

# Sport für alle

Factsheet XML Blockwoche

## **Autoren**

*Julien Grüter*

*Joshua Heller*

*Maurizio Hostettler*

*Stephan Stofer*

## **Semester**

*HSLU.BSCI.HS19.XML*

## **Webseite**

<http://xml.enterpriselab.ch/~iaheller/>

## <Auftrag />

### Auftraggeber

Ein Sportzentrum mit diversen Sportanlagen und einem Hotelbetrieb bietet verschiedenste Sport- und Freizeitangebote.

### Idee

Kundensegment im Behindertensport erweitern, Serviceangebote vergrößern im Bereich Sport und Erholung. Die Erreichbarkeit durch einen Webauftritt gewährleisten

### Vorteile

Die Webseite bietet einen Barrierefreien Zugang, die Möglichkeit direkt Online zu buchen inklusive Belegungsdiagramm. Eine automatisierte Bestätigung liefert direkt ein PDF Dokument.

### Vision

Ein neues Zentrum für Behindertensport etablieren, die Akzeptanz für Behindertensport fördern und dank dem erweiterten Angebot können Meisterschaften im Ort durchgeführt werden.

## <Architektur />

Im Web Browser werden die Technologien (X)HTML, CSS und JavaScript genutzt. Die Kommunikation zwischen dem Client und Webserver verläuft über PHP. Für die Verarbeitung, Darstellung und Haltung der Daten werden XML Technologien genutzt.

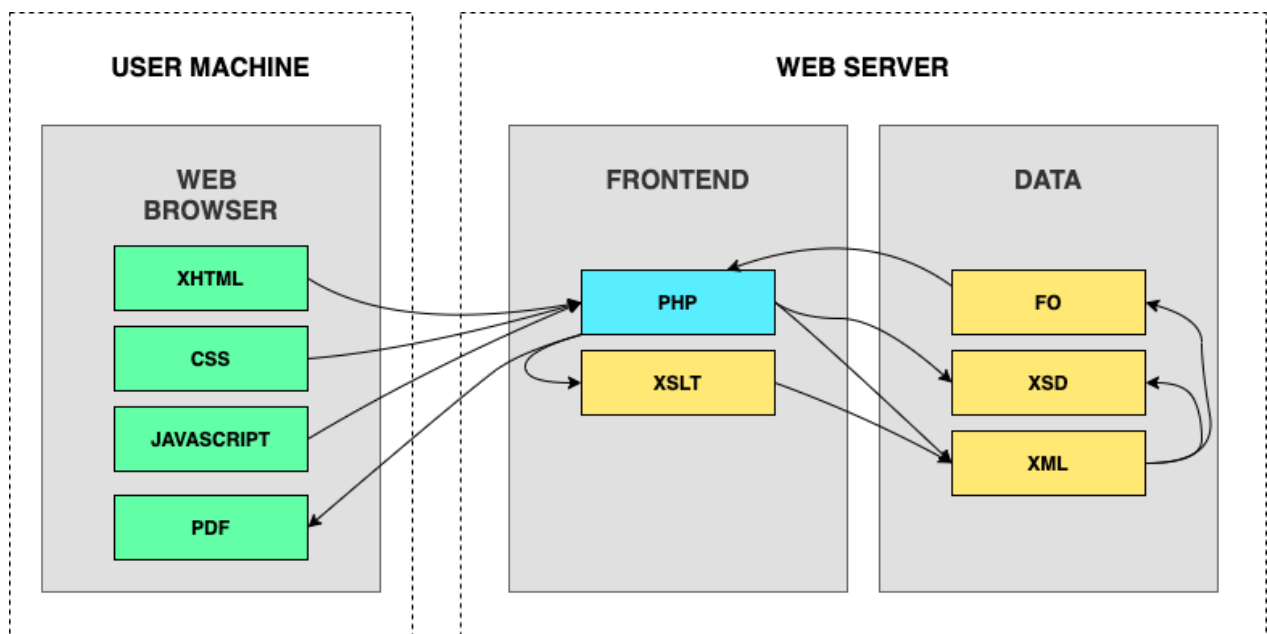


Abbildung 1 Technologienübersicht

Bei http GET wird ein XML Dokument über ein XSL zu einer XHTML Seite transformiert. Bei einer http POST Methode wird immer eine serverseitige Validierung vorgenommen und die Werte werden gegen ein XML Schema validiert.

