Modelling Distributed Network Security in a Petri Net and Agent-based Approach

The Herold Project
www.herold-security.de

MATES 2010
September 28th

Simon Adameit, Tobias Betz, Lawrence Cabac, Florian Hars, Marcin Hewelt, Michael Köhler-Bußmeier, Daniel Moldt, Dimitri Popov, José Quenum, Axel Theilmann, **Thomas Wagner**, Timo Warns, Lars Wüstenberg

Overview

Introduction

- Networks omnipresent today
- Data and services accessible via a network
- Have to be protected from...
  - unauthorised access
  - (malicious) tampering
  - ...
- Need for network security

Introduction

- Traditional perimeter approaches problematic
- Cell-based approaches lessen problems and
- Also closer to modern scenarios
- Herold project aims to provide **distributed network security management**
- This presentation features early results

Overview

Herold Overview

SPONSORED BY THE

Federal Ministry of Education and Research

- 2 Year funded research project
- Funded by the BMBF (Grant No. 01BS0901)
- Cooperation between:
  - **PRESENSE Technologies GmbH**
    www.pre-sense.de
  - **Theoretical Foundations of Computer Science Group (TGI), University of Hamburg**
    www.informatik.uni-hamburg.de/TGI/
  - **N@Work**
    www.work.de

n@work INTERNET INFORMATIONSSYSTEME GmbH

Herold Overview

- Distributed system for a novel agent-oriented approach to distributed network security
- Core: Efficient and secure configuration of network security components (NSCs)
- Concurrent, cooperative design

Herold Overview

- Aims to provide activities associated with network security management
  - Define abstract security goals
  - Define a concrete security policy
  - Choose how and where to enforce the policy
  - Monitor and analyse enforcement
  - ...

Herold Summary

- Main Concepts:
    - Hierarchy of policies
    - Cooperative design of policies
    - Localisation
    - Cooperative enforcement by NSCs

Herold Summary

Problems:

- Distributed environment
- Concurrent behaviour
- Complex dynamics
- Security application requirements

Solutions:

