Двадцать один пример IoT-платформ

Я уже пишу, наверное, двадцатую статью про "Интернет Вещей". Но всегда, как в первый раз, вынужден рассказывать, объяснять и доказывать. Но тут появился запрос не про объяснения, но про приведение примеров.



Как вы знаете, у меня есть небольшой частный телеграм-канал "ЗаТелеком", куда я выплескиваю новости, старости или просто заметки. Ну, и один пост почему-то вызвал "особый интерес" в виде массы репостов и упоминаний. Он был про "самые модные в этом сезоне IoT-платформы". Короткий простой пост с девятью ссылками на эти самые платформы.

Ну, поскольку интерес к теме проявился нешуточный, я решил сделать вот этот материал, где все и объясню. В очередной раз. Ну, вот первая попытка была.

Что есть IoT-платформа?

Начнем с простого: что есть IoT-платформа, и как отличить оригинал от подделки?

Это очень, на самом деле, сложный вопрос, на который я ответить не смогу. И пытаться не буду.

Но вот есть гораздо более умные парни (разумеется, не наши), которые все понятно объяснили. Вот здесь оригинал, а ниже - краткий переклад на русский.

IoT-платформы - это такая штука, которая объединяет собственно "вещи" и "интернет". По сути - это ключевой инструмент разработки IoT-приложений и сервисов, объединяющий физические объекты и Сеть.

При этом многие вендоры, которые пытаются "держать нос по ветру", предлагают "IoT-платформы", которые в корне различаются между собой. И в ряде случаев, не являются "платформой" в широком смысле слова, но совершенно очевидно имеют основания себя таковой считать - есть "вещь", есть некий ресурс в интернете, который принимает/передает данные от/к "вещи". И что-то там делает (пытается делать) с этими данными. Следовательно, претендовать на высокое звание дом образцового быта платформы вполне себе может. Притом, что четкого и конкретного определения IoT-платформы попросту не существует.

По мнению авторов "IoT Analytics", полноценной IoT-платформой следует считать такую платформу, которая позволяет разрабатывать соответствующие приложения/решения (IoT Application Enablement Platform).

А вот четыре типа платформ, которые называют "IoT-платформами", однако они не вполне подходят под классификацию IoT Analytics:

- Connectivity / M2M platforms. Платформы в своей работе фокусируются на связи умных объектов через телекоммуникационные сети, но редко на обработке сигналов от датчиков (пример такой платформы: Sierra Wireless с продуктом AirVantage).

