each region - independent power, cooling, and physical security)
- protecting **network**: zero trust approach (network and account boundaries)
- network connectivity includes thousands of VPCs, accounts, and on-premises networks (AWS Transit Gateway - acts as a hub that controls how traffic is routed among all the connected networks)
- **automated creation of resources** (Terraform/CloudFormation/Bicep) - the concept of Infrastructure as a Code
- define system security configuration and maintenance (**hardening**, minimization and patching)
- operating system **authentication and authorizations** (for example, users, keys, and access levels)

Public cloud security tools

- each public provider has its own set of security tools,
- even if tools are available, configuration is not trivial,
- hardening the cloud resources comes with a price.

BUT:

- being a constant target of multiple attacks, public clouds are generally very secure,
- monitoring of events is already integrated into the cloud provider's tools.

AWS SECURITY

- access policy: avoid using root, use MFA, disable credentials that are unused for 90 days or less, ensure access keys rotation every 90 days, IAM password policy (same as for any OS)
- automate building resources and their configuration
- use multilayered network
- perform regular backups
- **AWS has a lot of security tools:**
 - **AWS Config:** assess, and audit of configuration of AWS resources
 - **AWS CloudTrail:** checks changes, records AWS API calls
 - **AWS Config:** configuration management of supported AWS resources in your account
 - **AWS CloudWatch:** applications monitoring (visualisation of metrics, logs, events)
 - **AWS Guard Duty:** threat detection system
 - **AWS Shield:** DDos protection
 - **AWS Identity Manager** and **AWS organisations**

AWS IAM

- **users, groups, roles configuration,**
- **allow/deny policies** associated with a specific service,
- cross-account access (we create an account and a policy that allows users to get access to services in other accounts)
- **temporary credentials** (e.g. to access a few objects in S3 bucket),
- identity federation (identity broker that works as an intermediary between corporate users and AWS services to grant the authentication and authorization process without creating the users in IAM)
- Other security options:
 - **Certificate manager**
 - **WAF**
 - **Application Load Balancer**
 - **HSM** = hardware encryption
 - **Amazon Macie** - ML-based security component (automatic discovery, classification and access monitoring of sensitive data)
 - **Amazon Cognito** - SSO for web applications (enables Google/FB/.. IdP)

Why is MFA ineffective?

- legacy protocols are still used, such as FTP, POP3,
 - MFA is not requested for all users (exceptions for privileged users),
 - least privileged access is therefore not implemented sufficiently,
 - using privileged accounts for daily operations increases the attack surface,
 - access to IdP is not restricted,
 - problem of token/cookie theft,
 - secrets are not handled properly (credentials found in code (Github), in documentation),
 - no encryption for transfers means travelling in plain text.
- Solution is IAM and no local accounts, centrally managed users and accounts, with short-lived credentials and access based on least privilege.
 - Infrastructure as a code - automated creation of resources and orchestration (e.g. OpenTofu, Terraform..)
 - To check IAM solutions of public cloud providers, follow this link: <https://julian-wieg.medium.com/visualizing-multi-cloud-iam-concepts-63525967c0a7>

Why is monitoring ineffective?

- not all native signals are monitored,
- data in SIEM is limited (cloud providers usually set a limit for the maximum number of events per second)
- log retention is very short,
- auditing is not enabled (in SaaS it is usually enabled to some extent by default)
- log visualisation makes sense if it is for a longer period of time, where anomalies are more apparent (logs are limited, storage is a cloud asset that one needs to pay for)

Bastion host problem

Bastion host allows access to the cloud environment. Needed for SSH and RDP connections.

- host hardening should be in place,
- often insecure protocols are exposed to the internet (RDP),
- often protocols are poorly configured (e.g. SSH should have account lockout after multiple successive failed attempts, root login should be disabled, and password login as well),
- access to the host should be limited.

- Direct access to the public cloud is replaced by IAM - so no more accounts on the VMs, but short-term credentials to VMs via IAM.
- Users should be limited on the account level - e.g. AWS Service Control Policies.

