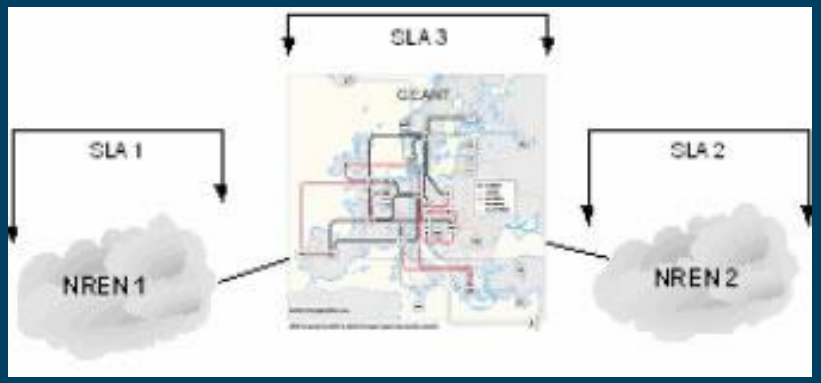
Connect. Communicate. Collaborate

SLA Module Implementation in AMPS

GEANT2 Workshop
June 2007
Cambridge UK

Mary Grammatikou
GRNET

The establishment of end-to-end SLAs for each PIP service instance from one edge of the extended PIP domain to another, is based on the intermediate per-domain SLAs.



The goal of each such SLA is to provide end-to-end control on PIP service in a multi-domain context. The end-to-end SLA has to be established so as to contain all related and necessary parameters for the service instance specification between the two endpoints requiring PIP service, regardless of the underlying network and intermediate domains.

Connect. Communicate. Collaborate

An SLA is a set of technical and non-technical parameters agreed between users and NRENs. We define an SLA component as two objects, namely Administrative Level Object and a Service Level Object:

- ALO (Administrative Level Object) part, as a set of administrative-legal parameters
- SLO (Service Level Object): a set of parameters and their corresponding values that define the technical parameters of PIP service provisioning to a PIP flow

Connect. Communicate. Collaborate

Administrative Level Object Parameters

- Full contact details, SLA Duration, Service Availability for the SLA Duration, PIP service availability estimation, Domain Acceptance Response Time, Reservation Request Lead Time, Failure Indication Response Time, Set of actions taken

Service Level Object Parameters

- PIP Description (IP Source-Destination Pair), Performance (One-Way Delay, IP Delay Variation, Bandwidth, Packet Loss, MTU), Reliability (Allowed Maximum Downtime, Time-To-Repair)

