Defence Research and Development Canada
Recherche et développement pour la défense Canada

DEFENCE R&D DÉFENSE

Policy Based Network Management System Design Document

J. Spagnolo, D. Cayer

The scientific or technical validity of this Contract Report is entirely the responsibility of the contractor and the contents do not necessarily have the approval or endorsement of Defence R&D Canada.

Defence R&D Canada – Ottawa
CONTRACT REPORT
DRDC Ottawa CR 2005-109
September 2005

Canada

Policy Based Network Management System Design Document

J. Spagnolo, D. Cayer
NRNS Incorporated

NRNS Incorporated
4043 Carling Avenue
Ottawa, ON
K2K 2A3

Contract Number: W7714-3-800/001/SV

Contract Scientific Authority: Dr. S. Zeber, DRDC Ottawa, (613) 991-1388

Contract Scientific Advisor: Mr. Tim Symchych, CRC, (613) 949 - 3070

The work described in this report was sponsored jointly by the Department of National Defence under the work unit 15BF and by the Communications Research Center Canada under the work units 15CS and 15CV.

Defence R&D Canada – Ottawa

Contract Report

DRDC Ottawa CR 2005-109

September 2005

Terms of release: The scientific or technical validity of this Contract Report is entirely the responsibility of the contractor and the contents do not necessarily have the approval or endorsement of Defence R&D Canada or the Communications Research Centre.

Terms of release: The information contained herein is proprietary to Her Majesty and is provided to the recipient on the understanding that it will be used for information and evaluation purposes only. Any commercial use including use for manufacture is prohibited. Release to third parties of this publication or information contained herein is prohibited without the prior written consent of Defence R&D Canada.

Document Revision History

Version 0.1	31-Mar-2005	Complete DRAFT submitted Scientific Authority for review.
Version 1.0	31-Mar-2005	Incorporated comments from Defence R&D Canada and the Communications Research Centre.
Version 1.1	23-Sep-2005	Updated document to incorporate the development of the Policy Server.

Table of Contents

List of Figures

1. Introduction

Policy Based Network Management (PBNM) systems provide an automated means to configure and administer Policy Enforcement Point (PEP) devices such as virtual private network (VPN) gateways, firewalls and routers. The Policy Decision Point (PDP) takes high level policies as input and produces lower level PEP specific policies as output. The PBNM system can process different types of policies. When evaluating policies, the PDP must identify and resolve conflicts within competing policies as well as take into consideration external factors such as the time-of-day and the current threat level.

A PBNM system alleviates the need for network administrators to manually configure numerous network devices in order to implement local policy changes. We also introduce the concept of policy negotiation for inter-domain policies[1] such as inter-domain security policies. Negotiable policies are not complete policy documents and therefore the PDP cannot directly implement them. Instead, the local PDP must exchange policy proposals with a PDP in a remote administrative domain. Policy proposals contain all the negotiation parameters needed by the other party to correctly evaluate the proposed policy against the local policy. A PDP can accept a proposed policy in whole or in part or it can reject the proposed policy. If both parties accept the other party's proposed policy in whole or in part, each party merges the local and remote policy proposals to form a complete policy document that the PDP can implement. The PBNM system automatically reconfigures network devices as required to implement negotiated policies.

2. Purpose

This document presents an architecture and system level design for a generic PBNM framework that supports policy negotiation. The PBNM system facilitates the compilation of policies, the storage of policies, the exchange of policies, the evaluation and negotiation of policies, as well as the implementation and enforcement of policies. Section 5 provides an overview of a specification for negotiable inter-domain security policies. The PBNM system can also support static policies such as Quality of Service (QoS) policies, which are implemented directly within the local administrative domain and do not require negotiation.

3. Architecture Overview

The PBNM architecture includes five components as shown in Figure 1. The PDP serves as the system's central nervous system and interacts with all the other components that include a Policy Repository, a Policy Editor, PEP devices as well as Policy Negotiation Proxy (PNP) devices.

Four of the five components reside in the controlled network. These components have no requirement to communicate with systems that reside in the uncontrolled network[2]. The

[1] Note that inter-domain policies may also be intra-organizational policies as some "domains" will be identified parts of larger organizations.

[2] The PDP may have to configure devices, such as border routers, that reside outside of the controlled network perimeter.

PNP device however must undertake a dialogue with remote PNP devices. As such, a separate system located in a demilitarized zone (DMZ) houses the PNP. Security conscious organizations should also consider housing the PDP system in its own security zone within the controlled network. Since the PDP device controls the configuration of security devices such as firewalls and VPN gateways, the PDP should be offered additional protection against compromise.

