12-2015

A Proposed Integrated Data Collection, Analysis and Sharing Platform for Impact Evaluation

Andreas Kipt
University of California Berkeley

Waylon Brunette
University of Washington

Jordon Kellerstrass
University of California Berkeley

Matthew Podolsky
University of California Berkeley

Javier Rosa
University of California Berkeley

See next page for additional authors

Citation Details

Kipf, A., et al., Aproposed integrated data collection, analysis and sharing platform for impact evaluation. Development Engineering (2015),

Development Engineering ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

Development Engineering

journal homepage: www.elsevier.com/locate/deveng

A proposed integrated data collection, analysis and sharing platform for impact evaluation

Andreas Kipf [a], Waylon Brunette [b], Jordan Kellerstrass [a], Matthew Podolsky [a], Javier Rosa [a], Mitchell Sundt [b], Daniel Wilson [a], Gaetano Borriello [b], Eric Brewer [a], Evan Thomas [c,*]

[a] University of California, Berkeley, 450 Sutardja Dai Hall, MC: 1764, Berkeley, CA 94720-1764, United States
[b] University of Washington, 101 Paul G. Allen Center, Department of CS&E, 185 Stevens Way, Seattle, WA 98195-2350, United States
[c] Portland State University, Department of Mechanical and Materials Engineering, 1930 SW 4th Ave., Portland, OR 97201, United States

ARTICLE INFO

Article history:
Received 24 March 2015
Received in revised form
25 November 2015
Accepted 8 December 2015

Keywords:
Cloud computing
ICT4D
Impact analysis
Provenance
Global development

ABSTRACT

Global poverty reduction efforts value monitoring and evaluation, but often struggle to translate lessons learned from one intervention into practical application in another intervention. Commonly, data is not easily or often shared between interventions and summary data collected as part of an impact evaluation is often not available until after the intervention is complete. Equally limiting, the workflows that lead to research results are rarely published in a reproducible, reusable, and easy-to-understand fashion for others. Information and communication technologies widely used in commercial and government programs are growing in relevance for international global development professionals and offer a potential towards better data and workflow sharing. However, the technical and custom nature of many data management systems limits their accessibility to non-ICT professionals. The authors propose an end-to-end data collection, management, and dissemination platform designed for use by global development program managers and researchers. The system leverages smartphones, cellular based sensors, and cloud storage and computing to lower the entry barrier to impact evaluation.

1. Introduction

Efforts to assess the impact of global poverty reduction projects, such as solar lighting installations, latrines, water pumps and filters, and cookstoves, often rely on data collected through person-to-person surveys, subjective observations, and/or expensive and time-consuming experimental studies. Data is frequently recorded by hand and processed on a per-project basis. These conventional approaches have limitations that can impact the value of the derived data. In the case of surveys and observations, research has shown surveys often overestimate adoption rates due to courtesy bias (where the participant is attempting to please the surveyor) (Manun'Ebo et al., 1997) or recall bias (tendency to forget details in more distant past) (Stanton et al., 1987). Furthermore, the presence or repeated visits of observers or enumerators can cause reactivity—influencing the behavior they are measuring (Zwane et al., 2011). And even with well-designed experimental studies such as randomized controlled trials, the data collected and subsequent impact analysis are often not available until well after the intervention is considered complete. This can delay providing input to subsequent interventions. Overarching these challenges is the bespoke nature of most data collection, analysis and sharing systems that are either (1) basic and limited or (2) expensive.

A user-configurable, online data management platform supported by robust electronic data collection tools may improve data quality and sharing and reuse as well as program accountability and performance in the global development sector. Further, such a platform could reduce the redundant implementation and management of many common components (e.g., database servers, transmission protocols, and privacy arrangements) and enable researchers to spend their time and money on the development project and impact analysis instead. In this paper, we propose a design for a platform like this called Mezuri (Esperanto for "measure").

Kepler (Altintas et al., 2004), Conveyor (Linke et al., 2011), Taverna (Hull et al., 2006), Mobyle (N'eron et al., 2009), DHIS2 (Manya et al., 2012) and Open Foris (Miceli et al., 2011) are

Abbreviations: ODK, Open data kit; M&E, Monitoring and evaluation; SUMs, Stove use monitors

* Corresponding author.
E-mail addresses: kipf@cs.berkeley.edu (A. Kipf), wrb@cse.uw.edu (W. Brunette), kellerstrass@cs.berkeley.edu (J. Kellerstrass), podolsky@cs.berkeley.edu (M. Podolsky), javirosa@cs.berkeley.edu (J. Rosa), msundt@cse.uw.edu (M. Sundt), dlwilson@berkeley.edu (D. Wilson), gaetano@cse.uw.edu (G. Borriello), brewer@cs.berkeley.edu (E. Brewer), evan.thomas@pdx.edu (E. Thomas).

http://dx.doi.org/10.1016/j.deveng.2015.12.002

examples of existing scientific workflow and data management platforms. These platforms have several limitations, such as their supported programming languages. For example, while Kepler allows users to integrate R and MATLAB code into workflows, code written in other languages can only be integrated in the form of web services, which might already be too difficult for most users. Additionally, some of these platforms are domain specific and thus only provide a limited set of algorithms. Mobyle, for instance, focuses on algorithms for the bioinformatics domain. DHIS2 is exclusively for health workflows and while it provides built-in analysis features and a web API, any customized processing code is run outside of the system. Further, these platforms only, if at all, track the provenance of data (i.e., how data was collected or processed) from the point where it enters to the point where it leaves the platform. As these platforms do not contain data gathering functionality they neither capture the provenance of the originally collected data nor the provenance of the final outputs which leave the system (e.g., visualizations).