➡ Agents

➡ Petri nets

➡ PAOSE

➡ Herold

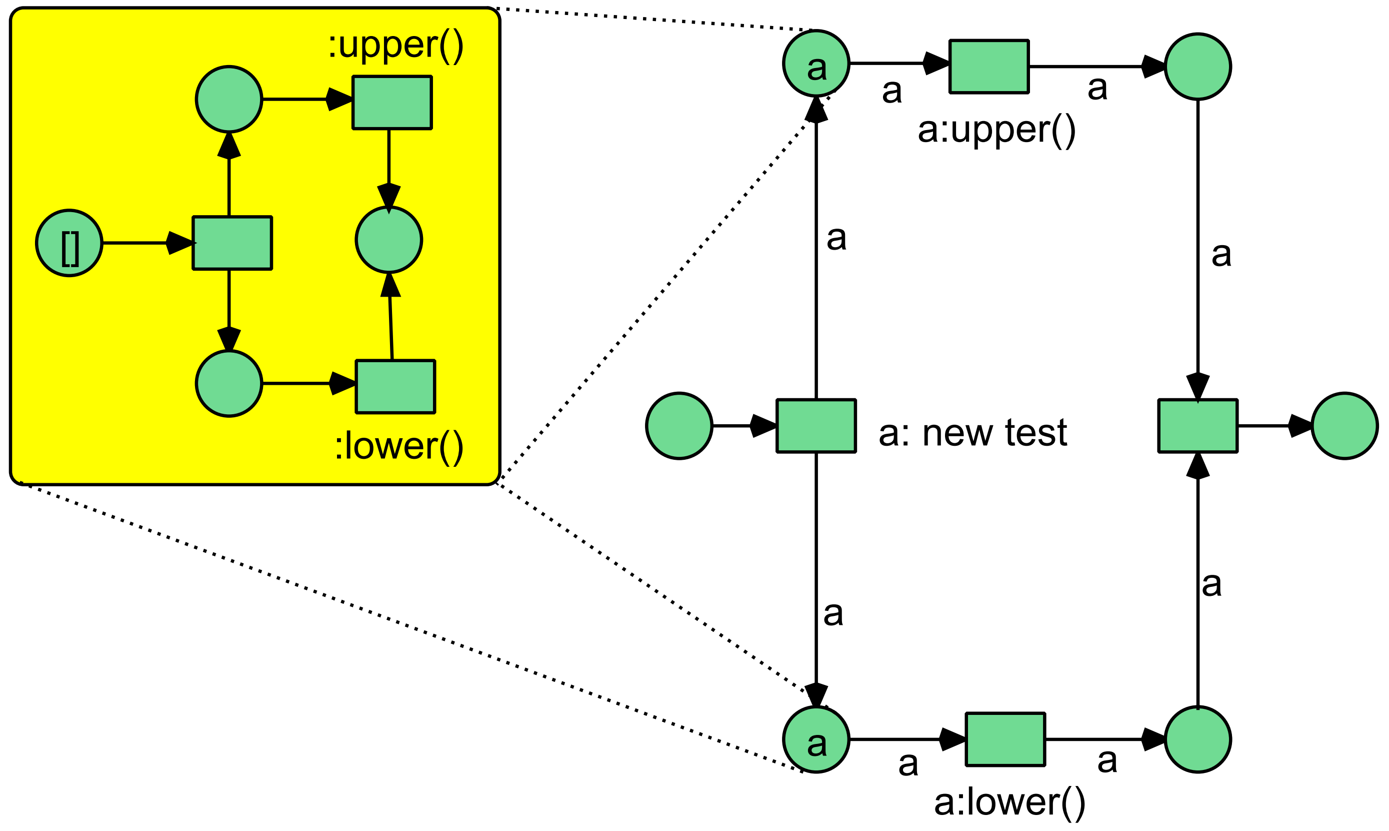
Overview

Modelling Background

- Reference Nets
- RENEW
- MULAN / CAPA
- PAOSE

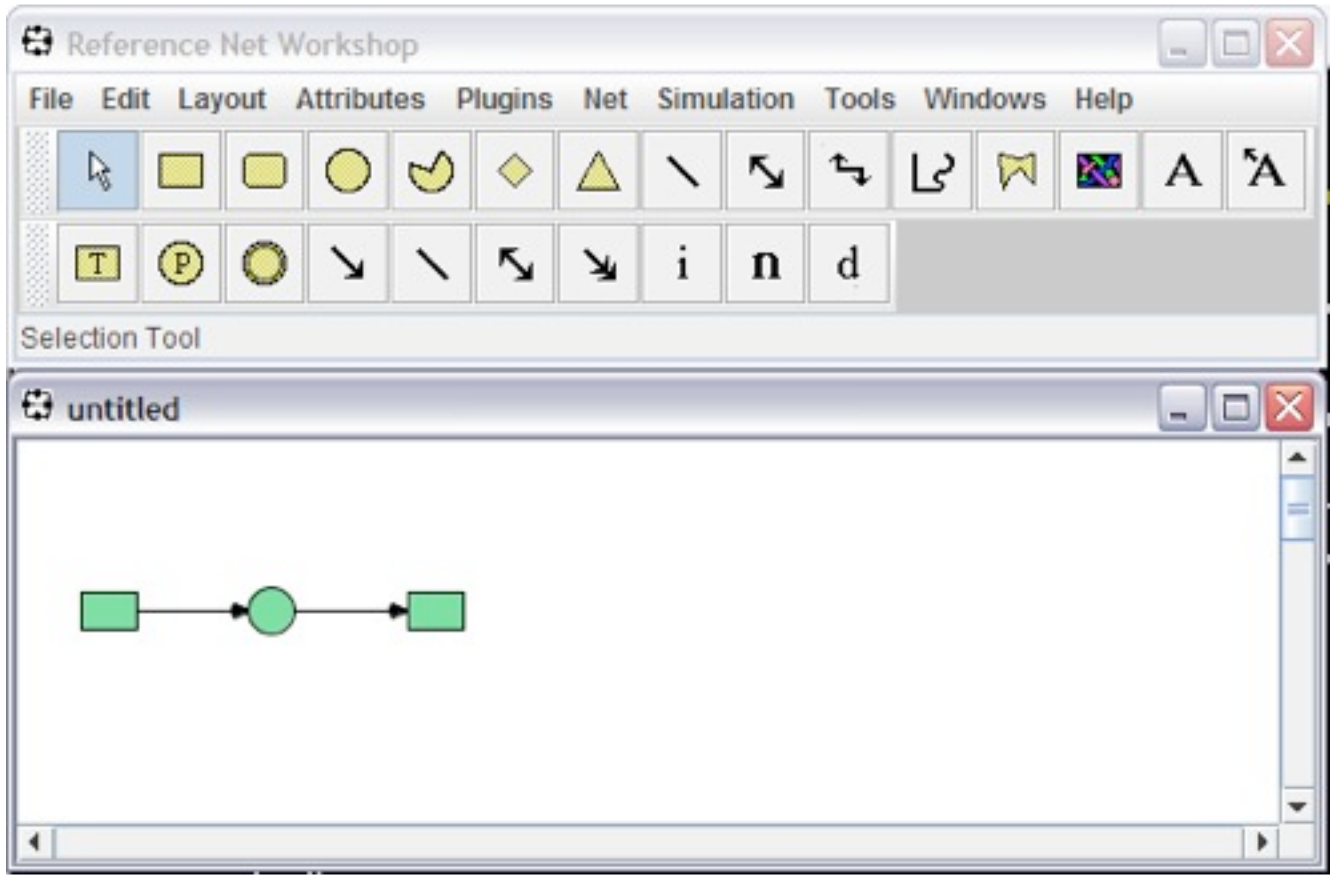
Reference Nets

- High-level Petri net formalism
- Tokens are references to other objects
- Nets-within-nets paradigm
- Synchronous channels are used for communication between and in nets



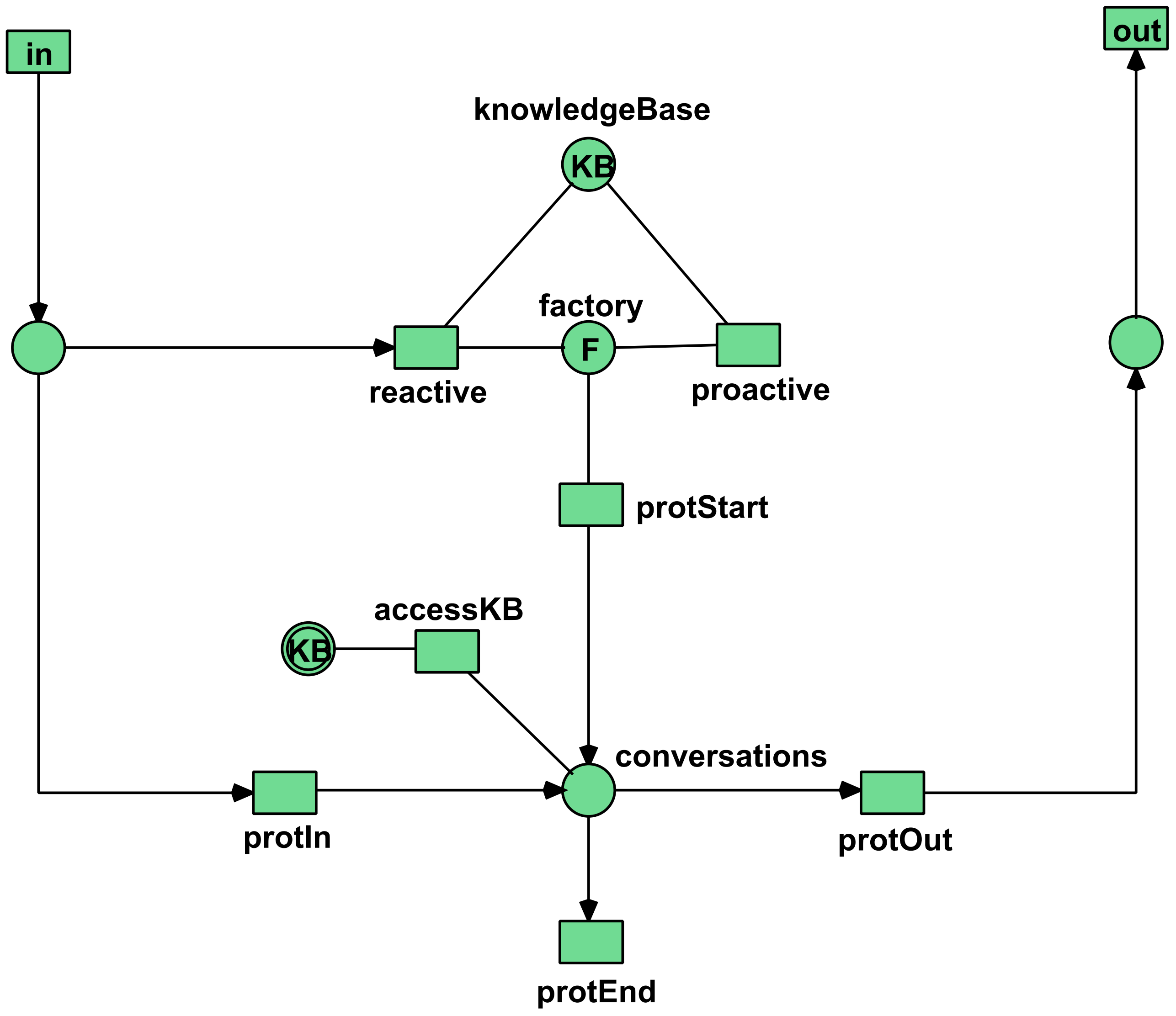
Renew

- Editor and simulator for different net formalisms
- Especially designed for reference nets
- Serves as development and runtime environment
- Freely available at www.renew.de


Reference Net Workshop
File Edit Layout Attributes Plugins Net Simulation Tools Windows Help
i n d
Selection Tool
untitled