Folgende Darstellung zeigt alle definierten Use Cases:

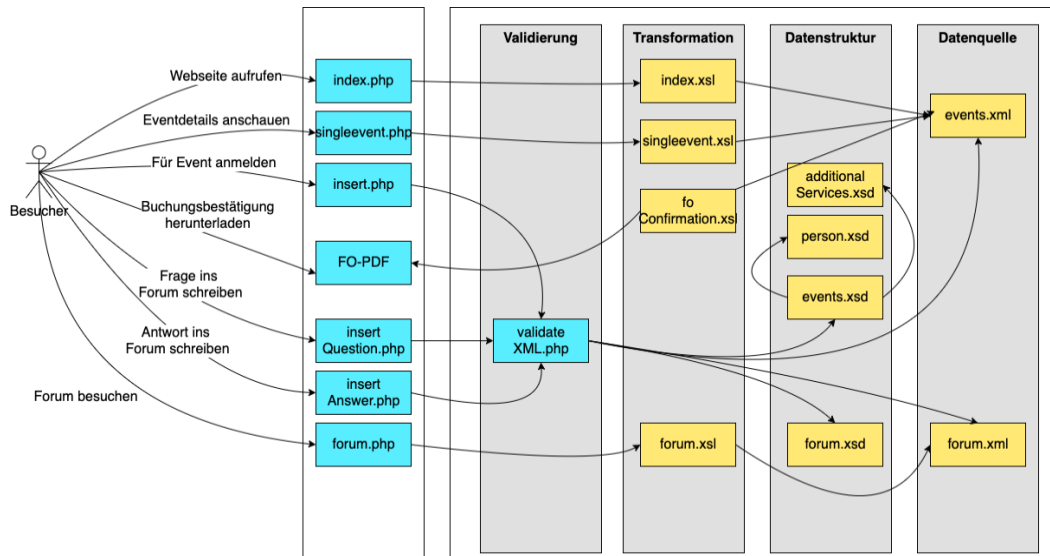


Abbildung 2 Detaillierte Use-Case Übersicht

## <Verwendete Frameworks />

### Bootstrap

Das Framework wird als Basis für das Design (unter anderem auch für Responsive Design) genutzt.

### Fontawesome

Es werden die Icons für Öffnungszeiten, Standort und Kontakt genutzt.

## <Technische Stolpersteine />

### XSLT in XML Quelle ersetzen

Das XML «events.xml» wird für mehrere Funktionen (Übersicht, Anmeldeseite, Buchungsbestätigung) der Webseite als Datenbasis genutzt. Dabei soll für jede Funktion ein anderes XSLT verwendet werden. Das Problem wird mit der PHP-Klasse «XSLTProcessor» gelöst. Damit wird beim Aufruf von einem der genannten Funktionen das XSL Dokument ausgetauscht.

### Docker Read-Only

Zu Beginn haben wir ein Docker File genutzt, um den PHP Webserver, mit der Webseite als Inhalt, zu bauen. Die ersten Probleme sind beim Anmeldeformular aufgetreten. Die Anmeldedaten können nicht ins XML geschrieben werden, da Docker Container read-only sind. Daraufhin haben wir uns für XAMPP entschieden.

## **XAMPP und XSLT**

Der XAMPP hat keine XSL Transformationen ausgeführt. Es musste im php.ini den XSLT-Support manuell eingeschaltet werden.

## **Git**

Da die Verwendung von git im Rahmen eines solchen Projektes relativ neu ist, hat die Implementation von git selber auch einiges an Zeit in Anspruch genommen. Für ein neues Projekt ist nicht implizit klar welche Branch-Strategie man verwenden will und das muss jeweils im Team abgemacht werden.

## **DOM Manipulationen in php**

Das Erstellen/Bearbeiten von XML Strukturen via DomDocument funktioniert zwar sehr gut. Es geht jedoch nicht einfach von der Hand und wirkt umständlich. Dies ist einerseits gegeben durch die PHP Syntax, andererseits aber auch durch die Verschachtelung der Attribute und Elemente.

## **Debugger sehnlichst vermisst**

Ein Debugger für die XSLT-Transformation, insbesondere bei FO, haben wir in etlichen Situationen vermisst. Dies hatte uns viel Nerven und Zeit gekostet.

# **<Einsatz von nicht XML-Technologien />**

## **PHP**

Pro Seite existiert ein PHP Script. Beim Aufruf der PHP Scripts wird das XSLT Sheet im XML Dokument ersetzt, damit das korrekte XHTML Dokument geladen und angezeigt wird. Wir haben keine andere Variante gefunden, um in einem XML Dokument mehrere XSLT anwenden zu können.

## **CSS**

Da Design und Responsive Design nicht im Fokus dieser Arbeit sind, jedoch die Bedienbarkeit und Darstellung verbessern, haben wir uns für das Bootstrap Framework entschieden. Dabei wird das CSS von Bootstrap genutzt.

## **JAVASCRIPT**

Client-seitige Validierung der Forms wird mit JavaScript durchgeführt. Dies dient er Benutzerfreundlichkeit und ermöglicht den User auf falsche Inputs hinzuweisen. Server-seitig wird jedoch mit dem Input gegen das XML Schema validiert.

# **<Fazit />**

Wir alle im Team sind positiv überrascht wie schnell wir, trotz einer uns vorher nicht geläufigen Technologie, eine Webseite entwickeln konnten. XML in Verbindung mit gut definierten Schemas und der Möglichkeit diese zu transformieren sind ein mächtiges Werkzeug.

Für das Erstellen einer Webseite, würden wir nach wie vor auf konventionelle Technologien setzen. Wir sind jedoch überzeugt, dass die Kombination der XML Technologien überall da Sinn macht, wo Dokumente in einem grossen Mass dynamisch erstellt und bearbeitet werden.