influxdata

How Sudokrew Used InfluxDB to Build a Battery Management Solution for Blue Planet Energy

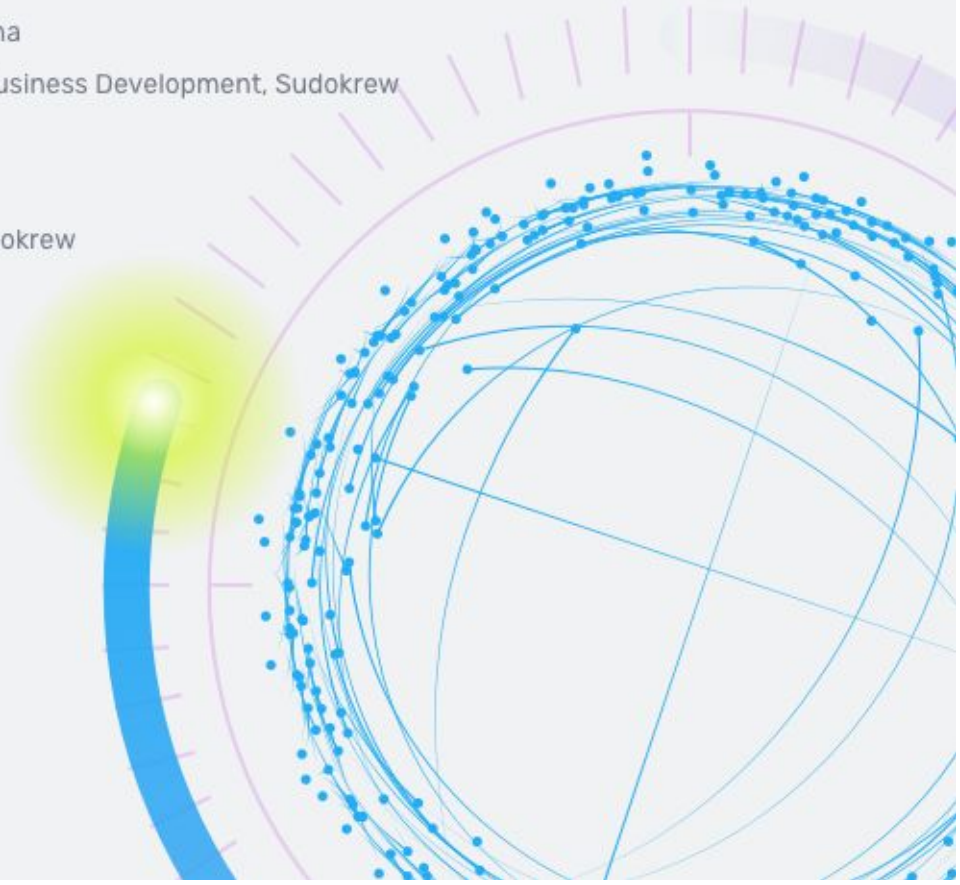
AN INFLUXDATA CASE STUDY

Spencer Toyama

Co-Founder/Business Development, Sudokrew

Tony Gaskell

Developer, Sudokrew



August 2018

Company in brief

Founded in 2013, Sudokrew Solutions is a startup software services and solutions provider operating in Honolulu, Hawaii. Sudokrew's goal is to establish a viable and quality software development resource locally and outside the Hawaiian Islands.

Sudokrew specializes in product development, custom software solutions, as well as legacy system integration and expansion. It is a full stack development shop and software consultancy, building web and mobile apps for businesses and nonprofits.

Sudokrew have worked with Fortune 50 enterprise businesses to bring them up to speed on the modern web (stealth/ninja projects) by augmenting their development teams, and they've helped startups get their initial build for an MVP. Sudokrew love working on meaningful products and developing beautiful user experiences with technology.

Case overview

Sudokrew's customer, Blue Planet Energy, sells an energy storage unit that allows residential and commercial buildings equipped with solar panels to store unused solar energy. They tasked Sudokrew with developing a clean energy storage device management and battery intelligence software solution. The solution required a time series database able to scale with increasing devices and registers, support an HTTP API and basic aggregation methods and to provide dedicated hosting.

Sudokrew was able to leverage InfluxDB to create a solution to gather metrics that help monitor and control the storage units remotely, extending the life of the battery, improving the experience of the user, and offering superior customer service.

Device and battery management solution designed for Blue Planet Energy powered by InfluxDB

> *"The database that we were using was like a black box system. We didn't have much visibility into it. The only thing that we could tell was: is the main process still running?"*
>
> ***Tony Gaskell,** developer*

The business problem

Energy and sustainability are big issues in Hawaii, where a fair number of oil tankers are shipped in everyday, just to keep the lights on. Resiliency in the energy grid, due to threat of natural disaster, is critical. This is why Sudokrew works on energy apps and values working with companies such as RevoluSun, Blue Planet Energy, Blue Planet Foundation, and Elemental Excelerator, who are all presenting solutions critical to the island community's survival.

