CoreGRID

Managed by

European Research Consortium for Informatics and Mathematics
ERCIM

PROTOTYPE IMPLEMENTATION OF A DEMAND DRIVEN NETWORK MONITORING ARCHITECTURE

Augusto Ciuffoletti, Yari Marchetti
INFN-CNAF (Italy)

Antonis Papadogiannakis, Michalis Polychronakis
FORTH (Greece)

Summary

- Intro (3 slides)
- Architecture (11 slides)
- Prototype (5 slides)

Network Monitoring

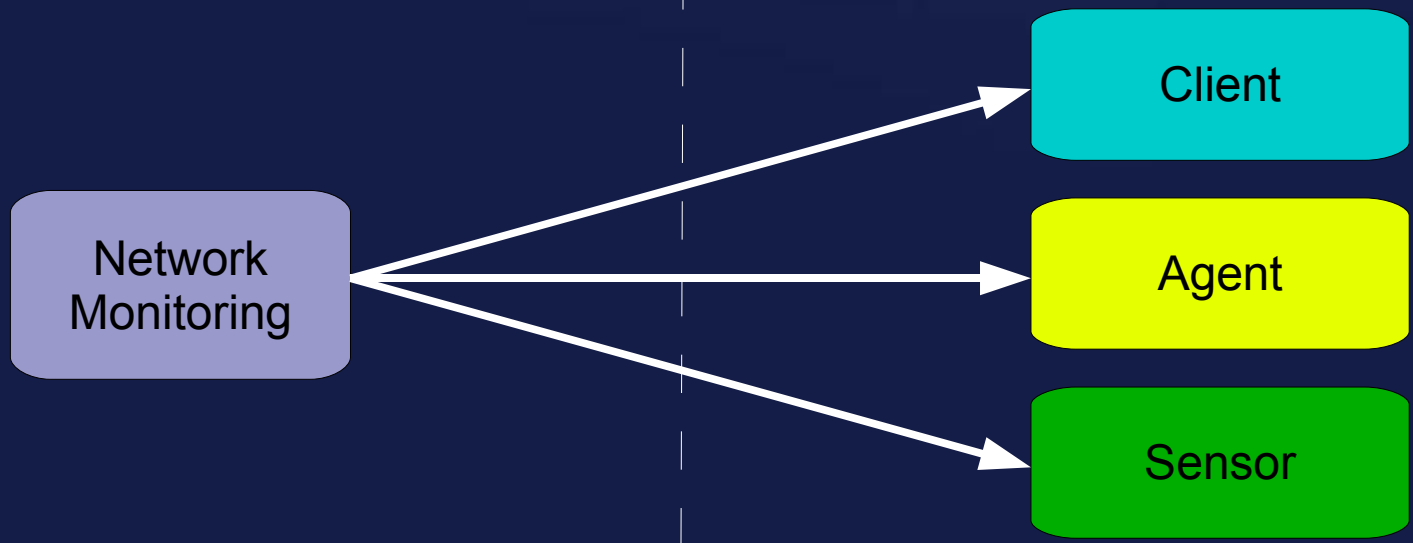
- Accounting
- Fault detection and QoS enforcement
- User side monitoring
- How To?
  - Dedicated tools
  - Timed measurements
  - Pre-scheduled sessions
- Presently mostly addressing admin usage

On demand network monitoring

- **Hardwired** monitoring exhibits serious issues:
  - ignores user needs
  - quadratic growth with system size (data, computation and traffic)
- **On demand** network monitoring:
  - suits **user** needs (timing and modality)
  - **grows** linearly with system load
  - **security** management simplified
  - observation rendered **at once**

Monitoring sessions

- A unique infrastructure regulates the activation of monitoring sessions
- Distinct agents implement distinct functionalities

