

Controle d'accès pour les grandes infrastructures critiques. Application au réseau d'énergie électrique.

Amine Baïna

▶ **To cite this version:**

Amine Baïna. Controle d'accès pour les grandes infrastructures critiques. Application au réseau d'énergie électrique.. Informatique [cs]. INSA de Toulouse, 2009. Français. NNT : . tel-00432841

HAL Id: tel-00432841
https://theses.hal.science/tel-00432841

Submitted on 17 Nov 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par *l'Institut National des Sciences Appliquées de Toulouse*
Discipline ou spécialité : *Systèmes Informatiques Critiques*

Présentée et soutenue par *Amine BAINA*
Le *29 septembre 2009*

Titre : *Contrôle d'Accès pour les Grandes Infrastructures Critiques : Application au réseau d'énergie électrique*

JURY

Nora CUPPENS-BOULAHIA, Rapporteur, Enseignant-Chercheur, Telecom Bretagne
Danielle BOULANGER, Rapporteur, Professeur des Universités, Université de Lyon III
Yves DESWARTE, Directeur de Thèse, Directeur de Recherche, LAAS-CNRS
Anas ABOU EL KALAM, Encadrant, Maître de Conférences, ENSEEIHT
Mohamed KAÂNICHE, Président, Directeur de Recherche, LAAS-CNRS
Paulo VERISSIMO, Examinateur, Professeur des Universités, Université de Lisbonne
Pascal SITBON, Examinateur, Ingénieur Recherche & Développement, EDF

Ecole doctorale : *Ecole Doctorale Systèmes*
Unité de recherche : *LAAS-CNRS*
Directeur(s) de Thèse : *Yves DESWARTE*
Rapporteurs : *Nora CUPPENS-BOULAHIA, Danielle BOULANGER*

Résumé

En raison de ses vulnérabilités physiques et logiques, une infrastructure critique (IC) peut subir des défaillances, et en raison des interdépendances entre IC, de simples défaillances peuvent avoir des conséquences dramatiques sur l'ensemble de l'infrastructure.

Dans notre travail, nous nous concentrons principalement sur les systèmes d'information et de communication (l'IIC : infrastructure d'information critique) dédiés au réseau d'énergie électrique. Nous proposons une nouvelle approche pour répondre aux problèmes de sécurité que rencontre une IIC, plus particulièrement, ceux liés au contrôle d'accès et à la collaboration. Le but est d'offrir à chaque organisation faisant partie de l'IIC la possibilité de collaborer avec les autres, tout en maintenant un contrôle sur ses données et sa politique de sécurité internes.

Nous avons modélisé, et développé PolyOrBAC, une plateforme de contrôle d'accès collaboratif, basée sur le modèle de contrôle d'accès OrBAC et sur la technologie des Services Web, cette plateforme est applicable dans le contexte d'une infrastructure critique en général, et plus particulièrement dans le cadre d'un réseau électrique.

MOTS CLES : Infrastructures Critiques, Système d'information, Infrastructure d'Information Critique, Résilience, Sécurité, Contrôle d'Accès, OrBAC, PolyOrBAC, Collaboration, Interopérabilité, Services Web, CRUTIAL, Réseau d'énergie électrique.

Abstract

Because of its physical and logical vulnerabilities, critical infrastructure (CI) may suffer failures, and because of the interdependencies between CIs, simple failures can have dramatic consequences on the entire infrastructure.

In our work, we mainly focus on information systems and communications (CII: Critical Information Infrastructure) dedicated to the electrical power grid. We proposed a new approach to address security problems faced by an IIC, particularly those related to access control and collaboration. The goal of this study is to provide each organization belonging to the IIC the opportunity to collaborate with others while maintaining control over its data and its internal security policy.

We modeled and developed PolyOrBAC, a platform for collaborative access control, based on the access control model OrBAC and on the Web Services technology, this platform is applicable in the context of a critical infrastructure in general, and more particularly to an electrical power grid.

KEYWORDS: Critical Infrastructure, Information System, Critical Information Infrastructure, Resilience, Security, Access Control, OrBAC, PolyOrBAC, Collaboration, Interoperability, Web Services, CRUTIAL, Electric Power Grid.

Remerciements

Les travaux présentés dans ce mémoire ont été réalisés au Laboratoire d'Analyse et d'Architecture des Systèmes au sein de l'équipe Tolérance aux fautes et Sûreté de Fonctionnement informatique. Pour la qualité de vie que j'ai pu apprécier durant ces quatre années, je remercie Messieurs Malik Ghallab et Raja Chatila, les directeurs successifs du laboratoire durant cette période, pour m'avoir accueilli dans leur laboratoire. Pour des raisons similaires, je tiens à remercier Monsieur Jean Arlat et Madame Karama Kanoun, qui ont dirigé le groupe TSF, pour m'avoir permis de m'intégrer dans l'équipe et ses activités de recherche.

Vu le chemin parcouru et les obstacles surmontés, je suis particulièrement reconnaissant envers Messieurs Yves DESWARTE et Anas ABOU EL KALAM, respectivement Directeur de Recherche au LAAS-CNRS, et maître de conférence à l'ENSEEIHT, pour leurs directives, leur encadrement et leur aide théorique et technique approfondie durant mes quatre ans de thèse. J'ai énormément apprécié leur sérieux, leur serviabilité, disponibilité, et leur soutien tout au long de mes travaux de recherche et de la phase de rédaction.

J'ai été honoré que Monsieur Mohamed KAÂNICHE, Directeur de Recherche au LAAS-CNRS Toulouse, ait accepté de présider mon jury de thèse. J'en profite pour remercier également :

- Nora CUPPENS-BOULAHIA, Enseignant-Chercheur, à Telecom Bretagne, Rennes
- Danielle BOULANGER, Professeur à l'Université de Lyon III, IAE, Lyon
- Yves DESWARTE, Directeur de Recherche au LAAS-CNRS, Toulouse
- Anas ABOU EL KALAM, Maître de Conférences à l'ENSEEIHT Toulouse, IRIT
- Paulo VERISSIMO, Professeur à l'Université de Lisbonne, FCUL, Lisbonne
- Pascal SITBON, Ingénieur Recherche et Développement, EDF, Paris

pour l'honneur qu'ils m'ont fait en acceptant de participer à mon jury de thèse.

Je suis particulièrement reconnaissant envers Madame Nora CUPPENS-BOULAHIA et Madame Danielle BOULANGER pour avoir accepté la charge de rapporteur. Mes remerciements vont également à tous les membres du projet CRUTIAL au Portugal, Italie, Belgique pour leur sympathie et leur convivialité.

Comme je l'ai souligné au début, le groupe de recherche TSF est une grande famille, je tiens donc à remercier l'ensemble de ses membres pour l'ambiance qu'ils ont su instaurer. Je n'oublie pas non plus l'ensemble des personnes travaillant dans les différents services au LAAS, tout particulièrement Monsieur Yves CROUZET, et Madame Gina BRIAND.

Je tiens à remercier également les doctorants qui ont partagés comme moi l'épreuve de force que représente une thèse (Etienne, Youssef, Ludo, Benji, Gad, Oss, Zaghlul, Ahmed, Manel, Bob, Nam, Géraldine, Masri, Minh, Dam) et tous ceux que j'oublie.

Pour finir, je tiens à remercier ma famille, mes frères et sœurs, pour leur soutien sincère et leurs encouragements continus, merci à tous. Ma douce Doha, merci pour ta patience, ton soutien et ton support qui a été essentiel, merci pour les longues conversations téléphoniques qui remontent le moral et redonnent le sourire.

Enfin, je tiens à dédier tout naturellement ce mémoire (et tous les efforts qui ont été dépensé pour) à mes parents Aloustad Abdelkhalek BAINA, et Alhajja Naïma BENALI.

Table des matières

RESUME 3

ABSTRACT 5

REMERCIEMENTS 7

TABLE DES MATIERES 11

TABLE DES ILLUSTRATIONS 15

TABLE DES TABLEAUX 17

INTRODUCTION GENERALE ET PROBLEMATIQUE 19

CHAPITRE 1. SECURITE DES SYSTEMES D'INFORMATION DANS LES INFRASTRUCTURES CRITIQUES 25

1.1 LES INFRASTRUCTURES CRITIQUES 26
1.1.1 Classification des infrastructures critiques 26
1.1.2 Exemples d'infrastructures critiques 26
1.1.3 Interdépendance des infrastructures critiques 27
1.1.4 Vulnérabilités et menaces dans les infrastructures critiques 31
1.2 DEFINITION ET ENJEUX D'UNE INFRASTRUCTURE D'INFORMATION CRITIQUE 33
1.2.1 Besoins et exigences de sécurité dans les infrastructures critiques 33
1.2.2 Interdépendance des infrastructures d'information critiques 34
1.2.3 Vulnérabilités et menaces dans les infrastructures d'information critiques 35
1.3 ÉTUDES ET TRAVAUX EXISTANTS 37
1.3.1 IRRIIS – Integrated Risk Reduction of Information-based Infrastructure Systems 37
1.3.2 TCIP – Trustworthy Cyber Infrastructure for the Power Grid 38
1.3.3 CRUTIAL – CRitical UTility InfrastructurAL resilience 39
1.4 CONCLUSION DU CHAPITRE 1 41

CHAPITRE 2. MODELES ET POLITIQUES DE SECURITE 43

2.1 POLITIQUES ET MODELES DE SECURITE 44
2.2 MODELES DE CONTROLE D'ACCES TRADITIONNELS 45
2.2.1 Modèle de contrôle d'accès basé sur les rôles (RBAC) 45
4.2.1.1 Avantages du modèle RBAC 46
2.2.1.1 Inconvénients du modèle RBAC 47
2.2.2 Modèle de contrôle d'accès basé organisation (OrBAC) 47
2.2.2.1 Concept de rôle et relation Habilite () 48
2.2.2.2 Concept de vue et relation Utilise() 49
2.2.2.3 Concept d'activité et relation Considère() 49
2.2.2.4 Concept de contexte et relation Définit() 50
2.2.2.5 Expression de politiques de sécurité dans le modèle OrBAC 51
2.3 MODELES DE CONTROLE D'ACCES POUR LA COLLABORATION 53
2.3.1 Gestion centralisée de la sécurité 54

2.3.2 Gestion décentralisée de la sécurité 55
2.3.3 Le modèle Multi-OrBAC 56
2.3.3.1 Règles de sécurité Multi-OrBAC 57
2.3.3.2 Avantages et inconvénients de Multi-OrBAC 58
2.3.4 Modèle d'organisation virtuelle 59
2.3.4.1 Modèle de sécurité pour les organisations virtuelles 60
2.3.4.2 Avantages et inconvénients du modèle d'organisation virtuelle 62
2.3.5 Le modèle O2O (Organisation 2 Organisation) 62
2.3.5.1 Présentation de la notion de VPO 62
2.3.5.2 Dépendance des droits d'accès avec les rôles 63
2.3.5.3 Gestion des utilisateurs et rôles 64
2.3.5.4 Avantages et inconvénients du modèle O2O 64
2.4 CONCLUSION DU CHAPITRE 2 65

CHAPITRE 3. POLYORBAC : SCHEMA-CADRE DE SECURITE POUR LA COLLABORATION 67

3.1 EXTENSION D'ORBAC AUX SYSTEMES COLLABORATIFS 68
3.2 COLLABORATION ET INTEROPERABILITE AVEC LES SERVICES WEB 69
3.2.1 Architecture Orientée Services (SOA) 69
3.2.2 Normes des services Web 70
3.3 EXTENSIONS DU MODELE ORBAC POUR LES SERVICES WEB 71
3.3.1 Politique globale et politiques locales 71
3.3.2 Images de services Web et utilisateurs virtuels 74
3.3.3 Limites des services Web et nécessité de vérification à l'exécution 75
3.4 EXPRESSION ET VERIFICATION DES INTERACTIONS PAR SERVICES WEB 76
3.4.1 Contrats et politiques-contrats 76
3.4.2 Expression des politiques-contrats sous forme d'automates temporisés 77
3.4.2.1 Expression des permissions 77
3.4.2.2 Expression des interdictions 78
3.4.2.3 Expression des obligations 78
3.4.2.4 Identification des situations de conflits 79
3.4.3 Vérification des politiques-contrats 80
3.5 RECAPITULATION DU FONCTIONNEMENT GENERAL DE POLYORBAC 81
3.5.1 Création et publication d'un service Web par un prestataire 81
3.5.2 Négociation et signature du contrat 81
3.5.3 Invocation et exécution du service Web 81
3.6 CONCLUSION DU CHAPITRE 3 83

CHAPITRE 4. ÉTUDE DE CAS ET APPLICATION DE POLYORBAC 85

4.1 INFRASTRUCTURE DE PRODUCTION, TRANSPORT, ET DISTRIBUTION D'ENERGIE ELECTRIQUE 86
4.2 DESCRIPTION DU SCENARIO DE DELESTAGE 88
4.2.1 Architecture de l'infrastructure 88
4.2.2 Déroulement du scénario de délestage 90
4.3 ARCHITECTURE DE L'INFRASTRUCTURE D'INFORMATION CRITIQUE 93
4.4 VISION POLYORBAC DU SCENARIO DE DELESTAGE 94
4.4.1 Définition des services Web 94
4.4.2 Contrôle d'accès et vérification pour SW1-demande_d'armement 96
4.4.2.1 Règles OrBAC pour la demande d'armement par le TS CC 96
4.4.2.2 Automate de SW1-demande_d'armement au niveau du TS CC 97
4.4.2.3 Règles OrBAC de SW1-demande_d'armement dans la politique du DS CC 98
4.4.2.4 Automate de SW1-demande_d'armement dans le CIS du DS CC 98
4.4.3 Contrôle d'accès et vérification pour SW2-ordre_d'armement 100
4.4.3.1 Automate de SW2-ordre_d'armement du côté DS CC 100
4.4.3.2 Règle OrBAC de l'armement dans la politique du DS SS 101
4.4.3.3 Automate de SW2-ordre_d'armement du côté du DS SS 101
4.5 CONCLUSIONS DU CHAPITRE 4 103

CHAPITRE 5. MISE EN ŒUVRE ET IMPLEMENTATION DU SCHEMA-CADRE POLYORBAC105

5.1 OBJECTIF DE LA MISE EN ŒUVRE 106
5.1.1 Présentation du scénario de l'expérimentation 106
5.1.2 Besoins et contraintes du prototype 107
5.2 MISE EN PLACE DE L'EXPERIMENTATION 108
5.2.1 Description de la plateforme d'émulation 109
5.2.2 Définition des composants de base liés à PolyOrBAC 109
5.2.3 Présentation des outils de l'expérimentation 111
5.2.4 Implémentation des mécanismes de collaboration par services Web 111
5.2.5 Implémentation des mécanismes de contrôle d'accès OrBAC 113
5.2.6 Implémentation des mécanismes de vérification des interactions par contrats 116
5.3 EXECUTION DU SCENARIO EN FONCTIONNEMENT NORMAL 117
5.4 FONCTIONNEMENT DU SCENARIO EN PRESENCE D'ERREURS 121
5.4.1 Double invocation du même service 122
5.4.2 Erreur d'Invocation non attendue 123
5.4.3 Violation d'une condition OrBAC sur le contexte 123
5.4.4 Erreur due à une tentative d'extension de privilège 124
5.4.5 Erreur liée à une échéance temporelle non respectée 124
5.4.6 Erreur dans les contrats en raison d'un message non attendu 125
5.5 CONCLUSION DU CHAPITRE 5 126

CONCLUSION GENERALE 129

BILAN DES CONTRIBUTIONS 129

PERSPECTIVES DE RECHERCHE 131

ANNEXES 135

BIBLIOGRAPHIE 141

Table des illustrations

Figure 1: Photo satellite de l'etat du reseau electrique 20 h avant le blackout du 14 aout 2003. 20
Figure 2: Photo satellite de l'etat du reseau electrique 7 h apres le blackout du 14 aout 2003. 20
Figure 3: Plan du manuscrit. 22
Figure 4: Classification simplifiee des infrastructures critiques. 26
Figure 5 : Interdependances entre les differentes infrastructures [Rinaldi et al., 2001] 28
Figure 6 : Les differentes dimensions des interdependances [Rinaldi, 2004] 29
Figure 7 : Interdependances entre les reseaux electriques europeens. 31
Figure 8 : Architecture generale du projet IRRIIS. 38
Figure 9 : Architecture des nœuds CRUTIAL. 40
Figure 10 : Positionnement de la notion de role dans le fonctionnement de RBAC. 46
Figure 11 : La relation « detient » et la relation « joue » 46
Figure 12 : La relation « Habilite ». 48
Figure 13 : La relation « Utilise ». 49
Figure 14 : La relation « Considere ». 50
Figure 15 : La relation « Definit ». 51
Figure 16 : Structure du modele OrBAC [Abou El Kalam *et al.*, 2003]. 52
Figure 17 : Collaboration a travers la notion de super-organisation. 54
Figure 18 : Collaboration en utilisation des liaisons « pair a pair » entre organisations. 56
Figure 19 : Ébauche du diagramme de classe pour la classe-association RdO. 57
Figure 20 : Exemple d'instanciation d'une regle de securite Multi-OrBAC. 58
Figure 21 : Notion d'Organisation Virtuelle (VO) [Nasser et al., 2005]. 59
Figure 22 : Le modele OrBAC au sein d'un environnement sous forme de grille [Nasser, 2006]. 61
Figure 23 : Fonctionnement du modele O2O [Coma 2006] 63
Figure 24 : Description simplifiee du fonctionnement des services Web. 71
Figure 25 : Collaboration entre les differentes organisations. 73
Figure 26 : L'architecture de l'approche PolyOrBAC. 74
Figure 27 : Image de service Web et utilisateur virtuel. 75
Figure 28 : Modelisation des permissions. 78
Figure 29 : Modelisation des interdictions. 78
Figure 30 : Modelisation des obligations. 79
Figure 31 : Modelisation des situations de conflits. 79
Figure 32 : Infrastructure de production d'electricite [Garrone et al., 2007] 86
Figure 33 : Infrastructure d'information et de communication d'une IPE [Garrone et al., 2007] . 87
Figure 34 : Architecture generale d'une IPE. 87
Figure 35 : Systeme de controle a distance des TSO et DSO [Garrone et al., 2007]. 88
Figure 36 : Les flux d'information et protocoles de communication [Garrone et al., 2007]. 89
Figure 37 : Signaux et mesures echanges dans le scenario de delestage. 91
Figure 38 : Diagramme de sequence du scenario de delestage. 92
Figure 39 : Architecture CRUTIAL d'une infrastructure d'information critique. 93
Figure 40 : Application du modele PolyOrBAC au scenario de delestage. 95
Figure 41 : Application des notions d'image de service Web et d'utilisateur virtuel au scenario. 95
Figure 42 : Automate de SW1-demande_d'armement au niveau du TS CC. 97

Figure 43 : Automate de SW1-demande_d'armement au niveau du DS CC. 99
Figure 44 : Automate pour SW2-ordre_d'armement du cote du DS CC. 100
Figure 45 : Automate pour WS2-arming-order au niveau du DS SS. 101
Figure 46 : Application du modele PolyOrBAC au scenario de delestage. 102
Figure 47 : Organisations et CIS au sein de l'IIC consideree. 108
Figure 48 : Plateforme PolyOrBAC. 110
Figure 49 : Panneau de controle du TSO. 112
Figure 50 : Panneau de controle du DSO. 113
Figure 51 : Outil OrBAC Designer : niveau concret. 114
Figure 52 : Outil OrBAC Designer : niveau abstrait 114
Figure 53 : Specification d'une regle de controle d'acces avec l'outil OrBAC Designer. 115
Figure 54 : Verification statique des politiques-contrats grace aux automates temporises. 116
Figure 55 : Exemple d'automate temporise pour la representation de politiques-contrats. 116
Figure 56 : Messages de controle du TSO. 117
Figure 57 : Messages JorBAC du TSO. 118
Figure 58 : Messages de controle du DSO. 118
Figure 59 : Messages de controle du DSO. 119
Figure 60 : Messages JOrBAC du DS CC. 119
Figure 61 : Envoi du message de confirmation d'armement du DS CC au TS CC. 120
Figure 62 : Signal d'urgence sur le panneau de controle du TSO. 120
Figure 63 : Signes de l'arret des sous stations armees. 121
Figure 64 : detection de la reception d'un message errone. 122
Figure 65 : connexion au TSO avec les identifiants roger/roger. 124
Figure 66 : Contrat dans un etat d'erreur apres un timeout. 125
Figure 67 : Boutons d'appel des web-services permettant le hacking. 125
Figure 68 : Le contrat signale la reception d'un message errone. 126

Table des tableaux

Tableau 1: Les coupures d'electricite les plus spectaculaires depuis 1999 [AFP 20/09/2003]..... 21
Tableau 2 : Infrastructures critiques les plus connues. 27
Tableau 3 : Énumeration des vulnerabilites des infrastructures d'information critique..... 36
Tableau 4 : Definition d'une permission avec le formalisme OrBAC. 52
Tableau 5 : Representation de la regle de controle d'acces OrBAC du cote client. 82
Tableau 6 : Representation de la regle de controle d'acces OrBAC du cote prestataire. 82
Tableau 7 : Representation de la regle OrBAC pour SW1 du cote TS CC. 96
Tableau 8 : Representation de la regle OrBAC pour SW1 du cote DS CC. 98
Tableau 9 : Representation de la regle OrBAC pour SW2 du cote DS SS..... 101
Tableau 10 : Mecanismes utilises et composants de PolyOrBAC. 109

Introduction générale et problématique

Dans le contexte actuel de crise économique, il est primordial de fournir tous les moyens permettant de maîtriser les risques et les menaces pouvant toucher de près ou de loin l'économie. Dans ce sens, de nombreuses études et activités de recherche sont menées pour optimiser et rendre plus sûres les infrastructures essentielles pour l'économie mondiale, en particulier au niveau européen.

L'infrastructure d'énergie électrique représente un des composants les plus importants de l'économie européenne en raison du fait que de nombreuses autres infrastructures en dépendent pour leur fonctionnement et une défaillance au niveau du réseau électrique peut avoir des conséquences catastrophiques sur les autres secteurs économiques.

Il existe plusieurs facteurs qui conditionnent le fonctionnement de l'infrastructure électrique, tout particulièrement en Europe. Les conditions actuelles du marché, avec en particulier la dérégulation du marché de l'électricité, les pressions économiques et financières et la variabilité de l'offre et de la demande, font partie des ces facteurs. L'infrastructure électrique regroupe de nombreuses parties prenantes, de différentes tailles (allant de multinationales à des petites et moyennes entreprises et aux particuliers), avec des fonctions variées (production, transport, distribution, commercialisation, courtage, autorités de régulation, etc.), et s'étendant sur plusieurs pays.

En raison de ce caractère international, l'infrastructure électrique européenne doit obéir à différentes normes de production, transmission, et distribution d'énergie. Il faut aussi satisfaire les besoins des clients en qualité de service (en particulier la disponibilité), et répondre aux différentes exigences économiques (c'est-à-dire les objectifs de rentabilité des actionnaires), fonctionnelles et organisationnelles (au sein de chaque établissement, et de chaque entreprise).

Plus spécifiquement, il existe des particularités fonctionnelles du système électrique qui font par exemple, qu'une défaillance apparemment minime d'un équipement peut provoquer, à cause des interdépendances au sein de l'infrastructure de transport et de distribution électrique, des phénomènes de cascade et d'escalade, qui peuvent conduire à des défaillances généralisées (arrêts complets du système électrique, ou « *black-out* ») avec des conséquences potentiellement dramatiques sur l'ensemble du réseau et sur les usagers.

Effectivement, en raison des interdépendances entre les différentes infrastructures électriques de plusieurs zones géographiques, il peut se produire des défaillances en cascade (la défaillance

d'une infrastructure conduisant à la défaillance des autres) ou en escalade (des défaillances mineures se combinant pour provoquer des conséquences graves). Une simple panne peut produire une coupure électrique généralisée comme le *black-out* du 14 août 2003 qui a touché l'Amérique du Nord. Cette coupure totale de la fourniture d'électricité d'une région des États-Unis et du Canada qui a causé 6 milliards de dollars de pertes et de dégâts et a touché 50 millions de personnes selon le ministère de l'énergie des États-Unis [Amin, 2003], n'a duré que quelques heures, et avait pour origine une simple surcharge d'une ligne de transmission, qui a été amplifiée par des pannes en escalade et des pannes au niveau du système de contrôle et de commande. L'origine de cette catastrophe économique était surtout liée à l'infrastructure physique, mais en ce qui nous concerne nous allons nous concentrer sur les causes liées au système d'information gérant l'infrastructure d'énergie électrique. Pour résumer les effets dévastateurs de ce *black-out*, voici une énumération non exhaustive des conséquences relevées :

- 6 milliards de dollars de pertes financières,
- 50 millions de personnes touchées,
- 61800 mégawatts d'énergie perdus,
- 100 centrales de production électrique arrêtées,
- 100 stations de distribution d'électricité arrêtées,
- 35 usines automobiles arrêtées,
- 12 aéroports fermés,
- 400 vols annulés,
- 1,5 million de résidents de Cleveland sans eau,
- 13500 km^2 touchés,
- 8 états américains et 2 provinces canadiennes affectés,
- plusieurs centaines de personnes sont restées enfermées dans les métros et les ascenseurs.

Les figures suivantes présentent l'état du réseau électrique nord-américain avant et pendant la panne électrique.

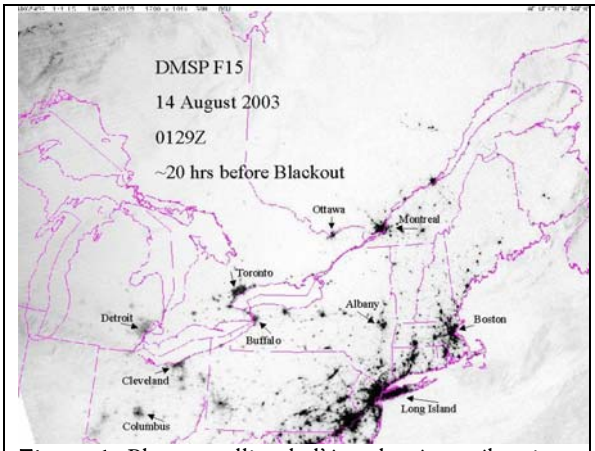


Figure 1: Photo satellite de l'état du réseau électrique 20 h avant le blackout du 14 août 2003.

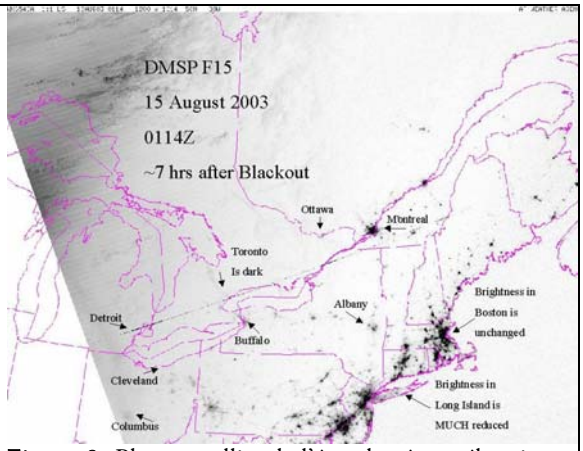


Figure 2: Photo satellite de l'état du réseau électrique 7 h après le blackout du 14 août 2003.

Sur la Figure 1, qui montre l'état du réseau électrique 20h avant le *black-out*, on remarque que les villes de Toronto, Detroit, Cleveland, Colombus, Ottawa, sont actives, et que

la région de Long Island est très active. Sur la Figure 2, qui montre l'état du réseau 7 heures après le *black-out*, on remarque que les villes de Toronto, Detroit, Cleveland, Colombus, Ottawa, ne sont plus actives, la région de Long Island a une activité électrique très réduite. Pour mettre en évidence les risques d'occurrence de ce genre d'incident, il est intéressant de citer quelques *black-outs*, classés dans le Tableau 1 par nombre de personnes touchées:

Lieu	Date	Personnes touchées
France	26 décembre 1999	3,6 millions de foyers
Inde	2 janvier 2001	200 millions
États-Unis (San Francisco)	18 janvier 2001	1 million
Nigéria	juin 2001	30 à 50 millions
Colombie	16 mars 2002	15 millions
Philippines	mai 2002	30 à 40 millions
Argentine (Buenos Aires)	24 novembre 2002	2 millions
Algérie	3 février 2003	34 millions
Amérique du Nord	14 août 2003	50 millions
Suède/Danemark	23 septembre 2003	4 millions de foyers
Italie	28 septembre 2003	60 millions
Europe (origine : Hambourg)[1]	4 novembre 2006	>15 millions de foyers

Tableau 1: Les coupures d'électricité les plus spectaculaires depuis 1999 [AFP 20/09/2003].

Comme l'infrastructure électrique, les autres infrastructures critiques (IC) sont caractérisées par la présence de vulnérabilités physiques et logiques, et à l'ère de l'information, le fonctionnement des IC dépend de plus en plus du bon fonctionnement de leurs systèmes d'information et de communication, c'est-à-dire de l'infrastructure d'information critique (IIC).

Dans cette thèse, nous nous intéressons en particulier à la sécurité des IIC dédiées aux réseaux d'énergie électrique. Ces IIC intègrent des technologies de l'information fragiles, présentant des menaces spécifiques : dysfonctionnement du système informatique, plantage, fautes d'interaction, en particulier les mauvaises manipulations accidentelles ou volontaires (c'est-à-dire les malveillances), etc. Si l'infrastructure de production, de transport et de distribution d'électricité est affectée par une panne électrique classique (telle qu'une surcharge de ligne de transmission ou distribution), ceci peut conduire à des pannes importantes. D'un autre côté, des défaillances au niveau de l'IIC peuvent conduire aux mêmes résultats dramatiques.

La sécurité informatique des infrastructures critiques constitue une branche de la sécurité des systèmes d'information qu'on nomme protection des infrastructures critiques, dont la sécurité informatique des infrastructures de production, de transport et de distribution d'électricité en particulier constitue un cas d'étude particulier. Il est primordial de renforcer la sécurité et la protection des réseaux d'énergie électrique. La protection des infrastructures critiques comprend l'étude, la conception et l'exécution des mesures conservatoires visant à réduire le risque que l'infrastructure critique rencontre des actes de guerre, de désastre, de

[1] <http://www.robert-schuman.eu/question_europe.php?num=qe-46>, <http://www.ucte.org/_library/otherreports/Final-Report-20070130.pdf>

vandalisme, de sabotage ou des accidents. Dans cette thèse, nous nous focalisons en particulier sur les problèmes de sécurité du système d'information, et nous proposons des solutions pour le contrôle d'accès et à la collaboration au sein d'une IC, basées sur le modèle de contrôle d'accès OrBAC et la technologie des services Web.

La suite du manuscrit est organisée comme suit :

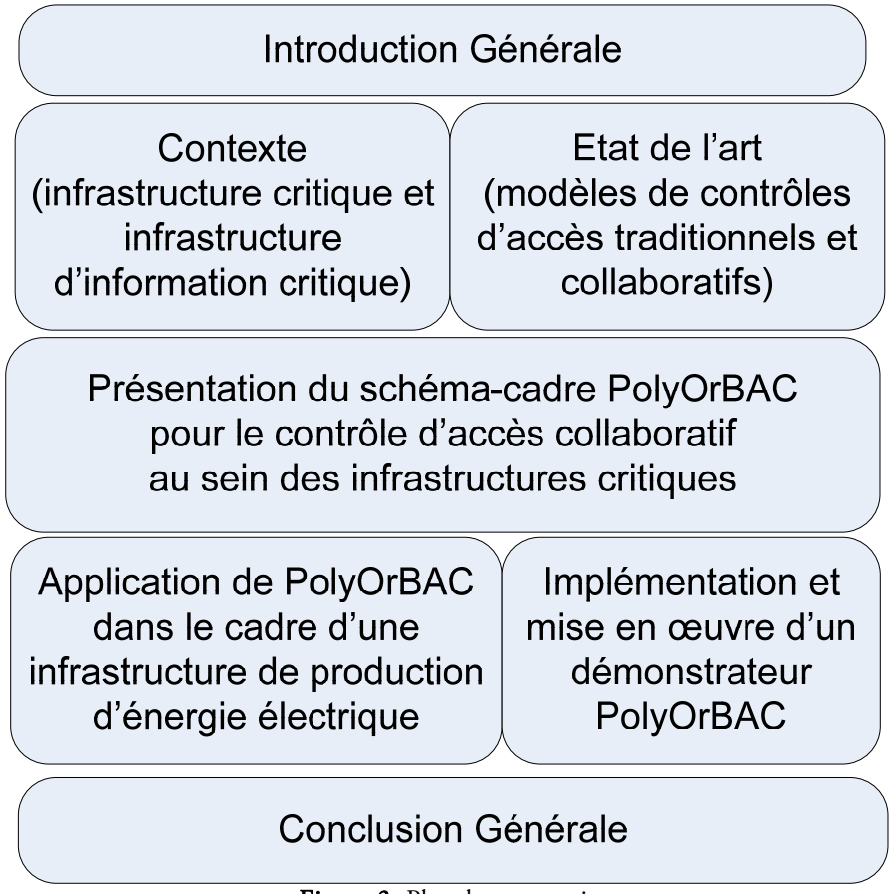


Figure 3: Plan du manuscrit.

Le premier chapitre (contexte) décrit de façon détaillée les différentes caractéristiques d'une infrastructure critique, ainsi que celles d'une infrastructure d'information critique. Ce chapitre décrit également les différentes vulnérabilités et menaces pesant sur les différents types d'infrastructure, ainsi que les besoins fonctionnels, les exigences de performance et les spécifications de sécurité de ces infrastructures. Enfin, ce chapitre détaille quelques travaux pour la protection des infrastructures critiques, parmi lesquels le projet européen CRUTIAL, qui représente le contexte de notre travail, et qui traite des moyens pour sécuriser les infrastructures critiques, et plus particulièrement les infrastructures pour la production, transport, et distribution d'énergie électrique.

Le deuxième chapitre (état de l'art) présente différents modèles de sécurité traditionnels (RBAC, OrBAC) et d'autres modèles pour la collaboration d'organisation (Multi-OrBAC, O2O) existants, en détaillant à chaque fois, les différents avantages et limites de chaque modèle.

Le troisième chapitre (présentation de PolyOrBAC) récapitule les différents besoins d'une infrastructure d'information critique et les différentes propriétés que doit fournir un modèle de politique de sécurité pour la collaboration. Dans un deuxième temps, nous présentons PolyOrBAC, le schéma-cadre que nous proposons pour répondre aux différents besoins de contrôle d'accès collaboratif que doit satisfaire une infrastructure d'information critique. Nous montrons que notre schéma repose sur deux composants principaux, le modèle de contrôle d'accès OrBAC, et la technologie des services Web, ce qui permet de cumuler les différents avantages d'OrBAC et des services Web, puis de proposer un modèle de contrôle d'accès collaboratif.

Le quatrième chapitre (étude de cas et application) décrit ensuite la manière dont PolyOrBAC peut être appliqué dans le contexte d'une infrastructure critique dédiée à l'énergie électrique : on décrit un scénario réaliste et représentatif, puis on montre comment le contrôle d'accès et l'interopérabilité nécessaire dans ce type d'infrastructure sont couverts par PolyOrBAC.

Le cinquième chapitre (mise en œuvre) présente une implémentation qui a été réalisée pour montrer l'applicabilité de l'approche et la facilité de mise en œuvre de notre schéma. Cette implémentation utilise des technologies répandues telles que Java, services Web (XML, SOAP, WSDL, etc.), et fournit ainsi une maquette portable sur n'importe quelle architecture Unix, et rapidement adaptable sur toute autre architecture non-Unix.

Pour conclure, nous récapitulons les contributions apportées par cette étude et proposons des extensions et perspectives possibles de ce travail, en particulier la possibilité d'intégrer une gestion de différents niveaux d'intégrité et de criticité existants au sein d'une même organisation ou entre plusieurs organisations collaborant.

Chapitre 1. Sécurité des systèmes d'information dans les infrastructures critiques

La vie publique, l'économie et la société dans son ensemble dépendent dans une très large mesure du bon fonctionnement de certaines infrastructures critiques (IC), comme l'approvisionnement en énergie ou les réseaux de télécommunications. Notre travail de thèse vise à protéger ces infrastructures. L'utilisation des technologies de communication et de l'information ont envahi d'autres infrastructures, les rendant plus intelligentes, de plus en plus interconnectées, complexes, interdépendantes et donc plus vulnérables.

Dans le but de fixer le contexte des travaux de notre thèse, qui est l'environnement des infrastructures critiques, nous allons dans un premier temps donner une vue générale sur la sécurité dans les systèmes d'information et de communication des infrastructures critiques ; dans un deuxième temps, nous allons décrire les différents besoins et critères de sécurité dans ce genre d'infrastructures.

Ce chapitre est donc composé de deux parties :

- Dans la première, nous donnons une description détaillée des infrastructures critiques (IC) puis des infrastructures d'information critiques (IIC). Nous commençons tout d'abord par rappeler les différentes caractéristiques des IC et des IIC, en tenant compte des besoins de sécurité. Nous présentons ensuite le domaine de la protection des IC et plus particulièrement la protection des IIC.

- Dans la deuxième partie, nous détaillons quelques études et travaux existants qui proposent des moyens pour répondre aux besoins et critères de sécurité des IC en général et des IIC en particulier, tout en tenant compte des concepts fondamentaux de sécurité cités précédemment.

1.1 Les infrastructures critiques

Afin de bien comprendre et mettre en évidence les différents concepts liés au domaine des IC et IIC, cette section dresse un état de l'art décrivant les infrastructures critiques en général et les infrastructures d'information critiques en particulier. Il s'agit de détailler leurs caractéristiques et particularités qui les différencient des autres systèmes d'information, leurs besoins organisationnels en général et leurs exigences de sécurité en particulier, et enfin les vulnérabilités et menaces touchant chaque type infrastructure.

Une infrastructure critique (IC) [Moteff & Parfomak 2006] est constituée d'un ensemble d'organisations, installations, équipements, biens logiques et physiques, qui a une importance vitale ou critique [Masse et al., 2003] pour le fonctionnement d'une économie en particulier, ou de la société humaine en général et dont la défaillance, l'arrêt de fonctionnement ou la dégradation (même temporaire) peuvent avoir un impact potentiellement dramatique sur le bien-être économique et social d'une nation.

1.1.1 Classification des infrastructures critiques

La Figure 4 récapitule les trois catégories les plus importantes dans lesquelles on peut classer l'ensemble des infrastructures critiques : infrastructures d'approvisionnement qui regroupent entre autres les fournitures d'eau, de gaz et de carburant (pétrole, fuel, etc.), infrastructures tertiaires qui regroupent entre autre les services bancaires, sécuritaires (par exemple, police, armée), et sanitaires, et enfin les infrastructures de transport aérien, terrestre et maritime.

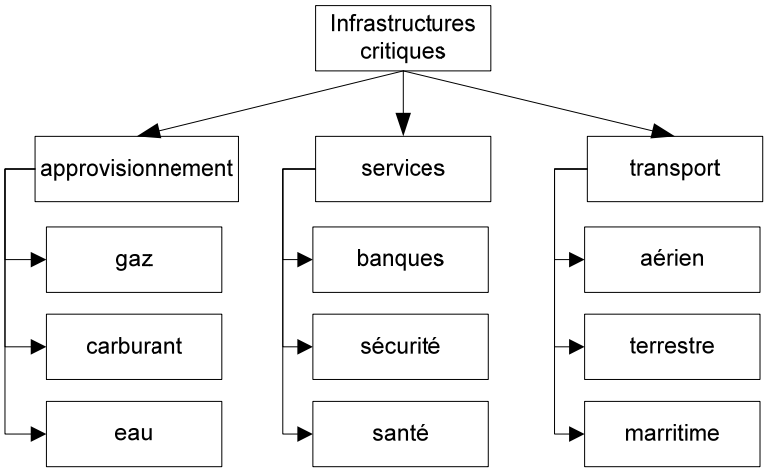


Figure 4: Classification simplifiée des infrastructures critiques.

1.1.2 Exemples d'infrastructures critiques

Parmi les exemples les plus significatifs d'infrastructures critiques [Knight et al., 1998], il faut citer les équipements et les installations pour la production, le transport et la distribution d'énergie électrique et de gaz naturel, ou encore la production, le raffinement et la distribution de carburant. Ces infrastructures sont parmi les plus importantes d'une nation, car de leur fonctionnement correct dépendent une multitude d'autres infrastructures critiques. À une

échelle plus grande, l'ensemble des infrastructures mondiales dépend du réseau d'énergie électrique mondial (l'ensemble interconnecté des réseaux d'énergie électrique). Le Tableau 2 récapitule d'autres infrastructures importantes.

Type d'Infrastructure	Exemple
Les services de production, d'approvisionnement et de stockage d'énergie	électricité, carburant, gaz, etc.
Nourriture et agriculture	sécurité, production, distribution, etc.
Les réseaux de transport	routier, maritime, ferroviaire, aéroport, etc.
La santé	hôpitaux, services d'approvisionnement en sang et en matériels, pharmaceutiques, services médicaux dans des situations d'urgence, etc.
Les services financiers	banques, bourse, assurances, etc.
Les services de sécurité	police, armée, pompiers, services de secours, services d'urgence, justice, prisons, etc.
Les télécommunications	Téléphone, réseaux informatiques, etc.
Les grands groupes industriels ou économiques	Commerce, presse, construction, audiovisuel, etc.
Approvisionnement, stockage et distribution d'eau	Eau potable, gestion et traitement des eaux usagées, etc.

Tableau 2 : Infrastructures critiques les plus connues.

1.1.3 Interdépendance des infrastructures critiques

En raison de besoins économiques, organisationnels et fonctionnels de plus en plus importants, les grandes infrastructures critiques (par exemple. énergie, transport, télécommunications, etc.) dépendent de plus en plus les unes des autres. Ces interdépendances peuvent être bipartites ou multiples entre plusieurs IC. Des exemples de telles interdépendances sont détaillés dans la Figure 5.

Certaines de ces interdépendances sont relatives à l'infrastructure de production d'électricité (IPE), dans la mesure où l'IPE dépend des autres infrastructures (par exemple, le besoin de l'IPE en carburant pour faire fonctionner ses générateurs, le besoin de l'IPE en technologies de l'information pour faire fonctionner ses systèmes de contrôle-commande, son besoin en eau pour le refroidissement de certains composants) et d'un autre côté, presque toutes les autres infrastructures dépendent de l'IPE.

Par ailleurs, ces infrastructures critiques intègrent, pour des besoins évidents de fonctionnement et de gestion de données, des technologies et systèmes d'information et de communication eux-mêmes fragiles et présentant des risques de défaillance, en particulier vis-à-vis des malveillances volontaires, mais également vis-à-vis des fautes physiques et fautes d'interaction ou de conception involontaires du système. Ces interdépendances (entre IC, entre IC et IIC, et entre IIC appartenant à différentes IC) peuvent conduire à des défaillances en

cascade [Amin, 2003] (lorsque la défaillance d'une infrastructure conduit à la défaillance d'autres infrastructures) et des défaillances en escalade (lorsqu'une défaillance mineure dans une infrastructure provoque rapidement une panne grave de la même infrastructure) [Amin, 2003].

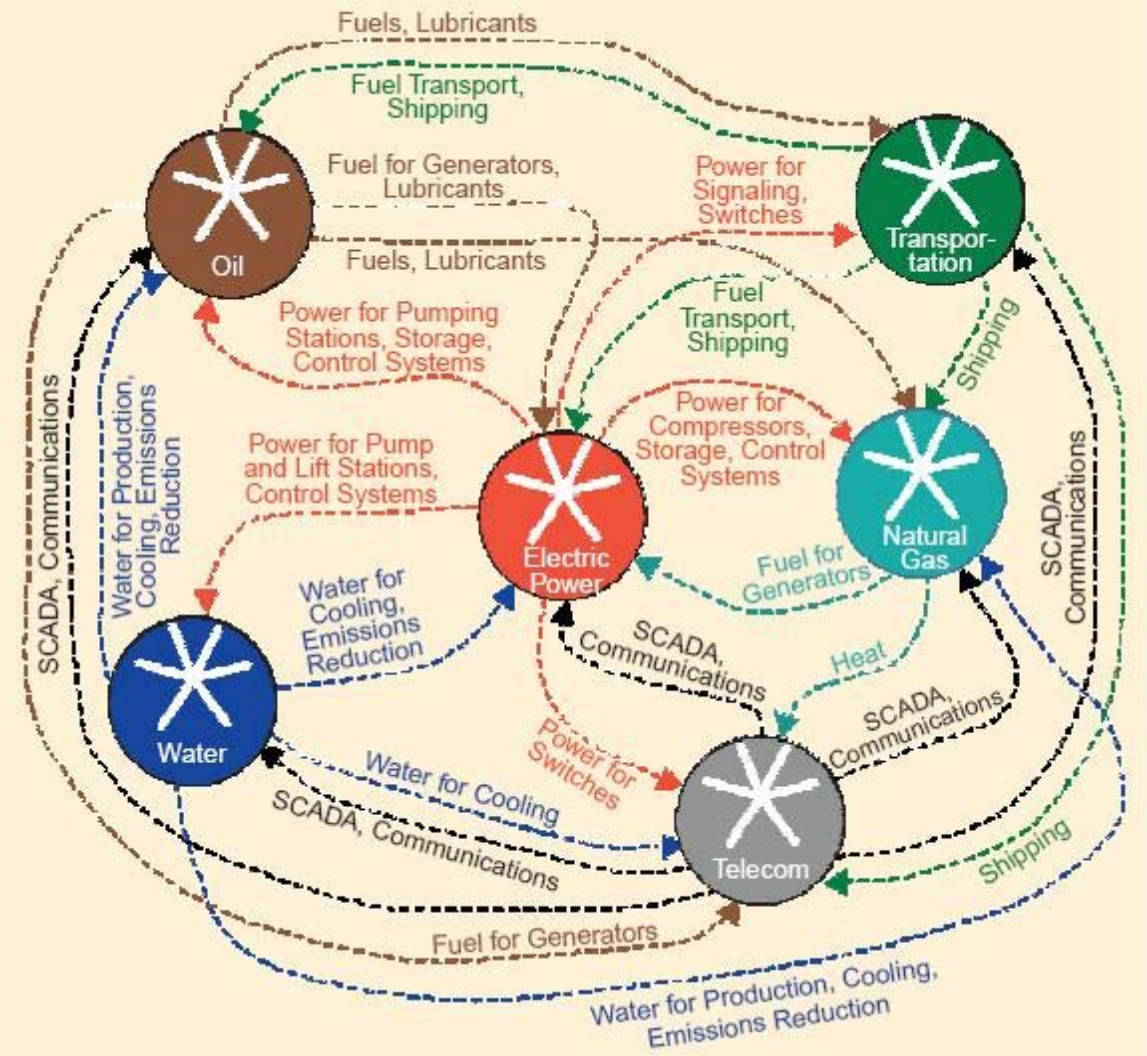


Figure 5 : Interdépendances entre les différentes infrastructures [Rinaldi et al., 2001]

En outre, ces phénomènes d'interdépendances, et donc de pannes en cascade potentielles, sont amplifiés par l'ouverture et la dérégulation des marchés, autrefois contrôlés par des monopoles d'état et maintenant ouverts à une concurrence féroce. Ces facteurs économiques rendent la gestion et le fonctionnement des IC plus cruciaux encore et doivent être pris en considération. D'un autre côté, la taille importante (en étendue et en structure) de l'infrastructure critique introduit différentes vulnérabilités (et donc des menaces correspondantes). Il faut tenir compte par exemple des problèmes de disponibilité des ressources et services et des dysfonctionnements des systèmes informatiques en particulier. De plus, les différentes interdépendances et connexions entre les différentes IC agissent sur l'économie de façon directe en raison des différents contrats et accords économiques établis avec les différentes infrastructures, ce qui rend la gestion d'une IC plus complexe que si cette dernière était complètement isolée. Pour mettre en évidence la complexité de ces interdépendances, nous

décrivons les différents facteurs, conditions, et critères qui interviennent au niveau des interdépendances dans la Figure 6.

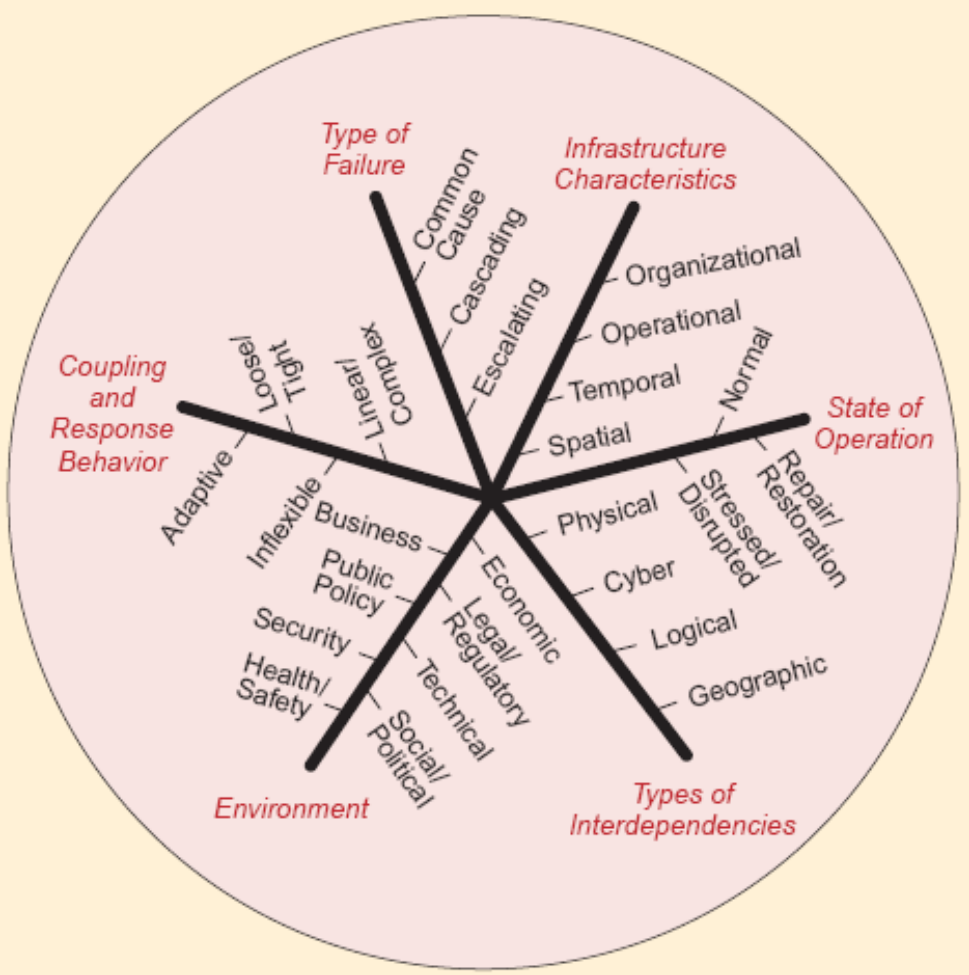


Figure 6 : Les différentes dimensions des interdépendances [Rinaldi, 2004].

La Figure 6 classe ces facteurs selon six dimensions (type de panne, caractéristiques des infrastructures, type d'interdépendances, environnement, type de couplage entre infrastructures et systèmes, état du système opérationnel). Pour ne citer que quelques facteurs, il est nécessaire de gérer les caractéristiques opérationnelles, organisationnelles, temporelles et spatiales des infrastructures, ces caractéristiques pouvant affecter leur capacité à s'adapter à l'évolution des conditions de fonctionnement du système. Les interdépendances peuvent être logiques ou physiques et il s'agit de prendre en compte, entre autres, les caractéristiques économiques, techniques et sociales de l'environnement. De même, le type de couplage (fort ou faible) entre les infrastructures peut influer sur les caractéristiques opérationnelles des différentes infrastructures et entre en ligne de compte des défaillances en escalade et cascade car il touche aux conditions de fonctionnement des systèmes et donc d'évolution des défaillances. Il existe une autre perspective concernant les interdépendances qu'il est nécessaire de mettre en évidence. Nous identifions en particulier trois types d'interdépendances concernant une infrastructure critique :

- Une première classe d'interdépendance existe entre une infrastructure critique en général et le réseau d'énergie électrique [Knight *et al.*, 1998] qui l'approvisionne en

énergie pour son fonctionnement [Laprie *et al.*, 2007]. Une panne minime (coupure d'une seule ligne de distribution par exemple) au niveau du réseau de transport et de distribution d'électricité peut conduire à l'arrêt complet du fonctionnement de certaines infrastructures critiques (comme lors du blackout nord-américain d'août 2003, où le réseau de transport souterrain s'est arrêté à cause de la panne électrique) [Amin, 2003].

