



Progress Webinar: GUI trifft auf Web mit Chromium Embedded

Mike Fechner, Consultingwerk

Consultingwerk

- Independent IT consulting organization
- Focusing on **OpenEdge** and **related technology**
- Located in Cologne, Germany, subsidiaries in UK and Romania
- Customers in Europe, North America, Australia and South Africa
- Vendor of developer tools and consulting services
- Specialized in GUI for .NET, Angular, OO, Software Architecture, Application Integration
- Experts in OpenEdge Application Modernization



Mike Fechner

- Director, Lead Modernization Architect and Product Manager of the SmartComponent Library and WinKit
- Specialized on object oriented design, software architecture, desktop user interfaces and web technologies
- 31 years of Progress experience (V5 ... OE12)
- Active member of the OpenEdge community
- Frequent speaker at OpenEdge related conferences around the world



SmartComponent Library Entwicklerframework

- Unterstützt das Sichern der Investitionen in ihre OpenEdge basierenden Anwendungen
- Framework wurde entworfen sowohl um bestehende OpenEdge Anwendungen zu modernisieren als auch um eine solide und zukunftssichere Grundlage für neue Anwendungen bereit zu stellen
- In der Cloud und on-premise
- OERA, CCS
- Die Architektur des Frameworks ist auf Integration mit kommenden Technologien und die Umsetzung neuer Business Requirements ausgelegt

User Interface Flexibilität

- Windows Desktop User Interfaces mit .NET
- Angular Web Anwendungen (Kendo UI)
- Mobile Anwendungen (NativeScript)
- Offene Schnittstellen (z.B. RESTful)
- Partner User Interfaces (z.B. AKIOMA)



Kendo UI
THE ART OF WEB DEVELOPMENT



Agenda

- **Modernisierung von OpenEdge Anwendungen**
- CefSharp
- CefSharp vs. Microsoft WebView2 SDK
- Nutzen des Web Browser Controls in der Anwendung
- Hybride Anwendungsarchitektur
- Authentifizierung
- Navigation innerhalb einer SPA (Angular)
- Interaktion mit einer SPA (Angular)

Modernisierungsziele

- Funktionale Anforderungen
 - Hinzufügen neuer Fach-Module
 - Verändern der bestehenden Fach-Logik
- Nichtfunktionale Anforderungen
 - Neues User-Interface / User-Experience
 - Moderner Desktop
 - Web
 - Mobile
 - Neue Architektur
 - Performance (z.B. im WAN), vermeiden von Citrix