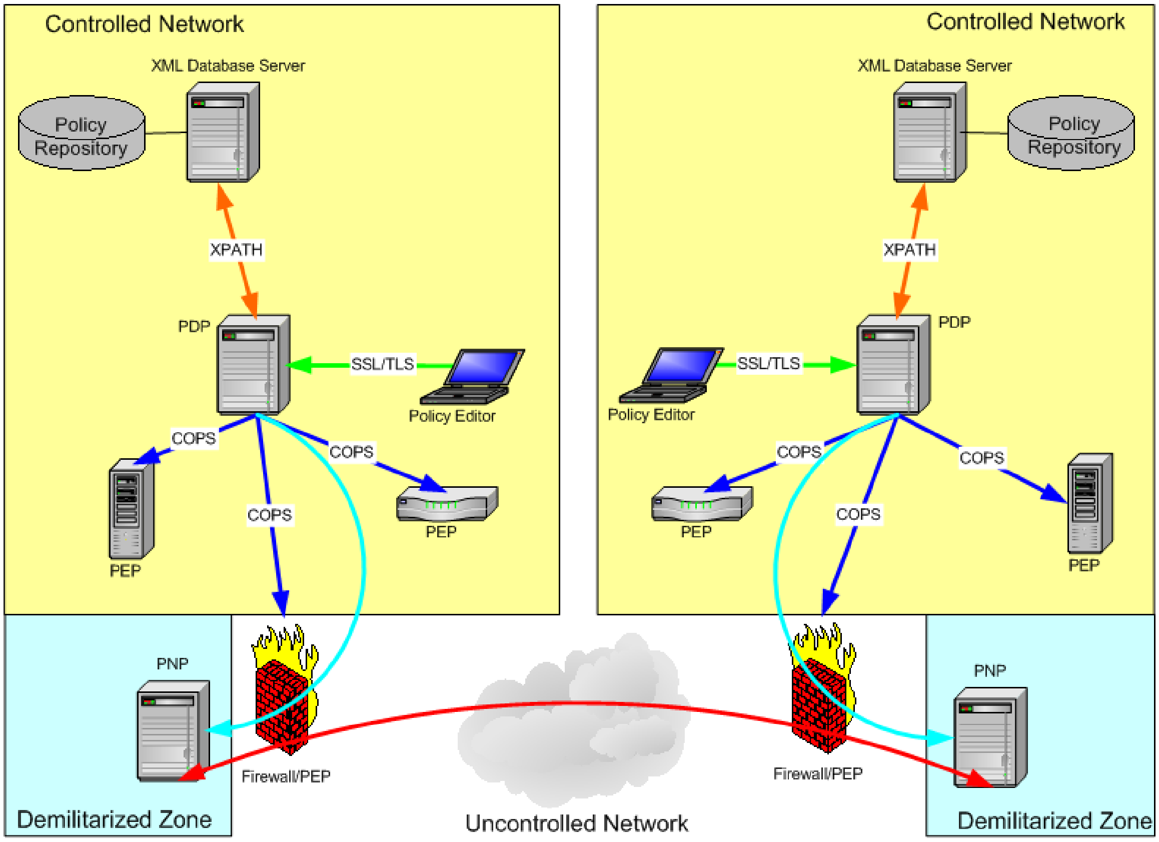

Figure 1 - PBNM System Architecture

The PDP communicates with the Policy Repository using XPATH [XPATH] as the query language. Communication with PEP devices is facilitated with the use of the Common Open Policy Service (COPS) protocol [RFC2748]. The Policy Editor makes use of an authenticated/secure communication channel based on Secure Socket Layer (SSL) or Transport Layer Security (TLS). Communication between the PDP and the PNP as well as inter-PNP communication is achieved with authenticated/secure communication channels – likely SSL/TLS.

The PBNM system provides a generic framework for processing, negotiating and implementing policies. As such the framework does not possess any knowledge of the policy document contents. Although policies are expressed using the eXtensible Markup Language (XML), the framework does not understand the structure or format of the policy document. This results in a highly extensible PBNM system that can support different types of policies without requiring any modifications to the framework. In reality however, policies are not truly generic and as such some part of the system must possess the necessary knowledge of policies to perform policy specific processing. For this purpose we define a software component of the PDP called the Policy Processing Unit (PPU).

The PDP must include a PPU implementation for each type of supported policy. However, the object oriented design of the PBNM system allows for common functionality to be implemented once within PPU super classes and inherited by policy specific PPU classes. The BasePPU class provides the generic capabilities required by all PPU implementations. The BasePPU class accepts new policies from the Policy Editor but does not perform any policy specific processing. The NegotiablePPU class extends the BasePPU class and implements the policy negotiation protocol described in section 4.8. As with the BasePPU class, the NegotiablePPU class does not perform any policy specific processing. Instead, the policy specific PPU classes implement all the policy specific processing. Policy specific PPU classes that require policy negotiation must extend the NegotiablePPU class.

The Policy Server collects low level policies from PPUs and disseminates the low level policies to PEP devices using the COPS-PR protocol. The PPU supplies policy updates to the Policy Server as a list of old policies and new policies. The Policy Server determines the difference between the old policies and the new policies and produces deltas in the form of a list of policy (add/remove) decisions, which it forwards to the appropriate PEP devices.

4. Concept of Operation

This section provides a more detailed description of the various PBNM system components and outlines the interactions between them. The PBNM system is driven by high-level policy documents. This section describes how these policy documents are compiled, validated and stored as well as how these policy documents form the basis for policy negotiation and the subsequent policy implementation.

This section also introduces the concept of policy negotiation objects and policy negotiation artifacts, which are the residual policy objects produced as part of the negotiation process. Policy negotiation objects include Policy Proposal objects, Negotiation Transcript objects, Merged Policy objects, Policy Refresh objects and Policy Withdraw objects. Policy negotiation objects are described in detail in section 4.8.

4.1 Policy Repository

The PDP stores policy documents as well as policy negotiation artifacts in the Policy Repository. The Policy Repository is a native XML database that allows XML encoded policies to be stored in their native XML form without the need to map the policies to some other data structure. XPATH, an XML query language, facilitates addressing of parts within an XML document.

4.2 Integrity and Authenticity

The PDP ensures the integrity and authenticity of all policy objects, whether policy documents or policy negotiation objects, with the use of digital signatures. The creator of a policy object applies its digital signature to the policy object when it creates the object. In the case of a policy document, the digital signature belongs to the authorized individual who compiled the policy document. In the case of policy negotiation objects, the PDP that created the policy object authenticates the policy object with its digital signature[3].

Policy objects that are not created by the local PDP contain a second digital signature when stored in the policy repository. The PDP applies its digital signature on the object but preserves the original digital signature of the creator. In the case of policy documents, the original digital signature identifies the authorized individual that compiled the policy document. In the case of a policy negotiation object created by a remote PDP, the original digital signature identifies the remote PDP that created the policy object.

4.3 Certificate Status Checking

All PBNM components check the revocation status of all certificates used for signing policy documents and policy negotiation objects as well as certificates used to authenticate communication channels.

4.4 Authorization

The PDP makes use of the identity contained within digital signatures to perform authorization checks. The types of authorization checks supported by the PBNM system include:

1. Which authorized individuals can edit policy documents? This is based on the digital signature applied to the policy document.

