Verizon IoT Integration to ThingSpace: Wearables

October 2016

Author: Quentin Robinson, Carl Wang, Mimi Tam

1 Executive Summary

Wearables Technology concerns any electronic products that are designed to be worn on people, animals or even insects in order to collect data through sensors and to affect change through command actuation, depending on the capability and capacity of the wearable device. The increasingly diverse market for Wearable devices will reach over $150 billion annually by 2026 . For Wearables OEMs/Manufacturers/Vendors, quick development and fast deployment of wearables translates to optimal productivity and positive impact on bottom lines.

Verizon's Internet of Things (IoT) Product Suite and its ThingSpace Platform, running over America's most reliable 4G LTE network now brings another innovatively differentiating product to the marketspace – the Verizon Device Connectivity Platform or VDCP. VDCP is a turnkey connectivity solution that enables application developers too easily and rapidly prototype an application into a wearable product. It consists of a hardware reference platform and a thin-layer connectivity adaptation software kit that are pre-integrated and certified to run on the Verizon network.

The embedded connectivity kit library was developed to simplify the process it takes to setup and configure an LTE CAT-1 Module on the Verizon network. By using the Device Connectivity Platform, a watch manufacturer can quickly integrate modem functions into their code, irrespective of what type of modem they are using.

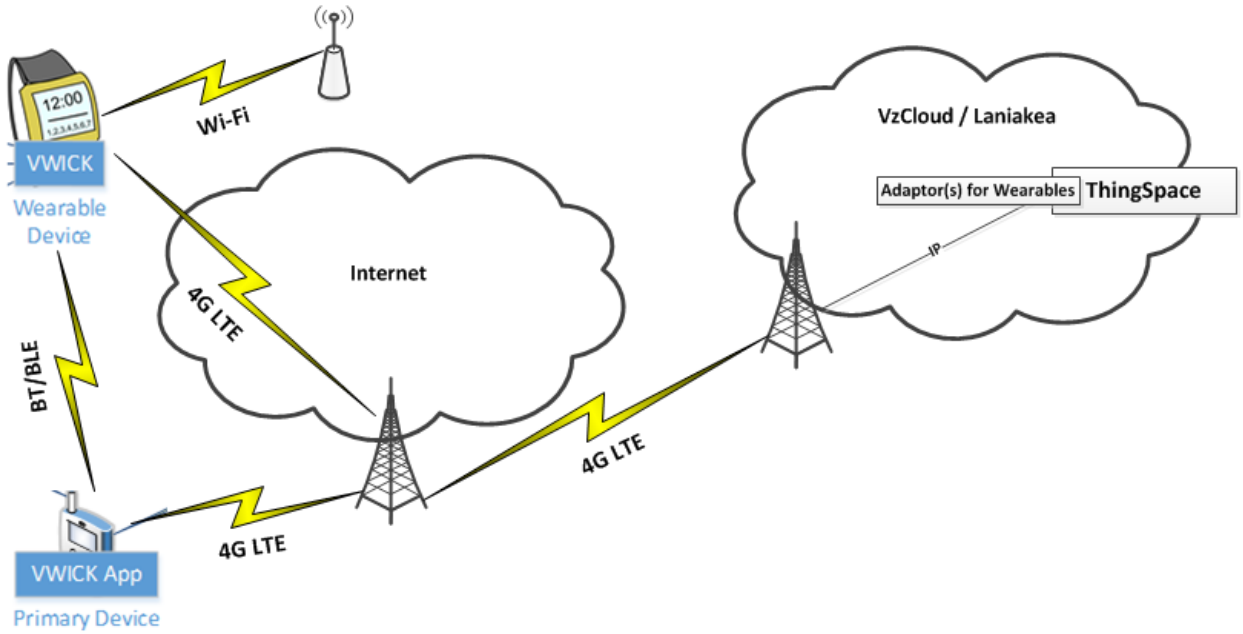


Figure 1: Wearables to ThingSpace System Architecture

The Verizon Device Connectivity Platform system features:

- *Simplified LPA (Local Profile Assistant) for eUICC activation/deactivation*
- *ThingSpace platform integration*
- *Tests SIMs with trial accounts provided*

Some example target applications:

- *GPS and fitness watches*

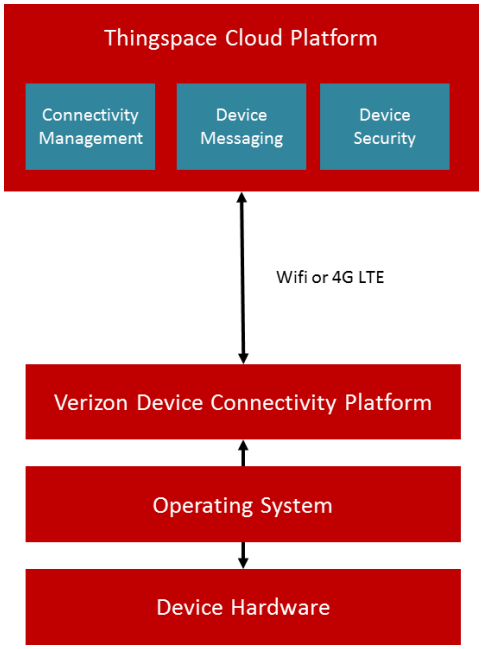
- *Fitness trackers*
- *Enterprise focused Wearables*
- *Head mounted wearables*

Components of the Wearable Reference Design:

- *Sequans CAT-1 modem*
- *ST Micro MFF2 embedded SIM*
- *Broadcom Bluetooth/WIFI*
- *Sony GPS chip-on-board design with shielding*
- *Integrated single cellular antenna design plus Bluetooth/WIFI/GPS antenna*
- *RTOS SDK that handles activation, includes a Wearable focused connection manager and device management functions*
- *Complete schematics and BOM breakout*

2 Ecosystem Enablement

Verizon's IoT product portfolio includes device and clouds elements that work for both consumer and enterprise marketspaces. In the Verizon IoT ecosystem, VDCP is the device layer that enables wearable device and app developers to create products/solutions in a snap. The Thingspace cloud platform manages the cloud connection to pass messages and device management. Together these elements provide a comprehensive solution for on-boarding a device to the Verizon network.



By adopting the VDCP, wearable solution providers or device manufacturers can quickly add application service features provided by the VDCP on top of the connectivity hardware (BT, WiFi, Cellular modem) functions by integrating the device drivers to VDCP hardware abstraction layer. Adopting VDCP also enables addition/replacement of existing hardware with new connectivity hardware irrespective of what types of modem they are using. The VDCP enables scaling of existing product lines for different market sectors.

The VDCP stack is a middleware between the device client app and the device hardware. It provides a simple way to setup and configure an LTE CAT-1 module to run on the Verizon Wireless network. Using the VDCP library stack (a RTOS software stack), the wearable application on a device can interface with any CAT-1 cellular modem driver. The device client app that incorporates the VDCP stack is pre-certified by Verizon to run on the Verizon Wireless network as well as pre-authenticated to gain access to the ThingSpace platform.

This model delivers two main areas of focus:

- Scalable Connectivity Hardware

The VDCP reference hardware defines the hardware requirements to guide the hardware implementation that conforms to the Verizon radio and network certification. For providers that are interested in supporting multiple hardware platforms, the VDCP thin-layer software adaptation layer offers the generic hardware adaptation layer (HAL) that enables isolation of design/implementation of a new hardware platform to device driver integration and/or BSP (Board Support Package) customization minimizing integration effort and time to market.

- Expanding Software Solutions

  VDCP software offers built-in power management, data transport management, and Verizon network access management in addition to the Internet Service APIs and the generic HAL. VDCP software may be integrated to the device client app by including only selected libraries to add the management capability to the wearable solution.

3 Details Behind The Verizon Device Connectivity Platform

The V-Wick library is being developed to simplify the process it takes to setup and configure an LTE CAT-1 (or CAT-M in later release) Modem Module on the Verizon network. The V-WICK library accelerates the integration of modem/WiFi/Bluetooth low energy functionality for wearable manufacturer into their solution and provides Verizon services (i.e. cellular subscription and ThingSpace).

The figure below is the high level logical view of the Verizon Device Connectivity Platform System on the Wearable Device.

