Les matinales de l'embarqué : L'Internet des Objets, quel système d'exploitation utiliser ?

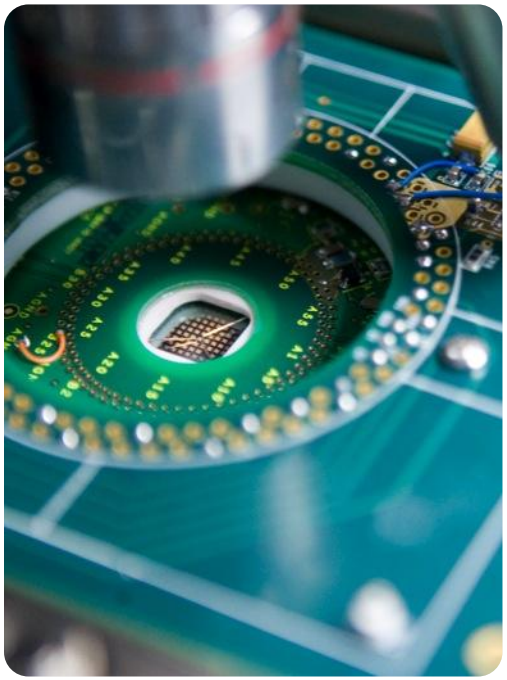
L'apport de l'OS temps-réel NI Linux RT pour les objets connectés

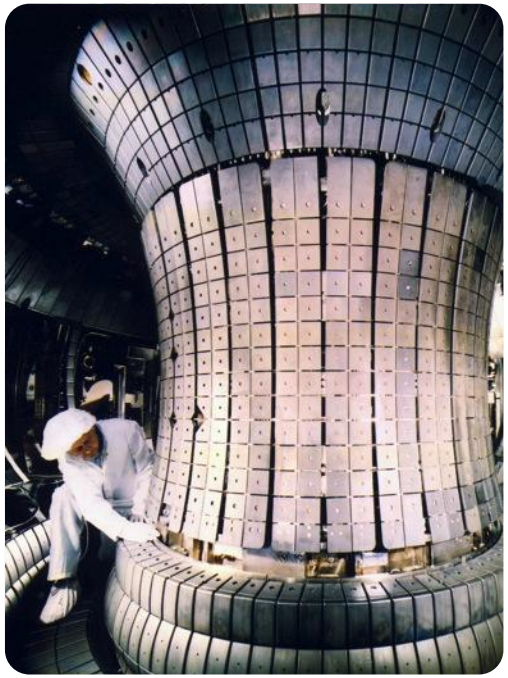
Antonin Goude
Ingénieur Produit Embarqué
National Instruments France

NATIONAL INSTRUMENTS

Our Mission

We equip engineers and scientists with tools that accelerate productivity, innovation, and discovery.





