

# **Web services**

Martin Kuba, ÚVT MU

# Brief web services history

- 1989 - WWW invented
- 1991 - HTTP 0.9 specified
- 1992 - Internet at Masaryk University :-)
- 1993 - Mosaic web browser
- 1993 - CGI interface for executing programs
- 1995 - JavaScript introduced by Netscape
- 1996 - SSL 3.0
- 1998 - XML 1.0
- 1998 - SOAP 1.1 by Microsoft
- 2003 - SOAP 1.2 by W3C (never used)
- 2004 - WS-Interoperability Basic Profile

# Brief web services history (2)

- 2000 - REST defined by Roy Fielding
- 2001 - JSON format invented
- 2004 - GMail and Google Maps
- 2004 - Web 2.0 hype, Mash-ups
- 2005 - AJAX (Asynchronous JavaScript)
- 2005 - Yahoo! offers JSON web services
- 2006 - OpenID 2.0
- 2008 - HTML5 (First Public Working Draft)
- 2010 - OAuth 1.0
- 2010 - mobile devices with Android
- 2012 - OAuth 2.0

# Brief web services history (3)

- 2013 - responsive web design as answer to mobile devices with differing screen sizes
- 2006-2013 - cloud computing (Amazon 2006, Microsoft 2008, Google 2013)
- 2014 - HTML5 finalised

# SOAP versus REST

- enterprises prefer complicated stack
  - XML
  - SOAP, WSDL, WS-Interoperability
  - WS-\* (WS-Security, WS-Addressing, ...)
  - persistent connections - queues
  - RPC based
  - complex tools and frameworks
- Internet crowd prefers simplicity
  - JSON
  - web APIs described as HTTP requests to URLs
  - AJAX in browsers
  - transient connections - TCP/IP, HTTP
  - scalable using REST

# Web APIs

- well-known APIs
  - Google APIs (Calendar, GMail, Maps, Picasa, ...)
  - Facebook API
  - Twitter API
  - based on HTTP+JSON+SSL+OAuth
- third party clients
  - web, mobile (Android, iOS, ...), desktop, embedded
- OAuth
  - developer registers an application at API provider
  - user authorises the application to use certain operations in the API, giving the application a token
  - application uses the token to use the API on behalf of the user

# JSON - JavaScript Object Notation

```
{
  kind: "calendar#events",
  etag: "\"GZxpEFttrDA0mLHnWRxLHHWPGwk/vpPwPyIKi2CubgzCW0VY8MIHGPa\"",
  summary: "EGI.eu Events",
  updated: "2013-04-22T06:00:02.000Z",
  timeZone: "Europe/Amsterdam",
  accessRole: "reader",
  items: [
    - {
      kind: "calendar#event",
      etag: "\"GZxpEFttrDA0mLHnWRxLHHWPGwk/Z2NhbDAwMDAxMjY5ODQ0NDcwMDkzMDAw\"",
      id: "vs17ehlthhfrlgke0a0o98hors",
      status: "confirmed",
      htmlLink: https://www.google.com/calendar/event?eid=dnMxN2VobHRoaGZybgdrZTBhMG85OGhvcnMgZXZlbnRzQGvnaS5ldQ,
      created: "2010-02-12T08:47:42.000Z",
      updated: "2010-03-29T06:34:30.093Z",
      summary: "EGEE to EGI Transition Meeting for User Community and Operations",
      description: "A focus on the transition of the EGEE NA2, NA3 and NA4 activities to the EGI era with significance followed by more general transition of EGEE operations to NGI operations from Tuesday afternoon. A detailed agenda is available at /conferenceDisplay.py?confId=1",
      location: "Nikhef",
      creator: {
        email: "steven.newhouse@egi.eu",
        displayName: "Steven Newhouse"
      },
      organizer: {
        email: "events@egi.eu",
        displayName: "EGI.eu Events",
        self: true
      },
      start: {
        dateTime: "2010-03-01T13:00:00+01:00"
      },
      end: {
        dateTime: "2010-03-03T12:00:00+01:00"
      },
      visibility: "public",
      iCalUID: "vs17ehlthhfrlgke0a0o98hors@google.com",
      sequence: 0
    },
  ],
}
```

