Ecole Polytechnique Fédérale de Lausanne

Master Thesis

Design of a spatial data infrastructure for low-cost distributed air quality sensors : a contribution to the Chicago Array of Things project

author :
Anaïs Ladoy
Department of Environmental Sciences and Engineering

supervised by :
Dr. Stéphane Joost
Laboratory of Geographic Information Systems (LASIG, EPFL)

Prof. Luc Anselin
Center for Spatial Data Science (CSDS, The University of Chicago)

August 17, 2018

Contents

1 Introduction

The environmental awareness of the world has considerably increased throughout the past decades and it is now recognized that global actions are needed to protect the planet (World Health Organisation, 2017). In point of fact, the United Nations have dedicated four of these Sustainable Development Goals (SDGs) to the Earth preservation: Responsible Consumption and Production, Climate Action, Life Below Water, and Life on Land (United Nations, n.d.).

The Paris Agreement (United Nations Treaty Collection, 2016) that emerged from the 2015 United Nations Climate Conference (COP 21) is a step further toward the achievement of the SDG relative to the climate action, where 196 countries agreed to keep, for this century, the increase of global warming to "well below 2°C". Each participating country shall translate the Paris agreement into national agendas and regularly report on its contribution to global warming. In addition to the major concerns as the lack of binding enforcement mechanism, an important question emerges from these agreements. In order to undertake direct actions and to assess the progress made to achieve the fixed objectives, legislators need to understand the current state of the surrounded environment and establish future trends of environmental parameters that help to prevent new environmental issues. This is the role of environmental monitoring (Weston, 2011).

1.1 Environmental monitoring

Downes et al. (2002) classify monitoring in four categories according to the different objectives :

- **Long-term monitoring** aims to provide background measure for long-term dynamics of natural systems. They provide frameworks upon which shorter term or localized changes (e.g. anthropogenic impact) could be measured against.
- **Compliance monitoring** aims to ensure that a stipulated regulation is followed. For example, the U.S. Environmental Protection Agency (EPA) performs Air Compliance Monitoring to ensure compliance with the Clean Air Act (CAA) and Water Compliance Monitoring to ensure compliance with the Clean Water Act (CWA).
- **Impact monitoring** aims to assess the human impact exposure on the natural environment with the objective of taking remedial measures to prevent or minimize such impacts.
- **Environmental monitoring** aims to gain some indication of the current state of the environment by the systematic sampling of air, water, soil or biota (Weston, 2011).

Analysis of the data collected by environmental monitoring sensors allows identifying future trends in environmental conditions, supporting policies development and implementation, and assessing their cost-effectiveness or not. Artiola, Pepper, and Brusseau (2004) provide a non-exhaustive list of additional benefits of environmental monitoring specific to diverse fields of application (Table 1).

BOX 1.1 *Knowledge-Based Regulation and Benefits of Environmental Monitoring*

Protection of public water supplies: Including surface and groundwater monitoring; sources of water pollution; waste and wastewater treatment and their disposal and discharge into the environment

Hazardous, nonhazardous and radioactive waste management: Including disposal, reuse, and possible impacts to human health and the environment

Urban air quality: Sources of pollution, transportation, and industrial effects on human health

Natural resources protection and management: Land and soil degradation; forests and wood harvesting; water supplies, including lakes, rivers, and oceans; recreation; food supply

Weather forecasting: Anticipating weather, long- and short-term climatic changes, and weather-related catastrophes, including floods, droughts, hurricanes, and tornadoes

Economic development and land planning: Resources allocation; resource exploitation

Population growth: Density patterns, related to economic development and natural resources

Delineation: Mapping of natural resources; soil classification; wetland delineation; critical habitats; water resources; boundary changes

Endangered species and biodiversity: Enumeration of species; extinction, discovery, protection

Global climate changes: Strategies to control pollution emissions and weather- and health-related gaseous emissions

Table 1: Knowledge-based regulation and benefits of environmental monitoring (Artiola et al., 2004)

Despite its undisputed benefits, several countries still not have an effective monitoring network (United Nations, 2003). Even when it is the case, spatial and temporal gaps can importantly restrict the knowledge we can infer from the collected data (Gouveia, Fonseca, Câmara, & Ferreira, 2004) and in turn, the ability to use them efficiently for decision-making (United Nations, 2003).

Nowadays, technological advances in the sensing field have motivated the development of low-cost sensors that can be densely deployed in a multitude of environmental systems including marine environments, air quality, biodiversity and forestry (Stepenuck and Green, 2015; Gouveia et al., 2004). This trend shifts data collection from a small number of formal institutions using expensive and large monitoring stations toward a decentralized and diverse network of stakeholders using low-cost distributed sensors (Buytaert, Dewulf, De Bièvre, Clark, & Hannah, 2016).

This new paradigm of environmental monitoring carried out the subject addressed by this master thesis, with a focus on urban air quality, an important concern for human health (Benammar, Abdaoui, Ahmad, Touati, & Kadri, 2018).

1.2 Context

During the last decades, an increasing number of epidemiological studies have found associations between air pollution and an increased morbidity and mortality (Mannucci, Harari, Martinelli, & Franchini, 2015). In the United Nations, 2003 Report, recommendations were already made for a strengthening of air pollution monitoring.

However, according to a recent report of the World Health Organization (2016), no improvement in outdoor air quality has been made over the last decade and 90% of the world's urban population was still exposed to $PM_{2.5}$ concentrations exceeding the WHO air quality guidelines in 2014 (Figure 1). The lack of monitoring of air pollution levels, sources and consequences on public health has been identified as a major obstacle to the reduction of mortality caused by air pollution (World Health Organization, 2016).

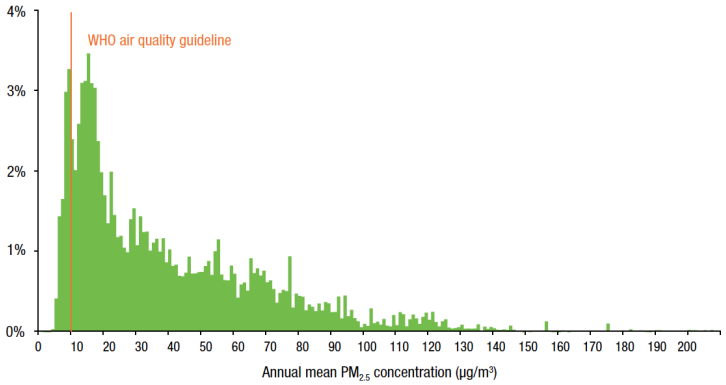


Figure 1: Distribution of the world's urban population by the concentration of particulate matter with an aerodynamic diameter of 2.5 μm or less ($PM_{2.5}$) in 2014. *(source: World Health Organization, 2016)*

Indeed, an effective air quality monitoring should provide valuable information to help authorities to take direct actions for air pollution reduction such as traffic or industrial activity control or land use management (Xie et al., 2017).

In the United States, the Environmental Protection Agency (EPA) has setting up a national air quality monitoring network, composed of fixed stations with Federal Reference Method (FRM) or Federal Equivalent Method (FEM) instruments (White, n.d.). Their expensive cost ($\sim \$10,000$) (Clements et al., 2017), allows them to provide high-quality time-series data (usually at an hourly resolution) but only at sparse point-based locations with spatial resolutions in the order of several hundred kilometers (Marjovi, Arfire, & Martinoli, 2015).

As air pollution depends predominantly on local emission sources, the sparsity of the regulatory monitoring network makes difficult the assessment of air pollution field trends and human exposure to pollutants (Kumar et al., 2015). Recent technological advances have brought a promising solution to overcome the problem of the low spatial resolution with low-cost air quality sensors. Although the data produced by these sensors are not as precise than the ones collected by FRM/FEM instruments, they can be used in a high number of locations simultaneously and allow the assessment of air quality at a high time and spatial resolution (World Meteorological Organization, 2018).

It is precisely to take advantage of this new technology that a research study dedicated to the urban air quality in Chicago, Array of Things (AoT), has been launched two years ago. The AoT is "an urban sensing project, a network of interactive, modular sensor boxes that will be installed around Chicago to collect real-time data on the city's environment, infrastructure, and activity for research and public use. It will essentially serve as a 'fitness tracker' for the city, measuring factors that impact livability in Chicago such as climate, air quality and noise" ("Array of Things," n.d.). The sensors platforms used in the AoT project (i.e. Waggle platform) are designed as part of a research project elaborated by the Argonne National Laboratory (ANL) and are installed at the top of traffic signal poles across Chicago to benefit the City's power supply. Each node contains several components including low-cost air quality and meteorological sensors, cameras, linux controllers that send, every 30 seconds

approximately, observations to the ANL server (see the nodes architecture in Figure 2). Raw data will be open, free and available to the public through different portals. Moreover, several third parties will further analyse the collected data to develop applications addressing challenges as urban air quality, traffic safety or urban flooding.

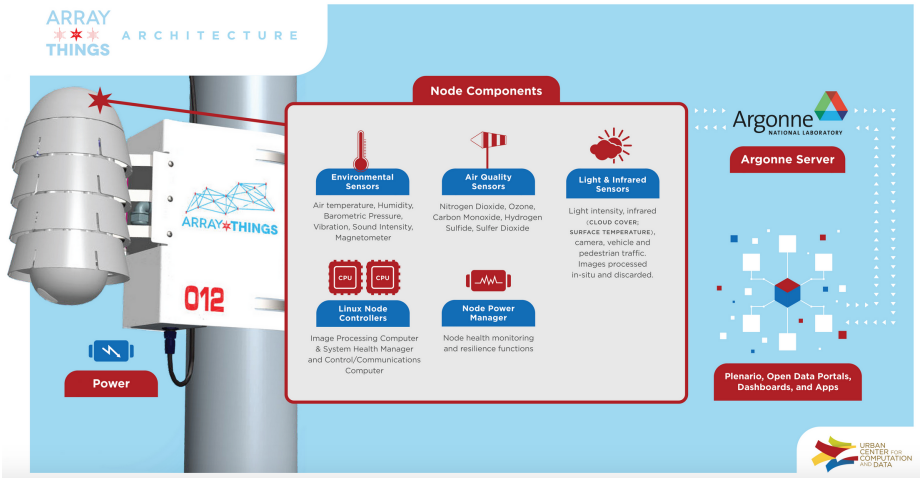


Figure 2: AoT node architecture *(source: AoT, https://arrayofthings.github.io)*

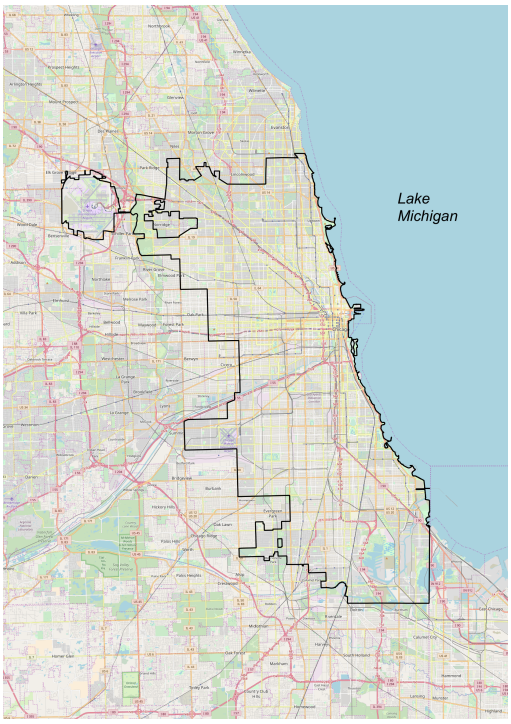
The installation of a total of 105 nodes throughout the city will allow obtaining information about the air that 80% of the Chicagoan population breathe (Figure 3b) and will provide a unique opportunity to monitor the air quality in the city and implement direct actions against air pollution.

The organization of the AoT project is complex and involves several stakeholders [1] and numerous partnerships (see the full list on the AoT website ("Array of Things," n.d.)). The different AoT members have handled the sensors platforms development, the monitoring network design, the nodes deployment, and the sensors calibration and they will in the longer term, ensure the maintenance of the monitoring network, the collection of the distributed sensors data, the data storage on ANL servers and the distribution of raw data through different platforms. However, no analysis or modeling will be conducted as part of the AoT project and this task will be done by any third-parties who might be interested in working with the data.

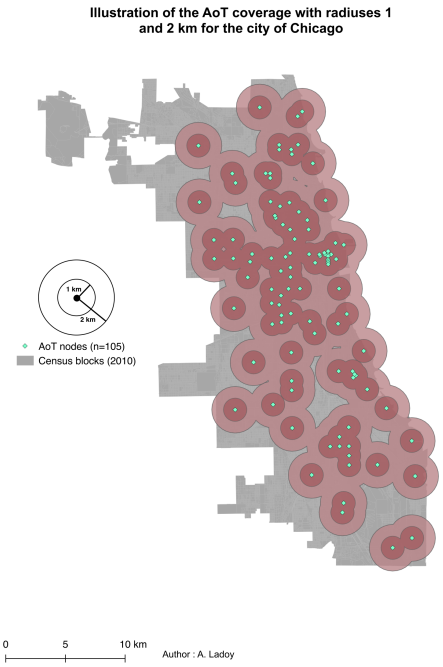
In this context, the Center of Spatial Data Science (CSDS) in the University of Chicago has started the Air Quality Project ("GeoDaCenter/airquality," n.d.) in January 2018 to analyze air quality data from AoT sensors, as part of "The Partnership for Healthy Cities (PHC)" (n.d.). Supported by Bloomberg Philanthropies in partnership with the World Health Organisation, the PHC

[1] Urban Center for Computation and Data (http://www.urbanccd.org/), Computation Institute (https://ci.uchicago.edu/), Argonne National Laboratory (https://www.anl.gov/), University of Chicago (https://www.uchicago.edu/), City of Chicago (https://www.cityofchicago.org/city/en.html), School of the Art Institute of Chicago (http://www.saic.edu/)

financially helps a global network of cities, including Chicago, to implement interventions against noncommunicable diseases and injuries.



(a) Chicago boundaries (*source: Chicago Data Portal, https://data.cityofchicago.org/, OpenStreetMap Basemap)*



(b) AoT monitoring network planned coverage

Figure 3: Definition of the AoT study area (3a) and coverage provided by the AoT sensors (3b). Based on the total population US 2010 census (Chicago Data Portal, https://data.cityofchicago.org/), the AoT nodes cover 42% of the total population with a 1km radius around each node and 80% of the total population with a radius of 2km around each node. *Notes: Values corresponding to the population coverage are slightly overestimated as we have summed the population of intersecting blocks, some of them are not completely contained in the buffer. The nodes used in this analysis are the ones initially planned (n=105).*

For this project, the CSDS has defined three distinct objectives, all related to the AoT air quality data, according to the different interests of three stakeholders namely the AoT team, Bloomberg Philanthropies, and the Chicago Department of Public Health (CDPH). They respectively consist of :

1. The creation of interpolated maps of AoT distributed sensors air quality data;

2. The identification of air pollution drivers with a similar method like the one used in the New York City Community Air Survey (Matte et al., 2013; Clougherty et al., 2013);

3. The intra-urban variation assessment of air pollutant (especially $PM_{2.5}$) and their potential impact on health.

The second objective was a key milestone for the PHC Grant, and its achievement by the end of 2018 is a priority in the Air Quality Project. A roadmap has been elaborated to detail its implementation (see Figure 12 in Appendix, p. 44).

In addition to the ubiquitous need of air quality sensors data, drivers and spatio-temporal predictors will also be required in the modeling approaches (a preliminary list is presented in Figure 13 in Appendix, p. 45).

To achieve the different objectives of this large-scale project, the just mentioned spatial data resources have to be accessible to the whole CSDS team and the data collection have to be accurate, up to date, consistent and centralized to avoid a duplication and a fragmentation of data. This is the role of a Spatial Data Infrastructure (SDI) (Phillips, Williamson, & Ezigbalike, 1999).

Unfortunately, important delays have occurred in the AoT project and the majority of AoT sensors are still in a calibration phase. This procedure consists of normalizing sensor responses with FRM/FEM ones (i.e. regulatory monitors) at collocated sites in order to quantify sensor drift and help bound uncertainty (Kumar et al., 2015; Clements et al., 2017).

In order to fulfill the exigencies of the PHC grant, the CSDS has decided to search for new sources of distributed air quality sensors data. They also decided to develop an initial framework able to handle the extraction and storage of the different spatial data resources and, of the calibrated AoT data as soon as these data will become available.

1.3 Goal and objectives

In this context, the goal of this master thesis is to design an SDI for distributed air quality sensors data to provide a framework to facilitate the downloading, the analysis and in turn the understanding of the spatial distribution of air quality in the city of Chicago.

The following requirements (following Urbano et al., 2010) were identified for the SDI design :

1. *Data scalability.* Distributed sensors can record several observations per minute at numerous locations depending on the size of the monitoring network. A persistent and large data storage is required to handle this important amount of data.

2. *Periodic and automatic data acquisition.* New observations are collected every day by monitoring sensors and are usually made available via data access points (Vitolo, Elkhatib, Reusser, Macleod, & Buytaert, 2015). It requires automated procedures to extract and store new data either continuously or at regular intervals depending on the time resolution needed.

3. *Management of spatial information.* Geodata is information that can be geographically referenced in some consistent manner using latitude/longitude, national coordinates, postal codes, etc. (Awange & Kyalo Kiema, 2013). Thus, the spatial location must be efficiently stored and tools must be available for spatial manipulation.

4. *Management of temporal aspect.* A specificity of environmental sensors is the production of time series data, which consists of repeated measurements of several parameters over time (Dunning & Friedman, n.d.). Thus, these components must be stored and we must ensure the efficiency of data manipulation relative to the temporal aspect.

5. *Heterogeneity of applications.* The complex nature of environmental sensors data implies that data should be explored, analyzed, visualized by a wide range of task-oriented applications.

6. *Integration of different data sources.* The infrastructure should allow the integration of other spatial data resources, in our case the drivers and spatio-temporal predictors data required for the Air Quality Project modeling stage (see Figure 13 in Appendix, p. 45). It will provide an opportunity to expand the collected information and thus improve the understanding of the studied phenomenon.

7. *Multi-user support.* The infrastructure should be accessed locally or remotely by the different CSDS collaborators.

8. *Data sharing.* The SDI must be comprehensible and reproducible by the different actors involved in the project. Thus, it must include standard data definition, methods details, etc.

9. *Flexibility.* Additional needs might appear along the project and the SDI must be flexible enough to minimize the changes that need to be made to the infrastructure.

10. *Cost-effectiveness.* The cost-effectiveness of the SDI is an important added value that guarantees its accessibility.

Taking into consideration the requirements mentioned above, the three main objectives of this research will be to :

- identify which air quality sensors data we want to store in the SDI, where to collect them and at which frequency;

- design a centralized spatial database where the selection of attributes we want to store, the definition of standards and the modeling of the interrelationships between data sets will be crucial for an efficient integration of the different data resources;

- automate the process by the implementation of a workflow, which will ensure the maintenance and the consistency of the SDI;

An example of such an SDI is presented in Figure 4. The heterogeneous data sources must be extracted and inserted in a spatial database that can be accessed through different third-party applications that will, in turn, enable data exploration, spatial analysis, and visualization. The output has then to be stored back in the database.

The components of the planned SDI are displayed in red and the main steps leading to its implementation are presented in the next section.

