

FS18

WEB ENGINEERING & DESIGN 3

Introduction

T. Bocek

thomas.bocek@hsr.ch

Slides partially by Silvan Gehrig

Learning Objectives

The learning objective of this lecture is to acquire knowledge and expertise in single page applications (SPA) and to be able to understand its implementation.

The students are able to...

- ... **differentiate between a traditional web application and an SPA**
- ... **explain key concepts of an SPA**
- ... **to write a simple Vue.js application**

- **In the News**
- **Introduction**
 - Traditional Web vs. SPA
- **Build a Non-SPA application with Spring Boot**
 - Setup the environment
- **Build an SPA with Spring Boot**
- **Introduction to Vue.js**
 - Hello World
 - Input / Form Binding
 - Lists
 - Components

URL: b.socrative.com/student/

Room: HSR

Single Page Application (SPA) - In the News

■ 04.02.2018: Single Page Application Is Not a Silver Bullet

- Today not a question: “to SPA or not to SPA”
- Rather: “which client-side framework should we use for our SPA”
- Migrated from WordPress to React/Redux/Immutable.js (data in an Immutable.js object is never mutated, a new copy is always returned)
 - Page load time increased 2x
- Application became huge ~650KB zipped
 - Download, unzip
 - Parse 2.6MB, execute
- Tests with Airbnb, Twitter, Medium
 - Site need 2-3 sec. to become usable

■ Pros

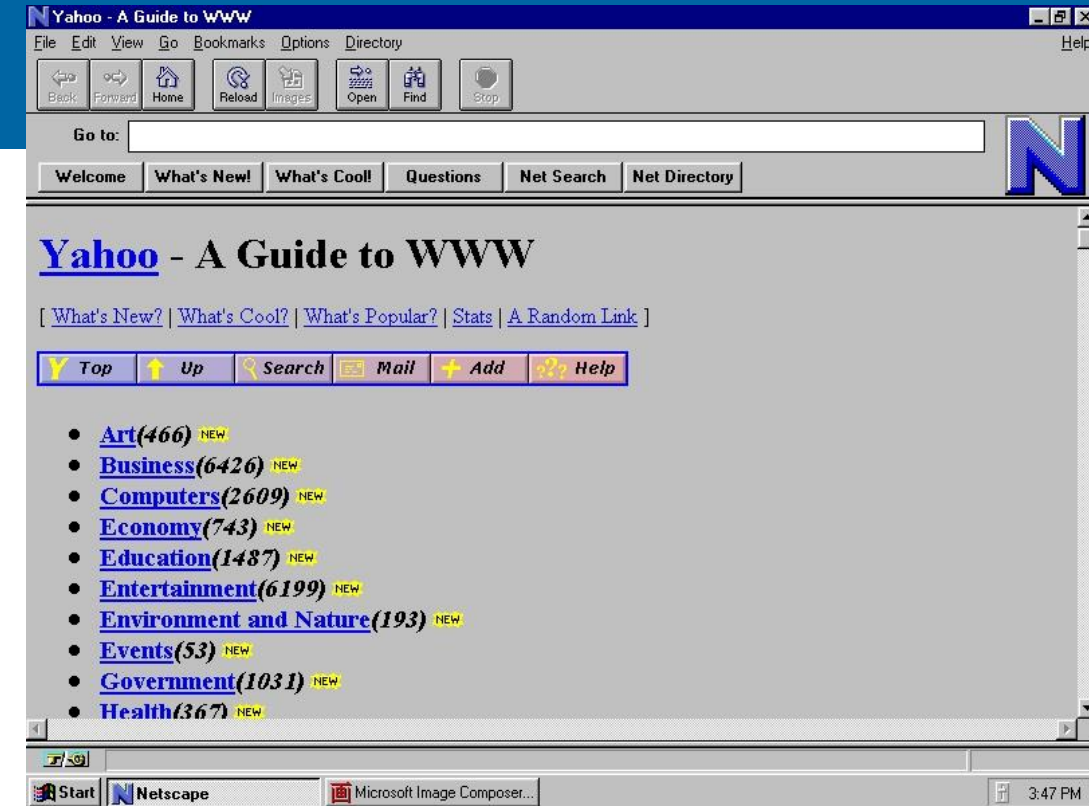
- No page reloads after clicking
- Granular actions are fast
- Development of complicated features easier (especially when we need to interact between screens)
- Possible to create almost “native” experience, with fallbacks if no internet is there - PWA

■ Cons

- Broken “back”, “open in a new tab” button
- Broken “refresh”
- Slow?

