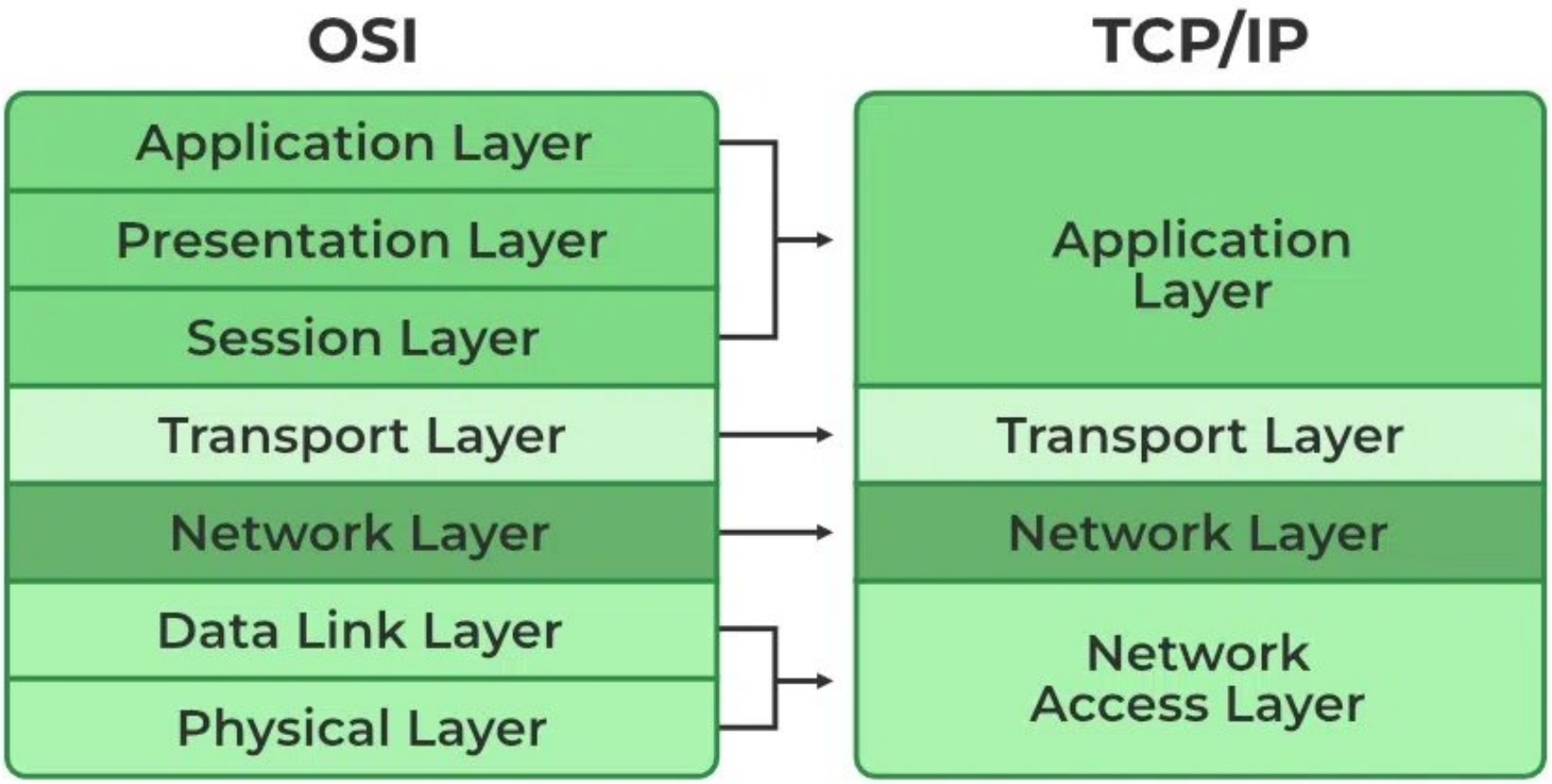
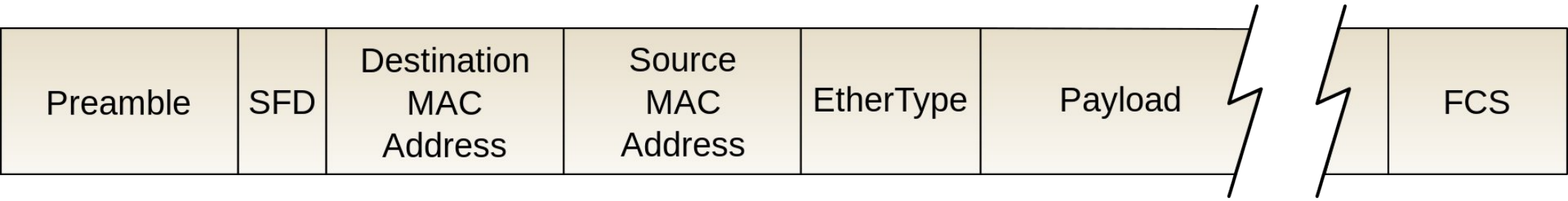
Cybersecurity
a.a. 2025-2026

Franco Arcieri


OSI
TCP/IP
Application Layer
Presentation Layer
Session Layer
Transport Layer
Network Layer
Data Link Layer
Physical Layer
Application Layer
Transport Layer
Network Layer
Network Access Layer

Struttura del pacchetto **ETHERNET**

- 6 bytes Destination Ethernet Address (Tutti 1 se broadcast, …)
- 6 bytes Source Ethernet Address
- 2 bytes Length or Type Field
- per IEEE 802.3 è il numero di bytes di dati
- per ethernet I o II è il "packet type", sempre > 1500(05DC)
- 46 bytes fino a 1500 sono dati! I pacchetti troppo corti devono essere riempiti fino ad almento 46 bytes
- 4 bytes (Frame Check sequence)



[redacted]

Network protocols

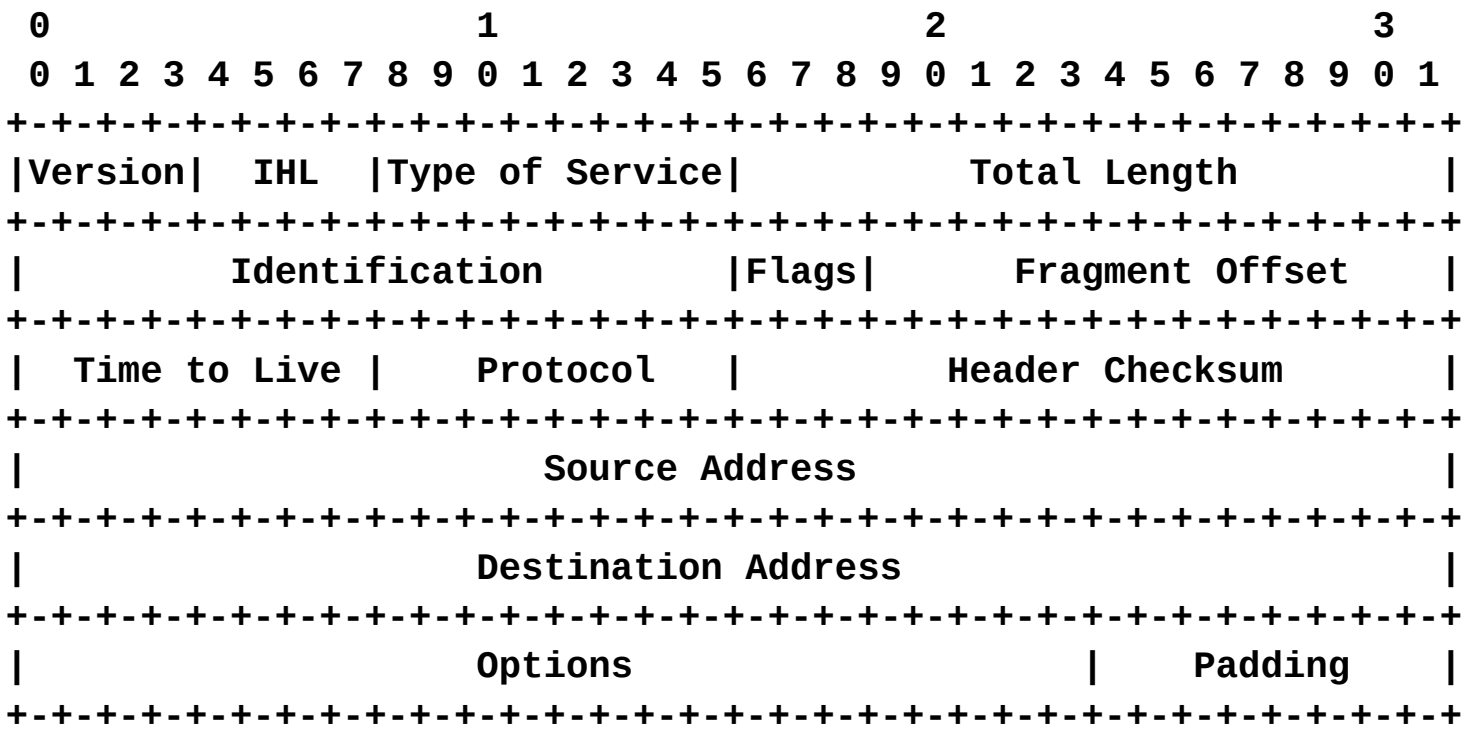
- Descrizioni dei tipi Ethernet
- 0x0000 0x05DC IEEE 802.3 Length Fields
- 0x0600 0x0600 Xerox XNS IDP
- 0x0800 0x0800 DOD IP
- 0x0801 0x0801 X.75 Internet
- 0x0802 0x0802 NBS Internet
- 0x0803 0x0803 ECMA Internet
- 0x0804 0x0804 CHAOSnet
- 0x0805 0x0805 X.25 Level 3
- 0x0806 0x0806 ARP (for IP and CHAOS)
- 0x0807 0x0807 Xerox XNS Compatibility
- 0x081C 0x081C Symbolics Private
- 0x0888 0x088A Xyplex
- 0x0900 0x0900 Ungermann-Bass network debugger
- 0x0A00 0x0A00 Xerox 802.3 PUP
- 0x0A01 0x0A01 Xerox 802.3 PUP Address Translation
- 0x0A02 0x0A02 Xerox PUP CAL Protocol (unused)
- 0x0BAD 0x0BAD Banyan Systems, Inc.
- 0x1000 0x1000 Berkeley Trailer negotiation

I protocolli incapsulati in ethernet

- 0x0000, 0x05DC, IEEE 802.3 Length Fields
- 0x0600, 0x0600, Xerox XNS IDP
- 0x0800, 0x0800, DOD IP
- 0x0801, 0x0801, X.75 Internet
- 0x0802, 0x0802, NBS Internet
- 0x0803, 0x0803, ECMA Internet
- 0x0804, 0x0804, CHAOSnet
- 0x0805, 0x0805, X.25 Level 3
- 0x0806, 0x0806, ARP (for IP and CHAOS)
- 0x0807, 0x0807, Xerox XNS Compatibility
- 0x081C, 0x081C, Symbolics Private
- 0x0888, 0x088A, Xyplex
- 0x0900, 0x0900, Ungermann-Bass network debugger
- 0x0A00, 0x0A00, Xerox 802.3 PUP
- 0x0A01, 0x0A01, Xerox 802.3 PUP Address Translation
- 0x0A02, 0x0A02, Xerox PUP CAL Protocol (unused)
- 0x0BAD, 0x0BAD, Banyan Systems, Inc.
- 0x1000, 0x1000, Berkeley Trailer negotiation
- 0x1001, 0x100F, Berkeley Trailer encapsulation for IP
- 0x1066, 0x1066, VALIS Systems
- 0x1600, 0x1600, VALID Systems
- 0x3C01, 0x3C0D, 3Com Corporation
- 0x3C10, 0x3C14, 3Com Corporation
- 0x4242, 0x4242, PCS Basic Block Protocol
- 0x5208, 0x5208, BBN Simnet Private
- 0x6000, 0x6000, DEC Unassigned
- 0x6001, 0x6001, DEC MOP Dump/Load Assistance
- 0x6002, 0x6002, DEC MOP Remote Console
- 0x6003, 0x6003, DEC DECnet Phase IV
- 0x6004, 0x6004, DEC LAT
- 0x6005, 0x6005, DEC DECnet Diagnostic Protocol: DECnet Customer Use
- 0x6007, 0x6007, DEC DECnet LAVC
- 0x6008, 0x6008, DEC Amber
- 0x6009, 0x6009, DEC MUMPS
- 0x6010, 0x6014, 3Com Corporation
- 0x7000, 0x7000, Ungermann-Bass download
- 0x7001, 0x7001, Ungermann-Bass NIU
- 0x7002, 0x7002, Ungermann-Bass diagnostic/loopback
- 0x7007, 0x7007, OS/9 Microware
- 0x7020, 0x7028, LRT (England)
- 0x7030, 0x7030, Proteon
- 0x7034, 0x7034, Cabletron
- 0x8003, 0x8003, Cronus VLN
- 0x8004, 0x8004, Cronus Direct
- 0x8013, 0x8013, SGI diagnostic type (obsolete)
- 0x8014, 0x8014, SGI network games (obsolete)
- 0x8015, 0x8015, SGI reserved type (obsolete)
- 0x8016, 0x8016, SGI bounce server (obsolete)
- 0x8019, 0x8019, Apollo
- 0x802E, 0x802E, Tymshare
- 0x802F, 0x802F, Tigan, Inc.
- 0x8035, 0x8035, Reverse ARP (RARP)
- 0x8036, 0x8036, Aeonic Systems
- 0x8038, 0x8038, DEC LANBridge
- 0x8039, 0x8039, DEC DSM
- 0x803A, 0x803A, DEC Aragon
- 0x803B, 0x803B, DEC VAXELN
- 0x803C, 0x803C, DEC NSMV
- 0x803D, 0x803D, DEC Ethernet CSMA/CD Encryption Protocol
- 0x803E, 0x803E, DEC DNA
- 0x803F, 0x803F, DEC LAN Traffic Monitor
- 0x8040, 0x8040, DEC NetBIOS
- 0x8041, 0x8041, DEC MS/DOS
- 0x8042, 0x8042, DEC Unassigned
- 0x8044, 0x8044, Planning Research Corporation
- 0x8046, 0x8046, AT&T
- 0x8047, 0x8047, AT&T
- 0x8049, 0x8049, ExperData (France)
- 0x805B, 0x805B, VMTP (Versatile Message Transaction Protocol, RFC-1045, Stanford)
- 0x805C, 0x805C, Stanford V Kernel production, Version 6.0
- 0x805D, 0x805D, Evans & Sutherland
- 0x8060, 0x8060, Little Machines
- 0x8062, 0x8062, Counterpoint Computers
- 0x8065, 0x8065, University of Massachusetts, Amherst
- 0x8066, 0x8066, University of Massachusetts, Amherst
- 0x8067, 0x8067, Veeco Integrated Automation
- 0x8068, 0x8068, General Dynamics
- 0x8069, 0x8069, AT&T
- 0x806A, 0x806A, Autophon (Switzerland)
- 0x806C, 0x806C, ComDesign
- 0x806D, 0x806D, Compugraphic Corporation
- 0x806E, 0x8077, Landmark Graphics Corporation
- 0x807A, 0x807A, Matra (France)
- 0x807B, 0x807B, Dansk Data Elektronic A/S (Denmark)
- 0x807C, 0x807C, Merit Intermodal
- 0x807D, 0x807D, VitaLink Communications
- 0x807E, 0x807E, VitaLink Communications
- 0x807F, 0x807F, VitaLink Communications
- 0x8088, 0x8088, Xyplex
- 0x8089, 0x8089, Xyplex
- 0x808A, 0x808A, Xyplex
- 0x809B, 0x809B, AppleTalk and Kinetics AppleTalk over Ethernet
- 0x809C, 0x809C, Datability
- 0x809D, 0x809D, Datability
- 0x809E, 0x809E, Datability
- 0x809F, 0x809F, Spider Systems, Ltd. (England)
- 0x80A3, 0x80A3, Nixdorf Computer (West Germany)
- 0x80A4, 0x80B3, Siemens Gammasonics, Inc.
- 0x80C0, 0x80C0, Digital Communication Associates
- 0x80C1, 0x80C1, Digital Communication Associates
- 0x80C2, 0x80C2, Digital Communication Associates
- 0x80C3, 0x80C3, Digital Communication Associates
- 0x80C6, 0x80C6, Pacer Software
- 0x80C7, 0x80C7, Applitek Corporation
- 0x80C8, 0x80CC, Intergraph Corporation
- 0x80CD, 0x80CD, Harris Corporation
- 0x80CE, 0x80CE, Harris Corporation
- 0x80CF, 0x80D2, Taylor Inst.
- 0x80D3, 0x80D3, Rosemount Corporation
- 0x80D4, 0x80D4, Rosemount Corporation
- 0x80D5, 0x80D5, IBM SNA Services over Ethernet
- 0x80DD, 0x80DD, Varian Associates
- 0x80DE, 0x80DE, Integrated Solutions TRFS (Transparent Remote File System)
- 0x80DF, 0x80DF, Integrated Solutions
- 0x80E0, 0x80E3, Allen-Bradley
- 0x80E4, 0x80F0, Datability
- 0x80F2, 0x80F2, Retix
- 0x80F3, 0x80F3, Kinetics, AppleTalk ARP (AARP)
- 0x80F4, 0x80F4, Kinetics
- 0x80F5, 0x80F5, Kinetics
- 0x80F7, 0x80F7, Apollo Computer
- 0x80FF, 0x8103, Wellfleet Communications
- 0x8107, 0x8107, Symbolics Private
- 0x8108, 0x8108, Symbolics Private
- 0x8109, 0x8109, Symbolics Private
- 0x8130, 0x8130, Waterloo Microsystems
- 0x8131, 0x8131, VG Laboratory Systems
- 0x8137, 0x8137, Novell (old) NetWare IPX
- 0x8138, 0x8138, Novell
- 0x8139, 0x813D, KTI
- 0x9000, 0x9000, Loopback (Configuration Test Protocol)
- 0x9001, 0x9001, Bridge Communications XNS Systems Management

Il protocollo IP

Struttura dei pacchetti DOD IP (a partire dal byte 14 del pacchetto ETHERNET ovvero dal primo byte dei dati a livello ethernet)
Documenti di riferimento: RFC791, www.protocols.com



- Version
  - Version field indicates the format of the Internet header.
- IHL
  - Internet header length is the length of the Internet header in 32-bit words. Points to the beginning of the data. The minimum value for a correct header is 5.
- Type of service
  - Indicates the quality of service desired. Networks may offer service precedence, meaning that they accept traffic only above a certain precedence at times of high load. There is a three-way trade-off between low delay, high reliability and high throughput.
  - Bits 0-2: Precedence
    - 111 Network control.
    - 110 Internetwork control.
    - 101 CRITIC/ECP.
    - 100 Flash override.
    - 011 Flash.
    - 010 Immediate.
    - 001 Priority.
    - 000 Routine.
  - Bit 3: Delay
    - 0 Normal delay.
    - 1 Low delay.
  - Bit 4: Throughput
    - 0 Normal throughput.
    - 1 High throughput.
  - Bit 5: Reliability
    - 0 Normal reliability.
    - 1 High reliability.
  - Bits 6-7: Reserved for future use.
- Total length
  - Length of the datagram measured in bytes, including the Internet header and data. This field allows the length of a datagram to be up to 65,535 bytes, although such long datagrams are impractical for most hosts and networks. All hosts must be prepared to accept datagrams of up to 576 bytes, regardless of whether they arrive whole or in fragments. It is recommended that hosts send datagrams larger than 576 bytes only if the destination is prepared to accept the larger datagrams.

- Flags
  - 3 bits. Control flags:
    - Bit 0 is reserved and must be zero.
    - Bit 1: Don’t fragment bit:
    - 0 May fragment.
    - 1 Don’t fragment.
    - Bit 2: More fragments bit:
    - 0 Last fragment.
    - 1 More fragments.
- Fragment offset
  - 13 bits. Indicates where this fragment belongs in the datagram. The fragment offset is measured in units of 8 bytes (64 bits). The first fragment has offset zero.
- Time to live
  - Indicates the maximum time the datagram is allowed to remain in the Internet system. If this field contains the value zero, the datagram must be destroyed. This field is modified in Internet header processing. The time is measured in units of seconds. However, since every module that processes a datagram must decrease the TTL by at least one (even if it processes the datagram in less than 1 second), the TTL must be thought of only as an upper limit on the time a datagram may exist. The intention is to cause undeliverable datagrams to be discarded and to bound the maximum datagram lifetime.
- Protocol
  - Indicates the next level protocol used in the data portion of the Internet datagram.
- Header checksum
  - A checksum on the header only. Since some header fields change, e.g., Time To Live, this is recomputed and verified at each point that the Internet header is processed.
- Source address / destination address
  - 32 bits each. A distinction is made between names, addresses and routes. A name indicates an object to be sought. An address indicates the location of the object. A route indicates how to arrive at the object. The Internet protocol deals primarily with addresses. It is the task of higher level protocols (such as host-to-host or application) to make the mapping from names to addresses. The Internet module maps Internet addresses to local net addresses. It is the task of lower level procedures (such as local net or gateways) to make the mapping from local net addresses to routes.

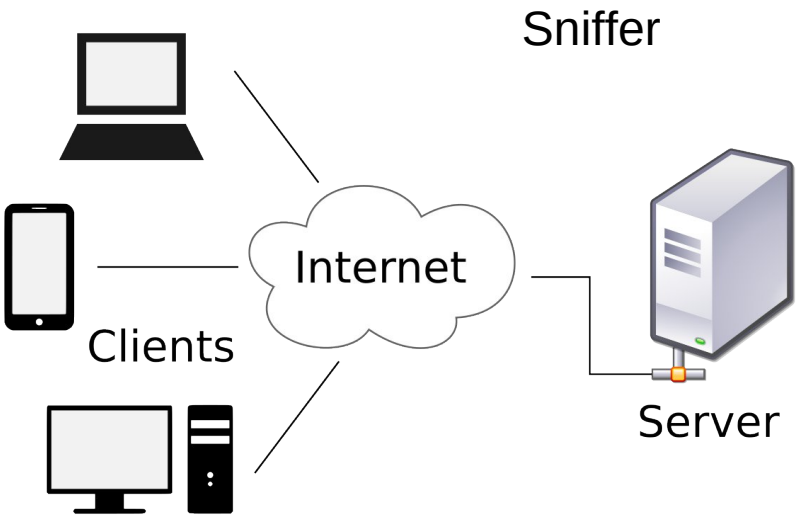
- Options
  - Options may or may not appear in datagrams. They must be implemented by all IP modules (host and gateways). What is optional is their transmission in any particular datagram, not their implementation. In some environments, the security option may be required in all datagrams.
- The option field is variable in length. There may be zero or more options. There are two possible formats for an option:
  - A single octet of option type.
  - An option type octet, an option length octet and the actual option data octets.
- The length octet includes the option type octet and the actual option data octets.
- The option type octet has 3 fields:
  - 1 bit: Copied flag. Indicates that this option is copied into all fragments during fragmentation:
    - 0 Copied.
    - 1 Not copied.
  - 2 bits: Option class
    - 0 Control.
    - 1 Reserved for future use.
    - 2 Debugging and measurement.
    - 3 Reserved for future use.
- 5 bits: Option number.
- Data
  - IP data or higher layer protocol header.

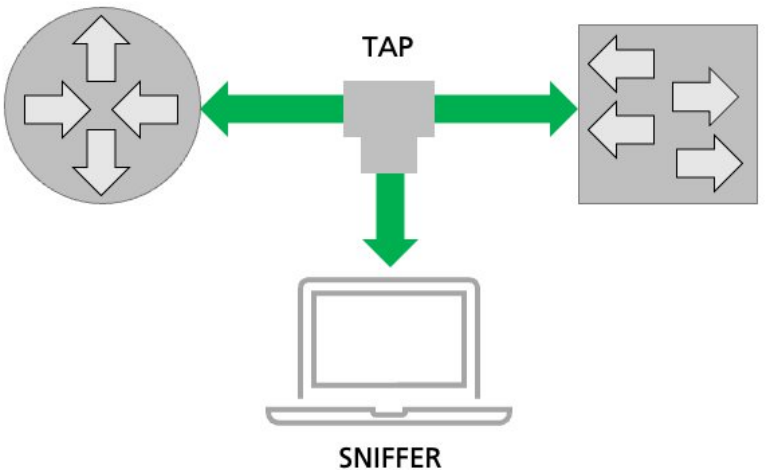
I protocolli incapsulati in IP

0, Reserved, Reserved
1, ICMP, Internet Control Message
2, IGMP, Internet Group Management
3, GGP, Gateway-to-Gateway
4, IP, IP in IP (encasulation)
5, ST, Stream
6, TCP, Transmission Control
7, UCL, UCL
8, EGP, Exterior Gateway Protocol
9, IGP, any private interior gateway
10, BBN-RCC-MON, BBN RCC Monitoring
11, NVP-II, Network Voice Protocol
12, PUP, PUP
13, ARGUS, ARGUS
14, EMCON, EMCON
15, XNET, Cross Net Debugger
16, CHAOS, Chaos
17, UDP, User Datagram
18, MUX, Multiplexing
19, DCN-MEAS, DCN Measurement Subsystems
20, HMP, Host Monitoring
21, PRM, Packet Radio Measurement
22, XNS-IDP, XEROX NS IDP
23, TRUNK-1, Trunk-1
24, TRUNK-2, Trunk-2
25, LEAF-1, Leaf-1
26, LEAF-2, Leaf-2
27, RDP, Reliable Data Protocol
28, IRTP, Internet Reliable Transaction
29, ISO-TP4, ISO Transport Protocol Class 4
30, NETBLT, Bulk Data Transfer Protocol
31, MFE-NSP, MFE Network Services Protocol
32, MERIT-INP, MERIT Internodal Protocol
33, SEP, Sequential Exchange Protocol
34, 3PC, Third Party Connect Protocol
35, IDPR, Inter-Domain Policy Routing Protocol
36, XTP, XTP
37, DDP, Datagram Delivery Protocol
38, IDPR-CMTP, IDPR Control Message Transport Protocol
39, TP++, TP++ Transport Protocol
40, IL, IL Transport Protocol
41, SIP, Simple Internet Protocol
42, SDRP, Source Demand Routing Protocol
43, SIP-SR, SIP Source Route
44, SIP-FRAG, SIP Fragment
45, IDRP, Inter-Domain Routing Protocol
46, RSVP, Reservation Protocol
47, GRE, General Routing Encapsulation
48, MHRP, Mobile Host Routing Protocol
49, BNA, BNA
50, SIPP-ESP, SIPP Encap Security Payload
51, SIPP-AH, SIPP Authentication Header
52, I-NLSP, Integrated Net Layer Security TUBA
53, SWIPE, IP with Encryption
54, NHRP, NBMA Next Hop Resolution Protocol
61, any host, any host internal protocol
62, CFTP, CFTP
63, any net, local network
64, SAT-EXPAK, SATNET and Backroom EXPAK
65, KRYPTOLAN, Kryptolan
66, RVD, MIT Remote Virtual Disk Protocol
67, IPPC, Internet Pluribus Packet Core
68, any file, distributed file system
69, SAT-MON, SATNET Monitoring
70, VISA, VISA Protocol
71, IPCV, Internet Packet Core Utility
72, CPNX, Computer Protocol Network Executive
73, CPHB, Computer Protocol Heart Beat
74, WSN, Wang Span Network
75, PVP, Packet Video Protocol
76, BR-SAT-MON, Backroom SATNET Monitoring
77, SUN-ND, SUN ND PROTOCOL-Temporary
78, WB-MON, WIDEBAND Monitoring
79, WB-EXPAK, WIDEBAND EXPAK
80, ISO-IP, ISO Internet Protocol
81, VMTP, VMTP
82, SECURE-VMTP, SECURE-VMTP
83, VINES, VINES
84, TTP, TTP
85, NSFNET-IGP, NSFNET-IGP
86, DGP, Dissimilar Gateway Protocol
87, TCF, TCF
88, IGRP, IGRP
89, OSPFIGP, OSPFIGP
90, Sprite-RPC, Sprite RPC Protocol
91, LARP, Locus Address Resolution Protocol
92, MTP, Multicast Transport Protocol
93, AX.25, AX.25 Frames
94, IPIP, IP-within-IP Encapsulation Protocol
95, MICP, Mobile Internetworking Control Pro.
96, SCC-SP, Semaphore Communications Sec. Pro.
97, ETHERIP, Ethernet-within-IP Encapsulation
98, ENCAP, Encapsulation Header
99, any scheme, any private encryption scheme
100, GMTP, GMTP
255, Reserved, Reserved

Sniffing della rete

- tcpdump, tool più famoso per fare sniffing
  - Wireshark il più semplice





ARP

- The Address Resolution Protocol uses a simple message format containing one address resolution request or response. The packets are carried at the data link layer of the underlying network as raw payload. In the case of Ethernet, a 0x0806 EtherType value is used to identify ARP frames.
- The size of the ARP message depends on the link layer and network layer address sizes. The message header specifies the types of network in use at each layer as well as the size of addresses of each. The message header is completed with the operation code for request (1) and reply (2). The payload of the packet consists of four addresses, the hardware and protocol address of the sender and receiver hosts.
- The principal packet structure of ARP packets is shown in the following table which illustrates the case of IPv4 networks running on Ethernet. In this scenario, the packet has 48-bit fields for the sender hardware address (SHA) and target hardware address (THA), and 32-bit fields for the corresponding sender and target protocol addresses (SPA and TPA). The ARP packet size in this case is 28 bytes.

