ODSI Key Achivement:

The Protokernel PIP

PIP - Introduction

- **Goal**: Construct a flexible, minimal and (mathematically) secure proved kernel as highest privileged software component (TCB)
- State of the art:
  - Small kernel design: microkernel and nanokernel trends in 90s
  - Secure proved: *“proving existing kernels is hard”*
  - Only one open-source secure proved micro-kernel: SeL4

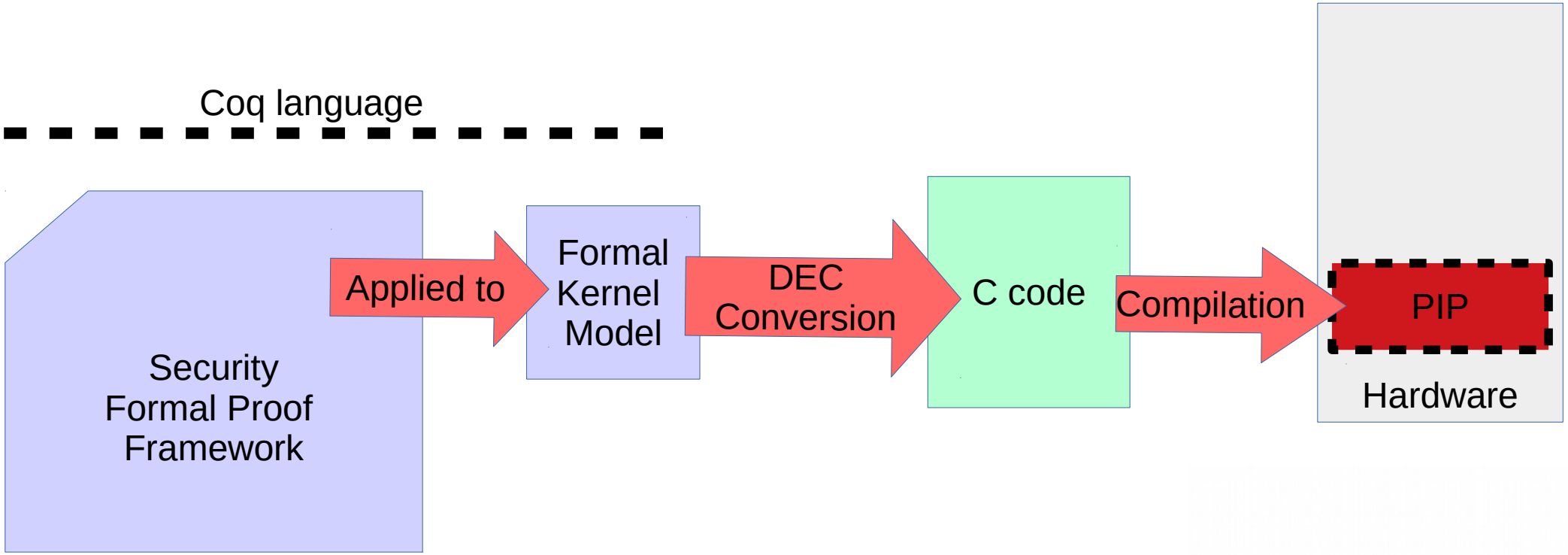
PIP - Design choices (1/2)

- **Smallest size possible**
  - smaller proofs: 200K proofs LoC for 500-1000 LoC
- **Proof oriented**
  - Using a subset of the formal language CoQ to describe the kernel model
  - Same formal language used for kernel model and associated proofs

PIP - Design choices (2/2)

- **Only memory isolation**
  - Seen as Hierarchical isolation model
  - Flexible for security model
- **Model to C translation**
  - Keeping semantic at *“instruction-level”*
  - Proved-translation via the translator DEC

