

A Lightweight Web Components Framework for Accessing Generic Data Services in Environmental Information Systems

Eric Braun, Alessa Radkohl, Christian Schmitt, Thorsten Schlachter and Clemens Döpmeier

Abstract This article presents a lightweight framework that allows web components to access data services via standard interfaces. The framework is used in combination with the Generic Microservice Backend (GMB) project which provides a wide range of different services for data management, analysis and visualization e.g. for the implementation of web based environmental information systems. The framework allows an easy implementation of web components by using a centralized communication and controller module that can handle interactive events in the frontend of a web application and data requests towards the backend. The framework can be used to create complex dashboard and portal server pages with a responsive design without much programming.

Keywords Web components · Environmental information systems · Data services · Generic microservice backend (GMB)

1 Introduction

The Web based Information System Team (WebIS) of the Institute for Applied Computer Science (IAI) at the Karlsruhe Institute of Technology (KIT) performs research on the usage of innovative Internet technologies for implementing large scale information systems for application fields, such as environmental information systems, collaborative scientific data and information platforms, smart energy system solutions or Internet of Things (IoT) applications.

In the last years, the focus lied on conceptualizing and implementing an innovative and very generic data management infrastructure, which can be reused in many application contexts. This microservice architecture based framework called the Generic Microservice Backend (GMB) uses Docker containers as runtime

E. Braun (✉) · A. Radkohl · C. Schmitt · T. Schlachter · C. Döpmeier
Institute for Applied Computer Science, Karlsruhe Institute
of Technology (KIT), Karlsruhe, Germany
e-mail: eric.braun2@kit.edu

environment for generic data management services, which can be independently scaled and distributed onto different runtime infrastructures, such as Cloud and/or Big Data infrastructures, private computing clusters with container runtime support, e.g. based on the OpenStack runtime environment, or even single server or development laptops.

GMB services are already used in several environmental information systems, such as the environmental information portals of several German states based on the LUPO portal framework, e.g. the environmental information portal of Baden-Württemberg¹ where the GMB is part of the server side Web Cache for providing fast access to environmental data of different types (e.g. for serving geo data, measurement data, text and binary assets and different forms of master data) to environmental information portals but also to mobile environmental applications. Furthermore, the GMB is used for other applications, like energy system applications as required for the Energy Lab 2.0 and Energy System 2050 (Düpmeier et al. 2016; Hagenmeyer et al. 2016) projects.

While the functionality provided by the GMB services can be easily accessed from client applications by using well-defined REST-based service APIs, in the past, each application using the GMB developed its own dedicated client side data access, application and display logic for displaying GMB data or accessing data driven services. An analysis of the corresponding code in different applications revealed that this code is largely redundant across different applications. Even the presentation logic is often similar and visualizations of the data or interactive data forms for gathering data only differ in minor aspects and style. Therefore, the question arose if application development benefits from a client framework that unifies the code for accessing, processing and visualizing service provided data in client side applications.

Since applications using the GMB are mainly web based or hybrid apps, in a first step, different web technologies were evaluated to find a suitable approach and base technologies for setting up a JavaScript and HTML based web client application framework for handling GMB data in client applications. Since different applications using the GMB services use different JavaScript frameworks, e.g. AngularJS or Reactive.js, for implementing the web application logic, the evaluation quickly showed that the focus lied on standardizing on one of these frameworks or even building a new generic JavaScript framework for web application development. Instead, it can be stated that all modern JavaScript web application frameworks going the direction of using modern web component technology for allowing application developers to build their own complex UI components which can then be reused for different applications. Therefore, the basic idea was to conceptualize and develop a lightweight web framework which can provide an easily extendable library of reusable web components for accessing and presenting GMB data which can be used in different web application contexts side by side with frameworks like

¹<https://www.umwelt-bw.de/>.

AngularJS and Reactive.js, or in more simple use cases by just using only the developed framework and e.g. jQuery.

This article describes the WebIS Web Components (W2C) framework in more detail and how it interacts with the GMB service infrastructure. Additionally, some examples of components and applications are shown that demonstrate the usability of the components for a broad range of environmental applications. Finally, a conclusion is drawn and an outlook completes this article.

