Secure the next generation of IoT devices

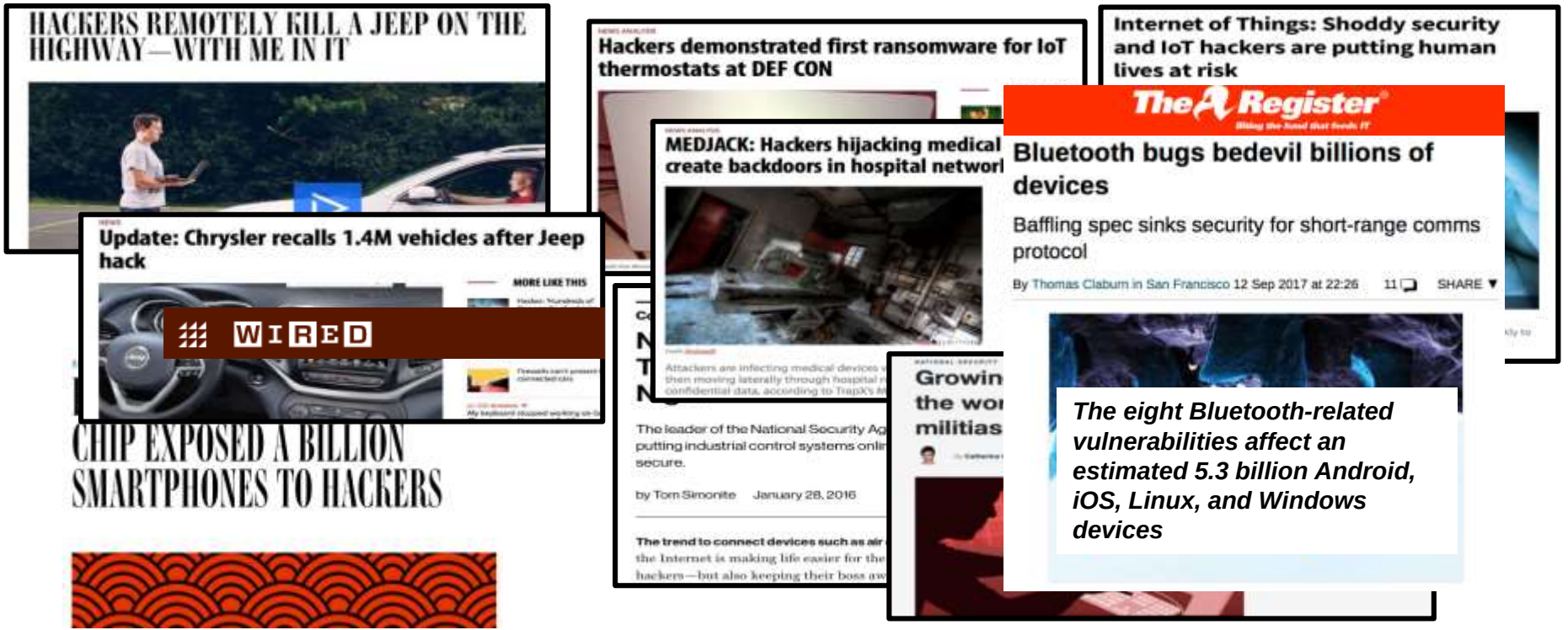
David Källberg, Field Application Engineer, IAR Systems

Agenda

- Security challenges for IoT devices
- Secure product development
- Integrating security into your workflow

Security challenges for IoT devices

Internet of Threats



And also…

IP theft

- Software is stolen in manufacturing and reengineered
- Software is compromised at user site and reengineered

Overproduction (“The night shift”)

Third-party manufacturers run unsolicited production batches

Lack of concerns lead to loss of revenue, potential brand damage, liabilities etc.

Practical issues

- How to prevent malware injection
- How to prevent unsolicited rollback of old version
- How to prevent users from using deprecated versions

Secure product development

Security from the ground-up

Security should be considered from the start of the design but many organizations...

- Do not design the application with security in mind.

  => Vulnerabilities can creep into the application.
- Focus on functionality and speed rather than security.
- Have strong requirements on shorter time to market.

  => Security is often an afterthought.

Prerequisites

Secure MCU: MCU with security features

- Protected memory areas for e.g. key and certificate store, crypto accelerators etc.