In contrast to most of these platforms, Mezuri is conceived as a more broadly applicable, user-configurable data collection, analysis and sharing platform for global development professionals. Mezuri aims to provide end-to-end support for collecting provenance data and allows users to choose from a variety of programming languages and even combine different languages when implementing their workflows. Our proposed platform builds on existing efforts to collect data with smartphones and cellular-based sensors and digital surveys in global development settings, and combines these technologies with online data tools. Specifically, Mezuri will extend Open Data Kit (ODK) by building upon ODK 2.0's (Brunette et al., 2013) infrastructure to create an integrated data collection platform with provenance, processing, analysis, and sharing. Mezuri will use ODK 2.0 infrastructure and protocols to enable end-to-end integration with the ODK 2.0 tool suite. Additionally, Mezuri will leverage existing remote sensor data collection systems like Get All The Data (Pannuto et al., 2013) and SWEETSense (Thomas et al., 2013). These systems were selected because of the team's existing expertize with these platforms. In this paper, we identify the needs of researchers based on user surveys (see Section 2) and example applications (see Section 3), derive corresponding engineering requirements (see Section 4), and propose a design that addresses these requirements (see Section 5). Finally, we will outline future technology challenges (see Section 6).

2. Identification of needs

Mezuri aims to be a broadly accessible data collection and processing platform that helps global development experts build workflows that meet their field operational and research needs. To identify those needs, we conducted interviews with potential users. We interviewed 17 researchers engaged in global development impact analysis projects. Our interview consisted of 19 questions organized into the following categories: collection, processing, analysis, sharing, provenance, security and privacy. Each of those categories addressed different aspects of the potential users' workflows.

In most cases, interviewees reported that survey data collection is conducted by enumerators in the field using phones or tablets with tools like ODK Survey. Conducted surveys have up to several hundred questions and thousands of participants. In some cases, interviewees are collecting instrumentation data from energy, power, and temperature sensors. Some of the sensors store their data on local storage and wait for it to be manually collected by humans, whereas others have a cellular connection and are able to automatically send their data to the cloud. Some projects produce raw sensor data of many terabytes per month. Once collected, survey data is mostly cleaned using tools such as Open Refine. We found that the programming language R is the most common among our interviewed researchers, followed by Python and C/C++. Our interviewees stated that a common processing step is correlating survey and sensor data in time, especially when both data sets contain GPS coordinates.

Once the data has been processed and analyzed, researchers often share their data. We found that the methods of sharing vary strongly among our interviewees, and include emails, cloud storage services, scientific data platforms, and web sites or databases of research groups. Only about half of our interviewees stated that they share their code on platforms such as GitHub, using emails, or describe it in their publications. Others do not share it as they do not want it to be public or consider it as highly project-specific. Of the 17 interviewed researchers, 13 reported that they need to keep track of the provenance of their data. Further, we found that 13 of our interviewees are dealing with sensitive data, including, but not limited to, personal identifiable information. Security and privacy arrangements include access control checks using databases or protected files, encryption of data that is in transit, and de-identification of data prior to its publication.

3. Example applications

The following examples of monitoring and evaluation applications using ICTs helped inform Mezuri's requirements and design. These applications guided the development of the Mezuri prototype as archetypal use cases that combine sensors, smartphone surveys, and data storage, analysis, and sharing.

3.1. Cookstove use monitoring

One example of a typical monitoring and evaluation case study is our cookstove work in Darfur, Sudan. In this study we compared objective cookstove adoption measured by sensors versus user-reported adoption measured by surveys (Wilson et al., 2014).

The Berkeley–Darfur Stove is a high-efficiency wood-burning cookstove developed by Lawrence Berkeley National Laboratory and the University of California, Berkeley. As of December, 2014, more than 35,000 of these stoves had been distributed in and around internally displaced persons (IDP) camps in North and South Darfur. Beginning in July 2013, 180 participants were chosen by local camp leaders (Omdas) based on the criteria that participants had not previously been recipients of a Berkeley–Darfur Stove. Of the 180 participants, 170 received instrumented stoves that included sensors. The purpose of these sensors was to record a time-series of cookstoves' temperatures that could validate adoption of the cookstoves. Previous work has shown that temperature measurements can be used to detect and measure cookstove usage over time (Ruiz-Mercado et al., 2012). The sensors, termed Stove Use Monitors or SUMs, were built upon the Maxim DS1922E model iButton data logger.

Over the 3.5 months of the experiment, 180 participants were surveyed twice using ODK Collect for baseline and follow-up surveys. A third interaction took place in the form of a second follow-up when SUMs were removed from stoves. The five administrative units all received their cookstoves and baseline surveys over one week in July 2013. However, follow-up surveys were not conducted after the same interval for all groups. Instead, administrative units were followed up with one at a time at two-week intervals to spread out enumerator resources (the follow-up survey was much more onerous than the baseline survey) and to distribute sensors with different sampling rates across the population.

Please cite this article as: Kipf, A., et al., A proposed integrated data collection, analysis and sharing platform for impact evaluation. Development Engineering (2015), http://dx.doi.org/10.1016/j.deveng.2015.12.002

Data processing for both survey and SUMs data was performed using a series of R and MATLAB scripts. Source code was version controlled and shared via Dropbox. As the data analysis development was cleanly divided, this scheme worked well for this particular case study. The most significant intellectual effort of this case study was development of the algorithm that identified and labeled cooking events in the temperature time-series. Data processing operations included spot-checks, cleaning up data, normalizing data, generating the set of cookstove events using this algorithm, and finally creating summary statistics. In this case study, the spot-check workflow was largely ad hoc, Excel-based, and irreversible, while the data processing workflow was based in MATLAB and R and was non-destructive and reversible. Approximately 250 h of expert effort was required to write and validate code to analyze this dataset. This effort would have been beyond the capabilities of any non-technical organization (e.g., most small NGOs). Additionally, although this particular case study only had a single individual managing and performing data analysis, any future collaborative analysis would necessitate a more robust code sharing and version management system. Regarding security and management of data, the aforementioned data processing workflow using R and MATLAB scripts necessitates data leaving the data collection ecosystem of ODK/OneWire. This less controlled processing environment forfeits the data versioning and security infrastructure already in use in ODK. Finally, the techniques and environment used to process the data for this case study are not scalable or replicable in any meaningful way for other studies. Even if the processing scripts would be available via services such as GitHub, it is still hard for others to understand them. In most cases, cookstove researchers would need to engage computer scientists to set up the code and to connect data sources, which is both cost and time intensive.

