МИРЭА
Российский
технологический
университет

Технологические основы Интернета Вещей

Лекция 6. Туманные технологии. Программные платформы Интернета Вещей

Жматов Дмитрий Владимирович
кандидат технических наук, доцент
доцент кафедры Математического обеспечения и стандартизации информационных технологий

Управление узлом и базовое ПО

Это относится к управлению In-Band (IB) и определяет, как туманный узел общается с другими узлами в своей области. Узлы также управляются через этот интерфейс для обновлений, статуса и развертывания. Базовое ПО может включать в себя операционную систему узла, пользовательские драйверы и прошивку, протоколы связи и управление, управление файловой системой, программное обеспечение для виртуализации и контейнеризацию для микросервисов. Этот уровень стека программного обеспечения касается почти любого другого слоя в OpenFog Reference Architecture. Типичными особенностями базового ПО являются:

- обнаружение сервиса – позволяет ситуативные модели, доверительные отношения между облаками; обнаружение узла – позволяет добавлять туманные узлы и присоединять их к кластерам, подобным облачным кластерам;
- управление состоянием – позволяет различные вычислительные модели для вычислений с сохранением состояния и без состояния для многих узлов;
- управление Pub/Sub – позволяет переносить данные, а не вытягивать их. Кроме того, позволяет использовать уровни абстракции при разработке программного обеспечения.

Эталонная архитектура OpenFog или, в любом случае, как и любая туманная архитектура, должна обеспечивать уровни развертывания. То есть, туманная архитектура не просто ограничена облаком, подключенным к туманному шлюзу, соединенному с несколькими датчиками. На самом деле существует множество топологий, зависящих от масштаба, пропускной способности, нагрузки и экономических факторов, которые могут быть разработаны. Эталонная архитектура должна подходить для множества топологий, так же, как реальное облако может динамически масштабироваться и балансировать нагрузку на основании спроса.

Виртуализация оборудования

Как и обычные облачные системы, OpenFog определяет аппаратное обеспечение как уровень виртуализации. Приложения не должны иметь привязки к определенному набору оборудования. Здесь система должна балансировать нагрузку через туман и перемещаться по ресурсам или добавлять ресурсы по мере необходимости. Все аппаратные компоненты виртуализированы на этом уровне, включая вычисления, сеть и хранилище.

Безопасность узла OpenFog

Консорциум определяет этот уровень как часть безопасности оборудования стека. Туманные узлы более высокого уровня должны иметь возможность контролировать туманные узлы более низкого уровня как часть иерархии в топологии. Одноранговые узлы должны иметь возможность наблюдать за своими соседями на востоке-западе. Этот слой также выполняет следующие обязанности:

- криптование;
- мониторы слежения и физической безопасности;
- инспекция и мониторинг пакетов (с востока на запад и с севера на юг).

Сеть

Это первый компонент уровня аппаратной системы. Сетевой модуль представляет собой модуль связи восток–запад и север–юг. Сетевой уровень осведомлен о туманной топологии и маршрутизации. Он выполняет физическую функцию маршрутизации к другим узлам. Это большое отличие от традиционной облачной сети, которая виртуализирует все внутренние интерфейсы.

Например, родительский узел, управляющий четырьмя другими дочерними узлами, все из которых подключены к камерам, может нести ответственность за объединение видеоданных из четырех источников и объединение (слияние) содержимого изображения вместе для создания поля зрения на 360°. Для этого он должен знать, какой дочерний узел относится к тому или иному направлению, и он не может делать это произвольно или случайно. Требования к сетевому компоненту:

- устойчивость, если линия связи отказывает. По сути, может потребоваться понять, как перестроить сеть, чтобы сохранить поток данных;
- сетевой уровень также является местом для передачи данных и переупаковки от не-IP-датчиков к IP-протоколам. Примерами этого являются Bluetooth, Z-Wave и проводные датчики;
- обработка отказов;
- связывание с различными коммуникационными сетями (Wi-Fi, проводной, 5G);
- обеспечение типичной сетевой инфраструктуры, необходимой при развертывании предприятия (безопасность, маршрутизация и т. д.).

Ускорители

Другим аспектом OpenFog, чем он отличается от других облачных схем, является понятие услуг ускорителя. Ускорители теперь обычны в виде GPGPU и даже FPGA для предоставления услуг для обработки изображений, машинного обучения, компьютерного зрения и восприятия, обработки сигналов и шифрования/дешифрования. OpenFog предусматривает, что туманные узлы могут снабжаться ресурсами и распределены по мере необходимости. В иерархии можно заставить ферму узлов второго или третьего уровня обеспечить дополнительные вычислительные средства по мере необходимости динамически.

Можно осуществить работу других форм ускорения в тумане, например:

узлы, предназначенные для массового хранения в случае, если необходимо создать большое облако данных;

узлы, которые включают в себя альтернативные линии связи, такие как спутниковая радиостанция в катастрофическом случае, когда не работает вся наземная связь.

Вычислительная часть стека аналогична вычислительной функциональности уровня Nova в OpenStack.

Основные функции:

- выполнение задачи;
- мониторинг и обеспечение ресурсов;
- балансировка нагрузки;
- запросы возможностей.

Хранение

Среда хранения архитектуры поддерживает интерфейс низкого уровня к туманному хранилищу. Типы хранения, с которыми мы сталкивались ранее, такие как озера данных или память рабочего пространства, могут понадобиться на границе для жесткого анализа в реальном времени. Слой хранения также будет управлять всеми традиционными типами устройств хранения, такими как:

- массивы виртуальной памяти;
- заменяемые диски;
- flash-накопители;
- RAID;
- шифрование данных.

Инфраструктура аппаратной платформы. Абстракция протокола

Уровень инфраструктуры – это не столько фактический уровень между программным и аппаратным обеспечением, сколько физическая и механическая структура туманного узла. Поскольку туманные устройства будут находиться в суровых и удаленных местах, они должны быть прочными и устойчивыми, а также автономными. OpenFog определяет случаи, которые необходимо учитывать при развертывании тумана, в том числе:

- размеры, мощность и весовые характеристики;
- системы охлаждения;
- механическое закрепление и удержание механики обслуживания;
- устойчивость к физическому воздействию и отчетность.

Уровень абстракции протокола связывает самые нижние элементы системы IoT (датчики) с другими слоями туманного узла, другими туманными узлами и облаком. OpenFog поддерживает модель абстракции для идентификации и связи с сенсорным устройством через слой абстракции протокола. Абстрагируя интерфейс к датчикам и периферийным устройствам, гетерогенную смесь датчиков можно развернуть на одном туманном узле, например, аналоговые устройства, которые проходят через цифро-аналоговые преобразователи, а также цифровые датчики. Даже интерфейсы к датчикам могут быть индивидуализированы, например, Bluetooth для температурных устройств в транспортных средствах может подключаться к другим датчикам двигателя, датчикам сопряжения SPI на различных автомобильных электроприборах и датчикам GPIO к различным датчикам открытия двери и случая кражи. Абстрагируя интерфейс, верхние слои стека программного обеспечения могут обращаться к таким разрозненным устройствам с помощью стандартизованного подхода.

