DI-FCT-UNL
Segurança de Redes e Sistemas de Computadores
*Network and Computer Systems Security*

Mestrado Integrado em Engenharia Informática
MSc Course: Informatics Engineering

1º Sem, 2020/2021

# 1. Introduction (Part I)

# Part I - Initial Concepts and Terminology

# What is a Secure System ?
# How to Define a Secure System ?

Suggested readings in provided biblkiography:

- Stallings, Network Secuirty Essentials – Applications and Standards, Ch.1, § 1.1
- Stallings, Computer Systems Security – Principels and Paradigms, Ch.1, § 1.1, § 1.2

# How to define a "Secure System" ?

**Possible definition :**

A system that never revealed vulnerabilities or a system that operates and has never been subject to any attack

Intrinsically or paradoxically, this definition says that …

TEHERE ARE NO SECURE SYSTEMS !
IMPOSSIBILITY !

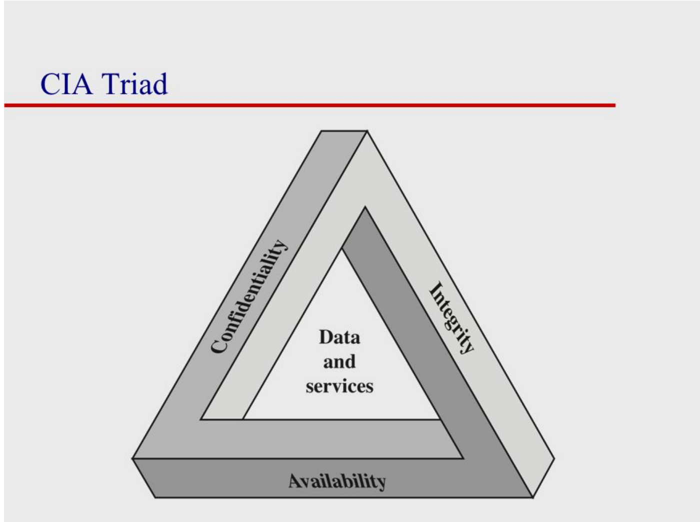
☹ Why ?

☹ Ok … this definition is not interesting ..!

# Another definiton …

Defnition from NIST (in Compjuter Security Handbook, 1995) , referred in W. Stallings, Network Security Essentials – Applications and Standards, Ch. 1, 1.1 – Computer Security Concepts

The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability, and confidentiality of information system resources (includes hardware, software, firmware, information/ data, and telecommunications).



Protection:

**Security services and related security mechanisms** designed, implemented and operating in the system as countermeasures (or security guarantees) against potential threats and attacks from adversaries (or opponents)

Objectives (as Security Objectives) in a Main Triad: C,I,A – using here the FIPUS PUB 199 Framework):

Main objectives, achieved by providing the system with the correspondent Security Properties (that must be implemented as the System Security Services)

Scope of Assets (or Security Dimensions):

Computer Systems (Resources) and Internetworking (Communication) Resources

# C.I.A Triad: FIPS PUB 199 (by NIST)

## Main objectives + sub-categorizations and related requirements

Data Processing Confidentiality

Confidential Computation (Confidential Computing)

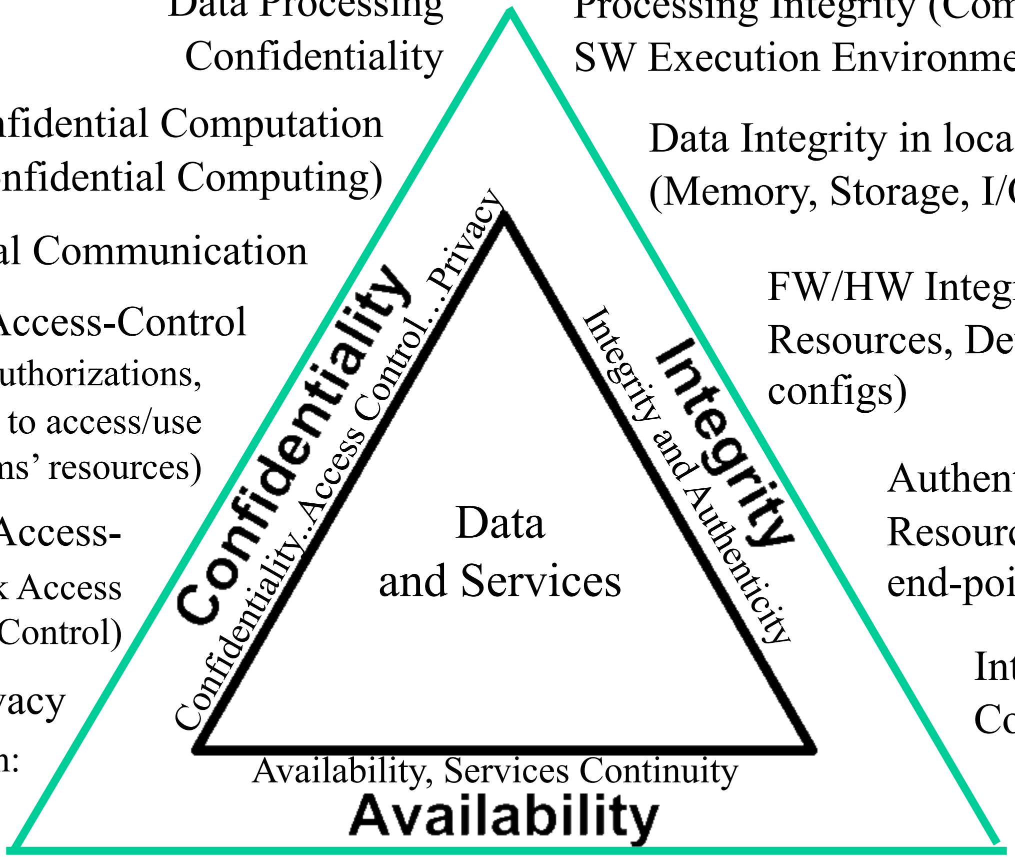
Confidential Communication

Access-Control (Authorizations, Permissions to access/use Systems' resources)

Communication Access-Control (Network Access Control)

Privacy

Privacy Preservation: Data and Computation Privacy, Private Communication, Anonymization (ex., for Censorship Circumvention)


Confidentiality
Confidentiality..Access Control...Privacy
Integrity
Integrity and Authenticity
Data
and Services
Availability, Services Continuity
Availability

Processing Integrity (Computations: SW Execution Environment)

Data Integrity in local processing (Memory, Storage, I/O)

FW/HW Integrity (Ex., HW Resources, Devices, BIOS-configs)

Authenticity of System Resources (components, end-points)

Integrity of Data in Communication Flows)

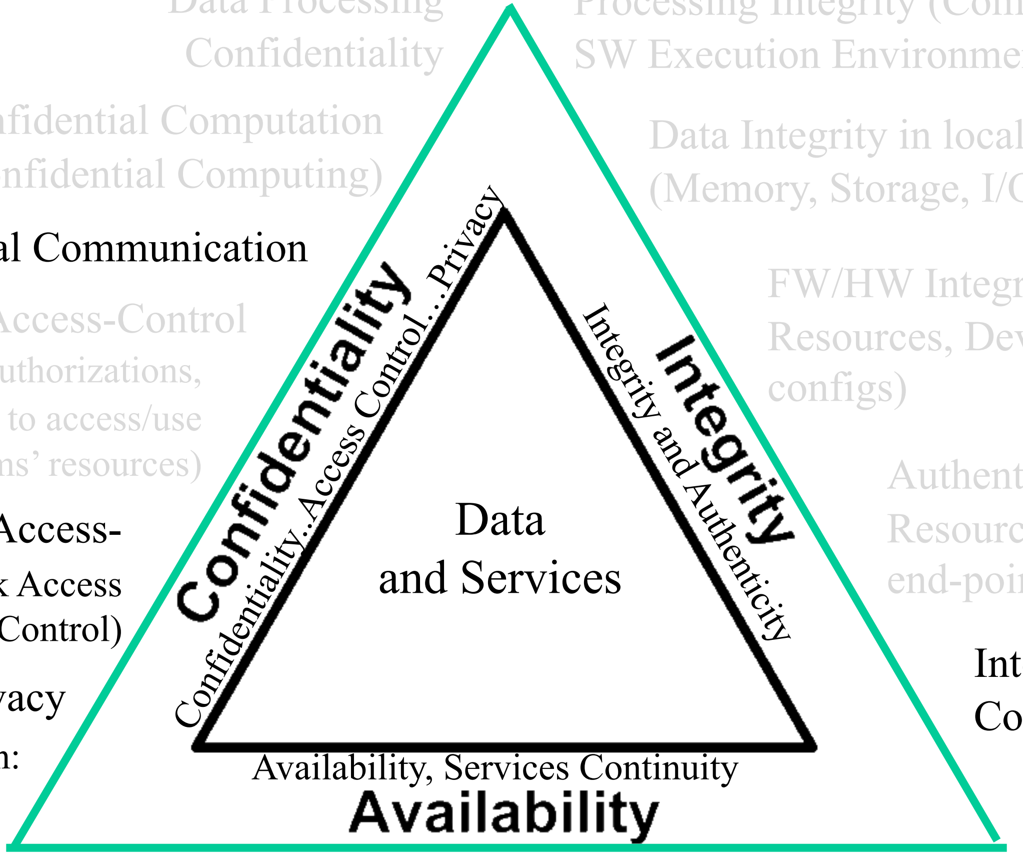
Authenticity of Data in Communication Flows and Communication Endpoints

Availability of Computer Systems' Resources and Services

Availability of Communication Services (Networks, Communication Channels)

# C.I.A Triad: FIPS PUB 199

## Communications Security: Secure Channels


Confidentiality
Confidentiality..Access Control...Privacy
Integrity
Integrity and Authenticity
Data
and Services
Availability, Services Continuity
Availability

Data Processing Confidentiality

Confidential Computation (Confidential Computing)

Confidential Communication

Access-Control (Authorizations, Permissions to access/use Systems' resources)

Communication Access-Control (Network Access Control)

Privacy

Privacy Preservation:

Data and Computation Privacy,

Private Communication, Anonymization (ex., for Censorship Circumvention)

Processing Integrity (Computations: SW Execution Environment)

Data Integrity in local processing (Memory, Storage, I/O)

FW/HW Integrity (Ex., HW Resources, Devices, BIOS-configs)

Authenticity of System Resources (components, end-points)

Integrity of Data in Communication Flows)

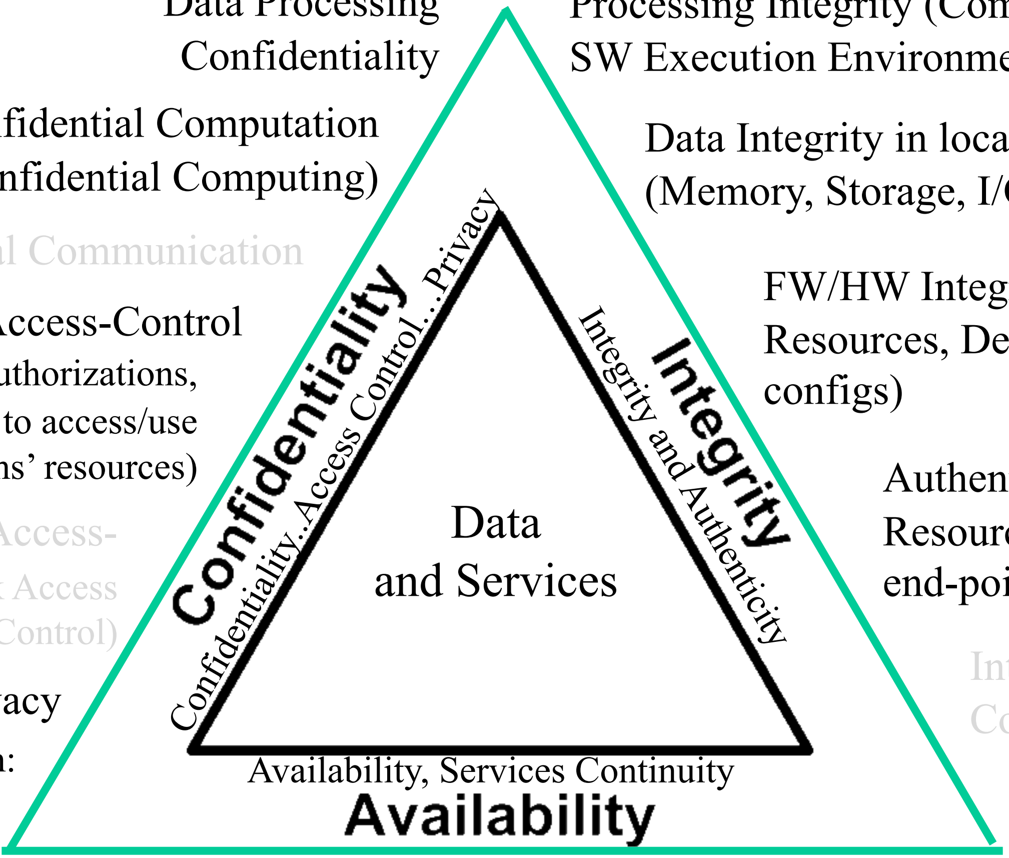
Authenticity of Data in Communication Flows and Communication Endpoints

Availability of Computer Systems' Resources and Services

Availability of Communication Services (Networks, Communication Channels)

# C.I.A Triad: FIPS PUB 199

## Systems Security



Confidentiality
Confidentiality..Access Control…Privacy
Integrity
Integrity and Authenticity
Data
and Services
Availability, Services Continuity
Availability

Data Processing
Confidentiality

Confidential Computation
(Confidential Computing)

Confidential Communication

Access-Control
(Authorizations,
Permissions to access/use
Systems' resources)

Communication Access-
Control (Network Access
Control)

Privacy

Privacy Preservation:
Data and
Computation Privacy,
Private Communication,
Anonymization (ex., for
Censorship
Circumvention)

Processing Integrity (Computations:
SW Execution Environment)

Data Integrity in local processing
(Memory, Storage, I/O)

FW/HW Integrity (Ex., HW
Resources, Devices, BIOS-
configs)

Authenticity of System
Resources (components,
end-points)