3.2. Water filter use monitoring

In the context of a five-month randomized controlled trial of household water filters and improved cookstoves in rural Rwanda (Barstow et al., 2014), data was collected from intervention households on product compliance using (1) monthly surveys on smartphones (DoForms, an ODK commercial derivative) and direct observations by community health workers and environmental health officers, and (2) sensor-equipped filters and cookstoves deployed for about two weeks in each household (Portland State SWEETSense). This application demonstrates the combined value of sensors and surveys. The survey and sensor data was relayed over cellular networks to relational databases where the data was combined and processed with R scripts. The manual and custom elements of this study may be replaced by the Mezuri platform in subsequent deployments.

However, a question remains as to whether the usage of these monitoring devices influences the very behavior that they monitoring. In other words, does the presence of the monitoring instrument cause reactivity and thus present another source of bias. The second is whether instrumented monitoring presents an opportunity to provide feedback to the program population in ways that could encourage or reinforce the target behavior. In an ongoing study, researchers are undertaking a trial to characterize participant reactivity to instrumented monitoring.

3.3. Water service provisioning

To provide access to safe drinking water in support of the Millennium Development Goals, international donors and governments have installed improved water point sources throughout developing countries resulting in an apparent increase in access to improved water supplies in rural areas from 58% in 1990 to 79% in 2010 (WHO/UNICEF, 2012). However, reliable, sustained water service delivery remains a challenge (Foster, 2013). In rural sub-Saharan Africa, where hand pumps are the most common technology, 25–40% of improved water sources are non-functional at any one time, and many never get repaired (Foster, 2013; Ponce de Leon et al., 2015). This has resulted in an overstatement of the impact of improved water supply interventions. Non-functional water sources disproportionately affect women and girls who commonly are responsible for retrieving water for their households from often distant supplies when improved sources fail. Failure of improved water sources can have a significant impact on human health through the transmission of disease via unsafe water.

In 2014 and 2015, with funding support from UK DFID and the GSM Association, part of our research group installed 181 sensors in rural water pumps in Rwanda in a longitudinal study of the effectiveness of a sensor-based maintenance program (CellPump project). In three experimental arms, the current model of operation and maintenance was compared against a “best practice” circuit rider model that includes a “call us” feature for communities to report pump outages, against an “ambulance service” model where the sensors notified an online technician dispatch platform. The platform also collected smartphone maintenance records to map technician and service performance. Sensors were installed in all three arms, but only in the ambulance service were pump technicians privy to functionality data. The study ran for seven months between November 2014 and May 2015. An R program was used to process and analyze the sensor and maintenance team data. The group used fractional logit regression to model the relationship between pump functional time and maintenance condition, and accelerated failure time models to examine differences in time to repair following pump failure. We found that the “ambulance service” model that uses real-time sensors to monitor water pumps performed best compared to the other approaches.

Implementers of water, energy, and infrastructure projects may realize an economic incentive to utilize Mezuri-like systems. At the operational level, sensor and smartphone survey monitoring of water pumps has the potential to reduce system downtime, reduce the number of visits to a village that currently is part of a traditional circuit-rider model for manually monitoring pumps, and therefore reduce the cost per liter of water delivered. In real terms, this hypothesis has the potential to save critical operations and maintenance dollars from reduced site visits, while vastly improving data collection, increasing the quality of data, and improving overall project accountability to donors.

3.4. High resolution data for increasing grid flexibility

We are also participating in a project that collects high-resolution data to increase grid flexibility in high renewable energy and resource constrained environments.

The penetration of wind and solar renewable energy is occurring across many different regions, incomes, and levels of development. Nicaragua has emerged as an energy transition leader, producing ~40% of its total generation from non-large hydropower renewable resources, with 20% of total generation coming from wind energy alone (Ponce de Leon et al., 2015). Although this low-carbon transition is certainly laudable, the need for non-fossil grid flexibility resources has increased.

Together with Niuera, a National Geographic Energy Challenge Grantee based in Berkeley, the Renewable and Appropriate Energy Lab (UC Berkeley), and the Nicaraguan National Engineering University, we have recently begun a project in Managua to gather high-resolution demand side data from households and micro-enterprises with thermostatically controlled loads (TCLs, such as

Please cite this article as: Kipf, A., et al., A proposed integrated data collection, analysis and sharing platform for impact evaluation. Development Engineering (2015), http://dx.doi.org/10.1016/j.deveng.2015.12.002

refrigerators). The project integrates behavioral approaches to understand how users interact with large cooling loads, educational approaches to better inform users of the energy data that is being collected, and control and machine learning approaches for optimal power grid management. The project's main objective is the design and implementation of a wireless sensor network to enable flexible energy loads to provide cost-effective wind energy grid integration and societal co-benefits in Nicaragua.

The researchers integrate data from several sources. A tablet-based ODK baseline survey was implemented in 2014 on 300 micro-enterprises (chicken shops, meat shops, milk and cheese vendors, and mom & pop shops), and 40 micro-enterprises (treatment and control) and 20 households (treatment and control), which were randomly selected to participate in the project, with 30 treatment units receiving a FlexBox (Ponce de Leon et al., 2015). The FlexBox is the design of a wireless sensor network that monitors and controls TCLs, integrates TCL state information with household-level electricity metering, and combines this information with grid level data in the cloud. Sensor data includes inside refrigerator temperature, household ambient temperature, refrigerator electricity consumption, refrigerator door openings via a magnetic switch, and total household electricity consumption. Other data sources include weather data from a weather station close to the study units, high-resolution (hourly) grid generation and demand data from all technologies (wind, solar, geothermal, oil, etc.) in the country, monthly ODK table surveys on project participants, and text messages.

