

IoT Middleware for Water Management [†]

Matej Senožetnik ^{1,2,*}, Zala Herga ^{1,2}, Tine Šubic ¹, Luka Bradeško ¹, Klemen Kenda ^{1,2},
Kristina Klemen ³, Petra Pergar ³ and Dunja Mladenec ^{1,2}

¹ Artificial Intelligence Laboratory, Jožef Stefan Institute, 1000 Ljubljana, Slovenia; zala.herga@ijs.si (Z.H.); tine.subic@ijs.si (T.Š.); luka.bradesko@ijs.si (L.B.); klemen.kenda@ijs.si (K.K.); dunja.mladenec@ijs.si (D.M.)

² Jožef Stefan International Postgraduate School, 1000 Ljubljana, Slovenia

³ Ljubljanski Urbanistični zavod d.d., 1000 Ljubljana, Slovenia; kristina.klemen@luz.si (K.K.); petra.pergar@luz.si (P.P.)

* Correspondence: matej.senozetnik@ijs.si; Tel.: +386-51-242-602

[†] Presented at the 3rd EWaS International Conference on “Insights on the Water-Energy-Food Nexus”, Lefkada Island, Greece, 27–30 June 2018.

Published: 31 July 2018

Abstract: Groundwater management is important for all urban systems. Thus, data needs to be available on request for various decision-makers and stakeholders. This article presents conceptual and implemented framework for collecting, analyzing and sharing of groundwater data for various purposes. It allows controlled access to data that is continuously collected from different feeds and transformed into a common format. With this approach, the latest data as well as historical records are always available for real-time queries and further analysis. The proposed system can be extended to cover other areas of data collection in the future.

Keywords: data gathering infrastructure; internet of things; data preprocessing; groundwater; groundwater data

1. Introduction

Recently, more and more previously private or proprietary data are becoming available via openly accessible online sources. One type of this data is information on groundwater level, which is essential for providing sustainable water management in urban areas. However, many of these data sources are still evolving and rarely can they truly be described as open data in terms of technical availability and semantic annotation. For scientists, practitioners and other stakeholders open and fast access to the data is crucial in order to better understand the groundwater issues and to make better-informed decisions. Additionally, data-driven modeling requires usage of additional data sources such as weather data and predictions, social data and urban development data.

The contribution of this paper is a conceptual architecture and a real-world system implementation for groundwater monitoring, which consists of four main components: Data retrievers, Data collector service, API Management service and Watchdog service. Data retrievers are implemented as scripts that periodically poll the data from raw data sources, filter and then transmit them to the Data collector service. Its job is to apply pre-processing and store the data into an internal database. API Management service provides data access control via a graphical user interface, which provides a simple user and access management toolkit. Watchdog service is used for monitoring retriever scripts for failures and non-responsiveness, upon which the Watchdog will attempt to restart the scripted task, as well as alert the system administrator via email.

The remainder of the paper is organized as follows. Section 2 gives an overview of the related work. In Section 3 we provide a detailed description of the architecture and interaction between the components.

2. Materials and Methods

Half of the world's megacities are groundwater-dependent and over 40% of water supply in Europe is based on water pumped from aquifers beneath urbanized areas [1]. There is a need for comprehensive groundwater management in terms of underground structures, quantity, quality and temperature [2,3].

For sustainable urban groundwater management, it is essential to keep the data easily available to the users including a user-friendly application for processing, visualization and dissemination of groundwater monitoring information. Therefore, relevant stakeholders can make better-informed decisions regarding groundwater management: not only about water supply but also about other important urban issues. The interaction of groundwater with other urban systems is well recognized and is increasingly important on the everyday city agenda, for example in consideration of base flow provision to urban and peri-urban rivers (blue networks), flood risk, management of blue-green infrastructure (e.g. sustainable drainage systems), adverse effects on underground infrastructure, control of underground construction, and impacts of industrial legacy on water quality. These different examples highlight the range of urban groundwater processes that operate at different temporal and spatial scales [4]. The interaction of several domains, stakeholders and growing demands to reach the goal of sustainable groundwater management have forced the expert practitioners to find and research new methods and ways to collect, analyze and model various data about groundwater within the urban areas.