ARP

Internet Protocol (IPv4) over Ethernet ARP packet

Octet offset	0	1
0	Hardware type (HTYPE)	
2	Protocol type (PTYPE)	
4	Hardware address length (HLEN)	Protocol address length (PLEN)
6	Operation (OPER)	
8	Sender hardware address (SHA) (first 2 bytes)	
10	(next 2 bytes)	
12	(last 2 bytes)	
14	Sender protocol address (SPA) (first 2 bytes)	
16	(last 2 bytes)	
18	Target hardware address (THA) (first 2 bytes)	
20	(next 2 bytes)	
22	(last 2 bytes)	
24	Target protocol address (TPA) (first 2 bytes)	
26	(last 2 bytes)	

- Hardware type (HTYPE)
  - This field specifies the network link protocol type. Example: Ethernet is 1.[2]
- Protocol type (PTYPE)
  - This field specifies the internetwork protocol for which the ARP request is intended. For IPv4, this has the value 0x0800. The permitted PTYPE values share a numbering space with those for EtherType.[2][3]
- Hardware length (HLEN)
  - Length (in octets) of a hardware address. Ethernet address length is 6.
- Protocol length (PLEN)
  - Length (in octets) of internetwork addresses. The internetwork protocol is specified in PTYPE. Example: IPv4 address length is 4.
- Operation
  - Specifies the operation that the sender is performing: 1 for request, 2 for reply.
- Sender hardware address (SHA)
  - Media address of the sender. In an ARP request this field is used to indicate the address of the host sending the request. In an ARP reply this field is used to indicate the address of the host that the request was looking for.
- Sender protocol address (SPA)
  - Internetwork address of the sender.
- Target hardware address (THA)
  - Media address of the intended receiver. In an ARP request this field is ignored. In an ARP reply this field is used to indicate the address of the host that originated the ARP request.
- Target protocol address (TPA)
  - Internetwork address of the intended receiver.
- ARP protocol parameter values have been standardized and are maintained by the Internet Assigned Numbers Authority (IANA).[2]
- The EtherType for ARP is 0x0806. This appears in the Ethernet frame header when the payload is an ARP packet and is not to be confused with PTYPE, which appears within this encapsulated ARP packet.

ARP

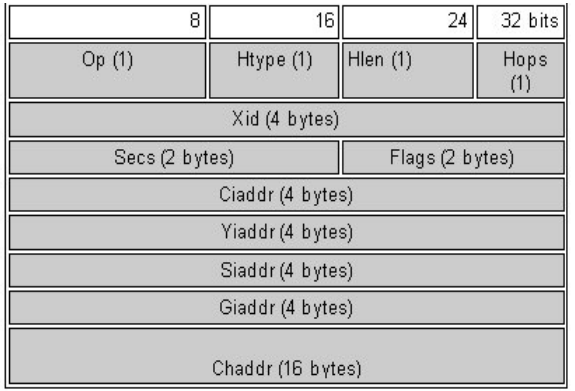
- Con il protocollo ARP posso scoprire chi ha uno specifico indirizzo IP
- Ma come posso scoprire la topologia della mia rete?
  - In particolare quante schede di rete / device sono presenti nella mia rete locale?
- Ma come faccio a andare su internet, o meglio, come posso comunicare con un dispositivo che si trova su un’altra rete locale?
- Posso scoprire la topologia di una rete locale cui non appartengo?

Cerchiamo gli indirizzi MAC della mia rete locale

- Conoscere il vostro MAC address
- Conoscere il vostro IP address
- Inviare una richiesta ARP a tutti gli indirizzi IP definiti tramite la maschera di sottorete (in genere 255.255.255.0 e quindi i 256 indirizzi IP che ricavate dal vostro IP sostituendo al byte meno significativo di valori da 0 a 255
- Per le richieste di cui avete risposta, allora conoscerete l’associazione MAC address, indirizzo IP

Struttura dei pacchetti DHCP (immediatamente successivi ai pacchetti DOD IP)

- Documenti di riferimento: RFC1531, www.protocols.com
- The Dynamic Host Configuration Protocol (DHCP) provides Internet hosts with configuration parameters. DHCP is an extension of BOOTP. DHCP consists of two components: a protocol for delivering host-specific configuration parameters from a DHCP server to a host and a mechanism for allocation of network addresses to hosts.



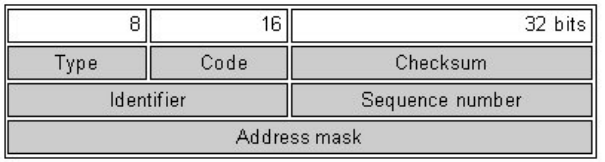
DHCP header structure

- DHCP header structure
  - Op
    - The message operation code. Messages can be either BOOTREQUEST or BOOTREPLY.
  - Htype
    - The hardware address type.
  - Hlen
    - The hardware address length.
  - Xid
    - The transaction ID.
  - Secs
    - The seconds elapsed since the client began the address acquisition or renewal process.
  - Flags
    - The flags.
  - Ciaddr
    - The client IP address.
  - Yiaddr
    - The "Your" (client) IP address.
  - Siaddr
    - The IP address of the next server to use in bootstrap.
  - Giaddr
    - The relay agent IP address used in booting via a relay agent.

ICMP

- Struttura dei pacchetti ICMP (immediatamente successivi ai pacchetti IP)
- Documenti di riferimento: RFC792, www.protocols.com
- IETF RFC792 defines the Internet Control Message Protocol (ICMP). ICMP messages generally contain information about routing difficulties with IP datagrams or simple exchanges such as time-stamp or echo transactions.
- ICMP header structure

Type	Code	Description
0		Echo reply.
3		Destination unreachable.
3	*0*	*Net unreachable.*
3	*1*	*Host unreachable.*
3	*2*	*Protocol unreachable.*
3	*3*	*Port unreachable.*
3	*4*	*Fragmentation needed and DF set.*
3	*5*	Source route failed.
4		Source quench.
5		Redirect.
5	0	Redirect datagrams for the network.
5	1	Redirect datagrams for the host.
5	2	Redirect datagrams for the type of service and network.
5	3	Redirect datagrams for the type of service and host.
8		Echo.
11		Time exceeded.
11	0	Time to live exceeded in transit.
11	1	Fragment reassemble time exceeded.
12		Parameter problem.
13		Timestamp.
14		Timestamp reply.
15		Information request.
16		Information reply.

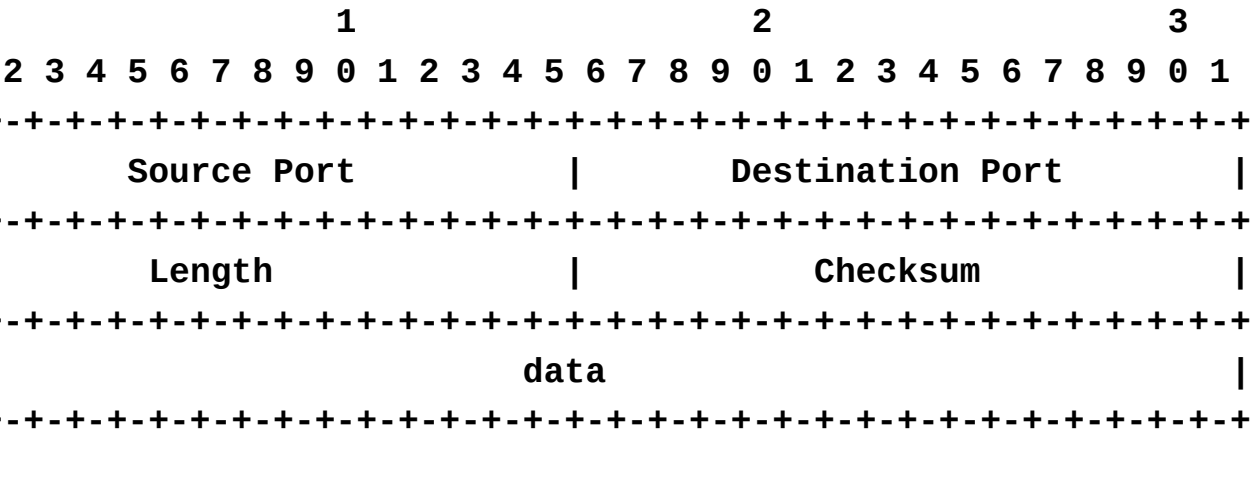


ICMP header structure

- Checksum
  - The 16-bit one’s complement of the one’s complement sum of the ICMP message starting with the ICMP Type. For computing the checksum, the checksum field should be zero.
- Identifier
  - An identifier to aid in matching requests/replies; may be zero.
- Sequence number
  - Sequence number to aid in matching requests/replies; may be zero.
- Address mask
  - A 32-bit mask.

UDP

- **Struttura dei pacchetti UDP (immediatamente successivi ai pacchetti DOD IP)**
- **Documenti di riferimento: RFC768, www.protocols.com**

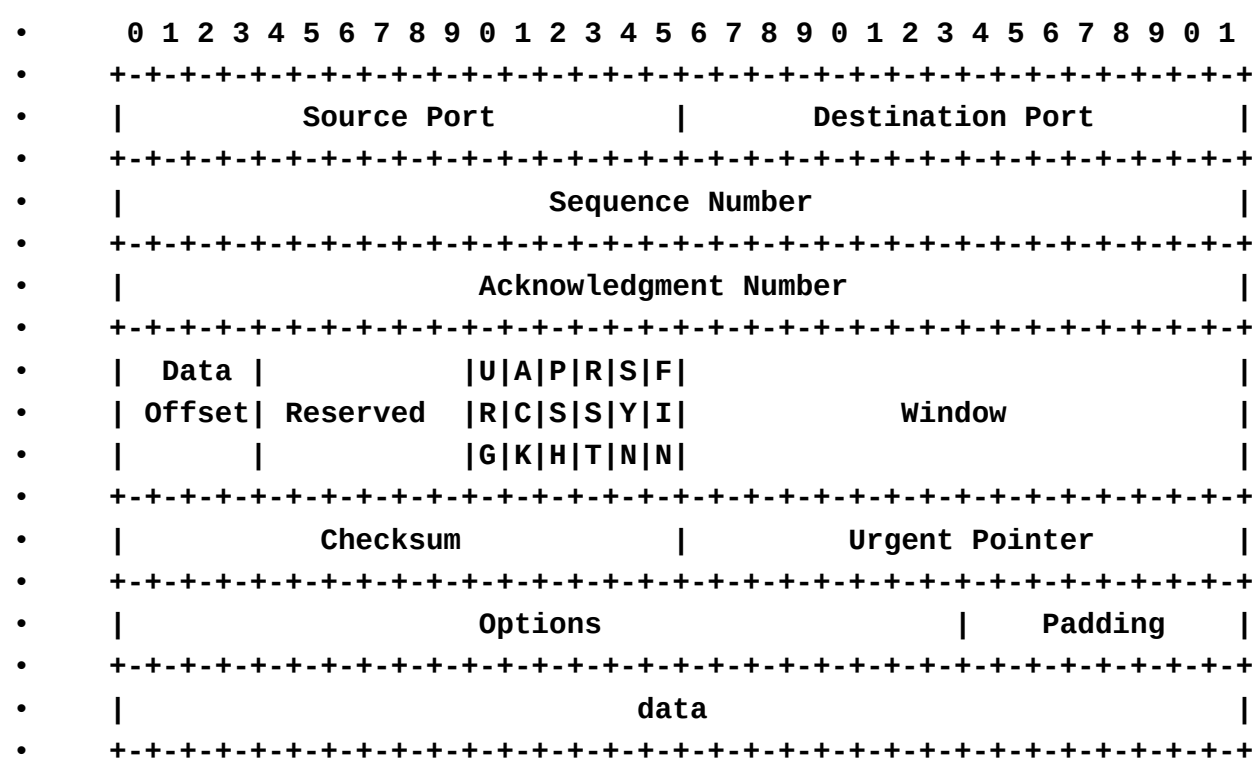


TCP Header Format

- **The User Datagram Protocol (UDP), defined by IETF RFC768, provides a simple, but unreliable message service for transaction-oriented services. Each UDP header carries**

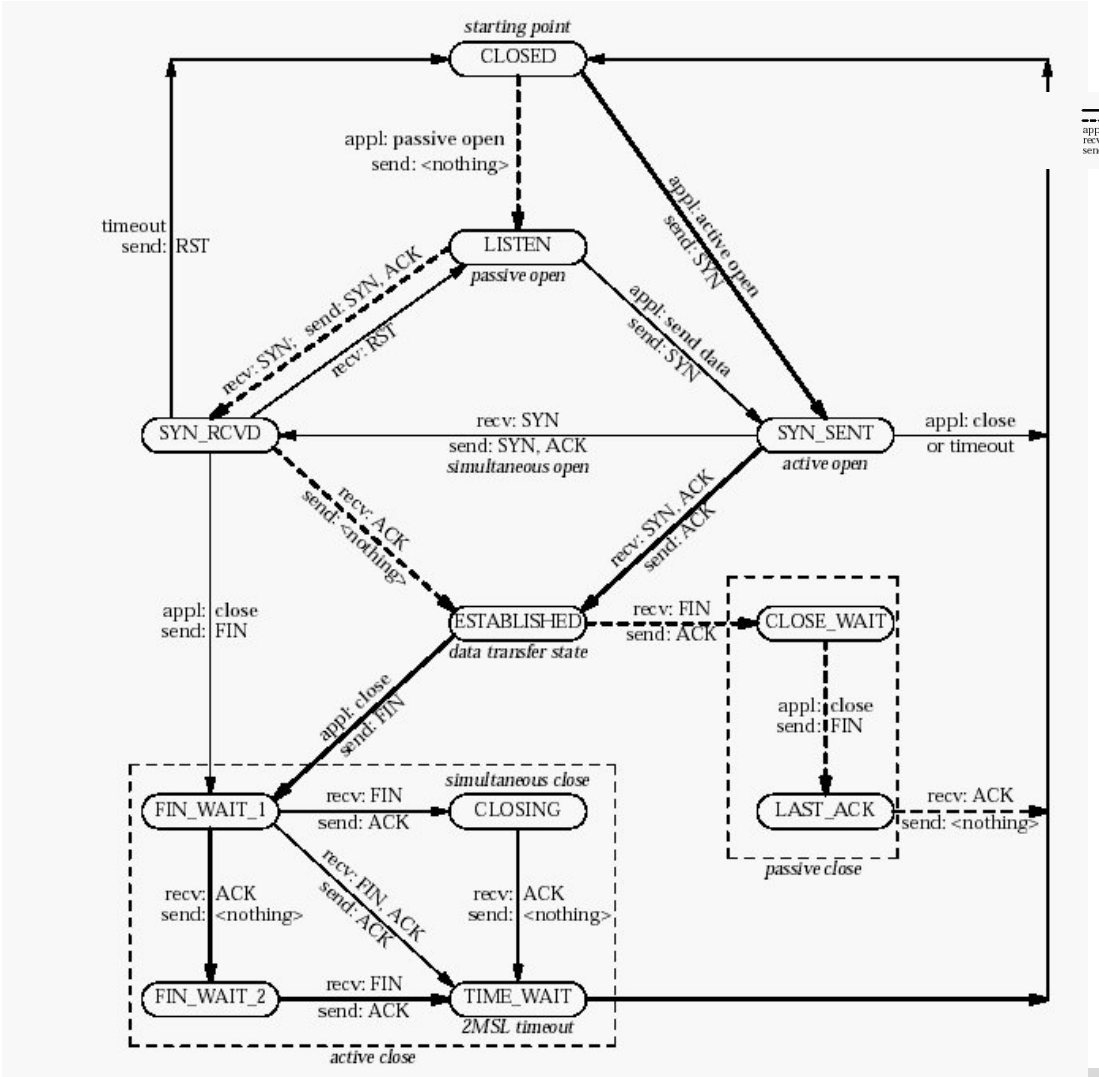
TCP

- **Documenti di riferimento: RFC793, www.protocols.com**
- **Struttura dei pacchetti TCP (immediatamente successivi ai pacchetti DOD IP)**



- Source port
  - Source port number.
- Destination port
  - Destination port number.
- Sequence number
  - The sequence number of the first data octet in this segment (except when SYN is present). If SYN is present, the sequence number is the initial sequence number (ISN) and the first data octet is ISN+1.
- Acknowledgment number
  - If the ACK control bit is set, this field contains the value of the next sequence number which the sender of the segment is expecting to receive. Once a connection is established, this value is always sent.
- Data offset
  - 4 bits. The number of 32-bit words in the TCP header, which indicates where the data begins. The TCP header (even one including options) has a length which is an integral number of 32 bits.
- Reserved
  - 6 bits. Reserved for future use. Must be zero.
- Control bits
  - 6 bits. The control bits may be (from right to left):
  - U (URG) Urgent pointer field significant.
  - A (ACK) Acknowledgment field significant.
  - P (PSH) Push function.
  - R (RST) Reset the connection.
  - S (SYN) Synchronize sequence numbers.
  - F (FIN) No more data from sender.
- Window
  - 16 bits. The number of data octets which the sender of this segment is willing to accept, beginning with the octet indicated in the
- Checksum
  - 16 bits. The checksum field is the 16 bit one’s complement of the one’s complement sum of all 16-bit words in the header and text. If a segment contains an odd number of header and text octets to be checksummed, the last octet is padded on the right with zeros to form a 16-bit word for checksum purposes. The pad is not transmitted as part of the segment. While computing the checksum, the checksum field itself is replaced with zeros.
- Urgent Pointer
  - 16 bits. This field communicates the current value of the urgent pointer as a positive offset from the sequence number in this segment. The urgent pointer points to the sequence number of the octet following the urgent data. This field can only be interpreted in segments for which the URG control bit has been set.
- Options
  - Options may be transmitted at the end of the TCP header and always have a length which is a multiple of 8 bits. All options are included in the checksum. An option may begin on any octet boundary.
  - There are two possible formats for an option:
  - A single octet of option type.
  - An octet of option type, an octet of option length, and the actual option data octets.
  - The option length includes the option type and option length, as well as the option data octets.
  - The list of options may be shorter than that designated by the data offset field because the contents of the header beyond the End-of-Option option must be header padding i.e., zero.
- A TCP must implement all options.
- Data
  - TCP data or higher layer protocol

TCP Transiction diagram

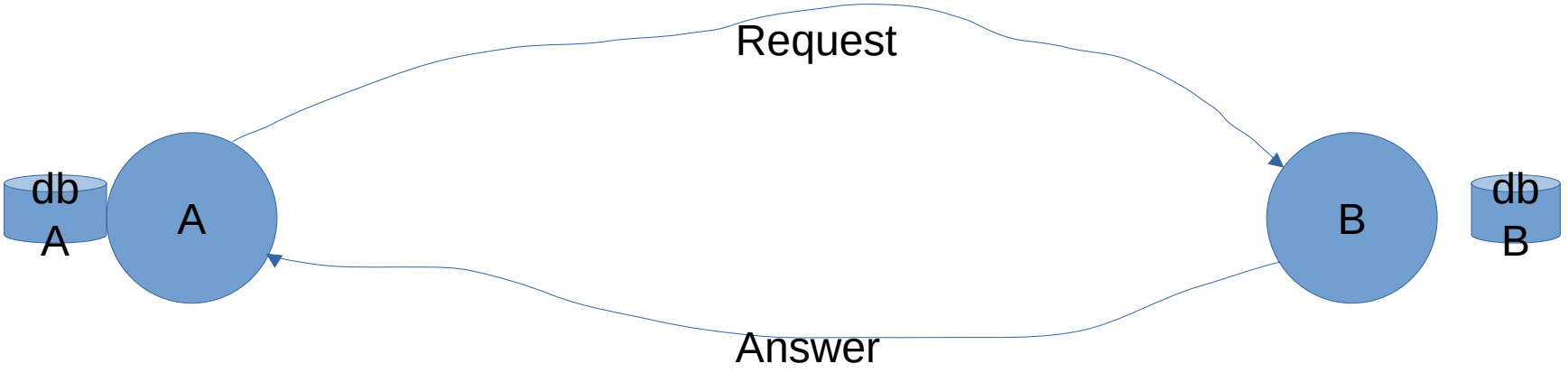


- A connection progresses through a series of states during its lifetime.
    - The states are: LISTEN, SYN-SENT, SYNRECEIVED,ESTABLISHED, FIN-WAIT-1, FIN-WAIT-2, CLOSE-WAIT, CLOSING, LAST-ACK, TIME WAIT, and the fictional state CLOSED. CLOSED is fictional because it represents the state when there is no TCB, and therefore, no connection. Briefly the meanings of the states are:

- LISTEN represents waiting for a connection request from any remote TCP and port.
- SYN-SENT represents waiting for a matching connection request after having sent a connection request.
- SYN-RECEIVED represents waiting for a confirming connection request acknowledgment after having both received and sent a connection request.
- ESTABLISHED represents an open connection, data received can be delivered to the user. The normal state for the data transfer phase of the connection.
- FIN-WAIT-1 represents waiting for a connection termination request from the remote TCP, or an acknowledgment of the connection termination request previously sent.
- FIN-WAIT-2 represents waiting for a connection termination request from the remote TCP.
- CLOSE-WAIT represents waiting for a connection termination request from the local user.
- CLOSING represents waiting for a connection termination request acknowledgment from the remote TCP.
- LAST-ACK represents waiting for an acknowledgment of the connection termination request previously sent to the remote TCP (which includes an acknowledgment of its connection termination request).
- TIME-WAIT represents waiting for enough time to pass to be sure the remote TCP received the acknowledgment of its connection termination request.
- CLOSED represents no connection state at all.

- A TCP connection progresses from one state to another in response to events. The events are:
- - the user calls
- - OPEN, SEND, RECEIVE, CLOSE, ABORT, and STATUS;
- - the incoming segments
- - particularly those containing the SYN, ACK, RST and FIN flags
- - the timeouts.

- The two transitions leading to the ESTABLISHED state correspond to the opening of a connection, and the two transitions leading from the ESTABLISHED state are for the termination of a connection. The ESTABLISHED state is where data transfer can occur between the two ends in both the directions.
- If a connection is in the LISTEN state and a SYN segment arrives, the connection makes a transition to the SYN_RCVD state and takes the action of replying with an ACK+SYN segment. The client does an active open which causes its end of the connection to send a SYN segment to the server and to move to the SYN_SENT state. The arrival of the SYN+ACK segment causes the client to mo ve to the ESTABLISHED state and to send an ack back to the server. When this ACK arrives the server finally moves to the ESTABLISHED state. In other words, we have just traced the THREE-WAY HANDSHAKE.
- In the process of terminating a connection, the important thing to kee p in mind is that the application process on both sides of the connection must independently close its half of the connection. Thus, on any one side there are three combinations of transition that get a connection from the ESTABLISHED state to the CLOSED state:
    - This side closes first:
- ESTABLISHED -> FIN_WAIT_1-> FIN_WAIT_2 -> TIME_WAIT -> CLOSED.
    - The other side closes first:
- ESTABLISHED -> CLOSE_WAIT -> LAST_ACK -> CLOSED.
    - Both sides close at the same time:
- ESTABLISHED -> FIN_WAIT_1-> CLOSING ->TIME_WAIT -> CLOSED.
- The main thing to recognize about connection teardown is that a connection in the TIME_WAIT state cannot move to the CLOSED state until it has waited for two times the maximum amount of time an IP datagram might live in the Inter net. The reason for this is that while the local side of the connection has sent an ACK in response to the other side's FIN segment, it does not know that the ACK was successfully delivered. As a consequence this other side might re transmit its FIN segment, and this second FIN segment might be delayed in the network. If the connection were allowed to move directly to the CLOSED

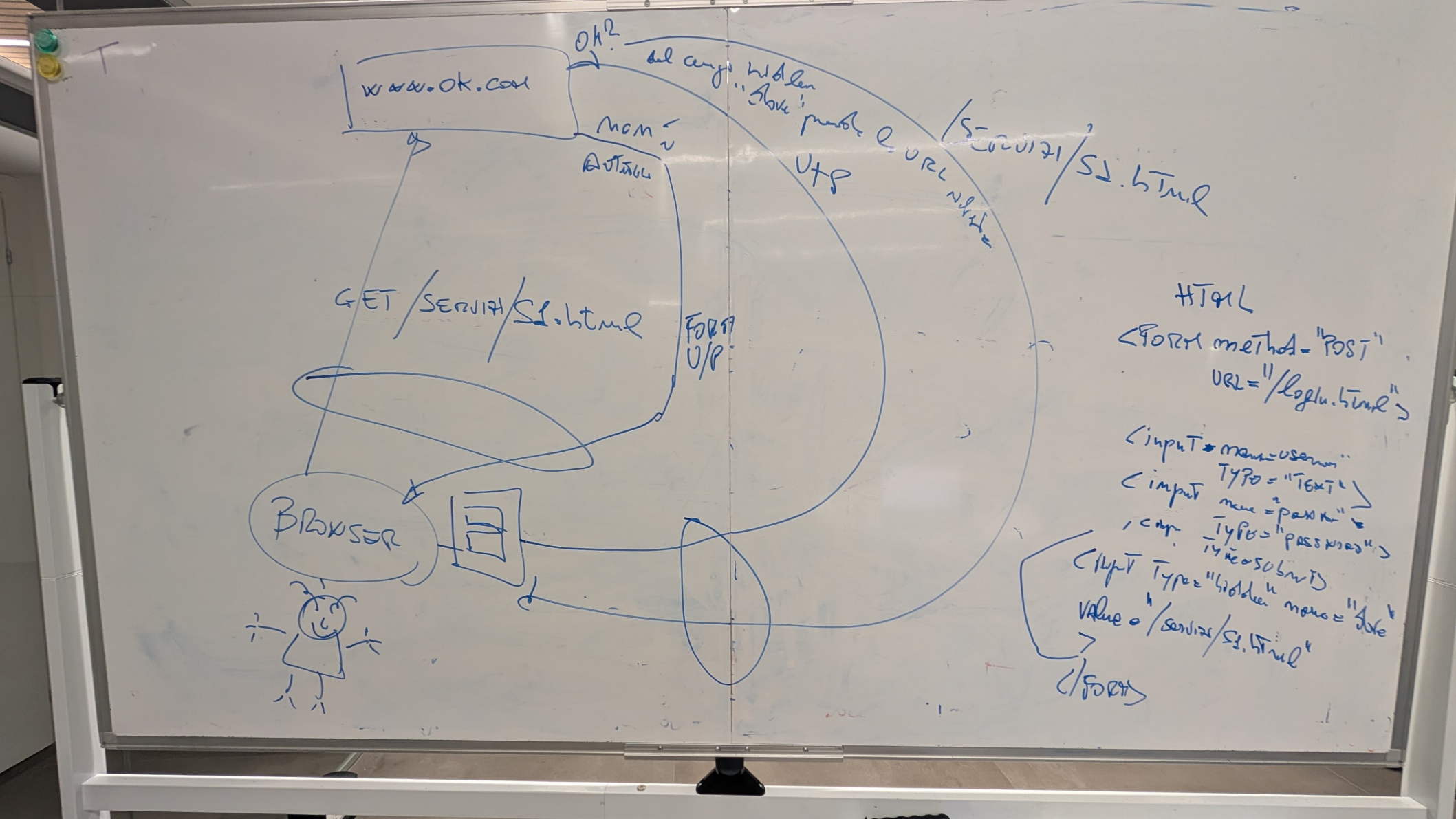


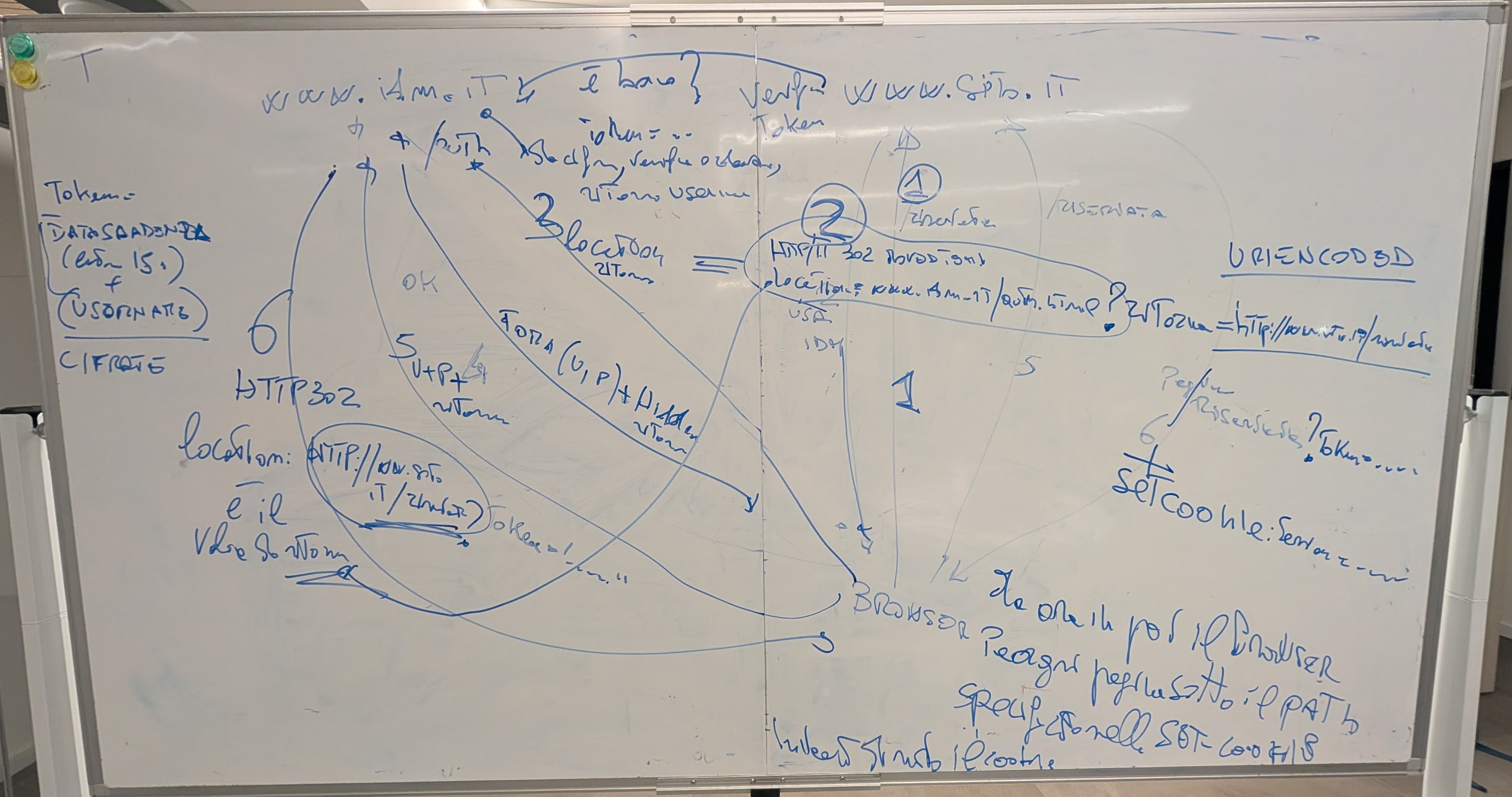
Modello di una transazione “ELEMENTARE”

A chiede a B qualcosa e non riceve risposta

Cosa è successo?