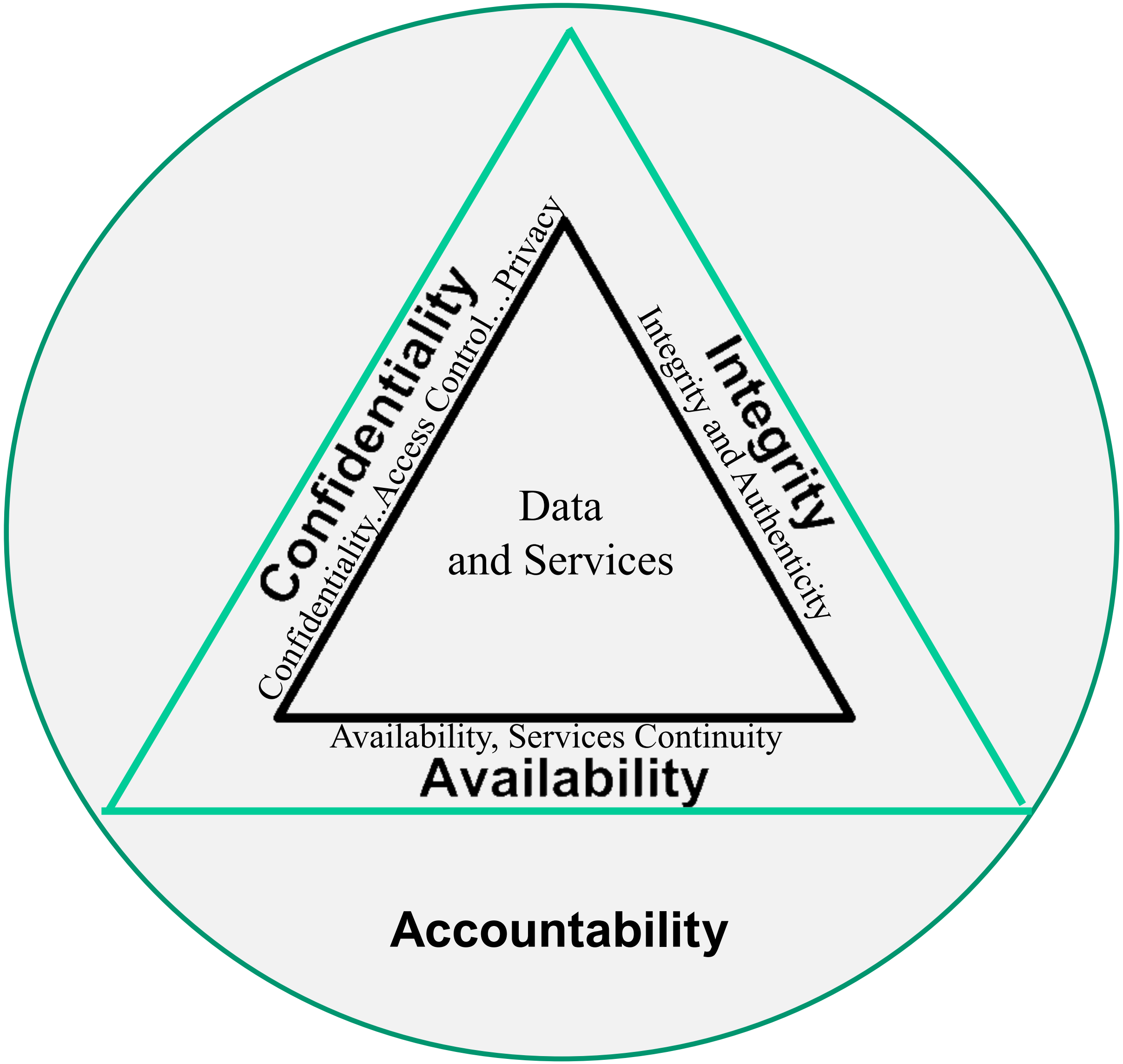
Integrity of Data in
Communication Flows)

Authenticity of Data in
Communication Flows
and Communication
Endpoints

Availability of Computer Systems' Resources
and Services

Availability of Communication Services (Networks,
Communication Channels)

# Secure Systems must be Managed and Audited!


Confidentiality
Confidentiality..Access Control…Privacy
Integrity
Integrity and Authenticity
Data
and Services
Availability, Services Continuity
Availability
Accountability

Security Logging

Forensics

Security Auditing and Monitoring

Security Assessment

Verification of Security Compliance

Patching/Reparing, Maintenance

© DI/FCT/UNL, Henrique João L. Domingos

# Security by Design vs. Operational Security

**Security by Design:**

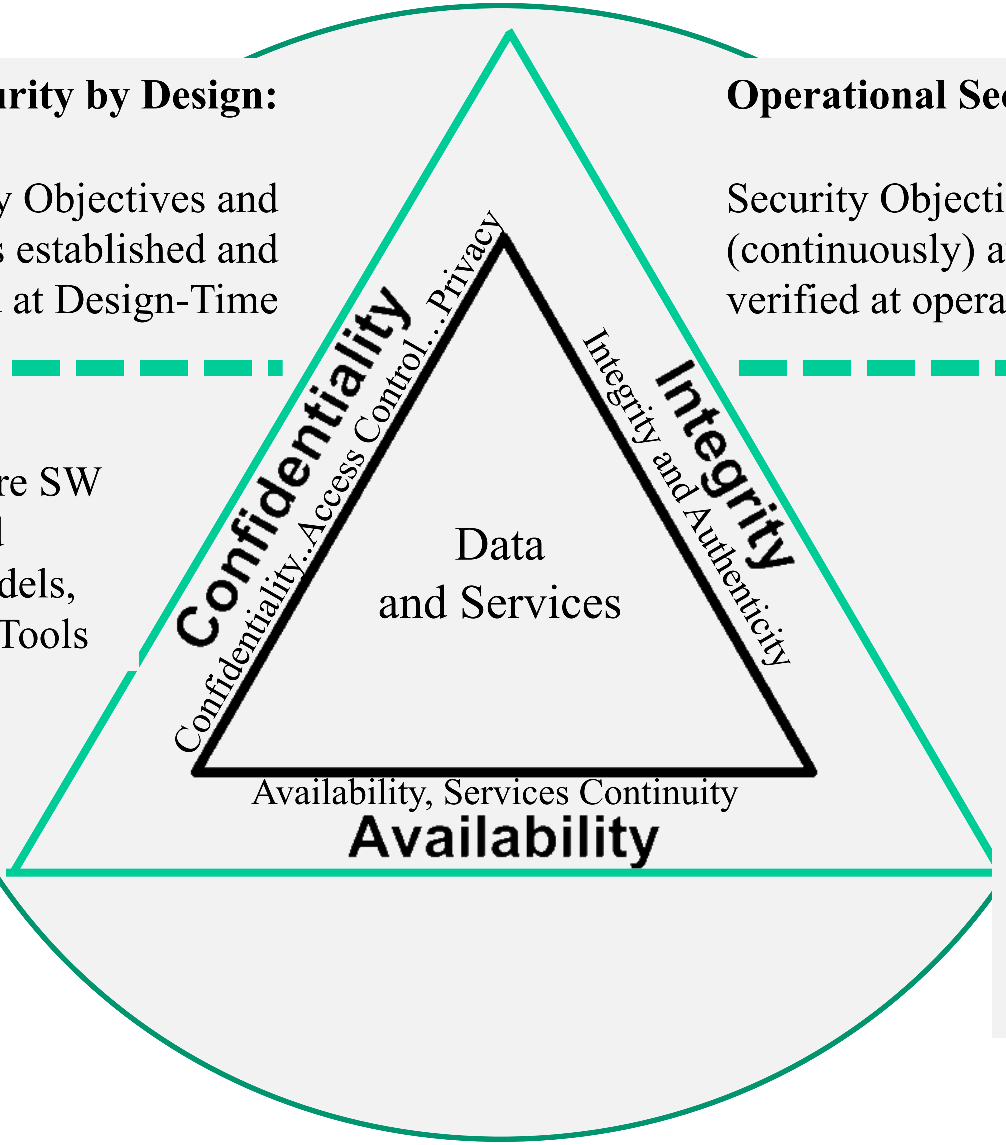
Security Objectives and Properties established and verified at Design-Time

**Operational Security**

Security Objectives and Properties (continuously) audited, monitored and verified at operation (run) time

Security Design Techniques, Secure SW Programming and Development Models, Mechanisms and Tools

Operational security, involving (continuous) good practices, correct deployments and configurations or parameterizations, and use of white/grey or black auditing/monitoring and vulnerability assessment tools, during the entire cycle of the system operation


Confidentiality
Confidentiality..Access Control…Privacy
Integrity
Integrity and Authenticity
Data and Services
Availability, Services Continuity
Availability

# Security by Design vs. Operational Security
# Security is a Process …

## Design Time and Development (Security By Design)

- SW/FW/HW Design, Development Methods and Tools
- SW Security Foundations, Programing Languages and Runtimes
- SW Security, Static Analysis
- Minimization and isolation of TCBs and Trusted Execution Environments (ex., Isolation, Virtualization Containerization, Trusted Execution Environments, Trusted Computing Modules, HW-Shielded Solutions)

## Operational Security (Runtime, Security as a Process)

- Verification and maintenance of correct operation in the op. lifecycle
  - Detection/correction of errors and defects, hardening & patching,
- Security Auditing and Dynamic Analysis (runtime): Inspection Methods and Verification/Auditing Tools
- Mix of White-Box (Code Inspection), Gray-Box, Black Box Approaches
  - Ex. PEN Testing and Evaluation; "The defender" leaning and performing as an adversary"
- (Continuous) Identification of Potential Vulnerabilities and revision of Adversary Model Assumptions and Exposed Attack Surfaces

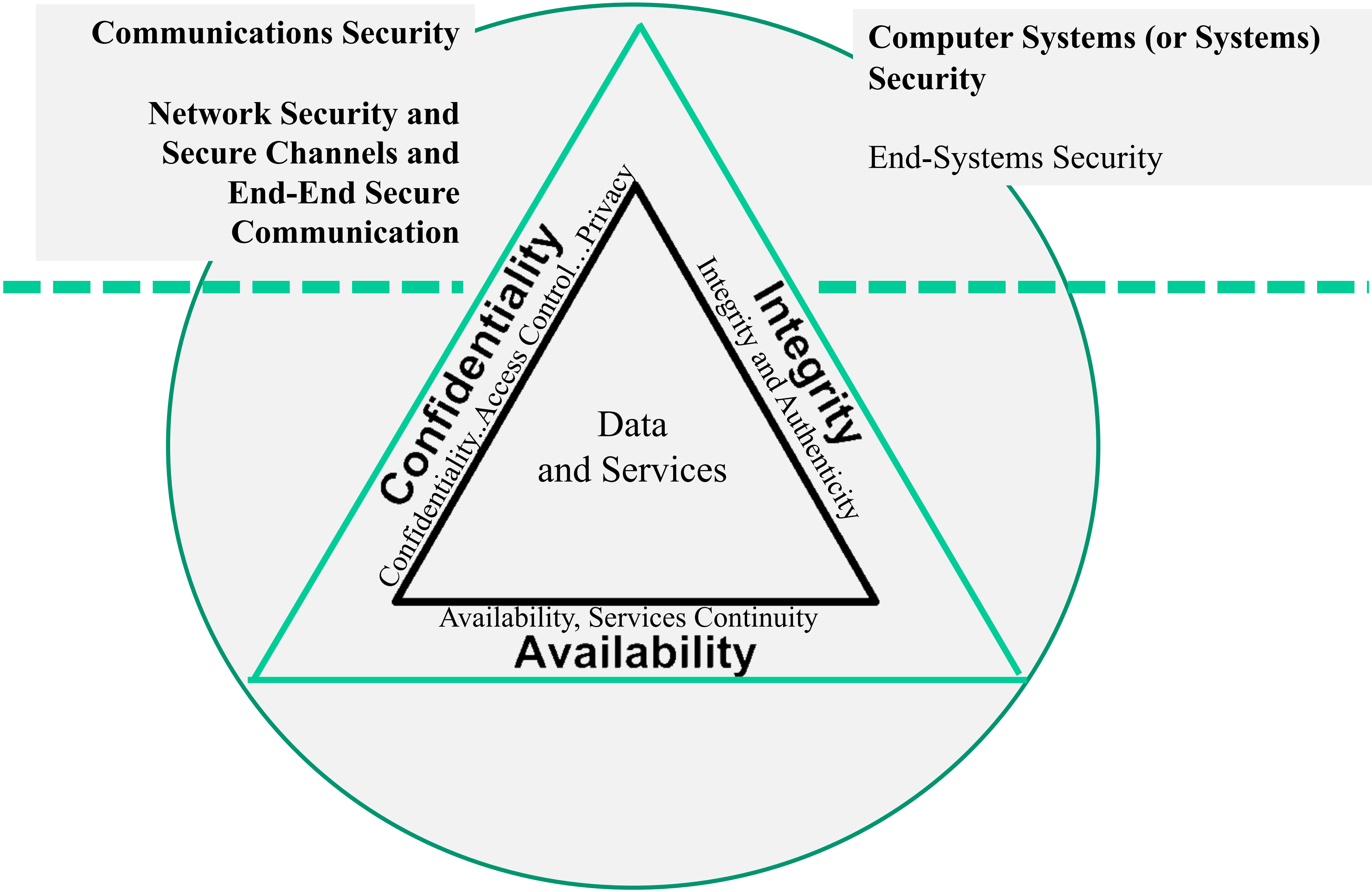
# Commnications Security vs. Computer Systems Security