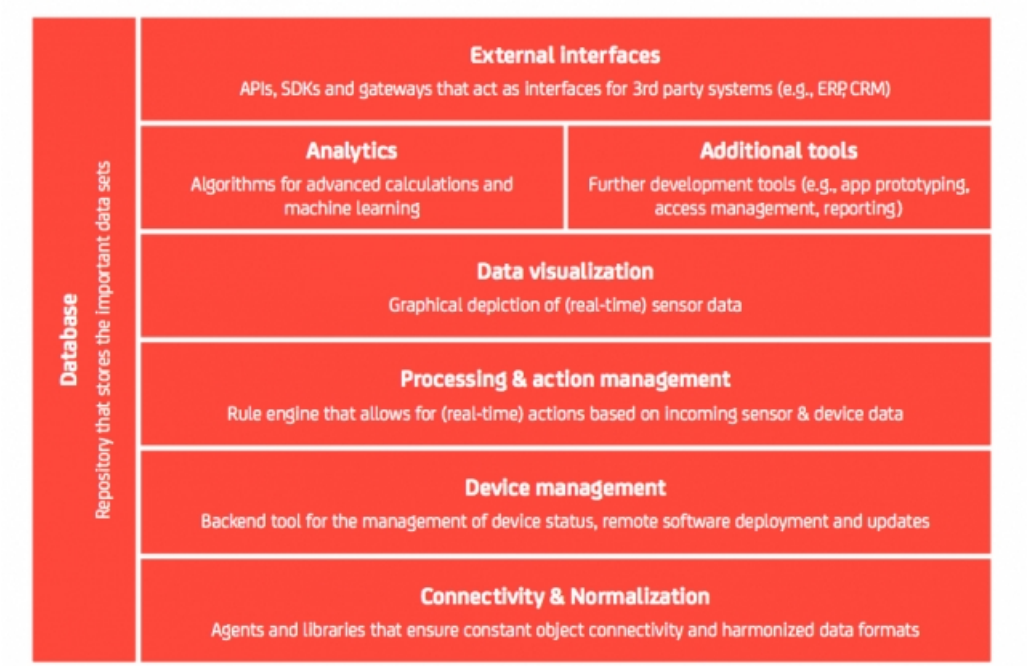
- IaaS backends. Инфраструктура-как-сервис-серверы, предоставляющие хостинг-пространство и вычислительные мощности для приложений и сервисов, ранее оптимизировались для десктопов и мобильных приложений, но сейчас в фокус попал и IoT (пример - IBM Bluemix, но не IBM IoT Foundation).
- Hardware-specific software platforms. Некоторые компании, продающие умные гаджеты, создают собственный программный бэкенд и рассуждают о нем, как об IoT-платформе. Но так как эта платформа носит закрытый для всех остальных характер, правомерность такого наименования сомнительна (например - Google Nest).
- Consumer/Enterprise software extensions. Существующие пакеты корпоративного программного обеспечения и операционные системы типа MS Windows 10 становятся все более открытыми для интеграции IoT-устройств. В настоящее время эта область еще недостаточно развита, чтобы называться IoT-платформой, но будущее у нее очень перспективное.

В общем все запутано и ясности в терминологии нет. Что еще и усугубляется "модностью" темы и желанием разработчиков IoT-платформ комбинировать фантазии маркетологов, как, например, это делает IBM (IoT Foundation application enablement platform + Bluemix IaaS backend).

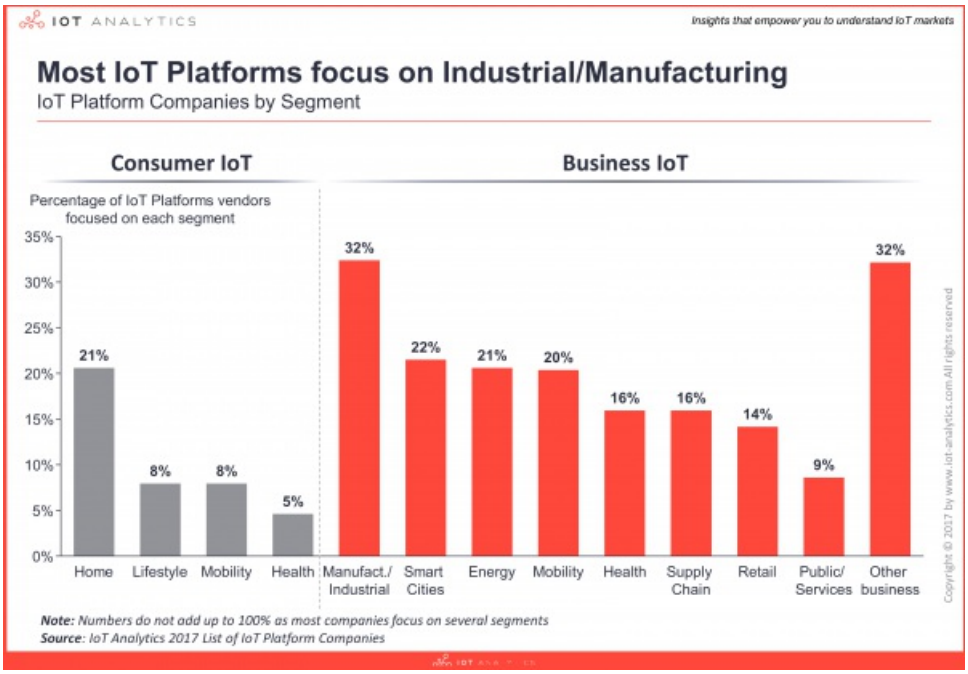
И вот парни из IoT Analytics сделали интеллектуальное усилие, и выделили восемь компонентов полноценной IoT-платформы:

1. **Связь и нормализация** (Connectivity & normalization): сведение различных протоколов и форматов данных в один "программный" интерфейс, гарантируя точную передачу данных и взаимодействие со всеми устройствами.
2. **Управление устройствами** (Device management): обеспечение правильной работы подключенных "интернет-вещей", их конфигурирование, бесперебойную работу, "накатку" патчей и обновлений. Причем, не только ПО собственно "вещей", но и приложений, работающих на устройстве или пограничных шлюзах.
3. **База данных** (Database): тут все достаточно понятно и прозрачно - масштабируемое хранилище данных от "вещей". Требования к этим данным, попытка навести порядок в обработке и перенос данных из, например, разных "платформ" или вовсе к информационным системам "третьих лиц".
4. **Обработка и управление действиями** (Processing & action management): данные, полученные от "вещей" в конечном итоге влияют на события в реальности. По этому "платформа" должна уметь строить процессы, "триггеры событий" и прочее "умные действия" на основе конкретных данных датчиков.
5. **Аналитика** (Analytics): данные от "вещей" являются ценными сами по себе. Поэтому существование комплекса средств их анализа является всенепременным требованием к "платформе". Если сюда включить еще и средства кластеризации данных и глубокого машинного обучения вплоть до прогнозирующей аналитики,то ценность "платформы" очевидно растет.
6. **Визуализация** (Visualization): всю вышеперечисленную аналитику и вообще было бы неплохо показать таким образом, чтобы людям было понятно, приятно и красиво. Строить графики, модели, просто визуализировать то, что происходит с "вещами". Ну, и просто удобный интерфейс.
7. **Дополнительные инструменты** (Additional tools): набор инструментов, который позволяет разработчикам IoT создавать прототипы, тестировать и пробовать различные системы. Приложения, виджеты, машапы - вот это все. Желательно, чтобы не очень углубляться в код и хардкор-программирование.
8. **Внешние интерфейсы** (External interfaces): интеграция с помощью платформы - одна из главных возможностей. Мир интернет-разработки сегодня не терпит замкнутых решений. Всегда может потребоваться передача и обмен со сторонними системами. Поэтому настоящая IoT-платформа обязана иметь интерфейсы прикладного программирования (API), комплекты разработки программного обеспечения (SDK) и шлюзы.

Вот картинка для общего понимания (я не стал ее переводить):



На рынке уже имеются сотни платформ. [IoT Analytics насчитали их более 450 и число продолжает расти](). При этом различные платформы появляются из-за различных стратегий входа различных компаний, видения развития, подходов и начальных условий. Клиентские сегменты, кстати:



Стартапы, производители оборудования и сетевого оборудования, корпоративного программного обеспечения и компаний по управлению мобильностью - все они конкурируют, чтобы вывести на рынок лучший продукт.

Попробуем разобрать эти стратегии:

- Стратегия "органического роста" снизу-вверх (Organic bottom-up approach): то есть, платформа начинает расти от "вещей" - сначала появляются некие устройства (обычно универсальные датчики), а потом начинается расширение и улучшение функционала, выстраиваются связи, подключаются другие "железки". Так, например, начиналась платформа Ayla Networks, которая возникла с контроллеров STM32F3.
- Стратегия "сверху вниз" (Organic top-down approach): обратная ситуация, когда сначала появлялась аналитическая составляющая, а потом к ней пытались приспособить как можно больше разных "вещей". Пример - упомянутая IBM IoT Foundation.
- Стратегия "партнерства" (Partnership approach): создание альянсов для развития продукта. Кто-то "приносит" "вещи", кто-то "админку". Пример: альянс GE Predix (это тот самый "Дженерал Электрик", у которого активов по Миру на триллион) и PTC Thingworx. Последняя компания в России больше известна пакетом компьютерной алгебры Mathcad.
- "Слияния и поглощения" (M&A approach): любимый капиталистический прием по целевой скупке

интересных решений крупным игроком. Например, Amazon в 2015 году купил 2lemetry, и теперь это все называется AWS IoT. Вариант стратегии - объединение бизнесов, как, например, Nokia и Alcatel-Lucent.
- Ну, и есть еще "Инвестиционный подход" (Investment approach): тактические инвестиции по всей экосистеме IoT. Так любит поступать, например, Cisco.

