2014 Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing

Midgar: Domain-Specific Language to generate Smart Objects for an Internet of Things platform

Cristian González García, Jordán Pascual Espada, Edward Rolando Núñez-Valdez, Vicente García-Díaz
Department of Computer Science
University of Oviedo
Oviedo, Spain
gonzalezgarciacristian@hotmail.com, pascualjordan@uniovi.es, nunezedward@uniovi.es, garciavicente@uniovi.es

Abstract— A great number of people have Smart Objects in their daily life: Smartphones, cars, tablets, computers, Smart TVs or micro-controllers. Furthermore, these objects have Internet connection. A great number of these objects have various sensors: accelerometers, GPS, pressure, light, temperature, gravity or proximity. The essence of the concept of Internet of Things is to interconnect all these heterogeneous and ubiquitous objects among them. Some Smart Cities or Smart Homes allow a part of this concept. But this has a problem, the software that an object needs to be interconnected with the IoT platform. To create this software users need to develop the application and need developer skills. In this paper we propose a solution to this problem. We propose a graphical Domain-Specific Language for creating the necessary software for interconnecting any object with an IoT platform. For this proposal we will use the IoT Midgar platform.

Internet of Things; Model-Driven Engineering; Domain-Specific Language; Smart Objects;

I. INTRODUCTION

People have a lot of object with Internet connection in their daily life. They have Smartphones [1], [2], tablets, Smart TVs, micro-controllers [3], [4], Smart Tags [5], computer and laptops. Even now, cars have Internet connection and smart operative systems. These object can be interconnected among them and facilitate the daily life of people with automation and notifications. This is the objective of Smart Homes [6], Smart Cities [7], [8] and Smart Earth [9]–[11]. No matter the situation, localization, accessibility to sensor, size, scenario or danger situation [12]. The only requirement is they must have access to the Internet [13].

In the literature, we can found examples of this interconnection. The most common are the freezer [14], the industry [6], Smart Homes [6], [13], automations in universities [5] and environmental monitoring [10]. Other examples were made with micro-controllers and Smartphones [3], [4], [6], [15].

However, for interconnecting objects we must develop one or more programs. In our case, for using IoT, we need an IoT platform. For this reasons, users need to develop an application in their object for interconnecting with an IoT platform. Furthermore, the application must send and receive messages and these messages must be standards with the platform and others objects. In the case the user need interconnect two or more different objects, the user must develop different applications: different code; different values in the objects, for example, in its sensors; maybe a different programming language, for example, if one object is an Android Smartphone (Java), other is their computer (various) and other is an Arduino micro-controller (C). Because this problem and those related to the software crisis [16], users must learn various development skills and they repeat, maybe, the same or similar process in different objects.

Due to the above, the purpose of this proposal is the application of Model-Driven Engineering (MDE) to design a Domain-Specific Language (DSL) to abstract these problems. With this DSL we will implement a graphic editor for creating objects that can be interconnected with an IoT platform. This will allow to create objects for different platforms without development skills. Users only will need knowledge about the problem domain. Furthermore, these objects will be able to interconnect between themselves through the IoT Midgar platform [15].

The remainder of this paper is structured as following: In section 2, we explain the state of the art in where we discuss about Internet of Things, Smart Objects, Model-Driven Engineering, Domain-Specific Languages, the more important current IoT platforms and we present our research on applications that generate objects. In section 3 we explain the proposal. We present the idea about generation of objects for the Internet of Things platform and the proposal architecture. Section 4 contains the conclusions of this proposal. Finally, in the Section 5 we describe some possible options as future work of this proposal.

978-1-4799-4331-9/14 $31.00 © 2014 IEEE
DOI 10.1109/IMIS.2014.48

II. STATE OF THE ART

Each object has different protocols for sending messages. Every application uses their own messages. Each operative system has different language for programming the applications and has different libraries. These condition are the first problem of IoT and we can abstract it in one phrase: the interconnection of heterogeneous objects needs to use the same protocol or standard for working in the same network [15]. The second problem is the application creation for people that not have development skills. To develop an application, people need development skills in different things: platform, protocols, API, programming language and problem domain. Furthermore, people need time, practice and test the application.

