Autonomic Transport Framework

for large family of communicating objects

Ernesto EXPOSITO

ernesto.exposito@laas.fr

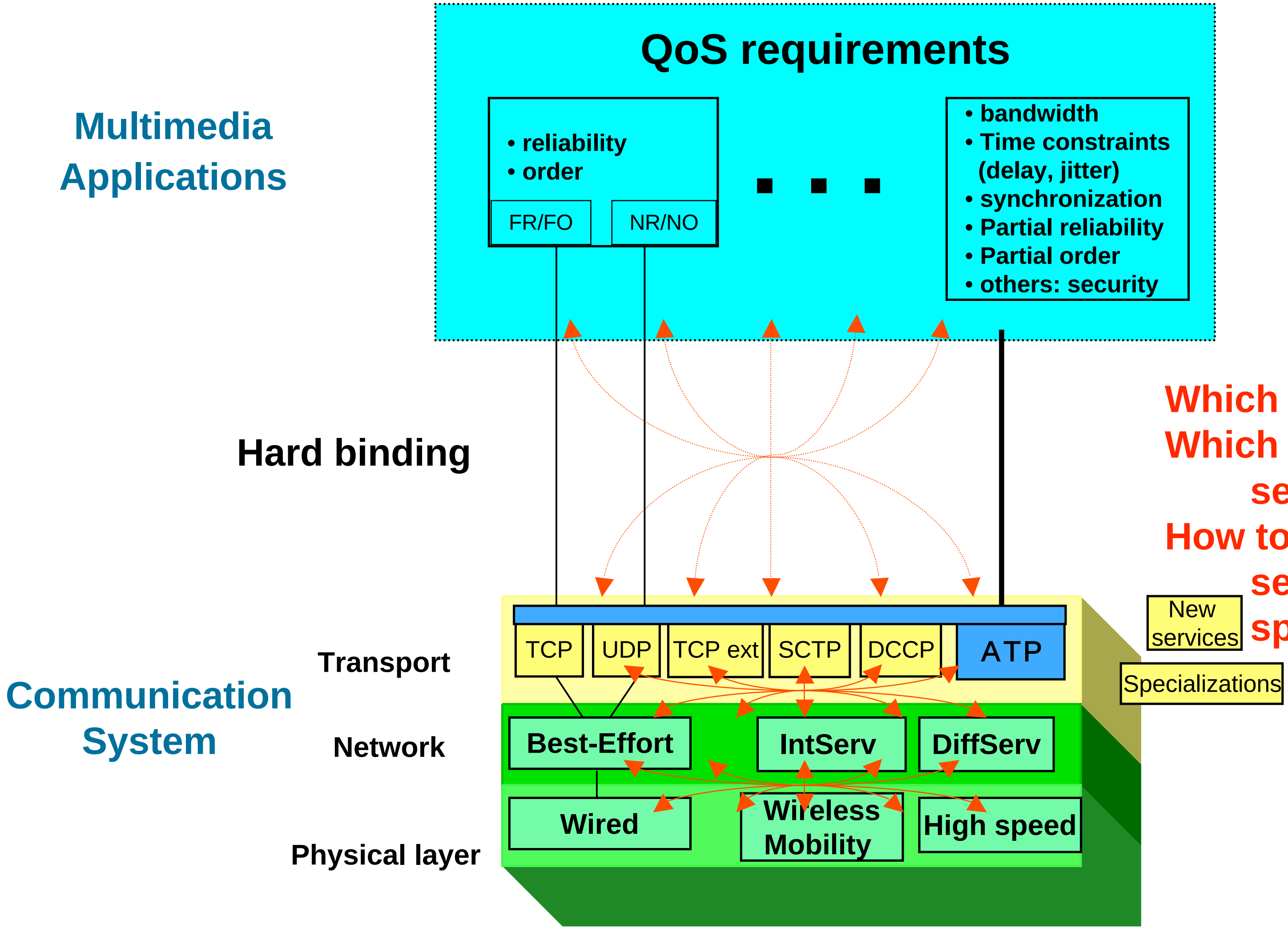
LAAS/CNRS - OLC group, Toulouse

MCF - INSA de Toulouse

Outline

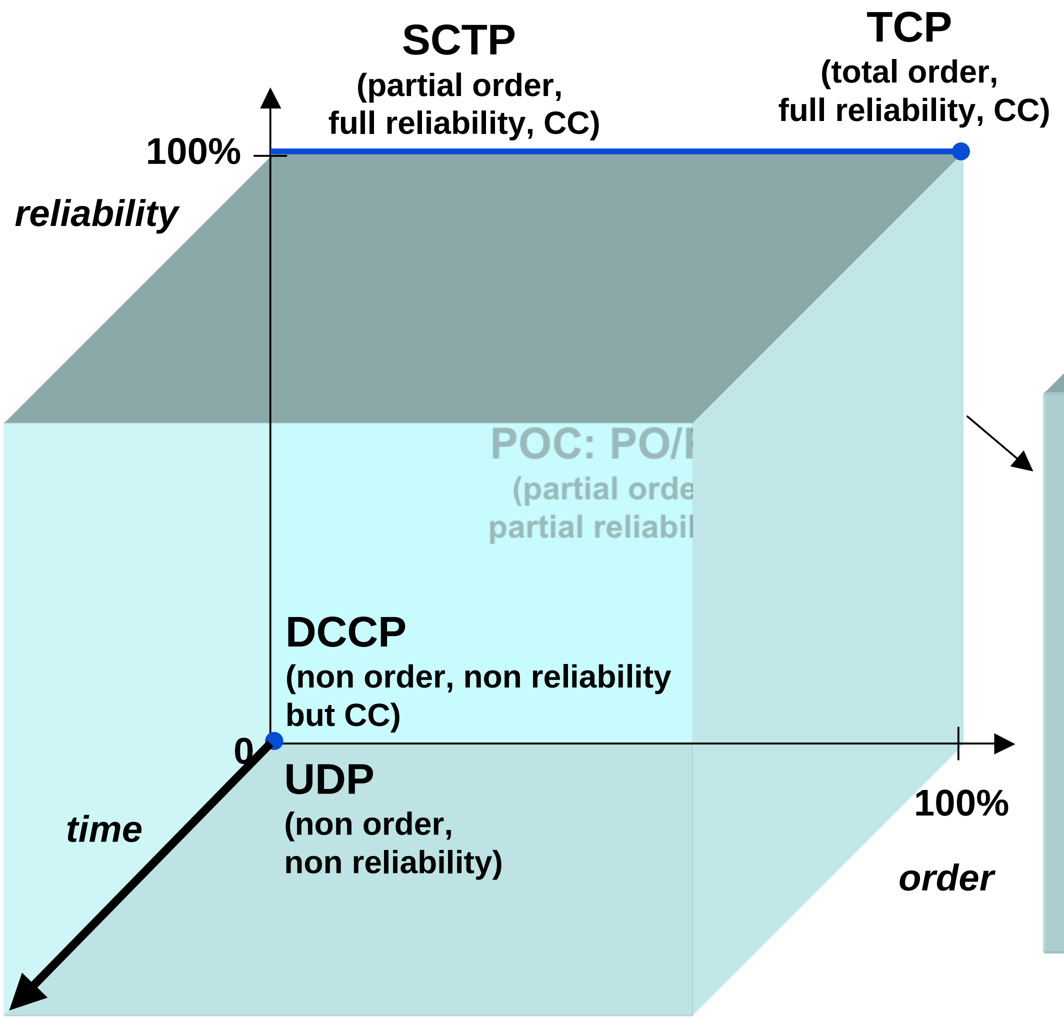
- Motivation
- The Autonomic Transport Protocol Framework
  - Design requirements
  - Communication patterns
  - Autonomic approach
  - Semantic for composition
- Conclusions and perspectives