**Communications Security**

**Network Security and Secure Channels and End-End Secure Communication**

**Computer Systems (or Systems) Security**

End-Systems Security



Confidentiality

Confidentiality..Access Control…Privacy

Integrity

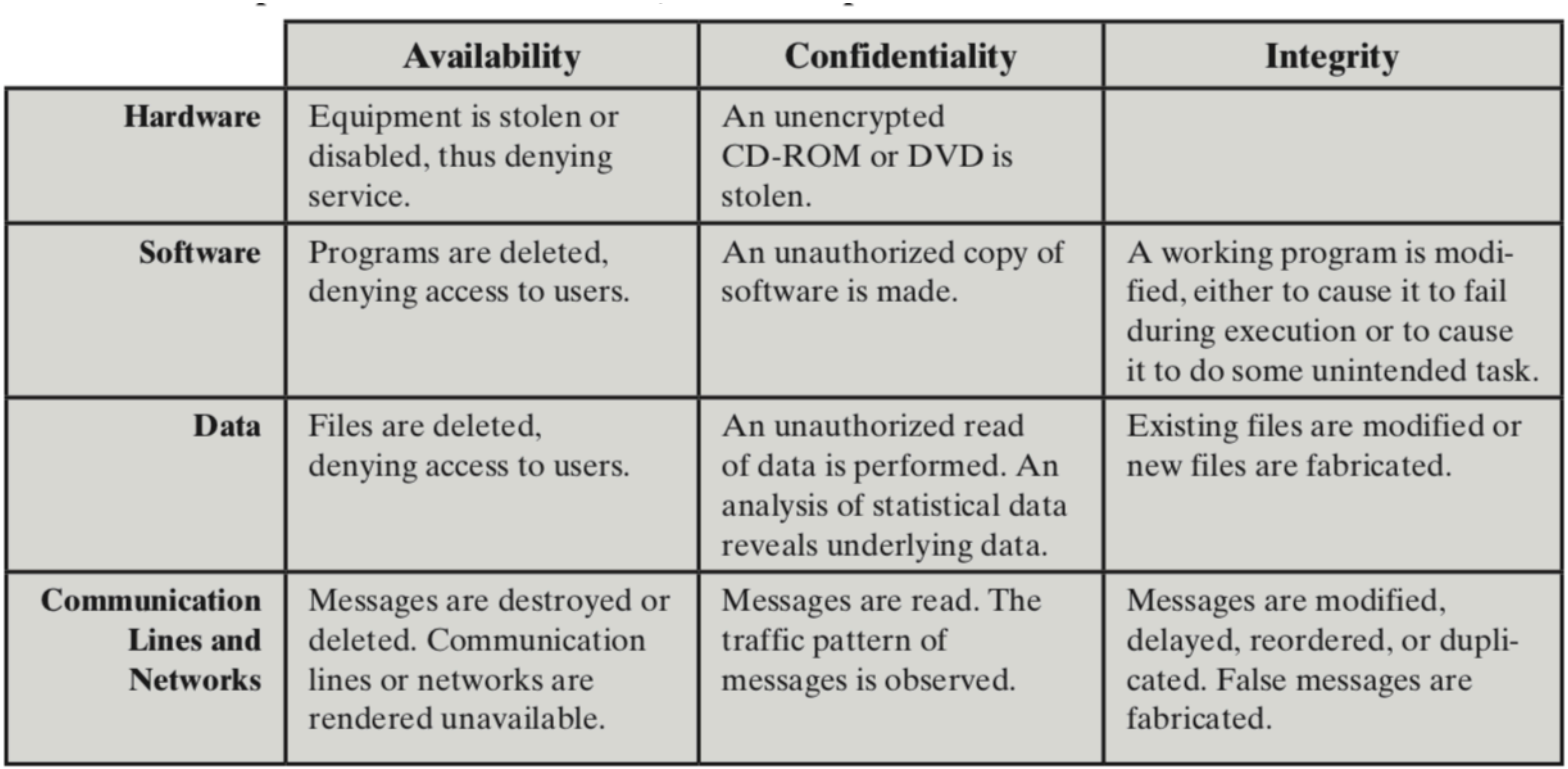
Integrity and Authenticity

Data
and Services

Availability, Services Continuity

Availability

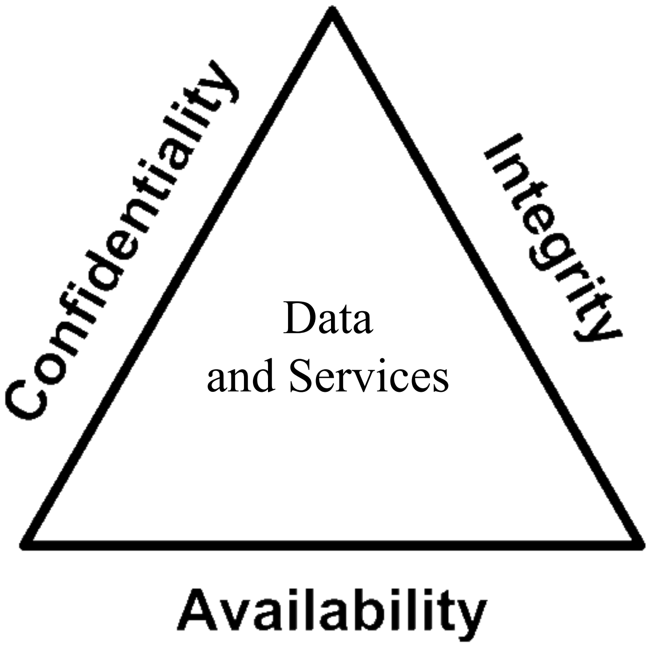
# Computer vs. Network Security



| | Availability | Confidentiality | Integrity |
|---|---|---|---|
| **Hardware** | Equipment is stolen or disabled, thus denying service. | An unencrypted CD-ROM or DVD is stolen. | |
| **Software** | Programs are deleted, denying access to users. | An unauthorized copy of software is made. | A working program is modified, either to cause it to fail during execution or to cause it to do some unintended task. |
| **Data** | Files are deleted, denying access to users. | An unauthorized read of data is performed. An analysis of statistical data reveals underlying data. | Existing files are modified or new files are fabricated. |
| **Communication Lines and Networks** | Messages are destroyed or deleted. Communication lines or networks are rendered unavailable. | Messages are read. The traffic pattern of messages is observed. | Messages are modified, delayed, reordered, or duplicated. False messages are fabricated. |

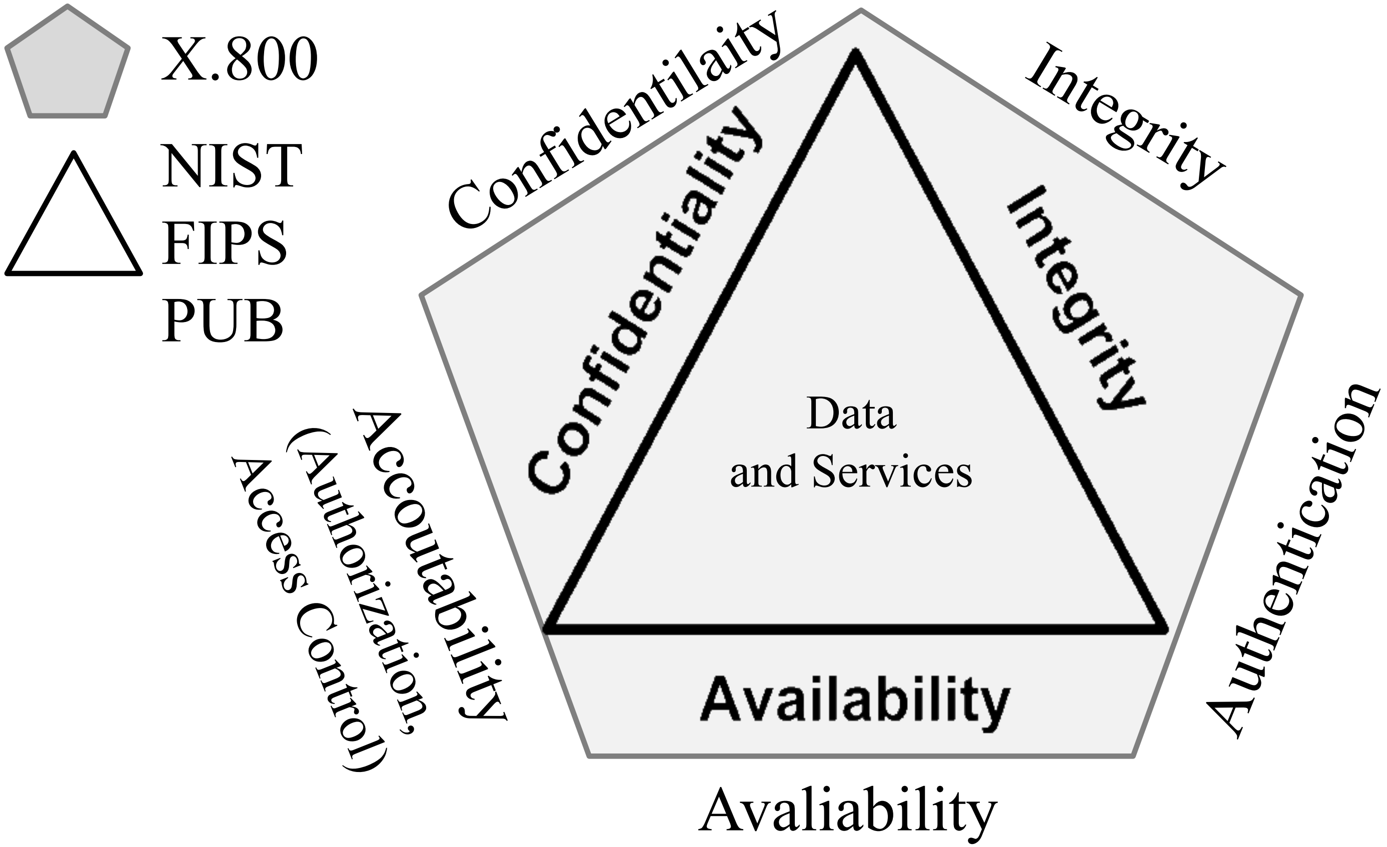
# C.I.A Triad: FIPS PUB 199 (by NIST)

## Main objectives + sub-categorizations and related requirements

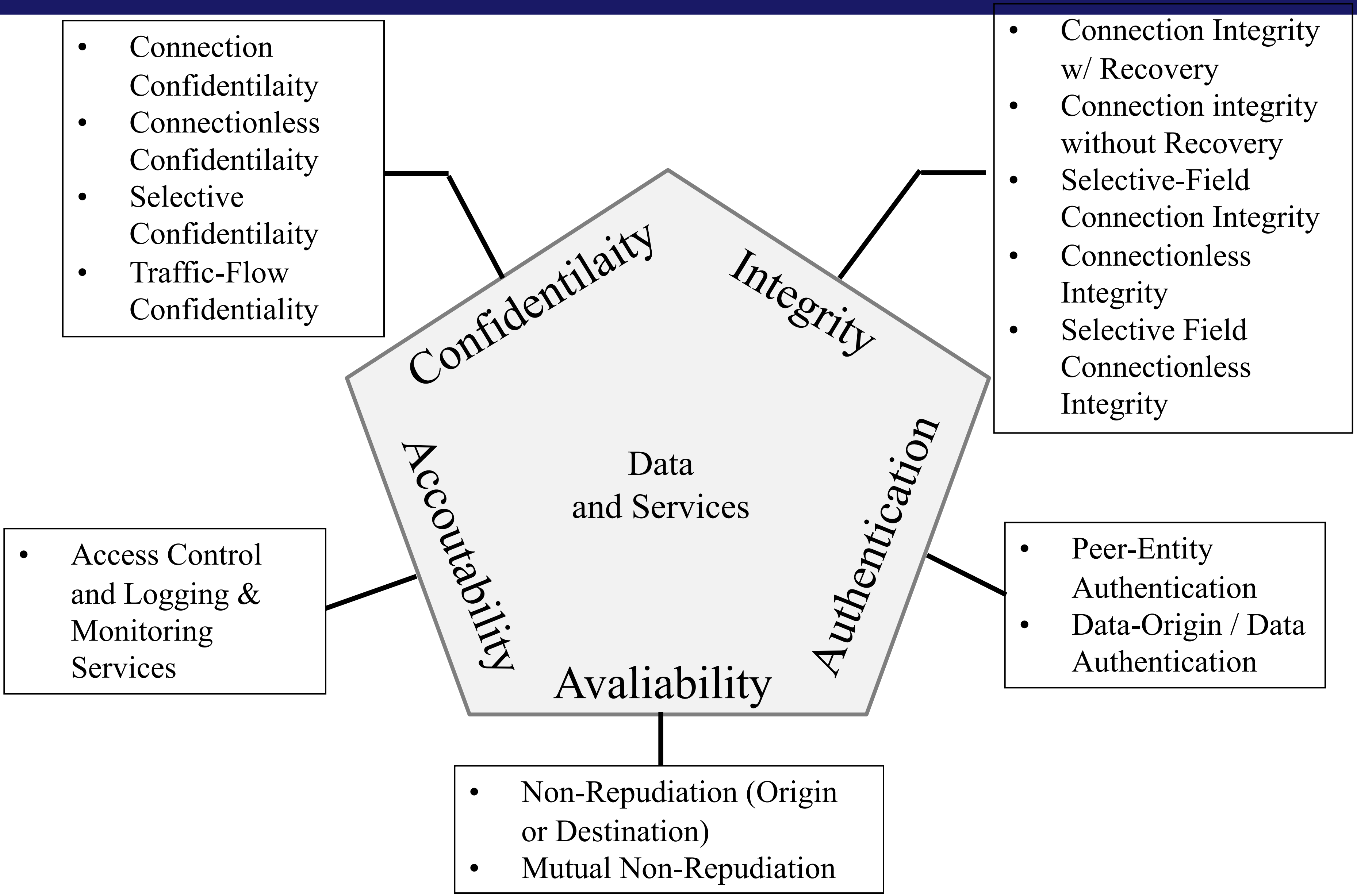


Objectives in FIPS PUB Security Framework involves different security properties (sometimes defined as "root-properties" and "sub-instanted notions" in other frameworks) Here we have a mapping with security prpeties in the ISO X.800 Security Framework (as we will present later) and also IETF RFC 4949 (IETF Internet Security Glossary) and IETF Interet Standards and Terminology

# C.I.A Triad: FIPS PUB 199 (by NIST) Framework vs. OSI X.800 Framework  (and also IETF RFC 4949)



# X.800 Sub-Instantiations of Security Services



- Connection Confidentilaity
- Connectionless Confidentilaity
- Selective Confidentilaity
- Traffic-Flow Confidentiality

- Connection Integrity w/ Recovery
- Connection integrity without Recovery
- Selective-Field Connection Integrity
- Connectionless Integrity
- Selective Field Connectionless Integrity

- Peer-Entity Authentication
- Data-Origin / Data Authentication

- Non-Repudiation (Origin or Destination)
- Mutual Non-Repudiation

- Access Control and Logging & Monitoring Services

# What is a Secure System: Definition for the CSNS Course Approach

# Redefining a "Secure System" for our CNCS Course ?

**Secure System (in the context of the CSNS Course):**

A System designed with secure objectives addressed as verifiable security properties implemented by security services built from valid security mechanisms afforded to attain the applicable objectives of preserving authentication, integrity, confidentiality, access-control, privacy and accountability in protecting principals, information and computation assets, including HW, SW, FW, Data and Communication Channels.

**Security services** are designed to implement countermeasures against attack vectors (in a defined attack typology and adversary model), to avoid vulnerabilities and to minimize risk, …

… we need a well-defined threat or adversary model, the identification of the system security surface and know about the correct security mechanisms that must be established in well-identified, verifiable and minimized trust computing base (TCB) assumptions

# Redefining a “Secure System” ?

**Secure System (in the context of the CSNS Course):**

A System designed with secure objectives addressed as verifiable security properties implemented by security services built from valid security mechanisms afforded to attain the applicable objectives of preserving authentication, integrity, confidentiality, access-control, privacy and accountability in protecting principals, information and computation assets, including HW, SW, FW, Data and Communication Channels.

