

Gewinner Preis des Fördervereins 2018

Preisträgerin:	Lisa Möller (Master of Science)
Thema der Arbeit:	Konzeption eines Real-Time Public-Scoreboard-Systems und Evaluation geeigneter Webtechnologien
Fakultät:	Informatik, Mathematik und Naturwissenschaften (IMN)
Studiengang:	Medieninformatik Master
Betreuer:	Prof. Dr. Michael Frank, Johannes Lüder
Kooperationspartner:	ST Sportservice GmbH



KURZZUSAMMENFASSUNG

In dieser Arbeit wurde ein **Public-Scoreboard-System** konzipiert, mit dem elektronische Anzeigetafeln (Scoreboards) bei Sportveranstaltungen verwaltet werden. Zunächst wurden die Anforderungen analysiert und die **Architektur des Systems** ausgearbeitet. Ausgehend davon wurde evaluiert, wie die einzelnen Komponenten des Systems realisiert werden können.

Für die Umsetzung des Systems wurden populäre **Webtechnologien** wie Angular, Vue.js, React, Electron und Redux untersucht.

ANFORDERUNGEN

Zunächst wurde ein umfangreicher Anforderungskatalog erstellt. Er dient als lückenlose Beschreibung des Systems und deckt alle möglichen Anwendungsfälle ab.

Das System besteht aus einer Menge von Scoreboards, die während eines Sportturniers in **Echtzeit** aktuelle Ergebnisse und Informationen anzeigen. Die Entscheidung, welche Inhalte auf welchem Scoreboard zu welchem Zeitpunkt angezeigt werden, ist abhängig von:

- Sportveranstaltung und Sportart
- Standort des Scoreboards (Publikums- oder Sportlerbereich)
- Status der Veranstaltung (z.B. „anstehend“, „laufend“, „Pause“)

Die angezeigten Inhalte (Screens) werden unter bestimmten Bedingungen (Triggern) gruppiert. Alle Screens und Trigger werden in Program Wheels koordiniert. Ein Program Wheel ist der Ablaufplan für ein Scoreboard.

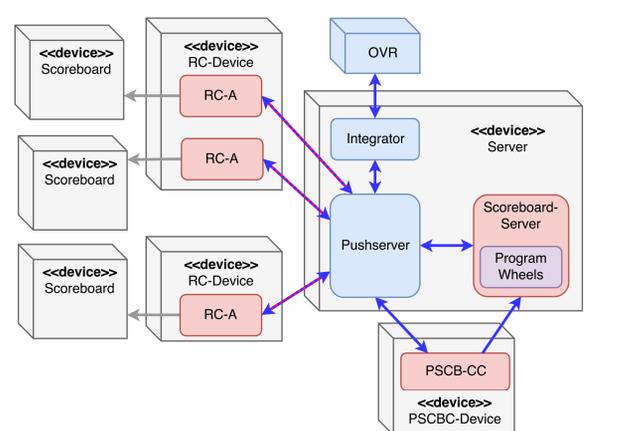


ABBILDUNG 1: Systemskizze des Public-Scoreboard-Systems mit der Betrachtung der Netzwerkverbindungen

ARCHITEKTURANALYSE

Das System ist aus mehreren Komponenten zusammengesetzt. Bereits bestehende Komponenten konnten wiederverwendet werden, um das System flexibel in die Systemlandschaft des Unternehmens einzugliedern.

Dazu zählt ein On-Venue-Results-System (**OVR**), das alle sport-spezifischen Daten liefert. Ein **Integrator** transformiert die Daten in ein passendes Format, das von einem **Pushserver** verarbeitet wird. Dieser Server speichert alle Daten und stellt sie in Kanälen bereit, die von den Scoreboards abonniert werden können.