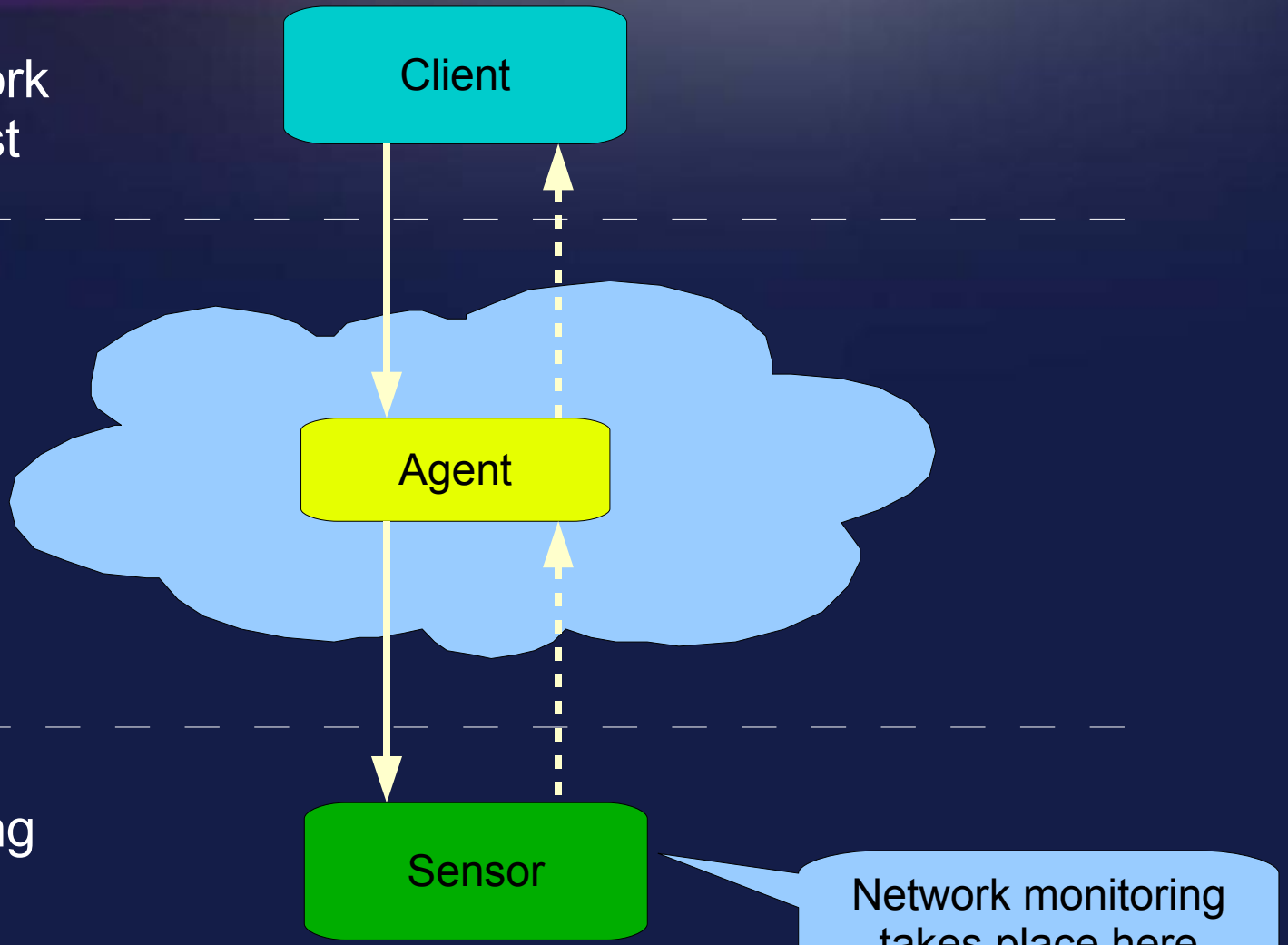


Architecture layout

Produces a network monitoring request

Transport infrastructure

Passive monitoring effector



The Client

- Represents the entity that is interested in network monitoring results (e.g. a Workflow Manager)
- Submits a **request** to an agent specifying, **traffic description** (e.g. endpoints, ports), and the expected results
- Expectation includes traffic description (e.g. endpoints, ports), syntax (e.g. precision), semantics (e.g. tool name), metadata (e.g. max bandwidth)
- Requests are submitted to a “well known” local Agent

