

# EUROTECH, «ИНТЕРНЕТ ВЕЩЕЙ» И «ОБЛАКО УСТРОЙСТВ»

**СЕРГЕЙ ДРОЗДОВ**  
serge@fiord.com

**СЕРГЕЙ ЗОЛОТАРЕВ**  
zolotarev@fiord.com

В статье представлены программные и некоторые аппаратные решения «Интернета вещей» (IoT), «Интернета устройств» компании Eurotech. Компания Eurotech является одним из родоначальников (наряду с IBM) протокола MQTT (Message Queue Telemetry Transport) для нового направления в развитии глобальной сети и разрабатывает инструментальные средства и платформы, такие как ESF и EDC. По оценкам экспертов, к 2020 году количество устройств, находящихся в Сети, возрастет до нескольких десятков миллиардов. Общаться устройства будут не только с людьми, но и между собой (Machine-to-Machine, M2M), используя при этом принципиально новые протоколы, такие как MQTT, а также облачные сервисы. Многие аппаратные средства Eurotech, одного из мировых лидеров в области встраиваемых компьютерных технологий, поддерживают новую концепцию.

## «ИНТЕРНЕТ ВЕЩЕЙ»: НА ПОРОГЕ БУРНОГО РОСТА

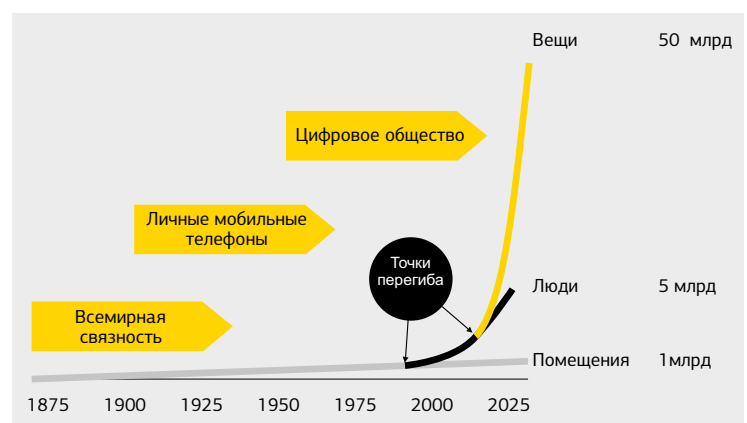
«Интернет вещей» («Интернет для вещей», «Internet of Things», IoT) — это концепция, ориентированная на «интеллект» физических объектов (датчиков, устройств), то есть вещей, а не людей. Это видение и концепция существовали уже давно, но в последние годы резко возросло количество и разнообразие вещей, которые могут быть подключены к Интернету, а также технологий, используемых для идентификации, обнаружения и обмена информацией. Эти технологии достигнут критической массы и экономической точки перелома в ближайшие несколько лет. Они отражают колоссальные перемены в роли Интернета — от предоставления информации и развлечений для людей до обеспечения соедине-

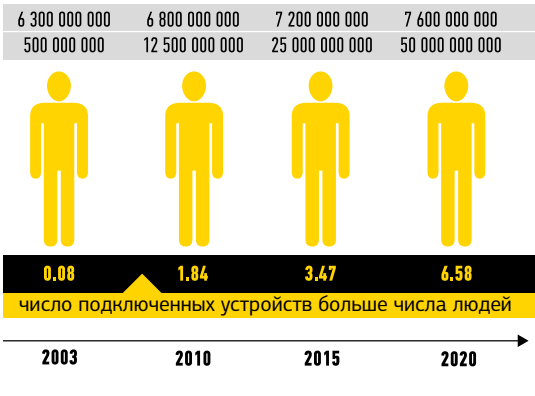
ния нарождающегося класса приложений с поддержкой устройств (device-enabled applications).

Известная консалтинговая компания Gartner назвала десять главных технологий и тенденций, которые будут стратегическими начиная

с 2012 года [1]. Gartner называет стратегической технологию, которая потенциально может оказать значительное влияние на весь мир в ближайшие три года. Стратегической может быть существующая технология, которая достигла зре-

**РИС. 1.** ▶  
Прогноз Ericsson динамики роста числа устройств, подключенных к сетям связи





**РИС. 2. ▲**  
По оценке Cisco, IoT появился на свет в промежутке между 2008 и 2009 годами

лости и/или стала доступна для более широкого спектра применений. В этот список вошла технология «Интернет вещей». IoT будет иметь непосредственное отношение к таким новым направлениям развития цивилизации, как «умные» города и дома, фабрики, автомобили и транспорт, экология, интеллектуальная медицина и здравоохранение.

Сошлемся на несколько первоисточников, в которых подчеркивается стремительное развитие IoT. По данным IDC, в 2015 году к Интернету будут подключены более 16 млрд устройств [2]. Компания Ericsson представила (рис. 1) свой прогноз [3].

