

Emoncms Developer

OpenEnergyMonitor project was founded out of a desire for open-source tools to help us relate to our use of energy, our energy systems and the challenge of sustainable energy. We want to create an open system that anyone is free to build, customise and develop upon.

Emoncms is an open-source software platform for logging, visualising and analysing energy, temperature and other environmental time-series data developed as part of the OpenEnergyMonitor project.

This is a role to help with the development and running of emoncms.org including: scalability and the implementation of billing (see appended section for detail). Ideally we are looking for someone with existing experience of the OpenEnergyMonitor system, community and emoncms.

Salary: £23,000 full time or pro-rata *

Location: Ideally we are looking for someone to join us at the Megni / OpenEnergyMonitor office near Bangor, North Wales but we will also consider a degree of remote working coupled with a number of days per month in our office.



Emoncms has grown into a versatile and powerful software application, we are proud of our achievements so far, however there is more work to do to take Emoncms to the next level and for emoncms.org to continue to be a reliable and useful service.

We are looking for a PHP and JavaScript developer with an enthusiasm for working on open-source sustainable technology and a willingness to engage with an active open-source community.

Essential

- Knowledge and experience of PHP and JavaScript.
- Experience of building web-applications.
- Experience of Linux and web server administration
- Web-application security awareness

Highly Desirable

- Experience contributing and managing distributed open-source software projects
- Experience scaling and administering medium sized web applications at server and application level.

Megni and OpenEnergyMonitor

Megni energy monitoring is run by Glyn Hudson and Trystan Lea. We are based in Snowdonia, North Wales. We are both also founders and core developers of the OpenEnergyMonitor project.

We are a small business, there are currently 3 of us full time and 3 part time. The OpenEnergyMonitor shop is based in Eryri (Snowdonia) North Wales. Our hardware is manufactured and assembled locally. In our spare time we are keen climbers, walkers and mountain bikers. See the OpenEnergyMonitor about page for a better idea of who we are: <https://openenergymonitor.org/?q=about>

To apply

If you're interested in this role we would love to hear from you. Please send an email to trystan.lea@openenergymonitor.org. Tell us a bit about yourself, your experience, your interest in the field and the OpenEnergyMonitor project. Please provide any relevant example links to previous work that you have been responsible for (e.g: github repositories, blog, personal software development website).

We look forward to hearing from you.

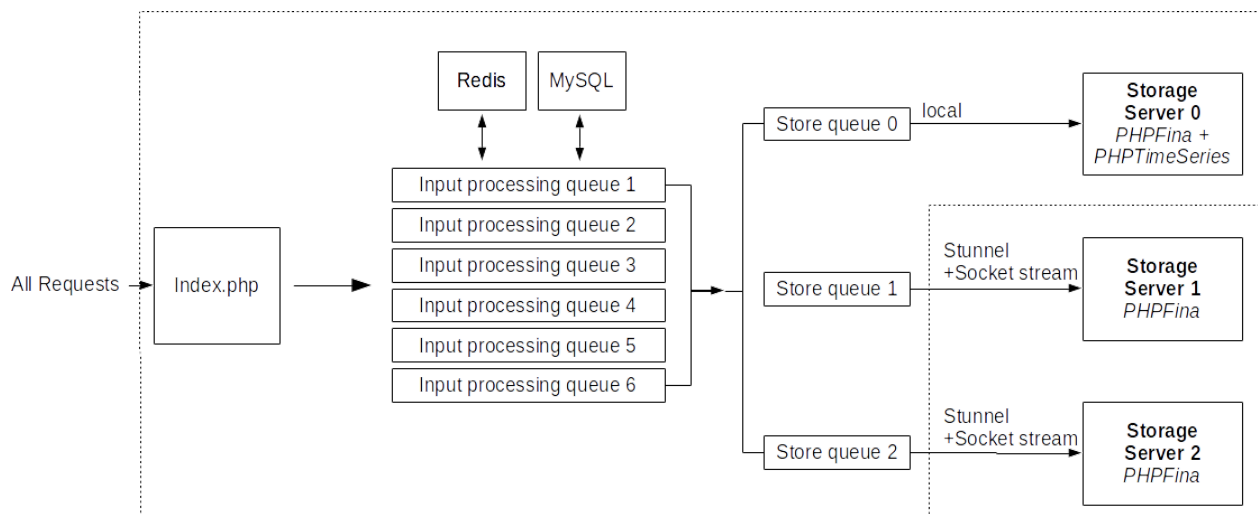
** The salary offer is in line with our own pay at Megni*

Scaling emoncms.org

This description is also available on the emoncms.org readme here:
<https://github.com/emoncms/emoncmsorg>

Current emoncms.org architecture and scaling challenge

Diagram of current (Aug 2017) emoncms.org input processing and data storage architecture:

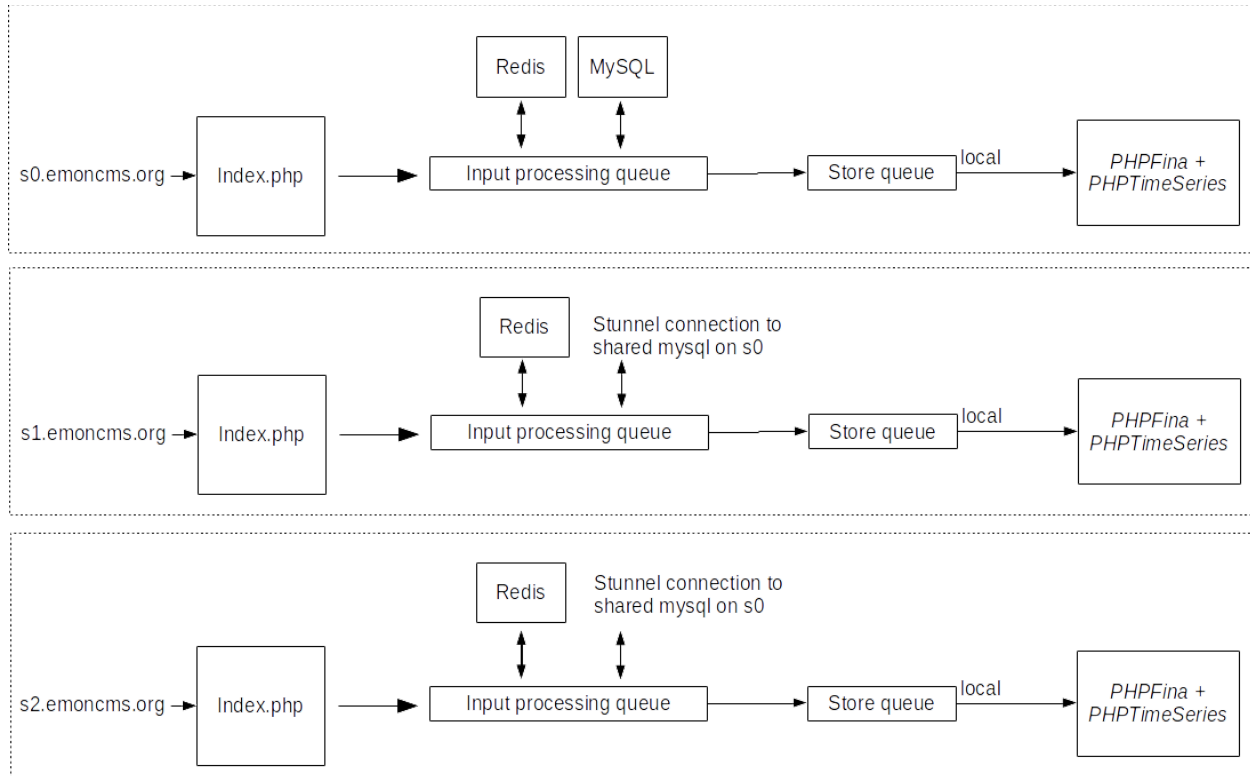


1. All requests including data from monitoring equipment arrives at the emoncms front controller `index.php` on emoncms.org. These requests are a mixture of HTTP and HTTPS requests.
2. For input/post and input/bulk data posts, `index.php` determines the target user from the `apikey` in the request.
3. These requests are then passed to a script called `fast_input.php` which decodes the post and bulk data before directing the input data into 6 input processing queues, split by `userid` e.g: users 1 to 3500 are directed to input queue 1.
4. The input processing queues then work through the inputs in parallel at a speed that is largely determined by the latency on the `redis` and `mysql` socket connections.
5. The resulting feed insert/updates are passed to their respective storage queues 0,1,2.
6. Storage server 0 is on the main server alongside the full emoncms installation.
7. Storage server 1 and 2 are on separate machines. Feed data to be written is streamed across to these servers via a socket stream over an encrypted stunnel.

The present implementation provides storage server scalability and has allowed emoncms to grow for a number of years. It does not however provide sufficient scalability of the entry point (apache2) and input processing stage as these are all running on a single machine.

With an increasing number of requests arriving at emoncms.org an increasing number of apache2 instances are spun up eating up cpu and memory. At the same time there is plenty of idle cpu and memory on the separate storage servers.

An alternative approach could be to split emoncms.org more fully over multiple servers so that each server hosts a share of requests, input processing and storage. Each server would be responsible for a block of users. The mysql database is located centrally ensuring that all user-id's and feed-id's are unique across emoncms.org. Each server has its own redis server for caching of fast changing feed meta-data such as last time and value limiting the request volume to the central mysql server.



The challenges as we see it with this approach are:

- This solution adds the complexity of different entry points e.g: s0.emoncms.org, s1.emoncms.org.
- There would need to be a way of directing users to the correct server from emoncms.org.
- Requesting data across multiple servers may also become more complicated.

If you would be interested in helping us solve these challenges or have another alternative approach to consider we would love to hear it!