Direct-BT: BLE Programming with C++ & Java

Sven Göthel, *Göthel Software e.K.*

Java User Group Hamburg

November 3, 2022

Trademarks, Licenses, ...

- The *Bluetooth*® word mark and logos are registered trademarks owned by Bluetooth SIG, Inc.
- C++ is a Programming Language as described in ISO/IEC 14882:2020 , see isocpp for status details.
- *Java* is a registered trademark of Oracle and/or its affiliates.
- *TinyB* was licensed under the The MIT License (MIT) and the Intel Corporation holds its copyright from the year 2016.
- *Direct-BT* was licensed under the The MIT License (MIT) and Göthel Software e.K. and Zafena AB hold its copyright from the year 2020. Detailed copyright information in link.

Bluetooth Terminology

- Host – Bluetooth Enabled Device
- HCI – Host Controller Interface, handles all interactions between the host and the actual BT adapter (radio controller)
- L2CAP – Logical Link Control Adaption Protocol, passes data between higher layers and the baseband layers (i.e. remote endpoint)
- GAP – Generic Access Profile, send by peripheral when advertising
- ATT – Attribute Protocol, data packet description for attributes (data)
- GATT – Generic Attribute Protocol, hierarchy of served data covering all services etc using ATT
- SMP – Security Manager Protocol, defines secure pairing

Direct-BT Resources

- Direct-BT Git, Overview, Details, ...
- Supported Platforms
- Tested Bluetooth Adapter
- Using Direct-BT Applications
- Programming with Direct-BT (C++ & Java API doc and examples)
- Building Direct-BT (incl. unit-testing & cross-build)
- Direct-BT Origins
- Direct-BT, Bluetooth Server and Client Programming in C++ and Java (Part 1)
- Direct-BT C++ Implementation Details (Part 1)

Why does it exist?

- Zafena AB required robust and high-performance Bluetooth LE for GNU/Linux and Java
- *Intel's TinyB* (C++, *Java* on GNU/*Linux*) failed the requirements
  - Discovery- and connection timing
    - Connections could take up to 10s w/ *TinyB*
    - No real-time knowledge about connection-loss nor scanned devices …
  - Handle multiple devices concurrently
  - Real-time event driven low-overhead architecture
    - Uses D-Bus layer adding overhead and not allowing low-level protocol access ...
    - No native Bluetooth protocol information nor HCI error codes
    - No advertising packages visible …
  - Support for Secure Connections (SMP)
  - Allow programming peripheral devices, i.e. advertising GATT server

Why does it exist?

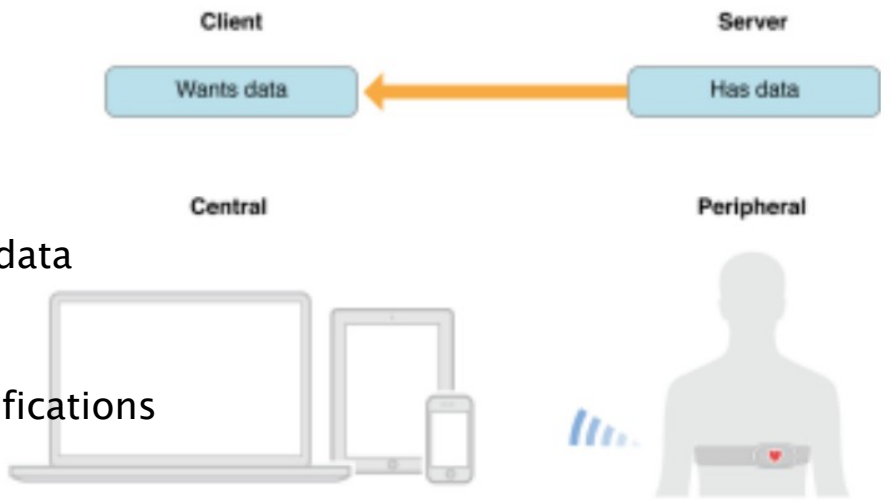
- *Direct-BT* started around April 2020, initially along *TinyB* (dropped)
- Allow support of POSIX platforms other than *Linux* w/ its *BlueZ*/kernel
- Directly using the kernel's (host) HCI and L2CAP channels w/o overhead
- We are thankful for *TinyB*, which enabled us to create this solution
- *Direct-BT* is licensed under The MIT License (MIT)

Where is it being used?

