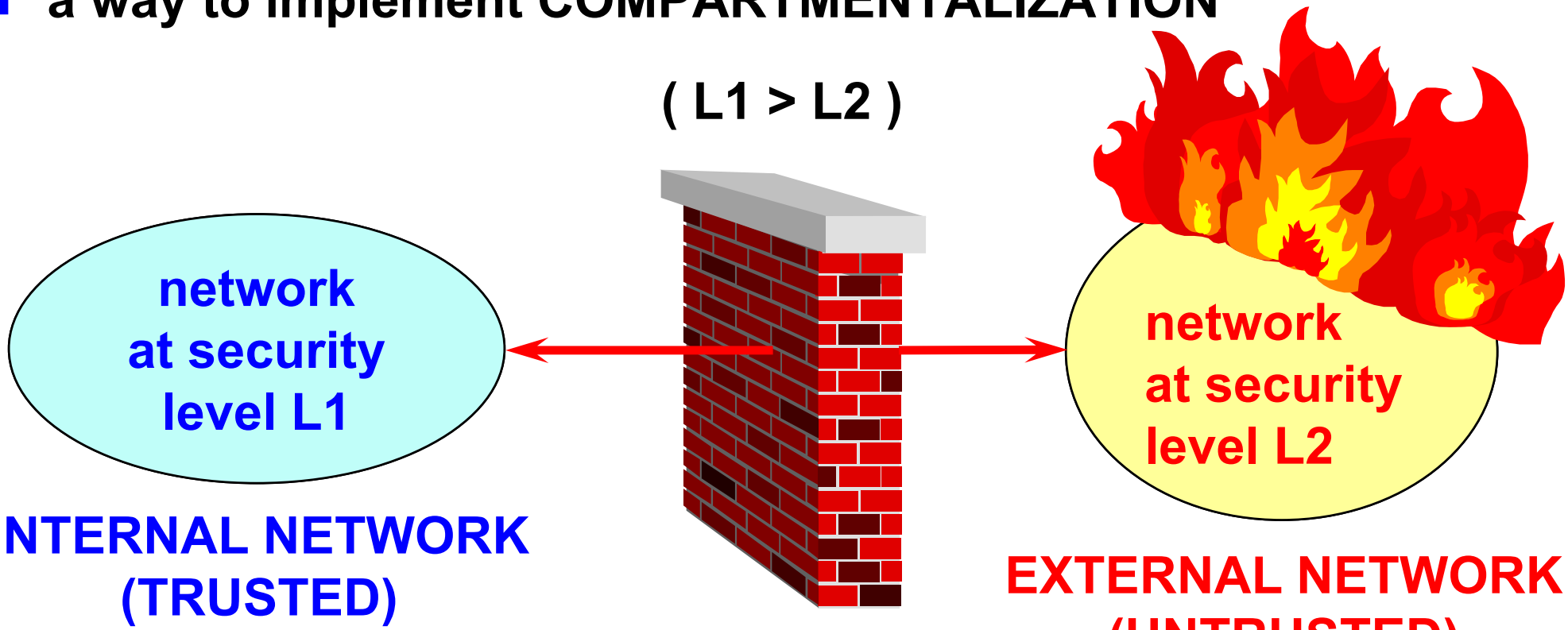
Firewall and IDS/IPS

Antonio Lioy
< lioy @ polito.it >

Politecnico di Torino
Dip. Automatica e Informatica

What is a firewall?

- firewall = wall to protect against fire propagation
- controlled connection between networks at different security levels = boundary protection (network filter)
- a way to implement COMPARTMENTALIZATION



© A.Lioy (Politecnico di Torino, 2005-2024)

Ingress vs. Egress firewall

- **ingress firewall**
  - incoming connections
  - typically to select the (public) services offered
  - sometimes as part of an application exchange initiated by my users
- **egress firewall**
  - outgoing connections
  - typically to check the activity of my personnel (!)
- **easy classification for channel-based services (e.g. TCP applications), but difficult for message-based stateless services (e.g. ICMP, UDP applications)**

© A.Lioy (Politecnico di Torino, 2005-2024)

THE THREE COMMANDMENTS OF FIREWALL

I. the FW must be the only contact point of the internal network with the external one

II. only the “authorized” traffic can traverse the FW

III. the FW must be a highly secure system itself

D.Cheswick
S.Bellovin

© A.Lioy (Politecnico di Torino, 2005-2024)

Authorization policies

"All that is not explicitly permitted, is forbidden"

- higher security (gatekeeper)
- more difficult to manage

permitlist, allowlist

"All that is not explicitly forbidden, is permitted"

- lower security (open gates)
- more easy to manage

blocklist, denylist

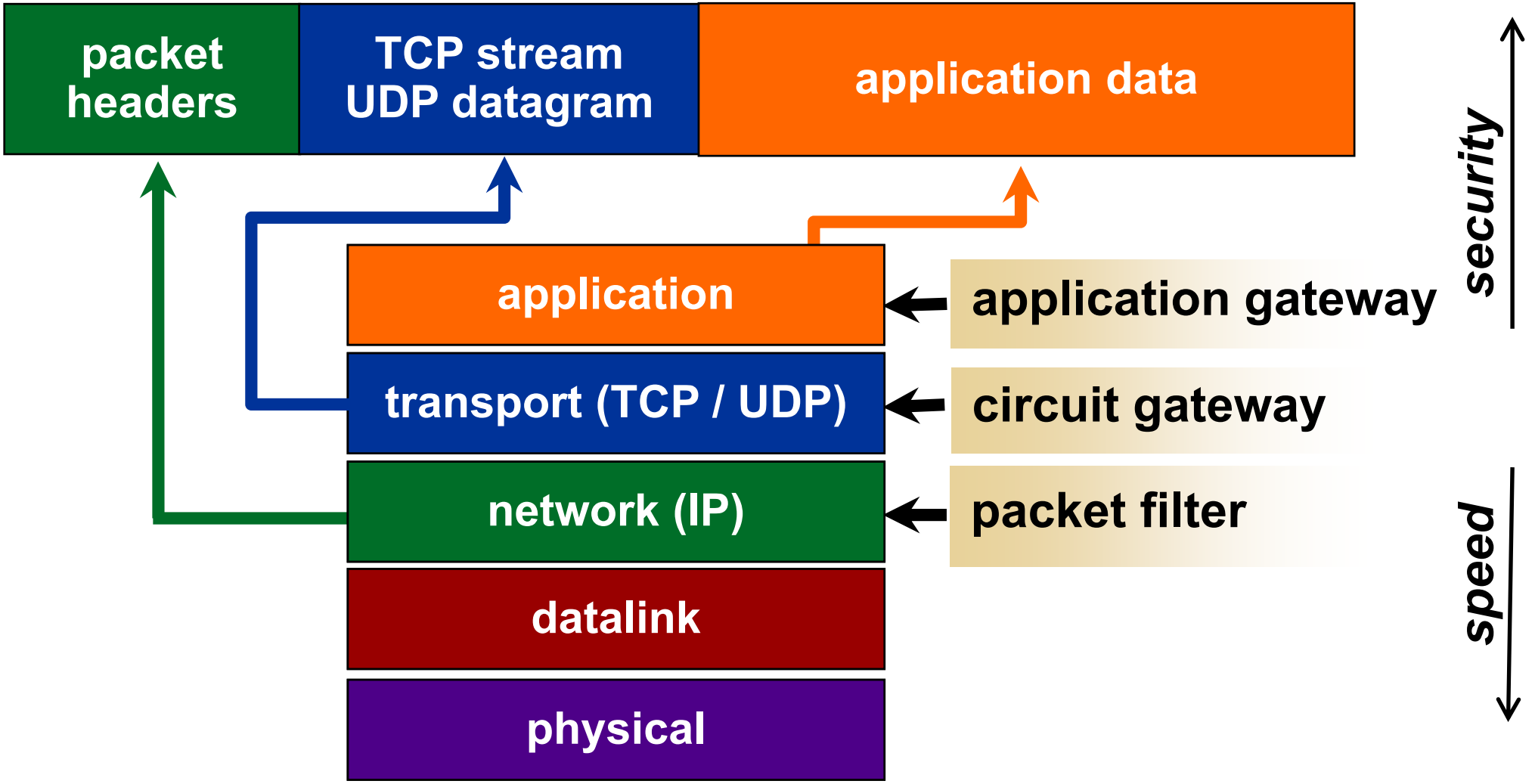
© A.Lioy (Politecnico di Torino, 2005-2024)