2. Which authorized individuals can submit policy documents to the PDP? This is based on the certificate used to authenticate the communication channel between the Policy Editor and the PDP.

3. Which authorized remote PDP entities can negotiate on behalf of a remote administrative domain? This is based on the digital signature applied to policy negotiation objects.

4. Which local PNP entities can facilitate policy negotiation on behalf of the local administrative domain? This is based on the certificate used to authenticate the communication channel between the local PNP and the local PDP.

5. Which authorized remote PNP entities can facilitate the policy negotiation on behalf of a remote administrative domain? This is based on the certificate used to authenticate the communication channel between the remote PNP and the local PNP.

[3] High security environments could encrypt the database contents to prevent unauthorized viewing of sensitive information.

When the PDP system starts, the PDP retrieves the Trusted Authorities file from the Policy Repository. This XML document identifies the trusted certificate authorities for the remote administrative domains that the PDP can interact with. The Trusted Authorities file identifies the trusted certificate authority for a specific administrative domain by the distinguished name of the certificate authority as well as by the key identifier of the certificate authority's key.

The PBNM system protects the Trusted Authorities file with digital signatures. The Trusted Authorities file includes two digital signatures. The first digital signature was applied by the trusted individual that compiled the file and the second digital signature was applied by the local PDP that validated the file.

4.5 Resolver Service

When the PDP system starts, the PDP retrieves Resolver Mapping files from the Policy Repository. These XML document provide a mapping of high level constructs such as administrative domains, security classes and services to low level constructs such as Internet Protocol (IP) addresses, protocols, port numbers, and cryptographic algorithms..

The PBNM system protects the Resolver Mapping files with digital signatures. The Resolver Mapping files include two digital signatures. The first digital signature was applied by the trusted individual that compiled the file and the second digital signature was applied by the local PDP that validated the file.

4.6 Policy Validation

The PBNM system makes use of a XML schema definition to define the correct structure and format of a specific policy document. Although the XML schema definition ensures that the document is well formed (i.e. ordinality, cardinality, exclusivity), the schema alone cannot determine if the policy conforms to the policy specification. The PBNM system includes a generic Policy Validation Engine that validates the policy document against the policy specification. The rules for the Policy Validation Engine are expressed in XML. The Policy Validation engine is described in [IDPE].

4.7 Policy Retrieval, Compilation and Submission

Authorized individuals compile policy documents using the Policy Editor that is specific to that particular type of policy. The Policy Editor establishes a secure communication channel to the PDP authenticated with the individual's public certificate in order to retrieve the latest policy document. The PDP extracts the certificate used to authenticate the communication channel from the Policy Editor and confirms that the individual is authorized to compile policy documents. Since more than one instance of the Policy Editor may attempt to edit the latest version of the policy document, the PDP must lock the policy document to prevent concurrent editing of the document.

When the authorized individual completes the editing of the policy document, the Policy Editor ensures that the policy document conforms to the appropriate XML schema definition, and makes use of the Policy Validation Engine to validate that it fully adheres to the associated policy specification. When the authorized individual is ready to submit the policy document to the PDP, the Policy Editor applies the individual's digital signature to the updated policy document before presenting it to the PDP. The message exchanged between the Policy Editor and the PDP contains a header that identifies the type of policy included in the payload. The PDP uses this information to forward the policy document to the associated PPU for further processing.

The PPU validates the digital signature on the policy document to confirm its integrity and to authorize its use. The PPU ensures that the policy document conforms to the appropriate XML schema definition, and makes use of the Policy Validation Engine to validate that it fully adheres to the associated policy specification. If the PPU deems the submitted policy document to be authentic, authorized and valid - it adds a unique identifier to the policy document; it applies the PDP digital signature to the policy document (while preserving the original signature); and it stores the submitted policy document in the Policy Repository.

The PPU compares the newly submitted policy against the current policy. For static policies that do no require negotiation, the PPU informs the Policy Server of the changes in the policy so the Policy Server can reflect the changes within affected PEP devices. For policies that require negotiation, the PPU begins negotiation with a new administrative domain, withdraws its previously negotiated policies with a discarded administrative domain, or renegotiates with an existing administrative domain. When policies must be renegotiated, the PPU leaves the associated configuration within PEP devices in place but sets a time limit for the renegotiation process to complete. If the renegotiation process does not complete successfully within the allotted time, the PPU instructs the Policy Server to remove the associated configuration information from all affected PEP devices.

4.8 Policy Negotiation

The PPU manages the policy negotiation process with a remote administrative domain. The policy negotiation process yields five types of policy negotiation objects – Policy Proposal objects, Negotiation Transcript objects, Policy Refresh Objects, Merged Policy objects, and Policy Withdraw objects.

All policy negotiation objects contain a policy type, a policy object type, unique identifier and a validity period. All policy negotiation objects include a digital signature that identifies the PDP that created the object. The digital signature in conjunction with the unique identifier and validity period protects against impersonation and replay attacks. Each PDP stores both locally generated and received policy negotiation objects as artifacts in the local Policy Repository for audit purposes as well as to quickly re-establish negotiated policies without incurring the full cost of the negotiation process. For policy negotiation objects generated by a remote PDP, the local PDP applies a second digital signature to the object to signify that the object was previously processed by an authorized PDP in the local administrative domain.

The PPU produces a Policy Proposal object for each administrative domain identified within the current policy. The PPU responds to a remote Policy Proposal object with a Negotiation Transcript object. The Negotiation Transcript object is a copy of the original remote Policy Proposal object, but it also includes a statement-by-statement response in the form of a status (Accept, Reject or Not Evaluated) as well as an optional result and an optional reason. The status associated with the top-level node in the Negotiation

Transcript object indicates whether the PPU accepts the remote Policy Proposal object in whole or in part or whether the PPU rejects the remote Policy Proposal object in its entirety. The transmission of an acceptable Negotiation Transcript object in conjunction with the reception of an acceptable Negotiation Transcript object results in a successful policy negotiation. A Policy Proposal object contains the unique identifier of the current policy document thereby linking it to a specific instance of a policy document. A Negotiation Transcript object contains the unique identifiers of both the local and remote Policy Proposal objects thereby linking it to a specific instance of those objects. Section 4.8.3 provides a complete description of Policy Negotiation object linkage.

