3.– 6. September 2012
in Nürnberg

Herbstcampus

Wissenstransfer
par excellence

Programming Windows Store Apps with JavaScript and WinRT

Rainer Stropek

software architects gmbh

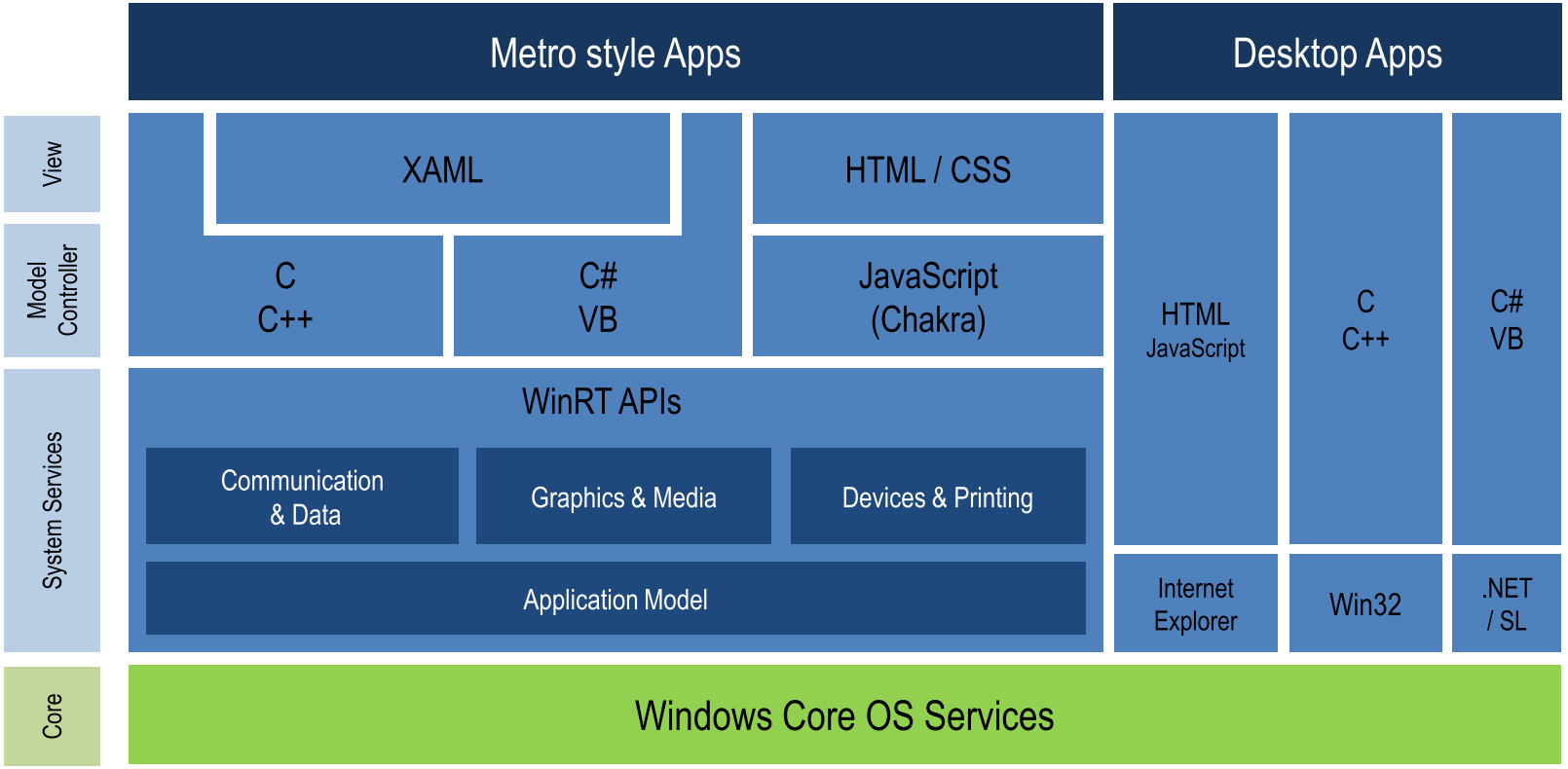
Introduction

- software architects gmbh
- Rainer Stropek
  - Developer, Speaker, Trainer
  - MVP for Windows Azure
  - rainer@timecockpit.com
  - @rstropek

