glue.things – a Mashup Platform for wiring the Internet of Things with the Internet of Services

Robert Kleinfeld
Senior Research Engineer
Fraunhofer FOKUS
Berlin, Germany
robert.kleinfeld@fokus.fraunhofer.de

Lukasz Radziwonowicz
Research Engineer
Fraunhofer FOKUS
Berlin, Germany
lukasz.radziwonowicz@fokus.fraunhofer.de

Charalampos Doukas
Senior Research Engineer
CREATE-NET
Trento, Italy
charalampos.doukas@create-net.org

ABSTRACT

Internet of Things (IoT) is a trending topic. Market analysts predict a large growth of the entire IoT market with an explosion of new products. In this paper, we argue how the data of these devices can be connected in an efficient and extensible way. For this purpose, we present glue.things a mashup platform for wiring data of Web-enabled IoT devices and Web services. The work carried out in this paper addresses concepts and technologies that are referred as "web-friendly IoT technologies". We elaborate aspects of device integration, real-time communication and data stream mashups. These aspects are discussed on features and technologies of the prototype implementation glue.things.

General Terms

Design, Prototype Implementation

Keywords

Internet of Things; Internet of Services; Web of Things; Mashup Platform; Discovery; Composition; Gateway; Web of Things Hub; Web Tools

1. INTRODUCTION

The Internet of Things (IoT) makes many of the familiar devices in our lives, from door locks to tollbooths to refrigerators, suddenly Internet-connected, smartphone-accessible, and responsive. In particular, with respect to the consumer space, many products and services have already crossed over into the IoT, including kitchen and home appliances, lighting and heating products. The numbers being forecast for the IoT are truly astonishing. Gartner find that the numbers of everyday and enterprise devices that will soon be connected to the Internet will be huge: 1.9 billion devices today, and 9 billion by 2018, according to Gartner estimates, roughly equal to the number of smartphones, smart TVs, tablets and wearable computers, and PCs combined [1]. Especially market analysts predict that this trend will drive trillions in economic value as it permeates consumer and business life [2]. Despite all these positive predictions, the question arises, how are the "things" in the IoT actually put together?

This challenging question manifests itself in a large number of IoT infrastructures that are built in silos with a lack of interoperability at the gateway, service and application layer [3]. With a dynamic IoT market and new products becoming available all the time, it is inefficient and time-consuming for developers to support all these products [4]. The number of communication protocols and IoT technologies is steadily increasing, and more and more APIs and data models are released for a specific domain and product. It can therefore be concluded that there has been a large number of IoT infrastructures and proprietary technologies in the recent years. For each specific product developers have to implement or integrate a gateway for collecting device data and controlling its settings.

Against this background, the IoT landscape can be grouped into a number of categories with specific features and technologies. These categories include hardware boards with micro controller, Web-enabled IoT devices, IoT platforms and hybrid approaches such as Web of Things (WoT) hubs [5]. Popular examples for hardware boards are Arduino[1], openPicus[2] with Flyport[3], Spark[4] and many more. They are providing Software Development Kids (SDKs) or client libraries for device communication with a specific integration effort for accessing these devices in consumer applications. Furthermore there is a huge market growth of Web-enabled IoT devices in the smart home and wearable device domain. Products such as Philips Hue[5] and Fitbit[6] supporting Application Programming Interfaces (APIs) based on Representation State Transfer (REST) for easy device integration, but with a gateway as broker between device and Web service of the provider. This trend is set to continue: more and more products will come out with Web-based interfaces built on established Web standards. There are IoT platforms supporting specific application domains such as openHAB [6] in the smart home environment. They provide Integrated Development Environments (IDEs), proprietary APIs, data models and script engines. However, these IoT platforms moving in a Web-centric direction with REST APIs and Web-based dashboards. In contrast, WoT hubs abstract form device-specific gateways and technologies. With the purpose to provide a set of Web-based APIs and tools for developing IoT applications in an efficient environment. Platforms such as Xively [7], Temboo[7] or Carriots[8] providing an entire programming environment including Web-based APIs, SDKs, client libraries for specific hardware boards and dashboard for device management. WoT hubs are located at the highest level of the IoT software stack (gateway, service and application layer). In conclusion, the vertical nature of this IoT software stack has contributed to fragmentation of its infrastructure. In this fragmented landscape, Web technologies are the most promising approach for interoperability and standards.

