Mobile- and Web-based Software

Lecture 1: Introduction to web technologies

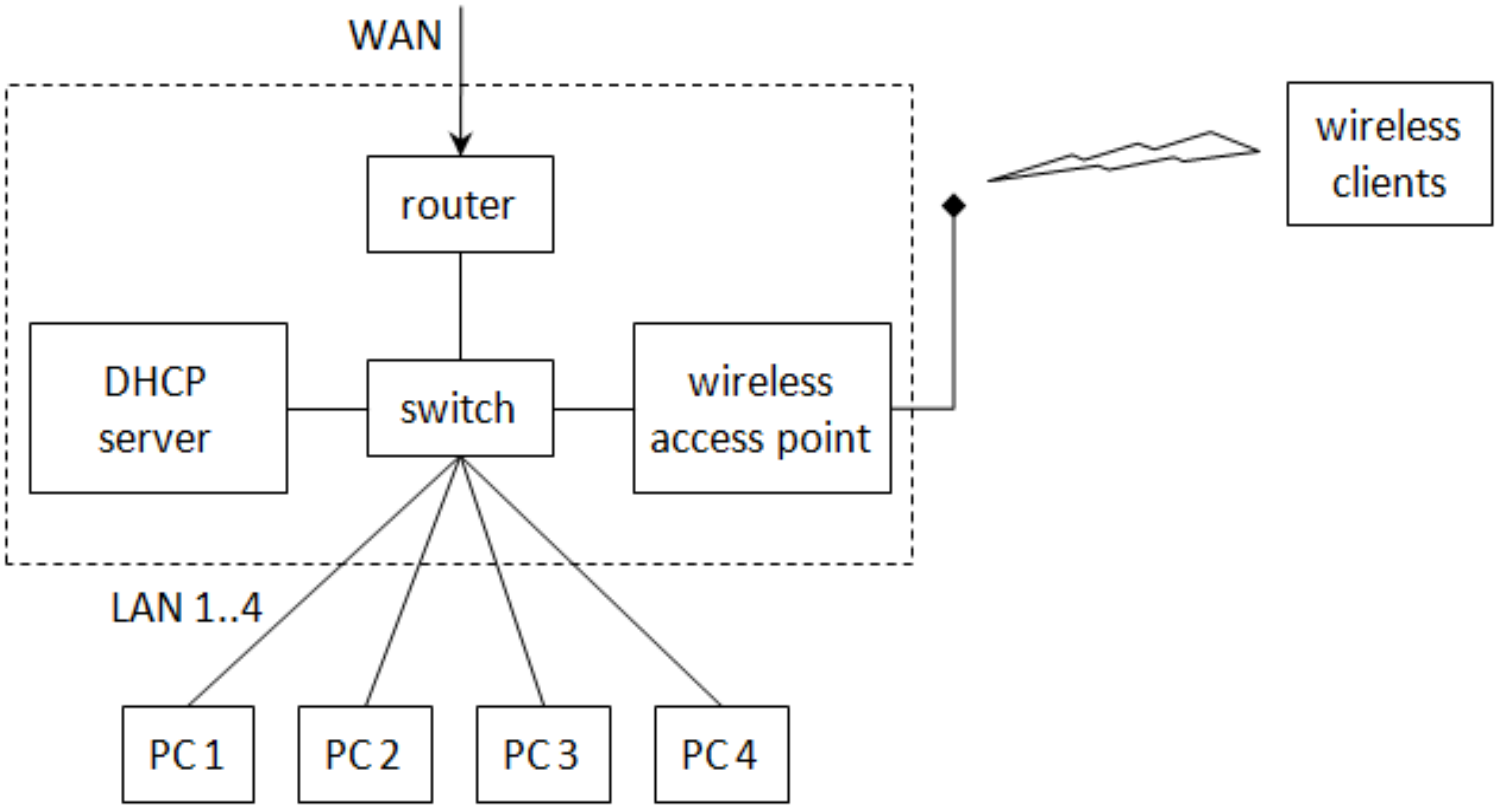
David Sik
david.sik@aut.bme.hu

BME AUT
Automatizálási és
Alkalmazott
Informatikai Tanszék

Networks

Network devices

Network topology



Network devices - Router

- Examines the header of the packages to select the best route for each
- Route selection based on a routing table
  - route print
- Broadcasts and injured packages are filtered out
  - Makes the traffic of a segment local
- Devides the network to subnetworks (subnet)

Route print

VS2012 x86 Native Tools Command Prompt

```
C:\Program Files (x86)\Microsoft Visual Studio 11.0\VC>route print
===========================================================================
Interface List
 21...00 15 5d 51 35 58 ......Hyper-V Virtual Ethernet Adapter #3
 19...54 4f 60 e9 ea 0a ......Check Point Virtual Network Adapter For SSL Networ
k Extender
  9...94 de 80 27 c1 a3 ......Hyper-V Virtual Ethernet Adapter #2
  1...........................Software Loopback Interface 1
  4...00 00 00 00 00 00 00 e0 Microsoft ISATAP Adapter
  5...00 00 00 00 00 00 00 e0 Microsoft 6to4 Adapter
  6...00 00 00 00 00 00 00 e0 Microsoft ISATAP Adapter #2
===========================================================================

IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0     152.66.188.4   152.66.189.215     10
         10.65.0.0      255.255.0.0    152.66.188.52   152.66.189.215     11
         127.0.0.0        255.0.0.0         On-link         127.0.0.1    306
         127.0.0.1  255.255.255.255         On-link         127.0.0.1    306
  127.255.255.255  255.255.255.255         On-link         127.0.0.1    306
      152.66.188.0    255.255.252.0         On-link    152.66.189.215    266
     152.66.188.64  255.255.255.224    152.66.188.95   152.66.189.215     11
    152.66.189.215  255.255.255.255         On-link    152.66.189.215    266
    152.66.191.255  255.255.255.255         On-link    152.66.189.215    266
       169.254.0.0      255.255.0.0         On-link     169.254.80.80    261
     169.254.80.80  255.255.255.255         On-link     169.254.80.80    261
  169.254.255.255  255.255.255.255         On-link     169.254.80.80    261
         224.0.0.0        240.0.0.0         On-link         127.0.0.1    306
         224.0.0.0        240.0.0.0         On-link     169.254.80.80    261
         224.0.0.0        240.0.0.0         On-link    152.66.189.215    266
  255.255.255.255  255.255.255.255         On-link         127.0.0.1    306
  255.255.255.255  255.255.255.255         On-link     169.254.80.80    261
  255.255.255.255  255.255.255.255         On-link    152.66.189.215    266
===========================================================================
Persistent Routes:
  None

IPv6 Route Table
===========================================================================
Active Routes:
 If Metric Network Destination      Gateway
  5   1011 ::/0                     2002:c058:6301::c058:6301
  5   1026 ::/0                     2002:c058:6301::1
  1    306 ::1/128                  On-link
  5   1010 2002::/16                On-link
  5    266 2002:9842:bdd7::9842:bdd7/128
                                    On-link
 21    261 fe80::/64                 On-link
  9    266 fe80::/64                 On-link
  9    266 fe80::605a:d0a0:bb80:b891/128
                                    On-link
 21    261 fe80::89fd:4c4b:30b9:687a/128
                                    On-link
  1    306 ff00::/8                  On-link
 21    261 ff00::/8                  On-link
  9    266 ff00::/8                  On-link
===========================================================================
Persistent Routes:
  None
```

Network devices- Switch

- Among network segments
  - Using the same protocol
  - Even among segments using different physical cabling
  - Uplink port: connecting multiple switches
- Explores the network
  - Rapid Spanning Tree Protocol (RSTP)
- Stores the MAC address of the senders
- It only sends the package to its port where the target of the package is. If it cannot be determined it sends it to every port.
  - Multiple broadcast domains
  - No collisions

Network devices- Switch

- Unmanaged switch
  - Simple plug-and-play
- Managed switch
  - Management console,
  - Security settings
  - Disable ports
  - MAC filtering, network bandwidth limit
  - VLAN (virtual LAN)

Network topology

Identifying network devices

Identifying network devices

- Network Interface Card (NIC)
- Physical address (MAC - Media Access Control)
  - 6 byte long address: 3 bytes to identify the manufacturer + 3 bytes unique identifier
  - E.g. 94-DE-80-27-C1-A3