FW: basic components

- **packet filter / screening router / choke**
  component that filters traffic at network level
- **bastion host**
  secure system, with auditing
- **application gateway (proxy)**
  service that works on behalf of an application, with access control
- **dual-homed gateway**
  system with two network cards and routing disabled

© A.Lioy (Politecnico di Torino, 2005-2024)

A which level the controls are made?



© A.Lioy (Politecnico di Torino, 2005-2024)

Firewall technologies

- **different controls at various network levels:**
  - (static) packet filter
  - stateful (dynamic) packet filter
  - cutoff proxy
  - circuit-level gateway / proxy
  - application-level gateway / proxy
  - stateful inspection
- **differences in terms of:**
  - controls to be performed (= threats detected)
  - performance
  - protection of the firewall O.S.
  - keeping or breaking the client-server model

© A.Lioy (Politecnico di Torino, 2005-2024)

Packet filter

- **historically available on routers, nowadays in every OS**
- **packet inspection at network level**
  - IP header
  - transport header
- **rule example:**
  - permit incoming connections to our web server
    “src any dst 10.1.2.3/0.0.0.0 tcp 80 allow”
  - only our internal DNS server can query DNS external servers
    "src 10.1.2.1/0.0.0.0 dst any udp 53 allow"
  - typically, there is an (implicit) "deny all" rule at the end
  - order is important (first match principle)

© A.Lioy (Politecnico di Torino, 2005-2024)

Packet filter: pros and cons

- **independent of applications**
  - good scalability
  - approximate controls: easy to “fool”
    (e.g. IP spoofing, fragmented packets)
- **good performance**
- **low cost (available on routers and in many OS)**
- **difficult to support services with dynamically allocated ports (e.g. FTP)**
- **complex to configure**
- **difficult to perform user authentication**

© A.Lioy (Politecnico di Torino, 2005-2024)

Application-level gateway

- composed by a set of proxies inspecting the packet payload at application level
- often requires modifications to the client application
- may optionally mask / renumber the internal IP addresses
- when used as part of a firewall, usually performs also peer authentication
- top security!! (e.g. against buffer overflow of the target application)
- difference between forward-proxy (egress) and reverse-proxy (ingress)
- rule example:
  - deny dangerous HTTP methods “PUT, DELETE deny”

© A.Lioy (Politecnico di Torino, 2005-2024)

Application-level gateway (I)

- **rules are more fine-grained and simple than those of a packet filter**
- **every application needs a specific proxy**
  - delay in supporting new applications
  - heavy on resources (many processes)
  - low performance (user-mode processes)
- **SMP may improve performance**
- **completely breaks the client/server model**
  - more protection for the server
  - may authenticate the client
  - not transparent to the client
  - the proxy OS may be attacked

© A.Lioy (Politecnico di Torino, 2005-2024)

Application-level gateway (II)

- **problems with application-level security techniques that do not permit traffic inspection (e.g. TLS)**
- **variants:**
  - transparent proxy
    - less intrusive for the client
    - more work (packet rerouting + dst extraction)
  - strong application proxy (checking semantics, not just syntax)
    - only some commands/data are forwarded
    - this is the only correct configuration for a proxy

© A.Lioy (Politecnico di Torino, 2005-2024)

Circuit-level gateway (I)

- **a generic proxy (i.e. not “application-aware”)**
  - creates a transport-level circuit between client and server …
  - … but it doesn’t understand or manipulate in any way the payload data
  - … it just copies between its two interfaces the TCP segments or UDP datagrams (if they match the access control rules)
  - … but, in doing this, it will re-assemble the IP packets and hence it will provide protection against some L3/L4 attacks

© A.Lioy (Politecnico di Torino, 2005-2024)

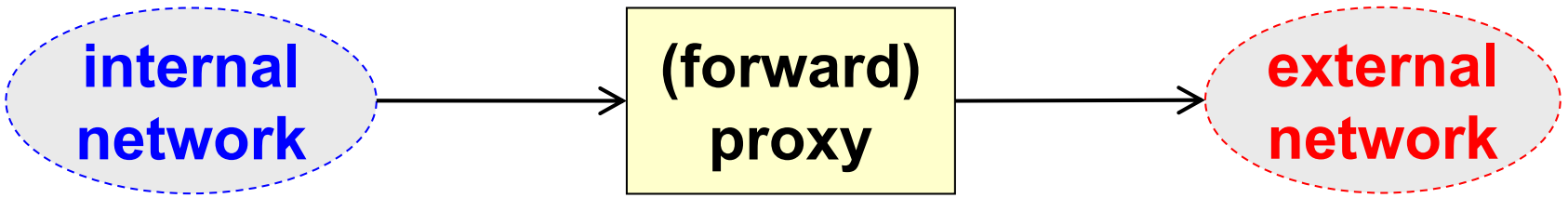
Circuit-level gateway (II)

- **breaks the TCP/UDP-level client/server model during the connection**
  - more protection for the server
    - isolated from all attacks related to the TCP handshake
    - isolated from all attacks related to the IP fragmentation
  - may authenticate the client
    - but this requires modification to the application
- **still exhibits many limitations of the packet filter**
- **SOCKS is the most famous one**

© A.Lioy (Politecnico di Torino, 2005-2024)

HTTP (forward) proxy

- **a HTTP server acting just as a front-end and then passing requests to the real server (external)**
- **it's an egress control**
- **benefits (besides network ACL):**
  - shared cache of external pages for all internal users
  - authentication + authorization of internal users
  - various controls (e.g. allowed sites, transfer direction, data types, …)



