SWEDISH ICT SICS

PART OF RI.SE

LUND UNIVERSITY

Security Challenges and Solutions in IoT systems

Christian Gehrmann, April 5, 2016

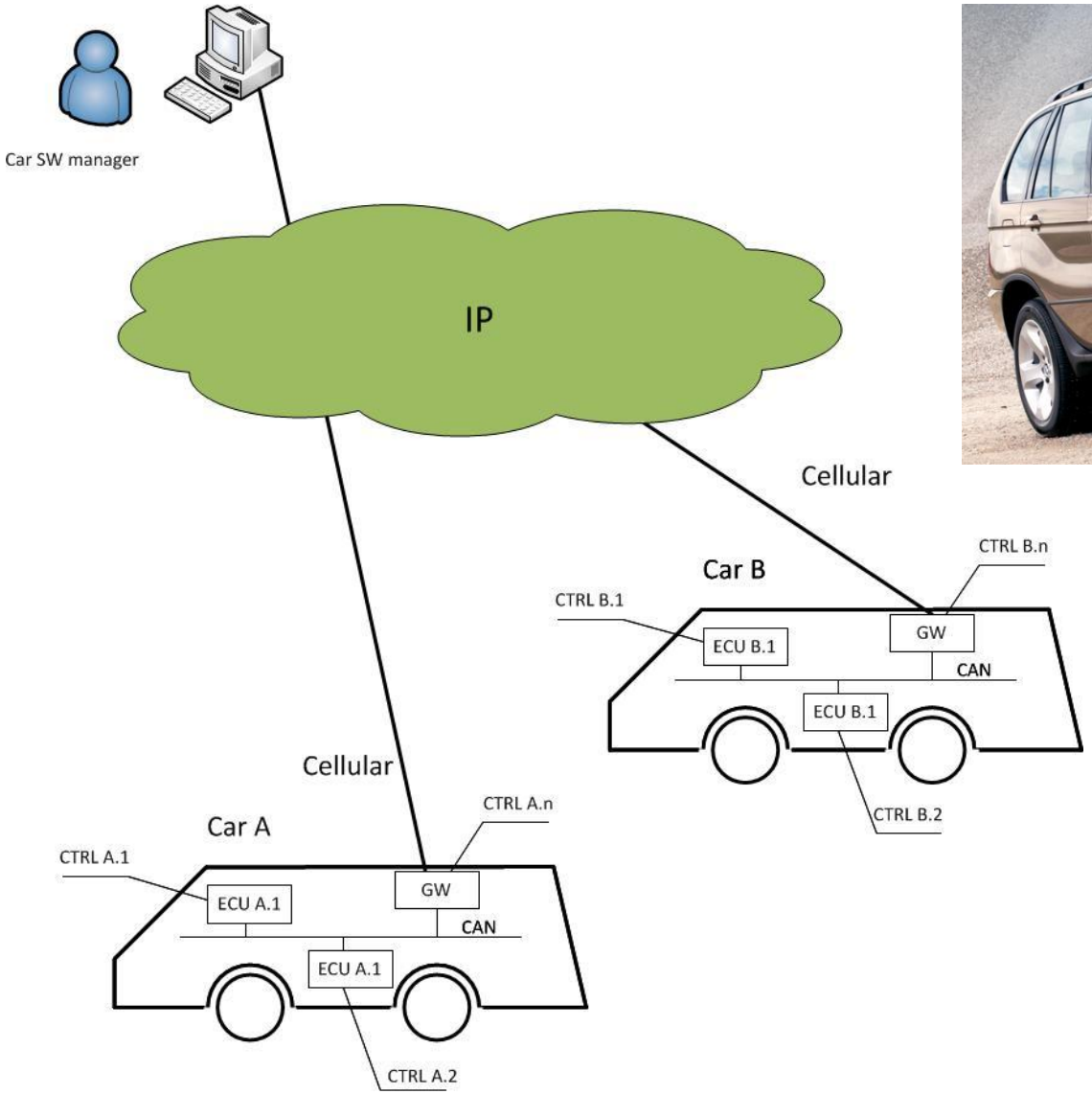
Contents

- IoT systems – different scenarios
- Major threats/attacks
- Other major security challenges
- Future solutions - potential directions
  - DDoS prevention
  - Stealth prevention
  - Secure SW upgrade
  - Authorization solutions
  - Key management
- Conclusions

IoT System?

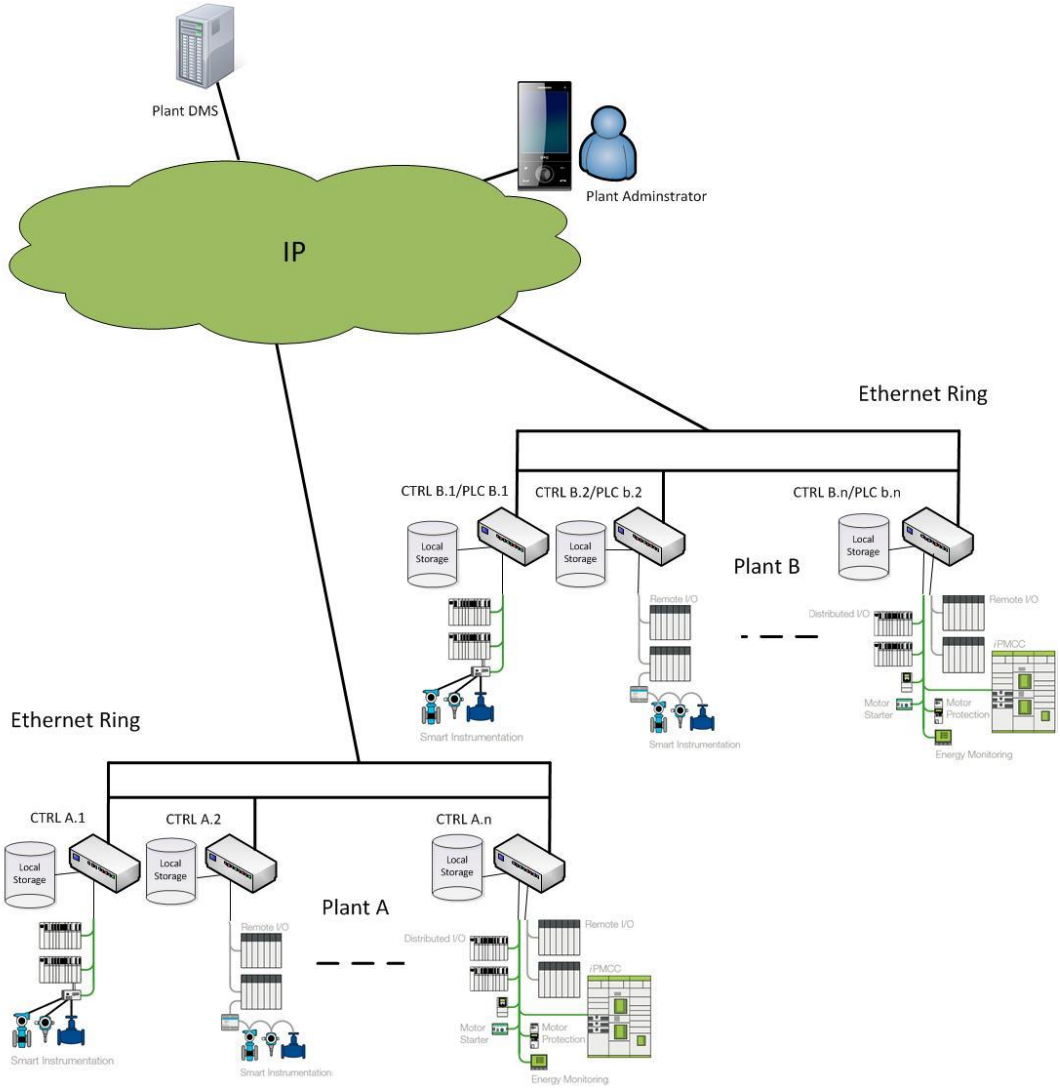
- System of connected devices, vehicles, buildings etc.
- Gartner: ”In the post mobile world the focus shifts to the mobile user who is surrounded by a mesh of devices extending well beyond traditional mobile devices“
- Complex?
  - Some system are very complex, others rather simple….