Querying the MAC address

- ipconfig / all → Physical Address

VS2012 x86 Native Tools Command Prompt

```
Ethernet adapter vEthernet (Qualcomm Atheros AR8161 PCI-E Gigabit Ethernet Contr
oller (NDIS 6.30) Virtual Switch):

   Connection-specific DNS Suffix  . : aut.bme.hu
   Description . . . . . . . . . . . : Hyper-V Virtual Ethernet Adapter #2
   Physical Address. . . . . . . . . : 94-DE-80-27-C1-A3
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::605a:d0a0:bb80:b891%9(Preferred)
   IPv4 Address. . . . . . . . . . . : 152.66.189.215(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.252.0
   Lease Obtained. . . . . . . . . . : 2015. július 10. 10:48:56
   Lease Expires . . . . . . . . . . : 2015. július 10. 22:48:55
   Default Gateway . . . . . . . . . : 152.66.188.4
   DHCP Server . . . . . . . . . . . : 152.66.188.4
   DHCPv6 IAID . . . . . . . . . . . : 160751232
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-1A-A8-E5-D2-94-DE-80-27-C1-A3

   DNS Servers . . . . . . . . . . . : 152.66.188.4
                                       152.66.116.1
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

Why isn't the physical address (MAC) enough?

- Hard to remember, hard to change → not flexible enough.
- Cannot be used to determine if two devices are in the same subnetwork
- Doesn't mean anything (though, it identifies the manufacturer)
- A higher level of abstraction is needed

Solution: IP addresses

- IP address
- Subnet mask
- Default gateway, router

IP address

- Uniquely identifies the hosts
  - Different hosts may sometimes have the same IP address (NLBS, NAT, proxy).
  - One host may have multiple IP addresses
- Static / Dynamic
  - If we don't know the address of a host we cannot communicate with it
- It's got meaning
  - The host can be located, the owner can be determined
  - http://whatismyipaddress.com
    - In Hungary it is regarded to be personal data
    - Hierarchical: from left to right it gets more specific
  - Services to hide the IP address
    - (Tor, anonym proxy).

IP address

- Registered globally
  - Internet Assigned Number Authority (IANA)
  - Regional Internet Registries (RIRs):
    - AFRINIC, APNIC, ARIN, LACNIC, RIPE NCC
  - Internet Service Providers (ISPs)
- Addresses and address ranges can be bought

Class	First address	Last address	Number of subnets	Number of hosts
A	0.0.0.0	127.255.255.255	128 (2^7)	16 777 216 (2^{24})
B	128.0.0.0	191.255.255.255	16 384 (2^{14})	65 536 (2^{16})
C	192.0.0.0	223.255.255.255	2 097 152 (2^{21})	256 (2^8)

  - 4.3 billion IP addresses, it is sold in ranges
  - Running out of addresses: untill September 2011.
    - classless IP addressing

Versions

- IPv4
  - 32 bit long addresses (2^{32} = 4.294.967.296 addresses)
- IPv5
  - Internet Streaming Protocol
- IPv6
  - 128 bit long addresses
  - $2^{128} = \approx 340 \times 10^{36}$ addresses
    - 340.282.366.920.938.463.463.374.607.431.768.211.456

IP address

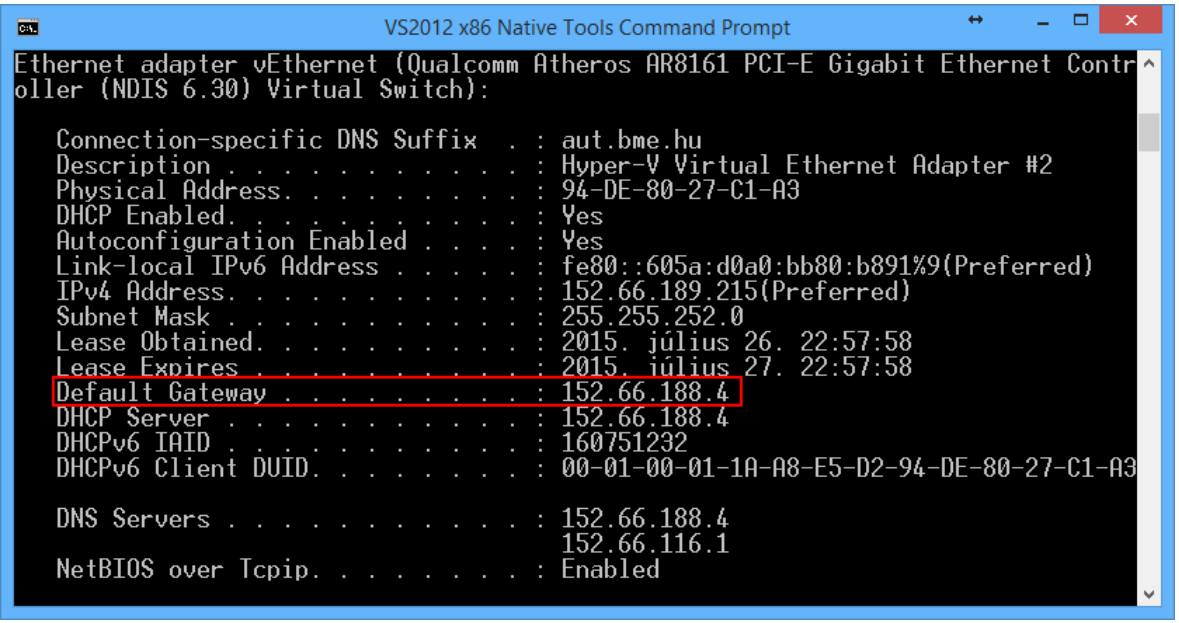

NSC NATIONAL SECURITY CONTRACTORS
CONNECT
DISCONNECT
USERNAME:
PASSWORD:
OVERRIDE [ADMIN.ACCESS] DISR.POS=MY.POS [**_**_**]
RUN SCRIPT
SECURE PORT - XXX_XXXXXX
[NEW IP ADDRESS]
93.455.43.232
78.397.17.610

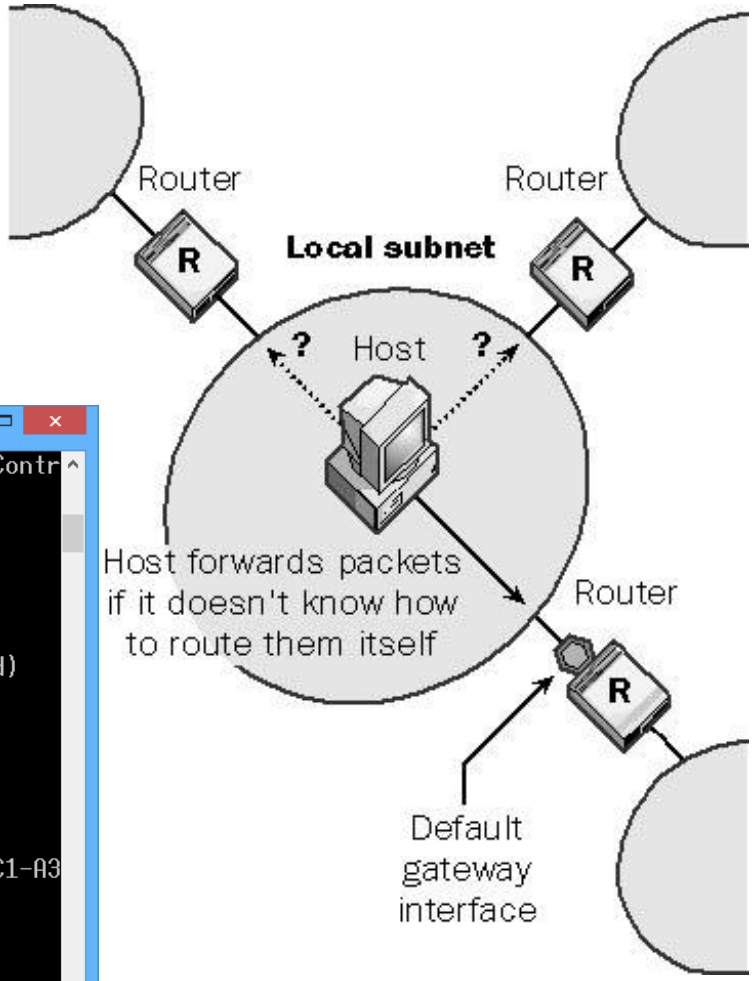
Network address (subnet mask)

- 32 bit long address devided to 4 octets
- The sender uses the IP address of the target to determine if the package has to be sent directly to the target host or to the router
  - Performing a bitwise AND operation on the IP address and on the subnet mask is used to find out the Network ID (IP) and the Host ID (IP)
  - Hosts with the same Network ID are in the same network
- Hosts with different Network ID addresses can communicate with each other through routers

Default gateway

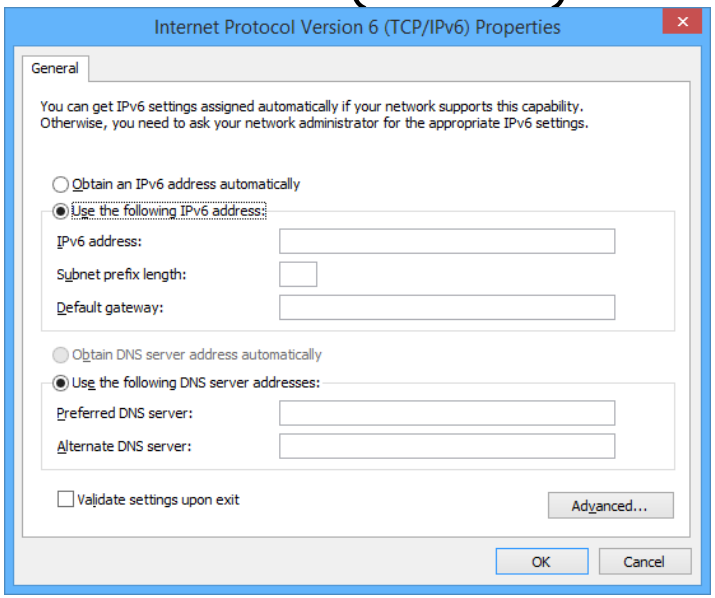
- Connects a subnet with other subnets

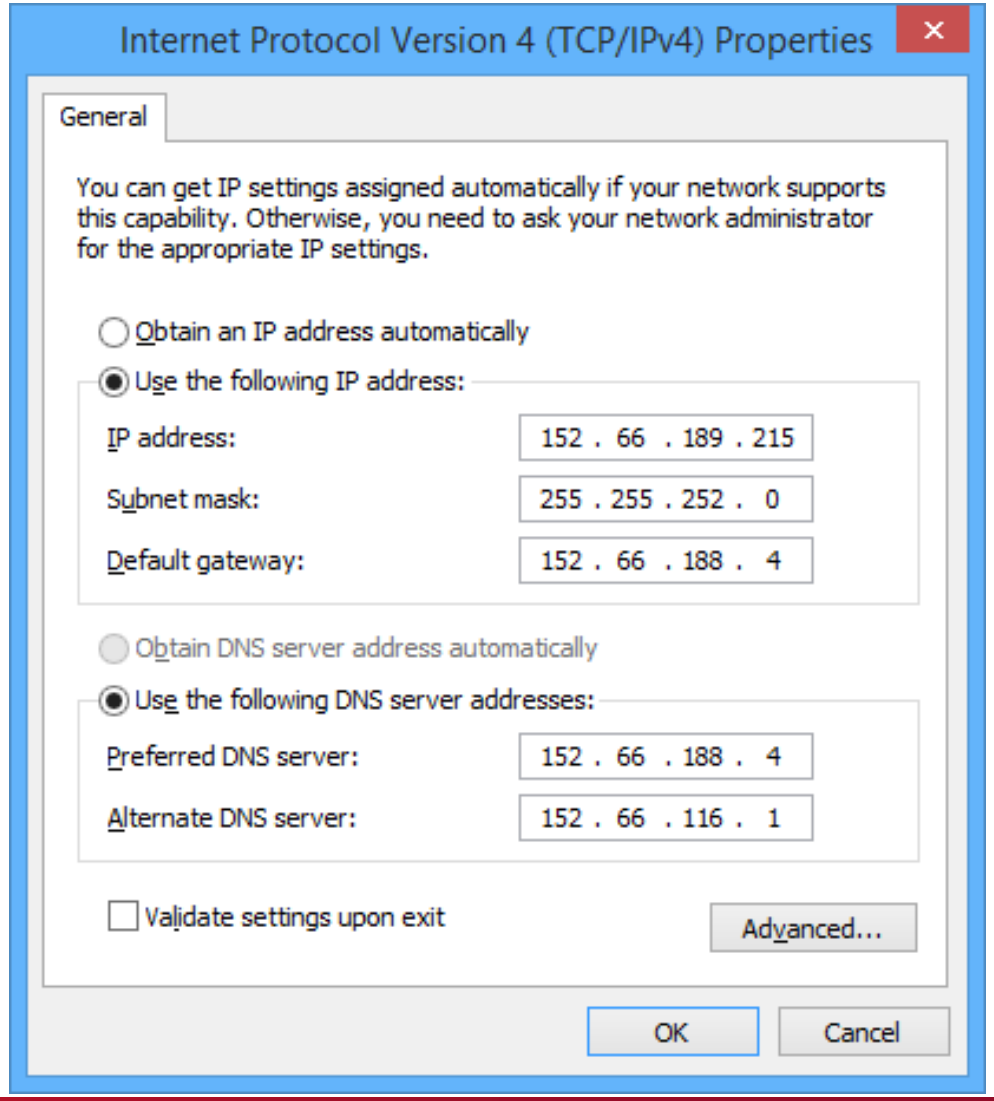




Manual IP address configuration

- Has to be done on each host
  - Used on servers
  - Independent of other servers (DHCP)





Automatic IP address configuration (DHCP)

Dynamic Host Configuration Protocol (DHCP)

- Broadcast based solution→ routers filter it
- Types:
  - **Static allocation**: based on the MAC address
  - **Dynamic allocation**: specifying an IP address pool
  - **Automatic allocation**: specifying an IP address pool, the server tries to give the same IP to the same host

Automatic IP address configuration (DHCP)

- Lease time: the address is valid in this period
  - The host tries to renew it before expiration
    - `ipconfig /release` or `ipconfig /release6`
    - `ipconfig /renew` or `ipconfig /renew6`
- No authorization!
  - Unauthorized (rogue) DHCP server
    - Intentionally or unintentionally
  - It gives IP address to everyone in wireless network
  - Hosts may manually override the IP address obtained from the DHCP server

Automatic IP configuration without DHCP

- Microsoft: **Automatic Private IP Addressing** (APIPA)
- When no DHCP server is available is chooses an address
  - From range 169.254.0.1-169.254.255.254
  - zero configuration
- Configures no default gateway → only works in a subnet, hosts cannot communicate outside of the subnet

Debugging

- Is TCP/IP stack working on my computer?
  - `ping localhost`
- Do I have network access?
  - `ipconfig /all`
- Can I access the routert?
  - `ping routeraddress`
- Do I have internet connection?
  - `ping www.aut.bme.hu`

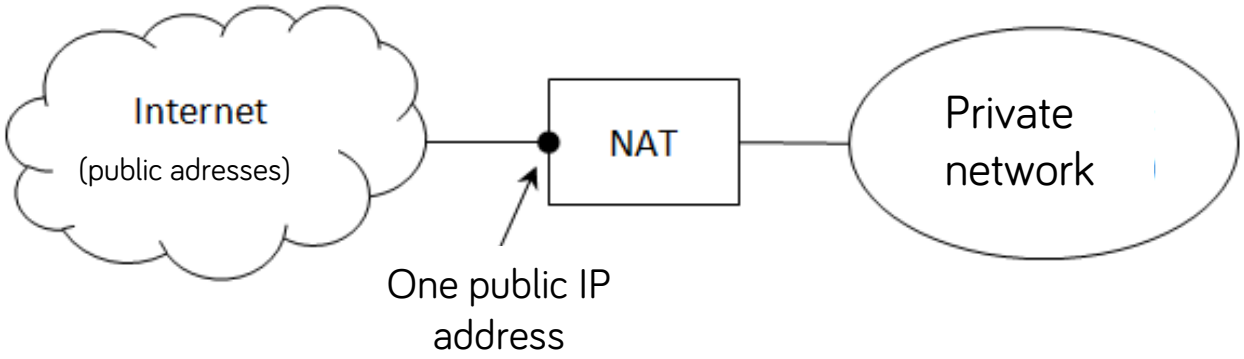
Private IP ranges

- Cannot be accessed in a public network
- **Automatic Private IP Addressing (APIPA):**
  - 169.254.0.0 /16 (169.254.0.0 – 169.254.255.255)
- **CIDR (Classless Inter-Domain Routing) notation**: the number of „1” bits in the subnet mask

Class	CIDR notation	First and last address
1 db A	10.0.0.0 /8	10.0.0.0 – 10.255.255.255
16 db B	172.16.0.0 /12	172.16.0.0 – 172.31.255.255
256 db C	192.168.0.0 /16	192.168.0.0 – 192.168.255.255

Network Address Translation (NAT)

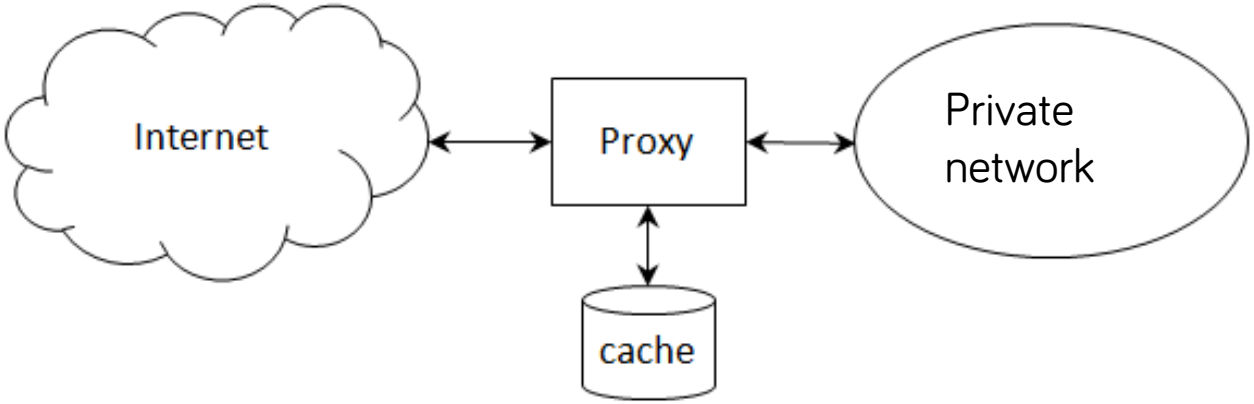
- **Address translation**: hosts in the internal network with private IP addresses can communicate through a single public IP address (e.g. SoHo cable model).



- Hosts behind the public IP address cannot be accessed from the public network (e.g. Internet Connection Sharing).

Proxy server

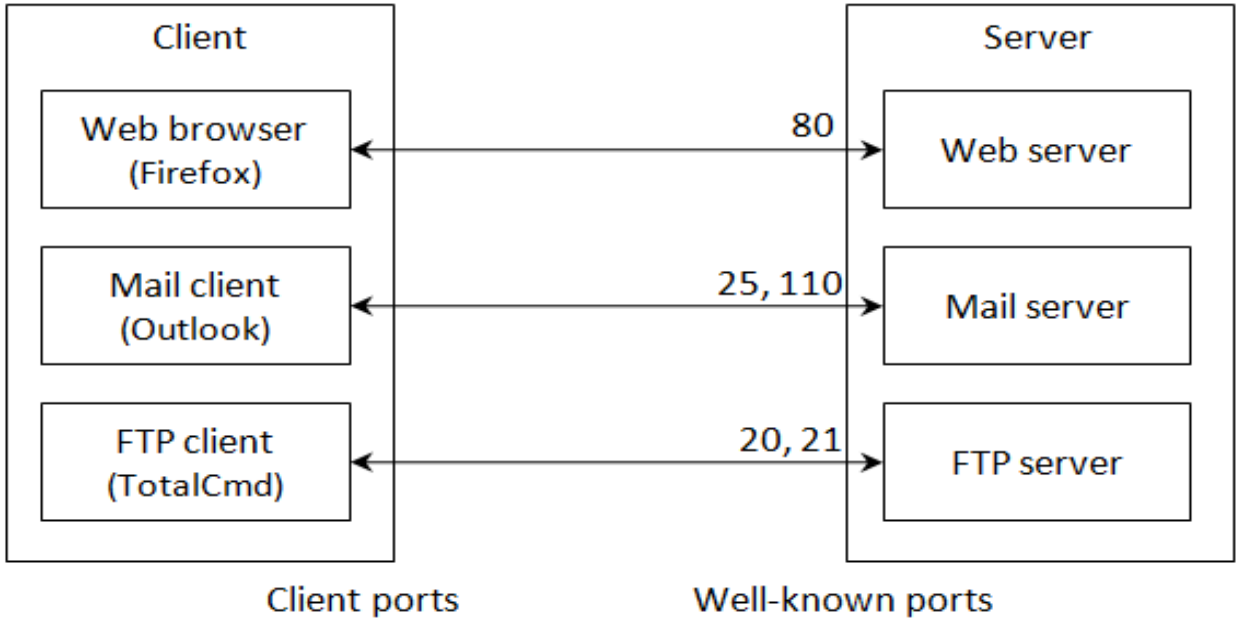
- A server application that forwards the client's requests to other servers
- Proxy hides the clients from the public
  - Anonym proxy: this is actually the proxy that downloads the requested content, intentionally hides the client's IP address
    - E.g. The Onion Router (Tor) http://www.torproject.org
- **Cache:** caches already downloaded data → speed, bandwidth



Connecting applications

- **Socket**: IP address + TCP/UDP port
  - „transport endpoint”
  - The operating system binds them to applications (binding)
- Port
  - Number between 0 – 65535
  - inbound, server port
  - outbound, client port
- Inbound ports:
  - 1-1023: well-known ports
  - 1024-49151: registered ports
  - 49152-65535: dynamic (private) ports

Ports



Services	Mailing	Administration	Don't allow on firewalls
• HTTP: 80 • HTTPS: 443 • FTP: 20, 21 • DNS: 53	• SMTP: 25 • SMTPS: 465 • POP3: 110 • POP3S: 995 • IMAP4: 143 • IMAP4S: 993	• SSH: 22 • RDP: 3389 • PPTP: 1723 • Telnet: 23	• NetBIOS: 137, 138, 139, 445 • LDAP: 389 • MS SQL: 1433, 1434

C:\Windows\System32\drivers\etc\services

