Policy Framework Proposal

Ciaschini Vincenzo, Ferraro Andrea,
Rubini Gianluca, Zappi Riccardo
INFN – CNAF

4/3/04

Problem Description

- Need to deploy VO-wide policies.
- Need to respect local site policies.
- Need to specify policies relating to the behavior of the grid as a whole.

Current status

- Policies are decided purely on a local site basis (LCAS,grid-mapfile, GACL).
  - they are only ACLs.
- There are no VO policies.
  - VO themselves are just list of users with some attributes attached.

Previous Art

- CAS
  - Allows specification of just everything, but:
  - Completely removes control from site admins.
  - Requires VO to know everything about the layout and internals of farms.
- LCAS
  - Only a static ACL.
  - Deployed on local sites only.

Policy examples

- Users belonging to group /vo/a may only submit 10 jobs a day.
- Users belonging to group /vo/b should have their jobs submitted on the max priority queue.
- User “some user” is banned from the CNAF site.

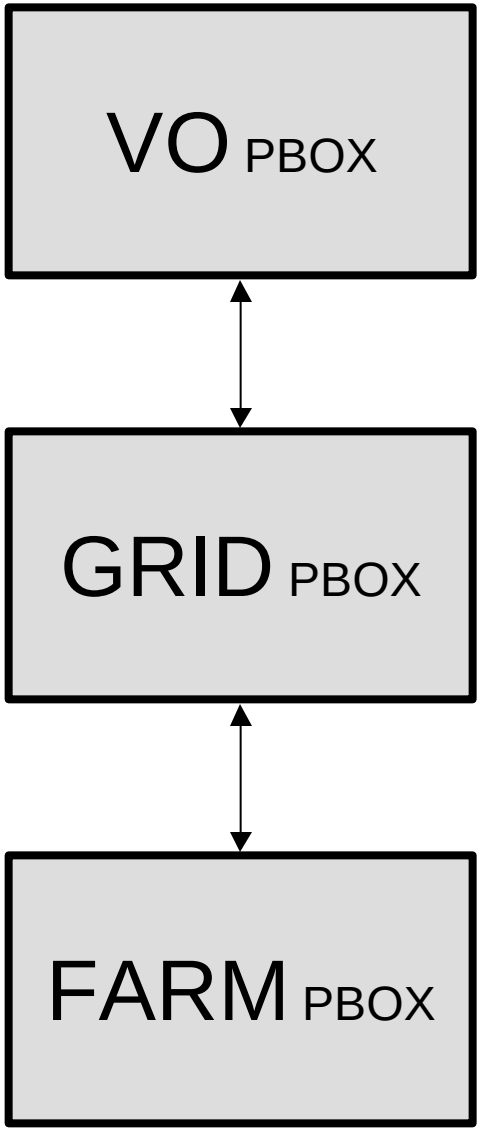
Requirements

- The system should:
  - Be VO-based and distributed.
  - Be highly configurable and able to define and enforce previously unknown types of policies.
  - Leave total control on local sites to local admins.
  - Be capable of express policies requiring a global view of the grid.
  - Be compliant to existing protocols and not require their redesign.

Our Proposal: PBOX

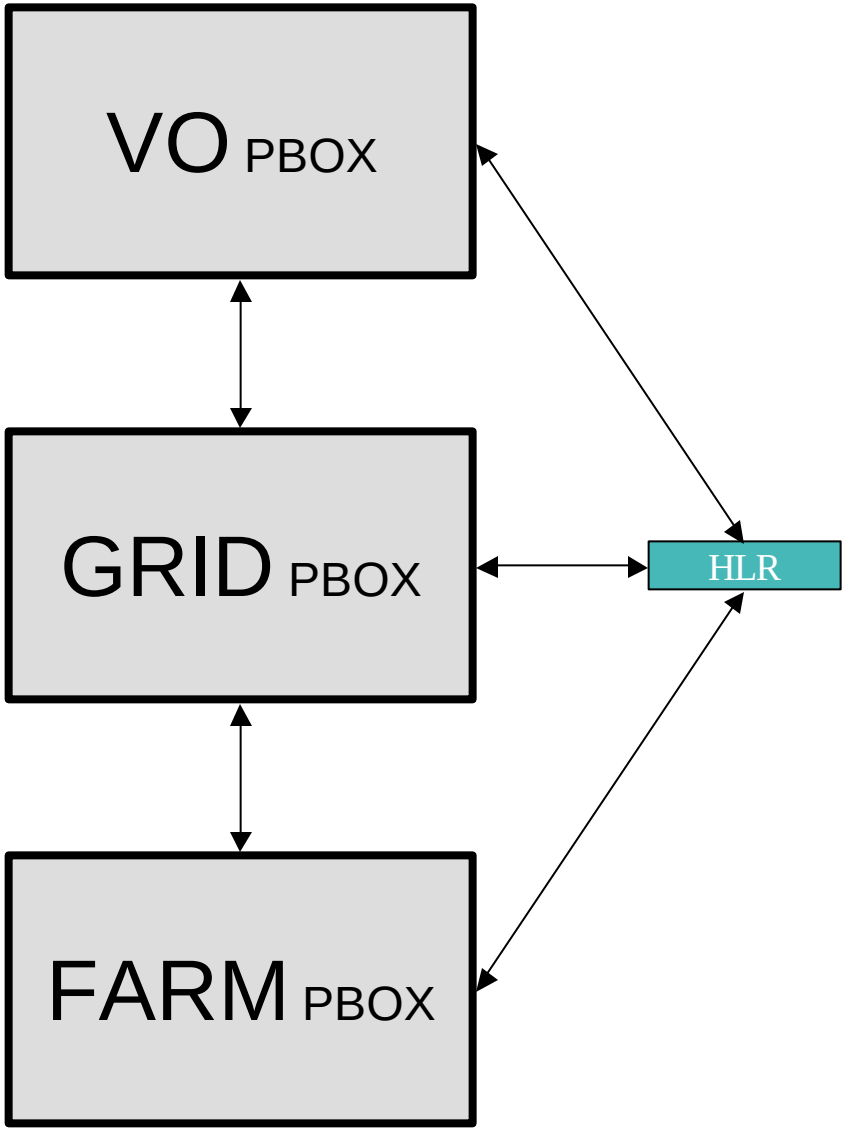
- An independent sets of modules that can be “plugged in” in the current architecture.
- Standards Compliant (RBAC, XACML, GSI)
- Distributed architecture.
- Leveled list of PBOXes (VO PBOX, Grid PBOX, Farm PBOX, possibly subFarm PBOX, etc…)

