

# Makahiki+WattDepot: An open source software stack for next generation energy research and education

Philip M. Johnson<sup>1</sup>, Yongwen Xu<sup>1</sup>, Robert S. Brewer<sup>1</sup>, George E. Lee<sup>1</sup>, Andrea Connell<sup>1</sup>

<sup>1</sup>University of Hawaii at Manoa, POST Building, Room 317, 1680 East-West Road, Honolulu, HI 96822 USA  
johnson@hawaii.edu, yxu@hawaii.edu, rbrewer@hawaii.edu, gelee@hawaii.edu, connell4@hawaii.edu

## ABSTRACT

Need abstract here

## Keywords

smart grid, energy research, open source, game engine, energy repositories

## 1. INTRODUCTION

Several characteristics of the traditional electrical energy infrastructure of industrial societies have remained unchanged for almost 100 years. First, energy production has been centralized in power plants using “firm” energy sources such as coal, oil, nuclear, or hydro. Second, centralized production has promoted centralized control of the grid, typically through a single or small number of utilities with public regulation over their policies and rates. Third, centralized production and control has led to the predominance of “macro-grids”, or grid infrastructures designed to service hundreds of thousands to millions of consumers. Finally, traditional grids have been designed to minimize the information about energy required by consumers to utilize the service. The typical consumer needs to know almost nothing more than how to plug an appliance into an outlet, and can assume that this exceedingly simple “user interface” will provide virtually unlimited amounts of high quality energy at any time for a relatively small cost.

Unfortunately, the accelerating world-wide growth in demand for energy is leading to a breakdown in this approach to electrical energy infrastructure. Petroleum products such as coal and oil are nonrenewable, are no longer reliably inexpensive, and have been found to have a variety of adverse environmental effects. Nuclear energy, while low in emissions, has risks that have caused countries such as Japan and Germany to reevaluate this approach to energy generation.

Addressing these emergent problems has led to the conceptualization of a “smart grid”, where a variety of decentralized, intermittent, renewable energy sources (for example, wind, solar, and wave) would provide most or all of the power required by small-scale “micro-grids” servicing hun-

dreds to thousands of consumers. Such a smart grid will require consumers to transition from passive to active participation in maintaining efficient and effective use of the grid’s electrical capabilities. For example, these “smart consumers” should be able to tailor their use of electrical energy to the types and amount of energy available in the grid at any point in time; minimizing the overall use of non-renewable resources as well as peak loads on the grid.

Satisfying the radically different requirements and operating assumptions of this next generation grid requires new kinds of software that enable research and experimentation into the ways that electrical energy production and consumption can be collected, analyzed, visualized, and provided to consumers in a way that enables a transition from passive to active participation. Since 2009, we have been designing, implementing, and evaluating an open source software “stack” to facilitate this research. This software stack consists of two custom systems called WattDepot<sup>1</sup> and Makahiki<sup>2</sup>, along with the open source components they rely upon (Java, Restlet, Postgres, Python, Django, Memcache).

In the full version of this paper, we will detail the novel features of WattDepot and Makahiki, our experiences using them for research and education, and additional ways they can be used for next generation energy research and education. The remainder of this extended abstract briefly introduces WattDepot and Makahiki and our research experiences to date.

## 2. WATTDEPOT

Software for energy collection, storage, and analysis tends to come in two flavors that support two ends of the scalability spectrum. At one end are utility-scale SCADA systems and protocols which are intended to manage macro-grid data [5, 6, 8]. At the other end are “personal scale” systems such as those provided by energy meter or solar panel manufacturers which are intended to manage information about single households [3, 9]. We designed WattDepot to support a middle ground that we refer to as “enterprise-level” energy management, in which data concerning energy production and consumption of hundreds to thousands of households can be usefully managed. Figure 1 illustrates the architecture of the system, where WattDepot sensors send data from meters attached to energy devices to a server,

---

ICT4S 2013: Proceedings of the First International Conference on Information and Communication Technologies for Sustainability, ETH Zurich, February 14-16, 2013. Edited by Lorenz M. Hilty, Bernard Aebischer, Göran Andersson and Wolfgang Lohmann.  
<http://e-collection.library.ethz.ch>

<sup>1</sup><http://wattdepot.googlecode.com>

<sup>2</sup><http://github.com/csd1/makahiki>

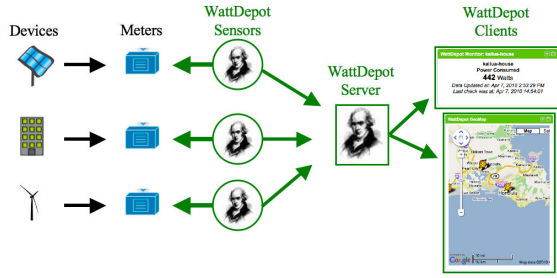


Figure 1: Architecture of WattDepot

which can then be queried by clients to provide visualizations and analyses.

