z

BitSec: A secure microkernel for deeply embedded systems

Jim Huang (黃敬群)

National Cheng Kung University, Taiwan

Dec 17, 2016 / OS2ATC 2016

NCKU National Cheng Kung University

2015 年 7 月，安全研究人员 Charlie Miller 和 Chris Valasek 永远地改变了汽车行业“车辆安全”的概念。他们展示了黑客能够远程攻击一辆 2014 款 Jeep Cherokee，禁用其变速器和刹车。这一发现导致菲亚特克莱斯勒前所未有地召回 140 万车辆

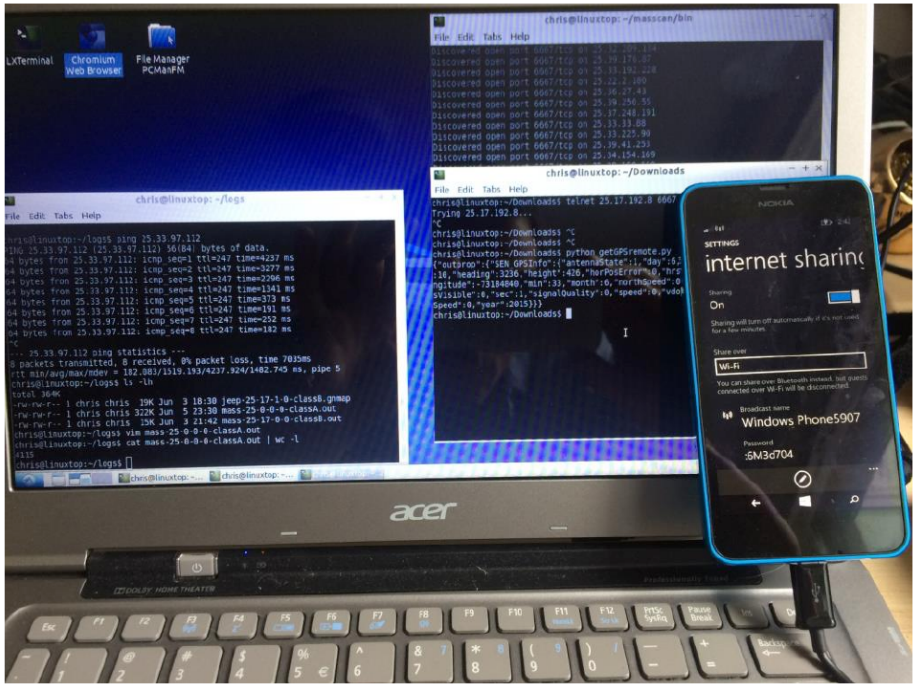
Source:
http://www.leiphone.com/news/201512/DEGhPfKRnyRxaGmS.html