To reach this goal, we propose abstract the problem by using Model-Driven Engineering for creating a Domain-Specific Language. With this, people only need knowledge on the problem domain, knowledge to use an editor and people will need less time to develop and test the application. This editor will create the applications for any system that has operative system and applications will send their data for communicating with the platform and other objects.

A. Internet of Things

Internet of Things (IoT) is the interaction among heterogeneous and ubiquitous objects with each other [5], [6], [13], [17], [18].

People expected to be an important technology in the coming years. As the National Intelligence Council of United States says [19], the ONU [17] and other authors in the literature [5]. They expect in the future more Machine to Machine (M2M) interconnections than human-human interconnections. These objects will interact among themselves and they will perform certain actions when certain conditions are met [20]. Nonetheless, IoT require three things for it to exist as said [5]: integrated intelligence, connectivity and interaction. For these, RFID, NFC, sensors and WSAN are the IoT base "are the atomic components that will merge the real world with the digital world" [17].

These objects can obtain the environment data for processing in a Smart Object, for example, a Smartphone, or for sending to the server. But, we need to create the application for these objects. The application must perform various things. First, the application must obtain the data that the object had read from the environment. After, it must interconnect the object with an IoT platform to send the data. During this process, the application needs to hear requests for the platform with data of other objects that they interconnect with the object of this application

B. Smart Objects

Smart Objects are identifiable physical elements during their useful life, that interact with other objects, the environment and can act intelligently under certain conditions through an autonomous behaviour [6], [17]. Examples of Smart Object are the Smartphones, Smart TVs, tablets, micro-controllers, laptops, some cars and some freezer prototypes.

C. Model-Driven Engineering

Model-Driven Engineering or MDE [21] appeared to solve the problems of Software Crisis [16], [22]. These problem can have a solution with MDE because MDE automates processes steps in the software production [23]. With this we obtain reduced the design complexity and the implementation. With this we obtain a much more reliable software and with more sophisticated functionalities [23].

In this proposal, we will obtain an abstraction over the General Purpose Languages (GPL) (C and Java) with the graphical editor because we will apply MDE for obtain the models of this problem. Thus, users will be able to create their applications with platform independence, without errors and with more quality.

D. Domain-Specific Languages

A Domain-Specific Language (DSL) is a specific solution to resolve a specific problem of a concrete domain. A DSL increases the productivity, reduces bugs, is portable, has an easy maintenance and allows reutilization for different purposes [24], [25]. However, a DSL has a worse efficiency than native codification and has a higher difficulty to create it because the developers need an abstraction of the specific problem [24], [26].

In this proposal we will create a Domain-Specific Language for a graphical editor. This DSL will allow create applications for heterogeneous objects. With this applications, these objects will be able to connect it with an IoT platform, in our prototype, with Midgar. With this idea, users will be able to create these applications without development skills.

E. Internet of Things platforms

At present, we can encounter different Internet of Things platforms. These platforms allowed interconnect heterogeneous objects with others: Arduino micro-controllers, Smartphones, cars or computers. We can divide the actual platform in for groups: business platforms, research platforms, platforms in beta state and Open Source platforms.

Examples of business platforms are Xively [27], Exosite [28], SensorCloud [29] and Etherios [30]. In research platforms group we can found three IoT platforms: Midgar [15], Paraimpu [31], QuadraSpace [32] and SIoT [33]. The third group is the most complicated because some platforms need an approved account or an invitation to access. In this group are ThingSpeak [34], Sensorpedia [35], [36], SenseWeb [37], [38], Evrythng [39] and Open.Sen.se [40]. The last group has the Open Source platform, this is the case of Nimbits [41].

These are great IoT platforms but in some case users need develop code in the platform to interconnect various objects using the platform. The other problem is that users need develop the application in the objects to interconnect with

these platforms. Midgar have a web editor for resolve the first problem. For the second problem, we propose a solution in this paper.

F. Applications that generate software to manage Smart Objects

For creating an application, the user needs developer skills. However, there exist different applications to facilitate this for different platforms. This application can generate application of specific domains. For example, with Minibloq [42], users can create an application for using an Arduino micro-controller. Another example that exists for Android is AppsGeyser [43]. Multiplatform examples are AppsBuilder [44] and Infinite Monkeys [45] that support Android, iOS and HTML5.