Our use of WattDepot has led to a novel set of capabilities to support this middle ground.

First, unlike personal-scale systems that are typically tied to a particular manufacturer’s product, WattDepot is agnostic about the kinds of meters used to monitor energy production and consumption data, and whether the data is personal-scale or utility-scale. It provides a REST protocol for data transmission that can be used to implement clients for a wide variety of devices; the major constraint is that these devices need to have network access. WattDepot clients can be written in any language that supports the HTTP protocol. We provide a high-level client libraries for Java and JavaScript.

Second, WattDepot can represent aggregations of power sources. For example, a building might have multiple meters monitoring energy consumption, one per floor. WattDepot can represent the power consumed by individual floors, as well as an aggregate source representing the building as a whole. Aggregations can be nested, so that floors can be aggregated into buildings, buildings into neighborhoods, and neighborhoods into cities.

Third, WattDepot automatically performs data interpolation when necessary. For example, a meter might provide a snapshot of energy usage once per hour for a given device. Clients can request the power consumed by this device at any time instant, and WattDepot will automatically provide interpolation when the requested time does not match a time for which actual sensor data is available.

Fourth, WattDepot is architecturally decoupled from the underlying data storage technology. This supports experimentation with both traditional relational as well as NoSQL technologies, and facilitates scalability. Currently, WattDepot implements support for Derby, Postgres, and BerkeleyDB storage systems.

Fifth, WattDepot is designed to support both Platform-as-a-Service (PaaS) and local installation. We have successfully deployed WattDepot to the Heroku cloud-based hosting service.

Sixth, WattDepot implements support for “ephemeral” data.

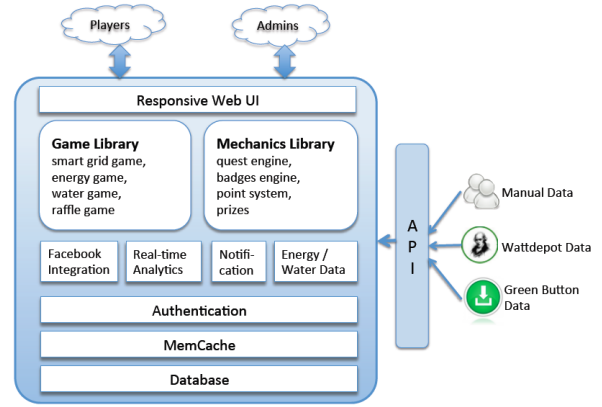


Figure 2: Architecture of Makahiki

In some application scenarios, it is useful to send energy data to the WattDepot server quite frequently (i.e. every few seconds) so that clients can monitor current energy consumption with low latency. However, that rate of data sampling is not necessary for historical analyses, which may only require energy data sampling at the rate of every few minutes. WattDepot supports this situation through ephemeral data, which creates an in-memory window during which all recently received energy data is available for retrieval, but stored in the repository only at a much lower sampling rate.

Finally, WattDepot can be effectively used for simulation and what-if scenario development, as well as for management of live energy data. This makes it appropriate as a kind of technological “scaffolding” for smart grid applications, where WattDepot can provide clients with simulated production and consumption data early in development, with the simulated data transitioning to live data as these sources go online later in development.

### 3. MAKAHIKI

The feature set of WattDepot creates attractive infrastructure for management of energy data, but research suggests that effective participation of consumers in a next generation smart grid requires more than simple feedback to consumers about their consumption, particularly given the passive nature of their involvement for the past 100 years.

The second component of our open source software stack, Makahiki, represents research intended to create synergy between the need to create knowledge and engagement regarding energy and the ability of so-called “serious game” techniques and energy feedback to create participation and engagement [1, 2, 4, 7]. In Makahiki, online game mechanics are employed with the goal of affecting real-world energy behaviors. The ultimate goal is to not just affect energy behaviors during the course of the game, but to produce long lasting, sustained change in energy behaviors and outlooks by participants. Figure 2 illustrates the architecture of Makahiki.

