

The Edge Browser Control – The Most Powerful Access Control You Are Probably Not Using



Microsoft Visual Basic for Applications - Slider_0.0.10 - [clsWebBridge (Code)]

Datei Bearbeiten Ansicht Einfügen Debuggen Ausführen Extras MZ-Tools Add-Ins Fenster ?

Ansicht

Kopiere nach Einfügen aus Optionen

Projekt - Database

(Allgemein) HandleTimer

Database (Slider_0.0.10)

- Microsoft Access Klassenobjekt
 - Form_frmSlider
- Module
 - modSliderHtml
 - modWebBridge
- Klassenmodule
 - clsSlider
 - clsWebBridge

Code Explorer

Name

- Attach
- BrowserReady
- Class_Initialize
- Class_Terminate
- CommandReceived
- ControlName (Get)
- C_MODULE

Eigenschaften - clsWebBridge

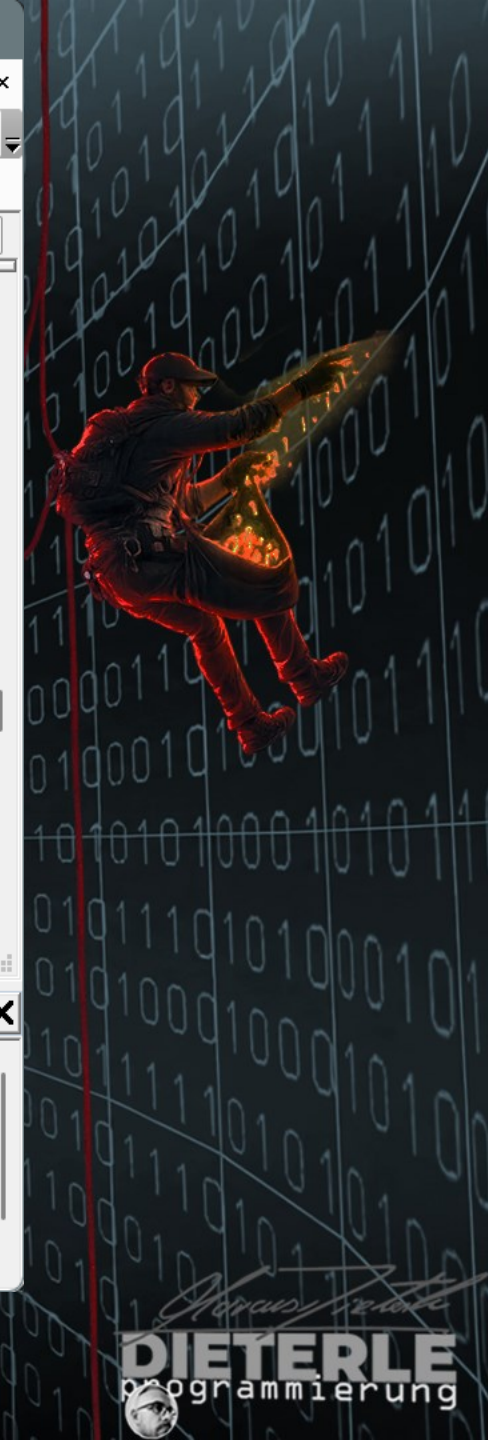
clsWebBridge ClassModule

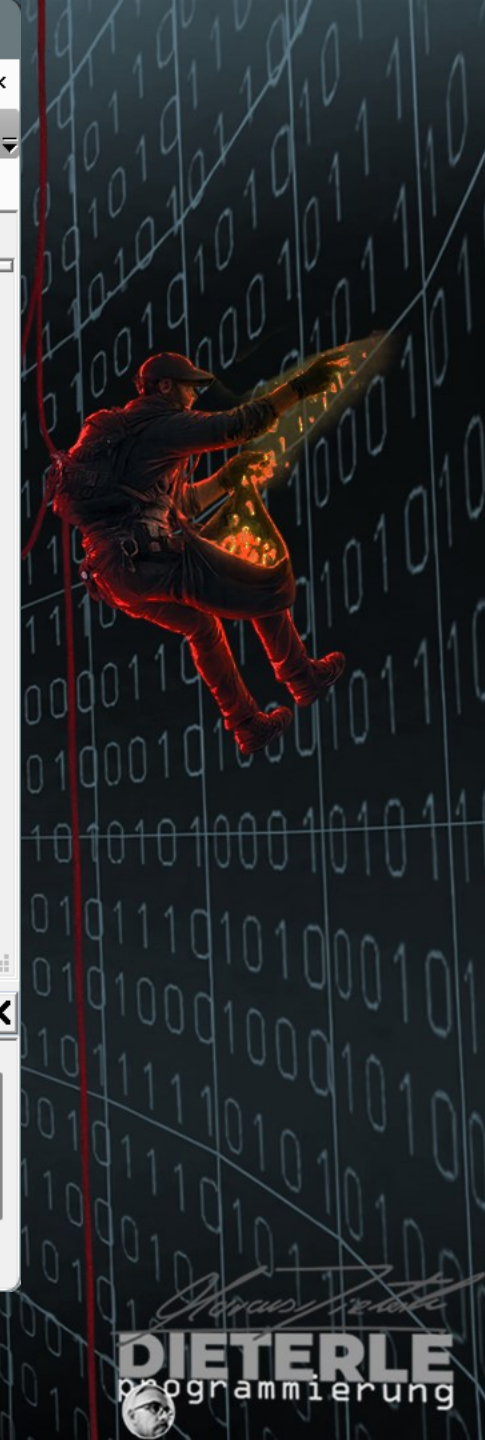
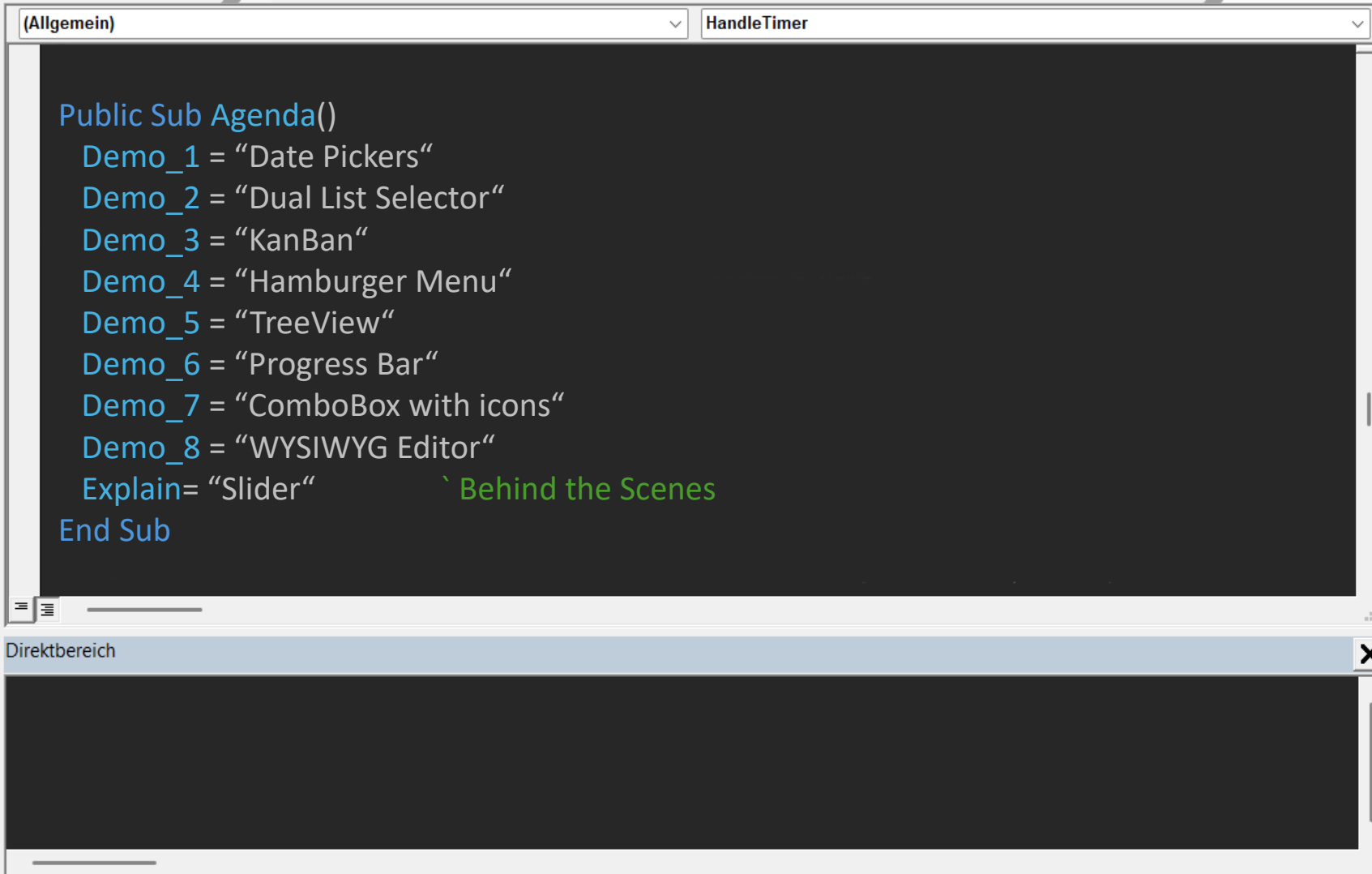
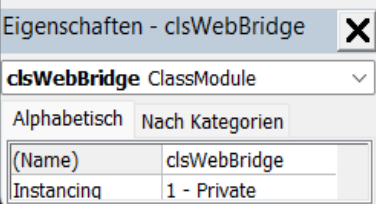
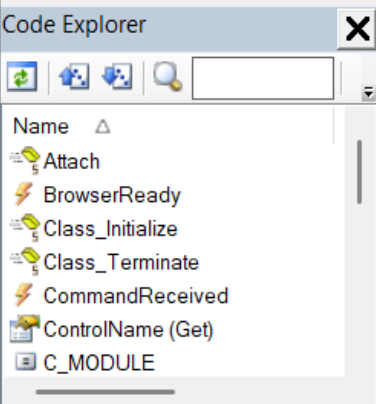
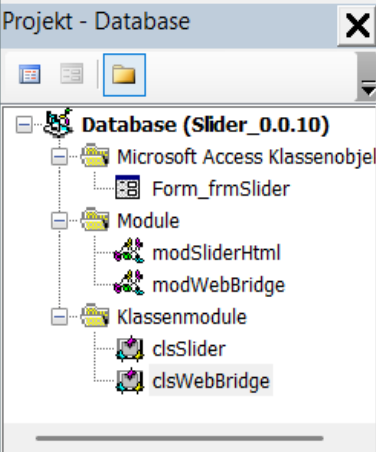
Alphabetisch Nach Kategorien