```
# Copyright (c) 1993-2004 Microsoft Corp.
# This file contains port numbers for well-known services defined by IANA
# Format:
# <service name>  <port number>/<protocol>  [aliases...]   [#<comment>]

echo                7/tcp
echo                7/udp
discard             9/tcp    sink null
discard             9/udp    sink null
systat             11/tcp    users                   #Active users
systat             11/udp    users                   #Active users
daytime            13/tcp
daytime            13/udp
qotd               17/tcp    quote                   #Quote of the day
qotd               17/udp    quote                   #Quote of the day
chargen            19/tcp    ttytst source           #Character generator
chargen            19/udp    ttytst source           #Character generator
ftp-data           20/tcp                            #FTP, data
ftp                21/tcp                            #FTP. control
ssh                22/tcp                            #SSH Remote Login Protocol
```

netstat –a

VS2012 x86 Native Tools Command Prompt

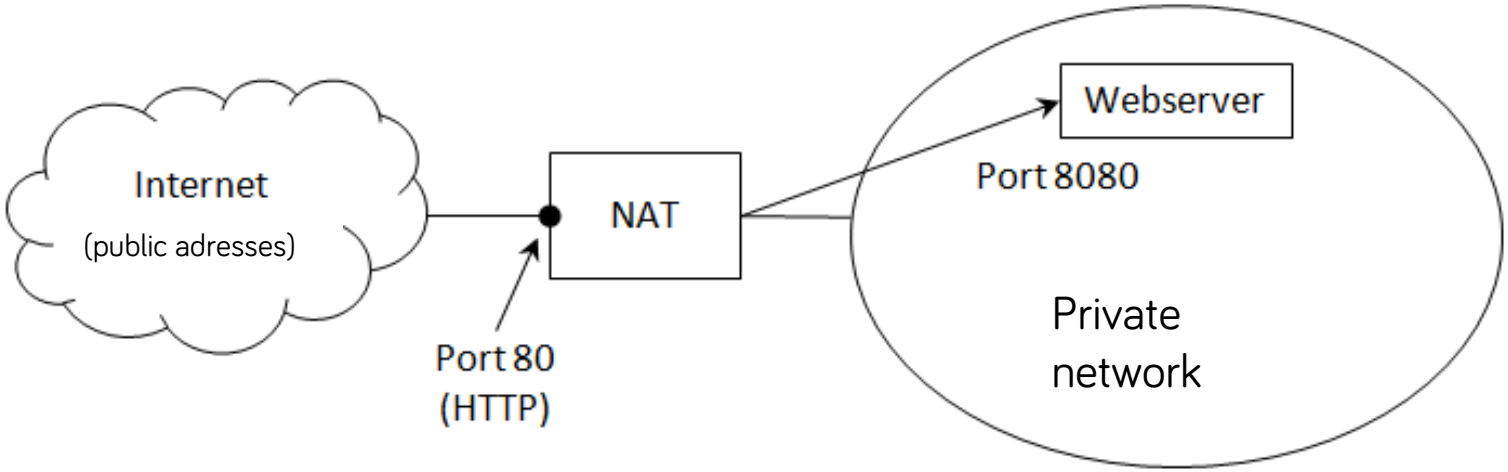
```
C:\Program Files (x86)\Microsoft Visual Studio 11.0\VC>netstat -a

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:80             GinaPC:0               LISTENING
  TCP    0.0.0.0:135            GinaPC:0               LISTENING
  TCP    0.0.0.0:445            GinaPC:0               LISTENING
  TCP    0.0.0.0:1025           GinaPC:0               LISTENING
  TCP    0.0.0.0:1026           GinaPC:0               LISTENING
  TCP    0.0.0.0:1027           GinaPC:0               LISTENING
  TCP    0.0.0.0:1028           GinaPC:0               LISTENING
  TCP    0.0.0.0:1029           GinaPC:0               LISTENING
  TCP    0.0.0.0:1071           GinaPC:0               LISTENING
  TCP    0.0.0.0:1093           GinaPC:0               LISTENING
  TCP    0.0.0.0:2179           GinaPC:0               LISTENING
  TCP    0.0.0.0:2383           GinaPC:0               LISTENING
  TCP    0.0.0.0:3389           GinaPC:0               LISTENING
  TCP    0.0.0.0:17500          GinaPC:0               LISTENING
  TCP    127.0.0.1:1043         GinaPC:5354            ESTABLISHED
  TCP    127.0.0.1:1044         GinaPC:5354            ESTABLISHED
  TCP    127.0.0.1:1045         GinaPC:1046            ESTABLISHED
  TCP    127.0.0.1:1046         GinaPC:1045            ESTABLISHED
  TCP    127.0.0.1:1135         GinaPC:5939            ESTABLISHED
  TCP    127.0.0.1:1181         GinaPC:27015           ESTABLISHED
  TCP    127.0.0.1:1213         GinaPC:0               LISTENING
  TCP    127.0.0.1:1434         GinaPC:0               LISTENING
  TCP    127.0.0.1:2467         GinaPC:19872           ESTABLISHED
  TCP    127.0.0.1:2473         GinaPC:2474            ESTABLISHED
  TCP    127.0.0.1:2474         GinaPC:2473            ESTABLISHED
  TCP    127.0.0.1:5354         GinaPC:0               LISTENING
  TCP    127.0.0.1:5354         GinaPC:1043            ESTABLISHED
  TCP    127.0.0.1:5354         GinaPC:1044            ESTABLISHED
```

Port forwarding

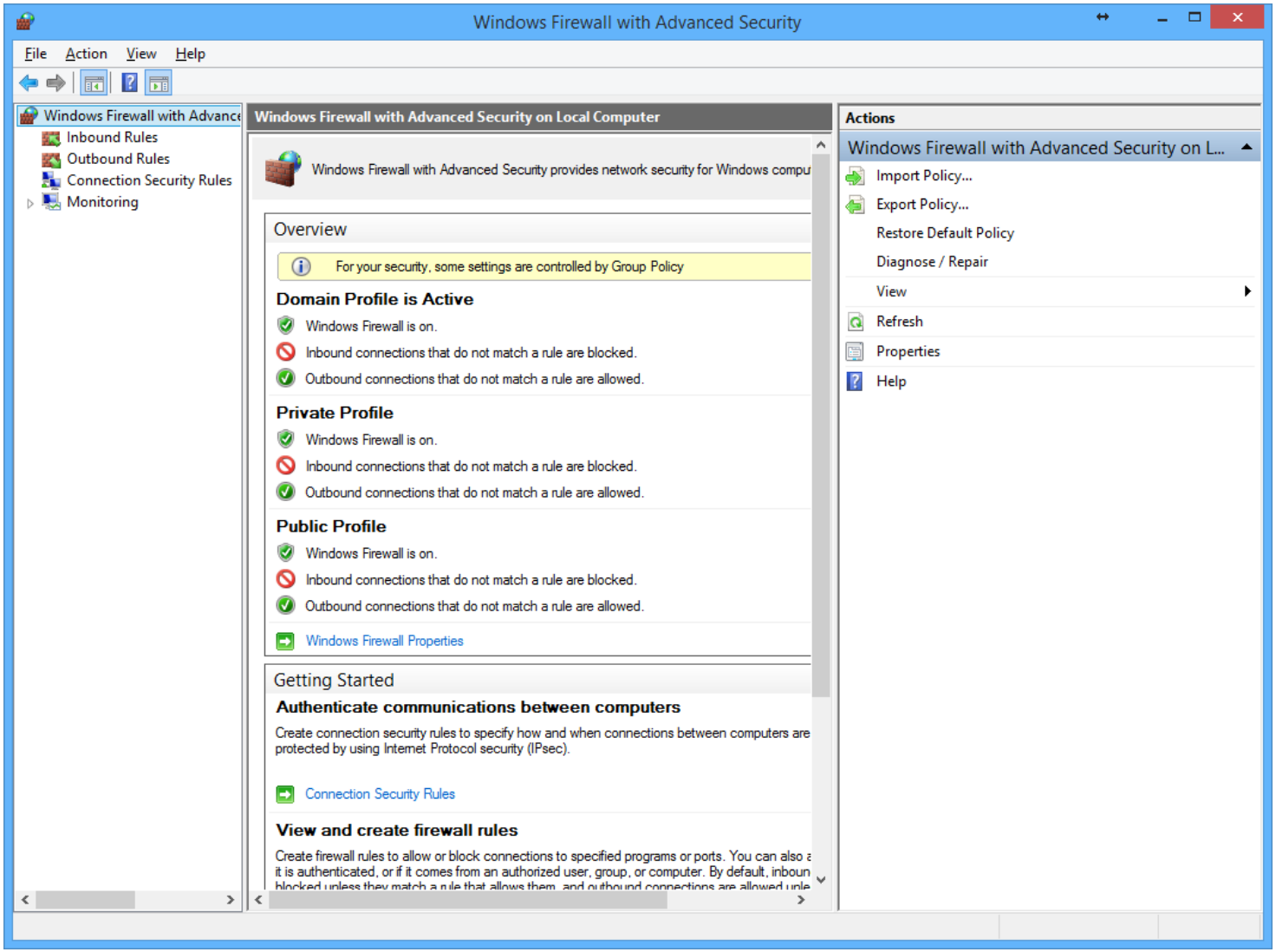
- Configuring the gateway to send all packets received on a particular port to a specific machine on the internal network
- E.g. hosting a webserver in an internal subnet



Firewall

- Software or hardware component to filter out unauthorized traffic
- Packet filtering:
  - Basic service of each firewall
  - Rule set to allow or deny packages
  - Enable (open) or disable (close) ports
- Inbound and outbound traffic

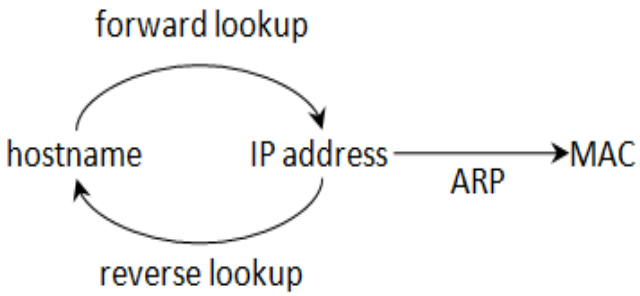
Firewall configuration



How is the Internet working?

Address resolution

Address resolution



- Friendly name→ IP address
- Alias (e.g. neptun)
- **Fully Qualified Domain Name (FQDN)**
  - E.g neptun.bme.hu
  - Labels delimited by points (label, 1-63 chars)
  - Theoretically it starts with a point but we don't write it
  - Max. 255 chars
  - It contains the **DNS suffix** (e.g. bme.hu)
    - FQDN can be determined based on the Alias
    - Easier for the user (e.g. http://intranet)

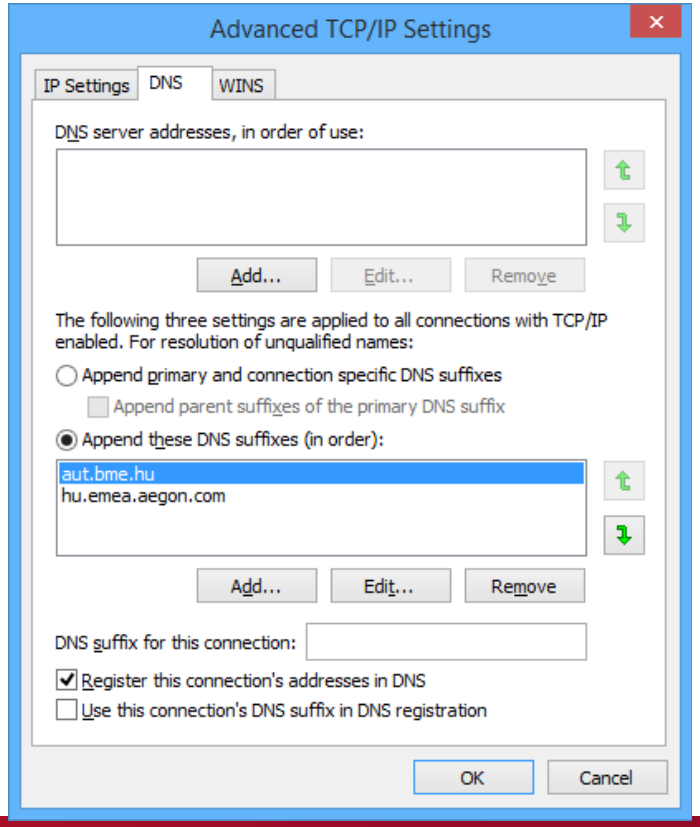
Address resolution - configurations

- This PC → Properties

Computer name, domain, and workgroup settings

Computer name:	GinaPC
Full computer name:	GinaPC.aut.bme.hu
Computer description:	
Domain:	aut.bme.hu

- Network and Sharing Center
  - > Change adapter settings
  - > Properties
  - > TCP/IPv4 Properties
  - > Advanced: DNS suffix



Address resolution- nslookup

```
nslookup 217.20.130.99
```

VS2012 x86 Native Tools Command Prompt

```
C:\Program Files (x86)\Microsoft Visual Studio 11.0\VC>nslookup 217.20.130.99
Server:  ns3.aut.bme.hu
Address:  152.66.188.4