Makahiki consists of a configurable game engine that can be customized to the needs of different organizations. It includes a library of pre-built game “widgets” that imple-

ment a variety of game mechanics. Using the widgets, an organization can create a custom energy challenge in which players can compete individually and/or in teams to earn the most points by reducing their energy consumption as well as by learning about energy concepts in general. Some of the pre-built widgets include:

### 3.0.1 Smart Grid Game

The *Smart Grid Game* widget shown in Figure 3, is the primary place players go to learn about energy issues and earn points. Actions are organized into a grid of squares (hence the name “Smart Grid”) and organized by category columns. The grid contains four different types of actions: activities, commitments, events, and excursions.

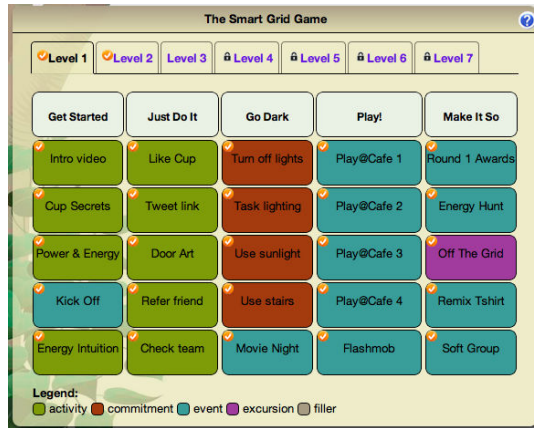


Figure 3: *Smart Grid Game*

*Activities* are the most basic task available in the Smart Grid. In order to get points for an activity, a player will have to provide a response to the administrators. These responses can be a short textual answer or an uploaded picture. Administrators access a special admin section of the web application to approve or deny submissions. If a submission is approved, the player will receive the points for their submission. Otherwise, a player will be sent a website notification informing them that their submission was not approved, and a textual description by an administrator of why it was rejected. The player can change and resubmit their response and still earn the full point value for that task.

*Commitments* are pledges that the player will do something sustainable for a period of five days. Examples include: reducing shower time, taking the stairs, and turning off the lights when leaving a room. Because these commitments are not verifiable, they are worth fewer points than activities. Furthermore, a player can only have up to five active commitments at any given time. After the five day period is up, the player can then declare that they completed the commitment and immediately earn their points. They can then sign up for another commitment, including the one they just completed.

*Events and excursions* are tied to real world activities. Events are held on campus while excursions take place off campus. Seating is limited, so players are asked to sign up for events they wish to attend. Players that do so are provided with a 2 point signup bonus. Players can also set up a reminder that

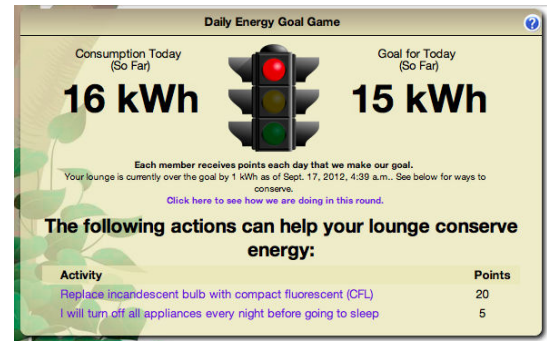


Figure 4: *Daily Energy Goal Game* visualization

is sent to their email and/or their mobile phone before the event takes place. At the event, an administrator will hand out attendance codes printed on slips of paper that can be entered on the website. These attendance codes are generated by Makahiki and can only be used once. To discourage players from signing up and not attending, a 2 point penalty is assessed to players who do not submit an attendance code. If the player submits an attendance code for the event after receiving this penalty, the penalty is reversed.

Not all of the tasks in the Smart Grid Game are necessarily available at the start of the game. We implemented a set of predicates that can be used to determine if a task is locked or unlocked for a player. These predicates include: completed a certain number of tasks within a category, completed all tasks within a category, completed certain tasks, and time-based unlocking (available after a certain date).

These predicates are implemented using a limited subset of Python and can be changed within the Django admin interface. Competition designers can use logical operators to combine any of these functions in order to organize the players' path through the Smart Grid Game.

### 3.0.2 Daily Energy Goal Game

The *Daily Energy Goal Game* widget provides a way for players to earn points by reducing their current consumption from a baseline that is typically determined prior to the challenge. Both the historical baseline data and the current consumption is typically provided by API calls from Makahiki to an underlying WattDepot server.