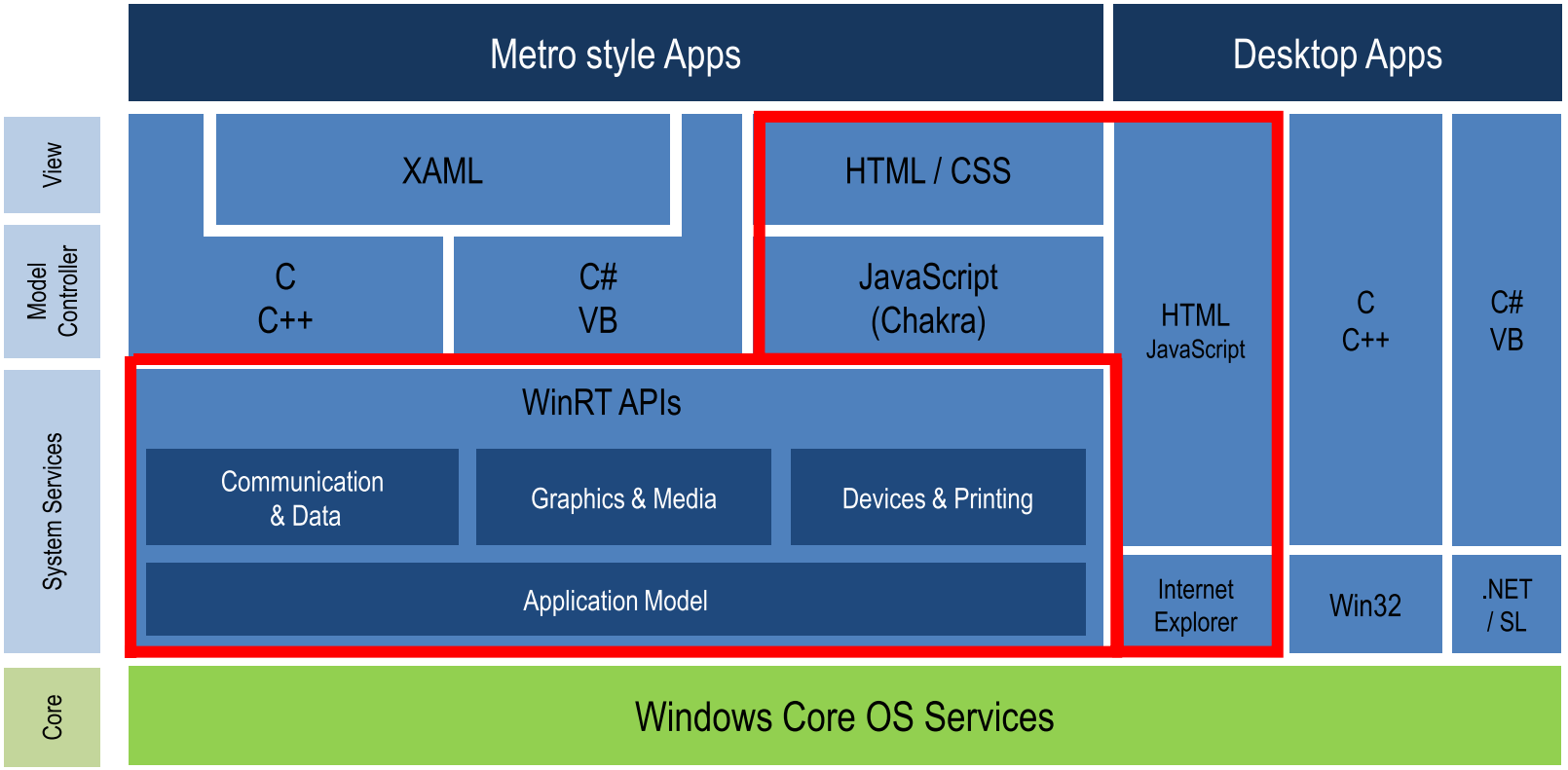
time cockpit

http://www.timecockpit.com

WinRT System Architecture



WinRT System Architecture



Windows Store Apps in JavaScript?

Goals…

- Reuse existing knowledge from the web for developing Windows applications
- HTML + JavaScript are 1st class citizens for developing Windows Store Apps
- "Fast and Fluid" also with HTML + JavaScript

Prerequisites…

- Use all the technologies you know from IE 10: HTML5 + CSS3 + JavaScript
- Full access to platform API (WinRT)
- Hardware Acceleration, Plugin-Free, etc.

Goals of This Session

- Build a Windows Store App from scratch using HTML + JavaScript
- Introduce the WinJS Library
- Access API of Windows 8 (WinRT) from JavaScript
- One integrated demo used throughout the entire session

Non-Goals of the Session:

- General HTML/CSS/JavaScript introduction
- Training about WinJS, WinRT or Windows Store Apps

Tip: Download slides and check hidden slides for code snippets

Important Notice

- **For demo purposes** the sample has been simplified compared to a real-world Windows Store app
- In practise you have to remember additional requirements in order to get your app into the Windows Store, e.g.:
  - Handle different screen resolutions and view modes (e.g. portrait/landscape, snapped, etc.; read more)
  - Support navigation patterns (e.g. links, back/forward, etc.; read more)
  - App Lifecycle (e.g. launch, activation, suspend, etc.; read more)

JavaScript Module Pattern

```
(function ( ) {
    // create variable
    // var name = "Andi und Rainer";

    // create method
    // var sayHello = function ( ) {
    //    alert(name);
    // };
})( ); // calling this method.
```