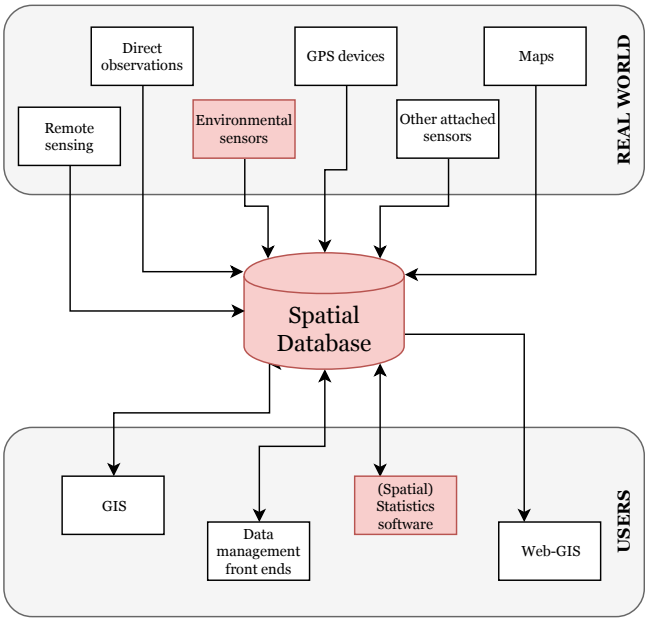


Figure 4: Schema of a possible spatial data infrastructure that combines information from heterogeneous sources in a centralized spatial database where it is accessed remotely or locally by client applications for manipulation, visualization, and analysis. Outputs are then stored back in the database. In red are the components implemented in this project. Reproduced from Urbano et al. (2010)

.

1.4 Methodology

Due to important delays in the AoT project and a lack of coordination between the partners, it was not possible to use systematic interviews of the different stakeholders and to elaborate a precise framework for the SDI.

Following the needs of the Air Quality Project, the first step of the SDI design was to define which air quality data would be stored in the SDI. For this purpose, other options of distributed air quality sensors data sources have been considered in order to complete AoT sensors data, including data from regulatory monitoring network and data from volunteer monitoring programs. For each data source, different parameters had to be chosen including the data extraction frequency, the time aggregation level, the data spatial extent, etc.

The important constraints relative to data access and availability have mainly dictated the SDI design and the priority was to guarantee the flexibility of the infrastructure and the maximization of the information stored.

A literature review has permitted to select different software and tools to ensure an efficient storage and retrieval of the geodata.

Furthermore, several scripts were elaborated to ensure the different tasks inherent to an SDI needs including data extraction, data standardization and database update.

2 Literature review

2.1 Air Quality Observation Systems

According to the World Health Organization, 2016 Report, air pollution monitoring is conducted in 3,000 cities worldwide. The same report alerts about the lack of urban air quality measurements in many cities, which makes it impossible to identify air pollution sources and thus limit the ability of decision-makers to assess risks, set targets and measure progress.

Since 1990, the Clean Air Act (US EPA, 2015) governs the air quality in the United States. Under this law, the U.S Environmental Protection Agency (EPA) has established limits for the concentration of six major air pollutants [2], known as criteria pollutants, across all the country. The compliance to these limits, defined as National Ambient Air Quality Standards (NAAQS), is assessed through the use of EPA's Air Quality System (AQS) monitors data.

These reference stations provide highly accurate measurements for a variety of pollutants at the cost of expensive and large monitoring devices (Marjovi et al., 2015), resulting on a sparse monitoring network. Modeling approaches are used to form a macroscopic view of pollution field trends across the country (Kumar et al., 2015) but the spatial resolution is not sufficient enough to provide information about localized gradients of potential importance to health protection (Nuria Castell et al., 2017).

The recent technological progress in remote sensing offers a valuable opportunity for air quality monitoring as it extends significantly the spatial and temporal coverage offered by fixed-monitoring networks (Committee on Environment, Natural Resources, and Sustainability, 2013). U.S. Atmospheric Remote Sensing programs are led by the National Aeronautics and Space Administration (NASA) and the National Oceanic and Atmospheric Administration (NOAA) that have launched in the last two decades a suite of satellites [3] allowing the measurements of columns of Aerosol Optical Depth (AOD), O_3, H_2O, CO, CH_4, SO_2, NO_2, $CFCs$, other pollutants and atmospheric parameters such as temperature, cloud properties and water vapor (Committee on Environment, Natural Resources, and Sustainability, 2013).

There is an extensive literature of how to integrate satellite imagery in urban air quality studies especially for $PM_{2.5}$ measurements as it has been found that satellite-derived aerosol optical depth (AOD) measurements were correlated to $PM_{2.5}$ concentrations (Kloog, Nordio, Coull, and Schwartz, 2012; Kloog, Nordio, Coull, and Schwartz, 2014; Stafoggia et al., 2017). The Multiangle Implementation of Atmospheric Correction (MAIAC) algorithm applied to AOD data in combination with spatio-temporal predictors allows the prediction of $PM_{2.5}$ concentration at 100m-scale across Switzerland (de Hoogh, Héritier, Stafoggia, Künzli, & Kloog, 2018). It is important to note that the 100m resolution results

[2] Criteria Air Pollutants: Ground-level Ozone, Particulate Matter, Carbon Monoxide, Lead, Sulfur Dioxide, Nitrogen Dioxide.

[3] Including Terra, Aqua, Aura, CALIPSO (Cloud-Aerosol Lidar and Infrared Pathfinder Satellite Observation), as well as NOAA-17, NOAA-18, NOAA-19, and Suomi NPP (National Polar-orbiting Partnership).

of a mixed model comprising the ground-based monitoring stations and other spatial-temporal predictors, the daily MAIAC spectral AOD being available at a 1km resolution.

Although satellite data can help to fill the gap of ground-based monitoring stations, this procedure is complex as satellite observations do not directly correspond to in situ measurements of pollutant concentrations and it involves the integration of models and ground-based information (Committee on Environment, Natural Resources, and Sustainability, 2013). Furthermore, the availability of satellite data depends on meteorological conditions (e.g. unavailable in cloudy days) that can create gaps in spatial coverage. Therefore, satellite imagery can only be used as a complement to ground-based or aircraft measurements.

During the last decade, an increasing number of low-cost sensors ($< \$2,500$) measuring air particles and gases have appeared on the market (Clements et al., 2017), becoming an interesting solution to overcome the lack of small-scale measurements of air quality. Their characteristics and applications are the subjects of the next section.

2.2 The emergence of distributed low-cost sensors in Air Quality monitoring

Progress in developing low-cost micro-scale sensing technology has reduced the cost of air pollution sensors from thousands to hundreds of dollars (Kumar et al., 2015) and we are now experiencing a paradigm shift in how and who is monitoring air quality (Núria Castell, Viana, Minguillón, Guerreiro, and Querol, 2013; Lewis and Edwards, 2016).

Compared to the traditional fixed-site air monitoring devices, this new technology aims to be small, easy to use, user-friendly and with end-to-end solutions (Kumar et al., 2015).

These significant advantages allow the measurements of gas and particles concentrations at the hyperlocal level and at a very high time resolution (Williams, 2018), providing two main fields of sensors application that was impossible with traditional approaches :

- Sensors can be deployed as dense fixed monitoring networks (ubiquitous monitoring) to assess pollution variability across a city or even at a smaller scale as community level.
- The integration of Global Positioning System (GPS) devices has brought out the emergence of a mobile sensing system approach. Personal wearable sensing platforms provide new capabilities to assess the air pollution health impacts (Nuria Castell et al., 2017) and transportation sensing networks, where sensors are anchored to moving sensor carriers, allow a finer spatial resolution, a wider coverage with fewer nodes and a cheaper maintenance (Marjovi et al., 2015).

Recent studies highlight the fascinating diversity in terms of sensors, measured parameters, monitoring objectives, end-users, spatial and temporal scales.

As a matter of example, Marjovi et al. (2015) modeled at a city-scale level in Lausanne (Switzerland) the Lung Deposited Surface Area (LDSA) that quantifies human exposure to ultrafine particles. For this purpose, they developed a spatially and temporally dynamic network coverage where sensing nodes were anchored to ten public buses, measuring several air quality parameters (e.g. CO, NO_2, $LDSA$). Bikes (Bigazzi and Figliozzi, 2015; Peters et al., 2014), pedestrians (Zwack, Paciorek, Spengler, & Levy, 2011) and cars (Hatzopoulou et al., 2017) can also be used as sensor carriers.

Peters et al. (2014) used a bicycle platform in the streets of Antwerp (Belgium) to assess the cyclist exposure to ultrafine particles (UFP) and black carbon (BC), revealing a high variability of both pollutants between and within streets level. One of the major drawbacks of mobile air quality monitoring is the incompleteness of temporal and spatial coverage making their data more suitable for measuring personal exposure than assessing air pollutants distribution across a city (Xie et al., 2017).

As people can spend up to 90% of their time in enclosed environments (Klepeis et al., 2001), indoor air quality monitoring is an important field when assessing human exposure to pollution, which is now possible with low-cost sensors. Mohammadyan et al. (2017) assessed the relationship between indoor and outdoor particulate air pollution at six primary schools in Sari (Iran) and identified the total area of windows and the number of students in a classroom as predictors for PM_1 levels.

The AoT project is a perfect example of ubiquitous monitoring where more than a hundred low-cost sensors are deployed across Chicago, allowing the assessment of air quality at near real-time and at a micro-scale level. Moreover, we can highlight two major differences with the previously mentioned studies: First, the project is not planned to be time limited and it will become a permanent source of air quality measurements for the city of Chicago. Moreover, it aims to have strong educational purposes, starting with the public solicitation to identify community priorities and thus optimize the monitoring network design. An eight-week course "Lane of Things" has also been proposed in the Chicago's Lane Tech High School where students have learned about computer science concepts and have been encouraged to gather environmental data in their school such as noise level in hallways or humidity in gyms (Thornton, 2018).

Encourage public participation in environmental monitoring can both increase the amount of air quality observations collected and promote citizen's involvement in environmental protection (Gouveia et al., 2004), and it became easier with the emergence of low-cost sensors.

To our knowledge, at least two volunteer monitoring programs are collecting air quality observations in Chicago.

The *Shared Air / Shared Action (SA2)* initiative is working with community-based organizations in four Chicago neighborhoods (Little Village, Southeast Side, Riverdale Community Area, and South Loop) to help residents collecting their own data about the pollutants they breathe using a system of both mobile and stationary monitors (Delta Institute, n.d.). This 3-years research project (2016-2019) received one of the six grants provided by the U.S. EPA through

the Science to Achieve Results (STAR) program for "Air Pollution Monitoring for Communities" (2014).

Another promising initiative is the *AirCasting* open-source platform which allows users to record, map and share health and environmental data using their smartphone ("AirCasting," n.d.). Several sensors devices [4] can be connected to the *AirCasting* application and georeferenced measurements (using the phone localization) collected during fixed or mobile sessions are displayed in real-time on the user's phone. For instance, the *AirBeam* is a portable, palm-sized air quality monitor that uses a light scattering method to estimate the number of ultrafine particles ($PM_{2.5}$) in the air (Figure 5).

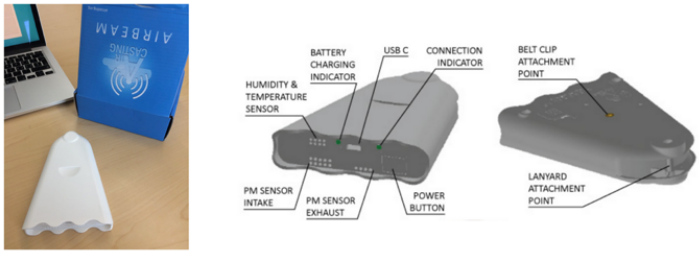


Figure 5: AirBeam sensor

The *AirCasting* platform permits to access to air quality data collected by the entire *AirCasting* community. Data corresponding to a specific mobile or fixed session can be directly visualized on the website or retrieved as JSON format through the *AirCasting* API. Another option is to use the CrowdMap that average all the *AirCasting* contributor's measurements for a given time range over squared areas.

Naturally, the shift of how air quality monitoring is done and who is collecting observations creates new challenges and unresolved issues (Clements et al., 2017; Gouveia et al., 2004).

First of all, it is important to note that even if some low-cost air quality sensors show a reasonable correlation with FEM/FRM monitors ($r^2 \sim 0.7$), testing results of a recent study from the U.S. EPA's Office of Research and Development (ORD) (Williams, 2018) have shown that performance was widely variable among low-cost sensors. Thus, in their actual state, low-cost sensors cannot be used for regulatory or compliance purposes (Clements et al., 2017).

As there are no Quality Assurance (QA) protocols or industry standardization in place yet (Clements et al., 2017), data coming from low-cost sensors are in a variety of output formats, sometimes without data labels, units, metadata or information about data accuracy. Not considering these issues can substantially affect the quality of the models used and the reliability of the findings.

[4] Arduino-powered AirCasting Air Monitor (temperature, humidity, carbon monoxide and nitrogen dioxide gas concentrations), Zephyr BioHarness 3 (heart rate, heart rate variability, R to R, breathing rate, activity level, peak acceleration and core temperature measurements), Zephyr HxM (heart rate measurements), Arduino-powered AirBeam2 (temperature, humidity, particulate matter concentrations)

Therefore, the integration of distributed low-cost sensors data to traditional air quality measurements could greatly enhance the knowledge in urban air quality but this task is complex and requires the implementation of an efficient environmental information system. The design of such infrastructure is presented in the following section.

2.3 Information systems for environmental data

A fundamental characteristic of environmental data is to be spatially distributed. If we consider a transport infrastructure, land use characteristics or air quality data, the information we can infer from these datasets will not be valuable if we do not know where it is happening ("Environmental spatial data," 2014). In order to produce valuable knowledge from environmental data, an SDI must be implemented (Awange & Kyalo Kiema, 2013).

Before the democratization of internet, environmental data were stored using operating system files, usually in several physical locations and in different standards or formats, making extremely difficult the access and the use of data (Bocher and Symposium, 2012; Ramakrishnan and Gehrke, 2003). The apparition of Database Management Systems (DBMS) was revolutionary in terms of data management as it brings a solution to several requirements exposed above. First, they provide a virtually unlimited data storage capacity compared to operating system files storage, the use of indexes can speed up the retrieval process and using a database as central data repository avoids data replication and the propagation of errors. In a DBMS, a data model (e.g. hierarchical model, network model, relational model, object-oriented model and object-relational model) describes how data will be stored, accessed and updated in the system (Awange & Kyalo Kiema, 2013). It allows to link and integrate different data sources and define complex relationship between them.

Relational databases are the simplest choice for data storage, even if the flexibility of the NoSQL approach starts being popular, especially when it deals with Big Data. In a relational data model, data are organized as tables where the different attributes are represented as columns and where each row is a record. The different tables are related through relation sets that will be used to perform queries with the Structured Query Languages (SQL).

A relational DBMS will ensure the consistency of the data stored in the database by the enforcement of integrity constraints (IC), which consist of rules specified on a database schema that will restrict the data that can be stored in an instance (Ramakrishnan & Gehrke, 2003).

Three types of IC are inherent to the relational model :

- **Domain integrity** specifies that all the values on an attribute will correspond to the same data type.
- **Entity integrity** is ensured by specifying a primary key to each table. The attribute chosen as the primary key should have to be unique and not null.
- **Referential integrity** is a constraint defined between two tables that will ensure that an attribute, namely a foreign key, will have a matching primary key in the other table or will be null.

However, relational databases have important limitations when dealing with spatial data (Lijing Zhang & Jing Yi, 2010). Indeed, a spatial location must be stored in addition to attribute and temporal characteristics and the complex relationships between spatial entities (i.e. hierarchy, generalization, etc.) must be efficiently modeled in order to perform spatial operations. Even if an RDBMS allows the manipulation of spatial data, its basic two-dimensional table structure will cause data redundancy and poor performance in spatial operations.

Object-relational database management systems as PostgreSQL or MySQL take advantage of the flexibility of object-oriented databases while keeping the simplest approach of relational models. Furthermore, most of them provide extensions that allow the storage and manipulation of spatial data. It is the case of PostGIS for PostgreSQL.

Even if a DBMS provides time data type management, in some monitoring systems, measurements are made at almost real-time and the amount of data that need to be stored grows exponentially with the number of sensors deployed and the number of parameters measured (Dunning & Friedman, n.d.). Consequently, it can significantly affect the proper functioning of the database. To prevent the overwhelming of the database, one of the two following solutions is usually employed :

The first option is to aggregate measurements according to a specific time range (e.g. hourly, daily). However, Pope and Wu (2014) have shown that temporal scale of observation and analysis may substantially affect what air pollution patterns revealed. With a too wide time aggregation, we could miss important trends in the measured parameter and it could affect the quality of the models. Moreover, a major concern in low-cost sensors application is the detection of outliers or instrument malfunctions. If outliers are not properly identified and handled, it can significantly bias our results, especially when averaging the observations. Hence, keeping fine-grained resolution may represent an added value.

The second option is to store data in a Time Serie Database System (TSDB), optimized for queries based on a timestamp or a range of time. According Bader, Kopp, and Falkenthal (n.d.), existing TSDBs can be subdivided in four groups : TSDBs that depend on already existing DBMS to store time-series data, TSDBs that are completely independent of other DBMS for the time-series data storage despite using those for storing metadata or additional information, RDBMs that can store time-series data and proprietary TSDBs that contain all commercially or freely available TSDBs that are not open source.

TimescaleDB appeared on the market last year and it is an interesting solution belonging to the first category of TSDB. Implemented as a PostgreSQL extension, it has the advantage to let the users work with time series data as regular PostgreSQL tables but with significant improvements in terms of data ingestion, query performances and time-oriented features (Paolini-Subramanya, 2018).

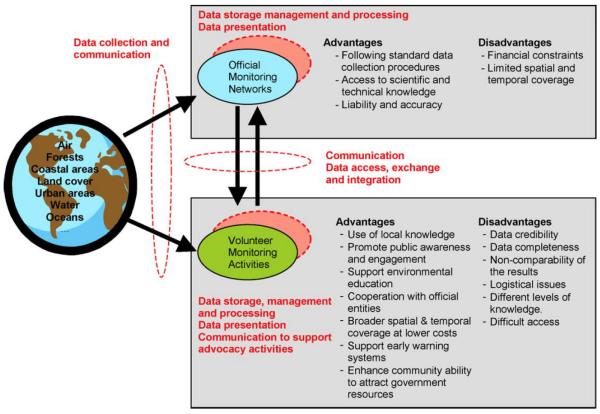


Figure 6: The role of an SDI in the integration of two different sources of distributed environmental sensors data. (*source: Gouveia et al., 2004*)

As new data will become available every day, data acquisition, database update, and views actualization must be appropriately scheduled (Awange & Kyalo Kiema, 2013).

The latter can be implemented in the form of a workflow which is by definition an pipeline execution with self-contained elements (e.g. scripts, web services, binary executables). It will allow the execution of different subtasks inherent to the SDI including data extraction, integration with other spatial resources and data processing (see an example of these tasks in Figure 6).

However, it is important to note that the design of an SDI is mostly driven by the research question, the spatial data and the various stakeholders involved in the project as highlighted in the two following definitions : according to Kuhn (2005), an SDI is "a coordinated series of agreements on technology standards, institutional arrangements, and policies that enable the discovery and use of geospatial information by users and for purposes other than those it was created for". Another definition characterizes an SDI as "a data infrastructure implementing a framework of geographic data, metadata, users and tools that are interactively connected in order to use spatial data in an efficient and flexible way" (The White House, 2002).