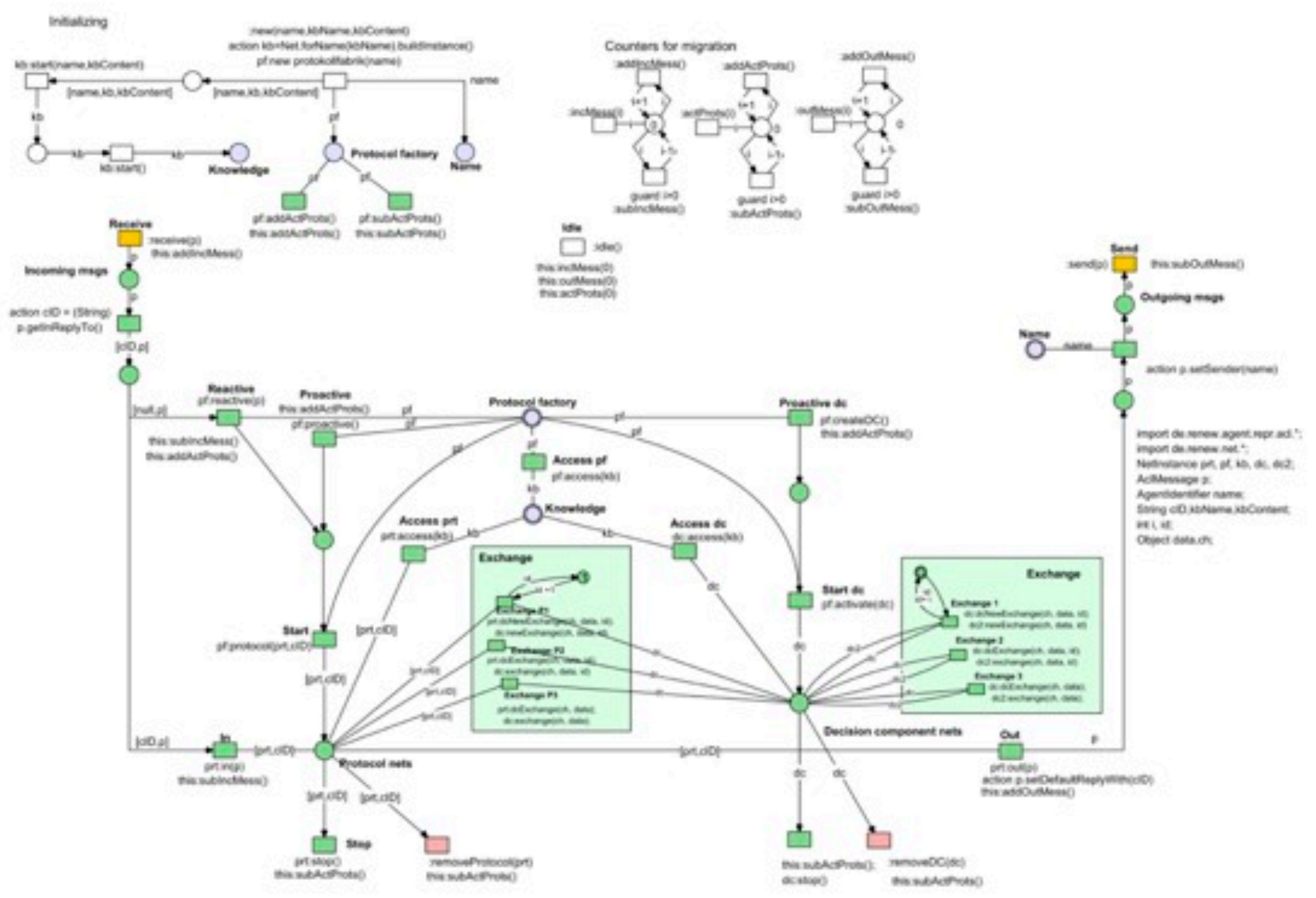
MULAN

- **MUL**ti **A**gent **N**ets
- Complete agent architecture modelled with reference nets
- Executable in RENEW as a conceptual framework



CAPA

- **C**oncurrent **A**gent **P**latform **A**rchitecture
- FIPA compliant extension of MULAN
- Replaces upper layers of MULAN to allow deployment in real-life networks



PAOSE

- **P**etri net-based **A**gent and **O**rganisation-oriented **S**oftware **E**ngineering
- Especially suited for developing systems with MULAN and CAPA
- Key Aspects:
  - Rapid prototyping
  - Three-dimensional modelling (Actors, Interactions, Ontology)
  - MAS as metaphor for development team
  - Tool support (RENEW) for different stages
- Visit www.paose.net for further information

Overview

General Assumptions

- Needed:
    - Simple model
    - Represents all relevant aspects of Herold project
    - Relatively “easy” to understand, present and handle
    - Iteratively rising in complexity

➡ A conceptual system for which an implementation is iteratively enhanced

General Assumptions

- Conceptual view:
  - One “stepping stone” that covers important aspects
  - Theoretical view
- Implementation
  - In the paper: Simpler “model zero”
  - In this presentation: More advanced model

Conceptual View

- Cell-based approach to network security
- Hierarchy of policies
- Technical actors are represented as agents
- Especially NSCs regarded as nodes of distributed systems

Network Model

- Fully connected network topology
- Unique addresses
- Focus of this network model are the NSCs
- Grouping of network nodes supported
- Limitation: Certain NSCs cannot be covered in this model due to implicit network topology

Policy Model

- Users share single global (total) policy
- Policy consists of an ordered set of rules
- Rules consist of
  - Source address and port
  - Target address and port
  - “allow” or “deny” for traffic between source and target
- Implicit rule for every non explicit one
- Rules over groups allow concise policy definitions

Use Cases

- View current policy
- Add/delete/modify/move rule within policy
- View status information
- View current NSCs
- Add/delete/modify NSCs
- View groups
- Add/delete/modify/rename group