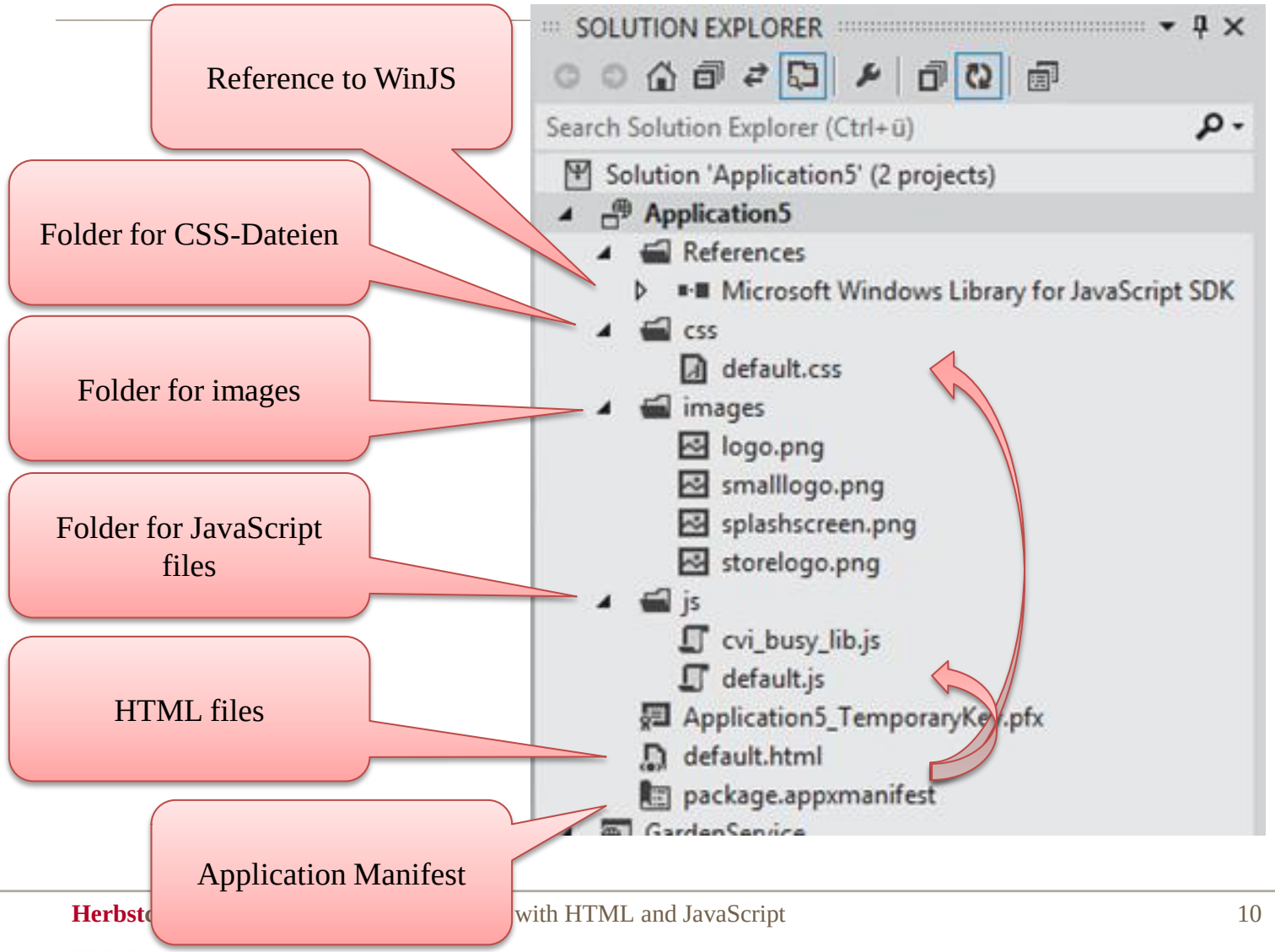
It is plain old JavaScript, also for Windows Store Apps!

Demo

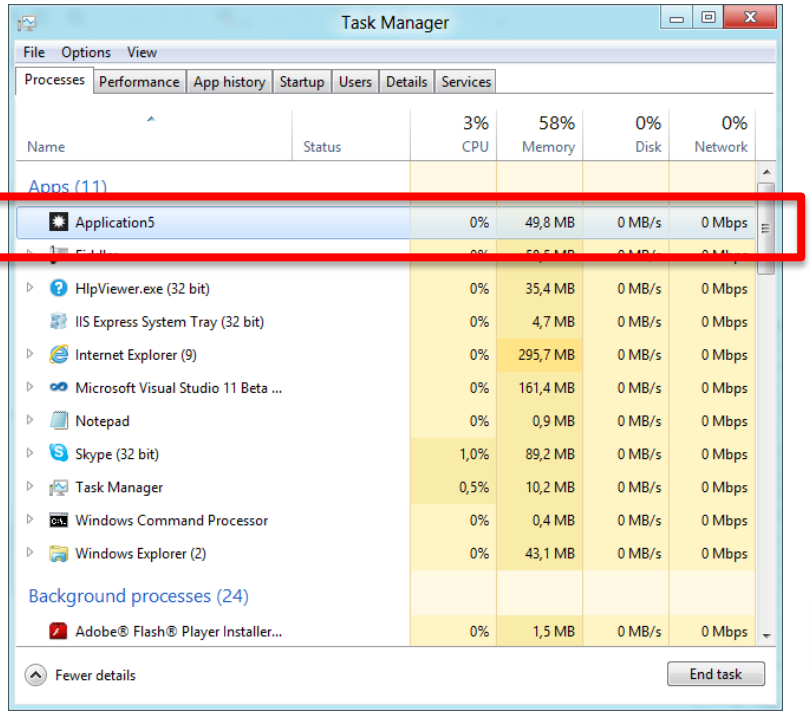
- Structure of a JavaScript Windows Store App
- Runtime environment for JavaScript Apps

