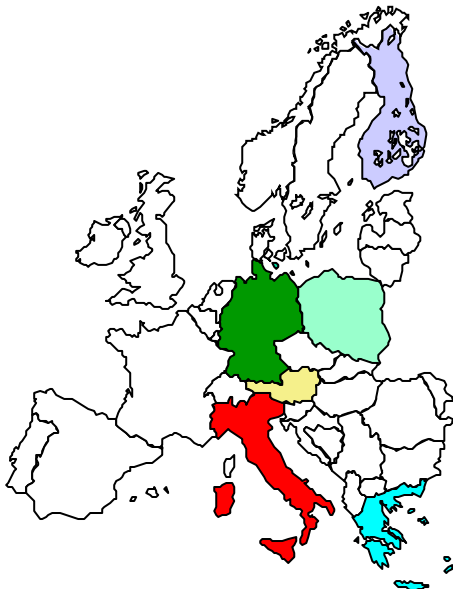


## Adaptive Resource Control for QoS Using an IP-based Layered Architecture

TEQUILA Workshop  
25-26 Jan 2001  
Amsterdam

Internet Design for SLS Delivery



*Bert F. Koch*

**SIEMENS**

*Stefano Salsano*



<http://www-st.inf.tu-dresden.de/aquila/>

## Outline

- Project Introduction
- AQUILA QoS Architecture
- Traffic Engineering Aspects
- Further Project Activities
- AQUILA Approach to SLS
- Future Plans

# Outline

- Project Introduction
- AQUILA QoS Architecture
- Traffic Engineering Aspects
- Further Project Activities
- AQUILA Approach to SLS
- Future Plans

