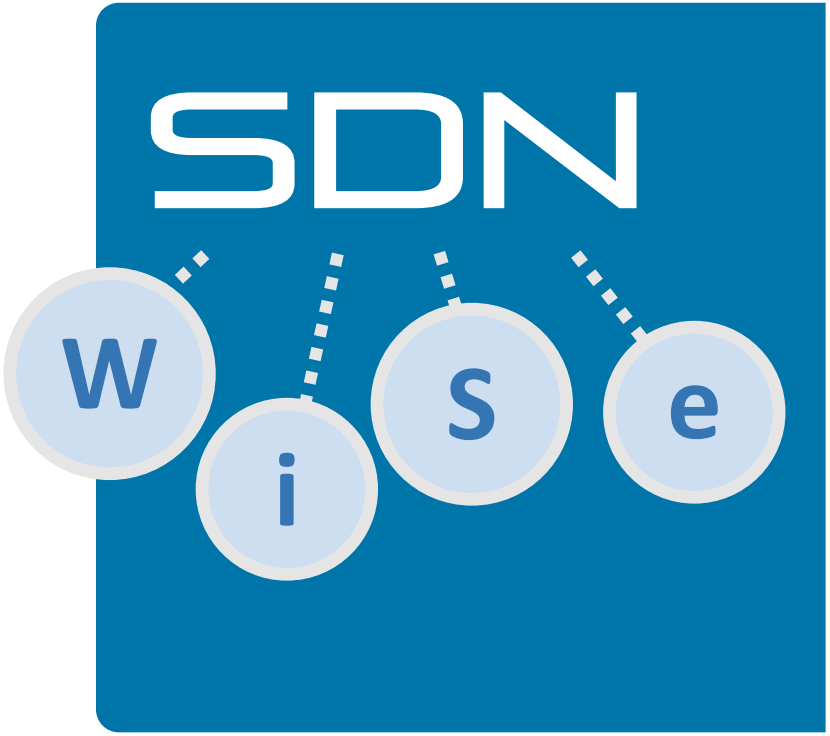


Networking issues for the Internet of Things

Giacomo Morabito

University of Catania

2015 – IEEE SPS – Italy Chapter
Summer School on Signal Processing

Acknowledgements

- The following are the results of the work carried out with many colleagues and reported in the following papers:
  - L. Atzori, A. Iera, and G. Morabito. "*The Internet of Things: A Survey*". **Computer Networks**. October 2010
  - A. Iera, G. Morabito, and L. Atzori. "*Understand the IoT evolution to master the IoT revolution*". Tutorial at **European Wireless**. May 2015
  - L. Galluccio, S. Milardo, G. Morabito and S. Palazzo. "*SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for WIreless Networks*". **IEEE Infocom**. April 2015
  - -- "*Reprogramming Wireless Sensor Networks by Using SDN-WISE: a Hands-On Demo*". **IEEE Infocom -- Demo**. April 2015
  - C. Buratti, A. Stajkic, G. Gardasevic, S. Milardo, M. D. Abrignani, S. Mijovic, G. Morabito, and R. Verdone. "*Testing Protocols for the Internet of Things on the EuWIn Platform*", **IEEE Internet of Things Journal**
  - A. C. Anadiotis, L. Galluccio, S. Milardo, G. Morabito and S. Palazzo. "*An Integrated Network Operating System for the Internet of Things: Design, Implementation and Experimentation*". **Under review**

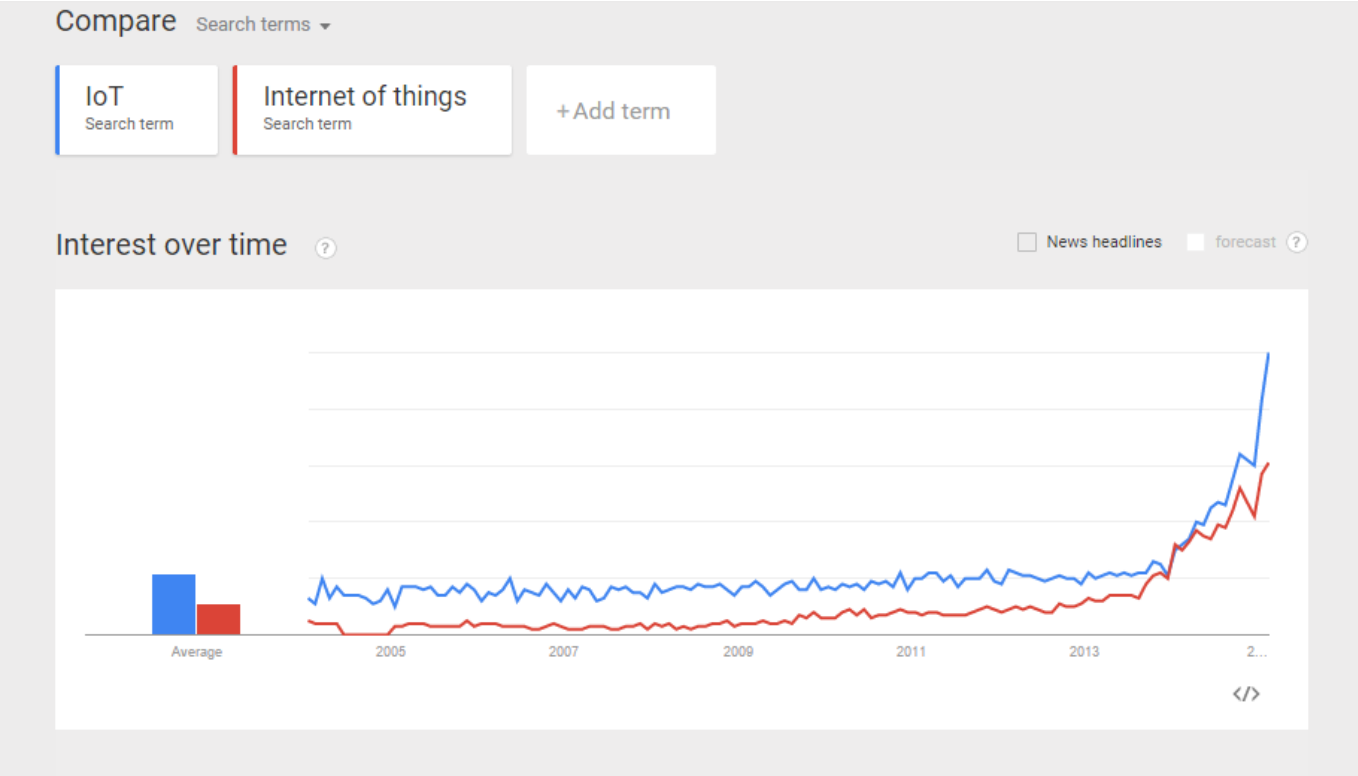
Outline

- Great, however…
- The need of a network operating system for the IoT
- Network operating systems
  - Open Network Operating System (ONOS)
- SDN for wireless sensor and actor networks
- Prototype implementation
- Conclusions

•L. Atzori, A. Iera, and G. Morabito. "*The Internet of Things: A Survey*". **Computer Networks**. October 2010
•A. Iera, G. Morabito, and L. Atzori. "*Understand the IoT evolution to master the IoT revolution*". Tutorial at **European Wireless**. May 2015

The “next” big thing in communications? General public

- 50 billion IoT devices by 2020 (Cisco Systems)
- Google trends (http//www.google.com/trends)