Public cloud network security

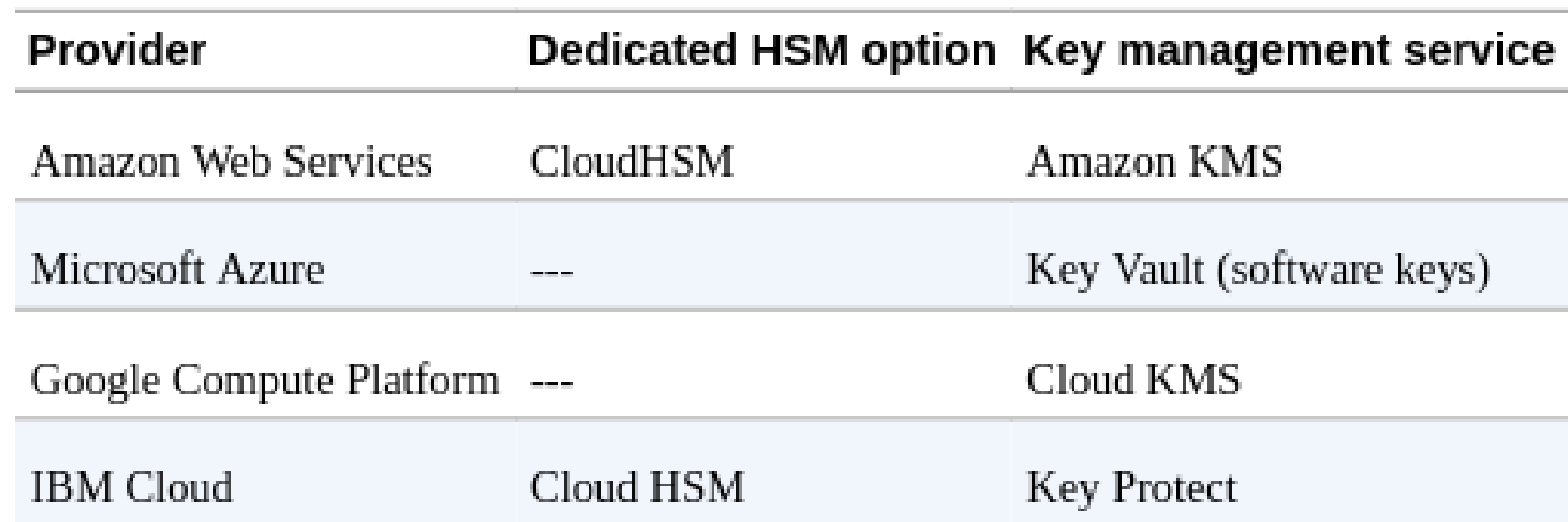
- **new infrastructure can be instantly added by any person or system with no expert skills,**
- ease of deployment and high rate of change make it very difficult to maintain overall control over the cloud environment (autoscaling, serverless computers),
- customer shares responsibility with the provider for securing the network,
- **baseline: educating dev-ops and users, establishing who is responsible for which aspect of security, the use of CIS benchmarks, incident response plan,**
- use tools to monitor and detect vulnerabilities and misconfigurations in cloud networks,
- use SIEM or threat detection solution,
- access should be limited with web application firewall, network ACLs, DoS protection.

APIs in public cloud

- in AWS Cloud all APIs are public by default
- **Service Control Policies: where you make limitations for APIs,**
 - whitelists,
 - limits to regions
 - etc.
- **AWS API Gateway** provides a number of ways to protect APIs from certain threats, like malicious users or bursts in traffic.
- You can protect your **API by generating SSL** certificates, configuring a **WAF**, setting throttling targets, and only allowing access to your API from a Virtual Private Cloud (VPC).

