Utah Water Research Laboratory
UtahStateUniversity

Advancing open source cyberinfrastructure for collecting, processing, storing and accessing high temporal resolution residential water use data

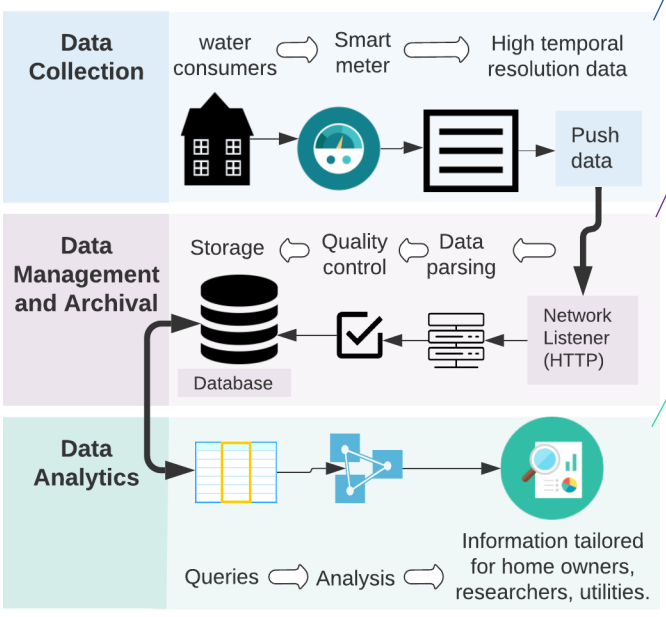
Camilo J. Bastidas Pacheco (camilo.bastidas@usu.edu) , Jeffery S. Horsburgh (jeff.horsburgh@usu.edu), Joseph C. Brewer, Robb J. Tracy, Juan Caraballo

Funding
Award Number: 1552444

1. Motivation

Collecting and managing high temporal (< 1 minute) resolution water use data is challenging due to cost and technical considerations (data volume and frequency).

2. Overall Architecture



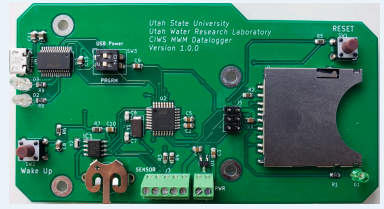
All the components developed are available for reuse.

GitHub HYDROSHARE

3. Implementation

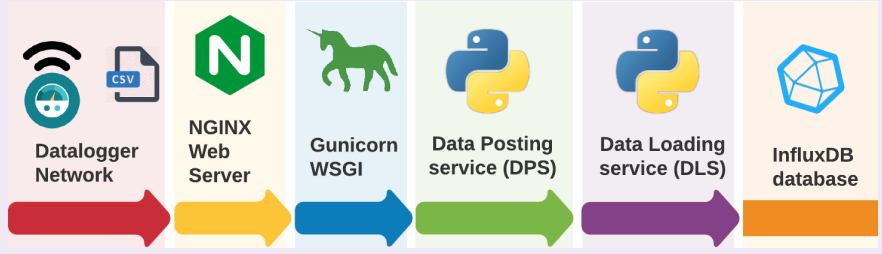
Data Collection

- User selectable temporal resolution
- Works on existing meters to collect data
- Generates daily files
- Low cost (< $150)



Data Management and Archival

- Modular
- Open source technologies
- Implemented in a Linux server

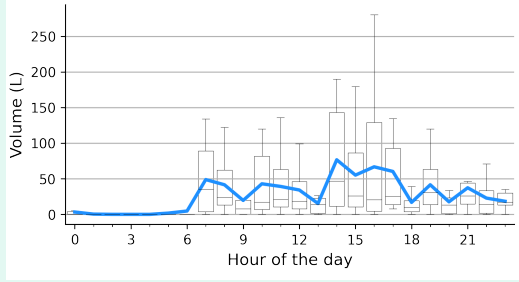


Data Analytics

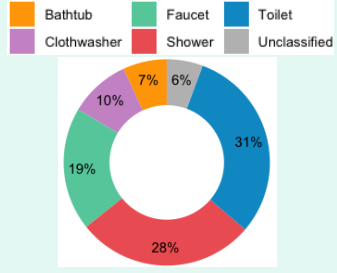
- Application Programming Interface (API) for database interaction
- Client software programs for retrieving, visualizing, and analyzing data

jupyter

Hourly average and variability



End uses volume distribution

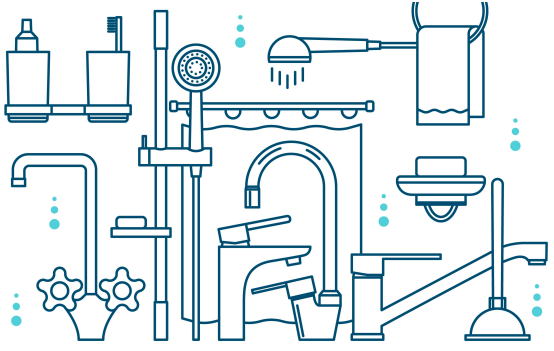


Scalability and Performance

- Simulated 500 dataloggers simultaneously sending 4-seconds daily data files to the system
- 100% Transmission success with no data loss
- Time required to post, parse, and load data into the database: less than 5 minutes

Introduction

- Achieving higher efficiency in urban water management and planning requires understanding of how water is used at the household level.



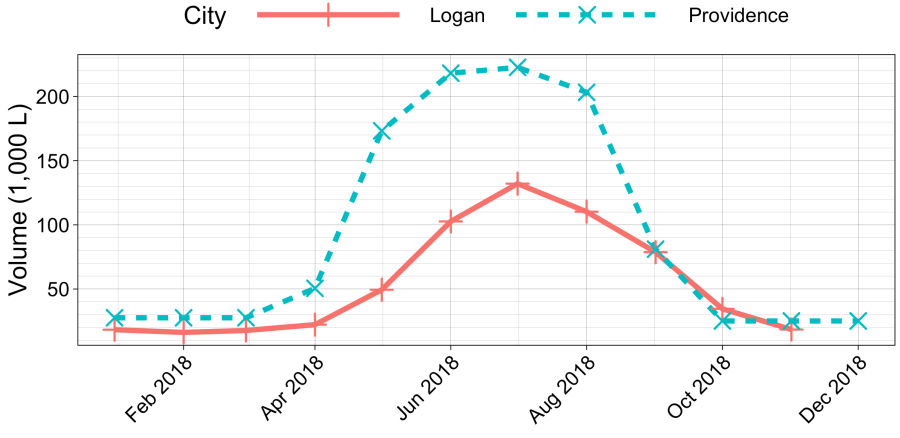
- Daily patterns in consumption, potential for water savings and distribution of water use across end uses are essential inputs to water demand estimation, leak identification, and design of water demand programs.

Introduction

- In the United States, meters are typically read monthly or quarterly.



- Our ability to characterize water demand is limited by the temporal resolution of the data collected.



Average Monthly water use for residential customers in Logan and Providence, Utah, 2018.

Introduction

- Higher resolution data can increase the accuracy of peak demand estimation and reduce leak volumes that can go undetected.



- Sub-minute resolution data is required to record and quantify end uses of water that have short duration



Introduction

- Obtaining this higher temporal resolution data at a scale larger than a few houses presents several challenges in terms of data collection, storage, management, and processing.