July 2015: Miller and Valasek takedown of Jeep

source: http://illmatics.com/Remote%20Car%20Hacking.pdf

D-Bus service responding to an open 3G port

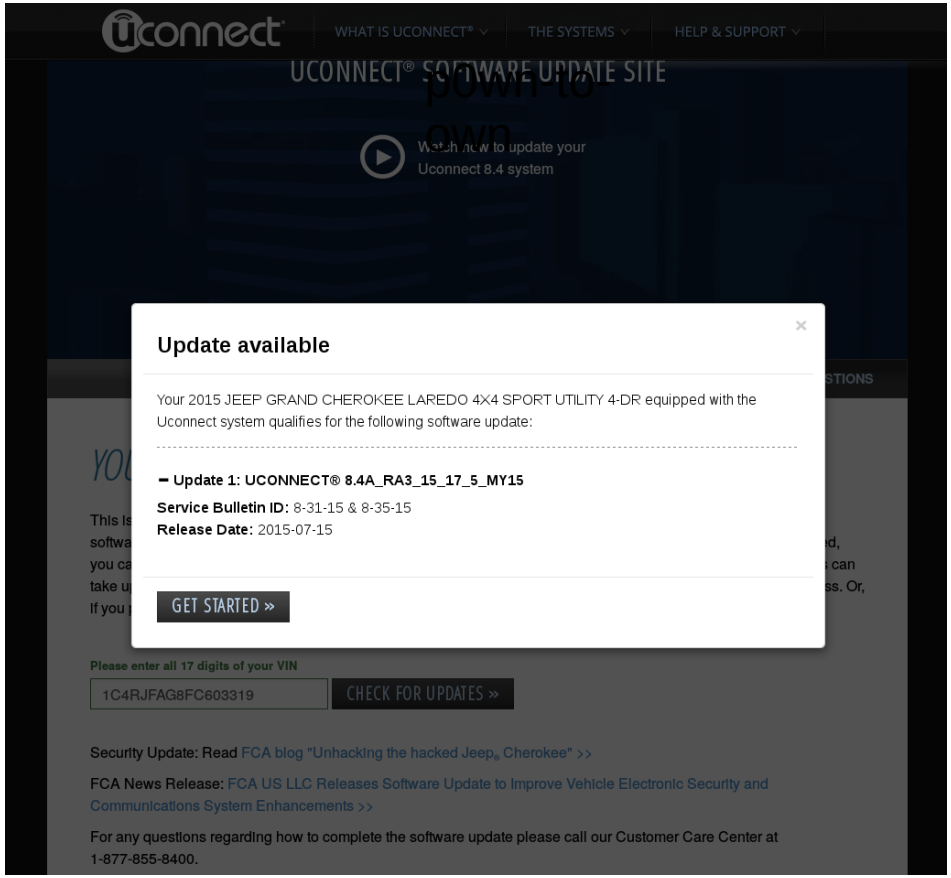
“To find vulnerable vehicles you just need to scan on port 6667 from a Sprint device. . . “




```
# netstat
Active Internet connections
Proto Recv-Q Send-Q  Local Address           Forei
tcp        0      0  144-103-28-21.po.65531 68.28
tcp        0     27  144-103-28-21.po.65532 68.28
tcp        0      0  *.6010                 *.*
tcp        0      0  *.2011                 *.*
tcp        0      0  *.6020                 *.*
tcp        0      0  *.2021                 *.*
tcp        0      0  localhost.3128         *.*
tcp        0      0  *.51500                *.*
tcp        0      0  *.65200                *.*
tcp        0      0  localhost.4400         localhost.65533
ESTABLISHED
tcp        0      0  localhost.65533        localhost.4400
ESTABLISHED
tcp        0      0  *.4400                 *.*                    LISTEN
tcp        0      0  *.irc                  *.*                    LISTEN
```

