SONY

Issues with bi-directional communication for CoAP and other IoT-related protocols

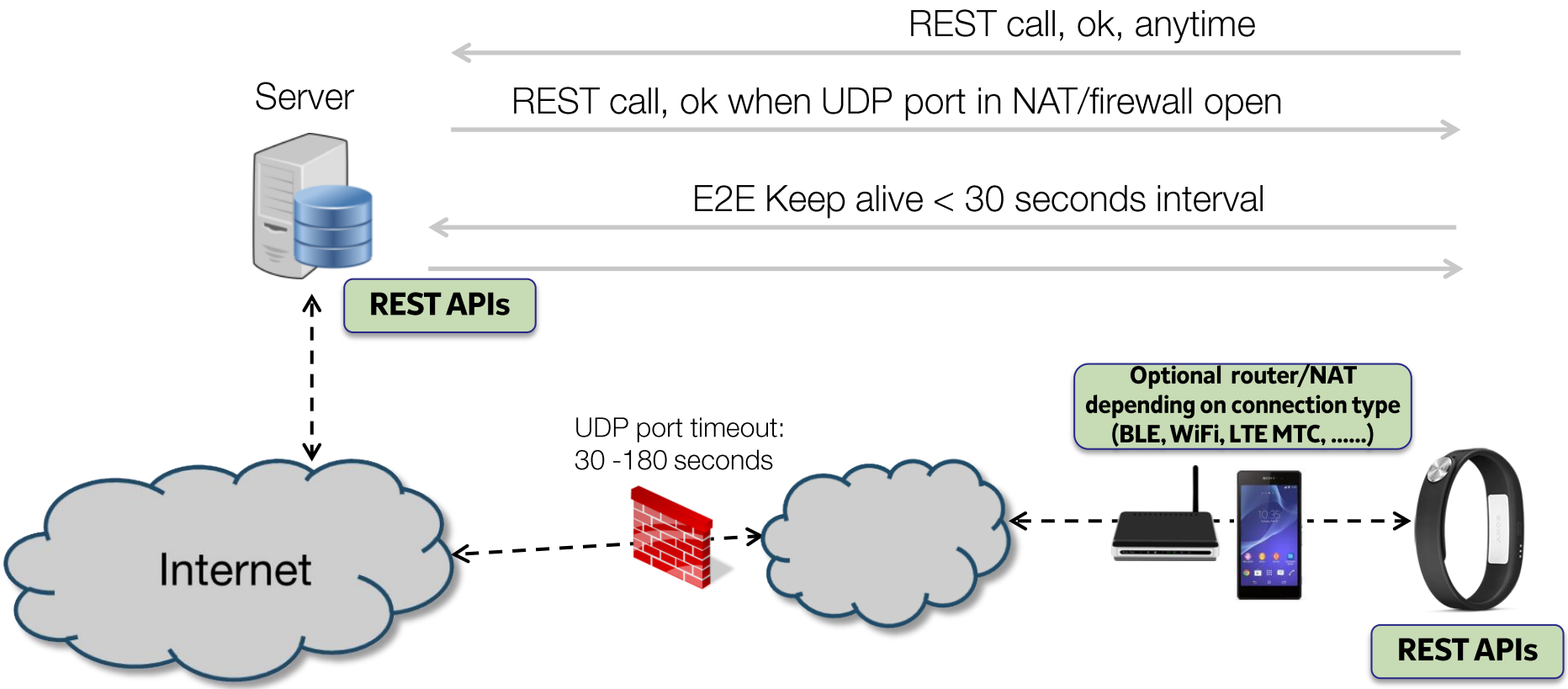
For discussion at the W3C WoT IG TF-AP meeting 2015-12-02

Claes Nilsson, Sony Mobile

Problem statement

- Many IoT use cases require both peers to act as both client and server.
  - See for example slide 16 and further in WoT standardization and example UC “Remote Health Monitoring”
- Use of IP and transport protocols such as HTTP and CoAP enables REST APIs to be exposed to the other peer.
- However, NATs/firewalls cause problems.
  - The IoT device can anytime send a request to a cloud server.
  - The ability for the cloud server to send a request to the IoT device is limited.
    - Request from the cloud server, acting as a client, must be sent over the same IP-address/portnr quadruple (source/dest address/port) that was used for a previous request from the IoT device
    - Firewall is open for requests from the clouid server a limited time after a request from the IoT device. For example see Nokia Research Center on Impact of Keep-Alive Messaging on Power Consumption

