

OData – The Best Way to Develop Crisis Management Tools? A qualitative and quantitative assessment

Tobias Hellmund

Fraunhofer Institute of Optronics, System
Technologies and Image Exploitation IOSB
tobias.hellmund@iosb.fraunhofer.de

Manfred Schenk

Fraunhofer Institute of Optronics, System
Technologies and Image Exploitation IOSB
manfred.schenk@iosb.fraunhofer.de

Hylke van der Schaaf

Fraunhofer Institute of Optronics, System
Technologies and Image Exploitation IOSB
hylke.vanderschaaf@iosb.fraunhofer.de

ABSTRACT

OData is an HTTP-based protocol that facilitates the implementation of queryable and interoperable RESTful APIs, making it a prominent choice in modern web application development. This research paper evaluates OData's functionalities and assesses its potential to support crisis management software.

To address this question, the study employs a Fit-Gap analysis comparing crisis management requirements with OData's technical capabilities. Additionally, the Crisis Object Database, developed using OData standards, was evaluated for usability through a System Usability Scale test. Results were compared to a legacy system using a proprietary API to identify performance differences.

Findings demonstrate that OData effectively supports the development of interoperable and user-friendly crisis management tools. The usability study reveals significant advantages of OData-based implementations over proprietary alternatives, highlighting OData's ability to fulfill critical requirements while maintaining flexibility. The study concludes that OData can serve as a robust foundation for crisis management software, though specific use cases require careful consideration to maximize its potential.

Keywords

Crisis Object Management, OData, REST, FROST-Server, System Usability

INTRODUCTION

In Crisis Management, situational awareness is a critical component for ensuring an effective response to emergencies and is commonly accepted as good predictor for performance in crisis situations (Endsley, 1988; Endsley, 1995). Crisis managers need accurate information about the context of an incident, particularly the characteristics of the location where the event is unfolding (Hellmund & Moßgraber, 2024). This includes knowledge of the infrastructure, such as buildings and critical facilities, as well as the presence of vulnerable populations that may be at heightened risk during a crisis. In German Civil Protection, all crisis managers are tasked to gather a broad variety of information, even before a crisis occurs (Land Baden-Württemberg). To efficiently gather and utilize these context-specific information, web-based Geographic Information Systems (GIS) have become commonly applied tools in crisis management (Bischoff, 2024; Cinetic Internet Systemhaus GmbH, 2024; DIVERA GmbH, 2023; Fireboard, 2023; Hellmund et al.; Korte, 2024). Web based GIS technology allows for the integration, visualization, and analysis of spatial data, enabling decision-makers to assess risks and plan responses more effectively. However, leveraging the full potential of GIS in crisis management requires

interoperable and flexible systems that can retrieve, exchange, search, filter and manipulate relevant data quickly and reliably. As this is a common problem for every domain of software, standards for common use cases have been established. Standards help ensure that systems can exchange and utilize data from diverse sources, fostering interoperability and scalability.

In this research paper, we explore how open standards, particularly OData (Open Data Protocol), can help supporting crisis managers and enhance their situational awareness to allow for more efficient crisis response. OData provides a robust framework for accessing and managing spatial data through web services, making it an interesting choice for integrating diverse data sources into a GIS-based crisis management platform.

The objective of this paper is to compare crisis management requirements to the capabilities of the OData standard and to make a statement about the general capability of OData to effectively support crisis management. We formulate the research questions:

Research Question 1: Does OData align with the requirements of crisis management?

Research Question 2: Can OData serve as foundation for crisis management software?

We try to answer the research questions with the following approach: with crisis management experts, we have identified relevant laws and guidelines that define the tasks and responsibilities for them. The analysis of these documents is accompanied by inductive categorization, forming the requirements base. We then compare these requirements in a Fit Gap Analysis with the functions provided by OData. As next step, we present the Crisis Object Database, a software application whose functionalities are largely based on a Representational State Transfer Application Programming Interface (REST-API) adhering to the OData standard. For this software, we conduct a System Usability Score test (SUS) with crisis management experts and compare it to a legacy implementation of the Crisis Object Database with a proprietary API. Though the SUS primarily judges the usability of a software, we hope to gain indirect insights about the effectivity of an OData backed crisis management software and the general capability of OData to meet the requirements of crisis managers, enabling us to answer the given research questions.

PRELIMINARIES

In the following section, we introduce technical preliminaries relevant to this paper.

The Representational State Transfer Paradigm

The Representational State Transfer paradigm (REST) is an architectural style designed for managing data and enabling lightweight integration between web-based services (Fielding, 2000). REST operates on top of the HTTP protocol, leveraging its methods (GET, POST, PUT, PATCH, DELETE, etc) to retrieve, create, update, and delete resources. A core characteristic of REST is its statelessness: each client request to the server must contain all the information necessary to process the request, as no session state is stored on the server (Fielding et al., 2017). REST interfaces are flexible and can work with any data model, though JSON (JavaScript Object Notation) is commonly used. This combination of simplicity and versatility makes REST interesting for crisis management use cases, in which large amounts of data need to be processed, as they can easily be managed and distributed over several systems. The REST paradigm does not make any assumptions about the data model and can therefore be transferred to many use cases.

OData – Capabilities and Technical Introduction

The previous section introduced REST as common mean to create and request data over HTTP. Yet, for many use cases, simply retrieving data is not sufficient to fulfill the requirements of crisis managers. Depending on factors such as incident location, crisis type, or crisis phase, only specific subsets of available data are pertinent. Focusing on relevant data prevents informational overload and enhances situational awareness (Hellmund, Schenk, Hertweck, & Moßgraber, 2019b). To achieve this, the OASIS OData standard (OData) has been developed to complement REST interfaces, enabling structured data filtering and selective data retrieval. The standard has been developed by the OASIS OData Technical Committee, which includes members from Microsoft, IBM, Red Hat, SAP SE. For it being a widely adopted open standard, several libraries and tools for building and consuming OData services have emerged (OData, 2023). OData is an extension to the REST paradigm that enhances its functionality: while REST provides a general framework for web services and the access of their data, OData adds specific conventions, query capabilities, and metadata standards. This approach eases out the implementation of common use cases: with OData, operators can specify the information relevant to the current crisis, and the service responds by providing a tailored data subset of its stored data. The following section outlines key querying mechanisms available through OData, including comparisons, mathematical operations, logical operators, string

functions, and geospatial and temporal filters. The necessity for information about the *location* of crisis relevant object data in the affected area, as well as their semantics has been discussed (Hellmund, Schenk, Hertweck, & Moßgraber, 2019a). The geospatial querying capabilities of OData can address geospatial questions such as whether a objects are located in a specific at-risk area, which addresses an important feature (Landtag Rheinland-Pfalz, 2023; Landtag Rheinland-Pfalz, 2024).

In the following, we introduce a subset of available OData operators. Comparison Operators, such as *equal*, *not equal*, *greater than* or *less than* allow finding resources with specific property values, such as names or available resources. Logical Operators, such as *logical and*, *logical or* or *negation* allow combining and inverting filters. With String Operators, such as *substring*, *startswith*, *endswith*, *tolower*, *indexof* the form of strings can be manipulated. Geospatial Functions, such as *geo.intersects*, *geo.length*, *geo.distance*, *st_within* or *st_overlaps* allow analyzing the geospatial relation of two points. Temporal Functions, such as *now*, *mindatetime*, *time*, *year*, *hour* allow analyzing time intervals etc.

OData is agnostic to security functions such as authentication and authorization. Since it is built on top of the Hypertext Transfer Protocol (HTTP), applications can leverage HTTP-based mechanisms for authentication and authorization. However, these security concerns must be explicitly addressed, and any OData service within the crisis management domain should not be deployed without careful consideration of these aspects.