The Sensor

- The component which is in charge of effecting the observations: located on a possibly specialized host, uses **passive** monitoring techniques
- Receives monitoring request from the client, delivered by a “well known” local Agent, containing the specs of the traffic to monitor, and expected results
- Configures traffic **filters** and formats response **packets**
- Packets containing observations are delivered **at once**, embedded into a UDP stream

The Agent

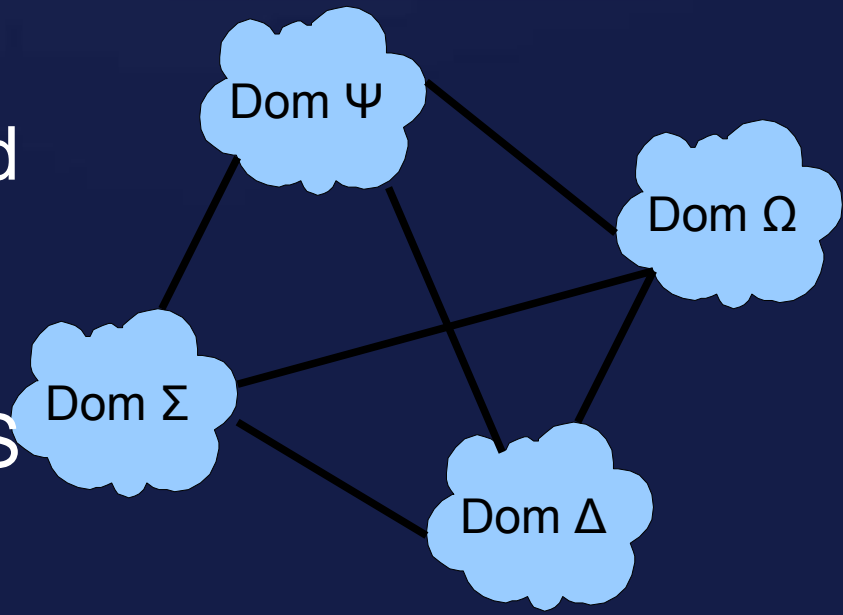
- The basic component of the transport infrastructure
- Two basic functionalities:
  - Routes **requests** from Clients to Sensors
  - Maintains **streams** from Sensors to Clients
- Routes are computed using a **global database** representing the partitioning of the system into **domains**
- **Clients** and **Sensors** are aware of Agents located in the same domain

