

# Responsible, Automated Data Gathering for Timely Citizen Dashboard Provision During a Global Pandemic (COVID-19)

TOBIAS RÖDDIGER, MICHAEL BEIGL, and DOMINIK DÖRNER, Karlsruhe Institute of Technology (KIT), Germany

MATTHIAS BUDDE, Disy Informationssysteme GmbH and Karlsruhe Institute of Technology (KIT), Germany

---

Creating a public understanding of the dynamics of a pandemic, such as COVID-19, is vital for introducing restrictive regulations. Gathering diverse data responsibly and sharing it with experts and citizens in a timely manner is challenging. This article reviews methodologies of COVID-19 dashboard design and discusses both technical and non-technical challenges associated. Advice and lessons learned from building a citizen-focused, automated county-precision dashboard for Germany are shared. Within four months, the web-based tool had 5 million unique visitors and 70 million sessions. Three developers set up the basic version in less than one week. Early on, data was screen scraped. An iterative process improved timeliness by adding more fine-grained data sources. A collaborative online table editor enabled near real-time corrections. Alerting was setup for errors, and statistics apply for sanity checking. Static site generation and a content delivery network help to serve large user loads in a timely manner. The flexible design allowed to iteratively integrate more complex statistics based on expert knowledge built on top of the collected data and secondary data sources such as ICU beds and citizen movement.

**CCS Concepts:** • Software and its engineering → *Software prototyping*; • Computer systems organization → *Data flow architectures*; • Human-centered computing → *Geographic visualization*;

Additional Key Words and Phrases: Covid-19, dashboard, data gathering, pandemic, tutorial

**ACM Reference format:**

Tobias Röddiger, Michael Beigl, Dominik Dörner, and Matthias Budde. 2020. Responsible, Automated Data Gathering for Timely Citizen Dashboard Provision During a Global Pandemic (COVID-19). *Digit. Gov.: Res. Pract.* 2, 1, Article 9 (November 2020), 9 pages.

<https://doi.org/10.1145/3428471>

---

## 1 INTRODUCTION

COVID-19 fundamentally changes society with government regulations that impact every individual's life in one way or another [12]. With information always available at the tip of our fingers, there is an ongoing global interest to understand the developments and government's response to a crisis in near real-time (e.g., social media [9]).

---

We acknowledge support by the KIT-Publication Fund of the Karlsruhe Institute of Technology.

Authors' addresses: T. Röddiger, M. Beigl, and D. Dorner, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany; emails: {roeddiger, beigl}@teco.edu, dominik.doerner@student.kit.edu; M. Budde, Disy Informationssysteme GmbH, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany; email: budde@teco.edu.



This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2020 Copyright held by the owner/author(s).

2639-0175/2020/11-ART9

<https://doi.org/10.1145/3428471>

The dashboard presented by the Johns Hopkins University showed a timely response that helped to understand the global dynamics of the SARS-CoV-19 outbreak [5]. However, especially when the virus started to spread across countries, little information was available for citizens on a sub-national level in a centralized manner (e.g., at county precision). Even though there is a strong desire by citizens and governments to get rapid access to such information, time constraints limit the availability of, e.g., a visually rich dashboard. Additionally, information overload and various data formats make the integration process complex [11]. Related research stressed the importance of the latest technological advancements in Global Information Systems (GIS) to provide access to geographical COVID-19-related information [8]. However, little insights have been published on collecting case numbers in an automated and responsible manner from public resources.

This work presents the system architecture and lessons learned from developing a non-commercial fine-grained COVID-19 dashboard for the general public. Developing such a dashboard is challenging, especially when it has to contain timely and accurate information. From a technical perspective, a system needs to be

- *lightweight*, so that it can be quickly set up and deployed,
- *dynamic*, so that it can be easily adapted to changing data sources, user requirements, and so on,
- *highly automated*, so that it can provide timely information,
- *cost-effective*, so that it can potentially be operated by low-funded parties without commercial interest, and
- *scalable*, as in a pandemic situation, the potential interest in timely data can yield high peaks of access to the platform.