- Medical Trial (C++)
  - Real-time sensor data monitoring
  - Enabling to control / intervene
- Protocol Analysis via a man-in-the-middle repeater solution
- Connected Medical Device Terminal (Lab & Home, Java)

What is it good for?

- Usable via same API for C++ and Java
  - Implemented in C++17
  - Provides JNI Binding for seamless Java programming
- Program different BLE roles
  - *Slave* or *Peripheral*, the advertising sensor
    - Setup advertising data and security
    - Process GATT Server requests and send notifications
    - ***Server*** of sensor data
  - *Master* or *Central*, the initiator
    - ***Client*** using the Slave/Peripheral ***Server***
    - Real-time scanning incl. low level advertising data
    - Detailed security setup and initiate connection
    - Query and process GATT data and receive notifications

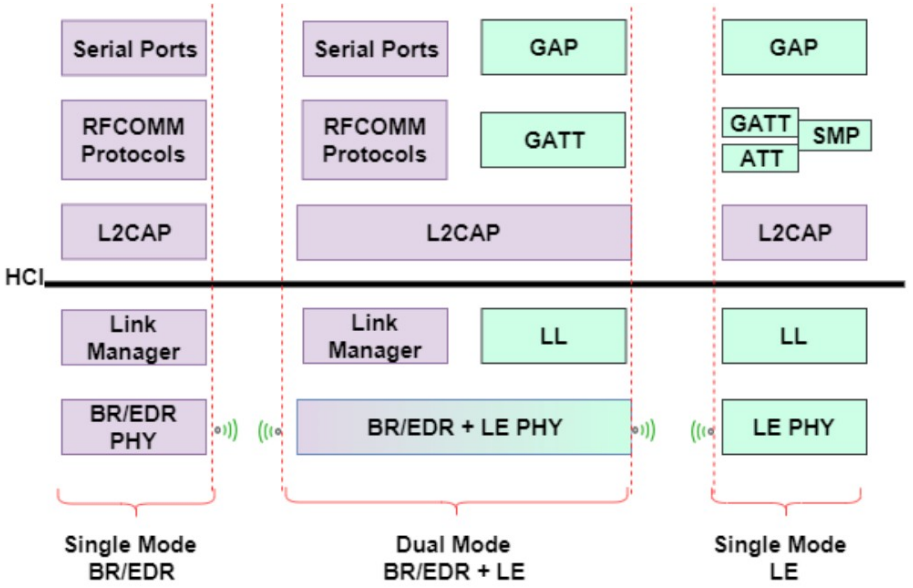


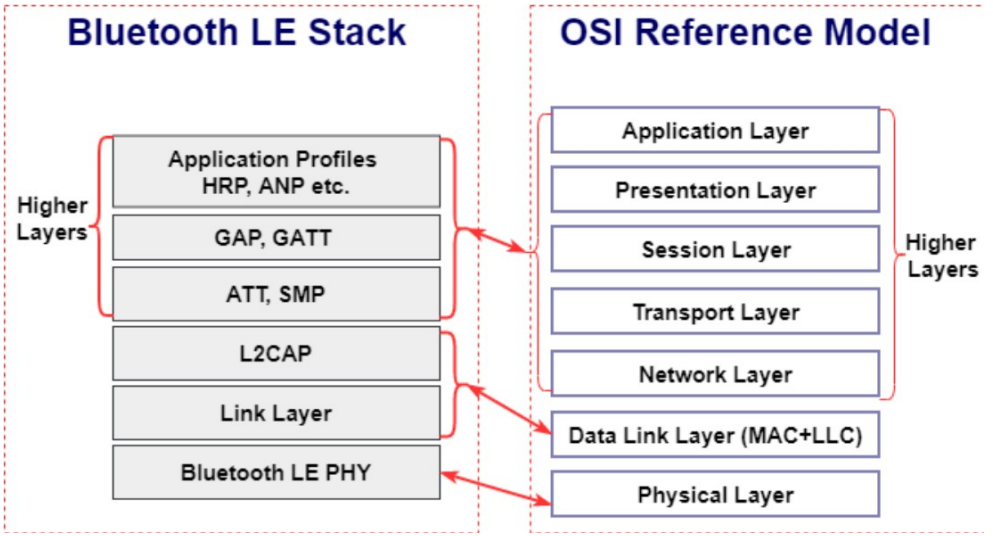
What is it good for?

