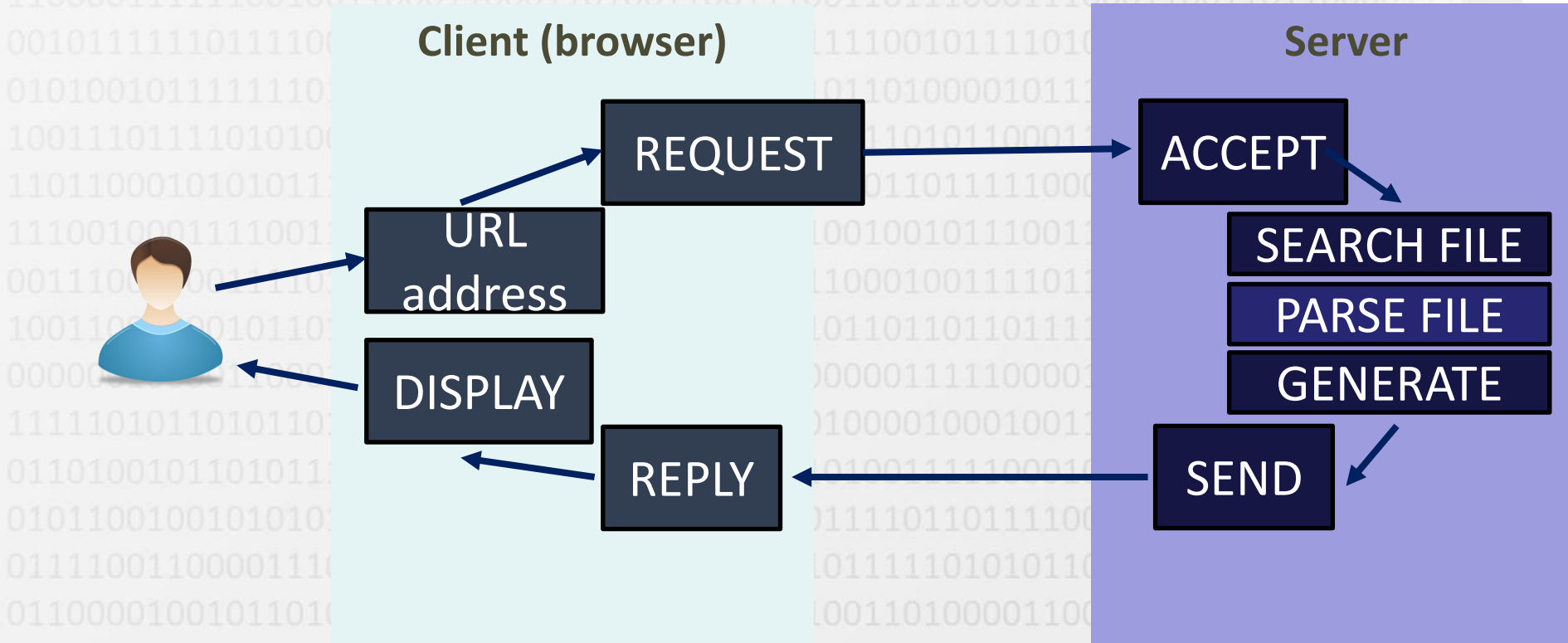


Haladó fejlesztési technikák

Adat-kódolási módszerek
Internetes adatmegosztás
SOAP, REST

Szerveroldali webprogramozás



GET/POST

- https://www.google.hu/search?biw=1356&bih=780&tbm=isch&sa=1&ei=Co8IWu-4Jlee6ASd-rC4DA&q=get+request+php&oq=get+request+php&gs_l=psy-ab.3..0i24k1.31945.32304.0.32490.4.4.0.0.0.137.457.2j2.4.0....0...1.1.64.psy-ab..0.4.456...0i19k1j0i30i19k1j0i5i30i19k1j0i8i30i19k1.0.E0Bilc-HLVE#imgrc=avXEoB6d96FGaM

- [protocol]://[web_server_host_ip]/[file]?

action=mitcsinalj&action2=megmitcsinalj

Változó

Változó

SEO



1 Protocol 2 Sub-Domain 3 Domain 4 2nd Level domains 5 Folder/Path 6 Page



`www.domain.com/home/web-page`



`www.domain.com/home?id=12654`

Webes szolgáltatások

- Szokványos bemenet: GET/POST/SESSION/COOKIES
(extra input: manapság az egyéb HTTP kulcsszavakat is használjuk)
- Szokványos kimenet: HTML
- Sokszor nem egyszerű file-t akarunk letölteni, hanem műveletet akarunk végrehajtani a szerveren
- Cél: mindezt egyszerűen, platform- és nyelv-függetlenül, könnyen implementálhatóan
- Első kérdés: adatkódolás, lehetőleg platform- és nyelvfüggetlenül
- Második kérdés: adattovábbítási módszerek

Alacsony szintű adatkódolás

- **Nyelvfügő** – a legegyszerűbb típustól kezdve
- **bool**
 - „Encode as string” vs „Binary encoding”
 - Előbbi esetben: 0/F/False/FALSE/HAMIS? Utóbbi esetben: hány biten?
- **int/long**
 - „Encode as string” vs „Binary encoding”
 - Utóbbi esetben: MSB first vs LSB first?
- **float/double**
 - „Encode as string” vs „Binary encoding”
 - Előbbi esetben: tizedesvessző vagy tizedespont; normalizált alak vagy nem?
- **string/char**
 - Modern magas szintű nyelveken (C#, Java) általában fix és mindent jól/értelemszerűen kódol (UTF8 / Unicode)
 - Kevésbé modern magas szintű nyelveken (PHP) fix és furcsaságok (pl. strtoupper vs mb_strtoupper)
 - Alacsony szintű nyelveken (C++) tetszőleges és SOK furcsaság (char[], char*, wchar_t*, std::string, std::wstring, QString, CString, fbstring)

Bonyolultabb adat kódolása - XML

```
<?xml version="1.0"?>
<bookstore>
  <book category="COOKING">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="WEB">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

- Szigorú formátum
- XML deklaráció + XML node + XML attribute
- Bonyolultabb adatszerkezet elkódolása is lehetséges!
- „Encode as string”
- Bármilyen encoding (UTF8!!!)
- Schema/DTD lehetséges

Bonyolultabb adat kódolása - JSON

```
<empinfo>
  <employees>
    <employee>
      <name>Scott Philip</name>
      <salary>£44k</salary>
      <age>27</age>
    </employee>
    <employee>
      <name>Tim Henn</name>
      <salary>£40k</salary>
      <age>27</age>
    </employee>
    <employee>
      <name>Long yong</name>
      <salary>£40k</salary>
      <age>28</age>
    </employee>
  </employees>
</empinfo>
```

```
{ "empinfo" :
  {
    "employees" : [
      {
        "name" : "Scott Philip",
        "salary" : £44k,
        "age" : 27,
      },
      {
        "name" : "Tim Henn",
        "salary" : £40k,
        "age" : 27,
      },
      {
        "name" : "Long yong",
        "salary" : £40k,
        "age" : 28,
      }
    ]
  }
}
```

- Nincs schema = flexibilisebb (bár lehetséges sémát írni)
- „Encode as string”
- UTF8 encoding, bár nem kötelező

Bonyolultabb adat kódolása - YAML

