

## Общий обзор темы, над которой работали

- область применения
- примеры использования (use cases)
- перспективность
- описание поставленных целей и выполненных заданий
- резюме по достигнутым результатам
- предложения по развитию темы для себя и/или других команд

## Сам доклад:

Интернет вещей (*Internet of Things, IoT*) — концепция вычислительной сети физических объектов («вещей»), оснащённых встроенными технологиями для взаимодействия друг с другом или с внешней средой, рассматривающая организацию таких сетей как явление, способное перестроить экономические и общественные процессы, исключаящее из части действий и операций необходимость участия человека.

Концепция сформулирована в 1999 году как осмысление перспектив широкого применения средств радиочастотной идентификации для взаимодействия физических объектов между собой и с внешним окружением. Наполнение концепции «интернета вещей» многообразным технологическим содержанием и внедрение практических решений для её реализации начиная с 2010-х годов считается восходящим трендом в информационных технологиях, прежде всего, благодаря повсеместному распространению беспроводных сетей, появлению облачных вычислений, развитию технологий межмашинного взаимодействия, началу активного перехода на IPv6 и освоению программно-конфигурируемых сетей.

## В нашей работе использовался IoT проект AllJoyn:

**AllJoyn** — открытый, платформонезависимый проект, в рамках которого разработан инструментарий для создания приложений и сервисов, которые могут автоматически группироваться в P2P-сеть, подключаясь к соседним устройствам, используя прямое соединение по WiFi или Bluetooth. В итоге AllJoyn позволяет сформировать самоорганизующуюся P2P-сеть, функционирующую только за счет прямой связности между её участниками, не привязанную к каким-либо централизованным узлам и не требующую наличия доступа к интернет или к сотовой сети.

Исходные тексты проекта распространяется в рамках лицензии Apache. Рабочие прототипы приложений доступны для Microsoft Windows, Linux, Android, iOS, OS X, OpenWRT. Из языков программирования поддерживаются C, C++, Java, JavaScript и д.р..

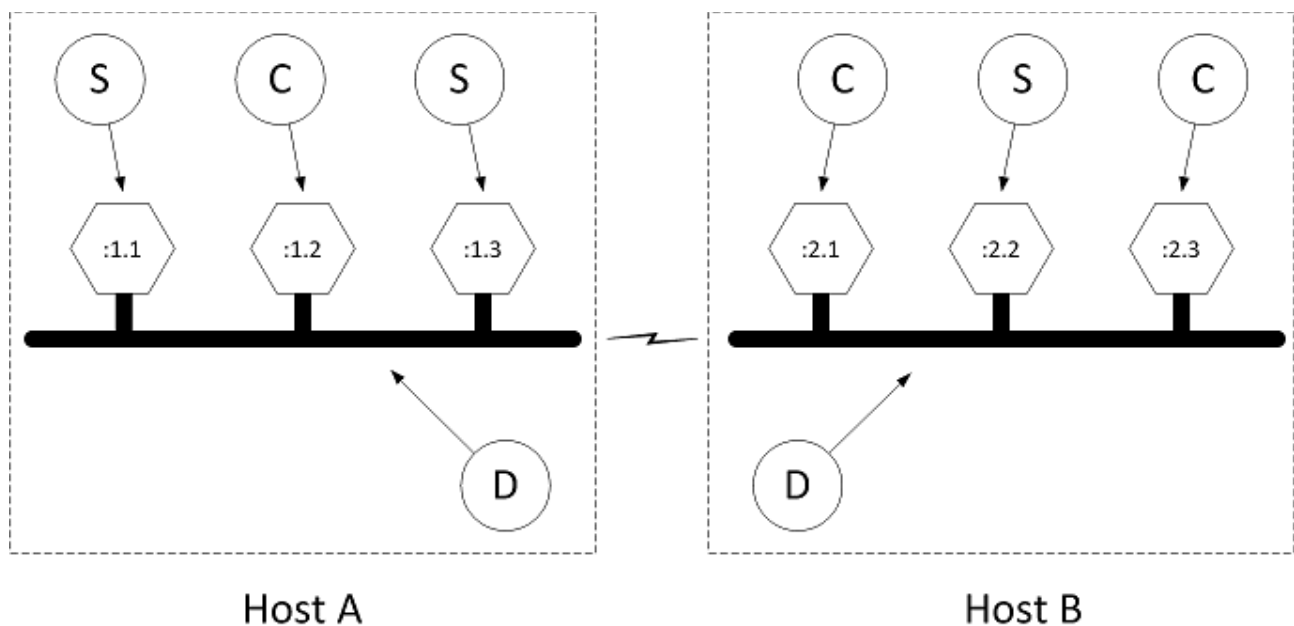
Из особенностей AllJoyn отмечается:

- Прозрачное определение доступных в области видимости устройств и сервисов;
- Поддержка аутентификации и шифрования трафика;
- Автоматическое управление маршрутизацией и передачей сообщений внутри p2p-сети;
- Объектно-ориентированная модель разработки приложений;
- Низкие задержки при передаче данных;

- Поддержка сжатия заголовков;
- Возможность использования как надежного транспортного протокола, гарантирующего доставку, так и очень быстрого ненадежного способа (например, может использоваться для игр).
- Поддержка создания связи вида "точка - многоточка".

Основная идея AllJoyn заключается в предоставлении простого способа по обеспечению связи между собой различных типов устройств. В качестве возможных областей применения называется создания многопользовательских игр, инструментов по организации групповой работы, различных социальных сервисов и систем обмена файлами.

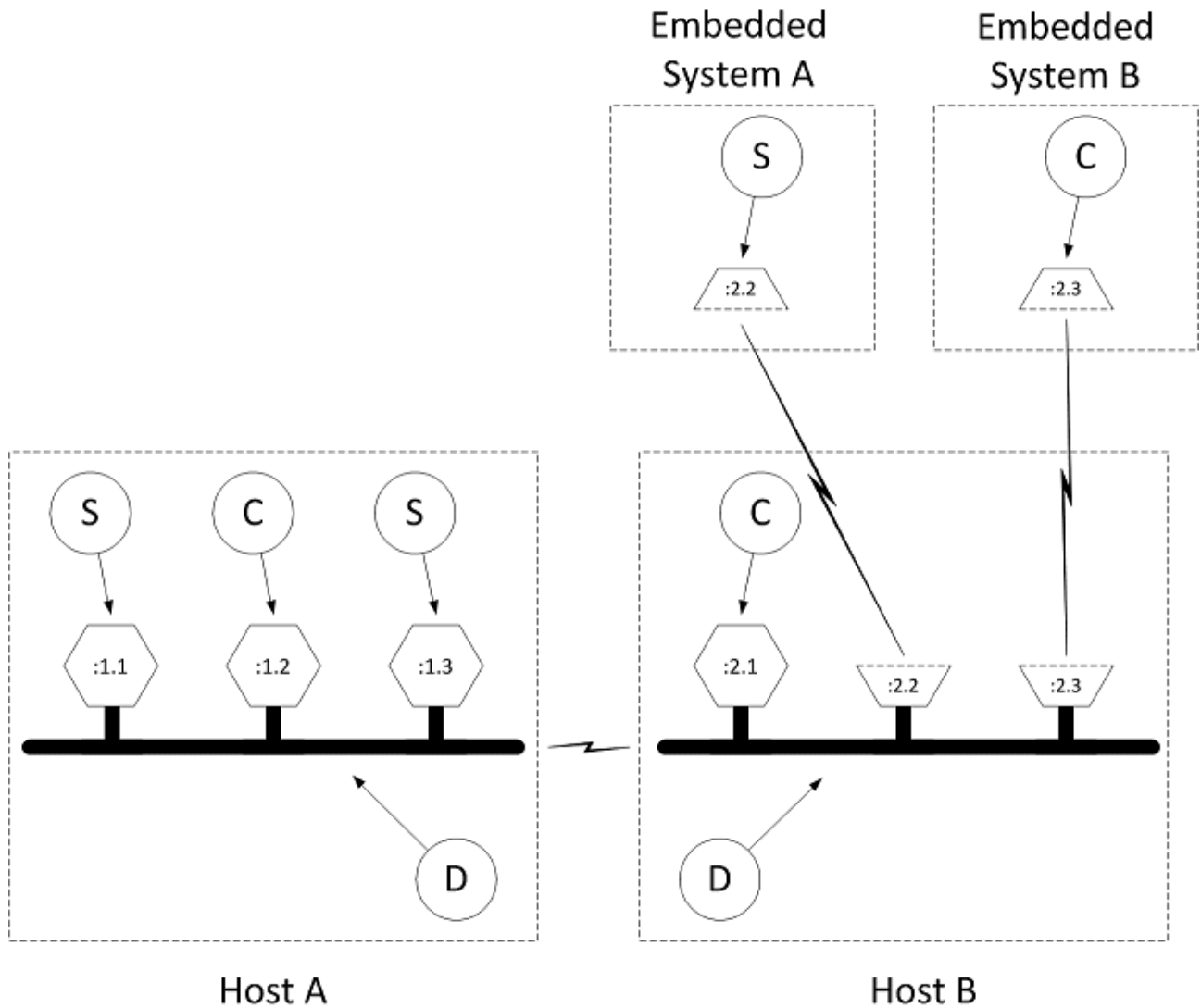
#### Общая структура AllJoyn:



Где D — демон (он же роутер), S — сервис, C — клиент.

Существуют 2 версии Библиотеки AllJoyn: AllJoyn Standart Core Library (AJSCL) и AllJoyn Thin Core Library (AJTCL)

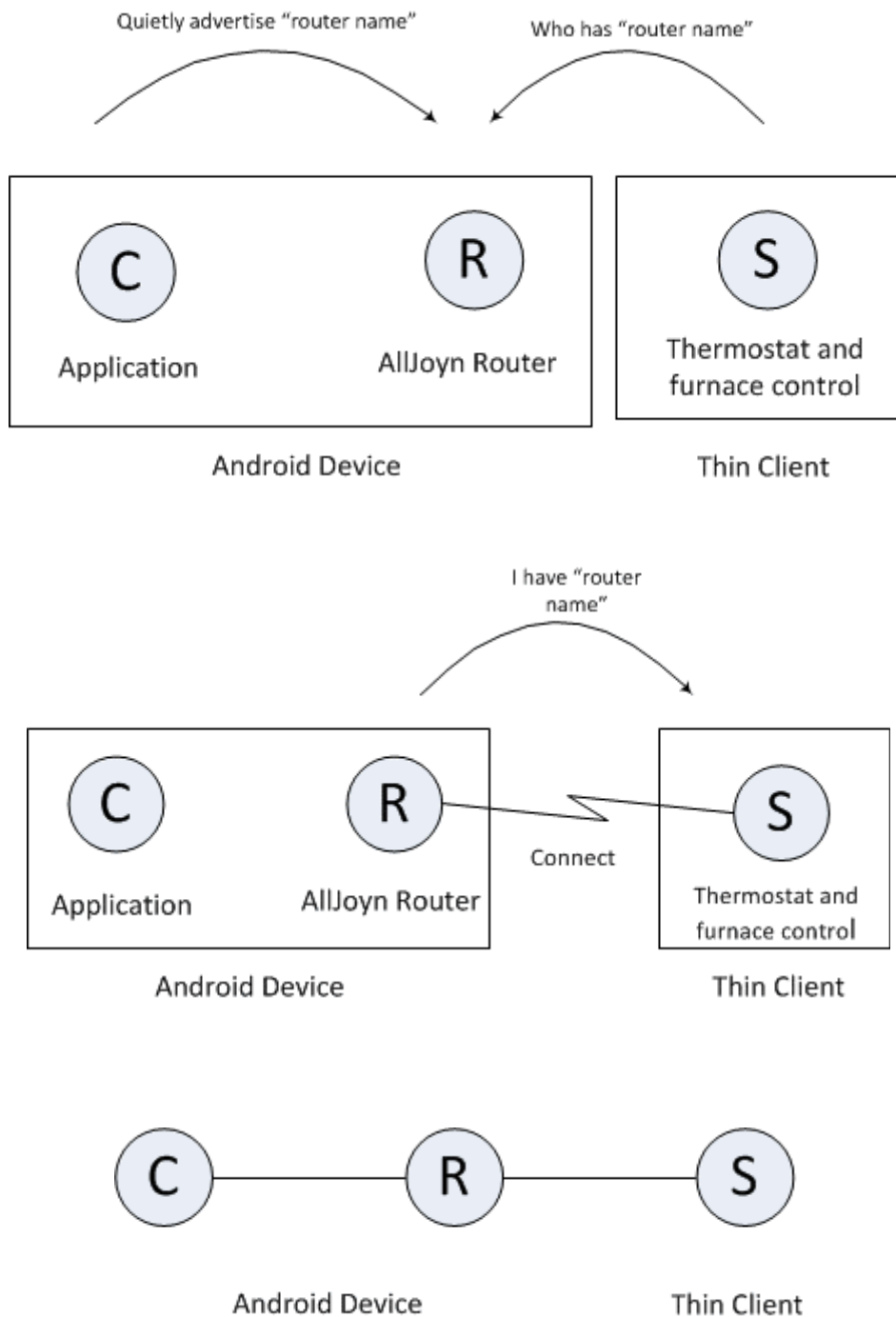
AJTCL не содержит собственного роутера, что позволяет запускать её на встраиваемых системах.