National Cheng Kung University

Without Over-the-Air Updates, Jeep is stuck

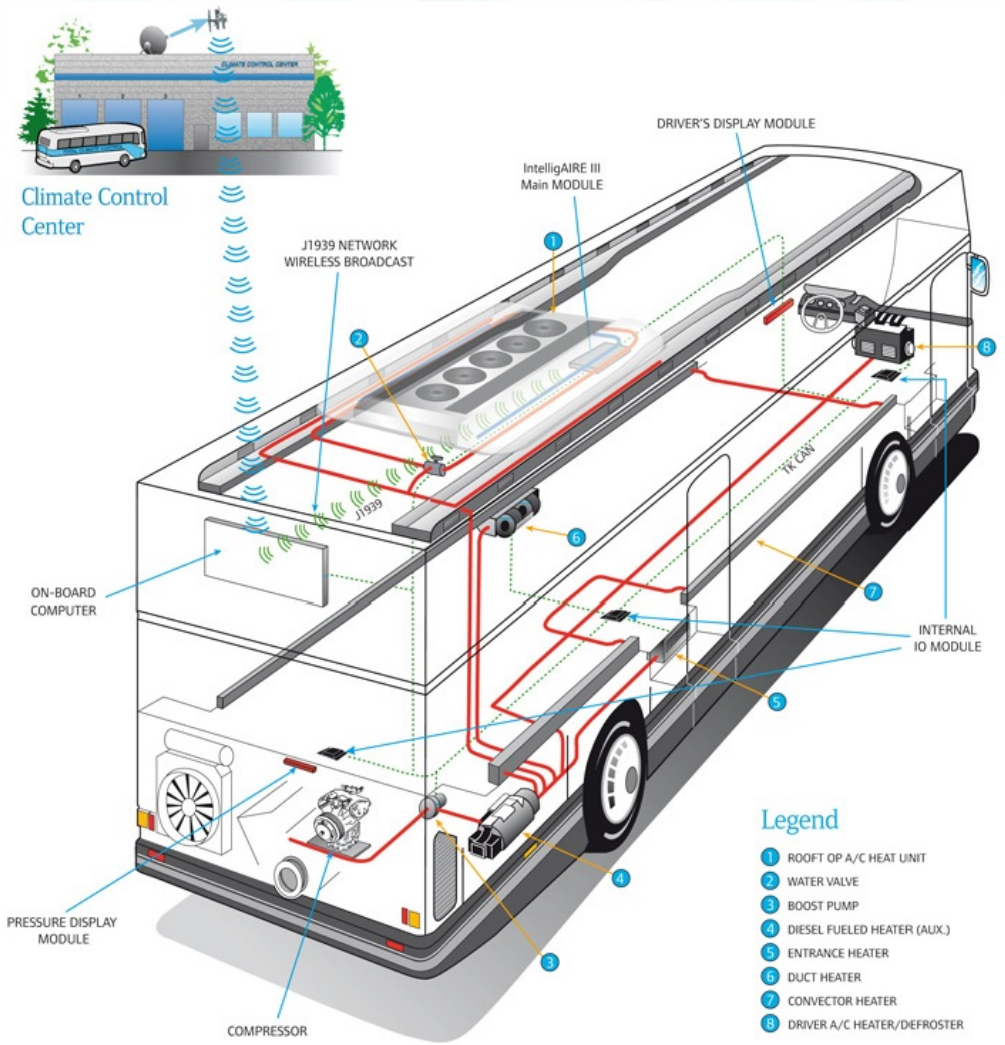


Dec. 2015 view of Uconnect update

National Cheng Kung University

Connectivity may be a bad choice

"Shuttle bus withJ1939 air conditioning,"
Metropolitan Atlanta
Rapid Transit Authority,
http://can-newsletter.org

