*IEEE/ACM GRID2005 workshop*
*Seattle - November 2005*

Grid'5000

# HIPernet

## a Decentralized Security Infrastructure for large scale Grid environments


**Julien Laganier**

**Pascale Vicat-Blanc Primet**

INRIA – Reso

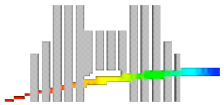
LIP Laboratory

Ecole Normale Supérieure de Lyon

France

Pascale.primet@inria.fr


CNRS CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE



INRIA

# Outline

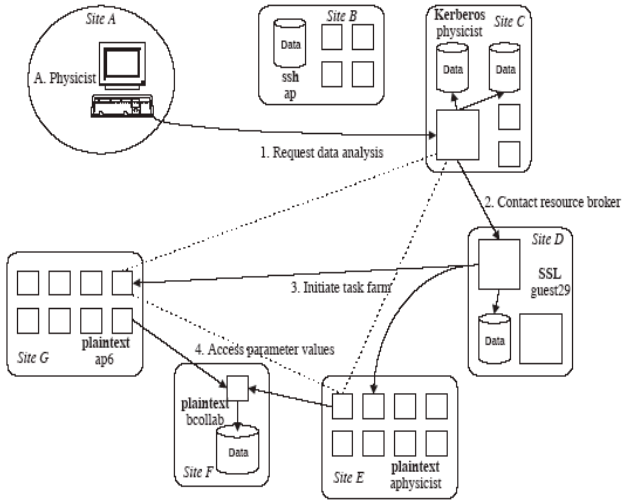
- Security in Grids
- Key concepts for decentralized security
- HIPernet design and implementation
- Conclusion and perspectives

# Security in Grids

Grid and global computing is about sharing a **large number** of distributed resources amongst a **multidomain wide area** context.

Key issues [Foster 98]:

- Access to shared resources
- Secure Communication Channels



# Grid security: constraints

- Relationships between grid entities (e.g. users, services, resources, virtual organizations, etc.) are **dynamic, and possibly short-lived.**
- The **network interconnect is dynamic** :
  - might grow, diminish, move, etc. It is not a fixed entity w.r.t. location and constitution.
- The entities would like to use the communication infrastructure **transparently**:
  - end-use, applications, tools and APIs behave like if they were using a regular TCP/IP Internet or intranet.

# Key issues:

- **Access to shared resources**

->Make an **authorization decision** when a shared resource is accessed:

  - who can do what (and how)

- **Secure Communication Channels**

->**protection of data and flows** during exchanges:

  - Data : integrity, authentication, confidentiality…
  - Flow : authentication, confidentiality, …

->**protection of resources** during Grid life:

  - Channel and grid : isolation, Deny of Service prevention, intrusion prevention…

# Authorization: who can do what (and how)

- Authentication-centric Approach:
  - Authorization decision process relies on authentication,
  - Requires Access Control (ex. ACL in firewall) : defining who (e.g. John Smith) is allowed to do what (e.g. send through the firewall TCP packets with destination port number 80.
- Authorization-centric Approach:
  - Do not rely on identification to make an authorization decision

- in Grids : Authentication-centric approach:
  - Requires a Public Key Infrastructure,
  - Requires the entities of the grid to trust all the Certificate Authorities (CA) of the security domains composing it.
  - Do not scale in number of domains & present security issues

# PK Technology and PKI (Carlisle Adam's 10Y survey 2004)

- **Public key technology**
  - Each entity in a collection has a **pair of keys**
    - Alice has $pub_A$, $priv_A$
    - Encryption, digital-signature. possible (mathematical operations)
- Public Key Infrastructure (PKI)
  - Makes PK technology available to applications and environments that wish to use it
    - Enc, d-sig. possible (security operations)
  - **Key pair bound to an entity identifier** in a way that makes it useful to a variety of apps