Motivation: QoS and transport layer evolution



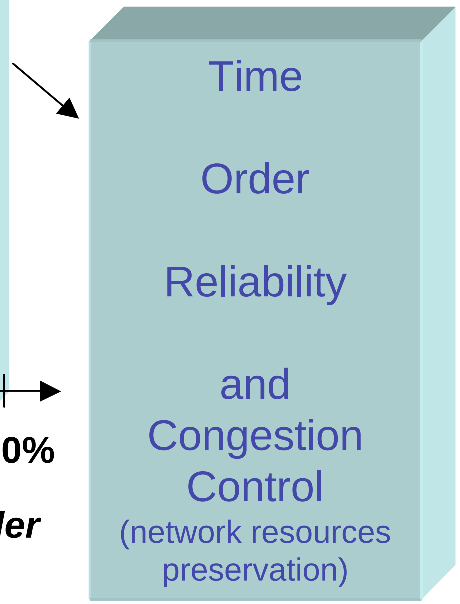
Which transport service?
Which Transport/Network service composition?
How to incorporate new services or service specializations?

Motivation: transport services space



Traditional transport protocols:

- Order
- Reliability
- Congestion Control

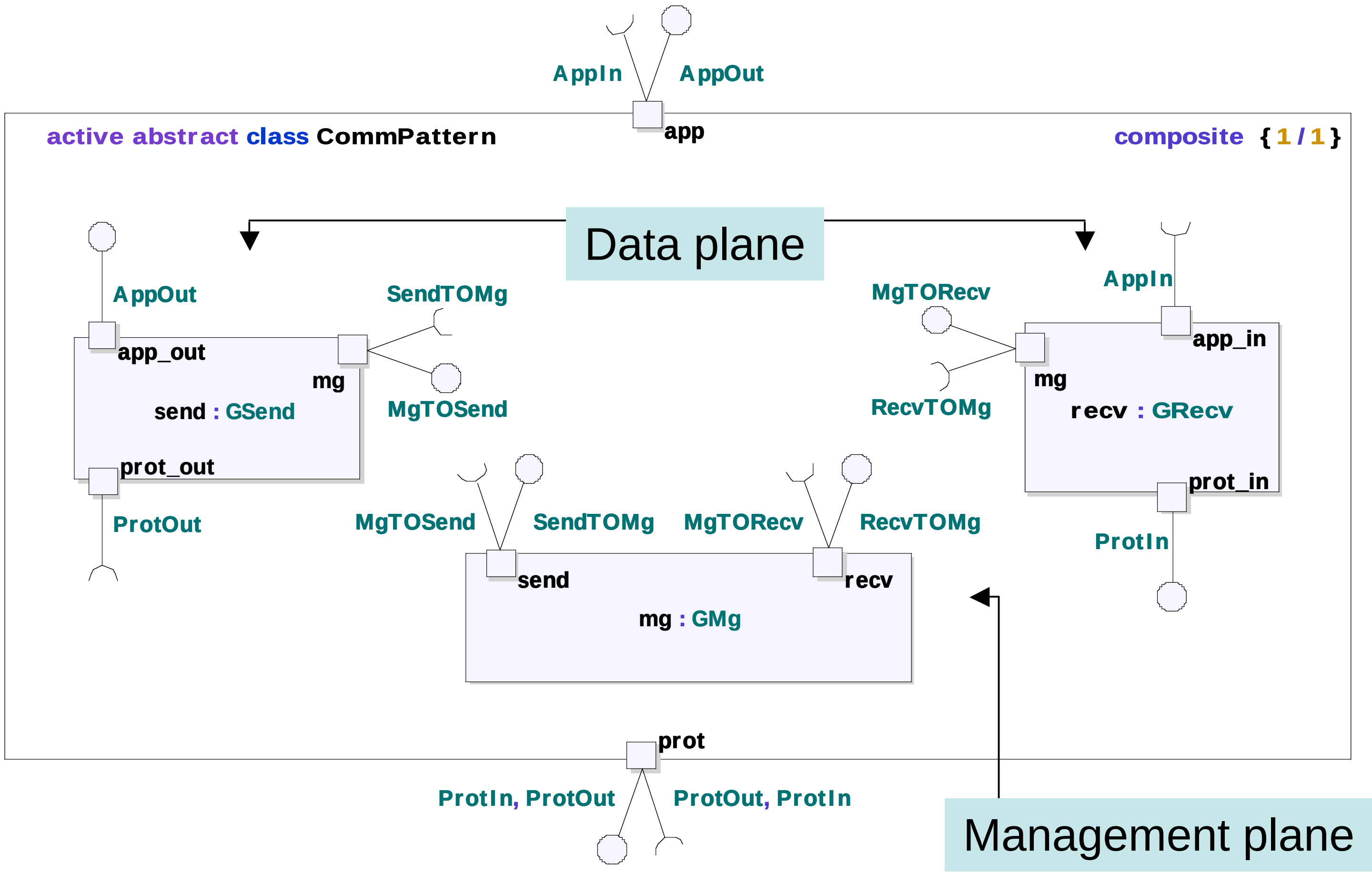


Autonomic Transport Protocol Framework

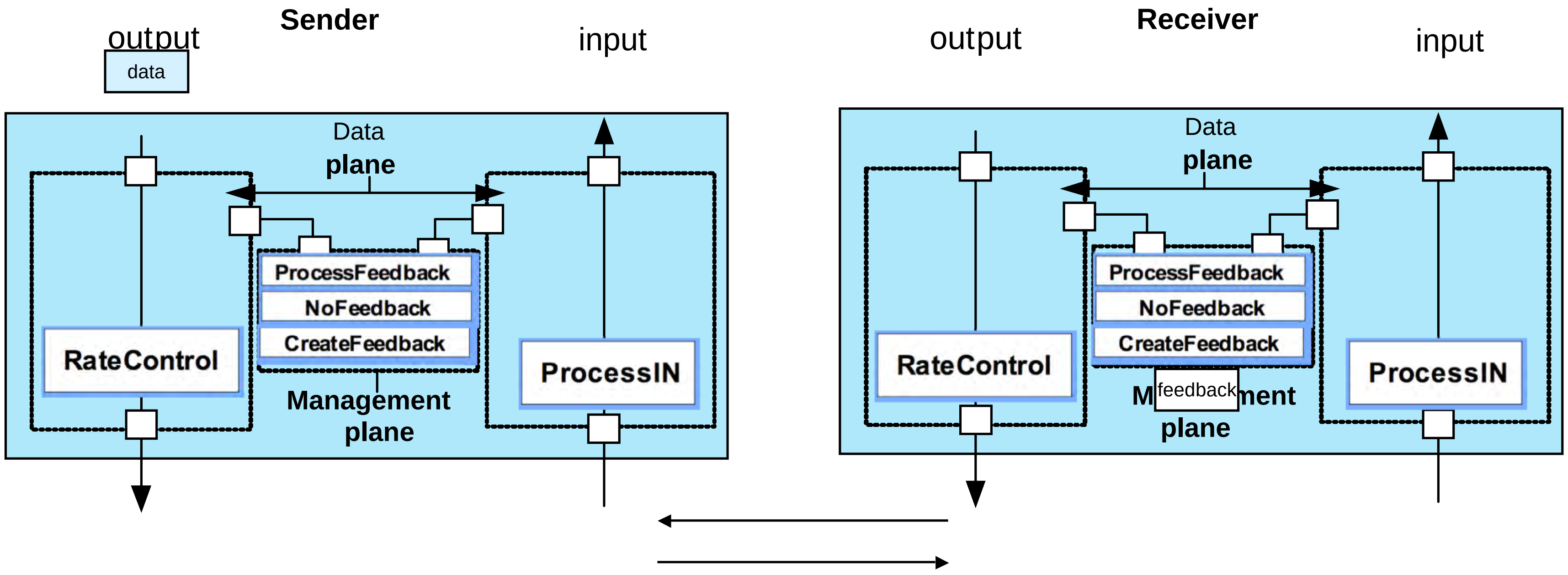
Requirements

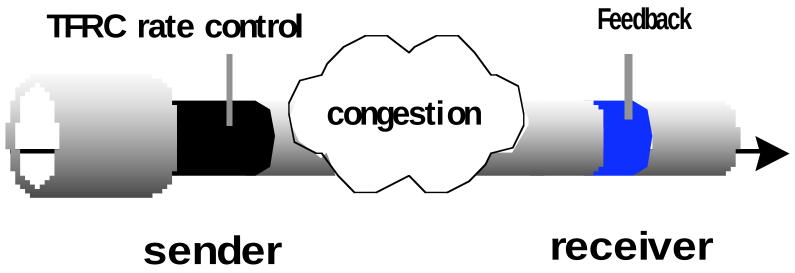
- Composite architecture
  - Existing mechanisms
    - Error control