- À l'ère de l'information, le fonctionnement des infrastructures critiques dépend de plus en plus du bon fonctionnement de systèmes d'information et de communication. Ces systèmes forment donc eux-mêmes une infrastructure critique (télécommunications, logiciels, matériel, réseaux d'informations et de communication, y compris Internet pour des fonctions peu critiques) [Bialas, 2006]. Actuellement, chaque infrastructure critique dépend de son système d'information et de communication et tout dysfonctionnement dans ce dernier est considéré comme une menace sur l'infrastructure critique elle-même. Dans ce sens, la deuxième classe d'interdépendance à prendre en compte réside dans les interdépendances entre une IC et son IIC [Bialas, 2006]. Plus particulièrement, une défaillance mineure (à cause d'un défaut de fonctionnement, erreur humaine, ou intrusion malveillante par exemple) au niveau de l'infrastructure d'information critique, peut avoir comme conséquence (après pannes en cascade et escalade) une panne généralisée de l'infrastructure critique globale elle-même.
- Enfin, il faut bien sûr tenir compte de l'interdépendance entre l'infrastructure d'information critique et le réseau d'énergie électrique [Laprie *et al.*, 2007]. Chaque réseau d'énergie électrique dépend elle-même d'un système d'information et de communication spécifique, en particulier d'un ou de plusieurs systèmes de contrôle (par exemple, les systèmes SCADA), dont le but est de surveiller le fonctionnement du réseau, de collecter toutes les informations nécessaires et de réaliser les différentes opérations pour adapter en permanence le fonctionnement du réseau d'énergie électrique aux conditions internes et externes.

L'infrastructure de production, transport et distribution d'énergie électrique a donc une importance particulière, mais elle présente aussi une complexité qui lui est propre. En particulier, il est primordial de détailler les différentes interdépendances entre les différents réseaux électriques géographiquement voisins, tels que les réseaux européens qui partagent et négocient leurs productions et distributions respectives d'énergie. La Figure 7 présente les échanges d'électricité entre les différents pays européens.

Ces interdépendances rendent la gestion de la sécurité de chaque sous-réseau d'énergie électrique encore plus importante, puisqu'une défaillance (même minime) dans ce sous-réseau (ou de son système d'information, ou d'un de ses composants) peut avoir une conséquence directe sur l'ensemble du réseau.

La multitude d'interdépendances entre différents réseaux d'énergie, entre les infrastructures critiques, entre leurs systèmes d'information respectifs ne constitue pas la seule faille de sécurité à prendre en compte. Dans la section suivante, nous détaillons différentes vulnérabilités et menaces existant dans les infrastructures critiques.

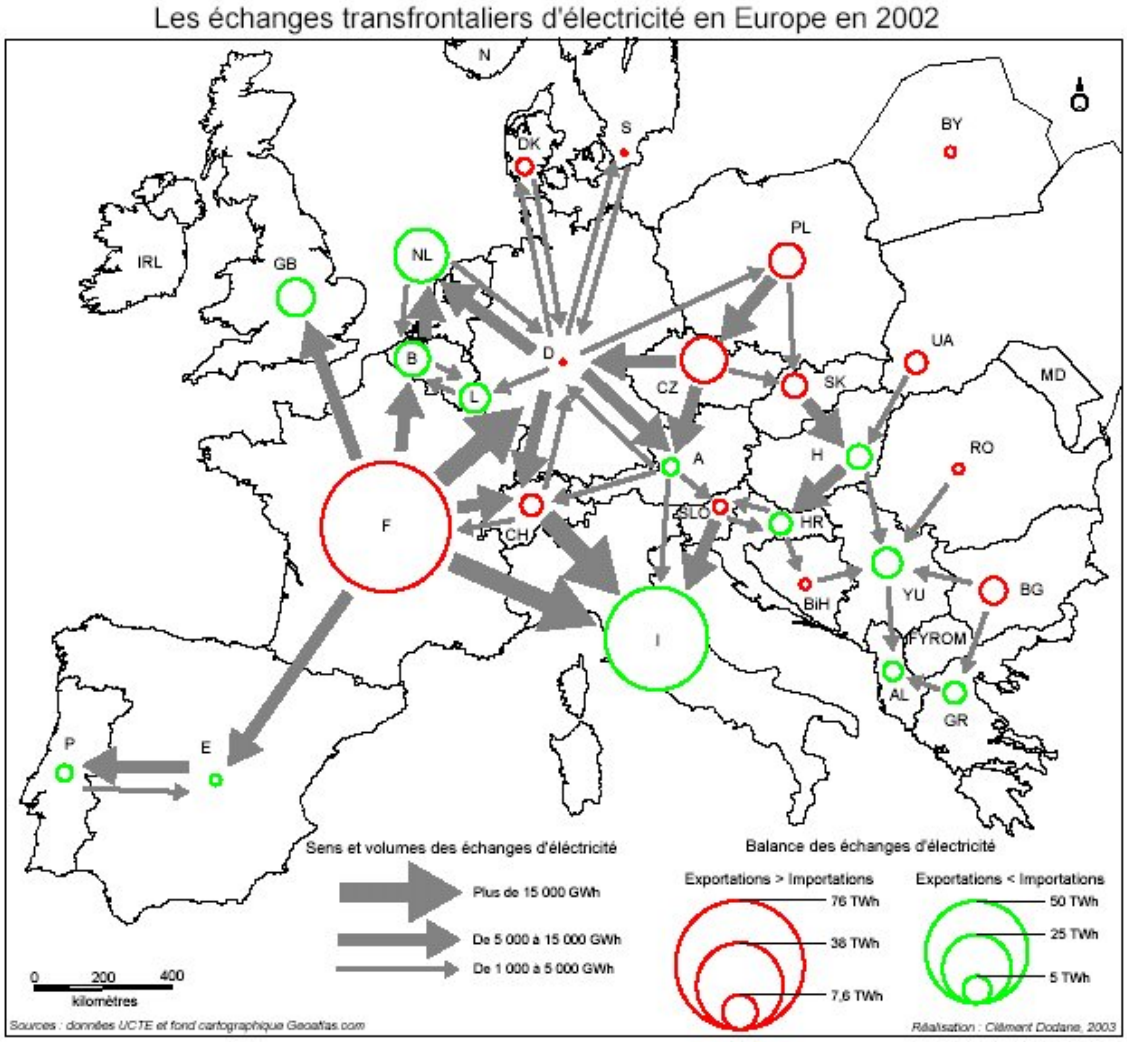


Figure 7 : Interdépendances entre les réseaux électriques européens[2].

1.1.4 Vulnérabilités et menaces dans les infrastructures critiques

Afin de couvrir l'ensemble des vulnérabilités et menaces liées à une infrastructure critique, il nous semble nécessaire de différencier plusieurs niveaux de vulnérabilités : vulnérabilités au niveau de l'infrastructure critique, vulnérabilités au niveau du réseau d'énergie électrique, vulnérabilités au niveau de l'infrastructure d'information critique. Ces trois catégories sont détaillées dans les paragraphes suivants :

1. Vulnérabilités touchant l'infrastructure critique [Sikich, 1998]. Parmi ces vulnérabilités, on peut citer :
   - Les différentes interdépendances entre deux IC, entre les organisations qu'elles composent, ou entre l'IC et l'IIC qu'elle inclut, ne sont pas vraiment des vulnérabilités mais elles peuvent y conduire.
   - Le vieillissement de composants matériels et physiques utilisés.

[2] http://geoconfluences.ens-lsh.fr/doc/breves/2003/03-5.htm

- L'absence de savoirs et connaissances suffisants pour la gestion d'une IC particulière.
- L'utilisation de certaines technologies standardisées ayant des vulnérabilités (par exemple des protocoles comme http, des systèmes d'exploitation vulnérables, etc.), connues du grand public (disponibilité de l'information).
- Les infrastructures critiques ont une architecture dynamique, ce qui pose un problème de mise à jour des matériels et de l'organisation générale.
- Une infrastructure critique contient des composants hétérogènes qu'il faut utiliser et il faut veiller à leur fonctionnement correct et continu.
- Les infrastructures critiques ont généralement une taille considérable, il est donc nécessaire de tenir compte du besoin de passage à l'échelle (*scalability*) et de la multiplicité des composants qu'elle peut inclure, de leur complexité et de la complexité de leurs interconnexions.

2. Vulnérabilités touchant le réseau d'énergie électrique [Amin, 2002], parmi lesquelles on peut citer les coupures de ligne physiques, les surcharges de lignes de transport ou de distribution, les pannes au niveau des centrales, des stations de transport ou de distribution, etc.
3. Vulnérabilités touchant l'infrastructure d'information critique [Sikich, 1998]. Ces vulnérabilités seront détaillées de façon plus approfondie dans la section suivante qui donne une description des infrastructures d'information critiques et introduit les points de sécurité sur lesquels nous nous concentrons dans cette thèse.

Afin de prendre en compte ces différentes vulnérabilités, il est donc primordial de renforcer la protection de ces infrastructures critiques. On peut considérer trois points de sécurité à traiter :

- La protection des infrastructures critiques [Dacey, 2002], qui inclut la prise en compte des différentes vulnérabilités [Ordonez, 2006], et failles de sécurité [Powner *et al.*, 2005], ainsi que la définition et gestion de politiques de sécurité pour ces infrastructures [Moteff, 1998].
- La protection des réseaux d'énergie électrique [Knight *et al.*, 1998] et [Hauser *et al.*, 2008].
- Et enfin, la protection des infrastructures d'information critique [Bologna et al., 2006]. En effet, dans le but de protéger une infrastructure critique, il est nécessaire de protéger le système d'information de cette dernière, ceci constitue un point central dans cette étude.

Dans le but de mieux comprendre le fonctionnement opérationnel d'une infrastructure d'information critique, cette section vise à donner une définition et une description détaillée d'une infrastructure d'information critique en général, puis à détailler les caractéristiques, les besoins, les vulnérabilités et menaces touchant ce genre d'infrastructure.

1.2 Définition et enjeux d'une infrastructure d'information critique

L'infrastructure d'information critique peut être menacée par des groupes criminels, par des services de gouvernements étrangers hostiles, par des terroristes, par des *hackers*, ou par des menaces internes. En outre, ces IIC intègrent des technologies de l'information fragiles (protocoles complexes, utilisation de systèmes d'exploitation vulnérables, logiciels et matériels ayant des failles connues), devant faire face à des menaces spécifiques (défaut de fonctionnement, erreurs de programmation, erreurs humaines, malveillances, etc.), pouvant mener à des pannes plus graves au niveau de l'IC qu'elles contrôlent et donner en fin de compte les mêmes résultats qu'une défaillance globale de l'IC elle-même. Les besoins et exigences en sécurité ainsi que les menaces touchant ce genre d'infrastructure seront détaillées dans la section suivante.

1.2.1 Besoins et exigences de sécurité dans les infrastructures critiques

Dans le but de prendre en compte les différentes caractéristiques d'une infrastructure d'information critique dans le cadre de notre étude, nous devons gérer et mettre en évidence de manière plus claire les différentes propriétés et exigences de sécurité, la disponibilité des services offerts, l'intégrité des données utilisées et leur confidentialité vis-à-vis de tiers non autorisés, etc. En particulier, afin de faire face à l'évolution de l'environnement de l'infrastructure critique, la structure de l'IIC doit prendre en compte un certain nombre de caractéristiques conceptuelles et fonctionnelles [Moteff & Parfomak 2006] que l'on va détailler dans ce qui suit:

- **Complexité :** généralement, afin d'assurer des besoins opérationnels et fonctionnels, une IIC contient des systèmes (logiques et physiques) complexes, ouverts, hétérogènes, distribués et collaboratifs, dont certains doivent être accessibles par des utilisateurs internes et externes. Dans cette optique, il est absolument nécessaire de prévoir des outils et moyens afin de gérer au mieux cette complexité des systèmes d'information inclus dans l'IC, ainsi que les systèmes et les composants spécifiques à l'IC elle-même (par exemple, les systèmes de production, transport et distribution de l'énergie électrique dans le réseau d'énergie électrique).
- **Caractère multinational :** afin de favoriser le développement économique et de favoriser les aspects de collaboration, une IIC peut s'étendre sur plusieurs zones géographiques, potentiellement dans différents pays voisins (par exemple, le réseau électrique européen qui relie entre-autres, la France, la Belgique, l'Italie, l'Espagne, etc.). Cette extension territoriale et géographique de l'IIC implique de tenir en compte des procédures et normes différentes de création, d'administration et d'utilisation des systèmes d'information qui peuvent être propres à chaque région ou pays.
- **Extensibilité en taille et structure :** pour prendre en compte les changements continuels de structure et l'aspect dynamique de l'architecture de l'IC en général et de l'IIC en particulier, l'IIC doit assurer des propriétés de souplesse, flexibilité et extensibilité en taille et en structure, en nombre d'utilisateurs et de composants, etc.
- **Caractère multi-organisationnel :** pour des raisons économiques, fonctionnelles et organisationnelles, une IIC interconnecte plusieurs organisations qui sont

susceptibles de collaborer. Plus particulièrement, ces différentes organisations qui participent à une même IC ou appartenant à différentes IC, peuvent coopérer afin de négocier l'utilisation de certaines ressources ou l'achat de certains services. Ceci permet aux différentes organisations de fournir des ressources à des utilisateurs externes tout en gardant leur indépendance et autonomie. D'un autre côté, avec l'ouverture et la dérégulation des marchés, certaines de ces organisations peuvent être en concurrence entre elles, et donc mutuellement méfiantes. C'est le cas en particulier en Europe, où des entreprises régionales, nationales ou multinationales sont en concurrence mais doivent coopérer pour produire, transporter et distribuer l'énergie électrique.

Certaines caractéristiques des infrastructures d'information critiques se rattachent particulièrement à la notion de dépendance et d'interdépendance entre IIC, c'est pour cette raison que nous avons regroupé ces caractéristiques dans la section suivante.

1.2.2 Interdépendance des infrastructures d'information critiques

Les organisations qui sont parties prenantes dans une infrastructure critique présentent généralement les caractéristiques suivantes :

- **Collaboration :** les différentes organisations participant à une même IIC ont un besoin vital de collaboration [Scholand et al., 2005], c'est-à-dire de partager et échanger des données, des services et des ressources internes [Dacey, 2004], [Brock, 2000]. Il est primordial de noter que la collaboration nécessite le partage de ressources et l'externalisation de services par chaque organisation, le tout dans une architecture décentralisée, favorisant l'interopérabilité par l'inter-connectivité, mais aussi induisant une certaine interdépendance. Ces trois notions sont détaillées dans ce qui suit.
- **Inter-connectivité :** les différentes organisations participant à une IIC doivent être interconnectées. Ceci nécessite l'utilisation de composants standards (routeurs, pare-feux, …) disposés localement dans chaque organisation.
- **Interdépendance :** l'inter-connectivité peut induire une interdépendance entre les différents composants. Le fonctionnement d'une organisation a potentiellement des conséquences sur les autres organisations qui lui sont reliées, ce qui peut soulever des problèmes de sécurité qui doivent être résolus à la fois localement par chaque organisation, et au niveau de l'IIC par une gestion conjointe de la sécurité globale.
- **Interopérabilité :** une autre notion importante à prendre en compte dans notre gestion de l'interdépendance est l'interopérabilité. Il est donc souhaitable que les organisations, interconnectées et interdépendantes utilisent un standard commun de communication, de partage et d'échange de données.
- **Concurrence et suspicion :** malgré l'intérêt commun à coopérer, les différentes organisations peuvent être en concurrence et donc se méfier les unes des autres en raison des mêmes intérêts financiers en conflits qui les ont poussées à chercher des moyens pour coopérer. Il est également important de noter qu'il est souhaitable que le couplage entre les organisations soit faible pour réduire l'interdépendance, en

particulier vis-à-vis de la sécurité. Par exemple, chaque organisation doit pouvoir définir sa propre politique de sécurité, ses objectifs de sécurité, ses services, ses applications, ses systèmes d'exploitation, etc. Ainsi, il faut fournir à chaque organisation des moyens pour coopérer et contrôler ses interactions avec les autres organisations, tout en conservant son autonomie.

- **Autonomie :** pour réduire l'interdépendance avec des organisations dont on se méfie, il est important que chaque organisation soit autant que possible autonome dans la gestion de ses ressources et de ses personnels. Du point de vue de la sécurité, cela signifie que chaque organisation peut choisir en toute indépendance ses moyens d'authentification de ses utilisateurs et les mécanismes de contrôle d'accès qui mettent en œuvre sa propre politique de sécurité. Les interactions avec d'autres organisations doivent être compatibles avec cette politique, ces moyens et ces mécanismes, et ne doivent rien révéler de la structure interne de l'organisation (en particulier sur ses ressources et ses utilisateurs), ni de ses évolutions éventuelles.
- **Contrôle d'accès et sécurité :** la coopération nécessite que certaines organisations fournissent des services qui puissent être utilisés par d'autres, tout en respectant la politique de sécurité de chaque organisation et son autonomie. Ceci signifie qu'il faut que la fourniture et l'utilisation de ces services soit compatibles avec la politique de sécurité de l'organisation qui fournit le service aussi bien qu'avec celle de l'organisation qui l'utilise, et donc que les moyens qui mettent en œuvre ces politiques contrôlent à la fois la fourniture et l'utilisation de ces services. Dans notre étude nous nous intéressons tout particulièrement aux besoins de collaboration existants au sein d'une IIC, pour le partage et l'échange d'information de façon sécurisée et donc aux règles et contraintes de sécurité qu'il faut définir pour cela.

Considérons par exemple que trois réseaux de distribution d'énergie au niveau de trois pays différents, la France, l'Italie et l'Espagne, veuillent collaborer, il est alors nécessaire de définir des mécanismes pour assurer le contrôle d'accès localement dans les trois systèmes d'information et des mécanismes pour assurer la collaboration sécurisée entre eux. À une échelle généralement plus petite, on retrouve les mêmes besoins de contrôle d'accès et de collaboration au sein de chaque IIC.

1.2.3 Vulnérabilités et menaces dans les infrastructures d'information critiques

Nous nous intéressons dans cette section à énumérer certaines vulnérabilités et menaces importantes qui touchent les IIC [Sikich, 1998]. En général le système d'information souffre d'une architecture hétérogène de ses réseaux et d'un nombre de connexions considérables et non sécurisées avec des réseaux internes et externes. En particulier, le système d'information d'une infrastructure critique doit faire face à cinq types de menaces distinctes qu'on détaille dans le Tableau 3.

Type de vulnérabilités	Vulnérabilités
Défaillances organisationnelles	• Manque de ressources. • Manque d'effectifs de contrôle. • Insuffisance de la réglementation en ce qui concerne les responsabilités, le transfert de compétence, les règles de remplacement. • Insuffisance de circulation de l'information entre les différentes composantes. • Dépendances vis-à-vis d'autres organisations. • Sous-traitance. • Gestion des livraisons. • Mise en réseau (effet domino ou cascade). • Insuffisance de mise en œuvre et / ou de contrôle de la mise en œuvre des installations physiques/logiques.
Erreurs humaines	• Types d'erreurs: maladresses, mauvaise appréciation, comportement inapproprié, opérations erronées. • Causes d'erreur: surmenage, ignorance / incompétence, négligence, *social engineering*.
Défaillances techniques	• Perturbations ou pannes liées au système: erreurs de programmation, d'exécution. •Spécificités des composants sur étagère (COTS). •Perturbations lors de l'exécution de mises à jour.
Vulnérabilités induites par des attaques et menaces délibérées	• Groupe criminel, service de renseignement extérieur, *hackers* / intrus, guerre de l'information, menaces internes, terroristes, intérêts militaires, intérêts économiques. virus (vers, troyens).
Vulnérabilités induites par les systèmes d'information	• Logiciels et matériels complexes. • Logiciels non sûrs, problème de rustines (*patches*), de mise à jour. • Maliciels (virus, chevaux de Troie…), outils automatiques de scan ou d'attaque, etc. • Complexité de la mise en réseau des systèmes d'information. • Manque ou absence de sensibilisation aux problèmes de sécurité. • Manque de personnel qualifié. • Fréquentes restructurations et modifications. •Mauvaise administration des systèmes et réseaux. • Contexte d'urgence non prévu et non préparé.

Tableau 3 : Énumération des vulnérabilités des infrastructures d'information critique.

En raison des différentes vulnérabilités et menaces touchant les infrastructures d'information critiques, il est primordial de prévoir des moyens et des mesures pour protéger ce type d'infrastructures, c'est l'objet de la protection des infrastructures d'information critiques (CIIP en anglais). Cette branche de la sécurité donne des éléments sur la manière de répondre aux problèmes et exigences de sécurité du système d'information et de communication d'une infrastructure critique, tout en tenant compte des particularités qui le différencient d'un autre système d'information. Dans la section suivante, nous allons détailler trois projets internationaux qui développent des approches pour gérer la sécurité au sein d'infrastructures critiques. 1) Le projet IRRIIS, 2) le projet TCIP, 3) le projet Européen CRUTIAL dans lequel s'intègre notre travail.

1.3 Études et travaux existants

Afin de mettre en évidence l'état de l'art dans le domaine de la sécurité des infrastructures critiques en général et des infrastructures d'information critiques en particulier, nous décrivons dans cette section, trois travaux menés dans le cadre de projets de recherche visant à répondre à certains besoins et exigences d'une infrastructure critique, dont le projet CRUTIAL dans lequel s'intègre notre travail.

1.3.1 IRRIIS – Integrated Risk Reduction of Information-based Infrastructure Systems

Le projet européen IRRIIS[3] (Réduction des risques inhérents aux infrastructures de l'information) vise à augmenter la fiabilité, la viabilité et la résilience des IIC (voir Figure 8). Ses principaux objectifs sont de :

- déterminer un ensemble d'exigences de sécurité détaillées des IIC en se basant sur l'analyse de scénarios et de données concrètes.

- développer une technologie améliorée d'intergiciel (MIT, pour *Middleware Improved Technology*), c'est-à-dire une collection de composants logiciels qui visent à faciliter la communication entre les différentes infrastructures et les différents fournisseurs d'infrastructures. En gérant les actions de reprise et la stabilité en cas de situations critiques, les éléments du MIT doivent permettre de renforcer la sécurité des infrastructures critiques complexes.

- construire un environnement de simulation pour l'expérimentation contrôlée des IC, en mettant l'accent sur l'interdépendance des infrastructures. Le simulateur sera utilisé pour approfondir la compréhension des infrastructures critiques et de leurs interdépendances, pour identifier d'éventuels problèmes, développer des solutions appropriées, valider et tester les composants MIT, et enfin diffuser les concepts novateurs, les résultats et les produits à d'autres secteurs incluant des infrastructures d'information critiques.

[3] http://www.irriis.org/

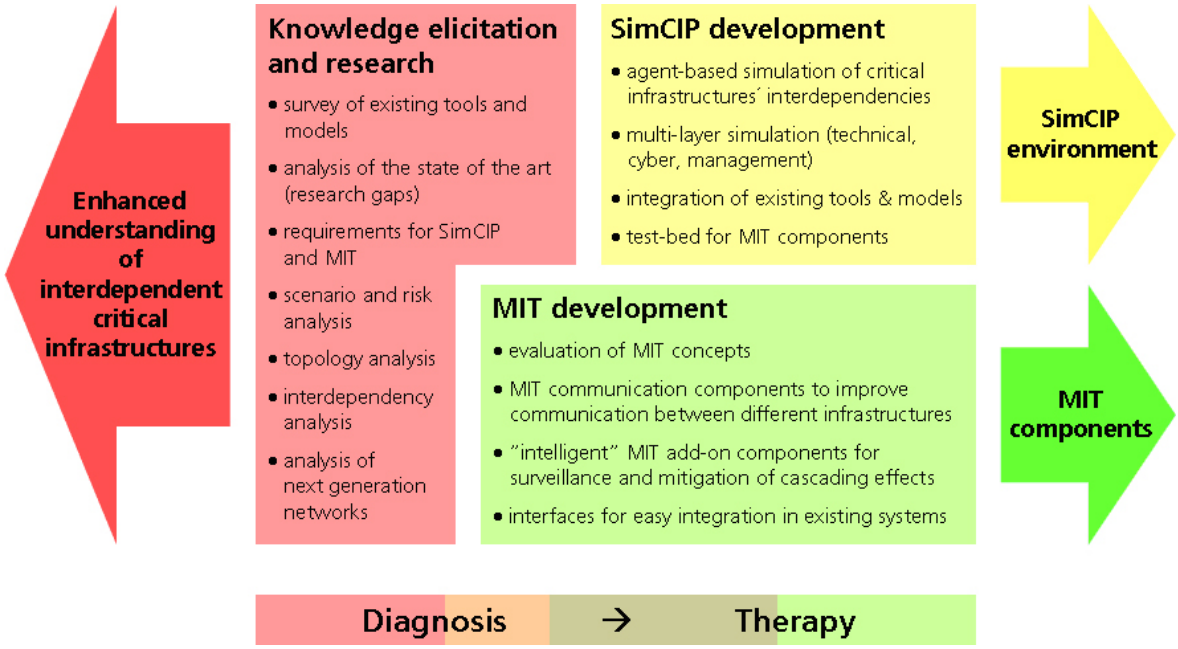


Figure 8 : Architecture générale du projet IRRIIS.

1.3.2 TCIP – Trustworthy Cyber Infrastructure for the Power Grid

Les chercheurs de l'Université de l'Illinois à Urbana-Champaign, du *Dartmouth College*, de la *Cornell University* et de l'Université de l'état de Washington proposent de protéger les réseaux d'énergie électrique américains, en améliorant le fonctionnement de ces réseaux et leur architecture afin de les rendre plus sûrs et fiables. Le projet TCIP[4] (Cyber-infrastructure de confiance pour le réseau électrique) est financé par la *National Science Foundation* avec le soutien du Département de l'Énergie (*Department of Energy*) et le Département de la Sécurité Nationale (*Department of Homeland Security*). La base de ce projet est qu'aujourd'hui la qualité de vie dépend de la continuité de la fourniture d'électricité, laquelle dépend à son tour du bon fonctionnement d'infrastructures d'information et de communication qui sont menacées par des cyber-attaques malveillantes ou des pannes accidentelles. Ces risques proviennent de cyber-pirates ayant un accès à des réseaux de contrôle, ou créant des attaques par déni de service sur les réseaux eux-mêmes, mais aussi de causes accidentelles, telles que les catastrophes naturelles ou les erreurs humaines.

L'essentiel de la recherche du projet TCIP est axé sur la sécurisation des composants de bas niveau des systèmes de communications et de gestion de données qui gèrent le réseau électrique, ce qui permet d'assurer un fonctionnement fiable, aussi bien dans des conditions normales, que sous la menace de cyber-attaques et/ou dans des situations d'urgence. L'impact de ces activités se ressent à tous les niveaux du projet : au niveau matériel, des fonctionnalités avancées sont proposées pour le contrôle de l'énergie, tout en assurant un contrôle d'accès approprié et en préservant la vie privée des clients. Une gestion du matériel a été développée pour fournir des mécanismes de détection et de recouvrement dans les composants électriques. De même, des processeurs ont été conçus pour effectuer efficacement les calculs

[4] <http://www.iti.uiuc.edu/content/tcip-trustworthy-cyber-infrastructure-power-grid>

cryptographiques afin de faciliter les communications entre les sous-stations et centres de contrôle du réseau électrique. Au niveau du réseau, des protocoles sont élaborés pour fournir des moyens efficaces, rapides et sécurisés pour traiter les données du système de contrôle, pour fournir en temps utile des données et assurer des processus dans les systèmes de contrôle et fournir les contrôles d'identité et d'accès ainsi que la négociation de la confiance à l'intérieur du réseau électrique. Ces protocoles sont en cours d'élaboration en prenant en compte la prochaine génération des exigences de communication et de contrôle, afin de fournir les éléments d'un réseau électrique plus robuste, sûr, rapide et adaptatif. De nouvelles techniques sont mises au point pour détecter, réagir et réparer les éléments endommagés après des cyber-attaques, tout en préservant l'intégrité, la disponibilité et le fonctionnement en temps réel. Enfin, un environnent combiné de simulation et de bancs d'essai a été développé pour analyser les scénarios d'un véritable réseau électrique et pour valider l'efficacité des réalisations mises en œuvre dans le cadre du projet. Ces innovations fourniront des orientations claires vers une nouvelle génération d''infrastructure d'information adaptée aux réseaux électriques, plus fiable, rapide et sécurisée, contribuant ainsi à fournir un fonctionnement continu de l'infrastructure électrique du pays.

1.3.3 CRUTIAL – CRitical UTility InfrastructurAL resilience

Le projet CRUTIAL[5] [Veríssimo *et al.*, 2008], [Abou El Kalam *et al.*, 2009b] examine les nouvelles technologies d'information et de communication connectées en réseau pour la gestion des réseaux de transport et de distribution d'énergie électrique. Dans ce genre d'infrastructure, les composants qui contrôlent les processus de transport d'électricité doivent être reliés à des infrastructures d'information, par le biais d'intranets, qui peuvent être à leur tour interconnectés par des réseaux à grande échelle (*Wide Area Network* ou WAN). L'approche novatrice de CRUTIAL réside dans la modélisation des infrastructures interdépendantes en tenant compte des multiples dimensions de l'interdépendance, pour réaliser de nouveaux modèles d'architecture, résistant aux défaillances accidentelles et aux attaques malveillantes [Garrone et al., 2007]. Les objectifs du projet sont les suivants:

- Proposition de modèles et d'architectures pour faire face aux menaces liées à l'ouverture, l'hétérogénéité et l'évolutivité qu'on demande aux infrastructures électriques.
- Analyse critique des scénarios dans lesquels des défaillances dans l'infrastructure d'information provoquent de graves répercussions sur l'infrastructure électrique elle-même.
- Proposition d'architectures distribuées fiables et résilientes permettant le contrôle et la gestion du réseau électrique.

Le projet CRUTIAL a visé plusieurs objectifs de différents niveaux : a) identifier et décrire des scénarios du système de contrôle, b) fournir des méthodes de modélisation pour comprendre et maîtriser les diverses interdépendances, c) développer un banc d'essai intégrant le réseau d'énergie électrique et l'infrastructure d'information, d) proposer des configurations et architectures tolérantes aux fautes, e) fournir les moyens qualitatifs et quantitatifs pour

[5] <http://crutial.cesiricerca.it/>

l'identification, l'analyse et l'évaluation des scénarios identifiés. Les résultats ont été validés sur des bancs d'essai des réseaux électriques. La Figure 9 présente l'architecture spécifique des CIS (Crutial Information Switch) qui comptent parmi les composants « clé » de l'architecture CRUTIAL.

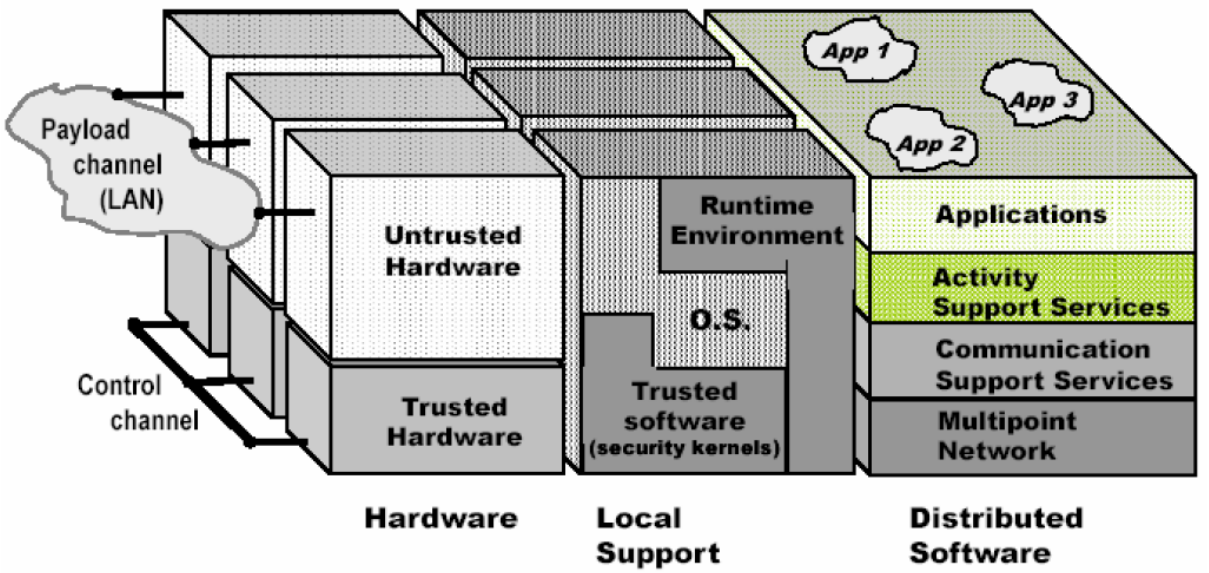


Figure 9 : Architecture des nœuds CRUTIAL.

L'utilisation des CIS sera détaillée dans le chapitre 4. L'architecture complète des CIS est détaillée dans le déliverable D4 « *Preliminary Architecture Specification* » réalisé dans le cadre du projet CRUTIAL [Abou El Kalam et al., 2007b].

Les résultats du projet contribueront à la conception et l'évaluation de nouveaux systèmes d'énergie électrique et infrastructures de l'information. Ainsi, ils permettront de réduire les pannes électriques répétitives, en termes de fréquence, de durée et d'étendue, et de fournir une meilleure vision des réseaux de transport et de distribution d'énergie électrique pour améliorer la résilience de ce type d'infrastructure.

Dans le cadre de ce projet, nous avons développé un cadre conceptuel pour prendre en compte les exigences de sécurité des infrastructures critiques en général et pour les infrastructures d'information critique en particulier. C'est ce que ce manuscrit va détailler, en commençant par définir les caractéristiques et les besoins en sécurité des infrastructures critiques et des infrastructures d'information critiques, ensuite en récapitulant l'état de l'art des modèles de contrôles d'accès existants, en proposant une schéma-cadre de sécurité qui répond aux besoins précédemment énumérés, en l'appliquant sur l'étude de cas pratique d'une infrastructure d'énergie électrique, et enfin en illustrant l'utilisation de notre schéma-cadre de sécurité sur un scénario réaliste implémenté sur un prototype de système d'information et de communication contrôlant une simulation d'une infrastructure électrique.

1.4 Conclusion du chapitre 1

Dans ce premier chapitre, nous avons introduit les notions d'infrastructure critique et infrastructure d'information critique afin d'identifier nos besoins en matière de sécurité. Nous avons ensuite énuméré les vulnérabilités les plus importantes qui touchent les infrastructures critiques et les infrastructures d'information critiques.

Nous avons également présenté deux projets de recherche qui proposent quelques moyens pour se protéger vis-à-vis des vulnérabilités précédemment décrites, en proposant des approches de sécurité dans les infrastructures d'information critiques. Dans la description de ces projets, nous avons mis l'accent sur le but, l'organisation et les contributions de chaque projet. Enfin, nous avons présenté le projet CRUTIAL, dans lequel s'intègre notre travail.

Le prochain chapitre donne un état de l'art des modèles de contrôles d'accès traditionnels et collaboratifs, en décrivant le fonctionnement, les avantages et les contributions de chaque modèle et en analysant comment ces modèles peuvent répondre ou non aux besoins de collaboration et de contrôle d'accès que nous avons énumérés précédemment.

Chapitre 2. Modèles et politiques de sécurité

Dans le chapitre précédent, nous avons présenté la problématique des infrastructures critiques et des infrastructures d'information critiques qui leur sont liées ; en particulier, nous nous sommes intéressés aux besoins en sécurité de chaque type d'infrastructure. Ensuite, nous avons décrit les différentes vulnérabilités et menaces touchant particulièrement les infrastructures d'information critiques et nous avons montré comment améliorer leur sécurité grâce à la protection des infrastructures d'information critiques. Puis, nous avons présenté certains travaux qui essaient de répondre aux besoins de sécurité d'une infrastructure d'information critique, en tenant compte des différentes vulnérabilités énumérées auparavant. Pour conclure, nous avons présenté le projet européen CRUTIAL, qui traite de la sécurité des infrastructures critiques, et dans lequel s'intègre cette thèse. Nous avons conclu par la nécessité de traiter deux points importants de la sécurité : la collaboration sécurisée et le contrôle d'accès.

Dans ce chapitre, il s'agit d'analyser l'état de l'art en ce qui concerne les modèles de contrôle d'accès et politiques de sécurité. Nous discuterons les limites et les avantages des différents modèles, le but étant de déterminer quel modèle pourrait convenir à nos exigences de contrôle d'accès et de collaboration pour créer un cadre conceptuel de contrôle d'accès adapté au contexte des infrastructures critiques et qui réponde aux besoins décrits dans le chapitre précédent.

2.1 Politiques et modèles de sécurité

La politique de sécurité organisationnelle est définie dans les Critères Communs (*Common Criteria)* [CC, 2006] comme un ensemble de règles, procédures, codes de conduite ou lignes directrices de sécurité qu'une organisation impose pour son fonctionnement.

Une politique de sécurité est spécifiée par :

- des objectifs de sécurité qui doivent être satisfaits, par exemple : "les soumissions à des appels d'offres doivent rester secrètes (c'est-à-dire, ne doivent pas être divulguées à des organismes concurrents)";
- des règles exprimant la manière dont le système peut évoluer de manière sécurisée, par exemple : "l'organisation propriétaire d'une ressource (par exemple, un fichier) est autorisée à accorder des droits d'accès sur la ressource à d'autres organisations qui coopèrent avec elle".

Une politique de sécurité ne garantit pas, à elle-seule, la sécurité et le bon fonctionnement du système. Pour en vérifier la cohérence, la politique de sécurité est généralement associée à un modèle formel, qui permet d'abstraire la politique de sécurité, de gérer sa complexité, de détecter et de résoudre les situations conflictuelles et de vérifier que tous les objectifs de sécurité sont couverts par les mesures préalablement identifiées. En outre, la politique de sécurité doit être mise en œuvre par les mécanismes de sécurité appropriés, par exemple, des listes de contrôle d'accès (ou ACL, pour *Access Control Lists*) [Harrison *et al.*, 1976], des règles de filtrage au niveau des pare-feux, des moyens cryptographiques, etc. Dans la pratique, ces mécanismes peuvent être sélectionnés (par exemple, au moyen d'un produit commercial sur étagère) et configurés, ou spécifiés, développés et mis en œuvre spécifiquement, si aucun produit existant ne satisfait totalement les objectifs de la politique ou n'implémente complètement les règles de la politique.

La plupart des politiques de sécurité ne sont en fait que des politiques d'*autorisation*. L'autorisation vise à permettre aux utilisateurs enregistrés de réaliser toutes les actions légitimes, tout en empêchant que des utilisateurs non-enregistrés mènent quelle qu'action que ce soit dans le système, et des utilisateurs enregistrés réalisent des actions qui ne sont pas légitimes. Les politiques d'autorisation sont le plus souvent mises en œuvre par les mécanismes de *contrôle d'accès* présents dans la plupart des systèmes informatiques ou dans les équipements de réseaux (routeurs, pare-feux, etc.). Mais bien souvent il est important que la politique de sécurité contienne aussi des règles d'obligation, en plus des règles d'autorisation et d'interdiction, et rares sont les produits existants capables d'implémenter des règles d'obligation.

Un modèle de sécurité peut être défini comme étant un formalisme permettant de représenter la politique de sécurité, de l'abstraire, d'en réduire la complexité, et d'aider à en vérifier la complétude (c'est-à-dire que les propriétés satisfont toutes les exigences), la cohérence (c'est-à-dire que les règles sont suffisantes pour satisfaire les objectifs) et la conformité (c'est-à-dire que les mécanismes mis en œuvre implémentent les règles). De même que la plupart des politiques de sécurité ne sont en général que des politiques d'autorisation, la plupart des modèles de sécurité ne sont que des *modèles de contrôle d'accès*.

Les systèmes d'information actuels sont de plus en plus ouverts, distribués et multi-organisationnels, et l'aspect de collaboration représente un point important à prendre en compte. La collaboration, dans les systèmes d'information, permet à un système de bénéficier des ressources d'autres systèmes [Dacey, 2004], [Brock, 2000]. Le contrôle d'accès est alors une question importante dans les systèmes collaboratifs pour protéger la confidentialité et l'intégrité des données tout en respectant l'autonomie des systèmes collaborant. De nombreuses approches existent, qui essaient de traiter les différents aspects du contrôle d'accès collaboratif.

Nous classons les différents modèles de sécurité selon deux catégories principales : les modèles de sécurité traditionnels, conçus pour gérer la sécurité d'une organisation, et les modèles de sécurité collaboratifs, conçus pour gérer la sécurité d'un ensemble d'organisations qui doivent coopérer, comme c'est le cas pour les IIC que nous avons présentées dans le premier chapitre. Dans cette section, nous avons choisi de nous focaliser sur certains modèles de contrôle collaboratifs tels que Multi-OrBAC [Abou El Kalam & Deswarte, 2006], OV [Nasser *et al.*, 2005] et O2O [Coma, 2006], [Cuppens et al., 2006], et sur certains modèles de contrôle d'accès traditionnels tels que RBAC [Ferraiolo & Kuhn, 1992] et OrBAC [Abou El Kalam et al., 2003] sur lesquels sont basés les modèles collaboratifs que nous avons choisi d'étudier.

2.2 Modèles de contrôle d'accès traditionnels

Dans la prochaine section, nous allons décrire certains modèles de contrôle d'accès existants dont nous avons besoin pour comprendre la problématique du contrôle d'accès. Les modèles présentés Multi-OrBAC et O2O se basent respectivement sur les modèles de contrôle d'accès RBAC pour le premier et OrBAC pour les deux autres. Ainsi, pour permettre la bonne compréhension des modèles de contrôle d'accès collaboratifs présentés, nous commençons d'abord par la description des modèles RBAC et OrBAC.

2.2.1 Modèle de contrôle d'accès basé sur les rôles (RBAC)

RBAC (pour *Role-Based Acces Control* [Ferraiolo & Kuhn, 1992], [Sandhu *et al.*, 1996]) est un modèle de contrôle d'accès basé sur les *rôles*. Un rôle représente de façon abstraite une fonction identifiée dans l'organisation (par exemple, chef de service, ingénieur d'étude, etc.). À chaque rôle, on associe des permissions (représentées par des droits d'accès). Une permission est un ensemble de droits correspondant aux tâches qui peuvent être effectuées par un rôle. RBAC définit quels utilisateurs ont accès aux ressources en fonction des rôles qui leur sont assignés, et l'accès aux ressources est limité aux utilisateurs auxquels on a attribué un rôle qui permet d'accéder à ces ressources. Chaque utilisateur se voit attribuer un ou plusieurs rôles et une ou plusieurs permissions sont attribuées à chaque rôle. Ainsi, dans RBAC, la politique de contrôle d'accès ne s'applique pas directement aux utilisateurs comme dans les précédents modèles de contrôle d'accès (DAC [Harrison & Ullman 1976], ou MAC [TCSEC, 1985]), les permissions ne sont plus associées de manière directe aux sujets, mais par le biais de rôles, qui regroupent des sujets qui remplissent les mêmes fonctions.

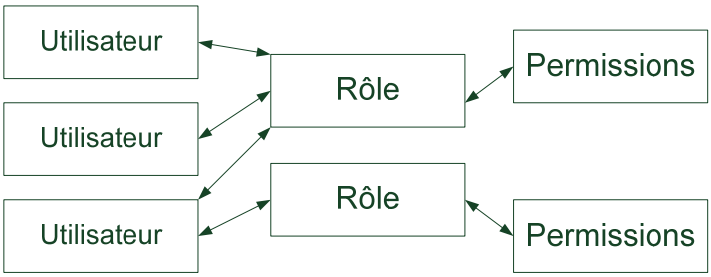


Figure 10 : Positionnement de la notion de rôle dans le fonctionnement de RBAC.

L'un des avantages de RBAC est qu'il n'est pas nécessaire de mettre à jour l'ensemble de la politique de contrôle d'accès si un nouveau sujet est créé, il suffit juste d'assigner un rôle à ce sujet. Les politiques basées sur les rôles visent donc à faciliter l'administration de la sécurité.

Le modèle entité-association décrit par la Figure 11 présente deux relations, « *détient* (*rôle, permission*) » et « *joue* (*sujet, rôle*) », qui définissent précisément les permissions accordées à chaque sujet. Un rôle peut détenir plusieurs permissions et une même permission peut être détenue par plusieurs rôles. De même, un sujet peut jouer plusieurs rôles et, inversement, un rôle peut être joué par plusieurs sujets. Ainsi, si le docteur Dupont est à la fois chirurgien et directeur de l'hôpital, en tant que chirurgien il aura le droit d'accès aux dossiers médicaux, alors qu'en tant que directeur il pourra accéder aux informations administratives.

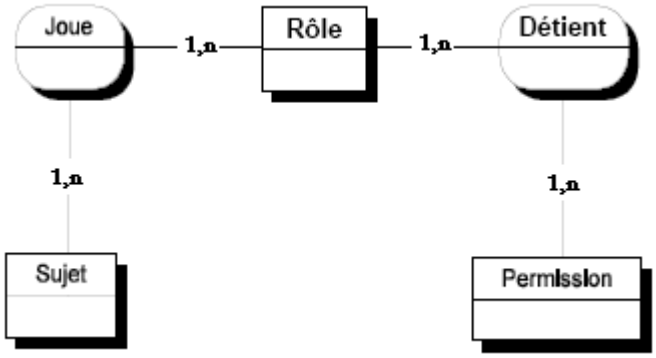


Figure 11 : La relation « détient » et la relation « joue »

La façon la plus simple de définir les rôles au sein d'une organisation consiste à identifier pour chaque rôle les tâches nécessaires à la fonction représentée par le rôle, à identifier les objets que ces tâches utilisent, puis à définir les droits d'accès nécessaires pour réaliser ces tâches sur ces objets, c'est-à-dire les permissions (ensemble de droits, objets), et finalement, à associer ces permissions aux rôles. L'affectation des sujets aux rôles est une tâche à faire séparément et probablement par d'autres administrateurs.

4.2.1.1 Avantages du modèle RBAC

L'analyse des politiques basées sur les rôles permet de conclure qu'elles sont relativement faciles à administrer et suffisamment souples pour s'adapter à chaque organisation. En effet, la définition des rôles reflète la structure de l'organisation. Ainsi avec RBAC, il est facile d'ajouter un utilisateur : il suffit de lui assigner les rôles qu'il doit jouer en fonction des fonctions qu'il

exerce dans l'organisation. De même, il est relativement facile de faire évoluer les tâches suite à la création ou la modification d'un objet : il suffit de mettre à jour les privilèges des rôles concernés. RBAC permet donc de réduire les erreurs d'administration de la politique de sécurité et de maîtriser la complexité de représentation et de gestion des droits d'accès, à l'aide des deux relations présentées ci-dessus : la relation « *détient* » associant les droits aux rôles et la relation « *joue* » associant des rôles à des utilisateurs. Les instances de la première relation restent relativement stables dans le temps et peuvent donc être définies par l'administrateur responsable de la sécurité. À l'inverse, les instances de la deuxième relation changent fréquemment et sont souvent définies par un autre employé de l'organisation, par exemple un opérateur habilité. Le passage par cette entité intermédiaire, le rôle, réduit considérablement les erreurs d'administration. RBAC contribue également à maîtriser la complexité de la gestion des droits d'accès, grâce au mécanisme d'héritage entre les rôles : les rôles peuvent être structurés de façon hiérarchique, un (sous-)rôle héritant des permissions du rôle dont il dépend hiérarchiquement. Par exemple, le rôle *chirurgien* possède toutes les permissions du rôle *médecin*.

2.2.1.1 Inconvénients du modèle RBAC

Le principal inconvénient de RBAC réside dans la difficulté de gérer et d'implémenter des règles du type « *seul le médecin traitant peut lire les informations médicales du dossier d'un patient* ». Pour résoudre ce problème avec RBAC, il faut soit créer autant de rôles « *médecin traitant du patient X* » que de patients, soit mettre en œuvre des règles supplémentaires dans l'application (par exemple, la gestion de la base de données des dossiers médicaux), règles qui ne sont pas exprimables dans le modèle RBAC. Le modèle RBAC présente aussi d'autres inconvénients. Tout d'abord, le concept de permission est primitif. En effet, dans le modèle RBAC, rien n'est dit sur l'usage ou la structure des permissions, considérant qu'ils dépendent de la mise en œuvre concrète du modèle. Il serait préférable d'ajouter au modèle une structure générique de permission. Par ailleurs, le concept de hiérarchie de rôles est quelque peu ambigu : la hiérarchie de rôles ne correspond généralement pas à la hiérarchie organisationnelle. Par exemple, directeur d'hôpital est une fonction hiérarchique supérieure à celle de simple médecin, et pour autant, un directeur d'hôpital qui n'est pas médecin n'a pas les permissions associées au rôle médecin. Il faut aussi constater que la distinction entre le concept de rôle et celui de groupe est généralement floue. Enfin, il difficile d'appliquer le modèle RBAC à la définition d'une politique de sécurité d'un système comportant plusieurs organisations qui coopèrent.

2.2.2 Modèle de contrôle d'accès basé organisation (OrBAC)

Le modèle OrBAC [Abou El Kalam et al., 2003] (*Organization-based Access Control*) est basé sur les mêmes principes que son prédécesseur RBAC, en intégrant de nouvelles notions. L'idée principale est d'exprimer la politique de sécurité avec des entités abstraites et de séparer complètement la représentation de la politique de sécurité de son implémentation. Le modèle OrBAC, est centré sur le concept d'organisation (une organisation est un groupe structuré d'entités actives), et tous les autres concepts d'OrBAC sont définis par rapport à l'organisation. "*La clinique privée du Languedoc*", "*le service des urgences de l'hôpital Purpan*", sont des exemples d'organisation.

OrBAC structure les *sujets*, les *objets* et les *actions* respectivement en *rôles* (comme dans RBAC), *vues* (notion proche de celle utilisée dans VBAC [Gabillon, 2004], [Bertino & Mesiti 2000], [Lentzner, 2004]) et *activités* (notion proche de celle utilisée dans TBAC [Thomas & Sandhu 1997]). De même qu'un rôle est une représentation abstraite d'un groupe d'utilisateurs exerçant une fonction dans une organisation, une activité représente une ou plusieurs actions et une vue un ou plusieurs objets. OrBAC définit aussi une notion de contexte comme une situation spécifique qui conditionne la validité d'une règle. On distingue donc deux niveaux dans OrBAC : un niveau abstrait dans lequel l'administrateur définit la politique de sécurité par des règles sur les entités abstraites (rôles, activités, vues) sans s'inquiéter de la façon dont l'organisation implémente ces entités, et un niveau concret où des entités actives (typiquement des processus informatiques) exécutent des actions sur des objets, sous le contrôle de mécanismes de protection qui mettent en œuvre les règles définies dans la politique. OrBAC bénéficie aussi d'une représentation formelle, qui permet en particulier de détecter de façon statique les conflits entre les règles [Cuppens *et al.*, 2007].

2.2.2.1 Concept de rôle et relation Habilite ()

Dans OrBAC, le concept de *rôle* permet de modéliser précisément comment l'organisation habilite les *sujets*. Un sujet est soit une entité active, c'est-à-dire un utilisateur, soit une organisation. Par exemple, "Jean", "Marie", etc., peuvent être des sujets, tout comme l'organisation "*département comptable de la clinique privée du Languedoc*", etc.