Automotive

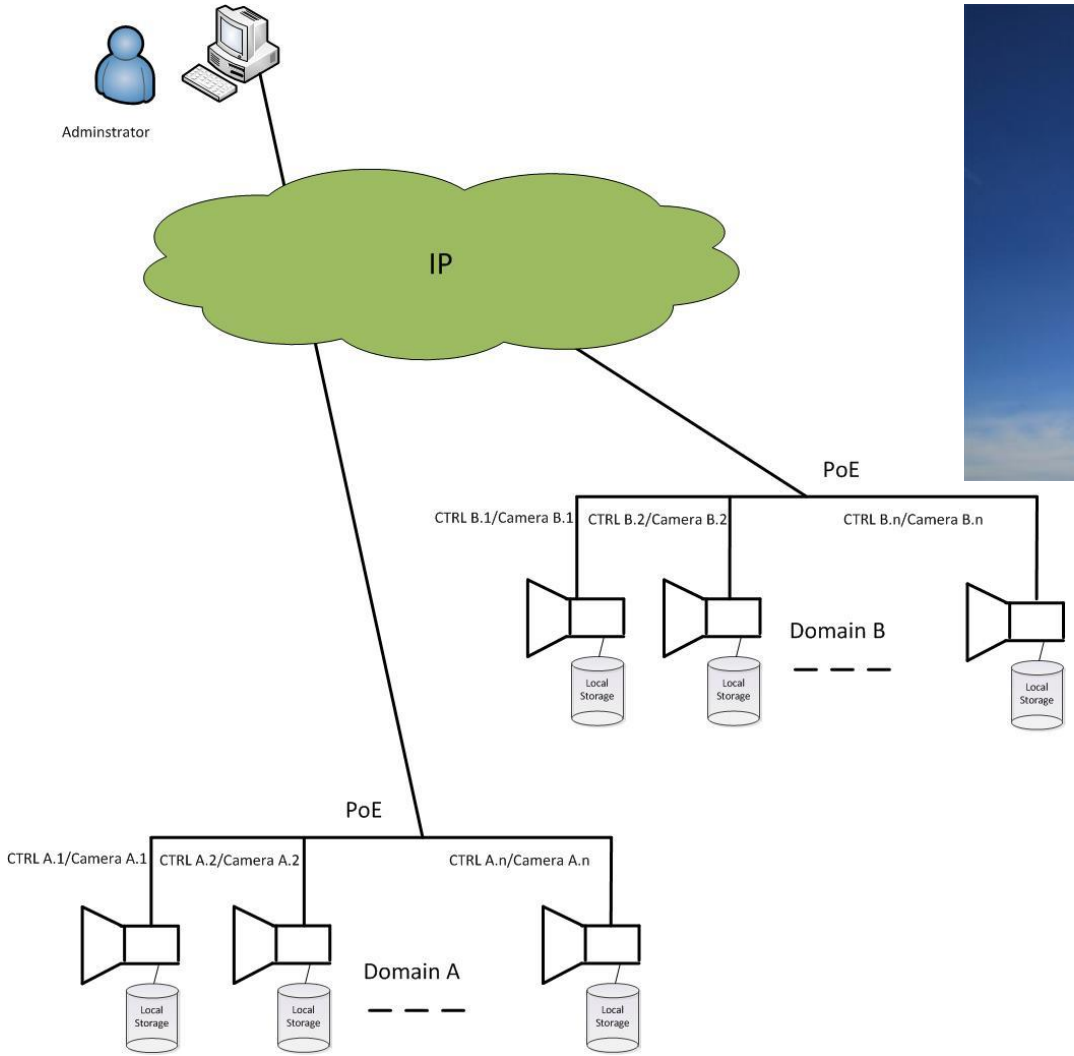

Car SW manager
IP
Cellular
Cellular
Car B
CTRL B.n
CTRL B.1
ECU B.1
GW
CAN
ECU B.1
CTRL B.2
Car A
CTRL A.n
CTRL A.1
ECU A.1
GW
CAN
ECU A.1
CTRL A.2