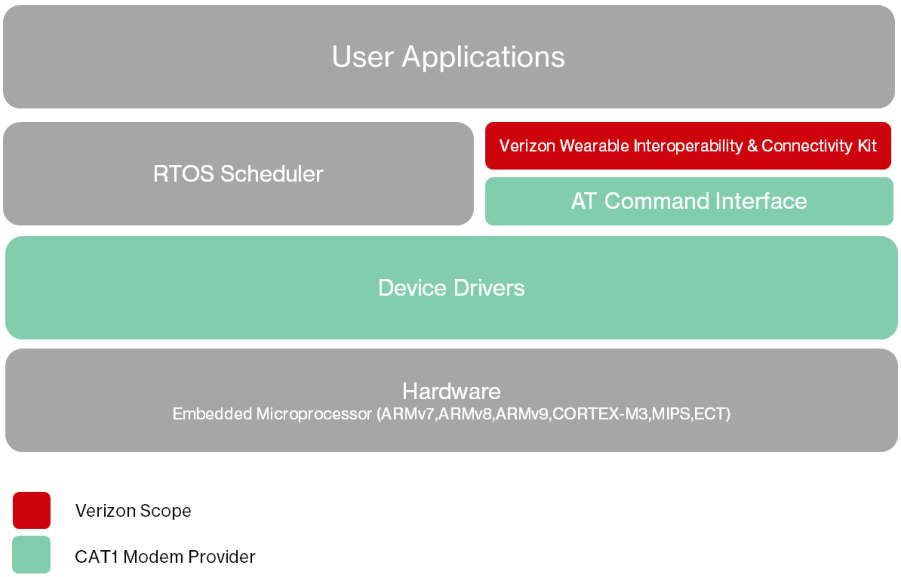


Figure 2: High Level Verizon Device Connectivity Platform Software Stack System Overview

The VDCP software stack consists of several modules that can be integrated in part or in whole to the device client application. It includes three core functions:

1. Application Services

    - These are application interfaces that provide support to applications for specific services such as HTTP, JSON, and Verizon specific services (e.g. Wireless User Subscription Management and ThingSpace Device Messaging APIs support).

2. Connectivity Manager

    - These are the core functionalities that implement automated wireless connectivity selection (i.e. from BLE, Wi-Fi or 4G LTE cellular), power management and wireless device control.

3. Hardware Device Abstraction

    - This is a thin layer of abstraction that hides the hardware from the higher layer software that interacts with it. It allows the individual hardware components on the Wearable device to be easily replaceable with minimum impact (if any impact at all) on the DEVICE CONNECTIVITY PLATFORM software that runs over it.

Items in green below are defined APIs that provide building blocks to implement and utilize key system functionality. Including:

- Network Session API: Used to open and maintain data connections using data streams
- HTTP API: Used to open and maintain HTTP based data connections using data streams
- Streaming API: Used to open and maintain data stream definitions. These streams source and sink information using the Transport layer.
- Feature API: Used to select features and permissions within the V-Wick library
- OS Abstraction API: Used to configure and maintain OS features within the API
- Utilities API: Used to perform data stream parsing and storage activities
- BLE Data Access API: Used to provide the means of sending and receiving GATT Attribute data and receiving BLE Event notifications.