RELATED WORK

(Web-based) GIS applications are commonly applied for the creation of Situational Displays to foster situational awareness (Bischoff, 2024; Fireboard, 2023; Hellmund & Moßgraber, 2024; Korte, 2024). In the following, we present different web based tools for crisis management and discuss, how their backend makes data available to the associated frontend.

Ushahidi

Ushahidi is a web application that makes information collected via crowdsourcing available on a situational map. Ushahidi has been deployed in various crises, such as monitoring the COVID-19 situation in Togo (Hersman, Okolloh, Rotich, & Kobia) or during the explosion at the Beirut port in 2022, as shown in Figure 1 (Ushahidi, Inc., 2024a). Custom questionnaires for crowd sourcing can be created and are answered by citizens through a mobile application. The collected data is visualized using generic representations within the web application. A map displays georeferenced data, graphs represent time-series data, and a table allows filtering of the data. Various map sources can be integrated into the map view. Additionally, data sources such as Twitter, email, or SMS can be connected. Ushahidi is developed in PHP and stores data in a relational database (such as MySQL or PostgreSQL). The application is open-source and released under the GNU LGPL license (Ushahidi, Inc., 2024b). The platform offers a REST-API to access the managed data (Ushahidi, Inc., 2023). The paths contain both functional paths, such as */login*, */register*, */register-password*, as well as entity paths, such as */categories*, */layers*, */contacts* and */messages*. The API documentation does not contain information about the existence of searching functionalities implemented in the REST-API.

- */login*
- */register*
- */reset-password*
- */surveys*
- */posts*
- */categories*
- */categories/{id}*
- */collections*
- */saved-searches*
- */permissions*
- */roles*
- */users*
- */layers*
- */contacts*
- */notifications*
- */messages*
- */exports*
- */api-keys*
- */csv*
- */media*
- */hxl*
- */webhooks*
- */webhooks/{id}*
- */config*
- */config/{id}*
- */config/key/{key}*
- */dataproviders*
- */dataproviders/{id}*

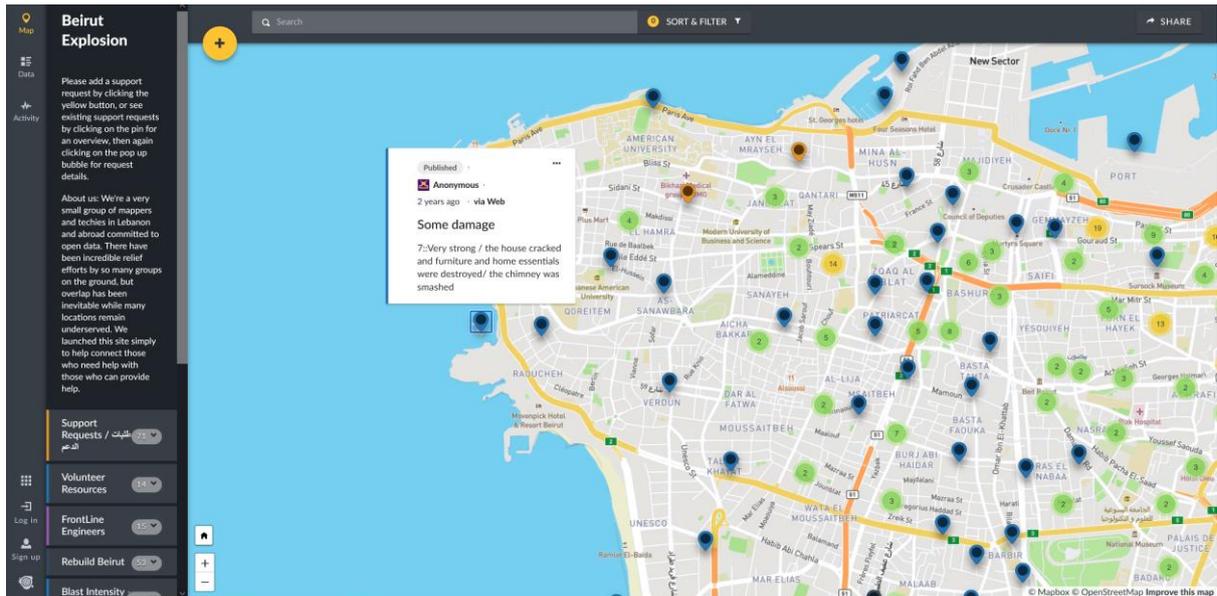


Figure 1: A screenshot of an Ushahidi situational display during the Beirut Explosion 2022 (Ushahidi, Inc., 2024a).

Eden

The Sahana Foundation develops Eden, an open-source platform for disaster management, development, and environmental management. The application is browser-based and aims to present crisis-relevant information. It is implemented in Python, using PyQT as web framework. The application is freely available as an open-source solution under the MIT license (Boon et al., 2024). Eden was notably deployed during the 2015 Nepal earthquake (Sahana Software Foundation, 2024).

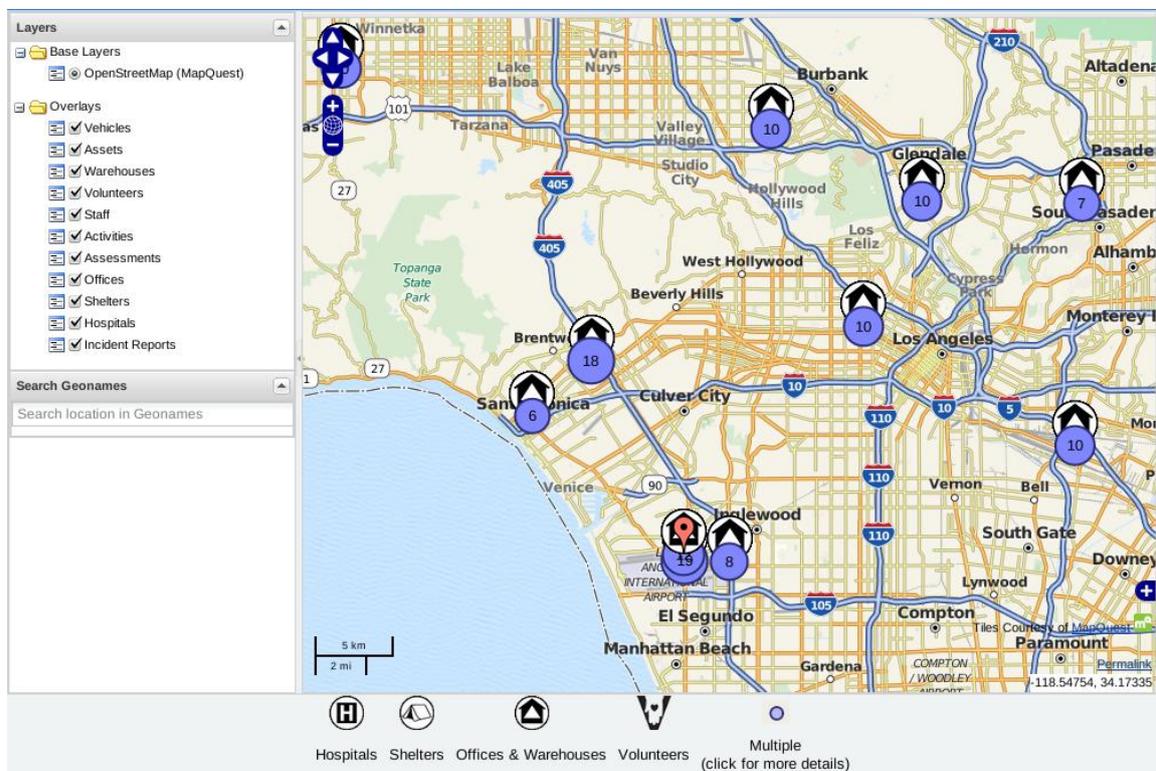


Figure 2: An example deployment of Eden showcasing data in Los Angeles (Tressel et al., 2011).