M RN 4164

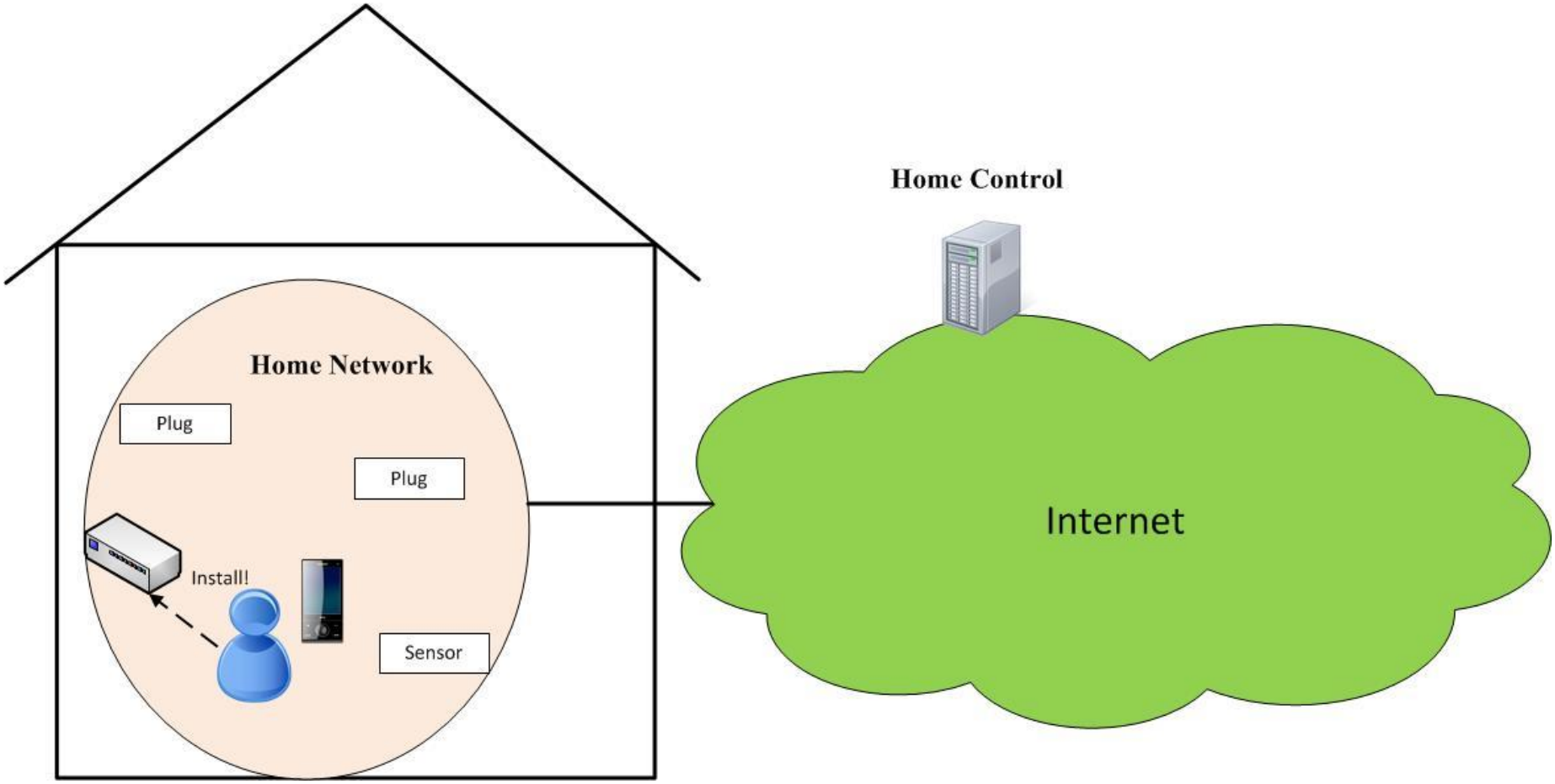
Industry control system



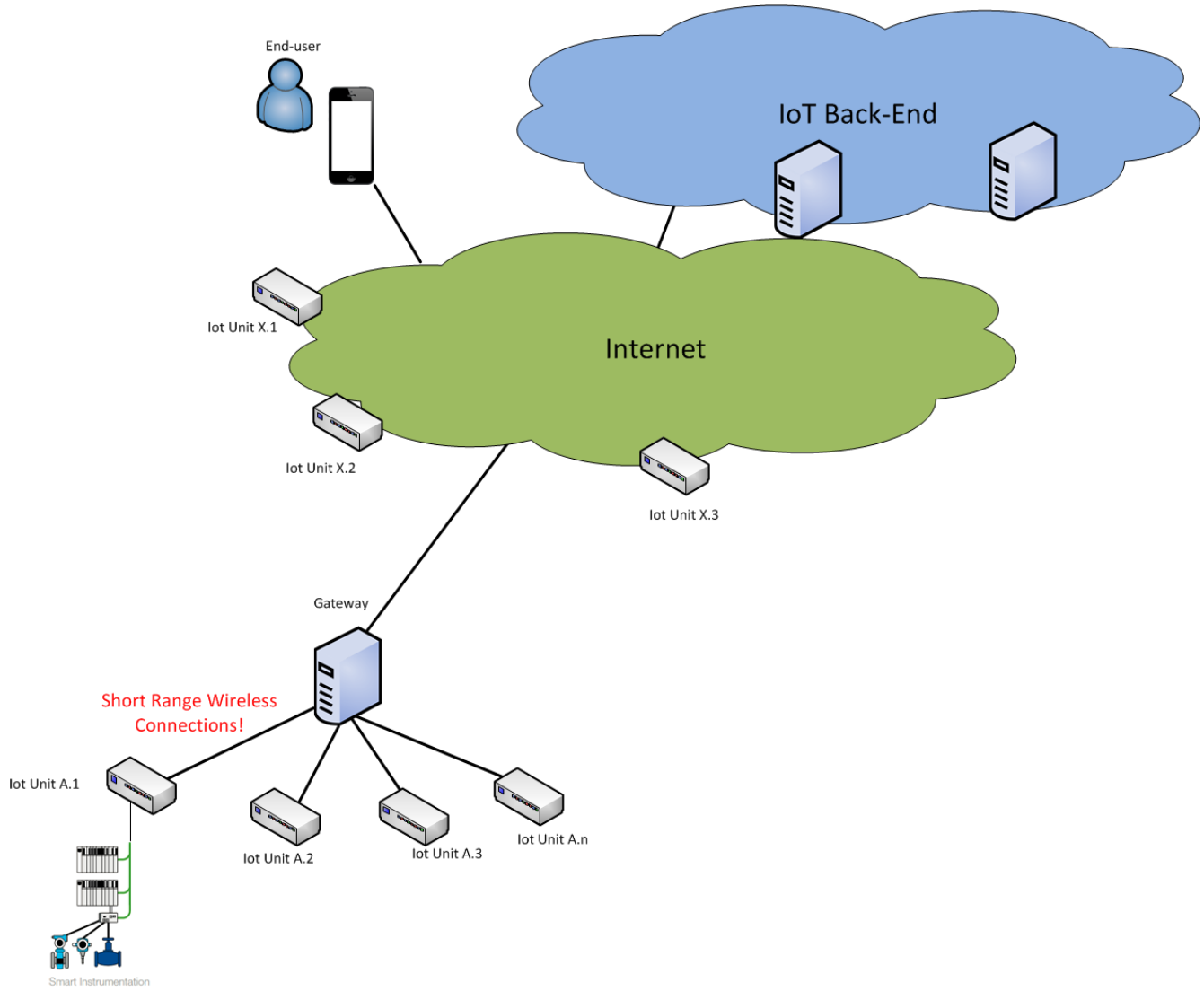
Security System



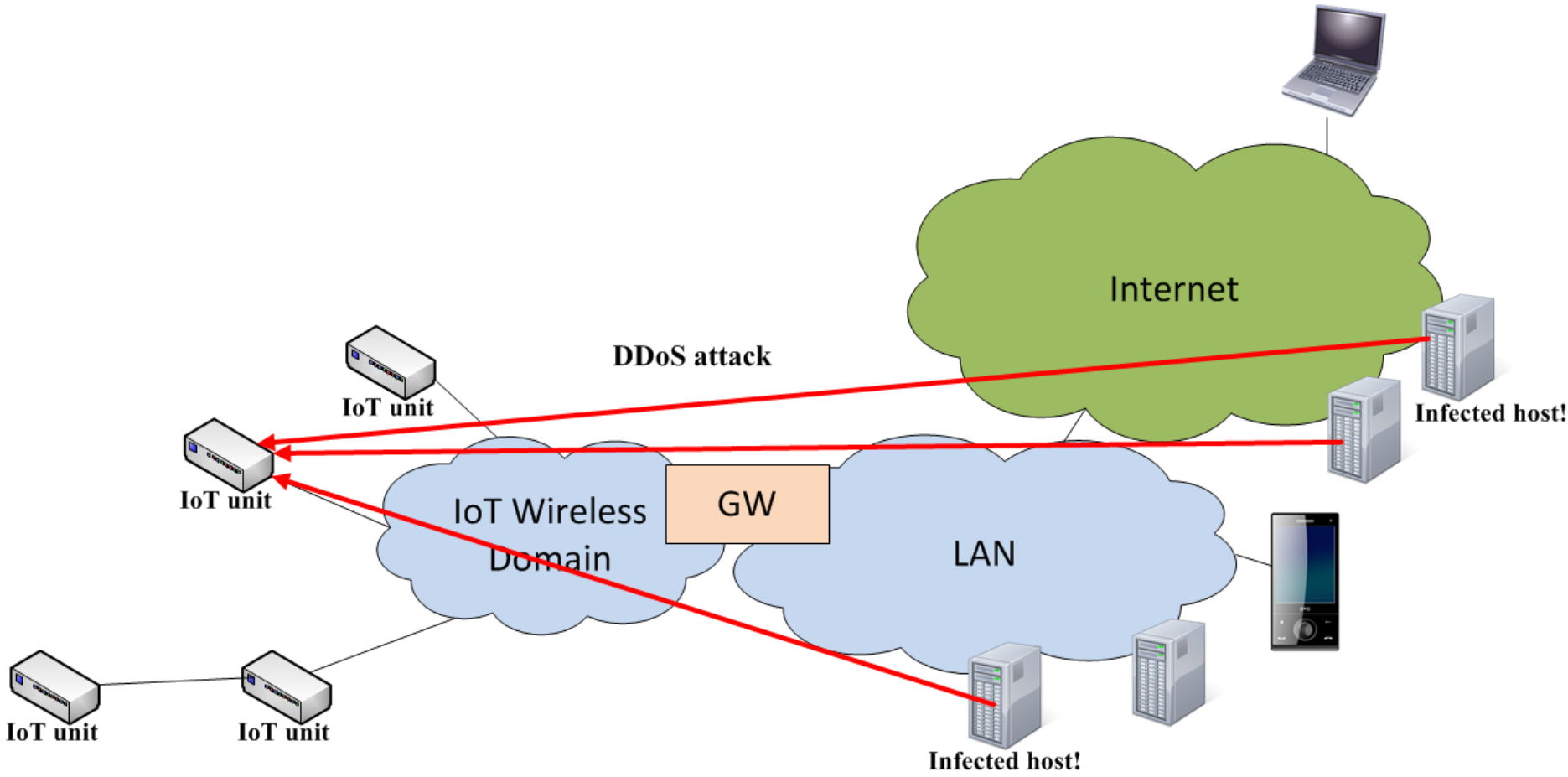
Home Control



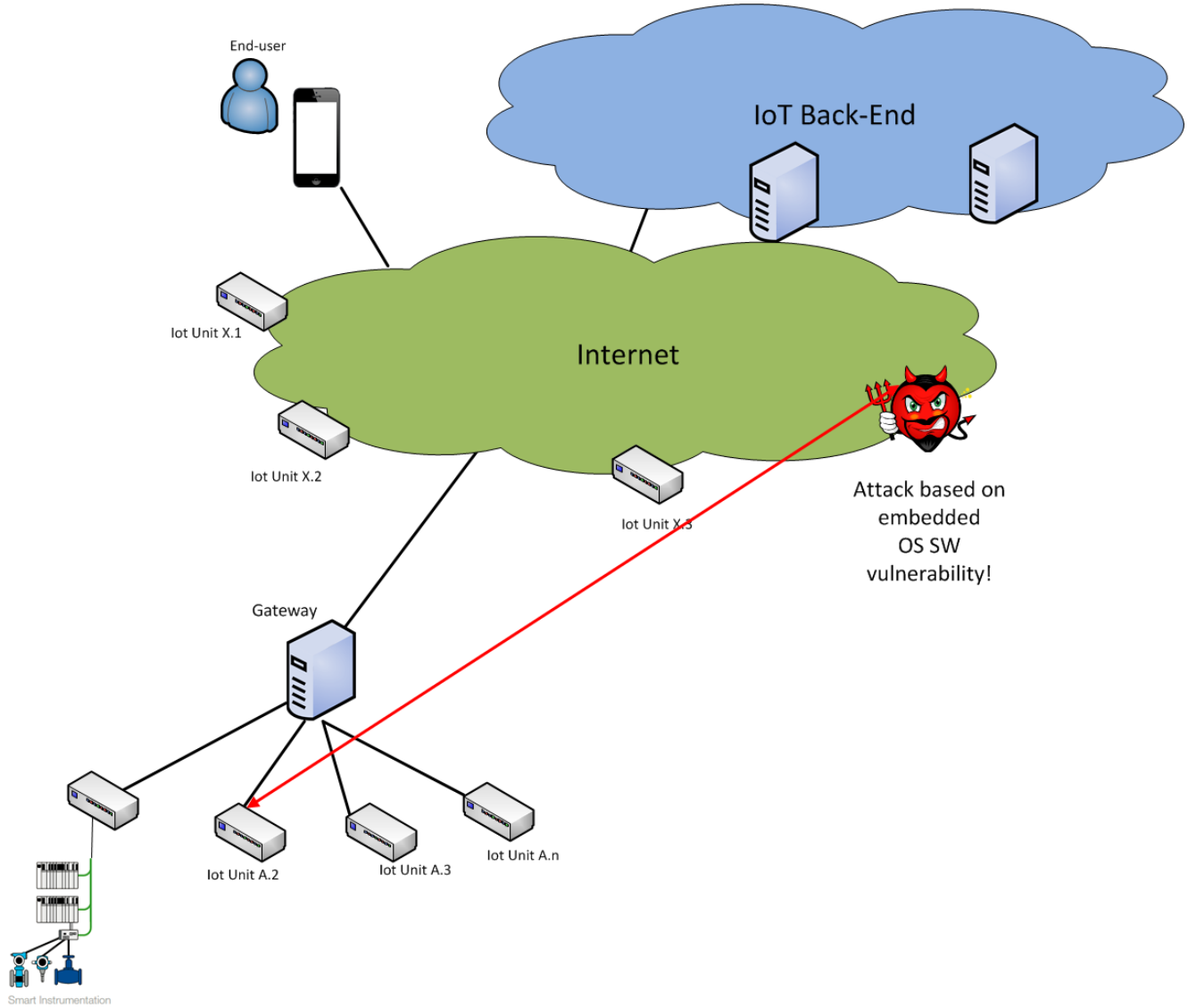
IoT infrastructure – typical system



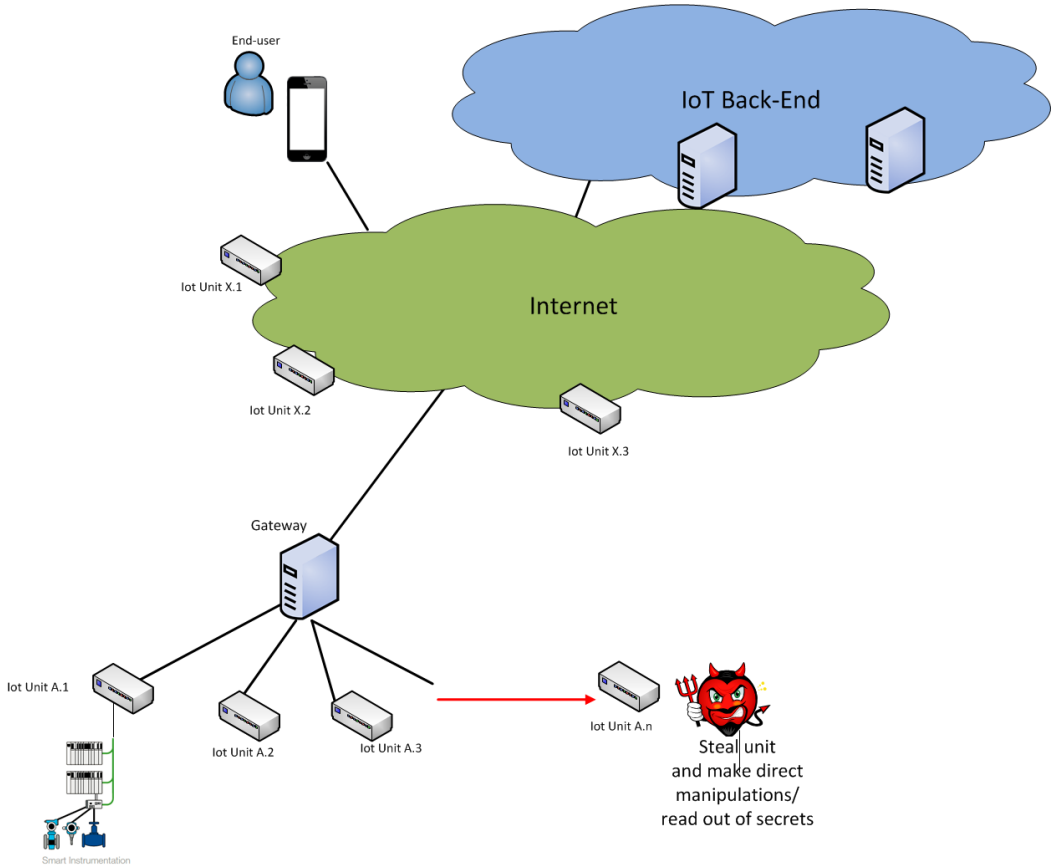
DDoS attack


Internet
DDoS attack
IoT unit
IoT unit
IoT Wireless Domain
GW
LAN
Infected host!
IoT unit
IoT unit
Infected host!

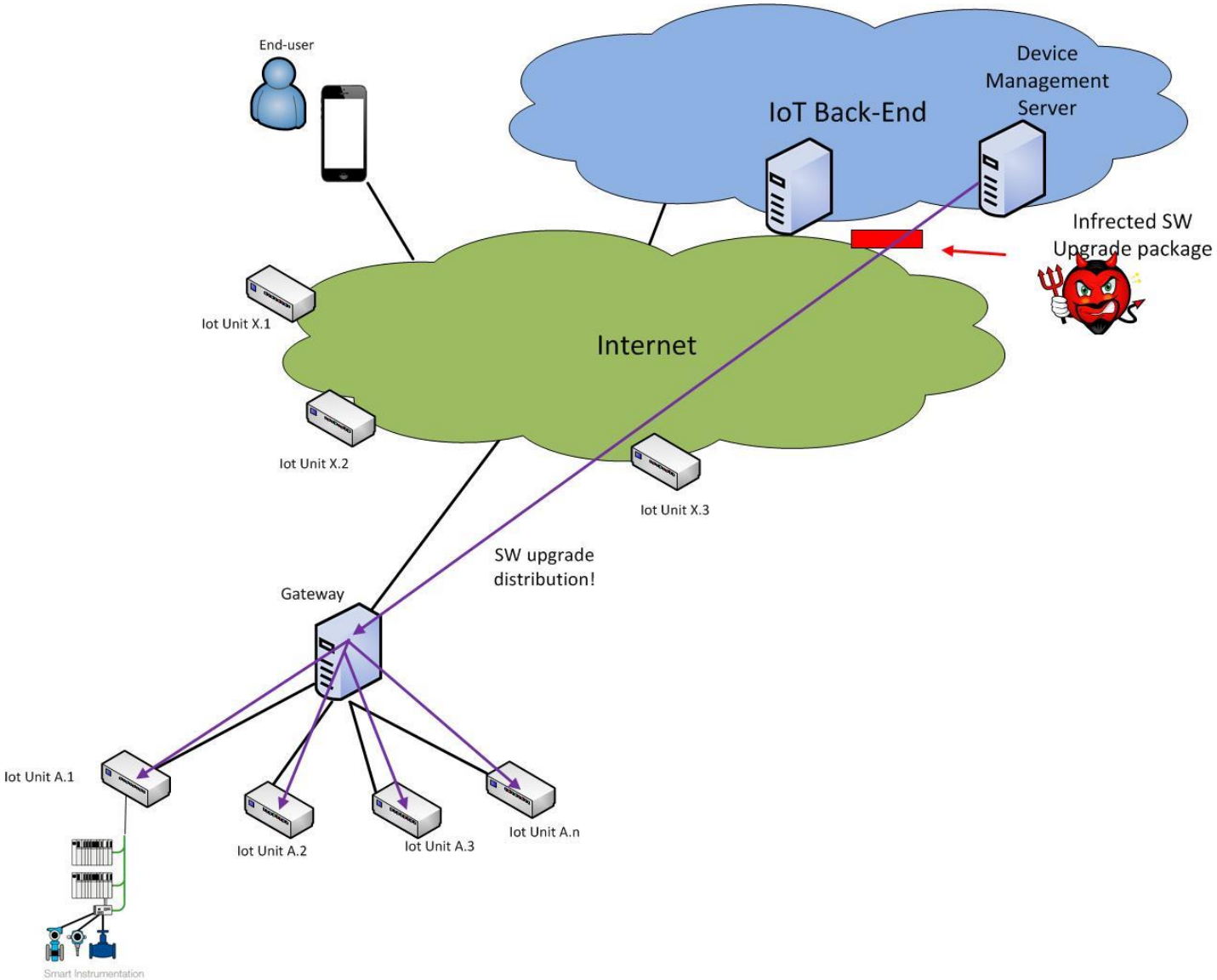
Network based attacks



