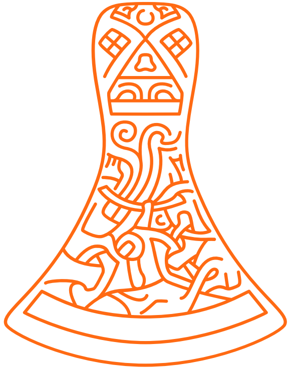
DYALOG

Glasgow 2024

WC Plugins



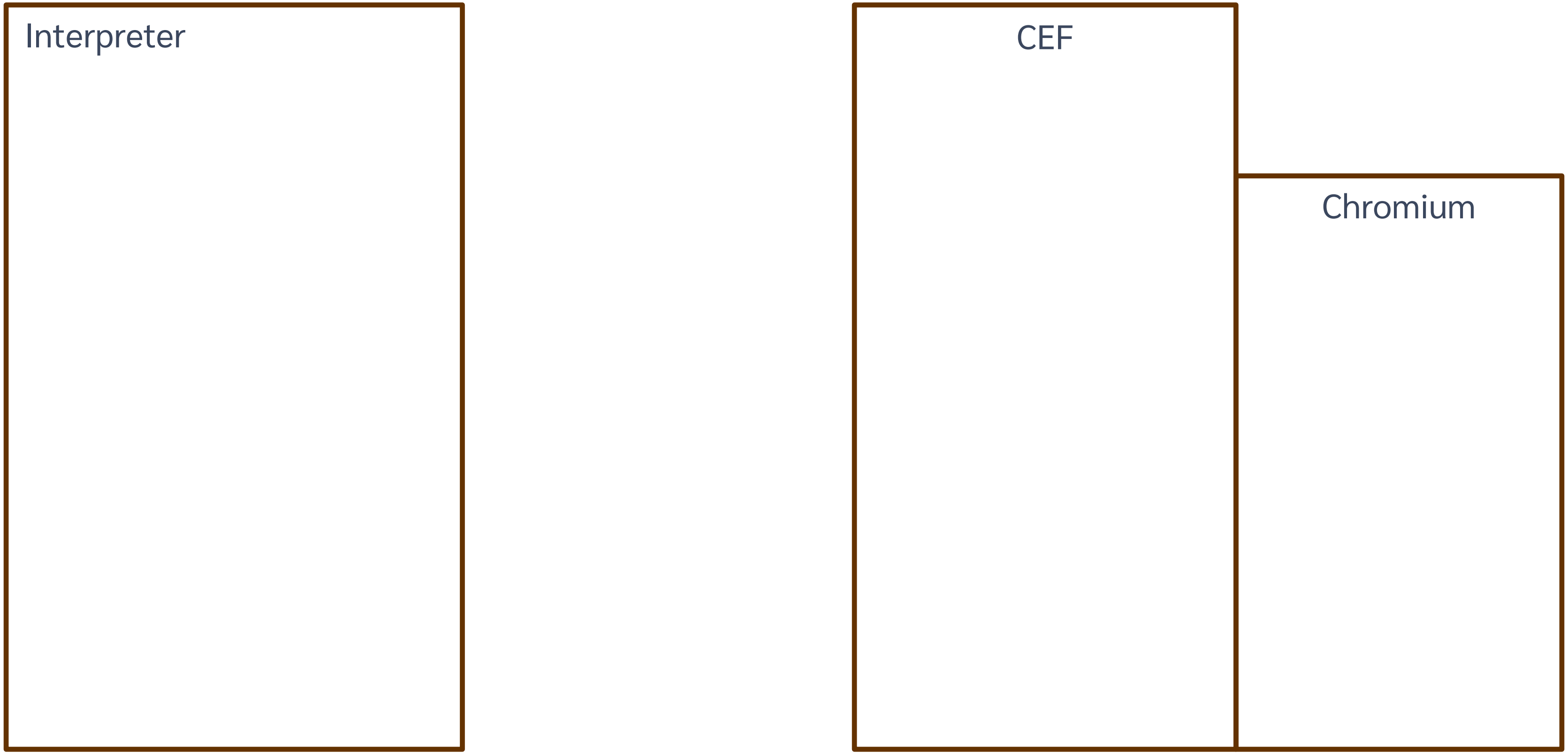
John Daintree

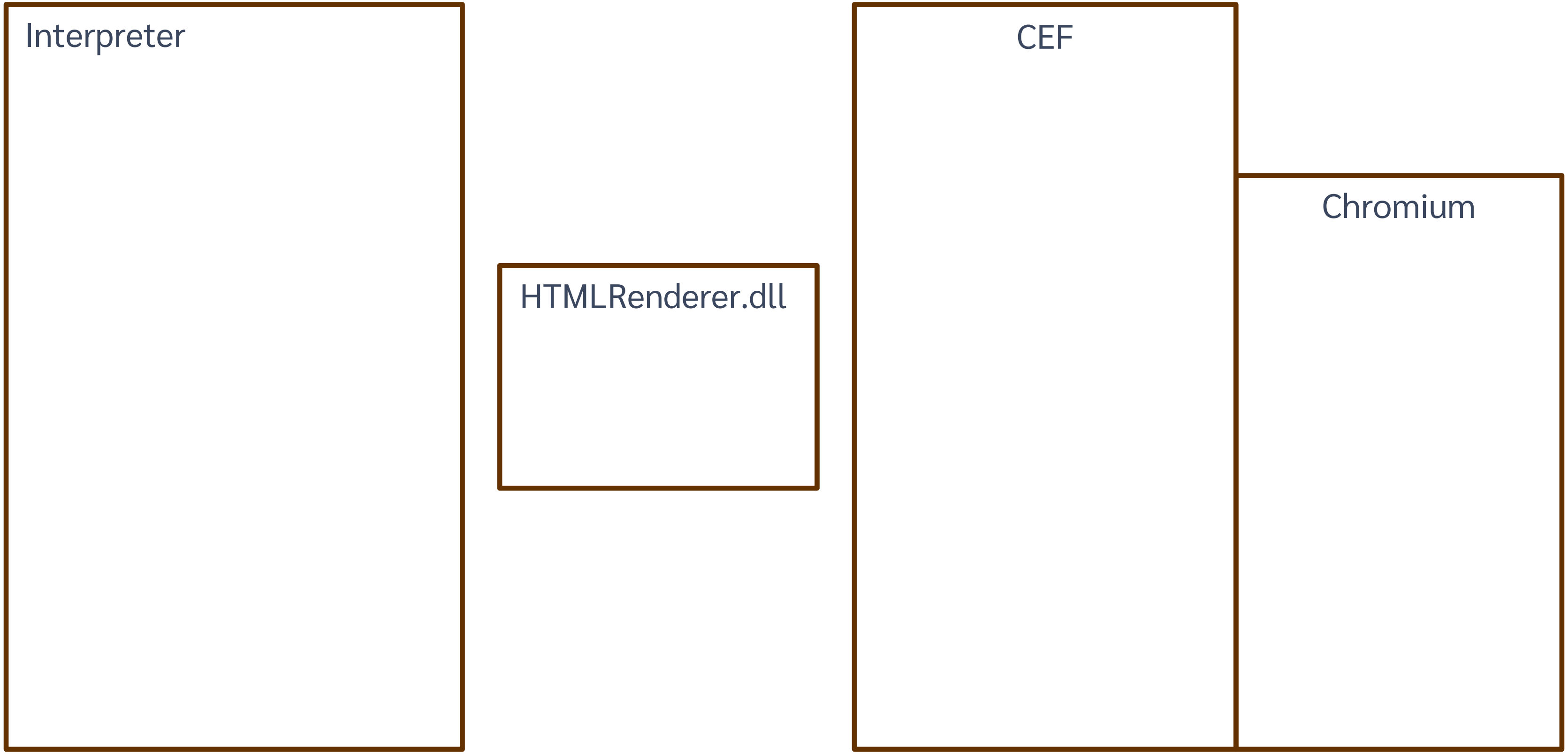
⎕WC Plugins

- What?
  - A toolkit to allow ⎕WC objects to be defined and implemented in external (3rd party) code

⎕WC Plugins

- Why?
  - Open sourcing HTMLRenderer
    - Faster migration to new versions of CEF
  - Make Conga "First Class"
  - Extensibility