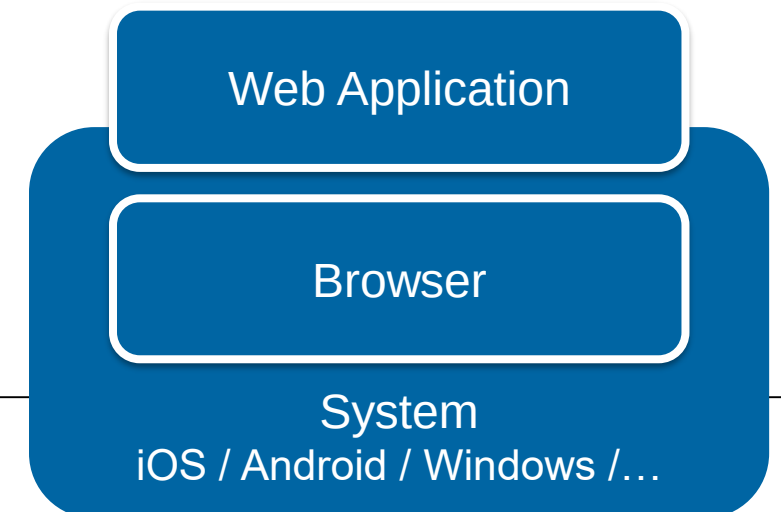
Introduction – Preface

- Browsers are getting more powerful ([history](#))
 - Nearly every device accommodates a browser
 - ...more and more hardware access
 - Camera
 - Geo-Information
 - ...and even more
 - <https://developer.mozilla.org/en-US/docs/WebAPI>
 - <http://mobilehtml5.org>, <https://caniuse.com>



<https://www.w3.org/2010/Talks/0921-html5-plh/web10.html>

***Today, browsers provide the required feature-set
and performance to run sophisticated, web-based applications***



Introduction – Browser-based Applications

■ Benefits

- No need for a client-backup
- Platform independent
- No software update
- Software can be provided as a Services (SaaS)
 - «pay as you go» (charges based on usage)

■ Liabilities

- No data sovereignty
- Limited/restricted hardware access
- No operation system access
- Runs everywhere
- ([SEO compatibility](#))

Single Page Application (SPA)

■ Website

- Interacts with user by rewriting parts of the DOM
- No need to load entire page from server
- Example non-SPA: <https://hsr.ch>
- Example SPA: <https://webmail.hsr.ch>

■ SPA moves logic from the server to the client

- Server typically data API service
- "Thin Server Architecture"

■ Lack of JavaScript execution on crawlers

- Content may not be indexed – email no issue, documentation pages big issue

The image displays two browser windows side-by-side, illustrating the difference between a standard website and a Single Page Application (SPA).

Top Window (hsr.ch): Shows the official website of Hochschule für Technik Rapperswil (HSR). The network tab is open, showing a list of requests including HTML, CSS, JavaScript, and images. The total load time is 1.92 s.

Bottom Window (webmail.hsr.ch): Shows the webmail interface. The network tab is open, showing requests for JSON data. The total load time is 26 ms.