Датчики, приводы и системы управления

Это нижний уровень стека IoT: датчики и граничные устройства. Эти устройства могут быть умными, немыми, проводными, беспроводными, располагаться ближе, дальше и т. д. Однако объединение в том, что они каким-то образом сообщаются с туманным узлом, а тот несет ответственность за обеспечение, защиту и управление каждым датчиком.

Amazon Greengrass и лямбда-функции

В этом разделе мы рассмотрим альтернативную туманную службу Amazon Greengrass. Amazon предоставляла в течение многих лет ведущие облачные сервисы и инфраструктуру мирового класса, такие как AWS, S3, EC2, Glacier и другие. С 2016 г. Amazon инвестировала в новый стиль обработки граничных данных под названием Greengrass. Greengrass – это расширение AWS, которое позволяет программисту погружать клиента в туман, шлюз или интеллектуальное сенсорное устройство.

Аналогичным образом другие туманные фреймворки, цель которых состоит в том, чтобы обеспечить решение для сокращения времени ожидания и времени отклика, снизят затраты на пропускную способность и обеспечат безопасность края. Особенности Greengrass следующие:

- данные из кэша в случае потери соединения;
- синхронизация данных и состояния устройства с облаком AWS при повторном подключении;
- локальная безопасность (службы проверки подлинности и авторизации);
- брокер сообщений на устройстве и вне устройства;
- фильтрация данных;
- управление и контроль устройством и данными;
- объединение данных;
- работа в автономном режиме;
- итеративное обучение;
- вызов любой службы AWS непосредственно из Greengrass на крае.

Чтобы использовать Greengrass, программа создаст облачную платформу в AWS IoT и определит определенные лямбда-функции в облаке. Эти функции Lambda затем назначаются граничным устройствам и развертываются на тех устройствах, на которых запущен клиент и которые уполномочены выполнять Greengrass Lambdas. В настоящее время функции Lambda написаны на Python 2.7.

Shadows – это абстракции JSON в Greengrass, которые представляют состояние устройства и функции Lambda. При необходимости они синхронизируются с AWS.За кулисами связь между Greengrass на краю и AWS в облаке осуществляется через MQTT.

Пример Lambda-определения, используемый в Greengrass, будет выглядеть следующим образом. С консоли в AWS мы запускаем командную строку; мы запускаем следующий инструмент и определяем функцию Lambda по имени:aws greengrass create-function-definition --name “sensorDefinition”
В результате получим следующее:

```
{
 "LastUpdatedTimestamp": "2017-07-08T20:16:31.101Z",
 "CreationTimestamp": "2017-07-08T20:16:31.101Z",
 “Id”: “26309147-58a1-490e-a1a6-0d4894d6ca1e”,
 "Arn":"arn:aws:greengrass:us-west-
2:123451234510:/greengrass/definition/functions/26309147-58a1-490e-a1a6-0d4894d6ca1e",
 "Name": "sensorDefinition"
}
```

Теперь мы создаем объект JSON с определениями функции Lambda, используем указанный выше идентификатор и вызываем функцию create-func-tiondefinition-version из командной строки:
Executable – исполняемая функция Lambda по имени;
MemorySize – это объем памяти, выделяемый обработчику;
Timeout – время ожидания в секундах до истечения времени ожидания таймера.
Ниже приведен пример объекта JSON для использования с функцией Lambda:

```
aws greengrass create-function-definition-version --function-definition-id
"26309147-58a1-490e-a1a6-0d4894d6ca1e". --functions
'[
{
   "Id": "26309147-58a1-490e-a1a6-0d4894d6ca1e",
   "FunctionArn": "arn:aws:greengrass:us-west-
2:123451234510:/greengrass/definition/functions/26309147-58a1-490e-a1a6-
0d4894d6ca1e",
   "FunctionConfiguration": {
      "Executable": "sensorLambda.sensor_handler",
      "MemorySize": 32000,
      "Timeout": 3
   }
```

Туманные топологии

Туманные топологии могут существовать во многих формах, и архитектору необходимо учитывать несколько аспектов при разработке туманной системы «из конца в конец». В частности, при разработке топологии вступают в силу ограничения, такие как стоимость, нагрузка на процессор, интерфейс производителя и передача между востоком-западом. Туманная сеть может быть простой, как граничный маршрутизатор с поддержкой тумана, соединяющий датчики с облачным сервисом. Она также может усложняться в многоуровневой туманной иерархии тумана с разной степенью способности обработки в каждом слое и ролями на каждом уровне, одновременно перераспределяя нагрузки обработки, когда и где это необходимо (с востока на запад и с севера на юг).

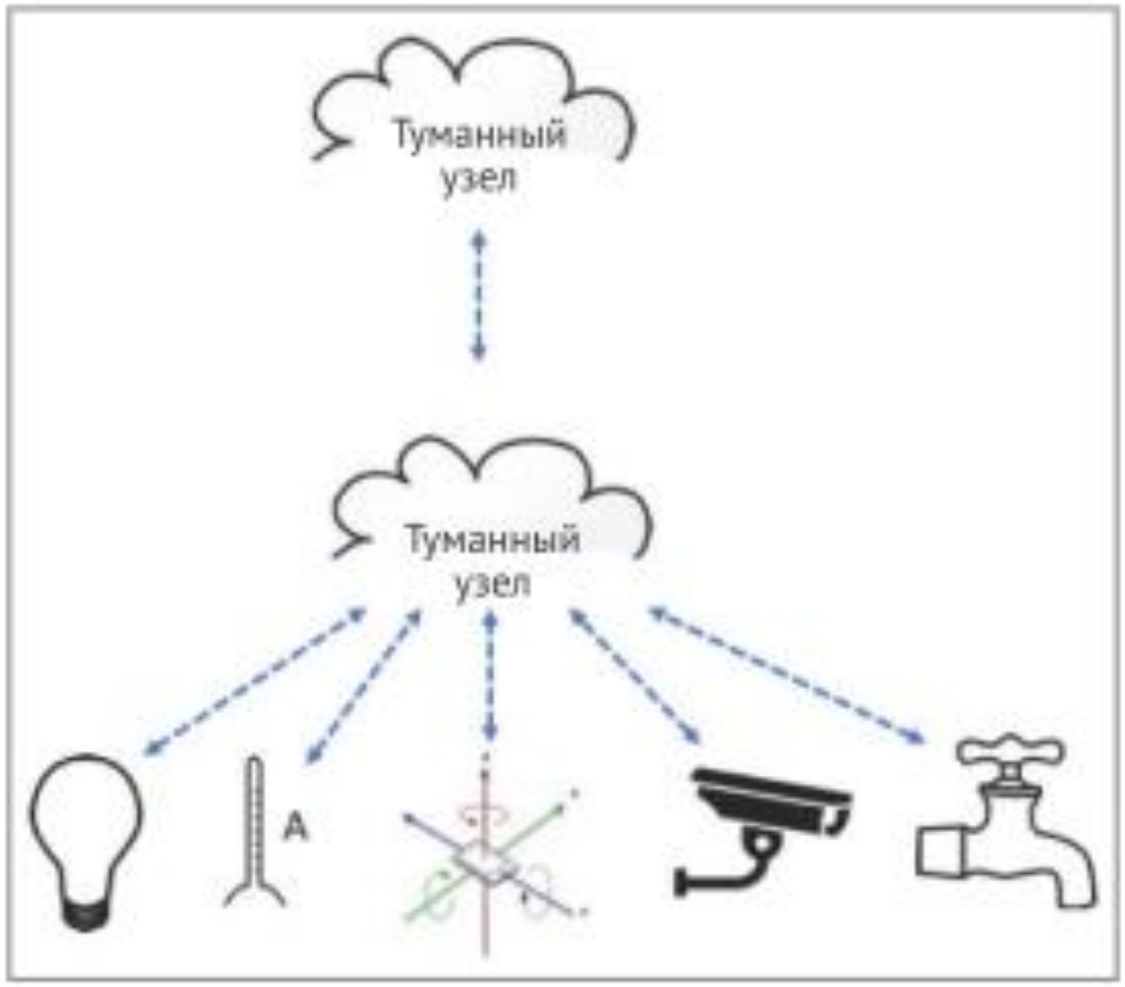
Определяющие факторы моделей основаны на следующем:

- уменьшение объема данных – например, система собирает неструктурированные видеоданные с тысяч датчиков или камер, агрегирует данные и ищет конкретные события в режиме реального времени. Если это так, то сокращение набора данных будет значительным, так как тысячи камер будут ежедневно производить сотни ГБ данных, а туманным узлам нужно будет перевести большие объемы данных в простые формы: «да», «нет», «опасность», «токены безопасности события»;
- количество граничных устройств – если система IoT представляет из себя всего лишь один датчик, то набор данных мал и может не оправдывать использование граничного туманного узла. Однако, если число датчиков растет или, в худшем случае, рост количества датчиков непредсказуем и динамичен, то туманная топология может потребовать масштабирования вверх или вниз динамически.

Туманные топологии

возможности туманного узла – в зависимости от топологии и стоимости некоторые узлы могут лучше подходить для подключения к системам WPAN, тогда как другие узлы в иерархии могут иметь дополнительные возможности по обработке для машинного обучения, распознавания образов или обработки изображений. Примером могут быть граничные туманные узлы, которые управляют безопасной сеткой Zigbee и имеют специальное оборудование для аварийных ситуаций или безопасности WPAN. Выше этого уровня будет существовать узел обработки тумана, который будет иметь дополнительную оперативную память и GPGPU для поддержки потоков необработанных данных из шлюзов WPAN;
надежность системы – архитектору может потребоваться рассмотрение форм отказа в модели IoT. Если один крайний туманный узел дал сбой, другой мог бы занять его место для выполнения какого-либо действия или сервиса. Этот случай важен в жизненно-критических случаях или при работе в реальном времени. Таким же образом дополнительные туманные узлы могут подключаться по требованию; избыточные узлы могут потребоваться в ситуациях обеспечения отказоустойчивости. В случае отсутствия дополнительных избыточных узлов некоторая обработка может совместно производиться соседними узлами за счет использования системных ресурсов и задержки, но система будет продолжать функционировать. В случае сбоя туманного узла или сбоя связи с узлом сторожевой таймер сигнализирует о событии сбоя для облака и может выполнять локальные критические действия. Хорошим примером является случай, когда туманный узел сбоит при контроле трафика на шоссе; соседний узел может увидеть
отказ точки, предупредить облако о событии и подать сигнал на щит на шоссе, чтобы уменьшить скорость.
Простейшим решением для тумана является блок обработки на краю (шлюз, тонкие клиенты, маршрутизатор), расположенные рядом с матрицей датчиков. Здесь туманный узел может использоваться в качестве шлюза для сети или mesh-сети WPAN и обмениваться данными с хостом.

Туманная топология



Следующая базовая туманная топология включает облако в качестве родителя по туманной сети. Туманный узел в этом случае будет собирать данные, защищать край и выполнять обработку, необходимую для связи с облаком. Эту модель отделяет от граничных вычислений то, что сервисные и программные уровни туманного узла разделяют отношения с облачным фреймворком.

Определение цифровой платформы

Еще в 1999 г. Билл Гейтс в своей книге «Бизнес со скоростью мысли» предложил идею цифровой нервной системы предприятия.

Тогда основой такой нервной системы стала электронная почта и стремительно развивающиеся веб-технологии. Идея взаимодействия на уровне предприятия или организации была сконцентрирована на получении руководителем обратной связи от сотрудников, с предпочтением в сторону потока информации о проблемах или препятствий в работе. В целом, тогда это позволяло повысить эффективность бизнес-решений.

Определение цифровой платформы

Цифровая платформа - программная среда, в которой аппаратные средства интегрируются с прикладными решениями, повышающими эффективность всех сфер жизни общества.

SCADA

Система диспетчеризации - это программный пакет, включающий в себя системы сбора данных, их пересылки, обработки информации и отображения данных о протекании процессов автоматического управления, состояния оборудования и т.п.

SCADA

SCADA (аббр. от англ. Supervisory Control And Data Acquisition – диспетчерское управление и сбор данных) – программный пакет, предназначенный для разработки или обеспечения работы в реальном времени систем сбора, обработки, отображения и архивирования информации об объекте мониторинга или управления.

SCADA является АСУ ТП верхнего уровня, АСКУЭ, системы экологического мониторинга, научного эксперимента, автоматизации здания и т. д.

SCADA

SCADA-системы решают следующие задачи:

- Обмен данными с «устройствами связи с объектом» (то есть с промышленными контроллерами и платами ввода-вывода) в реальном времени через драйверы.
- Обработка информации в реальном времени.
- Логическое управление.
- Отображение информации на экране монитора в удобной и понятной для человека форме.
- Ведение базы данных реального времени с технологической информацией.
- Аварийная сигнализация и управление тревожными сообщениями.
- Подготовка и генерирование отчетов о ходе технологического процесса.
- Осуществление сетевого взаимодействия между SCADA ПК.
- Обеспечение связи с внешними приложениями (СУБД, электронные таблицы, микроконтроллеры).