Interpreter
CEF
Chromium

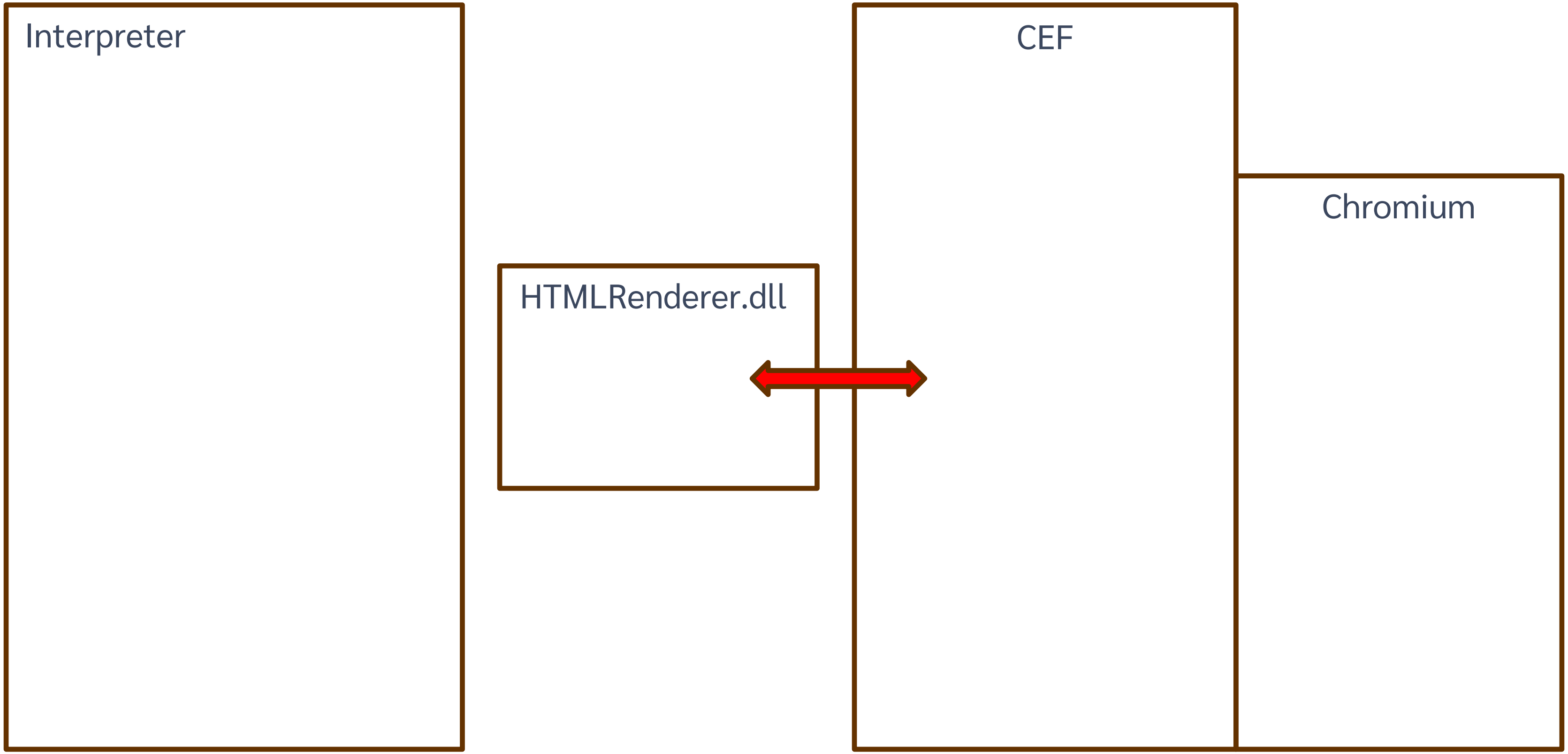


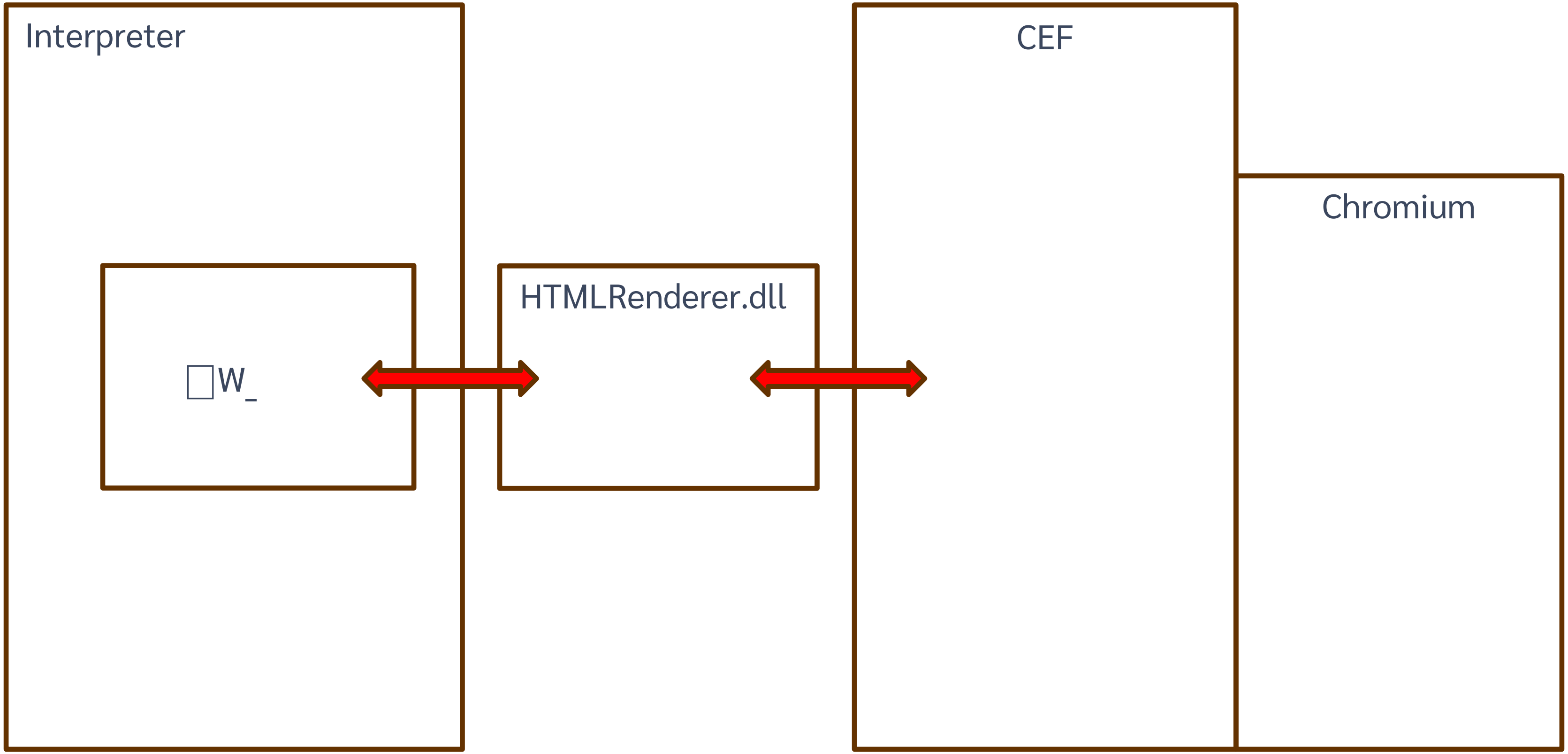
Interpreter

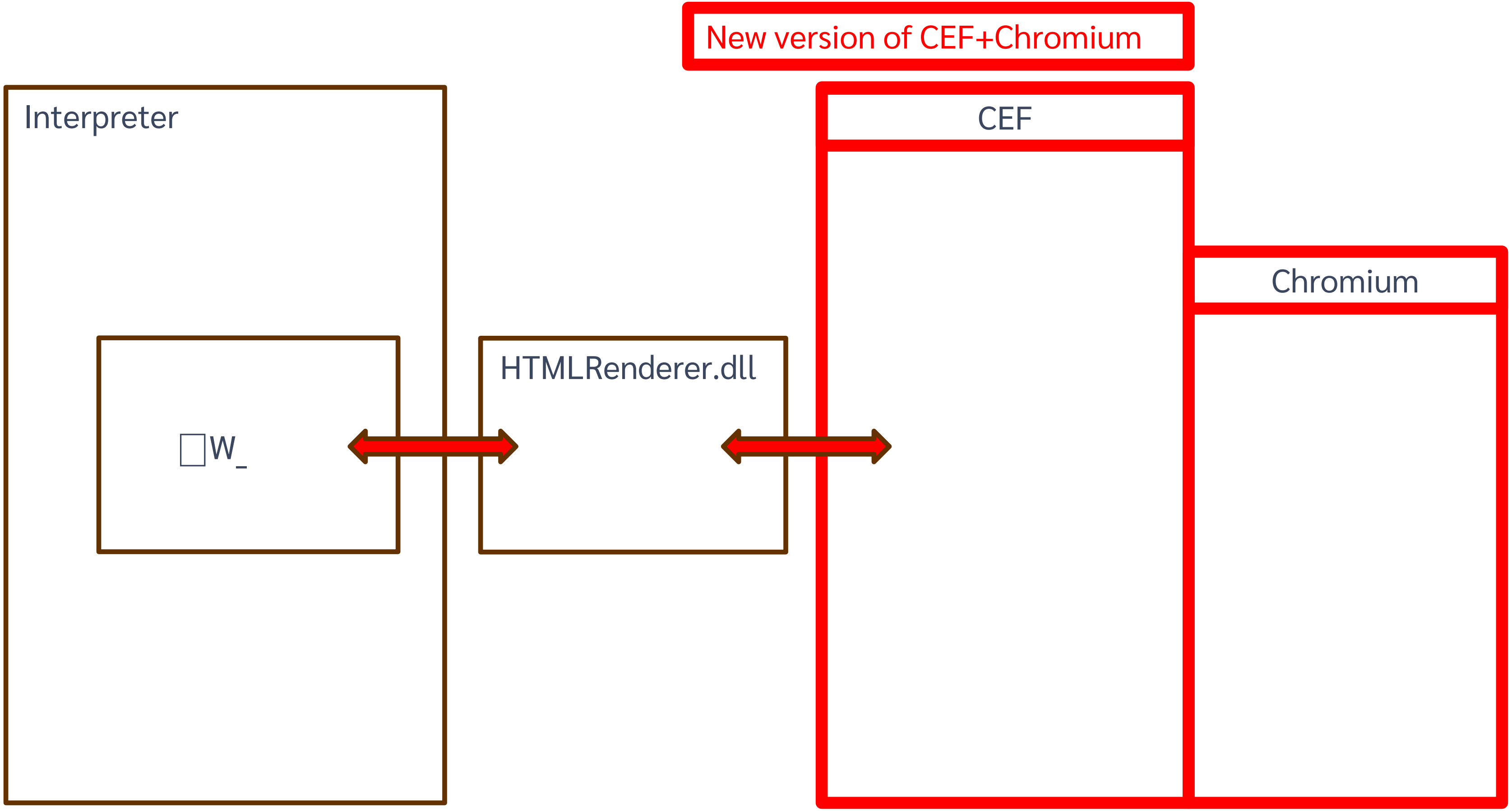
HTMLRenderer.dll

CEF

Chromium


Interpreter
HTMLRenderer.dll
CEF
Chromium


Interpreter
⎕W_
HTMLRenderer.dll
CEF
Chromium



New version of CEF+Chromium

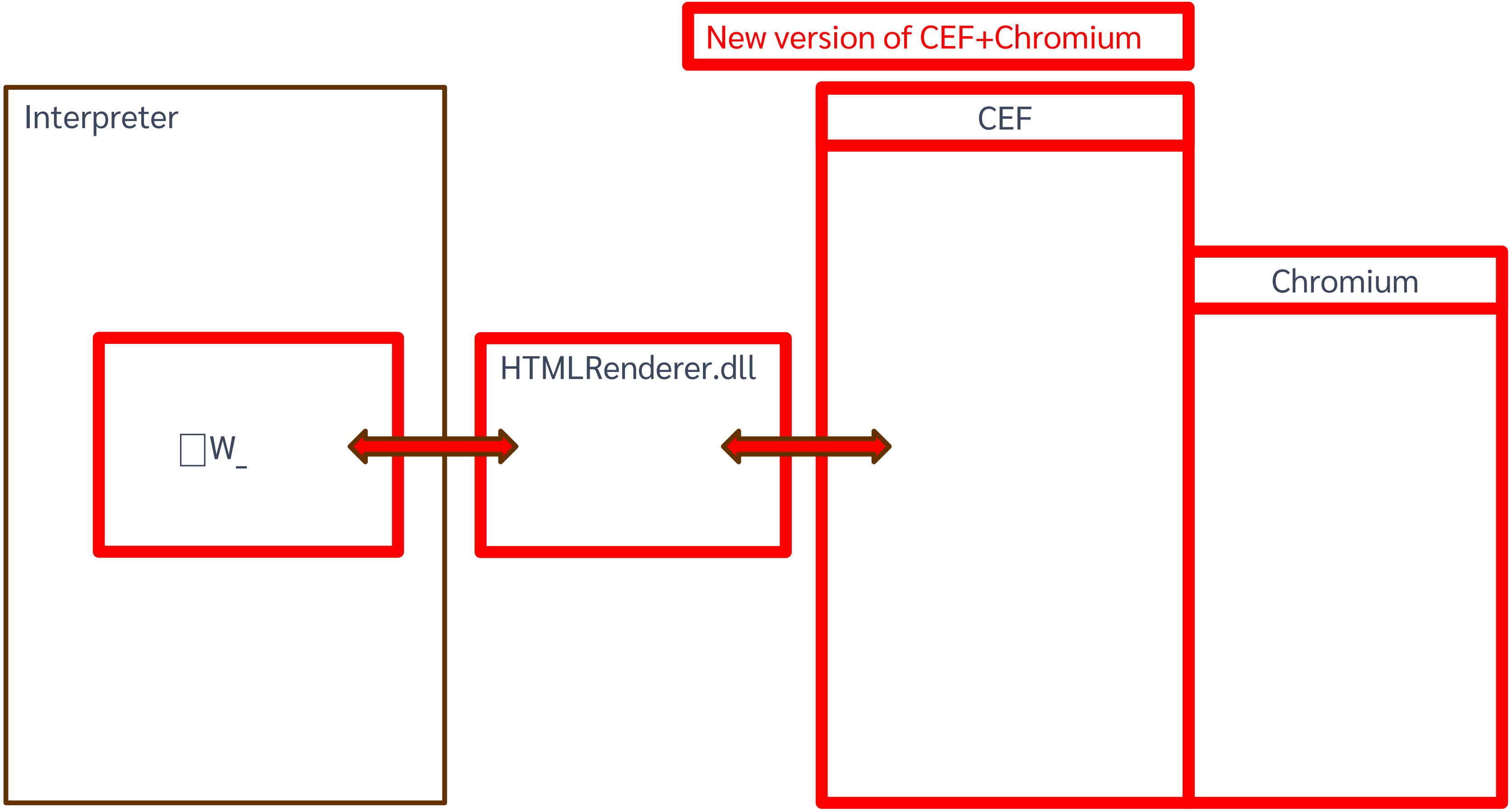
Interpreter

⎕W_

HTMLRenderer.dll

CEF

Chromium



New version of CEF+Chromium

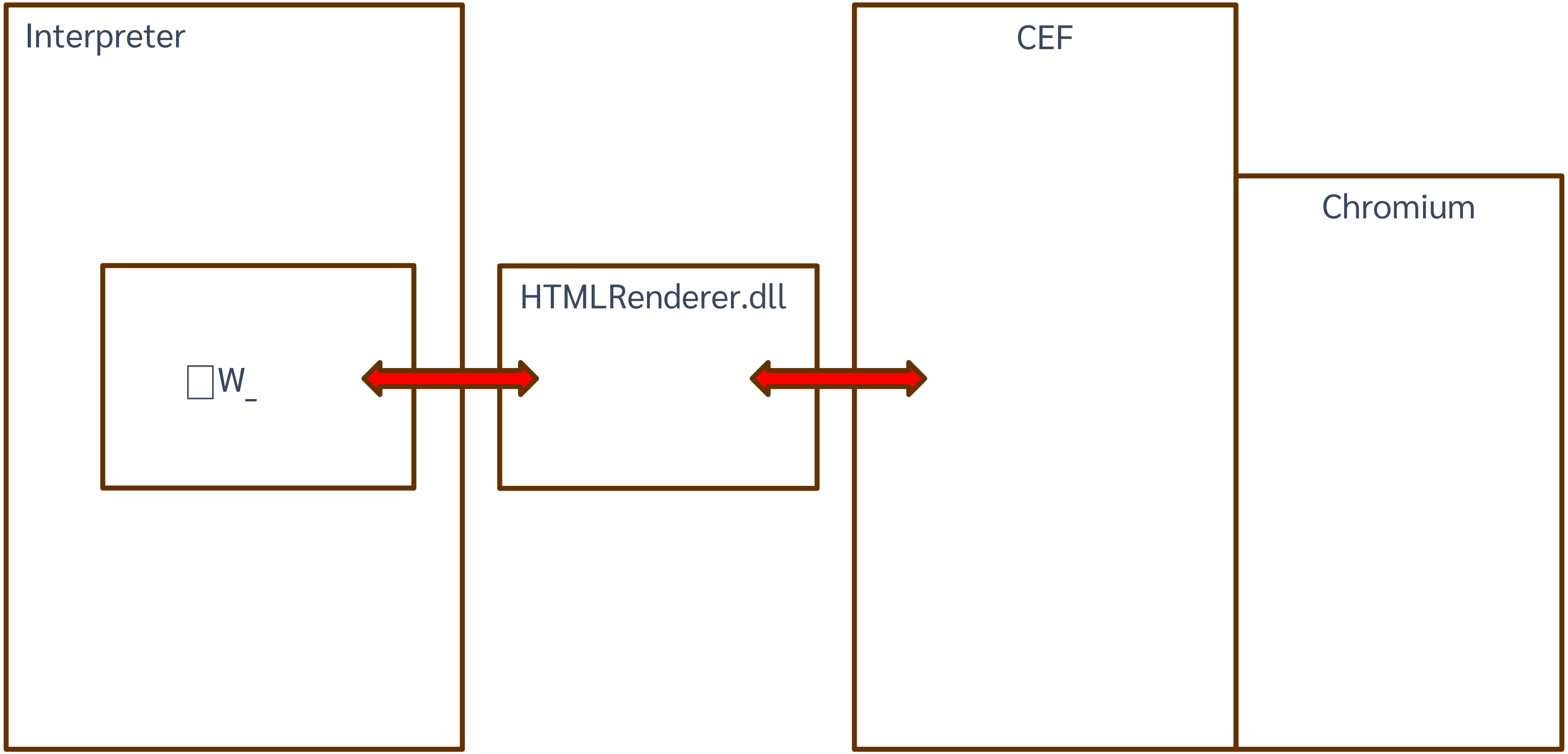
Interpreter

