Dr. Sebastian Obermeier, ABB Corporate Research Switzerland, Nov 2009

Certificate Management for Embedded Industrial Systems

Co-Authors
Hadeli, ABB Corporate Research, Switzerland
Ragnar Schierholz, ABB Corporate Research, Switzerland
Robert Enderlein, EPFL Lausanne, Switzerland

Embedded Devices

Embedded Device Usage

- Substation environment:
  - Intelligent Electronic Devices (IED)
  - Gateways
  - Remote Terminal Unit (RTU)
  - Human Interface Devices (HID)
  - Primary Equipment (Sensors, Circuit breakers)

- Process control environment:
  - OPC client, OPC server
  - Controllers
- Robotics:
  - Robot controller
- …

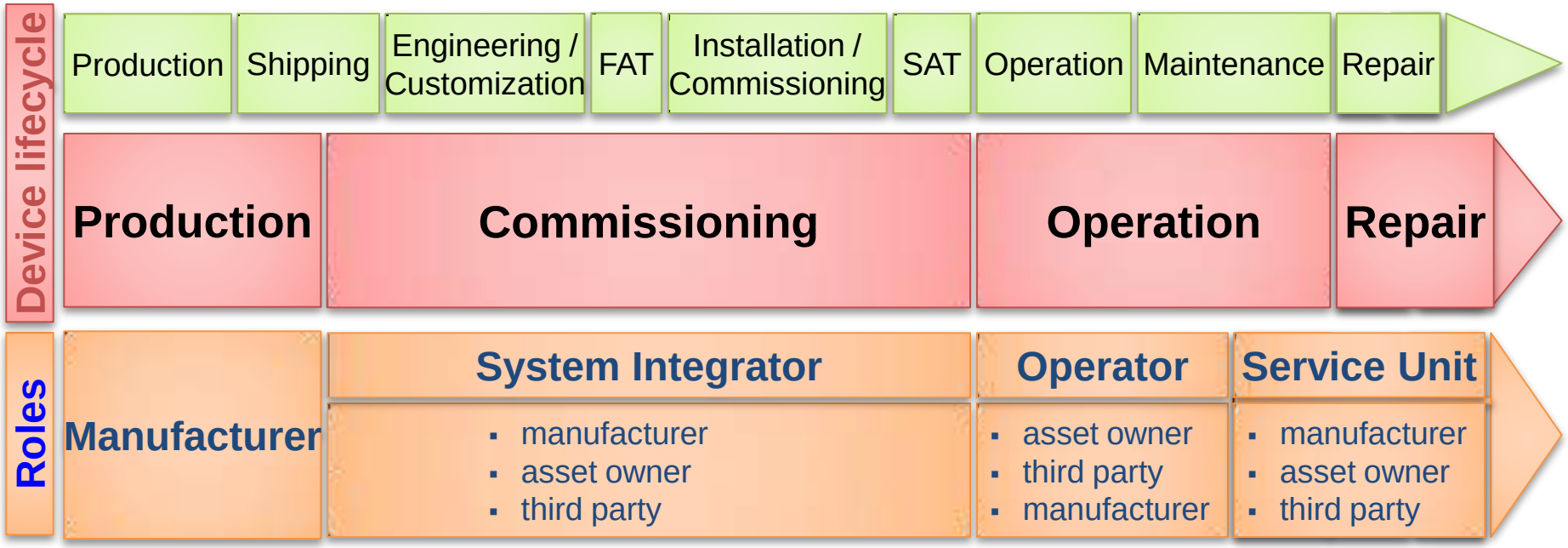
Certificate Management – Standards Demand

- Use of X.509 certificates demanded by various standards:
  - IEC 62351:
    - IEC 60870-5-104
    - IEC 61850 MMS
  - Secure DNP3/TCP
  - OPC-UA
  - …
- But: Management of certificates is not clarified or declared out-of-scope

Outline

- Embedded device lifecycle
- Certificate basics
- Initial certificate installation
- Certificate replacement
- Certificate revocation

Embedded Device Lifecycle



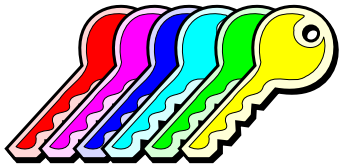
Office IT vs. Industrial Software Systems
Similar, but sufficiently different

	Office IT	Process control systems
Primary object under protection	Information	Physical process
Primary risk impact	Information disclosure, financial	Safety, health, environment, financial
Security focus	Security of central servers	Stability of decentralized field devices
Availability	95 – 99% (accept. downtime/year: 18.25 - 3.65 days)	99.9 – 99.999% (accept. downtime/year: 8.76 hrs – 5.25 mins
Determinism	Hours to months	Milliseconds to hours
Operating environment	Interactive, transactional	Interactive, real-time
Problem response	Reboot, patching/upgrade	Fault tolerance, online repair

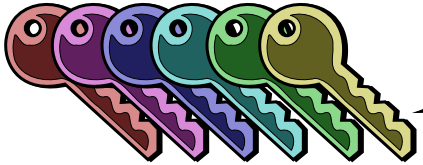
Security 101

Icons Used in this Presentation

Public keys



Private keys



Signatures

Certificates

Root (self-signed)

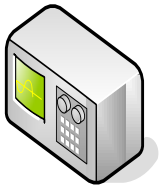
Issued by CA

Symmetric crypto

Key

Encrypted message

Devices



Embedded device

Server

Operator

Adversary

Security

- Availability
  - A third party cannot disrupt communication
- Authentication + Integrity
  - A third party cannot insert or change data
- Confidentiality
  - A third party cannot understand the data
- Non-repudiation
  - The sender of the data cannot later deny he sent it

Confidential & Authenticated Channels

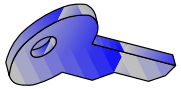
- The communicating partner is known (authentication)
- The message was not altered in transit (integrity)
- **The contents of the communication is confidential**

A confidential channel can be established:

- By using an **authenticated channel** + asymmetric cryptography
- By using **trusted certificates**
- By having a **shared secret** in common
- By using an **out-of-band** mechanism:
  - Meeting face-to-face
  - Regular mail

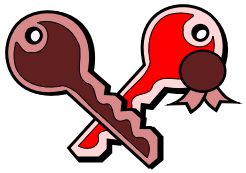
Authentication/Confidentiality with Cryptography

- Authentication can be achieved with cryptography:



- Symmetric cryptography
  - Secret keys need to remain confidential on both ends
  - $O(n^2)$ keys required

- Asymmetric (public-key) cryptography
  - Public keys need to be authenticated
  - Private keys known by one entity only
  - O(n) keys required
  - Usually used for **key distribution**, then symmetric crypto is used



Certificates

- A certificate contains:
  - A public key
  - Identity information
  - An expiration date

  Signed by a Certificate Authority (CA)

- The Root CA's certificate is signed by itself.