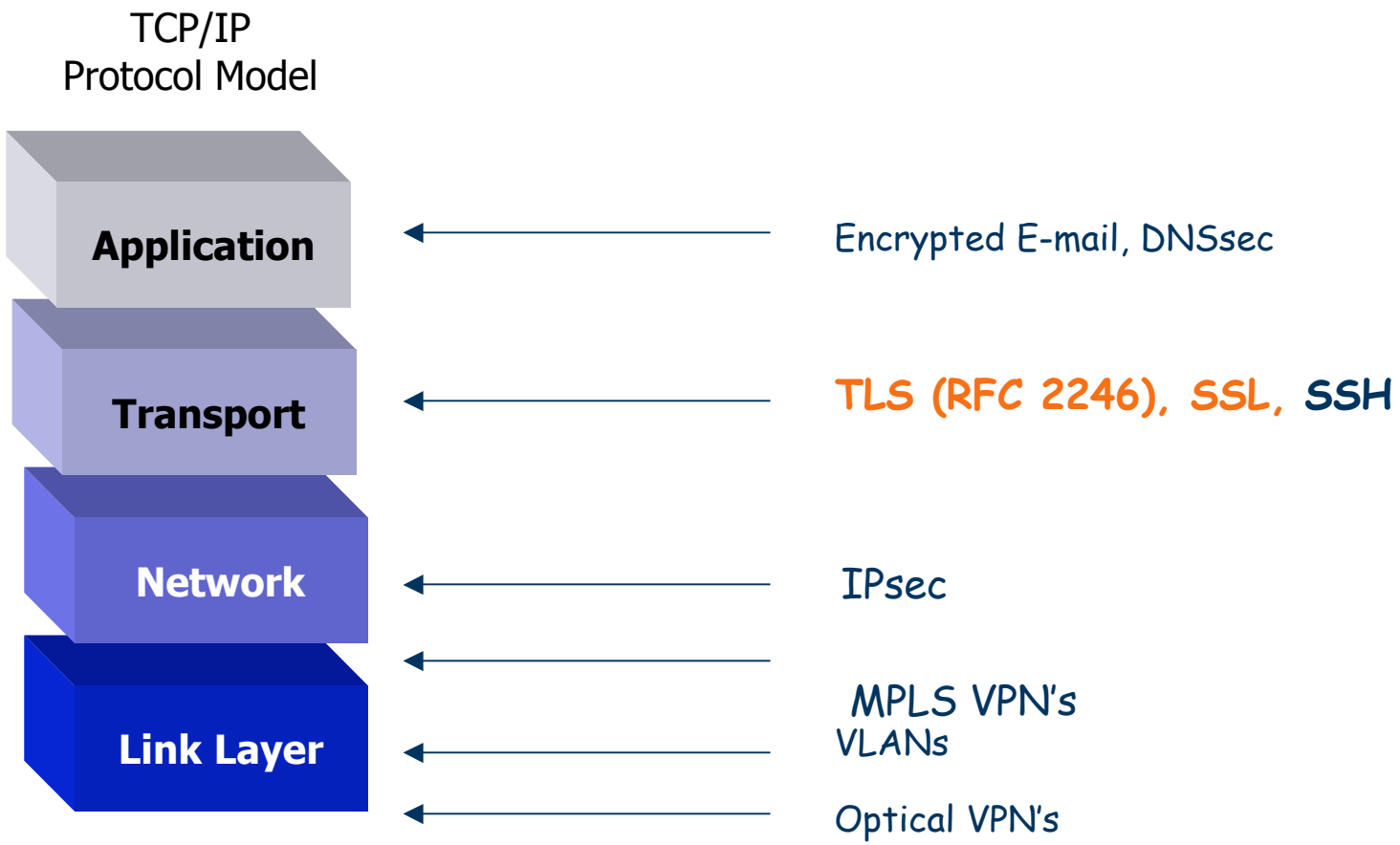
# PKI (cont'd)

