White paper Energy Flexibility Platform and Interface (EF-Pi)

How the developments in the Energy Flexibility Market can be accelerated by the Energy Flexibility Platform and Interface

TNO

June 1, 2015
Authors: Bram van der Waaij, Wilco Wijbrandi, Mente Konsman

White paper Energy Flexibility Platform and Interface (EF-Pi)

How the developments in the Energy Flexibility Market can be accelerated by the Energy Flexibility Platform and Interface

Introduction

In this document we describe how the developments in the Energy Flexibility market can be accelerated by the Energy Flexibility Platform and Interface (EF-Pi). This document is for those who are interested in Energy Flexibility:

- Developers of Smart Grid services
- Manufacturers of smart appliances
- Operators of smart grid services
- Users of smart devices and appliances

For many years, power grids have formed a reliable infrastructure, which transports power generated at power plants towards domestic, commercial and industrial consumers. However, the way we use the power grid is changing. First of all, we are using electricity for new energy intensive applications such as heating buildings and transportation. Secondly, we are introducing renewable energy sources like solar panels and wind turbines, which do not provide steady levels of production. Finally, energy is being generated at a small scale by devices such as solar panels and micro-CHP's, which means that power has to be inserted at many different locations in the power grid.

These three trends have a high impact on the stability of the power grid itself. Charging electric vehicles requires enormous amounts of power. Instead of a continuous flow from power plant towards consumers, generation becomes much more fluctuating because most renewable energy sources cannot deliver a steady and continuous power production. Solar panels produce energy only when the sun is shining. During the night or when it is cloudy, very little or no energy is generated. Wind turbines do not produce power when there is too little, or too much, wind to safely turn the blades of the turbine. As such, renewables often have an intermittent character and cannot be planned in advance. Additionally, the electricity flows in the power grid are changing, because energy is inserted into the grid at different locations, such as from households.

These trends introduce two problems for the power grid: First of all, it becomes more difficult to balance the production and consumption in the power grid. When production and consumption are not in balance, the power grid will fail. Having many renewable energy sources means that there still need to be traditional power plants available which can produce electricity when there is no renewable energy available. Secondly, as the usage of electrical heating and electrical vehicles increases, the transportation capacity of the power grid will soon be inadequate. Increasing the capacity is very costly. A distribution system operator (DSO), the owner of the power grid, would like to avoid or postpone these investments as much as possible.

Sometimes consumers are willing to provide some flexibility in the energy consumed or produced by their devices. For example, an electric vehicle doesn't have to start charging right away when it is plugged in in the afternoon if the owner doesn't need the car until the next morning. By adding ICT and intelligence to the power grid, we could schedule the car to start charging during the night when there is more capacity or renewable energy available in the power grid. A power grid which is made more intelligent with this technology is called a Smart Grid. The system that automatically schedules consumption and production is called the Smart Grid service.

Over the years a lot of different Smart Grid service approaches have been developed to intelligently utilize the flexibility offered by smart appliances; appliances that can communicate with Smart Grid services. These Smart Grid services all have their own optimization strategies and characteristics, making them incompatible with each other. For example, PowerMatcher [PowerMatcher] is an algorithm that tries to balance energy production and consumption in real time, where Triana [Triana] is an algorithm that tries to make plans of consumption and production in advance.

Also, there are a lot of different appliances which are able to provide flexibility such as washing machines, heat pumps, micro-CHP's and fridges. Each vendor has their own API for communicating with these appliances, and there is a lot of diversity in the way these appliances can communicate. Examples include Ethernet, ZigBee, Z-wave, Wi-Fi and PLC. The management of all this variety on both the Smart Grid services and the appliance side introduces a big challenge for rolling out Smart Grids on a large scale.