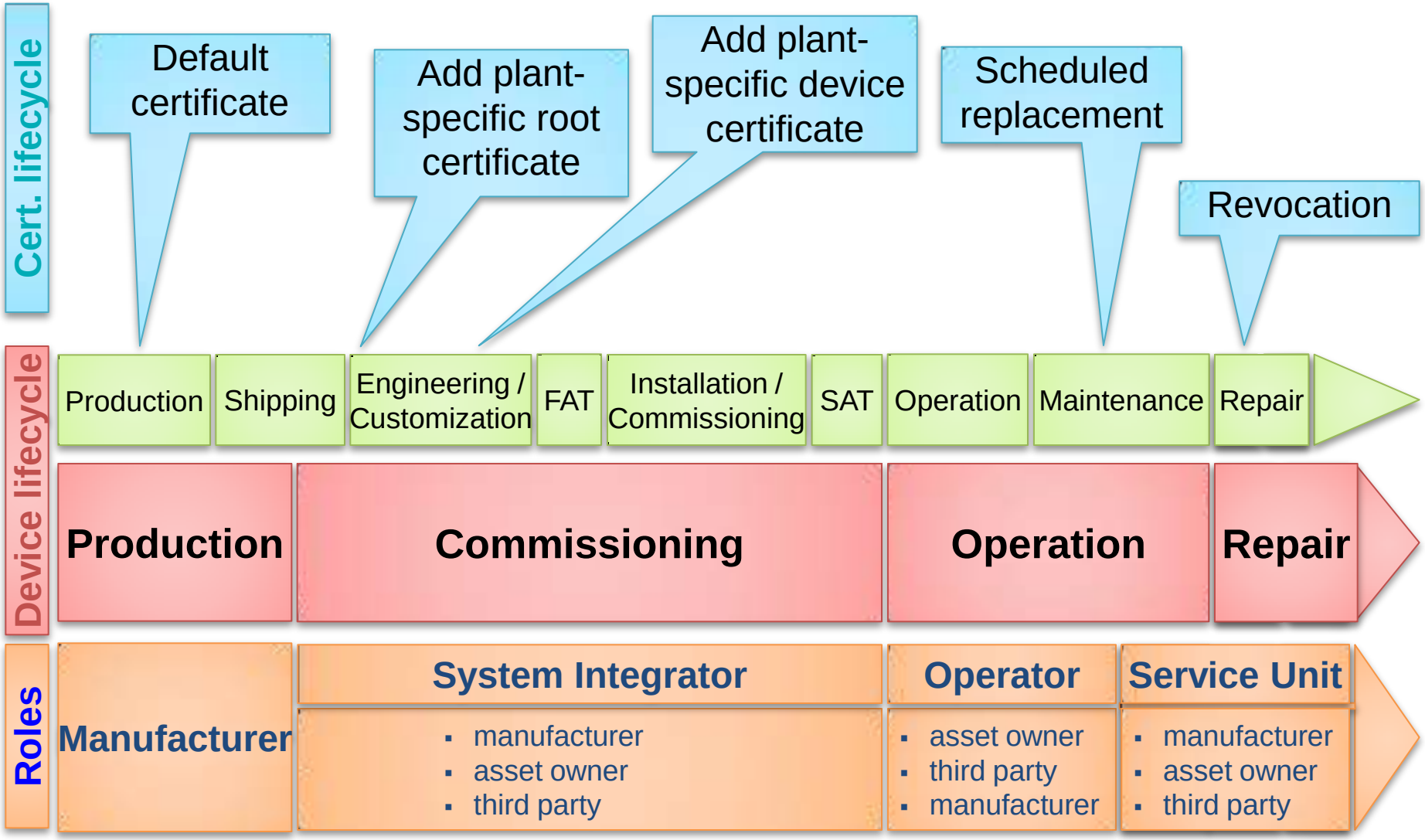
The Authentication Problem

- Messages are signed by the private key of an entity
- Whose corresponding public key (certificate) was signed by an intermediate CA's private key
- Whose corresponding public key (certificate) was signed by the root CA's private key
- Whose corresponding public key was authenticated by …?

Certificate Authorities

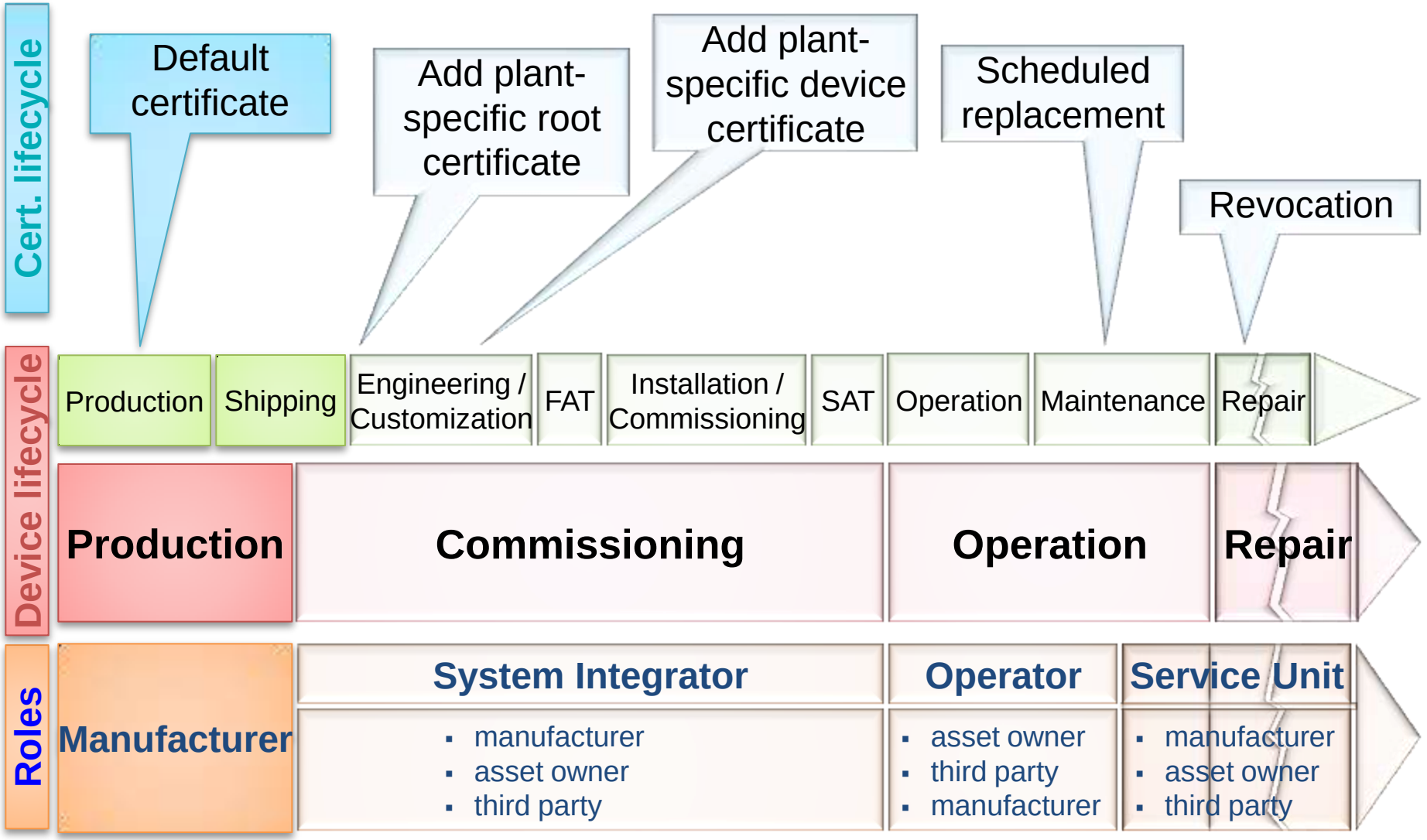
- There are at least 2 different root certificate authorities:
  - Manufacturer CA
  - System integrator CA
- Optionally, several layers of intermediate CAs