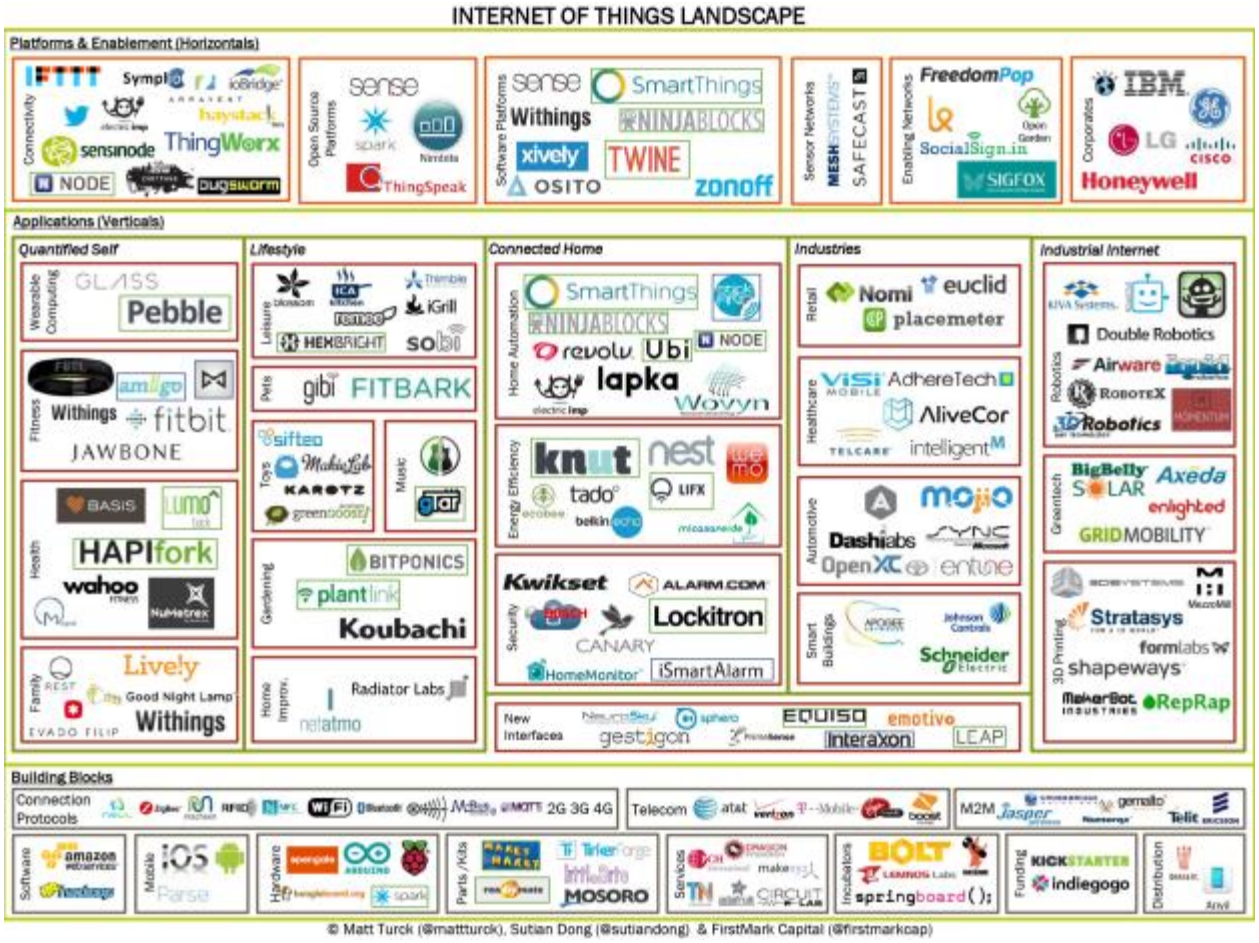


The "next" big thing in communications? Scientific community

Most downloaded papers:

- IEEE (4th):
  - A. Zanella, N. Bui, A. Castellani, L. Evangelista, M. Zorzi, "**Internet of Things** for Smart Cities", 2014
    - Top 10 for almost 1 year
- ACM-SIGCOMM:
  - F. Bonomi, R. Milito, J. Zhu, S. Addepalli, "Fog computing and its role in the **Internet of Things**", 2012
    - BTW: 4th and 5th are about SDN and ONOS. We'll talk later about these...
- Elsevier – Computer science:
  - L. Atzori, A. Iera, G. Morabito, "The **Internet of Things**: a Survey", 2010
    - BTW: the second is "**Internet of Things** (IoT): A vision, architectural elements, and future directions"

The “next” big thing in communications? Industry



From techcruch.com

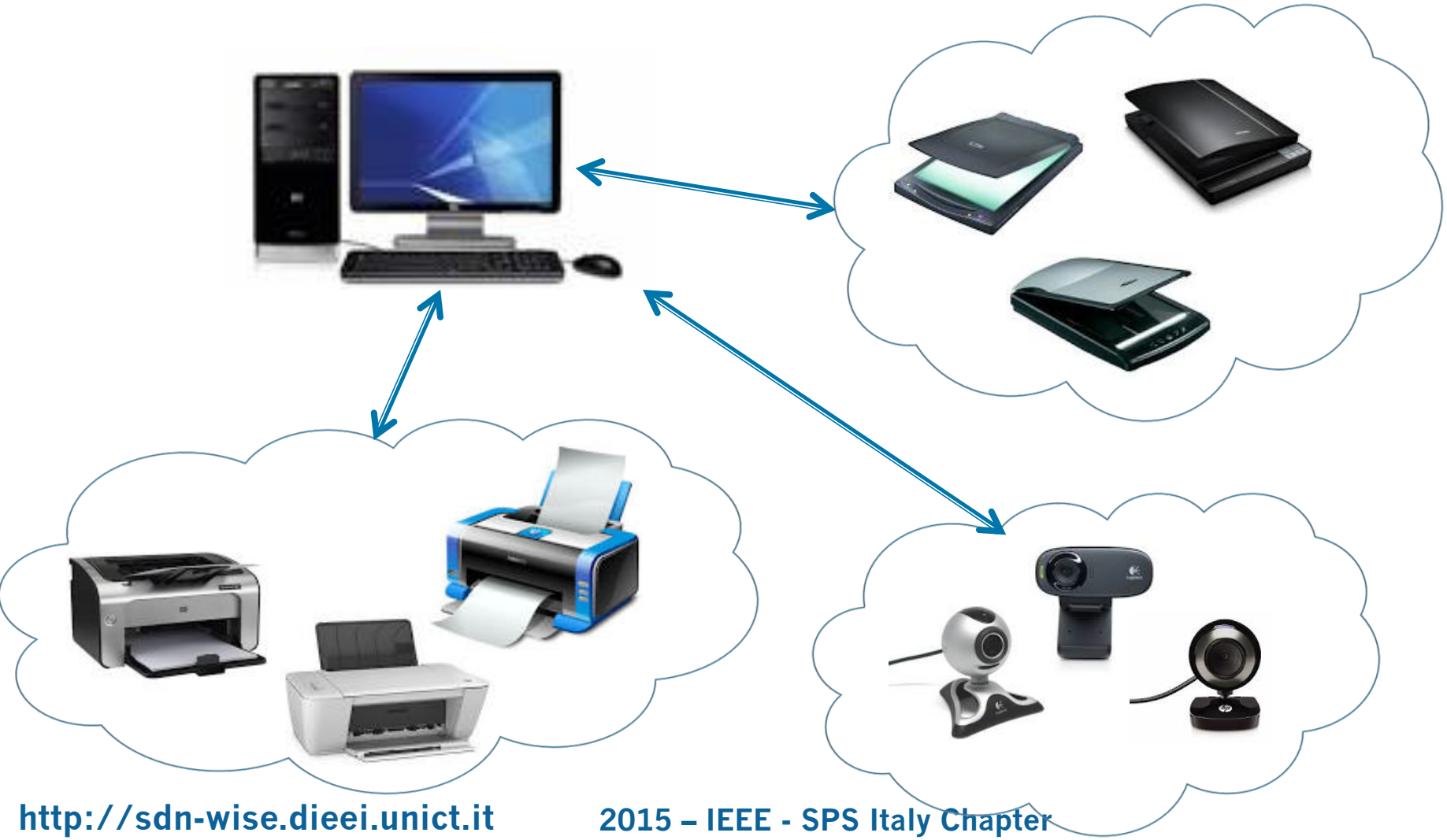
The standards

I GENERATION	Tagged objects	EPCglobal	
	Sensor networks	IEEE 802.15.4	
	Object description	IEEE 1451	
II GENERATION	Internetworking	IETF 6LoWPAN	IETF ROLL RPL
	Web of Things	IETF CoAP	OASIS DPWS
	Architecture	ITU-T FS M2M	oneM2M
III GENERATION	Cloud computing	TIA TR50	
	Social networking	*Missing*	
	Future Internet	IETF ICNRG	
	Semantics	W3C SSN	

Intranets **of Things instead of** *Internet* **of Things**

- Several IoT platforms developed independently without a clear reference architecture → **Fragmented technological landscape**
  - Low interoperability
  - Low expandability
  - Low reusability
- Where would you test your new algorithm for in-network data processing?

Existence of heterogeneous platforms: not a new problem



Existence of heterogeneous platforms: how it was addressed?