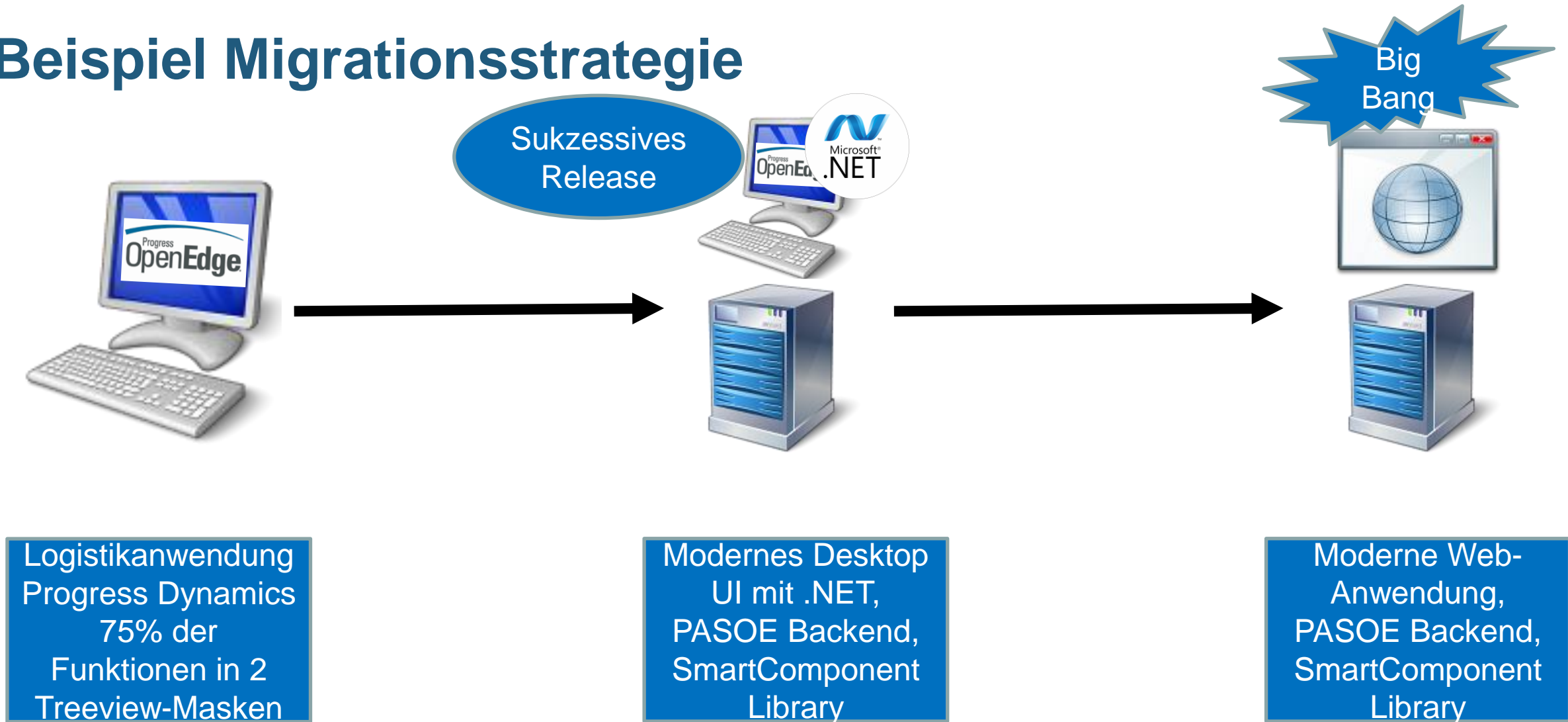
Was macht die
Anwendung

Wie löst die
Anwendung die
Anforderungen

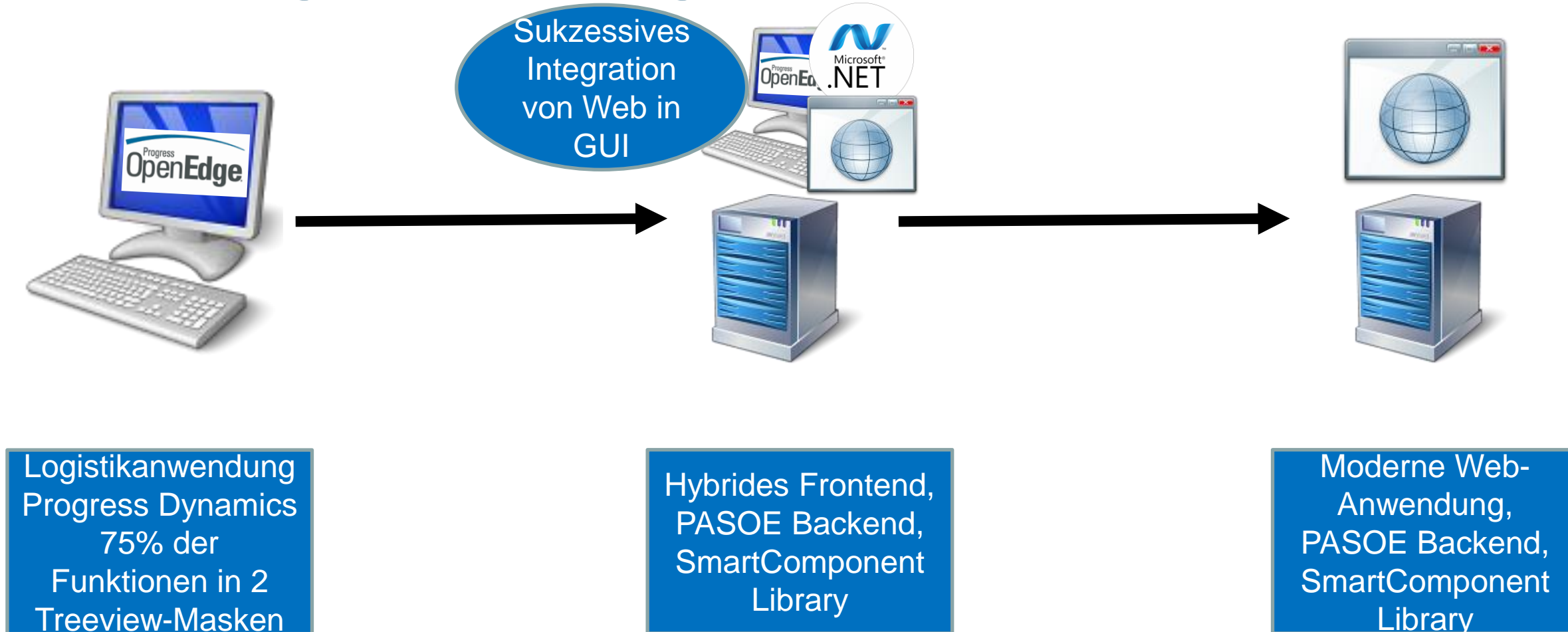
Modernisierungsstrategien

- Modernisierung der vollständigen Anwendung
 - Von TTY oder ABL GUI zu modernem Desktop, Web oder Mobile
 - Entscheidung über die „finale“ UI Technologie
 - Muss es immer ein Web Frontend sein?
 - Direkt vom Status-Quo zur finalen Lösung – oder über Zwischenschritt
- Umsetzen erster neuer Features in einer neuen Technologie
 - Mobiler Client für Teile der Anwendung
 - REST/RESTful Schnittstellen für Teile der Anwendungslogik
 - Reduzieren des Risikos
 - Erfahrung mit neuer Technologie sammeln

Beispiel Migrationsstrategie



Beispiel Migrationsstrategie



Agenda

- Modernisierung von OpenEdge Anwendungen
- **CefSharp**
- CefSharp vs. Microsoft WebView2 SDK
- Nutzen des Web Browser Controls in der Anwendung
- Hybride Anwendungsarchitektur
- Authentifizierung
- Navigation innerhalb einer SPA (Angular)
- Interaktion mit einer SPA (Angular)

CefSharp

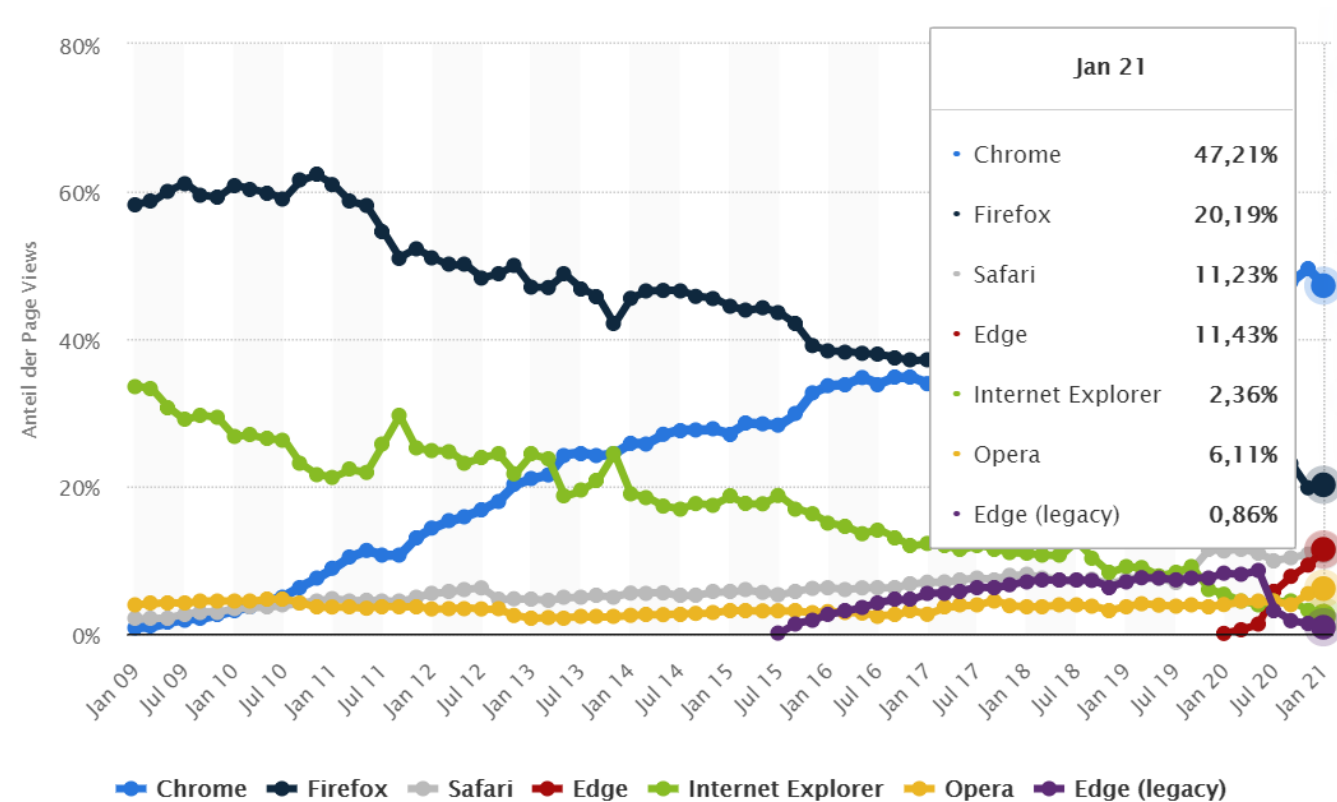
- **Chromium**
 - Basiert auf der Open-Source Code-Basis des Google Chrome Web Browsers
- **Embedded**
 - Browser Komponente konzipiert zur Einbindung in eigene Anwendung
- **Framework**
 - System zur Entwicklung eigener Browser-Anwendung
 - aber keine selbständige Browser-Anwendung

CefSharp

- Chromium Embedded Framework (CEF)
- Open-Source Projekt seit 2009
- BSD Lizenz
- CefSharp ist populäres Open-Source Projekt und bindet CEF in .NET Anwendungen ein
- Basierend auf Google's Open-Source Chromium, der Basis von Chrome
- CEF ist programmiert in C++; CefSharp ist programmiert in C#
- **WinForms** und WPF Web Browser Control
- OpenEdge GUI Anwendung kann WinForms Control nutzen

Web Browser Marktanteile, Deutschland, Januar '21

- Quelle: Statista



- <https://de.statista.com/statistik/daten/studie/13007/umfrage/marktanteile-der-browser-bei-der-internetnutzung-in-deutschland-seit-2009/>

CefSharp

- CEF/CefSharp Defacto Standard für Browser Einbindung in Anwendungen
- Nutzung in Anwendungen und Games wie z.B.: Adobe Autodesk, Amazon, Battle.net, Bitdefender, BlueStacks, Evernote, Facebook Messenger, GTA Online, Kaspersky, Intel League of Legends, MATLAB, Power BI, QuarkXPress, Second Life, Solidworks, Spotify, Steam, Unity3D, Unreal Engine usw.