Policy Refresh objects serve three purposes - they signal the completion of a successful negotiation sequence; they indicate that previously negotiated policies remain in force, or they re-establish previously negotiated policies without incurring the full cost of the negotiation process. For instance Policy Refresh objects re-establish previously negotiated policies should the two parties lose connectivity for an extended period of time. This can be caused by prolonged network outages or when a PDP is unavailable to do a system failure. A Policy Refresh object contains the unique identifiers of both the local and remote Negotiation Transcript objects thereby linking it to a specific instance of those objects.

The PPU creates a Merged Policy object at the completion of a successful negotiation sequence. A Merged Policy object combines the information from the locally generated Negotiation Transcript object and the Negotiation Transcript object received from the remote PDP to produce a complete policy document that the PDP can implement. The PPU informs the Policy Server of the changes in the negotiated policy so the Policy Server can reflect the changes within affected PEP devices. A Merged Policy object contains the unique identifiers of both the local and remote Negotiation Transcript objects thereby linking it to a specific instance of those objects.

The PPU sends a Policy Withdraw object to a remote PDP to indicate that it is withdrawing the previously negotiated policies. The PPU informs the Policy Server so the Policy Server can remove all associated configuration from affected PEP devices. The Policy Withdraw object does not reference other Policy Negotiation objects since a PDP must have the ability to respond to Policy Proposal and Policy Refresh objects when it does not possess any previous negotiation state information.

4.8.1 Policy Negotiation Sequence

A policy negotiation sequence requires the mutual exchange of Policy Proposal objects, acceptable Negotiation Transcript objects and Policy Refresh objects. If the policy negotiation sequence yields a successful negotiation, the periodic exchange of Policy Refresh objects maintains policy negotiation state.

Figure 2 illustrates a policy negotiation sequence. The negotiation state, which is shown for each peer, includes:

WAIT PP The local peer sent a Policy Proposal to the remote peer and is now waiting for a Policy Proposal from the remote peer.

WAIT NT	The local peer received the remote peer's Policy Proposal and sent an acceptable Negotiation Transcript to the remote peer. The local peer is now waiting for a Negotiation Transcript from the remote peer.
WAIT PR	The local peer received the remote peer's acceptable Negotiation Transcript and sent a Policy Refresh to the remote peer. The local peer is now waiting for a Policy Refresh from the remote peer.
NEGOTIATED	The local peer received the remote peer's Policy Refresh that signalled the successful completion of the negotiation sequence.

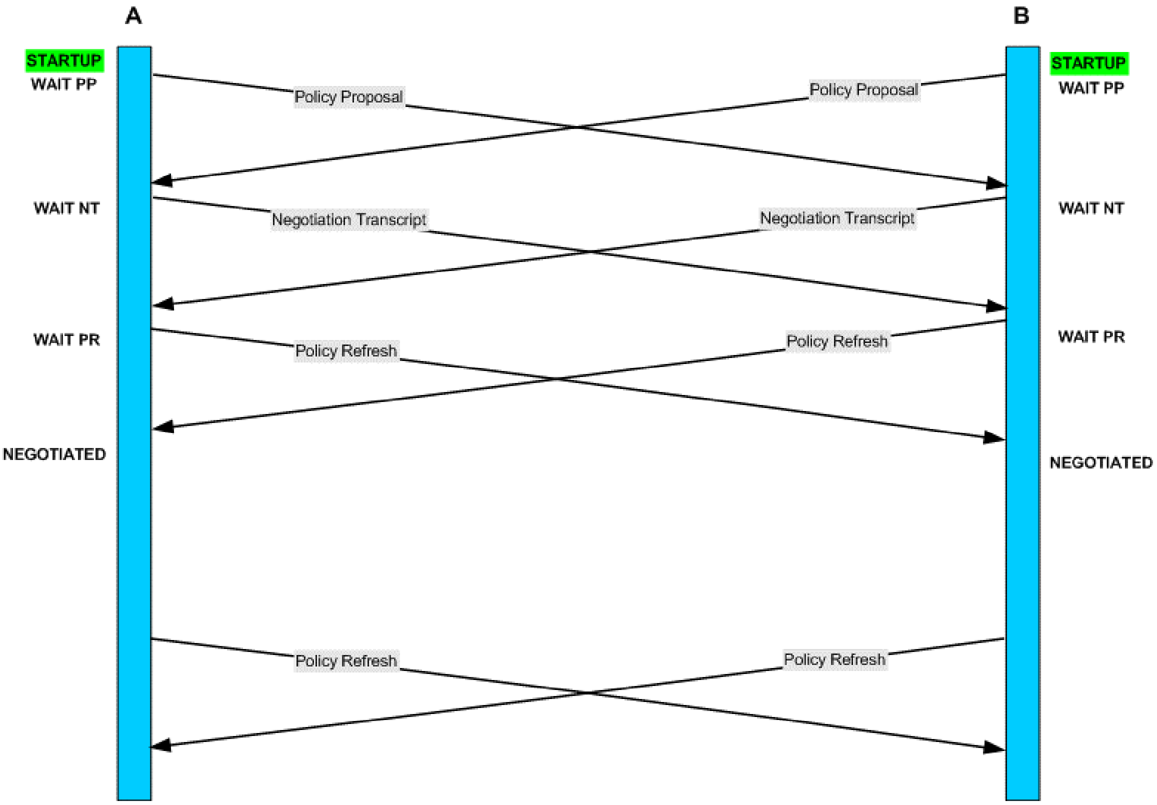

Figure 2 - Synchronized Policy Negotiation Sequence

4.8.2 Policy Negotiation State Transitions

Figure 3 illustrates the policy negotiation state transition diagram. In addition to the states previously defined in section 4.8.1, Figure 3 introduces the following states:

IDLE — A policy negotiation sequence ended unsuccessfully. Wait for either the local peer or remote peer to alter their policy.