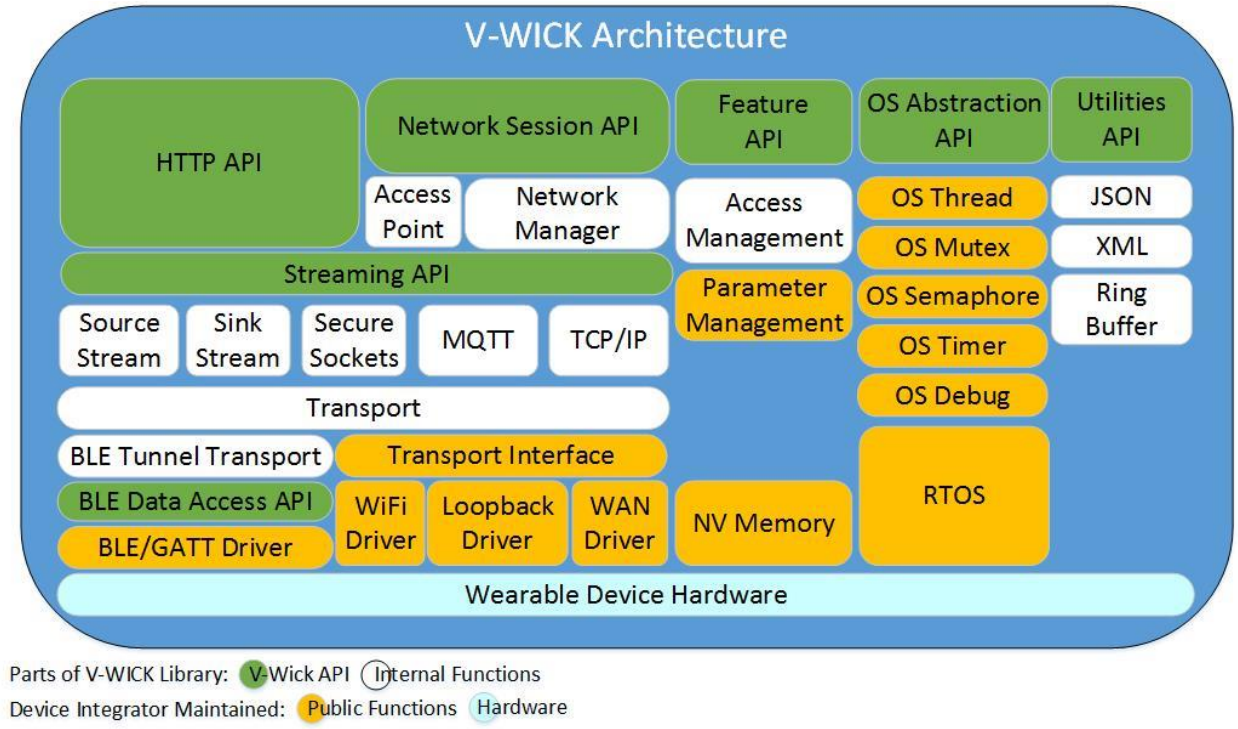


Figure 3: VDCP Architecture

Items in orange are present in the SDK distribution as a reference design. These components will need to be updated to suit your device hardware implementation.

- Transport Interface: Provides the needed connections between the Network Manager and the low level transport drivers
- WiFi Driver: Driver that supports the device's WiFi Hardware
- WAN Driver: Driver that supports the device's WAN Modem Hardware
- BLE/GATT Driver: Driver that supports the device's BLE Stack and Hardware
- Secure Sockets, TCP/IP: Provides messaging protocol services to the Streaming API. This software is layered above the transport layer code.
- NV Memory: A Non-Volatile Memory driver that provides access to VDCP data objects that must be retained across reboots
- Parameter Management: Stores configuration settings and enforces read/write permissions. This utility is used by the Access Management code
- RTOS: Selected by the Device Integrator, The RTOS has needed connections to the OS Abstraction Layer as defined in the OS Thread, OS Mutex, OS Semaphore, OS Timer and OS Debug components that need to be updated by the integrator

Items in white are shown to describe the functionality available within the library.

- Network Manager: Used to define and maintain data connection objects. It works with the streaming API to define a combination of messaging protocols, sockets and transport layer paths
- Access Point: Used to configure and provide access to access point functionality within the WiFi driver. This includes setting the Access Point name and security settings
- Transport: Provides a connection between the Steaming API and the Device Integrator's Transport Interface

- Loopback Driver: Provides simple testing functionality that forwards data from the Source Stream to the Sink Stream
- Access Management: Enforces access to privileged functionality within the VDCP library. An up-to-date certificate is required for the library to allow access
- JSON, XML, Ring Buffer: Utilities that are provided to perform data parsing and storage activities

3.1 Built-in Advantages in Security & Connectivity

The VDCP software package has a Hardware Adaptation Layer (HAL) which allows hardware to integrate with the VDCP (i.e. chip/module manufacturers can pull in the VDCP libraries to build a VDCP - enabled device client) and have this device be managed in ThingSpace.

1. VDCP enabled devices are manageable via TS-Manage (a ThingSpace Management UI Portal) thru an Adaptor created by ThingSpace.

2. VDCP enabled devices are connected to Verizon's backend Billing system upon creation of a Customer Account in the system.

3. VDCP enabled devices can take advantage of the single pane of glass principle observed in ThingSpace, i.e. partake in the holistic family of services including the device life cycle management, the billing and charging system, the security provisions, the performance and monitoring system, the Location service, the logging, reporting and Data Analytics services and a whole host of other utilitarian "services".