Bring your own bottle browser

The image shows the acronym 'BYOB' rendered in a stylized, glowing neon font. The letters are outlined in a bright cyan color and filled with a vibrant magenta. The background is a dark, textured surface that resembles a brick wall, with the lighting creating a strong glow around the letters.

BYOB

REAM > SmartComponentLibrary > Develop123 > ABL > Assemblies > CefSharp > x64 >

Name	Änderungsdatum	Typ	Größe
cache	10.02.2021 10:22	Dateiordner	
locales	10.02.2021 10:22	Dateiordner	
swiftshader	10.02.2021 10:22	Dateiordner	
cef.pak	30.10.2020 06:32	PAK-Datei	1.902 KB
cef_100_percent.pak	30.10.2020 06:36	PAK-Datei	210 KB
cef_200_percent.pak	30.10.2020 06:36	PAK-Datei	284 KB
cef_extensions.pak	30.10.2020 06:36	PAK-Datei	1.275 KB
CefSharp.BrowserSubprocess.Core.dll	03.02.2021 13:16	Anwendungserwe...	1.213 KB
CefSharp.BrowserSubprocess.Core.pdb	03.02.2021 13:16	Program Debug D...	6.292 KB
CefSharp.BrowserSubprocess	03.02.2021 13:16	Anwendung	7 KB
CefSharp.BrowserSubprocess.exe	27.01.2021 10:32	Configuration-Qu...	1 KB
CefSharp.BrowserSubprocess.pdb	03.02.2021 13:16	Program Debug D...	16 KB
CefSharp.Core.dll	03.02.2021 13:15	Anwendungserwe...	1.861 KB
CefSharp.Core.pdb	03.02.2021 13:15	Program Debug D...	7.588 KB
CefSharp.Core	03.02.2021 13:16	XML-Quelldatei	89 KB
CefSharp.dll	03.02.2021 13:15	Anwendungserwe...	1.000 KB
CefSharp.Example.dll	03.02.2021 13:16	Anwendungserwe...	947 KB
CefSharp.Example.pdb	03.02.2021 13:16	Program Debug D...	228 KB
CefSharp.OffScreen.dll	01.12.2020 13:41	Anwendungserwe...	31 KB
CefSharp.OffScreen.pdb	01.12.2020 13:41	Program Debug D...	66 KB
CefSharp.pdb	03.02.2021 13:15	Program Debug D...	1.884 KB
CefSharp.WinForms.dll	03.02.2021 13:16	Anwendungserwe...	31 KB
CefSharp.WinForms.pdb	03.02.2021 13:16	Program Debug D...	60 KB
CefSharp.WinForms	03.02.2021 13:16	XML-Quelldatei	51 KB
CefSharp	03.02.2021 13:15	XML-Quelldatei	1.344 KB
chrome_elf.dll	30.10.2020 06:24	Anwendungserwe...	1.006 KB
Consultingwerk.CefSharpSupport.dll	03.02.2021 13:49	Anwendungserwe...	11 KB

Embedded Browser

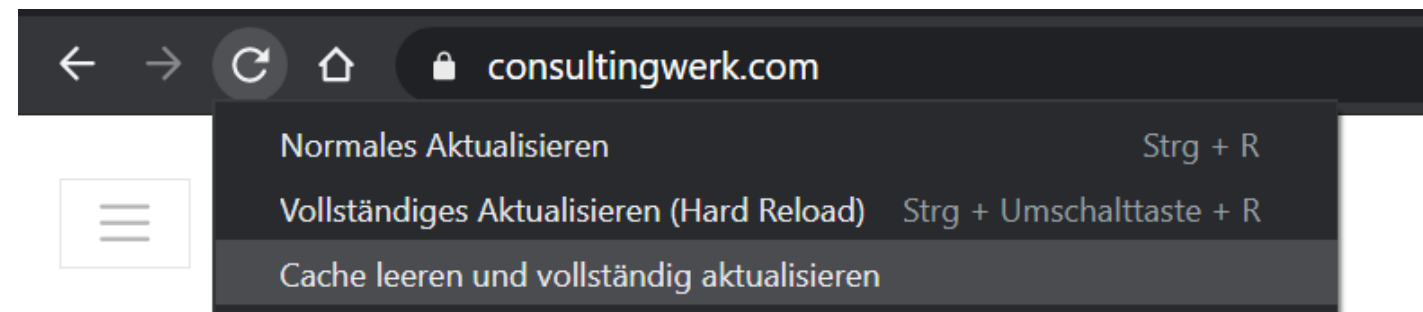
- C++ und C# DLL's im Anwendungsverzeichnis
- Browser Sub-Prozess im Anwenderverzeichnis
- Keine Abhängigkeit zu Google Chrome Installation
- Einzige Abhängigkeiten Visual C++ Runtime und .NET Framework

Vorzüge eines Embedded Browsers

- Intensive Integration mit eigener Anwendung möglich (wie später in dieser Präsentation noch gezeigt)
- Moderner Web Browser
- Release Upgrades unter eigener Kontrolle
 - Keine Abhängigkeit vom User
 - Verzögern von Browser-Updates, falls erforderlich
 - keine Überraschung durch plötzliche Secure-Cookie Anforderungen
 - keine Überraschung durch plötzliches Verwerfen von SSL Root Zertifikaten
 - Test der eigenen Web Anwendung mit neuer Browser-Version vor deren Rollout als embedded Komponente

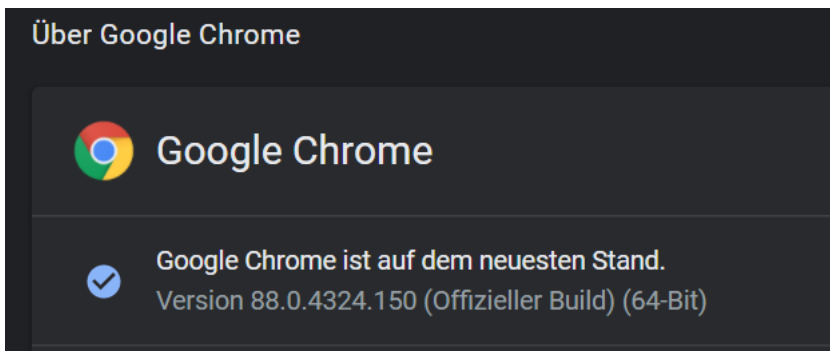
Kontrolle über den Browser-Cache

- Browser Cache von elementarer Bedeutung für Performance
- Web-Entwickler zu Tester: „Hast Du den Browser-Cache geleert?“, „Hast Du den Cache wirklich **vollständig** geleert?“, „Öffne mal den Browser-Debugger und dann Rechts-Click auf den Refresh-Button!“
- CefSharp Konfiguration steuert Cache-Verzeichnis
- Kann z.B. bei Anwendungsupdates automatisiert/kontrolliert gelöscht werden
- CefSharp kann komplett auf Cache on Disk verzichten



Stabile Versionen verfügbar kurz nach Chrome

- Stand 10.02.2021



- Chrome Version 88.0 wurde am
- 19. Januar '21 released

CefSharp Builds:

Releases

CI Builds

Every commit on `master` produces a `Nuget` package. Use at your own risk!

