Internet Of Things Software Update Challenges: Ownership, Software Security & Services

Samita Chakrabarti
samita.chakrabarti@ericsson.com
Ericsson

Abstract: *This paper focuses on today's practice on 'over the air' software updates and possible challenges of remote software updates for Internet Of things—i,e. potential considerations of the real-life issues with device ownership, Security challenges and Service providers' business models in designing the network architecture and protocol choices. In conclusion, it notes down a few high level requirements for IoT software updates.*

Introduction:

Internet Of Things (IoT) comes with the notion of ubiquitous presence of devices, a web of things connecting every aspect of our lives potentially everywhere - home, travel, agriculture, enterprises, healthcare and so on. The IoT devices can be as capable as a smartphone and Internet enabled video-camera or as impoverished as tiny sensors inside the automotives, home healthcare devices or on the wearables. They may be connected with the Internet or they may be connected only within the Enterprise environment.

It has been proven by business experience that often over the air (OTA) software updates are far more cost effective and timely than manual software updates. Rapid roll-out of fixes and increased reliability and function of the IOT devices are part of the benefits of remote software updates. Today, we enjoy OTA software update on our smartphones in the Mobile technologies and as well as in the IT fixed services such as laptops and tablets. Most of them work like a charm – most of them are well tested and we trust them to do the right things. But the question is how to ensure correctness of the software updates on the remote group of sensors or in a critical IoT device which has zero tolerance for errors introduced by the new updates. Of course authentication and encryption of update software are mandatory in order to avoid any unwanted attacks on the critical operations controlled by sensor data measurements.

As examples, let's consider Open Mobile Alliance Device management [1] technology that is used today as a standard for over the air software updates for many deployments such as smart phones and automotive like Tesla[2]. The WIRED article[3] describes that Tesla owners received a recall notice from National Highway Traffic Safety Administration alerting them about a charger plug for a potential fire hazard. Tesla reportedly completed the fix for its 29,222 vehicle owners via the remote software update. Tesla owners also receive other software updates, such as setting configurations for suspension settings allowing the cars to deal with collision situations better. This is a giant step towards deployment of

Internet Of things software update with success. In future this kind of technique can be useful for self-driving cars downloading necessary software on the fly or updating software on the home IoT devices at a time from the same manufacturer. One thing to note here, that the software updates only take place when the car is at rest on the driveway or at the owner's garage.

Business Relationships and Services:

Many of the IoT deployment today run on Industry verticals. Most often a startup company delivers its own set of IOT devices of a particular RF technology and a technology specific gateway or an application gateway software to talk to their cloud application. Often the companies may work with a reputed Service Provider partner to offer their services through that partner. In other cases, the Service Provider or Operator owns the Operational and device management software and they work with the device vendors and their gateways to use the gateway as an entry point of software update download. The gateway determines when and how to further install the software updates on all or selected IoT devices it manages – thus in this case, the Service Provider has a trust relationship with the gateway through authentication mechanism and delegates the software update responsibility to the gateway. In another model the Service provider places distributed software update server and the gateway device takes the responsibility to initiate the software update to a group of devices. For large IoT devices such as cars, video camera or smart meters may directly be reached by the Service providers for the updates. However, we need to also think about regulatory policies for software updates and any issues that might cause major business or environmental or health issues due to incorrect software updates or bugs in the software. Today most software updates are interactive and the users are aware of them, but in IoT automatic software updates , the authentication, identity check on both sides , verification of existing software update requirements, time-of-operations are extremely critical.

Security and Privacy in Software Updates:

It is extremely important to have a verification procedure in the IoT Software update so that the service provider for software update can make sure that the particular batch of IoT devices are the right target for the updated software. Often the IoT devices are not seen by the Cloud side of the services as the information is maintained at the local gateway level . The gateway somehow needs to be trusted and authenticated but it should share the device group software (existing version, key, certificate etc.) to the Service Provider – so that they can previously verify for correct operations.
The security should be handled in multiple ways – 1) Device security(identity, encrypted storage of identity) 2) Security on the wire between the device and the Service provider or the gateway 3) Network security between the devices gateway to the Service provider 4) Security of the storage of software application 5) End to end Security .