Hawaii has intermittent energy that floods the grid, and energy storage (which differs from energy generation) is needed to manage how much solar energy is coming into the market or into the grid. Properly managing the power entering the grid is vital for grid stability and grid sustainability and can help prevent frying lines and electronics. The reason Henk Rogers founded Blue Planet Energy was to solve the market's energy storage problem,

The largest barrier in clean energy technology, like solar, is that its sources are intermittent and difficult to predict. Blue Planet Energy is solving the problem of intermittent energy generation by providing a suite of storage solutions, ranging from residential to utility grade lithium iron phosphate batteries. Sudokrew was tasked with building a software suite to manage sales, deploy devices, monitor energy usage, and provide a unified infrastructure to connect every user and vertical of the business.

Sudokrew had no information into what was executing on their servers and often relied on guesswork. Though they had monitoring configured, alerts weren't set to the correct levels for their storage usage. They needed to track information that is vital to maintaining batteries or big storage units, such as battery status and cycle count. Sudokrew first defined end-user needs:

- View how much energy is remaining in the battery
- View of energy consumption by device
- Create an understanding of how consumption and renewable production of energy relates to the energy level of the battery
- Receiving the correct data for the device they have

Then Sudokrew set out to determine the applications and infrastructure necessary to meet those needs.

The technical problem

Sudokrew needed to set up data visualization, but first, they needed a way to receive that data. Blue Planet Energy, as a battery manufacturer, had the forethought to allow data from their inverter to be transmitted, but it was still missing an endpoint to receive and store all of that data as well as consumption data from facilities and individual devices. This meant building endpoints that connected energy hardware to software, along with the infrastructure necessary to manage a never-ending flow of data that could be archived, recalled and indexed to users, dealers and administrators.

Scaling issues with the legacy database

Scaling issues were being reported by all users ranging from admins and dealers not being able to see device status, to end-users facing graphs that were slow-rendering or not showing at all. Further, the issues were inconsistent. Sudokrew was unable to reproduce them, and sometimes the issues would resolve on their own. There could have been issues at the device level, when it would send data to their API, issues with their message queues that transfer data to their database, or issues at the application layer, either retrieving or rendering data.

Overall, the problem was that data was not available. This directed Sudokrew to reconsider their existing database, which failed to provide the data visibility they sought and only showed if the main process is still running and produced error logs. With Sudokrew's existing infrastructure, data needed to be manually migrated if they ran out of space. As a single instance, backups were their only line of defense. Their unanswered questions made them feel helpless to the situation: "What happens if a database doesn't come back up again? What happens if one of the backups that we were relying on failed us?" They couldn't implement features because they were limited by their lack of visibility.

Query performance

Sudokrew needed a solution to meet the different data aggregation requirements of mobile and desktop views. When working with time series data, you can get a lot of data quickly if you're not careful with the levels that you're grouping your data by. A desktop application might be able to handle a day's worth of data at minute aggregation, because a computer is powerful enough to render that into the graph. But on a mobile device, an end-user could be on a cell network, Wi-Fi, or on Edge, so different aggregation requirements are necessary.

The solution

> *"What we got with going with the hosted solution was the dedicated support. That was worth it for us, in the end, because that allowed us to just focus on the application."*
>
> ***Tony Gaskell,** developer*

Why InfluxDB?

Understanding their use case and limitations was key to choosing a time series database, as were the design decisions they had to consider up front and the resulting possibilities.

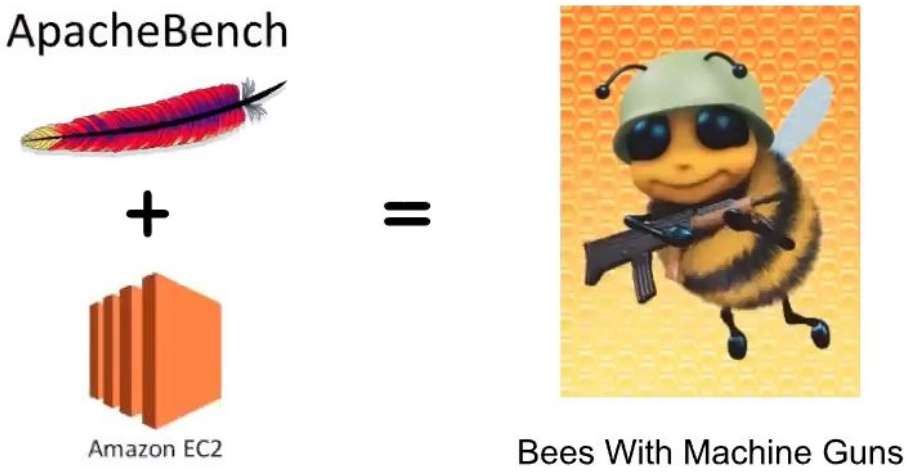
After Googling the top ten time series databases and getting leads, Sudokrew set their database criteria to pare down their search space:

- Must be able to scale as more devices and registers are added to the system
- Needed to be able to support an HTTP API (which is how they were currently interfacing with the existing database)
- Should be able to support basic aggregation methods
- Preferable to have a hosted solution

Sudokrew narrowed down their search to three alternatives. One was Elasticsearch, which wasn't designed as a time series database and didn't seem like the right fit. That narrowed their selection to Prometheus or InfluxDB. Both seemed like good choices, yet InfluxDB had dedicated hosting, which led Sudokrew to choosing it.