Overview of Device & Certificate Lifecycle



Initial Certificate Installation

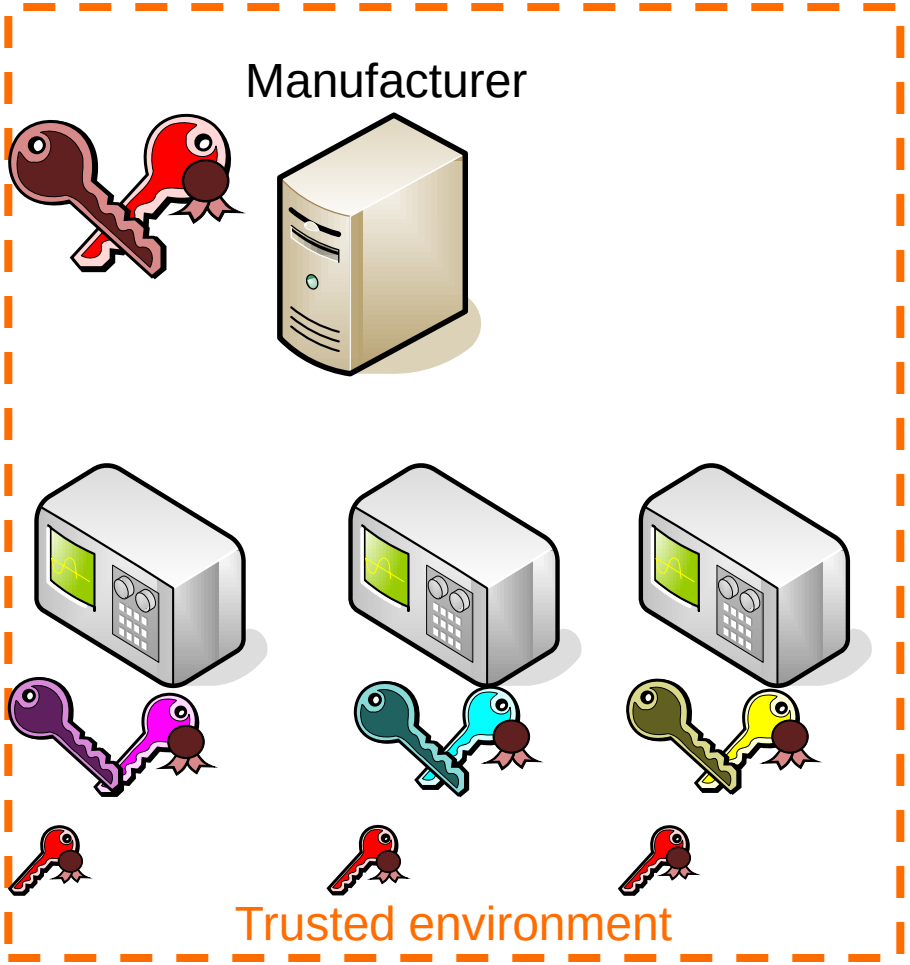
Overview of Device & Certificate Lifecycle



Issues

- Device production is anonymous: before shipping, the manufacturer does not always know where/how devices will be used
- Clients wish to choose their CA
- Automate as much as possible: nobody wants to perform an additional manual step on 50+ devices
- Chain of trust should be intact

Device Manufacturing



System integrator

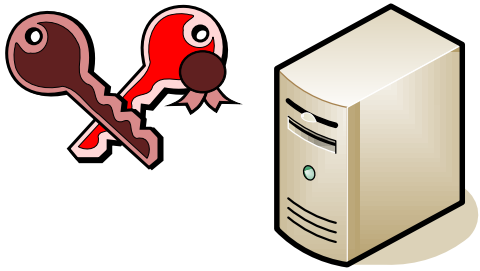
The manufacturer installs:

- a default private key
- a default certificate
- his own root certificate

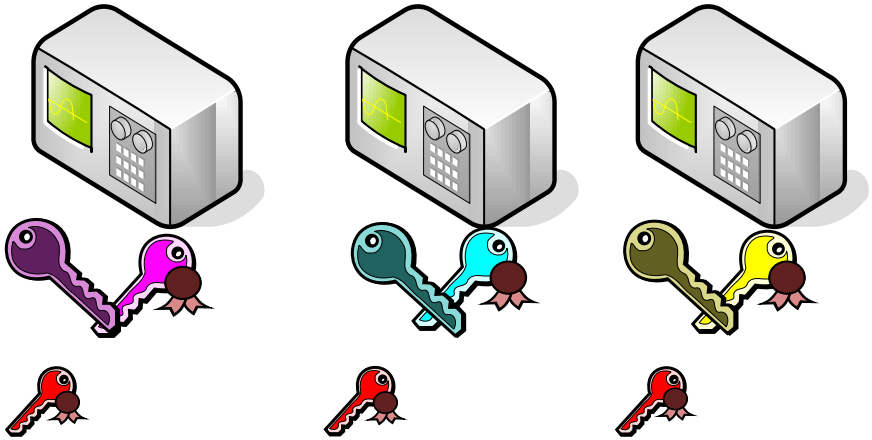
on each device

Shipping

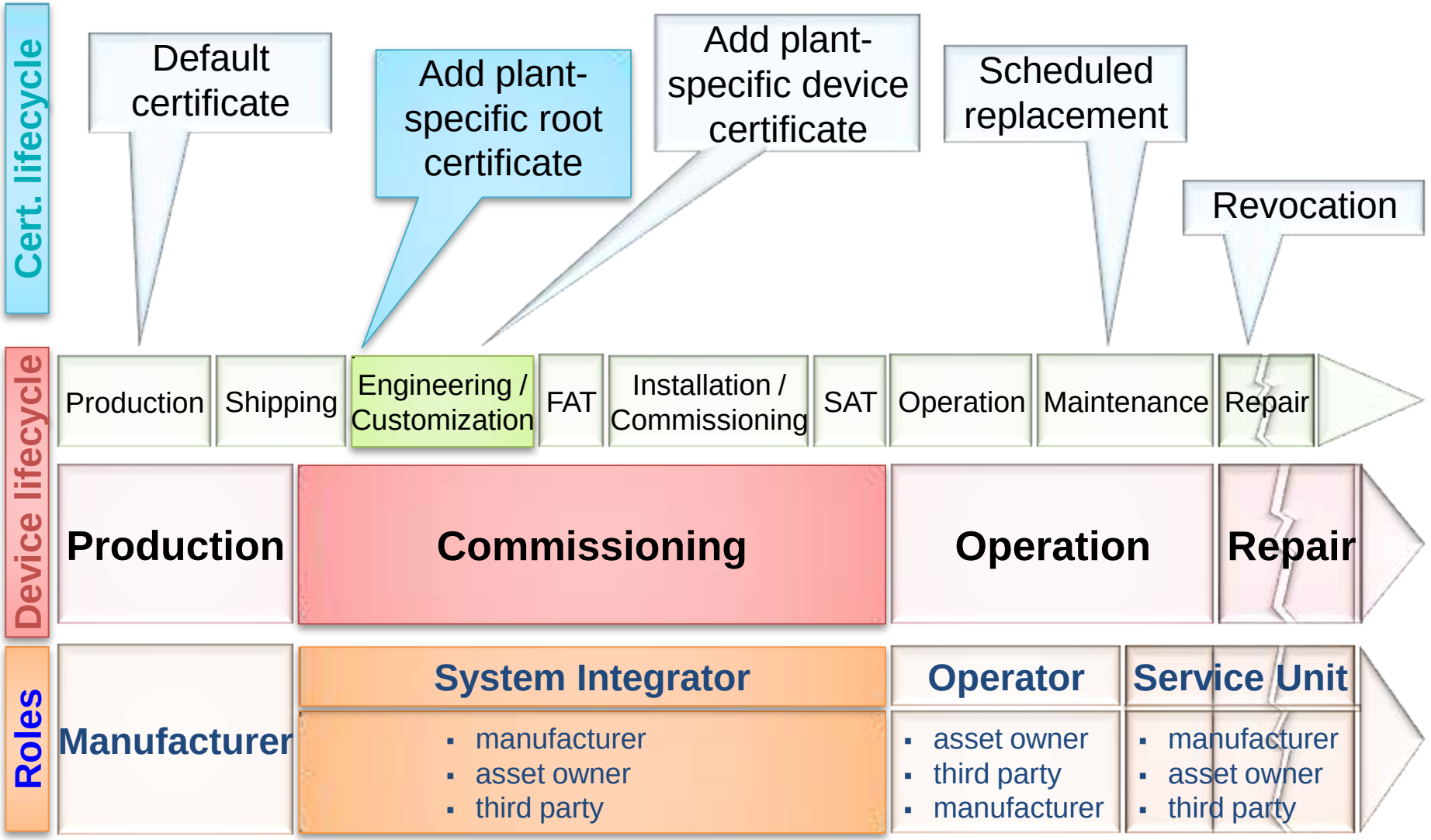
Manufacturer



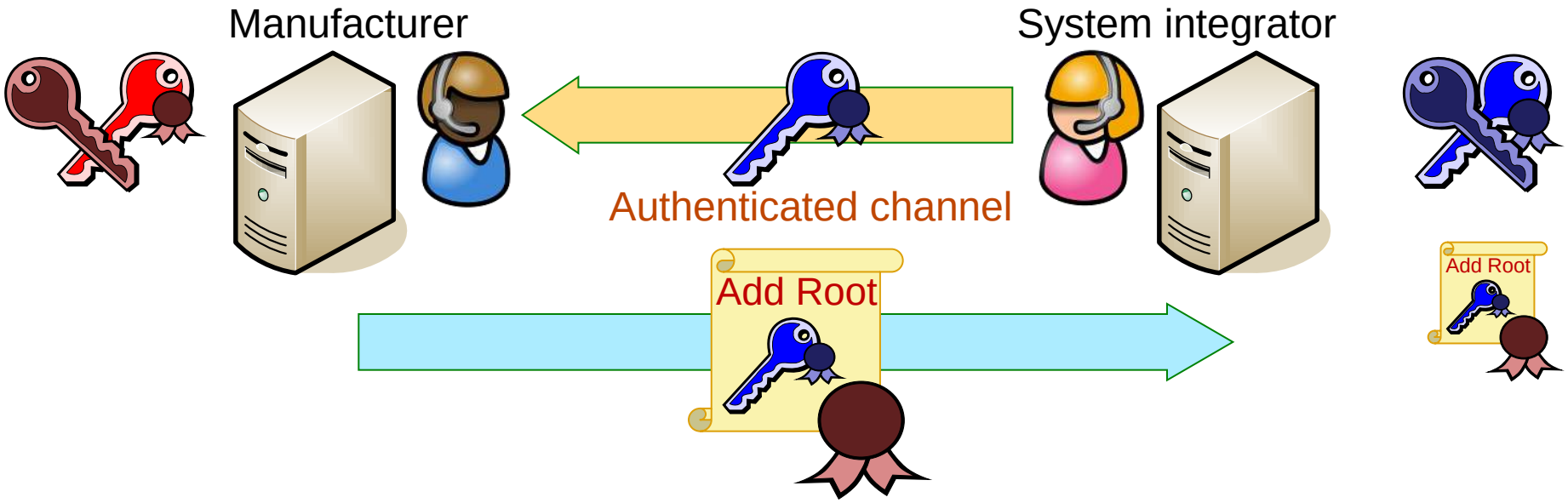
System integrator



Overview of Device & Certificate Lifecycle



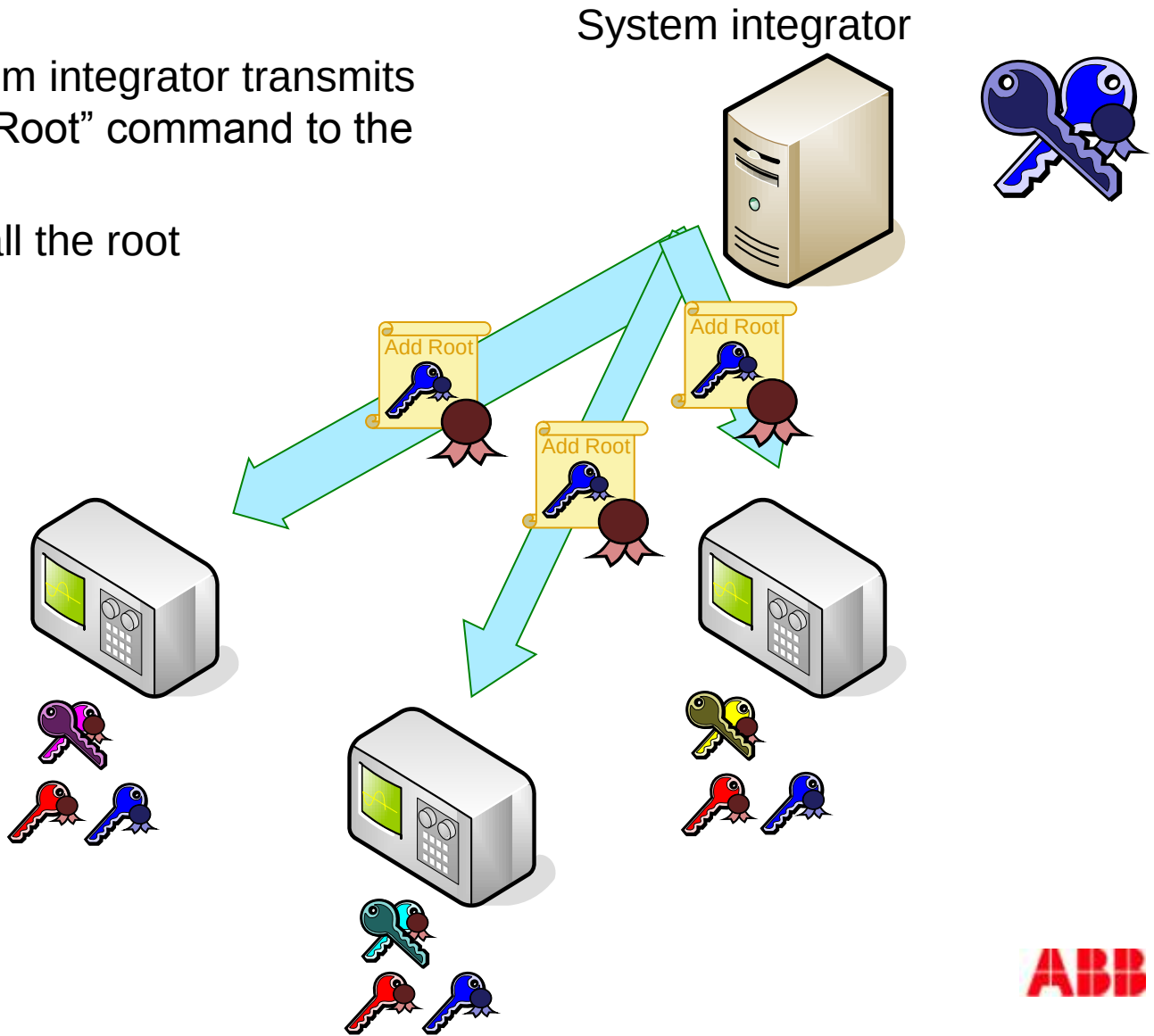
Customization



- The system integrator sets up its Certificate Authority
- He transmits his root certificate to the manufacturer
