

THINGER: WEB-ORIENTED PLATFORM FOR INTERACTION BETWEEN SMART THINGS

N. Pilipenko, L. Voskov

Moscow State Institute of Electronics and Mathematics of the National
Research University Higher School of Economics, Russia
pilipenko-na@yandex.ru, voskov@yandex.ru

In the present article we give an overview of some existing data brokerage platforms for Internet of Things, and show their limitations. We finally propose our solution Thinger - a Web-Oriented data brokerage platform.

THINGER: WEB-ОРИЕНТИРОВАННАЯ ПЛАТФОРМА ДЛЯ ВЗАИМОДЕЙСТВИЯ «УМНЫХ ВЕЩЕЙ»

Н.А. Пилипенко, Л.С. Восков

Московский институт электроники и математики
Национального исследовательского университета «Высшая школа
экономики», Россия
pilipenko-na@yandex.ru, voskov@yandex.ru

В данной работе производится краткий обзор существующих посреднических платформ данных для Интернета Вещей. Выявляются их основные недостатки. И предлагается реализация Web-ориентированной посреднической платформы Thinger.

1. Введение

Интернет Вещей (Internet of Things) [1] – это единая сеть, объединяющая в себе как виртуальные объекты, так и объекты реального мира. «Умная Вещь» (Интернет-вещь) – любой материальный или виртуальный объект, подключенный к сети Интернет.

Согласно исследованиям компании Cisco, сейчас число устройств, подключенных к сети Интернет, превысило 13 миллиардов, и в среднем каждые последующие 5 лет их количество будет удваиваться. Сегодня массово стали появляться личные «Умные вещи» для досуга, энергосбережения, наблюдения за состоянием здоровья и пр..

Большинство подобных устройств объединены в закрытые экосистемы, которые не могут взаимодействовать и легко использоваться совместно. Причиной этого является отсутствие единых стандартов и наличие множества протоколов и технологий. Переход на Web-ориентированную платформу с применением открытых Web-стандартов позволит использовать единые форматы

представления данных и протоколы для организации взаимодействия и совместного использования устройств.

Целью настоящей работы является разработка Web-ориентированной платформы для взаимодействия интернет-вещей в реальном времени.

Для достижения поставленной цели необходимо решение следующих задач:

- аналитический обзор существующих решений;
- исследование требований, методов и алгоритмов решения поставленной задачи;
- разработка архитектуры и прототипа посреднической платформы данных.

Объектом исследования является концепция посреднической платформы для Интернета Вещей. Предмет исследования состоит в анализе структуры объекта, требований, методов, алгоритмов и средств его реализации.

2. Web Вещей

Следующим логическим шагом развития Интернета Вещей является переход на более высокий уровень – прикладной. Web Вещей (Web of Things) [2] предполагает интеграцию «Умных Вещей» на Web уровень (уровень приложений).

В основе данного подхода лежит WebAPI, использующий REST (Representational State Transfer)-архитектурный стиль (RESTful API) [2]. WebAPI обеспечивает возможность связи различных ресурсов и доступа к ним, используя стандартные возможности HTTP такие как аутентификация, шифрование, кэширование, сжатие и пр.

Отличительные особенности REST-архитектурного стиля:

1. Модель приложений, ориентированная на данные. "Всё", что предоставляет услуги, становится ресурсом. Любой ресурс имеет свой идентификатор (URI).

2. Доступ к ресурсам осуществляется через единый интерфейс. Допустимы 4 операции над ресурсом. GET – получение данных. PUT – обновление данных. DELETE – удаление ресурса. POST – создание нового ресурса.

3. Использование самоописываемых форматов данных. Для общения устройств могут быть использованы XML, JSON, CSV и KML форматы данных.

3. Посреднические платформы данных

Основной целью использования посреднических платформ данных является упрощение поиска, контроля, визуализации и обмена данными с различными «Вещами». В основе данного подхода лежит централизованное хранилище данных.

Новейшие M2M и IoT системы, такие как AirVantage, ThingWorx и Axeda предоставляют набор инструментов для создания частных решений. Они не нацелены на открытое использование.

Такие системы как Xively, Open Sen.se, ThingSpeak, Paraimpu, SenseWeb и другие больше нацелены на удовлетворение потребностей Web-разработчиков. Они предоставляют набор инструментов для обработки, хранения и визуализации данных, могут быть интегрированы с социальными сетями и прочими системами.

4. Существующие проблемы

Однако существует ряд проблем, связанных с использованием таких платформ:

- необходимость периодического опроса сервера или приборов для приема/передачи данных приводит к формированию избыточной информации;
- использование протокола HTTP повышает требования к производительности устройств и сети по сравнению со специализированными протоколами;
- помимо передачи периодически изменяющихся данных некоторые приборы могут передавать и потоковые данные (например, видео).

5. Предлагаемое решение

Thinger является посреднической программной платформой данных для взаимодействия интернет-вещей и включает в себя несколько базовых компонент.

