CompTIA SECURITY+

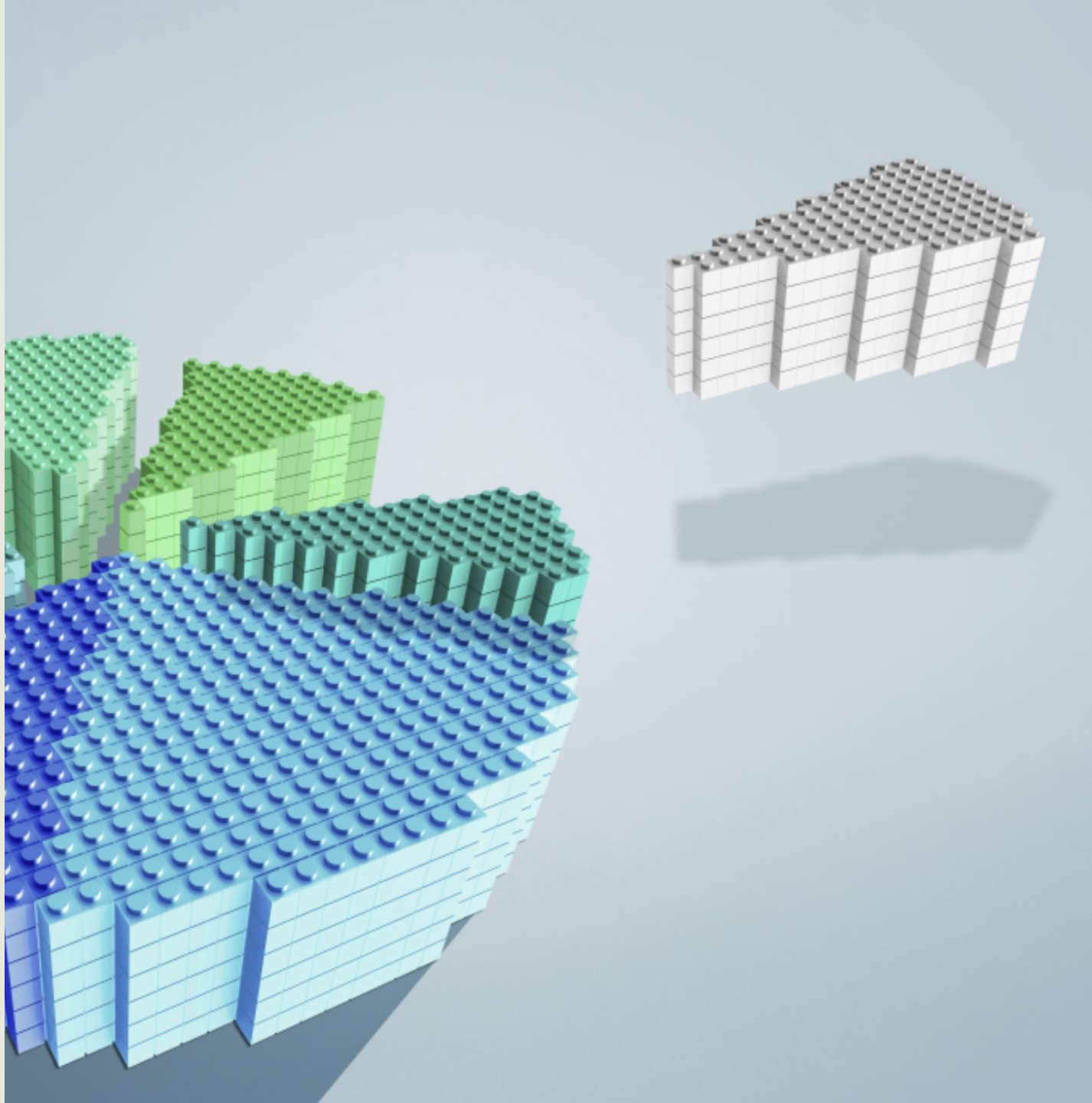
MITIGATION TECHNIQUES

Objectives

- Examine segmentation, isolation, and access control models
- Compare configuration and patch management
- Explore least privilege and separation of duties
- Look at encryption for access controls, monitoring, and visibility
- Learn about decommissioning and offboarding
- Compare hardening techniques

SEGMENTATION AND ISOLATION

- Segmentation divides a computer network into smaller parts
- The purpose is to improve network performance and security
- Other terms that often mean the same thing are network segregation, network partitioning, and network isolation
- Segmentation and isolation are logically and physically accomplished in network infrastructures using zoning
- Zoning (segmentation) is a logical design approach used to control and restrict access

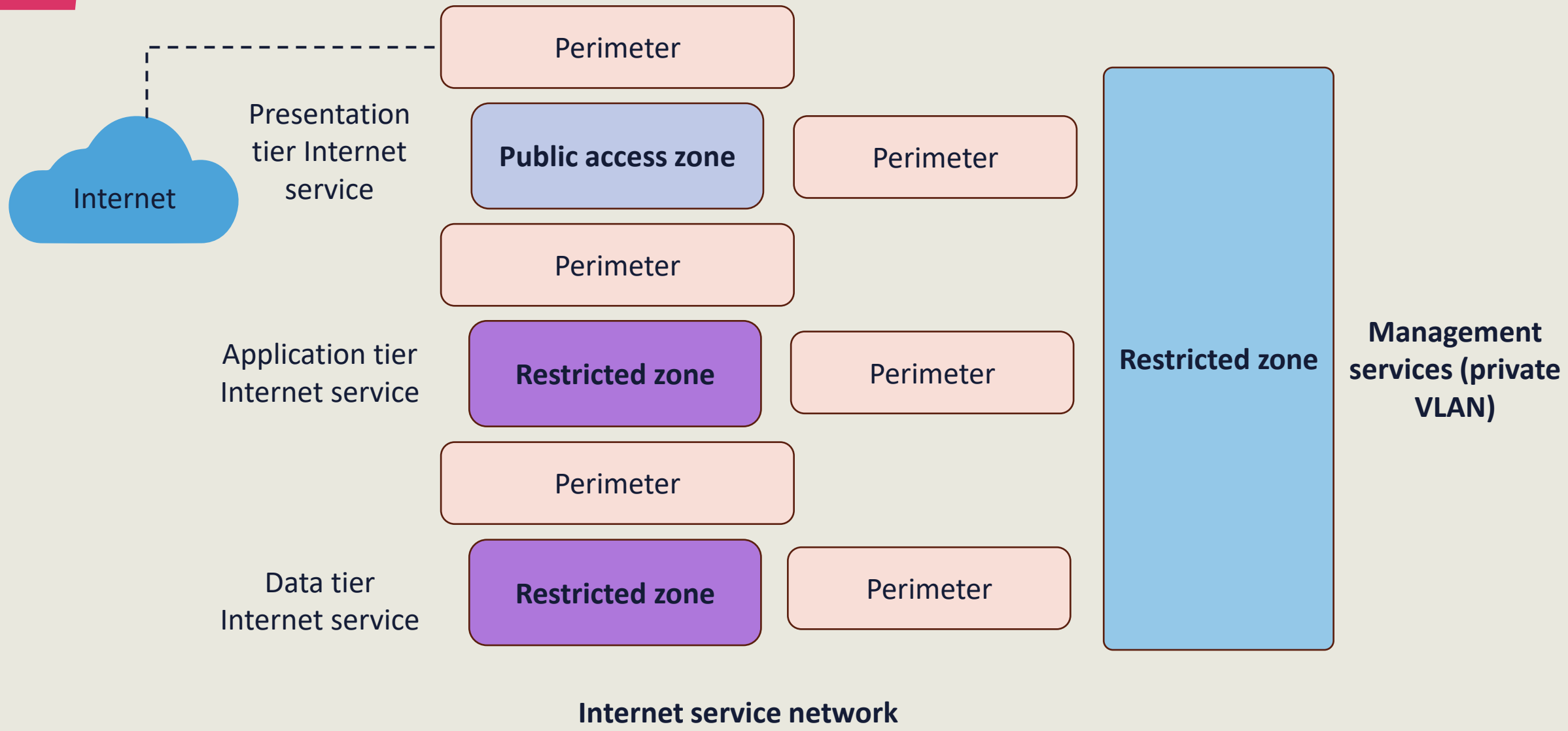


SEGMENTATION AND ISOLATION

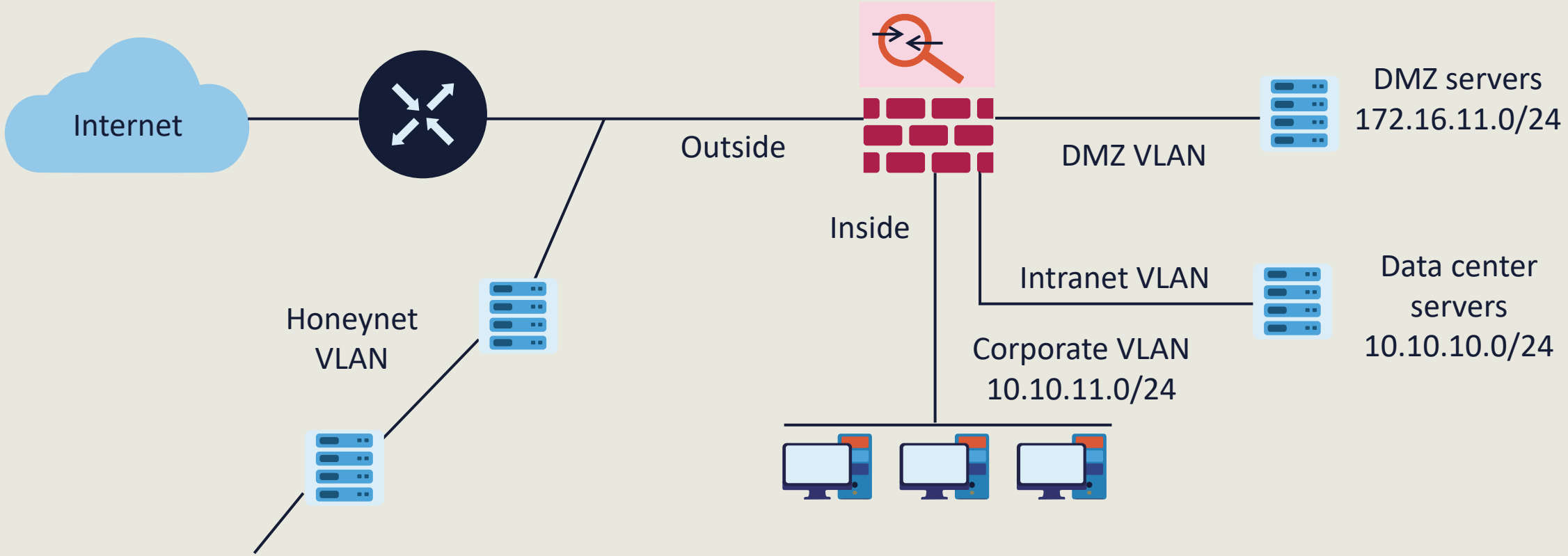
Each zone has fundamental characteristics defined by the security:

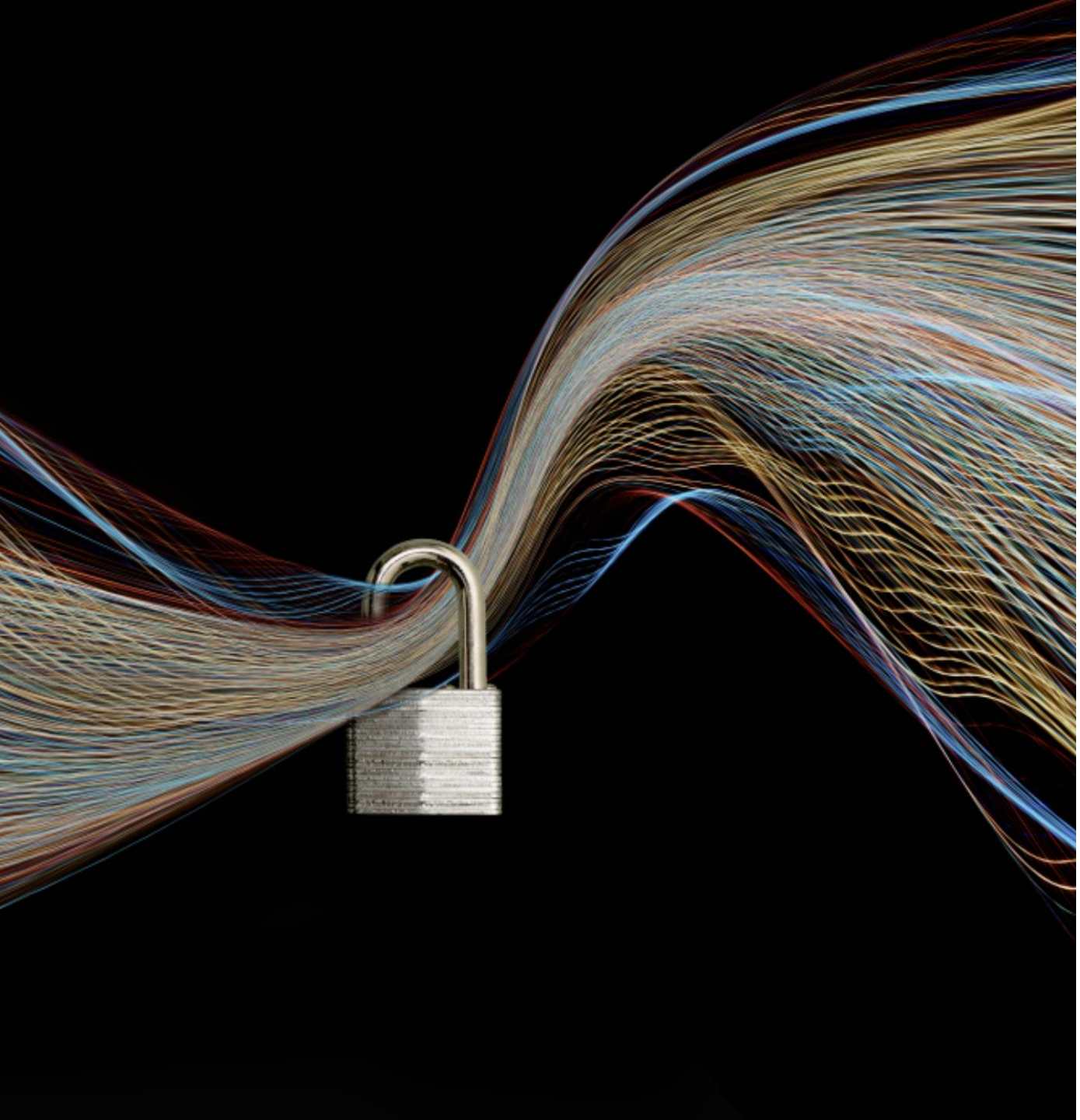
- Every zone contains one or more separate, routable networks
- Every separate, routable network is contained within a single zone
- Every zone connects to another zone via a perimeter that contains zone interface points (ZIPs)
- The only zone that may connect to the public zone is the public access zone (PAZ) (DMZ)

LOGICAL ZONING



PHYSICAL/LOGICAL ZONING



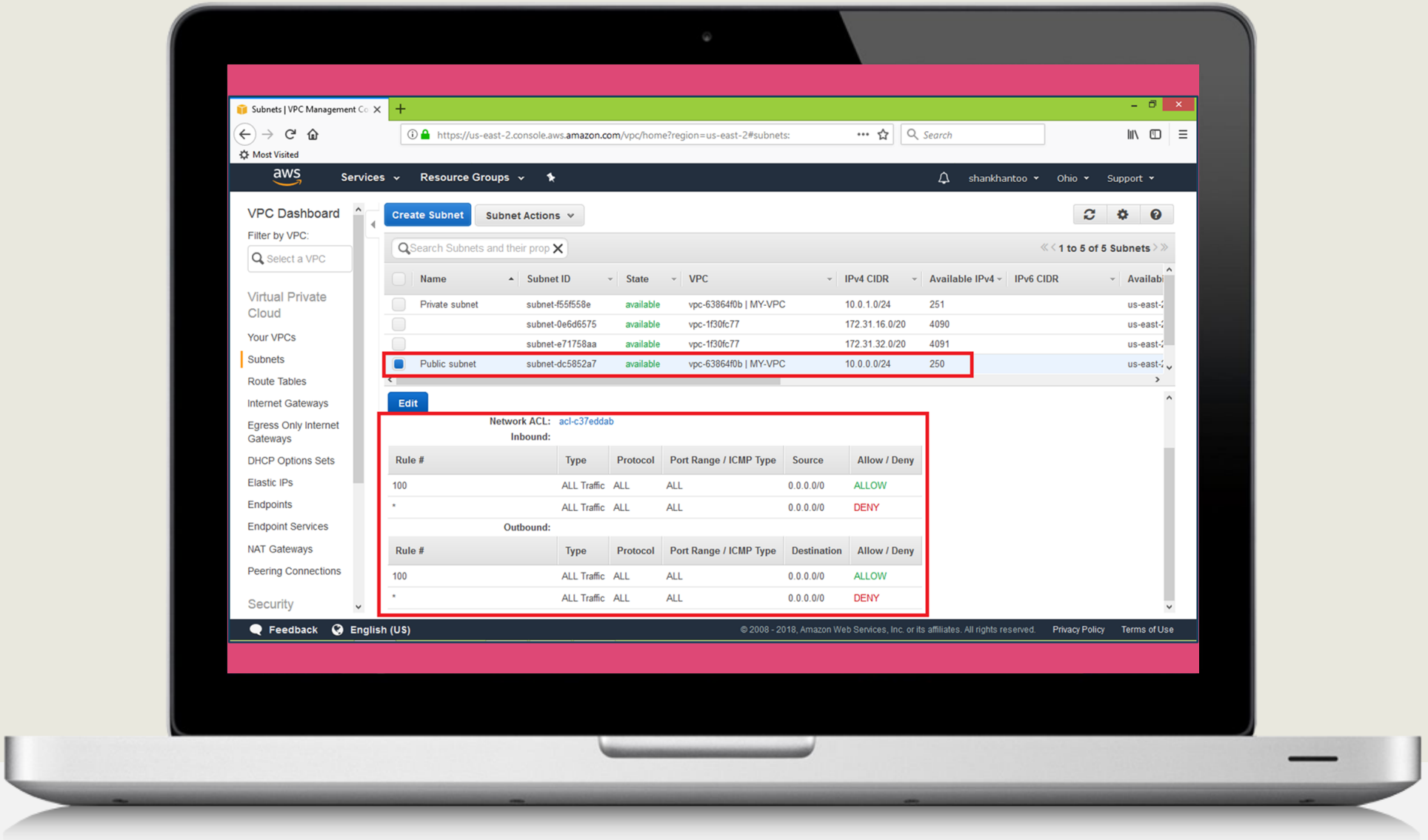


ACCESS CONTROL LISTS (ACLS)

