Power Factors – SEEDS Gateway to PI Integration

High-Level Technical Overview

Andrew Scott, ascott02@gmail.com, August 28th, 2016

Audience:

- Andy Sugiarto and the technical team at Power Factors
- Kate Schneider and the technical team at ZDCA

High-Level Goals:

- Integrate newly acquired assets (approximately 250 SEEDS Gateways)
- Establish secure communications to Power Factors network
- Aggregate data into Power Factors' OSI Soft, PI System

Background:

Power Factors recently acquired several SunEdison locations which are still running SEEDS Gateway data acquisition servers. The SEEDS Gateways are configured on a site-by-site basis relative to the network and hardware installed (environmental kits, inverters, meters, trackers, etc.) at each location. The SEEDS Gateway software polls the data points off of all devices once a minute, by default. The GW then sends the results of the poll to the enterprise, once a minute, as an XML message. Power Factors would like to explore possible solutions to collect the data from the SEEDS Gateway servers into Power Factors' PI System. Two possible ways are explored here.

Overview of this Document:

- Current Architecture
- Two Proposed Ideas:
  - Use the parallel data feed feature
  - Put the data on a different data path

Author/Revisions:

Initial Draft	Andrew Scott	August 26th, 2016
Second Edit	Andrew Scott	August 28th, 2016

Current Architecture Diagram:

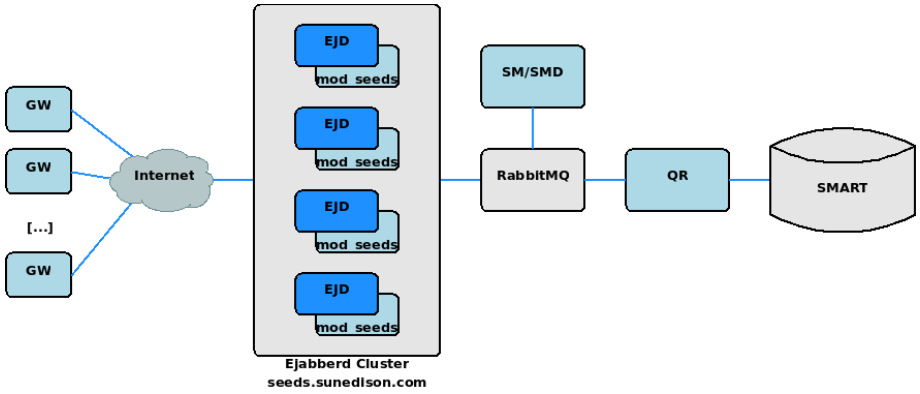

Figure 1: Gateways communicate with SunEdison Ejabberd cluster over encrypted XMPP.

Gateway Hardware:

- Advantech Uno Embedded Intel
- 1.6 Ghz Atom Processor (1 Core)
- 512 Megs or 1 Gig of RAM
- 2 or 4 Gig Compact Flash

Gateway Software:

- Operating System is Debian 6, Linux filesystem, fileutils, proc, iptables, shell, perl, and python
- Custom C Software – "SEEDS" (SunEdison Environmental Data and Statistics)
  - Built against Loudmouth XMPP library for Jabber communication with enterprise
  - Built against libc6 (stdlibc) available with Debian 6
  - Several other libraries for handling serial communication protocols (Modbus, DNP3, etc.)
  - Data is logged locally to a SQLlite, embedded DB (file locked)
- Apache with Python Interface (Small Local Admin Interface for SEEDS Devices)
- Gateways are already configured for local network and local devices.
  - Imagine at each GW endpoint, there is an entire LAN full of smart devices that the Gateway Custom software is polling once a minute.
  - These configurations are already set up and will not need to be modified.

Gateway Network:

- Each location has different LAN configuration with some defaults*
- Each location connects to the Internet by Cell Modem (with unique pub IP) or DSL

  * Note: Most gateways configured eth0: 192.168.13.2 and cell modem internal: 192.168.13.1

Gateway security and Authentication:

- Iptables firewall
- The gateways are configured to only allow access on ports 22 and 80
  - Login locally with a admin or root accounts*
  - Login over ssh with a admin or root specific ssh keys*
  - Login with administrator user for Web administration access
    - Allows polling of devices
    - Testing configuration

  * Note: Some locations have an additional vendor specific user.

Enterprise Software:

- Ejabberd cluster with custom plugin: mod_seeds
  - Manages authentication (each gateway has unique JID and password with Ejabberd)
  - Converts XML/XMPP messages to JSON and publishes on RabbitMQ
- Several in-house tools pick up messages from queue then
  - Process/redistribute among other queues/processes in the stack
  - Message is eventually logged to SMART, Ticketing and/or Seeds-Manager DBs
- Front-ends, such as SunEdison Connect Display Data from DB
- All Gateways are configured to talk to seeds.sunedison.com on 5222 using XMPP/SSL
- All Gateways are configured to use seeds.sunedions.com for NTP

Parallel Data Feed Feature:

One possible solution for rapid integration of legacy SEEDS Gateways into Power Factors' network and PI system is to leverage the parallel data feed feature in the SEEDS Gateway software.