Although this research project is ongoing, data from surveys and sensors are beginning to shed light on the challenges and potential for demand response in countries similar to Nicaragua. Behavioral patterns with cooling loads are starkly different from those found in California, loads are particularly vulnerable to daily weather variations, and surveys and conversations suggest that incentives for future project participation could include both monetary incentives (micro payments) and "social contribution" cues (inviting project participants by suggesting they are contributing to society). Without the combination of sensors and surveys, such an objective and thorough evaluation would not be possible.

3.5. Commonality

The archetypal deployments described in the previous subsections describe how the combination of sensors and surveys can lead to an accurate evaluation of interventions. For example, the adoption rate interpreted by the sensors varied from the household reporting surveys. In the Sudan cookstove use case, 96.5% of households reported primarily using the intervention stove, while the sensors indicated 73.2% use. In the water filter use case, 96.5% of households reported using the intervention filter regularly, while the sensors indicated no more than 90.2%. The sensor-collected Rwanda data estimated use to be lower than conventionally-collected data both for water filters (approximately 36% less water volume per day) and cookstoves (approximately 40% fewer uses per week). An evaluation of intra-household consistency in use suggests that households are not using their filters or stoves on an exclusive basis, and may be both drinking untreated water at times and using other stoves ("stove-stacking"). These results provide additional evidence that surveys and direct observation may exaggerate compliance with household-based environmental interventions (Thomas et al., 2013). Additionally, the results from the water service provisioning study demonstrated that sensor-based notifications to online dashboard platforms used to trigger maintenance services resulted in significant improvements in both time to repair after pump failure and overall functional time when compared to both the nominal service model and a "best-practice" circuit-rider approach.

The above use cases illustrate that the authors were repeatedly involved in projects that required a significant amount of infrastructure to be built to combine the processed and analyzed sensor data with the corresponding survey data. Mezuri was born out of the idea that there are many metrics and evaluation use cases that have similar properties: (1) combining remote sensing data with survey data could improve evidence-based research; (2) a domain expert was needed to write the custom data processing scripts; (3) while data collection tools such as ODK have simplified collection of data, domain users were struggling to easily join sensing and surveying data together. Looking at these commonalities exposed an opportunity to build a generalizable platform that is designed to take remote sensors readings, process the readings with customized code, and analyze the resulting data. We also observed that custom data processing pipelines were continually being recreated for new but similar projects, making it difficult for the processing and analysis methodology to be shared and applied to other projects. Our overarching goal in creating Mezuri is to create a platform that people will want to use since it will save them both time and resources and enable the greater development community to benefit from shared knowledge and interoperability.

4. Design requirements

The Mezuri platform addresses three key functions of any global development data collection effort, including the collection, processing, and sharing and analysis of data.

4.1. Collection

Because of varying deployment requirements, Mezuri is intended to be flexible with regards to where in the end-to-end system the storage and signal processing of sensor data occurs. Therefore, Mezuri should allow raw data processing to be performed anywhere, including a mobile device, independent infrastructure, or even the Mezuri cloud infrastructure itself.

Surveys: Many development interventions focus on human impact and frequently measure this impact by surveying the recipients of the intervention, making surveys a key M&E tool. Electronic survey collection involves the task of encoding a survey in a format suitable for data collection, deploying the survey to collection devices, conducting the survey, and storing survey responses securely. Data privacy is valued by interviewees and researchers, and it is an essential requirement in any research involving humans.

There are organizations that are currently using cell-phone based surveys and Internet-based visualization for data collection and communication from the field (e.g., Akvo/Water for People FLOW, World Bank WSP, mWater, and mWash).

Sensors: Surveys and other common methods for assessing behavioral practices are known to have certain methodological shortcomings. Surveys often overestimate adoption rates due to reporting bias where the participant is trying to please the surveyor, or recall bias where the participant does not remember the information correctly (Manun'Ebo et al., 1997; Stanton et al., 1987). Additionally, it is known that the act of surveying can itself impact later behavior (Zwane et al., 2011).

The use of instrumentation to provide feedback on water, sanitation, energy and infrastructure activities is not entirely new, though it is still largely confined to research applications.

4.2. Processing

Once data is collected, it is processed into a usable form,

Please cite this article as: Kipf, A., et al., A proposed integrated data collection, analysis and sharing platform for impact evaluation. Development Engineering (2015), http://dx.doi.org/10.1016/j.deveng.2015.12.002

requiring data cleaning, signal processing, and analysis, such as event detection or correlation of data. Often, researchers correlate survey and sensor data to get a more accurate picture of an intervention and find discrepancies in the data. Geo-spatial and temporal queries are common practice. Survey data is often encoded into a normalized form suitable for computing statistics and demographic information. After cleanup, final processing may convert sensor data into events and behaviors, such as flow-rate and pressure changes into water use measurements (Thomas et al., 2013) or temperature changes into cooking events (Wilson et al., 2014). Survey data may also be used to inform these transformations. Finally, as researchers in different fields have divergent skill sets they often want to process their data with familiar tools. Thus, Mezuri needs to be designed with a rich processing component that supports multiple programming languages (e.g., R, Python) for different user capabilities and phases of the data processing.

4.3. Analysis and sharing

For the Mezuri platform to successfully support a variety of projects and professionals with varying degrees of technical expertize, it is important for the design to include simple user-interfacing components. For example, individual tasks can be built by filling in one or multiple inputs and a processing tool that can operate on these inputs. The user may see which tasks he or she has run based on whether those tasks are finished or still processing, and has the option to terminate tasks in progress. The user may also schedule tasks (e.g., periodically cleaning data as a device reports to Mezuri) or build workflows by chaining together tasks using processing tools that have compatible output and input schemas as illustrated in Fig. 1. Mezuri's web interface will ultimately contain an online suite of shared analysis tools and visualization options. Many users who transition to the platform may already have sophisticated software and methods for doing analysis and visualization. Therefore, Mezuri also needs to allow users to export data in various formats, including CSV and Google Fusion Tables, although security guarantees no longer apply once data is exported. Other tabs on the same page provide information about the Mezuri project, raw data exploration, and optionally the visual output of finished tasks. Users can then share their data along with its provenance with others.