Dans le domaine médical, les rôles "*cardiologue*", "*infirmière*" ou "*médecin*", sont joués par des utilisateurs alors que les rôles "*service des urgences*" ou "*unité des soins intensifs*" sont joués par des organisations. En utilisant le concept de rôle, nous disposons d'un moyen efficace pour structurer les sujets et mettre à jour les politiques de sécurité lorsque de nouveaux sujets sont insérés dans le système. Comme les sujets jouent des rôles dans des organisations, nous introduisons une relation entre ces entités : la relation « *Habilite* ».

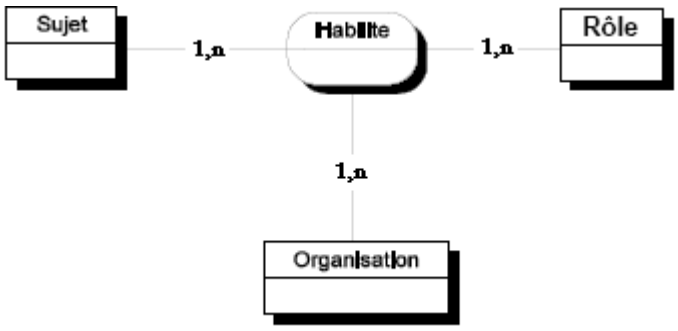


Figure 12 : La relation « Habilite ».

Les deux exemples suivants illustrent le fait que les sujets sont soit des utilisateurs, soit des organisations :

- Habilite(Purpan, Jean, cardiologue) signifie que l'hôpital Purpan habilite Jean dans le rôle cardiologue, et

- Habilite(Rangueil, ICU31, unité_de_soins_intensifs) signifie que l'hôpital Rangueil habilite l'unité ICU31 dans le rôle d'unité de soins intensifs.

2.2.2.2 Concept de vue et relation Utilise()

Le concept de *vue* [Gabillon, 2004], [Bertino & Mesiti 2000] permet de modéliser de quelle façon l'organisation utilise les objets, un objet étant une entité non active, comme un fichier, un message électronique, un formulaire imprimé, etc. Dans le domaine médical, les objets peuvent être des dossiers administratifs, les dossiers médicaux et les dossiers chirurgicaux des patients. La vue sert à structurer les objets : une vue correspond à un ensemble d'objets qui satisfont une même propriété. Par exemple dans un système d'informations administratives d'un hôpital, la vue "dossier administratif" correspond à l'ensemble des dossiers administratifs des patients, quelle que soit leur forme (tuple dans une base de données, fichier de texte, ou document XML). De même, la vue "dossier médical" correspond aux dossiers médicaux des patients. Dans la mesure où les vues caractérisent la manière dont les objets sont utilisés dans l'organisation, nous avons besoin d'une relation qui lie ces trois concepts : la relation Utilise. Si *org* est une organisation, *o* est un objet et *v* est une vue, alors Utilise(*org*, *o*, *v*) signifie que *org* utilise l'objet *o* dans la vue *v*.

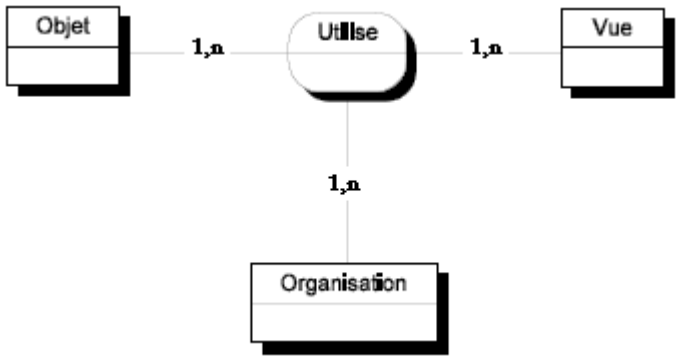


Figure 13 : La relation « Utilise ».

Ainsi une même vue peut être définie différemment suivant l'organisation considérée. La vue "dossier médical" peut être définie à l'hôpital Purpan comme un ensemble de documents Word et comme un ensemble de documents Latex à l'hôpital Rangueil :

- Utilise(Purpan, F31.doc, dossier_médical) signifie que l'hôpital Purpan utilise le fichier F31.doc comme un dossier médical, et
- Utilise(Rangueil, F32.tex, dossier_médical) signifie que l'hôpital Rangueil utilise le fichier F32.tex comme un dossier médical.

2.2.2.3 Concept d'activité et relation Considère()

Le concept d'*activité* permet de modéliser de quelle façon l'organisation réalise des *actions*, une même activité pouvant être implémentée de différentes façons dans une même organisation. Typiquement, une action peut correspondre des actions informatiques comme "lire", "écrire", "envoyer", etc. De la même manière que les rôles et les vues sont des abstractions

des sujets et des objets, l'activité est définie comme une abstraction des actions, servant à les structurer : une activité correspond à des actions qui ont un objectif commun. En considérant que les rôles regroupent les sujets qui remplissent les mêmes fonctions et les vues correspondent à des ensembles d'objets qui satisfont une même propriété, les activités associent les actions qui réalisent la même opération. "Consulter", "modifier", "transmettre", sont des exemples d'activités. La relation « *Considère* » sera utilisée pour associer les entités Organisation, Action et Activité. Plus précisément, si *org* est une organisation, *a* est une action et *A* est une activité, alors Considère(org, a, A) signifie que l'organisation *org* considère l'action *a* comme faisant partie de l'activité *A*.

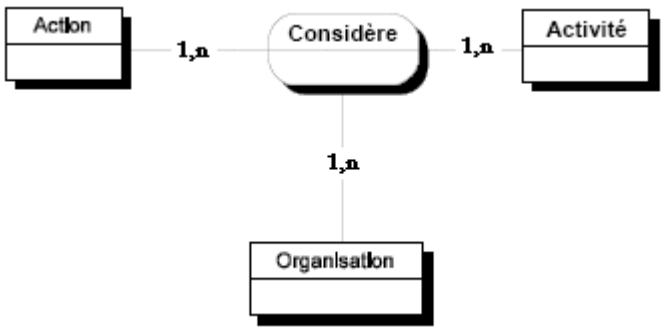


Figure 14 : La relation « Considère ».

Selon la façon dont sont implémentées les vues (par exemple, fichier de texte ou tuple dans une base de données), l'activité pourra correspondre à différentes actions (par exemple, lecture sur un fichier ou sélection dans une base de données). Ainsi, l'activité de consultation d'un dossier médical peut correspondre dans l'organisation "hôpital Purpan" à l'action "lire" un fichier, mais peut tout aussi bien correspondre à l'action "select" sur une base de données dans l'hôpital Rangueil :

- Considère(Purpan, lire, consultation) signifie que l'hôpital Purpan considère "lire" comme une action de l'activité "consultation", et
- Considère(Rangueil, select, consultation) signifie que l'hôpital Rangueil considère "select" comme une action de consultation.

2.2.2.4 Concept de contexte et relation Définit()

Le contexte [Cuppens & Miège 2003] peut être défini comme toute information qui caractérise la situation d'une entité ou qui spécifie les circonstances concrètes dans lesquelles les organisations accordent des permissions à des rôles pour réaliser des activités sur des vues. Par exemple : quand l'utilisateur a-t-il le droit d'accéder à une information ? d'où l'accès est-il possible ? Dans le domaine médical, par exemple, le contexte permet d'exprimer des notions comme : l'urgence, les processus de soins habituels, l'exclusion mutuelle, les attributs temporels, etc. D'une manière générale, le contexte peut être défini comme un attribut d'une classe, ou comme une classe associée à une organisation.

La relation *Définit* permet de modéliser de quelle manière l'organisation définit des contextes dans lesquels des utilisateurs réalisent des actions sur des objets. Un contexte [Cuppens & Miège 2003] spécifie des circonstances dans lesquelles les organisations accordent des permissions (ou interdictions) de réaliser des activités sur des vues. Il faut noter que le contexte permet facilement d'exprimer des caractéristiques comme "médecin traitant", ce que le modèle RBAC ne permettait pas.

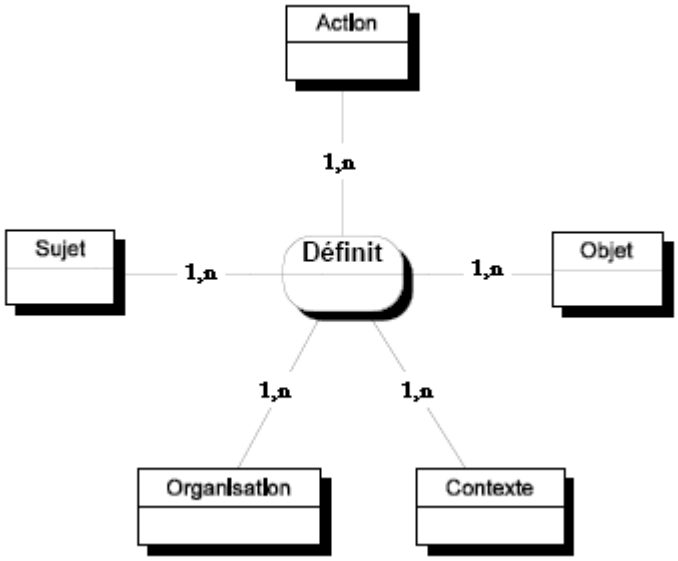


Figure 15 : La relation « Définit ».

Les entités organisation, sujet, objet, action et contexte sont liées par une nouvelle relation appelée « *Définit* ». Considérons par exemple les deux faits suivants : Définit(Purpan, Jean, lire, F31.doc, urgence) et Définit(Rangueil, Marie, lire, F32.tex, médecin_traitant). Si le premier fait est vrai, alors Jean n'a pas besoin d'être le médecin traitant du patient pour consulter son dossier médical F31.doc. Si le second fait est vrai, alors Marie doit être le médecin traitant du patient pour lire le dossier médical F32.tex. Cela signifie que, sauf en cas d'urgence, les médecins ne peuvent consulter que les dossiers médicaux de leurs patients.

2.2.2.5 Expression de politiques de sécurité dans le modèle OrBAC

Une autre caractéristique d'OrBAC est que les règles exprimées dans ce modèle peuvent définir des permissions, des interdictions, des obligations et des recommandations. Ce modèle est donc beaucoup plus puissant qu'un simple modèle de contrôle d'accès. Ces règles sont de la forme Permission|Interdiction|Obligation|Recommandation(*org ; r ; v ; a ; c*), où *org* est une organisation, *r* un rôle, *v* une vue, *a* une activité et *c* un contexte. La dérivation des droits d'accès ou des obligations (c'est-à-dire l'instanciation des règles de sécurité) peut être formellement exprimée comme suit :

org ∈ Org, s ∈ S, α ∈ Actions, o ∈ O, r ∈ R, a ∈ Activités, v ∈ V, c ∈ C,
Permission (org, r, v, a, c) ∧
Habilite (org, s, r) ∧
Considère (org, α, a) ∧
Utilise (org, o, v) ∧
Définit (org, s, α, o, c)
→ Est_permis(s, α, o)

Tableau 4 : Définition d'une permission avec le formalisme OrBAC.

Cette formule s'interprète de la façon suivante : si dans l'organisation '*org*', le rôle '*r*' est autorisé à effectuer l'activité '*a*' sur la vue '*v*' quand le contexte '*c*' est vrai, et si dans l'organisation '*org*', le rôle '*r*' est assigné au sujet '*s*', et si dans l'organisation '*org*', l'action 'α' fait partie de l'activité '*a*', et si dans l'organisation '*org*', l'objet '*o*' fait partie de la vue *v*, et si le contexte '*c*' est vrai pour le quadruplet (*org*, *s*, α, *o*), alors le sujet '*s*' a le droit d'exécuter l'action 'α' sur l'objet '*o*'.

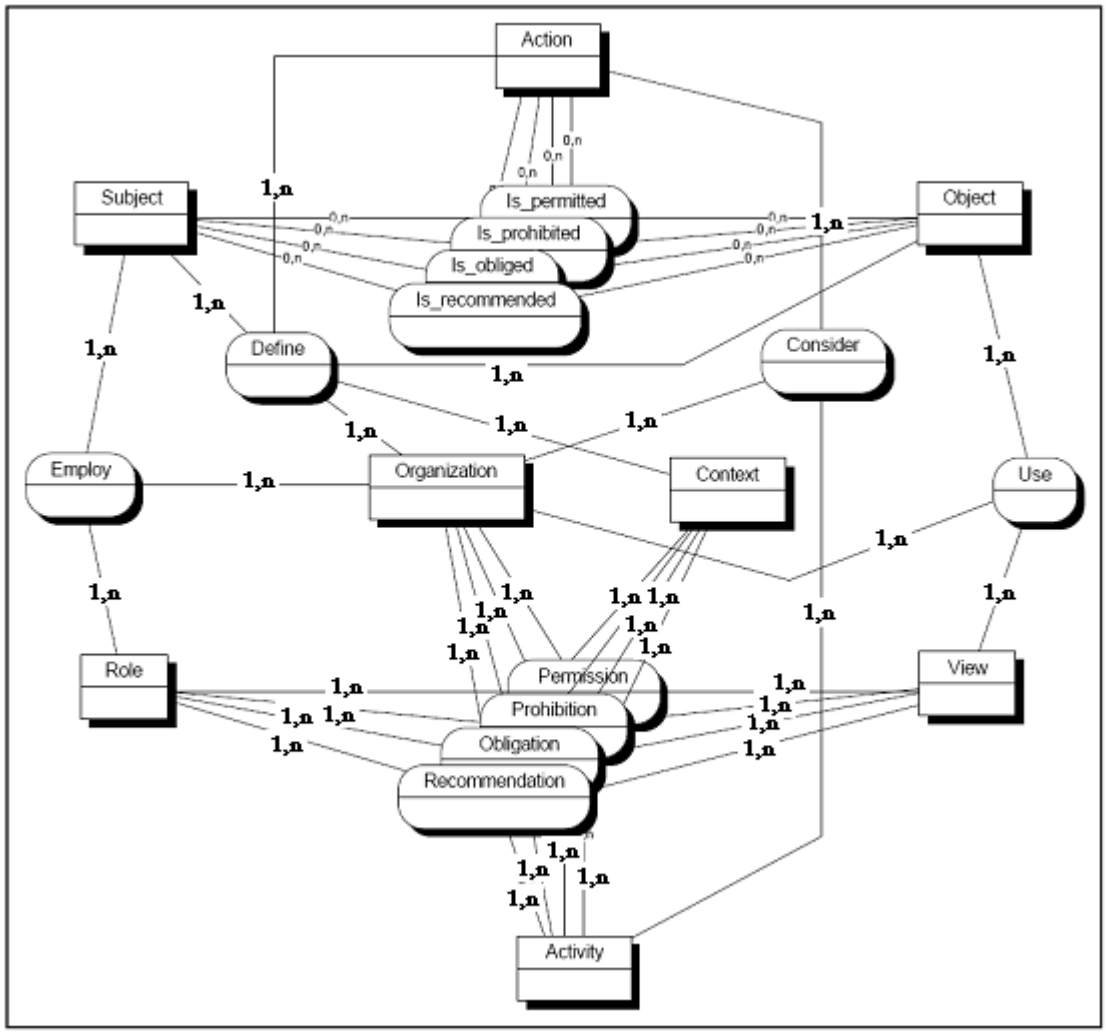


Figure 16 : Structure du modèle OrBAC [Abou El Kalam *et al.*, 2003].

Rappelons-le, une politique de sécurité réglemente les accès au système à travers des permissions, des interdictions, des obligations et des recommandations. Elle spécifie les accès autorisés aux entités passives par des entités actives et régule les actions opérées sur le système. Au niveau concret (c'est-à-dire dans le système informatique), lorsqu'un processus (s'exécutant pour le compte d'un sujet) réalise une action sur un objet, cet accès est contrôlé par des mécanismes de protection qui autorisent, interdisent ou déclenchent automatiquement (dans le cas d'une obligation) l'accès en fonction de règles de permission, d'interdiction, d'obligation ou de recommandation qui sont définies dans la politique (au niveau abstrait) à condition qu'elles soient valides dans le contexte courant (évalué au niveau concret).

Ainsi, en nous appuyant sur les différents concepts et relations du modèle OrBAC, nous pouvons exprimer la politique de sécurité d'une organisation comme un ensemble de permissions, d'interdictions, d'obligations et de recommandations [Miège, 2005]. Mais ceci ne suffit pas lorsqu'on considère plusieurs organisations qui doivent coopérer [Dacey, 2004], [Brock, 2000]. Nous nous intéressons donc à un état de l'art des modèles de sécurité pour la collaboration et l'interopérabilité.

Notons seulement qu'OrBAC, de par sa définition récursive du concept d'organisation[6], il est aisé de représenter une hiérarchie d'organisations qui peuvent coopérer entre elles. Ainsi, pour permettre la collaboration entre organisations différentes, il suffit d'ajouter une super organisation virtuelle, au sommet d'une hiérarchie contenant toutes les autres. Cette super-organisation pourrait ainsi définir sa politique de sécurité en OrBAC, et l'imposer à toutes les organisations. Mais bien sûr ceci serait aux dépens de principe d'autonomie que nous avons mis en avant dans le premier chapitre.

2.3 Modèles de contrôle d'accès pour la collaboration

Avant de présenter certains modèles de gestion de la sécurité pour les systèmes collaboratifs, il faut distinguer deux approches pour gérer la sécurité de tels systèmes : une gestion centralisée et une gestion décentralisée.

Les modèles de sécurité pour le contrôle d'accès collaboratif peuvent être classés en deux catégories principales, selon que le contrôle d'accès est centralisé ou qu'il est réalisé dans chaque organisation vis-à-vis des autres, c'est-à-dire de pair à pair (voir Figure 18). Dans les modèles de contrôle d'accès centralisé, toutes les décisions pour savoir si un utilisateur d'une organisation *A* est en mesure d'accéder à un objet donné *O* dans une organisation *B*, sont prises et mises en œuvre en un point central qui peut être une *super-organisation* qui impose sa politique de sécurité à toutes les autres organisations. Dans le contrôle d'accès pair à pair, chaque organisation est responsable de ses propres décisions de contrôle d'accès et de leur mise en œuvre.

[6] Une organisation est un ensemble de sujets, un sujet étant soit un utilisateur soit une organisation

2.3.1 Gestion centralisée de la sécurité

La première approche principale pour définir la sécurité d'organisations qui collaborent consiste à imposer une politique de sécurité globale et centralisée [de Capitani & Samarati, 1996], comme schématisé dans la Figure 17.



Figure 17 : Collaboration à travers la notion de super-organisation.

En 2008, Lin, Rao et Bertino [Lin *et al.*, 2008] ont proposé un modèle de sécurité pour le contrôle d'accès collaboratif selon cette première approche. L'architecture de sécurité de référence est celle de XACML [OASIS, 2003]. L'idée principale est basée sur la décomposition de la politique de sécurité globale appliquée par l'ensemble des organisations collaborant, sans pour autant compromettre les besoins en autonomie et la confidentialité de chacune. Il faut pour cela intégrer dans la politique de sécurité de chacune des organisations certains composants de la politique globale. La mise en œuvre de cette politique consiste alors à évaluer de façon efficace les requêtes provenant des différentes organisations, tout en garantissant la cohérence des décisions de contrôle d'accès. Cette étude est très intéressante, mais elle suppose qu'il est possible de définir une politique de contrôle d'accès globale appliquée sur un environnement collaboratif. Si les organisations qui acceptent de collaborer sont mutuellement méfiantes, une telle politique ne peut être définie que par une *autorité* reconnue par l'ensemble de ces organisations (éventuellement constituée de façon ad hoc par ces organisations). Une telle autorité joue alors le rôle de la super organisation, et la politique de sécurité globale qu'elle définit doit tenir compte des structures internes (ressources, rôles, etc.) de chaque organisation, ce qui implique une divulgation d'information contraire à la confidentialité que souhaitent les organisations participantes. D'autre part, établir une telle autorité dans des infrastructures s'étendant sur plusieurs pays, avec des législations et des standards différents, peut poser des problèmes insurmontables. De plus, il peut être très difficile d'intégrer dans la politique de sécurité d'une organisation des composants de la politique globale d'une infrastructure quand cette organisation participe à d'autres infrastructures, qui elles-mêmes peuvent imposer chacune sa propre politique globale.

Une variante de cette approche centralisée consiste à proposer une plateforme unifiée pour appliquer une multitude de politiques de contrôle d'accès au sein d'un seul et même système central [Jajodia *et al.*, 2001], [Bertino *et al.*, 1996]. Pour cela, ces études proposent de décrire les politiques de sécurité dans un langage qui permette de spécifier des autorisations positives (permissions) et négatives (interdictions), d'incorporer des moyens pour détecter et résoudre les conflits, et de définir une stratégie de décision qui puisse porter sur différents niveaux (utilisateurs, groupes, objets, ou rôles), selon les besoins. L'avantage principal de cette technique est de permettre de spécifier de façon unifiée les différentes politiques de contrôle

d'accès qui peuvent coexister au sein d'un même système et être appliquées par le même mécanisme de décision. Néanmoins, cette plateforme n'apporte aucune structuration dans les interactions entre organisations, ce qui conduit à de grands risques d'incohérence entre les politiques de sécurité dès qu'on l'applique à des systèmes de taille réaliste. Enfin, la décision de contrôle d'accès étant prise par un mécanisme centralisé, ce mécanisme joue le rôle d'une « super organisation », avec tous les inconvénients cité précédemment, ajoutés au fait qu'il constitue un point unique de défaillance vis-à-vis de la disponibilité et de la sécurité.

2.3.2 Gestion décentralisée de la sécurité

La deuxième approche pour définir la sécurité d'organisations qui collaborent consiste à regrouper ou même à fusionner les politiques de sécurité des différentes organisations en une politique globale unique [Cuppens *et al.*, 1998]. Cette approche a l'avantage de laisser à chaque organisation le contrôle de sa propre politique de sécurité, ce qui réduit les problèmes de confidentialité.

Le processus de regroupement de politiques de sécurité peut utiliser la notion de *médiateur* [Wiederhold, 1992], [Dawson *et al.*, 2000] qui sert de point de liaison entre les différentes politiques, mais ce n'est pas souhaitable dans le sens où un tel médiateur doit être digne de confiance, ce qui repose, à un degré moindre, les mêmes problèmes que la super organisation de la section précédente. En 2005, Shehab et Bertino [Shehab *et al.*, 2005] ont présenté une plateforme distribuée fournissant une interopérabilité sécurisée pour des environnements collaboratifs sans médiateur, la décision de contrôle d'accès étant prise indépendamment dans chaque domaine. Cette étude est intéressante dans le sens où elle permet à des organisations ayant leurs propres politiques de sécurité de collaborer de façon sécurisée sans médiateur. Néanmoins, cela suppose d'établir des *chemins d'accès* entre les domaines de contrôle d'accès, et donc des modifications spécifiques des politiques de contrôle d'accès de chaque domaine.

En 2003, Lorch et proctor [Lorch et al., 2003] ont présenté XACML, un langage de contrôle d'accès, comme l'un des composants d'une plateforme d'autorisation distribuée et interopérable. Ce travail illustre la façon dont l'autorisation peut être déployée au sein de systèmes distribués et décentralisés, et comment on peut interconnecter les composants du système d'autorisation réparti. XACML [OASIS, 2003] est utile pour spécifier des politiques complexes dans une large variété d'applications, systèmes et environnements distribués. Néanmoins, XACML a quelques limitations, la flexibilité et l'expression du langage induit un coût en complexité et verbosité, entre autres, et il est difficile de travailler directement sur les fichiers de politiques XACML hétérogènes, à la fois pour des raisons syntaxiques et sémantiques. Des outils sont en cours de développement pour aider à spécifier et à manipuler les politiques en XACML, mais il faudra du temps pour que les praticiens de la sécurité maîtrisent ce langage et ces outils, et soient capables de les appliquer à des problèmes d'une complexité réaliste.

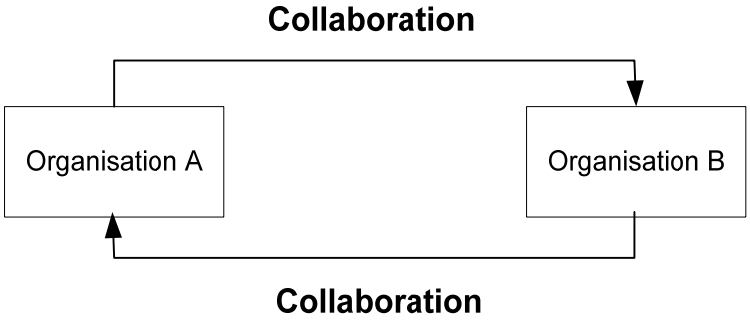


Figure 18 : Collaboration en utilisation des liaisons « pair à pair » entre organisations.

En 2008, Sturm et Dittrich [Sturm et al., 2008] ont présenté une méthode de contrôle d'accès dans les réseaux pair à pair, utilisant les mécanismes de contrôle d'accès des machines participantes. Les participants au réseau pair à pair exportent leurs politiques de contrôle d'accès en XACML [OASIS, 2003]. L'étude propose deux techniques pour combiner ces politiques. La première consiste à établir une correspondance (*mapping*) entre les politiques exportées : l'accès par l'un des participants aux données d'un autre est conditionné par un *contrat* établi entre les deux parties, ce contrat définissant la correspondance entre la politique de l'appelant et celle de l'appelé. La seconde technique consiste à utiliser une base de données commune, contenant tous les droits de tous les participants, et partagée par le système pair à pair lui-même. Même si cette étude se limite au partage de données dans les réseaux pair à pair, contexte fort éloigné du nôtre, elle est intéressante et présente certaines similitudes avec l'approche que nous proposerons au chapitre suivant, en particulier par la notion de contrats.

Il est donc clair que l'approche décentralisée présente de nombreux avantages par rapport à l'approche centralisée, et les trois modèles que nous allons analyser plus en détail adoptent tous plus ou moins cette approche décentralisée.

2.3.3 Le modèle Multi-OrBAC

Multi-OrBAC [Abou El Kalam & Deswarte 2006] est une extension d'OrBAC pour les systèmes multi-organisationnels. C'est un modèle de sécurité dynamique et adaptable, permettant d'un côté de spécifier des politiques de sécurité différentes dans chaque organisation, et de l'autre d'imposer des règles pour les interactions entre organisations qui soient compatibles avec les politiques de chaque organisation. Ce modèle vise donc des applications et systèmes complexes, hétérogènes, interopérables et distribués. Pour cela, les concepts de rôles, vues, activités, ont subi certaines modifications conceptuelles.

Un sujet ne joue pas forcément le même rôle dans toutes les organisations d'un contexte multi-organisationnel. C'est pourquoi, dans Multi-OrBAC, la notion de *rôle dans l'organisation* (RdO) est définie comme une extension de la notion de rôle d'OrBAC.

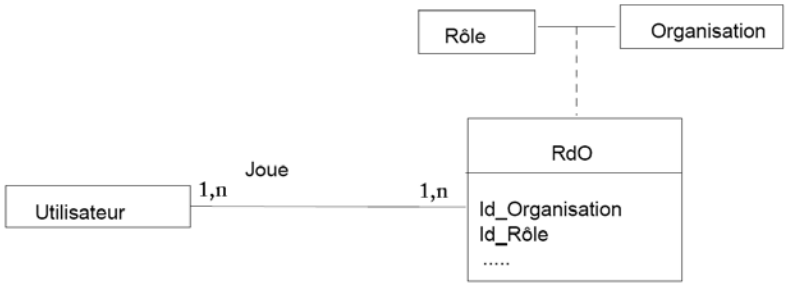


Figure 19 : Ébauche du diagramme de classe pour la classe-association RdO.

De même, une vue (abstraction des objets) peut être définie différemment selon l'organisation, ce qui amène à plusieurs définitions possibles d'une même vue, d'où la notion de *vue dans l'organisation* (VdO).

De la même manière, une activité (abstraction des actions) peut être définie différemment selon les organisations, conduisant à plusieurs définitions possibles d'une même activité, d'où la notion d'*activité dans l'organisation* (AdO).

Ainsi de la même manière que les notions de RdO, VdO et AdO ont été définies, Multi-OrBAC définit la classe-association *contexte dans l'organisation* (CdO). Une instance de CdO pourrait être par exemple "urgence dans l'hôpital Rangueil". Évidemment, l'administrateur de sécurité de l'organisation correspondante doit spécifier, pour chaque règle qu'il définit dans la politique de son organisation, des conditions qui seront évaluées en temps réel pour décider si un CdO est vrai ou pas à un instant donné.

2.3.3.1 Règles de sécurité Multi-OrBAC

Une organisation *org* accorde au RdO *r* la permission (ou l'interdiction, ou l'obligation, ou la recommandation) de réaliser l'AdO *a* sur la VdO *v*, si le contexte CdO *c* est vérifié. Cette particularité se traduit par la relation Type_accès suivante : Type_accès (organisation, rôle, activité, vue, contexte). Un accès peut être une permission, interdiction, obligation, recommandation.

Dans Multi-Orbac (voir Figure 20) comme dans OrBAC, les règles sont définies au niveau abstrait, à l'aide des entités : organisations, RdO, VdO, AdO et CdO. Cependant, les règles de la politique d'une organisation peuvent être définies sur des entités d'une autre organisation. Ainsi, une règle de la politique de l'organisation *O*1 peut se rapporter à un RdO de l'organisation *O*3 et une VdO de l'organisation *O*2, les organisations *O*1, *O*2 et *O*3 étant toutes différentes. Cependant, dans ce cas, *O*1 doit être une super organisation de *O*2, car une organisation ne peut définir de règles que pour des objets lui appartenant, ou appartenant à l'une de ses sous-organisations. Par exemple, la politique de sécurité de l'hôpital Purpan ne doit porter que sur des objets de l'hôpital Purpan, qu'ils soient définis par des vues au niveau de la politique de sécurité de l'hôpital ou au niveau d'une politique d'un des services de l'hôpital.

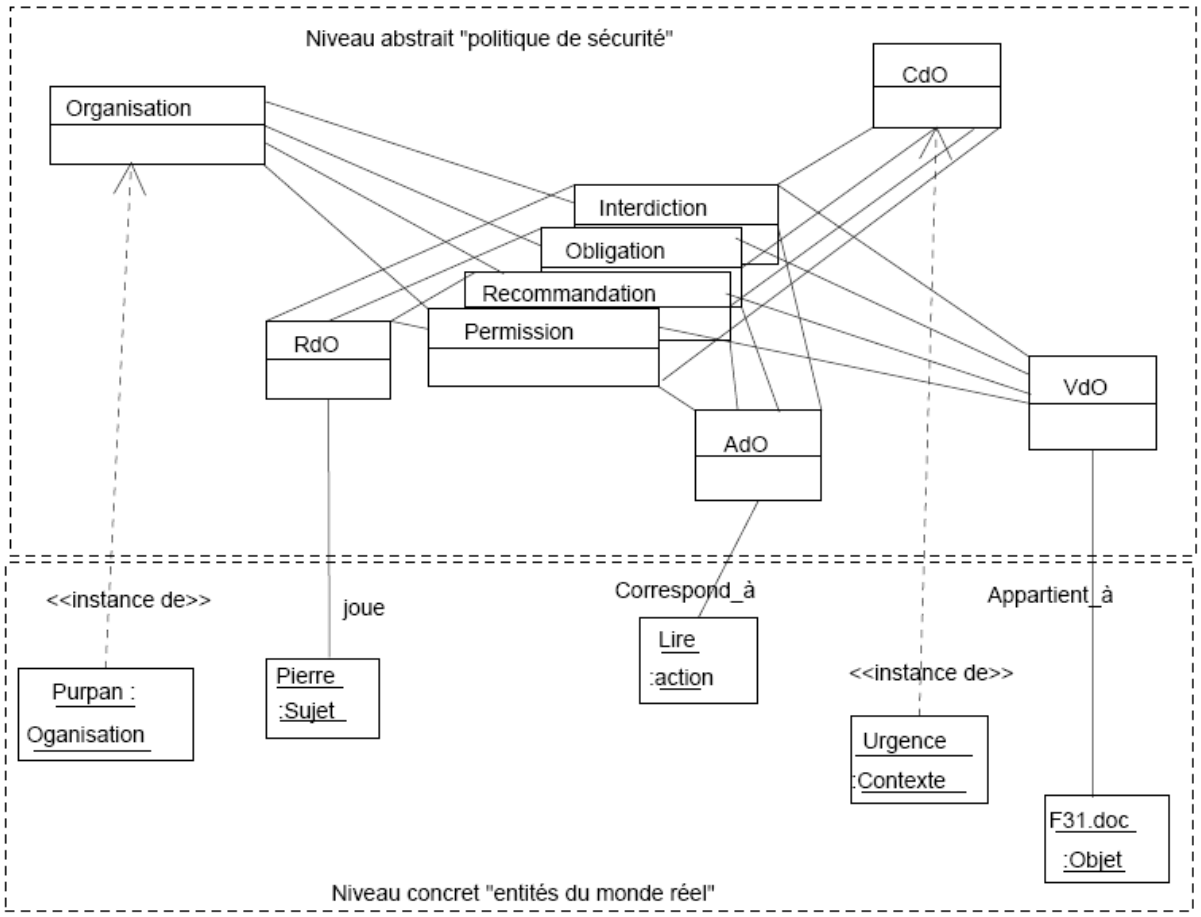


Figure 20 : Exemple d'instanciation d'une règle de sécurité Multi-OrBAC.

2.3.3.2 Avantages et inconvénients de Multi-OrBAC

Multi-OrBAC garde la méthodologie très pertinente d'OrBAC et l'adapte pour créer des architectures multi-organisationnelles, coopératives, distribuées et interopérables. Le fait que Multi-OrBAC définisse les concepts de rôle, activité, vue et contexte par rapport à une organisation, lui confère une flexibilité et une extensibilité (*scalability*) importante. Multi-OrBAC offre aussi un formalisme qui permet de décrire les politiques de sécurité d'organisations coopérant dans un même système tout en facilitant :

- l'élimination d'ambiguïtés dans la spécification et la vérification de sa complétude ou de sa cohérence.
- la détection et la résolution de conflits entre les règles de la politique, entre les objectifs de sécurité, ou entre les règles et les objectifs.
- la gestion de problèmes d'incohérence entre les politiques de sécurité hétérogènes de chaque organisation.

Le principal défaut de Multi-OrBAC réside dans le fait que la définition de la politique de sécurité de chaque organisation doit prendre en compte, donc connaître, des entités appartenant à d'autres organisations, et suppose donc une confiance entre les organisations pour ce qui concerne la gestion de ces entités. Ainsi, par exemple, si la politique de l'organisation *O*1 comporte des règles accordant des permissions à des rôles définis par l'organisation *O*2,

implicitement, *O*1 doit faire confiance à *O*2 pour l'authentification de ses utilisateurs et pour leur attribuer des rôles légitimes. Ceci est en contradiction avec les exigences d'autonomie et de confidentialité que nous avons discutées dans le chapitre 1.

2.3.4 Modèle d'organisation virtuelle

Les besoins de collaboration entre organisations et les facilités technologiques donnent aujourd'hui naissance à une nouvelle forme d'organisation et de coopération que l'on nomme *virtuelle*. Une Organisation Virtuelle (OV) est un ensemble d'organisations reliées par les technologies d'information (TI), poursuivant la réalisation commune d'un projet ou d'une activité économique [Nasser et al., 2005].

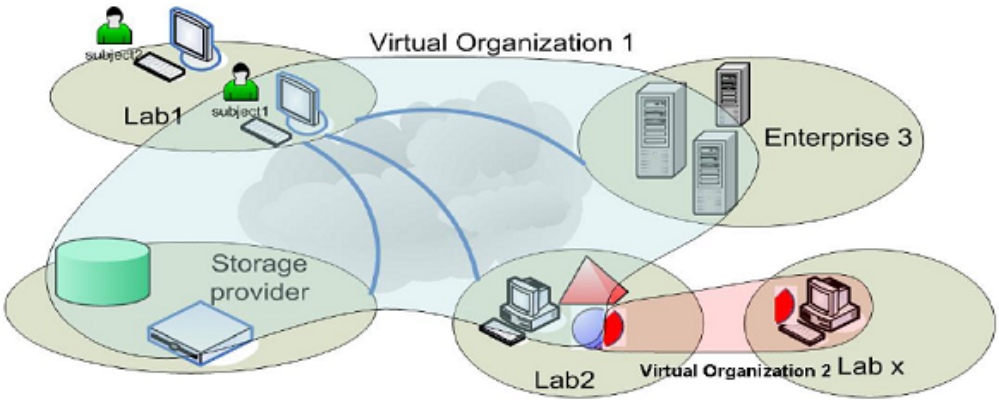


Figure 21 : Notion d'Organisation Virtuelle (VO) [Nasser et al., 2005].

Via les ressources humaines et matérielles mises à disposition par et pour les organisations membres, celles-ci pourront réaliser les processus nécessaires à la réalisation du but recherché lors de la mise en place de l'organisation virtuelle. Une **O**rganisation **V**irtuelle définit une communauté virtuelle formée de plusieurs organisations ou entreprises qui unissent leurs compétences et ressources pour répondre à une opportunité qu'elles n'auraient pu prendre en charge seules.

Cette alliance est en général définie pour une durée limitée : une fois le bien ou service livré, le regroupement est démantelé. Les infrastructures des technologies de l'information sont au cœur de l'organisation virtuelle et le support de ses activités. Des plateformes d'interconnexion offertes par une tierce partie peuvent faciliter la mise en œuvre de l'organisation virtuelle. Contrairement aux organisations traditionnelles, les frontières de l'organisation virtuelle restent ambiguës. Ces frontières sont définies par la stratégie de chacune des organisations pour réaliser les tâches qui lui sont confiées. La concrétisation du concept d'organisation virtuelle passe forcément par la définition d'une politique de sécurité

2.3.4.1 Modèle de sécurité pour les organisations virtuelles

Le travail présenté par B. Nasser dans sa thèse [Nasser, 2006], avait pour objectif de définir les contours de l'organisation virtuelle à travers une politique de sécurité, plus particulièrement concernant le contrôle d'accès afin de concrétiser la mise en place de cette organisation virtuelle via une structure logique distribuée aux organisations membres. Dans ce sens, ce travail propose une approche intéressante pour créer et gérer une organisation virtuelle. Il présente une méthodologie pour l'expression des interactions au niveau de l'organisation virtuelle. Ces interactions constituent la référence pour la mise en place d'une politique de contrôle d'accès chez les différents partenaires.

Ce travail répond aux questions suivantes : premièrement, comment définir une politique de contrôle d'accès trans-organisationnelle (pour l'organisation virtuelle). Deuxièmement, comment spécifier une politique de contrôle d'accès propre à chaque collaboration tout en respectant l'autorité de chaque organisation sur ses utilisateurs et ressources. Et enfin, comment minimiser l'effet de la structure interne de l'organisation membre sur la coopération dans l'OV. Une politique de contrôle d'accès en termes des entités classiques *sujet*, *action* et *objet* n'est pas suffisante pour répondre à ces questions. À titre d'exemple, elle ne permet pas de spécifier facilement qu'un utilisateur d'une organisation est autorisé à réaliser une action sur un objet d'une organisation partenaire. Pour cela, cette étude a adopté les abstractions d'OrBAC en termes de *rôles*, *activités* et *vues*.

Par ailleurs, l'étude propose que l'accès inter-organisationnel au sein de l'organisation virtuelle se fasse en interfaçant les organisations membres via des rôles dits « exportés ». Les droits associés à un rôle exporté sont spécifiés par l'organisation propriétaire des ressources (qui *importe* le rôle). Un partenaire de l'OV qui importe un rôle peut l'attribuer à ses propres utilisateurs pour bénéficier des droits qui lui sont associés. Pour modéliser la politique de contrôle d'accès abstrait, l'étude s'est donc appuyée sur le modèle OrBAC, en l'étendant (comme Multi-OrBAC) pour exprimer les relations inter-organisationnelles au sein d'une même OV.

Cette collaboration peut être concrétisée via deux approches : la traduction des rôles importés en rôles locaux, ou la spécification d'une politique spécifique pour chaque OV dans chaque organisation participante. Dans ces deux cas, les rôles sont utilisés pour permettre l'accès des utilisateurs d'un partenaire aux ressources de l'autre. Ces rôles sont bien liés à l'organisation virtuelle comme contexte de collaboration. De cette façon il est possible de modéliser l'appartenance d'une organisation à plusieurs organisations virtuelles ainsi que la collaboration entre deux organisations dans plusieurs organisations virtuelles.

Par ailleurs, l'étude a déterminé les entités et les vues administratives qu'il faut rajouter à AdOrBAC [Cuppens & Miège 2003], le modèle d'administration d'OrBAC, pour permettre la gestion de ces nouvelles relations. Et enfin, cette étude a analysé les aspects de déploiement de l'organisation virtuelle selon le modèle proposé, et une architecture pour l'authentification et l'autorisation a été proposée.

Afin de mieux comprendre le fonctionnement de cette approche, nous détaillons par la suite un exemple montrant l'utilisation du modèle OrBAC dans le contexte d'organisations virtuelles. Prenons le cas où une organisation *Org*2 fournit les ressources (fournisseur) à une autre organisation *Org*1 qui contient les utilisateurs (consommateurs). La mise en place de l'OV nécessite la définition des entités participantes des deux côtés, y compris les rôles, les vues, les activités et les autorités d'administration. Afin de modéliser cet environnement avec OrBAC, nous commençons par la gestion de l'organisation virtuelle elle-même, en considérant que l'OV est une organisation selon le formalisme OrBAC. Afin de décrire la politique de l'OV, il faut prendre en compte les attributs des organisations participantes, avec un nom pour décrire chaque OV à laquelle peut participer une ou plusieurs organisations. Il est également possible de tenir compte d'une date d'expiration au-delà de laquelle la coopération est terminée.

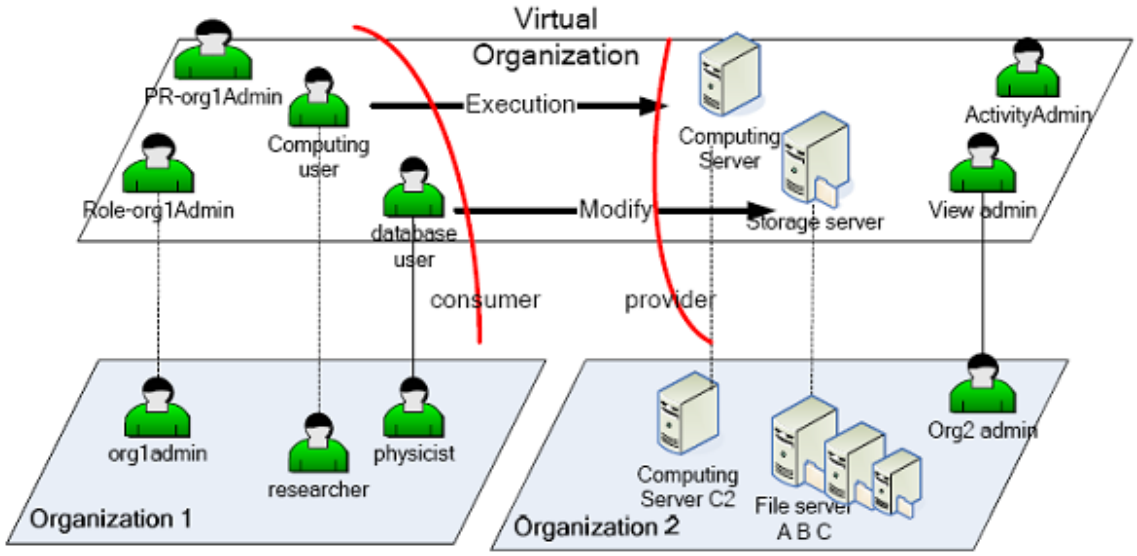


Figure 22 : Le modèle OrBAC au sein d'un environnement sous forme de grille [Nasser, 2006].

Au niveau de l'administration de la sécurité, il est nécessaire de définir les rôles d'administrateurs dans l'organisation virtuelle. De la même façon, les vues et les activités correspondantes doivent être définies pour l'organisation virtuelle. D'un côté, il existe un utilisateur "org1admin" qui joue le rôle "Role-org1Admin", avec la responsabilité d'assigner/révoquer les rôles pour l'organisation *Org*1. Ce même utilisateur "org1admin" joue aussi le rôle "PR-org1Admin", avec la responsabilité d'assigner/révoquer les permissions pour l'organisation *Org*1. De l'autre côté, il existe un utilisateur "org2admin" qui joue le rôle "ViewAdmin", avec la responsabilité d'assigner/révoquer les vues aux objets pour l'organisation *Org*2. Ce même utilisateur "org2admin" joue aussi le rôle "ActivityAdmin", avec la responsabilité d'assigner/révoquer les activités aux actions pour l'organisation org2. Nous devons faire correspondre le niveau concret aux entités définies dans le niveau abstrait : Role-org1Admin a la permission d'attribuer un rôle à un utilisateur donné, ViewAdmin a la permission d'intégrer un objet dans une vue donnée. ActivityAdmin a le droit de considérer une action dans une activité donnée, et PR-org1Admin a le droit de permettre à un rôle de faire une activité sur une vue donné.

De cette façon, cette approche permet aux différents partenaires de participer à la définition de la politique de contrôle d'accès, tout en maintenant l'autonomie et la flexibilité dans la gestion de leurs systèmes locaux. L'attribution d'un utilisateur ou d'une ressource dépend du domaine de la politique locale ainsi que des paramètres de performance et la qualité de service. Néanmoins, ce formalisme doit être déployé avec une plateforme de négociation pour mettre en place une telle OV multi administrée.

2.3.4.2 Avantages et inconvénients du modèle d'organisation virtuelle

Ce travail est très intéressant en ce qu'il propose entre autres une étude approfondie sur la manière de définir des politiques de contrôle d'accès des organisations virtuelles, et sur la définition et l'utilisation des différentes entités (rôles, vues, activités). Néanmoins nous n'allons pas nous baser directement sur ces résultats, car notre objectif principal est de définir des politiques de contrôle d'accès distinctes et autonomes pour chaque organisation faisant partie du processus de collaboration, sans utiliser ou intégrer la notion d'organisation virtuelle, qui, comme Multi-OrBAC revient à définir une vision commune d'une certaine politique de sécurité globale de l'organisation virtuelle, chaque organisation devant *importer* des entités définies par d'autres, ce qui suppose une certaine confiance mutuelle.

2.3.5 Le modèle O2O (Organisation 2 Organisation)

O2O (*Organization to Organization*) [Cuppens et al., 2006], [Coma, 2006] est une approche formelle qui permet de gérer l'interopérabilité et la collaboration entre des entités ayant leurs propres politiques de sécurité définies dans différentes organisations. O2O est aussi une extension du modèle OrBAC (Organization-Based Access Control), basée sur deux concepts clés : *Virtual Private Organization* (VPO) et *Role Single-Sign-On* (RSSO).

2.3.5.1 Présentation de la notion de VPO

Chaque organisation gère les accès venant des organisations externes à ses propres ressources, en créant une sous-organisation, appelée VPO (*Virtual Private Organization*).

La notion de VPO est basée sur la notion de *Virtual Organization* (VO). Une VPO permet à une organisation effectuant une transaction avec d'autres organisations de garder le contrôle sur ses propres ressources auxquelles les autres organisations peuvent accéder au cours de la collaboration. Si une organisation Alice.org veut interagir avec une organisation Bob.org, chacune des organisations définit sa propre VPO, respectivement appelées A2B (pour Alice2Bob) et B2A (voir Figure 23).

La VPO A2B contient une politique de contrôle d'accès OrBAC, gérée par l'organisation Alice.org, qui définit comment les sujets de l'organisation Alice.org ont accès aux ressources de l'organisation Bob.org et de même, B2A contient une politique de contrôle d'accès OrBAC, gérée par Bob.org, qui définit la manière dont les sujets de l'organisation Bob.org accèdent aux ressources de l'organisation Alice.org. Chaque organisation définit une "sphère d'autorité", qui

définit une relation entre les organisations : une organisation *B* est dans la sphère d'autorité d'une autre organisation *A* si la politique de contrôle d'accès qui s'applique à *B* est définie et administrée par *A*.

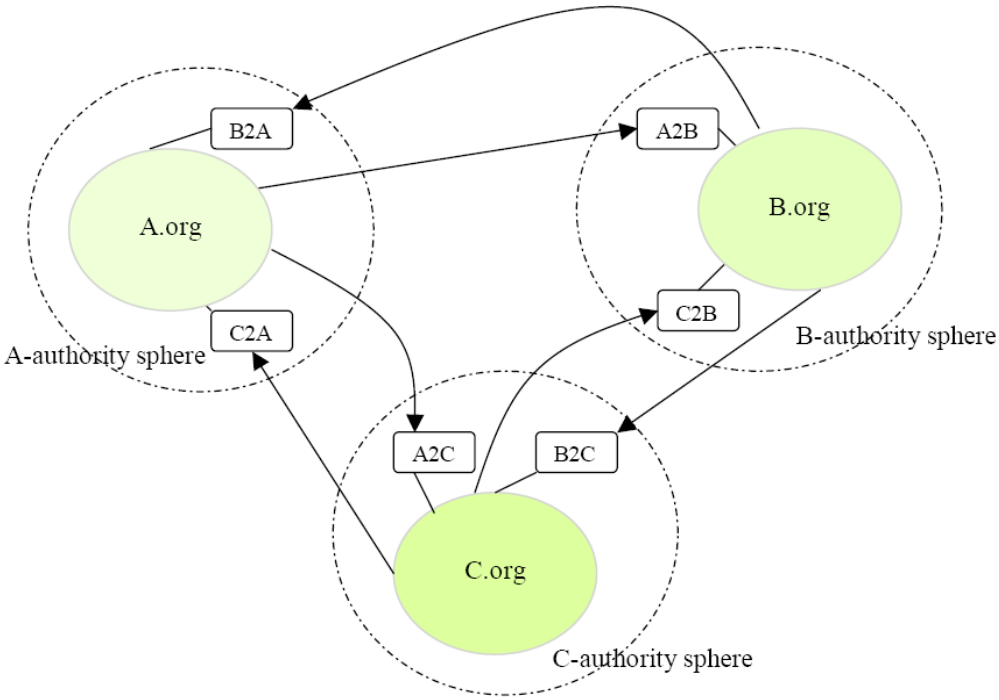


Figure 23 : Fonctionnement du modèle O2O [Coma 2006].

Une VPO est donc une entité dynamique créée pour atteindre un but d'interopérabilité et qui disparaît une fois que ce but n'est plus souhaité. Ainsi, chaque fois qu'une organisation *A* veut accéder à certaines ressources de *B*, il faut créer une VPO A2B et puis supprimer cette VPO quand la collaboration cesse. Si par exemple, 100 transactions ont lieu, il faudrait créer 100 VPO, puis les supprimer. Cela induit une lourdeur, une perte de temps et de ressources. Ce problème est vraiment important dans le cas des processus répétitifs (par exemple dans le cas d'accès journaliers, ou hebdomadaires). Une autre solution serait d'établir une VPO une fois pour toute, mais cela ouvrirait une possibilité d'interaction permanente, même en dehors d'une transaction. Pour chaque couple d'organisations voulant collaborer, il faut créer une VPO : A2B, A2C, A2D, etc. Si *A* veut collaborer avec 100 autres organismes, il faut créer 100 VPO A2{autre organisation}, chacune de ces VPO étant gérée par la sphère de l'autorité de l'autre organisation à laquelle *A* demande une ressource. De même, si 100 organisations veulent accéder à des ressources de *A*, *A* sera obligée de créer et de gérer 100 VPO (par exemple, B2A, C2A, D2A, E2A, etc.). Ce fonctionnement est particulièrement lourd.