**Security services** are designed to implement countermeasures against attack vectors (in a defined attack typology and adversary model), to avoid vulnerabilities and to minimize risk, …

… we need a well-defined threat or adversary model the identification of the system security surface and know about the correct security mechanisms that must be established in well-identified, verifiable and minimized trust computing base (TCB) assumptions

# Security vs. Risk and Risk Mitigation

Thinking on RISK:

Security as the Minimization (or Mitigation) of Risk

RISK: Measure of the extent to which system / system entity is vulnerable and threatened by a potential threat circumstance: a measure of the possibility of a vulnerability exploitation (attack)

RISK = VULNERABILITIES x THREAT-Potential

RISK (t) = VULNERABILITIES(t) x THREAT-Potential (t)

# Important principles

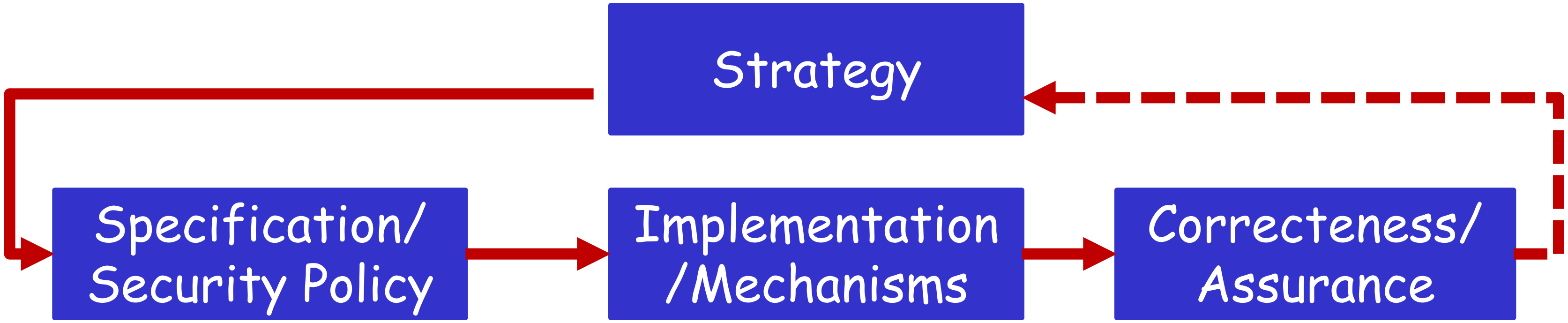
- No Security by Obscurity Policies
  - Sometimes a temptation, not necessarily wrong
  - But risky … not the most interesting approach
  - The more studied, published, known, scrutined, audited a security solution is, resisting all attempts to discover vulnerablities, the more secure it is !

- Security is a Process (not an end)
  - Must be analyzied during all the life-cycle of systems, from the design-time and during all the operation time (until the system will be declared as obsolete and no more secure)

- Interesting mind setting: be paranoid ! Try to think as your adversary… She/He can have lots of advantages !
  - This is way we need adversary models in design time and systems' auditing and vulnerablity assessments in operation time

# Computer Security Strategy

# Overall Generic Strategy



Strategy

Specification/ Security Policy

Implementation /Mechanisms

Correcteness/ Assurance

**Security Plans**

- Assets and value of assets
- Vulnrabilities
- Potential threats and attacks

- - - - -

**Tradeoffs and Risk Management:**

- Security vs. Usability
- Cost of security vs. Cost of Incidents, failures and recoveries
- Risk management tradeoffs

**Security services and typology of mechanisms**

- Prevention
- Detection
- Response
  - System-Based
  - Human-Operation processes
  - CERT
- Recovery
  - Reactive Recovery
  - Pro-active Recovery
  - Tolerance (Attack/ Intrusion/Fault/ Tolerance)

**Assurance**

Confidence on security policy enforcement

- Design
- Implementation
- Operation

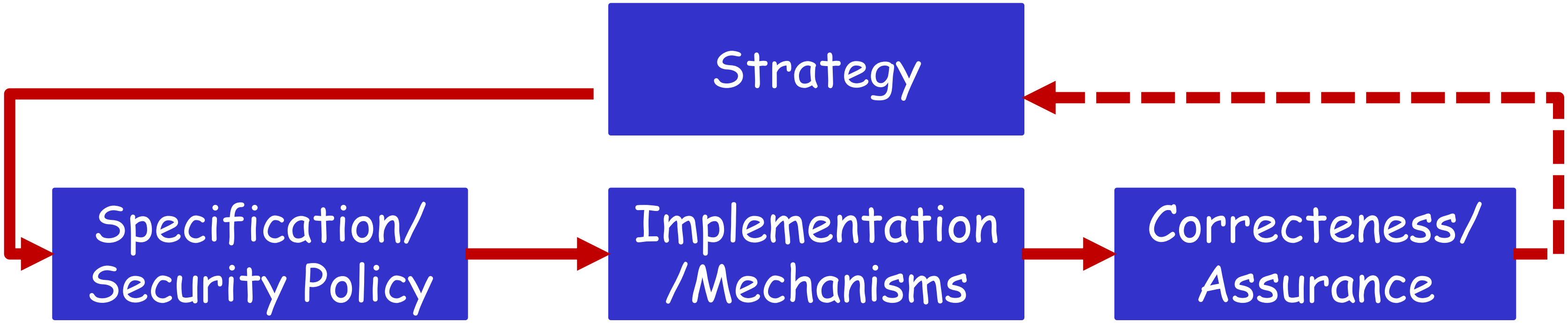
**Compliance**

- Law, Ethics, Regulations, Organizational Rules, …

**Evaluation (Assessment)**

- Testing
- Vulnerabilty assessment
- Auditing and Monitoring

# Strategy vs Security Frameworks and Standards



Strategy

Specification/ Security Policy

Implementation /Mechanisms

Correcteness/ Assurance

**ISO 17999 – 27001,**
**ISO/IEC Standards**

**ITU-T**

**OSI X.800**

**NIST / FIPS-PU**

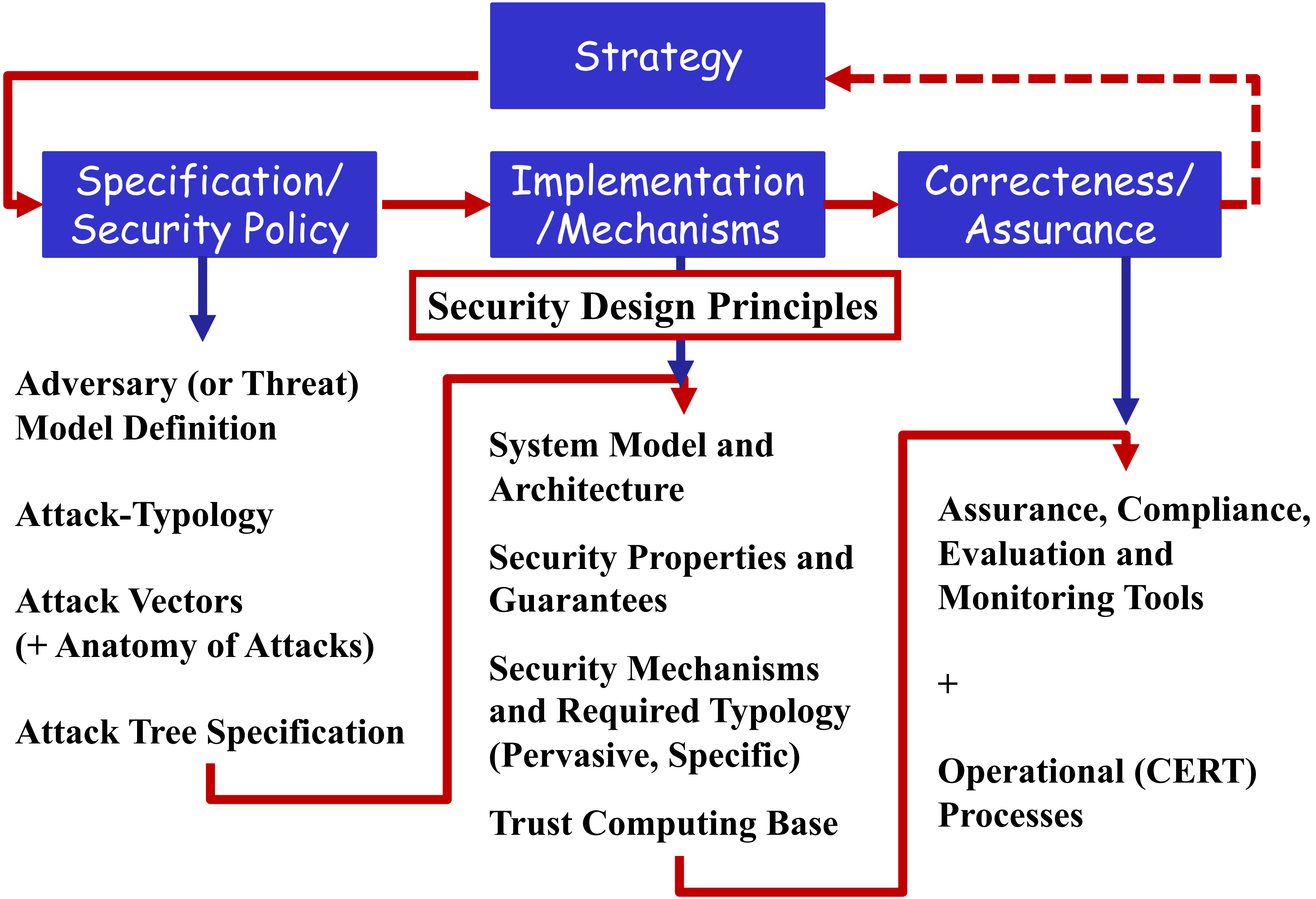
**ISOC / IETF (RFCs)**

Organizational Security Standards and Frameworks

Technical (Engineeing) Frameworks Standards

Operational Security Solutions and Technology

# Approach Model for Computer Networks and Systems Security Engineeirng



Strategy

Specification/ Security Policy

Implementation /Mechanisms

Correcteness/ Assurance

Security Design Principles

Adversary (or Threat) Model Definition

Attack-Typology

Attack Vectors
(+ Anatomy of Attacks)

Attack Tree Specification

System Model and Architecture

Security Properties and Guarantees

Security Mechanisms and Required Typology (Pervasive, Specific)

Trust Computing Base

Assurance, Compliance, Evaluation and Monitoring Tools

+

Operational (CERT) Processes

# Important Concepts, Definitions and Terminology

Suggested readings in provided biblkiography:

- Stallings, Network Secuirty Essentials – Applications and Standards, Ch.1, § 1.1
- Stallings, Computer Systems Security – Principels and Paradigms, Ch.1, § 1.1, § 1.2

# Initial Important Concepts, Notions and Terminology

- Security vs. Risk vs. Vulnerability
- Threats vs. Attacks vs. Incidents
- Passive Attacks vs. Active Attacks
- Insider Attacks vs. Outsider Attacks
- Owners/Subjects, Risk, Countermeasures, Asset, Threats and Threat Agents: relation between these notions
- Attacks Typology
- Notion of "Principal" (or Subject)
- Attack Surface and Attack Trees
- Adversary (or Threat) Model Definition
- Trust Computing Base
- Computer Security Strategy and Security Policies

Suggested readings in provided biblkiography:

- Stallings, Network Secuirty Essentials – Applications and Standards, Ch.1, § 1.1
- Stallings, Computer Systems Security – Principels and Paradigms, Ch.1, § 1.1, § 1.2

# Threat Agents

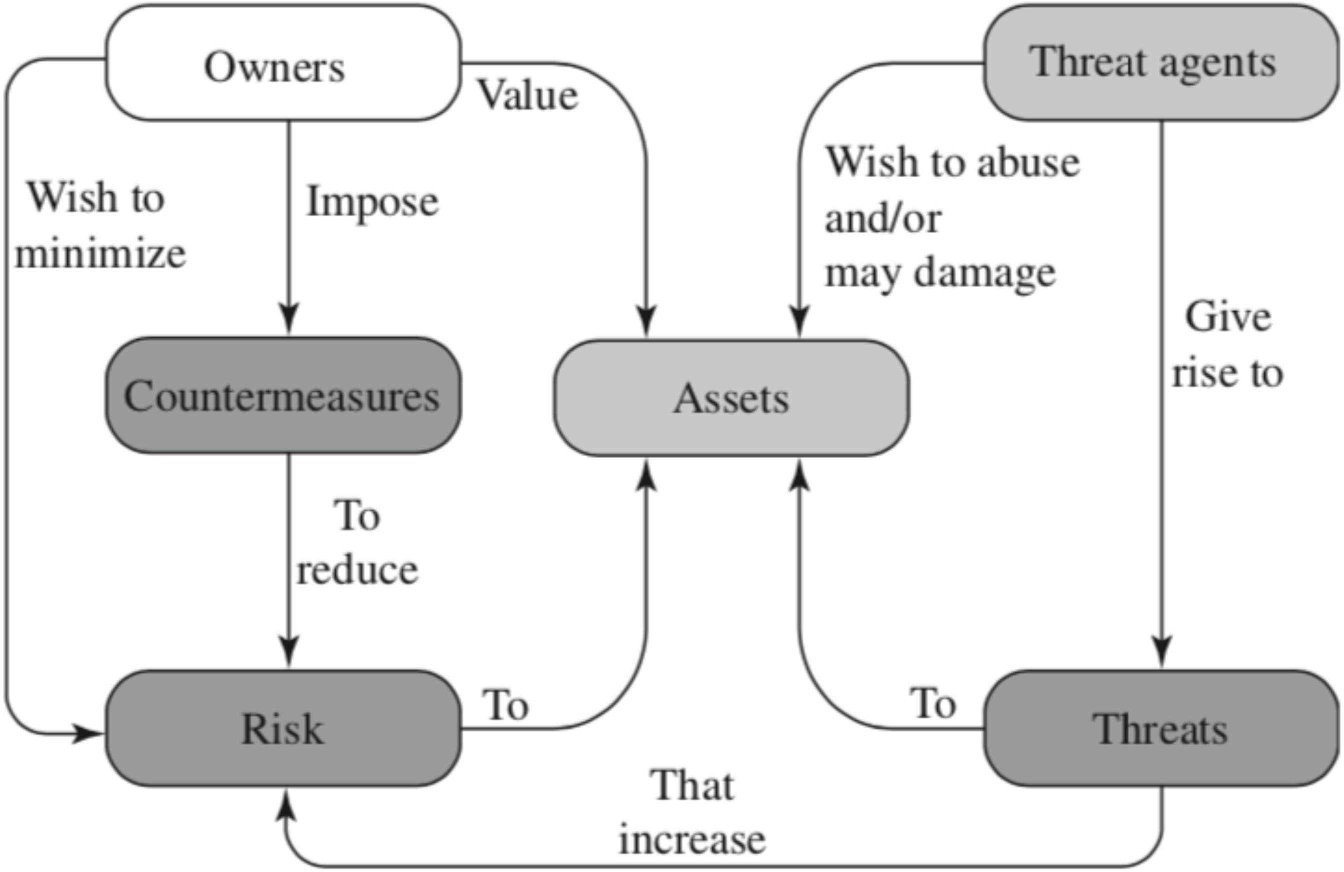
- Threat agents, adversries or opponents
  - Agents (entities) conducting intentions for detrimental and illicit activities against systems and system's resources
  - Attackers: threat agent that conducted concrete actions that materialized a potential threat, exploiting a vulnerability
  - Who are the threat agents or attackers ?
    - Individuals, groups, organizations, governments
      - Can use different tools (expolits) used as instruments or means for attack vectors
  - But also …
    - Users (unconscious actions, misuses, abuse of privileges, naïve-actions, careless-operations/actions)

# Vulnerabilities and Threats

- Vulnerablity
  - A weakness in a system, procedures, intrenal controls or implementation flaws that can be exploted or triggered by a threat source
- Threat
  - Aany circumstance or event with the potential to adverserly impact computer and information systems and related resources (**as assets**) and systems' operations, causing the risk of incorrect behaviour
- Countermeasures
  - Devices, techniques, mechanisms, and services used/combined in order to counter the intentions of threat or the realization of attacks

