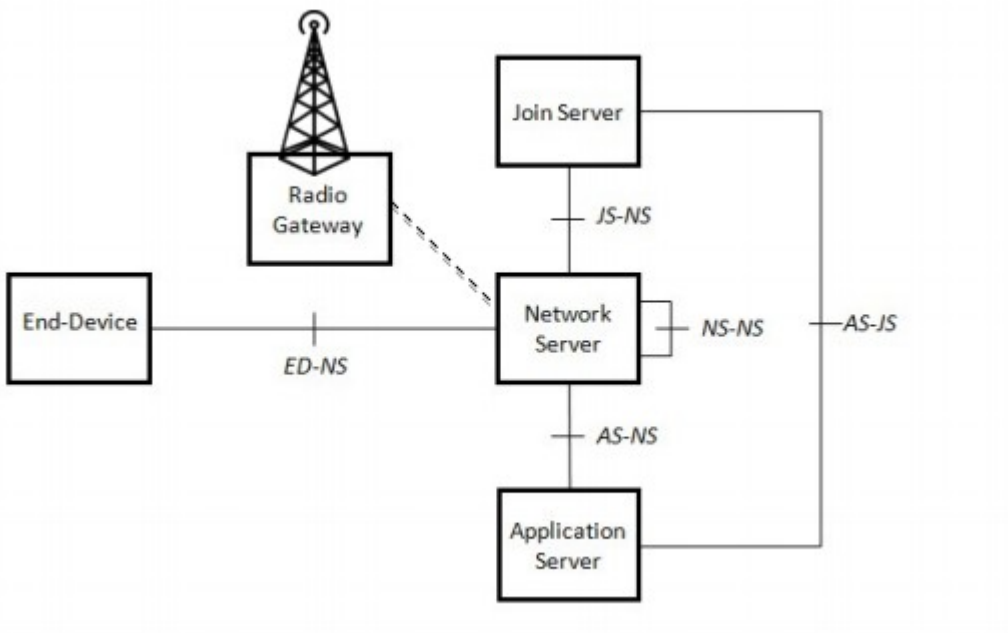
LoRaWAN Authentication with RADIUS

draft-garcia-radext-radius-lorawan-01

Dan Garcia-Carrillo (University of Murcia)
Rafael Marin-Lopez (University of Murcia)
Arunprabhu Kandasamy (Acklio)
Alexander Pelov (Acklio)

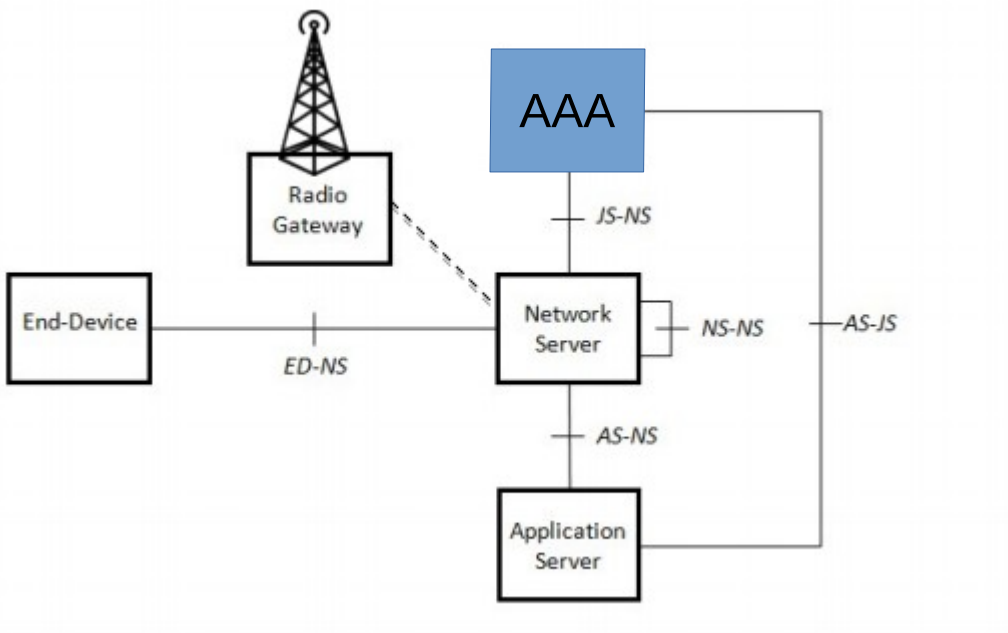
LoRaWAN – *Long Range Wide Area Network



- Long range:
  upto 20km
  **(depending on environment)
- Low Power:
  25mW, 20yrs battery life
- Data Rate:
  Upto 50kbps
- Payload:
  11-242 bytes