    - Rate control
    - Congestion control
  - New mechanisms
    - Specialization of existing mechanisms
      - Adaptive Partial reliability
      - Adaptive congestion control
- Autonomic behavior:
  - Autonomic managers
  - Self-configuring
  - Self-optimizing

Composite communication pattern



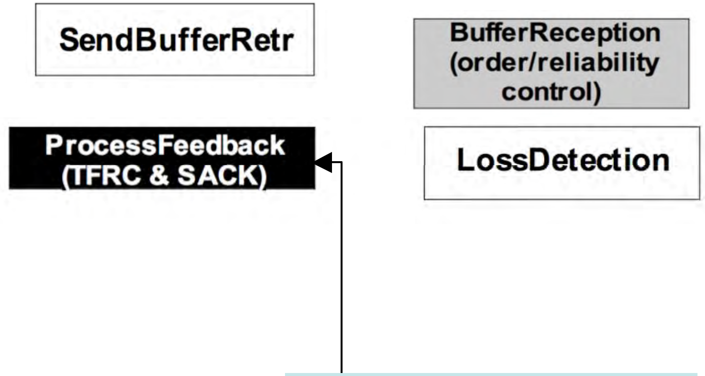
Example1: Composite for TCP-Friendly Rate Control (TFRC)



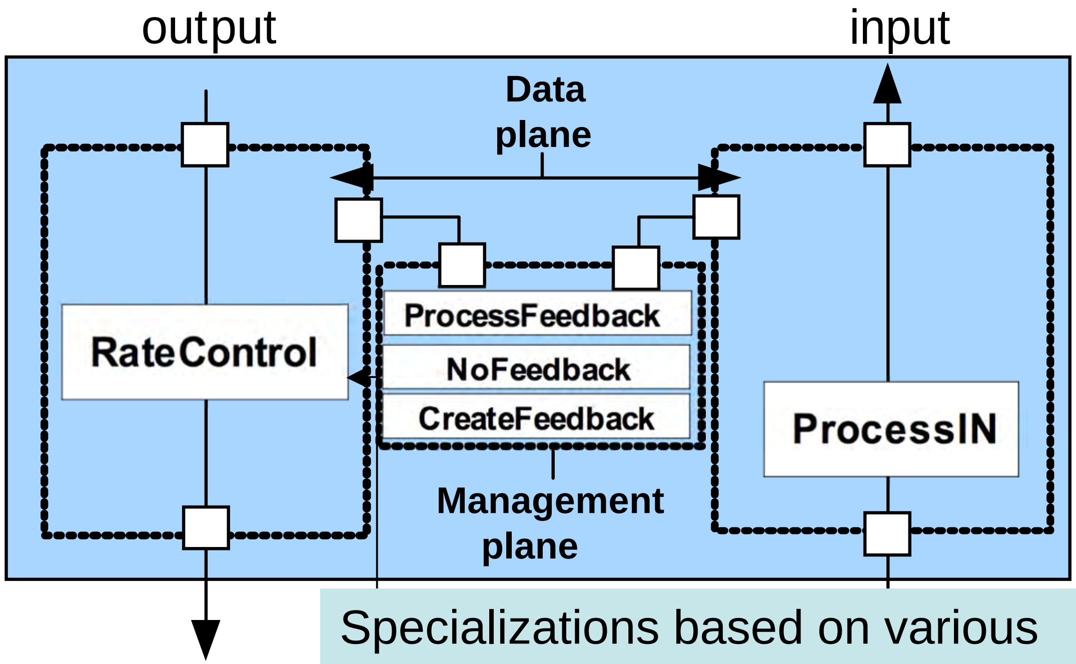


Example2: Composition/specialization of basic mechanisms reliability + rate/congestion control

Congestion control and partial reliability
(TFRC & PR)



