Future Network Architectures
Recursive Internet Architecture (RINA)

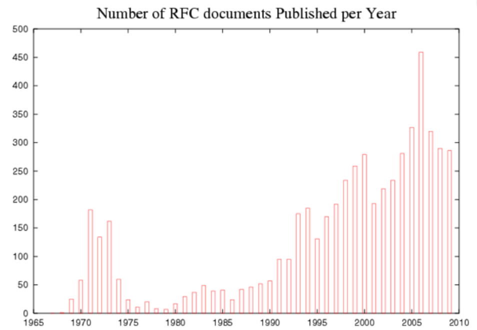
Dimitri Staessens – Ghent Uni. iMinds (BE)

Francesco Salvestrini – Nextworks s.r.l. (IT)

Miquel Tarzan – Fundació i2CAT (ES)

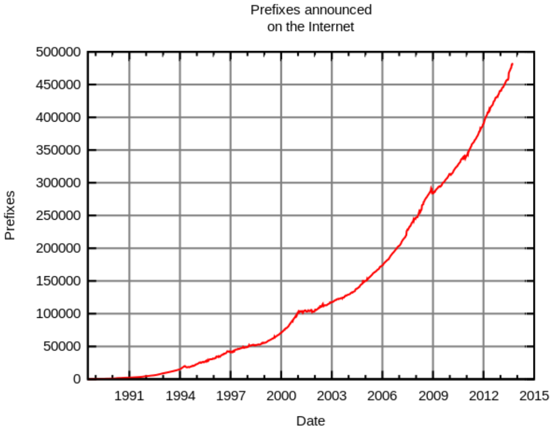
Challenges faced by (Inter)network engineers

- explosion in the complexity of the overall system (hundreds of protocols and thousands of standards documents)



- weak security

- scalability issues with the routing system
  - (IPv6/BGP multihoming)
  - Mobile end-users



- no QoS support

The Internet is a live environment

- ever growing customer base
- ever growing number of devices
- new and more demanding services

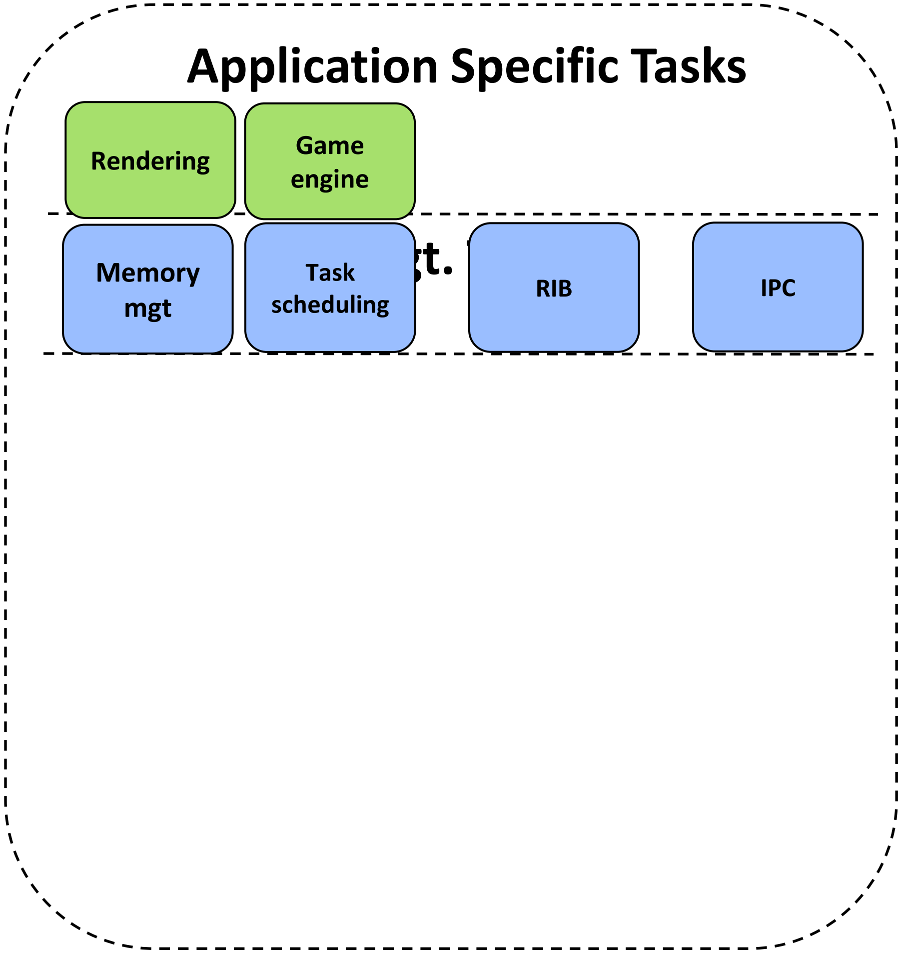
- RAD of services
- fast deployment

- “whac-a-mole” approach to problemsolving

A brief introduction to the Recursive Internet Architecture

RINA

Application Process



Components

- Application specific tasks
- Management tasks

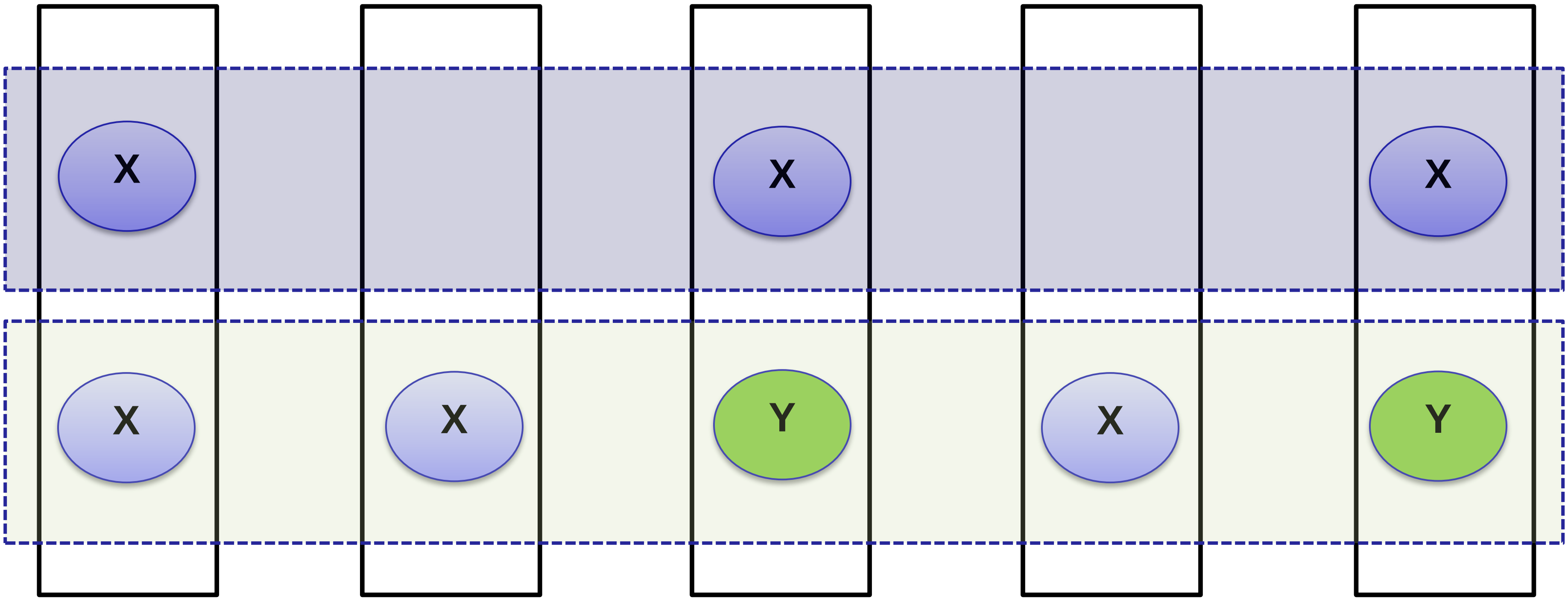
Mechanism

- Static, invariant parts

Policy

- Dynamic, variant parts
- occurs in pairs
  - Sender
  - Receiver

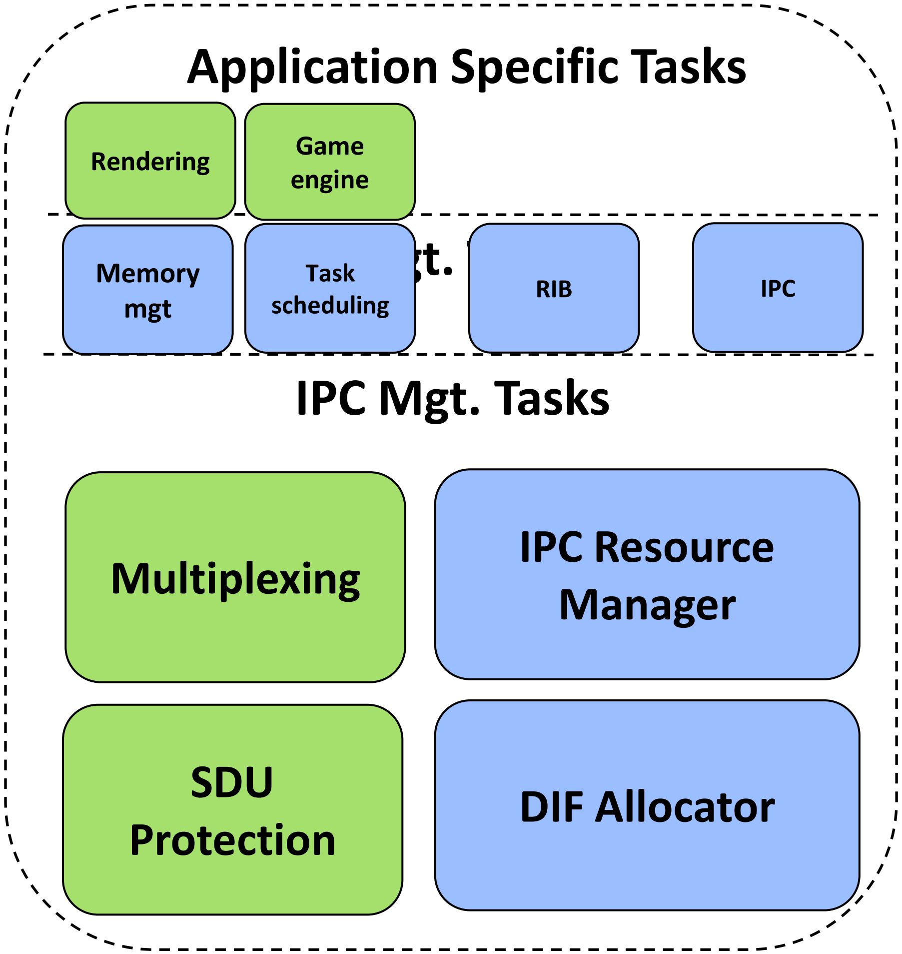
Distributed Application



Processing system: hardware and software capable of executing programs as Application Processes that can coordinate via shared memory ("test and set")