```
--- !clarkevans.com/^invoice
invoice: 34843
date : 2001-01-23
bill-to: &id001
  given : Chris
  family : Dumars
  address:
    lines: |
      458 Walkman Dr.
      Suite #292
    city : Royal Oak
    state : MI
    postal : 48046
ship-to: *id001
product:
- sku      : BL394D
  quantity : 4
  description : Basketball
  price     : 450.00
- sku      : BL4438H
  quantity : 1
  description : Super Hoop
  price     : 2392.00
tax : 251.42
total: 4443.52
comments: >
  Late afternoon is best.
  Backup contact is Nancy
  Billsmer @ 338-4338.
```

SCALAR

COLLECTIONS

MULTI-LINE COLLECTIONS

LISTS/DICTIONARIES

MULTI-LINE FORMATTING

- Ritkán használt: ROS, PHP Symfony, Amazon AWS
- Schema nem lehetséges
- „Encode as string”, UTF8 encoding javasolt

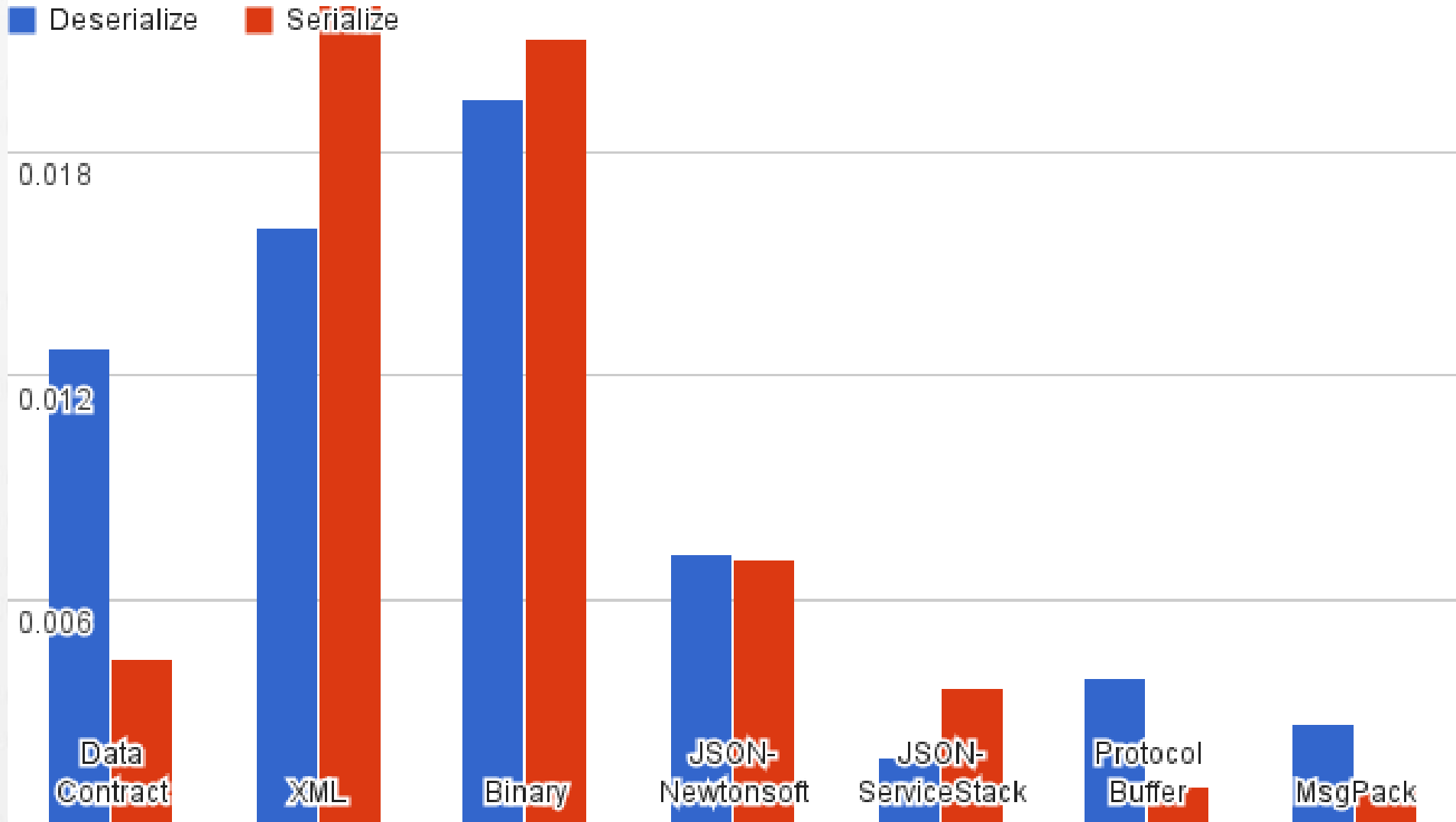
Type (formatter)	Binary Formatter	Soap Formatter	XML Serializer Class	Data Contract Serializer	Net Data Contract Serializer	Protobuf-net
Best Performance	Yes	No	No	No	No	Yes
Readable with other platforms (non .NET)	No	Yes	Yes	Yes	No	Yes
Serialize object private members	Yes (deep serialization)	Yes (deep serialization)	No (shallow serialization)	Yes	Yes	Yes
Serialize generic collections (*)	Yes	No	Yes	Yes	Yes	Yes
Easy control of how each member is serialized	No (**)	No (**)	Yes (by using attributes)	Yes	Yes	Yes
Saves metadata in output stream for deserialization	Yes	Yes	No	No	Yes	No

(*) System.Collection.Generic namespace

(**) By using custom serialization / deserialization

Serialization - sebesség

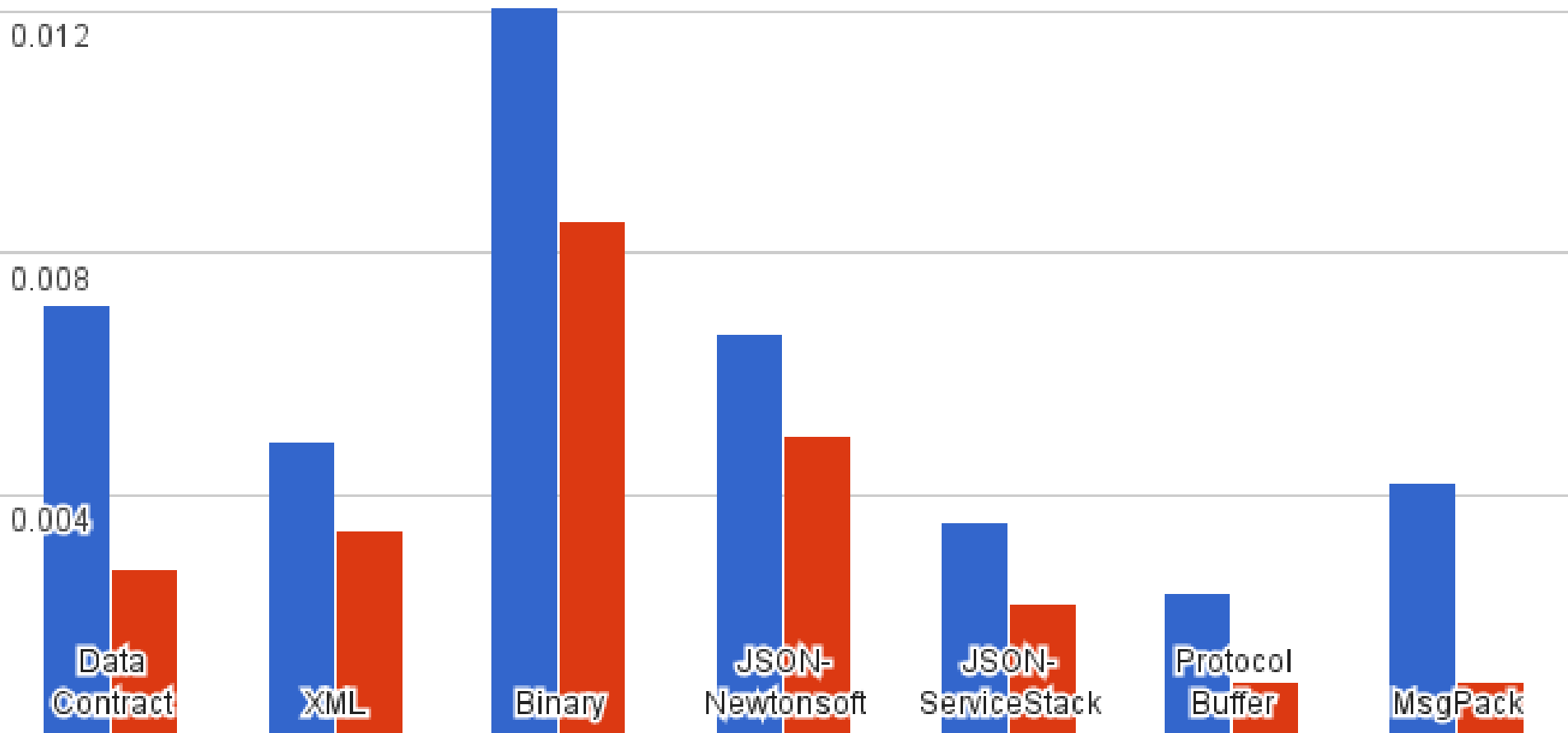
Speed for small data



Serialization - sebesség

Speed for large data

■ Deserialize ■ Serialize



Adattovábbítás – Alacsony szinten

- TCP/UDP: megbízható és sorrendtartó stream VAGY gyorsabb datagram
- „Encode as binary”
- Minden esetet, kódolást, formátumot nekünk kell kezelni
(pl: szeretném meghívni a GetModelExtras metódust azzal az autóval, aminek márkája „BMW”, a modellnév „i8”.
