

Система для реализации сценариев интернета вещей на основе облачных технологий

08, август 2015

**Трибушков Д. Э.^{1,*}, Кочетков А. А.¹,
Попов А. Ю.¹**

УДК: 004.72+004.771

¹Россия, МГТУ им. Н.Э. Баумана

^{*}dk-team@list.ru

Требования к облачной платформе

Интернет вещей это концепция, развивающая идеи “умных технологий” (умного дома, умного города и пр.), при которой многочисленные электронные приборы, системы и сенсоры имеют возможность обмениваться информацией друг с другом и управляются через Интернет [1,2]. Концепция Интернета вещей предполагает включение в каждое бытовое устройство, будь то пылесос, холодильник или стиральная машина, модуль подключения к интернету с возможностью взаимодействия с домашним компьютером или смартфоном владельца дома.

Подобное решение может быть реализовано как облачное приложение, предоставляющее сервис по управлению различными аспектами жизнеобеспечения домов, офисов, посёлков и т.д. При этом оно не является подменой существующих решений, а фактически, является интеграционным решением, позволяющим использовать существующие сторонние коммерческие и открытые решения управления домом/офисом. Цель такой интеграции – обеспечить единый интерфейс для доступа к любой сторонней системе и единую среду для конфигурирования процессов управления объектами и мониторинга за их состоянием, а также сделать возможным предоставлять умный дом/офис как коммерческую услугу. Важным результатом такого проекта может быть внедрение результатов в учебный процесс. В частности, опираясь на существующую облачную инфраструктуру МГТУ им. Н.Э.Баумана [3], студенты и научно-исследовательские коллективы могут вести разработку перспективных проектов. На рисунке 1 показан пример взаимодействие облачной платформы с управляемыми объектами (умными домами/офисами/посёлками). Таким образом, само решение не взаимодействует с датчиками и исполнительными механизмами, установленными на объектах, а является единым «фасадом» для управления объектами через сторонние системы.



Рис. 1. Взаимодействие облачной платформы с управляемыми объектами.

Сервис-ориентированная архитектура (Service-oriented Architecture, SOA) - модульный подход к разработке программного обеспечения, основанный на использовании распределенных, слабо связанных, заменяемых компонентов, оснащенных интерфейсами для взаимодействия по стандартизированным протоколам.

Основная специфика данного решения, влияющая на архитектуру, определяется следующими факторами:

- Облачная платформа должна позволять масштабировать все сервисы для управления большим количеством управляемых объектов – до десятков и сотен тысяч без влияния на производительность и доступность решения;
- Решение должно предоставлять средства и интерфейсы для интеграции со сторонними системами, реализующими те или функции управления домами;
- Решение должно обеспечивать развёртывание и исполнение собственных сервисов на облачной платформе для обеспечения всех преимуществ последней по масштабированию, безопасности, мониторингу;
- Иерархический и многоаспектный характер управляемых объектов. С одной стороны, мы имеем пространственную иерархию: комната, этаж, дом, посёлок и т.п. С другой стороны, есть такие аспекты функционирования, как безопасность, электрообеспечение, климат, бытовое оборудование и т.д.

Таким образом, архитектура облачного решения должна обладать следующими качествами:

1. Предоставлять открытые интерфейсы и спецификации для интеграции со сторонними системами;
2. Обеспечивать масштабирование сервисов и данных, прежде всего горизонтальное масштабирование;

3. Поддерживать различные представления управляемых и возможность их добавления;

Выбор подхода и платформы для реализации облачной инфраструктуры

Основой управления объектами является так называемый сценарный подход, когда в зависимости от конкретных условий исполняются сценарии по управлению тем или иным аспектом. Например, при превышении некоторой заданной температуры могут включаться кондиционеры, открываться окна, выключаться отопление. Каждый такой сценарий зависит не только от внутренней температуры, но и от внешней, наличия сезонного отопления и множества других факторов.