Computing system: a collection of processing systems under the same management domain with no restrictions on connectivity

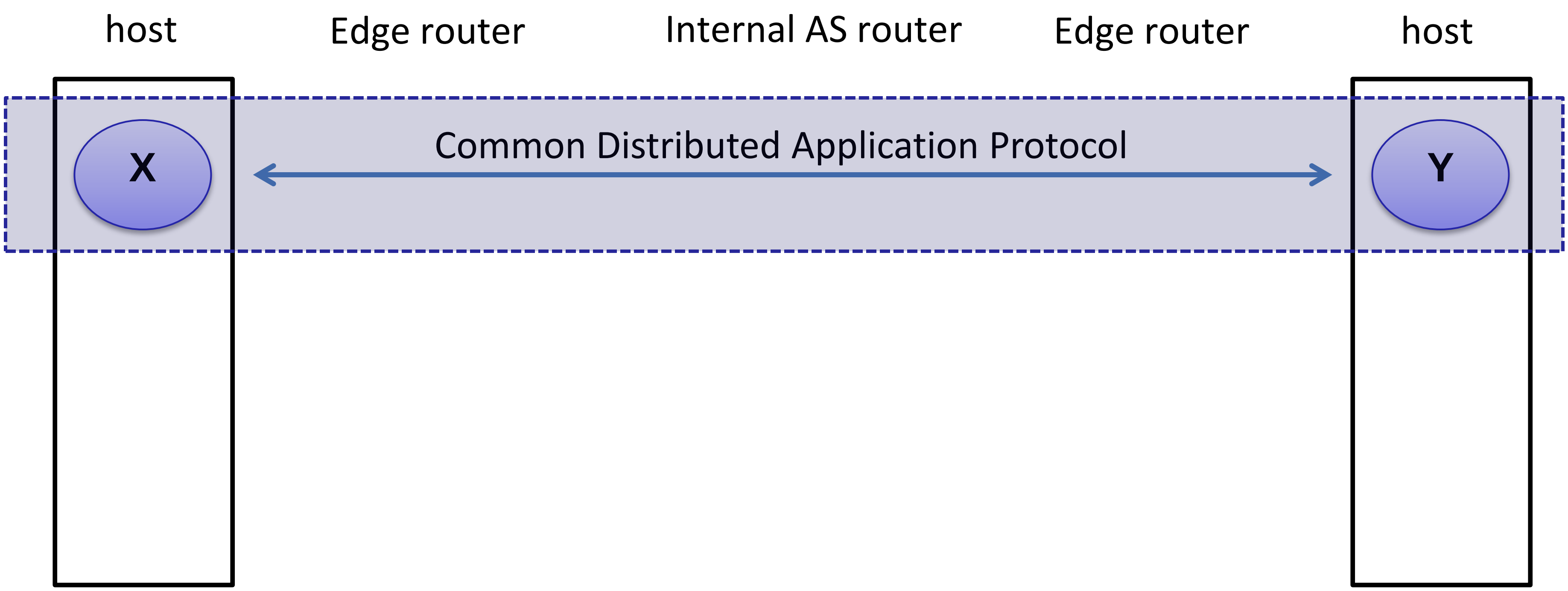
Application Process



Components

- Application specific tasks
- Management tasks
- IPC Management tasks
  - DIF Allocator: Finds remote application processes
  - IRM: manages DA requests
  - Multiplexing: SDU's from different tasks
  - SDU protection: Integrity and security

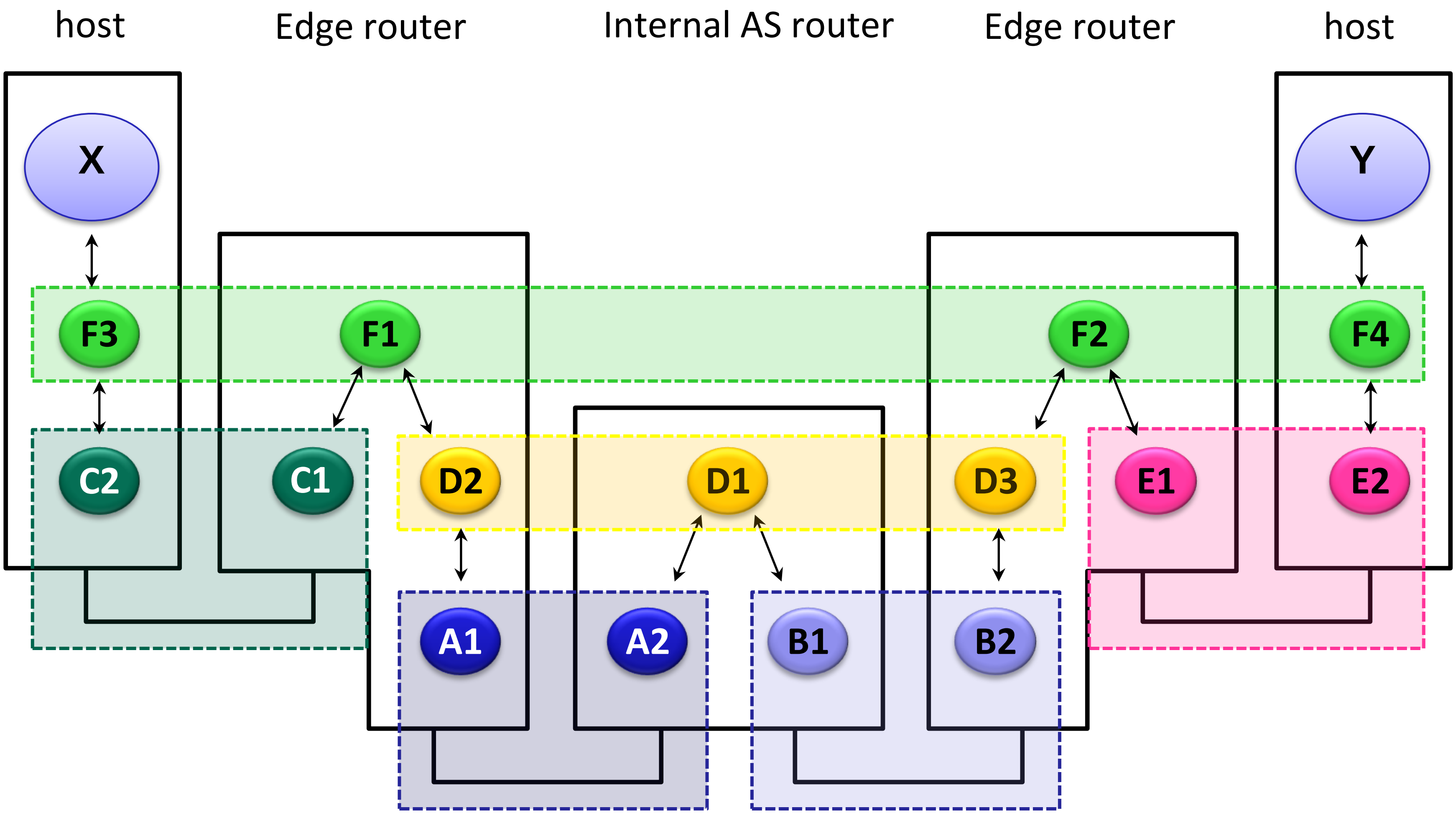
Distributed Applications Provide IPC services



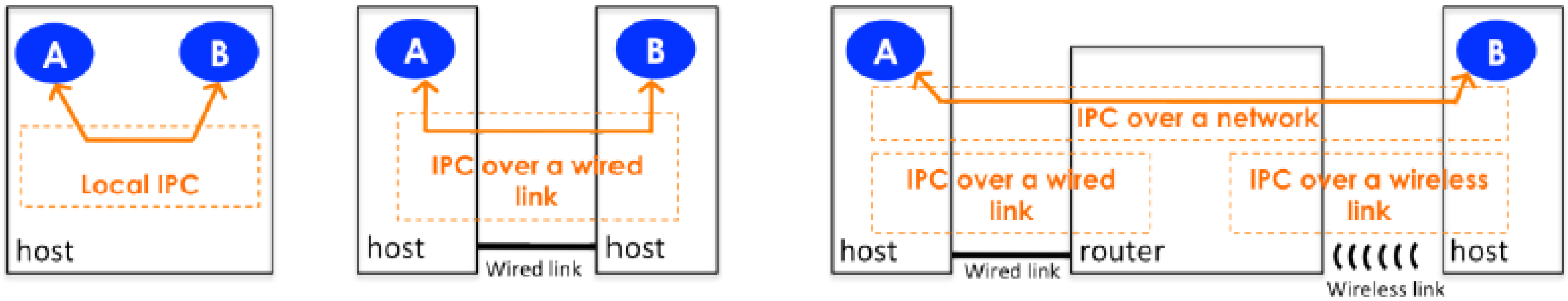
Common Distributed Application Protocol (CDAP)

- Perform operations on RIB objects
  - **Create/Delete**
  - **Read/Write**
  - **Start/Stop**
- But what about different applications?
  - The objects they manipulate
  - Control and sequencing of operations
  - …