Sudokrew architected a benchmarking process, set up a trial version of InfluxDB Cloud, and imported some of the data from their production database into that instance. They used Bees With Machine Guns — an open source tool built on ApacheBench and Amazon EC2 — that allows you to spin up a lot of EC2 instances and DDoS your own servers.

Benchmarking the Databases



This infrastructure met Sudokrew's requirement to simulate a system of distributed devices that could scale a system to any number of devices with a variable amount of registers. They felt that this was going to be an accurate simulation of having many devices that weren't on their local system. Some devices were in the cloud, hitting their servers at a rate that was a lot higher than they expected. InfluxDB also easily fit into the many services that Sudokrew had built for their existing database. Here are some of the performance gains shown by their benchmark test.

Benchmark Results: InfluxDB Passed with Flying Colors

Duration	Aggregation	Requests per second	Average response time
Current	1 second	~ 36x more	~ 44x faster
1 day	15 minutes	~ 8x more	~ 10x faster
1 week	1 hour	~ 7.6x more	~ 7.6x faster
1 month	1 hour	~ 2.3x more	~ 2.3x faster

Total: 1000 requests

Concurrency: 50 requests

After deploying InfluxDB, Sudokrew was able to focus on the application and system. They were able to convert data from an XML payload that they were getting from eGauge devices into InfluxDB schema.

Sudokrew had collaborated with eGauge Systems (a manufacturer of utility-grade energy monitoring systems that can receive production data of solar systems as well as energy consumption data of common devices) to integrate eGauge devices into the Blue Planet Energy application. eGauge devices have a password-protected built-in dashboard and built-in internet connection. Sudokrew have a data pipeline set up with eGauge. Once eGauge is in their system, then they can assign it to people within that system.

Translating XML Into an InfluxDB Schema

XML payload

```
<?xml version="1.0" encoding="UTF-8" ?>
<data serial="0x00000000">
  <ts>1284607004</ts>
  <r t="P" n="Grid"><v>5196771697</v></r>
  <r t="P" n="Solar"><v>21308130148</v></r>
</data>
```

Binary data payload

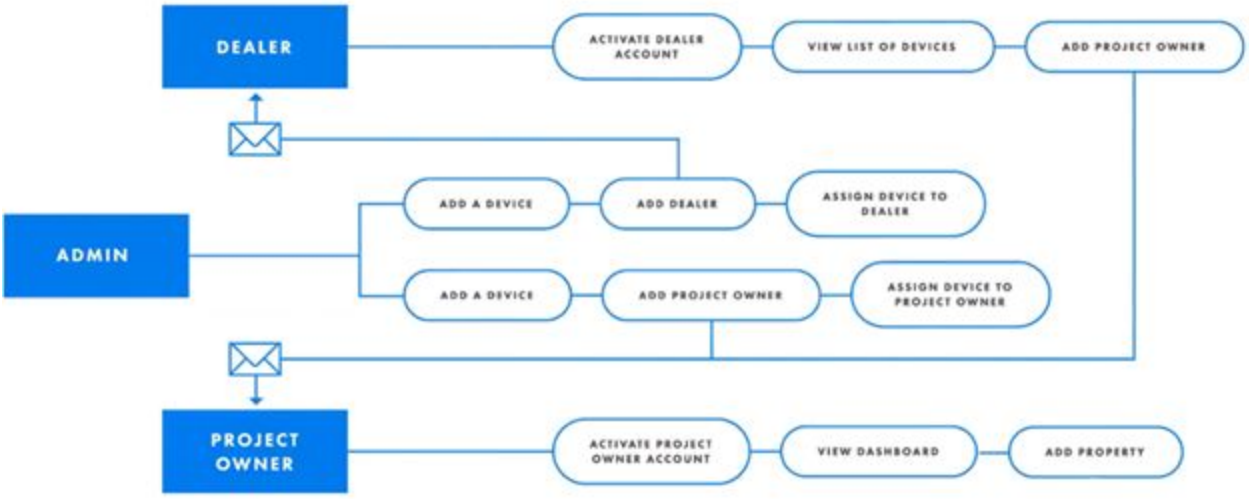
```
devices,register=Grid,unit=P,serial=0x00000000
value=5196771697 1284607004000000000
devices,register=Solar,unit=P,serial=0x00000000
value=21308130148 1284607004000000000
```

By reading the InfluxData documentation, Sudokrew learned how to set up their tags to make queries perform well. To keep data under control and avoid running out of storage space, they asked themselves while configuring the database: "How much data are we expecting to come in? How long should we hold onto that data? What format should that be? Do we want to keep averages, or do we want to keep a running total?" They could now answer these questions because they gained control over how the data was handled. They looked into retention policies and decided to roll their seconds data into minute aggregations that they could hold onto.