- WinForms v88.2.40-RCI3909
- Wpf v88.2.40-RCI3909
- OffScreen v88.2.40-RCI3909

Pre-release

- WinForms v88.2.40-pre
- Wpf v88.2.40-pre
- OffScreen v88.2.40-pre

Stable

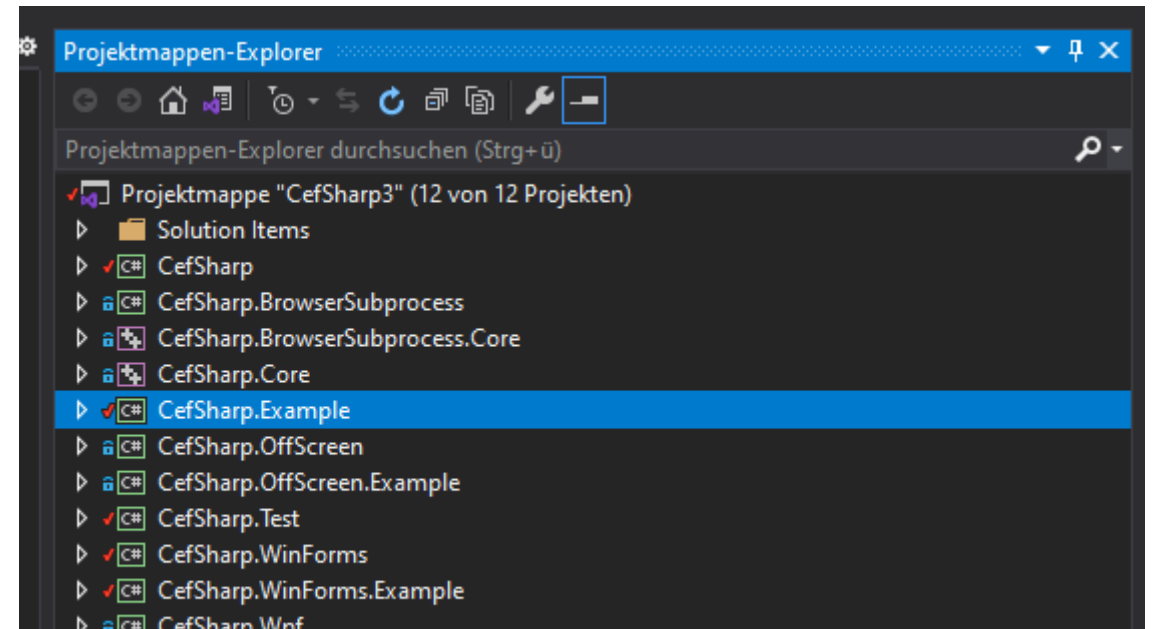
- WinForms v87.1.132
- Wpf v87.1.132
- OffScreen v87.1.132

Download CefSharp

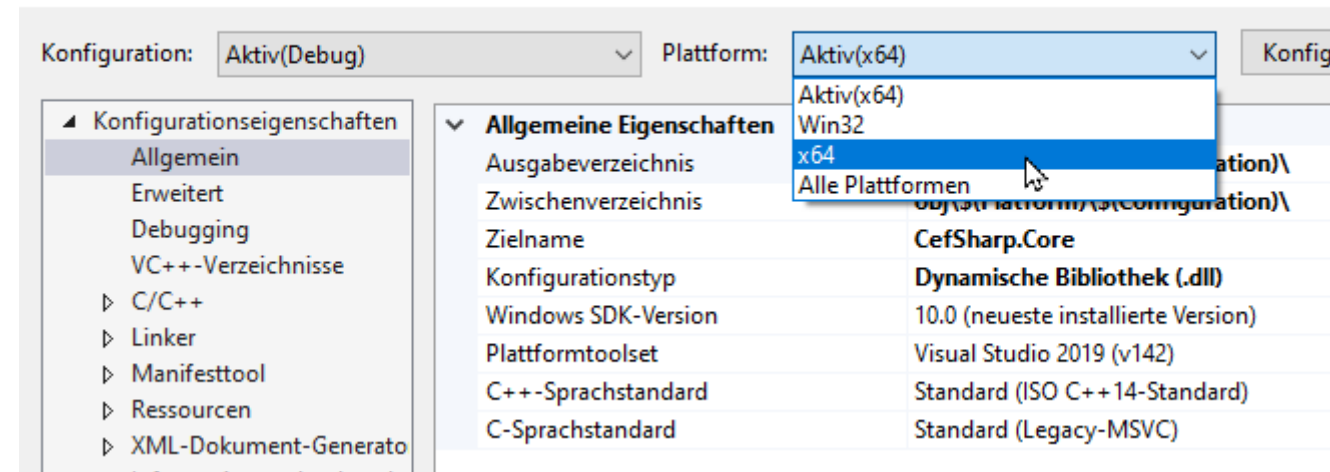
- Quellcode via git clone von <https://github.com/cefsharp/CefSharp>
- Binaries <https://www.nuget.org/packages/CefSharp.WinForms/>
- Wir nutzen den Quellcode – debuggen hilft gelegentlich beim Verständnis
- Enthält die Beispielanwendung als Referenz
- Für Integration in OpenEdge haben wir eine eigene Support Assembly geschrieben
 - zur Event-Synchronisation aus Hintergrund-Threads
 - Settings Wrapper, etc.

Visual Studio

- CefSharp Projektmappe
- Wirklich unkompliziert im Build!
Selbst in der Visual Studio
Community Edition
- Projekte müssen getrennt für 64
oder 32 bit Runtime (prowin.exe
oder prowin32.exe) kompiliert
werden (verschiedene DLL's)



CefSharp.Core-Eigenschaftenseiten



Thread-Synchronisation von Events

```
public void RegisterEvents (ChromiumWebBrowser browser)
{
    try
    {
        this.WebBrowser = browser;

        browser.AddressChanged += Browser_AddressChanged;
        browser.ConsoleMessage += Browser_ConsoleMessage;
        browser.FrameLoadEnd += Browser_FrameLoadEnd;
        browser.FrameLoadStart += Browser_FrameLoadStart;
        browser.IsBrowserInitializedChanged += Browser_IsBrowserInitializedChanged;
        browser.JavascriptMessageReceived += Browser_JavascriptMessageReceived;
        browser.LoadError += Browser_LoadError;
        browser.LoadingStateChanged += Browser_LoadingStateChanged;
        browser.StatusMessage += Browser_StatusMessage;
        browser.TitleChanged += Browser_TitleChanged;
    }
    catch (NullReferenceException err)
    {
        MessageBox.Show(err.ToString(), "RegisterEvents");
    }
}
```

```
protected void OnTitleChanged (TitleChangedEventArgs e)
{
    if (this.InvokeRequired)
        this.BeginInvoke (new Action<TitleChangedEventArgs> (OnTitleChanged), new[] { e });
    else
        this.TitleChanged (this.WebBrowser, e);
}
```