Несколько иные оценки приводит компания Cisco [4]. Cisco IBSG прогнозирует, что к 2015 году к Интернету будет подключено 25 млрд, а к 2020 году — 50 млрд устройств (рис. 2). (Отметим, что эти прогнозы не учитывают ускоренного развития интернет-технологий и устройств.) А компания Gartner предсказывает, что к 2020 году количество устройств, находящихся в Сети, возрастет приблизительно до 60 млрд. Хотя существуют и гораздо менее оптимистичные прогнозы.

**ОТ РАДУЖНЫХ ПРОГНОЗОВ К ТЕХНОЛОГИЧЕСКИМ РЕШЕНИЯМ: ПРОТОКОЛ MQTT**

В настоящее время в мире ведутся активные работы в области IoT, как отдельными компаниями (Intel, Cisco, IBM, Eurotech, HP, Alcatel-Lucent, Ericsson, Google), так и в рамках международных альянсов (например, IPSO, www.ipso-alliance.org). Компания

Eurotech является не только одним из мировых лидеров в области встраиваемых компьютерных технологий [5, 6], но и плодотворно работает в других направлениях (в том числе в области развития суперкомпьютеров). Делаются также попытки стандартизации различных аспектов в области IoT в рамках международных организаций, таких как ITU, IEEE, 3GPP и др. Кроме того, активные работы ведутся в отдельных странах, например в США, Китае, и в Европейском Союзе. Внимание к IoT и M2M-проблематике уделяется и в России. В целом можно сказать, что процесс кристаллизации нового направления в IT-индустрии идет полным ходом.

Но уже сейчас есть результат: достаточно широкое распространение протокола MQTT (о котором пойдет речь далее) и готовых к применению решений на его основе. По мнению многих экспертов, протокол MQTT имеет ряд достоинств, по сравнению с протоколом HTTP: меньшие накладные расходы на передачу данных и меньшая полоса пропускания (табл. 1). Для своей работы он не требует постоянного соединения между клиентом и сервером (как в случае HTTP). MQTT также хорошо адаптирован к работе по каналам связи с низкой пропускной способностью. Сравнение HTTP и MQTT можно найти в материалах IBM [7]. Также сошлемся на доклад [9]. В нем приводятся некоторые сравнительные данные по результатам различных экспериментов с использованием протоколов HTTP и MQTT (табл. 1). MQTT (Message Queue Telemetry Transport) — протокол обмена со-

общениями, реализующий модель «публикация/подписка» и предназначенный для связи компьютеризированных устройств. Протокол был предложен еще в 1999 году, и его авторами считаются Энди Стенфорд-Кларк (Andy Stanford-Clark) из IBM и Арлен Ниппер (Arlen Nipper) из Argcom (теперь эта компания входит в состав Eurotech Group как Eurotech, Inc.). Протокол изначально был создан для датчиков, отселяющих состояние труб, однако позже сфера его деятельности была расширена, и он нашел свое применение во множестве встраиваемых приложений, а также в смартфонах. Например, социальная сеть Facebook применяет этот протокол для обмена сообщениями (Facebook Messenger). Спецификация MQTT открыта и доступна на сайте www.mqtt.org. В настоящее время есть два варианта спецификации MQTT: MQTT v3.1 — основная спецификация для сетей на базе TCP/IP и MQTT-S v1.2 для датчиков и встраиваемых устройств в сетях, отличных от TCP/IP, например ZigBee. (В этой статье мы будем всегда иметь в виду только вариант MQTT v3.1 для сетей TCP/IP). Отметим, что в ноябре 2011 года компания IBM открыла исходные коды своего программного обеспечения PANO для управления и обмена сообщениями, в том числе на основе протокола MQTT. В IBM говорят, что решение, предоставленное открытому фонду разработки Eclipse Foundation [10], должно открыть возможности сетевого взаимодействия для большого количества электронных устройств, датчиков и промышленных приложений.

**ТАБЛИЦА 1. ЭКСПЕРИМЕНТАЛЬНОЕ СРАВНЕНИЕ ПОЛОСЫ ПРОПУСКАНИЯ HTTP И MQTT (АВТОР HENRIK SJÖSTRAND)**

Протокол	HTTP	MQTT
Чтение одного блока данных с сервера	302 байт	69 байт (примерно в 4 раза меньше)
Запись одного блока данных на сервер	320 байт	47 байт (примерно в 7 раз меньше)
Чтение 100 блоков данных с сервера	12 600 байт	2445 байт (примерно в 5 раз меньше)
Запись 100 блоков данных на сервер	14 100 байт	2126 байт (примерно в 7 раз меньше)

И более того — в IBM рассчитывают, что со временем MQTT станет главным протоколом «Интернета вещей».

