2.– 5. September 2013
in Nürnberg

Herbstcampus

Wissenstransfer
par excellence

Echt jetzt ?!

Echtzeitanwendungen für das Web mit SignalR

Tobias Krügel

Conplement AG

Einführung

Agenda

- Communication Strategies
- SignalR API
  - Server
  - Client
  - Demo

Agenda

Communication Strategies

Communication Strategies

- **Problem**
  - **Web Communication = HTTP**
  - **HTTP eignet sich nicht für Push Benachrichtigungen (Request, Response)**

Wie können Push Notifications für Web-Anwendungen realisiert werden?

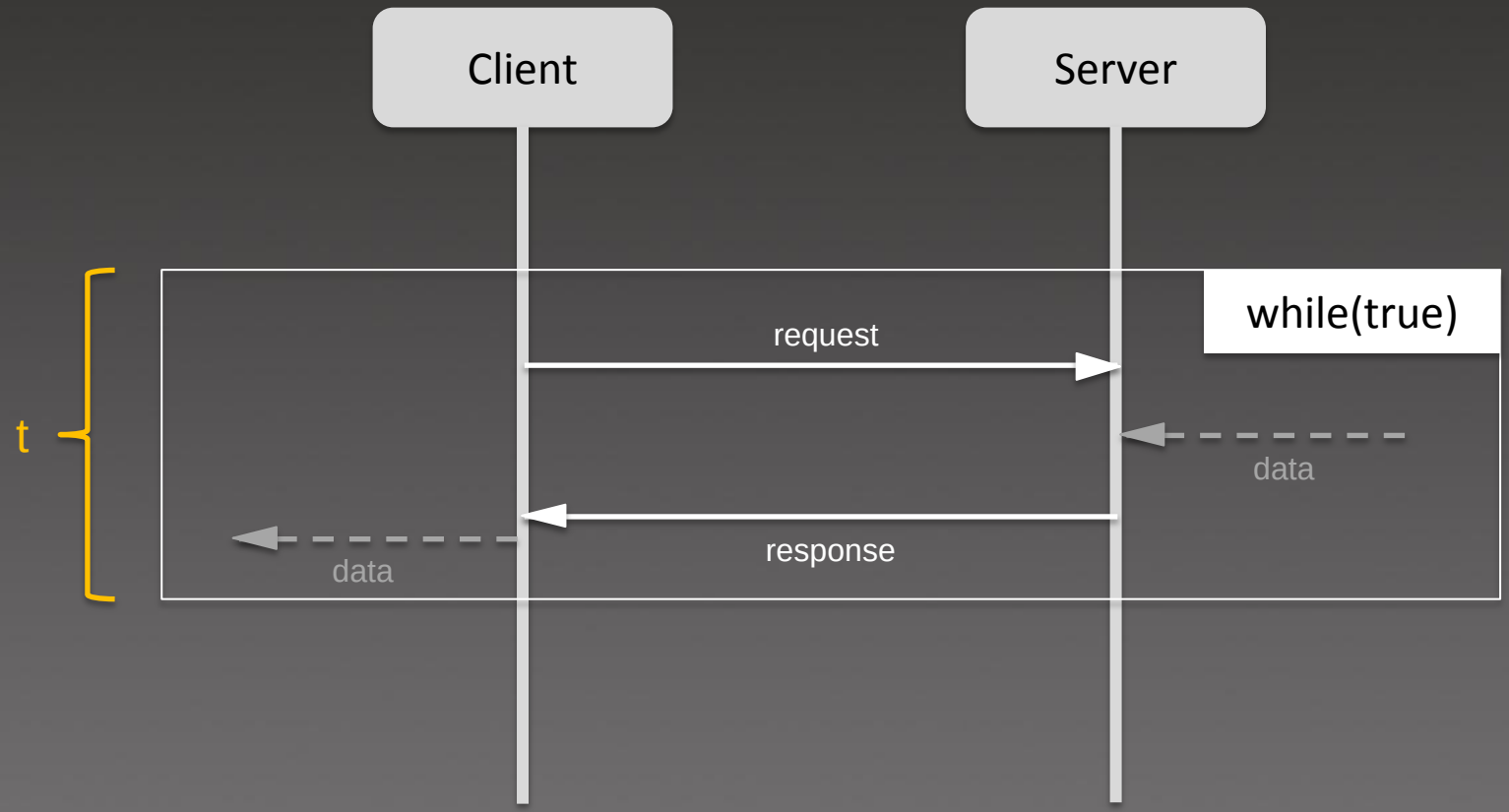
Communication Strategies

- Lösungsansätze
  - Periodic Polling
  - Long Polling
  - Forever Frame
  - Server-Sent Events (SSE)
  - WebSockets