- Cyberinfrastructure can help solve data management challenges and enable more widespread collection of higher temporal resolution water use data for utilities and researchers.

The term "cyberinfrastructure" integrates hardware and software tools, as well as data networks (NSF, 2007).

Introduction

We present an architectural design and a complete prototype implementation of a Cyberinfrastructure for Intelligent Water Supply (CIWS) with components for:

a) data collection;

b) data communication;

c) storage, management and archival; and d) visualization and analysis.

Methods

The CIWS architecture and our prototype implementation were designed to address the following requirements:

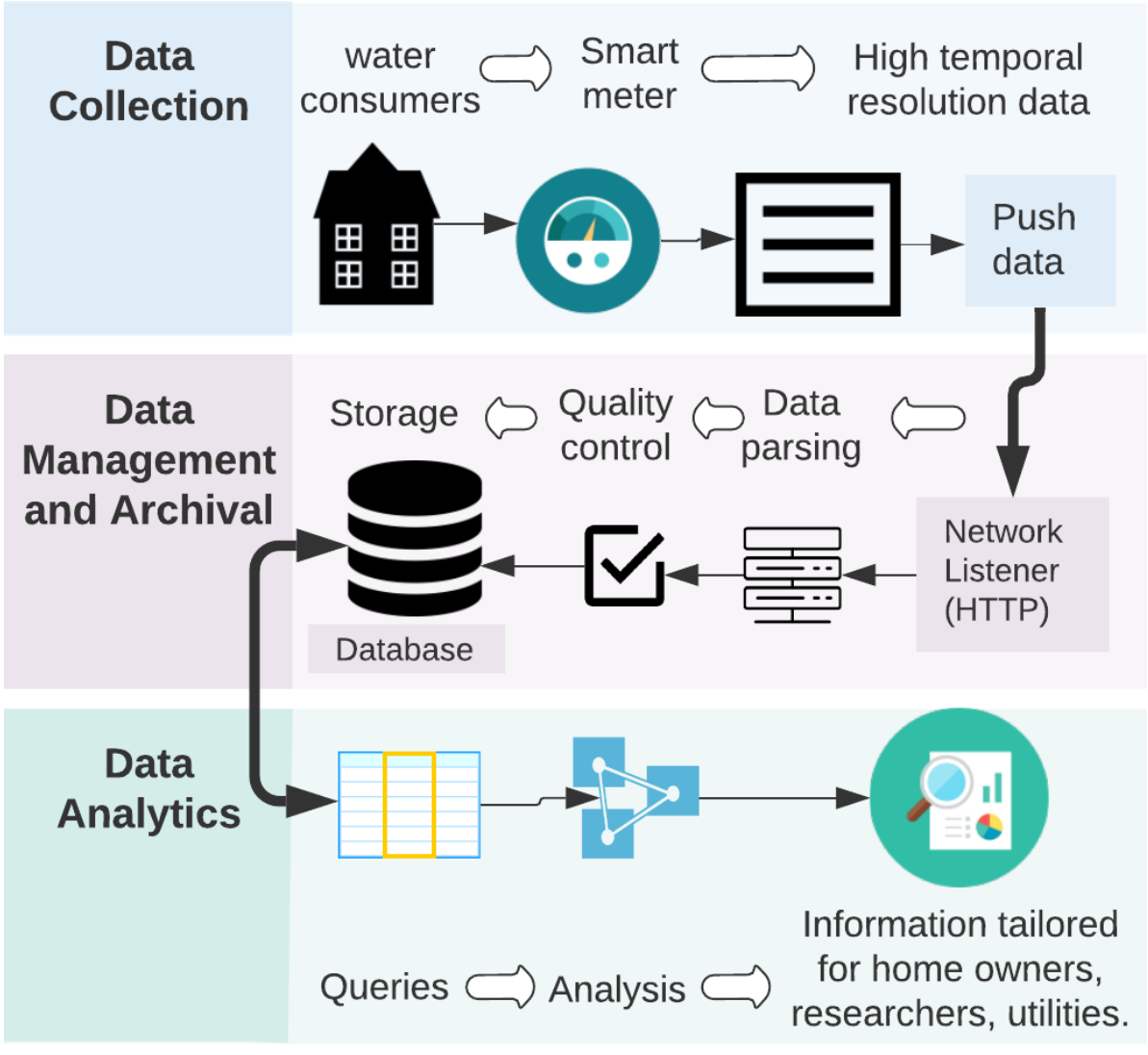
- An open architecture that could be implemented using a variety of technologies;
- Open source software development to facilitate its deployment and use by other users, reduce costs, and provide a platform for future improvement by others while advancing financial feasibility of larger scale implementations;
- A modular design, so each component of CIWS can be used, or advanced, independently;
- Accept input data from different meters and measurement devices (sensors) to address heterogeneity in urban water meter technology;

Methods

The CIWS architecture and our prototype implementation were designed to address the following requirements:

- Capacity to manage “push” data retrieval from metering devices and storing of data in a centralized server;
- Scalable to accommodate a large data volume while remaining responsive to queries for subsets of time series data of varying sizes;
- Support production of analysis and insights that meet the needs of different audiences.

CIWS design


Data Collection
water consumers
Smart meter
High temporal resolution data
Push data
Data Management and Archival
Storage
Quality control
Data parsing
Network Listener (HTTP)
Database
Data Analytics
Queries
Analysis
Information tailored for home owners, researchers, utilities.

Implementation

Push Based Data Collection for Single Family Residential Homes

- Water use in single family residential homes is quantified, to a large extent, using analog, positive displacement water meters.

- The volume of water that has passed through the meter is usually the only variable recorded by this type of meter.

Data collection

- We built a low-cost (≈$150) monitoring system for collecting high temporal resolution residential water use data without disrupting the operation of commonly available water meters.

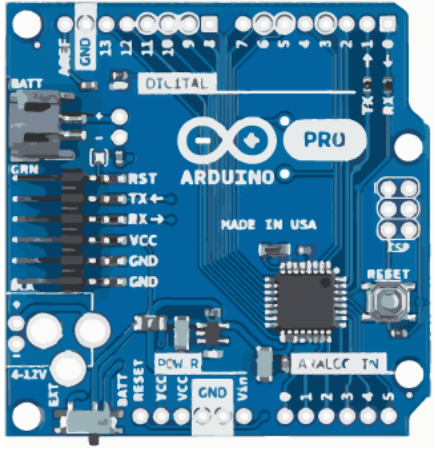
- The datalogger is installed on top of analog, magnetically driven, positive displacement, residential water meters.

- Variable time resolution interval for data collection.