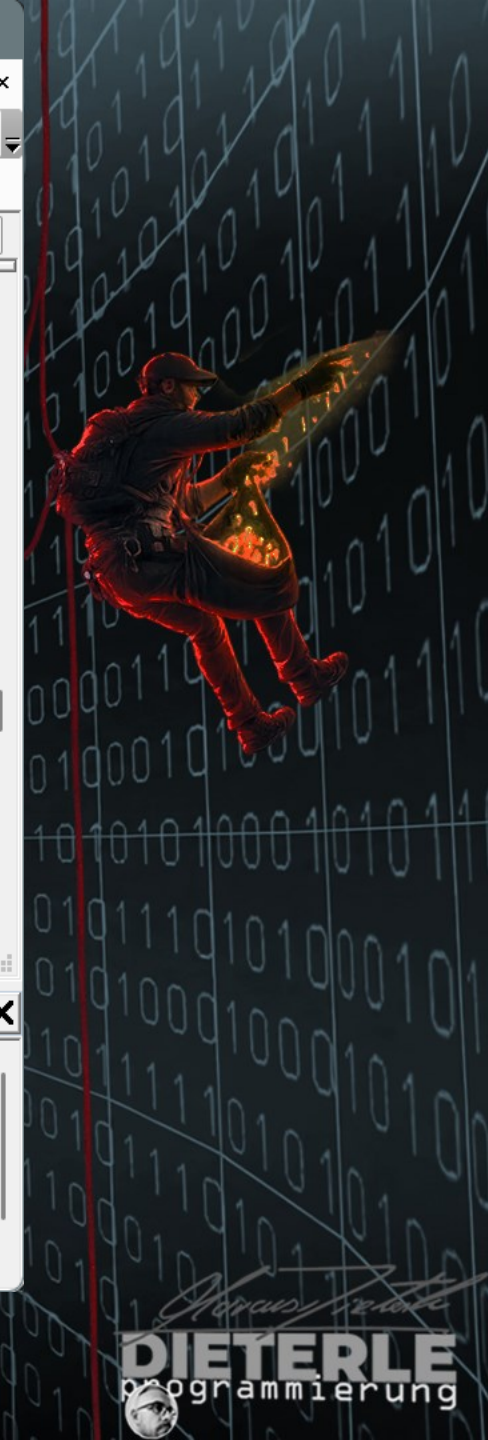
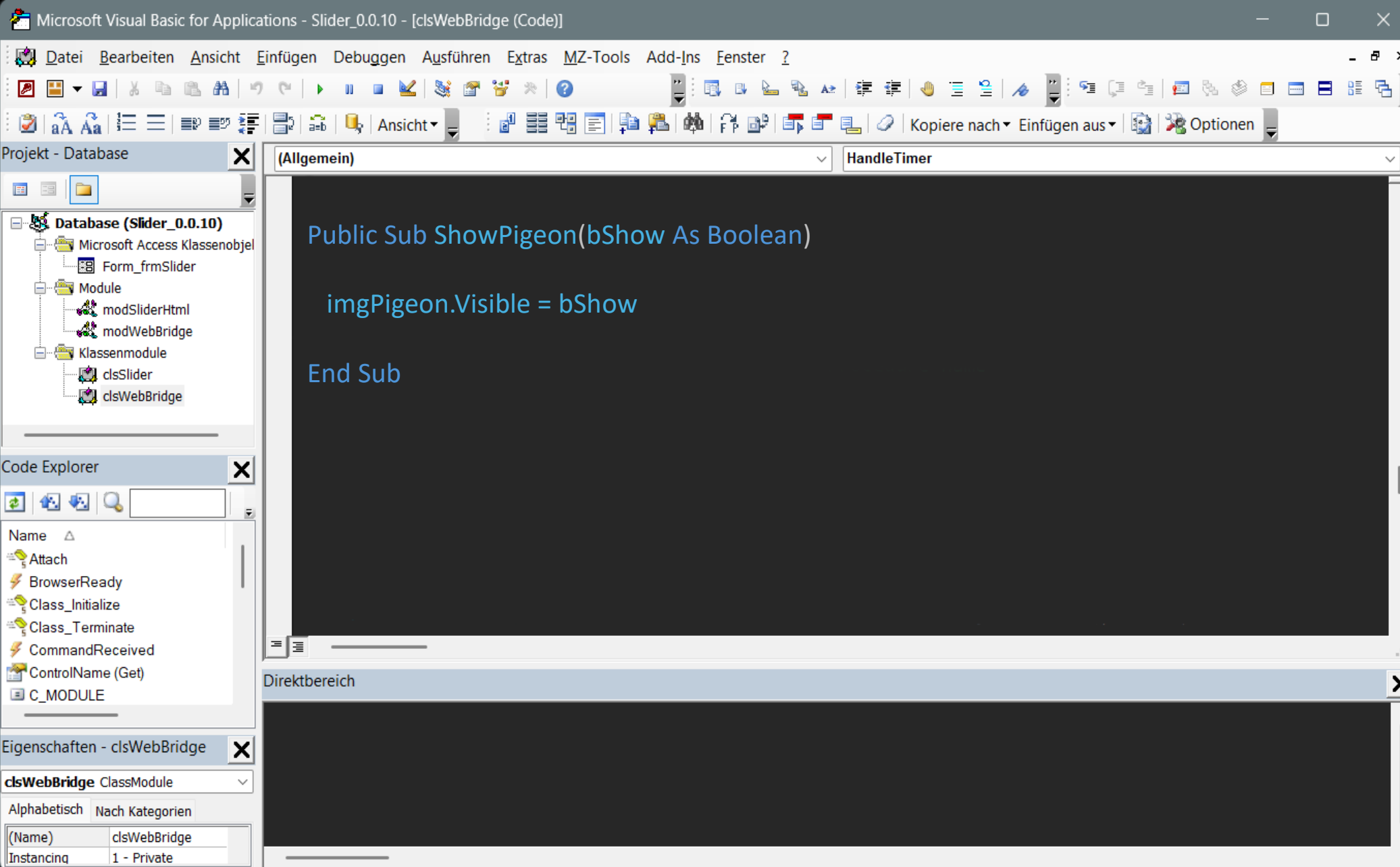
(Name)	clsWebBridge
Instanzierung	1 - Private