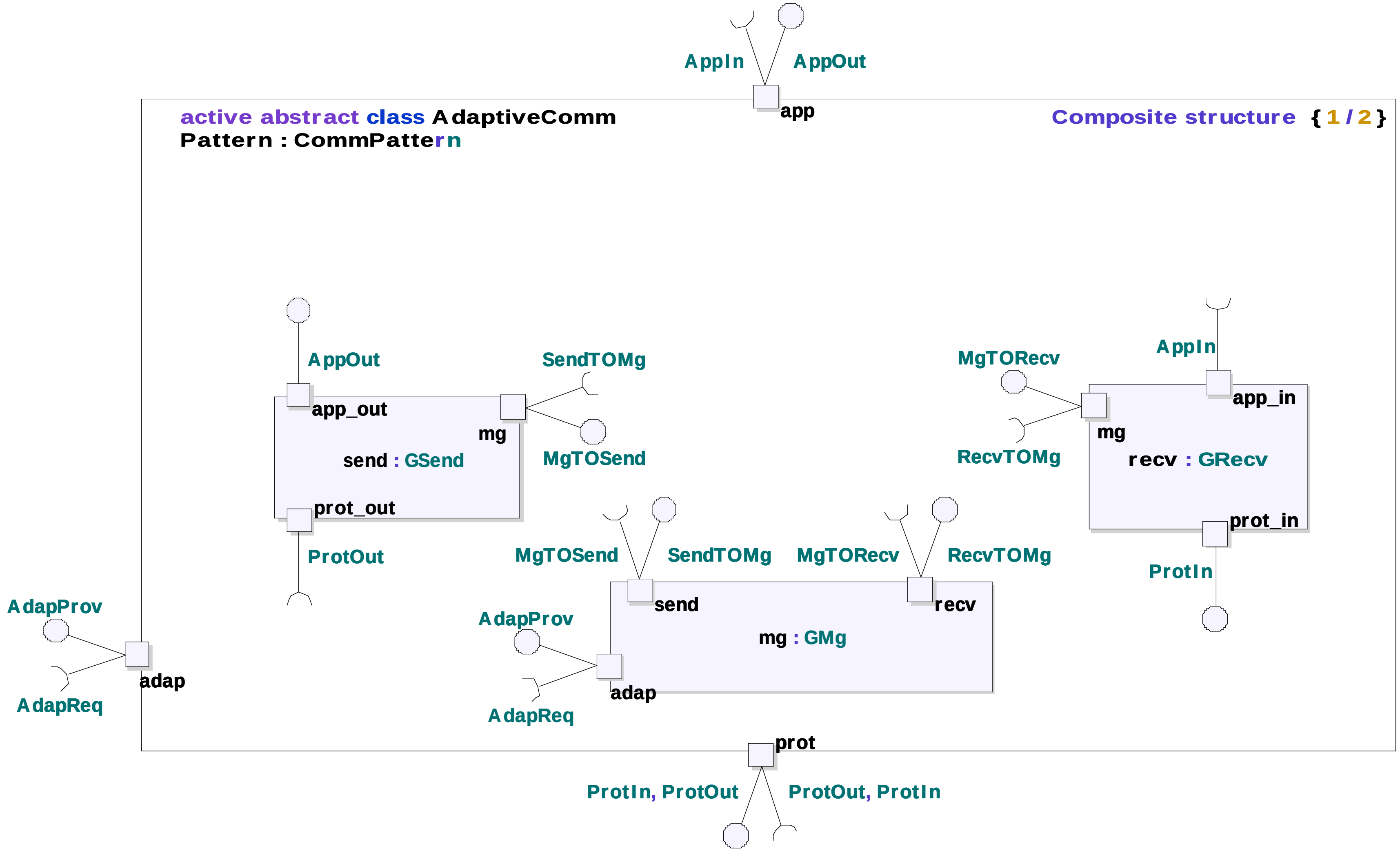
Standard congestion control:
TFRC
(TCP-friendly Rate Control)



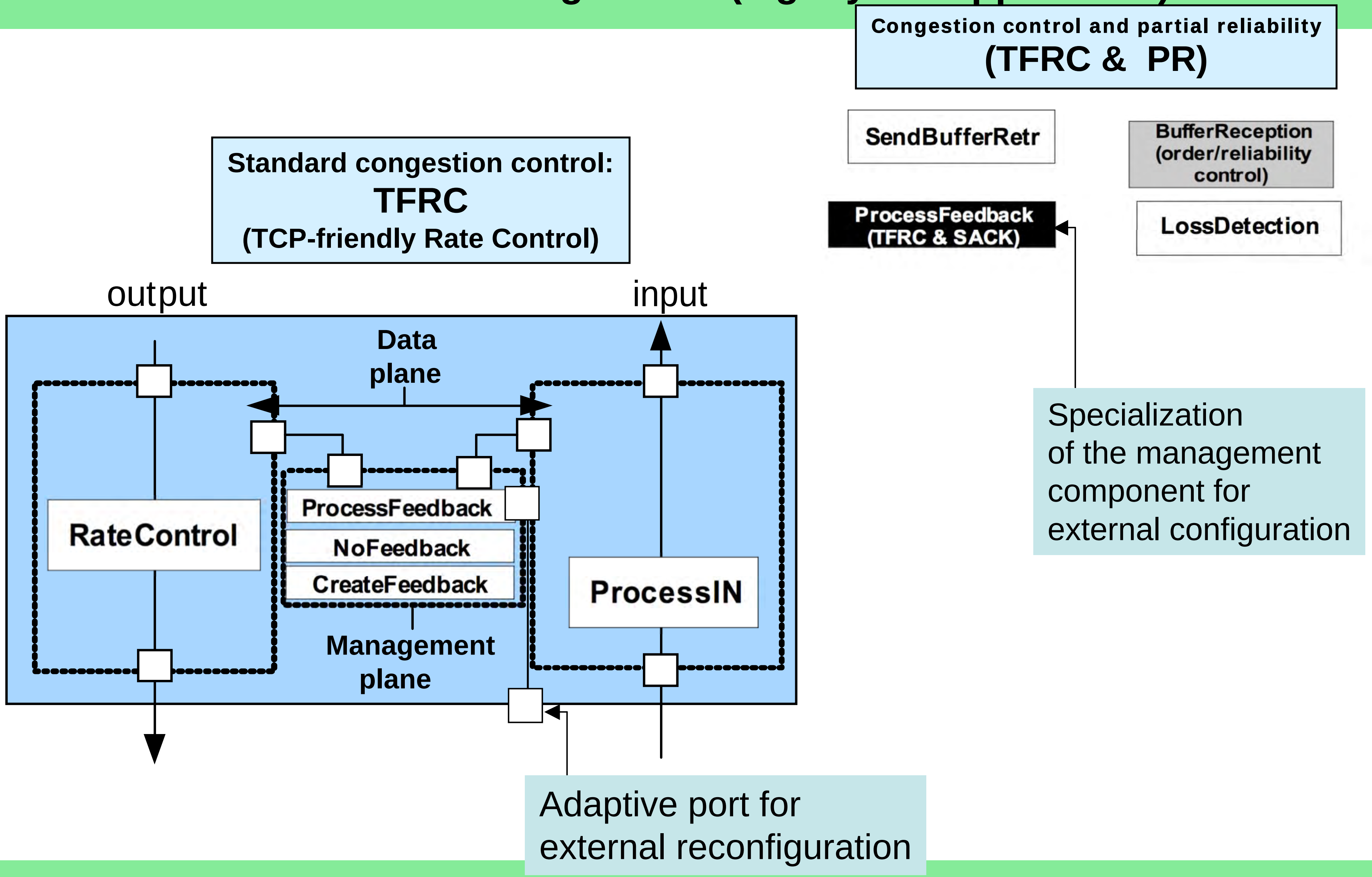
Specialization of a management component

Specializations based on various rate control algorithms
(Best-effort and guaranteed network services)