Word

Adobe

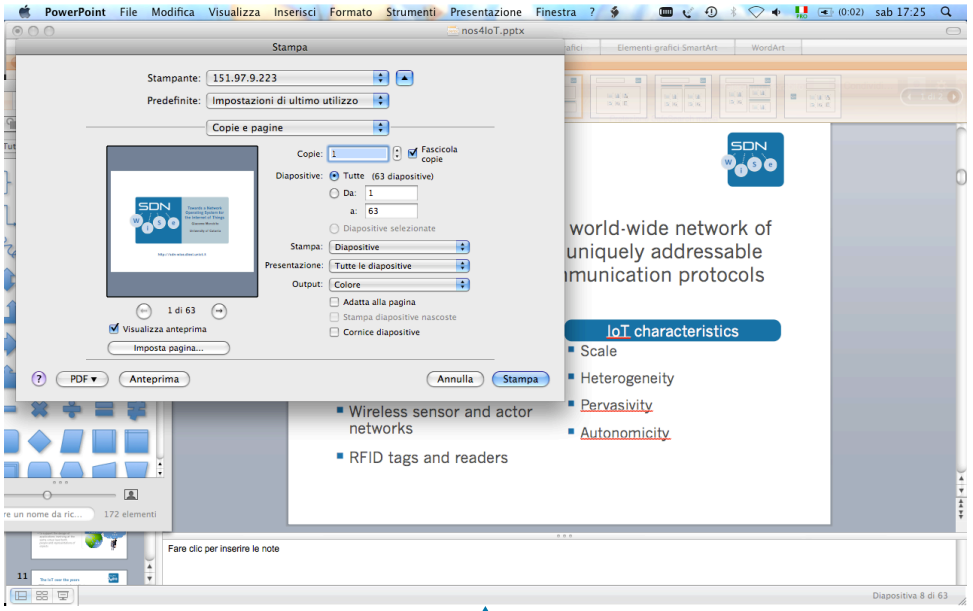
PPT

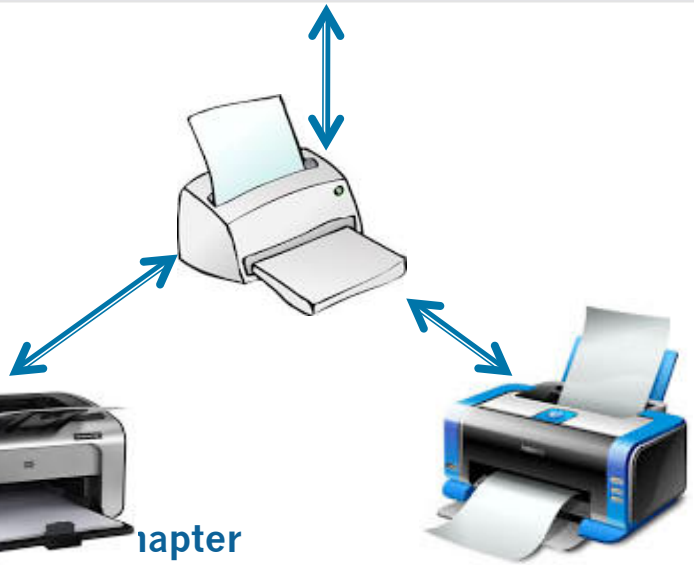
Printing service

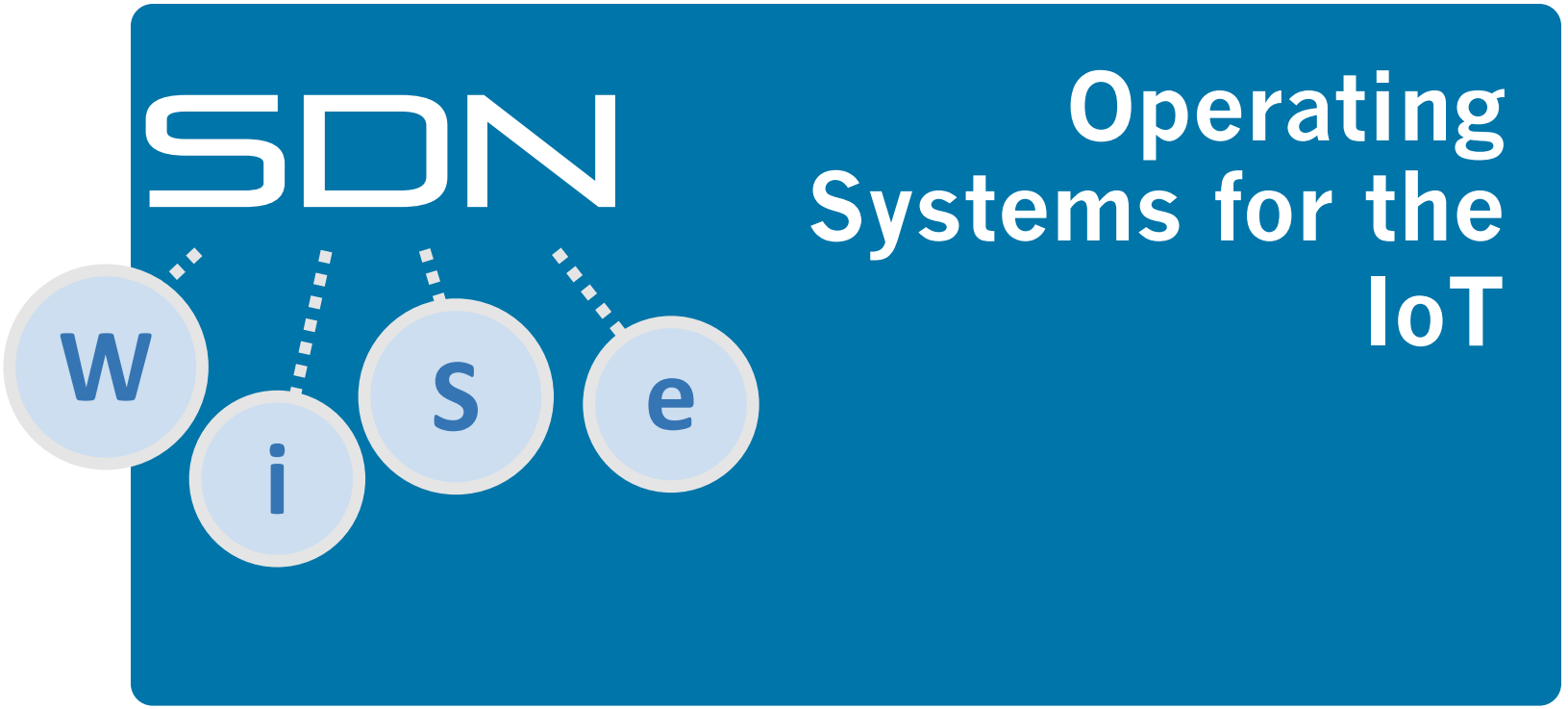
Operating system

Printer abstraction

Driver






SDN
wise
Operating Systems for the IoT

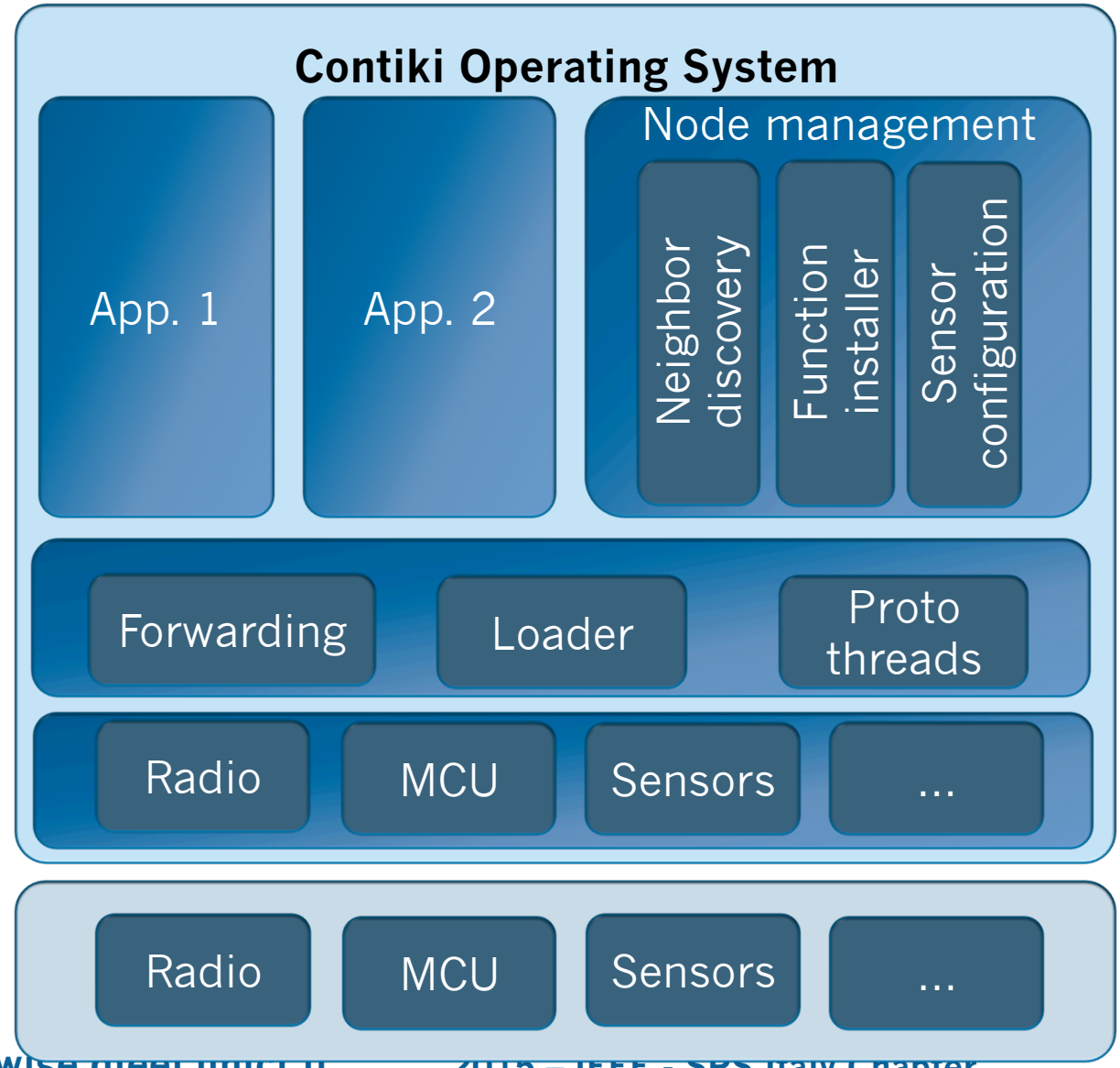
Existing operating systems for the IoT

- Contiki
- RIOT
- CCN-Lite

Contiki

- Open source OS for the IoT
- Supports:
  - IPv6 and IPv4
  - 6LOWPAN
  - RPL
  - CoAP
- Active community
- Industrial interest
- Network simulator: Cooja