PBox leveled organization



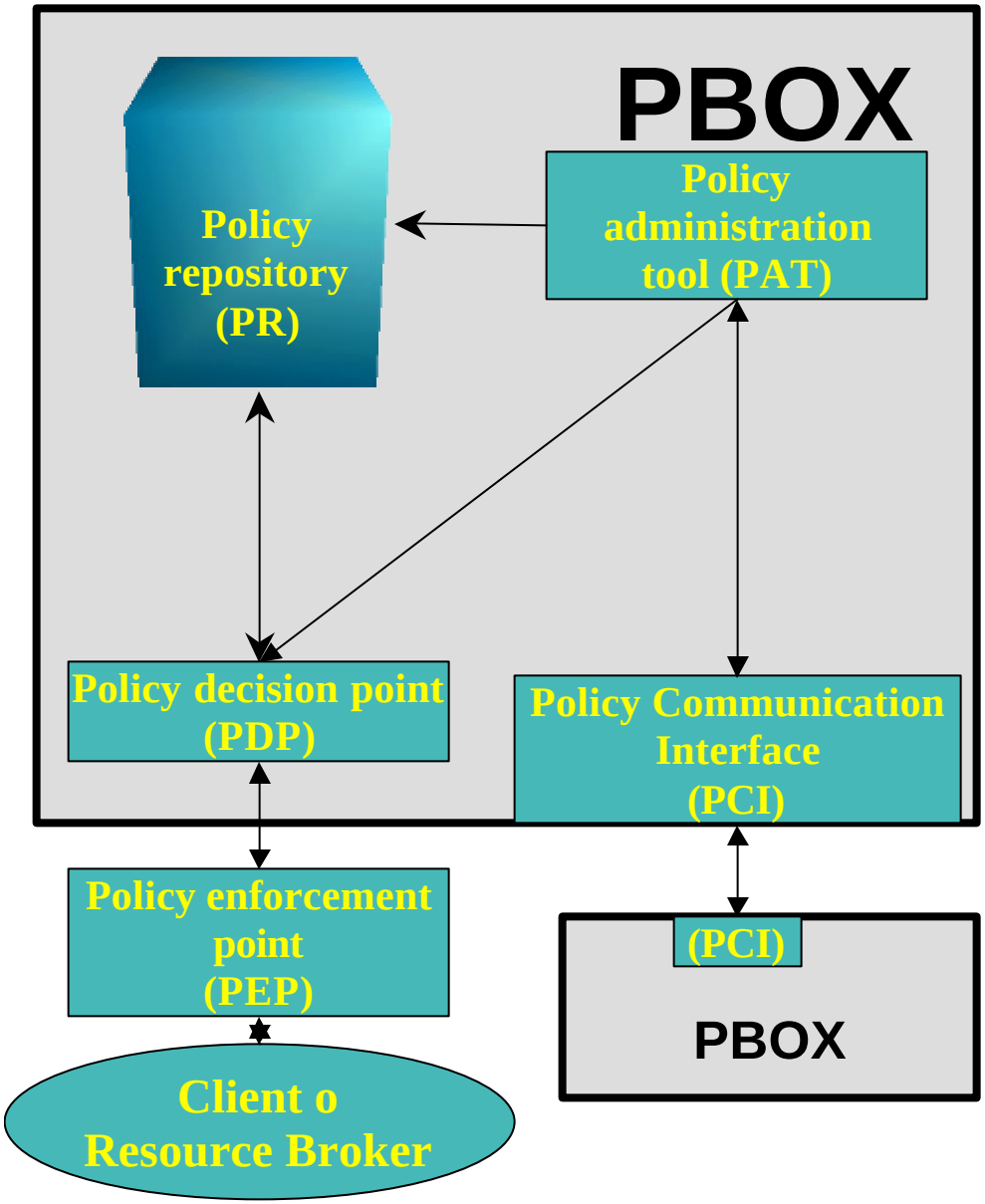
- PBoxes distribute policies between themselves.
- Grid PBoxes are, for example, Grid.it or LCG, or EGEE PBoxes.

