sceye-Webscan Principle

the modern way to integrate sceye into any web-driven business application

The limits of our existing SDK regarding browser integration

Since 2008 we offer with our sceye-SDK several tools like ActiveX and .Net-components to integrate sceye into any Windows application. Technically it is possible to use especially the ActiveX component for an embedded solution in a browser, but with some strong limitations.

- Limited support - Microsoft Internet Explorer only
- The browser settings need to allow the execution of ActiveX-components
- which create a potentially security leak.
- no real integration - for the scanned files / harddrive-access needed

Conclusion: The existing SDK is a perfect way to integrate sceye into real applications, but it is not suitable for web-applications.

versus

The modern way to integrate sceye into a browser

With sceye-Webscan we introduce a new way to integrate sceye into a webdriven application. A more powerfull, reliable and future orientated way with a lot of advantages against the old fashioned way.

- Support for all HTML5 compatible browsers
- No changes on browser settings to allow any extended access
- No need for the installation of Browser-Plugins
- Real integration with a clean and save access to the scanned images

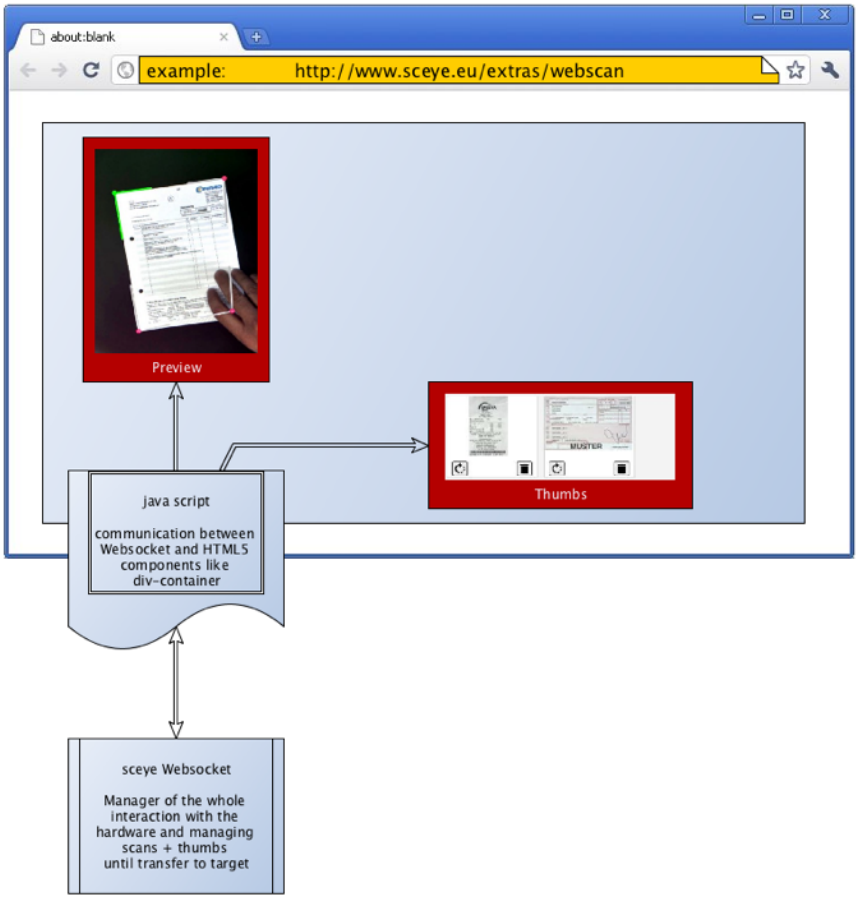
Beside eliminating the disadvantages we even created some more powerfull advantages like:

- Access to all settings + event-based interaction with the scanner
- Managing a set of scans which belong together (Mulitpage documents)
- Simple changes of scans like rotating + deleting supported
- Presenting thumbnails of the scans - High resolution preview of the scans
- PDF-export before uploading the file to the target



- The website can be provided by any webserver from any place.
- The sceye-related components like Preview window, control panel and thumbnails of the scanned pages can be customized to fit perfectly into the individual website.
- The sceye is installed locally - running a tiny local webserver which offers access to the scanner via Websockets. (Minimal cpu and memory usage)
- This way the sceye-content of the website is powered by a localhost-access.
- All communication through the websocket connection is done via JSON objects.
- Scanned images can be uploaded as a multipage PDF document (other formats also possible) to the server / target.
- A clean and save way to integrate a scanner into the browser and so into any web-driven business application.