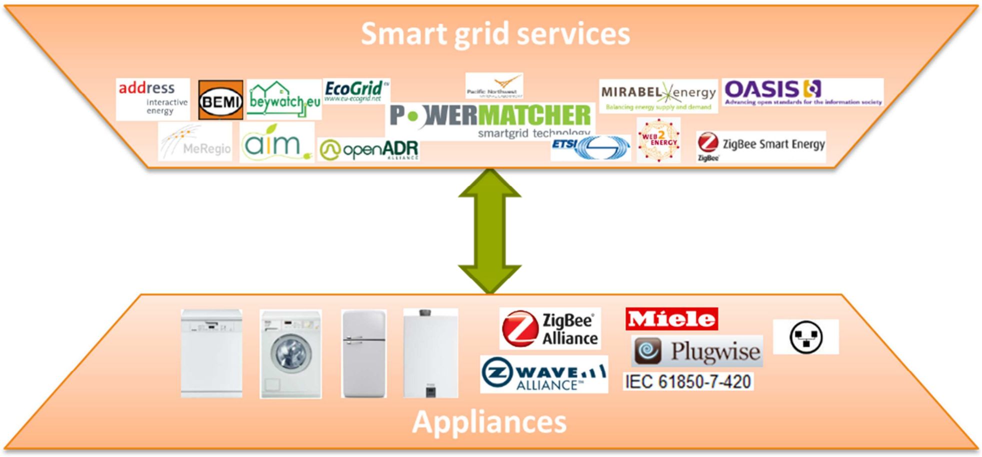


FIGURE 1: MANY SMART GRID APPROACHES VERSUS MANY APPLIANCES

The Smart Grid problem

Nowadays, most Smart Grid services are tightly coupled to particular smart appliances. There is no standard way to describe flexibility in producing or consuming electricity. The flexibility is implicit and appliance specific. This introduces problems for all the stakeholders involved:

Developers of Smart Grid services need to build interfaces for a large amount of appliances, which will cost them a lot of effort. Since their time is limited, there will always be appliances that are not compatible.

Manufacturers of smart appliances on the other hand, have many Smart Grid services they need to interface with. Not having interfaces to major Smart Grid services will give them a disadvantage.

The ***Operators of Smart Grid services*** need to manage many different software packages and cannot switch from a Smart Grid technology because they are restricted to the Smart Grid services compatible with the smart appliances in their portfolio.

And finally, ***End users*** are restricted in the type of smart appliances they can use. An additional issue could be privacy infringement due to direct connection between the Smart Grid service and the appliances.

The Energy Flexibility solution

A group of companies and institutions recognized these problems and pledged to solve them together. They founded the Flexible power Alliance Network (FAN) [FAN]. FAN is an independent network of companies and institutions that jointly develop and manage the international, open FAN standard. The current members of FAN are: Accenture, Alliander, CGI, DHPA, ECN, Stedin, Technolution and TNO. Together, they are developing the Energy Flexibility Platform and Interface (EF-Pi, formerly known as FPAI)

The EF-Pi, which is the subject of this white paper, aims to create an interoperable platform that is able to connect to a variety of appliances and support different Smart Grid services. This way, no appliances need to be changed when a consumers switches from one Smart Grid Service to another. At the same time the EF-Pi makes it easier for service providers to introduce new services, since they do not depend on the types of appliances the end user owns.

Approach

The goal of the Energy Flexibility Platform and Interface (EF-Pi) approach is to decouple Smart Grid services from the customer appliances.

This opens up the markets and gives the customer freedom of choice in Smart Grid services. The End user should be able to combine it with all the connected appliances he already owns in his house, without losing control and ownership.

The EF-Pi is an open-source software platform that runs on low-power hardware located at a convenient place in the building. The EF-Pi communicates directly with smart appliances inside the building. The EF-Pi has an easy to use interface, which the end user can use to configure and control his own appliances and get insight in how his appliances are functioning.

The core of the EF-Pi is the Energy Flexibility Interface (EFI). The EFI is a generic interface which appliance manufacturers can use to describe energy flexibility, and which Smart Grid service developers can use to describe how they want to use this flexibility. The EFI effectively provides a common language for both sides, facilitating interoperability between all Smart Grid services and smart appliances.