- Allow stateless (static) traffic filtering and management of IPv4 and IPv6 traffic to and from a network interface or virtual local area network (VLAN)
- Contain ordered rules or access control entries (ACEs) to permit (allow) or deny (block) based on Internet Protocol (IP), Transmission Control Protocol (TCP), and User Datagram Protocol (UDP) services and ports, as well as Internet Control Message Protocol (ICMP) messages and codes
- Function as additional infrastructure defense-in-depth mechanisms
- Have an implicit deny-all as the last entry applied if nothing matches

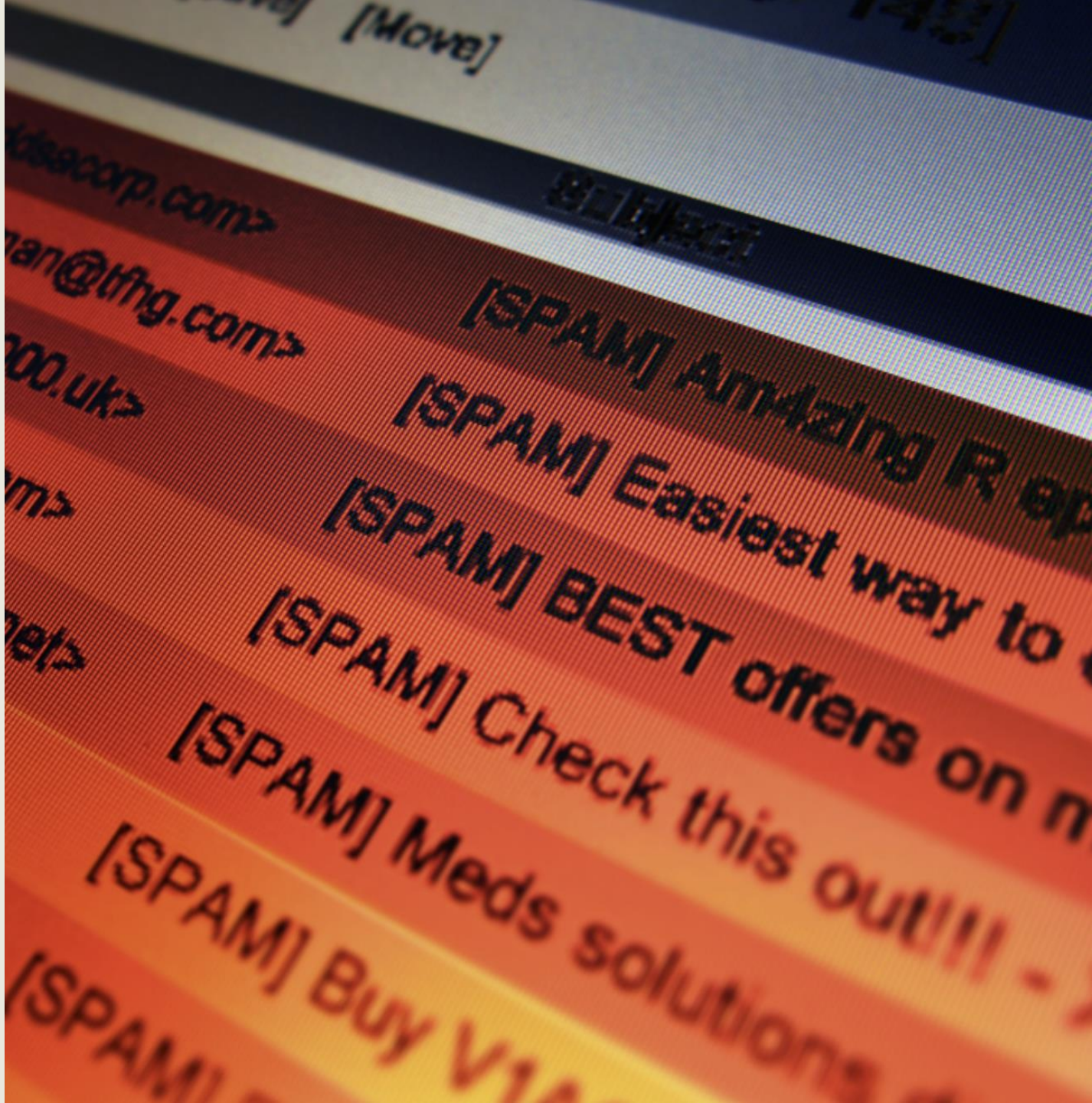
NETWORK ACCESS CONTROL LISTS (NACLS)

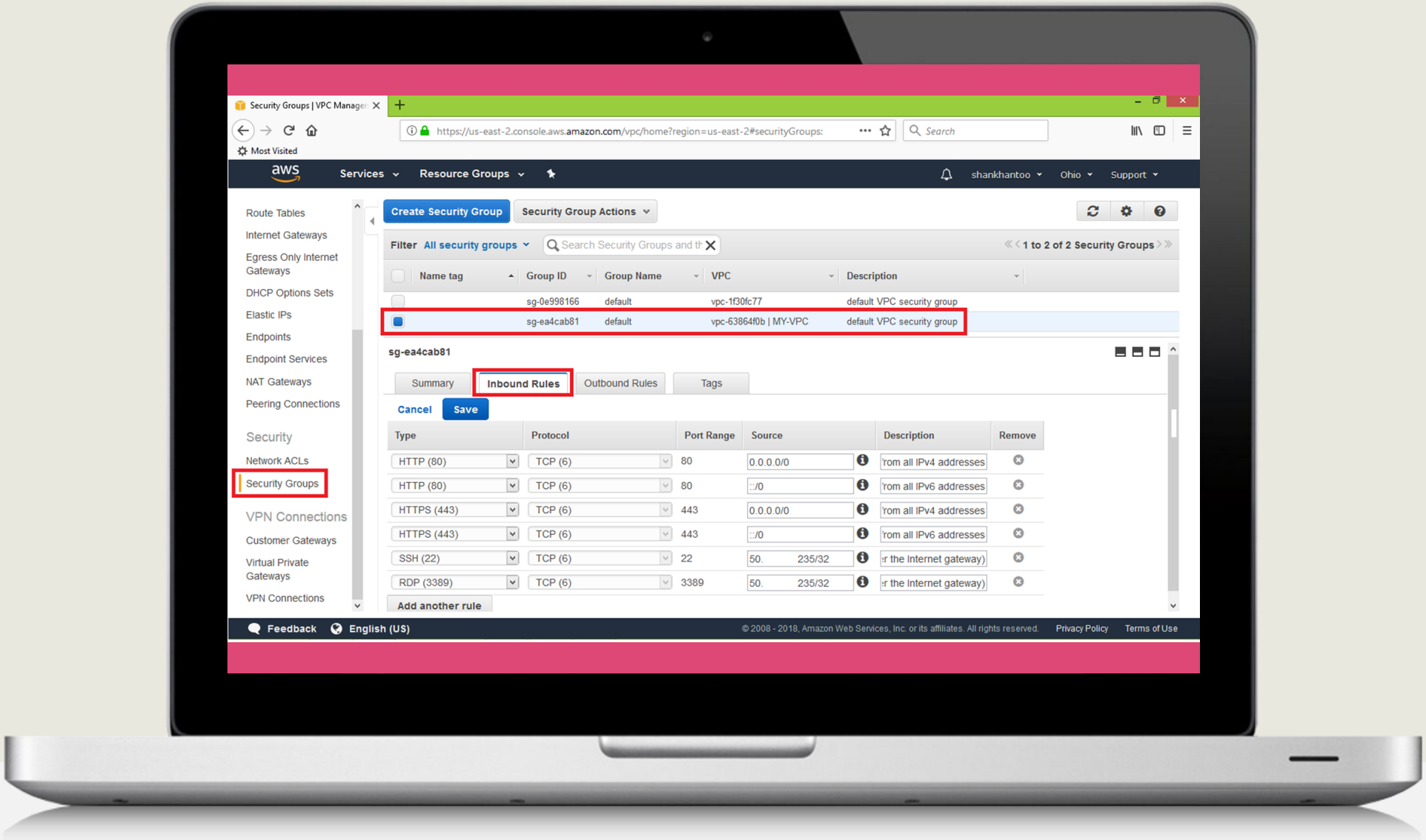
- Are most often static inbound and outbound access lists applied to virtual networks or virtual private clouds
- Apply to all instances, containers, appliances, etc. in the virtual network (VNet)
- Are typically configured with the same techniques as traditional access lists


Subnets | VPC Management Co
https://us-east-2.console.aws.amazon.com/vpc/home?region=us-east-2#subnets:
Search
Most Visited
aws
Services
Resource Groups
shankhantoo
Ohio
Support
VPC Dashboard
Filter by VPC:
Select a VPC
Virtual Private Cloud
Your VPCs
Subnets
Route Tables
Internet Gateways
Egress Only Internet Gateways
DHCP Options Sets
Elastic IPs
Endpoints
Endpoint Services
NAT Gateways
Peering Connections
Security
Create Subnet
Subnet Actions
Search Subnets and their prop
1 to 5 of 5 Subnets
Name
Subnet ID
State
VPC
IPv4 CIDR
Available IPv4
IPv6 CIDR
Private subnet subnet-f55f558e available vpc-63864f0b | MY-VPC 10.0.1.0/24 251
subnet-0e6d6575 available vpc-1f30fc77 172.31.16.0/20 4090
subnet-e71758aa available vpc-1f30fc77 172.31.32.0/20 4091
Public subnet subnet-dc5852a7 available vpc-63864f0b | MY-VPC 10.0.0.0/24 250
Edit
Network ACL: acl-c37eddab
Inbound:
Rule # Type Protocol Port Range / ICMP Type Source Allow / Deny
100 ALL Traffic ALL ALL 0.0.0.0/0 ALLOW
* ALL Traffic ALL ALL 0.0.0.0/0 DENY
Outbound:
Rule # Type Protocol Port Range / ICMP Type Destination Allow / Deny
100 ALL Traffic ALL ALL 0.0.0.0/0 ALLOW
* ALL Traffic ALL ALL 0.0.0.0/0 DENY
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

SECURITY GROUPS

- Are commonly stateful "allow-list" firewalls that apply to layer 3 and layer 4 network traffic
- Can be applied to a virtual load balancer and instance virtual interface:
  - These operate at the hypervisor level attached to the virtual elastic network interfaces (eth0)
- Are called network security groups (NSGs) if applied to an entire virtual network
- Have no explicit deny rules like NACLs, but rather have an implicit deny if nothing matches the "allow-list"
- Evaluate all rules before a decision is made




Security Groups | VPC Manager
https://us-east-2.console.aws.amazon.com/vpc/home?region=us-east-2#securityGroups:
Search
Most Visited
aws
Services
Resource Groups
shankhantoo
Ohio
Support
Route Tables
Internet Gateways
Egress Only Internet Gateways
DHCP Options Sets
Elastic IPs
Endpoints
Endpoint Services
NAT Gateways
Peering Connections
Security
Network ACLs
Security Groups
VPN Connections
Customer Gateways
Virtual Private Gateways
VPN Connections
Create Security Group
Security Group Actions
Filter All security groups
Search Security Groups and th
1 to 2 of 2 Security Groups
Name tag
Group ID
Group Name
VPC
Description
sg-0e998166 default vpc-1f30fc77 default VPC security group
sg-ea4cab81 default vpc-63864f0b | MY-VPC default VPC security group
sg-ea4cab81
Summary
Inbound Rules
Outbound Rules
Tags
Cancel
Save
Type
Protocol
Port Range
Source
Description
Remove
HTTP (80) TCP (6) 80 0.0.0.0/0 rom all IPv4 addresses
HTTP (80) TCP (6) 80 ::/0 rom all IPv6 addresses
HTTPS (443) TCP (6) 443 0.0.0.0/0 rom all IPv4 addresses
HTTPS (443) TCP (6) 443 ::/0 rom all IPv6 addresses
SSH (22) TCP (6) 22 50. 235/32 r the Internet gateway)
RDP (3389) TCP (6) 3389 50. 235/32 r the Internet gateway)
Add another rule
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

PERMISSIONS

- Permissions that principals have can be dictated and enforced by the network operating file system (Linux or Windows) or using a directory service as in:
  - **Read** (r) permission to access the file's contents
  - **Write** (w) permission to modify or change the contents of a file
  - **Execute** (x) permission to execute the contents of a file
- One can change a Linux file and directory permissions with the **chmod** command, which stands for "change mode"

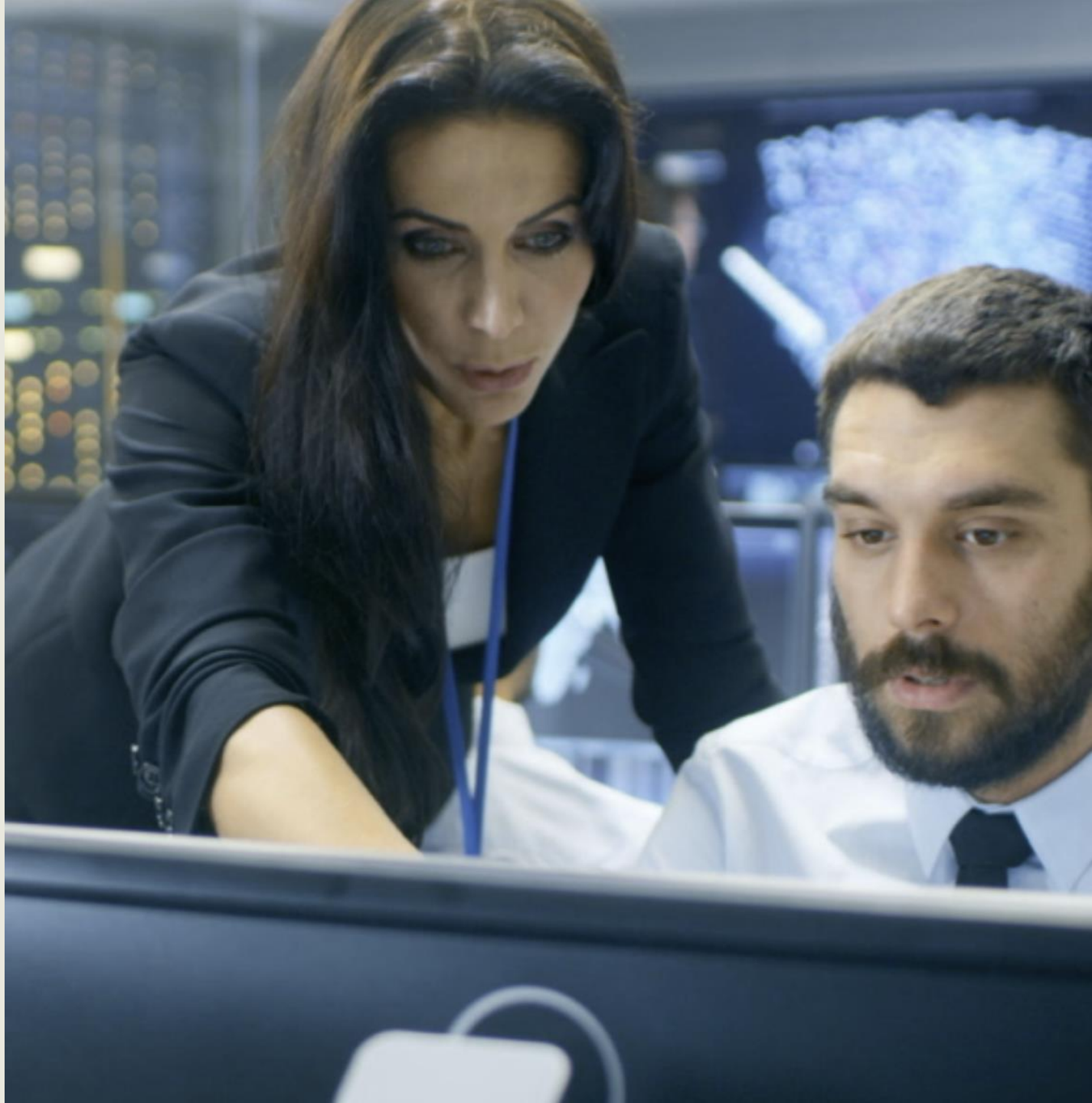
CONFIGURATION MANAGEMENT

- The goal of configuration management (CM) is to ensure that accurate and meaningful information is readily available regarding the configuration of applications and services along with the configuration items (CI) that support them
- It includes all relationships and dependencies between the CIs:
  - Objects include hardware, software, networks, sites, vendors, suppliers, and people

CONFIGURATION MANAGEMENT

CM is a governance and systems life cycle process for ensuring consistency among all assets (configuration items) in an operational environment:

- Classifies and tracks individual CIs
- Documents functional capabilities and interdependencies
- Verifies the effect a change to one configuration item has on other systems



CONFIGURATION MANAGEMENT

CM practices offer the required data about assets and their configurations, including their interactions with other assets, which assists administrators and managers with

- Problem resolution
- Incident response
- Network component deployment
- Strategy formulation
- Budgetary forecasting
- Overall decision-making