SCADA

Требования, предъявляемые к SCADA-системам

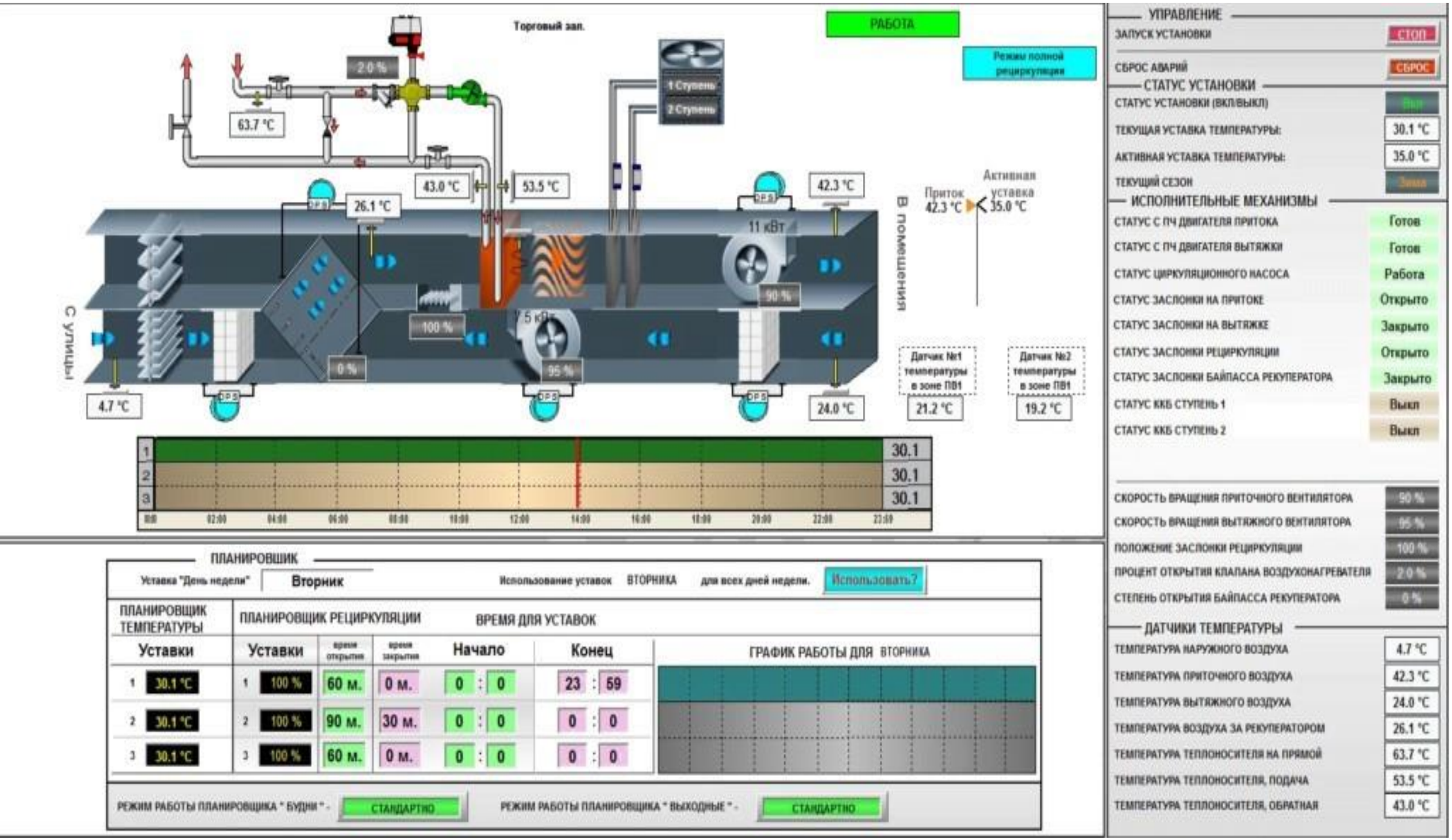
- надёжность системы (технологическая и функциональная);
- безопасность управления;
- точность обработки и представления данных;
- простота расширения системы.

SCADA

SCADA-системы предназначены для:

- более точного ведения технологического процесса, стабилизации качества продукции и уменьшения процента брака;
- уменьшения действий оператора, с целью концентрации его внимания на выработке более эффективных решений по управлению процессом;
- программного контроля правильности выработки команд дистанционного управления и, следовательно, минимизации количества ошибок, допускаемых операторами;
- автоматического выявления и оповещения об аварийных и предаварийных ситуациях;
- предоставления полной необходимой информации персоналу в виде различных отчётов;
- анализа факторов, влияющих на качество готовой продукции.

SCADA


Торговый зал.
РАБОТА
Режим полной рециркуляции
1 Ступень
2 Ступень
20 %
63.7 °C
43.0 °C
53.5 °C
26.1 °C
42.3 °C
11 кВт
90 %
100 %
0 %
95 %
С улицы
В помещения
Приток 42.3 °C
Активная уставка 35.0 °C
4.7 °C
24.0 °C
Датчик №1 температуры в зоне ПВ1
21.2 °C
Датчик №2 температуры в зоне ПВ1
19.2 °C
1
2
3
30.1
30.1
30.1
00:00
02:00
04:00
06:00
08:00
10:00
12:00
14:00
16:00
18:00
20:00
22:00
23:59
ПЛАНИРОВЩИК
Уставка "День недели"
Вторник
Использование уставок ВТОРНИКА для всех дней недели.
Использовать?
ПЛАНИРОВЩИК ТЕМПЕРАТУРЫ
ПЛАНИРОВЩИК РЕЦИРКУЛЯЦИИ
ВРЕМЯ ДЛЯ УСТАВОК
Уставки
Уставки
время открытия
время закрытия
Начало
Конец
ГРАФИК РАБОТЫ ДЛЯ ВТОРНИКА
1 30.1 °C
1 100 %
60 м.
0 м.
0 : 0
23 : 59
2 30.1 °C
2 100 %
90 м.
30 м.
0 : 0
0 : 0
3 30.1 °C
3 100 %
60 м.
0 м.
0 : 0
0 : 0
РЕЖИМ РАБОТЫ ПЛАНИРОВЩИКА " БУДНИ " -
СТАНДАРТНО
РЕЖИМ РАБОТЫ ПЛАНИРОВЩИКА " ВЫХОДНЫЕ " -
СТАНДАРТНО
УПРАВЛЕНИЕ
ЗАПУСК УСТАНОВКИ
СТОП
СБРОС АВАРИЙ
СБРОС
СТАТУС УСТАНОВКИ
СТАТУС УСТАНОВКИ (ВКЛ/ВЫКЛ)
ТЕКУЩАЯ УСТАВКА ТЕМПЕРАТУРЫ:
30.1 °C
АКТИВНАЯ УСТАВКА ТЕМПЕРАТУРЫ:
35.0 °C
ТЕКУЩИЙ СЕЗОН
ИСПОЛНИТЕЛЬНЫЕ МЕХАНИЗМЫ
СТАТУС С ПЧ ДВИГАТЕЛЯ ПРИТОКА
Готов
СТАТУС С ПЧ ДВИГАТЕЛЯ ВЫТЯЖКИ
Готов
СТАТУС ЦИРКУЛЯЦИОННОГО НАСОСА
Работа
СТАТУС ЗАСЛОНКИ НА ПРИТОКЕ
Открыто
СТАТУС ЗАСЛОНКИ НА ВЫТЯЖКЕ
Закрыто
СТАТУС ЗАСЛОНКИ РЕЦИРКУЛЯЦИИ
Открыто
СТАТУС ЗАСЛОНКИ БАЙПАССА РЕКУПЕРАТОРА
Закрыто
СТАТУС ККБ СТУПЕНЬ 1
Выкл
СТАТУС ККБ СТУПЕНЬ 2
Выкл
СКОРОСТЬ ВРАЩЕНИЯ ПРИТОЧНОГО ВЕНТИЛЯТОРА
90 %
СКОРОСТЬ ВРАЩЕНИЯ ВЫТЯЖНОГО ВЕНТИЛЯТОРА
95 %
ПОЛОЖЕНИЕ ЗАСЛОНКИ РЕЦИРКУЛЯЦИИ
100 %
ПРОЦЕНТ ОТКРЫТИЯ КЛАПАНА ВОЗДУХОНАГРЕВАТЕЛЯ
20 %
СТЕПЕНЬ ОТКРЫТИЯ БАЙПАССА РЕКУПЕРАТОРА
0 %
ДАТЧИКИ ТЕМПЕРАТУРЫ
ТЕМПЕРАТУРА НАРУЖНОГО ВОЗДУХА
4.7 °C
ТЕМПЕРАТУРА ПРИТОЧНОГО ВОЗДУХА
42.3 °C
ТЕМПЕРАТУРА ВЫТЯЖНОГО ВОЗДУХА
24.0 °C
ТЕМПЕРАТУРА ВОЗДУХА ЗА РЕКУПЕРАТОРОМ
26.1 °C
ТЕМПЕРАТУРА ТЕПЛОНОСИТЕЛЯ НА ПРЯМОЙ
63.7 °C
ТЕМПЕРАТУРА ТЕПЛОНОСИТЕЛЯ, ПОДАЧА
53.5 °C
ТЕМПЕРАТУРА ТЕПЛОНОСИТЕЛЯ, ОБРАТНАЯ
43.0 °C