SignalR

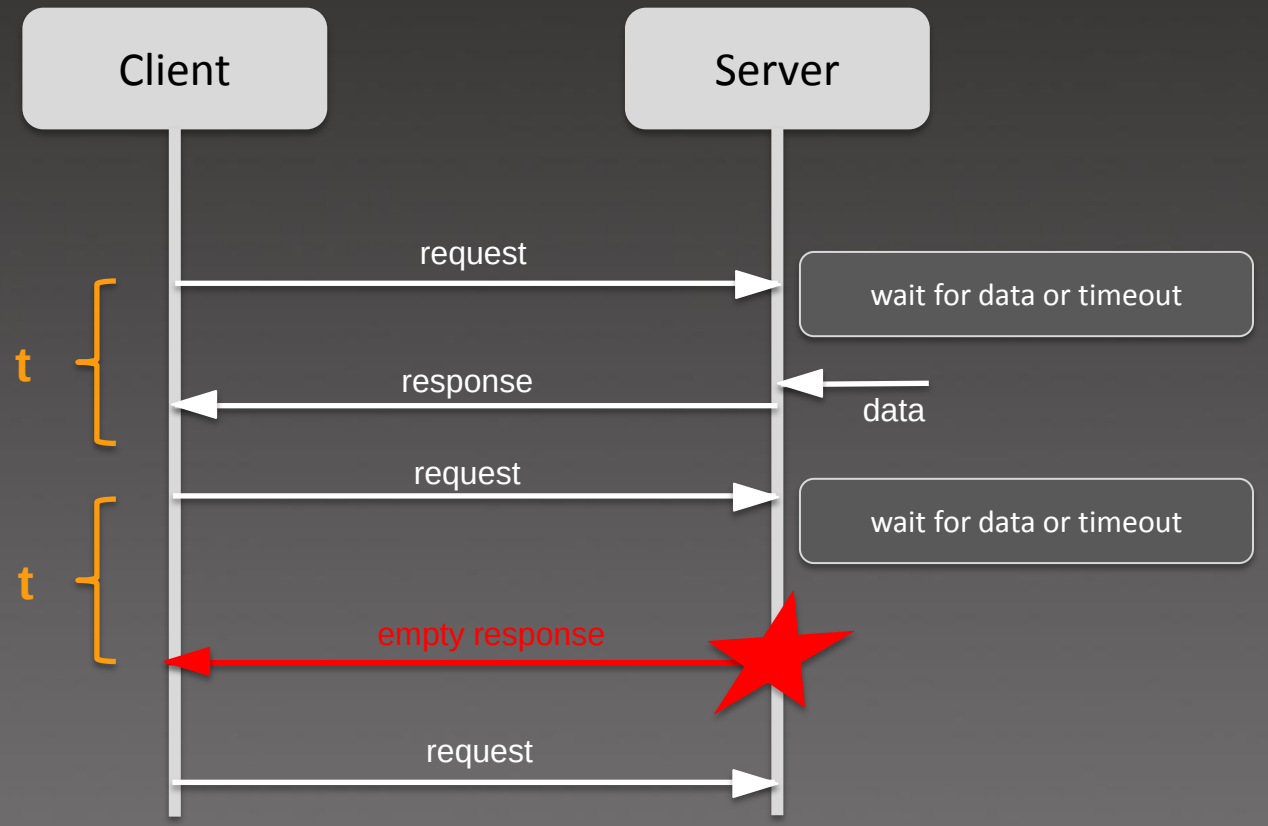
Communication Strategies

- **Periodic Polling**



Communication Strategies

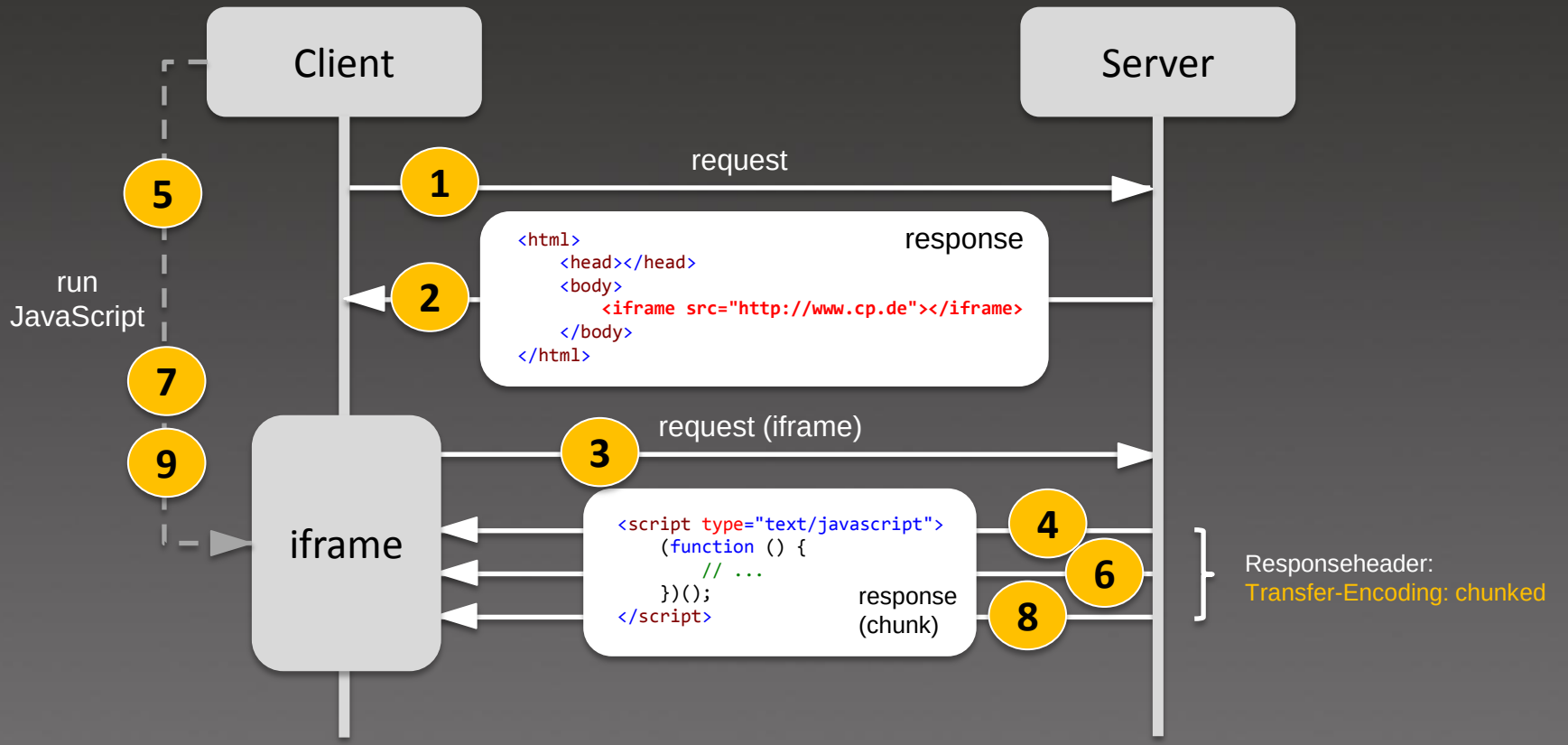
- **Long Polling**



http://msdn.microsoft.com/de-de/magazine/hh882442.aspx

Communication Strategies

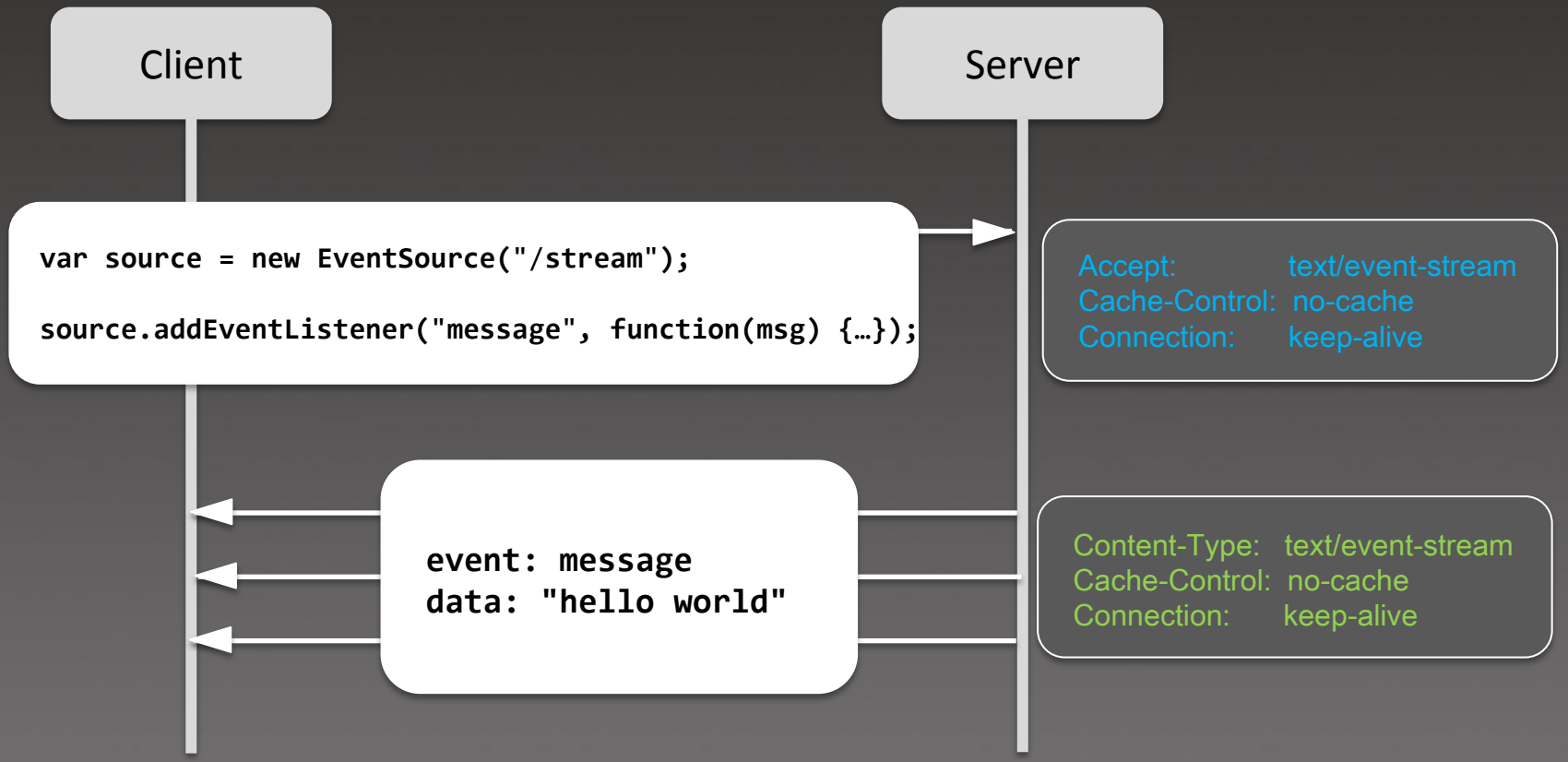
- **Forever Frame (Reverse Ajax)**



http://en.wikipedia.org/wiki/Comet_%28programming%29

Communication Strategies

- **Server-Sent Events**