[1] http://arduino.cc/

[2] http://www.openpicus.com/

[3] http://www.openpicus.com/site/products

[4] https://www.spark.io/

[5] http://www2.meethue.com/

[6] http://www.fitbit.com/

[7] https://www.temboo.com/

[8] https://www.carriots.com/

REST APIs, JSON data models, Web Sockets, Publish / Subscribe mechanisms such as XMPP and JavaScript-based Open Source projects for IoT data stream management are tried and proven in production practice [8].

The work carried out in this paper addresses concepts and technologies that are referred to WoT hubs. The paper elaborates aspects of device integration and real-time communication (Web Sockets, MQTT and CoAP based on real-time data stream networks such as MeshBlu[9], PubNub [9] and servIoTicy [10]), data stream mashups, triggers / actions and finally distributed deployment of these mashups. One special focus of this work is on the composition of data streams from Web services and Web-enabled IoT devices. All these mentioned aspects will be discussed in depth based on the prototype implementation glue.things. glue.things is a coherent and robust mashup platform covering both delivery and management aspects of device data streams, applications and their integration. In this respect, glue.things builds on proven real-time communication networks for easy device integration and data stream management.

This paper is structured as follows: section 2 presents additional approaches of WoT hubs and mashup solutions and how they are related to the concept of glue.things. Section 3 introduces glue.things including architecture, features and benefits with a focus on plugin-able real-time data stream networks. Section 4 presents the core components of the mashup platform. Finally, section 5 outlines further research questions as result of the current work as well as future steps regarding distributed deployment, compositions and an easy integration of devices from the consumer perspective.

2. Background

The term mashup is pretty broad and widespread. A good definition for a mashup tool is a solution that allows developers to combine data streams and then visualize that data through a Web or any mobile application [11]. Usually, mashups are Web applications that can be created quickly using standard Web services (e.g., REST) and components (e.g., Widgets). Early mashup tools are Microsoft Popfly [12] and Yahoo Pipes [13]. Particular, the term and underlying approach is of greater concern in the IoT. While the IoT is still taking shape, it's already make incredible strides as a new frontier for the connected world in which people, devices, environments and virtual objects are all connected and capable of interaction. There's a new wave of technology platforms targeting the need to bridge these sophisticated communications, as well as hardware manufacturers producing physical devices and sensors to power the IoT. There are innovative tools and technologies for developers to build and deploy powerful IoT applications, tools for device manufacturers to quickly add new connected services to products and even technologies for everyday users to enable automated communications and actions between devices and social networks and so much more. For this background there are various providers with mashup tools which also building on REST interfaces for the IoT. Well-known examples for such tools are Clickscript [14], Open Sen.se [15], WotKit [16] as well as Paraimpu [17]. For all four of these, commercial companies have been created products to bring these innovations to the market.

Clickscript is a Firefox plugin written on top of an Ajax library that allows users to visually create Web mashups by connecting building blocks of resources (Web sites) and operations (greater than, if/then, loops, etc.). Since it is written in JavaScript, Clickscript cannot use resources based on proprietary service protocols. However, it can easily access RESTful services, such as those provided by Web applications. This makes it straightforward to create Clickscript building block that represent application. In comparison to Clickscript it should be mentioned that Node-RED[10] as visual Web-based mashup tool getting more and more attention in the IoT. Node-RED has a similar approach in creating Web mashups by connecting building blocks. However, Node-RED is an event-processing engine that basically prevents hardware and software developers from having to continually reinvent the wheel. Node-RED doesn't take away the need to write code altogether, but it does reduce it, and in the process both lowers the technical barrier and allows users to focus on the creating, rather than on the doing. Node-RED provides a user interface where users' drag-and-drop blocks that represent components of a larger system. In Node-RED's case usually the devices, software platforms and Web services are blocks that are to be connected. Further blocks can be placed in between these components to represent software functions that wrangle and transform the data in transit. Node-RED flows are represented in JavaScript Object Notation or JSON and can be easily exported to the clipboard to be imported into Node-RED or shared online.