The original data feed in the SEEDS Gateway stack had the Gateways pushing data, once a minute, to a Jabber Server over XMPP/SSL connection. The parallel data feed extended that architecture to allow for an additional Jabber Server to be configured: every minute, it would push data to both Jabber Servers.

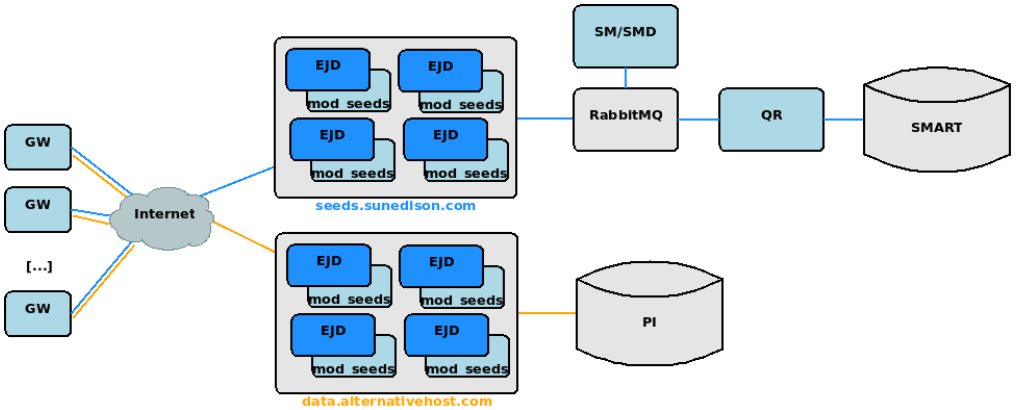

Figure 2: SEEDS Gateways can push data to two Jabber servers at once, using a feature that is already present in the software.

Necessary Steps to Implement:

- Setup secondary Jabber server/cluster
  - Must be able to handle load and scale (special tuning)
  - Import Roster Information from current system
- Develop a way to write-out data to flat file from Jabber Server
- Network Jabber server host filesystem with PI Server
- Configure PI-UFL to upload data into PI
- Configure each gateway* to use a secondary Jabber server – Must know IP addresses and credentials

* Note: The SEEDS Gateway parallel data feed software is part of a separate Debian package. This solution assumes that apt.sunedison.com is still functioning or that we have a copy of the .deb file.

Thoughts/Discussion:

- Medium development in Enterprise (mostly in making a Jabber server scale and export to flatfile)
- Minor configuration change on Gateway
- Still connected to SunEdison network (impacted by outages)
- Must replicate some of that enterprise stack
- Data still flows to SunEdison too
- Needs Jabber server/cluster system administrator
- Further administration is done through the command line and monitoring Jabber server

Put Data on Different Data Path:

Another idea is to eliminate the XMPP/SSL connection entirely, sever ties with the SunEdison network, and put the data on entirely different data path. This idea was successfully prototyped while at SunEdison.

A small, lightweight client was developed and installed on the gateway that intercepted the data for the enterprise and sent it to a highly available, load balanced Web service. The Web service, in the prototype, wrote the data to a network mounted flatfile, where the PI UFL uploader periodically scanned and imported any found files. The files were written in CSV format.

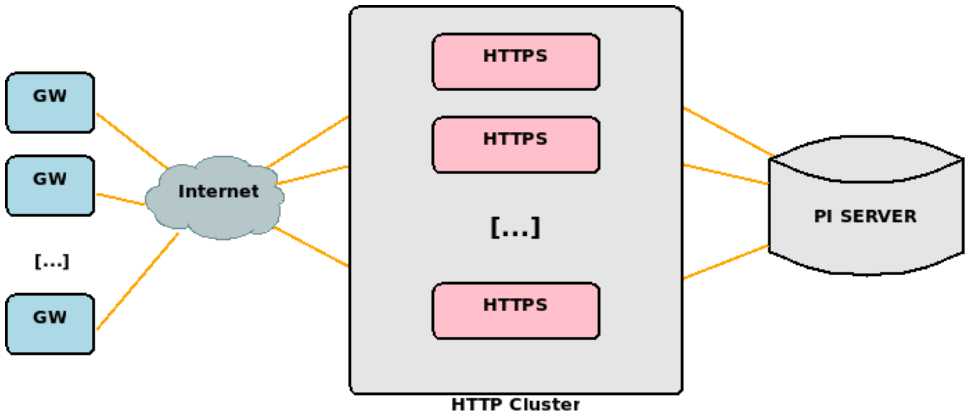

Figure 3: It is possible with a small development effort to put the data on an completely different, and much more scalable data path.

Necessary Steps to Implement:

- Write small client (in C, Perl, or Python) that runs on the gateway, captures data and sends to Web Service
- Provision N lightweight Web Service hosts (HTTPS hosts), that capture messages and writes to filesystem in CSV format
- Network HTTPS host server filesystems with PI Server
- Configure PI-UFL to upload data into PI
- Configure each gateway – Must know IP addresses and credentials
  - Network copy the install script to each gateway and run the install script
- Further administration is done through command line and PI system

Thoughts/Discussion:

- Small development on gateway
- Small development on enterprise
- Much more scalable and fault tolerant
- Does not need dedicated administrator
- Note impacted by SunEdison's network performance