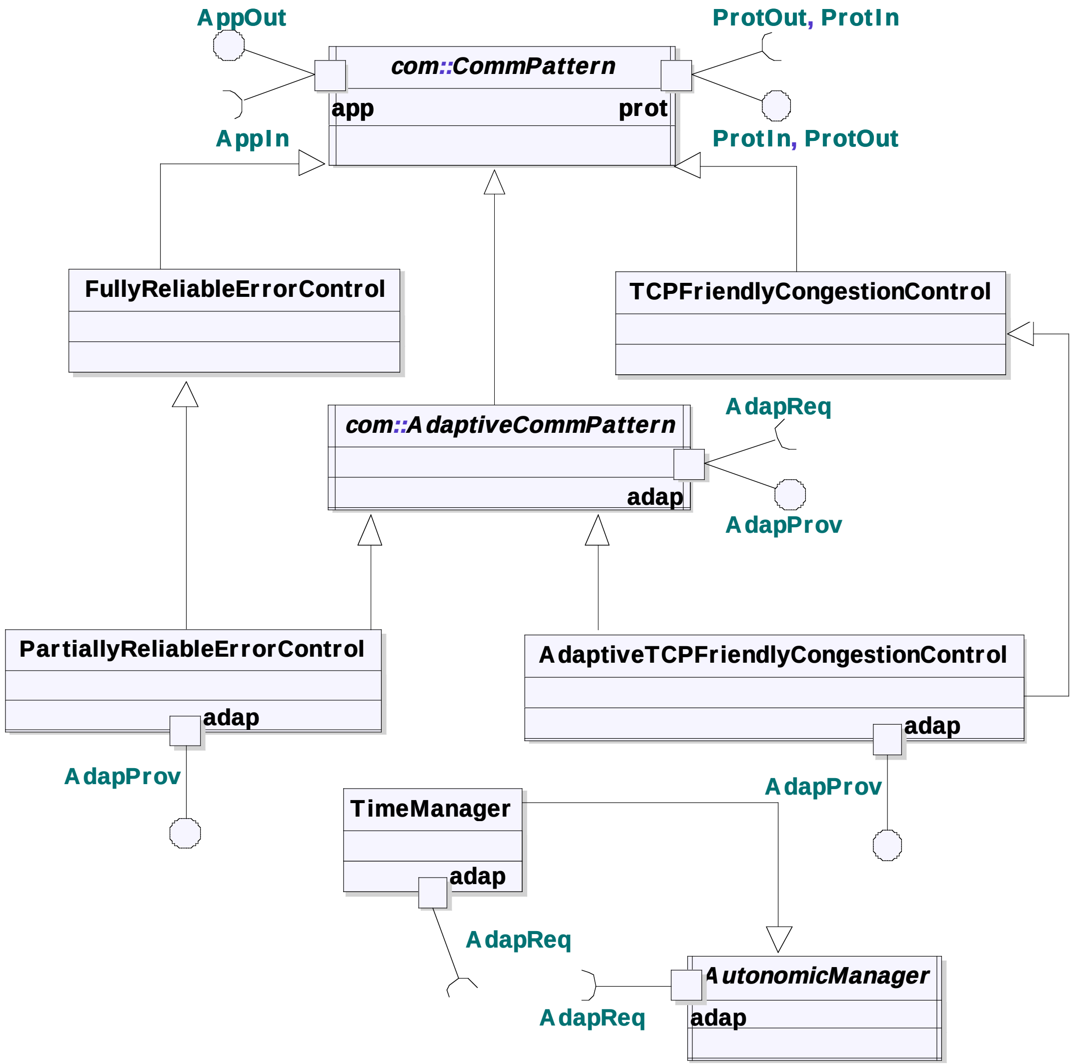
Adaptive composite communication pattern



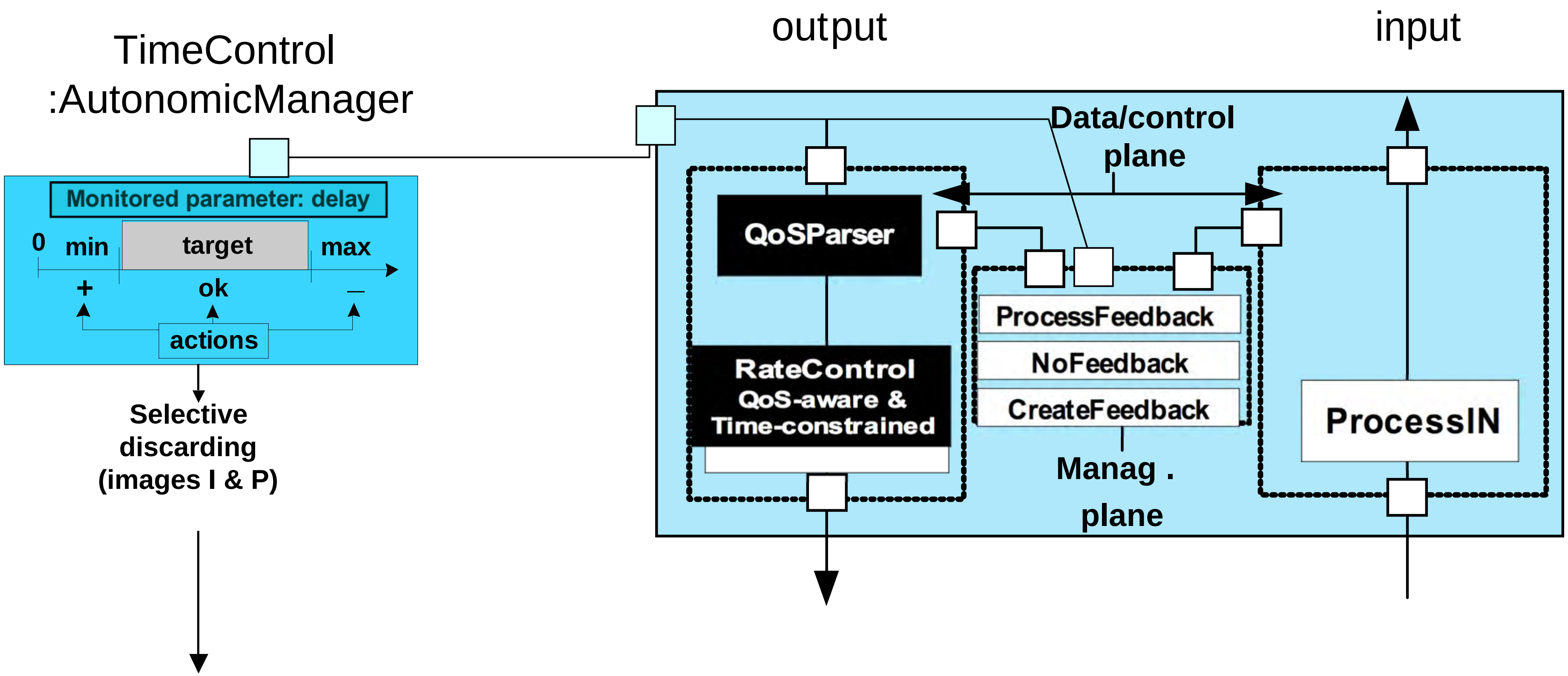
Example3: Adaptive port for external reconfiguration (e.g. by the application)