Minibloq is a graphical editor to generate processes for Arduino micro-controller. It abstracted the fundamental programming structures: condition, loops, delay and variables. These were combined with the abstraction of Arduino: input/output pins, display, buzzer and motors. With this abstraction users can create processes to execute inside Arduino micro-controller.

AppsGeyser allows to select the type of the application that you desire: website, browser, page, various games and more. Depending on your election, the application presents diverse options to create it.

In the case of AppsBuilder, users can select various characteristics that desire in their application: social networks, sound, video, news, images, text and many more. The next step is the selection of theme between 83 and the menu button design. Once the user has selected the characteristics, AppsBuilder shows these options to personalise these characteristics. The user can select the icon, the state, the header style, the navigation menu, the disposal cell, the menu style, the look & feel, the configuration and the user management options. During the process the user can review the application in mobile site, iPhone, iPad and Android.

The first step in Infinite Monkey is to introduce the application name and your personal data. After, you can choose to start between four templates or an empty application. Over the empty template, users can drag and drop different functions, for example, calendar, blog, social networks and many others. During this process, user can preview the current application.

Minibloq is a good program, but only allows to create processes that work inside Arduino. We will create a graphical editor to create processes to send and receive data for interconnecting with an IoT platform. This allows read environment data and allows perform action according the data that it received of others objects through IoT platform. Users will must select the input/output pin number and the object that have in the pin, for example, "Motor in pin 5" or "Led in pin 10".

AppsGeyser, AppsBuilder and Infinite Monkeys are good platforms to create applications for mobile or webs, but these do not support for sensors or for send and receive data and perform an action in according these data. For these reason, we will create and graphical editor to support these functionalities.

For this, we will create a graphical editor that support the creation of application for some devices. Users will be able select the device (Arduino or Android) and the device type (Arduino Uno, Arduino Mega, Android Nexus 4 or Android Samsung Galaxy S3). Then, the graphical editor will help in according the decisions and will show the supported sensors and material. After users define their application, the graphical editor will create it. Then, users only will deploy their application in their devices and start it. Automatically, the application will work with IoT platform.

III. Proposal

Midgar is an Internet of Things platform specifically developed to investigate the problems of these platforms and the interconnection of objects. Midgar has a graphical editor to generate applications to interconnect already created between them. The problem is that users must develop these applications. For this, in this proposal, we try to find a solution for application development for objects by users without development skills. In this section, we will describe the idea.

A. Generation of objects for the Internet of Things platform

This DSL will allow create the necessary software for an object. This software will allow that objects to connect to Midgar IoT platform and send their data and receive the data from others object. Through a graphical editor, users will be able to create the object they want for their platform. Users will choose sensors that their Smartphones have or elements that they plug in their micro-controller.

The graphical editor will create objects for different platforms, for example, Smartphones, computers or micro-controllers. These objects will interconnect with the platform without requiring users to develop anything and the objects will send the data that users have chosen. The only things users need is the knowledge about de domain and the applications they want to use.

B. Proposal's architecture

The system's architecture will be divided in three layers: **Object Definition** (Figure 1), **Service Object Generation** (Figure 2) and **Objects** (Figure 3). Each one is a process within the global set of the infrastructure.

The first layer (Figure 1), **Object Definition**, is the layer where users interact with the graphical editor to define their object (Figure 1-1). In the graphical editor, users will be able to selected the sensor that they want in their smartphone applications in the case of their smartphone has these sensors. Users must choose the smartphone type. Then, the graphical editor shows the supported sensors in this model. In the case of micro-controllers, users must the

micro-controller model and they will elect components and their pin in the micro-controller.
Once the user completes the definition of the object, the user will select the generation. In the step 2 (Figure 1), the graphical editor will generate the DSL with all information for making the object application that the user wanted. The **DSL** will contain all the information in XML format and the graphical editor will send it to the second layer: **Service Object Generation** (Figure 2).