## Consortium

**SAG**      **Siemens (Co-ordinator), Germany**

**I&C  
manufacturer**

**BAG**      **Bertelsmann mediaSystems, Germany**

**DTA**      **T-Nova Deutsche Telekom, Germany**

**TAA**      **Telekom Austria, Austria**

**ELI**      **Elisa Communications, Finland**

**TPS**      **Polish Telecom, Poland**

**Internet Service  
Providers  
and  
Network Operators**

**NTU**      **National Technical University of Athens, Greece**

**WUT**      **Warsaw University of Technology, Poland**

**COR**      **CoRiTel, Italy**

**TUD**      **Dresden University of Technology, Germany**

**SPU**      **Salzburg Research, Austria**

**Universities  
and  
Research  
Institutes**

**QSY**      **Q-Systems, Greece**

**Web application  
provider**

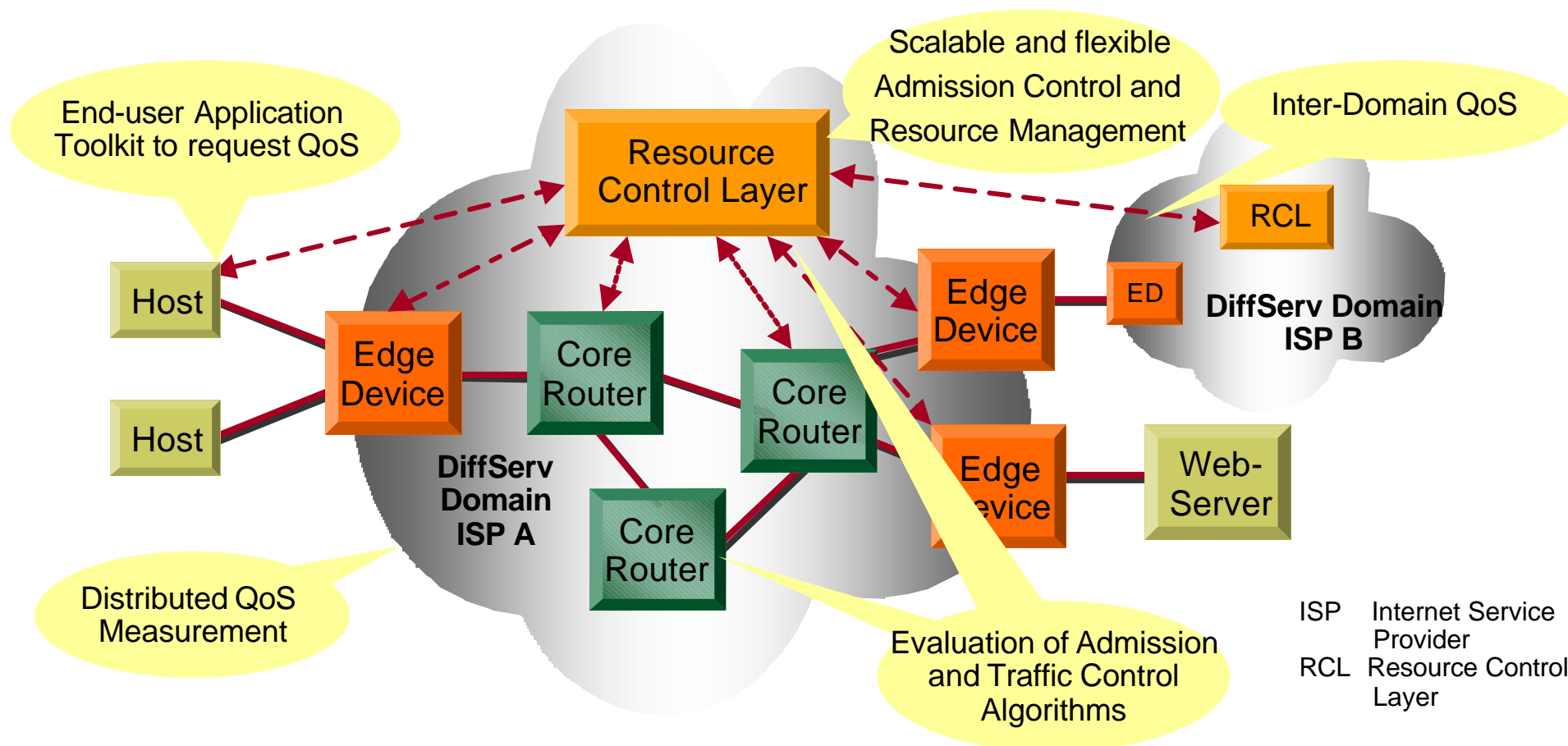
## Work Break Down Structure

- **Workpackage 0: Project Management**
- **Workpackage Group 1: System Architecture and Traffic Issues**
  - WP 1.1: Requirement Analysis
  - WP 1.2: Specification
  - WP 1.3: Traffic Studies and Engineering
- **Workpackage Group 2: Prototype Implementation**
  - WP 2.1: Service and Resource Control
  - WP 2.2: QoS aware Applications and User Services
  - WP 2.3: Distributed QoS Measurements
- **Workpackage Group 3: Integration and Trial**
  - WP 3.1: System and Network Integration
  - WP 3.2: Trials and Measurements
  - WP 3.3: Exploitation and Business Models

## Main Objectives

- Investigate dynamic end-to-end QoS Provisioning in IP Networks
- Implement Prototypes of a QoS Architecture for a Carrier Grade DiffServ Core Network
- Support a wide Range of Applications by providing a QoS Toolkit / API
- Continuously analyse Customer Requirements, Market Situations and Technological Trends and develop Business Models
- Contribute to Standardisation Bodies like IETF, ITU, ETSI, etc.

## Main Innovations



## end-to-end Quality of Service

## Design Goals

### ■ Scalable Architecture

- Distributed building blocks
- Autonomous operation of elements

### ■ Failure Proof

- Failure of an element should only degrade (if at all), not disable the operation of other elements

### ■ Based on DiffServ Core Network

- Use of existing, commercial routers
- Enable migration path from current networks



## Outline

- Project Introduction
- AQUILA QoS Architecture
- Traffic Engineering Aspects
- Further Project Activities
- AQUILA Approach to SLS
- Future Plans

## Resource Control Layer (RCL)

### ■ Tasks of the Resource Control Layer

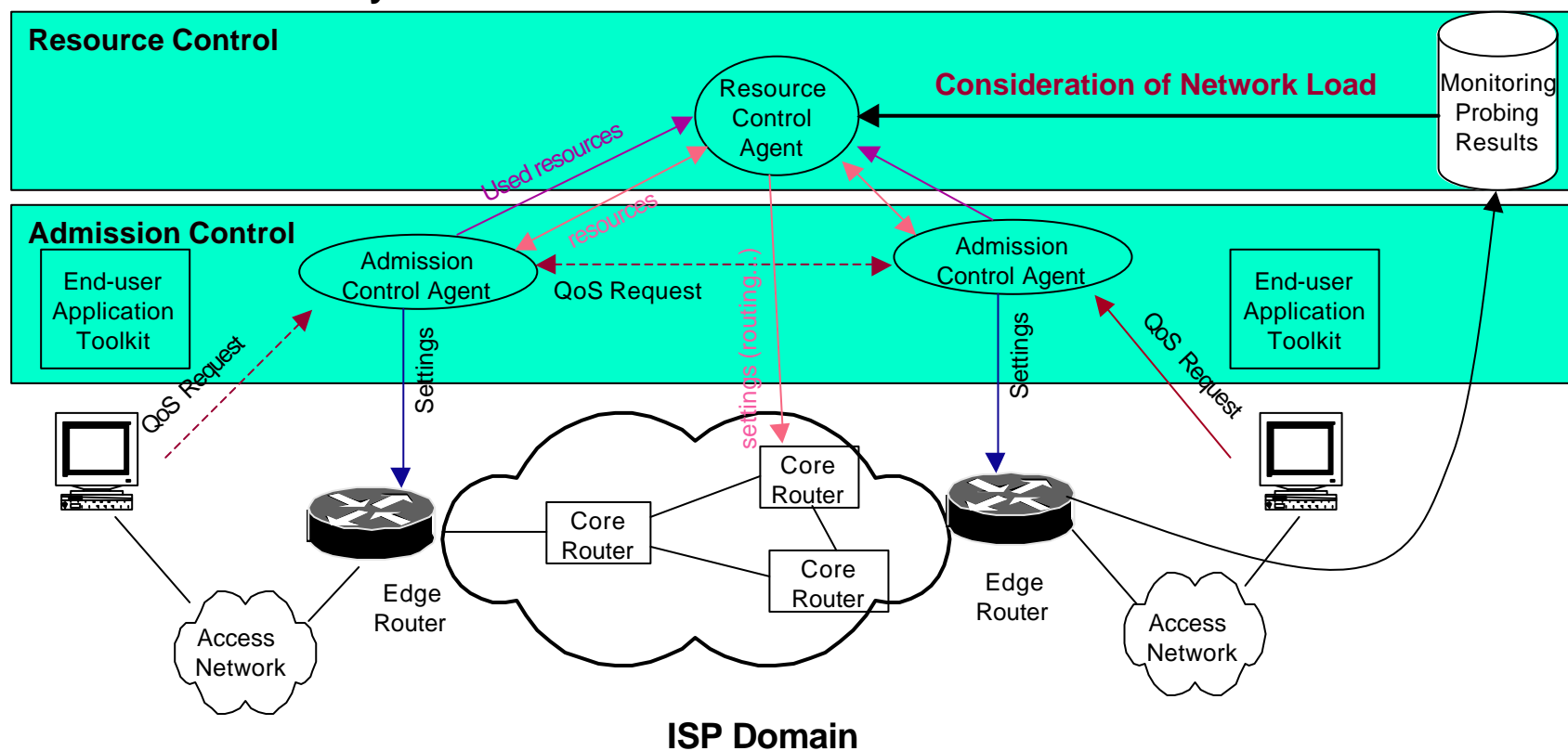
- Admission Control to limit the amount of prioritised traffic
- Resource Management
- QoS Interface

