SMS Alerts from Unplugged Electric Cars

We recently purchased an electric car (~90 miles, e-golf) to get free access to carpool lanes and noise free driving experience. Unfortunately, the advertised range of electric cars can be limiting when commute involves uneven terrains. Soon, it became apparent that our electric car must be charged every night, or we are at risk being stranded.

When we purchased the car, we thought that the electric cars will be equipped with some sms type alert setup to alert owners when they are not being charged for overnight charging (e.g., in our case the low tariff and off peak hours are between 11pm-7am). But, after we scratched past all the on-board software and smartphone apps that came with the car there were none of those options. It was hard to believe that most electric cars will not alert you when you forget to plug it in for overnight charging. At the most they will provide you charging status, which is meaningless.

So how do we address it? In my view an ideal solution should send us an SMS alert when car is not being charged in night. Below is one simple solution I came out that we've been using for close to two years. (In case you want to try it out please do read the disclaimer Disclaimer.)

The solution involves equipping the smart utility power meter with an ability to send sms alerts when power consumption level is below the threshold of an plugged in electric car. This system has an upfront cost of $100 and subsequent cost less than a $1 per month.

Many smart power meters (e.g., Fig. 1) have provision to interface with smart monitoring modules such as rainforest eagle (e.g., Fig. 2).

Fig. 1: PG&E smart power meter

It's Simple: Just turn it on. EAGLE automatically connects wirelessly to your smart meter. An easy web page allows you to direct your data where you want.

It's Secure: Encrypted wireless link. Only your registered device can talk to your meter.

[View Back]