PIP – Development workflow

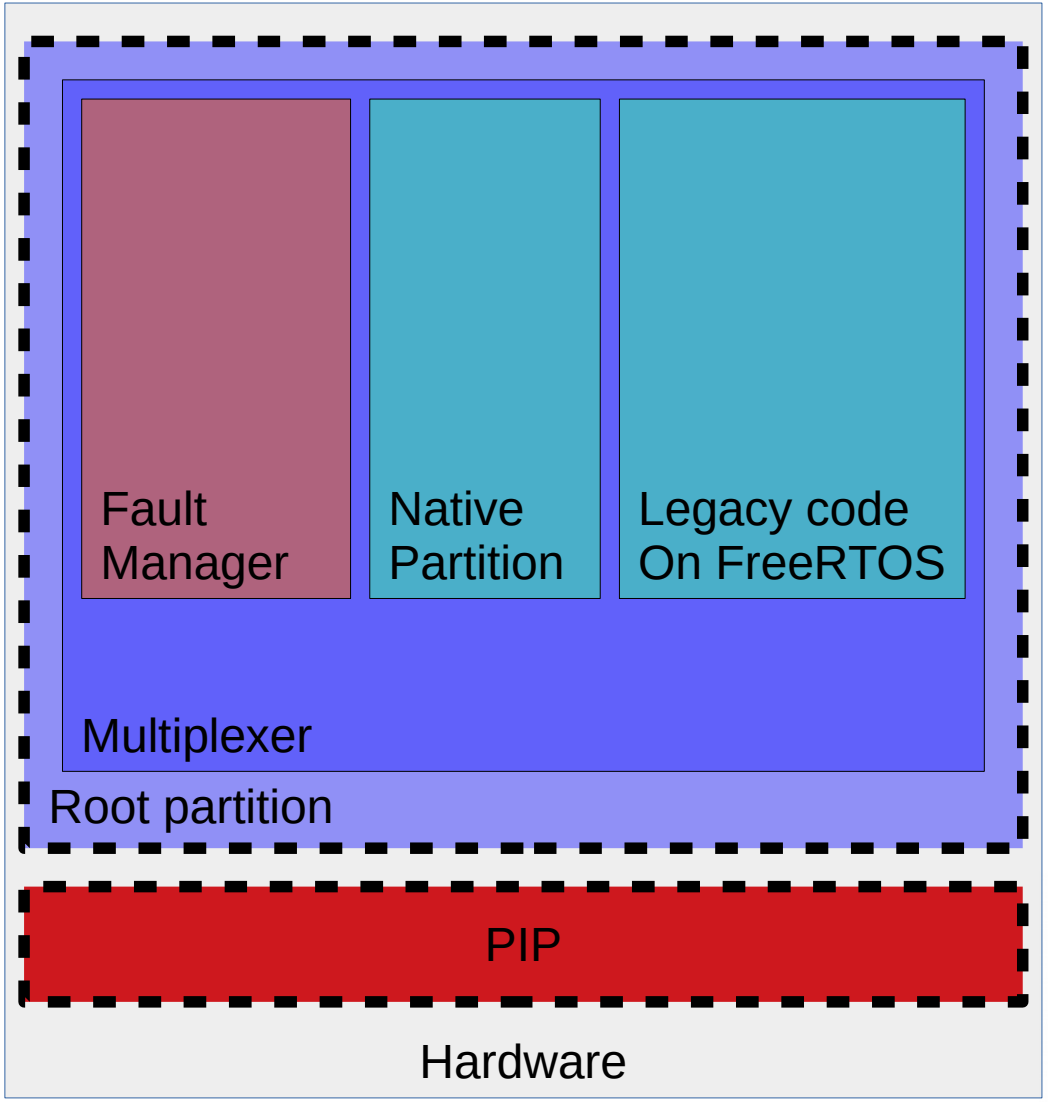


Celtic-Plus
Smart Connected World

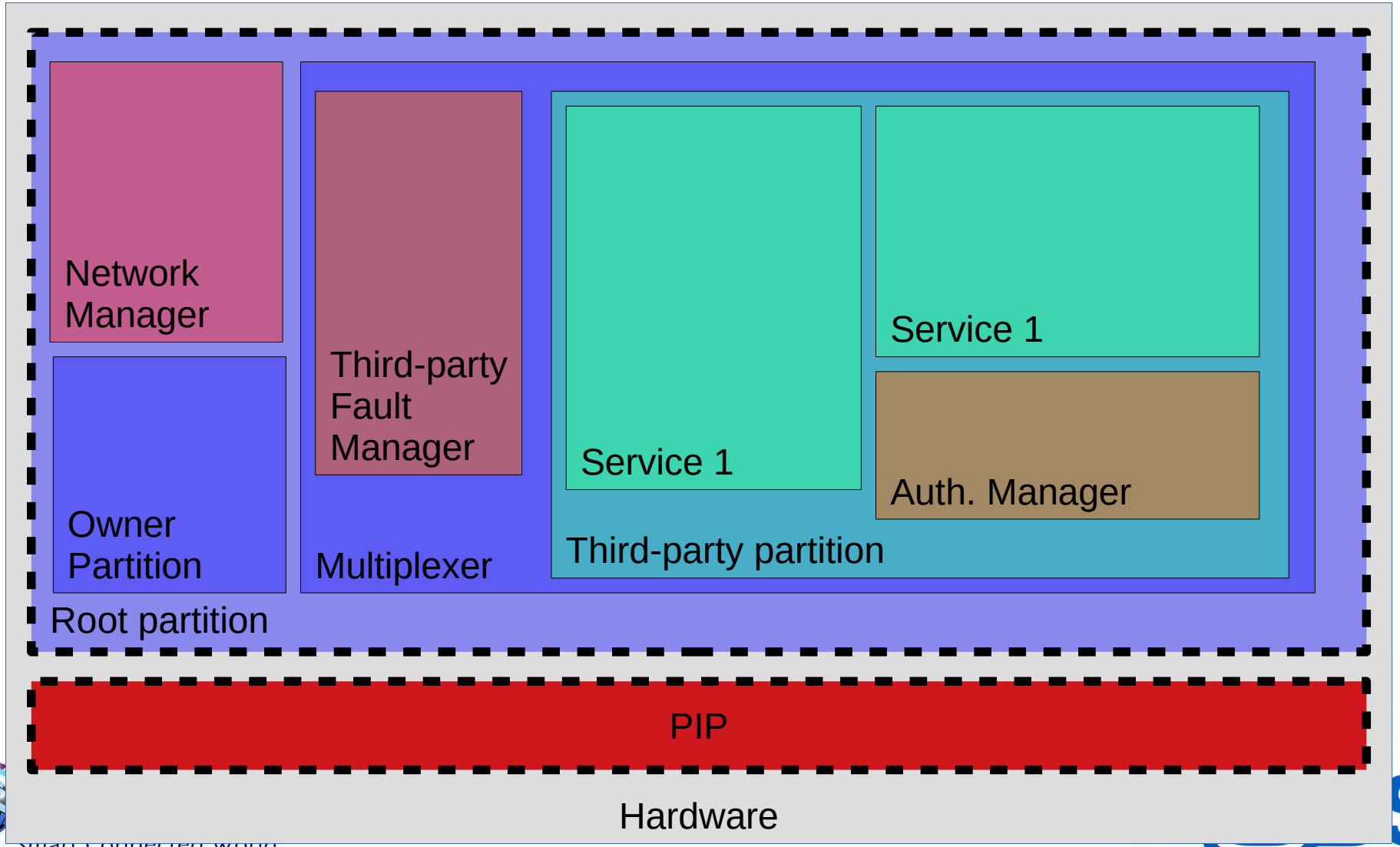
ODSI

PIP - Results

- **PIP is fully functional**
  - Behavior model written in a formal and portable language
  - Target: Intel Gallileo v2
- **FreeRTOS portage**
  - Legacy code can be ported to run on the top of Pip
- **Multiplexer (ORANGE)**
  - Designed to managed faulty partitions



Celtic-Plus
Smart Connected World
ODSI


Network Manager
Third-party Fault Manager
Service 1
Service 1
Owner Partition
Multiplexer
Auth. Manager
Third-party partition
Root partition
PIP
Hardware

PIP - Bootstrapping an ecosystem

- **PIP is Open Source**
  - Available on github https://www.github.com/2xs/
  - v0.3 released last week
- **Workshop ENTROPY 2018 (25th and 26th of January 2018)**
  - PIP has been presented to security OS research community
- **Meeting PIP User Club (7th December 2018)**
  - Offering help to industrial partners to develop solutions

Celtic-Plus
Smart Connected World

OSI