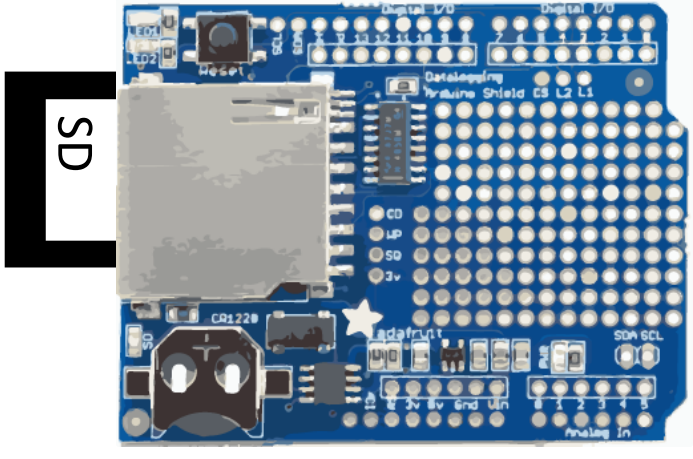


Data collection

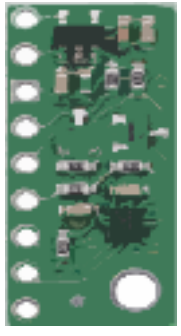
The system couples:



An Arduino Pro microcontroller board



A Datalogging shield customized for this specific application

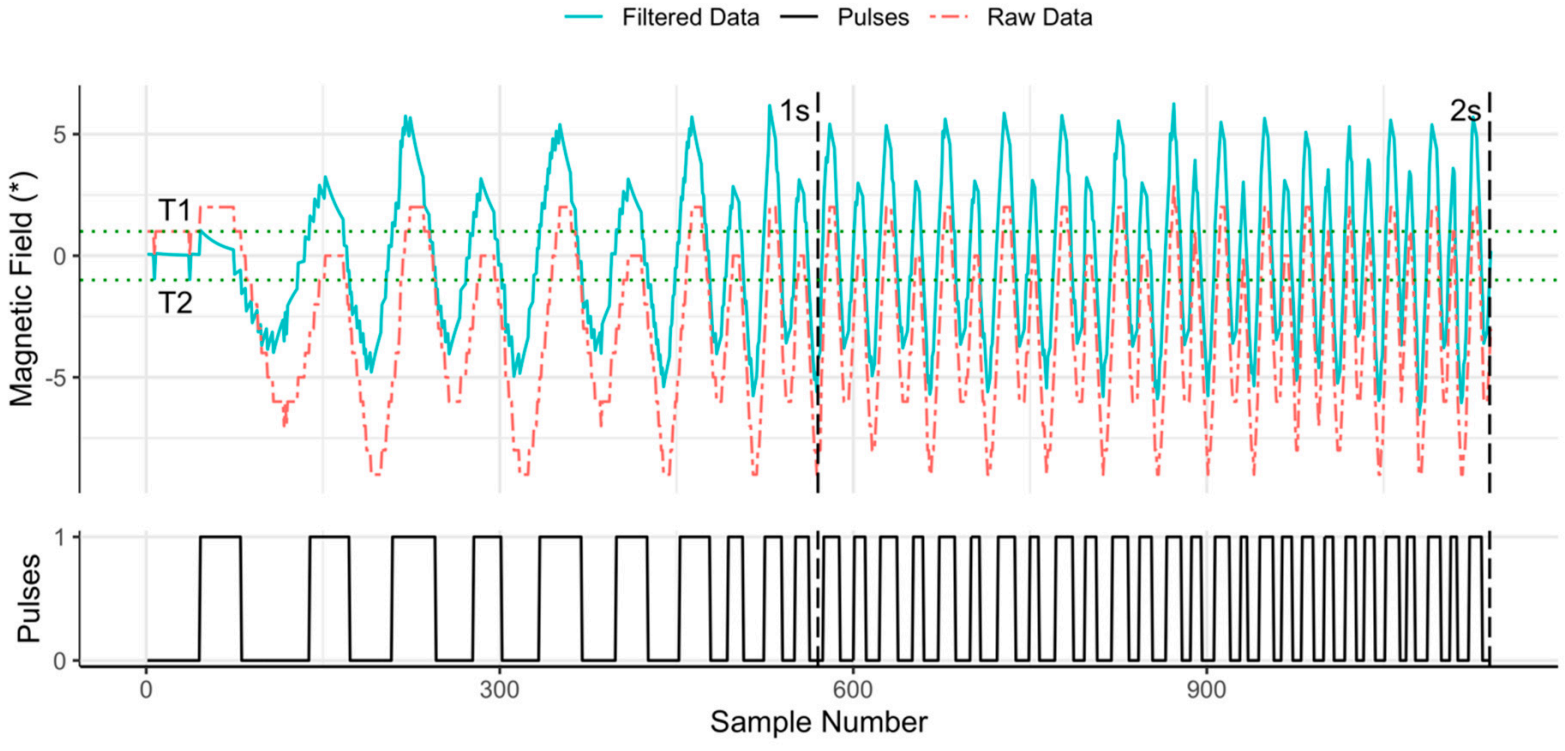


A LIS3MSDL magnetometer sensor.

Data Collection.

Principle of Operation

The magnetometer measures the magnetic field outside the meter's register (red dotted line).



Data Collection.

Principle of Operation

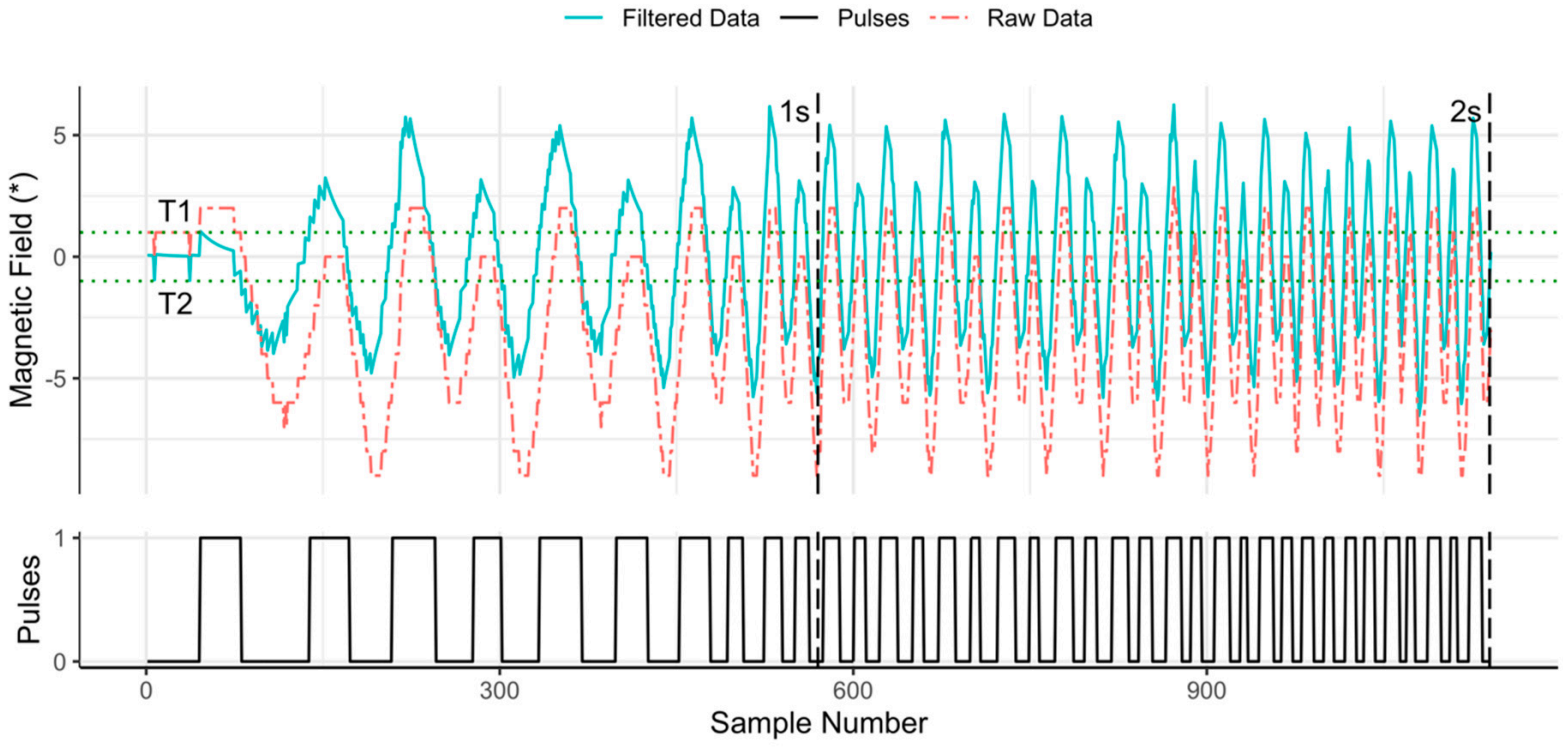
The filter produces a signal centered around 0 (blue solid line).