Rifacciamoci a quanto già conosciamo


www.ok.com
GET /servizi/s1.html
BROWSER
HTML
<FORM method="POST"
URL="/login.html">
<input name="username"
TYPE="TEXT">
<input name="password"
TYPE="PASSWORD">
<input TYPE="hidden" name="state"
value="/servizi/s1.html">
</FORM>


www.sito.it
Verifica Token
Token=
CIFRATE
HTTP 302
BROWSER
URLENCODED
SETCOOKIE:

- TCP protocol
  - Reliable
- Le comunicazioni TCP sono, in genere, affidabili nel senso che se A invia un “pacchetto” verso B, e se il pacchetto non raggiunge B, A, automaticamente, lo riinvia.
- Lo strato TCP “conta” i trasferimenti e per ogni invio da A verso B e da B verso A, segnala al suo peer (il layer responsabile dello scambio dati) sia ACK, sia SEQ
  - Quanti dati ho inviato finora, e quanti ne ho ricevuti
- In questo modo sono gestiti sia gli invii “ripetuti”, sia gli invii “non pervenuti”
  - In particolare, la macchina a stati finiti del TCP,ha un insieme di stati di WAIT che gestiscono il rinvio dei dati oppure la cancellazione della comunicazione
  - Nel senso che se A (layer TCP di A) ha inviato un pacchetto a B e dopo “N” millisecondi (possibilmente anche in una comunicazione successiva, dopo cioè lo scambio di altri dati) non ha ricevuto da B l’ACK relativa alla posizione dei byte inviati,allora A rinvia tutto o parte del pacchetto già inviato.
  - Questo modello è “riproponibile” nella transazione tra A e B?

Il protocollo HTTP

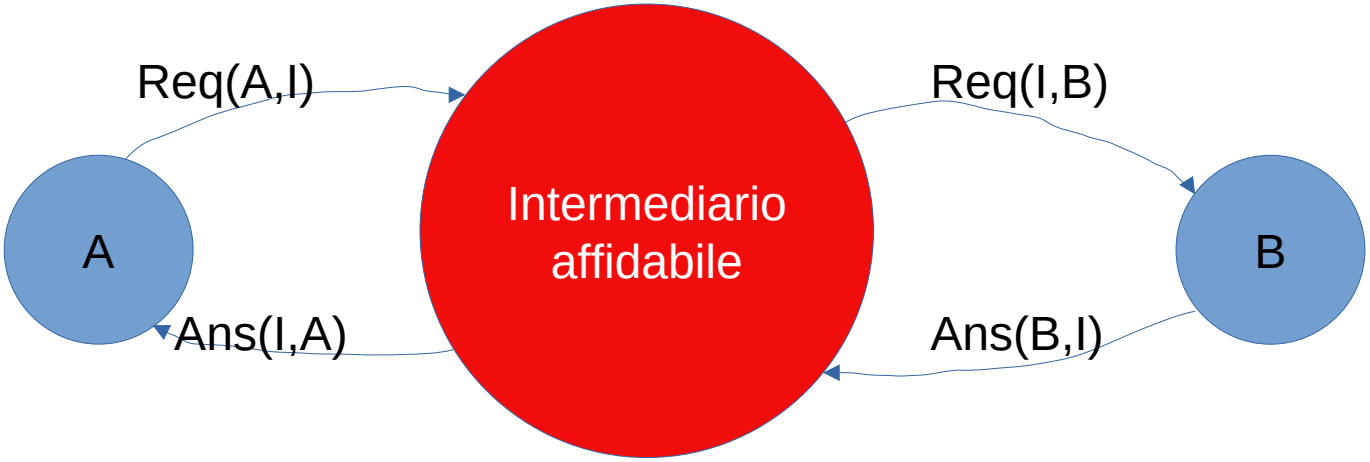
- Stateless
  - Il server NON memorizza informazioni sulla transazione
- Quali sono le soluzioni tecnico/organizzative utilizzate nel commercio on line per risolvere il problema della “mancata” ricezione della risposta?

A invia una richiesta a B

A non riceve risposta

Cosa è accaduto?

- 1) B non ha ricevuto la richiesta: ha impatto?
  - L’impatto è scarso poiché la transazione non è stata elaborata e quindi se A rinviasse la stessa richiesta non ci sarebbe la doppia spesa
- 2) B è “crashato”
- 3) B ha elaborato la richiesta, ha inviato la risposta che si “è fermata” sulla rete
  - L’impatto sulla transazione è elevatissimo: se A rinviasse la stessa richiesta, farebbe una doppia spesa.

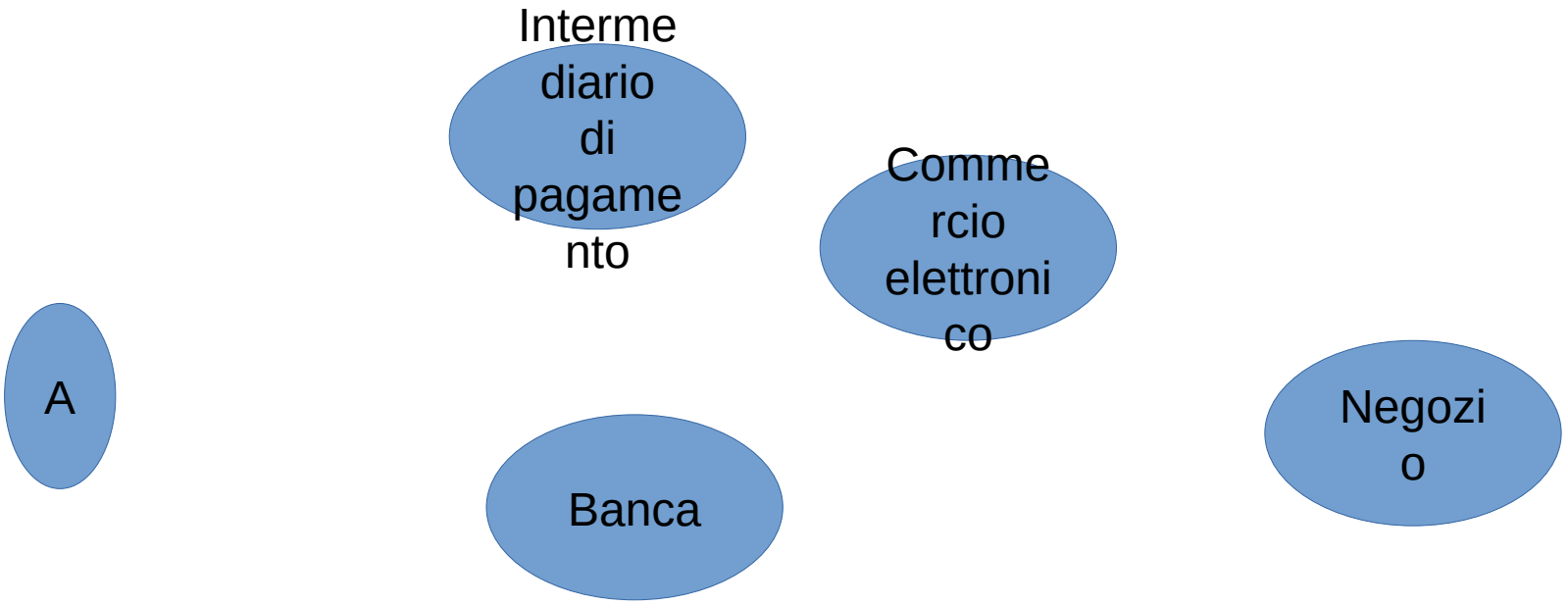


Si ricorre a un intermediario affidabile che si cura della ricezione delle richieste di A, le inoltra a B e viceversa

1) non risolve il problema, !!!!Lo Duplica!!!!

2) l'infrastruttura dell'intermediario affidabile dovrà essere dimensionata almeno come l'infrastruttura di B

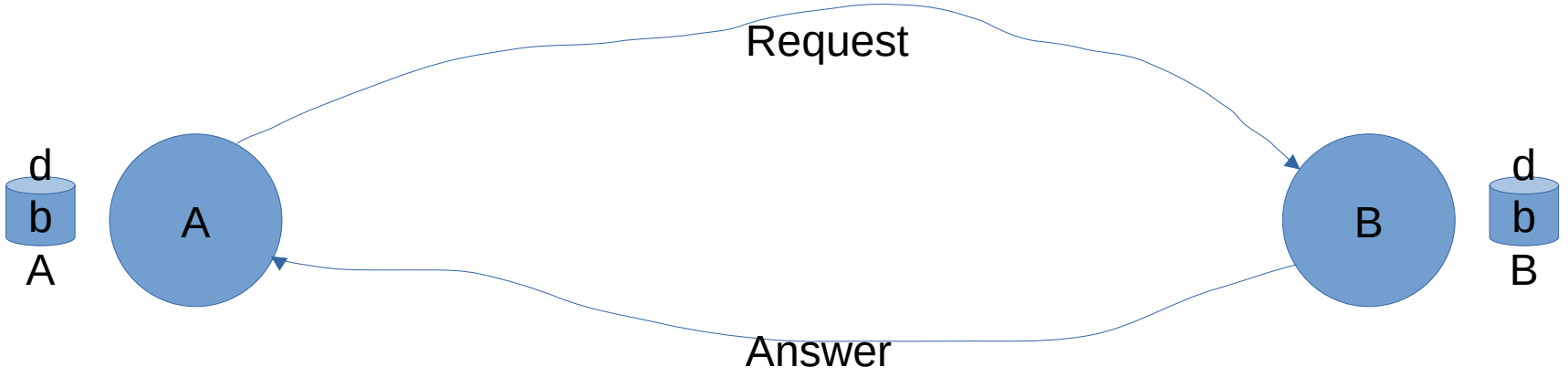
Nel caso reale del commercio on line la transazione è molto più complessa poiché aumentano gli attori coinvolti



La transazione diventa un “grafo” orientato => è descritta da un Partial order, da un workflow multiorganizzativo

Esercitazione

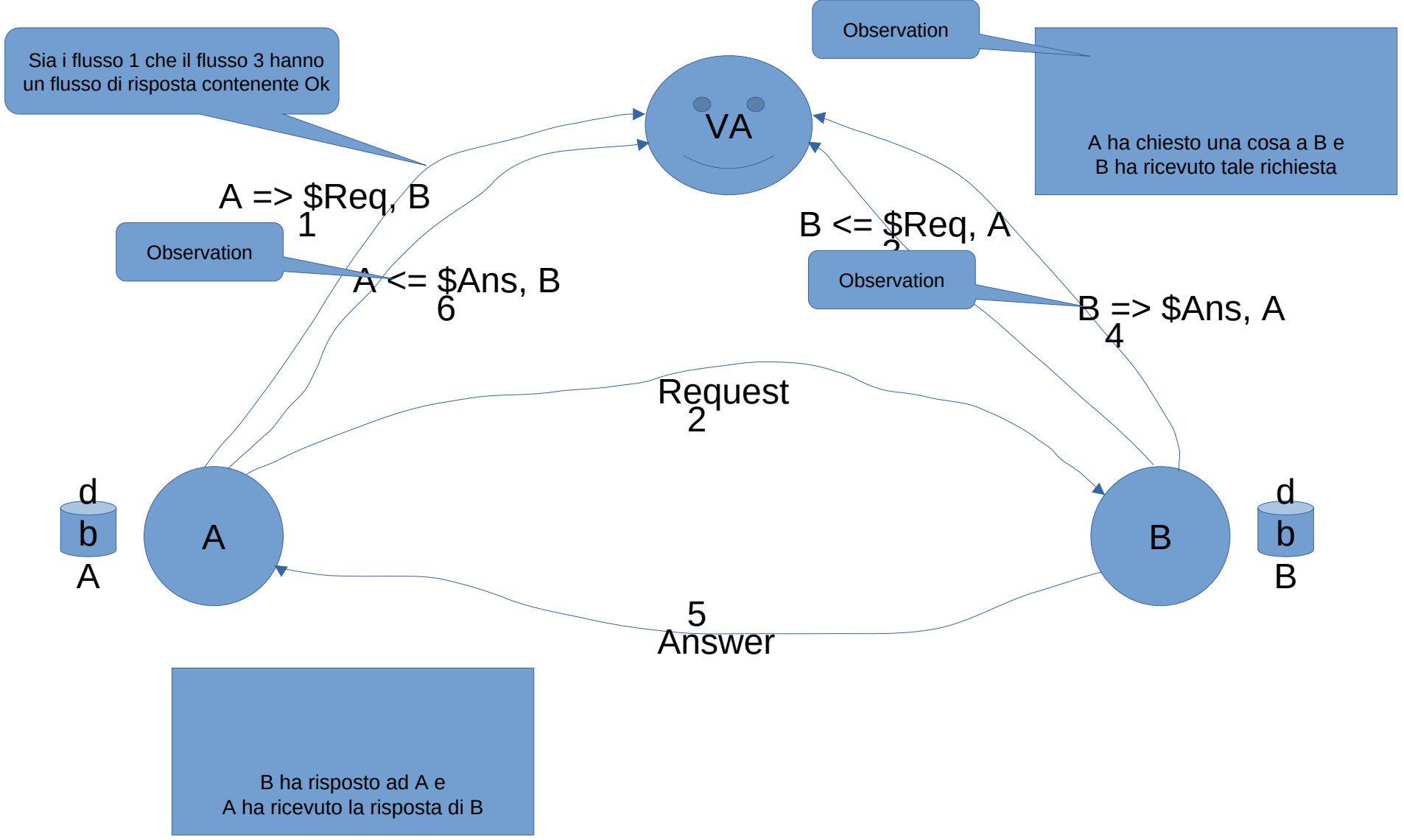
- Inviduare i casi reali di contenzioso sorti tra i clienti e i grandi siti di e-commerce



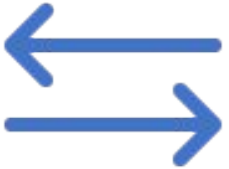
Modello di una transazione “ELEMENTARE”

A chiede a B qualcosa e non riceve risposta

Cosa è successo?


Sia i flusso 1 che il flusso 3 hanno un flusso di risposta contenente Ok
Observation
A ha chiesto una cosa a B e B ha ricevuto tale richiesta
VA
A => $Req, B
1
B <= $Req, A
3
Observation
A <= $Ans, B
6
Observation
B => $Ans, A
4
Request
2
d
b
A
A
B
B
d
b
B
5
Answer
B ha risposto ad A e A ha ricevuto la risposta di B

VA



Anche se A e/o B inviano più volta una stessa comunicazione alla VA, la vaildazione delle transazioni continua a essere corretta

????Ma non è che questo meccanismo di comunicare alla VA il fatto che A sta chiedendo qualcosa a B e viceversa, sia in violazione della Privacy????

Qual’è la domanda che ci si pone in caso di contenzioso

- Io (A) ho inviato al commerciante (B) una richiesta d’acquisto per 1 libro dal titolo: …..
- B mi ha fatto pagare ma non ho mai ricevuto il libro.
-
- É necessario che A invii alla VA la richiesta in chiaro? Se A inviasse HASH(richiesta)? E se B inviasse HASH(richiesta ricevuta)?, ecc.?

Per risolvere un contenzioso

È sufficiente che l’osservazione sia univocamente associabile alla richiesta!!

Esempio

A dice: Ho inviato il mio codice fiscale
VA dice: A ha inviato 2139741269DEF3
Se HASH(codice fiscale)=2139741269DEF3 allora c’è il non ripudio!

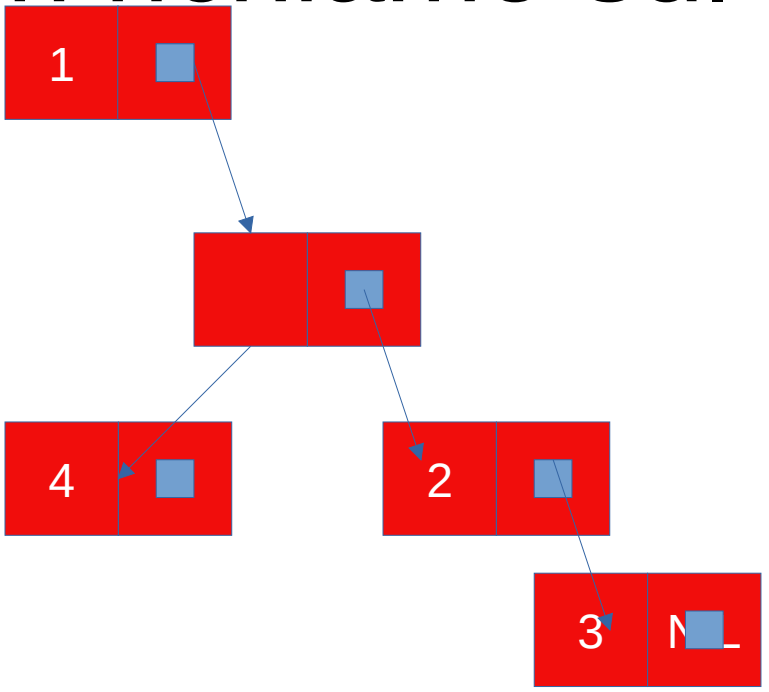
VA

Le osservazioni inviate alla VA	hanno dimensione (in byte) estremamente piccola rispetto alla comunicazione di richiesta/risposta che A invia a B e viceversa
Il dimensionamento della VA	è limitato dalle osservazioni e quindi la VA non ha la necessità di essere dimensionata come B (nel caso in cui B sia un server di commercio elettronico)
La VA riceve	comunicazioni molto brevi e l'unico lavoro che deve svolgere è quello di "accoppiare" tali comunicazioni al fine di ottenere "quaterne" di validazione delle transazioni.

Esercitazione

- Completare/arricchire la dispensa con le vostre deduzioni/analisi
- Definire una struttura di “observation” per la validazione di un sito che eroga servizi (commercio elettronico, P.A., Ente, Banca, Azienda, …), rispetto a uno o due servizi “importanti” che potrebbero portare a situazioni di contenzioso
- Suggerimento: l’applicabilità di un modello a terza parte come è la Validation Authority (VA) si estende anche alla gestione di anomalie di sicurezza (attacchi, pagine mai visitate, pagine troppo visitate, …).

Un richiamo sul linguaggio LISP



Una lista è una sequenza di conses

(1 (4) 2 3)

In una lista circolare l’ultimo cons punta al primo

Contenzioso/Litigation

Il sig. A e il sito di commercio elettronico E, hanno scelto la VA come soluzione per risolvere i casi di contenzioso

Hanno cioè firmato un “cooperation agreement”

Il sig. A dichiara di aver effettuato una transazione di C.E. con il sito E, acquistando un prodotto, ma di non avere ricevuto risposta da E, che comunque ha introitato il prezzo del prodotto.

Il sig. A chiede quindi il rimborso totale di quanto pagato.

La form di acquisto prodotti pubblicata da E, e approvata da tutti nel cooperation agreement, è la seguente

Campi: CF acquirente, ID prodotto, Quantità, Prezzo da pagare.

Le observation

Nel modello appena discusso, quali e come sono utilizzate come obervation? (Campi: CF acquirente, ID prodotto, Quantità, Prezzo da pagare)

CF, ID, Qty, Price

Cosa richiedereste alla VA voi per risolvere il contenzioso?

Chiedo alla VA
- A ha effettivamente inviato la richiesta di Acquisto?
- E ha effettivamente ricevuto la richiesta di Acquisto?
- E ha effettivamente confermato la richiesta di Acquisto?
- A ha effettivamente ricevuto la conferma di Acquisto?

Privacy

Come fa la VA e confermare i quattro punti precedenti?

E Come fa la VA a dimostrare che questi fanno riferimento ad A, E e al prodotto?

- La domanda del giudice è:
  - Mi dici lo stato della transazione tra il sig. A il cui CF è XXXXX, il sito di CE E, in relazione al prodotto identificato con codice ID, di quantità Q e di prezzo totale Tot.

In ogni caso non è necessario che le informazioni richieste siano “in chiaro”, né quelle poste dal Giudice, né quelle archiviate nella VA.

Se tutte le informazioni sono archiviate come HASH e se le richieste del giudice sono poste direttamente in formato HASH, la VA può in ogni caso confermare l’esito/lo stato della transazione economica.

Esercizio

- In relazione a un sito di C.E. o di fornitura di servizi a voi noto
  - Individuare gli elementi della comunicazione che consentono di Validare una transazione
    - NB: al giorno d'oggi, il protocollo più utilizzato per le transazioni elettroniche è HTTP con HTML e con i servizi erogati in REST JSON/POST(body/uri) o similari XML/SOAP.
    - NB: se dovete validare una transazione elettronica, non è detto che siate interessati alle componenti CSS o immagine o HTML delle pagine di servizio
  - Individuare impatto della VA nelle pagine del sito (nessun impatto impatto minimo, modifica totale del sito, ...)

Autenticazione

- Le tre A che poi sono diventate quattro
- Autenticazione
  - Qualcosa che io conosco!!!! La più diffusa!!!!
  - Qualcosa che mi appartiene (biometria) => dovrebbe garantire la autenticazione “in presenza”
  - Qualcosa che solo io posseggo (otp, microchip, ...)
- Autorizzazione
  - Privilegi di accesso alle risorse
  - L’autorizzazione consente di separare Autenticazione da Autorizzazione, sono in genere due sistemi informativi diversi
- Accounting
  - Uso delle risorse
- Audit
  - Tracciatura dei servizi (!!!Non sono i log dei web server [apache, nginx, iis, ---], Non sono i log applicativi [tomcat, REST/JSON, gli application server!!!)
  - La tracciatura dei servizi DEVE essere un fatto “oggettivo”, cioè indipendente dal sistema di servizi stesso.
  - Nei fatti deve essere in analogia alle reti, un layer a sé stante

Autenticazione a terza parte

- Sistema di autenticazione
  - Www.auth.com
- Sistema di servizi
  - Www.services.com
- Utente
  - Usa un browser per accedere a un servizio/risorsa “protetto” e deve autenticarsi presso www.auth.com per poter poi accedere alla specifica pagina di www.services.com

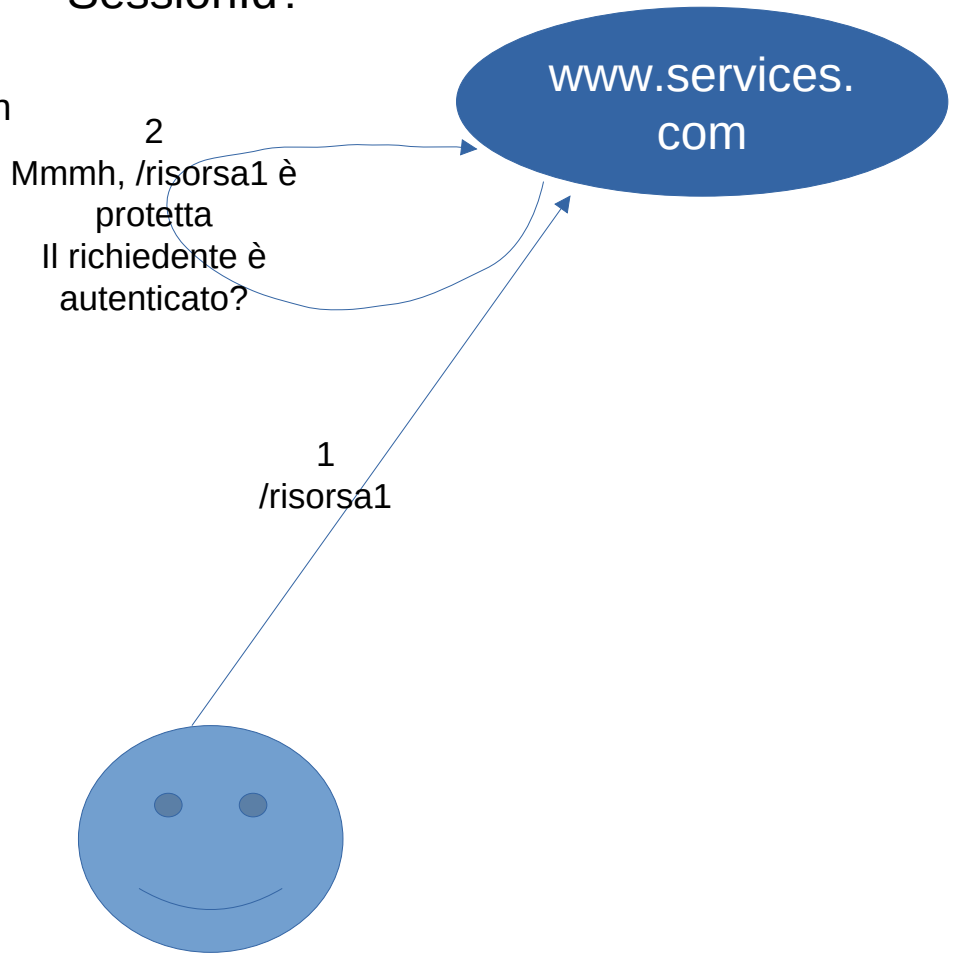
cookie

Se il server (www.server.com) invia al browser un cookie associato al suo dominio di riferimento (www.server.com/url…)	• Allora il browser quando e solo quando richiederà l'uso di una risorsa individuata dalla stessa uri (o da una uri sottostante) assocerà alla richiesta il cookie ricevuto
Dove trovo nella comunicazione HTTP i set dei cookie e gli invii dei cookie?	• negli header del protocollo HTTP
Richiesta http	• Riga metodo e uri • Righe header…. • Riga vuota • Eventuale parte dati (body)
Risposta http	• Esito richiesta (es: HTTP/1.1 200 OK) • Righe di header • Riga vuota • Eventuale parte dati

Cookie?
SessionId?