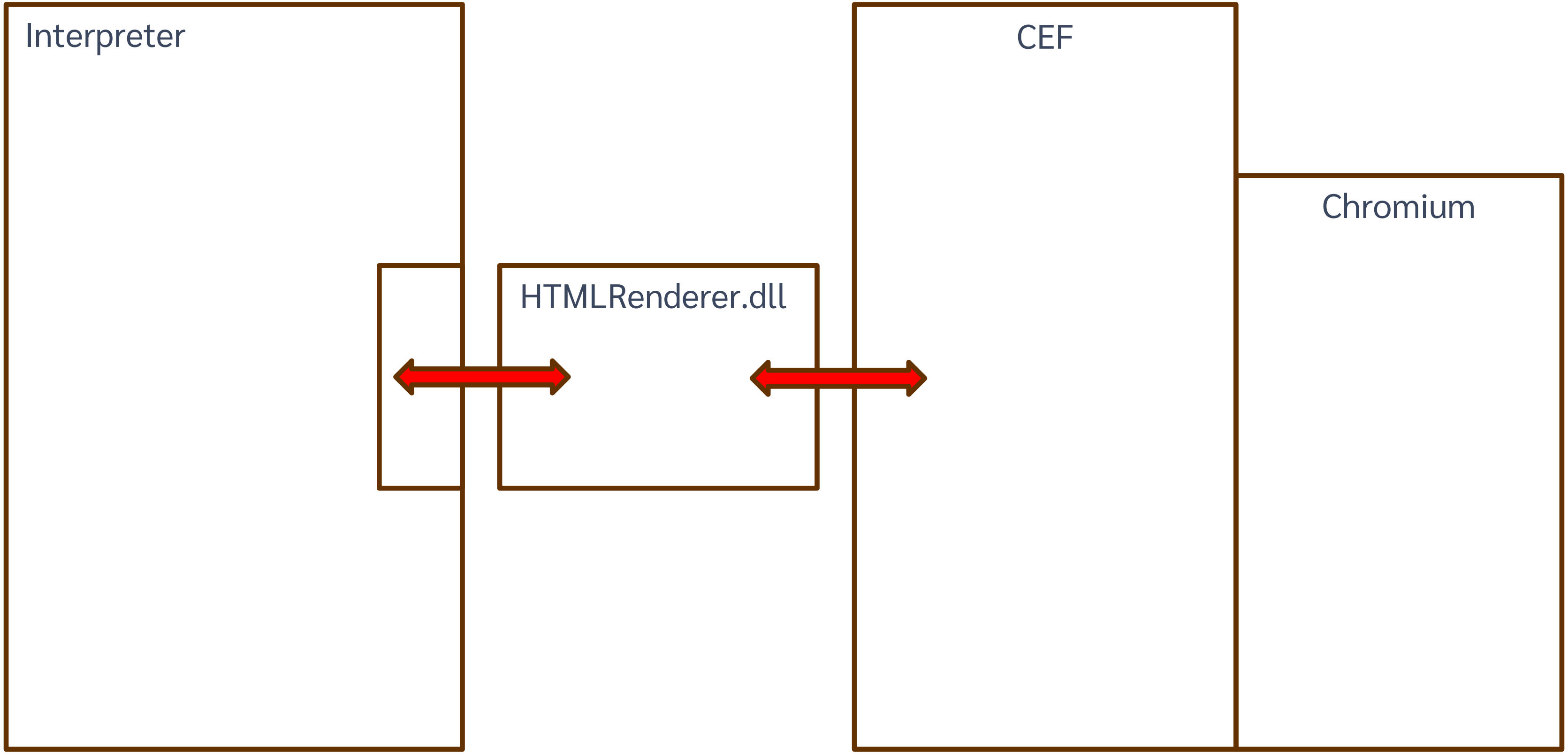
⎕W_

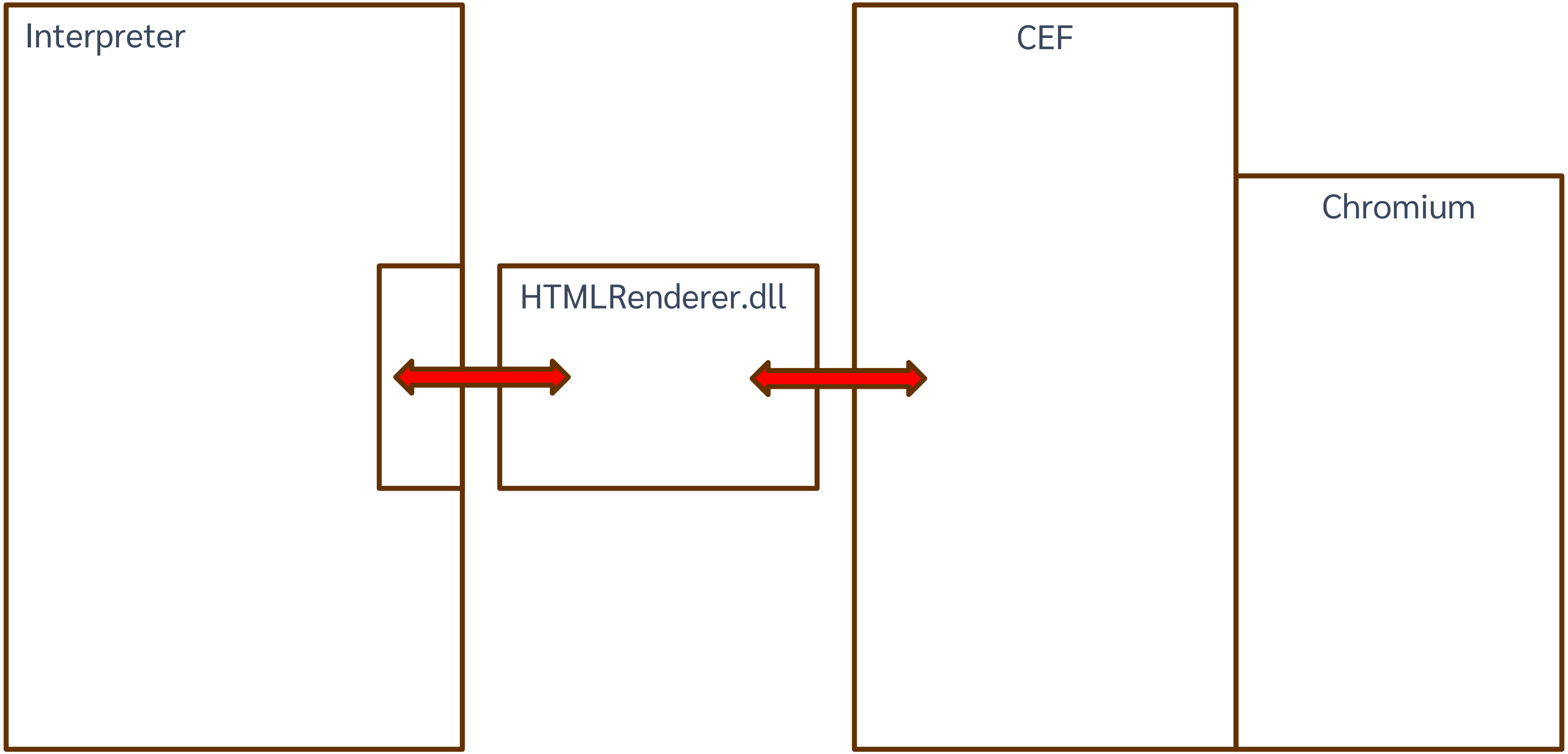
HTMLRenderer.dll

CEF

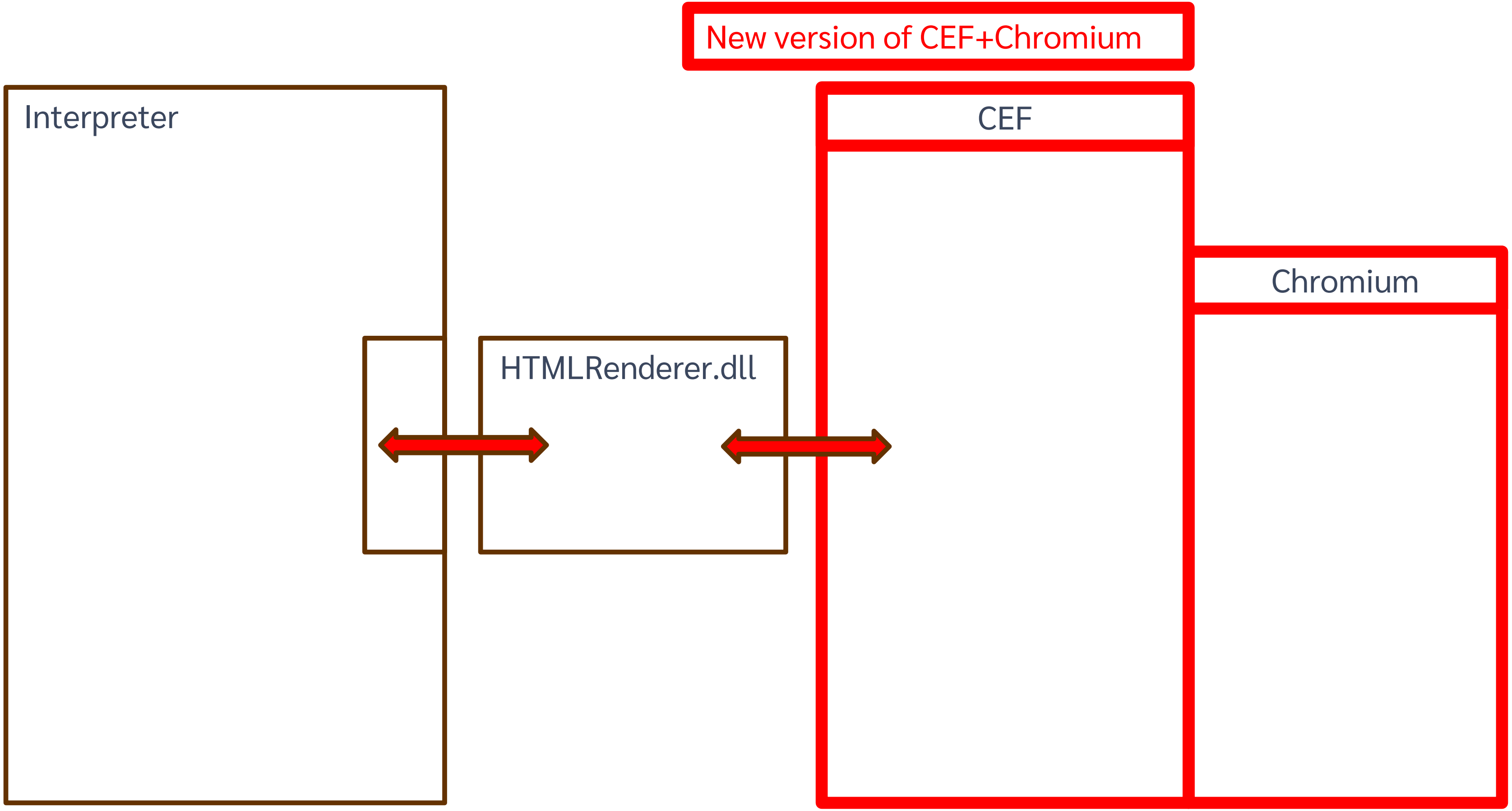
Chromium

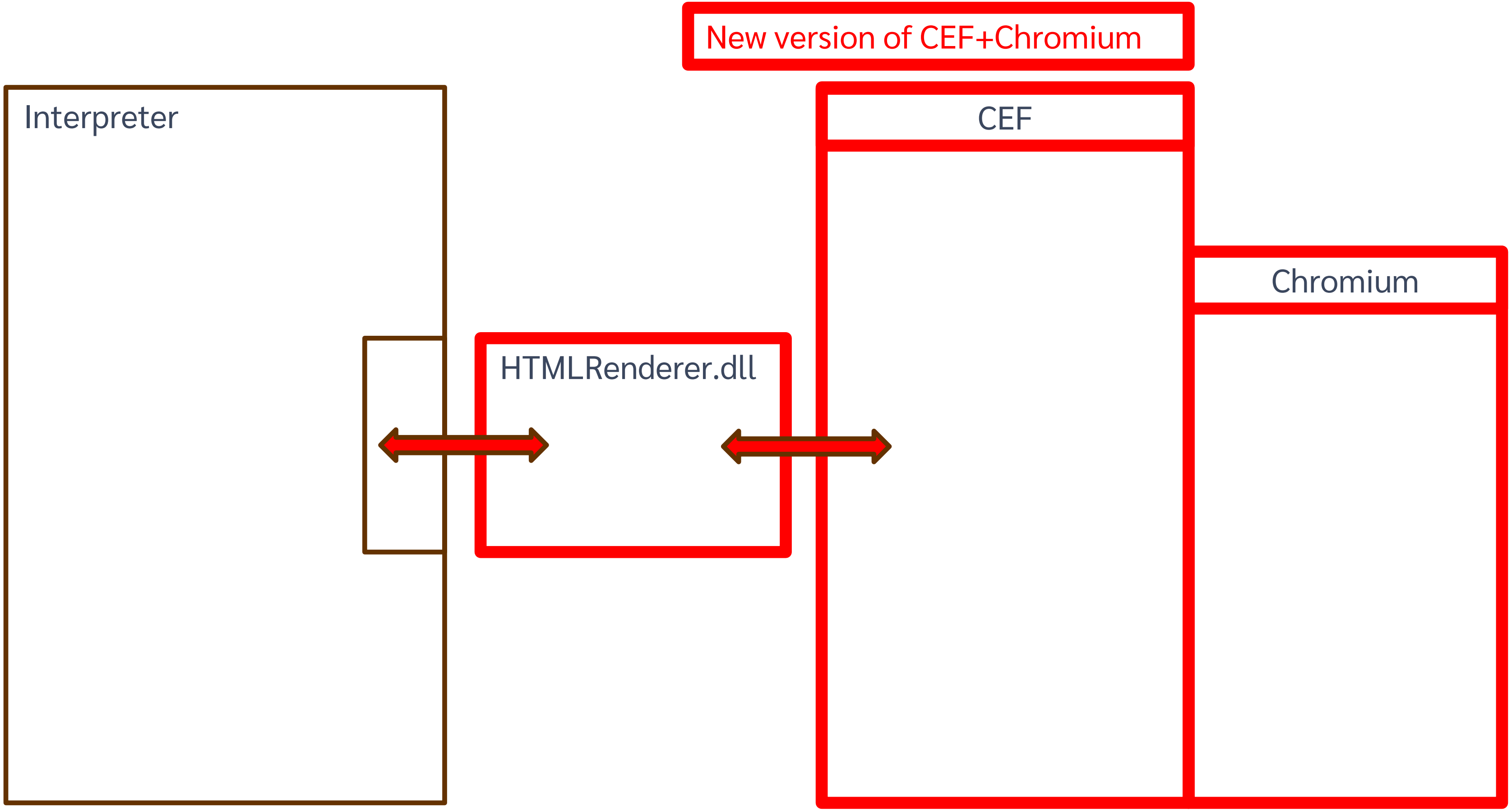

Interpreter
⎕W_
HTMLRenderer.dll
CEF
Chromium

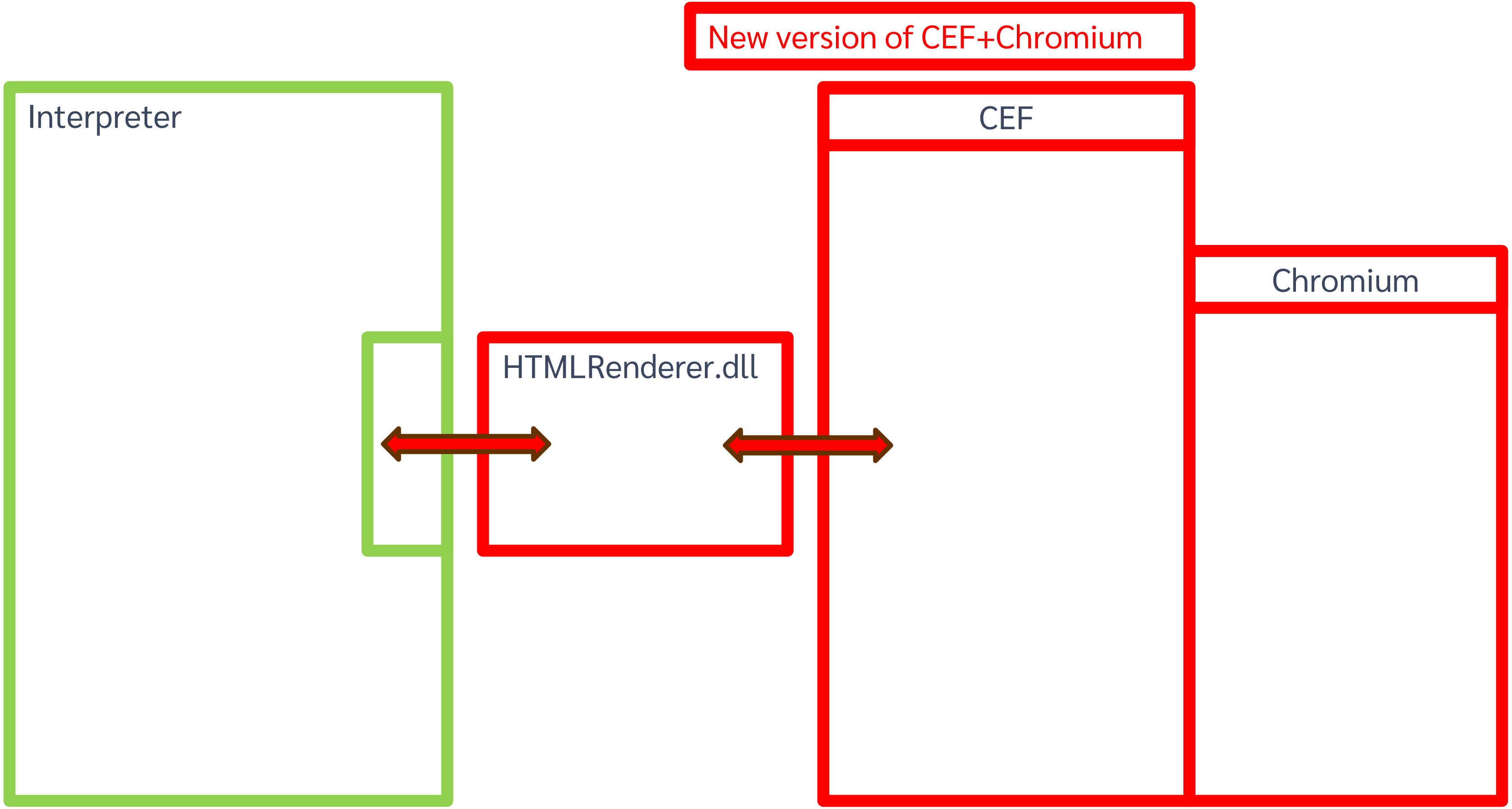

Interpreter
HTMLRenderer.dll
CEF
Chromium



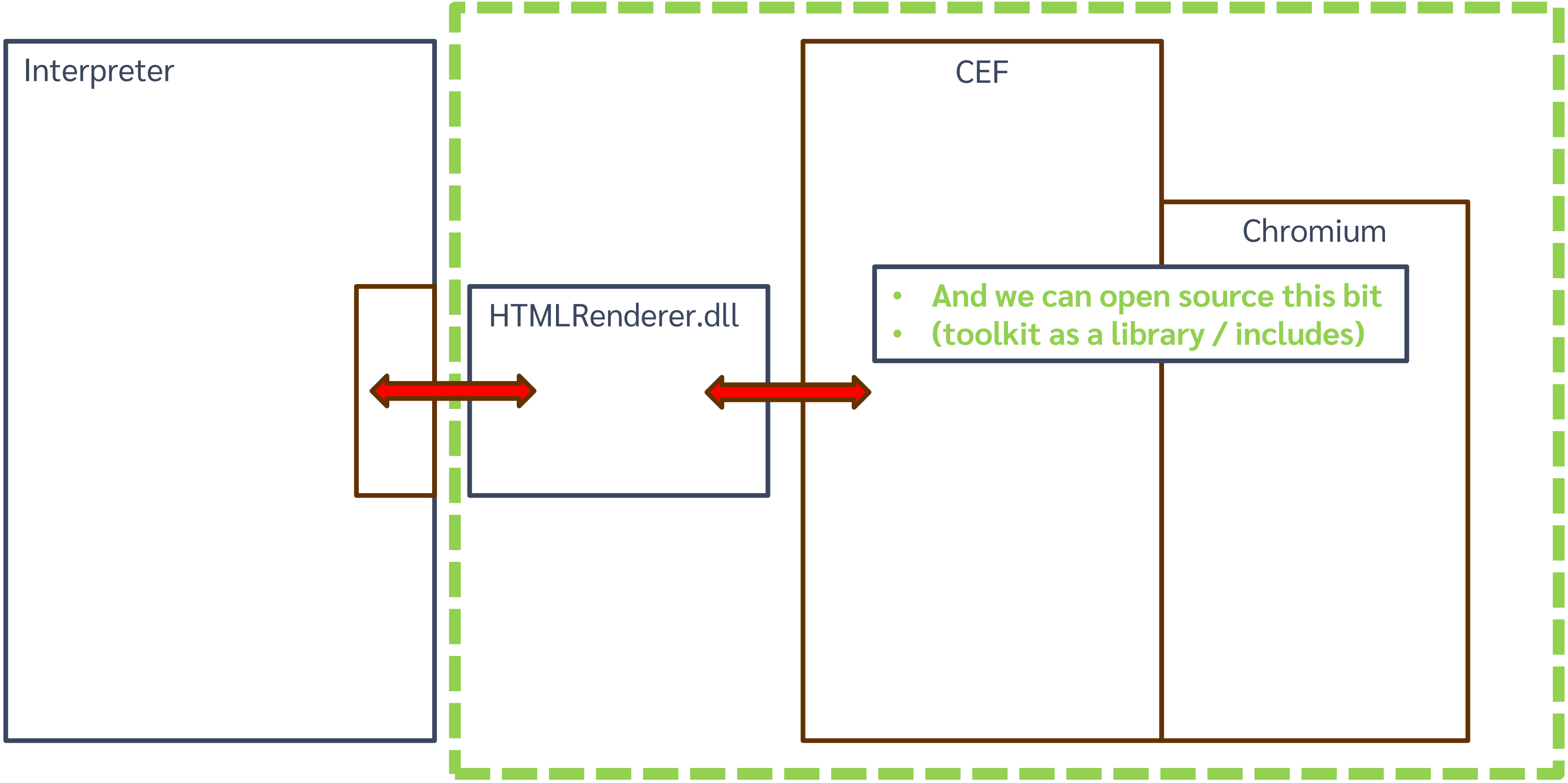
WC Plugins


New version of CEF+Chromium
Interpreter
HTMLRenderer.dll
CEF
Chromium


New version of CEF+Chromium
Interpreter
HTMLRenderer.dll
CEF
Chromium