ESRI ArcGIS Online

ArcGIS Online and ArcGIS Dashboard are browser-based applications from ERSI, designed for the visualization of georeferenced information. ArcGIS Online is a browser-based GIS tool that allows users to create map applications. Within ArcGIS Online, maps can be created based on geographic data, such as geometric shapes or key geographical locations. ArcGIS Dashboard enables the creation of interactive dashboards. It also provides the capability to display time-series data and key performance indicators. Both applications are utilized in disaster management. For instance, the ArcGIS Dashboard was employed by the Robert Koch Institute to monitor and display the COVID-19 situation in Germany during the pandemic (Robert-Koch-Institut, 2023). Figure 3 shows an image of the dashboard.

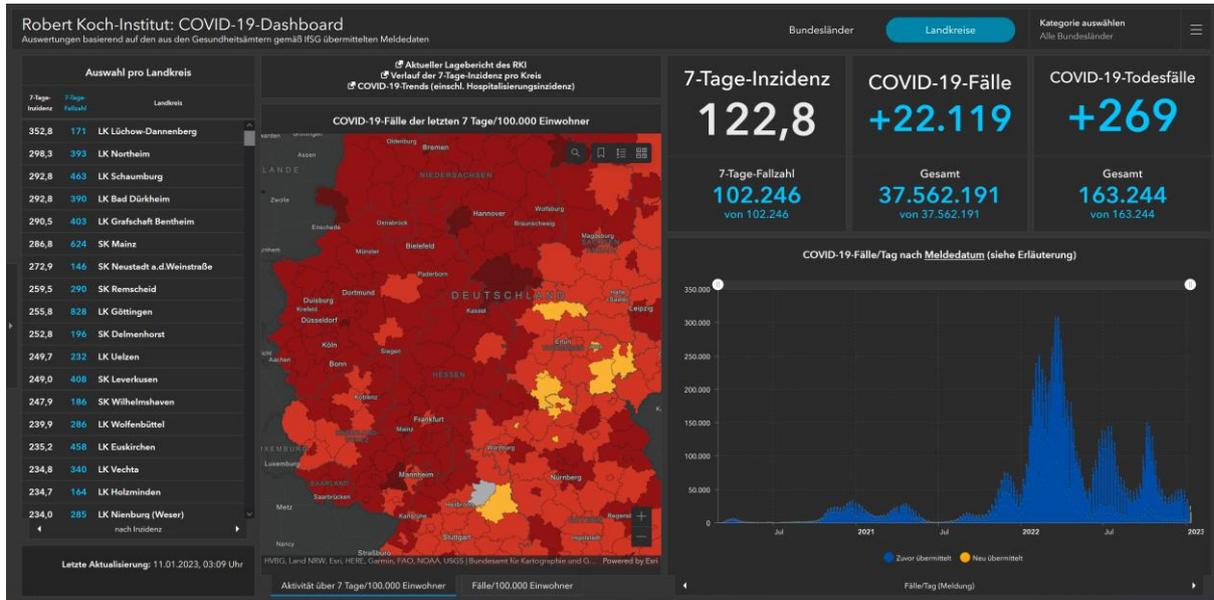


Figure 3: The ESRI ArcGis Covid-19 Dashboard (Robert-Koch-Institut, 2023).

During the heavy rainfalls in Middle-Europe 2021, for example, ArcGIS Online was used, to give a brief overview of the logistics and supplies available for the first responders, as shown in Figure 4. Figure 4: The ESRI ArcGIS Online logistics dashboard for the Ahrtal-flood.

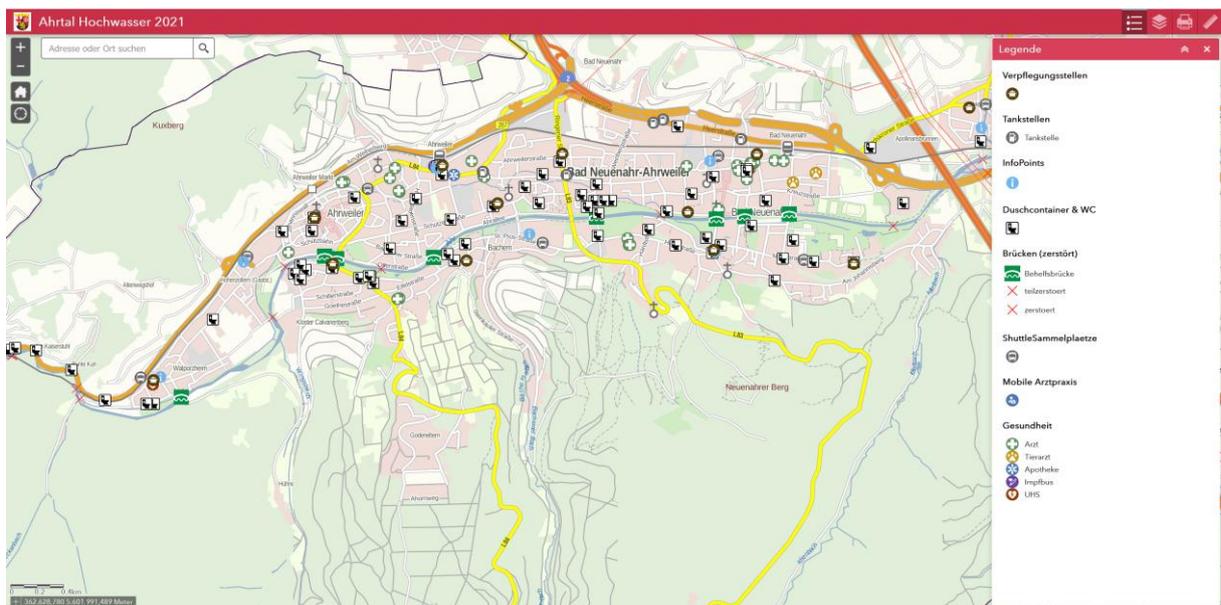


Figure 4: The ESRI ArcGis Online logistics dashboard for the Ahrthal-flood.

The described tools can interact with services from ESRI through a REST-API, which allow temporal filtering through a *where*-clause in the request, as well as geospatial filtering through dedicated properties. The API

implements a subset of standards, such as the Open Geospatial Consortium. No clear information was found about the APIs, which is why we assume a proprietary format of the API.

EVALUATION OF ODATA FOR CRISIS MANAGEMENT APPLICATIONS

In the following, we discuss the utility of REST-based OData services for crisis management applications. For the evaluation we combine quantitative and qualitative approaches. In the first step, we conduct a systematical qualitative fit gap analysis: through expert workshops, we create a set of requirements found in guidelines and regulations for crisis management. The found set is compared against the previously discussed capabilities of a REST-based OData API. Secondly, we conduct a usability evaluation of a software, that implements such API with the help of expert users and analyze the results quantitatively. The following section introduces the methods in more detail.

Fit-Gap-Analysis

To evaluate OData against the requirements of crisis managers, we utilize data from three previously conducted workshops. These workshops were attended by crisis management experts and served the purpose of requirements elicitation for different software. These software tools are not presented in this paper, yet the relevant regulations crucial for the day-to-day tasks of the crisis management experts will be discussed. We used these regulations, which include administrative guidelines, framework recommendations, or similar documents, to find tasks which could be supported by software, as they not only define the legal framework within which crisis managers operate but also establish the responsibilities and tasks crisis managers must carry out. All workshops were attended by experts who, at the time, were actively serving in the public crisis management sector in Germany, with participation ranging from 6 to 12 individuals per workshop. In an inductively categorization process, the found tasks were sorted into different types of software, they could be integrated into. We formed the categories *Situational Display*, *Communication*, and *Data Management*. The tasks then were translated into functional software requirements for a hypothetical application. After this, we evaluated the capabilities of OData against the found requirements. The process is depicted in Figure 5 and described in the following.