Contiki: Architecture



Core

Drivers

Hardware

Contiki hardware

MCU/SoC	Radio	Platforms	Cooja simulation support
RL78	ADF7023	EVAL-ADF7023DB1	-
TI CC2538	Integrated	cc2538dk	-
TI MSP430x	TI CC2420	exp5438, z1	Yes
TI MSP430x	TI CC2520	wismote	Yes
Atmel AVR	Atmel RF230	avr-raven, avr-rcb, avr-zigbit, iris	-
Atmel AVR	TI CC2420	micaz	Yes
Freescale MC1322x	Integrated	redbee-dev, redbee-econotag	-
ST STM32w	Integrated	mb851, mbxxx	-
TI MSP430	TI CC2420	sky, jcreate, sentilla-usb	Yes
TI MSP430	TI CC1020	msb430	-
TI MSP430	RFM TR1001	esb	Yes
Atmel Atmega128 RFA1	Integrated	avr-atmega128rfa	-
Microchip pic32mx795f512l	Microchip mrf24j40	seed-eye	-
TI CC2530	Integrated	cc2530dk	-
RC2300/RC2301	Integrated	sensinode	-
6502	-	apple2enh, atari, c128, c64	-
Native	-	native, minimal-net, cooja	Yes

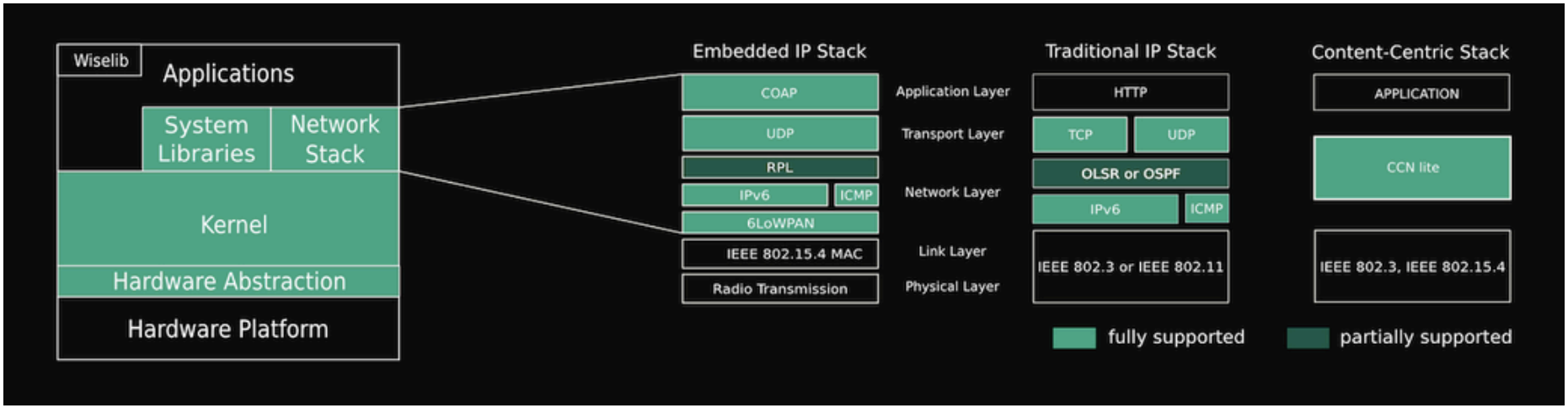
RIOT

- HW Support: MSP430 , ARM7 , CORTEXM0-4, **X86**
- Drivers for many transceivers and sensors
- SW Support:
  - AODVv2
  - 6LoWPAN
  - RPL
  - TCP with header compression for 6LoWPAN
  - **CCN-lite**
  - OpenWSN
  - CoAP, CBOR, and UBJSON

RIOT

- **Modular**: When programming a node you can decide which modules to be loaded
- Written in C/C++
- Has multithreading and RealTime operations
- Same memory space requirements as TinyOS

RIOT Stack


Wiselib
Applications
System Libraries
Network Stack
Kernel
Hardware Abstraction
Hardware Platform
Embedded IP Stack
COAP
UDP
RPL
IPv6
ICMP
6LoWPAN
IEEE 802.15.4 MAC
Radio Transmission
Application Layer
Transport Layer
Network Layer
Link Layer
Physical Layer
Traditional IP Stack
HTTP
TCP
UDP
OLSR or OSPF
IPv6
ICMP
IEEE 802.3 or IEEE 802.11
Content-Centric Stack
APPLICATION
CCN lite
IEEE 802.3, IEEE 802.15.4
fully supported
partially supported

RIOT and the others

OS	Min RAM	Min ROM	C Support	C++ Support	Multi-Threading	MCU w/o MMU	Modularity	Real-Time
Contiki	< 2kB	< 30kB	●	✗	●	✓	●	●
Tiny OS	< 1kB	< 4KB	✗	✗	●	✓	✗	✗
Linux	~ 1MB	~ 1MB	✓	✓	✓	✗	●	●
RIOT	~ 1.5kB	~ 5kB	✓	✓	✓	✓	✓	✓

✓ Full support
● Partial support
✗ No support

CCN-LITE

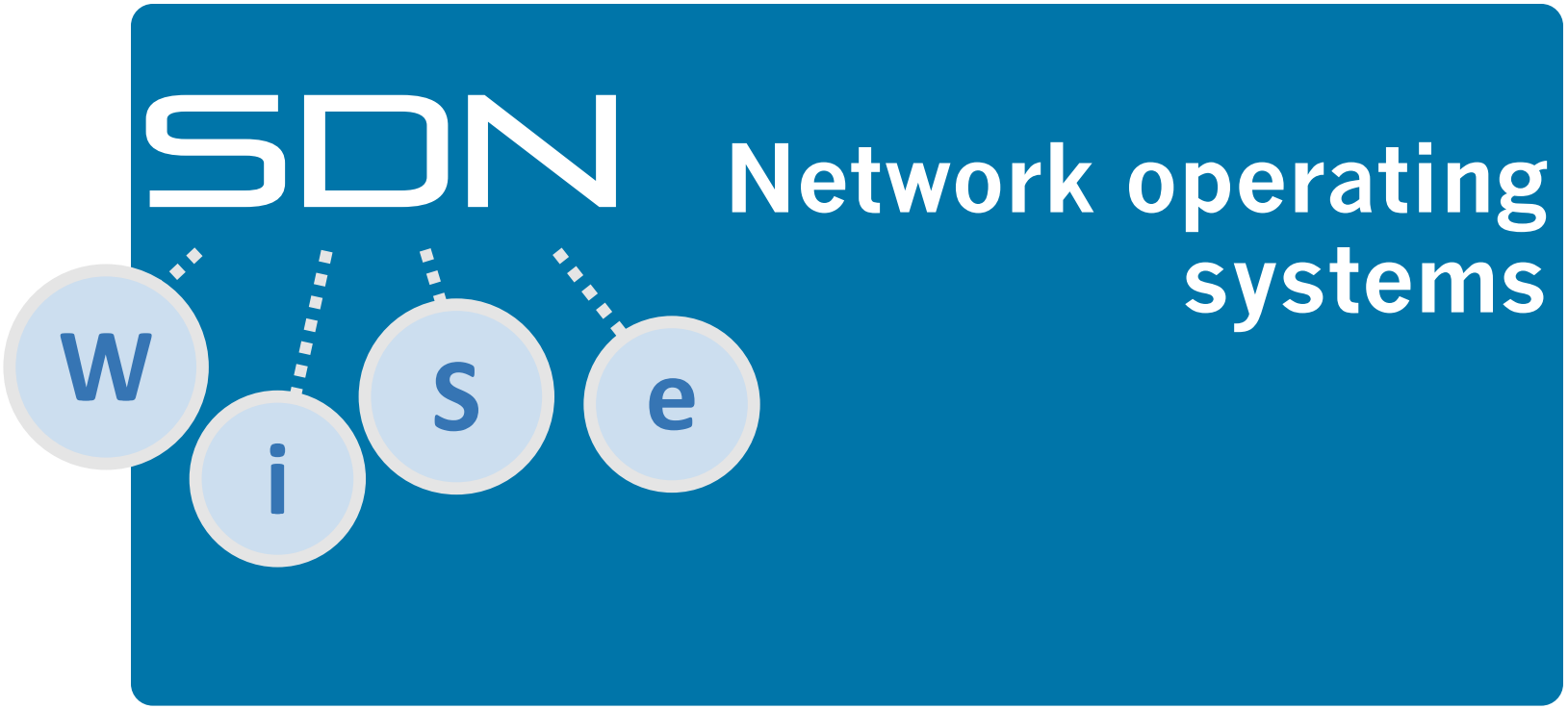
- CCN-lite has been included in the RIOT operating system for the Internet of Things (IoT): http://www.riot-os.org/
- Objective of CCN-lite has been to make the most popular implementation of a content-centric networking client (CCNx) runnable in a IoT device

Great! However…

- IoT applications can be developed without considering the specific features of the hardware platform
- Only/mostly for 6LOWPAN nodes
- For what concerns networking:
  - Is it possible to deploy new routing algorithms application specific?
  - Is it possible to decide the path according to the values measured by a sensor?
  - Is it possible to change the network topology depending on the characteristics of the flow?
- For DSP researchers:
  - If you develop a new scheme which requires packets to follow specific routes, how would you test it?