Direct Physical attacks



SW upgrade attacks



Major attacks in IoT systems, summary

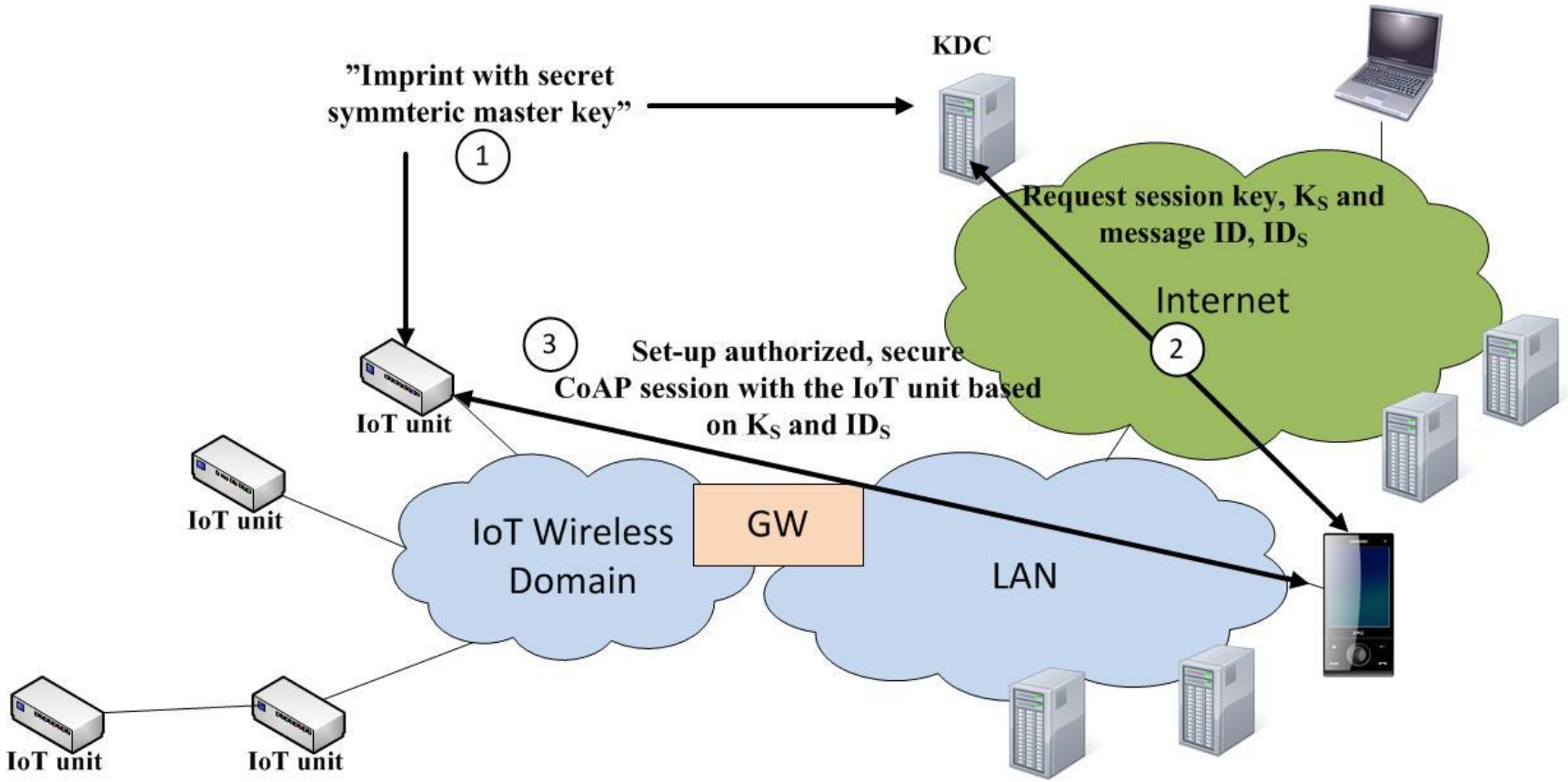
- Dos and DDoS on battery driven resource constraint IoT units (typically communicating using low power wireless link technologies)
- Networked based attacks utilizing weaknesses in embedded operating systems and/or protocol implementations
- Direct physical attacks against IoT units (probing, stealing devices and their memories etc.)
- SW vulnerabilities in upgrade packages