NATIONAL INSTRUMENTS

The company

Annual Revenue: $1.17 billion

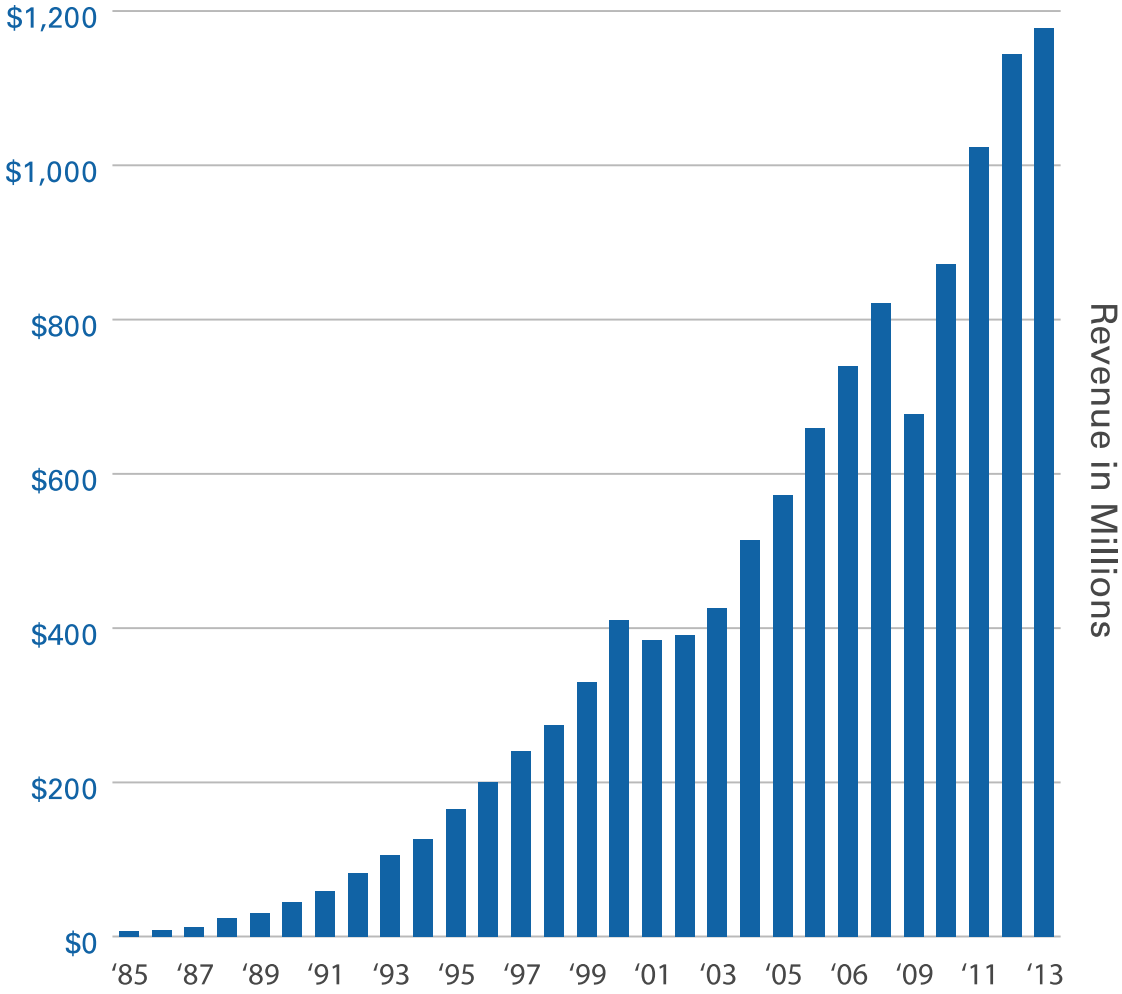
Global Operations:
Approximately 7,100 employees; operations in almost 50 countries