IoT Analytics утверждает, что в настоящее время продажа основного продукта становится побочным процессом, а основные бизнес-интересы строятся вокруг "торговли данными", что довольно интересная мысль, но на российском рынке пока "цифровой нефтью" интересуются лишь спецслужбы.

В прогнозе IoT Analytics присутствует простая мысль, которую я много раз уже повторял в своих опусах про "интернет вещей": очень скоро мы увидим создание абсолютно новых отраслей, которые становятся возможными с повсеместным использованием "множества устройств, подключенных к интернету".

Для того чтобы эти модели "полетели" практически все готово - есть аппаратное обеспечение, имеется развитая инфраструктура передачи данных, есть очень хорошее программное обеспечение. Осталось чуть-чуть, и...

Но двинемся дальше.

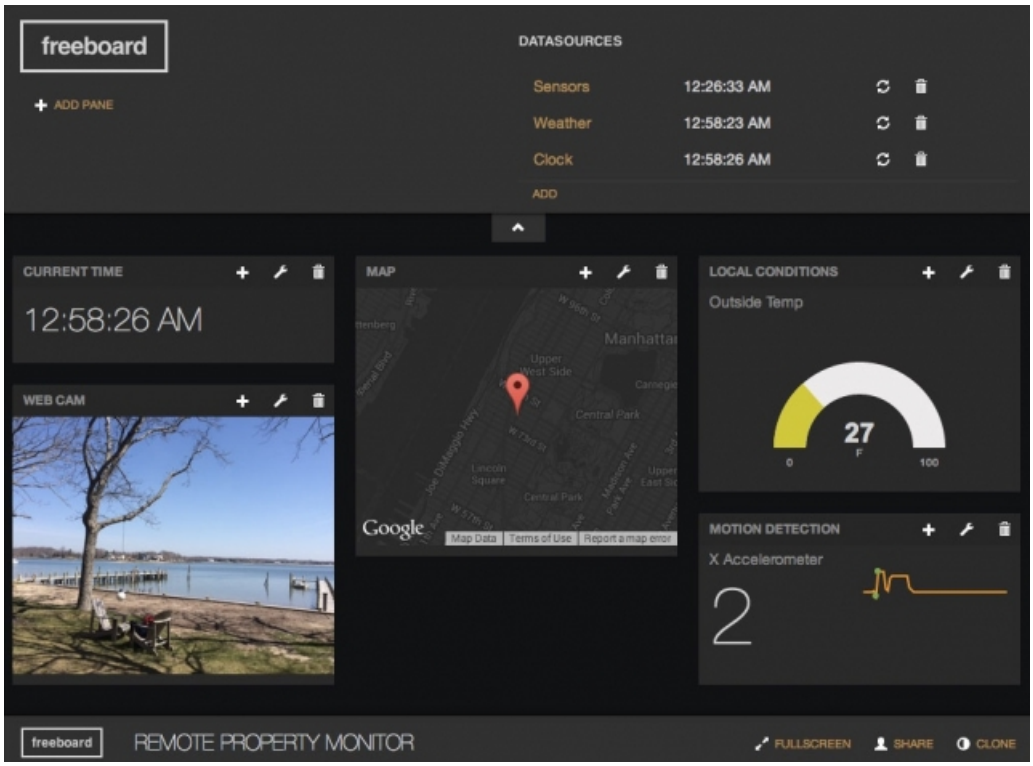
Двадцать одна открытая платформа "Интернета вещей"

Поскольку список большой, я не стану по каждой платформе писать подробно. Но все эти проекты объединяет одно - открытый исходный код. На каждую сделаны ссылки, но не каждую ссылку я проверил методом установки и изучения кода. Нужно понять и простить. Ссылки в алфавитном порядке.