SCADA-система обычно содержит следующие подсистемы:

- Драйверы или серверы ввода-вывода — программы, обеспечивающие связь SCADA с промышленными контроллерами, счётчиками, АЦП и другими устройствами ввода-вывода информации.
- Система реального времени — программа, обеспечивающая обработку данных в пределах заданного временного цикла с учётом приоритетов.

SCADA-система обычно содержит следующие подсистемы:

- Человеко-машинный интерфейс (HMI, англ. Human Machine Interface) – инструмент, который представляет данные о ходе процесса человеку оператору, что позволяет оператору контролировать процесс и управлять им.
- Программа-редактор для разработки человеко-машинного интерфейса.

SCADA-система обычно содержит следующие подсистемы:

- Система логического управления – программа, обеспечивающая исполнение пользовательских программ (скриптов) логического управления в SCADA-системе. Набор редакторов для их разработки.
- База данных реального времени – программа, обеспечивающая сохранение истории процесса в режиме реального времени.

SCADA-система обычно содержит следующие подсистемы:

- Система управления тревогами – программа, обеспечивающая автоматический контроль технологических событий, отнесение их к категории нормальных, предупреждающих или аварийных, а также обработку событий оператором или компьютером.
- Генератор отчетов – программа, обеспечивающая создание пользовательских отчетов о технологических событиях. Набор редакторов для их разработки.

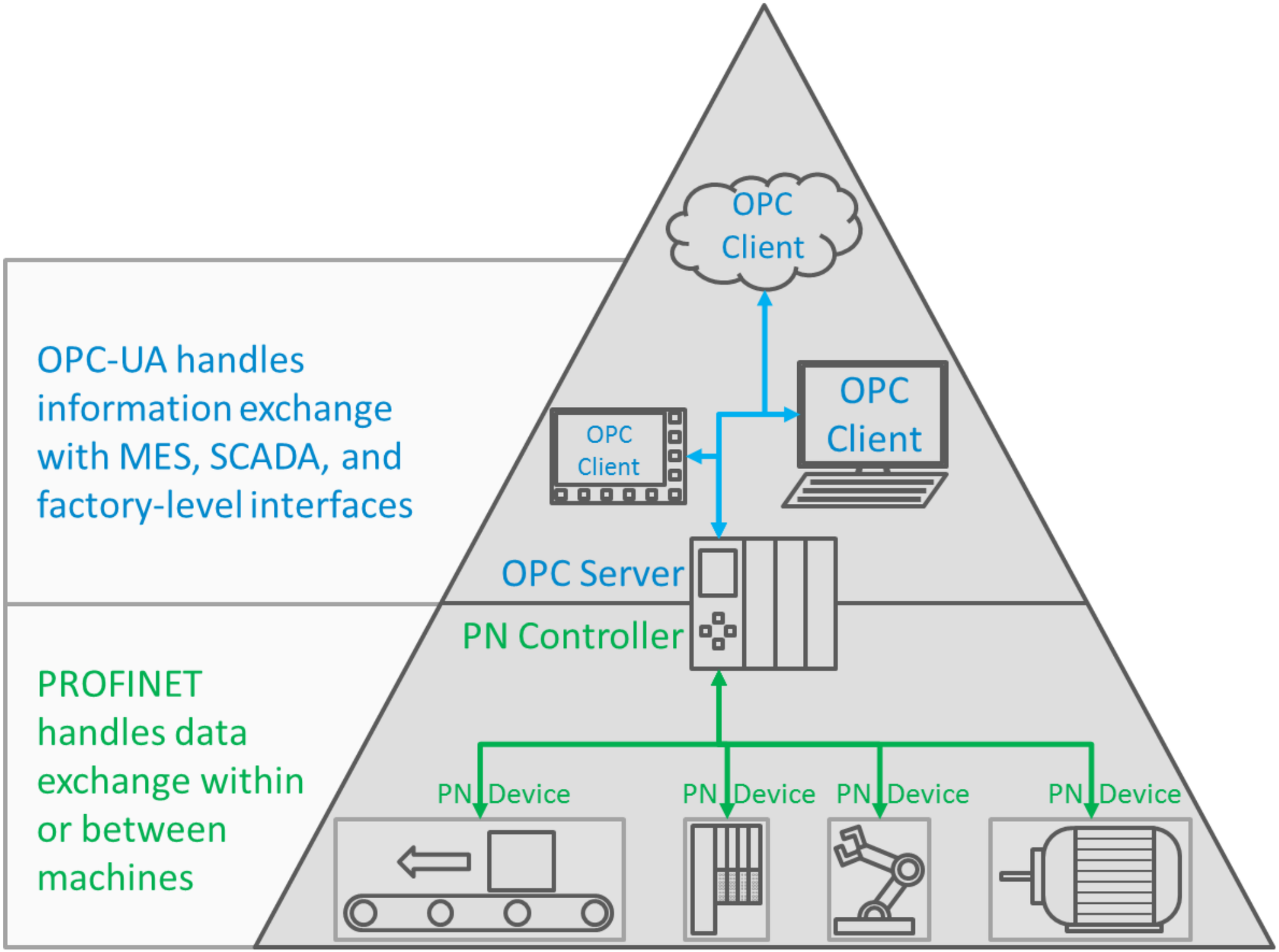
SCADA-система обычно содержит следующие подсистемы:

- Внешние интерфейсы – стандартные интерфейсы обмена данными между SCADA и другими приложениями. Обычно OPC, DDE, ODBC, DLL и т. д.

OPC UA

OPC Unified Architecture (англ. Унифицированная архитектура OPC) – спецификация, определяющая передачу данных в промышленных сетях и взаимодействие устройств в них.