The monitoring software

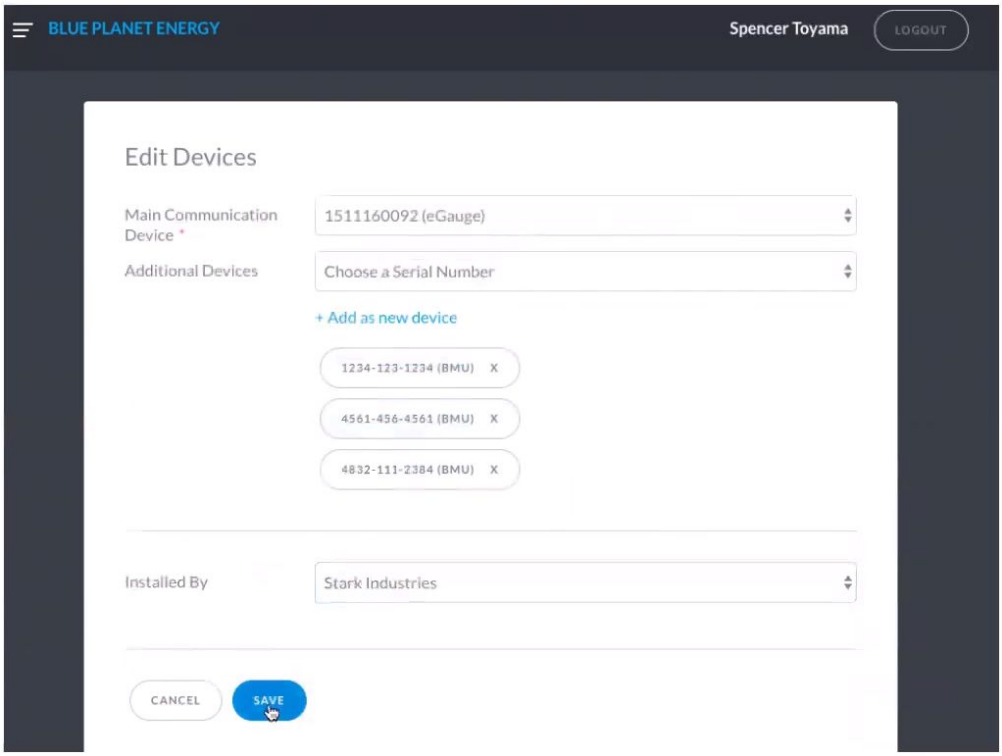
The monitoring software that Sudokrew built for Blue Planet Energy is broad in scope. It has an admin portal that can commission devices and dealers and assign those devices to dealers. Dealers have their own portal where they can activate their account, view all devices that have been assigned to them and then add project owners (home or office users of the battery). Project owners have access to the dashboard and can have property on it.

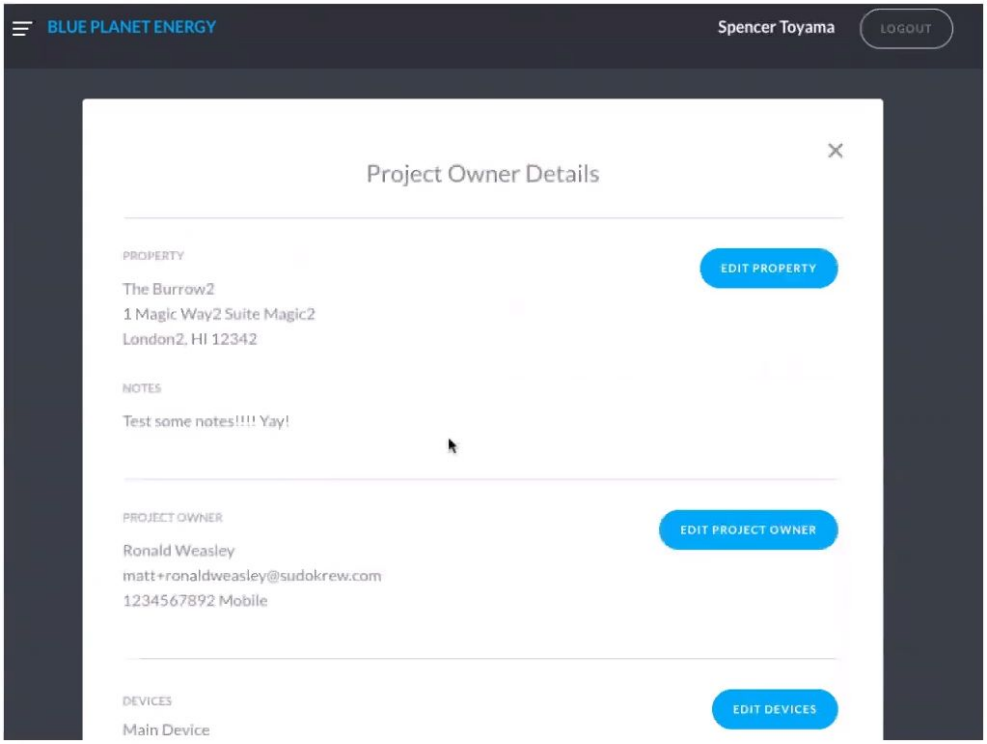
Admin Dashboard to Commission Users and Devices