PBox leveled organization



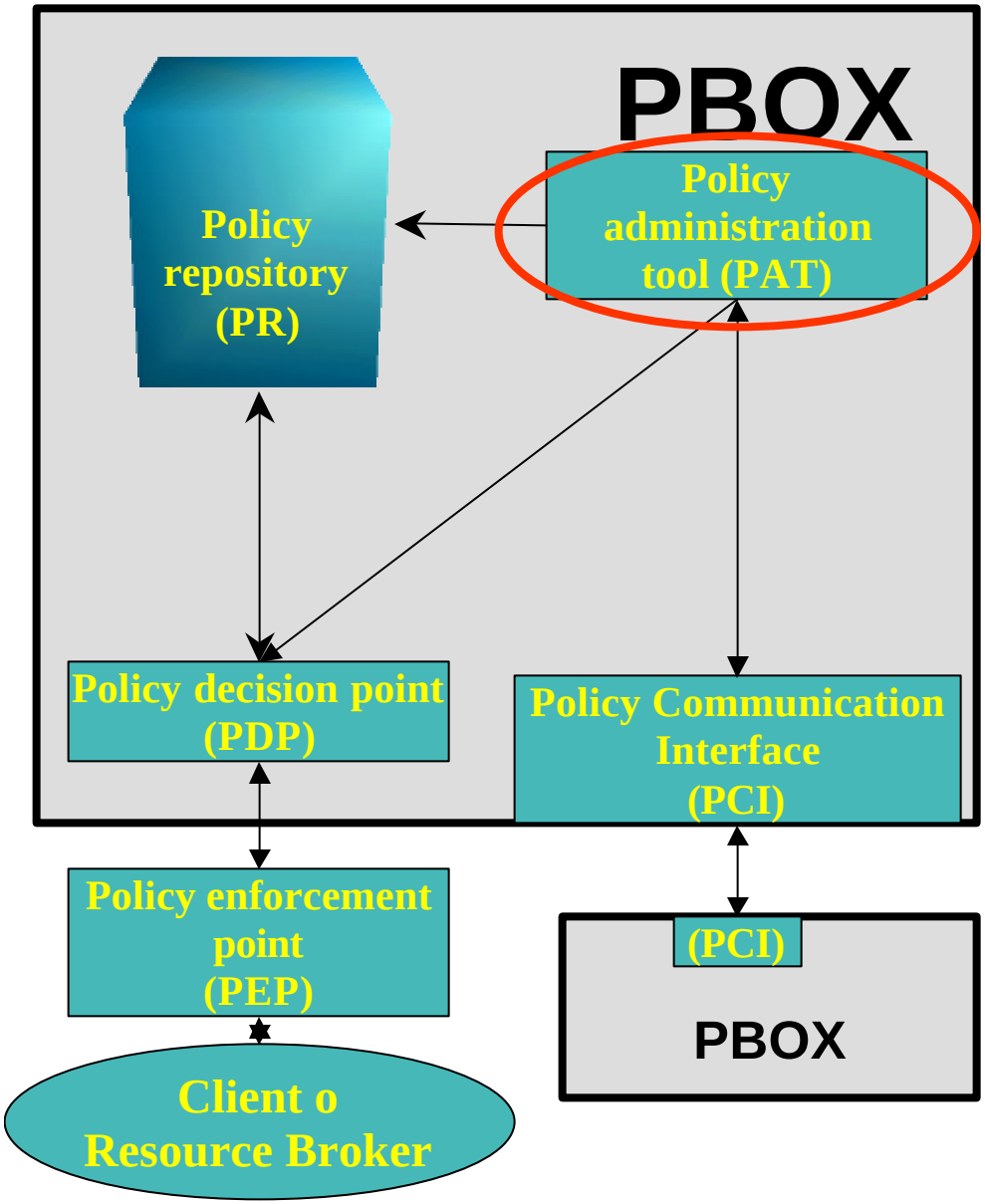
- PBoxes distribute policies between themselves.
- Grid PBoxes are, for example, Grid.it or LCG, or EGEE PBoxes.
- An HLR, part of the accounting system, is necessary for accounting policies.

PBox Structure



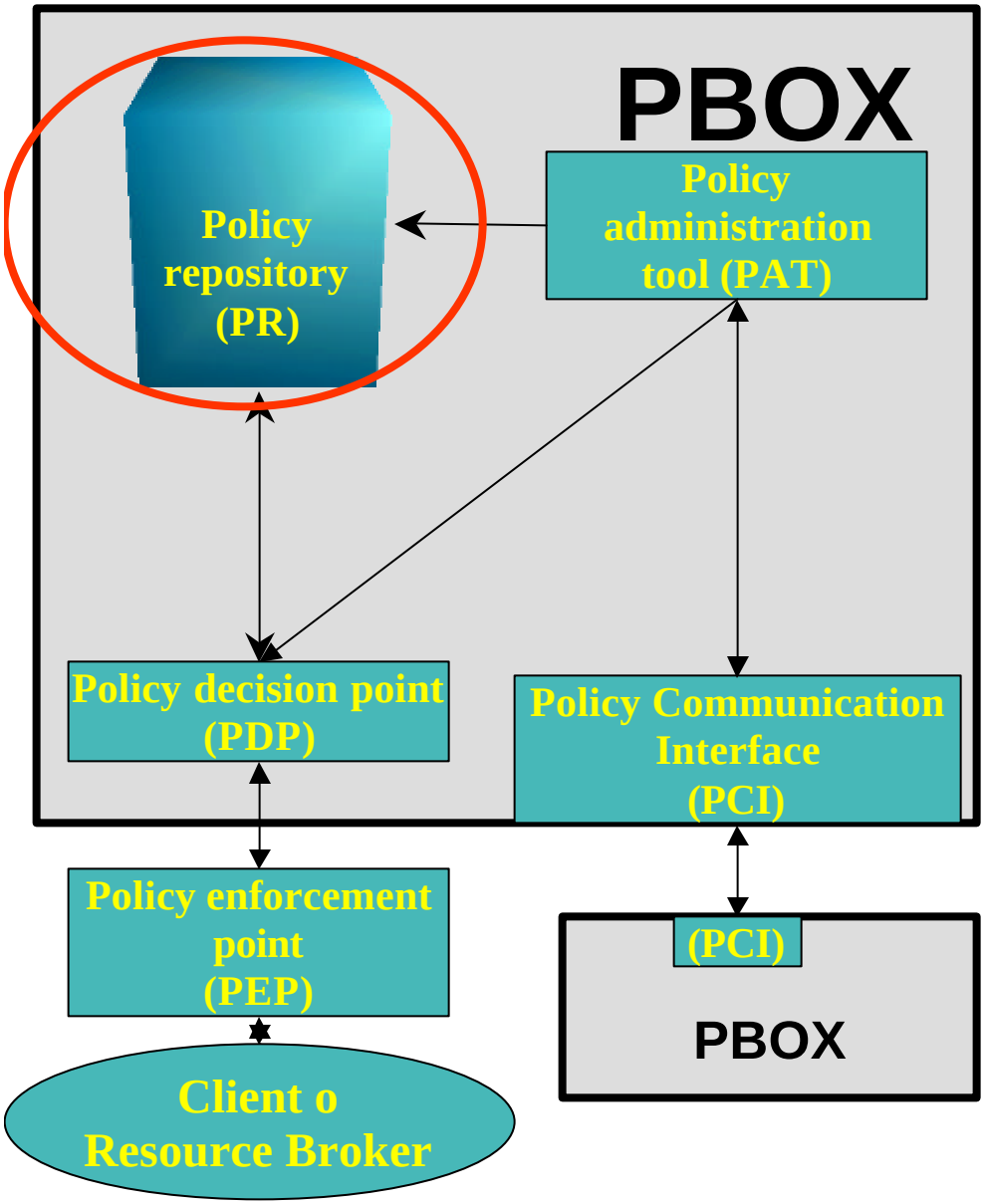
- PAT: An administrative tool to manage policies.
- PR: A database containing current policies and an history of older ones.
- PDP: A module making and communicating decisions regarding policies.
- PCI: A communication interface between 2 PBoxes
- PEP: A client-side module contacting PDP and receiving a response.