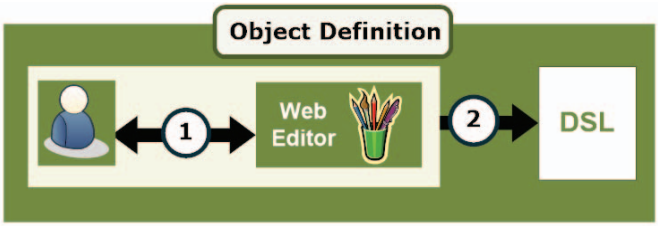

Figure 1 Object Definition

The **Service Object Generation** (Figure 2-1) will receive this **DSL**. The Service Object Generation will process the DSL to create the object applications. The **Processor** will read the XML and it will create the tree with all sentences that the user had described. Later, in the step 2 (Figure 2), **Midgar** will **create and compile** the application. Applications will be generated in different language (Figure 2-3). This depends on the final platform application. For example, it will generate Java for Android, C for Arduino, C# for Windows Phone or Objective-C for iPhone (Figure 3).

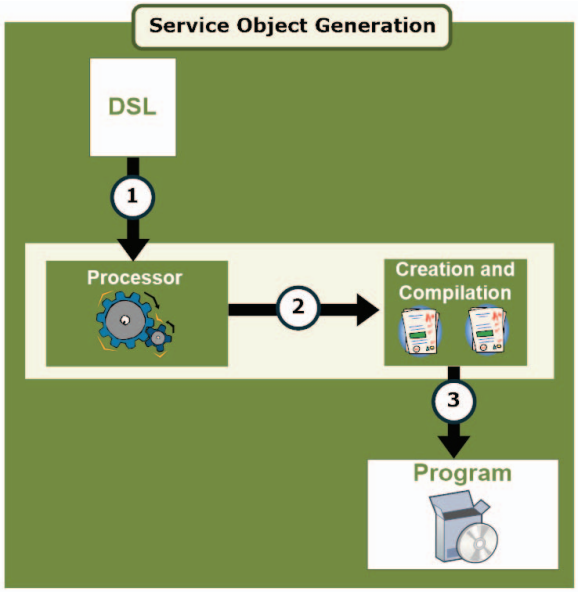

Figure 2 Service Object Generation

Applications generated by the graphical editor will allow the connection between objects and Midgar platform using JSON format for sending data. We will use JSON instead XML because JSON is less verbose and it consumes less data when it is sent through the Internet.

The data sent to Midgar will contain the registration Id and the sensor data that users had selected in their platforms or a message for consulting if the platform has messages for this object. When the platform will receive the data, it will send the check message or, in case of existing, the stack messages of other objects for this object.

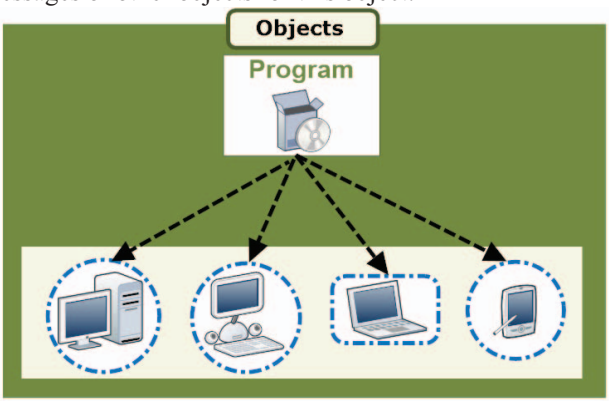

Figure 3 Objects

IV. Conclusions

We present in this paper a proposal for creating the necessary object applications for interconnecting with an IoT platform. With this proposal, users will be able to create the necessary application for their object without developing it. We want that any user without developer skills can connect their object with an IoT platform. In our case, we use a previous IoT research platform: Midgar.
Users only will need the basic knowledge for using a graphical editor: users will need to use a mouse and keyboard and drag and drop elements in browser and how to select and write in textboxes. Then, users will be able to have their application in their devices and only will need to start it.
With this solution, we will search that any user can have their application for using IoT and objects with an easy way.

V. Future Work

Smart Objects are the present and Internet of Things is the future. The Midgar Project proposes some solution to facilitate the interaction between these technologies and users. The next steps for the future works in Midgar are:

- **Improve the Domain-Specific Language and graphic editor to put artificial intelligence in the Smart Objects**: Create a graphic editor and a DSL, similar to the one shown in this proposal, but it offers different options to insert artificial intelligence and conditions in the generated applications.
- **Security and privacy between objects and IoT network**: Study of the security and privacy methods that any protocol to use for sending personal data should. This is necessary to prevent

the violation of data transferred from and to the objects.