2 Generic Microservice Backend (GMB)

In this chapter only a brief overview on the GMB is given. More details on the GMB can be found in Braun et al. (2017) and Schlachter et al. (2017). Figure 1 shows how the Generic Microservice Backend (GMB) is divided into several distinct services which can be accessed by an integrated REST-based service API provided by a gateway. The web component framework (W2C) can be seen as JavaScript client framework for accessing the GMB in applications through a server side gateway.

The GMB provides a bundle of separated and scalable microservices for data management, data analysis and visualization that are access controlled, load balanced and integrated through a central gateway as external access point. Figure 1 shows that each data management service uses its own database to store different types of data in e.g. different types of database management systems (polyglot data management). The *Master Data Service* manages structured objects that describe real objects, such as a nature protection area, lake or river, or different assets of a company, e.g. a measurement station, measurement equipment or a sensor. Typically, this kind of data can be stored in a NoSQL document oriented database. The *Time Series Service* manages time series data using dedicated databases that are optimized for typical time series queries and operations. The *Digital Asset Service* is responsible for files that can be stored as separate documents together with associated metadata. E.g., the persistence layer of this service can consist of one or more content or asset management systems, such as Alfresco.²

Beside these data services, Fig. 1 shows two additional services which allow to store semantic information about the data contained in the data services. The *Schema Service* stores definitions of the common structure of master data objects (e.g. schema) and how structures interrelate. Schemas are stored in JSON schema format. The *Link Service* manages relationships between different data instances in the separate services. These can be relationships between different master data objects but also between a master data object and digital assets or time series. Together, these services allow the instrumentation of complex, integrated data

²<https://www.alfresco.com/>.

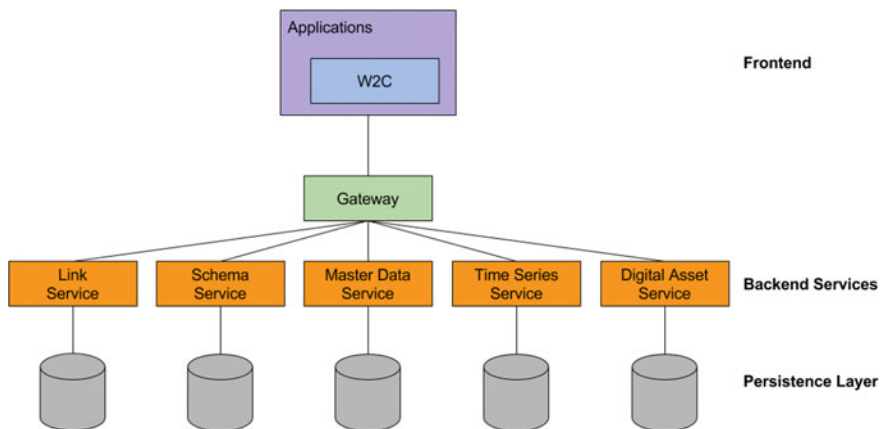


Fig. 1 Communication between the GMB services and the W2C through a central gateway

management models for applications on top of the GMB which cover a broad range of application types.

Each of these services has its own API (using REST Tilkov 2009), but all APIs are exposed through a common integrated API via the gateway. To allow standard access to the data services the GMB API, the gateway supports standard APIs such as Atom feed format or OpenSearch wherever possible. The W2C framework presented in the next chapter uses the gateway API to access the GMB services thereby also instrumenting standard APIs as much as possible. This allows compatibility of the W2C data interfaces with many other services providing the same or similar APIs.

3 A Lightweight Web Component Framework for Accessing Data Services

The basic architecture of the lightweight web component framework (W2C) for accessing GMB and other data services is depicted in Fig. 2.