Security issues

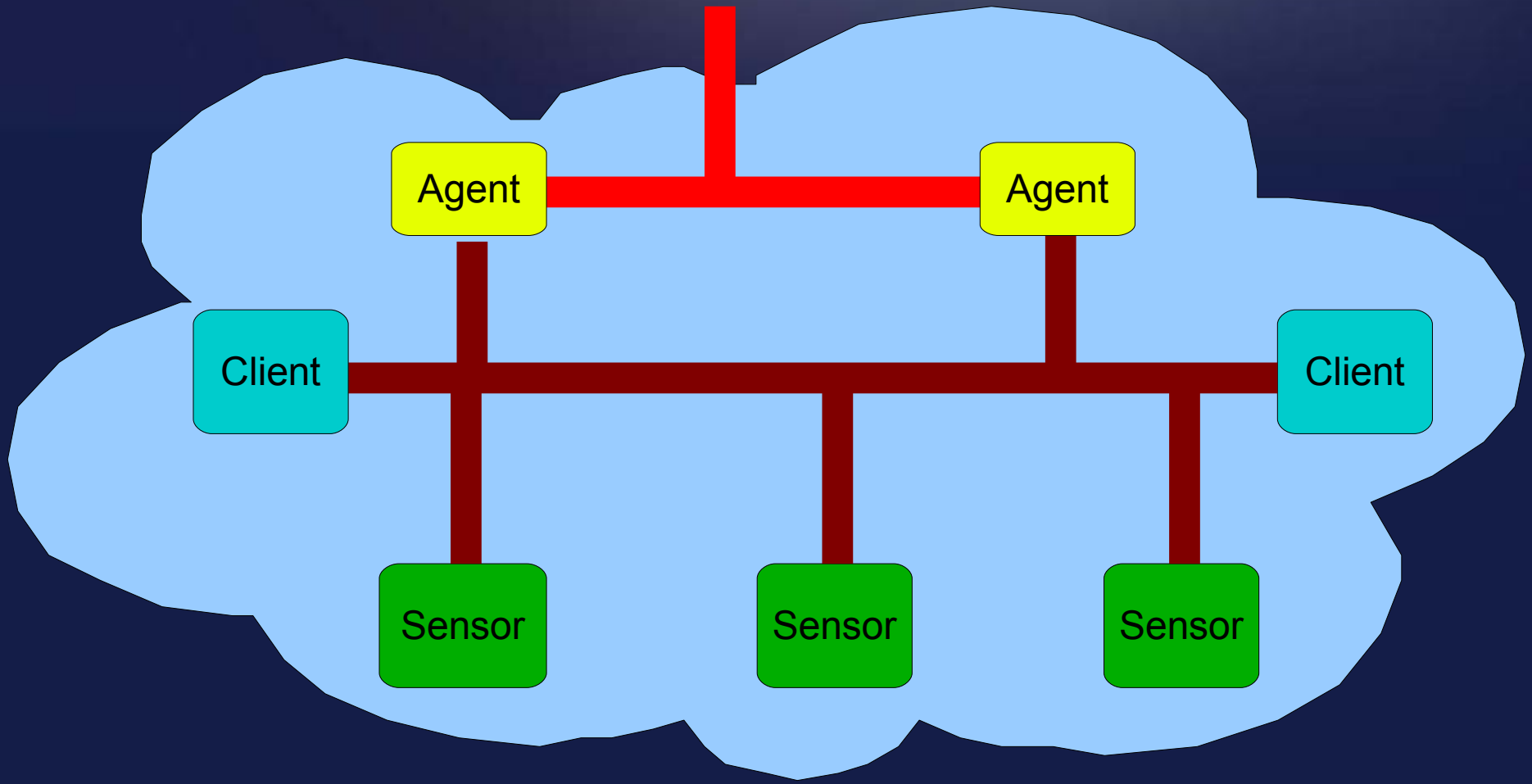
- Traffic between Agents is **authenticated** using public keys certified by a CA, stored in the global database
- Authentication is needed to avoid spoofing, **encryption seems unnecessary** in most cases
- Traffic between Agents and Sensors or Clients is authenticated using public keys managed using criteria local to the Domain
- Response payload may be encrypted according to criteria negotiated between the Client and the Sensor

Domains

- A Domain is a set of Clients and Sensors that operate under control of one or more Agents
- A hierarchical arrangement of domains seems pointless
- Domains composition is recorded in a global database
- The global database can be implemented using LDAP or DNS (a hyperscalable gossip based solution is under study)



Insight of a domain



Communication: Requests

- Agents offer a **SOAP** interface to accept Requests
- A Request is an **XML document** (a demo XSD has been produced) divided into two parts:

```
    <nmsd:NetworkMonitoringSession
xmlns:nmsd="http://www.di.unipi.it/
~augusto/schema/NetworkMonitoringSessionDescription-0.3.xsd"
SessionId="456@wma@zeus" StartAt="2007-09-17T12:00:00.000-01:00"
Duration="2H">
        <NetworkElement SourceDomain="zeus" Destination="ares" />
        <RequestFrom TarskId="WF245"
WorkflowMonitoringAgentId="wma@zeus" />
        <Route>
           <NextAgent Agent="nma@zeus" Index="1"/>
        </Route>
```