# Owners/Subjects, Risk, Asset, Threats and Threat Agents

- Ref. IETF RFC 4949, Internet Security Glossary
- Relations between the notions



# Threats vs. Attacks and Incidents

Attacks

- Concretizations of threats

Incidents

- Manifestation of Attacks (can have different levels of severity and damage)

Attacks can be perceived or not

- Perceived: we say "detected" (detected incidents)
- Not perceived: We say Silent, Non-Detected (possibly not perceived incident)

Attacks can take place in exploring a vulnerability not known before the time of the attack Also Known as exploitations of Zero-Day Vulnerabilities (so ... Possibly we can not have a remedy during sometime)

# Attacks vs. Failures

Failure: a manifestation of non-correct or unexpected behavior

- A failure is a deviation of the correct expected state of the system, as defined in the system model and design specifications

We can think on "attacks" as "failures"

- Attacks can induce an incorrect (unexpected) behavior by exploiting a vulnerability (seen as a potential fault)

We can distinguish between:

- Accidental failures
  - Accidental factors (or accidental vectors)
- Induced or injected failures (more related to Attacks)
  - Caused by "Attackers" or "Opponents" (or Attack Vectors, or Malicious-Vectors)

# Attacks vs. Failures (more)

Attacks, (as failures) can be:

- Non-Detected (no manifestation of incidents)
- Detected without Recovery
  - Ex., Fail-Stop Model / Stop after Attack
- Recovered after Detection (Reactive Recovery)
- Recovered without Detection (Pro-Active Recovery)
- Tolerated (Tolerance means the system can work correctly and resiliently, under availability conditions, even when attacked
  - Fault Tolerance vs. Attack Tolerance (ex., Intrusion Tolerance)

# Passive and Active Attacks

Attacks can also be characterized as:

- Passive Attacks
  - When the consequence don't affect the correct state or the correct behavior of the system) and does not affect systems' resources
    - Interesting: tend to be easier to defend, but harder to detect

- Active Attacks
  - When the consequences modify the correct behavior of the system with alteration of systems' resources and operation
    - Interesting: sometimes easier to defend, but harder to detect

# Insider and Outsider Attacks

Attacks also can be characterized as:

- Insider Attacks
  - When the attack origin (initiation) is from an entity inside the security perimeter
    - Ex., Abuse of Privileges by Authorized Entities, Malicious Use, No Approved Actions or Procedures in Systems' operation
    - Unconscious actions, bad-use/mistakes
    - Ex.; Passive Insider Attacks: Honest but Curious System Administrators
- Outsider Attacks
  - Attacks originated out of the system's perimeter by unauthorizes or illegitimate users
  - Lots of attacker/opponent and/or psychological profiles:
    - Amateur pranksters, script-kiddies, organized criminals (cybercrime), terrorists, hostile governments (cyberwar), and different *hat hackers (black, gray, red, ethical, … etc.)

# Attack Typology

- The definition of attack types, characterized in a conceptual framework

- The use of correct terminology for attack types is important for the understanding of the related definitions that are usually behind the design of security countermeasures: security properties, security services and security mechanisms

# Ex., Attack Typology (RFC 4949)

*Generic attack types (see Stallings, Computer Security – Principles and Practices, Chap. 1, § 1.2)*

- Data/Information (Unauthorized) Disclosure
- Exposure
- Interception
- Inference
- Intrusion
- Deception
- Falsification (Forging)
- Repudiation
- Disruption
- Incapacitation
- Corruption
- Obstruction
- Usurpation – Misappropriation or Misuse

# Ex., Attack Typology (ex., X.800)

*Base attacks (see Stallings, Network Sec. Essentials – Ch.1, § 1.3 - Security Attacks or Stallings – Computer Security, Ch.1, § 1.2)*

Emphasis on Communication Channels:

- Passive Attacks:
  - Release of Message Contents (Message Disclosure)
  - Traffic Analysis
- Active Attacks
  - Masquerading (can relate to other active attacks, as well as, Message Spoofing and Sybil Attacks)
  - Replay or Message Replaying
  - Modification of messages (Message Tampering)
  - Denial of Service (DoS)

# Notion of Principal (or Subject)

- A named entity, well identified in a system (Name/UID, in a certain level of abstraction) representing the ownership and responsibility of system resources and related management actions

- Different levels of abstraction for principals and subjects:
  - A Data-Link Endpoint Identifier (ex., Ethernet MAC Address)
  - A DNS FQN or an IP endpoint (Network Level)
  - An interprocess communication endpoint in a protocol (ex., <ip, port, transport protocol>
  - A WebServer or WebService Endpoint (URL)
  - An Object Component identified by an ObjectID (UID)
  - A Content-Addressable Resource
  - A User (username/user ID)

# Attack Surfaces

- All the reachable and explotable vulnerabilities in a system
- Examples:
  - TCP and UDP Open Ports
  - Servioces Available on the Inside of Firewalls
  - Code processing or dispatching requests, involving any content: Data, Email-Messages, XML/SOAP messages, Protocol Payloads, Documents or any Specific Echange Data Formats and Messages
  - Interfaces
  - APIs, Web Forms, SQL Query/Response Endpoints (in DB Engines)
  - Users (ex.,Users, System Administrators, Employes) with access to sensitive information that can act as targets for social engineering threats
  - Applications and Services that can be used as indirect targets for threats in other systems or systems' components

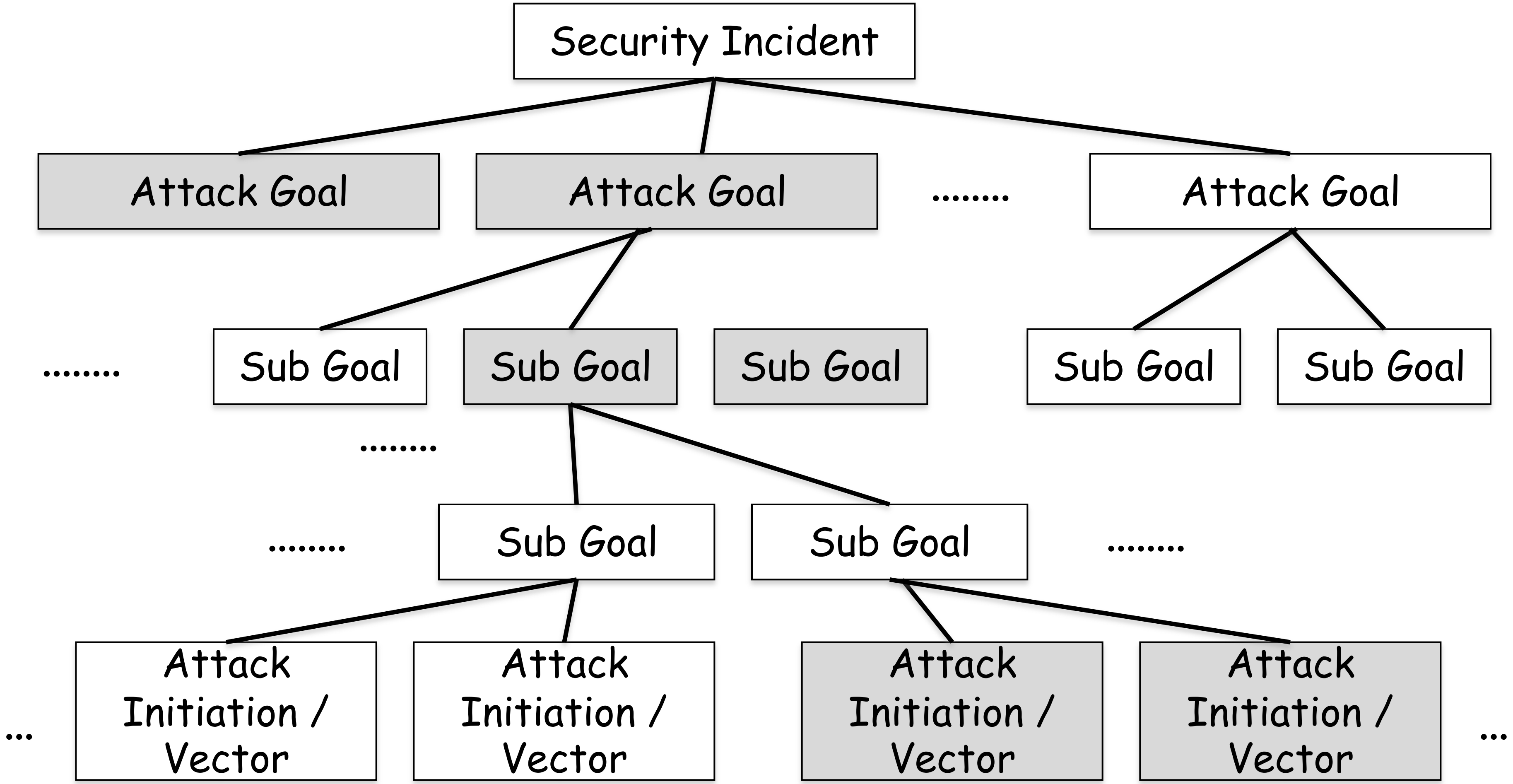
- Attack surface can be categorized:
  - Network Attack Surface
  - Computer Systems and their Sofware/Firmware/Hardware Attack Surfaces (OS, Applications, MW Services, Drivers, Computer Devices, Hardware Components)
  - Human Attack Surfaces (Social/Human Behaviours, Human Errors/Mistakes, Incorrect Actions from Trusted Insiders)

# Categories of Attack Surfaces

**Attack Surfaces can be categorized (and subcategorized) , ex:**

- **Network Attack Surface**
  - LANs, WLANs, Internet
  - Exploitable Vulnerabilities in Network Protocols
- **Software Attack Surface**
  - Vulnerabilities in SW, Applications (ex., Web Applications and Services), Utilities
  - Operating System Code
- **Human Attack Surface**
  - Vulnerabilities created by users (errors/mistakes or malicious actions)
  - Also related to Social Engineering Attacks, Bad-Operation, Abuse of Authority or incorrect/malicious actions from trusted insiders or outsourcing operation

# Representation of Attack Trees



**See in [CS] the example for a possible Attack Tree (Internet Banking Authentication Service)**

# Components of the security strategy

- Security specifications and policy or policy enforcements
- Implementation
  - Key-Fundamental security design principles
  - Implementation of services from mechanisms
    - Complementary courses of approach
      - Prevention
      - Detection
      - Response
      - Recovery
        - » Reactive Recovery
        - » Pro-active Recovery
        - » Fault/Intrusion Tolerance Guarantees
- Assurance and Evaluation
    - Foundations, Confidence, Auditing Criteria, Testing
    - Possible use of formal proofs, analytics or mathematical proofs

# Perimeter vs. Deep (or in Depth) Defenses

- Perimeter Defenses
  - Define protected and delimited perimiters (networks, subnets, groups of systems, or specific systems) using detection and prevention means to avoid undesirable (or anomalous) interactions between permimeters
    - Ex., Network segments protected by Firewalls or Intrusion Detection Systems and monitored by SIEM Platforms in a Data Center
    - Form of Isolation of Security Domains

- Deep Security
  - More complex and possibly more effective, composed by defenses at all the levels of possible attack surfaces in all Systems' Resources
    - Ex., OS Security, Security Devices, Firmware/Hardware Security Components, SW Security, Application-Level Security
    - Different forms of Deep Isolation and Containment or Resiliency Solutions (ex., Replication)

# Adversary Model Definition

"A defender must think as her/his adversary or opponent

# Adversary Model Definition and Relevance …

*The art of war teaches us to rely not on the likelihood of the enemy's not coming, but on our own readiness to receive him; not on the chance of his not attacking, but rather on the fact that we have made our position unassailable.*

**—*The Art of War*, Sun Tzu**

# Adversary Model is Critical Issue

*A system without an adversary definition cannot possibly be insecure; it can only be astonishing…*

*… astonishment is a much underrated security vice.*
(Principle of Least Astonishment)

**Virgil Gligor, MIT, On the Evolution of Adversary Models**

1. New Technologies often require a New Adversary Model Definition. What if you use old/mismatched ones ?

2. Continuous Vulnerability State: use old Adversary Models for New Technologies

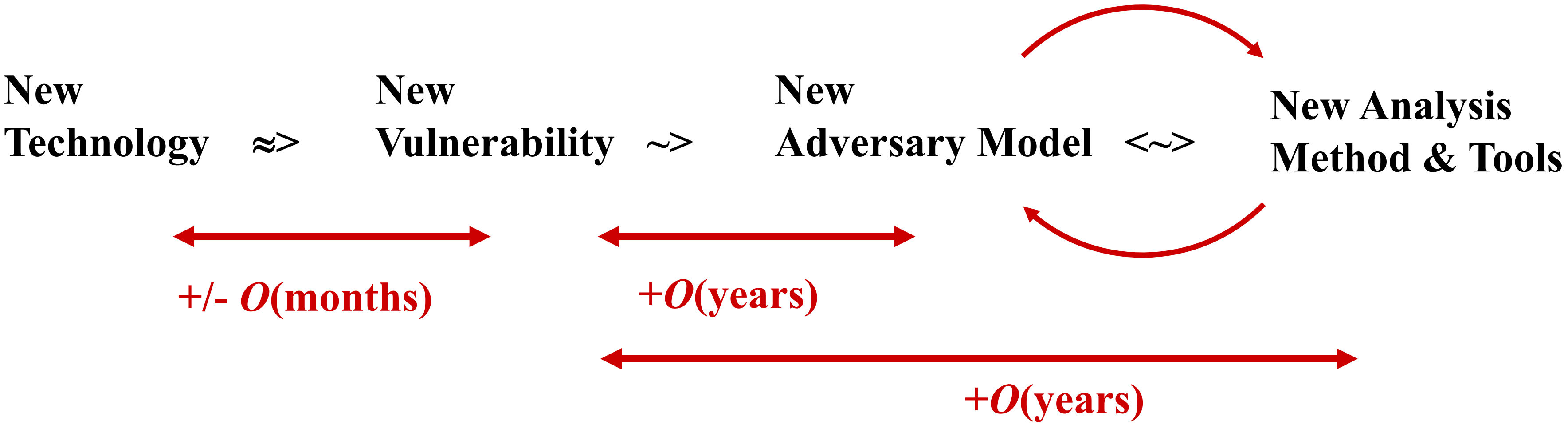
3. **Challenge: Define (New Adversary Models and Security Protocols to Handle New Threats in a Timely Manner**
   Redefine the Adv. Model => New Security Design …
   Is it possible ? Realistic ?