Предупреждаю, что в процессе осмысления можно сломать мозг. Слишком много концептуальных терминов и названий, которые я вообще затрудняюсь перевести.

1. AllSeen Alliance (AllJoyn) - Фреймворк взаимодействия "вещей" AllJoyn, созданная AllSeen Alliance (ASA), который является проектом Linux Foundation. Есть мнение, что это одна из самых популярных IoT-платформ с открытым исходным кодом.

2. Bug Labs - пример стратегии разработки "снизу вверх". Парни из "Лаборатории ошибок" (красиво, кстати), начали разработку и железок для IoT под заказ, но потом обросли кучей веб-приложений и стали платформой, которая зарабатывает на разработке "нестандартных решений". Например, таких. Bug Labs в настоящее время имеет платформу для обмена сообщениями и оповещений "dweet" и "Freeboard - дашборд для создания панелей вывода и визуализации IoT".



3. DeviceHive - платформа управления всевозможными устройствами на базе DataArt - форк AllJoyn (первый пункт). Предназначен для быстрого развертывания на популярных облачных сервисах: Azure, AWS, Apache Mesos и OpenStack. DeviceHive фокусируется на анализе больших данных с использованием таких инструментов, как ElasticSearch, Apache Spark, Cassandra и Kafka. Работает на любом устройстве, на котором запущено Ubuntu Snappy Core. Собственно ПО построено по модульному принципу с идеологией "промежуточного шлюза". Ну, то есть собственно "вещь" подключается к бордеру в виде, например, Raspberry PI (вот список поддерживаемого железа, где работает Snappy Core), а затем уже "цепляется" к облаку, где развернут DeviceHive.

4. DSA - Архитектура распределенных служб (Distributed Services Architecture). Достаточно сложная для понимания штука, которая предполагает создание полносвязных сетей "вещей". Имеет три элемента: DSBroker, DSLink и nodeAPI. Попробуйте сами разобраться.

5. Eclipse IoT (Kura) - IoT-решение от Eclipse Foundation. Основа заключается в наличии API-контейнера Kura API. Написан на Java / OSGi и платформе агрегации для приложений M2M, работающих на служебных шлюзах. Kura использует Eurotech Everywhere Cloud IoT, легко интегрируется с Apache Camel. Подпроекты Eclipse IoT включают в себя инфраструктуру протоколов обмена сообщениями Paho, и полный стек MQTT Mosquitto для "легких серверов" со средой Eclipse SmartHome. Существует также Java-реализация протокола CoAP под названием Californium и еще что-то. Много всего.

6. Kaa - проект поддерживается компанией CyberVision, и представляет собой масштабируемую инфраструктуру IoT, предназначенную для достаточно больших сетей. Платформа имеет серверную функцию REST-сообщений для служб, аналитику и управление данными. На базе Apache Zookeeper можно создавать кластерные системы, масштабируемые "до куда угодно". SDK Kaa поддерживают разработку на Java, C ++ и C. Имеются библиотеки организации связи "клиент-сервер", аутентификацию, шифрование, хранение и сортировку данных.

7. Macchina .io - универсальная среда для разработки приложений IoT-шлюзов, работающих на железках под Linux. Macchina.io уже включает поддержку огромного количества датчиков и технологий подключения, включая Tinkerforge bricklets, кучу датчиков на XBee/ZB, GPS/GNSS приемники и еще кучу всего.

8. GE Predix - выше эта платформа уже упоминалась. Это по сути PaaS (платформа как услуга) для промышленного IoT и базируется на Cloud Foundry. Платформа умеет "управлять активами, обеспечивать безопасность устройств и готовить аналитику в режиме реального времени. Ну, и все остальное, что должны уметь платформы, конечно это сбор данных, их хранение и обеспечение доступности. "Дженерал Электрик" разработали GE Predix, прежде всего, для собственных нужд. Соответственно имеет некоторую отраслевую специфику - электроэнергетика. Predix считается одной из самых успешных IoT-платформ и утверждается, что генерит разработчику порядка 6 млрд. долларов.