### ■ Design Goals

- Simpler than ATM (no explicit reservation along the data path)
- Scalable approach for Admission Control (distributed Admission Control, separation of Admission Control and Resource Management)

# Scalable Architecture for RCL

## Resource Control Layer



## Resource Control Layer (1)

### ■ Basic Idea of DiffServ Network

- Provide some (fixed) prioritised traffic classes within the network
- Guarantee QoS by limiting amount of prioritised traffic at the network edge (limited resources)

### ■ Additional Benefit of the Resource Control Layer

- Dynamically shift resources between network edges ( $P$  resource pools)
- Take into account the actual resource usage of the network (2nd Trial)

## Resource Control Layer (2)

### ■ Admission Control Agent

- Authenticates user
- Authorises and checks request
- Locates ingress and/or egress edge router (IP roles)
- Requests resources from the resource control agent
- Admits / rejects new flows
- Installs policies in ingress router

### ■ Resource Control Agent

- Manages resources
- Checks availability of requested resources
- (Re-)distributes resources as needed

## Resource Pools

### ■ Resource Limits

- Limit amount of QoS traffic from each edge router

### ■ Group neighboured Routers

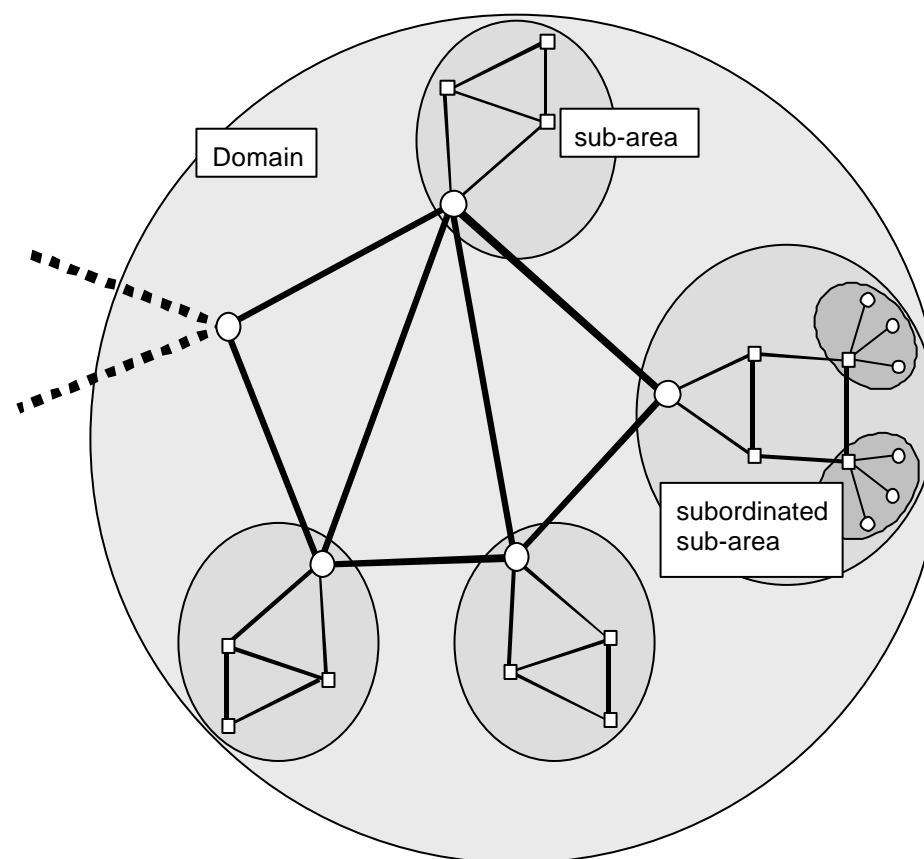
- Limit amount of QoS traffic from each group

### ■ Dynamic Distribution

- Dynamically shift resources within group

### ■ Hierarchical Structure

- “Groups of groups”



## Outline

- Project Introduction
- AQUILA QoS Architecture
- Traffic Engineering Aspects
- Further Project Activities
- AQUILA Approach to SLS
- Future Plans

## Network Services

### ■ Characterisation

- Delivery of services to the customer
- Defined by the network operator
- Provides a specific QoS, expressed by statistical or deterministic statements about delay, loss, ...