LOCAL CONFLICT — A local conflict prevents the compilation of a Policy Proposal. Only a change to the local policy allows for a transition out of this state.

DISENGAGED — The remote peer was removed from the local policy.

REMOTE WITHDRAW — The remote peer sent a Policy Withdraw. Only the reception of a Policy Proposal from the remote peer allows for a transition out of this state.

TRY REFRESH — The remote peer stopped sending periodic Policy Refresh objects. Attempt to refresh the previously negotiated policy.

State transitions typically include the event that causes the state transition and the action to take in advance of the state transition. The text embedded within lines interconnecting the various states includes the event that caused the state transition on top and the action to take before the state transition on the bottom. Lines leading into a yellow decision diamond only may include an event, while lines leaving the decision diamond only may include an action.

The events that drive the state machine include the reception of policy negotiation objects, an indication that the local policy may have changed and should be rechecked (RECHECK POLICY), an indication that the remote peer was removed from the local policy (WITHDRAW POLICY), as well as timer events that ensure that the state machine does not stall.

An indication that the local policy may have changed and should be rechecked can result in three different outcomes.

1. The Policy Proposal for the remote peer did not change. No action is required and no state transition follows. The event is ignored.

2. A local conflict prevents the compilation of a Policy Proposal for remote peer. Send a Policy Withdraw to the remote peer and proceed to the LOCAL CONFLICT state. This does not apply to the REMOTE WITHDRAW or DISENGAGED states.

3. The policy proposal for the remote peer changed, send a new Policy Proposal and proceed to the WAIT PP state. This does not apply to the REMOTE WITHDRAW or DISENGAGED states.

When the local peer receives a Policy Proposal, it evaluates the Policy Proposal to determine if it must respond with an acceptable Negotiation Transcript or a rejected Negotiation Transcript. If the local peer sends an acceptable Negotiation Transcript, it proceeds to the WAIT NT state. If the local peer sends a rejected Negotiation Transcript, it proceeds to the IDLE state and waits for either the local peer or remote peer to alter their policy.

A timer event during the negotiation sequence typically causes the entire negotiation sequence to be restarted. The NEGOTIATED state includes two active timers. One controls the periodic transmission of Policy Refresh objects, while the other ensures the periodic reception of Policy Refresh objects from the remote peer.

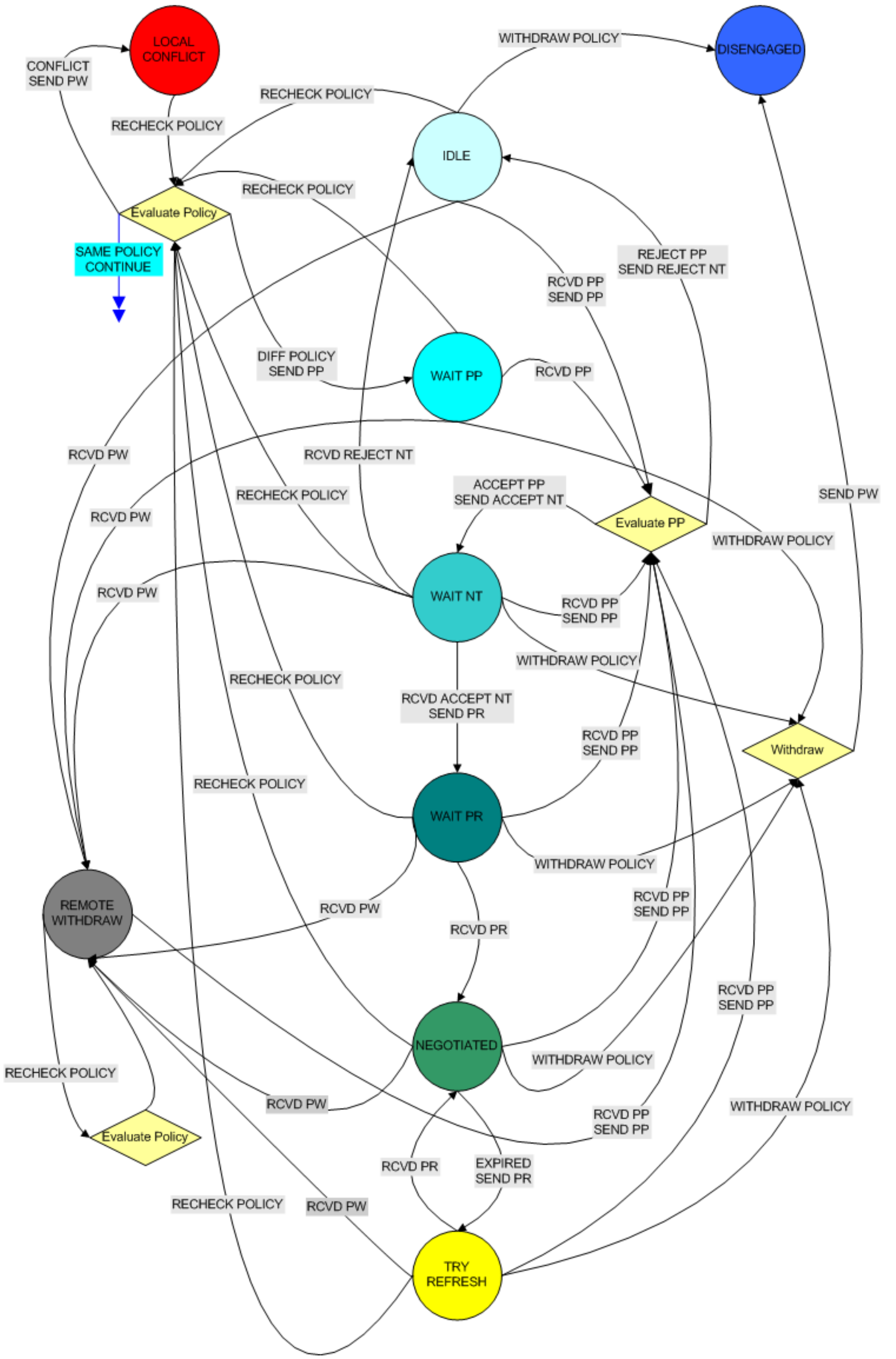

Figure 3 – Policy Negotiation State Transitions

4.8.3 Policy Negotiation Object Linkage

Figure 4 illustrates how the PDP system links the policy negotiation objects for a specific remote administrative domain to a specific instance of a policy document.