Other major security issues

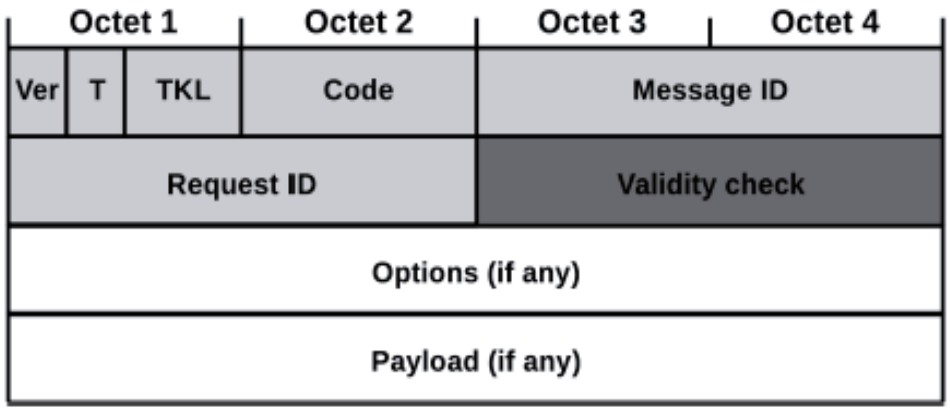
- Device credential provisioning, update and ownership ”roll-over”
- Device recovery at critical SW failure
- Dynamic authorization and access control

Security solution examples

Detecting Battery Drain attack with short Message Authentication Code (I)



Detecting Battery Drain attack with short Message Authentication Code (II)



Include a short validity check, i.e. MAC in the CoAP header for instance

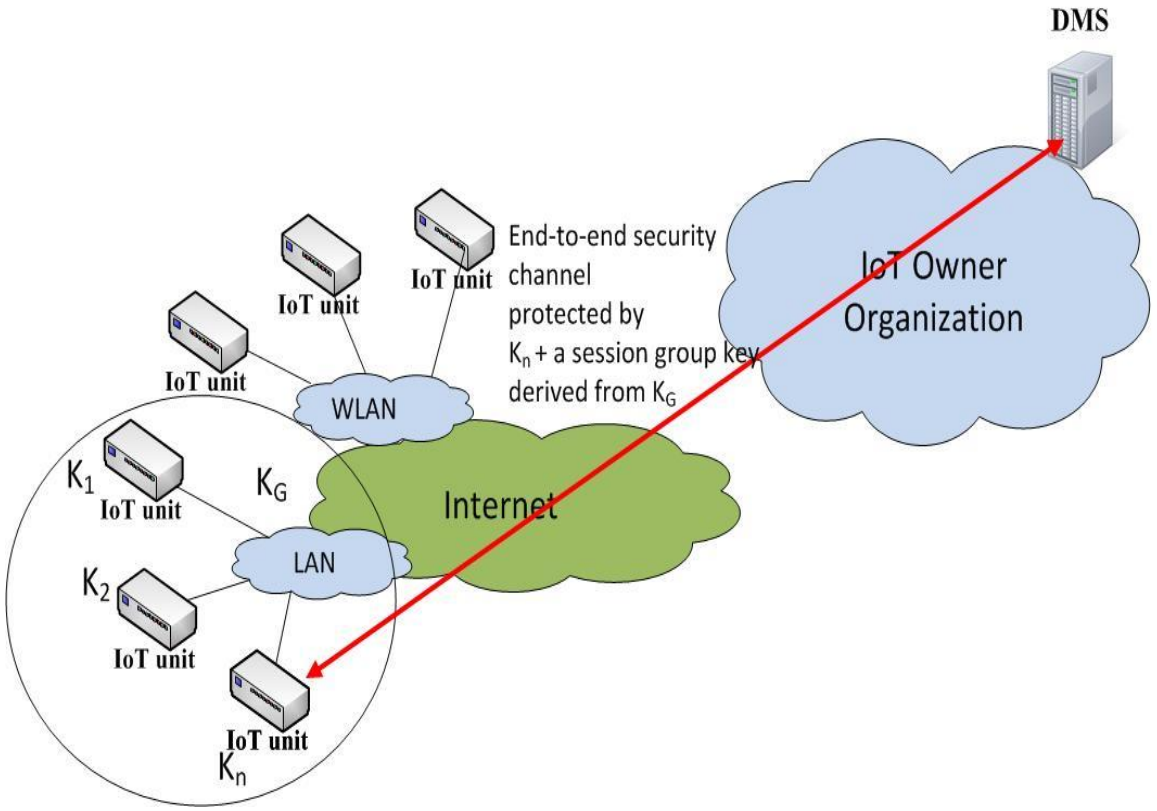
Detecting Battery Drain attack with short Message Authentication Code (III)

- Procedure at Iot Device side:
  - Use a pre-share key and the ID field in the CoAP header to find a ”session key” and calculate a short MAC which is compared with a MAC field in the header
  - If the values co-inside accept the message as valid
  - If a *large number* of invalid packages arrives within a relative short time period, take action like
    - Shut down network interface
    - Power down for a period
    - Etc.
- References:
  - C. Gehrmann, M. Tiloca and R. Höglund, “SMACK: Short Message Authentication CheckK Against Battery Exhaustion in the Internet of Things” In: The 12th IEEE International Conference on Sensing Communication and Networking (SECON 2015), Seattle, Washington, USA, 2015.
  - M. Tiloca, C. Gehrmann and L. Seitz, "On improving resistance to Denial of Service and key provisioning scalability of the DTLS handshake", International Journal of Information Security, pp. 1-21, Springer, 2016.

Device theft protection (I)

- Some different options:
  - Tamper resistant protection of keys etc. *on device* using secure hardware modules
  - Physical protected location of device
  - Key calculation schemes dependent on key material from *several locally present* units (see next slide)

Device theft protection (II)



K_G - Local group shared secret

Individual shared secrets:

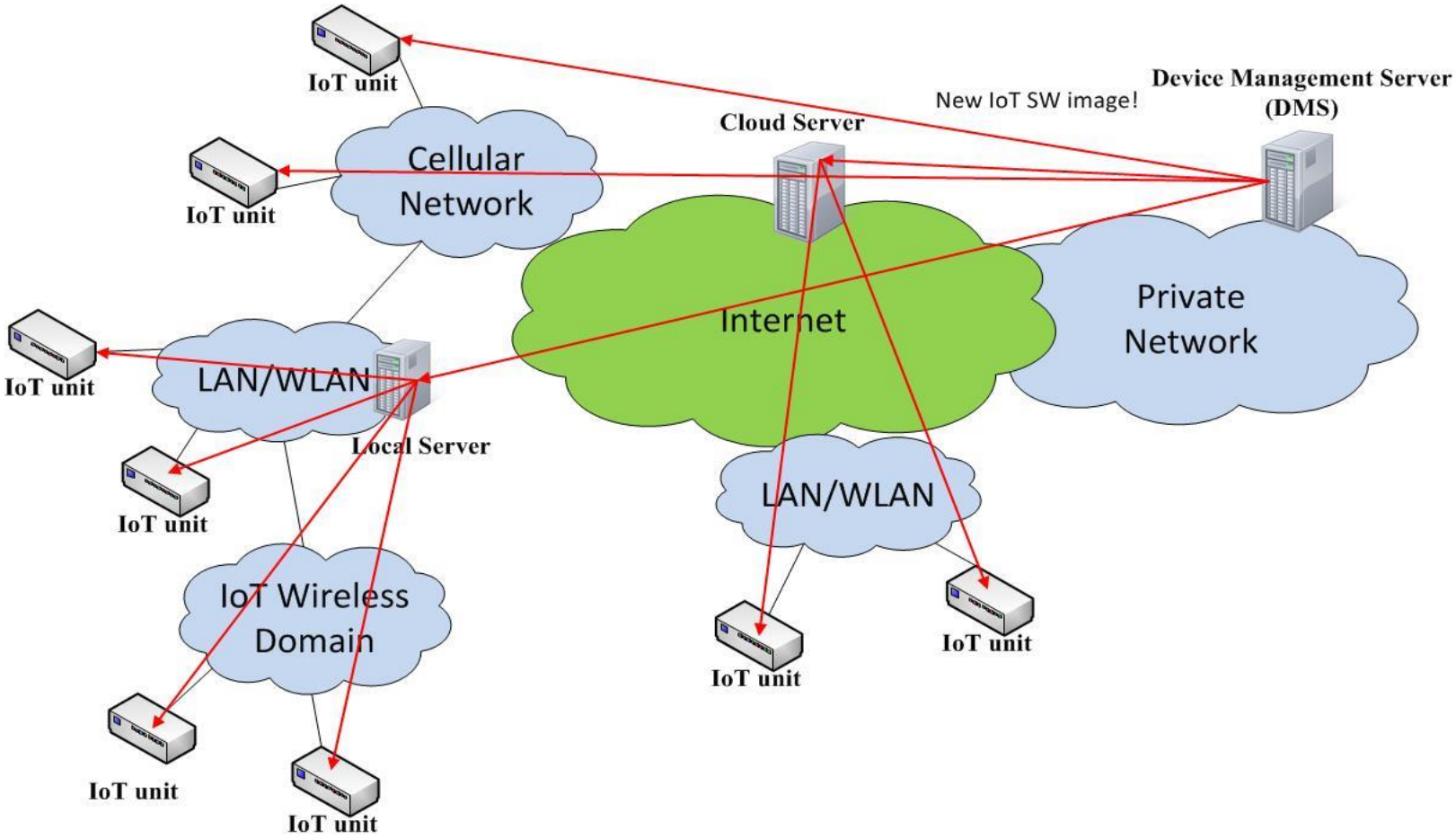
Group G

- K1 – shared secret with unit 1 in group G
- K2 – shared secret with unit 2 in group G
  :
- Kn – shared secret with unit n in group G

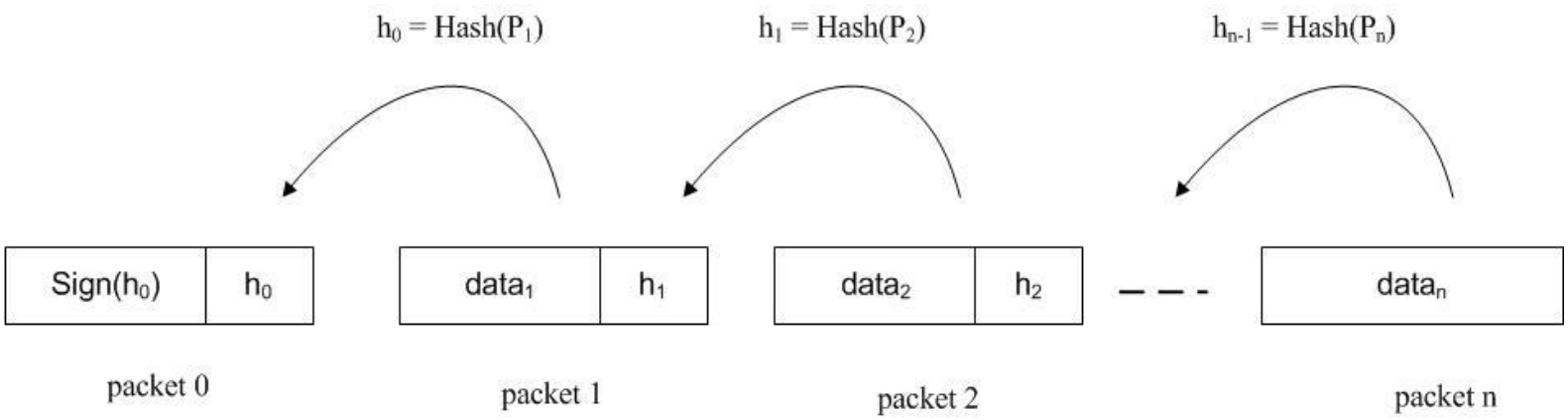
Group B

:

Secure SW upgrade (I)



Secure SW upgrade (II)

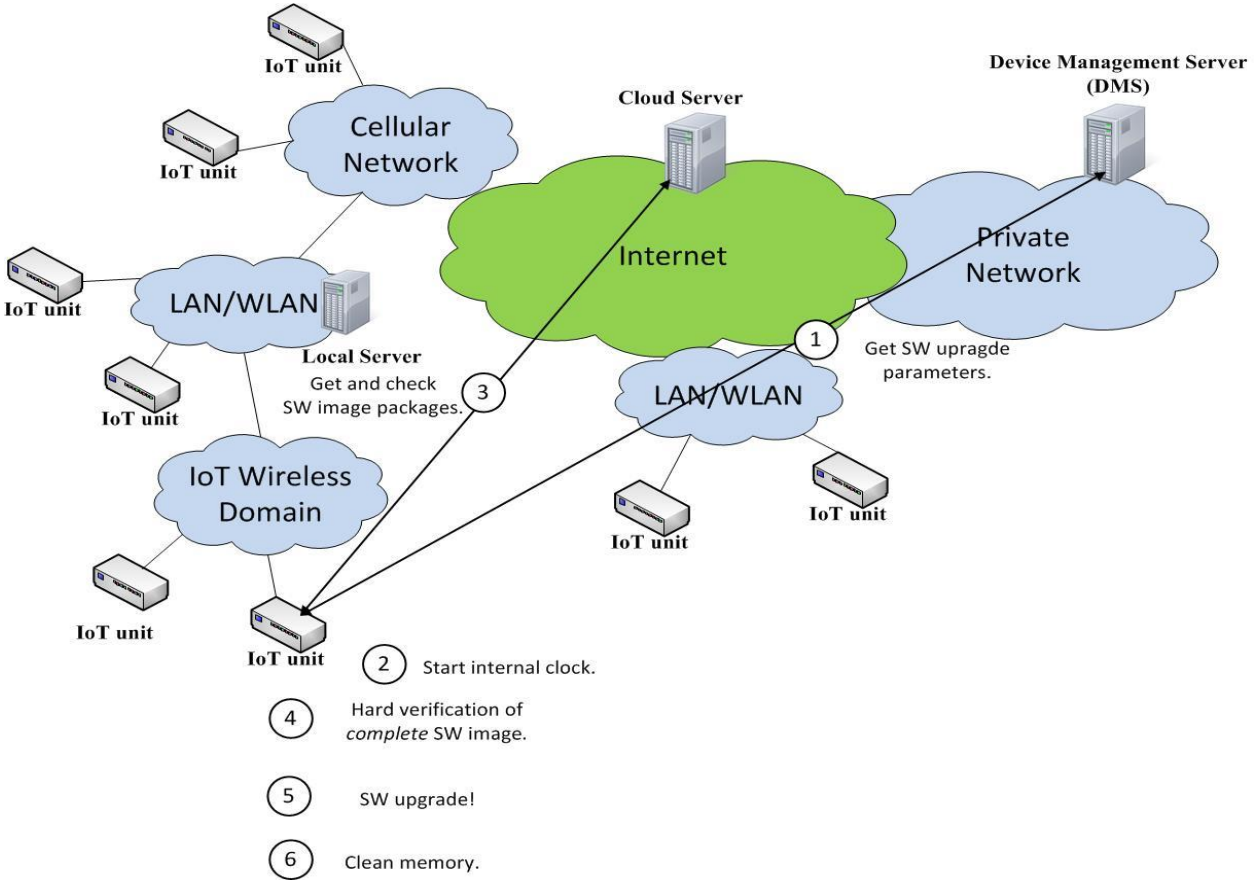


J. Deng, R. Han and S. Mishra, "Secure Code Distribution in Dynamically Programmable Wireless Sensor Networks", Information Processing in Sensor Networks, IPSN 2006, pp. 292-300, 2006.

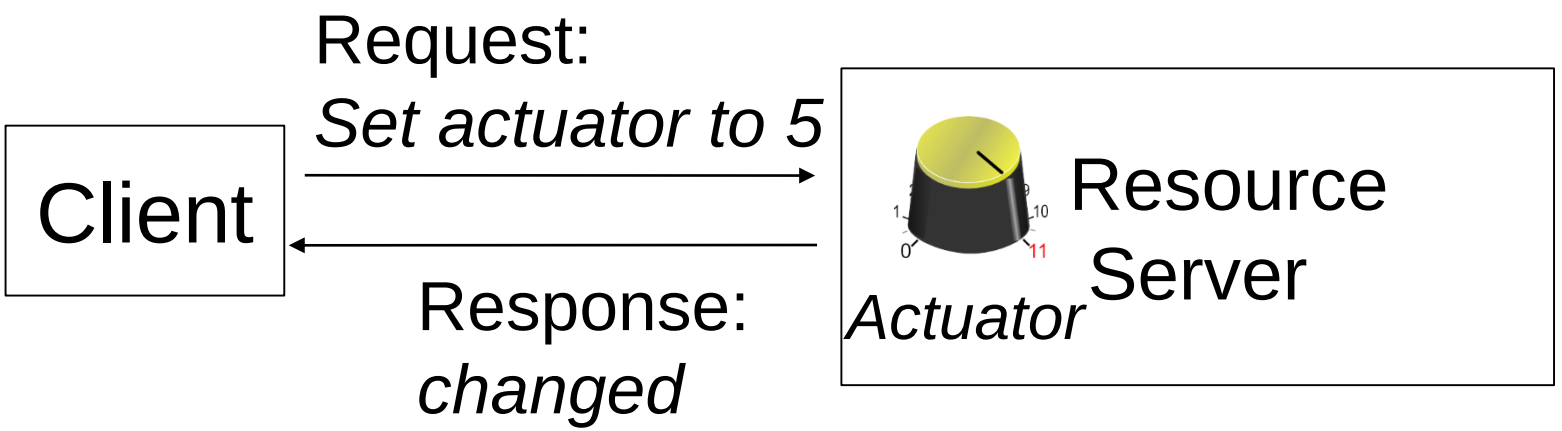
Secure SW upgrade (III)

- SW upgrade based on *existing security relation* between DMS and IoT unit
- Let the DMS do the following:
  - Generate random symmetric integrity protection and encryption keys.
  - Split the SW update image into n distinct parts.
  - Use the selected symmetric keys to generate n distinct SW upgrade packages
  - Distribute the SW packages to one or several SW image distribution servers.
  - Notify the IoT units of the availability of a new SW upgrade image and contact each of the IoT units in the system, set-up a secure connection with each of them and transfer securely the SW update parameters, including the generated symmetric SW upgrade protection keys and the *final SW package hash* (need not be signed), to the units.