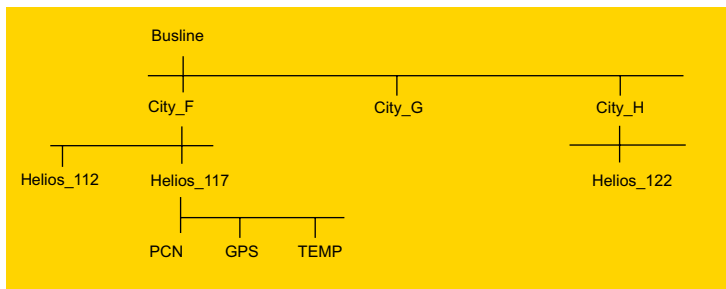
В преамбуле к спецификации MQTT говорится о его особенностях:

- Принята модель обмена сообщениями «публикация/подписка», которая обеспечивает связь «один ко многим».
- Транспорт обмена сообщениями не зависит от содержания данных.
- Для обеспечения подключения к базовой сети используется протокол TCP/IP.
- При доставке сообщений существует три вида качества сервиса (Quality of Service, QoS):
  - ◇ At most once (QoS = 0, «не более одного раза») означает, что публикатор выполняет однократную отправку сообщения, но не предпринимает шаги, кроме предусмотренных TCP/IP, чтобы убедиться в том, что сообщение доставлено. Этот уровень иногда называют «выстрелил и забыл», так как сообщение отправляется адресату без проверки получения.
  - ◇ At least once (QoS = 1, «по крайней мере один раз»): доставка сообщения проверяется, однако разрешается доставлять его более одного раза.
  - ◇ Exactly once (QoS = 2, «только один раз») гарантирует доставку сообщения лишь один раз.

Каждое увеличение уровня QoS приводит к дополнительной нагрузке на процессор и сеть. Выбор QoS может повлиять на общие возможности масштабирования решения по обмену сообщениями, а также создает дополнительную нагрузку, связанную с хранением недоставленных сообщений. Следовательно, нужно правильно выбрать соответствующий уровень QoS для каждого опубликованного сообщения. Как правило, следует использовать более низкие значения QoS, если только не требуется более жесткая гарантия доставки сообщений.

Преимущества:

- Небольшие накладные расходы на транспортном уровне (заголовок фиксированного размера длиной 2 байт).



**Рис. 3.** ◀ Пример Eurotech структурированного пространства тем в протоколе MQTT

- Протокол обмена сведен к минимуму для уменьшения сетевого трафика.
- Встроенный механизм контроля соединения.

Важным элементом всей концепции в MQTT является понятие «тема». В обмене сообщениями по модели publish/subscribe адресаты сообщений называются темами (topics). В протоколе MQTT используется иерархическое пространство тем, то есть структура тем должна позволять подписчикам и публикаторам указывать тему адресата с разными степенями точности. Длина темы не может превышать 32767 символов. Элементы темы отделяются друг от друга разделителями (/) и подстановочными элементами (#, +), которые позволяют определить одного или нескольких адресатов. Кроме темы, передаются данные (payload, data) — значения переменных.

Приведем пример (рис. 3) из материалов компании Eurotech [8] логически сконструированного пространства тем сообщений для гипотетического приложения автобусной компании.

Вот некоторые содержательные примеры использования этого пространства тем:

- FastBus/City\_F/Helios117/TEMP — данные температурного датчика автобуса 117.
- FastBus/City\_F/+ /TEMP — данные всех температурных датчиков всех автобусов в городе City\_F.
- FastBus/City\_F/Helios117/+ — данные всех датчиков автобуса 117 в городе City\_F.
- FastBus/City\_F/# — данные

всех датчиков всех автобусов в городе City\_F.

- FastBus/# — данные всех датчиков всех автобусов во всех городах.
- FastBus/+ /+ /TEMP — данные всех температурных датчиков всех автобусов во всех городах.

Как мы уже говорили, MQTT обеспечивает обмен сообщениями в режиме «публикация/подписка», который позволяет устройствам посылать и получать данные и сигналы тревог, когда возникает некоторое событие (Event-driven application). В модели с одним издателем и многими подписчиками можно отправлять информацию из одной точки многим другим устройствам или «слушателям», которые заинтересованы в получении информации. Это похоже на концепцию социальной сети Twitter. Один человек размещает информацию, а многие абоненты одновременно просматривают ее. Встраиваемые устройства могут использовать протокол MQTT для сбора данных от нескольких устройств с ограниченной пропускной способностью и предоставления информации многим подписчикам. В результате система является относительно простой для настройки и предоставляет идеальный коммуникационный сетевой протокол для облачных решений в мире устройств.