2.3.5.2 Dépendance des droits d'accès avec les rôles

Nous ne pouvons pas considérer que les utilisateurs ayant un rôle spécifique dans leur organisation, vont avoir les mêmes privilèges dans d'autres organisations collaborant, puisque la définition des rôles peut être (et est très probablement) différente dans chaque organisation. Le principe RSSO (introduit dans O2O) permet à un sujet donné de garder le même rôle quand il

accède à une autre organisation, mais en tenant compte des privilèges définis dans la VPO. Un rôle *médecin* dans une organisation *A* peut avoir des privilèges différents d'un autre rôle *médecin* dans une autre organisation *B*. Cela implique que l'organisation A connaisse quels privilèges sont attribués au rôle *médecin* dans l'organisation *B* et donc une divulgation d'information de *B* vers *A*, et réciproquement (*B* doit connaître les rôles de *A* et leur signification), ce qui n'est acceptable que si les deux organisations se font mutuellement confiance. D'un autre côté, si l'on attribue à un utilisateur de *A* les mêmes droits dans *B* que ceux que lui attribue son rôle dans *A*, ceci réduirait significativement l'autonomie de *B*, sans résoudre pour autant les problèmes de confidentialité.

2.3.5.3 Gestion des utilisateurs et rôles

La VPO B2A est associée à une politique de sécurité qui gère la façon dont les sujets de l'organisation B accèdent à des données de l'organisation A. B est l'organisation bénéficiaire, A est l'organisation offrant. B2A est dans la SA (**S**phère d'**A**utorité) de A. De plus, pour un ensemble (sujet, action, objet), le sujet appartient à B, l'objet appartient à A et l'action appartient à A. Considérons que l'utilisateur X veut jouer le rôle de "médecin" dans un organisme A, nous devons définir la règle suivante :

- Permission (bh2ah, médecin, consultation, dossier médical, urgence)

et la définition des rôles suivants :

- Habilite (bh2ah, X, médecin)
- Habilite (b_hosp, X, médecin)

Cela implique que pour chaque utilisateur de *B* qui joue le rôle *médecin* dans *B* et qui veut jouer un rôle *médecin* dans *A*, cet utilisateur doit être défini dans B2A, puis on doit lui attribuer le rôle *médecin* dans *A*. Ce processus n'est pas efficace. De plus, *A* et *B* peuvent avoir une définition différente du rôle *médecin*. Le modèle O2O nécessite donc de gérer la compatibilité entre les données (en particulier, les rôles) des différentes organisations collaborant. Cela pourrait être résolu par une entité de plus haut niveau, créée pour gérer cette compatibilité. Mais une telle entité devrait avoir connaissance des structures de *A* et de *B*, ce qui pose à la fois un problème de confidentialité, et d'autonomie (ou de souveraineté). La solution adoptée par O2O consiste à faire gérer la correspondance entre rôles par la connaissance mutuelle des rôles et des utilisateurs dans les deux organisations concernées et aussi par la confiance dans la relation établie entre utilisateur et rôle par l'organisation de l'utilisateur. Ceci exige donc une confiance mutuelle entre les deux organisations.

2.3.5.4 Avantages et inconvénients du modèle O2O

L'approche O2O présente plusieurs avantages, entre autres l'utilisation du RSSO qui permet à un sujet donné de garder le même rôle quelle que soit l'organisation à laquelle il accède, mais avec des privilèges définis par la VPO de l'organisation à laquelle il accède. Les privilèges diffèrent d'une VPO à une autre, chaque organisation appliquant sa propre politique de sécurité pour gérer l'interopérabilité.

Toutefois, comme on l'a vu cette approche a des limites :

- Pour qu'une organisation *A* accepte que des utilisateurs d'une autre organisation *B* puissent jouer un rôle équivalent dans *A*, il faut que *B* fasse confiance à *A* sur la façon d'affecter ses rôles (ce qui est une perte d'autonomie pour *B*), et que *A* fasse confiance à *B* au point de lui fournir les informations nécessaires sur ses rôles et sur leur affectation aux utilisateurs (ce qui est une perte de confidentialité pour *A*).
- À chaque fois que dans une collaboration une organisation *A* doit utiliser des ressources d'une organisation *B*, *B* doit créer une VPO, et la détruire dès que cette utilisation de ressource n'est plus utile. Comme il s'agit d'un processus assez lourd et complexe, ceci induit un coût important.

2.4 Conclusion du chapitre 2

Dans ce chapitre, nous avons étudié et analysé l'état de l'art dans le domaine des politiques et modèles de sécurité traditionnels (RBAC, OrBAC), et les modèles de sécurité pour la collaboration (Multi-OrBAC, OV, O2O).

Pour chaque modèle, nous avons détaillé les limites et les avantages. Le but étant de déterminer quel modèle pourrait convenir à nos besoins en contrôle d'accès et collaboration afin de créer une plateforme de contrôle d'accès collaboratif répondant aux besoins et critères de sécurité décrits dans le chapitre précédent. Cette analyse des modèles de contrôle d'accès collaboratifs existants doit nous aider à construire un nouveau modèle qui satisfasse les exigences de confidentialité et d'autonomie que nous avons établies au premier chapitre pour les IIC, exigences qui ne sont pas satisfaites par les modèles existants.

Dans le chapitre suivant, nous allons proposer le modèle PolyOrBAC, basé sur le modèle OrBAC et sur des interactions par services Web, en mettant l'accent sur la manière dont les différentes organisations collaborent et gèrent leurs politiques de contrôle d'accès respectives.

Chapitre 3. PolyOrBAC : schéma-cadre de sécurité pour la collaboration

Dans le premier chapitre, nous avons fixé le contexte de notre travail, en détaillant les besoins et les exigences de sécurité des infrastructures critiques en général et des infrastructures d'information critiques en particulier. Cette étude du contexte nous a permis d'identifier les spécificités et les caractéristiques d'une infrastructure critique, et plus particulièrement les besoins en sécurité et contrôle d'accès de l'infrastructure d'information critique qui la contrôle. Dans le deuxième chapitre, après avoir présenté des modèles de contrôle d'accès traditionnels tels que RBAC et OrBAC, nous avons établi un état de l'art sur les modèles de contrôle d'accès collaboratifs, en analysant plus en détail un schéma pour les organisations virtuelles, et deux autres pour des organisations multiples (O2O et Multi-OrBAC). Cet état de l'art nous a permis de mettre en évidence les lacunes des modèles existants vis-à-vis des exigences que nous avons identifiées pour les infrastructures d'information critiques.

Dans ce chapitre, nous proposons une nouvelle approche que nous avons appelée *PolyOrBAC*, pour modéliser et mettre en œuvre la sécurité et la collaboration entre organisations [Abou El Kalam *et al.*, 2007], [Baïna *et al.*, 2008], [Abou El Kalam & Deswarte, 2009], [Abou El Kalam *et al.*, 2009]. Cette approche se base sur deux composants principaux : (1) le modèle de sécurité OrBAC, utilisé pour spécifier la politique de sécurité locale à chaque organisation, ce modèle étant étendu pour prendre en compte les mécanismes de coopération avec les autres organisations, ainsi que (2) la technologie des services Web qui nous permet de fournir une plateforme de collaboration et d'interopérabilité entre les organisations, avec des extensions pour permettre de spécifier et de mettre en œuvre la sécurité sur les interactions. Nous montrons comment ces deux composants se complètent pour fournir un schéma-cadre de sécurité applicable dans le contexte des infrastructures critiques et plus particulièrement dans celui des infrastructures d'information critiques.

Le choix d'OrBAC pour modéliser les politiques de sécurité de chaque organisation est motivé par sa capacité d'abstraction qui facilite la spécification de politiques de sécurité sans entrer dans le détail des utilisateurs et des ressources que doit contrôler l'organisation. La définition récursive du concept d'organisation permet aussi de maîtriser la complexité en généralisant la notion d'héritage de rôle qui existait déjà dans RBAC. Également pour maîtriser la complexité et favoriser l'extensibilité (*scalability*), il est préférable de restreindre les interactions entre organisations à des relations bipartites, du type client-serveur, plutôt que de permettre des interactions sans contrainte, dont il serait plus difficile de spécifier et de mettre en œuvre la

sécurité. C'est pourquoi nous avons choisi la technologie des services Web comme modèle d'interaction dans PolyOrBAC.

PolyOrBAC se base donc sur deux composants principaux : une extension du modèle de sécurité OrBAC pour prendre en compte les services Web, et une adaptation de la technologie des services Web pour permettre une collaboration sécurisée entre les organisations. Cette extension et cette adaptation sont décrites dans les deux sous-sections suivantes.

3.1 Extension d'OrBAC aux systèmes collaboratifs

Dans PolyOrBAC, le modèle OrBAC est utilisé pour spécifier les politiques de contrôle d'accès locales au sein de chaque organisation faisant partie de l'IIC, avec des extensions pour la collaboration entre ces organisations. Grâce aux règles abstraites d'OrBAC, il est possible de spécifier dans un format unique des règles de sécurité de plusieurs systèmes distribués et hétérogènes, tout en offrant un fonctionnement adaptable et flexible pour chaque organisation.

OrBAC n'est pas le seul modèle qu'on peut utiliser pour définir des politiques de sécurité dans le contexte des services Web. En 2006, un modèle [Ardagna et al., 2006] a été proposé pour appliquer des politiques de contrôle d'accès aux services Web. Dans ce modèle, la politique, appelée *WS-Policy*, est exprimée en XACML [OASIS, 2003]. Cette étude présente des concepts de base pour la sécurisation des services Web et définit des conditions nécessaires pour assurer la mise en œuvre de services Web sécurisés. Elle propose aussi une architecture permettant d'appliquer les WS-Policies. Cette étude peut servir d'inspiration pour spécifier des politiques de sécurité dans les services Web, même si nous préférons utiliser le formalisme OrBAC, qui est très différent de celui de WS-Policy.

Il existe en outre différents travaux utilisant des modèles de contrôle d'accès pour la sécurité des services Web, mais ils ne satisfont pas réellement aux besoins des IIC. Ainsi en 2004, le modèle S-RBAC [Xu *et al.*, 2004] (Service-Oriented RBAC) a été proposé comme modèle de contrôle d'accès et comme architecture de sécurité axée sur les services Web. Par rapport à d'autres modèles basés sur RBAC, celui-ci a l'avantage de mieux prendre en compte les caractéristiques spécifiques de l'architecture orientée service (SOA) propre aux services Web. À notre connaissance, ce modèle est celui qui se rapproche le plus de la méthodologie de notre travail. Ce qui différencie PolyOrBAC, c'est que notre but principal est de gérer à la fois la sécurité de chacune des organisations et de leurs interactions par les services Web, et non pas seulement de définir des politiques de contrôle d'accès pour les services Web.

Avec PolyOrBAC, il est donc nécessaire de définir des politiques de contrôles d'accès pour chacune des organisations participant dans le processus de collaboration. Dans une interaction par service Web, nous pouvons distinguer deux organisations différentes : le *client* qui utilise le service et le *prestataire* qui fournit ou met à disposition le service. Le client (en tant qu'organisation) doit contrôler les accès que ses propres utilisateurs peuvent vouloir faire à des services externes et le prestataire doit contrôler les accès à ses ressources locales par ses propres utilisateurs comme par l'organisation cliente. Ainsi qu'on l'a vu au chapitre précédent, le modèle OrBAC est bien adapté pour spécifier les politiques de sécurité locales à chaque organisation,

mais ne gère pas directement les interactions entre organisations. Pour analyser comment l'étendre pour gérer ces interactions, nous allons d'abord analyser la technologie des services Web.

3.2 Collaboration et interopérabilité avec les Services Web

Les systèmes d'information étant de plus en plus ouverts, distribués et multi-organisationnels, la coopération est devenue un aspect très important à prendre en compte. La notion de *travail collaboratif assisté par ordinateur* (*Computer Supported Cooperative Work* ou CSCW)" a conduit à des développements comme COCA[7] [Li & Muntz, 1998], DCWPL[8] [Cortés & Mishra, 1996] et CSDL[9] [de Paoli & Tisato, 1994]. COCA et DCWPL construisent un environnement collaboratif et distribué à partir de spécifications de haut niveau, tout en séparant la coordination des fonctionnalités de traitement. Ainsi, les caractéristiques de coordination peuvent être modifiées facilement sans aucun changement au niveau des processus de traitement. COCA aide ainsi à développer des systèmes collaboratifs et à modéliser des politiques de coordination. DCWPL permet de spécifier de multiples mécanismes de coordination pour un même traitement. CSDL est un langage créé afin de permettre le développement de systèmes coopératifs depuis les étapes de spécification jusqu'aux étapes d'implémentation. Néanmoins, ces spécifications sont limitées à la coordination et ne couvrent pas l'ensemble des besoins de collaboration et d'interopérabilité rencontrés dans un environnement collaboratif. Il convient donc d'analyser plus en détail l'architecture orientée services (*Service Oriented Architecture* ou SOA).

3.2.1 Architecture Orientée Services (SOA)

L'architecture orientée services est un cadre général permettant de mettre en relation des ressources (c'est-à-dire des applications et des données) pour fournir des services à des consommateurs (qui peuvent être des utilisateurs finaux ou d'autres services). SOA peut également être considérée comme un style d'architecture des systèmes d'information qui permet de construire des applications en combinant des services interopérables à couplage lâche (*loose coopling*). Dans ce contexte, un service est défini comme un module de traitement créé pour satisfaire les besoins d'une entité qui peut être un utilisateur humain ou un autre logiciel. Les systèmes développés pour être compatibles avec l'architecture SOA peuvent être indépendants des technologies et des plates-formes telles que Java, .NET, etc. Par exemple, une même application peut utiliser des services écrits en langage C# s'exécutant sur des plateformes .NET et des services écrits en Java fonctionnant sur des plates-formes J2EE. En outre, les services qui s'exécutent sur une plate-forme peuvent eux-mêmes invoquer des services s'exécutant sur une autre, ce qui facilite la réutilisation. L'architecture orientée services n'est donc pas liée à une technologie spécifique et peut être mise en place en utilisant un large éventail de normes d'interopérabilité. La technologie des services Web obéit au style architectural SOA et fournit une collection de protocoles et de normes tels que SOAP, XML et WSDL. SOAP est un protocole de RPC bâti sur XML, et les services Web basés sur SOAP sont en train de devenir

[7] Collaborative Objects Coordination Architecture.
[8] Describing Collaborative Work Programming Language.
[9] Cooperative Systems Design Language.

l'implémentation la plus répandue de SOA. La section suivante présente certaines de ces normes ainsi que les différents avantages de cette technologie.

3.2.2 Normes des services Web

Les logiciels et applications développées dans différents langages de programmation et fonctionnant sur différentes plateformes peuvent utiliser les services Web pour échanger des données à travers des réseaux tels qu'Internet, de manière similaire à la communication entre processus d'une même station de travail. Ces fonctionnalités sont facilitées par l'utilisation des normes ouvertes définies par le *World-Wide Web Consortium* (W3C) : XML, SOAP, WSDL, UDDI).

- **XML** (*Extensible Mark-up Language*) [W3C, 2004] est un métalangage qui définit des formats de données communs pour décrire les données et faciliter leur transmission, interprétation et partage entre organisations, sur Internet ou dans des intranets. Tout individu ou groupe d'individus ou d'entreprises qui veulent partager de l'information de manière cohérente peuvent utiliser XML pour la définition des données. Il est particulièrement efficace pour définir des documents Web.
- **SOAP** (*Simple Object Access Protocol*) [W3C, 2003] est proposé par le W3C en tant que mécanisme de transport de données afin d'échanger des données (particulièrement sous format XML) entre des applications s'exécutant sur différents systèmes d'exploitation. SOAP spécifie comment encoder un en-tête HTTP et un fichier XML de telle sorte que deux programmes sur deux ordinateurs distincts puissent communiquer. Les messages SOAP peuvent être transportés en utilisant une variété de protocoles Internet, y compris SMTP, MIME et HTTP. SOAP, avec XML et WSDL, forme la base des services Web.
- **WSDL** (*Web Services Description Language*) [W3C, 2006] est un langage basé sur XML, utilisé pour décrire les services que les prestataires offrent aux clients.
- **UDDI** (*Universal Description, Discovery, and Integration*) [OASIS, 2005] est un annuaire basé sur XML, qui permet à des organisations dans le monde entier, de lister et faire connaître leurs services sur Internet, ce qui facilite l'interopérabilité entre les différentes organisations. UDDI agit comme un registre de publication et de localisation des services. Différentes applications peuvent accéder, par exemple en tâche de fond, à des registres UDDI, afin de trouver les services Web qu'elles souhaitent, ainsi que toutes les informations dont elles ont besoin pour les utiliser. UDDI est souvent comparé à un annuaire téléphonique, il permet aux différentes organisations de se faire connaître par nom, produits, localisations et services Web proposés.

Pour récapituler, XML est utilisé pour structurer les données, SOAP est utilisé pour échanger les données, WSDL est utilisé pour décrire les services disponibles et UDDI est utilisé pour publier les services disponibles. Pour reprendre l'analogie du téléphone, l'invocation d'un service Web correspond à un appel téléphonique où XML représente la forme de conversation, SOAP décrit comment effectuer un appel, UDDI est l'annuaire et WSDL décrit les entrées dans l'annuaire.

L'ensemble de ces outils permet de séparer la description abstraite des fonctionnalités offertes par un service des détails concrets du fonctionnement du service, en particulier sur sa localisation et la façon de l'appeler. Utilisés principalement comme un moyen pour les entreprises de communiquer entre elles et avec leurs clients, les services Web permettent aux organisations d'échanger des données sans qu'il ne soit nécessaire d'avoir une connaissance intime des systèmes d'information de chaque organisation. Nous pouvons résumer le fonctionnement des services Web par huit étapes principales comme indiqué dans la Figure 24.

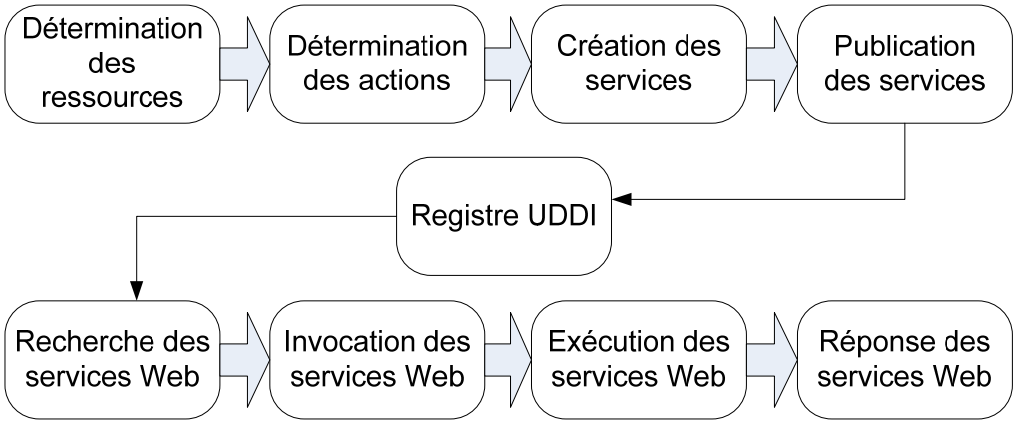


Figure 24 : Description simplifiée du fonctionnement des services Web.

Notons que la technologie des services Web permet aussi de résoudre de façon élégante les problèmes de concurrence (au sens de l'exécution simultanée) des applications : les invocations multiples d'un même service Web, par des utilisateurs appartenant à la même organisation ou à différentes organisations, sont gérées par le prestataire du service, qui peut les exécuter séquentiellement ou en parallèle, selon l'implémentation du service. Ce choix dépend du seul prestataire et ne doit pas avoir de conséquence pour les clients, pour autant que les performances (vues des clients) restent satisfaisantes. Les conséquences de ce choix sur la structuration des traitements et l'architecture du système d'information sont donc limitées au seul prestataire et ne sont pas visibles des autres organisations, ce qui renforce l'autonomie et la confidentialité. Les services Web fournissent donc des mécanismes de collaboration entre organisations qui sont intéressants dans notre contexte, parce qu'ils offrent des normes et des outils simples et bien acceptés dans de larges domaines, parfaitement compatibles avec les besoins des IIC.

3.3 Extensions du modèle OrBAC pour les services Web

3.3.1 Politique globale et politiques locales

Nous allons tout d'abord récapituler certains critères essentiels que nous devons prendre en compte dans la structure du schéma-cadre PolyOrBAC. Dans notre contexte, nous avons besoin d'un modèle de contrôle d'accès qui assure la collaboration entre les différentes organisations appartenant à l'IIC, sachant que chaque organisation définit sa propre politique de sécurité, met en place ses propres règles de fonctionnement pour ses utilisateurs, et héberge des ressources informatiques hétérogènes qui peuvent être complexes et distribuées sur des réseaux

interconnectés multiples. Le type de collaboration qu'on souhaite consiste à permettre à chaque organisation (et à ses utilisateurs) d'accéder de manière sécurisée aux ressources appartenant à d'autres organisations de l'infrastructure.

Pour cela, il faut mettre en place des mécanismes de contrôle d'accès sur les interactions entre organisations, donc au niveau des services Web, en fonction d'une politique de contrôle d'accès globale, commune aux organisations concernées. Il faut donc que la politique locale à chaque organisation prenne en compte les services Web fournis par cette organisation ainsi que ceux qu'elle utilise, la fourniture et l'utilisation de ces services obéissant à la politique globale. Il faut aussi que les politiques locales et globale soient compatibles et cohérentes en termes de sémantique et syntaxe, et qu'elles ne créent pas de contradiction ou de conflits entre elles.

La politique globale, quant à elle, doit gérer trois éléments primordiaux : l'interopérabilité, l'interconnexion et éventuellement l'interdépendance entre organisations. On pourrait imaginer de définir une politique globale à une IIC qui s'imposerait à toutes les organisations membres de l'infrastructure, mais outre que définir une telle politique globale serait un exercice difficile dès que l'infrastructure est d'une complexité significative, ceci serait nécessairement en contradiction avec les besoins d'autonomie des organisations, et d'évolutivité et d'extensibilité (au sens *scalability*) des infrastructures. Il est donc préférable de définir la sécurité des interactions au niveau de chaque service Web, ce qui ne réduit pas la généralité puisque toutes les interactions se font par des services Web. Il est aussi plus facile de gérer la compatibilité et la cohérence avec les politiques locales, puisque l'on doit considérer pour chaque service Web que deux organisations : le prestataire et le client.

Cette approche garantit l'autonomie de chaque organisation, puisque celle-ci peut librement décider de collaborer avec une autre organisation de son choix par service Web, ou au contraire de créer en interne le service dont elle a besoin ou de garder pour elle l'usage exclusif de ses ressources. Enfin, l'évolutivité et l'extensibilité de l'infrastructure sont favorisées, puisque l'adhésion d'une nouvelle organisation ou la séparation d'une organisation de l'infrastructure n'ont de conséquence que sur les organisations interagissant directement avec elles. La Figure 25 représente une configuration simple des interconnexions entre différentes organisations (cliente et prestataire) collaborant au moyen de services Web.

Par souci de simplification, nous avons supposé que tous les utilisateurs appartenant à l'organisation cliente sont connectés à une seule interface de Service Web, de l'autre coté, nous avons supposé que tous les services proposés par l'organisation prestataire sont accessibles de l'extérieur à travers une seule interface Web. En réalité, il existe plusieurs interfaces service Web pour les différents services Web offerts par un même prestataire, et il existe plusieurs interfaces service Web pour les différents utilisateurs d'une même organisation cliente. De plus les utilisateurs ne sont probablement pas connectés directement à cette interface, mais seulement à travers d'autres systèmes informatiques appartenant à l'organisation A.

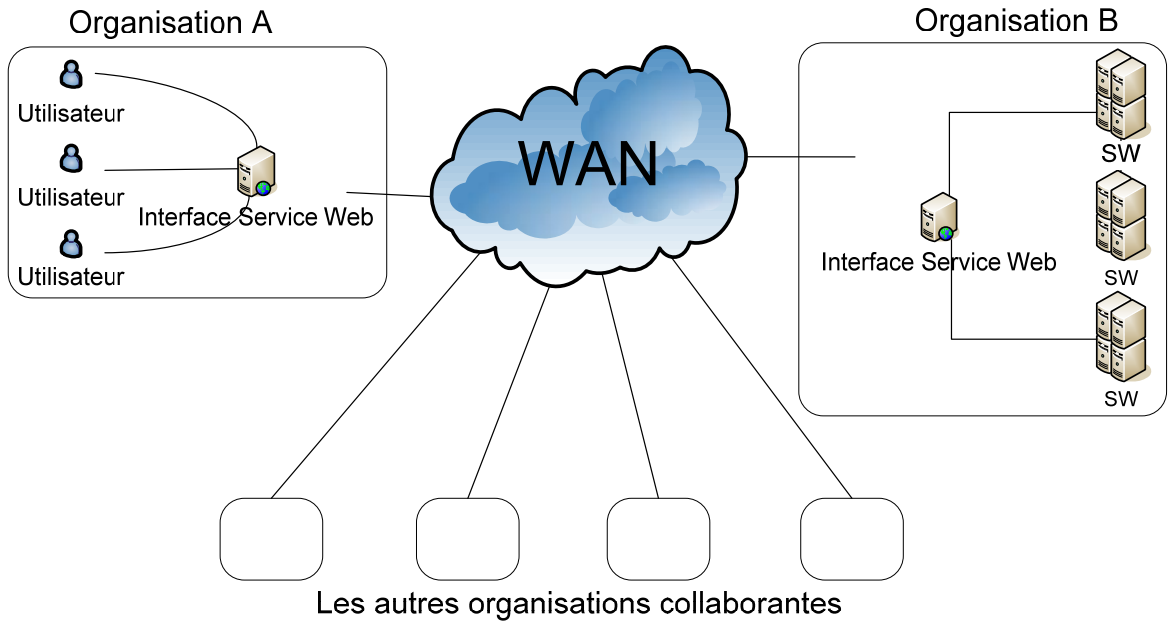


Figure 25 : Collaboration entre les différentes organisations.

La politique globale de sécurité de l'infrastructure d'information critique s'exprime par des règles définies sur chaque service Web en commun entre le client et le prestataire, plutôt que par une super-organisation imposant une politique unique à toutes les organisations. Nous verrons par la suite que ces règles font partie d'un contrat [Abou El Kalam *et al.*, 2008], négocié entre les deux organisations, client et prestataire du service Web. On appellera donc *politique-contrat* l'ensemble des règles de sécurité qui s'appliquent entre les deux organisations interagissant par un service Web. Pour un même service Web, il y a autant de politiques-contrats différentes que d'organisations clientes du service, et ces politiques-contrats peuvent différer selon les particularités de chaque contrat. Pour faciliter la vérification de cohérence syntaxique et sémantique entre la politique-contrat et les politiques locales du client et du prestataire, ces règles et ces politiques sont exprimées dans un même formalisme, compatible avec le modèle OrBAC présenté au chapitre précédent.

Dans PolyOrBAC, la politique globale de sécurité de l'infrastructure est donc constituée par l'ensemble des politiques-contrats en cours dans l'infrastructure, facilitant ainsi l'évolutivité et l'extensibilité de l'IIC. Quand une organisation rejoint l'infrastructure, il suffit d'établir une politique-contrat pour chaque service Web que la nouvelle organisation offre à une autre organisation de l'IIC et pour chaque service Web offert par une autre organisation à la nouvelle organisation. De même quand une organisation quitte l'infrastructure, il suffit d'éliminer les politiques-contrats qui liaient cette organisation aux autres ; pour les services Web qui étaient fournis par l'organisation quittant l'IIC, il peut être nécessaire que les clients établissent de nouvelles politiques-contrats avec d'autres organisations de l'IIC qui fournissent des services équivalents.

Dans cette vision de PolyOrBAC, chaque organisation gère ses propres ressources indépendamment, avec sa propre politique de sécurité locale, même si certaines de ces ressources peuvent être utilisées par des services Web s'exécutant localement pour le compte d'autres

organisations, conformément aux politiques-contrats. Chaque organisation est aussi responsable des actions de ses utilisateurs, actions qu'elle doit contrôler par sa politique locale, même si certaines de ces actions consistent à invoquer des services Web fournis par d'autres organisations, conformément aux politiques-contrats. Chaque organisation doit donc aussi authentifier chacun de ses propres utilisateurs, avec des mécanismes d'une force adaptée au risque d'usurpation d'identité et aux conséquences des abus que pourraient commettre ces utilisateurs. En revanche, il n'est pas utile qu'une organisation connaisse les utilisateurs appartenant à une autre organisation, ce serait même nuisible du point de vue de la confidentialité et de la vie privée.

Par rapport à une politique de sécurité, exprimée grâce à OrBAC, pour une organisation isolée, PolyOrBAC impose que la politique locale d'une organisation participant à une IIC tienne compte des interactions par services Web, pour contrôler l'exécution locale de ces services lorsqu'ils sont invoqués par une autre organisation, et pour contrôler l'invocation par ses propres utilisateurs de services offerts par d'autres organisations. Il faut donc étendre les fonctionnalités d'OrBAC pour prendre en compte ces services Web, et ceci est l'objet de la sous-section suivante.

3.3.2 Images de services Web et utilisateurs virtuels

Dans PolyOrBAC, nous introduisons deux nouvelles notions dans OrBAC concernant l'utilisation des services Web. Afin que la politique locale puisse contrôler l'invocation d'un service Web externe, nous introduisons la notion d'*image de service Web*, qui est une représentation locale du service distant à invoquer. De même, nous introduisons la notion *d'utilisateur virtuel* pour représenter une organisation cliente dans la politique locale de l'organisation qui fournit un service Web.

Prenons un exemple, schématisé par la Figure 26, où Alice, utilisatrice de l'organisation A, invoque un service Web SW1 fourni par l'organisation B.

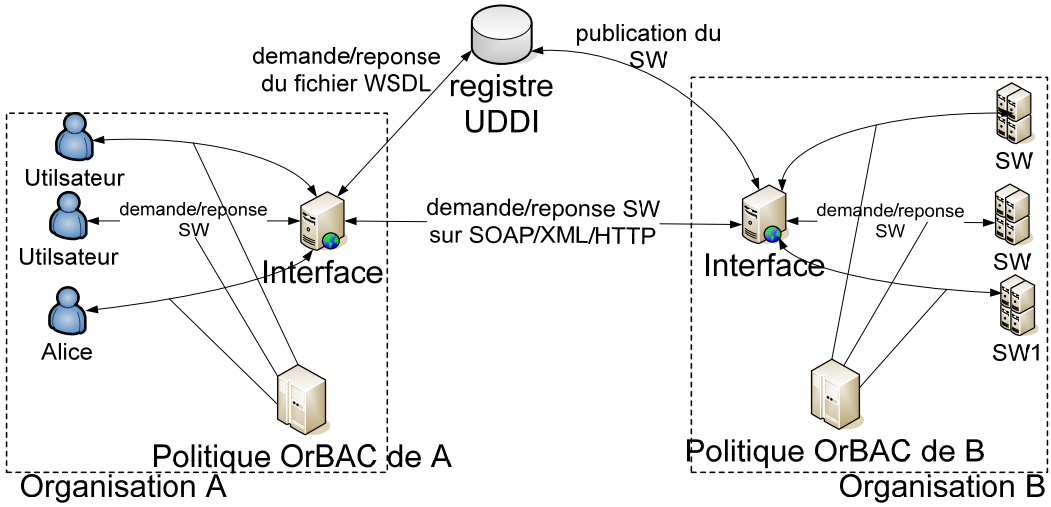


Figure 26 : L'architecture de l'approche PolyOrBAC.

Pour pouvoir exprimer dans la politique locale de A le droit d'invoquer le service Web SW1, l'administrateur de sécurité de A doit créer un objet "Image de SW1", avec une action "invoquer" sur cet objet. Pour qu'Alice puisse invoquer SW1, il faut insérer dans la politique de A une règle OrBAC qui donne à un rôle joué par Alice la permission d'exécuter l'activité correspondant à l'action "invoquer" sur une vue correspondant à l'objet "Image de SW1", et il faut que l'action "invoquer" sur "Image de SW1" exécute un programme qui envoie le message SOAP d'appel du service Web SW1 à l'organisation B (les types de paramètres et l'adresse du destinataire sont définis dans l'entrée UDDI correspondant à SW1, avec éventuellement des contraintes supplémentaires définies dans le contrat signé entre A et B pour SW1). De même, pour que l'exécution correspondant à cet appel de service Web SW1 soit contrôlée dans l'organisation B, il faut que l'administrateur de sécurité de B ait créé un utilisateur virtuel "UV_A" qui joue un rôle lui permettant d'exécuter l'activité correspondant au service Web SW1, et donc qu'il y ait une règle dans la politique locale de B qui donne cette permission à ce rôle. Il faut aussi que la réception de l'appel par l'organisation A au service Web SW1 déclenche l'exécution par UV_A du programme réalisant toutes les actions requises pour SW1 (voir Figure 27).

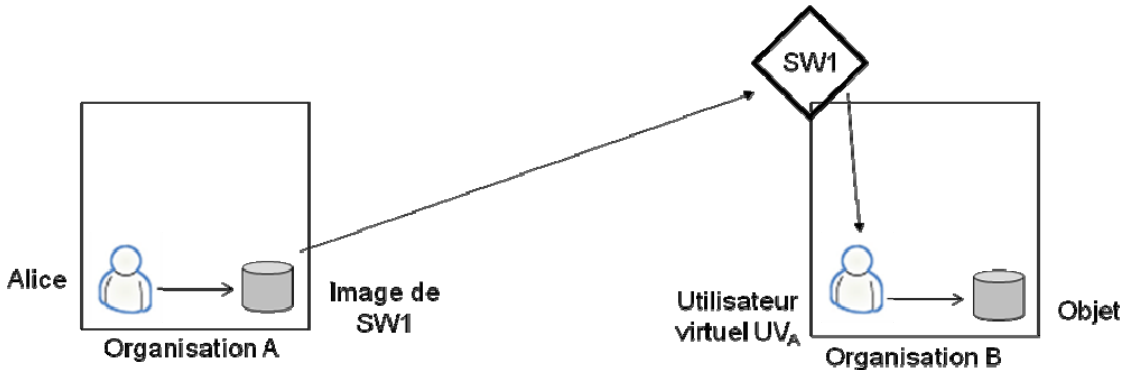


Figure 27 : Image de service Web et utilisateur virtuel.

Par cet exemple, on voit que l'extension d'OrBAC pour les services Web ne concerne que la création dans l'organisation cliente d'objets correspondant à des images de services Web fournis par d'autres organisations, et la création dans l'organisation prestataire d'utilisateurs virtuels correspondant aux organisations clientes. L'image de service Web et l'utilisateur virtuel sont créés lors de la signature du contrat entre l'organisation cliente et l'organisation prestataire. Les entités de base d'OrBAC (organisation, sujet, action, objet, rôle, activité, vue, contexte) ne sont pas changées, les règles de la politique locale concernant les services Web s'expriment donc de la même façon que les autres règles, et les mécanismes de sécurité qui les mettent en œuvre restent les mêmes.

3.3.3 Limites des services Web et nécessité de vérification à l'exécution

Nous avons montré que le modèle OrBAC permet bien d'exprimer les politiques de sécurité locales à chaque organisation participant à une IIC, même pour ce qui concerne les interactions avec d'autres organisations par services Web. Cependant, cela ne permet pas de vérifier si les interactions sont légitimes : les différentes organisations étant mutuellement méfiantes, on peut supposer que l'organisation cliente peut avoir intérêt à abuser des services fournis par l'organisation prestataire, ou que cette dernière peut avoir intérêt à ne pas fournir le service prévu avec la qualité escomptée. Il faut donc vérifier en temps réel à l'exécution si chaque

interaction est légitime, c'est-à-dire si elle obéit aux règles exprimées dans la politique-contrat correspondante, et dans le cas contraire collecter suffisamment d'information pour désigner l'organisation qui doit être tenue responsable de l'abus ou de la fraude. Ceci est l'objet de la section suivante.

3.4 Expression et vérification des interactions par services Web

Dans cette section, nous verrons comment sont négociés les contrats [Abou El Kalam *et al.*, 2008] entre clients et prestataires de services Web, comment les politiques-contrats sont définies dans ces contrats, comment on vérifie en temps réel à l'exécution si les interactions sont conformes ou non avec les politiques-contrats, et en cas de non-conformité, comment on collecte suffisamment d'information pour désigner l'organisation responsable et lui imposer les pénalités prévues au contrat.

3.4.1 Contrats et politiques-contrats

La phase de négociation de contrat consiste en une discussion entre une personne habilitée par l'organisation cliente et une personne habilitée par le prestataire, qui à la fin de la discussion se mettent d'accord sur un contrat commun, et signent le document électronique XML formalisant le contrat. Dans les cas les plus simples, un modèle de contrat sous forme XML peut apparaître dans le registre UDDI, être accepté par le représentant du client et donner lieu à une signature automatique par les deux parties, avec une infrastructure de gestion de clés (*Public Key Infrastructure* ou PKI) commune. La discussion et la signature du contrat doivent suivre un protocole assez simple à standardiser, sous le contrôle de la politique de sécurité de chacune des deux organisations : authentification des utilisateurs habilités à négocier et signer le contrat, vérification de leurs droits, etc. Le contrat doit indiquer précisément les fonctions du service Web et ses caractéristiques (y compris les performances et la qualité de service attendues, la responsabilité de chaque partie, le paiement, des pénalités en cas d'abus, etc.), mais aussi des règles de sécurité relatives à l'invocation et à la fourniture des résultats du service Web, l'ensemble de ces règles de sécurité formant la politique-contrat qui régit les interactions entre client et prestataire. Dans ce qui suit nous nous intéressons plus particulièrement aux politiques-contrats, les autres aspects (fonctionnels, qualitatifs ou financiers) de ces contrats étant hors de portée de cette étude.

Dès que le contrat est signé, chacune des organisations instancient, pour l'une l'image de service Web, et pour l'autre l'utilisateur virtuel, qui serviront aux contrôles d'accès de chacune des politiques locales. La politique-contrat, quant à elle, peut être exprimée en OrBAC, donc avec des permissions, interdictions, obligations ou recommandations, mais portant sur les interactions liées aux services Web, c'est-à-dire sur des échanges de message. Il s'agit donc plutôt de représenter des protocoles de communication par messages, chaque échange de message étant permis, interdit ou obligatoire selon les besoins fonctionnels du service Web, et selon le contexte, c'est-à-dire l'état d'avancement de l'exécution normale du service Web et d'autres informations liées à l'environnement. La politique-contrat d'un service Web peut donc s'exprimer sous forme d'un protocole d'échange de messages avec des modalités (permission, interdiction, obligation). Il est intéressant de représenter ce protocole par un modèle formel, ce

qui facilitera la vérification de certaines propriétés, soit de façon statique, soit à l'exécution. Nous avons choisi de baser notre modèle formel sur les automates temporisés, qui apportent à la fois des avantages de simplicité, et de richesse suffisante pour exprimer et vérifier les propriétés souhaitées.

3.4.2 Expression des politiques-contrats sous forme d'automates temporisés

Les automates temporisés (*timed automata*) ont été proposés pour décrire les comportements de systèmes en tenant compte de caractéristiques temporelles [Alur & Dill, 1994]. En fait, un automate temporisé est un automate fini incluant un ensemble d'horloges (*clocks*), c'est-à-dire des variables réelles et positives qui croissent de manière linéaire avec le temps. Les transitions d'un automate temporisé sont étiquetées par des gardes (*guards*), c'est-à-dire des conditions sur les valeurs d'horloges, des actions et des mises à jour, qui attribuent de nouvelles valeurs aux horloges. Notons que les transitions sont instantanées et que le temps est consommé dans les états : chaque état est étiqueté par un invariant qui représente une condition booléenne sur les horloges ; le temps d'occupation de l'état dépend de l'invariant, l'état est occupé si l'invariant est vrai.

La composition d'automates temporisés est obtenue par un produit synchrone : chaque action *a* exécutée par un automate temporisé correspond à une action avec le même nom *a* exécutée en parallèle par un autre automate temporisé. En d'autres termes, une transition qui exécute l'action *a* ne peut être déclenchée dans un automate que si une transition étiquetée par l'action *a* peut également être déclenchée dans l'autre automate. Les deux transitions sont déclenchées simultanément, ce qui correspond à un mécanisme de communication par *rendez-vous*. Enfin, il est important de noter que la représentation d'un système par un automate temporisé se prête bien à la vérification par modèle (*model-checking*). En particulier, il est possible de représenter une propriété en termes d'accessibilité de certains états, et l'analyse d'accessibilité des états (à partir d'une configuration initiale de l'automate) peut être effectuée automatiquement par des outils de model-checking adaptés.

Pour représenter une politique-contrat avec la richesse des modalités principales d'OrBAC (permissions, interdictions, obligations), il faut représenter ces modalités dans les automates temporisés. C'est l'objet des sections suivantes.

3.4.2.1 Expression des permissions

Tout d'abord, une permission (correpondant à une action qui est autorisée par les clauses de la politique-contrat) est simplement représentée par le biais de transitions dans les automates temporisés. Par exemple, dans la Figure 28, le système peut exécuter l'action *a* à tout moment, puis se comporter selon les possibilités définies dans l'automate A.

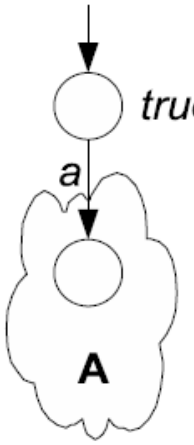


Figure 28 : Modélisation des permissions.

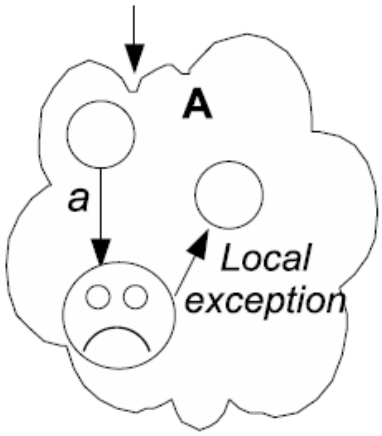


Figure 29 : Modélisation des interdictions.

3.4.2.2 Expression des interdictions

Pour ce qui concerne les interdictions, il faut en distinguer deux types dans les politiques-contrats :

- L'interdiction implicite: comme dans les politiques de contrôle d'accès positives, tout ce qui n'est pas explicitement autorisé est interdit, une action qui ne correspond à aucune transition dans l'automate est interdite.
- L'interdiction explicite: dans notre modèle, une interdiction explicite est représentée par une transition vers un *état d'échec* (illustré par une *frimousse*[10] triste). Les interdictions explicites, présentes dans OrBAC, permettent souvent d'exprimer des règles plus claires qu'une simple absence de permission. En particulier, cela permet d'exprimer des exceptions à des règles de permissions, ou de limiter la propagation des permissions dans les hiérarchies de rôles. Dans notre modèle, la transition vers un état d'échec correspond à la détection d'une action interdite (supposée malveillante), pour laquelle on peut définir un traitement d'exception destiné à contrer l'action malveillante ou à corriger les dégâts qu'elle pourrait avoir causés. Le non-respect d'une interdiction pourra aussi conduire à imposer les pénalités à l'organisation responsable de l'action interdite (voir 3.4.1).

3.4.2.3 Expression des obligations

Dans les politiques-contrats, les obligations correspondent à des actions qui sont nécessaires dans un certain contexte. Du point de vue logique, l'obligation est équivalente à une "interdiction de ne pas faire" et le non-respect d'une obligation est donc équivalent à une action interdite, et en tant que telle peut conduire à un traitement d'exception. Comme une obligation est nécessairement autorisée (une action peut ne pas être à la fois obligatoire et interdite), l'obligation doit être représentée par une transition dans l'automate temporisé. Toutefois, comme les obligations sont sémantiquement plus fortes que les permissions, il faut ajouter d'autres symboles pour décrire cette sémantique et pour faire la distinction entre ce qui est

[10] Le terme est approuvé par la Commission générale de terminologie et de néologie, Journal officiel du 16 mars 1998, pour traduire le terme anglais *smiley*.

obligatoire et ce qui est simplement autorisé mais pas obligatoire. Les obligations sont donc représentées par des transitions auxquelles sont associées des échéances temporelles (*time-outs*) sur des transitions et des invariants d'états : si la transition n'est pas activée avant l'échéance, une autre transition est déclenchée automatiquement vers un état d'échec, qui peut conduire à un traitement d'exception. Dans cette sémantique, l'obligation correspond à une action qui doit être réalisée dans un temps donné, la non-réalisation dans les temps étant équivalente à une interdiction. Ceci est schématisé dans la Figure 30, où *k* est un compteur de temps, *d* est l'échéance temporelle, *b* est l'action obligatoire, *a* est l'action déclenchée par l'échéance, conduisant au traitement d'exception A2. L'état à partir duquel l'action *b* est obligatoire possède un invariant (*k* <= *d*).

3.4.2.4 Identification des situations de conflits

Comme on vient de le voir, quand une interdiction explicite est violée ou lorsqu'une obligation n'est pas remplie, l'automate atteint un état d'échec. Cette situation ne survient que si l'une des deux parties, le client ou le fournisseur du service Web, ne respecte pas les règles de la politique-contrat. La représentation par automate temporisé aide à identifier la partie responsable de cette violation de politique, par l'analyse de la séquence états-transitions qui a conduit à l'état d'échec.

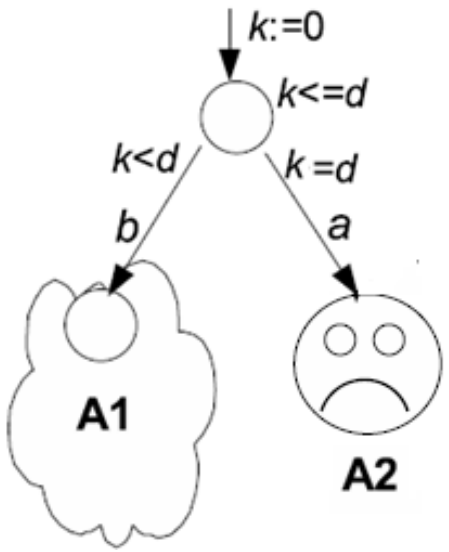


Figure 30 : Modélisation des obligations.

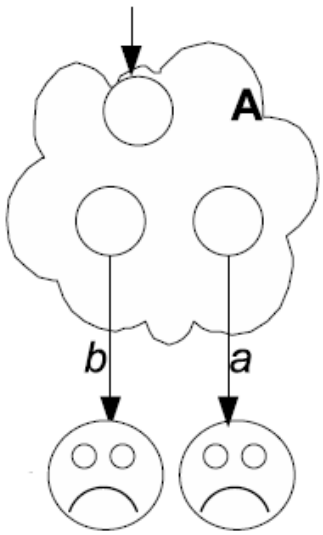


Figure 31 : Modélisation des situations de conflits.

Cette modélisation des différends (situations de conflit) permet donc non seulement d'identifier les anomalies et les abus (violations de droits), mais aussi d'identifier les activités (succession d'actions et d'interactions) qui ont conduit à ces situations, d'en déduire le responsable de l'anomalie (celui qui a réalisé une action explicitement interdite, ou n'a pas réalisé une action obligatoire dans les temps prévus), et donc de lui imposer les pénalités prévues au contrat. De plus, comme les conséquences peuvent être de gravités différentes et conduire à des pénalités graduées, ceci peut être représenté dans l'automate par des étiquettes sur les états d'échec.

3.4.3 Vérification des politiques-contrats

La modélisation des politiques-contrats par des automates temporisés permet de vérifier certaines propriétés par analyse statique. En particulier, la cohérence de la politique correspond au fait que tous les états normaux (c'est-à-dire les états qui ne sont pas des états d'échec), doivent être accessibles, ou autrement dit que le protocole est correct et correspond au fonctionnement normal. L'analyse statique permet aussi d'identifier des scénarios (séquences d'états-transitions) qui conduiraient aux états d'échec, et de vérifier qu'ils correspondent bien à des violations de la politique-contrat.

Mais le principal intérêt de cette modélisation est de permettre une vérification en temps-réel à l'exécution, par ce qu'on appelle *run-time model checking*. Dans le schéma-cadre PolyOrBAC, ceci correspond à vérifier sur chaque échange de messages de service Web (seul moyen d'interaction entre organisations), si le message correspond à une transition autorisée dans le modèle de la politique-contrat, à bloquer tout message qui ne correspond pas à une transition autorisée (interdiction implicite), et enfin à déclencher des actions de recouvrement (traitement d'exception) si un message correspond à une action explicitement interdite, ou si on ne reçoit pas un message correspondant à une obligation avant l'échéance temporelle prévue. La vérification à l'exécution fait donc partie des mécanismes de sécurité mis en œuvre dans PolyOrBAC pour sécuriser les interactions entre organisations, en mettant en application les politiques-contrats.

Du point de vue de l'architecture de l'infrastructure d'information critique, cette vérification doit se faire à l'interface entre organisations, au niveau des messages de services Web. Il est donc naturel de l'implémenter dans les passerelles réseaux ou les pare-feux qui connectent les réseaux internes de l'organisation à un réseau à large étendue (*Wide Area Network*, ou WAN) qui interconnectent toutes les organisations participant à l'infrastructure. Chacune des passerelles doit donc inclure un *run-time model checker*, qui vérifie à chaque réception de message d'un service Web sa compatibilité avec la politique-contrat du service Web. Ceci veut dire qu'à la signature du contrat entre le client et le fournisseur, l'administrateur de sécurité de chacune des deux organisations doit installer dans sa passerelle un modèle de la politique-contrat sous forme d'un automate temporisé. La passerelle sert alors de mécanisme de contrôle d'accès en laissant passer les messages autorisés, en bloquant les messages interdits, mais aussi en générant des messages vers le client ou vers le prestataire pour déclencher les traitements d'exceptions prévus en cas d'interdiction explicite ou de non-respect d'une obligation. Ces passerelles peuvent aussi maintenir un journal de tous les messages échangés pour un service Web. Ces messages étant signés par l'organisation émettrice (en fait, par la passerelle de cette organisation, en utilisant la PKI mentionnée au 3.4.1), ils peuvent servir de preuve devant un juge en cas de violation de la politique-contrat, pour faire imposer à l'organisation responsable de cette violation les pénalités prévues au contrat. Ces passerelles servent donc à la fois au contrôle d'accès et à l'audit de sécurité.

3.5 Récapitulation du fonctionnement général de PolyOrBAC

Cette section reprend le fonctionnement du schéma-cadre PolyOrBAC en décrivant les différentes étapes de la mise en œuvre des interactions entre organisations à l'aide de services Web.

3.5.1 Création et publication d'un service Web par un prestataire

Pour créer un service Web, le prestataire doit d'abord déterminer les ressources qui seront utilisées et mises à disposition par ce service. Puis il faut définir les fonctionnalités du service, c'est-à-dire les actions qu'il sera possible de réaliser sur ces ressources, que ce soit par des utilisateurs internes ou externes. Ensuite il faut implémenter ce service sous forme de programmes capables d'effectuer les actions sur les objets correspondant à ces ressources. L'administrateur de sécurité de l'organisation prestataire définit alors les rôles, activités et vues correspondant à ce service Web, et les liens entre activités et actions et entre vues et objets. Enfin il décrit la politique de sécurité OrBAC qui doit régir ce service Web. Le prestataire peut alors créer une description WSDL du service Web et la publier dans un registre public UDDI afin de faire connaître ce service aux autres organisations. Si un utilisateur habilité (manager) d'une telle organisation souhaite que son organisation puisse bénéficier de ce service Web, il va négocier un contrat avec l'organisation prestataire.

3.5.2 Négociation et signature du contrat

Les deux organisations désignent chacune un utilisateur habilité pour négocier le contrat, sur les aspects fonctionnels, qualité de service, et financiers, puis se mettent d'accord sur une politique-contrat commune, décrite par un automate temporisé, qui satisfasse les besoins de sécurité du service Web. Une fois le contrat signé, l'administrateur de sécurité du client installe le modèle de la politique-contrat dans sa passerelle, crée un objet image du service Web et l'action qui permettra de l'invoquer, ainsi que l'activité et la vue correspondantes, puis ajoute à la politique OrBAC de l'organisation une règle donnant la permission d'exécuter l'activité sur la vue à un rôle ou plusieurs rôles (nouveaux ou préexistants). De même, l'administrateur de sécurité du prestataire installe le modèle de la politique-contrat dans sa passerelle, crée un utilisateur virtuel représentant l'organisation cliente et lui assigne un rôle autorisé à exécuter l'activité correspondant au service Web.