http://www.html5rocks.com/en/tutorials/eventsource/basics/?redirect_from_locale=de

http://en.wikipedia.org/wiki/Server-sent_events

http://dev.w3.org/html5/eventsource/

Communication Strategies

- **Server-Sent Events**

Server-sent DOM events - Working Draft

Method of continuously sending data from a server to the browser, rather than repeatedly requesting it (EventSource interface, used to fall under HTML5)

*Usage stats:	Global
Support:	64.42%
Partial support:	0.06%
Total:	64.48%

Show all versions	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Blackberry Browser
								2.1	
								2.2	
						3.2		2.3	
						4.0-4.1		3.0	
	8.0	21.0	27.0			4.2-4.3		4.0	
	9.0	22.0	28.0	5.1		5.0-5.1		4.1	7.0
Current	10.0	23.0	29.0	6.0	16.0	6.0-6.1	5.0-7.0	4.2	10.0
Near future	11.0	24.0	30.0	7.0	17.0	7.0			
Farther future		25.0	31.0						

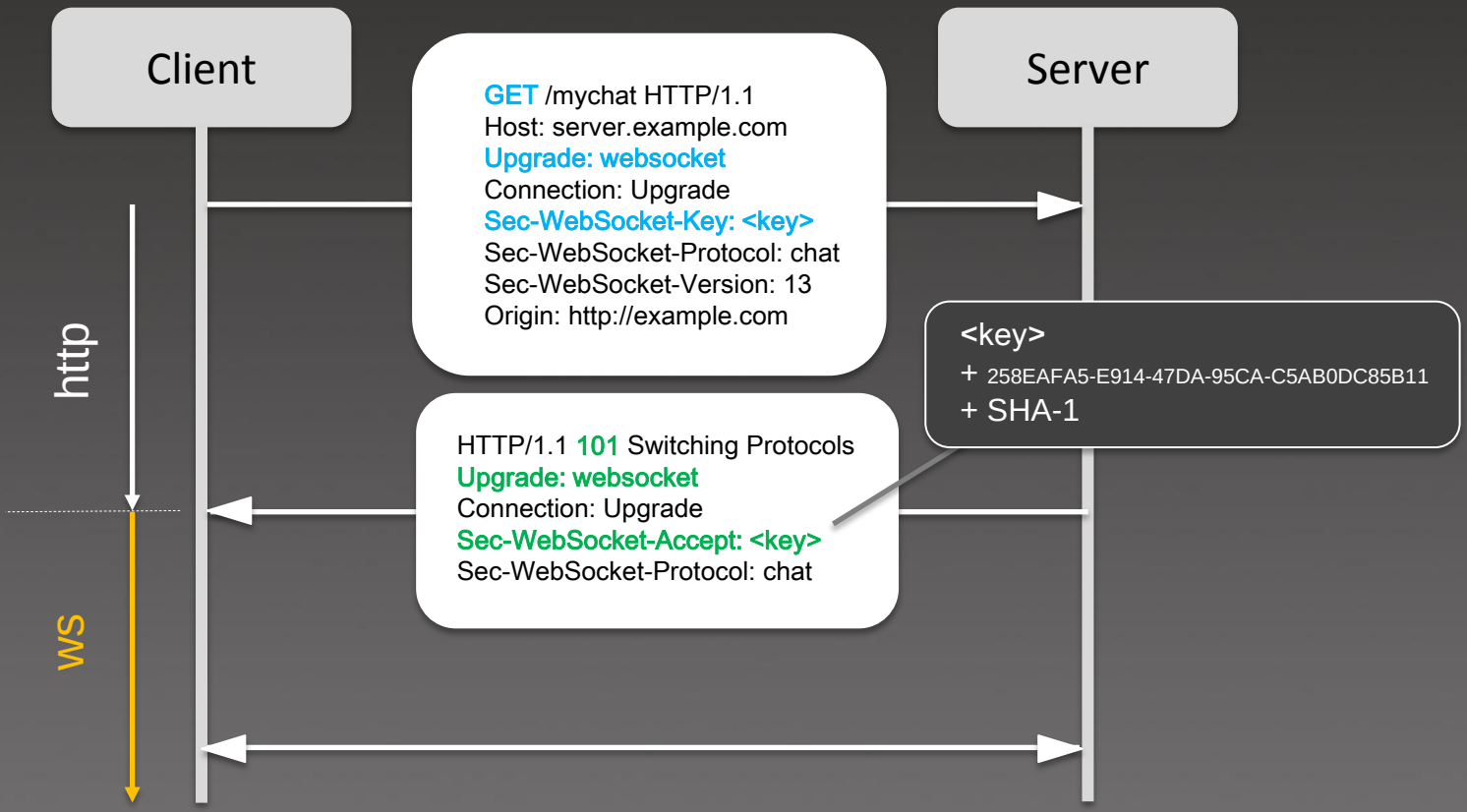
Notes | Known issues (1) | Resources (4) | Feedback | Edit on GitHub