Network Tab Data (Top Window):

Status	Method	File	Domain	Cause	Type	Transferred	Size	0 ms	640 ms	1.28 s	1.92 s
200	GET	/de/die-hsr/campus/	www.hsr.ch	document	html	16.36 KB	77.69 KB	→ 90 ms			
200	GET	merged-c85a94e177552c8014f0a5b8fc2d...	www.hsr.ch	stylesheet	css	13.21 KB	85.87 KB	→ 17 ms			
200	GET	008bda551e-e0048c4c6ec9501d06d0824...	www.hsr.ch	stylesheet	css	662 B	290 B	→ 42 ms			
200	GET	merged-af18a380abca34dc3766d160df62...	www.hsr.ch	stylesheet	css	12.23 KB	61.69 KB	→ 46 ms			
200	GET	modernizr.custom.64146.js	www.hsr.ch	script	js	2.03 KB	3.21 KB	→ 31 ms			
200	GET	jquery-2.1.4.min.js	www.hsr.ch	script	js	29.40 KB	82.44 KB	→ 64 ms			
200	GET	merged-932ad5a5dd7026d3a284e470483...	www.hsr.ch	script	js	93.16 KB	300.68 KB	→ 78 ms			
200	GET	merged-cb688c0c3737fa85500bb6ec9687...	www.hsr.ch	script	js	940 B	930 B	→ 54 ms			
200	GET	merged-64627f20f892b5f14d6b26ca743e...	www.hsr.ch	script	js	5.24 KB	20.50 KB	→ 48 ms			
200	GET	1.css?apiType=css&c=9afd8026-e775-46...	fast.fonts.net	stylesheet	css	348 B	0 B	→ 109 ms			
200	GET	modernizr.custom.64146.js	www.hsr.ch	script	js	2.03 KB	3.21 KB	→ 16 ms			
200	GET	logo_hsr.svg	www.hsr.ch	img	svg	5.63 KB	16.29 KB	→ 13 ms			

Network Tab Data (Bottom Window):

Status	Method	File	Domain	Cause	Type	Transferred	Size	0 ms	640 ms	1.28 s	1.92 s
200	POST	service.svc?action=GetFolder&EP=1&...	webmail.hsr.ch	xhr	json	1.81 KB	549 B	→ 29 ms			
200	POST	service.svc?action=UpdateUserConfig...	webmail.hsr.ch	xhr	json	1.60 KB	232 B				→ 26 ms

URL: b.socrative.com/student/

Room: HSR

■ A Single Page Application

...is a web application or web site that interacts with the user by dynamically rewriting the current page rather than loading entire new pages from a server.

In an SPA, either all necessary code – HTML, JavaScript, and CSS – is retrieved with a single page load, or the appropriate resources are dynamically loaded and added to the page as necessary, ...

Source: https://en.wikipedia.org/wiki/Single-page_application

■ Simple non-SPA example

- Spring Boot / [IntelliJ](#)

■ Create new project

- build.gradle

■ Add Spring 2.0.0RC1

```
buildscript {
    repositories {
        maven { url 'https://repo.spring.io/libs-milestone' }
    }
    dependencies {
        classpath 'org.springframework.boot:spring-boot-gradle-plugin:2.0.0.RC1'
    }
}
apply plugin: 'org.springframework.boot'
apply plugin: 'java'
version '1.0-SNAPSHOT'
sourceCompatibility = 1.8

repositories {
    mavenCentral()
    maven { url 'https://repo.spring.io/libs-milestone' }
}

bootJar {
    launchScript()
}

dependencies {
    compile('org.springframework.boot:spring-boot-starter-web:2.0.0.RC1')
    compile('org.springframework.boot:spring-boot-starter-freemarker:2.0.0.RC1')
}
```