Name:    index.hu
Address:  217.20.130.99


C:\Program Files (x86)\Microsoft Visual Studio 11.0\VC>
```

Static resolution

- Name – IP pairs stored locally on the computer
  - C:\Windows\System32\drivers\etc\hosts file.

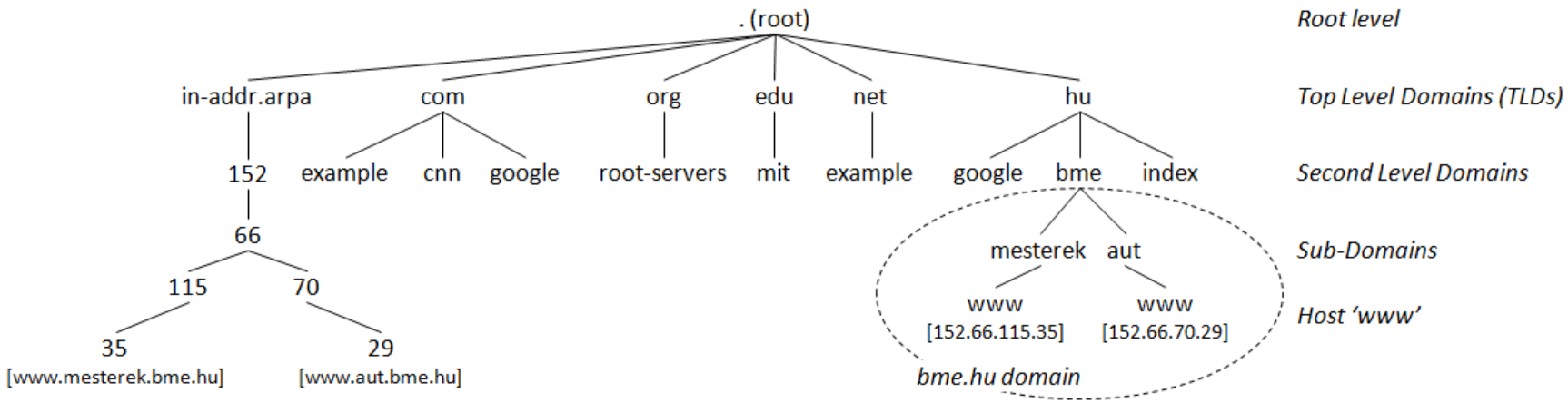
```
# For example:
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host
# localhost name resolution is handled within DNS itself.
#      127.0.0.1        localhost
#      ::1              localhost
10.65.240.95 test1
10.65.241.17 budstfs102
```

- Google: hosts file → mvps.org: Blocking unwanted parasites with a hosts file (http://winhelp2002.mvps.org/hosts.txt)

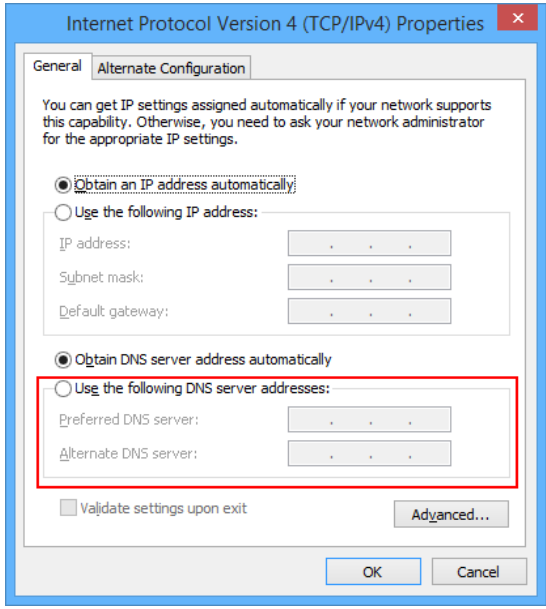
Domain Name System (DNS)

- Largest distributed live database
- TCP/UDP 53 port
- Hierarchical naming system
  - . – **root domain**
  - .gov, .mil, . edu, .com, .org, .net – **generic top level domain** (TLD, 20 ones)
  - .hu, .uk, .at – **country-code top level domain** (ccTLD, 248 ones)
  - .co.hu, .edu.at, .bme.hu – **second level domain**

Domain Name System (DNS)

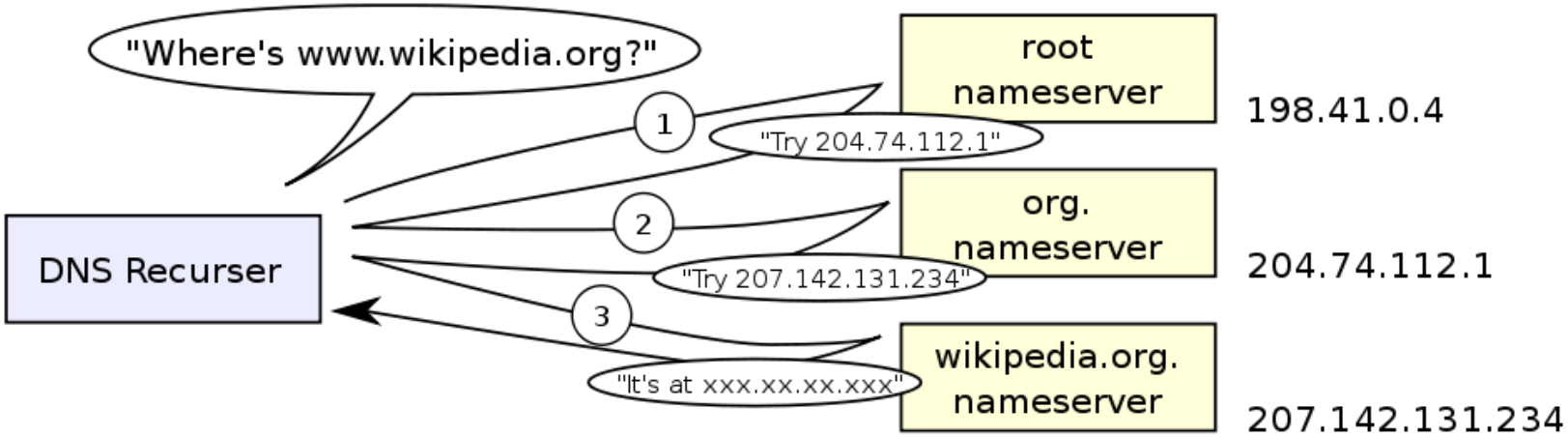


- Accentuated letters are also allowed
  - > www.magyarország.hu
- Network adapter properties
  - > IPv4 Properties →



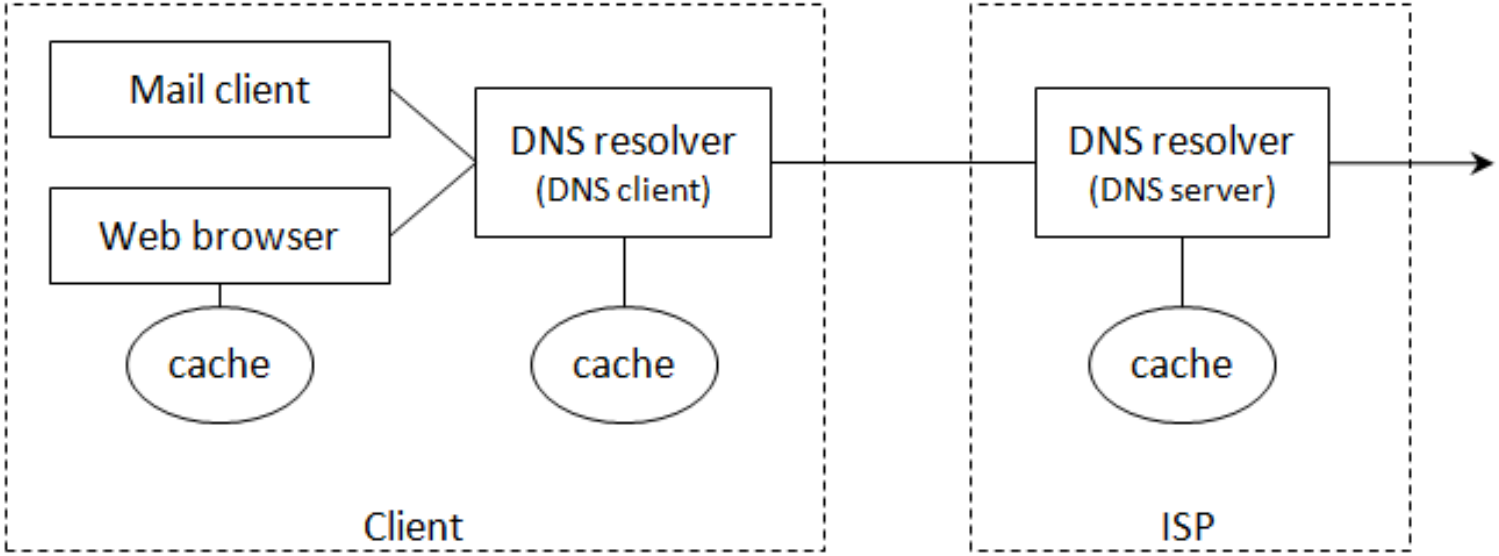
Root name servers

- http://root-servers.org/
- Geographically distributed (performance)
- Replications (reliability)
- Distributed Denial of Service (DDos) attacks
  - 2002., 2007.



DNS resolver cache

- DNS lookup takes time → cache
- **Time-To-Live (TTL):** expiration timeout



Local DNS cache

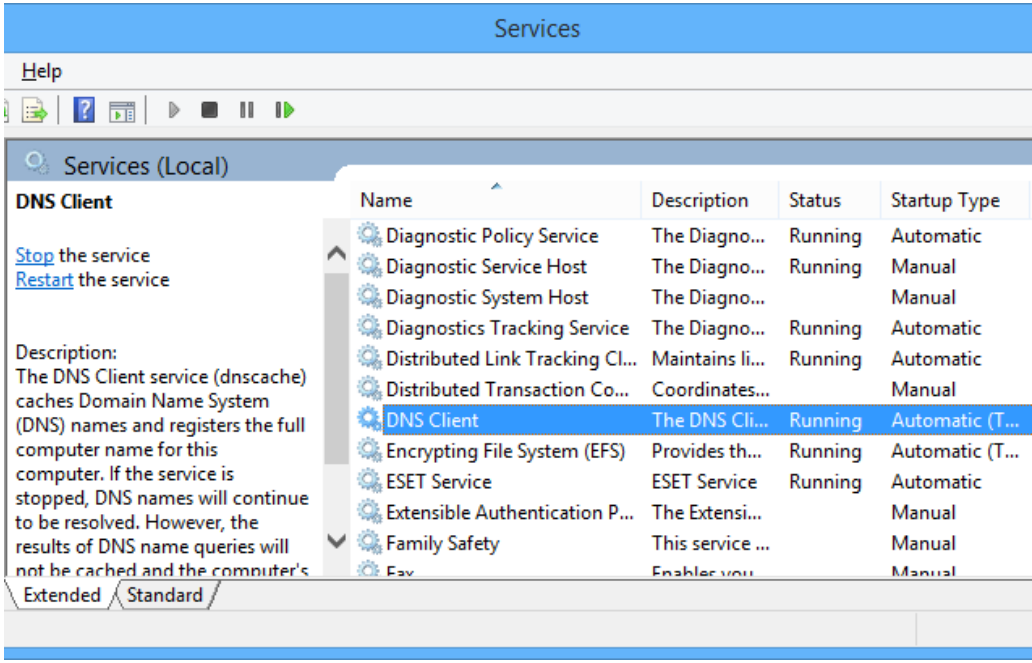
- Listázás: **ipconfig /displaydns**
- Törlés: **ipconfig /flushdns**

```
index.hu
    ----------------------------------------
    Record Name . . . . . : index.hu
    Record Type . . . . . : 1
    Time To Live  . . . . : 62
    Data Length . . . . . : 4
    Section . . . . . . . : Answer
    A (Host) Record . . . : 217.20.130.99


    Record Name . . . . . : ns.index.hu
    Record Type . . . . . : 1
    Time To Live  . . . . : 62
    Data Length . . . . . : 4
    Section . . . . . . . : Additional
    A (Host) Record . . . : 195.56.65.172


    Record Name . . . . . : ns.inventra.hu
    Record Type . . . . . : 1
    Time To Live  . . . . : 62
    Data Length . . . . . : 4
    Section . . . . . . . : Additional
    A (Host) Record . . . : 217.20.130.10


    index.hu
    ----------------------------------------
    No records of type AAAA
```



HyperText Transfer Protocol

Request-response

- It is always the client that initiates the communication, the server only responds (pull model)
- **User agent**: the identifier of the client, any application that can send HTTP requests
  - E.g. web browsers, RSS readers, mobile clients
- HTML5 websockets: push model

Connectionless

- After responding the server closes the connection
- HTTP 1.0: the socket connection is closed by default, unless a **Connection: Keep-Alive** header is received
- HTTP 1.1: the socket connection stays open by default, unless a **Connection: Close** header is received.

Stateless

- State is not preserved between requests
- HTTP itself cannot create user sessions
  - **Session**: all requests-responses between the first and last request of a user
    - A session doesn't necessarily require a login
    - Has time-out, e.g. 20 mins sliding timeout
  - E.g. cookie, hidden field, URL parameter

Structure of the request and the response

- General format of the request

```
Method RequestURI HTTP-Version <CR><LF>
header <CR><LF>
<CR><LF>
body
```

- General format of the response

```
HTTP-Version Status-Code Reason-Phrase <CR><LF>
header <CR><LF>
<CR><LF>
body
```

Example request

```
GET http://www.example.com/ HTTP/1.1
Accept: text/html, application/xhtml+xml, */*
Accept-Language: en-US,hu-HU;q=0.5
User-Agent: Mozilla/5.0 (compatible; MSIE 10.0; Windows
NT 6.1; WOW64; Trident/6.0)
Accept-Encoding: gzip, deflate
Host: www.example.com
DNT: 1
Connection: Keep-Alive
Pragma: no-cache
```

Example response

```
HTTP/1.1 200 OK
Accept-Ranges: bytes
Cache-Control: max-age=604800
Content-Type: text/html
Date: Wed, 21 Aug 2013 07:49:51 GMT
Etag: "3012602696"
Expires: Wed, 28 Aug 2013 07:49:51 GMT
Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT
Server: ECS (iad/1984)
X-Cache: HIT
x-ec-custom-error: 1
Content-Length: 1270
```

Parts of the request and the response

- Methods, verbs
- The requested resource
- Headers
- Status-Code
- Reason-Phrase

Methodes

- **GET**: download the requested resource from the server
- **POST**: submits data to the server in the body of the request
  - E.g. the content of the form
- **HEAD**: requests meta information about a resource
  - E.g. size, type, last modified
- **OPTIONS**: returns the HTTP methods supported by the server
- **DELETE**: deletes the given resource
- **TRACE**: echo's back the input (debug)
- **PUT**: submits the given resource

POST vs PUT

- PUT:
  - You need to know the exact UTL you are updating
  - You need full representation of the resource on the server
  - If it exists, it will update it. If not, it will create a new one
  - The update is done by replacing the source, that is why you need the whole source
- POST:
  - You don't need to know the exact URL
  - The server creates a new url for you, and send it back to you
  - Any processing or data is ok (with PUT it should be exactly as the existing)

How to Choose between PUT and POST

- **Use PUT when:**
  - The client is responsible for determining the URI of the new or updated resource.
  - You are replacing an entire resource or creating a resource with a client-defined identifier.
  - The operation needs to be idempotent, ensuring that repeated requests will have the same effect as a single request.
- **Use POST when:**
  - The server is responsible for assigning a new unique identifier for the created resource.
  - The operation does not have to be idempotent, or it involves creating new resources.
  - The action performed is complex or does not fit neatly into the CRUD model.

Code examples

- Imagine a flight tracking system with which users look up the status of American Airlines flight 123 at the following URL:

```
www.example.com/AA123
```

- To update flight AA123's status, a web service performs a PUT operation to that URL.
- The PUT operation includes a JSON or XML payload that fully describes the new status:

```
PUT URL: www.example.com/flights/AA123
PAYLOAD: {"status":"ontime", "gate":"b12"}
```

- Now, imagine we need to create a new entry for Air Canada flight 789.
- A URL that describes this flight does not exist yet, but we know that after creation the URL will be:

```
www.example.com/AC789
```

- Since we know the desired URL of the resource before requesting its creation, we must use a PUT operation.