Figure 5: Research Process

We found the following set of documents:

- Framework Recommendations for Evacuation Planning (AG Fukushima, 2014)
- Framework Recommendations for Disaster Protection in the Vicinity of Nuclear Facilities (Strahlenschutzkommission, 2015)
- Administrative Regulation of the Ministry of the Interior on the Strength and Organization of the Disaster Protection Service (Ministerium des Innern des Landes Nordrhein-Westfalen, 2024; Ministerium für Inneres, Digitalisierung und Migration, 2019)

The given documents were analyzed for real-world tasks and procedures. In an information condensing approach, we excerpted the tasks that can be supported through software. These tasks were abstracted into high-level application categories that could support the given task. The categories were created through inductive category formation systematically organizing the tasks. Once the tasks were categorized, they were formulated as software requirements. The application categories were:

- *Situational Displays*,
- *Communication*, and
- *Data Management*.

The categories and the respective requirements are shown in the tables below.

Table 1 shows the requirements that were mapped into the category *Situational Display*. RQ1 and RQ2 were established, as the suggestion was to use a single platform in which all relevant information for the situation displayed and is supported by current research (Mentler & Herczeg, 2013). RQ3 was derived from the demand, that Situational Displays should also be available before an actual escalation. These requirements can additionally

be categorized as architectural requirements, as they demand availability and integration capabilities. The following requirements RQ4, RQ5, RQ6, RQ7 can be understood as functional requirements, as they demand presentational functionalities, such as maps, in which additional information should be integrable (such as satellite data and settlement information), as well as the capabilities to extrapolate, search and filter data.

Table 1: Requirements for the application category "Situational Display"

Situational Display	
RQ1.	Several data sources and reports should be integrated into a single platform.
RQ2.	One platform should contain all information required to make decisions.
RQ3.	The Situational Display should be available in the pre-phase of the crisis.
RQ4.	A map should form the basis of the situational display.
RQ5.	The situational display should allow prognoses.
RQ6.	The Situational Display should take object data, terrain structure, settlement patterns and administrative structures into respect.
RQ7.	Searching and filtering capabilities should support the situational display.

Table 2 shows the requirements that were mapped into the category *Communication*. RQ8 was established, as in the documents the need for efficient communication between neighboring regions was stated. This includes ensuring that communication protocols in emergencies respect the hierarchical structure essential for crisis management. Additionally, collaboration with first aid organizations was emphasized, a priority that is also reflected in RQ9.

Table 2: Requirements for the application category "Communication"

Communication	
RQ8.	Communication to neighboring federal entities should be possible. Communication should be planned federatively over borders.
RQ9.	Communication to crisis management organizations (red cross, technical relief aid) should be possible.

Table 3 shows the requirements that were mapped into the category *Data Management*. Commonly, the regulations demanded the collection of certain data before an actual crisis and using the data when dealing with a crisis. This falls in the area of a *Situational Display* and can be explained through the similar requirement RQ6.

Commonly, the management of location data, contact persons or addresses for potentially relevant buildings were demanded in the regulations, see RQ10. The data model would need to be flexible enough, to add category specific information that is only relevant to a subset of the documented objects. For the objects specific documents, such as evacuation plans, had to be taken into regard (RQ16).

The categories explicitly named in the documents for RQ10 were:

- Infrastructure
- Emergency stations
- Iodine tablet distribution centers
- Hospitals
- Senior homes
- Care facilities
- Correctional facilities
- Gathering points
- Civil Protection Contact Points
- Emergency
- Accommodations
- Rehabilitation clinics
- Psychiatric facilities
- Tourist facilities
- Event venues
- Large enterprises
- Residential and work facilities for people with disabilities
- Livestock farms
- Gas stations
- Food supply centers
- Financial institutions
- Industrial companies
- Energy, gas, and water suppliers
- Wastewater disposal services
- Railway companies
- Water and shipping administration
- Air traffic control
- Medical care (general practitioners and specialists, pharmacies, pharmaceutical wholesalers, hospitals and clinics)
- Water sources

Additionally, crisis management forces and their equipment need to be monitored as well (RQ12, RQ13). To allow finding suitable objects or specific crisis management forces, textual, temporal and geospatial searching capabilities are required as well (RQ14, RQ15). For instance, for the distribution of iodine tablets within a 25-kilometer radius of an affected nuclear facility the number of available iodine packs and their location need to be findable. Another example are maintenance intervals, that need to be linked to crisis management vehicles and their equipment.

Table 3: Requirements for the application category "Data Management"

Data Management	
RQ10.	The platform should allow different object categories, such as infrastructure, including category specific information.
RQ11.	The platform should allow analyzing means of transportation.
RQ12.	The platform should allow analyzing crisis management forces.
RQ13.	The platform should allow analyzing the equipment status of crisis management forces.
RQ14.	Searching capabilities should support the data management application.
RQ15.	Stored object data should be geospatially filterable in flexible distances (<5km, <20km, <100km) to specific points, such as the epicenter of a disaster.
RQ16.	The platform should allow the creation and management of evacuation plans for different types of buildings/objects.

System Usability Evaluation

In the following, we describe the usability evaluation of the Crisis Object Database which relies on a REST-API and offers searching and filtering capabilities implemented through OData. First, we introduce the research object itself. Subsequently, we will present the results of the usability evaluation.

Crisis Object Database

The Crisis Object Database is a geographical information system for the management of data relevant for civil protection. As backend service, the Crisis Object Database uses the FROST-Server as data storage. Previously launched as implementation of the SensorThings API, the FROST-Server offers an extendable or even completely customizable data model (van der Schaaf, 2024a; van der Schaaf, 2024b). With plugins for CSV and GeoJSON result formats, established standards are supported, as well. The FROST-Server supports the database implementations PostgreSQL and MariaDB, which are opensource relational databases supporting geospatial data processing.

Frontend functionalities are implemented through WebGenesis, a web-based software framework developed by Fraunhofer IOSB for creating knowledge intensive information systems. It facilitates the integration of heterogeneous data and maps in various formats. As it adheres to multiple standards, WebGenesis facilitates the integration of different services, such as REST-based APIs (Hellmund et al., 2021; Hellmund et al., 2022; Hertweck, Hellmund, & Moßgraber, 2021; Reuter, Hellmund, Moßgraber, & Hertweck, 2023). WebGenesis serves as frontend to the application and offers an authentication layer. The FROST-Server itself is hidden and only accessible through WebGenesis as proxy. The proxy controls the requests and forwards legitimate requests to the FROST-Server.

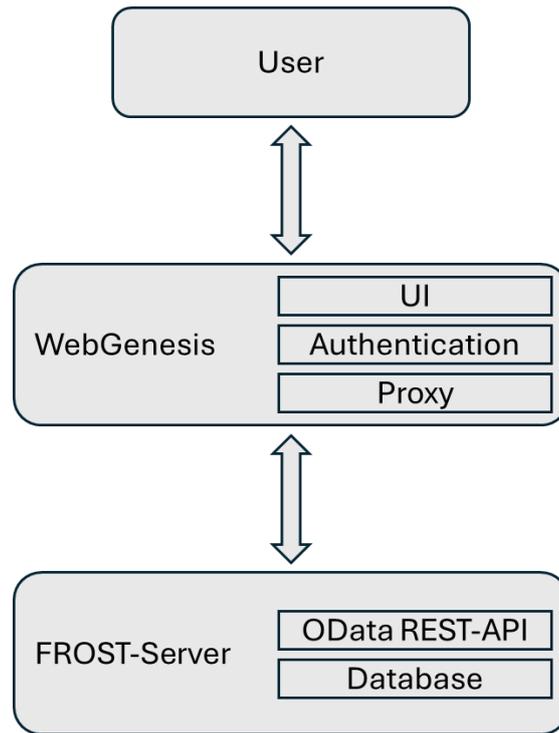


Figure 6: The Crisis Object Data Base Architecture.