*Alexander Pelov, Alper Yegin Slides from lp-wan BoF
**https://hal-institut-mines-telecom.archives-ouvertes.fr/hal-01331966

LoRaWAN – *Long Range Wide Area Network



- Long range:
  upto 20km
  **(depending on environment)
- Low Power:
  25mW, 20yrs battery life
- Data Rate:
  Upto 50kbps
- Payload:
  11-242 bytes

*Alexander Pelov, Alper Yegin Slides from lp-wan BoF
**https://hal-institut-mines-telecom.archives-ouvertes.fr/hal-01331966

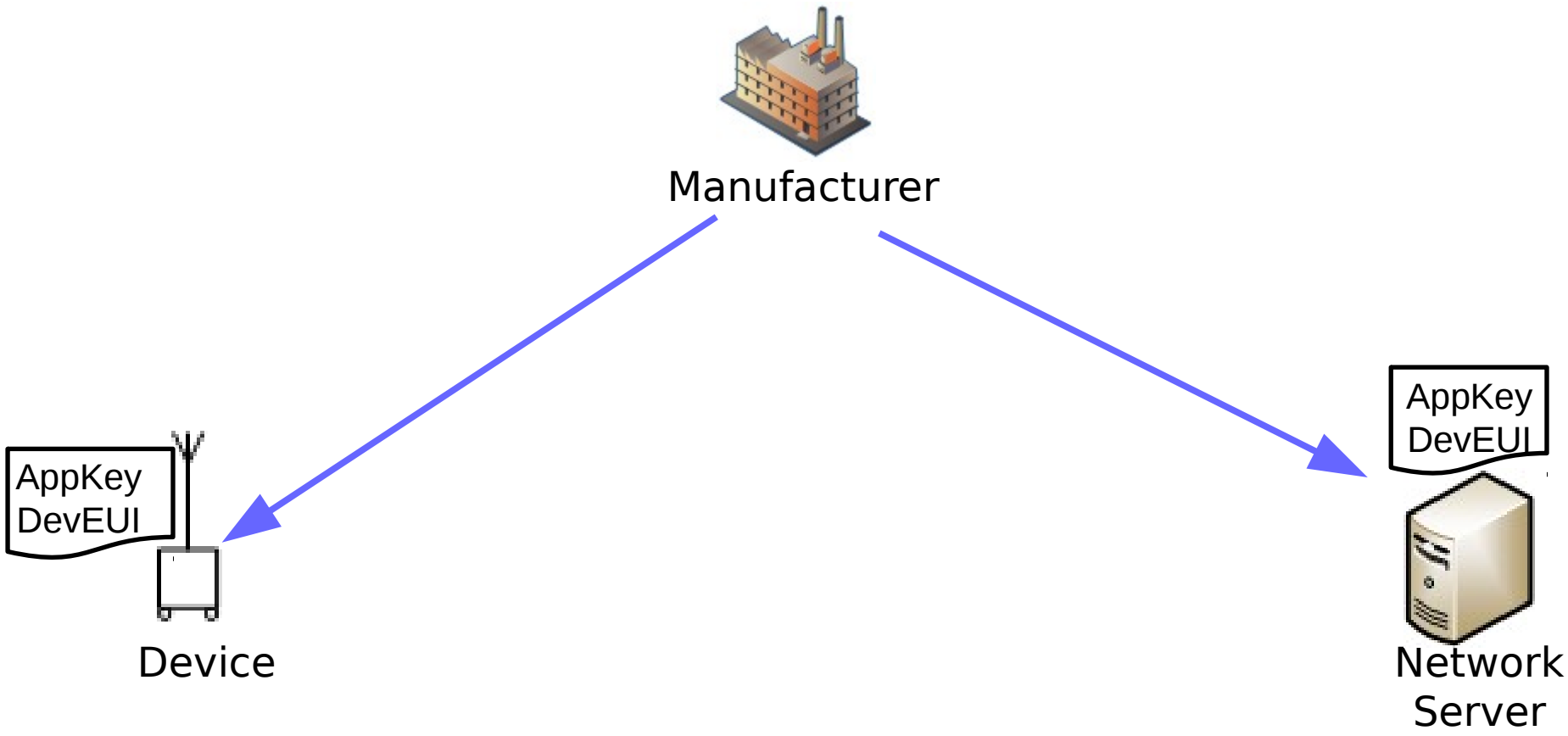