3.5.3 Invocation et exécution du service Web

Après l'étape de négociation, vient l'invocation et l'utilisation du service en question. Le Tableau 5 récapitule les étapes d'invocation du service Web, de la vérification des droits d'accès OrBAC et des interactions par échange de messages de service Web. Lorsqu'Alice, utilisatrice de l'organisation A, veut utiliser le service Web SW1 fourni par l'organisation B, les mécanismes de sécurité de l'organisation A vérifient s'il existe une règle dans la politique de sécurité OrBAC locale de A qui autorise cette demande : il faut pour cela qu'il existe une règle qui accorde, dans

le contexte actuel, la permission d'exécuter l'activité correspondante sur la vue correspondant à un rôle joué par Alice. Si l'invocation est autorisée, un message d'invocation de SW1 est envoyé par l'organisation A à sa passerelle qui reconnaît que ce message correspond à un contrat entre A et B pour SW1, et ce message est analysé par la passerelle en fonction de l'automate temporisé représentant la politique-contrat correspondante. Comme c'est le message d'invocation et qu'il provient de A, la passerelle signe le message et autorise l'envoi de ce message à la passerelle de B ; en parallèle, l'automate évolue vers un état où l'organisation B est obligée de répondre dans un délai fixé.

Permission (A, Rôle1, vue_SW1, accéder, contexte1) ∧ Habilite (A, Alice, Rôle1) ∧ Considère (A, invoquer, accéder) ∧ Utilise (A, image _SW1, vue_SW1) ∧ Définit (A, Alice, invoquer, image _SW1, contexte1) → Est_permis(Alice, invoquer, image_SW1)

Tableau 5 : Représentation de la règle de contrôle d'accès OrBAC du côté client.

La passerelle de B analyse ce message pour reconnaître qu'il vient de A et qu'il existe un automate pour SW1 et A, vérifie le contenu du message par rapport à sa copie de l'automate, et le transmet à l'interface service Web de l'organisation B ; en parallèle, l'automate évolue vers le même état que l'automate de A. La réception du message SOAP par l'interface service Web de B identifie le contrat avec A pour SW1, et provoque la création d'un processus pour le compte de l'utilisateur virtuel UV_A, pour exécuter le service SW1. Ceci correspond à l'activité "fournir" sur la vue "vue_de_SW1", pour laquelle il existe une permission dans la politique locale OrBAC du prestataire pour le Rôle2, et UV_A joue le rôle Rôle2. Les actions réalisant l'exécution du service Web sont donc autorisées, et ont pour résultat d'envoyer un message de réponse de l'organisation B à l'organisation A, avec les mêmes contrôles des politiques-contrat dans les passerelles.

Permission (B, Rôle2, vue_de_SW1, fournir, contexte2) ∧ Habilite (B, UV_A, Rôle2) ∧ Considère (B, exécuter, fournir) ∧ Utilise (B, ressource_SW1, vue_de_SW1) ∧ Définit (B, UV_A, exécuter, ressource_SW1, contexte2) → Est_permis (UV_A, exécuter, ressource_SW1)

Tableau 6 : Représentation de la règle de contrôle d'accès OrBAC du côté prestataire.

Notons que dans ces interactions, aucune des organisations ne voit les utilisateurs de l'autre ni ses ressources. Chaque organisation est donc totalement responsable de l'authentification de ses utilisateurs, et de contrôler les actions qu'ils réalisent. Si un utilisateur malveillant est authentifié (à tort ou à raison) et autorisé par A à invoquer SW1 avec des paramètres frauduleux qui peuvent provoquer des dégâts dans l'organisation B, c'est bien l'organisation A qui sera tenue responsable, et l'organisation B peut en apporter la preuve en montrant une copie du message signé par A correspondant à cette invocation. De la même façon, chaque organisation peut protéger ses ressources comme elle l'entend vis-à-vis de ses propres utilisateurs et vis-à-vis des requêtes de service Web venant d'autres organisations. Mais

elle est tenue par contrat à fournir des services, et si elle ne le fait pas (avec la qualité de service prévue au contrat), le client pourra en apporter la preuve en présentant la séquence des messages reçus à un juge[11].

3.6 Conclusion du chapitre 3

Dans ce chapitre, nous avons proposé une nouvelle approche que nous avons nommé PolyOrBAC. Cette approche se base sur deux composants principaux : le modèle de contrôle d'accès basé organisation, OrBAC, qu'on utilise pour spécifier des politiques de contrôle d'accès pour les différentes organisations concernées, et la technologie des services Web qui permet de fournir une plateforme de collaboration et d'interopérabilité entre ces mêmes organisations.

Dans le chapitre suivant, nous allons montrer comment ce schéma-cadre peut s'appliquer à une infrastructure critique, le réseau de transport et de distribution d'énergie électrique en Europe. Nous analyserons le fonctionnement de ce type d'infrastructure et décrirons des scénarios montrant les interactions entre organisations faisant partie de cette infrastructure. L'application de PolyOrBAC dans cet exemple illustrera les avantages de cette approche.

[11] Le protocole d'échange de messages doit prendre en compte les pertes de messages sur le réseau, les réémissions de messages, etc., à mettre en œuvre pour fiabiliser la connexion. En dernier ressort, c'est le juge qui décidera de la responsabilité de chacune des parties, au vu des séquences de messages envoyés et reçus par chacune d'elles.

Chapitre 4. Étude de cas et application de PolyOrBAC

Dans le chapitre précédent, nous avons proposé un nouveau schéma-cadre, PolyOrBAC, pour le contrôle d'accès collaboratif. Notre approche se base sur deux composants principaux : le modèle de contrôle d'accès basé organisation, OrBAC qui est utilisé pour spécifier des politiques de contrôle d'accès pour les différentes organisations concernées, et la technologie des services Web qui permet de fournir une plateforme de collaboration et d'interopérabilité entre ces mêmes organisations.

Dans ce chapitre, nous allons montrer comment PolyOrBAC peut s'appliquer aux infrastructures critiques, en prenant pour exemple les infrastructures de production, de transport, et de distribution d'énergie électrique. Dans un premier temps, nous donnerons une description détaillée de l'application et du fonctionnement de ce type d'infrastructure, et nous présenterons des scénarios mettant en évidence le fonctionnement de ces infrastructures en tenant compte des différents acteurs et composants communiquant en leur sein. Dans un deuxième temps, il s'agit de montrer comment PolyOrBAC est applicable du point de vue spécification de politiques de contrôle d'accès pour chaque organisation collaborant au sein de l'IIC et du point de vue collaboration entre ces mêmes organisations.

4.1 Infrastructure de production, transport, et distribution d'énergie électrique

Afin de bien comprendre le fonctionnement des infrastructures de production, transport, et distribution d'énergie électrique [Knight *et al.*,. 1998], nous allons dans un premier temps donner une description simplifiée de ce type d'infrastructure, que pour simplifier, nous appelons IPE. Une IPE représente un réseau logique et physique constitué d'un ensemble de fournisseurs et de consommateurs d'énergie, connectés par des lignes de transport et de distribution, l'ensemble étant contrôlé par un ou plusieurs centres de contrôle (voir la Figure 32).

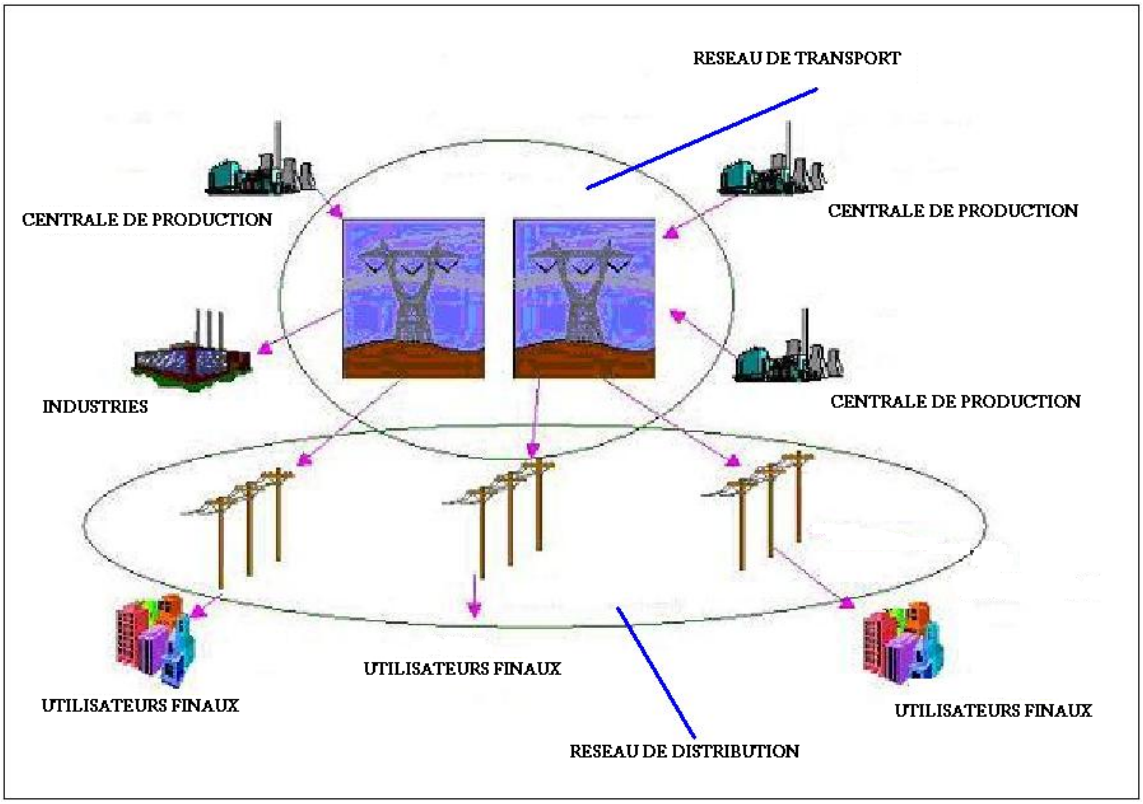


Figure 32 : Infrastructure de production d'électricité [Garrone et al., 2007].

À la sortie des principales centrales de production, l'électricité est portée à très haute tension (typiquement 400.000 Volts ou 225.000 Volts) pour être transportée sur de longues distances. Jusqu'au consommateur final, l'énergie électrique circule en empruntant différents réseaux de lignes aériennes et souterraines de niveaux de tension décroissants : le réseau de transport d'électricité au niveau international, national et régional, puis les réseaux de distribution locale exploités par les distributeurs d'électricité [RTE, 2005]. Les différents réseaux sont interconnectés par des sous-stations de transformation. La Figure 33 donne une vue détaillée de l'infrastructure d'information et de communication de l'infrastructure de production d'électricité sur laquelle nous allons nous basé pour l'étude du scénario réel.

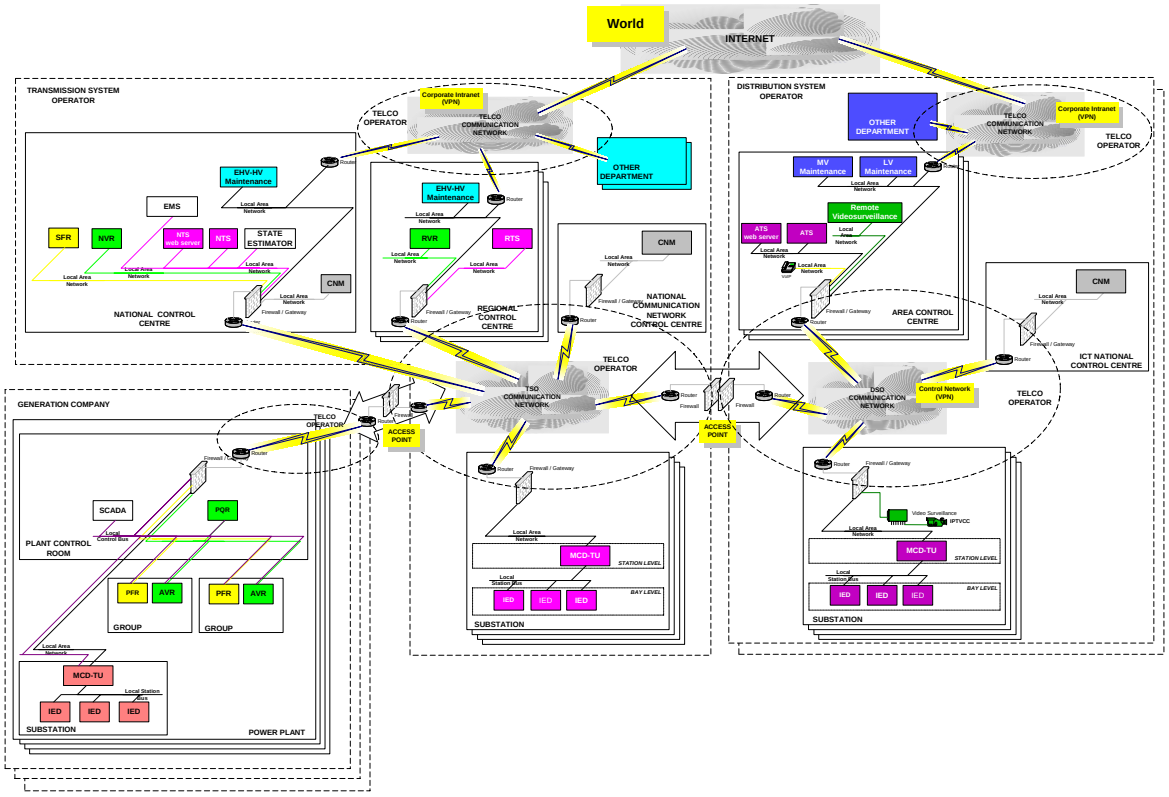


Figure 33 : Infrastructure d'information et de communication d'une IPE [Garrone et al., 2007]

Nous donnons ici (Figure 34) une description plus abstraite de l'IPE afin de montrer les différentes connections entre les différents composants de l'IPE (les compagnies de génération d'énergie, le réseau de transport d'énergie, le réseau de distribution d'énergie, les centres de contrôle nationaux, régionaux, et locaux, et les différentes sous stations).

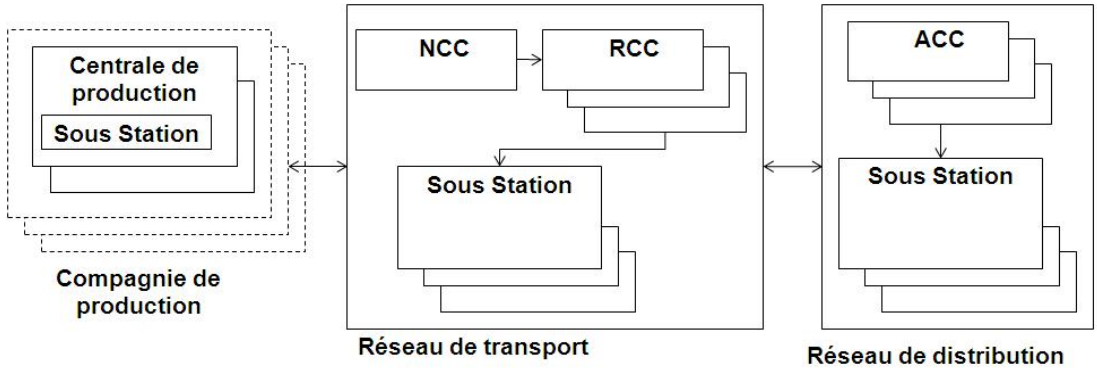


Figure 34 : Architecture générale d'une IPE.

Comme le montre la Figure 34, une ou plusieurs compagnies de production d'électricité sont responsables de plusieurs centrales électriques, connectées à un ou plusieurs réseaux de transport. Chaque réseau de transport se compose de lignes à très haute tension et de sous-stations de transport et est relié à des réseaux de distribution. Chaque réseau de distribution est composé de lignes à moyenne et basse tension et de sous-stations de distribution, et distribue de l'électricité aux abonnés (industries, secteur tertiaire, habitations, etc.). Un réseau de transport est géré par un centre de contrôle national (NCC pour *national control center*) et plusieurs centres de contrôle régionaux (RCC pour *regional control center*) tous contrôlés par un opérateur

TSO (*transmission system operator*). Un réseau de distribution est géré par un centre de contrôle local (ACC pour *area control center*) contrôlé par un opérateur DSO (*distribution system operator*) qui peut agir sur des MCD-TU (pour *Monitoring and Control Data Terminal Unit*, qui peut être traduit par unité terminale de surveillance et de commande) dans les différentes sous-stations sous sa responsabilité. Dans la section suivante, nous détaillons un exemple de scénario illustrant la collaboration ces organisations et ces acteurs dans un processus de délestage de charge en situation d'urgence.

4.2 Description du scénario de délestage

Dans un premier temps nous étudions l'architecture de la partie de l'infrastructure de production d'électricité (voir Figure 35) sur laquelle porte le scénario, et dans un deuxième temps, nous présentons le scénario pratique.

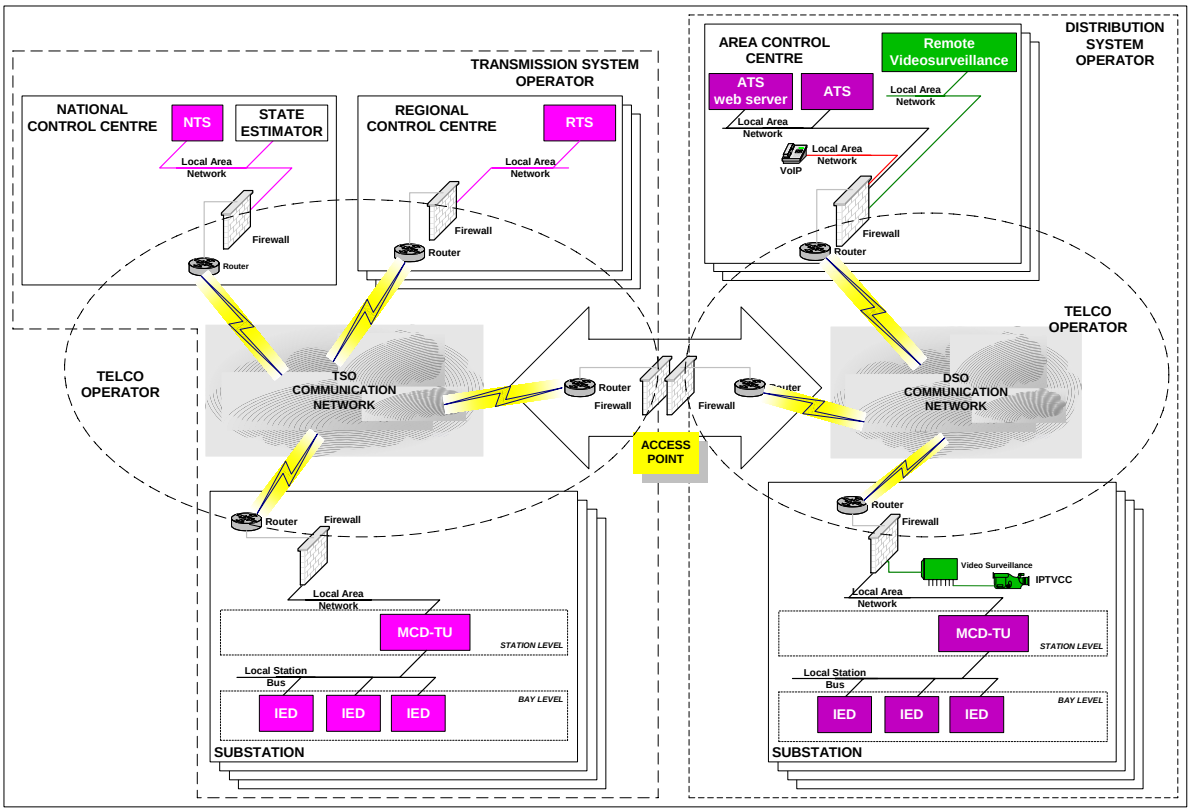


Figure 35 : Système de contrôle à distance des TSO et DSO [Garrone et al., 2007].

4.2.1 Architecture de l'infrastructure

Ce scénario considère la possibilité de défaillances en cascade liées aux menaces relatives aux technologies de communication et d'information qui pèsent sur les communications entre les centres de contrôle des TSO et DSO et leurs sous-stations dans des situations d'urgence (par exemple, lors de surcharges de lignes, ou lorsqu'une baisse de fréquence sur le système de transport risque de faire "décrocher" une centrale de production, ce qui accroîtrait le déséquilibre production-consommation, et provoquerait une défaillance en cascade qui

conduirait à un black-out). Dans ce type de situation, un délestage de la distribution d'électricité au niveau de certaines sous stations peut d'avérer nécessaire pour éviter un black-out.

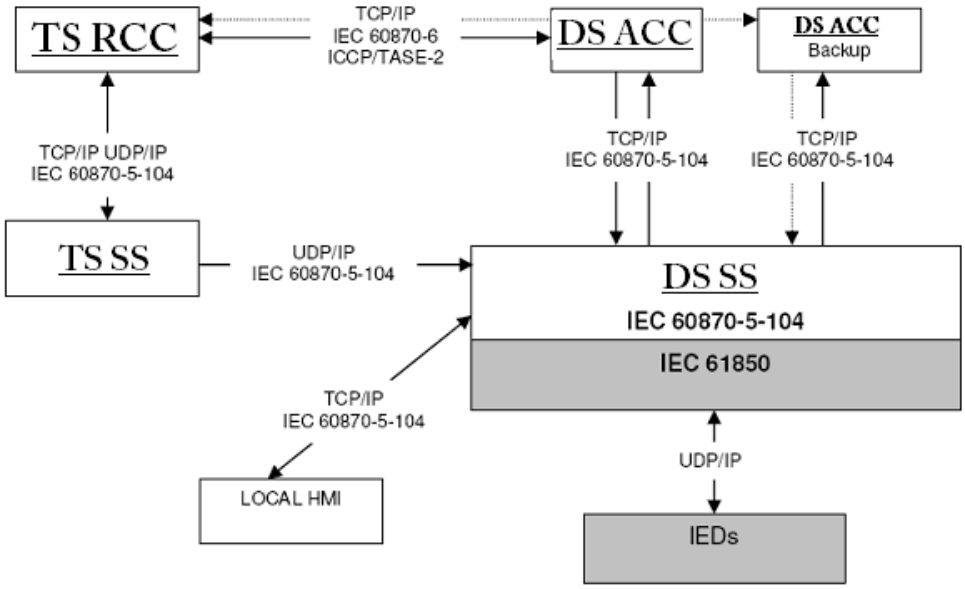


Figure 36 : Les flux d'information et protocoles de communication [Garrone et al., 2007].

L'objectif principal du scénario de délestage est d'analyser les problèmes de sécurité des systèmes d'information et de communication dans un cas concret. Ce scénario traite la sécurité des communications et des interactions entre les opérateurs de transport TSO et de distribution DSO dans des conditions d'urgence. La Figure 36 montre les différents liens entre le TSO, le DSO et leurs sous-stations respectives. Tout d'abord, nous énumérons les différents acteurs participants au scénario de délestage.

Les acteurs concernés sont :

- Le TSO (Transmission System Operator)
- Le DSO (Distribution System Operator)

Les systèmes de contrôle informatique concernés sont :

- Le centre de contrôle régional de transport (TS RCC, pour *Transmission System Regional Control Center*) où opère le TSO. Par simplification, nous utilisons par la suite la notation « TS CC ».
- Une sous-station de transport (TS SS, pour *Transmission System Substation*).
- Le centre de contrôle de distribution (DS ACC, pour *Distribution System Area Control Center*) où opère le DSO. Par simplification, nous utilisons par la suite la notation « DS CC »
- Des sous-stations de distribution (DS SS, pour *Distribution System Substations*).

Les flux d'information échangés sont :

- Les mesures électriques (puissance, tension, fréquence, etc.) transmises du DS CC vers le TS CC.

- Les signaux échangés entre TS CC et DS CC.
- Les commandes de la sous-station de transport aux sous-stations de distribution.
- Les demandes d'armement ou désarmement du centre de contrôle TS CC au centre de contrôle DS CC.

Les menaces à prendre en compte sur le système d'information et de communication sont :

- Les attaques, en déni de service ou autres, contre les sous-stations, lancées par des attaquants situés au niveau des centres de contrôle du transport (TS CC) ou de la distribution (DS CC).

- Les attaques sur les communications entre les centres de contrôle et les sous-stations, avec éventuellement l'envoi de commandes fictives.

Les effets en cascade et les contre-mesures correspondantes à prendre en compte sont :

- Des cyber-attaques peuvent être lancées alors que le réseau électrique se trouve dans des situations d'urgence ; vis-à-vis de cette menace, des contre-mesures de défense doivent être prévues, en particulier des actions de délestage automatique, afin d'éviter de graves dommages.

- Les effets de l'attaque informatique sur le système électrique dépendent du type et du nombre de composants impliqués. Pour éviter qu'une attaque n'ait de conséquence sérieuse sur le fonctionnement de l'infrastructure électrique, des techniques de diversification et de tolérance aux intrusions ont été développées dans le cadre du projet CRUTIAL, mais nous ne les développerons pas ici.

4.2.2 Déroulement du scénario de délestage

Le scénario que nous avons choisi concerne le délestage semi-automatique de la distribution d'électricité dans certaines zones en cas de problèmes de sous-production, de surconsommation, ou encore de coupure de ligne de transport : en cas d'urgence, pour éviter un effet de cascade qui conduirait à un black-out, il faut réduire la distribution dans des zones non critiques. Ce scénario est donc particulièrement intéressant puisqu'il est susceptible d'être la cible d'attaquants désirant provoquer un black-out (voir Figure 37).

La Figure 37 présente de façon simplifiée les organisations concernées par ce scénario et les signaux les plus importants échangés entre ces organisations dans des conditions normales et dans des situations d'urgence.

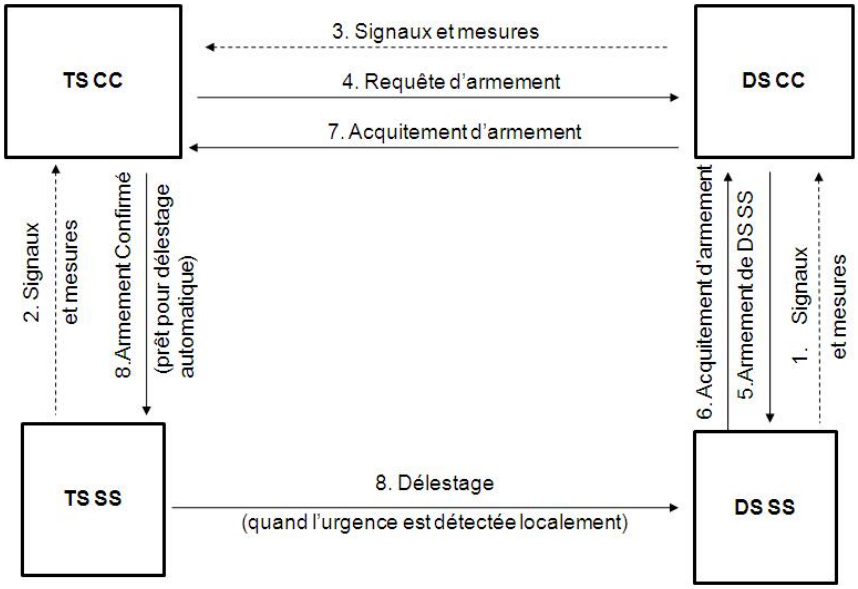


Figure 37 : Signaux et mesures échangés dans le scénario de délestage.

En fonctionnement normal, toutes les sous-stations DS SS envoient des signaux et mesures (puissance, tension, fréquence) au DS CC (1), qui lui-même transmet des signaux et mesures au TS CC (3). De même, les sous-stations TS SS envoient différents signaux et mesures à leur TS CC (2).

Le TS CC surveille le réseau électrique et son TSO, au vu des mesures et signaux reçus ou d'informations provenant du centre de contrôle national, peut identifier certaines situations d'urgence potentielles, qui pourraient nécessiter de réduire la puissance distribuée. Il prépare donc un plan de défense pour délester des zones de distribution particulières.

Pour mettre en œuvre ce plan de défense, le TSO envoie à chaque DSO des zones concernées des demandes de préparation de réduction de puissance (4). Le DSO en fonction de la réduction de puissance demandée et des conséquences relatives des coupures de courant dans les zones industrielles ou d'habitation au moment du délestage éventuel, sélectionne des sous-stations DS SS et le DS CC leur envoie des ordres d'armement (5). Ces DS SS arment alors leurs MCDTU et envoient un accusé de réception au DS CC (6).

Lorsque suffisamment de DS SS sont armées pour pouvoir délester le réseau de distribution de la puissance demandée, le DS CC envoie un accusé d'armement au TS CC (7). L'EMS[12] au niveau du TS CC envoie alors automatiquement un signal à la sous-station de transport TS SS de ce réseau de distribution (8) pour préparer un délestage automatique éventuel.

Pendant ce temps, s'il le juge utile, le DSO peut armer des DS SS et en désarmer d'autres, à condition de maintenir la puissance délestable à un niveau supérieur à celui de la demande faite par le TSO. En cas de détection d'une véritable situation d'urgence (déséquilibre entre la puissance disponible et la puissance consommée qui se traduit par une réduction de fréquence), un processus sentinelle automatique au niveau de la TS SS diffuse automatiquement un ordre de

[12] Energy Management System (Système de gestion d'énergie automatique)

délestage (8) à toutes les DS SS de son réseau de distribution, et seules les DS SS armées précédemment déclenchent le délestage sur leurs MCDTUs.

En revanche, si les conditions pouvant mener à un délestage disparaissent, le TSO peut envoyer une annulation de la demande de réduction de puissance aux DSO, qui eux-mêmes envoient un ordre de désarmement à toutes les DS SS armées.

La Figure 38 présente un diagramme de séquence UML qui récapitule les étapes du processus pour l'armement et le délestage des sous stations de distributions.

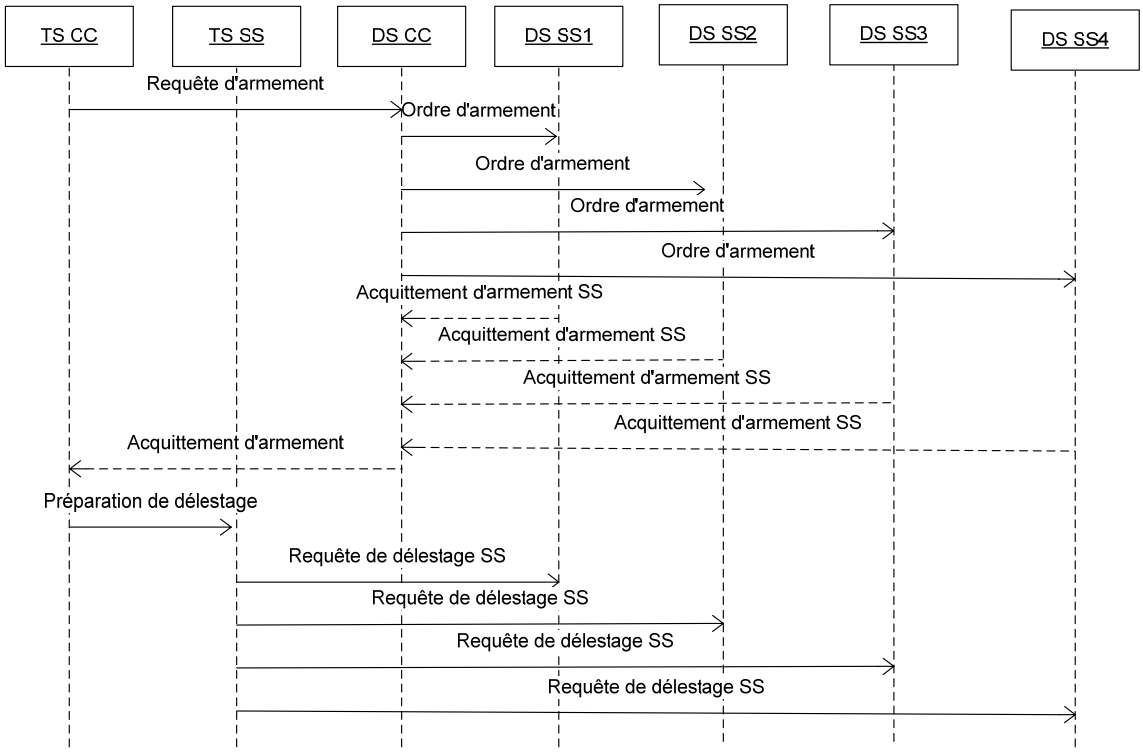


Figure 38 : Diagramme de séquence du scénario de délestage.

4.3 Architecture de l'infrastructure d'information critique

Pour montrer en détail comment on peut appliquer le schéma-cadre PolyOrBAC, nous allons nous baser sur l'architecture d'IIC (voir Figure 39) développée dans le cadre du projet CRUTIAL [Veríssimo *et al.*, 2008]. Dans cette architecture, une IIC peut être vue comme un réseau informatique étendu (*Wide Area Network*, ou WAN) interconnectant plusieurs réseaux locaux (*Local Area Network*, ou LAN) par des composants logiques appelés CIS (pour *CRUTIAL Information Switch*).

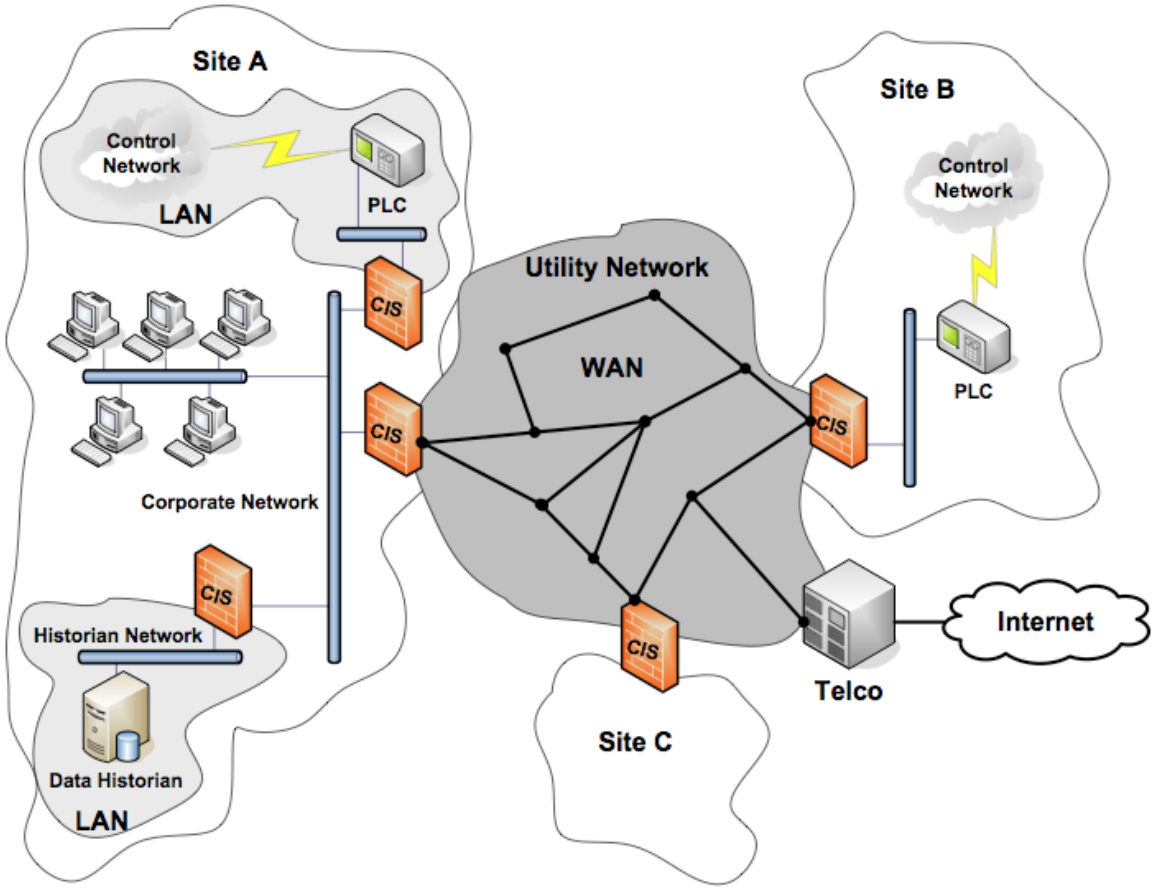


Figure 39 : Architecture CRUTIAL d'une infrastructure d'information critique.

Les CIS sont en fait intégrés à des équipements physiques comme des routeurs, passerelles, pare-feux ou autres, situés à la frontière entre le réseau local d'une organisation et le reste du monde. Le LAN d'une organisation peut comporter différents acteurs (par exemple, les centrales de production d'électricité, départements financiers, fournisseurs externes de service, etc.) et se compose d'un ou de plusieurs systèmes d'information et de communication. Une même entreprise, voire un même établissement peut comporter plusieurs LAN, mais dans PolyOrBAC chaque LAN correspond à une organisation. Chaque organisation propose ses services (l'accès et la possibilité de réaliser des actions sur des ressources internes) à d'autres organisations et chaque organisation possède ses propres applications (et installations logiques et physiques) et sa propre politique de sécurité. Dans la vision PolyOrBAC, les mécanismes prévus pour sécuriser les interactions entre les différentes organisations collaborant sont mis en œuvre au niveau des CIS interconnectant ces organisations.

4.4 Vision PolyOrBAC du scénario de délestage

4.4.1 Définition des services Web

Dans le scénario de la section 4.2, nous avons considéré différentes organisations :

- le centre de contrôle de transport (TS CC) où opère le TSO,
- une sous-station de transport (TS SS),
- le centre de contrôle de distribution (DS CC) où opère le DSO,
- des sous-stations de distribution (DS SS).

Dans l'architecture CRUTIAL, le système d'information et de communication de chacune de ces organisations est constitué d'un LAN sur lequel sont branchés divers équipements informatiques, interconnecté par un CIS aux autres organisations. Nous distinguons les différents services Web (voir Tableau 6) impliqués dans le scénario de délestage, en déterminant le client et le prestataire de chaque service.

Dans ce scénario, nous identifions quatre organisations (TS CC, DS CC, TS SS, DS SS) et cinq services (SW1-demande_d'armement, SW2-ordre_d'armement, SW3-préparation_de_délestage, SW4- activation_de_délestage, SW5-réintégration que nous ne détaillons pas outre-mesure du fait que c'est un service interne au niveau des DS SS).

Service	Prestataire	Client
SW1-demande_d'armement	DS CC	Opérateur TSO
SW2-ordre_d'armement	DS SS	Opérateur DSO
SW3-préparation_de_délestage	TS SS	EMS[13] au niveau du TS CC
SW4- activation_de_délestage	DS SS	Processus sentinelle de la TS SS
SW5-réintégration	DS SS	Opérateur DSO

Tableau 6 : Services Web utilisés dans le scénario.

Au sein de chaque organisation, les mécanismes de contrôle d'accès mettent en œuvre la politique locale de contrôle d'accès OrBAC, en particulier pour ce qui concerne l'invocation par un utilisateur local (réel ou virtuel) de services Web fournis par les autres organisations, les échanges de messages de service Web étant contrôlés par le CIS de chaque organisation, en fonction de la politique-contrat associée à ce service Web. C'est ce que l'on détaille ci-après.

[13] Energy Management System (Système de gestion d'énergie automatique).

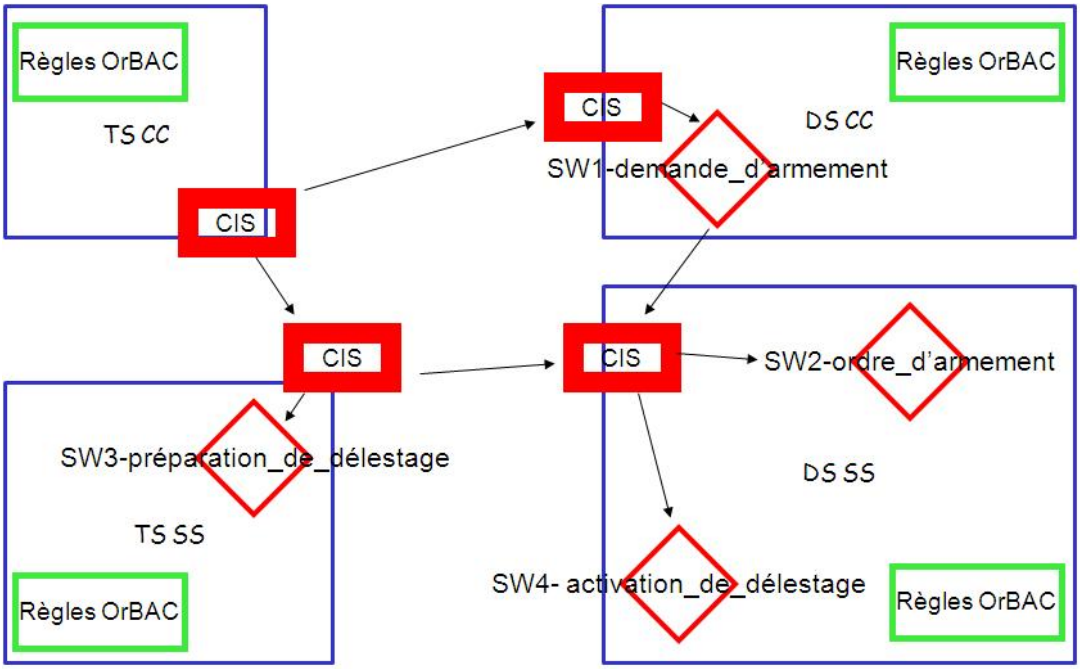


Figure 40 : Application du modèle PolyOrBAC au scénario de délestage.

Afin de mieux comprendre le couplage de la technologie des services Web et des politiques de contrôle d'accès OrBAC dans PolyOrBAC, nous allons dans ce qui suit détailler l'utilisation de certains de ces services Web, avec la vérification des différentes règles de contrôle d'accès définies dans la politique de sécurité de chaque organisation participant et la vérification des échanges de messages de service Web correspondants.

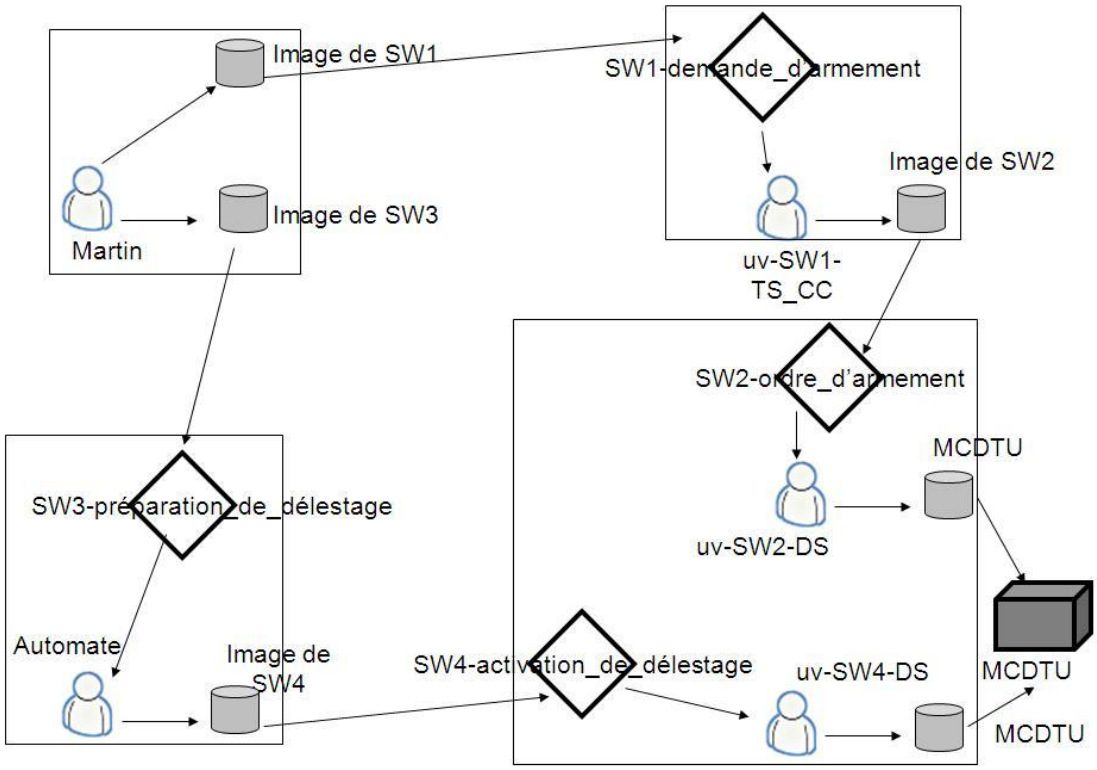


Figure 41 : Application des notions d'image de service Web et d'utilisateur virtuel au scénario.

4.4.2 Contrôle d'accès et vérification pour SW1-demande_d'armement

Le service Web SW1-demande_d'armement gère les interactions entre le TS CC et le DS CC. Ces interactions correspondent à la demande d'armement par le TSO (donc l'organisation TS CC) au DSO (donc l'organisation DS CC), ainsi qu'à la demande de désarmement (si les conditions de pré-urgence disparaissent), et à l'ordre de redémarrage (si le délestage a été réalisé).

4.4.2.1 Règles OrBAC pour la demande d'armement par le TS CC

Dans un premier temps, l'opérateur TSO constate un état de pré-urgence et envoie la requête de demande d'armement (SW1-demande_d'armement) au DS CC. Lorsque la personne (par exemple, Martin) qui est l'opérateur TSO invoque le service SW1, cela correspond à l'exécution de l'action *invoquer* sur l'objet *image_SW1*. Cette action est permise par la politique OrBAC du TS CC si l'utilisateur Martin a été authentifié, s'il joue bien le rôle TSO, et s'il existe une règle de permission pour le rôle TSO de réaliser l'activité correspondant à cette action sur la vue correspondant à cet objet dans le contexte de pré-urgence actuel. La séquence de règles OrBAC suivante présente les droits d'accès pour l'utilisation du service SW1 au niveau de l'organisation TS CC.

Permission(TS CC, TSO, accéder, vue_SW1, pré-urgence) ∧ Habilite(TS CC, Martin, TSO) ∧ Considère(TS CC, invoquer, accéder) ∧ Utilise(TS CC, image_SW1, vue_SW1) ∧ Définit(TS CC, Martin, invoquer, image_SW1, pré-urgence) ⇨ Est_permis(Martin, invoquer, image_SW1)

Tableau 7 : Représentation de la règle OrBAC pour SW1 du côté TS CC.

Cette séquence peut être interprétée comme suit : la politique OrBAC du TS CC contient un certain nombre de règles pour gérer le service Web SW1-demande_d'armement. Première règle : au niveau du TS CC, le rôle TSO a le droit, dans un contexte de pré-urgence, d'exécuter les actions incluses dans l'activité *accéder* sur la vue *vue_SW1* correspondant à SW1. Deuxième règle : nous attribuons le rôle TSO à Martin. Troisième règle : nous incluons l'action *invoquer* dans l'activité *accéder*. Quatrième règle : nous incluons l'objet *image_SW1* dans la vue *vue_SW1*. Cinquième règle : nous précisons que le sujet Martin a la possibilité de réaliser l'action *accéder* sur l'objet *image_SW1* dans un contexte spécifique pré-urgence. De cet ensemble de règles on déduit le prédicat Est_permis qui affirme que le sujet Martin a le droit d'exécuter l'action *accéder* sur l'objet *image_SW1*.

Par ailleurs, comme nous l'avons précisé au chapitre 3, pour chaque service Web un contrat doit être signé par le client et le prestataire, contenant la politique-contrat qui contrôle les échanges de messages. Cette politique-contrat est définie par deux automates temporisés, l'un du côté client, l'autre du côté prestataire, chacun installé dans le CIS de son organisation. La section suivante présente l'automate temporisé correspondant à SW1-demande_d'armement du côté du TS CC.

4.4.2.2 Automate de SW1-demande_d'armement au niveau du TS CC

Selon la politique-contrat du service SW1 et comme illustré dans la Figure 42, du côté du TS CC (le client de SW1), l'automate attend une invocation pour une demande d'armement (en provenance du TSO). Quand le message correspondant à cette invocation est intercepté par le CIS du TS CC, la transition correspondante (*WS1-arming-request*) est activée dans l'automate, une minuterie (*timer*) est initialisée (à 10 mn dans l'exemple de la figure) et l'automate atteint un état où il attend un acquittement *WS1-arming-request-ack* du DS CC. Si le délai expire sans avoir reçu l'accusé de réception de la part du DS CC, un message *WS1-arming-request-error* est envoyé au TS CC pour déclencher un traitement d'exception correspondant à cette situation, et l'automate atteint l'état d'échec *arming_failure*[14]. Inversement, dans des situations normales, lorsque le CIS du TS CC reçoit l'acquittement *WS1-arming-request-ack*, son automate va vers l'état où il sera prêt pour une action d'urgence, alors que la réception du message par le TS CC déclenche automatiquement le processus EMS, qui lui même invoquera le service SW3.

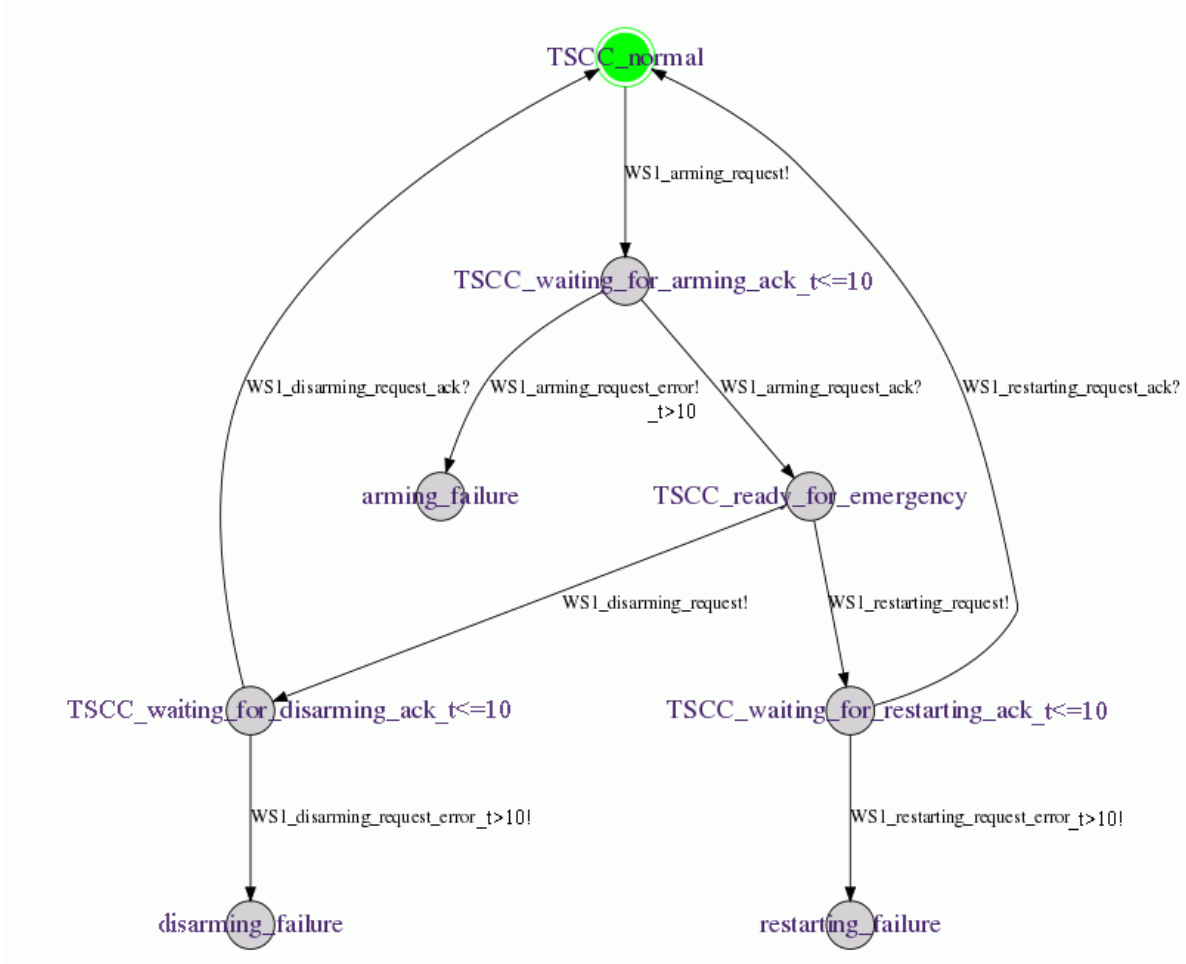


Figure 42 : Automate de SW1-demande_d'armement au niveau du TS CC.