REPLACE WITH SOMETHING SIMILAR

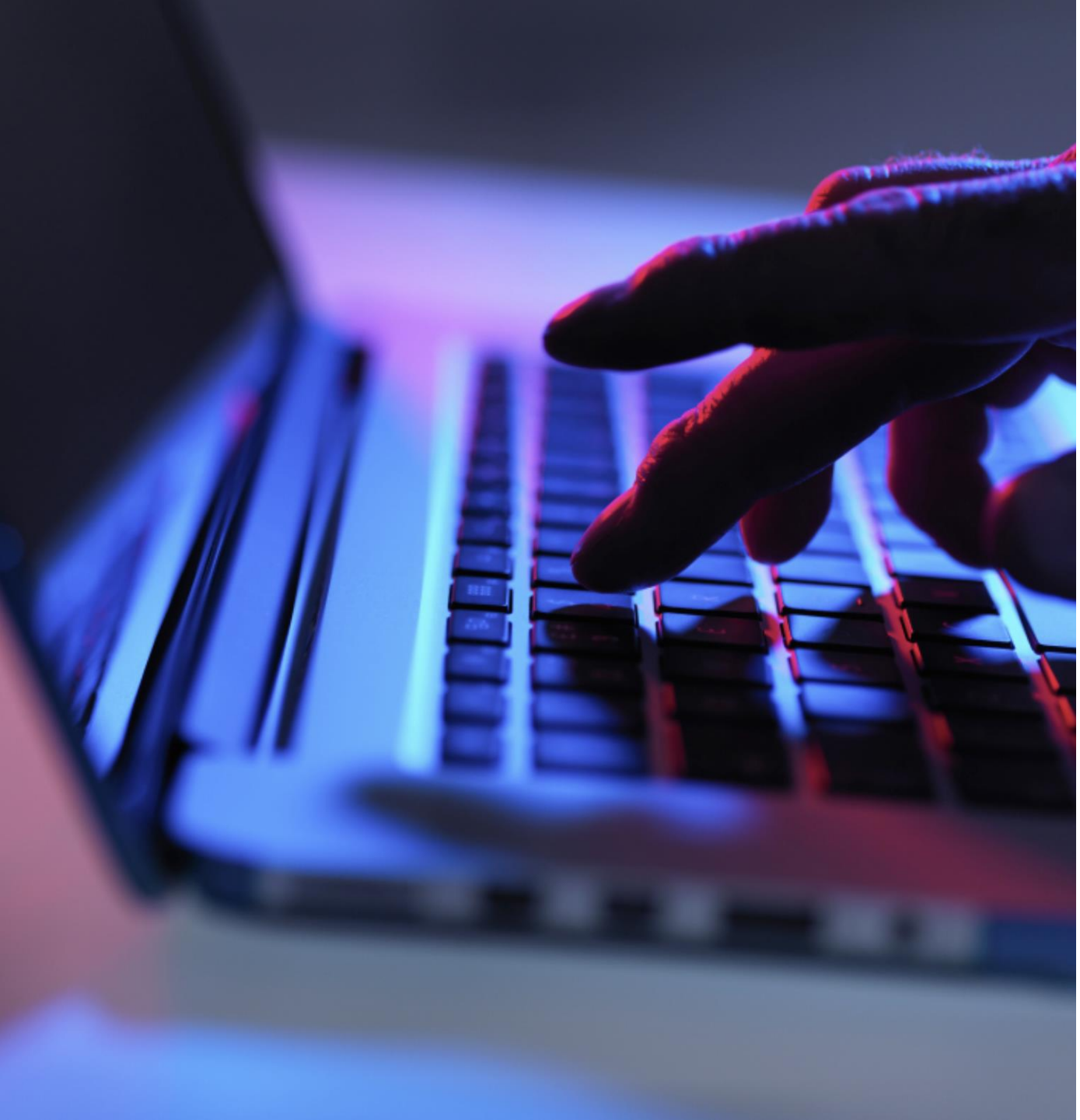
- A configuration management system (CMS) is a set of data, tools, utilities, and processes used to support configuration management
- All information should be tagged and labeled with a common unified schema, preferably using key-value pairs
- This data will populate the configuration management database (CMDB)
- Relational databases have been used historically
- NoSQL/document databases are emerging as a common solution
- A communication service provider (CSP) service such as AWS DynamoDB could be leveraged

PATCH MANAGEMENT

- Patch management is the process of applying (hopefully fully tested) updates to software, drivers, and firmware to protect against vulnerabilities
- Effective patch management helps ensure the best operating performance of systems, boosting productivity
- All systems need to be secured with patches, if possible
- The risks of disregarding patch management can cause exposure of business to leaks and breaches, loss of productivity, and loss of reputation

PATCH MANAGEMENT BENEFITS

- Protects all endpoints from attackers
- Keeps all systems running in an optimized fashion
- Promotes productivity within the organization
- Helps lower the cost of device life cycle maintenance and repair
- Supports laws, regulations, and compliance standards



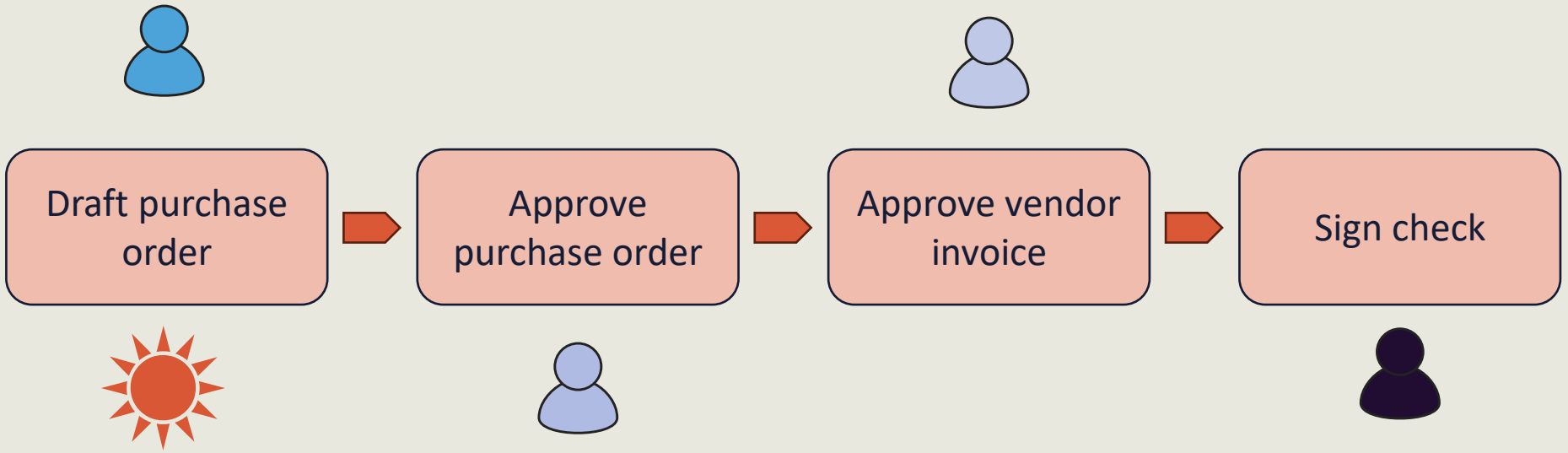
LEAST PRIVILEGE

- Is the principle that users and programs should only have the necessary privileges to complete their tasks, according to the National Institute of Standards and Technology (NIST)
- Is also referred to as "need to know" or staying within one's "pay grade" or classification level
- Is an aspect of authentications, authorization, and accounting (AAA) and identity and access management (IAM) where the subject has just the proper level or number of permissions and rights to perform the job role or responsibility and nothing more:
  - It should be built into all access control architectures
- Any deviation (escalation or elevation), if allowed, should go through an established change control IT service or service desk implementation

SEPARATION OF DUTIES

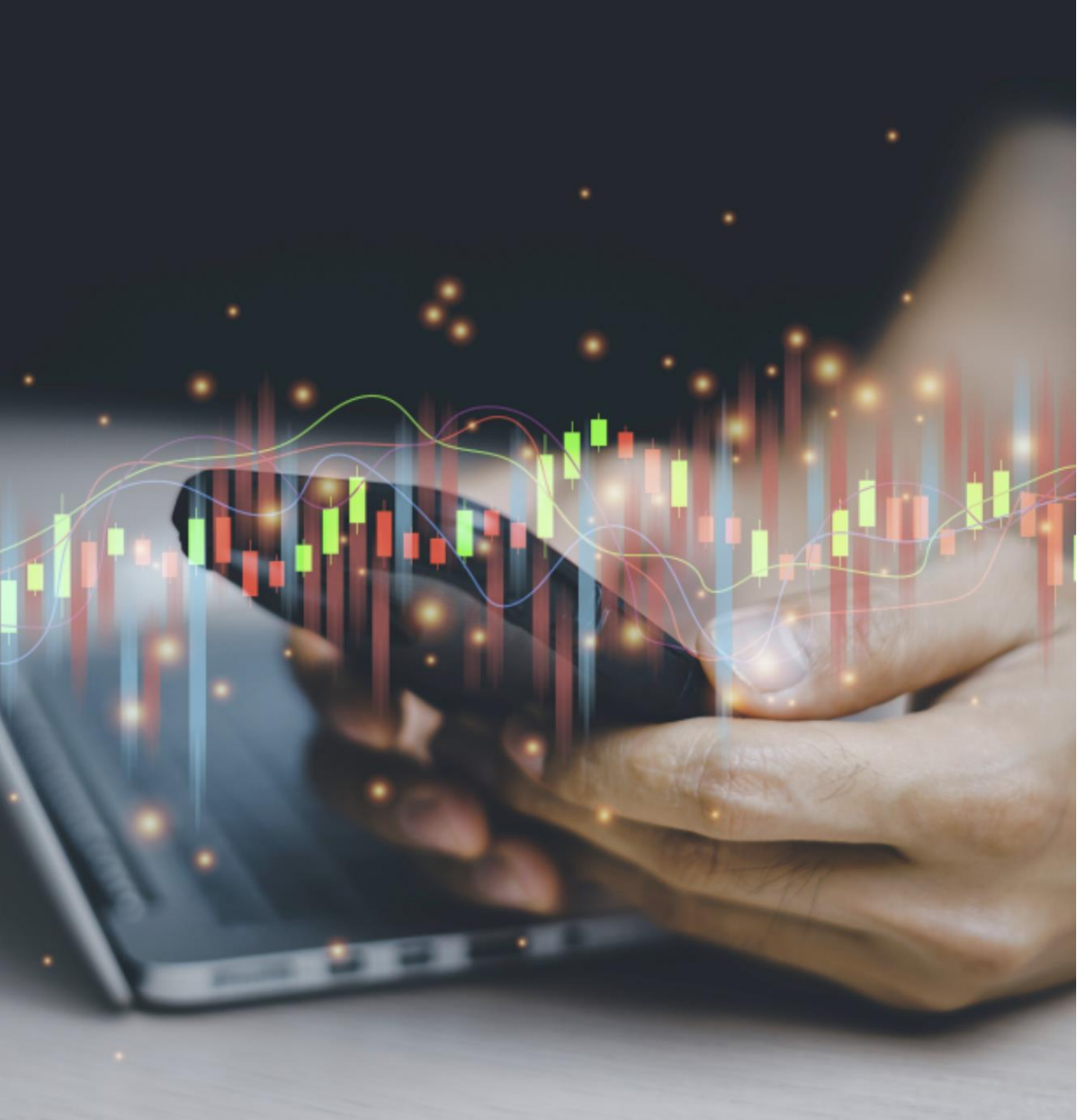
- Separation (also segregation) of duties (SoD) refers to the principle that no user should be given enough privileges to misuse the system on their own:
  - For example, the person authorizing a paycheck should not also be the one who can prepare them
- SoD can be enforced either statically (by defining conflicting roles) or dynamically (by enforcing the control at access time)
- An example of dynamic separation of duty is the two-person rule:
  - The first user to execute a two-person operation can be any authorized user, whereas the second user can be any authorized user different from the first

SEPARATION OF DUTIES



SoD

- SoD may also involve dual operator principles where two or more subjects are needed to modify or approve:
  - Example: Two signatures or cryptographic keys are required for certain actions
- Rotation of duties is also a related principle:
  - Example: Mandatory time off or forced vacations



ENCRYPTION IN ACCESS CONTROL

- Encryption helps protect private information or sensitive data and can enhance the security of communication between client apps and servers
- In essence, when data is encrypted, even if an unauthorized person or entity gains access to it, they will not be able to read it
- Origin authentication uses symmetric and asymmetric encryption keys in a variety of systems, including digital signatures

ENCRYPTION IN ACCESS CONTROL

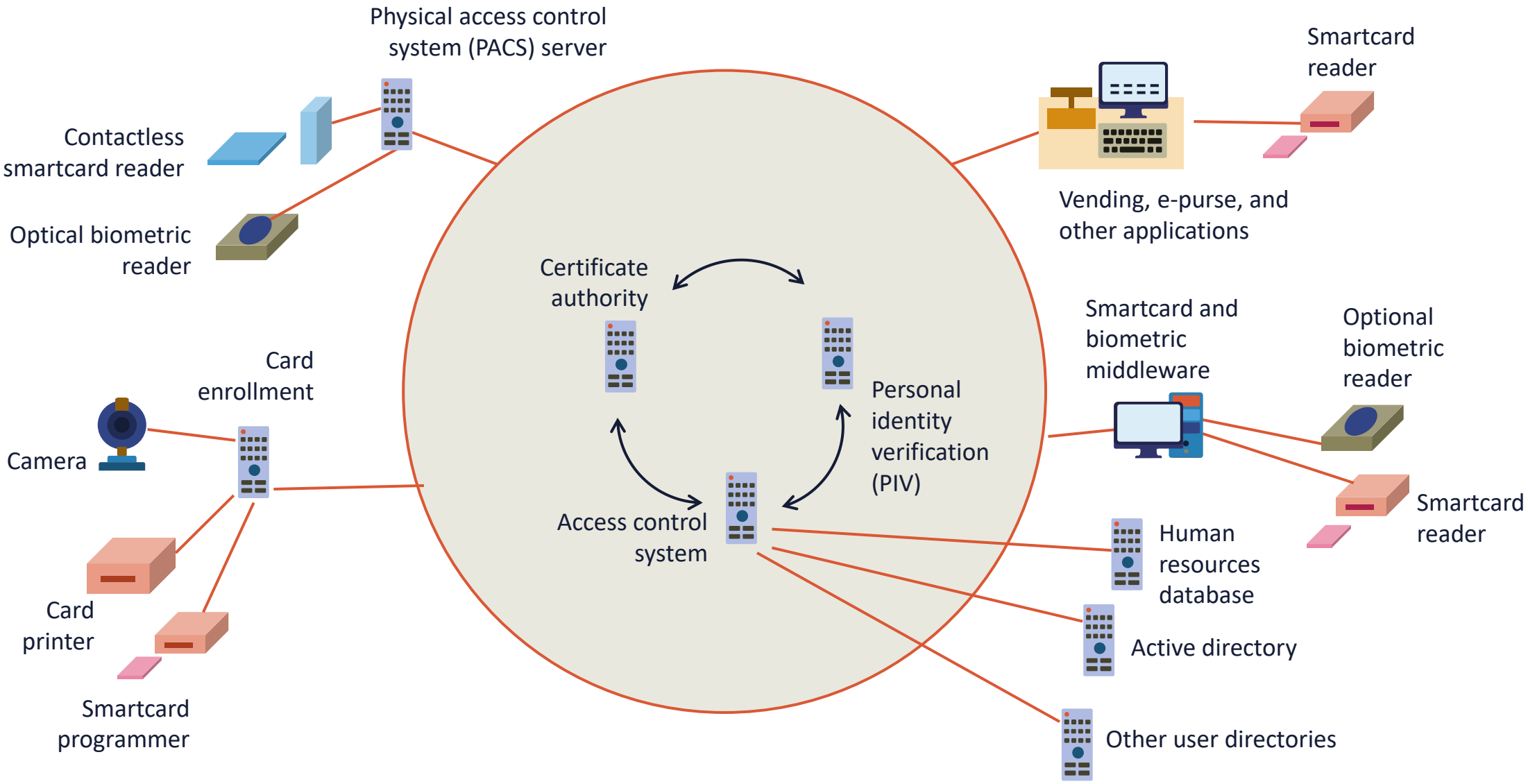
- For example, encryption technologies are involved in shielding private and secret information from unauthorized users, thus safeguarding confidentiality
- This is done by enciphering information in such a way that only authorized users – users with the right key – are able to access the decrypted data

MONITORING AND VISIBILITY OF ACCESS CONTROLS

- Access controls will determine which subjects have read access or visibility into critical access, such as sensitive data
- This visibility is vital for data in transit over remote channels and data stored at third-party locations like code repositories (Git), personal cloud storage, and cloud-based block, object, and file storage systems
- Access control mechanisms also must be completely visible and always monitored

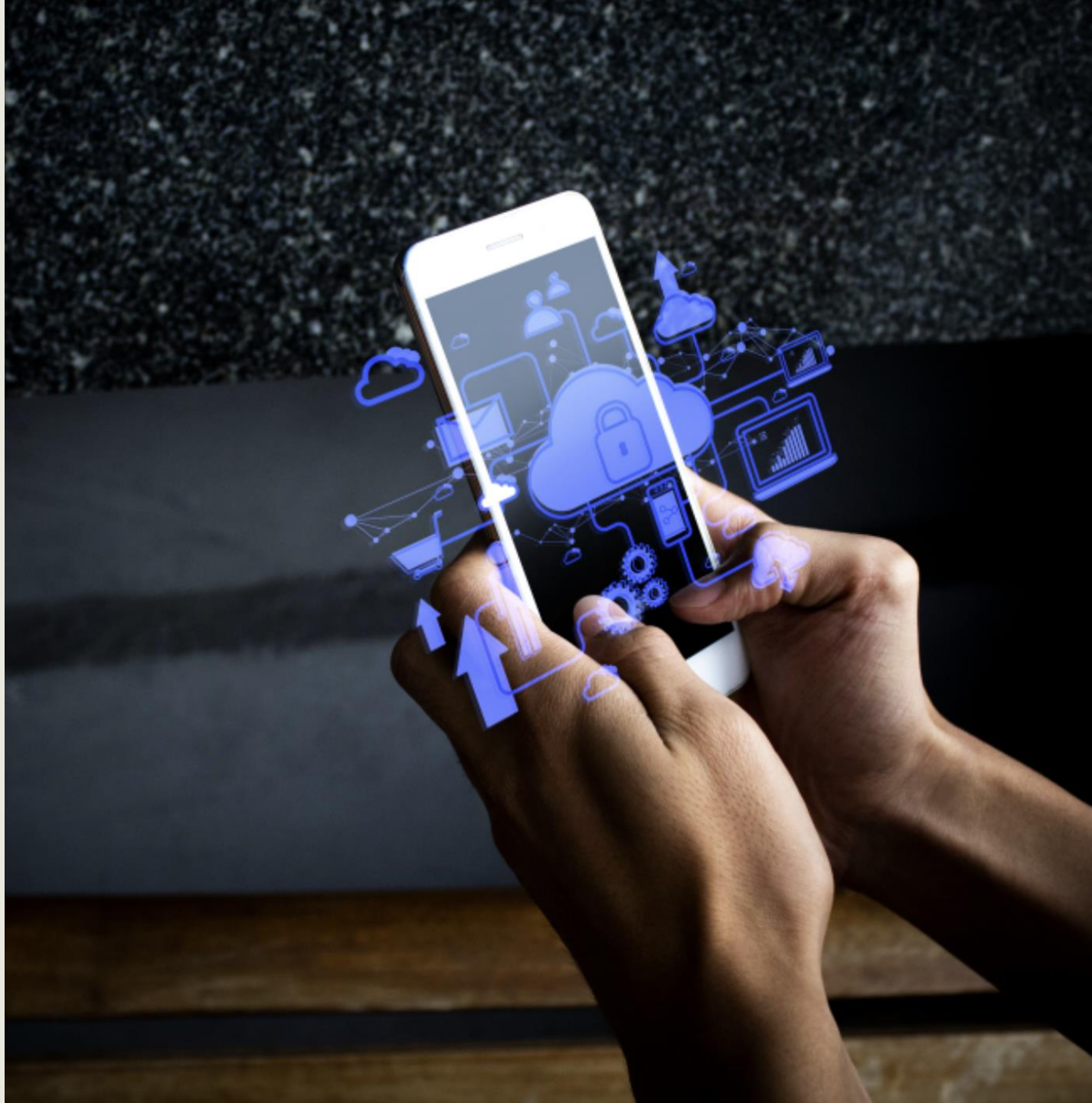
ACCESS CONTROL VISIBILITY

- It is becoming more common to automate monitoring visibility and sending feeds to locations such as security operations centers (SOC) or cloud security information and event management (SIEM)/security orchestration, automation, and response (SOAR) systems like Azure Sentinel
- Audits should be performed regularly to discover gaps or "privilege creep" that can occur with poorly maintained access control and inventory systems
- Other tools that can be used are compliance scanners, PowerShell scanners, vulnerability scanning, and penetration testing


Physical access control system (PACS) server
Contactless smartcard reader
Optical biometric reader
Smartcard reader
Vending, e-purse, and other applications
Certificate authority
Personal identity verification (PIV)
Access control system
Card enrollment
Camera
Card printer
Smartcard programmer
Smartcard and biometric middleware
Optional biometric reader
Smartcard reader
Human resources database
Active directory
Other user directories

DECOMMISSIONING AND OFFBOARDING

- Outgoing employees pose major security risks to organizations
- Security practitioners should make offboarding strategies more resilient
- Without secure off-boarding processes, enterprises expose themselves to a variety of risks, from the harmlessly accidental to the maliciously purposeful
- Risks include data theft, disgruntled leavers, shadow (ghost) IT, unauthorized Software as a Service (SaaS) usage, IT and HR siloed and out-of-sync, access not removed promptly



DECOMMISSIONING AND OFFBOARDING BEST PRACTICES

- Generate solid onboarding policies and processes
- Encourage proactive, interdepartmental collaboration with stakeholders
- Secure corporate assets, devices, and associated credentials
- Make sure there is complete visibility of employees' SaaS, cloud, and third-party access, usage, and permissions
- Monitor for uncommon or even risky behavior of outgoing staff members
- Handle all leavers respectfully and transparently

HARDENING SYSTEMS

- Classic methods of hardening systems involve shutting down TCP and UDP ports and services, including ICMP messages and codes
- Only necessary secured protocols should be used, for example, Secure Shell (SSH) instead of teletype network (telnet)
- Continual patch management initiatives must be implemented for all systems and applications
- Strict least privilege access controls should be used for all administrative users
- Ongoing monitoring and visibility must be instigated

HARDENING SYSTEMS

- Data, systems, and applications can be hardened using symmetric and asymmetric cryptosystems, including data at rest, in transit, and in use
- Endpoints are secured and hardened using trusted platform modules and endpoint detection and response tools:
  - Modern solutions such as Palo Alto Cortex XDR are considered next-generation endpoint detection and host-based intrusion detection and protection

HARDENING TECHNIQUES

Other hardening best practices involve the following tasks:

- Disabling all auto-configure features
- Replacing all default passwords with strong credentials
- Implementing strict password policies or passwordless solutions
- Removing all unnecessary and unauthorized software (personal cloud storage, type 2 hypervisors, exploit kits, etc.)

