RTEMS: Una Auditoría Inconclu. . .

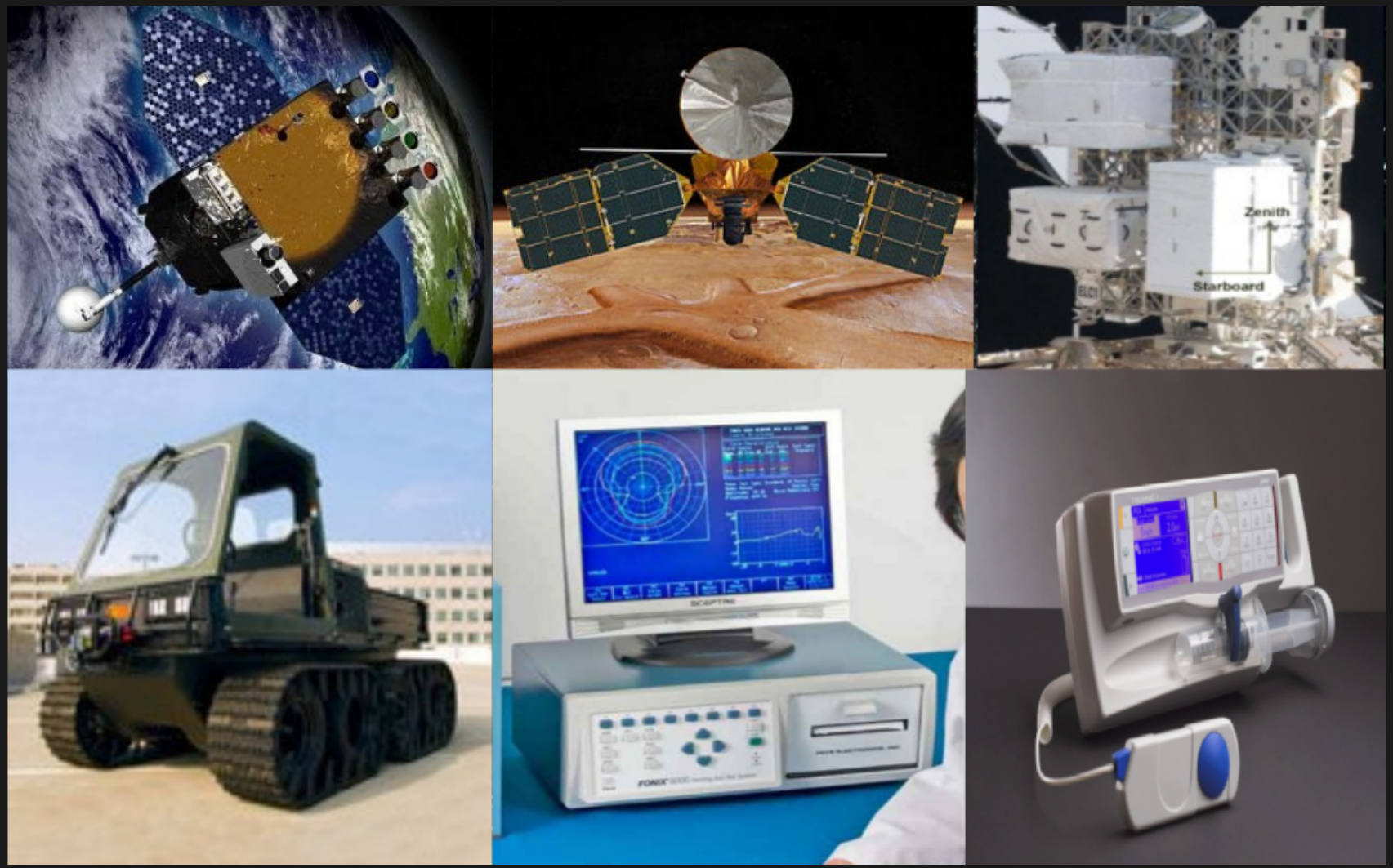
Lucas Molas & Christian Heitman
Fundación Sadosky

Security Jam Sessions
2015

R qué?