Distributed Applications Provide IPC services



Effectively extending the IPC model



IPC API

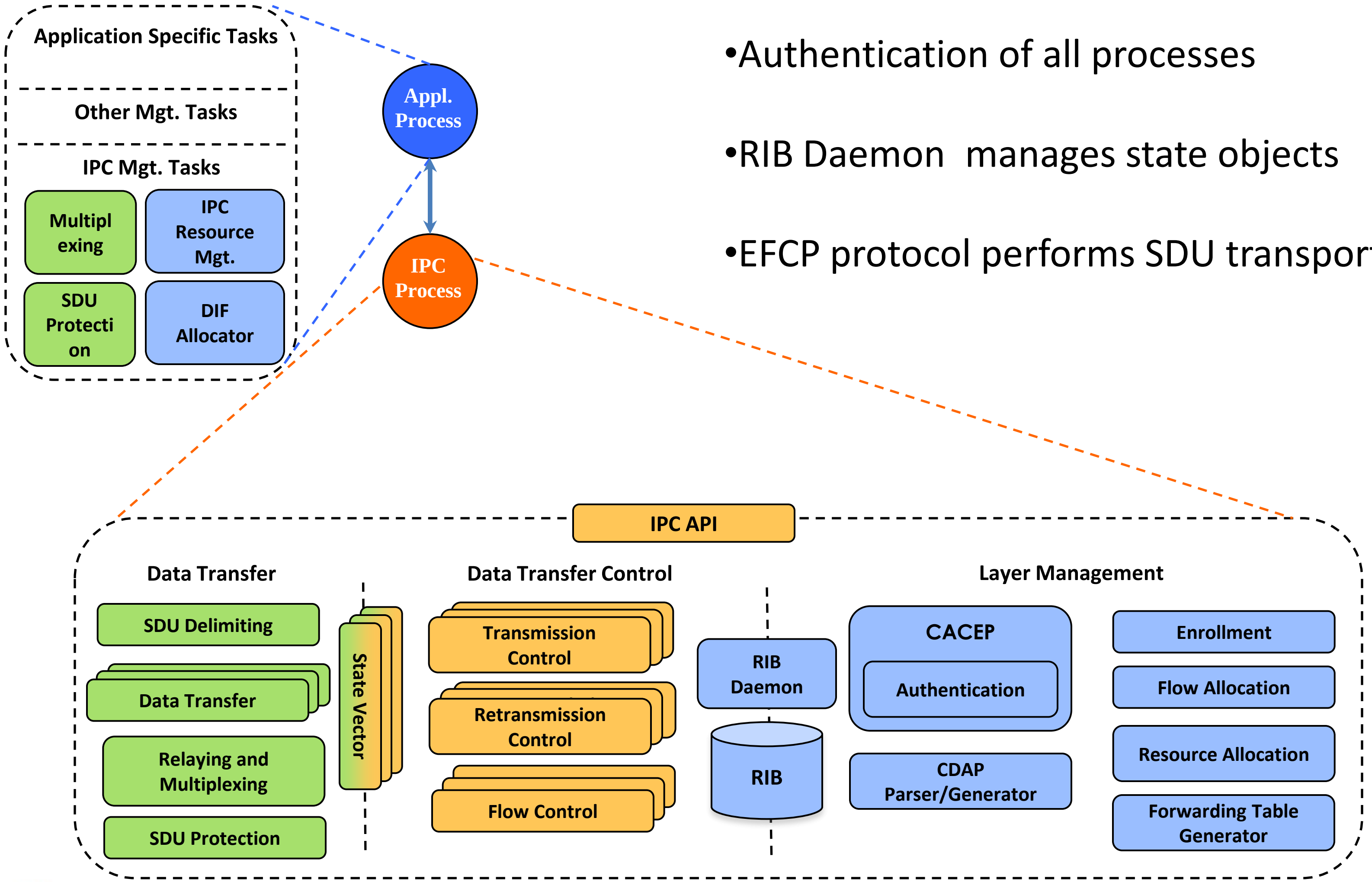
- APs communicate using a port, identified by a portId
- 6 operations:
  - int _registerApp(appName, List<difName>)

  - portId _allocateFlow(destAppName, List<QoSParams>)
  - int _write(portId, sdu)
  - sdu _read(portId)
  - int _deallocate(portId)

  - int _unregisterApp(appName, List<difName>)

- QoSParams are defined in a technology-agnostic way
  - Bandwidth-related, delay, jitter, in-order-delivery, loss rates, …

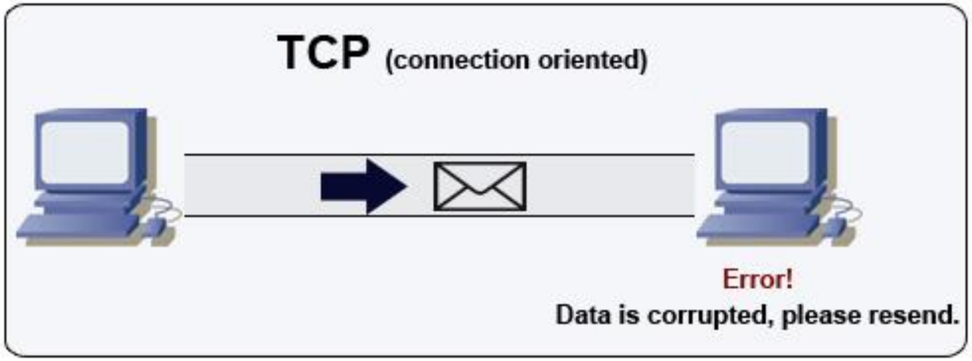
The IPC process

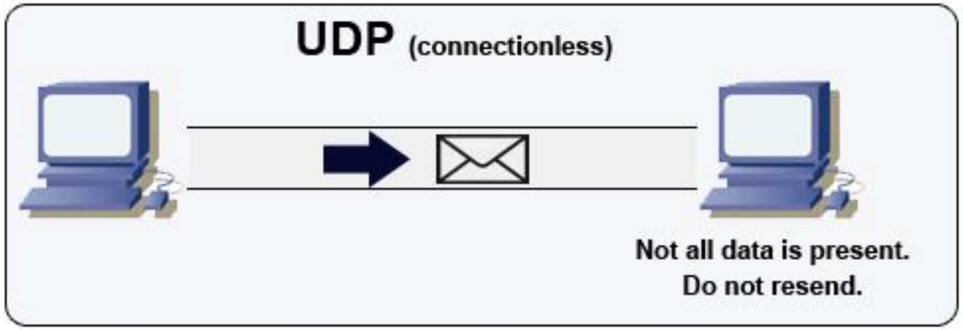


- Authentication of all processes
- RIB Daemon manages state objects
- EFCP protocol performs SDU transport

Error and Flow Control Protocol

- DTP
  - Fragmentation
  - Reassembly
  - Sequencing
  - Concatenation
  - Separation
- DTCP
  - Transmission control
  - Retransmission control
  - Flow control
- Loosely coupled by a state vector
- Based on Delta-t

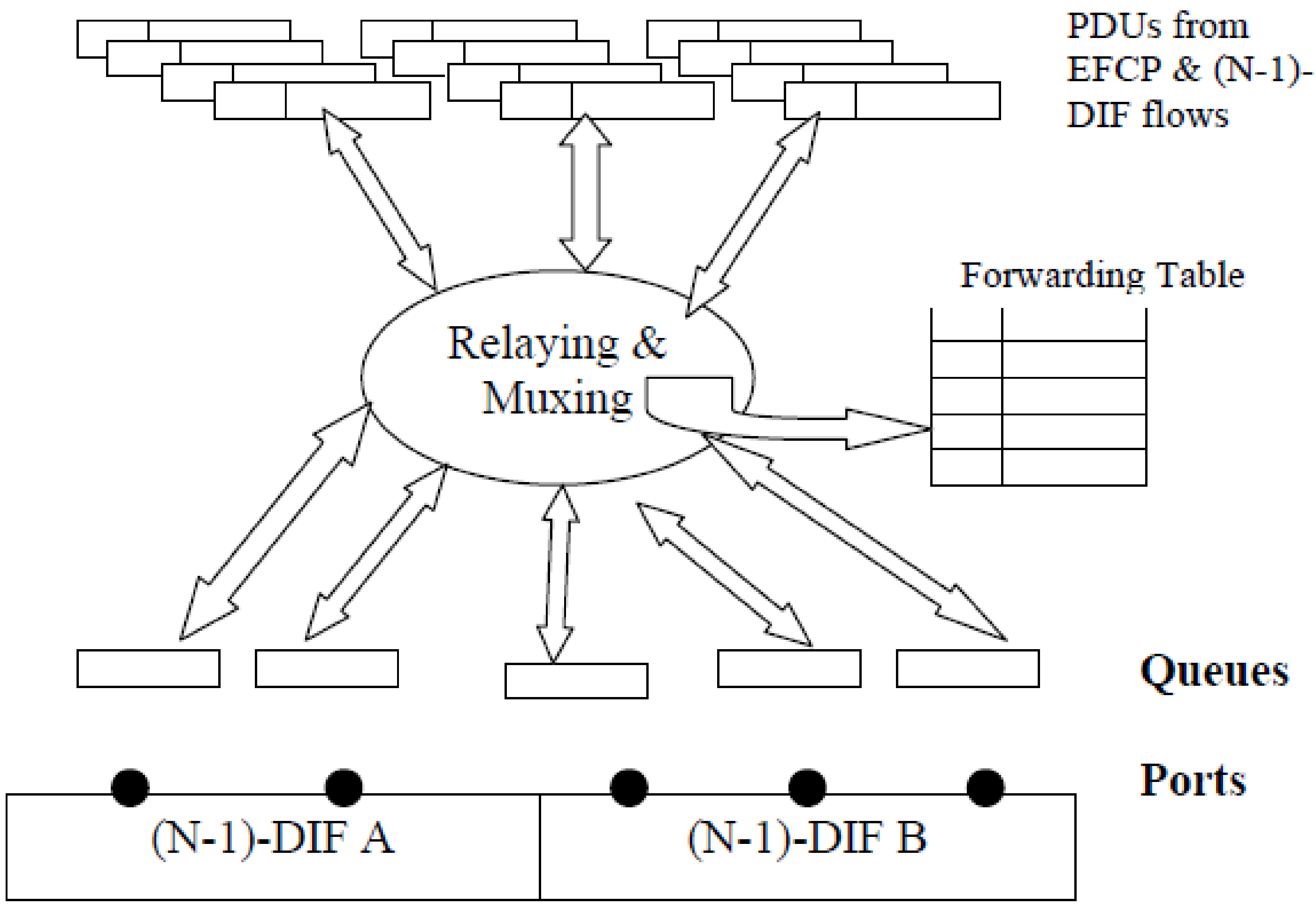




Delta-t (Watson, 1981)

- Developed at L.Livermore labs, unique approach.
  - Assumes all connections exist all the time.
  - keep caches of state on ones with recent activity
- Watson proves that the conditions for distributed synchronization are met if and only if 3 timers are bounded:
  - Maximum Packet Lifetime: **MPL**
  - Maximum number of Retries: **R**
  - Maximum time before Ack: **A**
- That no explicit state synchronization, i.e. hard state, is necessary.
  - SYNs, FINs are unnecessary
- 1981:Watson shows that TCP has all three timers and more.