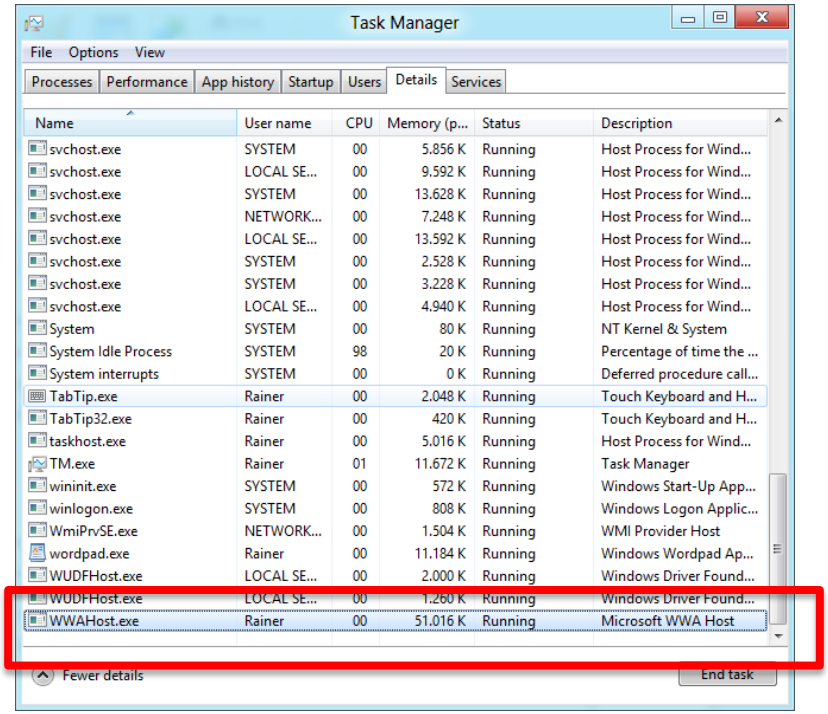
Background information in MSDN

Structure of a JavaScript App



Structure of a JavaScript App





Demo

- Controls and Data Binding

ListView, a WinJS Control

```
<!-- add a WinJS control in HTML -->
<div class="itemlist" data-win-control="WinJS.UI.ListView" […]></div>


// find control in DOM tree
var lv = document.body.querySelector(".itemlist");

// fill WinJS list (=data source for ListView)
var items = new WinJS.Binding.List();
for (var i = 1000; i < 1500; i++) {
    items.push("Hello World!");
}

WinJS.UI.processAll().then(function() {
    lv.winControl.itemDataSource = items.dataSource;
}
```

Demo

- Templates and Data Binding

ListView, a WinJS Control

```
<div class="splitpage">
     <div class="itemtemplate" data-win-control="WinJS.Binding.Template">
          <div><img src="#" data-win-bind="src: ImageUri; alt: Title" style="width: 100px" /></div>
          <div class="item-info"><h3 class="item-title win-type-ellipsis" data-win-bind="textContent: Title"></h3></div>
     </div>
     <header aria-label="Header content" role="banner"><h1>Garden Blog</h1></header>
     <section class="itemlistsection" aria-label="List section">
          <div class="itemlist" aria-label="List of this group's items" data-win-control="WinJS.UI.ListView"
               data-win-options="{ selectionMode: 'single', tapBehavior: 'directSelect' }">
          </div>
     </section>
     <section class="articlesection" aria-atomic="true" aria-label="Item detail" aria-live="assertive">
          <header class="header">
               <div class="text"><h2 data-win-bind="textContent: Title"></h2></div>
               <div>
                    <img class="article-image" src="#" data-win-bind="src: ImageUri; alt: Title"
                         style="width: 150px; max-height: 150px" />
               </div>
          </header>
          <article class="article-content" data-win-bind="innerHTML: Text"></article>
     </section>
</div>
```

ListView, a WinJS Control

```
function onselectionchanged(eventObject) {
    var listView = document.body.querySelector(".itemlist").winControl;
    // By default, the selection is restriced to a single item.
    listView.selection.getItems().then(function (items) {
        if (items.length > 0 && items[0]) {
            var details = document.querySelector(".articlesection");
            WinJS.Binding.processAll(details, items[0].data);
            details.scrollTop = 0;
        }
    })
}
```

Root node for data binding

Data Context for data binding

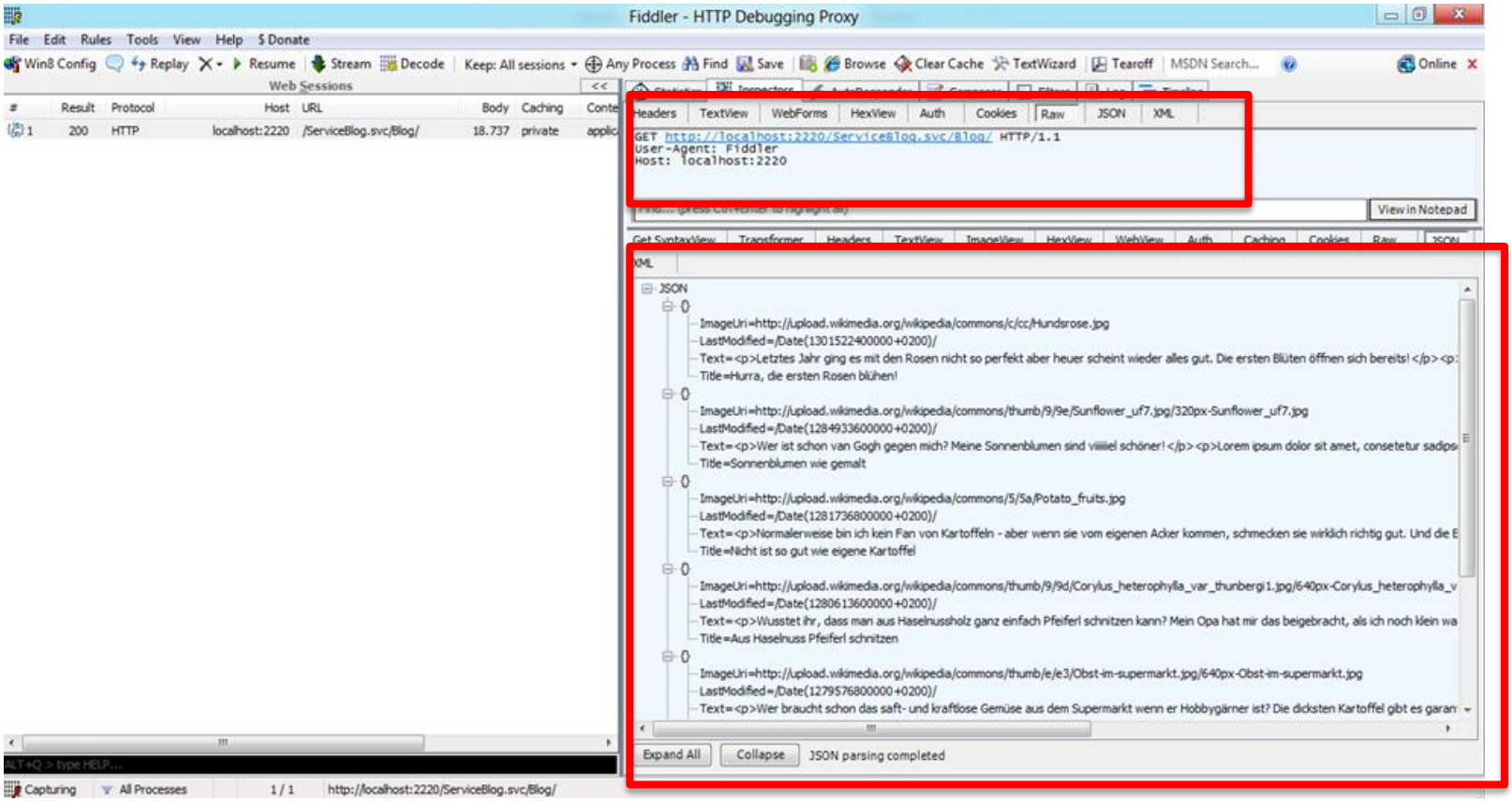
ListView, a WinJS Control

```
WinJS.UI.setOptions(lv.winControl, {
    itemTemplate: document.body.querySelector(".itemtemplate"),
    onselectionchanged: onselectionchanged.bind(this),
    itemDataSource: items.dataSource,
    layout: new WinJS.UI.ListLayout()
});
lv.winControl.selection.set(0);
```