- **Binding of key pair and identifier**
  - Validity of bindings
    - Authority (making & breaking)
    - Issuance process (syntax & dissemination)
    - Termination process (alerting)
  - Use of bindings
    - Anchor management process (augment & diminish)
    - Private key management process ("fit for purpose")
    - Binding validation process (trusting someone else's key)

# Binding of key pair and identifier

| PKI Solution | Authority | Issuance Process |
|---|---|---|
| X.509 | CA, AA. CA is owner / definer of namespace. | ASN.1 syntax. X.500 or LDAP directories. |
| PGP | No external authority. User is owner / definer of namespace. | BNF syn. Issued by key owner (e.g., Web page, e-mail sig., key server). |
| SPKI | Authorization granter. Relying party is owner / definer of namespace. | S-expression syntax. Issued based on SDSI names or pseudon. Ids. |

# Secure communication channels: virtual private networks



# Secure communication channels

| VPN Solutions | Protection | Efficiency | Scalability | Multipoint |
|---|---|---|---|---|
| Application layer | Data : authen, confidentiality | No | yes | No |
| Transport layer (TLS, SSL) SSH | Data, Flow: authen (confidentiality) | **Yes** if no confidentiality | Yes | Yes |
| Network layer (Ipsec) | Data, Flow, Channel, Grid: integrity, confid, authen, no replay | May be an Issue | Yes if end based | Yes if end-based |
| Lower layers (G-MPLS) | Data, Flow, Channel, Grid | Yes | No | Hard |

# Outline

- Security in Grids: requirements and issues
- **Key concepts for distributed security in Grids**
- HIPernet design and implementation
- Conclusion and perspectives

# Key concepts and tools

- While the size of multi-domain grid environment is increasing, do not reduce its **robustness, flexibility and overall security**.
  - -> No Centralized Authority (third party) for authorization
  - -> Multipoint secure dynamic & efficient communication channels
  - -> Transparence to application

- Tools : (based on PK technology)
  - **Merging identity and public key** (CBIDs)
  - **Autorization Certificate**, Certificate **Delegation** (SPKI)
  - **Host Identity protocol (HIP)**
  - **Virtualization**

# Cryptobased ID's (CBID's) [Montenegro 04]

Grid'5000

IPv6 (unicast) address format:

- prefix (64 bits) + HostID (64 bits) = 128 bits
- ***IPv6 CBAddress: HostID = hmac-64(imprint, sha1(PK))***
  - PK : public key
  - imprint: a 64-bit field (0's if not specified)
- ***CBI = hmac-128(imprint, sha1(PK))***

- Since CBIDs are statistically unique, a host only has to prove that it knows the private key (by signing) to prove that it owns the address or identifier…and that it is not using someone else's CBID…
- NO infrastructure required: crypto relation between address and signature…

# CBID's = Merging identity and public key

·**Strong cryptographic binding between PK and ID.**

Advantages :

- Statistically Unique and Cryptographically Verifiable (SUCV).
- Fully distributed, no infrastructure required.
- **Permit to avoid a CA** by binding the hosts' addresses or identifiers to their public key

·Vision for the grid:

- Everything has a CBID (Hosts, Users, Applications, Services..)
- Entities delegates rights to others entities
- CBID's delegates rights to others CBID's
- Uses Authorization Certificate (e.g., SPKI)

# CBID's Applications [ Castellucia 04, Bassi 03]

- Used to help solve several issues in the IPv6 world:
  - **identifier ownership** for mobility,
  - neighbor discovery,
  - multicast group membership
  - infrastructure-less opportunistic encryption
  - CBID's are also proposed for JXTA (http://crypto-id.jxta.org)

# SPKI certificate [RFC 2693 - Ellison 99]

A SPKI certificate has the following general structure:

*(sequence*
*(public-key object)*
*(cert object)*
*(signature object)*
*)*
*public-key and signature defined by the SPKI framework,*
*cert object is application dependent.*

# SPKI certificates

- Can be seen as an **ACL entry packed with an in-line PKI certificate.**
- They allow an entity to prove to another entity that he has **the right to perform some actions** on it, without the intervention of any trusted third party.
- SPKI certificates are very useful objects in a distributed and open system threatened by possible malicious attacks.