- **Performance of Smart Objects:** Research over the performance of applications in the different object for improving the performance the battery drain because the sensor reading and the continuous messages with the IoT platform.

REFERENCES

[1] F. Telefónica, "La Sociedad de la Información en España 2012," 2013.

[2] Y. M. Kang, C. Cho, and S. Lee, "Analysis of factors affecting the adoption of smartphones," in *First International Technology Management Conference*, 2011, pp. 919–925.

[3] A. Piras, D. Carboni, A. Pintus, and D. M. T. Features, "A Platform to Collect , Manage and Share Heterogeneous Sensor Data," in *Networked Sensing Systems (INSS)*, 2012, pp. 1–2.

[4] T. Yamanoue, K. Oda, and K. Shimozono, "A M2M System Using Arduino, Android and Wiki Software," in *2012 IIAI International Conference on Advanced Applied Informatics*, 2012, pp. 123–128.

[5] L. Tan, "Future internet: The Internet of Things," *2010 3rd Int. Conf. Adv. Comput. Theory Eng.*, pp. V5–376–V5–380, Aug. 2010.

[6] K. A. Hribernik, Z. Ghrairi, C. Hans, and K. Thoben, "Co-creating the Internet of Things - First Experiences in the Participatory Design of Intelligent Products with Arduino," 2011, no. Ice, pp. 1–9.

[7] M. Hasan, E. Hossain, and D. Niyato, "Random access for machine-to-machine communication in LTE-advanced networks: issues and approaches," *IEEE Commun. Mag.*, vol. 51, no. 6, pp. 86–93, Jun. 2013.

[8] A. Martinez-Balleste, P. Perez-martinez, and A. Solanas, "The pursuit of citizens' privacy: a privacy-aware smart city is possible," *IEEE Commun. Mag.*, vol. 51, no. 6, pp. 136–141, Jun. 2013.

[9] K. Aberer, "Smart Earth: From Pervasive Observation to Trusted Information," in *2007 International Conference on Mobile Data Management*, 2007, pp. 3–7.

[10] B. M. Howe, Y. Chao, P. Arabshahi, S. Roy, T. McGinnis, and A. Gray, "A Smart Sensor Web for Ocean Observation: Fixed and Mobile Platforms, Integrated Acoustics, Satellites and Predictive Modeling," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 3, no. 4, pp. 507–521, Dec. 2010.

[11] S. Wang, "Spatial data mining under Smart Earth," in *2011 IEEE International Conference on Granular Computing*, 2011, pp. 717–722.

[12] I. W. S. Y. S. E. C. Akyildiz, "A survey on sensor networks," *IEEE Commun. Mag.*, no. August, pp. 102–114, 2002.

[13] G. M. Lee and J. Y. Kim, "Ubiquitous networking application: Energy saving using smart objects in a home," *2012 Int. Conf. ICT Converg.*, pp. 299–300, Oct. 2012.

[14] S. Luo, H. Xia, Y. Gao, J. S. Jin, and R. Athauda, "Smart Fridges with Multimedia Capability for Better Nutrition and Health," in *Ubiquitous Multimedia Computing, 2008. UMC '08. International Symposium on*, 2008, pp. 39–44.

[15] C. G. García, C. P. García-Bustelo, J. P. Espada, and G. Cueva-Fernandez, "Midgar: Generation of heterogeneous objects interconnecting applications. A Domain Specific Language proposal for Internet of Things scenarios," *Comput. Networks*, Feb. 2014.

[16] E. Dijkstra, "The humble programmer," *Commun. ACM*, vol. 15, pp. 859–866, 1972.

[17] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Networks*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[18] N. Bari, G. Mani, and S. Berkovich, "Internet of Things as a Methodological Concept," in *2013 Fourth International Conference on Computing for Geospatial Research and Application*, 2013, pp. 48–55.

[19] The US National Intelligence Council, "Six Technologies with Potential Impacts on US Interests out to 2025," 2008.

[20] L. Atzori, A. Iera, and G. Morabito, "From 'smart objects' to 'social objects': The next evolutionary step of the internet of things," *IEEE Commun. Mag.*, vol. 52, no. 1, pp. 97–105, Jan. 2014.