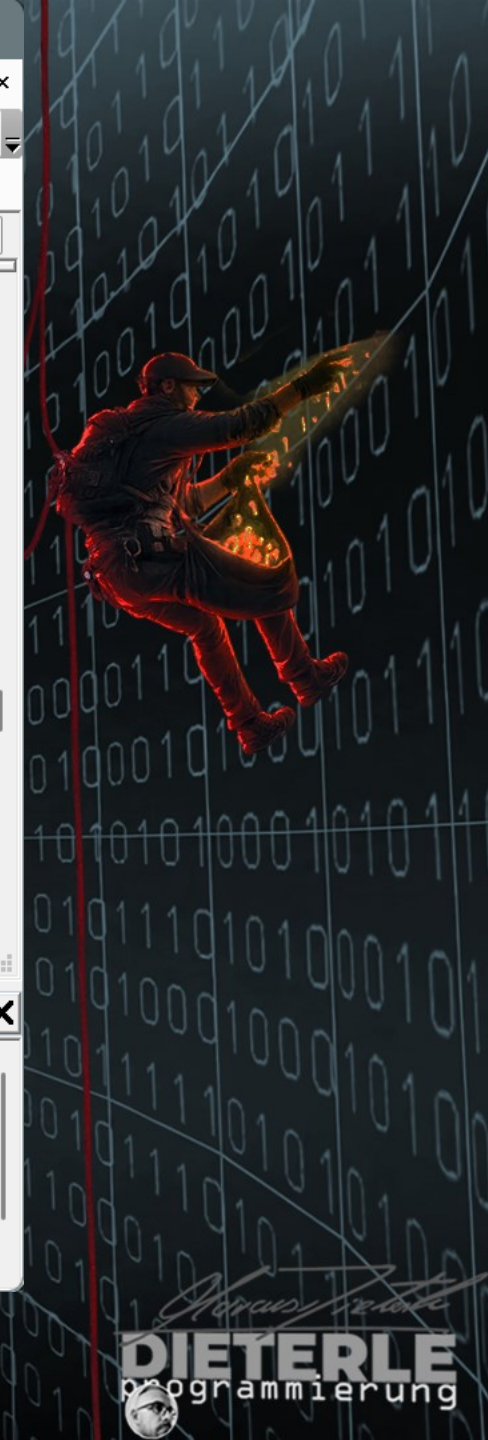
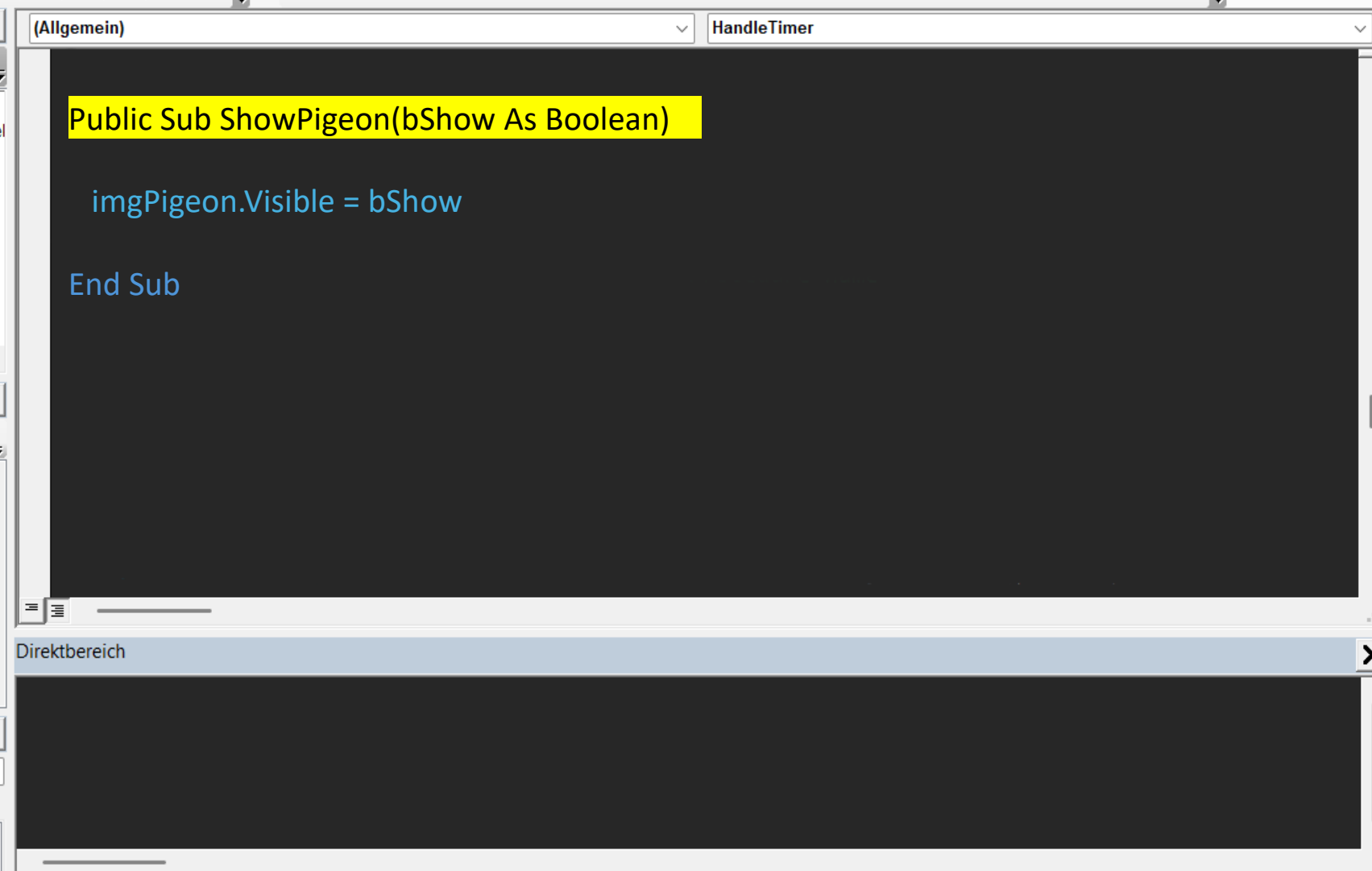
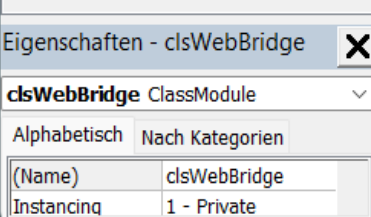
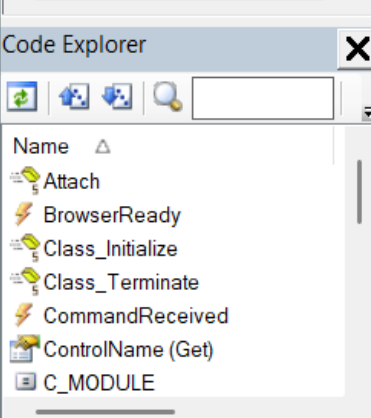
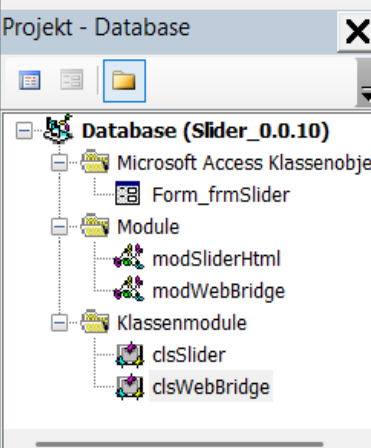
```
Public Sub Personal_Info()  
  
    Name = "Marcus Dieterle"  
    Age = "52"  
    Nation = "German"  
    Website = "www.dieterle-programmierung.de"  
    Email = "marcus@dieterle-programmierung.de"  
    LinkedIn = "www.linkedin.com/in/marcus-dieterle-557572222"  
    YouTube = "https://www.youtube.com/@DIETERLEprogrammierung"  
  
End Sub
```

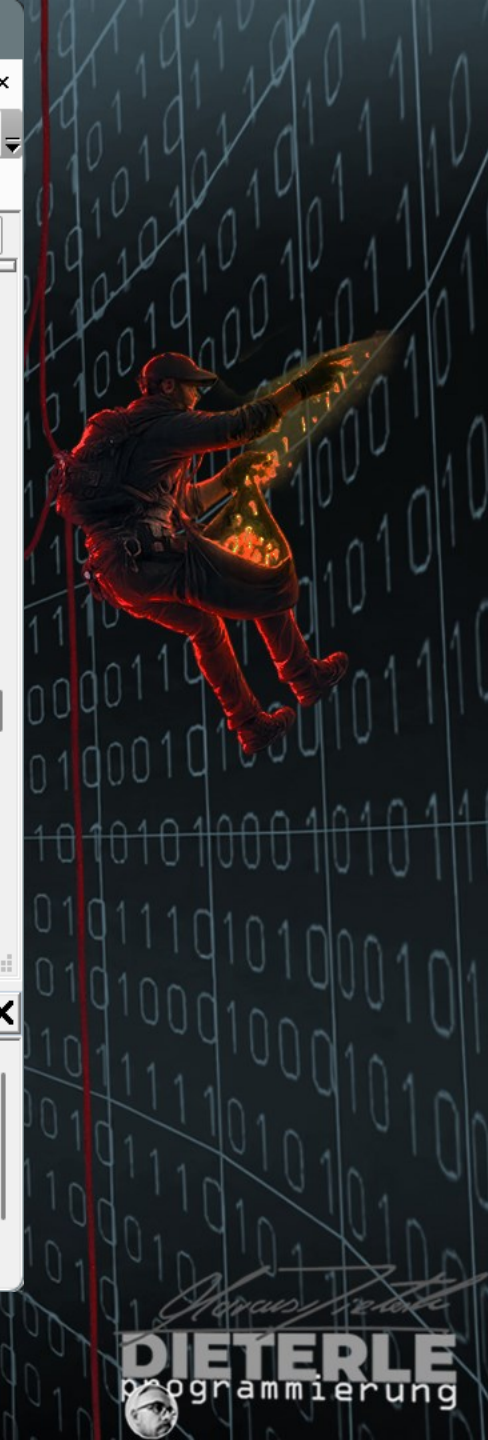
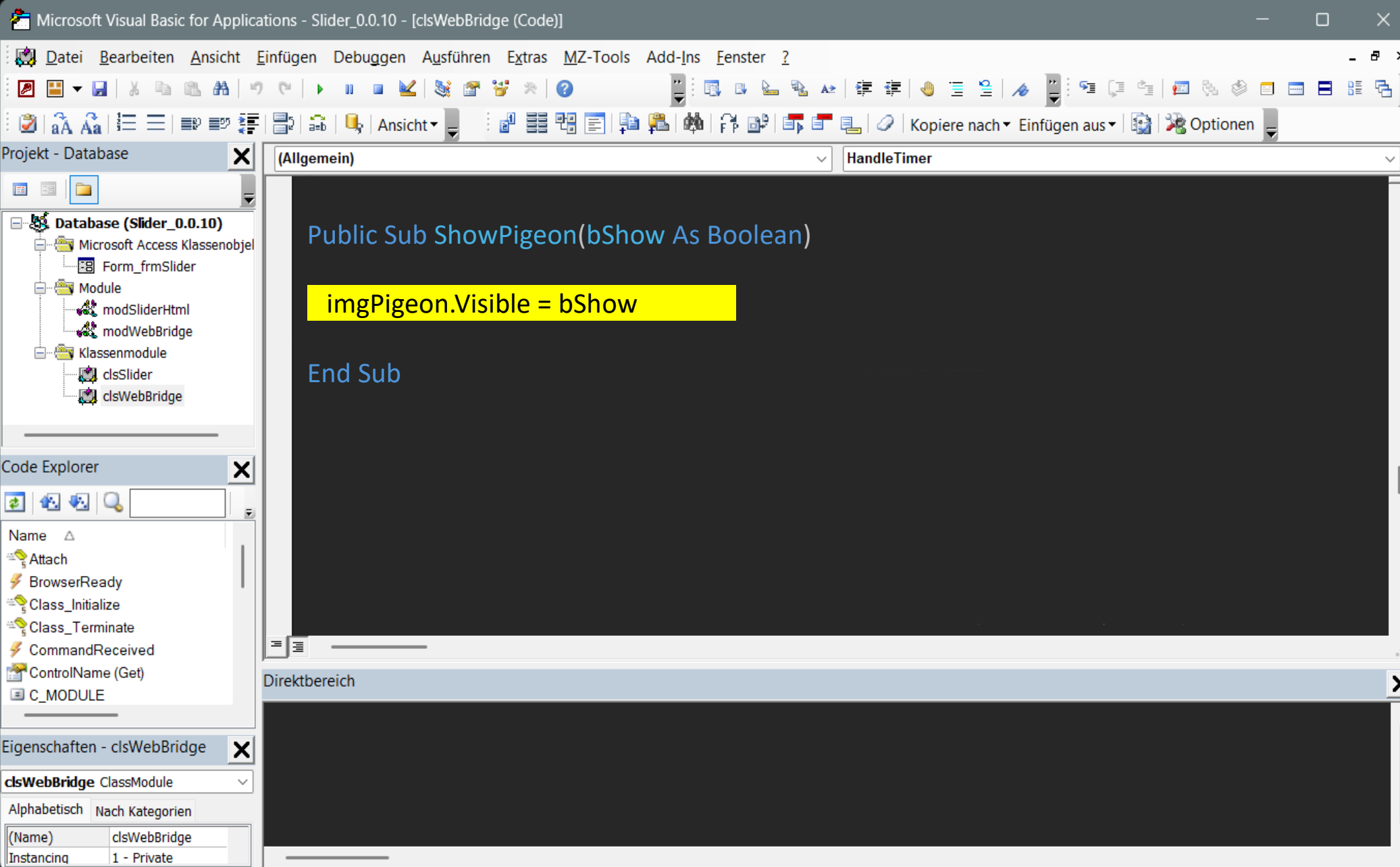
Direktbereich