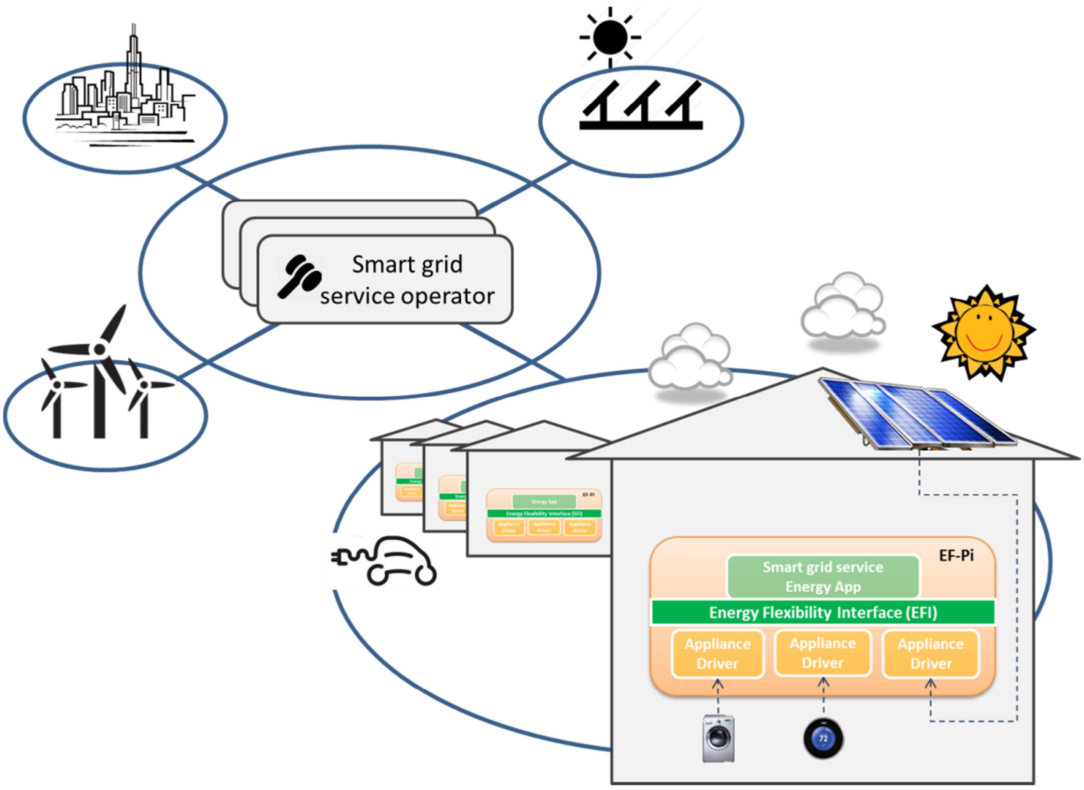


FIGURE 2: OVERVIEW OF A SMART GRID AND ITS COMPONENTS, TOGETHER WITH THE EF-PI AND EFI

Guiding principles

The solution the EF-Pi introduces is based on the following set of guiding principles:

- *End user autonomy:*
    - The end user should have full control over their appliances, whether they are used or not.
    - The end user should know which data is shared with other parties and have full control over which data is shared.
    - The use and specification of the appliances of the end user must stay as hidden as possible.
- *Cost efficiency and scalability:* The solution should be scalable towards nationwide deployments at a reasonable cost.
- *Open standards:* The solution must be based on open standards as much as possible.
- *Open software:* The solution must be freely available as open source software.

Flexibility

The first guiding principle states end user autonomy. Besides the decoupling, the EF-Pi uses explicit flexibility to conceal the appliances as much as possible, without losing the opportunity to offer flexibility to the Smart Grid service. The flexibility forms the pivot point between Smart Grid services and smart appliances.

Figure 3 shows an example of how the EF-Pi can be used. On the top you can see two Smart Grid service Operators for Triana and PowerMatcher. Each offering a different service towards the end user to manage their flexibility in return of some incentive. Below are the buildings of the end users, containing their smart appliances such as washing machines, building heating, electric vehicles & solar panels.

Since all the smart appliances have their own API and communication method, the EF-Pi needs to know how to communicate with these smart appliances. In order to do this, the right *Appliance Drivers* have been installed on the EF-Pi. An Appliance Driver is a software component that communicates with the appliance and communicates the available flexibility through the EFI to the *Energy App*, the software component that represents the Smart Grid service on the EF-Pi. An Energy App can connect with all available Appliance Drivers, however an Appliance Driver can only be controlled by one entity and therefore is connected to only one Energy App.

Energy Apps contain logic to manage the flexibility offered by one or more appliances. Many energy apps will use a connection to a central system of the demand response system to manage the flexibility from a broader perspective, such as Triana and the PowerMatcher. Some Energy Apps manage the flexibility only from a local perspective, for example to use the energy produced inside the same building as much as possible. Some buildings will use only one Energy App for all their flexibility, other buildings could use two or more Energy Apps to profit from different providers.