It's Flexible: You can view your energy consumption in many ways: point your web browser to rainforestcloud.com; use our convenient iPhone and Android apps; connect to our third party Cloud Partners
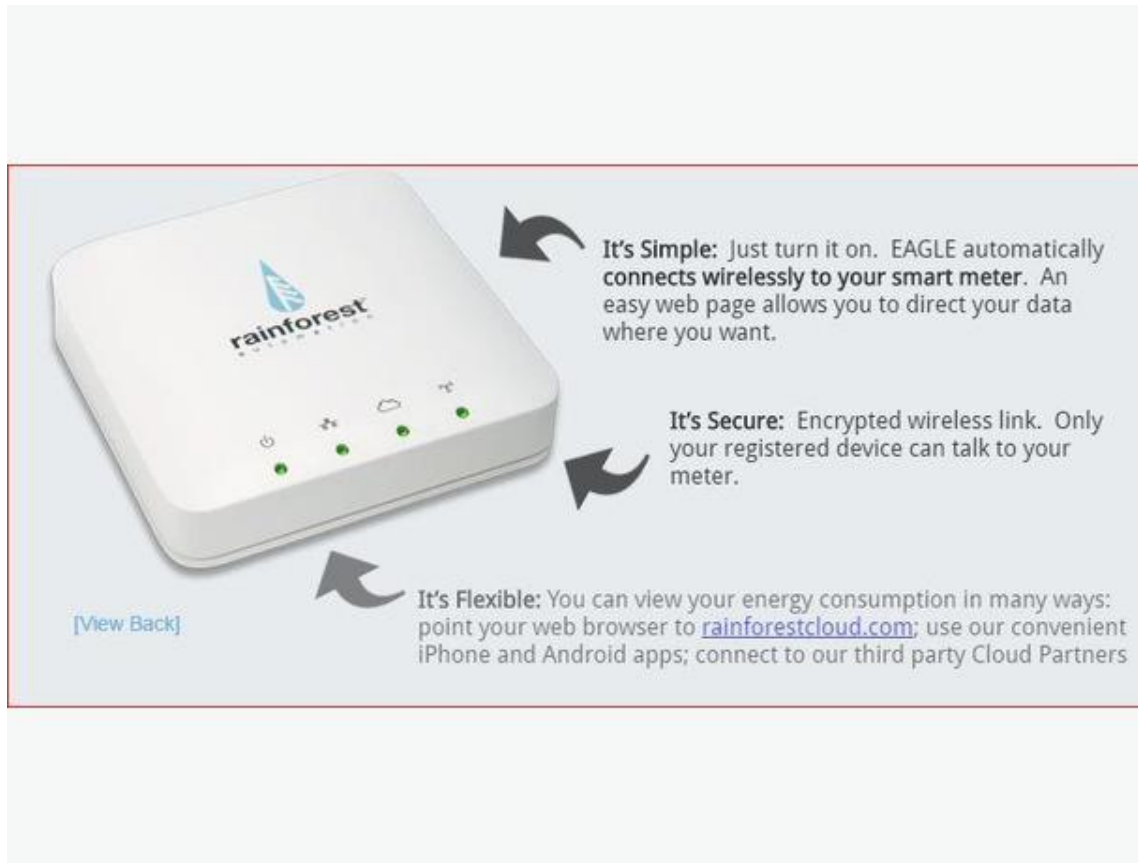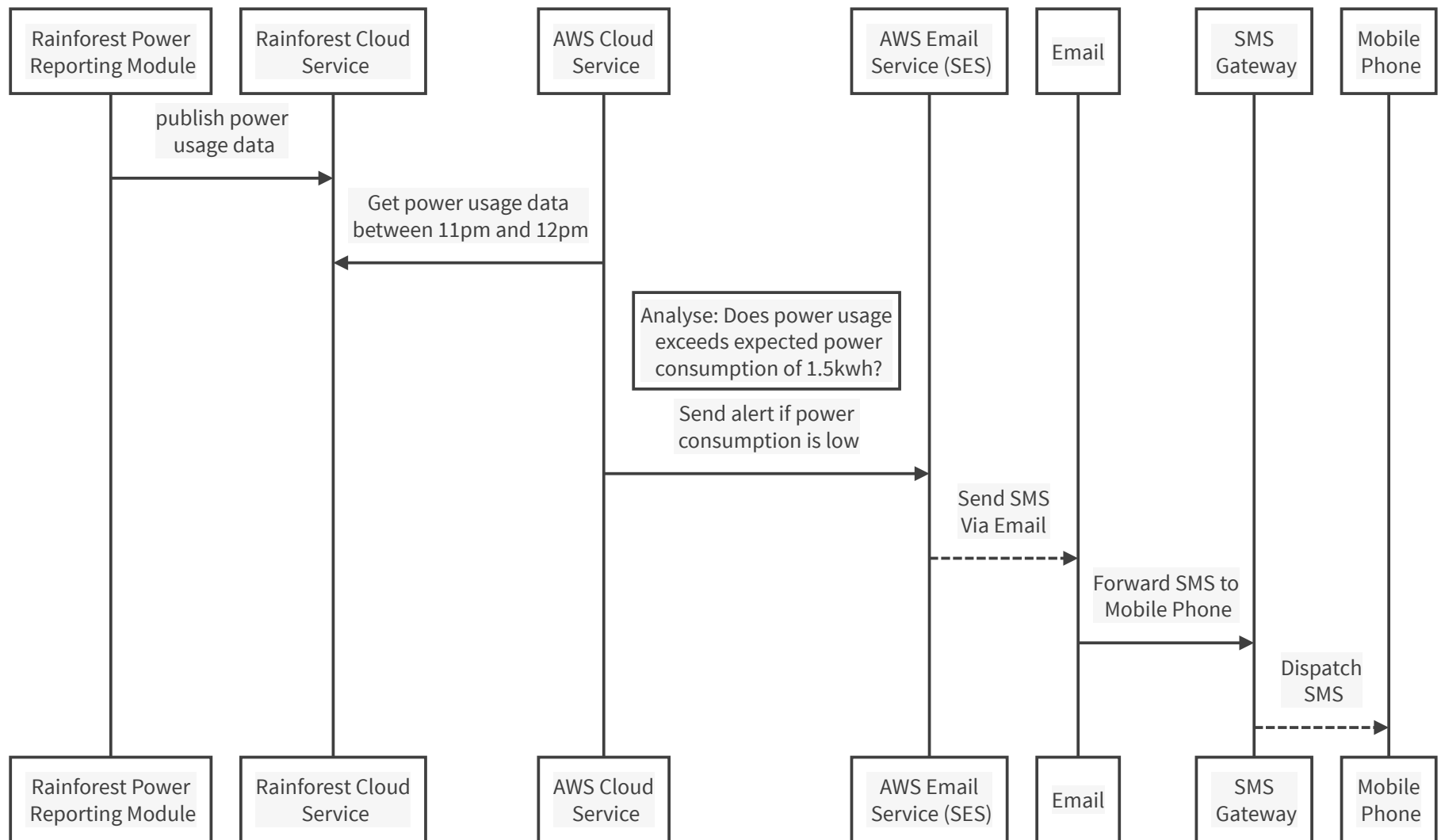
Fig. 2: Rainforest power reporting module