The spatial data that will reside in the SDI are presented in the following section.

3 Data

After having reviewed and assessed different options for distributed air quality sensors data sources in our study area (i.e. the city of Chicago), two have been selected: the first consists of criteria pollutants measurements collected by the U.S. official air quality monitoring network (i.e. AQS) and the second consists of the entire data set collected by the AoT distributed low-cost sensors.

There are three main advantages using AQS air pollution data: observations collected by the AQS instruments are highly accurate and can be used as reference values for the data recorded by the AoT low-cost sensors. Furthermore, the AQS monitoring network covers the whole country and provides historical air pollution data, which is extremely useful for determining the air quality regional background. Data characteristics, their access, and their format are detailed for each source in this section.

3.1 Air Quality System (AQS)

The AQS Data Mart is a 3.4 billion rows database containing all the information from the AQS network, namely the monitored ambient air quality data collected by EPA, state, local and tribal air pollution control agencies. It also contains metadata about each monitoring station and data quality assurance information ("AQS Data Mart," n.d.). The database is updated every Sunday [5] and is consolidated through the *AirData* website.

Another possibility would have been to use the *AirNow* website which provides forecasts and near real-time observed air quality observations from over 2,000 monitoring stations. However, as the *AirNow* primary purpose is to report the current and today's forecast Air Quality Index (AQI), mostly based on O_3 and $PM_{2.5}$ measurements, only these two criteria pollutants had a consistent dataset. Furthermore, the disadvantage of using real-time data is that only preliminary quality assessments are performed, thus they are not fully verified and validated through the EPA's AQS procedure ("About AirNow—AirNow.gov," n.d.). In this project, AQS data will serve as reference values, that is why it was more logical to consider *AirData* as our only EPA's data source and favor the data quality at the expense of temporal frequency.

The structure of the AQS database can be stated as: "Sites contain monitors which contain raw data, summary data, precision and accuracy data" (Tec, 2011). Thus, we can identify three components that we will be interested in :

- A **site** is identified by the state and the county to which it belongs and by a 4-digits number. Location data (coordinates, elevation, address) and administrative information (reporting agency name, years of monitoring, etc.) are also available.
- A **monitor** does not refer to a specific piece of equipment but to the pollutant measured by a specific device at a site. Thus, a monitor is

[5] Even if AQS is updated practically every day as reporting agencies have data ready to submit, data is required to be submitted by the end of the calendar quarter after the quarter in which it was collected. On May 1st, all the data for the prior year should be complete and correct.

uniquely identified by its site elements, a pollutant code and a parameter occurrence code (POC) that is used in case of a pollutant is measured by several devices at the same location. Information relative to the reporting agency and the collection method is stored.

- **Raw data** values are the pollutant levels reported by a monitor (e.g. sample). A sample is uniquely identified by the site and monitor information, and the timestamp at which it has been reported. A qualifier code may inform the user about natural or anthropogenic events that explain high, low or null raw data values.

Sites and monitors metadata are available as pre-generated CSV files in the AirData website and hourly, 8-hour average, daily and annual summaries data by pollutant are also available. AQS raw data can also be accessed through the *AirData* API where a query to the *\rawData* service returns the data collected by the AQS monitors. Several parameters (Table 2) may be assigned to the query to filter the data according to the site location, the measured parameter, the sample date, etc.

The use of the API for sample data request was preferred as it provides more flexibility and the data are updated weekly. *AirData* offers three data output formats depending on the user needs, and the AQCSV format, which is fully described in the AQS documentation, was the more convenient for this project as it is easy to link with the metadata files.

Query Parameter	**Description**
user	user ID
pw	User password
format	DMCSV (Full descriptions rather than codes and abbreviations), AQCSV (using Air Quality System codes), AQS (Pipe separator |)
pc	Parameter class (CRITERIA for criteria pollutants, CORE_HAPS for Urban Air Toxic Pollutants, MET for meteorological parameters)
param	5 digit AQS parameter code
bdate	begin date of data (YYYYMMDD format)
edate	end date of data (YYYMMDD format)
cbdate	return only values that changed after this date (YYYMMDD format)
cedate	return only values that changed before this date (YYYMMDD format)
state	2 digits FIPS code for the state
county	3 digits FIPS code for the county
site	4 digits AQS ID number
cbsa	5 digit Census code for the Core Based Statistical Area
csa	3 digit Census code for the Consolidated Statistical Area
minlat	Minimum Latitude
maxlat	Maximum Latitude
minlon	Minimum Longitude
maxlon	Maximum Longitude
dur	AQS duration code indicating sampling interval used for monitoring activities
frmonly	Y for Federal Reference (and Equivalent) Methods only, N will return data that is not FRM/FEM

Table 2: List of query parameters for AQS raw data request ("AQS Data API documentation," n.d.)

In this project, AQS data aim to serve as a reference for the AoT project both for data quality and value range (e.g. regional background). Hence, AQS raw data were filtered according to the monitoring location and the measured parameters to match AoT data.

Definition of the study area

Even if the AoT monitors are solely located in Chicago (Cook County), we decided to extend the bounding box area to the surrounding counties while extracting AQS data (Figure 7). The presence of heavy industries along the Lake Michigan, especially at the North-West of the State of Indiana can have a significant role on the pollutant spatial variation in Chicago, especially with the meteorological specificities of this region (e.g. wind and lake effect).

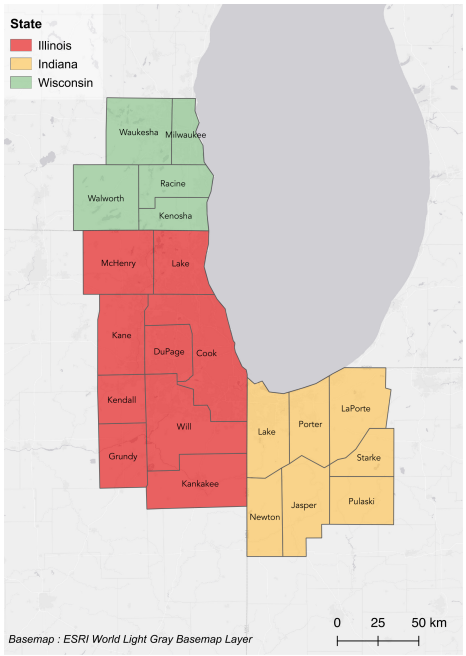


Figure 7: Counties considered for AQS raw data extraction

Selection of the parameters

Only the hourly observations relative to criteria pollutants (e.g. CRITERIA class code, see Table 3) have been extracted from the AQS database. Hourly was the smallest sampling frequency of AQS monitors and extracting more parameters was not considered useful as only criteria pollutants are measured by AoT sensors. However, depending on the future needs of the Air Quality project, additional information might be interesting, for example, the composition of particulate matter ($PM_{2.5}$ speciation). In this case, the API query would have to be modified to add another parameter class (e.g. CSN DART class code for particulate matter speciation).

The three AQS data sets (i.e. hourly observations for criteria pollutants, monitors and sites metadata) are downloadable in a CSV format and their respective structure is shown in Appendix (Figure 14, p. 46).

Pollutant	**Parameter Code**
Carbon Monoxide (CO)	42101
Lead (TSP) LC	14129
Lead PM10 LC FRM/FEM	85129
Nitrogen Dioxide (NO_2)	42602
Ozone (O_3)	44201
PM_{10} Total 0-10 μm STP	81102
$PM_{2.5}$ Local Conditions	88101
Sulfur Dioxide (SO_2)	42401

Table 3: Parameter codes for criteria pollutants

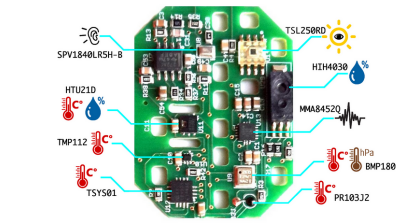
3.2 Array of Things (AoT)

The structure of the AoT data sets is composed of three parts, similarly to the structure of the AQS database:

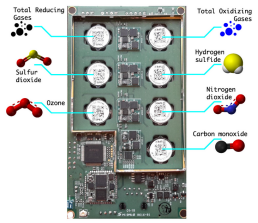
- A **node** is a monitoring network endpoint. More specifically, it gives information about the Waggle sensor platform as a whole including its geographical location. It is uniquely identified by an ID.
- A **sensor** is the device used to collect an environmental parameter and is uniquely identified by the subsystem it belongs, the sensor name and the measured parameter. A node is composed of several sensors.
- **Raw data** are the values recorded by a sensor (i.e. observation). An observation or sample is uniquely identified by the time at which the measurement was reported, the sensor used, the measured parameter and the node where the sampling was made.

As mentioned above, each sensor is part of a subsystem (i.e. board) in the Waggle architecture. The most commons are the Metsense board 8a that contains meteorological sensors, the Chemsense board 8b that contains all the air quality sensors except PM sensors, the Lightsense board 8c that contains sensors measuring light intensity and the Alphasense 8d which is the PM sensor (for PM_{10}, $PM_{2.5}$, and PM_1 measurements). Even if we will mainly use air quality data for the Air Quality Project, all the parameters measured by the AoT sensors will be stored in the infrastructure as other information including meteorological observations could be extremely useful (i.e. as spatio-temporal predictors).

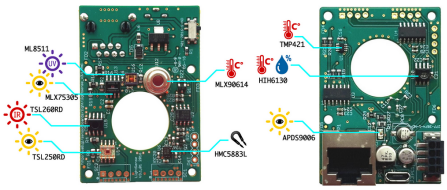
It is important to note that not all the sensors are present in each node. For instance, only 20% of the nodes contain Particulate Matter sensors (i.e. Alphasense subsystem) due to its expensive cost. Furthermore, more than half of the sensors composing the Waggle platform are measuring system parameters as internal temperature or humidity. For instance, too much humidity could seriously affect the proper functioning of air quality sensors, especially the ones measuring Particulate Matter (Williams, 2018).



(a) Metsense (also named Airsense)



(b) Chemsense



(c) Lightsense

(d) Alphasense

Figure 8: Common Waggle subsystems
(source: AoT, https://github.com/waggle-sensor/sensors/blob/develop/README.md)

Ultimately, AoT data are supposed to be available in two different access modes: in bulk download through the City of Chicago's Open Data Portal or with custom geospatial and temporal queries through the Plenar.io API.

A preliminary version of the Plenar.io API has been released in May but it is not fully operational. Among other issues, data are not up-to-date and temporal parameters in queries do not seem to work. These two accesses should not be operational before the end of the AoT sensors calibration stage.

Meanwhile, the ANL who is handling the AoT engineering part (e.g. sensors calibration and reparation, data collection, etc.) has provided an intermediary access to AoT data via a File Browser and it is the data source used in this thesis, being considered as the most reliable for now.

Specifically, the bulk download concerns two kinds of sensors data sets: the *Complete* data set contains the raw information sent by all the sensors and the *Public* data set is a subset of the *Complete* data set with only the raw data recorded by calibrated sensors. In other words, measurements values from the *Public* data set are supposed to be consistent, but a test analysis has revealed some inconsistencies in these data. Hence, we have decided to work with the complete data sets, which give an overview of the final structure we could have once all the sensors will be calibrated.

The *Public* and *Complete* data sets are either available for a specific node or for the complete monitoring network but they gathered all the data collected since the beginning of the project. As we are not interested in a specific node but more by a snapshot of the overall monitoring network, we used the second option. The three AoT data sets (i.e. raw data, nodes and sensors metadata) are available in a CSV format and their respective structure is shown in Appendix (Figure 15, p. 46).

4 Spatial Data Infrastructure

The SDI designed in this project is composed of three elements :

- The **spatial database** will store data and corresponding metadata from the two different sources of air quality sensors data (i.e. AQS and AoT).
- Several **scripts** will handle the data extraction, the data standardization and the insertion in the database.
- The **cron** job, a time-based job scheduler in Unix-like computer operating systems, will allow the automation of the workflow at regular intervals.

Compared to other spatial data resources, designing an SDI for distributed sensors data is complex for two reasons: first, the sampling frequency of environmental sensors could range from hourly to near real-time resolution which requires a frequent update of the SDI. Secondly, data are collected simultaneously at different locations, up to a hundred depending on the size of the monitoring network, which results in a large amount of data that need to be efficiently stored in the infrastructure.

The creation of a spatial database will ensure the efficient storage of distributed monitoring data and the cron job will ensure that the data stored are up-to-date.

4.1 Spatial database

4.1.1 Data model

AQS and AoT sensors data consist of three data sets each (i.e. observations, sites metadata and sensors/monitoring metadata), all available in a CSV format. For clarity, they are stored in the spatial database as two different schemas.

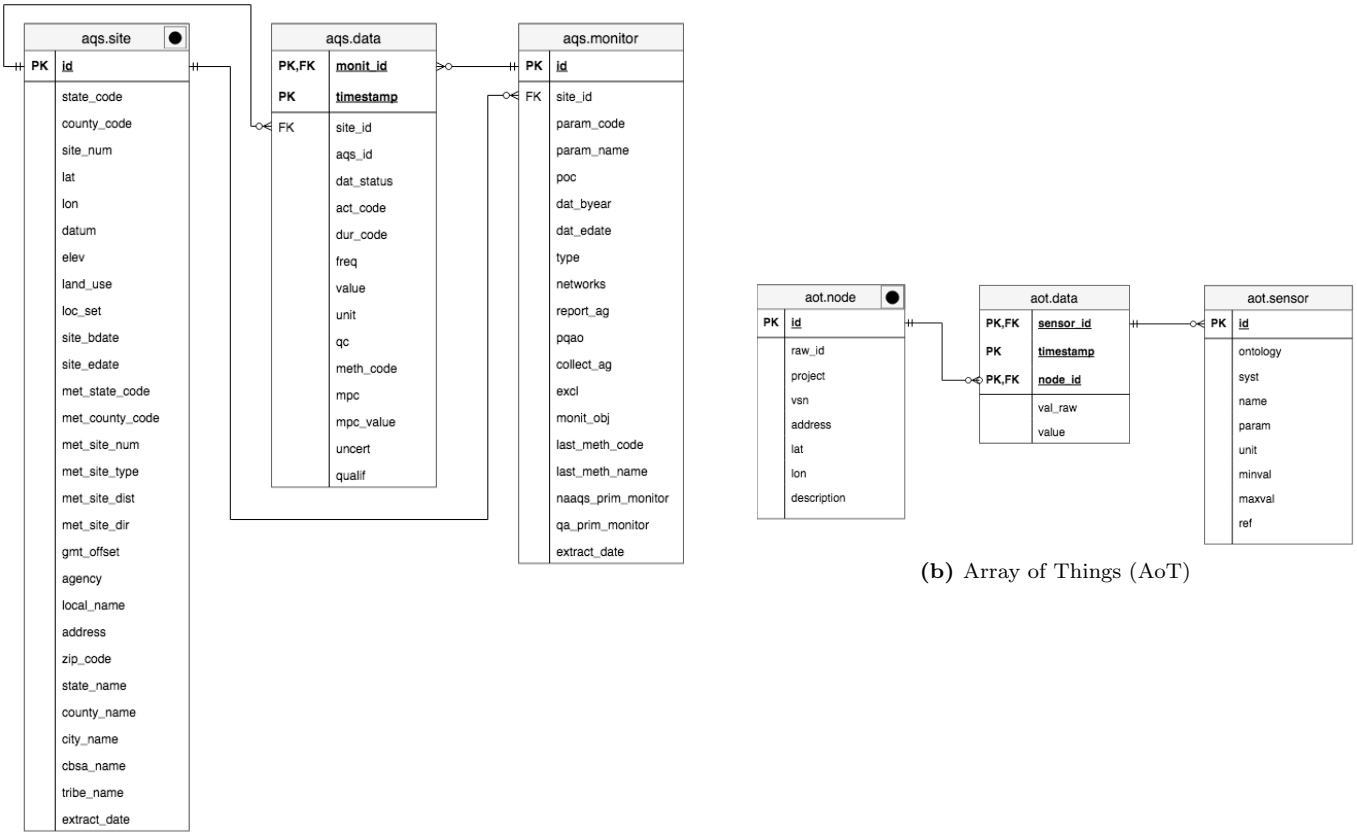
The Entity Relationship Diagram (ERD) for each data source, providing a high-level representation of the database structure is presented in Figure 9.

The main idea while designing the ERD was to stay as close as possible from the original data sets structure. Hence, it can be summarized as follows: each data source (i.e. AQS, AoT) is an entity set and the different data sets (i.e. site, monitor, data for AQS and node, sensor, data for AoT) are the entities composing the entity set.

The AQS entity set is presented in Figure 9a. Monitoring data (*aqs.data* table) are uniquely identify by the time at which the measurement was recorded (*timestamp* attribute) and the sampling characteristics (*monit_id* attribute).

A monitor instance can record zero or more data but a data instance must be recorded by exactly one monitor. Similarly, a data instance must be recorded at one site but many data can be recorded at the same site. Thus the data table relates to the site table and to the monitor table via many-to-one relationships.

Several changes can be observed compared to the structure of the data sets extracted from the AQS and presented in Appendix (Figure 14, p. 46). For instance, shorter names have been assigned to the different attributes and redundant information has been removed (e.g. State, County, City, CBSA and



(a) Air Quality System (AQS)

(b) Array of Things (AoT)

Figure 9: Entity Relationship Diagrams for the air quality sensors data

Tribe name that appeared both in the monitor and in the site data set) in order to simplify data manipulation.

Another important improvement in the data storage is the creation of a serial ID for the *site* table (*site_id* attribute) and the *monitor* table (*monit_id* attribute). As explained in the Data section [3.1], three attributes were necessary to uniquely identify a site and five for a monitor, which substantially slowing down data manipulation.

The AoT entity set is presented in Figure 9b.

Reported data (*aot.data* table) are uniquely identified by the time at which the measurement was recorded (*timestamp* attribute), the sensor identifier (*sensor_id* attribute) and the identifier of the node where the measurement was collected (*node_id* attribute). For some parameters, the value recorded by the sensor is not in a convenient unit for the user and is converted (e.g. internal temperature in the chemsense board). If it is the case, the value in the raw unit would be saved in the *val_raw* attribute. However, the unit recorded in the *aot.sensor* table corresponds to the *value* attribute and the *val_raw* attribute is an additional information that should not be used in the modeling stage.

As for the AQS schema, the data table relates to the node and the monitor

tables via many-to-one relationships. Indeed, an observation is recorded at exactly one node by a sensor, but a sensor can record several observations and several observations can be collected at the same node.

The number of data generated every day by the AoT sensors is colossal. In fact, measurements are taken every 30-seconds approximately for all the environmental sensors in the Waggle platform (approximately 50) at the 90 operating nodes, resulting to more than 6 millions observations a day. Compared to the initial structure of the data sets (Figure 15 in Appendix, p. 46), three major changes were brought to optimize the database performance: First, a new identifier was created for the node relation. Initially, the node identifier was a long character (e.g. *001e0610ba8f*) but indexing on this type of data is much slower than on integers values. Concerning the sensor relation (*aot.sensor* table), three attributes were necessary to uniquely identify an instance. Hence, a unique identifier was assigned to each combination. This operation has permitted to remove three sensor attributes in the data relation and replace them with a single identifier (*sensor_id* attribute).