The Merged Policy object contains the identifiers of specific instances of both the local and remote Negotiation Transcript objects. Both the local and remote Negotiation Transcript objects contain the identifiers of specific instances of both the local and remote Policy Proposal objects. The local Policy Proposal object contains the identifier of a specific instance of a policy document.

Like the Merged Policy object, the Policy Refresh object contains the identifiers of specific instances of both the local and remote Negotiation Transcript objects and as such is indirectly linked in the same fashion to the specific instance of a policy document.

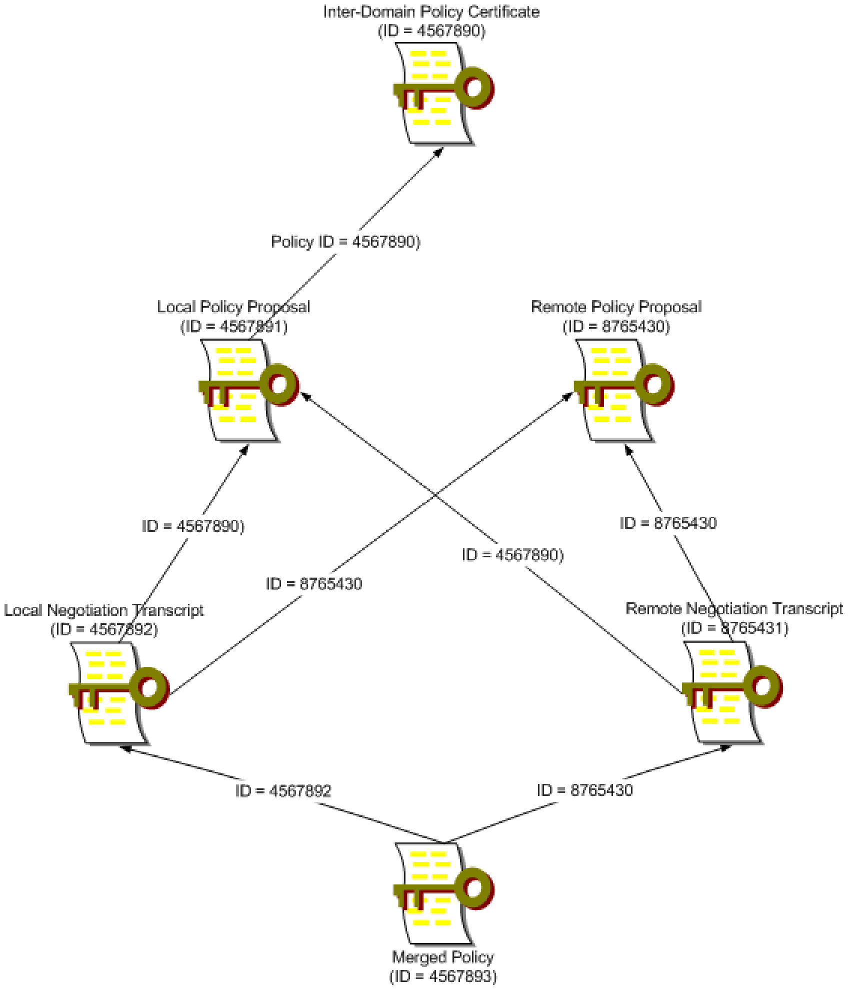

Figure 4 - Policy Object Linking

4.8.4 The Policy Negotiation Proxy

The Policy Negotiation Proxy (PNP) facilitates the policy negotiation dialogue on behalf of the PDP. The PDP communicates with the PNP using a single authenticated and secured communication channel. The local PNP establishes an authenticated and secured communication channel to the PNP device in each remote administrative domain that it must negotiate with. The local PNP uses its digital credentials to authenticate the communication channel to a remote PNP device. The local PNP uses the remote entity's certificate to ensure that the remote administrative domain authorized the remote entity to act as a PNP. The communication channel between the local PDP and the local PNP is also authenticated with certificates and also undergoes an authorization check by both the PDP and the PNP.

The PDP transmits control messages to the local PNP when it requires that the local PNP engage communication with a specific remote PNP device for a specific type of policy or that it disengage from a specific remote PNP device for a specific type of policy. The engage control message identifies the remote administrative domain and supplies the network information needed by the local PNP to communicate with the remote PNP device. Since the local PNP must interact with numerous PNP devices in remote administrative domains, the message header used to carry policy negotiation objects identifies the type of policy to permit policy negotiation objects for numerous negotiation sequences to be multiplexed within the single PDP/PNP communication channel.

In addition to the type of policy, the message header used to carry policy negotiation objects also identifies the type of policy negotiation object, the object unique identifier, as well as the validity period for the object. This allows the policy negotiation objects for different policy types to be multiplexed within the single PNP/PNP communication channel and allow independent negotiation sequences for different types of policies.

Like the PDP, the PNP does not interpret the contents of policy negotiation objects – it considers them to be opaque. In addition, the PNP does not maintain negotiation state. It simply transmits policy negotiation objects as and when requested by the local PDP. The PDP takes care of acknowledging the receipt of policy negotiation objects to the remote PDP.

The local PNP transmits status messages to the local PDP to report the state of the communication channel with remote PNP devices. When the local PNP establishes communication channel with a remote PNP, the local PNP transmits a status message to the local PDP indicating a valid communication channel with the specified remote PNP. If a previously established communication channel with a remote PNP is lost, the local PNP transmits a status message to the local PDP indicating a failed communication channel with the specified remote PNP. The PDP only transmits policy negotiation objects to the local PNP when the associated communication channel with the remote PNP is available.