```
PUT URL: www.example.com/flights/AC789
PAYLOAD: {"status":"late", "gate":"c17"}
```

HTTP POST method example

- to create a new customer

```
POST URL: www.example.com/customers
PAYLOAD: {"name":"Joe", "age":"29", "city":"ajax"}
```

- a possible URL generated by a database-driven application for this new customer:

```
www.example.com/customers/j567
```

- This is why the difference between a PUT and POST operation is often phrased as: To create an object, use a POST. PUT should be used for updates.

Particularities of methods

- **Safe methods**
  - Used only to download content, no side effect, doesn't change the state of the server
    - E.g. GET, HEAD, OPTIONS, TRACE.
  - Clients may retry
- **Idempotent methods**
  - The repeated execution has got the same effect as doing it just once (e.g. PUT, DELETE)
  - Safe methods are idempotent at the same time
- **POST is not idempotent** (e.g. posting a forum comment)
    - POST-Redirect-GET (PRG) pattern.

The requested resource

- **Uniform Resource Identifier (URI)**
- *„A Uniform Resource Identifier (URI) is a compact sequence of characters that identifies an abstract or physical resource."* (RFC 3986, 61.old)
- **[uri_scheme]:[scheme specific part]**
  - tel:+36 1 4633714
  - mailto:John.Doe@example.com
  - http://www.bme.hu
- Use „URI" instead of „URL"

Uniform Resource Locator (URL) RFC 3986

- Special URI for web pages
- Determines the location of the resource

```
  foo://example.com:8042/over/there?name=ferret#nose
  \_/   \______________/\_________/ \_________/ \__/
   |           |            |           |        |
scheme     authority       path        query   fragment
   |   _____________________|__
  / \ /                       \
  urn:example:animal:ferret:nose
```

- The server wont receive the fragment part, that is only used by the client
- General format:

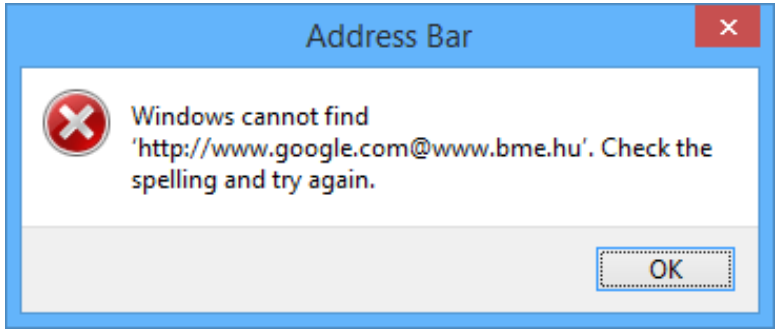
protocol://username:password@FQDN:port/path/file
?variable1=value1&variable2=value2#name

Phishing attack

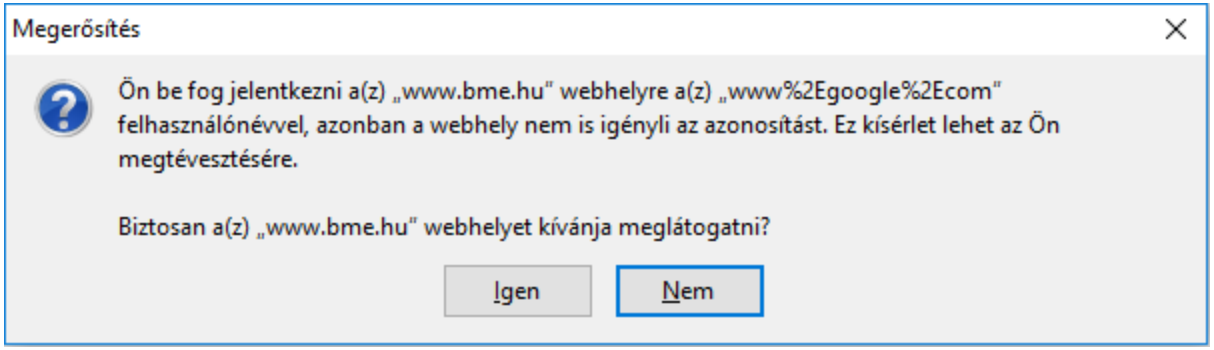
- http://www.otpbank.hu.example.com
- http://www.google.com@www.bme.hu

http://www.google.com@www.bme.hu

- Chrome loads www.bme.hu
- Internet Explorer



- FireFox



Types of URL

- It can be absolute or relative
  - **Absolute URL**: can be used in itself
    - http://www.bme.hu/hirek
  - **Relative URL**: compared to the actual document or to the root of the server (root relative)
    - /Oktatas/Lists/Szakiranyok
    - Image%20Library/BulletinImage.jpg
- Usually case-sensitive
  - Depends on server settings and encoding

Headers (RFC 2616 Section 14)

- **Server** related headers
  - `Date: Wed, 21 Aug 2013 08:41:30 GMT`
  - `Server: Apache`
- **Content** related headers
  - `Accept: text/html, image/jpeg`
  - `Accept-Encoding: gzip, deflate`
  - `Accept-Language: en-US,hu-HU;q=0.5`
  - `Content-Length: 3495`
  - `Content-Type: text/html`
  - `Content-Disposition: file name to save`
  - `Content-Encoding: gzip`

Headers (RFC 2616 Section 14)

- **Caching** related headers
  - `Cache-Control: no-cache`
  - `Expires: date`
  - `If-Modified-Since: date`
  - `Last-Modified: date`
  - `ETag: version`
- **Security** related headers
  - `Authorization: Basic` TXlEb21haW5cTXlDb21wdXRlcjpTdXBlclNlY3JldFBhc3N3b3Jk
  - `WWW-Authenticate: Basic realm="MyComputer"`
  - `X-Frame-Options: SAMEORIGIN`
  - `DNT: 1`

Headers (RFC 2616 Section 14)

- **Other** :
  - Referer: http://www.google.com/?k=szó
    - Properly it would be "referrer" but was mistyped in the specification
  - User-Agent: Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1;+SV1)
    - **User agent sniffing**: displaying different content for different devices
    - **User agent spoofing**: supplying an untrue User-Agent

Status-Codes RFC 2616 Sec. 10

- Full list: http://support.microsoft.com/kb/943891
- **1xx: Information**
  - 100 Continue
  - 101 Switching protocols (ld. WebSocket)
- **2xx: Successful**
  - 200 OK
  - 201 Created (ld. REST)
  - 204 No content
- **3xx: Redirect**
  - 301 Moved permanently
  - 302 Found (temporary move)
  - 304 Not modified

Status-Codes RFC 2616 Sec. 10

- **4xx: Client Error**
  - 400 Bad request
  - 401 Unauthorized
  - 403 Forbidden
    - 403.5: SSL required
    - 403.6: Forbidden: IP address rejected
  - 404 Not found
  - 405 Method not allowed
  - 410 Gone
  - 413 Request entity too large
  - 414 Request URI too long
- **5xx: Server Error**
  - 500 Internal server error
  - 503 Service unavailable

Reason-Phrase

- Recommendations in the standard
- The server can send back a custom error page
- The browser can display a friendly error message
  - Internet Options → Advanced → Browsing → Show friendly HTTP error messages
  - IIS Manager: Error Pages

State management

HTTP is stateless, however web applications need to handle user sessions and store state on the server

Problem

- HTTP is stateless
  - State is not preserved between HTTP calls
  - No user session is created
- Why do we need sessions a state preservation?
  - It is enough to log in once to a web app
  - Webshop can store the items in the cart
  - User profile settings are persisted
  - „memory for websites”

Solutions (client)

All session related data always travels between the client and the server

- Advantage: no server load
  - Scales well when there are many users
- Disadvantages:
  - Limited data size
    - Amount of data doesn't scale well
  - Data travels with each request/response
    - Bandwidth penalty
  - Data is visible for MITM attackers
    - Not safe

Solutions (client and server)

Session related data is stored on the server. Only the session ID is traveling between the browser and the server

- Advantages: disadvantages in the previous case
- Disadvantages:
  - Memory consumption
    - Doesn't scale well when there are many users
  - In case of server farms
    - Intelligent load balancing is needed (server affinity)
    - or state server → single point of failure

Where to store session data (client)

- URL parameter
- Hidden field
- Cookie
- HTML 5:
  - Local storage and session storage
  - IndexedDB
  - File system

Important (client)

- Attacker can see the traffic (eavesdropping)
  - HTTPS
- Attacker can change the data (tampering)
  - Digital signature
- Data can be lost (e.g. browser crashes, the user manually clears the data)
  - fallback mechanism is required
- Limited size
- HTML 5 support of different browsers are different
  - Keeps changing

Where to store session data (server)

- Session config in web.config

```
<sessionState
  mode="Off|InProc|StateServer|SQLServer„
  cookieless="true|false"
  timeout="number of minutes"
  stateConnectionString="tcpip=server:port"
  sqlConnectionString="sql connection string"
  stateNetworkTimeout="number of seconds"/>
```

Important (server)

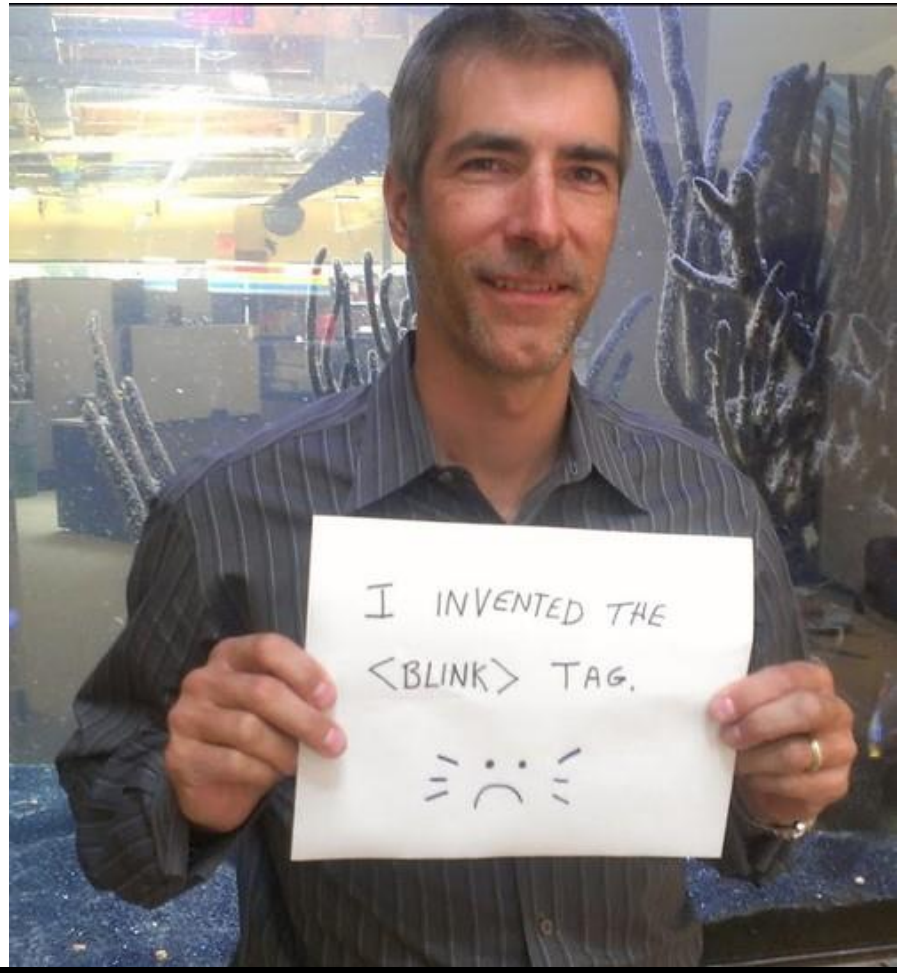
- Scaling problem in case of many users
  - Difficult to test
- Web app or webserver can restart anywhen (e.g. process crash, OS upgrade)
- Load balancing difficulties in case of server farms (server affinity or state server).

HTTP Cookie

- RFC 6265: **HTTP State Management Mechanism** (2011. april)
- Originally RFC 2109 (Lou Montulli, Netscape, 1997. feb.)
- HTTP cookies, also known as web cookies, internet cookies, or browser cookies, are small pieces of data generated by a web server when a user visits a website. These cookies are stored on the user's computer or device through the web browser. Multiple cookies can be stored on a user's device during a single browsing session.
- The goal of Cookies is to be used as memory for http. A cookie can contain a random sessionID, a username or anything that is not large!

Lou Montulli

- In 1991 Lynx brwoset
- BLINK tag
- Anim GIF
- Second webcam
- He invented the web cookie in 1994.

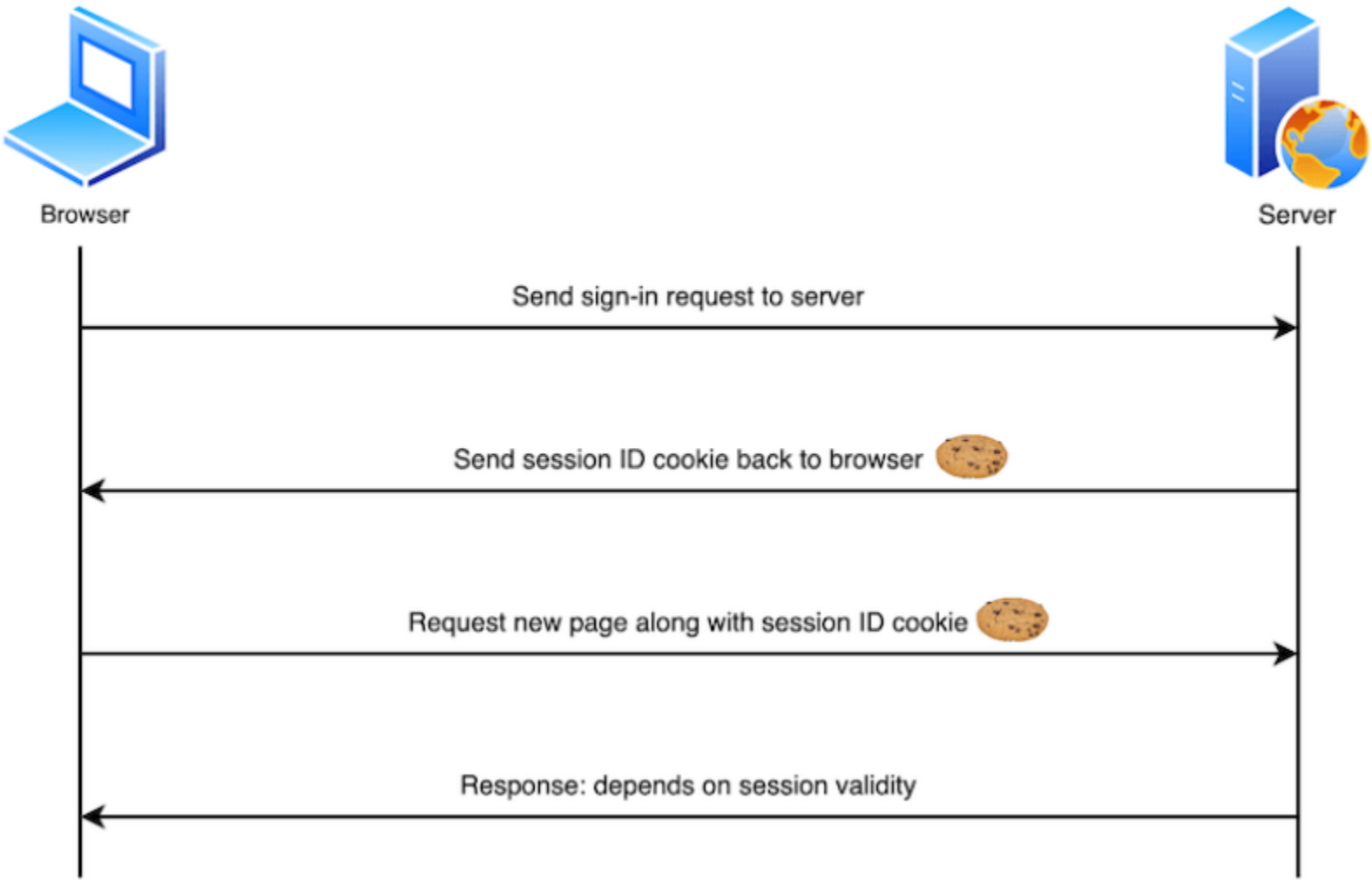


http://www.montulli-blog.com/2013/05/the-reasoning-behind-web-cookies.html

Cookie

- Goal: memory for HTTP
- There was an idea to identify browsers
  - Users can be tracked → the idea was rejected
- Create a SessionId, send it to the browser that it then always attaches to each subsequent request
- Don't allow cross site tracking
- Cookies of our time are 95% based on the original idea
- A cookie can contains a random sessionID, a username or anything that is not large!
- Needs to be deleted
  - When the browser is closed
  - When the computer restarts

Cookies


Browser
Server
Send sign-in request to server
Send session ID cookie back to browser
Request new page along with session ID cookie
Response: depends on session validity

3rd party cookie

- The problem came up around 1996
  - Cookies were not designed to track users across sites
- 2 solutions
  - Enable 3rd party cookies
    - Tracking cookies and companies are visible
    - Governments can restrict and regulate the collection of data
  - Disable user tracking
    - They will come up with something else that is hard to notice

Types of cookies

- **Session (in-memory/transient) cookie**
  - Only for the user session
  - Browser windows share them
- **Permanent (persistent) cookie**
  - Persisted to hard disk
  - „Remember me” checkbox on the login screen

Content of a cookie

- Name
- Value
- Expiration date
- Path
- Domain
- Secure
- HttpOnly

Cookies

- The following HTTP response instructs the receiving browser to store a pair of cookies:

```
HTTP/2.0 200 OK
Content-Type: text/html
Set-Cookie: yummy_cookie=chocolate
Set-Cookie: tasty_cookie=strawberry
```

- When a new request is made, the browser usually sends previously stored cookies for the current domain back to the server within a Cookie HTTP header:

```
GET /sample_page.html HTTP/2.0
Host: www.example.org
Cookie: yummy_cookie=chocolate; tasty_cookie=strawberry
```

HTTP headers

- **Set-Cookie**
- **Cookie**

- There is no dedicated header to delete a Cookie
  - Overwrite with an empty content or set it expired
- The browser sends it back to the server if the domain and the path is the same
  - Even when it is not needed for a given HTTP request
    - E.g. CSS → cookieless domain

Cookies

- You can also access existing cookies and set new values for them

```
console.log(document.cookie);
// logs "yummy_cookie=chocolate; tasty_cookie=strawberry"

document.cookie = "yummy_cookie=blueberry";

console.log(document.cookie);
// logs "tasty_cookie=strawberry; yummy_cookie=blueberry"
```

- with the Secure attribute and the HttpOnly attribute

```
Set-Cookie: id=a3fWa; Expires=Thu, 21 Oct 2021 07:28:00 GMT; Secure; HttpOnly
```

What is the max size of a cookie?

- The standard specifies a minimum for the maximum value (Section 6.1 Limits):
  - Min. 4096 bytes / cookie
  - Min. 50 cookies / domain
  - Min. 3000 cookies together
- Practical limit is: 4096 bytes / cookie.
- According to the RFC standard servers should try to decrease the number and the size of the cookies to minimize the required network bandwidth

Security

- Travels in clear text
  - Not encrypted, HTTPS, set the **Secure** flag
- The client stores it in clear text (privacy problem)
  - Encrypt the content
- Its content can be modified (tampered)
  - Integrity check, digital signature, HMAC
- Its origin cannot be verified
  - We cannot make it for sure that it was created by the server
    - Digital signature
  - An other browser can also send it back to the server (session hijacking)
    - Bind it to a client (IP address in the value field)

Security

- Client script can access and modify it
  - XSS attack → set the **HttpOnly** flag
- Persistent cookies can be sent to the server unwantedly after closing the browser
- Cookie store:
  - Per browser
  - Per user
  - „Private browsing”

shell:cookies

- IE: C:\Users\<USER>\AppData\Roaming\Microsoft\Windows\Cookies
- FF: C:\Users\<USER>\AppData\Roaming\Mozilla\Firefox\Profiles\<ID>\cookies.sqlite
- Chrome: C:\Users\<USER>\AppData\Local\Google\Chrome\User Data\Default\Cookies (SQLite)

Used when…

- Session management
- Personalization
- Tracking → privacy questions
  - **First-party cookie**: used by the visited site
  - **Third-party cookie**: used by an external domain (advertiesment, banner, web bug).

- Internet Options → Privacy → Advanced
- Chrome: Settings → Show advanced settings → Content settings
- Firefox 22: it was to be disabled but caused many compatibility issues → postponed
- 2005: 28% of the users block them, 22% of the users delete 3rd party cookies monthly

Web Storage (DOM Storage)

- Problems with cookies:
  - Limited size
  - Attached to every HTTP request
    - Security, peformance
  - Difficult to use with many browser windows (bound to HTTP requests)
- Solution: **Web Storage** (aka DOM Storage)
- Originally it was part of the HTML 5 specification
  - http://www.w3.org/TR/webstorage/
- W3C Recommendation (July 30. 2013.)

Web Storage (DOM Storage)

- Stores key-value pairs
- Both the key and the value are strings
- Other object types are converted to strings by the browser
- Compound types are worth converting to JSON format
  - JSON.parse(), JSON.stringify()

Web Storage – Size limit

- Browsers have to apply quotas
  - May ask consent from the user to use more space
- Recommended limit is 5MB/origin.
  - JavaScript string UTF-16 → 1 character = 2 byte → in some browsers (especially the Chromium based ones) it means only 2 500 000 chars
  - http://dev-test.nemikor.com/web-storage/support-test/

Firefox → about:config → dom.storage.default_quota

Web Storage - Types

- **Session storage:** information is preserved until the browser tab is closed
  - Data is available only to the current browser tab (per-page-per-window).
  - The session storage is not available when the HTML page is opened as a local file
- **Local storage:** preserves data until the user clears it
  - Each page of the given domain can access the stored data
  - The user can delete it anywhen ⍰ has to be prepared for that

http://html5demos.com/storage

Cookie vs Storage

Aspect	Cookie	Storage
Size limit	4KB	5MB (2.5MB)
Lifetime	Session and persistent	Session and local
Type of content	String	String
Network traffic	Travels	Doesn't travel
API	Client and server side	Only client side, supports event handling
Browser support	All	Almost all
Security	Can be attacked on client side and in the network traffic, but can be HttpOnly	Can be attacked on the client side

IndexedDB

- Shortcomings of DOM Storage
  - Can be used and is optimized for small-sized data
  - Supports only key-value pairs
  - Doesn't support searching among data
  - Synchronous API only
- First solution: **Web SQL Database**
  - November 18, 2010. W3C announced that Web SQL is deprecated and IndexedDB is the new direction

Indexed Database API

- **IndexedDB** is a low-level API for client-side storage of significant amounts of structured data, including files. This API uses indexes to enable high-performance searches of this data.
- W3C Recommendation (January 8., 2015.)
  - http://www.w3.org/TR/IndexedDB/
- Goal: store large amount of data + fast searching supported by indexes
- Applications:
  - Client side caching ⇒ better performance
  - Offline usage

Indexed Database API

- Asynchronous API
  - Requests can be defined with completion callbacks. The callback is executed when the request has completed or has failed
  - Non SQL-based requests
- There is a synchron API that is only available from web workers, browsers don't support it
- Key-value pairs but the value can be a complex type and the key can contain the property(ies) of the object
- Supports transactions

Indexed Database API

- The space limit is usually 50MB
  - With the consent of the user more space can be used
  - Same-origin policy applies to it
- Limitations
  - Doesn't support language dependent sorting
  - Doesn't support synchronization with server-side databases
  - No full text search, no searching as with the LIKE operator in SQL
- Browser support
  - IE10,
  - Firefox 4
  - Chrome 11
  - Opera 15
  - Safari and mobild browsers usually don't support it

Firefox: about:config ⇒ dom.indexedDB.warningQuota

```
// Store
localStorage.setItem("lastname", "Smith");

// Retrieve
document.getElementById("result").innerHTML = localStorage.getItem("lastname");
```

```
if (localStorage.clickcount) {
  localStorage.clickcount = Number(localStorage.clickcount) + 1;
} else {
  localStorage.clickcount = 1;
}
document.getElementById("result").innerHTML = "You have clicked the button " +
localStorage.clickcount + " time(s).";
```