Open Sen.se doesn't address the IoT, but the Internet of Everything (IoE) – a world in which humans, nature, machines, information, objects and environments all interact and communicate in different ways. Open Sen.se provides a platform for imagining, prototyping and testing new devices, installations, scenarios and applications for this new interconnected world. It's both free and easy to use, enabling both professionals and amateur hobbyists to experiment with ease. Users create dashboards called Sense Boards containing the user interface from a wide collection of plug-ins installed by the user to enter, visualize and process data. Developers can easily integrate devices such as a suitably equipped Arduino by sending or receiving data from named input and output feeds that containing time stamped values using its RESTful API. Output feeds allow Open Sen.se to not only collect sensor data, but also control things. Open Sen.se feeds can contain integers, float, boolean and string values. Currently, Open Sen.se devices cannot be shared with others on the system.

WoTKit serves as a sensor data aggregator, visualization, remote control and processing tool. It aggregates data from a variety of sensors, and allows simple control messages to be passed to actuators. WoTKit allows users to quickly find and subscribe to sensor data of interest, process data and visualize the data using widgets on a dashboard. Like Open Sen.se, the WoTKit provides a flexible dashboard; processing components are included and sensor feeds can contain numeric and string types, however, the WoTKit focuses less on the integration of applications into the platform user interface, rather providing basic built in visuals and processing components. WoTKit supports non-numeric feed values and includes a more comprehensive event-based processing engine. The WoTKit can create pipes that act as connectors, connecting sensor data to actuators such as email and other output modules.

Paraimpu is a social tool for connecting physical and virtual devices, composing and interconnecting them and sharing data and published objects on the social Web. Using Paraimpu, users can create customized applications for the IoT, enabling

[9] http://skynet.im/

[10] http://nodered.org/

users' devices and objects to react to environmental changes, activities or events. Ultimately, the result is a physical-virtual Web mashup in which users can tap into shared data and virtual objects to further their own applications and devices. Paraimpu provides a palette of configurable sensors, actuators and connections that provide processing capability such as filtering and mapping between sensor inputs and actuator outputs. Sensors and actuators can be public, allowing them to be shared between Paraimpu users who are also your friends.

The point is that all those solutions are perfectly valid for their specific applications domain. This paper will not reinvent the wheel of mashup tools but rather adapts and extends current approaches and technologies. Based on evaluations and requirement analyses, Node-RED was identified as the most promising tool for browser-based and visual data stream aggregation. The following sections will discuss the scope of glue.things and the extension of valid mashup approaches in the IoT domain. glue.things can be classified as IoT mashup platform that concentrates on users and less developers. For this reason, glue.things extends Node-RED with easy-to-use and predefined trigger and actions nodes for devices and Web services.

3. OBJECTIVE AND APPROACH

The objective of glue.things is to build an open and extensible hub for WoT applications based on the described IoT landscape in the introduction. Therefore, glue.things builds on established Open Source technologies for device integration and communication, composition of data streams and distributed deployment of these compositions. This section will describe the design approach of glue.things from a developer perspective by addressing the development process of WoT applications in glue.things. In the next step, core components of the architecture will be addressed. This section will conclude with features and benefits of glue.things.