4.4. Derived engineering requirements

Engineering requirements for Mezuri presented in Table 1 were derived from combining key functional requirements with compiled interview responses and the archetypal case studies.

5. Design

In this section, we discuss the preliminary design of the proposed Mezuri platform architecture aimed at addressing the above derived engineering requirements. For the sake of expediency, the first prototype of the Mezuri platform was built on top of existing ODK 2.0 infrastructure to enable the seamless integration of existing ODK 2.0 tools. Future versions of the platform will be based on updated ODK APIs as well as APIs of other data collection platforms. The overall Mezuri data workflow involves three steps, namely: the collection and import of data into the platform through surveys and sensors; the processing of data; and the analysis and sharing of data. Designing a one-size-fits-all platform for a diverse set of development organizations and researchers is difficult because of the varying scientific domains and divergent development use cases. Mezuri's design enables users to define their own data processing tools to customize the transformation of data and maximize user flexibility, while Mezuri tracks the provenance of the data. Mezuri's use of Docker containers allows users to integrate code written in a variety of programming languages (e.g., R, Python) into their workflows. Fig. 2 illustrates how data flows through the Mezuri platform.

Users can upload their data into the Mezuri Cloud in three different ways. First, they can upload intermediately stored data in batches, which is especially useful when data collection devices do not have an Internet connection. Second, data can be streamed directly to the platform, which is particularly important for time-critical applications, such as power monitoring solutions. Third, Mezuri platform is capable of pulling data from external sources on a regular basis. For example, users might want to pull weather data from public web services to use it within their workflows. As per default Mezuri uses the well-known CSV format to import data into the platform, exchange data between user code and internal datastores, and export data. Users can either use provided client programs to send their CSV data to Mezuri or directly communicate with REST services exposed by Mezuri. When importing data, users are prompted to provide information about the data's schema and provenance. Once imported, data is stored in cloud databases with high availability and durability guarantees.

The provenance requirement necessitates the use of an append-only write pattern. Longitudinal data is handled as successive versions on the original data. For example, periodic household surveys would lead to multiple versions.

Often, researchers create their own schemas by defining questions in surveys and the type of sensing data gathered. To have a broad choice of available processing tools for collected data, a common schema can be beneficial for a platform like Mezuri as it can ensure that data input into the system is in a proper format to be handled by existing tools. On the other hand, because many schemas are exceedingly domain specific, Mezuri will not enforce

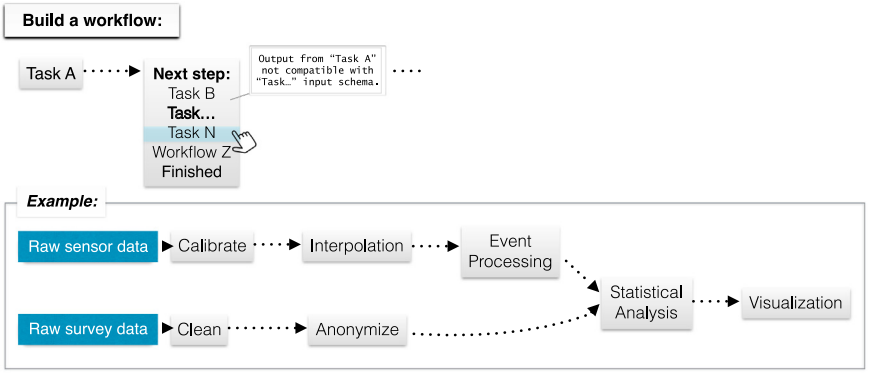

Fig. 1. Title: web interface to build a workflow. Description: this figure illustrates Mezuri's web interface for building a workflow.

Please cite this article as: Kipf, A., et al., A proposed integrated data collection, analysis and sharing platform for impact evaluation. Development Engineering (2015), http://dx.doi.org/10.1016/j.deveng.2015.12.002

Table 1
Derived engineering requirements.

Abstraction	The system shall allow interchanging of components of the platform (e.g., the backing datastore) to address varying user needs (e.g., storing both survey and sensor data). Additionally, an abstraction layer shall allow authentication throughout the platform.
Accuracy and transparency	All imported data and any subsequent corrections to that data are retained and held distinct from each other within the datastore (immutable). Each correction maintains provenance information distinct from that of the imported data (traceable). The system shall be capable of re-running a processing step upon a specific snapshot of a corrected data set and this shall produce the same results (deterministic).
Common schema	The schema of the data and the interfaces of the processing may be standardized to allow automatic determination and recommendation of processing tools.
Durability	The storage service shall be designed in a way that prevents any data loss when any single process of the system fails. Similarly, the processing of data shall support fail-overs to avoid recalculating entire workflows.
Isolation	Processes in the platform shall be isolated from each other to avoid side effects. Further, any user-defined code shall be isolated in its own runtime to protect the platform from malicious code.
Privacy	Often, data collected is sensitive and controlled by institutional review boards. The system shall be compatible with privacy, security and anonymization requirements of most IRBs.
Provenance	The platform shall track transformations performed on the data, including user modifications, to be auditable and to support repeatable workflows. At the same time, external provenance metadata shall be imported into the platform to have a complete trace of the provenance chain.
Revision of provenance metadata	Provenance metadata shall be immutable by default, however, updates to sensor configuration may occur after the data was imported into our platform. For auditing purposes, Mezuri shall track the history of any update.
Scalability	As the volume of data increases and the number of users grows, the platform needs to be able to scale to meet the additional data and processing requirements.
Sharing	Mezuri shall allow researchers to share their data, processing tools, and workflows within and outside of the platform. When sharing their tools within the platform, users should be able to share them as black boxes that allow others to process their data without having access to the actual (sensitive) code. Shared workflows should include initial input parameters and input data sets. Workflow owners should be able to control at which granularity level someone can access the workflow.

any single schema. Instead, we believe that users will be incentivized to use common schemas if they want to leverage available processing tools, thereby increasing the comparability of shared data from different projects.