The amount of available digital datasets in the field of groundwater resources management is increasing, especially for geospatial data, produced as part of numerous studies and research from various fields, stakeholders and in various countries. However, groundwater and related data are usually not shared among the stakeholders. Urban groundwater information connected with underground structures (like subway lines, sewer systems, heating conduits, etc.) changes continuously [5]. Groundwater data must be based on geospatial data. Significant efforts have been made over the last three decades for the development of the sophisticated geospatial system to produce geo-decisional information [6]. The effectiveness of the decision-making process is based on the provision of relevant information and adapted analysis tools. Because of the disparity of data and the lack of interoperability of existing information systems, they can often become underused or contain outdated and insignificant data. Therefore, data integration is the key to building a comprehensive decision support system for groundwater [7]. Traditional groundwater balancing addresses direct groundwater processes like recharge, interflow or discharge. It is also important to include data that addresses anthropogenic activities like surface sealing, construction and behaviors like water saving and policy development. As a result, more holistic evaluation of urban water management can be defined [4]. Some holistic groundwater models have already been developed in recent years [8,9]. For sustainable urban groundwater analysis and modeling, it is therefore essential to keep the data easily available to the users via the user-friendly interface.

Driven by the need for integration of all available data from distributed sources, many researches are focusing on the data integration process. Data integration process should be able to acquire a set of heterogeneous data residing at different sources and provide the end user with an integrated and reconciled view of data in order to extract explicit information at different levels of the decision-making process [10]. Data integration should also include standardization of data schema (including attributes that are common across different datasets). Sometimes, the full source-specific schema can be provided as well, but such data will be less useful since data consumers will have to anticipate additional information and either fill in the gaps or ignore such records. In our system, data integration is achieved with Retriever and Collector modules that transform data into the common format for database storage.

Moreover, data integration also gives users the opportunity to make an integrated analysis of prior information and a single geographical reference of spatial data. Furthermore it enables the specialization of data through relationships with geo-referenced geometric entities and mutual enrichment with semantic and geometric properties [11]. Integration of geographic data should

include much more than a simple overlay data in a geographic information system (GIS). It should be oriented towards exchange between individual components in various information systems [11].

2.1. Proposed Approach and Framework Description

Figure 1 shows an overview of the proposed architecture that consists of four main components: Retrievers, Collector, API Management and Watchdog. Retrievers are smart web or Internet of Things (IoT) agents that poll various external data sources (groundwater data, weather data and forecasts, sensor and historical or social data collections, etc.) periodically to ensure up-to-date data retrieval. Retrieved data is transformed into a common JSON format and stored in a local file or forwarded to a remote collection service. Collection service (or Collector) receives the data via HTTP POST requests and applies basic rule-based data pre-processing, including handling missing values, and may additionally transform the data into a standardized schema, which might not be followed by all the retrievers due to technical or legacy issues. Finally, the data is stored in a non-SQL database system (in our case Mongo DB) with separate real-time and historical data collections of a particular data type. Real-time and complete historical data are exposed via REST-like API endpoints. API Management Console can be used to ensure execution of user-required privacy policies. Finally, Watchdog module offers basic monitoring, diagnostics and a high-level overview of the system via a graphical web interface. The system can detect faulty retrievers, missing data and can significantly contribute to improving the system health, performance and administrator awareness. The proposed infrastructure can also be used in other domains, such as traffic management, weather observation and urban development.