Furthermore, some of the attributes names were slightly modified for the user comprehension. The creation of tables according to the ERD was done through SQL commands. The two scripts handling the creation of the database are shown in Appendix (`create_aqs.sh` (Listing 2, p. 46) for the AQS schema and `create_aot.sh` (Listing 3, p. 49) for the AoT schema).

4.1.2 Data integrity

The temporal aspect of the data stored in the spatial database (*aqs.data* and *aot.data*) is managed by defining a special data type to temporal attributes. PostgreSQL supports a TIMESTAMP format, which is assigned to the *timestamp* of both *aqs.data* and *aot.data* tables. Otherwise, all the attributes stored in the database were assigned to TEXT format at the exception of the primary keys and foreign keys. As no data cleaning is done before the data ingest in the spatial database, the objective was to prevent the interruption of the process due to wrong data formats.

Primary keys are represented in bold with the label **PK** in the ERD. They are specified when the tables are created with the following command (`create_aqs.sh`, Listing 2 in Appendix, p. 46).

```
PRIMARY KEY (id)
```

Concerning the AQS data table, the site characteristic (*monit_id* attribute) was not considered as a primary key because the monitor entity already informs about the monitoring site with the *site_id* attribute.

Foreign keys are identified by the label **FK** in the ERD and they are added to the different tables with the following SQL command (see Listing 2 and Listing 3 in Appendix, p. 46 and p. 49 respectively).

```
ALTER TABLE aqs.monitor
ADD FOREIGN KEY (site_id) REFERENCES aqs.site(id);
```

4.1.3 Spatial component

As discussed before, one of the main characteristics of the air quality sensors data is their spatial component that is indicated by the point shape entity pictogram in the ERD. The use of pictograms allows to easily identify spatial entities and their spatial attribute (e.g. Point in Figure 9) by minimizing the clutter of the ER diagram.

In both the AQS and AoT data sets, the spatial component corresponds to the latitude and longitude of the site where the sample was taken. Without a spatial extension, these attributes are considered as text fields in the PostgreSQL database.

The following operations were necessary to convert the database into a spatial database :

1. Add the PostGIS extension to the PostgreSQL database

```
CREATE EXTENSION IF NOT EXISTS postgis;
```

2. Add a geometry column to the tables containing the monitoring site latitude and longitude (i.e. *aqs.site* and *aot.node*). The *4326* argument corresponds to the EPSG code for the latitude and longitude coordinates on the WGS84 reference ellipsoid.

```
ALTER TABLE aqs.site ADD COLUMN geom geometry(POINT,4326);
ALTER TABLE aot.node ADD COLUMN geom geometry(POINT,4326);
```

3. Add a spatial index (Generalized Search Tree) to geometry columns, that will speed up spatial queries.

```
CREATE INDEX idx_geom ON aqs.site USING GIST(geom);
CREATE INDEX idx_geom ON aot.node USING GIST(geom);
```

4. Create a 2D point geometry from the latitude / longitude attributes

```
UPDATE aot.node SET geom = ST_SetSRID(ST_MakePoint("lon"::
    numeric,"lat"::numeric),4326) WHERE geom IS NULL;
UPDATE aqs.site SET geom=ST_SetSRID(ST_MakePoint("lon"::
    numeric,"lat"::numeric),4326) WHERE geom IS NULL;
```

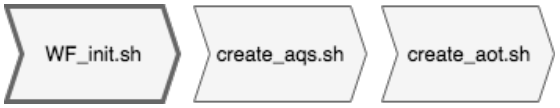
4.2 Automated workflow

The workflow can be divided into three parts according to the frequency of execution and the "big picture" is presented below. The `WF_init` (Listing 1 in Appendix, p. 44), `WF_aqs` (Listing 4 in Appendix, p. 51) and `WF_aot` (Listing 7 in Appendix, p. 54) bash shell scripts are contained environment that executes the following scripts, sometimes in addition to more general tasks. With this chained configuration, we can define a cron job only on the first element and

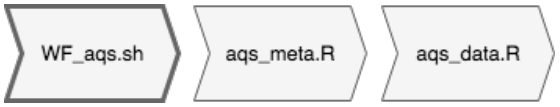
the entire workflow will be executed.

For the cron job to work and also to guarantee access to the database to the Air Quality Project collaborators, the entire infrastructure was built on a server. The Research Computing Center provides to the University of Chicago researchers an exclusive access to Midway, a high-performance computing cluster. The cluster allows to centralize the data storage and guarantees the fast computation of the workflow's tasks. It can be accessed through the command line with an SSH protocol.

```
ssh aladoy@la2.rcc.uchicago.edu
```



(a) Workflow to initialize the database



(b) Weekly workflow for AQS data

(c) Daily workflow for AoT data

Figure 10: Overview of the different workflows

All the scripts were written using the Bash or R programming languages.

The first part of the workflow (Figure 10a) is executed only once and aims to build the database in the way described in the previous section. More specifically, the `WF_init.sh` (Listing 1 in Appendix, p. 44) shell script starts the execution of two scripts: `create_aqs.sh` (Listing 2 in Appendix, p. 46), which handles the creation of the AQS schema, and `create_aot.sh` (Listing 3 in Appendix, p. 49), which handles the creation of the AoT schema.

PostgreSQL is widely supported by other programming languages which facilitate the connection with third-party applications.

More specifically, two client interfaces will be used in the workflow to interact with the spatial database :

- The psql interface is used in the bash shell scripts

```
export PGPASSWORD= **********
psql -h la2.rcc.uchicago.edu -p 5432 -U anais -d airchicago
```

- The RPostgreSQL driver is used in the R scripts

```
dbConnect(drv=RPostgreSQL::PostgreSQL(),host = "la2.rcc.
    uchicago.edu", port = 5432,user= "anais",password =
    "**********",dbname="airchicago",":memory:")
```

4.2.1 Weekly update of AQS data

The second part of the workflow (Figure 10b) is dedicated to AQS data. `WF_aqs.sh` (Listing 4 in Appendix, p. 51) is a contained environment that starts the execution of the `aqs_meta.R` (Listing 5 in Appendix, p. 51) and `aqs_data.R` (Listing 6 in Appendix, p. 52) scripts, which will handle the extraction and insertion of new information in the database at a regular frequency.

As the AQS Data Mart database is updated every Sunday, the minimum frequency for the execution of the AQS workflow was weekly and the cron job syntax is shown below :

```
30 00 * * 1 cd Scripts && sh WF_aqs.sh >> log/cron_aqs.log
```

This command line schedules the execution of the bash shell script `WF_aqs.sh` (Listing 4 in Appendix, p. 51) which is located in the folder *Scripts* every Monday at 00:30 and outputs are stored in a log file named `cron_aqs.log`.

In order to update the AQS metadata, the `aqs_meta.R` (Listing 5 in Appendix, p. 51) script executes the following operations :

1. Download the CSV files containing sites and monitors metadata from the *AirData* website
2. Transform the data frames according to the structure defined in the ERD (Figure 9a)
3. Retrieve the sites already present in the database and insert potential new sites

```
sites_db <- tbl(con, dbplyr::in_schema("aqs", "site")) %>%
    distinct(id,state_code, county_code, site_num) %>% collect
    ()
new_sites <- anti_join(sites, sites_db, by=c("state_code","
    county_code","site_num"))
dbWriteTable(con, c("aqs","site"),new_sites,row.names=F,append
    =T,temporary=F)
```

4. The same operation is conducted for the AQS monitors. As the *state_code*, *county_code* and *site_num* attributes are not stored in the *aqs.monitor* table, an additional operation is required to retrieve this information from the *aqs.site* table

```
sites_db <- tbl(con, dbplyr::in_schema("aqs", "site")) %>%
    distinct(id,state_code, county_code, site_num) %>% collect
    ()
monitors_db <- tbl(con, dbplyr::in_schema("aqs", "monitor"))
    %>% distinct(id,site_id, param_code, poc) %>% collect()
monitors <- inner_join(sites_db, monitors, by=c("state_code","
    county_code","site_num")) %>% rename(site_id=id)
```

```
new_monitors <- anti_join(monitors, monitors_db, by=c("site_id
    ","param_code","poc"))
new_monitors <- new_monitors %>% select(-c(state_code,
    county_code,site_num,lat,lon,datum,local_site_name,address
    ,state_name,county_name,city_name,cbsa_name,tribe_name))
dbWriteTable(con, c("aqs","monitor"),new_monitors,row.names=F,
    append=T,temporary=F)
```

Then, new data collected by the AQS monitoring network are extracted and appended to the *aqs.data* table with the `aqs_data.R` (Listing 6 in Appendix, p. 52) script according to the following procedure :

1. Extract last week data via the *AirData* API, where data are filtered to a subset of counties surrounding Chicago (Figure 7) and a subset of pollutants (Table 3).

2. Retrieve site and monitor identifiers from which the sample was taken and store the resulting information in new columns (*site_id* and *monit_id*)

```
sites_db <- tbl(con, dbplyr::in_schema("aqs", "site")) %>%
    distinct(id, state_code, county_code, site_num) %>%
    collect()
data <- inner_join(sites_db, data, by=c("state_code","
    county_code","site_num")) %>% rename(site_id=id)

monitors_db <- tbl(con, dbplyr::in_schema("aqs", "monitor"))
    %>% distinct(id, site_id,param_code, poc) %>% collect()
data <- inner_join(monitors_db, data, by=c("site_id","
    param_code","poc")) %>% rename(monit_id=id)
```

3. Transform the data frame according to the structure defined in the ERD (Figure 9a)

4. Insert new data in the *aqs.data* table

```
dbWriteTable(con,c("aqs","data"),data,row.names=F,append=T,
    temporary=F)
```

4.2.2 Daily update of AoT data

The third part of the workflow (Figure 10c) is dedicated to AoT data.

Even if data are collected at near real-time, a daily frequency was considered sufficient for the Air Quality Project needs.

Similarly to the syntax of the AQS cron job, the following command schedules the execution of the `WF_aot.sh` (Listing 7 in Appendix, p. 54) script located in the *Scripts* folder every day at 00:30 and save the output in a dedicated log file.

```
30 00 * * * cd Scripts && sh WF_aot.sh >> log/cron_aot.log
```

Contrary to the AQS workflow, all the scripts constituting the AoT workflow were written using the Bash programming language.

The `WF_aot.sh` (Listing 7 in Appendix, p. 54) will first start the `aot_extract.sh` (Listing 8 in Appendix, p. 54) execution to extract yesterday's data collected by the AoT sensors and corresponding metadata. Then, it will start the execution of the `aot_insert.sh` (Listing 9 in Appendix, p. 54) script to insert new data and update the database.

More specifically, the following tasks are implemented in the `aot_extract.sh` (Listing 8 in Appendix, p. 54) script :

1. Download the complete dataset from the ANL website

```
wget http://www.mcs.anl.gov/research/projects/waggle/downloads
    /datasets/AoT_Chicago.complete.latest.tar
tar xvf AoT_Chicago.complete.latest.tar
```

2. From the resulting data zip file, extract data corresponding to yesterday's date and save the output in a CSV file

```
gunzip -c AoT_Chicago.complete.$(date +'%Y-%m-%d')/data.csv.gz
     | (head -1 && grep 'date --date='yesterday' +'%Y/%m/%d'')
     > AoT_Chicago.complete.$(date +'%Y-%m-%d')/yesterday.csv
```

The algorithm used to update AoT metadata (i.e. nodes and sensors tables) is very similar to the one used for AQS metadata update. However, the implementation is done using the Bash programming language.

Below are explained the operations allowing the nodes metadata update (for sensors metadata update, see Listing 9 in Appendix, p. 54).

1. Create a temporary table with an identical structure to the nodes data set

```
CREATE      TEMP   TABLE tmp(
  raw_id       TEXT,
  project      TEXT,
  vsn          TEXT,
  address      TEXT,
  lat          TEXT,
  lon          TEXT,
  description TEXT,
  PRIMARY KEY (raw_id)
  );
```

2. Copy the CSV file containing the nodes metadata in the temporary table

```
COPY tmp FROM '/var/tmp/aot_data/AoT_Chicago.complete.$(date -
    d "yesterday" +'%Y-%m-%d')/nodes.csv' DELIMITER ',' CSV
    HEADER;
ANALYZE tmp;
```

3. Insert potential new nodes in the database thanks to a JOIN operation with the *aot.node* table. Here, the *seq_node* sequence, which was initialized during the creation of AoT tables (Listing 3 in Appendix, p. 49), works like a serial id.

```
INSERT INTO aot.node(id, raw_id, project, vsn, address, lat,
    lon, description)
SELECT nextval('seq_node'), tmp.raw_id, tmp.project, tmp.vsn,
    tmp.address, tmp.lat, tmp.lon, tmp.description
FROM tmp LEFT JOIN aot.node ON tmp.raw_id=aot.node.raw_id
WHERE aot.node.raw_id IS NULL
```

4. Drop the temporary table

```
DROP TABLE tmp;
```

Regarding the daily insertion of AoT data, a temporary table is also used to store the new observations but an additional step is required to retrieve the node (*node_id*) and sensor (*sensor_id*) identifiers from the database.

```
INSERT INTO aot.data(sensor_id, timestamp, node_id, val_raw,
    value)
SELECT S.id, tmp.timestamp, N.id, tmp.val_raw, tmp.value
FROM tmp LEFT JOIN aot.sensor S ON tmp.syst=S.syst AND tmp.
    sensor=S.name AND tmp.param=S.param
LEFT JOIN aot.node N ON tmp.raw_id=N.raw_id
WHERE S.id IS NOT NULL;
```

5 Discussion

Four main challenges were encountered in the SDI design :

The first constraint was the spatiotemporal nature of environmental sensors data that required both the efficient storage of spatial data type and the update of the database as new data are collected. The first is ensured by the addition of the PostGIS extension to the PostgreSQL database which allows the storage of all spatial data type (e.g. Line, Point, Polygon, MultiLineString, MultiPolygon) and the support of various spatial operations and spatial indexes (Lijing Zhang & Jing Yi, 2010). The temporal aspect was handled with the implementation on a computing cluster of a time-based job scheduler starting a workflow specially defined for each data source, at a daily and weekly frequency for the AoT and AQS data respectively.

The second constraint was to ensure the integration of two complementary sources of air quality sensors data, namely data collected by the highly reliable official monitoring network sensors and data reported by the low-cost AoT sensors. We faced some of the challenges highlighted by Clements et al. (2017) relative to the data standardization of low-cost air quality sensors systems (e.g. standardized definition of terms, units of measurements, file format). We tried to homogenize as much as we could the attributes names and a data dictionary has been elaborated for future database users (Figure 17 in Appendix, p. 63). Concerning the monitoring site appellation, we did not change the AoT term (i.e. *node*) to match the AQS appellation (i.e. *site*) as we thought it could be confusing for the different users, especially for the communication with the AoT team. Currently, all the data sets are in CSV format but the AoT data will ultimately be available through the Plenar.io API and the latter allows the extraction of AoT data in a JSON format. Hence, it will require additional data transformations before the insertion in the spatial database.

The third constraint was to design an SDI that could be easily extendible to the future needs of the Air Quality project. For now, there are some uncertainties about the information that needs to be stored in the database, the level of aggregation of the different data and the update frequency. Hence, we did not remove any attributes of the initial datasets, except the duplicated ones and the data sets were extracted with the finer level of aggregation (i.e. hourly for AQS data and approximately 30 seconds for AoT data). Data extraction frequency was dictated by the data sources update frequency (i.e. every Sunday for AQS and every day for AoT).

The fourth constraint was relative to the important delays that occurred in the AoT project and have seriously affected the data availability, the data access, and the data format. Currently, the AoT team offers two options for the bulk download: the entire dataset since the launch of the project, growing every day of several gigabytes or the dataset for a specific node. None of the options is convenient for the daily update of the SDI and we had no other choices than downloading the entire dataset daily (up to 60GB) which requires high computational power. That justifies why all the scripts relative to the manipulation of AoT data were written in Bash as this programming language

is more efficient with large files manipulation.

As said in the Data section, the AoT dataset contains both information about the environmental parameters measured (ontology *sensing*) and the system internal parameters (ontology *system*). During the calibration phase, having information about system parameters could be useful to identify unusual values. However, as the amount of data collected every day by AoT sensors is important, we have first assessed the number of observations relative to system parameters. On the 6,141,551 observations recorded on August 3, only 79,347 were related to system observations (the R code is provided below). Thus, the AoT system parameters data were kept for the moment as they will not cause a database overwhelming.

```
sensors <- read_csv('sensors.csv') %>% filter(grepl("sensing",
    ontology))
data <- read_csv('complete_data_20180803.csv')
sensing_data <- inner_join(data, sensors, by=c('subsystem','sensor
    ','parameter'))
nrow(data)-nrow(sensing_data)
```

5.1 Strengths

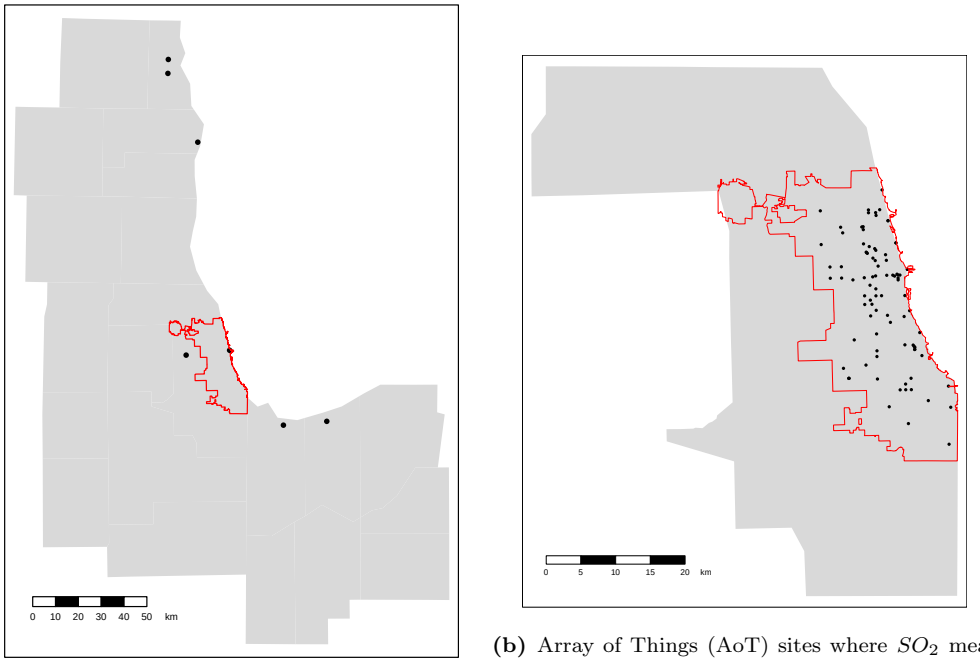
The SDI created in this project provides a powerful framework for the spatial analysis of the urban air quality in Chicago.

By combining data from a distributed low-cost monitoring network (i.e. Aot) and a reference monitoring network (AQS), we can benefit from both the high-spatial coverage offered by the AoT network and the high-quality measurements produced by the AQS sensors.

With the example of SO_2 monitoring sites, Figure 11 shows the enhancement of the regulatory monitoring network (11a) by the AoT distributed network (11b). Furthermore, the inclusion of AQS sensors data recorded in Chicago's surrounding counties allows obtaining information about the regional background and air pollutant transportation. As the AoT sensors are solely located on roadsides, the measured values will be mainly influenced by traffic conditions and having information about the regional background could be highly valuable to avoid biases in the modeling stage. Furthermore, air pollutant transportation could play an important role in the spatial distribution of air pollutants in Chicago as several heavy industries are located in the Lake Michigan south shoreline.