One part is a library of reusable web components (W2C Components) which can be used by authors to integrate data presentations of GMB data into web pages without programming. The second part is the W2C Base which eases the development of (new) components by providing common client side background logic. The W2C Base has three major responsibilities. It decouples the data access to the services from the components by providing a client side data cache (data broker) as data access interface for components using an asynchronous interface for optimal performance. Additionally, it implements functionality to bootstrap and load Polymer and the basic W2C infrastructure for all components on a web page (W2C Base Class). Finally, it provides functionalities for coordination of several components by

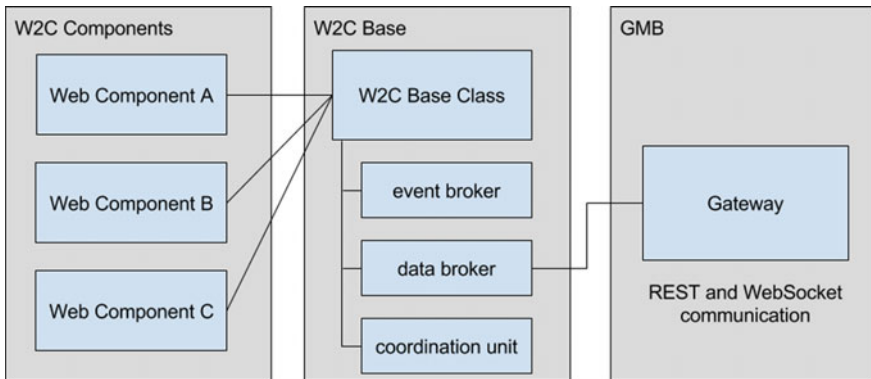


Fig. 2 Architecture of the W2C framework

providing event communication between loaded components (event broker) and exposing coordination functionality (coordination unit) via an API to coordination components. The last two features are necessary for coordinating several components on a web page so that they form a larger interactive single web page application.

With these functionalities in place, the W2C framework allows the aggregation of complex dashboard web pages that include several interactive components coordinated with each other by just assembling W2C web components on a page and configuring them for data access, coordination and event exchange. A click in one component can then e.g. trigger an update of data in the data broker and the arrival of the new data in the broker can result in a coordinated but asynchronous update of the information display of all web components having a dependency on these data.

In order to implement a new web component using W2C, the component just has to inherit from the W2C Base Class that exposes the basic functionalities to the component. The developer can then use all Polymer provided components, additional JavaScript libraries, the W2C provided functionality and already existing W2C components for implementation of their own component. The next chapter shows some sample components and how they are used in example applications.

4 Integration and Usage in Sample Applications

To gather requirements for the W2C concept several existing environmental information system applications were analyzed in advance to the development in relation on how web components enhance the usability of application developers and end users. After developing the basic concepts and first prototypical implementation these environmental applications were also used for the evaluation of the concept. For evaluation, parts of the existing information displayed in these

applications were newly implemented with corresponding web components to verify the usefulness of the approach. In the following two evaluation scenarios are presented to show how W2C components can be used in web applications.

4.1 Theme Park Environment

The Theme Park Environment is the official environmental web portal of the state Baden-Württemberg in Germany with the objective of enhancing the environmental awareness of the general public Ruchter et al. (2003). It contains information about soil, geology, nature areas, and related environmental issues in the state of Baden-Württemberg in the southwest of Germany. It was developed by the Institute for Applied Computer Science of Forschungszentrum Karlsruhe (Karlsruhe Research Center) and the State Institute for the Environment, Measurements, and Nature Protection on behalf of the Ministry of the Environment of Baden-Württemberg and continuously enhanced since its launch in 2006 Döpmeier et al. (2006).

Since the web user interface of the Theme Park was not built for mobile access from the beginning and mobile usage is growing rapidly, a redesign of the Theme Park is foreseen with the aim to fully support mobile devices besides desktop browsers by implementing a responsive and adaptive web user interface based on the current HTML5 standard and aiming also for a better accessibility.

Therefore, the KIT explored in Radkohl (2016) if web component technology would be a good solution to fulfill these tasks by implementing responsive and adaptive web components for the main information display use cases. As information system for the general public the Theme Park instruments many different types of media presentations, such as images, panoramas, slideshows, streaming videos or interactive maps for information display. Thus, for the evaluation of the usability of web component technology for the Theme Park several media and web content display components were developed which could be used in the redesigned Theme Park to display binary media assets and web contents.

Figure 3 above shows an advanced adaptive and responsive component for displaying both web text content and images aligned on the right side of the text by using another component for the display of the images together with associated metadata. The combined component can display the images at different positions depending on provided configuration parameters. The author of the web page can choose between different layout templates to define how text and images are laid out together (the screenshot shows a “text left and image column right” layout). On smaller displays the component repositions the images accordingly to responsive design principles (see the right part of Fig. 3). In addition, the component allows displaying metadata like the title, author information and links that lead to more information about the displayed topic.