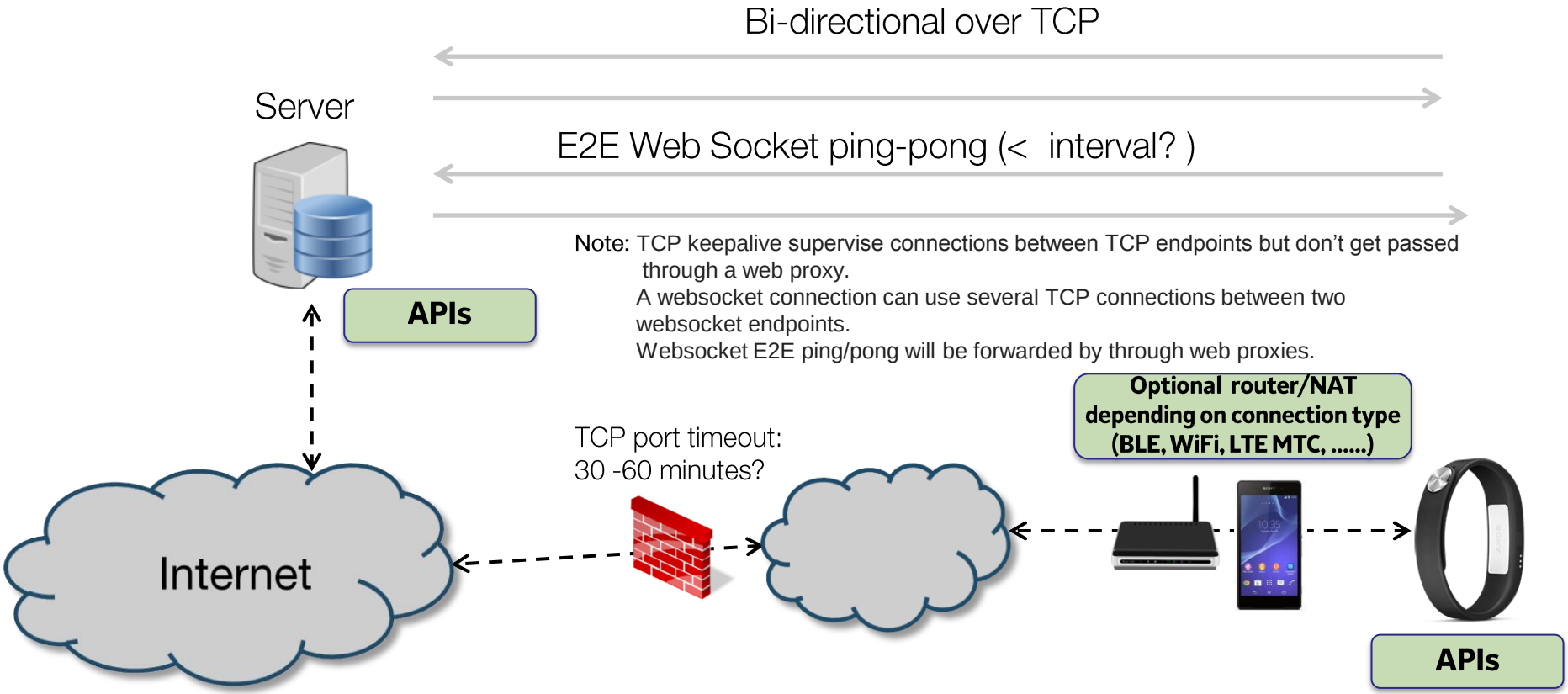
CoAP – Example architecture



Using Web Sockets

- For bi-direction communication Web Sockets is a better alternative
- However, Web Sockets may be inappropriate for constrained devices.

Web Sockets – Example architecture



Bluetooth sleep-mode and connection messages

- To save battery BLE goes into "sleep-mode" when the device is inactive but to keep the connection open BLE wakes up and sends "connection messages“
  - Interval of typically 1-2 seconds.
  - During this "wake-up" the BLE-device can send data to and receive data from the other peer. Accordingly, a BLE device can send and receive messages with a max delay of the BLE connection message interval.
- So, if we run the Bluetooth Smart, i.e. the IPv6 Bluetooth profile and CoAP/UDP, and the device is behind a NAT/Firewall, the device can respond to REST calls under the condition that the UDP port is open, with a delay of 1-2 seconds.
- However, the main issue is probably the battery drain caused by the keep-alive messaging needed to keep the NAT/firewall UDP port open.

LTE MTC modes

- LTE MTC provides low power, low data rate transmission with direct IPv6 connectivity with devices through the mobile network
- Modes:
  - Power Save mode: Wakes up periodically and could not be reached by any means, not even SMS, while the device is sleeping. The wake up interval is dynamically configurable in the device (seconds to hours).
  - Connected/idle mode: Wakes up every 1,28 seconds (DRX interval) and could communicate.
  - Under standardization: Possibility to, at the network side, configure the DRX interval from seconds to max 30 minutes.

Need to keep firewalls open depends on use case

- Options:
  - IoT device wakes up periodically (but not as often as needed to keep firewall open), sends request to cloud server and can receive subsequent queued requests.
  - IoT device sends periodical wake up messages to keep firewalls open meaning that it can receive requests from cloud server ”anytime”
    - Also consider medical use cases when cloud server needs to supervise connection with medical device.

How does existing solutions work?

- OMA LWM2M
  - Michael Koster: Device (LWM2M client) wakes up occasionally and sends an update to a Resource Directory server (LWM2M server) through the firewall. This provides enough time for the application to go back through the firewall tunnel and interact with the CoAP device to perform any pending transactions.

- Amazon AWS IoT
  - How do devices receive publications?