### ■ Implementation

- A network service is implemented by a traffic class

### ■ Usage

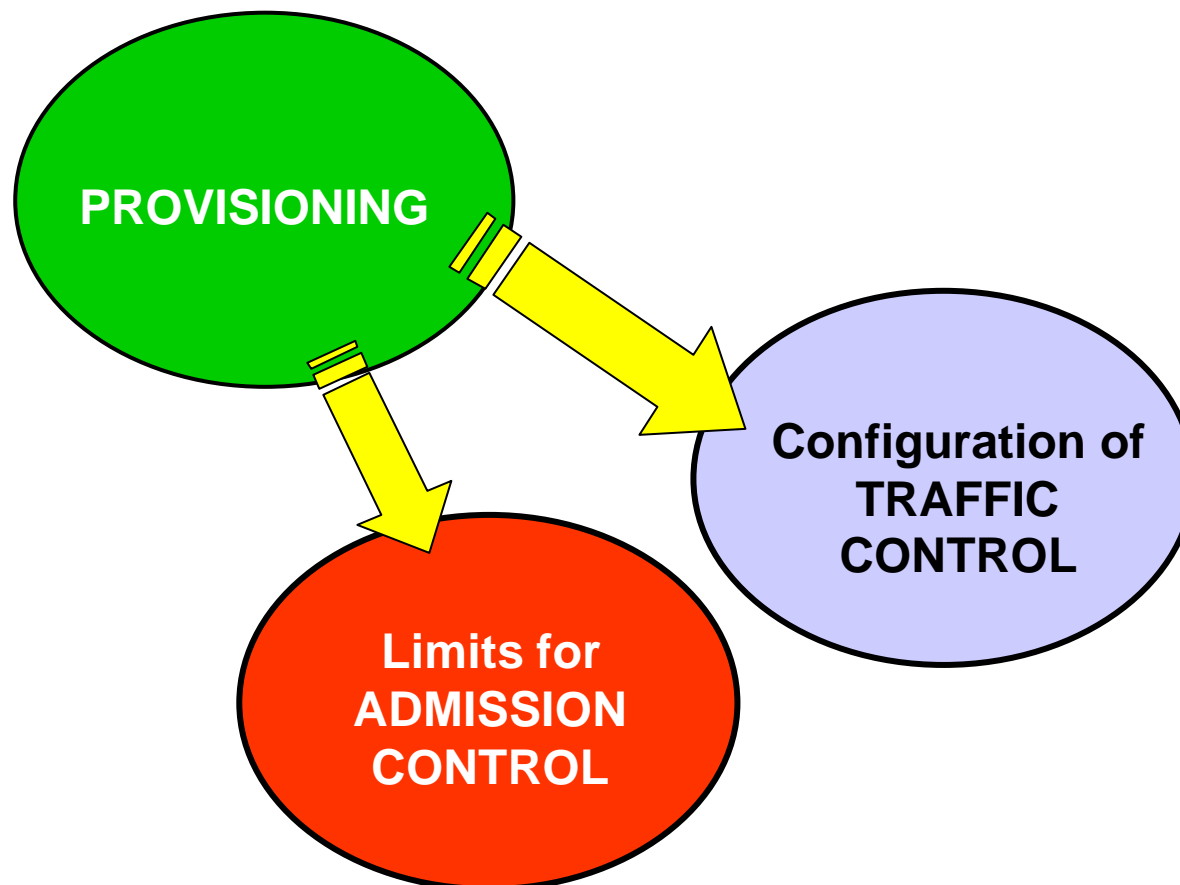
- The End-user Application Toolkit maps application demands to network services

### ■ Example: Premium CBR for IP Telephony and Voice Trunking

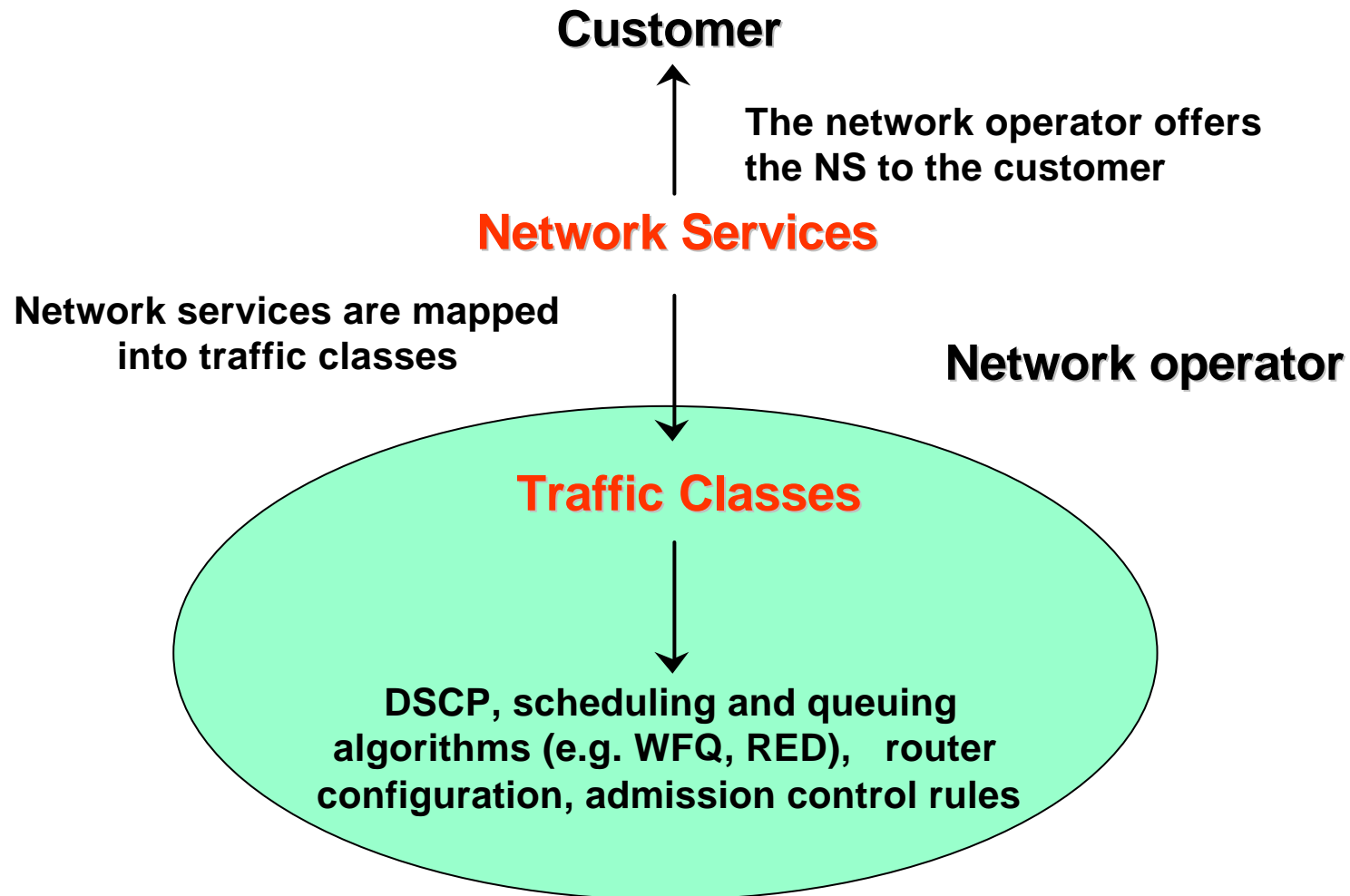
⌘ **Goal: only a few Network Services to allow clear service  
Differentiation (wrt QoS objectives)**