PBox Structure: PAT



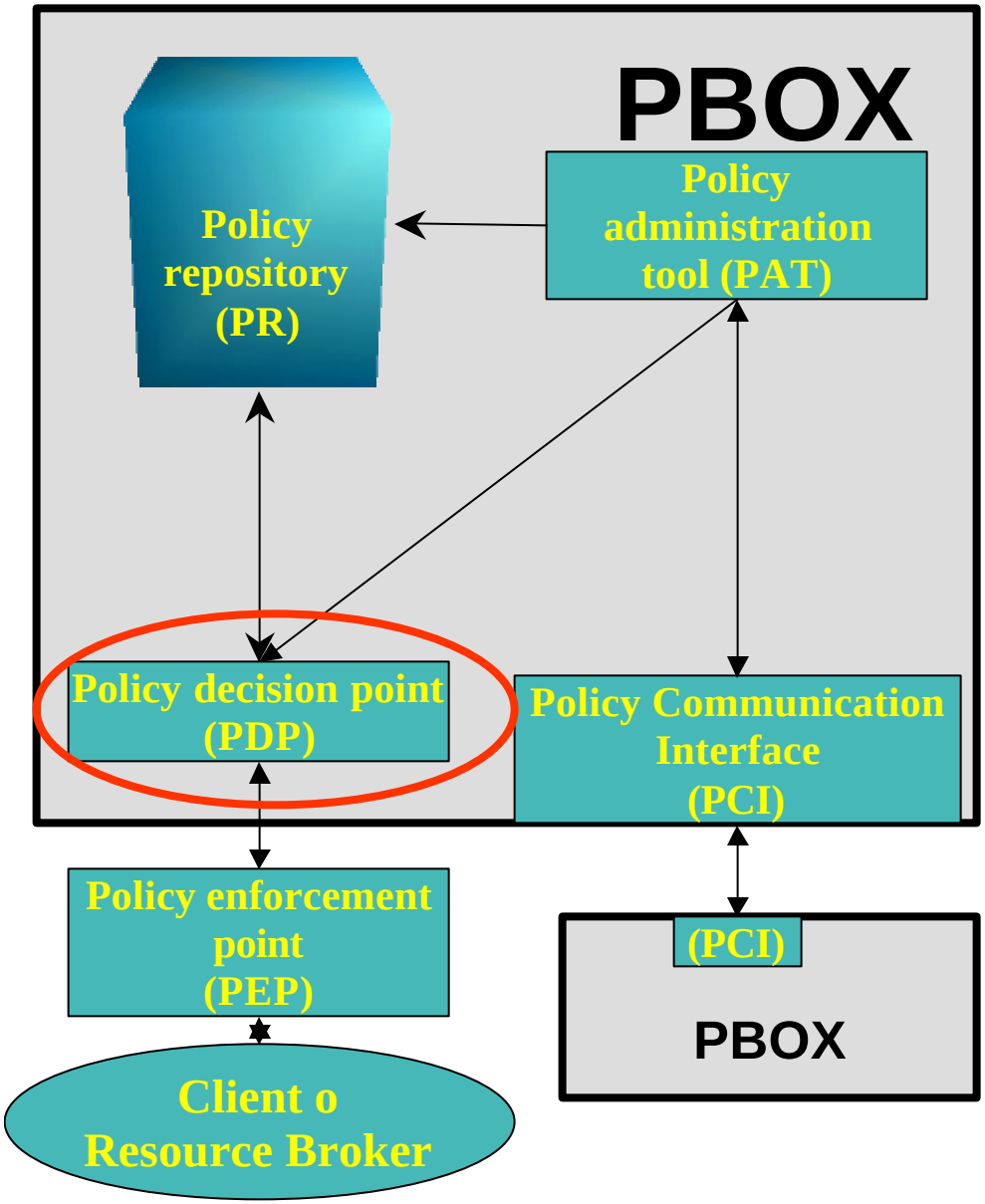
- PAT is the tool the policy admins use to insert, delete, modify their own policies, and approve or refuse policies coming from external PDPs.
- It also implements various views on the DB.
- Does not require exceptional performances.
- Holds a list of policies from other levels pending for approval.
- Communication with PDP and PR in the clear.