RMT

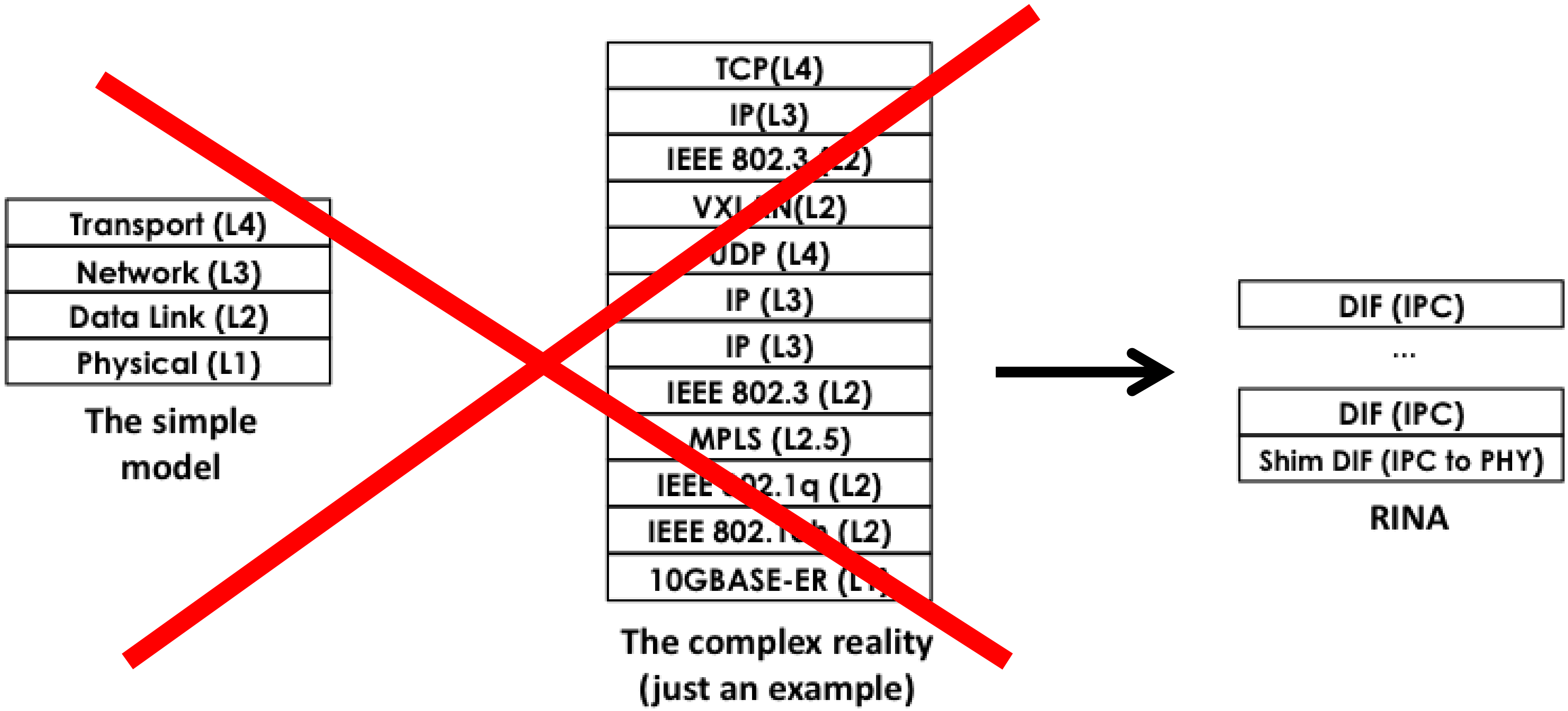


PDUs from
EFCP & (N-1)-
DIF flows
Forwarding Table
Relaying &
Muxing
Queues
Ports
(N-1)-DIF A
(N-1)-DIF B

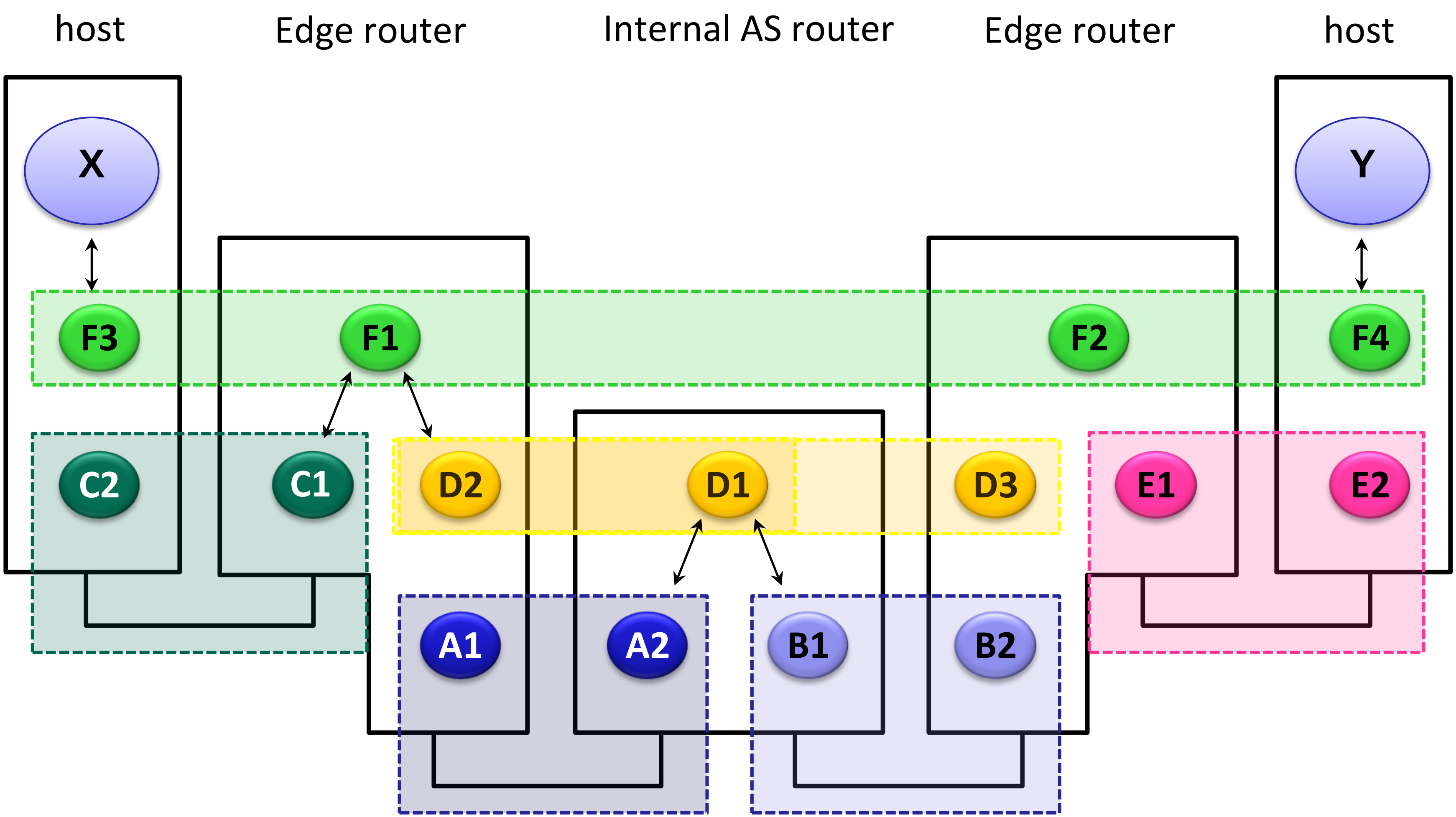
Shims

- Wrap a technology with the IPC API
  - Physical medium
  - Legacy technology
    - Ethernet
    - IP
  - Hypervisors
- Not required to add functionality
- So it's an "incomplete" DIF