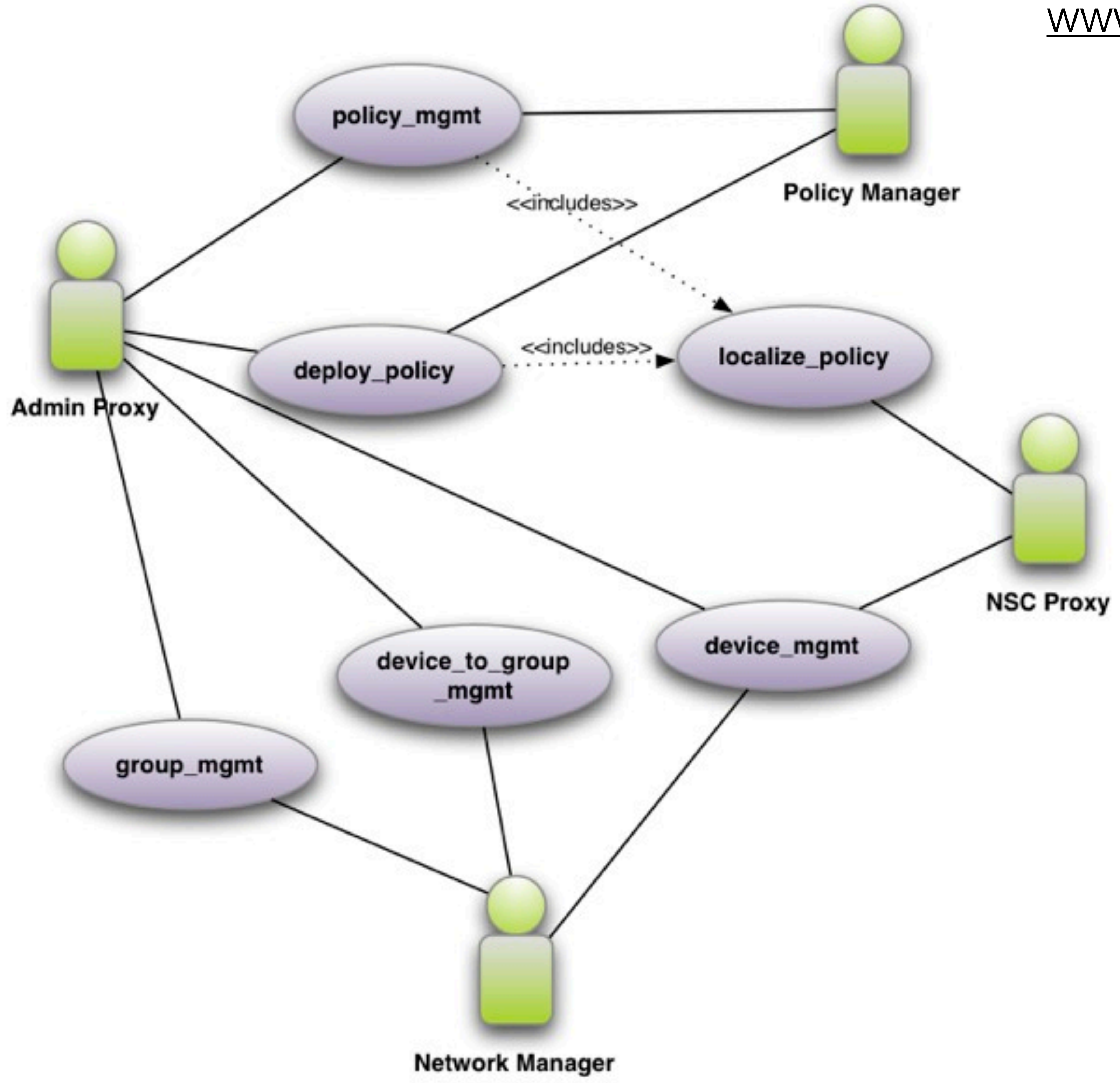
Overview

Implementation

- Working prototype realising Herold functionality
- Implemented using MULAN/CAPA agents
- RENEW serves as runtime environment
- Further along than “model zero” presented in paper

Agents

- AdminProxy
- Policy Manager
- Network Manager
- NSC Proxy


policy_mgmt
Policy Manager
<<includes>>
Admin Proxy
deploy_policy
<<includes>>
localize_policy
NSC Proxy
device_mgmt
device_to_group _mgmt
group_mgmt
Network Manager

Use Cases

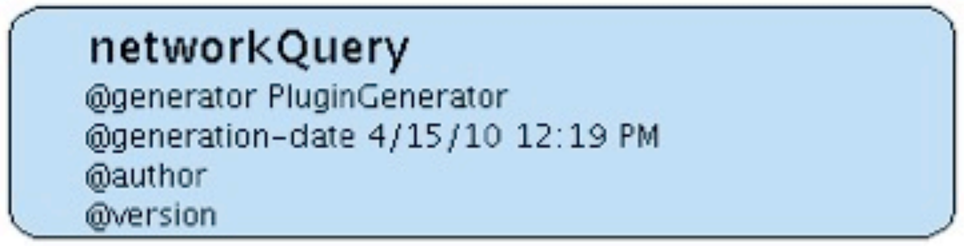
Interactions

- Interactions in MULAN/CAPA usually provide the largest part of the functionality
- In this scenario many similar interactions occur (e.g. add/delete/modify rule)
- Two options:
  - Model each interactions separately
  - Model a few interactions catering to many uses

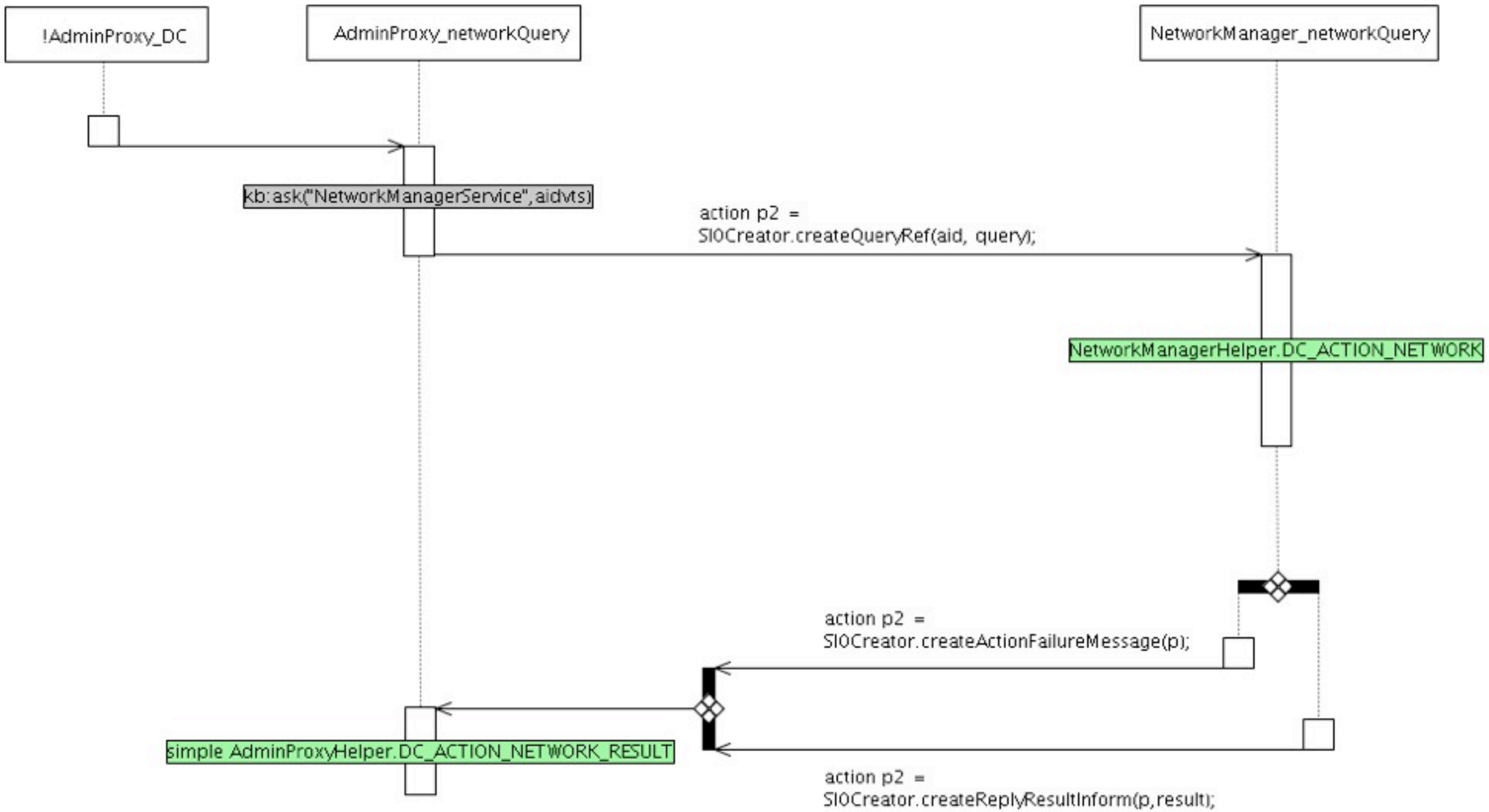
Interactions

- deployPolicyRequest
- localiseQuery
- localiseRequest
- networkQuery
- networkRequest
- policyQuery
- policyRequest
- prelocaliseQuery
- startAgent
- stopAgent