- Event driven workflow from adapter management via device discovery to GATT programming
- Multiple Bluetooth adapter may be used, plug & play
- Real-time remote sensor monitoring (scan and connected)
- Transparent access across all protocols
  - HCI result codes and (connection) parameter setup
  - Advertising/GAP, SMP and GATT data
  - Real-time detection of connection loss
- Multiple concurrent connections per adapter
  - Parallel processing of real-time data streams

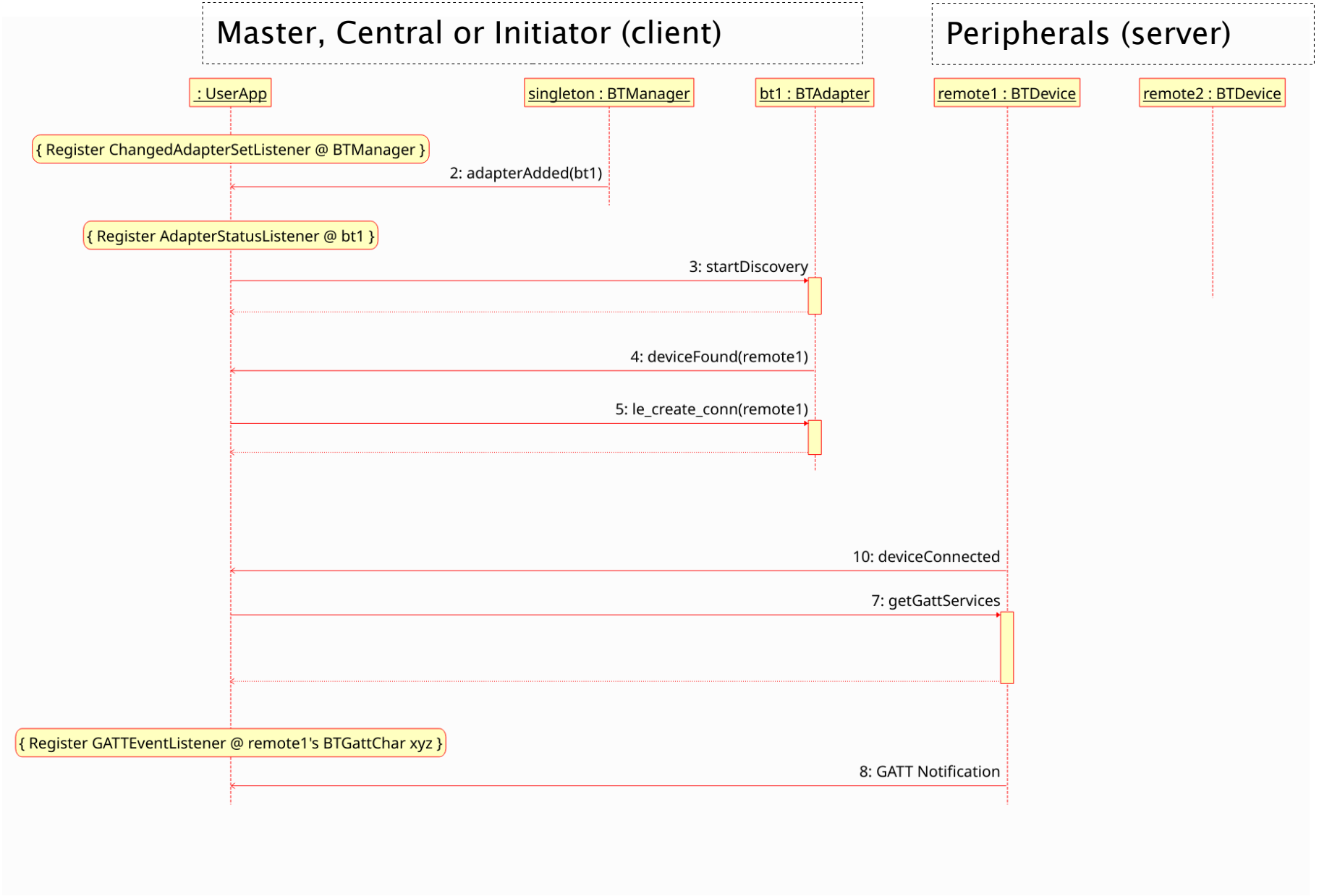
What is implemented?