Silvercreations AG Campusallee 9928 55768 Neubrücke **T** +49 6782 172738 **F** 99 e.rietzke@silvercreations.de www.silvercreations.de



sceye-Webscan allows a free designable way to integrate a Live-Preview component and a thumbnail component by using the style of an existing web-application.
With a div-container these elements can be placed everywhere in the website.
With a provided Javascript-class these container can be addressed and it's context can be updated - all driven by the local Websocket of sceye-Webscan.
If wished all scans can appear as Thumbnails and can be rotated or deleted anytime.
With a click at a Thumbnails the scan will show up in a Lightbox with highest resolution.
Once all pages of a document are scanned, sceye-Webscan can provide access to the native images or can export the images to a single tif or pdf file - ready for transfer.

Please visit our website to get a first example of such an integration.
http://www.sceye.eu/de/extras/webscan

Requirements for sceye-Webscan:

- A sceyeX model (5th generation)
- Windows OS (Vista, Win7, Win8) (32 or 64 bit)
- Basic software installation (version 5.13 or higher)
- Sceye-Webscan installation
- HTML5 compatible browser
- Javascript support
- Websocket support (http://caniuse.com/#feat=websockets)

For background information on websocket technology take a look at:

- https://en.wikipedia.org/wiki/WebSocket
- https://www.websocket.org/

Silvercreations AG Campusallee 9928 55768 Neubrücke **T** +49 6782 172738 **F** 99 e.rietzke@silvercreations.de www.silvercreations.de

How to use the websocket interface?

The following lines of sample code will give you a short impression how easy it is to integrate sceye-Webscan right into your own Web-driven business application.

```
SceyeGuiObj = new SceyeGui;          //The SceyeGui class handles the connection between the websocket part and your GUI part.
SceyeObj = new SceyeApi;             //The SceyeObj class is the SDK part (which we will provide) to make the communication more easily.
window.onload = function (evt) {
   SceyeObj.Init();
};
window.onclose = function (evt) {
   SceyeObj.Close();
};
window.onbeforeunload = function (evt) {
   SceyeObj.Close();
};
```

You can change settings via this type of calls: SceyeObj.SendCommand("Sceye.Options.Toggle." + OptionID);

For example:

```
Sceye.Options.Toggle.Colormode     // this call will switch to the next colormode.
Sceye.Options.Toggle.Laser         // this call will switch between Laser on and Laser off.
Sceye.Options.Toggle.Light         // this call will switch between Light on and Light off.
Sceye.Options.Contrast.Set.80      // this call will set the contrast value to 80.
```

The Scan is started with a click on a button which calls the API command. In your HTML code this could look like

```
<button class="sceye_buttons sceye_button_scan" id="sceye_scan_button" onclick=„SceyeObj.SendCommand('Sceye.Scan');">Scan</button>
```

This call is non-blocking. When the Sceye captured the image it will notify you via the API

```
   ws.onmessage = function (evt){
      if(IsObjectValid(evt)){
         if(IsObjectValid(evt.data)){
            var obj = JSON.parse(evt.data);

{... code deleted ...}

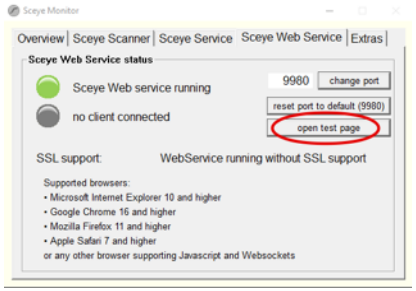
            if(obj['response'] === 'Sceye.Scan'){
               if(obj['result'] === "ok"){
                  SceyeGuiObj.AddThumbToThumbnailView(obj);
                        }
            }
```

Here the JSON object holds not only the result code but also the preview of the scanned image.
Scans are stored in the webserver until you start the save command.
When you perform the save command the server will collect the scanned images and build a pdf file or a tif file or will send you each image as individual jpg image (configurable).
The API will receive an event (similar to the above code) for each page (or the complete document in case of pdf or tif). The image data is sent as a base64-encoded string (containing the file) to your application.
With javascript you can send this string further to your host or do some other stuff within javascript.

How to get started?

Please install the following software on your computer:

- Sceye Software 5.28 installer and Sceye Web Service 1.6.1 installer: http://www.sceye.eu/extern/SceyeWebService_1_6_1_CompleteInstallerPackage.zip
- Installation instructions: http://www.sceye.eu/extern/Sceye_Web_Service_Version_1_6_1_Installation_instructions.pdf
- sceye Webscan API documentation: http://www.sceye.eu/extern/SceyeWebService_Interface_description_Version_1.6.0.pdf



Start sceye-Webscan as a demo in your browser.
The easiest way is by using the sceye Monitor (System-Tray) and pushing the „open test page“ button.

The Website is demonstrating the basic features of sceye-Webscan. For a deeper look into this example, please open the according index.html page under C:\Program Files (x86)\Silvercreations\SceyeWebService\www\

The Website is taking advantage of the SceyeApi.js file which is located under the following folder:
C:\Program Files (x86)\Silvercreations\SceyeWebService\www\include