Architecture and Infrastructure Concepts

Objectives

- Learn about architectural considerations
- Explore cloud computing
- Define Infrastructure as Code, serverless technologies, containers, and microservices
- Examine network infrastructures including centralized vs. decentralized design
- Discover virtualization
- Learn basics of ICS and SCADA
- Define the Internet of Things

RESILIENCE

- Resilience is the ability of a system to continue to:
  - operate under adverse conditions or stress, even if in a degraded or debilitated state
  - maintain essential operational capabilities
  - recover to an effective operational posture in a time frame consistent with mission needs
- Resilience is the ability of a workload to recover from infrastructure or service disruptions
- Administrators should be able to dynamically obtain computing resources to meet demand and mitigate disruptions
  - Disruptions can be misconfigurations or transient network issues

HIGH AVAILABILITY

- Availability is an aspect of resiliency expressed as a percentage of planned and unplanned downtime over an annual period (99.5, 99.9, 99.95, 99.99)
- High availability entails a system, component, or application operating at high capacity, continuously, without intervention, for a defined period of time
- A highly-available infrastructure is designed to deliver quality performance and handle different loads and failures with minimal or zero downtime

HIGH AVAILABILITY

- Reliability is a measure of percentage uptime, considering the downtime due only to faults, whereas Availability is a measure of the percentage uptime, considering the downtime due to faults and other causes such as planned maintenance
  - For two different systems, it is possible for one system to be more reliable but less available than the other

AVAILABILITY VS. DURABILITY

- Availability has historically been achieved through hardware redundancy so that if any component fails, access to data will remain
- Durability, on the other hand, refers to long-term data protection (i.e., the stored data does not suffer from bit rot, degradation, or other corruption)
  - Durability is concerned with data redundancy so that data is never lost or compromised
- **Example**: AWS S3 and Google Cloud are designed for 99.999999999% (11 nines) durability per object and 99.99% availability per year

OTHER ARCHITECTURAL CONSIDERATIONS

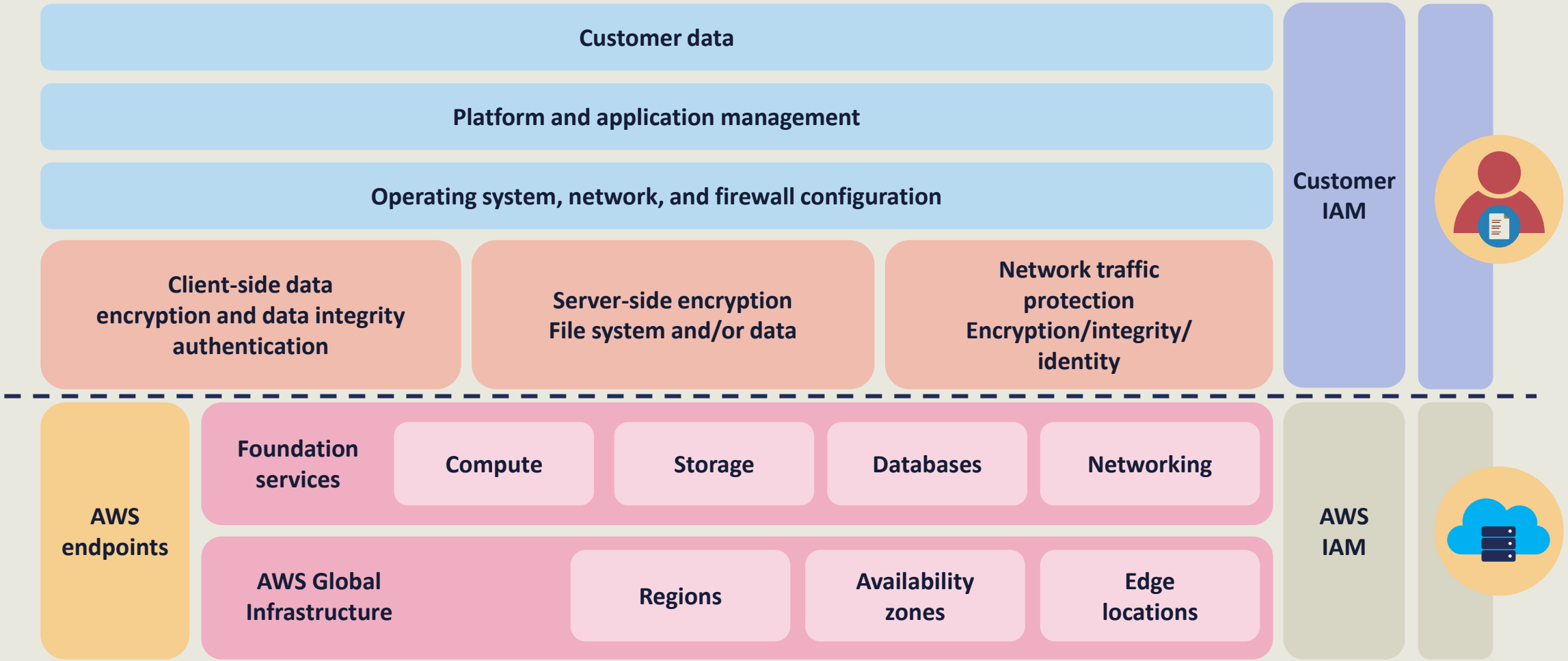
- Cost
- Responsiveness
  - Low latency and performance
- Scalability
  - Scaling out adds physical and virtual instances
  - Scaling up adds compute processor, memory) capacity
- Ease of deployment
  - Infrastructure as Code (IaC)
  - Patching automation
- Risk transference
  - Cloud, insurance, shared disaster sites
- Power

CLOUD COMPUTING: INFRASTRUCTURE AS A SERVICE (IAAS) ACCORDING TO NIST

Infrastructure as a Service is where the "capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer can deploy and run arbitrary software, including operating systems and applications.

The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications, and possibly limited control of select networking components (e.g., host firewalls)."

INFRASTRUCTURE AS A SERVICE AT AWS



CLOUD COMPUTING: PLATFORM AS A SERVICE (PAAS) ACCORDING TO NIST

Platform as a Service is the when the "capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages and tools supported by the provider.

The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly application hosting environment configurations."

PLATFORM AS A SERVICE

Development and software development kit (SDK) platforms for Java, PHP, Python, etc.

Container services for Docker and Kubernetes

Managed and fully managed relational and document databases

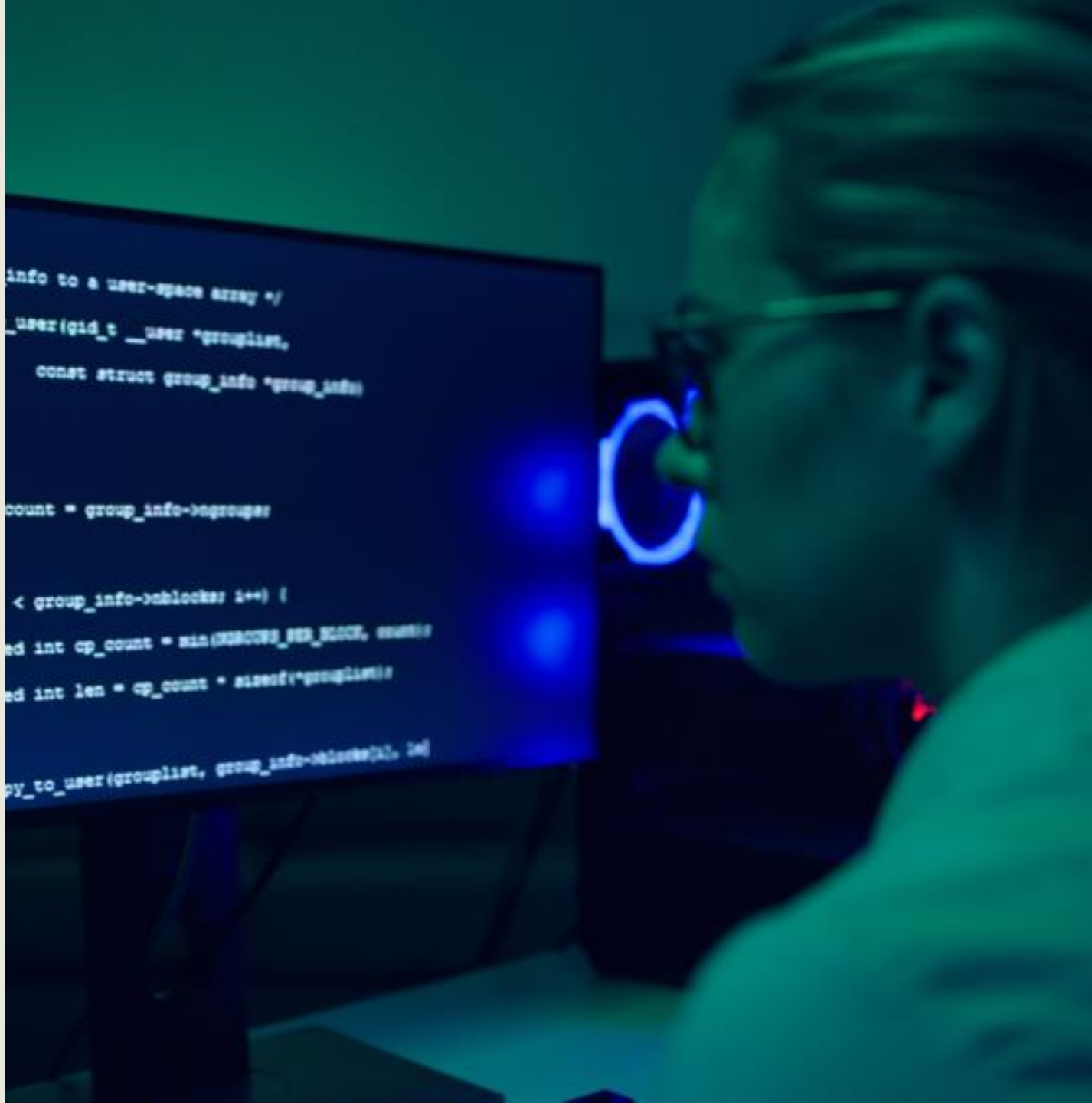
Managed security and threat modeling services

Single sign-on (SSO), machine learning (ML), artificial intelligence (AI), Internet of Things (IoT), blockchain, media services

CLOUD COMPUTING: SOFTWARE AS A SERVICE (SAAS) ACCORDING TO NIST

Software as a Service is when the "capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through a thin client interface such as a web browser.

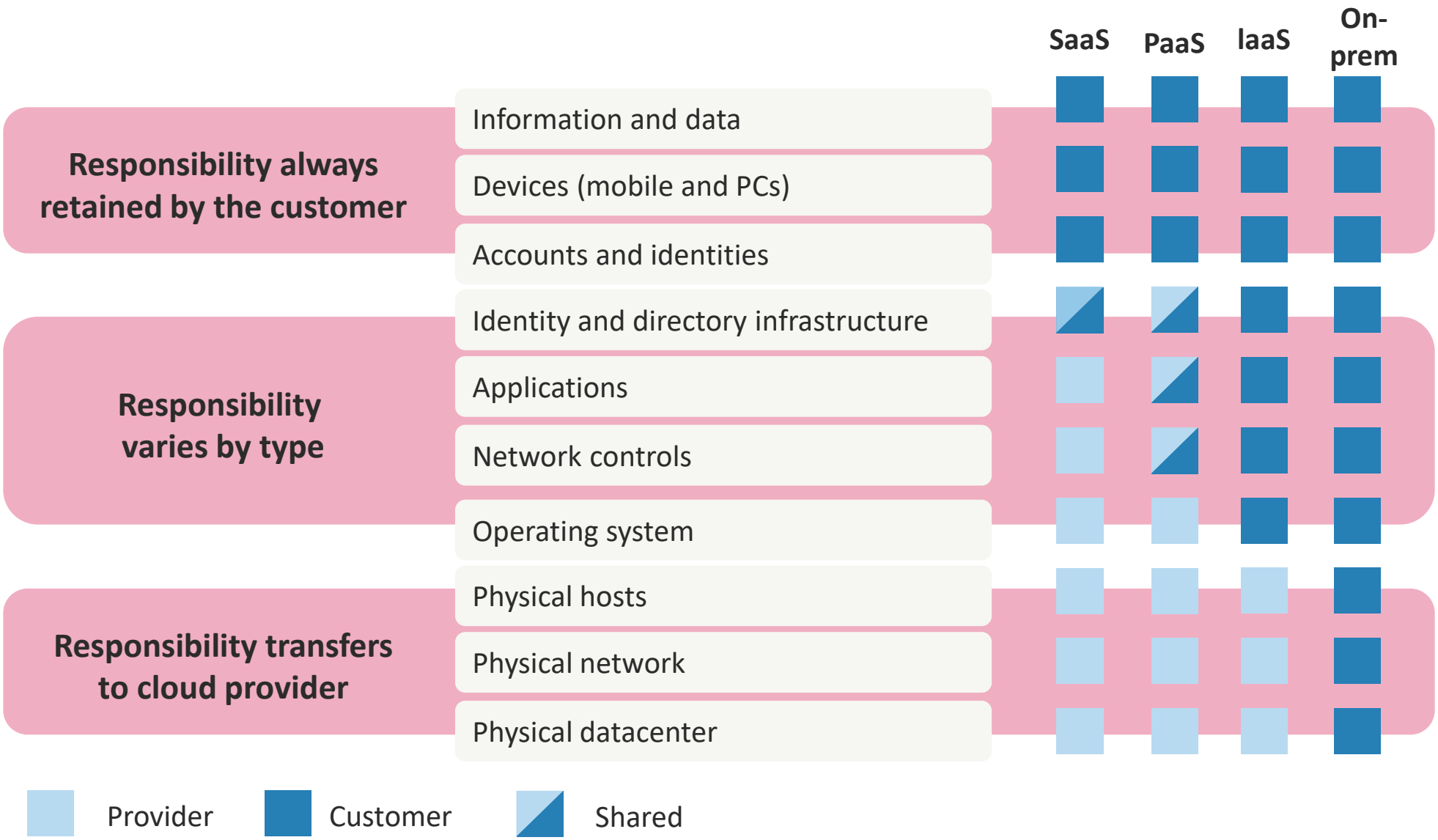
The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings."



SOFTWARE AS A SERVICE OFFERINGS

- Customer relationship management (CRM)
- Enterprise resource management (ERM)
- Human resources and workplace tools
- Finance, sales, and marketing services
- Payroll services
- Email, collaboration, and cloud storage
- Help desk and service desk
- Virtual call center
- Business analytics

AZURE CLOUD RESPONSIBILITY MATRIX



CLOUD DEPLOYMENT MODELS

Public Cloud

The organization runs an initiative (DevOps, DB) entirely at the cloud service provider (CSP) or has public customers for its deployed resources (web, E-commerce)

Private Cloud

A cloud scenario that supports a single organization and its internal customers either in the CSP or on-premises

Community Cloud

A consortium that uses a cloud environment for a particular use case (i.e., gaming community, metaverse, financial, healthcare, etc.)