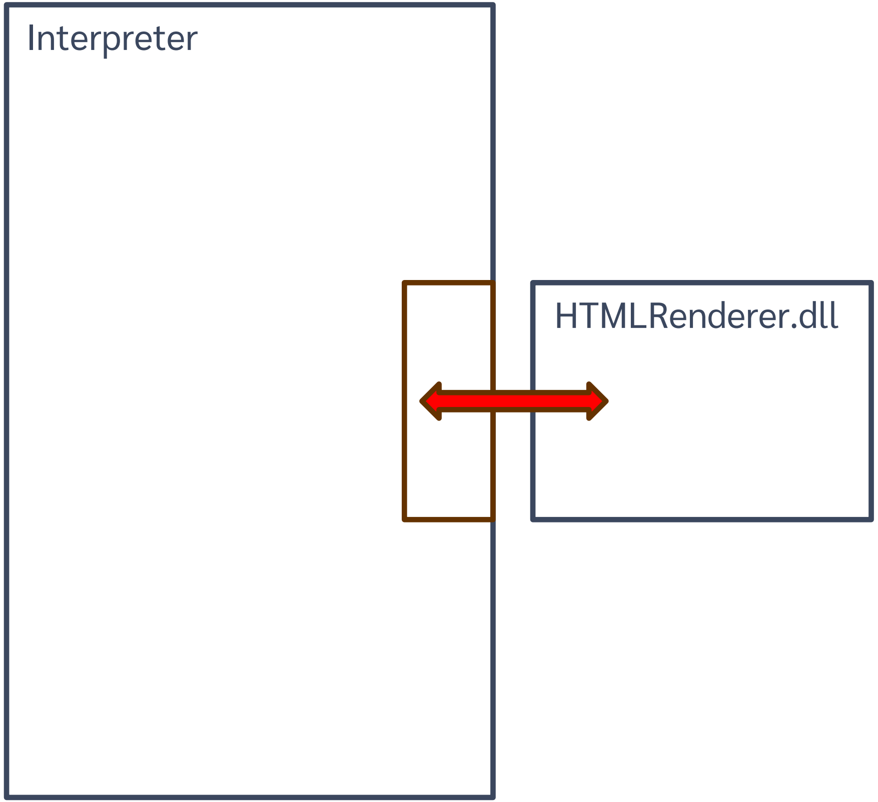

New version of CEF+Chromium
Interpreter
CEF
Chromium
HTMLRenderer.dll

WC Plugins

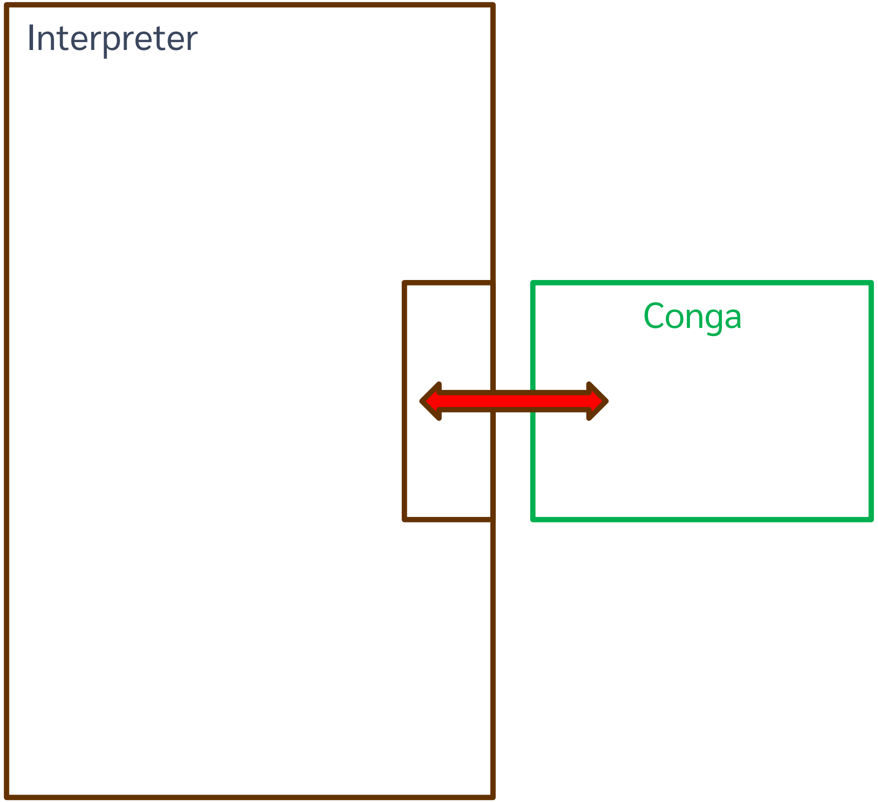


- And we can open source this bit
- (toolkit as a library / includes)

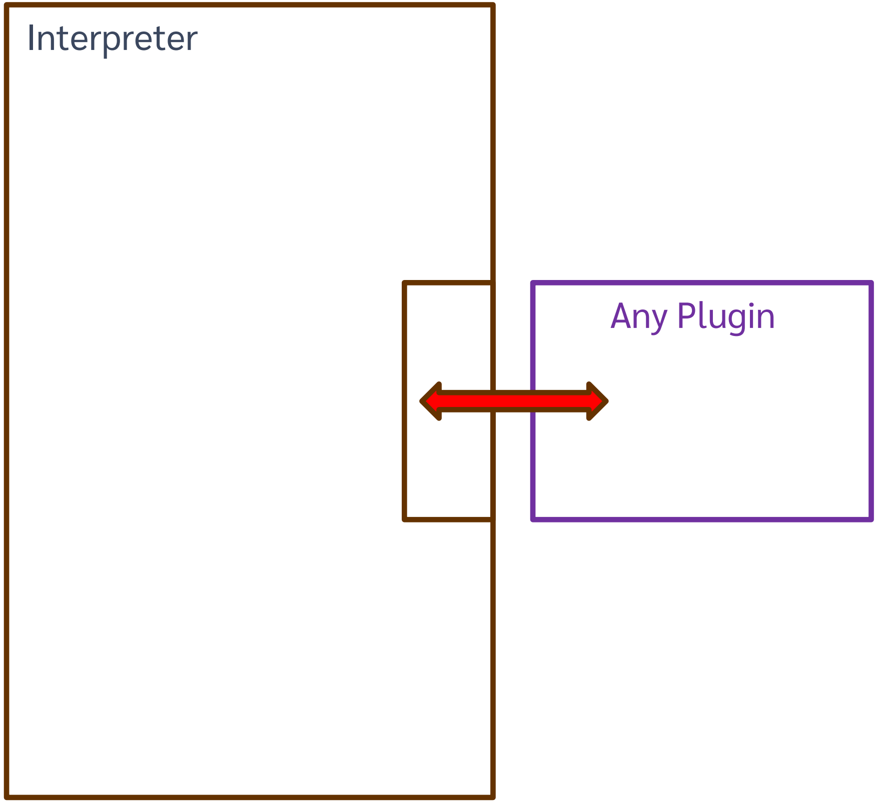
WC Plugins



WC Plugins



Interpreter
Conga

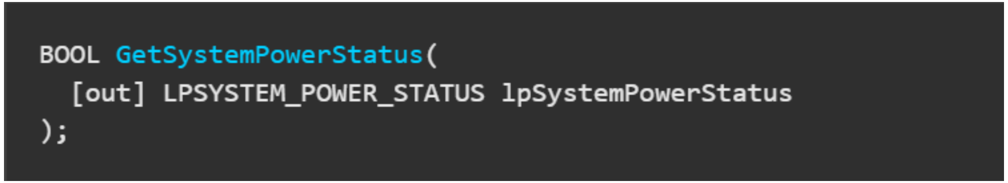


WC Plugins

Let's build a plugin

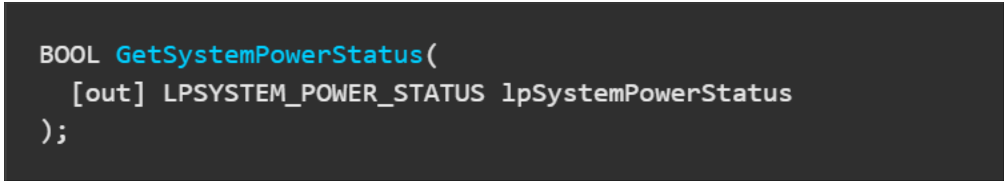
(We've got plenty of time)

System Power Status