LoRaWAN Authentication - Motivation

- LoRaWAN does not reuse standards
- Include a standard authentication, AAA, framework in LoraWAN.
  - AAA infrastructure proven to be well known, battle-tested techs. Deployed in the wild, since years.. ex: eduroam
  - *Scalable, federation aware

* “Security in IoT” lectures from Rafa Marin Lopez, University of Murcia Spain, at Telecom Bretagne, France

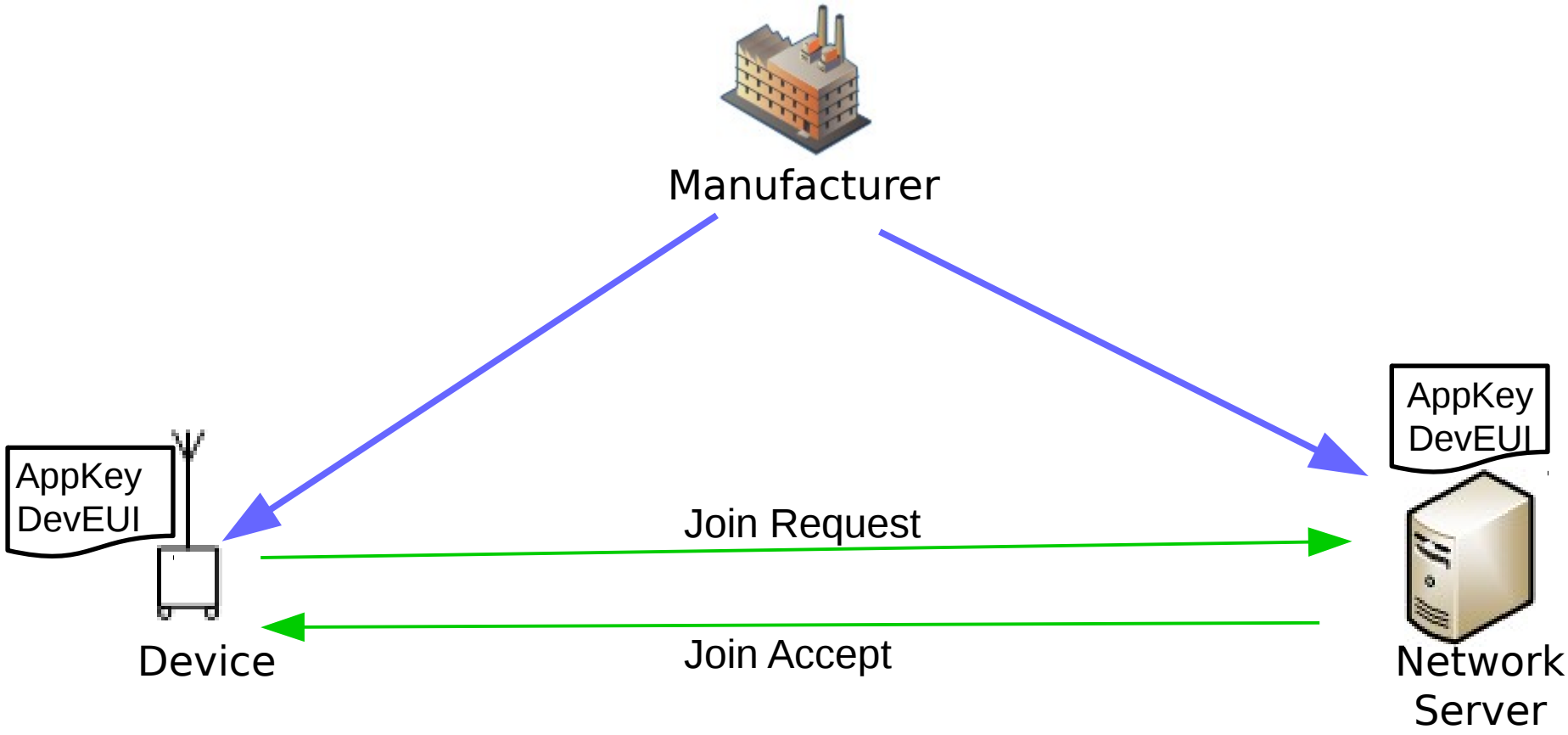
LoRaWAN Authentication - LoRaWAN 1.0 Join procedure