The monitoring module shown in Fig. 2 has an API to get power consumption level in real-time. So, all you need is a way to poll this API (after 11pm) and send SMS alert when power consumption level is below the expected level –power level after an electric car is plugged. Below is a solution along these lines.

Solution

These instructions can be followed by anyone who is interested. However, if you want to dig deeper you'll need knowledge of unix/macbook shell and python.

Sequence Diagram

Technologies

Here is a list of various technologies used in the solution. The detail explanation on how they are used is provided in the next section.

- PG&E smart meter (Fig. 1)

- Rainforest power monitoring module (Fig. 2, [Rainforest eagle ethernet zigbee](#))
- AT&T mobile plan (to send MMS via an MMS email address <mobile_phone_no>@mms.att.net
- AWS STS gateway to send emails
- AWS lambda function (in python) to invoke the power usage API and send alert to an email account.
- Am email account (gmail) to receive email from AWS STS and forward it to <mobile_phone_no>@mms.att.net
- AWS cron trigger to invoke the AWS lambda function between 11pm-12pm

Implementation Steps

Step 1

Buy and configure rainforest eagle (Ref. 2) to work with PG&E smart meter. The configuration instructions come with the module and are straightforward. The module requires internet and for that it needs one unused RJ45 port on back of your WIFI router (all WIFI routers have few of these spare ports). Once you are done with its configuration note down the *cloud_id*, *username*, and *password* (these are needed in Step 5 to connect to the power module via [rainforest cloud service](#)).

Step 2

Download ealert zip file which will be used in Step 5 to generate alerts using AWS lambda function and rainforest eagle API
- Download [ealerts.zip](#) and note the location of the file
- [optional] confirm md5sum for above file is "134f298b83d5aa3627e9fdc6c09c1683"
- [optional] ealerts.zip is generated from the source code located at [ealerts](#) (run build.sh to generate your own target/ealerts.zip file on Ubuntu 14.04LTS)

Step 3

Signup for Amazon AWS for free if you haven't already [AWS Free](#)

Step 4

Configure an AWS SES account to send out emails

- On AWS console pick "Services" followed by "SES" under "Messaging"
- Login to AWS SES and pick a region from upper right dropbox. I use Oregon as that provides the best value. Here is a direct link to create SES account in AWS SES Oregon. In any AWS region, you can find "SES" by picking "Services" (upper left tab) followed by "SES" under "Messaging" of AWS console.
- Click on "Go to Identity Management" shown in the main panel on lower left side Oregon IM
- Click on the SMTP settings shown as the 4th entry on left panel
- Note down the SMTP server name and port. For example for Oregon data center these are "email-smtp.us-west-2.amazonaws.com" and 587 respectively.
- Create "my smtp credentials" and on next page click "Show User SMTP Security Credentials". Save/copy the *username* and *password* to a safe location.
- In Step 7, you'll need to pick an email address which will be used to forward SMS email alerts from AWS SES service to your mobile phone. For example, I use a gmail for this purpose. You can also create a new gmail address and use it for ealerts purposes only. But, before this email can be used for sending alerts, you'll need to verify this email via "Verify a new Email Address" under "Email Addresses" link under "Identity Management" (left panel).