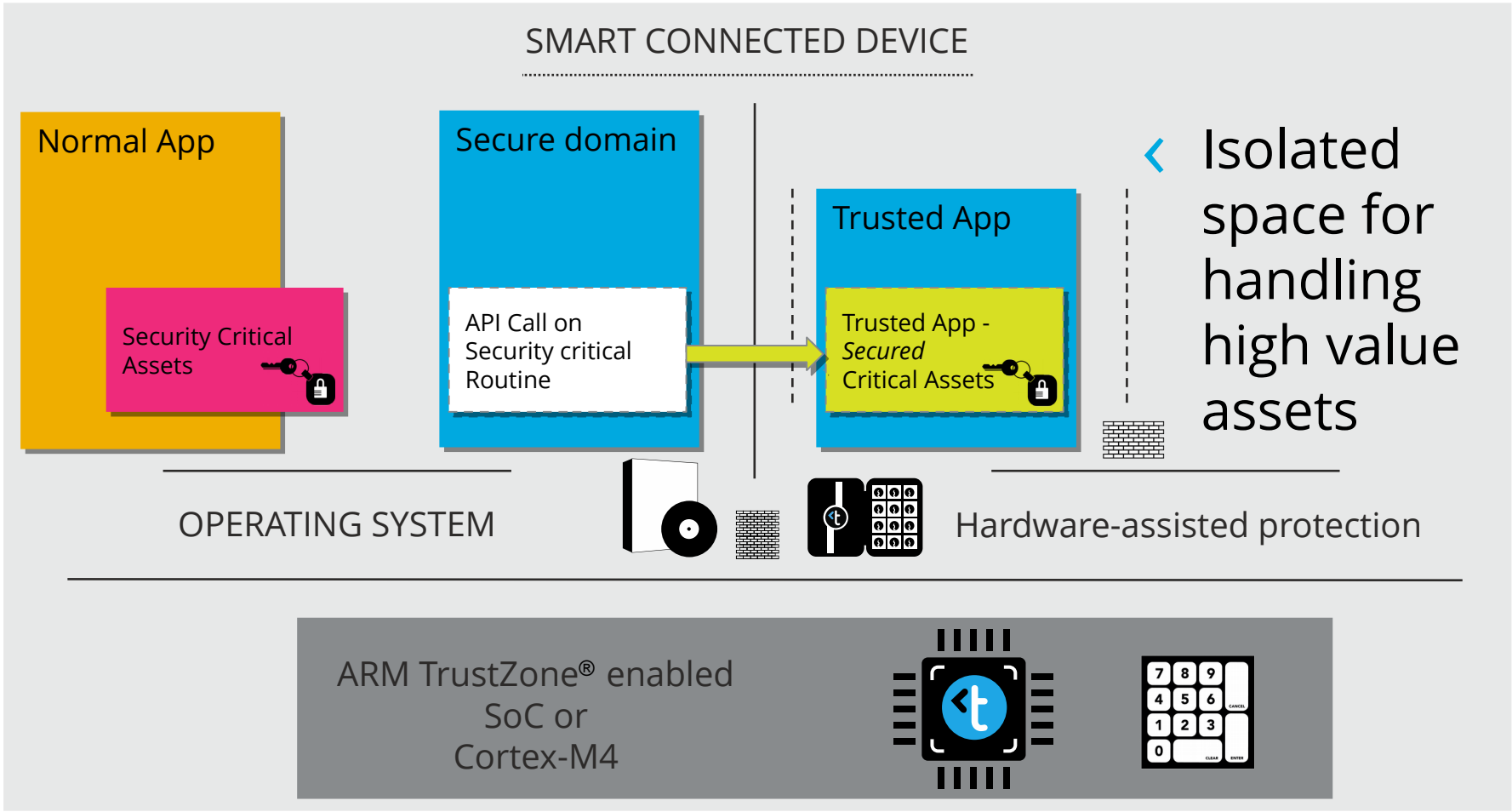


- The "Thermo King Intelligaire III"

National Cheng Kung University

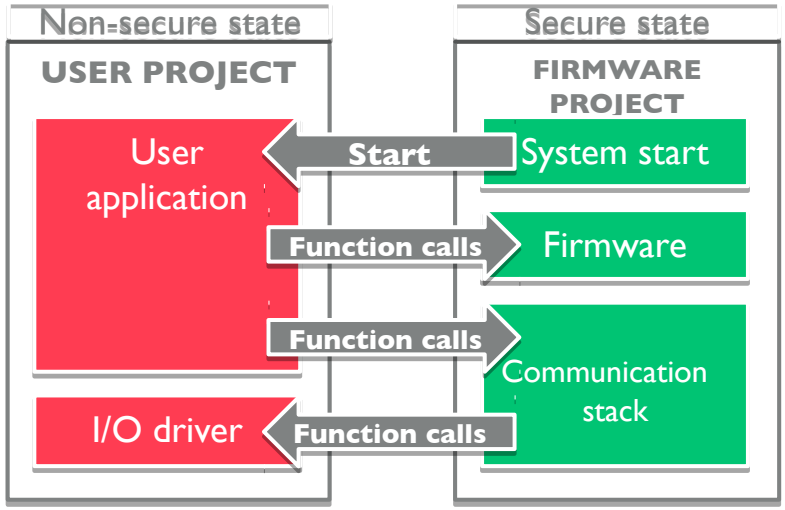
BitSec: secure microkernel / hypervisor

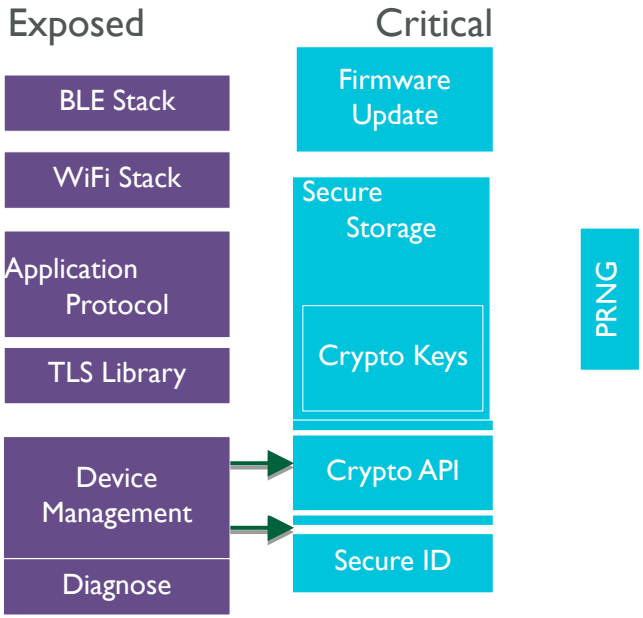
- Key assets exposed
- Key assets protected



Background of BitSec

- Learnt from uVisor, part of ARM mbed
  - Hardware-enforced security sandboxes
  - “Princle of Least Privilege”
  - Boxes are protected against each other and malicious code is contained
  - Per-box access control lists (ACL)
  - Restrict access to selected peripherals
  - Shared memories for box-box communication
- but, BitSec is lightweight and faster
- Apache License 2.0