Dans cet état, si la situation de pré-urgence disparaît, le TSO peut décider de désarmer l'ensemble des sous-stations qui ont été armées. Pour cela, le TS CC envoie le message *WS1-disarming-request* vers le DS CC et attend l'acquittement *WS1-disarming-request-ack* du DS CC,

[14] Par souci de simplicité, les automates des traitements d'exception ne sont pas décrits dans le présent document.

qui remettra l'automate dans l'état initial. Si le délai (fixé à 10 unités de temps dans la Figure 42) expire sans avoir reçu l'acquittement *WS1-disarming-request-ack* du DS CC, un traitement d'exception similaire au précédent est déclenché. Au contraire, si la situation d'urgence survient et que le TS SS déclenche le délestage, le TSO doit surveiller la fin de l'urgence, c'est-à-dire le moment où il peut demander le redémarrage du service au DS CC, qui reconnectera les DS SS qui ont délesté. Cette action se traduit par l'envoi du message *WS1-restarting-request*, puis l'attente de la confirmation *WS1-restarting-request-ack*, avant de revenir à l'état initial.

4.4.2.3 Règles OrBAC de SW1-demande_d'armement dans la politique du DS CC

Du côté du DS CC, la réception de la requête de SW1-demande_d'armement déclenche l'exécution pour le compte de l'utilisateur virtuel *uv-SW1-TS_CC* d'un processus qui affiche un message urgent sur la console du DSO. Cet accès (*uv-SW1-TS_CC* à la console) est vérifié selon la politique de contrôle d'accès du DS CC et est accordée en fonction de la séquence OrBAC décrite dans le Tableau 8.

Permission(DS CC, Rôle_SW1, afficher, console_DSO, pré-urgence) ∧
Habilite(DS CC, uv-SW1-TS_CC, Rôle_SW1) ∧
Considère(DS CC, écrire, afficher) ∧
Utilise(DS CC, terminal_1, console_DSO) ∧
Définit(DS CC, uv-SW1-TS_CC, écrire, terminal_1, pré-urgence)
⇨ Est_permis(uv-SW1-TS_CC, écrire, terminal_1)

Tableau 8 : Représentation de la règle OrBAC pour SW1 du côté DS CC.

Première règle : le rôle *Rôle_SW1* a le droit, quelque soit le contexte, d'afficher des messages d'urgence (activité *afficher*) sur la console du DSO (vue *console_DSO*). Deuxième règle : nous attribuons le rôle *Rôle_SW1* au sujet *uv-SW1-TS_CC*, qui représente le TS CC pour exécuter le service SW1. Troisième règle : nous incluons l'action *écrire* dans l'activité *afficher*. Quatrième règle : nous incluons l'objet *terminal_1* dans la vue *console_DSO*. Cinquième règle : nous précisons que le sujet *uv-SW1-TS_CC* a la possibilité de réaliser l'action *écrire* sur l'objet *terminal_1*, quelque soit le contexte. De cet ensemble de règles, on déduit le prédicat Est_permis qui affirme que le sujet *uv-SW1-TS_CC*, a le droit d'exécuter l'action *écrire* sur l'objet *terminal_1*.

4.4.2.4 Automate de SW1-demande_d'armement dans le CIS du DS CC

Nous allons maintenant analyser la façon dont l'automate de la politique-contrat liée à SW1-demande_d'armement vérifie dans le CIS du DS CC les invocations en provenance du côté client (Figure 43). Au départ, cet automate est en attente du message *WS1-arming-request* du TS CC. Quand ce message est intercepté, une minuterie est initialisée (à la valeur de 10 mn dans l'exemple de la figure), et l'automate atteint un état où il attend la réalisation d'une obligation, l'armement de suffisamment de DS SS, qui se traduira par l'envoi d'un acquittement *WS1-arming-request-ack* par le DS CC généré automatiquement par le processus lancé par le DSO pour armer les DS SS. Si ce message n'est pas intercepté par le CIS du DS CC avant que le délai maximum pour la réception définie par la minuterie ne se déclenche, l'automate atteint un

état d'échec (avec envoi du message *WS1-arming-request-error* au DS CC pour déclencher un traitement d'exception.

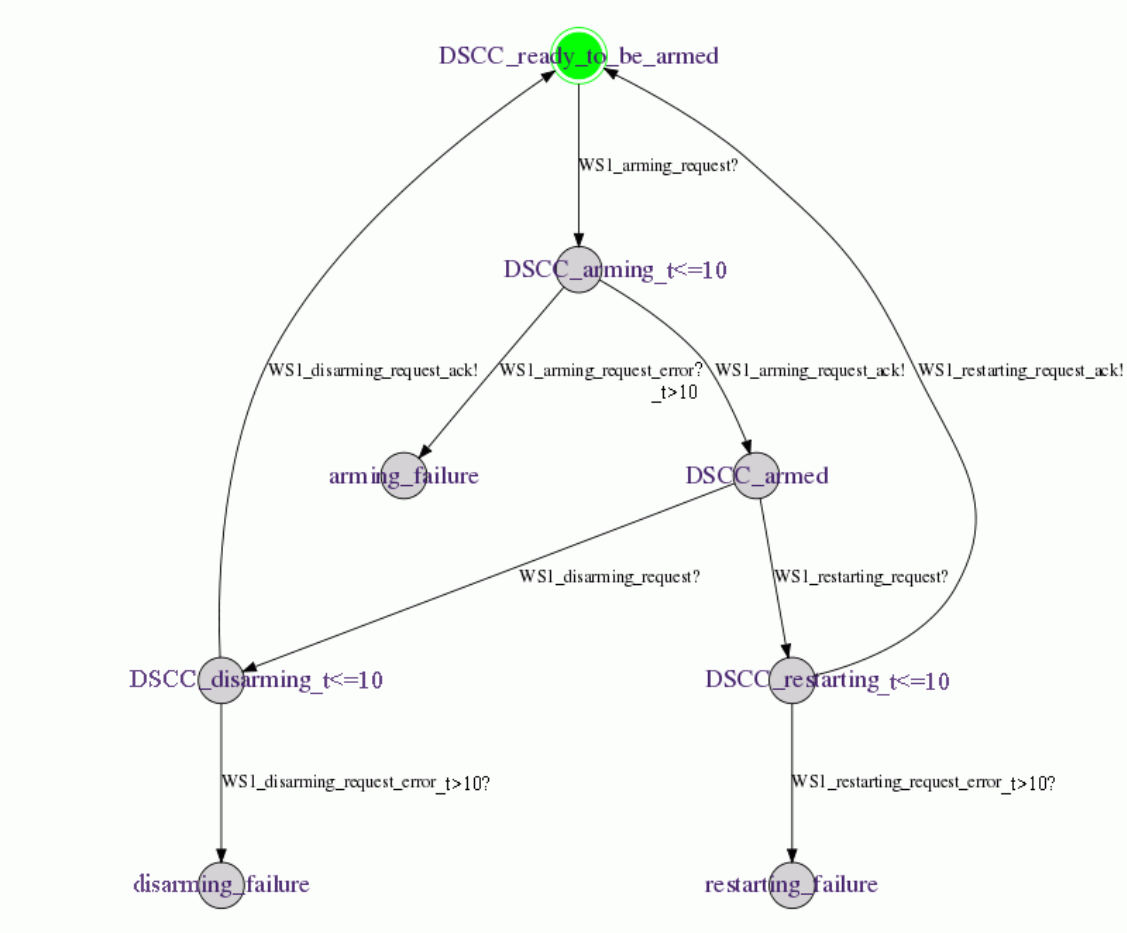


Figure 43 : Automate de SW1-demande_d'armement au niveau du DS CC.

Au contraire, si l'acquittement *WS1-arming-request-ack* est bien intercepté, l'automate atteint un état d'attente, où il reste :

- jusqu'à la réception d'un message *WS1-disarming-request* du TS CC qui déclenchera le désarmement par le DSO des DS SS, puis le retour à l'état initial après réception de l'acquittement *WS1-disarming-request-ack*. Si ce message n'est pas reçu dans les temps prévus, un traitement d'exception est déclenché.

- ou la réception d'un message de réinitialisation *WS1-restarting-request* provenant du TS CC (après la fin d'une urgence qui aura provoqué un délestage), qui demandera au DSO de redémarrer les DS SS qui ont déclenché le délestage. Une fois que ces DS SS auront redémarré (interception de l'acquittement *WS1-restarting-request_ack*), l'automate revient dans son état initial. Si ce message n'est pas reçu dans les temps prévus, un traitement d'exception est déclenché.

4.4.3 Contrôle d'accès et vérification pour SW2-ordre_d'armement

Le service Web SW2-ordre_d'armement gère les interactions entre le DS CC et chacune des DS SS qu'il contrôle. Ces interactions correspondent à l'ordre d'armement par le DSO (donc l'organisation DS CC) des sous-stations DS SS qu'il a choisies, ainsi qu'à l'ordre de désarmement (si la pré-urgence disparaît, ou si le DSO choisit d'armer d'autres sous-stations), et à l'ordre de redémarrage des DS SS qui ont délesté (une fois que l'urgence a disparu). Comme le fonctionnement est similaire à celui de SW1, nous ne détaillerons ici que certaines particularités de ce service Web.

4.4.3.1 Automate de SW2-ordre_d'armement du côté DS CC

Pour SW2, il y a autant de contrats que de sous-stations DS SS contrôlées par le DS CC.

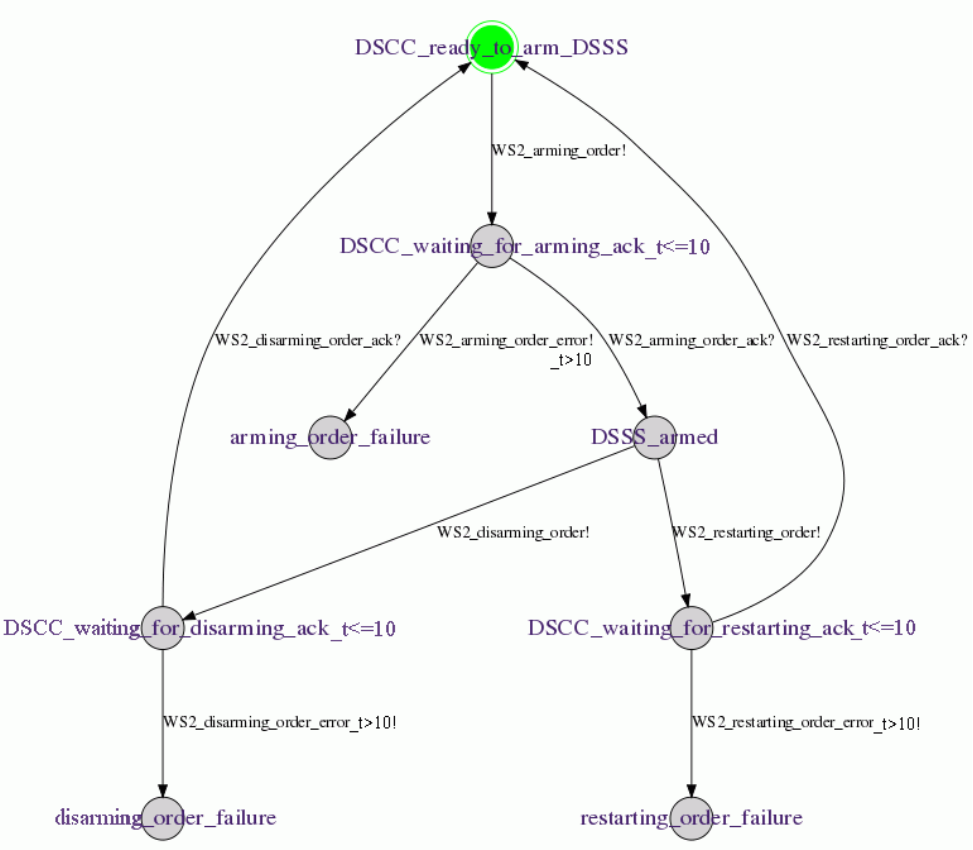


Figure 44 : Automate pour SW2-ordre_d'armement du côté du DS CC.

Dans le CIS du DS CC, il a donc autant d'automates, chacun correspondant à une seule DS SS. Lorsque le DSO arme une sous-station DS SS, il envoie un message *WS2-arming-order* à cette sous-station. Une minuterie est alors initialisée et l'automate atteint un état où il attend l'acquittement de l'armement (*WS2-arming-order-ack*) en provenance de la DS SS armée. Si le délai a expiré sans avoir reçu l'acquittement, un message *WS2-arming-order-error* est envoyé au DS CC pour déclencher un traitement d'exception. Inversement, dans les situations normales, lorsque le CIS intercepte le message *WS2-arming-order-ack*, l'automate se met dans un état d'attente (de désarmement ou de redémarrage), et l'information est transmise au DS CC, qui, lorsque suffisamment de DS SS ont été armées, envoie le message *WS1-arming-request-ack* au TS

CC. Lorsque le DSO décide de désarmer la DS SS, il envoie un message *WS2-disarming-request* à cette DS SS à désarmer et l'automate passe dans un état où il attend le message *WS2-disarming-request-ack* de la DS SS. Si le délai expire sans avoir reçu cet acquittement, un message *WS2_disarming_request_error* est envoyé au DS CC pour déclencher un traitement d'exception. Le redémarrage de la DS SS se fait de la même façon.

4.4.3.2 Règle OrBAC de l'armement dans la politique du DS SS

Lorsque le DSO invoque le service Web SW2-ordre_d'armement sur une sous-station, le processus correspondant est lancé pour le compte de l'utilisateur virtuel *uv-SW2-DS*, et ce processus exécute l'action *armement* sur l'objet *MCDTU*. Cette action est autorisée par la séquence OrBAC décrite dans le Tableau 9.

Permission(DS SS, RôleDSO, préparer, SS_Circuits, pré-urgence) ∧ Habilite(DS SS, uv-SW2-DS, RôleDSO) ∧ Considère(DS SS, armement, préparer) ∧ Utilise(DS SS, MCDTU, SS_Circuits) ∧ Définit(DS SS, uv-SW2-DS, armement, MCDTU, pré-urgence) ⇨ Est_permis(uv-SW2-DS, armement, MCDTU)

Tableau 9 : Représentation de la règle OrBAC pour SW2 du côté DS SS.

4.4.3.3 Automate de SW2-ordre_d'armement du côté du DS SS

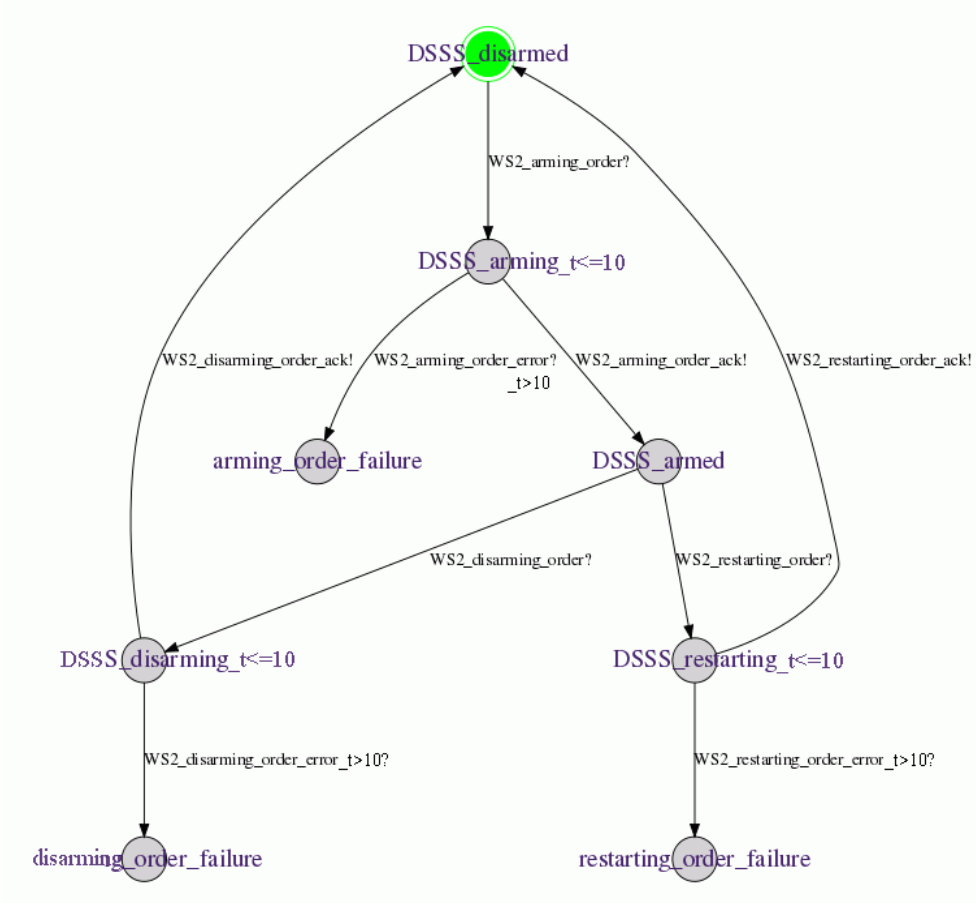


Figure 45 : Automate pour WS2-arming-order au niveau du DS SS.

Du côté de la sous station DS SS, l'automate dans son état initial attend le message *WS2-arming-order* (voir Figure 45).Lorsque l'armement de la DS SS est effectué, un acquittement *WS2-arming-order-ack* est envoyé au DS CC, le DS SS est à présent armé. Ensuite, si la DS SS reçoit le message *WS2-disarming-order*, elle effectue le désarmement et renvoie l'acquittement *WS2-disarming-order-ack* vers le DS CC et l'automate revient dans son état initial. Le fonctionnement est similaire pour le redémarrage.

La Figure 46 récapitule toutes les étapes de l'application du modèle PolyOrBAC au scénario de délestage. Par souci de simplification, nous n'avons pas montré toutes règles OrBAC, et tous les automates représentant les contrats au niveau des différentes sous stations.

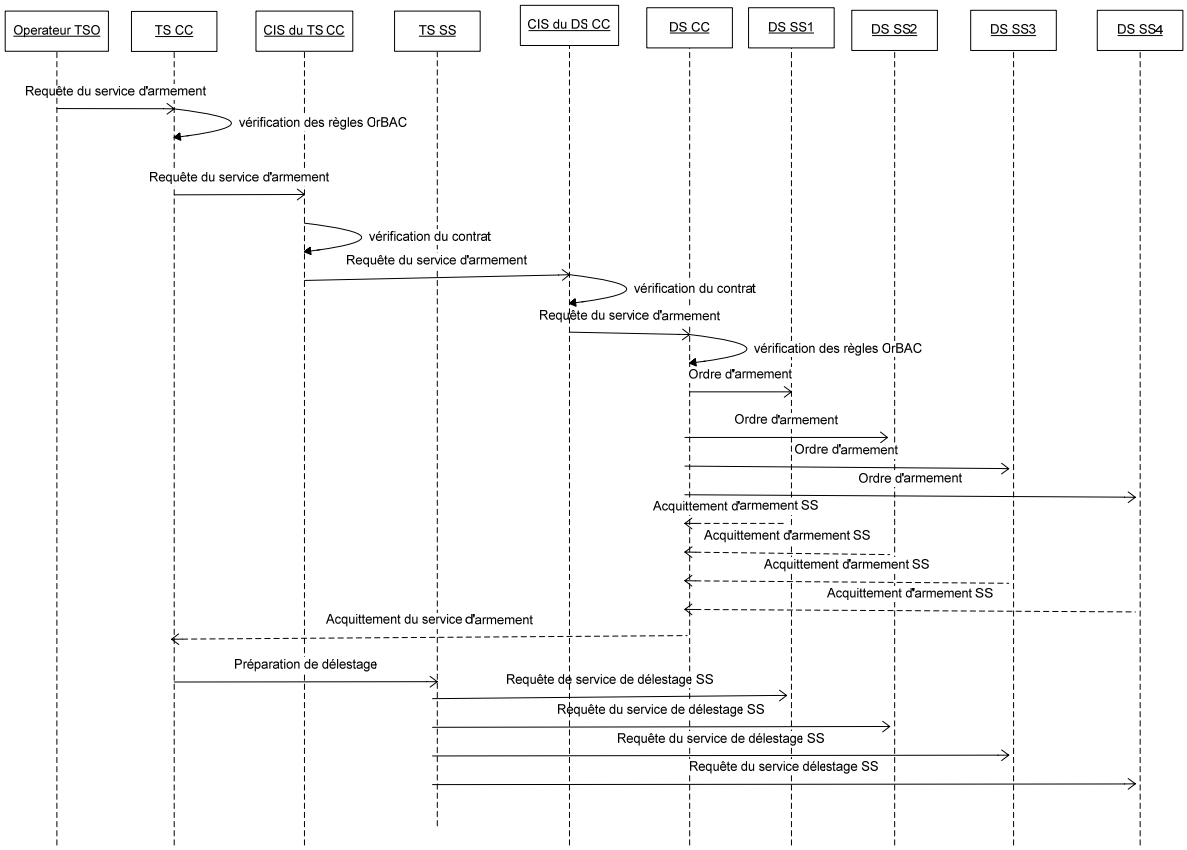


Figure 46 : Application du modèle PolyOrBAC au scénario de délestage.

4.5 Conclusions du chapitre 4

Dans ce chapitre, nous avons appliqué PolyOrBAC à une étude de cas réel d'infrastructure d'information critique, celle des infrastructures de production, de transport, et de distribution d'énergie électrique. Pour cela, nous avons dans un premier temps décrit le fonctionnement de ce type d'infrastructure et nous nous somme focalisé sur un scénario réel afin de mettre en évidence le fonctionnement de ces infrastructures en tenant compte des différentes acteurs et composants communiquant en leur sein et nous avons détaillé le fonctionnement d'un scénario qui pourrait être la cible d'une attaque pour provoquer un black-out. Dans un deuxième temps, nous avons mis en évidence l'applicabilité du schéma-cadre PolyOrBAC à ce scénario, en précisant les règles correspondantes dans les politiques OrBAC de chaque organisation collaborant au sein de l'infrastructure d'information critique et les automates temporisés qui vérifient les interactions entre ces mêmes organisations.

Chapitre 5. Mise en œuvre et implémentation du schéma-cadre PolyOrBAC

Dans le chapitre précédent, nous avons appliqué PolyOrBAC à une étude de cas réel, plus précisément à une infrastructure de production, de transport, et de distribution d'énergie électrique. Nous nous sommes particulièrement intéressés à la spécification de politiques de sécurité pour chaque organisation collaborant au sein de l'infrastructure d'information critique, mais aussi aux collaborations entre ces mêmes organisations.

Le but de ce chapitre est de montrer la possibilité de mise en œuvre de PolyOrBAC au sein d'un environnement physique réel, tout en tenant compte des différentes contraintes des systèmes et réseaux, puis en démontrant comment l'implémentation de chaque composant (collaboration grâce à la technologie des services Web, contrôle d'accès grâce au modèle OrBAC, et vérification des interactions grâce aux politiques-contrats et à leur représentation par automates temporisés).

L'application présentée dans ce chapitre a été développée par : Guillaume Canneaux, Benjamin Durand, Pierre Fersing, Nicolas de Roll Montpellier and Géraldine Van Saene dans le cadre de leur projet long à l'ENSEIHT [Canneaux et al., 2008a], [Canneaux et al., 2008b], le développement a été complété et finalisé par Adrien Nowak, stagiaire de l'INSAT au LAAS-CNRS, dans le groupe TSF [Nowak, 2009a], [Nowak, 2009b]. En outre, une grande partie de ce chapitre a été rédigée en s'aidant des rapports et manuels étudiants [Canneaux et al., 2008a], [Canneaux et al., 2008b], [Nowak, 2009a], [Nowak, 2009b].

5.1 Objectif de la mise en œuvre

Afin de mettre en œuvre notre schéma-cadre, et de prendre en compte les différents besoins spécifiés dans le chapitre 3, notre implémentation doit prendre en compte trois composants principaux : les mécanismes de collaboration grâce à la technologie des services Web, les mécanismes de contrôle d'accès locaux grâce au modèle OrBAC, et enfin la vérification des interactions grâce à l'utilisation des automates temporisés.

La partie implémentation passe par plusieurs étapes. Premièrement, nous devons définir les différents besoins et contraintes du prototype à implémenter. En second, nous devons déterminer le scénario réel que nous voulons mettre en évidence, et finalement nous devons décomposer l'implémentation en composants à développer puis à regrouper afin de fournir un prototype complet. Il faut donc prévoir, dans un premier temps, des moyens pour implémenter chaque composant ou couche applicative, et dans un deuxième temps, prévoir et spécifier l'environnement d'expérimentation au sein duquel nous allons montrer la mise en œuvre de PolyOrBAC en respectant les différentes contraintes et conditions qui apparaissent au sein de cet environnement.

5.1.1 Présentation du scénario de l'expérimentation

Le scénario que nous avons choisi d'implémenter est celui qui a servi d'exemple au chapitre précédent. Afin de maintenir le bon fonctionnement du réseau électrique en général et des réseaux de transport et de distribution en particulier, il faut être en mesure de déconnecter des sous-stations de distribution si le besoin se manifeste (consommation trop élevée par exemple), puis de les remettre en circuit une fois l'urgence passée. Lorsque l'opérateur TSO du système de transport prévoit une situation d'urgence, il envoie une demande de préparation au délestage d'une quantité Q de mégawatts à l'opérateur DSO du système de distribution. Ce dernier va devoir armer suffisamment de sous-stations pour répondre à la demande du TSO. Par la suite, si l'urgence survient, un délestage est lancé automatiquement, ce qui correspond à la déconnexion des sous-stations préalablement armées par le DSO. Le TSO peut également envoyer une demande de désarmement si l'urgence est passée, dans ce cas le DSO va devoir désarmer toutes les sous-stations armées. Afin de pouvoir contrôler tout cela, il est nécessaire que le TSO et le DSO soient informés de l'état du réseau électrique, de manière globale au niveau du réseau de transport, et de manière séparée pour chaque sous-station du réseau de distribution.

Dans notre implémentation, nous nous sommes intéressés aux cinq services Web suivants :

- SW1-demande_d'armement : le TS CC demande au DS CC d'armer une certaine quantité de MW,
- SW2-ordre_d'armement : le DS CC arme un certain nombre de sous stations afin de répondre à la demande du TS CC,
- SW3-préparation_de_délestage : le TS CC prévient la TS SS pour la préparer à faire un délestage éventuel.

- SW4- activation_de_délestage : si le besoin se manifeste, la TS SS envoie une demande de délestage aux différentes sous stations de distribution préalablement armées,
- SW5-réintégration : les différentes sous stations de distribution sont réintégrées automatiquement après leur délestage (service interne à la DS SS).

Ces cinq services Web sont déployés sur les centres de contrôle du transport TS CC et de la distribution DS CC, sur la sous-station TS SS du système de transport et sur les sous-stations DS SS du système de distribution. Le TS CC en voie des demandes à l'opérateur DSO du DS CC, qui lui-même invoque des services Web déployés sur les sous-stations DS SS. Le service Web *SW4- activation_de_délestage* est déployé par les DS SS et est invoqué par la sous-station TS SS.

Rappelons par ailleurs que chaque communication en provenance ou à destination d'une organisation passe par son CIS, et c'est dans le CIS que sont vérifiées les interactions par messages de services Web.

Les différents appels des services Web[15] du scénario qui nous intéressent sont comme suit :

- ✓ Le TSO invoque le service *SW1-demande_d'armement*, le message correspondant à cette invocation passe par le CIS du TS CC.
- ✓ Le CIS du TS CC transmet ce message au CIS du DS CC.
- ✓ Le CIS du DS CC transmet ce message au système d'information du DS CC, ce qui déclenche l'exécution d'un processus qui exécute le service Web SW1, c'est-à-dire l'affichage d'un message urgent sur la console du DSO.
- ✓ Le DSO invoque alors les services *Web SW2-ordre_d'armement* sur les DS SS qu'il a choisi d'armer.
- ✓ Le CIS du DS CC transmet les messages correspondant aux CIS des DS SS.
- ✓ Et enfin, le CIS de chaque DS SS concernée transmet le message à sa DS SS, ce qui déclenche l'exécution de l'armement.

Dans un premier temps, le démonstrateur doit simuler les communications entre les centres de contrôle du système de transport (TS CC) et du système de distribution (DS CC) et leurs différentes sous-stations. En général, une sous-station de transport TS SS alimente plusieurs sous-stations de distribution DS SS. Dans notre implémentation, nous considérons une seule TS SS et quatre DS SS.

5.1.2 Besoins et contraintes du prototype

Pour implémenter un prototype de notre plateforme, il faut prendre en compte un certain nombre de besoins et de contraintes organisationnelles, structurelles, physiques, et autres. Nous pouvons en citer quelques unes :

[15] Par souci de simplification, nous n'avons décrit ici que les messages échangés lors des invocations des services SW1, et SW2.

- Il est nécessaire de prendre en compte un nombre significatif d'organisations afin de mettre en œuvre complètement le scénario et ainsi l'extensibilité de PolyOrBAC ; ceci induit le déploiement des composants (collaboration, contrôle d'accès, et contrats) dans ces différentes organisations.
- Pour des raisons logistiques, nous disposons d'un nombre réduit de machines physiques, il faut donc trouver les moyens de simuler un réseau complet avec peu de ressources.
- Il est nécessaire de fournir un prototype qui peut être déployé sur n'importe quel système d'exploitation, ou en tous cas, sur un certain nombre d'entre eux pour tenir compte de l'hétérogénéité des IIC ; il faut donc prévoir la portabilité de notre prototype, de ses composants logiciels, systèmes et applicatifs.
- Afin de rester cohérent par rapport aux travaux existants dans le domaine, il est important d'éviter de ré-implémenter des outils déjà existants (tels que MotOrBAC[16] ou AdOrBAC[17]), ou des concepts déjà prouvés.

5.2 Mise en place de l'expérimentation

Dans le scénario choisi, il faut prendre en compte sept organisations distinctes : le TS CC, le DS CC, la TS SS, et les quatre DS SS, ainsi que les CIS (CRUTIAL Information Switch) affectés à chaque organisation. La Figure 47 montre comment les organisations et CIS sont organisés au sein d'une IIC selon l'architecture CRUTIAL.

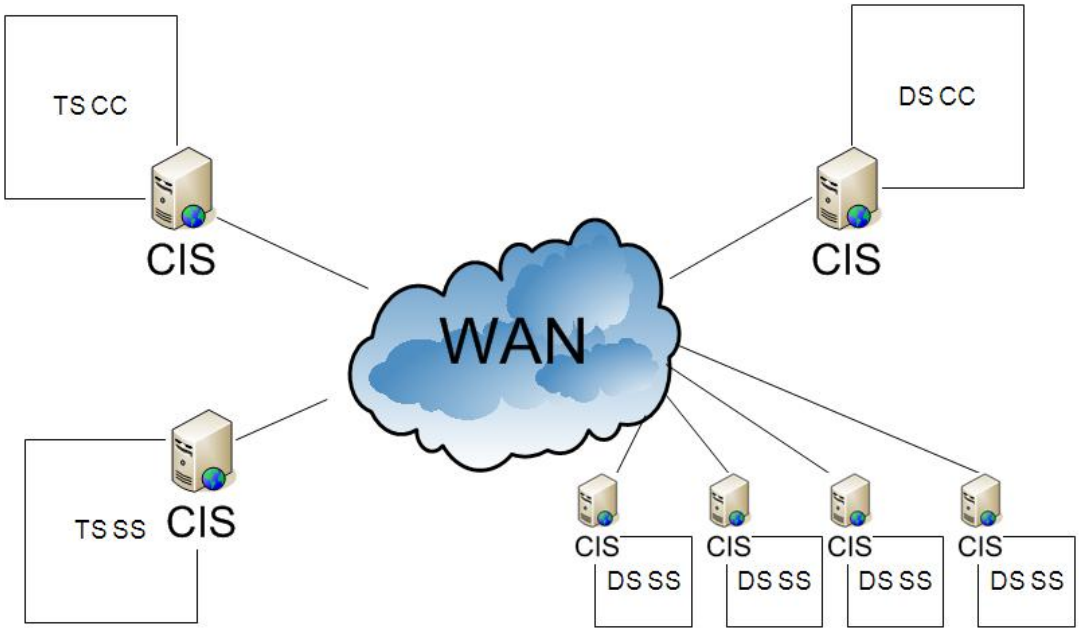


Figure 47 : Organisations et CIS au sein de l'IIC considérée.

Afin de représenter correctement la configuration de cette IIC, les sept organisations devraient être déployées sur au moins sept machines différentes, plus sept CIS utilisés comme des passerelles ou des pare-feux entre chaque organisation et les autres. Pour des raisons

[16] http://motorbac.sourceforge.net/
[17] http://www.orbac.org/index.php?page=orbac&lang=fr

logistiques, plutôt que d'utiliser autant de machines physiques distinctes que d'organisations, nous avons choisi d'émuler les différentes organisations et leurs CIS respectifs sur une ou deux machines physiques seulement.

5.2.1 Description de la plateforme d'émulation

Un moyen d'émuler ces machines, est d'utiliser un gestionnaire de machines virtuelles tel que Xen ou OpenVZ[18]. L'objectif de la virtualisation est de pouvoir exécuter plusieurs systèmes d'exploitation en même temps sur une machine physique, et de pouvoir les utiliser indépendamment. Dans notre démonstrateur, sur deux machines physiques, nous pouvons simuler un réseau de quatorze machines virtuelles : les sept systèmes d'information des sept organisations, et leurs CIS respectifs.

Dans notre prototype, nous avons choisi d'utiliser OpenVZ pour les raisons suivantes :

- C'est un outil très léger, qui nous permet de simuler les 14 machines avec 14 serveurs Apache Tomcat[19] et Axis[20] distincts dont nous avons besoin pour la mise en œuvre.
- Il a une gestion des ressources efficace et un outil d'administration pratique.
- Les performances de OpenVZ sont satisfaisantes car il n'émule pas les composants matériels.
- Enfin, il offre une virtualisation riche : chaque machine virtuelle a un nom qui lui est propre, sa propre adresse IP, ses propres utilisateurs et processus, etc.

5.2.2 Définition des composants de base liés à PolyOrBAC

Comme décrit précédemment, le démonstrateur doit mettre en œuvre à la fois les mécanismes de collaboration par services Web, les mécanismes de contrôle d'accès local correspondant aux politiques OrBAC locales, et la vérification par *model checking*[21] des interactions. Le Tableau 10 décrit les technologies et outils que nous avons choisi d'utiliser pour ces trois types de mécanismes.

	Niveau fonctionnel	Niveau local	Niveau des interactions
Technologie utilisée	Services web	OrBAC	Automates Temporisés
Outil utilisé	Axis, Tomcat, Java[22].	Java	UPPAAL[23]
Rôle	Collaboration	Contrôle d'accès local	Contrôle des interactions

Tableau 10 : Mécanismes utilisés et composants de PolyOrBAC.

La Figure 48 récapitule les différentes étapes d'exécution de notre plateforme. En premier lieu, l'utilisateur accède au panneau d'invocation des services Web. Au préalable, les

[18] http://old.openvz.org/
[19] http://tomcat.apache.org/
[20] http://ws.apache.org/axis/
[21] Outil développé par Adrien NOWAK, et Amine BAINA.
[22] http://java.sun.com/
[23] http://www.uppaal.com/

politiques-contrats ont été vérifiées de manière statique grâce à l'outil UPPAAL de modélisation et de vérification d'automates temporisés. De même les politiques locales de contrôle d'accès ont été définies grâce à l'outil OrBAC Designer[24] et elles sont vérifiées à l'exécution par le moteur de contrôle d'accès JOrBAC[25] (implémentation des mécanismes OrBAC en langage Java, qu'on détaillera dans la section 5.2.5), et enfin le fonctionnement des services Web est vérifié en temps réel grâce à un outil de model checking qui vérifie que chaque message de service Web correspond à une transition autorisée dans la politique-contrat correspondante.

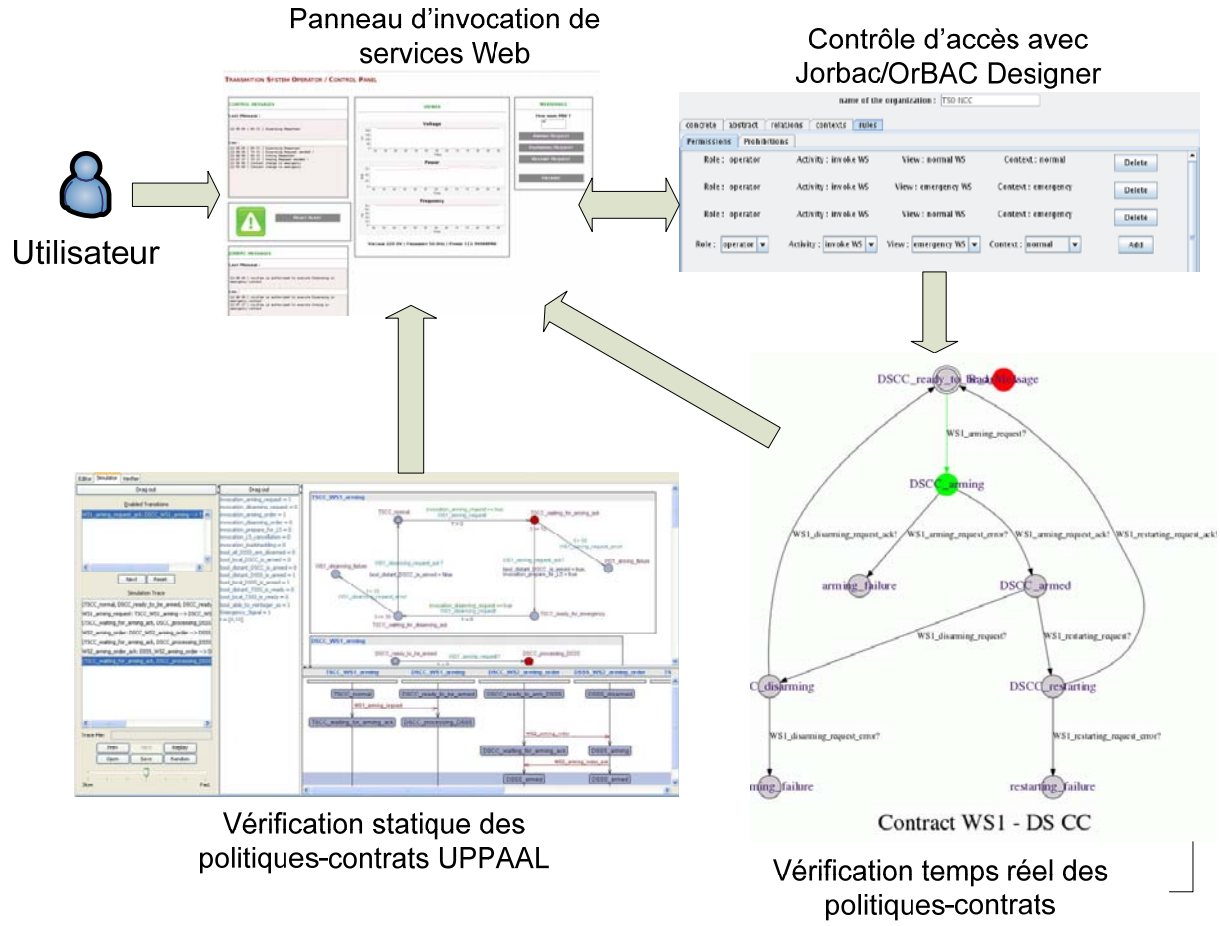


Figure 48 : Plateforme PolyOrBAC.

La section suivante présente les outils informatiques traditionnels utilisés pour les besoins de l'implémentation, tels que Java, Apache Tomcat, Apache Axis, etc.

[24] Outil développé par Guillaume Canneaux, Benjamin Durand, Pierre Fersing, Nicolas de Roll Montpellier et Géraldine Van Saene.

[25] Outil développé par Guillaume Canneaux, Benjamin Durand, Pierre Fersing, Nicolas de Roll Montpellier et Géraldine Van Saene.

5.2.3 Présentation des outils de l'expérimentation

Nous avons choisi d'utiliser Java comme langage de programmation, principalement pour des raisons de facilité d'évolution et de portabilité, ce qui facilitera la réutilisation éventuelle de notre plateforme par une autre équipe avec d'autres systèmes d'exploitation et pour d'autres applications. En outre, il est possible de créer facilement des utilisateurs (avec l'outil OrBAC Designer décrit dans la section 5.2.5) et de construire des services Web permettant aux applications de communiquer par RMI (pour *Remote Method Invocation* qu'on peut traduire par invocation de méthode à distance) en local et à distance. Les services Web et l'interface Web sont également réalisés en Java, grâce à l'utilisation de *servlets* et d'Axis. Afin de permettre une compilation rapide, et un déploiement fiable et automatique des applications Java, nous avons choisi l'outil de compilation Ant[26].

Dans notre application, nous avons choisi d'utiliser Apache Tomcat pour deux types d'applications : (1) fournir une interface Web pour les opérateurs du TS CC et du DS CC : notre interface est une application Web déployée avec Tomcat et contenant des pages JSP et des *servlets* ; (2) installer l'application Web Axis, qui fournit une plateforme de services Web basée sur XML ; l'outil Apache Tomcat offre non seulement un serveur Web simple, mais également une implémentation de *servlets* et de *JSP containers*.

Apache Axis est une plateforme de services Web basée sur XML, qui peut être installée sur le serveur Apache Tomcat. Axis permet de créer des services Web en Java, puis de les déployer sur différentes machines. Axis fournit également un moyen pour générer des fichiers WSDL de description des services Web. Enfin des classes Axis sont disponibles pour générer des clients de service Web à partir d'un fichier WSDL. La section suivante présente la façon dont nous avons implémenté les mécanismes de collaboration en se basant sur la technologie des services Web.

5.2.4 Implémentation des mécanismes de collaboration par services Web

Dans notre plateforme, les services Web sont utilisés pour mettre en relation les organisations collaborant pour partager ou échanger des données. Les interactions entre organisations sont limitées à des services Web où une organisation peut exécuter des services Web fournis par une autre. Pour cela, nous installons un serveur Apache Tomcat et Apache Axis sur toutes les machines des organisations où les services Web sont déployés : TS CC, TS CC CIS, TS SS, TS SS CIS, DS CC, DS CC CIS, et les différentes DS SS et DS SS CIS.

Pour les besoins de l'application, nous avons mis en œuvre des panneaux de contrôle affichés par des navigateurs Web, chaque panneau permet de lancer différents services Web qui accèdent à des données ou effectuent diverses opérations (pour récupérer des mesures, déconnecter une sous-station, etc.). Notre interface se compose de deux pages Web, correspondant au panneau de contrôle du TSO pour l'une, et au panneau de contrôle du DSO pour l'autre. La Figure 49 décrit le panneau de contrôle du TSO.

[26] http://ant.apache.org/

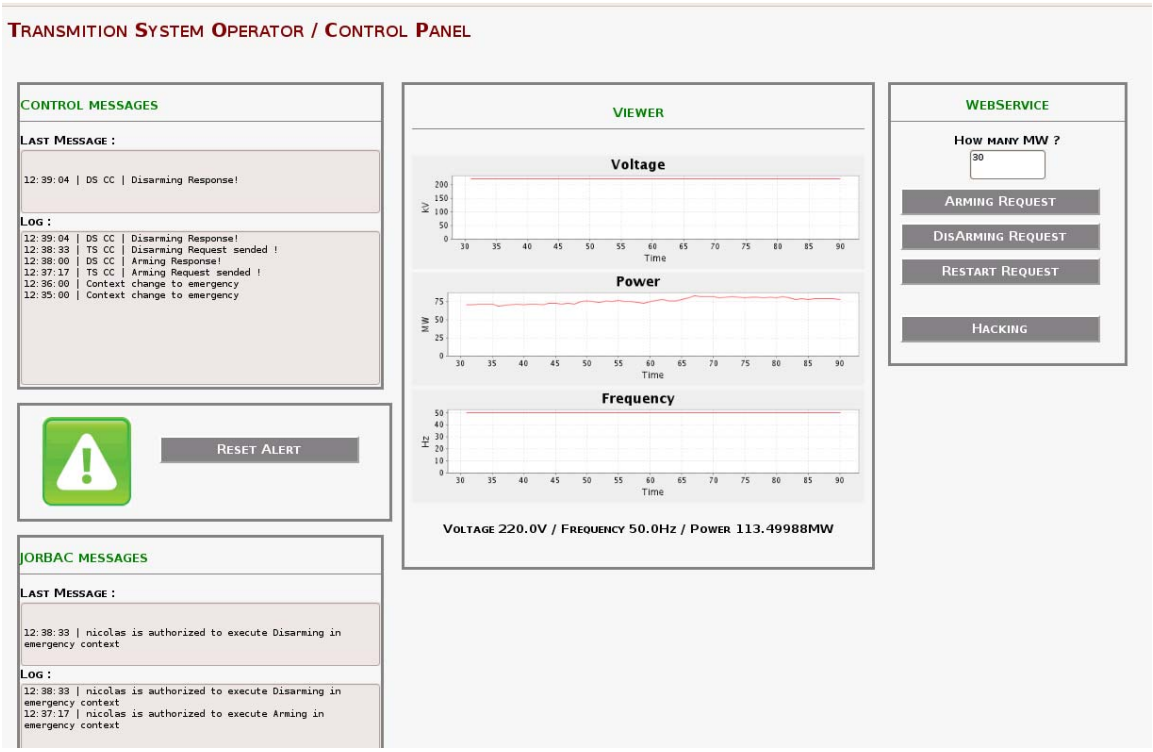


Figure 49 : Panneau de contrôle du TSO.

Le panneau du TSO fournit un journal (*log*) des différents messages reçus et envoyés par le TSO, un contrôleur JOrBAC des actions effectuées par le TSO, un ensemble de services Web qu'il peut invoquer (demande d'armement, demande de désarmement, demande de redémarrage). Des intrusions peuvent être simulées grâce à un bouton *hacking*, qui permet de modifier l'invocation des services Web pour que ceux-ci s'effectuent sans aucun contrôle jusqu'à ce que le mode normal soit réactivé.

Enfin le panneau contient un *visualiseur* sur lequel nous pouvons afficher les courbes de fréquence, tension et puissance délivrée au système de distribution. Pour simuler le fonctionnement d'une sous-station, nous avons aussi créé un outil qui génère des valeurs (fréquence, tension, puissance) représentatives du réseau électrique simulé. La Figure 50 décrit le panneau de contrôle du DSO.

Ce panneau dispose également d'un journal des différents messages reçus et envoyés par le DSO, d'un contrôleur JOrBAC des actions effectuées par le DSO, et d'un ensemble de services Web qu'il peut invoquer, tout particulièrement pour effectuer des actions au niveau des différentes sous-stations de distribution. La section suivante présente la manière dont nous avons implémenté une interface pour la gestion des contrôles d'accès d'une politique locale décrite en OrBAC (à l'aide d'OrBAC Designer).

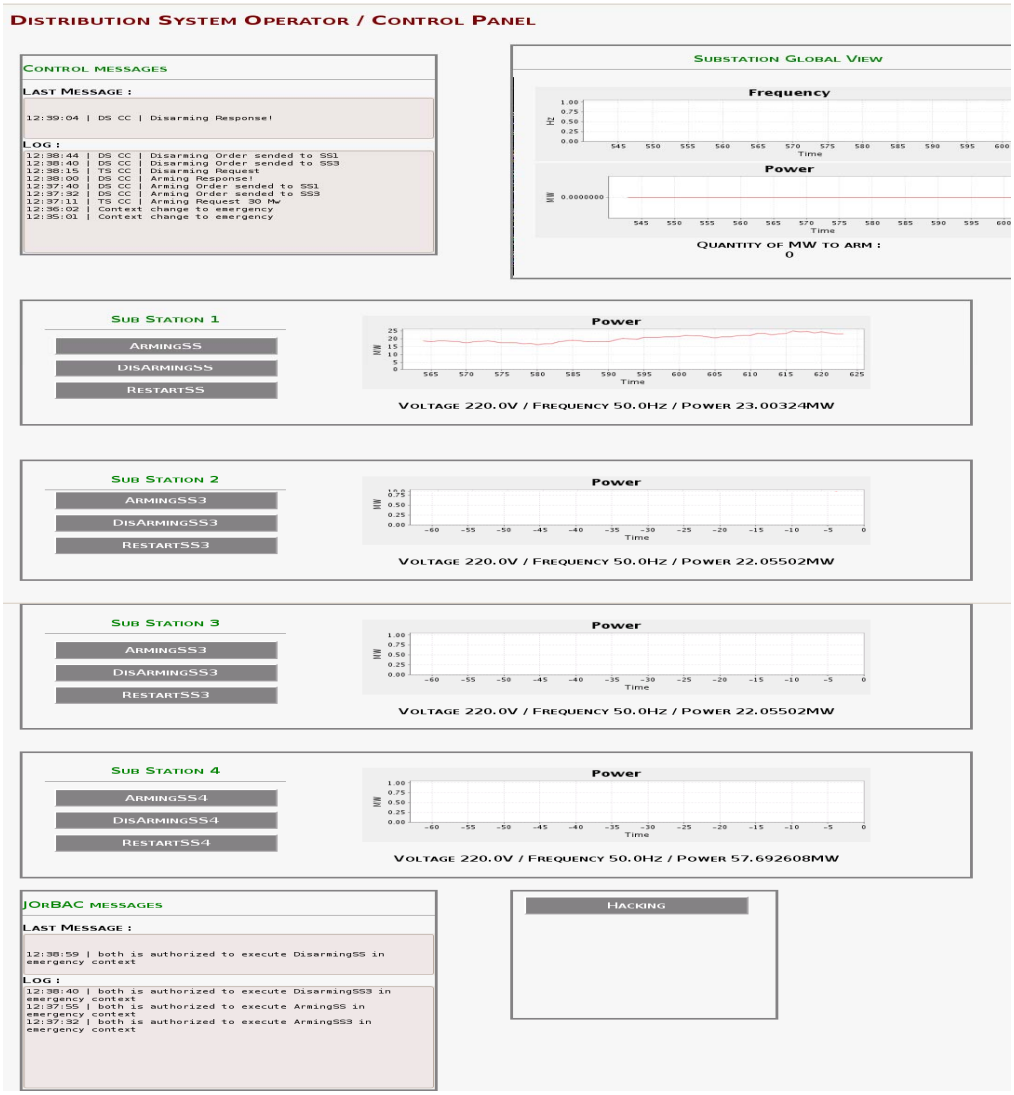


Figure 50 : Panneau de contrôle du DSO.

5.2.5 Implémentation des mécanismes de contrôle d'accès OrBAC

Dans cette partie, nous décrivons la mise en œuvre des différents contrôles d'accès utilisés, du contrôle des utilisateurs jusqu'au contrôle des actions effectuées. Dans le cadre de PolyOrBAC, le modèle OrBAC est utilisé pour la spécification et l'application d'une politique de contrôle d'accès locale au niveau chaque organisation, sachant que ces politiques de contrôle d'accès locales prennent en compte également les invocations de services Web distants, et d'un autre côté les accès par d'autres organisations aux services Web locaux. Chaque organisation est responsable de l'identification et de l'authentification de ses propres utilisateurs et de leurs actions, y compris leurs invocations de services Web distants.