The goal for each lounge is typically a percent reduction from their baseline usage of the previous two weeks. When a player goes to the energy page of the web application, they can view their lounge's current progress toward their daily energy goal. Near the end of the day, Makahiki checks the energy data from Wattdepot to see if a floor reached their goal. If the floor did reach their goal, each member of the floor that is participating in the game receives 20 points. The energy goal game provides a link between the energy conservation competition and the point competition.

Figure 4 shows what a player would see when they go to the energy page. The Daily Energy Goal display shows both their current progress and their goal so far for two reasons. First, everyone will be under their actual energy goal for

Prize	Value	Your Tickets	Total Tickets	Current Odds	Change Ticket Allocation
Guitar	\$250.00	68	839	8.1%	+1 -1
Navigation Print	\$52.00	2	71	2.8%	+1 -1
Energy Meter (1)	\$30.00	4	15	26.7%	+1 -1
Energy Meter (2)	\$30.00	3	19	15.8%	+1 -1

**Figure 5: Raffle Game for the Overall Round**

most of the day, so this display would not be very useful. Second, we have noticed that the students in the residence halls use more energy at night rather than during the day. Thus, it is easy to be under for most of the day and then jump over the goal at the very end. Displaying their goal so far provides a pace for players to follow.

### 3.0.3 Raffle Game

The *Raffle Game widget* provides a way to incentivize participation from all individuals, even those who are not in the running for a top prize. For every 25 points a player earns, they receive one virtual raffle ticket. Players can dynamically allocate their tickets to any raffle prizes they are interested in at any time, up to the end of the raffle.

Players can dynamically allocate their tickets to any raffle prizes they are interested in at any time, up to the end of the raffle. Each round of the competition has its own set of raffle prizes and any unused raffle tickets carry over to the next round. Raffle tickets are independent from a player's score; allocating a raffle ticket does not affect their rank. Figure 5 shows an example of the Raffle Game.

### 3.0.4 Social and Referral Bonuses

The *Social and Referral Bonus widgets* provide game mechanics that help encourage participation by providing additional points to players who participate in activities with other players and/or facilitate the entry of new players into an energy challenge.

The social bonus is an administrator option when a task is created in the Smart Grid Game. It awards extra points if the player has done the task with someone else. Examples of tasks with social bonus include attending an event, recording a song related to energy, or measuring a shower water flow rate. When a player submits a response for a task with a social bonus, the player can provide the email address of the person who jointly completed the task. Once the other player completes the task, the social bonus is awarded. Social bonuses are not bi-directional; if the second player doesn't provide the first player's email address, only the first player will get the social bonus.

Players are led through a setup process when logging into Makahiki for the first time. One of the steps in this process

is the referral bonus. If a player was referred by another player in the system, they can use this step to input their email address. Once the new player earns 30 points in the competition, both players are awarded a referral bonus of 10 points. Typically, going through the setup process gives you 25 points, so we wanted to encourage the new player to at least complete one additional task in order to get the referral bonus.

### 3.0.5 Quest Engine

One challenge we faced when designing Makahiki was providing adequate help to the player. The game needed to be intuitive, even if a new player coming to Makahiki has not participated in an energy competition. Unlike many web applications, such as webmail, Kukui Cup players do not know in advance what specific tasks they wish to accomplish. In an effort to provide a player with guidance through Makahiki after the setup process, we implemented the Quest Engine. Quests are used to guide the player through the various workflows of the site, like completing a task, signing up for an event, or allocating a raffle ticket. These quests can be created in the admin interface. They also use a set of predicates to determine unlock and completion conditions. These predicates are: participating in a task or type of task, completed a task or type of task, has a certain number of points (in a round or overall), completed a certain number of tasks in a category or of a given type, awarded a badge, wrote a post on their floor wall, and added a picture to their profile.

### 3.0.6 Real Time Game Analytics

Makahiki includes real-time game analytics that provides quantitative instrumentation to track when, where, and for how long each player accessed each page of the site and the interaction with each game components. Unlike generic web server logs, this feature could track per-player game-specific behaviors in real-time. For example, instrumentation could enable us to determine how players allocated and deallocated tickets to the Raffle game, or whether they watched, paused, or skipped over the video portion of an Activity game. Figure 6 shows an example of the game analytics components.