Demo

- Working with REST Services

REST Service in Fiddler



WinJS.xhr

```
function refreshBlogList() {
    WinJS.xhr({ url: "http://localhost:2220/ServiceBlog.svc/Blog/" }).done(function (feedResponse) {
        // convert feed to list of bindable objects
        var feed = JSON.parse(feedResponse.responseText);
        var items = new WinJS.Binding.List(feed);

        // set properties of list view (including data source for binding)
        var lv = document.body.querySelector(".itemlist");
        WinJS.UI.setOptions(lv.winControl, {
            itemTemplate: document.body.querySelector(".itemtemplate"),
            onselectionchanged: onselectionchanged.bind(this),
            itemDataSource: items.dataSource,
            layout: new WinJS.UI.ListLayout()
        });

        // automatically select first item
        lv.winControl.selection.set(0);
    });
}
```

WinJS.xhr

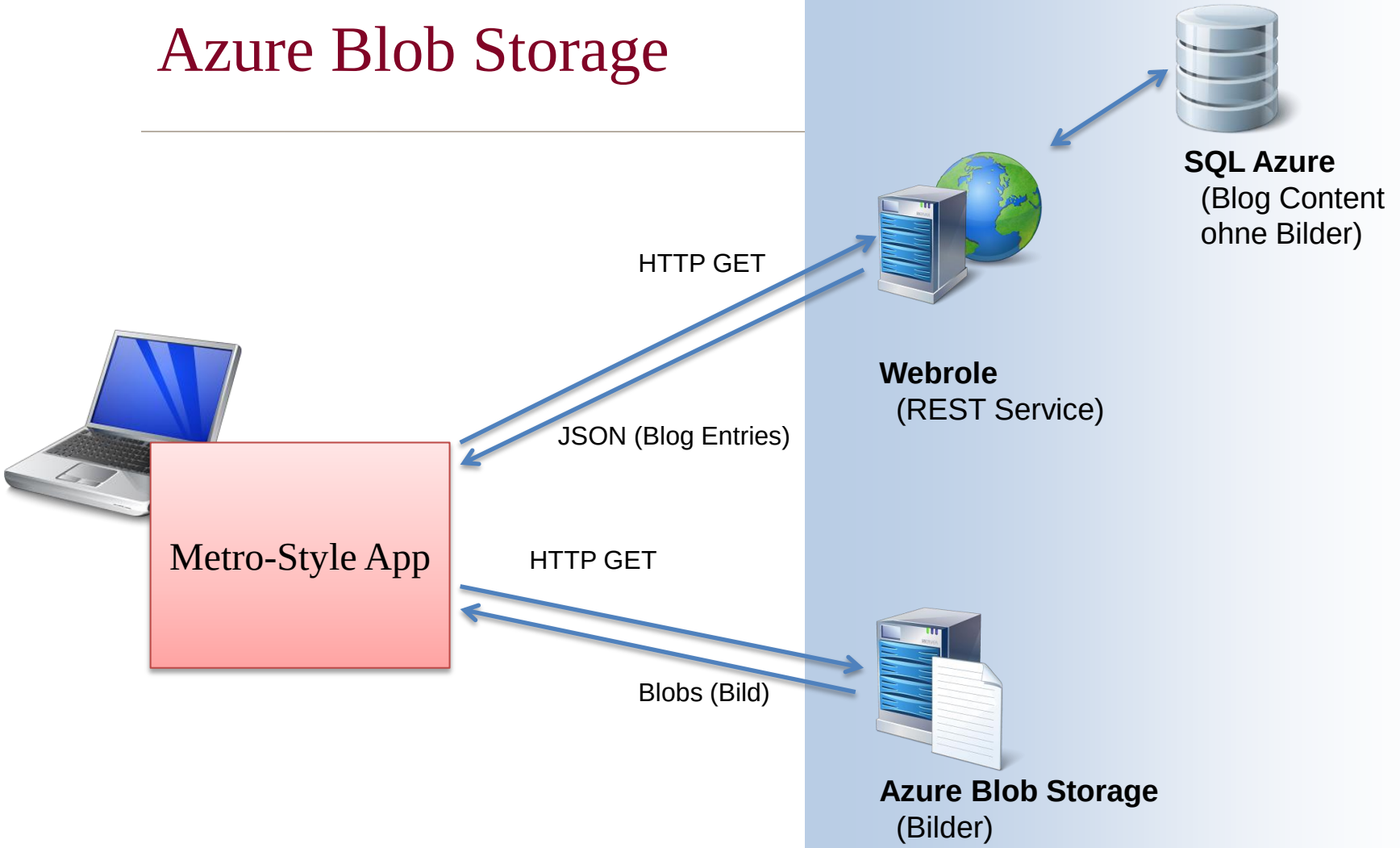
```
app.onactivated = function (eventObject) {
    // lookup list view in DOM
    var lv = document.body.querySelector(".itemlist");

    WinJS.UI.processAll().then(function () {
        refreshBlogList();
    });
}
```