Great! However, we also need…

- A **Network Operating System**…


SDN
W
i
S
e
Network operating systems

Network Operating Systems (NOS)

- Holistic network resource management
- Access network resources through dedicated services
- Integrate heterogeneous network elements through drivers that implement NOS functionality leveraging device-specific technology
- Support third-party network services deployment on top of them (e.g. Routing as a Service)

Open Network Laboratory

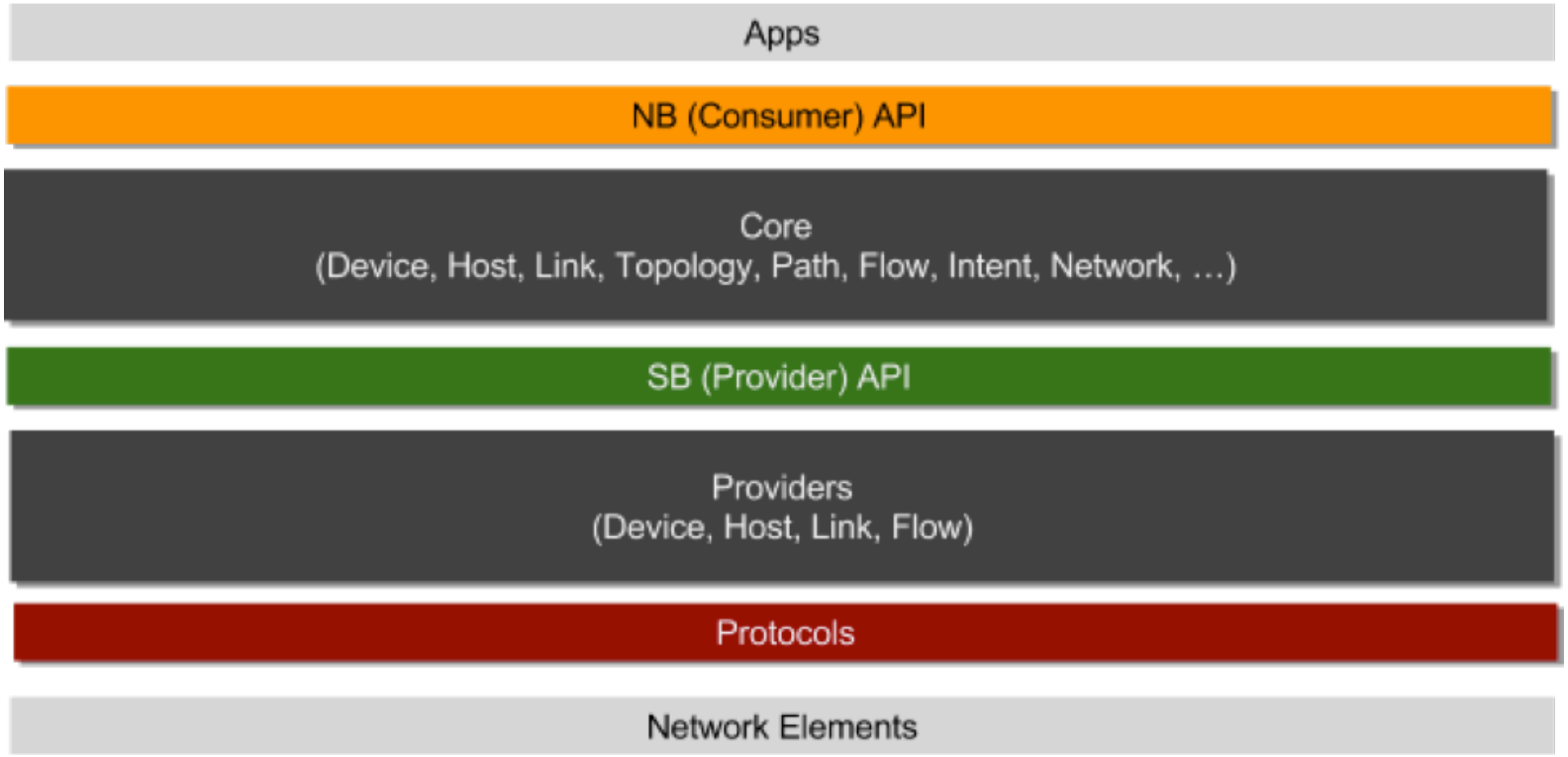
ON.LAB

- No profit established in 2012
- Partners (it might be outdated):

Open Network Operating System (ONOS)

- Open Source
- Java-based
- OSGi deployment
- Modular architecture
- Extensible components
- Originally designed for OpenFlow

ONOS Overall Architecture

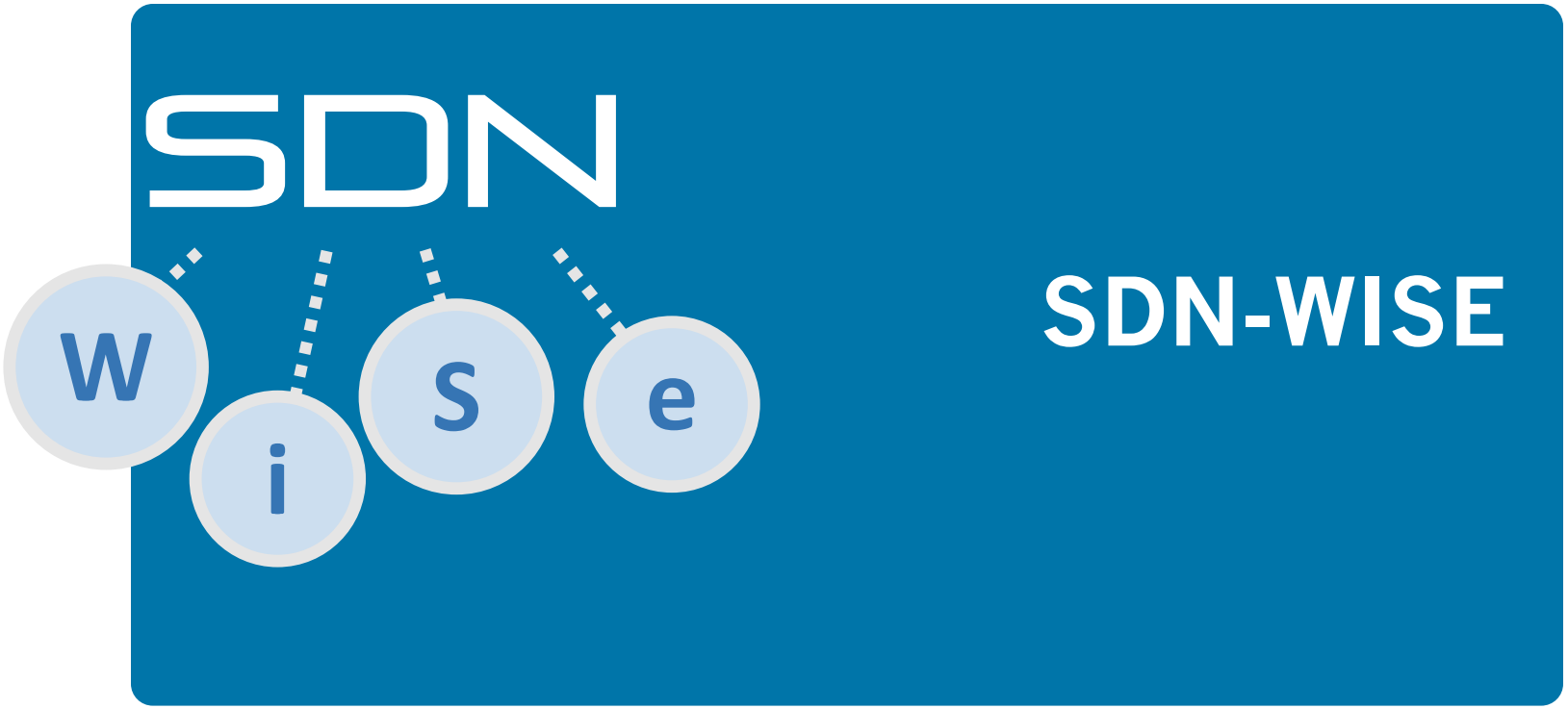


ONOS Functional Organization

- Subsystems consisting of elementary services deployed in several layers
- Communication between layers is established through system-wide APIs:
  - Northbound API provides network applications and services with access to ONOS subsystems
  - Southbound API enables the deployment of core services regardless the device-specific implementation details

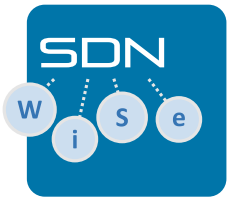
SDN for Network Management

- Separation of control and data plane
- Control plane remotely managed by dedicated services – *controllers*
- OpenFlow is the de facto standard for communication between controllers and network elements
- However: not all network elements can/do support OpenFlow



•L. Galluccio, S. Milardo, G. Morabito and S. Palazzo. "*SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for WIreless Networks*". **IEEE Infocom**. April 2015
•-- "*Reprogramming Wireless Sensor Networks by Using SDN-WISE: a Hands-On Demo*". **IEEE Infocom -- Demo**. April 2015

A few facts about wireless sensor networks



- Mature technology since early 2000s
- Challenging communication & networking environment
- Requirements extremely application specific

The bottom-line...