The following screenshot (Fig. 4) shows another web component which displays slideshows. This component uses the web component for image display mentioned

```
<text-component content-data="..."></text-component>
```

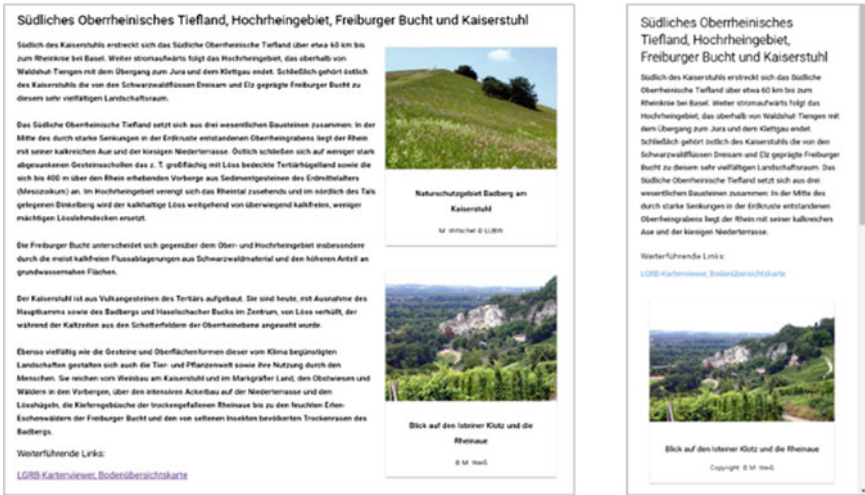


Fig. 3 Web component showing an article with accompanying images

```
<media-slideshow slideshow-data="..."></media-slideshow>
```

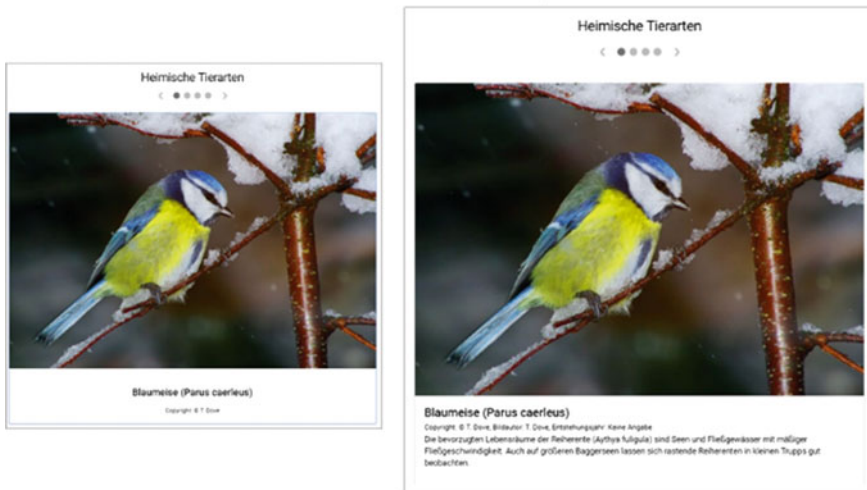


Fig. 4 Slideshow component which presents different images

above to embed this component into a slideshow viewer which cycles through the display of several media files integrated into one slideshow.

The slideshow component can be used in several contexts: for presentation of large scale header banners or smaller slideshows embedded into side columns.

Figure 5 shows a typical web page of the Theme Park as it is currently implemented. The mentioned web components can be used to re-implement key parts of this page by using the text-component for implementing the combined text and image part in the middle of the page in order to be more responsive and adaptable while the slideshow component can be used to provide a modern mobile friendly replacement for the slideshow component on the right side of the shown page which also understands gestures for scrolling through the images.

For the evaluation of the web components several pages of the Theme Park were recreated on a Liferay portal server by first creating a portlet wrapping the above-mentioned web components and the W2C framework, and then using the above discussed components for information display. The evaluation showed that all web pages of the current Theme Park could be easily recreated on a Liferay server by using only a few generic components for media and web content display. Combined with a new responsive corporate design based on e.g. Bootstrap a true responsive and adaptable web user interface can be created to implement the new version of the web user interface of the Theme Park, which would be fully responsible, adaptable and would feature even touch screen gestures for true mobile friendliness. All accessibility features can be implemented as inherent part of the web components so that the information displayed by the web components can also be used by screen readers and other browsers used by persons with disabilities.