4. Data in/out of VDCP enabled devices are protected when at-rest or in-transit.

4 Details Behind The Thingspace Device Messaging APIs

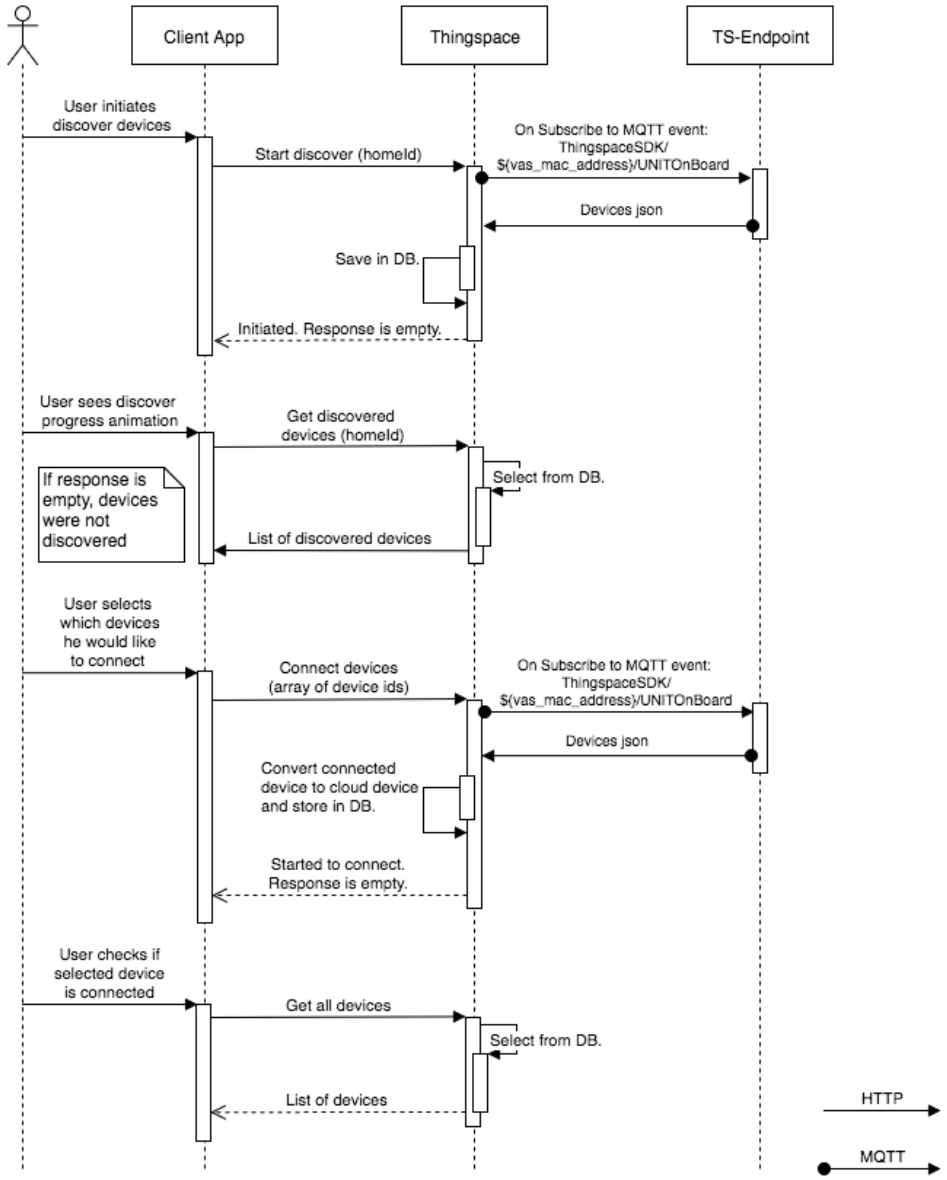
The ThingSpace platform supports secure, consistent APIs/services for configuring, managing, and interacting with a variety of "things" in the world. These include devices supporting smart cities, vehicles, sensors, the connected home, etc. The set of APIs/services provided by the ThingSpace platform comprises the common set needed by a variety of projects. APIs/services that are device-specific, such as those supporting Dakota, are considered project-specific additions that run on the ThingSpace platform, but are to be provided by the project itself.