No notes

http://caniuse.com/

Communication Strategies

- **Web Socket**

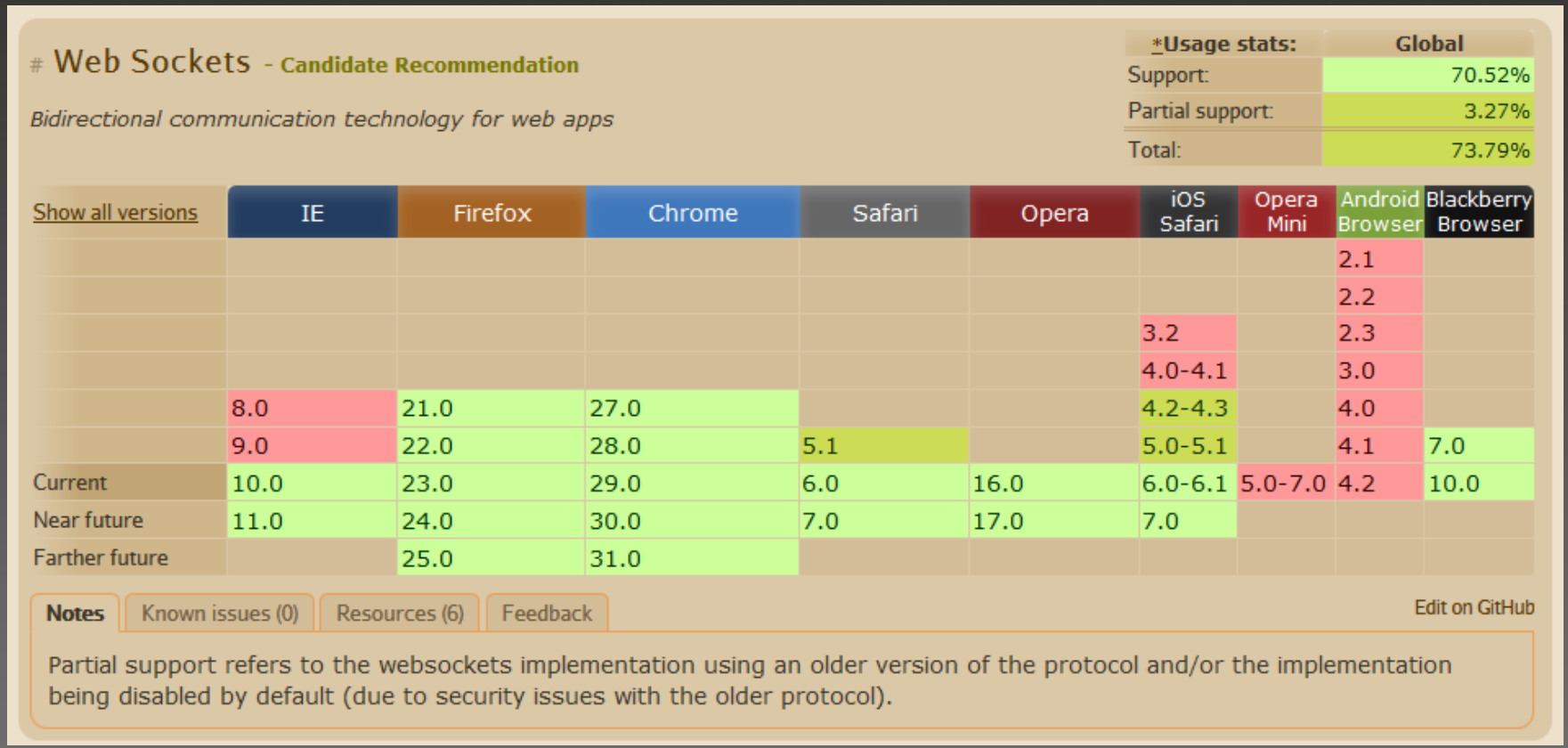


http://en.wikipedia.org/wiki/WebSocket

http://www.html5rocks.com/de/tutorials/websockets/basics/

Communication Strategies

- **Web Socket**



Web Sockets - Candidate Recommendation

Bidirectional communication technology for web apps

*Usage stats:	Global
Support:	70.52%
Partial support:	3.27%
Total:	73.79%

Show all versions	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Blackberry Browser
								2.1	
								2.2	
						3.2		2.3	
						4.0-4.1		3.0	
	8.0	21.0	27.0			4.2-4.3		4.0	
	9.0	22.0	28.0	5.1		5.0-5.1		4.1	7.0
Current	10.0	23.0	29.0	6.0	16.0	6.0-6.1	5.0-7.0	4.2	10.0
Near future	11.0	24.0	30.0	7.0	17.0	7.0			
Farther future		25.0	31.0						

Notes | Known issues (0) | Resources (6) | Feedback — Edit on GitHub

Partial support refers to the websockets implementation using an older version of the protocol and/or the implementation being disabled by default (due to security issues with the older protocol).

http://caniuse.com/

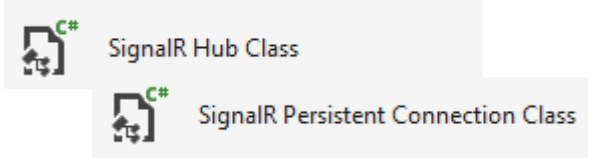
Server-API

Hubs

- Beispiel

```
using Microsoft.AspNet.SignalR;
using Microsoft.AspNet.SignalR.Hubs;

namespace SignalR.Hubs
{
    [HubName("chat")]
    public class ChatHub : Hub
    {
        public void SendMessage(string message)
        {
            Clients.All.newMessage(message);
        }
    }
}
```

ASP.NET and Web Tools 2012.2 update



Hubs

- IHub

```
namespace Microsoft.AspNet.SignalR.Hubs
{
    public interface IHub : IDisposable
    {
        Task OnConnected();
        Task OnReconnected();
        Task OnDisconnected();

        HubConnectionContext Clients { get; set; }

        HubCallerContext Context { get; set; }

        IGroupManager Groups { get; set; }
    }
}
```

Hubs

- Client Property

```
// Represents the calling client
Clients.Caller.newMessage(message);

// All connected clients except the calling client
Clients.Others.newMessage(message);

// All connected clients
Clients.All.newMessage(message);

// Returns a dynamic representation of all clients except the calling client ones specified
Clients.AllExcept("ConnectionId-1", "ConnectionId-2").newMessage(message);

// Returns a dynamic representation of the connection with the specified connection Id
Clients.Client("ConnectionId-1").newMessage(message);