Со сторонними системами взаимодействие может осуществляться как на локальном уровне отдельного управляемого объекта (дом, офис), так и на уровне облачной платформы. При этом локальный уровень интеграции с архитектурной точки зрения может отсутствовать без ущерба для функциональности решения. Уровень облачной платформы обязателен, к примеру, для интеграции с системами видеонаблюдения (ВН).

Это связано с тем, что большинство систем ВН требуют достаточно мощный сервер для обработки, хранения и раздачи видеопотока. С точки зрения стоимости решения и стоимости его поддержки на этапе эксплуатации для клиента установка такого сервера локально может оказаться нецелесообразным. Кроме того, часть сценариев управления домом или офисом невозможна без участия пользователя, подключение которого к сервису представляется целесообразным через облако, обеспечивающим конфигурирование, мониторинг и т.д. Всё это требует значительных ресурсов, которые локально устанавливать нецелесообразно.

С другой стороны, локальная интеграция более надёжна в случае разрыва связей между облаком и управляемым объектом. Для устранения таких рисков облачная платформа должна обеспечивать горячее резервирование каналов связи.

Задача выбора базовой облачной платформы заключается в обеспечении высокой производительности и масштабируемости в соответствии со спецификой нашего решения. По существу, есть два основных высокоуровневых паттерна построения облачной архитектуры: *multitenancy* и *multiinstance*. В первом случае единый экземпляр приложения обслуживает множество объектов, в случае *multiinstance* - для каждого управляемого объекта имеется отдельная копия приложения.

Выбор того или иного подхода очень важен, т.к. предполагается потенциально очень большое число управляемых объектов: десятки, сотни тысяч.

Оба подхода имеют свои преимущества и недостатки. *Multitenancy* архитектура более простая с точки зрения настройки и эксплуатации, проще резервируется, требует меньше вычислительных ресурсов, а значит и дешевле. С другой стороны, *multiinstance* архитектура более производительна, легче масштабируется. Учитывая, что в нашем случае мы обеспечили возможность построения *multitenancy* архитектуры, то предлагается гибридная схема, которая позволит использовать преимущества каждого подхода.

В основу архитектуры ядра облака положена event driven архитектура (EDA) для внутренней реализации и SOA для интеграции с общими облачными сервисами. Это означает, что связующим ядром является Event Bus – шина событий. Все взаимодействия проходят через неё. Исключением является лишь система видеонаблюдения CCTV, в которую поступает видеопоток и которая, в свою очередь, раздаёт видеопоток пользователям решения. Реализация Event Bus зависит от общего фреймворка, положенного в основу архитектуры ядра облака.

Преимуществом сервис-ориентированной архитектуры является то, что каждый компонент представляет собой отдельный сервис. Это означает, что сервисы могут располагаться на разных машинах и в разных точках мира. Также не составит труда заменить SOA-компонент на более новый, если предыдущий устарел.

Рассмотрим простейший случай одного управляемого объекта без масштабирования. В качестве объекта будем рассматривать отдельный этаж, зону, офис или дом.