Сообщение MQTT имеет фиксированный заголовок (табл. 2). Кроме того, для некоторых сообщений необходимы заголовок переменной длины и данные. Формат каждой

**ТАБЛИЦА 2. ФОРМАТ ФИКСИРОВАННОГО ЗАГОЛОВКА СООБЩЕНИЯ MQTT**

Бит	7 6 5 4	3	21	0
Байт 1	Тип сообщения Type	Флаг DUP	Уровень QoS	RETAIN
Байт 2	Оставшаяся длина			

**ТАБЛИЦА 3. ТИПЫ СООБЩЕНИЙ ПРОТОКОЛА MQTT 3.1**

Обозначение	Значение	Описание
Reserved	0	Зарезервировано
CONNECT	1	Запрос клиента на соединение с сервером
CONNACK	2	Подтверждение соединения
PUBLISH	3	Публиковать сообщение
PUBACK	4	Публиковать подтверждение
PUBREC	5	Публикация получена (гарантированная доставка, часть 1)
PUBREL	6	Publish Release (гарантированная доставка, часть 2)
PUBCOMP	7	Публикация завершена (гарантированная доставка, часть 3)
SUBSCRIBE	8	Запрос подписки клиента
SUBACK	9	Подтверждение подписки
UNSUBSCRIBE	10	Запрос отказа от подписки клиента
UNSUBACK	11	Подтверждение отказа от подписки
PINGREQ	12	Запрос PING
PINGRESP	13	Ответ PING
DISCONNECT	14	Клиент отсоединяется
Reserved	15	Зарезервировано

части заголовка сообщения описывается в соответствующих частях спецификации MQTT.

Байт 1 содержит тип сообщения и поля флагов (DUP, QoS level и RETAIN), байт 2 (по крайней мере один байт) — поле оставшейся длины. Все значения данных находятся в порядке big-endian: байты высшего порядка следуют перед байтами более низких порядков. Тип сообщения представляется в виде 4-разрядного значения без знака. Допустимые значения для версии протокола MQTT 3.1 указаны в таблице 3.

Оставшиеся биты байта 1 содержат поля DUP, QoS и RETAIN. Флаг DUP устанавливается, когда клиент или сервер пытаются повторно доставить сообщение типа PUBLISH, PUBREL, SUBSCRIBE или UNSUBSCRIBE. Это относится к сообщениям, где значение QoS больше нуля и подтверждение не требуется. Когда установлен бит DUP, переменный заголовок включает идентификатор сообщения (Message ID). Получатель должен рассматривать этот флаг как намек на то, что сообщение уже могло быть ранее получено. Не следует полагаться на обнаружение дубликатов. Флаг QoS указывает на уровень гарантий доставки сообщения PUBLISH. Флаг RETAIN исполь-

зуется только для сообщений PUBLISH. Когда клиент посылает PUBLISH серверу, то, если флаг RETAIN установлен в «1», сервер должен хранить (hold) сообщение после того, как оно было доставлено текущим подписчикам. Сохраненные сообщения должны сохраняться при перезагрузке сервера.

**ОТ ПРОТОКОЛА MQTT К ПЛАТФОРМЕ ДЛЯ ПРИЛОЖЕНИЙ**

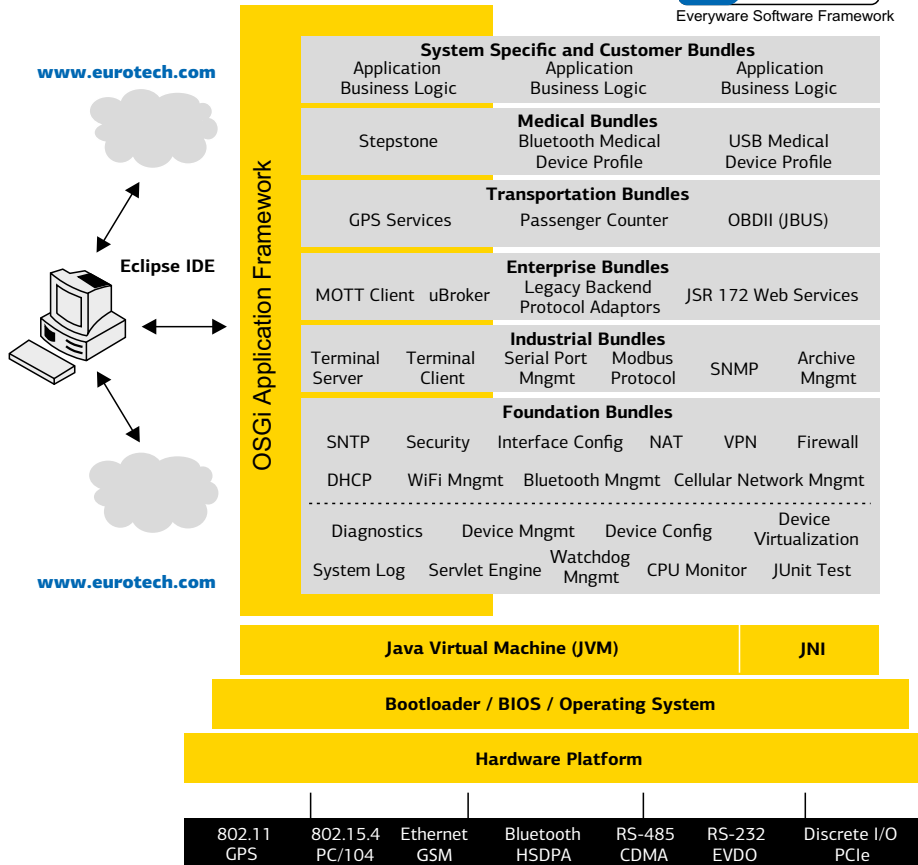
Наличие простого и гибкого протокола, такого как MQTT, — это только базис, на котором можно создавать «здание» приложения. Необходима инструментальная среда (платформа, framework) для быстрого создания конечного продукта. Компания Eurotech разработала такую платформу — Everyware Software Framework (ESF). Эта платформа включает ряд требуемых в таких случаях функций: управление устройствами (обновление ОС и приложений, управление конфигурацией), поддержку работы с несколькими одновременно выполняемыми приложениями и/или плагинами, управление журналом приложения, инфраструктуру обмена сообщениями приложения, обработку событий, управление модулем при-