1. Commissioning



LoRaWAN Authentication - LoRaWAN 1.0 Join procedure

1. **Commissioning**
2. **Over the Air Activation (Join Procedure)**



LoRaWAN Authentication - Message Definition

- The request (join-request)

Size(bytes)	8	8	2
Join Request	AppEUI	DevEUI	DevNonce

- The Response (join-accept)

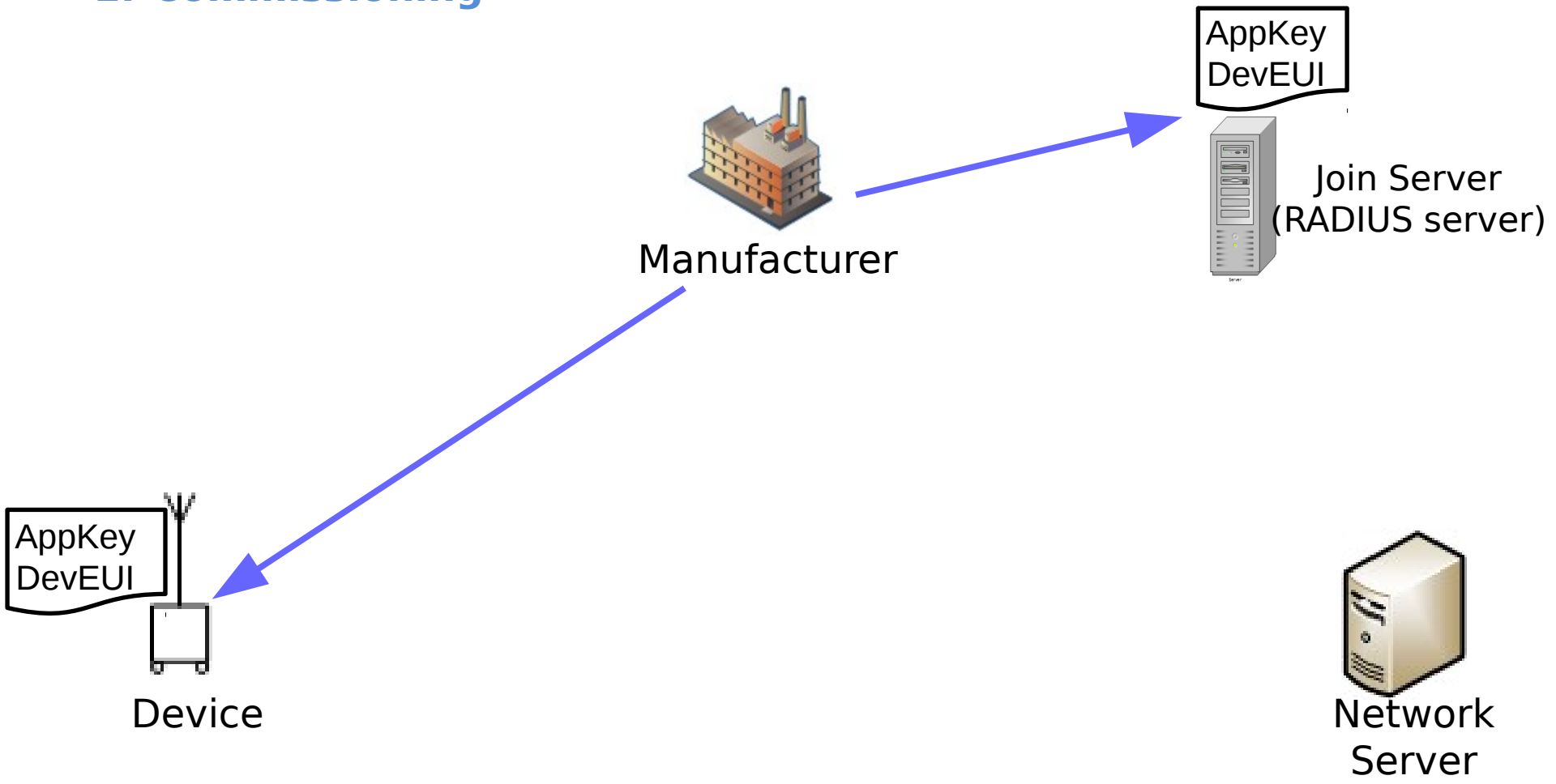
Size(bytes)	3	3	4	1	1	16(opt)
Join Accept	AppNonce	NetID	DevAddr	DLSettings	RxDelay	CFList