Admins have the ability to add users and devices to the platform.

How the Admin Dashboard Works





InfluxDB enabled Sudokrew to store production and consumption data and amounts stored over time, as shown below.

Mobile Monitoring Graphs

The above graphs, respectively, show the amount of energy stored in the battery; battery consumption over a 24-hour period; and battery storage over time (which gives users different time increments to query from).

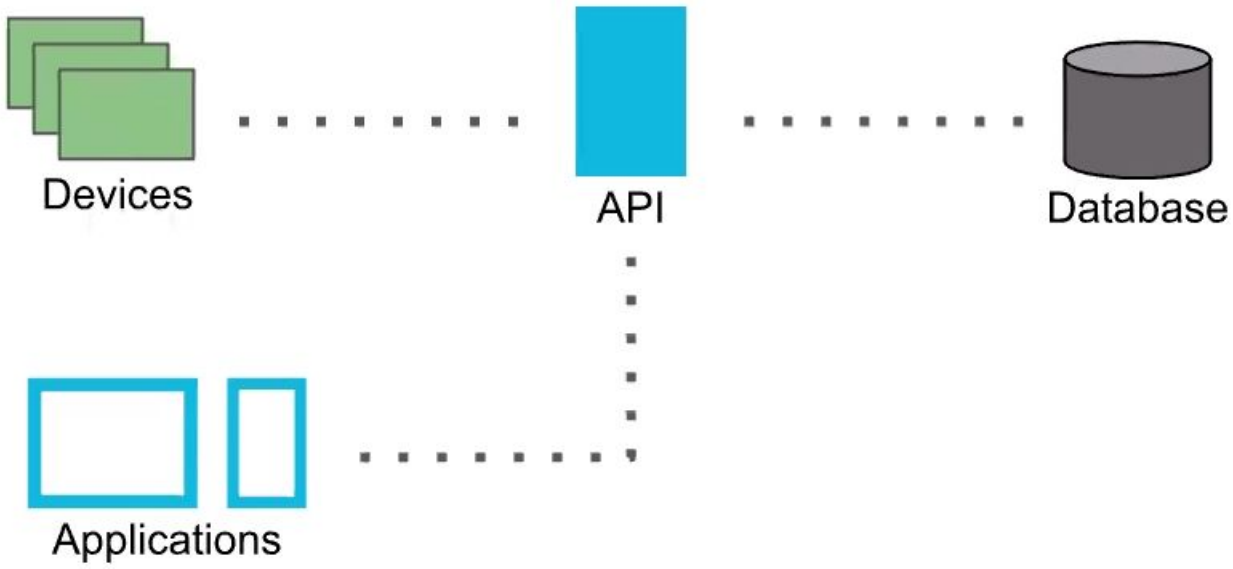
The above graphs show usage against production (enabling homeowners seeking a Net Zero lifestyle to see how their overall production of solar energy compares); a 24-hour view of usage against production; and usage by device (by registers, which are individually monitored parts of the breaker box). So customers could, for example, measure washing machine and dryer time, and see the usage, provided they have the right sensors hooked up.

Technical architecture

> *"It was clear to us that Influx was definitely going to be able to support our use case, and any growth, or unexpected growth, even in the future...We were able to focus again on the application and the system, being able to convert data from an XML payload that we were getting from eGauge devices into something that Influx could understand."*

Tony Gaskell, *developer*

High-Level Overview of Sudokrew Architecture



The dataflow is as follows:

- Sudokrew use eGauge devices (the sensors on the batteries) as the data logger. The eGauge device reports on the battery parameters such as state of charge and cycle count. The eGauge devices also monitor other aspects, such as kitchen usage, that are not directly monitored by the battery and have separate sensors for those.
- All the data that comes in is at the second aggregation level.
- Sudokrew pull the data to a set of APIs and into InfluxDB, then serve it up into their applications.
- For mobile devices, it was more performant for them to query the data by 30 minutes, to reduce the magnitude of the number of points of data coming back so that the graphs could render quicker.
- The data pulled into InfluxDB has tags such as register, unit, and serial. Since the configured devices don't have set register names, Sudokrew treat each register as an individual series with a single value. Because they are not sure what values are going to end up on a production device, they split apart the entire payload into separate series of data, and from there query and aggregate it back together however they need it for their application.