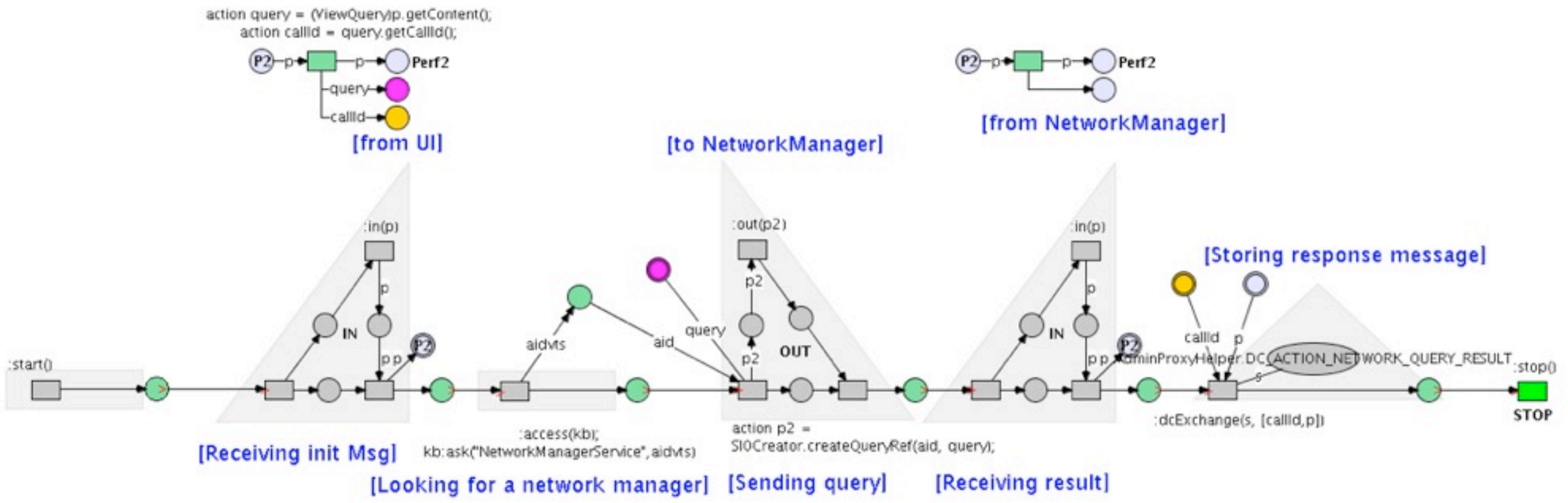
Agent Interaction Diagram



networkQuery
@generator PluginGenerator
@generation-date 4/15/10 12:19 PM
@author
@version


!AdminProxy_DC
AdminProxy_networkQuery
NetworkManager_networkQuery
kb:ask("NetworkManagerService",aidvts)
action p2 = SIOCreator.createQueryRef(aid, query);
NetworkManagerHelper.DC_ACTION_NETWORK
action p2 = SIOCreator.createActionFailureMessage(p);
simple AdminProxyHelper.DC_ACTION_NETWORK_RESULT
action p2 = SIOCreator.createReplyResultInform(p,result);