There are also diverse non-technical aspects to be considered, such as the granularity of information and including appropriate meta information, as in a pandemic, users are bound to be insecure and unfamiliar with the meaning and expressiveness of certain metrics. There is no documented dashboard development process for quickly responding to drastic and lasting developments such as a global pandemic. Our work focused on building an automated pipeline that unravels spatially localized, cost-efficient information on COVID-19 case numbers from a plethora of online resources in Germany. Aside from confirmed case reporting, we expanded the functionality to include detailed graphs and tables, e.g., on deaths, patients recovered, ICU capacity, and travel activity, as well as analytics, such as statistics and projection forecasts according to scientific models. So far, our dashboard had approximately 5 million unique visitors and 70 million user sessions. Our report serves to share advice and lessons learned through:

- a short review of existing COVID-19 related dashboards and their applied methodologies,
- a tutorial on how to establish a light-weight pipeline for a fully-automated, centralized dashboard for COVID-19 case reporting on a sub-national level from public sources in Germany with optimization for heavy user loads,
- a discussion of related non-technical aspects, such as data quality and legal barriers in data collection, and
- the usage habits of 5 million unique visitors and dynamics of usage through a small data-driven analysis.

Compared to the central dashboard by the German government,<sup>1</sup> which follows an official case reporting process, our system focused on collecting data from public resources such as tables on the websites of the federal governments, county administrations, and local news outlets. This process allowed us to launch a centralized website about two weeks earlier than any other platform in Germany and, at its core, matches the approach by the Johns Hopkins University. Additionally, to maximize flexibility, our website does not depend on GIS frameworks.

---

<sup>1</sup><https://experience.arcgis.com/experience/478220a4c454480e823b17327b2bf1d4>.

Table 1. Overview of Different COVID-19 Dashboards

Dashboard	Purpose	Scope	Granularity	Target Audience	Data Source(s)	Reliability	Latency
World Health Organization <a href="https://covid19.who.int">https://covid19.who.int</a>	[15] Case Reporting	Global	Country	General Public	Government Channels	Official Data	Days
Johns Hopkins University <a href="https://coronavirus.jhu.edu/map.html">https://coronavirus.jhu.edu/map.html</a>	[5] Case Reporting	National (USA); Global	US: County/City/Borough; Others: Country/Region	General Public	Official Bodies, Local News, Other Websites	n/a	Hours
/r/covidmapping <a href="https://covidmap.global">https://covidmap.global</a>	[10] Case Reporting	Global	Region/District/City	General Public	Local News, Crowdsourcing	Limited	n/a
Boston Children's Hospital <a href="https://healthmap.org/en">https://healthmap.org/en</a>	[2] Case Reporting	Global	Country/County	General Public	News Sites	n/a	Daily Update
CoronaVis <a href="https://coronavis.dbvis.de/de/">https://coronavis.dbvis.de/de/</a>	[1] Case Reporting, ICU Capacity	National (Germany)	District/City	General Public	Online Hospital Reports	Official Data	Days
LitCovid <a href="https://www.ncbi.nlm.nih.gov/research/coronavirus">https://www.ncbi.nlm.nih.gov/research/coronavirus</a>	[4] Literature Collection	Global	Paper Type/Country/Publisher/Chemicals	Experts	PubMed	Curated	Daily Update
Cytel Corporation Clinical Trial Tracker <a href="https://covid-trials.org">https://covid-trials.org</a>	[13] Clinical Trial Tracking	Global	City	Experts	International Clinical Trial Registries	Official Data	n/a
NextStrain <a href="https://nextstrain.org/ncov/global">https://nextstrain.org/ncov/global</a>	[6] Genomic Epidemiology	Global	City	Experts	GISAID Database	Research Data	n/a

Beyond case reporting on different granularity, which mainly addresses the general public, specialized systems target expert audiences with data on COVID-19 literature or clinical trial tracking.

## 2 RELATED WORK: A SHORT REVIEW OF COVID-19 DASHBOARDS

The massive impact of COVID-19 resulted in many initiatives around the globe. The following overview of dashboards (see also Table 1) is by no means a complete list of COVID-19 related resources. The main focus is to present an overview of different approaches covering various aspects of the pandemic.