Weitere CefSharp Features

- Integration mit Google Chrome Website Debugger Console
 - Browser Log Console
 - Tracing von http-Requests
 - Debugging von JavaScript Code
- Head-less Browsing möglich, nutzen des Browsers ohne Oberfläche
 - z.B. Grundlage von Unit-Test Frameworks für Webseiten
- Freies Setzen des „User-Agents“ (Browser Kennung)

- Klassische Controls, wie Back-Button, Adressleiste sind kein Teil von CefSharp und werden als Teil der Anwendung umgesetzt

Agenda

- Modernisierung von OpenEdge Anwendungen
- CefSharp
- **CefSharp vs. Microsoft WebView2 SDK**
- Nutzen des Web Browser Controls in der Anwendung
- Hybride Anwendungsarchitektur
- Authentifizierung
- Navigation innerhalb einer SPA (Angular)
- Interaktion mit einer SPA (Angular)

WebView

- Klassisches WebView: Active X Control oder .NET Control auf Basis von Internet Explorer
- Weit verbreitet in OpenEdge Anwendungen, z.B. auch als PDF Viewer
- Kein Support für moderne Web-Anwendungen
- Version abhängig vom installierten Internet Explorer
- Allerspätestens seit Release von Microsoft Edge „retired“

WebView2

- Basiert wie Edge auf Google's Chromium
- Vergleichbare Integrationsfeatures wie CefSharp
- **Scheint** ein wenig einfacher in der Nutzung ...
- Kein Webseiten Debugger verfügbar, lediglich Browser Console
- Nicht weit verbreitet, WebView2 .NET verfügbar seit Mitte Januar '21, C++ Komponente (nicht in OpenEdge nutzbar) seit Ende Oktober '20
- Keine Embedded Engine, wird als OS Komponente installiert
- Keine Kontrolle über Browser-Version
- Muss Reifeprüfung noch ablegen

Agenda

- Modernisierung von OpenEdge Anwendungen
- CefSharp
- CefSharp vs. Microsoft WebView2 SDK
- **Nutzen des Web Browser Controls in der Anwendung**
 - Hybride Anwendungsarchitektur
 - Authentifizierung
 - Navigation innerhalb einer SPA (Angular)
 - Interaktion mit einer SPA (Angular)

Nutzung von CefSharp

- Initialisierung mittels CefSettings Klasse

```
namespace CefSharp
{
    ... public abstract class CefSettingsBase : IDisposable
    {
        ... public CefSettingsBase();
        ... ~CefSettingsBase();

        ... public CefSharp.LogSeverity LogSeverity { get; set; }
        ... public string LogFile { get; set; }
        ... public string ResourcesDirPath { get; set; }
        ... public string LocalesDirPath { get; set; }
        ... public string Locale { get; set; }
        ... public bool IgnoreCertificateErrors { get; set; }
        ... public string UserDataPath { get; set; }
        ... public string RootCachePath { get; set; }
        ... public string CachePath { get; set; }
        ... public string BrowserSubprocessPath { get; set; }
        ... public bool MultiThreadedMessageLoop { get; set; }
        ... public bool ExternalMessagePump { get; set; }
        ... public bool CommandLineArgsDisabled { get; set; }
        ... public string JavascriptFlags { get; set; }
        ... public bool PackLoadingDisabled { get; set; }
        ... public string ProductVersion { get; set; }
        ... public int RemoteDebuggingPort { get; set; }
        ... public virtual Internals.CommandLineArgDictionary CefCommandLineArgs { get; }
        ... public virtual uint BackgroundColor { get; set; }
        ... public string AcceptLanguageList { get; set; }
        ... public bool PersistUserPreferences { get; set; }
        ... public string ApplicationClientIdForFileScanning { get; set; }
        ... public bool WindowlessRenderingEnabled { get; set; }
        ... public string UserAgent { get; set; }
        ... public int UncaughtExceptionStackSize { get; set; }
        ... public bool PersistSessionCookies { get; set; }
        ... public System.Collections.Generic.IEnumerable<CefSharp.CefCustomScheme> CefCustomSchemes {
            ... public void DisableGpuAcceleration();
            ... public sealed override void Dispose();
            ... public void EnablePrintPreview();
            ... public void RegisterScheme(CefSharp.CefCustomScheme cefCustomScheme);
            ... public void SetOffScreenRenderingBestPerformanceArgs();
            ... protected virtual void Dispose(bool A_0);
        }
    }
}
```

Code-Sample

```
/**
 * Purpose: Constructor for the BrowserForm class
 * Notes:
 * @param pcUrl Starting-URL to set the browser to
 */
CONSTRUCTOR PUBLIC BrowserForm (pcUrl AS CHARACTER):

    InitializeComponent().

    oBrowser = NEW ChromiumWebBrowser().

    oBrowser:Dock = DockStyle:Fill.
    browserPanel:ClientArea:Controls:Add (oBrowser) .
    chromiumWebBrowserEventHelper1:RegisterEvents(oBrowser).

    oBrowser:Load(pcUrl).

END CONSTRUCTOR.
```