// Returns a dynamic representation of a group
Clients.Group("groupName", "ConnectionId-1", "ConnectionId-2").newMessage(message);

// Returns a dynamic representation of all clients in a group except the calling client
Clients.OthersInGroup("groupName").newMessage("message");
```

Hubs

- Context Property

```
// Gets the connection ID of the calling client
string id = Context.ConnectionId;

// Gets the headers for the request
NameValueCollection headers = Context.Headers;

// Gets the query string for the request
NameValueCollection query = Context.QueryString;

// Gets the IRequest for the current request
IRequest request = Context.Request;

// Gets the cookies for the request
IDictionary<string, Cookie> cookies = Context.RequestCookies;

// Gets the IPricipal for the request
IPrincipal user = Context.User;
```

Hubs

- Groups Property

```
// Registers the client with the given connection id to
// the group with the given group name.
// If the groups doesn't exist, create a new group with
// the given name first.
Groups.Add(Context.ConnectionId, groupName);

// Removes the client with the given connection Id from
// the group with the given group name.
Groups.Remove(Context.ConnectionId, groupName);
```

Routing

- Configuration

Microsoft ASP.NET SignalR System.Web Components
Components for using ASP.NET SignalR in applications hosted on System.Web.

```
public static void RegisterRoutes(RouteCollection routes)
{
      HubConfiguration configuration = new HubConfiguration
      {
            EnableCrossDomain = true,
            EnableDetailedErrors = false,
            EnableJavaScriptProxies = true,
            Resolver = new DefaultDependencyResolver()
      };

      RouteTable.Routes.MapHubs(configuration);

      routes.MapRoute(
            url: "{controller}/{action} ",
            defaults: new { controller = "Home", action = "Index"}
      );
}
```

Registriet die von SignalR benötigten Routes:

~/signalR
~/signalR/hubs
~/signalR/negotiate
~/signalR/transport

Client-API

Proxies

- Auto generated
- Static
- Programmatically

Proxies

- Auto generated Proxy

```
HubConfiguration configuration = new HubConfiguration
{
      EnableJavaScriptProxies = true // Default
};

RouteTable.Routes.MapHubs(configuration);
```

Hubs müssen vor(!) allen anderen Routes definiert werden.