The dashboard by the *World Health Organization (WHO)* [15] follows official reports by government agencies shared by direct communication channels. Therefore, in general, the dashboard is behind secondary online resources, but it has very high reliability. For case tracking, the dashboard by the *Johns Hopkins University* [5] relies on the data reported by official bodies (e.g., WHO, CDC, ECDC, DXY.cn), local news, and health departments, as well as other data collection sites (e.g., worldometers<sup>2</sup>). This principle speeds up case number gathering. Initially, numbers were collected manually, but with continuous growth, a semi-automated process applies today. The commercial ArcGIS framework was used for building the frontend part, and there are even tutorials available on how to create such dashboards with the software [7]. Other information hubs, such as /r/covidmapping [10], apply crowdsourcing approaches to collect news articles on confirmed, recovered, quarantined, and death cases. However, in this example, data collection is strongly community-driven and therefore reliability may be limited [3] and the effort is high (200+ authors from 36+ countries). Similarly, automated approaches exist to gather news related COVID-19 information [2]; however, the resulting map is rather sparse. Other centralized solutions include gathering literature [4] by scanning PubMed based on keywords, which are then curated by human individuals supported by machine learning. Also global clinical trials are aggregated by automated but also manual search within online clinical study registries [13]. Two independent reviewers check every entry before publication on the website. Also, an overview of the genomic epidemiology [6], including spatial and temporal visualizations, is available. Data is sourced by pulling public databases such as NCBI and ViPR, and even Github repositories. Furthermore, data on intensive care unit capacity is available in many countries, e.g., in Germany the *German Interdisciplinary Association for Intensive Care and Emergency Medicine* collects such information manually through online reporting by the hospitals. Based on that data, dashboards were developed (e.g., [1]).

<sup>2</sup><https://www.worldometers.info/coronavirus/>.

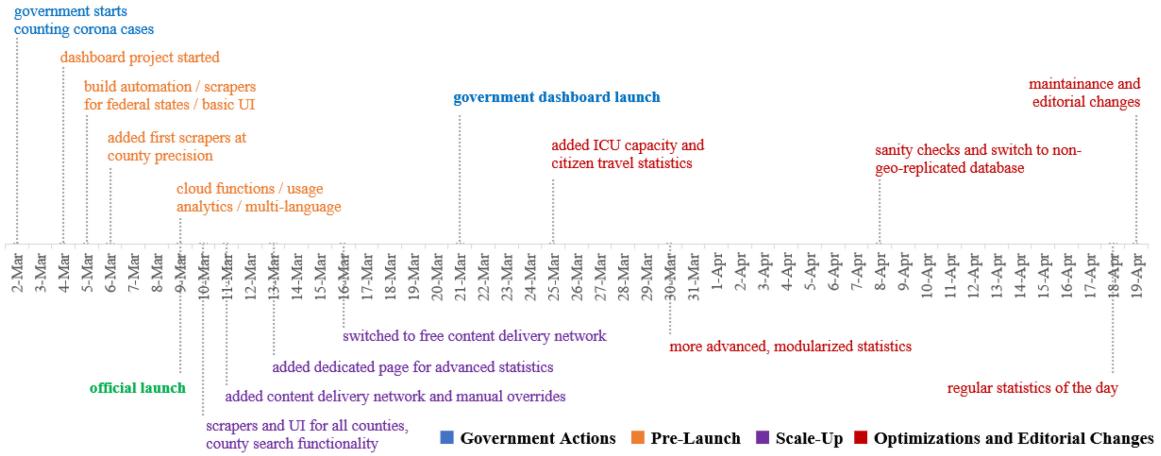


Fig. 1. The graph shows the different activities of our development process and illustrates how we were able to launch a basic version of the dashboard within less than a week.

### 3 BUILDING AN AUTOMATED CASE DATA GATHERING PIPELINE

The previous paragraphs introduced the methodology and approaches of different types of centralized COVID-19 information sources. We focused on building a COVID-19 dashboard for case numbers in Germany by combining a fully automated process with manual overrides, which, from a methodological perspective, matches the well known approach by the Johns Hopkins University. The following paragraphs focus on providing insights from a technical perspective on how we setup our website *coronazaehler.de*.