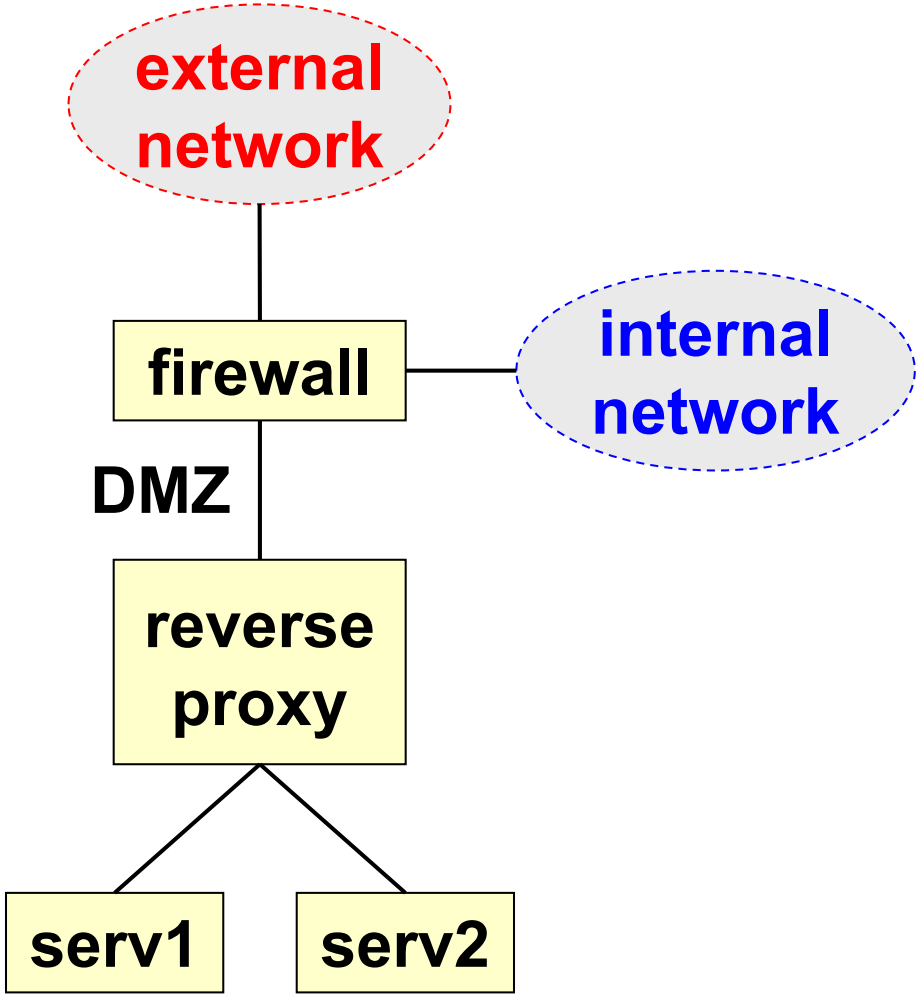
© A.Lioy (Politecnico di Torino, 2005-2024)

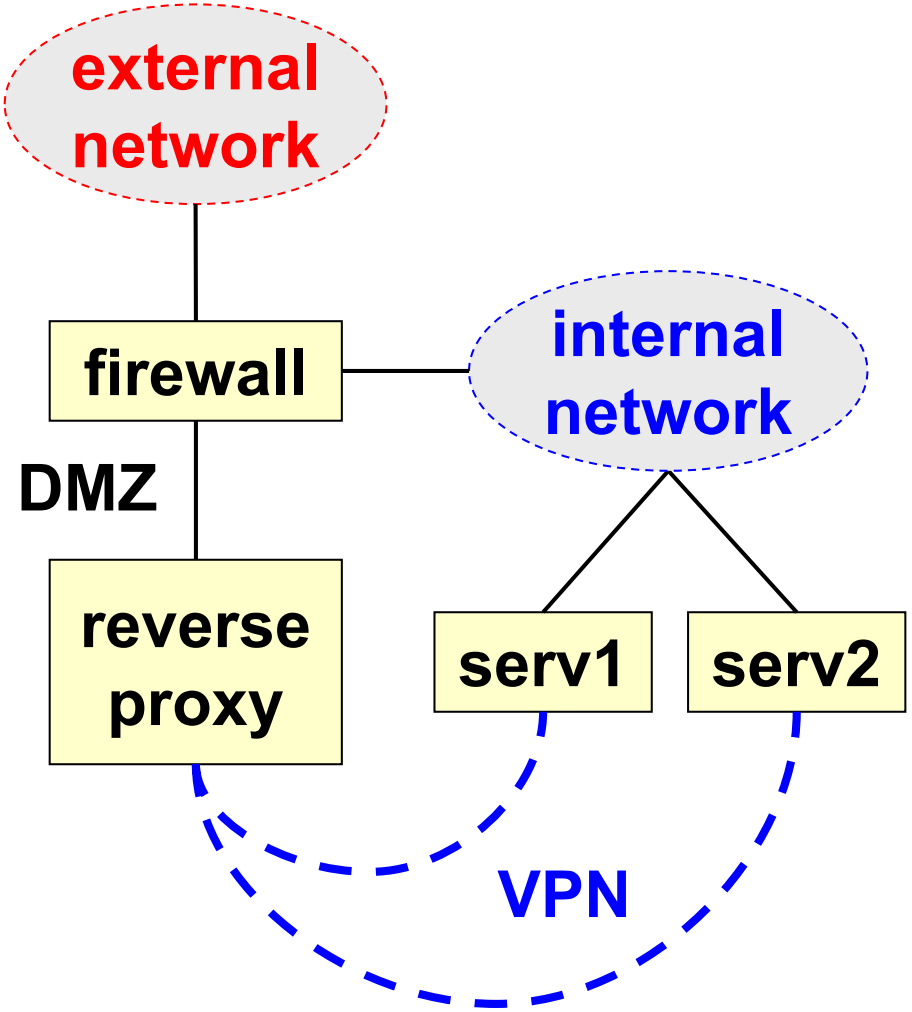
HTTP reverse proxy

- **HTTP server acting just as a front-end for the real server(s) which the requests are passed to**
- **implements network ACL & content inspection**
- **… plus additional benefits:**
  - obfuscation (no info about the real server)
  - TLS accelerator (with unprotected back-end connections …)
  - load balancer
  - web accelerator (=cache for static content)
  - compression
  - spoon feeding (gets from the server a whole dynamic page and feeds it to the client according to its speed, so unloading the application server)

© A.Lioy (Politecnico di Torino, 2005-2024)