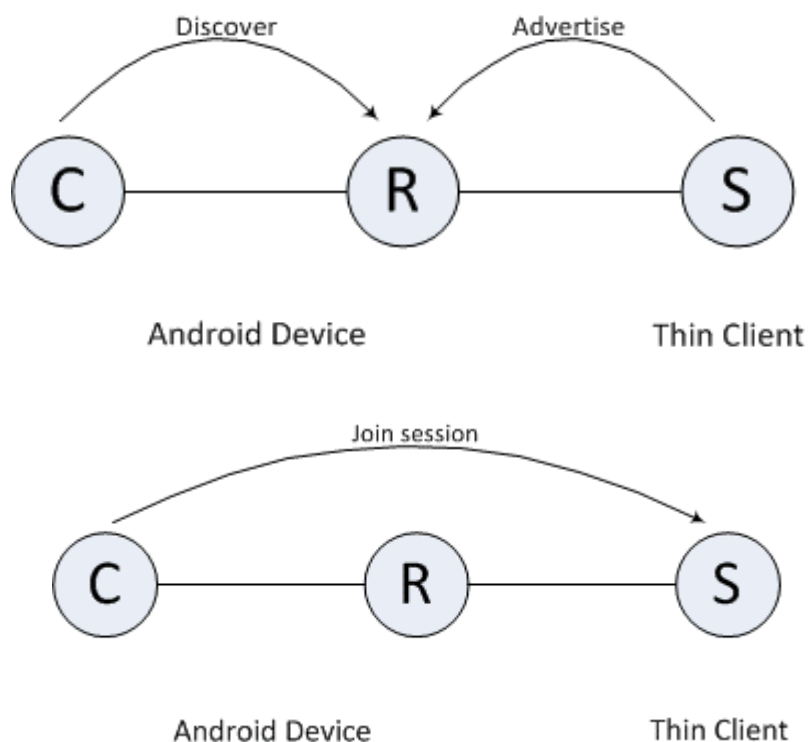


Процесс установки соединений разбивается на 3 фазы:

- 1) Фаза обнаружения
- 2) Фаза соединения
- 3) Фаза аутентификации

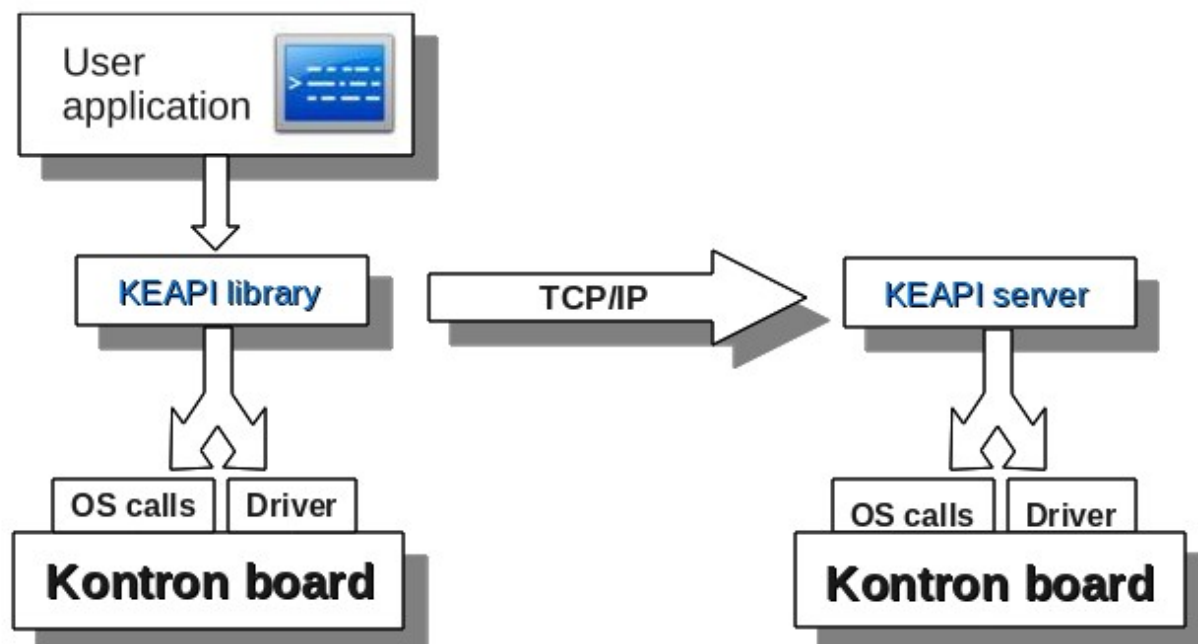
(Тут можно попробовать объяснить, что происходит)