Eredményként az extrák listája kell → minden kódolás kérdéses)

	<u>TCP/UDP</u>
<u>Overhead (format)</u>	SMALL (binary)
<u>Overhead (comm)</u>	SMALL (raw tcp/udp)
<u>MS/C# specific</u>	NO
<u>Complex types?</u>	NO (manual)
<u>Code generation</u>	NO
<u>Webserver needed</u>	NO
<u>Bi-directional</u>	YES
<u>Mature?</u>	YES!!
<u>Implementation diff.</u>	HARD!!

DO NOT reinvent the wheel...

- **SOAP**

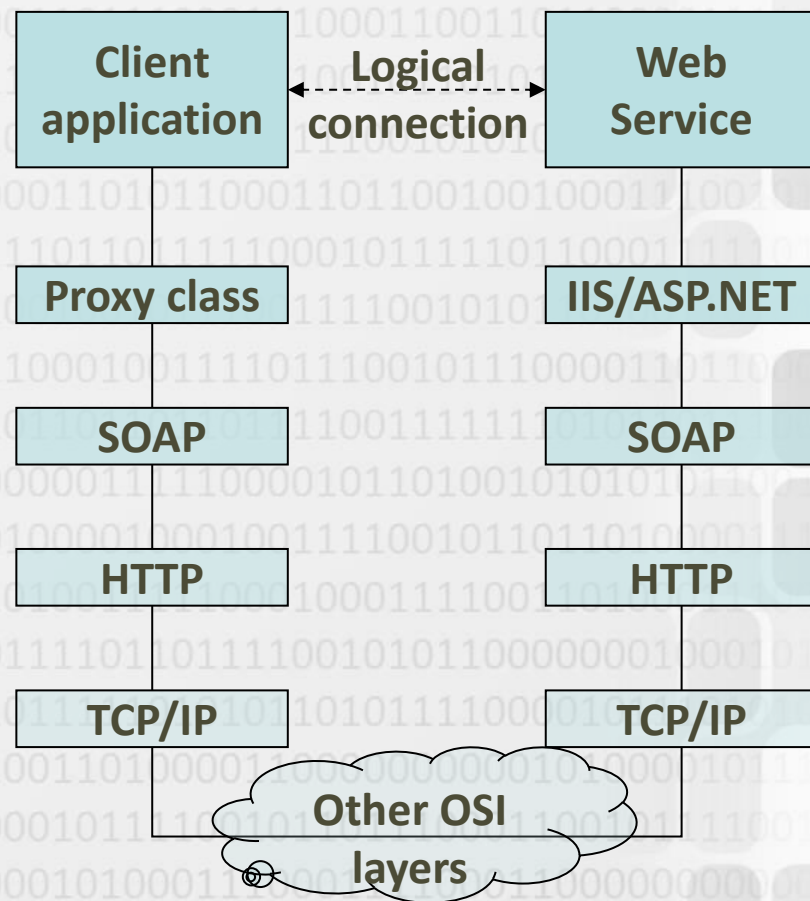
- SOKKAL régebbi
- Előre definiált SOAP XML formátumok metódusok meghívására, és bármilyen típusú paraméterek/eredmények továbbítására (array, list, object)
- Az XML üzeneteket tetszőleges protokoll továbbíthatja, tipikusan HTTP
- Nagyon könnyű implementálni (nyelvi+IDE támogatás!), lassú, nagy overhead

- **REST**

- Többnyire a meghívott eljárás neve után string/int paraméterek kellenek → HTTP GET URL, semmi más nem szükséges
- Esetleg az eljárás neve is opcionális, használhatóak az extra HTTP metódusok: PATCH, PUT, DELETE, OPTIONS, HEAD ...
- Komplex adat küldhető: HTTP POST-ban JSON (ritkán XML)
- A válasz tipikusan JSON (ritkán XML)
- Könnyű implementálni (több munka, mint a SOAP), közepes sebesség, közepes overhead (HTTP, vagyis nem raw TCP)

SOAP: MINDEN rejtett/automatizált

- *Simple Object Access Protocol*
- SOAP library elérhető szinte mindegyik programozási környezetben
- Proxy osztályok automatikus generálása
- A szerver proxy egyszerűen metódusokat definiál, a kliens proxy ezen metódusokat tartalmazza
- A metódushívás közbeni XML kommunikáció teljesen rejtett



SOAP vs REST

- A SOAP-ban nagy lehetőségek voltak, sok támogató protokoll készült hozzá (DISCO, UDDI) – mára alig használt, sebesség és konfigurációs gondok miatt
- Manapság nyílt végpont alig van, tipikusan belső intranetes kommunikációban használt (kivétel: OTP cardpay, MNB árfolyamok <http://www.mnb.hu/arfolyamok.asmx>)

The request:

```
GET /S
Host:
Content
Content
```

The response:

```
HTTP/1.1 200 OK
```

```
Content-Type: application/soap+xml; charset=utf-8
```

```
Content-Length: nnn
```

```
<?xml
```

```
<?xml version="1.0"?>
```

```
<env:En
```

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
```

```
xmlns:
```

```
xmlns:s="http://www.example.org/stock-service">
```

```
<en
```

```
<env:Body>
```

```
<
```

```
<s:GetStockQuoteResponse>
```

```
<
```

```
<s:StockPrice>45.25</s:StockPrice>
```

```
<
```

```
</s:GetStockQuoteResponse>
```

```
</en
```

```
</env:Body>
```

```
</env:
```

```
</env:Envelope>
```

SOAP vs REST

- *Representative State Transfer*
- SOAP = szigorú protokoll, REST = javasolt architektúrális felépítés
- Egyszerűbb üzenetek, saját kódolás, így kódgeneráció nincs (kivétel: WCF – MEX endpoint; Swashbuckle/NSwag komponensek)

The request:

```
GET /StockPrice
Host: example.org
Accept: text/xml
Accept-Charset: utf-8
```

The response:

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>
<s:Quote xmlns:s="http://example.org/stock-service">
  <s:TickerSymbol>IBM</s:TickerSymbol>
  <s:StockPrice>45.25</s:StockPrice>
</s:Quote>
```

Webes eljárás hívások tesztelése: Postman, SoapUI

The screenshot displays the Postman application interface. On the left, a sidebar shows a list of collections, with the first one containing several endpoints for a local WordPress installation. The main area is set to 'Normal' mode with 'Basic Auth' selected. The request URL is `http://local.wordpress.dev/wp-json` and the method is 'GET'. The response status is '200 OK' and the time taken is '143 ms'. The response body is displayed in 'Pretty' format, showing a JSON object with the following structure:

```
1 {
2   "name": "Local WordPress Dev",
3   "description": "Just another WordPress site",
4   "URL": "http://local.wordpress.dev",
5   "routes": {
6     "/": {
7       "supports": [
8         "HEAD",
9         "GET"
10      ],
11      "meta": {
12        "self": "http://local.wordpress.dev/wp-json/"
13      }
14    },
15    "/posts": {
16      "supports": [
17        "HEAD",
18        "GET",
19        "POST"
20      ],
21      "meta": {
22        "self": "http://local.wordpress.dev/wp-json/posts"
23      }
24    },
25    "accepts_json": true
26  },
27  "/posts/<id>": {
28    "supports": [
29      "HEAD",
30      "GET",
31      "POST"
32    ]
33  }
34 }
```


REST = Access anything

Rest API Basics

Typical HTTP Verbs:

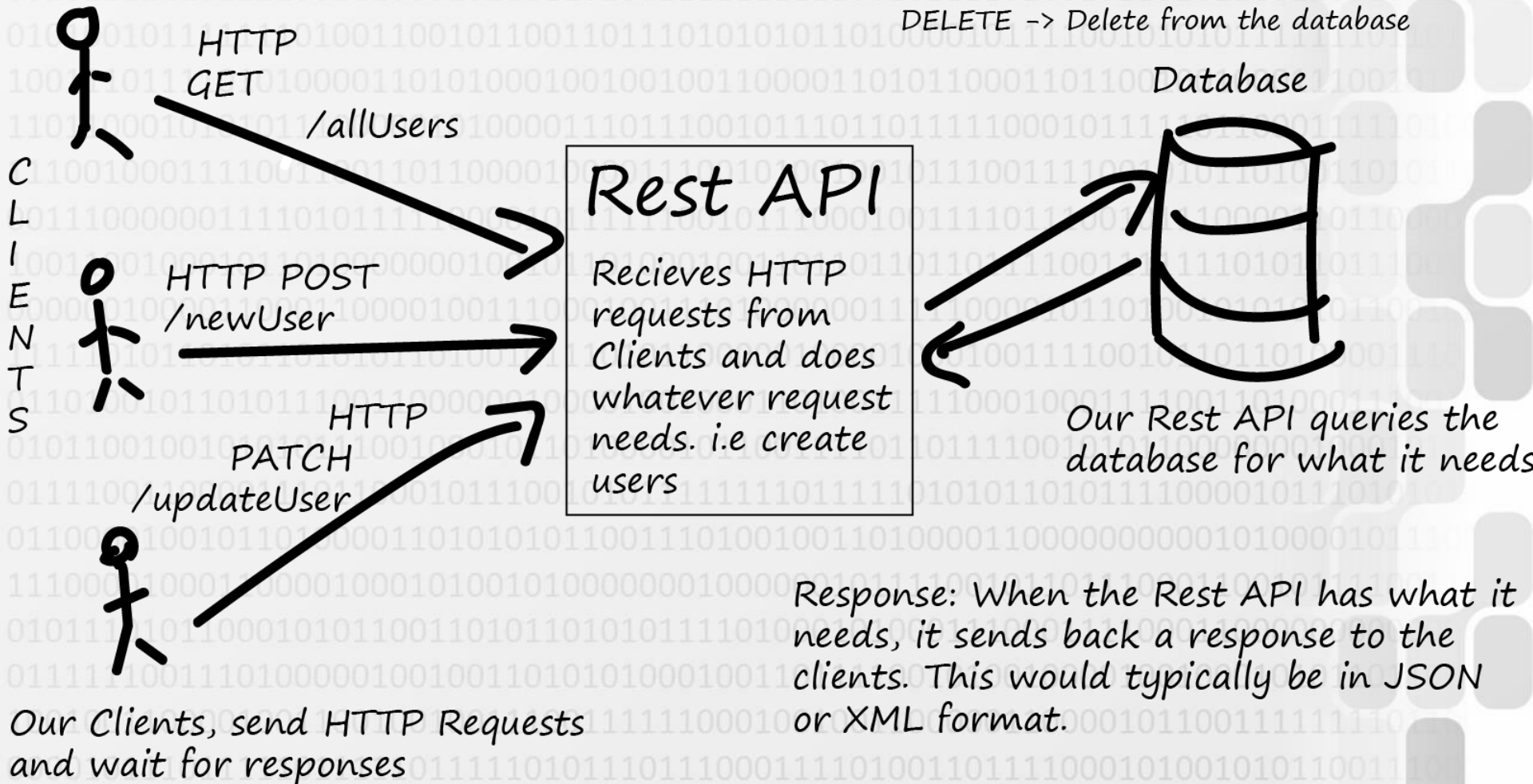
GET -> Read from Database

PUT -> Update/Replace row in Database

PATCH -> Update/Modify row in Database

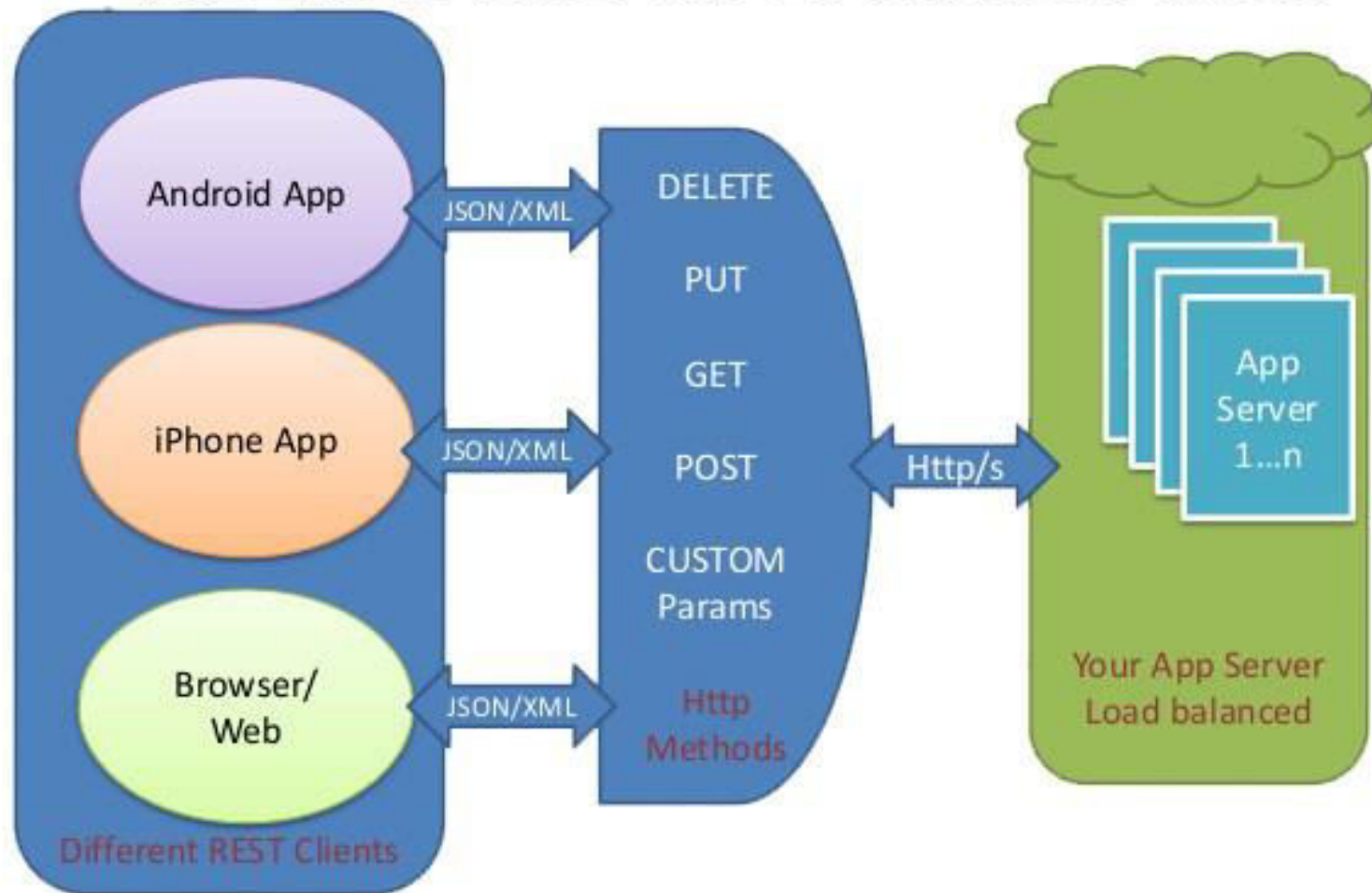
POST -> Create a new record in the database

DELETE -> Delete from the database



REST = Access anywhere

REST API Architecture



REST példák

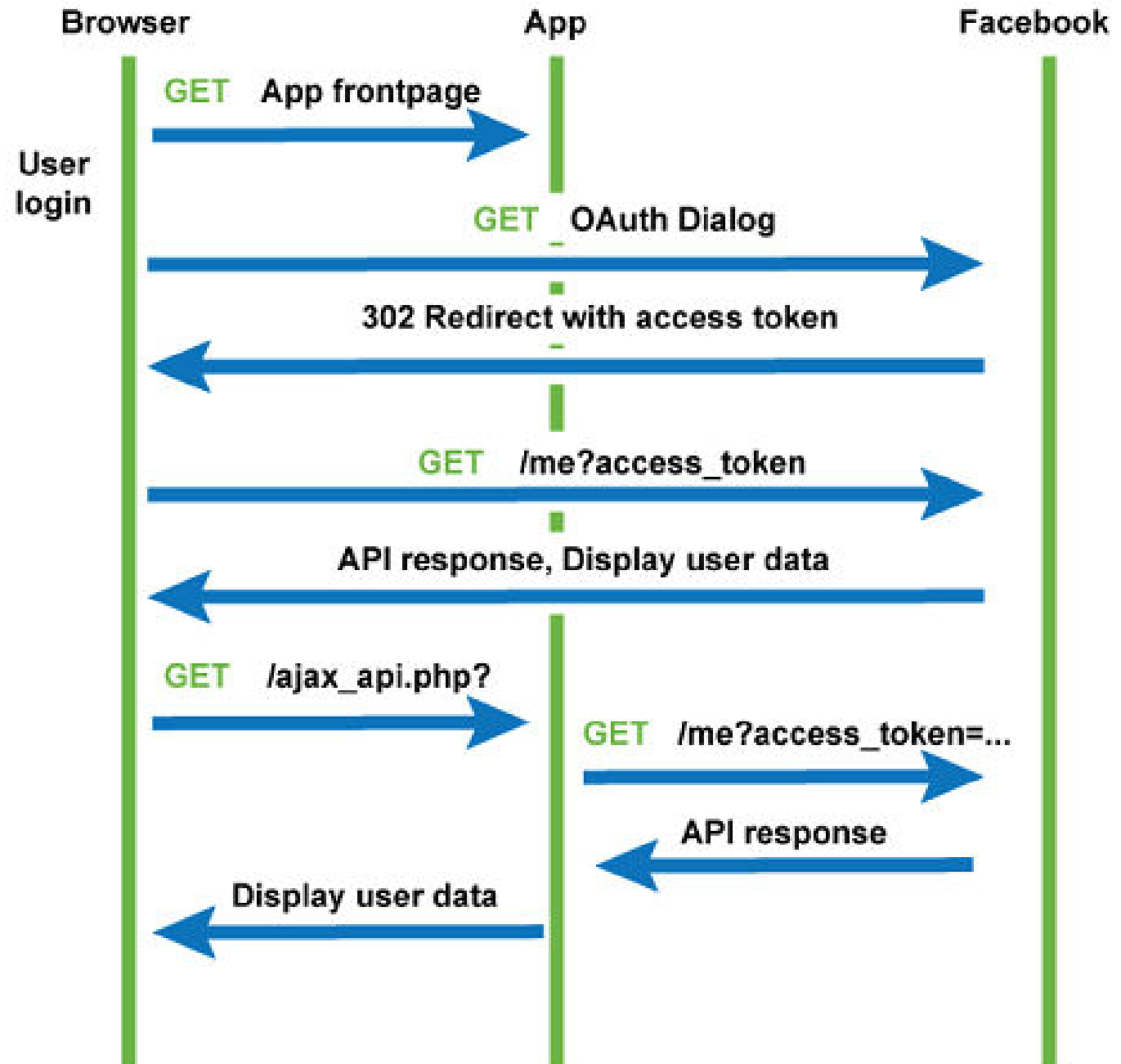
Cég-specifikus

- Facebook: graph API, marketing API
- Google: 138 API kategória (tavaly: 121)
(<https://developers.google.com/apis-explorer/#p/>)
- Twitter: ~200 API végpont
(<https://developer.twitter.com/en/docs/api-reference-index>)
- <https://www.programmableweb.com/apis/directory>

