RINA: Update on Research and Prototyping Activities

Global Future Internet Summit

September 14th, 2012

Eduard Grasa

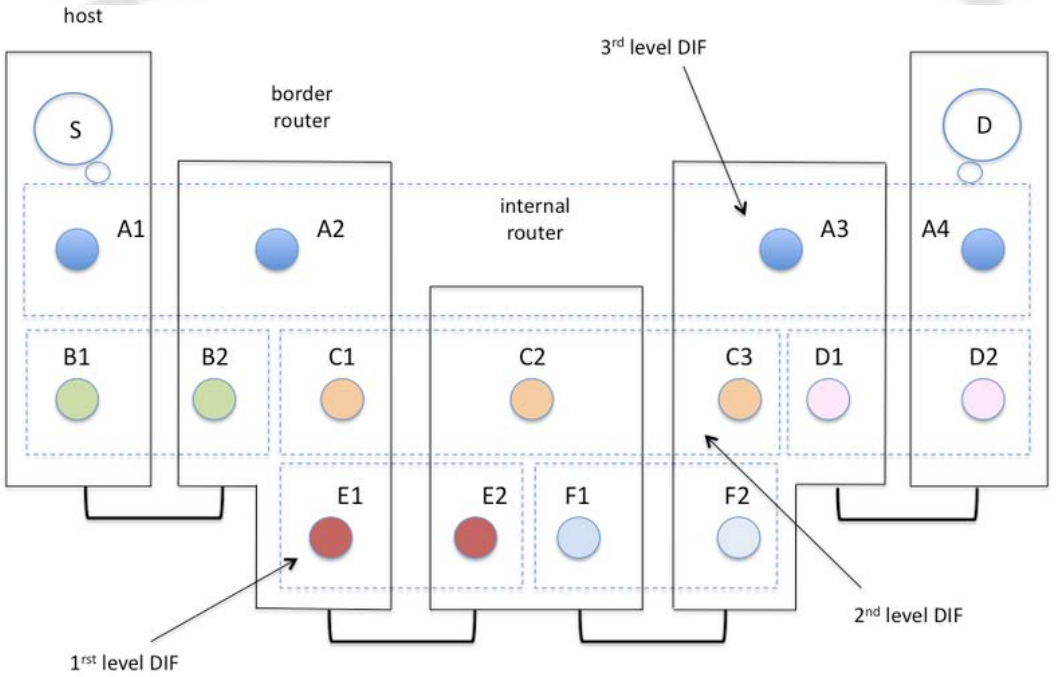
Research Manager @ DANA

Fundació i2CAT

Outline

- **Quick Introduction to RINA**
- Java prototype over IP
- Inter DIF Directory
- RINA Security Assessment
- FP7 IRATI Project: Kernel prototype over Ethernet

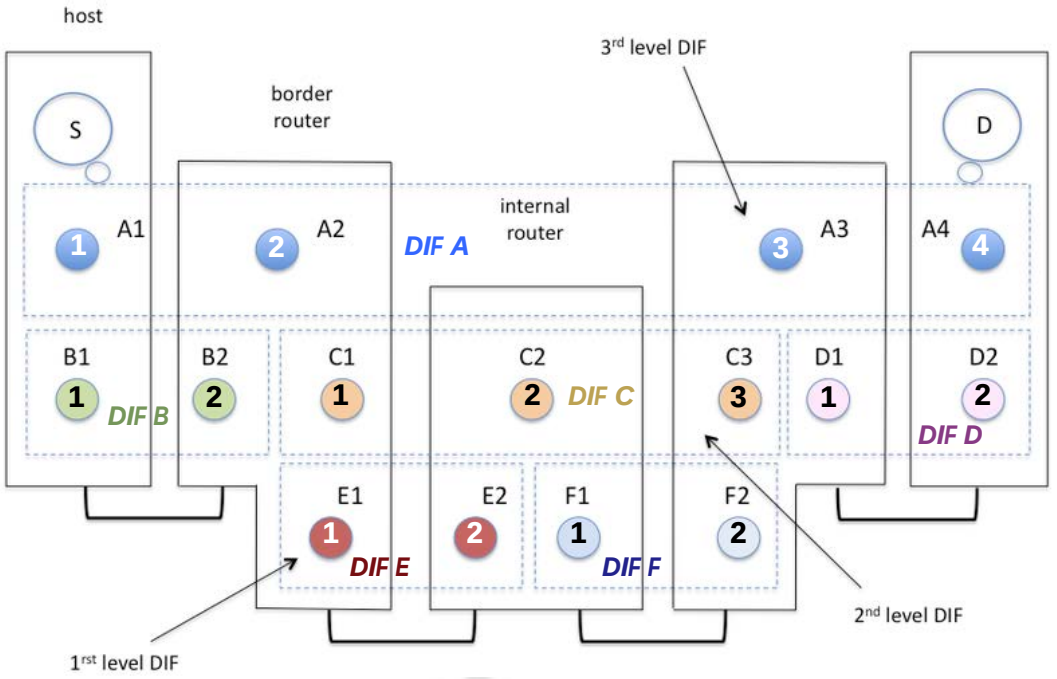
RINA Architecture



- A structure made of recursive layers that provide IPC (**Inter Process Communication**) services to applications on top
- There's a single type of layer that repeats as many times as required by the network designer
- Separation of mechanism from policy

- All layers have the same functions, with different scope and range.
  - Not all instances of layers may need all functions, but don't need more.
- A Layer is a Distributed Application that performs and manages IPC.
  - A Distributed IPC Facility (DIF)
- This yields a theory and an architecture that scales indefinitely,
  - i.e. any bounds imposed are not a property of the architecture itself.