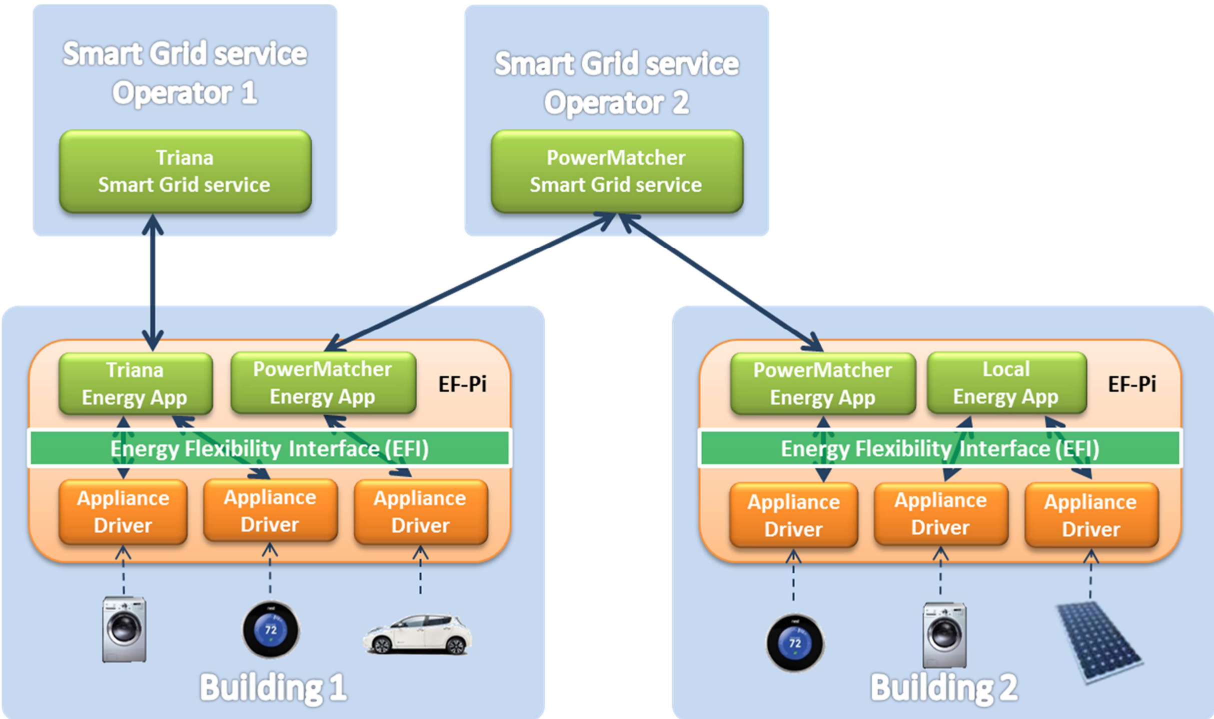


FIGURE 3: AN EXAMPLE USE-CASE FOR THE EF-PI

Since both the Appliance Drivers and the Energy Apps communicate solely through the EFI, they can be easily be reconfigured. In this example, the owner of Building 1 can decide to switch entirely to Smart Grid service Operator 2 by simply reconfiguring his appliances via the EF-Pi user interface.

Stakeholder gains

Using the EF-Pi approach, all mentioned stakeholders benefit:

Developers of Smart Grid services
EF-Pi enables interoperability of their technology with more smart appliances and reduces time to market.

Manufacturers of smart appliances
EF-Pi enables interoperability of their appliances with more Smart Grid services and reduces time to market.

Operators of Smart Grid services
EF-Pi provides a scalable, manageable, future proof solution.

End users
EF-Pi provides an updatable, high personalizable solution, it avoids vendor lock-ins and provides a user interface. There will be more freedom in choices of appliances and services.

Energy Flexibility Interface (EFI)

The Energy Flexibility Interface, in short EFI, forms the interface between the appliances drivers and the energy app. The EFI is the interface to express explicitly the energy flexibility capabilities of the appliances and the energy allocation requests from the Smart Grid service operators.

Connected to the EFI on one side are the Energy Apps, on the other side are the Appliance Drivers. The Appliance Driver can pass information about the energy flexibility the appliance has to offer, up to the energy app. The energy app can pass allocation requests down to the appliance driver, asking for a certain behavior from the appliances. The appliance and the end user always have the option to ignore the request and do something else. The end user is always ultimately in control, not the Smart Grid service operator.

The energy flexibility, from Appliance Driver to Energy App, is described in *control space* messages, defining the space in which the appliance can be controlled. It describes the flexibility at this moment and optionally in the (near) future. Allocation requests, from the Energy App down to the appliance driver, are described in *allocation* messages. See also Figure 4.