#### 3.1 Setup Process

Our approach followed a rigorous iterative approach from simple setup to fully automated pipeline. A first functional version was online within less than a week. In Section 3.2, we present the final architecture that we achieved, including considerations why we made certain decisions. However, first, we give an overview how we managed to set up our dashboard in short time. Figure 1 shows how we approached the development process.

A key factor for establishing timely dashboard reporting is the availability of the respective data. We employed web scrapers, i.e., software that automatically fetches websites and extracts, formalizes, and stores data from them. The timeline illustrates that within just two days, we could set up basic scrapers, and within six days, we had scrapers for all counties. Before launching, we had already added *Google Analytics* to understand the usage dynamics. After launching, activities were focusing on functionalities and increasing site stability. We quickly realized that the heavy user load needs a content delivery network (CDN) that we added two days after launch. Additionally, manual overrides became especially important in the beginning phase as it allowed us to do small corrections in real time. We soon realized that the CDN we had chosen was not cost-ideal, so we switched to a free alternative for static websites. We also decided to add sanity checks for the data scrapers running in the backend part of the website. Finally, we added a “statistic of the day,” which regularly shows different statistics directly on the website’s starting page. From then on, the changes we did were editorial (e.g., adding new statistics) and fixing small bugs.

#### 3.2 Overall Architecture

Figure 2 shows the website’s overall architecture. A set of cloud functions periodically scrapes data sources (e.g., websites and PDF documents) for new case numbers. Additionally, we set up a *Google Sheet* document for manual corrections. This building block was particularly useful during the beginning of the website, e.g., we could add

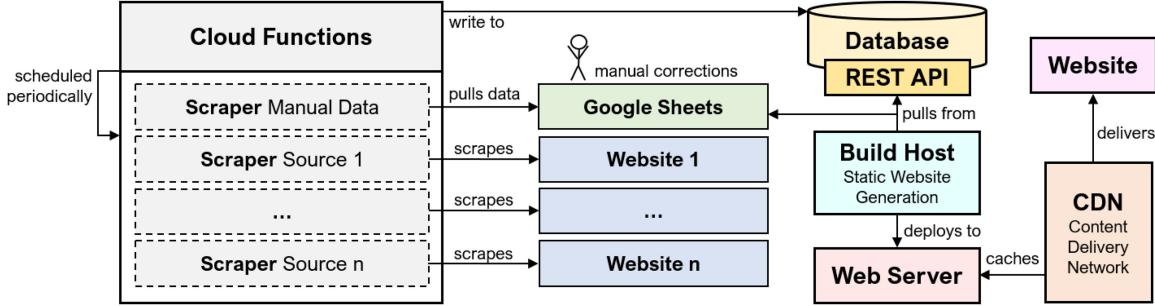


Fig. 2. Our architecture combines several components that allow periodic data collection and scaling up the website to potentially millions of users around the globe.

Table 2. Technology/Design Choices Made for Coronazaehler.de

Component	Common alternative	coronazaehler.de architecture
Scraper	<i>Centralized</i> : single point of failure, complicated to setup scheduling and alerting	<i>Cloud Functions</i> : flexible scheduling, parallelization, extensible, alerting
Hosting	<i>Self-Hosted</i> : high setup time and no integration with other services	<i>Cloud</i> : instantly available and integrates well with other services (e.g., cloud functions, DB)
Generation	<i>Dynamic</i> : high TX/RX, complex architecture	<i>Static</i> : build once and deploy globally
Availability	<i>Scaling</i> : expensive and for non-static content	<i>Caching</i> : free, requires little configuration
Database	<i>Replicated</i> : expensive, dynamic data loading	<i>Local</i> : cheap, pulled only for site generation
Editing	<i>On Database</i> : complex, custom edit frontend	<i>Table Editor</i> : fool-proof, no setup required
Frontend	<i>GIS</i> : bad mobile optimization and costs	<i>Custom</i> : responsive UI, no restrictions, free

The right column shows the technical choices made for our dashboard design, and the left column shows common alternatives and their drawbacks for the application case at hand.

data manually for counties that were not automated yet or also execute fixes of broken scrapers. The cloud functions write their results to a database that also stores the history at day precision. We serve the case data over a http-based REST API. *Google Sheets* also serves as generic database for dynamically created charts curated by hand. The generated website is automatically deployed to a webserver and pushed to a CDN (see below).