Naming and addressing in RINA



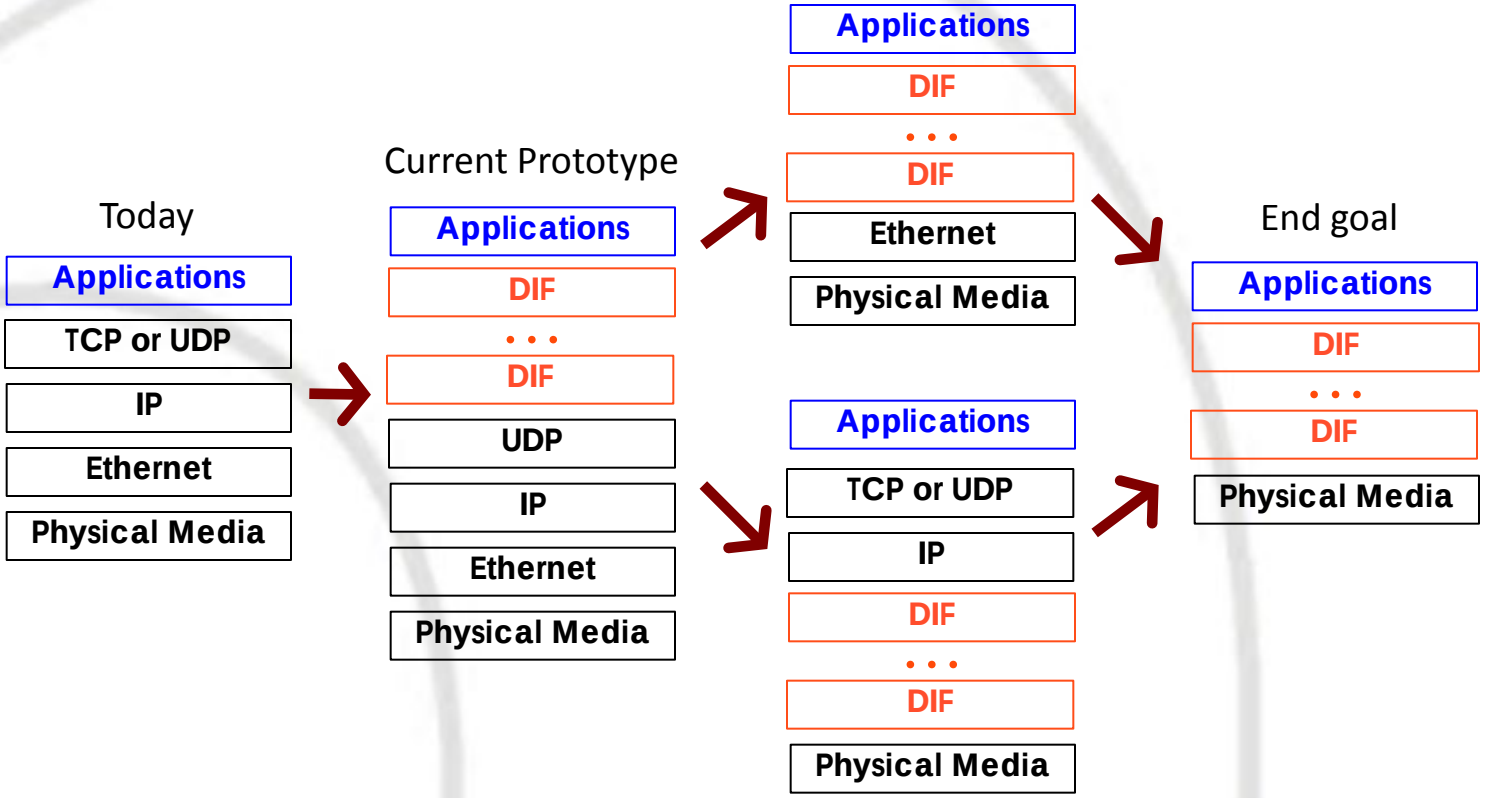
- All application processes (including IPC processes) have a name that uniquely identifies them within the application process namespace.
- In order to facilitate its operation within a DIF, each IPC process within a DIF gets a synonym that may have topological significance within the DIF (i.e. an address).

- The scope of an address is the DIF, addresses are not visible outside of the DIF.
- Each DIF has a directory that maps destination Application process names to DIF IPC Process addresses.
- Because the architecture is recursive, applications, nodes and PoAs are relative
  - For a given DIF of rank N, the process at the layer N+1 is an application and the process at the layer N-1 is a Point of Attachment.

Outline

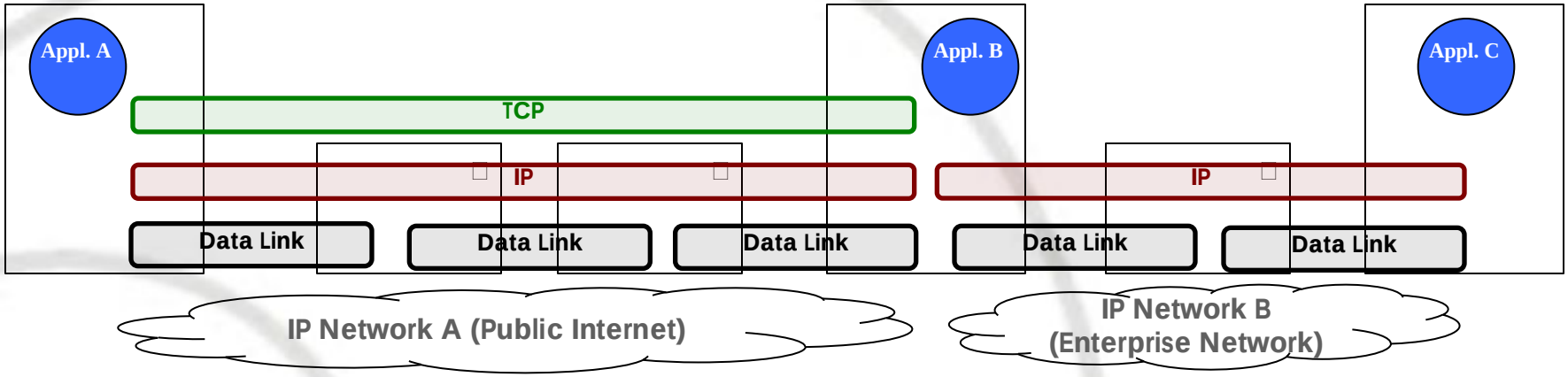
- Quick Introduction to RINA
- **Java prototype over IP**
- Inter DIF Directory
- RINA Security Assessment
- FP7 IRATI Project: Kernel prototype over Ethernet