- Se l’utente non è autenticato
  - SITO: gli mostro direttamente (gli rispondo) con la maschera di autenticazione (login)
  - USER: fill la maschera (inserisce username e password) e poi “click” sul bottone di login/accesso
  - BROWSER: invia (GET/POST, …) le credenziali al SITO su una specifica pagina

  - SITO: legge quanto ricevuto (username e password) verifica se presenti in archivio utenti (dove ci sono almeno username e password). Se presente che fa?
  - SITO: ad esempio decide di utilizzare i cookie
    - tramite la set-cookie trasferisce al browser una coppia key-value, con associati: dominio di riferimento (potrebbe essere la parte uri protetta, e durata

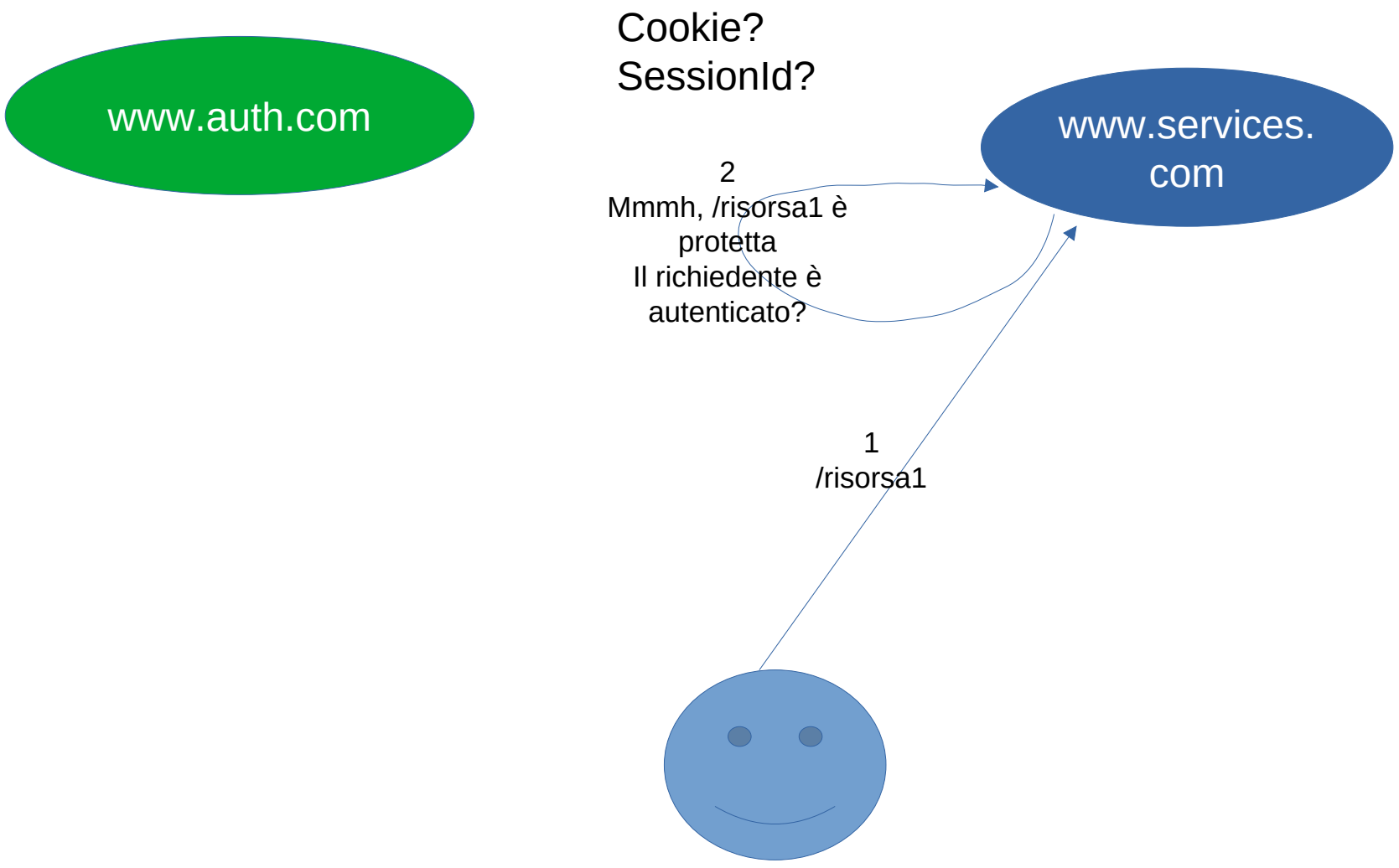


E se non poteste utilizzare i cookie? Ricordate che http è stateless!!!

- Invio ogni volta le mie credenziali
  - https://username:password@www.example.com/path
- Costruisco una uri “virtuale” per l’utente che si è autenticato e quindi lui accederà sempre ad uri del tipo , http://www.example.com/urivirtualeassociataallutenteautenticato/uri del sito
- Significa che nel gestore del sito ho una “map” che associa la uri virtuale all’utente.
- Per ogni richiesta che arriva verifico che la uri virtuale sia presente nella map e, se presente, tolgo la uri virtuale dalla richiesta e fornisco la risorsa reale richiesta
  - Inietto nella pagina un pezzo di codice javascript (oppure progetto le pagine direttamente con questo pezzo di codice) che aggiunge in fondo alla uri di ogni richiesta un codice unico. Esempio di richiesta modificata:
  - GET serviziopagina1.html?codiceunico=Xxxxxxxxx
  - Dove il valore XXXXXXXX diventa chiave della solita MAP che mi consente di verificare che l’utente abbia superato il login!!!

Esercitazione

- Arricchire e motivare le slide
- Fornire una motivazione che spieghi per quale motivo l’uso di una variabile globale per archiviare il token di sessione dell’utente (ricavata a fronte dell’atutenticazione e corrispondente a un cookie, a una uri virtuale, a una variabile k/v di sessione, …) è Errato!
- Fornire una soluzione alternativa


www.auth.com
Cookie?
SessionId?
www.services.
com
2
Mmmh, /risorsa1 è
protetta
Il richiedente è
autenticato?
1
/risorsa1

Il flusso di autenticazione (username/password) tra A (browser) e B(un web server) basato sui cookie

- A invia a B richiesta di accesso a risorsa protetta
  - GET /risorsa_01 …
- B riceve la richiesta e riconosce che il richiedente deve prima autenticarsi. B risponde con la form di autenticazione
- A riceve la form di autenticazione, a compila e la invia a B
- B riceve la form compilata, verifica/autentica le credenziali e risponde con la pagina /risorsa_01, precedentemente richiesta da A
- B deve organizzarsi per “ricordare” che A è stato autenticato. Darà un cookie ad A
- Struttura del cookie
  - Nome: session (indifferente)
  - Contenuto: il server crea un id della sessione e lo scrive qui.
  - Come e cosa per l’id sessione?
- B crea un codice=numero seriale (da un DB, UUID, random strong) e lo chiama id sessione
- Invia l’header: Set-Cookie: session=<codice>; (se non metto la durata, allora è un session cookie)
- Salva il <codice> in un archivio CDB, associandolo al nome utente (per le autorizzazioni)
- Durata: 15/20 minuti oppure “finché il browser è attivo” (session cookie che vanno, nominalmente, solo in ram)
- A riceve la pagina /risorsa_01 e riceve anche il cookie e prosegue nella navigazione protetta chiedendo la pagina /risorsa_02
- Nella richiesta A invierà l’header: Cookie: session=<codice>; …
- B riceve la richiesta di risorsa protetta /risorsa_02, con un cookie session associato e verifica che il cookie sia ok
- Due cose
- Autenticazione: verifica il <codice> rispetto all’archivio dei cookie CDB
- Autorizzazione

La precedente proposta viola un principio base

- Il protocollo HTTP e quindi il WEB è
  - Stateless
- Archiviare il numero di serie (il valore del cookie) in un DB viola questo principio e rende il sistema attaccabile per DOS

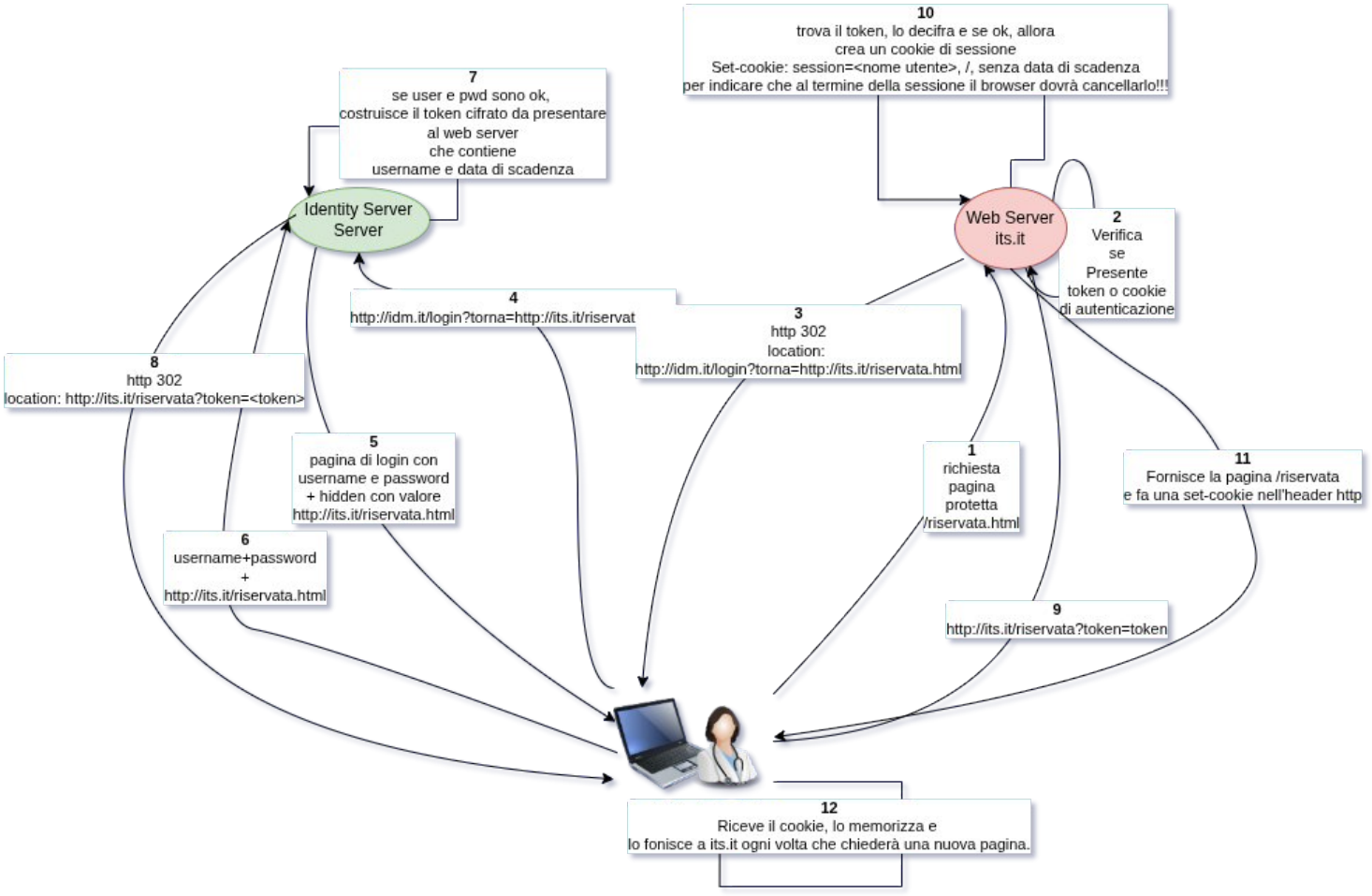
Il flusso di autenticazione (username/password) tra A (browser) e B(un web server) basato sui cookie

- A invia a B richiesta di accesso a risorsa protetta
  - GET /risorsa_01 …
- B riceve la richiesta e riconosce che il richiedente deve prima autenticarsi. B risponde con la form di autenticazione, dove ha inserito in un campo HIDDEN la ricorsa /risorsa_01 che era stata richiesta
- A riceve la form di autenticazione, a compila e la invia a B, che riceverà sia username e password, sia il campo hidden
- B riceve la form compilata, verifica/autentica le credenziali e risponde con la pagina /risorsa_01, precedentemente richiesta da A
  - B deve organizzarsi per “ricordare” che A è stato autenticato. Darà un cookie ad A
  - Struttura del cookie
    - Nome: session (indifferente)
    - Contenuto: il server crea un id della sessione e lo scrive qui.
      - Come e cosa per l’id sessione?
        - B crea un codice=AES256(username+time, miachiave), quindi potrei usare HMAC ma devo in ogni caso essere in grado di recuperare la username da questa informazione.
        - Invio l’header: Set-Cookie: session=<codice>; (se non metto la durata, allora è un session cookie)
    - Durata: 15/20 minuti oppure “finché il browser è attivo” (session cookie che vanno, nominalmente, solo in ram)
    - Prendendo i dati HIDDEN, B invia la pagina corretta a A
- A riceve la pagina /risorsa_01 e riceve anche il cookie e prosegue nella navigazione protetta chiedendo la pagina /risorsa_02
  - Nella richiesta A invierà l’header: Cookie: session=<codice>; …
- B riceve la richiesta di risorsa protetta /risorsa_02, con un cookie session associato e verifica che il cookie sia ok
  - Due cose
    - Autenticazione: verifica il <codice> rispetto all’archivio delle autorizzazioni !!! NB che l’archivio degli utenti DEVE esistere!

A: browser, B: web server, B1: altro web server, C: identity manager

- A chiede /ris_01
- B verifica che A non è autenticato e dice ad A che si deve autenticare su C: risponde con HTTP/1.1 302 Found
  Location: http://C/login?dovedevitornare=B&Suqualepagina=/ris_01
- A riceve la redirect e, automaticamente va sul sito:
  - http://C/login?dovedevitornare=B&Suqualepagina=/ris_01
- C riceve la richiesta e fornisce la FORM di autenticazione, nel form C può inserire due campi hidden che sono le variabili passate da B (dovedevitornare, suqualepagina)
- A riempie la form e la invia a C (con i due campi hidden)
- C autentica e rimanda A a B
  - A deve portarsi un token fornito da C e deve darlo a B
  - HTTP/1.1 302 Found
    Location: http://C/login?dovedevitornare=B&Suqualepagina=/ris_01&token=<qualcosa>

I due server condividono una chiave di cifra per cifrare (idm) e decifrare (its) il token di autenticazione costruito allo step 7



Esercitazione

- Nello schema precedente sono assenti due elementi
  - Formulazione del token
  - Verifica da parte di B che A sia stato correttamente autenticato da C

10K problem

- Realizzare un server in grado di gestire almeno 10K (oggi 1M/10M) transazioni al secondo
- Protocollo HTTP
- Question
  - Dovete realizzare un server web. Come layer di base dovrete realizzare un socket server TCP/IP. Che tecnologia/protocollo utilizzereste?
  - Il requisito è la gestione di 10K (o N) comunicazioni WEB “in parallelo” per ogni secondo”

Tecnologie per i socket server

- In genere si parte sempre con
  - S = socket(…);
  - bind(s, scheda di rete/indirizzo ip)
  - listen(s); //metto il socket in coda di ascolto
  - While true
  - {
    - news = accept(s)
    - Ora news è il socket dal quale potete leggere e scrivere
    - ManageTransaction(news);
      - While (not eof(news))
      - {
        - Richiesta = read(s);
        - Risposta = Elabora Richiesta
        - write(s, risposta);
      - }
  - }

Il problema dei socket

- READ
  - char Buffer [1024*1024];
  - int letti = read(s, buffer, 1024*1024)
    - ???Letti è pari a 2^20???
- WRITE
  - int scritti=write(s, buffer, 45345);
  - ???Scritti è pari a 45345???

Esempio WEB

- Il browser, dopo essersi collegato al server sul socket s, gli invia:
  - char buffer[1000000]
  - strcpy(buffer, “GET /pagina1.html HTTP/1.1\r\nHost: www.sdc2021 .com\r\nContent-length: 10\r\nConnection: keep-alive\r\n\r\n1234567890”;
  - int scritti = write(s, buffer, strlen(buffer));
  - Console.log(“Ho inviato: “, scritti, “ bytes”);
- Il server, che sta ricevendo dal client (dopo la accept), sul socket s
  - char buffer[1000000];
  - int letti = recv(soc, buffer, 1000000);
  - Console.log(“Ho ricevuto: “, letti, “ bytes”);
  - Quanto vale letti quando la recv termina (ritorna al chiamante)?

Socket sincroni vs asincroni

- Sincrono (I/O bloccante)
  - Write: torna solo dopo aver inviato almeno 1 byte
  - Read: torna solo dopo aver ricevuto almeno 1 byte
- Asincrono
  - Write: torna immediatamente, potrebbe non aver inviato nulla
  - Read: torna immediatamente, potrebbe non aver ricevuto nulla

Task

- In genere si parte sempre con
  - S = socket(…);
  - bind(s, scheda di rete/indirizzo ip)
  - listen(s); //metto il socket in coda di ascolto
  - While true
  - {
    - news = accept(s)
    - Ora news è il socket dal quale potete leggere e scrivere
    - Int i = fork();
    - If (i==0)
      - ManageTransaction(news)
  - }

task

- Task switching lento, costoso
- Max 2000/3000 task in una macchina media

- Esempio
  - APACHE HTTPD utilizza in una configurazione il task switching
  - Oracle DB, utilizza in una configurazione, il task switching

thread

- Thread switching è più efficiente
- Posso pensare di gestire 10000/15000 thread in contemporanea
  - S = socket(…);
  - bind(s, scheda di rete/indirizzo ip)
  - listen(s); //metto il socket in coda di ascolto
  - While true
  - {
    - news = accept(s)
    - Ora news è il socket dal quale potete leggere e scrivere
    - thread th = std::thread(ManageTransaction, news);
    - th.detach();
  - }

fibers/coroutines, epoll, ...

- Epoll
  - When ready to read s call f1(s)
  - When ready to write s call f2(s)
  - When error s call f3(s)
  - …
  - Nel kernel si formano tavole che contengono
    - Socket, callback associate al socket (read, write, error, connect, accept)

sleep

- Come si fa una sleep in javascript??
- Function sleep(secondi){
  - SetTimeout(<???>, secondi*1000);
- }

- Voglio una cosa di questo tipo
- Console.log(time);
- Sleep(3);
- Console.log(time);
  - Risultato
    - 12345
    - 12348

- Come si fa una sleep in c++?
- Function MySleep(int sec)
- {
  - sleep(sec);
  - Return;
- }
- Console.log(time);
- MySleep(3);
- Console.log(time);
  - Risultato
    - 12345
    - 12348

Coroutines/Fibers

- (define (IsEven N)
  - (if (zero? N)
    - #t
    - (IsOdd (1- N))))
- (define (IsOdd N)
  - (if (= N 1)
    - #t
    - (IsEven (1- N))))

- Complessità: O(N)
- Spazio: O(N) => stack !!!!

coroutine

- Func A (…, wtc)
  - ...
  - Call B(param, where to continue A)!!come se gli passasse ilPcounter+il contesto di esecuzione
  - …
- Func B(…, wtc)
  - …
  - Call A(param, where to continue B)
  - ...

Continuation

- (+ 10 20)
  - (lambda (x) (+ 10 x))

Closure

- Esempio C++
  - int a=10;
  - auto fun = [a] (int n) → int {return n+a;};
  - //in go a:= 10; => implica che a diventa intero
  - //in c++: auto a = 10; //a è int

Closure

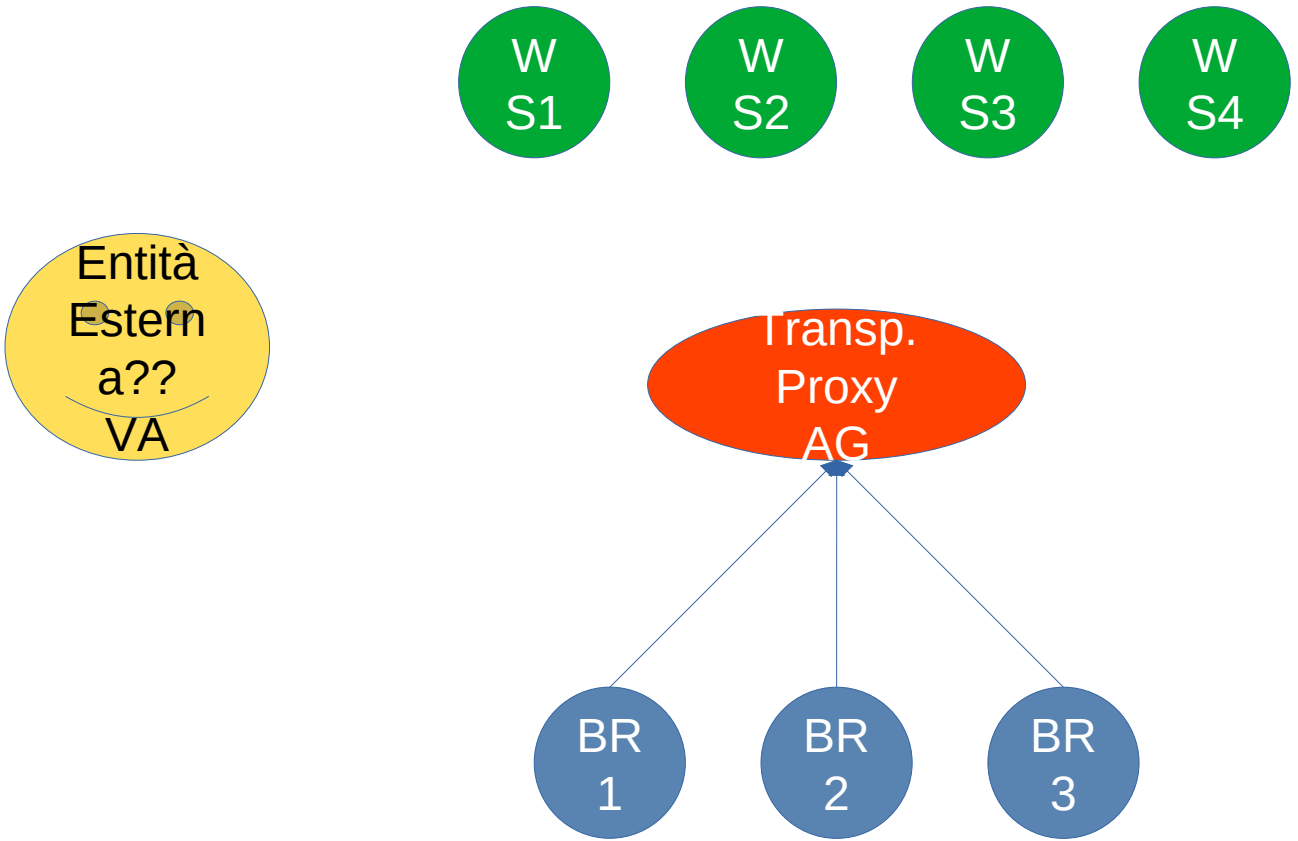
- Javascript

- function A(p1, p2) {
  - SetTimeout(() => {console.log(p1);}, 10000);
  - return false;
- }
- …
- A(10, 20); //?? dopo 10 secondi sulla console appare il valore 10!!!
- A(30, 40)
- B(….)

Closure in Scheme

- (define (Counter)
  - (let ((n 0)(p 100)(a “ciao”)
    - (lambda(x)
      - (set! n (+ n x)) n)))

Cooperazione



Modello di cooperazione

- I browser (BR1, BR2, BR3), tramite il transparent proxy (AG) accedono a servizi di alcuni web server (WS1, WS2, WS3, WS4).
- In un contesto di assoluta anonimità, in cui quindi non sono interessato a far autenticare le sessioni di lavoro svolte dai vari browser sui web server e nemmeno a localizzare/identificare in modo assoluto i browser stessi
  - Come posso monitorare e ricomporre le sessioni di lavoro svolte da un browser nei confronti di uno o più server?
    - Esempio: bs1 si collega a ws1 e naviga su alcune pagine. Poi si collega a ws2, poi a ws3 e poi torna su ws1
      - Il sistema di tracciatura deve potermi documentare che un browser (non importa la sua localizzazione) ha compiuto tali operazioni
- !!quando BR1 si collega a WS1 usa il nome di dominio di WS1. Il proxy trasparente è “trasparente”, quindi le comunicazioni passano per lui ma i browser non ne conoscono l’esistenza!!

Esempio di sessione multiorganizzazioni

- Br1 va su ws1
  - AG, che è in the middle, verifica se br1 ha inviato un cookie chiamato vaws1. Se non lo ha allora fa una redirect to VA fornendo la pagina richiesta da br1 e il dominio ws1
  - BR1 va da VA (in automatico)
    - VA verifica se BR1 ha un cookie fornito da VA. Se non lo a, gli da un cookie CVA=<unique value> e lo redirige di nuovo verso WS1 aggiungendo ai parametri della URI la coppia VA=<unique value> es: http://ws1/pagina?VA=<unique id>
  - BR1 va verso WS1, con queste nuove informazioni, e AG riconosce che nella “query” della URI c’è un parametro VA.
    - AG fa una nuova redirect su WS1, settando il cookie vaws1=<unique id> e passandogli come location la stessa e quindi ws1 e pagina
  - BR1 torna verso WS1 ma stavolta ha il cookie vaws1=<unique id>

Esempio di sessione multiorganizzazioni

- Ora BR1 va su ws2
  - AG verifica l’esistena di un cookie vaws2. Se non c’è lo rigira alla VA
- BR1 si presenta alla VA con il cookie CVA=<unique value>
  - VA fa una redirect passando come parametro nella query VA=<unique value>
- AG riconosce il CVA e fornisce (tramite redirect) a BR1 un cookie vaws2=<lo stesso unique id>
- ...

Esempio di sessione multiorganizzazioni

- Al termine AG riesce a tracciare le operazioni compiute da un browser poiché gli sono stati assegnati dei cookie per i vari server visitati ma tutti con lo stesso unique id
  - Quindi al termine ag potrà certificare: un browser (rappresentato in una sessione da uno specifico unique-id), ha navigato nel seguente modo: …

Esercitazione

- Descrivere il modello di tracciatura anonima qui presentato
  - In termini di implementazione (tramite una descrizione dei flussi)
  - In termini di potenziali vantaggi di utilizzo

Esercitazione

- Descrivere il modello di tracciatura anonima qui presentato
  - In termini di implementazione (tramite una descrizione dei flussi)
  - In termini di potenziali vantaggi di utilizzo

Il sistema delle Macroistruzioni

- Esempio in C/C++
  - #if, #ifdef, #ifndef, #define
  - Esclusione dalla compilazione
    - #if 0
    - Int a=20;
    - Printf(“%d\n”, a);
    - #endif
    - Le due istruzioni non sono compilate

Il sistema delle Macroistruzioni, sempre in C/C++

- #define MAX 100
- #ifdef MAX
  - Int vettore[MAX];
- #else
  - Int vettore[100];
- #endif
- !!!In C non è consentito scrivere
  - Int n=100;
  - Char cvet[n];
- #define max3(a,b,c) if (a > b) if (a > c) a if ((b > a)&&(b > c)) b; else c;
- …
- Int a=10; int b=20; int c= 30;
- Int max = max3(a,b,c);

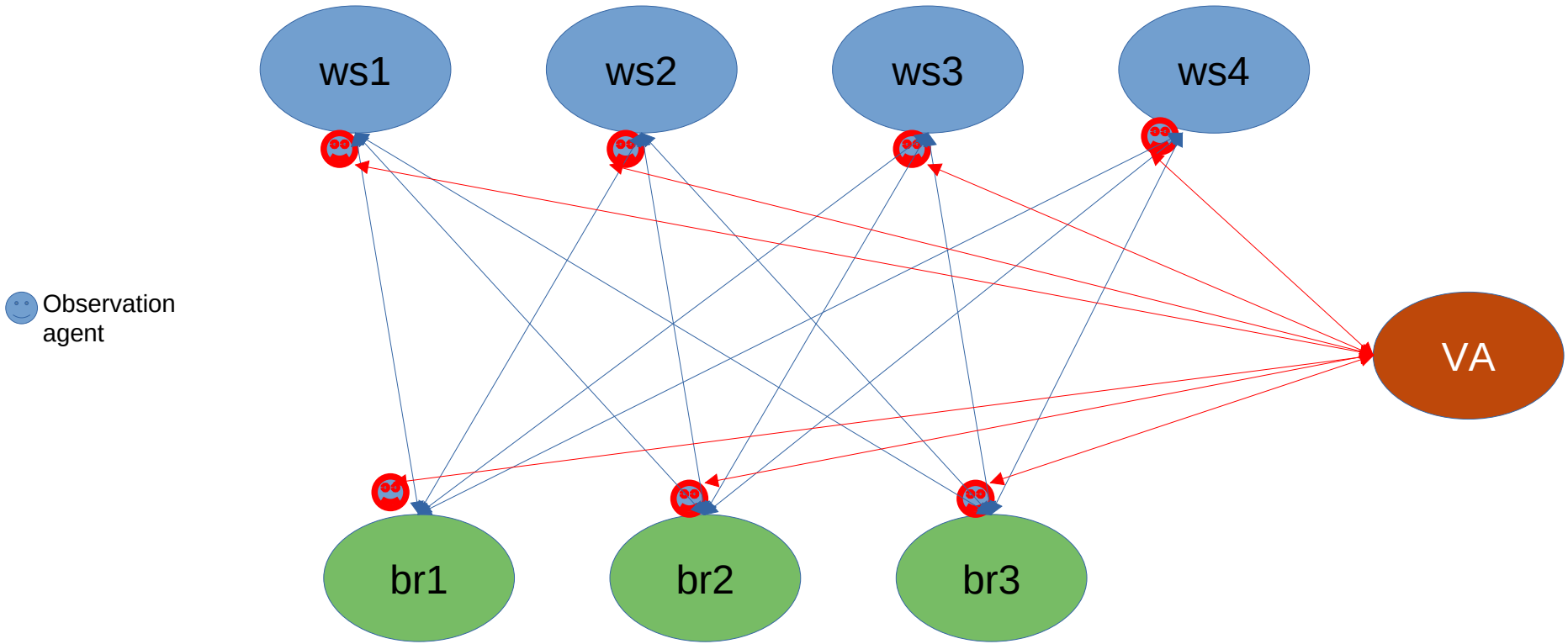
Nel linguaggio che preferite, costruite un nuovo operatore

- Operatore: when-not
- Sintassi: (when-not <espressione logica> <sequenza di istruzioni>)
  - ESEMPIO: function When-not (espressionelogica, let a=20; console.log(a); settimeout(….);!!!!difficile
  - Gli operatori non valutano i loro parametri ma creano un blocco di codice che sarà eseguito a tempo di esecuzione
- Manuale Guile da pag. 257 in poi: define-syntax

Ancora define-syntax

- Intendo scrivere un codice in un linguaggio di programmazione tale che
  - Incrementa di N (parametro) il valore della variabile passata
    - NB: la variabile DEVE cambiare valore nel programma chiamante, non nel programma chiamato
- Esempio (javascript)
  - Let a 100;
  - Aggiungi(a, 20);
  - Console.log(a) => 120
- !!Gli altri esempi sul doc: esempio.scm

Un’architettura generale per la cooperazione



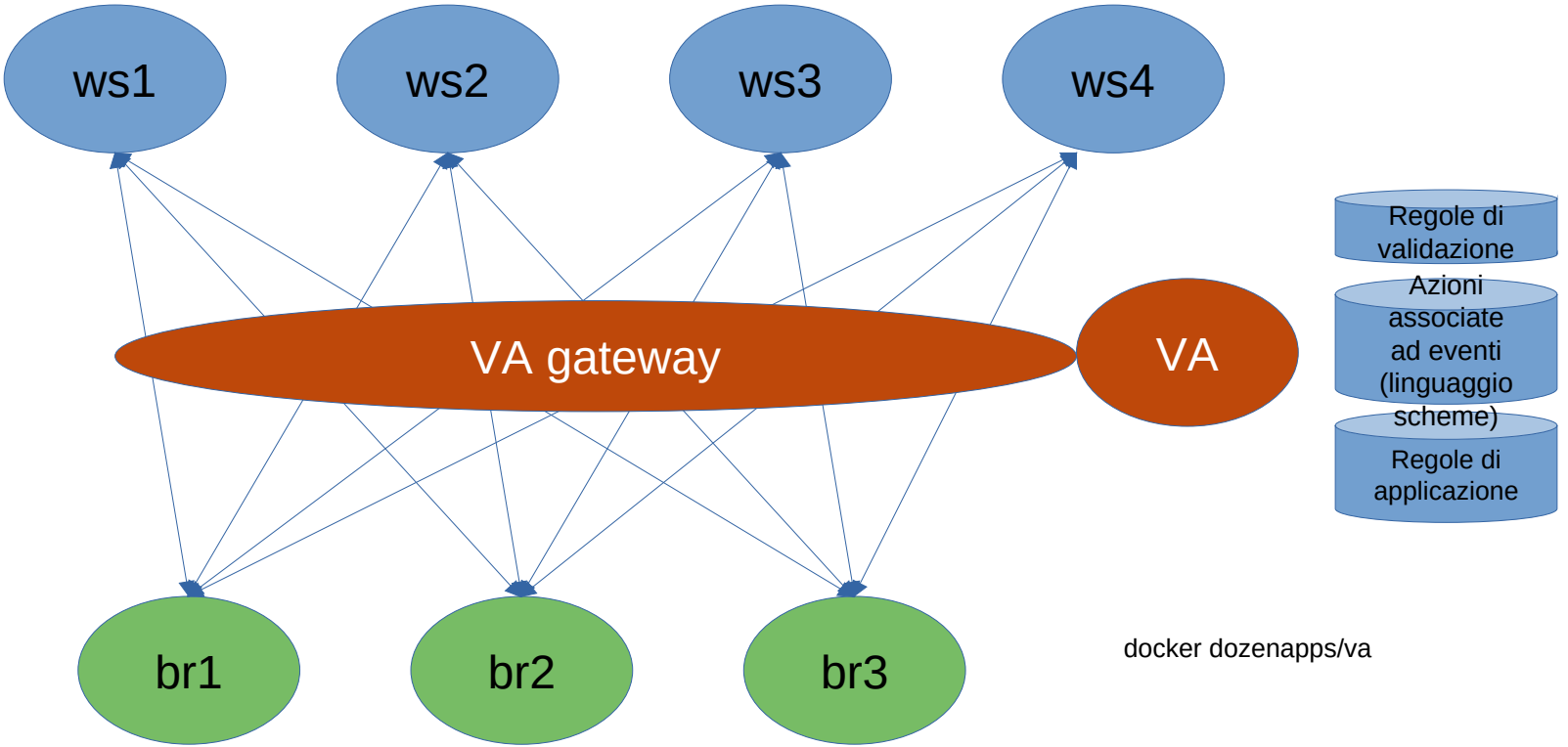
Fare un’analisi di sicurezza e monitoraggio del file Nasa

- Utilizzare Python per generare i report
- Pagine, dimensioni, frequenza, ...