To realize cloud functions, the database, regular build scheduling, and the static webserver, we use *Microsoft Azure*. As CDN, we apply *cloudflare*. The approach is highly cost-effective even though it serves a large user load. In Table 2, we compare design alternatives that we weighed in.

### 3.3 Web Scraping and Alerting

To screen scrape case data from different websites, we defined attributes that describe identifying information on how data arranges on each page. To ensure stability, we combine HTML tags, ids, and text titles. The main advantage of the multi-attribute approach is that scanning tends to be more robust against changes, e.g., newly introduced HTML elements. Each scraper pulls the data regularly and puts it into our own, standard naming scheme and representation of case numbers. For sanity-checks of the scraped data, we set up a cache table that holds the last accepted values for every county. This data helps to detect unexpected jumps in data (e.g., due to the format of the website changing). As an example, one data source had a temporary error where the case number was unexpectedly low, which we could detect and neglect that data point. We also ensure that the relation of deaths, recovered, and overall cases match to increase stability further. Unchanged case numbers are not stored. Most scrapers periodically run every 30 minutes. If anything goes wrong, then we receive email alerts.

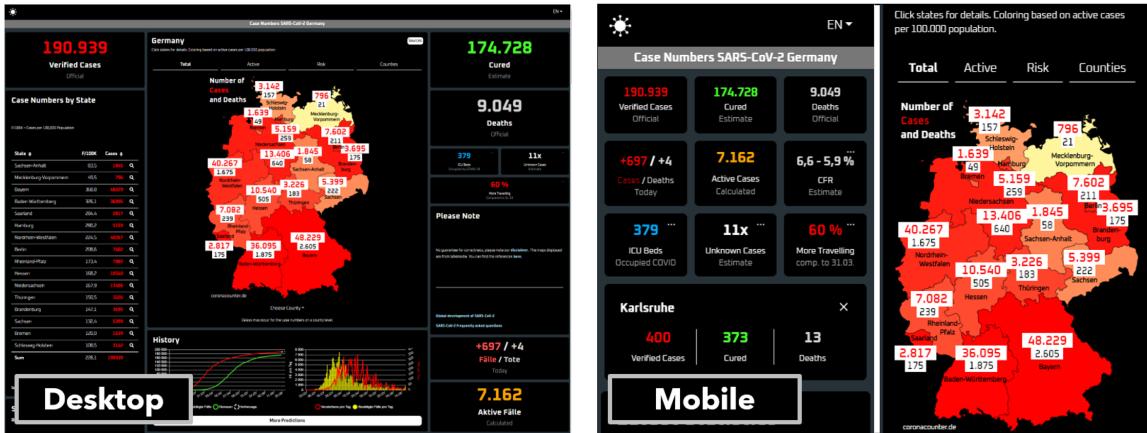


Fig. 3. The left screenshot shows the dashboard as desktop version, the two pictures on the right shows how the mobile version of (*blinded*) adapts to smaller screens.



Fig. 4. Build pipeline: Every 30 minutes, a node.js application on a Azure DevOps build agent compiles the webpage's static HTML from EJS templates and feeds it into the CloudFlare CDN.

### 3.4 Static Website Frontend

There are existing tutorials for GIS tools on how to quickly set up a COVID-19 case data dashboard [7]. However, we built the dashboard from scratch. We apply so-called *embedded JavaScript templating* (EJS) to generate a static website. For mobile and desktop layout, we use the Bootstrap framework. We utilize a mobile-first approach, where we first build the mobile version and then expand the concept on the desktop. Charts were built using *chart.js*, map visualizations with SVG files, which are markup-based images and freely available with author attribution. During website generation, we dynamically compute the coloring of the elements of the maps based on case numbers. Additionally, JavaScript events are defined for, e.g., SVG shapes to trigger button clicks. This combination enables interactive map visualizations. We also utilize *Google Sheets* for secondary computations from which we generate charts dynamically. Figure 3 shows our web interface.