RINA Adoption strategy

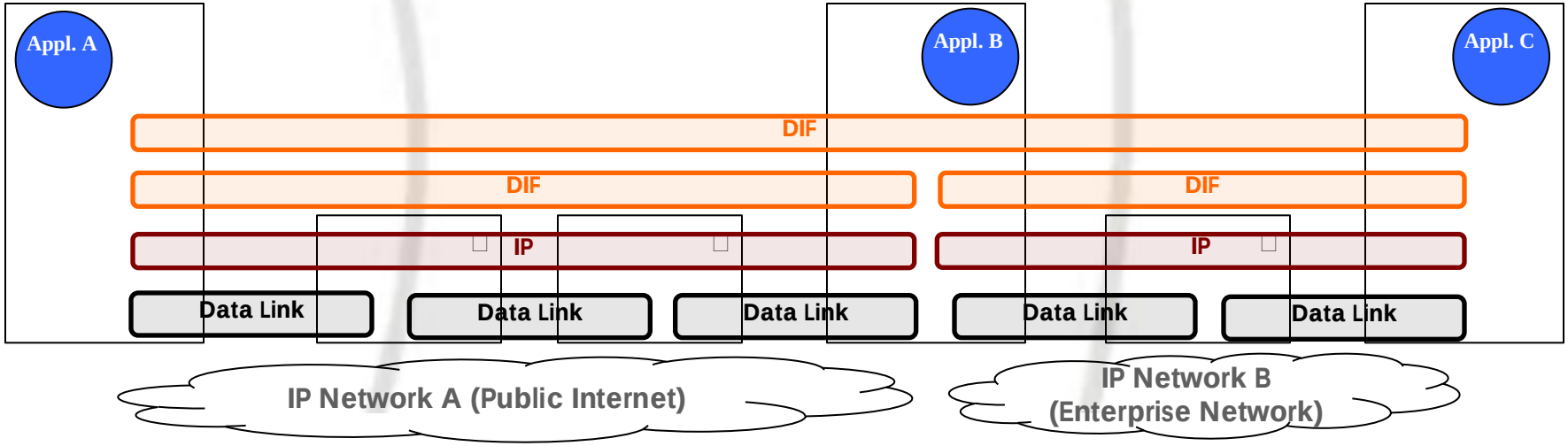


- Start as an overlay to IP, validate technology, work on initial concepts, develop DIF machinery.
  - Useful by itself: internetwork layer(s), decouple application from infrastructure, improved application API, support for multi-homing and mobility.

RINA over IP benefits: Internetwork layer(s)

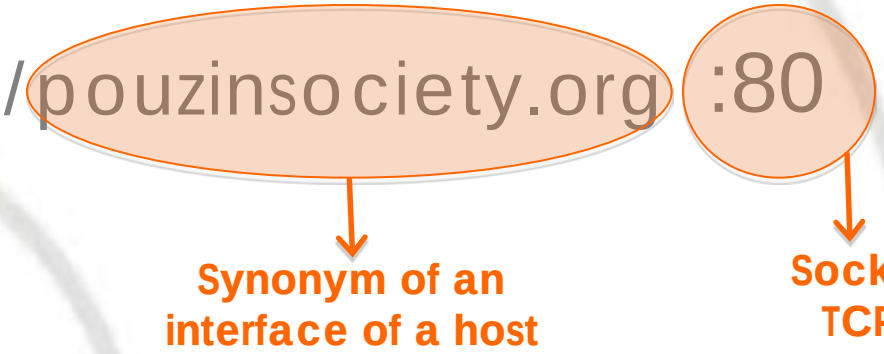


- What if application A wants to communicate with Application C?
  - It cannot do it, unless you start deploying middleboxes like NATs, application-layer gateways, … The architecture doesn't accommodate internetworking!



RINA over IP benefits: Separate applications from infrastructure

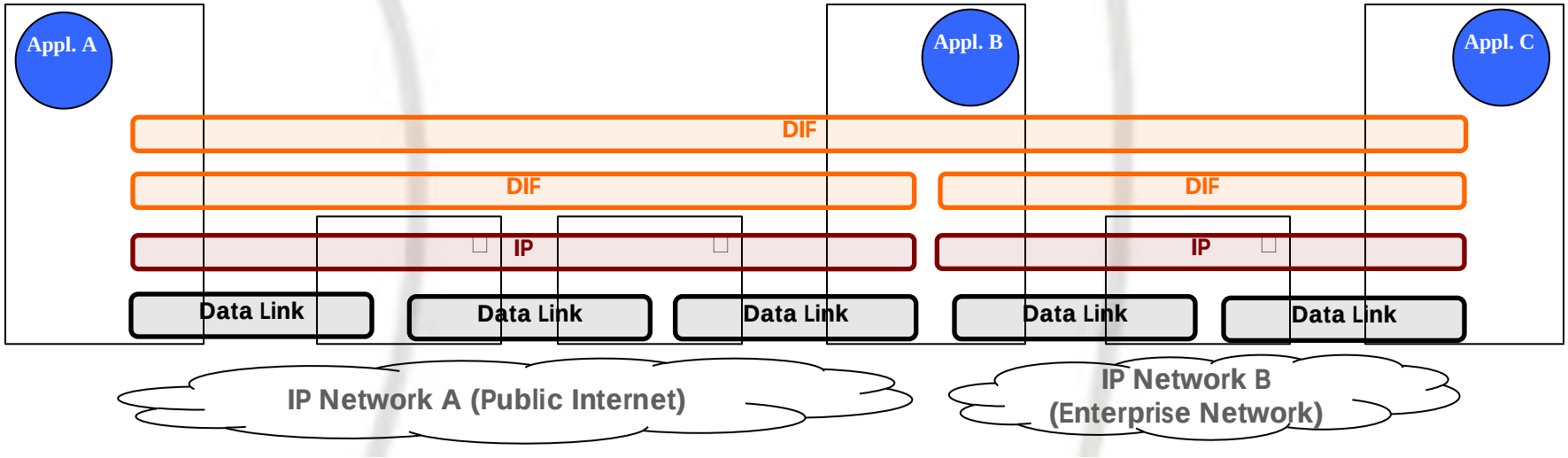
- The current application namespace is tied to IP addressing and TCP/UDP port numbers:



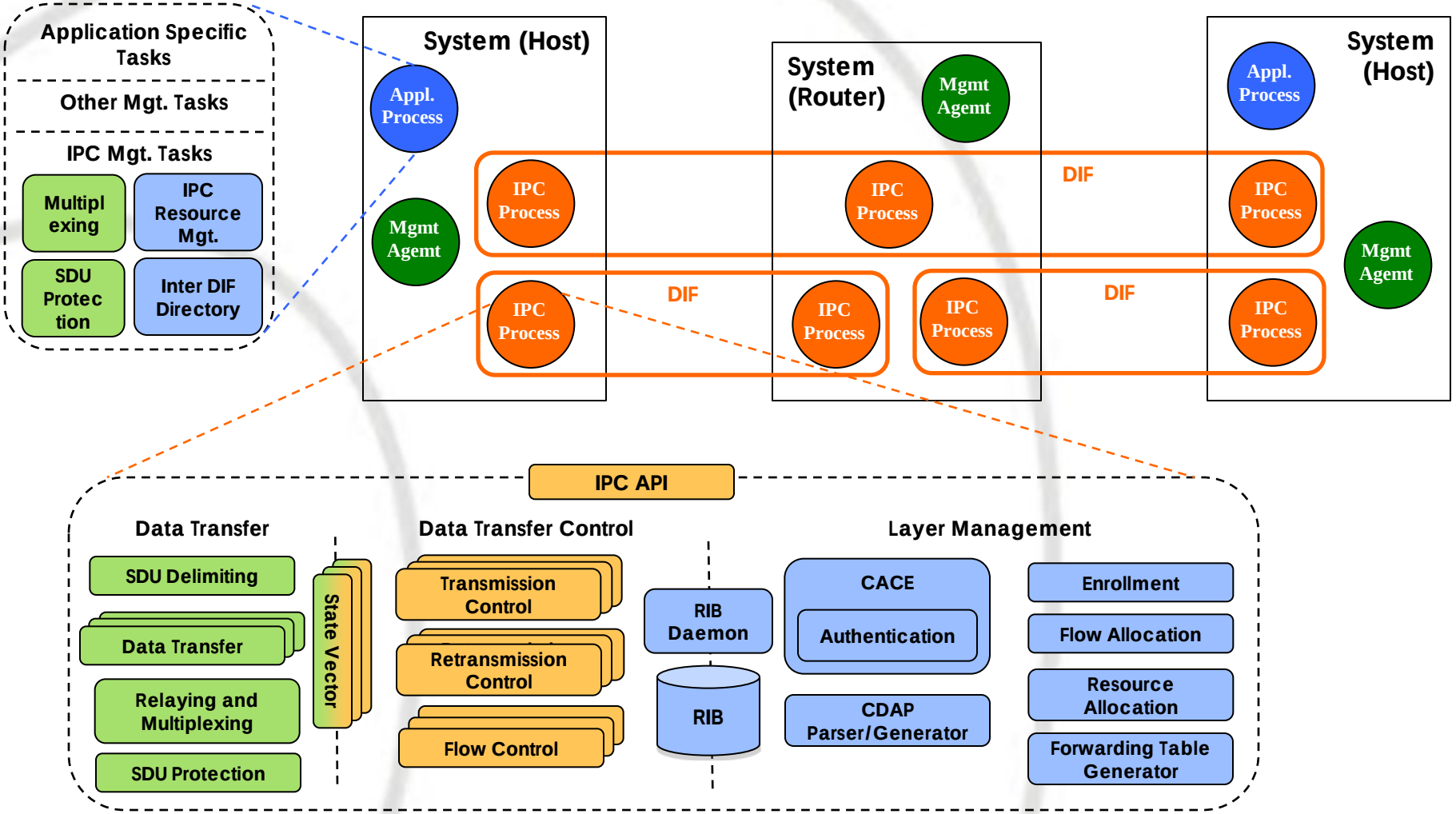
- This makes mobility hard to achieve
- In RINA applications have names that are independent of the layers below (DIFs)
  - Application names can be structured in a way that makes sense for the application
  - The application name doesn't contain the semantics of where the application is in the network (i.e. what is its point of attachment to the layer below)

RINA over IP benefits: Next generation VPN

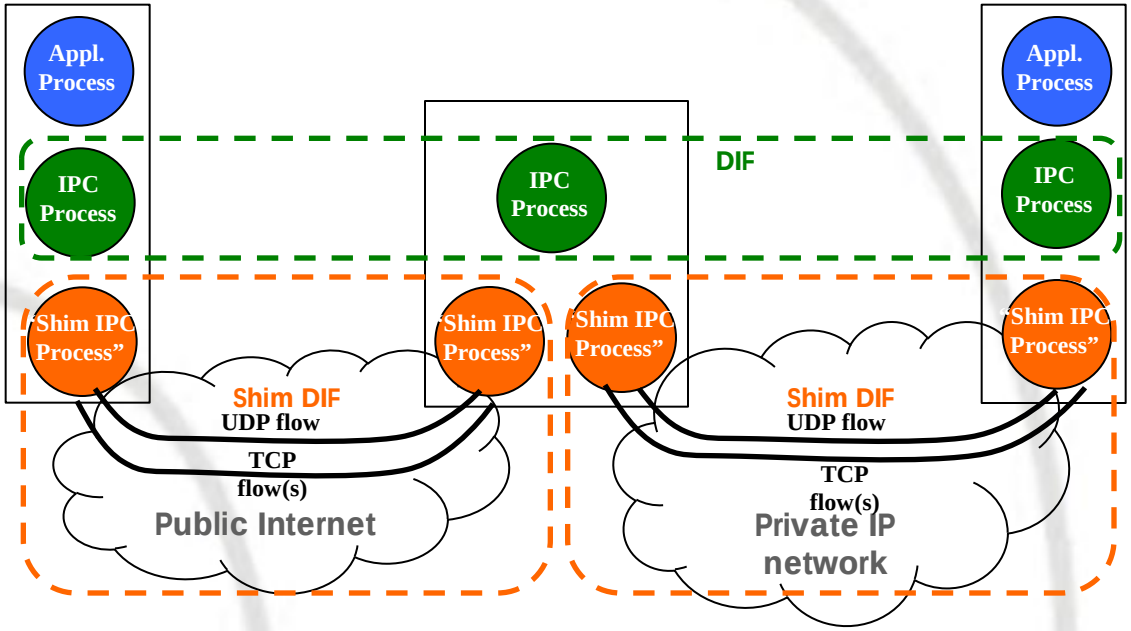
- DIFs are customizable VPNs that can span multiple IP networks.
  - Each DIF has its own addressing scheme, security mechanisms (authentication, authorization), routing strategy, resource allocation strategy (support for different levels of QoS), flow control strategy, data transfer/data transfer control, ...
  - Processes (and not systems) are members of the DIFs (different processes can access different DIFs in each system). Processes may not have access to the whole range of DIFs available on their system
  - DIFs open the door to VPNs optimized for certain applications



