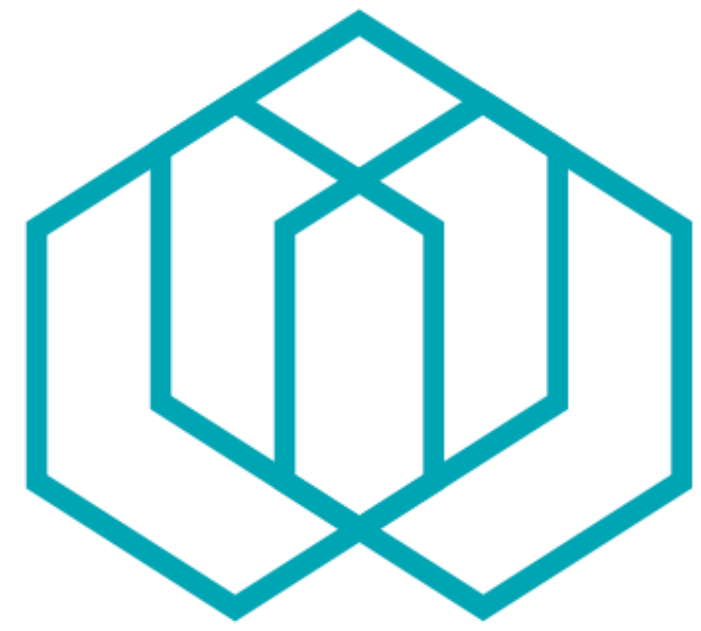


This work is focused on application approach in RINA Architecture, on its design and implementation in RINASim in OMNeT++. RINA yields a completely different approach of communication and manipulation of objects in distributed applications. Instead of passing objects directly between nodes, it manipulates with them locally in RIB. Contribution of this work is to extend the functionality of RINASim, by all the components cooperating in Distributed Application Facility.



RINASim

## Motivation

RINASim is a powerful simulator of RINA Architecture. It comes with a completely different approach in creating distributed applications.

Main goals:

- Create functional Distributed Application Facility with all components
- Design and implement an API for programming applications in RINASim
- Incorporate RIB and RIB Daemon component for managing communication between APs