- Gli agenti di osservazione operano direttamente sullo strato di comunicazione, estraggono le osservazioni e le inviano alla VA
- Come possono essere realizzati?
  - Trasparenti
    - Intrusivi
      - Tramite una soluzione IPTABLES (linux-livello 2)/Layered Service Provider(Windows-liv 3) posso introdurre un proxy trasparente in locale che si occupa di leggere dai pacchetti i dati di interesse e di inviarli alla VA. In questo caso il proxy locale può anche “fermare” le comunicazioni in attesa della risposta della VA.
        - Nel caso di HTTPS richiedono un MITM a livello SSL (e quindi la creazione dinamica di certificati SSL)
      - Browser extension
        - Nessun problema di HTTPS (poiché sta sul browser). Oggi problemi di privacy poiché si inizia a non accettare le browser extensions.
    - Non intrusivi
      - Ad esempio: basato su uno sniffer di rete. Problemi?
        - Comunicazioni Deframmentate
        - Più comunicazioni in parallelo (anche quelle che non ci interessano)
        - Assenza di un ordinamento temporale dei pacchetti
        - !!!!impossibile!!! leggere pacchetti HTTPS!
  - Non trasparenti
    - Nei quali l’applicazione/i sistemi sono “aware” relativamente agli agenti e alla VA

Modello semplificato

- Non è riduttivo, è semplificato rispetto alle complessità architetturali degli agenti



Application rules (in serie/in tempo reale)

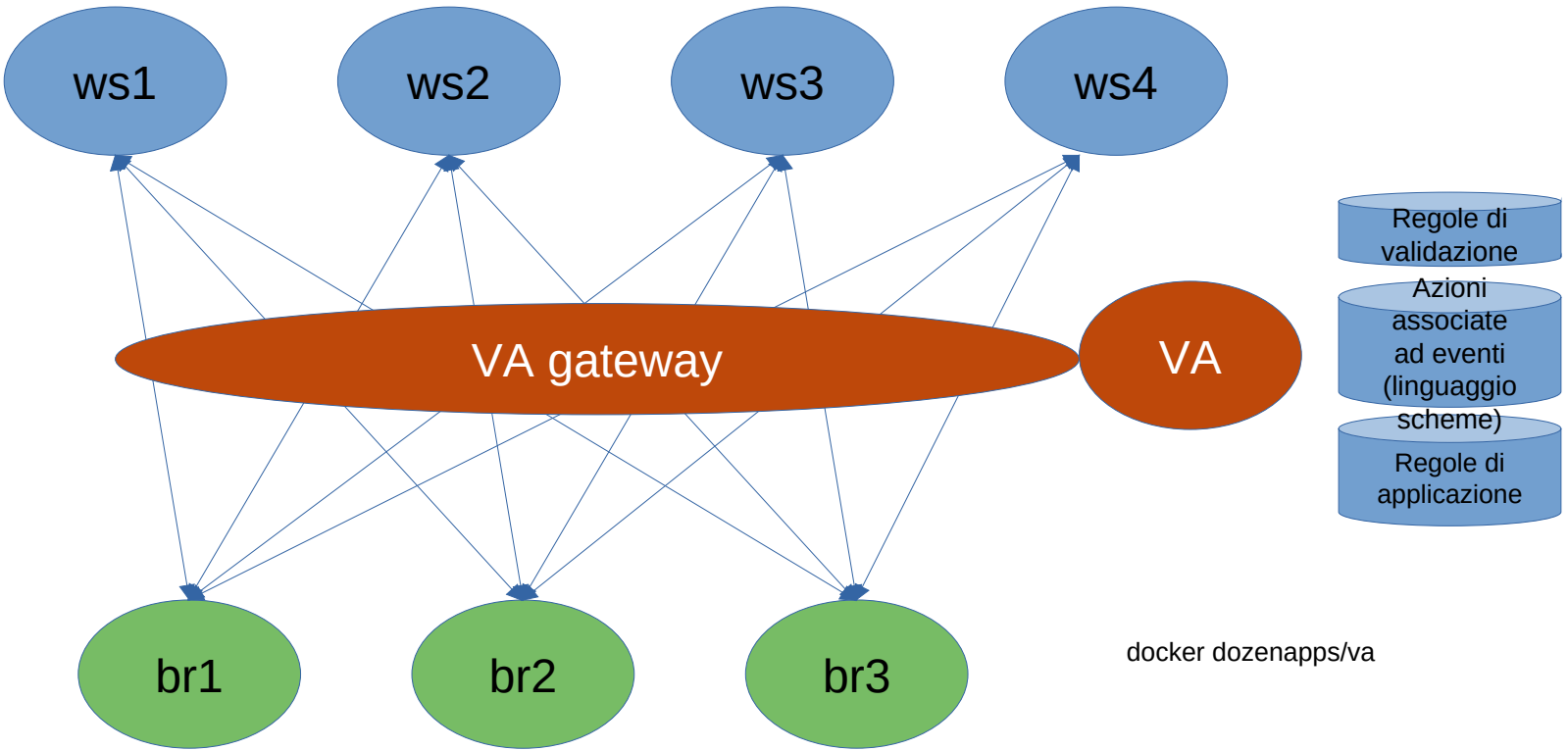
```
DEFINE urlset give_answer = { /ans, /ans/* };

DEFINE AR "answer"
        CONDITION
                http.url is in give_answer
        ACTION
                ANSWER "Ciao"
        ;
```

Validation rules (in parallelo, non in tempo reale)

```
DEFINE VR "USER-OK"
        CONDITION
                obs.event is net.send
                http.url is in api_rootsite
                http.query["username"] is in utenti_ok
        ACTION
                DEL CAT{http.query["username"], "-1"} from set errori
                DEL CAT{http.query["username"], "-2"} from set errori
                DEL CAT{http.query["username"], "-3"} from set errori
                DEL CAT{http.query["username"], "-4"} from set errori
                REPORT log { CAT { "Dropped all previous errors" } }
                SET "GOTOucla.edu"@v1="128.97.27.37:80"
                //SET "GOTOucla.edu"@v1="40.79.78.1:80"
                //enable  AR "GOTOucla.edu"
                //disable AR "GOTOucla1.edu"
```

Modello semplificato



docker dozenapps/va

Application rules (in serie/in tempo reale)

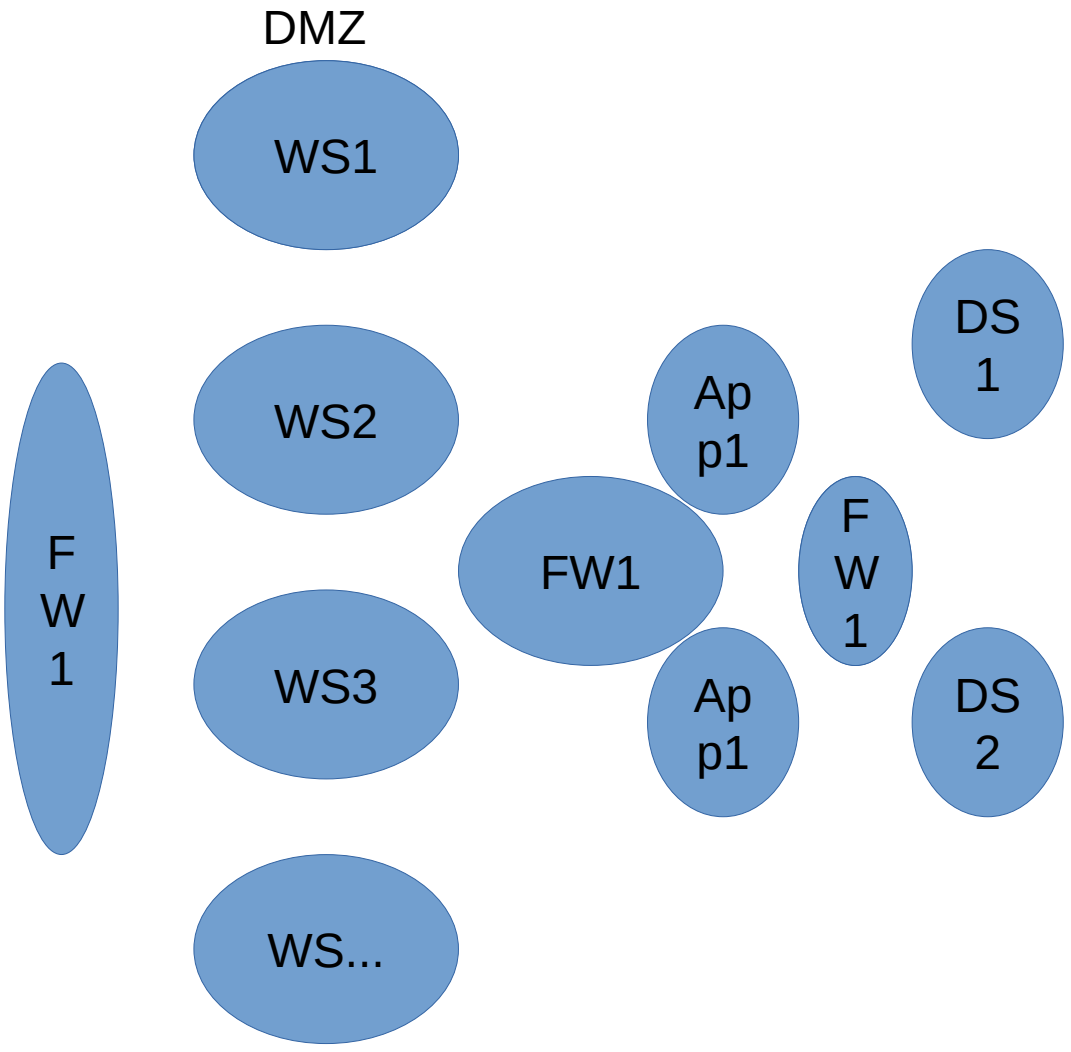
```
DEFINE urlset give_answer = { /ans, /ans/* };

DEFINE AR "answer"
        CONDITION
                http.url is in give_answer
        ACTION
                ANSWER "Ciao"
        ;
```

Validation rules (in parallelo, non in tempo reale)

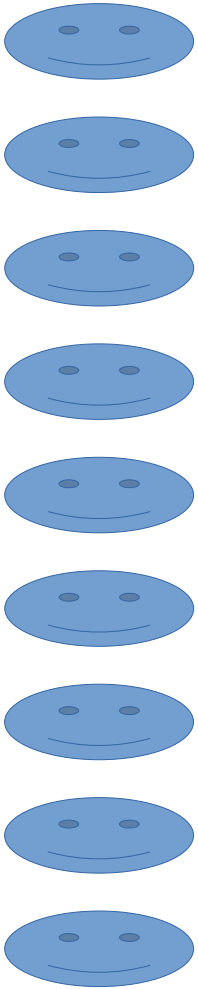
```
DEFINE VR "USER-OK"
        CONDITION
                obs.event is net.send
                http.url is in api_rootsite
                http.query["username"] is in utenti_ok
        ACTION
                DEL CAT{http.query["username"], "-1"} from set errori
                DEL CAT{http.query["username"], "-2"} from set errori
                DEL CAT{http.query["username"], "-3"} from set errori
                DEL CAT{http.query["username"], "-4"} from set errori
                REPORT log { CAT { "Dropped all previous errors" } }
                SET "GOTOucla.edu"@v1="128.97.27.37:80"
                //SET "GOTOucla.edu"@v1="40.79.78.1:80"
                //enable  AR "GOTOucla.edu"
                //disable AR "GOTOucla1.edu"
```

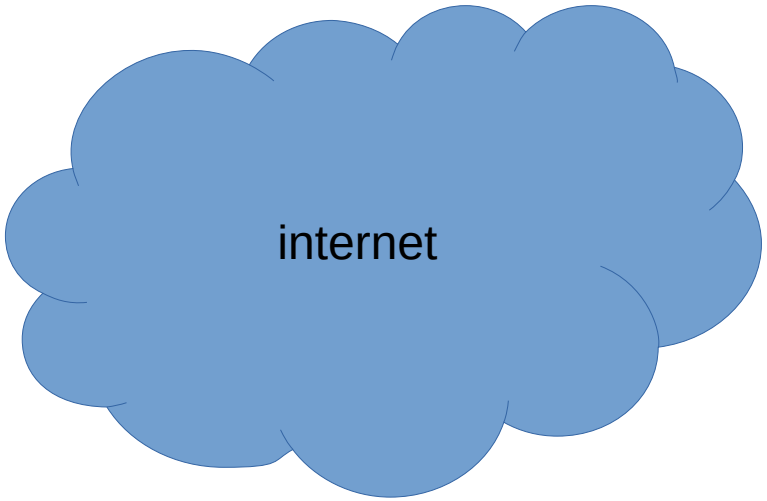
Un modello realistico di sistema di servizi



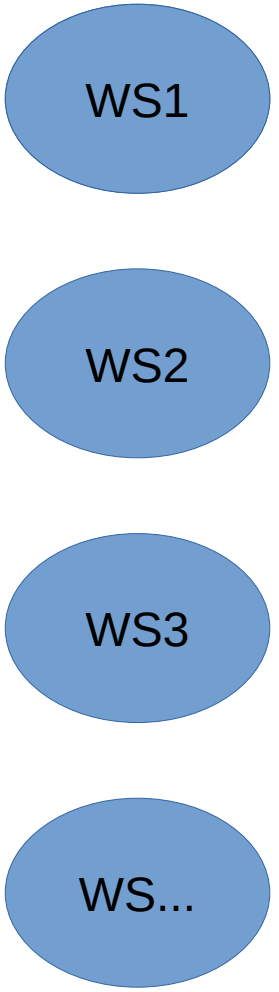
Evoluzione dei sistemi di servizi

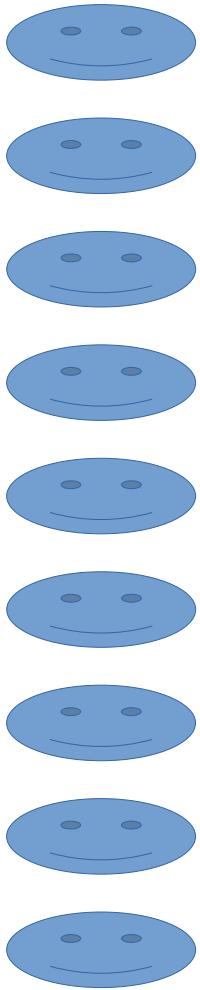
- Mainframe (anni ‘50/’60)
- Server dipartimentali
- client/server (ha dominato la scena degli anni ‘70/80/90)
- Sistemi WEB

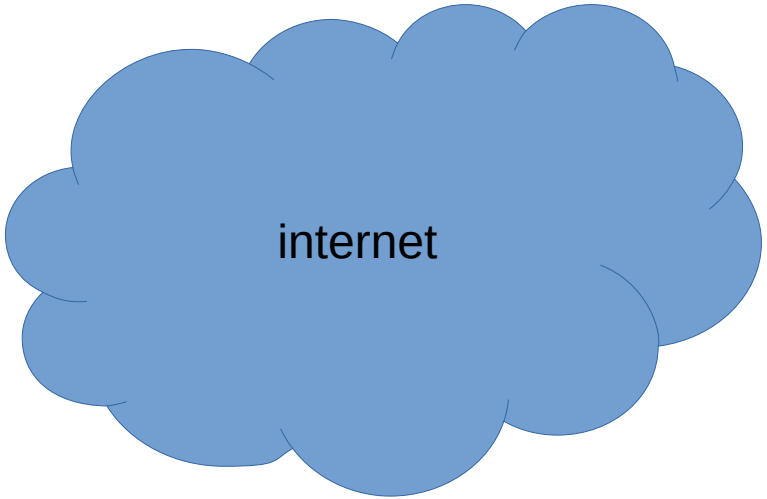




Il modello STATELESS ci consente di “incrementare” il numero di sistemi di front end a “costi minimi”
È sufficiente aggiungere una “macchina”







www.sdc2021.eu

WS’

WS’’

WS’’’

WS...

Data center

Ma se per motivi di efficienza triplico/quadruplico … il mio “sito web”, come gestisco gli utenti? Come faccio in modo che per gli utenti il mio sito sia sempre lo stesso (indirizzato dallo stesso IP/nome di dominio)? In che modo riesco a “bilanciare/distribuire” le richieste tra i vari web server (tutti uguali)?

Il problema del “load balancing”

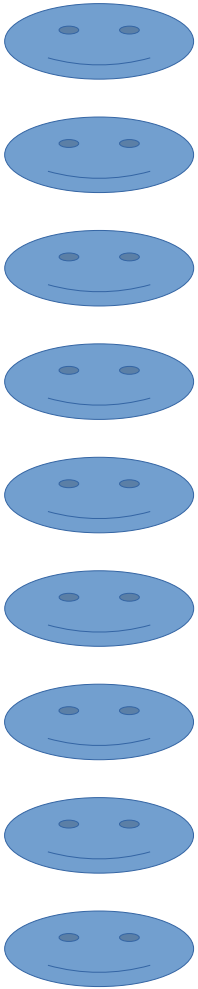
- Fornisco indirizzi IP/nomi di dominio diversi a diversi gruppi di utenti
  - Www.sdc2021A.eu
  - Www.sdc2021B.eu

Il problema del “load balancing”

- Realizzo un server DNS che, a rotazione, fornisce indirizzi IP differenti (i miei indirizzi IP) alle richieste di domain service
- Per lo stesso dominio: www.sdc2021.eu
  - Il DNS server risponde in un caso con 192.168.1.1, poi con 192.168.1.2, poi con 192.168.1.3, ….
  - 172…, 10...

Il problema del “load balancing”

- Inserisco, di fronte ai miei web server, un nuovo elemento che chiamerò di volta in volta
  - Firewall
  - Load balancer
  - Application gateway
  - Reverse proxy
  - ...



www.sdc2021.eu

internet

Front
end
di rete

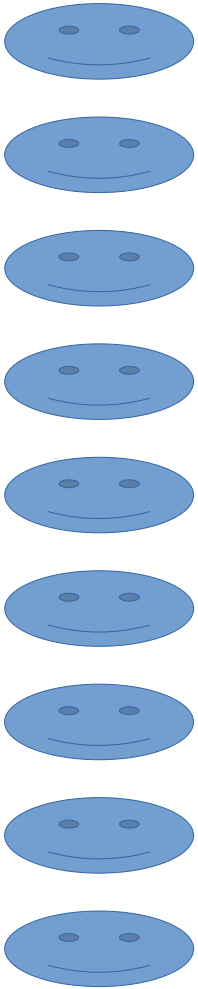
WS’

WS’’

WS’’’

WS...

- Bilanciare le richieste WEB
- nel caso di WS specializzati, assegnare i ws alle richieste
- gestire la sicurezza delle comunicazioni (prevenire attacchi)
- ...



www.sdc2021.eu

internet

Front
end
di rete

WS’

WS’’

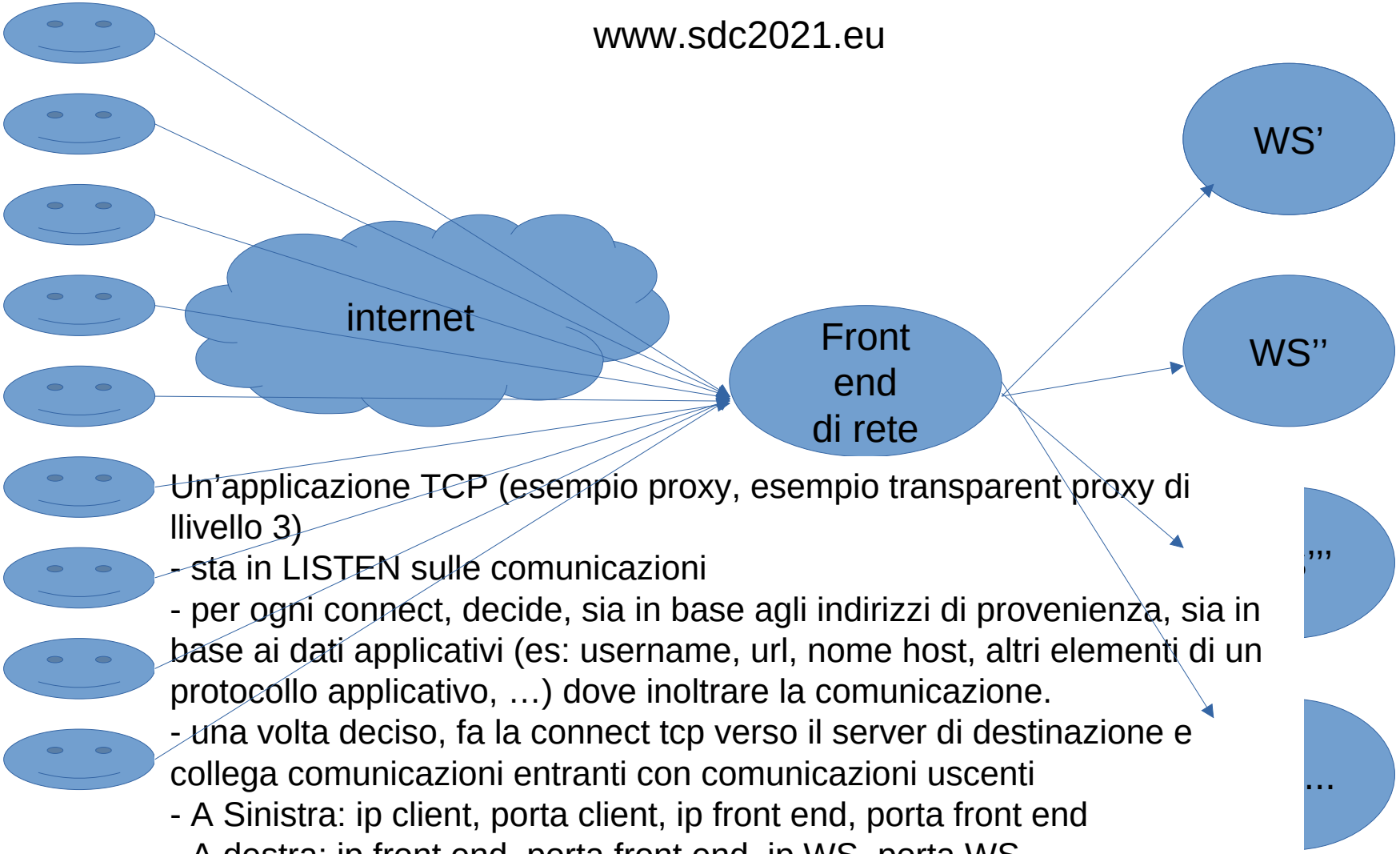
WS’’’

WS...