PBox Structure: PR



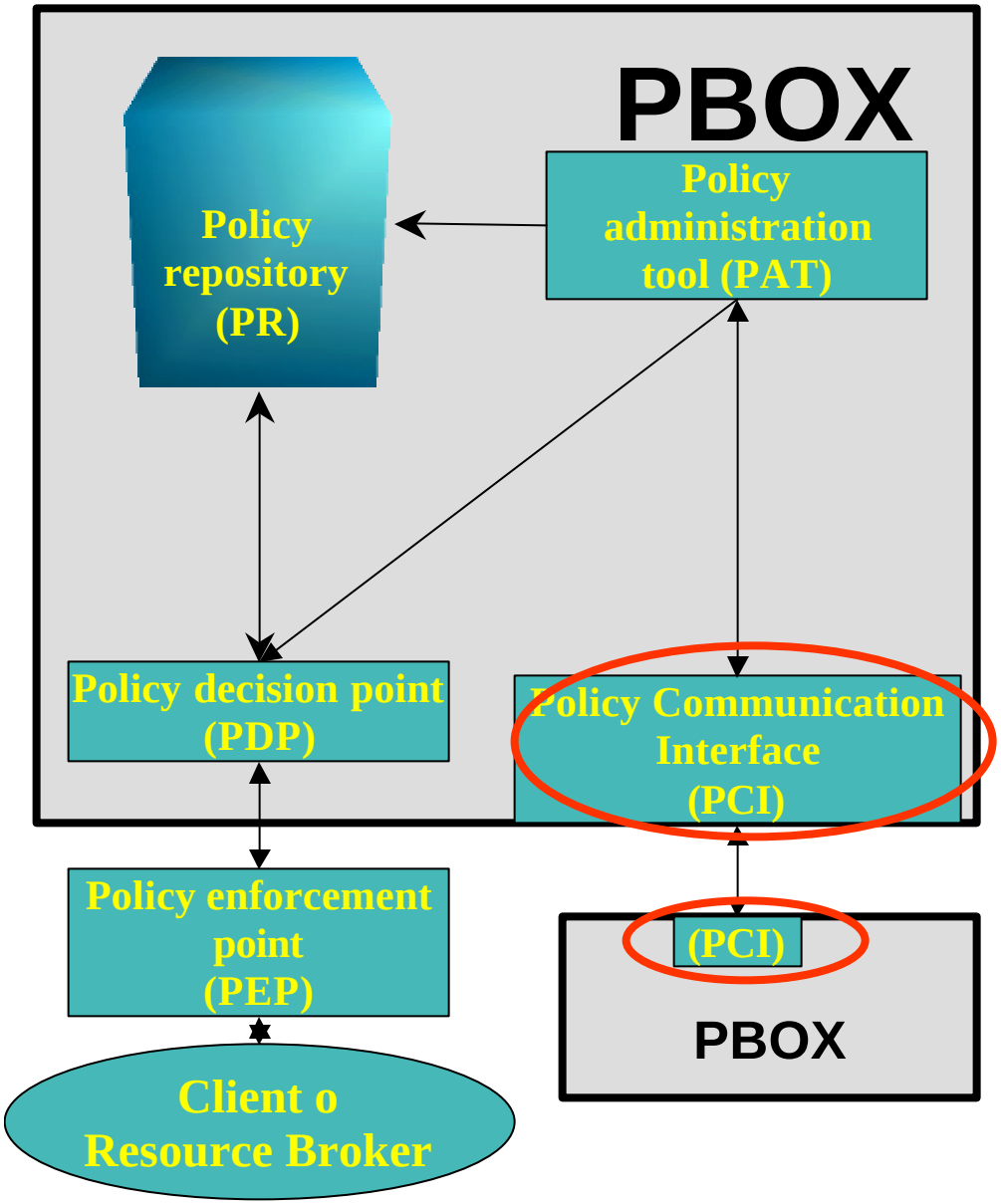
- Holds all active and old policies.
- RDBMS without need for XML support.
- Communicates with PDP and PAT in the clear.

PBox Structure: PDP



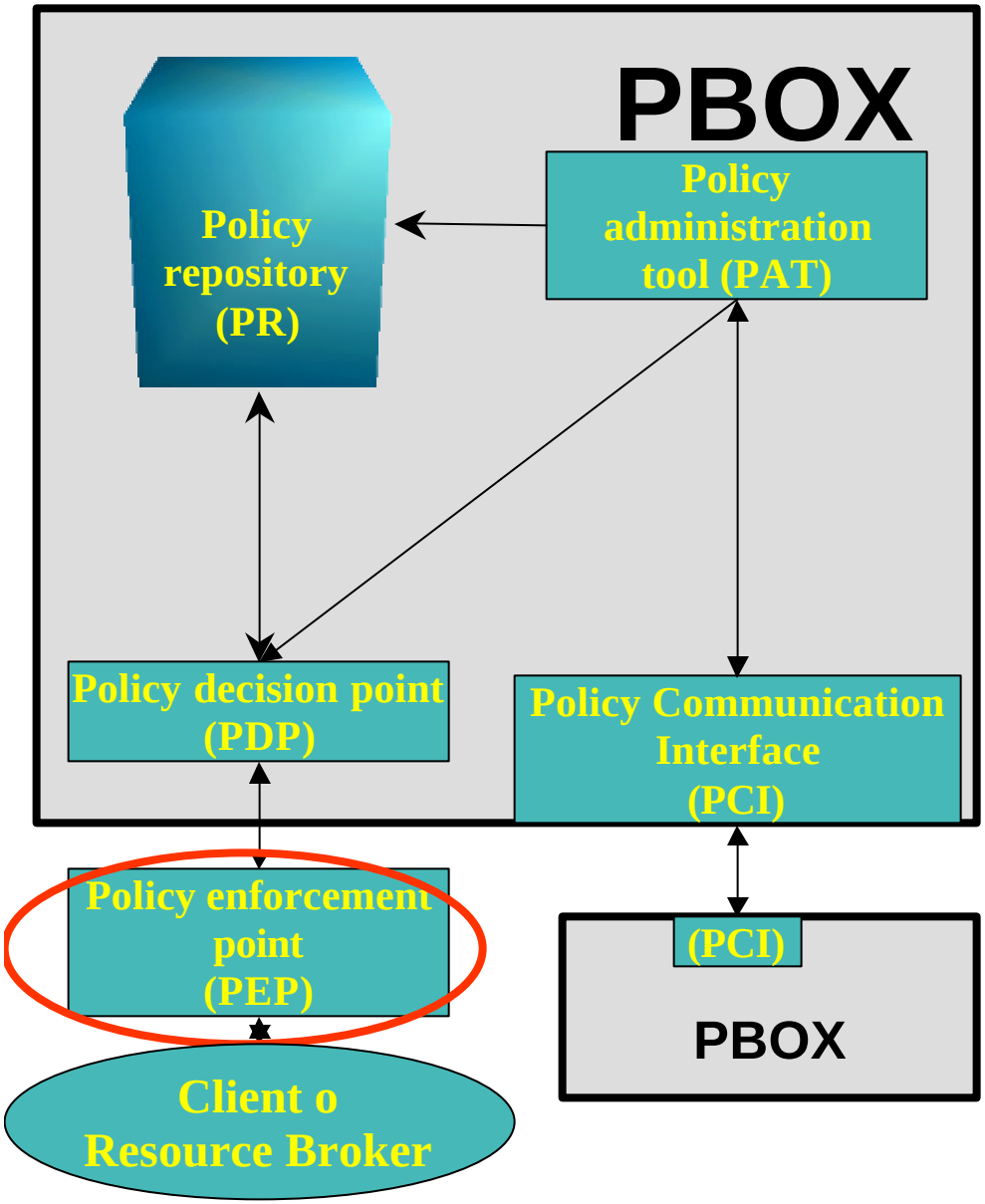
- Receives requests from clients and makes decisions depending on active policies.
- Takes full advantage of existing standards (Policies in XACML format)
- Efficiency is critical.
- Communication with PEP secure or insecure depending on configuration.
- Communication with PR on the clear.