Hybrid Cloud

A combination of the other three options or an edge computing environment – often bursting up during peak seasons

HYBRID CLOUD CONSIDERATIONS

- Hybrid cloud can also be a method for connecting infrastructure and applications between cloud-based resources and other resources that are not placed in the cloud
- The most common type of hybrid deployment is between the provider's public cloud and a standing on-premises enterprise private cloud
- Can be used to migrate, expand, or grow an organization's infrastructure into a cloud solution while linking internal systems to cloud resources
- Often used by organizations to "burst up" to the cloud during peak demand times or special situations

ON-PREMISES (PRIVATE) CLOUD

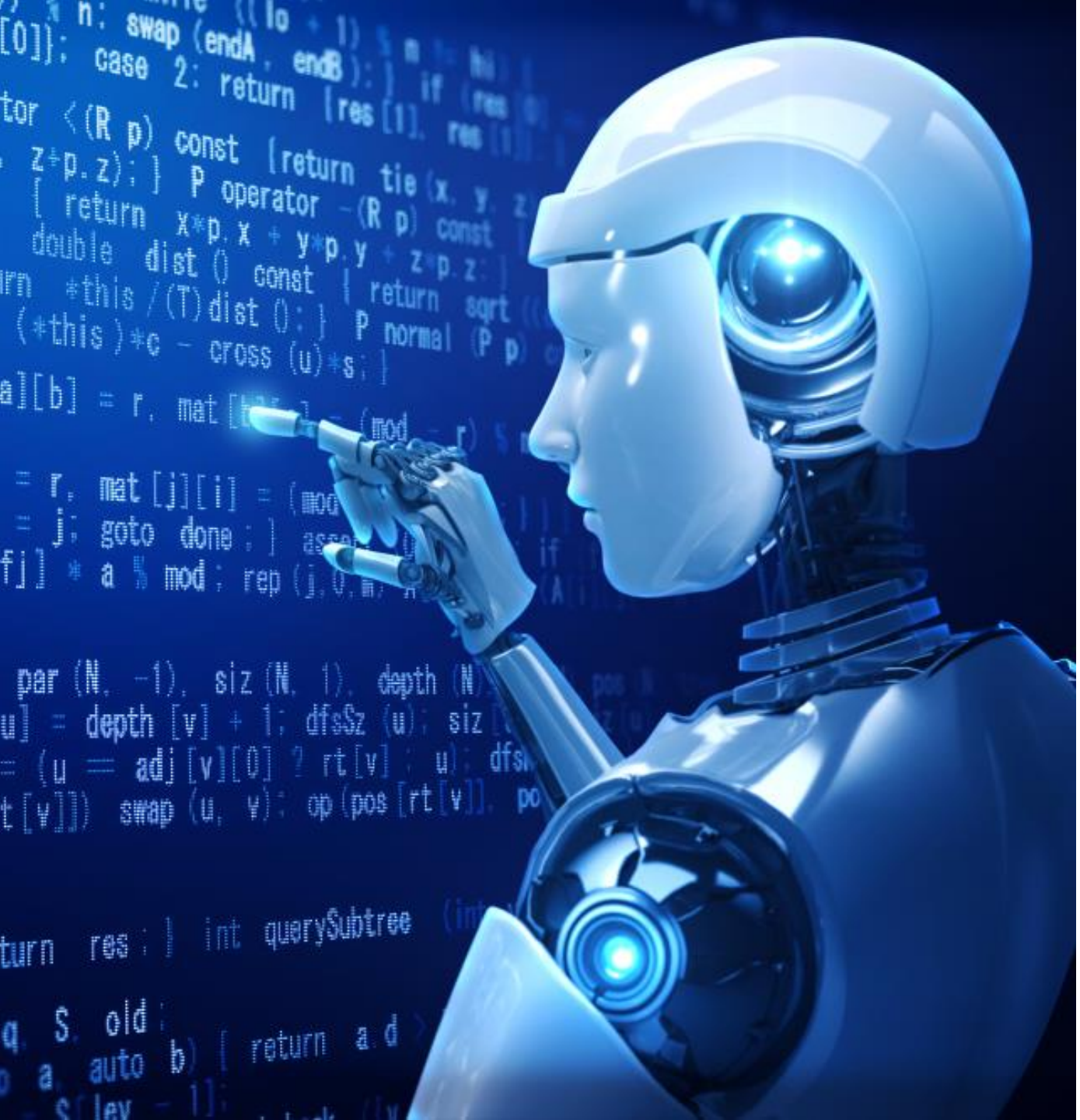
- Involves installing resources on-premises using virtualization and resource management tools, often called private cloud
- On-premises deployment does not provide many of the benefits of cloud computing but is often chosen for its ability to provide dedicated resources
- In most scenarios, this deployment model is the same as legacy IT infrastructure while using application management and virtualization technologies to try and increase resource utilization

THIRD-PARTY CLOUD VENDORS

- **Brokers** – local municipal partners, edge location, reciprocal partners
- **Auditors** – often CSA or SOC certified internal and external auditors
- **MSSP** – managed security service providers such as Fortinet offering cloud-based services (NGFW, NGIPS, EDR, visibility, SIEM+SOAR)
- **CASB** – assisting with SaaS providers for compliance, data loss prevention, and single sign-on
- **Direct Connection** – partners like Direct Connect, Interconnect, and ExpressRoute

INFRASTRUCTURE AS CODE

- IaC is the provisioning and operations of infrastructure using code instead of by manual processes
- Configuration files are created that contain the infrastructure specifications, which makes it easier to edit and distribute configurations
- It also ensures that admins provision the same environment every time
- IaC assists configuration management and helps to avoid undocumented, ad-hoc configuration changes

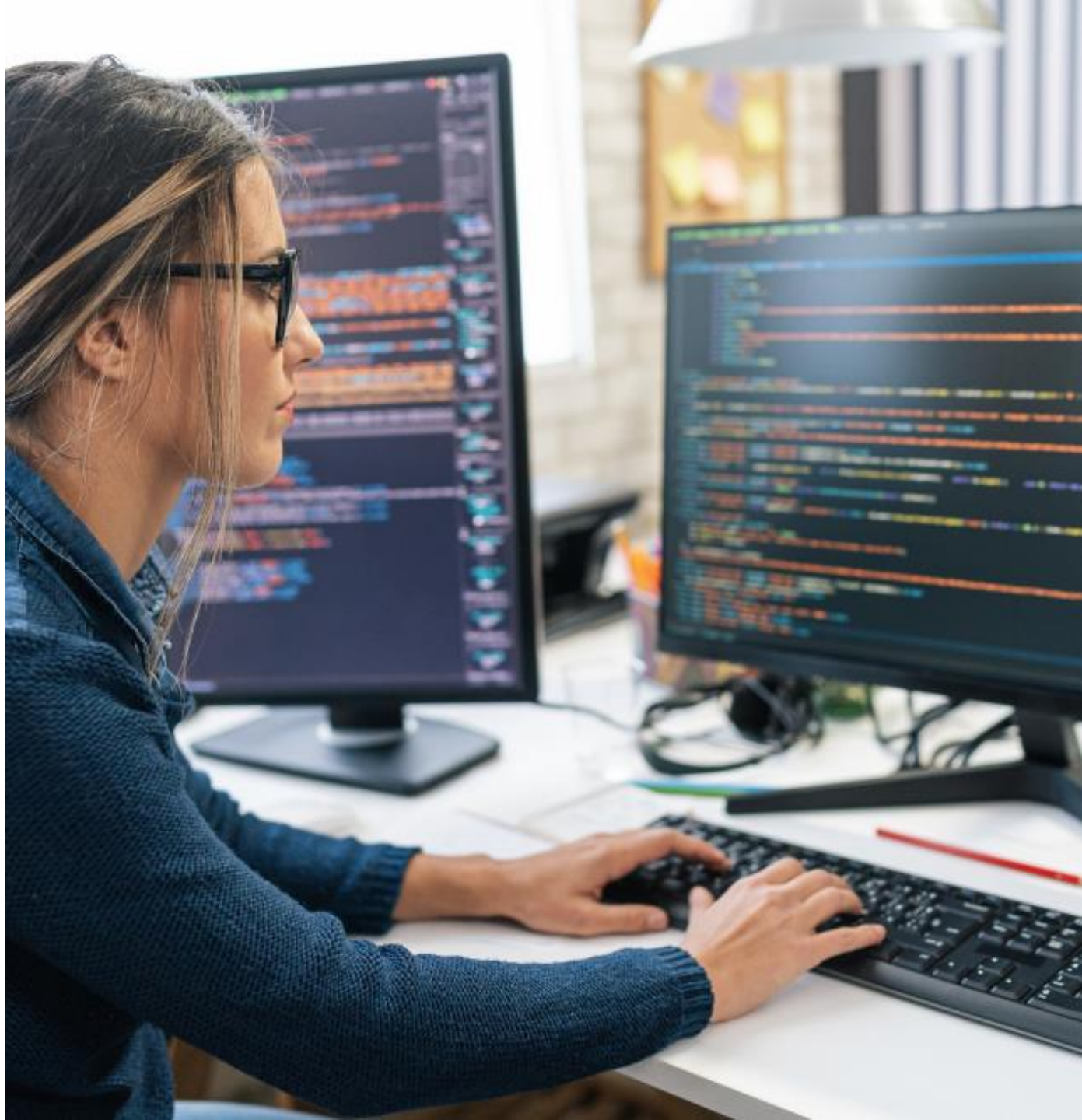


INFRASTRUCTURE AS CODE

- Version control is also an important part of IaC, and all configuration files should be under source control just like any other software source code files
- Deploying with IAC also means that architects can divide their infrastructure into modular components that can then be combined in different ways using automation and orchestration
- This is referred to a generating a "single source of truth" or "terraforming the environment"

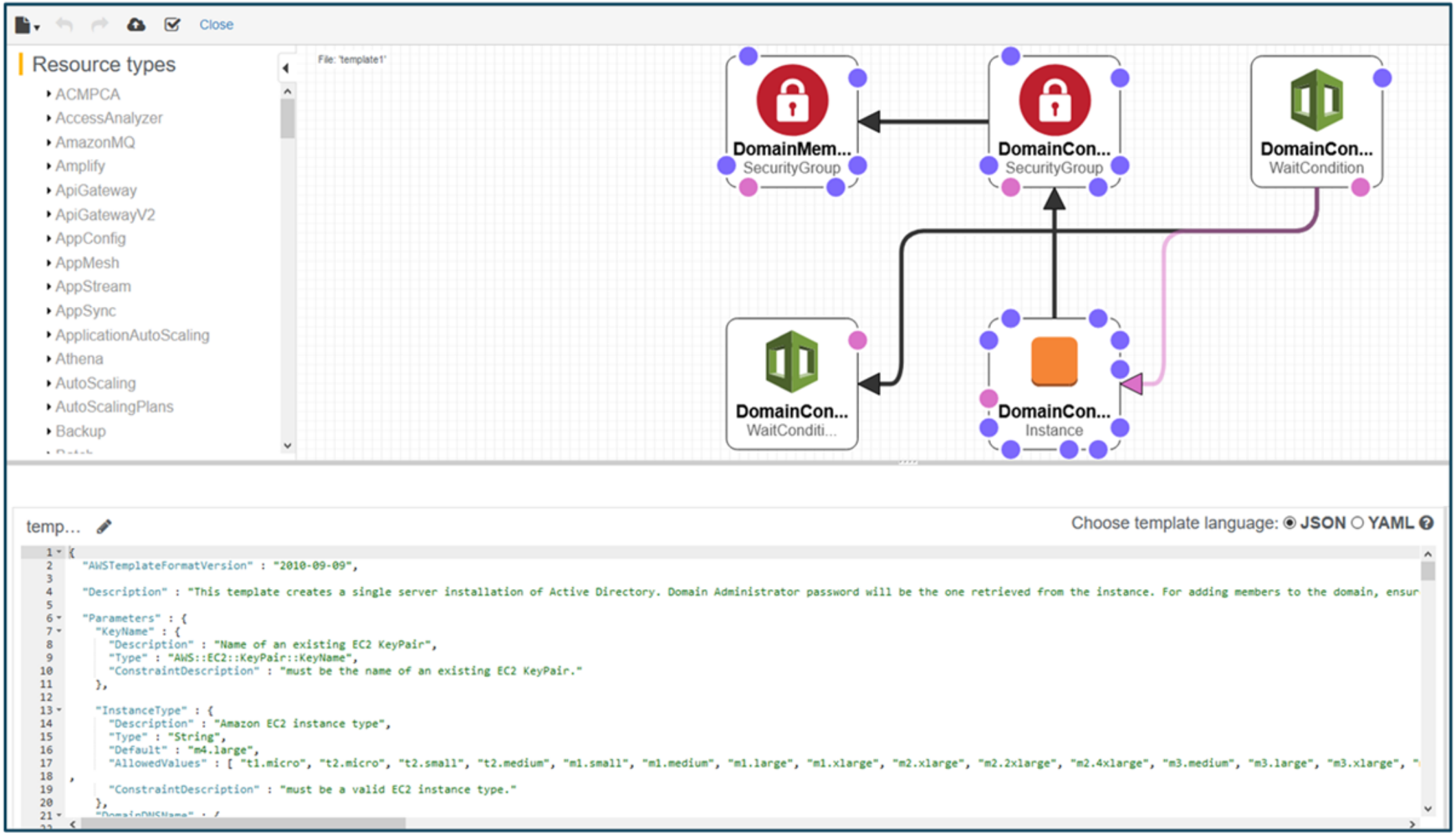
INFRASTRUCTURE AS CODE

- Automating with IaC also means that developers do not need to manually provision and manage servers, operating systems, containers, or microservices each time they develop or deploy an application
- Codifying the infrastructure offers a template to follow for provisioning
- An automation tool, such as Red Hat® Ansible® Automation Platform is a common IaC solution



INFRASTRUCTURE AS CODE

- Cloud services such as AWS CloudFormation empower customers to model, deploy, and manage AWS and third-party resources by handling the Infrastructure as Code
- The cloud template language comes in either JSON or YAML formats
- Customers can automate, test, and deploy infrastructure templates with continuous integration and delivery (CI/CD) automations
- Templates can also be used to set up lab environments for learning the cloud


Close
Resource types
ACMPCA
AccessAnalyzer
AmazonMQ
Amplify
ApiGateway
ApiGatewayV2
AppConfig
AppMesh
AppStream
AppSync
ApplicationAutoScaling
Athena
AutoScaling
AutoScalingPlans
Backup
File: 'template1'
DomainMem...
SecurityGroup
DomainCon...
SecurityGroup
DomainCon...
WaitCondition
DomainCon...
WaitConditi...
DomainCon...
Instance
temp...
Choose template language: JSON YAML
{
"AWSTemplateFormatVersion" : "2010-09-09",
"Description" : "This template creates a single server installation of Active Directory. Domain Administrator password will be the one retrieved from the instance. For adding members to the domain, ensur
"Parameters" : {
"KeyName" : {
"Description" : "Name of an existing EC2 KeyPair",
"Type" : "AWS::EC2::KeyPair::KeyName",
"ConstraintDescription" : "must be the name of an existing EC2 KeyPair."
},
"InstanceType" : {
"Description" : "Amazon EC2 instance type",
"Type" : "String",
"Default" : "m4.large",
"AllowedValues" : ["t1.micro", "t2.micro", "t2.small", "t2.medium", "m1.small", "m1.medium", "m1.large", "m1.xlarge", "m2.xlarge", "m2.2xlarge", "m2.4xlarge", "m3.medium", "m3.large", "m3.xlarge",
,
"ConstraintDescription" : "must be a valid EC2 instance type."
},

SERVERLESS TECHNOLOGIES

- Modern serverless solutions leverage modern cloud infrastructures to emulate the network operating system (NOS) environment without the need for a Windows or Linux-based servers
- These are technologies for running code, managing data, and integrating applications, all without managing servers
- They feature automatic scaling, built-in high availability, and a pay-for-use billing model to increase agility and optimize costs
- Functions, containers, and databases are common serverless solutions

CASE STUDY: AWS FARGATE SERVERLESS CONTAINERS

Without Fargate

Build your container image

Define and deploy the EC2 instances

Provision and manage compute and memory resources

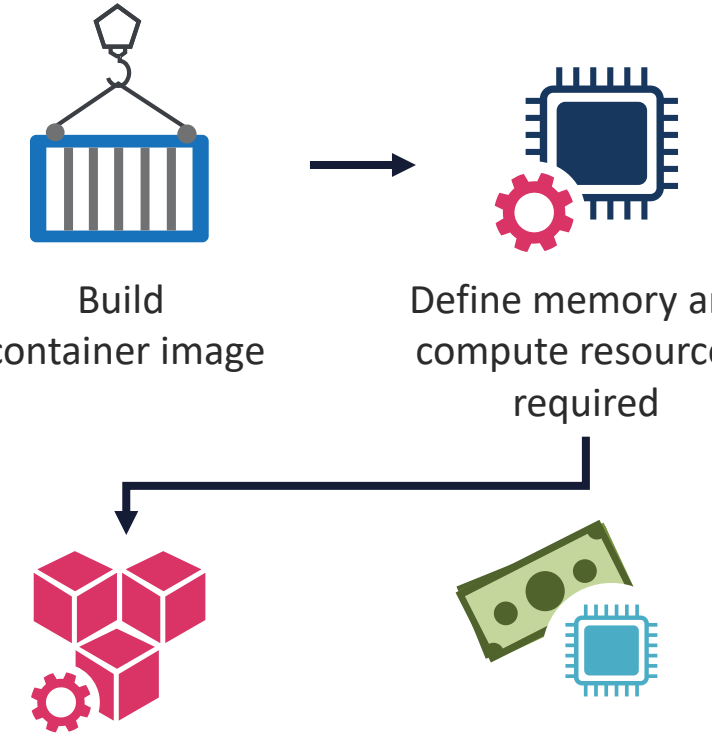
Isolate application in separate virtual machines (VMs)

Define and deploy the EC2 instances

Pay for EC2 instances

AWS Fargate

With Fargate



Build container image

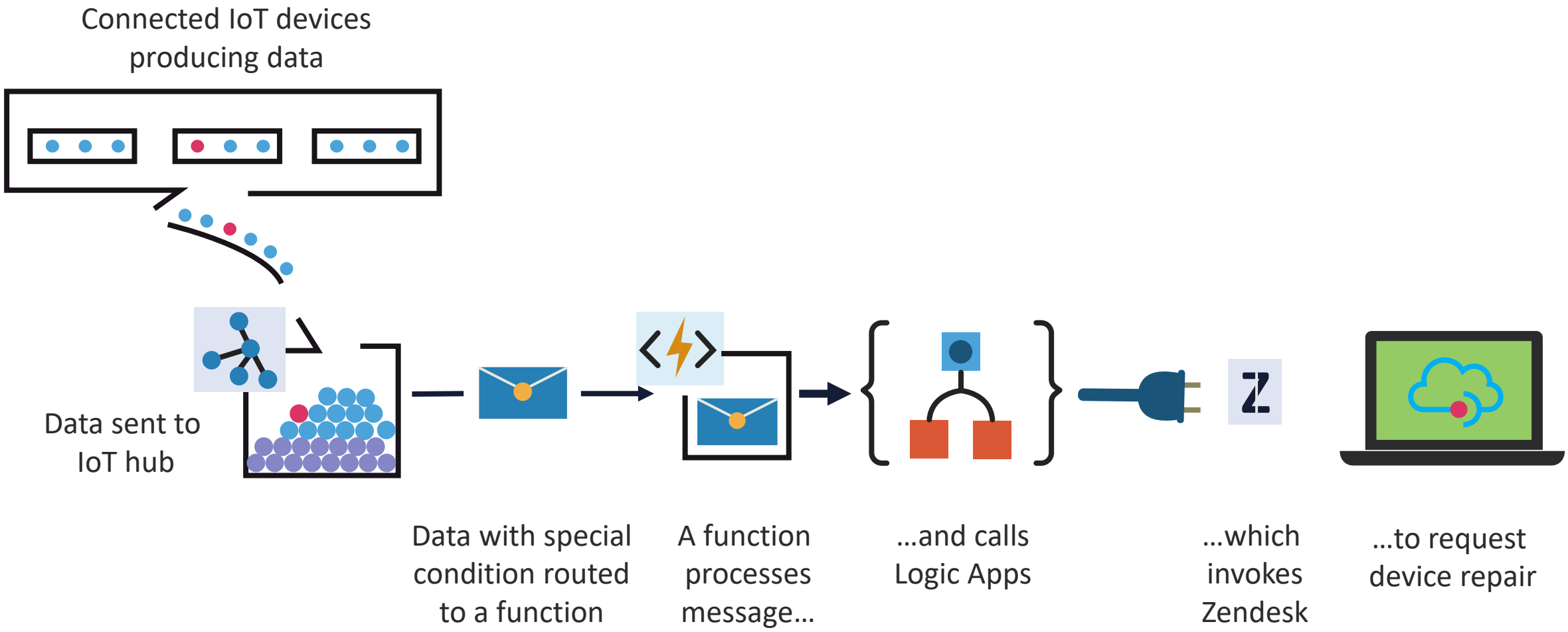
Define memory and compute resources required