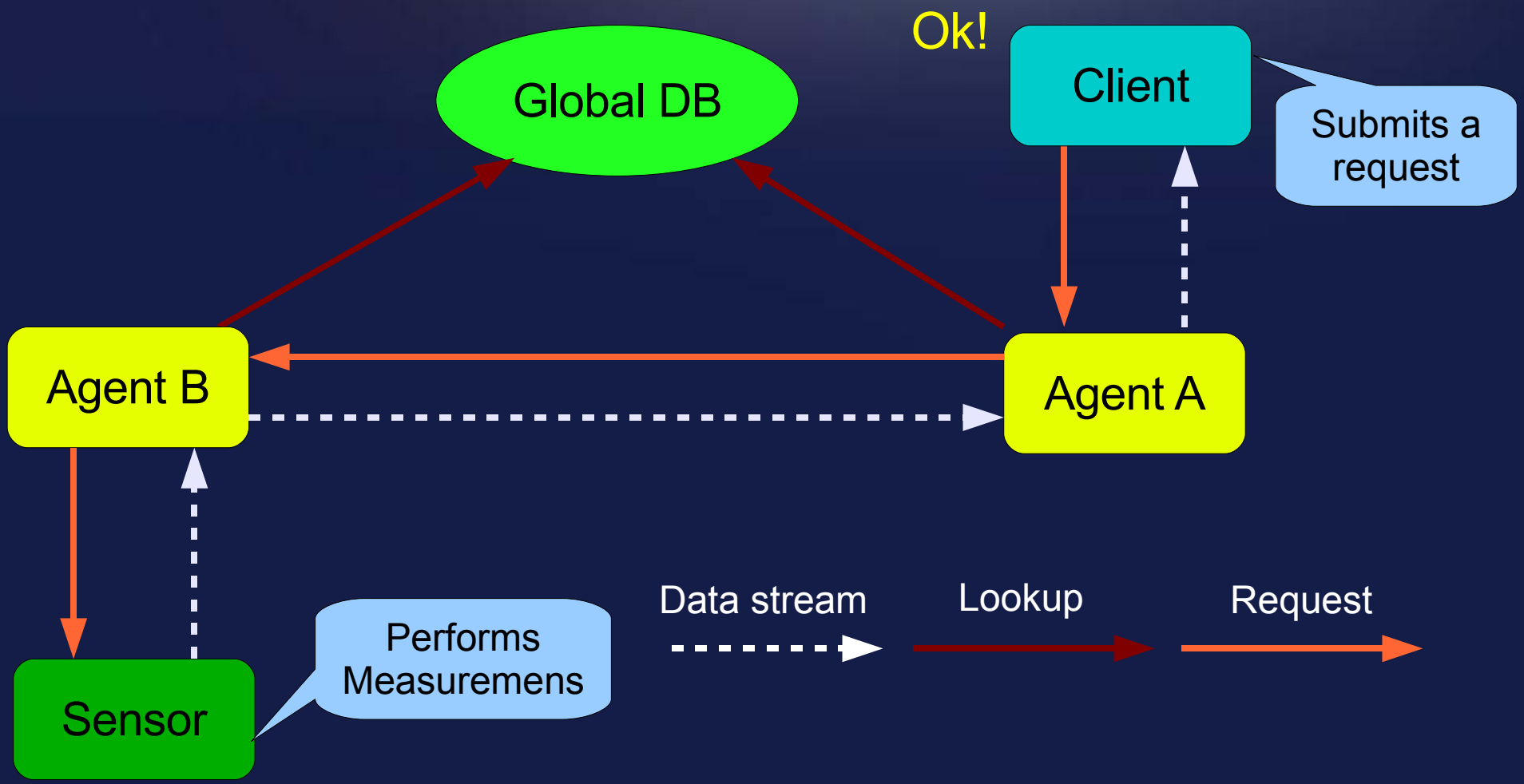
Communication: Data streams

- A stream is implemented as a sequence of **UDP** packets (a single session may generate several streams)
- **Packet loss** events entail detectable unavailability of some observations
- Route is determined using **request route** records, plus some immediate shortcut recipes
- Payload may be **encrypted** if network observations are considered sensitive information
- **Multicast** is in principle possible