Esempio più classico: load balancer
Tutti i WS sono in grado di gestire le stesse richieste, è solo un problema di carico di lavoro.
Il front end di rete deve quindi, secondo strategie differenti, redirigere una richiesta verso il server meno carico

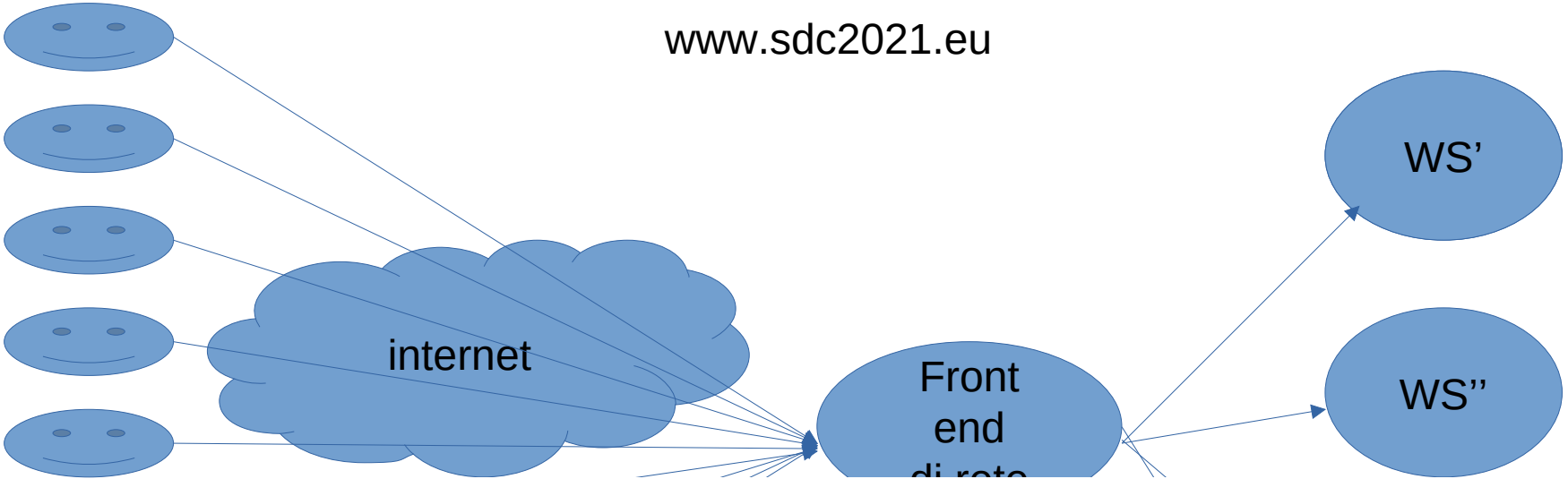
Un Firewall potrebbe andar bene?
- un firewall nell’accezione più tecnica lavora a “livello 2”, sul layer IP, non sul TCP
- 1) può filtrare in base agli indirizzi
- 2) può filtrare su “attributi” dei protocolli trasportati
- 3) puo “modificare” un frame, sostituendolo con uno differente

Domanda: ci sono vincoli sulla numerosità delle connessioni?
Una connessione IP è definita da una coppia (IP sorgente e IP destinazione). Il ruolo svolto da IPTABLES



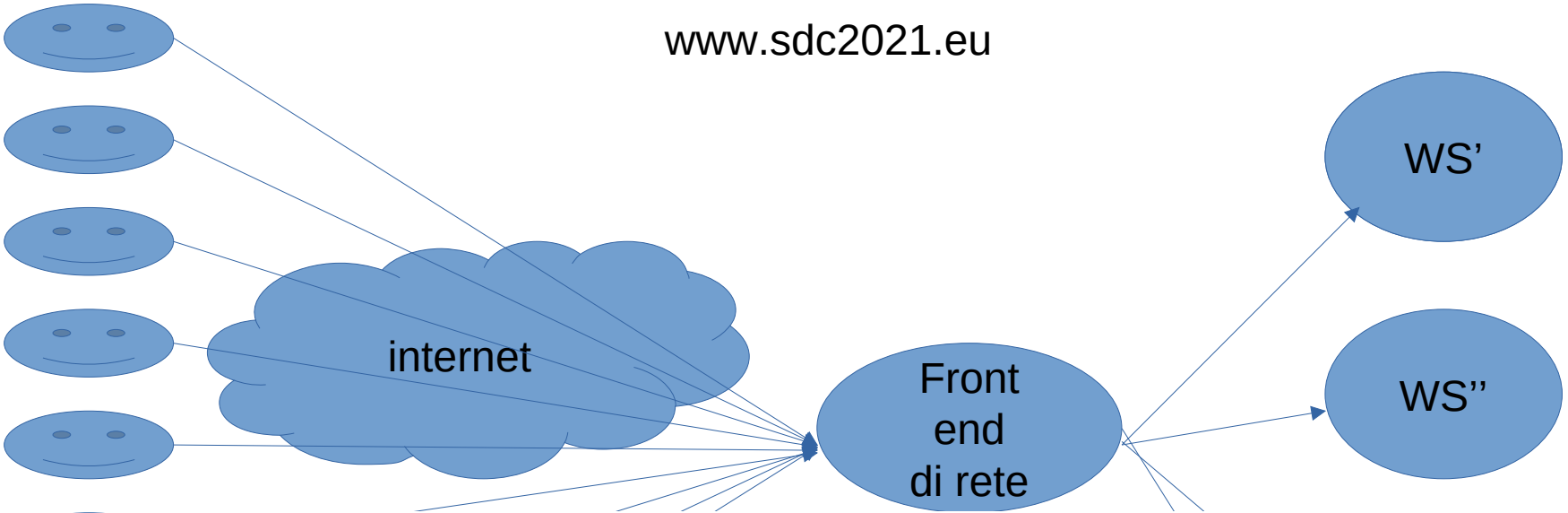
Un’applicazione TCP (esempio proxy, esempio transparent proxy di llivello 3)
- sta in LISTEN sulle comunicazioni
- per ogni connect, decide, sia in base agli indirizzi di provenienza, sia in base ai dati applicativi (es: username, url, nome host, altri elementi di un protocollo applicativo, …) dove inoltrare la comunicazione.
- una volta deciso, fa la connect tcp verso il server di destinazione e collega comunicazioni entranti con comunicazioni uscenti
- A Sinistra: ip client, porta client, ip front end, porta front end
- A destra: ip front end, porta front end, ip WS, porta WS

www.sdc2021.eu



C’è un limite al numero di comunicazioni? Esaminiamo il caso dei servizi WEB porta 80
- il front end di rete si presenta con un indirizzo IP e una porta nei confronti del territorio
- ogni WS fa lo stesso nei fronti del front end di rete

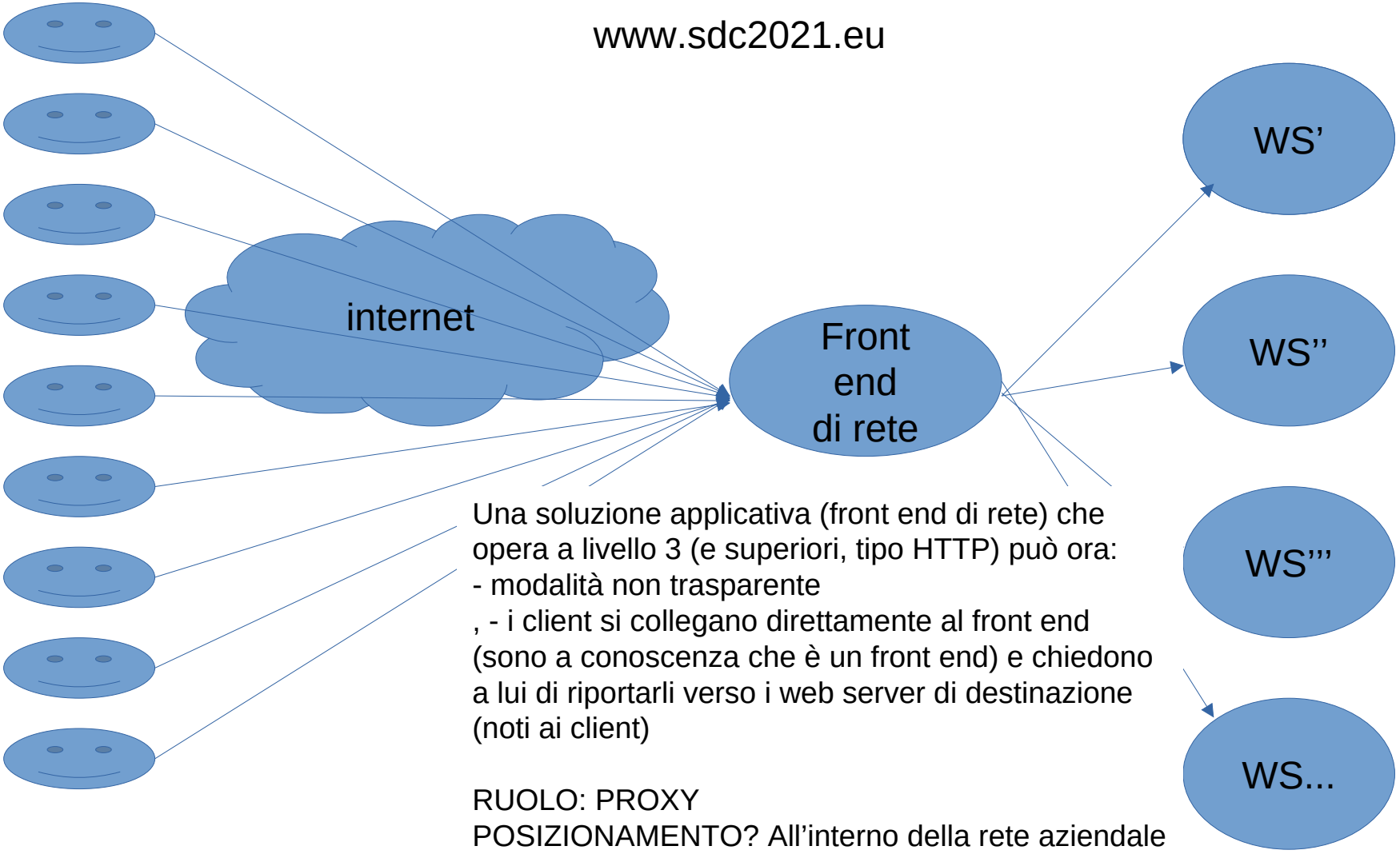
- quante comunicazioni possiamo avere in contemporanea senza esaurire le “risorse” di identificazione dei pacchetti TCP tra il territorio (i client su internet) e il front end?
- una comunicazione TCP è completamente descritta da 4 elementi
- ip src, porta src, ip dst, porta dst
, - quanti ip destinazione possiamo avere: 1
, - porte destinazione possiamo avere: 1
, - quanti ip sorgenti posso avere: 2^32
, - quante porte sorgenti posso avere: 2^16
Un computo grossolano ci porta a dire che le connessioni entranti sono limitate da 2^48



A fronte di 2^48 possibili connessione entranti, quante sono le possibili connessioni uscenti dal front end verso i web server di servizio?
- IP sorgenti: 1 quello del front end in rete interna
- porte sorgenti: 2^16
- IP destinazione: N(i web server)
- porte destinazione: 1xws, quindi 1
 Totale: N*2^16

Es: 4 ws, posso gestire al massimo 2^16*4=256000 connessioni contemporanee. Non è traffico ma collegamenti TCP “aperti”
- come risolvere?
- aumentando il numero di WS (un pochino costoso)

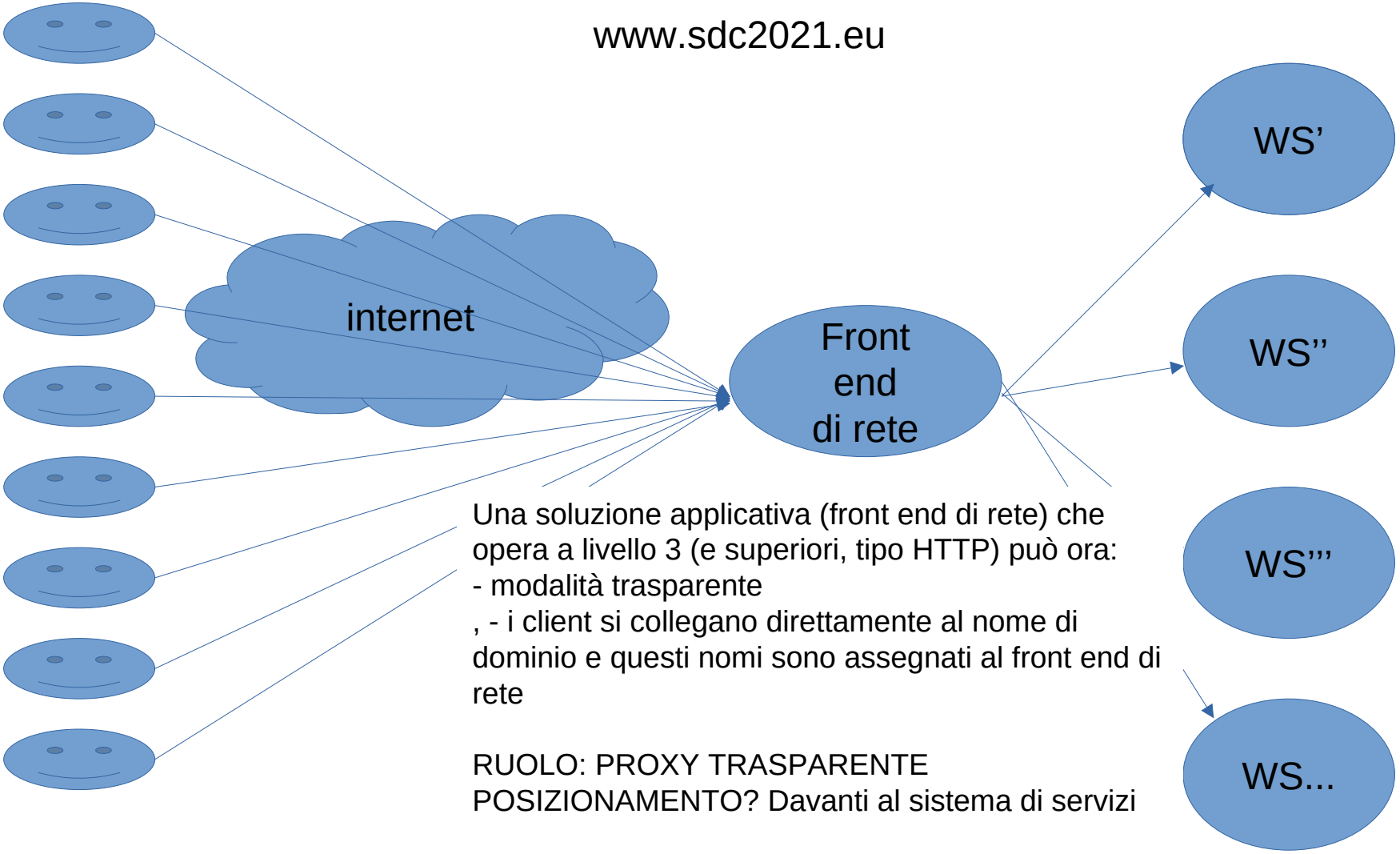
www.sdc2021.eu



Una soluzione applicativa (front end di rete) che opera a livello 3 (e superiori, tipo HTTP) può ora:
- modalità non trasparente
, - i client si collegano direttamente al front end (sono a conoscenza che è un front end) e chiedono a lui di riportarli verso i web server di destinazione (noti ai client)

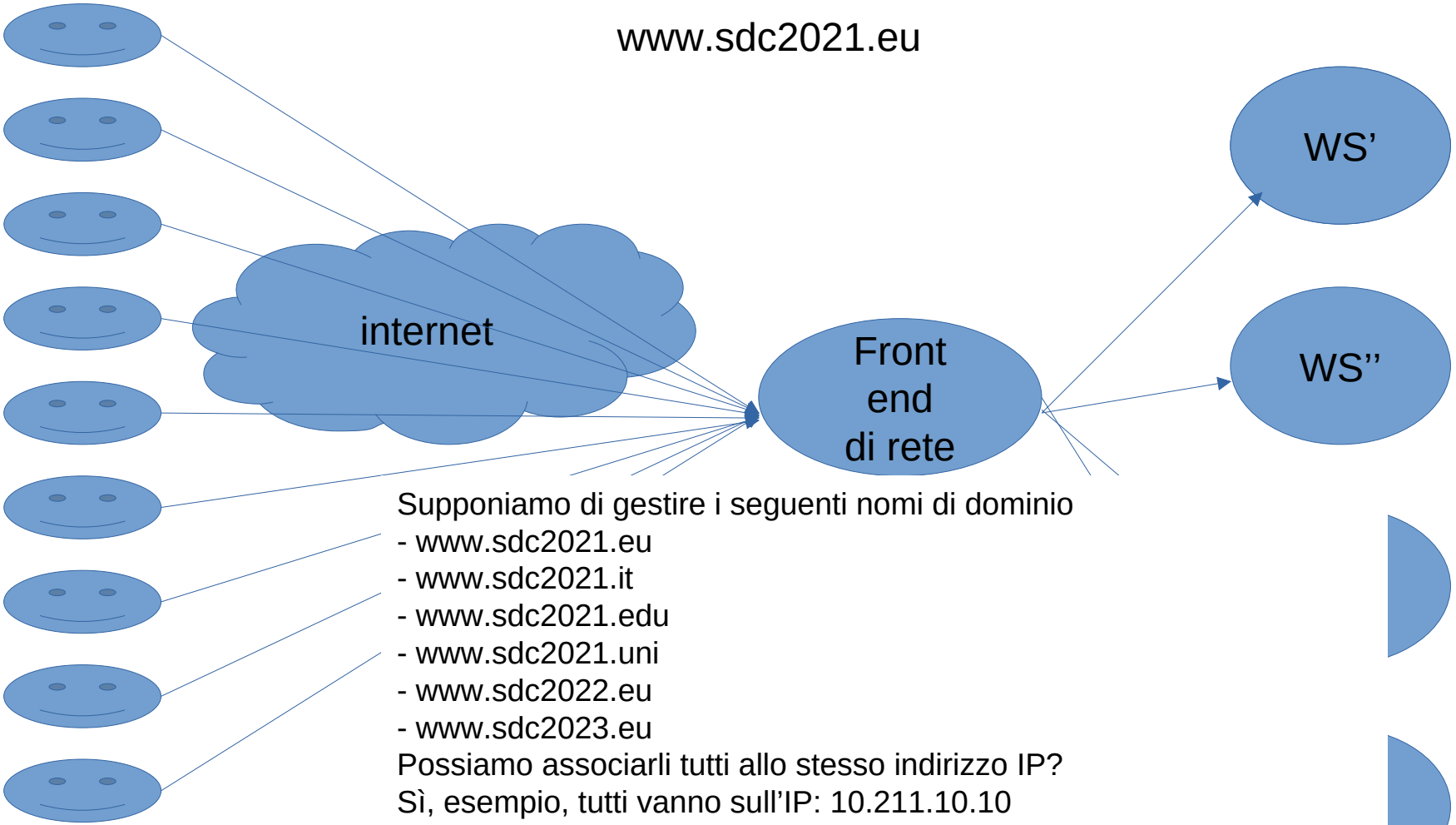
RUOLO: PROXY
POSIZIONAMENTO? All’interno della rete aziendale

www.sdc2021.eu



Una soluzione applicativa (front end di rete) che opera a livello 3 (e superiori, tipo HTTP) può ora:
- modalità trasparente
, - i client si collegano direttamente al nome di dominio e questi nomi sono assegnati al front end di rete

RUOLO: PROXY TRASPARENTE
POSIZIONAMENTO? Davanti al sistema di servizi

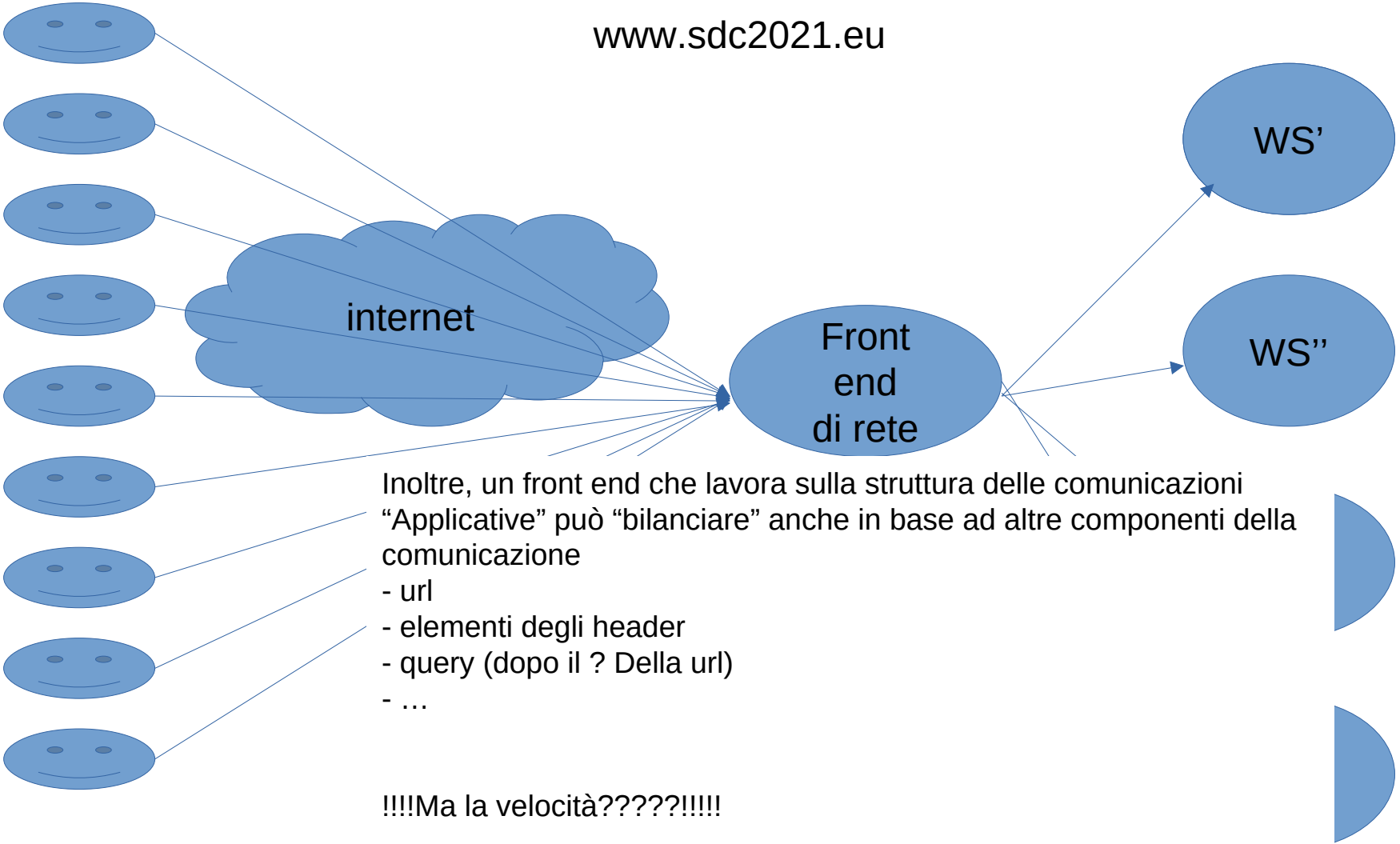


Supponiamo di gestire i seguenti nomi di dominio
- www.sdc2021.eu
- www.sdc2021.it
- www.sdc2021.edu
- www.sdc2021.uni
- www.sdc2022.eu
- www.sdc2023.eu

Possiamo associarli tutti allo stesso indirizzo IP?
Sì, esempio, tutti vanno sull’IP: 10.211.10.10

Se avessi lavorato a livello IP, allora la soluzione era:
Un indirizzo IP per ogni nome di dominio

www.sdc2021.eu

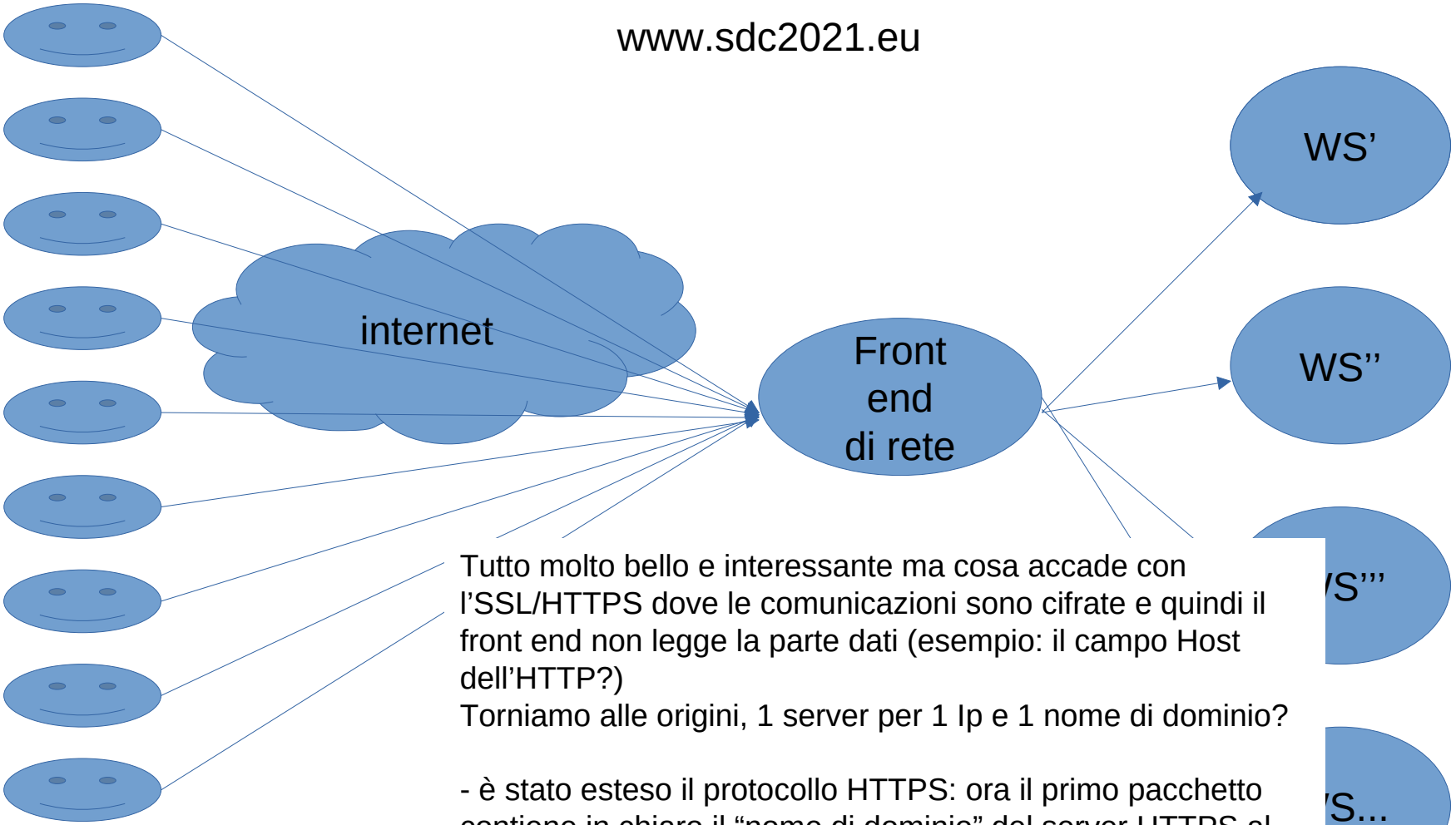


Inoltre, un front end che lavora sulla struttura delle comunicazioni “Applicative” può “bilanciare” anche in base ad altre componenti della comunicazione

- url
- elementi degli header
- query (dopo il ? Della url)
- …

!!!!Ma la velocità?????!!!!!

www.sdc2021.eu



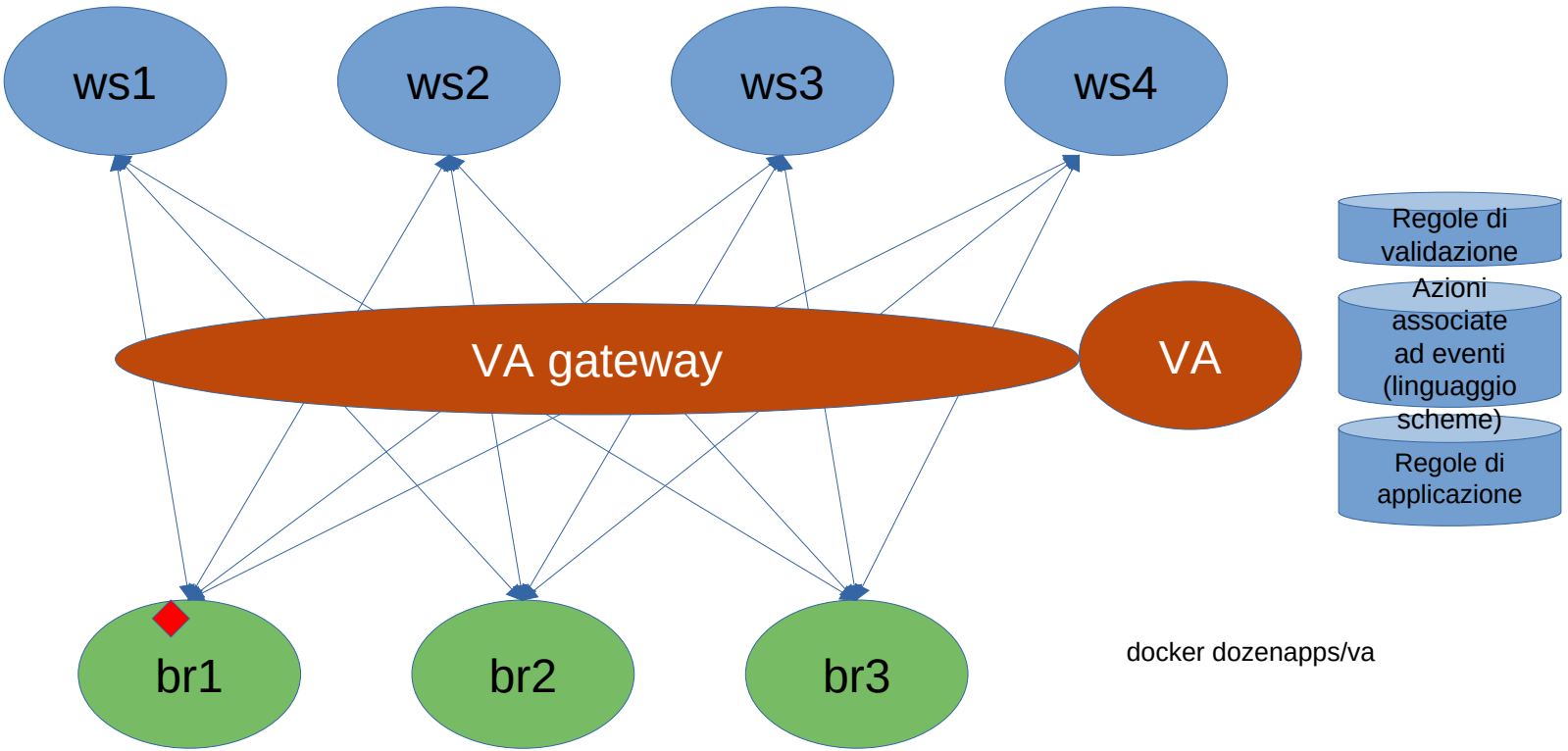
Tutto molto bello e interessante ma cosa accade con l’SSL/HTTPS dove le comunicazioni sono cifrate e quindi il front end non legge la parte dati (esempio: il campo Host dell’HTTP?)
Torniamo alle origini, 1 server per 1 Ip e 1 nome di dominio?

- è stato esteso il protocollo HTTPS: ora il primo pacchetto contiene in chiaro il “nome di dominio” del server HTTPS al quale intendo collegarmi.

Esercitazione

- Elencare tutti possibili elementi di una comunicazione TCP/IP di livello 4 (HTTP) che possano essere utilizzati come “observations” della VA
- Es:
  - IP sorgente
  - ….
  - Hostname (nell’header http)
  - ...