Step 5

Setup Credentials

- Click on "Services" tab shown on the left side of top bar of AWS Console (from Step 4 above)
- Click on "IAM" under the category "Security, Identity, & Compliance" (main panel)
- Click on "Users" and on add user
- Create a user with any name that you may like, e.g., ealerts
- Select "Programmatic Access" under select "Select AWS access type"
- Click on "Next Permissions" and select "Attach existing policies directly"
- Add following policies:
    - AmazonDynamoDBFullAccess

- Click "Next:Review" and then "Create User"
- Save/Copy "Access key ID" and "Secret access key" for later

Step 6

Setup Lambda function
- Click on "Services" tab shown on the left side of top bar or click on this link Lambda function
- Click on "Lambda" under the category "Compute" (main panel)
- Click on "Get Started Now" followed by "Create a Lambda function"
- Click on "Blank function" (the very first blueprint function in main panel)
- Click "Next" on "Configure Triggers" Page (will come back to it later in Step x)
- On "Configure Function" page
- use "ealerts" as Name
- use "send alerts" as Description
- select "Python 2.7" as Runtime
- On "Code Entry Type" drop down select "Upload a .ZIP file"
- Use "Function Package" to upload the zip file from Step 2
- Change Handler name to
ealerts.main_handler
- For "Role" pick "Create new role from templates(s)"
- For "Role name" use "simple_microservice_permissions"
- For "Policy template" use "Simple microservice permissions"
- No changes in "Advanced Settings"
- Click "Next"
- Click "Create function" button near lower right corner
- You should see a congratulation message

Step 7

Prepare event template. Fill in the appropriate details in the following json

```
{
"alert_type": "ElectricCarChargeAlert",
"eagle_username": "from Step 1",
```

```
  "eagle_password": "from Step 1",
  "eagle_cloud_id": "from Step 1",
  "smtp_server": "email-smtp.us-west-2.amazonaws.com",
  "smtp_port": 587,
  "smtp_email_from": "youremailhere@gmail.com",
  "smtp_email_to_list": ["youremailhere@gmail.com"],
  "aws_ses_access_key_id": "from Step 4",
  "aws_ses_secret_access_key": "from Step 4",
  "aws_db_region": "us-west-2",
  "aws_db_access_key_id": "from Step 5",
  "aws_db_secret_access_key": "from Step 5"
}
```

Step 8

Setup for a Polling Trigger to Monitor Power Consumption
- Click on "Services" tab shown on the left side of top bar
- Click on "Cloud Watch" under the category "Management Tools" (main panel)
- Click "Rules" under "Events" Category on left panel
- Add Rule
- Select "Schedule" and select "Cron Expression"
- Use `15-59/5 5 ? * * *` to trigger events between 10:15pm PST and 10:59pm PST. The cron expression is in GMT, so adjust it for your own time zone and daylight savings time if there is any.
- Under "Target" click on "add target" and for "Function" select lambda function name created in Step 6, which is "ealerts"
- Click on "Configure Input" and select "Constant (JSON text)"
- In the indicated textbox copy and paste the entire JSON from Step 7 (including curly braces)
- Click on "Configuration Details" button on lower right corner on the main panel
- Use "elerts_electric_car_trigger" both as name and as well as description and click "Create rule"

Step 9

Now alerts should start coming to the email address that you provided in Step 7 and earlier confirmed in Step 4. Next thing is to create a forwarding rule in your email settings that can forward the alerts as MMS to your mobile phone. If you have AT&T you'll need to first enable you mobile phone MMS (not SMS) account mymessages.wireless.att.com to be able to receive emails from your email address. (I did try sending MMS directly from SMS, but that didn't work as I ran into issues with the AT&T mobile service, which falls apart when confirming the email address as received by mobile phone.)

Step 10

The ealerts module is also capable of sending traffic alerts using Google Traffic APIs between two addresses (for example between you home and office and vice versa), which I'll let you figure out by looking at the templates provided @ 14

For any questions reach out to me @ https://www.linkedin.com/in/jhooda/