- The manufacturer signs an “Add root command” for the devices

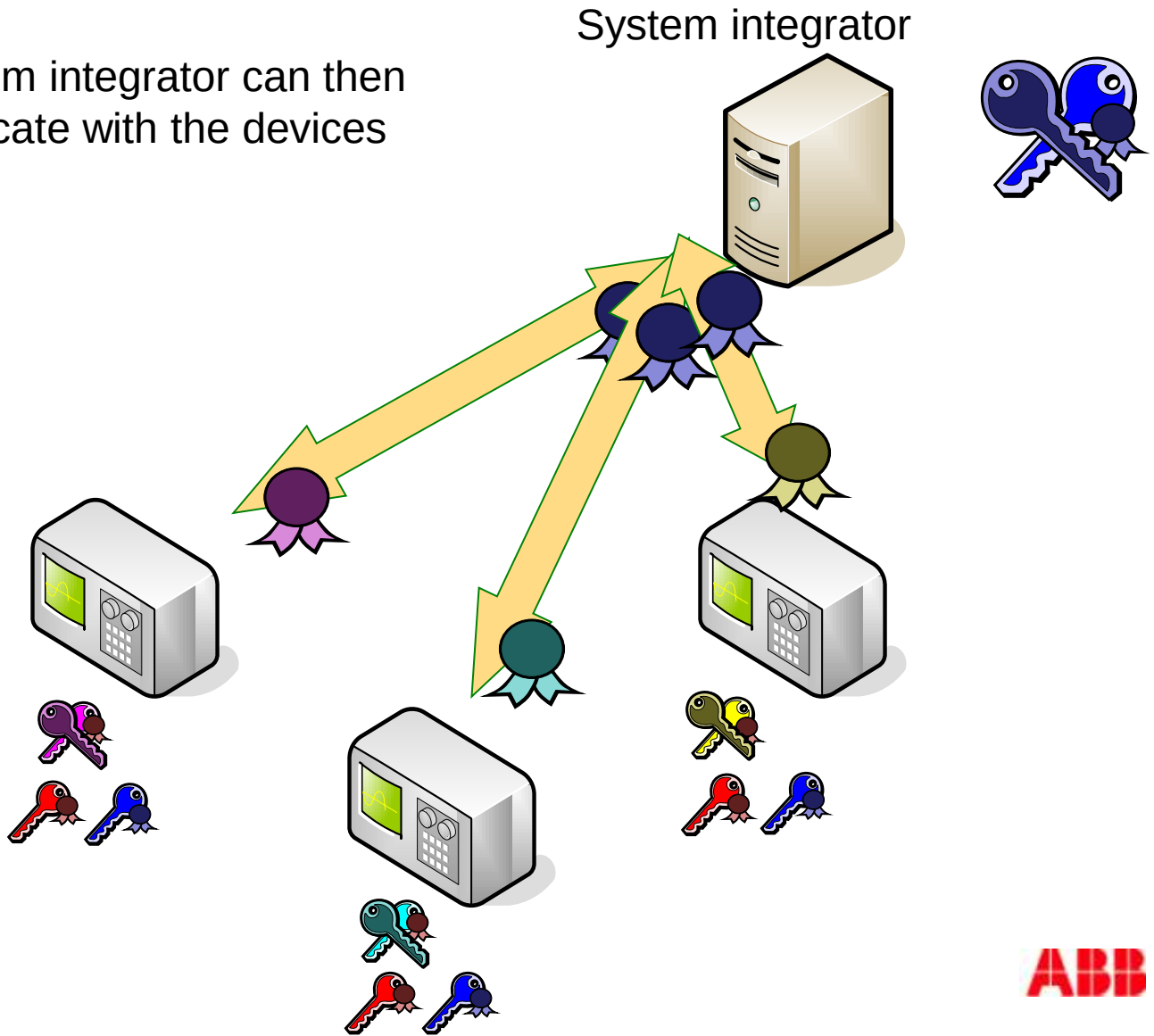
Customization

- The system integrator transmits the “Add Root” command to the devices
- That install the root certificate



Customization

- The system integrator can then communicate with the devices



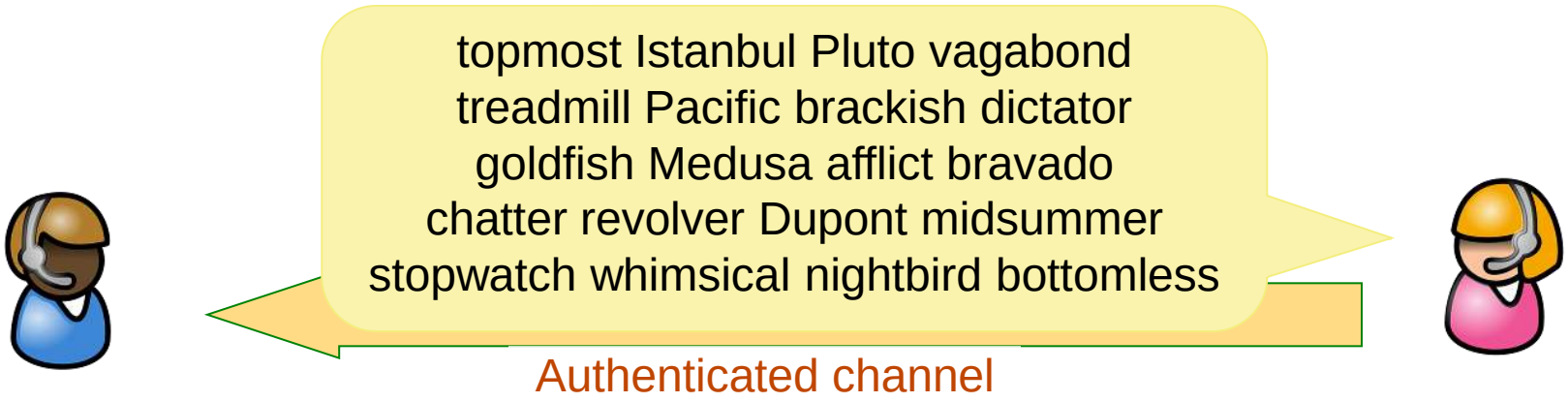
Getting an Authenticated Channel

An authenticated channel can be established:

- By using **trusted certificates** — **Not always possible**
- By having a **shared secret** in common — **Not desirable**
- By using an **out-of-band** mechanism:
  - Meeting face-to-face — **Delay/cost issues**
  - Regular mail — **Large delay**
  - Voice channel over the telephone — **Not practical?**
    - Transmit a cryptographic hash — **160 bits**
    - Use Short Authenticated Strings — **20 bits**

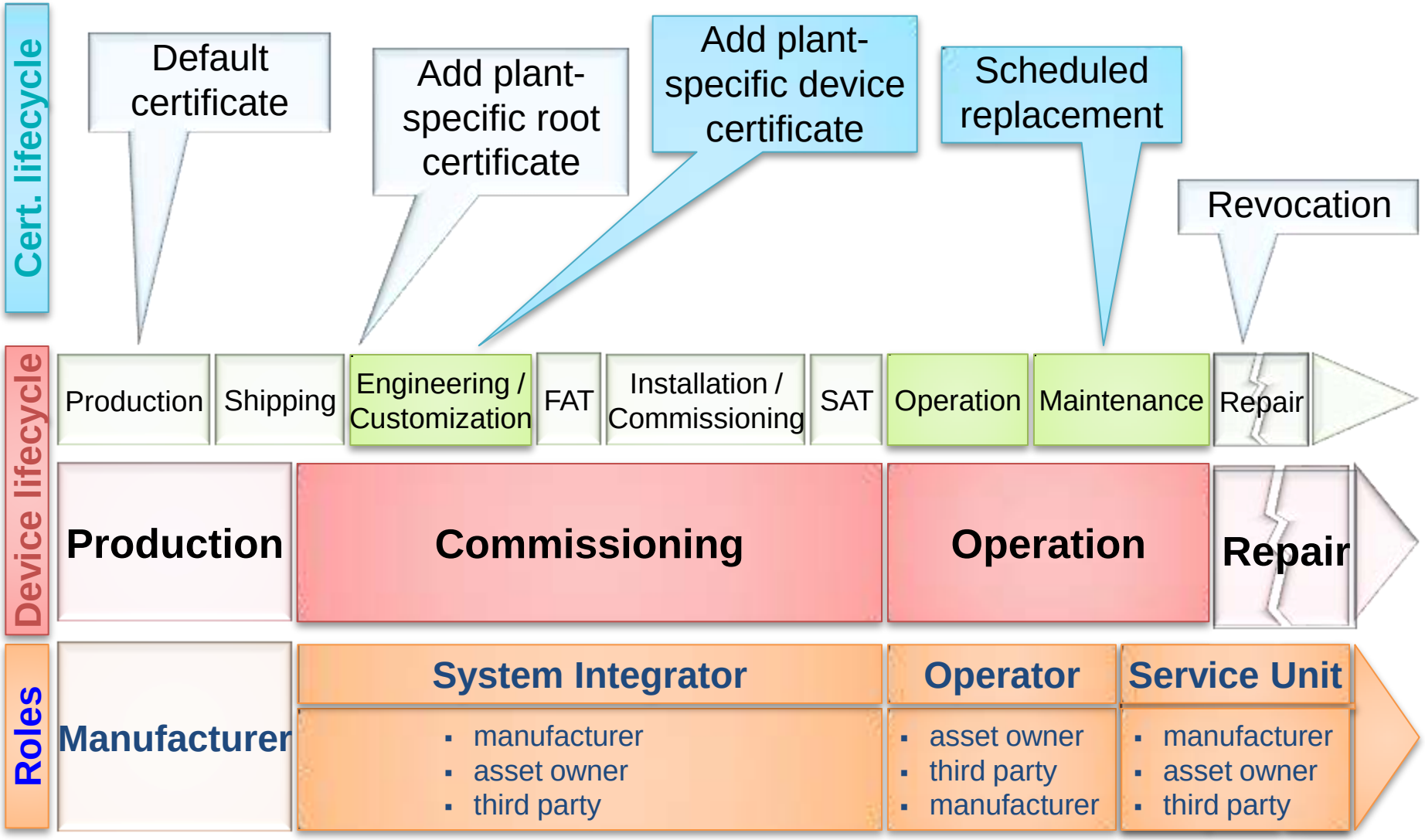
User Friendly Hash Transmission via Voice Channel

- Transmitting in hexadecimal notation is tedious and error prone:
  - ‘B’ and ‘D’ sound similar.
  - transposition, duplicate digits, omitted digits
- Can use the PGP word list instead:
  - phonetically distinct words
  - protection against transposition, duplicate and omitted words.



Certificate Replacement

Overview of Device & Certificate Lifecycle



Certificate Expiration

- Certificate expiry is a debated issue, may not be necessary for long key lengths
- Justifications for expiration:
  - Moore's Law
  - To mitigate advances in cryptanalysis
  - To reduce the time an attacker has to crack the key
  - To reduce the usefulness of compromised keys
  - Reduce the size of Certificate Revocation Lists
- However certificate replacement must:
  - Not compromise security
  - Maintain device availability

Key Generation Challenges

- RSA key generation is **computationally expensive** and requires a **probabilistic** algorithm

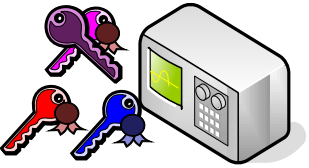
  → No problem for “office IT”, but for real-time systems

- Case 1: The device can handle it
  - Straightforward: use old certificates for authentication

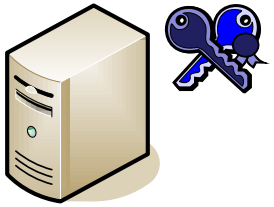
- Case 2: The device cannot handle it
  → The CA must generate the device’s key
  - Confidential channel required to transmit the private key
    - Cannot use old certificates for that (forward secrecy)

Case 1: The Device Handles the Key Generation

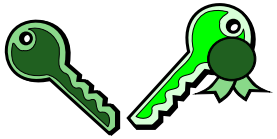
Device



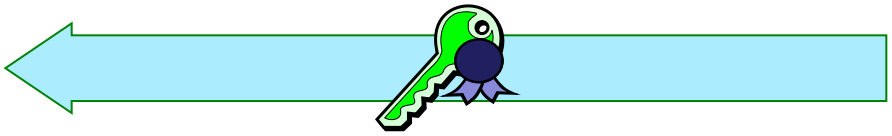
CA



Generate new key:
and certificate request:

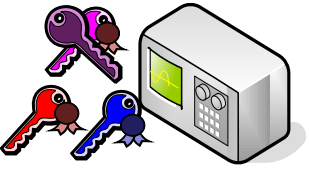


Generate certificate:

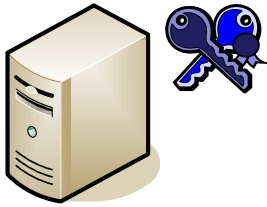


Case 2: The CA Handles the Key Generation

Device



CA

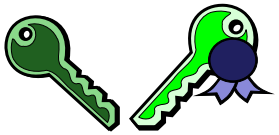


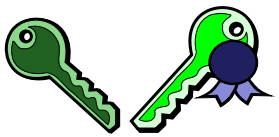
Replace key

TLS/SSL channel setup

Key agreement: Diffie-Hellman
Key authentication: RSA

Generate key & certificate:





Authenticated and encrypted channel (TLS/SSL)

Comparison

	Device	CA
Method 1	$O(n^4)$ time complexity prob.	$O(n^3)$ time complexity deterministic
Method 2	$O(n^3)$ time complexity det.	$O(n^4)$ time complexity probabilistic

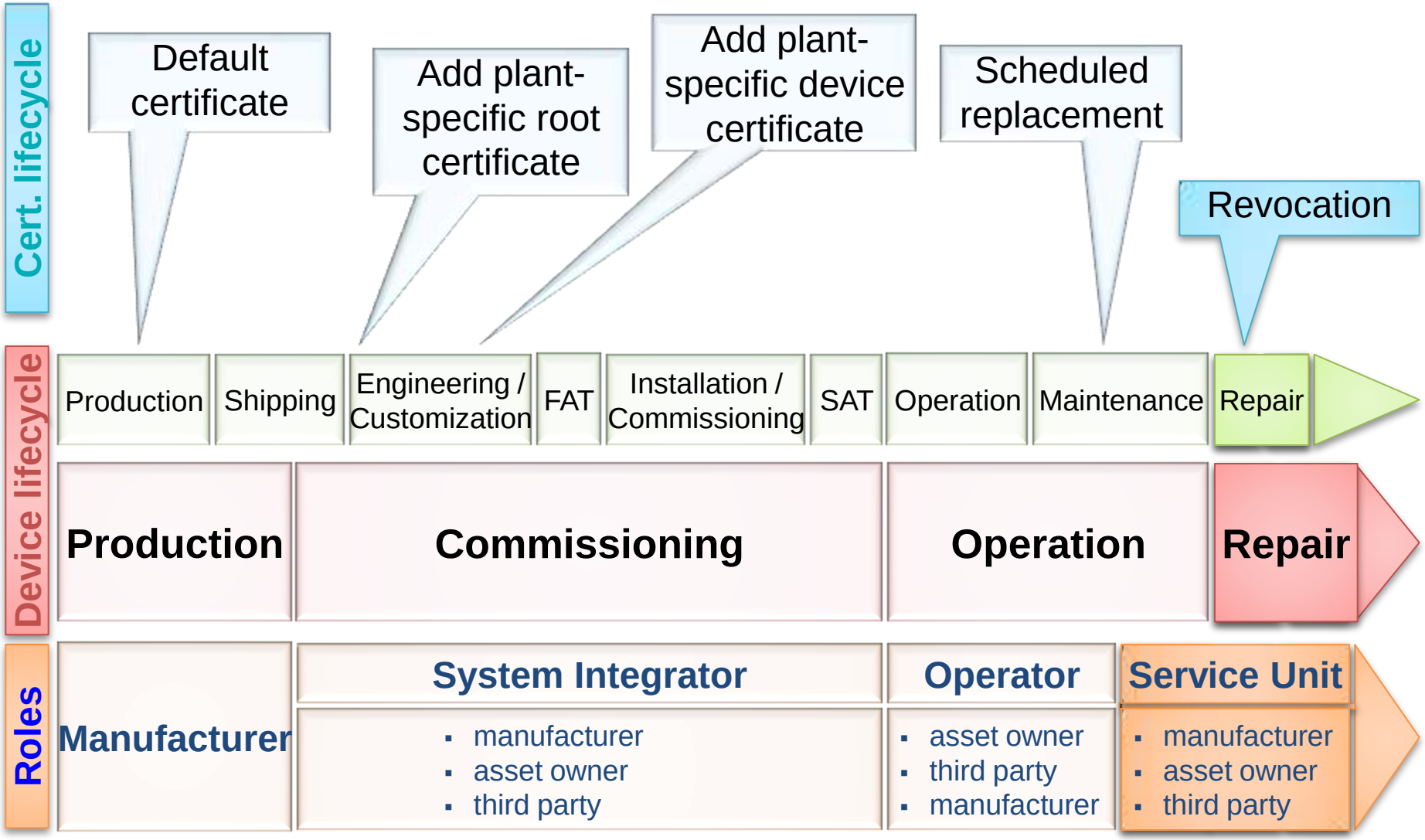
For reference:

RSA encryption / verification	$O(n^2)$ time complexity deterministic
RSA decryption / signature / modular exponentiation	$O(n^3)$ time complexity deterministic
RSA key generation	$O(n^4)$ time complexity probabilistic
DH parameter generation	$O(n^4)$ - $O(n^5)$ time complexity probabilistic