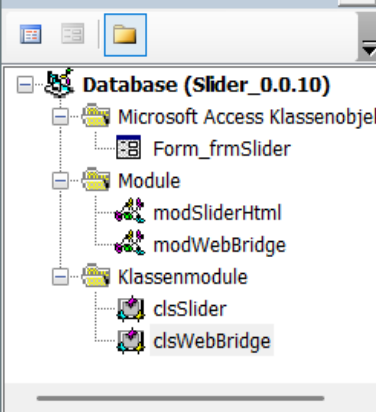




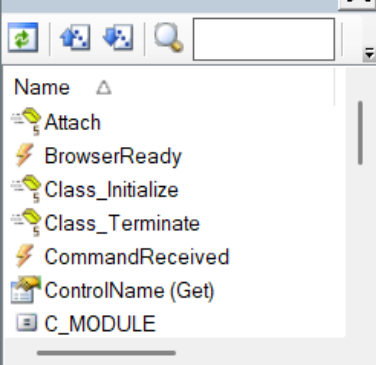




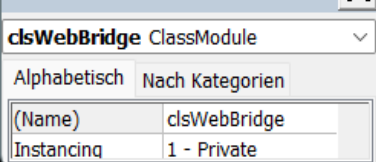
Projekt - Database



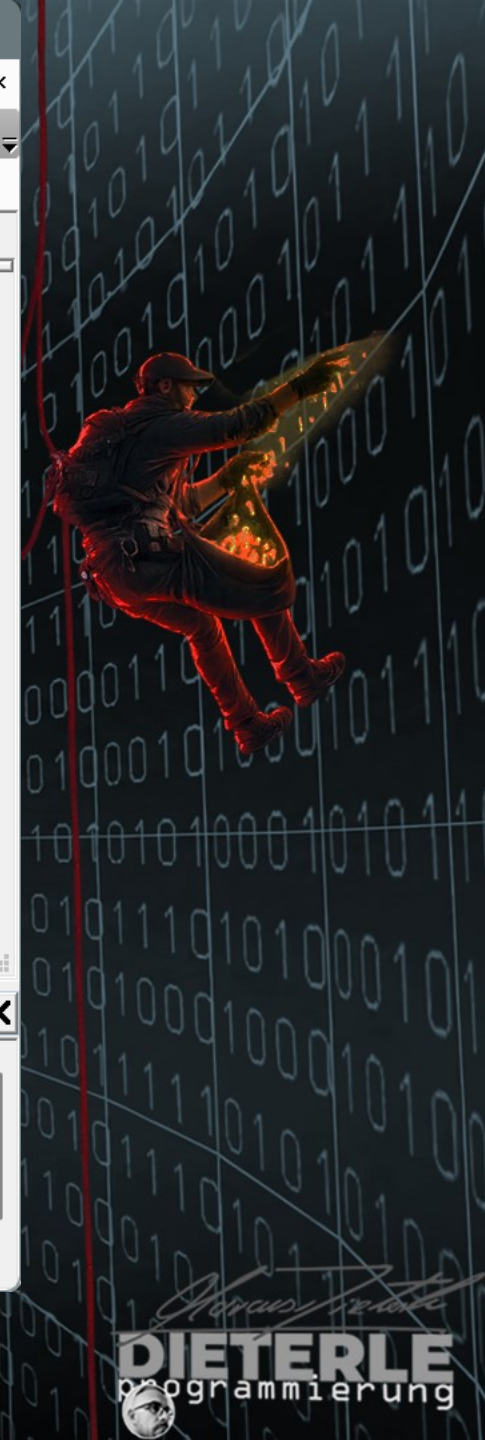
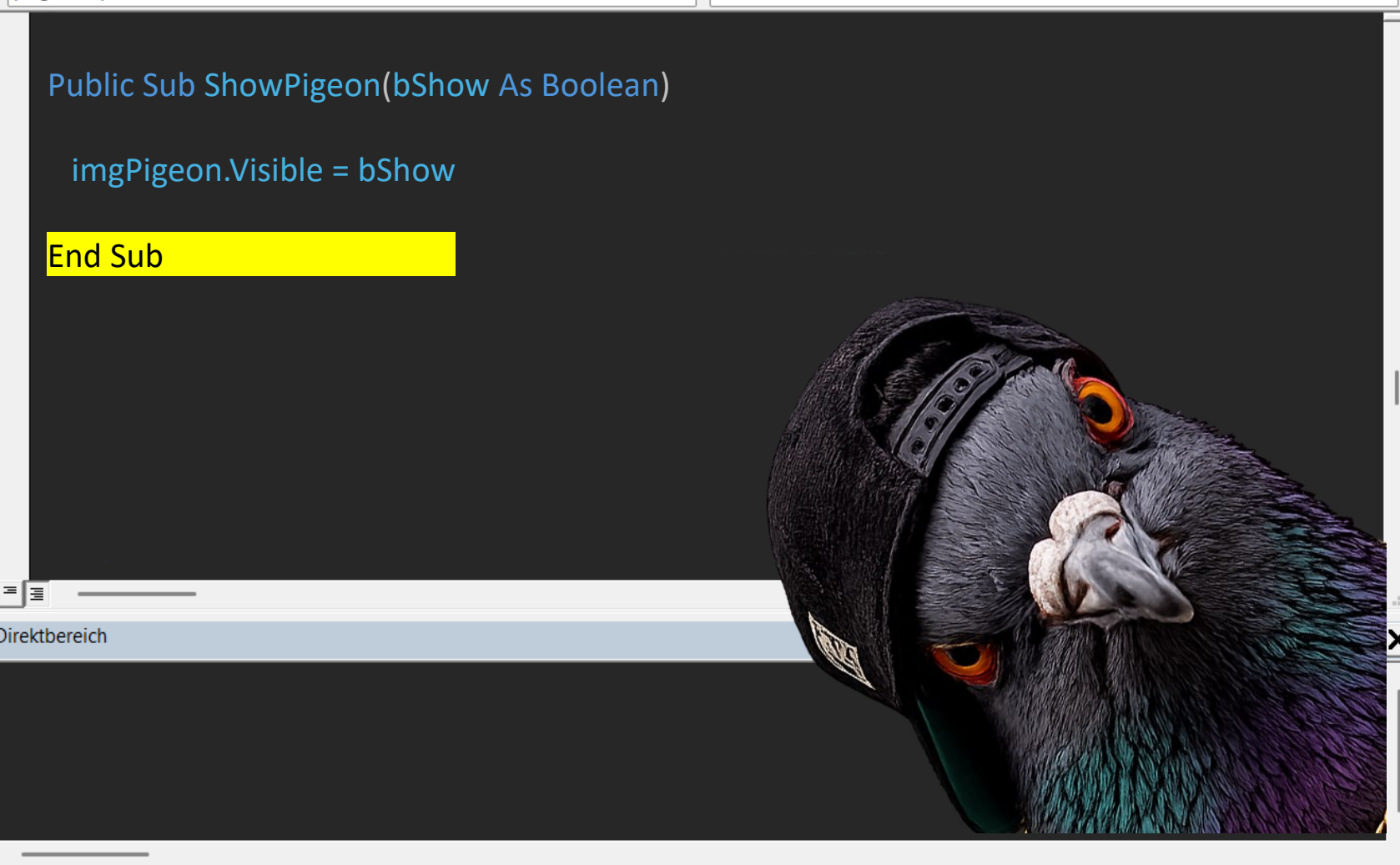
Code Explorer