Thus often vendors like Infineon and Gemalto are working on inserting identity and crypto function storing the identity on the HW of the IoT device. This information may be used as seed for private key of the device and can be used for secure communication including application downloads and firmware updates.

Open Mobile Alliance Software Update:

OMA-DM[1] software is usually provided by the OEM of the IOT hardware and the software is installed in the Cloud management server of the Service Provider. Meta-data might be added. The software is validated , then compressed and encrypted and distributed for updates. The secure protocols such as HTTPs and SSL mechanisms are used. In case of vehicle, it unpacks, validates the authenticity of the software and then distributes inside the vehicle among various delivery points. OMA –DM software updates are devided in "Software Over The Air" and "Firmware over the air" categories and they are distributed wirelessly utilizing several access technologies (Wi-Fi, Bluetooth, Cellular, Satellite). The delivery method could be direct from the Cloud server or via a proxy(Wifi-gw, Smart device etc.). Other accessory tools for software update validation may be part of the software update package.
OMA –DM[1] also defines the management objects to support functions in diagnostic and monitoring, connectivity, virtualization, policies. OMA-DM [1] defines Lightweight M2M[6] protocols and API based on COAP and DTLS over UDP or SMS as underlying protocols. They provide interfaces on Bootstrapping, Registration, Resource access and Reports.

Timing of IOT software Update:

The devices are mostly constrained or they are engaged in measurement which is used by the control loop continuously. Thus the intelligence in the IOT software update is crucial to find information on device sleep and wake schedule and determine when to update the software. Another possibility is to roll-out the software download on the IOT-devices when they are not active in operation and then activate the software when they are not temporarily engaged in control loop. This should depend on the local policy and regulation of the deployment. The dynamic or automatic software update might benefit with a protocol which can first test the software with a test IoT node and after successful trial on the testnode in that environment it can apply that software update in other devices . It might be better to update software in intervals not to cause disruption, else update the software in a specified down time. In order to maintain the time, it is needless to say that the IoT devices must be time-synchronized by a coordinator operated as a IoT-gateway or by the Service Provider Service network.

Summary Of Thoughts

Based on the above discussion on challenges of remote software updates on a group of resource constrained IoT devices using IETF and other standards technologies, here are some suggestions for considerations on designing the IoT software Update architecture:

- Using existing technologies on the wire protocols are desirable
- Consideration of deployment and regulatory and Service provider requirements are necessary
- End-to-end dynamic and automatic software updates are possible for certain class of IoT devices but the software update protocol should consider the case when a set of resource constrained updates are gated by a IOT GW device and/or a management server
- Device update timing and sequence of Updates could be part of the dynamic software update protocol
- Time synchronization of IoT-nodes and the software update provider might be necessary

- Having an automatic test of the to-be-downloaded software effectiveness might be investigated for critical IoT constrained nodes operations
- Multi-layer Security is needed in the IOT deployment scenario – Hardware level device security and encryption, Network access security and authentication of the IOT device and their gateways and security authentication between the Service provider network and IoT device/gateways are required
- Compression and Encryption of IOT software update are required end-to-end
- IoT Technology specific and Application specific data-models/profiles are needed to support different set of IoT devices networks (Bluetooth, DECT-ULE, NFC, 802.11ah, 802.11 etc.)
- Roll back consideration of a defective software update must be part of the software update design consideration

[1] OMA DM, Salvadore Scarpina, "OMA Device Management"

[2] Tesla, Tesla Motors- https://www.teslamotors.com/

[3] WIRED Article, Alex Brisbourne, "Tesla's Over-The-Air Fix: Best example yet on Internet Of Things?"

[4] OMA Device management Guidelines

[5] Secure Software and Firmware Update, Infineon, http://www.infineon.com/cms/en/applications/chip-card-security/internet-of-things-security/secure-software-and-firmware-update/

[6] LWM2M: OMA Lightweight M2M Specification