glue.things offers a mashup platform to connect TVs, wearable computing devices, and all of the consumer and business tools to the Internet. For integrating devices, glue.things offers two approaches. (i) First, client libraries leverage the glue.things REST API over HTTP, Web Sockets, MQTT and CoAP to make connecting devices to glue.things. (ii) Secondly, glue.things integrates real-time communication networks such as Meshblu (Open Source machine-to-machine instant messaging network and API), PubNub (Global Messaging Service for building real-time Web and mobile apps) and servIoTicy (data processing platform for the IoT). Given the devolved approach to glue.things, developers can access and communicate with a large variety of devices with low development and integration effort. glue.things supports home appliance devices such as Belkin WeMo Switch, Philips Hue; activity trackers such as Fitbit and Jawbone Up; hardware boards such as Arduino, openPicus Flyport and Spark. The glue.things dashboard supports the development process with features for device and data management, aggregation and manipulation of device data streams and distributed deployment of data stream mashups. The development process contains three steps: connect devices, build device data stream mashups, deploy and run these mashup applications on cloud platforms.

3.1 Development Process

3.1.1 Connect Devices

Developer connect devices to glue.things to make them available on the Web. Supported protocols are HTTP/TCP, STOMP/TCP, STOMP/WS, MQTT and CoAP. glue.things REST API provides communication features (pushing, retrieving and composing data). glue.things supports JSON-based data models for device updates and the description of the processing model. Web communication protocol provides features for creation and management of pub/sub subscriptions, invocations and status check of actuations. These features are available for developers via client libraries. For a more intuitive registration of devices, the dashboard supports device templates. Devices can be registered via a Web formula in a few steps.

3.1.2 Collect Data and Build Mashups

glue.things integrates a mashup tool to aggregate, manipulate, and mashup data streams from connected devices. This mashup tool enhances Node-RED [20] for visual stream compositions (automatic update of the node list if a new device is registered, multi-user and session capability). Developers grab the output of any data stream as JSON and use it in the final mashup application. glue.things mashup tool supports nodes for multiple Web-enabled IoT devices as well as Web services such as Foursquare[11], Twitter[12], Google Speech To Text[13], OpenCV [18] for facial recognition and recommendations. Complex mashups between IoS and IoT can be realized. Finally, developers have the option to define condition statements, triggers and actions for these mashups of data streams.

3.1.3 Deploy and Distribute Mashups

Mashups are deployed as Node-RED application including triggers, actions and authentication / authorization capabilities. glue.things mashup applications have different presentation for developers and end-users. Developers can access a mashup via a RESTful API for using the provided features in their own native or Web-based applications. End-users can access glue.things for browsing through a collection of mashup applications that can be used for their own purpose. Change device settings, define parameters for triggers / conditions, and apply these mashups to their own home environment including owned devices. Concepts for sharing these mashups with other end-users and distributed deployment on mobile devices are integrated.

[11] https://foursquare.com

[12] https://twitter.com/

[13] https://gist.github.com/alotaiba/1730160

3.2 Architecture

The architecture of glue.things builds on three layers that are shown in Figure 1: (i) device layer, (ii) service layer and (iii) application layer.

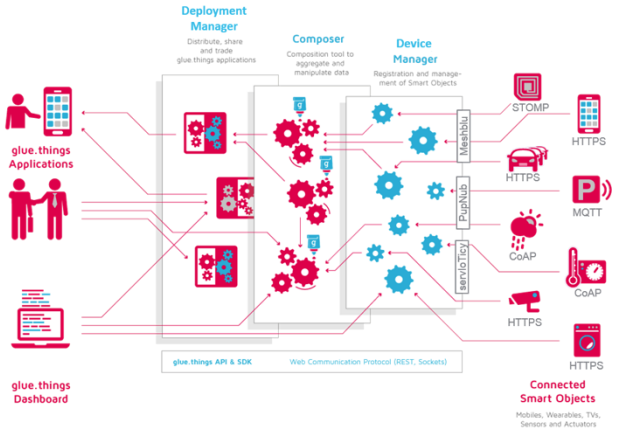

Figure 1: glue.things Architecture

(i) The device layer integrates Open Source real-time communication networks such as Meshblu, PubNub and servIoTicy. This approach has the advantage of device access to a large number of Web-enabled IoT devices. glue.things has not to rebuild multiple device gateways. Furthermore, these networks support messaging with real-time Web Sockets via Remote Procedure Calls (RPC), MQTT and CoAP. An important argument is the support of device directories to discover and query devices across the Internet. All networks are accessible via RESTful APIs. Therefore, glue.things builds on popular Web technologies and proven Open Source solutions. glue.things is open and extensible to integrate further real-time communication networks and gateways. All features can be accessed via the glue.things API.