Autonomic components



Example4: Autonomic Manager and adaptive component



H.263 VIDEO	Partial reliability (degradation)	Time constraints
Interactive :	(I,P) = {(100,100),(100,50),(50,0)	(Tmin,Tmax)= (25,400)

Example5: More specializations/compositions

Congestion control and partial reliability
(TFRC & PR)

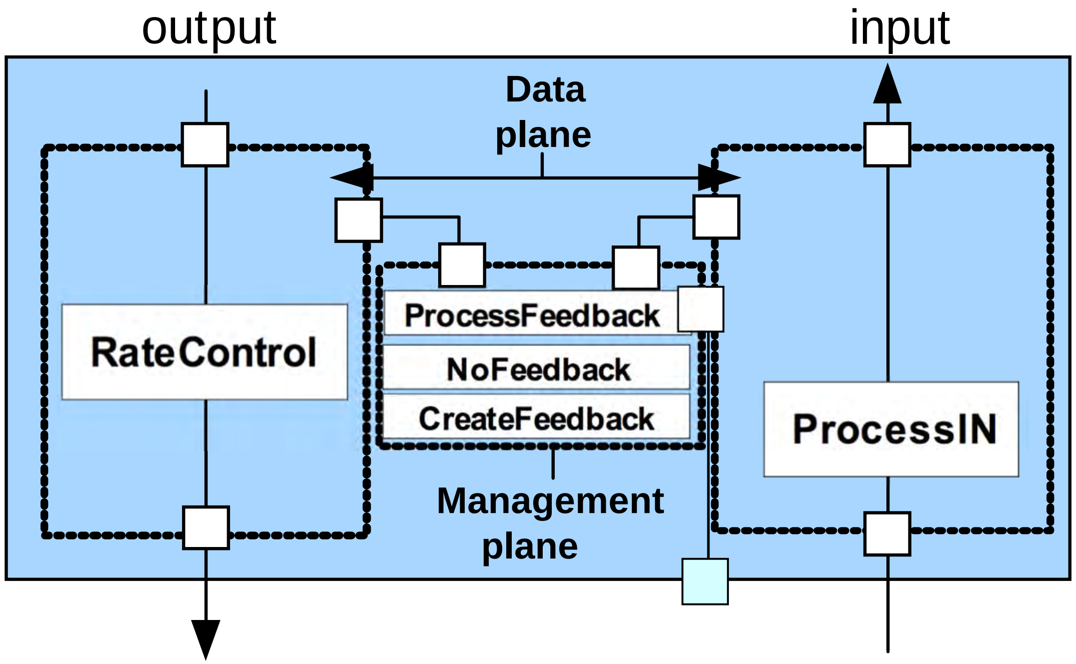
SendBufferRetr

BufferReception (order/reliability control)

ProcessFeedback (TFRC & SACK)

LossDetection

Standard congestion control:
TFRC
(TCP-friendly Rate Control)



Congestion control and differentiated and time-constrained partial reliability
TFRC & TD-PR

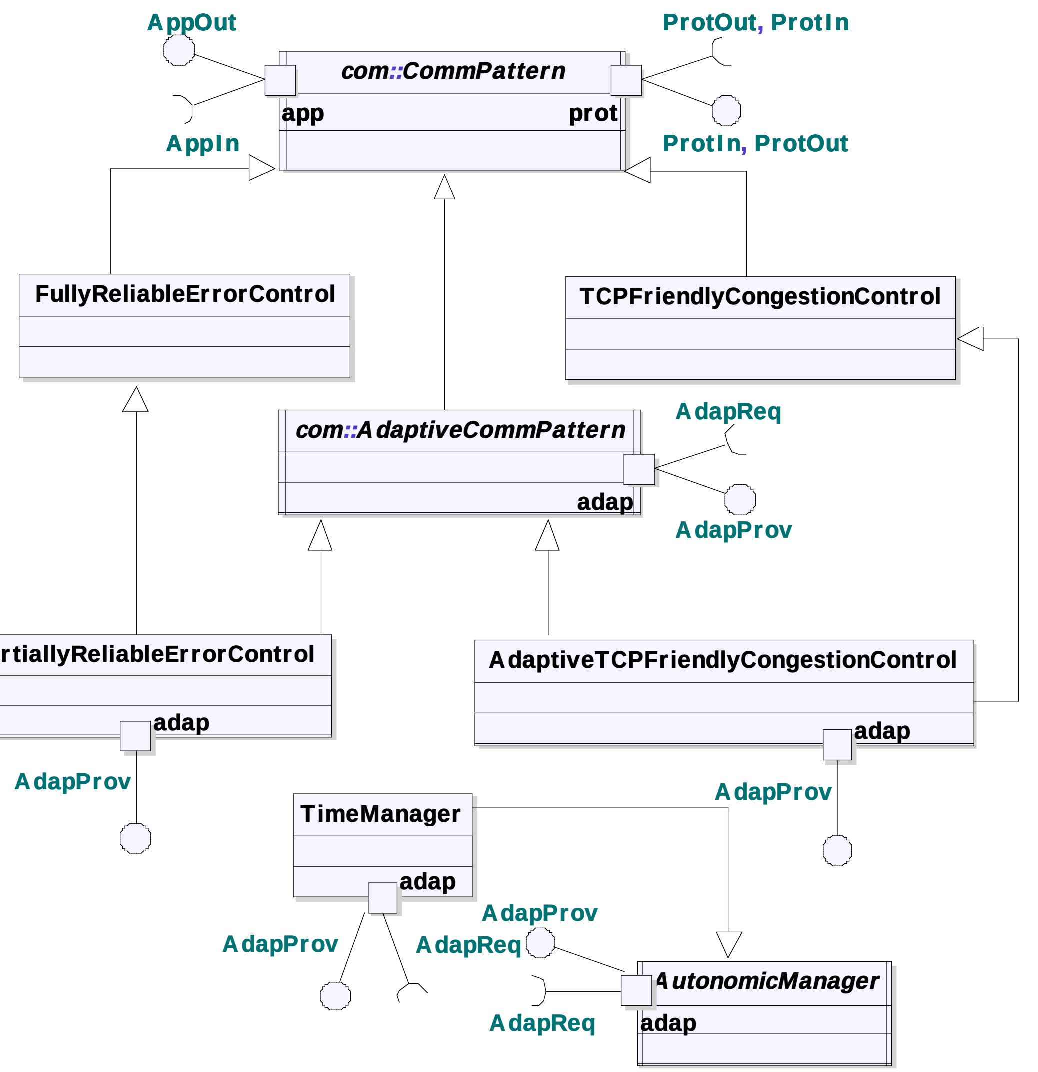
QoSParser

ProcessFeedback (TFRC & TD-SACK)