There is nothing like a one-fits-all solution

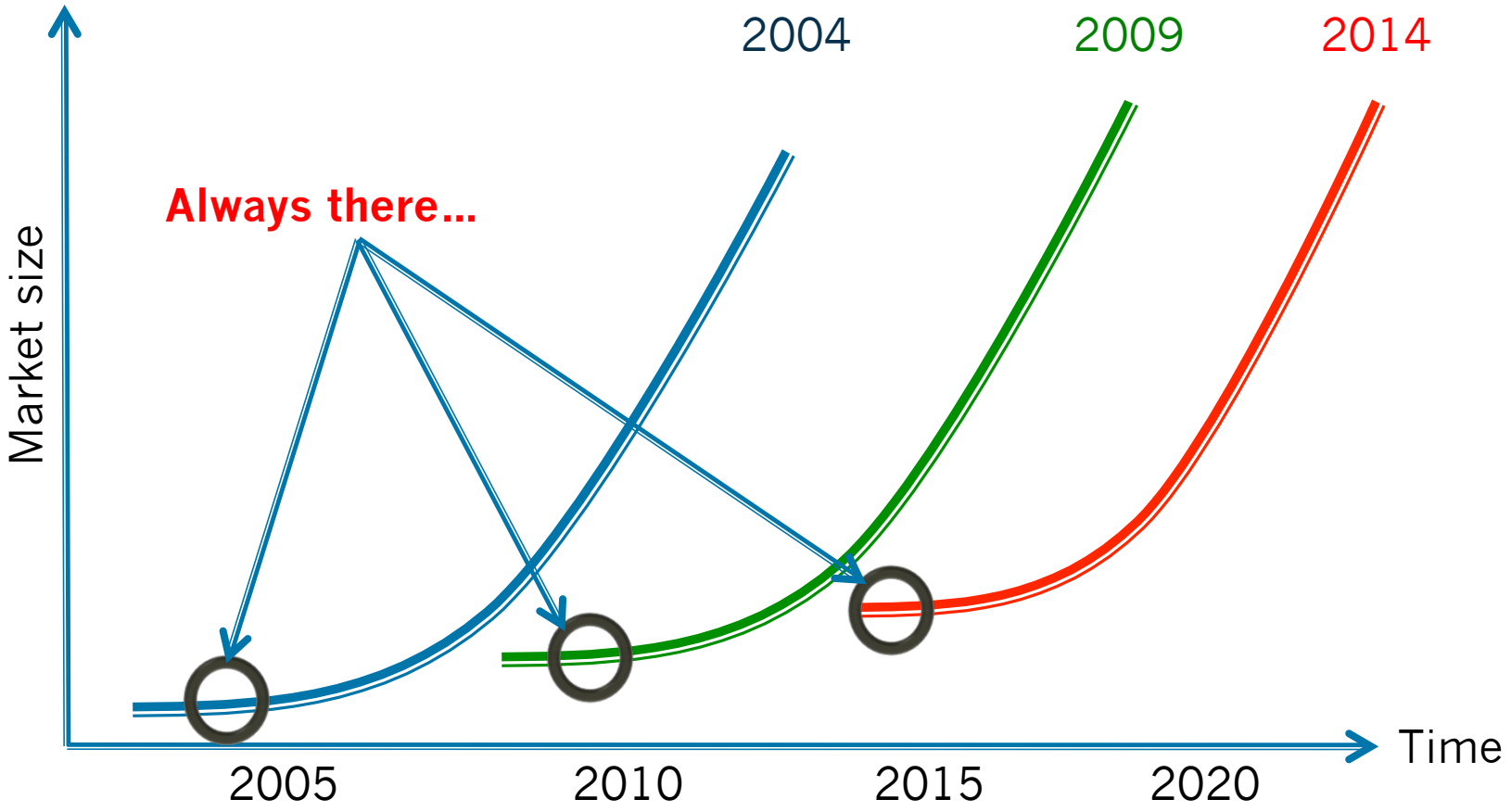
Upsides:

- Large number of solutions proposed
- Deep understanding of the WSN domain
- Zillions of papers, citations, academic promotions, projects

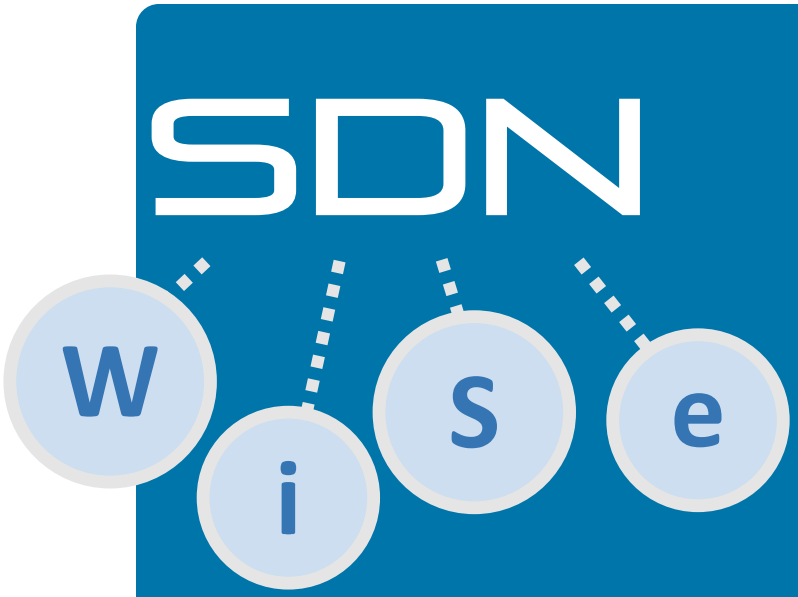
Downsides:

- High solution specialization
- Market fragmentation
- Burden on application developers
- Low reusability

The consequence…



It's not taking off!



Related work

SDN-WISE

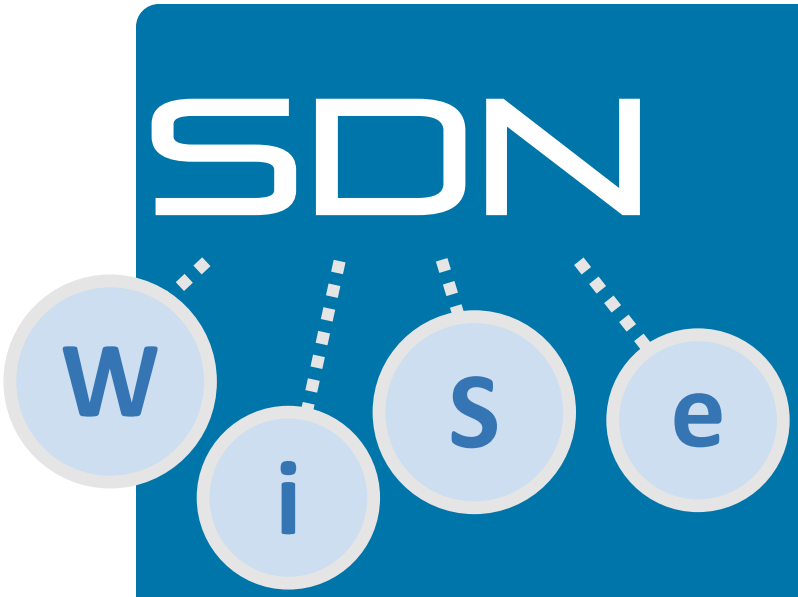
SDN & OpenFlow

- Software Defined Networking (SDN) clearly separates:
  - **Data plane:** run by network Switches
  - **Control plane:** implemented by a software program running on a server (the Controller)
- Modifying the behavior of the network as easy as it is installing a new piece of software on a PC
- **OpenFlow** is the most popular implementation of the SDN paradigm

SDN in WSNs

- Few attempts to extend SDN to WSNs:
  - Software Defined Wireless Networks (SDWN), 2012
  - Sensor OpenFlow, 2012
- Different requirements:

Traditional wired networks	WSNs
Velocity	Efficiency
	Flexibility
	Memory occupancy