n = size in bits of the key

Certificate Revocation

Overview of Device & Certificate Lifecycle



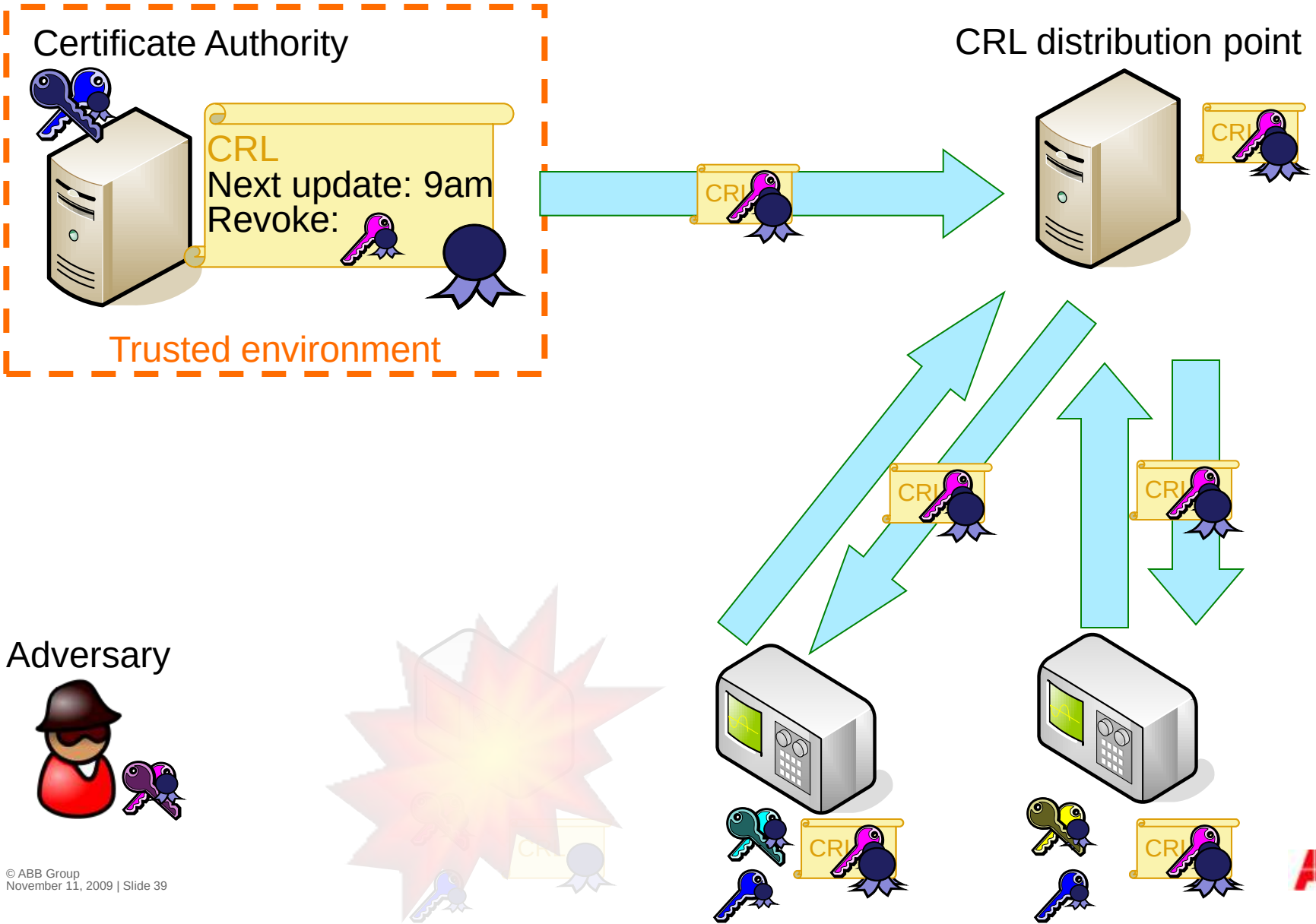
Revocation

- Invalidate a certificate before its expiration date
- Certificates need to be revoked:
  - If the information on the certificate no longer applies
  - If the subject is no longer trusted
  - If the private key of the device was compromised, or suspected to be
  - If the private key was lost
- Most expensive part of certificate management
- Requires the CA to update revocation information
- Freshness problems
  - Increasing the frequency of updates increases cost
- Vulnerable to denial of service attacks

Revocation Destroys Trust Relationships

- Revocation should be used only in **exceptional circumstances**
- A device without a valid certificate cannot authenticate itself anymore
- Trust relationship must be re-established
  - Can use the SAS protocol

Certificate Revocation Lists



Certificate Revocation Lists

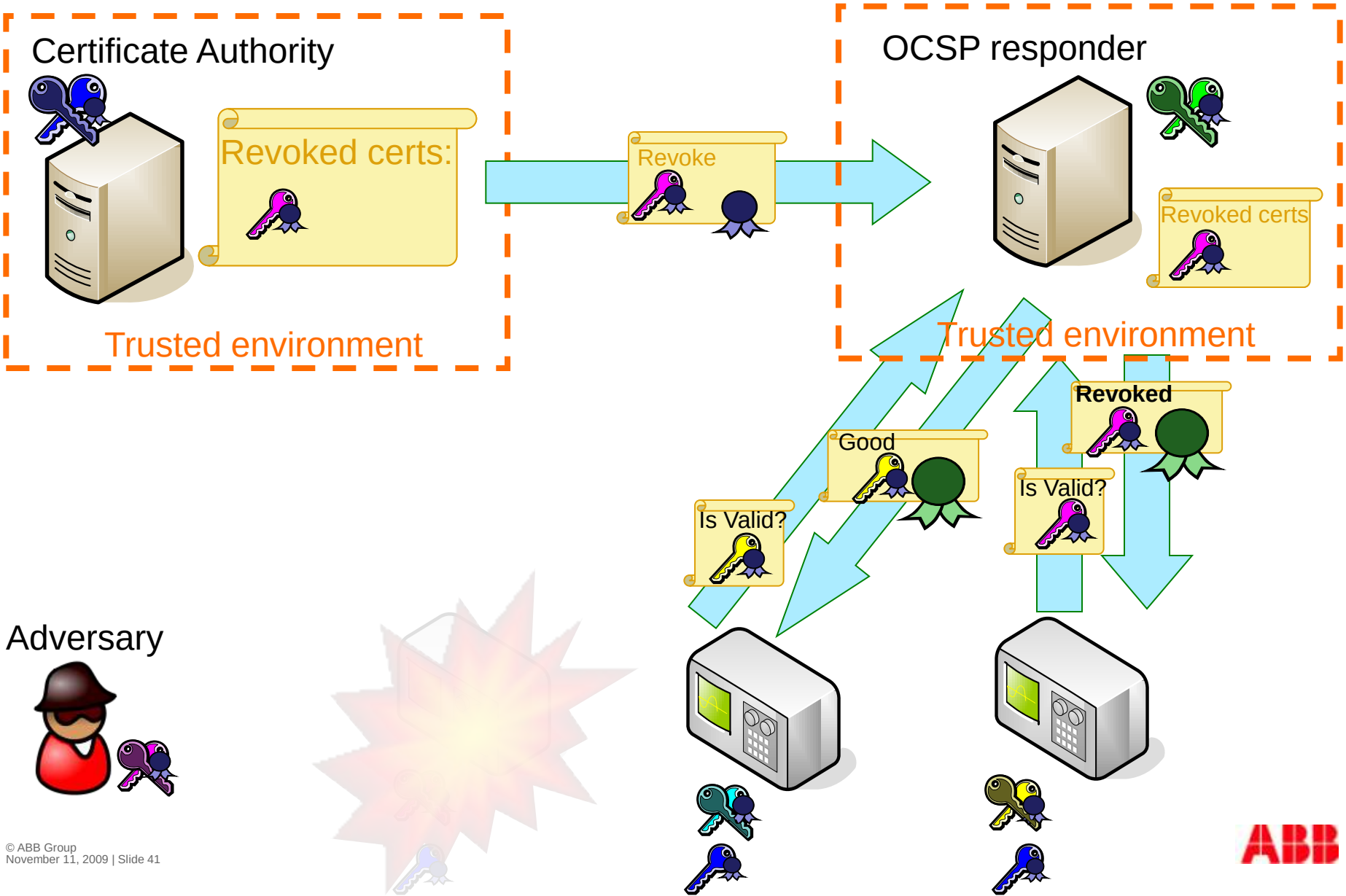
Each device must download the whole CRL at every update and store it in memory

- Cost proportional to number of revoked certificates
- Peak load on distribution point at each update

- Support not widespread
  - OpenSSL-0.9.8k has a couple of bugs with CRLs

- Seems practical

Online Certificate Status Protocol (OCSP)



Online Certificate Status Protocol (OCSP)

- Requires a permanent connection to the OCSP responder
- OCSP responder must be a trusted entity
- OCSP responder must do a lot of work
  - One RSA signature per request (alternative: ignore replay protection)
- Delayed certificate validation for devices:
  - One communication roundtrip + verify an RSA signature
- Not robust against denial of service

- Support even less widespread

Summary and Conclusion

- Use of certificates is crucial for authentication, integrity, confidentiality
- Certificate management for embedded devices is challenging
  - Affects the whole device's lifecycle
- Special requirements of embedded systems demand new solutions
  - Multiple parties involved
  - Example: partly shift work to the CA for generation of new keys
- Revocation is still an open research question
  - Replaceable by short-lived certificates?

Power and productivity
for a better world™

Additional Slides

Short Authenticating Strings (SAS)

Manufacturer

System integrator

m := message

RA := random (mod 10^6)

r := random 80 bits

c := SHA-256(m || RA || r)

Unsecured channel (TCP)

m || c

Cannot guess RA

Commit RA || r

Cannot change RA

RB := random (mod 10^6)

RB

RA || r

c' := SHA-256(m || RA || r)

abort if c' ≠ c

Open commitment

SAS' = RA+RB (mod 10^6)

SAS = RA+RB (mod 10^6)

14 15 92

SAS

abort if SAS' ≠ SAS

Authenticated channel (voice)