Although the PNP facilitates the policy negotiation between two administration domains, the PNP possesses no authority to influence the negotiation since all Policy Proposal objects must be signed by the PDP. A compromised PNP could however interfere with the negotiation, thus creating a denial of service.

4.9 Policy Enforcement Point

The PDP takes static policy documents or Merged Policy objects to produce a lower level policy suitable for PEP configuration. The PEP devices employ the Common Open Policy

Service (COPS) protocol for Policy Provisioning (COPS-PR) [RFC3084] to acquire its policies from the PDP.

COPS aware PEP devices must be configured with the network address of one or more Policy Servers within PDP systems. When a PEP device boots, it establishes a COPS session to its primary PDP and supplies information to the PDP in the form of a COPS configuration request that describes the device's capabilities (i.e. type of device, role). The PDP responds with all provisioned policies that are relevant to the PEP device. COPS structures its policy data as a tree based namespace referred to as Policy Information Base (PIB). Branches of the PIB represent classes (i.e. the type of configuration data), while the leaves of the tree represent actual instances of those classes.

The PEP device maintains its COPS session active to the PDP at all times. This allows the PDP to push policy changes (additions, deletions) to the PEP when relevant policies are modified. The PEP device may transmit at any time a new configuration request to the PDP to report changes in its capabilities or simply to disseminate status information to the PDP. The use of the persistent COPS session between the PDP and the PEP allows both devices to detect immediately when the other device reboots or fails.

The COPS protocol provides a mechanism to specify the encoding for objects encapsulated within COPS protocol data units. The COPS-PR designers identified a single encoding scheme for this purpose - Basic Encoding Rules (BER), but COPS can be extended to also carry objects encoded using alternative means such as XML.

4.10 Policy Server

High level policies described within static policy documents or Merged Policy objects produce lower level policies for many different types of PEP devices. Only the PPU implementations can produce the lower level policies since only the PPU implementations possess the necessary knowledge of policies to perform policy specific processing.

A PDP software component called the Policy Server collects low level policies from PPUs and disseminates the low level policies to PEP devices using the COPS-PR protocol. The PPU supplies policy updates to the Policy Server as a list of old policies and new policies. The Policy Server determines the difference between the old policies and the new policies and produces deltas in the form of a list of policy (add/remove) decisions. The Policy Server disseminates policy decisions to PEP devices based on the device type or role that the PEP device claims to fulfill.

Section 5 provides an overview of a specification for negotiable inter-domain security policies. Section 0 identifies the type of PEP devices required by the inter-domain security policy PPU to implement and enforce its policies.

4.11 System Restart

The PDP stores all policy documents and policy negotiation artifacts to persistent storage in the Policy Repository. If for any reason the PDP terminates and restarts, the PPU can retrieve the necessary policy documents and policy negotiation artifacts from the Policy Repository and quickly re-establish any previously negotiated policies without incurring the full cost of the negotiation process. The retrieval of all the necessary policy negotiation artifacts from the Policy Repository provides evidence of a previously successful negotiated policy. That being the case, the PPU starts the negotiation sequence in the TRY REFRESH state which causes the periodic transmission of Policy Refresh objects. If the remote peer responds with a valid Policy Refresh object, an immediate transition to the NEGOTIATED state occurs and the previously negotiated policy is implemented.

4.12 High Availability

Persistent storage of policy documents and policy negotiation artifacts also allows multiple PDP systems to operate in a high availability cluster. When the active PDP system fails or is taken offline, a standby PDP system can assume the PDP role and retrieve the necessary policy documents from the Policy Repository. For negotiated policies, all policy negotiation artifacts are also stored in the Policy Repository. This allows the PDP to quickly build a new Policy Refresh object with a current validity period and transmit the object to the remote administrative domain.

5. An Example of Negotiable Policies

A specification for negotiable inter-domain security policies is provided in [IDSP]. This specification describes the inter-domain security policies in detail and describes how these policies are negotiated. The specification is quite complex. The motive for compiling the inter-domain security policy specification was to create a requirements specification to assist in the design of a negotiated PPU for the PBNM system. It is not yet certain whether the specified inter-domain security policy would be useful in an operational environment.

This section provides a brief overview of the inter-domain security policy specification. Individuals that require more detained information are directed at the specification document [IDSP]..

5.1 Overview of Inter-Domain Security Policies

The organization security officer compiles inter-domain security policies that describe the conditions whereby the local administrative domain can establish a Virtual Private Network (VPN) with an external organization. The inter-domain security policies do not contain any detailed information about the other party’s infrastructure, and therefore the PDP cannot use these policies directly to configure PEP devices.

5.2 Policy Scopes

Inter-domain security policies are divided into three nested policy scopes. The global scope dictates policy for interactions with all external organizations. The coalition (or group[4]) scope groups together a number of external organizations with a common purpose. The administrative domain (AD) scope identities an external organization using an assigned AD name. An inter-domain security policy contains a single global scope and

[4] A group may define sub-organizations within the larger organization that share similar intra-organizational policies.

one or more group scopes, with the ability for each group scope to include numerous AD scopes.

The inter-domain security policy assigns a priority to each external AD. The PDP system uses the AD priority to resolve conflicts between different ADs. If two ADs include the same network address space as part of their infrastructure, the local AD resolves the conflict using the AD priority. Priorities also prevent the offering of certain services to a lower priority remote AD when the local AD is engaged with a higher priority remote AD. This causes the PDP to generate different Policy Proposal objects for the lower priority AD based on its negotiation state with the higher priority AD.

Policy statements defined in the global scope are automatically inherited by all nested group scopes and policy statements defined in each group scope are automatically inherited by all nested AD scopes. However, an inner scope may refine a policy statement defined in an outer scope. In the context of inter-domain security policies, refinement means to make the policy statement more secure.