Run and mange applications

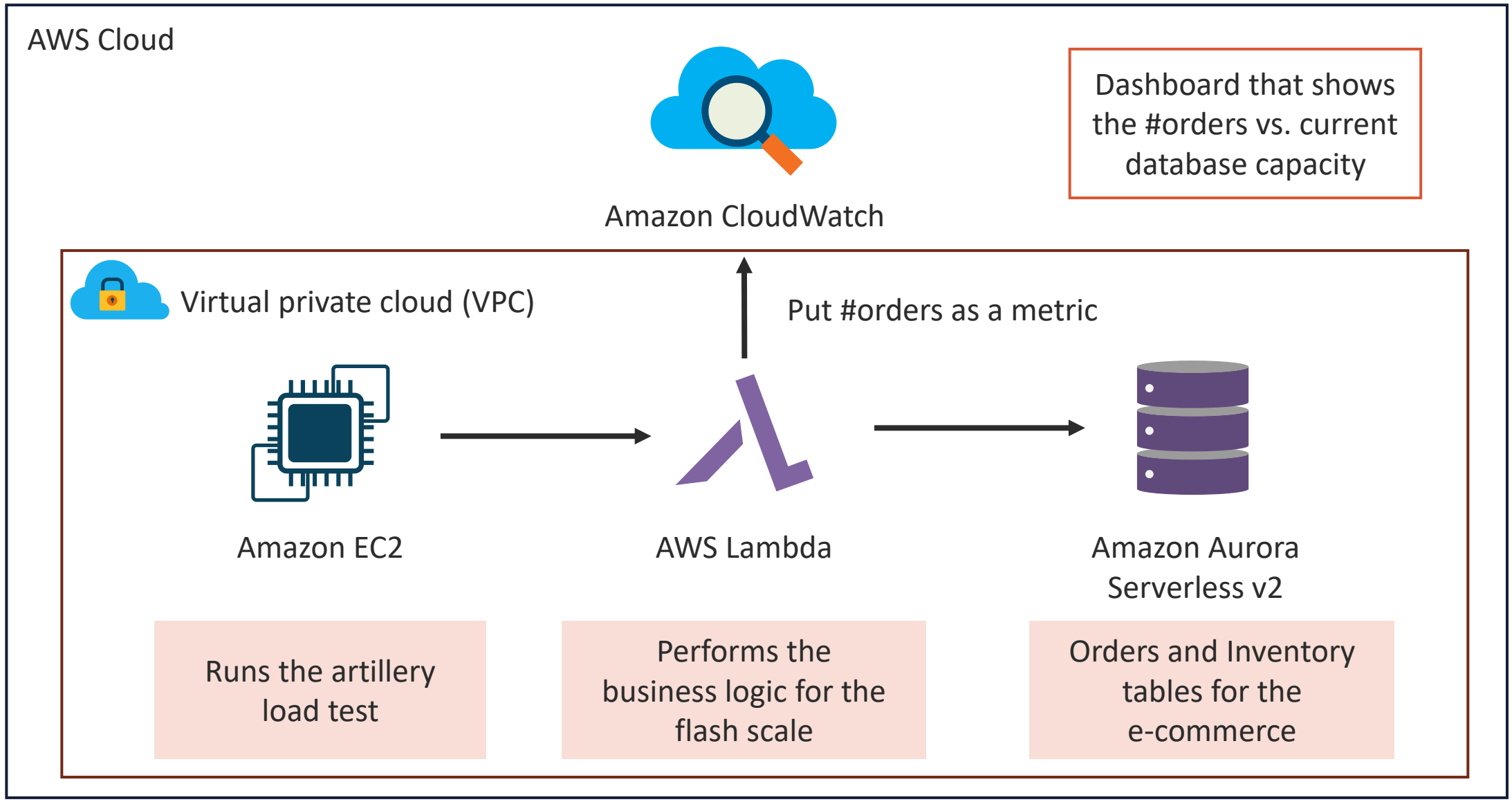
Pay for requested compute resources when used

Applications isolation by design

CASE STUDY: AZURE SERVERLESS FUNCTIONS



CASE STUDY: AWS SERVERLESS DATABASE WITH AURORA



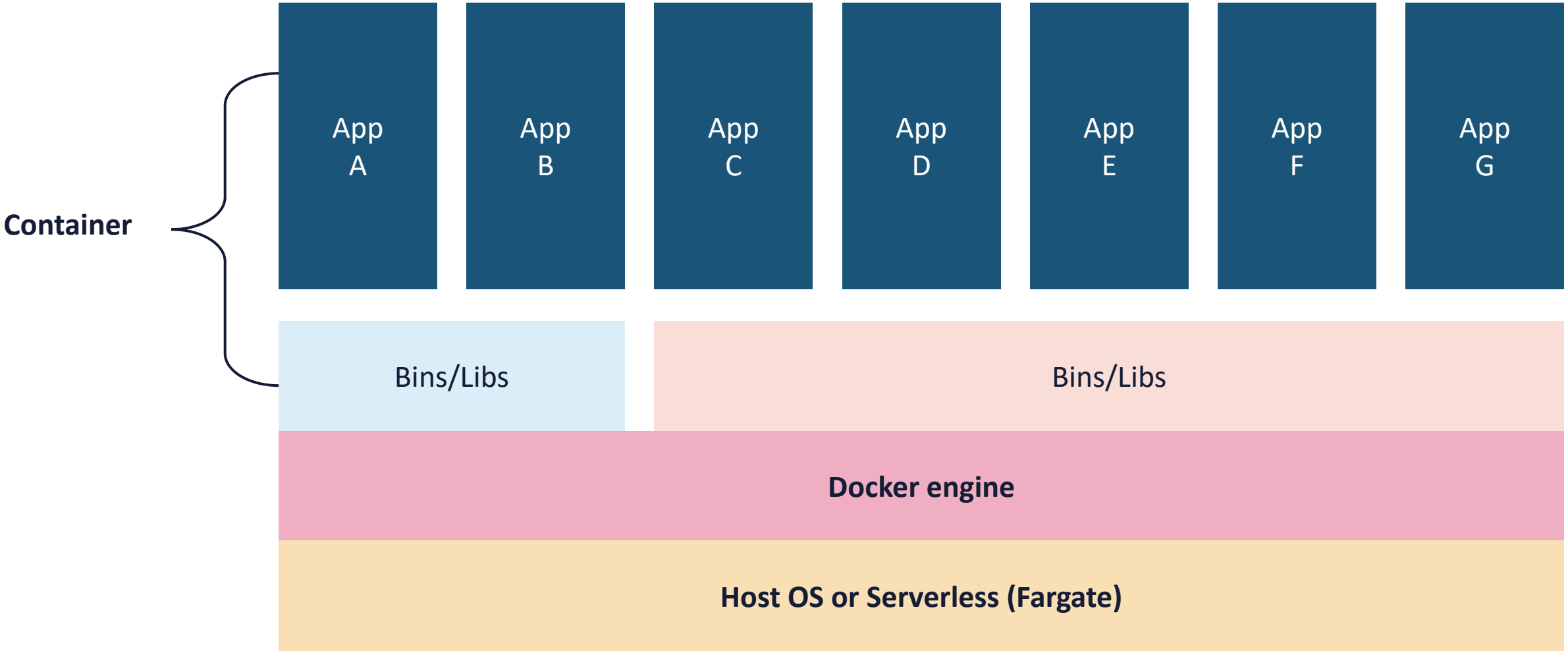
CONTAINERS

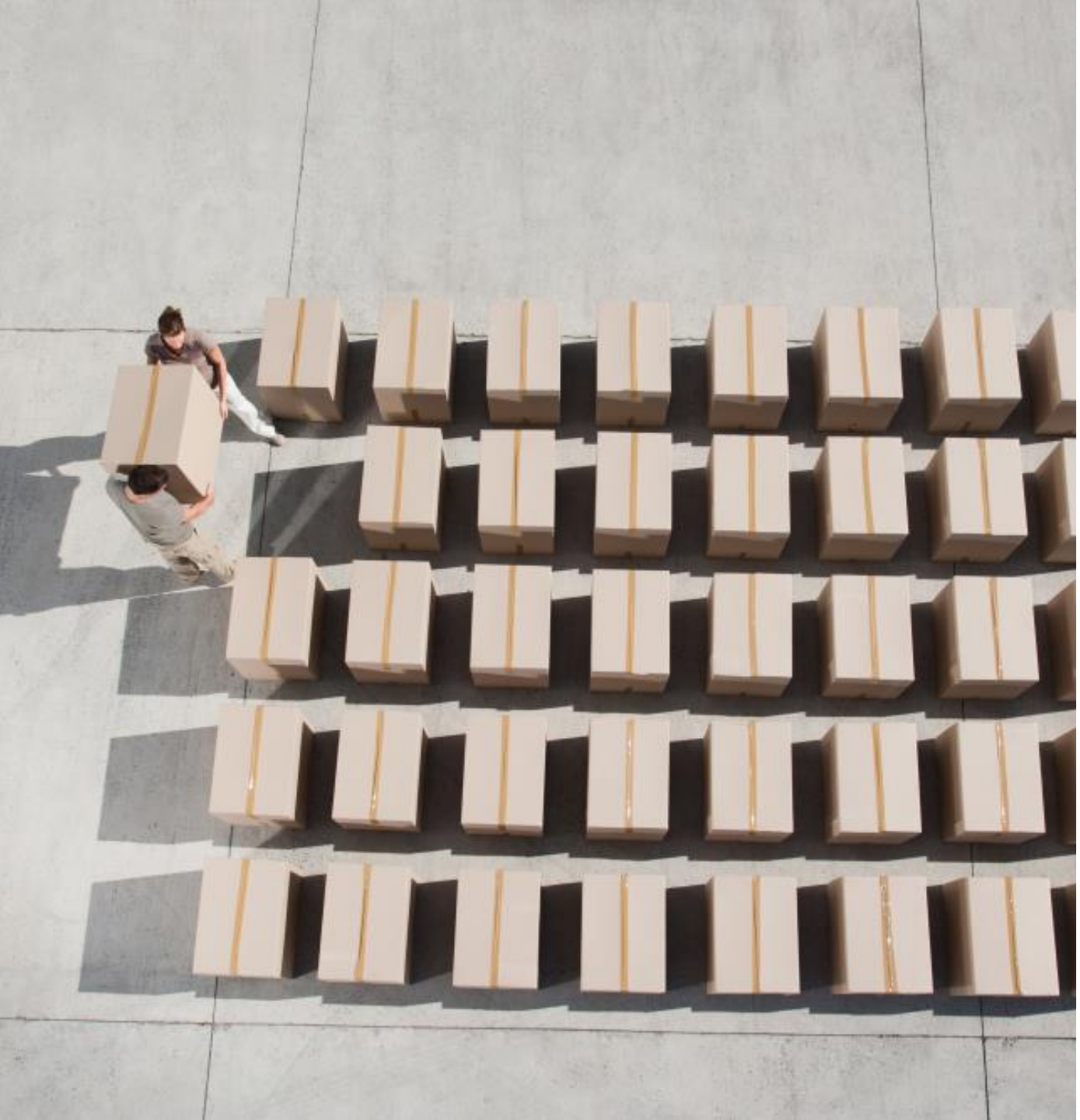
- A container is a discrete environment within an operating system (or a serverless architecture) where one or more applications can run and that is typically assigned all the resources and dependencies needed to function
- It is a modular and portable environment that includes the application binaries, software dependencies, and hardware requirements wrapped up into an independent, self-contained unit

CONTAINERS

- Containers are commonly used for processes and workflows in which there are important requirements for security, reliability, and scalability
- All cloud providers offer managed container development, automation, and orchestration services
- Containers can be server-based or serverless (AWS Fargate)

CONTAINERS

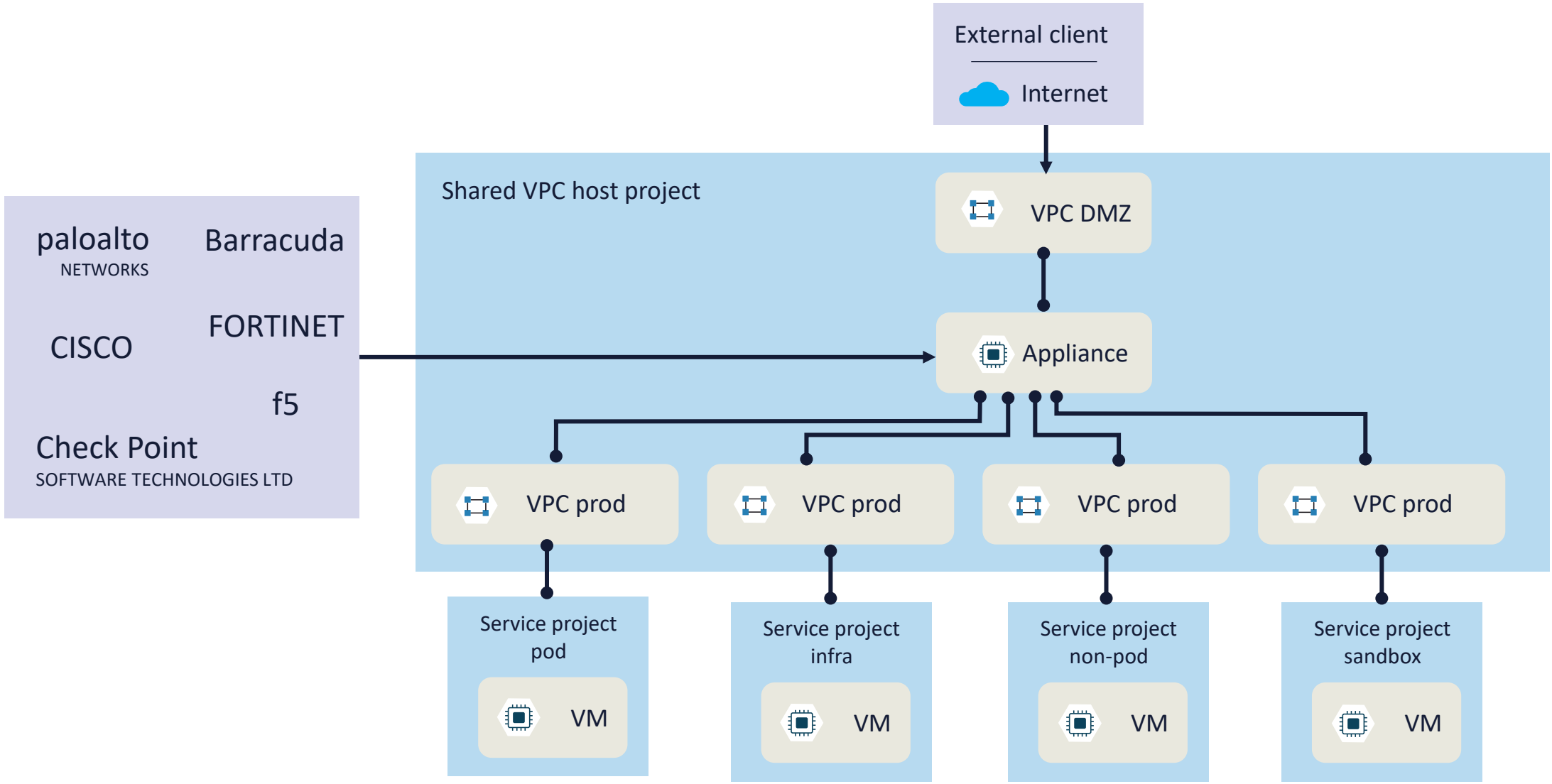

App A
App B
App C
App D
App E
App F
App G
Container
Bins/Libs
Bins/Libs
Docker engine
Host OS or Serverless (Fargate)



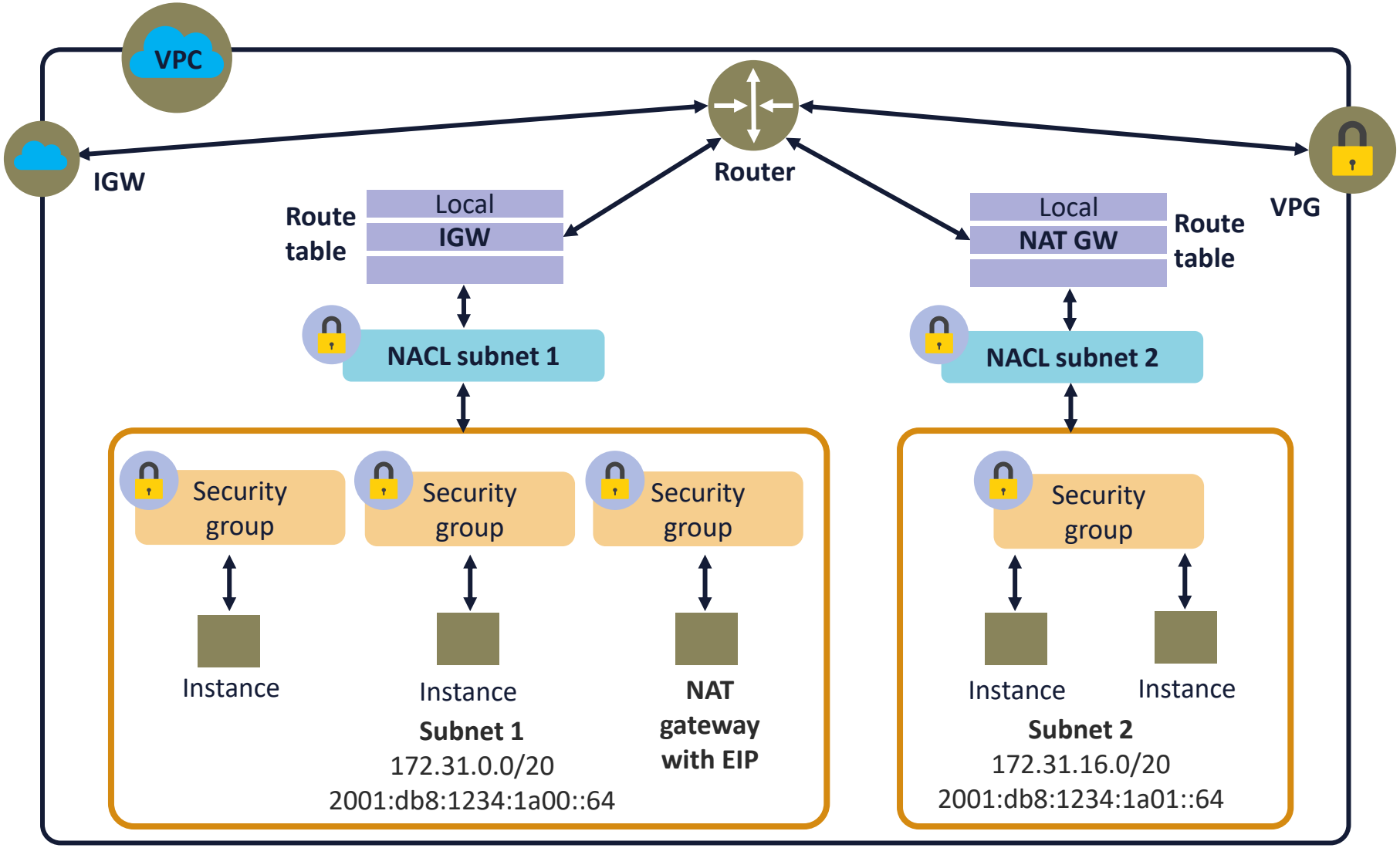
MICROSERVICES

- Microservices are specific service-oriented application components made up of small independent services that communicate over well-defined APIs for notification and process queueing
- They make applications and apps faster to develop and easier to scale by small, self-contained teams of developers
  - Microservices are about the design of software
  - Containers are about packaging software for deployment