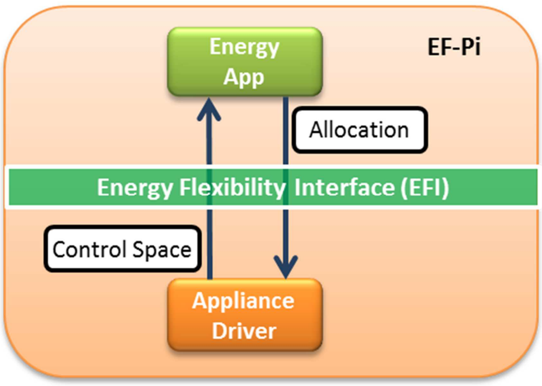


FIGURE 4: THE ENERGY FLEXIBILITY INTERFACE (EFI).

The *Appliance Driver* knows the technical capabilities of the appliance it is connected to. It continuously defines the operational boundaries, the control space, based on the user comfort settings. It has internal autonomous behavior if no instructions (allocations) are given, in order to keep the appliance running.

The *Energy App* chooses how to use the offered flexibility. If needed, it communicates with a Smart Grid service central system to collect more centralized incentives such as balancing the electricity production and consumption in the local street, or in the local city. It tries to find the optimal solution between the boundaries as defined by the user and the appliance (control space) and the Smart Grid service incentives. Based on this it makes allocations for the Appliance Driver.

Smart appliance categories

The characteristics of the flexibility are defined in four categories. The capabilities of an appliance must be matched to one or more of these categories.

Categorie	Description	Examples
Uncontrollable	Has no flexibility, is measureable and may provide forecast	Solar panel, Wind Turbine, TV, indoor lighting
Time Shiftable	Operation can be shifted in time, has a deadline	Washing machine, Dishwasher
Buffer	Flexible in operation for either production and/or consumption and operation is bound by a buffer	Freezer, Heat Pump, CHP, Battery, Electric Vehicle, Cooling systems
Unconstrained	Flexible in operation for production. The operation is not bound by a buffer	Gas Generator, Diesel Generator

Uncontrollable

The Uncontrollable category of appliances are those appliances that cannot be controlled and therefore cannot be used in a flexible manner. Information from these appliances is however relevant for Smart Grid services. This applies in particular to the sources of renewable energy such as solar panels. It is uncontrollable when the sun is shining and making the solar panel produce electricity. The same applies for inflexible domestic loads, such as a TV. We watch TV when we want and that can change by the minute.

Sometimes it is possible to predict the near future behavior of an uncontrollable device. There are systems which predict the sunshine for the next hour at a specific location. This information can be passed as a forecast.

Time Shiftable

Where the Uncontrollable category cannot be controlled, appliances from the Time Shiftable category can. This category is designed for appliances which have the flexibility to shift their usages in time. It is not about using an appliance less, but rather by being flexible when to start.

Examples of such appliances are washing machines and dishwashers. Both need to run at some moment in time to either wash clothes or dishes. The exact moment when this is done is not that important as long as it is finished before a specified deadline.

This method introduces flexibility in the electrical system by moving the usage of electrical energy in time. If the electrical system needs less consumption, the running of these devices can postponed to a better suiting moment. Once the device starts working, it performs its normal operation and can typically not be controlled anymore.

Buffer

Some appliances have flexibility in the form of a buffer. The usage is not directly shifted in time, as with the time shiftable. The buffer alters its usage in time to provide an optimum energetic solution. A buffer can produce or consume energy between a minimum and maximum fill level of the buffer.

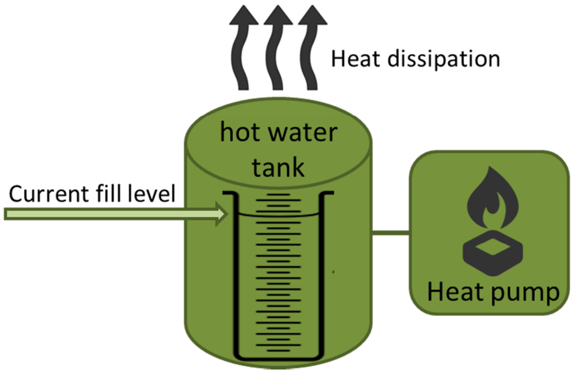
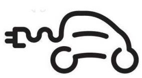


FIGURE 5: HOT WATER TANK BUFFER EXAMPLE

An good example of a buffer is a hot water tank, see also Figure 5. Connected to the tank is a heat pump to heat the water to a desired temperature. The tank has an indicator for the current fill level. Gradually heat is dissipating from the buffer, which can modeled. The hot water tank gives the heat pump more flexibility in when to consume electricity (and produce heat).

The Buffer category lists the flexibility in either production or consumption of energy that is available between the minimum and maximum fill level of the buffer. What is modeled is the filling or emptying of the buffer, which takes energy. What is actually in the buffer (e.g. hot water, electricity, pressure) is of no direct relevance for the EFI.