The infrastructure is entirely based on free and open source software, namely PostgreSQL for the DBMS and R and Bash programming languages for the different tasks operated in the workflow (i.e. data extraction, data standardization, and database update). Furthermore, the SDI is hosted on a remote server that will guarantee the proper functioning of the cron job and the database access to the different CSDS collaborators.

Beside its cost-effectiveness, the PostgreSQL provides the advantage to offer powerful client interfaces with the command line (psql) and the R environment (RPostgreSQL). Already used in the workflow to interact with the spatial



(a) Air Quality System (AQS) active sites where SO_2 measurements are collected

(b) Array of Things (AoT) sites where SO_2 measurements are collected

Figure 11: Respective locations of SO_2 AQS and AoT sensors.

database for the data insert and update process, they will offer a powerful tool to the Air Quality Project team during the modeling stage.

Furthermore, the creation of IDs, the removal of duplicated attributes and the creation of indexes will allow efficient data retrieval and data manipulation. Spatial operations will also be possible thanks to the PostGIS extension.

5.2 Weaknesses

The actual instability of the AoT project has complicated the SDI design process as we did not know the release date of calibrated data, their structure, and their accuracy. Hence, modifications in the SDI design could be required in the future.

For instance, frequent changes are done in the AoT data access and there are no guarantees that the AoT data structure will not be changed in the future. If it is the case, some additional tasks should be implemented, either by transforming new data according to the structure defined in the ERD (Figure 9b) or by creating a new table in the database.

In the literature review, we pointed out that a special attention should be paid to the amount of environmental sensors data that will be regularly added in the database. In case of distributed low-cost monitoring networks, observations are usually recorded at near real-time frequency and it can rapidly overwhelm the database, increasing exponentially with the number of nodes involved and the number of parameters measured (Bader et al., n.d.). Even if all the data

are currently efficiently stored, the daily insertion of 6 millions of observations recorded by the AoT monitoring network could become problematic.

For now, the TimescaleDB extension is considered as the best solution for the time series data storage as it could be added as a PostgreSQL extension. However, it requires a wide format for the time series data and it is currently not the case for the two data sources. The transformation into a wide format will require additional data transformation and could complicate SQL operations for data retrieval.

The other solution, also discussed in the literature review, would be to aggregate the AoT data according to a specific time interval (e.g. hourly, daily) which could vary for the different pollutants depending on the time resolution needed in the modeling stage. As no reliable data are currently produced by the AoT sensors, it was difficult to assess the efficiency of one solution or the other.

5.3 Toward a federation of Chicago air pollution monitoring systems?

The SDI implemented in this project and combining AoT and AQS air quality data is a first step toward the integration of volunteer initiatives to official monitoring network data.

In addition to providing micro-scale air quality observations that could significantly increase the quality of urban air pollution models, Gouveia et al. (2004) mentioned additional benefits of including volunteer monitoring programs data: first, it could increase environmental awareness of the public and this educational purpose is more and more included in volunteer monitoring initiatives (e.g. creation of the "Lane of Things" course as part of the AoT project). Moreover, it could offer a cost-effective solution for data collection (Aberer et al., 2010), especially for countries with limited funding.

A search of volunteer monitoring initiative that could enhance AoT observations has been conducted during the data collection stage.

The first initiative we found, namely the *Shared Air / Shared Action* (SA2) initiative (Delta Institute, n.d.), has not replied to a potential collaboration request.

The option of using *AirCasting* users data ("AirCasting," n.d.) has also been assessed. For this purpose, the CSDS ordered an *AirBeam* sensor that was configured and tested during a biking session in the south side of Chicago. Description and results of this test session are available in Appendix (Figure 16 in Appendix, p. 58). The conclusions are as follows : concerning the sensor reliability, we can observe logical trends of the different parameters (e.g. increase of the number of particles when stopping behind a public bus, lake effect on the relative humidity and temperature) but serious doubts remain about the measurements accuracy (e.g. number of $PM_{2.5}$ increasing instantaneously from 0 to 62 without noticeable extreme events). Concerning the use of data collected by the *AirCasting* community, it has to be done cautiously. First of all, it is rare that users give indications about the session configuration, for example, if they are inside or outside and it can yield to a significant bias in the results.

Furthermore, Marjovi et al. (2015) discuss the difficulties related to the analysis of spatially and temporally dynamic data (e.g. unpredictable and irregular coverage). Despite the fact we did not consider the *AirCasting* initiative as an added value for the project, this option should be re-evaluated if exposure measurements are needed (e.g. to link health data to air pollution data) in the future development of the Air Quality project. According to a recent report of the U.S. EPA Office of Research and Development (ORD) (Williams, 2018), the *AirBeam* sensor exhibits a strong correlation with the regulatory monitors in terms of hourly averages of PM concentrations.

These examples highlight the major obstacles that prevent the efficient integration of different air quality sensors data sources. First, there is an isolation of volunteer monitoring initiatives, which most of the time, focus on local issues and entirely design the data collection, data storage, and data sharing in the scope of their campaigns (Xie et al., 2017).

Furthermore, the lack of standard procedures relative to low-cost sensors makes difficult the integration of heterogeneous data sources (Clements et al., 2017). Data quality is often unknown and metadata on data sampling and collection is scarce (Gouveia et al., 2004).

To conclude, the increasing number of initiatives measuring air quality provides an exciting opportunity to get high-level resolution data but efforts should be made to maintain a good communication channel between the different stakeholders and to standardize sampling procedure and collected data formats (Phillips et al., 1999; Clements et al., 2017).

6 Conclusion

In this master thesis, a Spatial Data Infrastructure (SDI) was implemented, allowing the extraction of distributed air quality sensors data at regular intervals from two different monitoring networks and their integration in a centralized spatial database. In the context of the AoT project, a network of low-cost sensors has been deployed across the city of Chicago reporting air quality observations at near real-time and at a micro-scale level ("Array of Things," n.d.). As their analysis could provide valuable knowledge to identify air pollution drivers and support environmental policies development, the CSDS has been mandated as part of the Partnership for Healthy Cities, in order to conduct a spatial analysis of the data collected by the AoT sensors (see the implementation details in Document 7.1 in Appendix, p. 57).

Different constraints were required during the design of the SDI. Including the efficient storage and management of spatiotemporal data, the integration of disparate sources of air quality sensors data, and the automation of the process. Delays in the AoT project have prevented the access to calibrated data and to a stable data structure, which has complicated the design process.

The resulting SDI respects all the constraints defined at the beginning of the project. More specifically, the PostgreSQL/PostGIS database guarantees data scalability, the storage, and manipulation of spatial and temporal data and the centralized access of two heterogeneous air quality sensors data sources. Furthermore, a workflow composed of R and Bash scripts ensures the extraction of data from the two sources, the standardization according to the designed data model and the insertion and update of the database at a regular frequency (i.e. weekly for AQS data and daily for AoT data) thanks to a job scheduler. The SDI was entirely based on free and open source software and is located on a remote server allowing a multi-access to the different CSDS collaborators. In order to meet the future needs of the Air Quality Project, the SDI was conceived to ensure the extension to other data sources, the connection to different third-party applications including the R environment where the different models will be implemented and the data model comprehension with the elaboration of a metadata file (Figure 17 in Appendix, p. 63).

The SDI developed provides a unique framework to conduct the first analysis of the AoT distributed air quality sensors data and represents the first step toward a better knowledge of the urban air quality in the city of Chicago.

References

Aberer, K., Sathe, S., Chakraborty, D., Martinoli, A., Barrenetxea, G., Faltings, B., & Thiele, L. (2010). OpenSense: Open community driven sensing of environment. (pp. 39–42). doi:10.1145/1878500.1878509

About AirNow—AirNow.gov. (n.d.). Retrieved October 2, 2018, from https://www.airnow.gov/about-airnow

Air Pollution Monitoring for Communities. (2014). Retrieved August 3, 2018, from https://cfpub.epa.gov/ncer_abstracts/index.cfm/fuseaction/recipients.display/rfa_id/587

AirCasting. (n.d.). Retrieved August 4, 2018, from http://aircasting.org/

AQS Data API documentation. (n.d.). Retrieved July 28, 2018, from https://aqs.epa.gov/aqsweb/documents/ramltohtml.html#

AQS Data Mart. (n.d.). Retrieved July 28, 2018, from https://aqs.epa.gov/aqsweb/documents/data_mart_welcome.html

Array of Things. (n.d.). Retrieved August 12, 2018, from https://arrayofthings.github.io/

Artiola, J., Pepper, I. L., & Brusseau, M. L. (2004). *Environmental monitoring and characterization*. Elsevier.

Awange, J. L., & Kyalo Kiema, J. B. (2013). *Environmental geoinformatics*. Environmental Science and Engineering. doi:10.1007/978-3-642-34085-7

Bader, A., Kopp, O., & Falkenthal, M. (n.d.). Survey and comparison of open source time series databases, 20.

Benammar, M., Abdaoui, A., Ahmad, S., Touati, F., & Kadri, A. (2018, February 14). A modular IoT platform for real-time indoor air quality monitoring. *Sensors*, *18*(2), 581. doi:10.3390/s18020581

Bigazzi, A. Y., & Figliozzi, M. A. (2015). Roadway determinants of bicyclist exposure to volatile organic compounds and carbon monoxide. *Transportation Research Part D: Transport and Environment*, *41*, 13–23.

Bocher, E., & Symposium, O. S. G. R. (Eds.). (2012). *Geospatial free and open source software in the 21st century: Proceedings of the first open source geospatial research symposium, OGRS 2009 ; [held in nantes city, france, 8 - 10 july, 2009]*. Heidelberg: Springer. Lecture notes in geoinformation and cartography. OCLC: 794502767.

Buytaert, W., Dewulf, A., De Bièvre, B., Clark, J., & Hannah, D. M. (2016, April). Citizen science for water resources management: Toward polycentric monitoring and governance? *Journal of Water Resources Planning and Management*, *142*(4), 01816002. doi:10.1061/(ASCE)WR.1943-5452.0000641

Castell, N. [Nuria], Dauge, F. R., Schneider, P., Vogt, M., Lerner, U., Fishbain, B., ... Bartonova, A. (2017, February). Can commercial low-cost sensor platforms contribute to air quality monitoring and exposure estimates? *Environment International*, *99*, 293–302. doi:10.1016/j.envint.2016.12.007

Castell, N. [Núria], Viana, M., Minguillón, M. C., Guerreiro, C., & Querol, X. (2013). Real-world application of new sensor technologies for air quality monitoring. *ETC/ACM Technical Paper*, *16*.

Clements, A. L., Griswold, W. G., Rs, A., Johnston, J. E., Herting, M. M., Thorson, J., ... Hannigan, M. (2017, October 28). Low-cost air quality monitoring tools: From research to practice (a workshop summary). *Sensors*, *17*(11), 2478. doi:10.3390/s17112478

Clougherty, J. E., Kheirbek, I., Eisl, H. M., Ross, Z., Pezeshki, G., Gorczynski, J. E., ... Matte, T. (2013, May). Intra-urban spatial variability in wintertime street-level concentrations of multiple combustion-related air pollutants: The new york city community air survey (NYCCAS). *Journal of Exposure Science & Environmental Epidemiology*, *23*(3), 232–240. doi:10.1038/jes.2012.125

Committee on Environment, Natural Resources, and Sustainability. (2013, November). Air quality observation systems in the united states. NATIONAL SCIENCE AND TECHNOLOGY COUNCIL.

de Hoogh, K., Héritier, H., Stafoggia, M., Künzli, N., & Kloog, I. (2018, February). Modelling daily PM 2.5 concentrations at high spatio-temporal resolution across switzerland. *Environmental Pollution*, *233*, 1147–1154. doi:10.1016/j.envpol.2017.10.025

Delta Institute. (n.d.). Shared air, shared action: Empowering communities through air quality monitoring [Delta institute annual report]. Retrieved August 3, 2018, from http://delta-institute.org/annualreport/shared-air-share-action/

Downes, B. J., Barmuta, L. A., Fairweather, P. G., Faith, D. P., Keough, M. J., Lake, P., ... Quinn, G. P., et al. (2002). *Monitoring ecological impacts: Concepts and practice in flowing waters.* Cambridge University Press.

Dunning, T., & Friedman, E. (n.d.). Time series databases, 81.

Environmental spatial data: What is happening where? (2014, September 12). *European Environment Agency*. Retrieved August 15, 2018, from https://www.eea.europa.eu/articles/environmental-spatial-data-what-is

GeoDaCenter/airquality. (n.d.). Retrieved August 12, 2018, from https://github.com/GeoDaCenter/airquality

Gouveia, C., Fonseca, A., Câmara, A., & Ferreira, F. (2004, June). Promoting the use of environmental data collected by concerned citizens through information and communication technologies. *Journal of Environmental Management*, *71*(2), 135–154. doi:10.1016/j.jenvman.2004.01.009

Hatzopoulou, M., Valois, M. F., Levy, I., Mihele, C., Lu, G., Bagg, S., ... Brook, J. (2017). Robustness of land-use regression models developed from mobile air pollutant measurements. *Environmental Science & Technology*, *51*(7), 3938–3947.

Klepeis, N. E., Nelson, W. C., Ott, W. R., Robinson, J. P., Tsang, A. M., Switzer, P., ... Engelmann, W. H. (2001, July). The national human activity pattern survey (NHAPS): A resource for assessing exposure to environmental pollutants. *Journal of Exposure Science and Environmental Epidemiology*, *11*(3), 231–252. doi:10.1038/sj.jea.7500165

Kloog, I., Nordio, F., Coull, B. A., & Schwartz, J. (2012). Incorporating local land use regression and satellite aerosol optical depth in a hybrid model of spatiotemporal PM2. 5 exposures in the mid-atlantic states. *Environmental science & technology*, *46*(21), 11913–11921.

Kloog, I., Nordio, F., Coull, B. A., & Schwartz, J. (2014). Predicting spatiotemporal mean air temperature using MODIS satellite surface temperature measurements across the northeastern USA. *Remote sensing of environment*, *150*, 132–139.

Kuhn, W. (2005). Introduction to spatial data infrastructures. *Presentation held on March*, *14*, 2005.

Kumar, P., Morawska, L., Martani, C., Biskos, G., Neophytou, M., Di Sabatino, S., ... Britter, R. (2015, February). The rise of low-cost sensing for managing air pollution in cities. *Environment International*, *75*, 199–205. doi:10.1016/j.envint.2014.11.019

Lewis, A., & Edwards, P. (2016). Validate personal air-pollution sensors: Alastair lewis and peter edwards call on researchers to test the accuracy of low-cost monitoring devices before regulators are flooded with questionable air-quality data. *Nature*, *535*(7610), 29–32.

Lijing Zhang, & Jing Yi. (2010, August). Management methods of spatial data based on PostGIS. In *2010 second pacific-asia conference on circuits, communications and system* (pp. 410–413). 2010 second pacific-asia conference on circuits,communications and system (PACCS). doi:10.1109/PACCS.2010.5626962

Mannucci, P. M., Harari, S., Martinelli, I., & Franchini, M. (2015, September). Effects on health of air pollution: A narrative review. *Internal and Emergency Medicine*, *10*(6), 657–662. doi:10.1007/s11739-015-1276-7

Marjovi, A., Arfire, A., & Martinoli, A. (2015, June). High resolution air pollution maps in urban environments using mobile sensor networks. In *2015 international conference on distributed computing in sensor systems* (pp. 11–20). 2015 international conference on distributed computing in sensor systems. doi:10.1109/DCOSS.2015.32

Matte, T. D., Ross, Z., Kheirbek, I., Eisl, H., Johnson, S., Gorczynski, J. E., ... Clougherty, J. E. (2013, May). Monitoring intraurban spatial patterns of multiple combustion air pollutants in new york city: Design and implementation. *Journal of Exposure Science & Environmental Epidemiology*, *23*(3), 223–231. doi:10.1038/jes.2012.126

Mohammadyan, M., Alizadeh-Larimi, A., Etemadinejad, S., Latif, M. T., Heibati, B., Yetilmezsoy, K., ... Dadvand, P. (2017). Particulate air pollution at schools: Indoor-outdoor relationship and determinants of indoor concentrations. *Aerosol and Air Quality Research*, *17*(3), 857–864.

Paolini-Subramanya, M. (2018, January 5). TimescaleDB — time series data on postgres (done right) [Medium]. Retrieved August 8, 2018, from https://medium.com/@dieswaytoofast/timescaledb-time-series-data-on-postgres-done-right-14e5028124f0

Peters, J., Van den Bossche, J., Reggente, M., Van Poppel, M., De Baets, B., & Theunis, J. (2014). Cyclist exposure to UFP and BC on urban routes in antwerp, belgium. *Atmospheric Environment*, *92*, 31–43.

Phillips, A., Williamson, I., & Ezigbalike, C. (1999, June). Spatial data infrastructure concepts. *Australian Surveyor*, *44*(1), 20–28. doi:10.1080/00050326.1999.10441899

Pope, R., & Wu, J. (2014, September). Characterizing air pollution patterns on multiple time scales in urban areas: A landscape ecological approach. *Urban Ecosystems*, *17*(3), 855–874. doi:10.1007/s11252-014-0357-0

Ramakrishnan, R., & Gehrke, J. (2003). *Database management systems* (Third edition, international edition). OCLC: 264999030. New York: McGraw-Hill.

Stafoggia, M., Schwartz, J., Badaloni, C., Bellander, T., Alessandrini, E., Cattani, G., ... Lyapustin, A., et al. (2017). Estimation of daily PM10 concentrations in italy (2006–2012) using finely resolved satellite data, land use variables and meteorology. *Environment international*, *99*, 234–244.

Stepenuck, K. F., & Green, L. T. (2015). Individual- and community-level impacts of volunteer environmental monitoring: A synthesis of peer-reviewed literature. *Ecology and Society*, *20*(3). doi:10.5751/ES-07329-200319

Tec, L. M. I. (2011). AQS data dictionary, 425.

The Partnership for Healthy Cities (PHC). (n.d.). Retrieved August 22, 2018, from https://partnershipforhealthycities.bloomberg.org/

The White House. (2002, August 19). Office of management and budget (2002) circular no. a-16 revised.

Thornton, S. (2018, January 2). A guide to chicago's array of things initiative [Data-smart city solutions]. Retrieved August 19, 2018, from https://datasmart.ash.harvard.edu/news/article/a-guide-to-chicagos-array-of-things-initiative-1190

United Nations. (n.d.). Sustainable development goals. Retrieved from https://sustainabledevelopment.un.org/sdgs

United Nations (Ed.). (2003). *Environmental monitoring and reporting: Eastern europe, the caucasus and central asia*. New York: United Nations. OCLC: ocm53346617.

United Nations Treaty Collection. (2016, July 8). Paris agreement. Retrieved from https://treaties.un.org/pages/ViewDetails.aspx?src=TREATY&mtdsg_no=XXVII-7-d&chapter=27&clang=_en

Urbano, F., Cagnacci, F., Calenge, C., Dettki, H., Cameron, A., & Neteler, M. (2010, July 27). Wildlife tracking data management: A new vision. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *365*(1550), 2177–2185. doi:10.1098/rstb.2010.0081

US EPA, O. (2015, February 27). Overview of the clean air act and air pollution [U.s. EPA]. Retrieved July 26, 2018, from https://www.epa.gov/clean-air-act-overview

Vitolo, C., Elkhatib, Y., Reusser, D., Macleod, C. J., & Buytaert, W. (2015, January). Web technologies for environmental big data. *Environmental Modelling & Software*, *63*, 185–198. doi:10.1016/j.envsoft.2014.10.007

Weston, S. (2011). An overview of environmental monitoring and its significance in resource and environmental management. *School of Resource and Environmental Studies, Dalhousie University*.