CLOUD CUSTOMER NETWORK INFRASTRUCTURE: GOOGLE CLOUD PLATFORM (GCP)

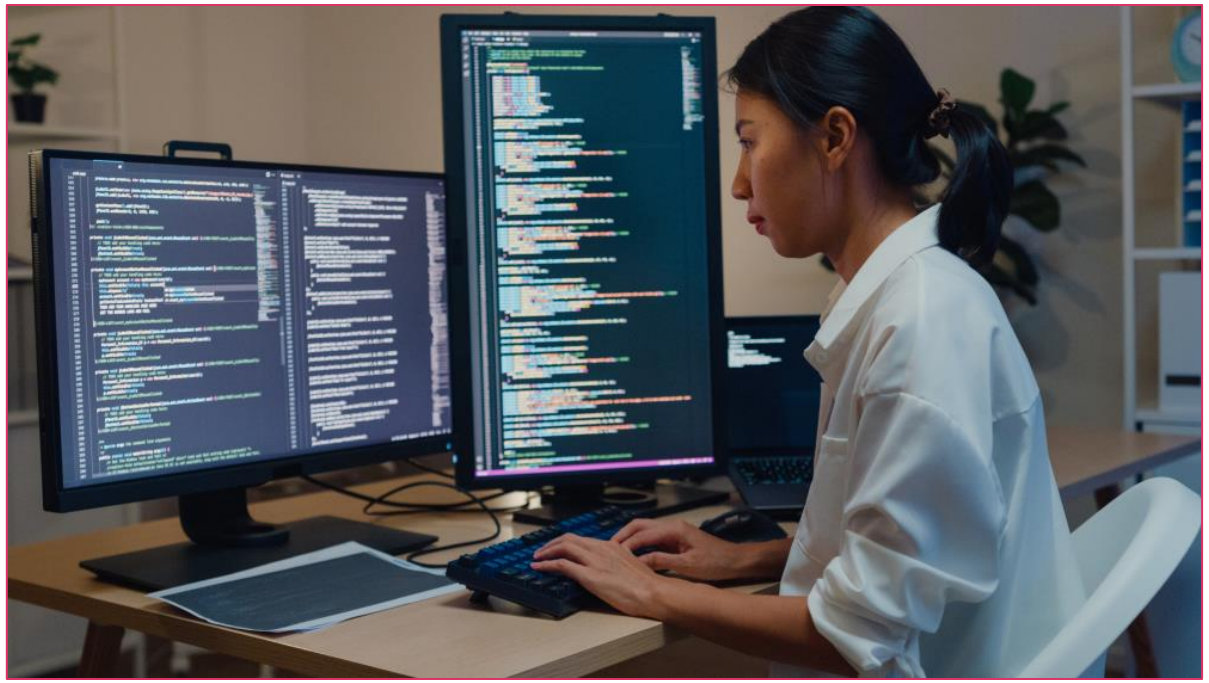


CLOUD CUSTOMER NETWORK INFRASTRUCTURE: AWS



SOFTWARE-DEFINED NETWORKING (SDN)

- Software-defined networking is a framework intended to make a network more flexible and easier to manage, especially with disparate hardware and graphical overlays
- SDN centralizes management by abstracting the control plane from the data forwarding function in the different networking devices

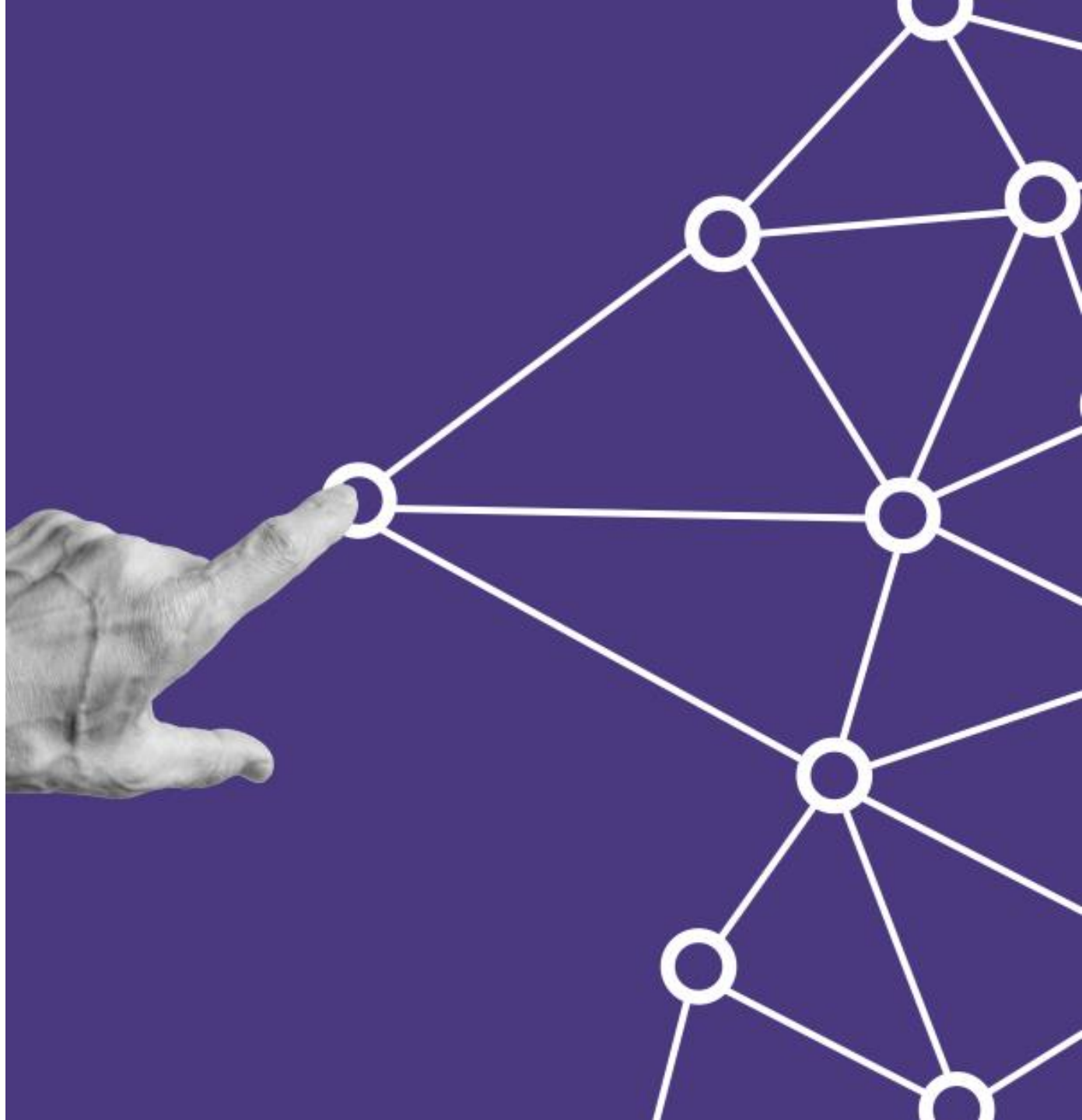


SDN

- An SDN architecture offers a centralized, programmable network consisting of the following:
  - The controller is the essential element of an SDN architecture that assists centralized management and control, automation, and policy enforcement across physical and virtual environments
  - Southbound application programming interfaces (APIs) relay information between the controller and the individual network devices
  - Northbound APIs transmit information between the controller and the applications and policy engines, to which an SDN looks like a single logical network device

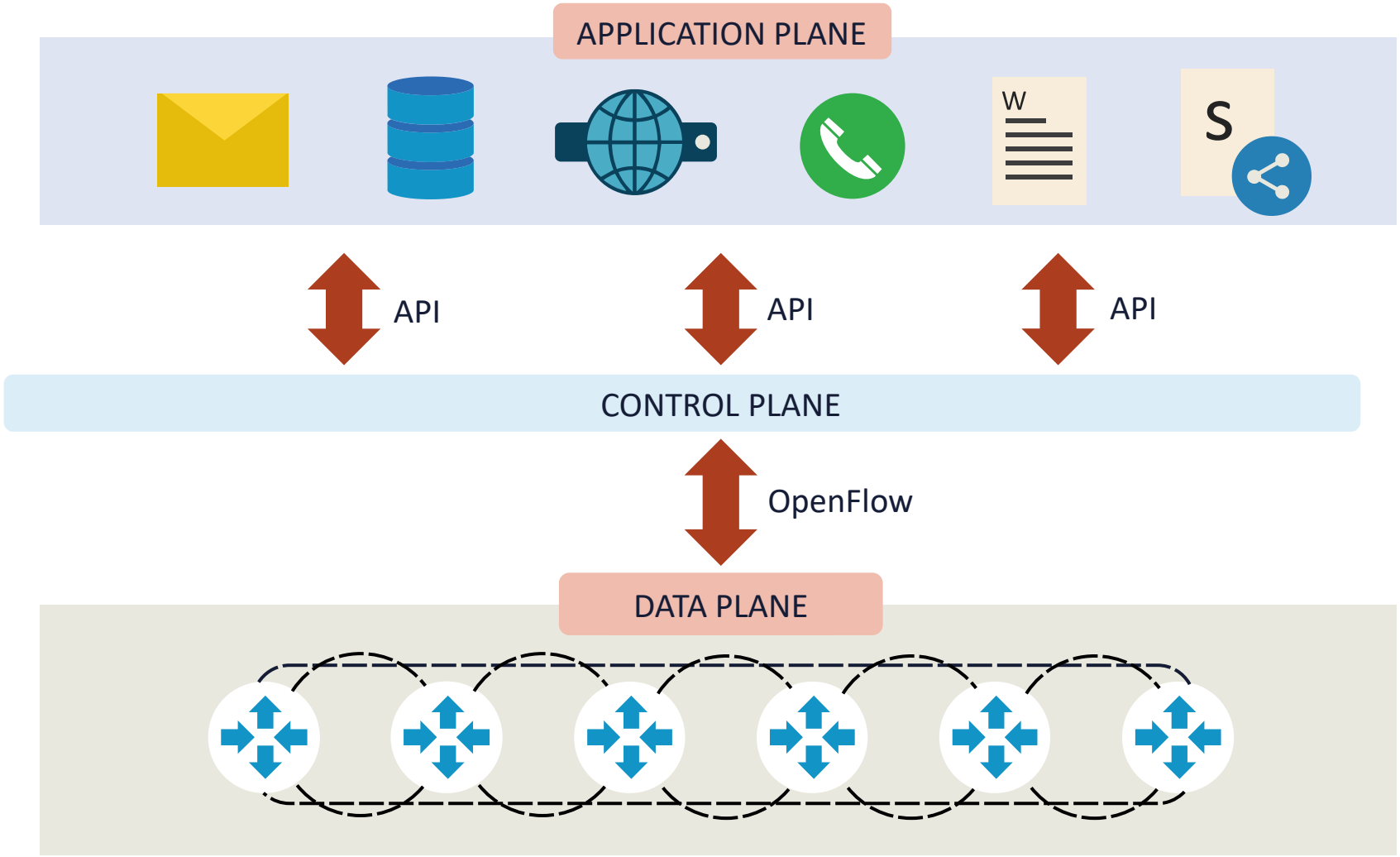
SDN CHARACTERISTICS

- Directly programmable
- Agile
- Centrally managed
- Programmatically configured
- Open standards-based and vendor-neutral



SOFTWARE-DEFINED NETWORKING

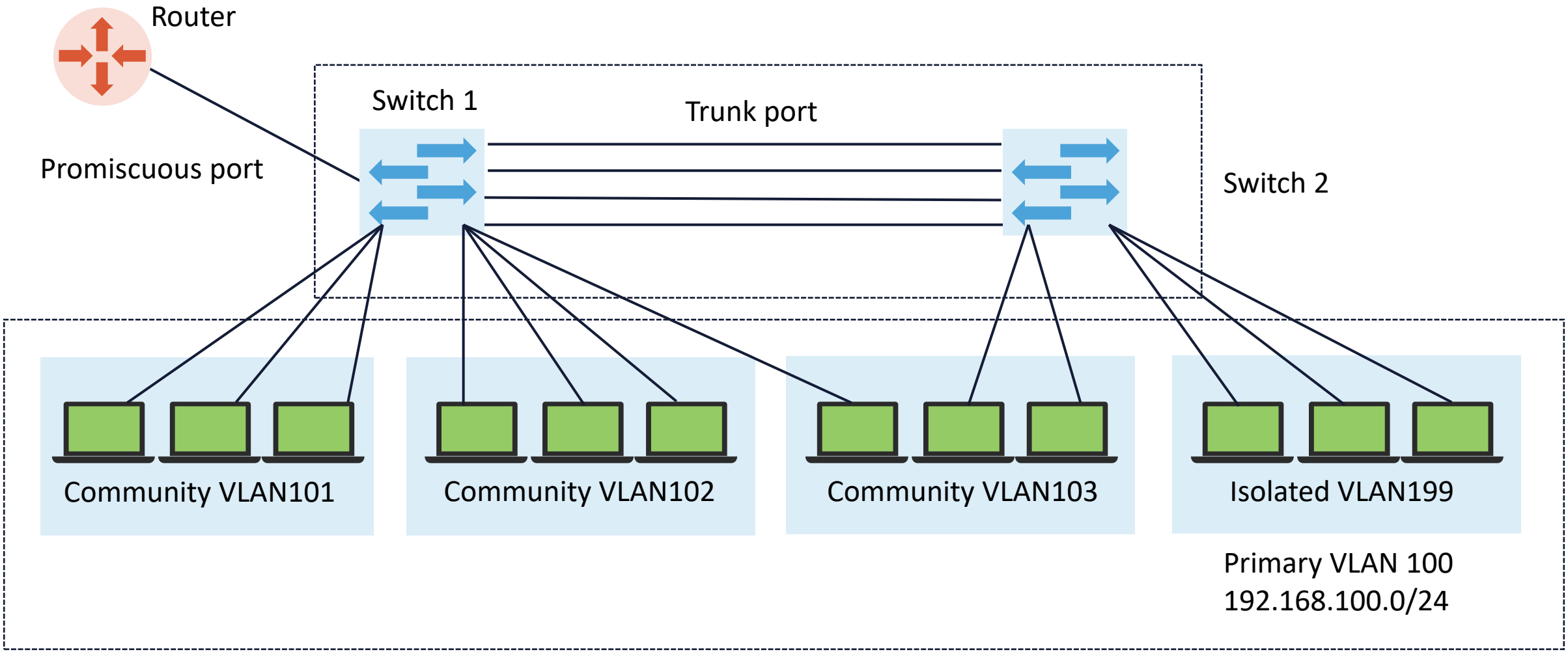
SDN



OTHER NETWORK INFRASTRUCTURE CONCEPTS

- Physical isolation
- Air-gapped
- Logical segmentation

LOGICAL SEGMENTATION



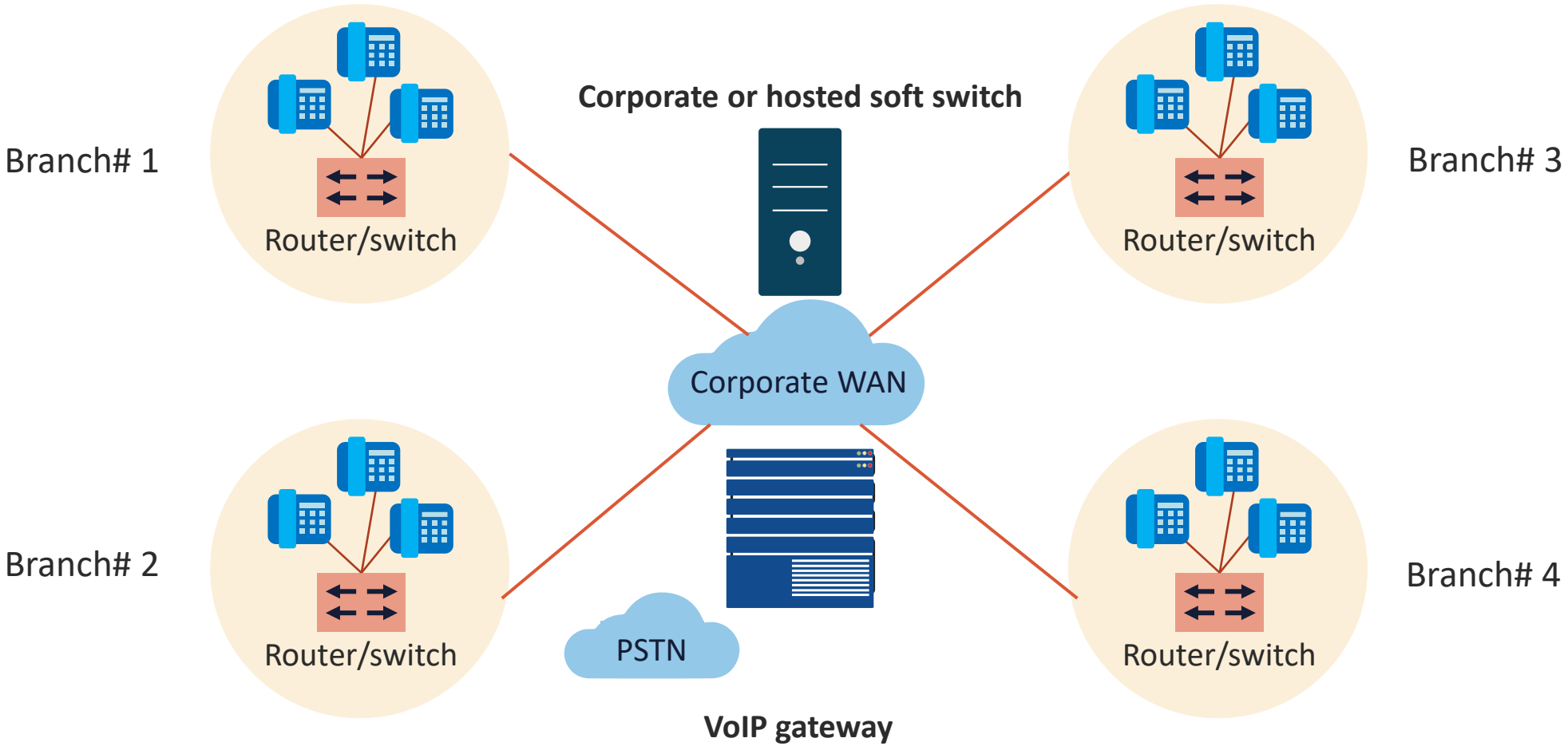
CENTRALIZED DESIGN

- Centralized systems typically deploy a client/server architecture where one or more client node communicates directly (or through a proxy) with a central server either physically or logically
- This is the most common type of system in many organizations where a client sends a request to a corporate intranet and receives a response