Thingspace integration can provide the following services

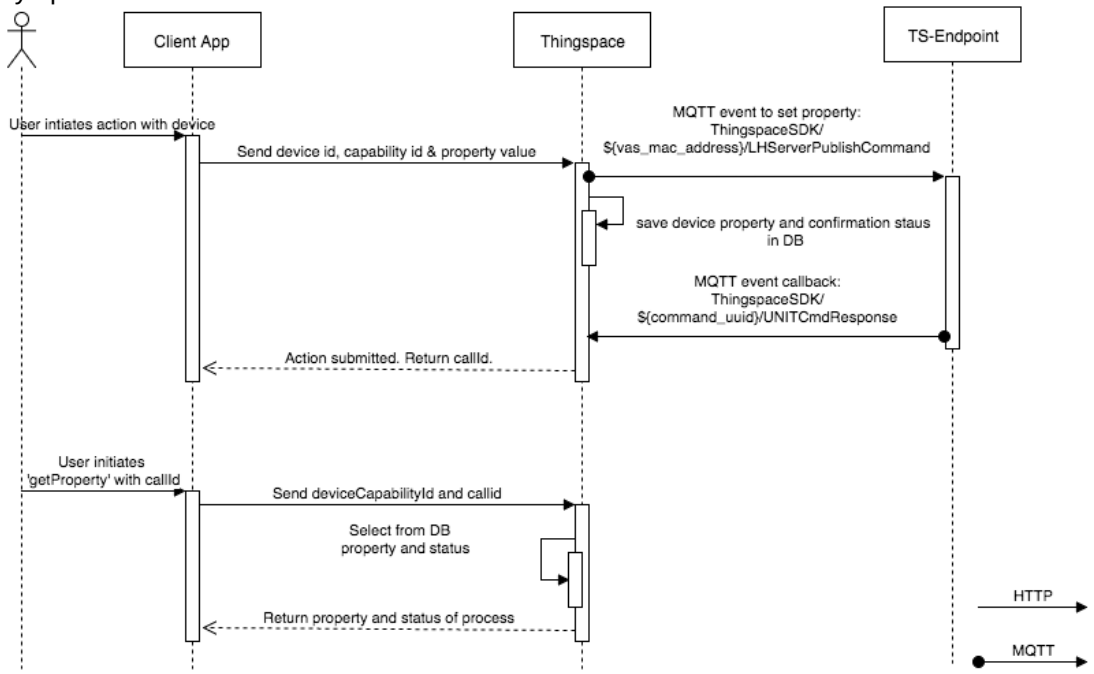
- Provisioning
- Access and identity management
- Monitoring
- Logging
- Messaging
- Notification

The ThingSpace APIs supported in the VDCP focus around messaging and notification. These communication stacks facilitate the exchange of data between a ThingSpace-Ready device and the ThingSpace platform using the MQTT messaging protocol. The MQTT client module is responsible for MQTT communication with Thingspace server. This module will initially publish all the topics which will be subscribed by the Thingspace server. This module will also subscribe the required topics from Thingspace server.

Device Discovery starts with the Thingspace Cloud receiving HTTP request. The Thingspace Cloud performs discover new devices by sending MQTT message to TS-Endpoint. When Thingspace cloud receives a Connect HTTP call it connects the specified device from HTTP request to its network (the exact process is device- and protocol-dependent).



Any device interaction must first be enrolled by the host device into Thingspace. The parameters for updating device property include device ID, capability ID and property value via HTTP request to Thingspace Cloud. The Cloud will trigger MQTT command to TS-Endpoint UNIT for property update. After change property action was submitted Thingspace Cloud returns response to Client Application with property update status.



5 References

Hayward, J., Chansin, G., and Zervos, H. (July, 2016). Wearable Technology 2016 – 2026 Markets, players and 10-year forecasts. IDTechEx. Retrieved from http://www.idtechex.com/research/reports/wearable-technology-2016-2026-000483.asp.

TS-Develop website. Connectivity API / Device Messaging API. Retrieved from https://thingspace.verizon.com/developer/apis#/.

Robinson, Q. & Viswanathan, M. (2016). *ThingSpace* SDK Software Design and Specification Version 1.1.

IoT Security Credentialing – UDM integration: Requirements Version 0.8