Broad Customer Base:
More than 35,000 companies served annually

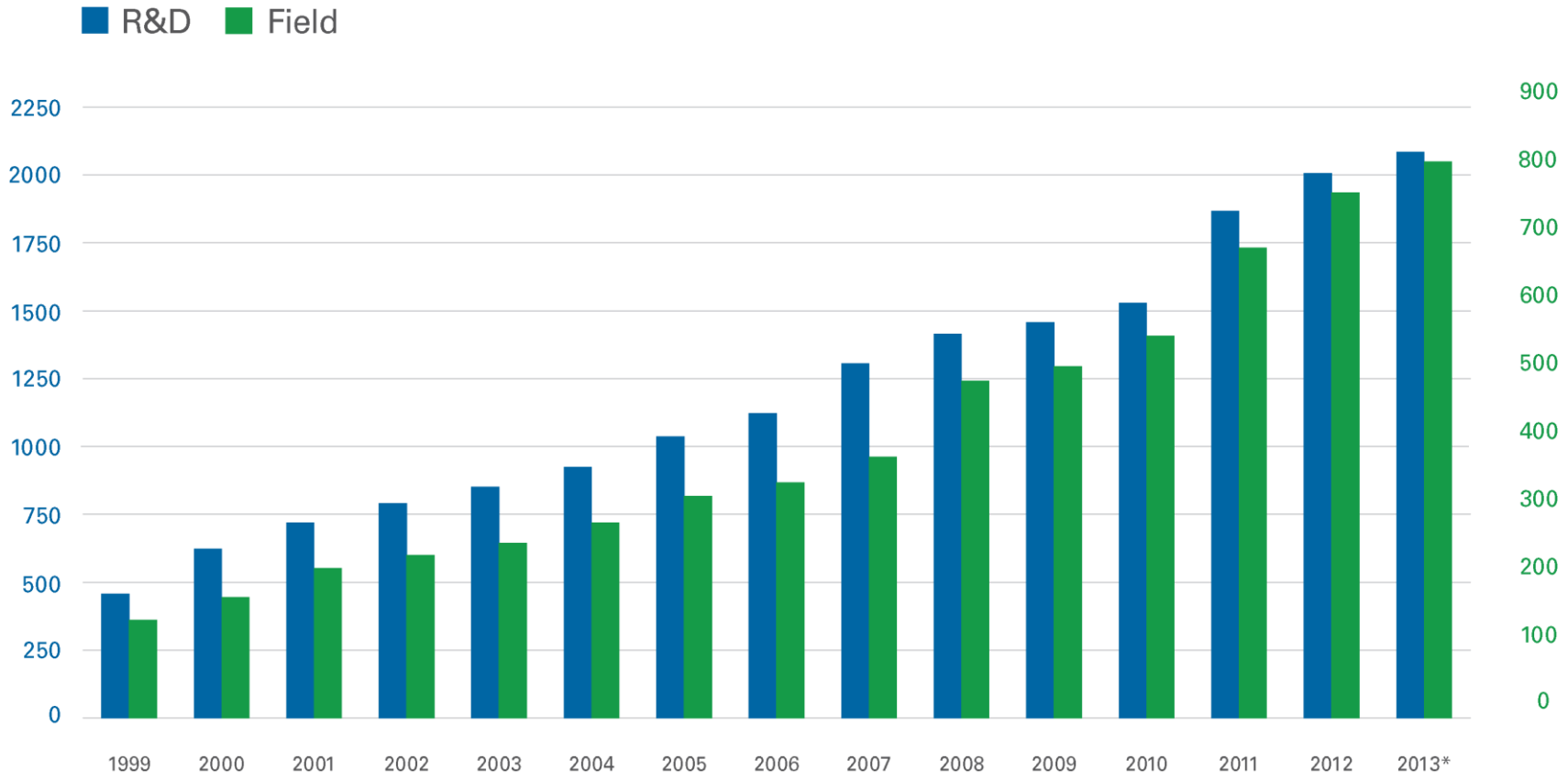
Diversity: No industry >15% of revenue

Culture: Ranked among the top 25 companies to work worldwide by the Great Place to Work Institute

Strong Cash Position:
Cash and short-term investments of $393 million at December 31, 2013