The global database

- It implements the **global state** of the infrastructure:
  - **Sensors** capabilities and managing agent
  - **Agents** public key
- We envision a **distributed** implementation:
  - each agent has access to a **local proxy** with part of the whole database
  - requests out of the cache and updates are **propagated**
  - implemented in LDAP (prototype)

Example


Ok!
Global DB
Client
Submits a request
Agent B
Agent A
Data stream
Lookup
Request
Performs Measuremens
Sensor

The prototype: targets

- To assess the feasibility of the whole design (complex, with many components)
- To identify existing techniques to implement the various functionalities
- To layout exhaustively the design of the proposal
- To assess the integration between the work of the two partners

NON TARGETS: demonstration of scalability, performance, user satisfaction.

The prototype: methodology

- Select well known, well supported technologies to concentrate on real issues
- Cleanly modularize the design to obtain a usable layout
- Make the prototype fully portable to support cooperation between partners

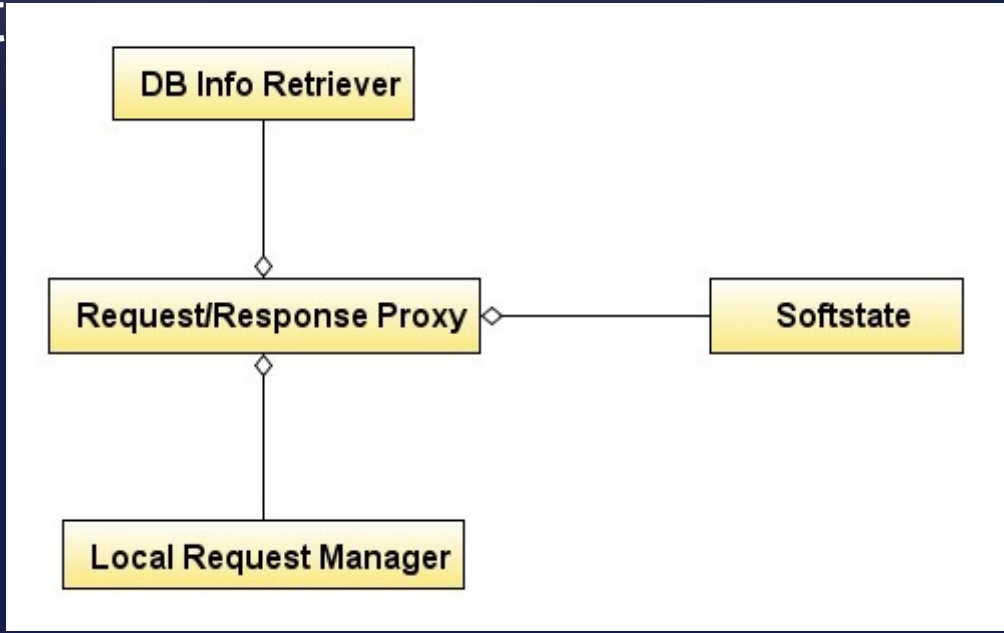
OUR OPTIONS:

Java for the language, SOAP and UDP for communication, LDAP for the database and virtualize the testbed (NETKIT)