LoRaWAN Authentication - Keys

- AppKey
  - specific for the end-device that is assigned by the application owner to the end-device
- NwkSKey
  - to decrypt the MAC commands
- AppSKey
  - to decrypt the Application specific data

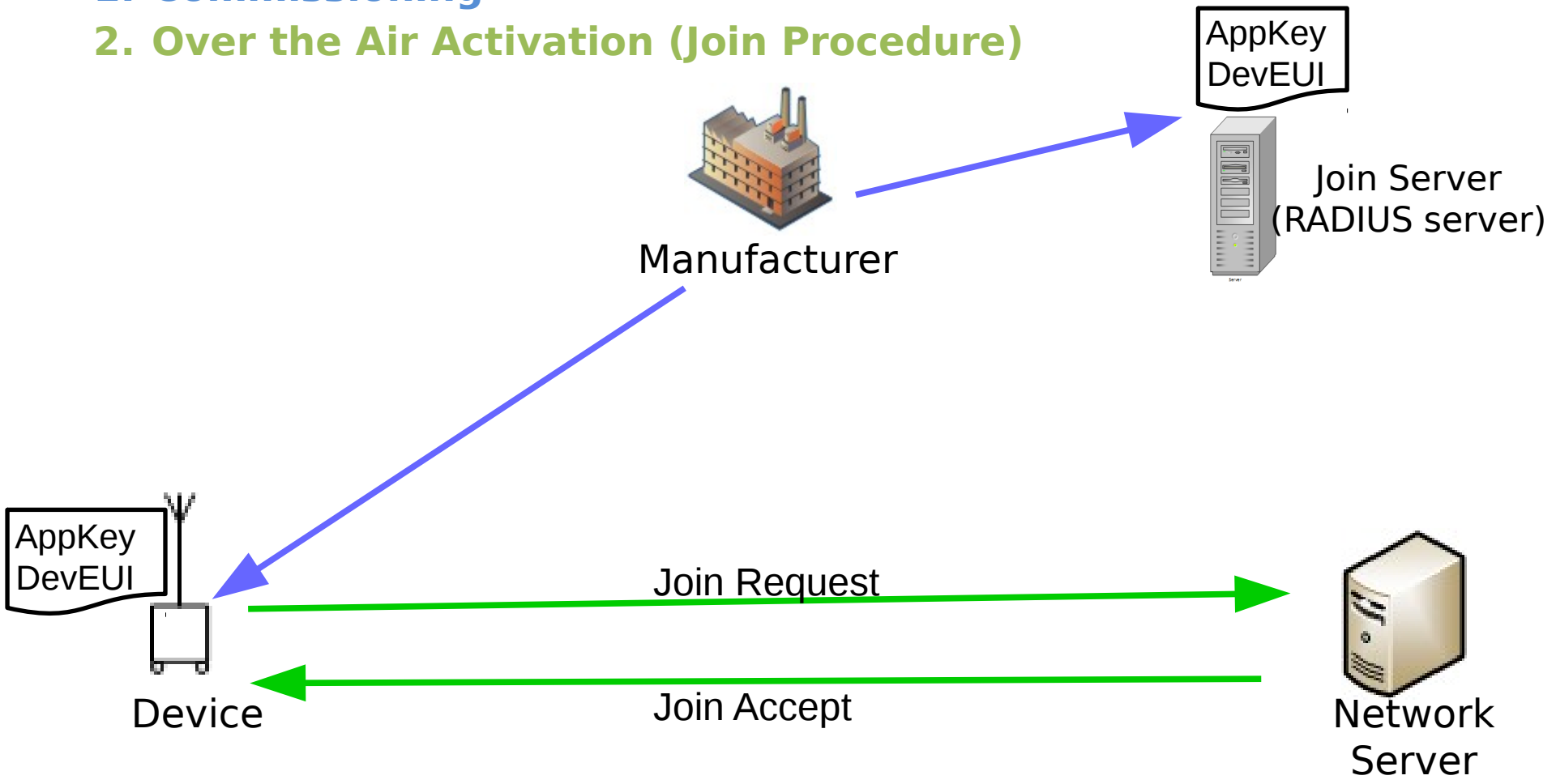
LoRaWAN Authentication - LoRaWAN 1.0 Join procedure