Secure SW upgrade (IV)

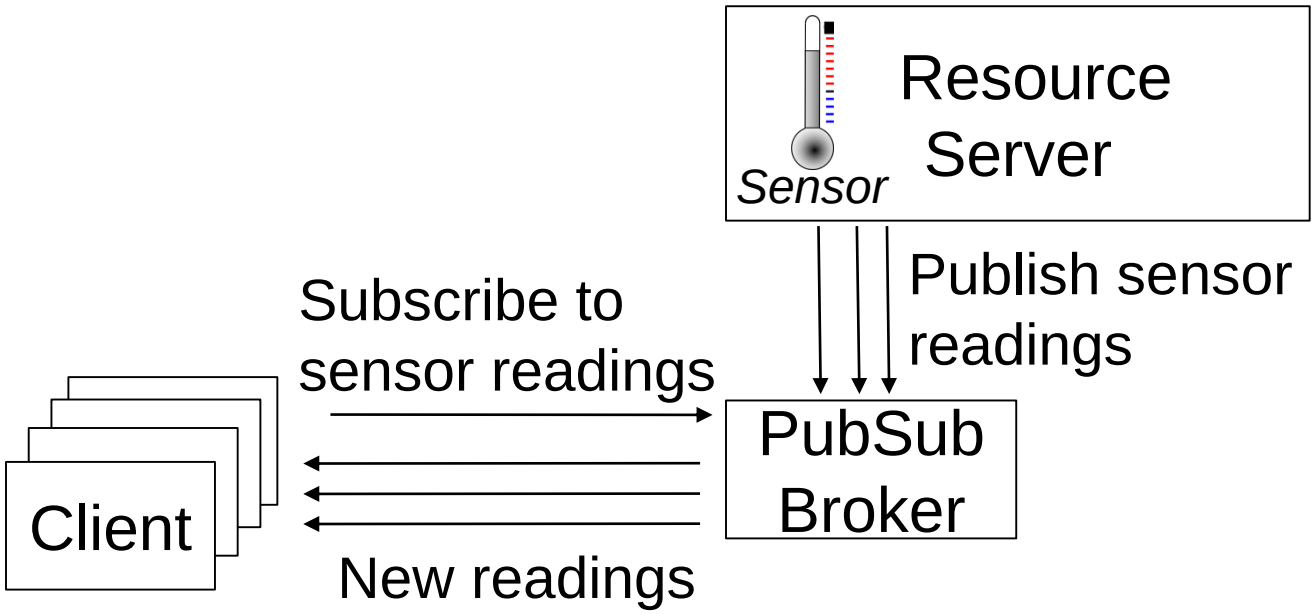


IoT Access control (I)



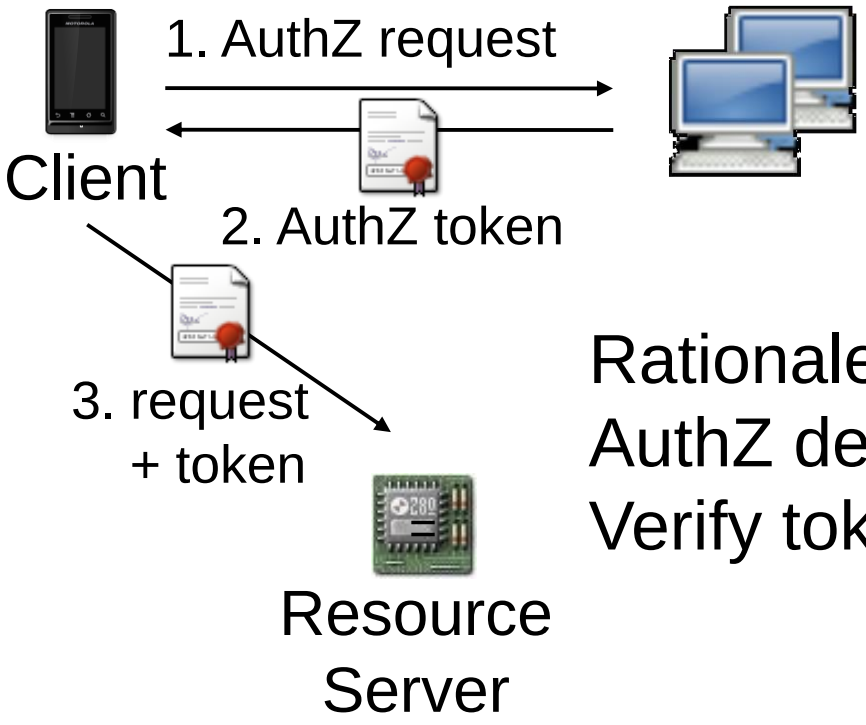
- RS needs to know C is authorized
- C needs to know that the response is from RS
- Integrity and replay protection for Request/Response
- Possibly encryption for Request/Response

IoT Access control (II)



- Access to sensor readings must be controlled
- Clients need to be able to verify the origin of a sensor reading and to detect replay or fraudulent messages

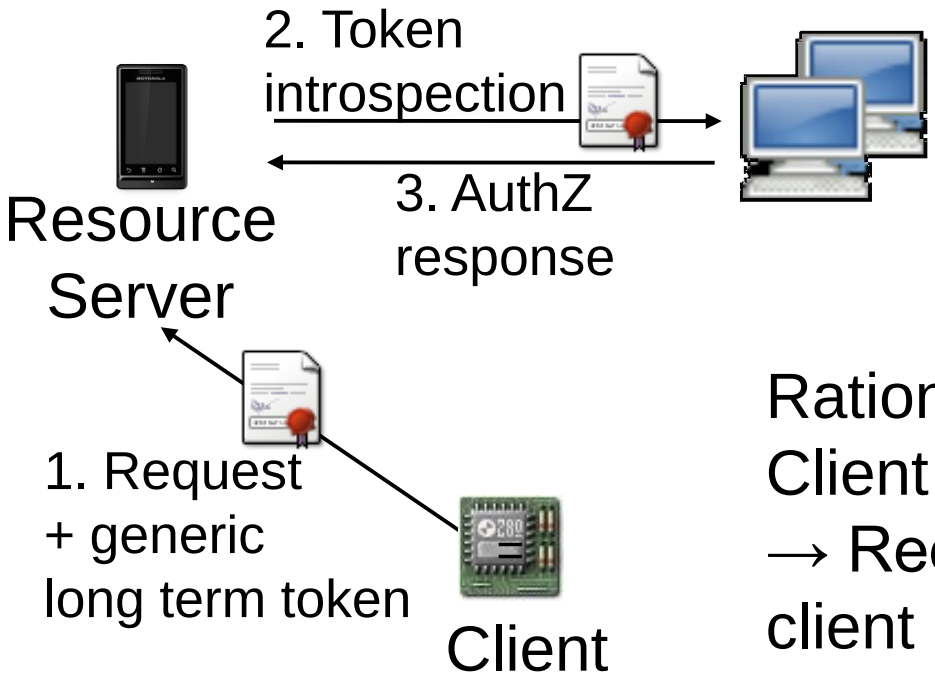
IoT Access control (III)



Rationale:
AuthZ decision = heavyweight
Verify token = lightweight

This is the basic approach, optimized for constrained servers.

IoT Access control (IV)



Rationale:
Client is the IoT device here
→ Reduce network traffic for the client

This approach is optimized for constrained clients

IoT Access control (V)

- Contributions to different IETF working groups
  - CoRE (Constrained Restful Environments)
  - ACE (authentication and authorization in constrained environments)
  - COSE (CBOR Object Signing & Encryption)
- Results: RFC 7744 (use cases and requirements)
- 4 active drafts
  - 2 have been adopted by IETF working groups (means they plan to publish them as RFCs)
    - Architecture (draft-ietf-ace-actors)
    - Authorization (draft-ietf-ace-oauth-authz)
    - Requirements for end-to-end security (draft-hartke-core-e2e-security-reqs)
    - Object security (draft-selander-ace-object-security)

Key Management (I)

- Providing key material to a large number of non-human operated units can be a rather cumbersome/expensive task
- Current mobile SIM-oriented approach does not scale well to large IoT infrastructures from device cost, trust model or maintenance cost points of view.
  - This is for the moment a major issue for dissuasion with respect to the model to use for 5G
  - Mobile operators are still very reluctant making any changes to the current SIM-oriented model
- IoT solutions are network agnostic and shall work in cellular and non-cellular systems => Proprietarily key management solutions are expected to dominate!