Another example of energy storage is a refrigerator or freezer, it stores energy by staying cold and when it warms up, one can say the buffer is slowly emptied. At a certain predefined point, i.e. temperature, the refrigerator or freezer will turn on and start cooling again, i.e. filling the buffer again. This filling does not need to be done at exactly the same temperature each time. It can be done a little sooner or later, depending on the current state of the electrical system. Of course, if the temperature rises too high, then there is no flexibility anymore and the appliance must be turned on.



Just as heat and cold are stored in buffers, so is electrical energy. This is a battery of course. It too fits within the Buffer category and is governed in much the same way as with heat and cold. The Electrical Vehicle (EV) is not much different from a battery when it is used in Vehicle to Grid[1] (V2G) scenarios. The exception is that where a battery is usually stationary, the EV can change location and its fill level can change unpredictably when is reconnects to the electrical system.

[1] Vehicle to Grid is a concept in which the battery of an electrical vehicle is used as a generic energy source when the demand for electricity is very high

Unconstrained

As discussed, all categories thus far have been constrained in one way or another. The Unconstrained control spaces have no such restrictions. The Unconstrained category lists the flexibility in either production or consumption of energy without being bound to a buffer or deadline.

A generator is an example. It can be switched on in order to provide extra power to the electrical system and can be switched off when the need has dissipated. There are only technical device constraints when it can or cannot run, such as the amount of diesel it has left, or how fast the generator may be started again after it has been stopped.

Open source runtime

In order for such a solution to be successful, we need to make it as easy as possible for all parties involved to make use of this interface. That’s why we implemented the EF-Pi as an open-source runtime for Building Energy Gateways, which can be installed in homes and small business environments.

The building energy gateway is typically a small, preferably low-power computer that can communicate with all the smart appliances inside the building. This means that this computer needs to be equipped with adapters for conventional physical communication methods for smart appliances, such as the Serial Port, Ethernet, Wi-Fi, ZigBee and Bluetooth. The Building Energy Gateway receives information about all the smart appliances and can send back instructions where applicable.

Apps

There are many types of smart appliances and Smart Grid services, and in the future there will be even more. As described in Chapter 2, the software components on the EF-Pi that communicate with the smart appliance are called Appliance Drivers, and the software components controlling the smart appliances through the EFI are called Energy Apps.

In the EF-Pi runtime, Appliance Drivers and Energy Apps are packaged in the form of an *App*. Apps can easily be installed in a few seconds from the EF-Pi Store, an online repository containing all the Apps available for the EF-Pi. This process is inspired by modern mobile operating systems (such as Android and iOS), where mobile applications can easily be installed from the AppStore or Google Play. Apps can define their own user interface as part of the Appliance Driver and/or Energy App.

When the user buys a new smart appliance, it has to connect it to the local network. In most cases the plug&play mechanism of the building energy gateway will recognize it. The EF-Pi will download the appropriate driver from the EF-Pi Store and connect it with the energy app of the Smart Grid service. Making the installation process as easy as possible for the end user accelerates the adoption of Smart Grid services.

EF-Pi User Interface

The end-user interacts with the user interface of the EF-Pi. From here, the user can manage and see the status of all the installed smart appliances and Smart Grid services. They can configure and control their appliances, and gain more insight into the energy consumption and production inside the building.

The EF-Pi user interface consists of several pages. The primary page is called the *dashboard*. Here the user can see an overview of all the installed Apps. Each App can add a so called widget to the dashboard: a small user interface showing the most important information and possibly some basic control options. Apps can also create their own page in the user interface, giving them more space to provide their functionality. Other pages that are always available are the system configuration page and the EF-Pi Store page.