Folgende Komponenten wurden spezifisch für das Public-Scoreboard-System entwickelt:

- **Public Scoreboard-Control Client:** Konfigurationstool zur Verwaltung von Screens und Program Wheels sowie für das Monitoring
- **Renderer-Client:** Computer, der an 1–4 Scoreboards geschaltet ist und das Rendering der Screens mit den Echtzeitdaten vornimmt
- **Scoreboard-Server:** Koordination und Ablaufsteuerung für die Anzeigen auf den Scoreboards anhand des Status der Veranstaltung, ist die Schnittstelle zwischen Renderer Client und Pushserver

Weitere betrachtete Eigenschaften bei der Architekturanalyse:

- Aufgabenverteilung der Komponenten
- Evaluation der client- oder serverseitigen Ausführung der Program Wheels
- Netzwerkverbindung zwischen den Komponenten
- Verwendete Übertragungskanäle (LAN, WLAN)
- Übertragene Daten

Die Skizze des Systems ist in ABBILDUNG 1 dargestellt.

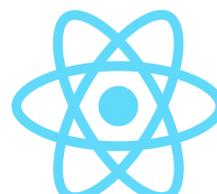
EVALUATION DER WEBTECHNOLOGIEN

Da das Public-Scoreboard-System als Webanwendung umgesetzt werden soll, wurden für die Umsetzung die Webframeworks bzw. Bibliotheken **Angular**, **React** und **Vue.js** evaluiert.

Die Nutzung einer Webanwendung bringt u.a. Vorteile beim zentralen Update-Handlings und bei der Nutzung eines Servers als zentrale Datenquelle. Frameworks sind *Programmgerüste*, um wiederkehrende Probleme einfacher zu lösen.



ANGULAR



REACT



VUE.JS

Für die Evaluation wurden die Frameworks anhand objektiver Kriterien verglichen, z. B. Performance, Syntax, Umfang und Erlernbarkeit. Im Kontext des Projekts sind die Unterschiede jedoch marginal. Zusätzlich wurden subjektive Faktoren einbezogen, z.B. Erfahrungen im Unternehmen. Daraus folgend wurde React für die Umsetzung des Public-Scoreboard-Systems empfohlen.

Im Rahmen der Arbeit wurde zudem der Einsatz von **Electron** und **Redux** evaluiert.

REALISIERUNG

Für die Realisierung wurden folgende Aspekte ausführlich in der Arbeit betrachtet und beschrieben:

- die Real-Time-Anforderung
- Routing in den Client-Komponenten
- die Integration des Pushservers
 - durch einen Pushserver-Client
 - durch die Organisation der Datenstruktur
- die Realisierung des Public Scoreboard-Control Clients
 - Aufbau der Weboberfläche
 - Komponentenarchitektur (komponentenbasierte Entwicklung)
 - Export- und Importmöglichkeiten von JSON-Files
 - Monitoring des Renderer Clients und der Program Wheels
- die Realisierung des Scoreboard-Servers
 - Lösung des Paging-Problems (Anzeigen der Inhalte eines Screens auf mehreren Seiten)
 - Betrachtung der Synchronität aller Anzeigen
 - Koordination der Program Wheels
- die Realisierung des Renderer Clients
 - Evaluation von Electron
 - Betrachtung der clientseitigen Umsetzung des Paging-Problems
- das Ausfallmanagement
 - Ausfall der einzelnen Komponenten
 - Ausfall des Netzwerks
 - Reaktion auf Ausfälle

FAZIT

Anhand der Zielsetzungen für das System wurde ein umfangreicher Anforderungskatalog erstellt; alle Schritte, die zur Anzeige von Daten auf Scoreboards notwendig sind, wurden betrachtet, beschrieben und auf die Realisierbarkeit überprüft.

Die Ausarbeitung des Anforderungskatalogs diente als Grundlage für die architektonische Ausarbeitung, bei der sich die Komponenten des Systems ergaben. Die Eingliederung in die bestehende Systemlandschaft geschah über Wiederverwendung bestehender Komponenten, wodurch Ressourcen gespart werden konnten.

Alle Systemteile wurden auf die Art der Realisierbarkeit überprüft. Dabei wurden Konzepte bestärkt, hinterfragt und überarbeitet, bis am Ende ein realisierbares Fundament für das Public-Scoreboard-System aufgebaut war.