ложения (Start/Stop/Unload/Load), статистику и метаданные исполнения, управление версиями.

ESF разработана на базе хорошо известной интегрированной среды Equinox OSGi Application Framework и Wind River 3.0 Linux. Это обеспечивает быструю разработку портируемых, безопасных, сетевых (Network-Centric) приложений на языке Java с помощью виртуальной машины Java ME (Java Micro Edition от IBM) с поддержкой аппаратной виртуализации через обширные Java-плагины. ESF обеспечивает легкую интеграцию промышленных протоколов, таких как Bluetooth, GPS, ZigBee, Ethernet и многих других (рис. 4).

Напомним, что OSGi (Open Services Gateway Initiative, инициатива доступа к открытым службам) — спецификация динамической плагиновой (модульной) шины для создания Java-приложений, которую разрабатывает консорциум OSGi Alliance [11]. Круг применения этой спецификации довольно широк, но основная область применения — создание встраиваемых систем. Реализация OSGi, которую Eurotech использует для ESF, получается из Eclipse Foundation. Она называется «Equinox», и это та же реализация от OSGi, которая работает на миллионах исполнительных систем Eclipse IDE, которые используют компании IBM, Oracle, Apple, Adobe, Wind River, BEA, Cisco, Intel и многие другие. Все функции ESF могут выполняться динамически в работающей системе без перезапуска системы и во время работы других приложений. Кроме того, все функции могут осуществляться дистанционно через любое сетевое подключение, что делает ESF оптимальной для распределенных встраиваемых устройств.

ESF отделяет оборудование от технической логики и техническую логику от бизнес-логики (здесь мы сохраняем принятую в Eurotech терминологию). Это разделяет упрощает разработку приложений, предоставляя общие уровни (слои) и позволяя OEM-производителям сосредоточиться на разработке кода приложения. К драйверам устройств обращаются как к сервисам, реализованным с помощью простого API, вместо сложного кода. Простой API устраняет необходимость изменения низкоуров-



невого кода операционной системы (ОС) или конфигурационных файлов ОС путем создания кода приложения и добавления процесса и вертикальных знаний в процесс разработки. Так как ESF базируется на стандартизированном ПО с открытым кодом, которое поддерживают крупные промышленные компании, OEM-производители могут легко портировать существующие приложения третьих фирм в их собственные устройства. ESF как ПО промежуточного уров-

ня (middleware) включает инструменты (рис. 5), которые позволяют изменять, перенастраивать и поддерживать приложения на протяжении всего жизненного цикла изделия с ориентацией на конкретные области применения. В настоящее время предлагаются сборки для промышленности (SCADA), связи, транспорта и медицины. Большинство новых системных проектов и продуктов компании Eurotech написаны на языке Java. Перечислим основные преимуще-