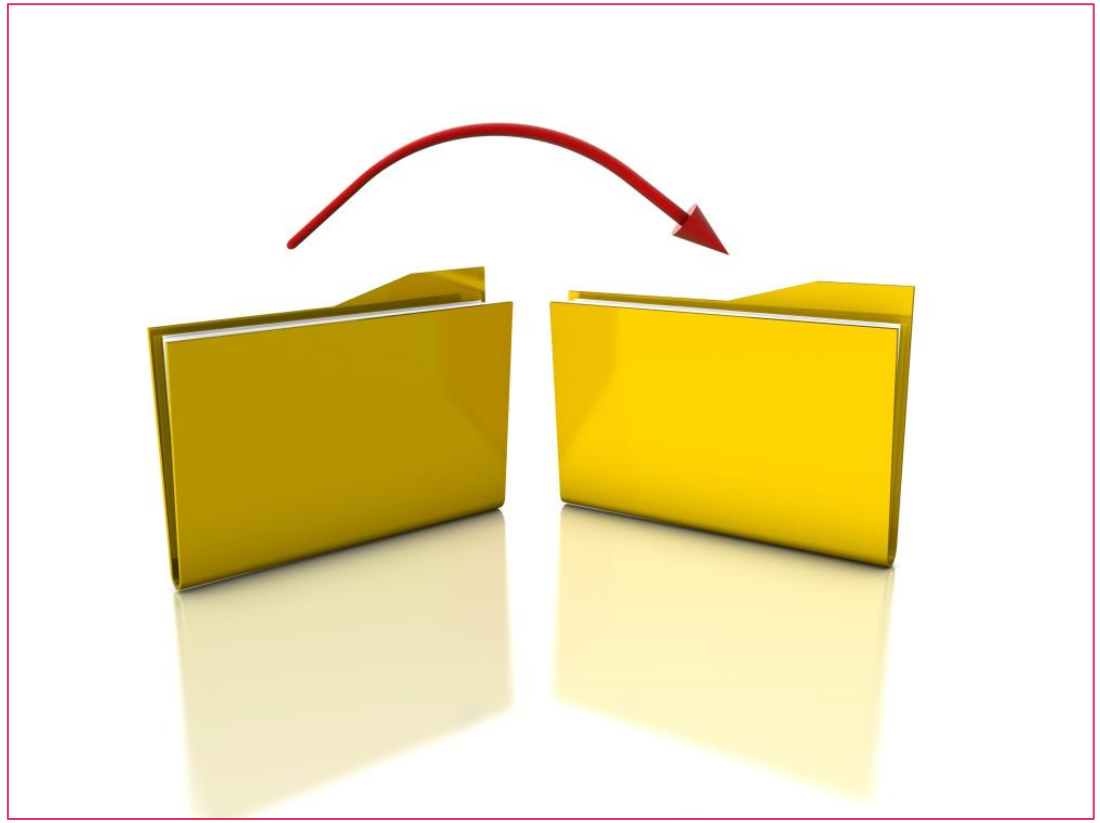
COMMON CENTRALIZED ATTRIBUTES

- Presence of a global clock: all client nodes sync up with the main clock of the central node
- There is one highly available central node that coordinates all the other nodes in the system
- Central node failure causes the entire system to fail because if the server is down, no other entity is there to send/receive responses/requests
- Centralized servers can, in many cases, leverage a hierarchy of intermediate down-level servers

CENTRALIZED WIDE AREA NETWORK (WAN)



DECENTRALIZED DESIGN

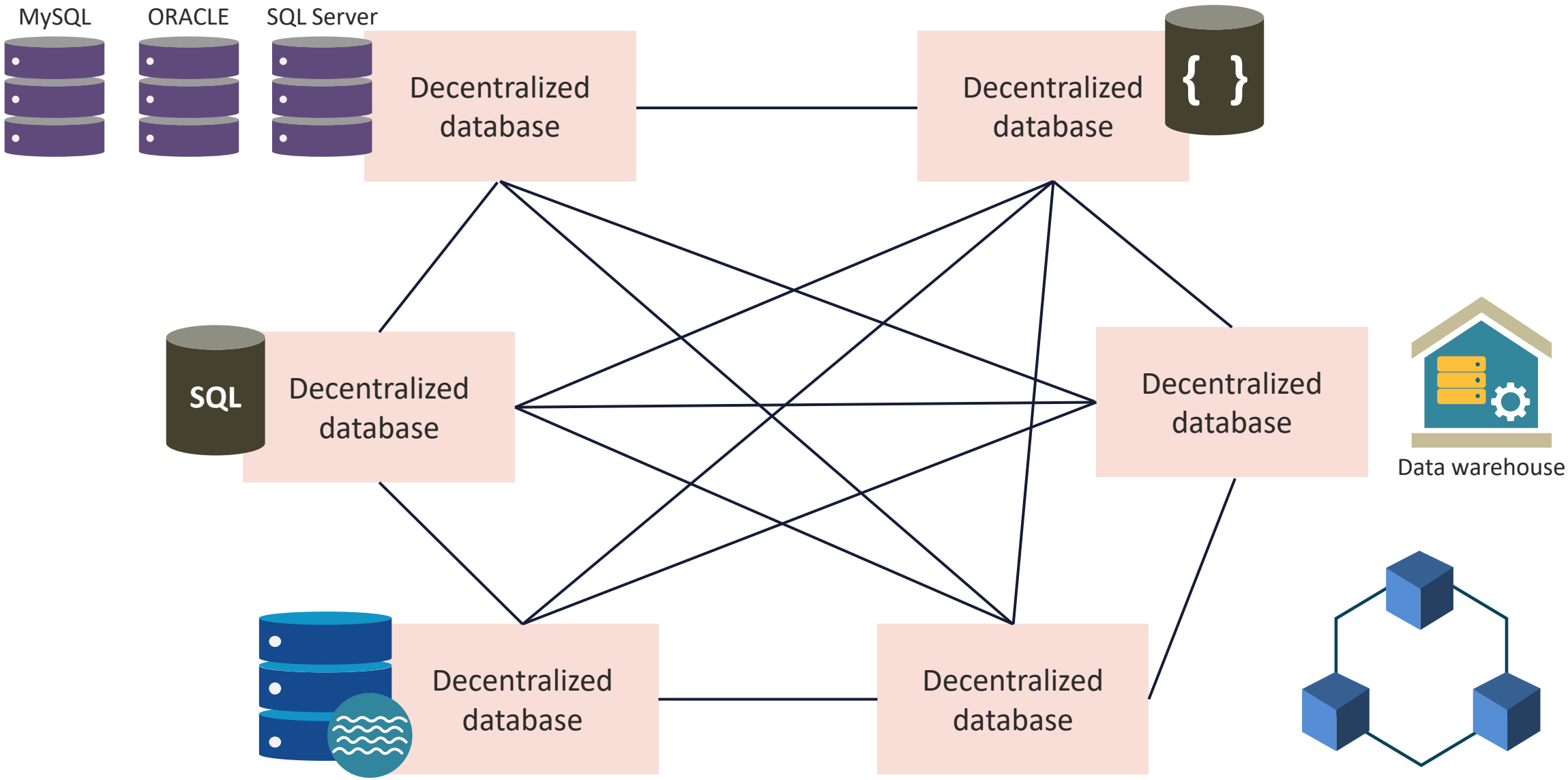


- In a decentralized system, every node makes its own decision
- The final behavior of the system is the aggregate of the decisions of each individual node or host
- There is no single entity that receives and responds to the request
- The requests are broadcasted or multicasted to the decentralized architecture

COMMON DECENTRALIZED ATTRIBUTES

- There is no global clock as each node is independent of each other, and therefore have different clocks that they run and follow
- Decentralized systems have multiple or shifting nodes and more than one unit which can listen for connections from other nodes
- One central node failure causes a part of the system to fail; not the whole system

DECENTRALIZED DATABASE NETWORK



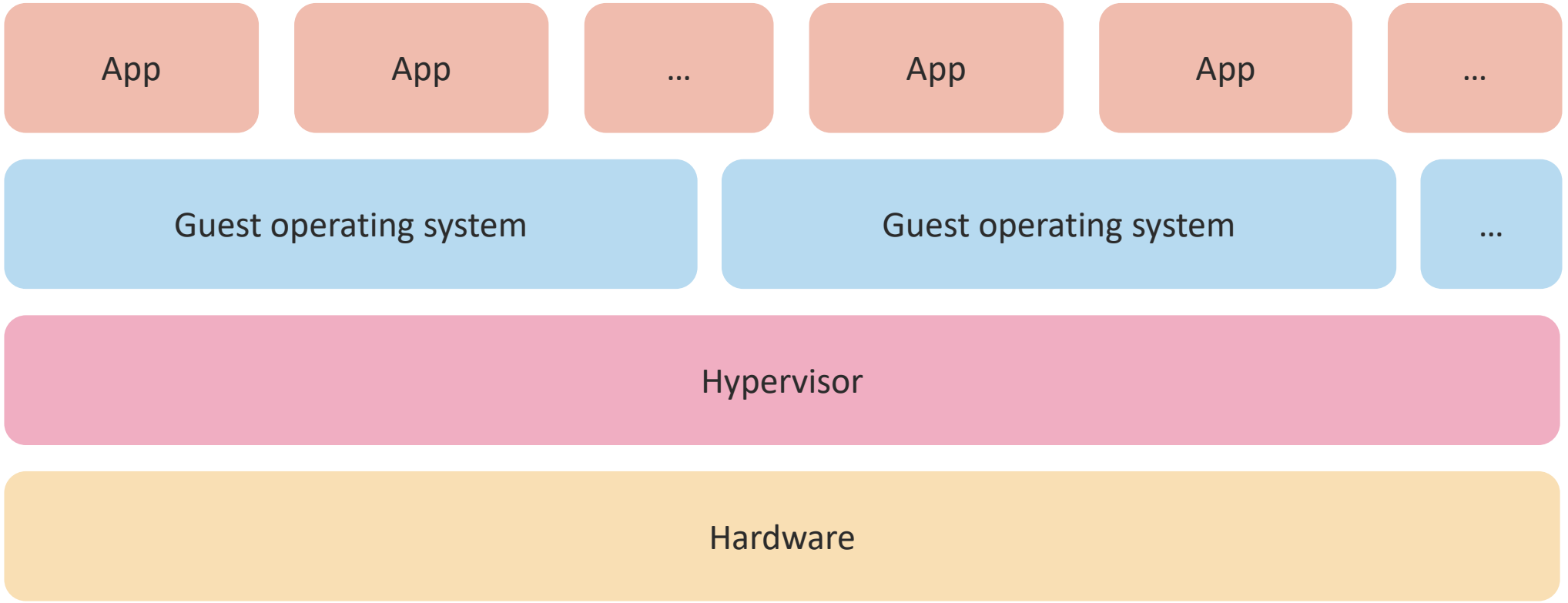
VIRTUALIZATION

- Virtualization is the process of running a virtual instance of a computer system in a layer abstracted from the underlying hardware server
- It most often refers to running multiple operating systems on a computer system simultaneously
- To the applications running on top of the virtualized machine, it can seem as if they are on their own dedicated operating system with libraries, dynamic link libraries (DLLs), and associated programs

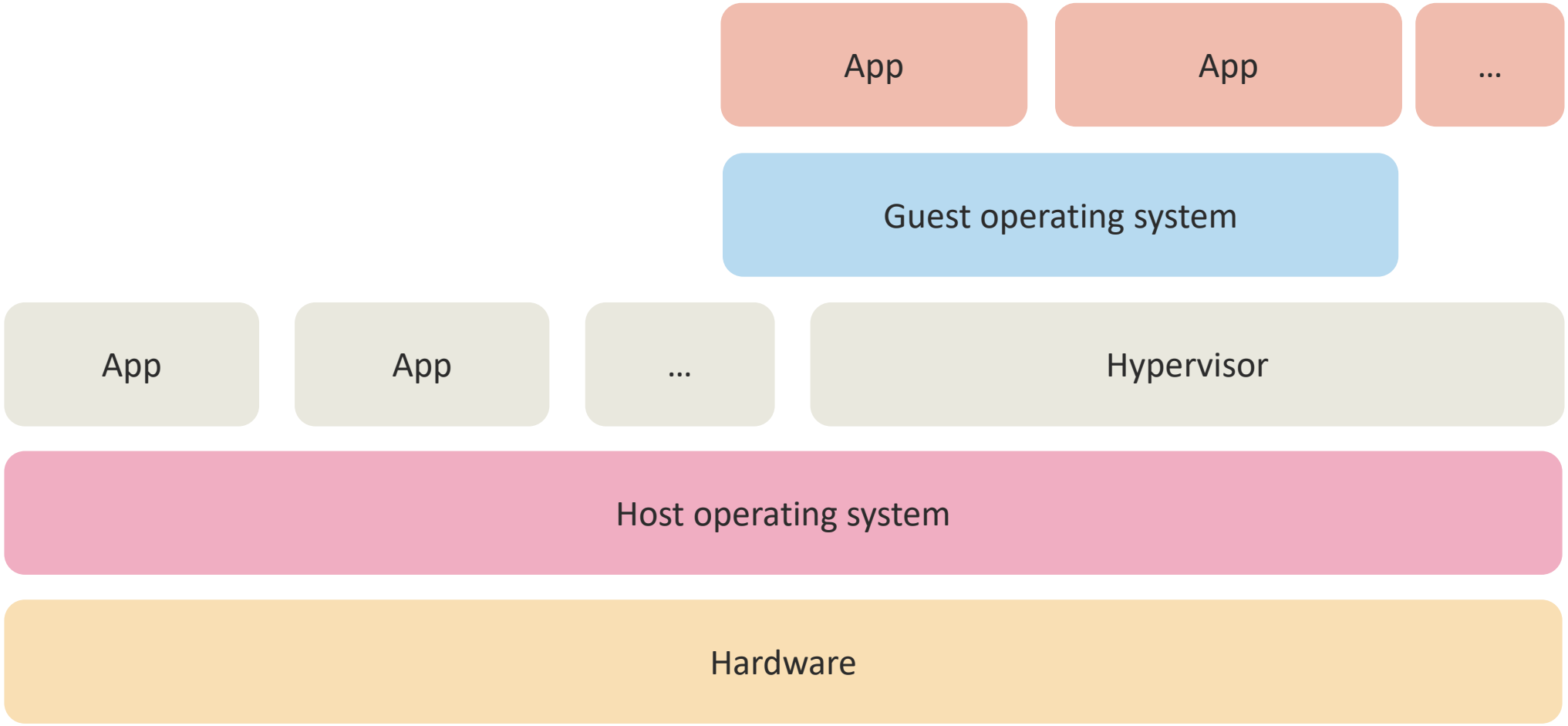
HYPERVISORS

- These are the virtual machine manager system and software that run one or more virtual machines
- It controls the interaction between the VMs and the underlying hardware
- Type I – bare metal or native
  - Runs directly on the underlying hardware
  - XenServer, KVM, Hyper-V, ESXi
- Type II – hosted
  - Runs on the OS installed on the hardware
  - Oracle VirtualBox 6, VMWare Player/Workstation

TYPE 1 HYPERVISORS



TYPE 2 HYPERVISORS



SUPERVISORY CONTROL AND DATA ACQUISITION (SCADA)

- Supervisory Control and Data Acquisition (SCADA) systems represent the software used to collect and send data to throughout facility systems and to cloud services
- Programmable logic controllers (PLCs) and other embedded systems are common hardware components
- Systems that are not air-gapped introduce various threats

INDUSTRIAL CONTROL SYSTEMS (ICS)

- Industrial control system (ICS) is a combined term that represents varied forms of control systems and related instrumentation, which include the devices, systems, networks, and controls used to operate and/or automate industrial and mechanical processes
- Each ICS typically functions differently and is built to electronically manage tasks efficiently
- Modern devices and protocols used in an ICS are used in nearly every industrial sector and critical infrastructure

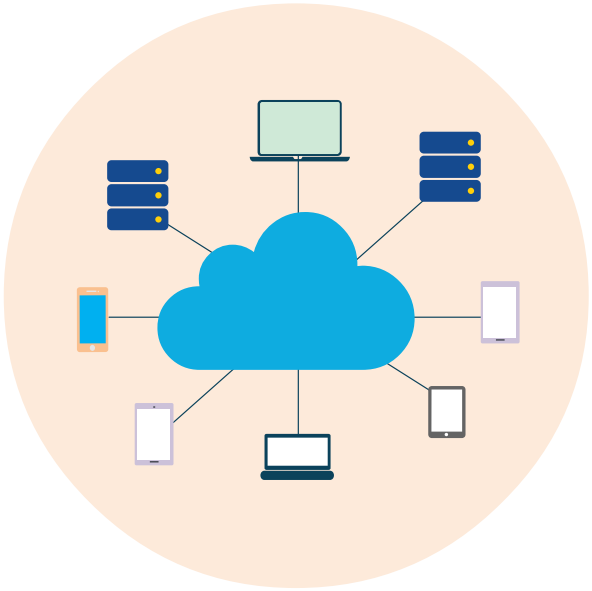
SCADA AND ICS SYSTEMS

- Facility and manufacturing control and management systems
- Water management systems
- Electric/nuclear power grid, solar, and wind farms
- Traffic signals and mass transit systems
- Environmental and manufacturing control systems

INTERNET OF THINGS

- The term IoT refers to the collective network of connected devices and the technology that facilitates communication between devices and the cloud, as well as between the devices themselves
- With the advent of inexpensive computer chips and high bandwidth networking, there are now billions of devices connected to the Internet using IPv4 and IPv6
- Everyday devices like toothbrushes, vacuums, cars, and machines can use sensors to collect data and respond intelligently to users

INTERNET OF THINGS



- Mobile devices
- Cameras
- Farm and ranch equipment
- Sensors
- Smart appliances
- Facility automation
- Medical devices and systems
- Vehicles and aircraft (drones)
- Smart meters
- Embedded devices and real-time operating systems (RTOS)