Reverse proxy: possible configurations





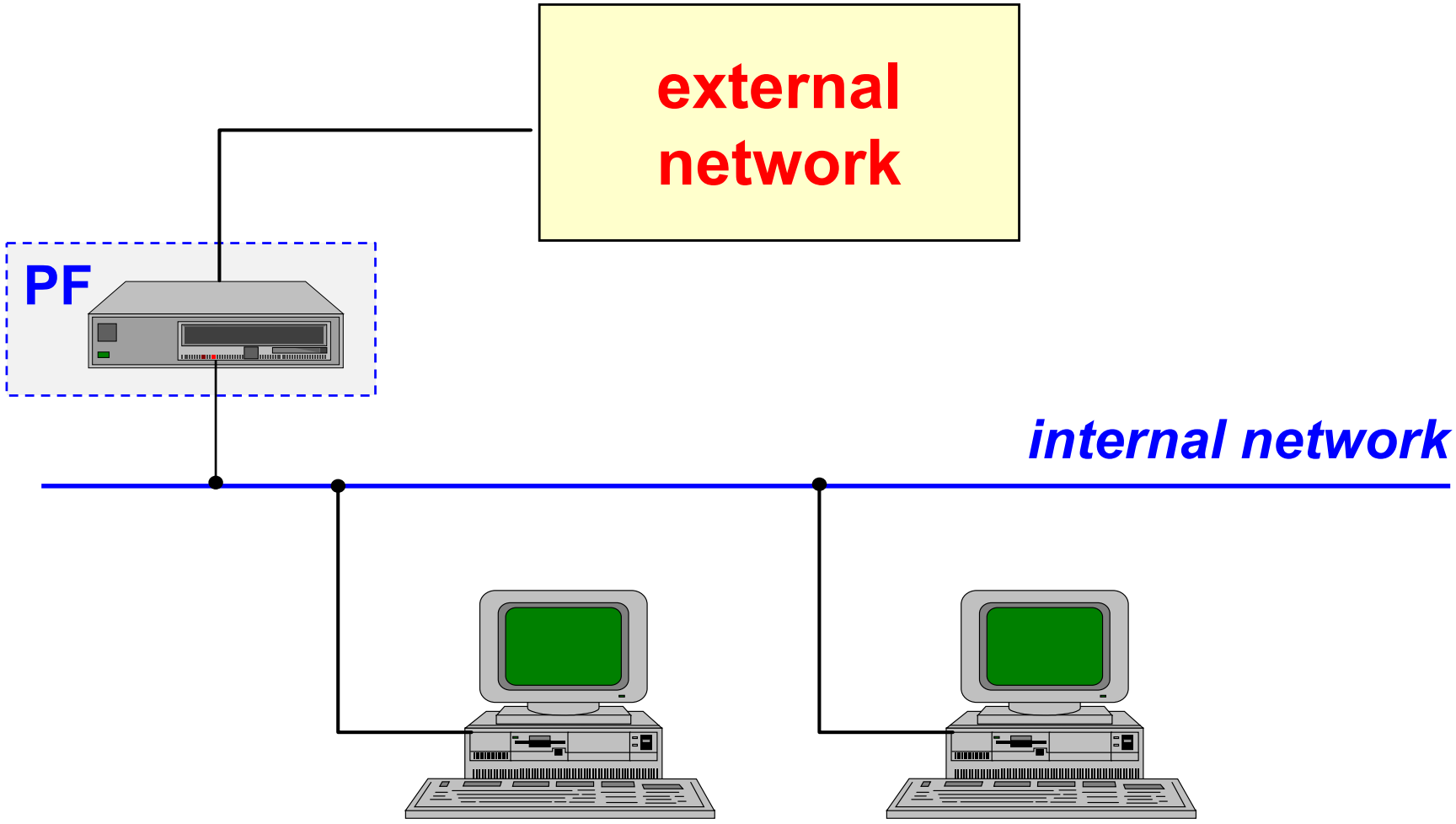
© A.Lioy (Politecnico di Torino, 2005-2024)

WAF (Web Application Firewall)

- **large use of web applications = many threats**
- **WAF = module installed at a proxy (forward and/or reverse) to filter the application traffic**
  - HTTP commands
  - header of HTTP request/response
  - content of HTTP request/response
- **ModSecurity**
  - plugin for Apache and NGINX (50% and 30% of worldwide HTTP servers)
  - OWASP ModSecurity Core Rule Set (CRS)

© A.Lioy (Politecnico di Torino, 2005-2024)

"Packet filter" architecture



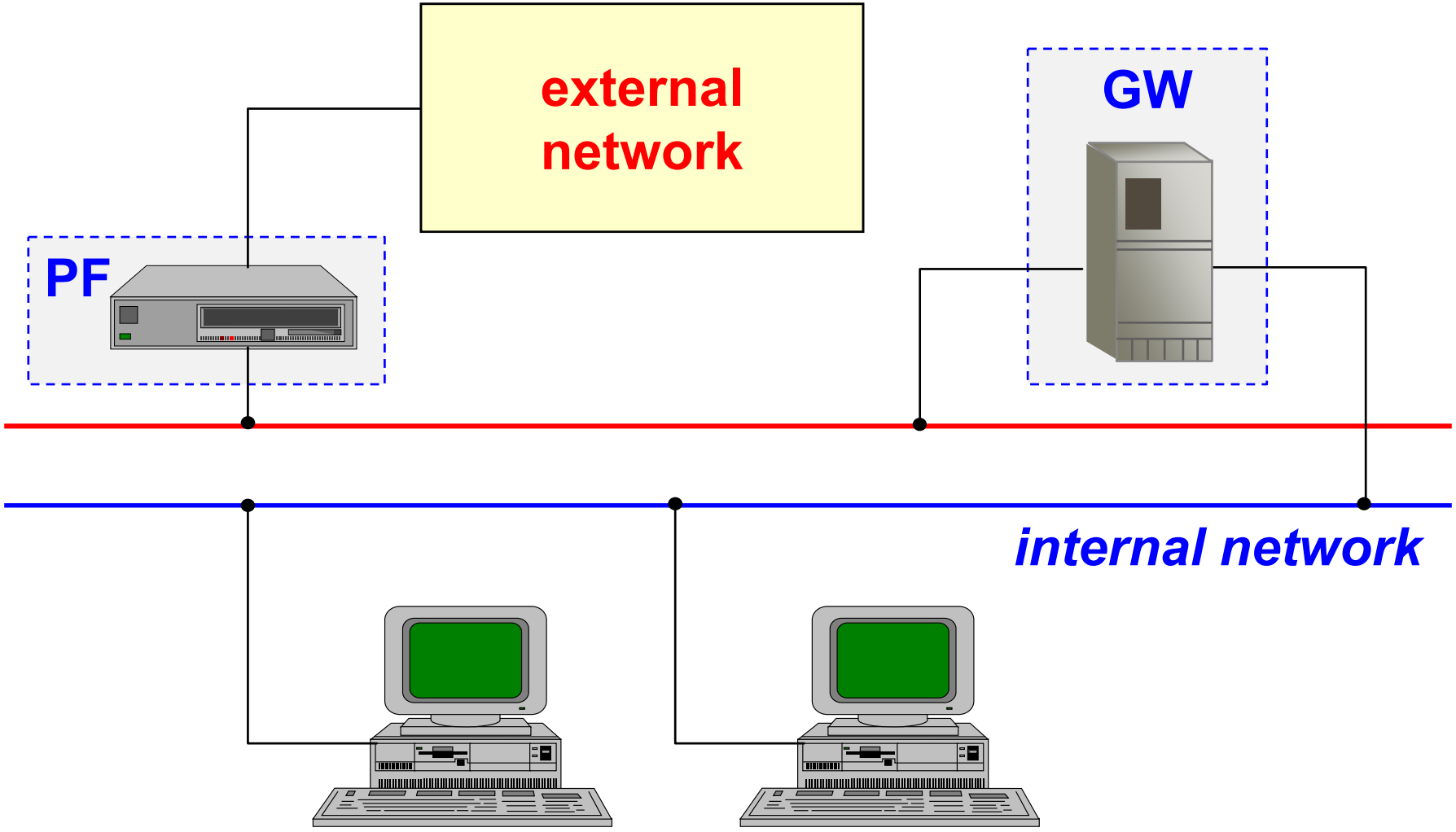
© A.Lioy (Politecnico di Torino, 2005-2024)

"Packet filter" architecture

- **exploits the packet filter to screen the traffic both at IP and upper levels**
  - if implemented with a router then it's a "screening router" and there's no need for extra dedicated hardware
- **no need for a proxy and hence no need to modify the applications**
- **simple, easy, cheap and ... insecure!**

© A.Lioy (Politecnico di Torino, 2005-2024)