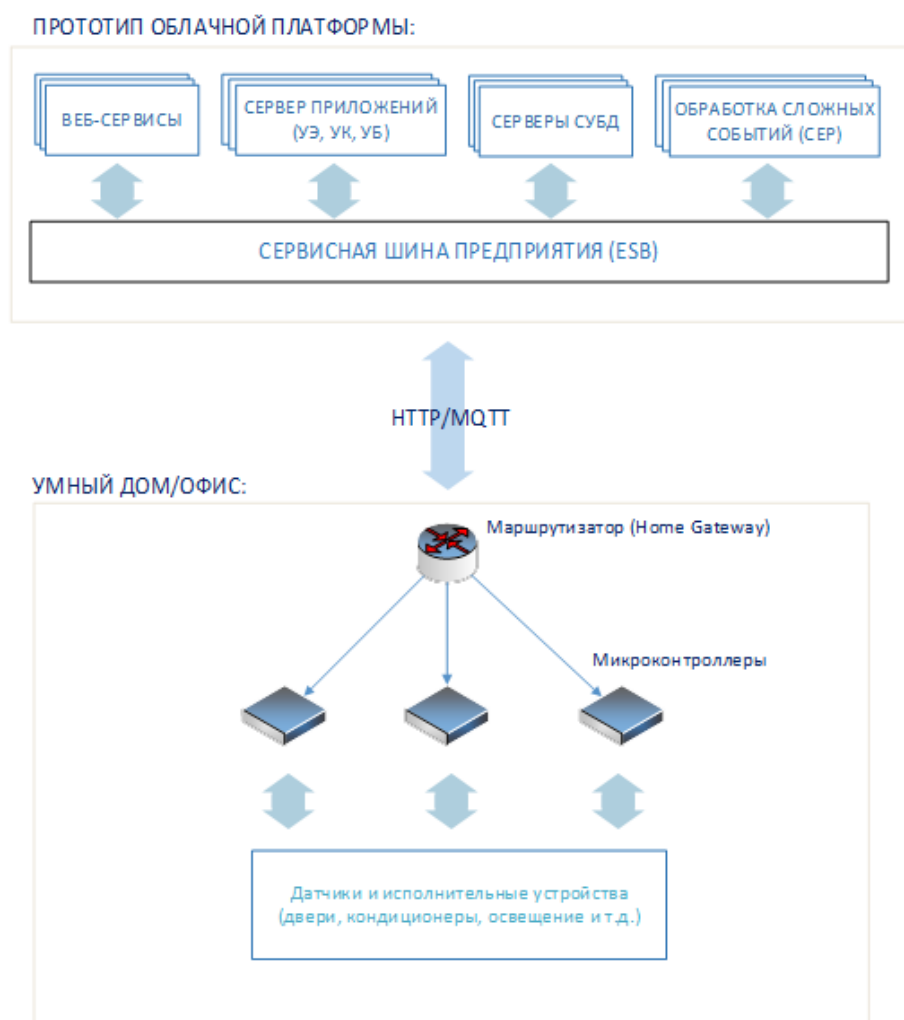


Рис. 2. Архитектура с одним управляемым объектом.

На рисунке 2 отображена общая архитектура решения для случая одного управляемого объекта.

Управление климатом (УК), управление безопасностью (УБ), управление электроэнергией (УЭ) – облачные приложения сторонних вендоров, интегрированные в данную облачную платформу.

Обработчик сложных событий (Complex Event Processor) – заключается в обработке множества событий, происходящих на всех уровнях системы, при этом идентифицируются наиболее существенные события из множества событий, анализируется их влияние и в режиме реального времени предпринимаются соответствующие действия.

Выводы

В результате данного исследования был описан сервис-ориентированный подход к разработке облачной платформы. Были проанализированы необходимые компоненты облака, их принципы работы и область применения. Разработаны структурная схема ядра облачной платформы, схема взаимодействия облачного сервиса и клиента. По итогам работы была представлена архитектура облачной платформы, предназначенной для размещения облачных приложений в сфере «Интернет вещей».

Список литературы

1. A Reference Architecture for the Internet of Things. // Сайт компании WSO2. Режим доступа: <http://wso2.com/whitepapers/a-reference-architecture-for-the-internet-of-things/> (дата обращения: 17.05.2015).
2. Device democracy. Saving the future of the Internet of Things. (2014) IBM Global Business Services Executive Report. // IBM Institute for Business Value. / IBM Corporation. Режим доступа: <http://public.dhe.ibm.com/common/ssi/ecm/gb/en/gbe03620usen/GBE03620USEN.PDF> (дата обращения: 17.05.2015)
3. Попов А.Ю., Чембаев В.Д. Опыт разработки отказоустойчивого комплекса облачных сервисов для поддержки образовательной и научно-исследовательской деятельности МГТУ им Н.Э. Баумана. // Инженерный вестник: Электронный научно-технический журнал МГТУ им Н.Э. Баумана. 2014. № 12. Режим доступа: <http://engbul.bmstu.ru/doc/745300.html> (дата обращения: 17.05.2015)