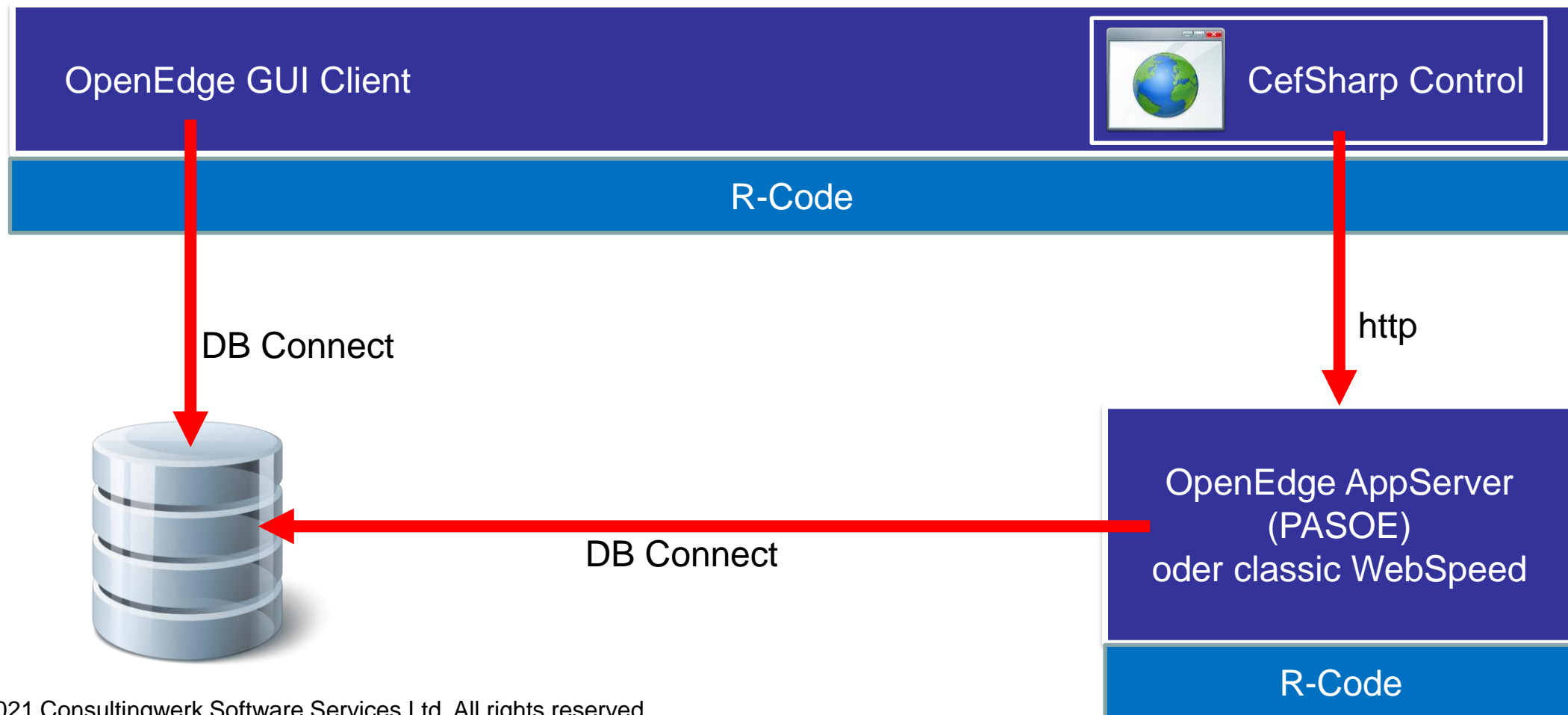
Demo

- DHL Tracking mit Übergabe der Sendungsnummer
- Einbettung Dokumentations-Webseite (Confluence)

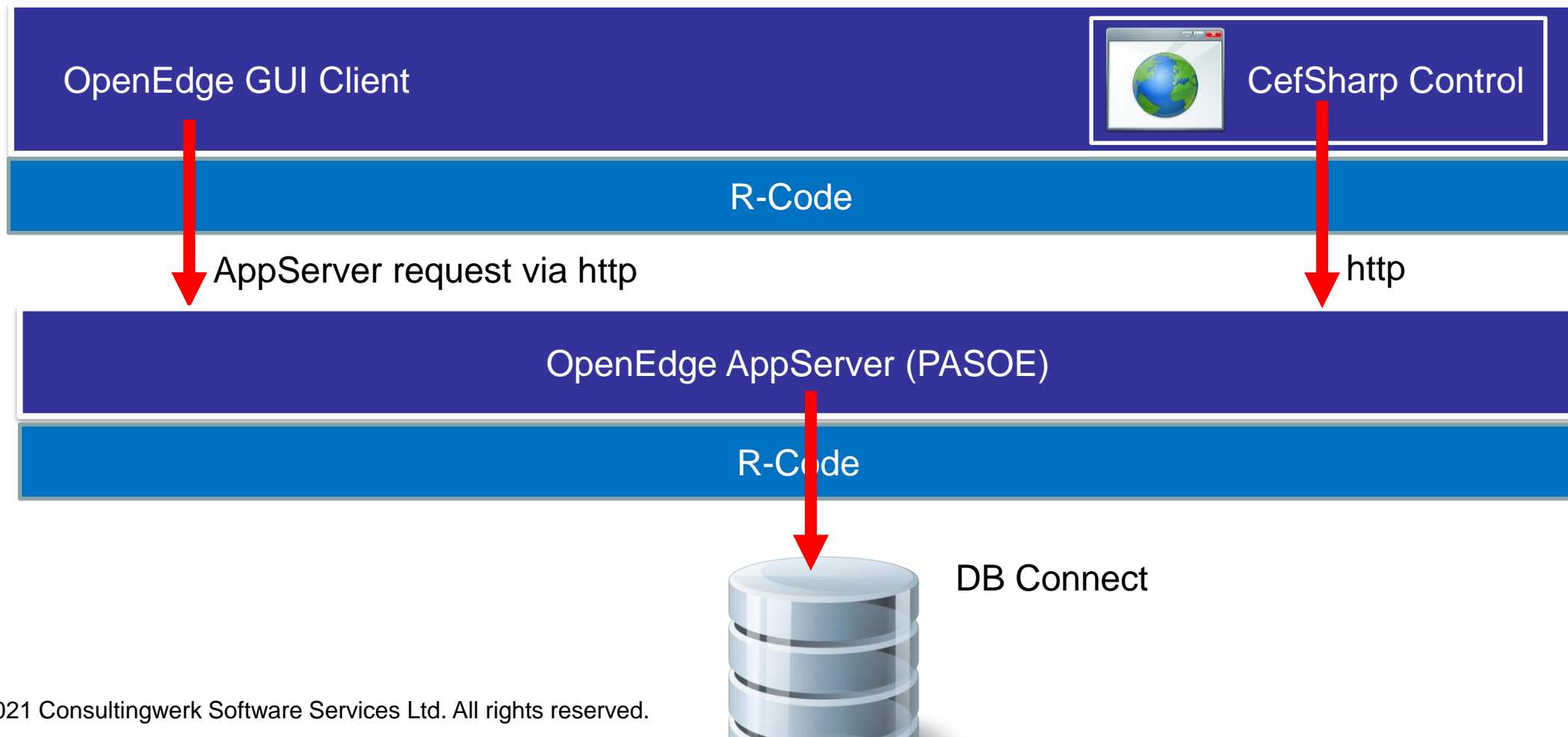
Agenda

- Modernisierung von OpenEdge Anwendungen
- CefSharp
- CefSharp vs. Microsoft WebView2 SDK
- Nutzen des Web Browser Controls in der Anwendung
- **Hybride Anwendungsarchitektur**
 - Authentifizierung
 - Navigation innerhalb einer SPA (Angular)
 - Interaktion mit einer SPA (Angular)

Architektur Integration in Fat Client OpenEdge GUI



Architektur Integration in Fat Client OpenEdge GUI



Agenda

- Modernisierung von OpenEdge Anwendungen
- CefSharp
- CefSharp vs. Microsoft WebView2 SDK
- Nutzen des Web Browser Controls in der Anwendung
- Hybride Anwendungsarchitektur
- **Authentifizierung**
- Navigation innerhalb einer SPA (Angular)
- Interaktion mit einer SPA (Angular)

Authentifizierung

- Technisch gesehen stellen die Desktop Anwendung und die darin laufender Web Anwendung zwei Anwendungen dar
- Aus Sicht des Anwenders sollten die beiden Anwendungen weitestgehend verschmelzen
- Mehrfache Anmeldung (zuerst an Desktop), dann in der Web Anwendung würde als störend empfunden
- Im schlimmsten Falle auch mehrfache Anmeldung in der Browseranwendung, wenn das CefSharp Fenster mehrfach geschlossen wird