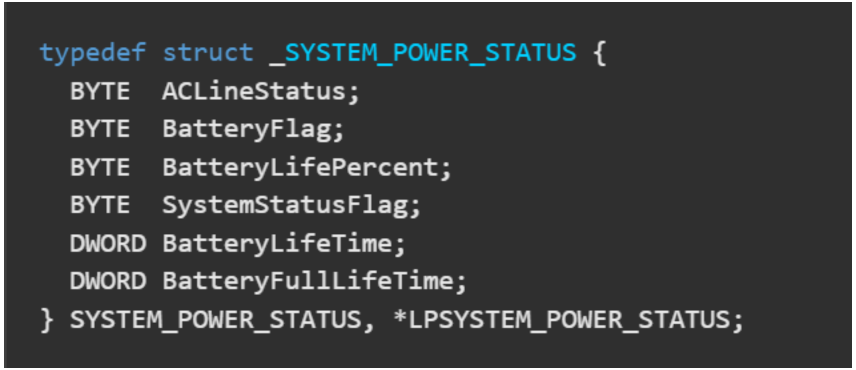


```
BOOL GetSystemPowerStatus(
  [out] LPSYSTEM_POWER_STATUS lpSystemPowerStatus
);
```

System Power Status



```
BOOL GetSystemPowerStatus(
  [out] LPSYSTEM_POWER_STATUS lpSystemPowerStatus
);
```



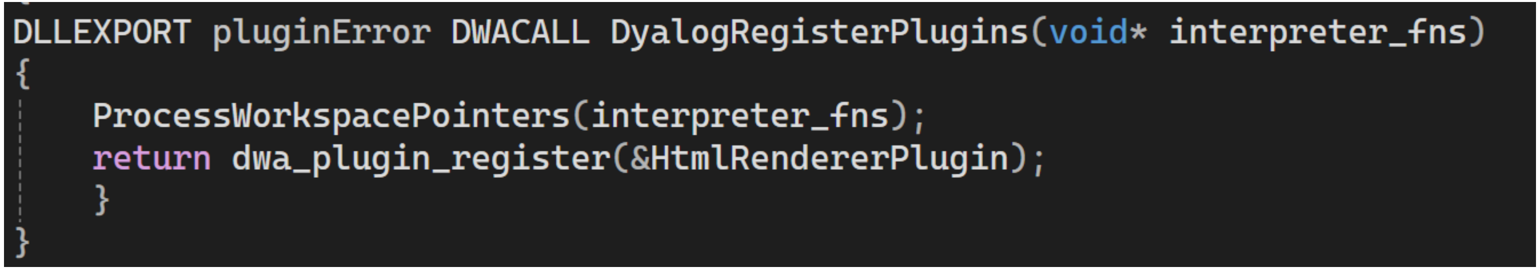
```
typedef struct _SYSTEM_POWER_STATUS {
  BYTE  ACLineStatus;
  BYTE  BatteryFlag;
  BYTE  BatteryLifePercent;
  BYTE  SystemStatusFlag;
  DWORD BatteryLifeTime;
  DWORD BatteryFullLifeTime;
} SYSTEM_POWER_STATUS, *LPSYSTEM_POWER_STATUS;
```