### 3.5 Continuous Integration Pipeline

To regularly run build jobs of the static website, we applied a build agent hosted on *Azure DevOps*. Every 30 minutes, it executes a Node.js application that compiles the templates and fills them with the latest information. It then pushes the files to the webserver and clears the CDN's cache. Therefore, the website's newest version is available on all nodes for all users and refreshed every thirty minutes. The build pipeline is shown in Figure 4.

### 3.6 Portability and Scalability

From a technical view, the core process and applied frameworks can be transferred to other countries without changes. The technologies that we present in our architecture are all cloud-based and are available from different providers. The main challenge that remains is to set up scrapers and prepare the database. It requires substantial manual work and close monitoring, especially during the beginning of the outbreak, as formats change regularly.

A great step towards portability would be an open-source implementation of our approach, which we hope to realize in the future.

Regarding scalability, a significant advantage of our proposed architecture is that all content is static. Therefore, the content delivery network can replicate files geographically on multiple servers. Accordingly, the website can scale to millions of simultaneous users worldwide and the load time of the page is also drastically improved.

### 3.7 Incremental and Collaborative Development

Two key factors in speeding up the development process was the incremental improvement and collaborative editing of the codebase. While building the website, we focused on the most critical features first. Initially, we would only scrape the data from state agencies. We continuously expanded our scrapers, until we managed to cover all 16 federal states and their counties. We then continued adding secondary data sources step-by-step. As we started with the development early on, we could exclude the counties that did not have cases in the beginning and focus on the ones that did. Using *git* as a versioning system, we could expand our codebase according to different aspects, and automatic deployment lets us focus on building instead of worrying about how to upload the latest data. After establishing a robust case data map system, we expanded the functionality to detailed graphs and tables. Initially, we only reported the confirmed cases but quickly grew to list deaths, recovered, and more. Then, we also started building statistics on top of the collected data, e.g., forecasts. Additionally, based on requests of users that visited our website, we added secondary data (e.g., ICU capacity) on how much citizens travel.

## 4 DISCUSSION

In this section, we discuss data quality of public sources and legal aspects of scraping websites for data. We also present insights into our user base to show that the presented architecture has successfully served millions of visitors.

### 4.1 Data Quality and Heterogeneity

To ensure data quality, we only picked trustworthy sources (e.g., state ministries, established newspapers). Additionally, we set up simple statistical tests to detect outliers. This helped us to catch many error values. Finally, scraped values are also cross-verified between sources. Because of data heterogeneity, it was challenging to accommodate all the different website formats. The only way we could tackle the problem was by a combination of probabilistic scraping (e.g., also to scrape correctly if a county had a typo) and manual adaption. Unfortunately, no standard format is available for the case data.

### 4.2 Legal Perspective

A recent U.S. supreme court ruling<sup>3</sup> concluded that scraping publicly available web resources is legal. Therefore, the proposed architecture might be applied without fearing repercussions. However, for some countries, e.g., Germany, constraints exist. First, technical barriers to protect against web scraping (such as CAPTCHAs) may not be circumvented. Second, if the data requires the login of a user, then it is not considered public. And finally, if the data is copyrighted (e.g., images), then such data may not be scraped and replicated without approval.

Overall, there may be various barriers to sharing public health data, which were reviewed systematically in the past by Van Panhuis et al. [14]. We recommend in general that developers implement their scraping strategies responsibly, e.g., by reducing the number of website calls per time interval to a reasonable number that suits the application's purpose. When in doubt, agreements with the original data provider should be considered. In the specific use-case of this work, we are mostly looking at non-commercial provisioning of government

---

<sup>3</sup>[https://www.supremecourt.gov/DocketPDF/19/19-1116/146325/20200625122719793\\_1%20-%20FINAL%20hiQ%20cert%20opp.pdf](https://www.supremecourt.gov/DocketPDF/19/19-1116/146325/20200625122719793_1%20-%20FINAL%20hiQ%20cert%20opp.pdf).