Connect. Communicate. Collaborate

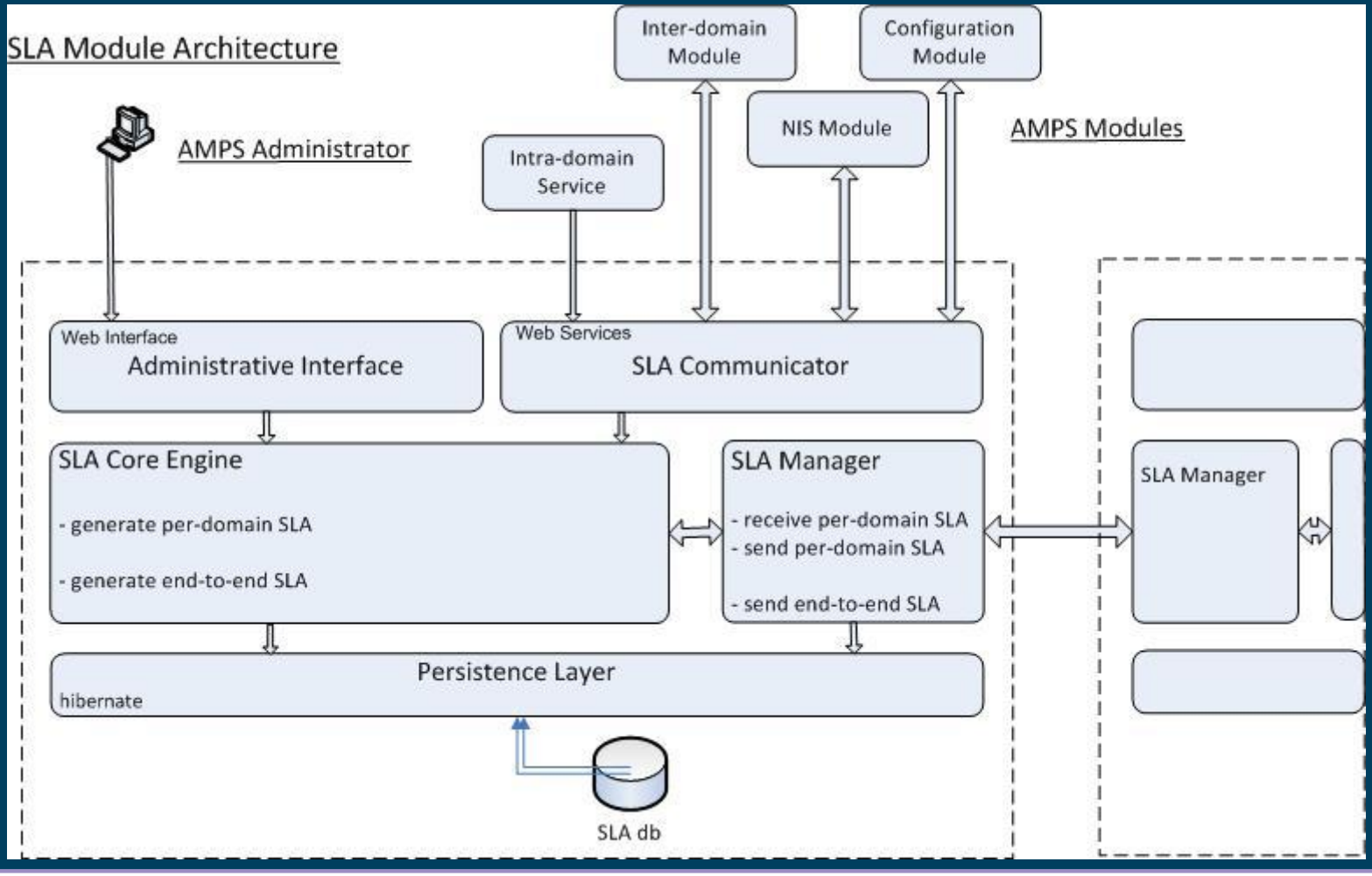
SLA Module Architecture

SLA Module APIs can be categorized into three categories:

- Administrative Interface: through a web interface AMPS administrator can manage and configure SLA module
- SLA Communicator: is responsible for the communication between SLA module and other AMPS's modules. By calling the appropriate web services (from NIS module and Configuration module) SLA module can collect required data.
- SLA Manager: through this entity SLA modules in different domains communicate with each other.

SLA Generation Process

- 1. Start of SLA generation process: SLA module receives a notification from Intra-domain service that a SLA must be established for a specific reservation.
- 2. SLA communicator interacts with other AMPS modules to retrieve required data, such as: capacity, MTU, size-of-queues.
- 3. SLA core engine generates per-domain SLA filling the corresponding Administrative and Service Level Objects and stores per-domain SLA to local database
- 4a. If this is not the first AMPS in the domain chain it sends the per-domain SLA to the SLA manager of the previous AMPS server.
- 5a. SLA manager receives the end-to-end SLA and it stores it to local database
- 4b. If this is the first AMPS in the domain chain SLA manager collects all the per-domain SLAs.
- 5b. SLA core engine generates end-to-end SLA with a merge of all per-domain SLAs.
- 6b. Stores end-to-end SLA to local database and sends it to all AMPS servers in the chain
- 7. End of SLA generation process and SLA module is ready to accept another SLA notification


SLA Module Architecture
Inter-domain Module
Configuration Module
NIS Module
AMPS Modules
AMPS Administrator
Intra-domain Service
Web Interface
Administrative Interface
Web Services
SLA Communicator
SLA Core Engine
- generate per-domain SLA
- generate end-to-end SLA
SLA Manager
- receive per-domain SLA
- send per-domain SLA
- send end-to-end SLA
SLA Manager
Persistence Layer
hibernate
SLA db

Connect. Communicate. Collaborate

Limitations

The following limitations apply for the AMPS SLA Module:

- SLAs are valid only as long as the AMPS assumptions assert, ie that routing will follow the path found by pathfinder. This may not be true in certain cases, eg outages, upgrades, static routes etc (All these should be well defined in Fault handling trouble ticket procedures field in ALO object)

- End-to-End SLAs are given for the part of the reservation path within the AMPS cloud. This should not be confused with the actual end-to-end SLA that the user will receive, which should also consider the SLAs of the non-AMPS domains within the end-to-end path, if they exist.

Connect. Communicate. Collaborate

Current status

- Design :
  - UML diagrams finished
- Implementation :
  - Persistence Layer currently implemented (Hibernate Objects and database)
  - In parallel we are installing and configuring perfsonar services for the monitoring and testing (we will need them for the testing and evaluation)

Connect. Communicate. Collaborate

Future Work

- We will continue the module implementation with the rest of the components (SLA manager, communicator, core engine and administrative interface)
- Bugs fixing
- Testing and evaluation using PerfoSonar and e2emonit frameworks
- We will use an application to test this implementation using GN2 network
- Extending the design using SLA not only for the reserved paths ? ?

GEANT2

DANTE

GÉANT2

Connect. Communicate. Collaborate

Thank You!

GÉANT2

GRNET
ΕΔΕΤ