Pick a number

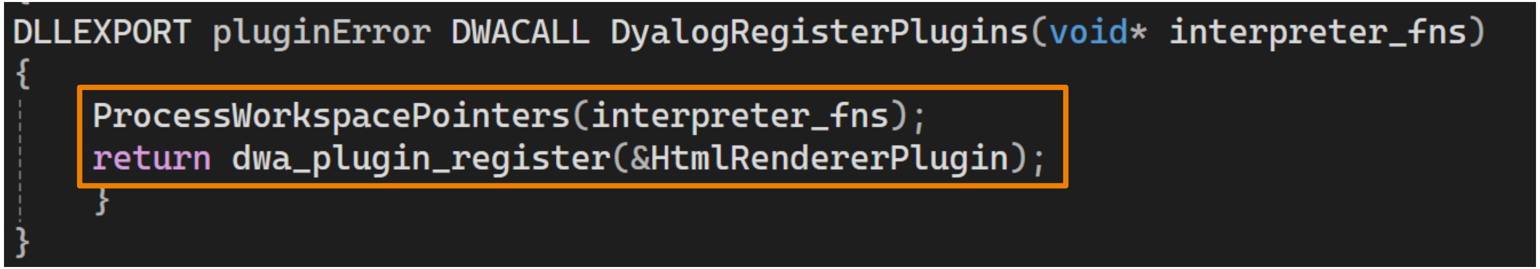
Demo 1

HtmlRenderer

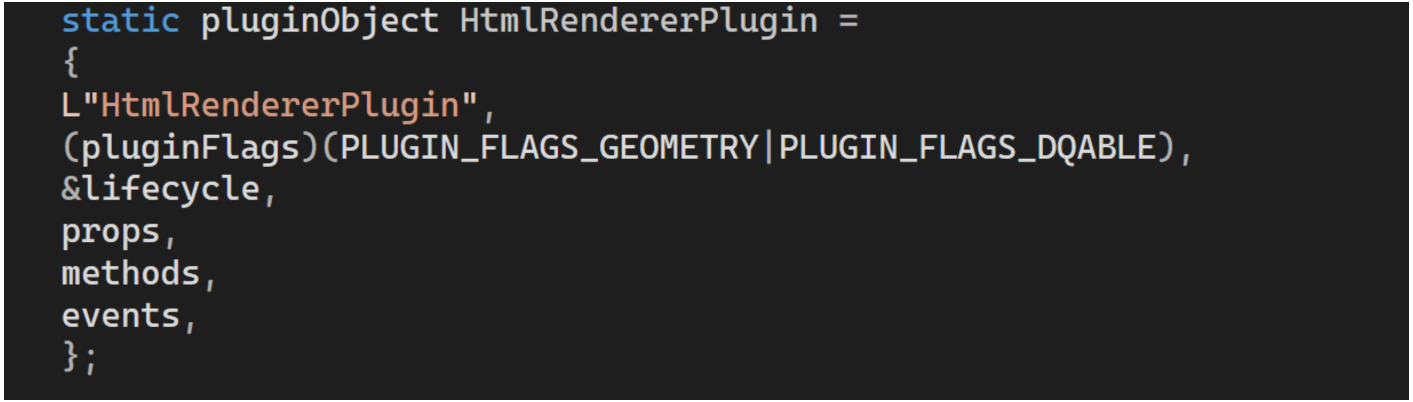
Demo 2



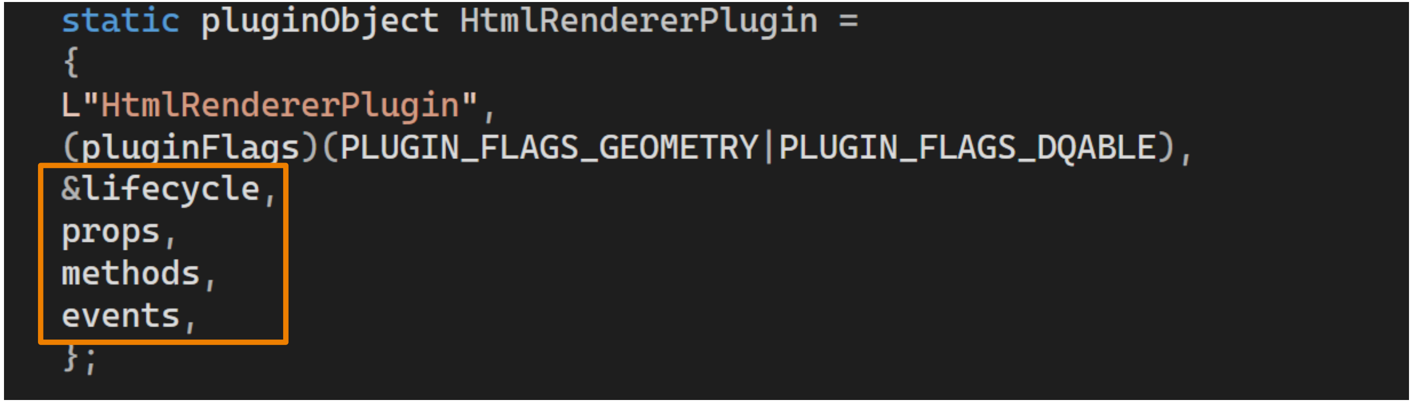
```
DLLEXPORT pluginError DWACALL DyalogRegisterPlugins(void* interpreter_fns)
{
    ProcessWorkspacePointers(interpreter_fns);
    return dwa_plugin_register(&HtmlRendererPlugin);
    }
}
```



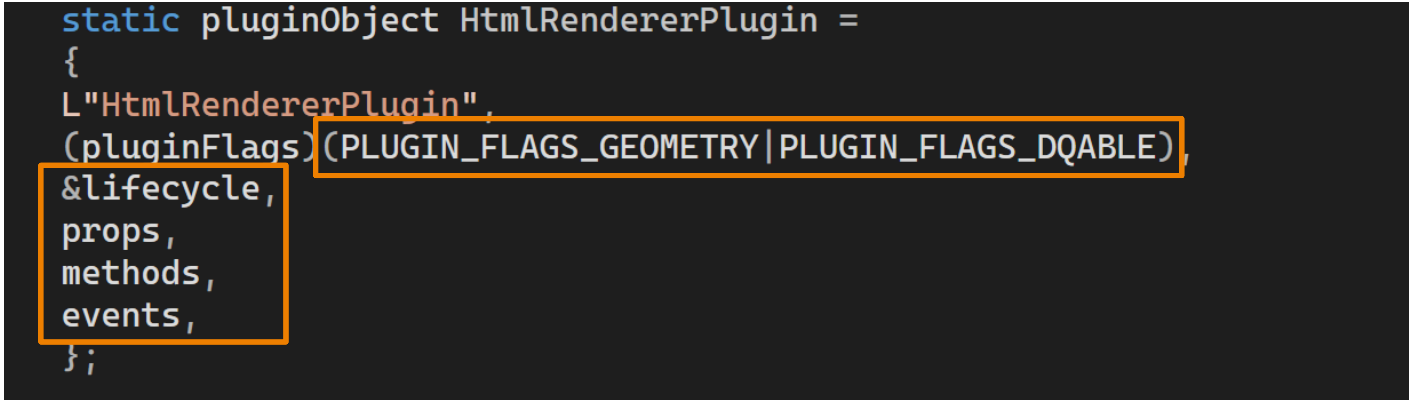
```
DLLEXPORT pluginError DWACALL DyalogRegisterPlugins(void* interpreter_fns)
{
    ProcessWorkspacePointers(interpreter_fns);
    return dwa_plugin_register(&HtmlRendererPlugin);
    }
}
```



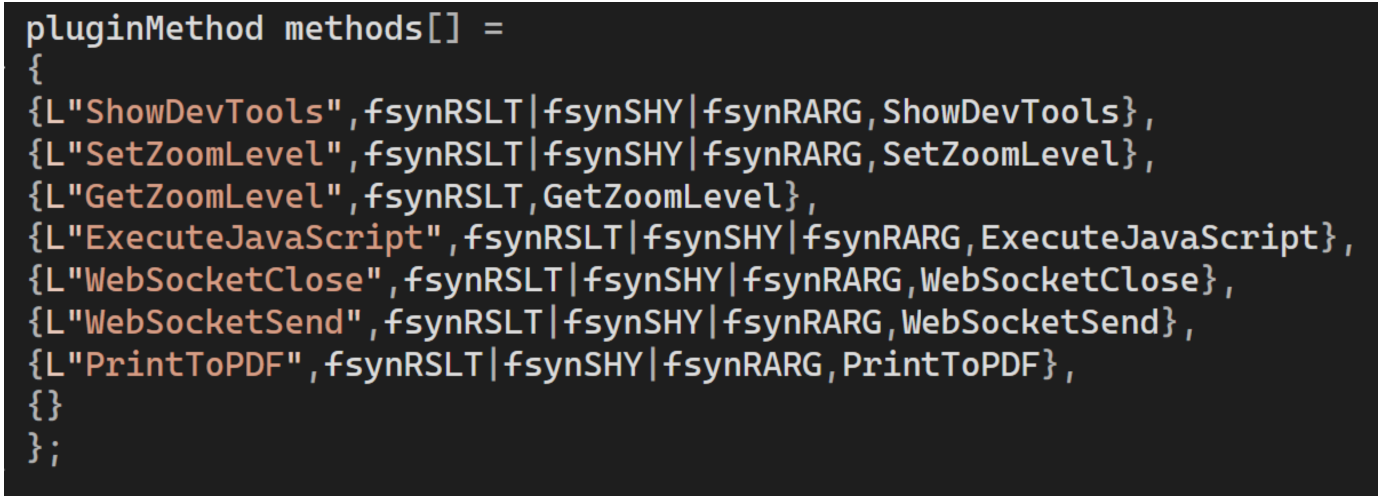
```
static pluginObject HtmlRendererPlugin =
{
L"HtmlRendererPlugin",
(pluginFlags)(PLUGIN_FLAGS_GEOMETRY|PLUGIN_FLAGS_DQABLE),
&lifecycle,
props,
methods,
events,
};
```



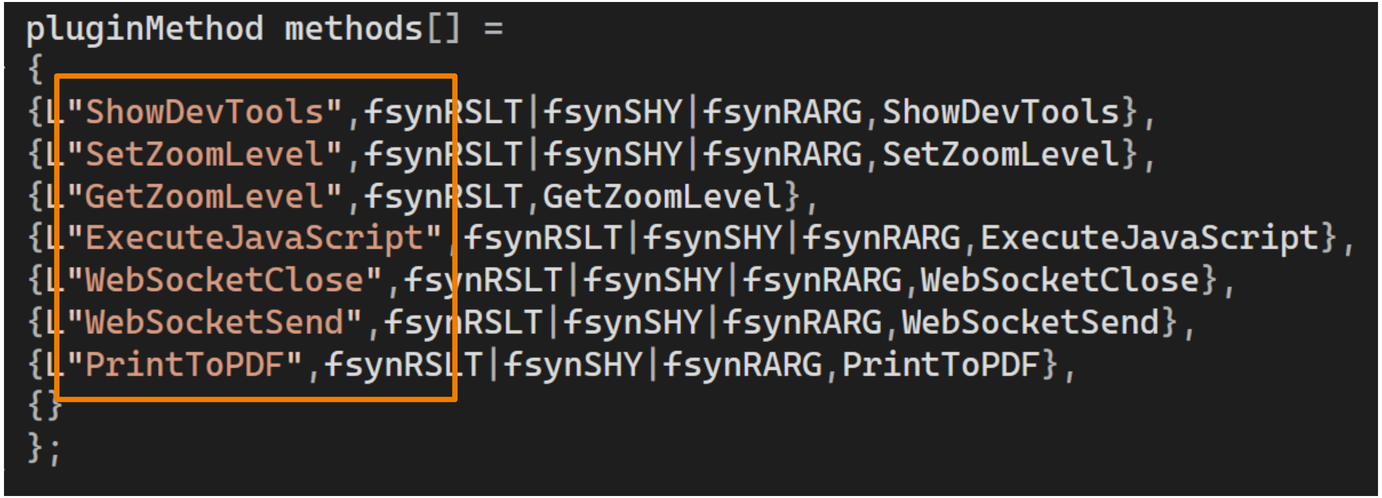
```
static pluginObject HtmlRendererPlugin =
{
L"HtmlRendererPlugin",
(pluginFlags)(PLUGIN_FLAGS_GEOMETRY|PLUGIN_FLAGS_DQABLE),
&lifecycle,
props,
methods,
events,
};
```



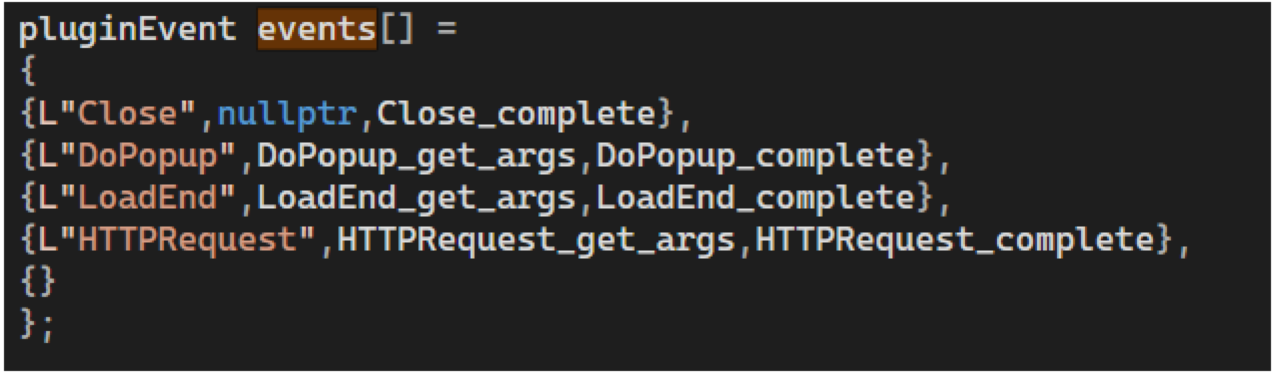
```
static pluginObject HtmlRendererPlugin =
{
L"HtmlRendererPlugin",
(pluginFlags)(PLUGIN_FLAGS_GEOMETRY|PLUGIN_FLAGS_DQABLE),
&lifecycle,
props,
methods,
events,
};
```