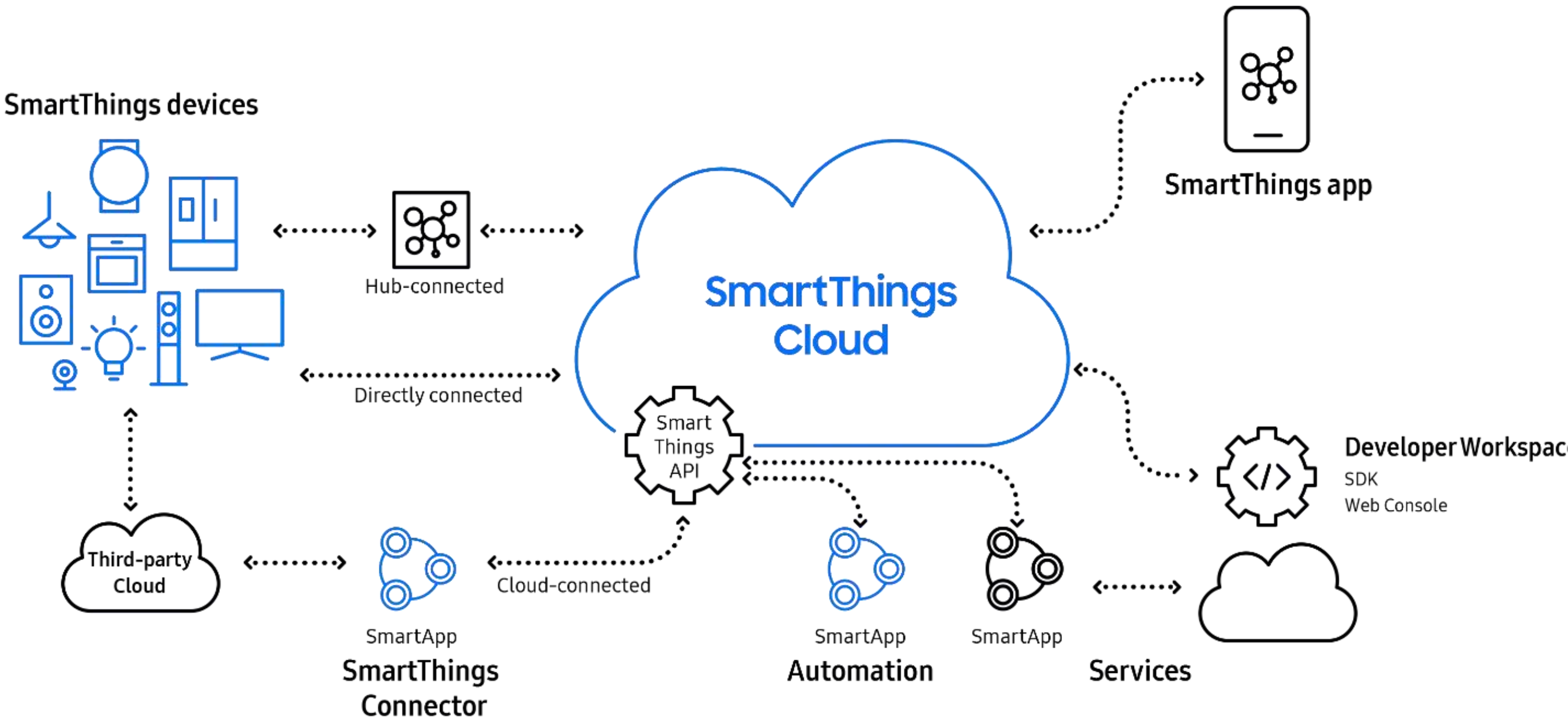
OPC UA


OPC-UA handles information exchange with MES, SCADA, and factory-level interfaces
PROFINET handles data exchange within or between machines
OPC Client
OPC Client
OPC Client
OPC Server
PN Controller
PN Device
PN Device
PN Device
PN Device

SmartThings

Система SmartThings - это широкомасштабный автоматизированный комплекс, представляющий собой центральный Хаб, а также целый ряд камер и датчиков. Все эти девайсы и гаджеты в рамках единой системы контролируется с помощью специального мобильного одноименного приложения, которое загружается непосредственно на мобильное устройство.

Архитекутра SmartThings



Способы подключения

Подключать устройства к платформе вы можете следующими способами:

- Hub-connected – через хаб-концентратор, актуально для устройств, которые своего выхода в Интернет не имеют, а соединяются через традиционные для такого рода систем беспроводные сети наподобие ZigBee.

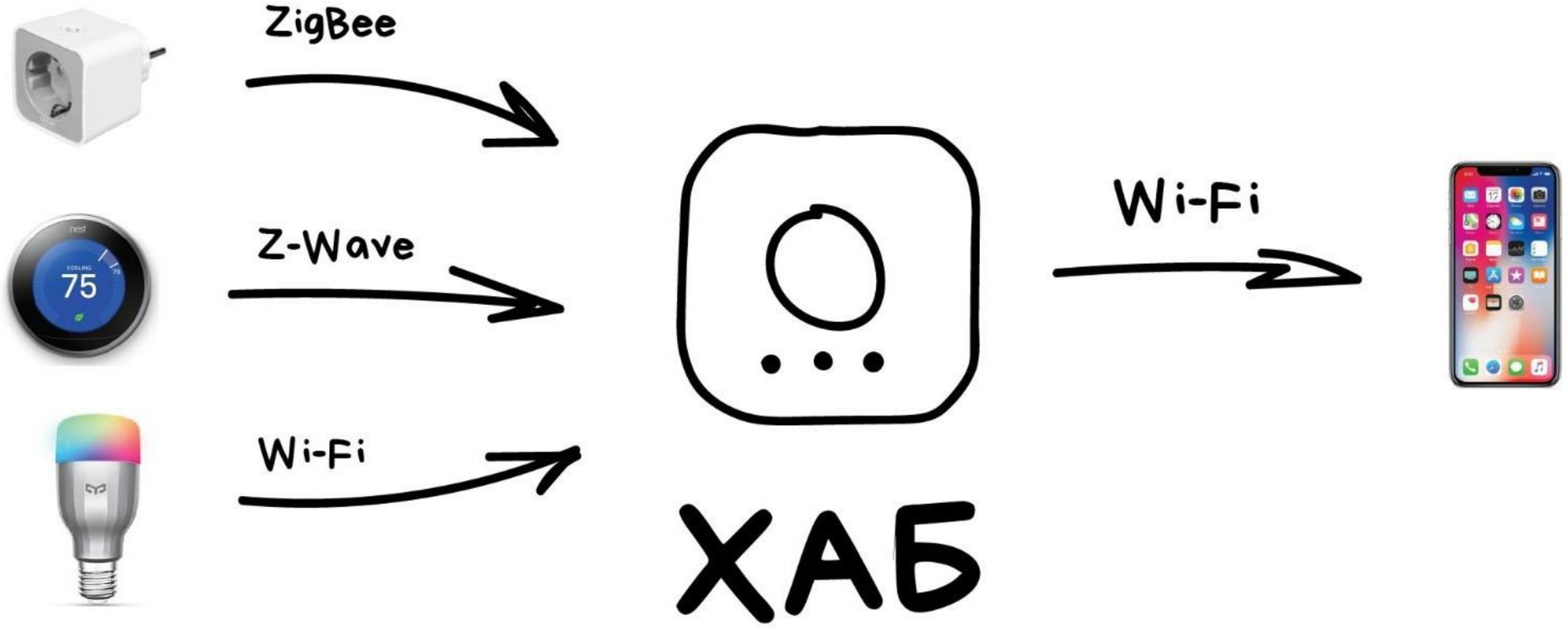
Способы подключения

Подключать устройства к платформе вы можете следующими способами:

- Directly-connected – через Интернет напрямую к облаку. Недавно SmartThings выпустили видеокамеру, лампочку и умную розетку, и они работают без хаба - через WiFi. Именно такой вариант мы будем реализовывать в данном руководстве.
- Cloud-connected – через стороннее облако для тех устройств, у которых уже есть своя облачная экосистема. Это сложнее, поэтому такой вариант рассматривать не будем.