Besides mandatory crisis object data, such as name or position, the FROST-Server stores meta-data about available crisis objects categories and their properties are available as well. This meta-data driven design approach allows adding specific information for a subset of the stored data, such as number of pupils for schools or number of available iodine packs in iodine tablet distribution centers. The data model is will not be discussed in this paper.

The different data retrieval and analysis functions of the Crisis Object Database are implemented in the backend through a FROST-Server. Therefore, the interface to the data is through Uniform Resource Locators (URLs). These URLs point to a resource or a set of it and have filters describing the resources of interest. Depending on the server configuration, the entry point to the server for example could be `http://localhost:8888/FROST-Server/ODATA_4.01/`, from which all registered entities are accessible. Figure 7 shows available entity paths, such as `CrisisObjects` for available crisis objects and `CrisisObjectCategories` for categories these objects can be assigned to.

```

  ▼ value:
    ▼ 0:
      name: CrisisObjects
      ▼ url: "http://localhost:8888/FROST-Server/v1.1/CrisisObjects"
    ▼ 1:
      name: CrisisObjectCategories
      ▼ url: "http://localhost:8888/FROST-Server/v1.1/CrisisObjectCategories"
  
```

Figure 7: The Main Access Point to the FROST-Server of the Crisis Object Database.

The following section gives a short introduction of other, exemplary URLs that should address the capabilities of the FROST-Server. The set of all crisis objects can be accessed through the path `http://localhost:8888/FROST-Server/ODATA_4.01/CrisisObjects`, whereas as the path `http://localhost:8888/FROST-Server/ODATA_4.01/CrisisObjects(1)` allows retrieving a single entity.

The given API also allows data retrieval for 1:n relationships, such as all categories for a crisis object via `http://localhost:8888/FROST-Server/ODATA_4.01/CrisisObjects(1)/CrisisObjectCategories`, as shown in Figure 8. Vice versa, the retrieval of all objects belonging to a specific category, is possible as well: `http://localhost:8888/FROST-Server/ODATA_4.01/CrisisObjectCategories(1)/CrisisObjects`.

```

value:
  0:
    @id: "http://localhost:8888/FROST-Server/ODATA_4.01/CrisisObjectCategories(7)"
    id: 7
    name: Hospital
    CrisisObjectCategoryParent@navigationLink: "http://localhost:8888/FROST-Server/ODATA_4.01/CrisisObjectCategories(7)/CrisisObjectCategoryParent"
    CrisisObjectCategoriesChildren@navigationLink: "http://localhost:8888/FROST-Server/ODATA_4.01/CrisisObjectCategories(7)/CrisisObjectCategoryChildren"
    CrisisObjects@navigationLink: "http://localhost:8888/FROST-Server/ODATA_4.01/CrisisObjectCategories(7)/CrisisObjects"
  1:
    @id: "http://localhost:8888/FROST-Server/ODATA_4.01/CrisisObjectCategories(13)"
    id: 13
    name: Iodine Tablet Distribution Point
    CrisisObjectCategoryParent@navigationLink: "http://localhost:8888/FROST-Server/ODATA_4.01/CrisisObjectCategories(13)/CrisisObjectCategoryParent"
    CrisisObjectCategoriesChildren@navigationLink: "http://localhost:8888/FROST-Server/ODATA_4.01/CrisisObjectCategories(13)/CrisisObjectCategoryChildren"
    CrisisObjects@navigationLink: "http://localhost:8888/FROST-Server/ODATA_4.01/CrisisObjectCategories(13)/CrisisObjects"

```

Figure 8: All Categories of the Crisis Object with id '1'.

By applying spatial filters, we can retrieve objects of a specific category, that are within a certain geometric form, such as `http://localhost:8888/FROST-Server/ODATA_4.01/CrisisObjects?$filter=st_within(position, geography'POLYGON((9 48.9, 9 49.1, 9.9 49.1, 9.9 48.9, 9 48.9)))')&$filter=CrisisObjectCategories/id eq 7` which in this example allows retrieving all crisis objects of the category '7' in the given geometric form. The category id '7' could for example indicate the category *hospital*, as shown in the image above.

Basic filtering capabilities, such as string functions, allow the retrieval of objects, for example elements with a specified name or other specific properties. Set operators such as *greater than*, *less than* or *equal* allow the retrieval of crisis objects that have items of defined quantities in stock, iodine tablets for example: `localhost:8888/FROST-Server/ODATA_4.01/CrisisObjects?$filter=iodineTabletStock gt 0`. This allows not only finding objects of a given category, but also finding only those suitable to respond to a given crisis.

The OData-capabilities of the FROST-Server also allow for filter on geospatial characteristics. The request for `http://localhost:8888/FROST-Server/ODATA_4.01/CrisisObjects?$filter=geo.distance(position, geography'POINT(8 48)') lt 1` finds all crisis objects less than 1 unit from the specified point, where the unit is defined in the server settings, which is usually degrees.

The previously discussed HTTP endpoints are made available to the frontend application through WebGenesis as authentication proxy.

Commonly, object data is entered into the system through web forms. Depending on the categorization of the entity input modalities for mandatory and optional data are created based on the meta-data model. Figure 9 shows the detail view of a specific entity stored in the FROST-Server. Through the REST-based nature, imports for larger amounts of data can be implemented as well. The Crisis Object Database implements a CSV importer, that allows mapping columns to fields of the FROST-Server to ease out importing large data batches.

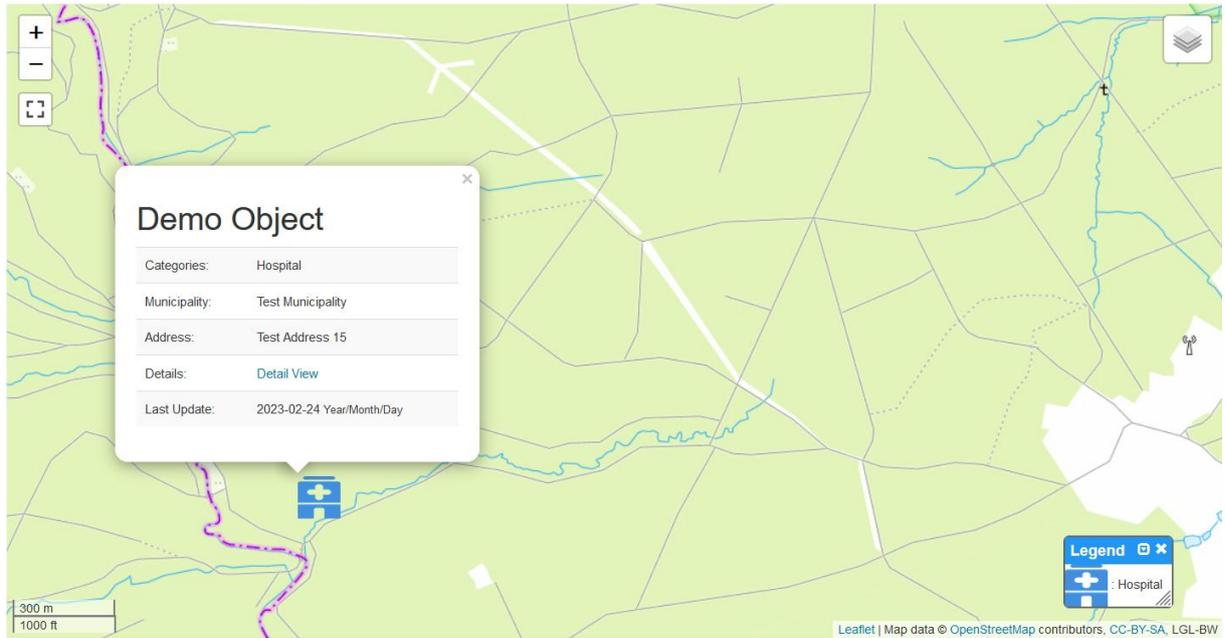


Figure 9: A test crisis object shown in a WebGenesis map.