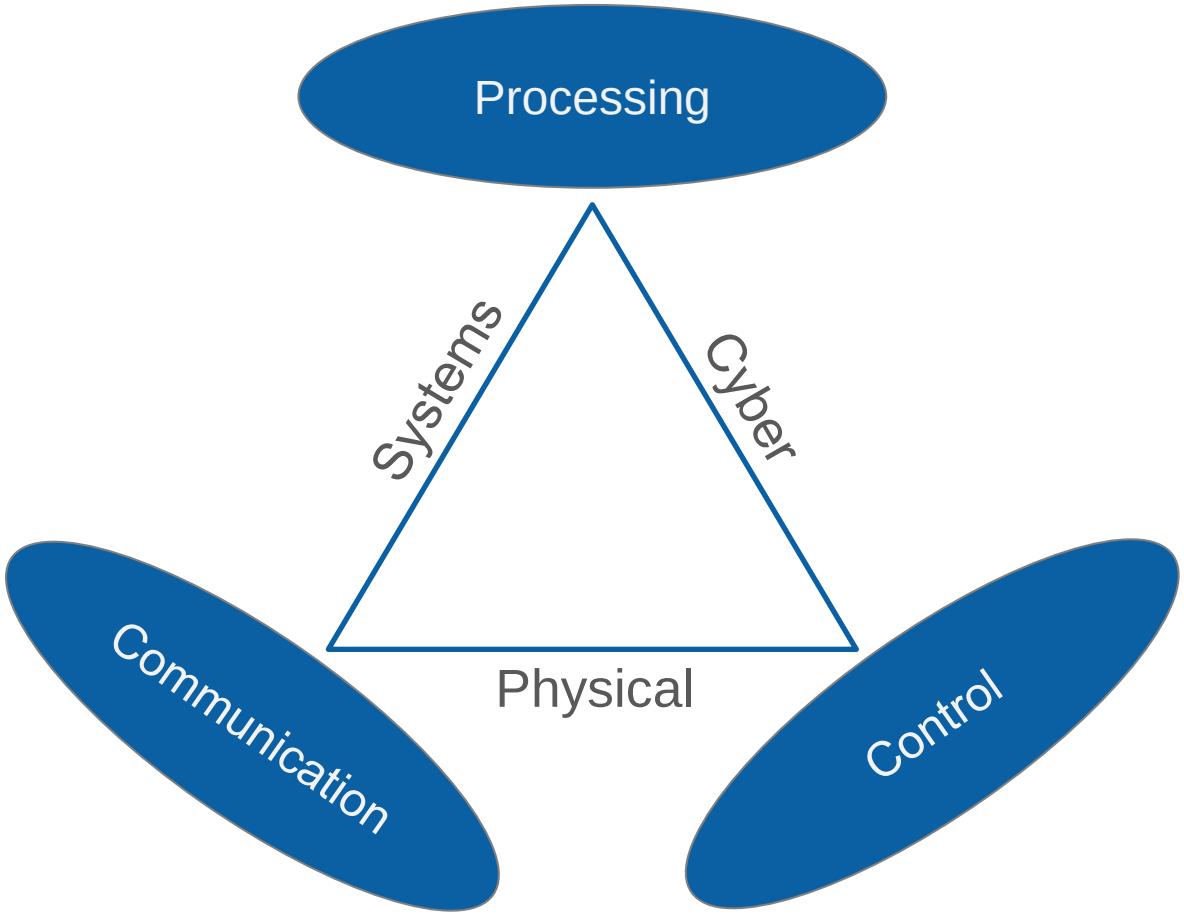


Investments to Drive Service and Innovation



* Represents National Instruments headcount in 2013

National Instruments - Internet of Things (IoT)