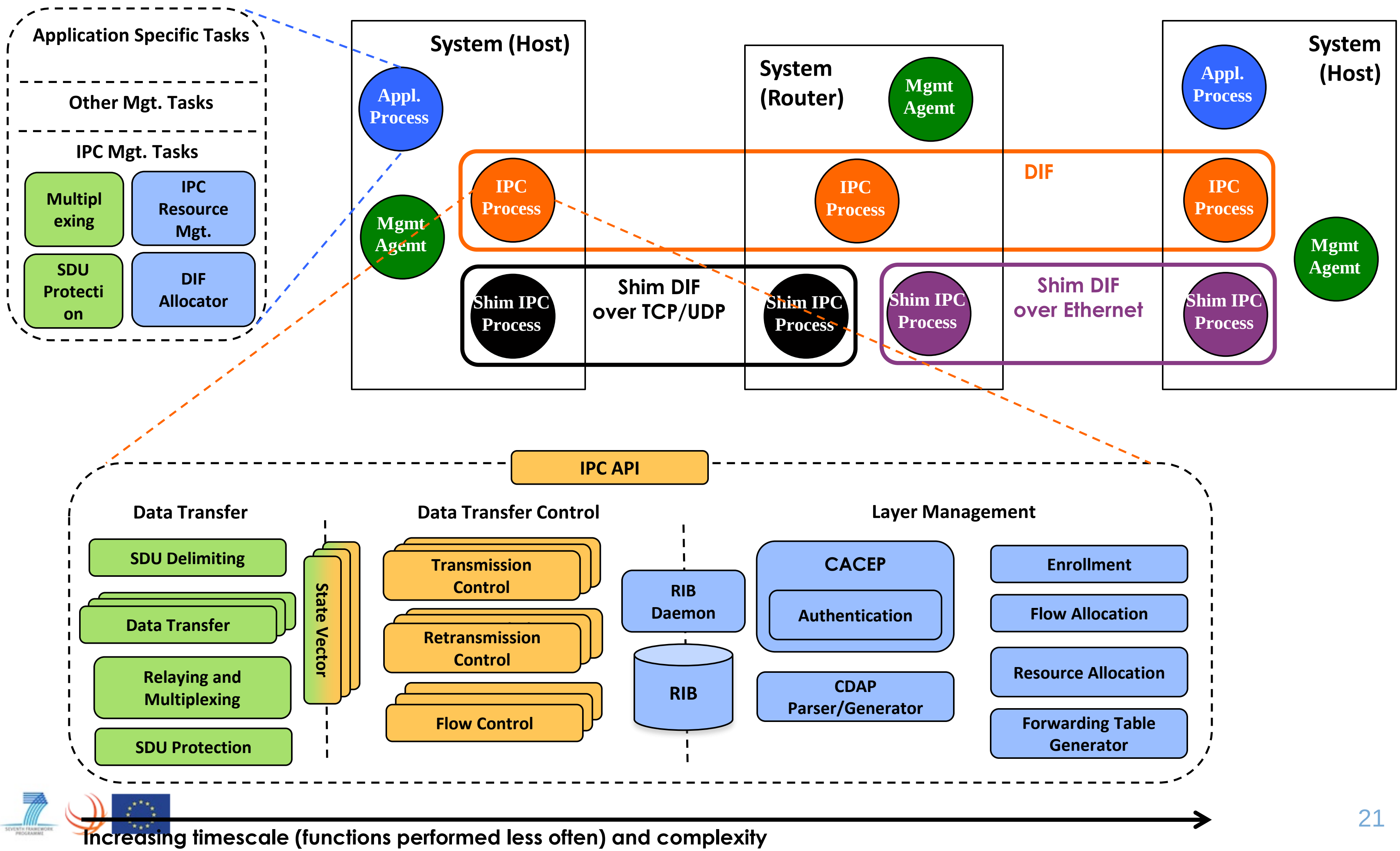
Basic concept of RINA



Bootstrapping a RINA network



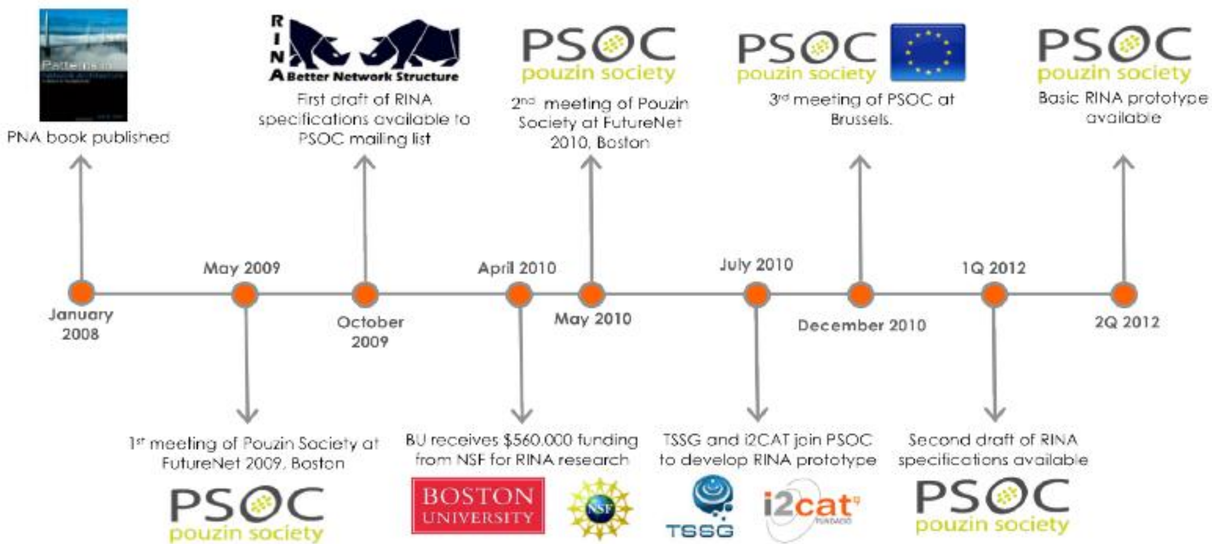
Architectural Model



PROTOTYPES

Pre-existing prototypes

- The first RINA prototype dates back in 2012
- This first implementation was a joint development of BU, i2CAT and WIT-TSSG
  - Targeting the validation of the theory and specs
  - Java based, user-space



- TRIA LLC (US) built another (closed-source) prototype (C based, user-space)

- **Later:**
  - **EC funded FP7 IRATI and PRISTINE projects to advance the research on RINA**
  - **GEANT showed interest in RINA as well, funding the IRINA project**

FP7 IRATI – OVERVIEW

IRATI - Introduction

- FP7 Project – **Jan 2013** to **Dec 2014** (2 years)
- 4 partners
  - [**Research**] Fundació Privada i2CAT (Spain)
  - [**Research**] iMinds VZW(Belgium)
  - [**SME**] Nextworks s.r.l. (Italy)
  - [**Industry**] Interoute (UK/Italy)
  - [**Academia**] Boston University (US)

IRATI – Objectives

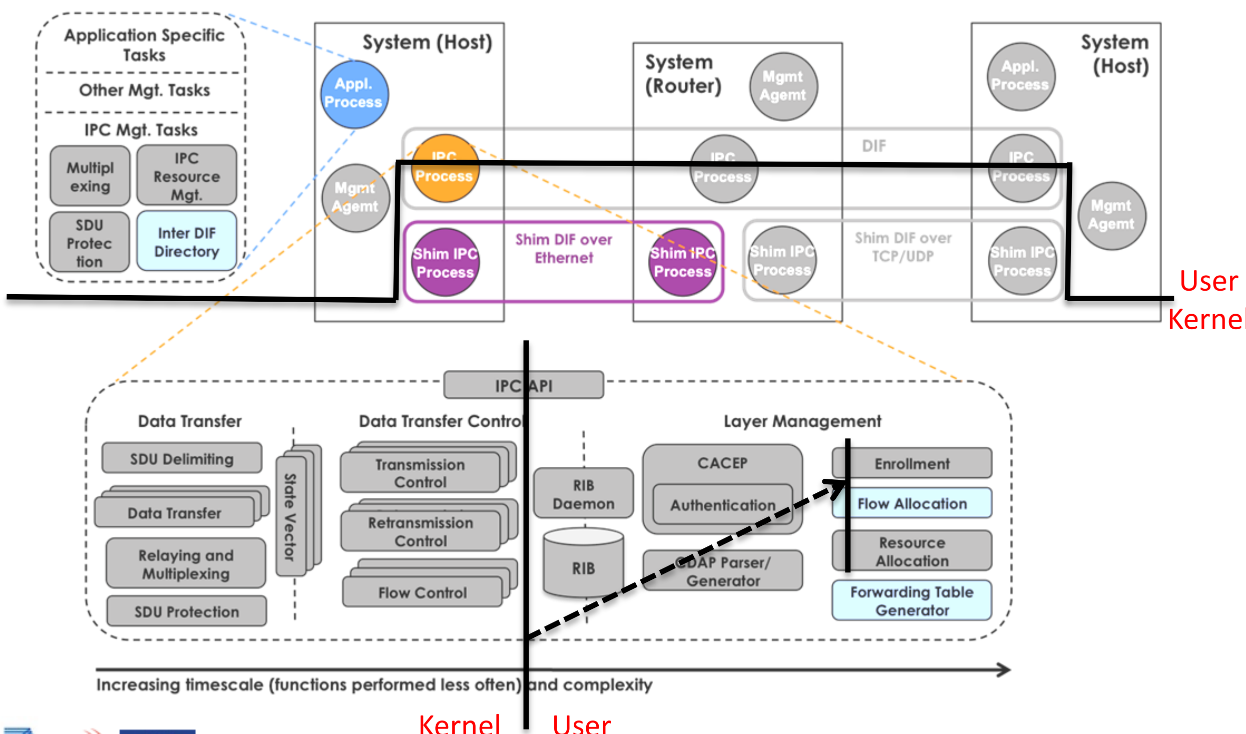
- IRATI' objectives:
  1. Enhancement of the RINA architecture reference model and specifications, focusing on DIFs over Ethernet
  2. **RINA open source prototype over Ethernet for a UNIX-like OS**
  3. Experimental validation of RINA and comparison against TCP/IP
  4. **RINA prototype for Hypervisors**
  5. **Interoperability with the PSOC RINA prototype over UDP/IP**
  6. Provide feedback to OFELIA in regards to the prototyping of a clean slate architecture

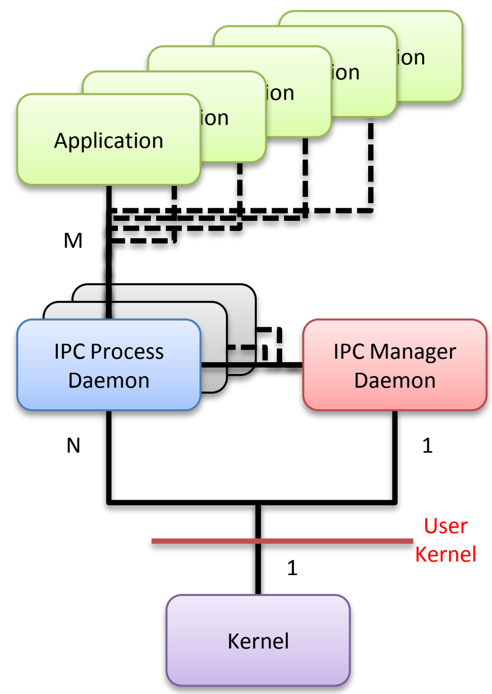
- The project targets the design and implementation of core functionalities **at processing system level**:
  - IPC Process / IPC Manager daemons
  - Transport and management layer