Fig. 5 Typical theme park environment web page

4.2 Web Components for Displaying Measurement Data

Many environmental information systems including the Theme Park Environment have to display diagrams showing measurement or other time series data. Currently, often only images of such diagrams are embedded into web pages. However, this poses problems for responsive web applications, because these images of diagrams typically do not scale well for different device display sizes. Images also do not allow exploring measurement data in a more interactive way.

Therefore, as another use case it was explored if web components could be used to ease the handling of interactive responsive visualizations of time series data in web applications. This use case led to the development of the FlexVis framework Braun et al. (2016) which provides a web component and microservice-based framework for integrating time series data driven visualizations into web applications based on implemented visualization components and pre-configurations of component instances that define the mapping between the visualization components and background data services.

Figure 6 shows a data visualization of NO₂ pollution in the air being an example use case needed for visualizing air quality data in environmental data portals.

W2C is compatible with the visualization framework FlexVis Braun et al. (2016) and allows to use FlexVis components together with other W2C components for such visualizations. The FlexVis visualization components are directly configurable using the framework. Additionally, the data binding of the FlexVis components can instrument the data broker of the W2C framework, and FlexVis components can participate in W2C event communication.

One use case of such types of web components is the home page of the Landesamt für Umwelt, Messungen und Naturschutz (LUBW) which is responsible

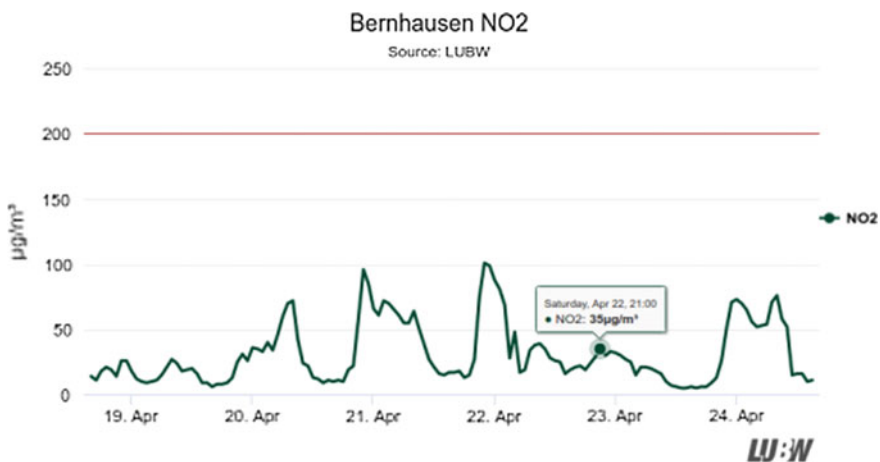


Fig. 6 Visualization of air pollution (NO₂) at a specific measuring station

for providing environmental measurement data for the German state Baden-Württemberg e.g. air quality. The LUBW plans to redesign their web pages e.g. providing information about air quality in a more responsive and interactive way. Early concepts for the new information display combine a map component for station selection, a selection component for physical properties and line charts for data visualization. Figure 7 on the next page shows an early prototype of the new web page. The map component, the line chart component and the property selection box are all implemented using W2C and use the data broker to fetch the relevant data for visualization.

Furthermore, the state of the different components is coordinated using the event broker and coordination unit. The user can select a measuring station on the map and the line chart shows the correct data for this station. Analogously, the property panel at the top can be used to change the physical measurement property the user is interested in which can result in a change of measurement stations shown on the map; and both the map component and the visualization component are updated correspondingly.