Microsoft ASP.NET SignalR JavaScript Client
JavaScript client for ASP.NET SignalR.

```
<script src="~/Scripts/jquery.signalR-1.0.1.js"></script>
<script src="signalr/hubs"></script>
<script type="text/javascript">

    $(function () {
          var chat = $.connection.chat;
          $.connection.hub.start({transport: "longPolling"});
    });

</script>
```

Liefert JavaScript Code, der sofort ausgeführt wird und die Hubs am Client registriert

Löst negotiation aus und stellt die Verbindung über eine Technologie (Long Polling etc.) her

"webSockets"
"foreverFrame"
"serverSentEvents"
"longPolling"

Proxies

- Client API

```
// Connection for all hubs
var connection = $.connection.hub;

// Returns a client side hub from the proxy
var chatHub = connection.proxies.chat;
var chatHub = $.connection.chat;

// Register client side callback (called from Server via Clients.All.addMessage("text");
chatHub.client.addMessage = function(msg) { /* ... */ };

// Call a server side method
chatHub.server.sendMessage("text")
      .done(function () { /* ... */ })
      .fail(function () { /* ... */ });

// Start the connection (starts negotiation)
$.connection.hub.start({ transport: "longPolling" })
      .done(function () { /* ... */ })
      .fail(function () { /* ... */ });
```

Proxies

- Static Proxy (1/2)

Microsoft ASP.NET SignalR Utilities
Command line utilities for ASP.NET SignalR, including performance counter installation and Hub JavaScript proxy generation.

```
C:\Work\SignalR\packages\Microsoft.AspNet.SignalR.Utils.1.0.1\tools>dir
 Datenträger in Laufwerk C: ist OSDisk
 Volumeseriennummer: A473-4523

 Verzeichnis von C:\Work\SignalR\packages\Microsoft.AspNet.SignalR.Utils.1.0.1\tools

23.04.2013  16:30    <DIR>          .
23.04.2013  16:30    <DIR>
23.04.2013  16:30            45.752 signalr.exe
               1 Datei(en),
               2 Verzeichnis(se), 390.725.554.176 Bytes frei

C:\Work\SignalR\packages\Microsoft.AspNet.SignalR.Utils.1.0.1\tools>_
```

```
>signalr.exe ghp /url:http://localhost:20716/
```

```
23.04.2013  16:47    <DIR>          .
23.04.2013  16:47    <DIR>
23.04.2013  16:47             1.274 server.js
23.04.2013  16:30            45.752 signalr.exe
               2 Datei(en),         49.026 Bytes
               2 Verzeichnis(se), 390.673.203.200 Bytes frei
```

Erzeugtes File muss in View Referenziert werden !!!

Proxies

- Static Proxy (2/2)

```
<script src="~/Scripts/jquery.signalR-1.0.1.js"></script>
<script src="~/Scripts/server.js"></script>
<script type="text/javascript">

    $(function () {
            var chat = $.connection.chat;
            $.connection.hub.start();
    });

</script>
```

Liefert ein statische JavaScript-File mit Hubs und Code, der sofort ausgeführt wird und die Hubs am Client registriert

Löst negotiation aus und stellt die Verbindung über eine Technologie (Long Polling etc.) her