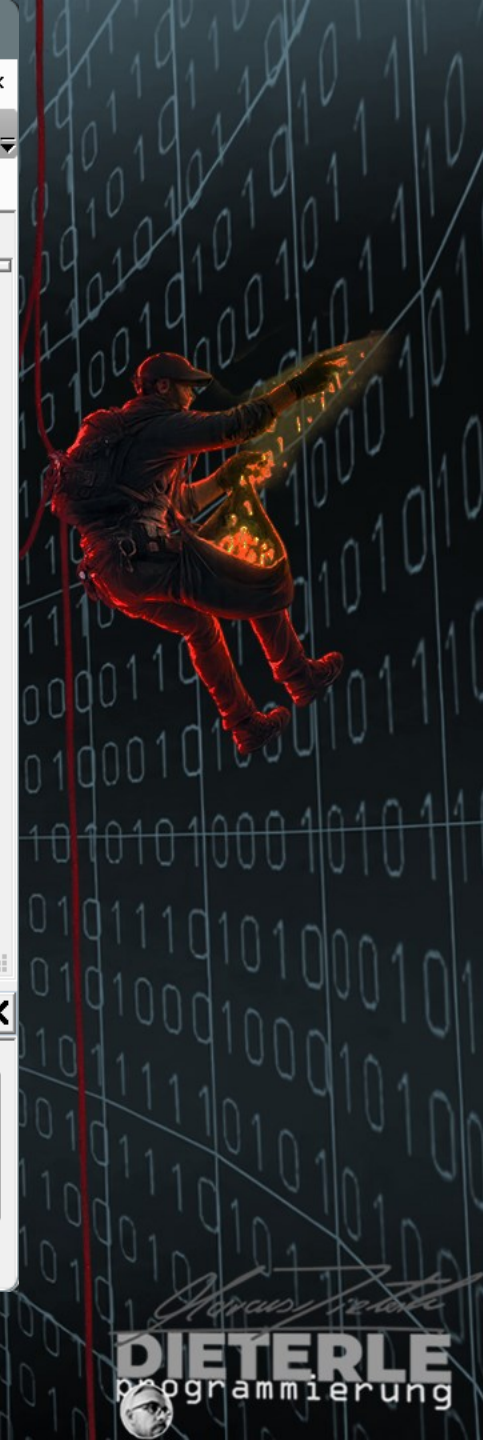
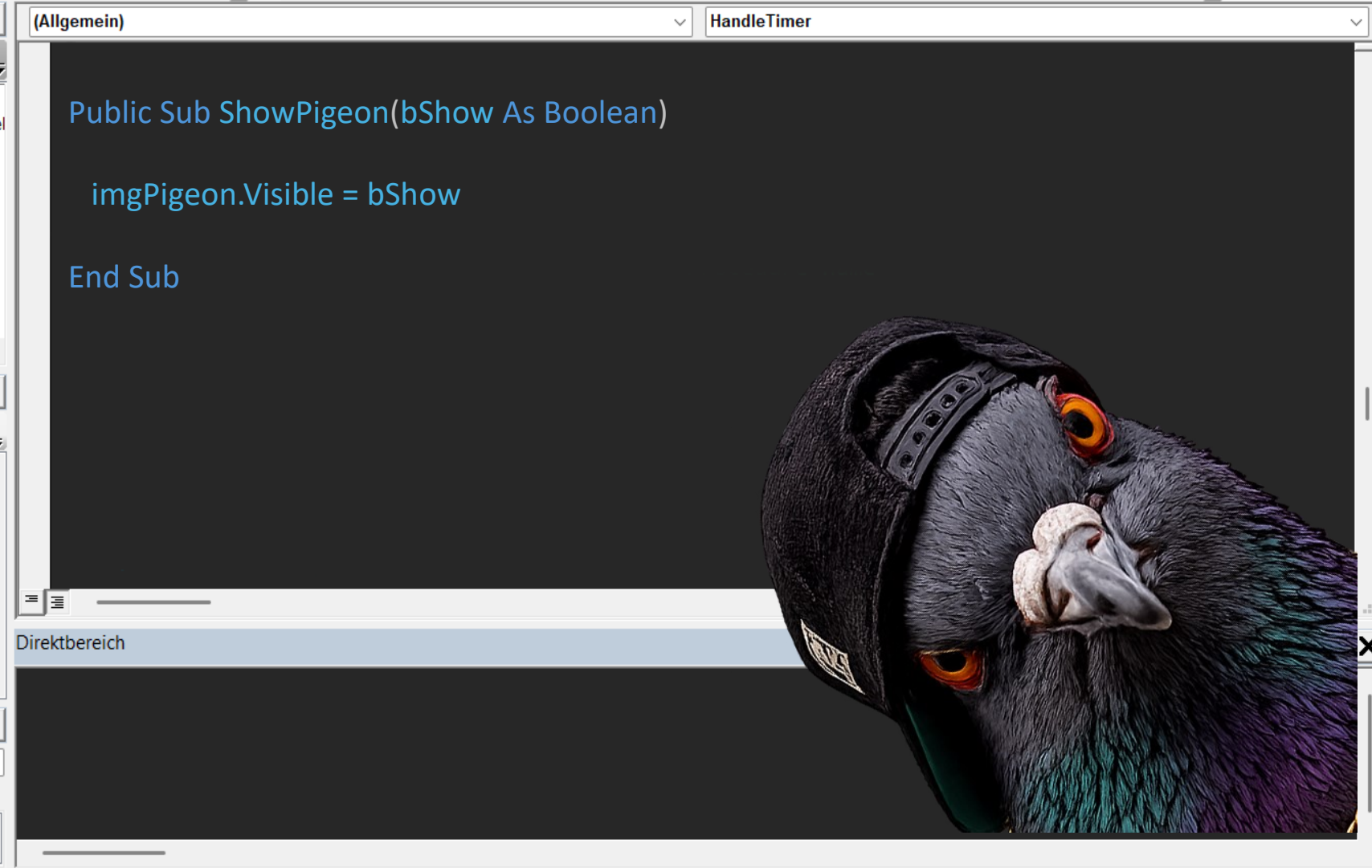
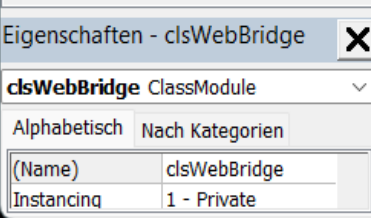
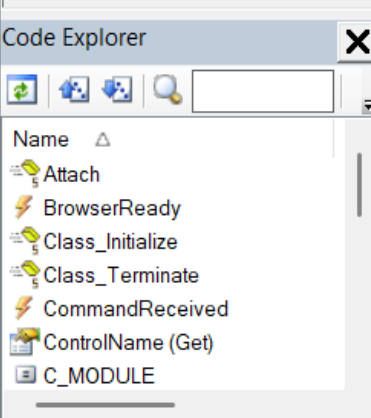
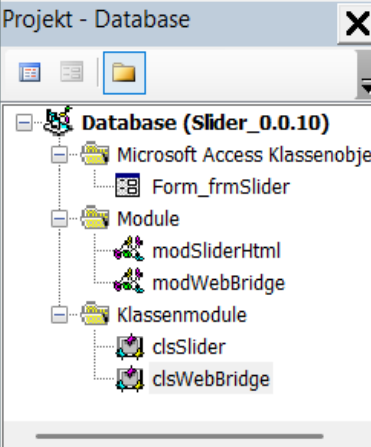


Eigenschaften - clsWebBridge



(Allgemein) HandleTimer





Why I "Misuse" the acEdgeBrowser as a Control

Microsoft Access lacks many of the modern UI components that users expect from modern desktop applications and web interfaces.

To overcome these limitations, I use the **acEdgeBrowser** as a foundation for building custom controls. By combining HTML, CSS, and JavaScript with Access.

I can:

- 👉 Provide data from Access to the control
- 👉 Display and visualize data in a modern UI
- 👉 Allow users to edit, select, or interact with the data
- 👉 Read the results back into Access for further processing

This approach enables rich, modern user experiences while staying within the Access environment.



System Requirements

The custom controls are built on top of the acEdgeBrowser control, which is available in:

- 👉 Microsoft Access for **Microsoft 365** (introduced around 2021)
- 👉 Microsoft Office / **Access 2024** and later

The acEdgeBrowser control hosts the Microsoft Edge WebView2 engine, enabling modern web technologies (HTML, CSS, and JavaScript) to run directly inside Access forms.

The Microsoft Edge **WebView2** Runtime is required:

- 👉 Win 11: Installed by default
- 👉 Win 10: Automatically installed and maintained with current versions of Microsoft Edge



Important Limitations

Although the acEdgeBrowser control is based on Microsoft Edge WebView2, Microsoft only exposes a limited subset of the WebView2 functionality through the Access object model. As a result, some capabilities available in native WebView2 applications are not accessible from Microsoft Access.

Impact for Developers:

When developing custom controls for Access, it is important to remember that you are working with an embedded WebView2 browser that exposes only a fraction of the capabilities available in a native WebView2 application. Design decisions and integration strategies should take these limitations into account.

Since features and events like WebMessageReceived + postMessage are missing, I built the webBridge with a 'mailbox function' to bridge the gap.



Hey Microsoft, are there any updates on the horizon?

DEMO TIME

Any Questions so far?



DEMO TIME



Date Pickers



Dual List Selector



Kanban To-Do List



Hamburger Menu



TreeView



Progress Bar



ComboBox with icons



WYSIWYG Editor



CHALK TALK TIME

Slider – Behind the Scenes

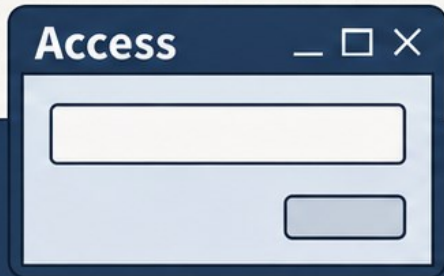


1a

The Challenge and the Solution

A Simple Illustration of the WebBridge Concept

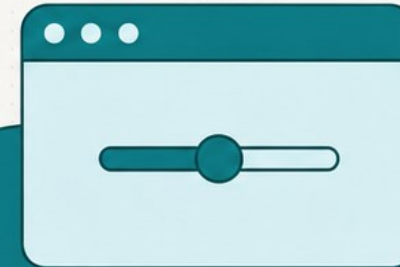
Access



**PROCESS
BOUNDARY**

No shared memory. No shared
variables. No direct calls.