Architectural model



The Shim DIF

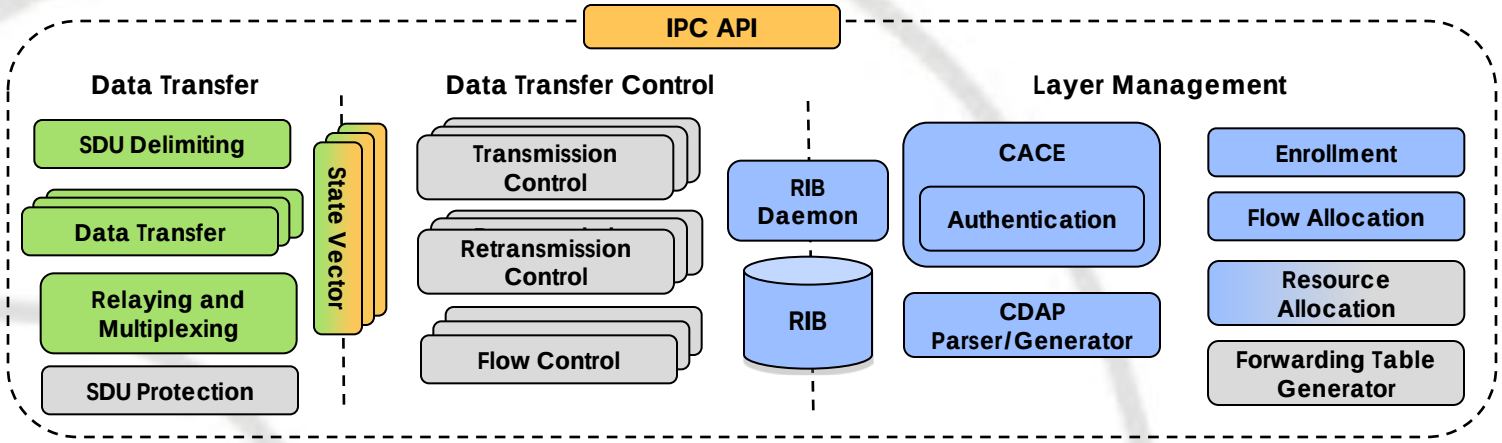


- The "shim IPC Process" for IP networks is not a "real IPC Process". It just presents an IP network as if it was a regular DIF.
  - Wraps the IP network with the DIF interface.
  - Maps the names of the IPC Processes of the layer above to IP addresses in the IP network.
  - Creates TCP and/or UDP flows based on the QoS requested by an "allocate request".

Implementation platform

- Implemented as part of the TINOS framework (a network protocol experimentation framework)
  - https://github.com/PouzinSociety/tinos
- Implemented in Java, using the OSGi technology (OSGi container provided by the Eclipse Virgo container)
  - OSGi is a component model that facilitates building modular Java applications
- Tested on Mac OS X and Linux Debian, but should be multi-platform (support all the platforms that Eclipse Virgo supports)
- Not yet fully integrated with TINOS (once it is, it will be possible to instantiate several "systems" within a single Java process, using XMPP as the underlying "physical substrate")

Current status & some conclusions



- The transition from an IPC model to a (RIB) programming model simplifies layer management (usually referred to as the control plane)
  - If a new function is required, don't need to add a new protocol, just add some objects and the logic to process actions upon them.
- Customization & innovation in networking becomes much easier
  - Just focus on the area you're interested in and change the right policies for yours (Currently "changing the policies" in the prototype means modifying the code, since developing a pluggable policy framework is out of the scope of the initial prototype)
- Interoperability with existing networking technologies is not an issue
  - Just wrap the XX layer as a shim DIF, and there you go

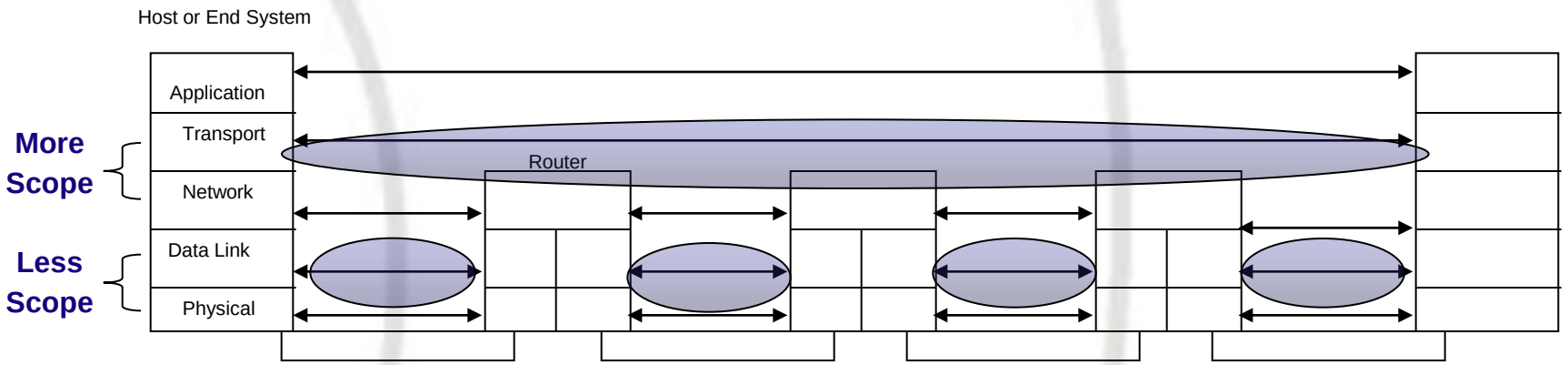
Outline

- Quick Introduction to RINA
- Java prototype over IP
- **Inter DIF Directory**
- RINA Security Assessment
- FP7 IRATI Project: Kernel prototype over Ethernet

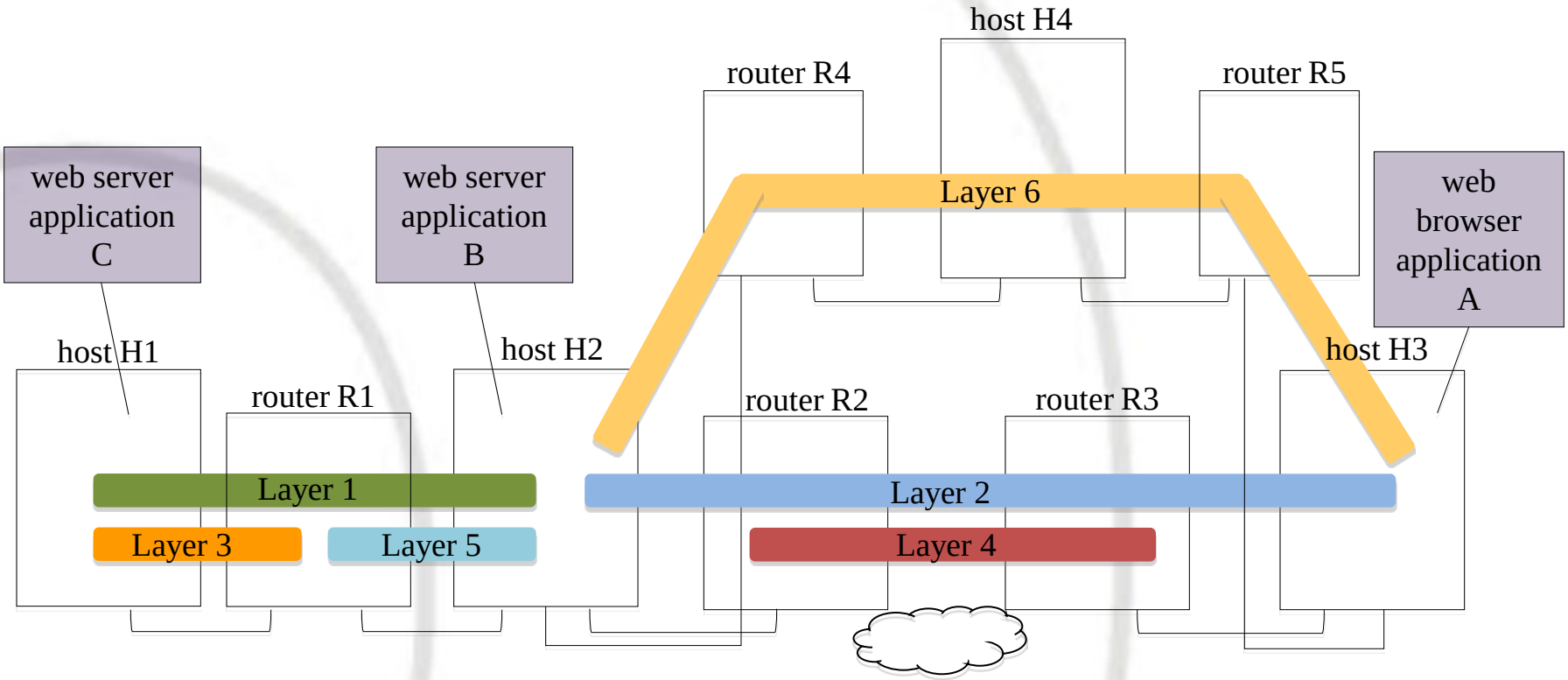
What's a layer?