URL: b.socrative.com/student/

Room: HSR

■ Setup Controller

- Application.java
- NoSpa.java
- Add freemarker template templates/nospa.ftl

■ Server renders the HTML

- Fills in the variables
- Sends HTML to browser

■ Build executable jar

- gradle build

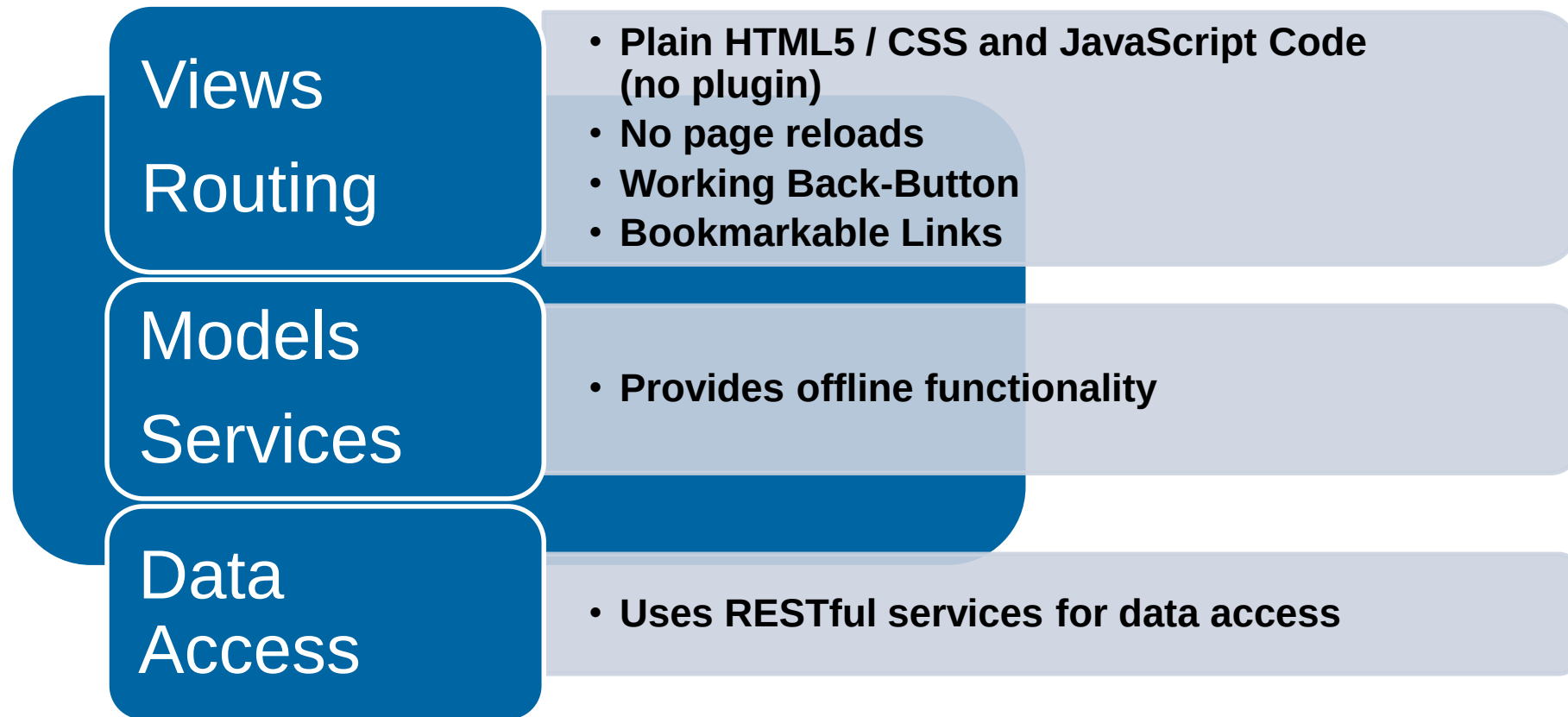
```
@SpringBootApplication
public class Application {
    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }
}
```

```
@Controller
public class NoSpa {
    @GetMapping("/nospa/{value}")
    public String nospa(Model model, @PathVariable("value") String value) {
        model.addAttribute("name", value.toUpperCase());
        return "nospa";
    }
}
```

```
<!DOCTYPE html>
<head>
    <meta charset="UTF-8">
    <title>Hello!</title>
</head>
<body>
<h2 class="hello-title">Hello ${name}!</h2>
</body>
</html>
```


- **Plain HTML5 / CSS and JavaScript Code (no plugin)**
- **No page reloads**
- **Working Back-Button**
- **Bookmarkable Links**
- **Provides offline functionality**
- **Uses RESTful services for data access**

Source: https://en.wikipedia.org/wiki/Single-page_application



Single Page Application (SPA)

■ Simple SPA example

- Java Spring / [IntelliJ](#)