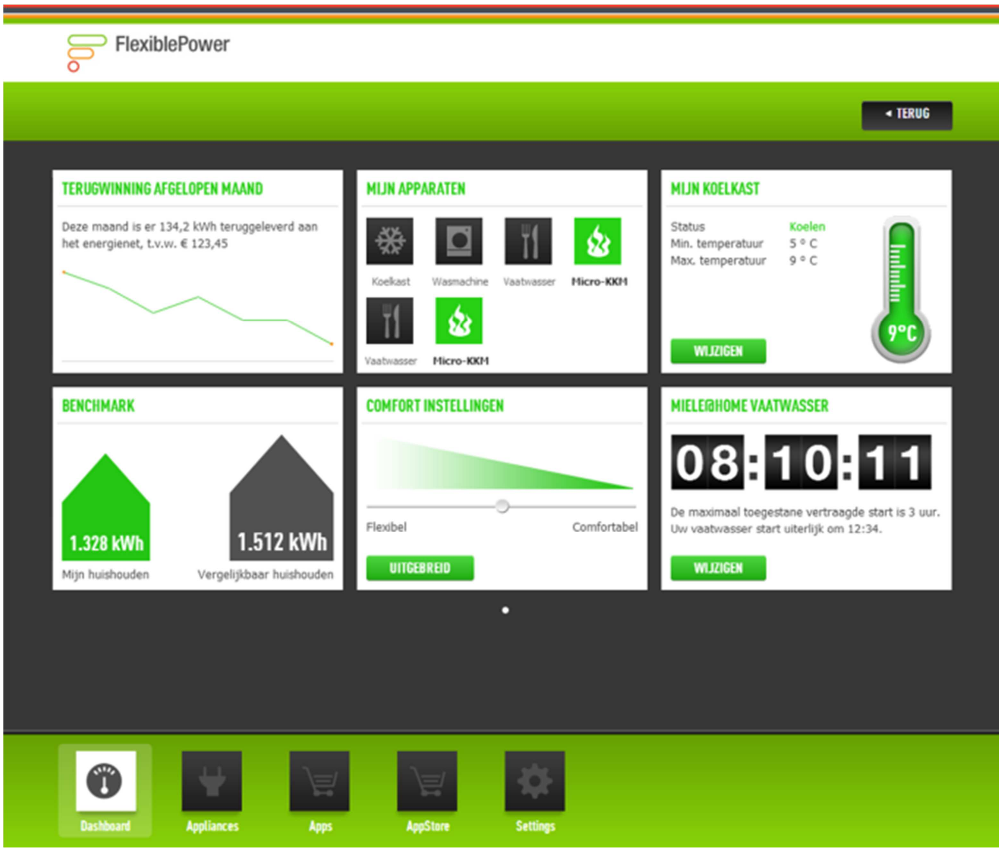


FIGURE 6: USER INTERFACE (IN DUTCH) WITH WIDGETS OF THE EF-PI.

Implementation

The EF-Pi is built on top of OSGi [OSGi], a modular service platform for the Java programming language. The advantage of Java is that compiled Java code runs on every software and hardware platform where a Java Virtual Machine is available, without the need for recompiling the software. This allows the EF-Pi to run on many different hardware platforms and operating systems. Most building energy gateways are low-power machines with an ARM processor. The EF-Pi runs perfectly fine on such computers, but for example also on a Raspberry Pi running Linux.

The EF-Pi is designed to make it easy to create drivers and energy apps for it. Very little OSGi knowledge is needed, only the handling of the EFI messages needs to be programmed. It is necessary to program the translation into appliance commands for the appliance driver, and for the energy app the algorithm to find the optimal solution between the user and appliance boundaries and the Smart Grid service incentives. The EF-Pi re-uses existing, proven OSGi components as much as possible.

The EF-Pi User Interface is implemented as a web application. It can be accessed from any device with a web browser in the same network as the EF-Pi.

Field trial support: remote management and monitoring

In order to support field trials with the EF-Pi on Building Energy Gateways, two additional services are added: remote management and monitoring.

Remote management

In some cases, such as field trials, it is desirable to remotely manage a large cluster of EF-Pi instances. Some use cases require operators to update configurations, update Apps or monitor the behavior of Apps. Therefore the EF-Pi has a Management Center, which allows operators to monitor and manage large clusters of EF-Pi instances. Using configuration templates, the operator can roll out new Apps or configuration changes to many EF-Pi instances easily. The EF-Pi and the Management Center use the industry standard protocol named CWMP (described in TR-096 [TR069]), which is published by the Broadband Forum.

Monitoring

Field trials often have a need to collect usage data of the participants. In a field trial end-users should agree that this information is allowed to be collected.

The Monitoring Framework allows every App running on the EF-Pi to define and report their own Observations. These observations can be processed by any component on the EF-Pi and can be collected in a central database for monitoring purposes. The Monitoring Framework allows App developers to identify and easily publish their own measurements to monitor and indicate the performance of the appliance, Smart Grid service or the App itself.

Getting started with EF-Pi

In order to promote the EF-Pi platform and accelerate market adoption, the EF-Pi is released as open-source software under the Apache 2.0 license. This license makes the source code of the EF-Pi platform publicly available for everyone who wants to inspect it, improve it or build (commercial) products based on the EF-Pi, without the need to pay licensing fees. While the EF-Pi platform is open source software, additions such as improvements and Apps for the EF-Pi do not have to be released under an open source license, although we would encourage everyone to do so.