RINASim GitHub Repository

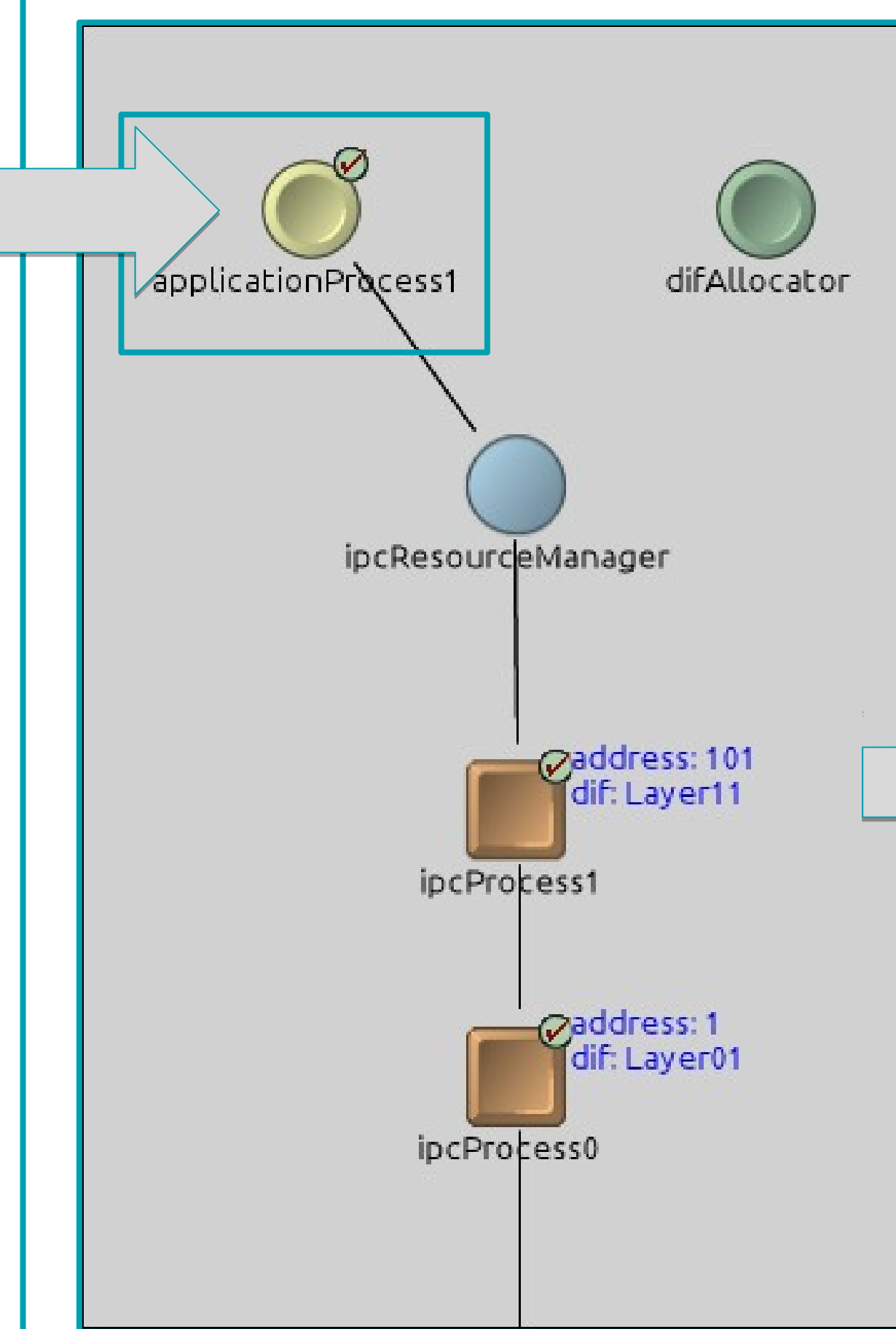