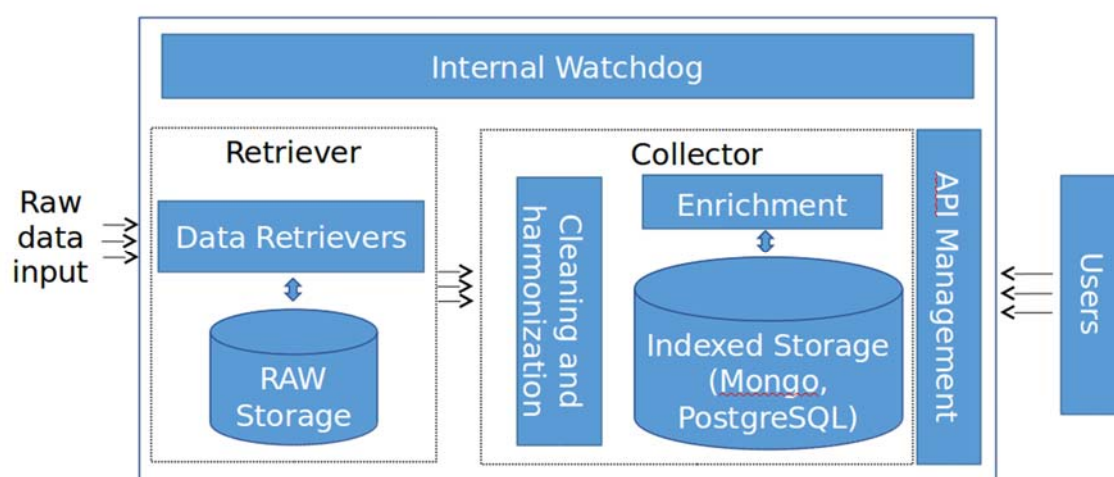


Figure 1. The architecture of our system for IoT middleware for water management.

2.1.1. Data Retrievers

Retrievers are services which extract data feeds by periodically polling an external data source for updates, transform polled data into a common format and then store them by either writing to a file or forwarding them to a collection service. Retrievers are extended from an abstract retriever object. This means that only small contributions are needed to modify or implement a new retriever when new data feeds become available. Additionally, all changes to the abstract retriever object are instantly applied also to particular retrievers. Retrievers handle all data as JSON/XML objects.

Some retrievers also implement push mechanisms, which means that data sources themselves send the new data as it is generated.

2.1.2. Collector

Collector is a dedicated service backed by a Mongo database, containing real-time data, as well as a historical database (as a separate collection). This separation was done to optimize real-time data collection. The server uses Java servlets for API endpoints for storing and retrieving the data. The data can be pushed to collector service by retrievers or directly by external sources such as sensors. Collector implements another layer for data transformation, since external sources may send non-standardized JSON. Transformed records are finally written to the appropriate collection in Mongo database. Historical collections store all the pushed records, while the real-time collection records are merely updated with new values to provide the most accurate information.

Finally, the data can then be queried via API endpoints, which allow different filters (ex: filter by time, location...) for easier access.

2.1.3. Internal Watchdog

Data retrievers can fail due to different reasons—sometimes host machines would shut down because of power failures or restart to install some updates and consequently shut down the retriever; there could be errors when the retriever is trying to connect to the data providers, the data could be corrupted, etc. In some cases, these failures are noticed immediately whereas in other situations it takes a while before anyone notices that the data is not being retrieved. With a large number of different retrievers and prolonged maintenance times we recognized the need for a system monitoring (watchdog) service. Watchdog service provides monitoring of the system as well as automatic reparation and alerting the user if the problem persists. The watchdog service is implemented in Java programming language and runs on top of Apache Tomcat application server.

2.1.4. API Management

The API management tool is composed of multiple components. User information and API endpoint metadata are stored in an internal NoSQL MongoDB database. An independent web server provides dynamic routing and forwarding for these endpoints. It also runs a graphical interface, accessible via the browser that can be used for managing user authentication and API accessibility, for both internal and forwarded endpoints. The rest of this subsection describes each of the components.

Graphical Web Interface

The service uses an online graphical user interface, accessible through the login screen, which allows the administrator to manage users and APIs. The user management section (see Figure 2) provides a form for adding new users, a list of existing users, their access rights and other information. Through this interface, we can also delete users, check their authentication tokens and select which API endpoints they can access.

API management section (see Figure 3) provides a form for adding APIs from external services to master access control list. The user must specify full API URL and API service hostname, as well as enter a project name, which is an internal designation in our API management tool, used for generating dynamic API endpoint URLs. Optionally, API notes may be added, such as the description of parameters or explanation of returned results.

As seen in the bottom panel of Figure 3, users can view the list of APIs entered into the system. Here, it is also possible to delete APIs, check notes or edit API properties, such as URLs and project names on a case-by-case basis. Each API status can be set to *private*, meaning end users need authentication token and appropriate access rights to call this API, or *public*, also allowing anonymous users without tokens access to this endpoint as well. APIs can also be fully disabled, for example when underlying service is in maintenance mode or unexpectedly unavailable.