(ii) The service layer integrates all features from the real-time communication networks to provide mechanisms for data and device management. glue.things builds on Node-RED as development and runtime environment for mashup applications. All registered devices are managed in a virtualized device container. The service layer facilitates features for device discovery, communication, data management and aggregation. Additionally, glue.things uses OAuth as open standard for authorization (especially for Web service authorization). Mashups of data streams are managed as compositions of JSON objects in combination with a Node.JS-based workflow engine.

(iii) The applications layer contains all user interfaces for device registration, configuration and monitoring. These interfaces are developed with the JavaScrip library AngularJS [19] as well as HTML and CSS. glue.things Dashboard combines all user interfaces in a coherent front-end for developers and end-users. Mashup applications are deployed as Node-RED service on CloudFoundry. They are available for further configuration and management in the user interface.

3.3 Features and Benefits

3.3.1 glue.things Dashboard

glue.things Dashboard offers an intuitive, Web-based interface for improving the method of application development for the IoT. Developers can connect, manage and monitor their devices with glue.things Device Manager. glue.things Dashboard contains a powerful mashup tool to aggregate, manipulate and mashup data input from connected devices. Developers take advantage of the glue.things Distribution Manager, for publishing and sharing device data streams and finally distribute the mashup application to potential customers.

3.3.2 Token Management

glue.things provides token management for devices and their data. API access is controlled by API tokens, they give developers the information what they can do and see. Tokens are defined by using glue.things fine grained policy and visibility management.

3.3.3 User Management

glue.things provides different role models for developers and end-users. Depending on the user type, glue.things supports individual views, access policies and privacy enforcements. Developers register their account and glue.things provides access to all management features, the glue.things API and SDK. End-users can access the collection of mashup applications to use them in their own home environment. Additionally, mashup application can be shared with other end-users.

3.3.4 Data Management

glue.things monitors and visualizes the output data of connected devices in a consistent interface. Developers can add new data channels or change their description and type. glue.things Dashboard retrieves real-time updates of device data directly in the glue.things Device Manager. glue.things Token Management provides the ability for defining policies and views of device data streams. Developers can aggregate, manipulate, and mashup device data streams with the glue.things Composer.

3.3.5 Device Management

Developers can easily connect devices such as TVs, wearables, mobiles, home appliances by using device templates with glue.things. Through device virtualization, connected devices are centrally managed and accessible from anywhere on the Internet (based on the real-time communication networks Meshblu and PubNub). Developers can remotely maintain, control and interact with connected devices regardless of their location. Furthermore, developers have the option to monitor the output of connected devices, set offsets, select threshold limits for data streams and modify the data visualization.

3.3.6 Deployment

glue.things builds on Open Source cloud platforms such as CloudFoundry [20], and supports Node.js to build and deploy mashup applications on glue.things. Developers can deploy their applications quickly, easily and in just one click directly from the glue.things Dashboard. glue.things Dashboard provides an overview of all mashup applications, recently activity, and collaborators, giving developers a cohesive interface to manage all mashup applications.

3.3.7 Developer Tools

Developers can access and manage devices with the glue.things API and SDK. glue.things supports various hardware combinations needed to create mashup applications for the IoT. The SDK implements libraries for a variety of platforms. glue.things API exploits the simplicity and power of the well-known REST implementation with the support of JSON and XML data formats.

4. GLUE.THINGS MASHUP PLATFORM

Using the glue.things mashup platform, developers will find all the necessary tools to get prototypes up and running quickly and easily. Tools include Device Manager, Composer and Deployment Manager covering the entire glue.things development process in one common user interface.