IIR Filter

$$y_n = 0.95 * y_{n-1} + x_n - x_{n-1}$$

x: input (red dotted line)

y: Output (blue solid line)

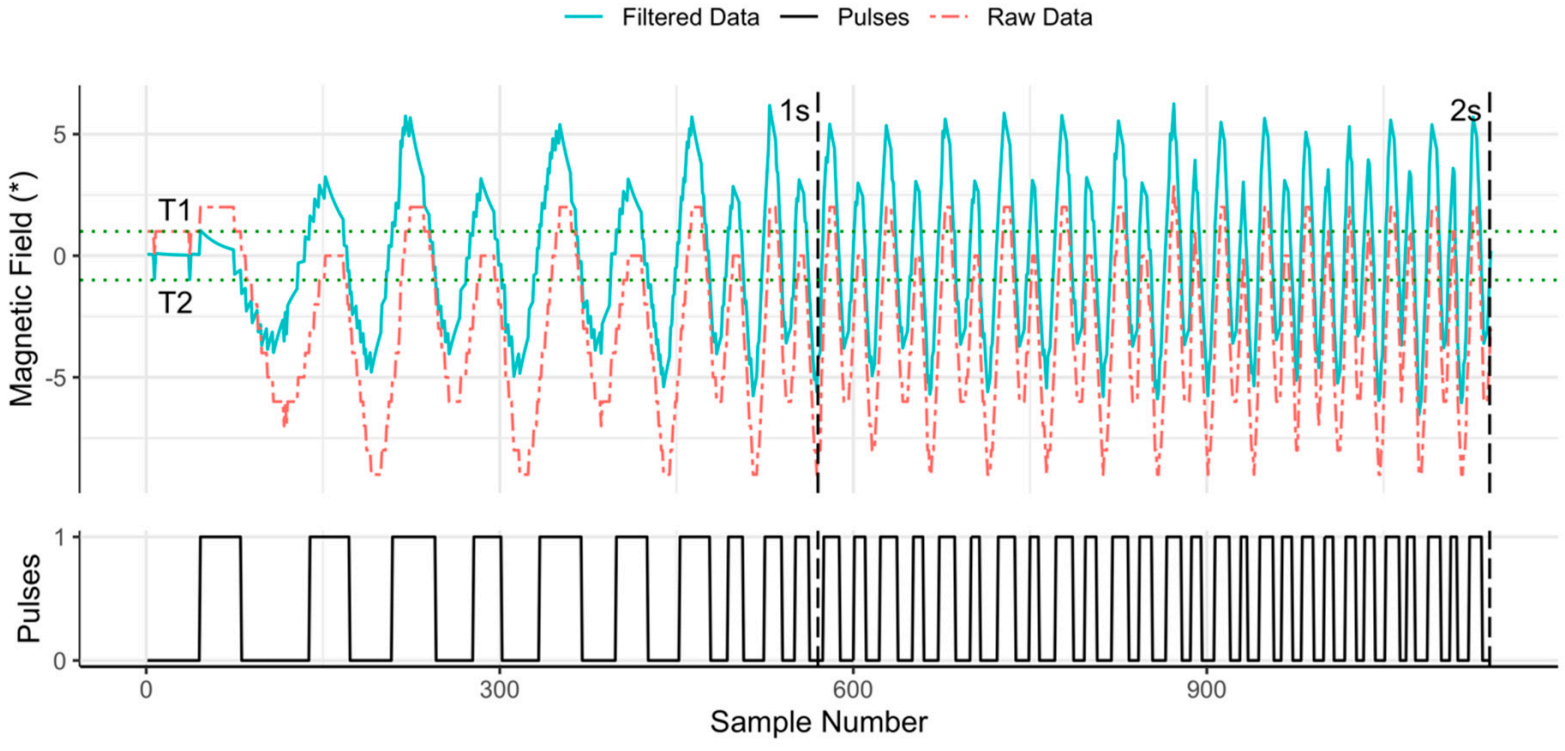


Data Collection.

Principle of Operation

Using a digital Schmitt Trigger around two fixed threshold values (T1 and T2) we count pulses over the filtered signal.

Resulting in the solid black line in the lower panel.



Data Collection. Principle of Operation

We record the date and time stamp and the number of pulses in each time interval and metadata needed to identify and track key information.

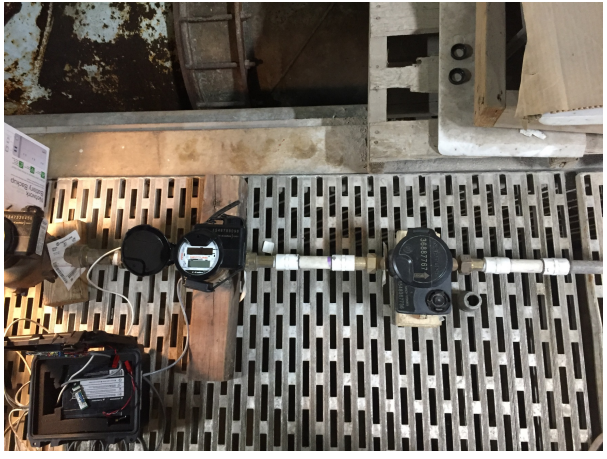
- The datalogger generates daily (configurable) CSV files.