Furthermore, the Crisis Object Database holds geometrical features for other predefined use cases, such as potentially flooded areas and evacuation zones in the perimeter of nuclear facilities. These allow filtering object data based on geospatial queries allowing to find previously created objects in the given area.

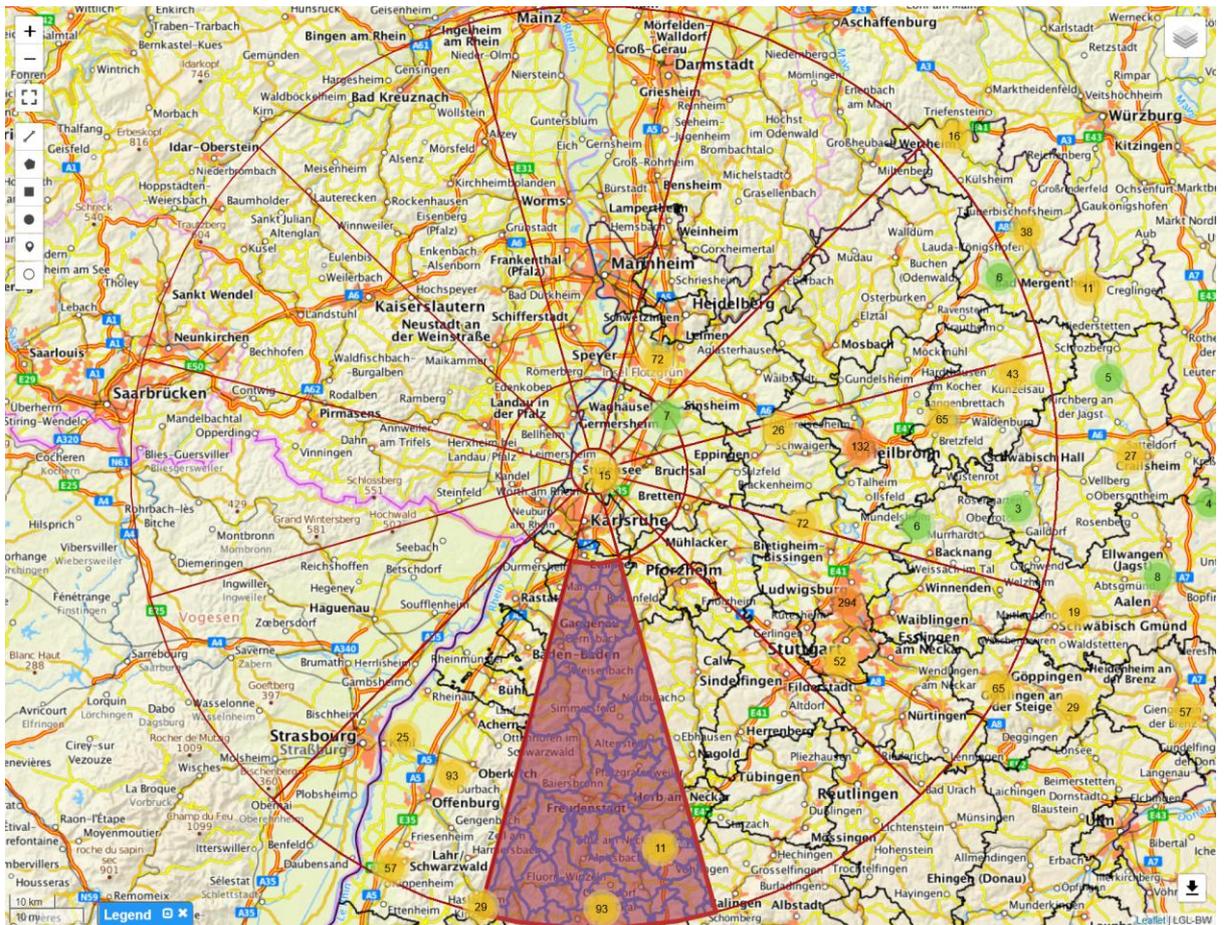


Figure 10: Crisis relevant sectors and crisis objects in a WebGenesis map.

Usability Test

The previously introduced Crisis Object Database is a system deployed and used by the crisis management forces of Baden-Württemberg, Germany. The REST-based functionalities were recently added as machine-readable access layer to the stored crisis objects. The previous version made the object data available through a proprietary interface based on SPARQL. The differences between the two systems led to differences in performance, data retrieval and search possibilities.

The new implementation was tested in a meeting on the 2nd of December 2024 before its final deployment. We decided to utilize this meeting to assess the usability of the Crisis Object Database, now with an OData conformant REST-API, as we had several crisis management experts reachable. To assess the usability, we conducted a test with the System Usability Scale. We decided on this method to minimize time consumption, as the study was not the focus of the appointment. Thus, we decided for this quick, yet comprehensive approach (Jordan, Thomas, Weerdmeester, & McClelland, 1996). The System Usability Scale contains ten different questions, where positive and negative formulated assertions alternate. The scale describes the perceived usability of a system on a scale between 0 and 100. A value 0 describes a virtually unusable system. A value of 100 describes the perfect usability. Lewis assumes, values of 68 and more describe *good* systems (Lewis, 2018). The questions are listed below.

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system quickly.
8. I found the system very awkward to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

The participants joined the appointment online from their respective workstations, from where they would also manage and organize responses during crisis management, when not being present in the crisis staff. All participants were, to the point of study, employed as members of the higher crisis management tasked with the management of crisis responses in the case of an emergency. The usability assessment was conducted after testing the new Crisis Object Database implementation. From four higher crisis management authorities from Baden-Württemberg, three participated with one person each. The subjects were available after the software test and participated in the assessment and answered all questions. The group consisted of two males and female. The questionnaires were presented after using the Crisis Object Database and discussing the Crisis Object Database for 90 minutes.

As comparison to the new implementation, the previous version of the Crisis Object Database was used for another 10 minutes. All participants knew and worked with the previous version of the software, which is why the duration was kept rather short. After this recapitulation, the usability of the second software was also evaluated at hand of the System Usability Scale. The OData and REST-API based evaluation is called *Set 1*, the SPARQL-based evaluation is called *Set 2* in the following. The System Usability Scale scores are discussed in the following.

Table 4: Comparing the System Usability Scale results for both Applications.

	Set 1	Set 2
Subject 1	72.5	40
Subject 2	70	45
Subject 3	82.5	47.5
Mean Value	75	44.17
Standard Deviation	5.4	3.11

Table 4 shows the results of the user study. The older implementation received a mean rating of 44.17 with a standard deviation of 3.11. The new REST-API based implementation received a mean score of 75 on the System usability scale with a standard deviation of 5.4. The effect size expressed as Cohen's *d* is 5.74. This difference in effect size is large, as an effect of 0.8 is considered "large". This difference is remarkable and will be critically discussed later. A Shapiro-Wilk-Test on both rating sets yielded $W = 0.89286$, $p\text{-value} = 0.3631$ for Set 1 and $W = 0.96429$, $p\text{-value} = 0.6369$ for Set 2. As the sets can be assumed as normally distributed, we applied a two-

sided t-test, to find if the mean usability scores are significantly different. As the subjects answered both questionnaires, we chose a paired t-test. We test the following hypothesis.

H_0 : the true difference in usability means is equal to zero.

H_1 : the true difference in usability means is not equal to zero.

The paired t-test shows that the mean usability of found in Set 1 is significantly higher than in Set 2, with an estimated mean difference of 30.83. The difference is statistically significant ($p= 0.009363$) with a calculated t-value of $t=10.262$. The 95%-confidence interval [17.9, 43.8] suggests that the true mean difference lies within this range. We reject H_0 and accept the alternative hypothesis, that the perceived usability of the re-implementation of the Crisis Object Database is higher than before. The results are shown in Table 5. All calculations were done with the software ‘R’; the used code can be found in the appendix.