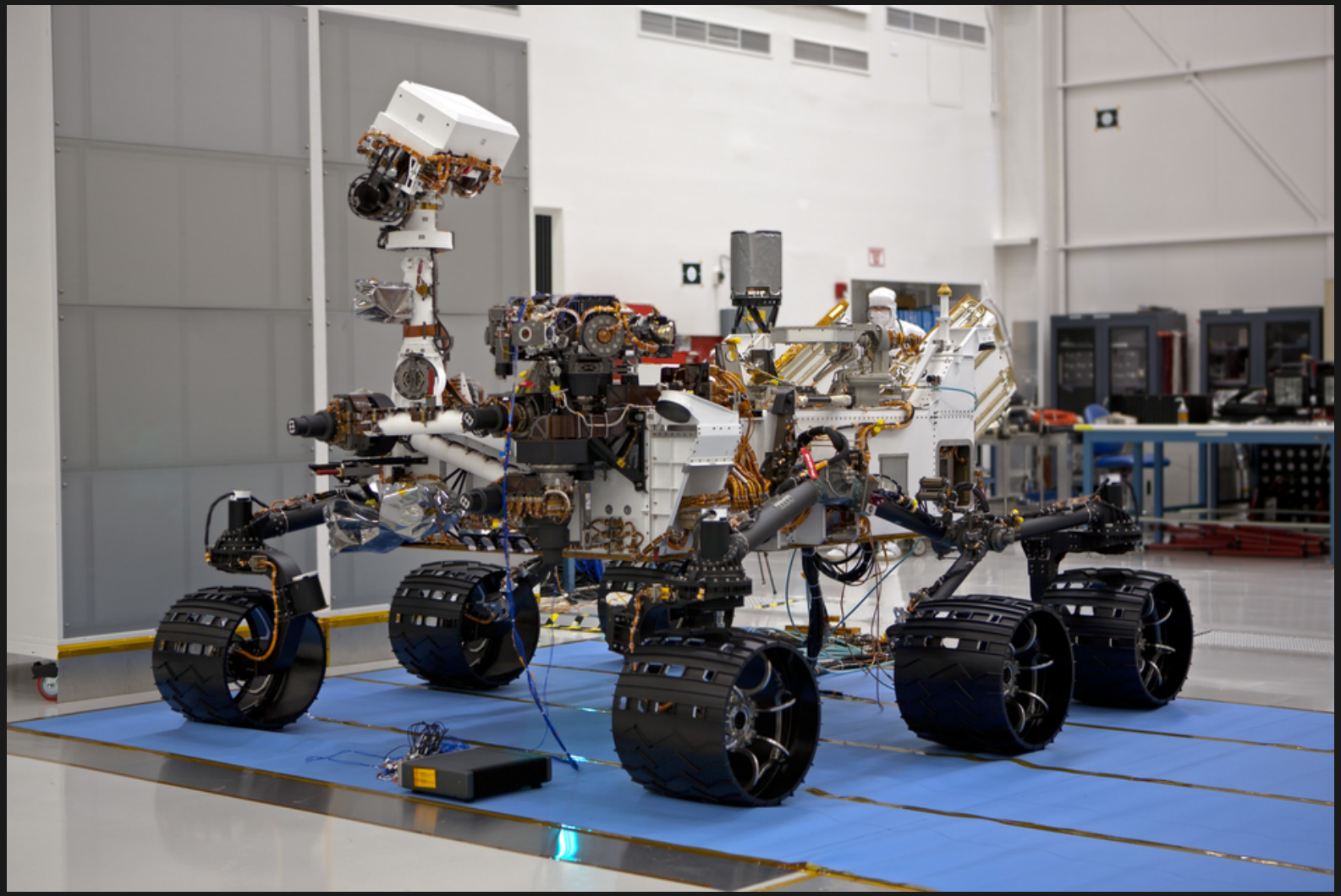
- ¿Qué es?
  - Sistema operativo de **tiempo real**
  - *Open Source*
- ¿Quién lo usa?
  - NASA
  - ESA
  - MITRE
  - JPL
  - ...
- ¿Dónde se usa?
  - Magnetospheric Multiscale Mission (NASA)
  - Curiosity (?)
  - Mitre Centaur
  - ...

¿Dónde se usa?



RTEMS Ejemplos

¿Dónde se usa?



NASA Curiosity

¿Dónde se usa?

MITRE Centaur

Una Auditoría Inconclu...

- Empezó como:
  - Proyecto de vinculación tecnológica Universidad-Empresa
  - Con el objetivo de desarrollar capacidades, conocimiento y tecnología para hacer una auditoria de seguridad de RTEMS
  - Ibamos a actuar como intermediarios/consultores/investigadores
  - Ibamos a contar con equipo donde corre RTEMS, utilizado para proyectos comerciales de la industria aeroespacial
- El proyecto nunca se concretó
- Terminó como:
  - Un “manual de referencia”
  - Reune notas sobre nuestra experiencia con RTEMS
  - Pueden encontrarla en GitHub