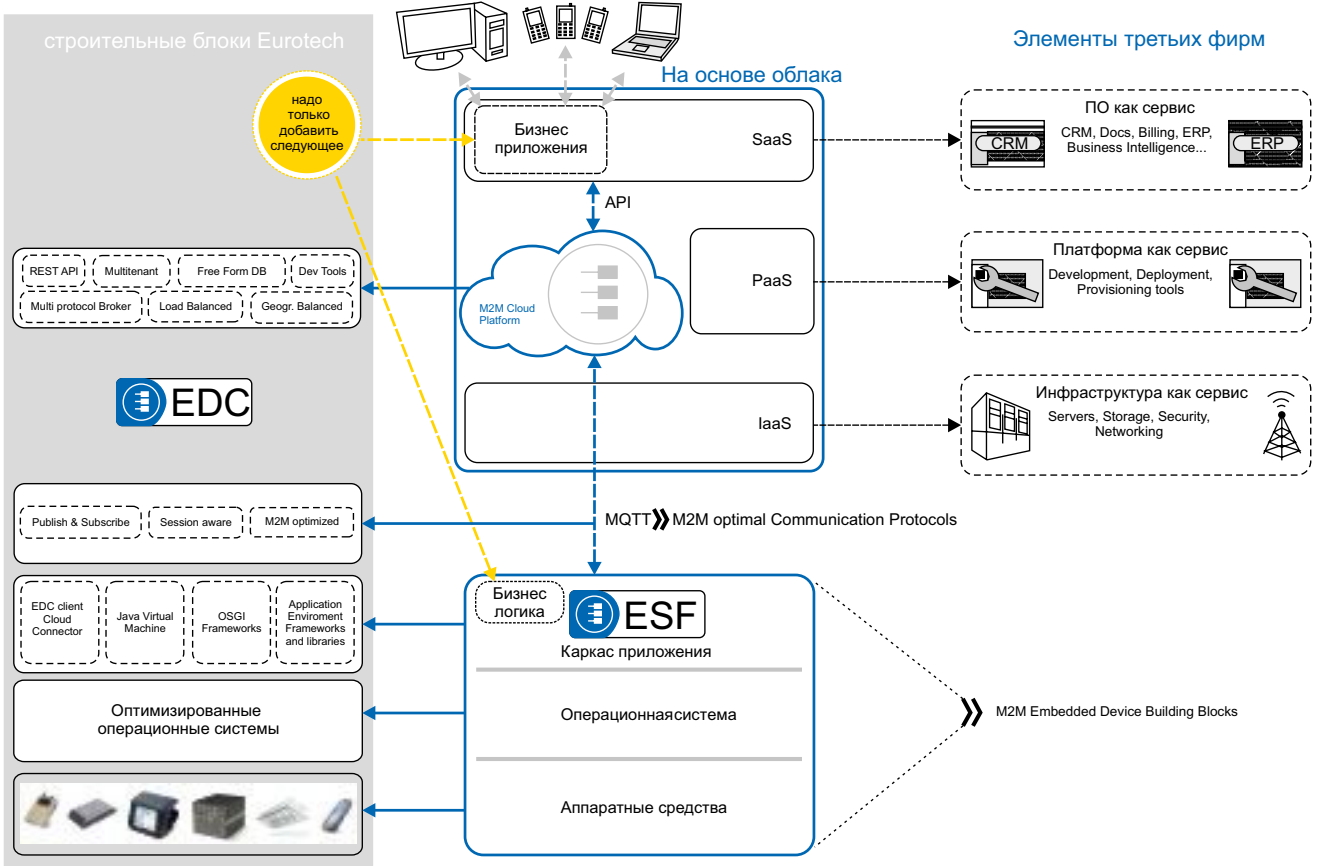
РИС. 5. ▾  
Пример интерфейса пользователя в ESF

РИС. 4. ▲  
Структура платформы ESF

ства встраиваемых приложений на этом языке:

- Java работает на всех операционных системах (повторное использование кода на разных платформах).
- Java-приложения запускаются в изолированной среде «виртуальной машины». В течение последних семи лет Eurotech постоянно использует Java, что привело к значительному уменьшению числа ошибок в приложениях во встраиваемых системах.
- Java — это интерпретируемый язык с «типизируемой безопасностью», что позволяет избежать нанесения катастрофического ущерба ОС.
- Java является сетевым (Network-Centric).
- Java — это полностью объектно-ориентированный язык программирования, который поддерживает реальную (native) многопоточность и JUnit-тестирование. Он имеет встроенное управление памятью и средство документирования (JavaDocs).
- Для Java имеется огромный репозиторий существующих программ, где можно найти исходный код прототипа для многих приложений.
- Реализация Java ME используется во встраиваемых системах в течение десятилетия, она проверена на соответствие стандартам и входит в состав ESF. Хорошо известны также правила лицензирования конечных продуктов на основе ESF.

Eurotech расширяет базовую функциональность Equinox и обеспечивает аппаратную виртуализацию, с которой OEM-производитель начинает свою работу и к которой он должен добавить функционал своей прикладной задачи («бизнес-логику»). Дополнительная функциональность от Eurotech включает аппаратную виртуализацию (одинаковую для аппаратных средств Eurotech, таких как DuraCoror, Catalyst, Zywan), конфигурацию устройства, системный журнал, управление платформой (Watchdog, CPU Temp, CPU Throttling, Sleep modes и т.д.), управление всеми доступными сетевыми устройствами, которые поддерживает платформа.