- Software-wise, the project:
  - Built a RINA SW prototype from scratch [**ready**]
  - Release it as FOSS [**end of October**]
  - tries to build up an Open Source community around the prototype

IRATI - Design decisions & fast/slow paths

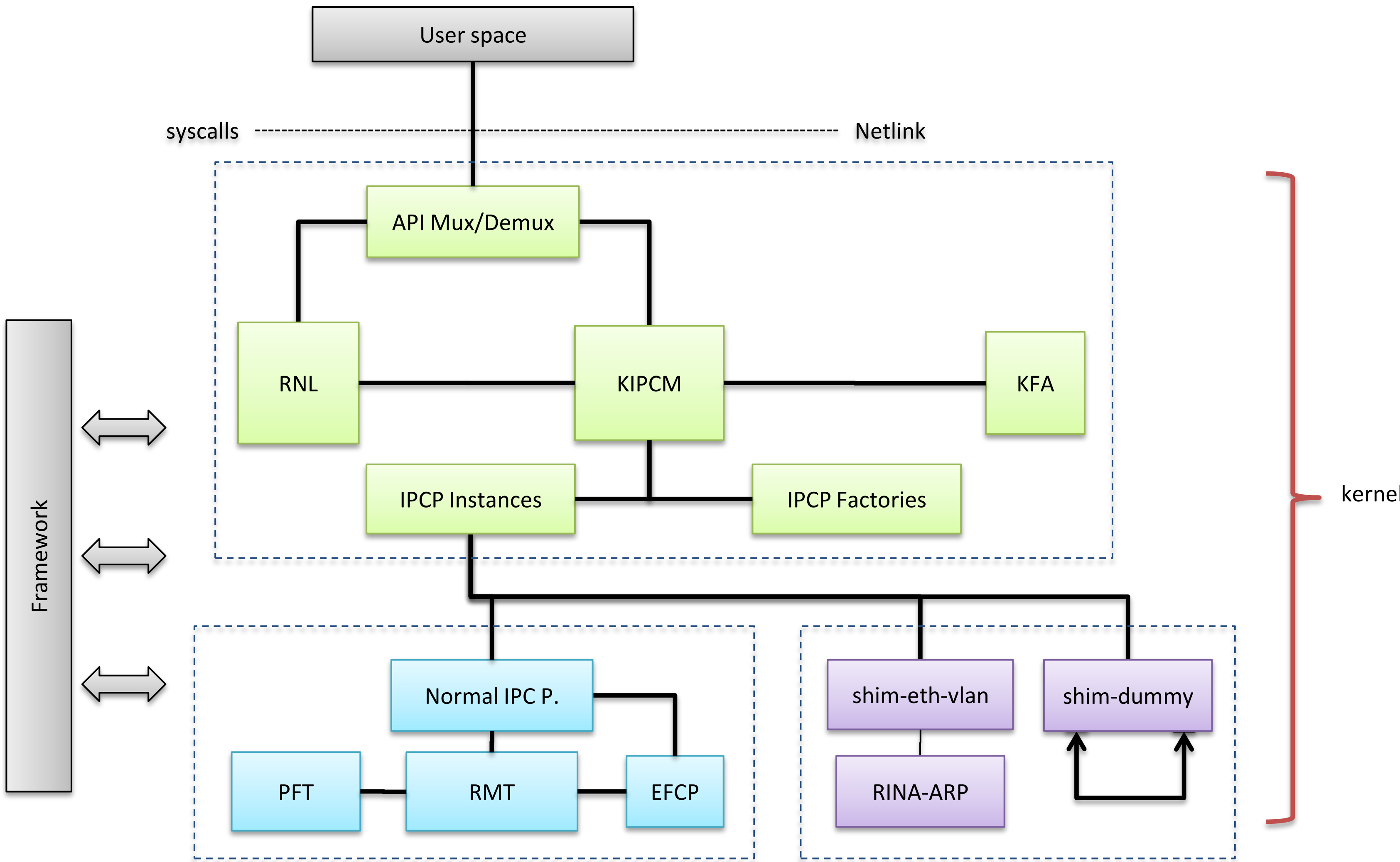
- We split the RINA architecture in different “lanes”
  - **Stringent timings → Fast-path → kernel-space [TRANSPORT LAYERS]**
  - **loose timings → Slow-path → user-space [MANAGEMENT LAYERS]**
- Placing SW components on different lanes, **depending on their timing requirements**





FP7 IRATI – KERNEL SPACE (THE TRANSPORT LAYERS)

The kernel-space HL SW arch

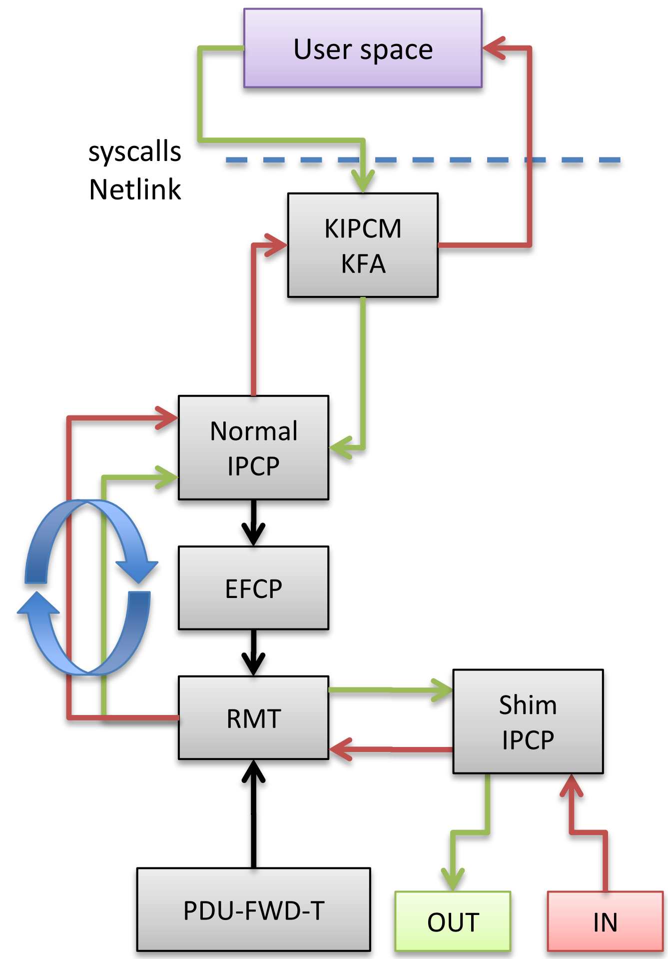


KIPCM & KFA

- **The KIPCM**
  - **Manages the lifecycle the IPC Processes and KFA**
  - **Abstract IPC Process instances**
    - Same API for all the IPC Processes
      - Regardless the type
    - maps: ipc-process-id → ipc-process-instance

- **The KFA**
  - **Manages ports and flows**
    - **Ports**
      - Flow handler
      - Port ID Manager
    - **Flows**
      - maps: port-id → ipc-process-instance

- **KIPCM + KFA**
  - Decouple user-interface from IPC Processes
  - **Are the Initial point where "recursion" is transformed into "iteration"**



IPC Process Instances & Factories

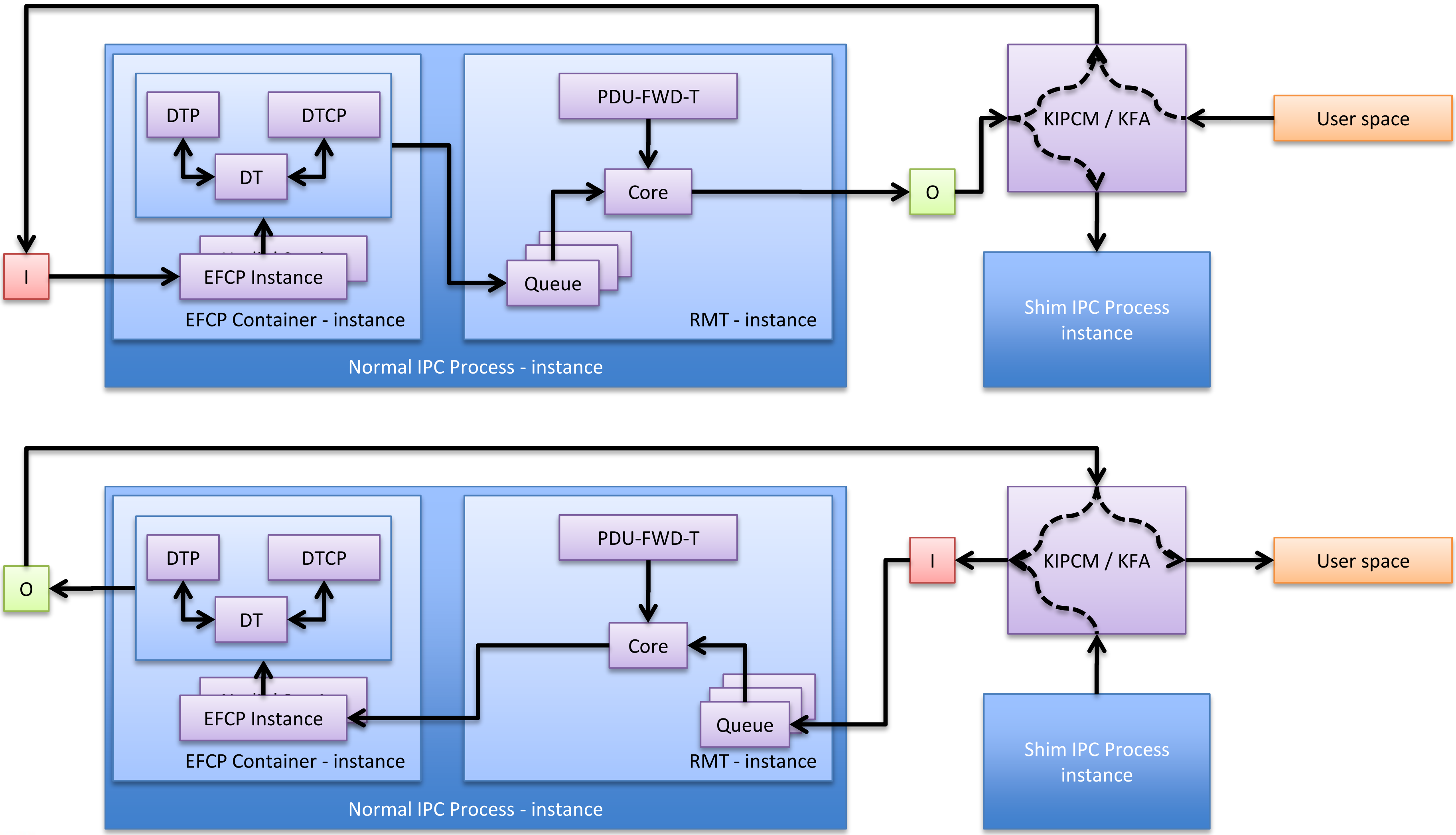
- **There are two "major" types of IPC Processes:**
  - Normal (EFCP + RMT …)
  - Shims
- **The interface is the same** regardless of their type
- **Each IPC Process implements its "core" details:**
  - Normal IPC Processes:
    - **The stack provides the implementation for all of them**
  - Shim IPC Processes:
    - **Each Shim IPC Processes provide its implementation**