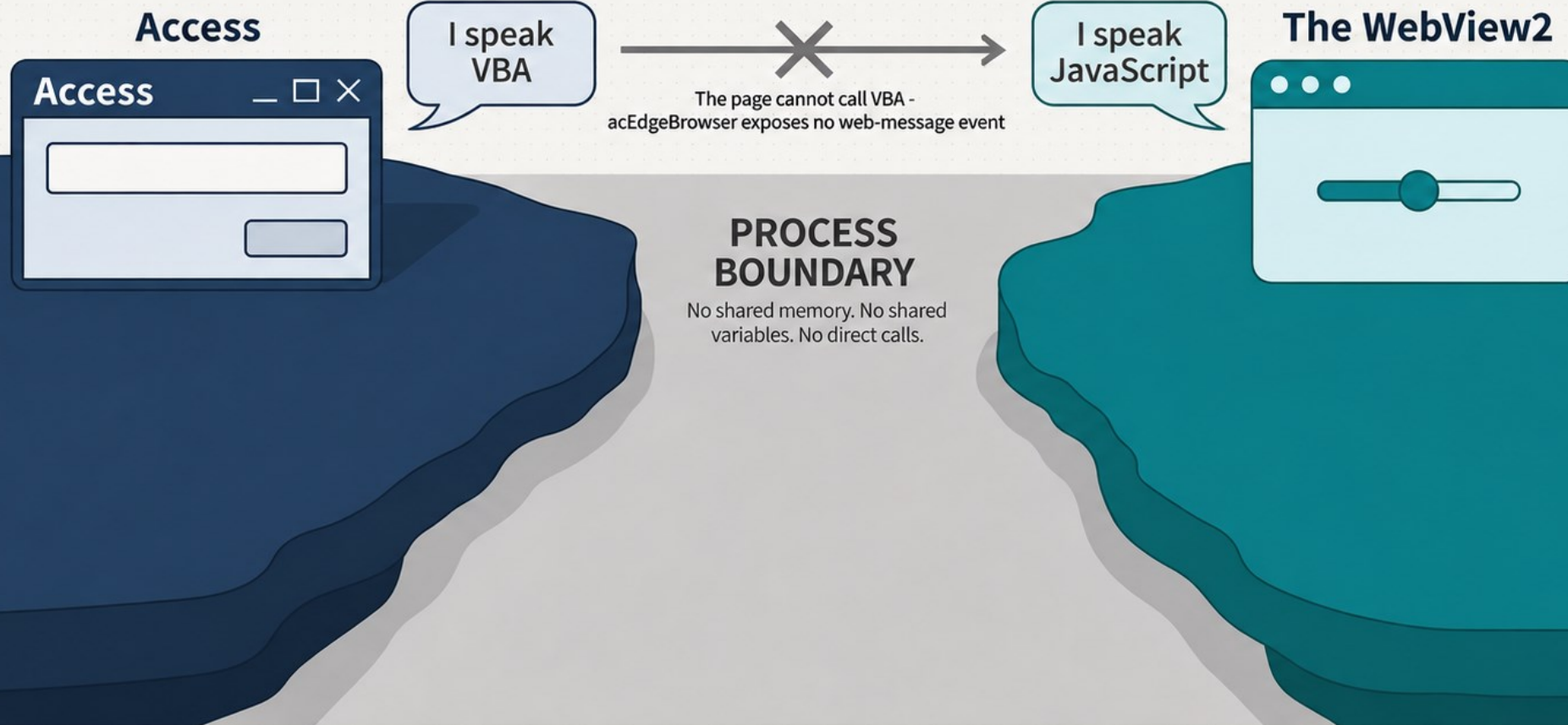
The WebView2



1b

The Challenge and the Solution

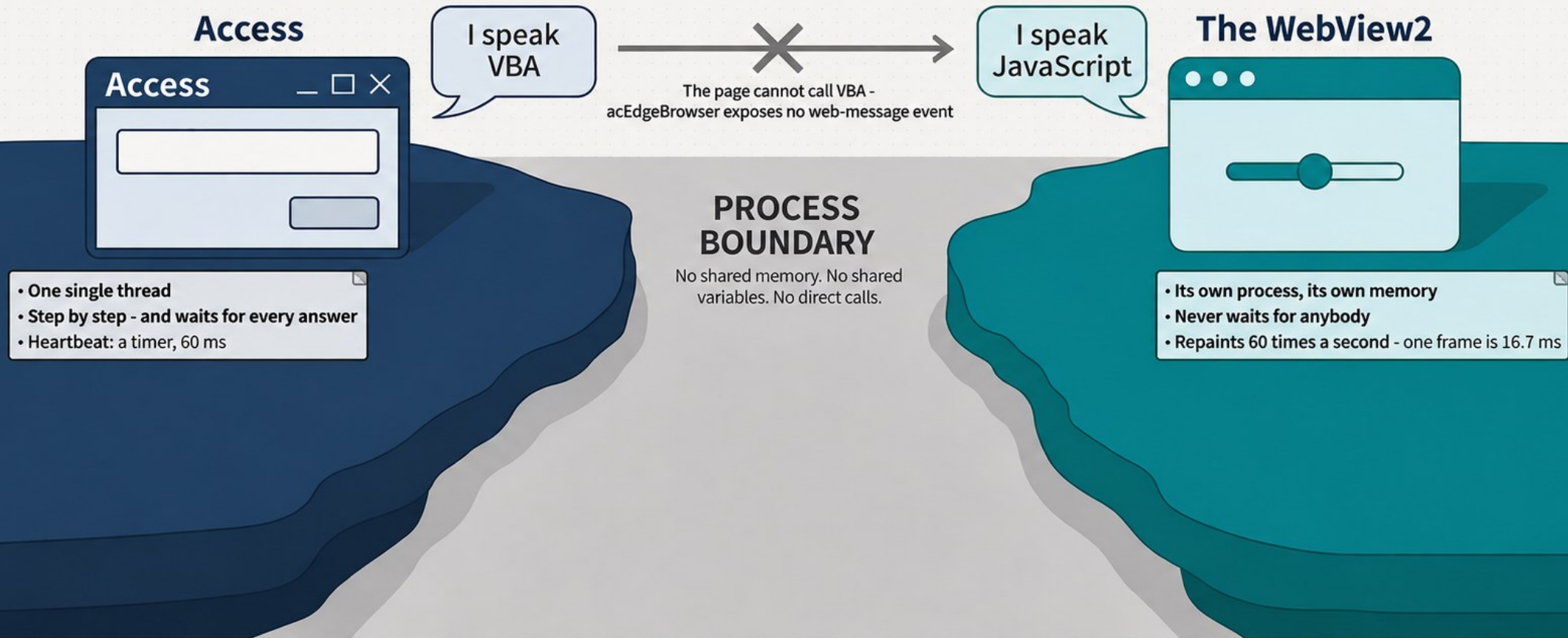
A Simple Illustration of the WebBridge Concept



1c

The Challenge and the Solution

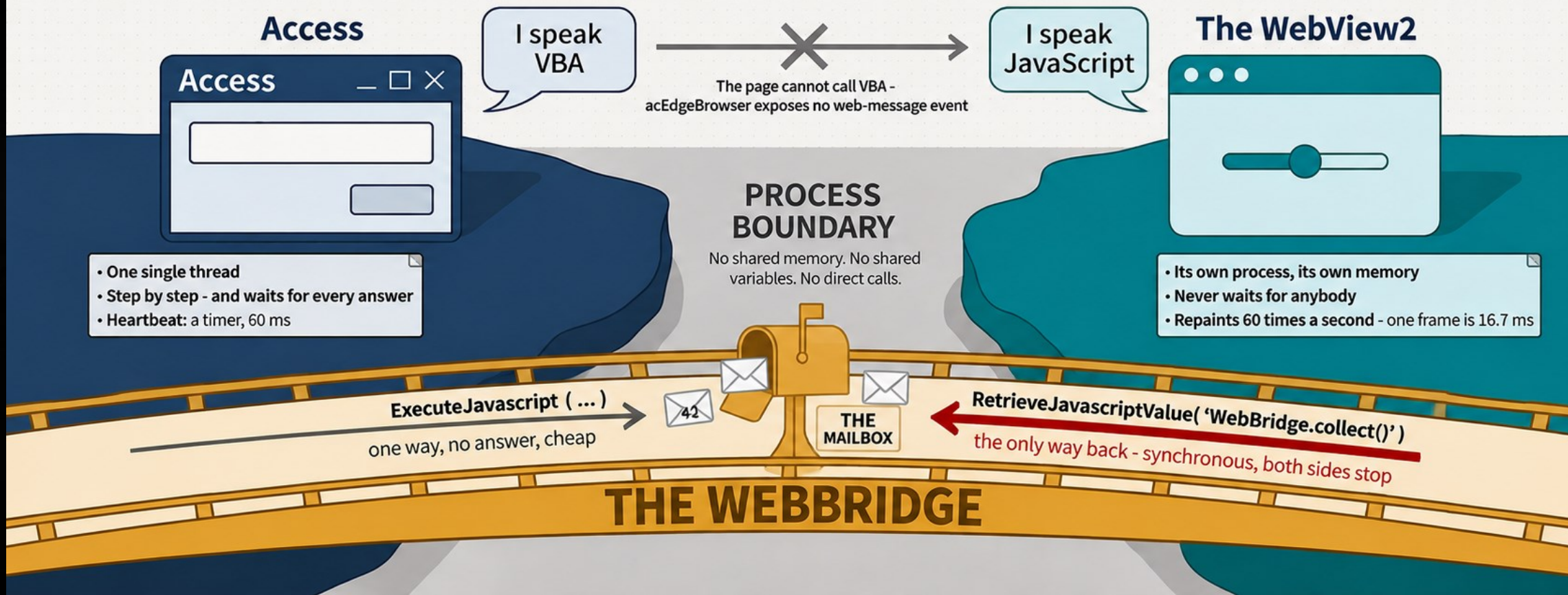
A Simple Illustration of the WebBridge Concept