Secure Boot Manager (SBM)

- Can decrypt images, check validity of images, program decrypted image to flash, download updates etc.
- Can interact with “secure programming” machines to maintain image integrity across supply chain
- Configurable to accommodate different needs and tradeoffs

Software mastering tool (Manufacturing image preparation)

- Security-enabled programming machines (i.e Data I/O)

You also need…

Root of Trust (RoT)
“The minimal set of software, hardware and data that is implicitly trusted in the platform ...”

Secure worlds: Certificates, keys, Secure Boot Loader configuration, versioning policy

- Enable creation of secure worlds, including certificates
- Enable easy selection of reconfigured secure worlds for different use cases

What if you were able to…?

… Integrate security into the existing workflow of development, debug, mastering and manufacturing

… Simplify certificate and key infrastructure implementation

… Enable easy handover from development to manufacturing

… Assist implementation of versioning and rollback policies

… Create scalable and robust secure products and solutions

Integrating security into your workflow

Embedded Trust

Streamlined security development in IAR Embedded Workbench

- ✓ Security Development Environment
- ✓ Integrated identity and certificate management
- ✓ Scalable Secure Boot Manager
- ✓ Secure deployment with integrated manufacturing mastering
- ✓ Release management with versioning and update infrastructure

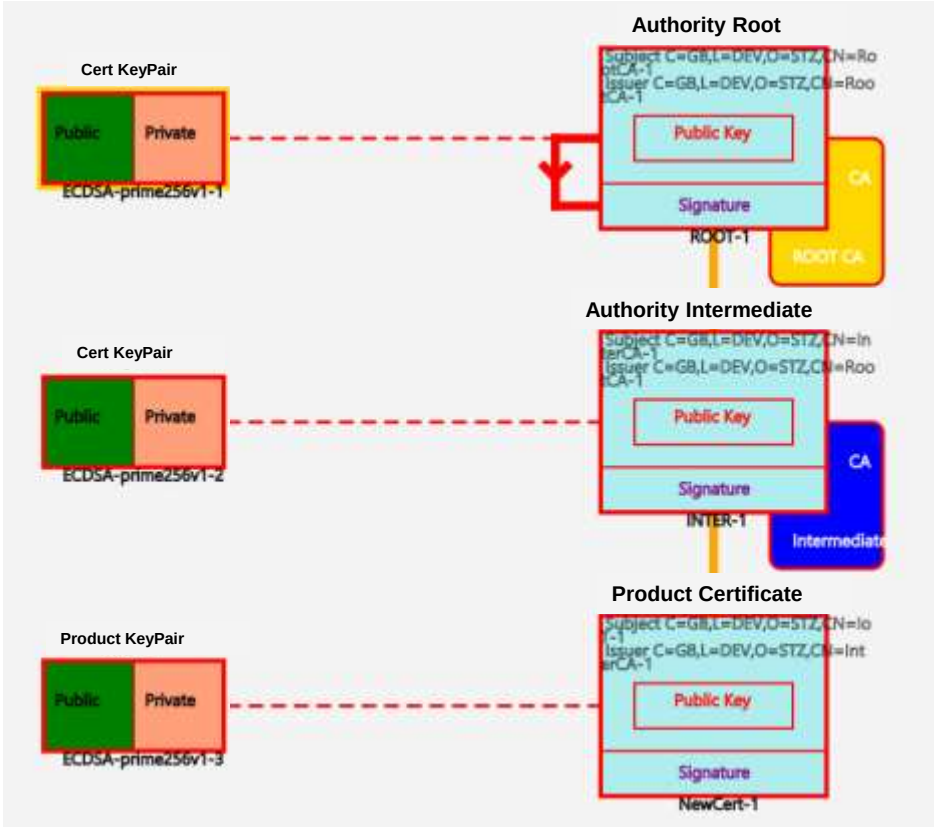
Embedded Trust

- Implements a secure bootloader
  - Ensures that only authorized code makes it to your device
  - Protects your customers from dangerous “insertions” in your code like keyloggers
- Protects your Intellectual Property (IP) by disallowing unauthorized production

Embedded Trust workflow

1. Define product RoT keys and certificates

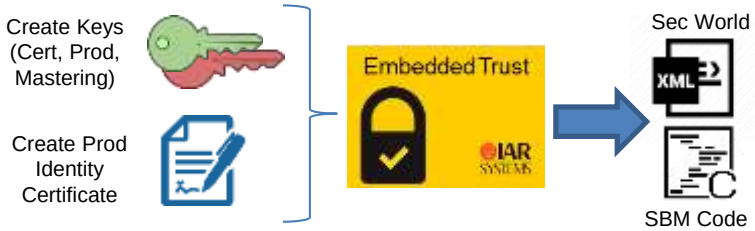
Keys and certificates



- Define the product certificate including the supporting chains
- Define the cryptographic product keys and certificate keys
- Specify the key and certificate parameters
- The definition of these items form the “Secure World” context that is configured into the SBM