Mögliche Lösung mit Securable

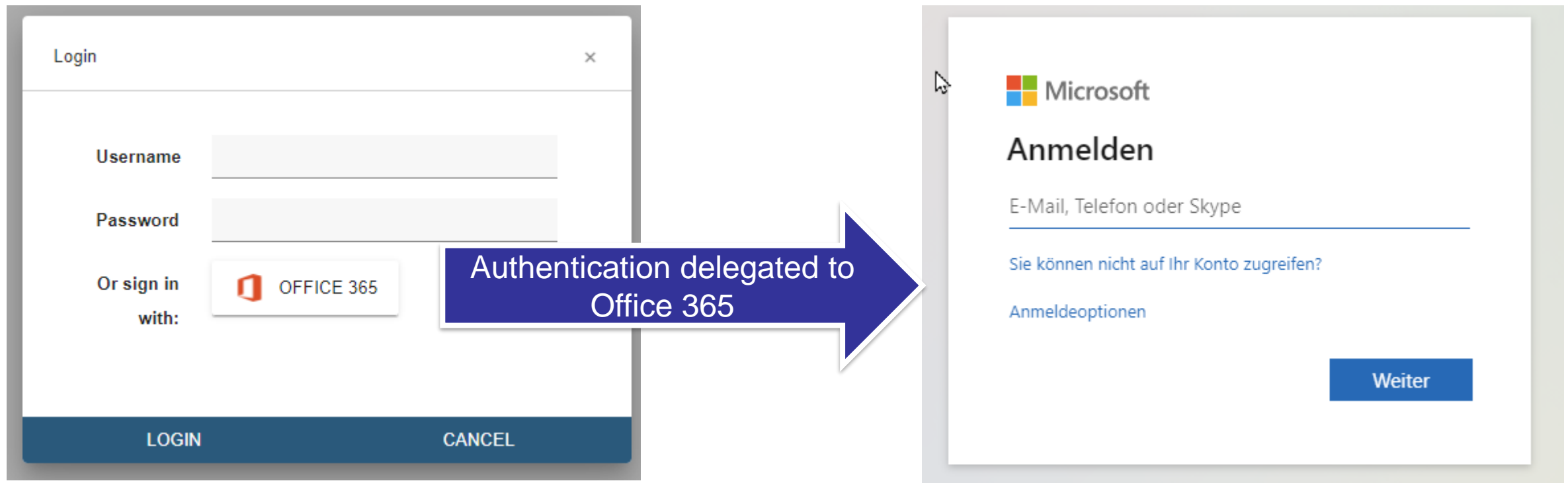
- Securable ist ein Framework zur Authentifizierung und Autorisierung von Nodeable
- <https://www.nodeable.io/securable>
- Bietet ABL API
 - Authentifizierung
 - Autorisierung
 - Pflege der Daten, z.B. User, Kennwörter und Zugriffsrechte
- Serverseitiges System, Deployment über mehrere Docker-Container



Securable

- Authentifizierung über
 - Username / Kennwort in securable
 - Social authentication providers
 - Facebook, Twitter, Google, LinkedIn, **Office 365**
 - Active Directory
 - Callback in OpenEdge Anwendung über PASOE oder OpenEdge Authentication Gateway
 - Multi-Factor Authentication (z.B. Code über SMS)
 - ...

Office 365 Login with Securable



Verschiedene Szenarien

- Desktop Anwendung nutzt **securable** zur Authentifizierung
- **Authentifizierung z.B. über Office 365**
- Securable bestätigt user-credentials
- Securable liefert Authentifizierungs-Token für die Web Anwendung
- Über das Authentifizierungs-Token erfolgt die sichere Anmeldung in der Web Anwendung, ohne dass der Benutzer sich noch einmal authentifizieren muss

- Für viele Anwender wird die Notwendigkeit sich über Office 365 zu authentifizieren positiv aufgenommen werden

Verschiedene Szenarien

- Desktop Anwendung nutzt **securable** zur Authentifizierung
- Authentifizierung z.B. **PASOE / OpenEdge Authentication Gateway mit dem bestehenden Anwendungslogin und Kennwort**
- Securable bestätigt user-credentials durch PASOE Call
- Securable liefert Authentifizierungs-Token für die Web Anwendung
- Über das Authentifizierungs-Token erfolgt die sichere Anmeldung in der Web Anwendung, ohne dass der Benutzer sich noch einmal authentifizieren muss

Agenda

- Modernisierung von OpenEdge Anwendungen
- CefSharp
- CefSharp vs. Microsoft WebView2 SDK
- Nutzen des Web Browser Controls in der Anwendung
- Hybride Anwendungsarchitektur
- Authentifizierung
- **Navigation innerhalb einer SPA (Angular)**
- Interaktion mit einer SPA (Angular)

Navigation innerhalb einer Angular SPA

- Moderne Web Anwendungen typisch als **single page web application**
- Navigation zu Anwendungsmasken über *Router* Komponente
- Navigation typischer Weise über JavaScript Funktionalität innerhalb der SPA
- Maskenlayout und Client-Code (JavaScript/TypeScript) entweder bereits im Browser geladen oder load on demand
- Adresse einer Maske kann auch über die URL angesteuert werden

- `http://localhost:3000/#/(view:user-maintenance)`

Hash Based Routing

- # Zeichen in der URL typischer Weise für Sprungmarke in (langen) Webseiten, z.B.:
https://de.wikipedia.org/wiki/Datenbank#Verschiedene_Formen_von_Datenbanksystemen
- Erlaubt Navigation innerhalb der Seite über URL – **ohne Reload**
- Router Komponenten von SPA Frameworks nutzen dieses Verhalten um Anwendungsmasken (Routen) über die Sprungmarke anzusprechen
- Sprungmarke hierbei nicht Bestandteil der HTML Seite

Hash Based Routing

- Navigation von

[http://localhost:3000/#/\(view:user-maintenance\)](http://localhost:3000/#/(view:user-maintenance))

zu

[http://localhost:3000/#/\(view:security-token-maintenance\)](http://localhost:3000/#/(view:security-token-maintenance))

- Navigation zu Maske/Route ohne Reload der SPA erlaubt flüssiges Wechseln zwischen Masken, Maskenstatus bleibt erhalten!
- Erlaubt z.B. Integration von Angular Web Masken in das Menü der Desktop Anwendung

Application Menu

Demo

- Rollbase Portal
- Rollbase Object Definition
- Rollbase Leads (Customer)
- Rollbase REST Logfile
- Customer Form (dynamic)
- Customer Form 2 (dynamic)
- Customer Form (static)
- Customers (Annotation Based)
- Angular Kundenmaske
- Consultingwerk Documentation
- CEF Token Maintenance**
- CEF User Maintenance

Welcome to the SmartComponent Library Demo! Click here to visit us on facebook!

Start x SmartFramework Maintenance [Security Token Maintenance] x

SmartFramework Reference Application

Mike Fechn



Token Code	Token Description
CanLaunchForms	Allows the user to launch forms directly from the ...
ServiceInterface.Create	Used by the SmartRequestAuthorizationProvider...
ServiceInterface.Delete	Used by the SmartRequestAuthorizationProv...
ServiceInterface.FetchData	Used by the SmartRequestAuthorizationProvider.
ServiceInterface.Modify	Used by the SmartRequestAuthorizationProvider...
ServiceInterface.SaveChanges	Used by the SmartRequestAuthorizationProvider.
StoreUserProfileForGroups	Controls if the user is authorized to store user pr...

ADD COPY SAVE CANCEL DELETE

Token Code *
ServiceInterface.Delete

Token Description
Used by the SmartRequestAuthorizationProvider when the Configuration
"UseDetailedRequestAuthorizationOnUpdate" is set to TRUE.