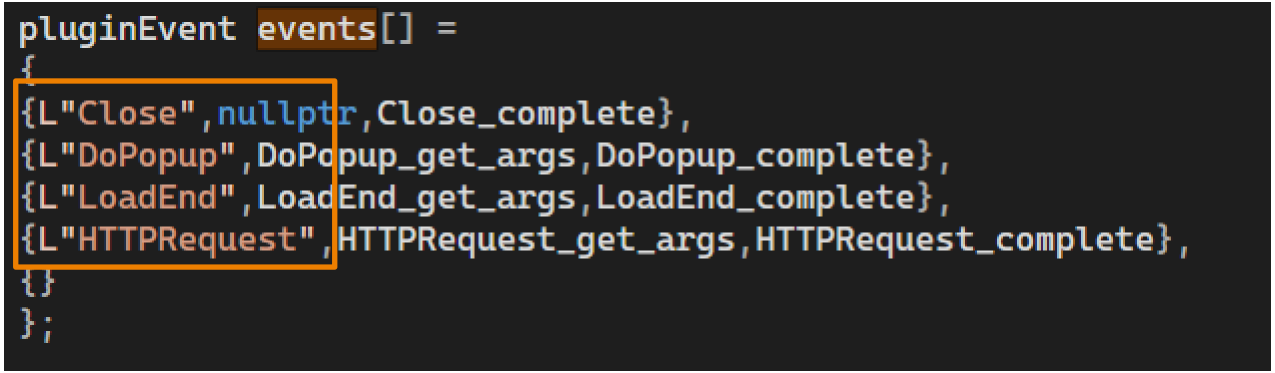
```
pluginMethod methods[] =
{
{L"ShowDevTools",fsynRSLT|fsynSHY|fsynRARG,ShowDevTools},
{L"SetZoomLevel",fsynRSLT|fsynSHY|fsynRARG,SetZoomLevel},
{L"GetZoomLevel",fsynRSLT,GetZoomLevel},
{L"ExecuteJavaScript",fsynRSLT|fsynSHY|fsynRARG,ExecuteJavaScript},
{L"WebSocketClose",fsynRSLT|fsynSHY|fsynRARG,WebSocketClose},
{L"WebSocketSend",fsynRSLT|fsynSHY|fsynRARG,WebSocketSend},
{L"PrintToPDF",fsynRSLT|fsynSHY|fsynRARG,PrintToPDF},
{}
};
```



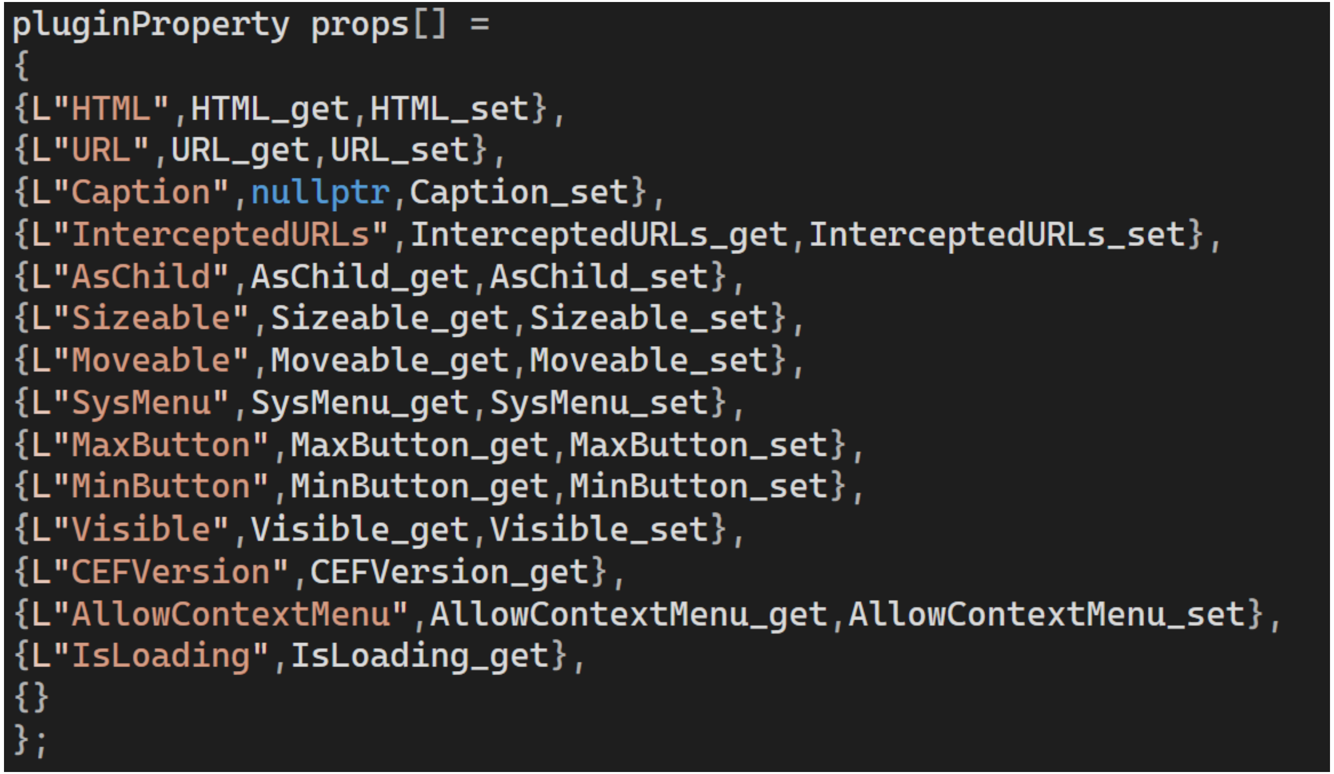
```
pluginMethod methods[] =
{
{L"ShowDevTools",fsynRSLT|fsynSHY|fsynRARG,ShowDevTools},
{L"SetZoomLevel",fsynRSLT|fsynSHY|fsynRARG,SetZoomLevel},
{L"GetZoomLevel",fsynRSLT,GetZoomLevel},
{L"ExecuteJavaScript",fsynRSLT|fsynSHY|fsynRARG,ExecuteJavaScript},
{L"WebSocketClose",fsynRSLT|fsynSHY|fsynRARG,WebSocketClose},
{L"WebSocketSend",fsynRSLT|fsynSHY|fsynRARG,WebSocketSend},
{L"PrintToPDF",fsynRSLT|fsynSHY|fsynRARG,PrintToPDF},
{}
};
```



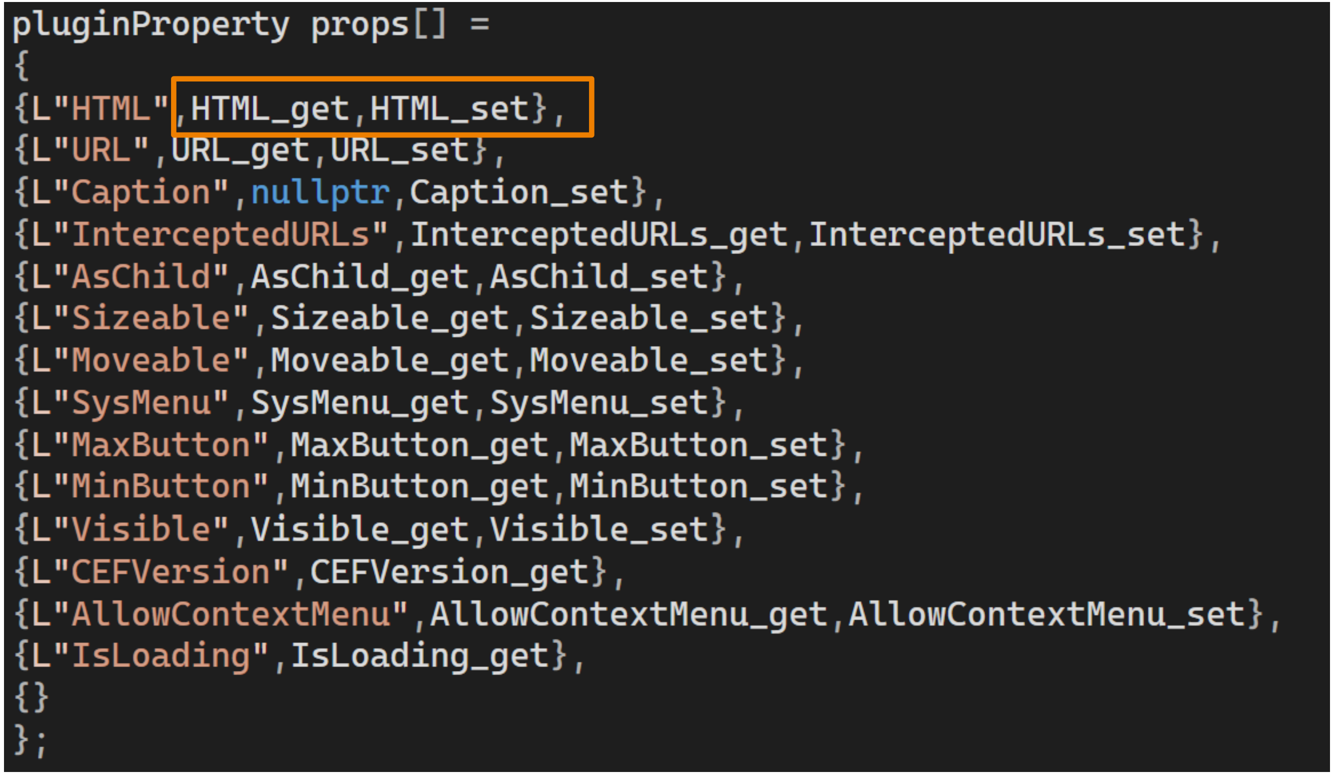
```
pluginEvent events[] =
{
{L"Close",nullptr,Close_complete},
{L"DoPopup",DoPopup_get_args,DoPopup_complete},
{L"LoadEnd",LoadEnd_get_args,LoadEnd_complete},
{L"HTTPRequest",HTTPRequest_get_args,HTTPRequest_complete},
{}
};
```



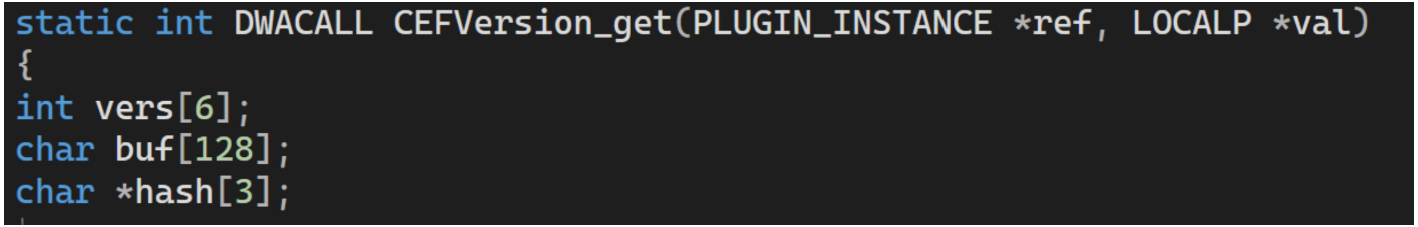
```
pluginEvent events[] =
{
{L"Close",nullptr,Close_complete},
{L"DoPopup",DoPopup_get_args,DoPopup_complete},
{L"LoadEnd",LoadEnd_get_args,LoadEnd_complete},
{L"HTTPRequest",HTTPRequest_get_args,HTTPRequest_complete},
{}
};
```



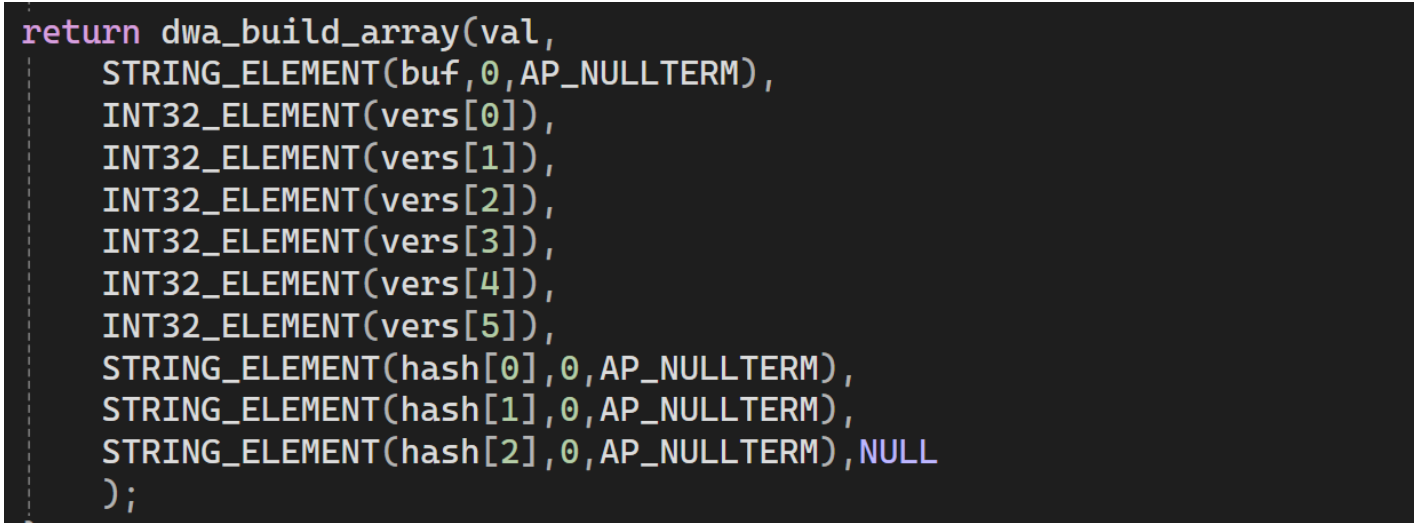
```
pluginProperty props[] =
{
{L"HTML",HTML_get,HTML_set},
{L"URL",URL_get,URL_set},
{L"Caption",nullptr,Caption_set},
{L"InterceptedURLs",InterceptedURLs_get,InterceptedURLs_set},
{L"AsChild",AsChild_get,AsChild_set},
{L"Sizeable",Sizeable_get,Sizeable_set},
{L"Moveable",Moveable_get,Moveable_set},
{L"SysMenu",SysMenu_get,SysMenu_set},
{L"MaxButton",MaxButton_get,MaxButton_set},
{L"MinButton",MinButton_get,MinButton_set},
{L"Visible",Visible_get,Visible_set},
{L"CEFVersion",CEFVersion_get},
{L"AllowContextMenu",AllowContextMenu_get,AllowContextMenu_set},
{L"IsLoading",IsLoading_get},
{}
};
```