- IPC Process instances are created via "templates", instantiated by IPC Process Factories (OOD/OOP)

IRATI – The Shims IPCs

- **The shims are the "lowest" components in the stack**
- **They have one interface**:
  - **North-Bound**: The IPC API (as all the other IPC Processes)
- They wrap the underlying technology (e.g. 802.1Q)
- There are currently 4 shims available:
  - **shim-eth-vlan**:
    - As defined in the spec, runs over 802.1Q
  - **shim-hv:**
    - Targets hypervisor-based environments (KVM/Qemu and Xen)
    - Allows removing unnecessary layering commonly used in traditional VM/HV environments (e.g. bridges, virtual-NICs), such layers ease the adoption BUT:
      - Reduce performances
      - Increase maintenance costs
  - **shim-tcp-udp**
    - Targets RINA over TCP/UDP
  - **shim-dummy**:
    - Not a "real" shim, it's for debugging/testing purposes
    - It's a sort of "loopback" shim (i.e. confined into a single host)

(S | P)DUs workflows



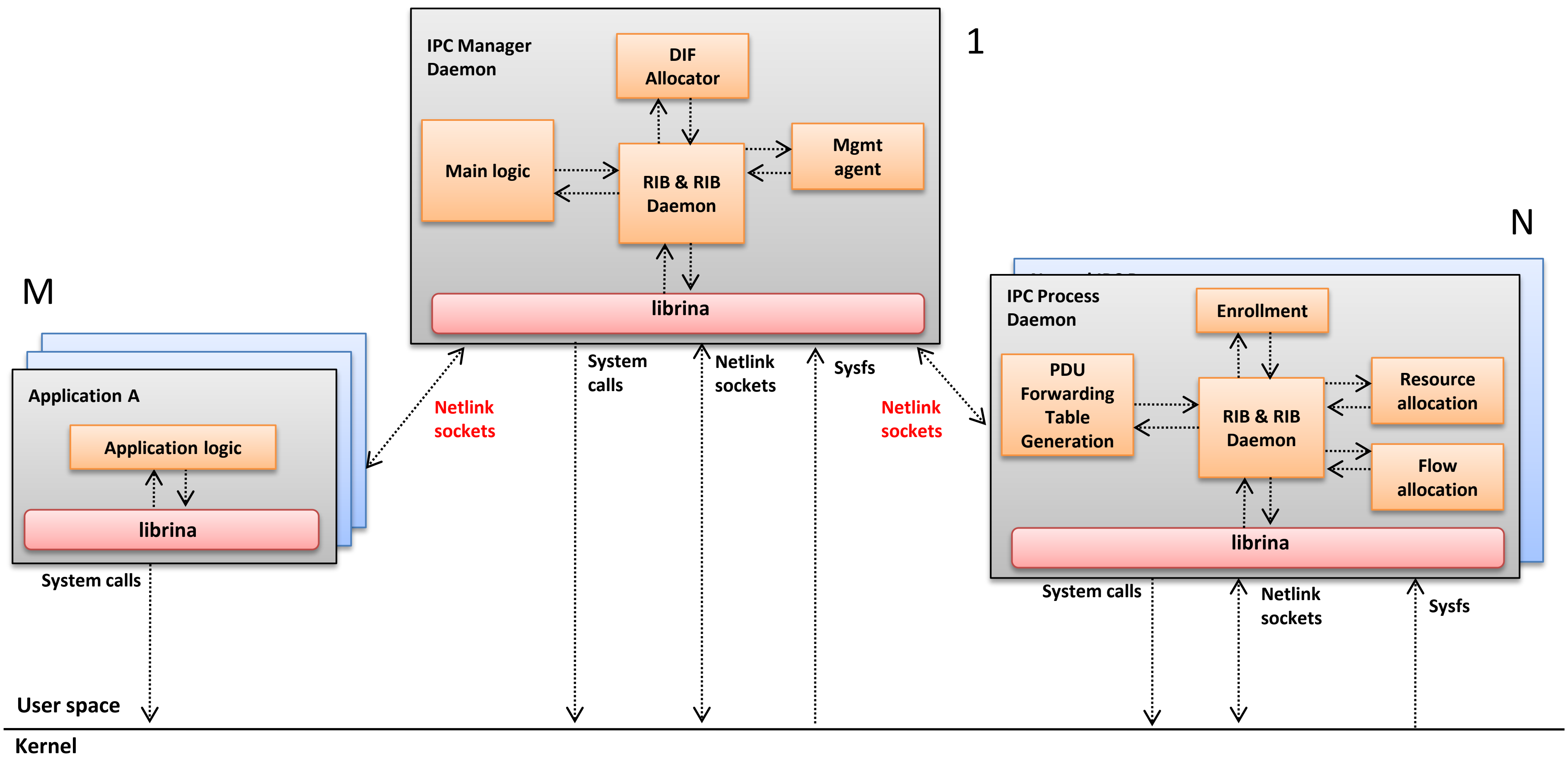
FP7 IRATI – USER SPACE (LAYER MANAGEMENT)

IPC Process & IPC Manager daemons

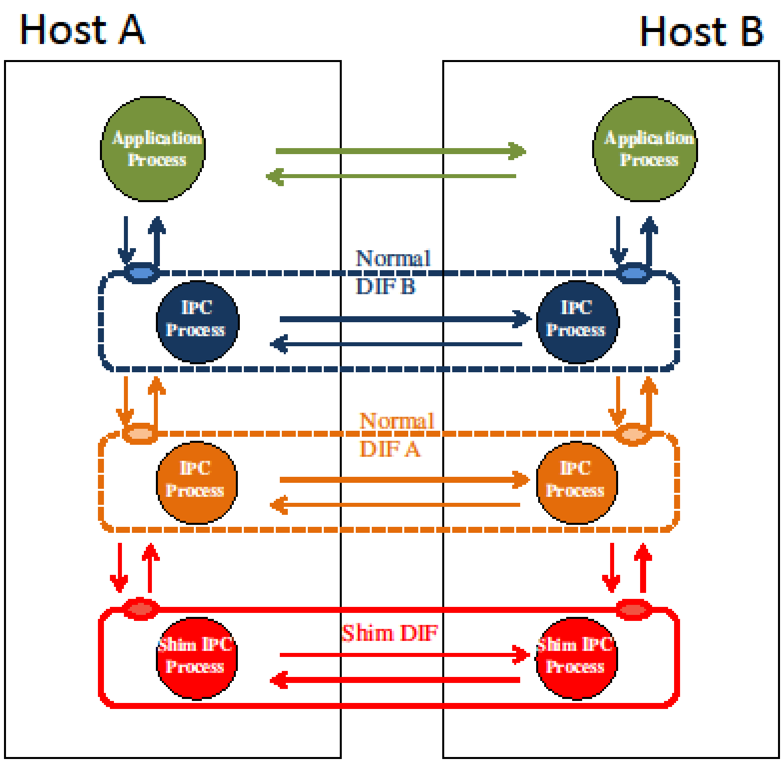
- **IPC Manager Daemon (as an OS process)**
  - **Manages the IPC Processes lifecycle**
  - **Broker between applications and IPC Processes**
  - **Local management agent**
  - **DIF Allocator client** (to search for applications not available through local DIFs)

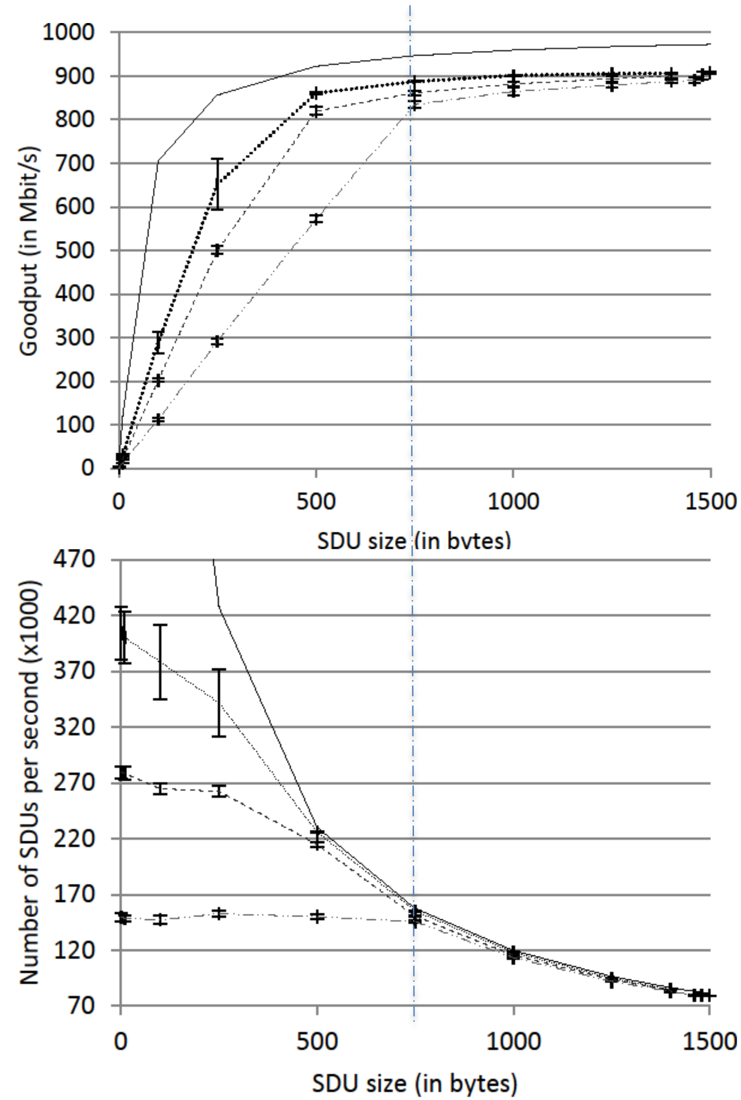
- **IPC Process Daemon (as an OS process)**
  - **Layer Management components of the IPC Process**
    - RIB Daemon, RIB
    - CDAP parsers/generators
    - CACEP
    - Enrollment
    - Flow Allocation
    - Resource Allocation
    - PDU Forwarding Table Generation
    - Security Management
- **These daemons:**
  - **Run as separate OS processes**
  - **Rely on a common framework (librina)**