Modello semplificato



Observations

- Networking (i 3 layer TCP/IP)
- HTTP
- Applicativo (applicazione WEB)

Observations: TCP/IP

- DLL
  - MAC address: il server sotto quali condizioni riesce a “leggere” il macaddress del client?
    - Deve essere nella stessa rete locale
  - Se così non fosse, quale macaddress leggerebbe il server?
    - Quello del proprio gateway/router/firewall
- IP
  - IP-sorgente
    - il server sotto quali condizioni riesce a “leggere” ip del client?
      - Il server al più riesce a leggere l’IP della macchina/router/proxy esposta su internet appartenente alla rete del client
      - A volte, nella DMZ (area dei web server), il firewall “nasconde” gli IP che provengono dal territorio, sostituendoli con il proprio IP
  - IP-destinazione
    - Nel caso in cui abbia una soluzione in cui sono in ascolto, con lo stesso codice, su più IP, allora l’IP destinazione mi consente di discriminare quale istanza di server è stata indirizzata)
- TCP
  - Porta-sorgente
    - Ancora meno “forte” dell’IP sorgente.
    - NB: la quaterna (ips, ipd, portas, portad) definisce quasi completamente una sessione

Le condizioni di questa analisi sono:
- non ho agenti installati presso i client ma ho esclusivamente un agente presso i server !!!
Questo richiede la presenza di un forte requisito di sicurezza!!!

Observations: richiesta HTTP

- Prima riga della richiesta HTTP
  - Metodo
  - Url
  - Versione del protocollo
  - Query
  - Tag, username, password (http://username@password:pippo.com/pagina.html#item1)
  - Host
- Righe successive
  - Content-length
  - Cookie
  - Altri elementi dell'header http
- Parte dati (applicativa)
  - Protocollo http prevede un body la cui lunghezza è definita dal content-length, oppure un body "chunked", tipi mime/octect string
  - La parte applicativa può essere
    - Strutturata
      - **JSON** (SOAP, GSOAP, …)
      - **Sequenze key/value:**

Observations: Risposta HTTP

- Codice risultato (prima riga)
  - HTTP/1.1 200 OK, 302, 501, 400, …
- Elenco degli header
  - Content-length
  - Set-cookie
  - Tutti gli altri header
- Body
  - Stessi formati e protocolli dell’invio da parte del client

Una VA embedded in un reverse proxy

- docker run -i dozenapps/va /bin/bash
- Strato della configurazione di rete e dei servizi
- Strato delle regole applicative e strato delle regole di validazione

Esercitazione

- Installare docker
- Installare dozenapps/va
  - Il comando
    - docker run -it --network=host dozenapps/va /bin/bash -i
      - Questo è il log del comando sulla mia macchina
      - docker run -it --network=host dozenapps/va /bin/bash -i
      - root@franco-XPS-15-9570:/vapps#
      - …!!notate che appare la cartella /vapps (è una cartella virtuale creata dal container)
    - Deve mostrare la prompt dei comandi. Uscire con CTRL-D
- Leggere: va_parser.xhtml

Semplici comandi Docker

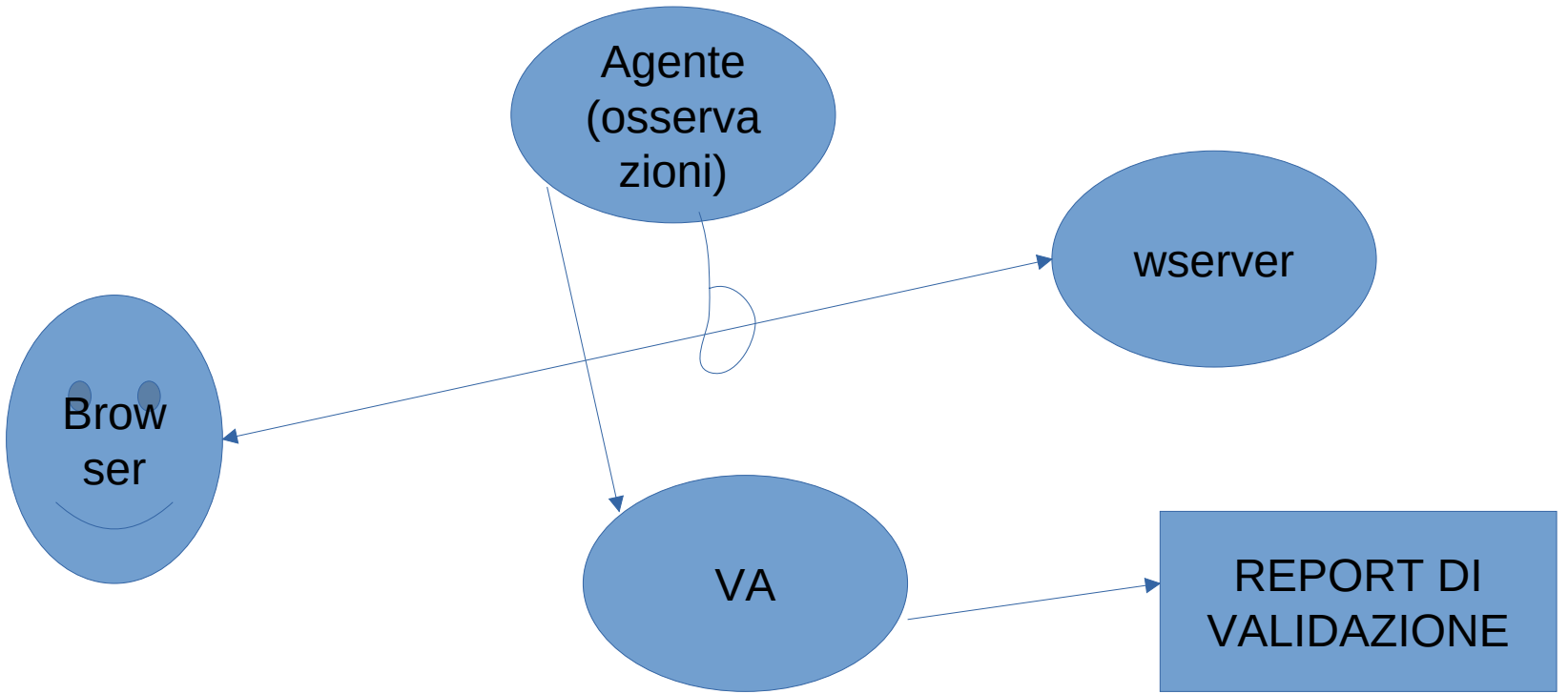
- docker run -it --network=host -v "$PWD":/vapps dozenapps/va /bin/bash
- docker ps
- docker images
- docker logs -f <pid>
- ...

Applicazione minimale della VA

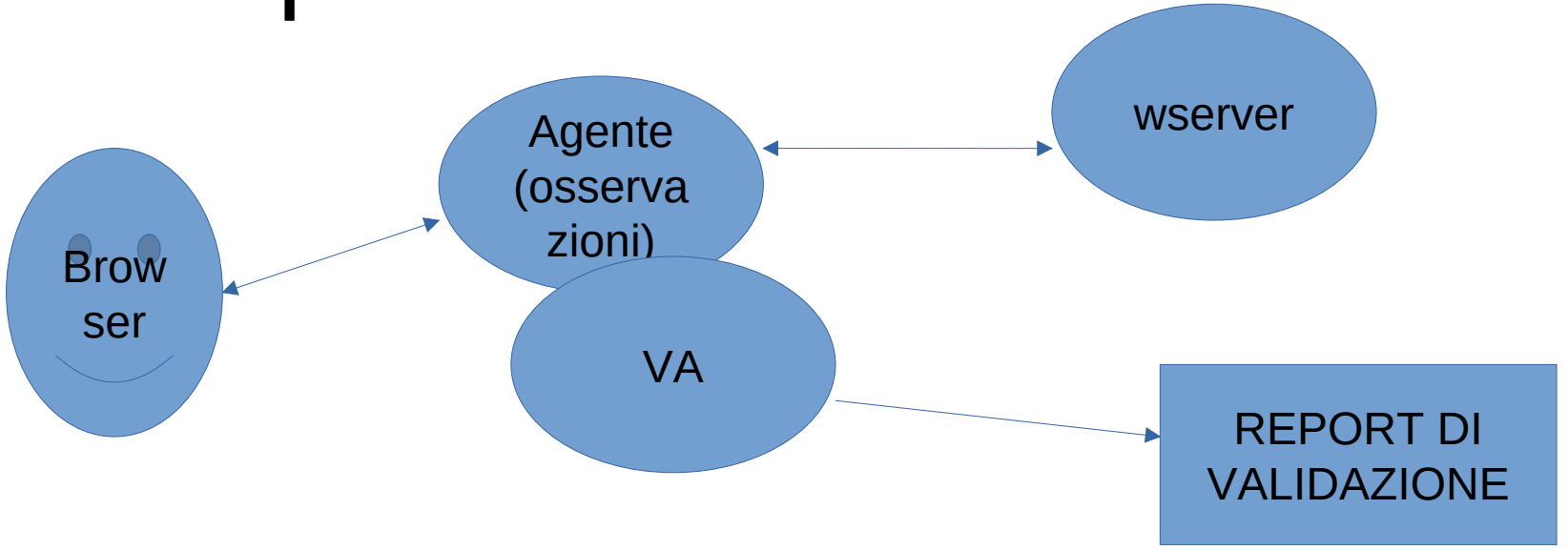
- Validazione della navigazione

- NB: la nostra architettura di sperimentazione richiede che il container operi anche come reverse proxy, per poter svolgere le funzioni di “agente” che invia le osservazioni!!

Architettura generale di sperimentazione



Architettura specifica di sperimentazione
Resta un’architettura realistica: pensate al mondo Cloud!!



Elementi operativi della VA

- .conf
- .ars
- .scm
- Cartella delle validazioni

Modello di sperimentazione semplice

- Dovete installare nodejs e npm
- Per installare un semplice web server:
  - sudo npm i -g http-server
- Per eseguire il web server nella cartella corrente
  - http-server . -a 127.0.0.1 -p 9999

Attivazione observer e VA

- Sdc-1604.ars
- Sdc-1604.conf
- Sdc-16.04.scm
  - È necessario che esista ma non siamo interessati ai suoi contenuti, per ora
- Creato cartella: report-1604

Esecuzione observer e va

- docker run -it --network=host -v "$PWD":/vapps dozenapps/va /va sdc-1604.conf

- => ManageLBUserArgsDefinitively started!!

- All’avvio, vanno in esecuzione
  - VM Scheme (guile)
  - Parser del linguaggio ARS

Sdc-1604.ars

- Elementi caratterizzanti del linguaggio
- Domanda:
  - Dovete acquisire “osservazioni” su comunicazioni di rete “in tempo reale”. Quali sono i criteri di progetto?
- Il riconoscimento delle osservazioni deve avvenire “in tempo reale”. Ad esempio riconoscere una specifica url, uno specifico insieme di indirizzi, ecc
  - Con le regex, potrei scrivere una regola per riconoscere, ad esempio, una specifica url
- Regex(“/folder[0-9]*/pagina[0-9][0-9].html”)
  - Utilizzereste il pattern matching (regex e similari)?
  - Il pattern matching può diventare “complesso” quanto vogliamo. Il riconoscimento della regola può diventare quadratico o, nei casi migliori, n*log(n)
- !!!la prima regola del linguaggio ARS è che tutte le operazioni devono avere un costo “costante” in modo che se la comunicazione è lunga N, il costo di analisi sia al più pari a c*N!!!

```
DEFINE ipset serverip = { *.*.*.*:* };
//nel linguaggio sono implementate strutture dati con costo di accesso "costante"
//non posso scrivere, ad esempio: 12*.1.*.4*:9*: costo di matching non lineare)


DEFINE ipset altri_server = { 192.168.*.*:8080, 160.*.*.*:80 };
//non è consentito: 1*.


//insiemi di stringhe


DEFINE set stringhe = {"ahsgf", "sdkjfhsk", "sdjfhskjhdf", "khsdgfhsd"};
//è una struttura dati con costo di ricerca indipendente dalla numerosità


//Gli insiemi di url seguono lo stesso modello degli ip
//Con * posso sostituire una pagina ma non parte del nome della pagina
DEFINE urlset lemieurl = { /pagina1/*, /pagina2/pag3/pag4/* }


(*
- Gli insiemi disponibili sono:
- set: insieme di stringhe
- ipset: insieme di indirizzi ip e porte
```

OBIETTIVO

- Consentire all’observer di Ossevare e inoltrare tutte le comunicazioni all’http server

- Ma le regole di validazione?
  - Sono le VR

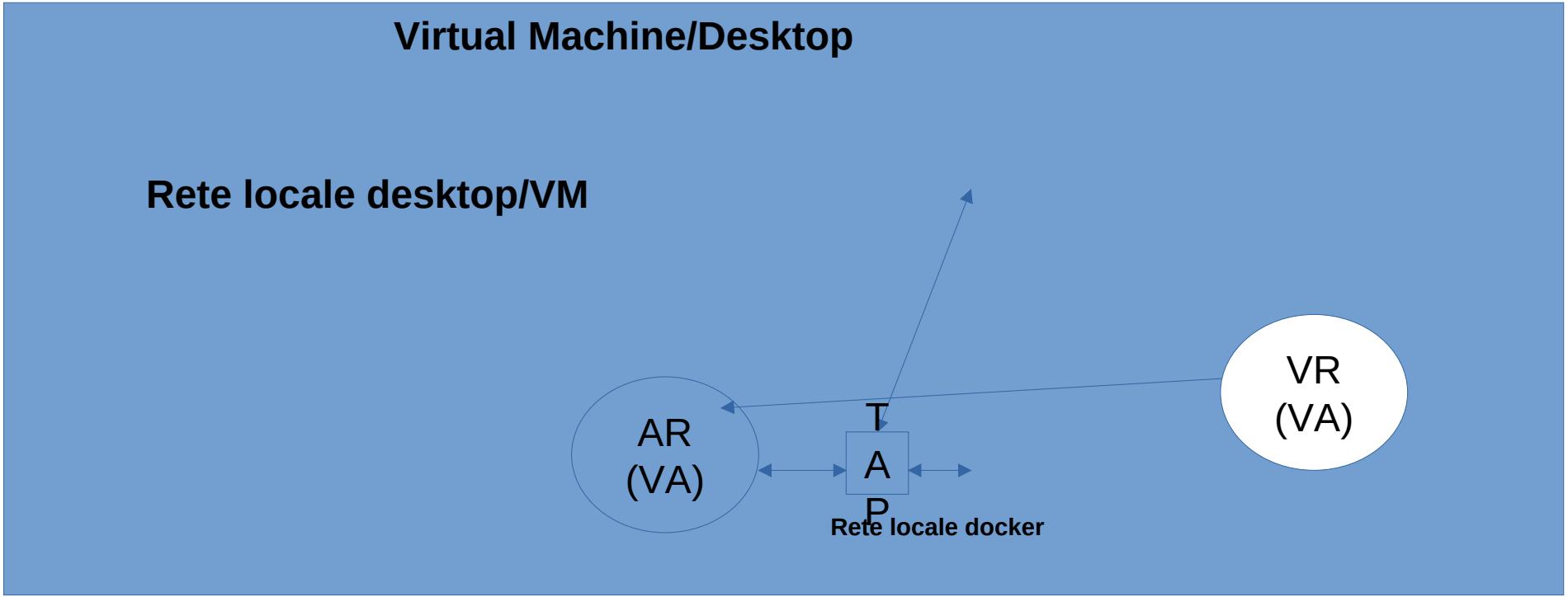
Esercizio

- Modificare il codice sdc-1604.ars al fine di far scattare la Validation rule solo per le comunicazioni dal browser al server
- Scrivere una seconda VR per le comunicazioni dal server al browser.
- Tracciare tutte le url di secondo livello: /*/*

Esercizio

- Modificare il codice sdc-1604.ars al fine di far scattare la Validation rule solo per le comunicazioni dal browser al server
- Scrivere una seconda VR per le comunicazioni dal server al browser.
- Tracciare tutte le url di secondo livello: /*/*

ARCHITETTURA DI FUNZIONAMENTO



TAP: dispositivo che effettua lo split di una comunicazione ethernet e la invia a più di una destinazione

Nostra applicazione

- WebServer in ascolto sulla porta 9999, IP della workstation (192.168.225.184)
- docker run -it -p 9900:80 -p 9901:81 -v "$PWD":/vapps dozenapps/va /va /vapps/sdc-1604.conf
  - Il container è in ascolto sulla porta 9900 della workstation e fa bridge sulla porta 80 all’interno del container
- Per esaminare in tempo reale cosa sta accadendo sulla rete
  - sudo tcpdump -i any -s 2000 -A port 9999

Un problema

- LE VR che abbiamo scritto finora, prendono solo un verso delle comunicazioni
  - Se dovessi verificare che un login è andato a “buon fine” ho la necessità di “sospendere” l’esecuzione di una “VR” (al riconoscimento della richiesta di autenticazione)
  - E attendere la risposta da parte del web server che conterrà l’esito dell’autenticazione

Esempio

- Impedire l’accesso alla pagina /va_parser.xhtml fino a quando il browser non navigherà con successo sulla pagina /esempio.scm
- La pagina rimarrà aperta per 10 secondi e poi sarà di nuovo bloccata

Esercitazione

- Elencare possibili modello di applicazione del modello di eliminazione/inserimento a tempo di valori in un set di url/cookie/header, … oppure di attivazione/disattivazione di Access Rules

- NB: non è necessaria la conoscenza della sintassi ARS. Dovete solo elencare, non realizzare

- Evidenziare livelli di sicurezza e privacy dei modelli presentati

Esercitazione

- Elencare possibili modello di applicazione del modello di eliminazione/inserimento a tempo di valori in un set di url/cookie/header, … oppure di attivazione/disattivazione di Access Rules

- NB: non è necessaria la conoscenza della sintassi ARS. Dovete solo elencare, non realizzare

- Evidenziare livelli di sicurezza e privacy dei modelli presentati

Ambiti di applicazione

- Controllo accessi
- Identity management
- Analytics
- Network Security, Cybersecurity
- 2nd factor authentication
- Alert & anomaly management
- Complex transactions Management
- Process mining
- ...

Un po’ di storia

- Esempio
  - Interfacce uomo macchina => separazione tra i vari strati applicativi
    - JAVA => MVC
  - Applicazione grafica
    - Inizialmente era tutto integrato: l’applicazione gestiva direttamente la rappresentazione dei dati
  - Oggi, html
    - css
    - Javascript (e sue evoluzioni)
    - Html
  - Oggi
    - Look and feel
    - Colori
    - Paradigma di interazione
    - Sono tutti al di fuori dell’applicazione
  - Le architetture sw a “strati”/”a cipolla” sono quelle vincenti
- I costi di sviluppo
  - Sono distribuiti, controllabili
  - Posso pianificare gli sviluppi nel tempo
  - Le interfacce sono **esterne** all’appicazione

Le comunicazioni

- Dalla gestione delle comunicazioni “direttamente” all’interno del codice,
- All’uso di layer/strati di comunicazione **esterni** al’’applicazione stessa

infrastruttura

- Le comunicazioni
- Le interfacce
- Autenticazione (identity provider)
- Sicurezza delle comunicazioni (HTTPS)
  - Non fanno più parte dell’applicazione

Infrastruttura (VA)

- ? l’applicazione deve gestire il controllo accessi?
- Un’applicazione deve erogare i servizi “applicativi” in base ai requisiti “applicativi”
- Esempio
  - Pagamento delle tasse on line
    - Le form devono rispettare le norme in vigore
    - L’app deve consentire di effettuare il pagamento corretto
- Tutta la sicurezza, identificazione, anomalie, privacy possono essere gestiti a livello di infrastruttura

Infrastruttura (VA)

- Su un’applicazione già **esistente e operativa** è possibile aggiungere controlli
  - Sicurezza
  - Accesso
  - Anomalie
  - Worlflow
  - ...

Infrastruttura (VA)

- È possibile descrivere nel linguaggio della VA la “sintassi” delle applicazioni “corrette”
  - La qualità di un sistema di servizi si misura rispetto al fatto che a ogni richiesta “ragionevole” il sistema risponde in modo “ragionevole” (rispondente ai requisiti)
- Come risponde un sistema a una richiesta “non ragionevole”?
  - anomalie

Attacchi web

- In genere è una POST su una pagina che non esiste
- Oppure è una POST/GET con una parte query non prevista dall'applicazione (troppo lunga, con un particolare formato, …)
- Se voi gestiste la sicurezza all'interno dell'applicazione, quando ci fosse una segnalazione di un nuovo tipo di attacco (es: GET con campo host molto lungo, parte dati con un numero di variabili (key) superiore a 128, ...), allora dovreste modificare l'applicazione

NB

- Un’applicazione ha un ciclo di vita ben superiore a quello dei suoi “sviluppatori”
  - COBOL, FORTRAN, Macro Assembler docet!!!
- Non posso ipotizzare di risparmiare facendo modificare agli sviluppatori un’applicazione per introdurre una nuova “protezione” rispetto agli attacchi
  - Ogni modifica a un’applicazione introduce “effetti collaterali” non identificabili a tempo della modifica
  - Nelle macchine a stati finiti ci sono stati “Don’t care”

Composizione di applicazioni

- Avete visto l’uso dell’operatore VA NEXT
- Quest’operatore ci consente di scrivere VR che operano su transazioni “complesse”, composte cioè da sequenze parzialmente ordinate di transazioni elementari
- Per ogni ACTION associata a una NEXT
  - Cioè per ogni transazione elementare riconosciuta come conclusa correttamente
- Posso attivare una AR (regola di accesso) specifica alla gestione del prossimo step del workflow

2nd factor authentication

- https://developers.yubico.com/U2F/Protocol_details/Overview.html
  - Quando avviene?
    - Dopo la prima autenticazione (login)
  - Come può essere realizzato?
    - Con semplici richieste (cose che conosco)
    - Con l’uso di strumenti esterni (cose che solo io posseggo)

2nd factor authentication

- https://developers.yubico.com/U2F/Protocol_details/Overview.html
  - Quando avviene?
    - Dopo la prima autenticazione (login)
  - Come può essere realizzato?
    - Con semplici richieste (cose che conosco)
    - Con l’uso di strumenti esterni (cose che solo io posseggo)

U2F

- User
  - Attiva browser e richiede l’accesso a server
- Server richiede le credenziali
- User fornisce le credenziali
- Server, noto lo user, invia un challenge
- Il Client (!!!!non lo user ma il suo browser) riconosce che è un challenge e richiede all’utente di inserire/associare il dispositivo di sicurezza che è stato in precedenza registrato dall’utente presso il server
- Il browser fa un challenge con il device e gli chiede di firmare il challenge stesso
- Il device (U2F) firma il challenge e lo riinvia al browser che a sua volta lo invia al server
- Server riconosce la firma e consente l’accesso

U2F: come è fatto il device U2F?

- Implementation
  - **Potrebbe essere una coppia di chiavi: pubblica e privata**
  - **Chi acquista il “device (oppure App di smartphone)” lo inizializza e SOLO a tempo di inizializzazione viene creata la coppia di chiavi**
- Ogni volta il challenge è firmato tramite la chiave privata
- La registrazione prevede che la chiave pubblica sia fornita al server “una tantum”

Vantaggi/Svantaggi

- È “definitivamente” associato all’utente, è come una smart card di autenticazione!
  - Quindi protegge chi?
    - Il server
  - Non protegge il client
- Non serve a cifrare
  - Il canale se fosse, sarebbe cifrato a termine dell’autenticazione fatta con una smart card/certificato digitale
- !!!non è questa l’implementation che cercavamo

U2F: come è fatto il device U2F?

- Implementation
  - **Un device che**
    - **contiene una chiave privata generata da una CA pubblica**
    - **Contiene un “vettore” di chiavi pubblica/privata, ognuna generata durante la registrazione a uno specifico sito web , dove la chiave è associata al sito web (alla url in fase di attivazione/registrazione)**

Vantaggi/Svantaggi

- Ogni coppia di chiavi è legata a uno specifico sito web, quindi se torno sullo stesso sito, userò la stessa coppia di chiavi
  - Questo protegge anche lo user
- A questo punto come è fatto il challenge?

U2F (dopo l’autenticazione)

- Server, noto lo user, invia un challenge
  - Significa che
    - Sa quanti challenge ha già fatto con questo user e quindi si aspetta che nella risposta ci sia un valore di challenge già svolti almeno pari a quello registrato presso il server
    - Il server conosce la chiave pubblica creata sul dispositivo esclusivamente per parlare con lui (quindi può cifrare con questa chiave una parte del challenge)
  - Il Client (!!!!non lo user ma il suo browser) riconosce che è un challenge e richiede all’utente di inserire/associare il dispositivo di sicurezza che è stato in precedenza registrato dall’utente presso il server
  - Il browser invia il challenge al device e gli chiede di firmare il challenge stesso, e gli dice anche la url alla quale si è collegato!!!
  - Il device (U2F) ritrova la coppia pubblica privata (in base alla url)
  - E firma il challenge con la chiave privata appena trovata, gli associa il numero di challenge già fatti e lo riinvia al browser che a sua volta lo invia al server
  - Server riconosce la firma, poiché conosce la chiave pubblica associata a lui e al device, e consente l’accesso
- La registrazione?
  - Viene concordata tra device e server la coppia di chiavi da utilizzare, al server viene fornita la pubblica

U2F

- Requisiti di sicurezza “rilassati” dal modello
  - Con javascript durante una sessione web, potete scrivere file sul FS del vostro PC?
    - Solo in spazi “riservati”, non sul FS generale
  - Con javascript durante una sessione web, potete accedere a device esterni?
    - No, si lavora sempre un una “SandBox”
  - Con javascript durante una sessione web, posso accedere ai dispositivi di rete (IP, TCP, …)?
    - No
- Questo modello ha “rilassato” i vincoli di sicurezza relativi alla comunicazione con device esterni
  - Comunica con il device U2F tramite USB o bluetooth
  - Sia di fronte a una situazione analoga alla “ottimizzazione locale” che è noto non migliora il funzionamento dell’interna applicazione, anzi, il più delle volte introduce effetti collaterali “disastrosi”
  - Qui è lo stesso: ho eliminato un vincolo a fronte di un modello di utilizzo che non è detto che abbia studiato tutti i possibili casi
  - In pratica si è venuta a creare una backdoor!!!

U2F: security

- Quali Attacchi sono possibili?
  - Phishing: un server simula di essere un altro server
    - No. Ogni server ha una chiave che ha concordato con me e che è unica e non nota a terze parti
  - MITM??
    - Se fosse completamente passante, allora non ci sarebbero problemi poiché non entra nel merito del challenge U2F ma, al termine, vedrà tutte le comunicazioni
      - Significa che in MITM che lavora sui certificati digitali (e quindi simula di essere il vostro server) in questo modello continua a funzionare, cioè ad effettuare attacchi che vanno a buon fine

MITM

Autenticazione HTTPS

- U: user
- M: MITM
- W: web server

Autenticazione HTTPS

- U: invia richiesta HTTPS a W
- W: risponde con una sfida cifrata tramite chiave privata e con un certificato digitale associato alla chiave privata
- U: verifica nell'elenco dei certificati "validi" se il certificato presentato da W è presente (qualcuno ha già accettato per voi l'elenco delle autorità di certificazione ritenute "valide")
  - Se presente, allora risponde al challenge e stabilisce la chiave "asimmetrica" di sessione con W

Autenticazione HTTPS con MITM

- U: invia richiesta HTTPS a W
  - MITM si collega a W e ottiene la sfida da parte di W
- W: risponde con una sfida cifrata tramite chiave privata e con un certificato digitale associato alla chiave privata
  - MITM risponde a U con un suo certificato che contiene lo stesso nome del certificato dell'web server fornito da W
- U: verifica nell'elenco dei certificati "validi" se il certificato presentato da W(MITM) è presente (qualcuno ha già accettato per voi l'elenco delle autorità di certificazione ritenute "valide" è ovvio che se MITM è riuscito a utilizzare una CA già presente in questa lista, allora tutti i certificati emessi da MITM saranno per voi validi)
  - Se presente, allora risponde al challenge e stabilisce la chiave "asimmetrica" di sessione con W
- Al termine U inserisce le proprie credenziali e MITM le potrà utilizzare sia nei confronti di W, sia nei confronti di altri siti.
  - MITM ha due chiavi si sessione
    - Una verso U
    - Una verso W
- U invia cifrato a MITM
- MITM decifra di cifra di nuovo e invia a W
- E viceversa

U2F e MITM

- C’è un miglioramento della sicurezza?
- Il protocollo consente di riconoscere la presenza di un MITM?
  - Il protocollo base, no!
- Ma, se, conoscendo il protocollo HTTPS
  - Nel protocollo HTTPS è presente un ChannelID che cambia per ogni sessione
  - Quindi se nel challenge tra WS e Device U2F rientrasse anche il Channel ID, allora la sicurezza sarebbe più elevata!!!
- Il server vede il ChannelID concordato con il MITM
- Il Cliente vede il Channel ID concordato con il MITM
- E, per la struttura stesa del protocollo HTTPS, questi due ChannelID non saranno praticamente mai uguali (li concordano congiuntamente i due peer) e quindi il MITM non potrebbe “forzare” la creazione di uno specifico ChannelID

Esercitazione

- Uso “a piacere” dell’ambiente VA
- Risultati attesi
  - Ho provato a fare questa cosa…
  - Ma ho avuto i seguenti problemi…
- Ognuno (non ogni gruppo ma ogni singolo) dovrà riportare un’esperienza di utilizzo, sia andata a buon fino, sia no.