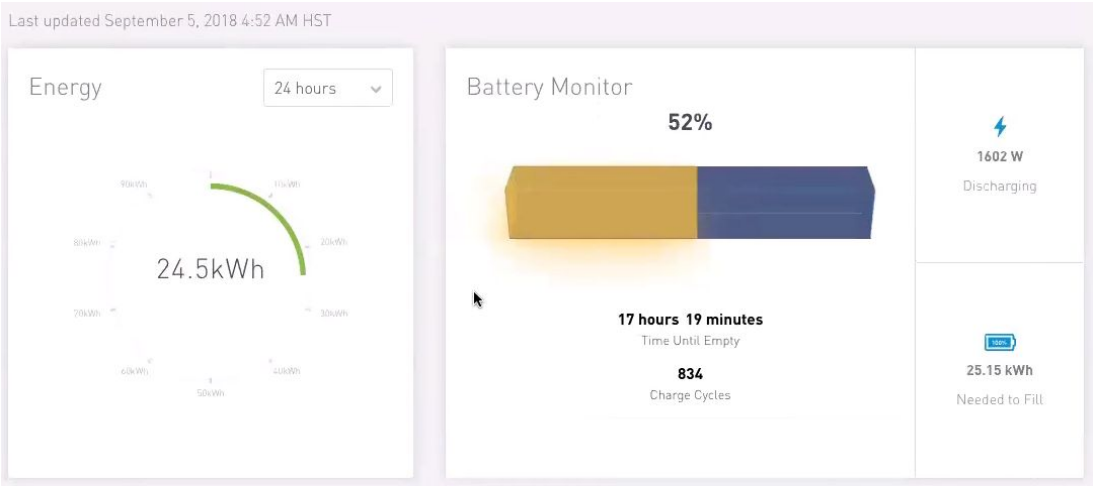
- Sudokrew has primarily used Chronograf for the aggregated view (that Blue Planet Energy see) of all the devices out there, which came with InfluxDB Cloud. They created templates for Blue Planet Energy whereby to find a device, they could just enter in its serial number, and also customize the views they wanted to see.

Results

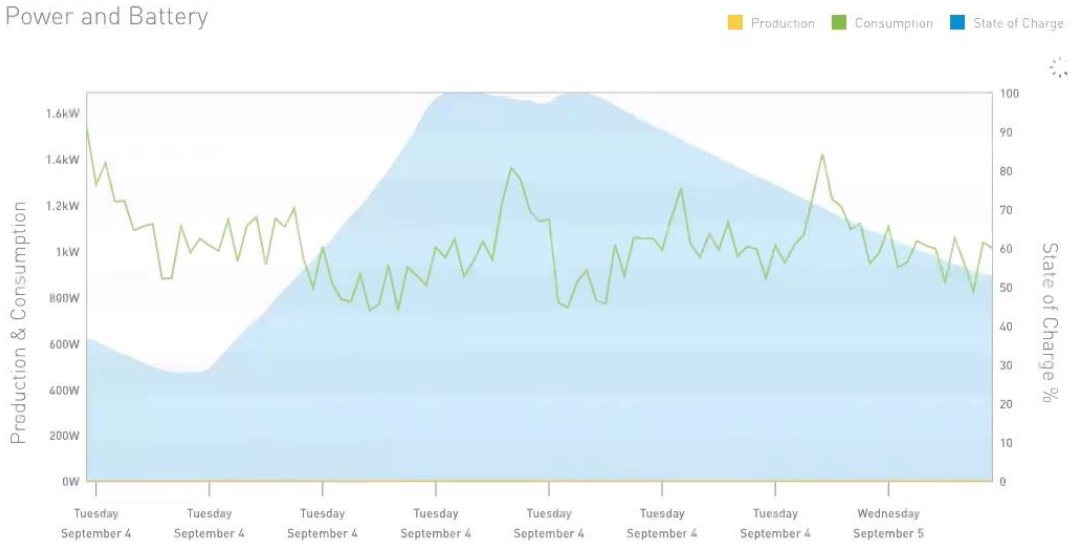
> *"From here, we have a lot of places that we can go forward with it [InfluxDB]. There are a lot of opportunities that are available to us now that we have control over our data. That was probably one of the best things that we got out of this so far."*
>
> ***Tony Gaskell,** developer*