# Certificate chain

- If "entity X delegates to entity Y the right R" is denoted by the notation X $-_R\rightarrow$Y , then

> *A $-_S\rightarrow$ B and B $-_U\rightarrow$ C implies that A $-_{S*U}\rightarrow$ C , i.e.,*
>
> *entity A delegates to entity C the **intersection of rights** S and T.*

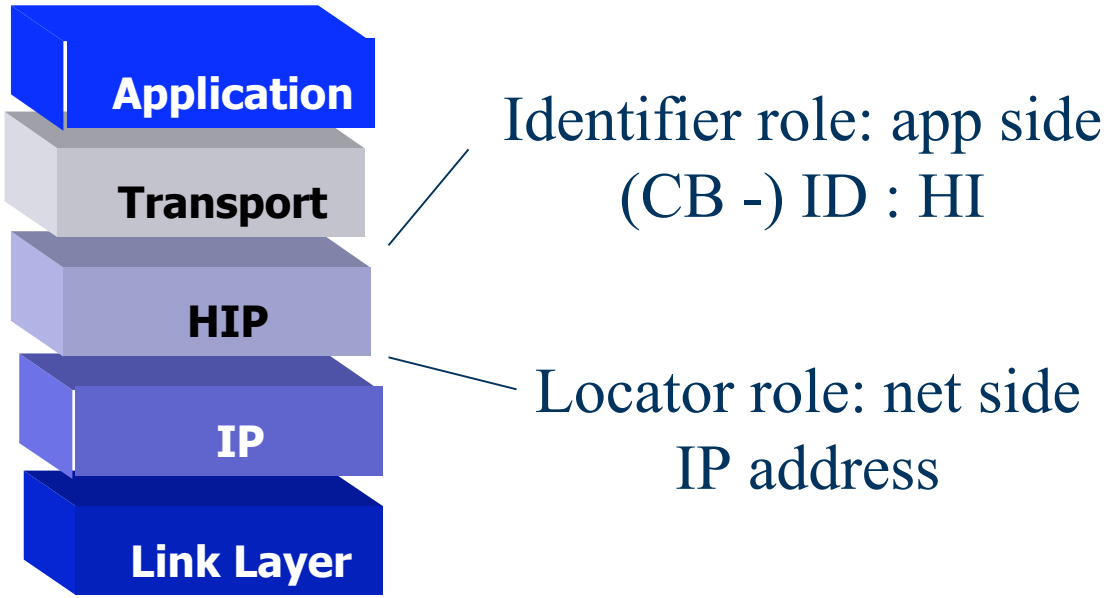
- For instance,

> *if A - $_{rwxr-xr-x}\rightarrow$ B and B - $_{r-xr-x---}\rightarrow$ C, then A- $_{r-xr-x---}\rightarrow$ C*

Delegation has lot of interesting properties

# Host Identity Protocol (HIP) [Moskowitz 05]

- a new protocol (IETF) that **identifies hosts with Public Keys.**
- HIP decouples the locator role from the identifier role of IP address as used in the TCP/IP stack,



# The Host Identity Protocol


- HIP is for mutual peer authentication:
  - based on a **Sigma-compliant Diffie-Hellman key exchange**,
  - using public-key identifiers from a
  - new **Host Identity name space**

- The protocol is designed to be resistant to Denial-of-Service (DoS) and Man-in-the-middle (MitM) attacks, and when used together with another suitable security protocol, such as Encapsulated Security Payload (ESP), it provides integrity protection and optional encryption for upper layer protocols, suchs as TCP and UDP.