[21] S. Kent, "Model Driven Engineering," *Comput. Comput. Soc.*, vol. 2335, no. 2, pp. 286–298, 2002.

[22] E. González, H. Fernández, and V. Díaz, "General purpose MDE tools," *IJIMAI*, vol. 1, 2008.

[23] B. Selic, "MDA manifestations," *Eur. J. Informatics Prof.*, vol. IX, no. 2, pp. 11–16, 2008.

[24] A. Van Deursen, P. Klint, and J. Visser, "Domain-specific languages: an annotated bibliography," *ACM Sigplan Not.*, 2000.

[25] A. Van Deursen, "Domain-specific languages versus object-oriented frameworks: A financial engineering case study," 1997.

[26] A. B. Reed, V. Halpern, and J. E. Starr, "Little languages: Little maintenance?," Apr. 1996.

[27] LogMeIn, "Xively," *https://xively.com/*, 2013. [Online]. Available: https://xively.com/. [Accessed: 21-Feb-2014].

[28] Exosite, “Exosite,” *http://exosite.com/*, 2013. [Online]. Available: http://exosite.com/. [Accessed: 21-Feb-2014].

[29] LORD MicroStrain, “Sensor Cloud,” *http://www.sensorcloud.com/*. [Online]. Available: http://www.sensorcloud.com/. [Accessed: 21-Feb-2014].

[30] Etherios, “Etherios,” *http://www.etherios.com/*, 2008. [Online]. Available: http://www.etherios.com/. [Accessed: 21-Feb-2014].

[31] A. Pintus, D. Carboni, and A. Piras, “Paraimpu,” *http://paraimpu.crs4.it/*, 2012. [Online]. Available: http://paraimpu.crs4.it/. [Accessed: 21-Feb-2014].

[32] “QuadraSpace,” *http://www.quadraspace.org/*, 2010. [Online]. Available: http://www.quadraspace.org/. [Accessed: 21-Feb-2014].

[33] Department of Electrical and Electronic Engineering (University of Cagliari), “SIoT,” *http://platform.social-iot.org/*, 2012. [Online]. Available: http://platform.social-iot.org/. [Accessed: 17-Dec-2013].

[34] IoBridge, “Thingspeak,” *http://www.thingspeak.com*, 2013. [Online]. Available: http://www.thingspeak.com. [Accessed: 21-Feb-2014].

[35] Oak Ridge National Laboratory, “Sensorpedia,” *http://www.sensorpedia.com/*, 2009. [Online]. Available: http://www.sensorpedia.com/. [Accessed: 21-Feb-2014].

[36] B. L. Gorman, D. R. Resseguie, and C. Tomkins-Tinch, “Sensorpedia: Information sharing across incompatible sensor systems,” *2009 Int. Symp. Collab. Technol. Syst.*, pp. 448–454, 2009.

[37] Microsoft, “SenseWeb,” *http://research.microsoft.com/en-us/projects/senseweb/*, 2008. [Online]. Available: http://research.microsoft.com/en-us/projects/senseweb/. [Accessed: 21-Feb-2014].

[38] A. Kansal, S. Nath, J. Liu, and W. I. Grosky, “SenseWeb : An Infrastructure for Shared Sensing,” *IEEE Multimed.*, vol. 14, no. 4, pp. 8–13, 2007.

[39] EVRYTHNG, “EVRYTHNG,” *https://www.evrythng.com/*, 2012. [Online]. Available: https://www.evrythng.com/. [Accessed: 21-Feb-2014].

[40] Sen.se, “Open.Sen.se,” *http://open.sen.se/*, 2011. [Online]. Available: http://open.sen.se/. [Accessed: 21-Feb-2014].

[41] I. Nimbits, “Nimbits,” *http://www.nimbits.com/*. [Online]. Available: http://www.nimbits.com/. [Accessed: 21-Feb-2014].

[42] “Minibloq.” [Online]. Available: http://blog.minibloq.org/.

[43] “AppsGeyser.” [Online]. Available: http://www.appsgeyser.com.

[44] “AppsBuilder.” [Online]. Available: http://www.apps-builder.com/.

[45] “Infinite Monkeys.” [Online]. Available: http://www.infinitemonkeys.mobi/.