■ Server now returns JSON

- Spa.java
- Application.java remains

■ Plain Javascript

- static/index.html
- ES6 - ECMAScript 2015 ([arrow](#))
- [For demonstration...](#)

```
@RestController
public class Spa {
    @GetMapping("/spa/{value}")
    public String spa(@PathVariable("value") String value) {
        return "{\"value\": \""+value.toUpperCase()+"\""};
    }
}
```

```
<!DOCTYPE html>
<head>
  <meta charset="UTF-8">
  <title>Hello!</title>

  <script type="text/javascript">
    fetch('/spa/'+window.location.hash.substring(1))
      .then(response => response.json())
      .then(body => {
        let element = document.querySelector('.hello-title');
        element.innerHTML = element.innerHTML.replace(/{{name}}/gi,
body.value);
      });
  </script>
</head>
<body>
  <h2 class="hello-title">Hello {{name}}!</h2>
</body>
</html>
```

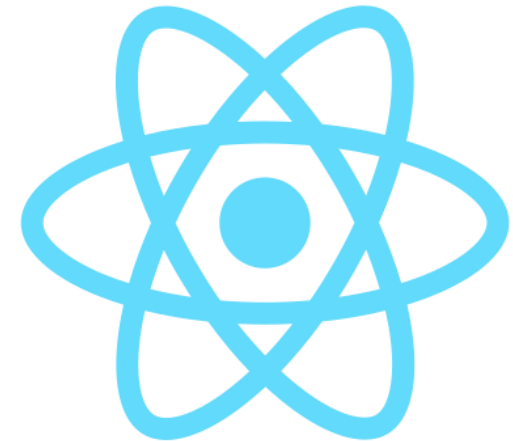
URL: b.socrative.com/student/

Room: HSR

VUE.JS

Vue.js Introduction

- Vue (pronounced /vju:/, like view) is a progressive framework for building user interfaces.
- ...On the other hand, Vue is also perfectly capable of powering sophisticated Single-Page Applications when used in combination with modern tooling and supporting libraries.
- 5 Best JavaScript Frameworks in 2017
 - Angular, React, Vue, Ember, Meteor
- Top 7 Most Popular JavaScript Frameworks in 2018
 - Angular, React, Ember, Aurelia, Meteor, Polymer, Vue
- Vue – low learning curve



Vue.js Introduction

■ Simple Vue example

- Load Vue library

■ Vue built-in template syntax

- {{name}}

■ Reactive

- The data and the DOM are now linked, and everything is now reactive
- `vm.name = 'hallo'`
- `#app` is mount point
 - [`vm.\$mount\('#app'\)`](#)

```
<!DOCTYPE html>
<head>
  <meta charset="UTF-8">
  <title>Hello!</title>
  <script src="https://cdn.jsdelivr.net/npm/vue"></script>
</head>
<body>
  <div id="app">
    <h2 class="hello-title">Hello {{name}}!</h2>
  </div>
  <script type="text/javascript">
    var vm = new Vue({
      el: '#app',
      data: {
        name: 'Hello Vue!'
      },
      created: function () {
        fetch('/spa/'+window.location.hash.substring(1))
          .then(response => response.json())
          .then(body => {
            this.name = body.value;
          });
      }
    });
  </script>
</body>
</html>
```


Vue.js Introduction

■ {{Template Syntax}}

- “Mustache” syntax (double curly braces)
- If {{}} cannot be used (e.g. in attributes), use v-bind - `<div v-bind:id=“someId”></div>`
- `{{ counter + 1 }}` works as well