# The same Google Cal event in XML

```
- <entry>
- <id>
  http://www.google.com/calendar/feeds/events%40egi.eu/private/full/vs17ehlthfrlgke0a0o98hors
  </id>
  <published>2010-02-12T08:47:42.000Z</published>
  <updated>2010-03-29T06:34:30.000Z</updated>
  <category scheme="http://schemas.google.com/g/2005#kind" term="http://schemas.google.com/g/2005#event"/>
- <title type="text">
  EGEE to EGI Transition Meeting for User Community and Operations
  </title>
- <content type="text">
  A focus on the transition of the EGEE NA2, NA3 and NA4 activities to the EGI era with significantly reduced EC funding during the first
  to NGI operations from Tuesday afternoon. A detailed agenda is available - https://www.egi.eu/indico/conferenceDisplay.py?confId=1
  </content>
  <link rel="alternate" type="text/html" href="https://www.google.com/calendar/event?eid=dnMxN2VobHRoaGZybGdrZTBhMG85OGhvc
  <link rel="self" type="application/atom+xml" href="https://www.google.com/calendar/feeds/events%40egi.eu/private/full/vs17ehlthfrlg
- <author>
  <name>Steven Newhouse</name>
  <email>steven.newhouse@egi.eu</email>
  </author>
- <gd:comments>
  <gd:feedLink href="https://www.google.com/calendar/feeds/events%40egi.eu/private/full/vs17ehlthfrlgke0a0o98hors/comments"/>
  </gd:comments>
  <gd:eventStatus value="http://schemas.google.com/g/2005#event.confirmed"/>
  <gd:where valueString="Nikhef"/>
  <gd:who email="events@egi.eu" rel="http://schemas.google.com/g/2005#event.organizer" valueString="events@egi.eu"/>
  <gd:when endTime="2010-03-03T12:00:00.000+01:00" startTime="2010-03-01T13:00:00.000+01:00"/>
  <gd:transparency value="http://schemas.google.com/g/2005#event.opaque"/>
  <gd:visibility value="http://schemas.google.com/g/2005#event.public"/>
  <gCal:anyoneCanAddSelf value="false"/>
  <gCal:guestsCanInviteOthers value="true"/>
  <gCal:guestsCanModify value="false"/>
  <gCal:guestsCanSeeGuests value="true"/>
  <gCal:sequence value="0"/>
  <gCal:uid value="vs17ehlthfrlgke0a0o98hors@google.com"/>
</entry>
</feed>
```



# AJAX

- Asynchronous JavaScript And XML
- does not need XML, uses JSON often ;-)
- based on introduction of XMLHttpRequest JavaScript object to web browsers
- asynchronous request to web server
- response processed in JavaScript
- same-origin policy (protocol,host,port)
- Cross-origin resource sharing (CORS)

# REST

- Representational State Transfer
- software *architecture style* for creating *scalable web services*
- invented by Roy Fielding, author of HTTP 1.1
- resources identified by URIs
- representations of resources as JSON, XML or other formats
- uses HTTP methods GET, PUT, DELETE and POST for manipulating resources

# REST (2)

- no IDL (Interface Description Language) so far
- API described in human natural language
  - e.g. “image can be changed by HTTP PUT request to /image/{imageID}”
- Richardson Maturity Model
  - level 1 - resources identified by URIs
  - level 2 - use of HTTP methods as verbs
  - level 3 - HATEOAS (Hypertext As The Engine Of Application State)
  - level 3 introduces discoverability, making a protocol more self-documenting

# HAL - Hypertext Application Language

- one of proposed standards for HATEOAS (level 3 in Richardson Maturity Model)
- format for JSON messages in REST APIs
  - every object has **\_links** property with links to operations on the object or to other objects
  - collections are wrapped in **\_embedded**
- supported by Spring HATEOAS Java library