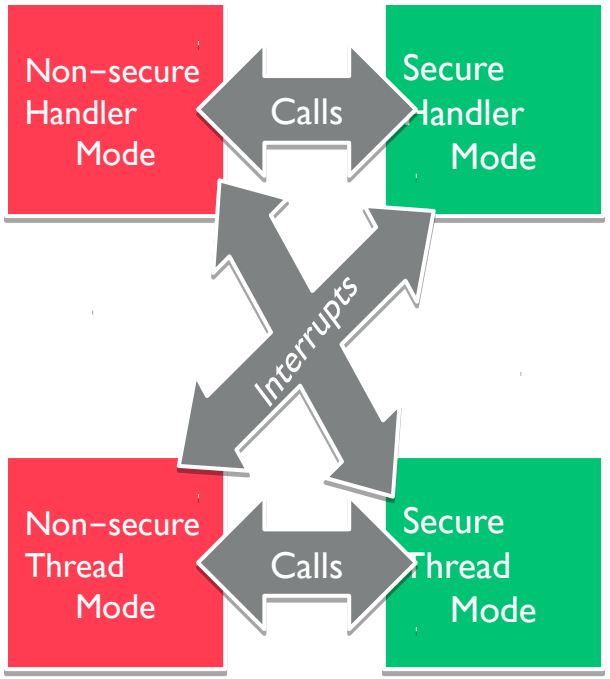




National Cheng Kung University

Cross-domain call in ARMv8-M

- Security inferred from instruction address
  - Secure memory considered to hold Secure code.
- Direct function calls across boundary
  - High performance and high security
  - Multiple entry points
  - No need to go via "monitor" for transitions.
- Uses Secure Gateway instruction "SG"
  - Only permitted in special Secure memory with Non-secure-callable attribute (NSC).



Properties of BitSec

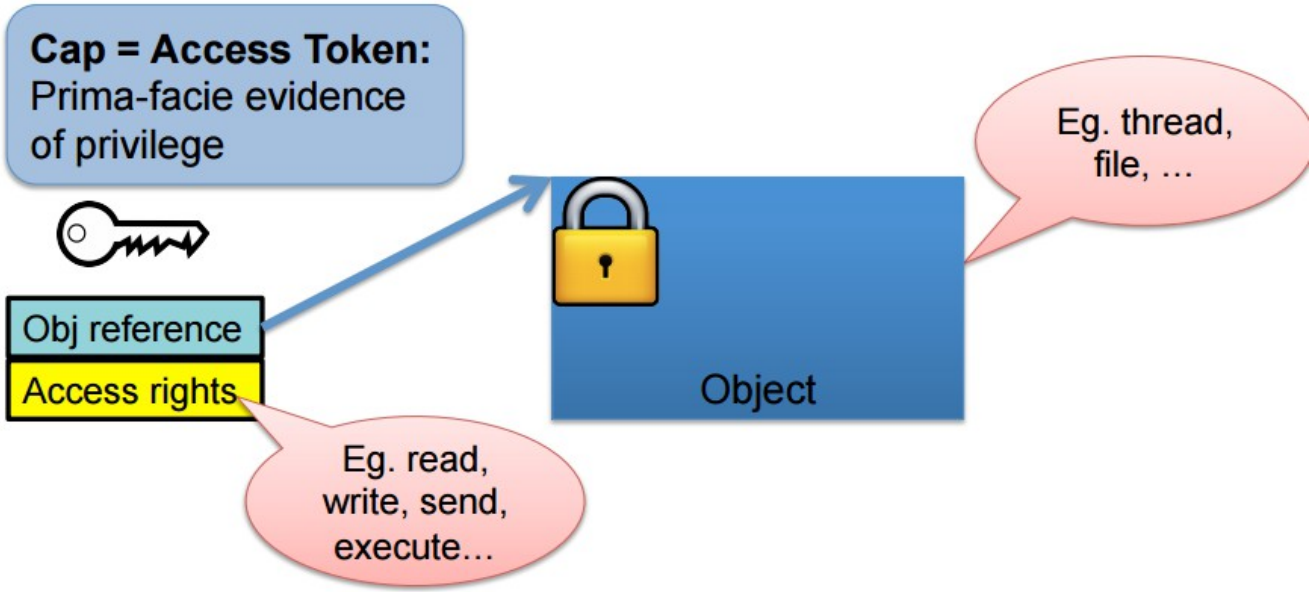
- ARMv7-M friendly: efficient application isolation
  - designed to use the ARMv7-M MPU for isolation
  - Ready for ARMv8-M TrustZone enablement
- third-generation microkernel
  - heavily inspired by seL4
- Focuses on minimality and security,
- Expresses all authority through explicit capabilities,
- Moves other mechanisms with security implications outside the kernel,
- explicitly targets systems with between 16 and 200 kiB of RAM. 2K LoC

Basic Concepts

- Object-oriented
  - object bundling together state and operations
- Capability-oriented
  - use of a capability, or key
  - object reference and a set of rights
- Messaging-oriented
  - single efficient message-transfer operation called IPC
  - operate on kernel objects
  - communicate between application tasks.