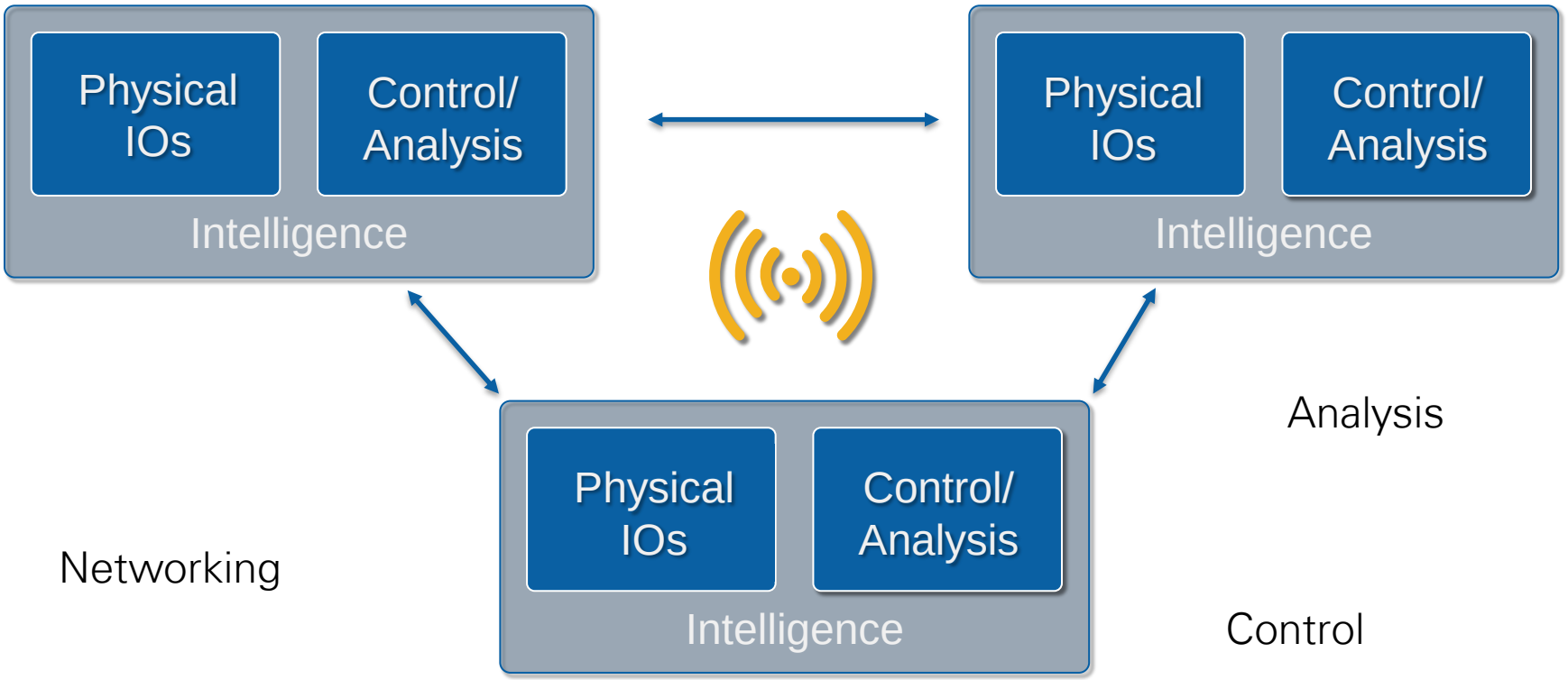


"Cyber-physical systems (CPS) enable the physical world to merge with the virtual leading to an Internet of things, data, and services."

– European Institute of Innovation and Technology

NATIONAL INSTRUMENTS

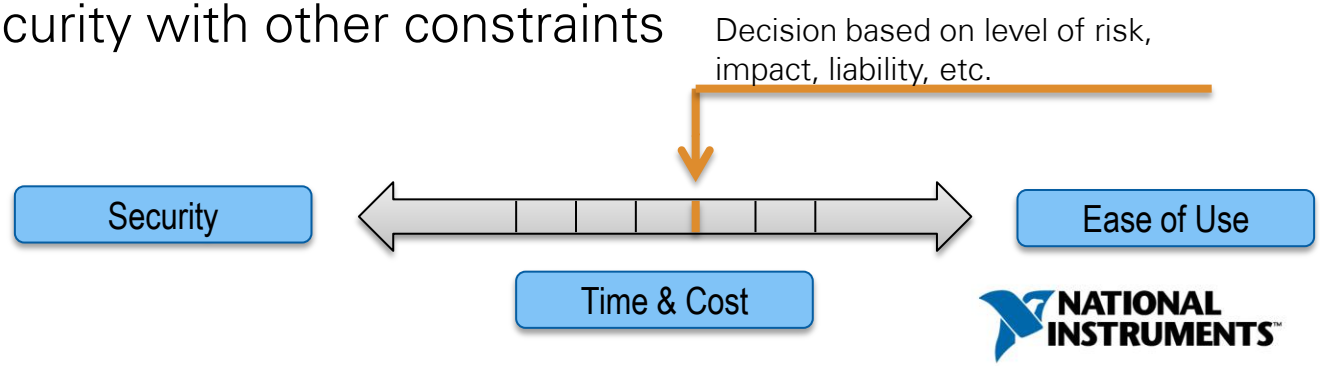
Cyber-phyical systems challenge



NATIONAL INSTRUMENTS

Key challenges for an IoT OS

- Flexibility
  - Fast evolution
  - Modularity
- Openness
  - Communication with 3rd party modules
  - Networking
- Security
  - Balancing security with other constraints



Which OS approach for the IoT?
NI Linux Real-Time

LONG TERM SUPPORT INITIATIVE

- Owned and maintained by NI
  - Custom built and optimized for NI embedded hardware
    - Supports ARM and IA64, with cross-compilers provided
  - Based on **OpenEmbedded** framework
  - NI Package Repository: **download.ni.com/ni-linux-rt/**
    - Over 3,000 packages
  - OS source: **github.com/ni**
- PREEMPT_RT
  - Enables real-time reliability through pre-emption, priority inheritance, and scheduling
  - Standard approach to real-time performance on Linux

NATIONAL INSTRUMENTS

Which OS approach for the IoT?
NI Linux Real-Time

Architectures	X86_64	ARMv7
Bootloader	GRUB	uBoot
Kernel Version	3.10-rt-ltsi	3.2-rt
File System	ext4	UBIFS
Desktop Environment	XFCE	N/A
Built With	Yocto/OpenEmbedded	
Package Management	OPKG	
Init System	SysV	
Device Manager	udev	