Key Management (II)

One possible model for key provisioning

Manufacture phase

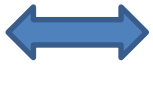
Hardware production

Generic soft credential module installation

Common credential for whole batch!

Deployment phase

Device credential configuration for batch!



End user

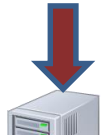
Device registration

Device credentials

Operating phase

Mutal authentication

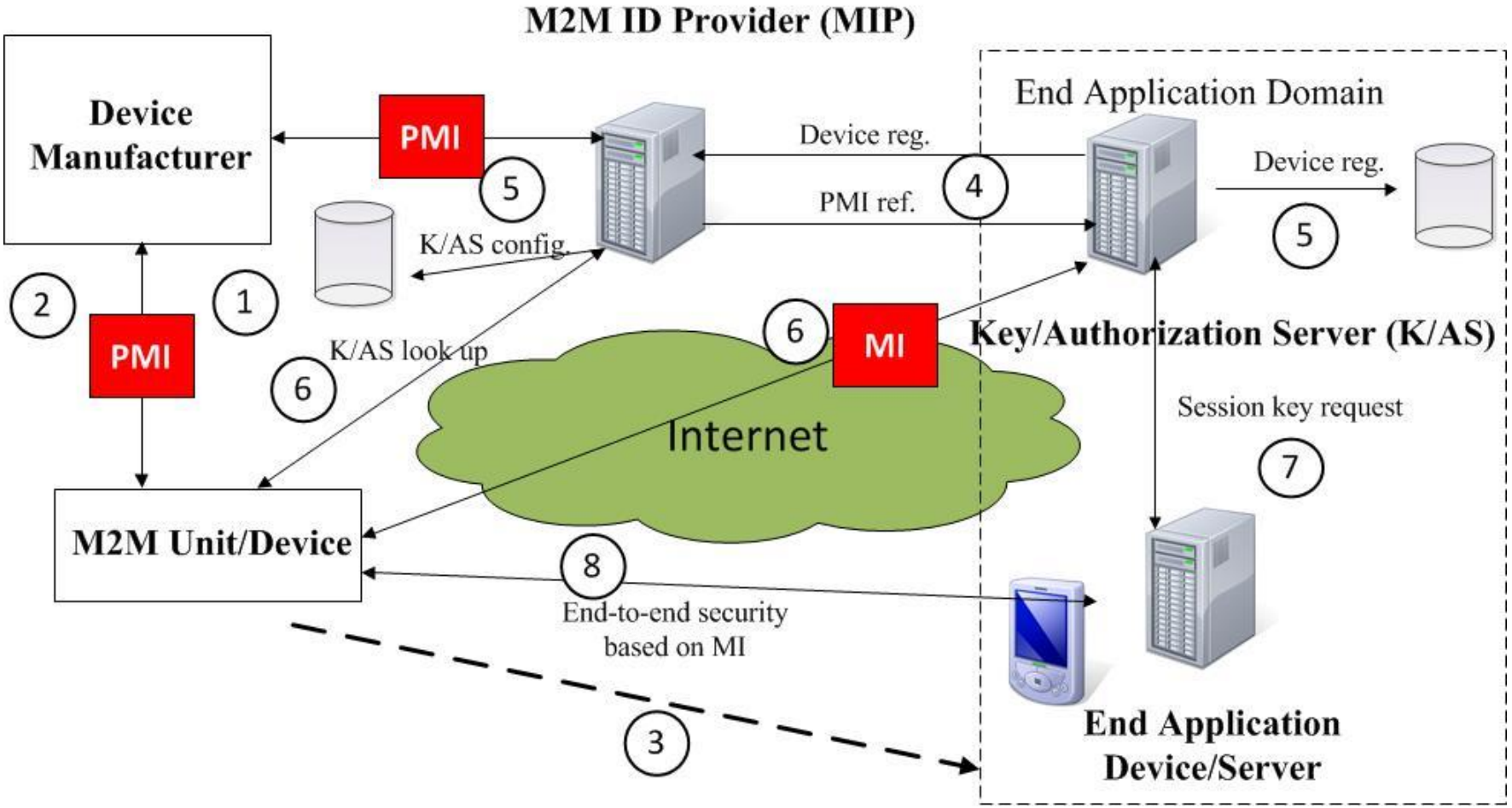
End user



Credential/firmware updates

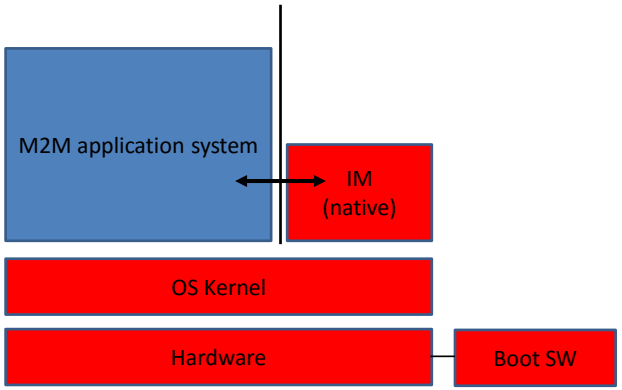
Fatal software error recovery

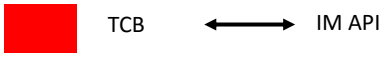
Key Management (III)

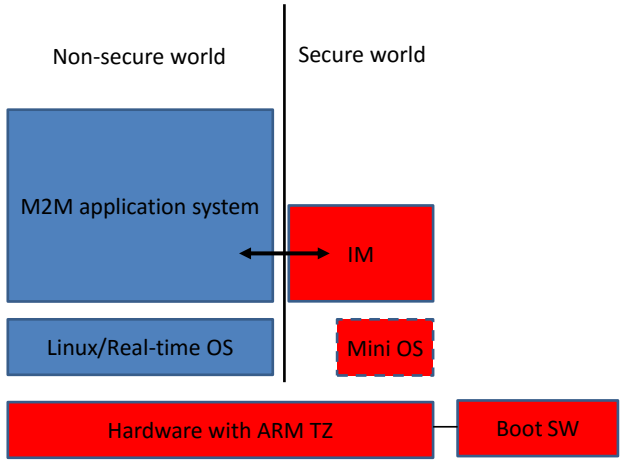


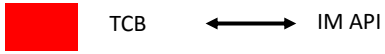
Key Mangement (IV)

Some identity module impl. options



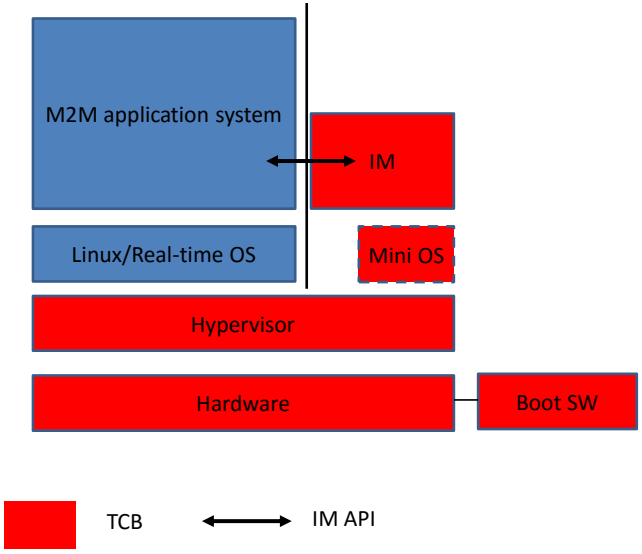


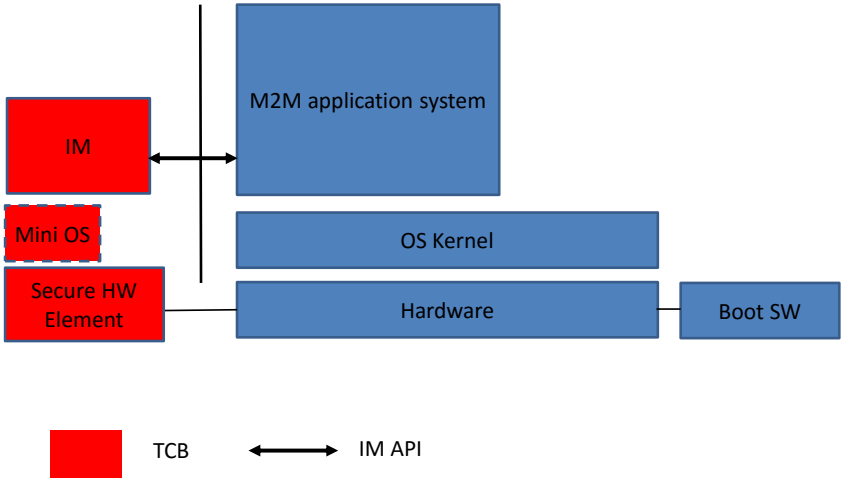




Key Mangement (V)

Some further identity module impl. options





Conclusions

- IoT systems require robust security solutions
  - “Old” attacks in slightly new settings
  - New models for credential management
- Standardized solutions will most probably dominate in the long time frame and proprietary solutions in the shorter time frame
- Good opportunities for novel security solutions and in turn also new business models