## 4. EXPERIENCES

As a preliminary assessment of the Makahiki+WattDepot software stack, we designed and implemented an energy challenge called the "Kukui Cup" for over 1,000 first year students living in the residence halls at the University of Hawaii in Fall, 2011. We installed 40 Shark 200-S meters throughout the residence halls, and used the ModBus WattDepot sensors to gather instantaneous power and cumulative energy every 15 seconds. Figure 7 shows a portion of the Go Low page, which contains two widgets (Power Meter and Daily Energy Goal Game) that are based upon WattDepot data.

Response to this initial challenge was very positive. Over 400 students participated, for an adoption rate of approximately 40%. In a survey of those participating students conducted near the end of the challenge, over 90% of them said they would play the game if it were offered next year. 60% of participants said "ease of use" was the thing they liked best

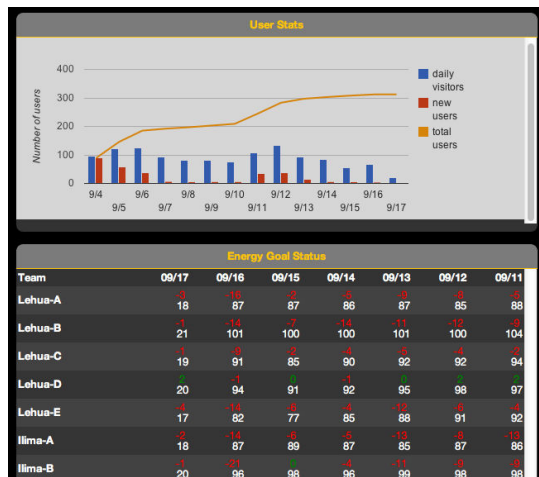


Figure 6: Real time game analytics

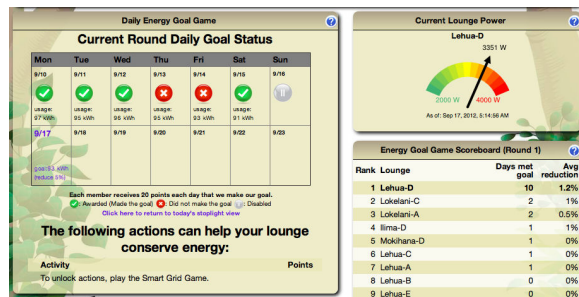


Figure 7: An energy challenge implemented in Makahiki that visualizes WattDepot data

about the website. 40% responded “Nothing” when asked what was confusing about the website, and 32% responded “Nothing” when asked what they would change about the website. The survey did yield insights into what could be improved, including the ability to introduce new games at points during the challenge, to provide better access to other player data, and simplify navigation.

The full version of this paper will go into more detail on our experiences as well as the changes and enhancements we have made to the Makahiki+WattDepot software stack in preparation for three energy challenges by different organizations to be held in Fall, 2012.

## 5. CONCLUSIONS

Need conclusion here.

## 6. ACKNOWLEDGMENTS

Add acknowledgements to sponsors and other KC team members.

## 7. REFERENCES

- [1] S. Darby. The effectiveness of feedback on energy consumption. Technical report, Environmental Change Institute, University of Oxford, 2006.

- [2] S. Deterding, D. Dixon, R. Khaled, and L. Nacke. From game design elements to gamefulness: Defining “gamification”. In *Proceedings of MindTrek*, 2011.
- [3] Control4 EMS 100. [http://news.cnet.com/8301-11128\\_3-20000731-54.html](http://news.cnet.com/8301-11128_3-20000731-54.html).
- [4] A. Faruqui, S. Sergici, and A. Sharif. The impact of informational feedback on energy consumption—a survey of the experimental evidence. *Energy*, 35(4):1598–1608, 2010.
- [5] Open Phasor Data Concentrator (OpenPDC). <http://openpdc.codeplex.com/>.
- [6] Open Source Home Area Network (OSHAN). <http://sourceforge.net/projects/oshan/>.
- [7] J. E. Petersen, V. Shunturov, K. Janda, G. Platt, and K. Weinberger. Dormitory residents reduce electricity consumption when exposed to real-time visual feedback and incentives. *International Journal of Sustainability in Higher Education*, 8(1):16–33, 2007.
- [8] Smart energy 2.0. <http://www.zigbee.org/SmartEnergy>.
- [9] The Energy Detective (TED) Home Energy Meter). <http://www.theenergydetective.com/>.