Operations

SDN-WISE

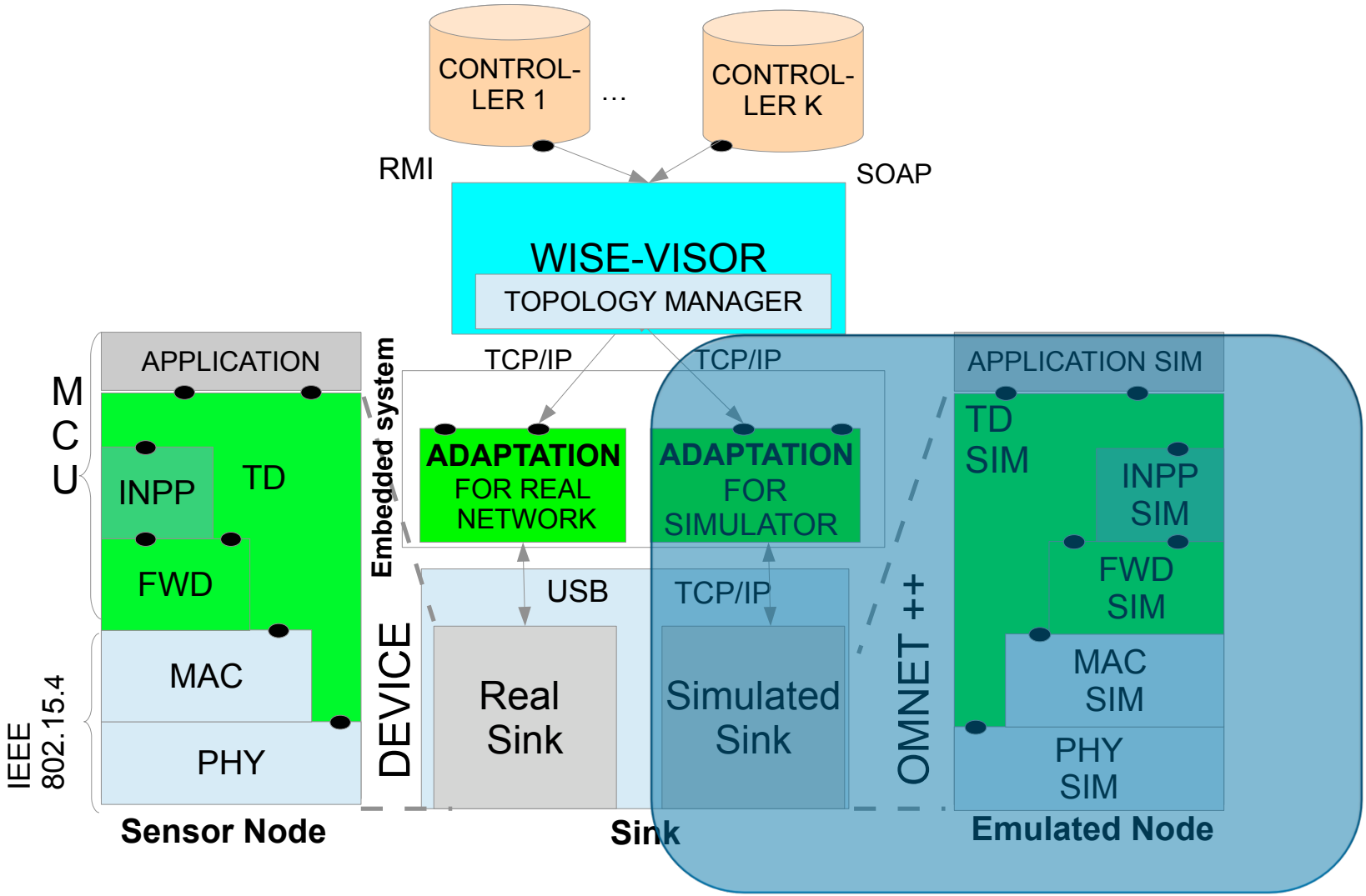
SDN-WISE: Basic concepts

- Directly derived by OpenFlow
- Separation (even physical) between
  - **data plane** (executed by sensor nodes)
  - **control plane** (executed by the Controller)
- When an event (e.g., the arrival of a packet) occurs sensor nodes behave as specified in the WISE Table
- If there is no relevant information in the WISE Table → Ask the Controller
- The Controller replies sending a new entry for the WISE Table
- A simple protocol defined to allow nodes to:
  - Learn the shortest path towards the (closest) sink(s)
  - Discover the neighboring nodes
  - Periodically report local information to the Controller (through the sink)

WISE Table

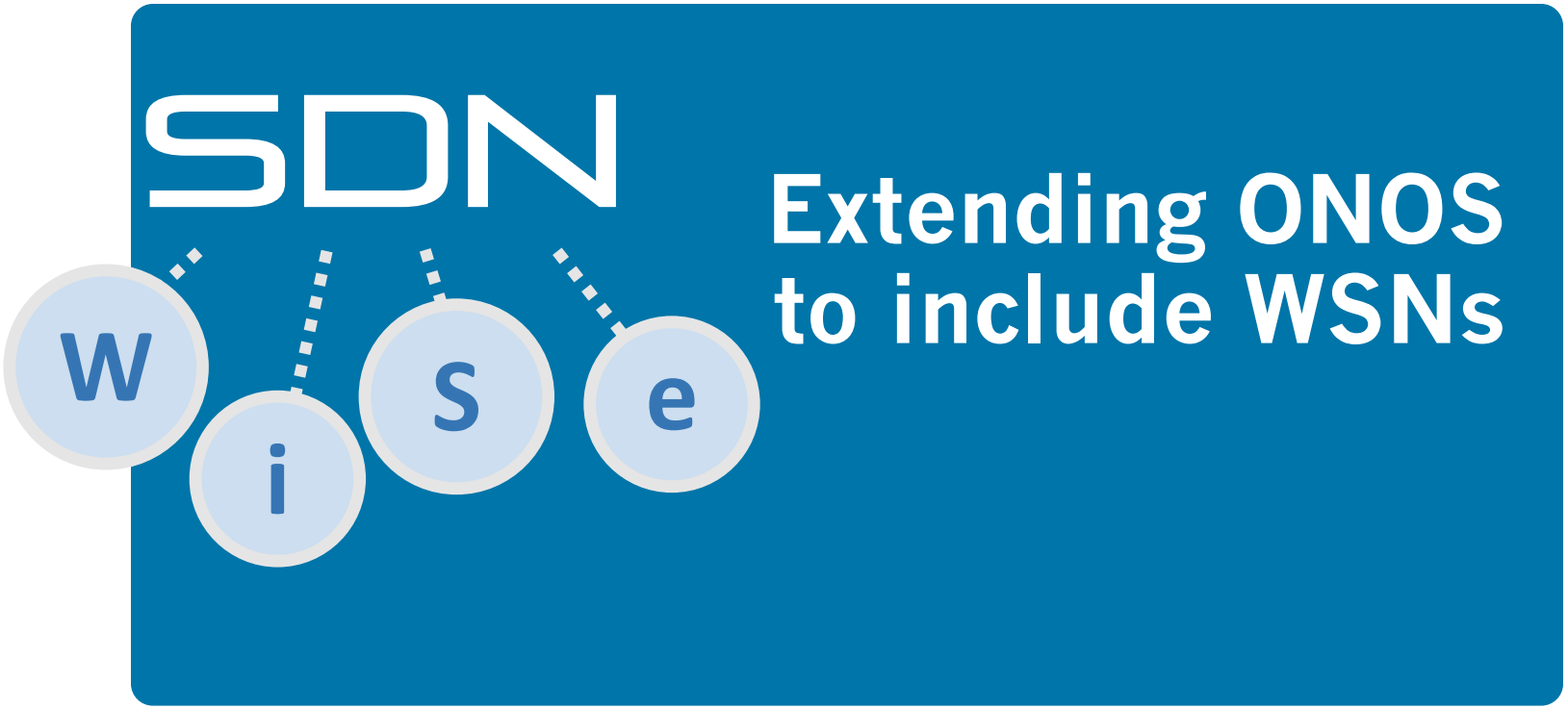
Matching Rule					Matching Rule					Matching Rule					Action					Statistics	
Op.	Size	S	Offset	Value	Op.	Size	S	Offset	Value	Op.	Size	S	Offset	Value	Type	M	S	Offset	Value	TTL	Counter
=	2	0	2	B	>	2	0	10	x_{Thr}	=	1	1	0	0	Modify	1	1	0	1	122	23
=	2	0	2	B	≤	2	0	10	x_{Thr}	=	1	1	0	1	Modify	1	1	0	0	122	120
=	2	0	2	B	-	0	-	-	-	-	0	-	-	-	Forward	0	0	0	D	122	143
=	2	0	2	A	=	1	1	0	0	-	0	-	-	-	Drop	0	0	-	-	100	42
=	2	0	2	A	=	1	1	0	1	-	0	-	-	-	Forward	0	0	0	D	100	32

SDN-WISE Architecture



Major features (compared to OpenFlow)

1. Statefulness → SDN-WISE nodes are Turing complete
2. Flexible definition of rules
3. Support of duty cycles
4. Support of multitenancy (beyond *slicing*)
5. Lots of deployment options and programming languages
6. Integration with simulation environments (OMNET++ & OPNET)