Proxies

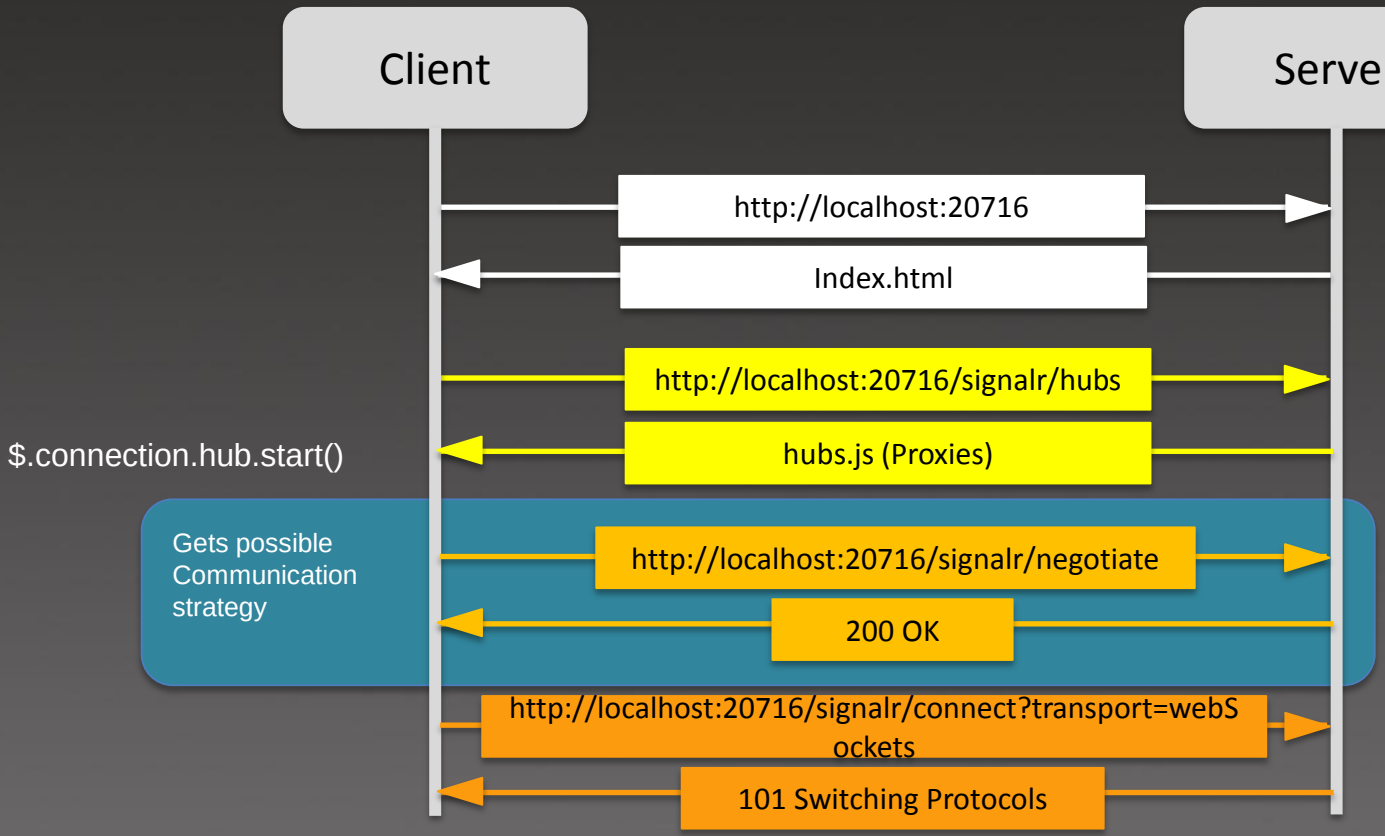
- Programmatic Proxy

```
<script src="~/Scripts/jquery-1.7.1.js"></script>
<script src="~/Scripts/jquery.signalR-1.0.1.js"></script>
<script type="text/javascript">

    $(function () {

            var connection   = $.hubConnection();
            var proxy        = connection.createHubProxy("chat");

            proxy.on("broadcast", function (message) {
                  $("#output").append(message);
            });

            connection.start().done(function () {
                  $("#send").click(function () {
                        proxy.invoke("send", $("#message").val())
                  } );
            });
    });

</script>
```

Agenda

Roundtrip

Roundtrip



1. Web Socket
2. Server-Sent Events
3. Forever Frame
4. Long Polling

GET localhost:20716	200 OK
GET hubs	200 OK
GET negotiate?_=1366723257766	200 OK
GET connect?transport=webSo...A%22chat%22%7D%	101 Switching Protocols

Ausblick

Ausblick

- SignalR Clients
  - Browser
  - Windows Phone 8
  - WPF / WebForms

- Hosting
  - ASP.NET
  - Self Hosting
  - Windows Azure Websites

Quellen

Longpolling:
http://msdn.microsoft.com/de-de/magazine/hh882442.aspx

Forever Frame:
http://en.wikipedia.org/wiki/Comet_%28programming%29

Server Sent Events:
http://www.w3schools.com/html/html5_serversentevents.asp
http://www.worldwidewhat.net/2011/07/using-server-sent-events-with-asp-net/
http://html5doctor.com/server-sent-events/
http://www.html5rocks.com/en/tutorials/eventsource/basics/?redirect_from_locale=de
http://en.wikipedia.org/wiki/Server-sent_events
http://dev.w3.org/html5/eventsource/

Web Sockets:
http://en.wikipedia.org/wiki/WebSocket
http://www.html5rocks.com/de/tutorials/websockets/basics/

SignalR:
http://signalr.net/
http://www.asp.net/signalr/overview/hubs-api/hubs-api-guide-javascript-client
http://www.asp.net/signalr/overview/getting-started/tutorial-getting-started-with-signalr-and-mvc-4

Vielen Dank !

Tobias Krügel
conplement AG
Tobias.kruegel@conplement.de