## Overview of Traffic Handling Approach



## Traffic Classes (1)

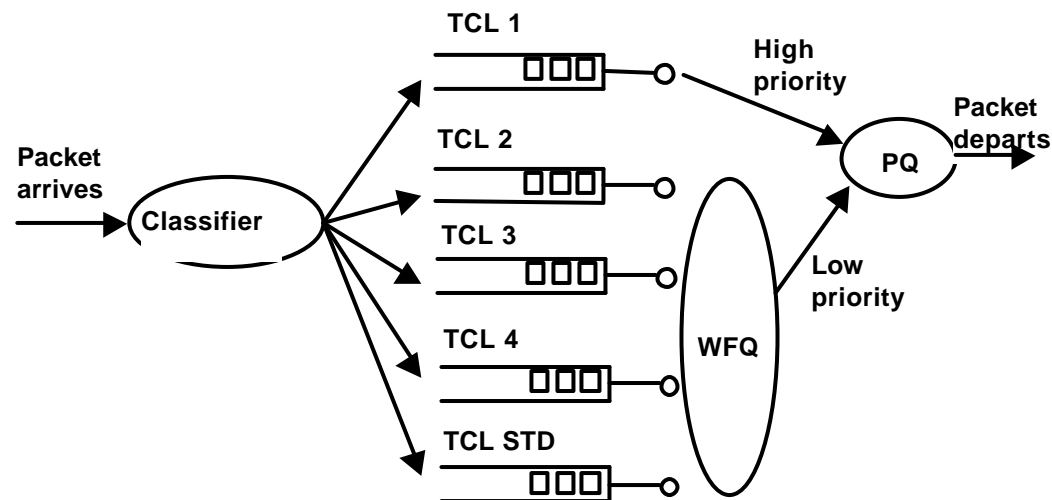


## Traffic Classes (2)

### ■ Five Traffic Classes have been specified

Network service	Premium CBR	Premium VBR	Premium MultiMedia	Premium Mission Critical	Standard
Traffic class	TCL 1	TCL 2	TCL 3	TCL 4	TCL STD

### ■ ... as well as the related Traffic Control Mechanisms in the Routers



# Admission Control

## ■ Traffic Characterisation

based on token bucket parameters declaration

## ■ Admission Control Algorithms

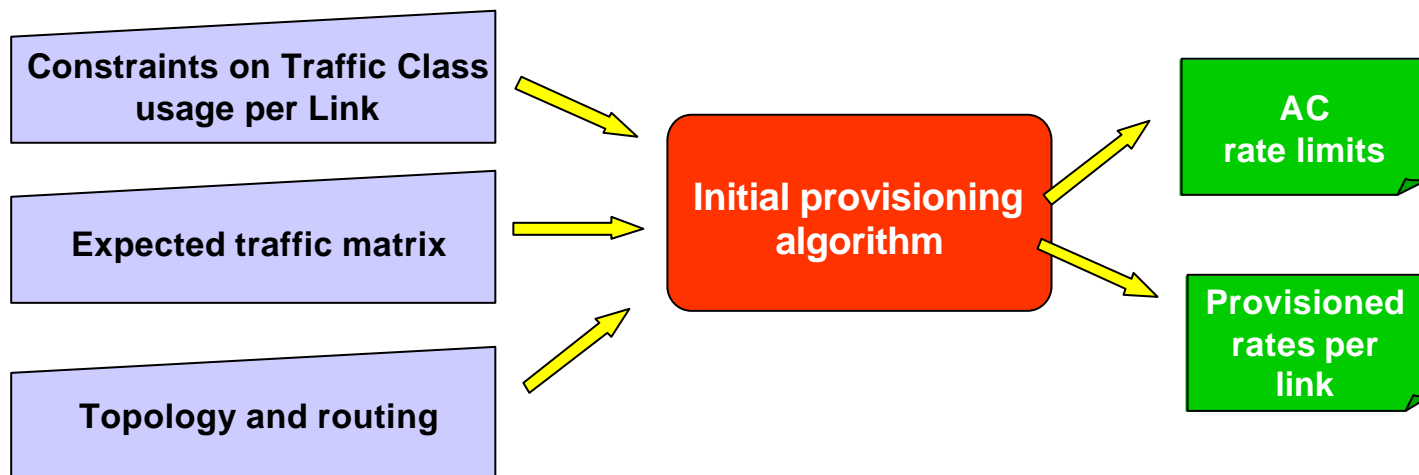
take into account the traffic parameters specified in the reservation request and compare with the provisioned admission control rate limits and the already allocated resources.