Embedded Trust Workflow

1. Define product RoT keys and certificates
2. Configure Secure Boot Manager

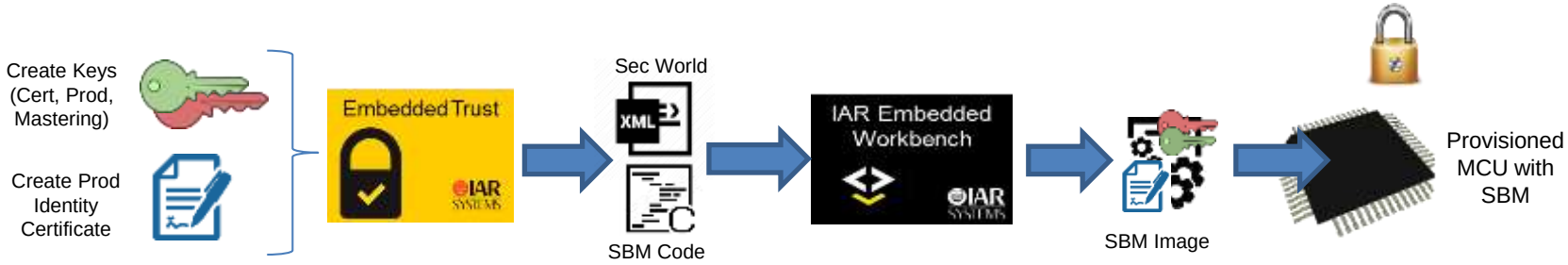


Secure Boot Manager

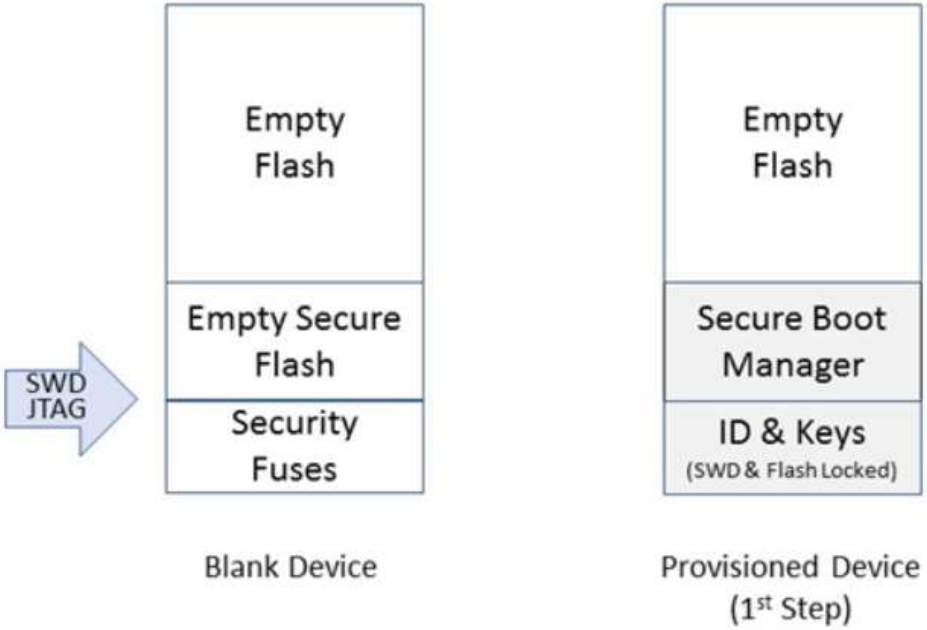
- OEM configurable SBM source code
- Integrates the Secure World context
- Only signed and encrypted code accepted
- Supports versioning and anti-rollback
- Supports modular updates
- API for SBM management functions and to leverage the RoT certificates and keys