Both presented scenarios show, how the different features of W2C can be used. In general, W2C components can be mixed with other web contents to create portal- and dashboard-like web pages composed out of static web content coming from an e.g. portal or content management server and interactive components that can communicate with each other and backend services by using the appropriate brokers. Since web components are implemented to be responsive, adaptive and

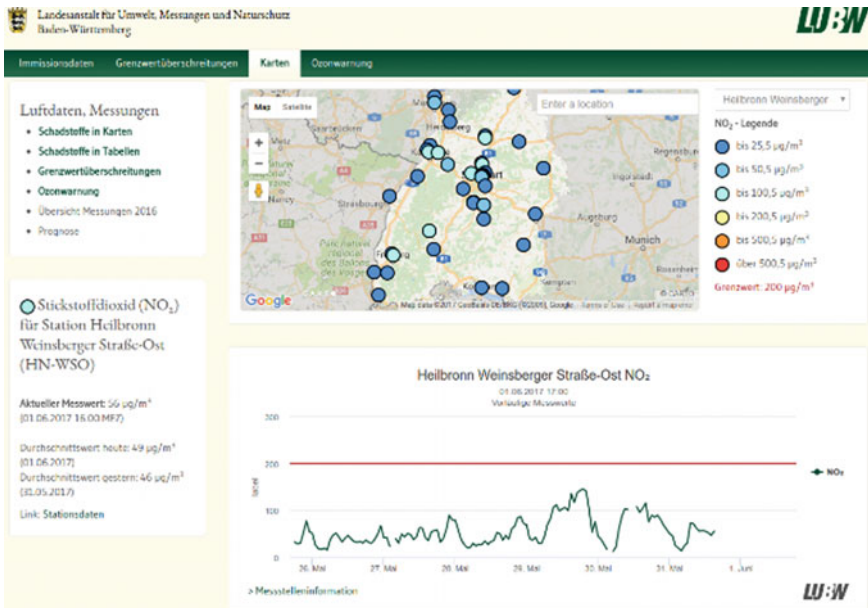


Fig. 7 New concept of LUBW webpage for air quality information

accessibility aware, their usage in web based applications simplifies the provision of more mobile friendly, fully accessible and highly interactive web pages.

5 Conclusion

The previous chapter showed that the W2C framework can ease the implementation of simple but also complex web applications. Furthermore, the article also showed, that the components can be used to instrument a large variety of use cases in different application scenarios. Using the event, data and coordination functionality of W2C, the web components can interact with the backend as well as with each other in a structured and controlled way. The Polymer library as well as the W2C base class help the developer to quickly create new components which provide the same elementary basic interaction elements. This harmonizes the interaction elements across different web components and makes the interaction possibilities more understandable for end users. Furthermore, many components exist that are ready to use and that already provide responsiveness, adaptability, accessibility, a unified design and high customizability.

In the future, W2C can provide even more web components for developers to choose from. Additionally, existing components can be enhanced for even more configurability. Finally, coordination configuration can be eased by implementing a special coordination configuration component.

References

- Braun E et al (2016) Generic web framework for environmental data visualization. In: *Advances and new trends in environmental informatics EnviroInfo 2016*, Berlin, Germany, 14–16 September 2016, pp 289–299. doi:[10.1007/978-3-319-44711-7_23](https://doi.org/10.1007/978-3-319-44711-7_23)
- Braun E et al (2017) A generic microservice architecture for environmental data management. ISESS
- Düpmeyer C et al (2006) Theme park environment as an example of environmental information systems for the public. In: *Environmental modelling and software 2006*, pp 1528–1535. doi:[10.1016/j.envsoft.2006.05.011](https://doi.org/10.1016/j.envsoft.2006.05.011)
- Düpmeyer C et al (2016) A concept for the control, monitoring and visualization center in energy lab 2.0. In: *Energy informatics. Lecture notes in computer science*, vol 9424. Springer, pp 83–94
- Hagenmeyer V et al (2016) Information and communication technology in energy lab 2.0: smart energies system simulation and control center with an open-street-map-based power flow simulation example. *Energy Technol* 4(1):145–162
- Radkohl A (2016) Konzepte und Best Practices für die Nutzung intelligenter UI-Komponenten zur Darstellung von Inhalten in webbasierten Informationssystemen. Bachelor thesis
- Ruchter M et al (2003) Web-based environmental information systems for the public: concepts, potentials, and applicability based on a case study
- Schlachter T et al (2017) Generic web cache infrastructure for the provision of multifarious environmental data. ISESS
- Tilkov S (2009) *Rest and HTTP*. dpunkt. verlag