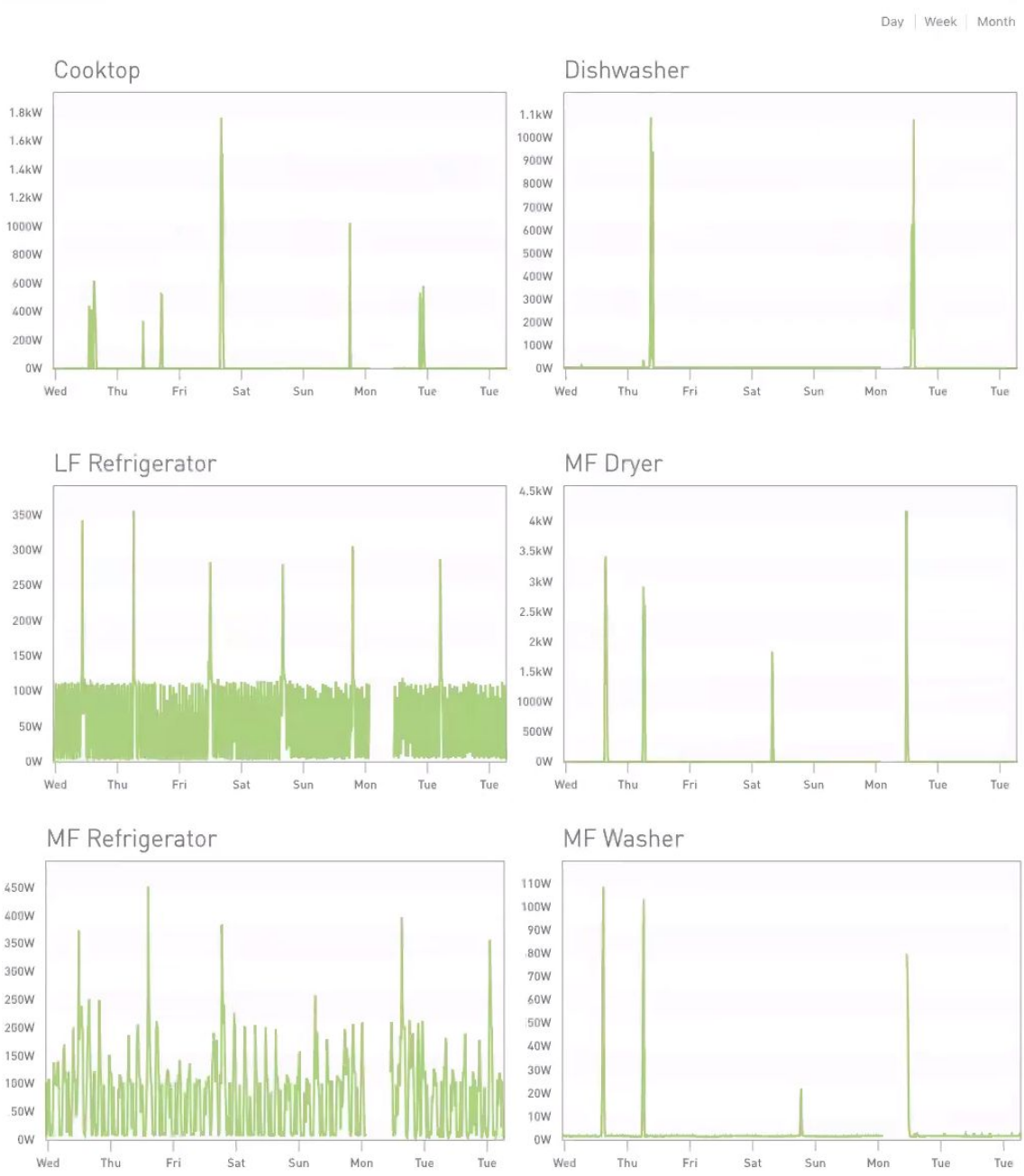
Gaining visibility into their servers gave Sudokrew the layer of transparency they sought. The open source nature of InfluxDB gave them visibility into the issues raised, the responses, and the developers responding to issues raised. With the warranty values coming in, Sudokrew could view the average cycle count across all devices, which they couldn't do before.