# The Host Identity Protocol


- Applications do not use directly IP addresses as end-point identifiers but
  - the Host Identity (HI) (public component of public-private key pair, or
  - the Host Identity Tag (HIT), a fixed size (i.e. compressed) representation of the HI obtained by truncating the output of a secure hash (e.g. SHA1) applied to the public key

The HIP layer is in charge of mapping HIs and HITs into appropriate locator IP address for the node

# Virtualisation and multipoint

- Secure group communication (multipoint VPN) is an important feature in grids
- Management and deployment of VPN overlays between very dynamic coalition of nodes (like grids) can be better tackled by the communicating endpoints themselves (as opposed to trusted third parties)

- The Host Identity Protocol is the key-enabler of end-to-end multipoint VPN overlays.

# Outline

- Security in Grids
- Key concepts for distributed security in Grids
- **HIPernet design and implementation**
- Conclusion and perspectives

# Security principles [J.H. Saltzer]

- **Economy of mechanism**: Keep a design as simple and small as possible.
- **Fail-safe defaults**: Base access decisions on permission rather than exclusion (i.e. default situation is lack of access.)
- **Separation of privilege**: Where feasible, a protection mechanism that requires two keys to unlock it is more robust and flexible than one that allows access to the presenter of only a single key.
- **Least privilege**: Every program and every user of the system should operate using the least set of privileges necessary to complete the job.
- **Least common mechanism**: Minimize the amount of mechanism common to more than one user and depended on by all users.
- **Psychological acceptability**: It is essential that the human interface be designed for ease of use