Add user

Username IJS Optimum

Email john@example.com

Organization Org

Select APIs

Submit

User listing

User	Organization	Email	Select API	Token	Delete	Rel Tok
Luka Bradesko	IJS	luka.bradesko@ijs.si	Select APIs	Show token	×	↺
Klemen Kenda	W4C	klemen.kenda@ijs.si	Select APIs	Show token	×	↺
Matej S	W4C	matej.senozetnik@ijs.si	Select APIs	Show token	×	↺
W4C Project	W4C		Select APIs	Show token	×	↺
Timon	IJS	tine.subic@ijs.si	Select APIs	Show token	×	↺

Export JSON
Apply changes

Figure 2. API Management tool web interface—User management.

Add API route

API URL http://atena.ijs.si:8080/CollectorAPIServer/undergroundWater

Service Host http://atena.ijs.si:8080/CollectorAPIServer

Project w4c

Slovenian underground water data.

Save Cancel

API Access Controls

Edit	API Route	Public	Active	Project	Notes	Delete
<input checked="" type="checkbox"/>	opendata/promet/burja/	Yes	Enabled	opendata		×
<input checked="" type="checkbox"/>	opendata/promet/burjaznaki/	Yes	Disabled	opendata		×
<input checked="" type="checkbox"/>	opendata/promet/events/	Yes	Enabled	opendata		×
<input checked="" type="checkbox"/>	opendata/vreme/report/	Yes	Enabled	opendata		×
<input checked="" type="checkbox"/>	uninova/get_weather_data	Yes	Enabled	uninova		×
<input checked="" type="checkbox"/>	w4c/pumpSensors	Yes	Enabled	w4c		×
<input checked="" type="checkbox"/>	w4c/undergroundWater	Yes	Enabled	w4c		×
<input checked="" type="checkbox"/>	w4c/weather	Yes	Enabled	w4c		×

Export JSON
Import JSON

Figure 3. API Management tool web interface—API management.

User Authentication

Upon being entered into the database in the web interface, each user has issued a uniquely generated authentication token (this token can be re-generated through the web interface by administrator). Two possible means of authenticating are possible. The primary method requires

users to pass their authentication token to the server in HTTP request header while making a request to our service. Upon receiving the request, server checks passed security token against database entries and then forwards the request along with appropriate parameters to requested external API service. If API endpoint is public, no authentication is necessary. This method allows actual API services to live isolated on internal networks, so users never actually come into direct contact with them. The secondary method does not actually route requests through our service but rather allows external APIs to call our authentication service and check, whether a user with a provided authentication token has been authenticated for this API or not.

3. Results

The proposed approach has been applied to groundwater data and traffic data collection. The results for the traffic data are shown in Table 1. There are 30 data retrievers collecting the traffic data in several cities in Europe. The presented data has been captured from our Watchdog portal on the 9th of May 2018.

Table 1. The frequency of records per hour for a single retriever script.

Name of the Dataset	Records/h
Bus Retriever (Ljubljana)	5828
Taxi Coordinates (Ljubljana)	4863
Motorhome Coordinates (Europe)	194
Parking Data OD (Ljubljana)	316
Collector Parking (Ljubljana and Vienna)	316
Collector Loop Sensors (Slovenia)	12,170
Loop sensors (Slovenia)	12,170
Collector Bicycle (Ljubljana and Vienna)	7912
Shared Bicycle (Ljubljana)	879
Shared Bicycle (Vienna)	6981
Truck Coordinates (Croatia and UK)	31,533
Traffic Events (Slovenia)	1
Wind (Burja) data (Slovenia)	10
Road cameras (Slovenia)	8668
Tweets (Slovenia)	1

Due to the limited number of publicly available water-related data and time constraints, we could not collect a large amount of data regarding the water domain. Table 2 shows how much data are collected for water domain on a daily basis. So far we have collected data for groundwater from Slovenia, pump sensors from Skiathos and weather for both Skiathos and Ljubljana. Each implemented sensor updates once per day, our system can, however, support much higher frequencies. Groundwater from Slovenia is collected from 518 stations divided into 28 regions and data from some stations are collected since 1960. Pump sensors from Skiathos are collected on the daily basis from three pumps. We collected data on daily consumption of water and pump working time.

Table 2. Frequency of records per day for a single retriever script.