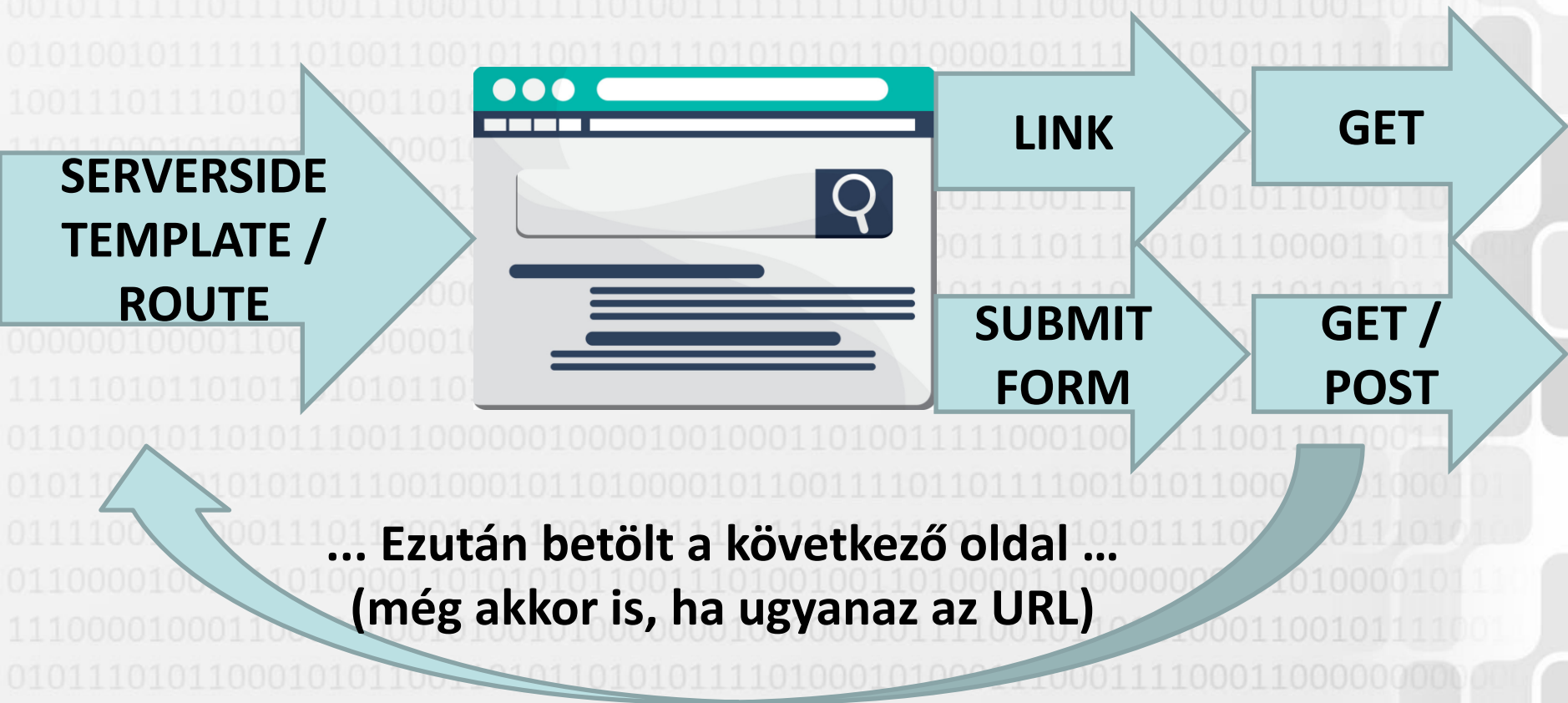
Gyártó-független

- Linked Open Data (LOD)
- OpenGraph
- Odata, Oauth

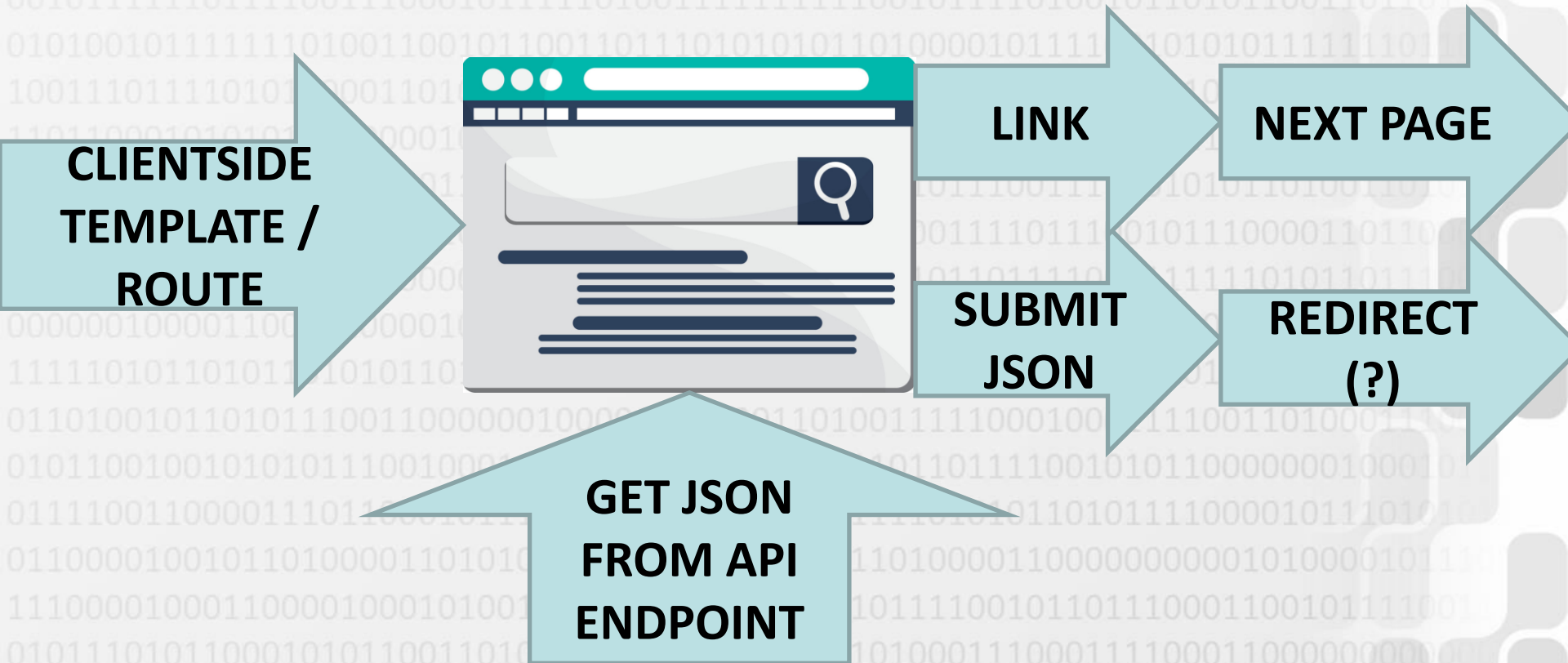
Oauth flow



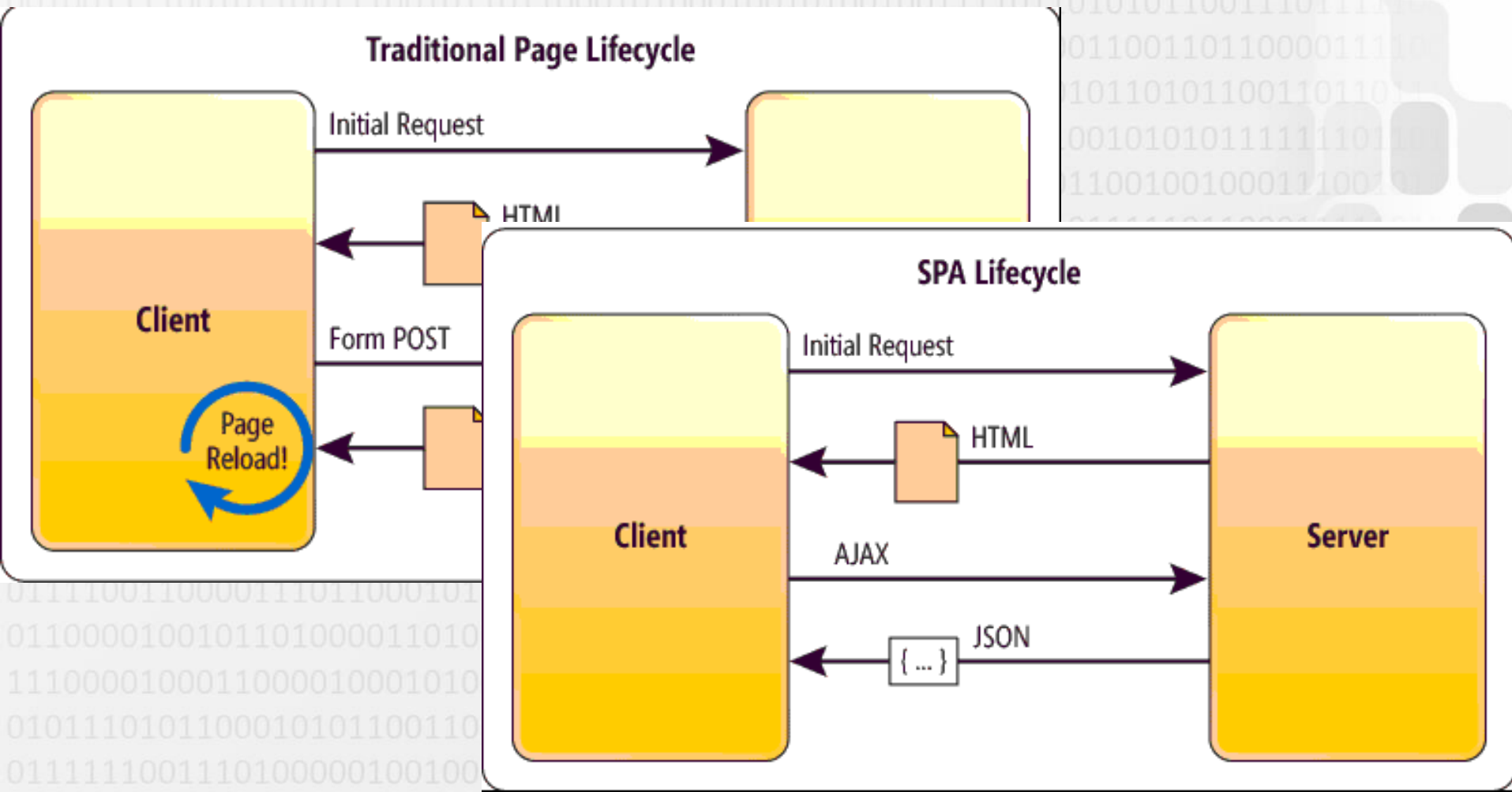
Klasszikus weboldalak



Modern weboldalak



Opcionális Redirect?



SOAP, REST

	<u>SOAP</u>	<u>REST</u>
<u>Overhead (format)</u>	BIG (XML)	OK (JSON)
<u>Overhead (comm)</u>	BIG	BIG (?)
<u>MS/C# specific</u>	NO	NO
<u>Complex types?</u>	YES	YES
<u>Code generation</u>	YES	NO (?)
<u>Webserver needed</u>	YES (?)	YES
<u>Bi-directional</u>	NO	NO
<u>Mature?</u>	YES!!	YES
<u>Implementation diff.</u>	EASY!!	OK

WCF

- Zseniális elv, nem annyira zseniális megvalósítás...
- Elv: ABC – Address, Binding, Contract.
 - Address: hol érhető el (ip/host + port), és milyen adattovábbítási mód van használva (http://, tcp://, udp://)
 - Binding: milyen módon kódolódnak és hogyan kezelődnek az adatok (WsHttpBinding, NetTcpBinding, MsMqBinding, NetPeerTcpBinding, MEX)
 - Contract: milyen metódusok/paraméterek/eredmények használhatóak...
 - A Contract hétköznapi Interface definíció mellé hétköznapi osztály implementációt igényel, az Address/Binding Config file-ból állítandó
 - A programkód **TELJESEN FÜGGETLEN A PROTOKOLLTÓL**
- Megvalósítás
 - Windows Communication Foundation
 - **Bonyolult** konfiguráció (ha bármi speciális kell, a SOAP nagyon egyszerű)