A layer is a set of application processes that maintain shared state over a scope

- Using layers in networks is a necessity because of the distributed shared state of different scopes



Application discovery involves also layer discovery



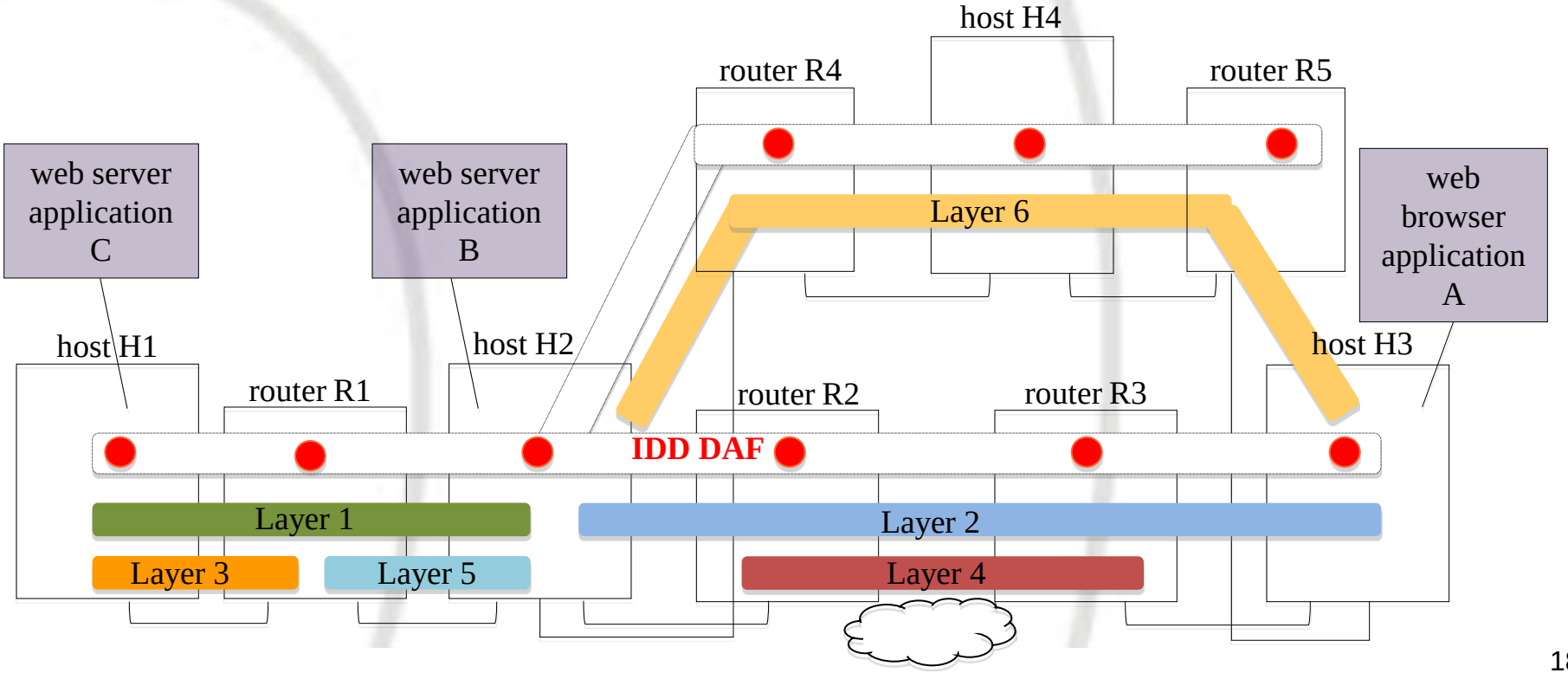
The InterDIF Directory (IDD)

- A distributed application, ***Distributed Application Facility*** (DAF) as it's called in RINA, which is a collection of two or more cooperating application processes in one or more processing systems, which exchange information using IPC and maintain shared state