## ■ Specific Admission Control Algorithms

have been defined for the different traffic classes and different (high speed / low speed) access links.

# Provisioning

## ■ Initial Provisioning



## ■ Building Resource Pools

- Resource pools are built when it is useful to dynamically share a bottleneck link among a set of access links

## Outline

- Project Introduction
- AQUILA QoS Architecture
- Traffic Engineering Aspects
- Further Project Activities
- AQUILA Approach to SLS
- Future Plans

## End-user Application Toolkit (EAT)

### ■ Middleware between QoS Network and Application

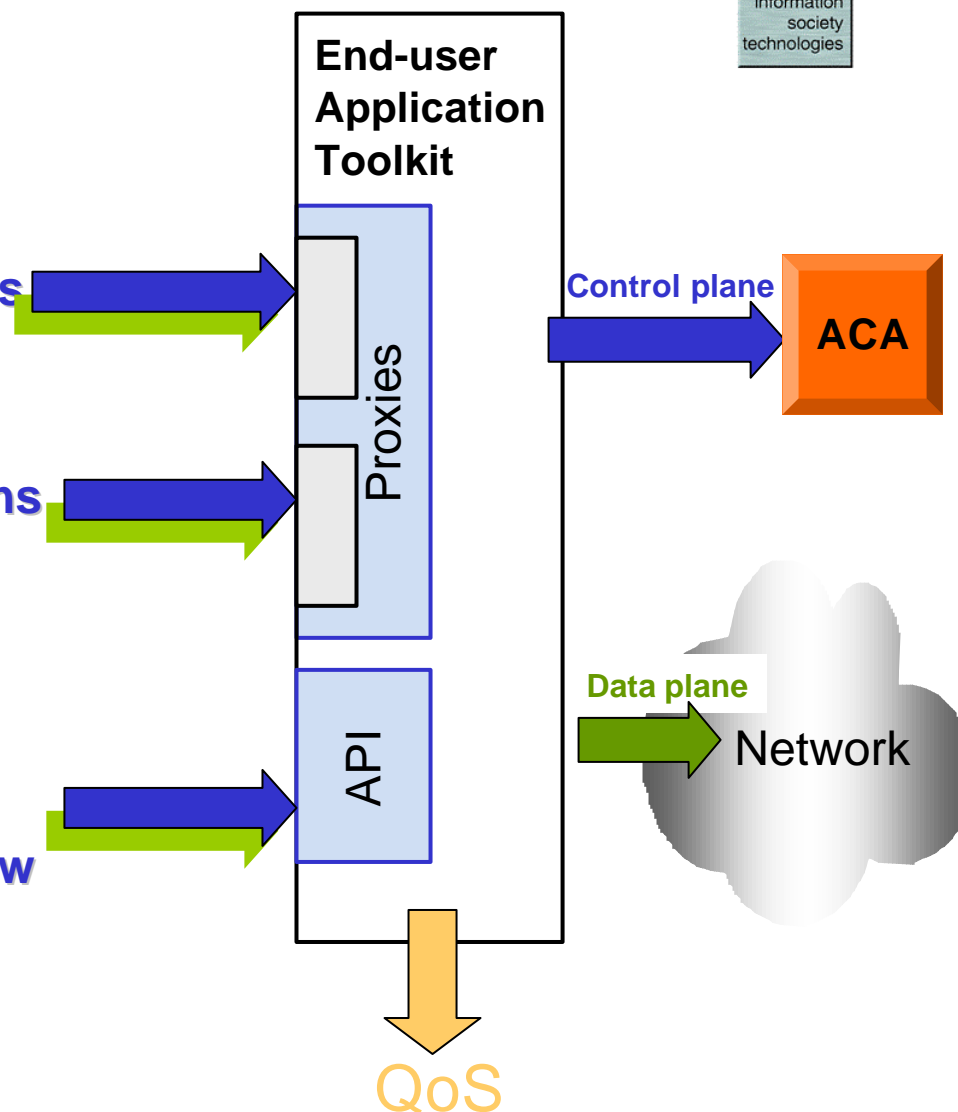
- Front end for access network
- QoS portal for application (legacy and QoS aware)
- Alternative, flexible approach for evaluating QoS reservations

### ■ EAT is requester of QoS reservation

- The requester may be the sender, receiver or a third party
- The requester initiates the reservation
- The requester is charged for the service

## Objectives

- **Enable Access to QoS to non QoS aware Legacy Applications**
- **Support QoS aware Applications (RSVP, DiffServ) / Support various QoS Request Methods**
- **Provide a Methodology and a Programming Interface to support the Construction of new QoS aware Applications**
- **Provide an End-user friendly QoS Access**