Table 5: Comparing the System Usability Scale results for both Applications.

Paired t-test results	
Estimated Mean Difference	30.83
p-value	0.009363
t-value	10.262
95%-interval	[17.9, 43.8]

Discussion

In the previous section, we have shown that an application based on an OData-conforming REST-API can achieve higher scores in a usability test, than an application with a proprietary API. Nevertheless, usability was only measured over a brief period of time making the results somewhat questionable. This now leads back to the research question, whether OData is capable of supporting crisis management experts, as we only have a good value of *usability*. To better answer this question, we return to the generated requirements and compare them to the technical capabilities of OData, presented in section OData – and the functions demonstrated in the section introducing the Crisis Object Database. In the following, we discuss the different capabilities.

RQ1: This requirement is not finally judged as fulfilled. While OData, as an open web standard, is designed to facilitate data analysis, the integration of multiple data sources heavily relies on the integration layer. However, standard conformity enables the integration of several layers. Yet, we judge this requirement’s fulfillment state as *neutral*. **RQ2:** Similarly, as data integration is beyond the scope of OData, it can facilitate this function but does not fulfill it. **RQ3:** OData does not conflict with this requirement. Thus, we assume it to be fulfilled. **RQ4:** As discussed, OData capabilities support working with geospatial coordinates. Additionally, the demonstrator included a map implementation. Therefore, this requirement is fulfilled. **RQ5:** This requirement depends on the data model. Nevertheless, if an appropriate forecasting calculation is available, OData’s temporal querying capabilities can enable access to forecast values. We assume this requirement to be fulfilled. **RQ6:** As demonstrated in the implementation, this requirement is fulfilled. **RQ7:** OData provides various querying capabilities, which can be implemented in frontend applications. Therefore, this requirement is fulfilled. **RQ8** and **RQ9:** OData does not address communication or user management. Hence, we assume that these requirements, which fall under the category *Communication*, are not fulfilled. **RQ10:** The demonstrator showed that a flexible data model can be supported using OData, allowing the categorization of different categories. Thus, this requirement is fulfilled. **RQ11, RQ12, and RQ13:** The fulfillment of these requirements depends on the data model. However, OData provides the necessary tools to support them. **RQ14:** This requirement is fulfilled. **RQ15:** OData’s geospatial search functions can locate objects within certain distances, thereby fulfilling this requirement. **RQ16:** While OData might allow the retrieval of stored objects from a server, this aspect was not implemented in the demonstrator, nor have we found relevant examples in research or technical documentation. The status of this requirement is unclear and should be addressed in future research. The previously discussed results can be found in Table 6.

Table 6: Fulfillment state of the found requirements.

Requirement	Fulfillment status
RQ1	Neutral
RQ2	Neutral
RQ3	Fulfilled
RQ4	Fulfilled
RQ5	Fulfilled
RQ6	Fulfilled
RQ7	Fulfilled
RQ8	Not fulfilled
RQ9	Not fulfilled
RQ10	Fulfilled
RQ11	Fulfilled
RQ12	Fulfilled
RQ13	Fulfilled
RQ14	Fulfilled
RQ15	Fulfilled
RQ16	Neutral

Through combining the results of the research approaches, we have shown that open web standards, such as OData conforming REST-services can fulfill the requirements of crisis management experts and create usable software. If used correctly and in the right crisis management software category, *usable* and *potent* software can be developed. In future research, further technical paradigms with similar use cases, such as GraphQL or SOAP should be considered as well to allow for a more complete assessment, what technology offers the best support in for crisis managers.

The usability tests have shown significant results, which we would like to discuss in the following. We believe that these remarkable results arise (partly) because we did not measure bias-free, as the participants conducted two assessments quick after another, likely influencing the results. We also want to point out further weaknesses in our method: through the System Usability Scale, we only achieve an *indirect*, as well as subjective measurement of the capability of OData supporting crisis management use cases. Furthermore, the *usability* does not give any concluding insights into the effectivity and efficiency, with which the subjects can deal with their tasks during a crisis. Lastly, the System Usability Scale can be criticized due to their inherent subjectivity. The remarkable results in the statistical analysis, such as Cohen's d and t-value therefore need to be handled with care.

Though OData offers capabilities that make it suitable for a deployment in the crisis management context, specific requirements, such as user authorization are not addressed by it. An OData-service for crisis management usage should not be used without an access control mechanism that restricts resource access on a fine-grained basis. As roles and access rights in crisis management are only hierarchical, but also overlain with exceptions and special cases defined through laws and other formalities, a dedicated user authorization service must be preceded. OData is not designed for this purpose and tested solutions, such as Role Based Access Control are certainly more fit for these requirements.

Nevertheless, the results this paper show that applications based on OData-compliant REST interfaces are highly potent for disaster protection professionals, if the use case is carefully considered, giving an answer to Research Question 1. We find OData to deliver required functions for many crisis management use cases and should be taken into respect, when developing new crisis management software and the use case is appropriate, answering Research Question 2.

CONCLUSION

In this paper, we have analyzed the capabilities of OData to support crisis management software applications. Through a combination of qualitative and quantitative methods we have investigated, if and how OData can support crisis management experts. The qualitative assessment included a document and guideline analysis generating crisis management expert requirements. These requirements were compared to the technical functions of OData, as well as a developed tool based on OData. Furthermore, we have conducted a usability study with crisis management experts, showing that OData allows the development of *usable* software tools and can significantly outperform software tools without a corresponding API. Due to the many requirements fulfilled and the positive results in the usability study, we evaluate this service as fundamentally capable of supporting crisis management, with the specific use case requiring careful consideration: OData can align with the requirements of crisis management experts to crisis management software and can also form the basis for the different data retrieval and analysis needs of them, if security aspects are addressed through an appropriate technical mediator. Through the integration of only experts on the matter, we achieve relevant assertions about the applicability of OData in crisis management.

ACKNOWLEDGMENTS

This research was indirectly funded by the Ministry of Interior, Digitization, and Migration Baden-Württemberg. The funding provided by the Ministry was limited to the Crisis Object Database, which subsequently formed the basis for the analysis presented in this paper.

We thank the higher crisis management authorities Freiburg, Tübingen, and Stuttgart for their availability in the user study.

APPENDIX

To allow reproducibility of the presented results, we share the R-code used to generate the results in section Usability Test.

```
# Dependency Installation (required for Cohen's d)
install.packages("lsr")
library(lsr)
# Vector setup
x <- c(10,9,8)
y <- c(7,6,6)
# Cohen's d:
cohensD(x,y)
# Shapiro Wilk Test
shapiro.test(x)
shapiro.test(y)
# t-test:
t.test(x, y, paired = TRUE, alternative = "two.sided")
```