- Bluetooth Specification on top of
  - OS BT adapter management,
  - its HCI and Link Layer (LL) to the adapter
- Implements HCI and L2CAP messaging via host
- Implements GAP, GATT/ATT and SMP protocols

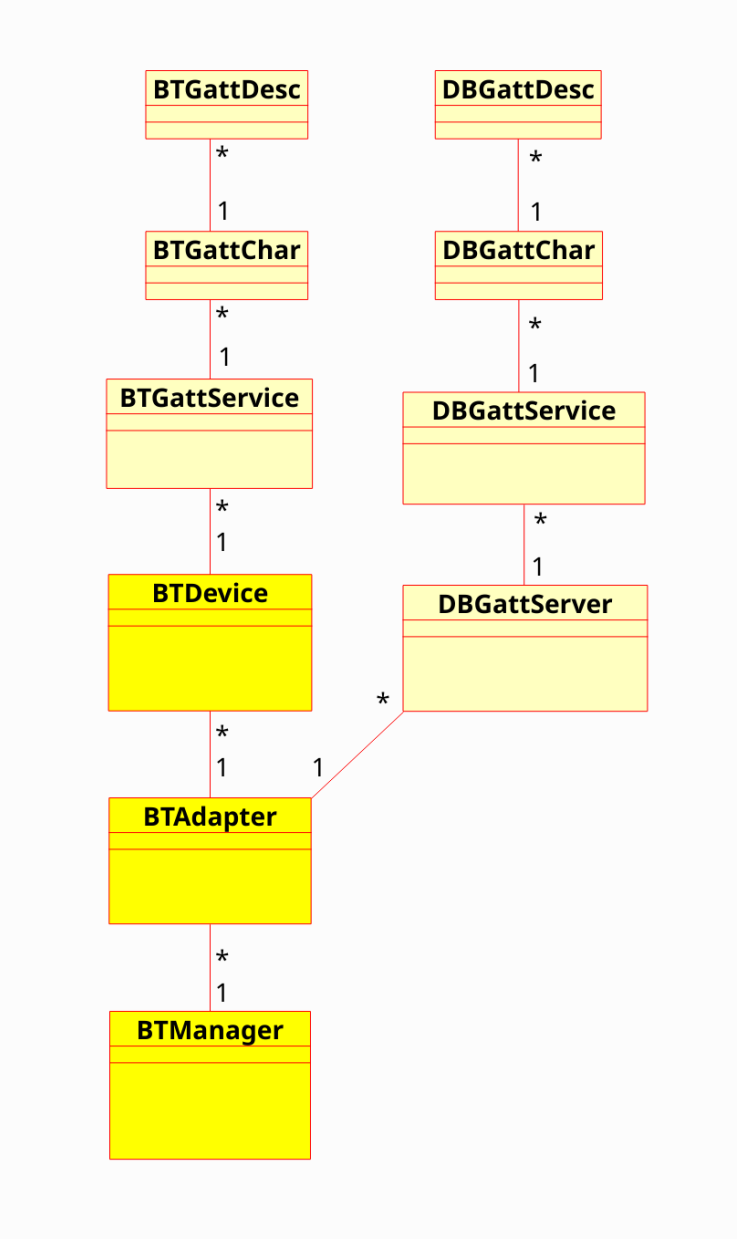




Direct-BT Sequence Chart 01



Direct-BT Type Hierarchy



Currently Supported Platforms

- Minimum language requirements
  - C++17
    - gcc 8.3 - 12.2.0, clang 10.0.0 - 14.0.6
  - Standard C Libraries
    - FreeBSD libc, GNU glibc or musl
  - Java 11 (optional)
    - OpenJDK 11 - 17
- Tested Operation Systems
  - GNU/Linux with BlueZ/kernel
    - Alpine Linux 3.16
    - Debian 10 Buster - 12 Bookworm
    - Ubuntu 18.04 - 22.04
- Prepared Operation Systems
  - FreeBSD 13.1
- Earmarked Operation Systems
  - MS Windows

Tested BT Adapter

- Bluetooth 4 Chipsets
  - CSR
  - Intel
  - BCM4345
  - BCM20xxxx
- Bluetooth 5 Chipsets
  - Intel (AX200, AX201, …)
  - Realtek RTL8761BU
  - Realtek RTL8761??
- See details in linked document @ title

Direct-BT Future

- Add Full BREDR Support
- Support further platforms
  - FreeBSD, MS Windows, ...
- Support BLE 5.2 Features
  - Iso-Channels, Audio, …

Göthel Software e.K. seeks contracted work from commercial users to maintain and enhance *Direct-BT*.

Q&A

- … ?
- … ?
- … ?

Thank You

- Direct-BT
  - https://jausoft.com/cgit/direct_bt.git/about/
- Contact
  - Sven Göthel, Göthel Software e.K.
    - https://jausoft.com
    - sgothel at jausoft dot com