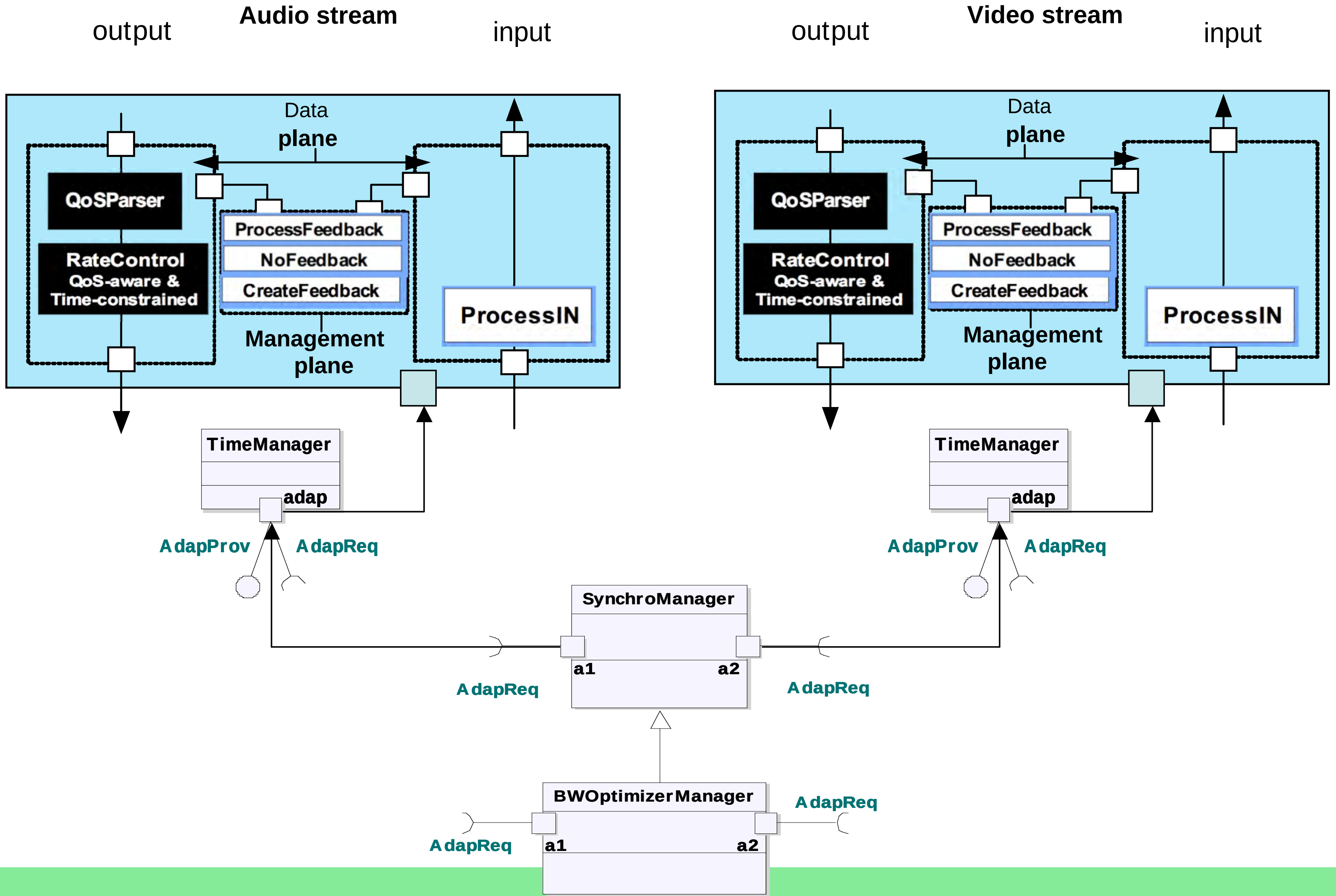
Congestion control (TD) et Partial reliability (TD)
TD-TFRC & TD-PR

RateControl QoS-aware & Time-constrained

Composition of autonomic components



Example6: Autonomic manager for VoD application based on the composition of two TD-TFRC instances



Autonomic managers orchestration

intraflow: com::AutonomicManager

Implicit Packet Meta Header
e.g. H.264 streams

intraApp : com::AutonomicManager

Based on inter-flow priorities
Application reqs/preferences

intraSystem : com::AutonomicManager

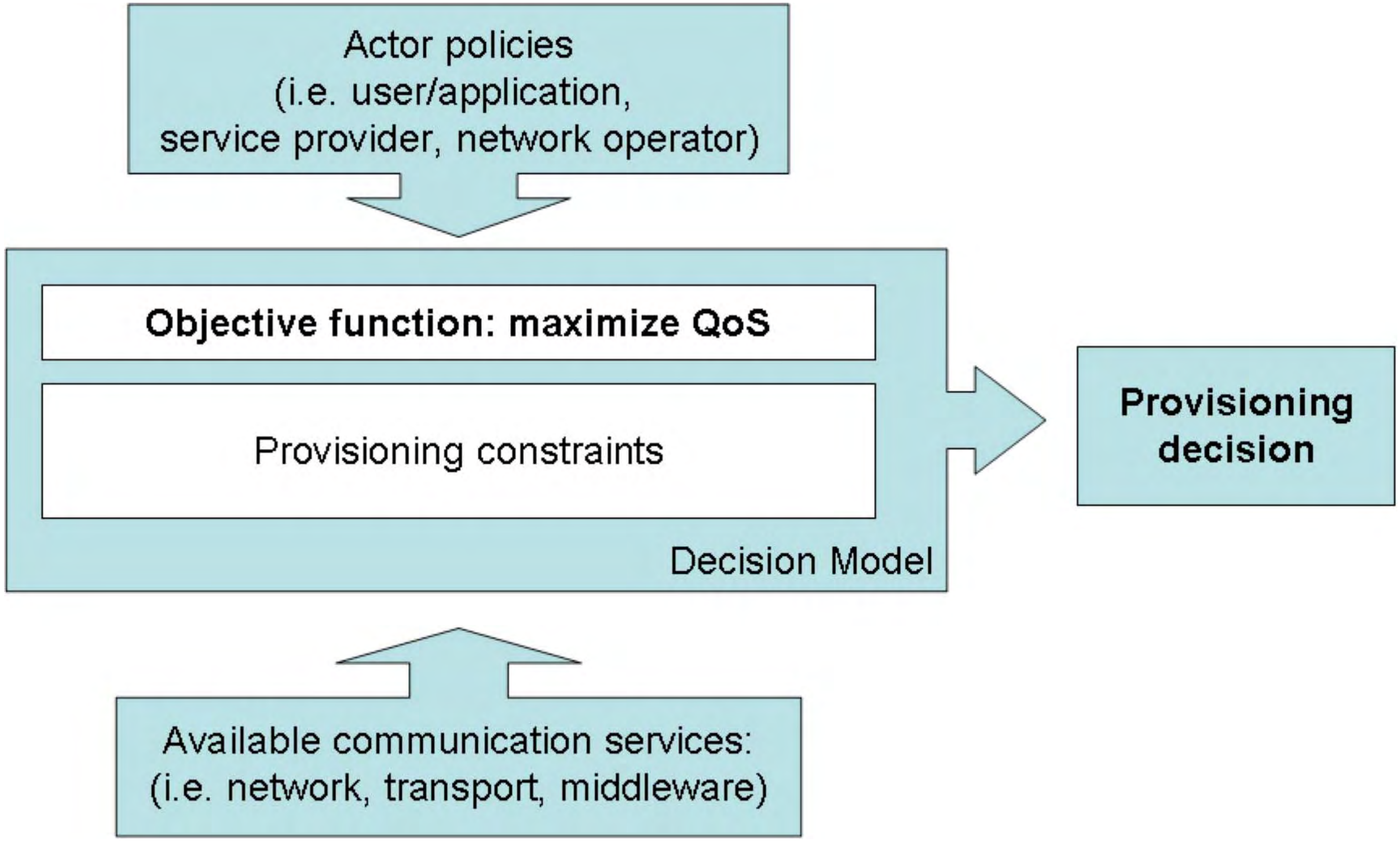
Based on inter-apps priorities
User reqs/preferences

intraGroup : com::AutonomicManager

Based on inter-user priorities
Group of user
reqs/preferences

QoS Semantic (reqs/preferences)