For processing data, users can either upload their own tools or choose from a list of available tools. Users can write processing tools in a variety of programming languages (e.g., R and Python) and incorporate them into their workflows. In other words, users can continue using the programming languages that they are used to and can even mix them within workflows. When uploading their own tools, users provide the source code implementing the logic of their processing steps, dependencies (e.g., required libraries), and the schemas of the inputs and outputs. This allows the system to match data tables with compatible processing tools. The key idea is to enable users to reuse processing tools provided by others and apply them to their own datasets without requiring any source code modifications. Processing tool authors have multiple advantages when openly sharing their tools, including that others will test and extend their tools, and make their data adhere to the tools' required input schemas, which will make their data comparable to the tool authors' own data.

For the integration of these user-defined tools, we considered three different tool execution environment options: (1) externally on web services, (2) on virtual machines (VMs), or (3) within Docker containers running on VMs. Other platforms such as Kepler (Altintas et al., 2004) allow users to integrate tools in the form of web services. While that is a good approach for the integration of sensitive code that otherwise cannot be integrated into the platform, it is a lot of effort for users to set up and maintain. In particular, users would need to define interfaces to those services, which might already be too difficult for most researchers in our target group. A second option was to run tools in individual VMs. While this approach yields a good isolation of user code, the memory and disk footprint of VMs is significant. In the interest of efficient use of resources, we thus would need to run multiple of these user-defined tools on a single VM, which leads to data security and performance isolation issues. Using Docker not only allows us to run a large number of tasks on a single VM and to set up user-defined execution environments within seconds, but also to package the dependencies of user code in a reusable way for others. Since Docker containers can be ported to any Unix-based system that runs Docker, users can test their tools locally before submitting them to the platform. Similar to how a source control system such as Git allows us to version user code, Docker allows us to version the execution environment, particularly the operating system and libraries used to run the code. By combining those two, we can capture the full provenance information of the tools used to transform data, which is a key requirement of our platform.

The drawback of using Docker is that it adds complexity to the platform. Ideally, the complexity of using Docker has to be hidden

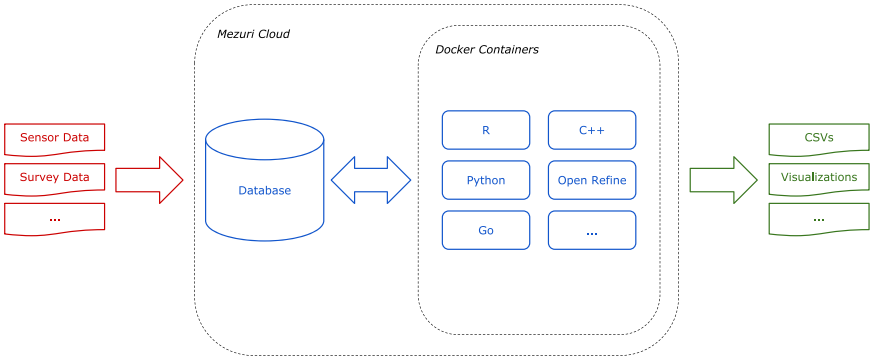

Fig. 2. Title: Mezuri data flow. Description: this figure illustrates the different components of the Mezuri platform and visualizes the data flow from the raw data (e.g., surveys and sensor data) to the final output data.

Please cite this article as: Kipf, A., et al., A proposed integrated data collection, analysis and sharing platform for impact evaluation. Development Engineering (2015), http://dx.doi.org/10.1016/j.deveng.2015.12.002

Fig. 3. Title: Workflow combining different programming languages. Description: this figure illustrates an example workflow that processes data using different programming languages.

from users to lower the entrance barrier. Thus, we need to provide users with an easy-to-use configuration interface where they can specify their code's dependencies and a rich set of simple language bindings enabling them to use their preferred programming language in the usual way.

Fig. 3 shows a workflow that combines different programming languages.

In such a workflow, users can mix processing tools written in different programming languages as long as the output and input schemas are compatible. Optionally, interactive processing tools, such as Open Refine, can be integrated into workflows. Mezuri maintains provenance by capturing the exact data sets that were used as inputs to user-defined tools. This allows users to show the differences between input and output data sets as well as to keep track of joins of multiple data sets. Most importantly, it enables users to drill down into how output data was generated. Similarly, the provenance data allows others to learn about data processing and analysis practices from existing workflows. The development of a mechanism that allows users to define how data is being transformed within their tools is left for future work.

Mezuri is based on a workflow design and exposes functionality as workflow components. All components within Mezuri are exposed as services to enforce access control and to keep the underlying components independent from the overall platform. There are three main services in Mezuri that abstract users from the implementation complexities of the overall data and workflow control logic. The Data Service abstracts logical datastores into a single service that can be further divided into specialized physical datastores (e.g., a relational database for survey data, a non-relational time-series database for sensor data) that contain survey data, sensor data, metadata and platform configuration data. The Tool Service stores and retrieves processing tools and Docker images enabling tool sharing between users and instantiates tools in the form of Docker containers. Further, the Tool Service tracks tool versions, including source code and execution environments, to allow for provenance queries. Finally, the Task Service manages data processing and analysis by creating and monitoring tasks and workflows.

Mezuri's design distinguishes between (1) definition and (2) provenance metadata. Definition metadata describes the schema of each set of imported data. For example, in a time-series of sensed temperature data researchers would specify the unit and the precision of the collected data as the definition metadata. Provenance metadata provides historical reference information and is generated as either internal or external provenance metadata to the Mezuri platform. Mezuri's internal provenance metadata enables researchers to understand which tasks were applied to their data to produce the results. This metadata includes tools, version information, inputs, and parameters used thereby enabling users to replay data workflows with the same or different data. External provenance metadata describes how the data input into Mezuri was initially captured, such as information about time, location, and survey and sensor configurations.