4.1 Device Manager

The Device Manager in Figure 2 is a Web-based tool for connecting and registering any device on glue.things. Once a device is connected, developers will retrieve real-time updates of the device data stream.

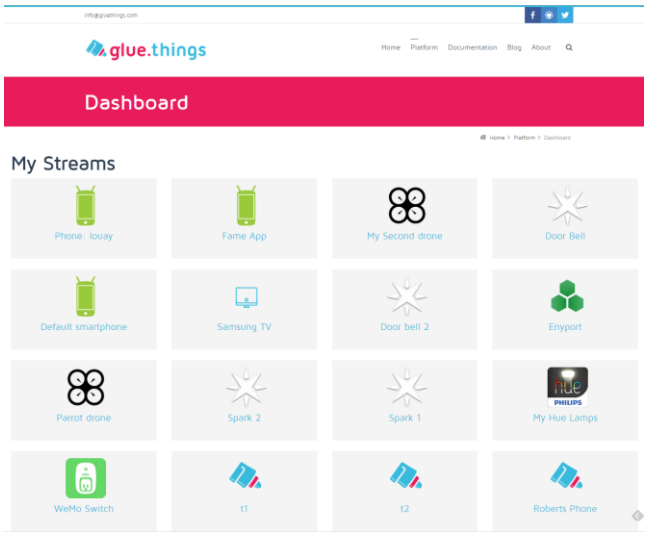

Figure 2: glue.things Device Manager

Developers can manage and organize one or multiple devices, monitor their status and configure access policies for applications talking to them. The output data of devices is visualized in a consistent interface. Depending on the data type, the interface provides the right visualization based on various predefined charts. Once adopted the already mentioned real-time communication networks, the device manager empowers developers with various device templates for mobile, wearables and home appliances. For example, a Philips Hue can registered with the Device Manager in a few clicks. Developers choose the device from a template. Afterwards the device will be discovered in the network. glue.things supports pre-settings of device templates. Developers are free to extend these templates with their own integration.

4.2 Composer

The Composer provides developers the capability to aggregate, manipulate, and mashup device data streams in a visual and intuitive way using a Web-based flow editor that makes it easy to wire together flows. The Composer is a powerful composition tool build on Node-RED as shown in Figure 3. Developers have a bunch of features for data manipulation: select data streams from a collection of devices and Web services; click and drop data streams on a canvas and connect them with flows for further modifications; combine many data streams into one, then sort, filter and manipulate it with basic operations. Finally, developers grab the composition and deploy it as mashup application on the glue.things runtime.

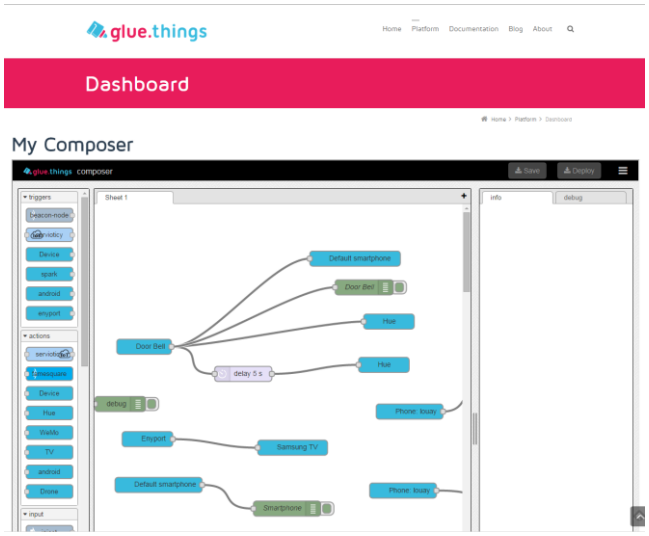

Figure 3: glue.things Composer

The Composer facilitates nodes for Web-enabled IoT devices and Web services. These mashups can be extended with condition statements, triggers and actions. For example, developers can create a simple mashup for Beacons, Foursquare and Philips Hue. One option for a trigger can be: if someone enters a meeting room, then check-in this person on Foursquare and turn on the Philips Hue. Accordingly, this mashup can be deployed as Node.JS application with a RESTful API and input form for configuration and device settings. glue.things provides various trigger and actions nodes for devices and Web services. Users are very fast in creating scenarios for the smart home application domain. Node-RED is extended from a developer tool to an end-user application based on presets and guidelines.