9. Home Assistant - платформа для массового использования для целей домашней автоматизации. Написана на Python.

10. Mainspring - платформа запилена на Java компанией M2MLabs. Довольно старенькая и страшненькая (как, впрочем, и все написанное на Java - шутка). Использует для коммуникаций "вещей" REST и предлагает инструменты настройки оборудования и моделирования.

11. Node-RED - этот инструмент визуальной разработки на Node.js. Имеется браузерный редактор "потоков", с помощью которого можно проектировать целые сети IoT с узлами и хабами. После узлы могут быть быстро развернуты как "среды выполнения" на куче серверов и/или в облаках. Обмен данными основан на JSON, что логично. Поддерживаются "вещи" на платах с Linux, а облачная поддержка - Docker, IBM Bluemix, AWS и Azure.

12. Open Connectivity Foundation (IoTivity). Совместная разработка Intel и Samsung, которые инвестировали в Open Interconnect Consortium (OIC) и UPnP Forum. Очень хотят стать ведущей группой стандартов на базе открытого кода для IoT. IoTivity поддерживает протоколы обмена данными на RESTful, JSON и CoAP.

13. OpenHAB - среда разработки для "умного дома" с открытым исходным кодом. Может (по задумке) работать на любом устройстве, способном запускать JVM. Модульная архитектура на уровне абстракции разделяет все используемые технологии и компоненты IoT на "элементы", которые поддерживает всевозможные правила, скрипты и процессы.

14. OpenIoT - Java-based платформа для создания IoT-приложений. Облачная, разумеется. Платформа включает промежуточное ПО датчиков и сенсорной сети, а также онтологии, семантические модели и аннотации для представления объектов IoT. Судя по гитхабу -

давненько не обновлялось.

15. OpenRemote - система, которую изначально разрабатывали для автоматизации зданий. OpenRemote отличается широкой поддержкой редких сетевых спецификаций и протоколов, например 1-Wire, EnOcean, xPL, Insteon и X10. Все остальное стандартно - правила, сценарии и события. Разумеется облачные инструменты проектирования для пользовательского интерфейса, установка и настройка, а также удаленные обновления и диагностика.

16. OpenThread - спин-офф от известной компании Nest (купленной Google за 3,2 млрд. долларов) с открытым исходным кодом. Заточен под устройства на 6LoWPAN, но есть поддержка и других протоколов. Работает на железных платформах от ARM, Atmel, Microchip, Dialog, Qualcomm и TI. OpenThread реализует сетевые роли модели "Thread": End Device, Router, Leader, and Border Router.

17. Physical Web / Eddystone - опен-сорс разработка Google. Это они пытались создать что-то очень похожее на iBeacon от Apple. Маячки на Bluetooth 4.0 с поддержкой "экономии энергии" (BLE) должны передавать URL-адреса на ваш смартфон. Идея состоит в том, что обладатели смартфонов могут взаимодействовать с любыми девайсами с поддержкой BLE, таким как парковочные счетчики, вывески или розничные продукты.

18. PlatformIO - система разработана на Python и включает в себя IDE, генератор проектов и веб-менеджер библиотек. Изначально разрабатывалась для доступа к данным с конечных точек на Arduino и ARM Mbed. Сейчас имеются готовые прошивки-настройки для более чем 200 плат и интегрируется с Eclipse, Qt Creator и другими IDE.

19. The Thing System - программное обеспечение на Node.js, предназначенное для смартфонов. Утверждается, что имеется поддержка "реальной автоматизации", а не просто уведомления - таких проектов, надо сказать, очень много. Есть что-то похожее на "самообучение" и искусственный интеллект, что позволяет организовать взаимодействие множества сценариев M2M взаимодействия. Отсутствие облачного компонента обеспечивает большую безопасность, конфиденциальность и контроль. Ну…

20. ThingSpeak - проект с пятилетней историей. Фокусируется на регистрации датчиков, отслеживании местоположения, триггерах и предупреждениях и анализе всех этих данных. Пользователи ThingSpeak могут использовать версию MATLAB для анализа и визуализации данных, не покупая лицензию от Mathworks.