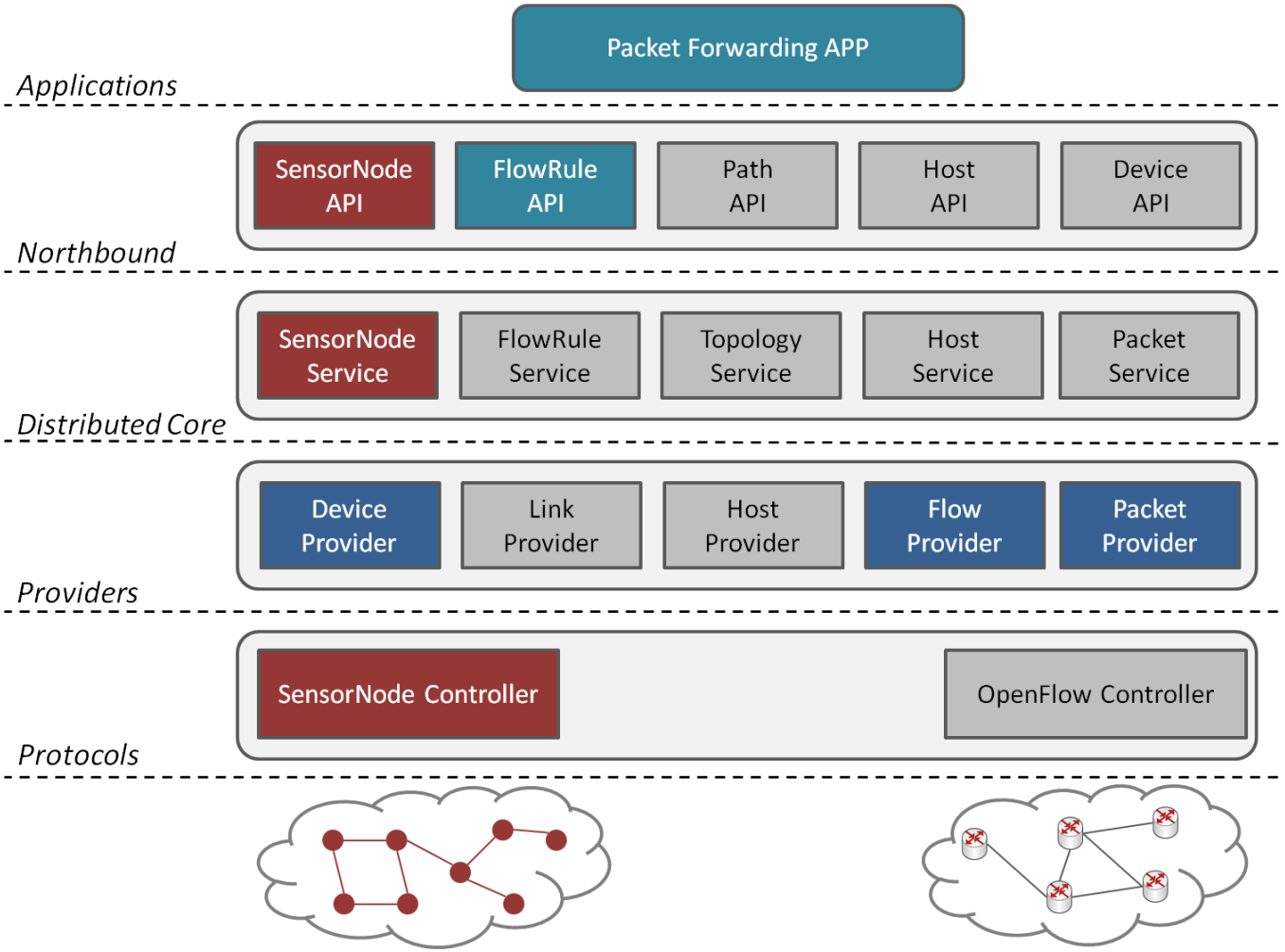


A. C. Anadiotis, L. Galluccio, S. Milardo, G. Morabito and S. Palazzo. "*An Integrated Network Operating System for the Internet of Things: Design, Implementation and Experimentation*". **Under review.**

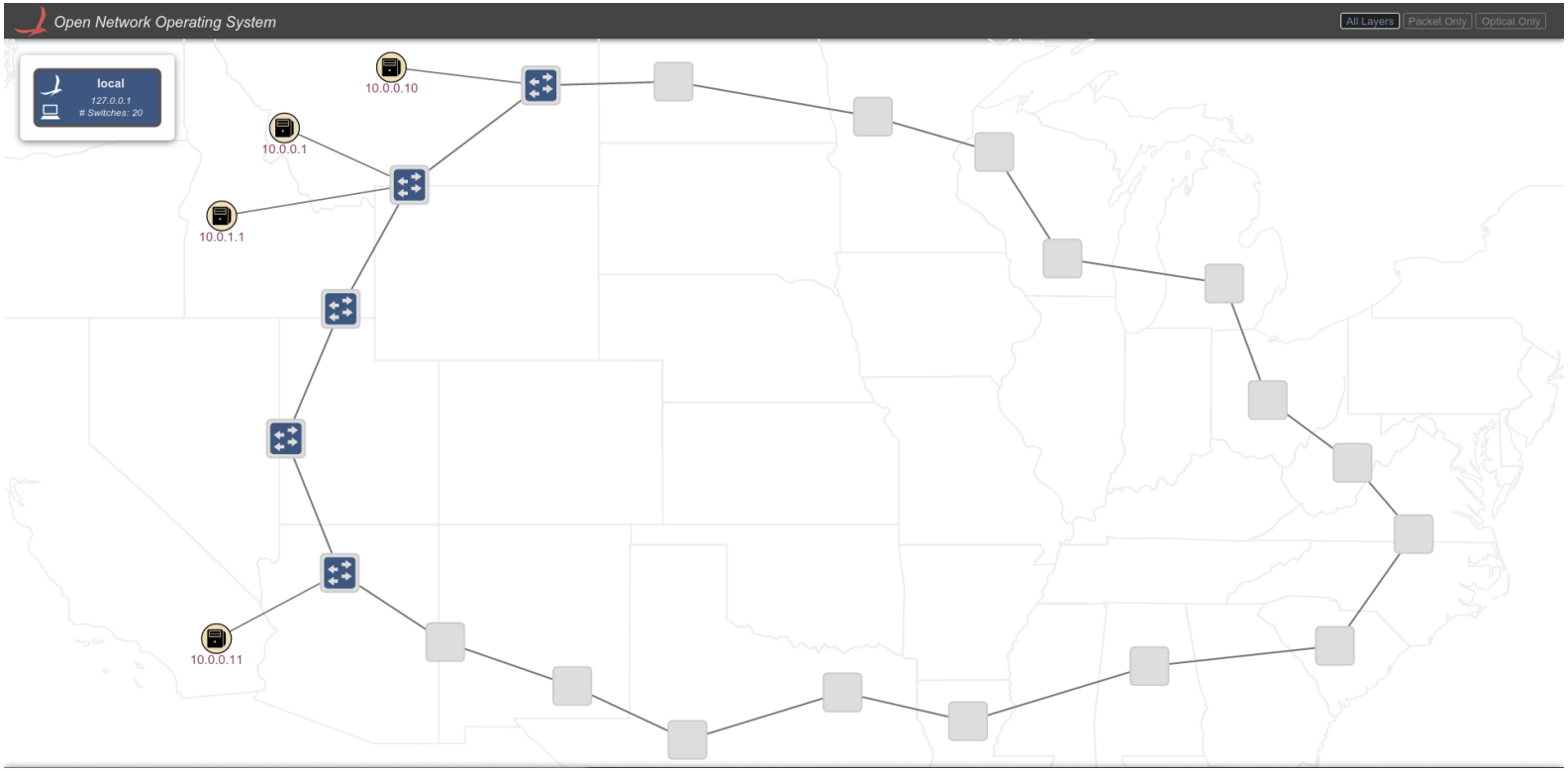
IoT Integration – ONOS Level

- Protocols layer: Implementation of the SDN-WISE driver
- Providers layer: Translation of SDN-WISE-specific details to ONOS low-level abstractions for network resources
- SB API: Unchanged
- Core: Introduction of new services for maintaining WSN-specific information
- NB API: Provision of WSN-specific abstractions, such as SensorNode API

An Integrated NOS for the IoT

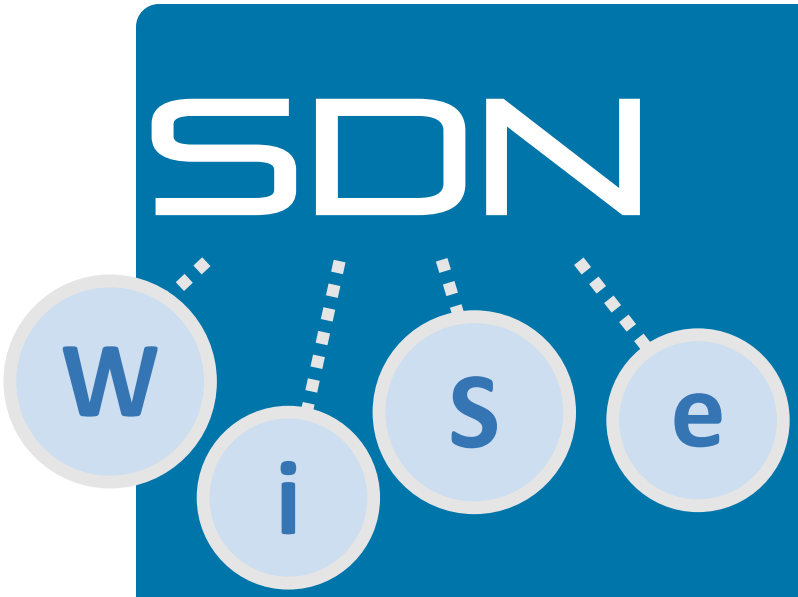


And from a User Point of View



Basic Advantages

- Holistic view of the topology in a device-level rather than a specific protocol level
- Re-use of ONOS components originally designed to support OpenFlow functionality:
  - FlowRules API has been extended to also support SDN-WISE; however the FlowRules service has remained the same
- All information regarding sensors are kept in the Core and any third-party application can access it through ONOS extended NB API



Conclusions and current work

Conclusions

- The IoT needs a network operating system to overcome fragmentation
- Network operating systems have attracted large attention by the R&D community
- However, so far focus has been on wired networks: other IoT components have specific features
- We have taken a few steps in this direction:
  - We have extended ONOS to integrate wireless sensor and actor networks
  - We have developed and tested SDN-WISE
  - We are experimenting the extended ONOS

A challenge for you

- Up to now *networking* has been a bunch of protocols
- Current trend: overcome this approach and create abstractions of network functionalities
- **Is the same abstraction process possible for DSP?**
  - Can you identify a (quasi) complete set of building blocks?
  - Can you define standard APIs?
  - Can you describe complex schemes as a sequence of entries in a table?

Q&A

http://sdn-wise.dieei.unict.it/