 - Teljes SOAP támogatás, teljesnek tűnő (fura) REST támogatás, korábbi MS/.NET technológiák/protokollok integrációja ugyanebbe a keretbe
- Támogatás
 - Nem terjedt el, mert nem reagáltak rá az MS az SPA+REST térnyerését

App.config fájl

App.config x EightBallService.cs IEightBallService.cs

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<configuration>
```

```
<system.web>
```

```
<compilation debug="true" />
```

```
</system.web>
```

```
<system.serviceModel>
```

```
<services>
```

```
<service name="EightBallDll.EightBallService">
```

```
<endpoint address="" binding="wsHttpBinding" contract="EightBallDll.IEightBallService">
```

```
<identity>
```

```
<dns value="localhost" />
```

```
</identity>
```

```
</endpoint>
```

```
<endpoint address="mex" binding="mexHttpBinding" contract="IMetadataExchange" />
```

```
<host>
```

```
<baseAddresses>
```

```
<add baseAddress="http://localhost:8732/Design_Time_Addresses/EightBallDll/EightBallService/" />
```

```
</baseAddresses>
```

```
</host>
```

```
</service>
```

```
</services>
```

```
<behaviors>
```

```
<serviceBehaviors>
```

```
<behavior>
```

```
<serviceMetadata httpGetEnabled="True"/>
```

```
<serviceDebug includeExceptionDetailInFaults="False" />
```

```
</behavior>
```

```
</serviceBehaviors>
```

```
</behaviors>
```

```
</system.serviceModel>
```

```
</configuration>
```