White, J. E. (n.d.). Overview of air quality monitoring in the u.s. and the EPA AirNow program, 37.

Williams, R. (2018, April 30). New paradigm for air pollution monitoring : 2014-2018 progress report, air and energy research program, 181.

World Health Organisation. (2017). Evolution of WHO air quality guidelines: Past, present and future. WHO Regional Office for Europe.

World Health Organization. (2016). *World health statistics. 2016, 2016,* OCLC: 968482612. Retrieved August 19, 2018, from http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=1482449

World Meteorological Organization. (2018, May). Low-cost sensors for the measurement of atmospheric composition: Overview of topic and future applications.

Xie, X., Semanjski, I., Gautama, S., Tsiligianni, E., Deligiannis, N., Rajan, R., ... Philips, W. (2017, December 1). A review of urban air pollution monitoring and exposure assessment methods. *ISPRS International Journal of Geo-Information*, *6*(12), 389. doi:10.3390/ijgi6120389

Zwack, L. M., Paciorek, C. J., Spengler, J. D., & Levy, J. I. (2011). Modeling spatial patterns of traffic-related air pollutants in complex urban terrain. *Environmental Health Perspectives*, *119*(6), 852.

7 Appendix

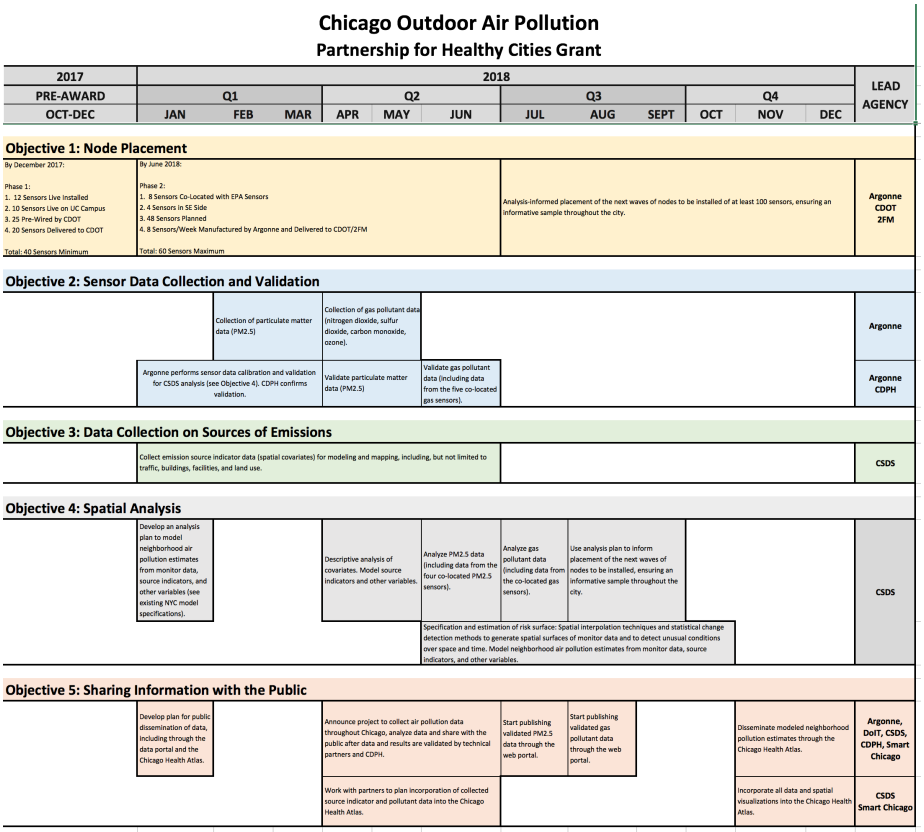
List of Figures

List of Tables

List of Documents

Listings

Figure 12: Air Quality Project roadmap for 2018



Chicago Outdoor Air Pollution
Partnership for Healthy Cities Grant

2017	2018												LEAD AGENCY
PRE-AWARD	Q1			Q2			Q3			Q4			
OCT-DEC	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEPT	OCT	NOV	DEC	
Objective 1: Node Placement													
By December 2017: Phase 1: 1. 12 Sensors Live Installed 2. 10 Sensors Live on UC Campus 3. 25 Pre-Wired by CDOT 4. 20 Sensors Delivered to CDOT Total: 40 Sensors Minimum	By June 2018: Phase 2: 1. 8 Sensors Co-Located with EPA Sensors 2. 4 Sensors in SE Side 3. 48 Sensors Planned 4. 8 Sensors/Week Manufactured by Argonne and Delivered to CDOT/2FM Total: 60 Sensors Maximum						Analysis-informed placement of the next waves of nodes to be installed of at least 100 sensors, ensuring an informative sample throughout the city.						Argonne CDOT 2FM
Objective 2: Sensor Data Collection and Validation													
		Collection of particulate matter data (PM2.5)		Collection of gas pollutant data (nitrogen dioxide, sulfur dioxide, carbon monoxide, ozone).									Argonne
	Argonne performs sensor data calibration and validation for CSDS analysis (see Objective 4). CDPH confirms validation.			Validate particulate matter data (PM2.5)		Validate gas pollutant data (including data from the five co-located gas sensors).							Argonne CDPH
Objective 3: Data Collection on Sources of Emissions													
	Collect emission source indicator data (spatial covariates) for modeling and mapping, including, but not limited to traffic, buildings, facilities, and land use.												CSDS
Objective 4: Spatial Analysis													
	Develop an analysis plan to model neighborhood air pollution estimates from monitor data, source indicators, and other variables (see existing NYC model specifications).			Descriptive analysis of covariates. Model source indicators and other variables.		Analyze PM2.5 data (including data from the four co-located PM2.5 sensors).	Analyze gas pollutant data (including data from the co-located gas sensors).	Use analysis plan to inform placement of the next waves of nodes to be installed, ensuring an informative sample throughout the city.					CSDS
						Specification and estimation of risk surface: Spatial interpolation techniques and statistical change detection methods to generate spatial surfaces of monitor data and to detect unusual conditions over space and time. Model neighborhood air pollution estimates from monitor data, source indicators, and other variables.						
Objective 5: Sharing Information with the Public													
	Develop plan for public dissemination of data, including through the data portal and the Chicago Health Atlas.			Announce project to collect air pollution data throughout Chicago, analyze data and share with the public after data and results are validated by technical partners and CDPH.			Start publishing validated PM2.5 data through the web portal.	Start publishing validated gas pollutant data through the web portal.			Disseminate modeled neighborhood pollution estimates through the Chicago Health Atlas.		Argonne, DoIT, CSDS, CDPH, Smart Chicago
				Work with partners to plan incorporation of collected source indicator and pollutant data into the Chicago Health Atlas.							Incorporate all data and spatial visualizations into the Chicago Health Atlas.		CSDS Smart Chicago

Listing 1: WF_init.sh

```
#!/bin/bash

#Initialize the database

# CONNECT TO THE DATABASE
export PGPASSWORD= ********** #Specify environment variables
RUN_ON_MYDB="psql -h la2.rcc.uchicago.edu -p 5432 -U anais -d airchicago"

$RUN_ON_MYDB <<SQL
CREATE EXTENSION IF NOT EXISTS postgis;
SQL

#Create AQS tables
sh create_aqs.sh

#Create AoT tables
sh create_aot.sh
```

Figure 13: Spatio-temporal predictors considered in the Air Quality Project

Name	Source(s)	File	Spatial Resolution	Time Resolution	Comments
Point Emission Data	NEI and CDPH	CSV	address-level	varies	Options for use: (1) Distance to nearest facility (either all or subset of high polluters), (2) Raster surface (over 100 available for Chicago area)
Area Emission Data: building heights as proxy or heating fuel	OSM, LiDar from Cook County GIS	varies	varies	varies	NEI area data not available below county-level
Road Emissions: (1) road length summary, (2) traffic volume summary	OSM, IDOT	SHP	varies	varies	Multiple proxies
Meteorological Data	NOAA monitoring stations	CSV	address-level	Daily, Monthly	few validated weather stations in metro area; weather underground data not reliabale and historical data is costly
NDVI - Greenness Index	MODIS	raster	1km, 500m, 250, grid	Monthly, Yearly	
Land Cover	SAL product, CMAP	raster	30m	2016, 2011, 2006	
Land Use	CMAP	raster	parcel	2013, 2010, 2005, 2001	
Elevation	National Land Elevation Model	raster	1m	yearly	alt: building height dataset
Demographic and SES data	ACS	CSV, SHP	tract	5-year average	

Figure 14: Structure of the Air Quality System data sets

```
Variables: 28
$ `State Code`            <chr>
$ `County Code`           <chr>
$ `Site Number`           <chr>
$ Latitude                <dbl>
$ Longitude               <dbl>
$ Datum                   <chr>
$ Elevation               <dbl>
$ `Land Use`              <chr>
$ `Location Setting`      <chr>
$ `Site Established Date` <date>
$ `Site Closed Date`      <date>
$ `Met Site State Code`   <chr>
$ `Met Site County Code`  <chr>
$ `Met Site Site Number`  <chr>
$ `Met Site Type`         <chr>
$ `Met Site Distance`     <int>
$ `Met Site Direction`    <chr>
$ `GMT Offset`            <int>
$ `Owning Agency`         <chr>
$ `Local Site Name`       <chr>
$ Address                 <chr>
$ `Zip Code`              <chr>
$ `State Name`            <chr>
$ `County Name`           <chr>
$ `City Name`             <chr>
$ `CBSA Name`             <chr>
$ `Tribe Name`            <chr>
$ `Extraction Date`       <date>
```

(a) Sites

```
Variables: 30
$ `State Code`            <chr>
$ `County Code`           <chr>
$ `Site Number`           <chr>
$ `Parameter Code`        <int>
$ `Parameter Name`        <chr>
$ POC                     <int>
$ Latitude                <dbl>
$ Longitude               <dbl>
$ Datum                   <chr>
$ `First Year of Data`    <int>
$ `Last Sample Date`      <date>
$ `Monitor Type`          <chr>
$ Networks                <chr>
$ `Reporting Agency`      <chr>
$ PQAO                    <chr>
$ `Collecting Agency`     <chr>
$ Exclusions              <chr>
$ `Monitoring Objective`  <chr>
$ `Last Method Code`      <chr>
$ `Last Method`           <chr>
$ `NAAQS Primary Monitor` <chr>
$ `QA Primary Monitor`    <chr>
$ `Local Site Name`       <chr>
$ Address                 <chr>
$ `State Name`            <chr>
$ `County Name`           <chr>
$ `City Name`             <chr>
$ `CBSA Name`             <chr>
$ `Tribe Name`            <chr>
$ `Extraction Date`       <date>
```

(b) Monitors

```
Variables: 20
$ site        <chr>
$ data_status <chr>
$ action_code <chr>
$ datetime    <dttm>
$ parameter   <chr>
$ duration    <chr>
$ frequency   <chr>
$ value       <chr>
$ unit        <chr>
$ qc          <chr>
$ poc         <chr>
$ lat         <chr>
$ lon         <chr>
$ GISDatum    <chr>
$ elev        <chr>
$ method_code <chr>
$ mpc         <chr>
$ mpc_value   <chr>
$ uncertainty <chr>
$ qualifiers  <chr>
```

(c) Data (AQCSV format)

Figure 15: Structure of the Array of Things complete data sets

```
Variables: 7
$ node_id     <chr>
$ project_id  <chr>
$ vsn         <chr>
$ address     <chr>
$ lat         <dbl>
$ lon         <dbl>
$ description <chr>
```

(a) Nodes

```
Variables: 8
$ ontology    <chr>
$ subsystem   <chr>
$ sensor      <chr>
$ parameter   <chr>
$ hrf_unit    <chr>
$ hrf_minval  <int>
$ hrf_maxval  <int>
$ datasheet   <chr>
```

(b) Sensors

```
Variables: 7
$ timestamp <chr>
$ node_id   <chr>
$ subsystem <chr>
$ sensor    <chr>
$ parameter <chr>
$ value_raw <chr>
$ value_hrf <chr>
```

(c) Complete data

Listing 2: create_aqs.sh

```
#!/bin/bash

#Create the database infrastructure for Air Quality System (AQS) data (creation of tables, constraints definition)

# CONNECT TO THE DATABASE
export PGPASSWORD= ********** #Environment variables
RUN_ON_MYDB="psql -h la2.rcc.uchicago.edu -p 5432 -U anais -d airchicago"

$RUN_ON_MYDB <<SQL
CREATE SCHEMA if not exists aqs;
SQL

#Create table for AQS data
$RUN_ON_MYDB <<SQL
CREATE  TABLE                           aqs.data     (
```

```
        monit_id                        INTEGER    NOT NULL,
        timestamp                       TIMESTAMP    NOT NULL,
        site_id                         INTEGER    NOT NULL,
        aqs_id                          TEXT,
        dat_status                      TEXT,
        act_code                        TEXT,
        dur_code                        TEXT,
        freq                            TEXT,
        value                           TEXT,
        unit                            TEXT,
        qc                              TEXT,
        meth_code                       TEXT,
        mpc                             TEXT,
        mpc_value                       TEXT,
        uncert                          TEXT,
        qualif                          TEXT,
        PRIMARY KEY (monit_id,timestamp)
);
SQL

#Create table for AQS site metadata
$RUN_ON_MYDB <<SQL
CREATE  TABLE                           aqs.site     (
        id                              SERIAL PRIMARY KEY,
        state_code                      TEXT NOT NULL,
        county_code                     TEXT NOT NULL,
        site_num                        TEXT NOT NULL,
        lat                             TEXT,
        lon                             TEXT,
        datum                           TEXT,
        elev                            TEXT,
        land_use                        TEXT,
        loc_set                         TEXT,
        site_bdate                      TEXT,
        site_edate                      TEXT,
        met_state_code                  TEXT,
        met_county_code                 TEXT,
        met_site_num                    TEXT,
        met_site_type                   TEXT,
        met_site_dist                   TEXT,
        met_site_dir                    TEXT,
        gmt_offset                      TEXT,
        agency                          TEXT,
        local_name                      TEXT,
        address                         TEXT,
        zip_code                        TEXT,
        state_name                      TEXT,
        county_name                     TEXT,
        city_name                       TEXT,
        cbsa_name                       TEXT,
        tribe_name                      TEXT,
        extract_date                    TEXT
);
SQL

#Add spatial component
$RUN_ON_MYDB <<SQL
ALTER TABLE aqs.site ADD COLUMN geom geometry(POINT,4326);
CREATE INDEX idx_geom ON aqs.site USING GIST(geom);
SQL

#Create table for AQS monitor metadata
```

```
$RUN_ON_MYDB <<SQL
CREATE  TABLE                           aqs.monitor (
        id                              SERIAL PRIMARY KEY,
        site_id                         INTEGER NOT NULL,
        param_code                      TEXT NOT NULL,
        param_name                      TEXT,
        poc                             TEXT NOT NULL,
        dat_byear                       TEXT,
        dat_edate                       TEXT,
        type                            TEXT,
        networks                        TEXT,
        report_ag                       TEXT,
        pqao                            TEXT,
        collect_ag                      TEXT,
        excl                            TEXT,
        monit_obj                       TEXT,
        last_meth_code                  TEXT,
        last_meth_name                  TEXT,
        naaqs_prim_monitor              TEXT,
        qa_prim_monitor                 TEXT,
        extract_date                    TEXT
);
SQL

#ADD FOREIGN KEYS
$RUN_ON_MYDB <<SQL
ALTER TABLE aqs.monitor
ADD FOREIGN KEY (site_id)               REFERENCES aqs.site(id);

ALTER TABLE aqs.data
ADD FOREIGN KEY (monit_id)              REFERENCES aqs.monitor(id);

ALTER TABLE aqs.data
ADD FOREIGN KEY (site_id)               REFERENCES aqs.site(id);
SQL


#ADD CONSTRAINTS FOR UNICITY
$RUN_ON_MYDB <<SQL
ALTER TABLE aqs.site
ADD CONSTRAINT site_un
UNIQUE (state_code, county_code, site_num);

ALTER TABLE aqs.monitor
ADD CONSTRAINT monit_un
UNIQUE (id, site_id, param_code, poc);

ALTER TABLE aqs.data
ADD CONSTRAINT data_un
UNIQUE (monit_id, site_id, timestamp);
SQL
```

Listing 3: create_aot.sh

```
#!/bin/bash

#Create the database infrastructure for the Array of Things (AoT) data (creation of
    tables, constraints definition)

# CONNECTION TO THE DATABASE
export PGPASSWORD= ********** #Environment variables
RUN_ON_MYDB="psql -h la2.rcc.uchicago.edu -p 5432 -U anais -d airchicago"

$RUN_ON_MYDB <<SQL
CREATE SCHEMA if not exists aot
SQL

#Create table for AoT sensor metadata
$RUN_ON_MYDB <<SQL
CREATE TABLE     aot.sensor (
        id         INTEGER NOT NULL,
        ontology TEXT,
        syst       TEXT     NOT NULL,
        name       TEXT     NOT NULL,
        param      TEXT     NOT NULL,
        unit       TEXT,
        minval     TEXT,
        maxval     TEXT,
        ref        TEXT,
        PRIMARY   KEY(id)
)
SQL


#Create table for AoT node metadata
$RUN_ON_MYDB <<SQL
CREATE TABLE        aot.node    (
        id            INTEGER  NOT NULL,
        raw_id        TEXT      NOT NULL,
        project       TEXT,
        vsn           TEXT,
        address       TEXT,
        lat           TEXT,
        lon           TEXT,
        description TEXT,
        PRIMARY KEY(id)
)
SQL

#Add spatial component
$RUN_ON_MYDB <<SQL
ALTER TABLE aot.node ADD COLUMN geom geometry(POINT,4326);
CREATE INDEX idx_geom ON aot.node USING GIST(geom);
SQL


#Create table for AoT data (complete dataset)
$RUN_ON_MYDB <<SQL
CREATE TABLE      aot.data    (
        sensor_id INTEGER   NOT NULL,
        timestamp TIMESTAMP NOT NULL,
        node_id    INTEGER   NOT NULL,
        val_raw    TEXT,
        value    TEXT,
        PRIMARY    KEY         (sensor_id, timestamp, node_id)
```

```
)
SQL


#ADD FOREIGN KEYS
$RUN_ON_MYDB <<SQL
ALTER TABLE aot.data
ADD FOREIGN KEY (sensor_id)            REFERENCES aot.sensor(id);

ALTER TABLE aot.data
ADD FOREIGN KEY (node_id)            REFERENCES aot.node(id);

ALTER TABLE aot.data
ADD FOREIGN KEY (sensor_id)            REFERENCES aot.sensor(id);

ALTER TABLE aot.data
ADD FOREIGN KEY (node_id)            REFERENCES aot.node(id);
SQL


#ADD CONSTRAINTS FOR UNICITY
$RUN_ON_MYDB <<SQL
ALTER TABLE aot.node
ADD CONSTRAINT node_un
UNIQUE (raw_id);

ALTER TABLE aot.sensor
ADD CONSTRAINT sensor_un
UNIQUE (syst,name,param);

ALTER TABLE aot.data
ADD CONSTRAINT data_un
UNIQUE (sensor_id, timestamp, node_id);
SQL


#CREATE SEQUENCES FOR NODE AND SENSOR TABLES
$RUN_ON_MYDB <<SQL
DROP SEQUENCE IF EXISTS seq_node;
DROP SEQUENCE IF EXISTS seq_sensor;

CREATE SEQUENCE seq_node;
CREATE SEQUENCE seq_sensor;
SQL
```