Так же в работе использовалась технология Kontron EAPI — это библиотека, позволяющая программистам легко создавать приложения для управления и мониторинга ресурсов плат, модулей, систем и платформ Kontron.

#### Структура:



### **Область применения.**

AllJoyn может быть использован во встраиваемых устройствах, в сфере развлечения (для социальных сетей, игр, обмена информации и т. д.), для видео/аудиозвонков и т.д..

### **Примеры использования:**

Технология AllJoyn используется в разных устройствах таких как:

- **Аудио устройства** (Gramofon by Fon, Monster SoundStage, Panasonic SH-ALL1C Network Audio Connector и др.) ,
- **Операционные системы** (Windows 10, Android и др.),
- **Коммуникационные устройства** (D-Link Connected Home Hub , LG U+ Router и др.),
- **Бытовая техника** (Heatworks MODEL 1, Heaven Fresh Smart Air Purifier HF-480 и др),
- **Телевизоры** (LG OLED TV EC9800 ~ 9300 series, LG CINEMA 3D SMART TV FHD 49" LB8700 и др.).

### **Перспективность.**

В ближайшие несколько лет ожидается быстрый рост направления IoT. Cisco Systems предсказывает, что число подключенных устройств и систем (от автомашин и домашних приборов до сенсоров, промышленных систем, камер наблюдения и медицинских устройств) в мире вырастет с 25 млрд. в прошлом году до более чем 50 млрд. в 2020 г. В недавнем отчете института McKinsey Global Institute утверждается, что в 2025 г. экономический эффект IoT может достичь 11,1 млрд. долл. И что интероперабельность систем IoT будет иметь важнейшее значение для их роста.

В разработке AllJoyn принимают участие специалисты Microsoft, Qualcomm, Lenovo, LG Electronics, Symantec, Sony и Panasonic. Следовательно, данная технология имеет очень высокие перспективы.

### **Описание поставленных целей и выполненных заданий**

#### **Поставленные задания:**

1) Ознакомиться со спецификацией KEAPI 3, сделать простое приложение `kjService`, которое линкуется с KEAPI, и выводит на печать `boardName` и `manufacturingDate`.

2) Создать `kjService` по образцу `ajttl/samples/basic/basic_service.c`. Перенести туда код для получения `boardInfo`, сделать интерфейс `boardInfo`. Аналогично создать `kjClient`, который будет пользоваться «услугами», предоставляемыми `kjService` и выводить на экран `boardName`. Так же надо было ознакомиться со AJ Thin Core и написать краткий конспект на русском языке.

3) Необходимо реализовать интерфейсы сервиса для

- `KEApiGetTempSensorCount` (Количество датчиков температуры)
- `KEApiGetTempSensorValue` (Показания датчиков температуры)
- `KEApiGetTempSensorInfo` (Информация о датчиках)

и вывести через клиент данные о всех температурах в системе.

**Резюме:**

Задание было полностью выполнено. Была создана сеть на трёх виртуальных машинах, каждая машина содержала по одному компоненту (клиент, сервис, демон).

**Предложения для развития темы:**

Необходимо провести исследования производительности AllJoyn, попробовать запустить несколько сервисов с одним клиентом. Так же, необходимо доработать демон, т.к.. его работа нестабильна. Создать интерфейсы для управления платой. Создать графический интерфейс для визуального отображения состояния плат.