MITM

Esercitazione

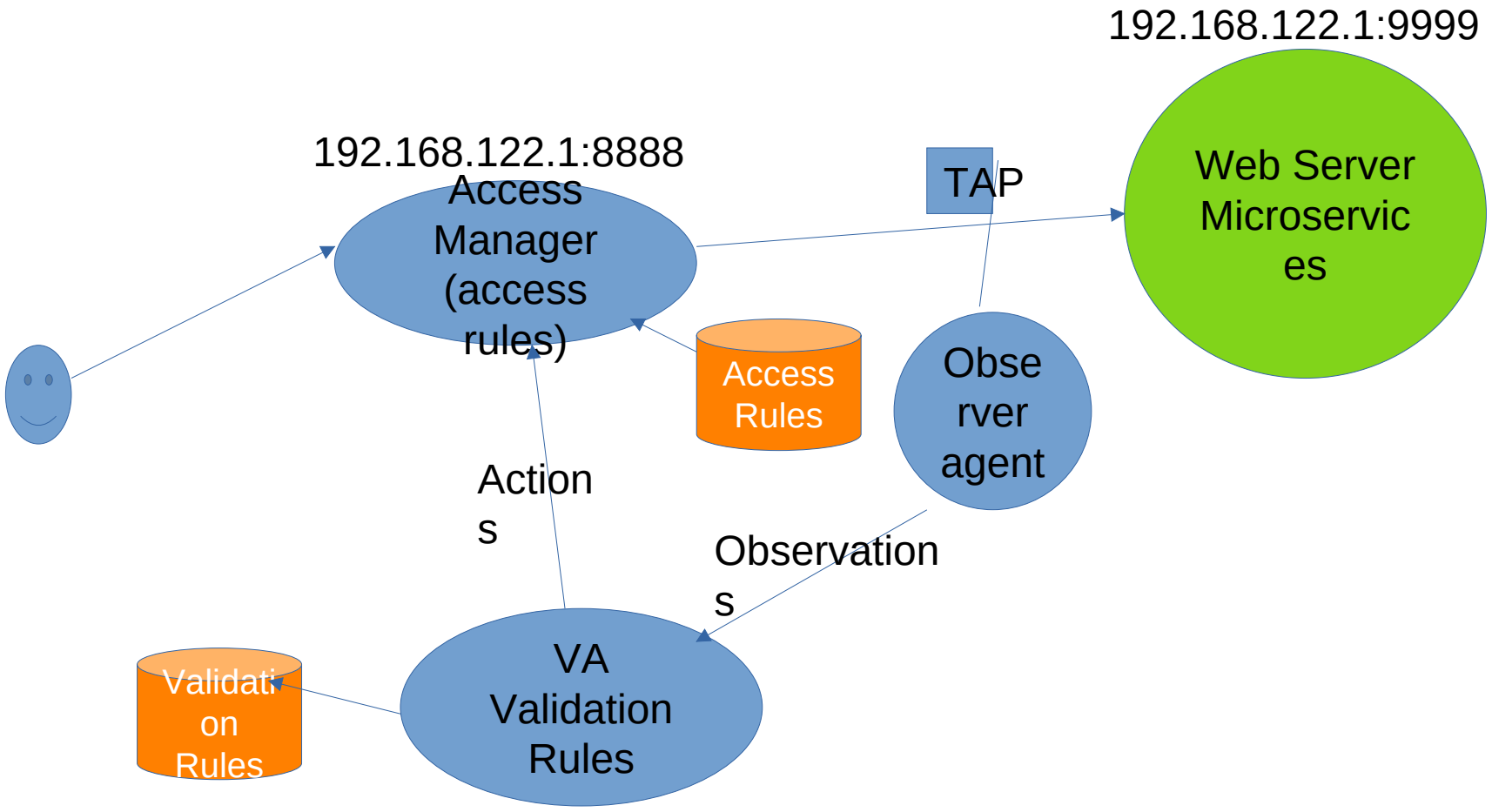
- Uso “a piacere” dell’ambiente VA
- Risultati attesi
  - Ho provato a fare questa cosa…
  - Ma ho avuto i seguenti problemi…
- Ognuno (non ogni gruppo ma ogni singolo) dovrà riportare un’esperienza di utilizzo, sia andata a buon fino, sia no.

Linguaggio ARS

- Linguaggio strightline
- Privo di cicli
- Con strutture di riconoscimento (pattern fissi) in tempo reale
- Estensioni verso il linguaggio guile/scheme

Gli hook verso il linguaggio GUILE

- Costruzione di microservizi
- Le AR sono analizzate una dopo l’altra, in serie, quindi l’ultima AR del file ARS in generale dovrebbe essere quella che risponde a tutti i casi non gestiti dalle precedenti
  - Se dietro c’è un Web Server l’ultima regola AR potrebbe essere il passaggio TCP.REDIRECT all’ip del web server
- Le VR sono analizzate “in parallelo”, quindi ogni volta se avete inserito N Validation Rule, sono tutte analizzate e quindi possono esserci più VR per le quali le condizioni sono contemporaneamente soddisfatte e che quindi scattano insieme


192.168.122.1:9999
192.168.122.1:8888
Access Manager (access rules)
TAP
Web Server Microservices
Access Rules
Observer agent
Actions
Observations
Validation Rules
VA Validation Rules

VR e AR: come costruirle

- È necessaria la conoscenza “sul campo” della struttura dei servizi. È un’attività di “Process Mining”:
  - Si ottiene dalle specifiche di progetto (progetto funzionale e dei protocolli applicativi)
  - Si ottiene “sul campo”, tramite sistemi di osservazione dei pacchetti sulla rete
- È quindi il risultato del “reverse engineering” dei protocolli applicativi

R.E. dei servizi su rete

- Applicativi di base
  - sudo tcpdump -i any -s 2000 -A port 9999
- Nel nostro caso
  - Richiesta
    - GET /msvc?nome=franco&servizio=accesso HTTP/1.1
  - Risposta
    - HTTP/1.1 200 OK
    - Content-Type: text/plain; charset=UTF-8
    - Connection: close
    - Content-Length: 24
    -
    - risultato=OK&valore=1221

I tempi delle reti

- Una sequenza del tipo
  - Q1=>A1
  - Q2=>A2
- Sulla rete può viaggiare in sequenze differenti
  - Q1Q2A2A1
  - Q2Q1A1A2
  - …
- Questo avviene in particolare quando ho più richieste “in parallelo”

Operatore NEXT

- L’operatore NEXT è di tipo generale, non è legato a una specifica sessione TCP/IP
- L’operatore NEXT clona la VR e la sospende in attesa che le sue conditions sia verificate
  - Infatti dopo la NEXT ci sono le condizioni
- Quindi quando scatta un operatore next per la VR, esempio, VR1, avremo in memoria due VR
  - La VR1 che parte dall’inizio (verifica delle condizioni iniziali)
  - La VR1 che parte dalle condizioni subito dopo la NEXT

Esempio

- VR
  - Condition
    - Exists Http.query[“nome”]
  - Action report log{http.query}
  - Next
    - Exists http.data[“risultato”]
    - Action report log
      { http.data[“risultato”] };
  - ;

- Req: GET /?nome=franco
  - Di queste ne invio N
- Ans: HTTP/1.1 …
  , body: risultato=Ok
- !!!Alla prima answer che soddisfa la condizione NEXT, **una** delle VR sospese sulla next sarà riattivata!!!

Next

- In alcuni casi non è necessario che sia presente un legame preciso tra prima e seconda parte.
  - Esempio: se sto validando una transazione di acquisto di un libro e vengono effettuate due transazioni “identiche” per lo stesso libro e lo stesso prezzo, allora il fatto che siano documentate/validate in modo “incrociato” non cambia la valenza della validazione
- Se due risposte di transazioni non sono tra loro distinguibili, allora significa che la distinguibilità non è oggetto di validazione.
- Quindi se a due richieste Q1 e Q2 ho due risposte identiche (A’, A’), allora il fatto di aver associato Q1 alla prima delle A’ o alla seconda non cambia la qualità della validazione.
  - Immaginate di avere un sito che risponde in modo asincrono con un messaggio di posta.
  - Quindi alla richiesta A1 risponde con un messaggio di posta che contiene solo “OK”
  - Alla richiesta A2 risponde con lo stesso messaggio.
  - Sicuramente saprò e potrò certificare che a fronte di due richieste sono giunte due risposte positive.
  - Cioè se la VA non riesce a distinguere una risposta dall’altra, neppure l’utente sarà in grado di farlo!!!!

Transazioni complesse

- Immaginiamo ora che U1 faccia una transazione T1 con S1
- E poi faccia una seconda transazione T2 con S2
- Allora la regola sarà:
- VR
  - Condition
    - U1 fa T1 verso S1
  - Next
    - U1 fa T2 verso S2
  - End

Transazioni complesse

- Immaginiamo ora che U1 faccia una transazione T1 con S1
- E poi faccia una seconda transazione T2 con S2
- Allora la regola sarà:

- VR
  - Condition
    - U1 fa T1 verso S1
  - VAR
    - SameUser=U1
  - Next
    - user SameUser fa T2 verso S2
  - End

Next

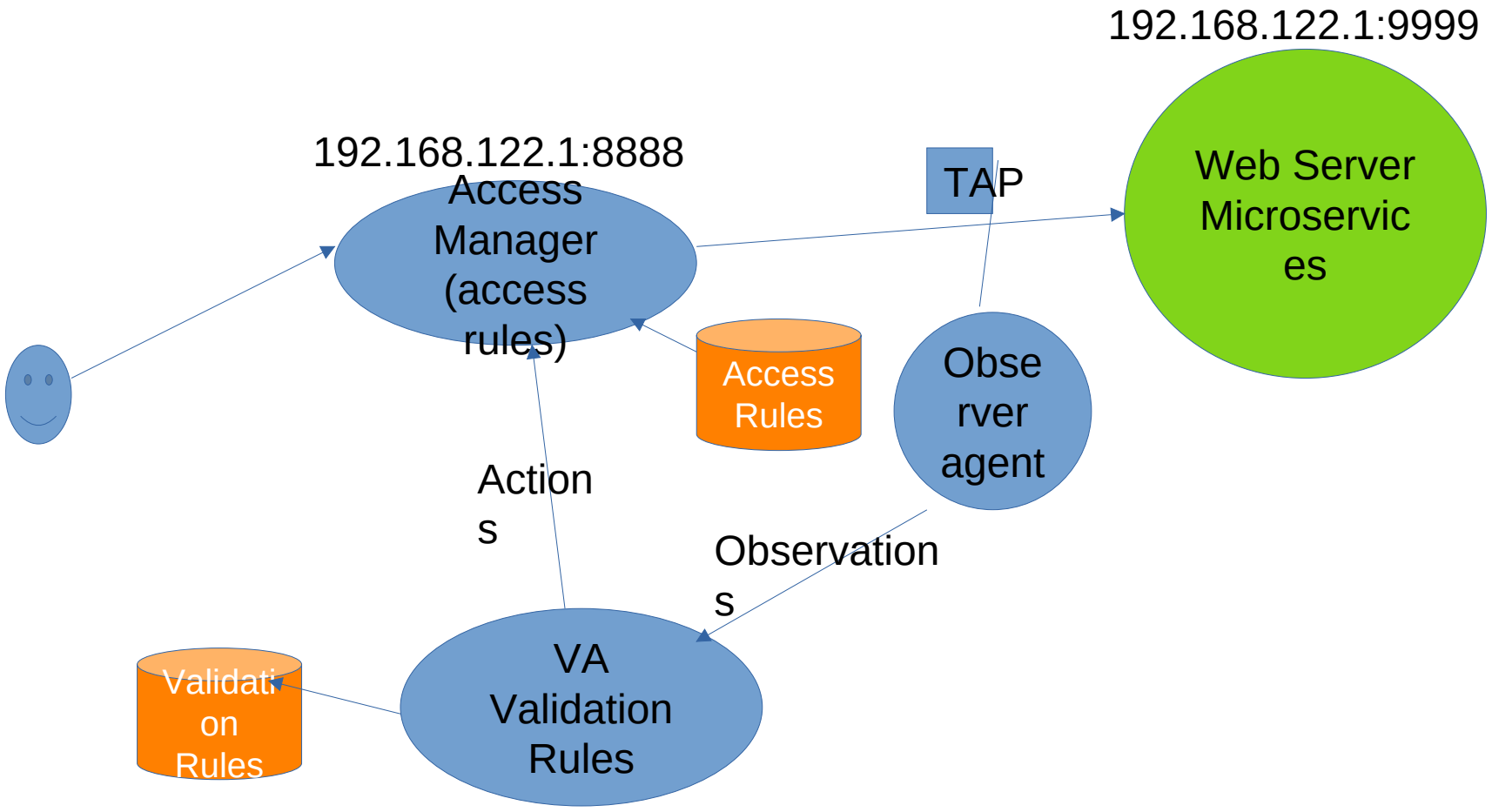
- La VR resta sospesa ma posso definire il contesto (VAR) che mi “seguirà” nella valutazione delle condizioni subito dopo la NEXT

Esempio

- L’utente franco chiede il servizio “accesso” e poi chiede il servizio “stampa”
  - Questa è una richiesta corretta, altre sequenze sono anomale
- curl "http://127.0.0.1:8888/msvc?nome=franco&servizio=accesso"
- curl "http://127.0.0.1:8888/msvc?nome=franco&servizio=stampa"

Esercitazione: definizione di un semplice WorkFlow

- Estendere VR a AR on modo tale che solo quando un utente USER avrà richiesto l’accesso, sarà abilitato il servizio “stampa” e solo per questo utente.
- Quando l’utente USER avrà correttamente usufruito del servizio “stampa”, non potrà usufruirne ulteriormente


192.168.122.1:9999
192.168.122.1:8888
Access Manager (access rules)
TAP
Web Server Microservices
Access Rules
Observer agent
Actions
Observations
Validation Rules
VA Validation Rules

Esercitazione: definizione di un semplice WorkFlow

- Estendere VR a AR on modo tale che solo quando un utente USER avrà richiesto l’accesso, sarà abilitato il servizio “stampa” e solo per questo utente.
- Quando l’utente USER avrà correttamente usufruito del servizio “stampa”, non potrà usufruirne ulteriormente

Linguaggio ARS delle regole di accesso e validazione

- Mysql
  - Caricamento di set di stringhe
  - Aggiunta di record dentro VR
- Interazione con linea di comando
- {"ar-disable", VaCommands::AR_DISABLE},
- {"ar-enable", VaCommands::AR_ENABLE},
- {"ar-list", VaCommands::AR_LIST},
- {"vr-add", VaCommands::VR_ADD},
- {"vr-disable", VaCommands::VR_DISABLE},
- {"vr-enable", VaCommands::VR_ENABLE},
- {"vr-list", VaCommands::VR_LIST},
- {"set-list", VaCommands::SET_LIST},
- {"set-get", VaCommands::SET_GET},
- {"set-add-value", VaCommands::SET_ADD_VALUE},
- {"set-del-value", VaCommands::SET_DEL_VALUE},

VR

- All’interno di una VR è possibile chiamare una funzione LISP
- Sintassi: stringa che inizia con “? ”
  - ADD CAT{"? (lambda (pbuf) (Show 1)(va::vaeng::AddToDict \"url_keys\" (mtfa-eis-get-current-uri pbuf) (mtfa-rand-alfanum 10 mtfa-char-set-alfa) \"60\" #f) \"dummy\")"} TO SET DUMMY

```
DEFINE urlset json_rootsite  = { /JSON  }; // */
DEFINE urlset api_rootsite  = { /API  }; // */
DEFINE urlset http_rootsite = { /, /* }; // */

DEFINE ipset all_ip = { *.*.*.*:* };

DEFINE AR "8083"
 , CONDITION
, , net.ipdst is in all_ip
 , ACTION
, , //ANSWER "Ciao"
, , tcp.redirect "? uno due tre quattro"
 , ;

(*

DEFINE VR "LOG"
, CONDITION
, , http.url is in http_rootsite
, ACTION
, , REPORT "all.txt" {http.uri}
, ;
*)
```

```
OBS.event is net.send
CALL vr_pippone WITH http.answer.data["0_FULL_DATA"], net.sesid, http.answer.code
ACTION
CALL vr_plutone WITH http.url, http.host, net.sesid
REPORT "all.txt" {http.uri}
;

//Prove di LISP CALL IN AR
DEFINE AR "AR lisp services management"
CONDITION
CALL ar_pippone WITH "" //http.host, http.uri
//http.url is in http_rootsite
ACTION
//CALL ar_plutone WITH http.url, http.host, http.method
answer "ciao"
;
```

VATEST.SCM

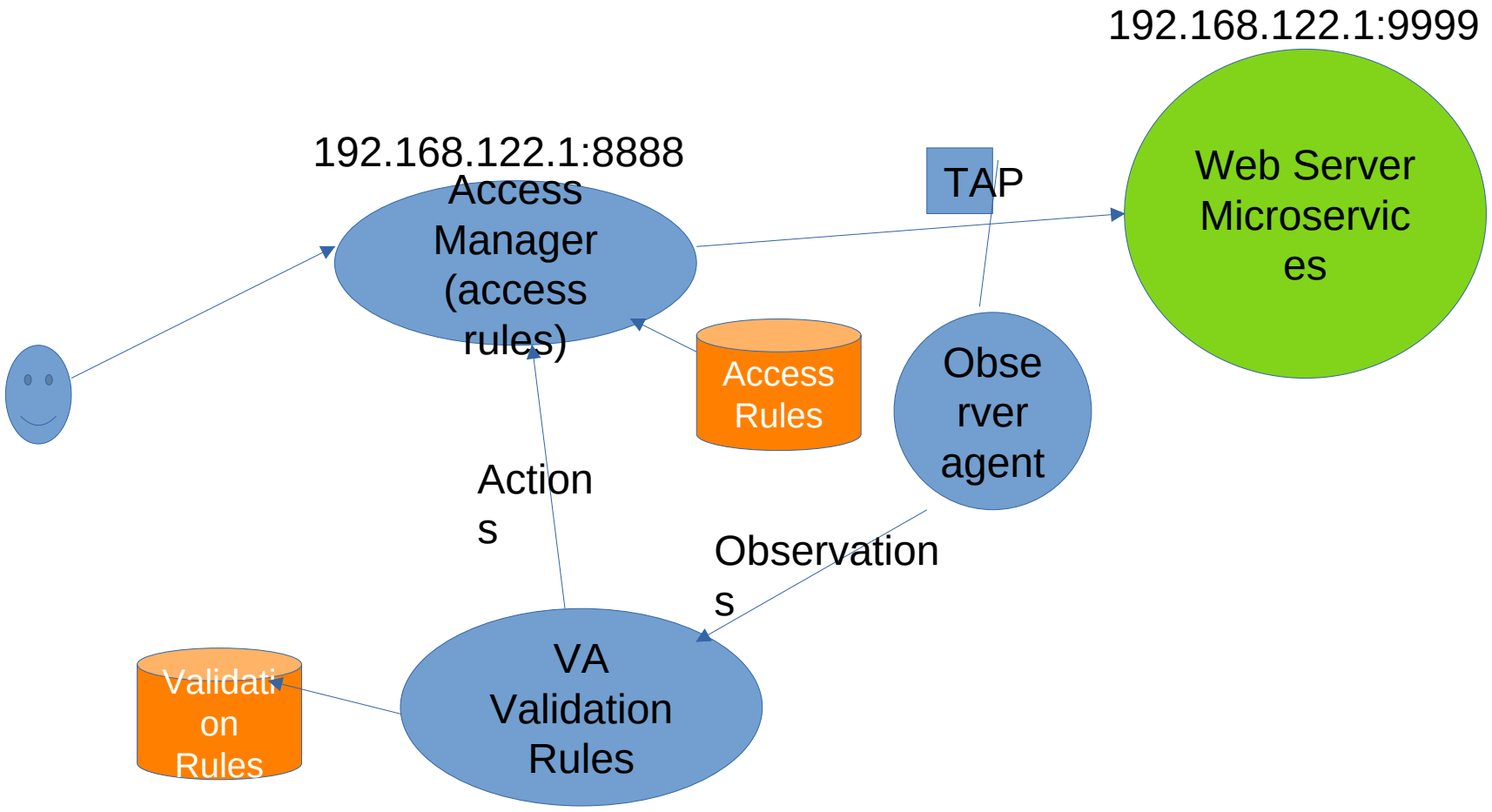
Next

- Discussione di proposte di progetto finale

```
DEFINE AR "lisp services management"
CONDITION
http.url is in api_rootsite
ACTION
//MANAGE "HTTP-API", "alfa", "beta", "gamma"
MANAGE "HTTP-API"
;

DEFINE AR "json lisp services management"
CONDITION
http.url is in json_rootsite
ACTION
MANAGE "JSON"
;

DEFINE AR "HTTP management"
CONDITION
http.url is in http_rootsite
ACTION
http.header["Forwarded"]="by=hidden;for=hidden;host=hidden;proto=https"
tcp.redirect "? uno due tre quattro"
;
```


192.168.122.1:9999
192.168.122.1:8888
Access Manager (access rules)
TAP
Web Server Microservices
Access Rules
Observer agent
Actions
Observations
Validation Rules
VA Validation Rules

```
DEFINE AR "lisp services management"
CONDITION
http.url is in api_rootsite
ACTION
//MANAGE "HTTP-API", "alfa", "beta", "gamma"
MANAGE "HTTP-API"
;

DEFINE AR "json lisp services management"
CONDITION
http.url is in json_rootsite
ACTION
MANAGE "JSON"
;

DEFINE AR "HTTP management"
CONDITION
http.url is in http_rootsite
ACTION
http.header["Forwarded"]="by=hidden;for=hidden;host=hidden;proto=https"
tcp.redirect "? uno due tre quattro"
;
```

KamRun

- Necessità di effettuare un’operazione aritmetica
- Tornare un valore dinamico per la redirect
- Aggiungere un valore “calcolato” a un insieme
- ...

CALL

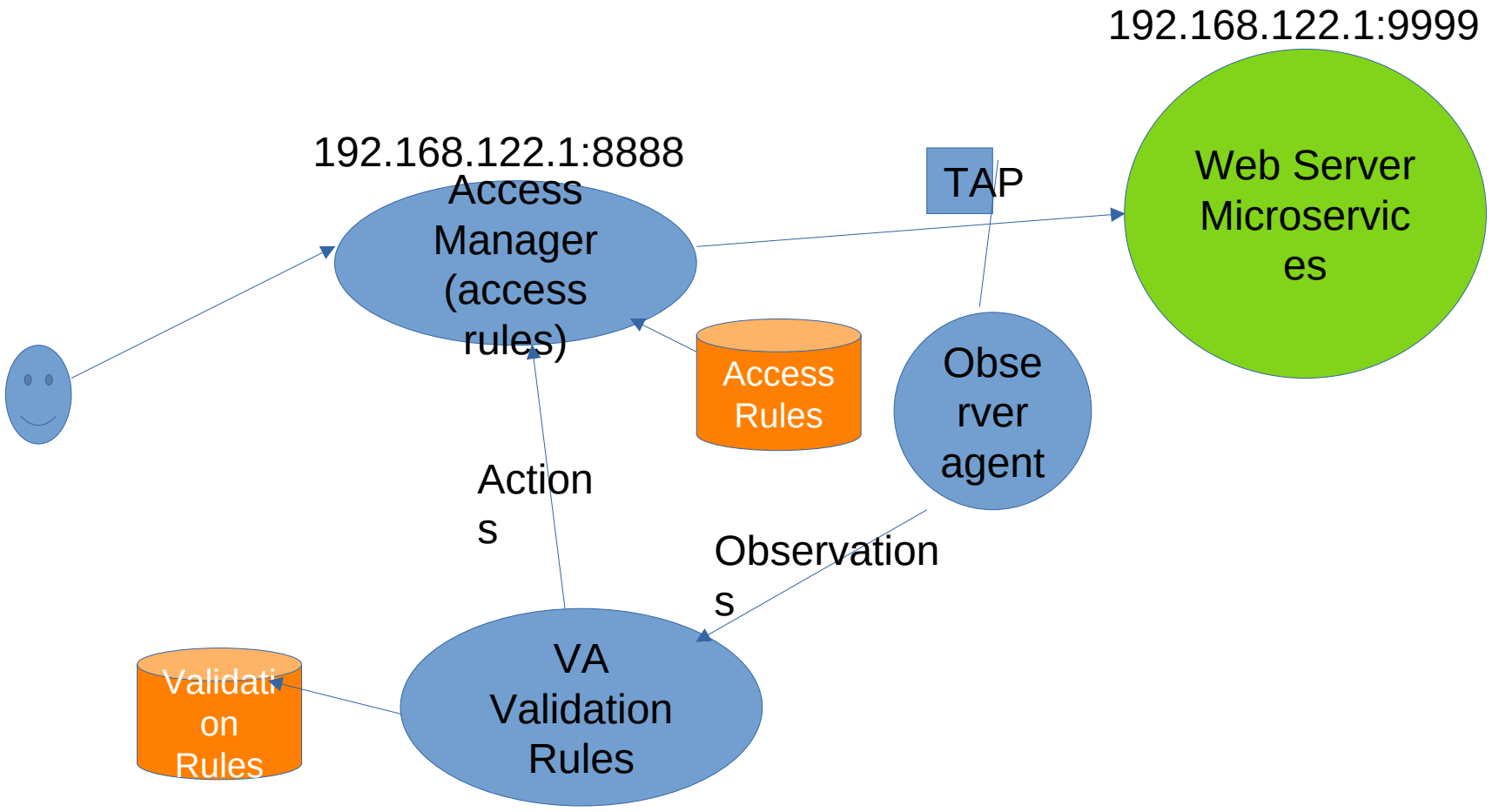
- Operano sia in ambiente VR sia in ambiente AR
- In AR viene passato l’ulteriore parametro PBUF
- Se chiamate nella condition
  - Devono tornare #t/#f per indicare condizione verificata o meno
- Se chiamate nella ACTION, il valore di ritorno non è definito

Alcuni utili operatori/macro

- ((cut + <> <>) 1 2)
- ((compose car cdr) ‘(1 2 3 4 5))
- ((const 10) 234 2434 234234 234)
- Identity
- Leggete il manuale per questi operatori!

esempio

- Memorizzare sia le pagine visitate, sia quante volte una pagina è stata visitata
  - Utilizzare una VR per la memorizzazione
  - Utilizzare una AR per fornire un report al richiedente


192.168.122.1:9999
192.168.122.1:8888
Access Manager (access rules)
TAP
Web Server Microservices
Access Rules
Observer agent
Actions
Observations
Validation Rules
VA Validation Rules

mtfa-fs3

- (define dict (mtfa-fs3-make))
- (mtfa-fs3-add dict "chiave 1" '(1 2 3 "jsdhf "))
- (mtfa-fs3-add dict "chiave 2" 19812391872938179283)
- (null? (mtfa-fs3-get dict "asiduaoiu"))(null? (mtfa-fs3-get dict "asiduaoiu"))
- (mtfa-fs3-get dict "chiave 1")
- (mtfa-fs3-for-each dict (lambda (k v) (Show! k " => " v)))
- Per eliminare una coppia K/V:
  - (mtfa-fs3-add dict "chiave 2" '())
  - In questo modo torna null e quindi risulta vuota se la interrogate per null?

Car e cdr, liste e conses

- (car (cdr '(1 2)))
- (cdr '(1 . 2))

Redirect da LISP

- eis::Redirect (where #:optional headers)
- (eis::Redirect “http://where…../?…..” ‘(“head1: val1” “header2: val2” ...))

Run guile inside docker

- docker run -it -v "$PWD":/vapps dozenapps/va /bin/bash -i

- Ovviamente aggiungendo, ad esempio:
- --network=host, lavorerete nella stessa rete del pc host

Perfect hash

- scheme@(guile-user)> (define ph (mtfa-ph-make))
- scheme@(guile-user)> (mtfa-ph-add ph "k1")
- $9 = 0
- scheme@(guile-user)> (mtfa-ph-add ph "k5")
- $10 = 1
- scheme@(guile-user)> (mtfa-ph-add ph "k9")
- $11 = 2
- scheme@(guile-user)> (mtfa-ph-add ph "k1229")
- $12 = 3
- scheme@(guile-user)> (mtfa-ph-add ph "k12233339")
- $13 = 4
- scheme@(guile-user)> (mtfa-ph-get ph "k12233339")
- $14 = 4
- scheme@(guile-user)> (mtfa-ph-get ph "k1")
- $15 = 0

Esempio di Uso della mtfa-fs3

- Ho un db “statico”
- Al db faccio query del tipo
  - “select password from utente where username = ‘franco’”
- Allora nella struttura fs3 posso memorizzare
  - Key: “select password from utente where username = ‘franco’”
  - Value: <il risultato della query>

```
(define utenti (mtfa-fs3-make))
;;NB: la select e la mysql select hanno una sintassi differente. Qui è solo come esempio
(let ((usr-pwd (mtfa-mysql-select "select utente, password from utenti")))
  (for-each (lambda (it)
              (mtfa-fs3-add utenti
                            (string-append "select password from utenti where username='"
                                           (car it) "'")
                            (cadr it))
            )
            usr-pwd)
)

;;Quando cercherete
;;(mtfa-fs3-get utenti "select password from utenti where username='franco'")
;;=> vi tornerà, se l'avete inserita, la password dell'utente
```