```
if (sessionStorage.clickcount) {
  sessionStorage.clickcount = Number(sessionStorage.clickcount) + 1;
} else {
  sessionStorage.clickcount = 1;
}
document.getElementById("result").innerHTML = "You have clicked the button " +
sessionStorage.clickcount + " time(s) in this session.";
```

	Cookies	Web Storage	IndexedDB
Storage	Small lookup table with pairs of key, data values	Strings only. Key, value storage	ObjectStore that can store any type of data including objects
Capacity	4KB	5MB-25MB	50MB upwards
Indexing	Not Available	Not Available	Available
API Call Type	Synchronous	Synchronous	Asynchronous
Operations performance	Directly performed	Directly performed	Transactional
Learning Curve	Low	Low	High

History API

- History API provides access to the browser's session through the history global object.
- Worked out by WhatWG as part of HTML 5
- Storing navigation based state information is a common issue
- Don't refresh the whole page but make the Back- Forward buttons work
- Earlier solutios: #! (hashbang) URL-ek
  - pl. https://twitter.com/#!/gyorgybalassy
  - Part after the **#** sign: URI fragment
    - Server won't receive it
    - Has to be handled from JavaScriptből

https://html.spec.whatwg.org/multipage/browsers.html#history

Where to store state?

- If the state needs to be bookmarked
  - In URL
- When the server side needs the data
  - URL, hidden field, cookie
- When the data is small and all browsers need to be supported
  - DOM Storage
- When there is a large amount of data, there is complex queries and browser support is not an issue
  - IndexedDB
- Navigation related data
  - History API

Where to store state?

From security's point of view

- Stored data is visible for everybody
  - Manual encryption is required
- Can be modified by anybody
  - Manual integrity protection

From reliability's point of view

- Data can partially or fully be deleted anywhen
  - fallback.
- The browser may not support the applied API
  - modernizr
- We may reach the disk quota
- The user may open new tabs or windows anywhen

HyperText Transfer Protocol Secure (HTTPS)

HTTPS

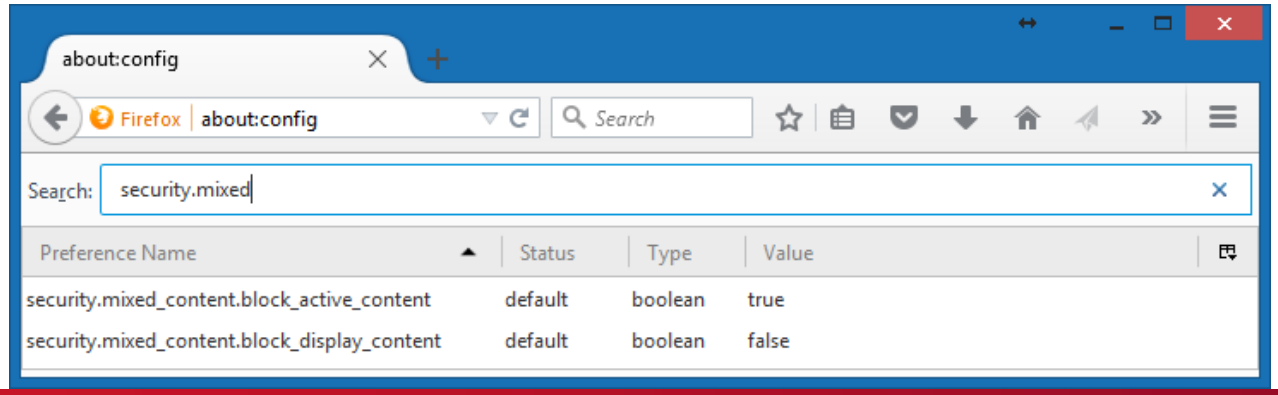
- Hypertext transfer protocol secure (HTTPS) HTTP transfer is not encrypted
  - HTTP + SSL (Secure Sockets Layer protocol)
  - HTTP + TLS (Transport Layer Security)
- „HTTP over SSL”
- Port: 443
- **https://** URI schema
- SSL 3.0 ? TLS 1.0 = SSL 3.1
- SSL/TLS can be used with other protocols as well

Features

- Server authentication: who is the client communicating with
  - Can also authenticate the client (mutual authentication), but is rarely used
- Encrypts the traffic: third party won't be able to read it (eavesdropping).
- Integrity protection: third party won't be able to modify it (tampering).

Features

- All references of the page must be **https://**, otherwise: mixed content warning.
- Firefox 23: Mixed content blocker
  - Active content
    - script, stylesheet, frame - blocked by default
  - Passive (display) content
    - image, audio, video, object - not blocked
  - No domain whitelist



Certificate

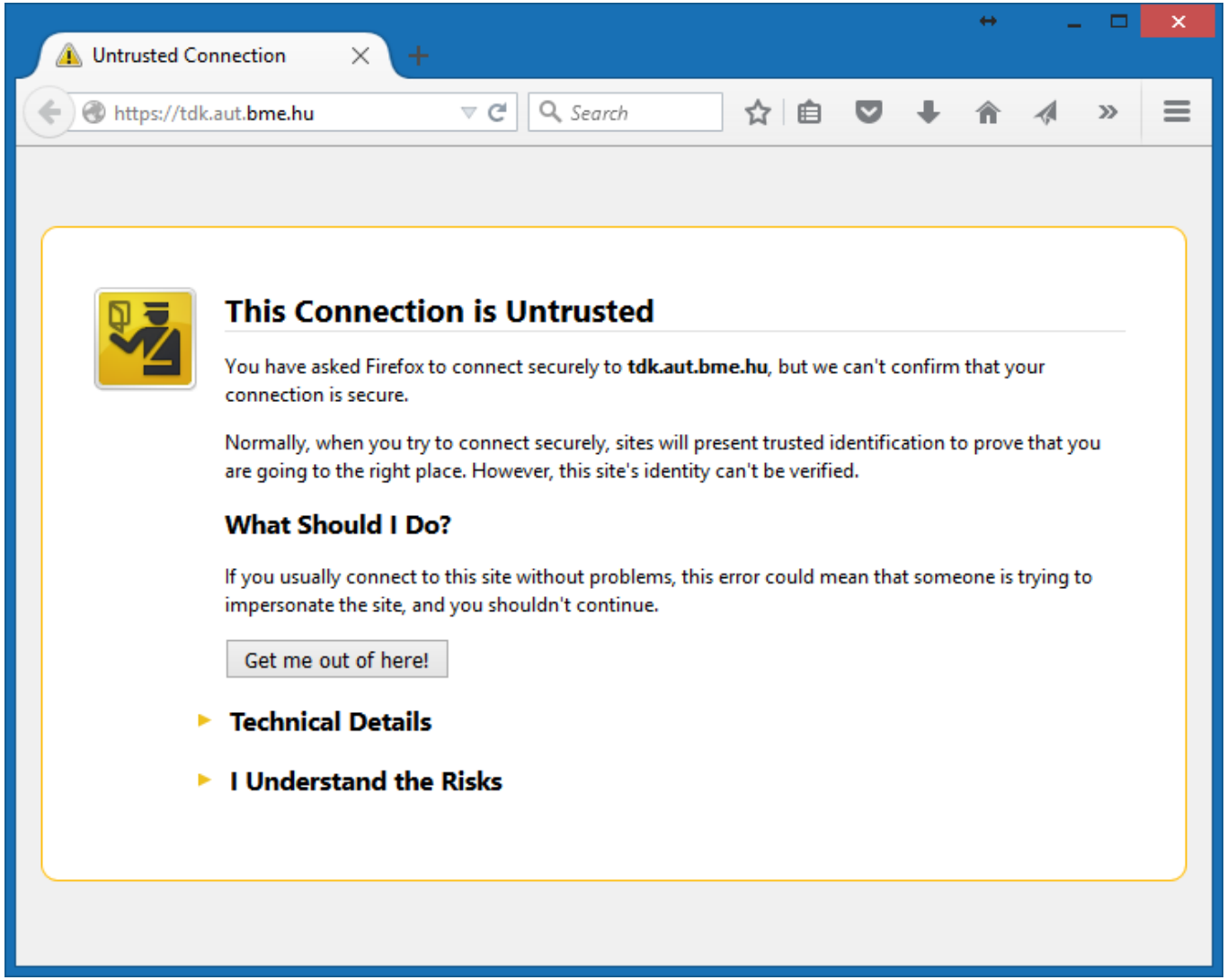
Principle: a trusted 3rd party confirms the authenticity of the server

- **Certificate (X.509 certificate)**
- **Certificate chain, chain of trust**
- **Certification Authority** (CA)
  - Subordinate CA (intermediate CA)
  - Root Certification Authority (Root CA)

Self-signed certificate

- Advantage:
  - Cheap
  - Fast
  - Flexible
- Disadvantages:
  - Doesn't authenticate the server
  - Can be attacked with man-in-the-middle attack
  - Users are taught to accept not authentic certificates

Certificate problems



Properties of a certificate

- Version
- Serial Number
- Signature algorithm
- Signature
- Issuer
- Valid from, Valid to
- Subject
- Public key
- Thumbprint algorithm
- Thumbprint (fingerprint)