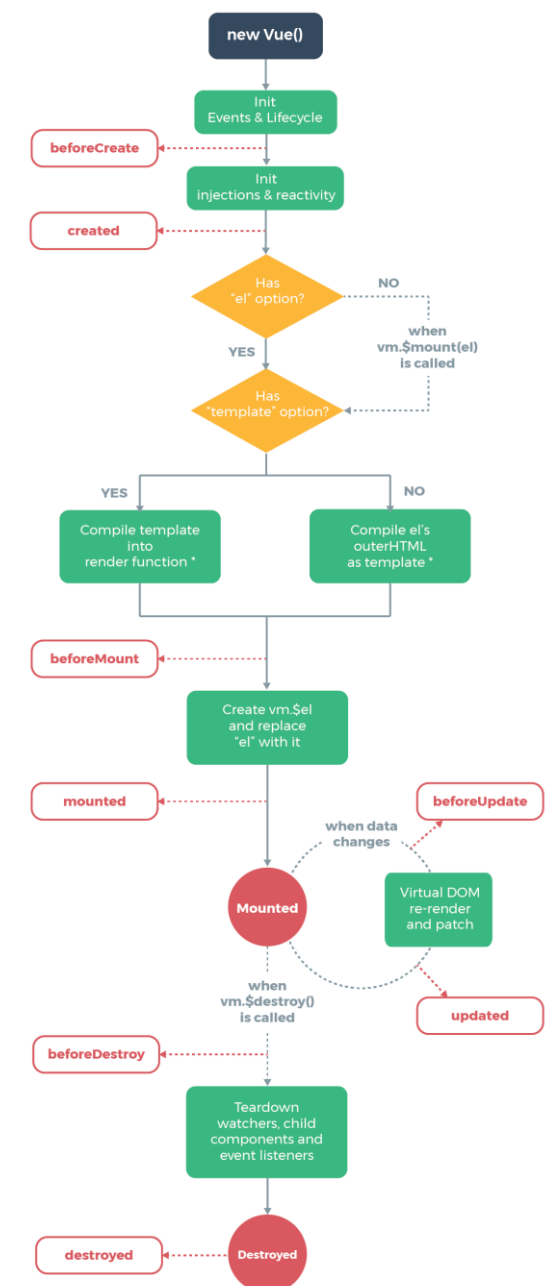
■ Lifecycle Hooks

- Created
- Mounted
- Updated
- Destroyed

■ Directives

- v-model

■ Example on [github](#)



* template compilation is performed ahead-of-time if using a build step, e.g. single-file components

■ Directives

- [v-on // @](#)
 - [Events](#), modifiers
- [v-once](#)
 - Rendered only once. Subsequent re-renders treated as static content and skipped
- [v-html](#)
- [v-if](#), [v-show](#), [v-for](#)
 - Groups with v-if on <template>

■ Computed vs Methods

- Caching!

■ Computed vs Watched Property

- If possible use computed – e.g., two properties – two watchers, one computed property

■ Example on [github](#)

changed from 10 to 9

Add 1

The button above has been clicked 10 times.

alt-c

Once counter: 0

Test

<h1>Test</h1>

1519048417494 | | 1519048412848

1519048417494 | | 1519048412848

URL: b.socrative.com/student/

Room: HSR

Vue.js Example

- Lets write a simple cryptocurrency portfolio manager
- Get a list of top 10 cryptocurrencies
 - <https://api.coinmarketcap.com/v1/ticker/?limit=10>
- Backend already provided, frontend only
 - [v-for](#)
- Add input field for currencies – amount
 - Calculate total
 - Calculate value for each currency
 - Round total ([filters](#))

• Bitcoin	11174.6	5	55873 BTC
• Ethereum	954.591	44.5	42479.2995 ETH
• Ripple	1.15328	0	XRP
• Bitcoin Cash	1546.8	0	BCH
• Litecoin	225.73	11.1	2505.6029999999996 LTC
• Cardano	0.393631	0	ADA
• NEO	137.157	0	NEO
• Stellar	0.458337	34432.123	15781.515959451 XLM
• EOS	9.7699	0	EOS
• IOTA	2.12612	0	MIOTA

total: 116639 USD

■ Components

- Reusable
- Extend basic HTML elements

```
// register  
Vue.component('my-component', {  
  template: '<div>A custom component!</div>'  
})
```

```
// create a root instance  
new Vue({  
  el: '#example'  
})
```

```
// html file  
<div id="example">  
  <my-component></my-component>  
</div>
```

```
//will produce  
<div id="example">  
  <div>A custom component!</div>  
</div>
```

Questions?

Prof. Dr. Thomas Bocek
thomas.bocek@hsr.ch