In a first prototype, we have implemented lean versions of the aforementioned services on top of the Google Cloud Platform. The Data Service uses ODK's 2.0 implementation with a Cloud SQL database as backend. The Tool and the Task Service are implemented using Cloud Endpoints that expose REST APIs. Further, we use the Docker Hub Registry to store Docker images as well as GitHub to manage user code. Docker containers are invoked and monitored using a custom Java implementation running on Compute Engine. Additionally, we implemented a language binding for R that allows users to communicate with the Data Service from within their tools. Input data is parsed into a R data frame, processed by custom user code and eventually passed back to the Data Service. Information about input and output data sets, processing tools, and execution environments used to transform data is captured in dedicated metadata tables in the Cloud SQL database. Using this metadata and recursive SQL queries, we were able to reconstruct the provenance information of given data sets and to find downstream effects of potentially erroneous data sets or tools.

Without disregarding the security literature on known cloud vulnerabilities, we assume one project owner using any popular cloud service for all of his or her data management and processing tasks is a reasonably secure environment for any potential Mezuri use cases. We therefore focus our attention on mitigating the risks that come along with collaboration, especially between parties who may not fully trust each other with their data or processing code. Our solution is a role-based access control system in which collaborators can be registered to have one or more user roles, each of which has corresponding intentions and privileges (see Table 2). One privilege is to be able to read aggregated data. In this case, a user or organization has valid reasons for wanting to learn accurate statistics from the database, but should not be able to learn anything about one or a few specific individuals in the dataset. An aggregation service will decide whether a user will be allowed to access a certain query result based on the user's trust level, a tunable measure of aggregation similar in spirit to k in the concept of k-anonymity, and an analysis of the user's previous requests. Specifically, access to a dataset is denied if the requested statistic is based on too few individuals or if the statistic could be combined with any previous result to learn information about too few individuals. A larger measure of aggregation designates a more privacy-preserving policy, although no aggregated data

Table 2
User roles, intentions, and privileges for access control.

User Roles	Intentions	Privileges
Project owners	Full access, zero overhead, manage collaborators	Read original data+collaborators' aggregated data, write, add or revoke users
Data collectors	Contribute data	Append
Funding agencies/partnering NGOs	Learn about the population, monitor deployment progress	Read aggregated data with a medium or high degree of trust
Colleagues/researchers	Test code	Read synthetic, noisy and/ or aggregated data with a high degree of trust
Potential collaborators	Contribute data, test code, learn about the population	Append, read contributions+aggregated original data with a low or medium degree of trust
Other researchers	Learn about populations	Read synthetic, noisy and/ or aggregated data with a low degree of trust

Please cite this article as: Kipf, A., et al., A proposed integrated data collection, analysis and sharing platform for impact evaluation. Development Engineering (2015), http://dx.doi.org/10.1016/j.deveng.2015.12.002

schemes are perfectly private while still providing useful information.

6. Future work

In this section, we will discuss some thoughts regarding technology challenges that we plan to tackle in the future.

Fully-featured database systems such as PostgreSQL offer a rich tool set to analyze data. With Mezuri, we aim to make these kind of tools available to a broader audience through easy-to-use interfaces. Our goal is to enable researchers to bring in their domain expertize in creating processing tools and workflows without needing them to develop system administrator skills. Additionally, researchers will have multiple benefits when using these systems through our platform which they would not have when using them standalone, including that our platform will keep track of provenance, care about durability, and offer rich data and tool sharing options. In other words, users will be able to leverage the functionalities of these systems (such as the geospatial capabilities of PostgreSQL) in their preferred programming languages without having to learn how to setup, use, and maintain these systems.

Also, researchers may need to process large amounts of data, a task that can be computation and memory intensive. To utilize the full potential of the cloud, we need to enable researchers to write scalable code that can be executed within the Mezuri architecture. Again, the challenge is to hide the complexity (of parallelization and distributed computation in that case) behind simple interfaces.

The goals we describe in terms of security and privacy can be strengthened by enforcing user roles and privileges with IAM (Identity and Access Management), offered by AWS and soon Google Cloud Platform. In addition to the aggregation privilege and supporting service described above, we can implement other similar privacy policies as enhanced permissions. In particular, Differential Privacy (Dwork, 2011) adds noise to query results proportional to sensitivity and can be used to generate synthetic datasets.

7. Conclusions

Taken together, the emergence of low cost cellular-based sensors and smartphones with online data analysis and management may improve the performance and transparency of global development interventions.

In this paper, we presented requirements gathered from various sources and several motivating real-world case studies that drove and verified the Mezuri design. We described the components of the Mezuri platform and how it addresses the needs of M&E and development engineering projects. The concept architecture is a major step towards building the Mezuri platform. We gave an overview how our platform can address and solve common problems and limitations of current workflows.

Our primary contributions include a comprehensive requirements analysis for a data platform for global development professionals based on user studies and the examination of the landscape through case studies. Additionally, we proposed a system design that addresses these requirements and demonstrated its feasibility through the implementation of a first prototype that uses Docker to enable users to process their data using a variety of programming languages. Mezuri will allow researchers to spend less time setting up backend infrastructure for data storage and processing and will allow them instead to focus on their research itself. Our platform will lower the barrier for less experienced researchers to collect and process data produced by emerging sensor technologies. Mezuri will support researchers' workflows from the point of raw data collection to the processed end results, while keeping track of end-to-end provenance of the data. We will allow users to share existing building blocks for processing and analyzing data and thus we minimize redundant efforts and promote knowledge exchange. Mezuri will also reduce costs and errors that can occur due to a lack of traceability. Finally, Mezuri will enable funders to measure and monitor the impact of projects more accurately to make more effective allocation decisions.