# Why an Adv. Def. is a fundamental concern ?

| 1. New Technology | ≈> Vulnerability ~> | Adversary | <~> Methods & Tools |
|---|---|---|---|
| -sharing user-mode programs& data;<br>- computing utility<br>(early – mid 1960s) | confidentiality and integrity breaches;<br>system penetration; | untrusted user-mode programs & subsystems | sys. vs. user mode (’62->)<br>rings, sec. kernel (’65, ‘72)<br>FHM (’75) theory/tool (’91)*<br>access. policy models (’71) |
| - shared *stateful* Services, e.g, DBMS,<br>net. protocols<br>dyn. resource alloc.<br>(early - mid 1970s) | DoS instances | untrusted user processes;<br>concurrent, coord. attacks | DoS = a diff. prob.(83-’85)*<br>formal spec. & verif. (’88)*<br>DoS models (’92 -> ) |
| - PCs, LANs;<br>public-domain Crypto<br>(mid 1970s) | read, modify, block, replay, forge messages | “man in the middle”<br>active, adaptive network adversary | informal: NS, DS (’78–81)<br>semi-formal: DY (‘83)<br>Byzantine (‘82 –>)<br>crypto attk models (‘84->)<br>auth. prot. analysis (87->) |
| - internetworking<br>(mid – late 1980s) | large-scale effects:<br>worms, viruses, DDoS (e.g., flooding) | geo. distributed, coordinated attacks | virus scans, tracebacks<br>intrusion detection<br>(mid ’90s ->) |

**2. Technology Cost -> 0, Security Concerns persist**

# The "Continuous State of Vulnerability"



New Technology ≈> New Vulnerability ~> New Adversary Model <~> New Analysis Method & Tools

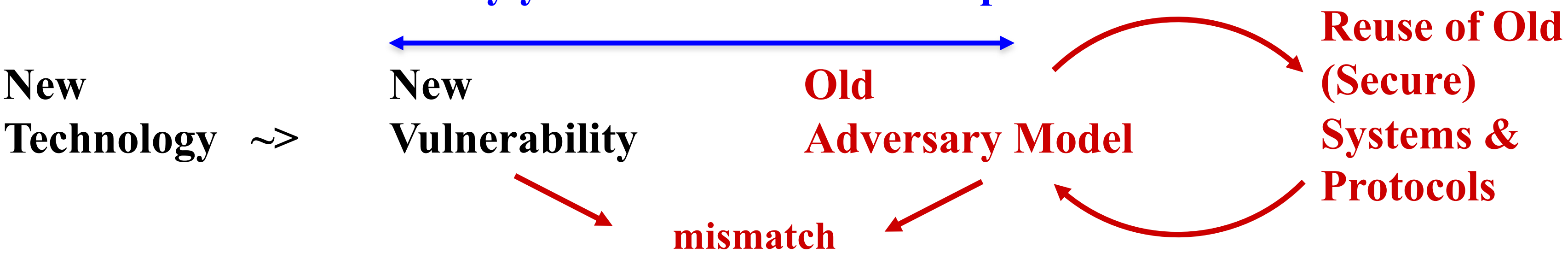
+/- *O*(months)

+*O*(years)

+*O*(years)

**… a perennial challenge ("fighting old wars")**

**This is why you must also audit and patch ☹ !**



New Technology ~> New Vulnerability

Old Adversary Model

Reuse of Old (Secure) Systems & Protocols

mismatch

# Approaching the adversary model

- You must “know” about your possible attacker ! And you must learn to know the same she/he knows !!!
- Be paranoid !
  - You must recognize her/his potential advantages !
    - What advantages ?
  - You must know her/his tools, methods, …
  - You must antecipate and characterize her/his attack-typology
  - You must anticipate her/his potential targets
  - You must know and avoid your potential vulnerabilities (before her/him)
  - Remember that the user is a possible “adversary”
    - Your must know implications of incorrect use
  - …
  - Evaluation of computer systems security as “adversaries”
    - Know / Discover vulnerabilities as the adversary does

# TCB - Trust Computing Base

# Trust Computing Base

- The trusted base and foundations beyond the security mechanisms and services
  - Foundations and Provable Properties
  - Trusted "Essential" Components We Can rely On
- For Security (Security Mechanisms and Services) we always depend on a TCB !
  - Why ?
- The better is that it must be Dependable, Delimited, Identifiable, Auditable, Verifiable, Minimal, Simple ...
  - Is it easy to address such criteria ?
  - "What means" minimal (abstraction level) ?
  - What is the "appropriate stack level" for the TCB Assumptions ?
  - Think on the current Large-Scale Distributed Systems

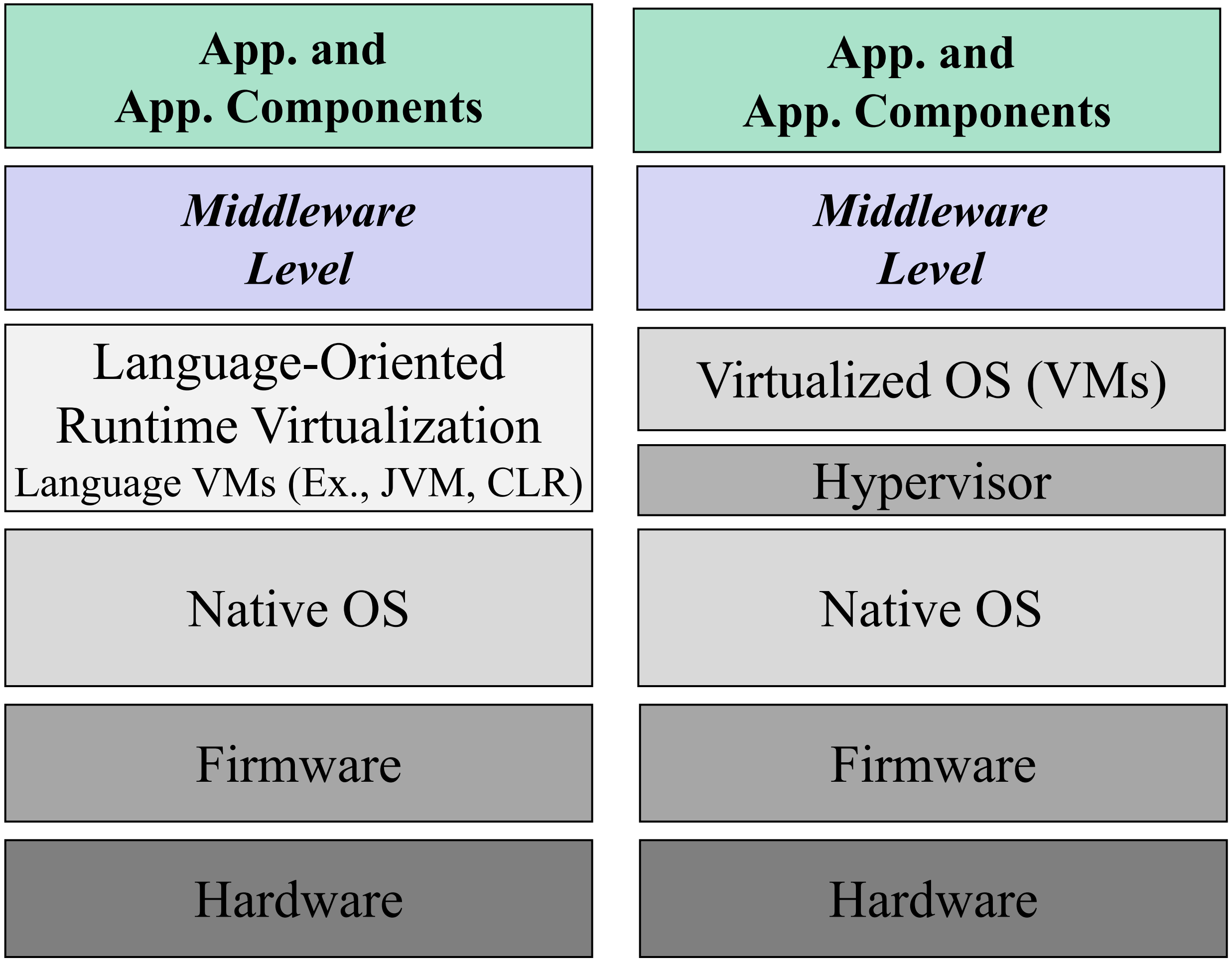
# Software/Hardware Stacks: Examples (1)

**Application Level:**
**- App. Containerization**
**- Middleware (Runtime) Libraries**

**Virtualization Level**

**Native OS Level**

**Firmware / Hardware Levels**



**App. and App. Components**

***Middleware Level***

Language-Oriented Runtime Virtualization
Language VMs (Ex., JVM, CLR)

Native OS

Firmware

Hardware

**App. and App. Components**

***Middleware Level***

Virtualized OS (VMs)

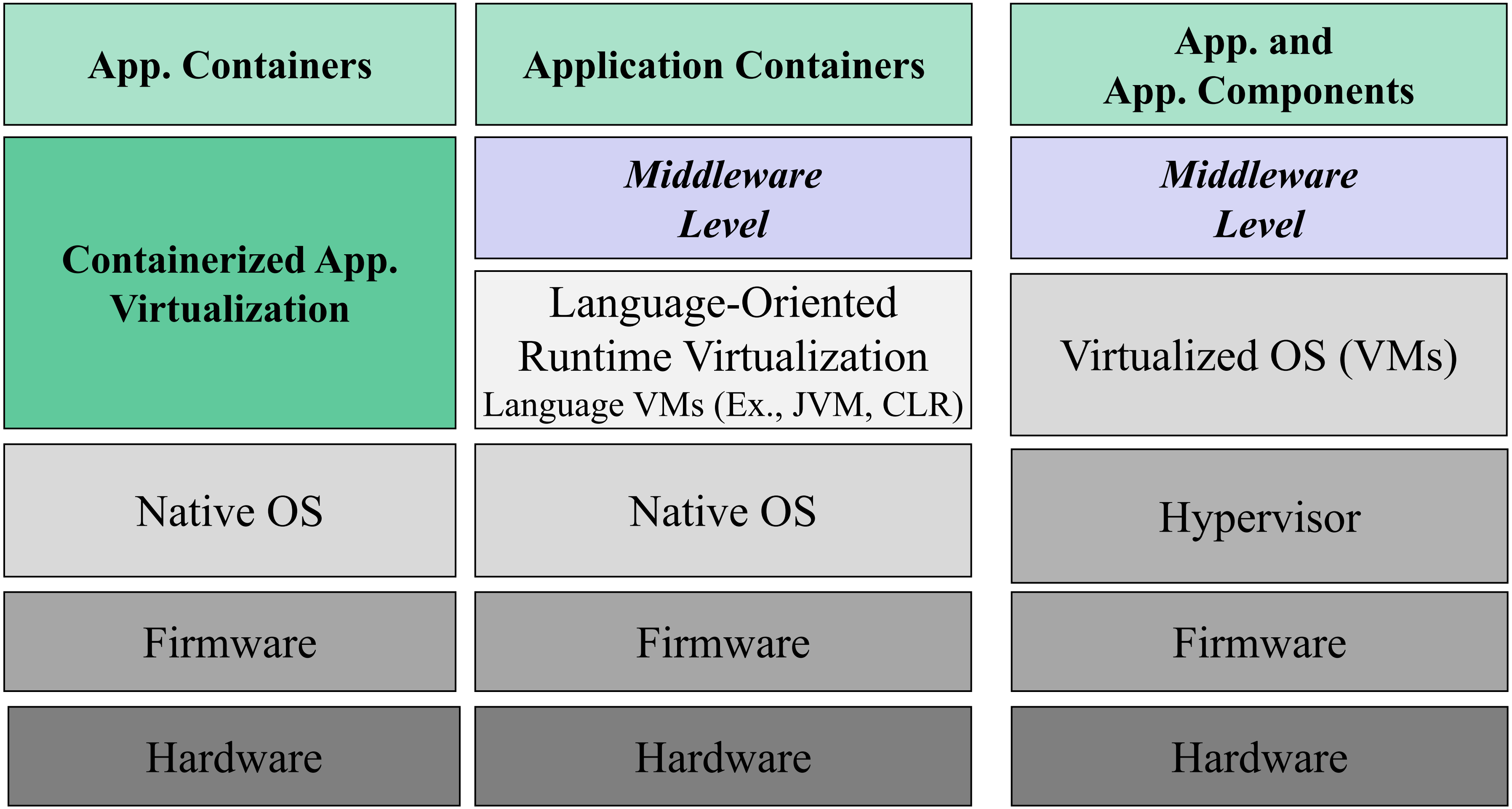
Hypervisor

Native OS

Firmware

Hardware

# Software/Hardware Stacks: Examples (2)



**App. Containers**

**Containerized App. Virtualization**

Native OS

Firmware

Hardware

**Application Containers**

***Middleware Level***

Language-Oriented Runtime Virtualization
Language VMs (Ex., JVM, CLR)

Native OS

Firmware

Hardware

**App. and App. Components**

***Middleware Level***

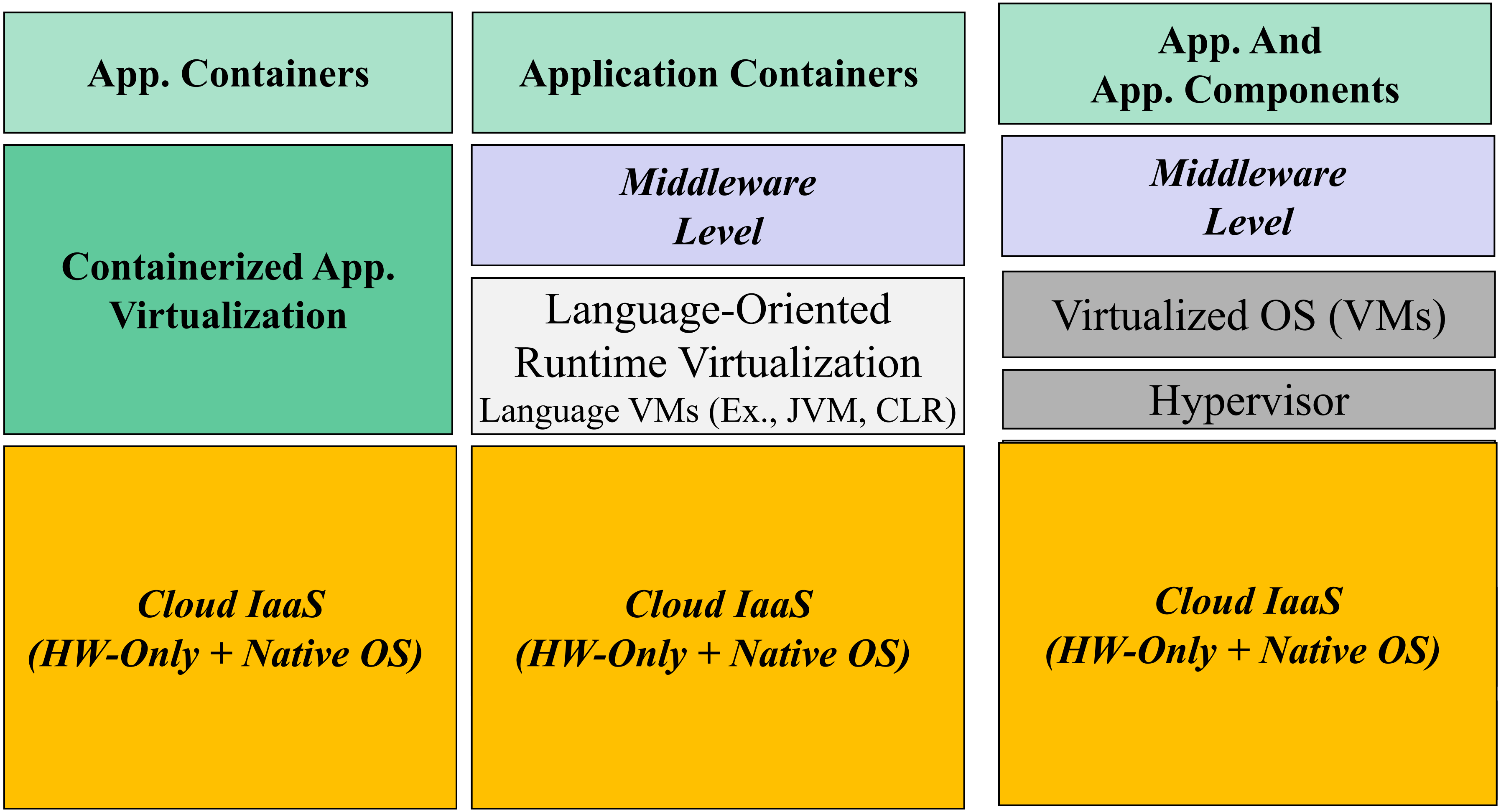
Virtualized OS (VMs)