Name of the Dataset	Records/Day
Groundwater levels	518
Pump sensors	4
Weather (Ljubljana and Skiathos)	30

4. Discussion

In this paper we proposed a framework, which can be used for monitoring groundwater, weather and another kind of temporal data. The framework consists of four main components:

Retriever, Collector, API Management Service and Watchdog. The main benefit of this system is that it supports ingestion of any spatiotemporal data with additional meta-information. In the water management domain, this system contributes to consistency, standardization and data sharing, which means that stakeholders can spend more time on data analysis instead of data retrieval and manipulation. The system was extensively tested and deployed in multiple interconnected instances. Due to loosely coupled architecture, where all the components interact via REST based APIs, modules can be built with arbitrary tools and programming languages. This is most important for retrievers, since certain types of data collection and processing may depend on functionalities in language-specific libraries. Modular architecture also contributes to the robustness of the system, since particular sub-modules do not affect any other components in the system. They can also be updated independently.

In the future this work could be extended in multiple directions. Firstly, the system could be improved by adding coverage for new data domains. Secondly, basic data analytics could improve preprocessing with data-cleaning (outlier detection and missing data imputation) algorithms.

Author Contributions: L.B. and K.K. designed the architecture, M.S. developed and adopted for general usage Retrievers and Collector with the collaboration of T.Š., Z.H. and L.B. developed API Management and Z.H. developed Watchdog. The paper was written as collaboration of all the authors, coordinated by M.S., K.K. and P.P. who influenced this work by their rich domain knowledge. D.M. contributed to the underlying research and supervised the presented work.

Acknowledgments: This work was supported by the Slovenian Research Agency and the ICT program of the EC under project OPTIMUM (H2020-MG-636160) and WATER4CITIES (H2020-MSCA-RISE-2016). The authors would like to thank Anja Polajnar for proofreading of the manuscript.

References

1. Wolf, L.; Morris, B.; Burn, S.U. *Water Resources Toolbox: Integrating Groundwater into Urban Water Management*; IWA: London, UK, 2006.
2. Attard, G.; Winiarski, T.; Rossier, Y.; Eisenlohr, L. Review: Impact of underground structures on the flow of urban groundwater. *Hydrogeol. J.* **2016**, *24*, 5–19.
3. Attard, G.; Rossier, Y.; Winiarski, T.; Eisenlohr, L. Urban underground development confronted by the challenges of groundwater resources: Guidelines dedicated to the construction of underground structures in urban aquifers. *Land Use Policy* **2017**, *69*, 461–469.
4. Bricker, S.H.; Banks, V.J.; Galik, G.; Tapete, D.; Jones, R. Accounting for groundwater in future city visions. *Land Use Policy* **2017**, *69*, 618–630.
5. Gaitanaru, D.; Priceputu, A.; Gogu, C.R. Geospatial Data Management Platform for Urban Groundwater. *Geophys. Res. Abstr.* **2012**, *14*, 10174.
6. Michael, F. Geographic information systems and science: today and tomorrow. *Ann. GIS* **2009**, *15*, 3–9.
7. Laraichi, S.; Hammani, A.; Bouignane, A. Data integration as the key to building a decision support system for groundwater management: Case of Saiss aquifers. *Groundw. Sustain. Dev.* **2016**, *2–3*, 7–15.
8. Velasco, V.; Tubau, I.; Vázquez-Suñé, E.; Gogu, R.; Gaitanaru, D.; Alcaraz, M.; Serrano-Juan, A.; Fernández-García, D.; Garrido, T.; Fraile, J.; Sanchez-Vila, X. GIS-based hydro geochemical analysis tools (QUIMET). *Comput. Geosci.* **2014**, *70*, 164–180.
9. Boukhemacha, M.A.; Gogu, C.R.; Sarpescu, I.; Gaitanaru, D.; Bica, I.A. Hydrogeological conceptual approach to study urban groundwater flow in Bucharest city, Romania. *Hydrogeol. J.* **2015**, *23*, 437–450.
10. Cali, A.; Calvanese, D.; De giacom, G.; Lenzerini, M. Data integration under integrity constraints. *Inf. Syst.* **2004**, *29*, 147–163.
11. Butenuth, M.; Gösseln, G.; Tiedge, M.; Heipke, C.; Lipeck, U.; Sester, M. Integration of heterogeneous geospatial data in a federated database. *ISPRS J. Photogram. Remote Sens.* **2007**, *62*, 328–346.