Protocol Net - AdminProxy

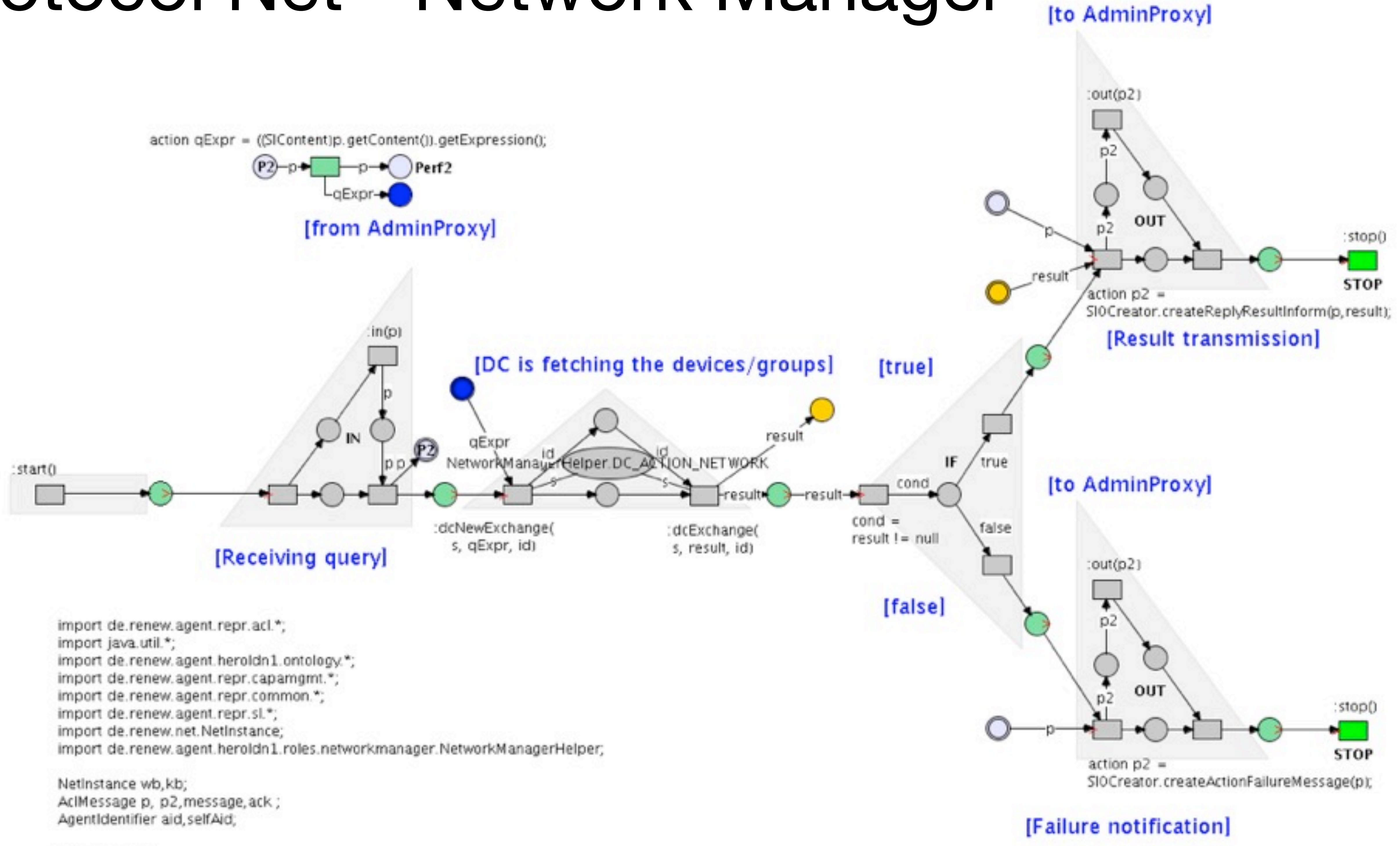

action query = (ViewQuery)p.getContent();
action callId = query.getCallId();
P2
p
p
Perf2
query
callId
[from UI]
[to NetworkManager]
P2
p
p
Perf2
[from NetworkManager]
:in(p)
p
IN
p p
P2
:start()
aidvts
aid
query
:out(p2)
p2
p2
OUT
:in(p)
p
IN
p p
P2
[Storing response message]
callId
p
AdminProxyHelper.DC_ACTION_NETWORK_QUERY_RESULT
s
:stop()
STOP
:dcExchange(s, [callId,p])
:access(kb);
kb:ask("NetworkManagerService",aidvts)
action p2 =
SlOCreator.createQueryRef(aid, query);
[Receiving init Msg]
[Looking for a network manager]
[Sending query]
[Receiving result]

```
import de.renew.agent.repr.acl.*;
import java.util.*;
import de.renew.agent.heroldn1.ontology.*;
import de.renew.agent.repr.capamgmt.*;
import de.renew.agent.repr.common.*;
import de.renew.agent.repr.sl.*;
import de.renew.net.NetInstance;
import de.renew.agent.heroldn1.roles.adminproxy.AdminProxyHelper;

NetInstance wb,kb;
AclMessage p, p2,message,ack ;
AgentIdentifier aid,selfAid;

Boolean bool;
boolean cond;
ViewQuery o, query;
Object[] os;
String s;
Vector v;
int y,id;
Iterator  it;
VTSet aidvts;
int callId;
```

Protocol Net - Network Manager


[to AdminProxy]
:out(p2)
p2
OUT
p2
p
result
:stop()
STOP
action p2 = SIOCreator.createReplyResultInform(p,result);
[Result transmission]
action qExpr = ((SIContent)p.getContent()).getExpression();
P2
p
p
qExpr
Perf2
[from AdminProxy]
:in(p)
p
IN
p p
P2
:start()
[Receiving query]
[DC is fetching the devices/groups]
qExpr
id
id
NetworkManagerHelper.DC_ACTION_NETWORK
s
s
result
result
result
:dcNewExchange(s, qExpr, id)
:dcExchange(s, result, id)
[true]
IF
true
cond
cond = result != null
false
[false]
[to AdminProxy]
:out(p2)
p2
OUT
p2
p
:stop()
STOP
action p2 = SIOCreator.createActionFailureMessage(p);
[Failure notification]

```
import de.renew.agent.repr.acl.*;
import java.util.*;
import de.renew.agent.heroldn1.ontology.*;
import de.renew.agent.repr.capamgmt.*;
import de.renew.agent.repr.common.*;
import de.renew.agent.repr.sl.*;
import de.renew.net.NetInstance;
import de.renew.agent.heroldn1.roles.networkmanager.NetworkManagerHelper;

NetInstance wb,kb;
AclMessage p, p2,message,ack ;
AgentIdentifier aid,selfAid;

Boolean bool;
boolean cond;
Object o,qExpr,result;
Object[] os;
String s;
Vector v;
int y,id;
Iterator  it;
VTSet aidvts;
```