Data encryption and key management

- some public cloud providers provide **HSM** = hardware security model, which is expensive, or **KSM** = key management service) that is connected to an external HSM



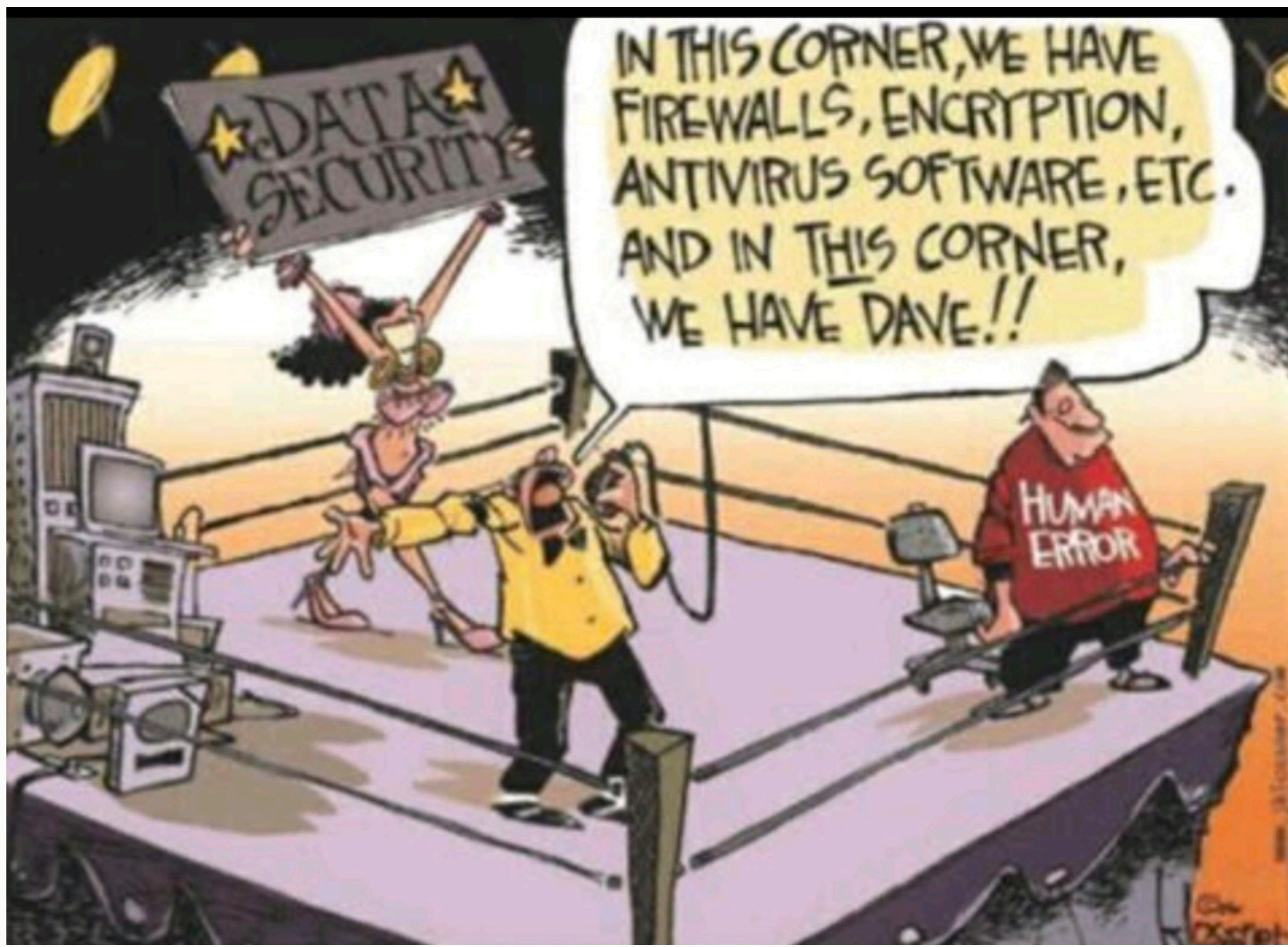
Provider	Dedicated HSM option	Key management service
Amazon Web Services	CloudHSM	Amazon KMS
Microsoft Azure	---	Key Vault (software keys)
Google Compute Platform	---	Cloud KMS
IBM Cloud	Cloud HSM	Key Protect

That was in 2021

In 2025

Today all public cloud providers offer HSM option for key management.