PBox Structure: PCI



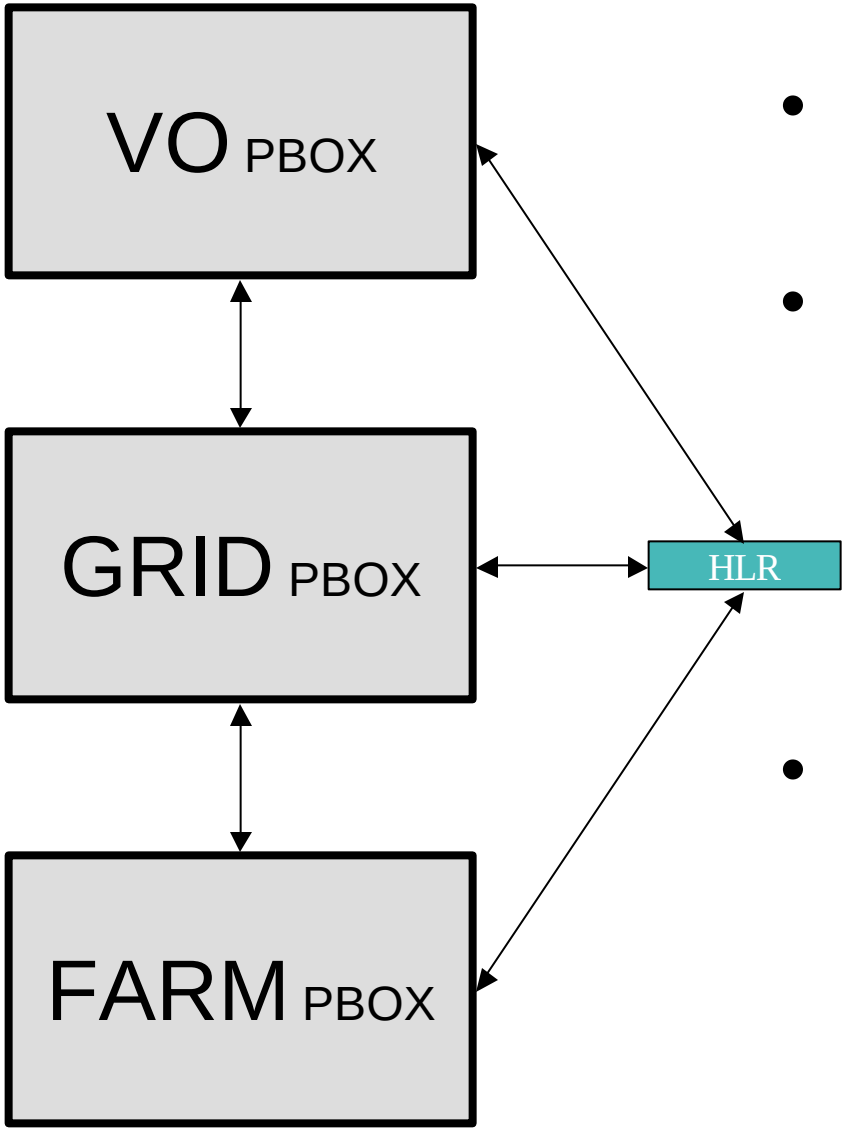
- Handles communication between different PDPs.
- Communications between PCIs are reliable, confidential, authenticated and integrity-checked. GSI will be used.

PBox Structure: PEP



- Module contacting the PDP to evaluate a request.
- Should be programmed directly into clients (RB, GTK, SE, etc…) by their developers.
- Will use an API that we will release together with P-BOX.
- Can return a string that should be interpreted by the client. These strings will be known in advance by clients' developers.

PBox Structure: HLR



- Third part software: part of the accounting system.
- Will keep track of how much resources have already been used from the set of the allotted ones.
- Not part of PBOX, but some policies require a functional accounting to be implemented.

Policy Format

- Two different formats:
  - XACML (eXtended Access Control Markup Language)
    - Completely standard as defined by the OASIS group and approved by GGF and with a well-defined semantics.
    - Will be used inside PDP and will be the “normative” form a policy.
    - Unfortunately, quite winded and difficult to understand. Site admins have already been resistant to its use.
  - PPL (P-BOX Policy Language)
    - Simple language to be used by site admins to write and review policies.
    - All PPL policies have a precise translation into XACML.
    - Much easier to read, write and understand.