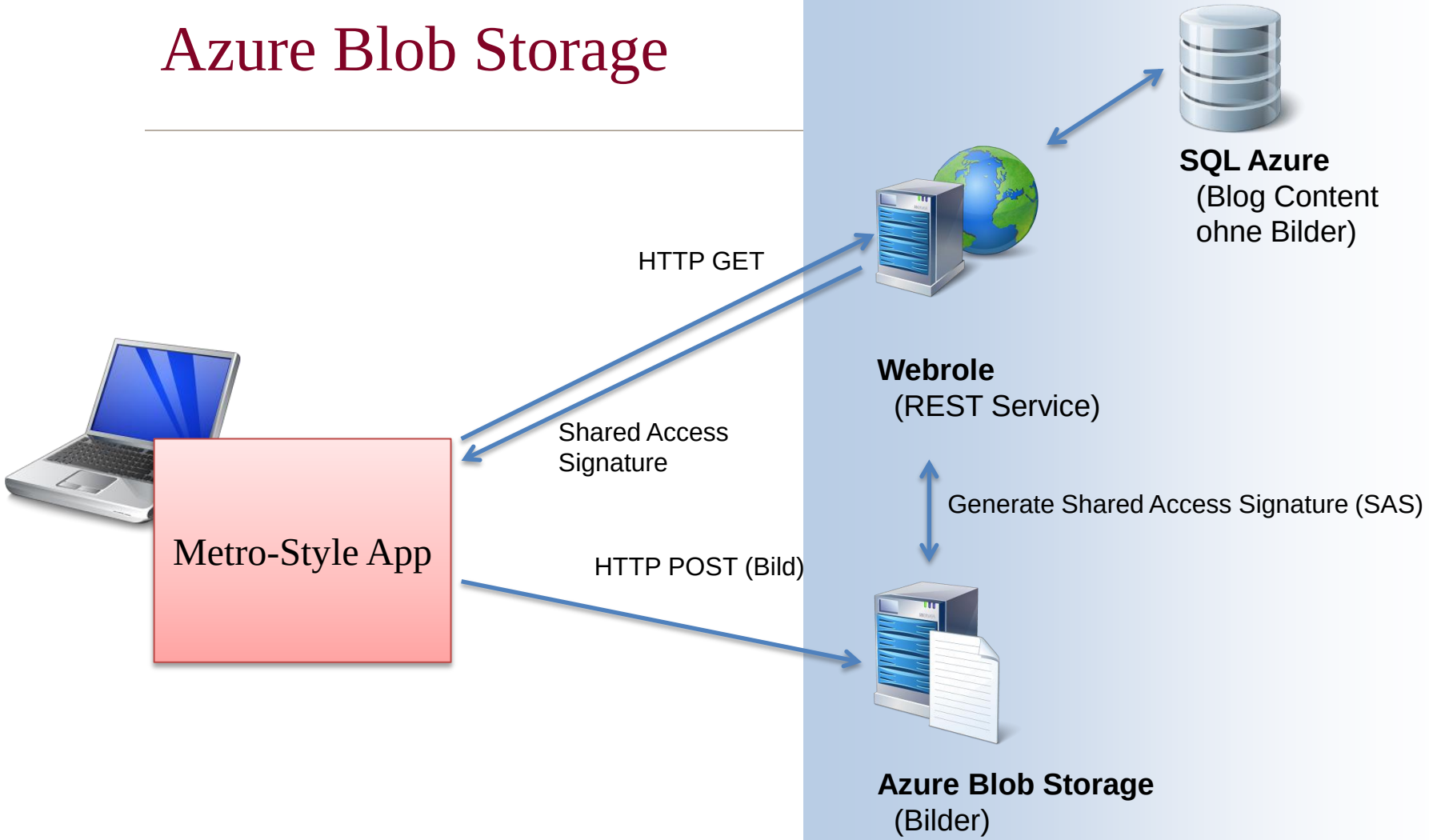
Demo

- AppBar, XHR and Azure

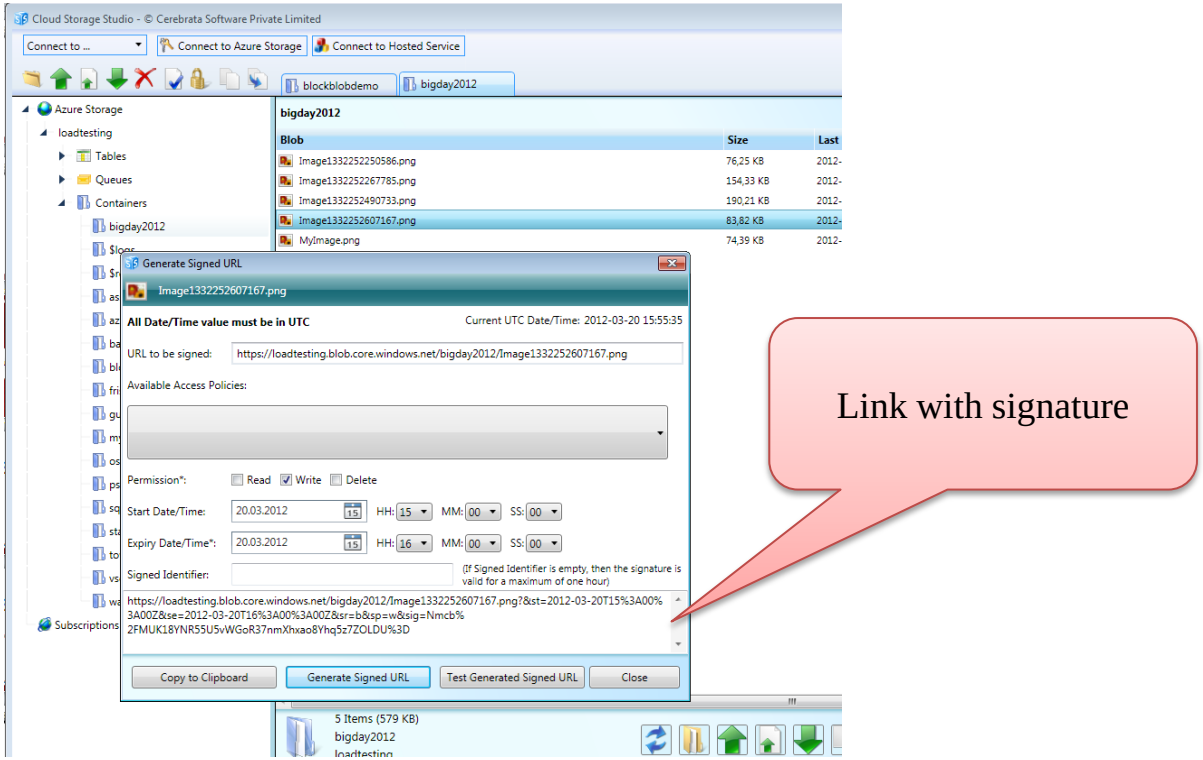
Azure Blob Storage



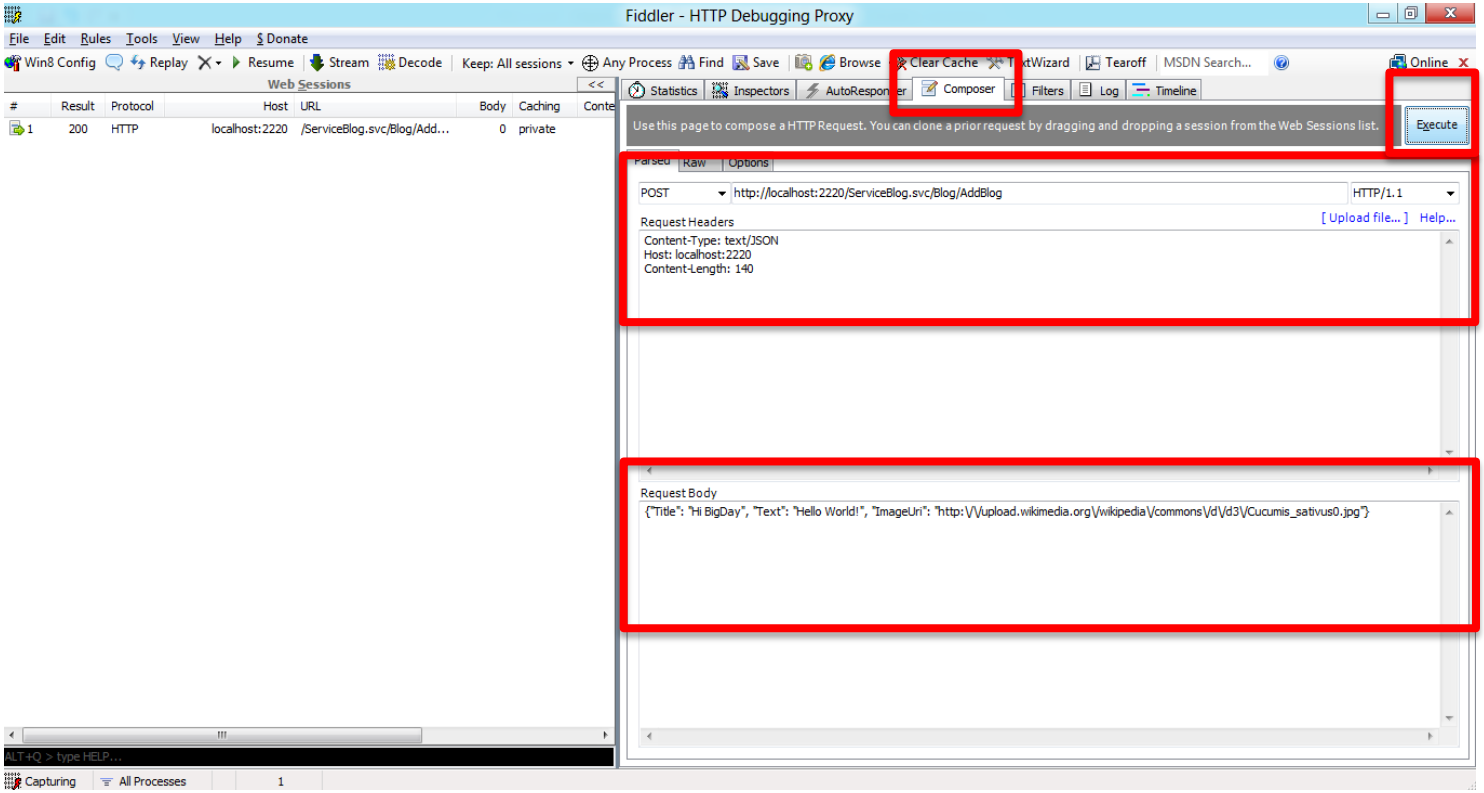
Azure Blob Storage



Azure Shared Access Signatures



INSERT with REST Service



Adding an AppBar

```
<div id="appbar" data-win-control="WinJS.UI.AppBar">
    <button data-win-control="WinJS.UI.AppBarCommand"
        data-win-options="{id:'cmd', label:'Create Entry', icon:'placeholder'}">
    </button>
    <button data-win-control="WinJS.UI.AppBarCommand"
        data-win-options="{id:'refresh', label:'Refresh', icon:'placeholder'}">
    </button>
</div>

// add button handler for appbar button
var cmdButton = document.getElementById("cmd");
cmdButton.onclick = onPostBlogEntryClick;
var refreshButton = document.getElementById("refresh");
refreshButton.onclick = refreshBlogList;
```

Using WinRT to Access the Webcam

```
function onPostBlogEntryClick() {
    // let user take a picture. Note that we use WinRT classes here.
    var capture = Windows.Media.Capture;
    var cameraCaptureUI = new capture.CameraCaptureUI();

cameraCaptureUI.captureFileAsync(capture.CameraCaptureUIMode.photo).then(
        function (capturedItem) {
            if (capturedItem) {
                […]
            }
        });
}
```

Using WinRT to Access the Webcam

```
if (capturedItem) {
      var rootDiv = document.body.querySelector(".itemlist");
      var busy = getBusyOverlay(rootDiv, { color: 'gray', opacity: 0.25 }, { type: 'oval', color: '#FFFFFF' });

      WinJS.xhr({ url: "http://localhost:2220/ServiceBlog.svc/Blog/GetSasForImage/" + imageName })
      .done(function (result) {
            var uploader = new Windows.Networking.BackgroundTransfer.BackgroundUploader();
            uploader.method = "PUT";
            uploader.setSourceFile(capturedItem);