Data security



Public cloud asset security

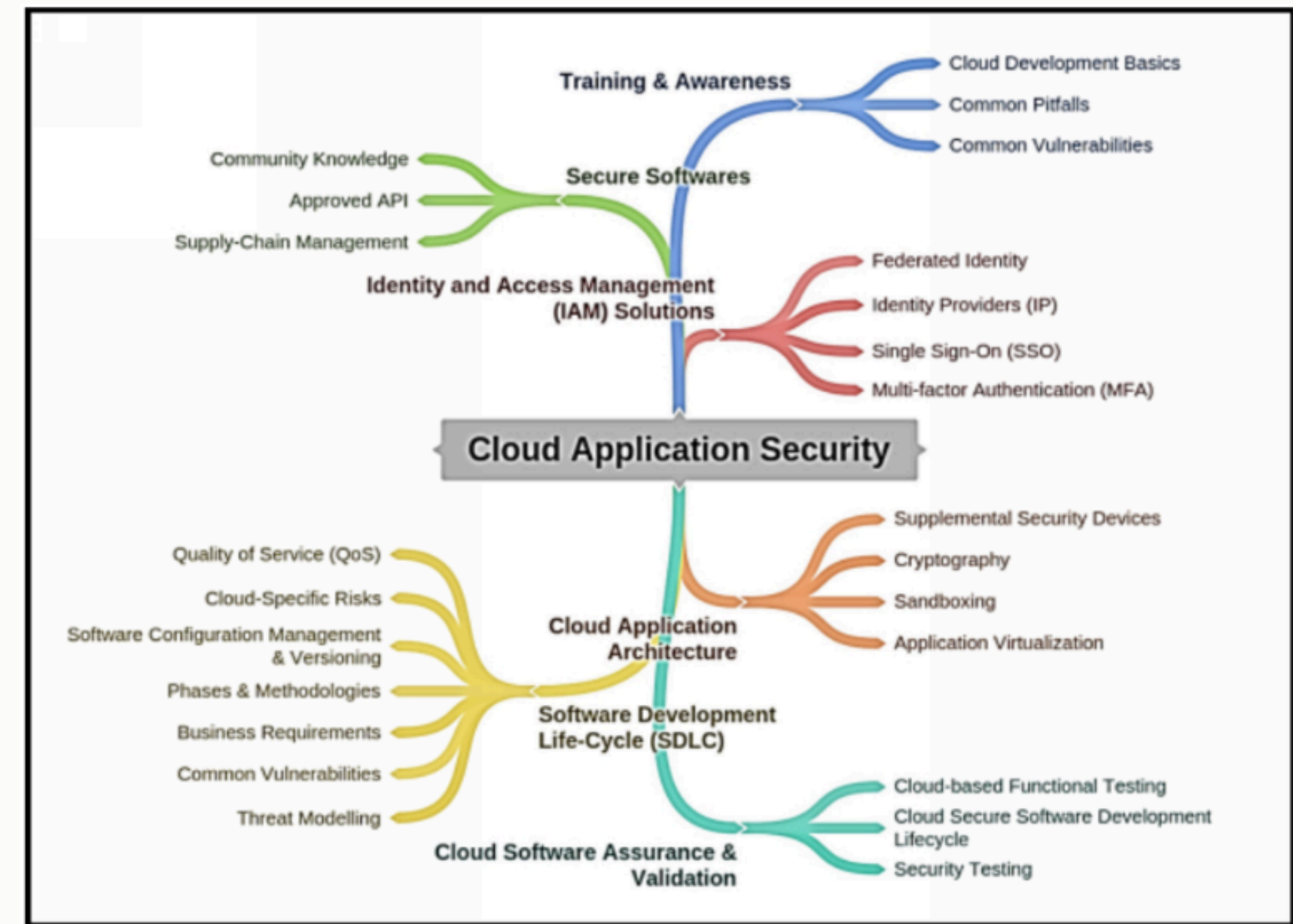
- assets in the cloud: VM, container, storage assets, images, network assets (Virtual Private Clouds (VPS), Content Delivery Network (CDN), DNS, load balancers, proxies etc.)
- each provider has its own provisioning method, they charge the provisioned assets
- due to the complexity of the cloud environment and lack of skills, provisioning should be done automatically - less opportunity for human error, the shift left approach can be used to apply security controls.
- **How to protect assets?**
 - Strong authentication and access control,
 - automated deployment with unit tests,
 - vulnerability scan, code tests etc.,
 - with continuous monitoring,
 - by placing the services into different network segments and
 - by having a disaster recovery plan in place.