ABC

a cím a baseAddress-ben.
A kötésből látszik, hogy
webszolgáltatás

MEX végpont
engedélyezése

Viselkedésdefiníció a
MEX-nek.

WCF Host

- **Szükséges hozzá a System.ServiceModel dll**
 - Referenciaként hozzá kell adni
- **Az App.Config fájlra itt lesz szükség**
- **A hoszt példányosítása a szolgáltatás típusának megadásával**
 - `ServiceHost host = new ServiceHost(typeof(EightBallService))`
- **A szolgáltatást megnyitjuk, majd a végén bezárjuk**
 - `using`: definiálja az objektum hatókörét, ezen kívül az objektum felszabadul

```
Program.cs x
EightBallHost.Program Main(string[] args)

namespace EightBallHost
{
    class Program
    {
        static void Main(string[] args)
        {
            using (ServiceHost host = new ServiceHost(typeof(EightBallService)))
            {
                host.Open();
                DisplayHostInfo(host);
                Console.WriteLine("The service is ready.\nPress any key to terminate...");
                Console.ReadKey();
            }
        }
    }
}
```

WCF Client

Add Service Reference?×

To see a list of available services on a specific server, enter a service URL and click Go. To browse for available services, click Discover.


Address:


http://www.mnb.hu/arfolyamok.asmx

Go


Discover


Services:


▲  MNBArfolyamServiceSoapImpl


 MNBArfolyamServiceSoap


Operations:


 GetCurrencies

 GetCurrencyUnits

 GetCurrentExchangeRates

 GetDateInterval

 GetExchangeRates

 GetInfo

1 service(s) found at address 'http://www.mnb.hu/arfolyamok.asmx'.

Namespace:

MNBService

Advanced...

OK

Cancel

WCF Client

```
MNBSERVICE.MNBARFOLYAMSERVICESOAPCLIENT client =  
    new MNBSERVICE.MNBARFOLYAMSERVICESOAPCLIENT();  
string xml = client.  
    GetCurrentExchangeRates(null).  
    GetCurrentExchangeRatesResult;  
Console.WriteLine(xml);  
Console.ReadLine();
```

```
<MNBCurrentExchangeRates>
```

```
  <Day date="2018-11-26">
```

```
    <Rate unit="1" curr="AUD">206,17</Rate>
```

```
    <Rate unit="1" curr="BGN">164,81</Rate>
```

```
    <Rate unit="1" curr="BRL">74,03</Rate>
```

```
    <Rate unit="1" curr="CAD">214,83</Rate>
```

```
    <Rate unit="1" curr="CHF">284,37</Rate>
```

```
    <Rate unit="1" curr="CNY">40,85</Rate>
```

```
    <Rate unit="1" curr="CZK">12,46</Rate>
```

```
    <Rate unit="1" curr="DKK">43,2</Rate>
```

Jelenlegi MS trend: WebAPI + SignalR

- **ASP History**

- ASP.NET MVC: Tradicionális weboldalak, ami elég, hiszen API-ra ott a WCF...
- Később: ASP.NET WebApi: JSON API végpontok
- ... Külön megközelítés, külön framework, külön őszosztály, külön config ...
- ASP.NET Core: A két megközelítés ésszerűbb egyesítése

- **SignalR**

- Az eddigi megvalósítások közül csak a TCP/UDP kétirányú
- A HTTP/1 alapból egyirányú. Emiatt a SOAP/REST is egyirányú
- (*Long polling, XHR polling, forever frame, server sent events*)
- A WCF net.tcp bindingja és a WsDualHttpBinding kétirányú...
- Régebbi ASP.NET + SignalR → jQuery + duplex kommunikáció
- Újabb ASP.NET Core + SignalR → npm + TypeScript + duplex kommunikáció
- Webes és self-hosted szerver is lehetséges; webes és self-hosted kliensekkel
- Elvileg platformfüggetlen és nyelvfüggetlen...

WCF + SignalR

	<u>WCF (net.tcp)</u>	<u>SignalR</u>
<u>Overhead (format)</u>	SMALL (custom)	OK (JSON)
<u>Overhead (comm)</u>	SMALL (custom)	BIG (?)
<u>MS/C# specific</u>	YES	NO? (JS)
<u>Complex types?</u>	YES	YES
<u>Code generation</u>	YES	NO (?)
<u>Webserver needed</u>	NO	YES (?)
<u>Bi-directional</u>	YES	YES
<u>Mature?</u>	YES?	NO?
<u>Implementation diff.</u>	EASY?	OK?

Technológiák - összegzés

	<u>TCP/UDP</u>	<u>SOAP</u>	<u>REST</u>	<u>WCF (net.tcp)</u>	<u>SignalR</u>
<u>Overhead (format)</u>	SMALL (binary)	BIG (XML)	OK (JSON)	SMALL (custom)	OK (JSON)
<u>Overhead (comm)</u>	SMALL (raw tcp/udp)	BIG	BIG (?)	SMALL (custom)	BIG (?)
<u>MS/C# specific</u>	NO	NO	NO	YES	NO? (JS)
<u>Complex types?</u>	NO (manual)	YES	YES	YES	YES
<u>Code generation</u>	NO	YES	NO (?)	YES	NO (?)
<u>Webserver needed</u>	NO	YES (?)	YES	NO	YES (?)
<u>Bi-directional</u>	YES	NO	NO	YES	YES
<u>Mature?</u>	YES!!	YES!!	YES	YES?	NO?
<u>Implementation diff.</u>	HARD!!	EASY!!	OK	EASY?	OK?

gRPC

- **.NET CORE 3-ban sincs még WCF**
(rossz nyelvek szerint soha nem is lesz – MS tagadja?)
 - SignalR van az ASP.NET core-ban, valószínűleg marad is, de nem lesz elterjedt technológia (imho)
 - WCF net.tcp binary: ??????
- **REST api továbbra is elérhető, sok sok library**
 - WebClient / HttpClient
 - NSwag (class generation from json replies / schema)
- **gRPC library: HTTP/2**
 - Grpc.io, opensource, MS javaslat is az erre való áttérés
 - Protobuf-net, grpc-net (github: protobuf-net.Grpc): nincs automata MEX
 - MyServer : IMyAmazingService
 - client = http.CreateGrpcService<IMyAmazingService>();
 - Hasonlóan egyszerű (talán az app.config hiánya miatt még jobban), mint a WCF
- **A WCF jövője: kétséges**