- *A 4-seconds time resolution was used in this study (configurable).*

```
Site #: 001
Datalogger ID #: 001
Meter Resolution: 0.033
Time,Record,Pulses
"20-3-16 17:53:34",1,1
"20-3-16 17:53:38",2,0
"20-3-16 17:53:42",3,0
"20-3-16 17:53:46",4,0
"20-3-16 17:53:50",5,38
"20-3-16 17:53:54",6,117
"20-3-16 17:53:58",7,119
"20-3-16 17:54:02",8,118
"20-3-16 17:54:06",9,119
"20-3-16 17:54:10",10,119
```

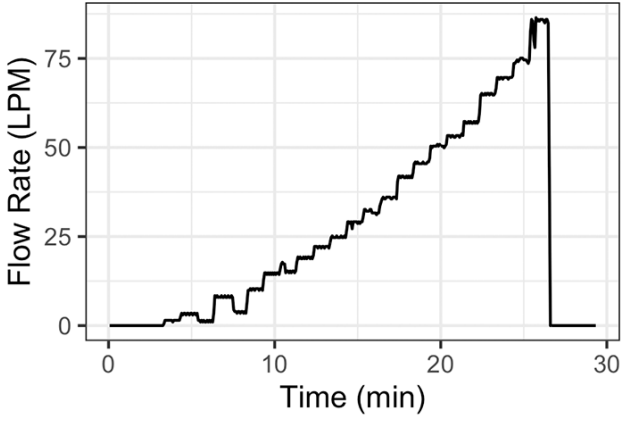
Data collection. Calibration

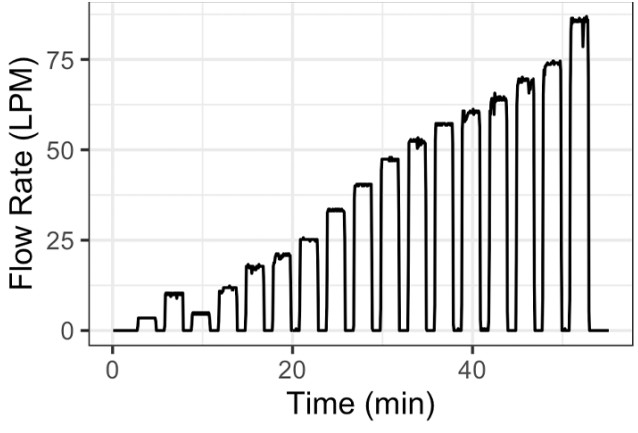
Laboratory setup to calculate the volumetric resolution (R), i.e., the volume of water that each pulse represents for different meters.



- $R = \frac{V_{olume}}{P_{ulses}}$

Repeated multiple times for different flow rates, and characteristics.





Data collection. Calibration

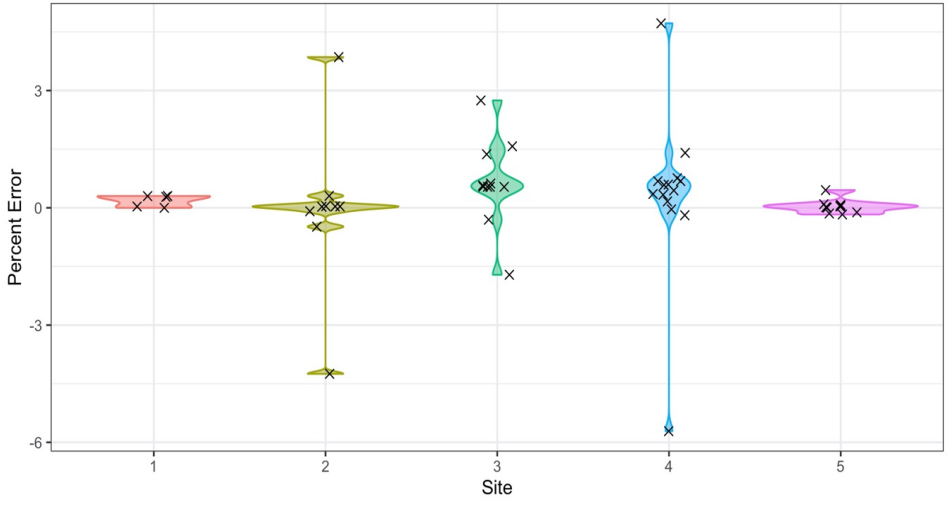
Volumetric resolution for the most commonly available water meters in Logan and Providence cities, Utah.

Meter Brand and Model	Size (in)	Pulse resolution (L/pulse)
Neptune T-10	1	0.1257
	5/8	0.0329
Master Meter BL	1	0.1575
	3/4	0.0957

Data collection. Accuracy and autonomy.

- Laboratory experiments: ± 1.5%
- Field deployments: ± 5% *

** After a learning period the accuracy in the field gets closer to that of laboratory experiments.*

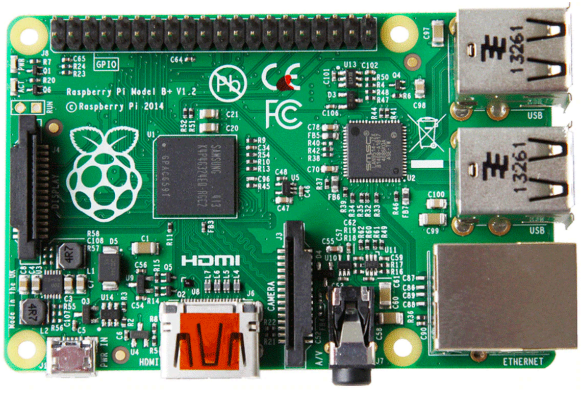


- Between 5 to 6 weeks of autonomous operation collecting data at a 4 seconds temporal resolution.

Data collection. Data Transmission.

- We added communication and computational capabilities to the CIWS datalogger by coupling it with a Raspberry Pi Model B or Model B+ single-board Linux computer.

- Same functioning principle. The R-Pi is turned on on a daily basis (configurable) to transmit CSV data files and turned off after completion of this task.

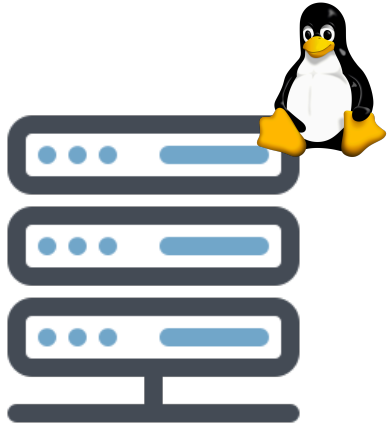




This device is called the ***CIWS-MWM NODE***. MWM denotes magnetically driven water meters.

Data Management and Archival.

The Data Management and Archival Layer components were deployed within a VMWare ESXi server environment hosted at Utah State University on a single virtual machine (VM) running the Ubuntu Linux Server Version 18.04 (Bionic Beaver) operating system.



ubuntu

Data Management and Archival.