Acknowledgments

This paper is dedicated to our dear friend and colleague Gaetano Borriello, who passed away during the publication process. Gaetano was not only one of the key leaders of the Mezuri project – and the person who gave it its name – but a true pioneer in the field of ICTD and the visionary leader of the ODK project. His work has benefited thousands of development projects and millions of people. We honor his legacy by continuing our efforts to "magnify human resources through technology" by adapting technology to work in resource-constrained environments. Gaetano was one of the first to point out that technology is not the solution, but only a tool to help people in global development. Technology can help people become more effective and help to stretch limited resources further by improving processes.

The authors also wish to acknowledge the contributions of Diego Ponce De Leon Barido, Bradford Campbell, Prabal Dutta, Noah Klugman, Clarice Larson, Madhav Murthy, and Pat Pannuto to this project. This work was supported by the Development Impact Lab (USAID Cooperative Agreement AID-OAA-A-13-00002), part of the USAID Higher Education Solutions Network. Andreas Kipf was supported by a fellowship within the FITweltweit program of the German Academic Exchange Service (DAAD).

References

Altintas, I., Berkley, C., Jaeger, E., Jones, M., Ludascher, B., Mock, S., 2004. Kepler: an extensible system for design and execution of scientific workflows. In: Scientific and Statistical Database Management, 2004. In: Proceedings of the 16th International Conference on, IEEE, pp. 423–424.

Barstow, C.K., Ngabo, F., Rosa, G., Majorin, F., Boisson, S., Clasen, T., Thomas, E.A., 2014. Designing and piloting a program to provide water filters and improved cookstoves in rwanda. PLoS One 9 (3), e92403. http://dx.doi.org/10.1371/journal.pone.0092403.

Brunette, W., Sundt, M., Dell, N., Chaudhri, R., Breit, N., Borriello, G., 2013. Open data kit 2.0: expanding and refining information services for developing regions, in: Proceedings of the 14th Workshop on Mobile Computing Systems and Applications, ACM, p. 10.

Dwork, C., 2011. Differential privacy. Encyclopedia of Cryptography and Security, pp. 338–340.

Foster, T., 2013. Predictors of sustainability for community-managed handpumps in sub-saharan africa: evidence from liberia, sierra leone, and uganda. Environ. Sci. Technol. 47 (21), 12037–12046.

Hull, D., Wolstencroft, K., Stevens, R., Goble, C., Pocock, M.R., Li, P., Oinn, T., 2006. Taverna: a tool for building and running workflows of services. Nucl. Acids Res. 34; , pp. W729–W732.

Linke, B., Giegerich, R., Goesmann, A., 2011. Conveyor: a workflow engine for bioinformatic analyses. Bioinformatics 27 (7), 903–911.

Manun'Ebo, M., Cousens, S., Haggerty, P., Kalengaie, M., Ashworth, A., Kirkwood, B., 1997. Measuring hygiene practices: a comparison of questionnaires with direct observations in rural zaire. Trop. Med. Int. health 2 (11), 1015–1021.

Manya, A., Braa, J., Øverland, L., Titlestad, O., Mumo, J., Nzioka, C., 2012. National roll out of district health information software (DHIS 2) in kenya, 2011–central server and cloud based infrastructure. In IST-Africa.

Miceli, G., Pekkarinen, A., Leppanen, M., 2011. Open foris initiative—tools for forest monitoring and reporting. Concept Note.

Ńeron, B., Ḿenager, H., Maufrais, C., Joly, N., Maupetit, J., Letort, S., Carrere, S., Tuffery, P., Letondal, C., 2009. Mobyle: a new full web bioinformatics framework. Bioinformatics 25 (22), 3005–3011.

Pannuto, P., Campbell, B., Dutta, P., 2013. GATD: a robust, extensible, versatile swarm dataplane. In: Proceedings of The First International Workshop on the Swarm at the Edge of the Cloud, SEC'13.

Please cite this article as: Kipf, A., et al., A proposed integrated data collection, analysis and sharing platform for impact evaluation. Development Engineering (2015), http://dx.doi.org/10.1016/j.deveng.2015.12.002

Ponce de Leon, D., Barido, J., Johnston, M., Moncada, D., Callaway, D., 2015. M Kammen, "Evidence and Future Scenarios of a low-carbon transition in Central America: A Case Study in Nicaragua. Environ. Res. Lett. 10, 104002.
Ruiz-Mercado, I., Canuz, E., Smith, K.R., 2012. Temperature dataloggers as stove use monitors (sums): field methods and signal analysis. Biomass Bioenergy 47, 459–468.
Stanton, B., Clemens, J., K, A., Rahman, M., 1987. Twenty-four hour recall, knowledge-attitude-practice questionnaires, and direct observations of sanitary practices: a comparative study. Bull. World Health Organ. 65 (2), 217–222.
Thomas, E.A., Barstow, C.K., Rosa, G., Majorin, F., Clasen, T., 2013. Use of remotely reporting electronic sensors for assessing use of water filters and cookstoves in rwanda. Environ. Sci. Technol. 47 (23), 13602–13610.
Thomas, E., Zumr, Z., Barstow, C., Fleming, M., Spiller, K., 2013. Remotely accessible and reconfigurable in-situ instrumentation to improve monitoring of global development interventions, sustainability.
WHO/UNICEF, 2012. Joint monitoring programme for water supply and sanitation data tables. ⟨http://www.wssinfo.org/data-estimates/tables/⟩.
Wilson, D., Idris, M.A., Abbas, O., Coyle, J., Kirk, A., Rosa, J., Gadgil, A., 2014. Comparing cookstove usage measured with sensors versus cell phone-based surveys in darfur, sudan. Technologies for Development: What is Essential?, EPFL.
Zwane, A., Zinman, J., Dusen, E., Pariente, W., Null, C., Miguel, E., 2011. Being surveyed can change later behavior and related parameter estimates. Proc. Natl. Acad. Sci. 108, 1821–1826.

Please cite this article as: Kipf, A., et al., A proposed integrated data collection, analysis and sharing platform for impact evaluation. Development Engineering (2015), http://dx.doi.org/10.1016/j.deveng.2015.12.002