1. **«Вещи» (Things).** Каждая вещь имеет уникальный идентификатор (например, <http://thinger.ru/api/things/mymeteo>). Вещи в системе представлены 2-мя составляющими: базовая информация, которую имеют все устройства (идентификатор, наименование, описание и пр.), и динамическая информация – набор полей, описывающих конкретные устройства (различные показатели и данные, получаемые от устройств). Например, показания метеостанции могут иметь следующий вид:

```
{"Name": "Temperature", "Value": "23", "Updated": "2013-04-06T12:46:47.307"}
```

2. **Поставщики данных (Data Providers)** позволяют подключать уже существующие устройства от различных производителей, предоставляющих RESTful API, используя открытые протоколы авторизации OAuth 1.0 и OAuth 2.0.

3. **Обработчики данных (Data Handlers)** позволяют создавать настраиваемые сценарии обработки данных и взаимодействия устройств средствами языка JavaScript, которые выполняются сразу же при изменении конкретных показателей устройств.

4. **Социальные провайдеры (Social Providers)** предназначены для интеграции устройств с социальными сетями.

5. **Виджеты (Widgets)** – визуальные представления устройств. Виджеты могут быть информационными (служат для визуализации данных) и управляющими (могут передавать команды и данные устройствам). Каждое устройство в системе может иметь множество виджетов. Один виджет может представлять сразу несколько устройств.

6. **Подсистема доступа (Access Subsystem)** предназначена для организации доступа к устройствам. Пользователи системы могут предоставлять и получать доступ к устройствам других пользователей.

Организация поддержки, в дальнейшем, в системе протокола MQTT [3] и технологии WebSocket [4] позволит:

- реализовать асинхронное взаимодействие между устройствами и сервером по модели публикация/подписка, исключив при этом необходимости в постоянном опросе;
- реализовать возможность передачи потоковых данных;
- снизить нагрузку на конечные устройства и сеть в целом.

6. Архитектура платформы

Thingier – это ASP.NET приложение, развернутое на платформе Windows Azure. Основные составляющие архитектуры Thingier [5]:

1. RESTful Web-сервис - предоставляет единый интерфейс для работы с данными.

2. Web-сайт – предназначен для визуализации данных и предоставляет расширенный инструментарий для работы с сервисом.

При дальнейшем развитии в архитектуру будут включены:

1. MQTT-сервер – для передачи данных по протоколу MQTT.
2. HTTP-MQTT мост – для трансляции данных между сервисом и MQTT-сервером.
3. WebSocket-сервер – для передачи данных в реальном времени между Web-приложениями.

Заключение

В рамках данной работы был выполнен аналитический обзор существующих посреднических платформ данных, определены их основные проблемы и исследованы технологии, которые могут быть использованы для решения этих проблем.

В результате данной работы была разработана Web-ориентированная посредническая платформа данных Thingier, которая позволяет получать доступ к различным «Умным устройствам» через единый ресурс, создавать сценарии взаимодействия и определять графические представления этих устройств. Платформа позволяет организовывать взаимодействие с устройствами через социальные сети. Thingier может взаимодействовать как с разработанными под

нее устройствами, так и со сторонними устройства различных производителей, предоставляющих RESTful API.

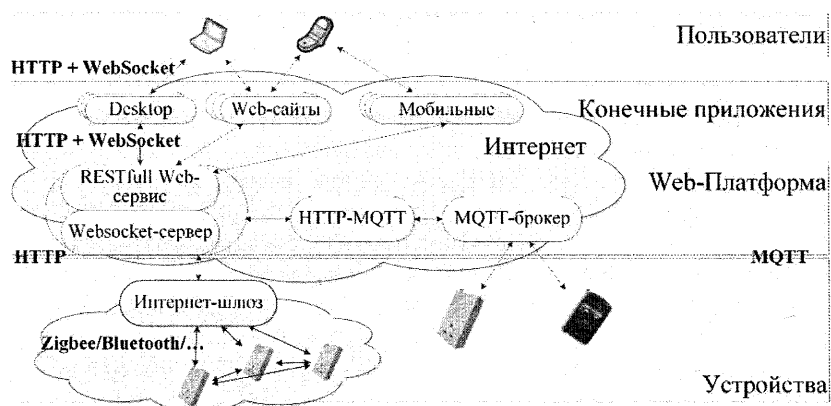


Рисунок 1. Пример работы платформы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК:

1. Восков Л.С. "Интернет вещей // "Новые информационные технологии". Тезисы докладов XX Международной студенческой конференции-школы-семинара, М.: МИЭМ, 2012 (89-94).
2. D. Guinard, V. Trifa, E. Wilde. A resource oriented architecture for the Web of Things // Internet of Things (IoT), 2010 (Tokyo, Japan).
3. Valerie Lampkin, Weng Tat Leong, Leonardo Olivera, Sweta Rawat, Nagesh Subrahmanyam, Rong Xiang "Building Smarter Planet Solutions with MQTT and IBM WebSphere MQ Telemetry", 2012.
4. Черновик спецификации W3C к протоколу WebSocket [Электронный ресурс]. URL: <http://dev.w3.org/html5/websockets/> (дата обращения 04.2013).
5. Пилипенко Н.А. Архитектура "Web-ориентированной платформы для взаимодействия интернет-вещей" // Научно-техническая конференция студентов, аспирантов и молодых специалистов МИЭМ НИУ ВШЭ. Тезисы докладов. - М. ~: МИЭМ, 2013 (136-138).