Способы подключения

Подключать устройства к платформе вы можете следующими способами:



Компоненты платформы

Хаб

Обычно для подключения разнообразной аппаратуры у вас дома должен стоять специальный маленький сервер «Умного дома», поддерживающий различные протоколы. Он же - хаб, концентратор, шлюз - названия бывают разные.

Приложение

SmartThings App - это приложение, которое можно установить на основные мобильные ОС и с его помощью: добавлять новые устройства в систему, контролировать уже добавленные, назначать правила автоматизации, получать уведомления через push и SMS, и многое другое.

Компоненты платформы

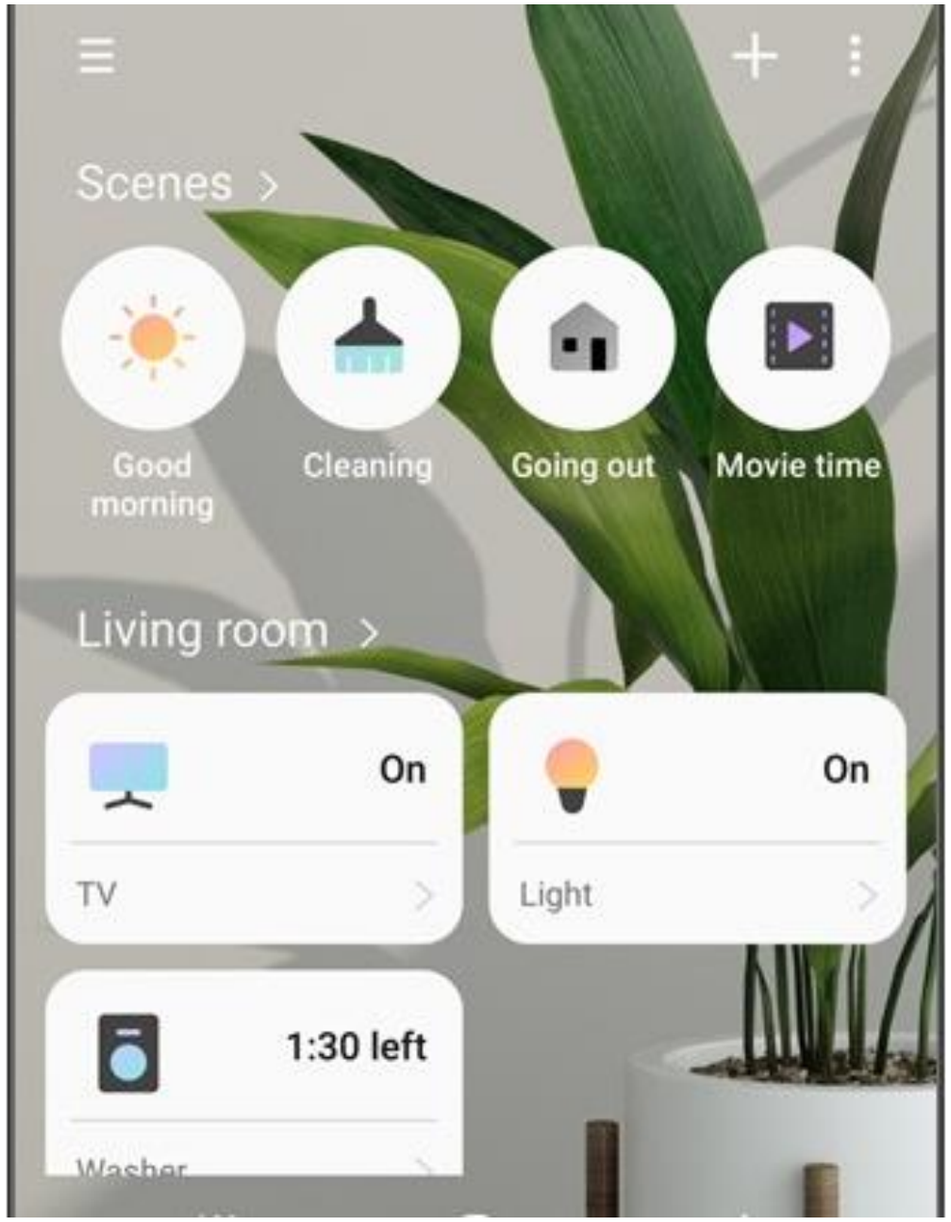
Абстракции

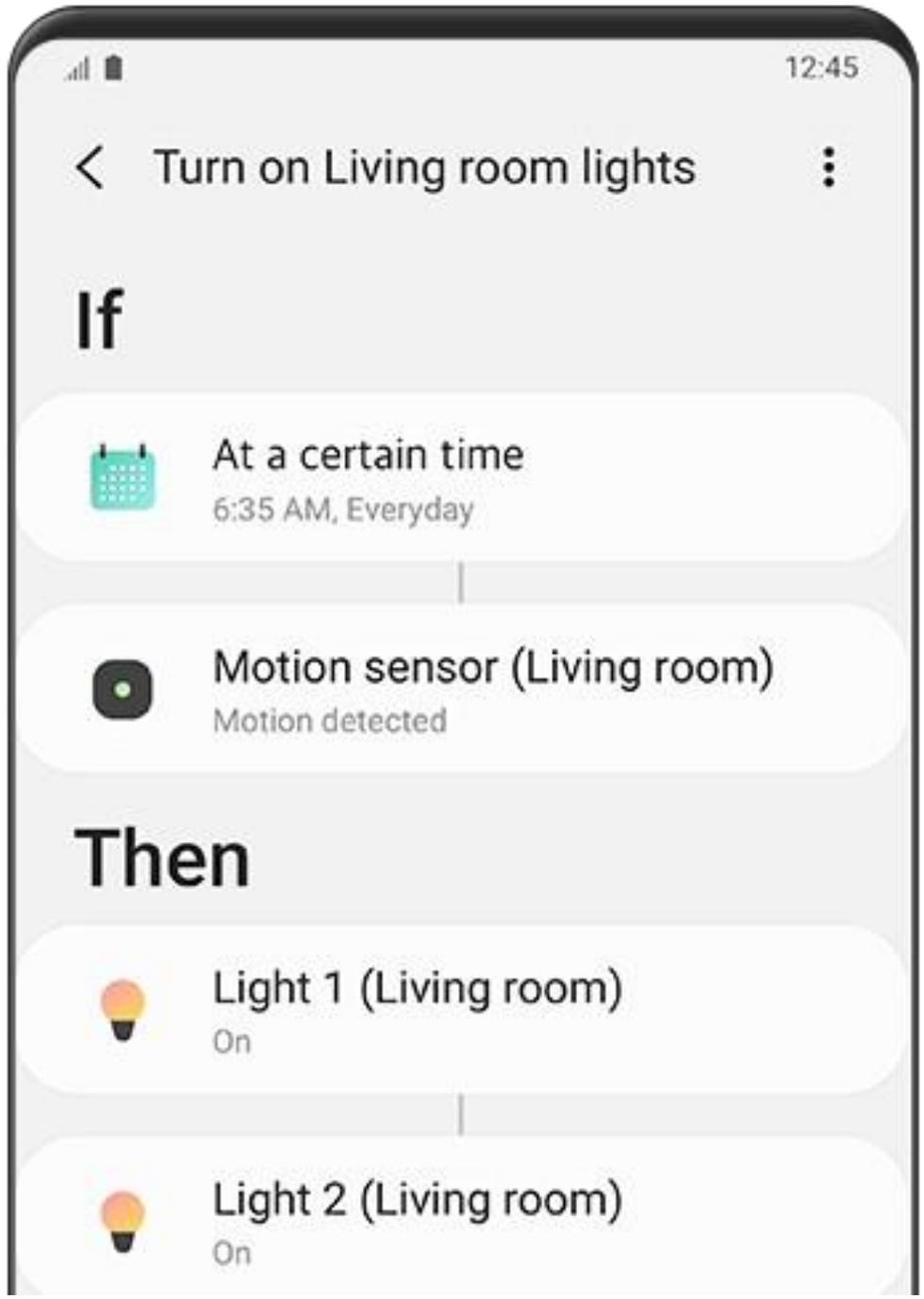
В SmartThings, как и во многих других платформах умного дома, есть следующие абстракции: комнаты, сцены, сценарии.