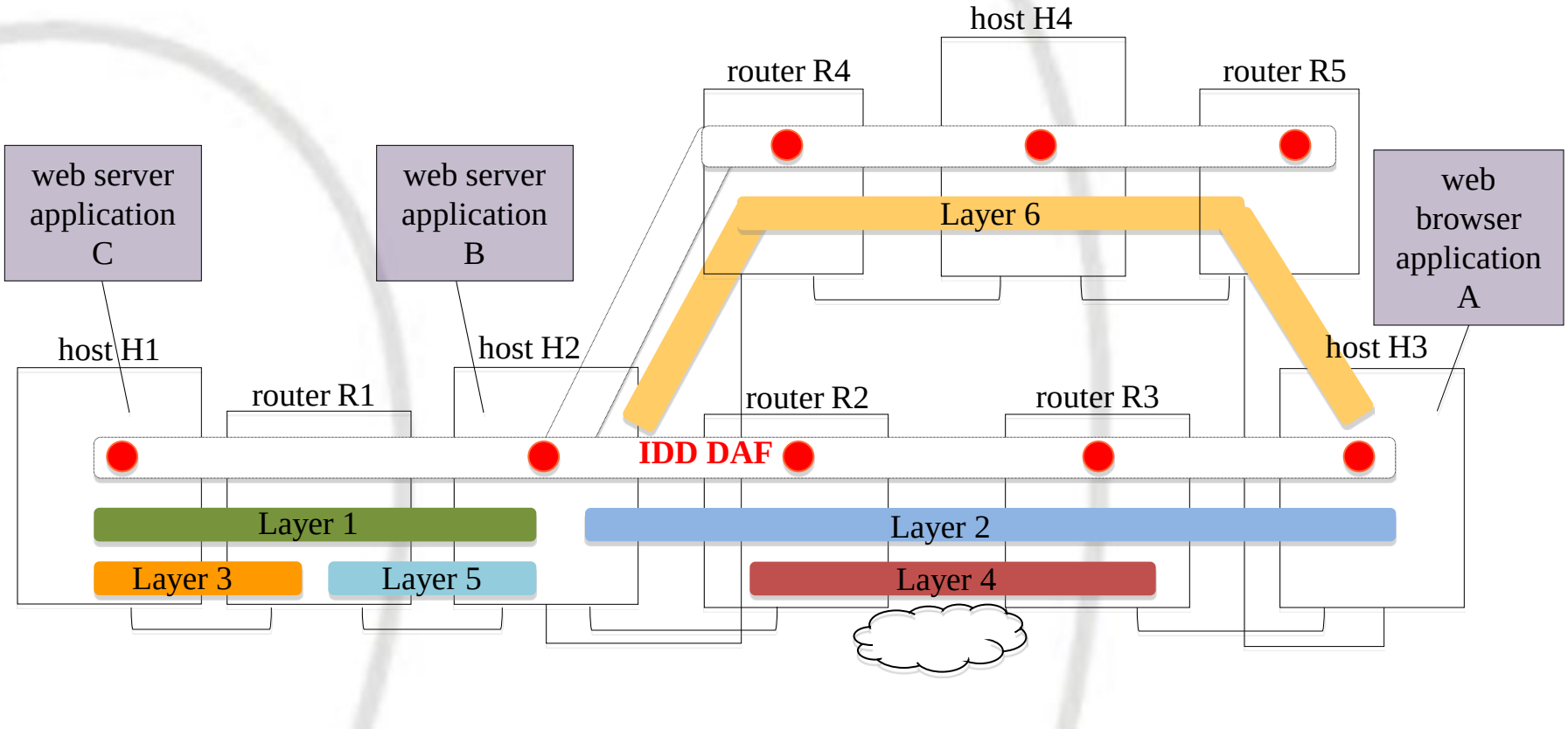
- The IDD DAF maintains a ***distributed database*** that keeps ***mappings of application names to list of supporting layers***

- The IDD is responsible for two main distinct functions:
  a) Discovery of the application
  b) Creation of the supporting DIF

- IDD-Request
  - Destination IDD Name, Source IDD Name, Requested Application Process Name, Access Control Information, Quality of Service, Termination Condition
- Forwarding of the request between the peer IDDs until the destination application is found or the pre-defined termination condition is met

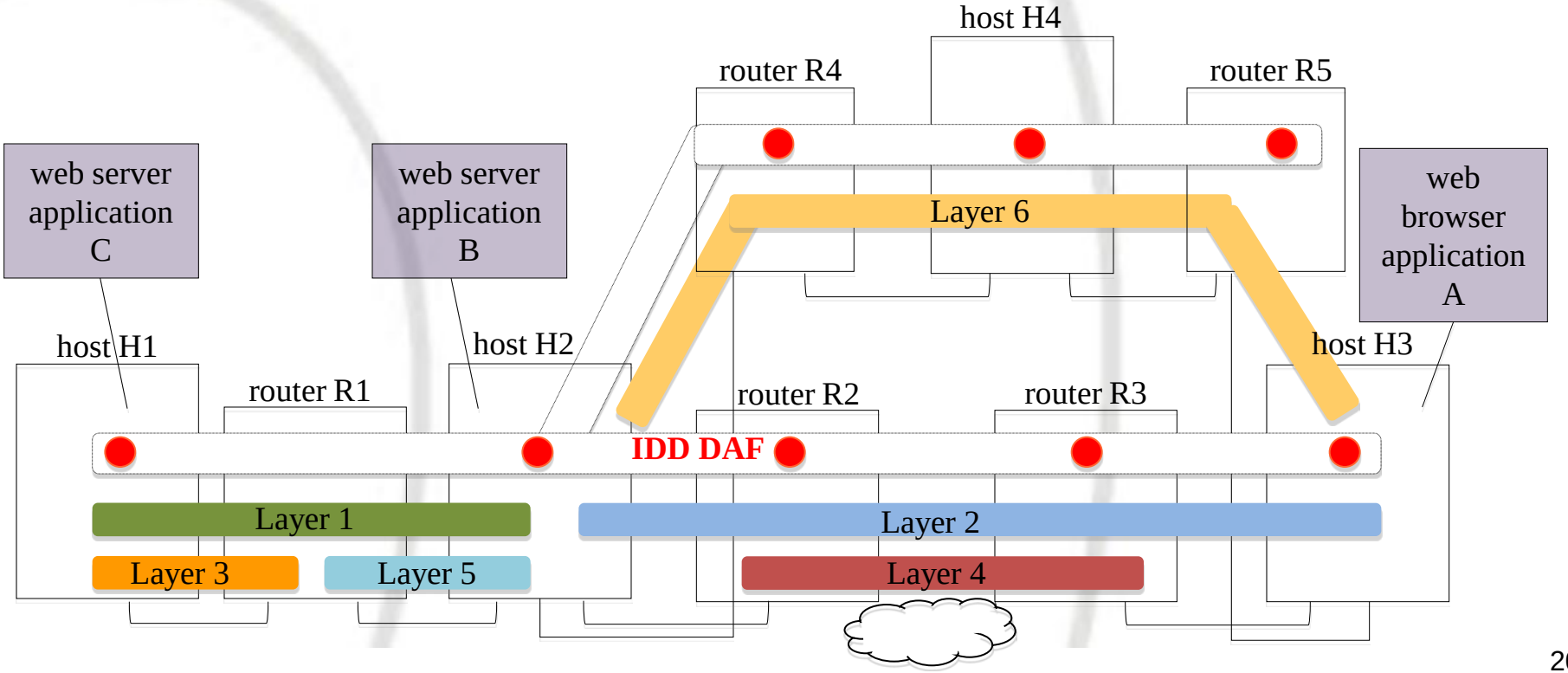


- Confirmation that the requested application is executing in the destination system and authorization check that the requesting application has the rights to access it



B) Creation of the supporting DIF

- A DIF supporting the communication between the two user applications has to be found
- This either involves creating a new DIF from scratch or expanding (joining) an existing one so that it spans from the source to the destination system