User space HL SW arch



Prototype’ performances





Theoretical maximum
Shim IPC process for 802.1Q
Normal IPC process over the shim IPC process for 802.1Q
Normal IPC process over a normal IPC proces over the shim IPC process for 802.1Q

FP7 PRISTINE – OVERVIEW

PRISTINE - Intro

- FP7 Project
  - Starts **Jan 2014**, ends **Dec 2016** (3 years)
  - 15 Partners (**Research**, **SMEs** and **Industry**)

PRISTINE - Objectives

- Design and implement the innovative internals of the RINA architecture that include the **programmable functions** for:
  - **security** of content and application processes,
  - supporting **QoS and congestion control** in aggregated levels, providing **protection and resilience**, facilitating more **efficient topological routing**
  - **multi-layer management** for handling configuration, performance and security.
- **Demonstrate** the applicability and benefits of this approach and its built-in functions in **three use-cases**
  - **Datacenter, Distributed cloud, Carrier network**

PRISTINE – (SW) Outcomes

- **PRISTINE will be:**
  - **Developing a Software Development Kit (SDK):**
    - From the IRATI prototype sources-base
    - Enable to customize the behaviour of the DIFs
    - Allow to plug-in/out policies dynamically
      - Policies in the transport layer
      - Policies in the management layers
  - **Developing innovative policies**
    - New policies, defined in the project will be: developed, tested and integrated
  - **Developing the first DIF Management System (DMS)**
    - Manages the DIFs in multiple Processing Systems (same administrative domain)
  - **IPC Process and IPC Manager daemons enhancements**
    - Management Agent
      - Coordinates the loading/installation/removal/unloading of policies in the Processing System
    - Reliability aspects
    - Measurements
    - Performances
    - …

GEANT3+ IRINA – OVERVIEW

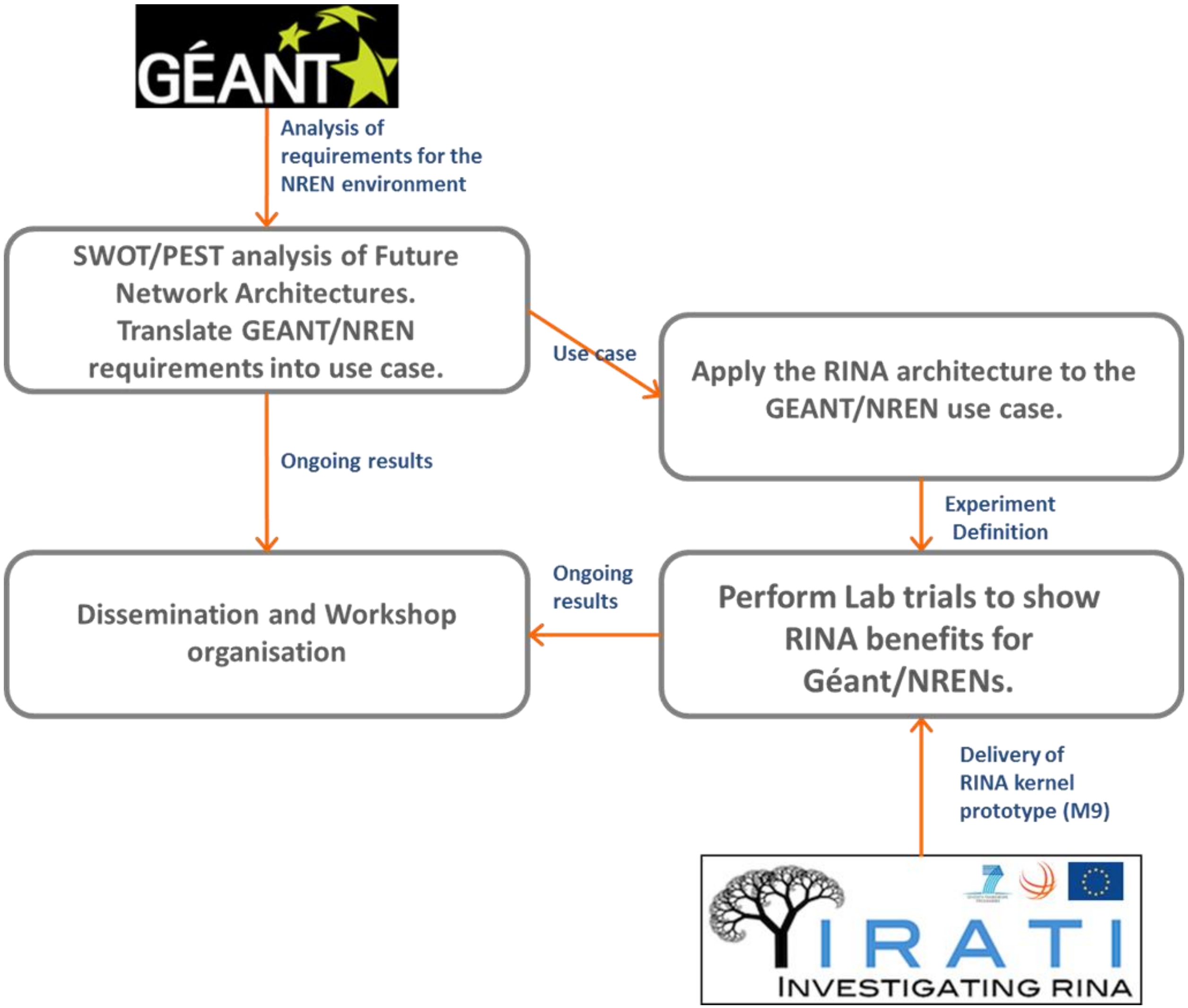
IRINA - Intro

- **I**nvestigating **RI**NA as the next generation GEANT and **N**REN network **a**rchitecture (IRINA)
- GEANT3+ project
  - Starts **Oct 2013**, ends **March 2015** (18 months)
- 4 Partners:
  - [**Research**] iMinds VZW(Belgium)
  - [**Research**] Fundació Privada i2CAT (Spain)
  - [**Research**] Waterford Institute of Technology – Telecommunications Software & Systems Group (Ireland)
  - [**SME**] Nextworks s.r.l. (Italy)

IRINA - Objectives

- Proposes to study RINA as the foundation of the next generation NREN and GEANT network architectures.
- Targets the following goals:
  - **Make a comparative study of RINA vs. the current networking SoTA** and the most relevant clean-slate architectures under research.
  - **Perform a use-case study** of how RINA could be better used in the NREN scenario
    - considering different deployment options, and illustrating the benefits that RINA can bring in terms of multi-homing, mobility, quality of service, programmability, virtualization and network management.
  - **Showcase a lab trial of the use-case study**
    - Utilizing a customized version of the FP7 IRATI stack, and the experimental facilities contributed by the project partners.
  - **Involve the NREN and GEANT community** in the different steps of the project, in order to discuss the project approach, the findings and to get valuable feedback.
    - The organization of a network architectures workshop in cooperation with GN3+ JRA1 will be a key instrument to achieve this objective.

IRINA – Overview/Objectives



STANDARDISATION

ISO and RINA

- **RINA addresses concerns identified by FN:**

  "Even though the current Internet is such an essential infrastructure, we see that there are many concerns about the following technical aspects of the current Internet, including IP based networks: **scalability**, ubiquity, **security**, **robustness**, **mobility**, **heterogeneity**, **Quality of Service** (QoS), **re-configurability**, context-awareness, **manageability**, economics, etc."

RINA specifications status (I)

- **RINA IPC Specification Reference Model**
  - **Basic concept of distributed systems**
  - **Distributed applications**
  - **Distributed InterProcess Communication (IPC API)**
- **DAF Base Specifications**
  - **Common Application Establishment Phase**
  - **Common Distributed Application Protocol**
  - **IPC Resource Manager Specification**
  - **DIF Allocator Specification**

RINA specifications status (II)

- **DIF Base Specifications**
  - Data Transfer Service Definition
  - Specification template for a Generic DIF Delimiting module
  - Error and Flow Control Protocol Specification (DTP + DTCP)
  - Relaying and Multiplexing Task Specification
  - Specification Template for a DIF SDU Protection Module
  - Specification Template for a Generic DIF SDU Protection Module
  - Basic Enrollment Specification
  - Flow Allocator Specification

RINA specifications status (and III)

- **Policy Specifications**
  - CRC 16 SDU protection module
  - DIF HDLC like SDU protection
  - DIF TCP UDP like SDU protection module
  - Retransmission timer expiry policy for a TCP like DIF
  - Round Trip Time (RTT) estimator policy for a TCP like DIF
  - Delimiting module for operating over the public Internet
  - Delimiting module for demo DIF
- **Pro-forma for policy specifications**
  - **No need to standardise everything**

Shims

- **Shim-eth-vlan**
- **Shim-tcp-udp**
- **Shim-hv**

Upcoming workshops

- Globecom Workshop "Alternatives to TCP/IP"
  - 8-12 December, Austin TX US
- RINA workshop
  - 28-29 January 2015, Ghent Belgium
- TERENA TNC 2015
  - 15-18 June, Porto, Portugal
- Summer school ~ August 2015