5.3 Security Classes

Inter-domain security policies assume the existence of previously defined and well known set of IPsec security parameters called security classes. Details such as encryption algorithms, hash algorithms and Internet Key Exchange (IKE) groups are defined within security classes. The PDP negotiates security classes and not individual IPsec parameters.

5.4 Declarations

The preamble section of each scope contains declarations that describe the local address space that external organizations can access through the VPN, the local name space that external organizations should use to access local network resources, and a list of acceptable security classes that the local domain will accept in establishing the IPSec based VPN. These declarations can be refined within an inner scope to specify less address space, constrained domain name space or fewer acceptable security classes.

5.5 Local and Remote Policy Controls

The preamble section may also include local and remote policy controls. Local policy controls place constraints on the subsequent definition of local service requirements and local service provisions. For local service requirements, they control which local AD network assets can access specific services within a remote AD. For local service provisions, they control which local AD network assets can provide specific services to a remote AD.

Remote policy controls impose constraints on the interactions of a remote AD with a third party AD while a remote AD is engaged with the local AD. The system supports two types of remote policy controls. The first type defines the acceptable services that a remote AD can offer to a third party AD. The second type defines the acceptable security classes that a remote AD can use to engage a third party AD. The local AD trusts the remote AD to implement the local AD’s remote policy controls, but the PBNM system does not currently define an auditing capability to ensure compliance.

5.6 Service Access Rules

The group and AD scopes include a service access rules section that comprises of local service requirements and local service provisions. A local service requirement describes a service that entities in the local AD expect to access in the remote AD. A local service provision describes a service that a server in the local AD expects to provide to entities in the remote AD. When negotiating inter-domain security policies, a local service requirement specified in the local AD must match a local service provision proposed by the remote AD, and a local service provision specified in the local AD must match a local service requirement proposed by the remote AD.

Service access rules contain a criticality attribute that defines the consequence of failing to accept the associated service access rule as part of the negotiation. If the criticality attribute is set to true, the whole policy must be rejected. If the criticality attribute is false, only the associated service access rule is rejected. The criticality attribute allows a policy to be accepted (in whole or in part). Service access rules may also contain time-of-day constraints as well as other service constraints and remote policy controls specific to the service access rule.

5.7 Policy Implementation and Enforcement

In order to enforce inter-domain security policies, the inter-domain security policy PPU requires support for the following types of PEP devices.

Inter-Domain VPN Gateway	A secure gateway device that establishes and maintains VPNs to foreign administrative domains.
Inter-Domain Firewall	A firewall that controls traffic flow to/from foreign administrative domains.
Inter-Domain Route Distribution Router	A router that advertises routes for foreign networks within the local routing domain
Inter-Domain Name Server	A DNS server that provides name resolution within the local administrative domain for the name space associated with foreign administrative domains.

References

[XPATH] http://www.w3.org/TR/xpath

[RFC2748] RFC 2748, " The COPS (Common Open Policy Service) Protocol ", D. Durham, J. Boyle, R. Cohen, S. Herzog,
R. Rajan, A. Sastry, January 2000

[RFC3084] RFC 3084, "COPS Usage for Policy Provisioning", K. Chan, J. Seligson, D. Durham, S. Gai, K. McCloghrie, S. Herzog, F. Reichmeyer,
R.Yavatkar, A. Smith, March 2001

[IDSP] "Discussion Paper - Specification of Inter-Domain Security Policies", Version DRAF T 0.3, NRNS Incorporated, December 2004

[IDPE] "Inter-Domain Policy Editor - System Implementation', Version DRAF T 0.1, NRNS Incorporated, January 2005

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in section 8.)

NRNS Incorporated
4043 Carling Avenue
Ottawa K2K 2A3

2. SECURITY CLASSIFICATION (overall security classification of the document, including special warning terms if applicable)

UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C or U) in parentheses after the title.)

Policy Based Network Management System Design Document (U)

4. AUTHORS (Last name, first name, middle initial)

Spagnolo, J., Cayer D.

5. DATE OF PUBLICATION (month and year of publication of document)

September 2005

6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc.)

18

6b. NO. OF REFS (total cited in document)

5

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Contract Report

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address.)

DRDC Ottawa/NIO Section
3701 Carling Avenue
Ottawa K1A 0Z4

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant)

15BF27

9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written)

W7714-3-800/001/SV

10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.)

10b. OTHER DOCUMENT NOS. (Any other numbers which may be assigned this document either by the originator or by the sponsor)

DRDC Ottawa CR 2005-109

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(x) Unlimited distribution
( ) Distribution limited to defence departments and defence contractors; further distribution only as approved
( ) Distribution limited to defence departments and Canadian defence contractors; further distribution only as approved
( ) Distribution limited to government departments and agencies; further distribution only as approved
( ) Distribution limited to defence departments; further distribution only as approved
( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.)

Full Unlimited

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

(U) This report presents an architectural design and concept of operation for a policy-based network management system. Policy-based network management (PBNM) systems provide an automated means to configure and administer Policy Enforcement Point (PEP) devices such as virtual private network (VPN) gateways, firewalls and routers. The Policy Decision Point (PDP) takes high level policies as input and produces lower level PEP-specific policies as output. The PBNM system can process different types of policies. When evaluating policies, the PDP must identify and resolve conflicts within competing policies as well as take into consideration external factors such as the time-of-day and the current threat level.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Common Open Policy Service (COPS) protocol
Inter-domain security policy
Network Managament
Policy
Policy-based network management
Policy Decision Point (PDP)
Policy Editor
Policy enforcement
Policy Enforcement Point (PEP)
Policy negotiation
Policy Negotiation Proxy (PNP)
Policy object
Polic Processing Unit (PPU)
Policy repository
Security policy
XML policy

Defence R&D Canada

Canada's leader in Defence
and National Security
Science and Technology

R & D pour la défense Canada

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale

DEFENCE DÉFENSE

www.drdc-rddc.gc.ca