4.3 Deployment Manager

Mashup applications are deployed as Node-RED application on CloudFoundry. Generally, glue.things is open to other or additional cloud platforms. Mashup applications provide a RESTful API for communication and further integration in native applications such as iOS or Android. As shown in Figure 4, these mashups are presented as collection in the user interface.

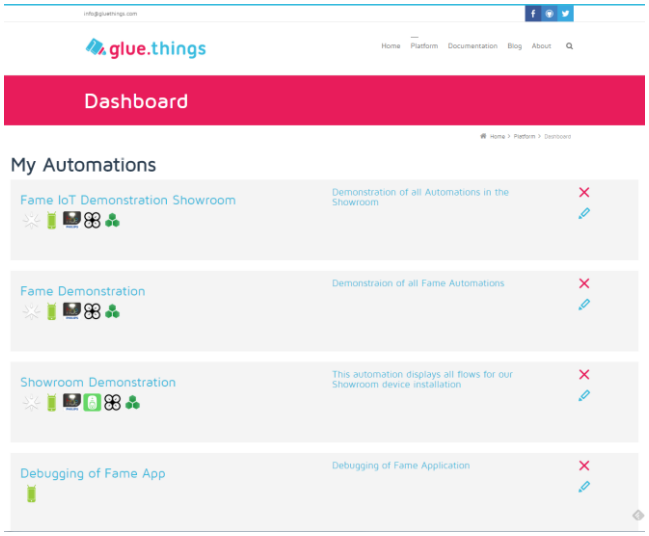

Figure 4: glue.things Mashups

End-users can use a mashup for their own purpose. Mashups of multiple devices, triggers, actions and device settings can be modified by the end-user. Going back to the before mentioned mashup of Beacon, Foursquare and Philips Hue. This mashup can be used for the home environment of a user. He can configure this mashup with his own Philips Hue light bulb, Beacon and his personal Foursquare credentials. glue.things mashup applications are highly flexible and distributable with a set of open Web APIs.

5. CONCLUSION

The paper presented glue.things as WoT hub for TVs, mobiles, wearables and home appliances. With the presented approach, glue.things addresses the current discussion how to integrate all the new and upcoming Web-enabled IoT products with Web technologies. glue.things goes not in the direction and builds a completely new platform. glue.things integrates and adapts popular concepts and Open Source solutions that are widely adapted across the IoT domain. Thereby, different concepts are addressed for real-time communication, device discovery across the Internet, mashups of IoT and IoS and distributed deployment. One particular focus was on the challenges for IoT mashups and their integration with Web services. Especially the combination of Web-enabled IoT devices and intelligent services such as Google Speech To Text offer innovative applications for the IoT market. As next step, glue.things will concentrate on the smart home domain for developing end-user scenarios. In particular, the integration of TV sets and gaming consoles as IoT devices is challenging because of their proprietary APIs.

6. ACKNOWLEDGMENTS

This work has been partially supported by the European Union under project reference number 317862 (COMPOSE) and by Fraunhofer-Institute for Open Communication Systems FOKUS, Berlin, Germany.