Listing 4: WF_aqs.sh

```
#!/bin/bash

#Workflow for AQS data

exec 1> log/WFaqs_$(date +'%Y%m%d').log 2>&1

#Extract and update AQS metadata
Rscript --vanilla aqs_meta.R

#Extract and insert of AQS data
Rscript --vanilla aqs_data.R
```

Listing 5: aqs_meta.R

```
#Extraction of AQS metadata (sites + monitors) and update of the database

#LIBRARIES
library(tidyverse)
library(DBI)
library(dbplyr)

#CONNECT TO THE DATABASE
con <- dbConnect(drv=RPostgreSQL::PostgreSQL(),host = "la2.rcc.uchicago.edu", port =
    5432,user= "anais",password = "**********",dbname="airchicago",":memory:")

#DATA EXTRACTION
#Read site metadata (zip file from https://aqs.epa.gov/aqsweb/airdata/download_files.html
    )
temp <- tempfile() #create a temporary file for the downloaded file
download.file("https://aqs.epa.gov/aqsweb/airdata/aqs_sites.zip",temp)
sites <- read_csv(unz(temp, "aqs_sites.csv"), col_types = cols(.default = "c")) #unzip
    and read the csv file
unlink(temp) #delete the temporary file
names(sites) <- c("state_code","county_code","site_num","lat","lon","datum","elev","land_
    use","loc_set","site_bdate","site_edate","met_state_code","met_county_code","met_site_
    num","met_site_type","met_site_dist","met_site_dir","gmt_offset","agency","local_name"
    ,"address","zip_code","state_name","county_name","city_name","cbsa_name","tribe_name",
    "extract_date")

#Convert site type
sites <- sites %>% mutate(lat=as.double(lat), lon=as.double(lon))

#Read monitors metadata (zip file from https://aqs.epa.gov/aqsweb/airdata/download_files.
    html)
temp <- tempfile() #create a temporary file in which we will save the downloaded file
download.file("https://aqs.epa.gov/aqsweb/airdata/aqs_monitors.zip",temp)
monitors <- read_csv(unz(temp, "aqs_monitors.csv"), col_types = cols(.default = "c")) #
    unzip and read the csv file
unlink(temp) #delete the temporary file
names(monitors) <- c("state_code","county_code","site_num","param_code","param_name","poc
    ","lat","lon","datum","dat_byear","dat_edate","type","networks","report_ag","pqao","
    collect_ag","excl","monit_obj","last_meth_code","last_meth_name","naaqs_prim_monitor",
    "qa_prim_monitor","local_site_name","address","state_name","county_name","city_name","
    cbsa_name","tribe_name","extract_date")

# #UNCOMMENT THE FOLLOWING PART FOR THE INITIAL INSERTION
# dbWriteTable(con, c("aqs","site"),sites,row.names=F,append=T,temporary=F)
# #Select _id that corresponds to monitor and add a new column site_id
# sites_db <- tbl(con, dbplyr::in_schema("aqs", "site")) %>% distinct(id,state_code,
    county_code, site_num) %>% collect()
# monitors <- inner_join(sites_db, monitors, by=c("state_code","county_code","site_num"))
```

```
    %>% rename(site_id=id)
# #Remove columns that are already in the site table
# monitors <- monitors %>% select(-c(state_code,county_code,site_num,lat,lon,datum,local_site_name,address,state_name,county_name,city_name,cbsa_name,tribe_name))
# dbWriteTable(con, c("aqs","monitor"),monitors,row.names=F,append=T,temporary=F)

#DATABASE UPDATE (add only rows that are not already in the database)
#aqs.site table update
sites_db <- tbl(con, dbplyr::in_schema("aqs", "site")) %>% distinct(id,state_code, county_code, site_num) %>% collect()
new_sites <- anti_join(sites, sites_db, by=c("state_code","county_code","site_num"))
dbWriteTable(con, c("aqs","site"),new_sites,row.names=F,append=T,temporary=F)

#aqs.monitor table update
sites_db <- tbl(con, dbplyr::in_schema("aqs", "site")) %>% distinct(id,state_code, county_code, site_num) %>% collect()
monitors_db <- tbl(con, dbplyr::in_schema("aqs", "monitor")) %>% distinct(id,site_id, param_code, poc) %>% collect()
monitors <- inner_join(sites_db, monitors, by=c("state_code","county_code","site_num")) %>% rename(site_id=id)
new_monitors <- anti_join(monitors, monitors_db, by=c("site_id","param_code","poc"))
new_monitors <- new_monitors %>% select(-c(state_code,county_code,site_num,lat,lon,datum, local_site_name,address,state_name,county_name,city_name,cbsa_name,tribe_name))
dbWriteTable(con, c("aqs","monitor"),new_monitors,row.names=F,append=T,temporary=F)

#Update geometry column for the site table
dbExecute(con, 'UPDATE aqs.site SET geom=ST_SetSRID(ST_MakePoint("lon"::numeric,"lat"::numeric),4326) WHERE geom IS NULL;')

#Remove all the variables and clear up R memory
rm(list=ls())
gc()
```

Listing 6: aqs_data.R

```
#Extraction of AQS criteria pollutants data for Chicago's surrounding counties and insertion in the database

#LIBRARIES
library(tidyverse)
library(DBI)
library(dbplyr)

#CONNECT TO THE DATABASE
con <- dbConnect(drv=RPostgreSQL::PostgreSQL(),host = "la2.rcc.uchicago.edu", port = 5432,user= "anais",password = "**********",dbname="airchicago",":memory:")

#Retrieve the last sampling date in the aqs.data table
bdate <- con %>% tbl(sql("SELECT MAX(timestamp) FROM aqs.data")) %>% pull() %>% format("%Y%m%d") #Pull out from the database in a single variable
#Today's date
edate <- format(Sys.Date(), "%Y%m%d")

# #FOR INITIAL INSERTION USE THE FOLLOWING BEGINNING DATE
# bdate <- "20180101"

#Function extracting data for a specific time period and a specific county.
# Request example : https://aqs.epa.gov/api/rawData?user=anais.ladoy@epfl.ch&pw=**********&format=AQCSV&pc=CRITERIA&param=&bdate=20150101&edate=20180101&state=17&county=031&dur=1
aqs_dat <- function(bdate,edate,state,county) {
  return (read_csv(paste0("https://aqs.epa.gov/api/rawData?user=anais.ladoy@epfl.ch&pw=**********&format=AQCSV&pc=CRITERIA&param=&bdate=",bdate,"&edate=",edate,"&state=",state
```

```
    ,"&county=",county,"&dur=1")))
}

#Run aqs_dat() for Chicago's surrounding counties
data <- data.frame()

#IL counties
counties.IL <- c('031','097','043','197','091','111','089','093','063')
for (i in 1:length(counties.IL)) {
  r.IL <- aqs_dat(bdate,edate,"17",counties.IL[i])
  data <- rbind(data,r.IL)
}

#IN counties
counties.IN <- c('089','127','111','073','091','149','131')
for (i in 1:length(counties.IN)) {
  r.IN <- aqs_dat(bdate,edate,"18",counties.IN[i])
  data <- rbind(data,r.IN)
}

#WI counties
counties.WI <- c('101','127','059','079','133')
for (i in 1:length(counties.WI)) {
  r.WI <- aqs_dat(bdate,edate,"55",counties.WI[i])
  data <- rbind(data,r.WI)
}

#Remove empty lines (NA lines)
data <- data %>% filter(!is.na(site) & site!='END OF FILE')

#Rename the columns as defined in the DB (needed for dbWriteTable to work)
names(data) <- c("aqs_id","dat_status","act_code","timestamp","param_code","dur_code","freq","value","unit","qc","poc","lat","lon","datum","elev","meth_code","mpc","mpc_value","uncert","qualif")

#Decompose aqs_id in state_code, county_code, site_num to allow the join operation with the site table
data <- data %>% mutate(state_code=substr(aqs_id,4,5), county_code=substr(aqs_id,6,8), site_num=substr(aqs_id,9,12))

#Add monit_id and site_id
sites_db <- tbl(con, dbplyr::in_schema("aqs", "site")) %>% distinct(id, state_code, county_code, site_num) %>% collect()
data <- inner_join(sites_db, data, by=c("state_code","county_code","site_num")) %>% rename(site_id=id)

monitors_db <- tbl(con, dbplyr::in_schema("aqs", "monitor")) %>% distinct(id, site_id, param_code, poc) %>% collect()
data <- inner_join(monitors_db, data, by=c("site_id","param_code","poc")) %>% rename(monit_id=id)

#Remove state_code, county_code, site_num, param_code, poc as they are already in the monitor and site tables
data <- data %>% select(-c(state_code,county_code,site_num,param_code,poc,lat,lon,datum,elev))

#Rearrange columns to match the DB structure
data <- data[, c("monit_id","timestamp","site_id","aqs_id","dat_status","act_code","dur_code","freq","value","unit","qc","meth_code","mpc","mpc_value","uncert","qualif")]

#INSERT NEW DATA IN THE DB
dbWriteTable(con,c("aqs","data"),data,row.names=F,append=T,temporary=F)
```

```

#Remove all the variables and clear up R memory
rm(list=ls())
gc()
```

Listing 7: WF_aot.sh

```
#!/bin/bash

#Workflow for AoT data

exec 1> log/WFaot_$(date +'%Y%m%d').log 2>&1

#Extract AoT sensors' yesterday data
sh aot_extract.sh

#Insert AoT's data in the database and update metadata
sh aot_insert.sh
```

Listing 8: aot_extract.sh

```
#!/bin/bash

#Extraction of AoT data and metadata

#Download the archive
wget http://www.mcs.anl.gov/research/projects/waggle/downloads/datasets/AoT_Chicago.complete.latest.tar
#Untar the archive
tar xvf AoT_Chicago.complete.latest.tar
#Extraction of yesterday's data, save the output in a CSV file
gunzip -c AoT_Chicago.complete.$(date +'%Y-%m-%d')/data.csv.gz | (head -1 && grep 'date --date='yesterday' +'%Y/%m/%d'') > AoT_Chicago.complete.$(date +'%Y-%m-%d')/yesterday.csv

#You can have issues running this command on Terminal (OSX) as Linux use GNU coreutils version of date and OSX use BSD legacy utilities.
#To fix this problem, you need to install the coreutils package (brew install coreutils) that provides GNU version of tools and replace date by gdate as follows :
# gunzip -c AoT_Chicago.complete.$(date +'%Y-%m-%d')/data.csv.gz | grep $(gdate --date='yesterday' +'%Y/%m/%d') > final_o5.csv

#Remove the folder
rm -r AoT_Chicago.complete.latest.tar
```

Listing 9: aot_insert.sh

```
#!/bin/bash

#Insertion of AoT's data and metadata update in the database

# CONNECT TO THE DATABASE
export PGPASSWORD= ********** #Environment variables
RUN_ON_MYDB="psql -h la2.rcc.uchicago.edu -p 5432 -U anais -d airchicago"

cp -R /home/aladoy/scripts/AoT_Chicago.complete.$(date -d "yesterday" +'%Y-%m-%d')/ /var/tmp/aot_data

#UPDATE NODE METADATA IN THE DATABASE
$RUN_ON_MYDB <<SQL
CREATE TEMP TABLE tmp(
  raw_id TEXT,
```

```
  project     TEXT,
  vsn         TEXT,
  address     TEXT,
  lat         TEXT,
  lon         TEXT,
  description TEXT,
  PRIMARY KEY (raw_id)
  );

COPY tmp FROM '/var/tmp/aot_data/AoT_Chicago.complete.$(date -d "yesterday" +'%Y-%m-%d')/nodes.csv' DELIMITER ',' CSV HEADER;
ANALYZE tmp;

INSERT INTO aot.node(id, raw_id, project, vsn, address, lat, lon, description)
SELECT nextval('seq_node'), tmp.raw_id, tmp.project, tmp.vsn, tmp.address, tmp.lat, tmp.lon, tmp.description
FROM tmp LEFT JOIN aot.node ON tmp.raw_id=aot.node.raw_id
WHERE aot.node.raw_id IS NULL;

DROP TABLE tmp;
SQL

#Update geometry column
$RUN_ON_MYDB <<SQL
UPDATE aot.node SET geom = ST_SetSRID(ST_MakePoint("lon"::numeric,"lat"::numeric),4326) WHERE geom IS NULL;
SQL


#UPDATE SENSOR METADATA IN THE DATABASE
$RUN_ON_MYDB <<SQL
CREATE TEMP     TABLE  tmp(
        ontology TEXT,
        syst     TEXT,
        name     TEXT,
        param    TEXT,
        unit     TEXT,
        minval   TEXT,
        maxval   TEXT,
        ref      TEXT,
        PRIMARY KEY (syst,name,param)
        );

COPY tmp FROM '/var/tmp/aot_data/AoT_Chicago.complete.$(date -d "yesterday" +'%Y-%m-%d')/sensors.csv' DELIMITER ',' CSV HEADER;
ANALYZE tmp;

INSERT INTO aot.sensor(id, ontology, syst, name, param, unit, minval, maxval, ref)
SELECT nextval('seq_sensor'), tmp.ontology, tmp.syst, tmp.name, tmp.param, tmp.unit, tmp.minval, tmp.maxval, tmp.ref
FROM tmp LEFT JOIN aot.sensor ON tmp.syst=aot.sensor.syst AND tmp.name=aot.sensor.name AND tmp.param=aot.sensor.param
WHERE aot.sensor.syst IS NULL AND aot.sensor.name IS NULL AND aot.sensor.param IS NULL;

DROP TABLE tmp;
SQL


#INSERT NEW DATA IN THE DATABASE
$RUN_ON_MYDB <<SQL
CREATE TEMP       TABLE      tmp(
        timestamp  TIMESTAMP,
```

```
        raw_id      TEXT,
        syst        TEXT,
        sensor      TEXT,
        param       TEXT,
        val_raw     TEXT,
        value       TEXT,
        PRIMARY KEY (timestamp, raw_id, syst, sensor, param)
        );

COPY tmp FROM '/var/tmp/aot_data/AoT_Chicago.complete.$(date -d "yesterday" +'%Y-%m-%d')/
    yesterday.csv' DELIMITER ',' CSV HEADER;
ANALYZE tmp;

INSERT INTO aot.data(sensor_id, timestamp, node_id, val_raw, value)
SELECT S.id, tmp.timestamp, N.id, tmp.val_raw, tmp.value
FROM tmp LEFT JOIN aot.sensor S ON tmp.syst=S.syst AND tmp.sensor=S.name AND tmp.param=S.
    param
LEFT JOIN aot.node N ON tmp.raw_id=N.raw_id
WHERE S.id IS NOT NULL;

DROP TABLE tmp;
SQL

#Remove the files
rm -r AoT_Chicago.complete.latest.tar
rm -r AoT_Chicago.complete.$(date +'%Y-%m-%d')
rm -f yesterday.csv
```

Document 7.1: Description of the models that will be implemented in the Air Quality Project

Land Use Regression (LUR) Model - Sensors Only

1. Calculate range, IQR for the predictor variables within buffers of sensor locations
   (a) Buffers (m) for analysis: 50, 100, 150, 200, 300, 500, 1000
   (b) Predictor Variables (list in 13): Point emissions, Area emissions (ie. traffic intensity and road length), Land Use, Population Density
2. Descriptive statistical summary of analysis comparing AoT, AirCasting, and EPA regulatory sensors
3. Replication of methods:
   (a) Univariate stepwise regression following LUR standard literature for baseline models, using optimal buffer distance for the region of study
   (b) Interpolation for region of study
   (c) Spatial autocorrelation testing of model results, following LUR literature
4. Potential extensions:
   (a) Factor analysis to account for correlation across predictor variables
   (b) Updated interpolation and/or spatial regression methods for final analysis

Hybrid Model - Sensors with Satellite Data

1. Aerosol Optical Depth (AOD) satellite data pre-processing and storage
2. Develop variable estimates for each 1km2 grid square in region:
   (a) Baseline Covariates: AOD, Planetary Boundary Layer, Meteorological data, NDVI
   (b) Spatiotemporal Predictor Variables: Point emissions, Area emissions (ie. traffic intensity and road length), Land Cover, Elevation
3. Replication of methods - Three-stage model as follows:
   (a) PM model fit for all AOD grid squares co-located with sensors
   (b) Model fit used to determine value for all AOD grid squares without sensors
   (c) Predicted PM model (generalized additive mixed model with spatial smoothing) fit for grid squares without AOD data

Figure 16: Results of the AirBeam sensor performance evaluation performed during a biking session in the south side of Chicago on April 26th

Performance evaluation of the AirBeam sensor

A. Ladoy

4/26/2018

Sensor carrier : Bike
Date : 04/26/2018
Time : 18:19 - 19:08
Location : South Side of Chicago
Registered account on the AirCasting platform : csds_aot

```
#List of AirCasting sessions for the csds_aot account

library('httr')
user_session.r<-GET('http://aircasting.org/api/sessions.json?q[usernames]=csds_aot')
user_session <- content(user_session.r)

flag <- integer()
user_session.meta<-data.frame()
for (i in 1:3) {
tryCatch(
  {
  print(paste('Session title :',user_session[[i]]$title,'Session id :', +
                user_session[[i]]$id,sep=' '))
  user_session.meta[i,'title'] <- user_session[[i]]$title
  user_session.meta[i,'id'] <- user_session[[i]]$id
  },
  error=function(err){
  message('On iteration ',i, ' there was an error: ',err)
  flag <<-c(flag,i)}
)
}
```

```
#Retrieve data for the session "irene"
session_id <- user_session.meta$id[1]

session.r <- GET(paste('http://aircasting.org/api/sessions/',session_id,'.json',sep=''))
session <- content(session.r)

library(data.table)

data.temp <- rbindlist(session$streams$`AirBeam2-F`$measurements, fill=TRUE)
data.rh <- rbindlist(session$streams$`AirBeam2-RH`$measurements, fill=TRUE)
data.PM25 <- rbindlist(session$streams$`AirBeam2-PM2.5`$measurements, fill=TRUE)

#Conversion of temperature measurements to Celsius degree
data.temp$value <- sapply(data.temp$value, function(x) (x-32)/1.8)
```

```
library(ggplot2)
library(gridExtra)

#Function to print results
```

```
descriptive_stat <- function(pollutant, unit, name, style) {
  if(missing(style))
  {
    style='equal'
  }

  #Conversion of time to POSIXct format
  pollutant$time <- as.POSIXct(pollutant$time,tz = "UTC",format = "%Y-%m-%dT%H:%M:%SZ")
  print(summary(pollutant$value)) # Descriptive statistics
  p1<-ggplot(pollutant, aes(value)) + geom_histogram()+labs(x=paste(name, '[',unit,']'))
  p2<-ggplot(pollutant,aes(time,value)) + geom_line(size=1)
  print(grid.arrange(p1, p2, ncol = 2))
}
```