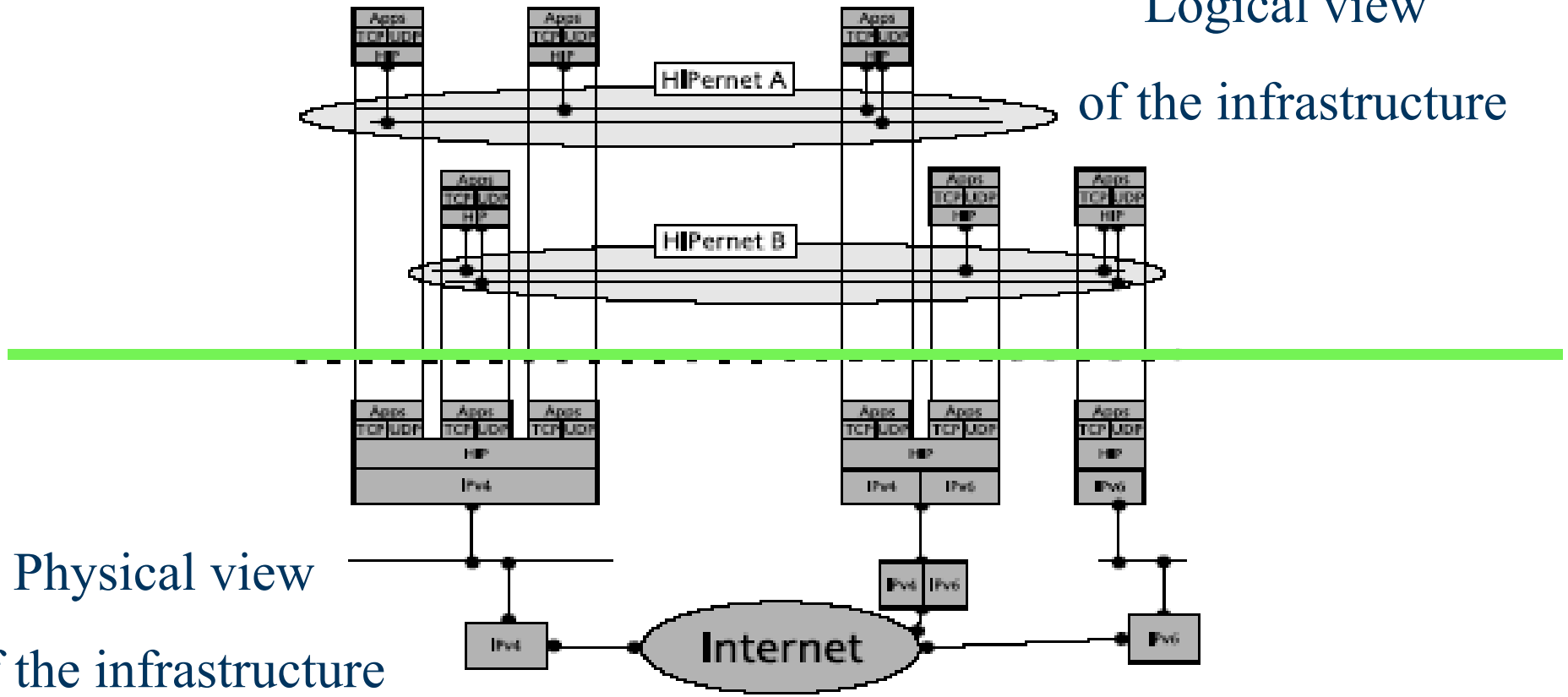
# HIPernet design goal

- Give the user the illusion he is using its own system, while in reality it is using multiple systems part of the Grid.
- **Combine the virtualization of both the network and the operating system**:
  - the OS is virtualized to permit multiple virtual nodes to cohabit on the same physical host,
  - the network is virtualized to permit multiple virtual overlay networks to cohabit on a shared communication infrastructure.
  - Networks overlays and nodes are kept isolated both at the network (IPsec) and OS (filesystems, IPCs, etc.) level.
- **Use existing security mechanisms: CBID, HIP, SPKI**

# The HIPernet model

Logical view

of the infrastructure




Physical view

of the infrastructure

# HIPernet: terminology

- **HIPernet**: a set of HIPernet Nodes & Hipernet channels under a single administrative domain.

- **HIPernet Entity:**
  - a participant in a security operation.
  - can be a user, a group of users, an application, a service, an organization, a computer node, etc.
  - is uniquely identified by a public key (HI) or its compact representation (HIT),
  - can delegate, or be delegated, specific access rights via SPKI authorization certificates.

# HIPernet: terminology

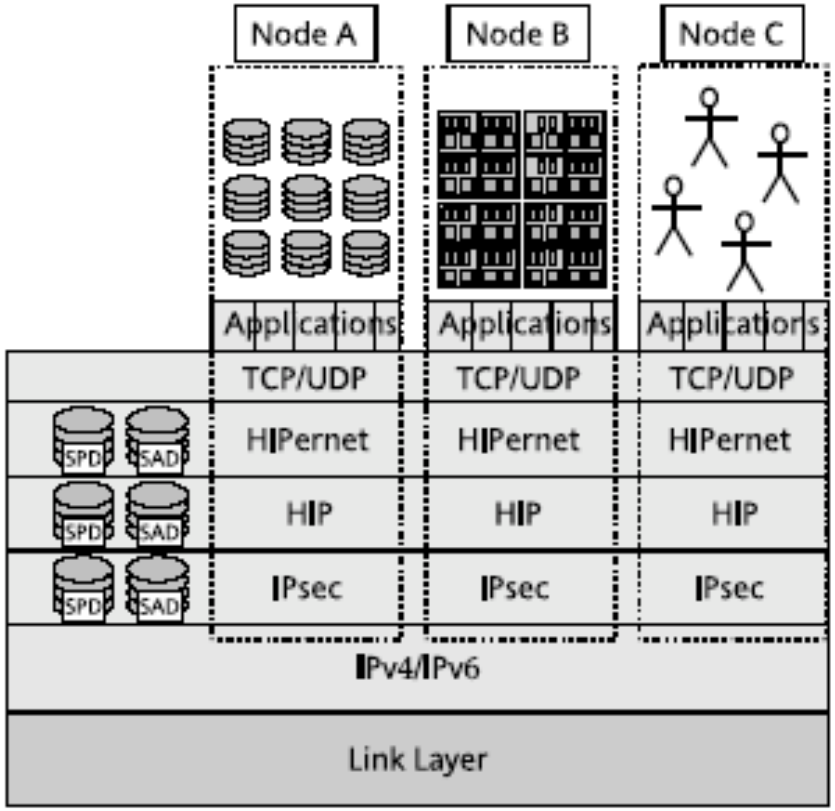
- **HIPernet Node** :
  - smallest security container for a communicating execution environment.
  - typically a zone (or jail) instance running into a HIP-enabled UNIX node, and attached to one or more HIPernet Channels. It is defined by the following parameters:
  - **HIT** of HIPernet Node
  - **Lists of HIPernet Channels** to which the HIPernet Node is attached.
- **HIPernet Channel**: A trust domain defining a set of HIPernet nodes which can communicate together. This is the smallest security container to isolate security perimeters in a HIPernet. It is defined by the following parameters:
  - **HIT** of HIPernet Channel
  - Parameters to contact the **HIPernet Resolution Service** for this channel.