Embedded Trust Workflow

1. Define product RoT keys and certificates
2. Configure Secure Boot Manager
3. Build the SBM and program MCU

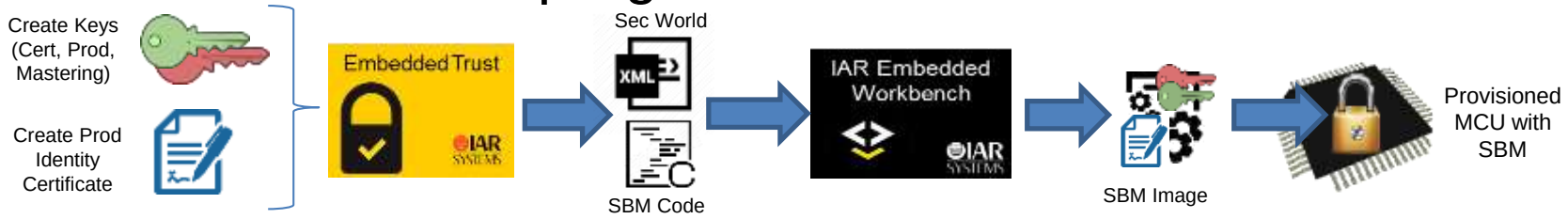


Provisioning the device

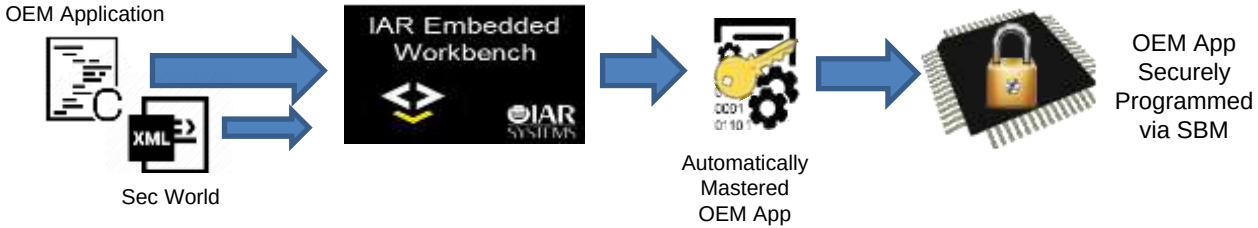


Embedded Trust workflow

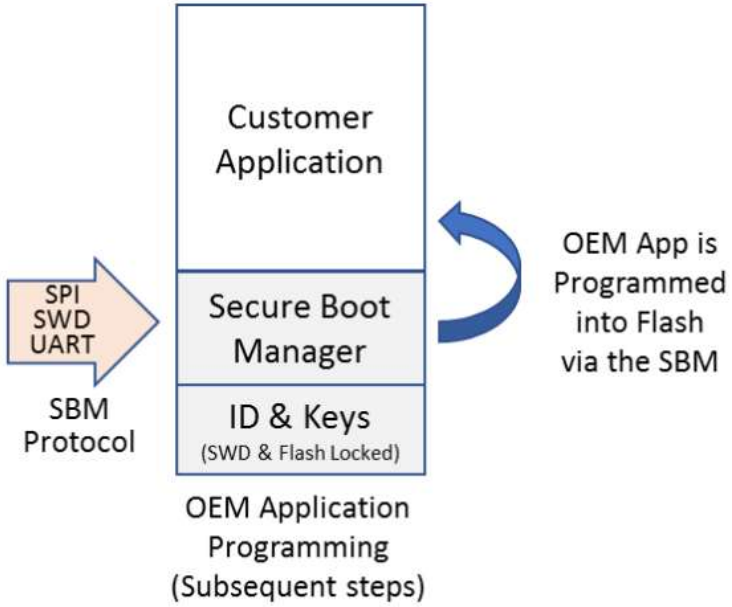
1. Define product RoT keys and certificates
2. Configure Secure Boot Manager
3. Build the SBM and program MCU



4. Customer's application SW is automatically mastered



Programming the application



Running the application

- Each time the device resets, the SBM checks the hash of the application
  - If the hash doesn’t match the one in protected memory, the application is not allowed to run
  - Prevents app tampering
- If the hash matches, the app will boot

Going to production

Enable development, debug, mastering, provisioning and manufacturing in one unified workflow

IAR SYSTEMS

SECURE THINGZ

Data I/O

Secure code checklist

- ✓ Evaluate security needs
- ✓ Partition firmware – isolate critical functions
- ✓ Regular security reviews
- ✓ Only reuse security reviewed code
- ✓ Use well-tested commercial components
- ✓ Implement secure life cycle management
- ✓ Define secure world from start

Summary

- Security is no longer optional.
- You need to establish trust as part of the development process.
- Embedded Trust integrates security into your workflow.

Want to learn more?

Visit our stand to get a demo of Embedded Trust.

Go to iar.com

Thank you for your attention!