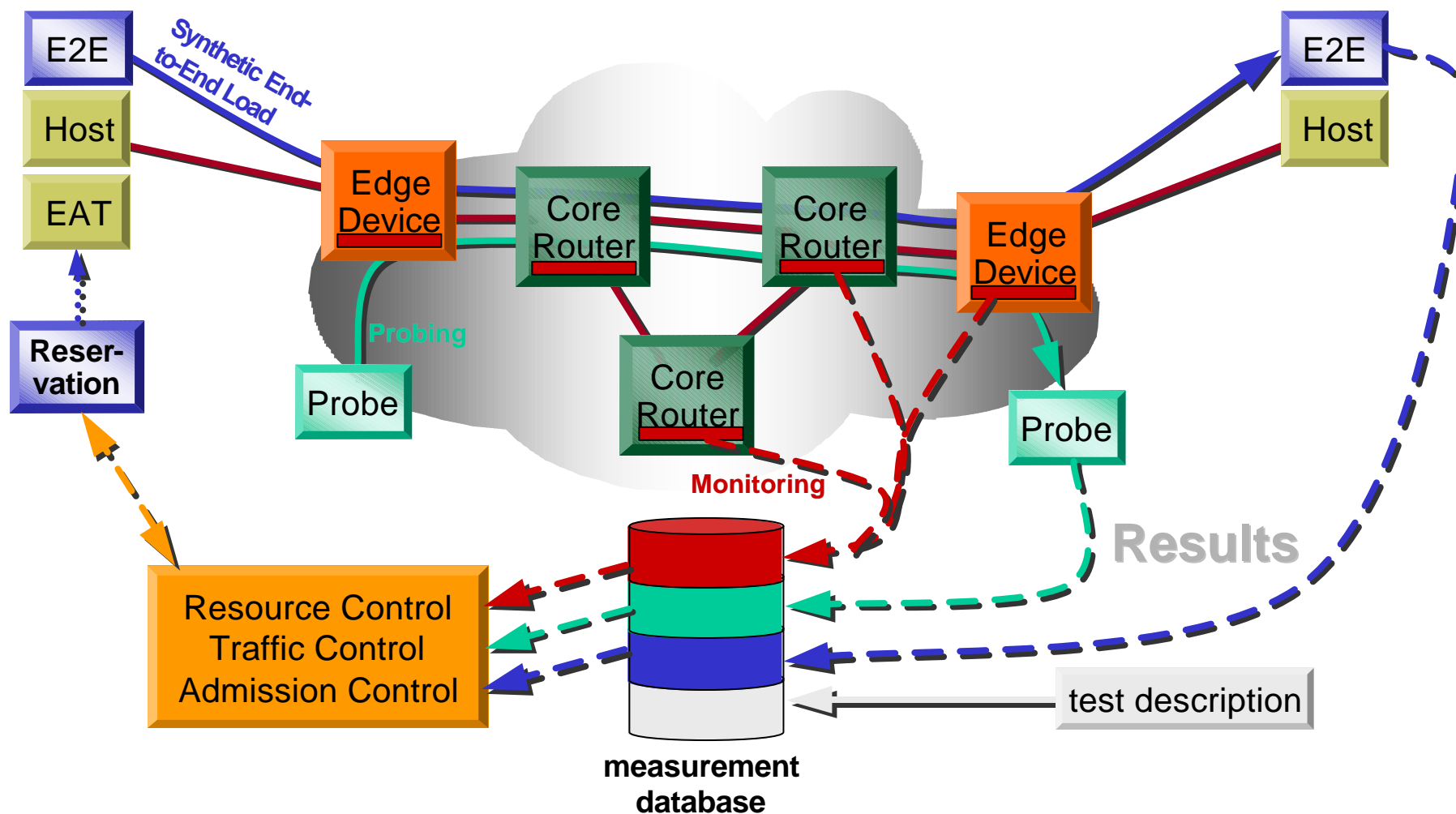


# Distributed QoS Measurements

## Packet Level

- **Measurement of end-to-end Parameters via Probes**
  - Examples: one-way delay, delay variation, packet loss
- **Collection of Performance Monitoring and QoS Parameters from Routers**
- **Definition and Implementation of Synthetic Flow Load Generators for Measurements of end-to-end QoS**
- **Storage of all Measurement Data in Measurement Database**

## Measurements



## Tasks in the First Trial (Feb/Mar 2001)

- Use Real Life Applications to “get a feel”
- To supplement the Environment with generated Quality Traffic
- To measure the achieved Network Traffic Characteristics
- To complement the measured Data with subjective User Impressions of the Applications' Performance/Behaviour
- To evaluate the measured Data
- To process all Information and provide Feedback to the othre WPs for further Developments/Refinements
- To support the Dissemination of the Results

## Trial at Three Different Sites

### ■ Warsaw (Polish Telecom)

- Reference Site
- Special Focus on Streaming Media / Video on Demand

### ■ Helsinki (Elisa Communications)

- Site with various Access Technologies (ADSL, 10 Mbps Ethernet, WLAN)
- Special Focus on realistic customer/end-user environments and different environments and usage (home, office, public access zone)

### ■ Vienna (Telekom Austria)

- Site with homogenous Layer 2 (Ethernet)
- Special Focus on low bandwidth real-time applications, VoIP, Multi-user network games

## Outline

- Project Introduction
- AQUILA QoS Architecture
- Traffic Engineering Aspects
- Further Project Activities
- **AQUILA Approach to SLS**
- Future Plans