1d

The Challenge and the Solution

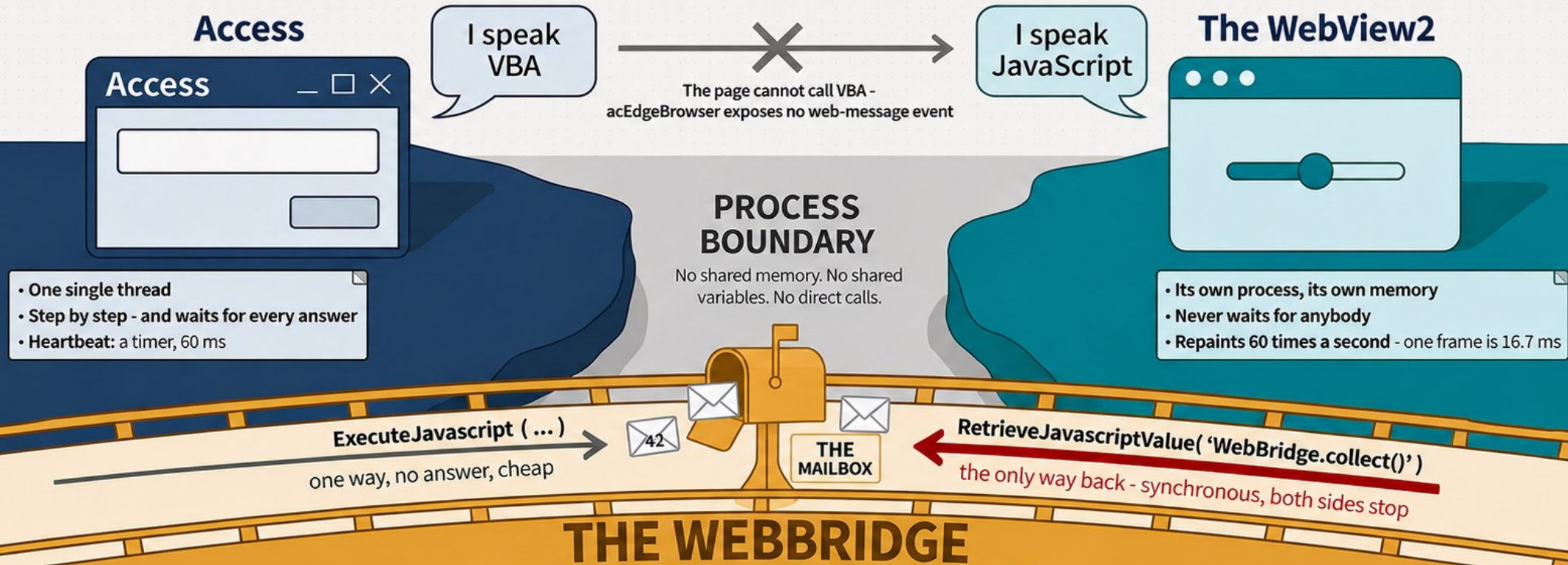
A Simple Illustration of the WebBridge Concept



1e

The Challenge and the Solution

A Simple Illustration of the WebBridge Concept



HOW THE MAILBOX WORKS



```
WebBridge.send('change', { value: 42 })
```

The page posts a letter - and REPLACES the letter of the same kind that is already in



So the box always holds exactly one letter, and it is always the newest value



```
WebBridge.collect()
```

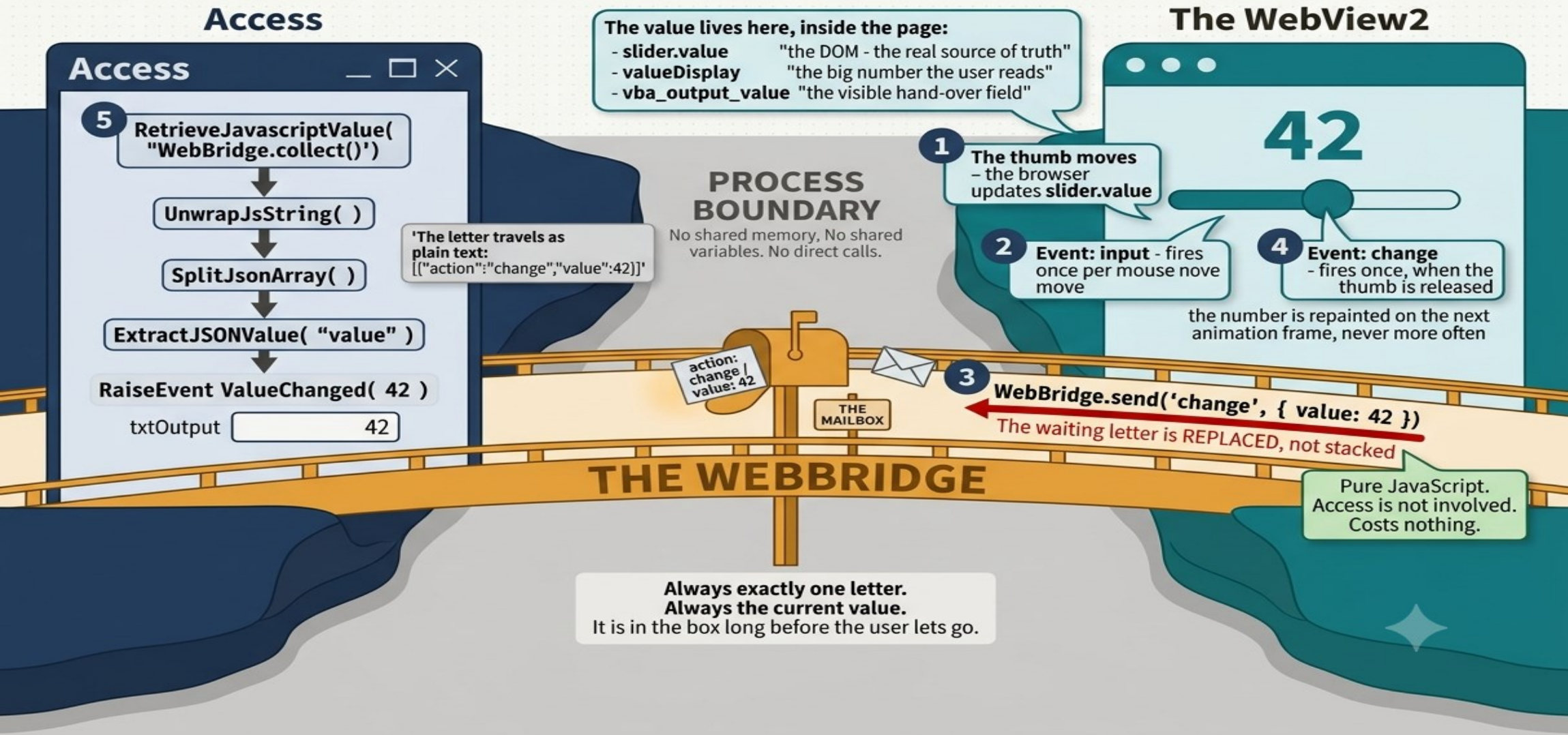
Access opens the box once and takes everything out in a single move

The page never knocks on the door. Access decides when to check the box.