QoS provisioning model based



QoS provisioning model based

Definitions

F = {f1, …., fn} is the set of n multimedia flows competing for the available services

Sj={s1, …., sm}: end-t o-end communication services (single/composition)

xij={0,1} is the decision variable associated to the use of the service sj to transmit the flow fi

aj : Maximum availability of service sj

pi={1..pmax} priority associated to the flow fi (e.g. user or application preferences)

gij : Gain in quality associating the use of the service sj for the flow fi

cij : cost of using the service sj for the flow fi

<u>Decision model</u>

Objective function:

$$\max z = \sum_{j=1}^{m} \sum_{i=1}^{n} p_i * g_{ij} * x_{ij}$$

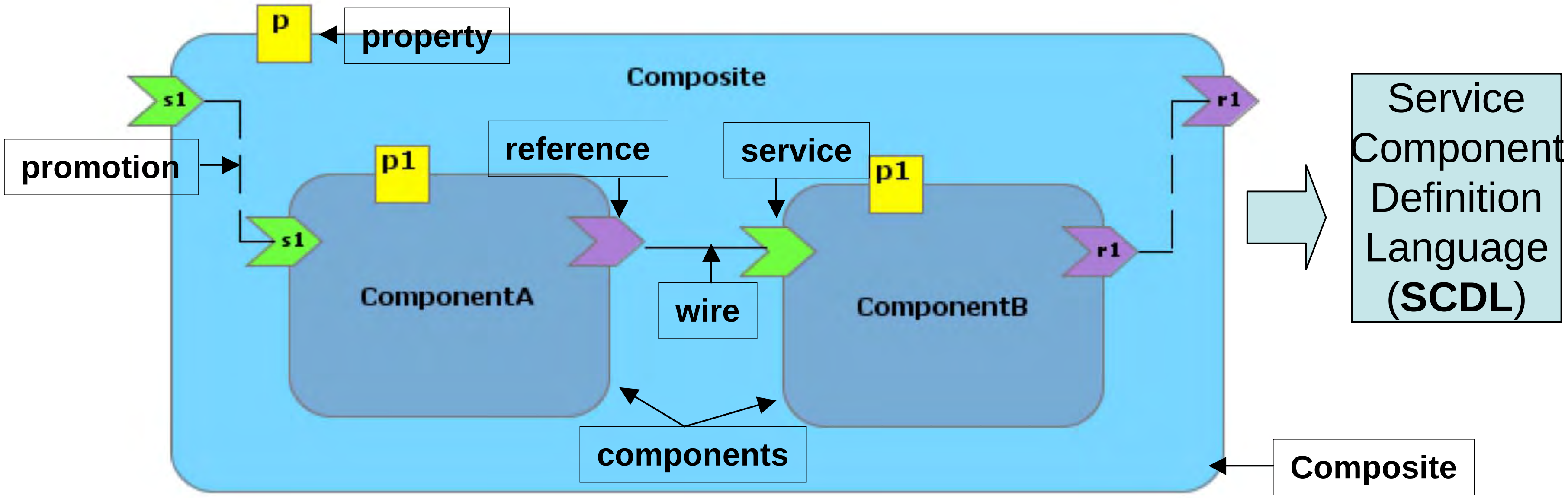
Subject to:

$$\sum_{j=1}^{m} c_{ij} * x_{ij} \le a_j$$

$$\sum_{j=1}^{m} x_{ij} \le 1, \forall i$$

Deployment: Service Component Architecture (SCA) approach

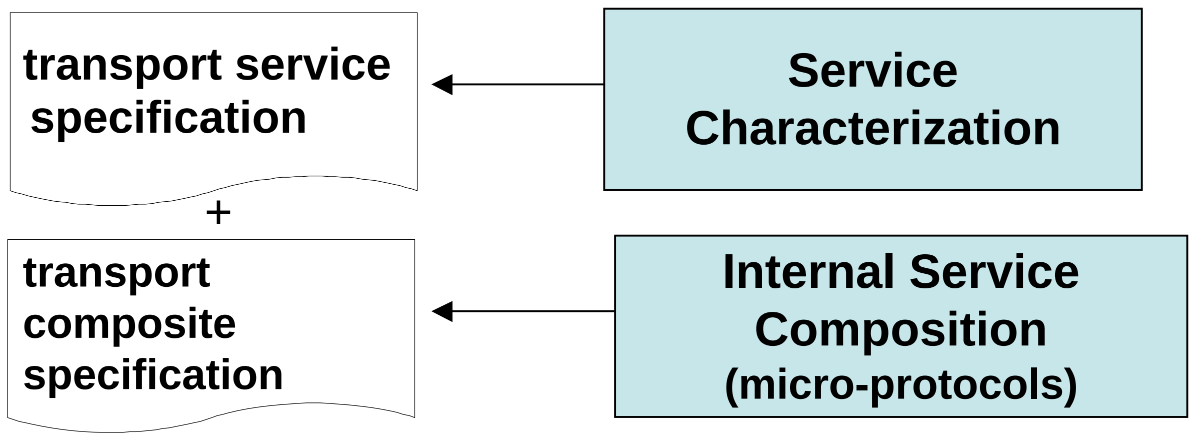
- Based on Service Component Architecture (SCA) [OASIS/OSOA]:
  - Provides a programming model for building systems based on SOA
- SCA definitions:
  - Service components: implement and use other services
  - Composites: assembly of components (including connections/bindings)



Semantic for Deployment

SCA approach based on
Ontologies for characterization and composition

- **QoS requirements**
- **Underlying network services**



Transport + Network services composition (EuQoS Project)

Network services \ Application profile	Stream *e.g. VoD*	Non-Stream *e.g. file transfert*
RT	ETP[] = UDP	ETP[EC]
NRT	ETP[gTFRC]	ETP[gTFRC+EC]
BE	ETP[TFRC+DT]	ETP[TFRC+DT+EC]

Conclusions and perspectives

- Communication Patterns specified (available U2 model)
  - Basic transport services
  - Adaptation
  - Autonomic management
  - Composition and deployment
- Semantic
  - Requirements and preferences
  - Composition (components and services)
- Projects/perspectives
  - Feel@home project: unicast (intra-inter homes) + mcast
  - Studies for deployment in resource-constrained (e.g. sensor transport services)

Thanks,

Questions…?