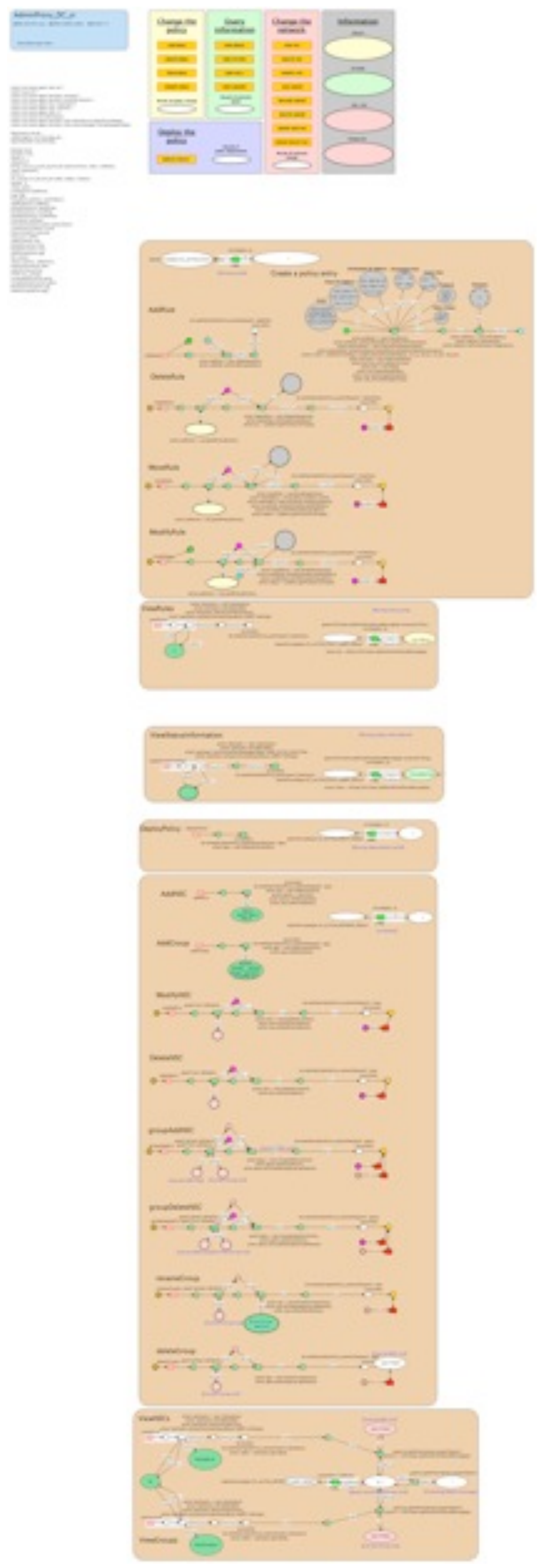
Decision Components

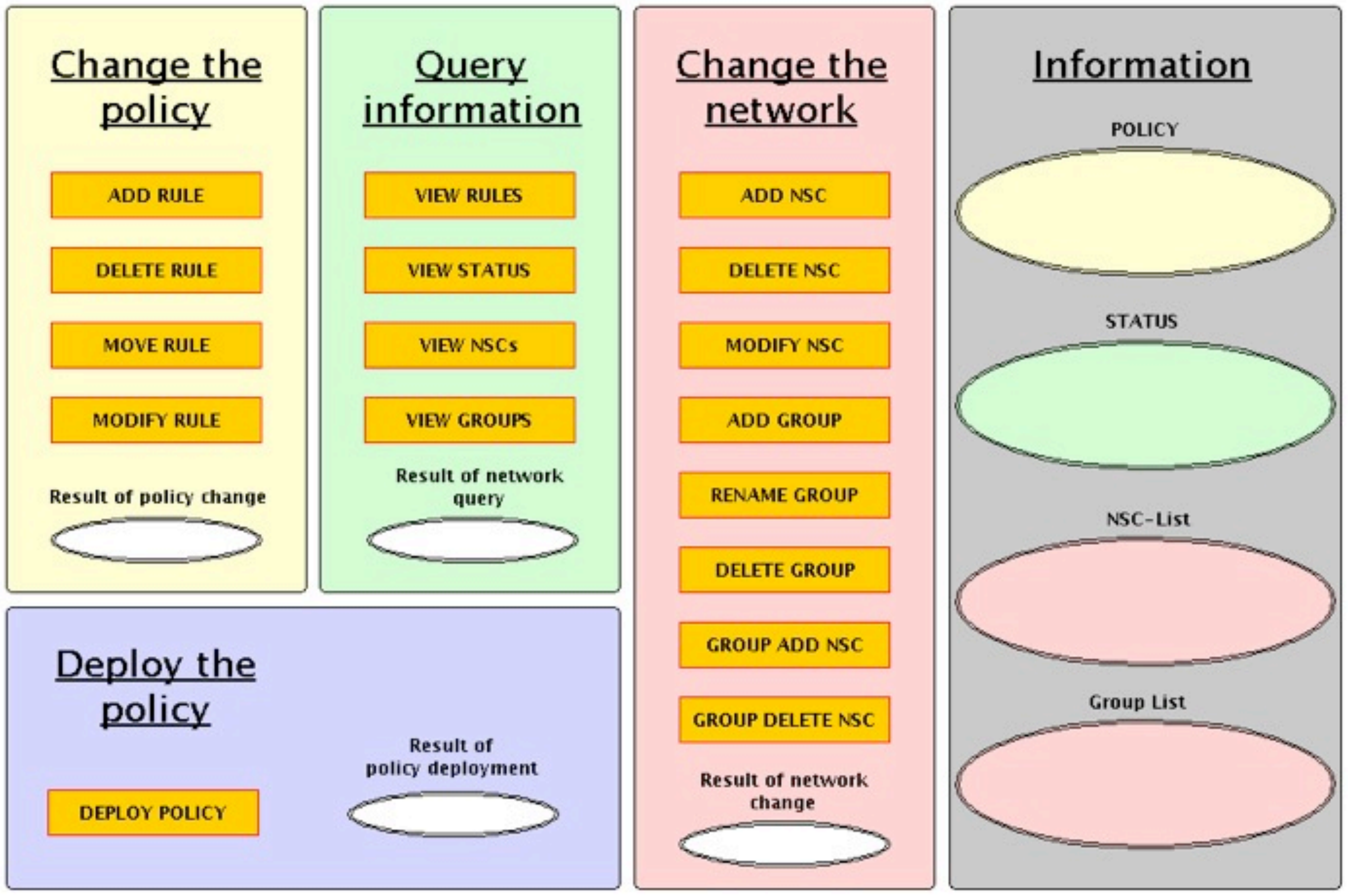
- Use of “template” interactions forces functionality into decision components (DCs)
- DCs can be viewed as special, constantly running protocol nets
- Every agent (in this context) possesses a number of DCs

Decision Components

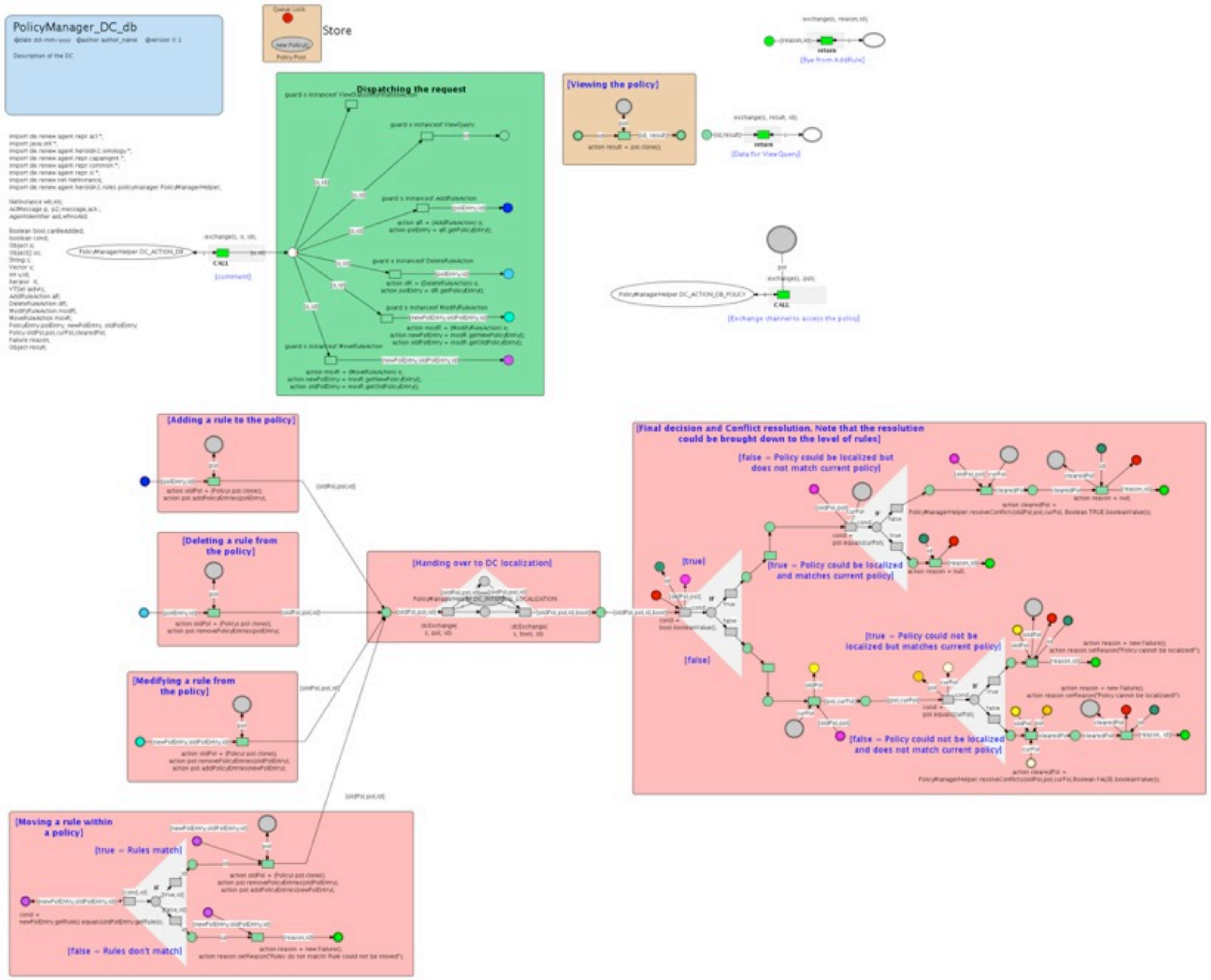
- **AdminProxy**
  - User Interface DC
- **Policy Manager**
  - (Top Level)
  - Database
  - Localisation
- **Network Manager**
  - (Top Level)
  - Database
  - Localisation
- **NSC Proxy**
  - (Top Level)
  - Localisation