21. Zetta - сервер-ориентированная IoT-платформа на Node.js и REST/WebSockets. Использует философию разработки, основанную "на потоках реактивного программирования" (не знаю что это). Рекламируется как API-first система с интерфейсами Siren hypermedia. Любое устройство представляется, как набор API REST, и связывается с облачными сервисами, где имеются инструменты визуализации и поддержки машинной аналитики Splunk. Платформа поддерживает любые Linux-платки и Arduino. Использует Heroku для создания георасределенных сетей, что тоже звучит очень круто.

А что в России?

Насколько мне известно, работы по созданию IoT-платформ в РФ идут днем и ночью. Тысячи энтузиастов перерабатывают гигабайты открытого исходного кода для создания систем или "умного дома", или для "сбора данных со счетчиков".

На большее фантазии не хватает.

Впрочем, если вы знаете российские хорошие годные проекты - напишите мне. Я сделаю обзор.

Пока же покажу одну систему, которую вполне можно назвать IoT-платформой - это iRidium Mobile (нет, не спутниковая система связи) из Нижнего Тагила. Земляки.

Это как раз платформа для управления "умным домом". По сути, это некий "облачный хаб", который умеет преобразовывать массу IoT-протоколов и сводить их к "порталу управления". Если еще понятнее - инсталляторы "умных домов", а также "умных зданий" и прочих систем интернет-автоматизации экономят массу времени, денег и усилий на разработке пользовательских интерфейсов и интеграции кучи всяких протоколов и данных. Можно подобрать соответствующие датчики и/или исполнительные устройства, подключить их "по инструкции", а все остальное отдать iRidium.

Где уже готовы интерфейсы - нужно только перетащить мышкой в нужное место и поставить "логотип заказчика". Ну, и чуток автоматизации там тоже есть - это же платформа.

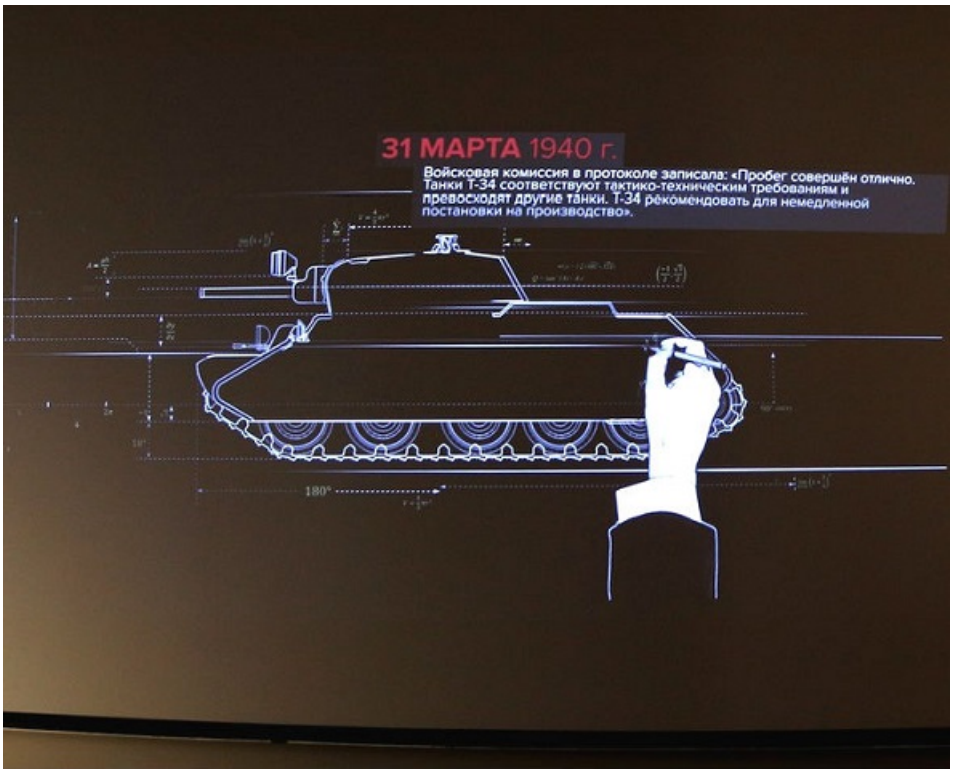
Более подробно на сайте. Я же покажу несколько фоточек, которые с сайта и утянул:

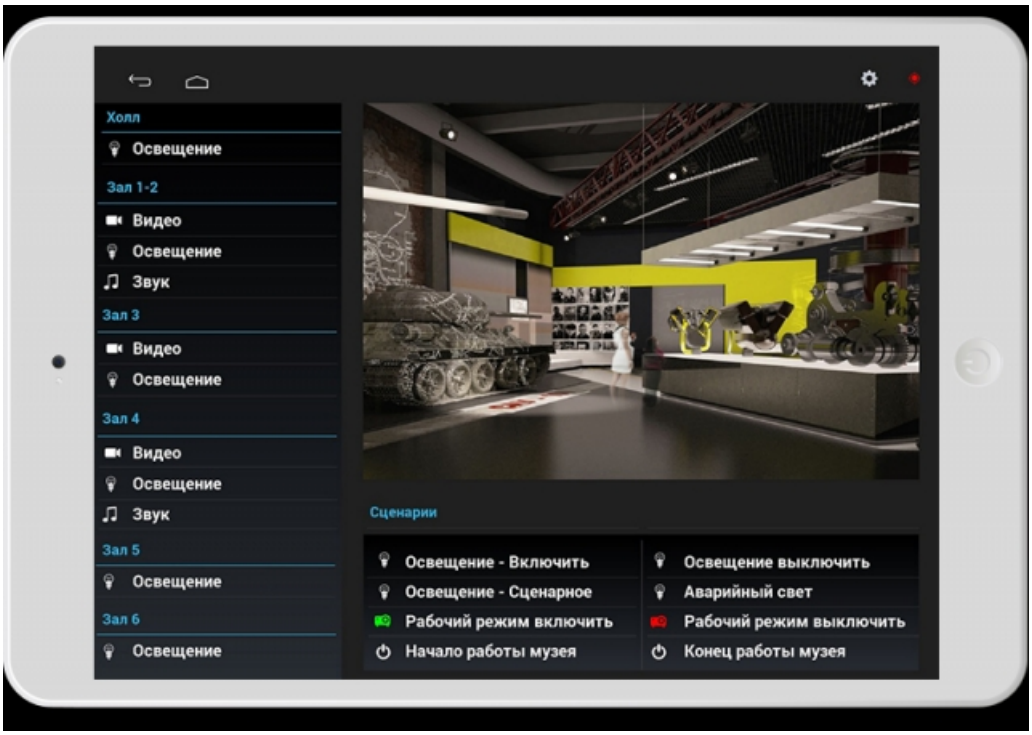
Например, "умный корабль":





Музей бронетехники:





Иди даже культовое сооружение "Храм Соломона" в Бразилии:




Iluminação
Teto 1
Teto 2
Teto 3
Teto 4
Teto 5
Menorá
Colunas
Cênica
Letreiro
Desligar Teto
22:45
On Air
Power AV
CAM
Áudio
Vídeo
Acionamentos
Blackout
Skyfold
Cortina
Esteira
Subir
Parar
Descer
Abrir
Fechar
ON
OFF
Volumes
Nave
Altar
Alto
Médio
Baixo
Painéis de LED
Ao vivo TV
Câmera Púlpito
Vídeo Conferência
AUX Pulpito
AUX Cabine
iPad
Black
Banner
NAVE
TECLADISTA