"Dual-homed gateway" architecture

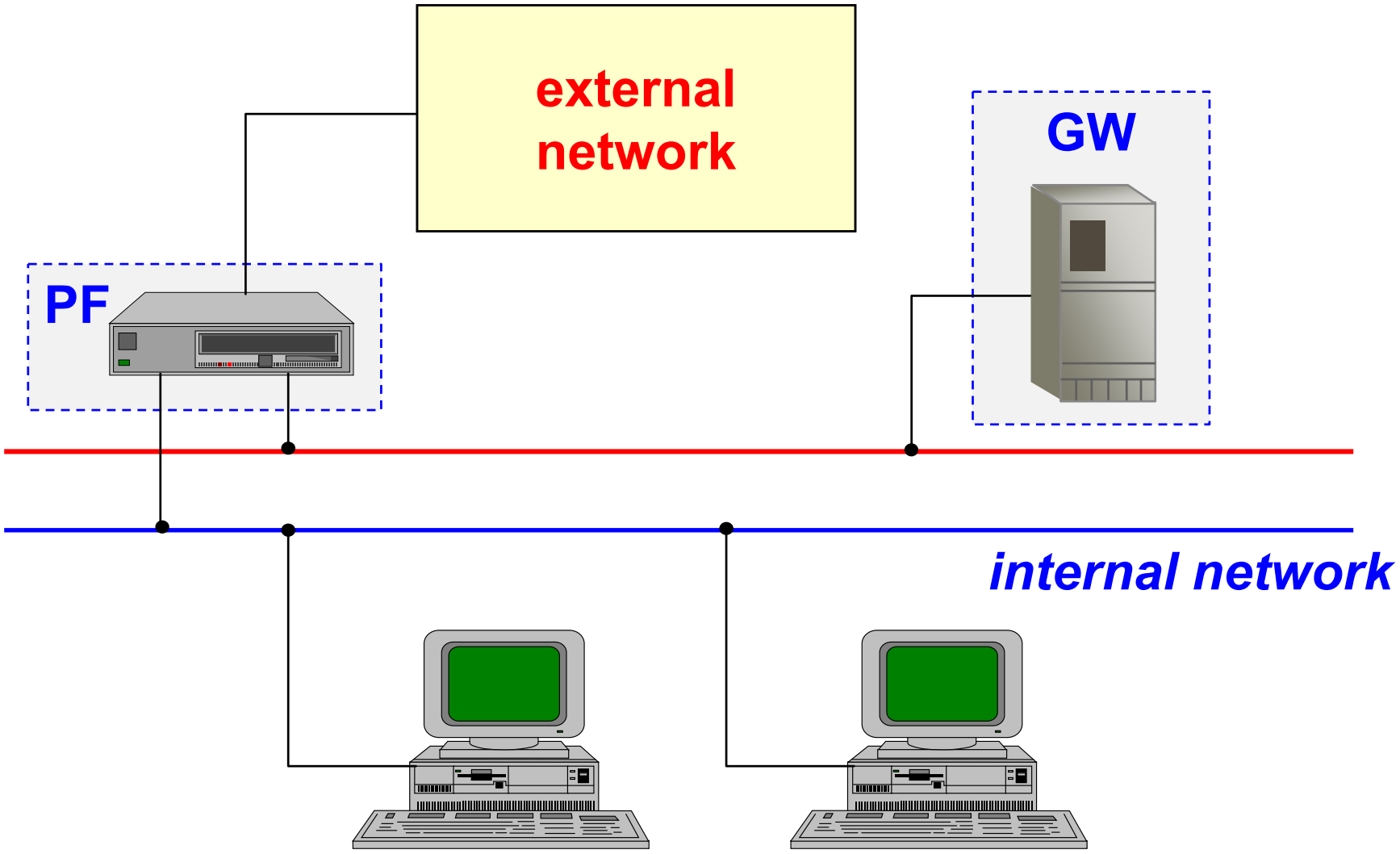


© A.Lioy (Politecnico di Torino, 2005-2024)

"Dual-homed gateway" architecture

- easy to implement
- small additional hardware requirements
- the internal network can be masqueraded
- unflexible
- large work overhead

"Screened host" architecture



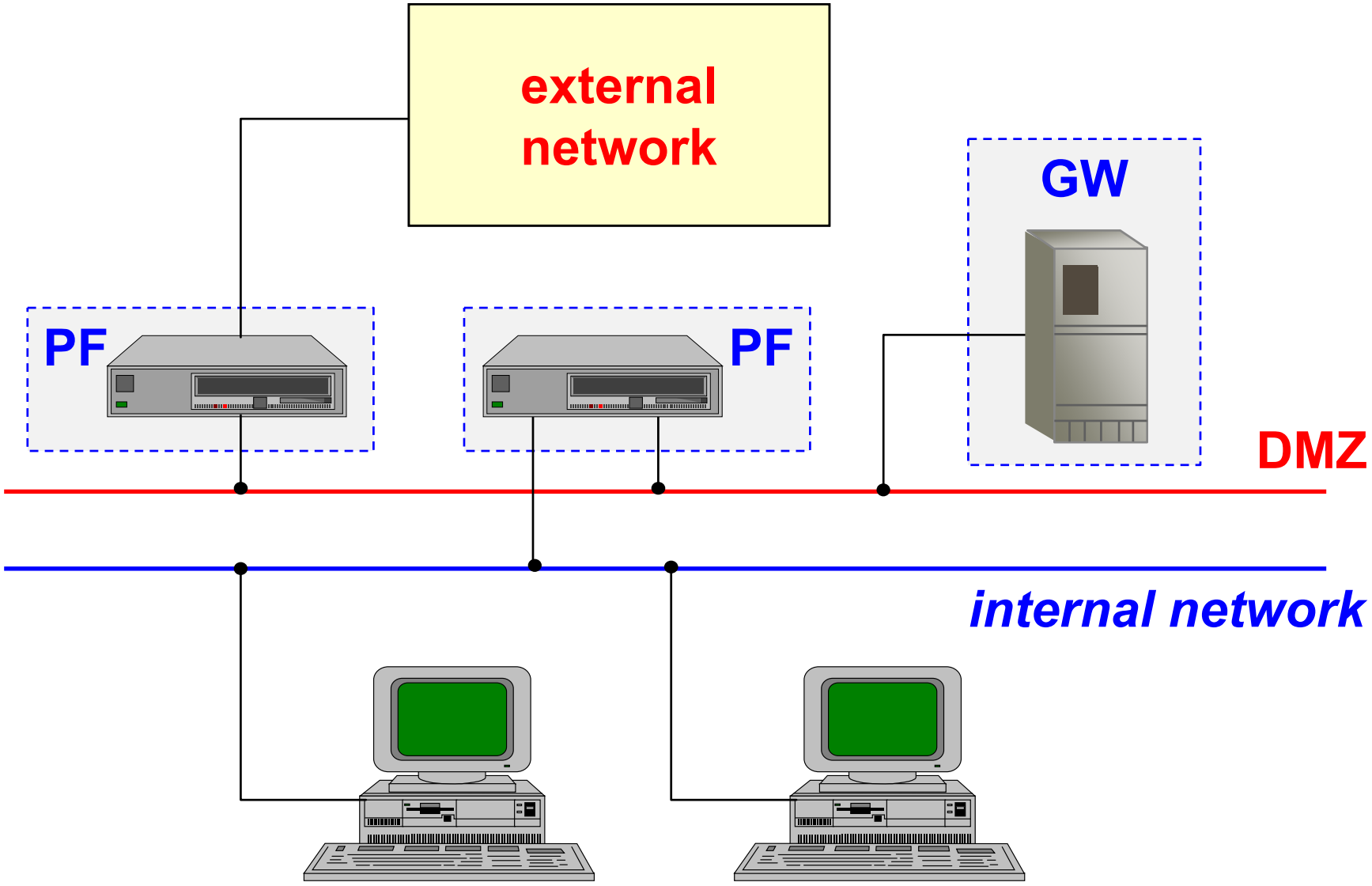
© A.Lioy (Politecnico di Torino, 2005-2024)