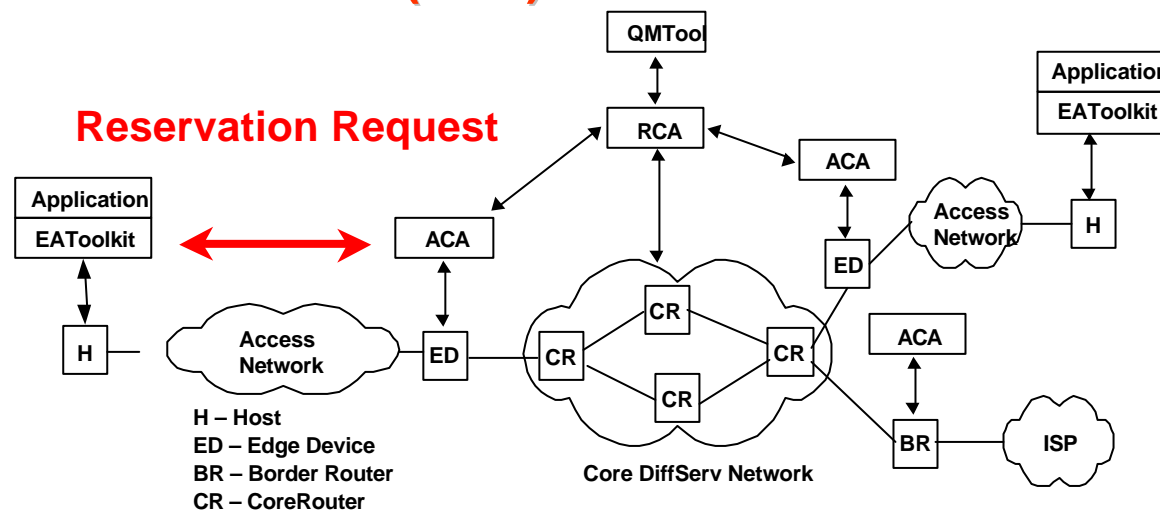
## Standardisation Issues

### ■ IETF Activities

- Joint preparation with TEQUILA and CADENUS for a BoF session at the Dec 2000 IETF Meeting in San Diego
- Goal is to start a new Working Group on **Service Level Specification** (SLS)
- AQUILA EAT-ACA interface modelled in accordance to the TEQUILA draft (*draft-tequila-sls-00.txt*)
- AQUILA draft on Network Services submitted (*draft-salsano-aquila-sls-00.txt*)
- Contribution to SLS framework draft (*draft-manyfolks-sls-framework-00.txt*)

## AQUILA Approach to SLS

- In the AQUILA architecture a “Reservation Request” is sent to the Admission Control Agent (ACA) by the “End-user Application Toolkit” (EAT)



- A Reservation Request specifies a Service Level Agreement

## AQUILA SLS Draft to IETF

- **Following input from TEQUILA (draft “Service Level Specification Semantics and Parameters”) we have described the semantic content of our SLS trying to align the terminology (see section 3 of our draft)**
- **Most of the concepts are very similar**
- **The most interesting difference is the concept of “predefined SLS types”**



## Semantic Content of SLS

### draft-aquila

- SLS type
- Scope
- Flow Identification
- Traffic description and conformance test
- -
- Performance Guarantees
- Service schedule
- -

### draft-tequila

- -
- Scope
- Flow Identification
- Traffic conformance test
- Excess Treatment
- Performance Guarantees
- Service schedule
- Reliability

## Network Services

- There has been lot of discussion in the SLS mailing list on the issue of “Well Known Services” vs. “Custom Services”
- “Well Known Services” restrict the definition of new services, but are very simple to handle
- “Custom” network services allow freedom in specifying the QoS parameters ... , but may be very difficult to handle

## Predefined SLS Types (1)

- **AQUILA Network Services are defined in terms of predefined values for the parameters (e.g. traffic descriptors, QoS requirements...) and of restrictions on the allowed combination of parameters**
- **A “predefined SLS type” is defined in terms of the generic SLS description**
- **A "predefined SLS type" simplifies the generic SLS description as it fixes values (or range of values) for a subset of the parameters in the generic SLS. It may also fix some relationships or dependencies between some parameters**

## Predefined SLS Types (2)

- This approach simplifies the SLS negotiation procedures and the SLS mapping into network internal mechanisms, yet it is more flexible than having “globally well known” services
- The framework is flexible enough to support both Custom Services and Predefined Services
- Predefined Services may play a fundamental role in SLS negotiation
- The use of Predefined Services is an option to the network provider

## Outline

- Project Introduction
- AQUILA QoS Architecture
- Traffic Engineering Aspects
- Further Project Activities
- AQUILA Approach to SLS
- Future Plans

## Future Project Plans (1)

### ■ Architecture

- Inter-Domain scenarios
  - SIBBS
  - SLS between ISPs
  - Reservation aggregation
- MPLS
  - VPN
  - QoS traffic engineering
  - MPLS-DiffServ interworking
- Multicast reservations
- Resource control loop (feedback from measurement to resource control)

## Future Project Plans (2)

### ■ Support of Applications

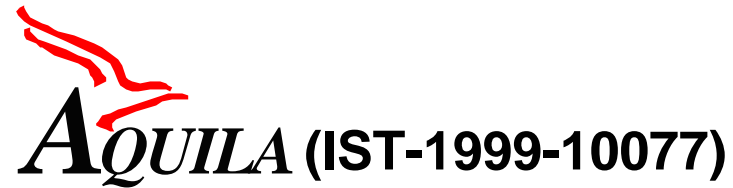
- RSVP as QoS signalling protocol
- Application Programming interface (API)

### ■ Management

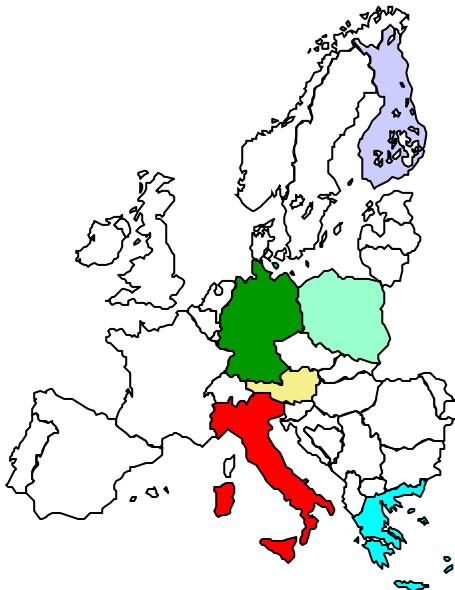
- Administration of network services
- Control of resource distribution parameters

### ■ Measurement

- Improved load generators for service verification
- Provide measurements for RCL feedback



**Adaptive Resource Control for QoS  
Using an IP-based Layered Architecture**



**Thank you for  
your attention !**

<http://www-st.inf.tu-dresden.de/aquila/>