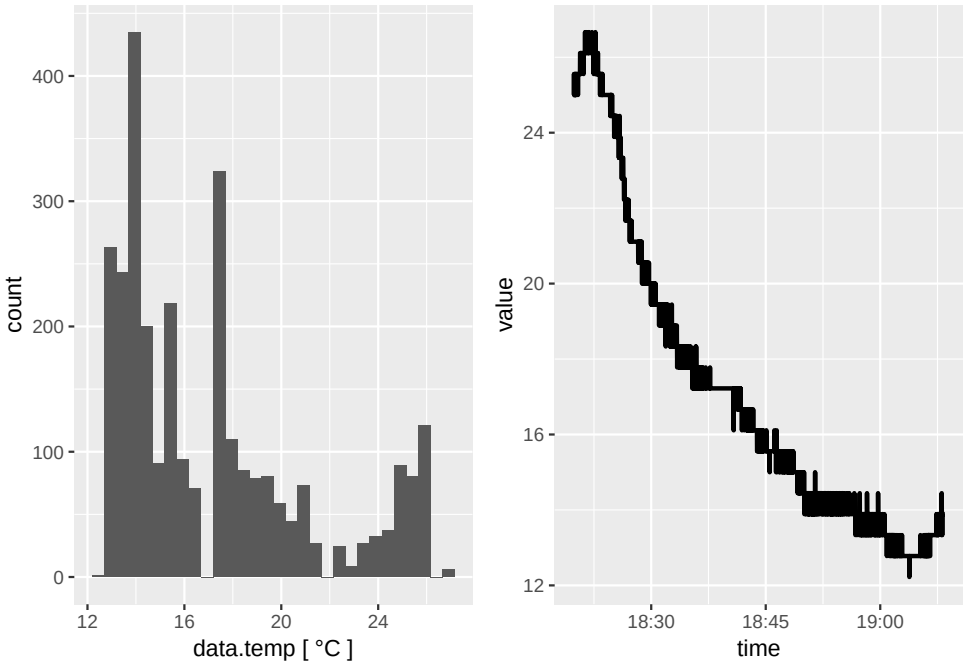
Temperature measurements

Follow the link below to see the map of recorded temperature measurements in the AirCasting platform : http://www.aircasting.org/map#/map_sessions?data=%7B%22location%22:%7B%22address%22:%22%22,%22distance%22:%2210%22,%22limit%22:false%7D,%22gridResolution%22:25,%22tags%22:%22%22,%22usernames%22:%22csds_aot,%20%22,%22time%22:%7B%22timeFrom%22:300,%22timeTo%22:1739,%22dayFrom%22:0,%22dayTo%22:365,%22yearFrom%22:2017,%22yearTo%22:2018%7D,%22heat%22:%7B%22highest%22:135,%22high%22:100,%22mid%22:75,%22low%22:45,%22lowest%22:15%7D,%22sensorId%22:%22%22,%22counter%22:1%7D&sessionsIds=%5B53596%5D&tmp=%7B%22tmpSensorId%22:%22Temperature-AirBeam2-F%20(F)%22%7D&didSessionsSearch=true&map=%7B%22zoom%22:14,%22lat%22:41.784973584470016,%22lng%22:-87.58172333350001,%22mapType%22:%22terrain%22%7D

```
descriptive_stat(data.temp,'°C',substitute(data.temp))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   12.22   13.89   16.11   17.10   18.89   26.67
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



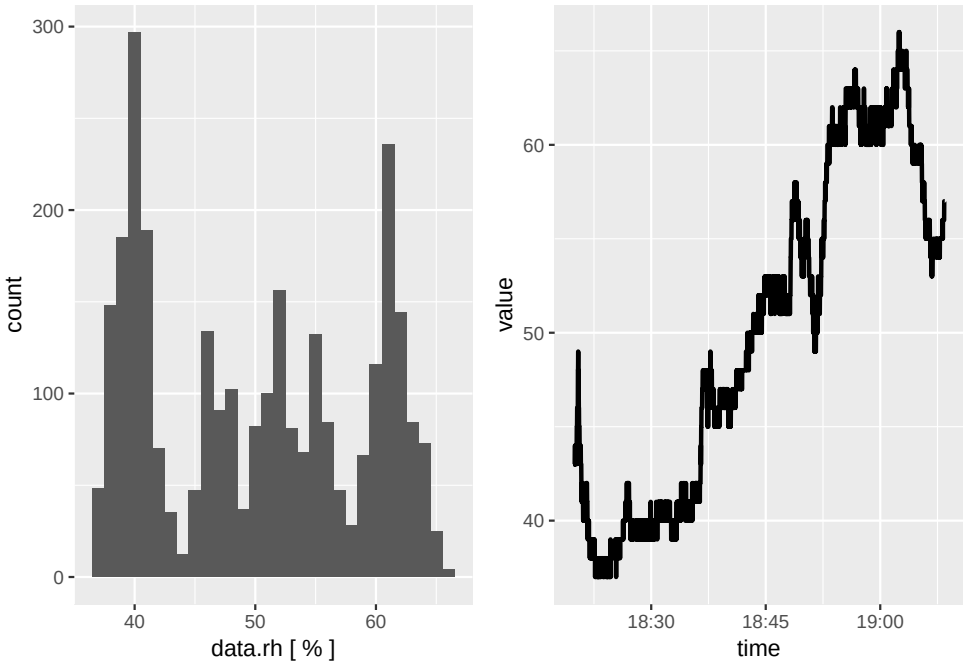

```
## TableGrob (1 x 2) "arrange": 2 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
```

Relative Humidity (RH) measurements

Follow the link below to see the map of recorded RH measurements in the AirCasting platform : http://www.aircasting.org/map#/map_sessions?data=%7B%22location%22:%7B%22address%22:%22%22,%22distance%22:%2210%22,%22limit%22:false%7D,%22gridResolution%22:25,%22tags%22:%22%22,%22usernames%22:%22csds_aot,%20%22,%22time%22:%7B%22timeFrom%22:300,%22timeTo%22:1739,%22dayFrom%22:0,%22dayTo%22:365,%22yearFrom%22:2017,%22yearTo%22:2018%7D,%22heat%22:%7B%22highest%22:100,%22high%22:75,%22mid%22:50,%22low%22:25,%22lowest%22:0%7D,%22sensorId%22:%22%22,%22counter%22:1%7D&sessionsIds=%5B53596%5D&tmp=%7B%22tmpSensorId%22:%22Humidity-AirBeam2-RH%20(%25)%22%7D&didSessionsSearch=true&map=%7B%22zoom%22:14,%22lat%22:41.784973584470016,%22lng%22:-87.58172333350001,%22mapType%22:%22terrain%22%7D

```
descriptive_stat(data.rh, '%',substitute(data.rh))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      37      41      50      50      59      66
```



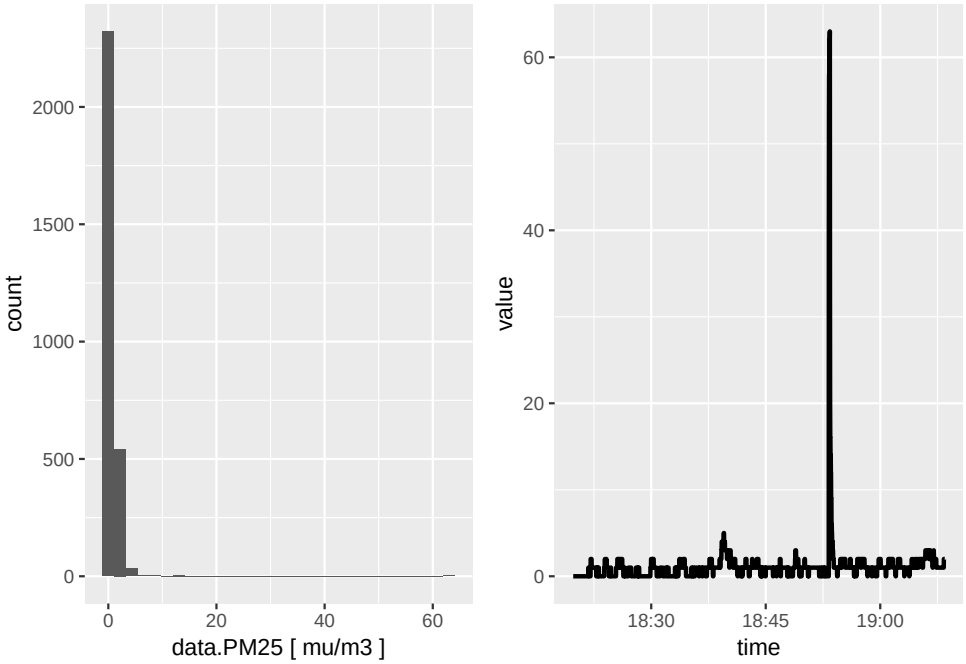

```
## TableGrob (1 x 2) "arrange": 2 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
```

PM2.5 measurements

Follow the link below to see the map of recorded PM2.5 measurements in the AirCasting platform : http://www.aircasting.org/map#/map_sessions?data=%7B%22location%22:%7B%22address%22:%22%22,%22distance%22:%2210%22,%22limit%22:false%7D,%22gridResolution%22:25,%22tags%22:%22%22,%22usernames%22:%22csds_aot,%20%22,%22time%22:%7B%22timeFrom%22:300,%22timeTo%22:1739,%22dayFrom%22:0,%22dayTo%22:365,%22yearFrom%22:2017,%22yearTo%22:2018%7D,%22heat%22:%7B%22highest%22:150,%22high%22:55,%22mid%22:35,%22low%22:12,%22lowest%22:0%7D,%22sensorId%22:%22%22,%22counter%22:1%7D&sessionsIds=%5B53596%5D&tmp=%7B%22tmpSensorId%22:%22Particulate%20Matter-AirBeam2-PM2.5%20(%C2%B5g%2Fm%C2%B3)%22%7D&didSessionsSearch=true&map=%7B%22zoom%22:14,%22lat%22:41.784973584470016,%22lng%22:-87.58172333350001,%22mapType%22:%22terrain%22%7D

```
descriptive_stat(data.PM25, 'mu/m3',substitute(data.PM25), 'jenks')
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   1.000   1.216   1.000  63.000
```

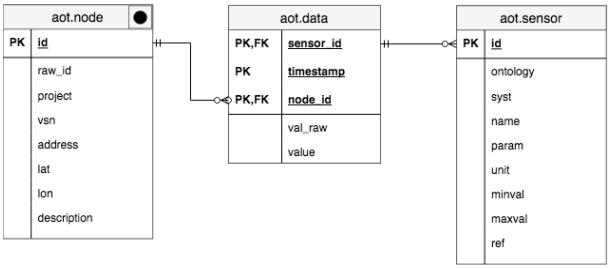



```
## TableGrob (1 x 2) "arrange": 2 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
```

Figure 17: Data documentation relative to the spatial database

DB - Readme

Array of Things (AoT)



aot.data table

- `sensor_id` - ID of the sensor that did the measurement.
- `timestamp` - Time at which the measurement was recorded.
- `node_id` - ID of node where the measurement was recorded.
- `val_raw` - Raw measurement value from sensor.
- `val_hrf` - Converted, "human readable" value from sensor.

aot.node table

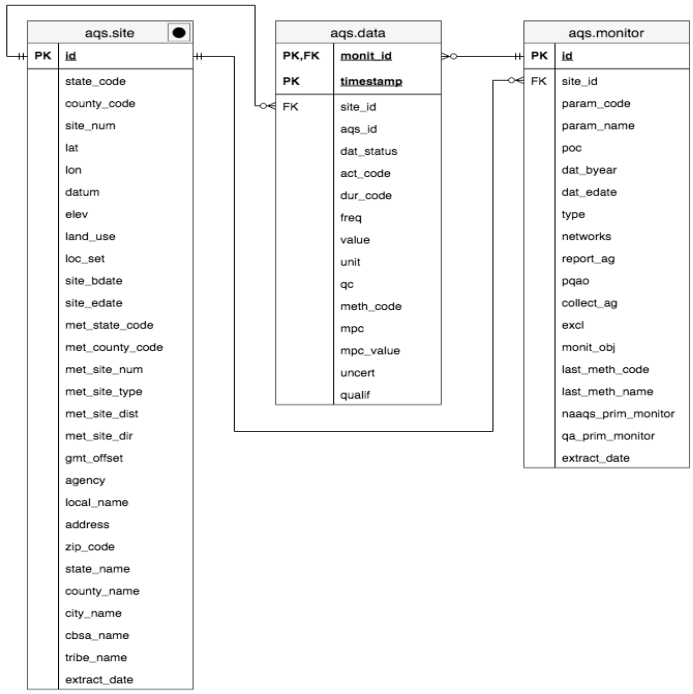
- `id` - ID of node.
- `raw_id` - ID of the node as defined by the Argonne team.
- `project` - Project which manages node (only Chicago for now)
- `vsn` - Public name for node. The VSN is visible on the physical enclosure.
- `address` - Street address of node.
- `lat` - Latitude of the node.
- `lon` - Longitude of the node.
- `description` - More detailed description of node's build and configuration.

aot.sensor table

- `id` - ID of the sensor
- `ontology` - Ontology of measurement.
- `syst` - Subsystem containing the sensor.
- `name` - Sensor name.
- `param` - Parameter measured by the sensor.

- `maxval` - Maximum value that the sensor can record.
- `ref` - Reference to the sensor's datasheet.

Air Quality System (AQS)



aqs.data table

- `monitor_id` - ID of the monitor that did the measurement.
- `timestamp` - Time at which the measurement was recorded.
- `site_id` - ID of the site where the measurement was recorded.
- `dat_status` - Status of the data. 0 = Preliminary 1 = Final. Data are denoted as final after the agency collecting and reporting the data certifies that they meet quality assurance requirements and are complete and correct in AQS. This is only required for criteria pollutants reported from state agencies.
- `act_code` - Action code for data ingesting. Not relevant to data obtained from QAD. Always blank.
- `dur_code` - Measurement (sampling) period in minutes.
- `freq` - How often the measurement is repeated (minutes). If measurements are taken multiple times per day (i.e., hourly), it is blank; otherwise, minutes equivalent (e.g., every day = 1440, and every other day = 2880).
- `value` - Data (sampled) value of the specified parameter.

- `qc` - The AIRNow code used to link to the quality control codes that describe the validity, invalidity, or questionable status of the measurement.
- `meth_code` - Three-digit AQS code that identifies the method used to perform the measurement.
- `mpc` - Measurement Performance Characteristic (MPC) is a performance measurement for the measurement taken. The only valid value for QAD responses is "MDL" meaning method (lower) detection limit.
- `mpc_value` - The value for the mpc (MDL) in the same units as the sample.
- `uncert` - Uncertainty needs to be in the same units as the specified parameter and is given using the 95% confidence level.
- `qualif` - AQS qualifier code(s) separated by spaces. Qualifiers indicate whether the data have been flagged by the submitter and the reason the sample was so flagged.

aqs.site table

- `id` - ID of the site
- `state_code` - The FIPS code of the state in which the site is located.
- `county_code` - The FIPS code of the county in which the site is located.
- `site_num` - A unique number within the county identifying the site.
- `lat` - Latitude of the monitoring site.
- `lon` - Longitude of the monitoring site.
- `datum` - The Datum associated with the Latitude and Longitude measures.
- `elev` - The elevation of the ground at the site in meters above mean sea level.
- `land_use` - A category describing the predominant land use within a 1/4 mile radius of the site.
- `loc_set` - A description of the setting within which the monitoring site is located. E.g., rural, urban, etc.
- `site_bdate` - The date when the site began operating.
- `site_edate` - The date on which the operating agency indicated that all operations ceased at this site.
- `met_state_code` - Where sites are required to collect meteorological data, they may be able to list a surrogate site from where the meteorological data will be used. If a "met" site is listed this contains the AQS State Code identifier for that site.
- `met_county_code` - If a "met" site is listed this contains the AQS County Code identifier for that site.
- `met_site_id` - If a "met" site is listed this contains the AQS Site Number for that site.
- `met_site_type` - If a "met" site is listed this contains the type of surrogate site. E.g., AQS site, National Weather Service site, etc.
- `met_site_dist` - If a "met" site is listed this contains the distance from this site to the met site in meters.
- `met_site_dir` - If a "met" site is listed this contains the direction from this site to the met site (true, not magnetic, direction).

- `local_name` - The name of the site (if any) given by the State, local, or tribal air pollution control agency that operates it.
- `address` - The approximate street address of the monitoring site.
- `zip_code` - The postal zip code in which the monitoring site resides.
- `state_name` - The name of the state where the monitoring site is located.
- `county_name` - The name of the county where the monitoring site is located.
- `city_name` - The name of the city where the monitoring site is located. This represents the legal incorporated boundaries of cities and not urban areas.
- `cbsa_name` - The name of the core bases statistical area (metropolitan area) where the monitoring site is located.
- `tribe_name` - If this site resides on tribal lands and the tribe has chosen to identify the site with tribal identifiers, this is the name of the tribe that owns the site.
- `extract_date` - The date on which this data was retrieved from the AQS Data Mart. This does not mean that all data is valid as of this date. Once site information is entered by the owning agency, it may not be updated as values change (e.g., location setting evolves from urban to suburban).

aqs.monitor table

- `id` - ID of the monitor
- `site_id` - ID of the site where the measurement was taken
- `param_code` - The AQS code corresponding to the parameter measured by the monitor.
- `param_name` - The name or description assigned in AQS to the parameter measured by the monitor. Parameters may be pollutants or non-pollutants.
- `poc` - The "parameter occurrence code" (POC). The POC is used to specify if more than one monitor is measuring the same parameter at the same site. For example, if there are two ozone monitors at a site, they would have different POCs.
- `dat_byear` - The year in which the earliest sample from this site is available in AQS.
- `dat_edate` - The date on which the most recent sample from this site is available in AQS. This is often the best way to determine if a monitor is still operating. Note that the reporting deadlines to AQS are generally lengthy - about 6 months for most parameters.
- `type` - An administrative or regulatory classification for the monitor.
- `networks` - A list of the monitoring networks (groups of monitors with common goals and procedures) to which the monitor belongs. If the monitor belongs to more than one network, the names will be separated with semicolons.
- `report_ag` - The name of the agency responsible for reporting data to AQS.
- `pqao` - The name of the Primary Quality Assurance Organization for the monitor. Monitors of the same parameter belonging to the same PQAO must meet aggregate quality assurance requirements.
- `collect_ag` - The name of the agency responsible for collecting data from the monitor.
- `excl` - If the agency operating the monitor has requested that data from this monitor be excluded from NAAQS calculations and the governing EPA regional office has agreed, the NAAQS standard(s)

and the years of exclusion are listed.

- `monit_obj` - Identification of the reason for measuring air quality by the monitor.
- `last_meth_code` - A three digit code representing the measurement method used by the monitor for its most recent sample (methods can change, but often do not). A method code is only unique within a parameter (that is, method 111 for ozone is not the same as method 111 for benzene).
- `last_meth_name` - The full description of the measurement method used by the monitor for its most recent sample (methods can change, but often do not).
- `naaqs_prim_monitor` - A flag indicating if this monitor is part of a collocated set of monitors at the site and it is the primary data source for NAAQS data comparisons.
- `qa_prim_monitor` - A flag indicating if this monitor is part of a collocated set of monitors at the site and it is the primary monitor for making quality assurance comparisons.
- `extract_date` - The date on which this data was retrieved from the AQS Data Mart. This does not mean that all data is valid as of this date. Once monitor information is entered by the reporting agency, it may not be updated as values change (e.g., location setting evolves from urban to suburban).