"Screened-host" architecture

- **router:**
  - blocks traffic INT > EXT unless from the bastion
  - blocks traffic EXT > INT unless goes to the bastion
  - exception: directly enabled services
- **bastion host runs circuit/application gateway to control the authorized services**
- **more expensive and complex to manage (two systems rather one)**
- **more flexible (skip control over some services / hosts)**
- **only the hosts/protocols passing through the bastion can be masked (unless the PF uses NAT)**

© A.Lioy (Politecnico di Torino, 2005-2024)

"Screened subnet" architecture



© A.Lioy (Politecnico di Torino, 2005-2024)

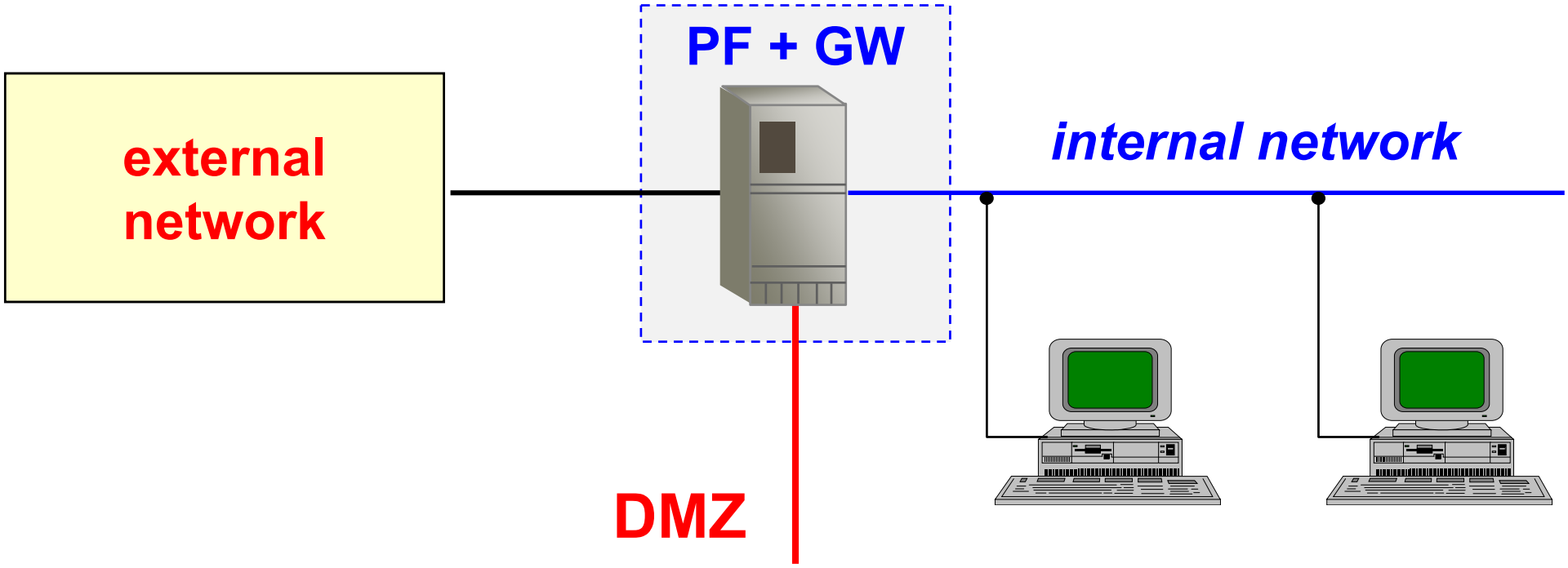
"Screened subnet" architecture

- **DMZ (De-Militarized Zone)**
- **the DMZ is home not only to the gateway but also to other hosts (tipically the public servers):**
  - Web
  - remote access
  - . . .
- **the routing may be configured so that the internal network is unknown**
- **expensive**

© A.Lioy (Politecnico di Torino, 2005-2024)

"Screened subnet" architecture (version 2)

- to reduce costs and simplify management often the PFs are omitted (and their function incorporated into the gateway)
- AKA “three-legged firewall”



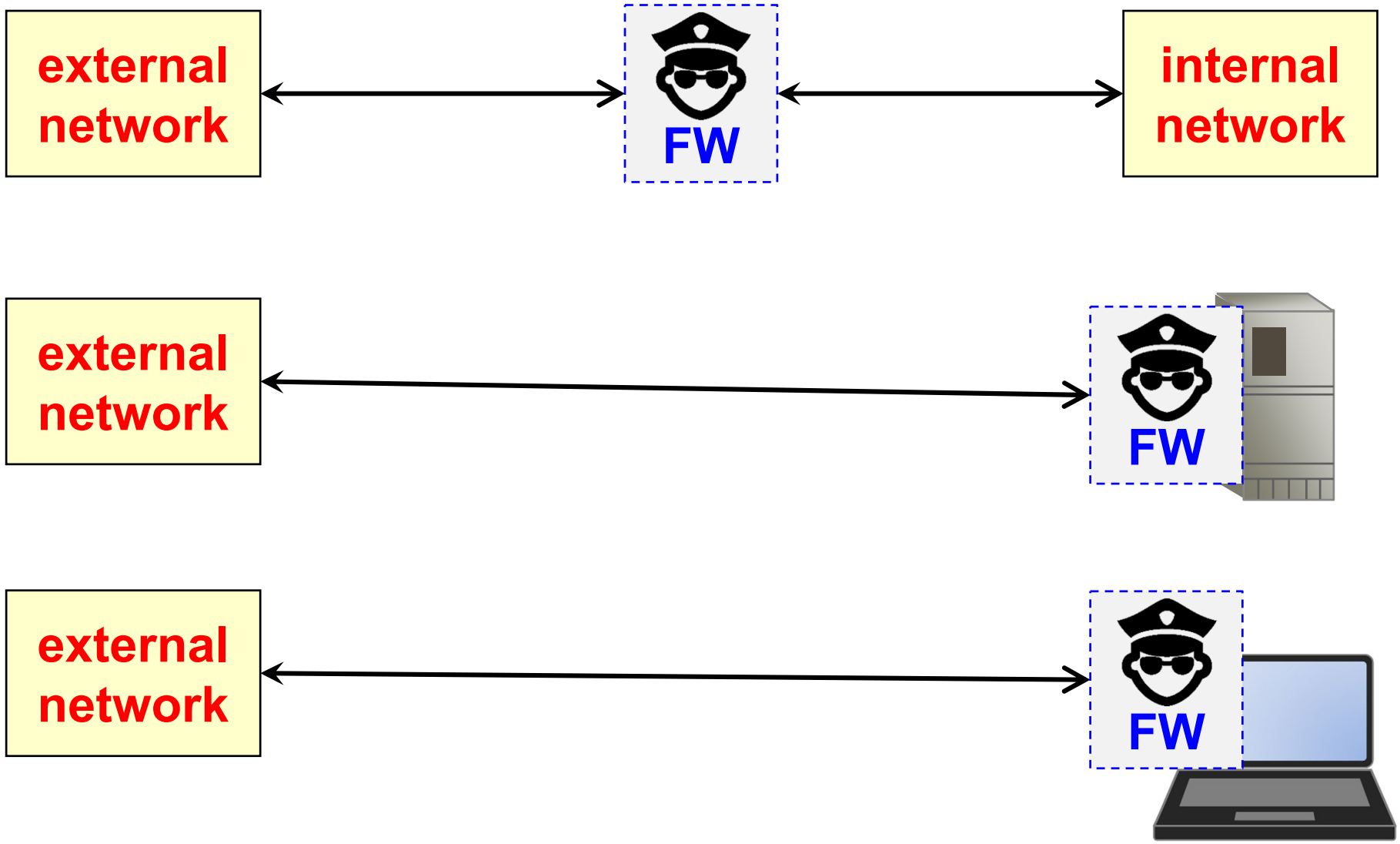
© A.Lioy (Politecnico di Torino, 2005-2024)