            var uri = new Windows.Foundation.Uri(JSON.parse(result.responseText));
            var upload = uploader.createUpload(uri);
            upload.startAsync().done(function (result) {
                  WinJS.xhr({ url: "http://localhost:2220/ServiceBlog.svc/Blog/AddBlog", data: JSON.stringify(newEntry),
                        type: "POST", headers: { "Content-Type": "text/JSON" } }).done(function () {
                        busy.remove();
                        refreshBlogList();
                  });
            },
            function (err) {
                  busy.remove();
            });
      });
}
```

Demo

- Share Contract

Implement the Share Contract

```
var dtm = Windows.ApplicationModel.DataTransfer.DataTransferManager.getForCurrentView();
dtm.addEventListener("datarequested", function (e) {
        // capture request member
        var request = e.request;

        // get selected item from list view
        var lv = document.body.querySelector(".itemlist").winControl;
        lv.selection.getItems().then(function (items) {
                if (items.length > 0) {
                        var item = items[0].data;

                        // set request title and description
                        request.data.properties.title = item.Title;
                        request.data.properties.description = "Garden Blog";

                        // set text content
                        var recipe = item.Text;
                        request.data.setText(recipe);

                        // set image content
                        // note that we use deferral-API here.
                        // (only do this if your operation is quite fast; 200ms time limit or so)
                        var uri = new Windows.Foundation.Uri(item.ImageUri);
                        var reference = Windows.Storage.Streams.RandomAccessStreamReference.createFromUri(uri);