Capabilities

- without holding additional authority, programs can only perform three operations on a key
  - Copy the key into a different key register
  - Send a message to the object designated by the key
  - Receive a message from the object designated by the key

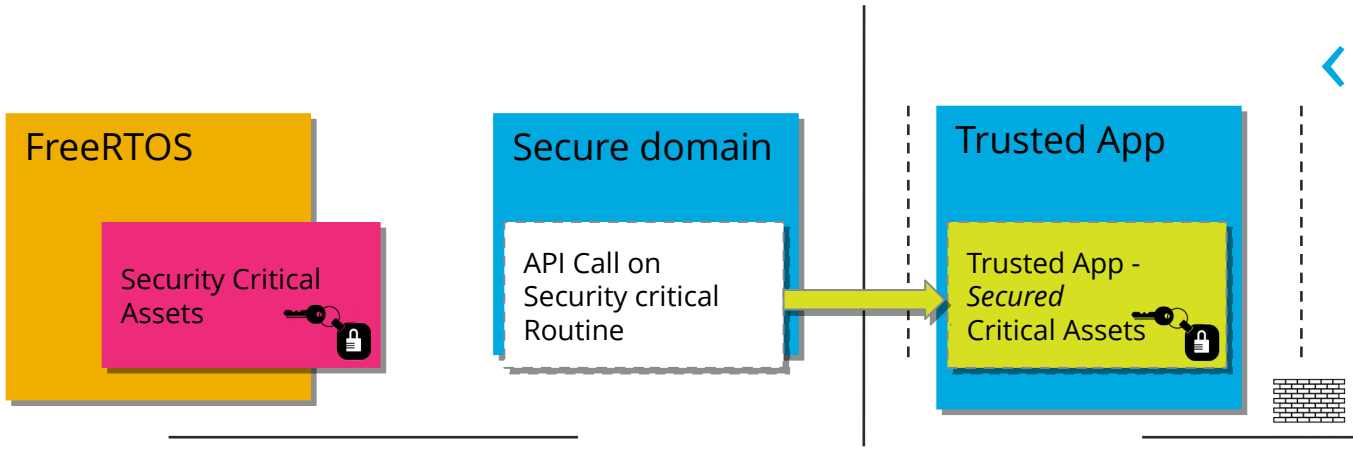


System Calls

- similar design as seL4
  - send, receive, yield
- IPC
  - synchronous rendezvous messaging model
  - messages are sent from one object to another directly
  - without being buffered in the kernel
- Copy key
  - Reads a key from one of current Context's Key Registers
  - Writes a duplicate of it into another

Case Study: FreeRTOS Integration

- context switch latency between FreeRTOS tasks: 2x overhead
- FreeRTOS on BitSec gains several features that are missing from the ARM_CM3 port
  - memory-protected environment
  - Ability to run entirely in unprivileged code
  - run a hybrid system
    - FreeRTOS drivers + (trusted) native BitSec drivers



- Isolated space for handling high value assets

Case Study: FreeRTOS Integration

- not a FreeRTOS API emulation layer or simulator.
  - actual FreeRTOS code, derived from the ARM_CM3 port
  - including the scheduler
- FreeRTOS System layer implements:
  - Allocation and deletion of OS objects (task/queue/heap)
  - Mutexes with priority inheritance
  - operation timeouts and time-slicing with preemption.
- Two contexts in FreeRTOS/BitSec
  - Task context
    - model Thread execution code; used to run FreeRTOS
  - Interrupt context
    - model Handler execution code such as ISR
    - implement some virtual interrupts

Virtual interrupts for guest OS

- Messages Model Supervisor Calls
  - Task and Interrupt Contexts share access to a Gate
    - called the System Gate (SG)
  - FreeRTOS sends BitSec IPC messages through SG
    - Requesting a context switch
    - Enabling/disabling interrupts
  - Interrupt context holds Service Key to task context
- Context Switches Multiplex the Task Context
- Message Dispatch Loop Multiplexes the Interrupt Context