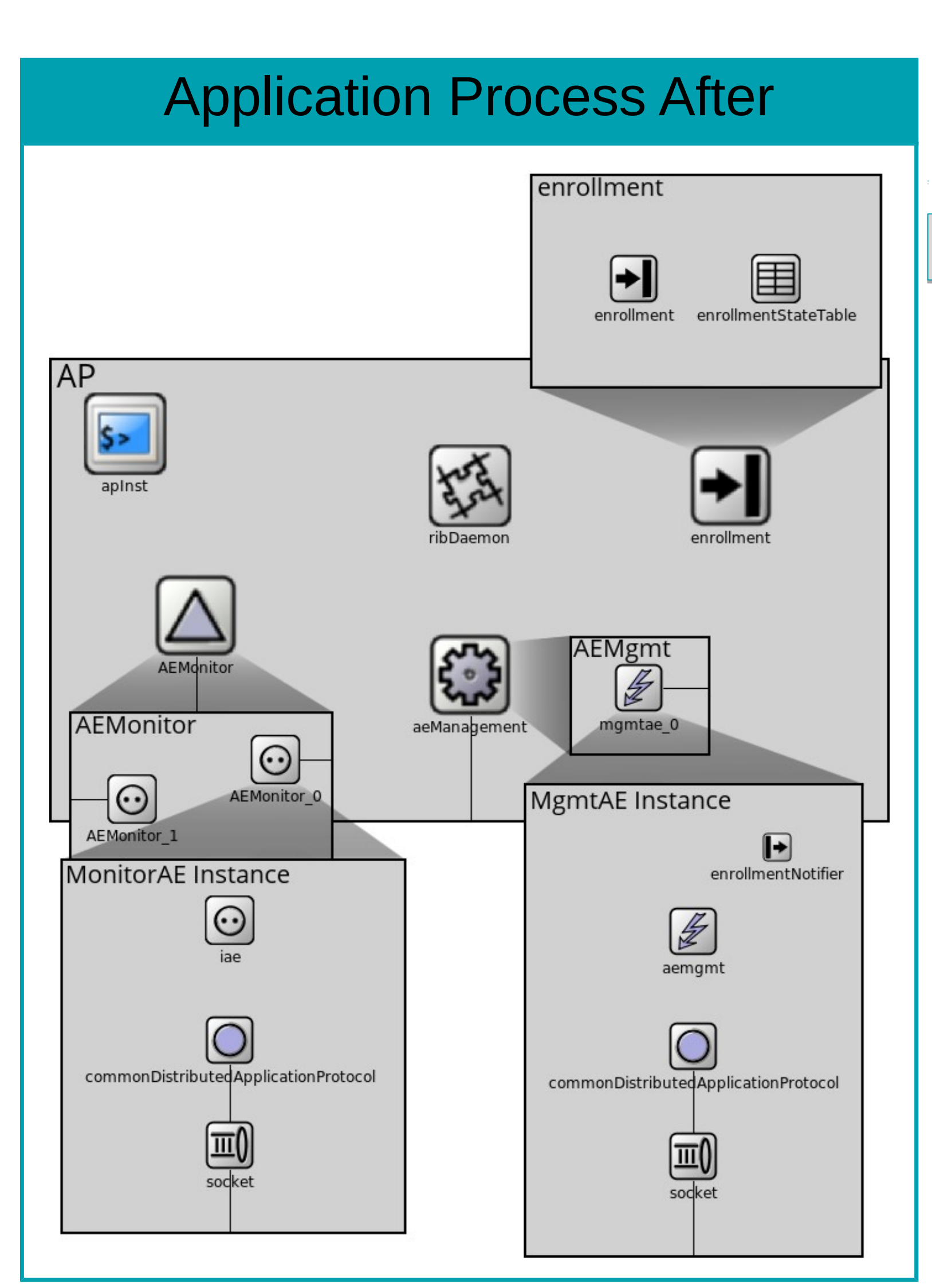
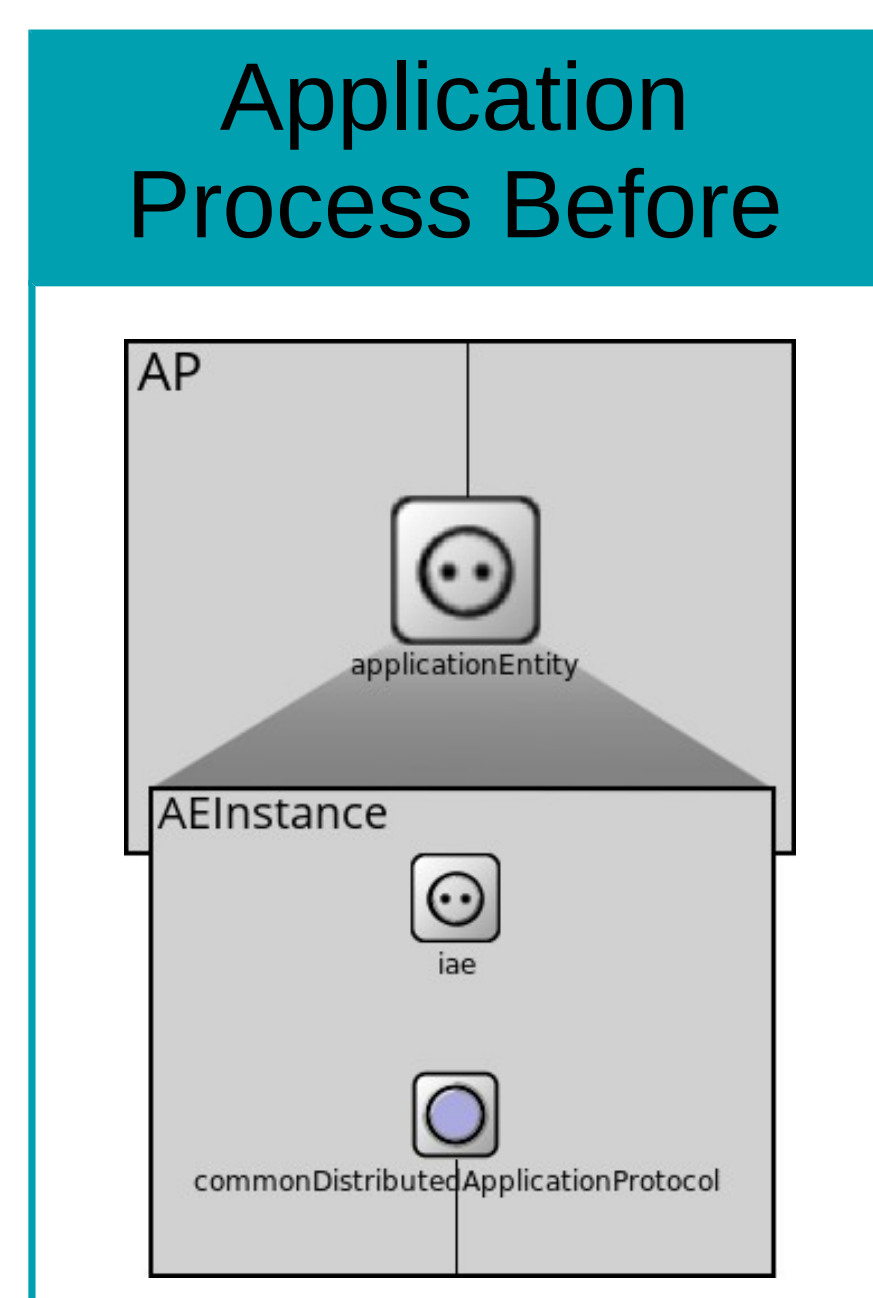
## Design