Local / personal firewall

- **firewall directly installed at the node to be protected**
- **typically, a packet filter**
- **w.r.t. a normal network firewall, it may limit the PROCESSES that are permitted:**
  - to open network channels towards other nodes (i.e. act as a client)
  - to answer network requests (i.e. act as a server)
- **important to limit the diffusion of malware and trojans, or to avoid configuration mistakes**
- **beware: in order to be effective, the firewall management MUST be separated from system management**

© A.Lioy (Politecnico di Torino, 2005-2024)

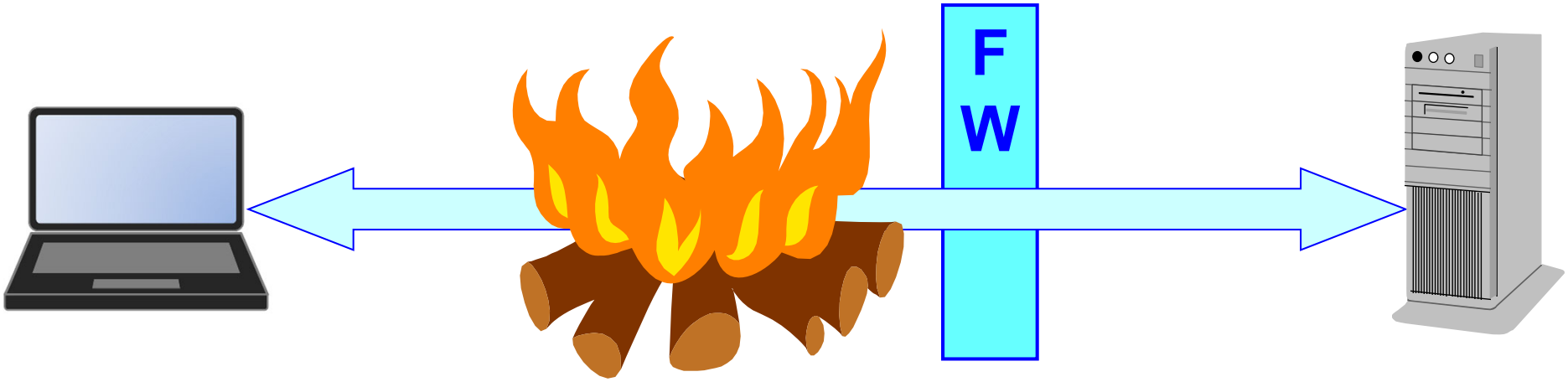
Network / local / personal firewall



© A.Lioy (Politecnico di Torino, 2005-2024)

Protection offered by a firewall

- **a firewall is 100% effective only for attacks over/against blocked channels**
- **the other channels require other protection techniques:**
  - VPN
  - “semantic” firewall / IDS
  - application-level security



© A.Lioy (Politecnico di Torino, 2005-2024)

Intrusion Detection System (IDS)

- **definition:**
  - system to identify actors using a system or a network without authorization (and their actions)
  - extendable to identify authorized actors violating their privileges
- **hypothesis:**
  - the behavioural “pattern” of non-authorized users differs from that of the authorized ones

© A.Lioy (Politecnico di Torino, 2005-2024)

IDS: functional features

- **passive IDS:**
  - detection of effects (e.g. via cryptographic checksum, tripwire)
  - traffic or payload pattern matching (attack/malware signature)
- **active IDS:**
  - “learning” = statistical analysis of the system behaviour
  - “monitoring” = active statistical info collection of traffic, data, sequences, actions
  - “reaction” = comparison against statistical parameters (reaction when a threshold is exceeded)

© A.Lioy (Politecnico di Torino, 2005-2024)

IDS: topological features

- **HIDS (host-based IDS)**
  - log analysis (OS, service or application)
  - internal OS monitoring tools
- **NIDS (network-based IDS)**
  - network traffic monitoring tools

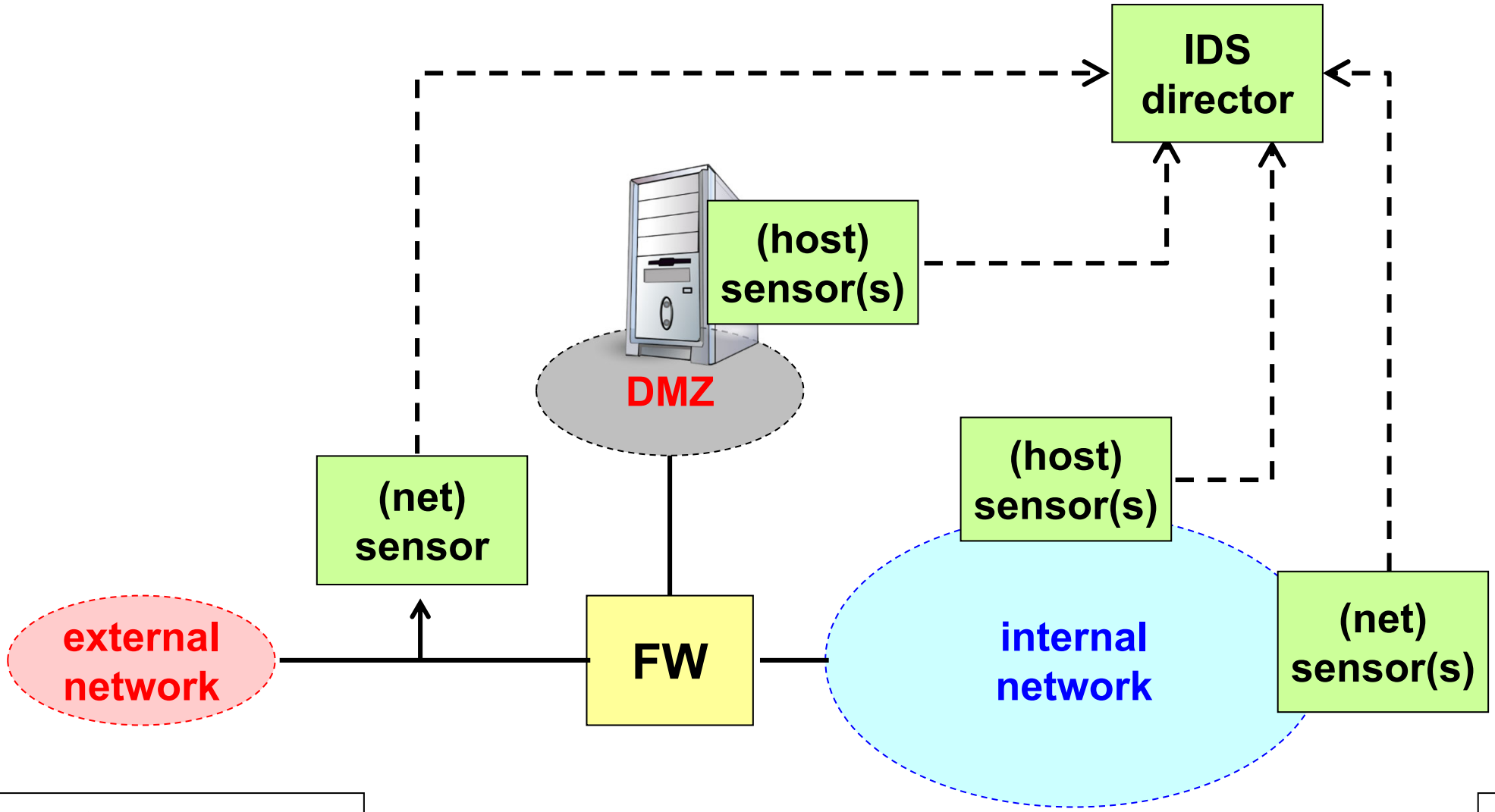
© A.Lioy (Politecnico di Torino, 2005-2024)