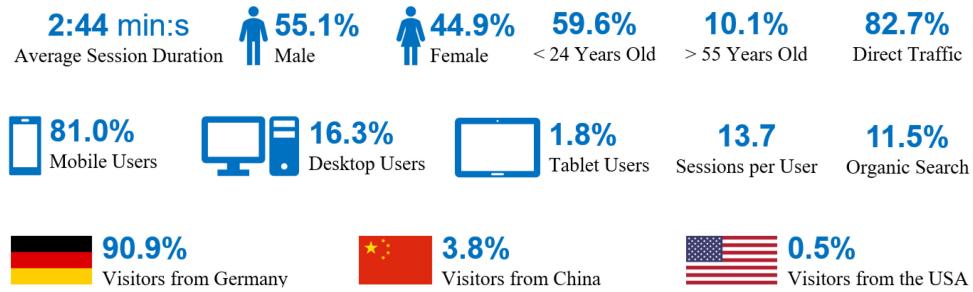


Fig. 5. Use habits of *coronazaehler.de* collected from March 1st till June 19, 2020 from around 500,000 users.

collected public health data, which is open to some extent in many legislations anyway. If an architecture like the one presented in this work is employed by official bodies themselves, then the problem of data access and integration should be even less relevant.

#### 4.3 Usage Habits

To underline the success and scalability of the proposed architecture and overall system, we present our website's usage habits. So far, *coronazaehler.de* had around 5 million unique visitors and about 70 million sessions. Out of those visitors, approximately 10% opted-in for *Google Analytics* running on the website. The numbers we present in Figure 5 are based on those visitors and were collected between March 1 and June 19, 2020.

The average session duration per user was 2 minutes and 44 seconds. About 50% of all visitors returned to the website at least once. Out of all visitors, 55.1% were male and 44.9% female; 59.6% of all visitors were under the age of 24; only 10.1% were older than 55 years; 81.0% of all visitors used a mobile phone, 16.3% a desktop device, and 1.8% a tablet. The majority of visitors was from Germany (90.9%), followed by China (3.8%) and the United States (0.5%); 82.7% of all visitors came directly to our website and 11.5% by organic search. Surprisingly, only 4.1% of all visitors came through social media and 1.6% by referral; 27% of all visitors had at least 101 sessions on the website and 51% had between 9 and 100 sessions. On average, every user had 13.7 sessions. Around 21% of all visitors had eight or less sessions. Out of all sessions, 5% include a visit to the page that contains detailed statistics. Every day during the week user numbers rapidly decline after 12 a.m. During the week, visitors start to come in at around 6 a.m., whereas on Saturdays and Sundays the numbers of visitors starts to rise later after around 7 a.m. User numbers remain stable after around 10 a.m. and tend to have small peaks after lunch and also after dinner.

## 5 CONCLUSION: CHALLENGES AND LESSONS LEARNED

Responsibly provisioning a case data gathering system in limited time is a challenging task, especially in times of drastic changes and widespread unsettledness as in a global pandemic. In this article, we have presented the development process and system architecture of a lightweight COVID-19 dashboard for the general public. The real-world deployment has shown that our design is very well equipped to deal with the challenges and the dynamics presenting themselves: We were able to quickly adapt to changing circumstances, interfaces and user requests, while still maintaining a highly automated, cost-effective and scalable platform. While we have discussed many lessons learned in this article, we find that three key takeaways that we discovered to be relevant during our development process stand out:

- **There is no free lunch in automated data collection**, as gathering case numbers from 16 federal states and 400+ counties still required substantial programming effort. Especially during the outbreak of the pandemic website formats would change and destroy our scrapers' logic. A major advantage was the modular approach of the cloud functions, which ensured that other parts of the system were not affected.

- **Building a critical technology tool “on top of” the user requires high responsibility**, especially in the early development process where no alerting was set up and malfunctions did occur, which created insecurities on the users’ side. A key success factor in controlling error states was the manual override component, which we used extensively until acceptable system stability was achieved.
- **Selecting the wrong technology stack can have a massive impact on costs**, for example, the initial content delivery network that we selected was not focused on delivering static websites. By switching our CDN provider, we could reduce our costs from around \$300 per day to \$0 for that component of our system.