- Extensions (opcionális)
  - Key usage
  - Subject Alternative Name (SAN)

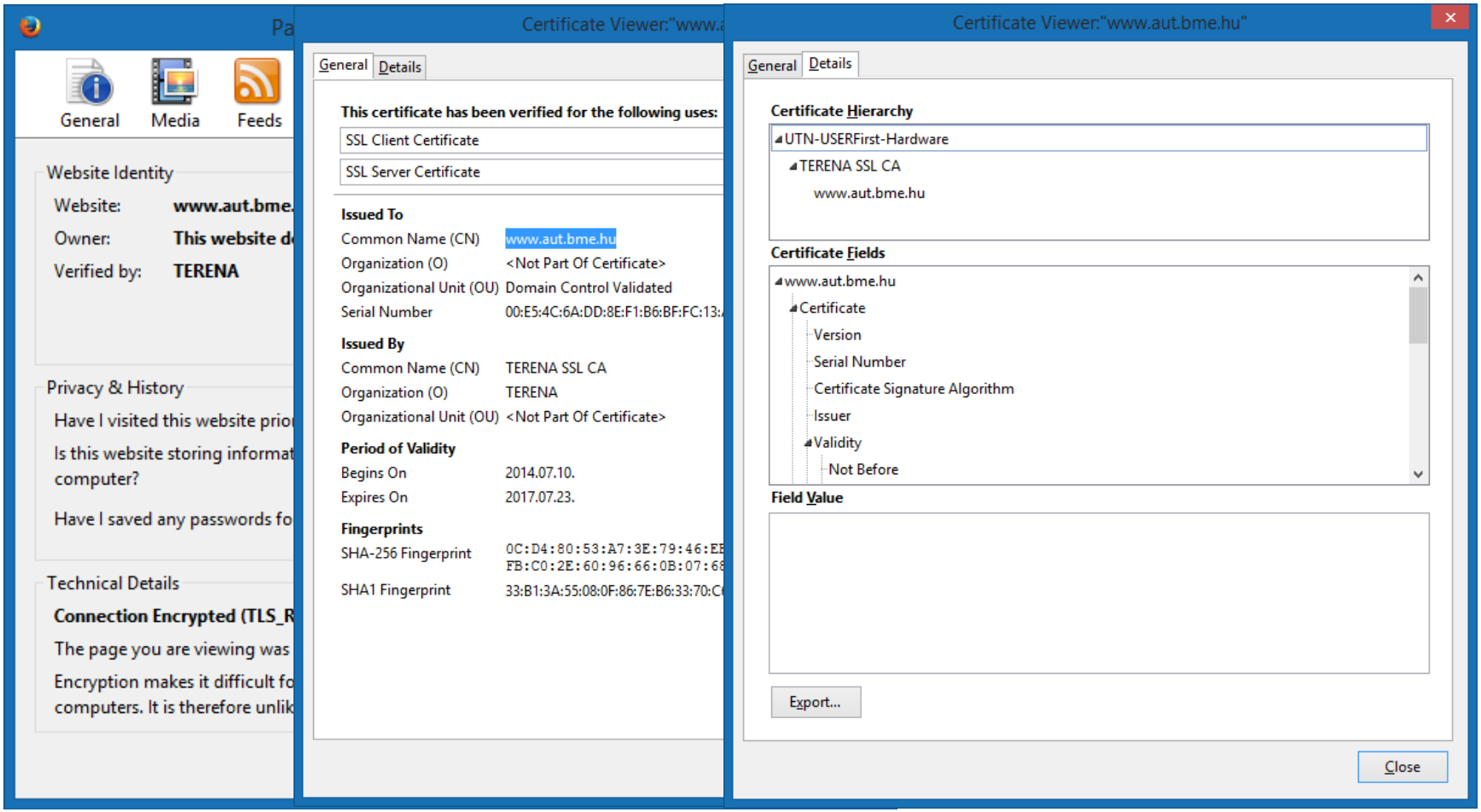
Private key of a certificate

The private key is not part of the certificate

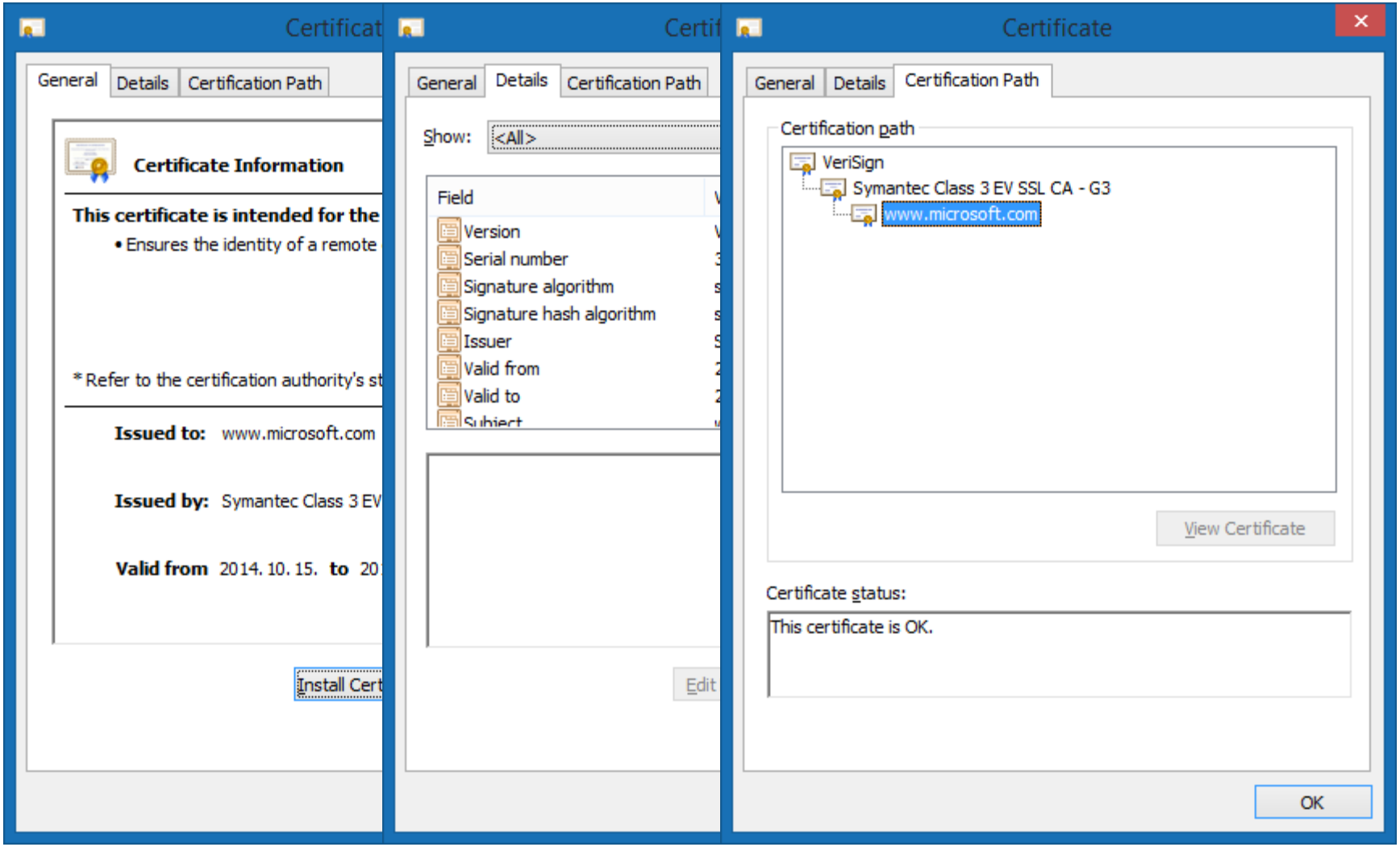
- Can be protected by password
- Can be exportable and non-exportable
- Server manages is, CA won't get it. CA only guarantees that the public key belongs to a certain owner
- File formats
  - .pem, .cer, .crt, .der, .p7b, .p7c, .p12, .pfx

Certificate of the AUT portal Firefox

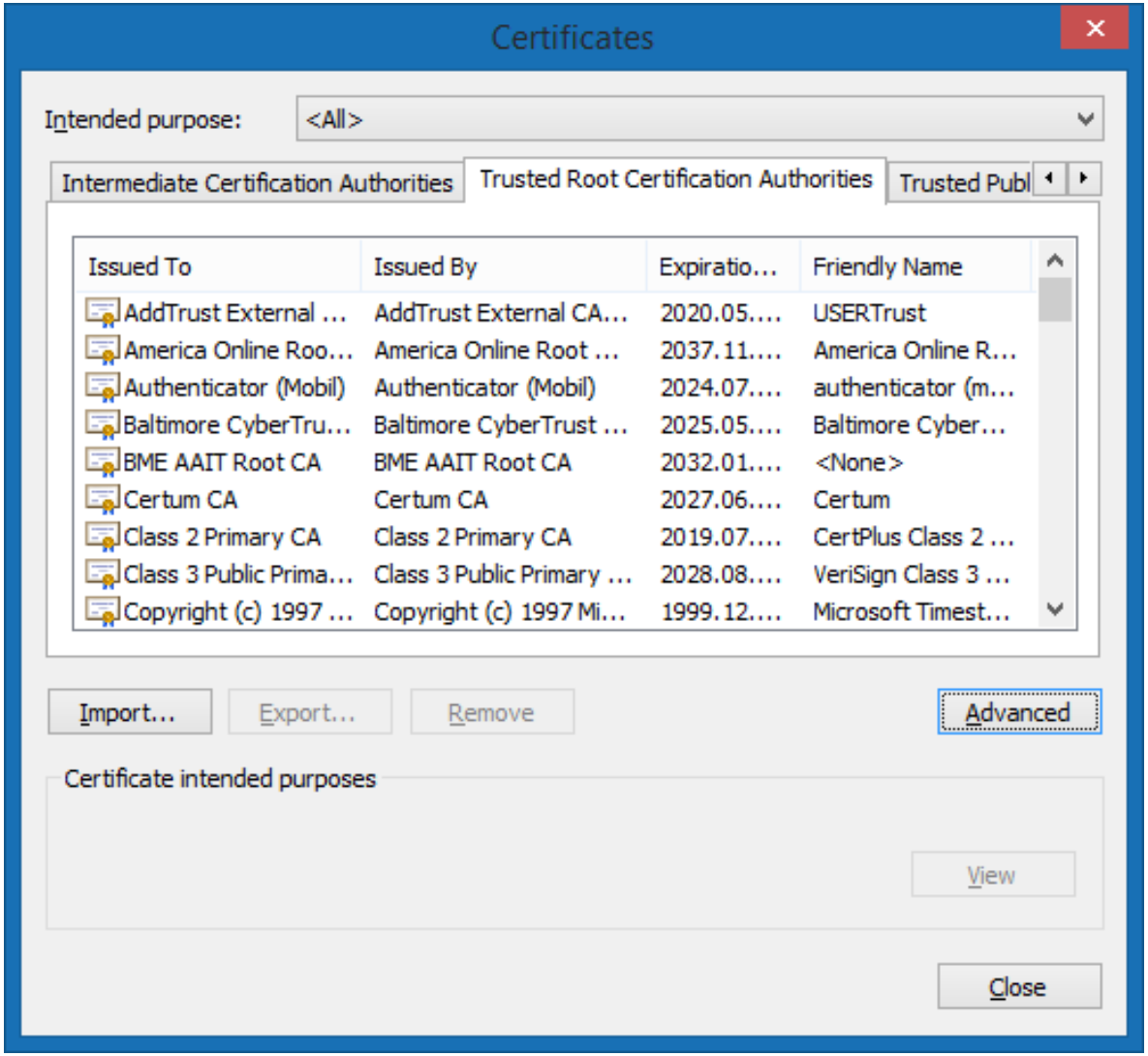
- https://www.aut.bme.hu



https://www.microsoft.com IE



IE Trusted Root certificates



Validating a certificate

Valid if all 4 conditions are fulfilled

1. **The issuer is trusted**
   - The browser need to trust all the items of the CA chain
   - The issuer of the root certificate must be in the list of Trusted Root CAs of the browser
   - A self-signed certificate doesn't identify the server but encrypts the network traffic
2. **Not expired**
   - The usual expiration time period is 1-3 years

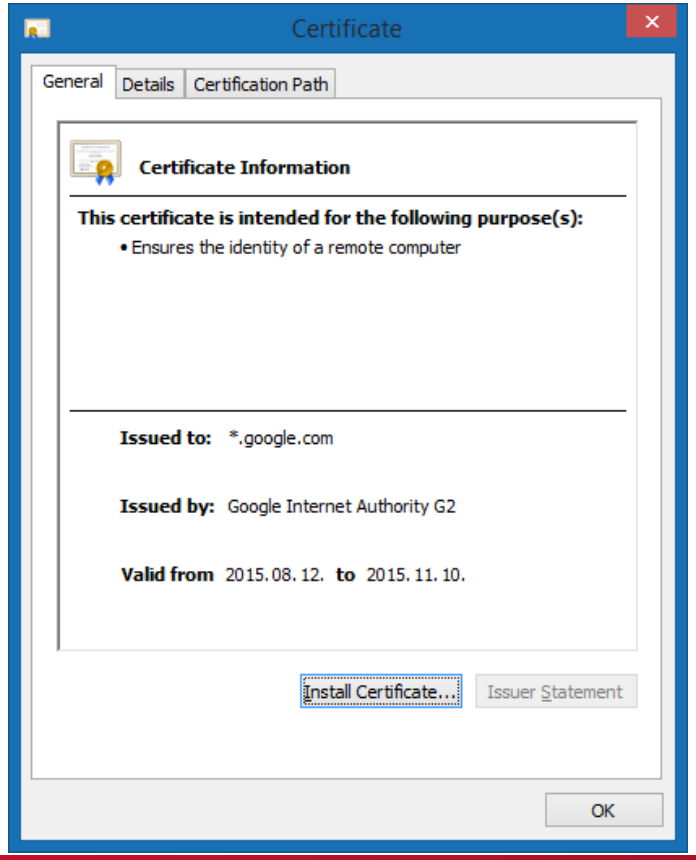
Validating a certificate

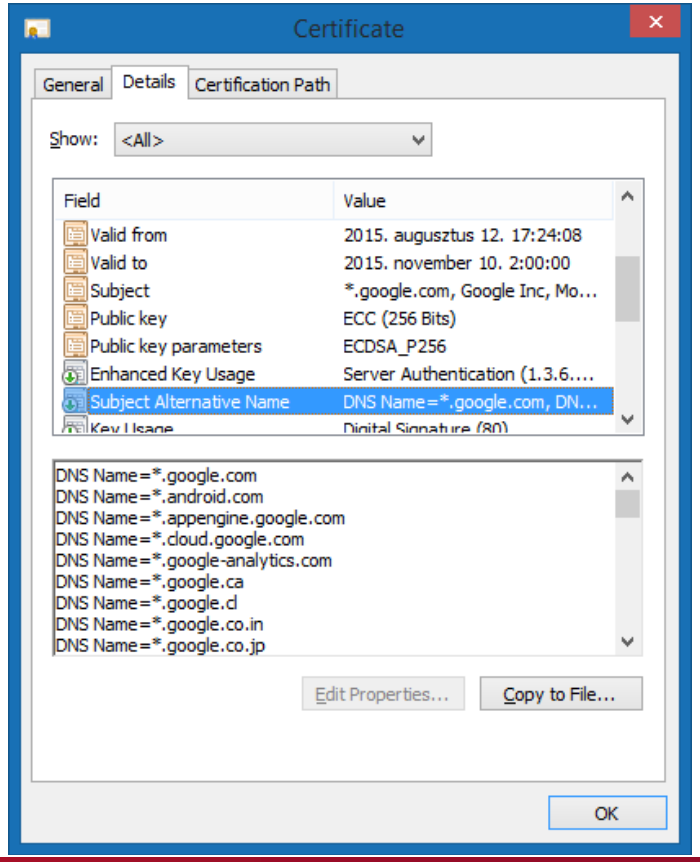
3. Issued for the actual server

- The (Common Name)CN in the Subject field must be the same as the FQDN of the requested site
  - https://example.com != https://www.example.com
- When there are multiple FQDN aliases they need to be redirected to the one that is the one the certificate refers to
- Wildcard certificate: ***.example.com**
  - To multiple subdomains
  - Only allows 1 level depth
  - Extended Validation is not supported

http://www.google.com

- *.google.com
- Multiple names in the Subject Alternative Name fiels



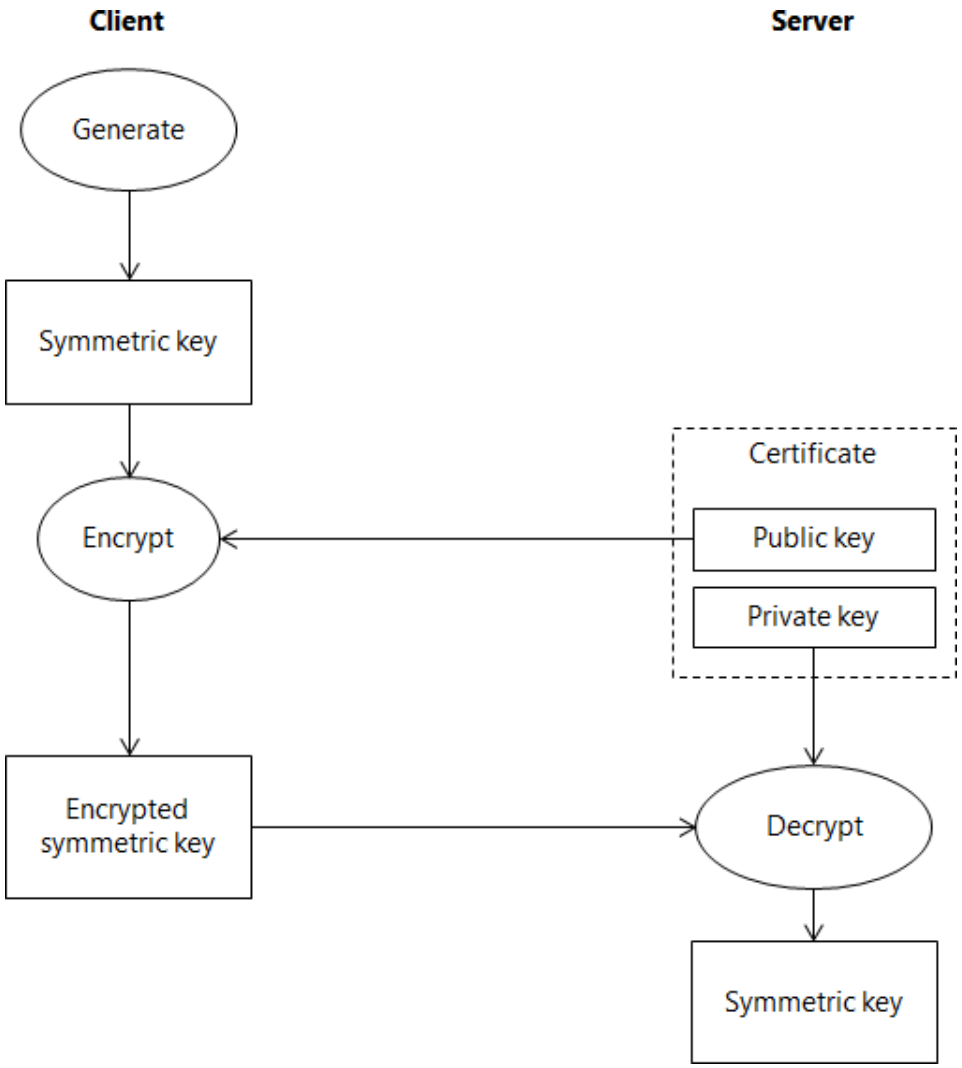


Validating a certificate

4. Not revoked

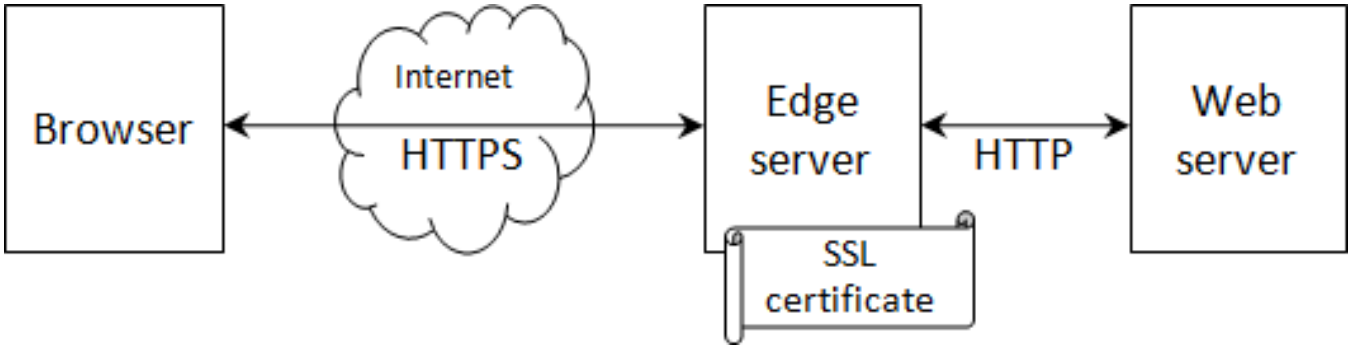
- The certificate or the CA can be compromised
  - 2001: VeriSign „Microsoft Corporation” https://technet.microsoft.com/en-us/library/security/ms01-017.aspx
  - March, 2011.: Iranian hackers used Comodo certificates for a man-in-the-middle attack
  - 2012: Trustwave issued subordinate root CA cert that was later used with MITM attacks
- Certificate Revocation List
  - Signed, contains TTL (24 hours), public list.
  - The URL is determined by the CRL Distribution Point contained in the certificate
- Online Certificate Status Protocol (OCSP, RFC 6960)
  - Can be used to query the status of a certificate from the CA
  - Client doesn’t need to process the wholeCRL

The process of key exchange



SSL termination

- Misbelief: SSL overloads the server
- January, 2010: Gmail starts using HTTPS
  - +1% CPU, +10kB memory/connection, +2% network overhead
  - https://www.imperialviolet.org/2010/06/25/overclocking-ssl.html
- Offloading the webserver from HTTPS related cryptography
- In case of server farms the certificate needn't be deployed to each server



Practical advices

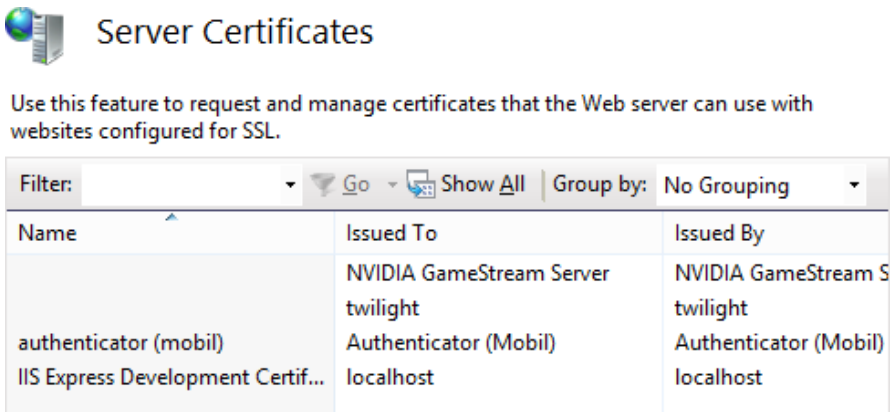
- Short checklist
  - Every sensitive data travels over HTTPS
  - HTTPS pages don't contain HTTP content
  - Authentication cookies are not used over HTTP
  - Secure flag is set for authentication cookies
  - Login pages use HTTPS

Certificates MMC snap-in

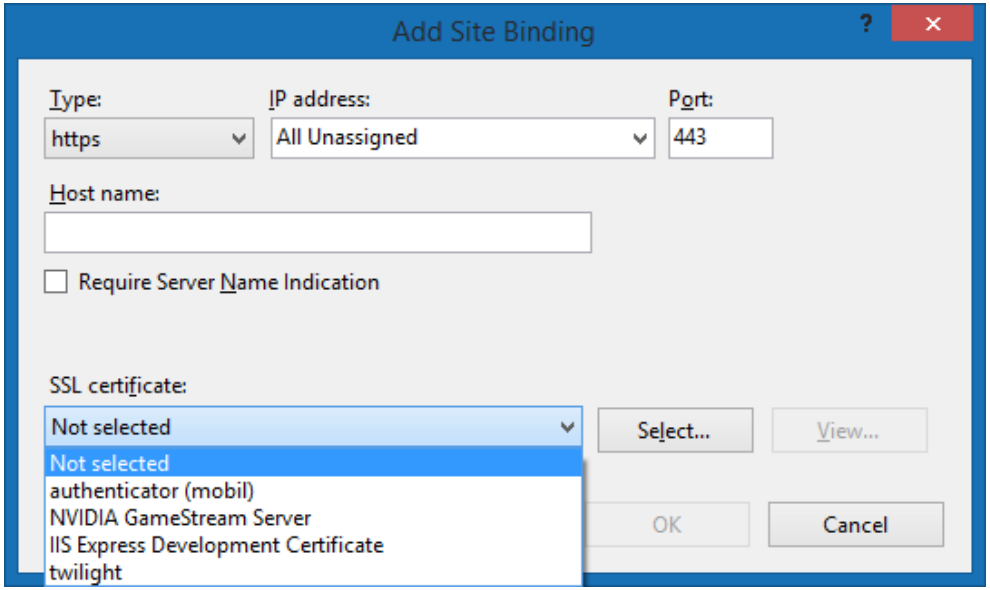
Visual Studio ▯ IIS Express ▯ Use SSL

IIS setting

- IIS Manager
  - > Server Certificates



- Site bindings



Practical advices II.

- All traffic over HTTPS sometimes cannot be achieved
  - Avoid mixed content: all used service has to support HTTPS:
    - Ad network
    - Image hosting service
    - Embedded contents from external services
      - Gravatar, Facebook, Google Analytics
  - CDN will be more expensive
  - Load balancers have to support SSL offloading
  - Has to keep maintaining certificates
  - From HTTP to HTTPS change-over may have SEO consequences

Thank you for your attention!