If you want to get started with the EF-Pi, you can find the source code, documentation and tutorials on the EF-Pi open source community website: http://flexiblepower.github.io/. Here you can also come in contact with other EF-Pi users and share ideas, features requests and issues with the EF-Pi.

Projects

This whitepaper is written as part of the Virtual Infrastructure Operating System (VIOS) project. Over the last three years, the VIOS project has been working on improving EF-Pi. The goal of the project was to come up with a ‘virtual infrastructure’ to achieve interoperability between Smart Grid services and appliances. One of the primary goals was to test and improve the interface between services and appliances. There are however many more projects in which the EF-Pi is used successfully, among which:

VIOS (2012-2015)
The project was executed with a consortium of partners with complementary skills: iNRG (Smart Grid software company), Technolution (technical integrator), Alliander (Distribution System Operator) and TNO (applied research institute). The project received a subsidy from RVO (the Netherlands Enterprise Agency).

During the course of the project EF-Pi was improved and tested for two seasons. Experiments in the first heating season (a holiday home in Bronsbergen, The Netherlands) resulted in improvements to the interface for optimizing the usage of flexibility that appliances have. This improved interface was tested in the second season (testfacility ‘EnTranCe’ in Groningen, The Netherlands).

ESC (2012-2015)
The ‘Energy Supply Cooperative’ project (ESC) is a consortium project with Karlsruhe Institute of Technology (KIT), Evohaus and TNO. ESC develops an energy management solution for optimizing own PV usage in a community of 73 (newly built) houses in Widdersdorf (Germany)
and also a (yet to be built) community in Hilden (Germany). In this community, EF-Pi and PowerMatcher are used to control heating clusters with CHPs.

HEGRID (2013-2015)
Hybrid Energy Grid Management (HEGRID) is developing a ‘multi-commodity’ matcher. The goal here is to optimize on multiple commodities like: gas, electricity and heat. Partners are: Siemens, TNO , Deutsche Telekom, University of Eindhoven, University of Twente and CWI. The multi commodity matcher (based on Triana, developed by University of Twente) will run on EF-Pi to control devices.

Lochem (2013-2015)
Lochem Energie is a smart grid initiative of the municipality of Lochem (The Netherlands). The project uses the EF-Pi to connect solar panels, electric cars and the medium voltage transformer within the power grid, with the PowerMatcher. Partners in the project are Locamotion, Lochem Energie, Universiteit Twente, Eaton Industries and Alliander.

Idego (2014-2015)
In the Idego (Intelligentie Duurzame Energie Gebouwde Omgeving) project ZON Energie Spanbroek, TNO, ProxEnergy and Priva work together to develop and test a ‘Heatmatcher’, to manage heat flexibility. EF-Pi is extended in Idego to be able to connect physical heat grids.

Future work

The current version of EF-Pi offers a complete set for the management of energy flexibility in homes and buildings. It is our ambition to let EF-Pi grow by working closely together with many parties including the open source community and new deployments.

Next to this the EF-Pi team has additional ambitions it is currently working on:

1. Standardizing the EFI messages. At the beginning of 2015 the first meeting has taken place with the ISO workgroup ISO/IEC JTC1 SC25/WG1.
2. Creating a cloud deployment of the EF-Pi together with internet connected small appliances. This will eliminate the need for a Building Energy Gateway at the premises.
3. Go to a large scale deployment of many thousands of houses. This would prove the maturity of EF-Pi.

Further reading

This whitepaper gives an overview of the need and approach taken by the Energy Flexibility Platform and Interface. For more information we refer to three websites:

- Source code of the EF-Pi and the wiki can be found at https://github.com/flexiblepower/
- Information about the EF-Pi product can be found on the community website: http://flexiblepower.github.io/. The EFI specification is part of the EF-Pi documentation, which can also be found on this website.
- Information about the organization, the Flexible power Alliance Network (FAN) can be found at: http://www.flexiblepower.org/

References

[FAN] Flexible power Alliance Network, http://www.flexiblepower.org/

[OSGi] OSGi Alliance, http://www.osgi.org/

[PowerMatcher] http://flexiblepower.github.io/technology/powermatcher/

[Triana] http://eprints.eemcs.utwente.nl/21154/01/thesis-final.pdf

[TR069] http://www.broadband-forum.org/technical/download/TR-069_Amendment-5.pdf