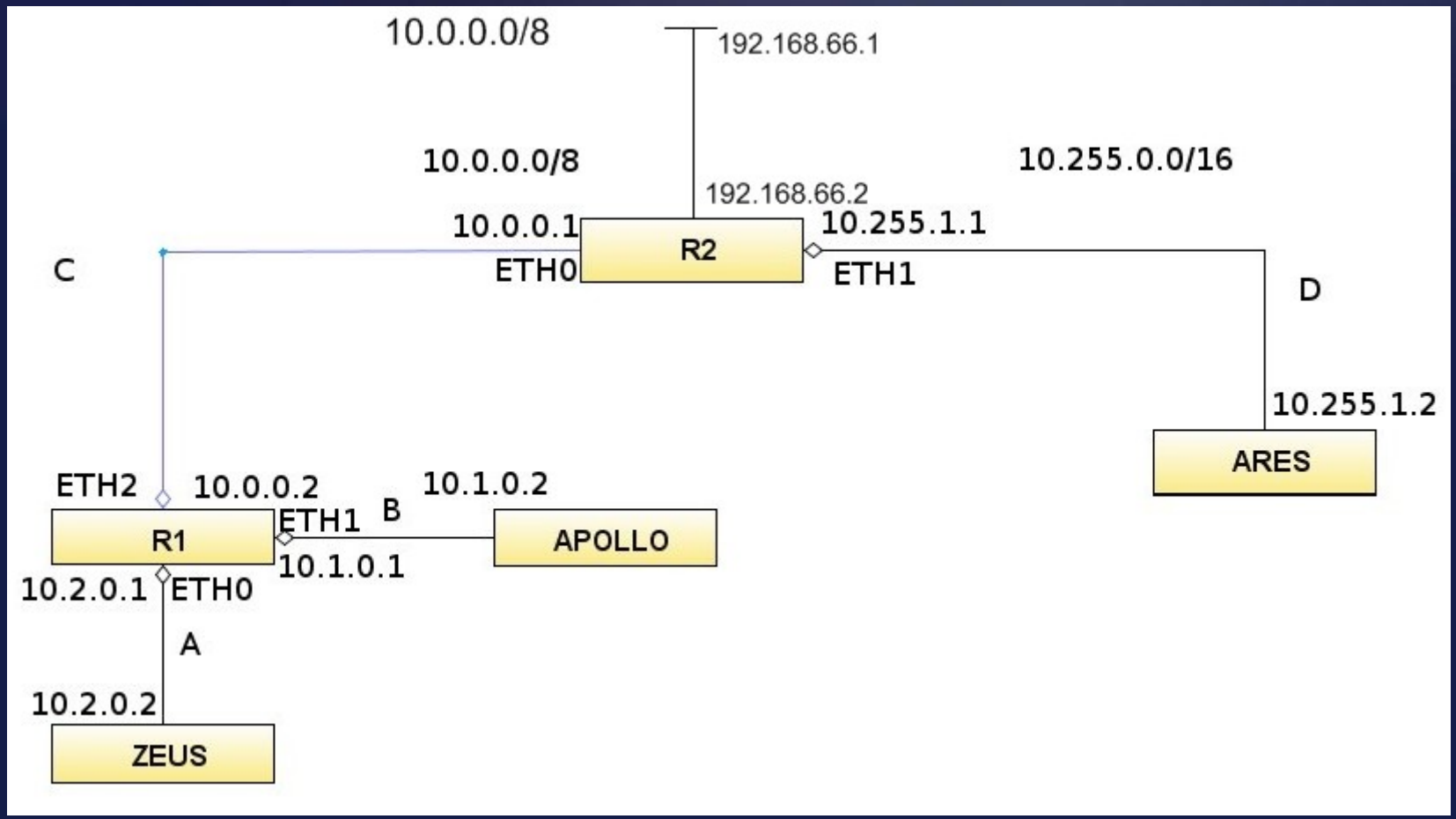
The prototype: a sample

Agent (requests):