Pour aider les administrateurs à définir et gérer la politique de sécurité de leur organisation, un outil graphique a été développé : OrBAC Designer. Cet outil permet à chacun de gérer une politique par la création de sujets, rôles, actions, activités, objets, vues et contextes. OrBAC Designer est une implémentation Java des mécanismes de contrôle d'accès basés sur le modèle OrBAC. Le contrôle d'accès s'effectue en utilisant des fichiers XML paramétrables représentant

les règles OrBAC. Pour les deux organisations principales distinctes, le réseau de transport et le réseau de distribution, il y a deux fichiers de règles d'accès : *ds_cc_policy.xml* et *ts_cc_policy.xml* correspondant respectivement aux organisations DS CC et TS CC. À chaque exécution d'une action, il est possible de vérifier que tel utilisateur a le droit d'exécuter telle action dans tel contexte.

L'administrateur a la possibilité de modifier les relations définies au niveau concret (sujets, actions, objets) et au niveau abstrait (rôles, activités, vues). Les deux figures suivantes (Figure 51, Figure 52) mettent en évidence les deux niveaux de définition de la politique de sécurité.

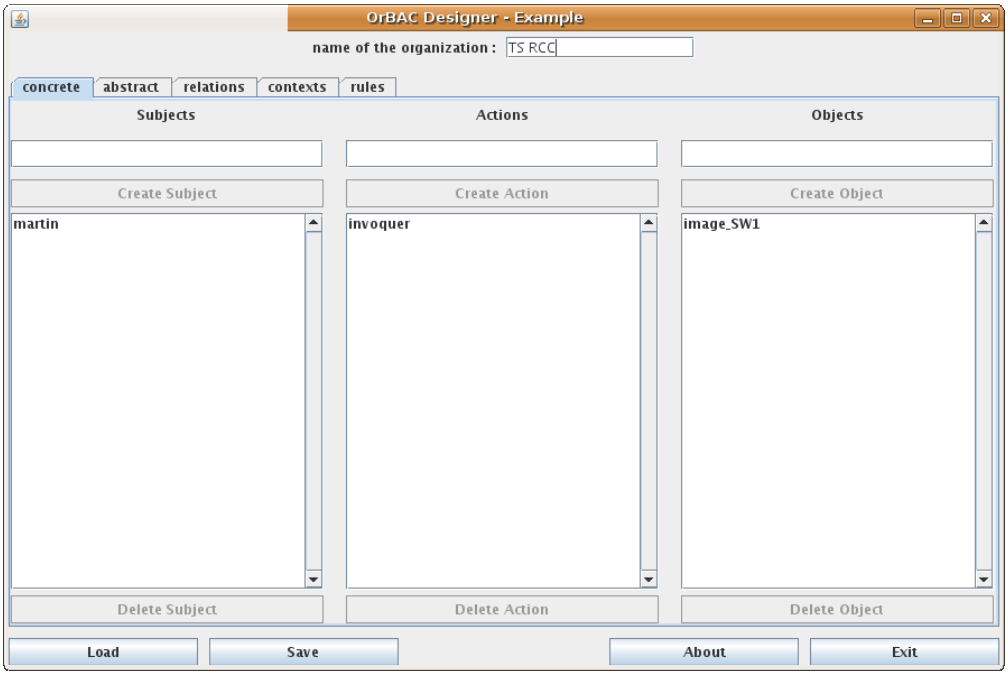


Figure 51 : Outil OrBAC Designer : niveau concret.

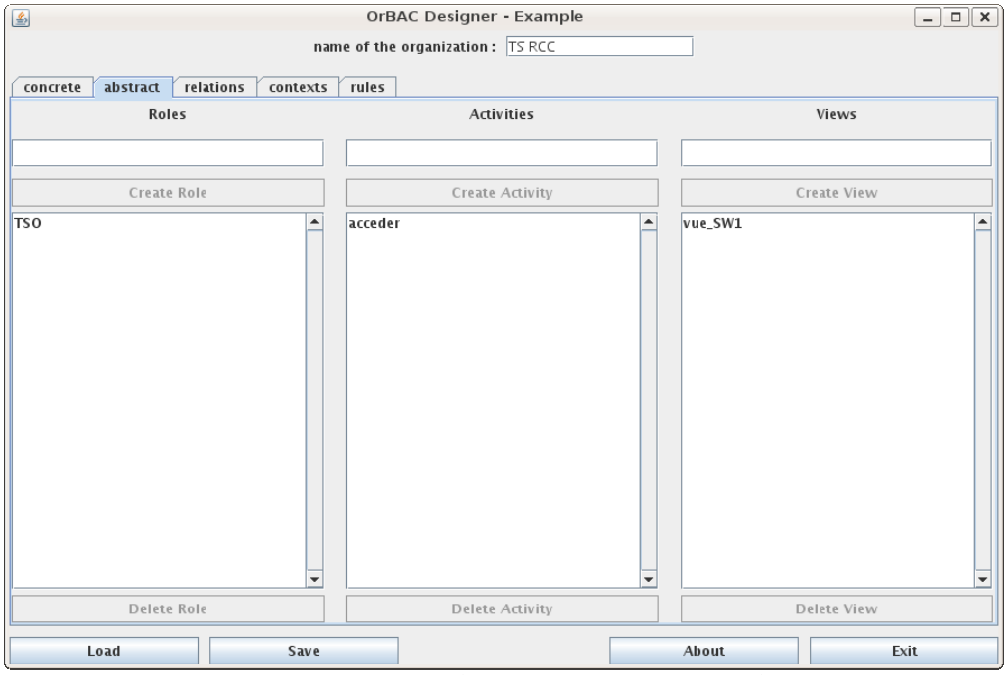


Figure 52 : Outil OrBAC Designer : niveau abstrait

Le but est de pouvoir spécifier des règles de contrôle d'accès prenant en compte les différentes entités abstraites et concrètes, y compris la notion de contexte qui conditionne l'application des règles. La Figure 53 illustre la spécification d'un certain nombre de règles de contrôle d'accès positives (permissions), parmi lesquelles nous trouvons celle donnant au rôle « *TSO* » le droit d'effectuer une activité « *accéder* » sur la vue « vue_SW1 » en contexte « pre-urgence ».

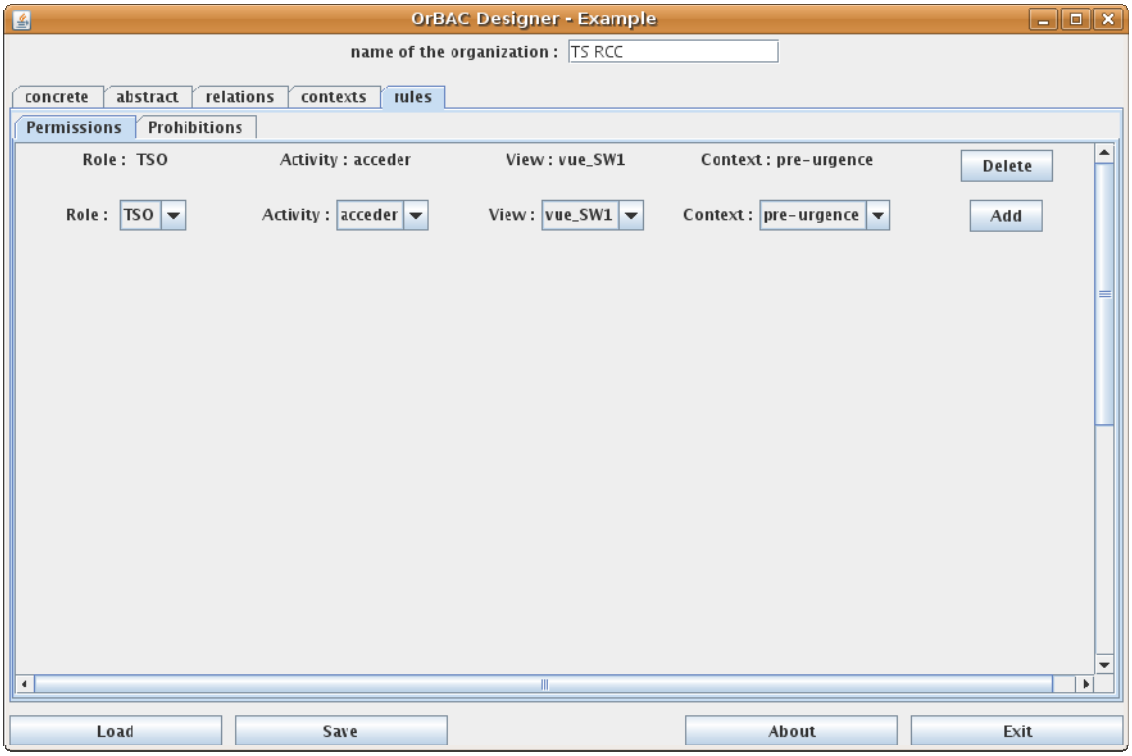


Figure 53 : Spécification d'une règle de contrôle d'accès avec l'outil OrBAC Designer.

Au moment de l'exécution, toutes les demandes d'action sur un objet sont examinées par un moteur de contrôle d'accès JOrBAC qui vérifie que cette demande correspond à une règle de la politique qui a été définie par l'outil OrBAC Designer. Si la règle existe dans la politique, l'accès est autorisé ; sinon l'accès est refusé, et la tentative d'accès est enregistrée dans un fichier de journalisation.

Si l'action sur l'objet est en fait une invocation d'un service Web distant par un utilisateur local, la première vérification est faite par le moteur JOrBAC de l'organisation cliente, puis si cette première vérification est positive, le CIS du client vérifie que l'appel est bien autorisé par la politique-contrat correspondante, puis le CIS du prestataire effectue la même vérification du côté du prestataire, et enfin le moteur JOrBAC du prestataire vérifie que les actions du processus exécutant le service Web sont valides dans la politique OrBAC du prestataire. La section suivante présente plus en détail les vérifications des politiques_contrats par les CIS.

5.2.6 Implémentation des mécanismes de vérification des interactions par contrats

Nous avons montré précédemment l'intérêt de l'utilisation des politiques-contrats pour vérifier les échanges de messages de services Web du côté du client et du côté du prestataire. La Figure 54 représente une capture d'écran de l'outil UPPAAL, qui nous sert à vérifier de façon statique et préalable les propriétés des automates temporisés qui décrivent les politiques-contrats préétablies pour l'exécution des services négociés.

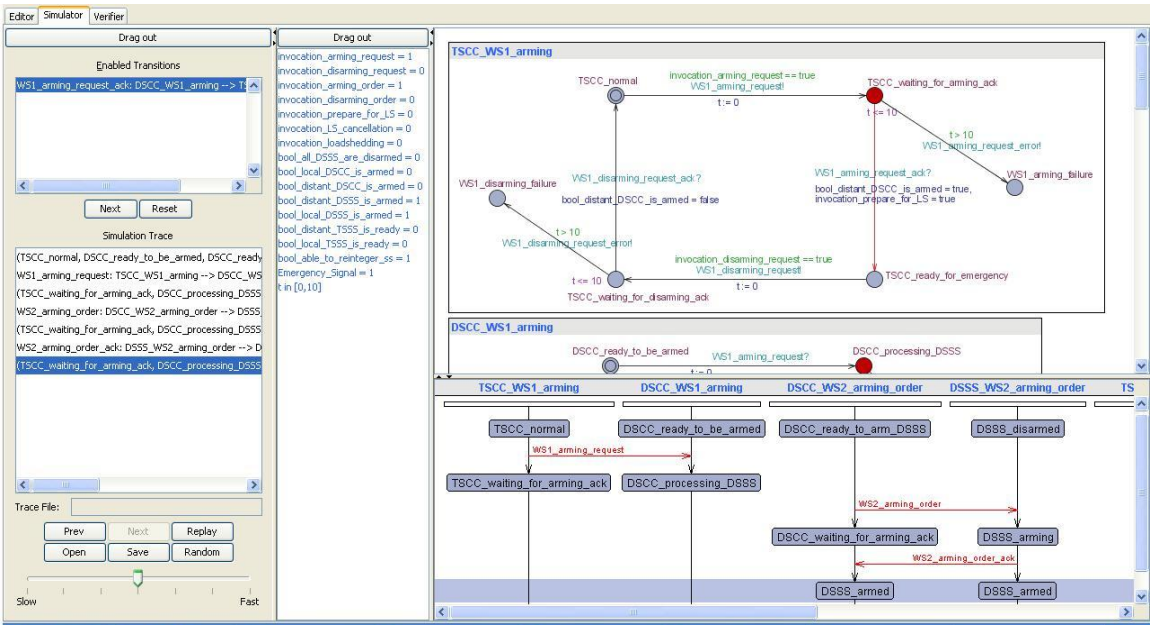


Figure 54 : Vérification statique des politiques-contrats grâce aux automates temporisés.

Cette figure démontre la possibilité qu'offre UPPAAL, dans la vérification de la cohérence (fenêtre en haut à droite, voir également Figure 55) de plusieurs automates temporisés (exprimant les politiques-contrats), et donc le bon déroulement du scénario incluant les différentes entités (fenêtre en bas à droite), et permet de suivre pas à pas les différents messages échangés (fenêtres de gauche). La fenêtre du milieu détaille la vérification des différentes conditions (booléennes) qui contrôlent le passage de l'automate par une transition spécifique.

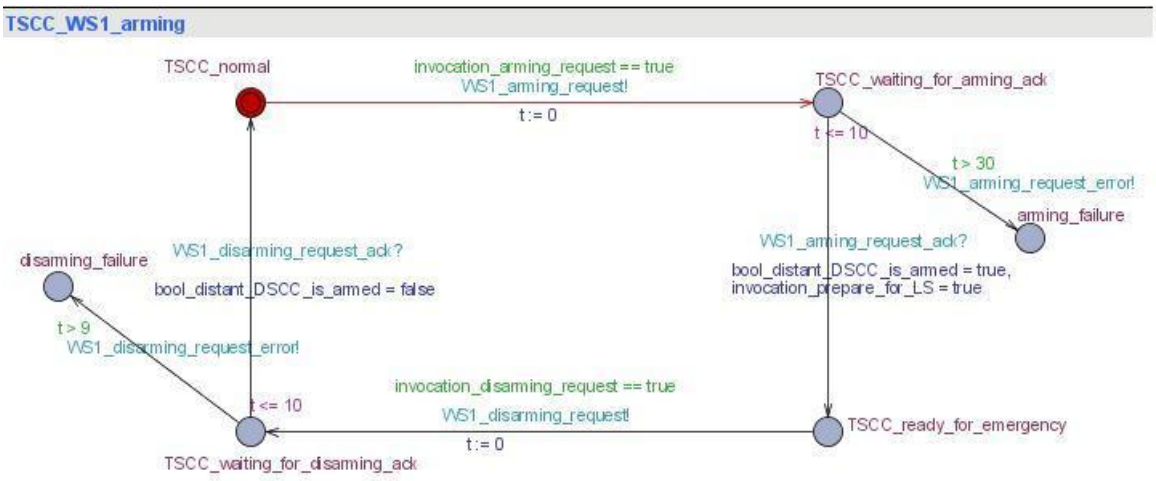


Figure 55 : Exemple d'automate temporisé pour la représentation de politiques-contrats.

Pour représenter une politique-contrat avec la richesse des modalités principales d'OrBAC (permissions, interdictions, obligations), il faut représenter ces modalités dans les automates temporisés dont un exemple est donné dans la Figure 55.

Un autre outil a été développé par notre équipe pour vérifier de façon dynamique et en temps réel le bon fonctionnement et déroulement des politiques-contrats. Nous rappelons ici l'utilisation des contrats. Tout d'abord, les organisations négocient l'utilisation de leurs services Web afin d'arriver à un accord. Cet accord est représenté par un contrat que les deux organisations sauvegardent sous forme d'automates temporisés. Lorsqu'un utilisateur va tenter d'exécuter un service, le contrat, et donc l'automate correspondant, est vérifié, si le contrat est respecté l'exécution suit son cours, sinon une exception est levée à l'exécution, puis loguée comme preuve de non-respect de la transaction.

5.3 Exécution du scénario en fonctionnement normal

Dans cette section, nous allons détailler le fonctionnement de notre plateforme lors de l'exécution d'un scénario représentatif (armement, délestage, et redémarrage). Ce scénario nécessite d'être dans un contexte de pré-urgence, c'est-à-dire un état où l'opérateur TSO du système de transport identifie un risque de surconsommation ou de sous-production. Dans ce contexte, le TSO décide de demander au système de distribution de se préparer pour une réduction de puissance d'un certain nombre de mégawatts, sachant que cette valeur doit être inférieure à la puissance globale des sous-stations que le DSO peut armer. Ceci correspond à une demande d'armement envoyée par le TSO au DSO.

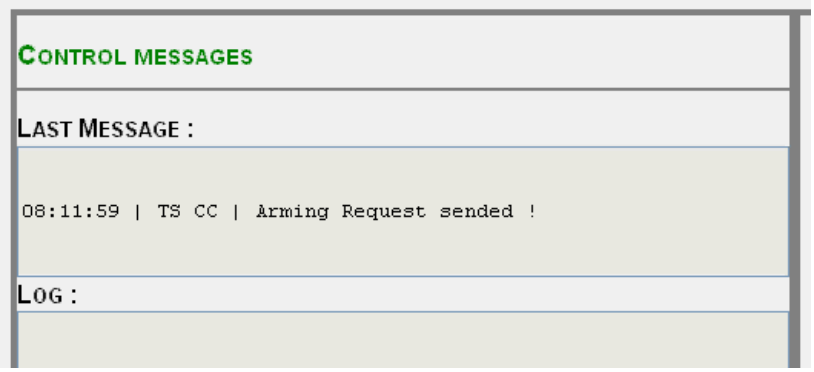


Figure 56 : Messages de contrôle du TSO.

La requête est tout d'abord analysée par le moteur de contrôle d'accès JOrBAC du TS CC, qui vérifie en particulier le contexte de pré-urgence. La Figure 57 montre un exemple de contrôle d'accès JOrBAC.

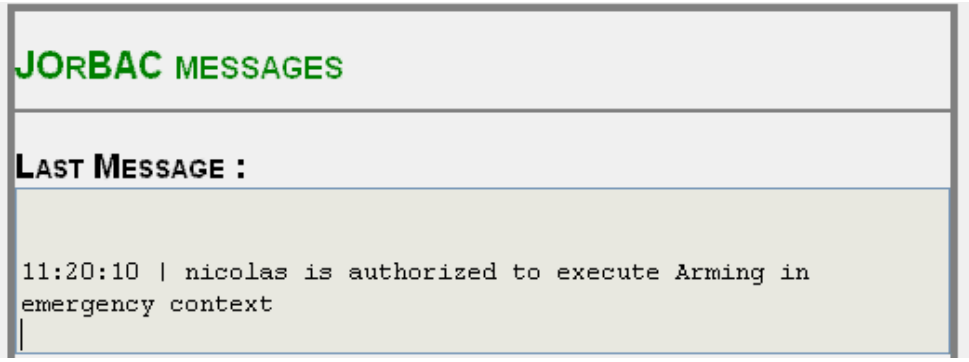


Figure 57 : Messages JorBAC du TSO.

Si JOrBAC autorise l'invocation du service Web SW1-demande_d'armement, le message d'invocation est d'abord analysé par le CIS du TS CC puis par celui du DS CC, en fonction des deux automates temporisés de la politique-contrat correspondante (celui du client TS CC et celui du prestataire DS CC). Comme ils sont conformes à la politique_contrat, le message est bien transmis au DS CC et déclenche l'affichage de la requête d'armement sur le panneau de contrôle du DS CC. La Figure 58 montre que la requête d'armement (armer 30 MW) envoyée par le TS CC est bien reçue par le DS CC.

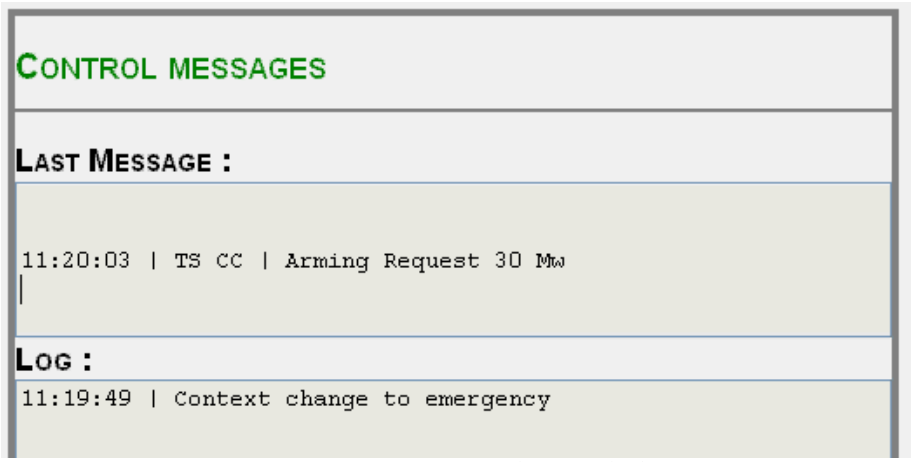


Figure 58 : Messages de contrôle du DSO.

Une fois la requête reçue par le DSO, celui-ci choisit les sous-stations à armer. La Figure 59 montre la requête d'armement de la sous-station SS2 par appel au service Web SW2-ordre_d'armement.

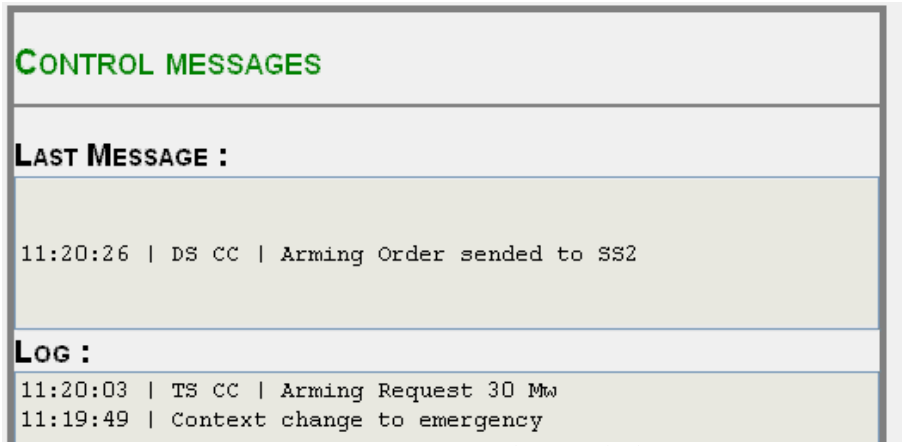


Figure 59 : Messages de contrôle du DSO.

Le moteur de contrôle d'accès JOrBAC du DS CC vérifie si l'utilisateur (ayant le rôle DSO) qui a invoqué ce service a le droit d'exécuter cette action sur l'objet image_de_SW2 dans le contexte actuel.

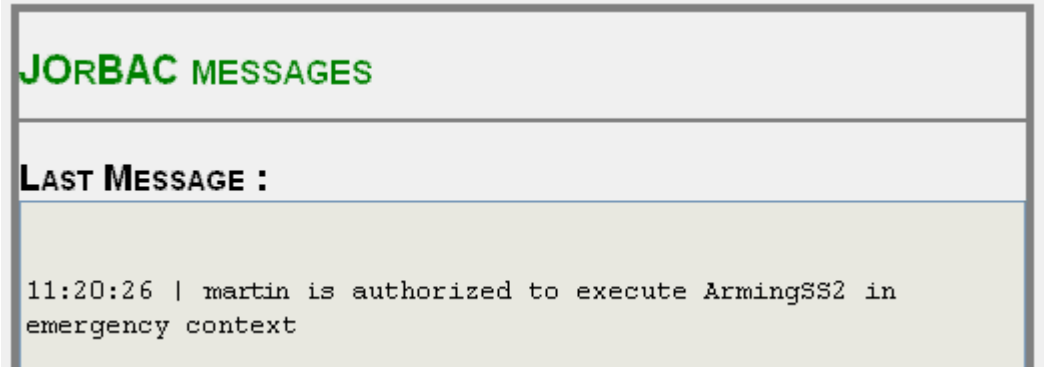


Figure 60 : Messages JOrBAC du DS CC.

La Figure 61 montre les différentes actions réalisées par le DSO. Sur le panneau de contrôle du DSO, il est nécessaire d'armer la première sous station DS SS1. En cliquant sur le bouton *ArmingSS* de SubStation1, le DSO envoie la demande d'armement à la première sous-station en invoquant son service Web SW2-ordre_d'armement. La sous-station DS SS1 effectue alors l'armement de son MCDTU et renvoie la confirmation d'armement. Le DSO va ensuite armer autant de stations que nécessaire pour arriver au seuil de puissance demandé. Quand le seuil est dépassé, le DS CC renvoie automatiquement une réponse au TS CC pour confirmer que l'armement s'est bien effectué et que le système de distribution est prêt pour le délestage prévu. Sur le panneau de contrôle du DSO, une station armée s'affiche en rouge, une fois l'armement effectué.

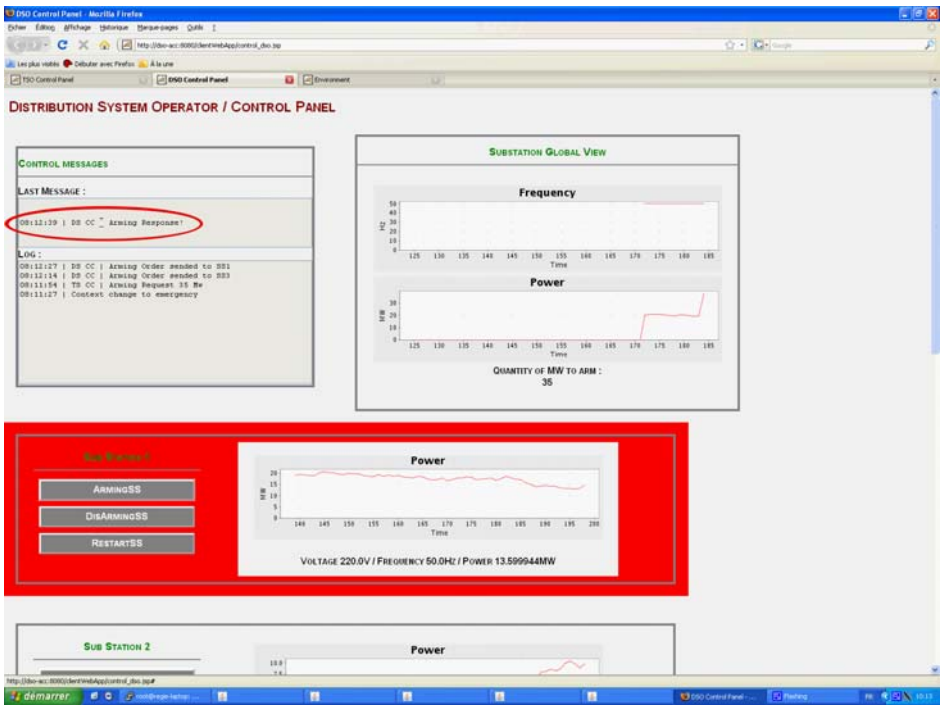


Figure 61 : Envoi du message de confirmation d'armement du DS CC au TS CC.

Une fois la confirmation d'armement est reçue par le TS CC, il prévient la TS SS pour se préparer à l'exécution d'un délestage automatique éventuel. Pour cela, le TS CC invoque le service Web SW3-préparation_de_délestage de la TS SS. À partir de ce moment-là, si la TS SS détecte une situation d'urgence (par exemple par une baisse de la fréquence en dessous d'un seuil fixé), la TS SS diffusera automatiquement à toutes les DS SS une invocation du service Web SW4- activation_de_délestage, et celles des DS SS qui sont armées se déconnectent. Dans notre démonstrateur, nous pouvons pour cela simuler une baisse de fréquence sur le réseau de transport dans un panneau spécifique "Environnement".

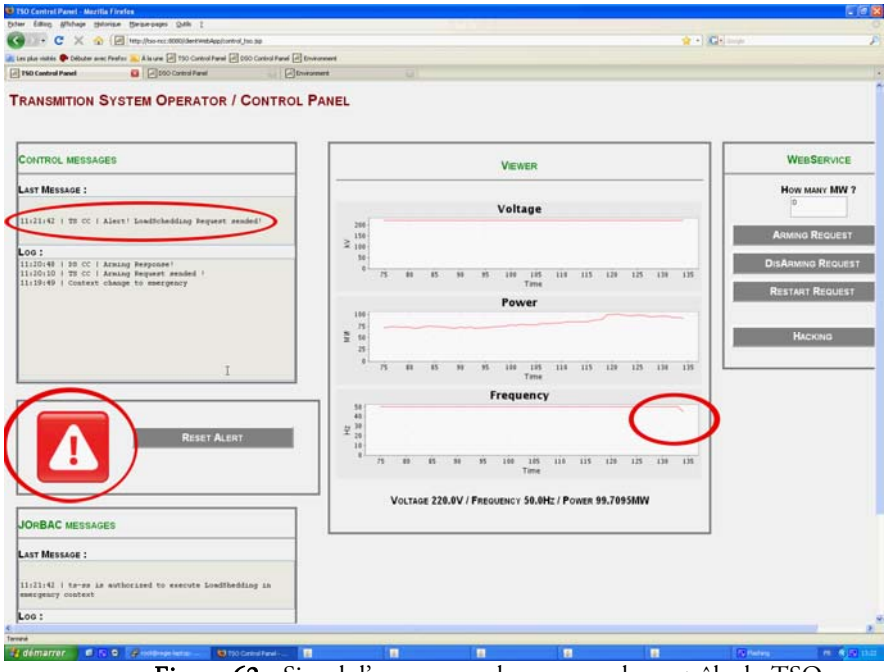


Figure 62 : Signal d'urgence sur le panneau de contrôle du TSO.

Dans le panneau du TSO, nous pouvons voir un message d'alerte annonçant l'exécution du délestage (Figure 62). Notons que lors de la réception d'un message d'alerte, un bouton d'alerte passe au rouge. Il y a deux manières de repasser au vert, soit automatiquement lorsqu'un nouveau message signale que l'alerte est passée, soit manuellement par le TSO lorsqu'il a pris des mesures pour traiter la situation d'alerte. Sur les courbes du panneau de commande du TSO, nous pouvons remarquer une légère baisse dans la courbe de fréquence puis un retour à la normale (après délestage). La Figure 63 montre les signes du délestage sur le panneau de contrôle du DSO, nous pouvons remarquer une baisse significative sur la courbe de puissance des différentes sous stations délestées.



Figure 63 : Signes de l'arrêt des sous stations armées.

Au niveau du DSO, nous pouvons voir pour les sous-stations armées (en rouge) les courbes et les valeurs numériques de puissance qui tombent à 0. Lorsque l'urgence est terminée sur le système de transport (par exemple, parce qu'un accroissement de la production permet de compenser la surconsommation), le TSO envoie une demande de redémarrage au DSO pour réintégrer des différentes sous-stations délestées. Sur le panneau du DSO, cette demande du TSO s'affiche, et le DSO envoie à son tour l'ordre de redémarrage à toutes les sous-stations armées puis délestées (invocation du service Web SW5-réintégration). Nous avons montré comment un fonctionnement normal de ce scénario est simulé dans notre démonstrateur. Dans la section suivante, nous allons simuler un certain nombre d'erreurs et montrer comment la plateforme traite ces erreurs.

5.4 Fonctionnement du scénario en présence d'erreurs

Dans cette section, nous allons montrer le fonctionnement erroné des services Web, en considérant plusieurs types d'erreur, notamment l'appel d'un même service Web deux fois de

suite (par exemple si un intrus observe le message du premier appel et tente de le rejouer), ou l'appel d'un service Web différent de celui attendu, ou encore des tentatives de violation de politique locale. Ces erreurs peuvent être considérées comme représentatives d'attaques par des intrus qui tenteraient de provoquer un black-out. Nous verrons également des erreurs se manifestant par le non-respect d'échéances temporelles ou par la perte de messages, qui sont représentatives d'attaques en déni de service sur les réseaux. Il est possible de simuler ces erreurs dans notre démonstrateur, soit en lançant des commandes erronées, soit en utilisant des mécanismes implémentés seulement pour tester le comportement en présence d'erreurs. Notons aussi que nous avons intégré au démonstrateur un mécanisme d'affichage des états de chaque automate temporisé de chaque CIS.

5.4.1 Double invocation du même service

Ceci peut être facilement simulé sur notre démonstrateur. Au niveau du TS CC, le TSO lance d'abord une demande d'armement pour une certaine puissance et reçoit l'acquittement correspondant, puis lance une deuxième fois la même demande d'armement, ce qui risquerait de faire désarmer l'ensemble des DS SS. L'erreur est détectée par le CIS du TS CC puisque l'automate, après la première demande et son acquittement par le DS CC, est dans un état où il attend un message soit de désarmement, soit de redémarrage (voir Figure 64) mais n'attend pas de nouvelle demande d'armement.

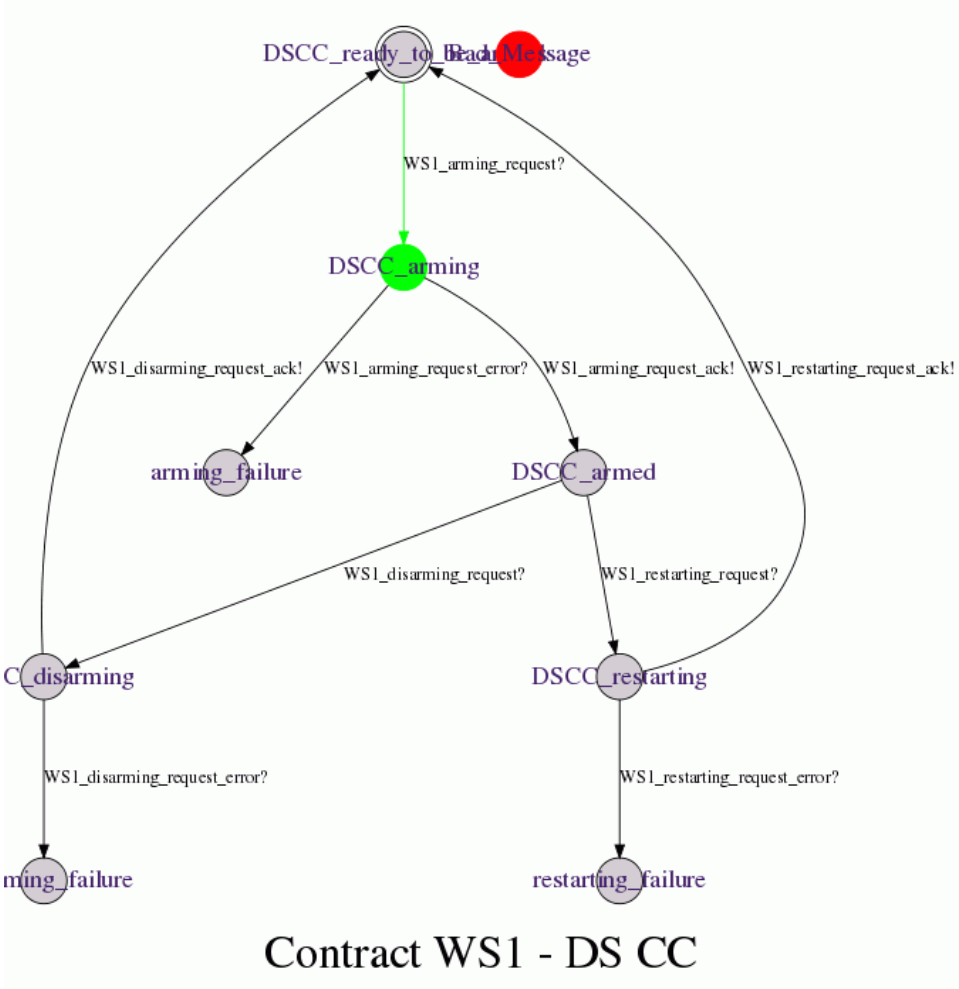


Figure 64 : détection de la réception d'un message erroné.

L'erreur déclenche alors l'envoi d'un message d'erreur au TS CC, et ce message s'affiche sur le panneau de contrôle du TSO : « Arming already executed | Denied by Policy ». Si l'intrus avait pris aussi le contrôle du CIS du TS CC, et que le second message (erroné) d'armement était reçu par le CIS du DS CC, celui-ci détecterait aussi l'erreur de la même façon, mais enverrait un message au DSO pour qu'il prenne des mesures correctives, suite à la défaillance du TS CC.

Notons qu'il serait difficile pour un intrus de rejouer ou de forger un tel message directement sur le réseau vers le DS CC en raison des signatures utilisées pour sécuriser les canaux de communication dans l'architecture CRUTIAL, mais même s'il réussissait à le faire, l'erreur serait détectée et traitée de la même façon par le CIS du DS CC.

5.4.2 Erreur d'Invocation non attendue

Dans ce second scénario d'erreur simulé sur notre démonstrateur, le TSO envoie une demande d'armement au DSO, et reçoit l'acquittement d'armement, puis lance une demande de désarmement au niveau du panneau de contrôle du TSO. En déclenchant un désarmement de toutes les stations armées au niveau du DSO et une fois que toutes les stations sont désarmées, une réponse de désarmement est envoyée au TSO. Jusque-là, il s'agit d'un fonctionnement normal. Si à ce moment le TSO lance une demande de redémarrage, cette erreur est détectée de la même façon que la précédente par les CIS.

5.4.3 Violation d'une condition OrBAC sur le contexte

Dans le démonstrateur, on a vu qu'on pouvait changer l'environnement simulé du réseau électrique en modifiant certains paramètres (par exemple, la fréquence sur le réseau de transport) grâce au panneau "Environnement". Ces paramètres définissent le contexte d'exécution, par exemple les situations de pré-urgence (risque de surconsommation à court terme) ou d'urgence (baisse de fréquence).

Pour simuler une erreur due au contexte, nous positionnons le système dans le contexte normal. Ensuite, nous tentons de faire lancer par le TSO une demande d'armement, de désarmement ou de redémarrage ou par le DSO un ordre d'armement, de désarmement ou de redémarrage. Ces actions sont interdites par les politiques OrBAC, puisque ces commandes ne peuvent être exécutées que dans un contexte de pré-urgence, ou après un délestage (pour le redémarrage). Ceci est détecté par le moteur JOrBAC, et conduit à l'affichage d'une alerte, par exemple "*TS CC Arming is not authorized | denied by Policy*". suivi d'un message du type "*<nom de l'utilisateur> is NOT authorized to execute Arming in normal context*".

5.4.4 Erreur due à une tentative d'extension de privilège

Supposons qu'un utilisateur, disons Roger, qui n'est pas habilité à jouer le rôle de TSO, tente d'envoyer une demande de désarmement. Ceci peut être facilement mis en œuvre sur notre démonstrateur.

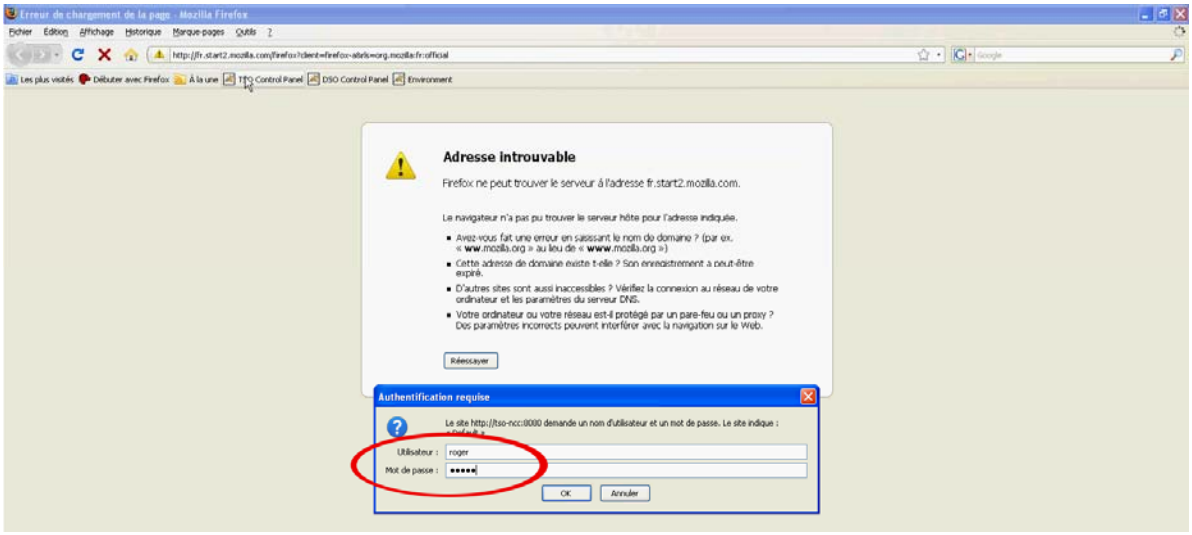


Figure 65 : connexion au TSO avec les identifiants roger/roger.

Cette action est en violation de la politique OrBAC et cela est détecté par le moteur JOrBAC, qui va afficher un message pour signaler que cette action n'est pas autorisée : "*TS CC arming is not authorized | denied by policy*" et pour quelle raison "*roger is not authorized to execute Arming in emergency context*".

5.4.5 Erreur liée à une échéance temporelle non respectée

Dans notre démonstrateur, il est facile de simuler une échéance temporelle non respectée. Par exemple, on peut faire lancer une demande d'armement par le TSO pour une certaine puissance en mégawatts, et faire en sorte que le DSO ne réalise pas son obligation d'armer suffisamment de sous-stations. Premièrement, nous réalisons la demande d'armement depuis le panneau de contrôle du TSO après avoir indiqué la quantité de mégawatts qu'il faut être prêt à délester. Une fois le message envoyé, nous attendons au moins une minute (durée de la minuterie définie dans le contrat mis en place dans le démonstrateur — dans une vraie infrastructure, ce délai serait probablement plus long).

La Figure 66 montre le fonctionnement de l'automate correspondant au service SW1-demande_d'armement pour le CIS du DS CC. Une erreur est détectée en raison du dépassement du délai sans réception de l'acquittement automatique. Nous mettons en évidence l'occurrence d'une erreur due à un non respect du timeout établi dans le contrat. Ceci est signalé dans notre démonstrateur, puisqu'il permet d'afficher l'état de chaque automate.

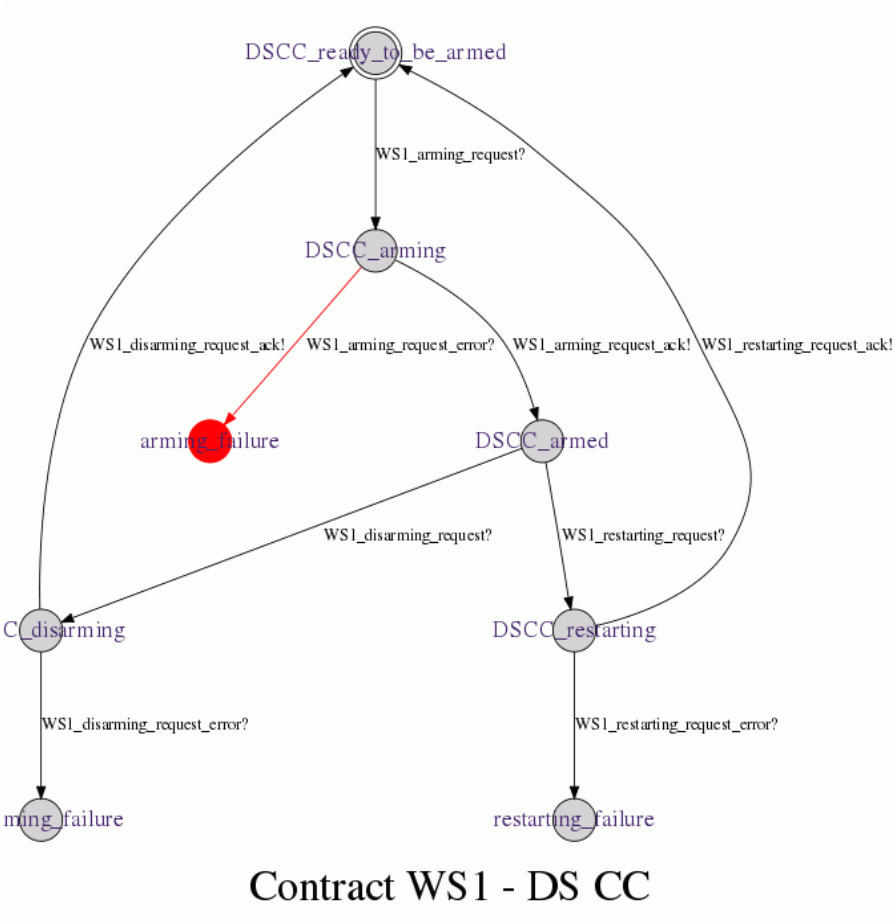


Figure 66 : Contrat dans un état d'erreur après un timeout.

5.4.6 Erreur dans les contrats en raison d'un message non attendu

Nous allons maintenant montrer l'utilisation du panneau *hacking* que nous avons intégré au démonstrateur pour le TS CC et le DS CC. Lorsqu'on clique sur le bouton "Hacking", les contrôles par le moteur JOrBAC local sont inhibés. Il est alors facile d'émettre des commandes qui seraient normalement bloquées par le contrôle d'accès de la politique locale.

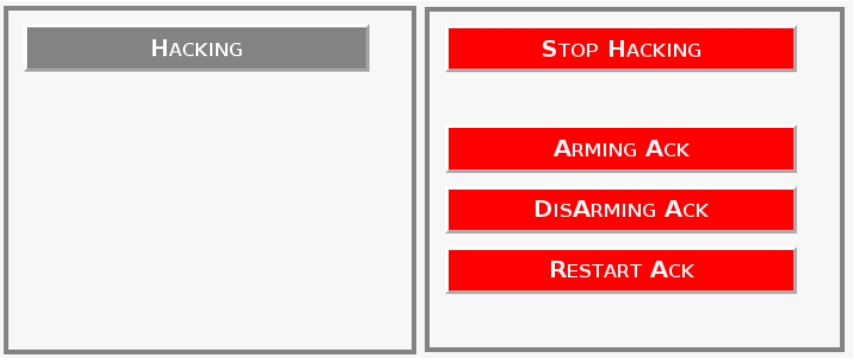


Figure 67 : Boutons d'appel des web-services permettant le hacking.

Par exemple, il est ainsi possible de faire lancer un acquittement de désarmement par le DSO, alors qu'aucune demande de désarmement n'a été reçue. Ceci est détecté par l'automate du CIS du DS CC pour SW1, comme indiqué par la Figure 68 : un état rouge apparaît dans l'affichage de l'automate, signalant la réception d'un message non autorisé dans l'état actuel.

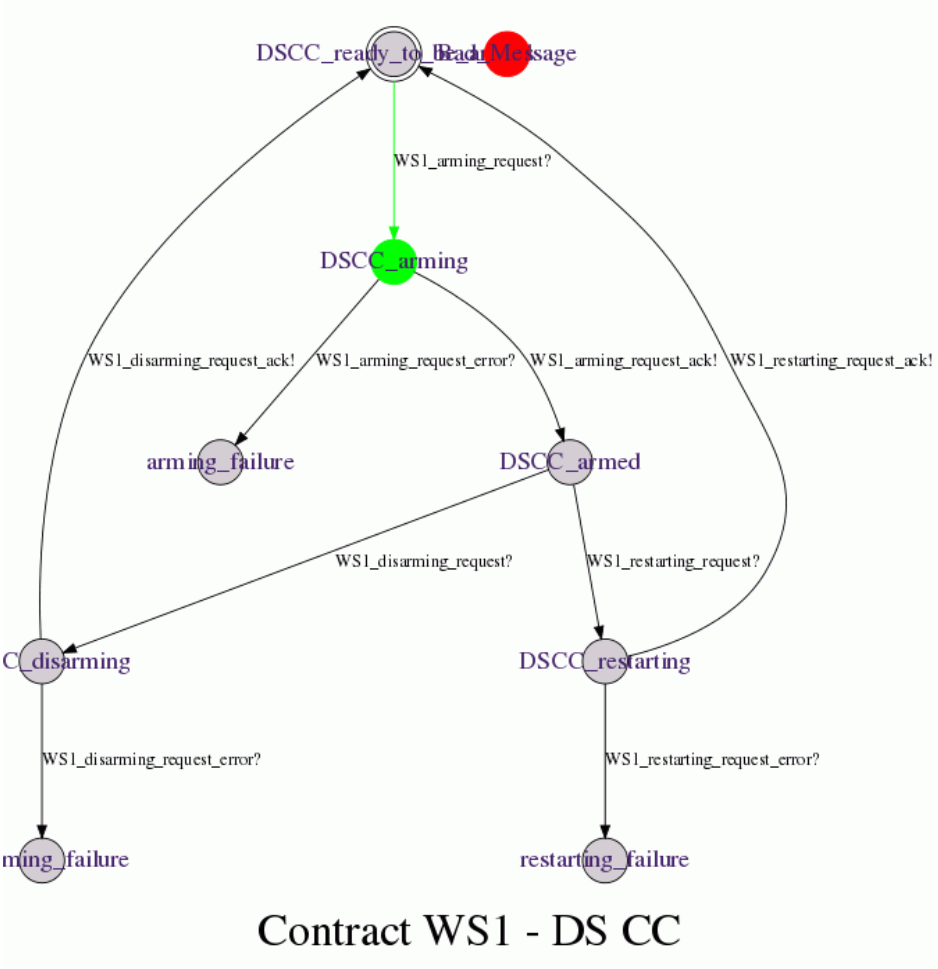


Figure 68 : Le contrat signale la réception d'un message erroné.

5.5 Conclusion du chapitre 5

Dans ce chapitre, nous avons montré l'application et l'utilisation du schéma-cadre PolyOrBAC à un scénario exécuté sur un démonstrateur simulant le fonctionnement d'une infrastructure de production, de transport et de distribution d'électricité. Nous avions pour objectif dans ce chapitre de décrire la mise en œuvre de PolyOrBAC sur des machines et un réseau simulant en détail une architecture CRUTIAL. Nous avons présenté les différents composants utilisés pour implémenter le démonstrateur. Nous avons montré comment ce démonstrateur met en évidence le fonctionnement normal du scénario, et enfin comment l'on traite les erreurs grâce aux différents mécanismes de contrôle d'accès d'OrBAC et de vérification statique et dynamique des politiques-contrats représentées par des automates temporisés.

Durant le test de l'implémentation, nous avons rencontré un certain nombre de difficultés parmi lesquelles nous pouvons citer les besoins importants en mémoire de chaque machine virtuelle, le fait est qu'une fois que les services ont été déployés sur l'ensemble des machines, l'exécution du prototype est de plus en plus en lente, ce qui empêchait le fonctionnement correct de la plateforme. Afin de remédier à ce type de problèmes, nous avons dû augmenter la capacité mémoire des machines physiques, et optimiser les différentes machines virtuelles, en ne gardant que les outils et les composants nécessaires au bon fonctionnement du prototype. Notons néanmoins que ce démonstrateur a été équipé des moyens d'observations et de paramétrage de l'environnement simulé qui permettent de jouer des scénarios variés, et d'observer leur fonctionnement normal et en présence d'erreurs. Il faut aussi mentionner les outils développés pour aider à la définition des politiques OrBAC et à la vérification statique des politiques-contrats.

Conclusion générale

En raison de ses vulnérabilités physiques et logiques, une infrastructure critique (IC) peut subir des défaillances, et en raison des interdépendances entre IC, de simples défaillances peuvent avoir des conséquences dramatiques sur l'ensemble des infrastructures critiques. Dans cette thèse, nous nous sommes intéressés principalement aux systèmes d'information et de communication ; c'est-à-dire l'infrastructure d'information critique, ou IIC. Nous avons proposé une nouvelle approche pour répondre aux problèmes de sécurité que rencontre une IIC, plus particulièrement, ceux liés au contrôle d'accès et à la collaboration. Le but est d'offrir à chaque organisation faisant partie de l'IIC la possibilité de collaborer avec les autres, tout en maintenant un contrôle sur ses données et sa politique de sécurité interne. Nous avons développé et modélisé PolyOrBAC, un schéma-cadre de contrôle d'accès pour la collaboration entre organisations, basé sur le modèle de contrôle d'accès OrBAC et sur la technologie des services Web. Nous avons également conçu des mécanismes de vérification des interactions par service Web grâce à des automates temporisés afin de garantir le fonctionnement prévu au cours des phases de collaboration. Nous avons montré l'applicabilité de cette plateforme dans le contexte des infrastructures critiques en général, et plus particulièrement dans le contexte des réseaux d'énergie électrique.