Demo

- Demo – Navigation zwischen Angular Forms aus dem Desktop Menü

Agenda

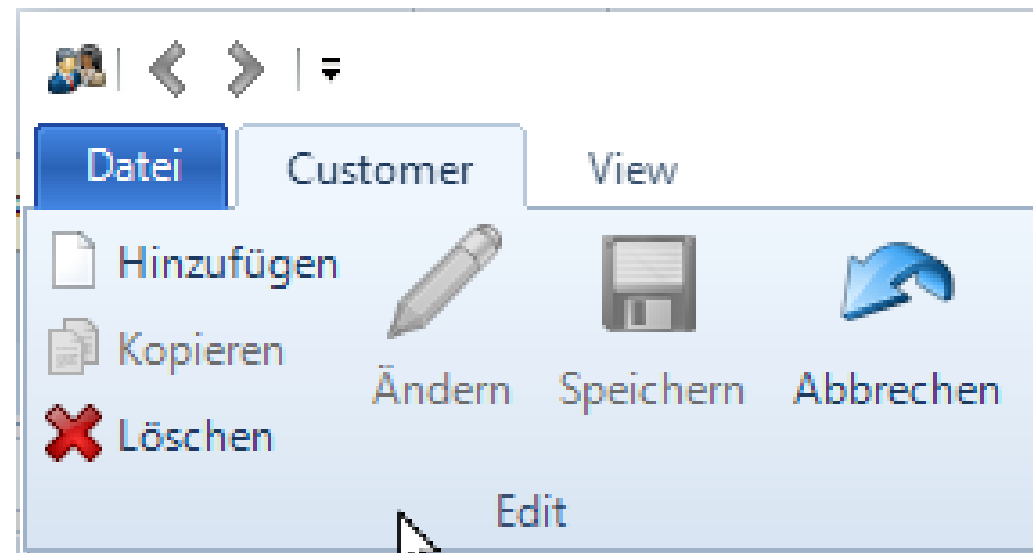
- Modernisierung von OpenEdge Anwendungen
- CefSharp
- CefSharp vs. Microsoft WebView2 SDK
- Nutzen des Web Browser Controls in der Anwendung
- Hybride Anwendungsarchitektur
- Authentifizierung
- Navigation innerhalb einer SPA (Angular)
- **Interaktion mit einer SPA (Angular)**

Interaktion mit der SPA (Angular)

- CefSharp unterstützt bidirektionale Kommunikation zwischen der Desktop Anwendung und der SPA
 - Aufrufen einer JavaScript Function vom Desktop aus
 - Auslösen eines Events durch die Browser Anwendung und Behandlung durch die Desktop Anwendung
- Interaktion direkt über das Browser-Control, kein Messaging über z.B. Web Sockets oder JMS erforderlich

Interaktion mit der SPA (Angular)

- Use-Case hier: Ersetzen von Buttons in der eingebetteten Web Anwendung durch Buttons im Ribbon der Desktop Anwendung
- Button Click Event und Enablen/Disablen der Buttons in Abhängigkeit vom Status der Maske



Aufruf von JavaScript Funktion über CefSharp

- Basiert auf der EvaluateScriptAsync Funktion des Frame Objektes von CefSharp (Webseiten können aus Framesets bestehen)

- z.B.

```
EvaluateScriptAsync('cefsharpCall("ButtonClick",  
                        {  
                            "toolbarInstanceName": "CustomerToolbar",  
                            "buttonName": "SaveChanges"  
                        });')
```

- **cefsharpCall** ist eine globale JavaScript Funktion welche dann den **ButtonClick** Handler des betreffenden Buttons auslöst
- http://cefsharp.github.io/api/55.0.0/html/M_CefSharp_IFrame_EvaluateScriptAsync.htm

IFrame.EvaluateScriptAsync Method

Version 55.0.0

Execute some Javascript code in the context of this WebBrowser, and return the result of the

Namespace: [CefSharp](#)

Assembly: CefSharp (in CefSharp.dll) Version: 55.0.0.0 (55.0.0.0)

▲ Syntax

```
C# C++
Task<JavascriptResponse> EvaluateScriptAsync(
    string script,
    string scriptUrl = "about:blank",
    int startLine = 1,
    Nullable<TimeSpan> timeout = null
)
```

Parameters

script

Type: [System.String](#)

The Javascript code that should be executed.

scriptUrl (Optional)

Type: [System.String](#)

is the URL where the script in question can be found, if any.

startLine (Optional)

Type: [System.Int32](#)

is the base line number to use for error reporting.

timeout (Optional)

Type: [System.Nullable<TimeSpan>](#)

The timeout after which the Javascript code execution should be aborted.

Async .NET Methode aus der ABL ...

```
METHOD PUBLIC JavascriptResponse ExecuteScript (pcScript AS CHARACTER,  
                                                pcUrl AS CHARACTER,  
                                                piLine AS INTEGER,  
                                                pcFrame AS CHARACTER):  
  
    DEFINE VARIABLE oTask AS System.Threading.Tasks.Task NO-UNDO.  
  
    DO WHILE oBrowser:IsLoading:  
        PROCESS EVENTS.  
    END.  
  
    oTask = oBrowser:GetBrowser():GetFrame(pcFrame):EvaluateScriptAsync (pcScript,  
                                                                           pcUrl,  
                                                                           piLine,  
                                                                           ?,  
                                                                           FALSE).  
  
    oTask:GetAwaiter():GetResult().  
  
    RETURN CAST(oTask, "System.Threading.Tasks.Task<JavascriptResponse>"):Result.  
  
END METHOD.
```

JavaScript Event durch CefSharp behandeln

ChromiumWebBrowser.JavascriptMessageReceived Event

Version 75.1.140

Event handler that will get called when the message that originates from CefSharp.PostMessage

Namespace: CefSharp.Wpf

Assembly: CefSharp.Wpf (in CefSharp.Wpf.dll) Version: 75.1.140.0 (75.1.140.0)

▲ Syntax

```
C# C++  
public event EventHandler<JavascriptMessageReceivedEventArgs> JavascriptMessageReceived
```

Value

Type: System.EventHandler<JavascriptMessageReceivedEventArgs>

JavaScriptMessageReceived
EventArgs Klasse liefert
„Value“ als System.Object,
z.B. als System.String
(ABL CHARACTER)

- http://cefsharp.github.io/api/75.1.x/html/E_CefSharp_Wpf_ChromiumWebBrowser_JavascriptMessageReceived.htm

JavaScript Event durch PostMessage auslösen

```
public sendCefsharpEvent(event: string, eventArgs?: any) {  
    if (!this.isCefsharpAvailable) {  
        warnInDevMode(  
            `CefSharp event will not be posted - CefSharp is not available.`  
        );  
        warnInDevMode(`Event: ${JSON.stringify(event)}`);  
        return;  
    }  
    console.log('sending event ', event);  
    const ev = {  
        eventType: event,  
        eventArgs,  
    };  
    this.CefSharp.PostMessage(JSON.stringify(ev));  
}
```

This.CefSharp wird durch CefSharp in den JavaScript Window-Namespace injiziert (injected)

Demo

- Demo – Steuerung der Funktionalität einer Angular Maske über Desktop Buttons

Fragen