# HIPernet infrastructure

- The communication primitives needs additionnal helps to function correctly; in particular,it is necessary to:
  - Authorize/Deny an entity the right to participate in a HIPernet channel.
  - Insert/Update/Remove an HIPernet node "identifier to locator" mapping.
  - Resolve an HIPernet node identifier into a locator.

- **HIPernet Registration Service**: A service allowing HIPernet Nodes to join and leave a HIPernet Channel.
- A single Resolution Service might be distributed over multiple HIPernet Nodes, and shared by multiple HIPernet Channels.
- It is implemented by either hierarchical (e.g. DNS) or flat distributed (e.g. DHT) lookup service.

# HIPernet node



# HIPernet advantages

- Members of a HIPernet have a consistent view of a single private TCP/IP VLAN overlay, which is independent from the underlying network topology.
- The security primitives afforded by a HIPernet are data origin integrity, anti-replay protection and confidentiality.
- The entities which needs an execution environment are provided with their own HIPernet node instance, which communicates through the HIP layer.
  - Isolation
  - Entity security policy database.

- **Transparence to application** : end to end properties preserved

# Conclusion and Future Works

- HIPernet is proposed as a **solution to secure a distributed computing platform infrastructure and their services**.
- Combine together existing security building blocks
  - CBID, Ipsec, HIP, operating system sandboxes and authorization certificates, SPKI
- Provides **location-independent security**
- Elegant architecture based on **IPng protocols**

- Implemented for UNIX-like operating systems (FreeBSD).
- Will be implemented in Linux, deployed and evaluated at large scale in GRID5000 testbed


# Questions?

# HIPernet authorization certificate

The HIT defines a trust domain, and can issue a certificate:

*(cert*
*(issuer (hit <hit>) )*
*(subject (hit <hit>) )*
*(tag <capability-name_1> (arg <arg_1>)*
*...*
*(arg <arg_i>) )*
*(tag <capability-name_2> (arg <arg_1>)*
*...*
*(arg <arg_j>) )*
*(propagate)*
*(online <online-type> <uris>)*
*(not-before <date1>)*
*(not-after <date2>)*
*)*

# Ex HIPernet authorization certificate

## for HIPernet resolution server and client

Grid'5000

:

```
(cert
(issuer (cbid <43fe:b89a:10c0:0fec:
a120:2c48:54ff:de93>)
(subject (cbid <2c48:54ff:1ae3:01bb: 0ab4:b89a:4f0e:389a>)
(tag <hipernet.resolver.client>
<user="*@example.com">
<ttl="1 hour">)
)
(tag <hipernet.resolver.server>
<user="*@example.com">
<ttl="10 days">)
)
(propagate)
(online-certificate-status-protocol
<43fe:b89a:10c0:a120:2c48:54ff:fec0:de93://crl/latest.html>)
(not-before <1/1/2004>)
(not-after <12/31/2009>)
)
```