The approach that we have taken focuses on the fast provision of citizen-focused dashboards with high timely accuracy and data collected from publicly available but heterogeneous sources and formats. More prominently than our solution, the dashboard by the Johns Hopkins University [5] has shown how important such information systems are to understand the dynamics of an outbreak. Therefore, we believe that our technical insights provide key aspects for future crisis management teams, government bodies and the general public in similar situations.

## REFERENCES

- [1] AG Keim-University Konstanz. 2020. CoronaVis by DBVIS Group. Retrieved from <https://coronavis.dbvis.de/de/>.
- [2] Boston Children’s Hospital. 2020. Healthmap - Outbreaks Near Me. Retrieved from <https://healthmap.org/en/>.
- [3] Matthias Budde, Andrea Schankin, Julien Hoffmann, Marcel Danz, Till Riedel, and Michael Beigl. 2017. Participatory sensing or participatory nonsense? Mitigating the effect of human error on data quality in citizen science. *Proceedings of the ACM Conference on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT'17)* 1, 3 (2017), 1–23.
- [4] Qingyu Chen, Alexis Allot, and Zhiyong Lu. 2020. Keep up with the latest coronavirus research. *Nature* 579, 7798 (2020), 193–193.
- [5] Ensheng Dong, Hongru Du, and Lauren Gardner. 2020. An interactive web-based dashboard to track COVID-19 in real time. *Lancet Infect. Diseases* 20, 5 (2020), 533–534.
- [6] James Hadfield, Colin Megill, Sidney M. Bell, John Huddleston, Barney Potter, Charlton Callender, Pavel Sagulenko, Trevor Bedford, and Richard A. Neher. 2018. Nextstrain: Real-time tracking of pathogen evolution. *Bioinformatics* 34, 23 (2018), 4121–4123.
- [7] Health GeoLab Collaborative. 2020. ArcGIS Solution Coronavirus Case Dashboard. Retrieved from [https://healthgeolab.net/DOCUMENTS/HGLC\\_Coronavirus\\_Case\\_Dashboard\\_starter\\_kit.pdf](https://healthgeolab.net/DOCUMENTS/HGLC_Coronavirus_Case_Dashboard_starter_kit.pdf).
- [8] Maged N. Kamel Boulos and Estella M. Geraghty. 2020. Geographical tracking and mapping of coronavirus disease COVID-19/severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) epidemic and associated events around the world: How 21st Century GIS technologies are supporting the global fight against outbreaks and epidemics. *Int. J. Health Geogr.* 19 (2020). Issue 1. DOI: <https://doi.org/10.1186/s12942-020-00202-8>
- [9] Raina M. Merchant, Stacy Elmer, and Nicole Lurie. 2011. Integrating social media into emergency-preparedness efforts. *New Engl. J. Med.* 365, 4 (2011), 289–291.
- [10] /r/covidmapping. 2020. COVID Map Global. Retrieved from <https://covidmap.global/>.
- [11] Axel Schulz, Heiko Paulheim, and Florian Probst. 2012. Crisis information management in the web 3.0 age. In *Proceedings of the International Conference on Information Systems for Crisis Response and Management (ISCRAM'12)*.
- [12] Catrin Sohrabi, Zaid Alsafi, Niamh O'Neill, Mehdi Khan, Ahmed Kerwan, Ahmed Al-Jabir, Christos Iosifidis, and Riaz Agha. 2020. World health organization declares global emergency: A review of the 2019 novel coronavirus (COVID-19). *Int. J. Surg.* (2020).
- [13] Kristian Thorlund, Louis Dron, Jay Park, Grace Hsu, Jamie I. Forrest, and Edward J. Mills. 2020. A real-time dashboard of clinical trials for COVID-19. *Lancet Digital Health* 2, 6 (2020), e286–e287.
- [14] Willem G. van Panhuis, Proma Paul, Claudia Emerson, John Grefenstette, Richard Wilder, Abraham J. Herbst, David Heymann, and Donald S. Burke. 2014. A systematic review of barriers to data sharing in public health. *BMC Public Health* 14 (2014). Issue 1.
- [15] World Health Organization (WHO). 2020. WHO Coronavirus Disease (COVID-19) Dashboard. Retrieved from <https://covid19.who.int/>.

Received June 2020; revised September 2020; accepted October 2020