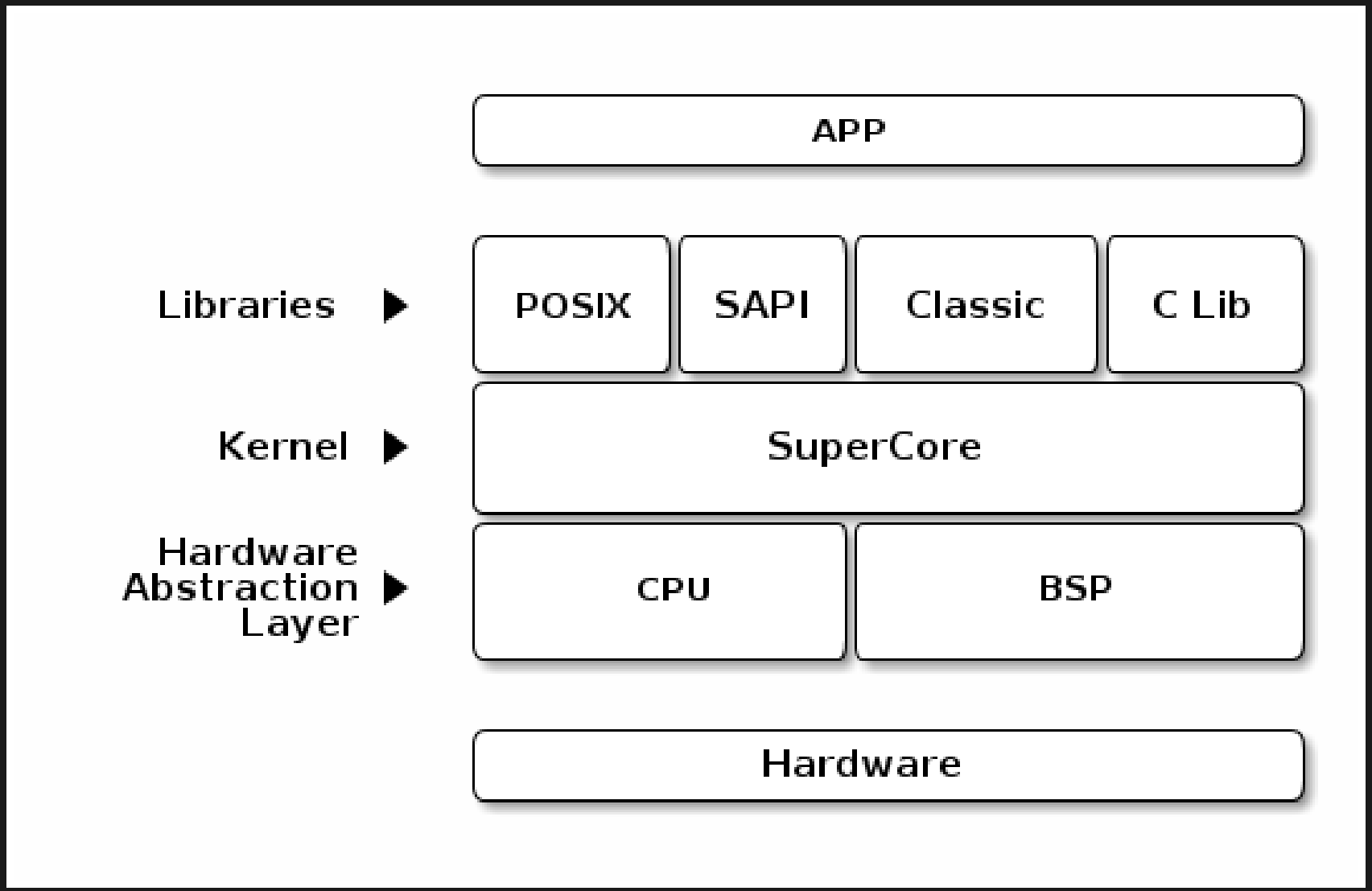
RTEMS

- RTEMS
  - **R**eal-**T**ime **E**xecutive for **M**ultiprocessor **S**ystems
  - Inicialmente la **M** era de *Military* (uy, qué mieditooo!)
- Arquitecturas
  - x86, ARM, PowerPC, MIPS, SPARC, ...
- Boards
  - Beagleboard, NXP LPC, PC Compatible for i386, Motorola MVME, LEON3, ...
- Es decir, soporta mucho *hardware out-of-the-box*

Todo Tiene un PERO...

- Muy poca documentación oficial
- Falta de *roadmap* (¿*Release* acutal? ¿Nueva versión?)
- La ayuda/consultoría oficial puede costar varios US$
- La empresa cobra por modificaciones particulares que luego se pueden incorporar o no al *master*

RTEMS



Estructura de RTEMS

Ejecutando RTEMS

- Instalación del ambiente de desarrollo
- Emulación RTEMS
- RTEMS sobre RaspberryPi
  - Soporta solamente la consola (UART)
  - No soporta SD *card*, USB, *Ethernet*, nada de lo interesante
  - JTAG *debugging*

Aspectos de Seguridad

- Manejo de procesos
  - *POSIX Profile 52*: *single process*, *multiple threads*
  - Todo corre en el máximo privilegio, cualquier vulnerabilidad es crítica
  - No hay *syscalls*, no es la clara separación entre el *kernel* y el *userspace*

Aspectos de Seguridad

- Modelo de memoria
  - *Flat Memory Model*: muy poco soporte de MMU
  - Implementa su propio `malloc` (`dlmalloc` básico)
  - No hay mecanismos de protección de *stack* ni *heap*

Aspectos de Seguridad

- Ejemplos de *payloads*
  - *Payloads* básicos para probar la explotación en RTEMS
  - Se simuló un *buffer overflow* y un ARM EVT *corruption*
  - *Payloads* de dos etapas

En Resumen...

- RTEMS se usa en proyectos sensibles: *drones*, satelites, vehículos terrestres, dispositivos médicos...
- No tiene en cuenta aspectos de seguridad ni a nivel *Kernel* ni a nivel aplicación

¡Gracias!