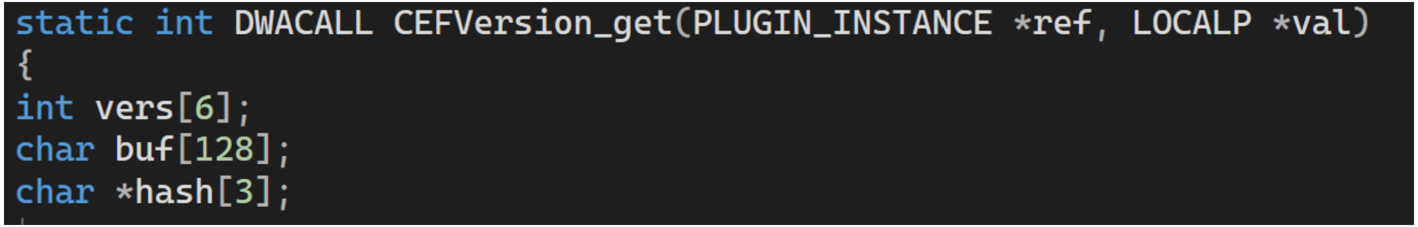
```
pluginProperty props[] =
{
{L"HTML",HTML_get,HTML_set},
{L"URL",URL_get,URL_set},
{L"Caption",nullptr,Caption_set},
{L"InterceptedURLs",InterceptedURLs_get,InterceptedURLs_set},
{L"AsChild",AsChild_get,AsChild_set},
{L"Sizeable",Sizeable_get,Sizeable_set},
{L"Moveable",Moveable_get,Moveable_set},
{L"SysMenu",SysMenu_get,SysMenu_set},
{L"MaxButton",MaxButton_get,MaxButton_set},
{L"MinButton",MinButton_get,MinButton_set},
{L"Visible",Visible_get,Visible_set},
{L"CEFVersion",CEFVersion_get},
{L"AllowContextMenu",AllowContextMenu_get,AllowContextMenu_set},
{L"IsLoading",IsLoading_get},
{}
};
```



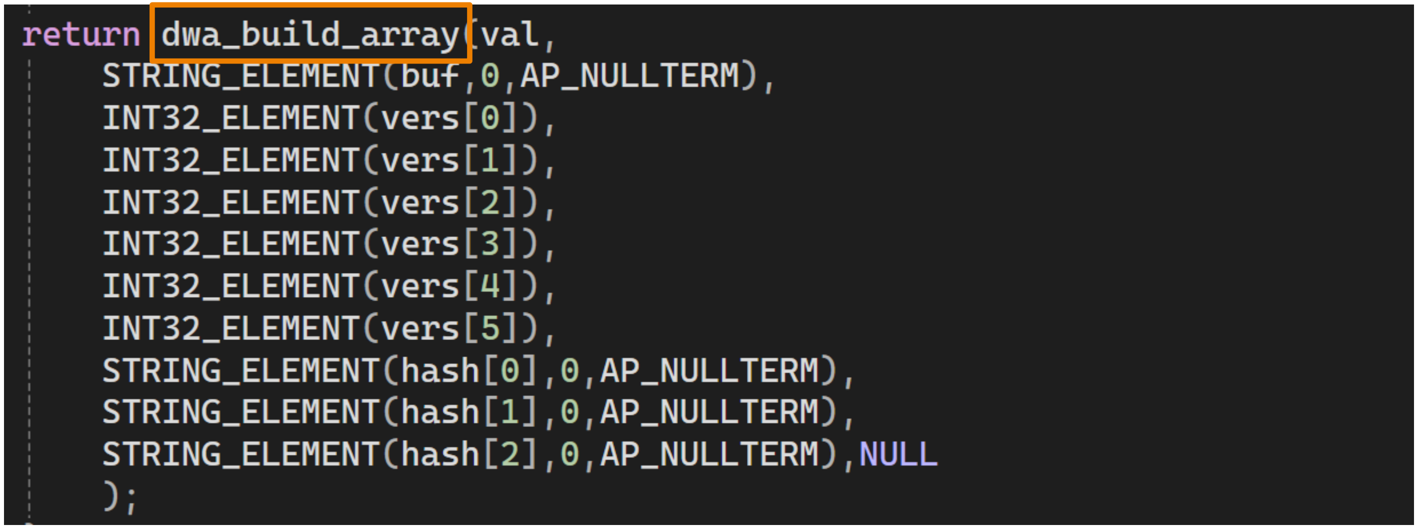
```
static int DWACALL CEFVersion_get(PLUGIN_INSTANCE *ref, LOCALP *val)
{
int vers[6];
char buf[128];
char *hash[3];
```



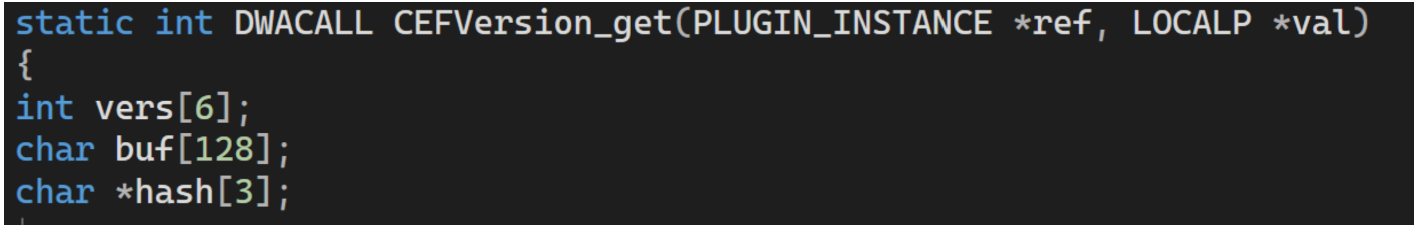
```
return dwa_build_array(val,
    STRING_ELEMENT(buf,0,AP_NULLTERM),
    INT32_ELEMENT(vers[0]),
    INT32_ELEMENT(vers[1]),
    INT32_ELEMENT(vers[2]),
    INT32_ELEMENT(vers[3]),
    INT32_ELEMENT(vers[4]),
    INT32_ELEMENT(vers[5]),
    STRING_ELEMENT(hash[0],0,AP_NULLTERM),
    STRING_ELEMENT(hash[1],0,AP_NULLTERM),
    STRING_ELEMENT(hash[2],0,AP_NULLTERM),NULL
    );
```



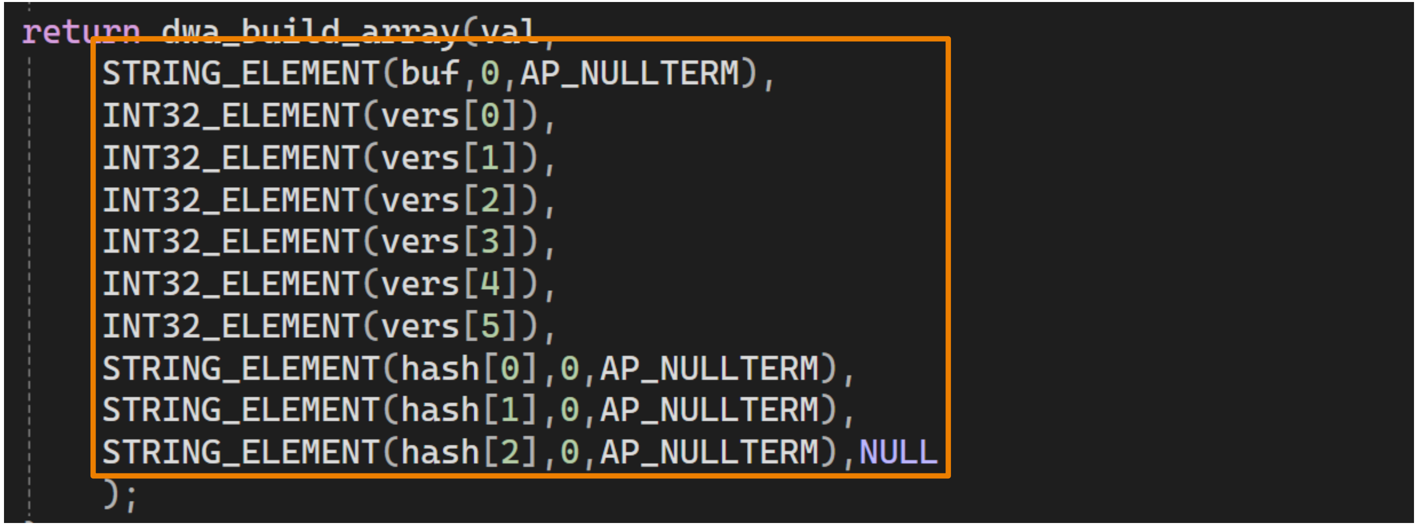
```
static int DWACALL CEFVersion_get(PLUGIN_INSTANCE *ref, LOCALP *val)
{
int vers[6];
char buf[128];
char *hash[3];
```



```
return dwa_build_array(val,
    STRING_ELEMENT(buf,0,AP_NULLTERM),
    INT32_ELEMENT(vers[0]),
    INT32_ELEMENT(vers[1]),
    INT32_ELEMENT(vers[2]),
    INT32_ELEMENT(vers[3]),
    INT32_ELEMENT(vers[4]),
    INT32_ELEMENT(vers[5]),
    STRING_ELEMENT(hash[0],0,AP_NULLTERM),
    STRING_ELEMENT(hash[1],0,AP_NULLTERM),
    STRING_ELEMENT(hash[2],0,AP_NULLTERM),NULL
    );
```



```
static int DWACALL CEFVersion_get(PLUGIN_INSTANCE *ref, LOCALP *val)
{
int vers[6];
char buf[128];
char *hash[3];
```



```
return dwa_build_array(val,
    STRING_ELEMENT(buf,0,AP_NULLTERM),
    INT32_ELEMENT(vers[0]),
    INT32_ELEMENT(vers[1]),
    INT32_ELEMENT(vers[2]),
    INT32_ELEMENT(vers[3]),
    INT32_ELEMENT(vers[4]),
    INT32_ELEMENT(vers[5]),
    STRING_ELEMENT(hash[0],0,AP_NULLTERM),
    STRING_ELEMENT(hash[1],0,AP_NULLTERM),
    STRING_ELEMENT(hash[2],0,AP_NULLTERM),NULL
    );
```

HtmlRenderer

So, I gave it to Bjorn

Conga

Bjørn Christensen

DYALOG

Elsinore 2023

Something about Kafka

Stefan Kruger

Dyalog + Kafka = True? https://www.youtube.com/watch?v=3mlGiCjS78Q

DYALOG

Elsinore 2023

Dyalog + Kafka = True?

Stefan Kruger

> “
>
> APL isn't an island, in order to do our work, we need to talk
> to other systems.
>
> ...
>
> And if we can make Dyalog talk to Kafka that would really open up opportunities to integrate APL Solutions [into] cloud-based data pipelines.

> Apache Kafka is a distributed event store and stream-processing platform. It provides a unified, high-throughput, low-latency platform for handling real-time data feeds.

not just

It's a message queue

So, I gave it to Bjorn

Conga

Bjørn Christensen

So, I gave it to Bjorn

~~Conga~~

Bjørn Christensen

So, I gave it to Bjorn

Kafka

Bjørn Christensen

Demo 3

⎕WC Plugins

- Easy (ish) to write
  - (I mean, let's not kid ourselves)
  - *No* workspace footprint
  - No nasty ⎕NA calls
- Fits in with the existing GUI
  - But doesn’t need it
- You can write a plugin for any platform