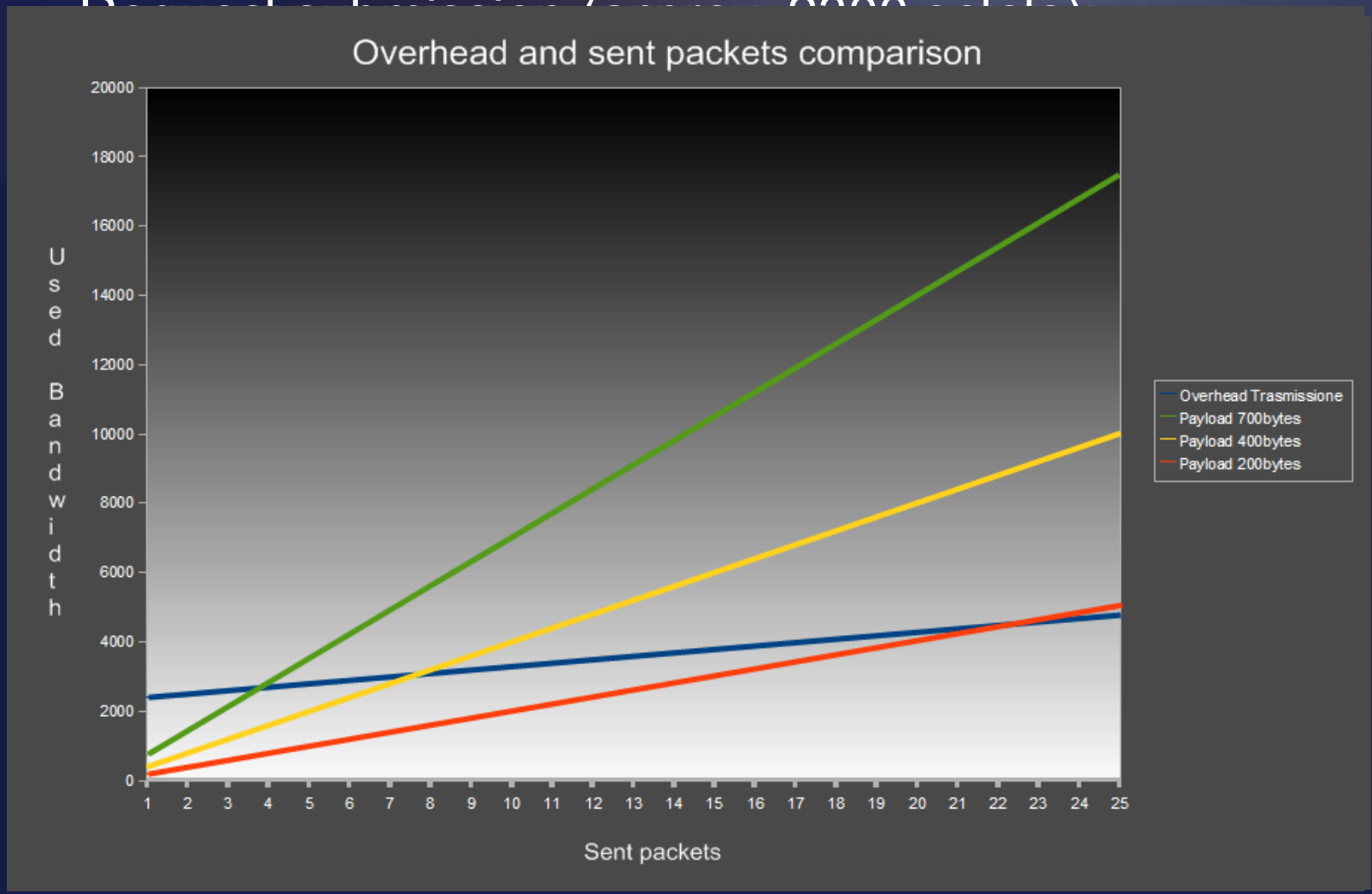
- DB info retriever: transparent API to query the distributed database
- Request/Response Proxy: effects routing
- Local request Manager: interacts with Sensors
- Softstate: records requests and controls routing



The prototype: the testbed



Profiling protocol overhead



Conclusions

- The infrastructure we have designed and prototyped is a step towards a network monitoring infrastructure for a Grid environment in the Internet scale
- To achieve the target, we opted for an **on demand** approach, specially suited for a Grid environment
- There is no experience with **on demand** monitoring, although the technology is ready
- The implementation of the prototype was carried out as a **Master thesis** by Yari Marchetti, who later joined the research group, in three months