- NI Linux RT enables the use of different languages for processor programming:
  - LabVIEW, C/C++, interoperates with LabVIEW-programmed FPGA
- Access FPGA technology without HDL expertise

Quality of Life on NI Linux Real-Time

- Convenience of a General Purpose OS
  - Desktop UI, File Manager, Terminal Emulator, Text Editor
  - Permissions, Application Isolation, Virtual Memory, no reboot required for Time Zone and IP Configuration changes
- Access to popular interpreters
  - Python, Perl, etc.
- Common Linux utilities
  - top, ps, netstat, etc.

NATIONAL INSTRUMENTS

Security on NI Linux Real-Time

- SSL enabled by default
  - Can programmatically install software over SSL
  - Can use public keys for SSH
- HTTPS-only communication possible
  - Can turn off HTTP version of the System Web Server
- IPTables available for setting up a firewall

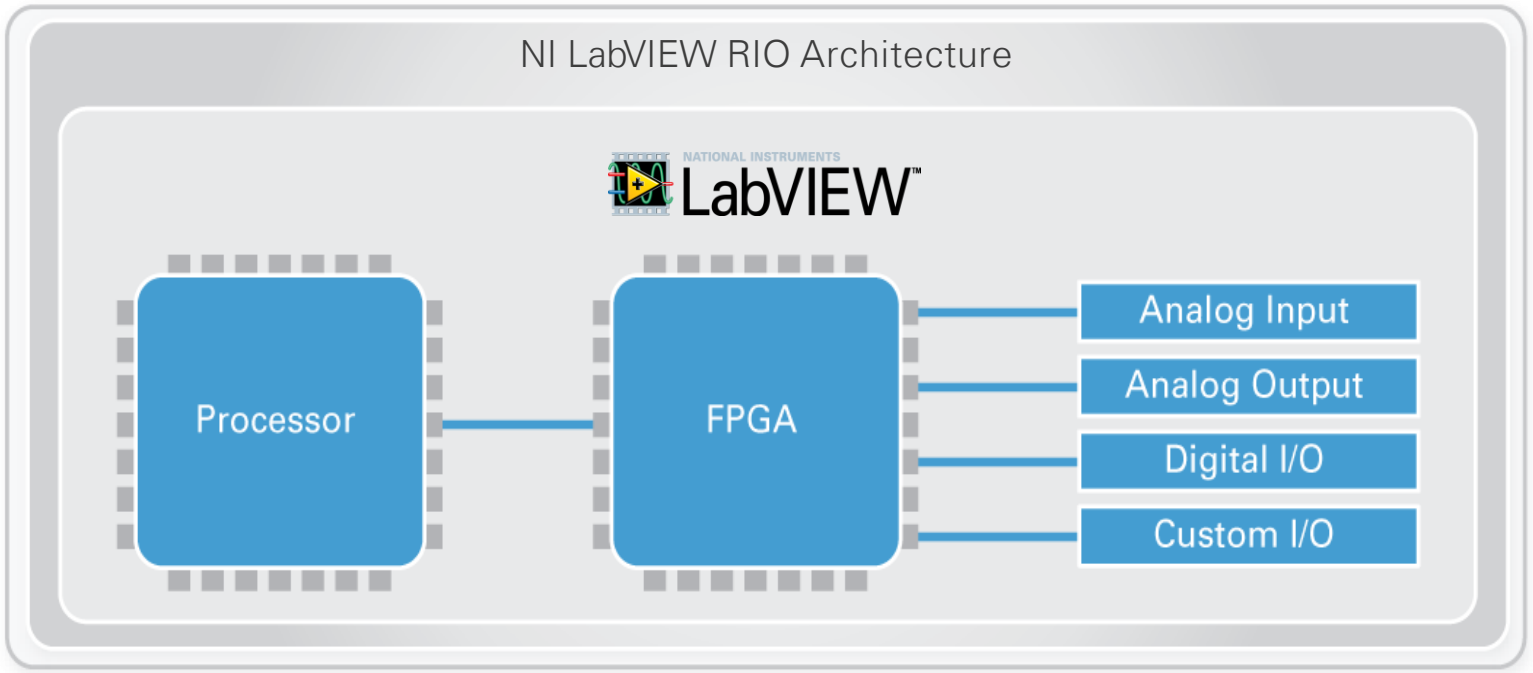
- OpenVPN available for setting up a VPN

OPENVPN™

- SELinux

NATIONAL INSTRUMENTS

The foundation for innovation



NATIONAL INSTRUMENTS

What platform can be targeted?