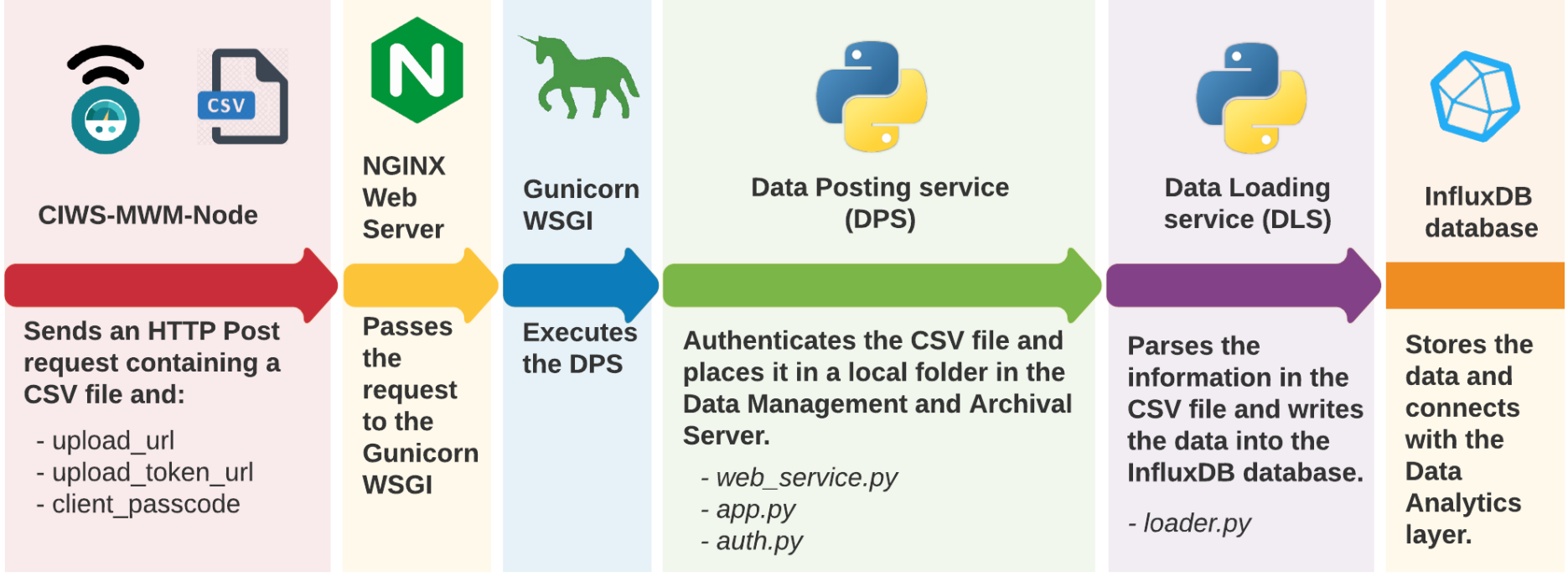
- The system is modular: *elements can be used, improved, modified separately.*

While some elements are tailored for the specific application presented, the system is generalizable:

- *Different dataloggers can be incorporated into the system regardless of data transmission technologies or capabilities.*
- *Additional parameters can be added.*

Data Management and Archival.

Push Based Data Collection for Single Family Residential Homes



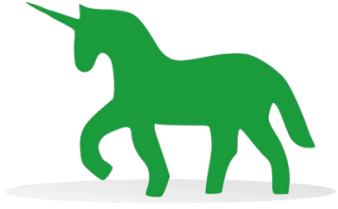
Data Management and Archival. Data Posting Service.

The DPS is a listener web service that receives and processes data files pushed to the server by the dataloggers.



It works integrated with two common technologies:

1) The web server software ***(NginX)*** that processes HTTP requests received by the server.
2) A Web Server Gateway Interface (WSGI) ***(Gunicorn)*** that runs the DPS application ***(Python)*** in response to the requests.





Data Management and Archival. Data Posting Service.

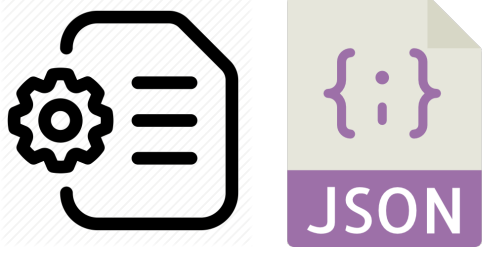
- The DPS uses a simple token-based authentication for the HTTP POST requests.



- A log file keeps track of the requests received by the DPS.



- Configured via settings stored in a user-modifiable JavaScript Object Notation (JSON) file



Data Management and Archival. Data Loading Service.

- Implemented in a single Python script.



loader.py

- Reads the CSV files from a local folder, parses the information contained in them and loads the data into an ***InfluxDB*** database.



It shares the configuration file with the DPS and uses a log file to track events.

Data Management and Archival. Database.

We tested multiple technologies and chose to use InfluxDB due to its time series oriented data structure, rapid query performance, and favorable disk space requirements.



It has client programming libraries are available for several popular programming languages, including Python, Go, C#, Java, PHP, Ruby, Scala, JavaScript, and R





Data Management and Archival. Database.

Database schema design:

Influx Key	InfluxDB Type	Data Type	Example Value
time	Time Index	Timestamp	2020-01-01 00:00:01
siteID	Tag	String	“1”
pulses	Field	Integer	5
dataloggerID	Tag	String	“1”



Data Analytics Layer.

- The developments in this layer serve as an example to illustrate possibilities for interaction with the data.



- This layer handles user authentication, interaction with InfluxDB, data query, visualization and analysis.



Data Analytics Layer.

- We developed an Application Programming Interface using Jupyter and Python.

- Configured (*user authentication, server connection*) using a JSON file.



Query	Python implementation.
Get **raw data for one or multiple sites**, between specific dates, or the entire record.	*get_data(site, startdate = None, enddate = None, measurement = 'QCData')*
Get **time aggregated data for one or multiple sites**, between specific dates, or the entire record.	*get_agg(site, function, t_res, startdate = None, enddate = None, Measurement = 'QCData')*

Data Analytics Layer.

We selected a single family residential property to test the CIWS functionality under a push based data retrieval model. We collected two weeks of data in this property, between **January 15, 2021 and January 28, 2021**, for the implementation described. All water use results presented are for this time period.

Characteristics:

- Five occupants, 3 of ages between 10 to 25 and 2 between 40 to 60
- Built in 2006, has 3 bathrooms and a total parcel area of approximately 12,000 ft^2.

Data Analytics Layer.

Once the data is available in a Python environment, multiple analysis can be conducted.

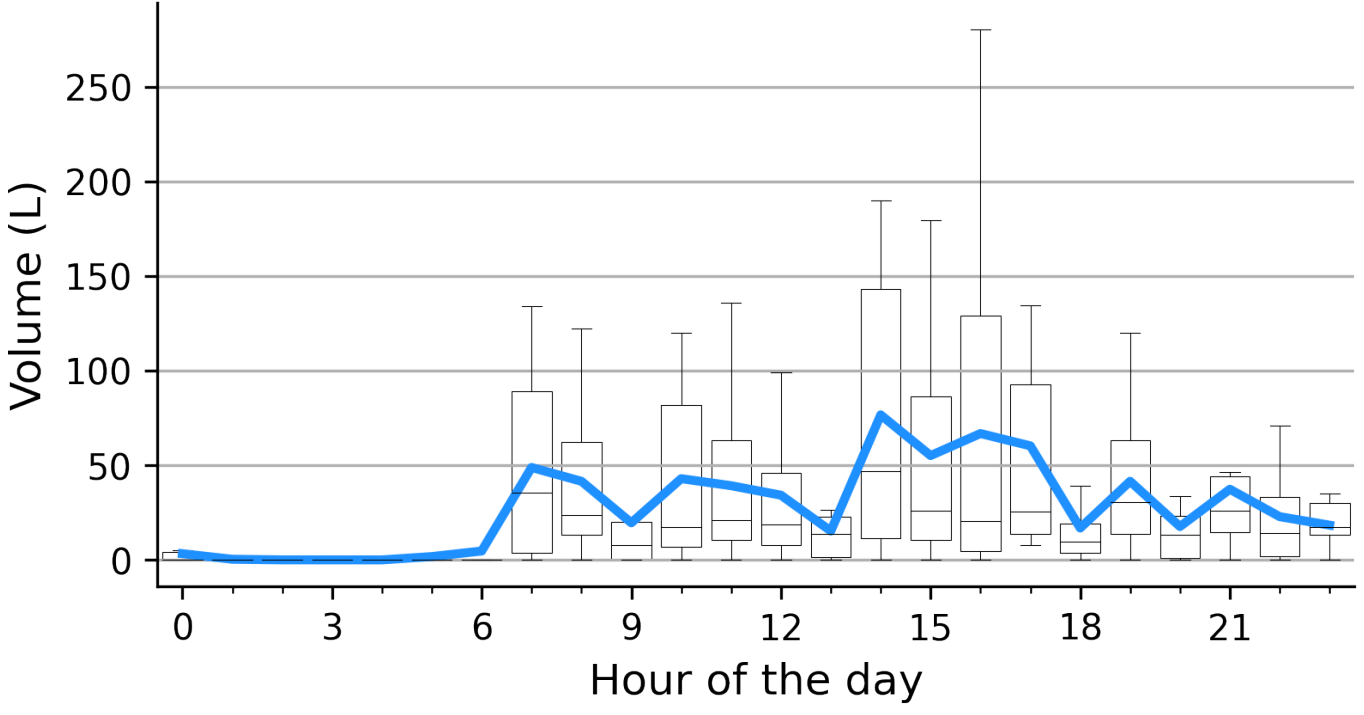
Currently implemented, as illustrative examples:

a) Water use statistics, from the raw data and time-aggregated values:

The **average daily water use** was 170.2 gallons, leading to a per capita average daily water use of 34 gallons. The **maximum daily water usage** observed during the period was 292.7 gallons, the **instantaneous peak** was 10 gpm and the **maximum hourly usage** registered was 74.1 gallons.

Data Analytics Layer.

Hourly average and variability

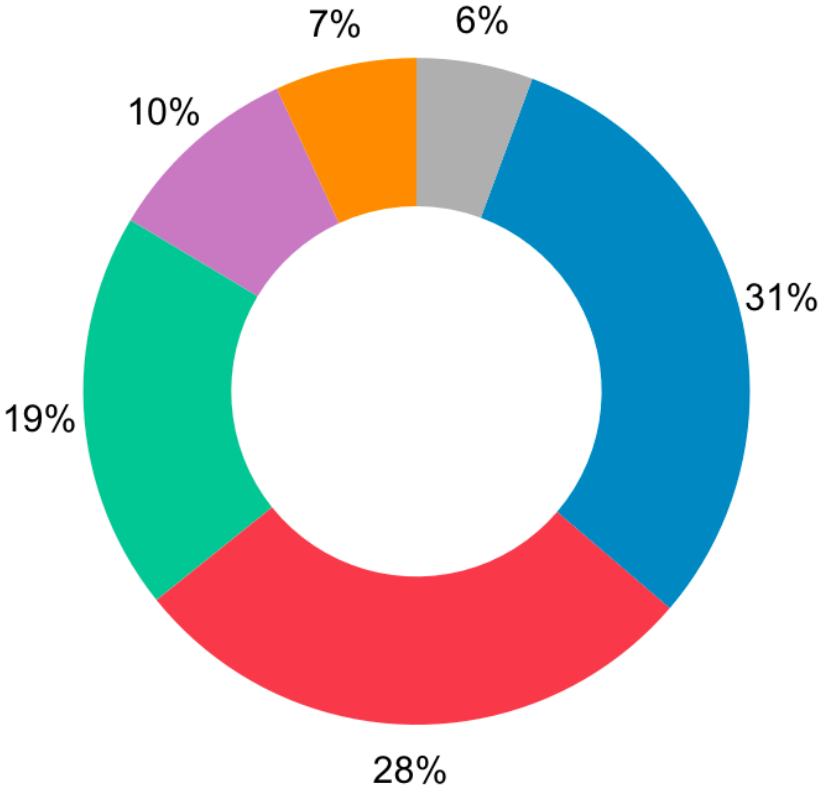


Data Analytics Layer.

- End uses of water classification via an open source, algorithm.

Volume distribution across end uses:

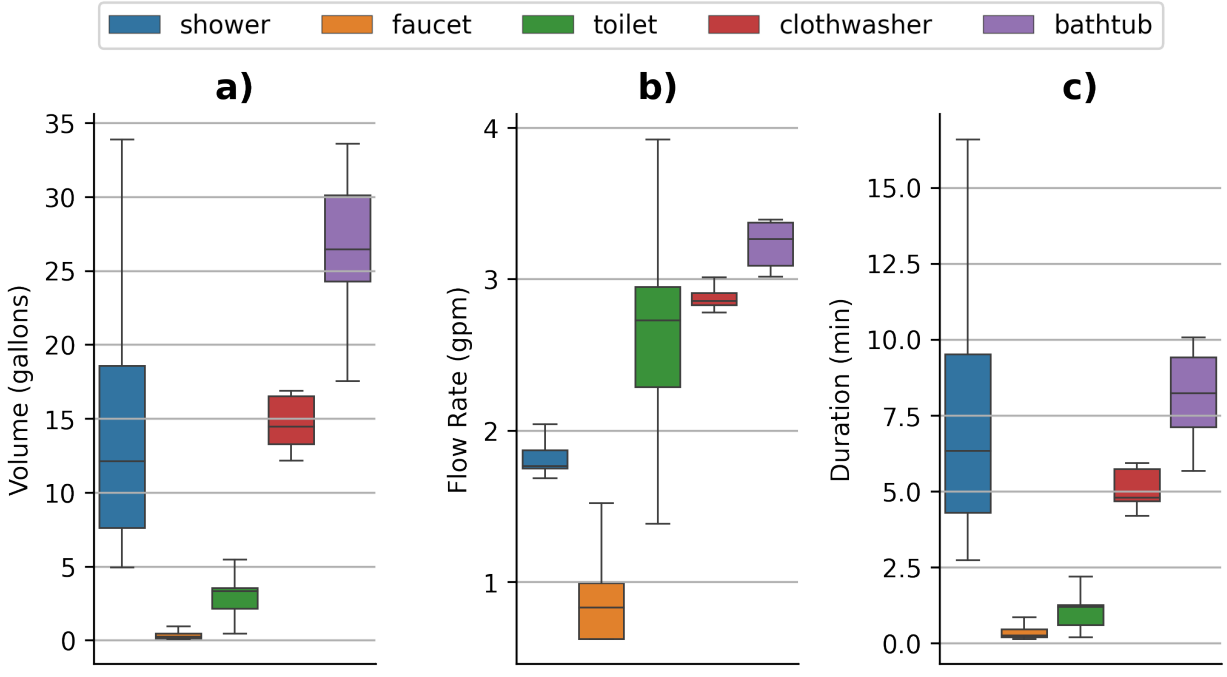




Data Analytics Layer.