                        request.data.properties.thumbnail = reference;
                        var deferral = request.getDeferral();
                        request.data.setBitmap(reference);
                        deferral.complete();
                }
        });
});
```

Summary

What you have seen…

- Structure of a Windows Store App with HTML + JavaScript
- Introduction into WinJS
- Accessing Windows 8 API (WinRT) from JavaScript

Use your existing knowlege about HTML and JavaScript for Windows Store Apps

Further readings:

- http://dev.windows.com
- http://design.windows.com

Vielen Dank!

Rainer Stropek
software architects gmbh

time cockpit is the leading time tracking solution for knowledge workers. Graphical time tracking calendar, automatic tracking of your work using signal trackers, high level of extensibility and customizability, full support to work offline, and SaaS deployment model make it the optimal choice especially in the IT consulting business.

Try time cockpit for free and without any risk. You can get your trial account at http://www.timecockpit.com. After the trial period you can use time cockpit for only 0,20€ per user and month without a minimal subscription time and without a minimal number of users.

time cockpit ist die führende Projektzeiterfassung für Knowledge Worker. Grafischer Zeitbuchungskalender, automatische Tätigkeitsaufzeichnung über Signal Tracker, umfassende Erweiterbarkeit und Anpassbarkeit, volle Offlinefähigkeit und einfachste Verwendung durch SaaS machen es zur Optimalen Lösung auch speziell im IT-Umfeld.

Probieren Sie time cockpit kostenlos und ohne Risko einfach aus. Einen Testzugang erhalten Sie unter http://www.timecockpit.com. Danach nutzen Sie time cockpit um nur 0,20€ pro Benutzer und Tag ohne Mindestdauer und ohne Mindestbenutzeranzahl.