Hypervisor

Firmware

Hardware

# Cloud-Based Stacks: Examples (1)



App. Containers

Containerized App. Virtualization

*Cloud IaaS (HW-Only + Native OS)*

Application Containers

*Middleware Level*

Language-Oriented Runtime Virtualization
Language VMs (Ex., JVM, CLR)

*Cloud IaaS (HW-Only + Native OS)*

App. And App. Components

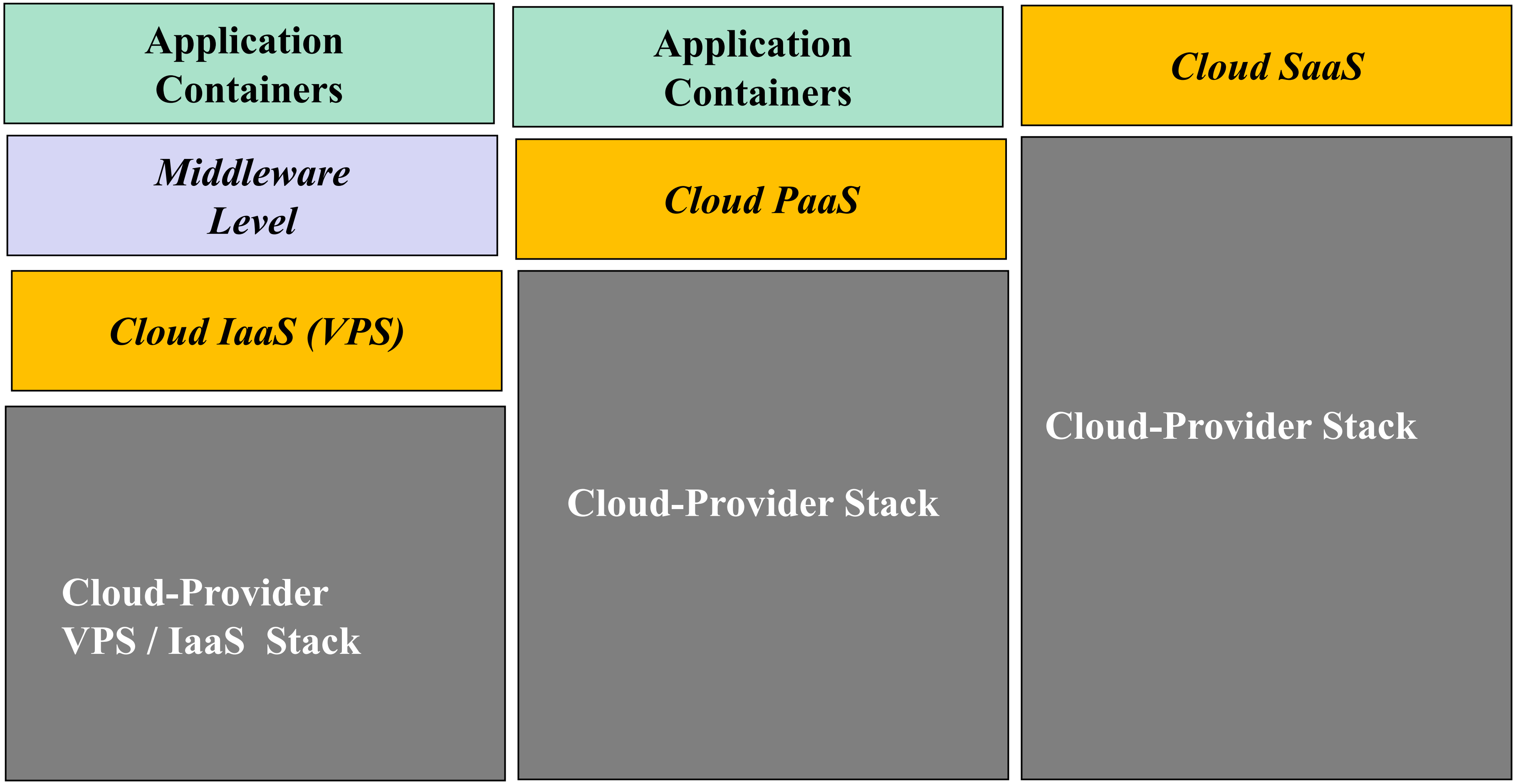
*Middleware Level*

Virtualized OS (VMs)