Bilan des contributions

Les principales contributions de cette thèse sont résumées ci-après :

- **Description des infrastructures critiques et infrastructures d'information critiques et définition des besoins en sécurité :** Nous avons identifié les besoins des IIC en matière de sécurité. Nous avons ensuite énuméré les vulnérabilités les plus importantes qui touchent les infrastructures critiques et les infrastructures d'information critiques. Nous avons également présenté deux projets de recherche qui proposent des moyens pour se protéger vis-à-vis des vulnérabilités précédemment décrites, en proposant des schémas de sécurité dans les infrastructures d'information critiques. Dans la description de ces projets, nous avons mis l'accent sur le but, l'organisation et les contributions de chaque projet. Enfin, nous avons présenté le projet CRUTIAL, dans lequel s'intègre notre travail.

- **État de l'art des modèles de contrôle d'accès pour la collaboration :** nous avons décrit le fonctionnement, les avantages et les contributions de chacun des principaux modèles de contrôle d'accès, en analysant comment ces modèles peuvent répondre

ou non aux besoins de collaboration et de contrôle d'accès que nous avions définis précédemment. Nous avons étudié et analysé l'état de l'art dans le domaine des politiques et modèles de sécurité traditionnels (RBAC, OrBAC), et les modèles de sécurité pour la collaboration (Multi-OrBAC, OV, O2O). Pour chaque modèle, nous avons détaillé les avantages et les limites, le but étant de déterminer quel modèle pourrait convenir à nos besoins en contrôle d'accès et collaboration. Cette analyse des modèles de contrôle d'accès collaboratifs existants nous a aidé à construire un nouveau schéma qui satisfait les exigences de confidentialité et d'autonomie que nous avions établies précédemment pour les IIC, exigences qui ne sont pas totalement satisfaites par les modèles existants.

- **Conception du schéma-cadre PolyOrBAC pour la collaboration sécurisée entre organisations :** Ce schéma est basé sur le modèle de contrôle d'accès OrBAC, que nous utilisons pour spécifier des politiques de contrôle d'accès pour les différentes organisations concernées, et sur la technologie des services Web qui permet de fournir une plateforme de collaboration et d'interopérabilité entre ces organisations. Nous avons aussi conçu des mécanismes de vérification des interactions de service Web grâce à des politiques-contrats définies à l'aide d'automates temporisés, de façon à garantir le respect des contrats de collaboration négociés et signés par les organisations concernées. Nous avons mis l'accent sur la manière dont les différentes organisations collaborent et gèrent leurs politiques de contrôle d'accès respectives. Nous avons introduit les notions d'*utilisateur virtuel* et d'*image de service Web*, qui facilitent l'expression de règles liées aux interactions au sein des politiques de contrôle d'accès locales à chaque organisation qui collabore dans l'IIC.

- **Étude de cas et applicabilité du modèle :** Nous avons montré comment ce schéma-cadre peut s'appliquer à une infrastructure critique particulière, le réseau de transport et de distribution d'énergie électrique en Europe. Nous avons analysé le fonctionnement de ce type d'infrastructure et décrit des scénarios montrant les interactions entre organisations faisant partie de cette infrastructure. L'application de PolyOrBAC dans cet exemple illustre les avantages de notre approche. Pour cela, nous avons décrit le fonctionnement de ce type d'infrastructure et nous nous sommes focalisés sur un scénario réel de délestage semi-automatique en cas d'urgence, scénario qui pourrait être la cible d'attaquants pour provoquer un black-out dans ces conditions particulières. L'applicabilité du schéma-cadre PolyOrBAC à ce scénario a été démontrée en précisant les règles correspondantes dans les politiques OrBAC de chaque organisation collaborant au sein de l'IIC et les automates temporisés qui vérifient les interactions entre ces mêmes organisations.

- **Développement d'un démonstrateur et applicabilité de notre approche dans un environnement émulé :** Nous avons développé un démonstrateur pour montrer en pratique l'application et l'utilisation du schéma-cadre PolyOrBAC au scénario décrit précédemment. Nous avions pour objectif de décrire la mise en œuvre de PolyOrBAC sur des machines et un réseau simulant en détail une architecture CRUTIAL. Nous avons présenté les différents composants utilisés pour

implémenter le démonstrateur. Nous avons montré comment ce démonstrateur met en évidence le fonctionnement normal du scénario, et enfin comment les erreurs sont traitées, grâce aux différents mécanismes de contrôle d'accès d'OrBAC et de vérification statique et dynamique des politiques-contrats représentées par des automates temporisés. Notons que nous avons équipé ce démonstrateur de moyens d'observation et de paramétrage de l'environnement simulé qui permettent de jouer des scénarios variés et d'observer leur fonctionnement normal et en présence d'erreurs d'une infrastructure critique de complexité raisonnable. Il faut aussi mentionner les outils que nous avons conçus et développés pour aider à la définition des politiques OrBAC et à la vérification statique des politiques-contrats.

Perspectives de recherche

Nous ne pouvons clôturer ce manuscrit sans aborder les prolongements et les perspectives souhaitables à nos travaux.

- **Développement et standardisation des notions originales que nous avons développées pour le contrôle d'accès collaboratif, telles que les images de service Web et utilisateurs virtuels :** nous pensons notamment à une standardisation W3C, ou OASIS. Pour cela il sera nécessaire de prendre en compte d'autres aspects propres à la création et au fonctionnement des services Web.

- **Prise en compte de la notion de disponibilité :** la disponibilité peut être traitée par l'utilisation entre autres de règles d'obligation, qui imposent que l'organisation prestataire fournira suffisamment de ressources pour répondre aux besoins et demandes des organisations clientes, même en cas d'événements imprévus tels que des défaillances ou les attaques que subiraient son système d'information et de communication.

- **Prise en compte de la notion d'intégrité :** notre approche peut être étendue pour contrôler les flux d'information, et en particulier interdire les flux d'informations venant de tâches ayant un niveau de criticité bas vers des tâches de plus haut niveau de criticité, sauf lorsque ces flux sont validés par des moyens adéquats, grâce à des mécanismes de tolérance aux fautes, conformément au modèle de Totel [Totel et al., 1998].

- **Application du schéma-cadre PolyOrBAC à une infrastructure de collaboration réelle :** il est notamment possible d'appliquer PolyOrBAC à des infrastructures de collaboration réelles, telles que les infrastructures de transport. Il faudra alors prendre en compte les besoins et spécificités de ces infrastructures qui différent de ceux d'un réseau d'énergie électrique. Ceci devrait être relativement facile si les infrastructures d'information appliquent l'architecture CRUTIAL, qui a servi de cadre au développement de PolyOrBAC et qui intègre déjà les mécanismes nécessaires au contrôle des interactions entre organisations, mais aussi d'autres

mécanismes de tolérance aux fautes et de protection en particulier vis-à-vis des attaques en déni de service.

- **Déploiement et test du démonstrateur sur un réseau électrique existant réel :** notamment au sein du réseau électrique français ou européen. Ceci devrait pouvoir se faire de façon progressive, en adaptant les concepts d'organisation et d'interaction par service Web, organisation par organisation, et en faisant le lien entre les politiques de sécurité de ces organisations et les mécanismes de contrôles d'accès déjà mis en place (pare-feux, gestion des droits d'accès dans les systèmes d'exploitation, les bases de données, etc.).

Annexes

Contrôle d'accès et vérification pour SW3-préparation_de_délestage

Le service Web SW3-préparation_de_délestage gère les interactions entre le TS CC et le TS SS. Ces interactions correspondent à la préparation de délestage par le TSO (donc l'organisation TS CC) à la TS SS, ainsi qu'à la demande de d'annulation de cette préparation pour délestage.

Règles OrBAC pour la préparation de délestage par le TS CC

Une fois l'acquittement reçu (en considérant que l'armement des DS SS a pu être effectué), le TSO doit envoyer une requête de préparation de délestage (SW3-preparation_de_délestage) pour préparer la TS SS à faire un délestage de charge sur les DS SS armées préalablement, et ce si et seulement si les conditions d'urgence sont remplies. Dans les situations d'urgence, lorsque le TSO décide de lancer l'activité de délestage, il invoque « SW3-préparation_de_délestage » de la TS SS. Lorsque la personne (par exemple, Martin) qui est l'opérateur TSO invoque le service SW3, cela correspond à l'exécution de l'action *invoquer* sur l'objet *image_SW3*. Cette action est permise par la politique OrBAC du TS CC si l'utilisateur Martin a été authentifié, s'il joue bien le rôle TSO, et s'il existe une règle de permission pour le rôle TSO de réaliser l'activité correspondant à cette action sur la vue correspondant à cet objet dans le contexte de pré-urgence actuel. La séquence de règles OrBAC suivante présente les droits d'accès pour l'utilisation du service SW3 au niveau de l'organisation TS CC.

Permission(TS CC, TSO, accéder, vue_SW3, pré-urgence) ∧ Habilite(TS CC, Martin, TSO) ∧ Considère(TS CC, invoquer, accéder) ∧ Utilise(TS CC, image_SW3, vue_SW3) ∧ Définit(TS CC, Martin, invoquer, image_SW3, pré-urgence) ⇨ Est_permis(Martin, invoquer, image_SW3)

Tableau 11 : Représentation de la règle OrBAC pour SW3 du côté TS CC.

Cette séquence peut être interprétée comme suit : la politique OrBAC du TS CC contient un certain nombre de règles pour gérer le service Web SW3-préparation_de_délestage. Première règle : au niveau de la TS CC, le rôle TSO a le droit, dans un contexte de pré-urgence, d'exécuter les actions incluses dans l'activité *accéder* sur la vue *vue_SW3* correspondant à SW3. Deuxième règle : nous attribuons le rôle TSO à Martin. Troisième règle : nous incluons l'action

invoquer dans l'activité *accéder*. Quatrième règle : nous incluons l'objet *image_SW3* dans la vue *vue_SW3*. Cinquième règle : nous précisions que le sujet Martin a la possibilité de réaliser l'action *accéder* sur l'objet *image_SW3* dans un contexte spécifique pré-urgence. De cet ensemble de règles on déduit le prédicat Est_permis qui affirme que le sujet Martin a le droit d'exécuter l'action *accéder* sur l'objet *image_SW3*.

Par ailleurs, comme nous l'avons précisé au chapitre 3, pour chaque Service Web un contrat doit être signé par le client et le prestataire, contenant la politique-contrat qui contrôle les échanges de messages. Cette politique-contrat est définie par deux automates temporisé, l'un du côté client, l'autre du côté prestataire, chacun installé dans le CIS de son organisation. La section suivante présente l'automate temporisé correspondant à SW3-préparation_de_délestage du côté du TS CC.

Après, si le TSO décide d'annuler la préparation de délestage, il invoque alors le "WS3-LS-cancellation" de la TS SS et attend de l'acquittement ("WS3-LS-cancellation-ack").

Automate de SW3-préparation_de_délestage au niveau du TS CC

Selon la politique-contrat du service SW3 et comme illustré dans la Figure 42, du côté du TS CC (le client de SW3), l'automate attend une invocation pour une préparation de délestage (en provenance du TSO). Quand le message correspondant à cette invocation est intercepté par le CIS du TS CC, la transition correspondante (*WS3-prepare_for_LS*) est activée dans l'automate.

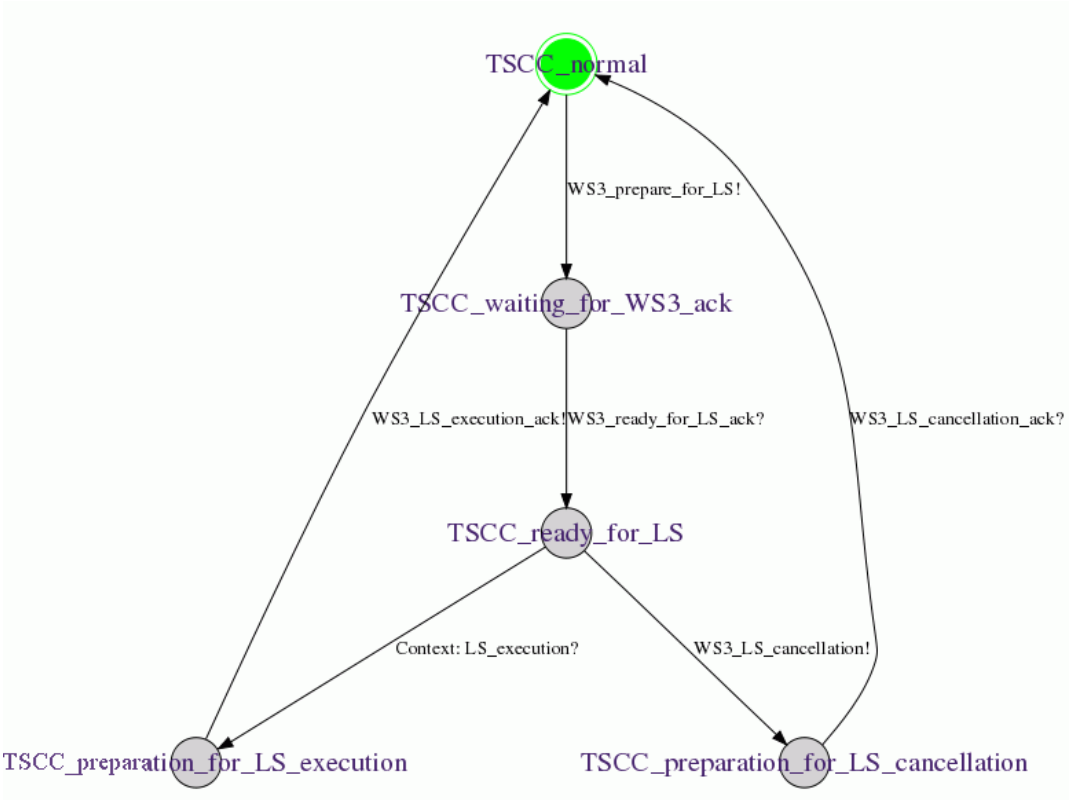


Figure 69 : Automate de SW3-préparation_de_délestage au niveau du TS CC.

L'automate atteint un état où il attend un acquittement *WS3-ready_for_LS_ack* du TS SS. Ensuite, si l'état d'urgence survient au niveau de la TS SS, cette dernière va exécuter le délestage en envoyant LS_execution au TS CC et en attendant le WS3-LS_execution_ack. Sinon, si la situation de pré-urgence disparaît, le TSO peut décider d'annuler la préparation de délestage. Pour cela, le TS CC envoie le message *WS3-LS_cancellation* vers le TS SS et attend l'acquittement *WS3-LS_cancellation_ack* du TS SS, qui remettra l'automate dans l'état initial.

Règles OrBAC de SW3-préparation_de_délestage dans la politique du TS SS

Du côté de la TS SS, la réception de la requête de SW3-préparation_de_délestage déclenche l'exécution pour le compte de l'utilisateur virtuel *uv-SW3-TS_CC*. Cet accès (*uv-SW3-TS_CC* à la console) est vérifié selon la politique de contrôle d'accès du TS SS et est accordée en fonction de la séquence OrBAC décrite dans le Tableau 8.

Permission(TS SS, Rôle_SW3, afficher, console_TSSS, pré-urgence) ∧
Habilite(TS SS, uv-SW3-TS_CC, Rôle_SW3) ∧
Considère(TS SS, écrire, afficher) ∧
Utilise(TS SS, terminal_3, console_TSSS) ∧
Définit(TS SS, uv-SW3-TS_CC, écrire, terminal_3, pré-urgence)
⇨ Est_permis(uv-SW3-TS_CC, écrire, terminal_3)

Tableau 11 : Représentation de la règle OrBAC pour SW3 du côté TS SS.

Première règle : le rôle *Rôle_SW3* a le droit, quelque soit le contexte, d'afficher des messages d'urgence (activité *afficher*) sur la console du TSSS (vue *console_TSSS*). Deuxième règle : nous attribuons le rôle *Rôle_SW3* au sujet *uv-SW3-TS_CC*, qui représente le TS CC pour exécuter le service SW3. Troisième règle : nous incluons l'action *écrire* dans l'activité *afficher*. Quatrième règle : nous incluons l'objet *terminal_3* dans la vue *console_TSSS*. Cinquième règle : nous précisions que le sujet *uv-SW3-TS_CC* a la possibilité de réaliser l'action *écrire* sur l'objet *terminal_3*, quelque soit le contexte. De cet ensemble de règles, on déduit le prédicat Est_permis qui affirme que le sujet *uv-SW3-TS_CC*, a le droit d'exécuter l'action *écrire* sur l'objet *terminal_3*.

Automate de SW3-préparation_de_délestage dans le CIS du TS SS

Nous allons maintenant analyser la façon dont l'automate de la politique-contrat liée à SW3-préparation_de_délestage vérifie dans le CIS du TS SS vérifie les invocations en provenance du côté client (Figure 43). Au départ, cet automate est en attente du message *WS3-prepare_for_LS* du TS CC, puis survient l'envoi d'un acquittement *WS3-ready_for_LS_ack* par le TS SS au TS CC exprimant que le délestage peut être effectué à présent.

Ensuite, l'automate est en mesure de traiter deux cas de figure :

- Si l'état d'urgence est détecté, le délestage est exécuté et le message LS_execution est envoyé de la part de la TS SS au TS CC, puis attend l'acquittement WS3_LS_execution_ack, et l'automate revient ainsi à l'état initial.

- Au contraire, si le TS CC décide d'annuler le délestage, il envoie un message WS3_LS_cancellation, la TS SS va donc annuler l'étape de préparation de délestage, va renvoyer un acquittement WS3_LS_cancellation_ack et reviendra à l'état initial.

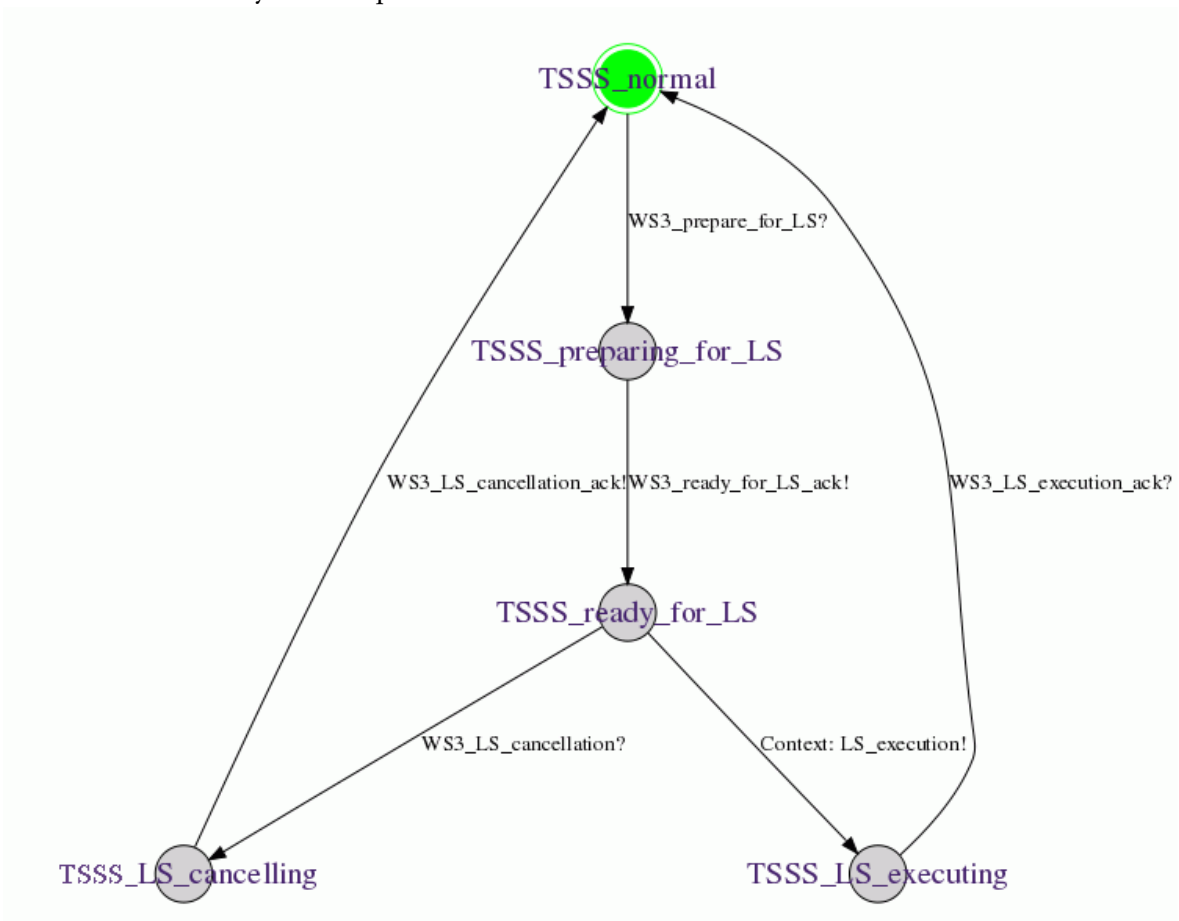


Figure 70 : Automate de SW3-préparation_de_délestage au niveau du TS SS.

Contrôle d'accès et vérification pour SW2-ordre_d'armement

Le service Web SW4-activation_de_délestage gère les interactions entre le TS SS et chacune des DS SS armées par le DS CC. Ces interactions correspondent à l'ordre de délestage des sous-stations DS SS.

Automate de SW4-activation_de_délestage du côté TS SS

Pour SW4, il y a autant de contrats que de sous-stations DS SS contrôlées par le DS CC. Dans le CIS du TS SS, il a donc autant d'automates, chacun correspondant à une seule DS SS.

Permission(TS SS, TSO, accéder, vue_SW4, urgence) ∧
Habilite(TS SS, Automate, TSO) ∧
Considère(TS SS, invoquer, accéder) ∧
Utilise(TS SS, image_SW4, vue_SW4) ∧
Définit(TS SS, Automate, invoquer, image_SW4, urgence)
⇨ Est_permis(Automate, invoquer, image_SW4)

Tableau 12 : Représentation de la règle OrBAC pour SW4 du côté TS SS.

Au niveau de la TS SS, lorsque des conditions d'urgence sont présentes (par exemple des conditions concernant la fréquence, puissance des sous stations de distribution), une sentinelle automatique (automate) au niveau de l'organisation TS SS, envoie une requête WS4-Load-Shedding-DSSS aux sous stations DSSS préalablement armées par le DSO, afin d'effectuer leur délestage respectif.



Figure 71 : Automate pour SW4-activation_de_délestage du côté du TS SS.

Règle OrBAC de l'armement dans la politique du DS SS

Lorsque la TS SS invoque le service Web SW4-activation_de_délestage sur les sous-stations préalablement armées, le processus correspondant est lancé pour le compte de l'utilisateur virtuel uv-SW4-TS, et ce processus exécute l'action délestage sur l'objet *MCDTU*. Cette action est autorisée par la séquence OrBAC décrite dans le Tableau 9.

Permission(DS SS, RôleDSO, préparer, SS_Circuits, urgence) ∧
Habilite(DS SS, uv-SW4-DS, RôleDSO) ∧
Considère(DS SS, délestage, préparer) ∧
Utilise(DS SS, MCDTU, SS_Circuits) ∧
Définit(DS SS, uv-SW4-DS, délestage, MCDTU, urgence)
⇨ Est_permis(uv-SW4-DS, délestage, MCDTU)

Tableau 12 : Représentation de la règle OrBAC pour SW4 du côté DS SS.

Automate de SW4-activation_de_délestage du côté du DS SS

Au niveau des différentes sous stations DS SS préalablement armées, l'opération de délestage est alors automatique à la réception du message WS4_Loadshedding_DSSS.



Figure 72 : Automate pour SW4-activation_de_délestage au niveau du DS SS.

Si le TSO s'aperçoit que la situation d'urgence n'est plus valable, il peut alors soit désarmer les sous stations de distribution préalablement armées, soit les « réintegrer » (remettre en fonctionnement normal) après leur délestage respectif, il en informe alors le DSO, qui invoque SW5-réintegration au niveau de chaque DS SS. Les sous stations de distribution sont de nouveau opérationnelles, et reprennent alors leur fonctionnement normal au sein du système de distribution.

Bibliographie

[Abou El Kalam *et al.*, 2003] Anas Abou El Kalam, Rania El Baida, Philippe Balbiani, Salem Benferhat, Frédéric Cuppens, Yves Deswarte, Alexandre Miège, Claire Saurel, Gilles Trouessin, "Organization Based Access Control", *4th IEEE International Workshop on Policies for Distributed Systems and Networks* (Policy'03), Côme, Italie, juin 2003, pp. 120-131. <http://www.orbac.org/publi/OrBAC/OrBac.pdf> (vérifié le 13/07/09).

[Abou El Kalam & Deswarte, 2006] Anas Abou El Kalam, Yves Deswarte, "Multi-OrBAC: a New Access Control Model for Distributed, Heterogeneous and Collaborative Systems", *8th IEEE International Symposium on Systems and Information Security* (SSI 2006), Sao Paulo, Brésil, 8-10 novembre 2006.
< http://hal.archives-ouvertes.fr/ccsd-00086523> (vérifié le 27/06/09).

[Abou El Kalam *et al.*, 2007a] Anas Abou El Kalam, Yves Deswarte, Amine Baina, Mohamed Kaaniche, "Access Control for Collaborative Systems: A Web Services Based Approach", *IEEE International Conference on Web Services* (ICWS 2007), 2007, pp. 1064-1071.
<http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4279707> (vérifié le 13/07/09).

[Abou El Kalam *et al.*, 2007b] Anas Abou El Kalam, Amine Baina, Hakem Beitollahi, Alysson Bessani, Andrea Bondavalli, Miguel Correia, Alessandro Daidone, Geert Deconincc, Yves Deswarte, Fabrizio Grandoni, Nuno Neves, Tom Rigole, Paulo Sousa, Paulo Veríssimo, *Preliminary Architecture Specification*, Deliverable D4, The CRUTIAL Project, FCUL, Lisbonne, Portugal, 2007, 106 pp.
<http://crutial.cesiricerca.it/content/files/Documents/Deliverables%20P1/WP4-D4-final.pdf> (vérifié le 13/07/09).

[Abou El Kalam *et al.*, 2008] Anas Abou El Kalam, Jérôme Ermont , Yves Deswarte, *Specification and Verification of Security Properties of e-Contracts*, Rapport LAAS n°08311, juin 2008, 7p. <http://www.laas.fr/~deswarte/Publications/08311.pdf> (Vérifié le 30/07/09).

[Abou El Kalam *et al.*, 2009] Anas Abou El Kalam, Yves Deswarte, Amine Baïna, Mohamed Kaâniche, *PolyOrBAC: a Security Framework for Critical Infrastructures*, Rapport LAAS N°09087, mars 2009, 28 p., à paraître dans *International Journal on Critical Infrastructure Protection*, Elsevier.
<http://www.laas.fr/~deswarte/Publications/09087.pdf>. (Vérifié le 30/07/09).

[Abou El Kalam *et al.*, 2009b] Anas Abou El Kalam, Amine Baina, Hakem Beitollahi, Alysson Bessani, Andrea Bondavalli, Miguel Correia, Alessandro Daidone, Wagner Dantas, Geert Deconinck, Yves Deswarte, Fabrizio Grandoni, Henrique Moniz, Nuno Neves, Paulo Sousa, Paulo Veríssimo, *Architecture, Services and Protocols for CRUTIAL*, Deliverable D18, The CRUTIAL Project, FCUL, Lisbonne, Portugal, 2009, 128 pp. <http://crutial.cesiricerca.it/content/files/Documents/Deliverables%20P3/WP4-D18-final.pdf> (vérifié le 07/10/09).

[Abou El Kalam & Deswarte, 2009] Anas Abou El Kalam, Yves Deswarte, "Critical Infrastructures Security Modeling, Enforcement and Runtime Checking", *3rd InternationalWorkshop on Critical Information Infrastructures Security* (CRITIS'08), 13-15 octobre, 2008, Frascati, Italy, à paraître dans Lecture Notes in Computer Science, Springer, 2009. <http://www.springerlink.com/content/840q672482007037/> (vérifié le 12/10/09).

[Alur & Dill, 1994] Rajeev Alur, David L. Dill, "A Theory of Timed Automata", *Theoretical Computer Science*, Vol. 126, No. 2, 1994, pp. 183-235. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.51.1093> (vérifié le 13/07/09).

[Amin, 2002] Massoud Amin, "Security Challenges for the Electricity Infrastructure", *Computer Magazine*, Vol. 35, No. 4, avril 2002, pp. 8-10. <http://index.ieeexplore.ieee.org/iel5/2/21810/01012423.pdf> (Vérifié le 27/06/09).

[Amin, 2003] Amin Massoud, "North America's Electricity Infrastructure: Are We Ready for More Perfect Storms?", *IEEE Security & Privacy*, Vol. 1, 2003, pp 19-25. <http://160.94.126.215/amin/Amin_IEEE_SP_Oct03.pdf> (Vérifié le 27/06/09).

[Ardagna *et al.*, 2006] Claudio Agostino Ardagna, Ernesto Damiani, Sabrina de Capitani di Vimercati, Pierangela Samarati, "A Web Service Architecture for Enforcing Access Control Policies", *Electronic Notes in Theoretical Computer Science*, Vol. 142, 2006, pp. 47-62. <http://spdp.dti.unimi.it/papers/vodca04.pdf> (vérifié le 13/07/09).

[Baïna *et al.*, 2008] Amine Baïna, Anas Abou El Kalam, Yves Deswarte, Mohamed Kaâniche, "A Collaborative Access Control Framework for Critical Infrastructures", Critical Infrastructure Protection II, Ed. Mauricio Papa & Sujeet Shenoi, *2nd Annual IFIP 11.10 Conference on Critical Infrastructure Protection*, 16-19 mars 2008, Arlington (VA USA), Springer, IFIP Series, ISBN 978-0-387-88522-3, pp.189-204. <http://www.springerlink.com/content/8l863582312285n1/fulltext.pdf> (Vérifié le 30/07/09).

[Bertino *et al.*, 1996] Elisa Bertino, Sushil Jajodia, Pierangela Samarati, "Supporting Multiple Access Control Policies in Database Systems", *IEEE Symposium on Security and Privacy*, Oakland, CA, mai 1996, pp. 94-107.

<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.40.4284> (vérifié le 13/07/09).

[Bertino *et al.*, 2000] Elisa Bertino, Silvana Castano, Elena Ferrari, Marco Mesiti, "Specifying and Enforcing Access Control Policies for XML Document Sources", *World Wide Web Journal*, Vol. 3, No. 3, 2000, pp. 139-151.
<http://www.springerlink.com/content/tpn715q101185762/> (vérifié le 27/06/09).

[Bialas, 2006] Andrzej Bialas, "Information Security Systems vs. Critical Information Infrastructure Protection Systems - Similarities and Differences", International Conference on Dependability of Computer Systems, 25-27 Mai 2006, *IEEE Computer Society*, pp. 60-67.
<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4024033&isnumber=4024018> (Vérifié le 27/06/09).

[Bologna *et al.*, 2006] Bologna Sandro, Giovanni Di Costanzo, Eric Luiijf, Roberto Setola, "An Overview of R&D Activities in Europe on Critical Information Infrastructure Protection (CIIP)", *Critical Information Infrastructures Security* (CRITIS 2006), Samos, Grèce, août-septembre 2006, pp. 91-102.
<http://www.springerlink.com/content/m762247677042467/> (Vérifié le 27/06/09).

[Brock, 2000] Jack L. Brock, *Critical Infrastructure Protection: Challenges to Building a Comprehensive Strategy for Information Sharing and Coordination*, Rapport du "United States General Accounting Office" No T-AIMD-00-268, 2000, 15 pp.
<http://www.gao.gov/archive/2000/ai00268t.pdf> (Vérifié le 27/06/09).

[Canneaux *et al.*, 2008a] Guillaume Canneaux, Benjamin Durand, Pierre Fersing, Nicolas de Roll Montpellier, Géraldine Van Saene, *Security over Critical Infrastructures : Report*, Rapport de Projet long 3ème année, ENSEEIHT, Toulouse, mai 2008, 38 pp.

[Canneaux *et al.*, 2008b] Guillaume Canneaux, Benjamin Durand, Pierre Fersing, Nicolas de Roll Montpellier, Géraldine Van Saene, *Security over Critical Infrastructures : User Manual*, manuel d'utilisateur pour Projet long 3ème année, ENSEEIHT, Toulouse, mai 2008, 18pp.

[Chandra, 2005] Jagdish Chandra, "Robust and Resilient Critical Infrastructure Systems", *38th Annual Hawaii International Conference* (HICSS '05), Hawaii 2005, p. 62-62.
<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=01173872> (Vérifié le 27/06/09).

[Coma, 2006] Céline Coma, "O2O: Managing Security Policy Interoperability with Virtual Private Organizations", *13th Annual Workshop of HP OpenView University Association*, Côte d'Azur, 21-24 mai 2006.
<http://www.springerlink.com/content/b18p558781307674/> (vérifié le 27/06/09).

[Cortés & Mishra, 1996] Mauricio Cortés, Prateek Mishra, "DCWPL: a Programming Language for Describing Collaborative Work", *ACM Conference on Computer Supported Cooperative Work* (CSCW '96), Boston, MA, USA, 16-20 novembre 1996, pp. 21-29. <http://portal.acm.org/citation.cfm?id=240080.240176> (vérifié le 13/07/09).

[CC, 2006] *Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model* (CCMB-2006-09-001), Version 3.1, Revision 1, septembre 2006. <http://www.commoncriteriaportal.org/thecc.html> (Vérifié le 27/06/09).

[Cuppens *et al.*, 2006] Frédéric Cuppens, Nora Cuppens-Boulahia, Céline Coma, "O2O: Virtual Private Organizations to Manage Security Policy Interoperability", *2nd International Conference on Information Systems Security* (ICISS 2006), Calcutta (Inde), 19-21 décembre 2006, Springer, LNCS n°4332, pp. 101-115. <http://www.springerlink.com/content/b18p558781307674/> (vérifié le 27/06/09).

[Cuppens *et al.*, 2007] Frédéric Cuppens, Nora Cuppens-Boulahia, Meriam Ben Ghorbel, "High-Level Conflict Management Strategies in Advanced Access Control Models", *Electronic Notes in Theoretical Computer Science* (ENTCS), Vol. 186, pp. 3-26, juin 2007. < http://portal.acm.org/citation.cfm?id=1275288 > (Vérifié le 27/06/09).

[Cuppens & Miège, 2003] Frédéric Cuppens, Alexandre Miège, "Modelling Contexts in the Or-BAC Model", *19th Annual Computer Security Applications Conference* (ACSAC '03), Washington, DC, USA, 2003, pp. 416-425. <http://www.acsac.org/2003/papers/118.pdf> (vérifié le 27/06/09).

[Cuppens & Miège, 2004] Frédéric Cuppens, Alexandre Miège, "Administration Model for Or-BAC", *International Journal of Computer Systems Science and Engineering* (CSSE'04), Vol. 19, No. 3, mai 2004, pp. 754-768. <http://www.springerlink.com/content/xnq0j84wgjy60ve1/> (vérifié le 27/06/09).

[Dacey, 2002] Robert F. Dacey, *Critical Infrastructure Protection: Significant Challenges Need to Be Addressed*, Rapport du "United States General Accounting Office" No. GAO-02-961T, 2002, 64 pp. <http://www.gao.gov/new.items/d02961t.pdf> (Vérifié le 27/06/09).

[Dacey, 2004] Robert F. Dacey, *Critical Infrastructure Protection: Improving Information Sharing With Infrastructure Sectors*, Rapport du "United States General Accounting Office" No. GAO-04-780, 2004, 69 pp. <http://www.gao.gov/new.items/d04780.pdf> (vérifié le 27/06/09).

[Dawson *et al.*, 2000] Steven Dawson, Shelly Qian, Pierangela Samarati, "Providing Security and Interoperation of Heterogeneous Systems", *Distributed Parallel Databases*, Vol. 8, No. 1, janvier 2000, pp. 119-145.

<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.26.931> (vérifié le 13/07/09).

[de Capitani & Samarati, 1996] Sabrina de Capitani di Vimercati, Pierangela Samarati, "Access Control in Federated Systems", *ACM SIGSAC New Security Paradigms Workshop*, Lake Arrowhead, CA, 31 août 1996, pp. 87-99.
<http://citeseer.ist.psu.edu/758070.html> (vérifié le 13/07/09).

[de Paoli & Tisato, 1994] Flavio de Paoli, Francesco Tisato, "CSDL: A Language for Cooperative Systems Design", *IEEE Transactions on Software Engineering*, vol. 20, no. 8, août 1994, pp. 606-616.
<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=00310670> (vérifié le 13/07/09).

[Ferraiolo & Kuhn, 1992] David F. Ferraiolo, D. Richard Kuhn, "Role-Based Access Controls", *15th National Computer Security Conference*, Baltimore MD, 1992, pp. 554-563. <http://csrc.nist.gov/rbac/ferraiolo-kuhn-92.pdf> (vérifié le 27/06/09).

[Gabillon, 2004] Alban Gabillon, "An Authorization Model for XML Databases", *ACM Workshop on Secure Web Service* (SWS'04), pp. 16-28, Fairfax, VA, USA, 2004. <http://portal.acm.org/citation.cfm?id=1111351> (vérifié le 27/06/09).

[Garrone *et al.*, 2007] F. Garrone, C. Brasca, D. Cerotti, D. Codetta Raiteri, A. Daidone, G. Deconinck, S. Donatelli, G. Dondossola, F. Grandoni, M. Kaaniche, T. Rigole, *Analysis of New Control Applications*, Deliverable D2, The CRUTIAL Project, CESI Ricerca, Milan, Italie, 2007, 193 pp.
<crutial.cesiricerca.it/content/les/Documents/Deliverables%20P1/WP1-D2-_nal.pdf> (vérifié le 13/07/09).

[Harrison *et al.*, 1976] Michael A. Harrison, Walter L. Ruzzo, Jeffrey D. Ullman, "Protection in Operating Systems", *Communication of the ACM*, Vol. 19, No. 8, New York, NY, USA, 1976, pp. 461–471.
<http://www.win.tue.nl/~setalle/dtm_tue/harrison-ruzzo-ullman.pdf> (vérifié le 27/06/09).

[Hauser *et al.*, 2008] Carl H. Hauser, David E. Bakken, Ioanna Dionysiou, K. Harald Gjermundrod, Venkata S. Irava, Joel Helkey, Anjan Bose, "Security, Trust, and QoS in Next-Generation Control and Communication for Large Power Systems", *International Journal of Critical Infrastructures 2008*, Vol. 4, No.1/2, 2008, pp. 3–16. <http://www.inderscience.com/storage/f971215102411683.pdf> (Vérifié le 27/06/09).

[Jajodia *et al.*, 2001] Sushil Jajodia, Pierangela Samarati, Maria Luisa Sapino, V. S. Subrahmanian, "Flexible Support for Multiple Access Control Policies", *ACM Trans. Database Systems*, Vol. 26, no. 2, juin 2001, pp. 214-260.
<http://citeseer.ist.psu.edu/old/759042.html> (vérifié le 13/07/09).

[Knight *et al.*, 1998] John C. Knight, Matthew C. Elder, James Flinn, Patrick Marx, *Analysis of Four Critical Infrastructure Applications*, University of Virginia, Computer Science Report No. CS-97-27, 19 septembre 1998. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.1.8509> (vérifié le 13/07/09).

[Laprie *et al.*, 2007] Jean-Claude Laprie, Karama Kanoun, Mohamed Kaâniche, "Modelling Interdependencies Between the Electricity and Information Infrastructures", *26th International Conference* (SAFECOMP 2007), Nüremberg, Germany, 18-21 septembre 2007, pp. 54-67. <http://dx.doi.org/10.1007/978-3-540-75101-4_5> (Vérifié le 27/06/09).

[Lentzner, 2004] Rémy Lentzner, *SQL 3 : Initiation et programmation*, Dunod/01 Informatique, ISBN 2100066331, 214 pp., octobre 2004. <http://www.wikio.fr/livres/sql-3-initiation-et-programmation-9782100066339-692788,b.html> (vérifié le 27/06/09).

[Li & Muntz, 1998] Du Li, Richard Muntz, "COCA: Collaborative Objects Coordination Architecture", *ACM Conference on Computer Supported Cooperative Work* (CSCW '98), Seattle, WA, USA, 14-18 novembre 1998, pp. 179-188. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.55.3814> (vérifié le 13/07/09).

[Masse *et al.*, 2003] John Moteff, Claudia Copeland, and John Fischer, *Critical Infrastructures: What Makes an Infrastructure Critical?*, Rapport du "Congressional Research Service" No. RL31556, 2003, 20 pp. <http://www.fas.org/irp/crs/RL31556.pdf> (Vérifié le 27/06/09).

[Miège, 2005] Alexandre Miège, *Definition of a Formal Framework for Specifying Security Policies : The Or-BAC Model and Extensions*, Doctorat Sécurité informatique, ENST - INFRES Informatique et Réseaux, ENST, 2005. <http://pastel.paristech.org/1376/> (vérifié le 27/06/09).

[Moteff, 1998] John Moteff, *Critical Infrastructures: Background, Policy, and Implementation*, Rapport du "Congressional Research Service" No. Code RL30153, 1998, 45 pp. <http://www.fas.org/sgp/crs/homesec/RL30153.pdf> (Vérifié le 27/06/09).

[Moteff & Parfomak 2006] John Moteff, Paul Parfomak, *Critical Infrastructure and Key Assets: Definition and Identification*, Rapport du "Congressional Research Service" No. RL32631, 2006, 19 pp. <http://fas.org/sgp/crs/RL32631.pdf> (Vérifié le 27/06/09).

[Nasser *et al.*, 2005] Bassem Nasser, Romain Laborde, Abdelmalek Benzekri, François Barrère, Michel Kamel, "Access Control Model for Inter-organizational Grid Virtual Organizations", *On the Move to Meaningful Internet Systems 2005: OTM Workshops*, Chypre, octobre 2005, pp. 537-551.

<http://dx.doi.org/10.1007/11575863_73> (vérifié le 27/06/09).

[Nasser, 2006] Bassem Nasser, *Organisation Virtuelle: Gestion de politiques de contrôle d'accès inter domaine*, Thèse de doctorat, Université Paul Sabatier, Toulouse, France, novembre 2006, 234 pp.

[Nowak, 2009a] Adrien Nowak, *Développement d'une plateforme de collaboration sécurisée basée services web et contrôle d'accès*, rapport de projet de fin d'etude, INSA, Toulouse, juin 2009, 23 pp.

[Nowak, 2009b] Adrien Nowak, *Manuel d'utilisation du démonstrateur CRUTIAL, manuel d'utilisation*, rapport de projet de fin d'étude, INSA, Toulouse, juin 2009, 57 pp.

[OASIS, 2003] OASIS, *XACML Specification*, V1.1, 24 juillet 2003. <www.oasis-open.org/committees/xacml/repository/cs-xacml-specification-1.1.pdf> (Vérifié le 27/06/09).

[OASIS, 2005] OASIS, *UDDI Specifications TC, Universal Description, Discovery and Integration*, v3.0.2, Février 2005. <http://www.uddi.org/pubs/uddi_v3.htm> (vérifié le 13/07/09).

[Ordonnez, 2006] Michael A. Ordonez, *Critical Infrastructure Protection: How to Assess and Provide Remedy to Vulnerabilities in Telecom Hotels*, Rapport de la "Naval Postgraduate School", Monterey, CA, 2006, 109 pp. <http://handle.dtic.mil/100.2/ADA457279> (Vérifié le 27/06/09).

[Powner, 2005] David A. Powner, *Critical Infrastructure Protection: Challenges in Addressing Cybersecurity*, Rapport du "United States General Accounting Office" No. GAO-05-827T, 2005, 26 pp. <http://www.gao.gov/new.items/d05827t.pdf> (Vérifié le 27/06/09).

[Rinaldi *et al.*, 2001] Steven M. Rinaldi, James P. Peerenboom, and Terrence K. Kelly, "Identifying, understanding and Analyzing Critical Infrastructure Interdependencies", *IEEE Control Systems Magazine*, Vol 21, décembre 2001, pp. 11-25. <http://www.ce.cmu.edu/~hsm/im2004/readings/CII-Rinaldi.pdf> (Vérifié le 27/06/09).

[Rinaldi, 2004] Steven M. Rinaldi. "Modeling and simulating critical infrastructures and their interdependencies", *37th Annual Hawaii International Conference on System Sciences* (HICSS'04). Hawaii 2004, 8 pp. <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=01265180>. (Vérifié le 27/06/09).

[RTE, 2005] RTE, *Schéma de développement du réseau public de transport d'électricité 2003-2013*, 2005. <http://www.rte-france.com/htm/fr/mediatheque/telecharge/schema/RTE_envirnmt_schema_developpement_complet.zip> (Vérifié le 27/06/09).

[Sandhu *et al.*, 1996] Ravi Sandhu , Edward Coyne, Hal Feinstein, Charles Youman, "Role-Based Access Control Models", *IEEE computer*, Vol. 29, No. 2, février 1996, pp. 38-47. <http://citeseer.ist.psu.edu/107920.html> (vérifié le 27/06/09).

[Scholand *et al.* 2005] Andrew J. Scholand, John M. Linebarger, Mark A. Ehlen, "Thoughts on Critical Infrastructure Collaboration", *International ACM SIGGROUP Conference on Supporting Group Work* (GROUP '05), Sanibel Island, Floride, USA, 2005, pp. 444-445. <http://www.sandia.gov/nisac/docs/Thoughts%20on%20Critical%20Infrastructure%20Collaboration.pdf> (Vérifié le 27/06/09).

[Shehab *et al.*, 2005] Mohamed Shehab, Elisa Bertino, Arif Ghafoor, "Secure collaboration in mediator-free environments", *12th ACM Conference on Computer and Communications Security* (CCS'05), Alexandria, VA, USA, 7-11 novembre 2005, pp. 58-67. <http://cobweb.ecn.purdue.edu/~iisrl/publications/pub/ccs-shehab.pdf> (Vérifié le 27/06/09).

[Sheldon *et al.*, 2004] Frederick Sheldon, Tom Potok, Andy Loebl, Axel Krings, Paul Oman, "Managing Secure Survivable Critical Infrastructures to Avoid Vulnerabilities", *8th IEEE International Symposium on High-Assurance Systems Engineering* (HASE 2004), 25-26 mars 2004, Tampa, FL, USA. IEEE Computer Society 2004, pp. 293-296. <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1281767&isnumber=28627>. (Vérifié le 27/06/09).

[Sikich *et al.*, 1998] Geary W. Sikich, Critical Infrastructure Vulnerability: An Overview of the Report to the President from the Commission on Critical Infrastructure Protection, 1998. <http://palimpsest.stanford.edu/byauth/sikich/elements.html> (Vérifié le 27/06/09).

[Thomas & Sandhu, 1997] Roshan K. Thomas, Ravi S. Sandhu, "Task-Based Authorization Controls (TBAC): A Family of Models for Active and Enterprise-oriented Authorization Management", *11th IFIP Working Conference on Database Security*, Lake Tahoe, 1997, pp. 166-181. <http://portal.acm.org/citation.cfm?id=679940> (vérifié le 27/06/09).

[Totel *et al.*, 1998] Eric Totel, Jean Paul Blanquart, Yves Deswarte, David Powell, "Supporting multiple levels of criticality", *28th IEEE Fault Tolerant Computing Symposium* (*FTCS-28*), Munich (Germany), Juin 1998, pp. 70–79. < http://portal.acm.org/citation.cfm?id=796884 > (vérifié le 27/07/09).

[TCSEC, 1985] *Trusted Computer System Evaluation Criteria*, United States Department of Defense, décembre 1985, DoD Standard 5200.28-STD. <http://csrc.nist.gov/publications/history/dod85.pdf> (Vérifié le 27/06/09).

[Veríssimo *et al.*, 2008] Paulo Veríssimo, Nuno Ferreira Neves, Miguel Correia, Yves Deswarte, Anas Abou El Kalam, Andrea Bondavalli, Alessandro Daidone, "The CRUTIAL Architecture for Critical Information Infrastructures", *Architecting Dependable Systems V*, Springer, LNCS 5135, 2008, pp. 1-27. <http://www.springerlink.com/content/v5n2137r64l1x036/>. (Vérifié le 27/06/09).

[Wiederhold, 1992] Gio Wiederhold, "Mediators in the Architecture of Future Information Systems", *IEEE Computer*, vol. 25, no. 3, mars 1992, pp. 38-49. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.46.3293> (vérifié le 13/07/09).

[W3C, 2003] W3C, *SOAP*, Version 1.2, W3C Recommendation, juin 2003. <http://www.w3.org/TR/soap12-part0/> (vérifié le 13/07/09).

[W3C, 2004] W3C, *Extensible Markup Language (XML)*, W3C Recommendation, février 2004. <http://www.w3.org/XML/> (vérifié le 13/07/09).

[W3C, 2006] W3C, *Web Services Description Language (WSDL)*, Version 2.0, W3C Candidate Recommendation, mars 2006. <http://www.w3.org/TR/wsdl> (vérifié le 13/07/09).

[Xu *et al.*, 2004] Feng Xu, Guoyuan Lin, Hao Huang, Li Xie, "Role-Based Access Control System for Web Services", *Fourth International Conference on Computer and Information Technology* (CIT'04), Wuhan, Chine, September 14-16, 2004, pp.357-362. <http://www2.computer.org/portal/web/csdl/doi/10.1109/CIT.2004.1357221> (Vérifié le 27/06/09).

Contrôle d'accès pour les grandes infrastructures critiques : Application au réseau d'énergie électrique

En raison de ses vulnérabilités physiques et logiques, une infrastructure critique (IC) peut subir des défaillances, et en raison des interdépendances entre IC, de simples défaillances peuvent avoir des conséquences dramatiques sur l'ensemble de l'infrastructure. Dans notre travail, nous nous concentrons principalement sur les systèmes d'information et de communication (l'IIC : infrastructure d'information critique) dédiés au réseau d'énergie électrique. Nous proposons une nouvelle approche pour répondre aux problèmes de sécurité que rencontre une IIC, plus particulièrement, ceux liés au contrôle d'accès et à la collaboration. Le but est d'offrir à chaque organisation faisant partie de l'IIC la possibilité de collaborer avec les autres, tout en maintenant un contrôle sur ses données et sa politique de sécurité internes. Nous avons modélisé, et développé PolyOrBAC, une plateforme de contrôle d'accès collaboratif, basée sur le modèle de contrôle d'accès OrBAC et sur la technologie des Services Web, cette plateforme est applicable dans le contexte d'une infrastructure critique en général, et plus particulièrement dans le cadre d'un réseau électrique.

MOTS CLES : Infrastructures Critiques, Système d'information, Infrastructure d'Information Critique, Résilience, Sécurité, Contrôle d'Accès, OrBAC, PolyOrBAC, Collaboration, Interopérabilité, Services Web, CRUTIAL, Réseau d'énergie électrique.

Access control for large critical infrastructures: Application to electrical power grid

Because of its physical and logical vulnerabilities, critical infrastructure (CI) may suffer failures, and because of the interdependencies between CIs, simple failures can have dramatic consequences on the entire infrastructure. In our work, we mainly focus on information systems and communications (CII: Critical Information Infrastructure) dedicated to the electrical power grid. We proposed a new approach to address security problems faced by an IIC, particularly those related to access control and collaboration. The goal of this study is to provide each organization belonging to the IIC the opportunity to collaborate with others while maintaining control over its data and its internal security policy. We modeled and developed PolyOrBAC, a platform for collaborative access control, based on the access control model OrBAC and on the Web Services technology, this platform is applicable in the context of a critical infrastructure in general, and more particularly to an electrical power grid.

KEYWORDS: Critical Infrastructure, Information System, Critical Information Infrastructure, Resilience, Security, Access Control, OrBAC, PolyOrBAC, Collaboration, Interoperability, Web Services, CRUTIAL, Electric Power Grid.