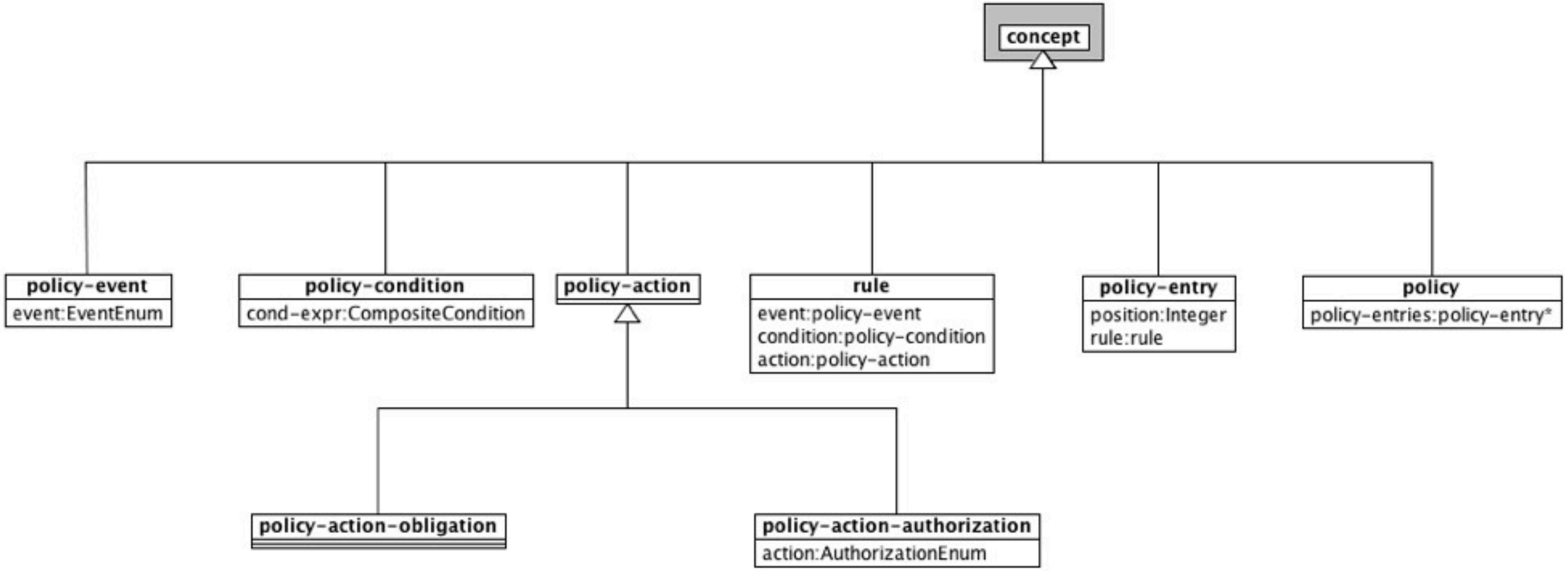
User Interface DC





Policy Manager Database DC


PolicyManager_DC_db
Store
Dispatching the request
[Viewing the policy]
[Adding a rule to the policy]
[Deleting a rule from the policy]
[Modifying a rule from the policy]
[Moving a rule within a policy]
[true – Rules match]
[false – Rules don't match]
[Handing over to DC localization]
[Final decision and Conflict resolution. Note that the resolution could be brought down to the level of rules]
[false – Policy could be localized but does not match current policy]
[true – Policy could be localized and matches current policy]
[true – Policy could not be localized but matches current policy]
[false – Policy could not be localized and does not match current policy]
[true]
[false]
[comment]
[Exchange channel to access the policy]
[Data for ViewQuery]
[Bye from AddRule]
CALL
return

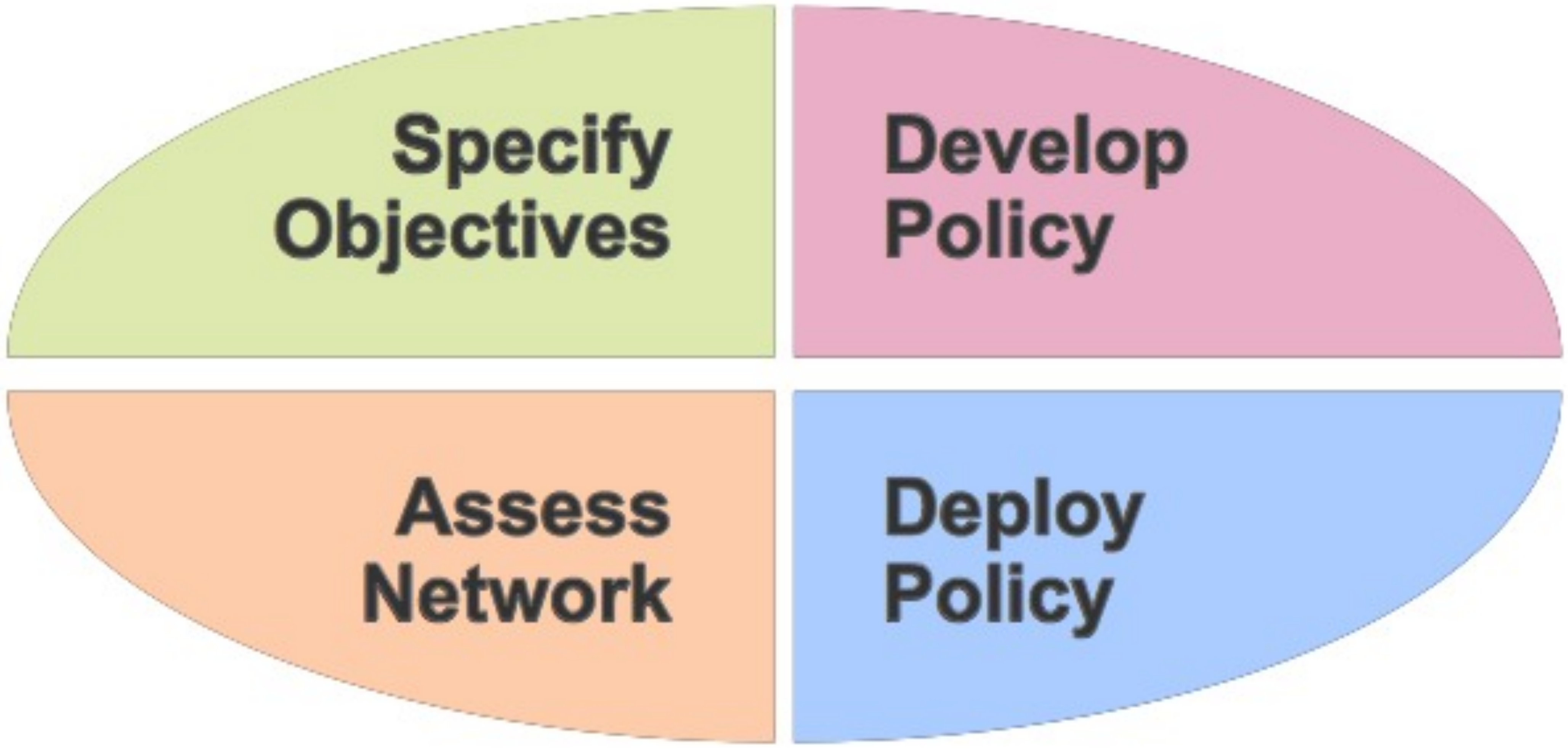


Ontology (partial)

Overview

Outlook - Data Models

- Extensions to network and policy model:
  - Explicit and complex network topologies
  - Abstract security objectives and best practices
  - Partial policies, policy templates, “policy pool”
  - ...



Herold Cycle

Further aspects

- Localisation
- Verification
- Distribution of Herold
- Versatility
- ...

The End

Thank You for Your attention

Questions? :)

www.herold-security.de