Устройства группируются в комнаты (Rooms). Вы просто присоединяете в приложении те или иные гаджеты к одной комнате, и называете её «Кухня», «Ванная», «Прихожая».

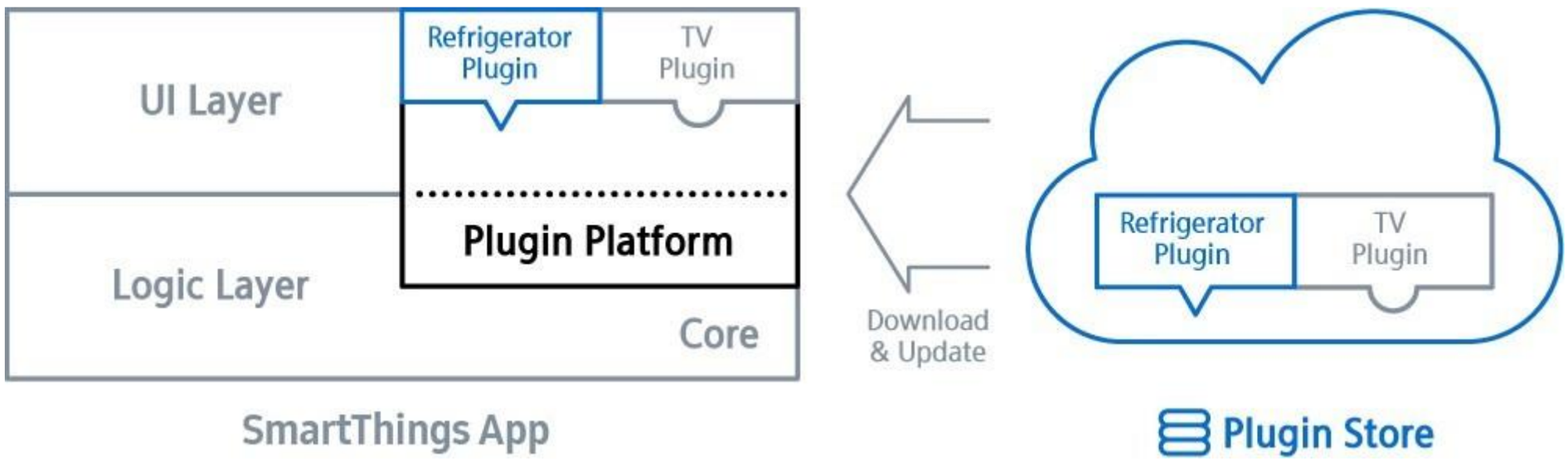
Ещё есть сцены (Scenes) - это ситуации: «сон», «отдых», «работа», «просмотр фильма» – каждая из ситуаций характеризуется своим набором состояний устройств.

Интерфейс Samsung SmartThings



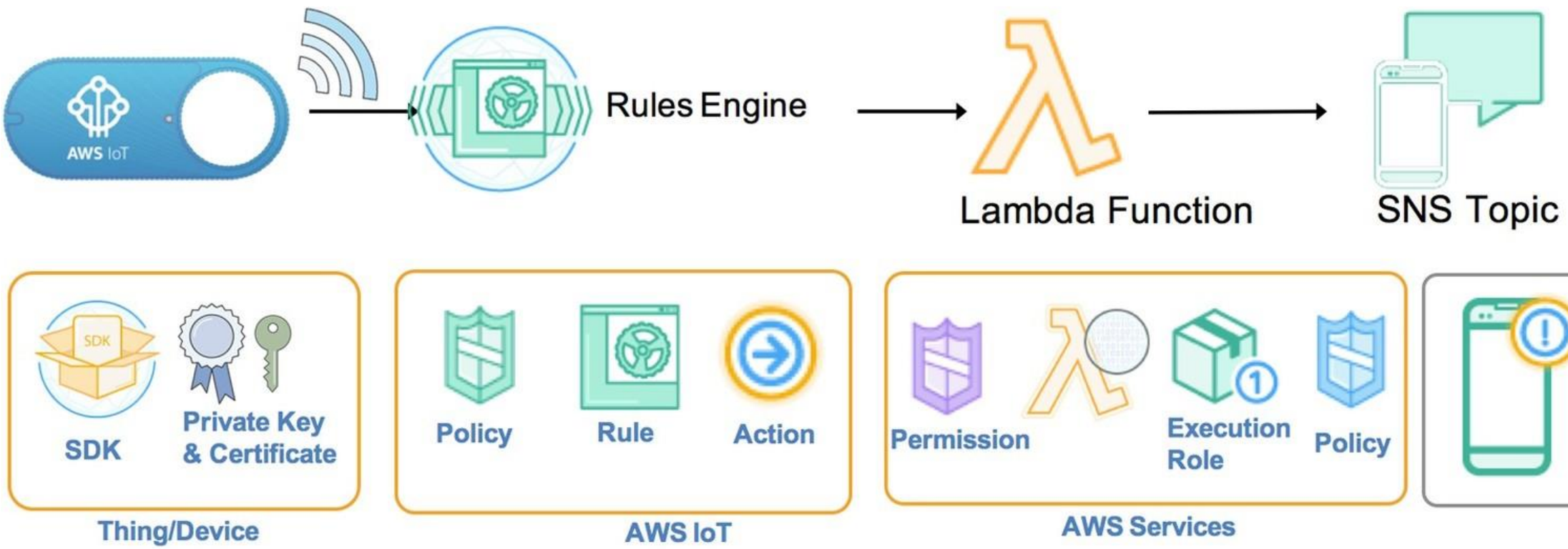


Плагины Samsung SmartThings



Архитектура цифровой платформы интернета вещей

На примере Azure



Rule: Select * from ‘iotbutton/+’