**EVERYWARE CLOUD: ОБЛАКО ДЛЯ УСТРОЙСТВ ОТ EUROTTECH**

Платформа Everyware Cloud (EDC) — это следующий шаг Eurotech на пути построения готовых решений для «Интернета вещей» [12]. Облачная платформа EDC упрощает управление устройствами и данными, предоставляя M2M-платформы, которые соединяют распределенные устройства и корпоративные

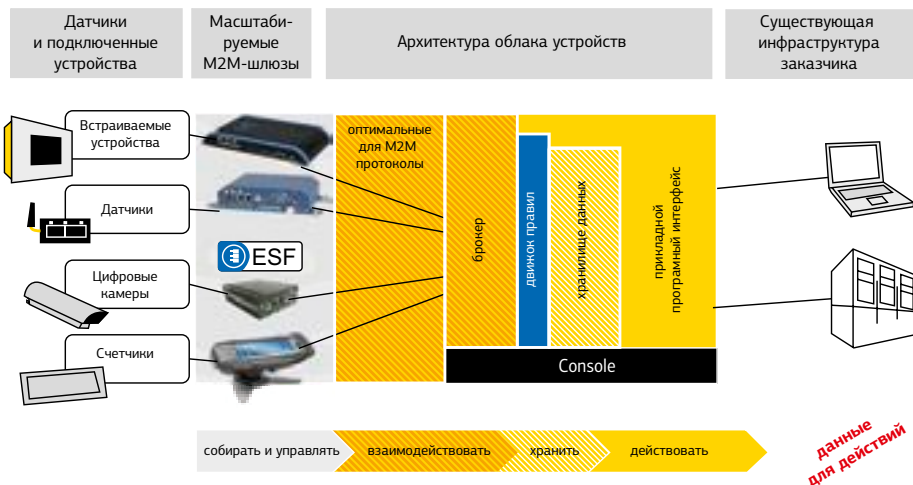
**Рис. 6. ▲** Облако устройств DaaS от Eurotech: Device+SaaS

бизнес-приложения, используя безопасные, открытые и надежные протоколы с поддержкой услуг. EDC базируется на протоколе MQTT (рис. 6). Специалисты Eurotech предлагают новый тип облака DaaS = Device+SaaS: «облако устройств».

**Рис. 7. ▼** Концепция Everyware Cloud

EDC автоматически сохраняет данные от устройств в распределенной, децентрализованной базе данных (БД), которая защищена от сбоев и является масштабируе-

мой. Эта база данных сохраняет любые данные в любом формате на период до 36 месяцев. БД также обеспечивает доступ к данным реального времени в исходной форме для их использования конечным приложением. Платформа позволяет бизнес-решениям мгновенно срабатывать на основе сложных математических и статистических правил, применяемых к потоку данных от устройства. EDC обеспечивает уникальные возможности, включая фильтрацию данных, непрерывность запросов, агрегирование и корреляцию между данными от устройств и распознавание событий по шаблону, обеспечение быстрого реагирования на основе сценариев обнаружения критически важных бизнес-ситуаций и набор уникальных возможностей для конечного приложения M2M. Платформа EDC спроектирована для поддержки высокой доступности, избыточности и резервирования. Она имеет возможность копирования данных на несколько узлов и работает с географически распределенными центрами данных. Эта избыточность обеспечивает автоматическое и прозрачное аварийное восстановление.



ние без дополнительной настройки или потери производительности.

На рис. 7 и в таблице 4 представлена общая структура EDC с привязкой к аппаратным средствам, выпускаемым Eurotech.

К приведенной в таблице 4 информации дадим небольшое пояснение относительно аббревиатуры REST API. REST (REpresentational State Transfer, «передача состояния пред-

ставления») — стиль построения архитектуры распределенного приложения, который описан и популяризован в 2000 году Роем Филдингом (Roy Fielding), одним из создателей протокола HTTP. Самой известной системой, построенной в значительной степени по архитектуре REST, является Интернет.