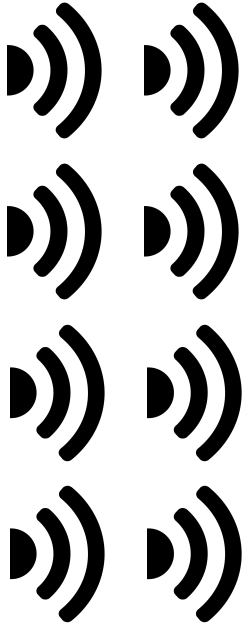
- End uses of water classification via an open source, algorithm.

Fixture characteristics:



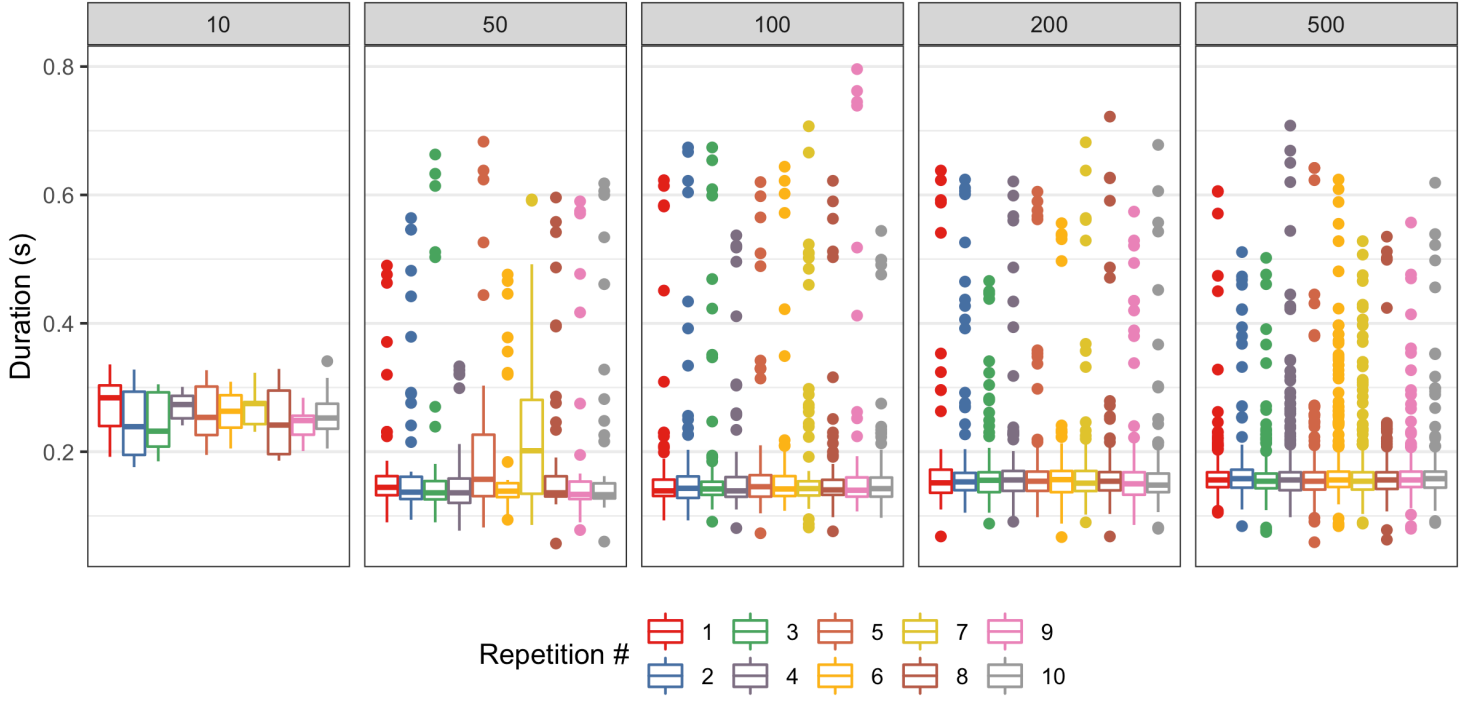
Scalability and performance metrics. DPS testing

- We simulated 10, 50, 100, 200, and 500 dataloggers simultaneously sending data into the system.
- Each batch was repeated 10 times.
- 100 % success transmission rate.



Scalability and performance metrics. DPS testing

Processing times, separated by the number of HTTP POST requests in the batch (10, 50, 100, 200, and 500) for each repetition, from 1 to 10. Duration is calculated as the final processing time minus the starting time of each individual POST request



Scalability and performance metrics. DLS testing

We simulated and different loading operations and repeated each 10 times. The results are:

Load Operation	Average duration (seconds)	Standard deviation (seconds)	Average time for processing 1 file (seconds)
1 file from 1 site	0.37	0.06	0.37
10 files from 1 site	3.96	0.14	0.40
1 file from 10 sites	4.67	0.23	0.47
50 files from 1 site	19.92	0.67	0.40
1 file from 50 sites	23.87	0.33	0.48
100 files from 1 site	39.87	1.05	0.40
1 file from 100 sites	47.48	0.89	0.47
500 files from 1 site	195.19	2.98	0.39
1 file from 500 sites	240.70	3.00	0.48

Scalability and performance metrics. Database querying test.

We measured the time it takes to download n days of data from a different number of sites:

Days of data (n)	Number of sites	Average duration (seconds)	Standard deviation (seconds)
1	1	0.17	0.02
1	5	0.81	0.03
1	10	1.62	0.04
7	1	1.16	0.04
7	5	5.74	0.07
7	10	11.39	0.07
30	1	4.51	0.27
30	5	22.46	0.52
30	10	45.47	1.24

Final remarks.

- **A complete cyberinfrastructure system that uses a layered approach to collect and manage high-temporal resolution water use data was developed and implemented.**

- The system was designed focusing on the scale of data collection that would be required for research projects conducted by utilities or other researchers

Final remarks.

- CIWS combines multiple open-source technologies. The modular design makes it easier to replace or update technology elements in the system, if needed.

- Additional tools can be added to system - e.g., more advanced analytics tools, enhanced authentication protocols.

- The system testing, performance metrics, and deployment demonstrate that CIWS can meet and significantly exceed the design considerations in terms of scale and performance.

Availability of the tools developed:

All CIWS elements and the case study data described are freely available for re-use.

- **CIWS MWM** datalogger documentation: https://doi.org/10.3390/s20133655
- **CIWS MWM** datalogger hardware (design and instructions to build, including a manufacturable PCB) and firmware: https://github.com/UCHIC/CIWS-MWM-Logger/tree/master/Hardware

- **CIWS MWM NODE** (datalogger with computational and wireless communication capabilities) hardware and fimrware: https://github.com/UCHIC/CIWS-WM-Node

Availability of the tools developed:

All CIWS elements and the case study data described are freely available for re-use.

- **Data Management and Archival** tools: https://github.com/UCHIC/CIWS-Server
- **Data Analytics Layer** and **study data**: http://www.hydroshare.org/resource/aaa7246437144f2390411ef9f2f4ebd0

- Disaggregation algorithm used: https://www.hydroshare.org/resource/1521eba67f1d4571ac5fe2b8c5e01035/

Acknowledgments:

- Utah Water Research Laboratory
- Providence and Logan cities utility companies.
- Nour Atallah, Arle J. Beckwith, Elijah West.

Contact information:
Jeffery S. Horsburgh (jeff.horsburgh@usu.edu)
Camilo J. Bastidas Pacheco (camilo.bastidas@usu.edu)