Hypervisor

*Cloud IaaS (HW-Only + Native OS)*

# Cloud-Based Stacks: Examples (2)



**Application Containers**

***Middleware Level***

***Cloud IaaS (VPS)***

**Cloud-Provider VPS / IaaS Stack**

**Application Containers**

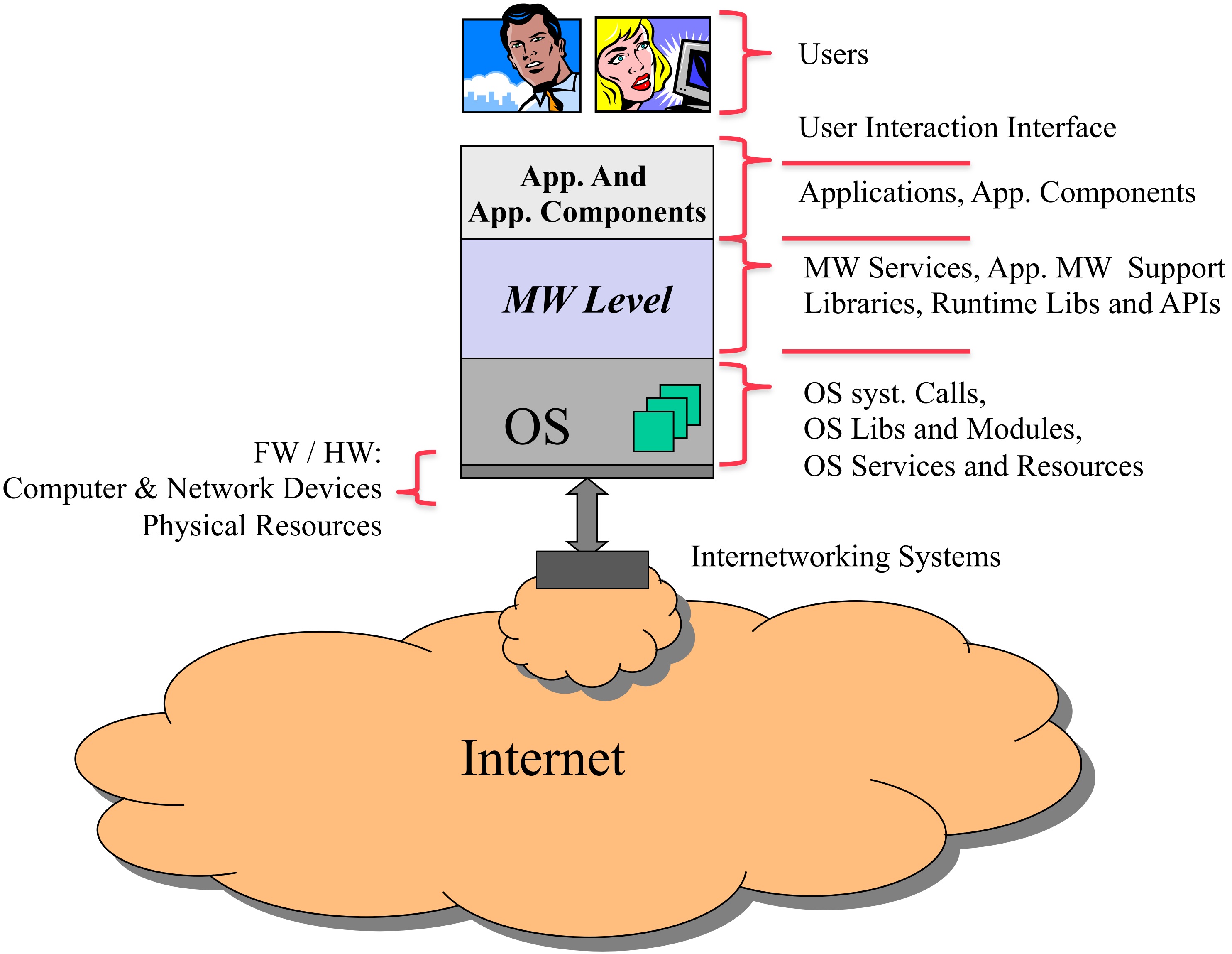
***Cloud PaaS***

**Cloud-Provider Stack**

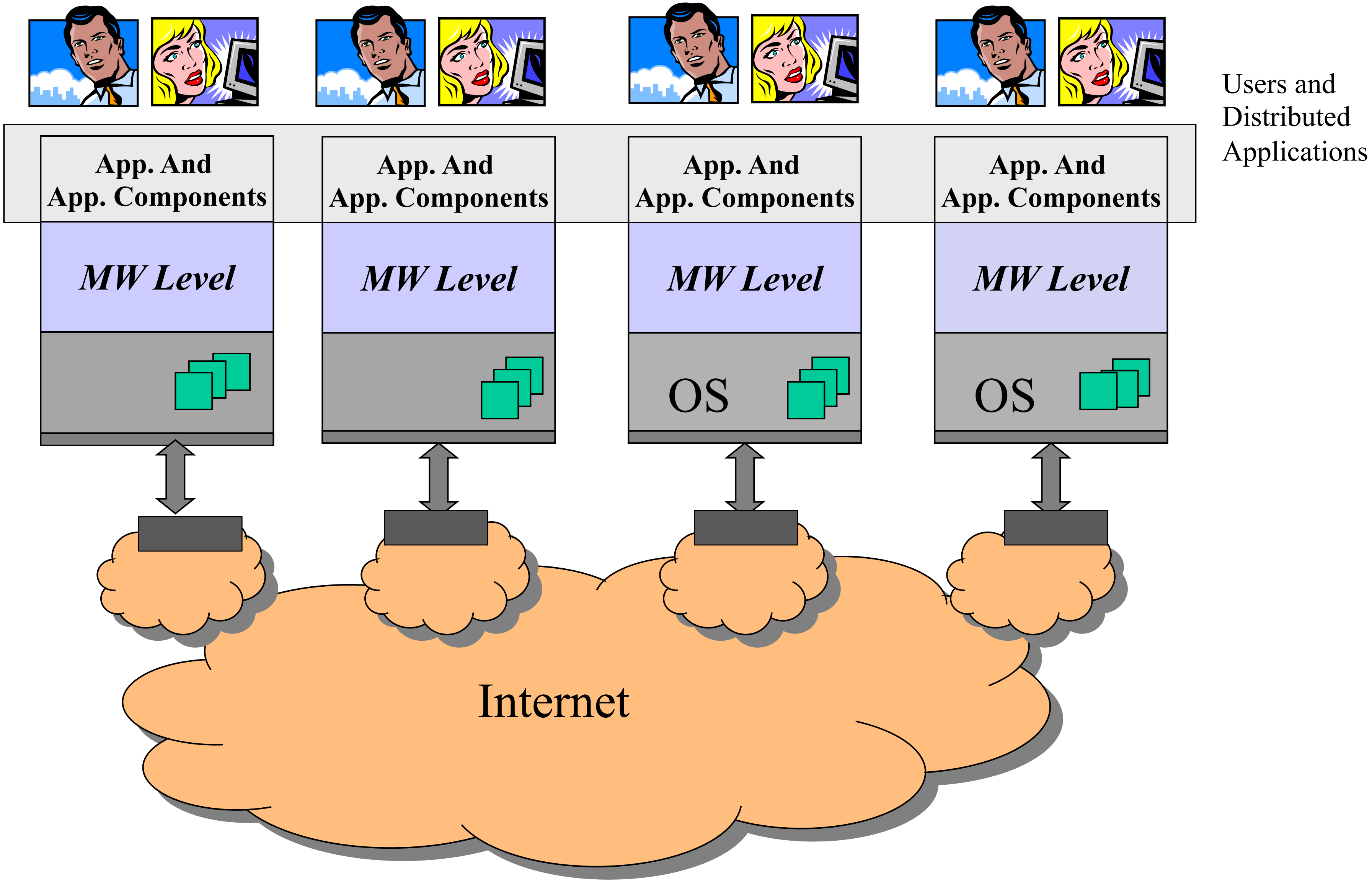
***Cloud SaaS***

**Cloud-Provider Stack**

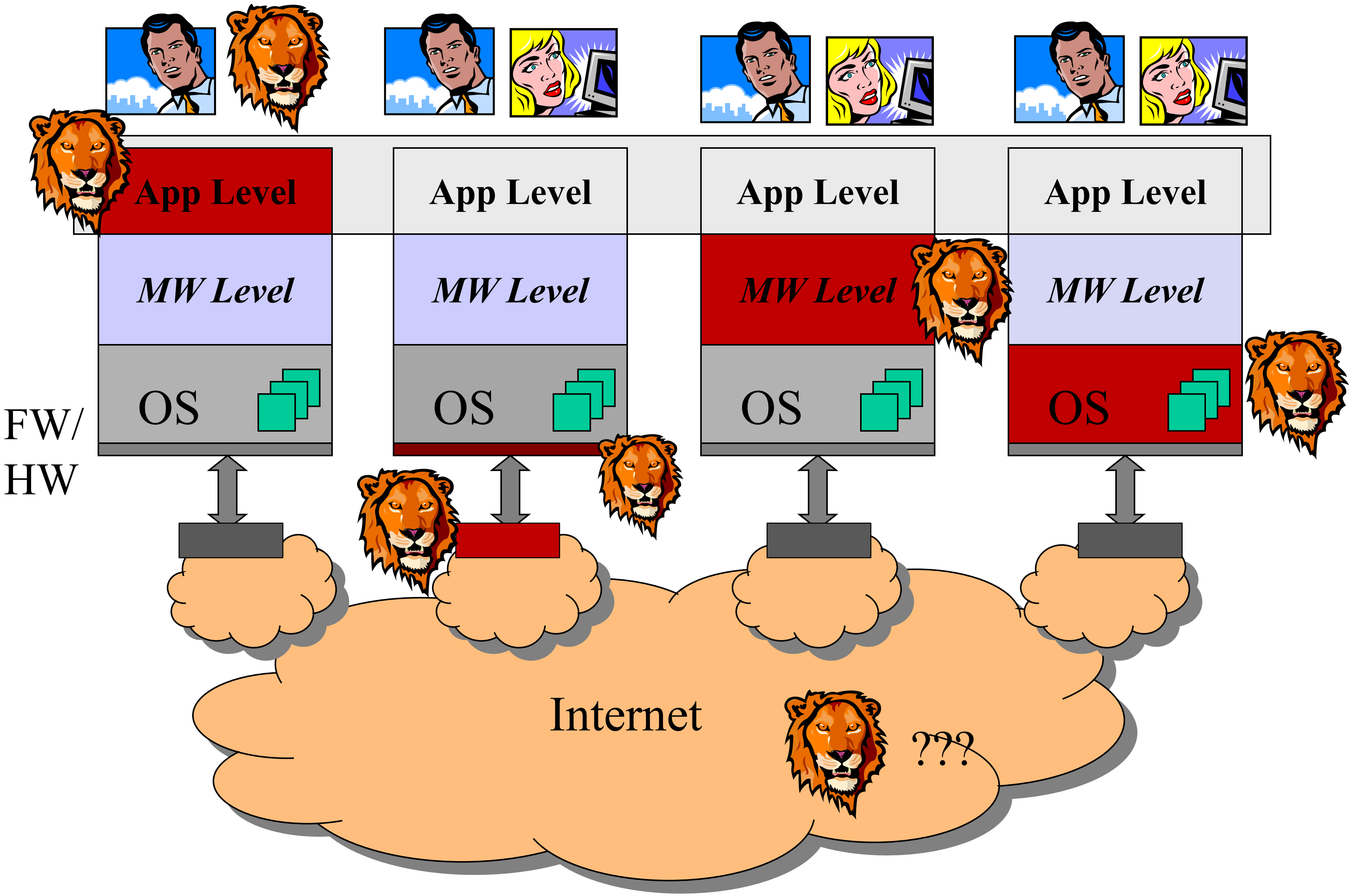
# Identification and delimitation of TCB


Users
User Interaction Interface
App. And
App. Components
Applications, App. Components
MW Level
MW Services, App. MW Support
Libraries, Runtime Libs and APIs
OS
OS syst. Calls,
OS Libs and Modules,
OS Services and Resources
FW / HW:
Computer & Network Devices
Physical Resources
Internetworking Systems
Internet

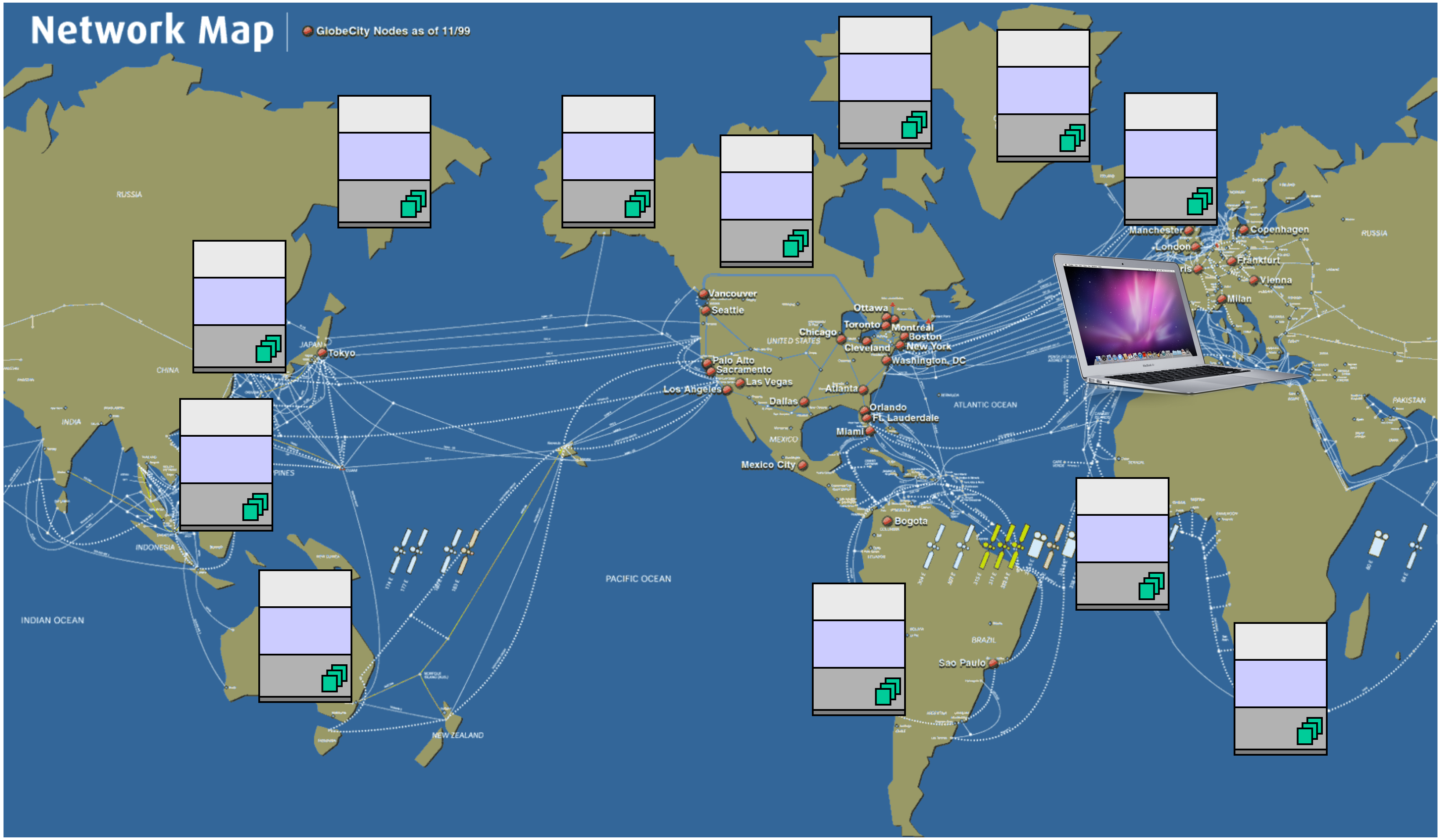
# Identification and delimitation of TCB



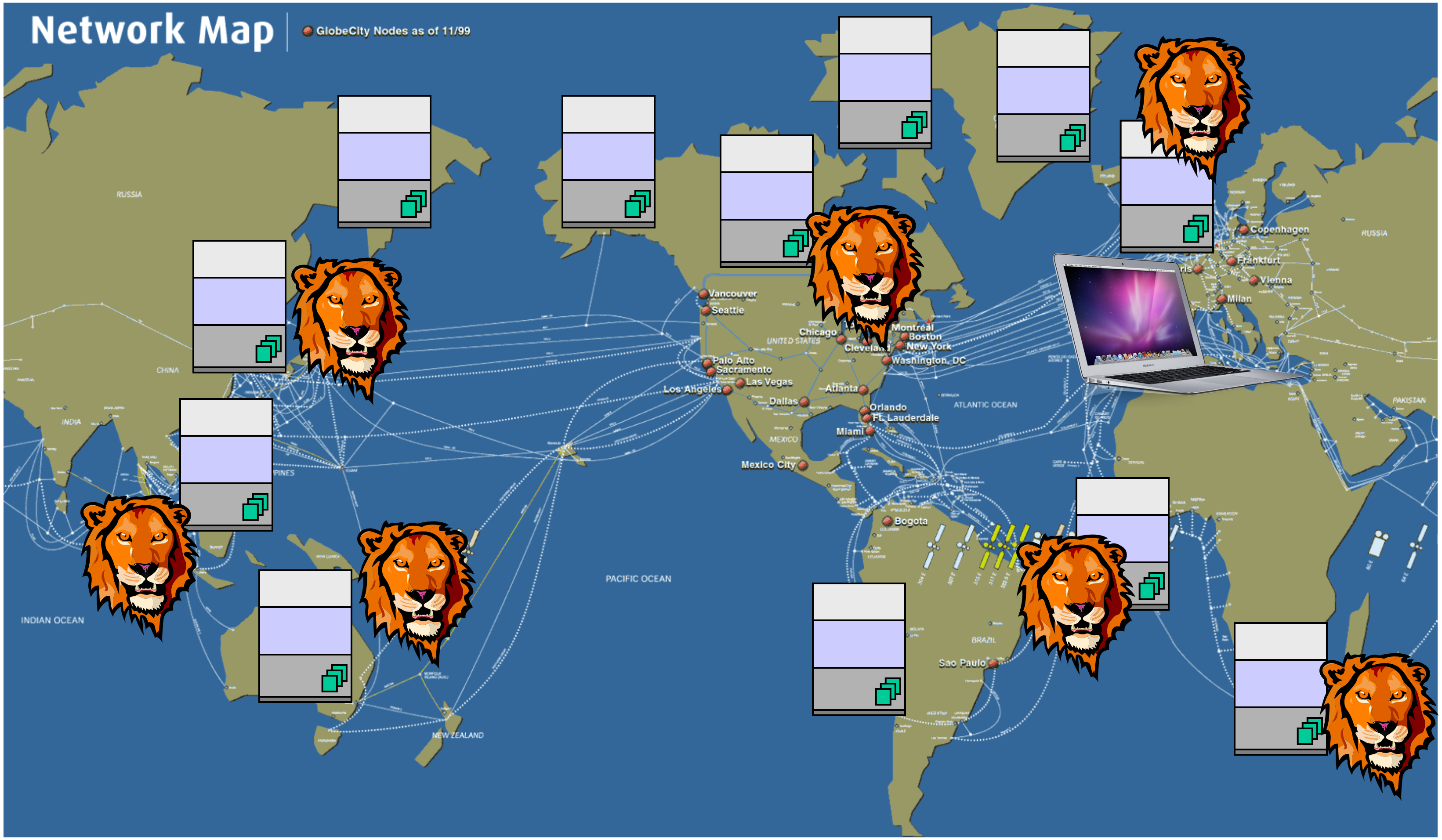
# Approach level and reduction of TCBs



# Distribution of TCBs

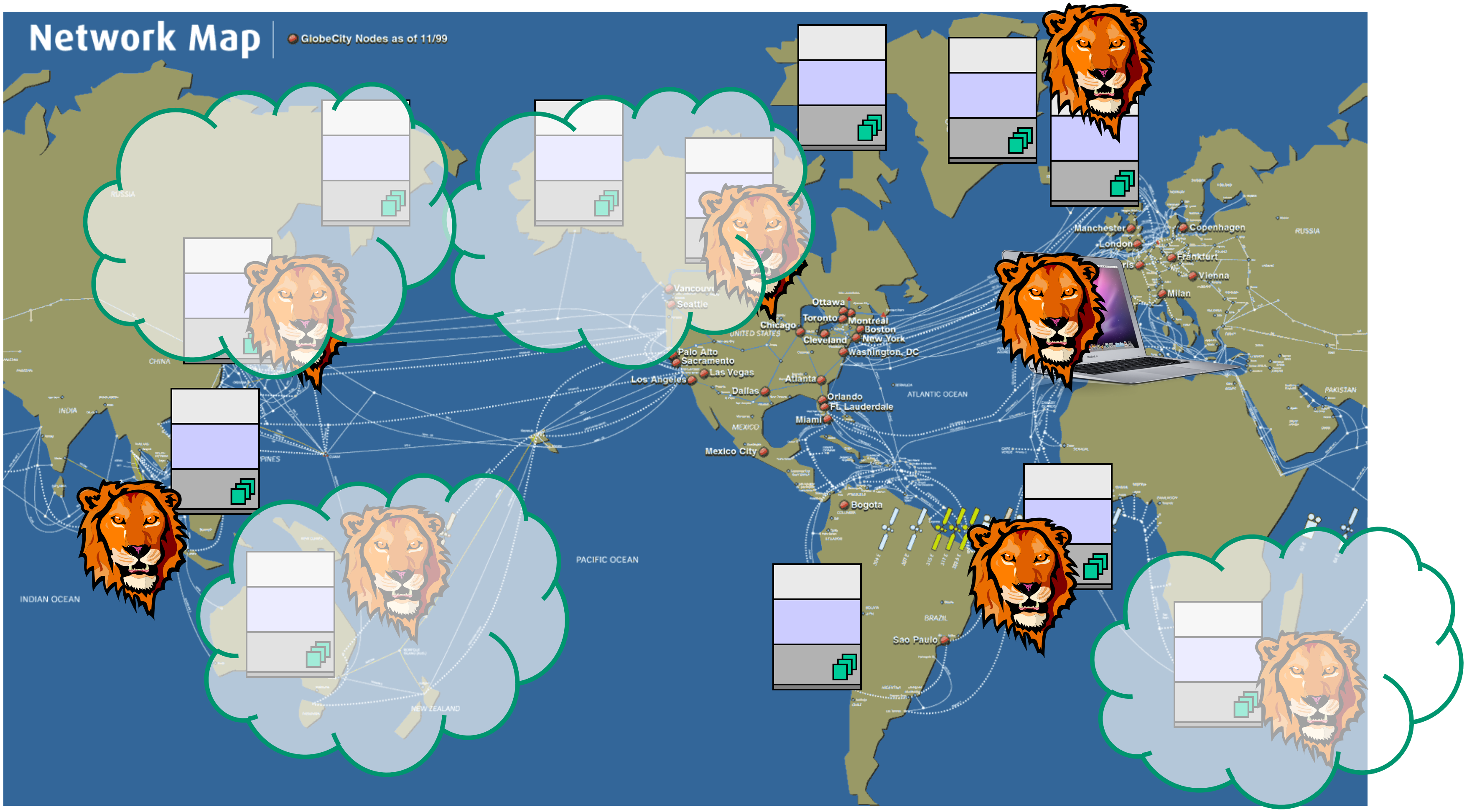


# How to address a TCB in a Large Scale Distributed System ?



Large-Scale Distributed Systems (Edge-Components, Decentralized Services/Components, Storage/Computing Clouds)

# How to address a TCB in a Large Scale Distributed System ?



Large-Scale Distributed Systems (Edge-Components, Decentralized Services/Components, Storage/Computing Clouds)

# Suggested Readings

- Review the slides …
- W. Stallings, L. Brown, Computer Security – Principles and Practice, Person, Ch.1 ( § 1.1- § 1.3)
- W. Stallings, Network Security Essentials – Applications and Standards, Ch.1 ( § 1.1- § 1.6)

See the Review Questions and Try to Answer