Sample Dashboards

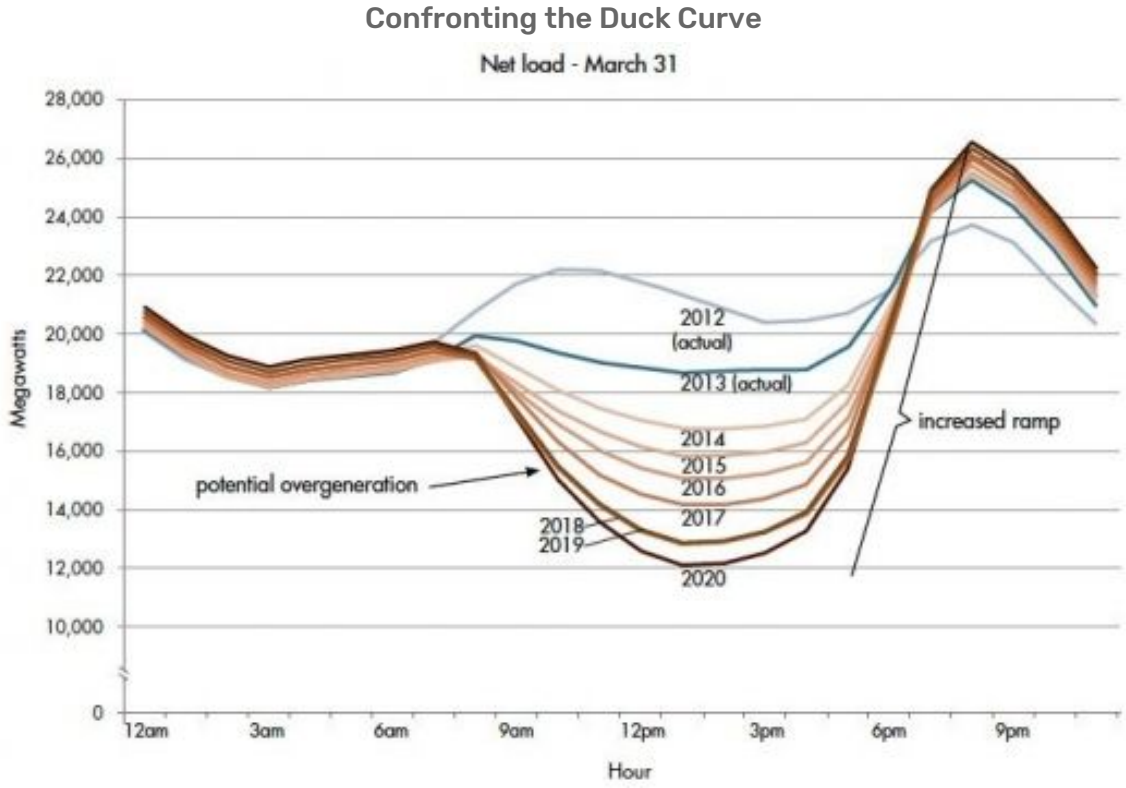



Power and Battery
Production
Consumption
State of Charge
Day
Week
Month
10kW
9kW
8kW
7kW
6kW
5kW
4kW
3kW
2kW
1kW
0W
Production & Consumption
100
90
80
70
60
50
40
30
20
10
0
State of Charge %
Sunday
August 5
Friday
August 10
Wednesday
August 15
Monday
August 20
Saturday
August 25
Thursday
August 30
Tuesday
September 4


Power and Battery
Production
Consumption
State of Charge
1.6kW
1.4kW
1.2kW
1kW
800W
600W
400W
200W
0W
Production & Consumption
100
90
80
70
60
50
40
30
20
10
0
State of Charge %
Tuesday
September 4
Tuesday
September 4
Tuesday
September 4
Tuesday
September 4
Tuesday
September 4
Tuesday
September 4
Tuesday
September 4
Wednesday
September 5



The general trends Sudokrew see are people waking up in the morning, turning on all their appliances, leaving their house for work, seeing the batteries charge up over the course of the day while the sun is out, and then slowly declining overnight, with the energy being pulled from the battery instead of from the grid. In the energy industry, this is called the duck curve and shows the difference in electricity demand and the amount of available solar energy throughout the day. Sudokrew, through their apps, are now able to see the individuals that make up the duck curve and aggregate values.



Sudokrew built their monitoring apps to create value for all stakeholders as they realized that transparency is essential for building customer relationships. For Blue Planet Energy, these monitoring apps presented an opportunity to create a more harmonious relationship with their customers by providing them with the same data that they see. So Blue Planet Energy can see cycle count, battery output, and other parameters related to the warranty, to battery health and life cycle. Having data at the center of that relationship, where they can talk to the customer openly about data, creates a more transparent relationship. Visibility into usage patterns and people's energy habits in turn helps project owners make decisions.

Sudokrew valued the dedicated support that came with InfluxDB Cloud, as it allowed them to focus on the application rather than on their infrastructure. That support also helped them resolve issues they faced in their design and schema. They found that the pressure to work on features that benefit their customers outweighs dealing with setting up and managing their own hosting.

Using InfluxDB as their time series database, Sudokrew is fulfilling their mission of “building world-class software for meaningful projects”.

About InfluxData

InfluxData is the creator of InfluxDB, the open source time series database. Our technology is purpose-built to handle the massive volumes of time-stamped data produced by IoT devices, applications, networks, containers and computers. We are on a mission to help developers and organizations, such as Cisco, IBM, PayPal, and Tesla, store and analyze real-time data, empowering them to build transformative monitoring, analytics, and IoT applications quicker and to scale. InfluxData is headquartered in San Francisco with a workforce distributed throughout the U.S. and across Europe.

Learn more.

InfluxDB documentation, downloads & guides

Download InfluxDB
Get documentation
Additional case studies
Join the InfluxDB community

799 Market Street
San Francisco, CA 94103
(415) 295-1901
www.InfluxData.com
Twitter: @InfluxDB
Facebook: @InfluxDB