Rina has an application layer called Distributed Application Facility (DAF). It is a collection of APs intended for some purpose. APs use IPC Processes participating in Distributed IPC Facility (DIF) to provide connectivity.

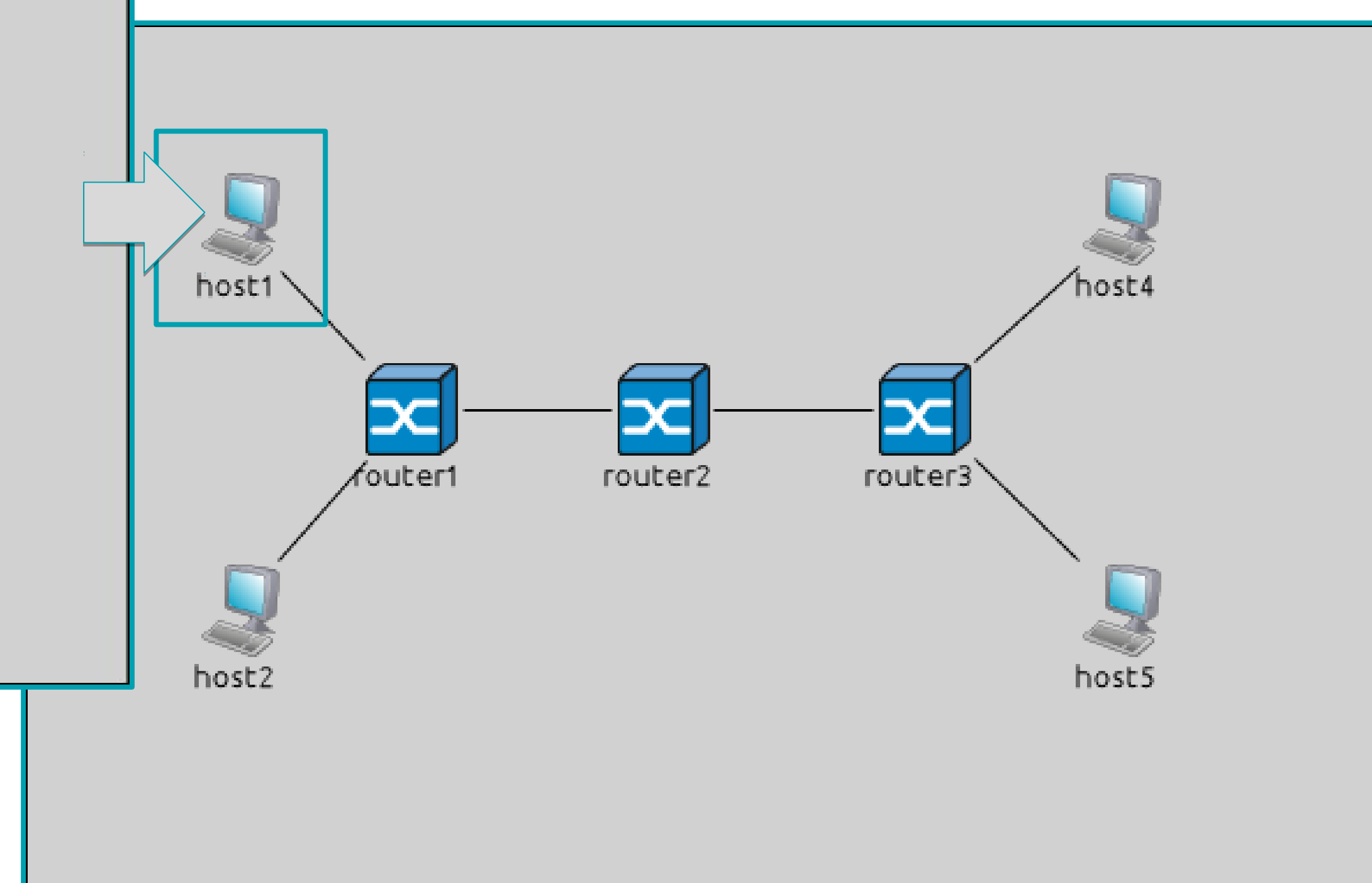
- Each AP contains Application Entities that manage communication, Resource Information Base (RIB) for storing application objects and RIBDaemon that manages and maintains data in RIB between each AP
- There are two initial phases of communication within DAF, Common Application Connection Establishment (CACE) and Enrollment, both managed by Enrollment module

## Application Process

- is no more only simple traffic generator!



- Each module has its own API for communication between them
- Only AE and AP should be programmable by users



## Contribution

- Design and implementation of all APs components
- API for creating more complex applications in RINASim environment
- Implementation of API in each module
- Reaction in each module for API call from other modules in form of callback functions
- Dynamic creation of AE and IAE modules
- CACE and Enrollment phases of communication within DAF
- Storing application objects in RIB
- Managing and distributing objects between APs using RIBDaemon component

## Conclusion

- Current Application Process has been extended with other modules
- There is an API in each module to communicate between them
- Finite state machines for CACE and Enrollment phases of communication were implemented
- The model was extended with a storage and distribution of application objects

## Future Work

- Creation of distributed applications and measurement of efficiency of traffic between nodes