How to assess a public cloud provider?

- does it have security certifications?
- does it comply with relevant data protection regulations and laws?
- evaluate the security controls they have in place, such as encryption, access controls, and authentication mechanisms,
- check incident response capabilities, read the SLAs and check how security is handled,
- check references, customer feedback, audit reports.

Cloud Application Security

- Cloud REST vs SOAP APIs
- application might not behave the same on-premises and in the cloud
- multi-tenancy problem and third-party administrators
- required security in deployment, development
- injection flaws are common
- broken authentication
- sensitive data exposure
- broken access
- using dependencies that are vulnerable
- security testing necessary

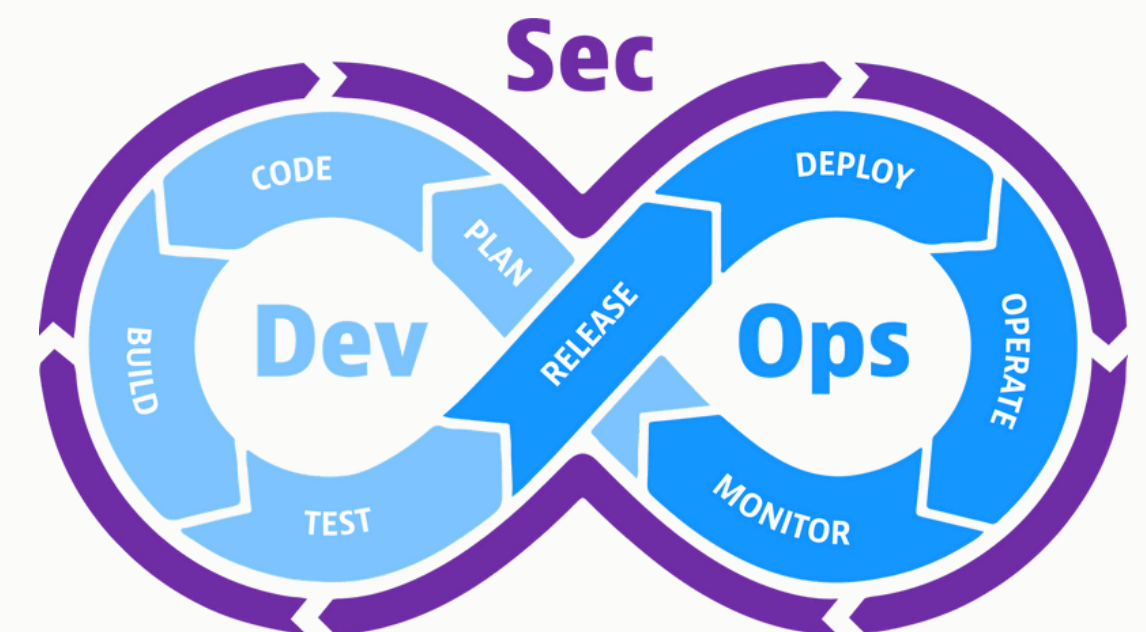


Source: CCSP workbook

DEV-SEC-OPS

Shift-left approach to dev-sec-ops focuses on security measures in all steps of the development cycle, to ensure security already before the release of the software or launch of a cloud service. In other words, testing is moved to the left on the project timeline.

- provide unit tests (for cross-browser testing, APIs, integration)
- optimise test environments (use temporary environments),
- provide functional tests,
- check the quality of code,
- comply with OWASP security standards on secure coding,
- apply automatic vulnerability scanning,
- provide documentation and check it with CI
- provide SBOM (software bill of materials)



DEV-SEC-OPS TOOLS

- style checks (code, commits etc.),
- **code quality and security testing** (pentesting),
- compliance checks
- **SAST** - Static Application System Testing = scanning source code for vulnerabilities,
- **DAST** - Dynamic Application System Testing = black-boxing testing, application is exposed to different attacks,
- **IAST** - Interactive Application System Testing, hybrid of SAST and DAST, analyses applications with manual and automatic tests,
- **RASP** - Runtime Application Self Protection, software integrated to application to prevent attacks during runtime (blocking traffic for example) - tools like OpenRASP, Sqreen,
- **secret detection** - scan code for secrets,
- dependency scanning for security issues - tools like WhiteSource, FOSSA,
- application hardening (**WAF**),
- security monitoring.