XACML vs PPL

```
<?xml version="1.0" encoding="UTF-8"?>
  <Policy xmlns="urn:oasis:names:tc:xacml:1.0:policy" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:oasis:names:tc:xacml:1.0:policy cs-xacml-schema-policy-01.xsd" PolicyId="ObligationPolicy"
RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides">
  <Target>
    <Subjects> <Subject>
      <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:rfc822Name-match">
        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">users.example.com</AttributeValue>
        <SubjectAttributeDesignator DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name" AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"/>
      </SubjectMatch>
    </Subject> </Subjects>
    <Resources> <Resource>
      <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:regexp-string-match">
        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">http://server.example.com/sensitive/.*
        </AttributeValue>
        <ResourceAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string" AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"/>
      </ResourceMatch>
    </Resource> </Resources>
    <Actions> <AnyAction/> </Actions>
  </Target>
<Rule RuleId="AllowAllReads" Effect="Permit"> <Target>
    <Subjects> <AnySubject/> </Subjects> <Resources> <AnyResource/> </Resources>
    <Actions> <Action> <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">read</AttributeValue>
      <ActionAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string" AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"/>
    </ActionMatch> </Action> </Actions>
  </Target> </Rule>
  <Rule RuleId="DenyOtherActions" Effect="Deny"/>
  <Obligations> <Obligation ObligationId="LogSuccessfulRead" FulfillOn="Permit">
    <AttributeAssignment AttributeId="user" DataType="http://www.w3.org/2001/XMLSchema#anyURI">
      urn:oasis:names:tc:xacml:1.0:subject:subject-id
    </AttributeAssignment>
    <AttributeAssignment AttributeId="resource" DataType="http://www.w3.org/2001/XMLSchema#anyURI">
      urn:oasis:names:tc:xacml:1.0:resource:resource-id
    </AttributeAssignment>
  </Obligation> </Obligations>
  <Obligations> <Obligation ObligationId="LogInvalidAccess" FulfillOn="Deny">
    <AttributeAssignment AttributeId="user" DataType="http://www.w3.org/2001/XMLSchema#anyURI">
      urn:oasis:names:tc:xacml:1.0:subject:subject-id
    </AttributeAssignment>
    <AttributeAssignment AttributeId="resource" DataType="http://www.w3.org/2001/XMLSchema#anyURI">
      urn:oasis:names:tc:xacml:1.0:resource:resource-id
    </AttributeAssignment>
    <AttributeAssignment AttributeId="action" DataType="http://www.w3.org/2001/XMLSchema#anyURI">
      urn:oasis:names:tc:xacml:1.0:action:action-id
    </AttributeAssignment>
  </Obligation> </Obligations>
</Policy>
```

XACML vs PPL

```
<?xml version="1.0" encoding="UTF-8"?>
  <Policy xmlns="urn:oasis:names:tc:xacml:1.0:policy" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:oasis:names:tc:xacml:1.0:policy cs-xacml-schema-policy-01.xsd" PolicyId="ObligationPolicy"
RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides">
 <Target>
   <Subjects> <Subject>
     <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:rfc822Name-match">
       <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">users.example.com</AttributeValue>
       <SubjectAttributeDesignator DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name" AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"/>
     </SubjectMatch>
   </Subject> </Subjects>
   <Resources> <Resource>
     <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:regexp-str
       <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#strin                                    e/.*
       </AttributeValue>
       <ResourceAttributeDesignator DataType="http://www.w3.org/2001                    chema#string" AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"/>
     </ResourceMatch>
   </Resource> </Resources>
   <Actions> <AnyAction/> </Actions>
 </Target>
```

subj email *@users.example.com ; res * ; act read attr * ; cond true; dec allow ‘log subj res act’;

subj * ; res * ; act *; cond true; dec deny ‘log subj res act’ ;

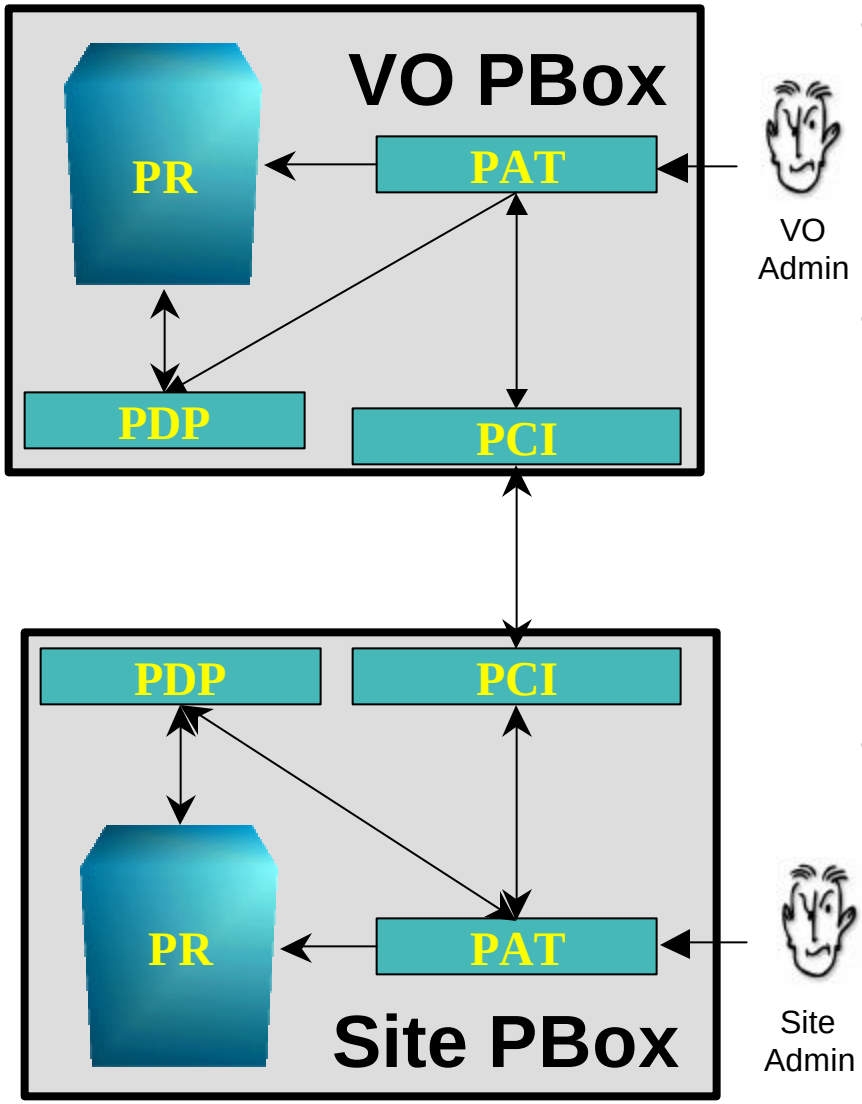
Type of Policies

- VO Policies
  - Only of interest to VOs.
  - Local sites do not need to take any special actions. In principle they do not need to know them.
- Site Policies
  - Only of interest to local sites.
  - VOs do not need to know them.
- Mixed Policies
  - Policies that are of interest both to Local sites and VOs.
  - Ban lists, Contractual agreements, etc…