- From prototyping to deployment using NI RIO Hardware

- **NI CompactRIO Controller**
  - Xilinx Zynq-7020
    - 667 MHz Dual-Core ARM Cortex-A9
    - Artix-7 FPGA Fabric

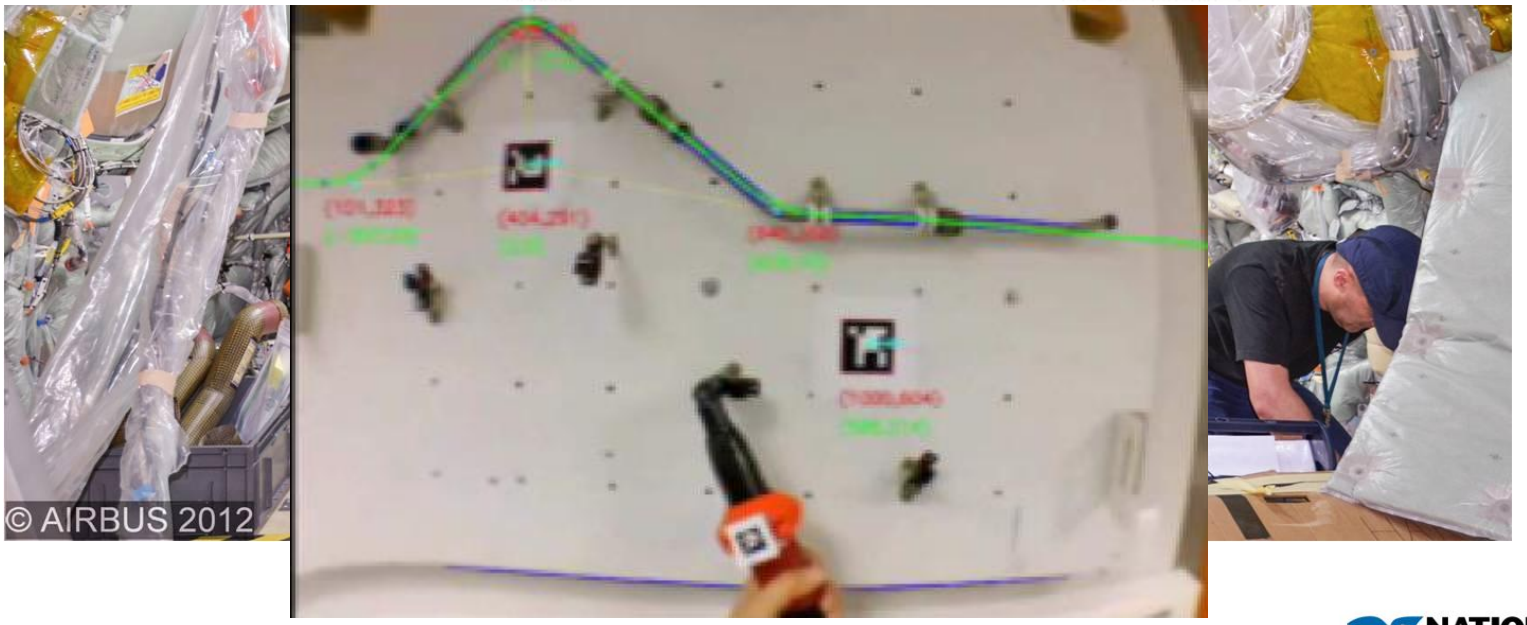
- **NI System on Module**
  - Xilinx Zynq-7020
    - 667 MHz Dual-Core ARM Cortex-A9
    - Artix-7 FPGA Fabric
  - 5.08 cm x 7.82 cm

Example: Airbus and NI collaboration
Intelligent devices for the Future of Aircraft Factory