2

2 - Slider Action: where the value lives

From the moving thumb to the text box in Access

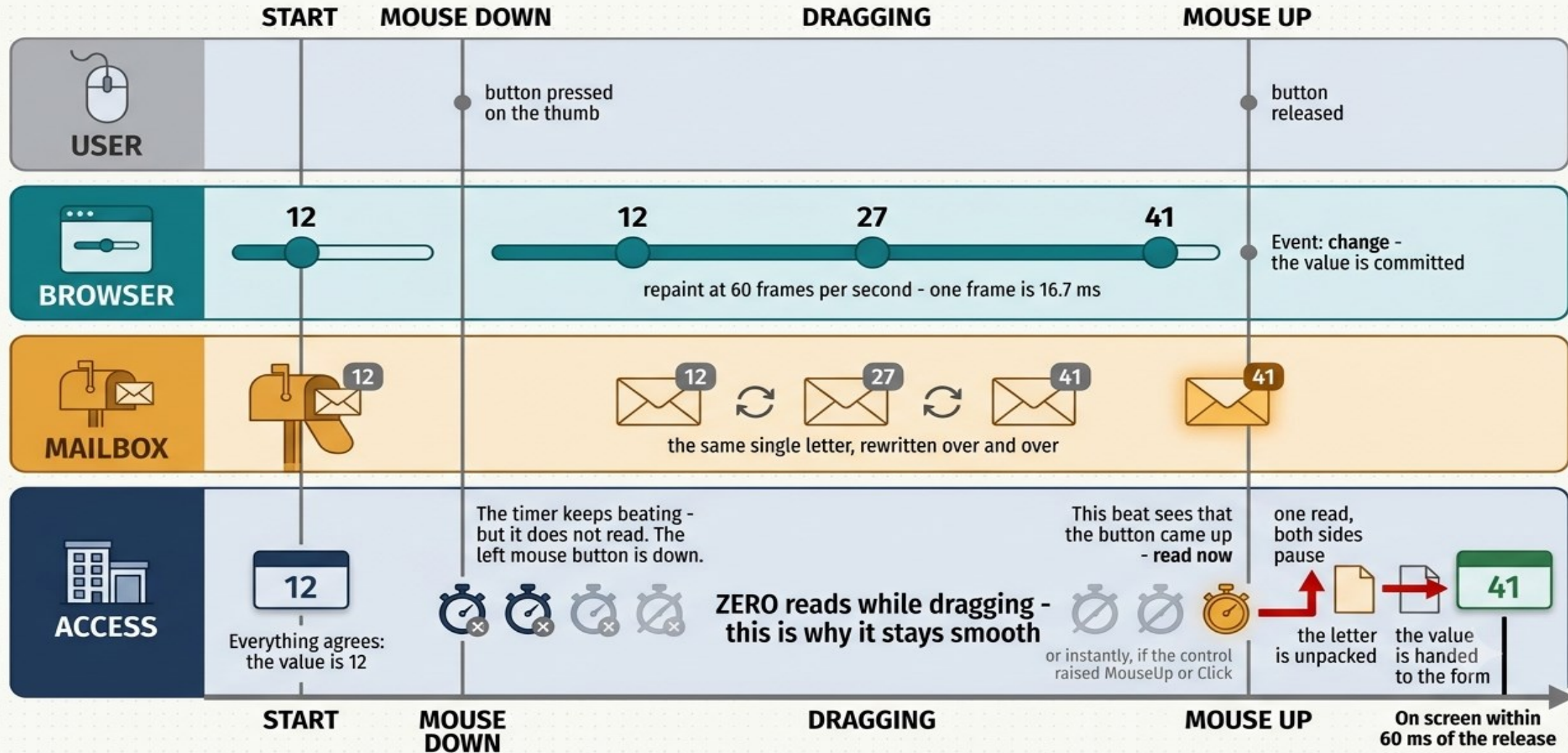


CHALK TALK TIME

Slider – Behind the Scenes

3 - One Drag, Second by Second

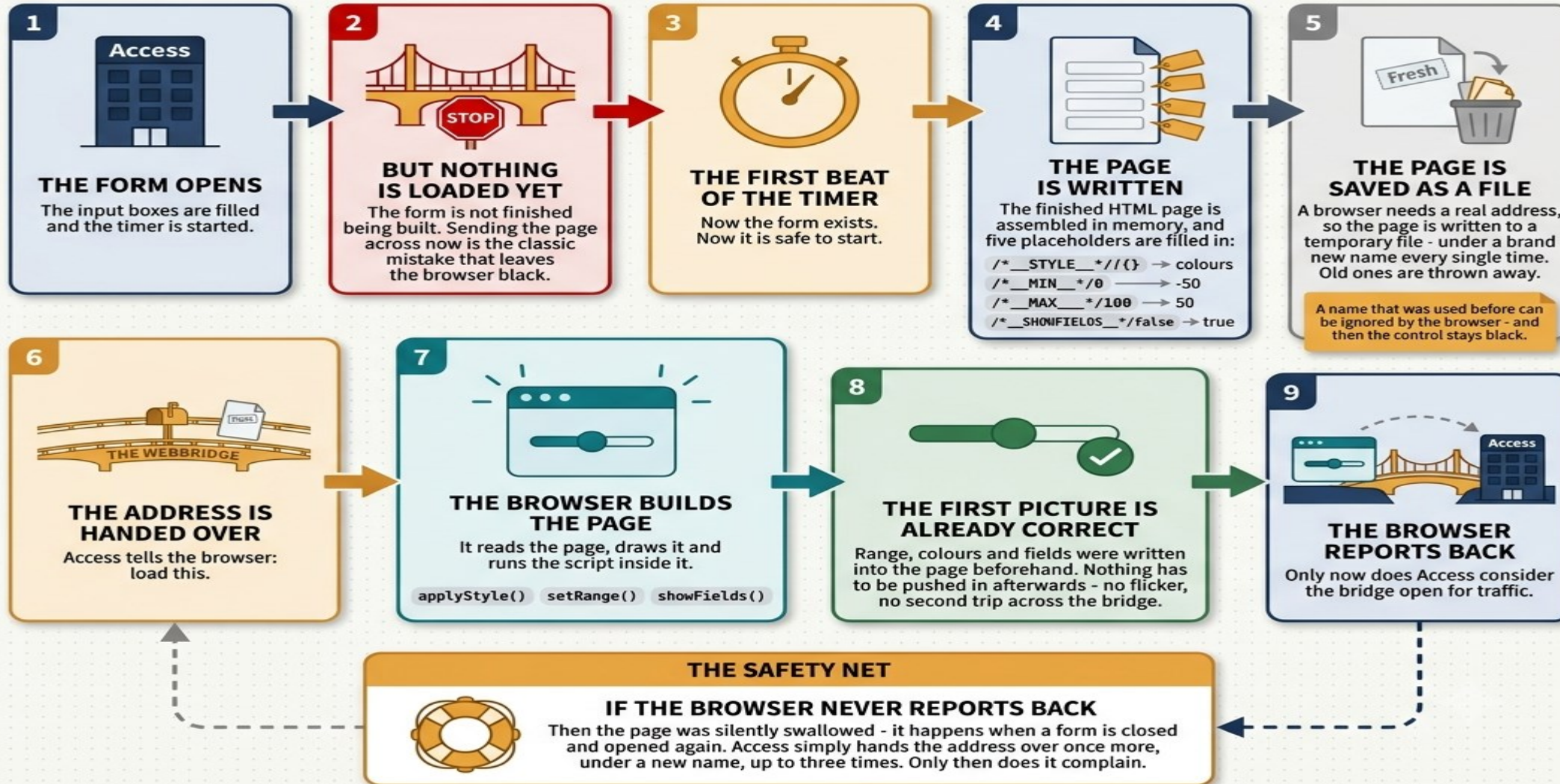
The value stays 12 everywhere - until the mouse button comes back up



4

4 - Opening the Form

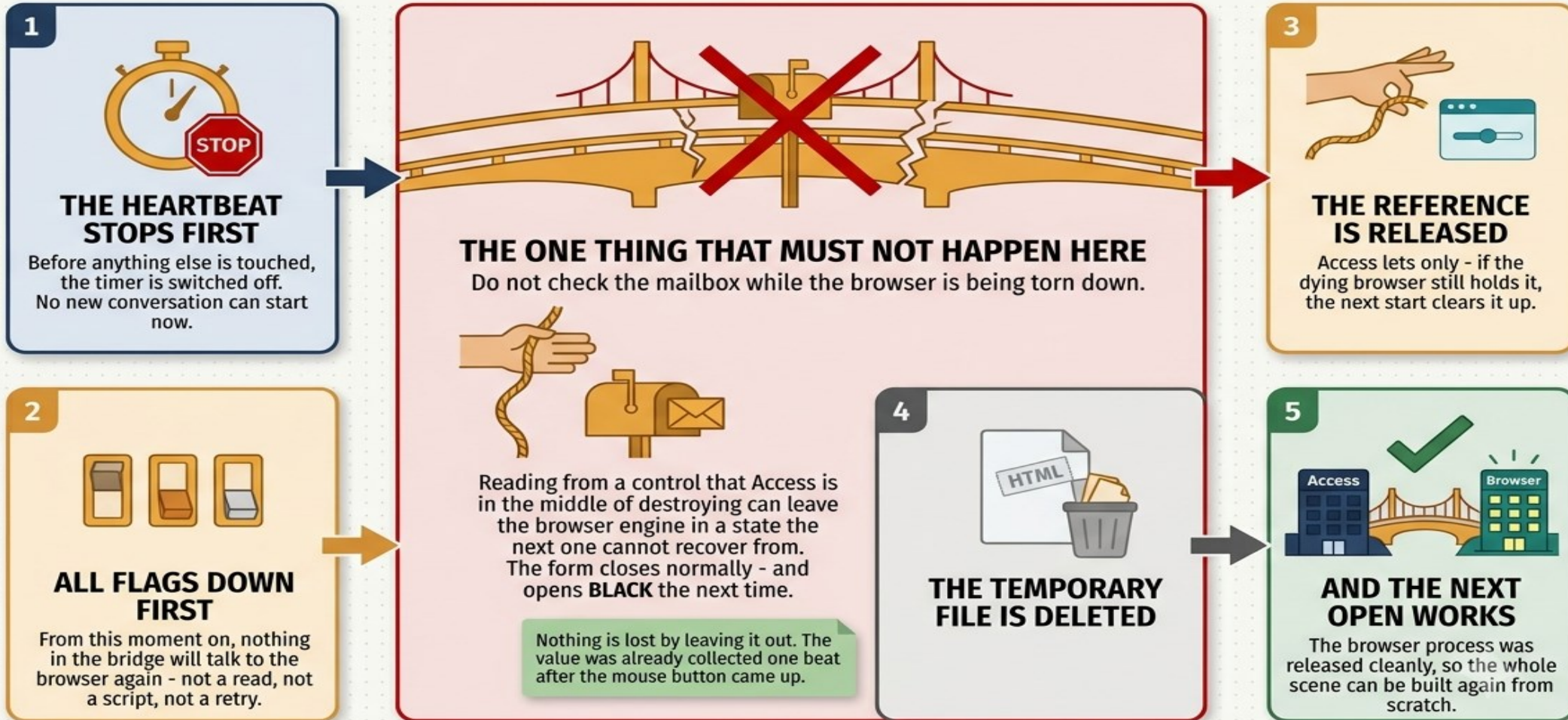
Nine steps from an empty window to a slider that already shows the right value



5

5 - Closing the Form

Let go of the browser in the right order - or the next open stays black



The rule in one line: talk to the browser only while it is fully alive.

6

HTML Slider inside MS Access: how it really works

What is cheap, what is expensive, and which part you can never make faster

COST OF EACH STEP

bar length is roughly to scale

	Build the HTML page in memory		about 1 ms
	Write it to the temporary file		2 - 5 ms
	Send a value across the bridge		nobody waits for an answer about 1 ms
	Check the mailbox - one read		small - but it stops the Access thread AND the browser mid-frame 1 - 10 ms
	The browser loads and paints the page		150 - 400 ms a full browser engine building a document in its own process

THE REAL TIME CONSUMER

- **Starting the page.** Everything Access does around it is noise by comparison.
- And it **cannot be fixed**. It is a browser engine doing exactly what a browser engine does. Nothing written in VBA makes that faster.
- So it is paid **ONCE**, at load - which is why the range, the colours and the fields are written into the page before it is ever opened.

THE ONE YOU CAN ACTUALLY CONTROL

- The read is not slow. It is **badly timed**.
- Twenty reads a second, each one stopping the browser inside a 16.7 ms frame, means the page misses its own deadline again and again. That is what sluggish looks like.
- The very same read, made **once**, after the button comes up, is invisible.



You cannot change HOW FAST the bridge is. You can change WHEN you use it.

Everything
you've seen
today is based
on exactly one
control.



**THE
GUI
GUY**

Dieterle
DIETERLE
programmierung