Policy Examples by Type (PPL)

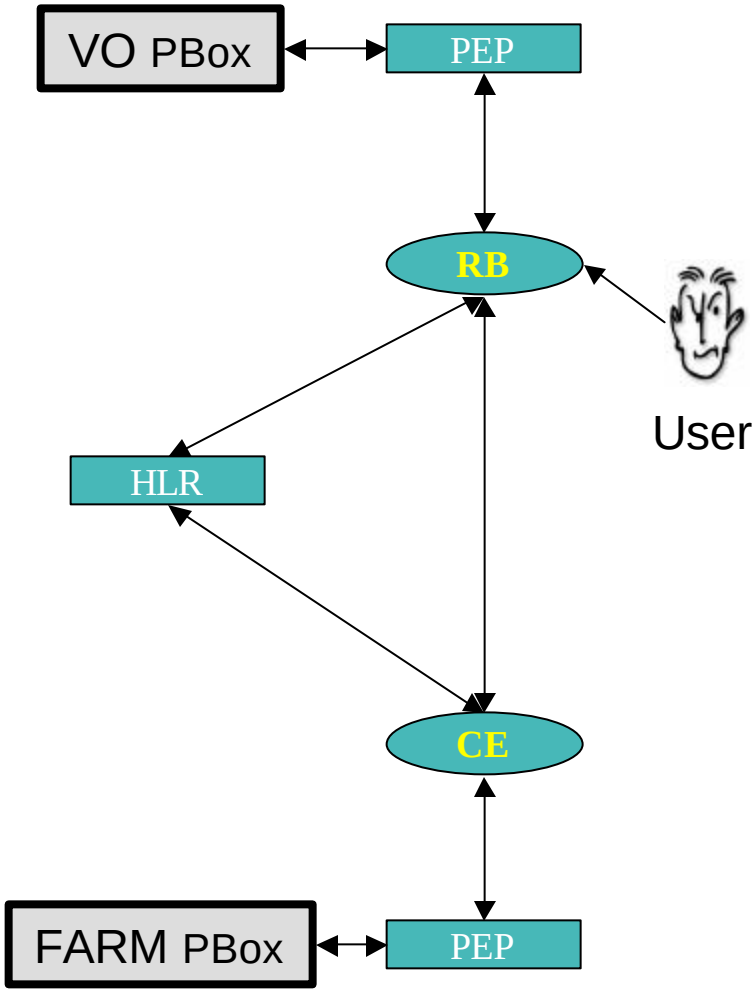
- VO Policy
  - Users belonging to the CMS-Italia subgroup of CMS may submit a maximum of 100 jobs.
  - subj attr /CMS/CMS-Italia ; obj * ; act submit ; cond user-submit-number <= 100; dec allow ‘’;
- Site Policy
  - Physical directory /disk6/cms is associated to published directory /data/cms.
  - subj attr /CMS ; obj *; act write; cond dir=/data/cms; dec allow ‘dir=/disk6/cms’;
- Mixed Policy
  - User Vincenzo Ciaschini cannot submit jobs at CNAF.
  - subj subject /C=IT/*/CN=Vincenzo Ciaschini/*; obj *.cnaf.infn.it; act submit; cond true; dec deny ‘’;

Flow Control: Policy Creation



- VO
  - VO Admin inserts a new policy into PAT.
  - PAT copies policy into PDP and PR, taking note of approval status with farms.
  - PAT sends policy to PCI, which in turn sends it to Farm PCI.
- SITE
  - Farm PCI inserts the received policy into a PAT queue, waiting for approval.
  - Farm Admin reviews received policy and decides whether to accept or refuse it.
    - If he accepts it, the policy is immediately communicated to his PDP and PR.
  - Farm Admin sends the answer to his PCI, which communicates it to VO PCI.
- VO
  - VO PCI receives answer from Farm PCI and communicates it to PAT.
  - PAT updates PR and PDP with the information about policy acceptance received from the farm and alerts VO admin.

Flow Control: Policy Enforcement



- The user submits a job to the RB.
- RB contacts HLR to get accounting information (space used, jobs, etc…)
- RB's PEP contacts VO PBox to see if the user is allowed to execute an action.
- VO PBox answers, possibly along with a list of CE where the policies allow the user to submit jobs.
- If all goes well, the job is submitted to a CE.
- CE's PEP contacts FARM PBox to verify that the user is allowed to submit a job.
- In case of a positive answer, CE contacts HLR to retract tokens.
- If latest operatoin went okay, the job is effectively submitted.

NOTES

Without HLR, policies requiring a VO-wide view of the grid cannot be implemented.

If a user skips the RB to submit a job directly to CE, VO policies are still enforced by the CE, and so the user risks submitting on a farm where policies do not allow him to submit, and so the operation fails.

Indicative Timeline

- Proposal: Here it is!
- Alpha release: late july 2004.
- Alpha testing and fixes: late july - late september 2004.
- Beta release: late december 2004.
- Beta testing and fixes: late december 2004 - late march 2005.
- Release 1.0: late july 2005.

Group membership and pointers

- Home Page:
  - INFNForge on http://infnforge.cnaf.infn.it/projects/pbox
- Group Members
  - Ciaschini Vincenzo (ciaschini@cnaf.infn.it)
  - Ferraro Andrea (andrea.ferraro@cnaf.infn.it)
  - Rubini Gianluca (grubini@cnaf.infn.it)
  - Zappi Riccardo (riccardo.zappi@cnaf.infn.it)
- External Collaborators
  - Guarise Andrea (guarise@to.infn.it)
  - Caltroni Andrea (andrea.caltroni@pd.infn.it)