1. Commissioning



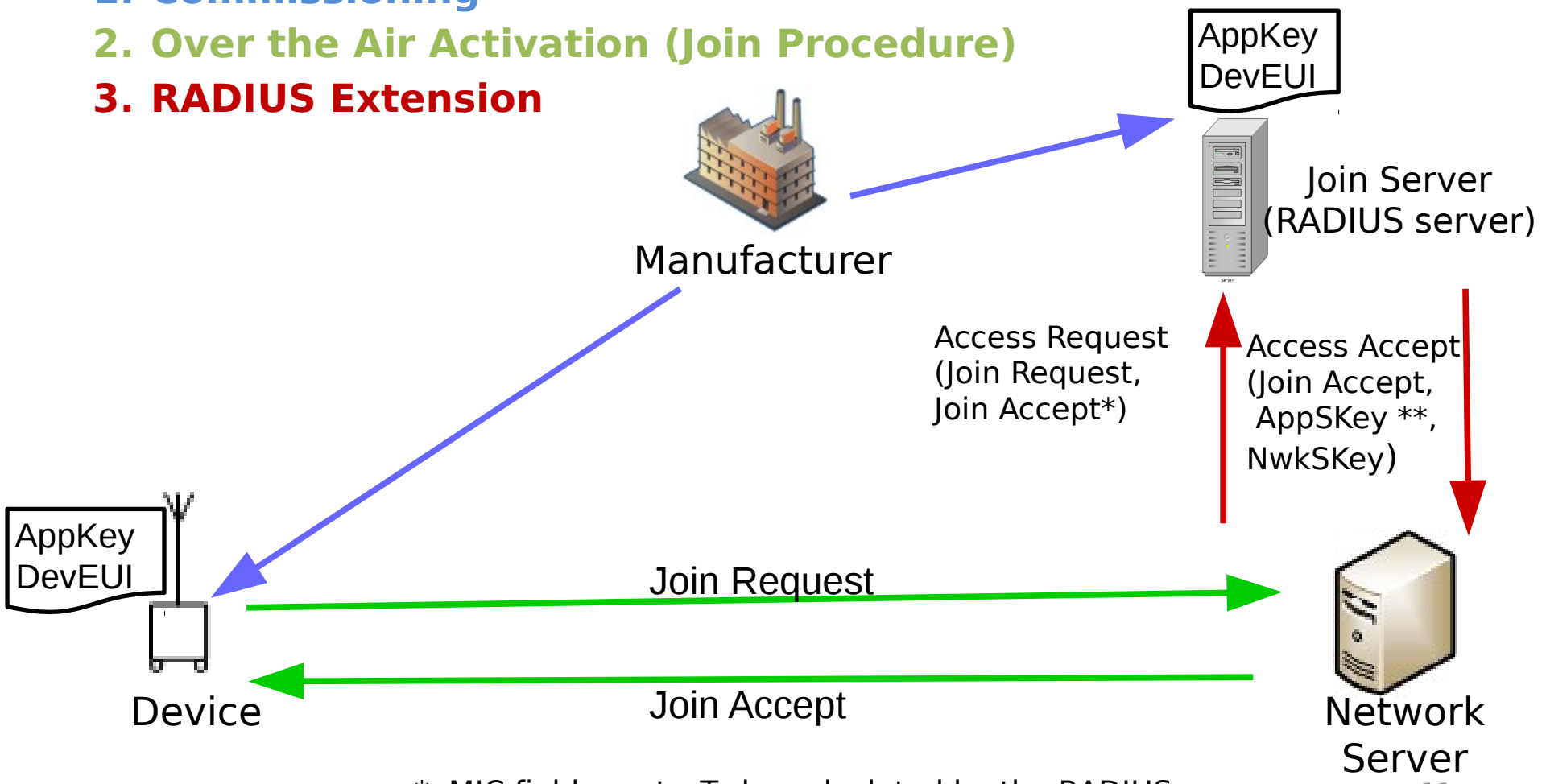
LoRaWAN Authentication - LoRaWAN 1.0 Join procedure