NIDS components

- **sensor**
  - checks traffic and logs looking for suspect patterns
  - generated the relevant security events
  - interacts with the system (ACLs, TCP reset, ...)
- **director**
  - coordinates the sensors
  - manages the security database
- **IDS message system**
  - secure and reliable communication among the IDS components

© A.Lioy (Politecnico di Torino, 2005-2024)

NIDS architecture



© A.Lioy (Politecnico di Torino, 2005-2024)

IPS

- Intrusion Prevention System
- to speed-up and automate the reaction to intrusions = IDS + distributed dynamic firewall
- a technology, not a product, with large impact on many elements of the protection system
- dangerous! may take the wrong decision and block innocent traffic
- often integrated in a single product IDPS

© A.Lioy (Politecnico di Torino, 2005-2024)

Next-Generation Firewall (NGFW)

- **application identification**
  - whatever network port is used
  - if possible, deciphering/re-ciphering the traffic
- **user identification**
  - integration with captive portal, 802.1x, or end-point authN (Kerberos, Active Directory, LDAP, …)
- **per-user and per-application policies**
- **filtering based also upon known vulnerabilities, threats, malware, …**

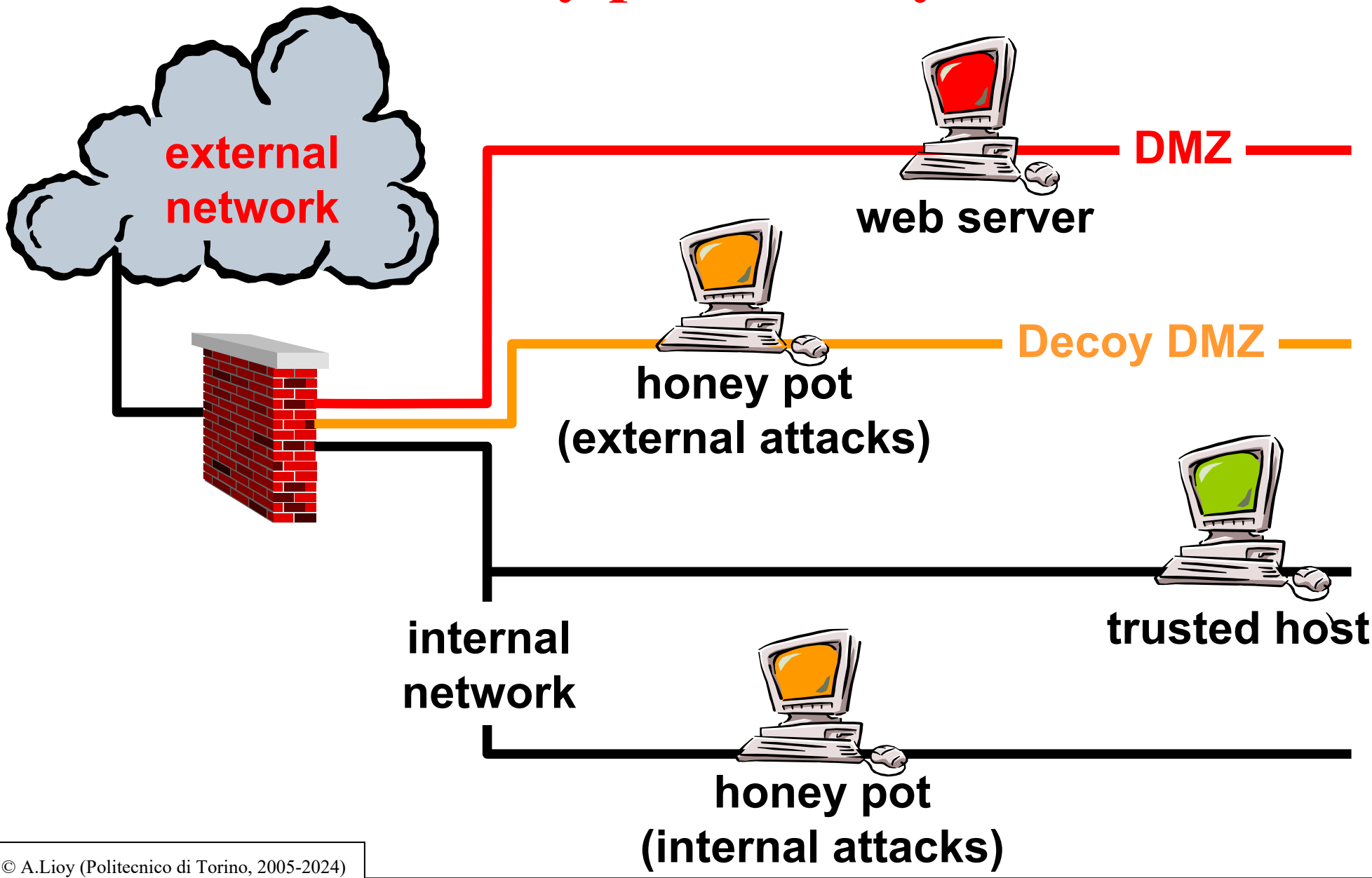
© A.Lioy (Politecnico di Torino, 2005-2024)

Unified Threat Management (UTM)

- **integration of several products in a single device**
  - UTM- or security-appliance
- **firewall, VPN, anti-malware, content-inspection, IDPS, …**
- **the actual capabilities depend upon the manufacturer**
- **mainly targeted to reduce the number of different systems, hence the management complexity and the cost**

© A.Lioy (Politecnico di Torino, 2005-2024)

Honey pot / Honey net



© A.Lioy (Politecnico di Torino, 2005-2024)