REFERENCES

- AG Fukushima (Ed.) (2014). *Rahmenempfehlung für die Planung und Durchführung von Evakuierungsmaßnahmen ein- schließlich der Evakuierung für eine erweiterte Region: RE Evakuierungsplanung*, from https://www.gesetze-bayern.de/Content/Resource?path=resources%2FBayVV_2154_I_2270_BayVV2154-I-2270-A001.PDF.
- Bischoff, F. (2024). *Lagekarte.info*. Retrieved September 18, 2024, from <https://www.team122.at/projekte/lagekarte-info/>.
- Boon, F., Koenig, D., Howden, M., Bhandari, B., Jain, S., & Podhorecký, J. (2024). *Sahana Eden Github Repository*. Retrieved December 11, 2024, from Sahana Software Foundation: <https://github.com/sahana/eden>.
- Cinetic Internet Systemhaus GmbH (2024). *ELStab - Stabsarbeit / Einsatzleitung Feuerwehr Software - FAQ*. Retrieved September 18, 2024, from Cinetic Internet Systemhaus GmbH: <https://www.elstab.de/services/faq/>.
- DIVERA GmbH (2023). *DIVERA 24/7 PRO*. Retrieved March 19, 2025, from DIVERA GmbH: <https://help.divera247.com/pages/viewpage.action?pageId=65821>.
- Endsley, M. R. (1988). Situation awareness global assessment technique (SAGAT). In *Proceedings of the IEEE 1988 National Aerospace and Electronics Conference* (pp. 789–795). IEEE.
- Endsley, M. R. (1995). Measurement of Situation Awareness in Dynamic Systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1), 65–84.
- Fielding, R. T., Taylor, R. N., Erenkrantz, J. R., Gorlick, M. M., Whitehead, J., Khare, R., & Oreizy, P. (2017). Reflections on the REST architectural style and "principled design of the modern web architecture" (impact paper award). In E. Bodden (Ed.): *ACM Digital Library, Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering* (pp. 4–14). New York, NY: ACM.
- Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures*: University of California, Irvine.
- Fireboard (2023). *Fireboard - die Einsatzführungssoftware - Fireboard*. Retrieved March 19, 2025, from <https://fireboard.net/>.
- Hellmund, T., & Moßgraber, J. (2024). Evaluating the mismatch between user requirements and existing situation display tools in administrative crisis management. *Journal of Contingencies and Crisis Management*, 32(2). Retrieved September 18, 2024.
- Hellmund, T., Moßgraber, J., Schenk, M., Hertweck, P., van der Schaaf, H., & Springer, H. (2021). The Design and Implementation of ZEUS: Novel Support in Managing Large-Scale Evacuations. In .
- Hellmund, T., Schenk, M., Hertweck, P., & Moßgraber, J. (2019a). Employing Geospatial Semantics and Semantic Web Technologies in Natural Disaster Management. In *SEMANTiCS (Posters & Demos)* .
- Hellmund, T., Schenk, M., Hertweck, P., & Moßgraber, J. (2019b). Employing Geospatial Semantics and Semantic Web Technologies in Natural Disaster Management. In *SEMANTiCS (Posters & Demos)* .
- Hellmund, T., Schenk, M., Moßgraber, J., Springer, H., Reuter, J., & Hertweck, P. ELD-BS: The Digital Situation Dashboard for Baden-Württemberg. In *ISCRAM 2022 Conference Proceedings – 19th International Conference on Information Systems for Crisis Response and Management* (pp. 233–240).
- Hellmund, T., Schenk, M., Moßgraber, J., Springer, H., Reuter, J., & Hertweck, P. (2022). ELD-BS: The Digital Situation Dashboard for Baden-Württemberg. In *ISCRAM* (pp. 233–240).
- Hersman, E., Okolloh, O., Rotich, J., & Kobia, D. *Togo Covid-19-Dashboard*. Retrieved December 11, 2024, from Ushahidi, Inc.: <https://togocovid19.ushahidi.io/views/map>.
- Hertweck, P., Hellmund, T., & Moßgraber, J. (2021). OPAL-The Toolbox for the Integration and Analysis of IoT in a Semantically Annotated Way. *Sensors (Basel, Switzerland)*, 21(12).
- Jordan, P. W., Thomas, B., Weerdmeester, B. A., & McClelland, I. L. (Eds.) (1996). *Usability evaluation in industry* (First edition). Bristol, PA, London, ©1996: Taylor and Francis.
- Korte, M. (2024). Mobile Lagekarte [Internet]. Gerlingen. Retrieved March 19, 2025, from <https://www.mobile-lagekarte.de/>.
- Land Baden-Württemberg. *Landeskatastrophenschutzgesetz: LKatSG*.
- Landtag Rheinland-Pfalz (2023). *Konsequenzen aus der Flutkatastrophe in Rheinland-Pfalz: Erfolgreichen Katastrophenschutz gewährleisten, Klimawandel ernst nehmen und Vorsorgekonzepte weiterentwickeln: Drucksache 18/8222*, from <https://dokumente.landtag.rlp.de/landtag/drucksachen/8222-18.pdf>.
- Landtag Rheinland-Pfalz (2024, August 02), 18/10000.

- Lewis, J. R. (2018). The System Usability Scale: Past, Present, and Future. *International Journal of Human–Computer Interaction*, 34(7), 577–590.
- Mentler, T., & Herczeg, M. (Eds.). 2013. *Ein benutzer- und aufgabenorientiertes mobiles Anwendungssystem für den Massenansturm von Verletzten*. German Medical Science GMS Publishing House.
- Ministerium des Innern des Landes Nordrhein-Westfalen (2024). *Verwaltungsvorschrift zur Ausführung der Richtlinie für die Beschaffung und Verwaltung der landeseigenen Einsatzmittel im Katastrophenschutz: Runderlass des Ministeriums des Innern* (No. 34-21.52.08.01), from https://recht.nrw.de/lmi/owa/br_bes_text?anw_nr=1&gld_nr=2&ugl_nr=215&bes_id=53343&menu=0&sg=0&aufgehoben=N&keyword=verwaltungsvorschrift.
- Ministerium für Inneres, Digitalisierung und Migration (2019). *Verwaltungsvorschrift des Innenministeriums über die Stärke und Gliederung des Katastrophenschutzdienstes: VwV KatSD*. Land Baden-Württemberg.
- OData (2023). *The Protocol for REST APIs*. Retrieved March 19, 2025, from <https://www.odata.org/>.
- Reuter, J., Hellmund, T., Moßgraber, J., & Hertweck, P. (2023). The KERES Ontology: Protecting Cultural Heritage from Extreme Climate Events. *Heritage*, 6(5), 4015–4041.
- Robert-Koch-Institut (2023). *COVID-19-Dashboard*. Retrieved January 11, 2023, from <https://experience.arcgis.com/experience/478220a4c454480e823b17327b2bf1d4>.
- Sahana Software Foundation (2024). *Sahana Deployments in Nepal*. Retrieved December 11, 2024, from <https://eden.sahanafoundation.org/wiki/Deployments/Nepal>.
- Strahlenschutzkommission (2015). *Rahmenempfehlungen für den Katastrophenschutz in der Umgebung kerntechnischer Anlagen: Empfehlung der Strahlenschutzkommission*. Bonn, from https://www.innenministerkonferenz.de/IMK/DE/termine/to-beschluesse/2015-12-03_04/anlage16.pdf?__blob=publicationFile&v=2.
- Tressel, P., Boon, F., Kohli, S., Koenig, D., Lopez, B., Lev, E. et al. (2011). *Sahana Eden Documentation*. Retrieved December 11, 2024, from Sahana Software Foundation: <https://archive.flossmanuals.net/sahana-eden/what-is-sahana-eden.html>.
- Ushahidi, Inc. (2023). *ushahidi platform V5: Ushahidi REST-API Documentation*. Retrieved December 11, 2024, from <https://docs.ushahidi.com/ushahidi-platform-rest-api-documentation>.
- Ushahidi, Inc. (2024a). *Beirut Explosion Dashboard 2020*. Retrieved December 11, 2024, from Ushahidi, Inc.: <https://beirut.ushahidi.io/views/map>.
- Ushahidi, Inc. (2024b). *Ushahidi GitHub Repository*. Retrieved December 12, 2024, from Ushahidi, Inc.: https://github.com/ushahidi/Ushahidi_Web.
- van der Schaaf, H. (2024a). *FROST®-Server: An open source implementation of OGC SensorThings API*. Retrieved December 11, 2024, from <https://www.iosb.fraunhofer.de/en/projects-and-products/frost-server.html>.
- van der Schaaf, H. (2024b). *FROST-GitHub Repository*. Retrieved March 19, 2025, from <https://github.com/FraunhoferIOSB/FROST-Server?tab=readme-ov-file#extendable-data-model-and-pluggable-apis>.