На сайте Eurotech приведен подробный пример создания при-

**ТАБЛИЦА 4. ДЕТАЛИЗАЦИЯ СВОЙСТВ EVERYWARE CLOUD**

<b>Датчики и устройства</b>	Широкий спектр датчиков и устройств во всех отраслях. EDC обеспечивает необходимое подключение для сбора данных и их распространения
<b>М2М-шлюзы</b>	Eurotech предоставляет ряд надежных аппаратных платформ, включая DuraCor, DuraNav и Helios. Эти платформы являются шлюзами между компонентами на уровне устройств и облаком
<b>Протокол MQTT</b>	Everyware Cloud базируется на протоколе MQTT. Программная платформа Everyware Software Framework (ESF) включает в себя MQTT как функцию коммуникации для обмена данными
<b>Брокер</b>	Основа технологии EDC — это брокер сообщений. Брокер работает в облаке и обеспечивает сообщениями сервер MQTT, на котором устройства публикуют свои данные. MQTT-клиенты могут также делать подписки. Когда полезные данные (payload) публикуются в облаке по определенной теме, данные доставляются клиентам, подписанным к этой теме
<b>Движок правил</b>	EDC использует движок правил на основе SQL для обработки входящих публикуемых данных, обеспечивая немедленную обработку сообщений и/или их модификацию на основе определенных пользователем критериев. Данные реального времени могут подвергаться статистической обработке. Правила могут генерировать действия, такие как отправка по электронной почте, SMS или в Твиттер, публикацию события в MQTT или вызов REST API
<b>Хранение данных</b>	EDC использует базу данных (отличную от SQL) для хранения данных от устройств
<b>API</b>	REST API обеспечивает доступ к запросам, данным и другим возможностям облака
<b>Консоль</b>	Консоль EDC обеспечивает графический интерфейс пользователя, в котором показываются все аспекты работы. Пользователь с правами администратора может управлять обычными пользователями, устройствами, данными и правилами через консоль. Обычные пользователи могут при необходимости просматривать данные
<b>Логика прикладной задачи</b>	Платформа EDC призвана обеспечить возможность создания распределенного приложения без необходимости программирования на уровне устройств или обслуживания сложных сетей данных

## ЛИТЕРАТУРА

- <http://www.crn.ru/news/detail.php?ID=58372>
- Грэйс Т., Миллер К. Построение Интернета вещей — <http://msdn.microsoft.com/ru-ru/magazine/hh852591.aspx>
- Бобровский С. ПАНО? В самый раз! — <http://www.pcweek.ru/idea/blog/idea/?category=1997#>
- Эванс Д. Интернет вещей: как изменится вся наша жизнь на очередном этапе развития Сети — <http://www.cisco.com/web/RU/news/releases/bt/2011/062711d.html>
- Дроздов С. Н., Золотарев С. В. Eurotech Group: передовые компьютерные технологии для ответственных применений // Автоматизация и IT в нефтегазовой области. 2012. № 2 (8).
- Дроздов С. Н., Золотарев С. В. Решения и платформы компании Eurotech Group для транспорта // Автоматизация в промышленности. 2010. № 1.
- Building Smarter Planet Solutions with MQTT and IBM WebSphere MQ telemetry. IBM Redbooks publication. Sept. 2012 — <http://www.redbooks.ibm.com/abstracts/sg248054.html>
- Modotti T. Device to Cloud: MQTT and the power of topic notation. Sept. 27, 2012 — <http://www.embedded.com/electronics-blogs/embedded-cloud-talkers/4397229/Device-to-Cloud>
- Sjstrand H. Low Latency Messaging for Mobile Apps Or When HTTP and Push Notifications Are Simply Not Enough. Using MQ Telemetry Transport (MQTT) for Mobile Applications, WebSphere Client Technical Professional. IBM Sverige — <http://devmobile.se/wp-content/uploads/2012/06/Low-Latency-Messaging-for-Mobile-Apps-MQTT-for-Mobile.pdf>
- <http://www.eclipse.org/paho/>
- <http://ru.wikipedia.org/wiki/DSGi>
- Introduction. Cloud Architecture — <http://everywarecloud.eurotech.com/doc/ECDevGuide/latest/1.02-Cloud-Architecture.asp>
- ESF Protocol/Application examples. GPS & Cloud Publishing Example — <http://esf.eurotech.com/doc/ESFDevGuide/latest/4.01-Cloud-GPS.asp>

ложения (включая исходный код на Java) в среде ESF и публикации данных от GPS в облаке [13]. Пример рассчитан на конечных пользователей и показывает все шаги, начиная от установки Eclipse IDE с последующей интеграцией ESF и ESF-проектов в Eclipse, добавления и использования ESF MQTT-клиента (плагины) к Eclipse. В этом примере создается приложение, которое собирает данные GPS с помощью ESF API, а затем показано, как опубликовать данные в облаке EDC (или подписаться к ним), используя протокол MQTT.

## ЗАКЛЮЧЕНИЕ

Закончить статью хотелось бы словами вице-президента консалтинговой компании Gartner Дэвида Кэрли о десяти важнейших технологиях 2012 года: «Эти топ-10 технологий будут стратегическими для большинства организаций, и IT-руководителям следует использовать этот список в своем стратегическом планировании, внимательно изучая эти технологии и то, как они вписываются в их ожидаемые потребности». С этой точки зрения представленная в статье информация должна быть полезной для отечественных разработчиков, так как, по мнению авторов, она способствует популяризации передовых технологий мировых лидеров в области IT-технологий, в том числе компании Eurotech. ●