7. REFERENCES

[1] Meulen, R. Gartner Says Worldwide Business Intelligence and Analytics Software Market Grew 8 Percent in 2013. In Gartner. http://www.gartner.com/newsroom/id/2723717

[2] Higginbotham, S. Connected Devices and the Internet of Things. In Bloomberg Businessweek. http://www.businessweek.com/articles/2013-01-03/ces-2013-connected-devices-and-the-internet-of-things

[3] Mateo P. Internet of Things Needs Standards in Order to Make Headway. In L'Atelier, the BNP Paribas Group's, 2013. http://www.atelier.net/en/trends/articles/internet-things-needs-standards-order-make-headway_425738

[4] Sundmaeker H., Guillemin P., Woelfflé S. Vision and Challenges for Realising the Internet of Things. In CERP-IoT Cluster of European Research Projects of the Internet of Things, European Commission (2009), 12-13.

[5] Blackstock M., Lea R. Toward interoperability in a web of things. In UbiComp '13 Adjunct Proceedings of the 2013 ACM conference on pervasive and ubiquitous computing adjunct publication, ACM Press (2013), 1565-1574.

[6] Bovet G., Hennebert J. Offering web-of-things connectivity to building networks. In UbiComp '13 Adjunct Proceedings of the 2013 ACM conference on pervasive and ubiquitous computing adjunct publication, ACM Press (2013), 155-1564.

[7] Becker M., Mueller J., Hotho A., Stumme G. A generic platform for ubiquitous and subjective data. In UbiComp '13 Adjunct Proceedings of the 2013 ACM conference on pervasive and ubiquitous computing adjunct publication, ACM Press (2013), 1175-1182.

[8] W3C, Workshop Report. In W3C Workshop on the Web of Things, 2014. http://www.w3.org/2014/02/wot/report.html

[9] Lerner R. M. At the forge: real-time messaging. In Linux Journal, Volume 2013 Issue 225, January 2013, Article No. 5, Belltown Media Houston TX (2013).

[10] Pérez J. L., Villalba A., Carrera D., Larizgoitia I., Trifa V. The COMPOSE API for the internet of things. In WWW Companion '14 Proceedings of the companion publication of the 23rd international conference on World wide web companion, ACM Press (2014), 971-976.

[11] Rumpel A., Meissner K. Requirements-Driven Quality Modeling and Evaluation in Web Mashups In Quality of Information and Communications Technology (QUATIC), 2012, Eighth International Conference on the, IEEE Press (2012), 319-322.

[12] Makki S. K., Sangtani J. Data Mashups & Their Applications in Enterprises. In Internet and Web Applications and Services, 2008, ICIW '08, IEEE Press (2008), 445-450.

[13] Rujiang B., Xiaoyue W. Using Yahoo! Pipes building Information Warehouse of Intelligence. In E-Health Networking, Digital Ecosystems and Technologies (EDT), IEEE Press (2010), 388-391.

[14] Mainetti L., Mighali V., Patrono L., Rametta P. A novel architecture enabling the visual implementation of web of Things applications. In Software, Telecommunications and Computer Networks (SoftCOM), IEEE Press (2013), 1-7.

[15] Piyare R., Park S., Yeong Maeng S., Hyeok Park S. Integrating Wireless Sensor Network into Cloud services for real-time data collection. In ICT Convergence (ICTC), IEEE Press (2013), 752-756.

[16] Blackstock M., Lea R. IoT mashups with the WoTKit. In Internet of Things (IOT), IEEE Press (2012), 159-166.

[17] Piras A., Carboni D., Pintus A. A platform to collect, manage and share heterogeneous sensor data. In Networked Sensing Systems (INSS), IEEE Press (2012), 1-2.

[18] Pulli K., Baksheev A., Kornyakov K., Eruhimov V. Real-time computer vision with OpenCV. In Magazine Communications of the ACM, Volume 55 Issue 6, ACM Press (2012), 61-69.

[19] Stepp M., Miller J. Web programming. In SIGCSE '14 Proceedings of the 45th ACM technical symposium on Computer science education, ACM Press (2014), 737-737.

[20] Dautov R., Kourtesis D., Paraskakis I., Stannett M. Addressing self-management in cloud platforms: a semantic sensor web approach. In HotTopiCS '13 Proceedings of the 2013 international workshop on Hot topics in cloud services, ACM Press (2013), 11-18.