Example

Static	Test	Package	Deploy	E2e
✔ lint:golang	✔ test:frontend:activation-portal:unit	✔ pkg:version	✔ preview:deploy:activation-portal	✔ preview:test:e2e:activation-portal
✔ lint:hadolint	✔ test:frontend:admin:unit	✔ preview:pkg:backend 9	✔ preview:deploy:api&seed	✔ preview:test:e2e:admin:batch 4
✔ lint:openapi	✔ test:frontend:aws:unit	✔ preview:pkg:default-backend	✔ preview:deploy:backend 6	✔ preview:test:e2e:aws
✔ lint:yaml	✔ test:frontend:azure:unit	✔ preview:pkg:frontend 6	✔ preview:deploy:cronjobs	✔ preview:test:e2e:enterprise-portal
✔ test:docker:compose	✔ test:frontend:enterprise-portal:unit	✔ preview:pkg:kpi-reporter	✔ preview:deploy:database	✔ preview:test:e2e:onboard
✔ test:e2e:activation-portal:static	✔ test:frontend:lib:unit	✔ preview:pkg:queue-manager	✔ preview:deploy:enterprise-portal	✔ preview:test:e2e:system-tools
✔ test:e2e:admin:static	✔ test:frontend:onboard:unit	✔ preview:pkg:seed:migrate 2	✔ preview:deploy:frontend 5	
✔ test:e2e:aws:static	✔ test:frontend:system-tools:unit	✔ preview:pkg:system-tools 3	✔ preview:deploy:mail	
✔ test:e2e:enterprise-portal:static	✔ test:golang		✔ preview:deploy:misc	
✔ test:e2e:onboard:static			✔ preview:deploy:system-tools 4	
✔ test:e2e:system-tools:static			✔ preview:deploy:unleash	
✔ test:frontend:activation-portal:static			✔ preview:queue:setup	
✔ test:frontend:admin:static				
✔ test:frontend:aws:static				
✔ test:frontend:azure:static				
✔ test:frontend:enterprise-portal:static				
✔ test:frontend:lib:static				
✔ test:frontend:onboard:static				
✔ test:frontend:system-tools:static				

Cloud application threats

- software vulnerabilities
- SQL injections
- Cross-site Request Forgery
- Buffer Overflow
- Data breach or data loss
- Insecure APIs
- DoS and DDoS
- Account Hijacking
- Malicious insider
- Spoofing
- Tampering of data
- Malware, ransomware
- Misconfiguration
- Supply chain attacks

Trends

- zero trust is the security model of the present and the future
- security in production and development process (DevSecOps)
- Automated security and AI integration
- Increase in using CSMP tools
- Growing IAM complexity
- Multi-cloud security solutions
- Privacy and confidential computing

Quiz

1. Who's responsibility is the network security in IaaS cloud?
2. What is the difference between virtualisation and cloud?
3. Which is one of the main problems with encryption?
4. Why is MFA often ineffective?
5. How can we protect data in the cloud?
6. What is a security concern when considering implementing software-defined networking (SDN)?
 - A. It has a decentralized architecture.
 - B. It increases the attack footprint.
 - C. It uses open source protocols.
 - D. It is cloud based.
7. Which security measures can prevent privilege escalation in the cloud?
8. How to protect development process?
9. Is the following true or false:
 - Obfuscation: The deforming of code to such a degree that even if the source code is obtained, it is not easily decipherable.
 - Masking: A weak form of confidentiality assurance that replaces the original information with asterisks or X's.
 - Data classification entails analyzing the data that the organization retains, determining its importance and value, and then assigning it to a category.