1. **Commissioning**
2. **Over the Air Activation (Join Procedure)**



LoRaWAN Authentication - LoRaWAN 1.0 Join procedure

1. **Commissioning**
2. **Over the Air Activation (Join Procedure)**
3. **RADIUS Extension**



* MIC field empty. To be calculated by the RADIUS server.
** Optional field.

LoRaWAN Authentication with RADIUS

- New RADIUS Attributes
  - JoinRequest (containing the join-request)
  - JoinAnswer (containing the join-accept)
  - Nwkskey (containing the NwkSKey)
  - Appskey (containing the AppSKey, optional)

- Keys are transported as RADIUS attributes

  Consideration for transporting key materials securely
    - Similar to RFC 6218 (Cisco Vendor-Specific RADIUS Attributes for the Delivery of Keying Material)
    - RFC6614 (Radius over TLS)

LoRaWAN Authentication with RADIUS

- Open Issues
  - The Join Request has AppEUI indicating the Organization, but to route JR and JA* through the AAA infrastructure we need to specify a realm (e.g. um.es).

00 11 22 33 44 55 66 77 88 99 AA  um.es

  - A mechanism for matching the AppEUI to the domain name of the organization is needed.
  - Possible solution would be a inverse approach of [RFC7043] using DNS.

LoRaWAN Authentication with RADIUS

- Proof of concept Implementation

- End Device (Nemeus)
  - usb key with Java app

- Base station (ExpEmB)
  - Intel Atom, 2GB RAM

- Lora Network server(Acklio)
  - implemented in golang

- RADIUS(bronze1man)
  -implemented in golang

Next: Implementation in FreeRADIUS

- [Lora Base station] www.expemb.com/en/product/multi-connectivity-service-gateway-sgwmc-x86lr-12132/
- [LoRa Network Server] www.ackl.io
- [End-device] www.nemeus.fr/en/mk002-usb-key
- [Radius] github.com/bronze1man/radius

LoRaWAN Authentication with RADIUS

- Acknowledgements
  - Thanks, to Sri Gundavelli, Yeoh Chun-Yeow, Alan DeKok, Stephen Farrell and Mark Grayson., for their valuable comments
  - This work has been possible partially by:
    - The SMARTIE project (FP7-SMARTIE-609062 EU Project)
    - The Spanish National Project CICYT EDISON (TIN2014-52099-R) granted by the Ministry of Economy and Competitiveness of Spain (including ERDF support).

Comments and Questions?

- Thanks for your attention

Backup Slides

- Key and MIC calculation

join-request
cmac = aes128_cmac(AppKey, MHDR | AppEUI | DevEUI | DevNonce)
MIC = cmac[0..3]

join-accept
cmac = aes128_cmac(AppKey,MHDR | AppNonce | NetID | DevAddr | RFU | RxDelay | CFList)
MIC = cmac[0..3]

NwkSKey = aes128_encrypt(AppKey, 0x01 | AppNonce | NetID | DevNonce | pad16)
AppSKey = aes128_encrypt(AppKey, 0x02 | AppNonce | NetID | DevNonce | pad16)

The pad16 function appends zero octets so that the length of the data is a multiple of 16.
MIC*[RFC4493]

Backup Slides

- Size of RADIUS attributes

  (incl. Type & length fields)

  Request = 25B

  Accept = 19B

  NwkSKey = 34B

  AppSKey = 34B

Backup Slides

- Nonce

  The DevNonce can be extracted by issuing a sequence of RSSI measurements under the assumption that the quality of randomness fulfills the criteria of true randomness