HD camera embedded on operator glasses

Processor embedded in operator suit

Embedded Image Processing Software

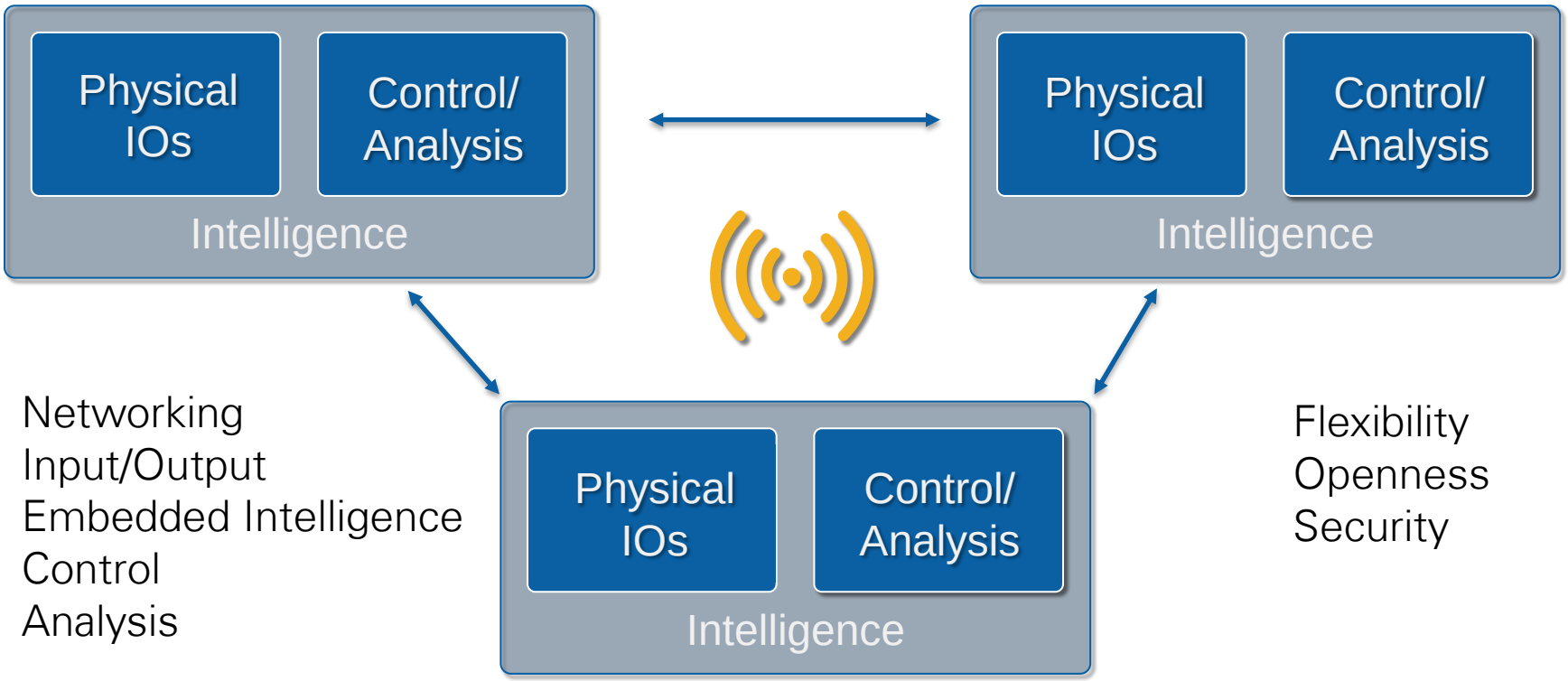


NATIONAL INSTRUMENTS

Example: Airbus and NI collaboration
Intelligent devices for the Future of Aircraft Factory

- Key points:
  - Open real-time OS based on Linux distribution
    - Desktop UI, Peripherals, System Administration, Real-Time schedulers
  - Reuse of internal libraries
    - Networking, Configuration Management, Simulation, Monitoring, etc.
  - Reuse C/C++ code in and alongside LabVIEW Real-Time built applications
    - FPGA Interface C API, System Configuration C API
  - Same approach for prototyping and deployment (cRIO to SoM)

Key Take-Aways



Networking
Input/Output
Embedded Intelligence
Control
Analysis

Flexibility
Openness
Security

Thank you

Questions?