Current status

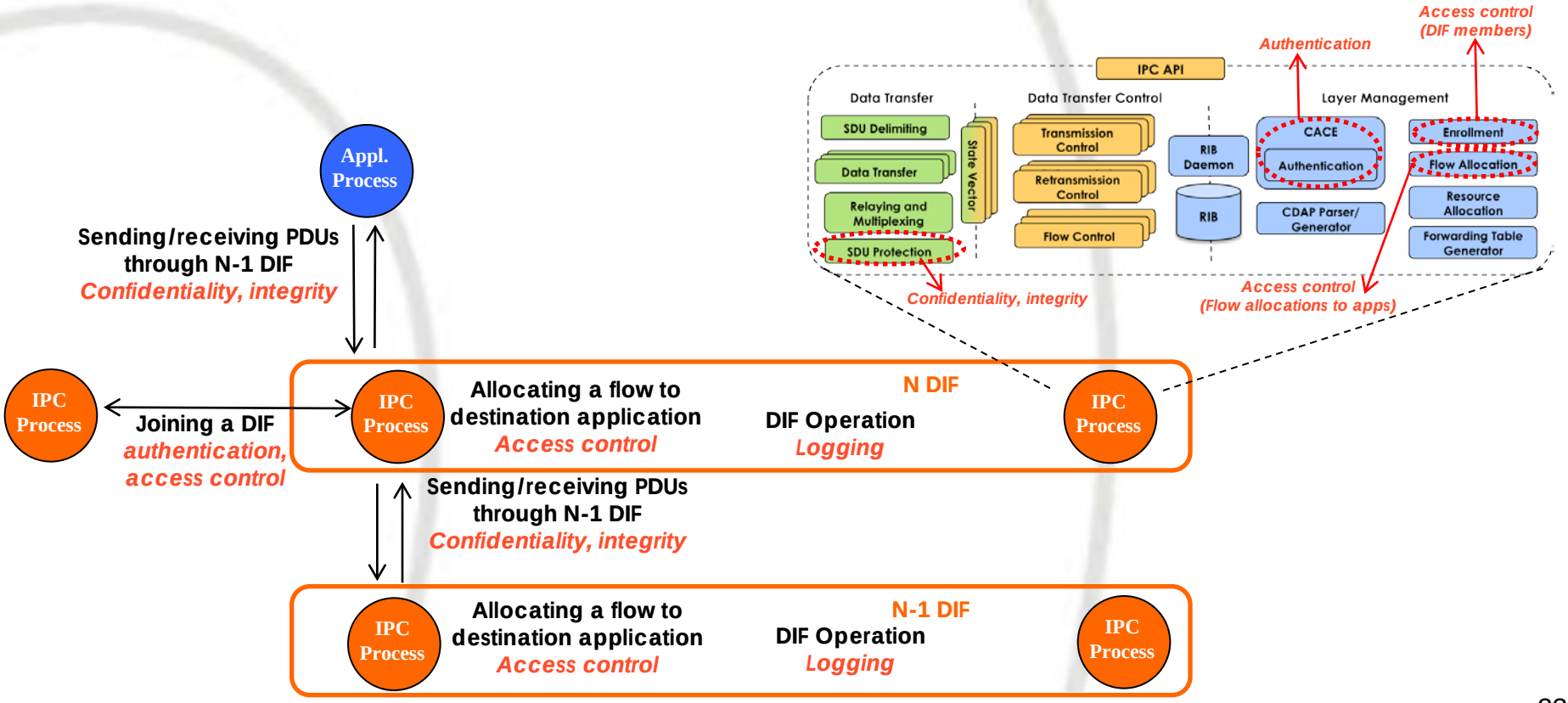
- Ph.D. thesis ongoing
- Defined the IDD Framework. Current research is focused in application discovery; looking at different policies to perform the IDD search, the replication of the directory information, and compare them in terms of:
  - Time to discover the requested application
  - Average number of messages generated on the IDD DAF per search
  - Average number of hops needed to locate the destination application
  - Time to distribute directory updates, and average number of messages generated by directory update
- Java simulator of the IDD is being implemented. After results have been tested in the simulator, the IDD framework and some selected policies will be ported to the prototypes

Outline

- Quick Introduction to RINA
- Java prototype over IP
- Inter DIF Directory
- **RINA Security Assessment**
- FP7 IRATI Project: Kernel prototype over Ethernet

Placement of security related functions

- Benefits of having an architecture instead of a protocol suite: ***the architecture tells you where security related functions are placed***.
  - Instead of thinking security at a protocol level, think security at a system level: no more 'each protocol has its own security', 'add another protocol for security' or 'add another box that does security'



DIFs are securable containers

- Master thesis on RINA security assessment (results to be published)
  - Possible attacks to a DIF
  - Information required to perform these attacks
  - Mitigation measures against these attacks
  - Comparison to the current Internet

- Concludes that ***DIFs are securable containers if proper standard security tools are used*** (authentication, access control, confidentiality, integrity and strong auditing)
  - *Securable = A structure used to hold or transport something that can be made to be not subject to threat*

- Again, with a proper structure in place, achieving better security in networks is much simpler and cost-effective than in the current situation

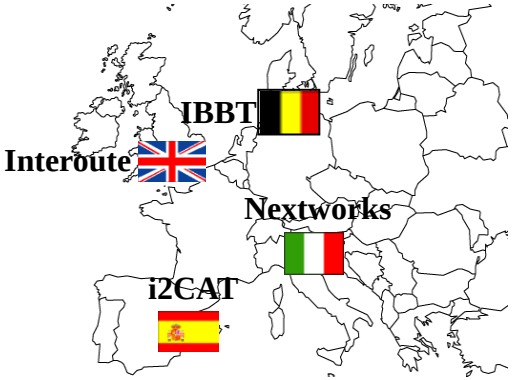
Outline

- Quick Introduction to RINA
- Java prototype over IP
- Inter DIF Directory
- RINA Security Assessment
- **FP7 IRATI Project: Kernel prototype over Ethernet**

Project at a glance

- What? Main goal
  - To **advance** the **state of the art of RINA** towards an **architecture reference model** and **specifications** that are **closer** to enable implementations deployable in **production scenarios**. The **design and implementation** of a **RINA prototype on top of Ethernet** will enable the **experimentation** and **evaluation of RINA in comparison to TCP/IP**.

- Who? 4 partners



- How? Requested 870.000 € funding to the EC to perform 5 activities
  - WP1: Project management
  - WP2: Architecture, Use cases and Requirements
  - WP3: Software Design and Implementation
  - WP4: Deployment into OFELIA testbed, Experimentation and Validation
  - WP5: Dissemination, Standardisation and Exploitation

Thanks for your attention!

You can contact me at eduard.grasa@i2cat.net

More information about RINA at http://rina.tssg.org, http://pouzinsociety.org, http://csr.bu.edu/rina

More information about the prototype at https://github.com/PouzinSociety/tinos/wiki/RINA-Prototype

More information about IRATI at http://irati.eu

i2cat FUNDACIÓ