# HAL example

```
{
-  _embedded: {
-    categories: [
-      {
        id: 1,
        name: "Food",
-      _links: {
-        self: {
          href: "http://localhost:8080/eshop/api/v1/categories/1"
        },
-        products: {
          href: "http://localhost:8080/eshop/api/v1/categories/1/products"
        }
      }
    ],
-    {
      id: 2,
      name: "Office",
-      _links: {
-        self: {
          href: "http://localhost:8080/eshop/api/v1/categories/2"
        },
-        products: {
          href: "http://localhost:8080/eshop/api/v1/categories/2/products"
        }
      }
    }
  },
- }
```

# Mash ups

- combine data from various sources
- typically a Google map with some geospatial data
  - ships - <http://www.marinetraffic.com/>
  - aircrafts - <http://www.flightradar24.com/>

# www.marinetraffic.com

**MarineTraffic.com**

Search English

Mapa Satelitní Simple

### GRONA AALSUM

**Flag:** Antigua Barbuda

**Ship Type:** Cargo - Hazard D (Recognizable)

**Status:** Underway

**Speed/Course:** 12.4 kn / 39°

**Length x Breadth:** 100 m X 14 m

**Draught:** 5.6 m

**Destination:** SODERTALJE

**ETA:** 2013-04-29 19:00 (UTC)

**Received (1509):** 0h 2min ago (AIS Source: )

[Show Vessel's Track](#)

[Distance to...](#)

[Ship Photos: 41](#)

[Upload a photo](#)

[Vessel's Details](#)

[More Actions](#)

**Ships Map**

Go to Area... ?

Go to Port... ?

Go To Vessel ?

**Notation & Display options:**

- Show Ship Names
- My Fleet
- Wind Now
- Passenger Vessels
- Cargo Vessels
- Tankers
- High Speed Craft
- Tug, Pilot, etc
- Yachts & Others
- Fishing
- Navigation Aids
- Unspecified Ships
- Ships Underway
- Anchored/Moored

**Quick Links:**

- Get an AIS receiver for free!
- Report your own position
- Receiving Stations

Available on the iPhone **App Store**

ANDROID APP ON

Data map ©2013 GeoBasis-DE/BKG ©2009, Google Podmínky použití Nahlásit chybu v mapě

# www.flightradar24.com

flightradar24  
LIVE AIR TRAFFIC

APPS INCREASE COVERAGE ABOUT DATABASE FORUM CHAT

UTC 12:27

Map 52

Search flight or airport...

Playback

Settings

Filter

Planes 6049

Premium

Like 202k Follow +1

Map link  
<http://fl24.com/49211658>

Latest twitter  
Are you a skilled web developer with plenty of ideas for how to improve <http://t.co...>  
1 hour ago

Latest Facebook  
Are you a skilled web developer with plenty of ideas for how to improve [www.flig...](http://www.flig...)  
1 hour ago

POP RON Špička mezi tablety!

GOOGLE NEXUS 10

ihned K DODÁNÍ!

Rupte zde >

REAL TIME ADS-B data 5 MIN DELAY FAA data

Want flightradar24 on your phone, tablet, Mac OS, or Windows 8?

Satellite Beach Flights

www.MyTravelGuide... Find Satellite Beach Flight Deals. Search Top Travel Sites & Book Now.

AdChoices

Want flightradar24 without ads?



# Federated identity

- many authentication mechanisms were developed for the web
  - username+password (hard to remember)
  - X509 digital certificate (complicated to get)
  - digest, Kerberos etc. (not much support in browsers)
- users forget passwords to rarely used accounts
- in federated identity, account from one organisation can be reused at others
- identity providers
  - OAuth - Google, Facebook, Twitter, ...
  - OpenID - Google, MojeID.cz, Seznam.cz, anybody
  - SAML - in academia - universities, Academy etc.

# SOAP/WSDL web services

- preferred in the enterprise world
- used as API for the Czech eGovernment "Data Boxes"
- SOAP is Simple Object Access Protocol
- WSDL is Web Service Description Language
- WS-Interoperability Basic Profile needed to ensure interoperability
  - requires SOAP1.1
- many WS-\* extensions